

GNAST DESIGN and CONFIGURATION GUIDE

For

EMC ScaleIO R1.31

CSBE Storage and Backup Engineering

DRAFT

Template Version: 2.0.7

Document Version#: 0.5

June 17th, 2015



Disclaimer

If printed, content is valid only for the day it was printed. Always refer to the online document for the current version. If the user elects to maintain copies of this procedure, it is their responsibility to verify the currency of the document by checking it against the online version. This document must be promptly removed from use when obsolete.

GNAST-[ProductName]-#### [Design/Standard] Document

FOR INTERNAL USE ONLY

Table of Contents

1	DOCUMENT PURPOSE	5
2	OWNER OF STANDARD	5
3	INTENDED AUDIENCE	6
4	PRINCIPLES.....	6
4.1	Revision History	6
4.2	Key Feature Changes.....	6
5	ASSEMBLY/SOLUTION DESCRIPTION	6
6	SCOPE	7
6.1	Technology.....	7
6.2	Applicability	7
7	COMPONENTS	8
7.1	List of Components	8
7.2	Operational Tools.....	8
8	ASSEMBLY/SOLUTION ARCHITECTURE	9
8.1	ScaleIO Architecture	9
8.1.1	ScaleIO Product Components.....	9
8.1.2	Additional Architecture Features.....	11
8.1.3	VMWare/ESXi implementation	12
8.1.4	ScaleIO Diagram	13
9	CLUSTER INSTALLATION & CONFIGURATION INSTRUCTIONS	14
9.1	Upgrade/Configuration Checklist.....	14
9.2	New Installation/Configuration Checklist.....	15
9.3	Planning Steps Checklist	15
9.3.1	ScaleIO Hardware	15
9.3.2	ScaleIO Server Build	16
9.3.3	ScaleIO Network	16

9.3.3.1	MDM/SDS/SDC Node Network Configuration.....	17
9.3.4	Determining Fault Sets Layout.....	17
9.3.5	Spare Capacity.....	17
9.3.6	RAM Read Cache.....	18
9.3.6.1	Disabling RAM Read Cache on SSD Pools.....	19
9.3.6.2	RAM Read Cache Write Handling Mode	19
9.3.7	Determine the Number of Protection Domains	19
9.3.8	Determine Number of Storage Pools.....	20
9.3.8.1	Pool Naming Convention	20
9.3.9	Cluster Planning Sheet.....	21
9.4	ScaleIO Cluster Installation	22
9.4.1	ScaleIO Software Repository	23
9.4.2	MDM Node Installation.....	23
9.4.3	Tie-Breaker Node Installation	24
9.4.4	SDS Node Installation	25
9.4.5	ScaleIO REST Gateway Installation	26
9.4.6	ScaleIO GUI/Dashboard Installation	28
9.5	ScaleIO Cluster Configuration	28
9.5.1	Cluster Configuration	28
9.5.2	Install ScaleIO License.....	35
9.5.3	REST Gateway Configuration.....	35
9.5.4	Tivoli Event Notification (SNMP/NetCool).....	36
9.5.5	Log Reconciliation (SEMs/ArcSight).....	37
9.5.6	Automated SBCC (HPNA).....	38
9.5.7	Chargeback and Performance Reporting (EMC SRM)	39
9.5.7.1	Chargeback	39
9.5.7.2	Performance reporting setup	40
9.6	ScaleIO Client (SDC) Installation/Configuration.....	42
9.6.1	Linux SDC Installation.....	42
9.6.1.1	Allowing Linux SDC Access to Multiple Clusters	43
9.6.2	Windows SDC Installation	44
9.6.2.1	Allowing Windows SDC Access to Multiple Clusters.....	45
9.6.3	VMWare SDC Installation	45
9.6.4	VMWare Plug-in Installation	48
9.6.5	Provision LUNs for Physical Environments (Linux/Windows)	50
9.6.6	Provision LUNs to Virtualized Environments (ESXi Hosts).....	53
9.6.7	Pool selection.....	54
9.6.8	Pool capacity planning.....	54
9.6.8.1	Capacity utilization	55
9.6.8.2	IOPS utilization.....	57
9.7	Backups.....	59
10	CONFIGURATION ASSURANCE.....	59
11	SCALABILITY & CAPACITY	59
12	AVAILABILITY MANAGEMENT.....	59

13	REGIONAL OR MERGER PARTNER CONSIDERATION	59
14	GLOBAL STANDARD (SECURITY).....	59
14.1	Authentication and Authorization.....	60
14.1.1	Administrative Access	60
14.1.2	Non-administrative Access	60
14.1.3	CLI authentication	60
14.2	Configuration	61
14.2.1	Required Initial Permission Changes.....	61
14.3	ScaleIO Accounts	61
15	RELEVANT LINKS	63
16	GLOSSARY/DEFINITION	63
17	APPENDIXES	64
17.1	Appendix A - Operational Procedures	64
17.1.1	License Management	64
17.1.1.1	Obtaining a permanent ScaleIO license	64
17.1.1.2	Increasing license capacity for existing clusters	66
17.1.1.3	Tracking ScaleIO license usage across Citigroup	66
17.1.2	Troubleshooting processes.....	70
17.1.2.1	Host connectivity to its volumes	70
17.1.2.2	Associating ScaleIO volume names with Linux device names	73
17.1.2.3	Associating ScaleIO volume names with ESXi device names.....	74
17.1.2.4	Logging ScaleIO tickets with EMC	75
17.1.2.5	Log file collection	75
17.1.3	Miscellaneous Cluster maintenance.....	76
17.1.3.1	Disk controller cache – HP	76
17.1.3.2	Identifying Disks in ScaleIO	77
17.1.3.3	Handling a disk failure in SDS node - HP	77
17.1.3.4	Replacing a Failed Disk Controller in SDS node - HP	79
17.1.3.5	Replacing a Failed SDS	81
17.1.3.6	Adding Storage Capacity to a Cluster	81
17.1.3.7	Adding Nodes to a Cluster	82
17.1.3.8	Shutting down a cluster for a power down.....	82
17.1.3.9	Temporarily removing an SDS from a cluster for maintenance.....	83
17.1.3.10	Accidental package deletion.....	85
17.1.3.11	Rebooting SDS servers.....	85
17.1.4	ScaleIO canned performance reports available with SRM	86
17.1.4.1	Pool and volume historical performance graphs	87
17.1.4.2	Other useful canned reports	89
17.2	Appendix B – Best Practices.....	91
17.2.1	MDM Nodes	91
17.2.2	Configuring automated MDM/TB replacement (Standby MDM)	91
17.2.2.1	Overview.....	91

17.2.2.2	Citi Best Practice.....	92
17.2.2.3	Setting up automatic MDM replacement (only).....	93
17.2.2.4	Setting up automatic MDM & TB replacement on Tornado++	94
17.2.2.5	Changing the retry_timeout and and retry_count value	95
17.2.2.6	Resetting MDMs after automatic replacement has taken place	96
17.2.2.7	Troubleshooting Standby MDM/TB	97
17.2.2.8	Disabling automated MDM/TB replacement.....	97
17.2.3	Tie-Breaker Nodes.....	98
17.2.4	SDS Nodes.....	98
17.2.4.1	Throttling SDS rebuild/rebalancing network bandwidth in R1.31	98
17.2.5	Tiered Storage Pools.....	99
17.2.6	Thin Provisioning/ Over Commitment	99
17.2.7	LUN Sizing.....	99
17.2.8	Obtaining and Installing Citi trusted certificates	100
17.2.8.1	Step 1 - Locate keytool application.....	100
17.2.8.2	Step 2 - Create keystore and RSA private key for host	100
17.2.8.3	Step 3 - Create CSR.....	101
17.2.8.4	Step 4 - Request signed certificate for host	102
17.2.8.5	Steps 5 & 6 - Download and import root and intermediate certs.....	105
17.2.8.6	Step 7 - Import signed certificate for host	106
17.2.8.7	Step 8 - Point ScaleIO gateway at new keystore	107
17.2.8.8	Step 9 - Restart ScaleIO gateway	107
17.2.8.9	Step 10 - Test webserver certificate verification	108
17.2.8.10	Step 11 - Copying the keystore to the second REST gateway when VIP in use	108
17.2.9	Virtual IP (VIP) for REST gateways	109

Disclaimer

If printed, content is valid only for the day it was printed. Always refer to the online document for the current version. If the user elects to maintain copies of this procedure, it is their responsibility to verify the currency of the document by checking it against the online version. This document must be promptly removed from use when obsolete.

1 Document Purpose

This document provides detailed installation and configuration instructions for use by all operational and regional engineering teams for [EMC ScaleIO R1.31.1277](#) deployment within Citi's [global](#) environment, to ensure adherence to configuration, support, and management standards as well as compliance to Citigroup Information Security Standards.

2 Owner of Standard

GNAST (Global Network Architecture and Standards Team) is the owner of this standard and the related documents. CATE (Citi Architecture & Technology Engineering) is the custodial organization. CitiNetwork, CitiSystem, & CitiStorage Platform Engineering technologists will drive and generate content.

3 Intended Audience

The intended audience is CitiNetwork, CitiSystems, & CitiStorage Engineering, Global and Regional Operations and Engineering personnel responsible for designing, building, supporting, and managing Citi corporate network environments.

4 Principles

4.1 Revision History

Version	Description	Date	Author/Organization
0.1	Initial EMC ScaleIO standard	1st Feb 2015	Scott Torgerson, Chris Howarth, Chris Clayton /CATE
0.5	Release For Testing	17 th June 2015	Scott Torgerson, Chris Howarth, Chris Clayton, Eliot Wilson

4.2 Key Feature Changes

This is the initial release of the EMC ScaleIO product.

5 Assembly/Solution Description

EMC ScaleIO is software that creates a SAN by aggregating servers local direct-attached storage (DAS) to form globally shared block storage accessible over IP. The lightweight ScaleIO software components are installed on the application hosts and inter-communicate via a standard LAN to handle the application I/O requests sent to ScaleIO block volumes. This decentralized block I/O flow combined with a distributed, sliced volume layout results in parallelism which allows it to scale too many nodes (servers).

A ScaleIO volume is a block device that is exposed to one or more hosts (much like a traditional SAN LUN). ScaleIO breaks each volume into a large number of 1MB data chunks which are spread across the servers and disks within the ScaleIO cluster. Each volume also has a corresponding mirror on different nodes and servers in the cluster. This layout goes a long way in eliminating hot spots across the cluster and allows for the scaling of the overall I/O performance of the system through the addition of nodes and or disks. Furthermore, this layout

enables a single application that is accessing a single volume to use the full IOPs of all the cluster's disks (much like XIV storage arrays).

Traditional storage systems typically combine system software with commodity hardware (which is comparable to application server hardware) to provide enterprise-grade resilience. ScaleIO provides similar enterprise-grade resilience by running the storage software directly on application servers. Designed for fault tolerance and high availability, ScaleIO handles all types of failures including failures of media, connectivity, nodes, software interruptions, and more.

The ScaleIO software features distributed auto-healing processes that can overcome media and node failures without requiring administrator involvement. Dynamic and elastic, ScaleIO enables administrators to add or remove nodes and capacity on the fly. The software immediately responds to changes by rebalancing the storage distribution to achieve a layout that optimally suits the new configuration. All performance and redistributions on ScaleIO can be monitored and observed on the ScaleIO dashboard which is a simple Java applet.

Because ScaleIO is hardware agnostic, the software works with various types of storage including magnetic disks, solid-state disks, and PCIe flash cards. It can be installed in an existing infrastructure as well as in green field configurations.

Based on the above statements, previous evaluation findings, and other factors, it has been determined that EMC ScaleIO will be certified for production use in Citigroup. The primary use case for ScaleIO will be in Virtualized environments (VMWare) along with support for Standalone/Physical environments.

6 Scope

This solution targets all global customers wanting to use 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 over IP networks instead of Fiber Channel.

6.1 Technology

EMC ScaleIO software based Storage Area Network (SAN).
All network traffic on IP Networks instead of Fiber Channel.
Storage exists on commodity servers disks instead of Cache centric Storage Arrays.

6.2 Applicability

Global datacenters or applications requiring Block Storage over IP networks instead of Fiber Channel.

7 Components

7.1 List of Components

The following tables list the required components and versions for the ScaleIO Cluster

Component Name	Capability Description
Base Components	
ScaleIO Cluster Software	V1_31.1277.3 is the certified version for the ScaleIO Cluster
ScaleIO Dashboard	1_31.1277.3 dashboard is used to both monitor and configure parts of the ScaleIO Cluster
REST Gateway	1_31.1277.3 REST Gateway is used to answer all REST API requests of the ScaleIO Cluster.
ScaleIO MDM	MDM is the Meta Data Manager part of the ScaleIO cluster and should only be installed on Linux RHEL 6.X SOE built standalone servers.
ScaleIO SDS	The SDS nodes are the servers that contribute storage to the ScaleIO cluster and should only be installed on Linux RHEL 6.X SOE built standalone servers.
ScaleIO SDC ¹	SDC software should only be installed on these O.S. types for this certification: RHEL 6.X (SOE Built) Windows 2008R2/2012 (SOE Built) ESXi 5.5 SP1, SP2 (SOE Built)

7.2 Operational Tools

The following tools listed in the table will have each ScaleIO Cluster integrated into their infrastructure to help with monitoring the solution.

Name	Capability Description	Result
Tivoli	Event Notification	ongoing
Hewlett Packard Network Automation	SBCC Automation	ongoing
Arcsight	Log Reconciliation	ongoing
EMC SRM	Chargeback and Performance reporting	ongoing

¹ Currently there is no ScaleIO SDC support for the Solaris and AIX Operating Systems

8 Assembly/Solution Architecture

8.1 ScaleIO Architecture

ScaleIO is scale-out software-only solution running on commodity hardware that leverages servers with local disks communicating with a proprietary protocol over TCP to create a shared pool of **block storage**. It is hardware agnostic, and will work with HDD, SDD as well as flash PCIe cards. ScaleIO does not provide file or object storage.

ScaleIO has been designed with auto-monitoring & auto-healing processes that deal with loss of disks and nodes without requiring administrator intervention. Nodes can also be added or removed on the fly, and the software will immediately respond to the changes rebalancing the storage across the new configuration.

8.1.1 ScaleIO Product Components

ScaleIO has three main components:

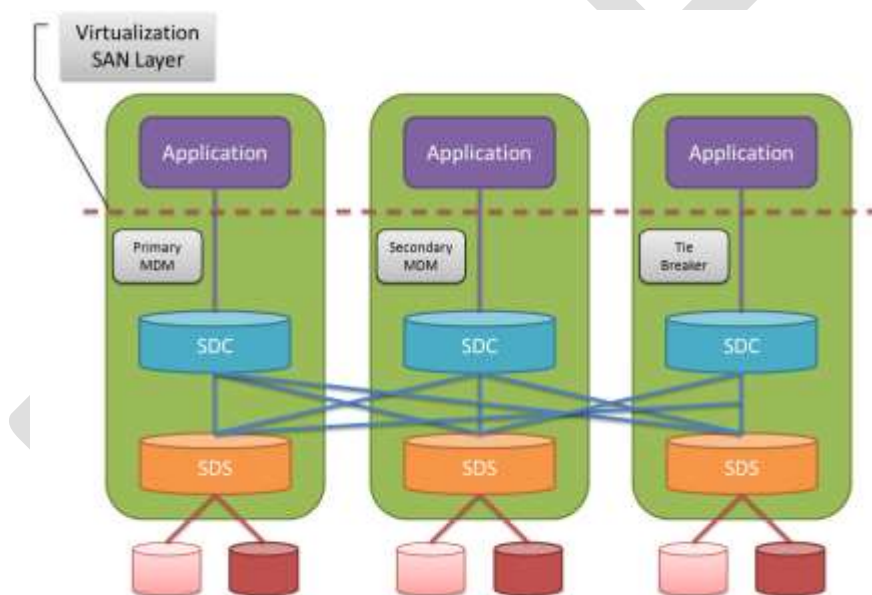


Figure 1 - ScaleIO Architecture and Components

SDC – ScaleIO Data Client – Device driver resident on each host whose applications require access to ScaleIO virtual SAN block devices. An SDC is somewhat similar to an iSCSI initiator. When an SDC gets a read request from an application, it uses lookup information to determine which SDS has the data, and then interacts directly with that SDS over IP networks.

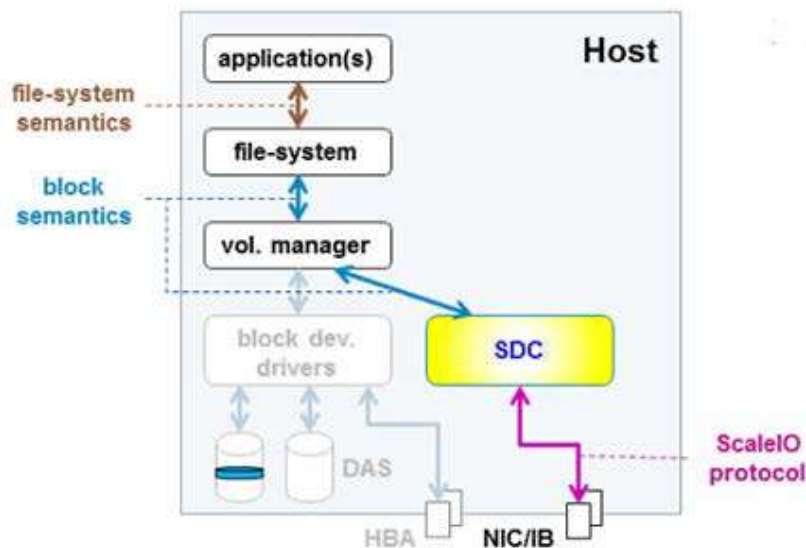


Figure 2 - ScaleIO Data Client (SDC)

SDS – ScaleIO Data Server – Software component resident on each host that contributes local storage to the central ScaleIO virtual SAN. An SDS is somewhat similar to an iSCSI target.

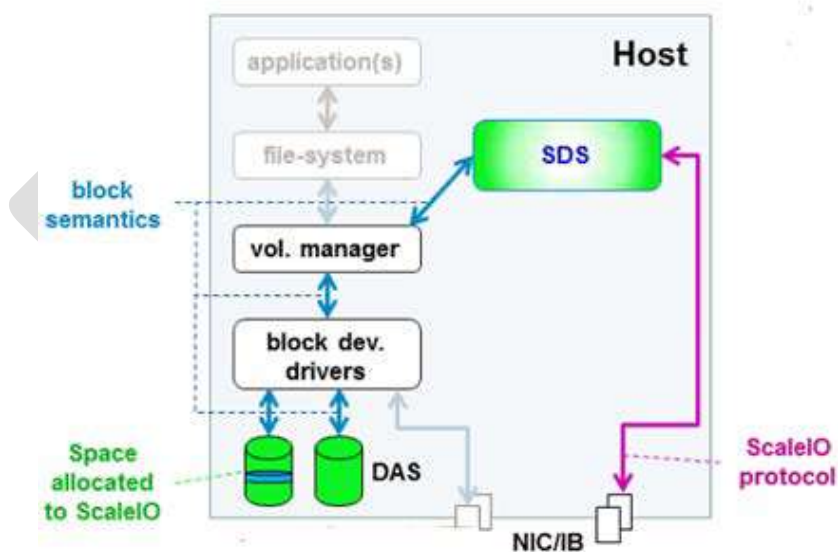


Figure 3 - ScaleIO Data Server (SDS)

An SDC and an SDS can also run in the same host in a “converged” configuration. In this model, application servers would also be acting as providers of a shared pool of storage.

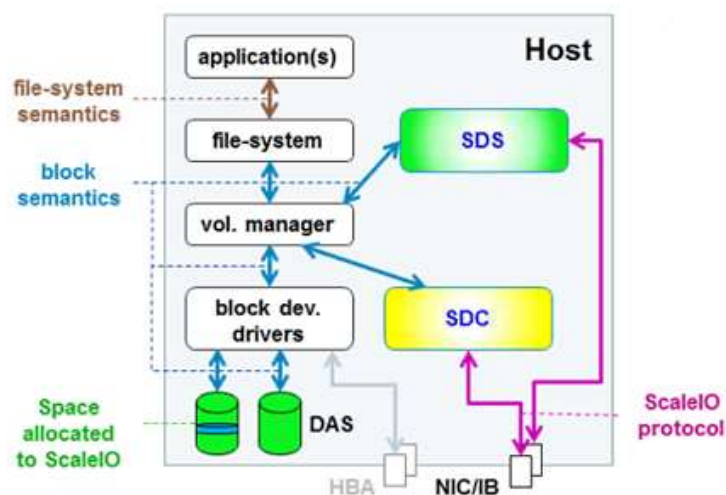


Figure 4 - SDS & SDC on same server (Converged Configuration)

MDM (Metadata manager) – This software component is the authority for re-mapping and data chunk ownership, and is also responsible for data migration decisions (such as rebuilds and rebalancing). User data never passes via the MDM (passes directly from SDSs to SDC) and hence is not a bottleneck. A tie-breaker instance is used to avoid split-brain.

8.1.2 Additional Architecture Features

Each volume presented to hosts is broken into 1MB data chunks (of which there are 2 copies on separate SDS nodes disk i.e. a mesh mirror very similar to XIV) that are scattered across the cluster in a balanced manner, providing parallel access by an application to all nodes and their disks.

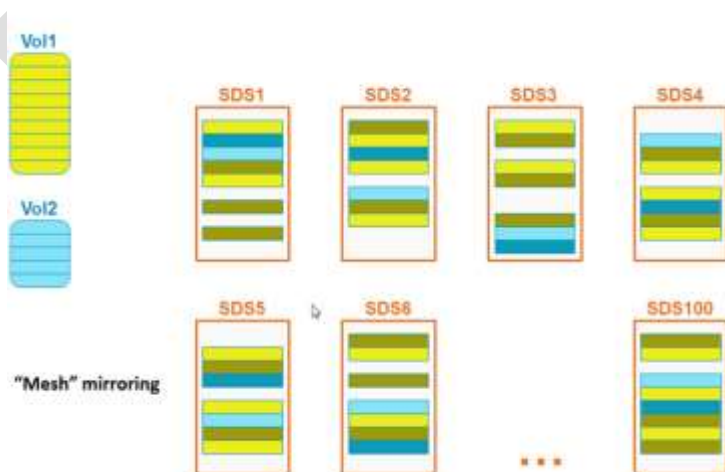


Figure 5 - ScaleIO 2-copy mirror scheme

In the event of a loss, ScaleIO immediately starts to create another mirror for the data chunks that were lost in the failure.

Each SDC knows where to retrieve its data from and hence data flow is de-centralized, which the vendor claims leads to linearly scalable performance and a solution can scale to hundreds or even thousands of nodes. The more nodes the configuration has, the greater the numbers of parallel IO channels are available and the higher the aggregated IO bandwidth and IOPS.

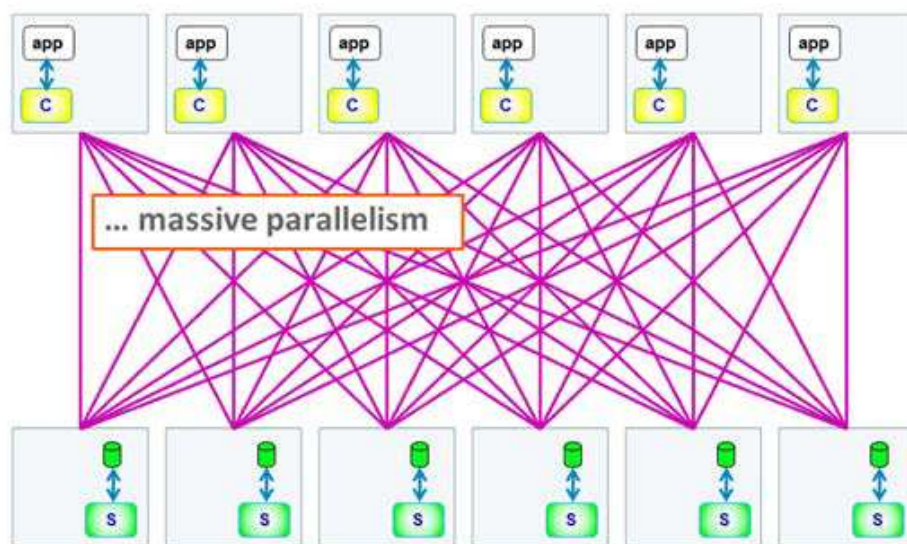


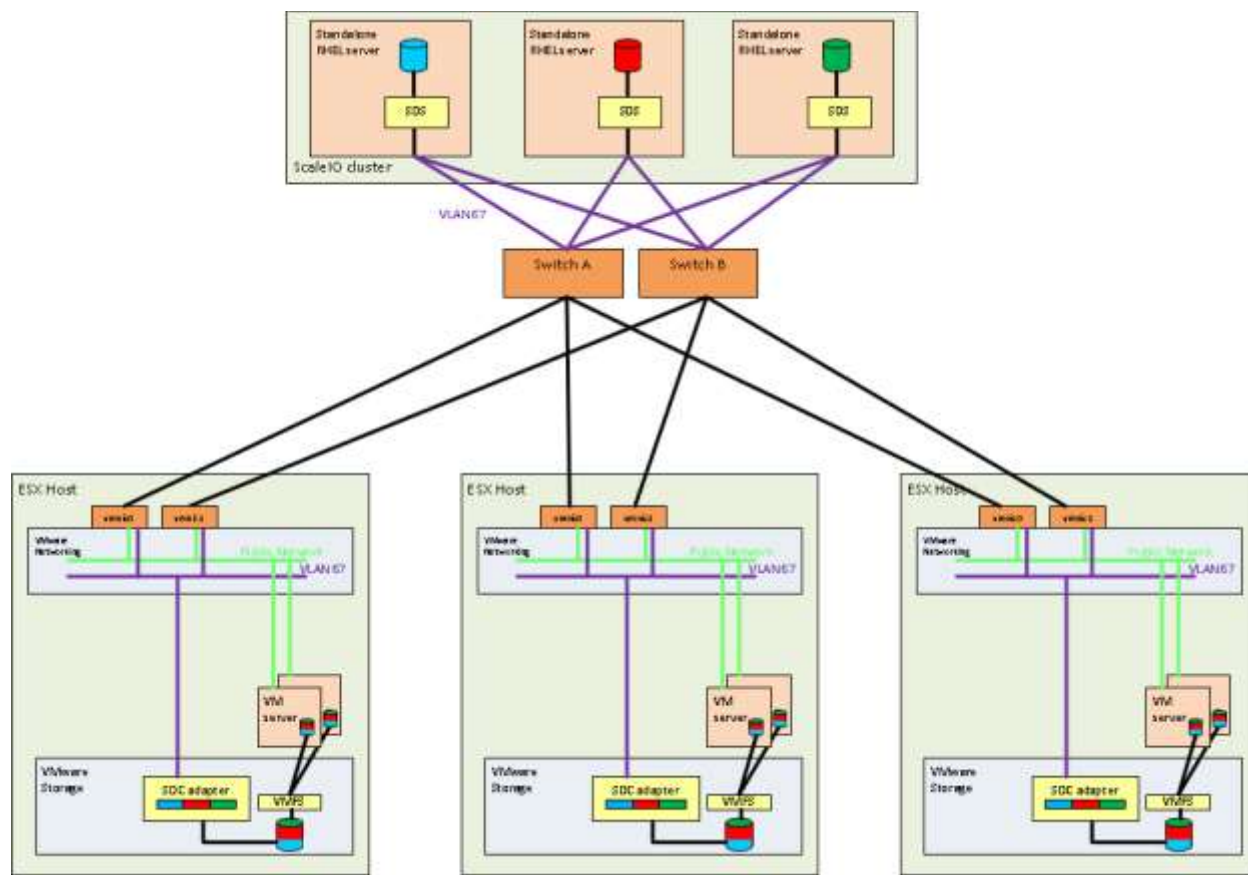
Figure 6 - Direct parallel access for high throughput low latency

ScaleIO has the facility for administrators to set maximum IOPS or bandwidth values per client or datastore.

ScaleIO also supports redirect-on-write (ROW) snapshots of volumes, and these snapshots can be consistent across multiple volumes and servers. This is currently no support for remote replication within ScaleIO.

8.1.3 VMWare/ESXi implementation

ScaleIO is able to provision LUN devices directly to the ESXi Hypervisor. The diagram below provides an architectural overview of this.

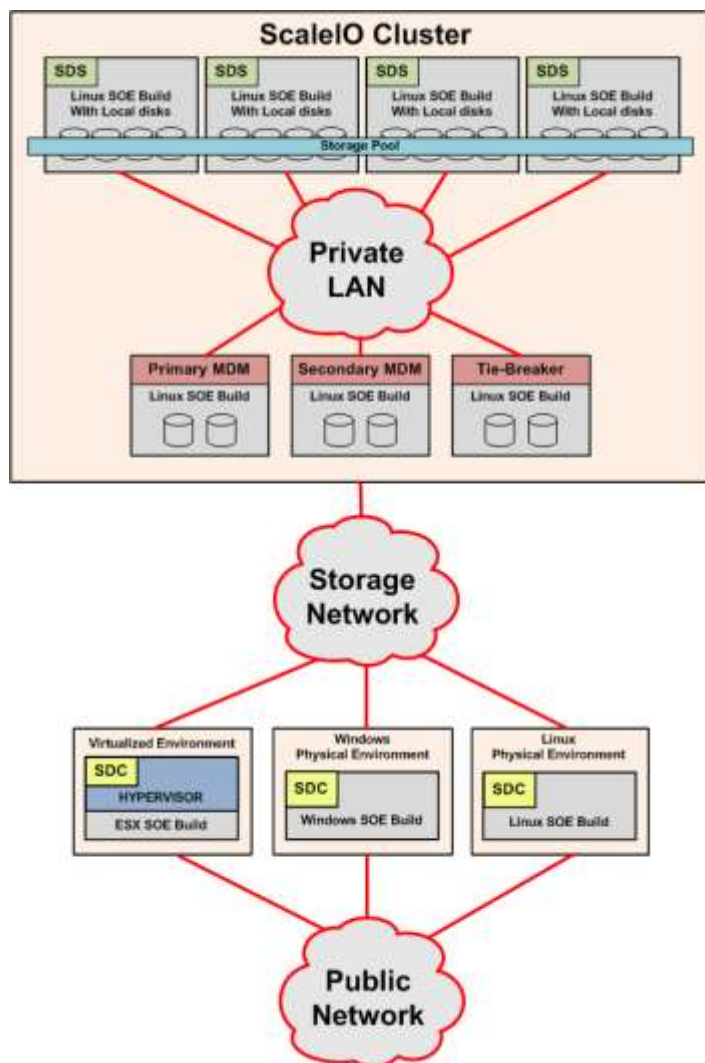


The top diagram shows a 3 node ScaleIO cluster. This cluster aggregates the storage of the 3 servers to create LUN devices. Each ESXi host has the ScaleIO SDC client driver installed on it. The ScaleIO cluster maps LUN devices to all the ESXi hosts. As shown above each LUN is composed of chunks of data from all 3 physical ScaleIO hosts. The LUNs are then formatted as VMFS devices and a Datastore is created. At this point administration of the storage is performed by VMware.

Note that the same Datastore is visible on all ESXi hosts and none of this storage is physically local. Consequently storage VMotion operations are quick.

8.1.4 ScaleIO Diagram

The following diagram shows a typical ScaleIO deployment. All servers that make up the ScaleIO Cluster (MDM, Tie-Breaker, and all SDS's) would be built out on Citigroup SOE Built Linux servers. On the cluster is built and configured, all disks on the SDS nodes would be grouped together to form a storage pool. After the pool is created supported O.S. types (virtualized or physical) on the LAN could then load the ScaleIO SDC and have LUNs provisioned from the cluster.



9 Cluster Installation & Configuration Instructions

The following sections of this document discuss the installation and configuration steps of the ScaleIO cluster. The section “New Installation and Upgrade Checklist” can be followed to ensure all components of the ScaleIO cluster are properly installed and configured. Each step is hyperlinked to a more detailed section describing the step.

9.1 Upgrade/Configuration Checklist

Since this is the first release of EMC ScaleIO, no upgrade checklist will be given. For all future releases of ScaleIO, a checklist will be given on how to upgrade an existing cluster from the previous version of ScaleIO.

Note that EMC do not support the ability to downgrade ScaleIO. Hence if an issue is encountered following an upgrade, EMC will provide a patch to this upgrade rather than the ability to downgrade.

9.2 New Installation/Configuration Checklist

The following is a list of steps that should be used when creating a new ScaleIO Cluster.

A Linux repository has been created and will be maintained by CATE that contains all the necessary software packages required for the ScaleIO installation.

- ✓ [Complete Planning Steps Checklist](#)
- ✓ [Install MDM Nodes](#)
- ✓ [Install Tie-Breaker Node](#)
- ✓ [Install All SDS Nodes](#)
- ✓ [Install REST Gateway](#)
- ✓ [Install ScaleIO GUI/Dashboard](#)
- ✓ [Configure ScaleIO Cluster](#)
- ✓ [Install ScaleIO License](#)
- ✓ Install [Linux](#), [Windows](#), or [VMWare](#), SDC Nodes
- ✓ [Install/Configure VMWare Plug-In \(VMWare environments Only\)](#)
- ✓ [Provision LUNs for Physical Environments \(Linux/Windows\)](#)
- ✓ [Provision LUNs for Virtualized Environments \(ESX Hosts\)](#)

9.3 Planning Steps Checklist

The following sections will detail information that should be gathered and determined culminating in a planning sheet that will be used for the installation on configuration of the ScaleIO Cluster. All items on this checklist should be completed prior to installation and configuration of a ScaleIO cluster

- ✓ [Validate ScaleIO Cluster Hardware](#)
- ✓ [Validate ScaleIO Cluster Networking](#)
- ✓ [Determine Number of Fault-Sets](#)
- ✓ [Determine Spare Capacity value](#)
- ✓ [Determine RAM Read Cache Value](#)
- ✓ [Determine Number of Protection Domains](#)
- ✓ [Determine the number of Storage Pools](#)
- ✓ [Put All Above Values in Planning Sheet](#)

9.3.1 ScaleIO Hardware

For the initial release, ScaleIO will be certified to be run on the Citi Tornado++ pod hardware configuration. A separate guide will be created that described in detail specific best practice for deployment of ScaleIO on Tornado++ hardware, and associated network configuration. The present document provides generic instructions on how to deploy ScaleIO on any hardware platform, and is aimed initially at POC uses cases. For POCs, other hardware stacks (e.g. Tornado) can also be used with agreement with the authors of this document.

Each node should be RHEL 6.4 (or higher) SOE built. ScaleIO on RHEL7 is not supported in Citi.

Documentation of the current Tornado++ hardware can be found here:

<https://catecollaboration.citigroup.net/domains/platstor/gdse/products/platforms/EMC%20ScaleIO/CSBE-15004-BP.pdf>

But in brief, a Tornado++ storage pod consists of:

- 4 racks of storage nodes, with each rack contains 12 SAS/SSD-based server and 4 NL-SAS servers
- ScaleIO will be deployed on the SAS/SSD servers only, but primarily utilizing the SAS disks. Separate NL-SAS servers will be used for both ScaleIO and other storage applications (e.g. an object store).
- The SAS/SSD servers contain 20*600GB SAS drives and 3*800GB SSD drives.
- Hence, the four storage racks of a Tornado++ pod will contain 960*600GB 10K SAS drives, and deliver roughly 100,000 host IOP, with an IOPS density of around 450IOPS/usable TB (assuming a 50%:50% R:W host workload with 40% read cache hits).

The approved RHEL versions for this release of ScaleIO to install on the above servers are listed in the [List of Components](#) table of this document.

9.3.2 ScaleIO Server Build

The servers used for ScaleIO should have an SOE RHEL6 build on them. Following build there are some ScaleIO specific configuration settings required:

- When configuring access, Host Access control should be enabled with login access permitted for the server Unix SA team and the storage admin team only.
- Before installing and configuring the ScaleIO software, confirm with the SA supporting the servers that hardware monitoring has been configured and that it has been confirmed that traps are successfully being received in Netcool.

9.3.3 ScaleIO Network

The ScaleIO network design has been designed to make best use of the available network bandwidth, whilst providing no single point of failure.

Prior to installing the cluster each server should have four networks configured: 2 public or client facing networks over which the SDCs connect to the SDS servers and for external access to the MDM servers and the gateway. Secondly 2 private networks should be configured for the cluster traffic to use.

Jumbo frames should be enabled on all interfaces

9.3.3.1 MDM/SDS/SDC Node Network Configuration

Detailed configuration of the network is provided in the T++ Configuration Document as this is specific to the particular ScaleIO cluster

9.3.4 Determining Fault Sets Layout

A Fault Set is a logical entity that contains a group of SDSs within a Protection Domain that have a higher chance of failing simultaneously. For example if they are all located within a single rack. By grouping them into a Fault Set, you are telling ScaleIO that the data replicas for all devices in this Fault Set, should be located on SDSs that are outside of this Fault Set.

Most ScaleIO clusters will be comprised of racks containing many servers in every rack. When installing the racks, fault sets should be taken into consideration. For Citi ScaleIO POC clusters, the following best practices for Fault Sets should be followed.

- A minimum of 4 Fault Sets should be used in any ScaleIO cluster.
- The number of Fault Sets in a ScaleIO cluster with multiple racks should be evenly divisible by the number of racks containing SDS Nodes (Ex: 4 Racks, 4 Fault Sets or 8 Racks, 4 Fault Sets).
- Whenever possible, each SDS Node rack belonging to a Fault Set should be powered by the same PDU's in the datacenter
- Whenever possible all Fault Set racks should be powered to different PDU's in the datacenter to provide maximum redundancy.

For Tornado++ pod hardware configurations, each ScaleIO Cluster will start out with (4) fault sets (1) for each of the (4) racks.

9.3.5 Spare Capacity

To ensure data protection during server failures, ScaleIO reserves 10% of the capacity by default, not allowing it to be used for volume allocation. To ensure full system protection during the event of an SDS node failure, ScaleIO administrators must make sure that the spare capacity is at least equal to the amount of capacity in a single SDS node along with enough capacity to lose a few extra disks. Note that a rebuild can also use "Unused" cluster space (without limit) once the dedicated spare capacity has been consumed i.e. a rebuild is not limited to only using the space marked as "spare" capacity.

If all SDS nodes contain equal capacity, it is recommended to set the capacity value to at least $2/N$ of the total capacity where N is the number of SDS nodes. Spare capacity is set at the pool level and the current value can be found using the following command.

```
# scli --query_storage_pool --protection_domain_name domain1 --storage_pool_name domain1-sp1 | grep Spare
Spare policy: 10% out of total
```

To set the spare capacity of each storage pool, use the following formula and commands.

Note: Since POC environments are relatively small, only a single SDS node of spare capacity should be configured. The below example would be for a production type environment.

```
### EXAMPLE ###

# Cluster has 4 Racks with 12 SDS Nodes in each Rack.
# (2) SDS Nodes for Spare / (48) SDS Nodes = 5%

# scli --modify_spare_policy --protection_domain_name domain1 --storage_pool_name domain-sp1 --
spare_percentage 5
Modifying the spare policy might leave some data unprotected in case of failure. Press 'y' to confirm that you want to
proceed ? y
Successfully modified spare policy.
```

9.3.6 RAM Read Cache

By default, the RAM read cache is enabled with a very small cache size on all SDS nodes, newly created storage pools, and newly created LUNs. However, when adding the SDS nodes to the ScaleIO cluster the RAM Read cache size for all SDS nodes should be adjusted during the ScaleIO cluster configuration. Prior to installation and configuration of the ScaleIO Cluster, the RAM read cache number should be determined and entered on the planning sheet (see section 9.3.8) for each SDS node.

By default, the amount of RAM cache that can be configured is limited by the server hardware configuration. Hence, regardless of what you set the READ RAM cache size to be, the amount of cache actually used by default will be limited to 25% on Dual-socket CPUs and 50% on Single-socket CPUs. To allow SDS nodes to actually use a higher cache setting, the following entries must be placed in the /opt/emc/scaleio/sds/cfg/conf.txt file.

Note: It would be best to make the edits to this file when the SDS rpm package is initially installed on the SDS.

```
rcache_allowed_high_memory_percentage=80
rcache_allowed_low_memory_percentage=80
rcache_allowed_high_memory_precentage=80
rcache_allowed_low_memory_precentage=80
numa_affinity=1000
numa_memory_affinity=0
```

If the above edits were not made when the SDS rpm package was installed or prior to the cluster being configured, the SDS service would have to be restarted on each SDS node where the configuration file is changed.

```
# /opt/emc/scaleio/sds/bin/delete_service.sh
# /opt/emc/scaleio/sds/bin/create_service.sh
```

Note: If the SDS service needs to be restarted on each SDS node, it should be done serially and any rebuild/rebalance should be allowed to complete prior to restarting services on the next SDS node.

After the above configuration file edits have been completed, the amount of total memory on each SDS Node will need to be determined. If the total amount of memory is *not* known for every SDS node, or if every SDS node in the ScaleIO Cluster does not have the same amount of memory the following command can be used to determine the amount of memory on an SDS node.

```
$ free -g
```

	total	used	free	shared	buffers	cached
Mem:	46	18	28	0	0	0
-/+ buffers/cache:		17	29			
Swap:	15	0	15			

After the amount of total memory is known *for each* SDS, the following formula should be used to determine the RAM Read Cache setting that will be used on each SDS nodes.

(Total Memory on SDS) * 0.65 = RAM Read Cache setting for this SDS
 46GB * 0.65 = 29.9 (round up to 30)

9.3.6.1 Disabling RAM Read Cache on SSD Pools

Employing RAM cache is not beneficial with SSD devices. RAM cache should only be enabled for HDDs pools. Hence, the RAM cache should be enabled for the SDS nodes, but then disabled for SSD-based storage pools. The following command should be run on any SSD pool.

```
# scli --set_rmcache_usage --protection_domain_name <domain> --storage_pool_name <name-of-flash-pool> --dnt_use_rmcache
```

9.3.6.2 RAM Read Cache Write Handling Mode

The `rmcache_write_handling_mode` parameter determines if the writes of the Storage Pools are stored in the SDS RAM read cache or if it to be passed through to the storage. For all ScaleIO environments this parameter should be left at the default value which is *cached*.

9.3.7 Determine the Number of Protection Domains

A Protection Domain is a set of SDSs. Each SDS belongs to one (and only one) Protection Domain. Thus, by definition, each Protection Domain is a unique set of [SDS Nodes](#) with all of their disks. A ScaleIO cluster can have multiple protection domains configured. But it is important to note that when doing this the performance of each protection domain will depend on the number of SDS Nodes in each.

For this initial release the following best practices should be followed for Protections Domains:

- Only one protection domain should be created for each POC ScaleIO Cluster
- Specific best practice will be provided for Tornado++ configurations

9.3.8 Determine Number of Storage Pools

Storage Pools allow the generation of different storage tiers in the ScaleIO system. A Storage Pool is a set of physical storage devices (SDS node disks) in a Protection Domain. Each storage device belongs to one (and only one) Storage Pool. When a Protection Domain is generated, by default it has one Storage Pool.

When a ScaleIO volume/LUN is created, the volume is distributed over all devices residing in the same Storage Pool. This allows more than one failure in the system without losing data. To provide consistent performance it is recommended that all devices in the Storage Pool will have similar storage properties (See [SDS Node best practice](#)). Mixing different disk types in SDS nodes that make up the storage pool would limit performance of that pool to the least performing disks.

For this initial ScaleIO release the following best practices for Storage pools for POC environments should be followed. **Best practice for Tornado++ environments is in preparation but will differ to that below, as there will be multiple HDD (SAS) pools and also an SSD pool.**

- Only one storage pool should be created for each drive type each ScaleIO cluster (See [Tiered Storage Pools best practice](#)).
- Follow [SDS Node best practice](#) of keeping all disk types the same in all SDS nodes in the same protection domain and storage pool.

9.3.8.1 Pool Naming Convention

Since there is no current pool type indicator in the REST Gateway output, the only way to determine the pool type is by the pool name. Pool types are required information so that HPNA can check the parameters that the different pool types require and for orchestrated provisioning. As a result, the following naming convention should be used for all storage pools.

[protection domain name]-sp[type][n]

SAS Pool example:	domain1-spsas1, domain1-spsas2, domain1-spsas3
SSD Pool example:	domain1-spssd1
NLSAS Pool example:	domain1-spnls1

Valid types are sas, ssd and nls for SAS, SSD and NL-SAS devices respectively.

9.3.9 Cluster Planning Sheet

Prior the beginning the ScaleIO Cluster installation and configuration, a planning sheet should be completed with all the necessary cluster components and the information required for each. Below is a sample Tornado++ planning sheet. The planning sheet and all the values should be used when doing the installation/configuration.

Component Role	Host Name	Rack # / Fault Set #*	Public IP	Private/ Cluster IP	List/range of SDS HDD devices	List/range of SDS SDD devices	Read RAM Cache Size (MB)
Primary MDM SDS #1 REST Gateway							
Secondary MDM SDS #2 REST Gateway							
Tie Breaker/ SDS #3							
Standby MDM** SDS #4							
Standby TB*** SDS #5							
Standby MDM** SDS #6							
Protection domain name							
HDD Storage pool name							
SDD Storage pool name							

* Each rack should be configured as a separate fault set

** The standby MDM is described in detail in the [Standby MDM](#) section, but in brief is an MDM process that can become a secondary MDM in the event of the extended loss of the either the current primary or secondary MDM, to ensure that the cluster does not run with just a single MDM for an extended period.

*** The standby TB is described in detail in the [Standby MDM](#) section, but is only applicable in environments with at least 5 nodes.

9.4 ScaleIO Cluster Installation

In the following example a small 4 node ScaleIO cluster will be installed and configured. The layout is shown in Figure 7:

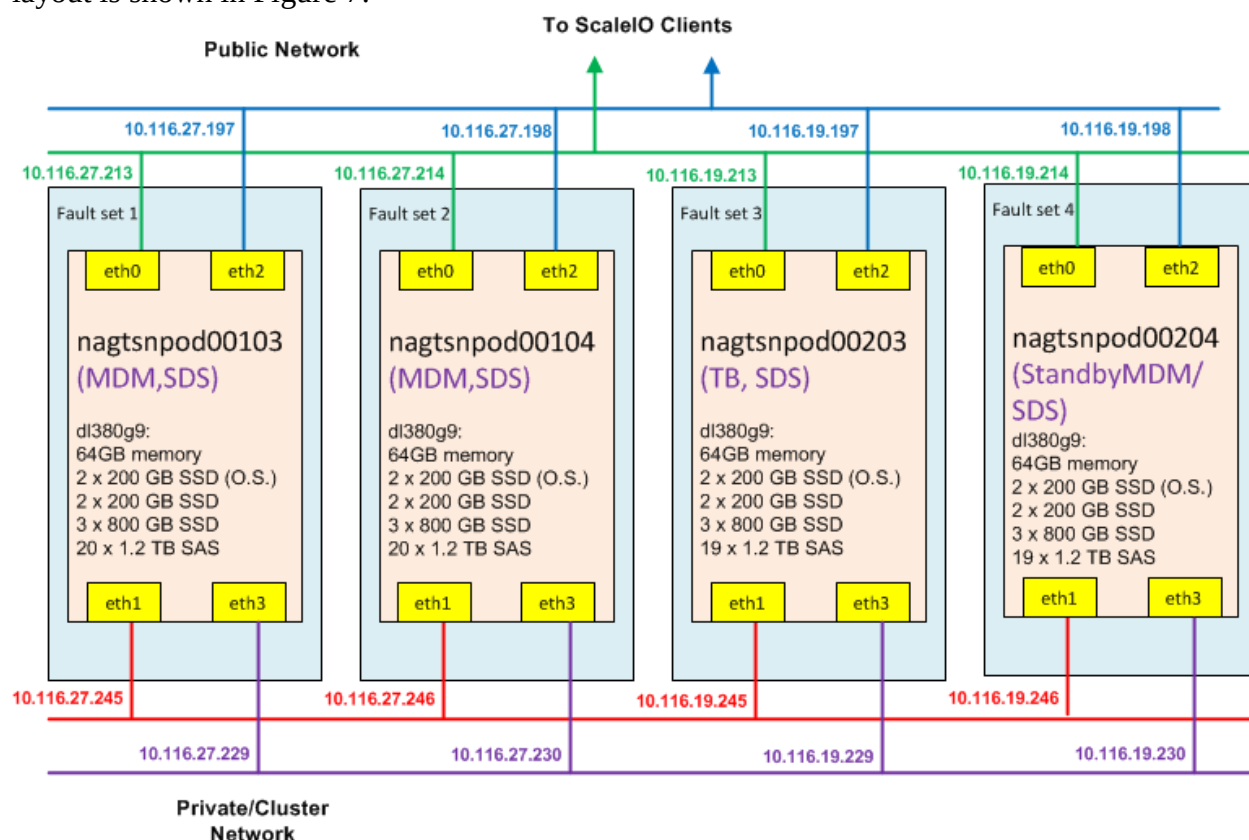


Figure 7 - An example of a small ScaleIO cluster

Each node has two public interfaces connecting to the public network to which ScaleIO clients connect. There are two separate private networks for the cluster traffic to use for communication. The planning sheet for this configuration is shown below.

Component Role	Host Name	Rack # / Fault Set #*	Public IP	Private/ Cluster IP	List/range of SDS HDD devices	List/range of SDS SDD devices	Read RAM Cache Size (MB)
Primary MDM SDS #1 REST Gateway	nagtsnpod00103	domain1-fs1	10.116.27.213 10.116.27.197	10.116.27.245 10.116.27.229	/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg,/dev/sdh,/dev/sdi,/dev/sdj, /dev/sdk,/dev/sdl, /dev/sdm,/dev/sdn, /dev/sdo,/dev/sdp,/dev/sdq, /dev/sdr,/dev/sds, /dev/sdt,/dev/sdu	/dev/sdx,/dev/sdy, /dev/sdz	16384MB
Secondary MDM SDS #2 REST Gateway	nagtsnpod00104	domain1-fs2	10.116.27.214 10.116.27.198	10.116.27.246 10.116.27.230	/dev/sdb,/dev/sdc,/dev/sdd, /dev/sde,/dev/sdf, /dev/sdg,/dev/sdh, /dev/sdi,/dev/sdj,	/dev/sdx,/dev/sdy, /dev/sdz	16384MB

					/dev/sdk,/dev/sdl,/dev/sdm,/dev/sdn,/dev/sdo,/dev/sdp,/dev/sdq,/dev/sdr,/dev/sds,/dev/sdt,/dev/sdu		
Tie Breaker/ SDS #3	nagtsnpod00203	domain1-fs3	10.116.19.213 10.116.19.197	10.116.19.245 10.116.19.229	/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg,/dev/sdh,/dev/sdi,/dev/sdj,/dev/sdk,/dev/sdl,/dev/sdm,/dev/sdn,/dev/sdo,/dev/sdp,/dev/sdq,/dev/sdr,/dev/sds,/dev/sdt,/dev/sdt	/dev/sdw,/dev/sdx,/dev/sdy	16384MB
Standby MDM** SDS #4	nagtsnpod00204	domain1-fs4	10.116.19.214 10.116.19.230	116.19.246 10.116.19.230	/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg,/dev/sdh,/dev/sdi,/dev/sdj,/dev/sdk,/dev/sdl,/dev/sdm,/dev/sdn,/dev/sdo,/dev/sdp,/dev/sdq,/dev/sdr,/dev/sds,/dev/sdt,/dev/sdt	/dev/sdw,/dev/sdx,/dev/sdy	16384MB
Protection domain name	domain1						
HDD Storage pool name	domain1-spsas						
SDD Storage pool name	domain1-spssd						

9.4.1 ScaleIO Software Repository

The ScaleIO rpms are located in one of the SOE yum repos. To access, create the following entry on each ScaleIO node:

```
# cat >/etc/yum.repos.d/soe-fssprod.repo <<EOF
[soe6fssproducts]
name=SOE File System and Storage Products
baseurl=http://rh6lab7-64.nam.nsroot.net/SOELinux/repos/testing/soe6fssproducts-$basearch
enabled=1
gpgcheck=0
EOF
```

9.4.2 MDM Node Installation

For the ScaleIO Cluster there should be two [MDM Nodes](#) installed, which will act as primary and secondary. CATE also recommends configuring a “standby MDM” that can automatically be configured into the cluster to replace either a primary or secondary MDM that has failed for an extended period. This can be configured later (see Best Practice on [Configuring automated MDM replacement \(Standby MDM\)](#)).

The following command should be used to install the MDM software on the primary and secondary MDM nodes. In our example we are installing on nodes nagtsnpod00103, nagtsnpod00104

```
# yum -y install EMC-ScaleIO-mdm.x86_64
soe6fssproducts | 2.5 kB 00:00
```

```

Setting up Install Process
Resolving Dependencies
--> Running transaction check
--> Package EMC-ScalerIO-mdm.x86_64 0:1.31-1277.3.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

.....

Total download size: 4.1 M
Installed size: 126 M
Downloading Packages:
EMC-ScalerIO-mdm-1.31-1277.3.el6.x86_64.rpm      | 4.1 MB  00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : EMC-ScalerIO-mdm-1.31-1277.3.el6.x86_64      1/1
Successfully finished repository formatting operation
mdm start/running, process 51667
  Verifying  : EMC-ScalerIO-mdm-1.31-1277.3.el6.x86_64      1/1

Installed:
EMC-ScalerIO-mdm.x86_64 0:1.31-1277.3.el6

Complete!

```

9.4.3 Tie-Breaker Node Installation

For the ScaleIO Cluster there should be one [Tie-Breaker Node](#) installed. The following command should be used to install the Tie-Breaker software on that node. In our example we are installing on node nagtsnpod00203

```

# yum -y install EMC-ScalerIO-tb.x86_64
soe6fssproducts      | 2.5 kB  00:00
soe6fssproducts/primary_db | 15 kB  00:00
soe6products         | 1.2 kB  00:00
soe6u5               | 1.2 kB  00:00
Setting up Install Process
Resolving Dependencies
--> Running transaction check
--> Package EMC-ScalerIO-tb.x86_64 0:1.31-1277.3.el6 will be installed
--> Finished Dependency Resolution

.....

Total download size: 1.5 M
Installed size: 39 M
Downloading Packages:
EMC-ScalerIO-tb-1.31-1277.3.el6.x86_64.rpm      | 1.5 MB  00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : EMC-ScalerIO-tb-1.31-1277.3.el6.x86_64      1/1
tb start/running, process 39570
  Verifying  : EMC-ScalerIO-tb-1.31-1277.3.el6.x86_64      1/1

Installed:

```

```
EMC-ScaleIO-tb.x86_64 0:1.31-1277.3.el6
```

Complete!

9.4.4 SDS Node Installation

In the ScaleIO cluster, any node that will contribute storage to the cluster should have the SDS software installed on it. To install the SDS software on an [SDS node](#), use the following command. In our example all 4 nodes require this: nagtsnpod00103, nagtsnpod00104, nagtsnpod00203, nagtsnpod00204

```
# yum -y install EMC-ScaleIO-sds.x86_64
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package EMC-ScaleIO-sds.x86_64 0:1.31-1277.3.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

.....

Install      1 Package(s)

Total download size: 1.6 M
Installed size: 74 M
Downloading Packages:
EMC-ScaleIO-sds-1.31-1277.3.el6.x86_64.rpm      | 1.6 MB    00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : EMC-ScaleIO-sds-1.31-1277.3.el6.x86_64      1/1
sds start/running, process 24618
  Verifying  : EMC-ScaleIO-sds-1.31-1277.3.el6.x86_64      1/1

Installed:
  EMC-ScaleIO-sds.x86_64 0:1.31-1277.3.el6

Complete!
```

Once the SDS node software installed, the following [RAM Read Cache configuration file changes should be made](#).

The following entries must be placed in the `/opt/emc/scaleio/sds/cfg/conf.txt` file.

```
rcache_allowed_high_memory_percentage=80
rcache_allowed_low_memory_percentage=80
rcache_allowed_high_memory_precentage=80
rcache_allowed_low_memory_precentage=80
numa_affinity=1000
numa_memory_affinity=0
```

Once the above changes have been made, the newly installed SDS service should be stopped and restarted using the following commands.

```
# /opt/emc/scaleio/sds/bin/delete_service.sh  
# /opt/emc/scaleio/sds/bin/create_service.sh
```

Note: If the above edits and service restart are not completed when the SDS rpm package is installed, or prior to the cluster being configured, the ScaleIO administrator would be forced to wait for any rebalance or rebuild activity to finish prior to restarting the SDS service on the next SDS nodes.

9.4.5 ScaleIO REST Gateway Installation

In order to make ScaleIO accessible via REST API, the ScaleIO REST Gateway must be installed and configured using the following commands. The Gateway should be installed on **both** the primary and secondary [MDM nodes](#) in order to provide two REST access points should one or other be down. The load on the REST gateway will not be high, and hence there is no need to load balance across both gateways. However, the REST API will still need to be accessible if the primary gateway is down. It will be the responsibility of the application that consumes the REST API to be able to try the alternate gateway if its primary gateway is not responsive.

Start by installing a REST gateway on the current primary MDM.

First install the required Java Run Time Environment (JRE). This can be obtained from the [Java Engineering Website](#).

```
# rpm -i /tmp/jre-1.7.0_65l64-linux.x86_64.rpm  
preinstall  
RPM_INSTALL_PREFIX=/opt/jre  
RPM_INSTALL_PREFIX0=/opt/jre  
postinstall  
RPM_INSTALL_PREFIX=/opt/jre  
RPM_INSTALL_PREFIX0=/opt/jre  
  
***  
*** HOME_OPT=/opt/jre/1.7.0_65l64  
***  
  
Done!
```

Due to a bug in ScaleIO R1.31 GA code it is currently necessary to create a link to the java binary if it does not already exist, else the execution of the post-install script during the installation of the REST gateway will fail to run correctly.

```
# ln -s /opt/jre/1.7.0_65l64/bin/keytool /usr/bin/keytool  
# ln -s /opt/jre/1.7.0_65l64/bin/java /usr/bin/java
```

Install the ScaleIO REST gateway package, defining the location of the Citi java installation with the environment variable `SIO_GW_JAVA`. Do NOT specify a gateway admin password on the installation command line. The gateway should be configured without a password being set to disable access to it.

```
# SIO_GW_JAVA=/opt/jre/1.7.0_65/64 SIO_GW_KEYTOOL=/opt/jre/1.7.0_65/64/bin
yum -y install EMC-ScaleIO-gateway-1.31
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package EMC-ScaleIO-gateway.noarch 0:1.31-1277.3 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
====
Package                Arch      Version      Repository
Size
=====
====
Installing:
EMC-ScaleIO-gateway    noarch    1.31-1277.3  soe6fssproducts
63 M

Transaction Summary
=====
====
Install                1 Package(s)

Total download size: 63 M
Installed size: 75 M
Downloading Packages:
EMC-ScaleIO-gateway-1.31-1277.3.noarch.rpm | 63 MB    00:06
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : EMC-ScaleIO-gateway-1.31-1277.3.noarch
1/1
  /usr/bin/java
found the Java executable in PATH
version 17
The GATEWAY_ADMIN_PASSWORD variable was not set.
This is not a mandatory parameter during installation. You can configure the Installation Manager
admin user's password later, using REST.
  /usr/bin/java
Certificate stored in file </opt/emc/scaleio/gateway/temp/certificates/ScaleIO.cer>
Verifying : EMC-ScaleIO-gateway-1.31-1277.3.noarch
1/1

Installed:
  EMC-ScaleIO-gateway.noarch 0:1.31-1277.3

Complete!
```

9.4.6 ScaleIO GUI/Dashboard Installation

NOTE: Prior to installing the GUI/Dashboard it must be validated that JRE 1.7 or above is present on the server.

The ScaleIO GUI/Dashboard used to monitor the ScaleIO Cluster is certified for Windows only. The Windows ScaleIO GUI/Dashboard can be installed using the TPM package named **CITI_WIN_EMCScaleIOGUI_131277_00**. This TPM package will simply drop the **EMC-ScaleIO-gui-1.31-1277.3.msi** executable in the **C:\Temp** directory on the server.

To install the ScaleIO GUI/Dashboard, double-click the ScaleIO-gui-1.31-1277.3.msi executable in the C:\Temp directory. Accept the licensing agreement and select the installation location.

NOTE: On Windows 2008 R2 an “EMC ScaleIO GUI” shortcut will be created automatically. On Windows 2012 or 2012 R2, users wanting a short cut will have to manually created by naming a desktop shortcut “EMC ScaleIO GUI” and pointing it at C:\Program Files (x86)\EMC\ScaleIO\GUI\ui-1.0-SNAPSHOT

After the GUI/Dashboard is installed the ScaleIO cluster can be seen by double-clicking on the shortcut and entering the current MDM Primary IP address, username, and password as shown below.



9.5 ScaleIO Cluster Configuration

The following sections of this document will discuss the configuration of the ScaleIO cluster. These sections should be followed in the order given until the cluster configuration is complete.

9.5.1 Cluster Configuration

Once all servers that will make up the ScaleIO cluster have had their roles defined (MDM, SDS, Tie-Breaker) and had all the necessary software installed, use the following steps to configure the cluster.

Tip: The following command will configure the use of command completion (using the TAB key) when running scli commands

```
# . /etc/bash_completion.d/scli
```

For example

```
# scli --query_all (TAB TAB)
```

```
--query_all           --query_all_device_latency_meters  --query_all_scsi_initiators  --query_all_sds  
--query_all_approved_sdc  --query_all_fault_set    --query_all_sdc              --query_all_volumes
```

Assign the primary MDM IP to the cluster. Note this is the [private network address](#). Also add the management IP address. This is used for external products communicating with the MDM

```
# scli --add_primary_mdm --primary_mdm_ip 10.116.27.245,10.116.27.229 --mdm_management_ip  
10.116.27.213,10.116.27.197,10.116.27.214,10.116.27.198,10.116.19.214,10.116.19.198
```

Before activating a Primary MDM you must read and accept the license agreement.

Press any key to read the license

If you accept the license agreement please press y

y

Successfully added a Primary MDM. Object ID 26fb852257a490dc

Change the initial ScaleIO admin password and authenticate with admin. Note that this internal ScaleIO authentication is not being used to control administrative access. Rather Powerbroker is controlling this (see section 14.1).

```
# scli --login --username admin --password admin
```

Logged in. User role is Administrator. System ID is 5e1ffa2c59b78a3e

Current password must be changed. Please use --set_password command

```
# scli --set_password
```

Enter old password:

Enter new password:

Re-enter new password:

Password changed successfully

```
# scli --login --username admin
```

Enter password:

Logged in. User role is Administrator

Add in the 4 read only functional accounts as described in 14.1.2. These are “readonly”, “snmpread”, “srmread” and “hpnaread”. The password used can be the same and well known as these are read only accounts. Only the first example is given below.

```
# scli --add_user --username readonly --user_role Monitor
```

User readonly created as Monitor ID=d8fcdc0600000001. Default Password is: 'LJfcA9p9'

```
# scli --login --username readonly
Enter password:
Logged in. User role is Monitor. System ID is 7a49a948009a412a
Current password must be changed. Please use --set_password command

# scli --set_password
Enter old password:
Enter new password:
Re-enter new password:
Password changed successfully
```

Add the secondary MDM, the Tie-Breaker, and switch cluster mode and give a name to the cluster (again using the private network IP addresses)

```
# scli --add_secondary_mdm --secondary_mdm_ip 10.116.27.246,10.116.27.230
Successfully added a Secondary MDM. Object ID 001e943079132933

# scli --add_tb --tb_ip 10.116.19.245,10.116.19.229
Successfully added a Tie-Breaker. Object ID 42f6bbb23db2d8e4

# scli --switch_to_cluster_mode
Successfully switched to cluster mode

# scli --rename_system --new_name RACaaS
Successfully renamed system to RACaaS
```

Query the cluster at this point to validate all information entered is correct.

```
# scli --query_cluster
Mode: Cluster, Cluster State: Normal, Tie-Breaker State: Connected
Primary IP: 10.116.27.245, 10.116.27.229
Secondary IP: 10.116.27.246, 10.116.27.230
Tie-Breaker IP: 10.116.19.245, 10.116.19.229
Management IP: 10.116.27.213, 10.116.27.197, 10.116.27.214, 10.116.27.198, 10.116.19.214, 10.116.19.198

Name: RACaaS
```

Add [Protection Domain](#), [Fault Sets](#), and [Storage Pools](#)

Note: Pool names should using the following convention. **[domain]-sp[type]**
SAS Pool Ex: domain1-spas or domain1-spas01
SSD Pool Ex: domain1-spssd
NLSAS Pool Ex: domain1-spnlzas

```
# scli --add_protection_domain --protection_domain_name domain1
Successfully created protection domain. Object ID b3f1b09500000000

# scli --add_fault_set --protection_domain_name domain1 --fault_set_name domain1-fs1
Successfully created a Fault Set. Object ID ae6f2ddb00000000

# scli --add_fault_set --protection_domain_name domain1 --fault_set_name domain1-fs2
Successfully created a Fault Set. Object ID ae6f2ddc00000001
```

```
# scli --add_fault_set --protection_domain_name domain1 --fault_set_name domain1-fs3
Successfully created a Fault Set. Object ID ae6f2ddd00000002

# scli --add_fault_set --protection_domain_name domain1 --fault_set_name domain1-fs4
Successfully created a Fault Set. Object ID ae6f2dde00000003

# scli --add_storage_pool --protection_domain_name domain1 --storage_pool_name domain1-spsas
Successfully created a storage pool. Object ID 3ee6185800000001

# scli --add_storage_pool --protection_domain_name domain1 --storage_pool_name domain1-spssd
Successfully created a storage pool. Object ID 3ee6185900000002
```

Set zero-padding for the storage pools (this will be required for data scrubbing in the future)

```
# scli --modify_zero_padding_policy --protection_domain_name domain1 --storage_pool_name domain1-
spsas --enable_zero_padding
Zero padding policy modified successfully

# scli --modify_zero_padding_policy --protection_domain_name domain1 --storage_pool_name domain1-
spssd --enable_zero_padding
Zero padding policy modified successfully
```

Add all SDS Nodes to the cluster. When adding SDS Nodes to the cluster, pay close attention to the [Fault Set](#) that each SDS is put in. See the [RAM Read Cache Best Practices](#) section of this document to determine cache settings.

In this example we are first creating the SDS devices and adding one of the SAS drives to the storage pool “domain1-spsas”.

For the SDS name, it is recommended to use the hostname of the SDS

For the device name it is recommended to use part of the persistent name which will always be valid for this disk. For example using the last 6 chars of the disk highlighted below

```
# ls -l /dev/disk/by-id/ | grep sdb | grep scsi | awk '{print $9}' | cut -c 33-
2161c0
```

Take care to ensure that the disk device names are correct as they can vary from server to server. Also the RAM cache is set to the value recorded in the planning sheet. The subsequent step will add in SSD devices.

```
# scli --add_sds --sds_ip 10.116.27.245,10.116.27.229,10.116.27.213,10.116.27.197 --sds_ip_role
sds_only,sds_only,sdc_only,sdc_only --device_name 2161c0 --device_path /dev/sdb --sds_name
nagtsnpod00103 --protection_domain_name domain1 --storage_pool_name domain1-spsas --fault_set_name
domain1-fs1 --rmcache_size_mb 10240
Successfully created SDS nagtsnpod00103. Object ID 2ac39f9700000000
```

```
# scli --add_sds --sds_ip 10.116.27.246,10.116.27.230,10.116.27.214,10.116.27.198 --sds_ip_role
sds_only,sds_only,sdc_only,sdc_only --device_name 7b9415 --device_path /dev/sdb --sds_name
nagtsnpod00104 --protection_domain_name domain1 --storage_pool_name domain1-spsas --fault_set_name
domain1-fs2 --rmcache_size_mb 10240
Successfully created SDS nagtsnpod00104. Object ID 2ac39f9800000001

# scli --add_sds --sds_ip 10.116.19.245,10.116.19.229,10.116.19.213,10.116.19.197 --sds_ip_role
sds_only,sds_only,sdc_only,sdc_only --device_name 78e5f6 --device_path /dev/sdb --sds_name
nagtsnpod00203 --protection_domain_name domain1 --storage_pool_name domain1-spsas --fault_set_name
domain1-fs3 --rmcache_size_mb 10240
Successfully created SDS nagtsnpod00203. Object ID 2ac3c6a800000002

# scli --add_sds --sds_ip 10.116.19.246,10.116.19.230,10.116.19.214,10.116.19.198 --sds_ip_role
sds_only,sds_only,sdc_only,sdc_only --device_name 604197 --device_path /dev/sdb --sds_name
nagtsnpod00204 --protection_domain_name domain1 --storage_pool_name domain1-spsas --fault_set_name
domain1-fs4 --rmcache_size_mb 10240
Successfully created SDS nagtsnpod00204. Object ID 2ac3c6a900000003
```

Next add in the remaining drives (below I have just added one additional SAS drive to the SAS pool from each server and one SSD drive to the SSD pool for each server.

First the SAS drives:

```
# scli --add_sds_device --sds_name nagtsnpod00103 --device_path /dev/sdc --storage_pool_name domain1-
spsas --device_name 2944df
Successfully added device 2944df to SDS. New device ID: f5509f7b00000001
# scli --add_sds_device --sds_name nagtsnpod00104 --device_path /dev/sdc --storage_pool_name domain1-
spssd --device_name 9e8c30
Successfully added device 9e8c30 to SDS. New device ID: f5509f7c00010001
# scli --add_sds_device --sds_name nagtsnpod00203 --device_path /dev/sdc --storage_pool_name domain1-
spssd --device_name fd6e18
Successfully added device fd6e18 to SDS. New device ID: f5509f7d0002000
# scli --add_sds_device --sds_name nagtsnpod00204 --device_path /dev/sdc --storage_pool_name domain1-
spssd --device_name 364119
Successfully added device 364119 to SDS. New device ID: f5509f7e00030001
```

For the SSD drives

```
# scli --add_sds_device --sds_name nagtsnpod00103 --device_path /dev/sdx --storage_pool_name domain1-
spssd --device_name f0c201
Successfully added device f0c201 to SDS. New device ID: f5509f7f00000002
# scli --add_sds_device --sds_name nagtsnpod00104 --device_path /dev/sdx --storage_pool_name domain1-
spssd --device_name 628200
Successfully added device 628200 to SDS. New device ID: f5509f8000010002
# scli --add_sds_device --sds_name nagtsnpod00203 --device_path /dev/sdw --storage_pool_name domain1-
spssd --device_name 09c60d
Successfully added device 09c60d to SDS. New device ID: f5509f8200020002
# scli --add_sds_device --sds_name nagtsnpod00204 --device_path /dev/sdw --storage_pool_name domain1-
spssd --device_name b00058
Successfully added device b00058 to SDS. New device ID: f5509f8100030002
```

Next the Dirty Region Log mode for all the SDSs should be set to hardened. This provides much more rapid rebuilds when SDS nodes are rebooted, with a small performance impact.

```
# scli --set_drl_properties --hardened --set_all
Successfully set data-region-logging properties
```

NOTE: After all SDS Nodes have been added to the cluster, a query should be run to validate all SDS Nodes are connected. A query should also be run to validate that all fault sets have the same number of SDS nodes.

```
# scli --query_all_sds
```

Query-all-SDS returned 4 SDS nodes.

```
Protection Domain b3f1b09500000000 Name: domain1
SDS ID: 2ac3c6a900000003 Name: nagtsnpod00204 State: Connected, Joined IP:
10.116.19.246,10.116.19.230,10.116.19.214,10.116.19.198 Port: 7072
SDS ID: 2ac3c6a800000002 Name: nagtsnpod00203 State: Connected, Joined IP:
10.116.19.245,10.116.19.229,10.116.19.213,10.116.19.197 Port: 7072
SDS ID: 2ac39f9800000001 Name: nagtsnpod00104 State: Connected, Joined IP:
10.116.27.246,10.116.27.230,10.116.27.214,10.116.27.198 Port: 7072
SDS ID: 2ac39f9700000000 Name: nagtsnpod00103 State: Connected, Joined IP:
10.116.27.245,10.116.27.229,10.116.27.213,10.116.27.197 Port: 7072
```

```
# scli --query_all_fault_sets --protection_domain_name domain1
```

Protection Domain has 4 Fault Sets:

```
Protection Domain ID: b3f1b09500000000 Fault Set: ID: ae6f2ddb00000000 Name: domain1-fs1 SDS Count: 1
SDS ID: 2ac39f9700000000 Name: nagtsnpod00103
Protection Domain ID: b3f1b09500000000 Fault Set: ID: ae6f2ddc00000001 Name: domain1-fs2 SDS Count: 1
SDS ID: 2ac39f9800000001 Name: nagtsnpod00104
Protection Domain ID: b3f1b09500000000 Fault Set: ID: ae6f2ddd00000002 Name: domain1-fs3 SDS Count: 1
SDS ID: 2ac3c6a800000002 Name: nagtsnpod00203
Protection Domain ID: b3f1b09500000000 Fault Set: ID: ae6f2dde00000003 Name: domain1-fs4 SDS Count: 1
SDS ID: 2ac3c6a900000003 Name: nagtsnpod00204
```

Now set the [cluster spare capacity](#) using the defined best practice. In our example there are only 4 SDSes in the cluster, so we need to set the spare capacity to be $\frac{1}{4} = 25\%$.

```
# scli --modify_spare_policy --protection_domain_name domain1 --storage_pool_name domain1-spsas --
spare_percentage 25
```

Modifying the spare policy might leave some data unprotected in case of failure. Press 'y' to confirm that you want to proceed ?
Successfully modified spare policy.

```
# scli --modify_spare_policy --protection_domain_name domain1 --storage_pool_name domain1-spsd --
spare_percentage 25
```

Modifying the spare policy might leave some data unprotected in case of failure. Press 'y' to confirm that you want to proceed ?
Successfully modified spare policy.

Now set the ScaleIO cluster to remote read-only. This will only allow commands to be run on the primary MDM node which is highly protected from a security perspective.

```
# scli --set_remote_read_only_limit_state --remote_read_only_limit_state enabled
```

Successfully enabled

```
# scli --query_remote_read_only_limit_state
```

Read-only limitation for remote users is enabled.

At this point the SDC restrictions should be enabled. This will allow LUNs to only be mapped to approved SDC nodes.

```
# scli --set_restricted_sdc_mode --restricted_sdc_mode enabled
```

Successfully enabled

```
# scli --query_restricted_sdc_mode
```

Restricted SDC mode is enabled.

Disable the Installation manager which we are not currently using and restart the gateway

```
# vi /opt/emc/scaleio/gateway/webapps/ROOT/WEB-INF/classes/gatewayUser.properties
```

```
...
Restricted SDC mode is enabled. #####Features Enabler#####
features.enable_gateway=true
features.enable_IM=false
features.enable_snmp=false
```

```
# /etc/init.d/scaleio-gateway restart
```

```
# /etc/init.d/scaleio-gateway status
```

The EMC ScaleIO REST Gateway is running. PID=12257

Once all the pools have been completed, the storage pool named “default” should be removed in the protection domain.

```
# scli --remove_storage_pool --protection_domain_name domain1 --storage_pool_name default
```

Successfully removed the storage pool default in Protection Domain domain1

At this point the RAM Read Cache should be disabled for any SSD pools that exist in any protection domain.

```
# scli --set_rmcache_usage --protection_domain_name domain1 --storage_pool_name domain1-spssd --dont_use_rmcache --i_am_sure
```

RAM Read Cache usage is set successfully

Query the license information and send it off to EMC to get a ScaleIO license. See [License Management](#) section of this document for more details.

```
# scli --query_license
```

Installation ID: 319113d342189175

SWID:

Maximum capacity: Unlimited

Usage time left: 29 days (Initial License)

Enterprise features: Enabled

The system was activated 1 days ago

Finally configure the standby MDM as described in the section [Configuring Automated MDM Replacement](#).

9.5.2 Install ScaleIO License

During the ScaleIO cluster configuration a license request to EMC will need to be made. The process is described in detail in the section on [License Management](#). Once the ScaleIO administrator has a valid license key, use the following command to enter that license.

```
# scl_i --set_license --license_key 2P5ZLLMLYIZO8LNBWQLE5W78LBLUPY102OP1
```

9.5.3 REST Gateway Configuration

After JRE and the REST gateway packages have been installed, the gateway properties file needs to be edited to contain the primary, secondary and standby MDM IP addresses. Do not set a gateway admin password. After the edits have been saved, the gateway needs to be restarted.

```
# vi /opt/emc/scaleio/gateway/webapps/ROOT/WEB-INF/classes/gatewayUser.properties

mdm.ip.addresses=10.116.27.213,10.116.27.197,10.116.27.214,10.116.27.198,10.116.19.214,10.116.19.198
gateway-admin.password=

do.not.update.user.properties.with.values.before.upgrade=true

# /etc/init.d/scaleio-gateway status
The EMC ScaleIO REST Gateway is running. PID=10341
# /etc/init.d/scaleio-gateway restart
# /etc/init.d/scaleio-gateway status
The EMC ScaleIO REST Gateway is running. PID=12120
```

NOTE: If the gateway is successful in restarting in the above commands, it should come back with a different PID as shown above.

From another server, test that the Gateway is properly configured by ensuring you can access the REST API and retrieve cluster information.

```
# mdm_ip=10.116.27.213

# curl -X GET -H 'Accept: application/json;version=1.0' -u admin https://$mdm_ip/api/login -k
Enter host password for user 'admin':
"YWRtaW46MTQyMzAwODUxNDUyNjo5ZGFiyZU3MWI0Mjk1NDhjNTQ1YzdiOTM2Zml2ZGUzMw"
# token=YWRtaW46MTQyMzAwODUxNDUyNjo5ZGFiyZU3MWI0Mjk1NDhjNTQ1YzdiOTM2Zml2ZGUzMw

# curl -X GET -H 'Accept: application/json;version=1.0' -u foo:$token https://$mdm_ip/api/instances -k |
python -mjson.tool
....
...
    ],
    "mappingToAllSdcsEnabled": false,
    "name": "esxvol1",
    "sizeInKb": 109051904,
    "storagePoolId": "d4c2a63000000001",
    "useRmcache": false,
    "volumeType": "ThickProvisioned",
    "vtreeld": "ac47ed5a00000000"
```

```
}
]
}
```

Do not forget to now install the REST gateway on the secondary MDM, repeating the above process.

Once the REST gateway is installed on both the current primary and secondary MDMs, you must replace the ScaleIO gateways' self-signed security certificates with Citi "trusted" certificates. This process is documented in the Best Practice section on [Obtaining and Installing Citi trusted certificates](#).

A Virtual IP (VIP) should also be requested for the gateways to provide a single connection point for REST clients that cannot support failover between multiple gateways (e.g. simple REST script). See the section on [Virtual IP \(VIP\) for REST gateways](#) for details on how to request a VIP for the gateways.

9.5.4 Tivoli Event Notification (SNMP/NetCool)

The ScaleIO gateway has a monitoring component which polls the MDM server for any alerts. Alerts are then sent as an SNMP trap to a configurable IP address.

To configure SNMP monitoring, first create a readonly MDM account for use polling the MDM server. This should already have been created in section 9.5.1.

```
# scli --add_user --username snmpread --user_role Monitor
User snmpread created as Monitor ID=d62db15900000001. Default Password is: 'G2XRIA30'

# scli --login --username snmpread
Enter password:
Logged in. User role is Monitor. System ID is 176dcd225102afe9
Current password must be changed. Please use --set_password command

# scli --set_password
Enter old password:
Enter new password:
Re-enter new password:
Password changed successfully
```

Next edit the gateway configuration file to point to the Netcool trap receiver server to pick up the alerts. This varies according to region. For Production environments:

Region	Primary Server/Alias	Secondary Server/Alias
EMEA	emeasystrap01.eur.nsroot.net	emeasystrap02.eur.nsroot.net
NAM	namsystrap01.nam.nsroot.net	namsystrap02.nam.nsroot.net
APAC	tiamsoap1sgp.apac.nsroot.net	tiamsoap1hkc.apac.nsroot.net

For UAT testing contact “*CATE Global Netcool Engineering”.

```
# vi /opt/emc/scaleio/gateway/webapps/ROOT/WEB-INF/classes/gatewayUser.properties

#####Features Enabler#####
features.enable_gateway=true
features.enable_IM=false
features.enable_snmp=true

#####SNMP properties#####
# MDM credentials are only required for SNMP trap support
# For security reasons, it is recommended to use the credentials of a read-only user.
snmp.mdm.username=snmpread
snmp.mdm.password=Citigroup1
# sampling frequency for SNMP traps, in seconds
snmp.sampling_frequency=15
# resend frequency of SNMP traps, in minutes. 0 will send all traps every sampling
snmp.resend_frequency=300
# Supports two ips for two traps receivers. Use a comma (,) or a semicolon (;) as a separator between the two
# receivers IP addresses.
snmp.traps_receiver_ip=namsysstrap01.nam.nsroot.net,namsysstrap02.nam.nsroot.net
snmp.port=162

# /etc/init.d/scaleio-gateway restart
# /etc/init.d/scaleio-gateway status
The EMC ScaleIO REST Gateway is running. PID=27394
```

The snmp.sampling_frequency parameter defines how frequently the MDM is polled to see if there is an Alert.

The snmp.resend_frequency parameter time is the period after which the same alert will be reported a second time. This prevents the same alert being sent repeatedly to the trap receiver, so generating lots of noise.

A test trap can be sent by rebooting an SDS server.

9.5.5 Log Reconciliation (SEMs/ArcSight)

ScaleIO clusters will send all configuration changes that occur on the ScaleIO cluster to SEMs/ArcSight rsyslog server for log reconciliation. All global/regional collectors have already been configured and updated to receive and process ScaleIO logs. However, in order for log reconciliation to start, the following ScaleIO cluster command must be run to send configuration changes to SEMs/ArcSight.

```
# sclli --start_remote_syslog --remote_syslog_server_ip [SEMs Regional Server Name] --
remote_syslog_server_port 514
Successfully started the remote syslog client
```

The SEMs/ArcSight Regional Server Name can be taken from the table below.

NOTE: The regional SEMs/ArcSight server names in the table below are VIPs. As such it is not necessary to have a primary and COB server configured when remote_syslog is enabled.

Region	Primary Server/Alias
NAM	namrsyslog-p01.nam.nsroot.net

EMEA	emearsyslog-p01.eur.nsroot.net
APAC	apacrsyslog-p01.apac.nsroot.net
LATAM	latamrsyslog-p01.lac.nsroot.net

Once the above steps have been completed, any ScaleIO cluster configuration changes, to any ScaleIO cluster, will be sent to all the specified SAN distribution lists for log reconciliation.

9.5.6 Automated SBCC (HPNA)

Hewlett Packard Network Automation (HPNA) is used to do an automated check on all configuration parameters defined in the best practices document. The following items listed below must be completed in order for HPNA to start validating that the current installed configuration is correct.

NOTE: The following instructions assume that the best practice of loading the REST Gateway on the primary and secondary MDMs was followed.

- The primary and secondary MDM nodes must be entered into the Citigroup Aperture system.
- The following items must correctly be completed in Aperture. 1 – All the correct IP address information for each network interface must be entered in the “*Network Information*” tab area. 2 – If the ILO IP Address is entered in the record, the DNS name must include **-ilo** in the name (ex: nagtdsdsaa001-**ilo**.nam.nsroot.net). 3 – NO IP Addresses should be entered in the “*IP Address*” tab area.

Network Information ①								
Port Type	Network Name	Speed	Topology	Duplex	Primary IP	IP Address	Terminal Server Port	Target I
NIC4	DBNull	10000 mb/sec	Gigabit Ethernet	Auto	No	10.59.72.74		rusgtdc-
NIC1	DBNull	10000 mb/sec	Gigabit Ethernet	Auto	Yes	10.59.73.74		rusgtdc-
NIC2	DBNull	10000 mb/sec	Gigabit Ethernet	Auto	No	10.59.73.74		rusgtdc-
ILO	DBNull	1000 mb/sec	Gigabit Ethernet	Auto	No	10.96.251.52 ②		susgtdc-
NIC3	DBNull	10000 mb/sec	Gigabit Ethernet	Auto	No	10.59.72.74		rusgtdc-
IP Address ③								
IP Address								

- The primary and secondary MDM nodes for each cluster will need to have the “*Machine Type*” field changed to “*ScaleIO_RESTGW*” in Aperture as shown below:

Configuration Information				
Machine Category	Pre-Contingency	Platform	Red Hat	Subsyst
Machine Type	① ScaleIO_RESTGW	OS	6Server	Cluster
Database Type	NA	Business Area	CTI / Internal	AMS ID
Backup Type	None	Redundant Type	No Redundancy	AMS ID

- A ScaleIO user with the “monitor” role must be created with the name “hpnaread” as shown below.

```
# scli --add_user --username hpnaread --user_role monitor
User hpnaread created as Monitor ID=d62dff7a00000003. Default Password is: 'dEY7kyio'
```

- The password for the hpnaread user should then be changed to Citigroup1.

```
# scli --login --username hpnaread
Enter password: dEY7kyio
Logged in. User role is Monitor. System ID is 176dcd225102afe9
Current password must be changed. Please use --set_password command

# scli --set_password
Enter old password: dEY7kyio
Enter new password: Citigroup1
Re-enter new password: Citigroup1
Password changed successfully
```

Once the above steps have been completed, HPNA will start validating the settings of the cluster being configured. The following HPNA policies have been created for ScaleIO

- EMC ScaleIO – Misc
- EMC ScaleIO – Pools

Each of the above policies has multiple rules associated with them that check many different parts of the ScaleIO configuration. If any settings in the Cluster configuration do not match what is in the HPNA policies and rules, an exception event will be sent to the “EMC ScaleIO” folder in CitiTools. The EMC ScaleIO folder will show up on the “*Exception Tracking & Reporting Summary*” page under the “*Configuration Assurance – Storage*” drop-down folder.

[Home Page](#) » [Tools Page](#) » [Exception Tracking & Reporting Summary](#)

Summary By: Status:

Active Exceptions By Category

	Name ↕
▶	Configuration Assurance - Network Critical
▶	Configuration Assurance - Network Lockdown
▶	Configuration Assurance - Network Non Lockdown
▶	Configuration Assurance - PSO
▶	Configuration Assurance - Storage 
▶	Configuration Assurance - Voice
▶	Configuration Assurance - VTM

All ScaleIO exceptions will also be sent to the **CTI Cloud Converged Infrastructure DL*.

9.5.7 Chargeback and Performance Reporting (EMC SRM)

9.5.7.1 Chargeback

All chargeback information for ScaleIO clusters supporting VMware environments will be provided using existing VMware chargeback mechanism.

In due course, physical environments will need to have chargeback provided via SRM (in a similar manner to SAN chargeback). However, at the time of writing SRM 3.6.2 SP2 does not have any client information available for ScaleIO clients and hence chargeback via SRM is not possible.

9.5.7.2 Performance reporting setup

Whilst the ScaleIO dashboard can be used to gain instantaneous performance metrics for a ScaleIO cluster, historical and trended performance information for the cluster and its components (e.g. pools) will be available via EMC SRM 3.6.2 SP2 or later.

Once you have completed configuring your cluster, you will need to ask the operational owner of EMC SRM in your datacenter to start to collect data for your new cluster, by configuring within the SRM Discovery Center (under Devices Management > EMC ScaleIO). You will need to provide the SRM admin with the following information:

- The username and password of the “srmread” ScaleIO account (monitor role) to allow EMC SRM to collect ScaleIO configuration and performance data.
- FQDN(s) IP addresses of both of your ScaleIO REST gateways. You should request that one gateway be set up as the “ScaleIO REST gateway” and the other as the “Failover ScaleIO REST gateway” when the SRM admin configures your cluster in the SRM ScaleIO collector. This ensures that SRM can still collect data from the failover gateway if the gateway that is configured as the primary is unavailable.
- Request data collection at 5 minute intervals.

Gateway	SRM Role
Primary gateway (on the nominal primary MDM)	ScaleIO REST gateway
Secondary gateway (on the nominal secondary MDM)	Failover ScaleIO REST gateway

A correct configuration (in the SRM Discovery Center > Devices Management > EMC ScaleIO pane) will look similar to the following example:

EMC ScaleIO

Collected Devices

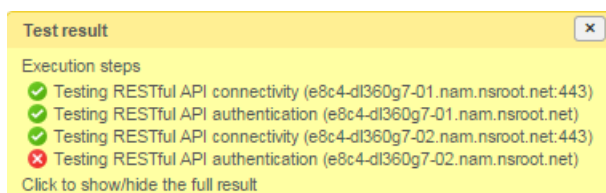
List of known EMC ScaleIO devices by installed location. Click on a line to edit the values.

Showing 1 to 4 of 4 entries

Search:

	Status	Server	Instance	ScaleIO version	ScaleIO REST Gateway IP or hostname	Do you want to configure failover gateway	Failover ScaleIO REST Gateway IP or hostname	ScaleIO username	ScaleIO password
		icl-emcprmvmd2	emc-scaleio	3.1.31 & later	e8c4-dl360g7-01.nam.nsroot.net	Yes	e8c4-dl360g7-02.nam.nsroot.net	srmdread	*****

Note: SRM has a ScaleIO gateway connectivity and authentication test available that the SRM admin can use to confirm connectivity. Due to a known issue in SRM 3.6.2 SP2, the authentication test against the second (failover) gateway will always fail, even when failover is configured (see below). The SRM admin can be told to ignore this test failure, as long as the connectivity test to the second (failover) gateway succeeds.



Correct SRM data collection can be confirmed either by viewing the collected data in the ScaleIO reports in the SRM report library, or by tailing the following log file on the Primary gateway.

tail -f /opt/emc/scaleio/gateway/logs/localhost_access_log.log

where you should see the SRM pulling data from this gateway every 5 minutes with output of the form:

```
10.40.53.141 -- [07/05/2015:08:18:49 +0000] "GET /api/logout HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:22:59 +0000] "GET /api/login HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:22:59 +0000] "GET /api/login HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:22:59 +0000] "GET /api/login HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:22:59 +0000] "GET /api/login HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:00 +0000] "GET /api/login HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:00 +0000] "GET /api/types/System/instances HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:00 +0000] "GET /api/types/System/instances HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:00 +0000] "GET /api/types/System/instances HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:00 +0000] "GET /api/types/System/instances HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:00 +0000] "GET /api/types/Device/instances HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:00 +0000] "GET /api/instances/System::5e19d49223c14dd5/relationships/ProtectionDomain HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:00 +0000] "GET /api/instances/System::5e19d49223c14dd5/relationships/ProtectionDomain HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:01 +0000] "GET /api/instances/System::5e19d49223c14dd5/relationships/Statistics HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:01 +0000] "GET /api/instances/System::5e19d49223c14dd5/relationships/ProtectionDomain HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:01 +0000] "GET /api/instances/StoragePool::35b50d8200000001 HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:01 +0000] "GET /api/instances/ProtectionDomain::6b85ec4300000000/relationships/Statistics HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:01 +0000] "GET /api/instances/ProtectionDomain::6b85ec4300000000/relationships/Sds HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:01 +0000] "GET /api/instances/ProtectionDomain::6b85ec4300000000/relationships/StoragePool HTTP/1.1" 200 -
10.40.53.141 -- [07/05/2015:08:23:02 +0000] "GET /api/instances/ProtectionDomain::6b85ec4300000000 HTTP/1.1" 200 -
etc
```

9.6 ScaleIO Client (SDC) Installation/Configuration

9.6.1 Linux SDC Installation

For a Linux server to have access to LUNs, the [ScaleIO SDC](#) driver must first be installed. The Linux SDC should only be installed on physical Linux servers (Not Linux VMs). Use the following commands install the Linux SDC software:

```
# MDM_IP=10.116.27.213,10.116.27.197,10.116.27.214,10.116.27.198,10.116.19.214,10.116.19.198 yum install EMC-ScaleIO-sdc.x86_64
Resolving Dependencies
--> Running transaction check
---> Package EMC-ScaleIO-sdc.x86_64 0:1.31-1277.3.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
.....
Downloading Packages:
EMC-ScaleIO-sdc-1.31-1277.3.el6.x86_64.rpm                | 2.8 MB    00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : EMC-ScaleIO-sdc-1.31-1277.3.el6.x86_64      1/1
Generating new UUID 74e6b069-baae-468c-89e4-a9f465b645e5
Updating file /bin/emc/scaleio/drv_cfg.txt with MDM (74e6b069-baae-468c-89e4-a9f465b645e5)
Updating file /bin/emc/scaleio/drv_cfg.txt with MDM (10.59.73.74,10.59.73.75,10.59.73.77)
Success loading module /bin/emc/scaleio/scini.ko
Found device special file /dev/scini
Loading MDMs from configuration file /bin/emc/scaleio/drv_cfg.txt
Successfully set UID 74e6b069-baae-468c-89e4-a9f465b645e5
Successfully processed: mdm 10.59.73.74,10.59.73.75,10.59.73.77
Executing boot_wait command. Waiting for 12 seconds
Successfully completed the boot_wait operation
Success configuring module
[ OK ]

Verifying : EMC-ScaleIO-sdc-1.31-1277.3.el6.x86_64      1/1

Installed:
EMC-ScaleIO-sdc.x86_64 0:1.31-1277.3.el6

Complete!
```

Next the storage administrators will need to [provision LUNs to the SDC](#). Once available the following items must be completed in order to create Linux native LVM volumes and mount them. First the following line must be added to the lvm.conf file in order for the native Linux volume manager to recognize ScaleIO LUNs.

```
# vi /etc/lvm/lvm.conf
```

ADD this line in the Types area of the lvm.conf file

```
types = [ 'scini', 16 ]
```

Once the above change is made and native volumes and filesystems are created on ScaleIO LUNs, the **_netdev** mount option must be used as shown below in order for the filesystems to remount after a reboot.

```
#### fstab Example Entry
/dev/mapper/scaleio-vol1 /scaleio_vol1 ext4 _netdev 0 0
```

9.6.1.1 Allowing Linux SDC Access to Multiple Clusters

There will be certain times when a Linux SDC will need to get LUNs to multiple ScaleIO clusters. In order to have an SDC see multiple ScaleIO clusters and remain persistent across reboots, the new cluster MDM IP addresses would have to be added to the `drv_cfg.txt` file using the following commands.

FIRST CAT THE FILE AND SEE THAT ONLY ONE SET OF CLUSTER MDMS IS SHOWN

```
# cat /bin/emc/scaleio/drv_cfg.txt
#Configuration file for the kernel module.

# One line stating the client GUID.
#ini_guid d9b8a46d-4abd-4ec9-abb2-7bd74f6b0ebe

# Additional line for MDM (Meta-data servers)
#mdm 10.20.30.40

ini_guid 5bd30fec-8fbd-4ecb-989d-73fdeade35b5
mdm 10.116.27.213,10.116.27.197,10.116.27.214,10.116.27.198,10.116.19.214,10.116.19.198
```

NOW ADD IN THE NEW CLUSTER MDM IPs

```
# /opt/emc/scaleio/sdc/bin/drv_cfg --add_mdm --ip
10.40.99.167,10.40.99.151,10.40.99.168,10.40.99.152,10.40.99.171,10.40.99.155 --file
/bin/emc/scaleio/drv_cfg.txt
Updating file /bin/emc/scaleio/drv_cfg.txt with MDM (10.40.99.167,10.40.99.151,10.40.99.168,
10.40.99.152,10.40.99.171,10.40.99.155)
Calling kernel module to connect to MDM (10.40.99.167,10.40.99.151,10.40.99.168,
10.40.99.152,10.40.99.171,10.40.99.155)
```

NOW CAT THE FILE AND SEE THAT THERE ARE TWO CLUSTER MDMS SHOWN

```
# cat /bin/emc/scaleio/drv_cfg.txt
#Configuration file for the kernel module.

# One line stating the client GUID.
#ini_guid d9b8a46d-4abd-4ec9-abb2-7bd74f6b0ebe

# Additional line for MDM (Meta-data servers)
#mdm 10.20.30.40

ini_guid 5bd30fec-8fbd-4ecb-989d-73fdeade35b5
mdm 10.116.27.213,10.116.27.197,10.116.27.214,10.116.27.198,10.116.19.214,10.116.19.198
mdm 10.40.99.167,10.40.99.151,10.40.99.168,10.40.99.152,10.40.99.171,10.40.99.155
```

9.6.2 Windows SDC Installation

For any Windows server needing LUNs from a ScaleIO Pool, the [SDC](#) for Windows will need to be installed. The Windows SDC should only be installed on physical servers (Not Windows VMs). The Windows SDC can be installed with the TPM package named **CITI_WIN_EMCScaleIOSDC_131277_00**. This TPM package will simply drop the **EMC-ScaleIO-sdc-1.31-1277.3.msi** executable in the C:\Temp directory on the server.

NOTE: The command window used to run the commands below should be opened as administrator

To install the Windows SDC open a command window and change to the C:\Temp directory and run the following command.

```
C:\Temp> msixexec /i EMC-ScaleIO-sdc-1.31-1277.3.msi
MDM_IP=10.116.27.213,10.116.27.197,10.116.27.214,10.116.27.198,10.116.19.214,10.116.19.198
```

When the above command is executed accept the licensing agreement and choose install and take all default to install the SDC component.



Once the installation is complete, and ScaleIO LUNs have been provisioned to the Windows server, the LUNs will appear as “Fibre Channel EMC ScaleIO SCSI Disk Device” on 2012 disk manager as shown below.



9.6.2.1 Allowing Windows SDC Access to Multiple Clusters

There will be certain times when a Windows SDC will need to get LUNs from multiple ScaleIO clusters. Unlike the Linux SDC, the Windows SDC does not have to save the MDM IP addresses to a file to have the information remain persistent. To add the Windows SDC to an additional ScaleIO cluster, use the following commands.

NOTE: The command window used to run the commands below should be opened as administrator

FIRST RUN A QUERY AND SHOW THAT ONE SET OF CLUSTER MDMS IS SHOWN

```
C:\> "C:\Program Files\emc\scaleio\sdclbin\drv_cfg" --query_mdms
Retrieved 1 mdm(s)
MDM-ID 5e19d49223c14dd5 SDC ID 43c9f0db00000015 INSTALLATION ID 2695178b5a6ec9a3
IPs [0]- 10.116.27.213 [1]- 10.116.27.197 [2]- 10.116.27.214 [3]- 10.116.27.198 [4]- 10.116.19.214 [5]- 10.116.19.198
```

NOW ADD IN THE NEW CLUSTER MDM IPs

```
C:\> "C:\Program Files\emc\scaleio\sdclbin\drv_cfg" --add_mdm --ip=10.40.94.136,10.40.94.137,10.40.94.142
Calling kernel module to connect to MDM (10.40.94.136,10.40.94.137,10.40.94.142)
```

NOW QUERY AGAIN AND VALIDATE THERE ARE TWO CLUSTER MDMS SHOWN

```
C:\> "C:\Program Files\emc\scaleio\sdclbin\drv_cfg" --query_mdms
Retrieved 2 mdm(s)
MDM-ID 5e19d49223c14dd5 SDC ID 43c9f0db00000015 INSTALLATION ID 2695178b5a6ec9a3
IPs [0]- 10.116.27.213 [1]- 10.116.27.197 [2]- 10.116.27.214 [3]- 10.116.27.198 [4]- 10.116.19.214 [5]- 10.116.19.198
MDM-ID 176dcd225102afe9 SDC ID 607206650000000a INSTALLATION ID 798644b514008d01
IPs [0]-10.40.94.136 [1]-10.40.94.137 [2]-10.40.94.142
```

9.6.3 VMWare SDC Installation

The ScaleIO client needs to be installed on each ESXi host in a cluster so that it can access volumes provisioned to it. This should be performed by an ESXi administrator. As a pre-requisite it is currently necessary to put the driver onto a datastore which is accessible by the ESXi host prior to installation.

WARNING: To avoid misconfiguration, before installing the VMware SDC client verify that storage network connections and proper static routes are in place as shown in the [VMware VSI Storage Design Document](#).

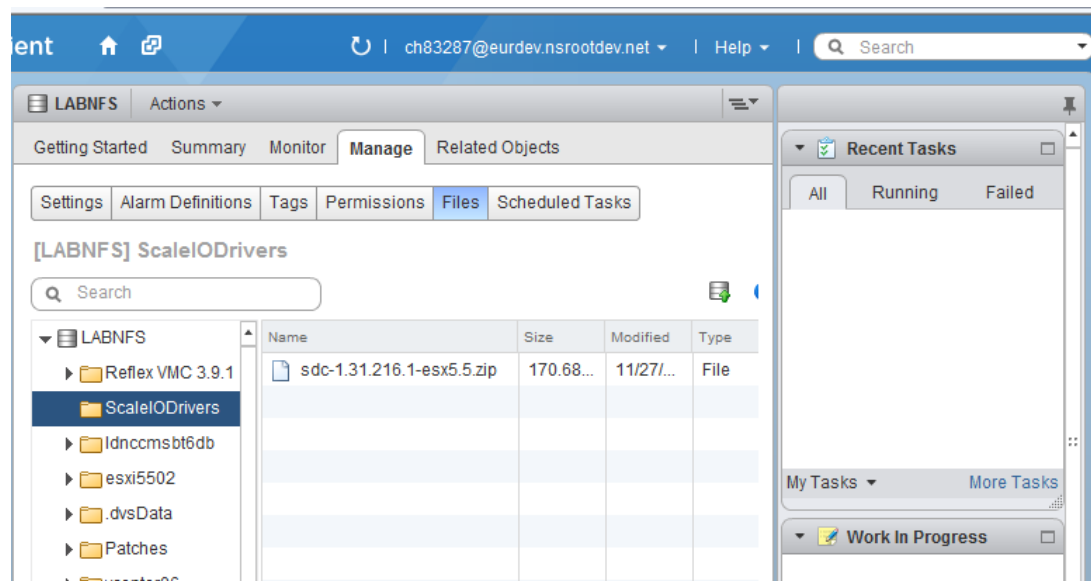
The ScaleIO client for ESXi is available as a vib file. For example:

sdc-1.31.216.1-esx5.zip

This can be downloaded from the GDS repos – for example:

http://rh6lab7-64/SOELinux/repos/testing/esxi55products-x86_64/

Use the VSphere Web UI to place the vib file onto a datastore. For example:



On each ESX host, login as root and install the vib file:

```
# esxcli software vib install -f -d "/vmfs/volumes/bbf70d5b-026092e0/scaleio/sdc-1.31.216.1-esx5.5.zip"
Installation Result
Message: Operation finished successfully.
Reboot Required: false
VIBs Installed: VMware_bootbank_scaleio-sdc_1.31-216.1-esx5.5
VIBs Removed:
VIBs Skipped:
```

Reboot the ESX host

Next configure the ESX host SDC. This will require you to point the ESX host to the MDM IP addresses. Also it is necessary to generate a random GUID for the ESXi host. To do this visit the following website:

<http://www.guidgenerator.com/>

This will provide a GUID – for example:

47d994b7-2e9a-4003-b998-3e5847a0502b

To point to three possible MDM addresses (for example 10.59.73.74,10.59.73.75,10.59.73.77):

```
# esxcli system module parameters set -m scini -p "loctlIniGuidStr=47d994b7-2e9a-4003-b998-3e5847a0502b
loctlMdmIPStr=10.59.73.74,10.59.73.75,10.59.73.77"
```

You can confirm that the setting with the command.

esxcli system module parameters list -m scini

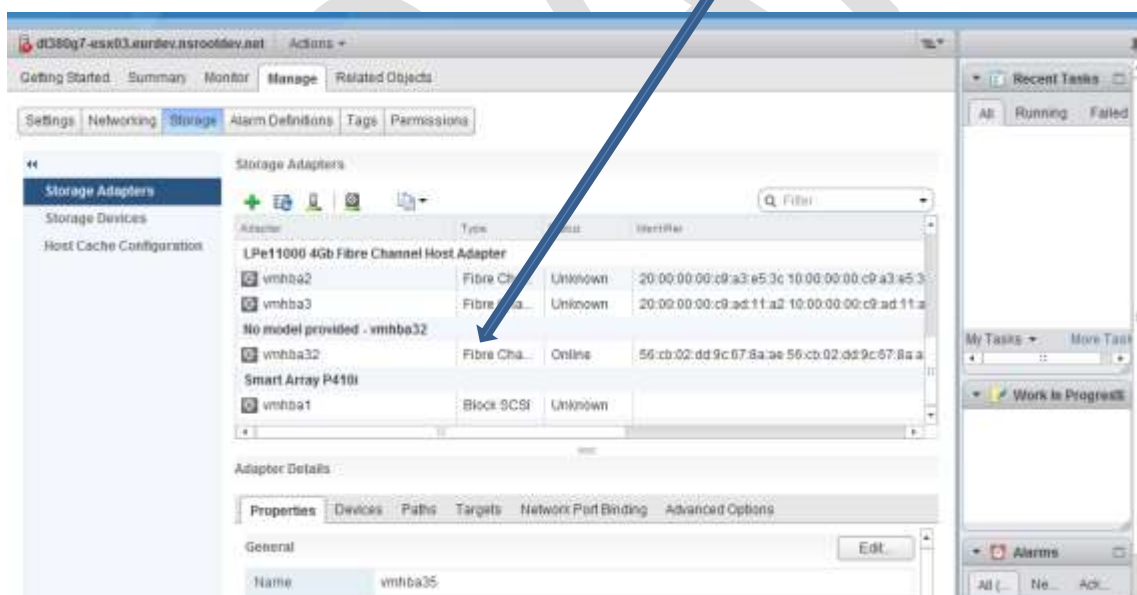
Name	Type	Value	Description
IoctlIniGuidStr	string	47d994b7-2e9a-4003-b998-3e5847a0502b	Ini Guid,
IoctlMdmIPStr	string	10.59.73.74,10.59.73.75,10.59.73.77	Mdms IPs

Load the ESXi module:

esxcli system module load -m scini

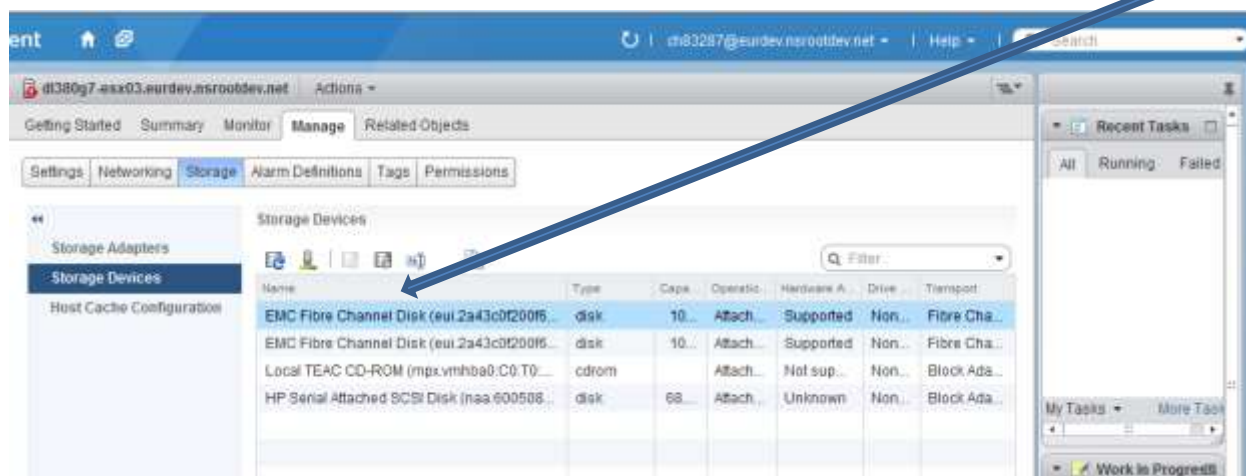
Important note: When the above “`esxcli system module parameters set`” is first run, you do not need to reboot the ESXi server, only load the module as shown above. However, if you later need to update the MDM address list (e.g. because it was typed in incorrectly) then you will need to reboot the ESXi server for the changes to be applied. Note also that, that the “`esxcli system module parameters list`” command above shows what has been set with the corresponding “set” command, but not what is live on the system. The reboot is needed to ensure that the changes made with the “set” command are activated.

At this point on the ESXi host you will see a new storage adapter which appears as an EMC Fibre Channel Disk. In this instance it is labeled vmhba32.



It will appear as an EMC Fibre Channel Disk

You will also see storage devices available once they are provisioned to the ScaleIO SDC:



These appear as EMC FC disks.

Finally increase the queue depth to 256 for each disk:

```
# esxcli storage core device set -d eui.2a43c0f200f6b390cb47bad00000004 -O 256
```

9.6.4 VMWare Plug-in Installation

The VMware vSphere plug-in has limited functionality at this current time. It can provide the ESXi administrator a read-only view of the cluster as remote access to ScaleIO is limited to read-only access. However the ScaleIO GUI provides equivalent information and has a better interface. For these reasons at this time it is recommended not to use this plug-in.

However, for completeness the following steps describe how to install this plug-in

```

VMware vSphere PowerCLI 5.5 Release 2
PowerCLI C:\temp\EMC-ScaleIO-vSphere-plugin-installer-1.31-256.2> .\ScaleIOPluginSetup-1.31-256.2.ps1

cmdlet ScaleIOPluginSetup-1.31-256.2.ps1 at command pipeline position 1
Supply values for the following parameters:
(Type ?? for Help.)
vCenter: UCENTER01
Username: eurdev\ch83287
Password: *****

Choose mode:
1 - To register ScaleIO plugin
2 - To Unregister ScaleIO plugin
3 - To create SUM template
4 - To exit the script
: 1

Welcome to ScaleIO Plugin Register Utility

Do you want to run the Standard or Advanced registration? [s/a]: s
Waiting for internal server to load (attempt 1)...
Connecting to vCenter UCENTER01...
WARNING: There were one or more problems with the server certificate for the
server vcenter01:443:

* The X509 chain could not be built up to the root certificate.
* The certificate's CN name does not match the passed value.

Certificate: [Subject]
E=support@vmware.com, CN=VMware default certificate,
OU=vCenterServer_2014.02.26_155316, O="VMware, Inc."

[Issuer]
E=support@vmware.com, CN=vcenter01.eurdev.nsrootdev.net,
OU=vCenterServer_2014.02.26_155316, O="VMware, Inc."

[Serial Number]
100002

[Not Before]
25/02/2014 15:54:29

[Not After]
24/02/2024 15:54:30

[Thumbprint]
2F7BE3C68B92107184F1E3C96A002628669CF501

The server certificate is not valid.

WARNING: THE DEFAULT BEHAVIOR UPON INVALID SERVER CERTIFICATE WILL CHANGE IN A
FUTURE RELEASE. To ensure scripts are not affected by the change, use
Set-PowerCLIConfiguration to set a value for the InvalidCertificateAction
option.

Successfully connected to vCenter UCENTER01
Registering ScaleIO extension...

Log out, and then log back in to vSphere web client. The plugin is downloaded up
on login to the vSphere web client. After you have logged back in, press ENTER.

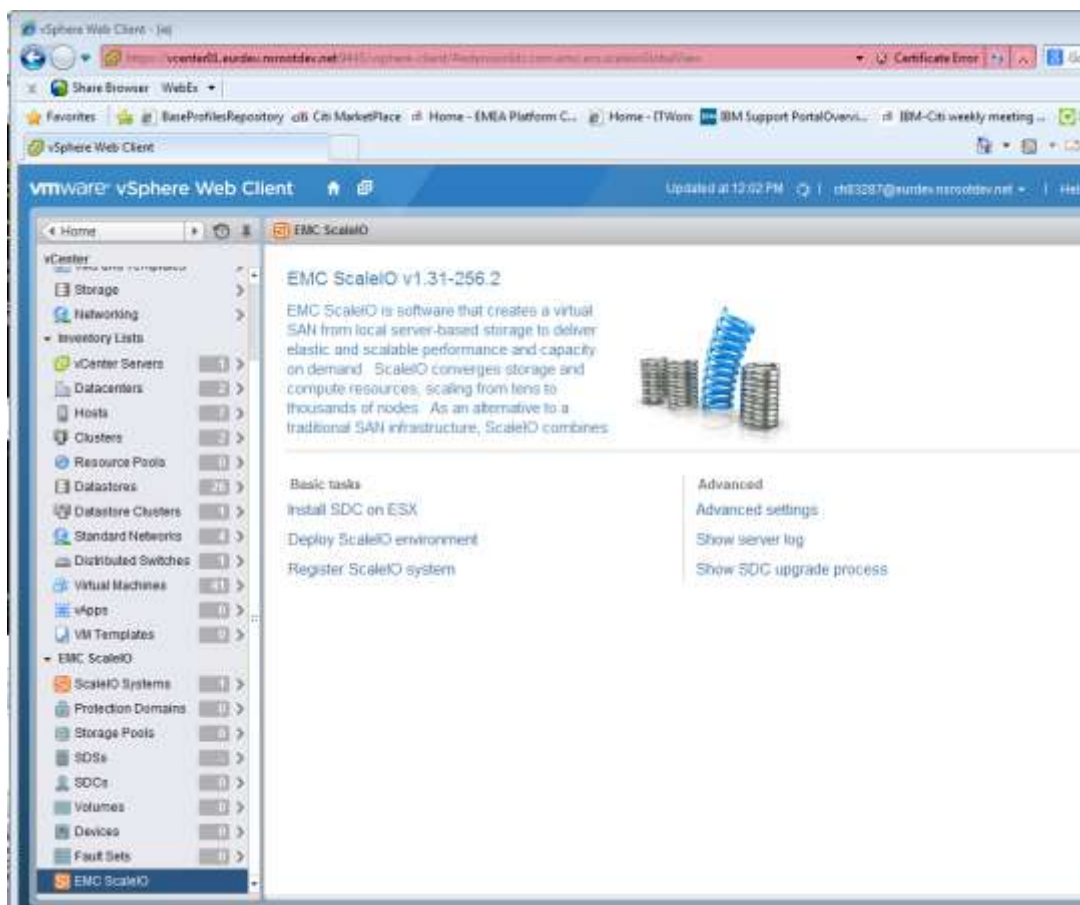
Choose mode:
1 - To register ScaleIO plugin
2 - To Unregister ScaleIO plugin

Log out, and then log back in to vSphere web client. The plugin is downloaded up
on login to the vSphere web client. After you have logged back in, press ENTER.

Choose mode:
1 - To register ScaleIO plugin
2 - To Unregister ScaleIO plugin
3 - To create SUM template
4 - To exit the script
: 4
Disconnecting from the vCenter UCENTER01...
PowerCLI C:\temp\EMC-ScaleIO-vSphere-plugin-installer-1.31-256.2>

```

Once complete, login to the vSphere Web Client and check you can now see the ScaleIO plugin:



To register a system select “Register ScaleIO system”. When prompted provide the MDM IP address and details of a read-only account. Once registered you will be able to view the components of the cluster, but will be unable to make any modifications to the cluster.

9.6.5 Provision LUNs for Physical Environments (Linux/Windows)

Create LUNs and validate the creation with a LUN query. [LUN sizing](#) guidelines discussed in the best practices section should be observed.

```
# scll --add_volume --volume_name scott01 --size_gb 32 --protection_domain_name domain1 --
storage_pool_name domain1-spsas
Successfully created volume of size 32 GB. Object ID 87e7741b00000000

# scll --add_volume --volume_name scott02 --size_gb 32 --protection_domain_name domain1 --
storage_pool_name domain1-spsas
Successfully created volume of size 32 GB. Object ID 87e7741c00000001

# scll --query_all_volumes
Query-all-volumes returned 2 volumes
Protection Domain b3f1b09500000000 Name: domain1
Storage Pool 3ee6185700000000 Name: default
<No volumes defined>
```

```
Storage Pool 3ee6185800000001 Name: domain1-spsas
Volume ID: 87e7741b00000000 Name: scott01 Size: 32.0 GB (32768 MB) Unmapped Thick-provisioned
Volume ID: 87e7741c00000001 Name: scott02 Size: 32.0 GB (32768 MB) Unmapped Thick-provisioned

Storage Pool 3ee6185900000002 Name: domain1-spssd
<No volumes defined>
```

Note that in environments where the clients are dual-connect to the storage nodes (e.g. Tornado++ environments), then it is possible on rare occasions that an SDC may appear twice (with different SDC IDs) if there are some connectivity issues to the new host. Hence, to avoid issues later, it is recommended that before approving the SDC of a dual-connected host to join the ScaleIO cluster, you check that only one of its IP addresses appears in the SDC listing.

You will need the host SA to supply both IP addresses (on eth0 and eth2) that are used by the client to connect to the storage network. In the example below, the new SDC is correctly connected via its secondary adapter (IP = 10.123.96.91) and the primary interface is disconnected (IP = 10.123.96.91) with no GUID shown.

```
# scli --query_all_sdc | egrep "10.123.96.91|10.123.32.91"
SDC ID: e321e57f00000024 Name: N/A IP: 10.123.96.91 State: Connected GUID: 5CB77F24-2269-4559-86A1-DE33C4982E02
Approved: no
SDC ID: e321e59300000038 Name: N/A IP: 10.123.32.91 State: Disconnected GUID: 00000000-0000-0000-0000-000000000000
Approved: no
```

Under normal circumstances, only one SDC ID will be shown. If an SDC is appearing twice, then you will need to follow the following process:

- 1) Ask the SA to remove the SDC software from their host
- 2) When completed, you should then remove the SDC entries from the cluster

```
# scli --remove_sdc --sdc_id e321e57f00000024
Successfully removed sdc with ID e321e57f00000024

# scli --remove_sdc --sdc_id e321e59300000038
Successfully removed sdc with ID e321e59300000038
```

- 3) Instruct the SA to re-install the SDC software on their host, and when completed, check as above that the SDC is now only appearing once.

Before a LUN can be provisioned to an SDC, the SDC must be approved by the ScaleIO administrator using the following commands.

```
# scli --query_all_sdc
MDM restricted sdc mode: enabled
Query all SDC returned 1 SDC nodes.
SDC ID: bf614a4f00000000 Name: N/A IP: 10.116.19.213 State: Connected GUID: 4E8119D3-0338-4C6C-89BA-466316B835AC Approved: no
  Read bandwidth: 0 IOPS 0 Bytes per-second
  Write bandwidth: 0 IOPS 0 Bytes per-second

# scli --query_all_approved_sdc
MDM restricted sdc mode: enabled

# scli --add_sdc --sdc_name nagtsnpod00203 --sdc_ip 10.116.19.213
Successfully approved sdc Object ID bf614a4f00000000 ip 10.116.19.213 guid N/A

# scli --query_all_approved_sdc
MDM restricted sdc mode: enabled
SDC ID: bf614a4f00000000 Name: nagtsnpod00203 IP: 10.116.19.213 State: Connected GUID: 4E8119D3-0338-4C6C-89BA-466316B835AC Approved: yes
  Read bandwidth: 0 IOPS 0 Bytes per-second
  Write bandwidth: 0 IOPS 0 Bytes per-second
```

The newly create LUNs can now be provisioned to an SDC.

```
# scli --map_volume_to_sdc --volume_name scott01 --sdc_name nagtsnpod00203
Successfully mapped volume scott01 to SDC nagtsnpod00203.

# scli --map_volume_to_sdc --volume_name scott02 --sdc_name nagtsnpod00203
Successfully mapped volume scott02 to SDC nagtsnpod00203.

# scli --query_sdc --sdc_name nagtsnpod00203
SDC ID: bf614a4f00000000 Name: nagtsnpod00203 IP: 10.116.19.213 State: Connected GUID: 4E8119D3-0338-4C6C-89BA-466316B835AC Approved: yes
  Read bandwidth: 0 IOPS 0 Bytes per-second
  Write bandwidth: 0 IOPS 0 Bytes per-second

Volume information (total 2 volumes):
  1: Volume ID: 87e7741c00000001 Name: scott02 Thick-provisioned Size: 32.0 GB (32768 MB)
  2: Volume ID: 87e7741b00000000 Name: scott01 Thick-provisioned Size: 32.0 GB (32768 MB)
```

Note that in R1.31 there is a bug that can cause the creation for large volumes to fail if executed in very quick succession (e.g. from inside a loop in a script). If creating volumes via a scripted loop with R1.31, you should put a “sleep 1” in your script to avoid the failures from occurring.

Note also that you cannot provision from a cluster that is currently rebuilding.

Once a new volume has been created, it may be necessary to throttling the amount of I/O that that volume will be allowed to service. See the hardware platform-specific guide (e.g. “ScaleIO Environments on Tornado++ Hardware” Best Practice guide) for details on which tiers require their volumes be throttled, and the instructions on how to do this.

9.6.6 Provision LUNs to Virtualized Environments (ESXi Hosts)

Prior to LUNs being provisioned to an ESXi environment, the SDC on each host in the cluster must be approved for LUNs while in restricted mode. Use the following commands to approve the ESXi SDC.

```
# scli --query_all_approved_sdc
MDM restricted sdc mode: enabled

# scli --add_sdc --sdc_name dl380g7-esx03--sdc_ip 10.154.219.172
Successfully approved sdc Object ID 43c9549b00000001 ip 10.154.219.172 N/A

# scli --query_all_approved_sdc
MDM restricted sdc mode: enabled
SDC ID: 43c9549b00000001 Name: dl380g7-esx03 IP: 10.40.219.172 State: Connected GUID: EBFC8BAB-5E57-4DBD-8BE2-0CBB973FDD73 Approved: yes
  Read bandwidth: 0 IOPS 0 Bytes per-second
  Write bandwidth: 0 IOPS 0 Bytes per-second
```

LUNs provisioned to ESXi hosts must be provisioned to all ESXi hosts in a cluster. This is to ensure all the hosts can see the same storage and thus permit Vmotion between the ESXi hosts with no storage Vmotion required.

```
# scli --add_volume --volume_name esxvol1 --size_gb 100 --protection_domain_name domain1 --
storage_pool_name domain1-spsas
Successfully created volume of size 32 GB. Object ID 3ab6c00f00000000

# scli --map_volume_to_sdc --volume_name esxvol1 --sdc_ip 10.154.219.172 --allow_multi_map
Successfully mapped volume esxvol1 to SDC 10.154.219.172.

# scli --map_volume_to_sdc --volume_name esxvol1 --sdc_ip 10.154.219.126 --allow_multi_map
Successfully mapped volume esxvol1 to SDC 10.154.219.126.

# scli --query_all_volumes
Query-all-volumes returned 1 volume
Protection Domain 346b28d300000000 Name: domain1
Storage Pool d4c2a62f00000000 Name: default
<No volumes defined>

Storage Pool d4c2a63000000001 Name: domain1-spsas
Volume ID: 0cb4549e00000000 Name: esxvol1 Size: 104.0 GB (106496 MB) Mapped to 2 SDC Thick-
provisioned
```

At this point the mapped volume will be visible on the ESXi hosts and should be used to map to a datastore.

Note that in R1.31 there is a bug that can cause the creation for large volumes to fail if executed in very quick succession (e.g. from inside a loop in a script). If creating volumes via a scripted loop with R1.31, you should put a “sleep 1” in your script to avoid the failures from occurring.

Once a new volume has been created, it may be necessary to throttling the amount of I/O that that volume will be allowed to service. See the hardware platform-specific guide (e.g. “ScaleIO

Environments on Tornado++ Hardware” Best Practice guide) for details on which tiers require their volumes be throttled, and the instructions on how to do this.

9.6.7 Pool selection

Where multiple pools of the same storage type are available with the same ScaleIO cluster for the required tier (e.g. the SAS tier in a Tornado++ configuration), then you will need to choose from which of these pools you should provision.

As a general rule, for new physical clients, or a new set of ESXi hosts forming a new mobility domain, you should provision out of the pool with the highest amount of storage capacity available for volumes. The Pool capacity planning section in this document describes how to determine this metric for a pool, as well as the %utilization of the pool.

For provisioning additional storage from the same tier to an existing client, the new volumes should be allocated from the same pool in that tier that the initial allocations were made from. In this context a client is one or more hosts that are provisioned the same shared storage (e.g. one or more physical servers or multiple ESXi servers in a mobility domain).

The maximum level of pool utilization is defined in the hardware-platform-specific ScaleIO Best Practice guide (e.g. the ScaleIO Environments on Tornado++ Hardware Best Practice document)

One exception to the above rules will be when a pool has free capacity, but is overloaded from a workload perspective (excessive IOPS). In such a situation, no further allocations should be made from that pool without further investigation of the pool loading. For each Citi certified hardware platform that ScaleIO is supported on, CATE will provide an estimate of the maximum number of host IOPS that a given pool can support. For example, these figures for the Tornado++ pools can be found in the document “ScaleIO Environment on Tornado++ Hardware” Best Practice document. The Pool capacity planning section in this document describes how to determine the IOPS metric for a given pool.

Best practice for physical hosts and environments where ScaleIO thin provisioning is in use will be added to a future update to this document. ScaleIO thin provisioning is not currently supported within Citi.

9.6.8 Pool capacity planning

When allocating storage from a mature ScaleIO pool, one should first ensure that the pool is not already overloaded from a capacity or IOPS perspective. Such checks are not required for a newly commissioned pool.

The IOPS capabilities for each pool in a Tornado++ installation can be found in the “ScaleIO Environments on Tornado++ Hardware” Best Practice guide. In this section, we outline how to determine pool capacity and IOPS utilization using various CATE certified tools.

9.6.8.1 Capacity utilization

Current capacity utilization of a pool (or pools) can be determined in a number of ways:

1) Via ScaleIO CLI:

```
# scli --query_storage_pool --storage_pool_name <POOL-NAME> --protection_domain_name <DOMAIN-NAME> | egrep "available"
```

For example

```
# scli --query_storage_pool --storage_pool_name domain1-spsas --protection_domain_name domain1 | egrep "available"
```

```
Storage Pool domain1-spsas (Id: 35b50d8200000001) has 7 volumes and 264.0 GB (270336 MB) available for volume allocation
```

This output tells you that volumes with a total usable (mirrored) capacity of 264GB can be created in this pool.

2) Via ScaleIO Dashboard:



The above figure shows the detailed report for a specific pool (domain1-spsas). Here we also see at the bottom that 264GB of space is available for volume creation. Capacity utilization (51.9%) is also shown. This figure is (used raw capacity) / (used raw capacity + free raw capacity), and (correctly) ignores separate capacity assigned as spare capacity for rebuilds.

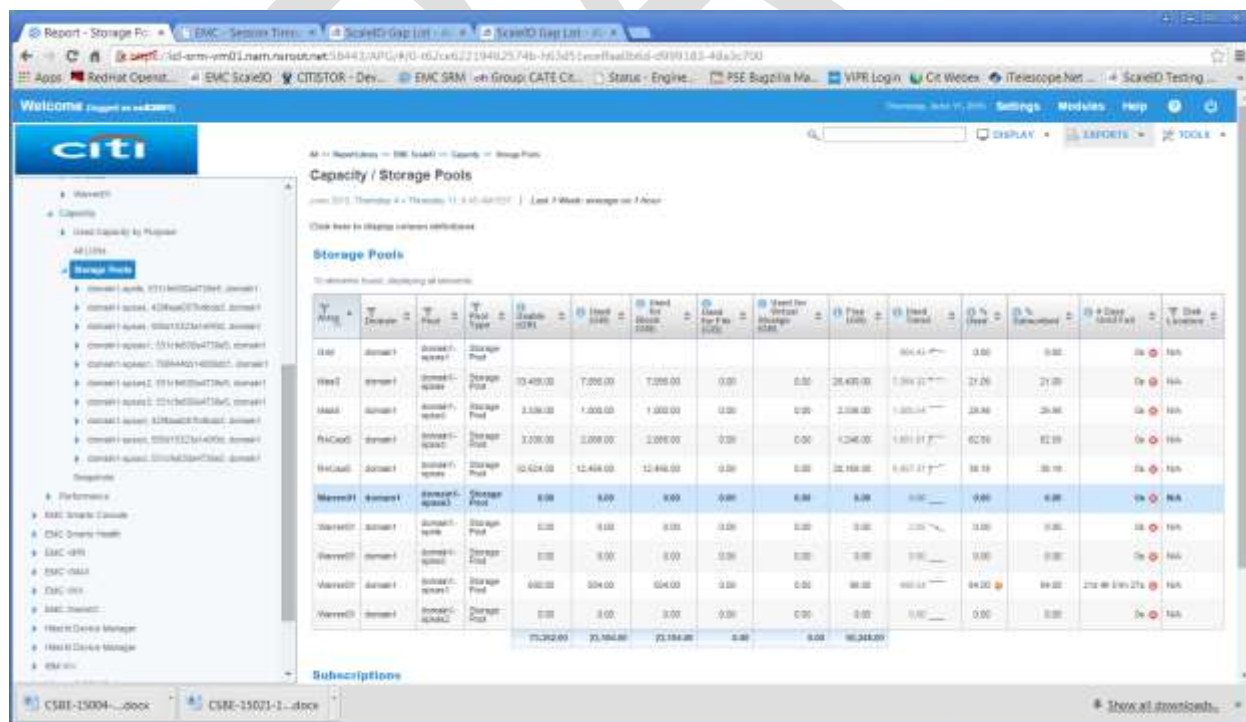
Alternatively, the % capacity in use information can be viewed for all pool simultaneously with the following dashboard view. This view does not provide the “Available for Allocation” metric, but does still show that there is plenty of free space available for new allocations. This example only shows one pool, as the cluster it was obtained from is only configured with a single pool.



3) Via ViPR SRM (with SRM ScaleO Solutions Pack 1.1.3 or later)

The following EMC SRM report has the advantage over the above ScaleIO tool-based reports in that it can report on multiple pools across multiple clusters in a single view. In this report the “Free (GB)” column is equivalent to the “Available for Allocation” reports in the ScaleIO tools, as described above, and indicated the capacity of volumes that can be created from a given pool.

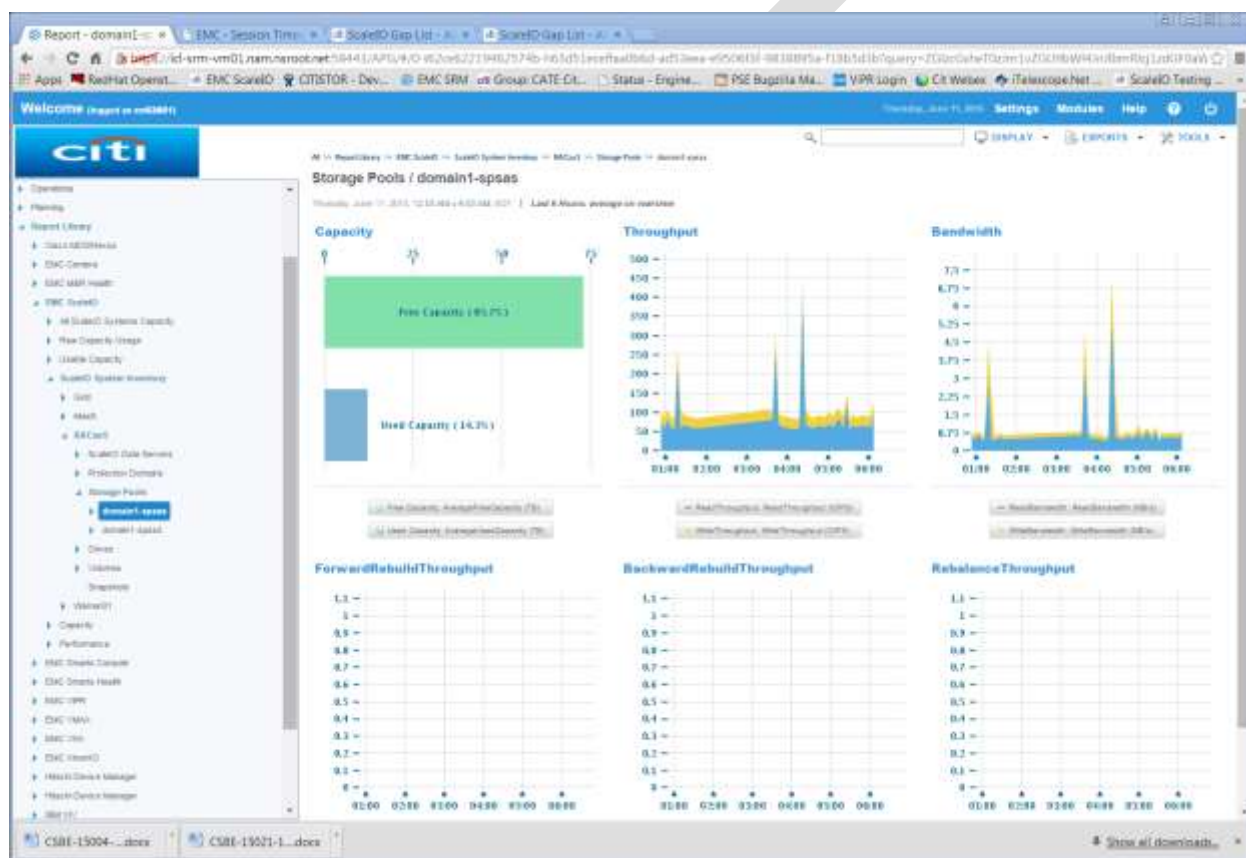
All>>>Report Library>>EMC ScaleIO>>Capacity>>Storage Pools



Pool-specific reports are also available that show both free and used capacities as a percentage (excluding the capacity that is marked as spare for that pool). **However, with the current release of the SRM ScaleIO SolutionsPack, these do not give accurate usage % information and must not be used.**

All>>Report Library>>EMC ScaleIO>>ScaleIO System Inventory>>*cluster-name*>>Storage Pools>>*pool-name*

This view also shows the average read and write IOPS (marked as Throughput) and Bandwidth (MB/s) of the pool for the time period selected.



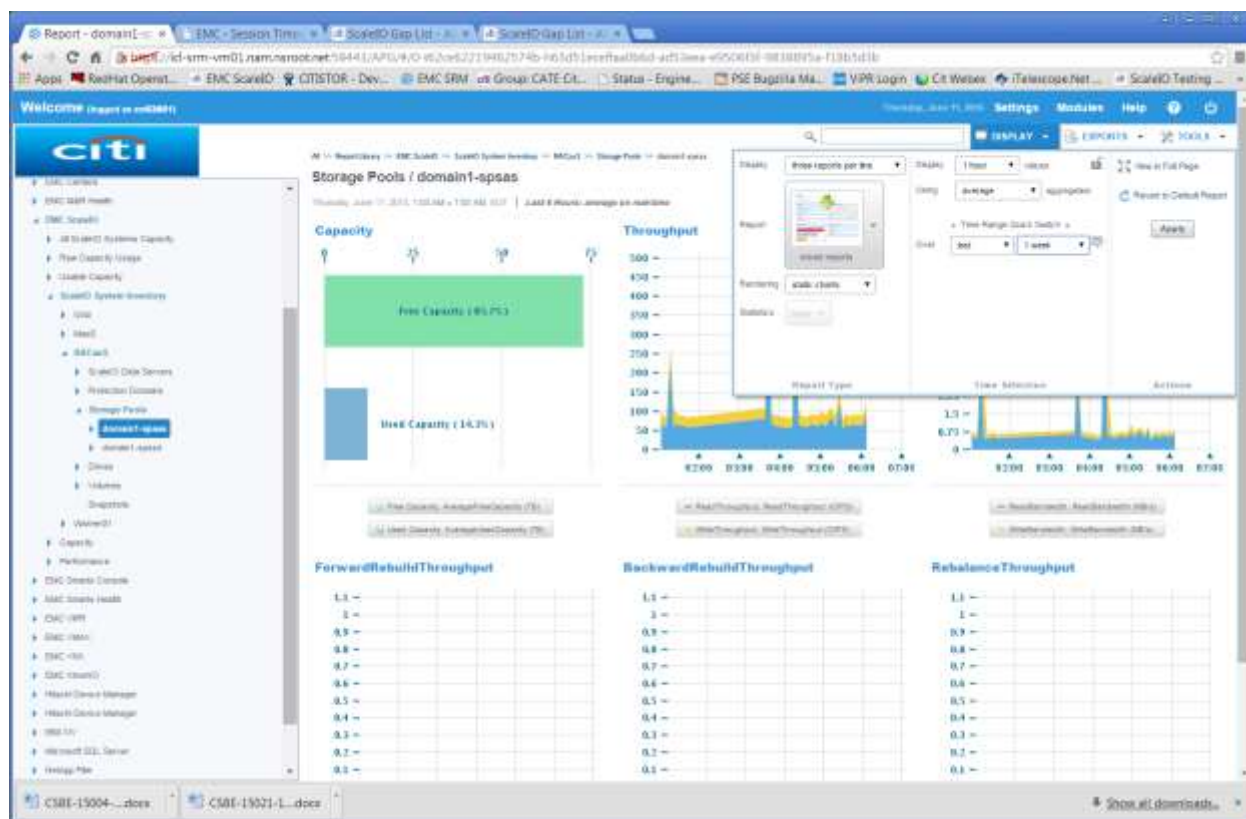
9.6.8.2 IOPS utilization

Via ViPR SRM (with SRM ScaleIO Solutions Pack 1.1.3 or later)

Historical (trended) performance data can be obtained from the canned EMC report for host IOPS for ScaleIO in SRM, as shown above.

All>>Report Library>>EMC ScaleIO>>ScaleIO System Inventory>>*cluster-name*>>Storage Pools>>*pool-name*

The period for which the performance data is shown can be selected using the “DISPLAY” pull down menu as shown below, and changing the “Time Range Quick Switch” entries. A minimum range of 1 week is recommended to get accurate estimate of the average IOPS load.



2) Via ScaleIO Dashboard:

If historical SRM data is not available, the ScaleIO Dashboard can show the current (instantaneous) pool IOPS. It cannot show any historical data.

The current IOPS usage across multiple pools within a cluster can be viewed using the following view (Backend > Application IO, with “By Storage Pools” selected).



The screenshot shows the ScaleIO Application I/O dashboard. The table displays IOPS usage across multiple storage pools. The columns are: Item, Total, IOPS, IO Size, Read, IOPS, IO Size, Write, IOPS, IO Size, and Read Write. The data is as follows:

Item	Total	IOPS	IO Size	Read	IOPS	IO Size	Write	IOPS	IO Size	Read Write
Warren1	2.3 MB/s	104	8.1 KB	5.8 MB/s	127	8.3 KB	104.0 KB/s	127	8.3 KB	104.0 KB/s
domain1	2.0 MB/s	104	8.1 KB	1.0 MB/s	127	8.3 KB	104.0 KB/s	127	8.3 KB	104.0 KB/s
domain1-spsas	2.0 MB/s	104	8.1 KB	1.0 MB/s	127	8.3 KB	104.0 KB/s	127	8.3 KB	104.0 KB/s

Important: You must ignore the Total IOPS count as this includes two backend writes IOs for every host write IO. The true total host IOPS can only be obtained by adding the Read and Write IOPS ($127 + 121 = 248$) and excluding the “2nd write” figures

9.7 Backups

Only the O.S. needs to be backed up. This is to ease the retrieval of host specific O.S. components such as GSF configuration parameters specific to the host. There is no requirement to backup any of the ScaleIO software components or data

10 Configuration Assurance

The Storage Operations team will verify that the ScaleIO Cluster was installed and configured properly and will verify that the [Installation](#) or [Upgrade](#) checklist was followed before the device is moved into production. Once the device is in production Storage Operations will utilize HPNA to verify that solution was installed and configured properly, and provide on-going configuration management.

For a list of HPNA policies and rules that will be created refer to the [Automated SBCC \(HPNA\) section](#) of this document.

11 Scalability & Capacity

EMC ScaleIO is designed to scale out to hundreds and (theoretically) thousands of nodes (small commodity servers). The parallel architecture and distributed volume layout deliver a massively parallel system that delivers I/O operations through a distributed system. As a result, both performance and capacity can scale linearly with the number of SDS servers and disks in the cluster. When more capacity is required, additional SDS nodes can be added to the cluster while normal I/O operations continue.

12 Availability Management

ScaleIO has been designed with auto-monitoring & auto-healing processes that deal with loss of disks and nodes without requiring administrator intervention. Nodes can also be added or removed on the fly, and the software will immediately respond to the changes rebalancing the storage across the new configuration.

13 Regional or Merger Partner Consideration

None.

14 Global Standard (Security)

This section describes how ScaleIO needs to be configured to ensure it is secure. The security model used by ScaleIO is rapidly evolving and the implementation in v1.31 is significantly

different from that which will be deployed in v1.40. This is described in detail in the following document:

http://ewiki.nam.nsroot.net/w/images/f/f6/ScaleIO_Security_Model_071014.pdf

Note this document is continuously evolving.

14.1 Authentication and Authorization

ScaleIO v1.31 has an internal user database and a small number of pre-defined roles. This internal authentication/authorization mechanism does not currently comply with the Citi CISS policies. Hence for the ScaleIO v1.31.1277 release it is intended not to use this for authentication, but wait for v2.0 at which point it will be fully integrated with Active Directory (AD).

14.1.1 Administrative Access

For ScaleIO v1.31.1277 all ScaleIO cluster administration will be performed by the root user account on the MDM servers in the cluster. No administrative tasks can be run on a server other than the MDM server. This is accomplished by configuring ScaleIO to disable all remote administrative privileges. Remote access is only possible as read-only.

To gain administrative access a user first has to log onto the primary MDM server via kerberos/LDAP. There is a host access control list defined in LDAP which limits who can login to the server. Next the user needs to switch to the root account. Root access is controlled by Powerbroker.

Once logged in as root, the administrator then needs to login to ScaleIO (which has an internal account database). As mentioned this login is not being used to control access, but we cannot completely disable within the ScaleIO software and so a well-known login/password will be used.

Emergency console access is controlled by PAR as per the GSF.

14.1.2 Non-administrative Access

There are a number of external systems which require external access to the ScaleIO cluster for read-only access. As mentioned earlier all remote access to the ScaleIO cluster is configured to be read-only by default, regardless of the role assigned to the account within ScaleIO

14.1.3 CLI authentication

When accessing the ScaleIO CLI, the application does allow for the password to be entered in clear text as shown below.

```
# scli --login --username admin --password Test1234
Logged in. User role is Administrator. System ID is 5e19d49223c14dd5
```

This method of gaining authentication via the CLI should never be used as this will cause the clear text password to be recorded in the UNIX history files for the user. When authenticating at the CLI the user should never use the `--password` flag which will always cause the user to be prompted for the password. As a result, the password will never be shown or recorded in clear text.

```
# scli --login --username admin
Enter password: *****
Logged in. User role is Administrator. System ID is 5e19d49223c14dd5
```

14.2 Configuration

To ensure that all remote access to the ScaleIO cluster is read-only run the following commands:

```
# scli --set_remote_read_only_limit_state --remote_read_only_limit_state enabled
Successfully enabled
# scli --query_remote_read_only_limit_state
Read-only limitation for remote users is enabled.
```

14.2.1 Required Initial Permission Changes

As a result of the internal ScaleIO CIVA Scan, there were certain files found to contain excess permissions that must be changed to meet compliance. A list of these files and permissions were turned over to EMC and will be resolved in the v1.32 release of ScaleIO. However, for the v1.31 release of ScaleIO, if any of the following files are contained on any node components of ScaleIO, the permissions must be changed to 640 as shown below.

```
# chmod 640 /opt/emc/scaleio/mdm_failover/cfg/conf.txt
# chmod 640 /opt/emc/scaleio/gateway/webapps/ROOT/resources/ScaleIO_Complete_Config.csv
# chmod 640 /opt/emc/scaleio/gateway/webapps/ROOT/resources/ScaleIO_Minimal_Config.csv
# chmod 640 /opt/emc/scaleio/gateway/webapps/ROOT/WEB-INF/classes/gatewayUser.properties
# chmod 640 /opt/emc/scaleio/gateway/webapps/ROOT/WEB-INF/classes/gatewayUser.properties.backup.merged
```

14.3 ScaleIO Accounts

The following ScaleIO user accounts are to be defined with the following passwords and roles. Note that at present these accounts are not used to control access and all are read-only apart from the admin account when accessed on an MDM

Account	Password	Role	Comments
readonly	Citigroup1	Monitor	General purpose read-only account for the dashboard, Vsphere Web client etc...
snmpread	Citigroup1	Monitor	SNMP read-only account
srmread	Citigroup1	Monitor	SRM read-only account via REST API
hpnaread	Citigroup1	Monitor	HPNA read-only account via REST API

DRAFT

15 Relevant Links

Document/Web Site	Location
CATE Scale-out commodity block storage position paper	https://catecollaboration.citigroup.net/domains/platstor/gdse/products/Eval/Scale_Out_Block_storage_position_paper_V1.0.pdf
ScaleIO v1.31 Requirements and Release Notes Document	https://catecollaboration.citigroup.net/domains/platstor/gdse/products/platforms/EMC%20ScaleIO/CSBE-15007-RN.pdf
ScaleIO Product Page	https://catecollaboration.citigroup.net/domains/platstor/gdse/products/techstdshmpgs/emcscaleIO.aspx
ScaleIO Standard Build Page	https://catecollaboration.citigroup.net/domains/platstor/gdse/products/platforms/EMC%20ScaleIO/ScaleIO_csb.aspx
ScaleIO Security Model	http://ewiki.nam.nsroot.net/w/images/f/f6/ScaleIO_Security_Model_071014.pdf
VMWare VSI Storage	https://catecollaboration.citigroup.net/domains/platstor/gdse/products/Designs/CSBE-11079-DD.pdf
ScaleIO on Tornado++ Hardware best practices	https://catecollaboration.citigroup.net/domains/platstor/gdse/products/platforms/EMC%20ScaleIO/CSBE-15004-BP.pdf

16 Glossary/Definition

Term or Acronym	Explanation or Definition
CATE	Citi Architecture & Technology Engineering
GNAST	Global Network Architecture and Standards Team
NMS	Network Management System
HPNA	Hewlett Packard Network Automation
MDM	ScaleIO Meta Data Manager
SDS	ScaleIO Data Server
SDC	ScaleIO Data Client

17 Appendixes

17.1 Appendix A - Operational Procedures

17.1.1 License Management

When ScaleIO is first installed, it is enabled with a temporary license that is valid for only 30 days. You need to obtain a permanent license before going into production.

You can get a ScaleIO either by raising an SR with EMC, or using the the Powerlink Self-service portal. Both are described below.

17.1.1.1 Obtaining a permanent ScaleIO license

When ScaleIO is first installed, it is enabled with a temporary license that is valid for only 30 days. You need to obtain a permanent license before going into production.

Licenses can only be obtained by the “Emerging Data and Network Technologies” team (their DL is *CTI GLOBAL Emerging Data and Network Technologies) as they have the required knowledge of the ELA details.

Citi has access to ScaleIO via an existing extended licensing agreement (ELA). ScaleIO licences can be drawn down against this agreement by opening an EMC Service Request (SR) on-line at <http://support.emc.com>. Licensing is only needed for the storage cluster, not client software. An example of the required SR text is shown below. The SR should only be raised by the “Emerging Data and Network Technologies” team, and they will substitute in the actual ELA number to be used.

This ticket is to request a key for a new ScaleIO Cluster.

The key is being requested under the Citi/EMC ELA "ELAXxxxxxx".

The details you need to generate the key are included below:

Installation ID:	356293a11a029b0e
Maximum capacity:	100TB raw (50TB usable)
Enterprise features:	Enabled
Usage time:	Permanent

Thanks

Eric Krejci

One can also directly generate a key oneself without raising an SR. See EMC document “ELA & VSLA LAC licensing Activation, Reporting and Downloads Help” for information on how to do this.

Citi/EMC ELA: You will need to obtain and substitute in the actual Citi/EMC ELA number from your Citi storage management organization as this commercially sensitive number cannot be put into this widely available document.

Installation ID: You can determine your ScaleIO cluster Installation ID by running the following command on your newly installed cluster.

```
# scli --query_license
For example:
# scli --mdm_ip <your-primary-MDM> --query_license
```

```
Installation ID: 798644b514008d01
SWID:
Maximum capacity: Unlimited
Usage time left: 10 days (Initial License)
Obfuscation feature is enabled
Snapshots feature is enabled
QoS feature is enabled
The system was activated 0 days ago
```

Note, that as the license is locked to the cluster Installation ID, you cannot request a permanent license prior to installing the software and configuring your new cluster. ***Note that if you reinstall the software, you will end up with a different license ID. Hence, you should not request a permanent licence until you are happy with your cluster installation.*** Note that if you upgrade an existing cluster (e.g. from R1.31 to R1.40), then you will retain your installation ID, and will not need a new license.

Maximum capacity: ScaleIO is licensed by raw capacity. As all data is mirrored, the usable capacity is half the raw capacity. It is recommended that you quote both requirements in the SR to avoid any confusion.

Once EMC has mailed you your license file. It can then be applied to the cluster with the following command:

```
# scli --set_license --license_file /tmp/798644b514008d01_2376166_12-Nov-2014.lic
Successfully set a new license
```

where 798644b514008d01_2376166_12-Nov-2014.lic is the license file provided by EMC.

You can confirm successful application of license by running the following command and checking that you know have "Unlimited" usage time left.

```
# scli --query_license
Installation ID: 798644b514008d01
SWID: HY1X0M2MBBFBCB
Maximum capacity: 4TB
Usage time left: Unlimited
Enterprise features: Enabled
The system was activated 0 days ago
```

17.1.1.2 Increasing license capacity for existing clusters

The capacity of an existing cluster can be expanded by opening an EMC Service Request (SR) on-line at <http://support.emc.com>. An example of the required SR text is shown below. The SR should only be raised by the “Emerging Data and Network Technologies” team.

This ticket is to request ADDITIONAL capacity for an existing ScaleIO Cluster.

The key is being requested under the Citi/EMC ELA "ELAXxxxxxx".

The details you need to generate the key are included below:

Installation (locking) ID:	356293a11a029b0e
Current Maximum capacity:	100TB raw (50TB usable)
ADDITIONAL capacity being requested:	50TB raw (25TB usable)
Enterprise features:	Enabled
Usage time:	Permanent

Thanks

Eric Krejci

17.1.1.3 Tracking ScaleIO license usage across Citigroup

See EMC document “**ELA & VSLA LAC licensing Activation, Reporting and Downloads Help**” for information on how to use the Powerlink licensing portal to:

- Get reports on current ScaleIO licensing usage against the Citi ELA. You will need to request the EMC account team to grant you access to run reports.

ScaleIO-specific reports can be accessed via the advanced search (under Activation> View Certificates > [Advanced Search] on the home screen), entering the ELA under the LAC and ScaleIO under the Product line.

[Home](#) [Entitlements](#) [Activation](#) [Reports](#) [Admin](#) [Help](#) [FAQs](#) [Log out](#)

Search Activation Certificates ?

Return

Use this page to search for machines for which to view and manage the activated items. To display additional criteria Certificates. For each machine, click View to view the certificate, or select an option from the Options list. [Click here](#) %indicates wildcard search

LAC: ELA

[Clear LACs] [Add more LACs]

%Machine name: []

[Clear List] [Enter List]

%Locking ID: []

[Clear List] [Enter List]

Software ID: []

Use these fields to search for machines based on the items that are installed. For products and families, there may be the find icon to search for and select a product. From each Family dropdown, select a product family that is installed equal to, less than, or greater than a specified value, then type the value into the accompanying field. If you specify machine to match the criteria.

Product line: ScaleIO

%Site name: []

%License Site Number: []

[What's this?]

Machine status: Active

This found all the ScaleIO licenses (two instances) associated with this ELA at the time of writing, as shown below.

Use the following criteria to search for machines that do NOT have selected products or families installed. To find a machine that does not have a specific product installed, type the product number into a Product # field, or the find icon to search for and select a product. To find a machine that does not have a specific family installed, select the family from a Family dropdown. There may be multiple product fields or family dropdowns. For example, if you provide two product numbers, only machines that have neither of those products installed match the criteria.

Product # []

Product # []

Family []

Family []

Search Certificates

Showing 1-2 of 2

	Options	Machine Name	Locking ID	Product Lines
View	Options	SCALEIO-356293a11a029b0e	20020e11a029b0e	ScaleIO
View	Options	ScaleIO	790044b51400b01	ScaleIO

Showing 1-2 of 2

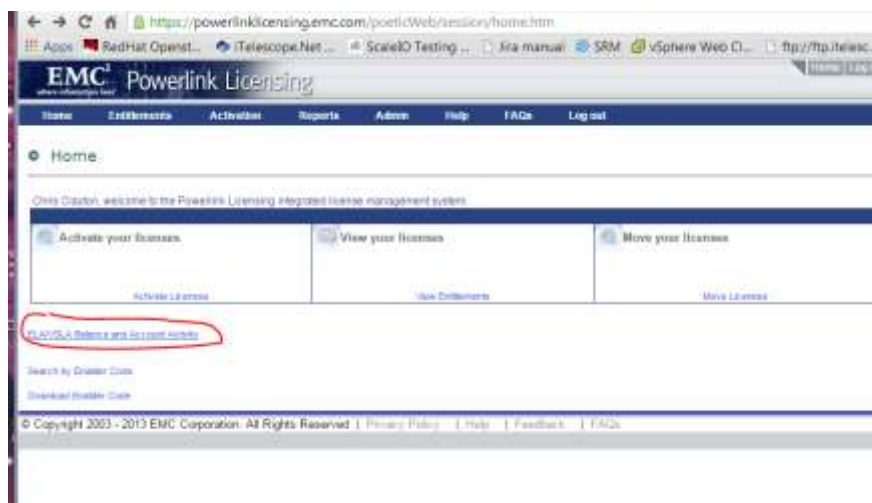
Also request from EMC that you be given access to the “ELA/SLA Balance and Account Activity” link on the home page. This red circled link below will not be visible to you unless your account has explicitly been granted access to this additional reporting tool.

GNAST-[ProductName]-#### [Design/Standard] Document

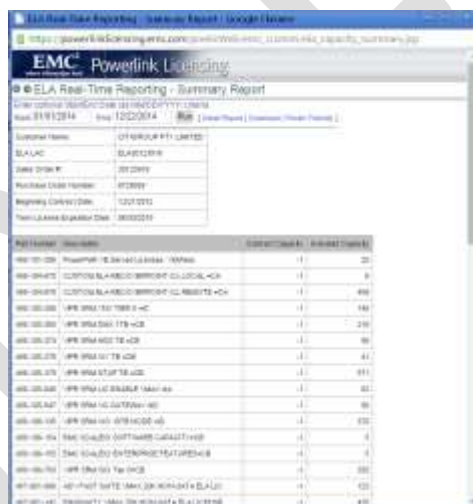
FOR INTERNAL USE ONLY

© Citigroup 2009

Page 67 of 116 GNAST Design and Configuration Guide Template



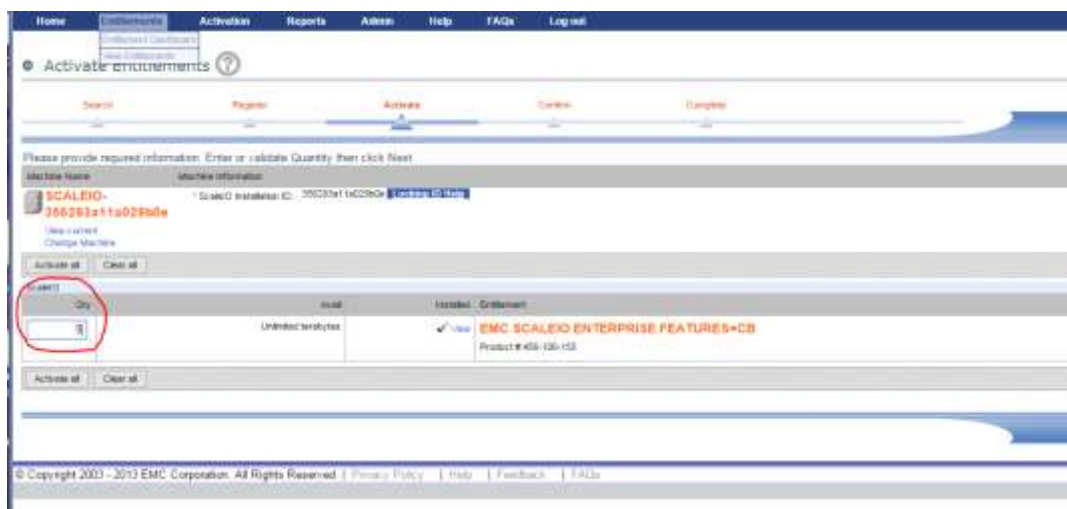
This view covers the entire ELA and can only filter on dates, but does show activated capacity.



License Name	License ID	License Type	License Status	License Capacity
EMC VPLEX	EMC VPLEX	EMC VPLEX	EMC VPLEX	EMC VPLEX
EMC VNX	EMC VNX	EMC VNX	EMC VNX	EMC VNX
EMC VNXE	EMC VNXE	EMC VNXE	EMC VNXE	EMC VNXE

For your information, the EMC document also describes how to use the EMC portal to:

- Get an initial license for a cluster. It is suggested that for the machine name, you use name of the form SCALEIO-<Cluster-Installation-ID> to provide a machine-independent “machine” for the cluster.
- Expand the capacity licensed for an existing. This can be achieved by searching machines with the installation ID (called the locking ID in EMC’s on-line license management system) and then requesting additional “Qty” for that server. In the example below, an additional 9TB of capacity is being requested for the cluster with Installation (locking) ID 356293a11a029b0e.



- Changing an Installation ID (should not be necessary unless you re-install software from scratch). This process is known as “rehosting” and you will need your “Machine Name” from the e-mail EMC sent to you with your license key. This will usually be “SCALEIO-
<installation-id>” for example “SCALEIO-798644b514008d01”.

Detailed information on how to use the EMC portal for the above activities is not included in the present document as it is subject to change by EMC without notice to Citi. Refer instead to the latest version of the EMC document “ELA & VSLA LAC licensing Activation, Reporting and Downloads Help”.

17.1.2 Troubleshooting processes

17.1.2.1 Host connectivity to its volumes

To aid troubleshooting, it is essential that hostnames are associated with IP addresses. This should already have been done when the SDC was approved access to the cluster.

```
# scli --rename_sdc --sdc_ip 10.116.19.214 --new_name nagtsnpod00204.nam.nsroot.net
Successfully renamed sdc 10.116.19.214 to nagtsnpod00204.nam.nsroot.net
```

If a client is unable to access its volume(s), then the following process should be followed:

- 1) Confirm that the IP address of the SDC is registered with the MDM and shows as connected. For example:

```
# scli --query_all_sdc | grep nagtsnpod00204
SDC ID: 8c0315cf00000000 Name: nagtsnpod00204 IP: 10.116.19.214 State: Connected GUID: CD1CB4D2-2F3F-4569-A6CD-32AAE71F5ECA Approved: yes

# scli --query_all_sdc | grep namgdse04
SDC ID: 43c7f50a00000001 Name: namgdse04 IP: 10.40.42.12 State: Disconnected GUID: EBFC8BAB-5E57-4DBD-8BE2-0CBB973FDD73

# scli --query_all_sdc | grep E8a7-IBMHS23-07
SDC ID: 6070cde500000000 Name: E8a7-IBMHS23-07 IP: 10.40.94.148 State: Connected GUID: 131BDB02-AB5A-46C0-A11F-A4D1F95D5C68 Approved: no
```

If there is no match, no SDC is registered for that IP address.

- 2) If the SDC not registered, confirm that that SA has installed the SDC software.
- 3) If not “Approved” you will need to request that the ScaleIO cluster manager approves your SDC to join the cluster. This is a one-off task
- 4) If the SDC is registered, but disconnected, there may be several possible causes.
 - a. During reboots the SDC shows disconnected, but should show as connected as soon as application host is back up.
 - b. There may be a network connectivity issue. Ask the SA to ensure that with the Primary or Secondary MDM can be pinged from the host.
 - c. Confirm that the correct MDM addresses were used by the SA when installing the SDC, and that BOTH MDM addresses are listed. If not all 6 addresses are listed, then the SDC

can become disconnected when an MDM failover occurs. In the example below, both the current primary, secondary MDMs and standby MDMs are correctly listed.

```
# cat /bin/emc/scaleio/drv_cfg.txt | grep mdm
mdm 10.116.27.213, 10.116.27.197, 10.116.27.214, 10.116.27.198, 10.116.19.213, 10.116.19.197
```

- d. If the incorrect (or only one) addresses were used, this can be modified on the application host with the command:

```
/opt/emc/scaleio/sdc/bin/drv_cfg --mod_mdm_ip
--ip <EXISTING_MDM_IP_ADDRESS>
--new_mdm_ip <NEW_MDM_IP_ADDRESSES> --file /bin/emc/scaleio/drv_cfg.txt
```

For example,

```
# /opt/emc/scaleio/sdc/bin/drv_cfg --mod_mdm_ip --ip 10.116.27.213 --new_mdm_ip 10.116.27.213,
10.116.27.197, 10.116.27.214, 10.116.27.198, 10.116.19.213, 10.116.19.197 --file /bin/emc/scaleio/drv_cfg.txt
Calling kernel module to modify MDM IP from (10.116.27.213) to (10.116.27.213,10.116.27.197,10.116.27.214,
10.116.27.198,10.116.19.213,10.116.19.197)
Successfully notified kernel module on new IP
Updating file /bin/emc/scaleio/drv_cfg.txt: MDM (10.116.27.213) to (10.116.27.213, 10.116.27.197, 10.116.27.214,
10.116.27.198, 10.116.19.213, 10.116.19.197)
```

NOTE: The --file parameter is required to ensure that the change is made persistent.

Restart the SDC instance

```
# service scini restart
Success unloading module
[ OK ]

Success loading module /bin/emc/scaleio/scini.ko
Found device special file /dev/scini
Loading MDMs from configuration file /bin/emc/scaleio/drv_cfg.txt
Successfully set UID ebfc8bab-5e57-4dbd-8be2-0cbb973fdd73
Successfully processed: mdm 10.116.27.213, 10.116.27.197, 10.116.27.214, 10.116.27.198, 10.116.19.213,
10.116.19.197
Executing boot_wait command. Waiting for 12 seconds
Successfully completed the boot_wait operation
Success configuring module
[ OK ]
```

Note: The parameter --add_mdm parameter should not be used, as this is for adding a second (separate) cluster to the SDC.

- e. As a last resort, the SDC package can be re-installed from scratch. The example below is for Linux.

```
# yum remove EMC-ScaleIO-sdc-1.30-426.0.el6

# MDM_IP=10.116.27.213,10.116.27.197,10.116.27.214,10.116.27.198,10.116.19.213,10.116.19.197 yum install
EMC-ScaleIO-sdc.x86_64
Updating file /bin/emc/scaleio/drv_cfg.txt with MDM (10.116.27.213, 10.116.27.197, 10.116.27.214, 10.116.27.198,
10.116.19.213, 10.116.19.197)
Success loading module /bin/emc/scaleio/scini.ko
```

```
Found device special file /dev/scini
Loading MDMs from configuration file /bin/emc/scaleio/drv_cfg.txt
Successfully set UID ebfc8bab-5e57-4dbd-8be2-0cbb973fdd73
Successfully processed: mdm 10.116.27.213, 10.116.27.197, 10.116.27.214, 10.116.27.198, 10.116.19.213,
10.116.19.197
Executing boot_wait command. Waiting for 12 seconds
Successfully completed the boot_wait operation
Success configuring module
[ OK ]
```

```
# /opt/emc/scaleio/sdc/bin/drv_cfg --query_vols
Retrieved 1 volume(s)
VOL-ID 3ab6c00f00000000 MDM-ID 5e19d49223c14dd5
```

- 5) If SDC is registered & connected, then confirm that devices are mapped to the correct IP address.

- a. Confirm that volumes are actually mapped to this SDC

```
# for vol in `scli --query_all_volumes| grep Volume| awk '{print $5}'`; do scli --query_volume --volume_name
$vol| egrep -e 'Volume ID:|SDC ID:|'; done
>> Volume ID: 3ab6c00f00000000 Name: scott01 Thick-provisioned Size: 32.0 GB (32768 MB)
    SDC ID: 43c7f50a00000001 IP: 10.40.42.12 Name: namgdse04.nam.nsroot.net
>> Volume ID: 3ab6c01000000001 Name: scott02 Thick-provisioned Size: 32.0 GB (32768 MB)
    SDC ID: 43c7f50a00000001 IP: 10.40.42.12 Name: namgdse04.nam.nsroot.net
>> Volume ID: 3ab6e71f00000002 Name: esxvol1 Thick-provisioned Size: 104.0 GB (106496 MB)
>> Volume ID: 3ab6e72000000003 Name: esxvol4 Thin-provisioned Size: 104.0 GB (106496 MB)
```

- b. Ask the SA to query all the active known volumes and send result to compare with mapping information above.

```
# /opt/emc/scaleio/sdc/bin/drv_cfg --query_vols
Retrieved 0 volume(s)
```

- c. If no volumes are visible, ask the SA to rescan for devices manually and query again.

```
On Linux:
# /opt/emc/scaleio/sdc/bin/drv_cfg --rescan
Calling kernel module to refresh MDM configuration information
Successfully completed the rescan operation
```

```
# /opt/emc/scaleio/sdc/bin/drv_cfg --query_vols
Retrieved 1 volume(s)
VOL-ID 3ab6c00f00000000 MDM-ID 5e19d49223c14dd5
```

```
On ESX:
Rescan using the vSphere GUI.
```

17.1.2.1.1 SDCs with multiple-interfaces

In environments where an SDC has multiple interfaces, the IP address shown below is the IP address that the SDC first communicated to the MDM with after the SDC software was installed.

If that interface is later unavailable, and the SDC is communicating with the MDM over a different interface (e.g. 10.116.19.198), then the `--query_all_sdc` output will correctly show that the state is Connected, but still show the original IP address associated with this host, not the one currently being used to communicate with the MDM. This is important to note when troubleshooting connectivity issues.

```
# scli --query_all_sdc | grep nagtsnpod00204
```

```
SDC ID: 8c0315cf00000000 Name: nagtsnpod00204 IP: 10.116.19.214 State: Connected GUID: CD1CB4D2-2F3F-4569-A6CD-32AAE71F5ECA Approved: yes
```

Note also, that one cannot provision storage to an IP address that is not currently available. In the situation where this is attempted, you will get the following error:

```
# scli --map_volume_to_sdc --volume_name cac01 --sdc_ip 10.116.19.214
```

```
Error: MDM failed command. Status: Could not find the SDC
```

In this example, 10.116.19.214 is down and interface 10.116.19.198 is being used. This issue should be avoided by always provisioning to the hostname.

```
# scli --map_volume_to_sdc --volume_name cac01 --sdc_name nagtsnpod00204
```

```
Successfully mapped volume cac01 to SDC nagtsnpod00204.
```

17.1.2.2 Associating ScaleIO volume names with Linux device names

Occasionally it may be necessary when troubleshooting to confirm the mapping between a Linux device and a ScaleIO volume name. This cannot be done wholly on the SDC.

On SDC, determine the ID of the volume associated with the linux device(s)

```
# ls -l /dev/disk/by-id/ | grep scini
```

```
lrwxrwxrwx 1 root root 12 Oct  8 16:56 emc-vol-5e19d49223c14dd5-3ab6c00f00000000 -> ../../scinia
```

The “5e19d49223c14dd5” component of the device ID is the cluster ID. The “3ab6c00f00000000” component is the volume ID. Then on the MDM, determine the volume name associated with the above volume ID.

```
# scli --query_all_volumes | grep 3ab6c00f00000000
```

Volume ID: 3ab6c00f00000000 **Name: scott01** Size: 32.0 GB (32768 MB) Mapped to 1 SDC Thick-provisioned

Or use the following alternate command:

```
# scli --query_properties --object_type VOLUME --object_id 3ab6c00f00000000 --properties NAME
```

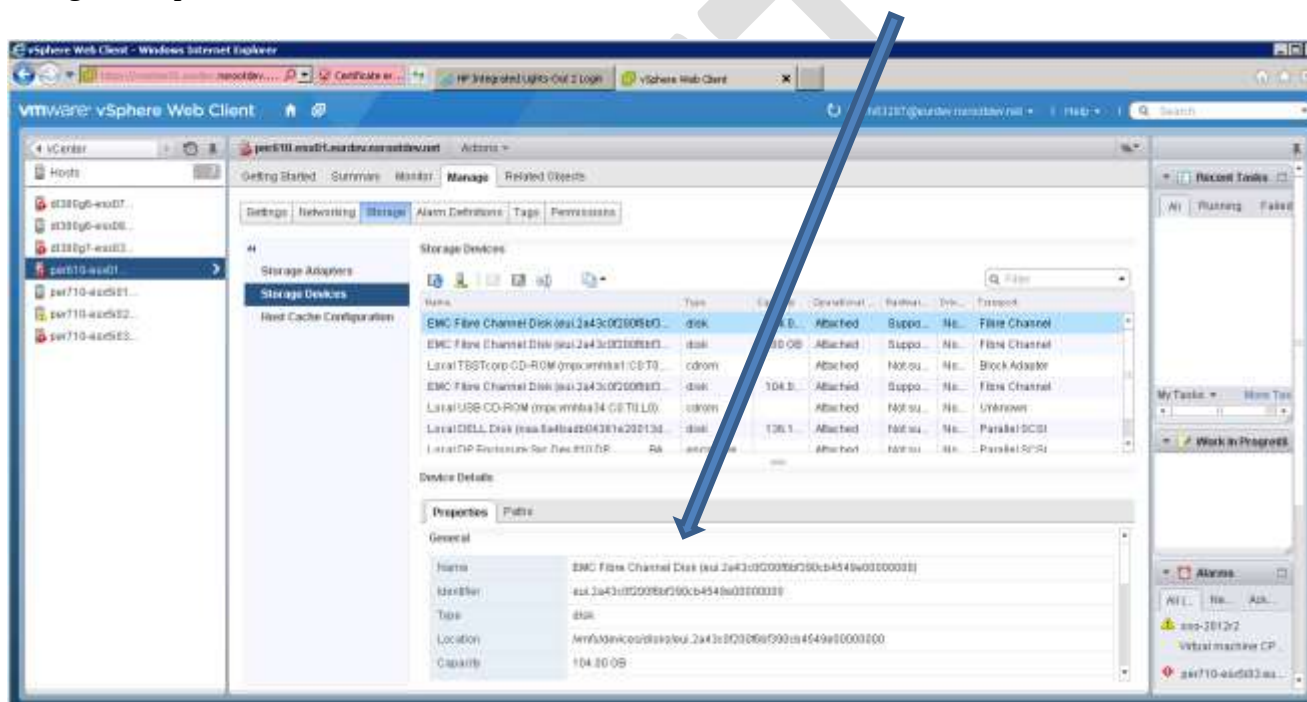
VOLUME 3ab6c00f00000000:

NAME **scott01**

Hence, we can confirm that volume scott01 is appearing as /dev/scinia on this Linux host.

17.1.2.3 Associating ScaleIO volume names with ESXi device names

Using the Vsphere Web Client, the ScaleIO volume ID can be determined:



In this case this is:

eui.**2a43c0f200f6bf390cb4549e00000000**

The first part of this ID is the cluster ID (shown in red). The second part is the volume ID (green)

On the ScaleIO cluster, the following command lists all volumes and their mappings to SDC clients:

```
# for vol in `scli --query_all_volumes | grep Volume | awk '{print $5}'; do scli --query_volume --volume_name $vol | egrep -e 'Volume ID:|SDC ID:'; done
```

...

```
>> Volume ID: 0cb4549e00000000 Name: esxvol1 Thick-provisioned Size: 104.0 GB (106496 MB)
    SDC ID: 9fc66e24000000005 IP: 10.154.219.126 Name: N/A
    SDC ID: 9fc66e23000000004 IP: 10.154.219.172 Name: N/A
```

Hence the volume name on the ScaleIO cluster is “esxvol1” in this case.

17.1.2.4 Logging ScaleIO tickets with EMC

If you need to log a ticket with EMC regarding ScaleIO, this should be done in the usual manner via telephone or support.emc.com. Note, however, the following additional points:

- 1) If your ticket is relating to a production-affecting issue, then state in your ticket that you want “immediate L3 escalation”. At the time of writing, much of the support for ScaleIO is still provided by the high-level team in Israel, so this often happens anyway, but you want to avoid getting delayed too long by a lower-skilled individual.
- 2) If it is a serious issue and you are not getting a satisfactory response, you can e-mail Alex.Khagar@emc.com who heads up the L3 support team.
- 3) If you are asked to collect logs, please do as this is pretty essential for EMC to troubleshoot the situation as they do not have dial-in access, as they would have on a traditional storage array such as VMAX. The log file collection process is described in the next section.
- 4) As EMC have no dial-in access, they can gain access to the system (if necessary) by requesting you join an EMC-hosted Webex session, with which you can provide them with access to your ScaleIO cluster whilst you watch what they are doing.

17.1.2.5 Log file collection

If you need to collect logs to send to EMC when opening a ticket, you should run the following undocumented command as root on each MDM server for which logs are needed:

```
# /opt/emc/scaleio/mdm/diag/get_info.sh
```

This will produce a compressed tar file of the server’s ScaleIO log files.

If you need to collect information on a tie-breaker, run the following as root:

```
# /opt/emc/scaleio/tb/diag/get_info.sh
```

If you need to collect information on a server that is only an SDS, run the following as root:

```
# /opt/emc/scaleio/sds/diag/get_info.sh
```

The primary MDM also has an events database that can be queried as root with the following command:

```
# /opt/emc/scaleio/mdm/bin/showevents.py
```

Type

```
# /opt/emc/scaleio/mdm/bin/showevents.py -h
```

to get help on using this useful script (e.g. filtering what output is selected based on time period or severity)

17.1.3 Miscellaneous Cluster maintenance

The following sections will discuss items that may have to take place as routine troubleshooting or maintenance in the ScaleIO cluster.

17.1.3.1 Disk controller cache – HP

The write cache on the HP controller should be enabled. To check if you have FBWC installed, run the following command:

```
# hpssacli ctrl all show status
```

```
Smart Array P420 in Slot 2
  Controller Status: OK
  Cache Status: OK
  Battery/Capacitor Status: OK
```

or with more detail

```
# hpssacli controller slot=2 show
```

and check that you have:

```
Cache Status: OK
Cache Ratio: 25% Read / 75% Write
Drive Write Cache: Disabled
No-Battery Write Cache: Disabled
Cache Backup Power Source: Capacitors
Battery/Capacitor Count: 1
Battery/Capacitor Status: OK
```

“Cache Status: OK” indicates that the controller cache is enabled. If you see: “Cache Status: Not Configured”, the controller cache is not enabled. To enable the controller cache run:

```
hpssacli ctrl slot=2 ld all modify arrayaccelerator=enable
```

The exact “Cache Ratio” ratio is not critical, but should be predominantly write as ScaleIO already has a large read cache. A setting of 25% Read / 75% Write is recommended.

```
# hpssacli controller slot=2 modify cr=25/75
```

You may also notice two options:

- Drive Write Cache
- No-Battery Write Cache

which must both be set to disabled, as they risk data loss if enabledⁱ². Do **not** run the command

```
hpssacli controller slot=2 modify dwc=enable
```

at any time as this enabled the disk write cache rather than the controller write cache.

The hpssacli for RHEL6 is part of the RHEL6 SOE from end of March 2015.

For more details on using hpssacli, see

https://catecollaboration.citigroup.net/domains/platstor/distsrvrhdw/X86Documents/RFP_Smart_Storage_Administrator_CLI_Deploy_Guide.pdf

17.1.3.2 Identifying Disks in ScaleIO

There are instances where a reboot of the ScaleIO server can result in drive mappings changing e.g. a disk which was /dev/sdc could reappear as /dev/sdd. For this reason it is advised in section 9.4.4 to use the disk scsi-ID for the “disk name” which is persistent. To view a disk’s mapping:

```
# scli --query_sds --sds_name sds-01
```

```
...
```

```
1: Name: c10473 Path: /dev/sdb Original-path: /dev/sdb ID: 6e53c4e700000000
```

The name is the device name you have assigned and will appear on the GUI. The path shows the current path to the device. The Original-path shows the path used when the device was first added.

17.1.3.3 Handling a disk failure in SDS node - HP

This section provides the end to end process to replace a failed disk in ScaleIO:

² **Drive Write Cache:** If enable means the array will use the onboard cache on the hard drive. It will write the data here first and then to the disk. If the server is not protected via UPS and loses power, data in the cache will be lost and cause data corruption on the hard drive.

No-Battery Write Cache: If you have no battery installed or it is failed you can enable this feature to use the on-board cache of the raid controller. However, if the server is not protected via UPS and loses power, data in the cache will be lost and cause data corruption on the hard drive.

In the event that a disk fails, ScaleIO will automatically alert that an SDS device is offline via an SNMP alert. This will also show in /var/log/messages. To view the failed disk in ScaleIO:

```
# for sds in `scli --query_all_sds | grep "SDS ID" | awk '{print $5}'`; do scli --query_sds --sds_name $sds |
egrep 'SDS|Path|State'; done
...
SDS 5f5330ca00000001 Name: sds-02
  1: 192.168.160.2 Role: SDS Only
    1: Name: sdb Path: /dev/sdb Original-path: /dev/sdb ID: 6e53c4ea00010000
      Storage Pool: domain1-sp1, Capacity: 135 GB Error-fixes: 0 State: Error
    2: Name: sdc Path: /dev/sdc Original-path: /dev/sdc ID: 6e53c4eb00010001
      Storage Pool: domain1-sp1, Capacity: 135 GB Error-fixes: 0 State: Normal
```

Use the HP Smart Array controller CLI to find the particular drive which failed:

```
# hpssacli
=> ctrl slot=2 show config
...
  logicaldrive 5 (136.7 GB, RAID 0, Failed)
    physicaldrive 2I:1:6 (port 2I:box 1:bay 6, SAS, 146 GB, Failed)
```

Flash the drive to ensure the hardware engineer replaces the correct drive. In this example logical drive 5 has failed

```
# => ctrl slot=2 ld 5 modify led=on
```

Replace the failed drive. Then stop flashing the drive and quite out of the Smart Array controller

```
# => ctrl slot=2 ld 5 modify led=off
# => quit
```

Re-enable the drive and verify it is available again:

```
# => ctrl slot=2 ld 5 modify reenable
# => ctrl slot=2 show config
...
  logicaldrive 5 (136.7 GB, RAID 0, OK)
    physicaldrive 2I:1:6 (port 2I:box 1:bay 6, SAS, 146 GB, OK)
...
# => quit
```

Remove the SDS device from ScaleIO (it will still show as failed at this point):

```
# scli --remove_sds_device --sds_name sds-02 --device_path /dev/sdb
SDS Device sds: sds-02, /dev/sdb is being removed asynchronously
```

Add the replaced drive to ScaleIO.

```
# scli --add_sds_device --sds_name sds-02 --device_path /dev/sdb --storage_pool_name domain1-sp1 --  
device_name sdb --no_test
```

Successfully added device sdb to SDS. New device ID: 6e53ebf900000001

Note: On occasion this command has been seen to fail the driver despite the fact it appears to be visible to the O.S. and the following error is reported:

```
# scli --add_sds_device --sds_name sds-02 --device_name=sdc --device_path=/dev/sdc --  
storage_pool_name=domain1-sp1
```

Error: MDM failed command. Status: Could not open SDS device

If this is encountered it will be necessary to reboot the node to re-add the SDS. After the reboot:

```
# scli --add_sds_device --sds_name sds-02 --device_name=sdc --device_path=/dev/sdc --  
storage_pool_name=domain1-sp1
```

Successfully added device sdc to SDS. New device ID: 6e54130a00010001

Note: Server Rebooted Prior to Disk Replacement: If after a disk drive has failed, the server is rebooted at any time prior to disk replacement per the above procedure, this will modify the underlying disk mappings on the server. For example, let's say drive "sdd" failed. If the server is rebooted, then what was drive "sde" will now appear as drive "sdd". Similarly all drive names below "sdd" in the server will have moved up one letter. This is not an issue for ScaleIO which actually puts data on a physical drive and scans the drives on reboot and so will correctly identify the drive. However care should be taken by the SA when swapping out drives. This is the reason the preferred way to reference a drive in ScaleIO is using the persistent ID of the drive. For example:

```
# ls -l /dev/disk/by-id/ | grep sdb | grep scsi | awk '{print $9}' | cut -c 33-  
2161c0
```

The last 6 letters are used for the drive name in ScaleIO.

17.1.3.4 Replacing a Failed Disk Controller in SDS node - HP

This section provides an overview of the end-to-end process to replace a failed HP disk controller in ScaleIO.

In the event that a disk controller fails, ScaleIO will automatically alert that SDS devices on the affected SDS are offline via an SNMP alert. There will also appear an entry in the ScaleIO logs similar to the following:

```
# /opt/emc/scaleio/mdm/bin/showevents.py
```

```
.....  
3414 2015-05-21 16:00:57.835 SDS_DEV_ERROR_REPORT ERROR Device error reported on SDS:  
192.168.160.2, Device: /dev/sdc.
```

This will also show in /var/log/messages. To confirm that ALL disks are affected (i.e. it is likely a disk controller issue) run the following command and confirm that ALL devices are in State Error:

```
# scli --query_sds --sds_name e8c4-dl360g7-01 | egrep 'SDS|Path|State'
SDS 8d50d10600000000 Name: e8c4-dl360g7-01
  1: 192.168.150.50 Role: SDS Only
    1: Name: sdb Path: /dev/sdb Original-path: /dev/sdb ID: 6e53c4ea00010000
      Storage Pool: domain1-sp1, Capacity: 135 GB Error-fixes: 0 State: Error
    2: Name: sdc Path: /dev/sdc Original-path: /dev/sdc ID: 6e53c4eb00010001
      Storage Pool: domain1-sp1, Capacity: 135 GB Error-fixes: 0 State: Error
    3: Name: newsdd Path: /dev/sdd Original-path: /dev/sdd ID: 14ef190500000002
      Storage Pool: domain1-sp1, Capacity: 135 GB Error-fixes: 0 State: Error
etc
```

To confirm, use the HP Smart Array controller CLI to confirm that the disk controller is not O.K.

```
# hpssacli ctrl all show status

Smart Array P420 in Slot 2
Controller Status: <Something other than "OK">
```

Once it is confirmed that the disk controller is at fault, then this will need to be replaced. Note that the disk controller is not a hot-swap component. The server would need to be powered off and opened up to replace the controller.

Note that as the disk controller has a write cache, there will likely be IOs in the now unavailable cache that did not make their way down to disk. **Hence it is essential that the data in these disks is not used again, as it may not be invalid.** If the ScaleIO rebuild that followed the loss of the disk controller has completed, then the data on the disks will already have been marked as invalid by ScaleIO and will be used. **So it is essential that the errors on the SDS devices are not cleared until the ScaleIO rebuild to reprotect the data on the "failed" disks has completed. This is essential to ensure that potentially bad data on the disks is not reused after the disk controller is replaced.**

After the disk controller has been replaced, verify the disk drives are all available again. Depending on the exact fault with the disk controller, the HP engineering may or may not be able to restore any changes to the default setting that you may have made. The new controller will pull the drive configuration from the drives, but other settings that were changed would need to be changed again on the new controller (e.g. if you changed the cache ratio away from the default settings).

```
# hpssacli ctrl slot=2 show config
```

All physical and logical devices should show as "OK".

If the ScaleIO rebuild has completed, and these devices are all marked as being empty in the ScaleIO GUI, **then (and only then)** the SDS device error states can be cleared to allow them to be made available to the cluster again.

IMPORTANT:

Do not clear the error states when the rebuild is still in progress.

Error states can then be cleared with the command:

```
# scli --clear_sds_device_error --sds_name <SDS-NAME> --clear_all
```

17.1.3.5 Replacing a Failed SDS

If an SDS fails requiring a rebuild of the servers, then it needs to be removed from the ScaleIO cluster and then added again as new. To remove a failed SDS from the cluster (in this case nagtsnpod00204)

```
# scli --remove_sds --sds_name nagtsnpod00204
```

After the server has been rebuilt, add it back to the cluster along with any required devices. For example:

```
# scli --add_sds --sds_ip 10.116.19.246,10.116.19.230,10.116.19.214,10.116.19.198 --sds_ip_role  
sds_only,sds_only,sdc_only,sdc_only --device_name 604197 --device_path /dev/sdb --sds_name  
nagtsnpod00204 --protection_domain_name domain1 --storage_pool_name domain1-spsas --fault_set_name  
domain1-fs4 --rmcache_size_mb 10240  
Successfully created SDS nagtsnpod00204. Object ID 2ac3c6a900000003
```

Finally add in any additional devices into other storage pools if required.

17.1.3.6 Adding Storage Capacity to a Cluster

To add extra capacity to a cluster, first add any new servers to the cluster (unless using disks in an existing SDS). Adding an SDS is described in section 17.1.3.7.

Next add the devices to the storage pool required. The example below adds a device /dev/sdw to storage pool domain1-spsd. The device name to be used corresponds to the last 8chars of the unique id of the disk e.g.:

```
# ls -l /dev/disk/by-id | grep sdb | grep scsi | awk '{print $9}' | cut -c 33-
```

2161c0

Now add the drive

```
# scli --add_sds_device --sds_name nagtsnpod00204 --device_path /dev/sdz --storage_pool_name domain1-  
spssd --device_name 2161c0
```

Successfully added device b00058 to SDS. New device ID: f5509f8100030002

17.1.3.7 Adding Nodes to a Cluster

First build the server per SOE build standards and install the ScaleIO packages. This is described in sections 9.3.2, 9.3.3 and 9.4.4. At this point the SDS node is ready to be added to the cluster.

To do so

add it back to the cluster The following is an example, adding the SDS along with a single disk.

Note that you will check to validate:

- The domain the SDS is to be added to
- The storage pool the SDS is to be added to
- The device name and path to be used.

```
# scli --add_sds --sds_ip 10.116.19.246,10.116.19.230,10.116.19.214,10.116.19.198 --sds_ip_role  
sds_only,sds_only,sdc_only,,sdc_only --device_name 604197 --device_path /dev/sdb --sds_name  
nagtsnpod00204 --protection_domain_name domain1 --storage_pool_name domain1-spsas --fault_set_name  
domain1-fs4 --rmcache_size_mb 10240
```

Successfully created SDS nagtsnpod00204. Object ID 2ac3c6a900000003

Next add in any additional drives to the relevant storage pools (see 17.1.3.6)

Finally harden the SDS (this command actually ensures all sds servers are write the journal files to disk so improving the time it takes to re-protect data if a server were to fail)

```
# scli --set_drl_properties --hardened --set_all
```

Successfully set data-region-logging properties

17.1.3.8 Shutting down a cluster for a power down

In the rare event that it is necessary to shut down a cluster (e.g. a power down affecting all fault sets) then the following procedure should be followed to deactivate protection domain(s).

Making a protection domain inactive blocks any rebuild and rebalancing activities, quiesces all application IO and disables access to volumes. It then move the Dirty Region Log (DRL) mode of all SDSs to “harden”, in preparation for reboot of the servers.

Use the dashboard to confirm that all storage is currently protected i.e. no storage in degraded (orange) state.

Deactivate the protection domain:

```
# scli --inactivate_protection_domain --protection_domain_name <protection-domain-name>
```

For example:

```
# scli --inactivate_protection_domain --protection_domain_name domain1
```

All application IOs will not be serviced. Press 'y' to confirm that you want to proceed ?

Protection Domain domain1 is being shutdown asynchronously

The optional flag `--force_inactivate` may need to be added to the above command if there is some ongoing activity that prevents the deactivation occurring.

Once the protection domain(s) is deactivated, the cluster can be shutdown. The following order is recommended:

1. SDS-only nodes
2. Tie-breaker node
3. Secondary MDM (cluster will now hang as no longer quorate)
4. Primary MDM

Once the server has been powered back on, the protection domain(s) can be activated with the command:

```
# scli --activate_protection_domain --protection_domain_name <protection-domain-name>
```

Note If a cluster component has failed during a power down such as an SDS or a disk, then you will be unable to re-activate the domain as ScaleIO notes there is a component not available. In this scenario if is required to force the re-activation so that the required repair can be undertaken. For example:

```
# scli --activate_protection_domain --protection_domain_name domain1 --force_activate
```

Some elements in the system may not be healthy. Press 'y' to confirm that you want to proceed ? **y**

Protection Domain domain1 is active

17.1.3.9 Temporarily removing an SDS from a cluster for maintenance

It will sometimes be necessary to take a node running SDS software out of service for an extended period (i.e. not just for a reboot).

Activity	Remove SDS from Cluster?	Comments
Server reboot	No	Data will be degraded whilst the server

		reboots but then will quickly be fully protected once the SDS is on-line again.
Extended server maintenance	Yes	This covers cases where the SDS will not be immediately returning to the cluster.

An SDS can be taken offline by running the following command on the node to be taken out of service.

```
SDS-node# /opt/emc/scaleio/sds/bin/delete_service.sh
```

However, this approach will leave the cluster temporarily degraded whilst the other SDSs re-protect the data that was on the SDS just taken out of service.

A safer (but more complex) approach is to remove the SDS from the cluster configuration in a controlled manner. During this process, data on the SDS being taken down is replicated to different nodes, and at no time in the process is data protection degraded. However, the removal process is asynchronous and might take a long time.

The process is as follows:

Record the configuration of the MDM prior to removal. This will be needed when you re-add the SDS later, if you have not previously stored the command used to originally add this SDS to the cluster.

```
# scli --query_sds --sds_name sds-01 | more
SDS 5f5330c900000000 Name: sds-01
Protection Domain: 346b28d300000000, Name: domain1
Fault Set: d7ede25f00000000, Name: domain1-fs1
DRL mode: Hardened
IP information (total 2 IPs):
  1: 192.168.160.1   Role: SDS Only
  2: 10.154.223.197 Role: SDC Only
Port: 7072
RAM Read Cache information:
128.0 MB (131072 KB) total size
Cache is enabled
RAM Read Cache memory allocation state is SUCCESSFUL
Number of IO Buffers: 1

Device information (total 3 devices):
  1: Name: sdb Path: /dev/sdb Original-path: /dev/sdb ID: 6e53c4e700000000
    Storage Pool: domain1-sp1, Capacity: 135 GB Error-fixes: 0 State: Normal
  2: Name: sdc Path: /dev/sdc Original-path: /dev/sdc ID: 6e53ebf900000001
    Storage Pool: domain1-sp1, Capacity: 135 GB Error-fixes: 0 State: Normal
  3: Name: sdd Path: /dev/sdd Original-path: /dev/sdd ID: 6e53c4e900000002
    Storage Pool: domain1-sp1, Capacity: 135 GB Error-fixes: 0 State: Normal
```

Remove the SDS

```
# scli --remove_sds --sds_name <SDS-name>
```

For example

```
# scli --remove_sds --sds_name sds-01
```

This is an asynchronous operation and will run in the background. You can either track progress in the dashboard, or run `query_all_sds` periodically. You can also abort the removal using the `abort_remove_sds` command, if necessary.

Once the SDS is shown as disconnect, you can carry out the required host maintenance.

When the SDS needs to be re-added to the cluster, this is achieved by re-running the original `scli --add_sds` command you used to configure this SDS into the cluster. An example, is given below (re-formatted for to make more readable):

```
# scli --add_sds --sds_ip 10.59.72.74,10.59.73.74
--sds_ip_role sds_only,sdc_only
--device_name sdb,sdc,sdd,sde,sdf,sdg,sdh,sdi,sdj,sdk,sdl,sdm,sdn,sdo,sdp,sdq,sdr,sds,sdt,sdu
--device_path
/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg,/dev/sdh,/dev/sdi,/dev/sdj,/dev/sdk,/dev/sdl,/dev/sdm,/d
ev/sdn,/dev/sdo,/dev/sdp,/dev/sdq,/dev/sdr,/dev/sds,/dev/sdt,/dev/sdu
--sds_name sds-01
--protection_domain_name domain1
--storage_pool_name domain1-spas
--fault_set_name domain1-fs1
--rmcache_size_mb 16384
```

17.1.3.10 Accidental package deletion

ScaleIO packages should never be removed from a server that is part of a cluster. In the event of accidental deletion of an SDS or MDM package by an SA, it may be still be possible to recover the cluster. If accidental deletion should occur, do not attempt to re-install the removed package and configure the node back into the cluster. Instead:

- Reach out to CATE who will be able to help assess the impact of the accidental package deletion and give advice on the best resolution path.
- Log a ticket with EMC. One of their Level 3 support staff may be able to reconfigure the node back into the cluster via a non-standard process, depending on which package has been removed.

17.1.3.11 Rebooting SDS servers

Sometimes it will be necessary to reboot an SDS server after OS patching.

It is essential that you always check the ScaleIO dashboard to ensure that there is no rebuild activity taking place or any degraded storage (orange in dashboard) for the pool supported by that server before rebooting an SDS node. Failure to do so will likely result in data unavailability.

17.1.4 ScaleIO canned performance reports available with SRM

The following applies to the SRM ScaleIO SolutionsPack 1.1.3.

The ScaleIO canned reports in the SRM Reports Library only report on the following metrics:

- Read and Write Throughput (what is usually referred to as IOPS)
- Read and Write Bandwidth (MB/s)

The metrics are available both as trends with time, or as average, minimum or maximum over a specified period.

The metrics are available at the following levels in the SRM Reports Library under “EMC ScaleIO”:

- Under “ScaleIO System Inventory”
 - ScaleIO drive
 - ScaleIO volume
 - ScaleIO storage pool
 - ScaleIO protection domain
 - ScaleIO Data Server (SDS)
- Under “Performance”
 - ScaleIO Cluster Throughput data
 - ScaleIO Cluster Bandwidth data
- Note that SDC-level data is not currently available.
- Rebuild (forwards and backwards) and rebalance throughput is also available for individual SDSes, pools, protection domain and clusters.
- Throughput and bandwidth information is also available as a summation across all ScaleIO clusters that SRM is collecting data for, but this information is not considered to be of practical value for troubleshooting.
- The most useful reports available for ScaleIO in SRM are the individual storage pools and volumes, and are discussed in more detail below.

17.1.4.1 Pool and volume historical performance graphs

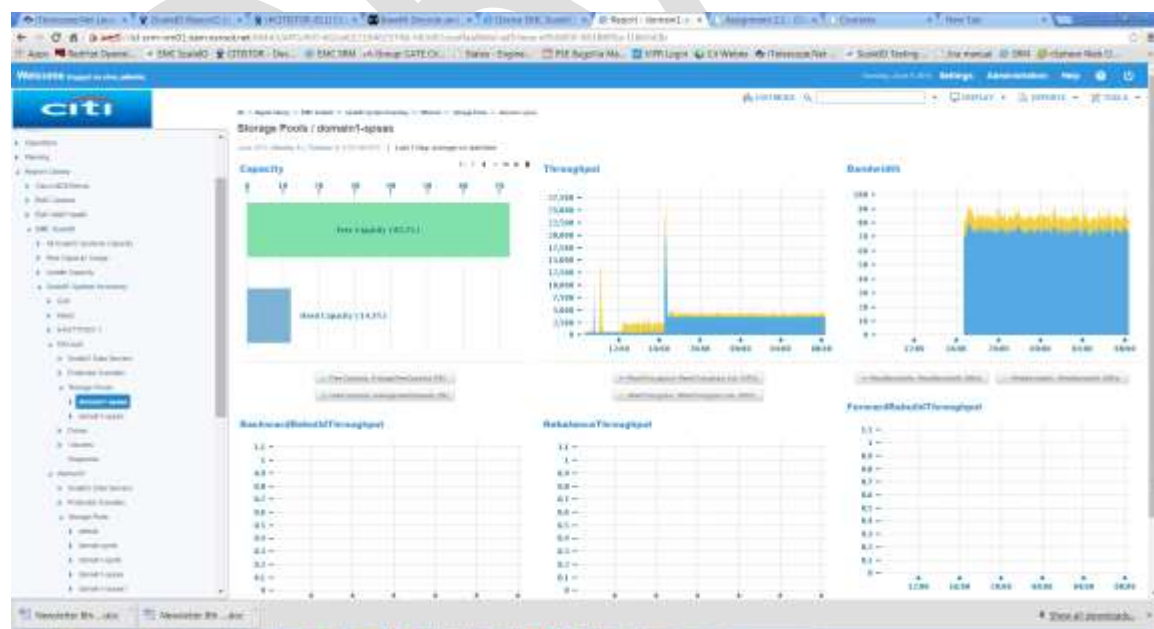
The locations of **graphical historical performance trend** reports in the SRM navigator for **individual pools or volumes** respectively are:

All>>Report Library>>EMC ScaleIO>>ScaleIO System Inventory>>*cluster-name*>>Storage Pools>>*pool-name*

All>>Report Library>>EMC ScaleIO>>ScaleIO System Inventory>>*cluster-name*>>Volumes>>*volume-name*

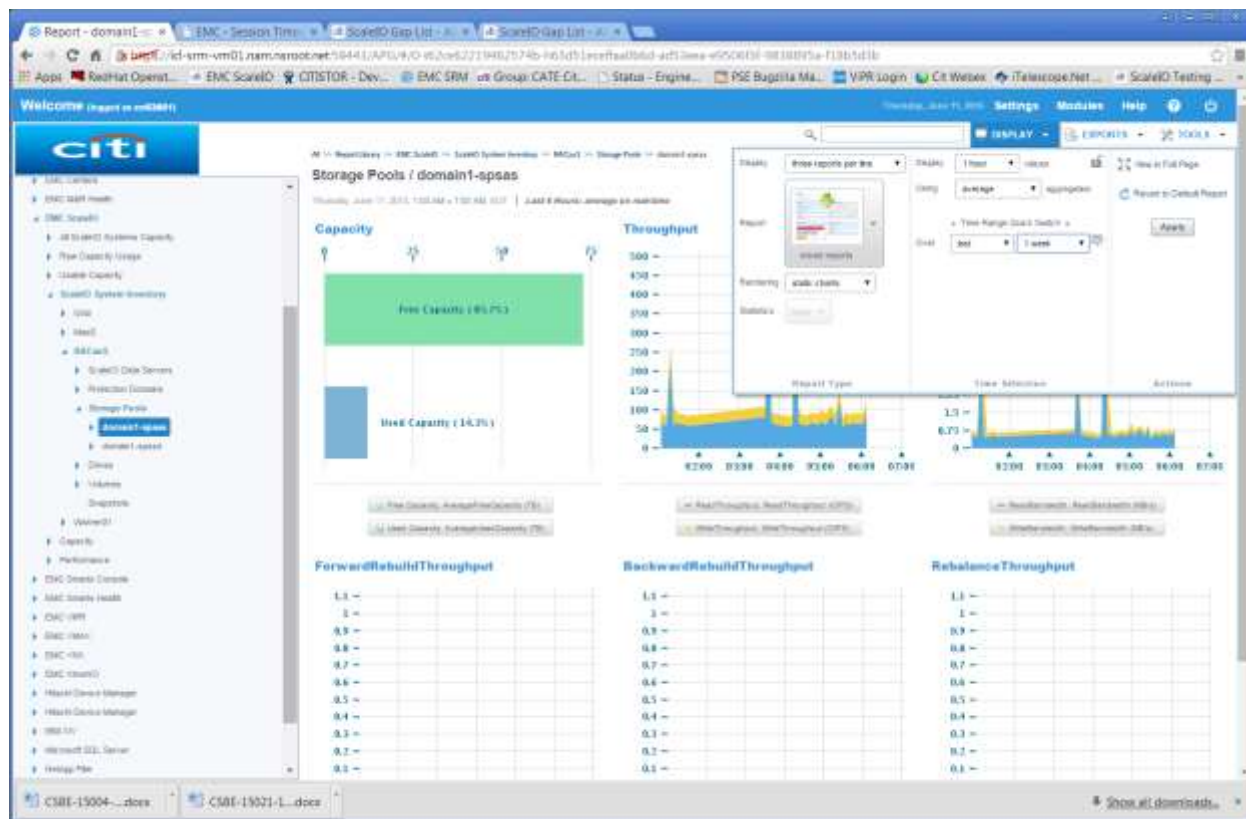


Example of a “ScaleIO System Inventory” single volume performance report



Example of a “ScaleIO System Inventory” single storage pool report

The reporting period for the graphical historical performance trend reports can be changed by clicking on the DISPLAY drop down menu at the top right of the SRM reports pane.



Modifying the Time Range for a performance report

Note also that on throughput and bandwidth graphs, you can also zoom in by using your mouse to define a subset time period from the currently visible data. The labelled buttons below a graph can also be pressed to remove that component from the plot (e.g. to remove either the read or write components in a graph in the example above).

The locations of **tabular** reports in the SRM navigator for **all pools or volumes** in a cluster are:

All>>Report Library>>EMC ScaleIO>>ScaleIO System Inventory>>*cluster-name*>>Storage Pools

All>>Report Library>>EMC ScaleIO>>ScaleIO System Inventory>>*cluster-name*>>Volumes



Clicking on a specific pool or a volume in a tabular report will bring up the corresponding **graphical historical performance trend** report for that volume (see earlier examples).

- **With the current release of the SRM ScaleIO SolutionsPack, the various free vs used % capacity bar charts for pools do not give accurate usage % information and must not be used.**
- The top level “EMC ScaleIO” report shows a variety of reports that sum cover all ScaleIO clusters for which there is data collected by your SRM instance. A few of these may be of some use for determining high-level statistics for management. However, as per the note above, the various free vs used % capacity bar charts for pools do not give accurate usage % information and must not be used.



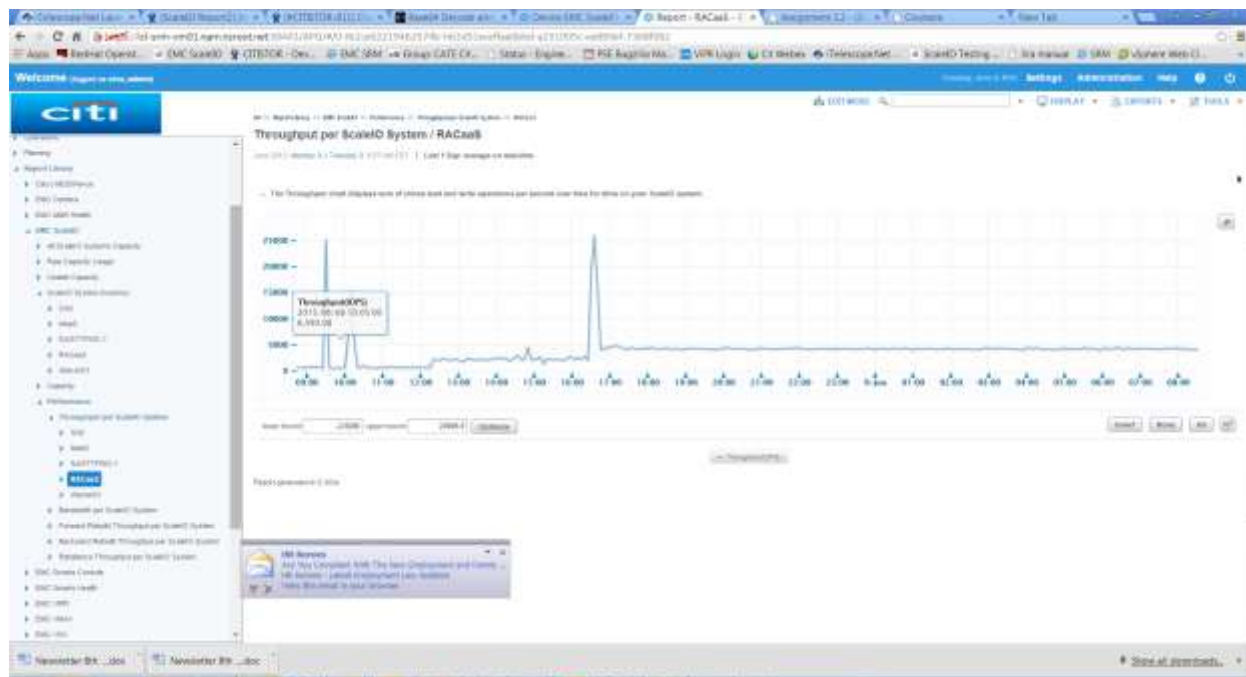
- Example of top-level “EMC ScaleIO” report
- “ScaleIO System Inventory” report showing a summary of the system components. To get a cluster component list, and then drill down into the details of a specific component.



FOR INTERNAL USE ONLY

- There is no canned report that shows in a single view all storage pools in a given cluster (or protection domain) as separate entities. Instead, you can only get a summation of all the pools in a cluster (or protection domain) “performance” plot. For example, the following report shows showing overall cluster IOPS.

All>>>Report Library>>EMC ScaleIO>>Performance>>cluster-name



Example of “Performance” report showing overall cluster IOPS

17.2 Appendix B – Best Practices

17.2.1 MDM Nodes

Each ScaleIO cluster will be made up of a Primary, a Secondary MDM and a Standby MDM Nodes. Stand-By MDM nodes are discussed in the next subsection. The primary MDM, secondary MDM, standby MDM and tie-breaker should be in separate fault domains (e.g. separate racks) where possible.

17.2.2 Configuring automated MDM/TB replacement (Standby MDM)

17.2.2.1 Overview

As an R1.31 ScaleIO cluster has only two MDMs (primary and secondary), then if one of these servers were to fail for an extended period, then the cluster would be vulnerable to a failure of the one remaining MDM, rendering data inaccessible.

EMC support an automated process to detect the failure of and replace with a standby MDM, a primary or secondary MDM that fails for an extended period, in order to ensure that the cluster continues to have both a working primary and secondary MDM.

This solution is described in detail in the EMC Technical Note “AUTOMATIC MDM CLUSTER NODE REPLACEMENT”. Below we only cover Citi Best Practice and the process to setup automated MDM replacement is documented via an example.

Important note: In a ScaleIO cluster with just a Primary and a Secondary MDM (A & B respectively):

- If the primary MDM A fails, the secondary MDM B become primary (and vice versa) with the cluster remaining in “Degraded” state (Cluster State: Degraded).
- If the secondary MDM B fails, the primary MDM A continues as primary, with the cluster remaining in “Degraded” state (Cluster State: Degraded). The failed secondary MDM B is still marked as secondary but is unavailable.

In contrast, with automatic MDM replacement, with primary, secondary and standby MDMs A, B & C respectively.

- If the primary MDM A fails, the secondary MDM B become primary but then the standby MDM C is then configured as the new secondary MDM, with the former primary MDM A being removed from the cluster (strictly now the new standby MDM, once that server is available again). The cluster state returns to “Normal” as it has both live primary and secondary MDMs (B & C in this example).
- If the secondary MDM B fails, the primary MDM A continues as primary, but now the standby MDM C is then configured as the new secondary MDM, with the former secondary MDM B being removed from the cluster. The cluster state returns to “Normal” as it has both live primary and secondary MDMs (A & C in this example).

17.2.2.2 Citi Best Practice

- One (only) standby MDM should be configured for a total of 3 possible primary MDMs within the cluster (current primary, current secondary and the standby MDM).
- One (only) standby Tie-Breaker (TB) should also be configured in Tornado++ configurations. See Tornado++ Best Practice for guidance on which server to locate the standby TB.

- ***As any one of THREE MDMs may now be the primary MDM, ALL three MDM IP addresses (current primary, secondary and standby MDM) must be configured on the SDC. Failure to do so could result in the SDC being unable to communicate with the current primary MDM.***
- All SDCs must first be configured with all 3 addresses before automated MDM replacement is configured on the MDMs. This is to ensure that SDCs know about all possible primary MDM addresses.
- Where possible, the primary, secondary, standby MDMs and tie-breaker should all be in separate racks.
- To ensure that the standby MDM is not invoked unnecessarily (e.g. during a reboot of an MDM server) the following parameter should be used when running the `/mdm_failover_post_install.py` setup script:

```

○ --retry_timeout=30      [default is 5s]
○ --retry_count=20        [default is 10]

```

`retry_timeout * retry_count` (600s in the above example) is the time that will pass from the time the cluster is degraded until the `mdm_failover` script takes control and brings up the standby MDM. The actual operation of the script (once invoked) includes replacing the faulty MDM and cluster resync (in case the secondary is replaced) in just a few seconds.

- Citi MDMs run on SDS servers. SDS servers (and hence the MDM servers) have multiple network interfaces. An SDC can only support a maximum of 8 MDM IP addresses. Hence, to work around this limit, only the private network interfaces on the SDS/MDM servers (those set with the `sds_only` role) should be specified as the MDM IP addresses when running the `/mdm_failover_post_install.py` setup script.

17.2.2.3 Setting up automatic MDM replacement (only)

1. Identify the server that will be configured as the standby MDM. This should be in a separate rack to the existing Primary MDM, Secondary MDM and TB. The private/cluster network IP addresses should be used. In this example, the roles are as follows:
 - Primary MDM : 10.116.27.245, 10.116.27.229
 - Secondary MDM : 10.116.19.246, 10.116.27.230
 - Tie-Breaker : 10.116.19.245, 10.116.19.229
 - Standby MDM: 10.116.19.246, 10.116.19.230

For Tornado++ environments a Standby TB should also be configure (see next subsection).

2. Install the MDM package onto the standby MDM server, if not already installed. [See MDM configuration.](#)
3. On the primary, secondary and standby MDM. In this example, on each of .74, .75 & .77 (i.e. all MDMs – primary, secondary or standby respectively) run the same identical command:

On Primary MDM

```
/opt/emc/scaleio/mdm_failover/bin/mdm_failover_post_install.py --mdms_list
[10.116.27.245,10.116.27.229]+[10.116.27.246,10.116.27.230]+[10.116.19.246,10.116.19.230] --tbs_list
[10.116.19.245,10.116.19.229] --username admin --password London1 --retry_timeout=30 --retry_count=20
Found 3 MDMs' IPs
Found 1 Tie-Breakers' IPs
Successfully started mdm_failover as a service, to stop it use /opt/emc/scaleio/mdm_failover/bin/delete_service.sh
On Secondary MDM
```

On Secondary MDM

```
/opt/emc/scaleio/mdm_failover/bin/mdm_failover_post_install.py --mdms_list
[10.116.27.245,10.116.27.229]+[10.116.27.246,10.116.27.230]+[10.116.19.246,10.116.19.230] --tbs_list
[10.116.19.245,10.116.19.229] --username admin --password London1 --retry_timeout=30 --retry_count=20
Found 3 MDMs' IPs
Found 1 Tie-Breakers' IPs
Successfully started mdm_failover as a service, to stop it use /opt/emc/scaleio/mdm_failover/bin/delete_service.sh
On Secondary MDM
```

On Standby MDM

```
/opt/emc/scaleio/mdm_failover/bin/mdm_failover_post_install.py --mdms_list
[10.116.27.245,10.116.27.229]+[10.116.27.246,10.116.27.230]+[10.116.19.246,10.116.19.230] --tbs_list
[10.116.19.245,10.116.19.229] --username admin --password London1 --retry_timeout=30 --retry_count=20
Found 3 MDMs' IPs
Found 1 Tie-Breakers' IPs
Successfully started mdm_failover as a service, to stop it use /opt/emc/scaleio/mdm_failover/bin/delete_service.sh
On Secondary MDM
```

where is xxxxxx is the admin password for the cluster.

17.2.2.4 Setting up automatic MDM & TB replacement on Tornado++

For Tornado++ environments, a standby TB should also be configured. This should be in the same rack as the configured TB.

1. Identify the server that will be configured as the standby MDM. This should be in a separate rack to the existing Primary MDM, Secondary MDM and TB. The private/cluster network IP addresses should be used. In this example, the roles are as follows:
 - Primary MDM : 10.116.27.245, 10.116.27.229
 - Secondary MDM : 10.116.27.246, 10.116.27.230
 - Tie-Breaker : 10.116.19.245, 10.116.19.229

- Standby MDM: 10.116.19.246, 10.116.19.230
 - Standby TB: 10.116.19.247, 10.116.19.231
2. Install the MDM package onto the standby MDM server, if not already installed. See [MDM configuration](#). Install the TB package onto the standby TB server, if not already installed. See [TB configuration](#).
 3. On the primary, secondary and standby MDM. In this example, on each of .74, .75 & .77 (i.e. all MDMs – primary, secondary or standby respectively) run the same identical command:

On Primary MDM

```
/opt/emc/scaleio/mdm_failover/bin/mdm_failover_post_install.py --mdms_list
[10.116.27.245,10.116.27.229]+[10.116.27.246,10.116.27.230]+[10.116.19.246,10.116.19.230] --tbs_list
[10.116.19.245,10.116.19.229]+[10.116.19.247,10.116.19.231] --username admin --password London1 --
retry_timeout=30 --retry_count=20
Found 3 MDMs' IPs
Found 2 Tie-Breakers' IPs
Successfully started mdm_failover as a service, to stop it use /opt/emc/scaleio/mdm_failover/bin/delete_service.sh
On Secondary MDM
```

On Secondary MDM

```
/opt/emc/scaleio/mdm_failover/bin/mdm_failover_post_install.py --mdms_list
[10.116.27.245,10.116.27.229]+[10.116.27.246,10.116.27.230]+[10.116.19.246,10.116.19.230] --tbs_list
[10.116.19.245,10.116.19.229]+[10.116.19.247,10.116.19.231] --username admin --password London1 --
retry_timeout=30 --retry_count=20
Found 3 MDMs' IPs
Found 2 Tie-Breakers' IPs
Successfully started mdm_failover as a service, to stop it use /opt/emc/scaleio/mdm_failover/bin/delete_service.sh
On Secondary MDM
```

On Standby MDM

```
/opt/emc/scaleio/mdm_failover/bin/mdm_failover_post_install.py --mdms_list
[10.116.27.245,10.116.27.229]+[10.116.27.246,10.116.27.230]+[10.116.19.246,10.116.19.230] --tbs_list
[10.116.19.245,10.116.19.229]+[10.116.19.247,10.116.19.231] --username admin --password London1 --
retry_timeout=30 --retry_count=20
Found 3 MDMs' IPs
Found 2 Tie-Breakers' IPs
Successfully started mdm_failover as a service, to stop it use /opt/emc/scaleio/mdm_failover/bin/delete_service.sh
On Secondary MDM
```

where is xxxxxx is the admin password for the cluster.

There is no requirement to run any commands on the Tie-Breaker nodes.

17.2.2.5 Changing the `retry_timeout` and `retry_count` value

Operational experience may require the retry timeout and count values to be modified. For example, they may need to be increased to ensure that MDM replacement does not take place on an MDM that takes longer to reboot than initially expected. To change these values, simply:

- 1) Stop the MDM failover service on EACH the primary, secondary and standby MDM servers

```
# /opt/emc/scaleio/mdm_failover/bin/delete_service.sh
```

- 2) Setup the standby MDM service again by re-running the command in the section on "[Setting up automatic MDM replacement \(only\)](#)" or "[Setting up automatic MDM & TB replacement on Tornado++](#)"

Note that the settings you originally used for setting up the standby MDM service can be seen in the file `/opt/emc/scaleio/mdm_failover/cfg/conf.txt`.

17.2.2.6 Resetting MDMs after automatic replacement has taken place

If the standby MDM has been invoked due to a prolonged outage of either a primary or secondary MDM, but the original MDM that it replaced is now available again, it is recommended that the standby MDM service be manually taken down to allow the automated MDM process to bring the "formerly-replaced" MDM back into service.

- 1) Ensure that the mdm failover service is running on all three mdm nodes. Run the following on each server running the MDM service and check for the two mdm_failover processes are running.

```
# ps -ef | grep mdm_failover
root  5405  1 0 08:15 ?      00:00:00 /bin/bash -f /opt/emc/scaleio/mdm_failover/bin/run_bin.sh
root  5410  5405  0 08:15 ?      00:00:01 /usr/bin/python /opt/emc/scaleio/mdm_failover/bin/mdm_failover.py --
log_dir /opt/emc/scaleio/mdm_failover/bin/../logs --conf_file /opt/emc/scaleio/mdm_failover/bin/../cfg/conf.txt
```

- 2) Run the following command on the current primary MDM server to monitor the mdm failover process

```
# tail -f /opt/emc/scaleio/mdm_failover/logs/mdm_failover.log
```

- 3) Temporarily shut down the current SECONDARY MDM service (i.e. the initial standby MDM).

```
# /opt/emc/scaleio/mdm/bin/delete_service.sh
```

After ~300 seconds the log file should show the automated mdm failover process being invoked and the output will confirm that the MDM that was replaced by the standby MDM is now the secondary MDM again.

If for some reason the automated MDM replacement service does not configure in a new secondary MDM, restart the mdm service on the current secondary to ensure that you have two live MDMs, and take a look at the troubleshooting process below.

- 4) Restart the MDM service on the standby MDM so that it is available to be substituted in again should there be another failure of either the primary or secondary MDM.

```
# /opt/emc/scaleio/mdm/bin/create_service.sh
```

17.2.2.7 Troubleshooting Standby MDM/TB

If the Standby MDM fails to be swapped into the cluster, ensure that:

- the mdm_failover service is running on the current primary MDM
- the mdm service is running on the Standby MDM.

A useful source of information on the status mdm_failover process can be found by tailing the mdm_failover log file on the current primary MDM

```
# tail -f /opt/emc/scaleio/mdm_failover/logs/mdm_failover.log
```

17.2.2.8 Disabling automated MDM/TB replacement

The automated MDM replacement service can be disabled by running the following command on ALL three MDM nodes.

```
# /opt/emc/scaleio/mdm_failover/bin/delete_service.sh
```

It can also be restarted with the command

```
# /opt/emc/scaleio/mdm_failover/bin/create_service.sh
```

Note that it is recommended that the automated MDM replacement service is disabled prior to activities such as upgrading ScaleIO MDM packages to ensure that the process of upgrading the package in a controlled manner does not trigger an MDM replacement.

17.2.3 Tie-Breaker Nodes

Each ScaleIO Cluster will have a Tie-Breaker node that will tell the ScaleIO cluster what to do in a split brained scenario. The Tie-Breaker software can be loaded on any SDS Node in the ScaleIO cluster not already also acting as an MDM, but where possible should be in a separate fault domain (e.g. separate rack) wherever possible.

For Tornado++ configurations, a standby Tie-Breaker should also be configured (see [Setting up automatic MDM & TB replacement on Tornado++](#)). The TB and the standby TB should be on different servers in the same rack (see ScaleIO Environments on Tornado++ Hardware Best Practice document for further details).

17.2.4 SDS Nodes

SDS nodes are the servers that will contribute disks to make up a ScaleIO [Storage Pool](#). As a result, it will mainly be racks of SDS Nodes that are the building block of the ScaleIO Cluster and their layout is crucial to other best practices in the cluster. The following best practices should be observed for all SDS nodes that make up a ScaleIO Cluster.

- Each SDS server can only be part of one [Protection Domain](#).
- Each SDS server in the cluster should have the same number of disk drives. If this is impossible, then each SDS server/node should only contribute the same exact number of disks to the ScaleIO cluster.
- Each SDS server should have the same amount of RAM. If this is not possible, then when enabling RAM Read Cache, each SDS should be configured to use the same amount of [RAM Read Cache](#).
- Each rack containing SDS Nodes, should have the same number of SDS nodes in a rack and should be connected to the same power source of all other SDS Nodes in the same [Fault Set](#).
- Each SDS server/node should have two dual NIC cards. One port from each NIC is connected to the each of the two client facing networks. The other port on each NIC is connected to the two private cluster networks.

17.2.4.1 Throttling SDS rebuild/rebalancing network bandwidth in R1.31

ScaleIO has IO priority policies to control the impact of rebuild and rebalance operations on client IO. However, during testing it was found that these policies are ineffectual at limiting rebuild/rebalancing impact due to a known but that is fixed in R1.32.

If the impact of rebuild or rebalancing operations are found to be unacceptable when using R1.31, then rebuild & rebalancing network usage limits can be capped. This is a crude workaround as rebuilds & rebalances will be capped even when there is no conflict with client IO.

The caps can be set with the “scli --set_sds_network_limits” command. For example,

```
# scli --mdm_ip 10.59.73.74 --set_sds_network_limits  
--protection_domain_name domain1 --rebuild_limit 100
```

You should initially try setting the limit to 100MB/s, but may need to increase if the rebuild/rebalance rate slows to an unacceptable level in your cluster.

17.2.5 Tiered Storage Pools

Tiered storage pools refers to pools created based on disk types. If each of the SDS nodes in a ScaleIO cluster contains both SSD and HDD drive types, a separate pool would be created with each of the drive types since each would perform differently. For this initial release of ScaleIO it will be supported to have multiple pools or tiered pools in a ScaleIO Cluster where the SDS nodes have different disk types.

17.2.6 Thin Provisioning/ Over Commitment

For the initial release of ScaleIO, thin provisioned LUNs will not be supported within Citi. This functionality is targeted for support in a later release, once testing shows it is more robust, and certain other functionality is supported within the ScaleIO (e.g. migration between pools).

Although thin LUNs should not be created with this ScaleIO release, thick LUNs can be over-committed in VMWare environments. All over-commitment should be managed at the VMWare level using Storage DRS.

17.2.7 LUN Sizing

With SAN Storage, LUN sizes were a standard across different storage platforms due to tech refreshes which could call for LUNs to move from one storage type to another. Keeping a LUN size standard allowed for Hardware based vendor data migration solutions to work across different platforms. Since none of the existing Storage Vendor array based data migration tools will work with ScaleIO, LUN size standards are not expected to be required for any ScaleIO physical environments. However, any environments migrating an existing application from SAN Storage over to a ScaleIO environment, should try and match both the number of LUNs and the size of the LUNs if host based migration tools require it.

For VMWare environments, the existing best practices on LUN sizes should be used that are outlined in the [VMWare VSI Storage](#) document.

The following best practice should be used when creating LUNs for ScaleIO physical environments

- Fewer bigger LUNs is better than many small LUNs. Meaning if a server needs 600GB of LUNs, it is better to give (6) 100GB LUNs instead of (12) 50GB LUNs.
- If an application is being migrated from a SAN environment to a ScaleIO environment, the same number of LUNs using the same sizes should be considered on the ScaleIO cluster if host based migration tools require it.
- VMWare environments should continue to use all existing LUN sizing standards

17.2.8 Obtaining and Installing Citi trusted certificates

The EMC self-signed certificates in the REST gateway webserver need to be replaced with Citi trusted certificates. This process must be executed either:

- 1) for EACH REST gateway (generates a certificate specific to that host)
- 2) once to create a single shared certificate that can be used on both hosts. This requires a Virtual IP (VIP) to first be set up. See the section [Virtual IP \(VIP\) for REST gateways](#) for further details on setting up a VIP.

The following subsection describes:

- 1) How to obtain the Citi certificates
- 2) How to create an alternative Citi keystore (/etc/citikeys/scaleiokeystore) for these keys. This is kept in a directory outside of the ScaleIO gateway installation directory to prevent it being overwritten when the ScaleIO gateway is upgraded or re-installed.
- 3) How to configure the ScaleIO REST gateway to use this alternative keystore
- 4) How to access the REST gateway leveraging SSL certificate verification

The “step” numbers below refer to the steps described in the ScaleIO R1.31 User Guide. Unless otherwise stated, the commands should be run on the ScaleIO gateway server itself.

17.2.8.1 Step 1 - Locate keytool application

Find the location of keytool on your server. It is a part of the Java (JRE or JDK) installation on your server. For example, it is likely to be under /opt/jre e.g. /opt/jre/1.7.0_65l64/bin/keytool.

17.2.8.2 Step 2 – Create keystore and RSA private key for host

Generate an RSA private key.

```
# mkdir /etc/citikeys
# /<path>/keytool -genkeypair -alias <ALIAS> -keyalg RSA -keystore
/etc/citikeys/scaleiokeystore
```

You will be prompted to provide the fully qualified domain name of the server on which you are running the REST gateway. You can use the short version of the hostname for the key alias e.g. e8c4-d1360g7-02. **Note that you must use the CSI number (168737) for ScaleIO for the organizational unit.** For example:

```
# mkdir /etc/citikeys

# /opt/jre/1.7.0_65164/bin/keytool -genkeypair -alias nagtsnpod00103 -keyalg RSA -
keystore /etc/citikeys/scaleiokeystore
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: nagtsnpod00103.nam.nsroot.net
What is the name of your organizational unit?
[Unknown]: 168737
What is the name of your organization?
[Unknown]: Citigroup Inc.
What is the name of your City or Locality?
[Unknown]: Warren
What is the name of your State or Province?
[Unknown]: NJ
What is the two-letter country code for this unit?
[Unknown]: US
Is CN= nagtsnpod00103.nam.nsroot.net, OU=149141, O=Citigroup Inc., L=Warren, ST=NJ,
C=US correct?
[no]: yes

Enter key password for <e8c4-d1360g7-02>
(RETURN if same as keystore password):
```

You also be prompted for a password for your keystore. You will also be prompted for a password for your key.

1. Key and keystore passwords should be identical. You can just hit return to keep the key password the same as the keystore password.
2. The same password can be used for all ScaleIO clusters
3. The password should be secure and secret
4. The password should conform to Citi standards for complexity

17.2.8.3 Step 3 – Create CSR

Generate a Certificate signing request (CSR) to file certreq.txt

```
# /opt/jre/1.7.0_65164/bin/keytool -certreq -keyalg RSA -alias nagtsnpod00103 -file certreq.txt -
keystore /etc/citikeys/scaleiokeystore
Enter keystore password:
```

Your CSR should look something like this:

```
# cat certreq.txt
-----BEGIN NEW CERTIFICATE REQUEST-----
```

```

MITIC8DCCAdgCAQAwEzELMAkGA1UEBhMCTkExCzAJBgNVBAGTAk5KMzQ8wDQYDVQQHEWZXYXJyZW4x
GDAWBgNVBAoTD0NpdGlnbm91cCwGSw5jLjELMAkGA1UECjxMCTkExJzAlBgNVBAMTHmU4YzQtZGwz
NjBnNy0wMi5uYW0ubnNyb290Lm5ldDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBBAK0w
EAiB5imtZhhZznupBQMYgYuCwj7VQNp6v1ao6iIASrpqZhOvtBX0mAFrZJ5TpT0+oQGjBkR41up1
I/mLLB1JRQ6/6/sPac0v7c9YYMYoh0PZQtPzQSaI4YPeU3UxQgB2Nzn29W21+Kx+SY8HXEkIdXHi
9hMr6rLMzbKwPlrPh9JzbY4uXAEijqNveqpb0if/XKNC2F3Mn4Pjczd3HdgQilpadeIKdsi4BY1T
0TuliCwE8uXUKNcR+97+s7wL4c2Q16U8k331qSfYwbCgTdoYv1U/cH0Wwhs0QPFjQ4a7/iBwCSSI
t1ldB00y9Ey/r2sZvm28RIB8s41DbYyZoamCAwEAAaAwMC4GCSqGSIb3DQEJJDjEhMB8wHQYDVR0O
BBYEFc6t2KbMEL++bzzp/OwmX/4YT35KMA0GCSqGSIb3DQEBCwUAA4IBAQA9J5icKKNXtp4hq3SF
iwWyftpcFPeAwAs/8g5sJTtSYrNrNgvNkmbbn+0Z+LZ1AHRZHMuzeJkA76duWpNtpt8FJ4Vegx0
xXLbce7ZN2BSE8gVwo+NF51vRiS40pPJ0kilDt9HcildaixR/X6/y3VndmGhYWMcz0mGA87RRBf9
pZyIP6pbqJDiyABq4tDTAH4qbc1+aZvU81cm3bWv/awehTuJa3cadDyWJl6POqwpYtZT8/bxgKA
95HNxbBY3Jac8LDZRz2DaNJG5j55ud9+LK3AhP0/dYD7SPZsI8WT3f+ntkWReIKB+gLqIA9JtwWu
Z+UScs2CYeDgWbxuPjCD
-----END NEW CERTIFICATE REQUEST-----

```

17.2.8.4 Step 4 – Request signed certificate for host

Use the CSR to request a signed certificate from Marketplace. Note that Entrust certificates are now EOL in Ciit so you need to get a Venafi certificate.

CMP - Venafi MS-PKI On Demand Device Certificate - 163513 - Create/Revoke Certificate:

https://cmp.nj.smb.com/marketplace/control/product/~productId=47119_29952_GLOBAL#

The following example shows the process for requesting a certificate for a lab environment. For a production environment, you need to specify the following:

1. “General Profile” Page of Wizard

Enter the following CSI ID and click on Lookup.

CSI ID: [168737](#)

Once the CSI number is input, other fields such as Application Short Name, Application Owner and GEID/SOEID of Application Owner will be auto-filled.

2. “SSL Cert Info” Page of Wizard

Requested Information	Data to be input by user
Organizational Unit By Region:	<Pick your region>
What action do you require:	Create
Certificate Authority:	PROD (or UAT, DEVELOPMENT or LAB if appropriate for your instance)
Target hostname:	<Enter/select the fully qualified domain name of your first REST gateway host e.g. nagtsnpod00103.nam.nsrout.net
Type of certificate:	MS CA Web Server
Port Number:	443
Service Generated or user provided CSR?	CSR
Enter CSR:	<Paste in the contents of your certreq.txt file> from Step 3
Certificate Subject Alternative Name (SAN) DNS name(s)	For servers with more than one interface, include all <u>public</u> network interface names (<u>including</u> the name of the server) that are registered for the node in

	DNS. The VIP for the gateways must also be included, if you have configured one. This must be a comma-separated list e.g. for a T++ configuration: siogwracaas.nam.nsroot.net, nagtsnpod00103.nam.nsroot.net, nagtsnpod00103-eth2.nam.nsroot.net, nagtsnpod00104.nam.nsroot.net, nagtsnpod00104-eth2.nam.nsroot.net
--	--

Wizard View: Answer each set of questions and click next. Once you are done, click ADD TO CART

1 SSL Cert Info

2 SSL Cert Info

* Organizational Unit by Region

us

* What action do you require

Create

Alternate Certificate Owner

No Value Set

Update

* Which Certificate Authority do you require for the certificate?

PROD

* Environment

PROD

Link Up Reset

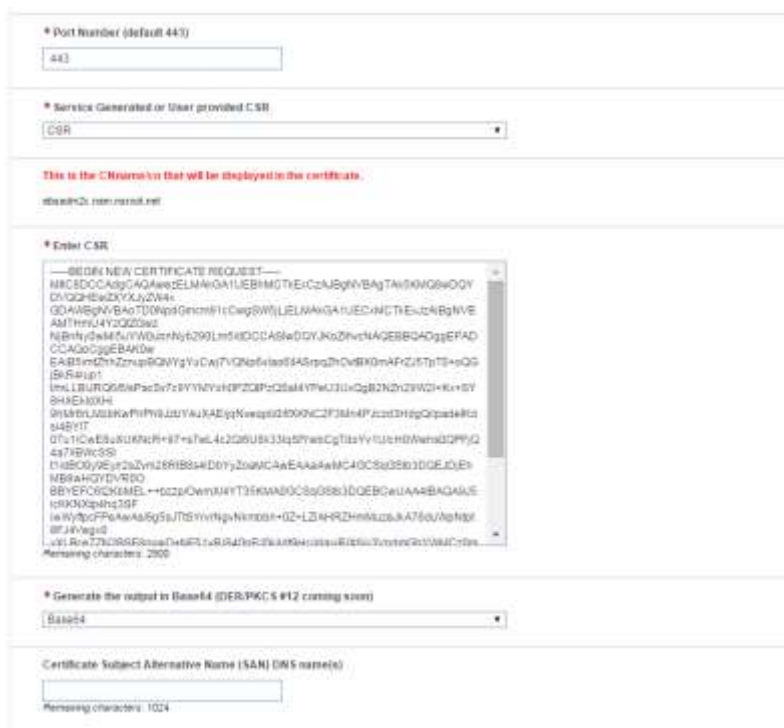
* Target Hostname

stasdr01.nam.nsroot.net

Link Up Reset

* What Type of Certificate is Required?

MS CA Web Server



Note again that for dual-homes systems you must include all public network names registered in DNS for all interfaces in the “*Certificate Subject Alternate Name (SAN) DNS name(s)*” box at the end of the form. These should be in a comma-separated list.

The following table summarizes what names should be used for the “Target hostname” and the “Certificate Subject Alternate Name (SAN) DNS names(s)”.

Gateway model	Target hostname	Certificate Subject Alternate Name (SAN) DNS names(s)
Multiple gateways accessible via a common Virtual IP (VIP) (e.g. T++ cluster) The <u>same</u> certificate is used for both REST gateways	The first gateway FQDN (Note that you cannot select the VIP here, as not registered as a server in aperture).	The FQDN of all the client network interfaces on all the gateway servers, <u>as well as the VIP FQDN</u> . This allows the gateways to be accessed directly as well as via the VIP.
Gateways with a single client network interface (e.g. MaaS POC cluster) You need a unique certificate needed for each REST gateway	The gateway FQDN	Leave empty (as in screenshot above)
Gateways with dual client network interfaces	The gateway FQDN	The FQDN of all the client network interfaces, including that listed as the target hostname. This allows the gateway to be accessed via any

You need a unique certificate needed for each REST gateway		interface e.g. when the interface associated with the gateway FQDN is down.
--	--	---

Fill in the business justification text box and then click on “Quick order”.

After approval, you will get a signed certificate back by e-mail that you will need to import back into the keystore when you get to Step 7. Copy the certificate into a file e.g. e8c4-d1360g7-02.pem.

17.2.8.5 Steps 5 & 6 – Download and import root and intermediate certs.

Download the Citi Trusted root and intermediate certificates in PEM format for your environment (PROD, UAT, Development or Lab). These CA certificate “chains” contain both the trusted root and intermediate certificates in a single file.

1) Download the CA Certificate Chain

Microsoft PKI CA Certificates	URL for certificate chain in PEM format
Production CA Certificates in PEM format	https://catecollaboration.citigroup.net/domains/security/isp_standards/dpstds/Microsoft%20PKI/Certificates/PROD/CitiInternalCAChain_PROD.pem
UAT CA Certificates in PEM format	https://catecollaboration.citigroup.net/domains/security/isp_standards/dpstds/Microsoft%20PKI/Certificates/UAT/CitiInternalCAChain_UAT.pem
Development CA Certificates in PEM format	https://catecollaboration.citigroup.net/domains/security/isp_standards/dpstds/Microsoft%20PKI/Certificates/DEV/Citi_DEVELOPMENT_chain.pem
LAB CA Certificates in PEM format	https://catecollaboration.citigroup.net/domains/security/isp_standards/dpstds/Microsoft%20PKI/Certificates/LAB/CitiInternalCAChain_LAB.pem

2) Take the downloaded file and split the root and intermediate keys up into separate text files for each of the certificates (e.g. cert1.pem, cert2.pem, etc).

The last entry in the certificate chain file is the trusted root, which then signs the next certificate higher up the file, and so up to the top certificate in the file. This top certificate signs the host-specific certificate that you obtain from CMP in Step 4. So you install the last certificate in the certificate chain file first in the chain file and then work up the entries.

For example, the file Production CA certificates file at the link above contains the following 6 certificates in order, and you install the 6th certificate, then the 5th certificate etc until all 6 are installed.

- 1st certificate: Citi Internal User CA 01
- 2nd certificate: Citi Internal User CA 02
- 3rd certificate: Citi Internal Device CA 01
- 4th certificate: Citi Internal Device CA 02
- 5th certificate: Citi Internal Policy CA
- 6th certificate: Citi Internal Root CA

Note that for certificate chains other than production, the number and names of these keys may vary. For example, in the Development CA certificates chain file, there are only 3 certificates.

You can confirm which key is in a given file with the following command. For example, you can check the contents of the 5th certificate (once you have placed it into a separate file) with the command

```
# /usr/bin/openssl x509 -text -in cert5.pem | grep Subject | grep DC
Subject: DC=net, DC=Citi, CN=Citi Internal Policy CA
```

- 3) Import these keys to the Citi scaleiokeystore one at a time, starting with the ROOT CA key (last certificate in the certificate chain file).

```
# /opt/jre/1.7.0_65164/bin/keytool -import -alias citiinternalrootca -keystore
/etc/citikeys/scaleiokeystore -trustcacerts -file ./cert6.pem

Trust this certificate? [no]: yes
Certificate was added to keystore

# /opt/jre/1.7.0_65164/bin/keytool -import -alias citiinternalpolicyca -keystore
/etc/citikeys/scaleiokeystore -trustcacerts -file ./cert5.pem
Enter keystore password:
Certificate was added to keystore

(certificate goes straight in without being asked if you trust it as signed by the root certificate)

# /opt/jre/1.7.0_65164/bin/keytool -import -alias citiinternaldeviceca02 -keystore
/etc/citikeys/scaleiokeystore -trustcacerts -file ./cert4.pem
Enter keystore password:
Certificate was added to keystore

(certificate goes straight in without being asked if you trust it as signed by the root certificate)

etc.
```

until all certificates have been imported.

Note that it does strictly not matter if you do not enter them in the correct order, but if you do not, you will just need to manually accept any certificate not signed by an already imported certificate.

```
Trust this certificate? [no]: yes
```

17.2.8.6 Step 7 – Import signed certificate for host

Now we finally install the SSL certificate for the host under the same alias that the CSR was created from in Steps 3 & 4.

Install the SSL certificate for the host that you received from Citi MarketPlace by e-mail into the keystore. You must use the same alias that your CSR was create from in Step 2 (e8c4-d1360g7-02 in the example in Step 2 above).

```
# /opt/jre/1.7.0_65164/bin/keytool -import -alias nagtsnpod00103 -keystore
/etc/citikeys/scaleiokeystore -trustcacerts -file ./nagtsnpod00103.pem
Enter keystore password:
Certificate reply was installed in keystore
```

- 4) Confirm the entries in your scaleio keystore. You should have one entry for each key that you imported. The root and intermediate keys show up as “trustedCertEntry” whereas the key for your host that you obtained via the CSR process will show up as “PrivateKeyEntry”.

```
# /opt/jre/1.7.0_65164/bin/keytool -list -keystore /etc/citikeys/scaleiokeystore
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 7 entries
citiinternaluserca02, Jan 20, 2015, trustedCertEntry,
Certificate fingerprint (SHA1): BE:76:9A:9A:33:3E:55:9A:AA:EA:C5:A9:DC:CD:BE:12:5C:60:8C:93
citiinternaldeviceca02, Jan 20, 2015, trustedCertEntry,
Certificate fingerprint (SHA1): 1C:0E:21:B3:4F:8F:E7:63:D8:7B:7C:A5:78:A0:21:7E:C2:08:7F:A0
citiinternaluserca01, Jan 20, 2015, trustedCertEntry,
Certificate fingerprint (SHA1): AA:AE:CB:32:5B:22:A6:F0:C7:D8:4C:FB:80:F6:54:AC:58:F6:B5:E8
citiinternaldeviceca01, Jan 20, 2015, trustedCertEntry,
Certificate fingerprint (SHA1): B0:79:E1:6C:1E:8F:A1:72:00:92:F4:0F:ED:93:10:A1:27:50:31:7E
nagtsnpod00103, Jan 7, 2015, PrivateKeyEntry,
Certificate fingerprint (SHA1): D3:91:90:EF:FB:F1:F8:38:1A:89:29:C2:1D:1E:24:CE:B2:95:C8:A3
citiinternalrootca, Jan 20, 2015, trustedCertEntry,
Certificate fingerprint (SHA1): DC:C1:D6:EF:F3:35:B5:3E:FD:87:1C:79:C6:70:D3:99:FE:B7:5F:31
citiinternalpolicyca, Jan 20, 2015, trustedCertEntry,
Certificate fingerprint (SHA1): AA:D8:F9:54:2D:C1:2C:1B:D5:FD:11:04:6F:5A:79:79:8B:80:92:FF
```

17.2.8.7 Step 8 – Point ScaleIO gateway at new keystore

Modify the ScaleIO webserver catalina.properties file to point to the Citi alternate keystore. Comment out the existing entries for the ScaleIO keystore and insert entries for the file location of your keystore and its password. Note that this file can only be read by root.

```
# vi /opt/emc/scaleio/gateway/conf/catalina.properties

keystore.file = /etc/citikeys/scaleiokeystore
keystore.password = Vinn608Qwt
#keystore.file = /opt/emc/scaleio/gateway/conf/certificates/.keystore
#keystore.password = HvbjQLdvCjwKc4fNeKiAN0F52I6KAp
```

17.2.8.8 Step 9 – Restart ScaleIO gateway

Restart the ScaleIO gateway service.

```
# service scaleio-gateway restart
```

17.2.8.9 Step 10 – Test webserver certificate verification

You can confirm that the gateway SSL certificate is working correctly with the following command. Note that you must specify the fully qualified domain names (FQDN) in the URL and supply the CA public certificates file you downloaded in Step 5 from as the cacert file.

```
# curl -X GET -H 'Accept: application/json;version=1.0' -u admin https://<FQDN>/api/login --cacert <CA public certificates file>
```

For example, this command requests a session key (the long string of numbers and letters) only if the REST server key is valid.

```
# curl -X GET -H 'Accept: application/json;version=1.0' -u admin https://nagtsnpod00103.nam.nsroot.net/api/login --cacert ./CitiInternalCAChain_PROD.pem
Enter host password for user 'admin':
"YWRtaW46MTQyOTAyNzQ1MzI5MjoyNDB1MjVjOTFjOTU3MjVhODJhZmYxNzY1M2I1ZjE5YQ"

# curl -X GET -H 'Accept: application/json;version=1.0' -u admin https://nagtsnpod00103-eth2.nam.nsroot.net/api/login --cacert ./CitiInternalCAChain_PROD.pem
Enter host password for user 'admin':
"YWRtaW46MTQyOTAyNzQ1MzI5MjoyNDB1MjVjOTFjOTU3MjVhODJhZmYxNzY1M2I1ZjE5YQ"
```

Now repeat all 10 steps for the second REST gateway, unless using a Virtual IP in front of the gateways. If using a Virtual IP go to Step 11 for instructions on how to copy to and configure the keystore on the second REST gateway

17.2.8.10 Step 11 – Copying the keystore to the second REST gateway when VIP in use

The same keystore can be used on the second gateway and hence the process to setup the second gateway is trivial, as it simply requires on to:

- ftp the keystore from the first gateway to /etc/citikeys/scaleiokeystore on second gateway

- Point second ScaleIO gateway webserver at the new keystore (identical to Step 8 above)

```
# vi /opt/emc/scaleio/gateway/conf/catalina.properties

keystore.file = /etc/citikeys/scaleiokeystore
keystore.password = Vinn608Qwt
#keystore.file = /opt/emc/scaleio/gateway/conf/certificates/.keystore
#keystore.password = HvbjQLdvCjwKc4fNeKiAN0F52I6KAp
```

- Restart the ScaleIO gateway (identical to Step 9 above)

```
# service scaleio-gateway restart
```

- Test webserver certificate verification (similar to Step 10 above, but with extra tests. See below)

In the examples below, the first REST gateway is `e8c4-dl360g7-01.nam.nsroot.net`, the second is `e8c4-dl360g7-02.nam.nsroot.net` and the VIP is `sigowwarren01.nam.nsroot.net`. All three FQDNs can be used to access the gateways i.e. either via the VIP or directly to a specific gateway.

```
# curl -X GET -H 'Accept: application/json;version=1.0' -u admin
https://sigowwarren01.nam.nsroot.net/api/login --cacert ./Citi_DEVELOPMENT_chain.pem
Enter host password for user 'admin':
"YWRtaW46MTQzMTEwMjQ1MzgZMT02YjIyYTQ4OTZhMDJiMmMxMjNkYzJkNjIzOTQxYTQzNw"
```

```
# curl -X GET -H 'Accept: application/json;version=1.0' -u admin
https://e8c4-dl360g7-01.nam.nsroot.net/api/login --cacert ./Citi_DEVELOPMENT_chain.pem
Enter host password for user 'admin':
"YWRtaW46MTQzMTEwMjUxOTY4ODozNzMyOTUwMTY4Zjg3NzIzNDM3ODczMTE5YTAzYTE1NA"
```

```
e8c4-dl360g7-01# curl -X GET -H 'Accept: application/json;version=1.0' -u admin
https://e8c4-dl360g7-02.nam.nsroot.net/api/login --cacert ./Citi_DEVELOPMENT_chain.pem
Enter host password for user 'admin':
"YWRtaW46MTQzMTEwMzMyMzY6MzQ0GfYWEzODVmM2VmMWRhNzNlMGM2MGRkODBhOGRkZg"
```

17.2.9 Virtual IP (VIP) for REST gateways

A virtual IP (VIP) should be setup to allow REST clients that do not support connection to multiple REST gateways to still have resilient connectivity to REST services. This is achieved by pointing the REST client at the VIP that load balances across the both ScaleIO REST gateways, and in doing so also provides resilient access should one of the gateways be unavailable.

A VIP for a production system can be requested via the following URL:
<https://globalnetworkcontrolcenter.ti.citigroup.net/lbo/request>

Detailed documentation for this process is available at <https://globalnetworkcontrolcenter.ti.citigroup.net/getfile/494618>

The e-mail address for the team that supported this process is “*GT GNO LB Request Team”

You will need the CSI ID (168737) and Approver Group (CTI GL EX CLOUD CONVERGED INFRASTRUCURE) for ScaleIO during the request process.

You should request the following VIP features during the request process:

- Environment: Internal/GSS
- VIP support for ports 80 & 443
- LB mode: Round Robin
- Persistence: Simple (this is to ensure that any REST client that is using a cached authentication token goes to the same gateway , as these are not interchangeable between gateways)
- Length: 10 Min.

Ignore (leave blank) any references to Ratio/Priority (not applicable to Round Robin) or to ECV.

You should request a VIP for VIP ports 80 and 443 for ONE of the subnets that the gateway is connected to on the client-facing network.

You should use the final Comments text box to request a specific DNS entry for the VIP, although this may be ignored. The following naming convention is recommended:

```
siogw<scaleio-cluster-name>
```

For example:

```
siogwracaas.nam.nsroot.net
```

An example screenshots of the process are shown on the following pages. Your entries will be different.

Load Balancer Operations - Change Request

Step 1 of 8: Load Balancer Region

[Load Balancer Basics](#)

Please select your region:

- ☒ NA/LATAM/Mexico
- ☐ EMEA
- ☐ ASPAC
- ☐ JAPAN
- ☐ AUSTRALIA

< Previous Next > << Restart

Load Balancer Operations - Change Request

Step 2 of 8: Load Balancer Type

[Load Balancer Basics](#)

Please select the type of load balancer you require:

- ☒ Local Area Load Balancer (VIP) ⓘ
- ☐ Wide Area Load Balancer (WIP) ⓘ

< Previous Next > << Restart

Load Balancer Operations - Change Request

Step 3 of 8: Request Type

[Load Balancer Basics](#)

Please select the type of request you are creating:

- ☒ New Vip/WideIP 
- ☐ Modify Existing VIP/WideIP 
- ☐ Delete VIP/WideIP or Member Server(s)/Virtual Server(s)
- ☐ Informational 

< Previous Next > << Restart

The following is just an example. Fill in your own information in the form below.

Load Balancer Operations - Change Request - New VIP

Step 4 of 8: Requestor Contact Information

[Load Balancer Basics](#)

Please provide the following contact information:

Contact Name:

Contact Phone:

Email Address:

Group Email:

Primary Owner SOEID: 

Secondary Owner SOEID: 

< Previous Next > << Restart

Load Balancer Operations - Change Request - New VIP

Step 5 of 8: Virtual Change Information

[Load Balancer Basics](#)

Please provide the following Virtual Change information:

SN Approval Group(s): CTI GL EX CLOUD CONVERGED INFRASTRUCTURE

CSI AppID: 168737

Application Instance: DEFINED STORAGE~GLBL~PROD~1

Datacenter Migration: ☐

Target Date: 05/26/2015

- Internal Standard Greenzone: Sun-Friday 19:00 to 04:00 EASTERN
- External Standard Greenzone: Sun-Friday 02:00 to 05:00 EASTERN

If a different time is desired please specify in the additional comments

< Previous Next > << Restart

Load Balancer Operations - Change Request - New VIP

Step 5 of 8: Virtual Change Information

[Load Balancer Basics](#)

Please provide the following Virtual Change information:

Target Date Info

Note: the "Target Date" requested is **NOT** the finalized implementation date for the change. All load balancer requests will be completed **on or before** this date **if possible**. If this request **MUST NOT** be deployed earlier than the "Target Date" requested, you are **REQUIRED** to provide those details in the requests comments section. You must also engage the assigned LB technician to ensure your scheduled date is coordinated properly. The finalized change record should reflect the agreed upon date.

OK Cancel

- Internal Standard Greenzone: Sun-Friday 19:00 to 04:00 EASTERN
- External Standard Greenzone: Sun-Friday 02:00 to 05:00 EASTERN

If a different time is desired please specify in the additional comments

< Previous Next > << Restart



Load Balancer Operations - Change Request - New VIP

Step 6 of 8: VIP Information

Load Balancer Basics

Please provide the following VIP information:

Environment: ☒ Internal/GSS ☐ Citiplex/Iplex

Data Center: GTDC - Georgetown

VIP: NEW VIP VIP Port: 443 ☐ Add Certificate?

LB Mode: Round Robin Persistence: Simple Length: 10 Min Add

VIP	DNS	VIP Port	Add Certificate?	LB Mode	Persistence	Length	
NEW VIP	*.nam.nsroot.net	443	☐	Round Robin	Simple	10 Min	✗

< Previous Next > << Restart

Load Balancer Operations - Change Request - New VIP

Step 6 of 8: VIP Information

Load Balancer Basics

Please provide the following VIP information:

Environment: ☒ Internal/GSS ☐ Citiplex/Iplex

Data Center: GTDC - Georgetown

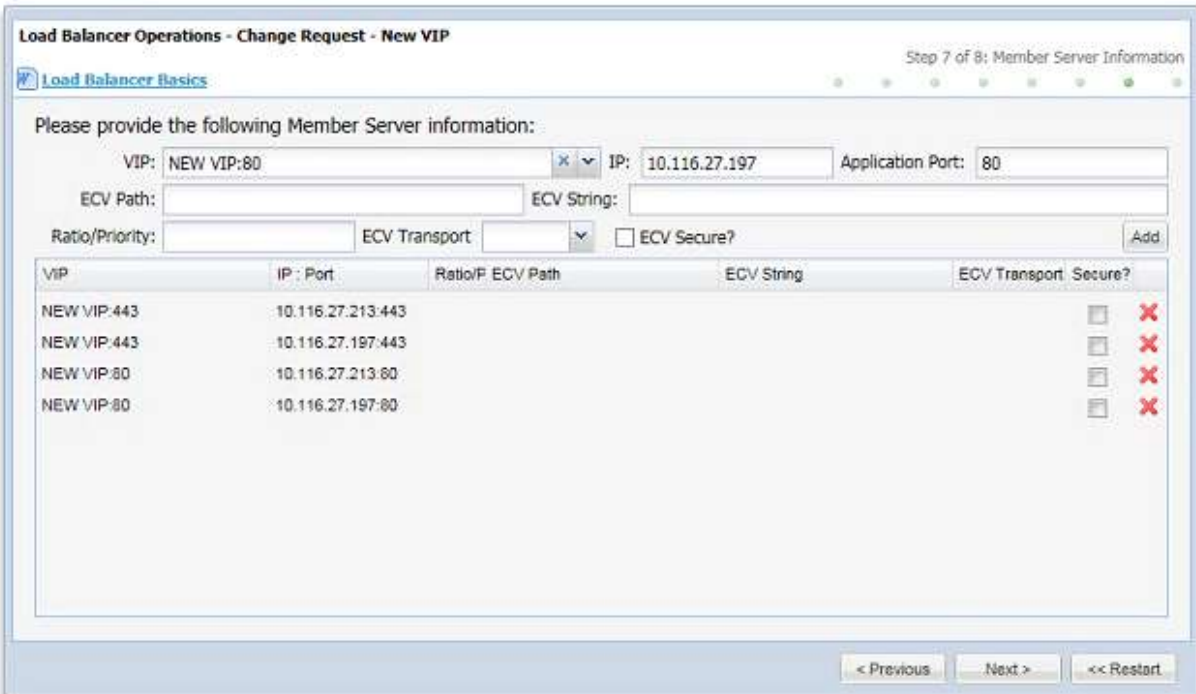
VIP: NEW VIP VIP Port: 80 ☐ Add Certificate?

LB Mode: Round Robin Persistence: Simple Length: 10 Min Add

VIP	DNS	VIP Port	Add Certificate?	LB Mode	Persistence	Length	
NEW VIP	*.nam.nsroot.net	443	☐	Round Robin	Simple	10 Min	✗
NEW VIP	*.nam.nsroot.net	80	☐	Round Robin	Simple	10 Min	✗

< Previous Next > << Restart

In the example below, VIPs for ports 80 and 443 are being configured across the IP addresses 10.116.27.213 & 10.116.27.214, both of which is on a different ScaleIO gateway but are in the same subnet.



Load Balancer Operations - Change Request - New VIP Step 7 of 8: Member Server Information

Load Balancer Basics

Please provide the following Member Server information:

VIP: NEW VIP:80 IP: 10.116.27.197 Application Port: 80

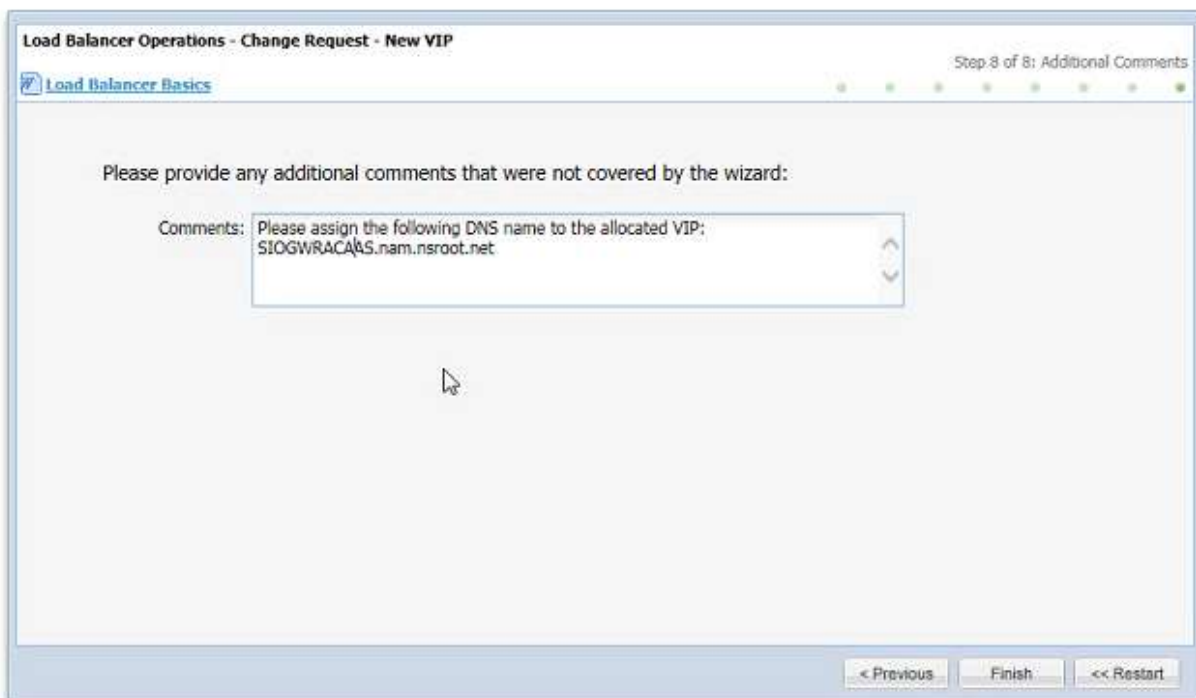
ECV Path: ECV String:

Ratio/Priority: ECV Transport: ☐ ECV Secure? Add

VIP	IP : Port	Ratio/P	ECV Path	ECV String	ECV Transport	Secure?
NEW VIP:443	10.116.27.213:443				☐	☒
NEW VIP:443	10.116.27.197:443				☐	☒
NEW VIP:80	10.116.27.213:80				☐	☒
NEW VIP:80	10.116.27.197:80				☐	☒

< Previous Next > << Restart

Use the following screen to request the specific DNS name that you want assigned to this VIP.



Load Balancer Operations - Change Request - New VIP Step 8 of 8: Additional Comments

Load Balancer Basics

Please provide any additional comments that were not covered by the wizard:

Comments: Please assign the following DNS name to the allocated VIP:
SIOGWRACAS.nam.nsroot.net

< Previous Finish >> Restart

When you click on finish, it will show a summary of the New Load Balancer request with all the information just entered.

Click on Submit and you will get an e-mail confirmation that the request has been successfully submitted.

Note that if the VIP is not set up in DNS with the name you requested in the comments field, but instead is assigned the Load Balancer request number as the name e.g,

B105185-vip.nam.nsroot.net

then you will need to change it yourself to your preferred name, via the following process:

Submit an automated marketplace request [here](#) for DNS object level changes.

- The request will be processed via automation once it is approved (it takes around 20mins)- This is applicable if the requester raised the request by selecting the option 'Execute Now'
- If the requester wants to schedule a particular request for a future date, there is an option to select the implementation date and hour (GMT)
- requester will be able to raise request for Add/Modify/Delete objects using this request.
- As there is no option to update the IP address of an object, requester has to submit a 'Delete' & 'Add' request to update the current IP with a new one.