Test Plan for ScaleIO 1.0.0 Fuel Plugin

Revision history

Version	Revision date	Editor	Comment
0.2	11.11.2015	Adrian Moreno (adrian.moreno@emc.com)	First draft.
0.3	12.11.2015	Adrian Moreno (adrian.moreno@emc.com)	Additional test cases and formatting.
0.4	01.11.2015	Adrian Moreno (adrian.moreno@emc.com)	Additional test cases

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.

Developer's specification

Is available on GitHub repository.

Limitations

Due to some software limitations, this plugin is currently only compatible with Mirantis 6.1 and CentOS.

Test strategy

The ScaleIO plugin creates a GUI element to collect the information necessary to deploy and configure EMC ScaleIO in the cluster nodes. The testing strategy is to confirm that all options in the GUI are handled properly and ScaleIO is successfully deployed and Cinder is properly configure to use the ScaleIO cluster as the block storage service.

Acceptance criteria

All tests should pass.

Test environment, infrastructure and tools

The test environment shall include 5 nodes. The following designations for the nodes:

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

Product compatibility matrix

ScaleIO Plugin version	Compatible Fuel version	OpenStack and OS Version	ScaleIO version
1.0.0	6.1	Juno on CentOS 6.5	1.32

System testing

Build ScaleIO Fuel plugin

Test Case ID	*build_scaleio_plugin*
Description	Verify that ScaleIO Fuel plugin builds successfully.

Steps	1. Clone the repository with all its submodules
	
	`git clone --recursive https://github.com/openstack/fuel-plugin-scaleio`
	
	2. Build the plugin using the “fpb” command tool.
	
	`fpb --build fuel-plugin-scaleio`

Expected Result	Outputs the message ‘Plugin is built’ and the “scaleio-1.0-1.0.0-1.noarch.rpm” package is created in the “fuel-plugin-scaleio” directory.

Install ScaleIO Fuel plugin

Test Case ID	*install_scaleio_plugin*
Description	Verify that ScaleIO Fuel Plugin can be installed into Fuel Master.

Steps	3. Download the plugin from the Fuel Plugins Catalog or build it from source.
	4. 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`
	
	5. Log into Fuel Master node and install the plugin using the Fuel CLI.
	
	`[root@fuel-master ~]# fuel plugins --install /tmp/fuel-plugin-scaleio-1.0-1.0.0-1.noarch.rpm`

Expected Result	Verify that the plugin is installed correctly:
	
	`[root@fuel-master ~]# fuel plugins`
	`id | name | version | package_version`
	`---|---|---|---`
	`9 | 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 4 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.
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 (2014.2.2-6.1)” 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 and 1 Compute nodes. 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.*

Check ScaleIO cluster state

Test Case ID	*check_scaleio_cluster*
Description	*Verify that the ScaleIO cluster state is Normal.*
Steps	1. From the Fuel master node, SSH into the primary ScaleIO MDM and run the following command. `scli --query_cluster`
Expected Result	*The output should show that the cluster state is Normal.*

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*

ScaleIO host failover

Test Case ID	*scaleio_host_failover*
Description	*Remove a ScaleIO MDM node and verify that ScaleIO is still operative.*

Steps	1. Identify a Controller node being used as ScaleIO MDM. You can use the following command to assert that is running the MDM component. If it outputs something, it is running the MDM.
	
	`pgrep mdm`
	
	2. Shutdown the host
	3. From the primary or secondary MDM node, run the following command. It should show that the cluster state is Degraded.
	
	`scli --query_cluster`
	
	4. Create volumes and attach them to running instances.
	5. Boot up the node.

Expected Result	*Even when the cluster is degraded because of a missing MDM node, ScaleIO can operate normally. Once the node is back online, the cluster will rebuild and its state will be back to Normal.*

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 fuel-plugin-scaleio==1.0.0
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 fuel-plugin-scaleio==1.0.0 fuel plugins id | name | version | package_version ---|----------|---------|---------
Expected Result	*The plugin is removed from Fuel.*

Appendix

№	Resource title
1	ScaleIO Fuel Plugin GitHub repository
2	ScaleIO-Cinder Fuel Plugin GitHub repository
3	ScaleIO User Guide
4	ScaleIO OpenStack Information