ScaleIO plugin for Fuel Documentation

version 2.0

EMC Corporation

April 06, 2016

Contents

Guide to the ScaleIOv2.0 Plugin for Fuel 6.1 and 7.0 **1**

Plugin Guide 1

Introduction 1

Purpose 1

ScaleIO Overview 1

ScaleIO Components 1

ScaleIO Cinder and Nova Drivers 2

Requirements 2

Limitations 2

Installation Guide 2

Install from source code 2

Install from Fuel Plugins Catalog 3

User Guide 3

Prepare infrastructure 3

Install ScaleIO GUI 4

Select Environment 4

Plugin configuration 5

Finish environment configuration 6

ScaleIO verification 7

Appendix 9

Guide to the ScaleIOv2.0 Plugin for Fuel 6.1 and 7.0

Plugin Guide

Introduction

Purpose

This document will guide you through the steps of install, configure and use of the **ScaleIOv2.0 Plugin** for Fuel. The ScaleIO Plugin is used to:

** deploy and configure a ScaleIO cluster as a volume backend for an OpenStack environment ** configure an Openstack environment to use existing ScaleIO cluster as a volume backend

ScaleIO Overview

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 and Nova Drivers

ScaleIO includes 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
Mirantis OpenStack	7.0

Limitations

1. Plugin is only compatible with Mirantis 6.1 and 7.0.
2. Plugin supports the only Ubuntu environment.
3. The only hyper converged environment is supported - there is no separate ScaleIO Storage nodes.
4. Multi storage backend is not supported.
5. It is not possible to use different backends for persistend and ephemeral volumes.
6. Disks for SDS-es should be unallocated before deployment via FUEL UI or cli.
7. MDMs and Gateways are deployed together and only onto controller nodes.
8. There is no ability to split data network traffic from replication traffic.
9. There is no fault sets support.
10. **Adding and removing node(s) to/from the OpenStack cluster won't re-configure the ScaleIO.**

    This is a limitation of the Fuel Plugin Framework which doesn't trigger task when those actions are performed. One exception here is new SDC compupents are registered in ScaleIO automatically. Registering other components is possible either vi running the task 'update_hosts' on controllers via FUEL cli or via adding component into ScaleIO cluster vi ScaleIO cli (scli) that available on controller nodes. For removal nodes it is needed to remove nodes from ScaleIO vi ScaleIO cli (scli).

Installation Guide

Install from source code

To install the ScaleIO Plugin from source code, you first need to prepare an environment to build the RPM file of the plugin. The recommended approach is to build the RPM file directly onto the Fuel Master node so that you won't have to copy that file later.

Prepare an environment for building the plugin on the **Fuel Master node**.

1. Install the standard Linux development tools:

   ```
   yum install createrepo rpm rpm-build dpkg-devel git
   ```

2. Install the Fuel Plugin Builder. To do that, you should first get pip:

   ```
   easy_install pip
   ```

3. Then install the Fuel Plugin Builder (the *fpb* command line) with *pip*:

   ```
   pip install fuel-plugin-builder
   ```

4. Clone the ScaleIO Plugin git repository (note the *--recursive* option):

   ```
   git clone https://github.com/cloudscaling/fuel-plugin-scaleio.git
   cd fuel-plugin-scaleio
   ```

5. Check that the plugin is valid:

```
fpb --check .
```

6. Build the plugin:

```
fpb --build .
```

7. Install plugin:

```
fuel plugins --install ./scaleio-2.0-2.0-1.noarch.rpm
```

8. Verify that the plugin is installed correctly:

```
[root@fuel-master ~]#fuel plugins
id | name                  | version | package_version
---|-----------------------|---------|----------------
 1 | scaleio               | 2.0     | 2.0.0
```

Install from Fuel Plugins Catalog

To install the ScaleIOv2.0 Fuel plugin:

1. Download it from the Fuel Plugins Catalog
2. Copy the *rpm* file to the Fuel Master node:

```
[root@home ~]# scp scaleio-2.0-2.0-1.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/scaleio-2.0-2.0-1.noarch.rpm
```

4. Verify that the plugin is installed correctly:

```
[root@fuel-master ~]# fuel plugins
id | name                  | version | package_version
---|-----------------------|---------|----------------
 1 | scaleio               | 2.0     | 2.0.0
```

User Guide

Once the Fuel ScaleIOv2.0 plugin has been installed (following the 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, 3 Network interfaces. The 3 interfaces will be used for the following purposes:

1. Admin (PXE) network: Mirantis OpenStack uses PXE booting to install the operating system, and then loads the OpenStack packages for you.
2. Public, Management and Storage networks: All of the OpenStack management traffic will flow over this network (“Management” and “Storage” will be separated by VLANs), and to re-use the network it will also host the public network used by OpenStack service nodes and the floating IP address range.

3. Private network: This network will be added to Virtual Machines when they boot. It will therefore be the route where traffic flows in and out of the VM.

In case of new ScaleIO cluster deployment 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. Additionally Cotrollers should play Cinder role, because of lack of custom role support in Fuel6.1. All Compute nodes are used as ScaleIO SDS and, therefore, contribute to the default storage pool. It is possible to enable SDS on Controllers node, that is usefull for evaluation and test puproses but usually it is not recommended in production environment. All nodes that will be used as ScaleIO SDS should have equal disk configuration. All disks that will be used as SDS devices should be unallocated in Fuel.

In case of existing ScaleIO cluster deployment the plugin deploys ScaleIO SDC component onto Compute and Cinder nodes and configures OpenStack Cinder and Nova to use ScaleIO as the block storage backend.

The ScaleIO cluster will use the storage network for all volume and cluster maintenance operations.

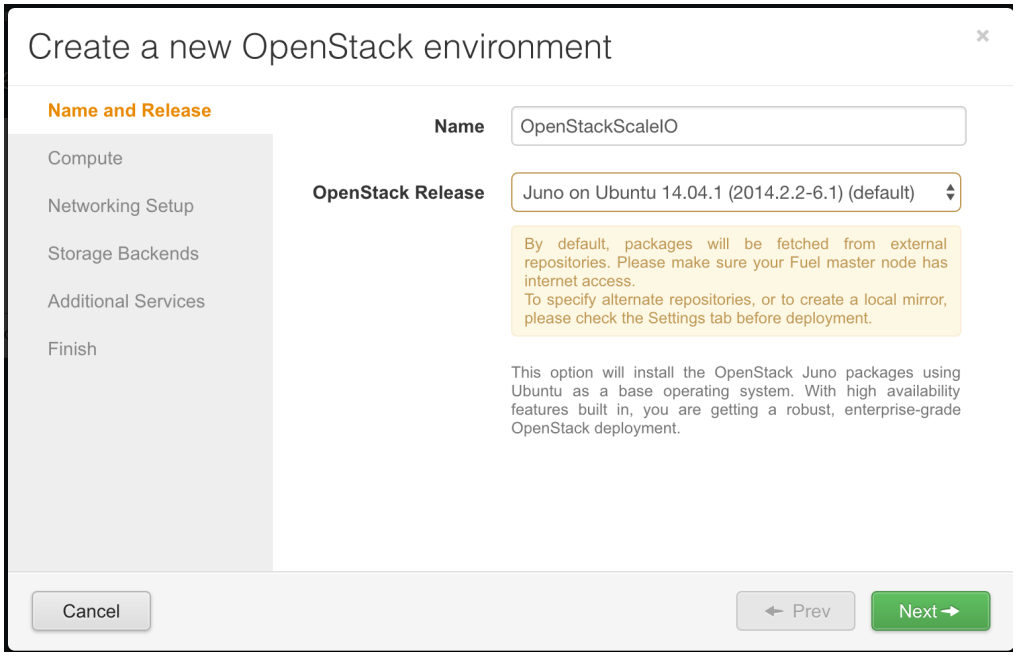
Install ScaleIO GUI

It is recommended to install the ScaleIO GUI to easily access and manage the ScaleIO cluster.

1. Make sure the machine in which you will install the ScaleIO GUI has access to the Controller nodes.
2. Download the ScaleIO for your operating system from the following link: http://www.emc.com/products-solutions/trial-software-download/scaleio.htm
3. Unzip the file and install the ScaleIO GUI component.
4. Once installed, run the application and you will be prompted with the following login window. We will use it once the deployment is completed.

Select Environment

1. Create a new environment with the Fuel UI wizard. Select "Juno on Ubunu 14.04" from OpenStack Release dropdown list and continue until you finish with the wizard.



2. Add VMs to the new environment according to Fuel User Guide and configure them properly.

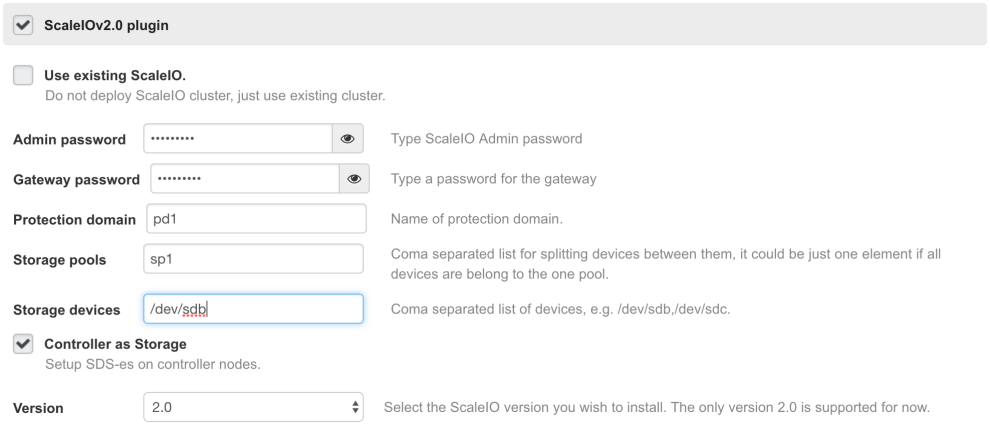
Plugin configuration

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.

2. In order to deploy new ScaleIO cluster together with OpenStack

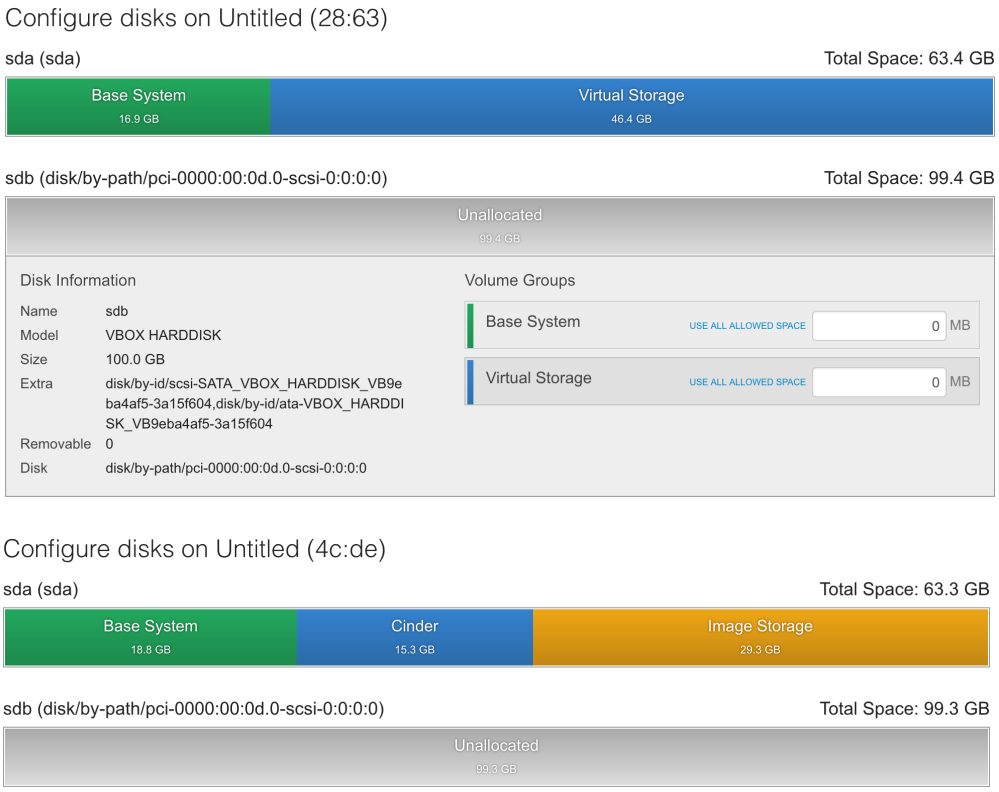
   a. Disable the checkbox 'Use existing ScaleIO'

   b. Provide Admin passwords for ScaleIO MDM and Gateway, list of Storage devices to be used as ScaleIO SDS storage devices. Optionally you can provide protection domain name and storage pool names.



   c. In case you want to speciafy different storage pools for different devices provide a list of pools corresponding to device paths, e.g. 'pool1,pool2' and '/dev/sdb,/dev/sdc' will assign /dev/sdb for the pool1 and /dev/sdc for the pool2.

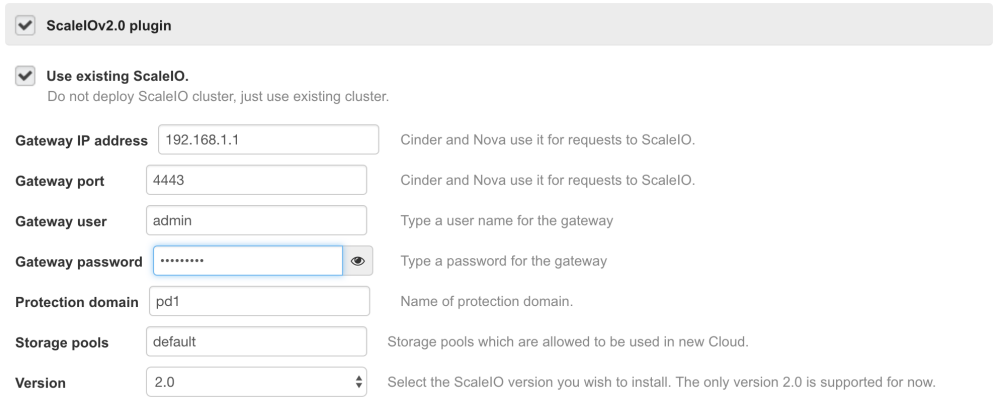
   d. Make disks for ScaleIO SDS devices unallocated. This disks will be cleand up and added to SDS-es as storage devices. Note, that because of current Fuel framwork limitation it is needed to keep some space for Cinder and Nova roles.



3. In order to use existing ScaleIO cluster

   a. Enable checkbox 'Use existing ScaleIO'

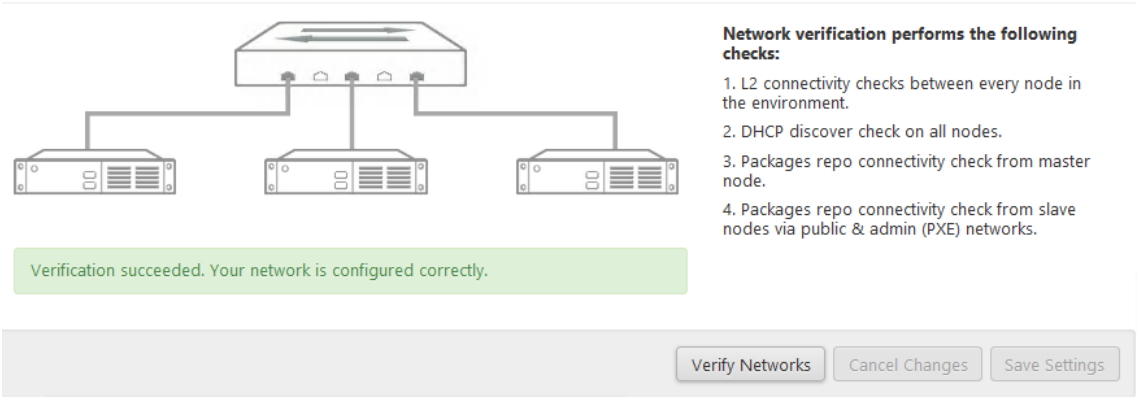
   b. Provide IP address and password for ScaleIO Gateway, protection domain name and storage pool names that will be allowed to be used in OpenStack. The first storage pool name will become the default storage pool for volumes.



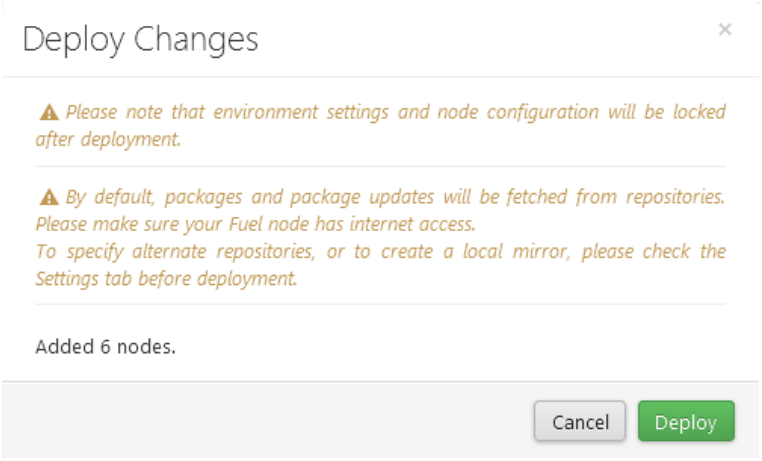
4. Take the time to review and configure other environment settings such as the DNS and NTP servers, URLs for the repositories, etc.

Finish environment configuration

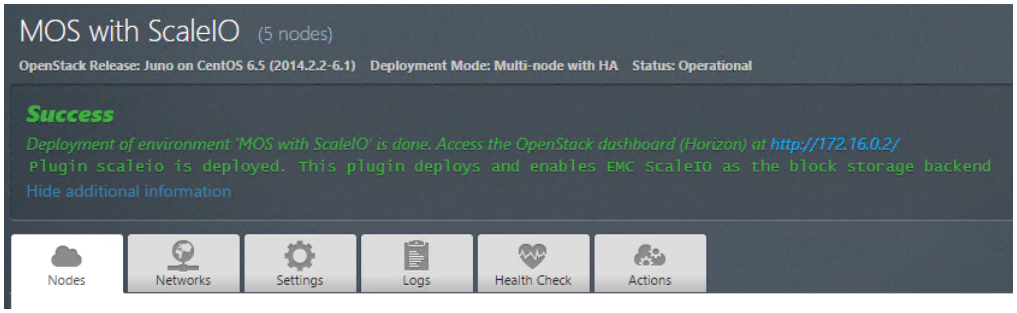
1. Go to the Network tab and configure the network according to your environment.
2. Run network verification check



3. Press Deploy button once you have finished reviewing the environment configuration.



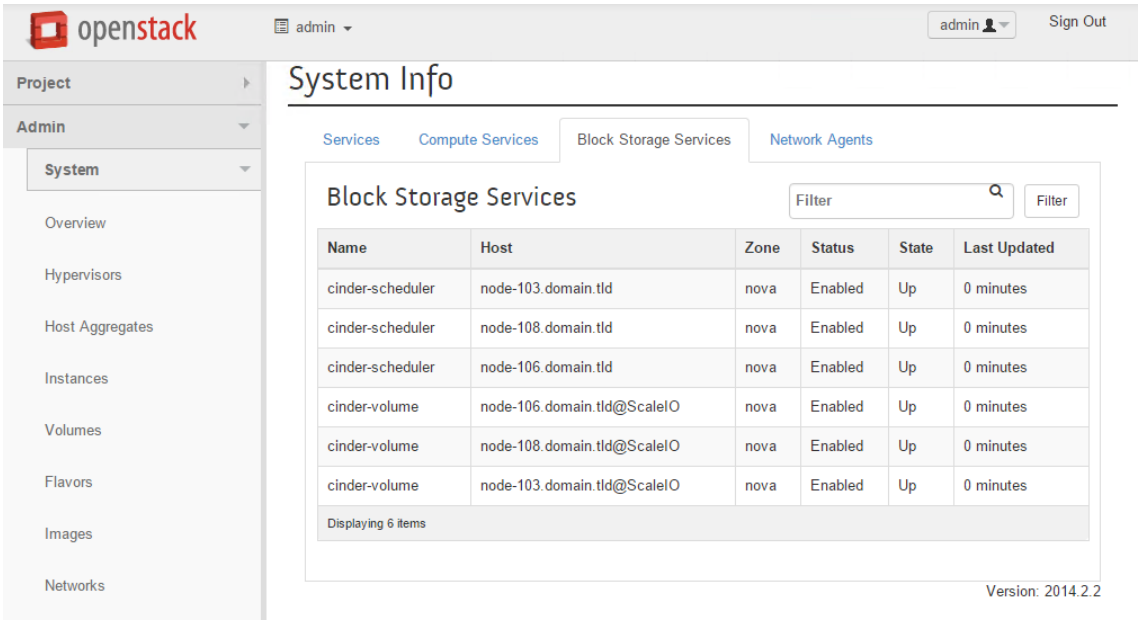
4. After deployment is done, you will see a message indicating the result of the deployment.



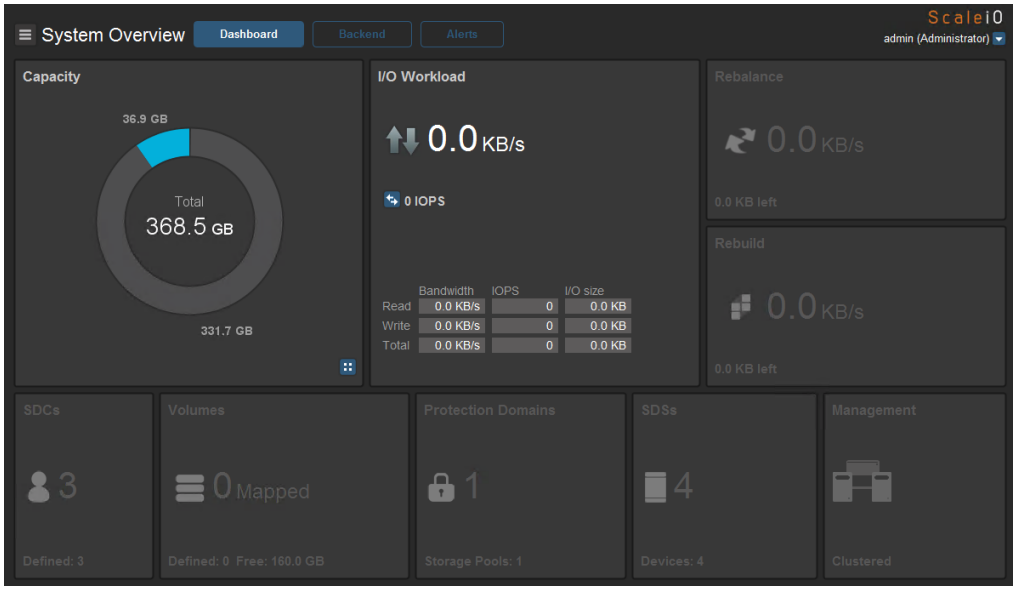
ScaleIO verification

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.

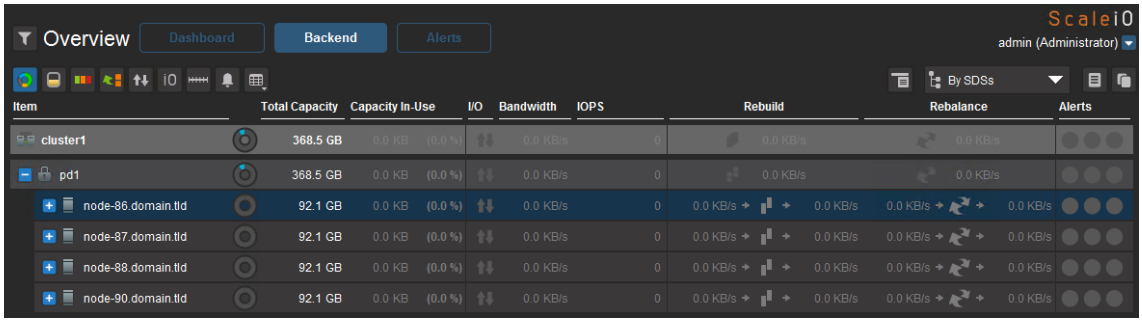
1. Login into the OpenStack cluster:
2. Review the block storage services by navigating to the "Admin -> System -> System Information" section. You should see the "@ScaleIO" appended to all cinder-volume hosts.



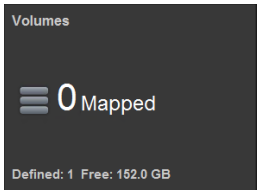
3. In the ScaleIO GUI (see Install ScaleIO GUI section), enter the IP address of the primary controller node, username *admin*, and the password you entered in the Fuel UI.
4. Once logged in, verify that it successfully reflects the ScaleIO resources:



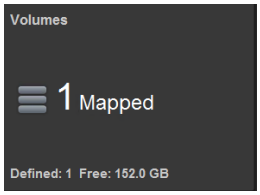
5. For the case of new ScaleIO cluster deployment click on the "Backend" tab and verify all SDS nodes:



6. Create a new OpenStack volume (ScaleIO backend is used by default).
7. In the ScaleIO GUI, 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 GUI will reflect the mapping.



Appendix

1. ScaleIO OpenStack information
2. Reference Architecture: EMC Storage Solutions With Mirantis OpenStack
3. OpenStack @EMC Cheat Sheet