Technical Report

NetApp SolidFire Enterprise SDS Running Microsoft SQL Server and Virtualized Infrastructure

Solution Overview

Josh Powell and Sufian Ahmad, NetApp

October 2020 | TR-4866 | Version 1.0

Abstract

This document introduces NetApp® SolidFire Enterprise Software-Defined Storage (eSDS) to the database administrator. Built with the same Element software as every NetApp SolidFire-based storage offering, SolidFire eSDS provides customers the freedom to deploy the Element storage operating system as containers, while obtaining hardware platforms from preferred vendors. This document provides guidelines to install Microsoft SQL Server on SolidFire eSDS, as well information on the practical application of using automation and replication features of SolidFire arrays.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Solution Overview

This solution document illustrates how to deploy a Microsoft SQL Server 2017 workload leveraging NetApp SolidFire eSDS. While SQL Server is the focus of this solution, many of the principles conveyed for NetApp SolidFire eSDS are applicable to other applications and workloads.

Many customers are looking to gain efficiencies by standardizing underlying hardware platforms across their data centers. In some cases, customers are looking to deploy and manage their preferred hardware vendor to run all their infrastructure services, including storage and data management systems. The NetApp SolidFire eSDS solution enables customers to gain further efficiency, standardization, and control over their operations.

Use Cases

This section describes the primary use cases that are addressed with this solution.

Database Consolidation

NetApp SolidFire eSDS provides an optimal storage system for database consolidation. Quality of service (QoS) is a unique feature of SolidFire that helps individual databases get the I/O throughput that they need without being affected by other databases that run in parallel on the same storage system. With QoS and data reduction efficiencies, you can achieve higher database density by having multiple SQL instances on a single cluster.

Data Protection and Disaster Recovery

SolidFire redundant data protection (DP) is a distributed replication algorithm that spreads two redundant copies of data across all drives within the entire cluster. For high capacity and performance SQL Server instances, you can group multiple storage volumes together by using Windows dynamic disks in a striped configuration.

SolidFire remote replication provides an efficient way of increasing data availability and reducing downtime. Synchronous, asynchronous, and snapshot replication are supported and can be configured between any products running supported versions of NetApp Element software.

Development and Testing

The ability to quickly recover from corrupted datasets and to create space efficient copies of datasets is of paramount importance in testing and developments environments. New volumes can easily be cloned from existing snapshots and can be coupled with QoS so that database clones can coexist with production instances without any performance degradation.

The CopyVolume feature of SolidFire eSDS allows you to refresh an existing cloned copy of a database without performing any file system remount operations. In this use case, you frequently refresh a copy of the database by only applying changes from the production copy.

Target Audience

The target audience for this solution includes the following groups:

- Microsoft SQL Server database administrators who want to understand the unique value of SolidFire eSDS.
- NetApp partners who would like to assist customers with architecting database solutions.
- Existing NetApp customers who are implementing Microsoft SQL Server on SolidFire clusters.

Solution Architecture

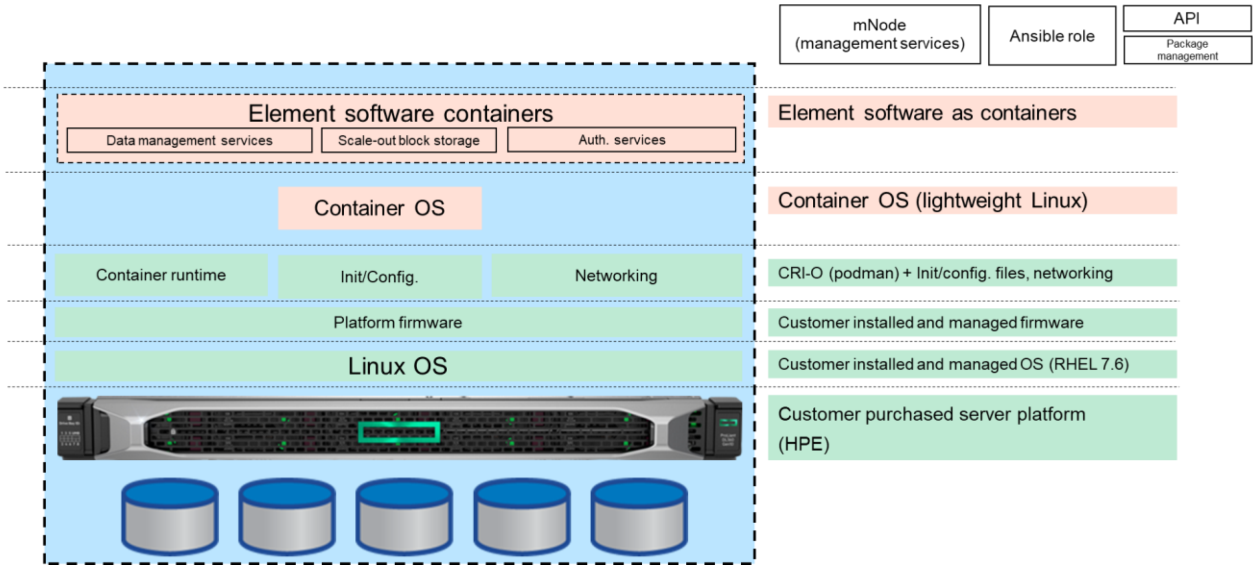
This validated solution uses SolidFire eSDS as primary storage for an environment consisting of VMware vSphere 6.7 with Microsoft SQL Server. Additionally, NetApp HCI is used as a target for the SQL Server volumes hosted on the SolidFire eSDS cluster, taking advantage of the Element software replication capabilities.

SolidFire eSDS Technology Overview

NetApp SolidFire eSDS provides enterprise-class shared storage for building a private cloud infrastructure. Built with the same Element software as every other NetApp SolidFire-based storage offering, the eSDS delivery model allows customers the flexibility to procure storage software independent of the underlying hardware. NetApp SolidFire eSDS delivers the Element storage software as separately installed storage containers that run on RedHat Enterprise Linux. This model satisfies the needs of customers who prefer or require a separate hardware vendor and increases flexibility and operational efficiencies.

Figure 1 illustrates the storage node components that are provided by the customer and by NetApp.

Figure 1) SolidFire eSDS node architecture.


mNode (management services)
Ansible role
API
Package management
Element software containers
Data management services
Scale-out block storage
Auth. services
Element software as containers
Container OS
Container OS (lightweight Linux)
Container runtime
Init/Config.
Networking
CRI-O (podman) + Init/config. files, networking
Platform firmware
Customer installed and managed firmware
Linux OS
Customer installed and managed OS (RHEL 7.6)
Customer purchased server platform (HPE)

Each storage node hosts three containers that are managed as a Linux service:

- **Element authentication container.** Provides authentication features.
- **Network watchdog container.** Service that handles high availability features and recovery from failures.
- **Element software container.** Hosts, starts, and monitors the Element software services.

SolidFire storage arrays running Element software feature a number of important enterprise-class capabilities that differentiate the NetApp software-defined offering from others in the market:

Scalable Infrastructure

SolidFire clusters can grow quickly from 4 to 40 nodes without any downtime or impact to production workloads. Add new nodes as needed and experience nondisruptive expansion with instant resource availability. Performance scales linearly as new nodes are introduced to the cluster and the pool of available IOPS is clearly displayed as they are assigned to applications through QoS of service policies.

High Availability Architecture

The shared-nothing architecture of SolidFire creates no single point of failure and can absorb multiple failures. The combined storage efficiency and QoS of SolidFire provides a compelling disaster recovery solution that enables the sharing of storage resources for multitenant applications without performance penalties. Auto-healing starts as soon as a drive or node failure is detected for reduced operation overhead.

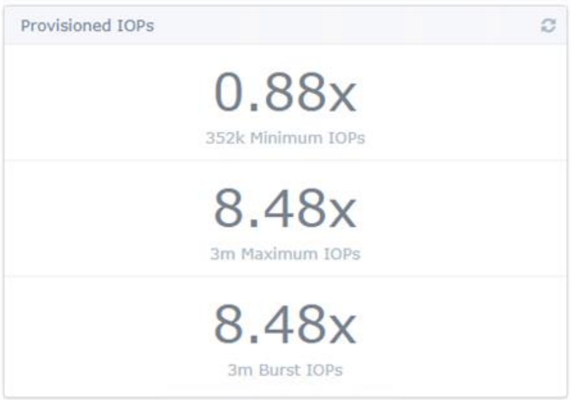
Guaranteed Performance

SolidFire eSDS QoS provides fine-grained control of performance, which allows a wide range of applications to coexist on a single cluster. QoS policies allow the quick application of QoS settings to groups of volumes and can be easily automated through the Element REST API. For Microsoft SQL Server architectures that contain multiple user databases with varying resource needs, multiple SolidFire volumes can be used, each with differing QoS policies.

Control over the minimum IOPS for a LUN is a unique feature that guarantees performance for applications residing in environments containing noisy neighbors.

The overall allocation of the storage cluster performance resources can be viewed from the Element manager overview page. This allows administrators to oversubscribe maximum IOPS while still guaranteeing performance for critical applications by ensuring the total assigned minimum IOPS stays within an acceptable limit.

Figure 2) IOPs provisioned in the cluster.



Provisioned IOPs

0.88x
352k Minimum IOPs

8.48x
3m Maximum IOPs

8.48x
3m Burst IOPs

Data Protection—Snapshots

DP is a key priority for businesses to guarantee a sustained production environment. Any interruption or downtime during production hours can cause substantial business losses. In this respect, NetApp Element software snapshots provide an excellent safeguard against any unintended downtime. Using NetApp Element snapshots, backups can be created and made available swiftly and reliably.

Data Protection—Replication

SolidFire eSDS provides synchronous and asynchronous replication to other NetApp products running Element software. Different replication methods are available based on the customer’s business continuity requirements. The NetApp Element storage feature offers three variants of replication:

- **Synchronous replication.** Synchronous replication provides continuous replication of data from the source to the target cluster—granted, there is acknowledgement from the target cluster. This solution is suitable for replicating data over short distances and for systems that are geographically closer.

Given the continuous nature of this replication and the required acknowledgment from the target cluster, there is the possibility of jitter and latency.

- **Asynchronous replication.** As with synchronous replication, asynchronous replication continuously replicates data from a source to a target cluster. However, asynchronous replication does not require an acknowledgement from the target cluster back to the client system. Asynchronous replication is best suited for business needs that require almost real-time replication without the performance penalty of waiting for acknowledgments from the target.

  Given the performance benefits of asynchronous replication, we implemented asynchronous replication for the solution described in this document.
- **Snapshot-only replication.** Snapshots of volumes can be taken at discrete points of time from the source cluster and can be scheduled to take place periodically at the source. Snapshots do not interfere with synchronous or asynchronous replication. For database administrators, the ability to take snapshots as a group is an important feature that ensures all snapshots for a SQL instance are taken together.

Storage Automation

NetApp Element software allows businesses to increase operational efficiencies and improve storage delivery with native REST-based APIs. This capability allows deep integration with management and orchestration platforms.

One popular method to access API-driven automation features for SolidFire storage platforms is to use the Linux `curl` command. Curl is a command used to send and retrieve information from servers and can be used to send REST API commands to the Element software. Curl provides a practical solution for scripting complex and otherwise time-consuming administrative tasks.

Another direct method for accessing the Element API is with PowerShell and the SolidFire Core module installed. This allows PowerShell administrators a comprehensive set of easy-to-use CLI commands that simplify management and provide powerful automation features.

This document provides example `curl` commands for common administrative tasks.

Solution Architecture Overview

The solution presented here features a high-performance SQL Server database design for online transaction processing. Key architecture points include:

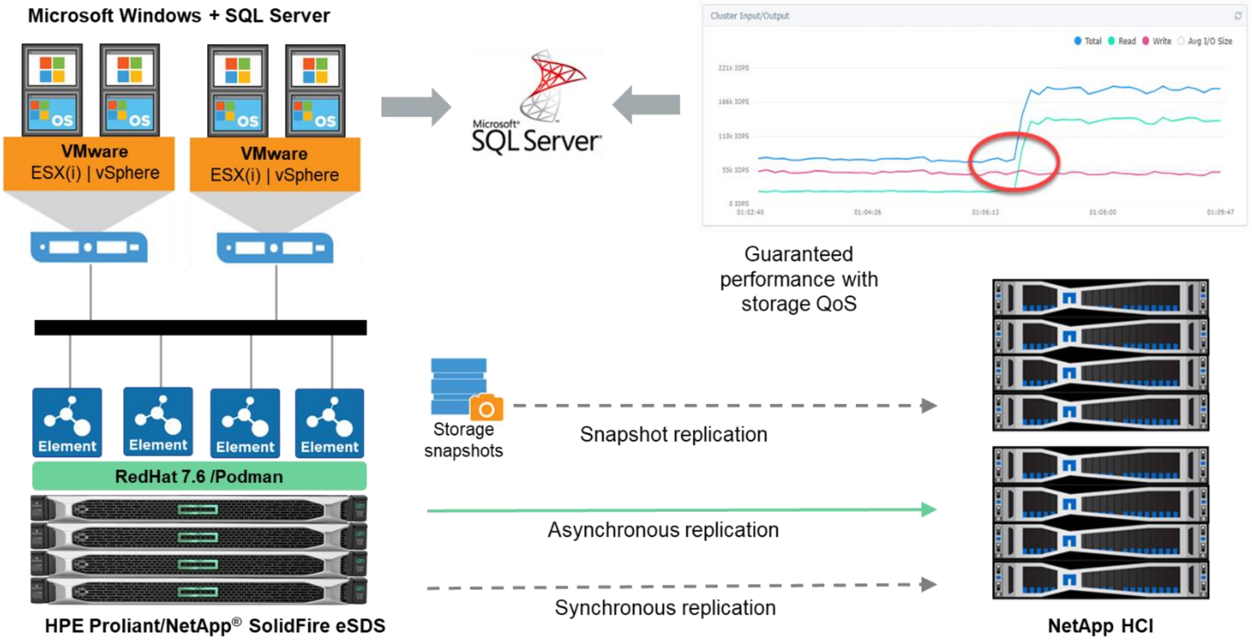
- Database components are laid out across multiple SolidFire eSDS volumes.
- A NetApp HCI cluster is used as a replication target for the SQL database and transaction log.
- QoS is used to guarantee performance for the SQL volumes as additional simulated application workloads are introduced to validate the design.

Architecture Diagram

For this solution deployment, we installed eight Microsoft SQL Server 2017 instances. To implement this, eight Microsoft Windows 2016 Server virtual machines (VMs) were first deployed on our VMware ESXi cluster. We then installed Microsoft SQL Server 2017 and the SQL Studio Management 2018 Tools on each VM.

Figure 3 illustrates the building blocks of the solution as well as replication to a separate NetApp HCI cluster.

Figure 3) Solution architecture diagram.



Hardware Requirements

Table 1 lists the hardware components that were used to implement the solution. The hardware components that are used in any particular implementation of the solution can vary based on business requirements.

Table 1) Hardware requirements.

Hardware	Quantity
HPE Proliant DL360 (SolidFire eSDS node)	4
NetApp HCI H410S storage node	4
NetApp HCI H410C compute node	4
Cisco Nexus 31108PC-V	2

Table 2 lists each of the SolidFire eSDS storage node specifications .

Table 2) SolidFire eSDS storage node specifications.

HPE/SolidFire eSDS Component	Description
Platform	HPE DL360 Gen 10
Processors	2x Intel Xeon Gold 5218 (2.3GHz/16 Cores)
Memory	24x 32GB DDR4 memory
Write cache	1x 375GB NVMe SSD
Capacity drives	9x 3.84TB NVMe SSD
Storage network interface card (NIC)	10/25Gb dual-port
Management NIC	1/10Gb NIC
Boot drives	2x M.2 240GB
Performance	100K 4K IOPS

HPE/SolidFire eSDS Component	Description
Operating system	RedHat Enterprise Linux 7.6
NetApp Element software	12.2

Software Requirements

Table 3 lists the software components that were used to implement the solution. The software components that are used in any particular implementation of the solution can vary based on business requirements.

Table 3) Software requirements.

Software	Version
VMware vSphere ESXi	ESXi 6.7 U3
Microsoft Windows Server	Standard Edition 2016
Microsoft SQL Server	SQL Server 2017
NetApp SolidFire management node (mNode)	12.2
NetApp HCI/Element software (replication target)	11.7

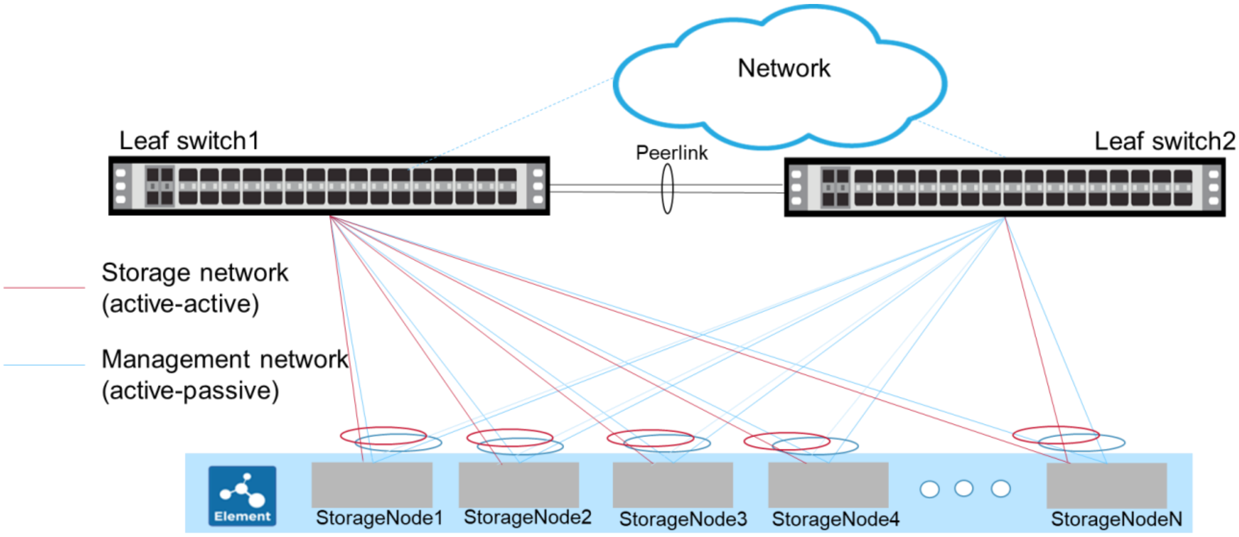
Design Considerations

Network Design

Redundant network connectivity is an important consideration for any storage design. NetApp recommends active-passive for the management interfaces and active-active for the storage interfaces.

Figure 4 illustrates a typical configuration with redundant network connections for both the management and network interfaces.

Figure 4) SolidFire eSDS network design example.



Compute Design

The compute hardware for running VMware vSphere ESXi must meet the requirements of the specific SQL Server database design. It must take into account the ability to scale to meet future growth.

To access storage resources on the SolidFire eSDS cluster, the VMware hypervisor and VMs must have adapters configured to communicate on the same iSCSI storage network as the SolidFire eSDS nodes.

For the solution outlined in this document, we used the VM settings (Table 4) in ESXi for each SQL Server instance.

Table 4) SQL Server VM settings.

SQL Server VM Settings	Quantity
vCPUs	8
Memory	16GB
OS hard drive	40GB
Virtual adapter – Intel 82574L (management network)	1
Virtual adapter – VMXNET3 (storage network)	4

Storage Design

Table 5 lists the four separate storage volumes that were assigned to each SQL Server instance to host the databases and transaction log.

Table 5) SQL Server storage volumes.

Content	Size	Drive Location / Name
SQL Server user database files 1	1000GB	`E: / MSSQL_DATA1`
SQL Server user database Files 2	1000GB	`F: / MSSQL_DATA2`
SQL Server transaction log	100GB	`G: / MSSQL_LOG`
SQL Server tempdb database files	100GB	`H: / MSSQL_TEMP`

The basic database storage design has the following characteristics:

- It does not use SQL Server partitions beyond the default configuration.
- It uses a dedicated volume and LUN for the SQL Server tempdb database, which can help improve its ability to handle higher I/O requirements.
- It uses a dedicated volume and LUN for the SQL Server transaction log, also for improved performance.
- It uses a dedicated volume and LUN for each user database. In this solution, the user database is split across two LUNs to provide additional concurrent operations and IOPS capability.

Depending on your business needs, you can run your SQL Server databases on VMs with VMware ESXi Server or Hyper-V Server. The following provisioning formats are supported:

- VMware ESXi Server with iSCSI, raw device mapping (RDM), virtual machine file system (VMFS), and virtual machine's disk (VMDK)
- Hyper-V Server with fixed, dynamic, and pass-through disks

Best Practices for QoS

This section describes a few key considerations for storage volumes hosting high priority databases.

iSCSI Service Distribution

Based on the QoS minimum IOPS settings, Element software distributes iSCSI sessions for volumes across storage nodes in order to ensure efficient resource allocation across the cluster. Therefore, it is important to not set minimum IOPS too low in most cases. Setting the minimum IOPS too low can result in multiple volumes being hosted from a single storage node, therefore an uneven distribution of iSCSI sessions across the storage nodes in a cluster. Be sure to provide some overhead to the most critical storage volumes.

Working with Minimum, Maximum, and Burst IOPS

The MaxIOPS setting can be used to limit the I/O rate of a given volume, but this setting is not otherwise used to prioritize workloads on SolidFire devices. MinIOPS is the sole setting that is used to prioritize workloads that are not otherwise limited by the MaxIOPS setting.

The MaxIOPS setting also alters the maximum bandwidth permitted for a given volume. Setting MaxIOPS very high, such as 200k IOPS, can help to increase performance for large block workloads that require high throughput rates. For example, database backup operations.

The following general recommendations help to illustrate this concept. This information is situation dependent and the settings will need to be adjusted based on the needs of the specific application.

Table 6 provides a list of recommended QoS settings for SQL Server database volumes.

Table 6) QoS recommendations for SQL Server volumes.

SQL Server Workload and Database Size	MinIOPS QoS Setting	MaxIOPS QoS Setting	BurstIOPS QoS Setting
• Light workloads • Small databases	5,000	200,000	200,000
• Moderate workloads • Medium sized databases	10,000	200,000	200,000
• Heavy workloads • Large databases	15,000	200,000	200,000

Deploying SQL Server on SolidFire eSDS

Deploying the solution involves the following tasks:

1. Prepare eSDS storage nodes
2. Create the SolidFire cluster.
3. Deploy the SolidFire management node.
4. Create the storage volumes in the NetApp Element UI or through the API and provide access to each SQL Server.
5. Create the SQL Server databases.
6. Configure Element-to-Element replication.

   Note: The actual deployment of the eSDS storage nodes is beyond the scope of this document. For more information see, SolidFire and Element 12.2 Documentation Center.

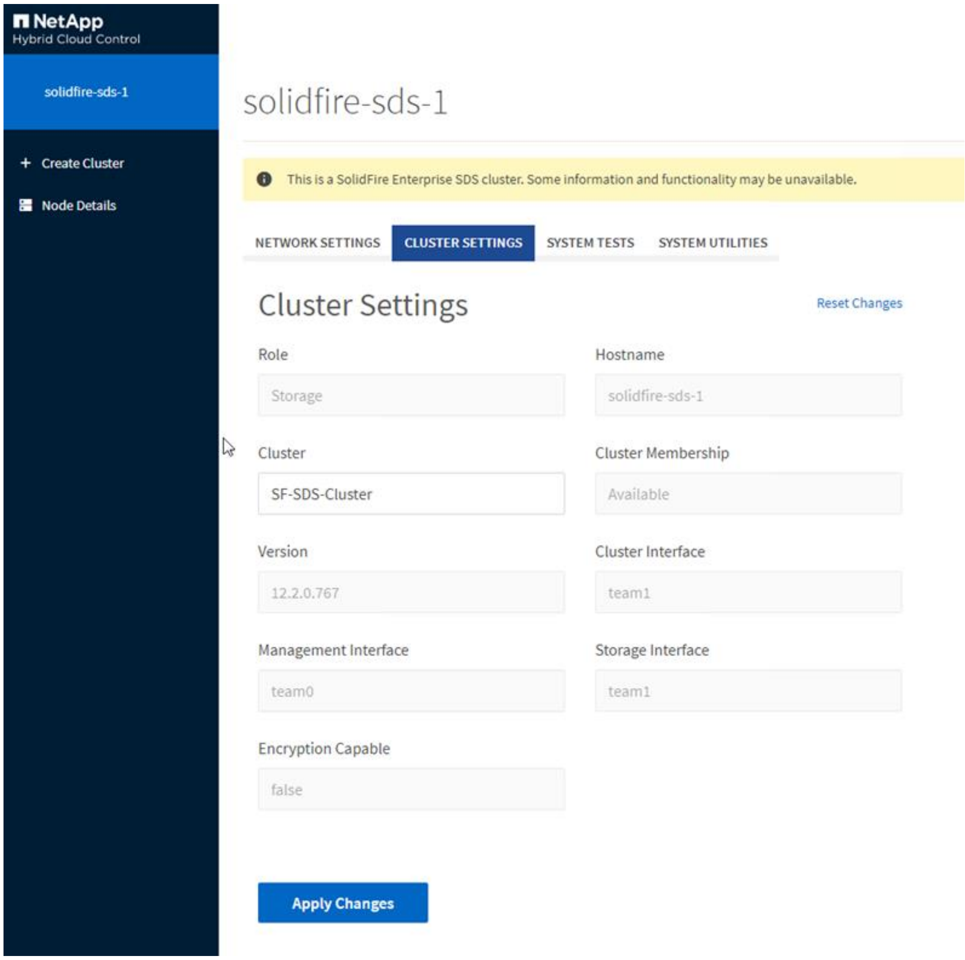
Create SolidFire eSDS Cluster

After the storage nodes are up and running, it is necessary to set the cluster name on each node and then create the storage cluster. Finally, you will need to add the drives to the newly created storage cluster.

To create the cluster, complete the following steps:

1. In a web browser, navigate to the management IP address of each storage node. The Hybrid Cloud Control dialog box is displayed.

2. Select Cluster Settings and provide a name for the cluster.



solidfire-sds-1

This is a SolidFire Enterprise SDS cluster. Some information and functionality may be unavailable.

Cluster Settings

Role: Storage

Hostname: solidfire-sds-1

Cluster: SF-SDS-Cluster

Cluster Membership: Available

Version: 12.2.0.767

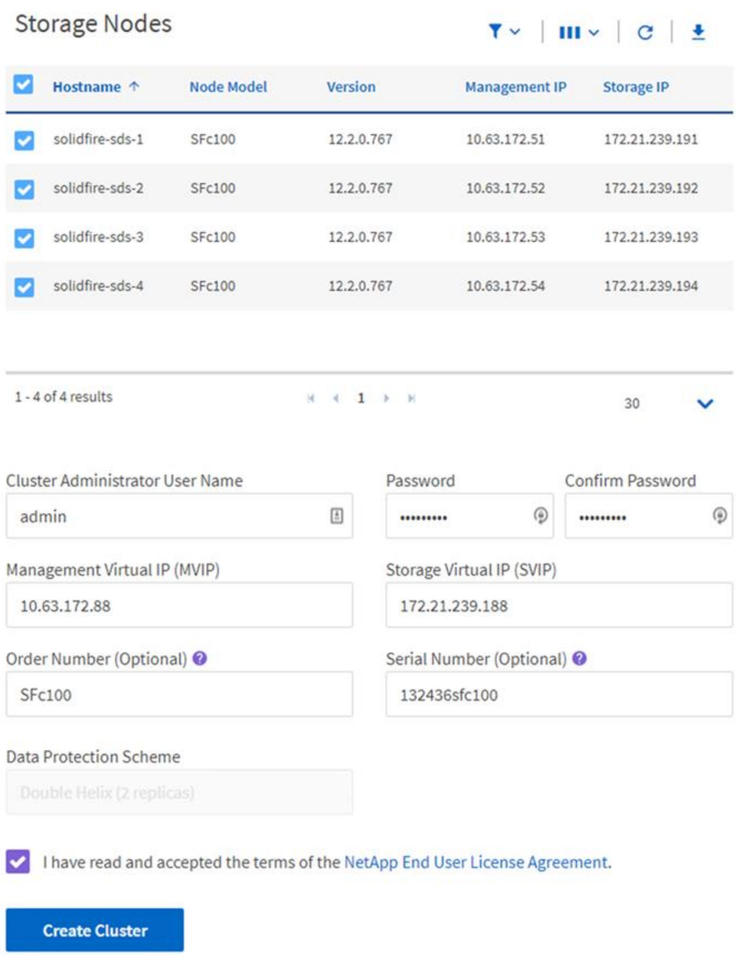
Cluster Interface: team1

Management Interface: team0

Storage Interface: team1

Encryption Capable: false

3. Select Apply Changes to save the new name.
4. After the cluster name has been set on each node, navigate to Create Cluster, select all nodes to be included in the cluster, and fill out all the required information. This information includes an administrator account user name and password, and the virtual IP addresses for both the management and storage networks.



Storage Nodes

☑	Hostname ↑	Node Model	Version	Management IP	Storage IP
☑	solidfire-sds-1	SFc100	12.2.0.767	10.63.172.51	172.21.239.191
☑	solidfire-sds-2	SFc100	12.2.0.767	10.63.172.52	172.21.239.192
☑	solidfire-sds-3	SFc100	12.2.0.767	10.63.172.53	172.21.239.193
☑	solidfire-sds-4	SFc100	12.2.0.767	10.63.172.54	172.21.239.194

1 - 4 of 4 results

Cluster Administrator User Name: admin

Password: •••••••••

Confirm Password: •••••••••

Management Virtual IP (MVIP): 10.63.172.88

Storage Virtual IP (SVIP): 172.21.239.188

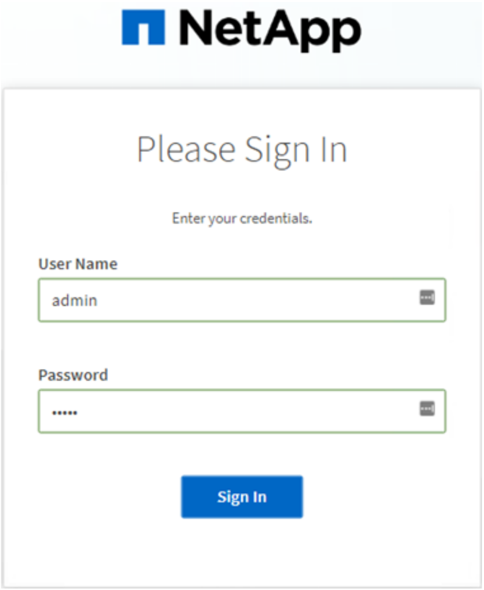
Order Number (Optional): SFc100

Serial Number (Optional): 132436sfc100

Data Protection Scheme: Double Helix (2 replicas)

☑ I have read and accepted the terms of the NetApp End User License Agreement.

5. Select Create Cluster to form the new cluster.
6. In a web browser, enter the IP address of management virtual IP address (MVIP) and log in to Element manager.



Please Sign In

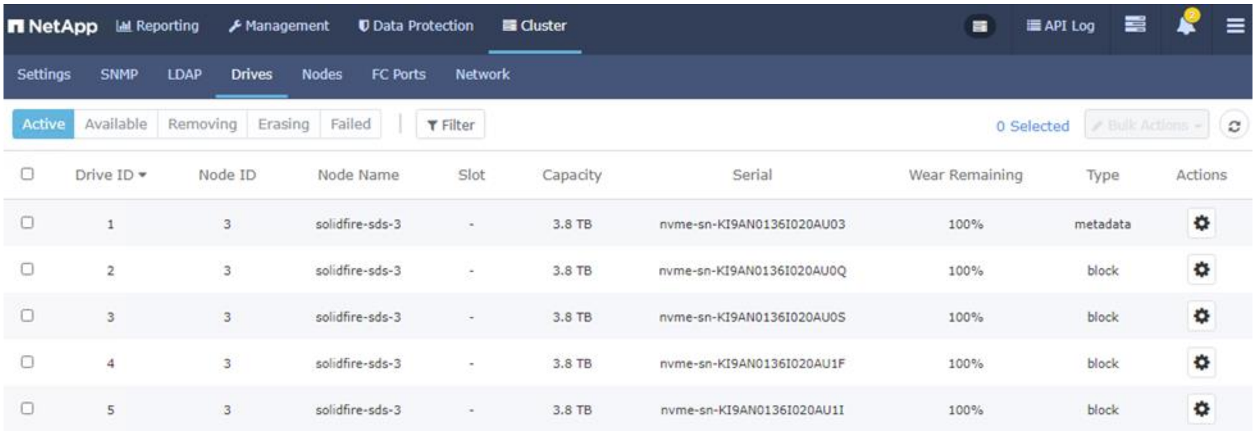
Enter your credentials.

User Name: admin

Password: •••••

7. Navigate to the Cluster tab and then Drives. Drives can be in Available, Active, Removing, Erasing, or Failed states. Select All Available Drives, go to bulk actions, and select Add to Cluster.

   All the available drives in the cluster should now be viewable from the Active state list.



□	Drive ID	Node ID	Node Name	Slot	Capacity	Serial	Wear Remaining	Type	Actions
□	1	3	solidfire-sds-3	-	3.8 TB	nvme-sn-KI9AN0136I020AU03	100%	metadata	⚙
□	2	3	solidfire-sds-3	-	3.8 TB	nvme-sn-KI9AN0136I020AU0Q	100%	block	⚙
□	3	3	solidfire-sds-3	-	3.8 TB	nvme-sn-KI9AN0136I020AU0S	100%	block	⚙
□	4	3	solidfire-sds-3	-	3.8 TB	nvme-sn-KI9AN0136I020AU1F	100%	block	⚙
□	5	3	solidfire-sds-3	-	3.8 TB	nvme-sn-KI9AN0136I020AU1I	100%	block	⚙

Alternately, you can use the Element API to set the individual node names and to create the cluster. The following format can be used with the `curl` command to send the appropriate API calls.

1. Set the cluster name on each storage node.

```
curl --request POST \
--url https://<node_mip>:442/json-rpc/12.0 \
--header 'content-type: application/json' \
--data '{
"method": "SetClusterConfig",
"params": {
"cluster": {"cluster": "<Cluster-Name>"}
}
}'
```

2. Create the storage cluster.

```
curl --request POST \
--url https://<ANY_NODE-MIP>/json-rpc/12.2\
--header 'content-type: application/json' \
--data '{
"method": "CreateCluster",
"params": {
"mvip":"<MVIP>",
   "svip":"<SVIP>",
   "serialNumber":"<Serial-Number>",
   "orderNumber":"<Order-Number>",
   "repCount":2,
   "username":"<Username>",
   "password":"<Password>",
   "nodes":["<SIP1>", "<SIP2>", "<SIP3>", "<SIP4>"],
   "acceptEula":true
    }
   }'
```

3. Add the drives to the cluster.

```
curl -u <username:password> --request POST \
--url https://<ANY_NODE-MIP>/json-rpc/12.2\
--header 'content-type: application/json' \
--data '{
   "method": "AddDrives",
   "params": {
   "drives" : [
   {
```

```
"driveID": "<ID#>"
},
{
"driveID": "<ID#>"
},
{
"driveID": "<ID#>"
    }
   ]
  }
}'
```

Deploy the SolidFire Management Node

The SolidFire mNode is used for system monitoring and troubleshooting. With the mNode, you can upgrade system services, manage cluster assets and settings, run system tests and utilities, configure Active IQ for system monitoring, and enable NetApp Support access for troubleshooting.

The mNode is downloadable as an ISO or OVA and is manually installed. It must be able to communicate on the SolidFire management network.

For a complete set of instructions on installing and configuring the SolidFire mNode, see the SolidFire and Element 12.2 Documentation Center.

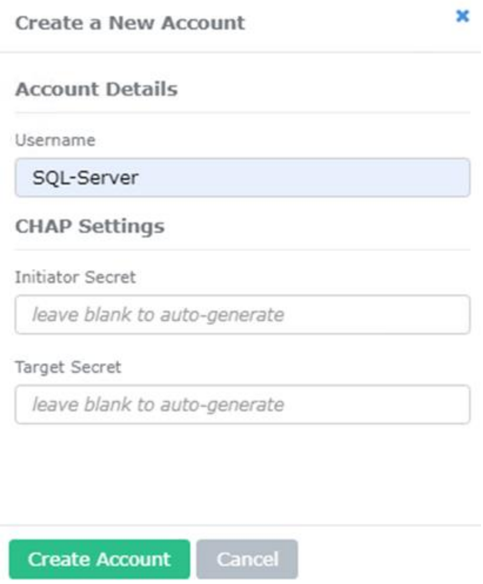
Configure Storage for SQL Server Databases

The tasks described in this section must be completed on the SolidFire eSDS cluster in order to assign storage to each SQL Server:

Configure an Account

To create an account in which the SQL Server storage volumes will reside, complete the following steps:

1. In the NetApp Element GUI, select Management > Accounts. The Account List screen dialog box displayed.
2. Click Create Account. The Create a New Account dialog box is displayed.



Create a New Account

Account Details

Username: SQL-Server

CHAP Settings

Initiator Secret: leave blank to auto-generate

Target Secret: leave blank to auto-generate

3. Enter a user name.

4. In the CHAP Settings section, enter the following information:
    a. Initiator Secret for CHAP node session authentication
    b. Target Secret for CHAP node sessions authentication

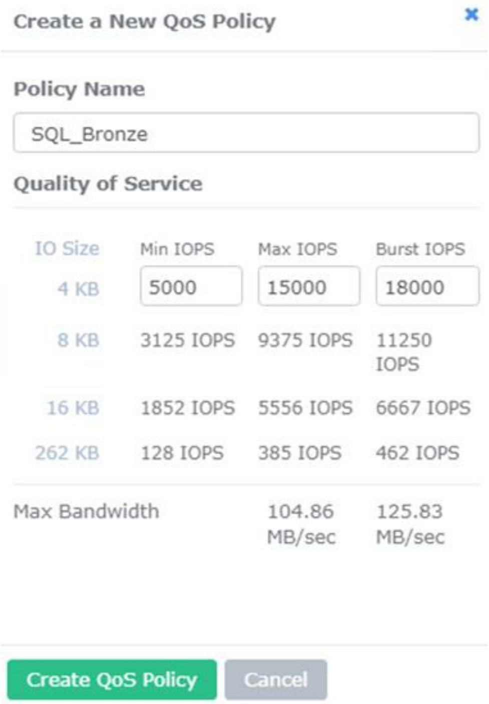
Alternately, the Element API (with the `curl` command) can be used to create an account.

```
curl --request POST \
--url https://<ANY_NODE-MIP>/json-rpc/12.2\
--header 'content-type: application/json' \
--data '{
"method": "AddAccount",
”params": {
"username": "<username>”,
"initiatorSecret": "<CHAP-Secret>",
"targetSecret": "<CHAP-Secret>"
    }
   }'
```

Create a QoS Policy

To create a QoS policy, complete the following steps:

1. In the Element GUI, select Management > QoS Policies.
2. Click Create QoS Policy. The Create a New QoS Policy dialog box is displayed.
3. Enter the QoS policy parameters as required.



Policy Name

SQL_Bronze

Quality of Service

IO Size	Min IOPS	Max IOPS	Burst IOPS
4 KB	5000	15000	18000
8 KB	3125 IOPS	9375 IOPS	11250 IOPS
16 KB	1852 IOPS	5556 IOPS	6667 IOPS
262 KB	128 IOPS	385 IOPS	462 IOPS
Max Bandwidth		104.86 MB/sec	125.83 MB/sec

Alternately, use the Element API (with the `curl` command) to create a QoS Policy.

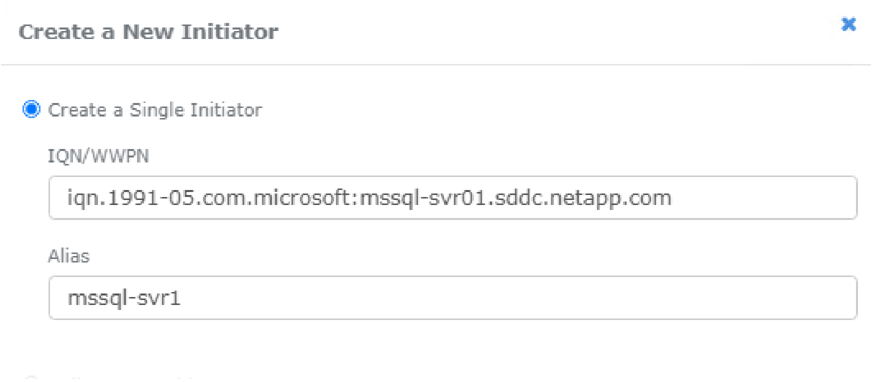
```
curl --request POST \
--url https://<ANY_NODE-MIP>/json-rpc/12.2\
--header 'content-type: application/json' \
--data '{
"method": "CreateQoSPolicy",
"params": {
"name": "<QoS-Policy-Name>",
"qos": {
```

```
"minIOPS": <MinIOPS>,
"maxIOPS": <MaxIOPS,
"burstIOPS": <BurstIOPS>
    }
   }'
```

Create an Initiator

To create an initiator, complete the following steps:

1. From the Element GUI, select Management > Initiators.
2. Click Create Initiator. The Create a New Initiator dialog box is displayed.
3. Enter the IQN name from the appropriate SQL Server.
4. Enter an Alias for the Initiator.



Create a New Initiator

● Create a Single Initiator

IQN/WWPN

iqn.1991-05.com.microsoft:mssql-svr01.sddc.netapp.com

Alias

mssql-svr1

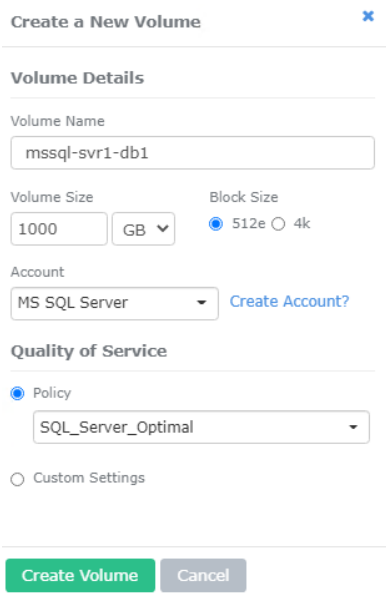
Alternately, the Element API can be used using curl to create the Initiator.

```
curl --request POST \
--url https://<ANY_NODE-MIP>/json-rpc/12.2\
--header 'content-type: application/json' \
--data '{
"method": "Create-Initiators",
"params": {
"initiators": [
{
"name": "<IQN-Initiator-Name>”,
"alias": "<Alias-Name>"
         }
       ]
     }
   }’
```

Create a Volume

To create a volume, complete the following steps:

1. In the Element GUI, select Management > Volumes. The Volumes List dialog box is displayed.
2. Click Create Volume. The Create a New Volume dialog box is displayed opens.


Create a New Volume

Volume Details

Volume Name: mssql-svr1-db1

Volume Size: 1000 GB

Block Size: ● 512e ○ 4k

Account: MS SQL Server

Quality of Service

● Policy: SQL_Server_Optimal

○ Custom Settings

3. Configure the following information:
   a. Enter the volume name (1 to 64 characters in length); for example, enter the name `mssql-svr1-data1`.
   b. Enter the size of the volume.
   c. Enter the account name for the volume.
   d. Click the Account drop-down menu and select the account that should have access to the volume.
   e. In Quality of Service options, select Policy and then select the QoS policy to be applied to the volume.
   f. Click Create Volume.
4. Repeat steps 1 through 3 for all volumes that are part of the SQL Server instance.

Alternately, the Element API can be used (with the `curl` command) to create the volumes.

```
curl --request POST \
--url https://<ANY_NODE-MIP>/json-rpc/12.2\
--header 'content-type: application/json' \
--data '{
"method": "CreateVolume",
"params": {
"name": "<Volume-Name>",
"accountID": <ID#>,
"totalSize": <Size-in-Bytes>,
"enable512e": <True/False>,
"qosPolicyID": <ID#>
    }
   }’
```

Create an Access Group

Access groups provide a security mechanism that allows specified volumes to be accessed by a specific iSCSI initiator; whereas CHAP authentication uses secret keys to validate the identity of the remote client.

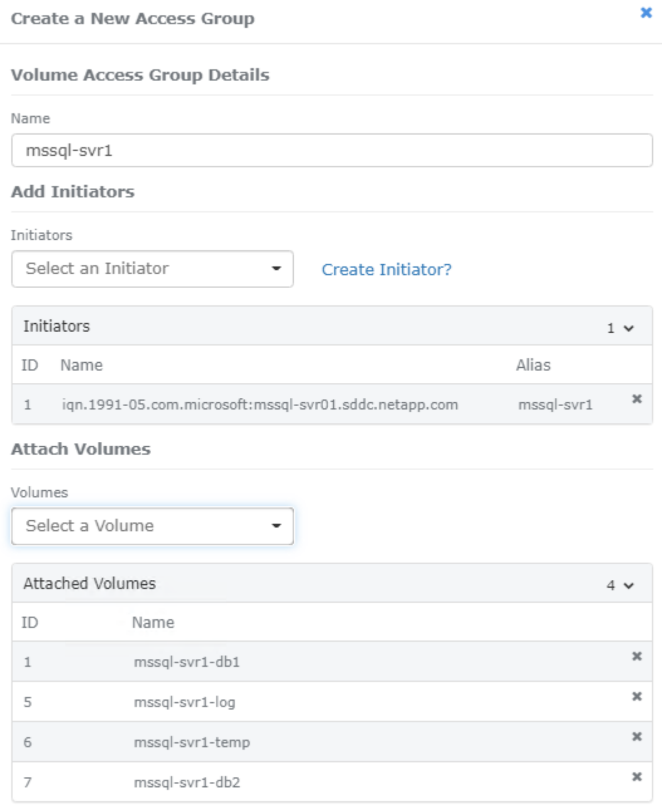
For this solution, we used access groups for authentication. Access groups use the IQN identifier of each SQL Server.

Volume access groups have the following system limits:

- A maximum of 64 IQNs.
- An IQN can belong to only one access group.
- A single volume can belong to a maximum of four access groups.

To create an access group, complete the following steps:

1. From the Element UI, select Management > Access Groups. The Access Groups List dialog box is displayed.
2. Click Create Access Group. The Create a New Access Group dialog box is displayed.



Create a New Access Group

Volume Access Group Details

Name: mssql-svr1

Add Initiators

Initiators: Select an Initiator

Create Initiator?

Initiators 1

ID	Name	Alias
1	iqn.1991-05.com.microsoft:mssql-svr01.sddc.netapp.com	mssql-svr1

Attach Volumes

Volumes: Select a Volume

Attached Volumes 4

ID	Name
1	mssql-svr1-db1
5	mssql-svr1-log
6	mssql-svr1-temp
7	mssql-svr1-db2

3. Enter a name for the volume access group.
4. Select the IQN from the initiator drop-down or click Create Initiator.
5. Under the Attach Volumes section, select the volumes to be mapped from the drop-down list.
6. Click Create Access Group.

The SQL Server database volumes are now listed as part of the selected volume access group and are ready to be mapped to the host operating system.

Alternately, the Element API can be used with the `curl` command to create an Access Group.

```
curl --request POST \
--url https://<ANY_NODE-MIP>/json-rpc/12.2\
--header 'content-type: application/json' \
--data '{
"method": "CreateVolumeAccessGroup",
"params": {
"name": "“<Access-Group-Name>",
"initiators": [“<IQN-Names>"],
"volumes": [<ID#’s>]
 }
}
```

For additional best practices on deploying Microsoft SQL Server on SolidFire arrays, see TR-4609: SQL Server Best Practices on NetApp SolidFire.

Configure Storage on Microsoft Windows

The following is a summary of the configuration steps to configure storage on each SQL Server VM. For detailed step-by-step instructions for configuring storage on Microsoft Windows Server, see TR-4609: SQL Server Best Practices on NetApp SolidFire.

1. Enable Jumbo Frames on the storage adapters.
2. Enable the Microsoft iSCSI Service.
3. Enable Microsoft Multipath I/O (MPIO).
4. Configure multiple iSCSI sessions for each SolidFire volume:
    a. Open the iSCSI Initiator utility.
    b. From the Discovery tab, add the Storage Virtual IP Address (SVIP) of the SolidFire eSDS cluster as a Target Portal to discover the assigned storage volumes.
    c. From the Targets tab, connect once to each storage volume for each storage adapter that will participate in MPIO.
    d. From the Volumes and Devices tab, click Auto Configure to ensure that the database file systems are available on boot when the SQL Server service starts.
5. Create file systems for the SQL Server databases.

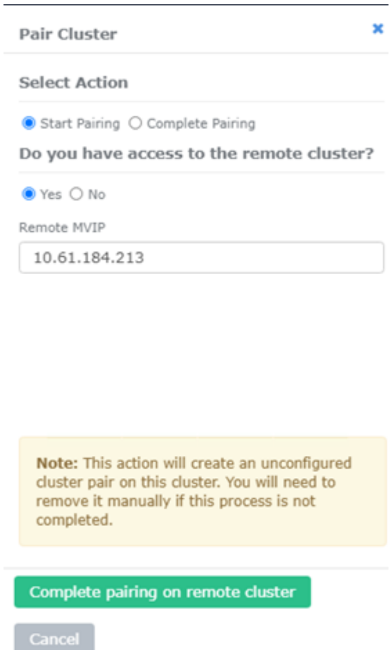
Configure Replication

Cluster pairing between the target and source SolidFire clusters is required before volume replication. All nodes on the source and target clusters must be able to communicate on management and storage networks. The user must have administrative privileges for the respective source and target clusters.

Cluster Pairing

To pair a cluster, complete the following steps:

1. In the Element GUI, select Data Protection > Cluster Pairs. The Cluster Pairs List dialog box is displayed.
2. Click Pair Cluster. The Pair Cluster dialog box is displayed.



Pair Cluster

Select Action

● Start Pairing ○ Complete Pairing

Do you have access to the remote cluster?

● Yes ○ No

Remote MVIP: 10.61.184.213

Note: This action will create an unconfigured cluster pair on this cluster. You will need to remove it manually if this process is not completed.

3. Select the Start Pairing option.
4. For the Do You Have Access to the Remote Cluster option, select Yes.
5. Click Complete Pairing on Remote Cluster. A webpage for the remote cluster opens automatically.
6. On the remote cluster, click Complete Pairing. Now your clusters should be paired.

Alternately, cluster pairing can be achieved through the Element API by using the `curl` command.

1. On the source cluster, issue the following command:

```
curl --request POST \
--url https://<ANY_NODE-MIP>/json-rpc/12.2\
--header 'content-type: application/json' \
--data
{
"method": "StartClusterPairing",
"params": {}
}
```

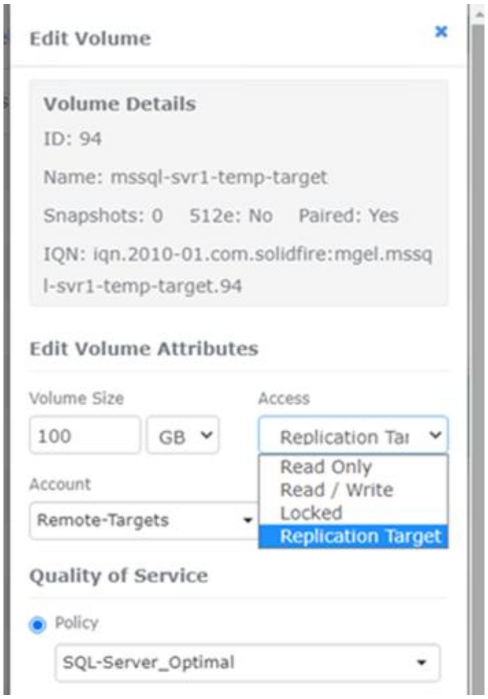
2. To complete the cluster pairing, run the following `curl` command on the target cluster. A volume pairing key is returned when the `StartClusterPairing` method completes and must be entered in the `clusterPairingKey` field.

```
curl --request POST \
--url https://<ANY_NODE-MIP>/json-rpc/12.2\
--header 'content-type: application/json' \
--data
{
        "id": 49,
        "method": "CompleteClusterPairing",
        "params": {
        "clusterPairingKey": "<Cluster_Key>"
         }
}
```

Volume Pairing

After the cluster pairing is complete, configure the volume pairing to begin the replication process. This example assumes you have administrative access to both clusters. If you only have access to one cluster, you must pair volumes by using a pairing key. For more information, see the [SolidFire and Element 12.2 Documentation Center]().

1. Create a volume on the target cluster to act as the target of the replicated volume.
2. In the target cluster Element GUI, select Management > Volumes.
3. Select the target volume > Edit and then change access for the volume to Replication Target.



Edit Volume

Volume Details

ID: 94

Name: mssql-svr1-temp-target

Snapshots: 0   512e: No   Paired: Yes

IQN: iqn.2010-01.com.solidfire:mgel.mssql-svr1-temp-target.94

Edit Volume Attributes

Volume Size: 100 GB

Access: Replication Tar

Read Only
Read / Write
Locked
Replication Target

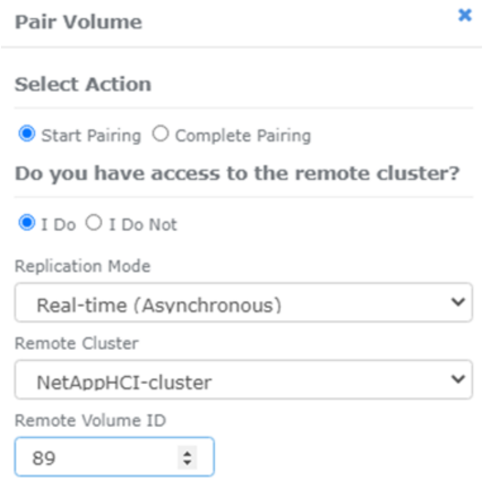
Account: Remote-Targets

Quality of Service

● Policy

SQL-Server_Optimal

4. In the Source cluster Element GUI, select Management > Volumes; a window with all volumes should display.
5. Select the volume that requires replication.
6. Select Action > Edit > Pair.
7. In the Pair Volume dialog box, select Start Pairing and then fill in the information about the target cluster and the volume ID of the target volume.



Pair Volume

Select Action

● Start Pairing ○ Complete Pairing

Do you have access to the remote cluster?

● I Do ○ I Do Not

Replication Mode: Real-time (Asynchronous)

Remote Cluster: NetAppHCI-cluster

Remote Volume ID: 89

8. Select Complete Pairing Volume on Remote Cluster. A webpage for the remote cluster opens automatically.
9. Your volumes should now be paired.

Alternately, initiating volume pairing at the source cluster can be achieved through the Element API by using the `curl` command.

```
curl --request POST \
--url https://<ANY_NODE-MIP>/json-rpc/12.2\
--header 'content-type: application/json' \
--data
{
        "id": <ID#>,
″method": "StartVolumePairing",
        "params": {
        "mode": "<Sync-Mode>",
        "volumeID": <ID#>
         }
}
```

To complete volume pairing, run the following `curl` command on the target cluster. A volume pairing key is returned when the `StartVolumePairing` method completes and must be entered in the `volumePairingKey` field.

```
curl --request POST \
--url https://<ANY_NODE-MIP>/json-rpc/12.2\
--header 'content-type: application/json' \
--data
{
"id": 57,
"method": "CompleteVolumePairing",
"params": {
"volumeID": <ID#>,
"volumePairingKey": "<Volume_Key>"
 }
}
```

Validation Results

To validate this solution, we ran a test workload on the eight SQL Server instances and introduced an additional simulated application workload using the Vdbench utility tool. In addition, we conducted a test simulating a failed storage node and observed the impact on our SQL Server workloads.

SQL Server Performance Validation

To simulate workload for SQL Server, HammerDB was used with a TPC-C profile simulating a write-intensive, online transaction load-processing workload. HammerDB is a highly scalable load testing application for a variety of enterprise databases. The load simulated 128 virtual users initiating a constant stream of continuous transactions.

We installed HammerDB on separate clients to a generate workload against our SQL Server instances.

Table 7 lists the feature settings that were used for the performance validation tests. We used the following QoS Policies for our SQL Server and Vdbench volumes.

Table 7) QoS settings for validation test.

QoS Policy Name	Min IOPS QoS Setting	Max IOPS QoS Setting	Burst IOPS QoS Setting
`SQL_Server_Optimal`	10,000	100,000	100,000
`Vdbench_Max`	2,000	12,000	12,000

For the first test, only the SQL Server workload ran. We observed expected results with approximately 10,000 IOPS per SQL Server instance with a write-intensive workload.

Multi-Workload Performance Validation

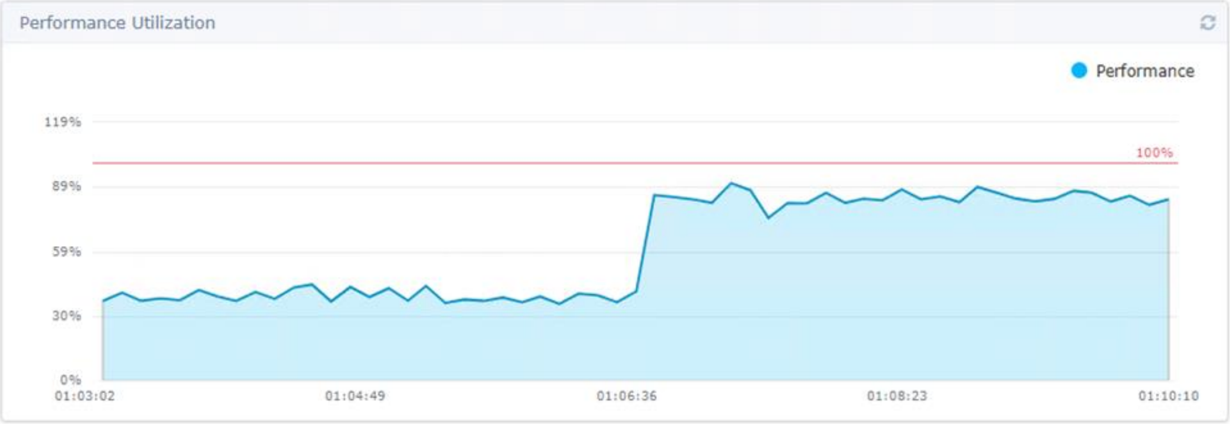
For the second test, in addition to the write-intensive SQL Server workload, we introduced an aggressive workload to the system.

We configured four additional Windows Server 2016 VMs configured with the Vdbench I/O load generation tool. Each Vdbench server VM was configured with four 100GB volumes from SolidFire eSDS cluster, for a total of 16 volumes.

The servers were remotely triggered to run an 8K block read workload that totaled more than 170K IOPS.

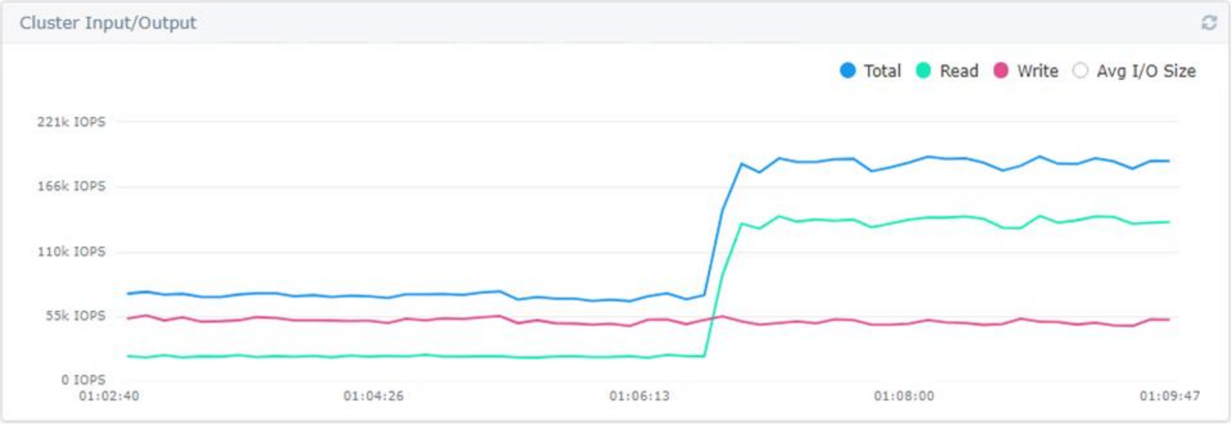
During this test, the cluster was running at approximately an 86% utilization rate, as illustrated in Figure 5. This graph is from the main Element overview page.

Figure 5) Cluster utilization during performance validation.



Because our SQL Server workload was write-intensive and Vdbench was running a read workload, we could easily view the impact of the additional I/O on the system from the Element overview page. There was virtually no impact on our SQL Server database performance. By using QoS with plenty of minimum IOPS, and ensuring that the maximum IOPS was set at a much level higher than our Vdbench volumes, we were able to easily guarantee performance for our critical SQL Server databases.

Figure 6) Cluster IOPs during performance validation.



Node Failure Validation

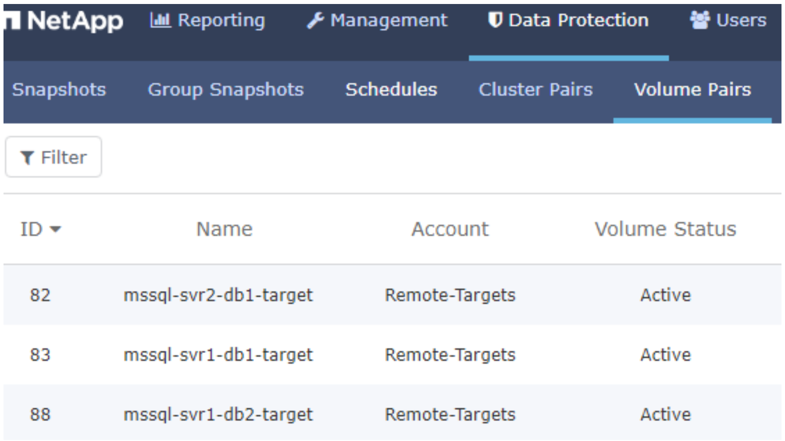
In order to test the high availability of the cluster, we simulated a catastrophic failure of one of the eSDS storage nodes in our cluster. First, we started our TPC-C online transaction processing workload by using HammerDB on all eight SQL Server instances. Then we performed a hard power off of one of our nodes and observed the effect on the cluster.

During the node failure test, the SQL Server database services remained online throughout. While those SQL Server instances that had at least one volume with iSCSI session on the affected node momentarily ceased operations, all iSCSI sessions were successfully redirected to existing storage nodes and I/O operations resumed within a short period of time.

Replication Validation

After setting up volume replication, the source and target volumes should appear in an active state. To validate the status of active volume pairs, verify the following:

1. From each cluster, select Data Protection > Volume Pairs.
2. Verify that the volume status displays as Active.



ID	Name	Account	Volume Status
82	mssql-svr2-db1-target	Remote-Targets	Active
83	mssql-svr1-db1-target	Remote-Targets	Active
88	mssql-svr1-db2-target	Remote-Targets	Active

To further verify that data replicated across clusters, we added unique content to one of our source volumes and mounted the target volume to ensure data had been replicated.

To mount the source target, complete the following steps:

1. Delete the volume pair.
2. Change the target volume access from Replication Target to Read/Write.
3. Provide access to the volume through an access group.
4. Mount the volume on the intended host system.

Conclusion

As this solution demonstrates, NetApp SolidFire eSDS provides the industry-leading storage features of SolidFire arrays in a software-defined model, allowing customers the choice of using their preferred hardware vendor. NetApp SolidFire's unique approach to QoS allows granular control on a per-volume basis, allocating the available performance resources of the storage cluster. This approach strengthens performance guarantees and provides database administrators confidence when running their workloads in a shared storage environment. NetApp SolidFire eSDS is an ideal solution for businesses looking for leading class DP, to consolidate demanding workloads, and to provide flexible data management for test and development environments.

Where to Find Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites:

- SolidFire product documentation
  https://docs.netapp.com/sfe-122/index.jsp
- Element API User Guide
  https://docs.netapp.com/sfe-122/topic/com.netapp.doc.sfe-api/home.html?cp=5_2
- SQL Server Best Practices on NetApp SolidFire
  https://www.netapp.com/media/10514-tr4609pdf.pdf
- NetApp Element Software Remote Replication
  https://www.netapp.com/pdf.html?item=/media/10607-tr4741pdf.pdf
- ESG Technical Validation: Guaranteeing Mixed-workload Performance with NetApp HCI
  https://www.esg-global.com/validation/esg-technical-validation-guaranteeing-mixed-workload-performance-with-netapp-hci

Version History

Version	Date	Document Version History
Version 1.0	October 20XX	Initial release.

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright Information

Trademark Information

TR-4866-1020