

# Guide to the ScaleIO Plugin for Fuel 6.1

This document will guide you through the steps of install, configure and use of the ScaleIO Plugin for Fuel

## Introduction

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)** is a lightweight block device driver that exposes ScaleIO shared block volumes to applications. The SDS runs on the same server as the application. This enables the application to issue a IO request and the SDC fulfills it regardless of where the particular blocks physically reside. The SDC communicates with other nodes over TCP/IP-based protocol, so it is fully routable.

**ScaleIO Data Service (SDS)** owns local storage that contributes to the ScaleIO storage pools. An instance of the SDS runs on every node that contributes some, or all its storage space (HDDs, SSDs) to the aggregated pool of storage within the ScaleIO virtual SAN. The role of the SDS is to actually perform the back-end IO operations as requested by an SDC.

**ScaleIO Metadata Manager (MDM)** manages the metadata, SDC, SDS, devices mapping, volumes, snapshots, system capacity including device allocations and/or release of capacity, errors and failures, and system rebuild tasks including rebalancing. The MDM uses a Active/Passive with a tiebreaker component where the primary node is Active, and the secondary is Passive. The data repository is stored in both Active and Passive. Currently, an MDM can manage up to 1024 servers. When several MDMs are present, an SDC may be managed by several MDMs, whereas an SDS can only belong to one MDM. If the MDM does not detect the heartbeat from one SDS, it will initiate a forward-rebuild.

**ScaleIO Gateway** is the HTTP/HTTPS REST endpoint. It is the primary endpoint used by OpenStack to actuate commands against ScaleIO. Due to its stateless nature, we can have multiples instances and easily balance the load.

## 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 Gateway.

## Requirements

Requirement	Version/Comment
Mirantis OpenStack	6.1

- This plugin will deploy an EMC ScaleIO 1.32 cluster on the available nodes and replace the default OpenStack volume backend by ScaleIO.
- A minimum of 3 Controller nodes and 1 Compute node are required.

## Limitations

Currently, this plugin is **only** compatible with Mirantis OpenStack 6.1 and CentOS 6.5 as the base OS.

# Installation Guide

## Install the Plugin

To install the ScaleIO Fuel plugin:

1. Download it from the [Fuel Plugins Catalog](#)
2. Copy the *rpm* file to the Fuel Master node:

```
[root@home ~]# scp fuel-plugin-scaleio-0.3-0.3.0-1.noarch.rpm  
root@fuel-master:/tmp
```

3. Log into Fuel Master node and install the plugin using the [Fuel CLI](#):

```
[root@fuel-master ~]# fuel plugins --install  
/tmp/fuel-plugin-scaleio-0.3-0.3.0-1.noarch.rpm
```

4. Verify that the plugin is installed correctly:

```
[root@fuel-master ~]# fuel plugins
```

id	name	version	package_version
9	fuel-plugin-scaleio	0.3.0	2.0.0

# User Guide

Once the Fuel ScaleIO plugin has been installed (following [Installation Guide](#)), you can create an *OpenStack* environments that uses ScaleIO as the block storage backend.

## Prepare infrastructure

At least 5 nodes are required to successfully deploy Mirantis OpenStack with ScaleIO.

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

Each node shall have at least 2 CPUs, 4GB RAM, 200GB disk, 4 Network interfaces. The 4 networks are:

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

Controllers 1, 2, and 3 will be used as ScaleIO MDMs, being the primary, secondary, and tie-breaker, respectively. Moreover, they will also host the ScaleIO Gateway in HA mode.

All nodes are used as ScaleIO SDS and, therefore, contribute to the default storage pool.

## Select Environment

1. Create a new environment with the Fuel UI wizard. Select "Juno on CentOS 6.5" from OpenStack Release dropdown list and continue until you finish with the wizard.

The screenshot shows the 'Create a new OpenStack environment' wizard in the Fuel UI. The wizard has a sidebar with steps: 'Name and Release' (selected), 'Compute', 'Networking Setup', 'Storage Backends', 'Additional Services', and 'Finish'. The main content area for 'Name and Release' contains a 'Name' field with the value 'scaleio\_env1', an 'OpenStack Release' dropdown menu set to 'Juno on CentOS 6.5 (2014.2.2-6.1)', and a yellow warning box that reads: 'Please make sure your Fuel master node has internet access. To specify alternate repositories, or to create a local mirror, please check the Settings tab before deployment.' Below the warning box is a descriptive text: 'This option will install the OpenStack Juno packages using a CentOS based operating system. With high availability features built in, you are getting a robust, enterprise-grade OpenStack deployment.' At the bottom of the wizard are three buttons: 'Cancel', 'Prev' (disabled), and 'Next' (active).

2. Add VMs to the new environment according to [Fuel User Guide](#) and configure them properly.

## Plugin configuration

1. Go to the Settings tab and scroll down to "ScaleIO plugin" section. You need to fill all fields with your preferred ScaleIO configuration. If you do not know the purpose of a field you can leave it with its default value.


☒ **Fuel plugin for ScaleIO**

Admin password	<input type="password" value="••••••••"/>	Type ScaleIO Admin password
Version	<input type="text" value="latest"/>	ScaleIO version
Gateway password	<input type="password" value="••••••••"/>	Type a password for the gateway
Cluster name	<input type="text" value="cluster1"/>	Name of the ScaleIO cluster
ScaleIO protection domain	<input type="text" value="pd1"/>	Protection domain for ScaleIO
ScaleIO storage pool	<input type="text" value="sp1"/>	First storage pool for ScaleIO
Storage pool size	<input type="text" value="100GB"/>	Size of the Storage Pool.

2. Take the time to review and configure other environment settings such as the DNS and NTP servers, URLs for the repositories, etc.

## Finish environment configuration

1. Go to the Network tab and configure the network according to your environment.
2. Run [network verification check](#)



**Network verification performs the following checks:**

1. L2 connectivity checks between every node in the environment.
2. DHCP discover check on all nodes.
3. Packages repo connectivity check from master node.
4. Packages repo connectivity check from slave nodes via public & admin (PXE) networks.

Verification succeeded. Your network is configured correctly.

Verify Networks

Cancel Changes

Save Settings

3. Press [Deploy](#) button to once you are done with environment configuration.

Deploy Changes

⚠ Please note that environment settings and node configuration will be locked after deployment.

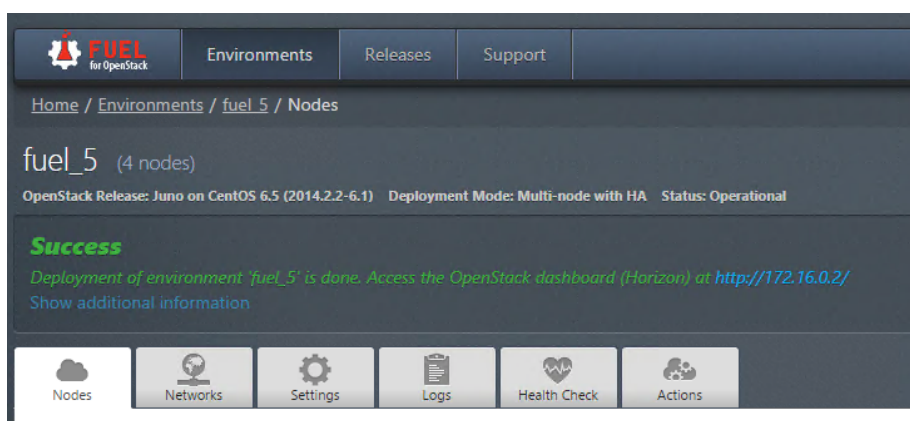
⚠ By default, packages and package updates will be fetched from repositories. Please make sure your Fuel node has internet access.  
To specify alternate repositories, or to create a local mirror, please check the Settings tab before deployment.

Added 6 nodes.

Cancel

Deploy

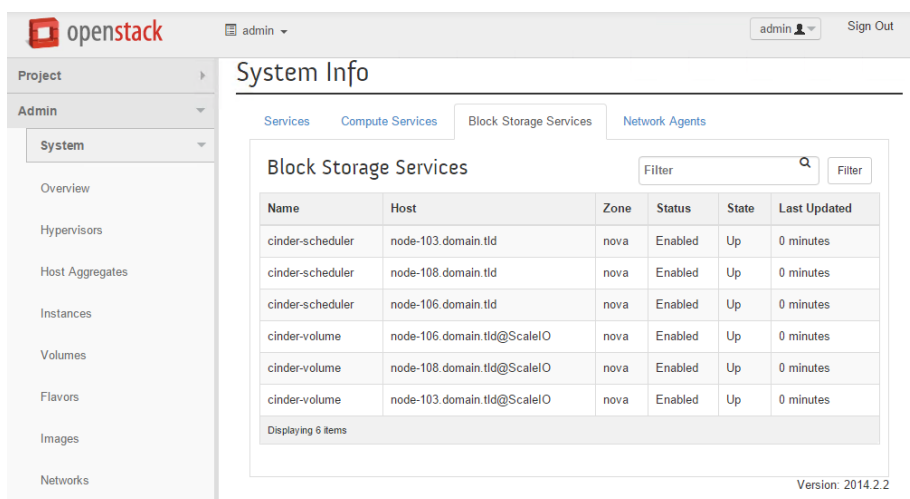
4. After deployment is done, you will see a message indicating the result of the deployment.



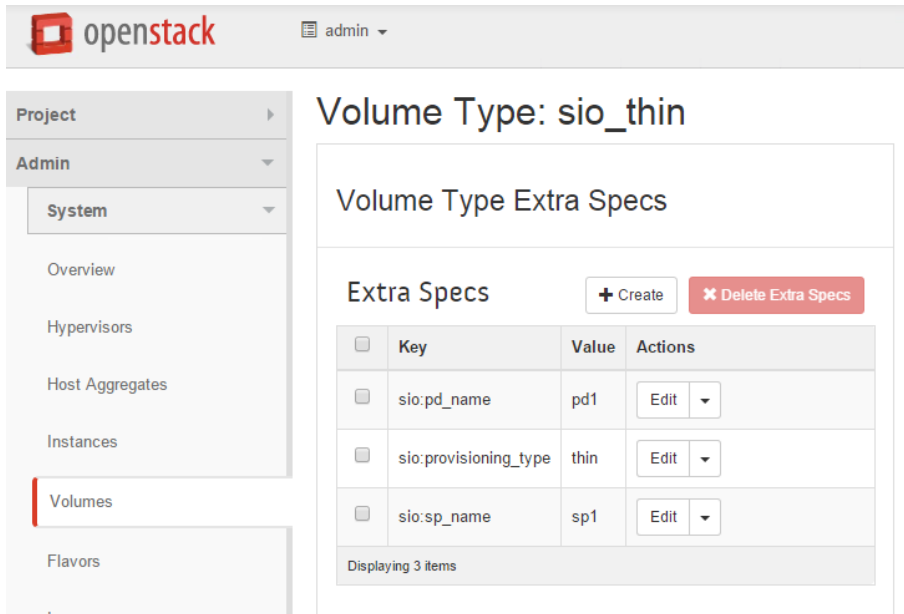
## ScaleIO verification

Once the OpenStack cluster is setup, we can make use of ScaleIO volumes. This is an example about how to attach a volume to a running VM.

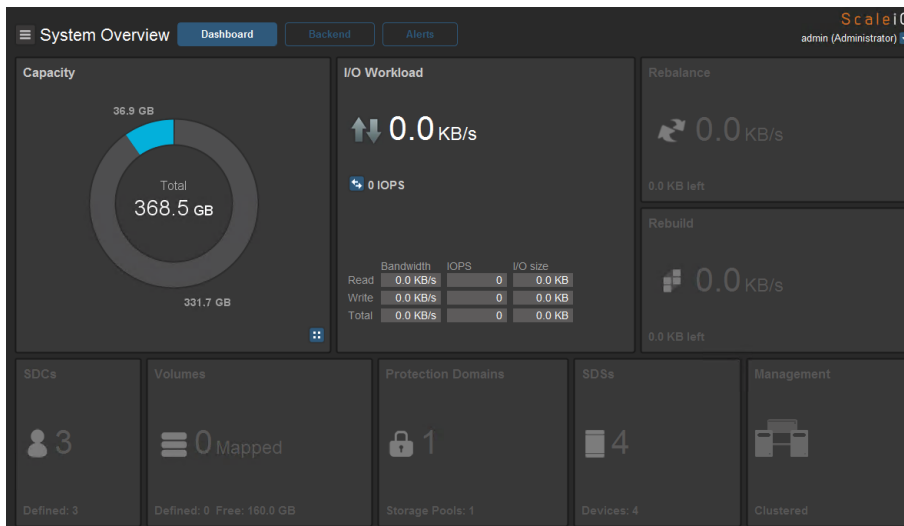
1. Login into the OpenStack cluster:
2. Review the block storage services by navigating to the "Admin -> System -> System Information" section. You should see the "@ScaleIO" appended to all cinder-volume hosts.



3. Review the System Volumes by navigating to "Admin -> System -> Volumes". You should see a volume type called "sio\_thin" with the following extra specs.



4. Open the ScaleIO Control Panel and verify that it successfully reflects the ScaleIO resources:

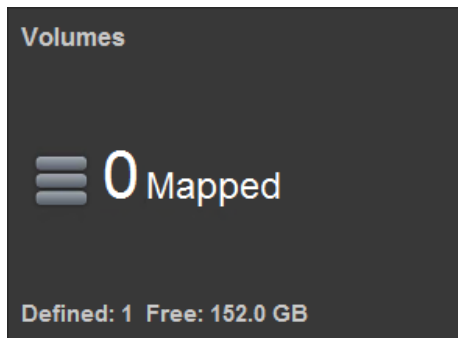


5. Click on the "Backend" tab and verify all SDS nodes:

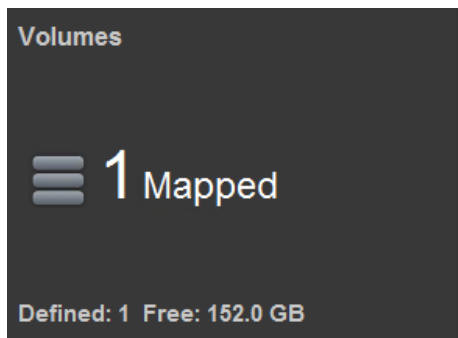
The screenshot shows the 'Backend' tab in the ScaleIO Control Panel. It displays a table with the following columns: Item, Total Capacity, Capacity in Use, I/O, Bandwidth, IOPS, Rebuild, Rebalance, and Alerts. The table lists the following items:

Item	Total Capacity	Capacity in Use	I/O	Bandwidth	IOPS	Rebuild	Rebalance	Alerts
cluster1	368.5 GB	0.0 KB (0.0 %)	0.0 KB/s	0.0 KB/s	0	0.0 KB/s	0.0 KB/s	
pd1	368.5 GB	0.0 KB (0.0 %)	0.0 KB/s	0.0 KB/s	0	0.0 KB/s	0.0 KB/s	
node-86.domain.tld	92.1 GB	0.0 KB (0.0 %)	0.0 KB/s	0.0 KB/s	0	0.0 KB/s	0.0 KB/s	
node-87.domain.tld	92.1 GB	0.0 KB (0.0 %)	0.0 KB/s	0.0 KB/s	0	0.0 KB/s	0.0 KB/s	
node-88.domain.tld	92.1 GB	0.0 KB (0.0 %)	0.0 KB/s	0.0 KB/s	0	0.0 KB/s	0.0 KB/s	
node-90.domain.tld	92.1 GB	0.0 KB (0.0 %)	0.0 KB/s	0.0 KB/s	0	0.0 KB/s	0.0 KB/s	

6. Create a new OpenStack volume using the "sio\_thin" volume type.
7. In the ScaleIO Control Panel, you will see that there is one volume defined but none have been mapped yet.



8. Once the volume is attached to a VM, the ScaleIO Control Panel will reflect the mapping.





# Appendix

1. [ScaleIO OpenStack information](#)
2. [Reference Architecture: EMC Storage Solutions With Mirantis OpenStack](#)
3. [OpenStack @EMC Cheat Sheet](#)