The ScaleIO plugin for Fuel Documentation

Release 2.1-2.1.2-1

EMC Corporation

October 27, 2016

CONTENTS

1 Plugin Guide **1**
1.1 Release Notes v2.1.2 . . . 1
1.2 Release Notes v2.1.1 . . . 1
1.3 Introduction . . . 1
1.4 Installation Guide . . . 3
1.5 User Guide . . . 3
1.6 Appendix . . . 13

CHAPTER 1

PLUGIN GUIDE

Release Notes v2.1.2

New Features

1. Non-hyperconverged deployment support. Separate ScaleIO role for ScaleIO Storage nodes. To enable this feature there is appropriate check-box in the plugin's settings. Note, that although there is a role for ScaleIO Storage the user still has to point devices in the 'Storage devices' settings. The role frees user from making ScaleIO disks unassigned. User can use devices with ScaleIO role as 'Storage devices' (with mapping to different storage pools) as well as 'XtremCache devices' (it is expected that user is aware which device are SSD actually, the plugin does not perform such check).

Release Notes v2.1.1

New Features

1. Mirantis Fuel 9.0 support.
2. RAM Cache (RMCache) support.
3. Using special FronEnd ScaleIO user in Cinder and Nova to access ScaleIO cluster instead of the 'admin' user.
4. Ability to keep Glance images on ScaleIO.

Fixed Bugs

1. Fixed algorithm of protection domain auto-creation if number of SDS-es becomes larger than a threshold in the plugin settings.

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.

Xtrem Cache (RFCache) is the component enabling caching on PCI flash cards and/or SSDs thus accelerating the reads of SDS's HDD devices. It is deployed together with SDS component.

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 com-

pute 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	8.0
Mirantis OpenStack	9.0

Limitations

1. Plugin is compatible with Mirantis Fuel 8.0 and 9.0.
2. Plugin supports only Ubuntu environment.
3. Multi storage backend is not supported.
4. It is not possible to use different backends for persistent and ephemeral volumes.
5. In hyper-converged deployment disks for SDS-es should be unallocated before deployment via FUEL UI or cli.
6. MDMs and Gateways are deployed together and only onto controller nodes.
7. Adding and removing node(s) to/from the OpenStack cluster won't re-configure the ScaleIO.

Installation Guide

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-2.1-2.1.2-1.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-2.1-2.1.2-1.noarch.rpm
```

4. Verify that the plugin is installed correctly:

```
[root@fuel-master ~]# fuel plugins
id | name                   | version | package_version
---|------------------------|---------|----------------
 1 | scaleio                | 2.1.2   | 3.0.0
```

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 (for 3-controllers mode cluster).

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. Each node which is supposed to host ScaleIO SDS should have at least one empty disk of minimum 100GB size. 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 for hosting ScaleIO MDM and ScaleIO Gateway services. Cinder role should be deployed if ScaleIO volume functionality is required. All Compute nodes are used as ScaleIO SDS. It is possible to enable ScaleIO SDS on Controllers node. Keep in mind that 3 SDSs is a minimal required configuration so if you have less than 3 compute nodes you have to deploy ScaleIO SDS on controllers as well. 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.

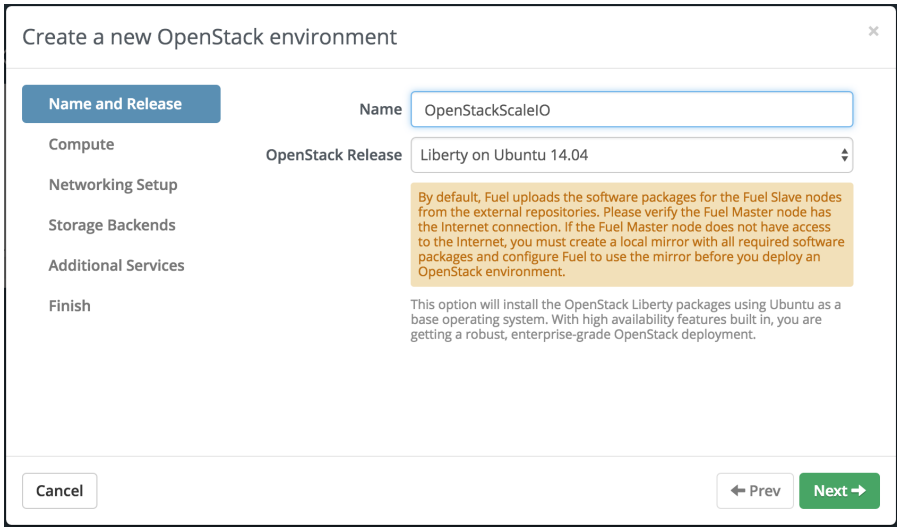
Install ScaleIO GUI

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

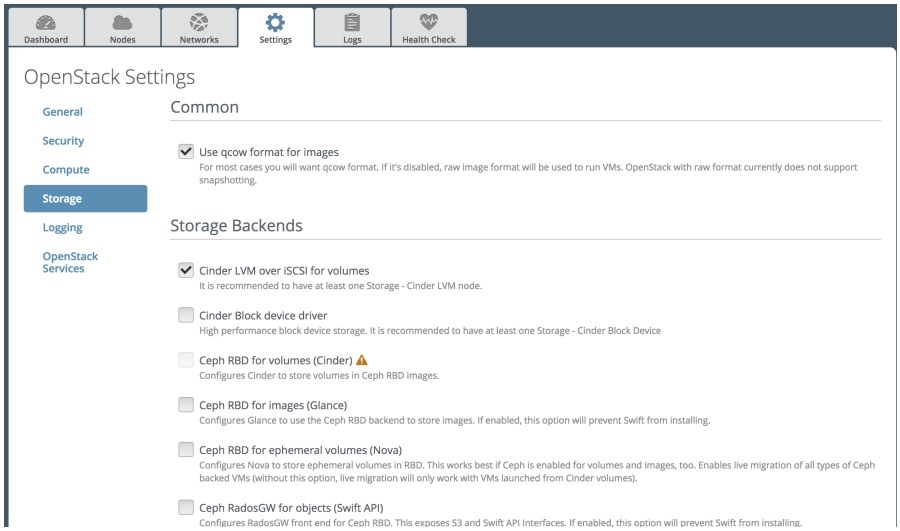
#. Create a new environment with the Fuel UI wizard. From OpenStack Release dropdown list select "Liberty on Ubunu 14.04" and continue until you finish with the wizard.



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

Plugin configuration

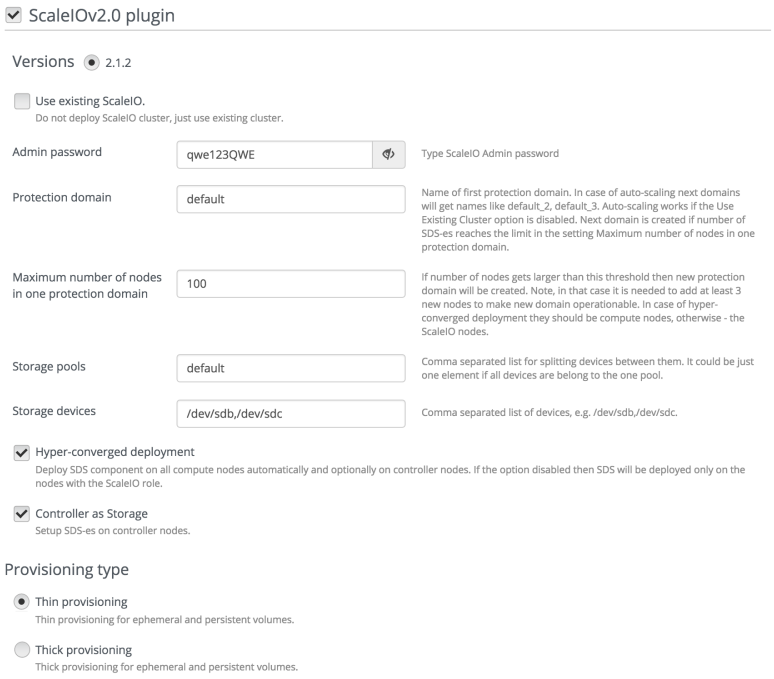
1. Go to the Settings tab and then go to the section Storage. You need to fill all fields with your preferred ScaleIO configuration. If you do not know the purpose of a field you can leave it with its default value.

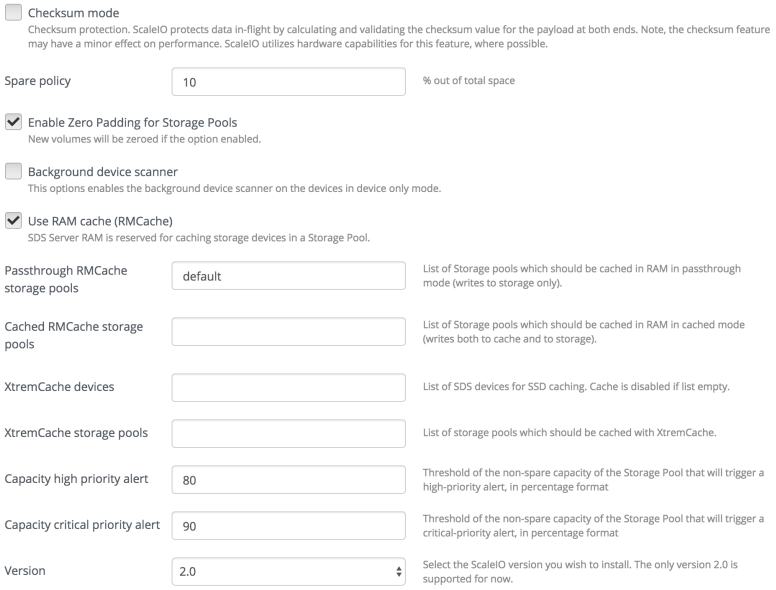


2. In order to deploy new ScaleIO cluster together with OpenStack

   a. Disable the checkbox ‘Use existing ScaleIO’

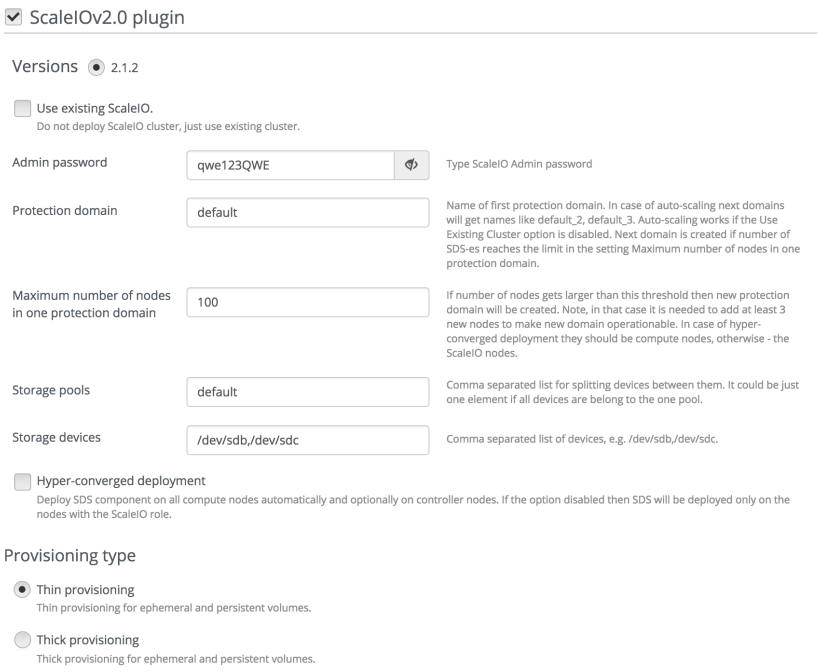
   b. Provide Admin passwords for ScaleIO MDM and Gateway, list of Storage devices to be used as ScaleIO SDS storage devices. Optionally you can provide protection domain name and storage pool names.

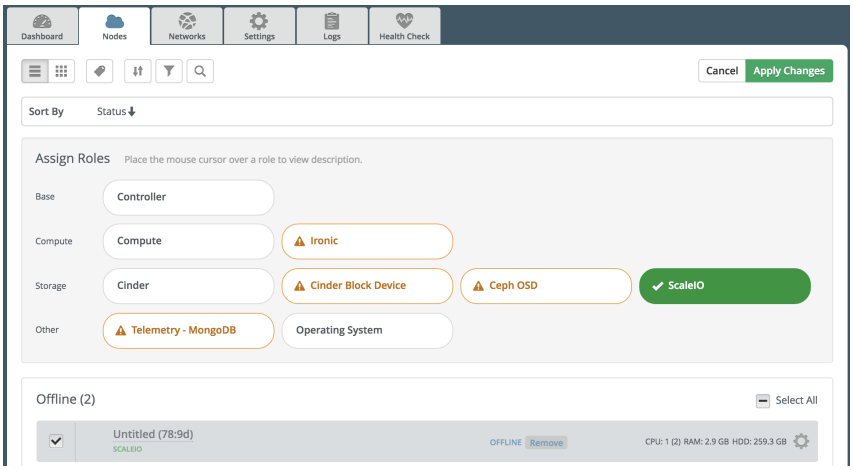


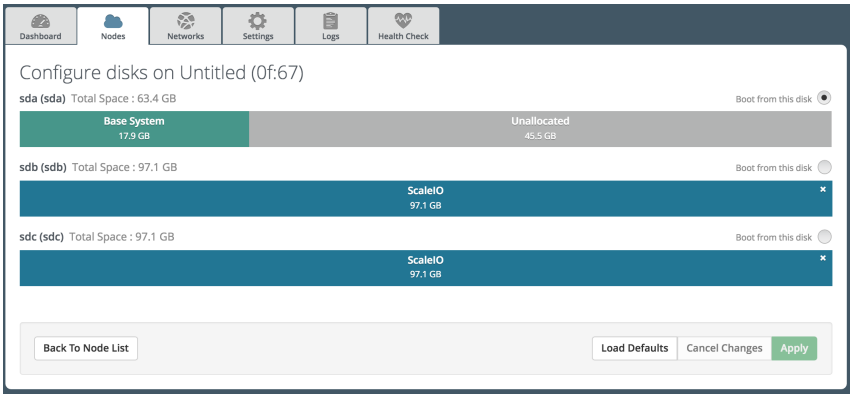


c. In order to use separate ScaleIO storage nodes disable check-box 'Hyper-converged deployment'.

In such kind of depllyment ScaleIO SDS component will be deployed only on the nodes with ScaleIO role. Although there is a role for ScaleIO Storage the user still has to point devices in the 'Storage devices' settings. The role frees user from making ScaleIO disks unassigned. User can use devices with ScaleIO role as 'Storage devices' (with mapping to different storage pools as described below) as well as 'XtremCache devices' (it is expected that user is aware which device are SSD actually, the plugin does not perform such check).

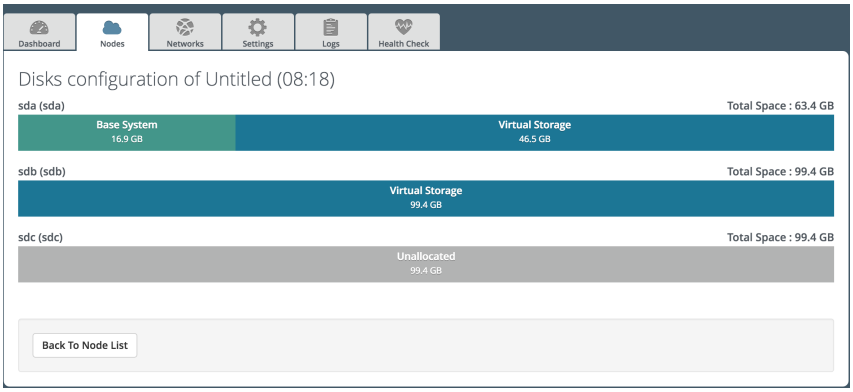


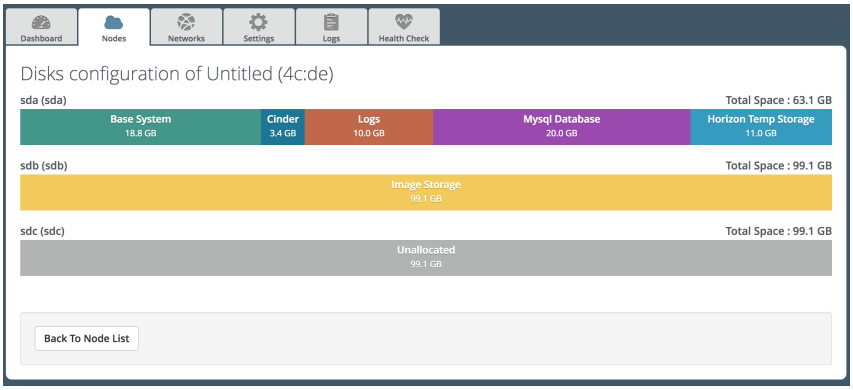




d. In case of hyper-converged deployment enabled make disks for ScaleIO SDS devices unallocated.

These disks will be cleaned up and added to SDSs as storage devices. Note, that because of current Fuel framework limitation it is needed to keep some space for Cinder and Nova roles.



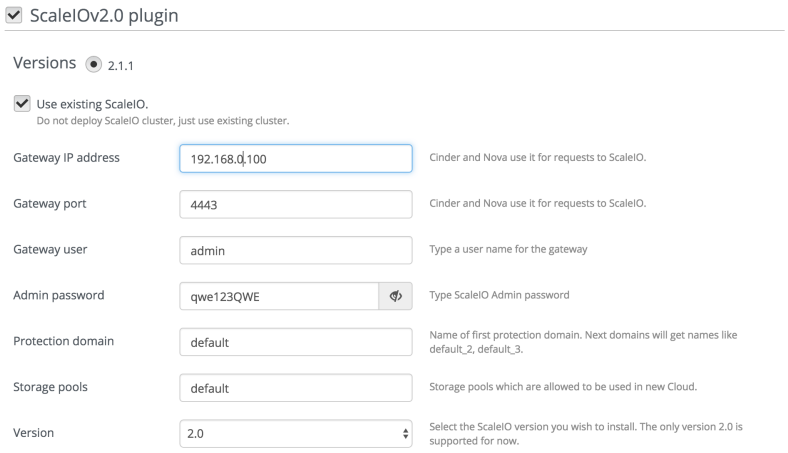


e. In case you want to specify different storage pools for different devices provide a list of pools corresponding to device paths, e.g. 'pool1,pool2' and '/dev/sdb,/dev/sdc' will assign /dev/sdb for the pool1 and /dev/sdc for the pool2.

3. In order to use existing ScaleIO cluster

a. Enable checkbox 'Use existing ScaleIO'

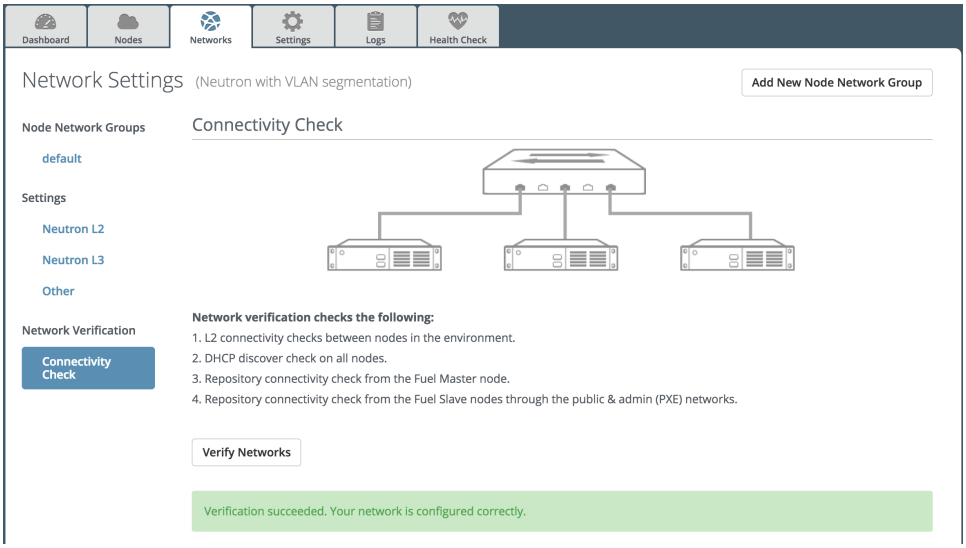
b. Provide IP address and password for ScaleIO Gateway, protection domain name and storage pool names that will be allowed to be used in OpenStack. The first storage pool name will become the default storage pool for volumes.



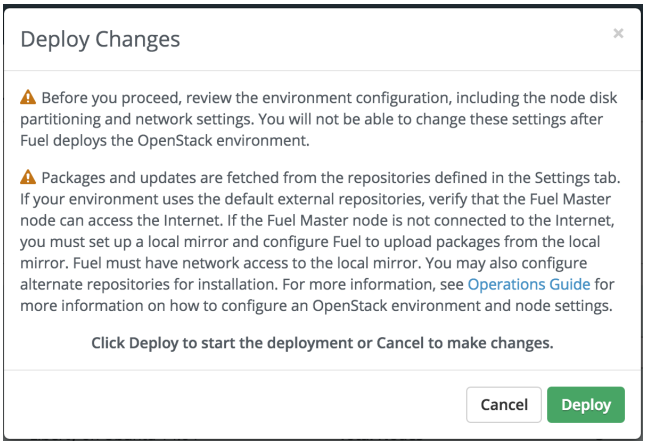
4. Take the time to review and configure other environment settings such as the DNS and NTP servers, URLs for the repositories, etc.

Finish environment configuration

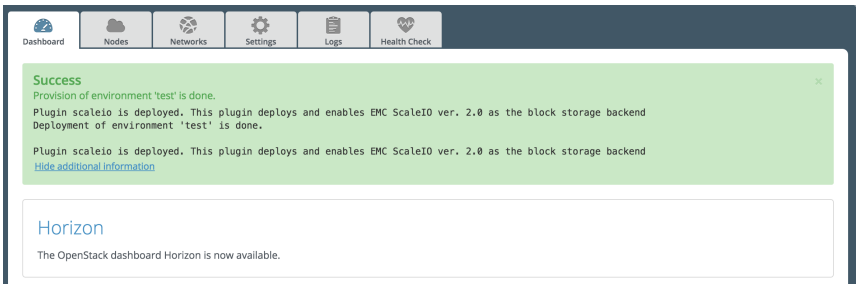
1. Go to the Network tab and configure the network according to your environment.
2. Run network verification check



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.

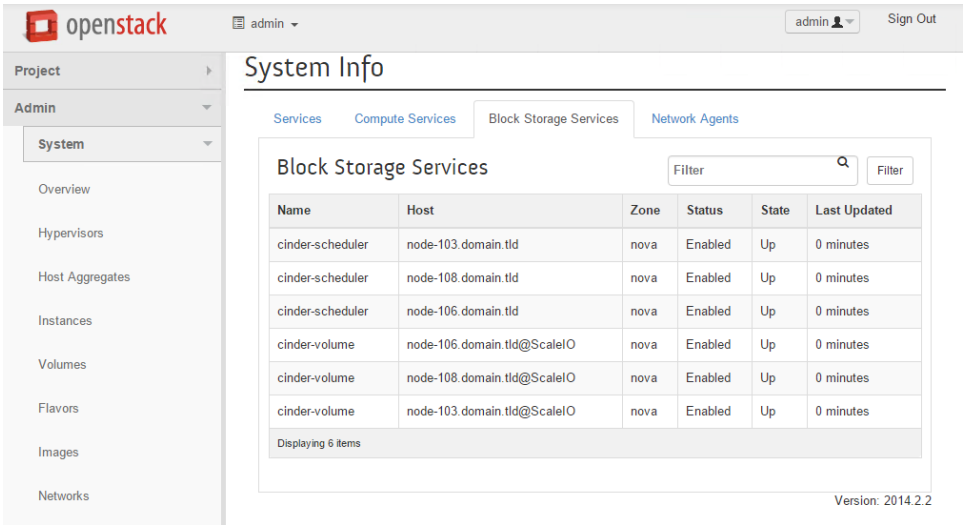


ScaleIO verification

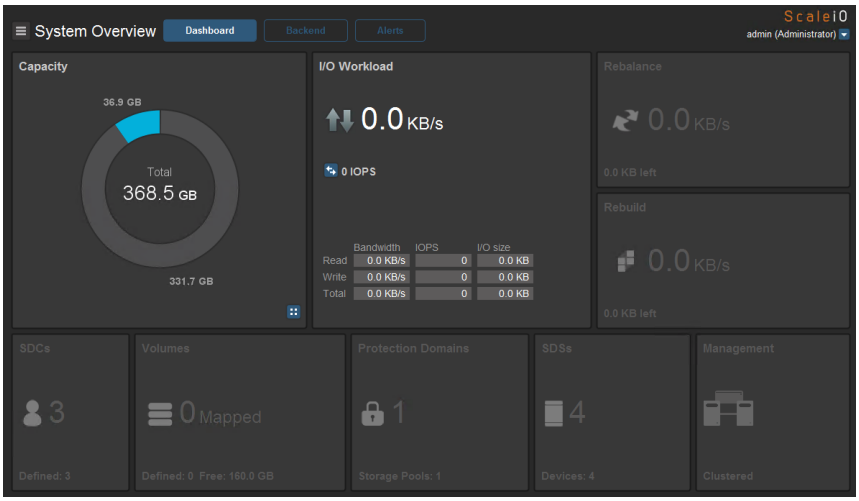
Once the OpenStack cluster is set up, you can make use of ScaleIO volumes. This is an example about how to attach a volume to a running VM.

1. Perform OpenStack Health Check via FUEL UI. Note, that it is needed to keep un-selected tests that are related to running of instances because they use a default instance flavour but ScaleIO requires a flavour with volume sizes that are multiple of 8GB. FUEL does not allow to configure these tests from the plugin.

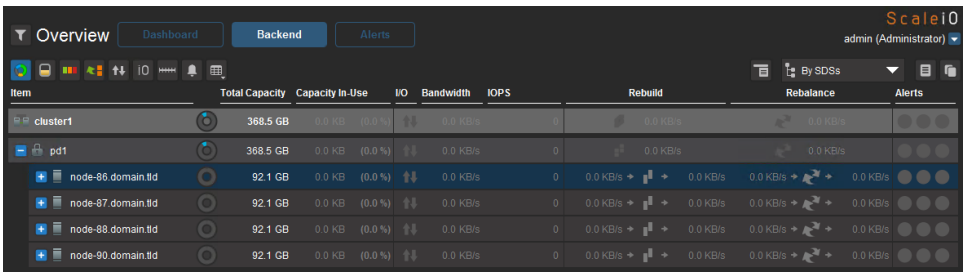
2. Login into the OpenStack cluster:
3. Review the block storage services by navigating to the “Admin -> System -> System Information” section. You should see the “@ScaleIO” appended to all cinder-volume hosts.



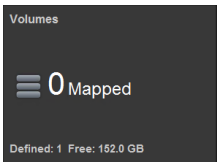
4. Connect to ScaleIO cluster in the ScaleIO GUI (see *Install ScaleIO GUI section*). In case of new ScaleIO cluster deployment use the IP address of the master ScaleIO MDM (initially it’s the controller node with the minimal IP-address but master MDM can switch to another controller), username *admin*, and the password you entered in the Fuel UI.
5. Once logged in, verify that it successfully reflects the ScaleIO resources:



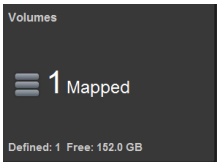
6. For the case of new ScaleIO cluster deployment click on the “Backend” tab and verify all SDS nodes:



7. Create a new OpenStack volume (ScaleIO backend is used by default).
8. In the ScaleIO GUI, you will see that there is one volume defined but none have been mapped yet.



9. Once the volume is attached to a VM, the ScaleIO GUI will reflect the mapping.



Troubleshooting

1. Deployment cluster fails.

- Verify network settings.
- Ensure that the nodes have internet access.
- Ensure that there are at least 3 nodes with SDS in the cluster. All Compute nodes play SDS role, Controller nodes play SDS role in case if the option 'Controller as Storage' is enabled in the Plugin's settings.
- For the nodes that play SDS role ensure that disks which are listed in the Plugin's settings 'Storage devices' and 'XtremCache devices' are unallocated and their sizes are greater than 100GB.
- Ensure that controller nodes have at least 3GB RAM.

2. Deploying changes fails with timeout errors if remove a controller node (only if there were 3 controllers in cluster).

- Connect via ssh to the one of controller nodes
- Get MDM IPs:

```
cat /etc/environment | grep SCALEIO_mdm_ips
```

- Request ScaleIO cluster state

```
scli --mdm_ip <ip_of_alive_mdm> --query_cluster
```

- If cluster is in Degraded mode and there is one of Slave MDMs is disconnected then switch the cluster into the mode '1_node':

```
scli --switch_cluster_mode --cluster_mode 1_node
     --remove_slave_mdm_ip <ips_of_slave_mdms>
     --remove_tb_ip <ips_of_tie_breakers>
Where ips_of_slave_mdms and ips_of_tie_breakers are comma separated lists
of slave MDMs and Tie Breakers respectively (IPs should be taken from
query_cluster command above).
```

3. **ScaleIO cluster does not see new SDS after deploying new Compute node.** It is needed to run update hosts task on controller nodes manually on the FUEL master node, e.g. 'fuel –env 5 node –node-id 1,2,3 –task update_hosts'. This is because FUEL does not trigger plugin's tasks after Compute node deploymet.

4. **ScaleIO cluster has SDS/SDC components in disconnected state after nodes deletion.** See previous point.
5. **Other issues.** Ensure that ScaleIO cluster is operational and there are storage pool and protection domain available. For more details see ScaleIO user guide.

Appendix

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