# Test Plan for ScaleIO 0.2 Fuel Plugin

[Test Plan for ScaleIO 0.2 Fuel Plugin](#h.9sjw9gygbth8)

[Revision history](#h.6rt6kxi0q67o)

[ScaleIO Plugin](#h.u6oeahaeh3zs)

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

[Limitations](#h.2b0cy5ovolma)

[Test strategy](#h.9jbr2unh6i9)

[Acceptance criteria](#h.nzhxdderq67)

[Test environment, infrastructure and tools](#h.pw7thep7oq4y)

[Product compatibility matrix](#h.cut28xqenhu1)

[Type of testing](#h.l24plbdk72x)

[<Case 1 title>](#h.ud54pvm25504)

[<Case 2 title>](#h.me240mqtn8ni)

[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)

[The Fuel Master node upgrade testing](#h.md720k5dygbn)

[Appendix](#h.sq6iwh656wfp)

# Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Revision date** | **Editor** | **Comment** |
| 0.1 | 11.11.2015 | Adrian Moreno  (adrian.moreno@emc.com) | First draft. |

# 

# ScaleIO Plugin

## 

EMC ScaleIO is a software-only server-based storage area network (SAN) that converges storage and compute resources to form a single-layer, enterprise-grade storage product. ScaleIO storage is elastic and delivers linearly scalable performance. Its scale-out server SAN architecture can grow from a few to thousands of servers.

ScaleIO uses servers’ direct-attached storage (DAS) and aggregates all disks into a global, shared, block storage. ScaleIO features single-layer compute and storage architecture without requiring additional hardware or cooling/ power/space.

Breaking traditional barriers of storage scalability, ScaleIO scales out to hundreds and thousands of nodes and multiple petabytes of storage. The parallel architecture and distributed volume layout delivers a massively parallel system that deliver I/O operations through a distributed system. As a result, performance can scale linearly with the number of application servers and disks, leveraging fast parallel rebuild and rebalance without interruption to I/O. ScaleIO has been carefully designed and implemented with ScaleIO software components so as to consume minimal computing resources.

With ScaleIO, any administrator can add, move, or remove servers and capacity on demand during I/O operations. The software responds automatically to any infrastructure change and rebalances data accordingly across the grid nondisruptively. ScaleIO can add capacity on demand, without capacity planning or data migration and grow in small or large increments and pay as you grow, running on any server and with any storage media.

ScaleIO natively supports all leading Linux distributions and hypervisors. It works agnostically with any solid-state drive (SSD) or hard disk drive (HDD) regardless of type, model, or speed.

**ScaleIO Components**

ScaleIO Data Client (SDC)

* Acts as Block Device Driver
* Exposes volumes to applications
* Service must run to provide access to volumes
* Over TCP/IP

ScaleIO Data Service (SDS)

* Abstracts storage media
* Contributes to storage pools
* Performs I/O operations

ScaleIO Metadata Manager (MDM)

* Not located in the data path
* Provides Monitoring and Configuration management
* Holds cluster-wide component mapping

## ScaleIO Cinder Driver

## ScaleIO includes a Cinder driver, which interfaces between ScaleIO and OpenStack, and presents volumes to OpenStack as block devices which are available for block storage. It also includes an OpenStack Nova driver, for handling compute and instance volume related operations. The ScaleIO driver executes the volume operations by communicating with the backend ScaleIO MDM through the ScaleIO REST Gateway.

## Product compatibility matrix

## This plugin is compatible with Mirantis Fuel 6.1 and EMC Scaleio 1.32.

**Limitations**

ScaleIO 1.32 does not support Ubuntu. Therefore, as Mirantis 7.0 only supports Ubuntu, this plugin is only compatible with Mirantis 6.1 with CentOS.

## System testing

## Install ScaleIO Fuel plugin

|  |  |
| --- | --- |
| Test Case ID | *install\_scaleio\_plugin* |
| Description | Verify that ScaleIO Fuel Plugin can be installed into Fuel Master. |
| Steps | 1. Download it from the Fuel Plugins Catalog. 2. Copy the rpm file to the Fuel Master node:   [root@home ~]# scp fuel-plugin-scaleio-1.0-1.0.0-1.noarch.rpm root@fuel-master:/tmp   1. Log into Fuel Master node and install the plugin using the Fuel CLI.   [root@fuel-master ~]# fuel plugins --install /tmp/fuel-plugin-xenserver-1.0-1.0.1-1.noarch.rpm |
| Expected Result | Verify that the plugin is installed correctly:  [root@fuel-master ~]# fuel plugins  id | name | version | package\_version  ---|-----------------------|---------|----------------  9 | fuel-plugin-scaleio | 1.0.0 | 2.0.0 |

## Prepare nodes

|  |  |
| --- | --- |
| Test Case ID | *prepare\_nodes* |
| Description | *Verify all controller/compute/storage nodes are ready for ScaleIO installation.* |
| Prerequisites | *At least 5 nodes are needed.* |
| Steps | 1. Create 3 or more Controller nodes and name them “Controller 1”, “Controller 2”, and so on. 2. Create 1 or more Compute nodes and name them “Compute 1”, “Compute 2”, and so on. 3. Create 1 or more Controller nodes and name them “Storage 1”, “Storage 2”, and so on. |
| Expected Result | *All nodes are successfully created.* |

## Create an OpenStack environment with ScaleIO Fuel Plugin

|  |  |
| --- | --- |
| Test Case ID | *create\_env* |
| Description | *Verify that an OpenStack environment created with ScaleIO Fuel Plugin has ScaleIO configuration parameters available, fill them, and deploy changes.* |
| Steps | 1. Create a new OpenStack environment from the Fuel Web UI and select “Juno on CentOS 6.5 (2014.2.2­6.1)” in the OpenStack release dropdown list. 2. Hypervisor is default to QEMU, Network is default to Nova Network and Storage is default to Cinder. Other options are disabled. 3. In Nodes Tab, add at least 3 Controller nodes, 1 Compute nodes and 1 Storage node. 4. In Networks tab, configure the network according to your needs and then click on the “Verify Networks” button. 5. In the Settings Tab, scroll down until the “ScaleIO plugin” section, enable it, and fill in all fields. Leave the default value if you do not know the purpose of that field. 6. Click on the “Deploy Changes” button |
| Expected Result | *Deployment is successfully executed.* |

## Verify block storage service

|  |  |
| --- | --- |
| Test Case ID | *verify\_block\_storage* |
| Description | *Verify that all cinder-volume services are identified as ScaleIO.* |
| Steps | 1. Login to Horizon with the admin user when the OpenStack deployment is finished. 2. Check the Storage tab under System Information. |
| Expected Result | *All cinder-volume hosts are identified as ScaleIO.* |

## Create a volume

|  |  |
| --- | --- |
| Test Case ID | *create\_volume* |
| Description | *Verify that volumes are created in OpenStack via ScaleIO.* |
| Steps | 1. Create a new volume from Horizon or the nova CLI and use “sio-thin” as the volume type. 2. Wait until the CLI or Horizon shows that the volume is ready. 3. Log into the ScaleIO Control Panel and verify that there is one volume created and none mapped. |
| Expected Result | *Volumes are created in OpenStack and reflected in ScaleIO* |

## Attach a volume to an instance

|  |  |
| --- | --- |
| Test Case ID | *attach\_volume* |
| Description | *Verify that volumes are attached in OpenStack and ScaleIO reflects them as mapped.* |
| Steps | 1. Attach a volume to an instance from Horizon or the nova CLI. 2. Wait until the CLI or Horizon shows that the volume is attached. 3. Log into the ScaleIO Control Panel and verify that the volume is marked as mapped. |
| Expected Result | *Volumes are attached in OpenStack and reflected in ScaleIO* |

**Verify Fuel Health Checks**

|  |  |
| --- | --- |
| Test Case ID | *verify\_health\_checks* |
| Description | *Ensure that all applicable health checks pass.* |
| Steps | 1. Within the Fuel Master, select the appropriate environment 2. Run all health checks and wait for completion |
| Expected Result | *All health checks pass* |

**Uninstall the plugin with deployed environment**

|  |  |
| --- | --- |
| Test Case ID | *uninstall\_plugin\_with\_deployed\_env* |
| Description | *Verify that ScaleIO Fuel Plugin cannot be uninstalled before all dependent environments are removed.* |
| Steps | fuel plugins ­­remove fuel­plugin­scaleio==0.2.0 |
| Expected Result | *400 Client Error: Bad Request (Can't delete plugin which is enabled for some environment.)* |

**Uninstall the plugin**

|  |  |
| --- | --- |
| Test Case ID | *uninstall\_plugin* |
| Description | *Verify that ScaleIO Fuel Plugin can be successfully uninstalled.* |
| Steps | fuel plugins ­­remove fuel­plugin­scaleio==0.2.0  fuel plugins  id | name | version | package\_version ­­­|­­­­­­­­­­­­­­­­­­­­­­­|­­­­­­­­­|­­­­­­­­­­­­­­­­ |
| Expected Result | *The plugin is removed from Fuel.* |

# Appendix

Provide any links to external resources or documentation here.

|  |  |
| --- | --- |
| **№** | **Resource title** |
| 1 | [ScaleIO Fuel Plugin GitHub repository](https://github.com/emccode/fuel-plugin-scaleio) |
| 2 | [ScaleIO-Cinder Fuel Plugin GitHub repository](https://github.com/emccode/fuel-plugin-scaleio-cinder) |