Abstract
Dell EMC™ PowerScale™ SyncIQ™ is an application that enables the flexible management and automation of data replication. This white paper describes the key features, architecture, and considerations for SyncIQ.

November 2020
## Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2018</td>
<td>Updated to Dell EMC template</td>
</tr>
<tr>
<td>March 2019</td>
<td>Completely re-written and updated</td>
</tr>
<tr>
<td>April 2019</td>
<td>Updated for OneFS 8.2. Added SyncIQ encryption and bandwidth reservation</td>
</tr>
<tr>
<td>August 2019</td>
<td>Added section for SyncIQ requiring System Access Zone</td>
</tr>
<tr>
<td>August 2019</td>
<td>Added section for ‘Source and target cluster replication performance. Updated SyncIQ worker calculations</td>
</tr>
<tr>
<td>October 2019</td>
<td>Other minor updates</td>
</tr>
<tr>
<td>October 2019</td>
<td>Added “Small File Storage Efficiency and SyncIQ” section. Updated SyncIQ Encryption X.509 certificate details.</td>
</tr>
<tr>
<td>January 2020</td>
<td>Updated for OneFS 8.2.2: Added section for 16 TiB SyncIQ implications and added Data Reduction section.</td>
</tr>
<tr>
<td>April 2020</td>
<td>Moved ‘SyncIQ password’ and ‘SyncIQ encryption’ sections under new ‘SyncIQ security’ section. Updated SyncIQ encryption section. Added SyncIQ encryption with self-signed certificates to Appendix.</td>
</tr>
<tr>
<td>May 2020</td>
<td>Updated ‘Isilon’ branding to ‘PowerScale’. Added ‘PowerScale’ nodes to ‘SyncIQ replication and data reduction’ section.</td>
</tr>
<tr>
<td>June 2020</td>
<td>Added ‘Target cluster Snapshot Alias’ section. Updated ‘Cascaded’ replication and ‘Whenever the source is modified’ sections.</td>
</tr>
<tr>
<td>August 2020</td>
<td>Updated ‘Running a SyncIQ job’ section with minor notes.</td>
</tr>
<tr>
<td>September 2020</td>
<td>Updated ‘SyncIQ encryption’ section for OneFS release 9.1.</td>
</tr>
<tr>
<td>November 2020</td>
<td>Updated ‘Cascaded’ and ‘Custom’ deployment topologies and added corresponding appendices.</td>
</tr>
</tbody>
</table>

## Acknowledgements

Author: Aqib Kazi

The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

Copyright © 2020 Dell Inc. or its subsidiaries. All Rights Reserved. Dell Technologies, Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners. [11/2/2020] [Technical White Paper] [H8224.17]
# Table of contents

- **Revisions** .................................................................................................................. 2
- **Acknowledgements** .................................................................................................... 2
- **Table of contents** ......................................................................................................... 3
- **Executive summary** ....................................................................................................... 7
- **Note to readers** ............................................................................................................. 7

1. **Introduction** ................................................................................................................ 8

2. **Deployment topologies** ............................................................................................... 9
   - 2.1 One-to-one .................................................................................................................. 9
   - 2.2 One-to-many .............................................................................................................. 9
   - 2.3 Many-to-one ............................................................................................................ 10
   - 2.4 Local target ............................................................................................................. 10
   - 2.5 Cascaded .................................................................................................................. 11
   - 2.6 Custom .................................................................................................................... 12

3. **Use cases** ................................................................................................................... 13
   - 3.1 Disaster recovery ..................................................................................................... 13
   - 3.2 Business continuance .............................................................................................. 13
   - 3.3 Disk-to-disk backup and restore .............................................................................. 14
   - 3.4 Remote archive ........................................................................................................ 14

4. **Architecture and processes** ......................................................................................... 15
   - 4.1 Asynchronous source-based replication .................................................................. 16
   - 4.2 Source cluster snapshot integration ....................................................................... 16
   - 4.2.1 Snapshot integration alleviates treewalks ............................................................ 17
   - 4.3 Processes .................................................................................................................. 18
   - 4.3.1 Scheduler ............................................................................................................. 18
   - 4.3.2 Coordinator ......................................................................................................... 18
   - 4.3.3 Primary and secondary workers ......................................................................... 18
   - 4.3.4 Target monitor ..................................................................................................... 19

5. **Data replication** ........................................................................................................... 20
   - 5.1 Initial replication ....................................................................................................... 20
   - 5.2 Incremental replication ............................................................................................. 21
   - 5.3 Differential replication or target aware sync ............................................................ 21

6. **Configuring a SyncIQ policy** ....................................................................................... 23
   - 6.1 Naming and enabling a policy .................................................................................. 23
   - 6.2 Synchronization and copy policies .......................................................................... 24
# Table of contents

6.3 Running a SyncIQ job .................................................................................................................. 25  
6.3.1 Manually ................................................................................................................................. 25  
6.3.2 On a schedule ......................................................................................................................... 26  
6.3.3 Whenever the source is modified ......................................................................................... 27  
6.3.4 Whenever a snapshot of the source directory is taken ....................................................... 29  
6.4 Source cluster directory ............................................................................................................. 30  
6.5 File matching criteria ............................................................................................................... 31  
6.6 Restricting SyncIQ source nodes .............................................................................................. 32  
6.7 Target host and directory ......................................................................................................... 33  
6.7.1 Target cluster SmartConnect zones ..................................................................................... 33  
6.8 Target snapshots ....................................................................................................................... 34  
6.8.1 Target cluster Snapshot Alias ............................................................................................. 35  
6.9 Advanced settings ..................................................................................................................... 37  
6.9.1 Priority .................................................................................................................................. 37  
6.9.2 Log Level ............................................................................................................................... 38  
6.9.3 Validate file integrity ............................................................................................................ 38  
6.9.4 Prepare policy for accelerated failback performance ......................................................... 38  
6.9.5 Keep reports duration ........................................................................................................... 39  
6.9.6 Record deletions on synchronization .................................................................................... 39  
6.9.7 Deep copy for CloudPools ................................................................................................. 39  
6.10 Assess sync ............................................................................................................................... 39  
7 Impacts of modifying SyncIQ policies ............................................................................................ 41  
8 SyncIQ performance rules ............................................................................................................ 42  
9 SnapshotIQ and SyncIQ ................................................................................................................ 44  
9.1 Specifying snapshots for replication ........................................................................................ 44  
9.2 Archiving SnapshotIQ snapshots to a backup cluster ............................................................ 45  
9.3 Target cluster SnapshotIQ snapshots ..................................................................................... 45  
10 SyncIQ design considerations .................................................................................................... 46  
10.1 Considering cluster resources with data replication ............................................................. 46  
10.1.1 Source and target cluster replication performance .......................................................... 47  
10.2 Snapshots and SyncIQ policies ............................................................................................... 47  
10.3 Network considerations ............................................................................................................ 47  
10.3.1 SyncIQ policy requirement for System Access Zone ......................................................... 48  
10.3.2 Network ports ....................................................................................................................... 48  
10.4 Jobs targeting a single directory tree ..................................................................................... 48  
10.5 Authentication integration ....................................................................................................... 48
# Table of contents

10.6 SyncIQ and Hadoop Transparent Data Encryption .......................................................... 48
10.7 Small File Storage Efficiency (SFSE) and SyncIQ ......................................................... 49

11 Failover and failback ........................................................................................................... 50
11.1 Failover ......................................................................................................................... 50
11.1.1 Failover while a SyncIQ job is running ................................................................. 51
11.2 Target cluster dataset ................................................................................................. 51
11.3 Failback ....................................................................................................................... 51
11.3.1 Resync-prep ........................................................................................................... 51
11.3.2 Mirror policy .......................................................................................................... 51
11.3.3 Verify .................................................................................................................... 51
11.4 Allow-writes compared to break association .............................................................. 52

12 Superna Eyeglass DR Edition .......................................................................................... 54

13 SyncIQ and CloudPools ..................................................................................................... 55
13.1 CloudPools failover and failback implications ............................................................ 55
13.2 Target cluster SyncIQ and CloudPools configuration .................................................. 56
13.2.1 CloudPools configured prior to a SyncIQ policy ................................................... 56
13.2.2 CloudPools configured after a SyncIQ policy ....................................................... 56

14 SyncIQ security .................................................................................................................. 57
14.1 SyncIQ encryption ........................................................................................................ 57
14.1.1 Configuring SyncIQ encryption ............................................................................ 58
14.1.2 Other optional commands .................................................................................... 60
14.1.3 Troubleshooting .................................................................................................... 61
14.2 SyncIQ pre-shared key ................................................................................................ 61

15 SyncIQ bandwidth reservations ......................................................................................... 64
15.1 Bandwidth reservation configuration .......................................................................... 64
15.2 Bandwidth reserve ....................................................................................................... 65
15.3 Bandwidth reservation scenarios ................................................................................ 65
15.3.1 Bandwidth reservation example 1: insufficient bandwidth .................................... 66
15.3.2 Bandwidth reservation example 2: insufficient bandwidth ..................................... 67
15.3.3 Bandwidth reservation example 3: extra bandwidth available ................................ 68

16 Monitoring, alerting, reporting, and optimizing performance ........................................ 69
16.1 Policy job monitoring .................................................................................................... 69
16.2 Performance monitoring ............................................................................................... 70
16.3 Alerts ........................................................................................................................... 70
16.4 Reporting ...................................................................................................................... 70
16.5 Optimizing SyncIQ performance .................................................................................. 71
# Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.5.1 Workers and performance scalability</td>
<td>71</td>
</tr>
<tr>
<td>16.5.2 Specifying a maximum number of concurrent SyncIQ jobs</td>
<td>73</td>
</tr>
<tr>
<td>16.5.3 Performance tuning for OneFS 8.X releases</td>
<td>73</td>
</tr>
<tr>
<td>17 Administration</td>
<td>75</td>
</tr>
<tr>
<td>17.1 Role-based access control</td>
<td>75</td>
</tr>
<tr>
<td>17.2 OneFS platform API</td>
<td>75</td>
</tr>
<tr>
<td>18 SyncIQ replication and data reduction</td>
<td>76</td>
</tr>
<tr>
<td>18.1 SmartDedupe</td>
<td>76</td>
</tr>
<tr>
<td>18.2 Isilon F810, H5600 and PowerScale nodes</td>
<td>76</td>
</tr>
<tr>
<td>19 16 TiB large file support and SyncIQ implications</td>
<td>77</td>
</tr>
<tr>
<td>20 OneFS version compatibility</td>
<td>78</td>
</tr>
<tr>
<td>21 SmartLock compatibility</td>
<td>79</td>
</tr>
<tr>
<td>21.1 Compliance mode</td>
<td>80</td>
</tr>
<tr>
<td>21.2 Failover and failback with SmartLock</td>
<td>80</td>
</tr>
<tr>
<td>21.3 SmartLock and SyncIQ security</td>
<td>80</td>
</tr>
<tr>
<td>22 Conclusion</td>
<td>81</td>
</tr>
<tr>
<td>A Failover and failback steps</td>
<td>82</td>
</tr>
<tr>
<td>A.1 Assumptions</td>
<td>82</td>
</tr>
<tr>
<td>A.2 Failover</td>
<td>82</td>
</tr>
<tr>
<td>A.3 Failback</td>
<td>83</td>
</tr>
<tr>
<td>A.3.1 Finalizing the failback</td>
<td>84</td>
</tr>
<tr>
<td>B SyncIQ encryption with self-signed certificates</td>
<td>85</td>
</tr>
<tr>
<td>B.1 Generate keys</td>
<td>85</td>
</tr>
<tr>
<td>B.2 Import keys and apply SyncIQ settings</td>
<td>85</td>
</tr>
<tr>
<td>B.3 Create an encrypted SyncIQ policy</td>
<td>87</td>
</tr>
<tr>
<td>B.4 Modify an existing SyncIQ policy for encryption</td>
<td>88</td>
</tr>
<tr>
<td>B.5 Additional SyncIQ information and optional commands</td>
<td>88</td>
</tr>
<tr>
<td>C Configuring cascaded replication</td>
<td>89</td>
</tr>
<tr>
<td>D Configuring custom replication</td>
<td>91</td>
</tr>
<tr>
<td>E Technical support and resources</td>
<td>94</td>
</tr>
<tr>
<td>E.1 Related resources</td>
<td>94</td>
</tr>
</tbody>
</table>
Executive summary

Simple, efficient, and scalable, Dell EMC™ PowerScale™ SyncIQ™ data replication software provides data-intensive businesses with a multi-threaded, multi-site solution for reliable disaster protection.

All businesses want to protect themselves against unplanned outages and data loss. The best practice is typically to create and keep copies of critical data, so it can always be recovered. There are many approaches to creating and maintaining data copies. The right approach depends on the criticality of the data to the business and its timeliness, in essence, how long the business can afford to be without it.

As the sheer amount of data requiring management grows, it puts considerable strain on a company’s ability to protect its data. Backup windows shrink, bottlenecks emerge, and logical and physical divisions of data fragment data protection processes. The result is increased risk with storing data and the growing complexity in managing it.

PowerScale SyncIQ offers powerful, flexible, and easy-to-manage asynchronous replication for collaboration, disaster recovery, business continuity, disk-to-disk backup, and remote disk archiving.

Note to readers

Prior to making changes on a production cluster, extreme caution is recommended. The concepts explained in this paper must be understood in its entirety before implementing data replication. As with any significant infrastructure update, testing changes in a lab environment is best practice. Once updates are confirmed in a lab environment a gradual roll-out to a production cluster may commence.
Introduction

1 Introduction

SyncIQ delivers unique, highly parallel replication performance that scales with the dataset to provide a solid foundation for disaster recovery. SyncIQ can send and receive data on every node in a PowerScale cluster, taking advantage of any available network bandwidth, so replication performance increases as the data store grows. Data replication starts and remains a simple process because both the replication source and target can scale to multiple petabytes without fragmentation into multiple volumes or file systems.

<table>
<thead>
<tr>
<th>Isilon Source Cluster</th>
<th>Isilon Target Cluster</th>
</tr>
</thead>
</table>

Figure 1 PowerScale SyncIQ parallel replication

A simple and intuitive web-based user interface allows administrators to easily organize SyncIQ replication job rates and priorities to match business continuity priorities. Typically, a SyncIQ recurring job is defined to protect the data required for each major Recovery Point Objective (RPO) in the disaster recovery plan. For example, an administrator may choose to sync every 6 hours for customer data, every 2 days for HR data, and so on. A directory, file system or even specific files may be configured for more- or less-frequent replication based on their business criticality. In addition, administrators can create remote archive copies of non-current data that needs to be retained, reclaiming valuable capacity in a production system.

SyncIQ can be tailored to use as much or as little system resource and network bandwidth as necessary, and the sync jobs can be scheduled to run at any time, in order to minimize the impact of the replication on production systems.
2 Deployment topologies

Meeting and exceeding the data replication governance requirements of an organization are critical for an IT administration. SyncIQ exceeds these requirements by providing an array of configuration options, ensuring administrators have flexible options to satisfy all workflows with simplicity.

Under each deployment, the configuration could be for the entire cluster or a specified source directory. Additionally, the deployment could have a single policy configured between the clusters or several policies, each with different options aligning to RPO and RTO requirements. For more information about configuration options, refer to Section 6, Configuring a SyncIQ policy.

2.1 One-to-one

In the most common deployment scenario of SyncIQ, data replication is configured between a single source and single target cluster as illustrated in Figure 2.

![Figure 2 SyncIQ one-to-one data replication](image)

2.2 One-to-many

SyncIQ supports data replication from a single source cluster to many target clusters, allowing the same dataset to exist in multiple locations, as illustrated in Figure 3. A one-to-many deployment could also be referenced as a hub-and-spoke deployment, with a central source cluster as the hub and each remote location representing a spoke.

![Figure 3 SyncIQ one-to-many data replication](image)
2.3 Many-to-one

The many-to-one deployment topology is essentially the flipped version of the one-to-many explained in the previous section. Several source clusters replicate to a single target cluster as illustrated in Figure 4. The many-to-one topology may also be referred to as a hub-and-spoke configuration. However, in this case, the target cluster is the hub, and the spokes are source clusters.

![Isilon Source Clusters](image1)

![Isilon Target Cluster](image2)

Figure 4  SyncIQ many-to-one data replication

2.4 Local target

A local target deployment allows a single PowerScale cluster to replicate within itself providing the SyncIQ powerful configuration options in a local cluster as illustrated in Figure 5. If a local target deployment is used for disaster readiness or archiving options, the cluster protection scheme and storages pools must be considered.

![Isilon Source and Target Cluster](image3)

Figure 5  SyncIQ local target data replication
2.5 Cascaded

A cascaded deployment replicates a dataset through a series of clusters. It allows a primary cluster to replicate to a secondary cluster, next to a tertiary cluster, and so on, as illustrated in Figure 6. Essentially, each cluster replicates to a next in the chain. For a cascaded SyncIQ implementation, consider how the replication start times are configured on the 2nd and subsequent clusters. Ensure the start times do not start before the SyncIQ job completes from the previous cluster.

For illustration purposes, consider a cascaded SyncIQ replication with the implementation in Figure 6.

![Figure 6 SyncIQ cascaded data replication](image)

As a best practice, configure the SyncIQ policies on the 2nd and subsequent clusters to use the “Whenever a snapshot of the source directory is taken” option, resulting in a consistent view of the source cluster’s data. For example, to configure the SyncIQ cascaded implementation in Figure 6, configure the SyncIQ policies ‘B-C’ and ‘C-D’ using the “Whenever a snapshot of the source directory is taken” option. Further, configure this option based on a real snapshot name, rather than an alias name.

For more information about this option, refer to Section 6.3.4, Whenever a snapshot of the source directory is taken. For an example configuration, using this implementation, refer to Appendix C, Configuring cascaded replication.
2.6 Custom

A custom deployment combines the previous deployments. For example, as illustrated in Figure 7, a primary cluster replicates to a secondary, and then the secondary replicates to a set of tertiary clusters. Essentially, this implementation is a combination of the ‘Cascaded’ and ‘One-to-many’ deployments.

Figure 7  SyncIQ cascaded and one-to-many data replication

For more information about this option, refer to Section 6.3.4, Whenever a snapshot of the source directory is taken. For an example configuration, using this implementation, refer to Appendix D, Configuring custom replication.
3 Use cases
PowerScale SyncIQ offers powerful, efficient, and easy-to-manage data replication for disaster recovery, business continuity, remote collaboration, disk-to-disk backup, and remote disk archive.

Figure 8 illustrates the typical SyncIQ architecture — replicating data from a primary to a target PowerScale cluster which can be local or remote. SyncIQ can also use the primary cluster as a target in order to create local replicas.

Figure 8  SyncIQ data replication over the LAN and WAN
SyncIQ provides the power and flexibility for the protection requirements of data-intensive workflows and applications.

3.1 Disaster recovery
Disaster recovery requires quick and efficient replication of critical business data to a secondary site. SyncIQ delivers high performance, asynchronous replication of data, providing protection from both local site and regional disasters, to satisfy a range of recovery objectives. SyncIQ has a very robust policy-driven engine that allows customization of replication datasets to minimize system impact while still meeting data protection requirements. SyncIQ automated data failover and failback reduces the time, complexity and risks involved with transferring operations between a primary and secondary site, in order to meet an organization’s recovery objectives. This functionality can be crucial to the success of a disaster recovery plan.

3.2 Business continuance
By definition, a business continuity solution needs to meet the most aggressive recovery objectives for the most timely, critical data. The SyncIQ highly efficient architecture provides performance that scales to maximize usage of any available network bandwidth and provides administrators the best-case replication time for aggressive Recovery Point Objectives (RPO). SyncIQ can also be used in concert with Dell EMC PowerScale SnapshotIQ software, which allows the storage of point-in-time snapshots in order to support secondary activities like the backup to tape.
3.3 Disk-to-disk backup and restore
Enterprise IT organizations face increasingly complex backup environments with costly operations, shrinking backup and restore windows, and stringent service-level agreement (SLA) requirements. Backups to tape are traditionally slow and hard to manage as they grow, compounded by the size and rapid growth of digital content and unstructured data. SyncIQ, as a superior disk-to-disk backup and restore solution delivers scalable performance and simplicity, enabling IT organizations to reduce backup and restore times and costs, eliminate complexity, and minimize risk. With PowerScale scale-out network-attached storage (NAS), petabytes of backup storage can be managed within a single system-as one volume, and one file system and can be the disk backup target for multiple PowerScale clusters.

3.4 Remote archive
For data that is too valuable to throw away, but not frequently accessed enough to justify maintaining it on production storage, replicate it with SyncIQ to a secondary site and reclaim the space on the primary system. Using a SyncIQ copy policy, data can be deleted on the source without affecting the target, leaving a remote archive for disk-based tertiary storage applications or staging data before it moves to offline storage. Remote archiving is ideal for intellectual property preservation, long-term records retention, or project archiving.
4 Architecture and processes

SyncIQ leverages the full complement of resources in a PowerScale cluster and the scalability and parallel architecture of the Dell EMC PowerScale OneFS™ file system. SyncIQ uses a policy-driven engine to execute replication jobs across all nodes in the cluster.

Multiple policies can be defined to allow for high flexibility and resource management. The replication policy is created on the source cluster, and data is replicated to the target cluster. As the source and target clusters are defined, source and target directories are also selected, provisioning the data to replicate from the source cluster and where it is replicated on the target cluster. The policies can either be executed on a user-defined schedule or started manually. This flexibility allows administrators to replicate datasets based on predicted cluster usage, network capabilities, and requirements for data availability.

Once the replication policy starts, a replication job is created on the source cluster. Within a cluster, many replication policies can be configured.

During the initial run of a replication job, the target directory is set to read-only and is solely updated by jobs associated with the replication policy configured. When access is required to the target directory, the replication policy between the source and target must be broken. Once access is no longer required on the target directory, the next jobs require an initial or differential replication to establish the sync between the source and target clusters.

**Note:** Practice extreme caution prior to breaking a policy between a source and target cluster or allowing writes on a target cluster. Prior to these actions, ensure the repercussions are understood. For more information, refer to Section 7, Impacts of modifying SyncIQ policies and section 11.4, Allow-writes compared to break association.

![Diagram showing architecture and processes](image)

**Figure 9** PowerScale SyncIQ replication policies and jobs

When a SyncIQ job is initiated, from either a scheduled or manually applied policy, the system first takes a snapshot of the data to be replicated. SyncIQ compares this to the snapshot from the previous replication job to quickly identify the changes that need to be propagated. Those changes can be new files, changed files, metadata changes, or file deletions. SyncIQ pools the aggregate resources from the cluster, splitting the replication job into smaller work items and distributing these amongst multiple workers across all nodes in the cluster. Each worker scans a part of the snapshot differential for changes and transfers those changes to the target cluster. While the cluster resources are managed to maximize replication performance, administrators can decrease the impact on other workflows using configurable SyncIQ resource limits in the policy.
Replication workers on the source cluster are paired with workers on the target cluster to accrue the benefits of parallel and distributed data transfer. As more jobs run concurrently, SyncIQ employs more workers to utilize more cluster resources. As more nodes are added to the cluster, file system processing on the source cluster and file transfer to the remote cluster are accelerated, a benefit of the PowerScale scale-out NAS architecture.

![Diagram of SyncIQ snapshots and work distribution]

Figure 10  SyncIQ snapshots and work distribution

SyncIQ is configured through the OneFS WebUI, providing a simple, intuitive method to create policies, manage jobs, and view reports. In addition to the web-based interface, all SyncIQ functionality is integrated into the OneFS command line interface. For a full list of all commands, run `isi sync --help`.

4.1 Asynchronous source-based replication

SyncIQ is an asynchronous remote replication tool. It differs from synchronous remote replication tools where the writes to the local storage system are not acknowledged back to the client until those writes are committed to the remote storage system. SyncIQ asynchronous replication allows the cluster to respond quickly to client file system requests while replication jobs run in the background, per policy settings.

To protect distributed workflow data, SyncIQ prevents changes on target directories. If the workflow requires writable targets, the SyncIQ source/target association must be broken before writing data to a target directory, and any subsequent re-activation of the synchronize association requires a full synchronization.

**Note:** Practice extreme caution prior to breaking a policy between a source and target cluster or allowing writes on a target cluster. Prior to these actions, ensure the repercussions are understood. For more information, refer to section 7, Impacts of modifying SyncIQ policies and section 11.4, Allow-writes compared to break association.

4.2 Source cluster snapshot integration

To provide point-in-time data protection, when a SyncIQ job starts, it automatically generates a snapshot of the dataset on the source cluster. Once it takes a snapshot, it bases all replication activities (scanning, data transfer, etc.) on the snapshot view. Subsequent changes to the file system while the job is in progress will not be propagated; those changes will be picked up the next time the job runs. OneFS creates instantaneous
snapshots before the job begins – applications remain online with full data access during the replication operation.

**Note:** This source-cluster snapshot does not require a SnapshotIQ module license. Only the SyncIQ license is required.

Source-cluster snapshots are named SIQ-<policy-id>-[new, latest], where <policy-id> is the unique system-generated policy identifier. SyncIQ compares the newly created snapshot with the one taken during the previous run and determines the changed files and blocks to transfer. Each time a SyncIQ job completes, the associated ‘latest’ snapshot is deleted and the previous ‘new’ snapshot is renamed to ‘latest’.

**Note:** A SyncIQ snapshot should never be deleted. Deleting a SyncIQ snapshot breaks a SyncIQ relationship, forcing a resync.

Regardless of the existence of other inclusion or exclusion directory paths, only one snapshot is created on the source cluster at the beginning of the job based on the policy root directory path.

**Note:** Deleting a SyncIQ policy also deletes all snapshots created by that policy.

### 4.2.1 Snapshot integration alleviates treewalks

When a SyncIQ job starts, if a previous source-cluster snapshot is detected, SyncIQ sends to the target only those files that are not present in the previous snapshot, as well as changes to files since the last source-cluster snapshot was taken. Comparing two snapshots to detect these changes is a much more lightweight operation than walking the entire file tree, resulting in significant gains for incremental synchronizations subsequent to the initial full replication.

If there is no previous source-cluster snapshot (for example, if a SyncIQ job is running for the first time), a full replication will be necessary.

When a SyncIQ job completes, the system deletes the previous source-cluster snapshot, retaining the most recent snapshot to be used as the basis for comparison on the next job iteration.
4.3 Processes

In order to understand how SyncIQ implements each policy, it is essential to understand the processes associated with data replication as illustrated in Figure 11.

### 4.3.1 Scheduler

Each PowerScale node has a Scheduler process running. It is responsible for the creation and launch of SyncIQ data replication jobs and creating the initial job directory. Based on the current SyncIQ configuration, the Scheduler starts a new job and updates jobs based on any configuration changes.

### 4.3.2 Coordinator

The Scheduler launches the Coordinator process. The Coordinators create and oversee the worker processes as a data replication job runs. The Coordinator is responsible for snapshot management, report generation, bandwidth throttling, managing target monitoring, and work distribution.

Snapshot management involves capturing the file system snapshots for SyncIQ. The snapshots are locked while in use and deleted after completion. Report management acquires job data from each process and combines this to a single report. Bandwidth throttling provides the Coordinator with bandwidth information to align jobs with available bandwidth. Target monitoring management is monitoring the target cluster’s worker process. And finally, work distribution maximizes job performance by ensuring all worker process have even utilization.

### 4.3.3 Primary and secondary workers

Primary workers and secondary workers run on the source and target clusters, respectively. They are responsible for the actual data replication piece during a SyncIQ job.
4.3.4 Target monitor

The target monitor provides critical information about the target cluster and does not participate in the data transfer. It reports back with IP addresses for target nodes including any changes on the target cluster. Additionally, the target monitor takes target snapshots as they are required.
Data replication

5 Data replication

When SyncIQ replicates data, it goes through one of three phases. The three phases are Initial, Incremental, and Differential. This section explains each phase.

Note: This section provides a detailed explanation of the SyncIQ data replication process. Many of the details in this section may not be necessary for implementing and managing SyncIQ. Understanding all of the steps in this section is not required. However, the details in this section are provided for a granular understanding of how SyncIQ data replication occurs, enabling a foundation of the concepts explained throughout this paper.

5.1 Initial replication

After a policy is configured, the first time it runs, an Initial Replication is executed. During the policy configuration, a user can configure a synchronization or copy policy.

The synchronization policy ensures the target cluster has a precise duplicate of the source directory. As the source directory is modified through additions and deletions, those updates are propagated to the target cluster when the policy runs next. Under Disaster Recovery use cases, the synchronization policy supports a failover to the target cluster, allowing users to continue with access to the same dataset as the source directory.

On the contrary, a copy policy is targeted for archive and backup use cases. A copy policy maintains current versions of files stored on the source cluster.

The first segment of the Initial Replication is the job start. A scheduler process is responsible for starting a data replication job. It determines the start time based on either the scheduled time or a manually started job. Once the time arrives the scheduler updates the policy to a pending status on the source record and creates a directory with information specific to the job.

After the creation of the initial directory with the SyncIQ policy ID, a scheduler process of a node takes control of the job. Once a node’s scheduler process has taken control of the job the directory is renamed again to reflect the node’s device ID. Next, one of the scheduler processes create the coordinator process and the directory structure is renamed again.

Once the directory structure is renamed to reflect the SyncIQ policy ID, node ID, and coordinator PID, the data transfer stage commences. The coordinator has a primary worker process start a treewalk of the current SyncIQ snapshot. This snapshot is named snapshot-<SyncIQ Policy ID>-new. On the target cluster, the secondary workers receive the treewalk information, mapping out the LINs accordingly.

During the treewalk and exchange of LIN information, a list of target node IP addresses is gathered through the target monitor process. At this point, the primary workers setup TCP connections with the secondary workers of target nodes for the remainder of the job. If a worker on a cluster crashes, the corresponding worker will also. In this event, the coordinator launches a new primary worker process and establish a new TCP connection with a secondary worker. If the coordinator crashes, the scheduler restarts the coordinator, and all workers must establish TCP connections again. The number of workers are calculated based on many factors. Refer to Section 16.5.1, Workers and performance scalability, for more information about calculating workers.

Now that the primary and secondary workers are created with TCP connections between each, data transfer is started between each set of workers.
As each set of workers completes data transfer, they go into an idle state. Once all workers are in an idle state, and the restart queue does not contain any work items, this indicates the data replication is complete. At this point, the coordinator renames the snapshot taken at the onset to snapshot-<SyncIQ Policy ID>-latest. Next, the coordinator files a job report. If the SyncIQ policy is configured to create a target-side snapshot, that is taken at this time. Finally, the coordinator removes the job directory that was created at the onset and the job is complete.

5.2 Incremental replication

An Incremental Replication of a SyncIQ policy only transfers the portions of files that have changed since the last run. Therefore, the amount of data replicated, and bandwidth consumption is significantly reduced in comparison to the initial replication.

Similar to the Initial Replication explained above, at the start of an incremental replication, the scheduler processes create the job directory. Next, the coordinator starts a process of collecting changes to the dataset, by taking a new snapshot and comparing it to the previous snapshot. The changes are compiled into an incremental file with a list of LINs that have been modified, added, or deleted.

Once all the new modifications to the dataset are logged, workers read through the file and start to apply the changes to the target cluster. On the target cluster, the deleted LINs are removed first, followed by updating directories that have changed. Finally, the data and metadata are updated on the target cluster.

As all updates complete, the coordinator creates the job report, and the replication is complete.

5.3 Differential replication or target aware sync

In the event where the association between a source and target is lost or broken, incremental replications will not work. At this point, the only available option is to run an initial replication on the complete dataset. Running the initial replication again, is bandwidth and resource intensive, as it is essentially running again as a new policy. The Differential Replication offers a far better alternative to running the initial replication again.

**Note:** Running an Initial Replication again after the source and target cluster association is broken has impacts not only on bandwidth and cluster resources, but also creates ballooning snapshots on the target cluster for snapshots outside of SyncIQ re-replication. A Differential Replication eliminates these concerns.

The term ‘Differential Replication’ is also referred to as ‘Target Aware Sync’, ‘Target Aware Initial Sync’, and ‘Diff Sync’. All of these terms are referencing a Differential Replication.

A Differential Replication, similar to an Incremental Replication only replicates changed data blocks and new data that does not exist on the target cluster. Determining what exists on each cluster is part of the differential replication’s algorithm. The files on the source directory are compared to the target directory to decide if replication is required. The algorithm to determine if a file should be replicated is based on if the file or directory is new, the file size and length, and finally the short and full hash of the file.

**Note:** Target Aware Synchronizations are much more CPU-intensive than regular baseline replication, but they potentially yield much less network traffic if both source and cluster datasets are already seeded with similar data.
The Target Aware Initial Sync feature, available only via the CLI. To enable target aware initial synchronization, use the following command:

```
isi sync policies modify <policy_name> --target-compare-initial-sync=on
```
6 Configuring a SyncIQ policy

SyncIQ is configured through policies. The policies provide the starting point of OneFS data replication. The policies offer a breadth of options for an administrator to configure data replication specific to a workflow. SyncIQ is disabled by default on Greenfield PowerScale clusters on OneFS 9.1 or newer. Enable SyncIQ by clicking “Activate SyncIQ” under Data Protection > SyncIQ. Once SyncIQ is enabled, encryption is required for new policies. For more information about configuring encryption, refer to Section 14, SyncIQ security.

SyncIQ configuration may depend on the Access Zone configuration. It is important to understand the impacts as SyncIQ policies are configured. For more information about best practices with Access Zones, refer to the PowerScale Network Design Considerations white paper. Prior to proceeding with a SyncIQ policy configuration, ensure the Access Zones best practices are considered. Additionally, the design of policies must consider other resources as stated in Section 10, SyncIQ design considerations.

The SyncIQ policies are configurable through the CLI or the web interface. To configure SyncIQ from the CLI, start with the command isisync policies --help.

To access the SyncIQ policies from the web interface, once logged in, click Data Protection > SyncIQ, then click the “Policies” tab. A new SyncIQ policy is created by clicking “Create a SyncIQ Policy”, displaying the “Create SyncIQ Policy” window, as displayed in Figure 12.

![Create SyncIQ Policy](image)

Figure 12 OneFS WebUI SyncIQ policy

6.1 Naming and enabling a policy

Taking the best practices stated above into account, the “Policy Name” field should be descriptive enough for administrators to easily gather the policy workflow, as several policies could be configured on a cluster. A unique name makes it easy to recognize and manage. Additionally, the “Description” field can be used to explain further.
The “Enable this policy” checkbox is a powerful option allowing an administrator to start configuration prior to a target cluster or directory being ready for replication. Temporarily disabling a policy allows for a less intrusive option to deleting a policy when it may not be required. Additionally, after completing the configuration for a policy, it can be reviewed for a final check, prior to enabling.

6.2 Synchronization and copy policies

SyncIQ provides two types of replications policies: synchronization and copy. Data replicated with a synchronization policy is maintained on the target cluster precisely as it is on the source – files deleted on the source are deleted next time the policy runs. A copy policy produces essentially an archived version of the data – files deleted on the source cluster will not be deleted from the target cluster. However, there are some specific behaviors in certain cases, explained below.

If a directory is deleted and replaced by an identically named directory, SyncIQ recognizes the re-created directory as a “new” directory, and the “old” directory and its contents will be removed.

**Example:**

If an administrator deletes “/ifs/old/dir” and all of its contents on the source with a copy policy, “/ifs/old/dir” still exists on the target. Subsequently, a new directory is created, named “/ifs/old/dir” in its place, the old “dir” and its contents on the target will be removed, and only the new directory’s contents will be replicated.

SyncIQ keeps track of file moves and maintains hard-link relationships at the target level. SyncIQ also removes links during repeated replication operations if it points to the file or directory in the current replication pass.

**Example:**

If a single linked file is moved within the replication set, SyncIQ removes the old link and adds a new link. Assume the following:

The SyncIQ policy root directory is set to /ifs/data.

/ifs/data/user1/foo is hard-linked to /ifs/data/user2/bar.

/ifs/data/user2/bar is moved to /ifs/data/user3/bar.

With copy replication, on the target cluster, /ifs/data/user1/foo will remain, and ifs/data/user2/bar will be moved to /ifs/data/user3/bar.

If a single hard link to a multiply linked file is removed, SyncIQ removes the destination link.

**Example:**

Using the example above, if /ifs/data/user2/bar is deleted from the source, copy replication also removes /ifs/data/user2/bar from the target.

If the last remaining link to a file is removed on the source, SyncIQ does not remove the file on the target unless another source file or directory with the same filename is created in the same directory (or unless a deleted ancestor is replaced with a conflicting file or directory name).

**Example:**
Continuing with the same example, assume that /ifs/data/user2/bar has been removed, which makes /ifs/data/user1/foo the last remaining link. If /ifs/data/user1/foo is deleted on the source cluster, with a copy replication, SyncIQ does not delete /ifs/data/user1/foo from the target cluster unless a new file or directory was created on the source cluster that was named /ifs/data/user1/foo. Once SyncIQ creates the new file or directory with this name, the old file on the target cluster is removed and re-created upon copy replication.

If a file or directory is renamed or moved on the source cluster and still falls within the SyncIQ policy's root path when copied, SyncIQ will rename that file on the target; it does not delete and re-create the file. However, if the file is moved outside of the SyncIQ policy root path, then with copy replication, SyncIQ will leave that file on the target but will no longer associate it with the file on the source. If that file is moved back to the original source location or even to another directory within the SyncIQ policy root path, with copy replication, SyncIQ creates a new file on the target since it no longer associates it with the original target file.

**Example:**

Consider a copy policy rooted at /ifs/data/user. If /ifs/data/user1/foo is moved to /ifs/data/user2/foo, SyncIQ simply renames the file on the target on the next replication. However, if /ifs/data/user1/foo is moved to /ifs/home/foo, which is outside the SyncIQ policy root path, with copy replication, SyncIQ does not delete /ifs/data/user1/foo on the target, but it does disassociate, or orphan it, from the source file, that now resides at /ifs/home/foo. If, on the source cluster, the file is moved back to /ifs/data/user1/foo, an incremental copy writes that entire file to the target cluster because the association with the original file has been broken.

### 6.3 Running a SyncIQ job

A SyncIQ Policy may be configured to run with four different options. Each of those options is explained in this section.

**Note:** Although SyncIQ offers many options for configuring a SyncIQ policy, as explained in this section, the “Whenever a snapshot of the source directory is taken” is the best practice and recommended configuration. For more information about this configuration, refer to Section 6.3.4, Whenever a snapshot of the source directory is taken.

#### 6.3.1 Manually

The manual option allows administrators to have a SyncIQ Policy completely configured and ready to run when a workflow requires data replication. If continuous data replication is not required and on an 'as needed' basis, this is the best option. Administrators can simply select the policy to run when it is required, limiting cluster overhead and saving bandwidth.

**Note:** Manual SyncIQ jobs still maintain a source snapshot that accumulates changed blocks. Therefore, it is recommended to run the manual job frequently, ensuring the source snapshot growth is limited.
6.3.2 On a schedule

Running a SyncIQ Policy on a schedule is one of the more common options. Once this option is selected, another drop-down appears, to specify the frequency of the job, as displayed in Figure 13.

![SyncIQ Job on a schedule](image)

Figure 13  SyncIQ Job on a schedule

Options include daily, weekly, monthly, or yearly. Once the frequency is selected further options appear to refine the frequency selection.

Before OneFS 8.0, a snapshot is always taken for scheduled jobs, even if no data changes have occurred since the previous execution. In OneFS 8.0, a policy parameter can be specified so that SyncIQ checks for changes since the last replication as the first step in the policy. If there are no changes, no further work will be done on that policy iteration, and the policy will report as “skipped”. If there are changes, the source data snapshot will be taken, and the policy will proceed. This capability reduces the amount of work performed by the cluster if there is no changed data to be replicated. To enable this behavior, check “Only run if source directory contents are modified” on the WebUI or specify `-skip-when-source-unmodified true` on the CLI.

**Note:** As a best practice, avoid the overlap of policy start times or have several policies running during the same time period. As explained in Section 10, SyncIQ design considerations, consider policy start times and cluster resources. As policies complete, monitor completion times and adjust policy start times to minimize overlap. Staggering policy start times is especially critical for a high-volume dataset.

6.3.2.1 RPO alerts

An option for sending RPO alerts is available when “On a Schedule” is selected for a running a job. Administrators can specify an RPO (recovery point objective) for a scheduled SyncIQ policy and trigger an event to be sent if the RPO is exceeded. The RPO calculation is the interval between the current time and the start of the last successful sync job.

**Note:** The RPO option only appears if RPO is enabled under SyncIQ global settings. From the web interface select **Data Protection > SyncIQ**, then select the “Settings” tab, and the “Enable RPO Alerts” checkbox is displayed.
Configuring a SyncIQ policy

For example, consider a policy scheduled to run every 8 hours with a defined RPO of 12 hours. Suppose the policy runs at 3 pm and completes successfully at 4 pm. Thus, the start time of the last successful sync job is 3 pm. The policy should run next at 11 pm, based on the 8-hour scheduled interval. If this next run completes successfully before 3 am, 12 hours since the last sync start, no alert will be triggered, and the RPO timer is reset to the start time of the replication job. If for any reason the policy has not run to successful completion by 3 am, an alert will be triggered, since more than 12 hours elapsed between the current time (after 3 am) and the start of the last successful sync (3 pm).

If an alert has been triggered, it is automatically canceled after the policy successfully completes.

The RPO alert can also be used for policies that have never been run, as the RPO timer starts at the time the policy is created. For example, consider a policy created at 4 pm with a defined RPO of 24 hours. If by 4 pm the next day, the policy has not successfully completed at least one synchronization operation, the alert will be triggered. As stated previously, the first run of a policy is a full synchronization and will probably require a longer elapsed time than subsequent iterations.

An RPO can only be set on a policy if the global SyncIQ setting for RPO is already set to enabled: isi sync settings modify --rpo-alerts true|false. By default, RPO alerts are enabled.

Individual policies by default have no RPO alert setting. Use --rpo-alert <duration> on the isi sync policies create or modify command to specify the duration for a particular policy.

6.3.3 Whenever the source is modified

The “Whenever the Source is Modified” option is also referred to as, ‘SyncIQ continuous mode’, or ‘Replicate on Change’. When the “Whenever the source is modified” policy configuration option is selected (or -- schedule when-source-modified on the CLI), SyncIQ will continuously monitor the replication data set and automatically replicate changes to the target cluster. Continuous replication mode is applicable when the
target cluster data set must always be consistent with the source, or if data changes at unpredictable intervals.

**Figure 15  SyncIQ source modified option**

**Note:** Practice extreme caution with the “Whenever the source is modified” option as it can trigger a large amount of replication, snapshot, and network traffic if the data is volatile. The source modified option is not synchronous data replication. Consider the cluster resources and frequency of dataset updates when applying this option. It may result in SyncIQ policies constantly running and excessive resource consumption. Another factor to consider is, by default, snapshots of the source directory are taken before each SyncIQ job. If the dataset is frequently modified, many snapshots are triggered, possibly conflicting with other snapshot activity. If selecting this option is necessary, ensure the sync delay is configured with ample time to encapsulate new data and allows for the policy to complete.

Events that trigger replication include file additions, modifications and deletions, directory path, and metadata changes. SyncIQ checks the source directories every ten seconds for changes, as illustrated in Figure 16.

**Figure 16  SyncIQ source modified policy triggers**

Before OneFS 8.0, jobs in Continuous Replication mode execute immediately after a change is detected. OneFS 8.0 introduces a policy parameter to delay the replication start for a specified time after the change is detected. The delay allows a burst of updates to a data set to be propagated more efficiently in a single replication event rather than triggering multiple events. To enable the delay for a continuous replication policy, specify the delay period in the “Change-Triggered Sync Job Delay” option on the GUI, as shown in Figure 15, or specify `--job-delay <duration>` on the CLI.

**Note:** As a best practice, if the “Whenever the source is modified” option is selected, configure the “Change-Triggered Sync Job Delay” option for a reasonable delay to propagate multiple updates into a single update.
Whenever a snapshot of the source directory is taken

A SyncIQ policy can be configured to trigger when the administrator takes a snapshot of the specified source directory and matching a specified pattern as displayed in Figure 17.

![Figure 17](image)

If this option is specified, the administrator-taken snapshot will be used as the basis of replication, rather than generating a system snapshot. Basing the replication start on a snapshot is useful for replicating data to multiple targets – these can all be simultaneously triggered when a matching snapshot is taken, and only one snapshot is required for all the replications. To enable this behavior, select the “Whenever a snapshot of the source directory is taken” policy configuration option on the GUI. Alternatively, from the CLI, use the flag, `--schedule=when-snapshot-taken`.

All snapshots taken of the specified source directory trigger a SyncIQ job to start, replicating the snapshot to the target cluster. An administrator may limit all snapshots from triggering replication by specifying a naming convention to match in the “Run job if snapshot name matches the following pattern:” field. By default, the field contains an asterisk, triggering replication for all snapshots of the source directory. Alternatively, from the CLI, if the flag `--snapshot-sync-pattern <string>` is not specified, the policy automatically enters an asterisk, making this flag optional.

The checkbox, “Sync existing snapshots before policy creation time”, only displays for a new policy. If an existing policy is edited, this option is not available. Alternatively, from the CLI, the flag `--snapshot-sync-existing` is available for new policies. The “Sync existing snapshots before policy creation time” option replicates all snapshots to the target cluster that were taken on the specified source cluster directory.

**Note:** The “Whenever a snapshot of the source directory is taken” is the best practice and recommended policy for scheduling SyncIQ policies. Further, the `when-snapshot-taken` SyncIQ policy schedule should be driven by first creating a SnapshotIQ policy on the source directory with the desired schedule. After configuring the SnapshotIQ policy, the `when-snapshot-taken` SyncIQ policy can be created or modified to use the SnapshotIQ schedule and the `--snapshot-sync-existing` option. For more information about SnapshotIQ and SyncIQ, refer to Section 9, SnapshotIQ and SyncIQ.

When snapshots are replicated to the target cluster, by default, only the most recent snapshot is retained and the naming convention on the target cluster is system generated. However, in order to prevent only a single snapshot being overwritten on the target cluster and the default naming convention, select the “Enable capture of snapshots on the target cluster” as stated in Section 6.8 Target snapshots. Once this checkbox is selected, specify a naming pattern and select the “Snapshots do not expire” option. Alternatively, specify a date for snapshot expiration. Limiting snapshots from expiring ensures they are retained on the target cluster rather than overwritten when a newer snapshot is available. The target cluster snapshot options map to `--target-snapshot-archive, --target-snapshot-alias, --target-snapshot-expiration, and --target-snapshot-pattern` in the CLI.
Note: If snapshots are configured for automatic capture based on a time-frequency, this triggers the SyncIQ policy to run. If SyncIQ policies are constantly running, consider the impact on system resources prior to configuring. As with any major storage infrastructure update, test in a lab environment prior to a production cluster update, ensuring all resource impacts are considered and calculated.

Alternatively, SyncIQ also provides an option for manually specifying an existing snapshot for SyncIQ replication, as explained in Section 9, SnapshotIQ and SyncIQ.

6.4 Source cluster directory

The Source Cluster section is used to specify where the source data resides that will be replicated to the target cluster, as displayed in Figure 18.

- Source Cluster
  - Source Root Directory
    - /ifs
  - Included Directories
    - Remove path
    - + Add another directory path
  - Excluded Directories
    - Remove path
    - + Add another directory path

![Figure 18  SyncIQ policy source cluster configuration](image)

A SyncIQ policy by default includes all files and folders under the specified root directory. Optionally, directories under the root directory can be explicitly included or excluded.

Note: As a best practice, avoid overlapping source directory SyncIQ policies with differing retention times to prevent nested snapshots.

If any directories are explicitly included in the policy configuration, the system synchronizes only those directories and their included files to the target cluster. If any directories are explicitly excluded, those directories and any files contained in them are not synchronized to the target cluster.

Any directories explicitly included must reside within the specified root directory tree. Consider a policy with the root directory /ifs/data and explicitly include the /ifs/data/media directory because it is under /ifs/data. When the associated policy runs, only the contents of the /ifs/data/media directory would be synchronized to the target cluster. However, the directory /ifs/projects, is not included, since this is not part of the /ifs/data tree.

If a directory is explicitly excluded within the specified root directory, all the contents of the root directory except for the excluded directory will be synchronized to the target cluster.

If both included and excluded directories are specified, every explicitly included directory will be replicated, and every other file, or directory, under the exclude directory, will be excluded from the replication dataset.

For example, consider a policy with the root directory /ifs/data, and the following directories explicitly included and excluded:
Explicitly included directories:

/ifs/data/media/music
/ifs/data/media/movies

Explicitly excluded directories:

/ifs/data/media/music/working
/ifs/data/media

In this example, all directories below /ifs/data/media are excluded except for those specifically included. Therefore, directories such as /ifs/data/media/pictures, /ifs/data/media/books, /ifs/data/media/games are excluded because of the exclude rule. The directory and all subdirectories of /ifs/data/media/music will be synchronized to the target cluster, except for the directory /ifs/data/media/music/working.

**Note:** Depending on the include and exclude directory configuration, SyncIQ performance may be impacted. If possible, avoiding an include and exclude configuration simplifies policy configuration and ensures performance is not degraded. As a best practice, test the impacts of include and exclude policies in a lab environment prior to a production cluster update. Alternatively, multiple policies can be configured with different source directories rather than creating a single policy with includes and excludes.

### 6.5 File matching criteria

In addition to refining the source dataset through the included and excluded directories, file matching further refines the selected source dataset for replication, as displayed in Figure 19.

**File Matching Criteria**

**IF Condition**

- Remove criteria
- Select Filter Type
- Delete this block | Add an "And" condition

**OR Condition**

- Remove criteria
- Select Filter Type
- Delete this block | Add an "And" condition

- Add an "Or" condition

**Figure 19**  SyncIQ policy file matching criteria

A SyncIQ policy can have file-criteria statements that explicitly include or exclude files from the policy action. A file-criteria statement can include one or more elements, and each file-criteria element contains a file attribute, a comparison operator, and a comparison value. To combine multiple criteria elements into a criteria statement, use the Boolean ‘AND’ and ‘OR’ operators. Any number of ‘AND’ and ‘OR’ file-criteria definitions may be configured.

However, when configuring file matching criteria, it is important to recognize the impact they have is dependent on the SyncIQ ‘Action’ selected above. If “Copy” was selected, more settings are available than “Synchronize” policies.
In both Synchronize and Copy policies, the wildcard characters *, ?, and [] or advanced POSIX regular expressions (regex) may be utilized. Regular expressions are sets of symbols and syntactic elements that match patterns of text. These expressions can be more powerful and flexible than simple wildcard characters. Isilon clusters support IEEE Std 1003.2 (POSIX.2) regular expressions. For more information about POSIX regular expressions, refer to the BSD manual pages. For example:

- To select all files ending in .jpg, use "*.jpg$.
- To select all files with either .jpg or .gif file extensions, use ".(jpg|gif)$.
- Include or exclude files based on file size by specifying the file size in bytes, KB, MB, GB, TB, or PB. File sizes are represented in multiples of 1,024, not 1,000.
- Include or exclude files based on the following type options: regular file, directory, or a soft link. A soft link is a particular type of POSIX file that contains a reference to another file or directory.

**Note:** With a policy of type Synchronize, modifying file attributes comparison options and values causes a resync and deletion of any non-matching files from the target the next time the job runs. This does not apply to Copy policies.

Copy policies also allow an administrator to select files based on file creation time, access time, and modification time.

**Note:** Specifying file criteria in a SyncIQ policy requires additional time to complete, degrading overall SyncIQ performance. Conversely, if the source directories are refined using the “Included” and “Excluded” directories option, as stated in section 6.4 Source cluster directory, performance is not impacted to the same degree as specifying the file criteria. However, depending on the configuration, “Includes” and “Excludes” could also impact performance significantly. If possible, the first preference is to create policies without includes, excludes, and file criteria. The second preference is to use includes and excludes and finally, the last preference is file criteria. As a best practice, test the impacts of file criteria, includes, and excludes in a lab environment to confirm performance, prior to a production cluster update.

### 6.6 Restricting SyncIQ source nodes

SyncIQ utilizes a node’s front-end network ports to send replication data from the source to the target cluster. By default, SyncIQ policies utilize all nodes and interfaces to allow for maximum throughput of a given policy. However, an administrator may want to exclude certain nodes from a SyncIQ policy. Excluding nodes from a SyncIQ policy is beneficial for larger clusters where data replication jobs can be assigned to certain nodes. In other cases, a client workflow may require a higher priority on a performance node over participating in data replication. From the policy configuration window, an option is available to run the policy on all nodes, or specifying a subnet and pool, as displayed in Figure 20.

**Restrict Source Nodes**

- **Run the policy on all nodes in this cluster**
- **Run the policy only on nodes in the specified subnet and pool**

Subnet and Pool: [Select a Subnet and Pool]

**Figure 20**  Restricting SyncIQ source nodes

By selecting a predefined IP address pool, administrators can restrict replication processing to specific nodes on the source cluster. This option is useful to ensure that replication jobs are not competing with other applications for specific node resources. Specifying the IP address pool allows administrators to define which networks are used for replication data transfer.
Note: By default, SyncIQ uses all interfaces in the nodes that belong to the IP address pool, disregarding any interface membership settings in the pool. To restrict SyncIQ to use only the interfaces in the IP address pool, use the following command line interface commands to modify the SyncIQ policy: 

```
isi sync policies modify --policy <my_policy> --force_interface=on
```

The same option is also available as a global SyncIQ setting, under Data Protection > SyncIQ and selecting the Settings tab. Administrators may use a single IP address pool globally across all policies or select different IP address pools for use on a per-policy basis.

Note: As stated in section 10.3.1, SyncIQ data replication is only supported through the System Access Zone since SyncIQ is not zone-aware. If a new SyncIQ policy is created or an existing policy is edited, an error is displayed if it is not configured for the System Access Zone. This zone requirement applies to both the source and target clusters.

To restrict sending replication traffic to specific nodes on the target cluster, an administrator can associate, globally or per policy, a SmartConnect zone name with the target cluster.

Note: Changing the default policy global settings only affects newly created policies; existing policies will not be modified.

6.7 Target host and directory

In the “Target Host” field, specify the IP address or fully qualified domain name of the target cluster. It is important to ensure the DNS hosts specified on the source cluster can resolve the FQDN of the target cluster.

In the “Target Directory” field, specify the directory where data from the source cluster is replicated. As stated above, it is recommended to consider the Access Zones best practices as the location of the target directory eases failover and failback operations in the future.

6.7.1 Target cluster SmartConnect zones

When a policy target cluster name or address is specified, a SmartConnect DNS zone name is used instead of an IP address or a DNS name of a specific node. An administrator may choose to restrict the connection to nodes in the SmartConnect zone, ensuring the replication job will only connect with the target cluster nodes assigned to that zone. During the initial part of a replication job, SyncIQ on the source cluster establishes an initial connection with the target cluster using SmartConnect. Once a connection with the target cluster is established, the target cluster replies with a set of target IP addresses assigned to nodes restricted to that SmartConnect zone. SyncIQ on the source cluster will use this list of target cluster IP addresses to connect local replication workers with remote workers on the target cluster.

To utilize target cluster SmartConnect zones, perform the following steps:

1. On the target cluster, create a SmartConnect zone using the cluster networking WebUI.
2. Add only those nodes that will be used for SyncIQ to the newly created zone.
3. On the source cluster, SyncIQ replication jobs (or global settings) specify the SmartConnect zone name as the target server name.

Note: SyncIQ requires a static allocation method of IP addresses and does not support SmartConnect Dynamic Allocation Method of IP address pools. If Dynamic Allocation IPs are specified, the replication job will fail with an error message in the log file and trigger an alert.
The same option is also available as a global SyncIQ setting, under **Data Protection > SyncIQ** and selecting the “Settings” tab. While SmartConnect node restriction settings are available per SyncIQ policy, often it is more useful to set them globally. Those settings will be applied by default to new policies unless they are overridden on a per-policy basis. However, changing these global settings will not affect existing policies.

**Note:** As stated in section 10.3.1, SyncIQ data replication is only supported through the System Access Zone since SyncIQ is not zone-aware. If a new SyncIQ policy is created or an existing policy is edited, an error is displayed if it is not configured for the System Access Zone. This zone requirement applies to both the source and target clusters.

### 6.8 Target snapshots

Depending on the administrator’s requirements, archiving snapshots may be required on the target cluster. Configuring snapshot archival on the target cluster is an optional configuration, as displayed in Figure 21.

**Target Snapshots**

![Enable capture of snapshots on the target cluster](image)

**Snapshot Alias Name**

Default name: SIQ-%{SrcCluster}-%{PolicyName}-latest

**SIQ_%{SrcCluster}_%{PolicyName}**

**Snapshot Naming Pattern**

Default pattern: SIQ-%{SrcCluster}-%{PolicyName}-Y-%m-%d-%H-%M-%S

**SIQ_%{SrcCluster}_%{PolicyName}-%Y-%m-%d-%H-%M**

**Snapshot Expiration**

- Snapshots do not expire
- Snapshots expire after...

Figure 21  SyncIQ target snapshots

By default, if the “Enable capture of snapshots on the target cluster” is *not* selected, the target cluster only retains the most recent snapshot, which is used during a failover.

To enable snapshot archiving on the target cluster, a SnapshotIQ license is required. When SyncIQ policies are set with snapshots on the target cluster, on the initial sync a snapshot will be taken at the beginning and the end. For incremental syncs, a snapshot will only be taken at the completion of the job.

**Note:** Prior to initializing a job, SyncIQ checks for the SnapshotIQ license on the target cluster. If it has not been licensed, the job will proceed without generating a snapshot on the target cluster, and SyncIQ will issue an alert noting that the license was not available.

Administrators can control how many snapshots of the target replication path are maintained over time by defining an expiration period on each of the target-cluster snapshots. For example, if a replication job is executed every day for a week (with target snapshots enabled), seven snapshots of the dataset on the target cluster are available, representing seven available versions of the dataset. In this example, if the target-cluster snapshot is configured to expire after seven days on a replication policy that is executed once per day, only seven snapshots will be available on the target cluster dataset.
Note: If snapshot-based replication is configured as explained in Section 6.3.4, Whenever a snapshot of the source directory is taken, and Section 9, SnapshotIQ and SyncIQ, target snapshot archival may be a necessity. If target snapshots are not archived, a separate snapshot copy is not retained when a new snapshot becomes available.

For more information about snapshots and SyncIQ, reference Section 9, SnapshotIQ and SyncIQ, and Section 10.2, Snapshots and SyncIQ policies.

### 6.8.1 Target cluster Snapshot Alias

Snapshot aliasing on a target cluster provides a powerful option for accessing SyncIQ based snapshots directly on a target cluster. A Snapshot Alias is recommended for scenarios where a workflow requires access to the most current snapshot on the target cluster and a consistent snapshot name. Once a SyncIQ policy is configured on the source cluster with a Snapshot Alias, an NFS export may be configured on the target cluster, pointing directly to the Snapshot Alias.

If a workflow uses the HEAD snapshot, or the most recent SyncIQ snapshot, errors occur if the snapshot is accessed directly when a SyncIQ job is running. The errors occur because the HEAD SyncIQ snapshot is in the process of updating.

Note: As a best practice, accessing SyncIQ based snapshots for client access is not recommended. For client access, a Snapshot Alias should be configured with an NFS export pointing to the Snapshot Alias.

To create or modify an existing SyncIQ policy with a snapshot alias on the target cluster, perform the following:

1. Create a new SyncIQ policy with a target cluster Snapshot Alias. On the source cluster:

   ```bash
   ```

   In the example above, the snapshot alias is defined as 'example_sync_snap'.

   Alternatively, to modify an existing SyncIQ policy on the source cluster:

   ```bash
   isi sync policies modify [Policy Name] --target-snapshot-archive=true --target-snapshot-alias=example_sync_snap
   ```

   Note: As a best practice, consider configuring a target snapshot expiration date that is reasonable for the workflow. The target snapshot expiration is configured using the `--target-snapshot-expiration` option in the CLI. This parameter specifies the expiration in seconds.

2. Run the new, or modified, policy from the previous step on the source cluster at least once to generate the Snapshot Alias. On the target cluster, create the NFS export pointing to the Snapshot Alias:

   ```bash
   isi nfs exports create --paths=[Target Cluster Directory defined in SyncIQ policy in Step 1] --snapshot=example_sync_snap
   ```

3. Mount the new NFS export on a Linux client:

   ```bash
   Linux-client# mount [Target cluster IP]:[Target Cluster Directory defined in Step 1] [Specify local mount location]
   ```
Alternatively, for a Microsoft Windows 10 client, enable the NFS service under “Control Panel > Programs > Programs and Features > Turn Windows features on or off.” Enable “Services for NFS” and mount the export from the Windows Command Prompt:

```bash
mount \[Target Cluster IP]\[Target Cluster Directory defined in Step 1] [Select a drive to map this mount]
```

As each new SyncIQ policy runs, the Snapshot Alias continues to point to the newest or HEAD SyncIQ snapshot. The Snapshot Alias pointer can be confirmed by listing the snapshot details. To check the snapshot details, perform the following:

1. On the target cluster, list the snapshots, using the ‘isi snapshot snapshots list’ command:

   ```bash
   isi9-s2-n1-l# isi snapshot snapshots list
   ID   Name                                   Path
   -------------------------------------------
   2    SIQisi9-s1-n1-foo-2020-06-25_11-08-10 /ifs/data/cls1
   3    example_sync_snap                      /ifs/data/cls1
   5    SIQ-Failover-foo-2020-06-25_11-08-14 /ifs/data/cls1
   -------------------------------------------
   Total: 3
   ```

   In this example, the snapshot ‘example_sync_snap’ is the Snapshot Alias defined in the SyncIQ policy on the source cluster.

2. From the snapshot list, view details for the Snapshot Alias by using the ‘Snapshot ID’. In this example, the Snapshot Alias ‘example_sync_snap’ maps to ID ‘3’:

   ```bash
   isi9-s2-n1-l# isi snapshot snapshots view 3
   ID: 3
   Name: example_sync_snap
   Path: /ifs/data/cls1
   Has Locks: No
   Schedule: -
   Alias Target ID: 2
   Alias Target Name: SIQisi9-s1-n1-foo-2020-06-25_11-08-10
   Created: 2020-06-25T11:08:13
   Expires: -
   Size: 4.00k
   Shadow Bytes: 0.00
   % Reserve: 0.00%
   % Filesystem: 0.00%
   State: active
   ```

3. Currently, the Snapshot Alias is pointing to the “Alias Target ID” 2. After running the SyncIQ policy again, the snapshots list is updated with the new snapshot:

   ```bash
   isi9-s2-n1-l# isi snapshot snapshots list
   ID   Name                                   Path
   -------------------------------------------
   2    SIQisi9-s1-n1-foo-2020-06-25_11-08-10 /ifs/data/cls1
   3    example_sync_snap                      /ifs/data/cls1
   7    SIQ-Failover-foo-2020-06-25_11-15-28 /ifs/data/cls1
   9    SIQisi9-s1-n1-foo-2020-06-25_11-15-31 /ifs/data/cls1
   -------------------------------------------
   Total: 4
   ```
4. Viewing more details on the Snapshot Alias, confirms it is pointing to the newest, or HEAD snapshot:

```
isi9-s2-n1-1# isi snapshot snapshots view 3
  ID: 3
  Name: example_sync_snap
  Path: /ifs/data/cls1
  Has Locks: No
  Schedule: -
  Alias Target ID: 9
  Alias Target Name: SIQ-isi9-s1-n1-foo-2020-06-25_11-15-31
  Created: 2020-06-25T11:08:13
  Expires: -
  Size: 4.00k
  Shadow Bytes: 0.00
  % Reserve: 0.00%
  % Filesystem: 0.00%
  State: active
```

The Snapshot Alias is now pointing the “Alias Target ID” 9, which is the newest, or HEAD snapshot.

6.9 Advanced settings

SyncIQ Advanced Settings provide several options to configure a SyncIQ policy, as displayed in Figure 22.

```
Advanced Settings

- Priority
  Normal (default)

- Log Level
  Notice

- Validate file integrity

- Prepare policy for accelerated failback performance

- Keep Reports For
  1 Year

- Record Deletions on Synchronization
  - Do not record when a synchronization deletes files or directories
  - Record when a synchronization deletes files or directories

- Deep Copy for CloudPools
  Deny
```

Figure 22  SyncIQ Policy Advanced Settings

6.9.1 Priority

From the “Priority” drop-down, as displayed in Figure 22, select a priority level for the SyncIQ policy. PowerScale SyncIQ provides a mechanism to prioritize particular policies. Policies can optionally have a priority setting – policies with the priority bit set will start before unprioritized policies. If the maximum number of jobs are running, and a prioritized job is queued, the shortest running unprioritized job will be paused by the system to allow the prioritized job to run. The paused job will then be started next.
Alternatively, to set the priority bit for a job from the CLI, use `--priority 1` on the `isi sync policies create` or `modify` command, which maps to “High” in the web interface. The default is 0, which is unprioritized, which maps to “Normal” in the web interface.

6.9.2 Log Level
From the “Log Level” drop-down, as displayed in Figure 22, specify a level of logging for this SyncIQ policy. The log level may be modified as required during a specific event.

SyncIQ logs provide detailed job information. To access the logs, connect to a node and view its `/var/log/isi_migrate.log` file. The output detail depends on the log level, with the minimal option being “Fatal” and the maximum logging option being “Trace”.

**Note**: Notice is the default log level and is recommended for most SyncIQ deployments. It logs job-level and process-level activity, including job starts and stops, as well as worker coordination information. Debug and Trace options should only be used temporarily as they create a significant number of logs.

6.9.3 Validate file integrity
The “Validate File Integrity” checkbox, as displayed in Figure 22, provides an option for OneFS to compare checksums on SyncIQ file data packets pertaining to the policy. In the event a checksum value does not match, OneFS attempts to transmit the data packet again.

6.9.4 Prepare policy for accelerated failback performance
PowerScale SyncIQ provides an option for an expedited failover process by running a ‘domainmark’ process. The data must be prepared for failover the very first time that the policy runs. This step only needs to be performed once for a policy and can take several hours or more to complete, depending on the policy and dataset. This step marks the data in the source directory to indicate it is part of the failover domain.

The “Prepare Policy for Accelerated Failback Performance” checkbox, as displayed in Figure 22, enables the domainmark process to run automatically when the policy syncs with the target. Running this automatically is an alternative to manually running it with the following command:

```
# isi job start DomainMark --root=<patm> --dm-type=synciq
```

**Note**: As a best practice, it is recommended to select the “Prepare Policy for Accelerated Failback Performance” checkbox during the initial policy configuration, minimizing downtime during an actual outage where time is of the essence. If an existing policy does not have this option selected, it may be selected retroactively, otherwise execute the CLI command before the first failover is required, to avoid extending the failover time.

To enable the accelerated failback from the CLI, set the `--accelerated-failback true` option either on policy creation or subsequently by modifying the policy. The domainmark job will run implicitly the next time the policy syncs with the target.

**Note**: The “Prepare Policy for Accelerated Failback Performance” option will increase the overall execution time of the initial sync job. After the initial sync, SyncIQ performance is not impacted.
6.9.5 Keep reports duration
The “Keep Reports” option, as displayed in Figure 22, defines how long replication reports are retained in OneFS. Once the defined time has exceeded, reports are deleted.

6.9.6 Record deletions on synchronization
Depending on the IT administration requirements, a record of deleted files or directories on the target cluster may be required. By default, OneFS does not record when files or directories are deleted on the target cluster. However, the “Record Deletions on Synchronization” option, as displayed in Figure 22, can be enabled if it is required.

6.9.7 Deep copy for CloudPools
PowerScale clusters that are using CloudPools to tier data to a cloud provider have a stub file, known as a SmartLink, that is retained on the cluster with the relevant metadata to retrieve the file at a later point. Without the SmartLink, a file that is tiered to the cloud, cannot be retrieved. If a SmartLink is replicated to a target cluster, the target cluster must have CloudPools active with the same configuration as the source cluster, to be able to retrieve files tiered to the cloud. For more information about SyncIQ and CloudPools, refer to Section 13, SyncIQ and CloudPools.

‘Deep Copy’ is a process that retrieves all data that is tiered to a cloud provider on the source cluster, allowing all the data to be replicated to the target cluster. Depending on if the target cluster has the same CloudPools configuration as the source cluster, ‘Deep Copy’ could be required. However, in certain workflows, ‘Deep Copy’ may not be required, as the SmartLink file allows for the retrieval of files tiered to the cloud.

The “Deep Copy for CloudPools” drop-down, as displayed in Figure 22, provides the following options:

- Deny: This is the default setting, allowing only the SmartLinks to be replicated from the source to the target cluster, assuming the target cluster has the same CloudPools configuration.
- Allow: This option also replicates the SmartLinks from the source to the target cluster, but this option also check the SmartLinks versions on both clusters. If a mismatch is found between the versions, the complete file is retrieved from the cloud on the source, and then replicated to the target cluster.
- Force: This option required CloudPools to retrieve the complete file from the cloud provider on to the source cluster and replicates the complete file to the target cluster.

**Note:** ‘Deep Copy’ takes significantly more time and system resources when enabled. It is recommended that ‘Deep Copy’ only be enabled if it is required for a specific workflow requirement.

6.10 Assess sync
SyncIQ can conduct a trial run of a policy without actually transferring file data between locations; this is referred to as an “Assess Sync”. Not only does an “Assess Sync” double-check the policy configuration, but it also provides an indication of the time and the level of resources an initial replication policy is likely to consume. This functionality is only available immediately after creating a new policy before it has been run for the first time. To run an “Assess Sync”, from the SyncIQ Policies tab, click “More” for the appropriate policy, and select “Assess Sync”, as displayed in Figure 23.
Configuring a SyncIQ policy

**Note:** As a best practice, it is recommended to run an “Assess Sync” to confirm the policy configuration and resource commitment prior to the replication requirement of the policy.
Impacts of modifying SyncIQ policies

SyncIQ policies may be modified and updated through the CLI or the web interface. The impact of the change is dependent upon how the policy is modified. Rather than modifying or deleting a policy when a suspension is required, the policy may also be disabled, allowing for it to be re-enabled with minimal impact at a later point.

After a policy is configured and the policy has run, SyncIQ will run either the initial replication again or a differential replication if the following variables are modified:

- Source directory
- Included or excluded directories
- File criteria: type, time, and regular expressions
- Target cluster, even if the new target cluster is identical to the old one
  - IP and DNS changes will not trigger a full replication. However, if the cluster GUID changes, the job will fail at runtime. Also, unlike the other settings, a manual reset of the affected policy is required in order to be able to run an associated job.

- Target directory

If a SyncIQ replication policy is deleted, replication jobs will not be created for the policy. Any snapshots and reports associated with the policy are also deleted. The target cluster will break the association to the source cluster, removing the local target entry, and the target directory will allow writes.
SyncIQ performance rules

Performance Rules provide several options for administrators to define limits of resource consumption for SyncIQ policies during specific times or continuously. Setting performance limits allows for minimal impact to high priority workflows but allows nodes to participate in replication within a defined set of resources.

SyncIQ uses aggregate resources across the cluster to maximize replication performance, thus potentially affecting other cluster operations and client response. The default performance configurations, number of workers, network use, and CPU consumption may not be optimal for certain data sets or the processing needs of the business. CPU and network use are set to ‘unlimited’ by default. However, SyncIQ allows administrators to control how resources are consumed and balance replication performance with other file system operations by implementing a number of cluster-wide controls. Rules are created to define available resources for SyncIQ policies for different time periods.

To view or create SyncIQ Performance Rules from the OneFS web interface, click Data Protection > SyncIQ and select the “Performance Rules” tab. Existing Performance Rules are displayed. Click “Create a SyncIQ Performance Rule”, to add a new rule, as displayed in Figure 24.

Figure 24 Creating a SyncIQ performance rule

From the Rule Type drop-down menu, select one of the following options:

- **Bandwidth**: This option provides a limit on the maximum amount of network bandwidth a SyncIQ policy can consume. Once “Bandwidth” is selected the “Limit” field changes to kb/s. In the “Limit” field, specify the maximum allowable bandwidth in kb/s.
- **File Count**: This option allows administrators to define a maximum number of files that replication jobs can send per second. Once “File Count” is selected, the “Limit” field changes to files/sec. In the “Limit” field, specify the maximum allowable files/sec.
SyncIQ performance rules

- CPU: This option limits the CPU consumption to a percentage of the total available. Once “CPU” is selected, the “Limit” field changes to “%”. In the “Limit” field, specify the maximum allowable “%” for the maximum CPU consumption.
- Workers: This option limits the number of workers available to a percentage of the maximum possible. Once “Workers” is selected, the “Limit” field changes to “%”. In the “Limit” field, specify the maximum percentage of workers.

These performance rules will apply to all policies executing during the specified time interval.

Node participation in a SyncIQ policy may be limited as described in section 6.6, Restricting SyncIQ source nodes, and section 6.7.1, Target cluster SmartConnect zones.

**Note:** While SyncIQ allows multiple Performance Rules to be created, it is important to recognize not all rules are applicable to every workflow and consider the impact on RPO times. Depending on the RPO requirements, a Performance Rule could severely impact replication times. In the initial implementation of rules, it is recommended to start with high maximum limits and gradually reduce as RPO times are monitored.

For more information about SyncIQ performance tuning, refer to Section 16.5, Optimizing SyncIQ performance.
SnapshotIQ and SyncIQ

9

SnapshotIQ and SyncIQ

OneFS provides an option to replicate a specific point-in-time dataset with SyncIQ. By default, SyncIQ creates a snapshot automatically at the start of a job. An example use case for this is when a specific dataset is required to replicate to multiple target clusters. A separate policy must be configured for each target cluster, resulting in each policy taking a separate snapshot and the snapshot could be composed of a different dataset. Unless the policies start at the same time and depending on how quickly the source is modified, each target cluster could have a different dataset. Therefore, complicating administrator management of multiple clusters and policies, as each cluster has a different dataset.

As stated in Section 6.3.4, Whenever a snapshot of the source directory is taken, SyncIQ policies provide an option for triggering a replication policy when a snapshot of the source directory is completed. Additionally, at the onset of a new policy configuration, when the “Whenever a Snapshot of the Source Directory is Taken” option is selected, a checkbox appears to sync any existing snapshots in the source directory.

Depending on the IT administrative workflows, triggering replication automatically after a snapshot may simplify tasks. However, if snapshots are scheduled to run on a schedule, this could trigger SyncIQ to run at a higher frequency than required consuming cluster resources. Limiting automatic replication based on a snapshot may be a better option.

SyncIQ offers many options to utilize snapshots. After reviewing this section, also reference Section 6.8, Target snapshots and Section 10.2, Snapshots and SyncIQ policies.

9.1 Specifying snapshots for replication

If a specific dataset must be restored to a specific point-in-time, SyncIQ supports importing a manually taken snapshot with SnapshotIQ for use by a policy. Importing and selecting the snapshot of a policy ensures administrators control the target cluster’s dataset by selecting the same snapshot for multiple policies.

To start a SyncIQ policy with a specified snapshot, use the following command:

```bash
isi sync jobs start <policy-name> [--source-snapshot <snapshot>]
```

The command replicates data according to the specified SnapshotIQ snapshot, as only selecting a snapshot from SnapshotIQ is supported. Snapshots taken from a SyncIQ policy are not supported. When importing a snapshot for policy, a SyncIQ snapshot is not generated for this replication job.

**Note:** The root directory of the specified snapshot must contain the source directory of the replication policy. This option is valid only if the last replication job completed successfully or if a full or differential replication is executed. If the last replication job completed successfully, the specified snapshot must be more recent than the snapshot referenced by the last replication job.

When snapshots are replicated to the target cluster, by default, only the most recent snapshot is retained, and the naming convention on the target cluster is system generated. However, in order to prevent only a single snapshot being overwritten on the target cluster and the default naming convention, select the “Enable capture of snapshots on the target cluster” as stated in Section 6.8, Target snapshots. Once this checkbox is selected, specify a naming pattern and select the “Snapshots do not expire” option. Alternatively, specify a date for snapshot expiration. Limiting snapshots from expiring ensures they are retained on the target cluster rather than overwritten when a newer snapshot is available. The target cluster snapshot options map to --target-snapshot-archive, --target-snapshot-alias, --target-snapshot-expiration, and --target-snapshot-pattern in the CLI.
9.2 Archiving SnapshotIQ snapshots to a backup cluster

Specifying a snapshot to replicate from is also an option for cases where SnapshotIQ snapshots are consuming a significant amount of space on a cluster. The snapshots must be retained for administrative requirements. In this case, the snapshots are replicated to a remote backup or disaster recovery cluster, opening additional space on the source cluster.

When replicating SnapshotIQ snapshots to another cluster, the dataset and its history must be replicated from the source cluster. Therefore, snapshots are replicated from the source in chronological order, from the first snapshot to the last. The snapshots are placed into sequential jobs replicating to the target cluster. Replicating in this process, allows the target cluster to create a snapshot with a delta between each job, as each job replicates a snapshot that is more up-to-date than the previous.

**Note:** As stated in Section 9.1, Specifying snapshots for replication, ensure target snapshots are configured and retained prior to initiating the archiving process.

If snapshots are not archived in chronological order, an error occurs, as displayed in Figure 25.

![Image](image.png)

**Figure 25** Out-of-order snapshots create Sync Policy Error

To ensure SyncIQ retains the multiple snapshots required to recreate the dataset, Snapshot IQ must be installed with archival snapshots enabled.

Once all snapshots are replicated to the target cluster, an archive of the source cluster’s snapshots is complete. The source cluster’s snapshots may now be deleted, creating additional space.

**Note:** Archiving snapshots creates a new set of snapshots on the target cluster based on the source cluster snapshots, but it does not “migrate” the snapshots from one cluster to another. The new snapshots have the same data, but with different data times. This may not meet compliance requirements for ensuring data integrity or evidentiary requirements.

9.3 Target cluster SnapshotIQ snapshots

Although a SyncIQ policy configures the target directory as read-only, SnapshotIQ snapshots are permitted. As a best practice, consider configuring target cluster SnapshotIQ snapshots at a differing schedule than the source cluster, providing an additional layer of data protection and a point-in-time dataset. Target snapshots could also be utilized as a longer-term retention option if the cost of storage space is less than that of the source cluster. In this arrangement, the source cluster snapshots are retained short term, target cluster SyncIQ snapshots are medium term, and the long-term archive snapshots are SnapshotIQ snapshots on the target cluster.
SyncIQ design considerations

10 SyncIQ design considerations

Prior to configuring data replication policies with SyncIQ, it is recommended to map out how policies align with IT administration requirements. Data replication between clusters is configured based on either entire cluster replication or directory-based replication. Designing the policy to align with departmental requirements ensures policies satisfy requirements at the onset, minimizing policy reconfiguration. When creating policies, Disaster Recovery (DR) plans must be considered, in the event of an actual DR event. DR readiness is a key factor to success during a DR event.

Failover and failback are specific to a policy. In the event of an actual DR event, failing over several policies requires additional time. On the contrary, if entire cluster replication is configured, only a single policy is failed over minimizing downtime. Additionally, consider that clients must be re-directed to the target cluster manually, through either a DNS update or by manual advisement. If entire cluster replication is configured, a single DNS name change will minimize impacts. However, DR steps may not be a concern if Superna Eyeglass is utilized, as explained in section 12, Superna Eyeglass DR Edition.

As policies are created for new departments, it is important to consider policy overlap. Although the overlap does not impact the policy running, the concerns include managing many cumbersome policies and resource consumption. If the directory structure in policies overlap, data is being replicated multiple times impacting cluster and network resources. During a failover, time is a critical asset. Minimizing the number of policies allows administrators to focus on other failover activities during an actual DR event. Additionally, RPO times may be impacted by overlapping policies.

During the policy configuration stage, select options that have been tested in a lab environment. For example, for a synchronize policy configured to run anytime the source is modified, consider the time delay for the policy to run. If this is set to zero, every time a client modifies the dataset, a replication job is triggered. Although this may be required to meet RPO and RTO requirements, administrators must consider if the cluster resources and network bandwidth can meet the aggressive replication policy. Therefore, it is recommended to test in a lab environment, ensuring the replication policy requirements are satisfied. Superna Eyeglass, explained in section 12, Superna Eyeglass DR Edition, provides additional insight into expected RPO and RTO times, based on a policy.

10.1 Considering cluster resources with data replication

As the overall architecture of SyncIQ Policies is designed, other factors to consider are the number of policies running together. Depending on how policies are configured, the cluster may have many policies running at once. If many policies are running together, cluster resources and network bandwidth must be considered. Under standard running conditions, the cluster resources are also providing client connectivity with an array of services running. It is imperative to consider the cluster and network utilization when the policies are running.

Given the number of policies running at the same time, administrators may consider staggering the policies to run a certain number of policies in a specific time period. Policy schedules can be updated to stagger policy requirements and run times, matching policies with the administration requirements.

While considering the number of policies running in a specified time period, the permitted system and network resources may also be tuned to meet administration requirements. OneFS provides options for tuning SyncIQ performance based on CPU utilization, bandwidth, file count, and the number of workers, as discussed in Section 8, SyncIQ performance rules. A higher level of granularity is possible by only allowing certain nodes to participate in data replication, as discussed in Section 6.6, Restricting SyncIQ source nodes. Administrators may also consider assigning a priority to each policy, as discussed in Section 6.9.1, Priority. As policies run, it
SyncIQ design considerations

is crucial to monitor cluster resources through the many available tools, as stated in Section 16, Monitoring, alerting, reporting, and optimizing performance.

10.1.1 **Source and target cluster replication performance**

During the design phase, consider the node types on the source and target cluster impacting the overall data replication performance. When a performance node on the source cluster is replicating to archive nodes on the target cluster, this causes the overall data replication performance to be compromised based on the limited performance of the target cluster’s nodes. For example, if a source cluster is composed of F800 nodes and the target cluster is composed of A200 nodes, the replication performance reaches a threshold, as the A200 CPUs cannot perform at the same level as the F800 CPUs.

Depending on the workflow and replication requirements, the longer replication times may not be a concern. However, if replication performance is time sensitive, consider the node types and associated CPUs on the source and target clusters, as this could bottleneck the overall data replication times.

10.2 **Snapshots and SyncIQ policies**

As snapshots and SyncIQ policies are configured, it is important to consider the scheduled time. As a best practice, it is recommended to stagger the scheduled times for snapshots and SyncIQ policies. Staggering snapshots and SyncIQ policies at different times ensures the dataset is not interacting with snapshots while SyncIQ jobs are running, or vice versa. Additionally, if snapshots and SyncIQ policies have exclusive scheduled times, this ensures the maximum system resources are available, minimizing overall run times. However, system resources are also dependent on any Performance Rules configured, as stated in Section 8, SyncIQ performance rules.

Another factor to consider is the impact on system resources if SyncIQ policies are triggered based on snapshots, as discussed in Section 6.3.4, Whenever a snapshot of the source directory is taken. For example, if a snapshot policy is configured to run every 5 minutes, the policy is triggered when the snapshot completes. Depending on the dataset and the rate of updates, SyncIQ could be far behind the newest snapshot. Additionally, a constant trigger of data replication impacts cluster resources. Consider how the snapshot frequency impacts overall system performance. Alternatively, rather than using snapshot triggered replication, consider manually running a SyncIQ policy with a specified snapshot, as explained in Section 9.1, Specifying snapshots for replication.

For more information about snapshots and SyncIQ, refer to Section 6.8, Target snapshots and Section 9, SnapshotIQ and SyncIQ.

10.3 **Network considerations**

As stated previously in Section 6.7.1, Target cluster SmartConnect zones, SyncIQ only functions under static IP pool allocation strategies. A dynamic allocation of IPs leads to SyncIQ failures.

During data replication, certain SyncIQ packets set the “Do not fragment” (DF) bit, causing the connection to fail if fragmentation is required. A common instance is if jumbo frames are configured on the cluster, but are not supported on all network devices, requiring fragmentation at a specific hop. If jumbo frames are configured, ensure they are supported end-to-end on all hops between the source and target cluster, eliminating the need for fragmentation. Otherwise, set the network subnet used by SyncIQ to an MTU of 1500. For more information about jumbo frames, refer to the PowerScale Network Design Considerations white paper.
For additional information about SyncIQ networking considerations, refer to the “SyncIQ Considerations” section in the PowerScale Network Design Considerations white paper.

### 10.3.1 SyncIQ policy requirement for System Access Zone
During the design phase of SyncIQ policies and network hierarchy, note that SyncIQ is not zone aware, requiring SyncIQ policies and data replication to be aligned with the System Access Zone. If a new SyncIQ policy, or an existing policy, is configured for anything other than the System Access Zone, the configuration fails with an error message. The SyncIQ requirement for this zone applies to the source and target clusters. Taking this requirement into account during the design phase allows administrators to plan policies, subnets, and pools accordingly, if SyncIQ replication must be limited to a set of nodes and interfaces.

### 10.3.2 Network ports
For a list of network ports used by SyncIQ, refer to the OneFS 8.2.1 Security Configuration Guide.

### 10.4 Jobs targeting a single directory tree
Creating SyncIQ policies for the same directory tree on the same target location is not supported. For example, consider the source directory `/ifs/data/users`. Creating two separate policies on this source to the same target cluster is not supported:

- one policy excludes `/ifs/data/users/ceo` and replicates all other data in the source directory
- one policy includes only `/ifs/data/users/ceo` and excludes all other data in the source directory

Splitting the policy with this format is not supported with the same target location. It would only be supported with different target locations. However, consider the associated increase in complexity required in the event of a failover or otherwise restoring data.

### 10.5 Authentication integration
UID/GID information is replicated, via SID numbers, with the metadata to the target cluster. It does not require to be separately restored on failover.

### 10.6 SyncIQ and Hadoop Transparent Data Encryption
OneFS 8.2 introduces support for Apache® Hadoop® Distributed File System (HDFS) Transparent Data Encryption (TDE), providing end-to-end encryption between HDFS clients and a PowerScale cluster. HDFS TDE is configured in OneFS through encryption zones where data is transparently encrypted and decrypted as data is read and written. For more information about HDFS TDE for OneFS, refer to the blog post Using HDFS TDE with PowerScale OneFS.

SyncIQ does not support the replication of the TDE domain and keys. Therefore, on the source cluster, if a SyncIQ policy is configured to include an HDFS TDE directory, the encrypted data is replicated to the target cluster. However, on the target cluster, the encrypted data is not accessible as the target cluster is missing the metadata that is stored in the IFS domain for clients to decrypt the data. TDE ensures the data is encrypted before it is stored on the source cluster. Also, TDE stores the mapping to the keys required to decrypt the data, but not the actual keys. This makes the encrypted data on the target cluster inaccessible.
10.7 Small File Storage Efficiency (SFSE) and SyncIQ

OneFS’ Small File Storage Efficiency (SFSE) provides a feature for small files in OneFS, packing them into larger files, resulting in increased storage efficiency. If a SyncIQ policy is configured for a SFSE dataset, the data is replicated to the target cluster. However, the SFSE dataset is unpacked on the source cluster prior to replication. If the target cluster has SFSE enabled, the dataset is packed when the next SmartPools job runs on the target cluster. If the target cluster does not have SFSE enabled, the dataset remains unpacked.
11 Failover and failback

This section provides an explanation of the failover and failback processes. For a detailed set of instructions, refer to appendix A, Failover and failback Steps.

Under normal operation, SyncIQ target directories can be written to only by the SyncIQ job itself – all client writes to any target directory are disabled, this is referred to as a protected replication domain. In a protected replication domain, files cannot be modified, created, deleted or moved within the target path of a SyncIQ job.

SyncIQ provides built-in recovery to the target cluster with minimal interruption to clients. By default, the RPO (recovery point objective) is to the last completed SyncIQ replication point. Optionally, with the use of SnapshotIQ, multiple recovery points can be made available, as explained in Section 9, SnapshotIQ and SyncIQ.

Note: SyncIQ Failover and Failback does not replicate cluster configurations such as SMB shares and NFS exports, quotas, snapshots, and networking settings, from the source cluster. PowerScale does copy over UID/GID ID mapping during replication. In the case of failover to the remote cluster, other cluster configurations must be configured manually. An application such as Superna Eyeglass can be used to replicate the configuration information, as discussed in Section 12, Superna Eyeglass DR Edition.

11.1 Failover

In the event of a planned or unplanned outage to the source cluster, a failover is the process of directing client traffic from the source cluster to the target cluster. An unplanned outage of the source cluster could be a disaster recovery scenario where the source cluster no longer exists, or it could be unavailable if the cluster is not reachable.

On the contrary, a planned outage is a coordinated failover, where an administrator knowingly makes a source cluster unavailable for disaster readiness testing, cluster maintenance, or other planned event. Prior to performing a coordinated failover, ensure a final replication is completed prior to starting, ensuring the dataset on the target matches the source.

To perform a failover, set the target cluster or directory to Allow Writes.

Note: As a best practice, configure DNS to require single forwarding change only. During an outage, this minimizes downtime and simplifies the failover process.
11.1.1 Failover while a SyncIQ job is running

It is important to note that if the replication policy is running at the time when a failover is initiated, the replication job will fail, allowing the failover to proceed successfully. The data on the target cluster is restored to its previous state before the replication policy ran. The restore completes by utilizing the snapshot taken by the replication job after the last successful replication job.

11.2 Target cluster dataset

If for any reason the source cluster is entirely unavailable, for example, under a disaster scenario, the data on the target cluster will be in the state after the last successful replication job completed. Any updates to the data since the last successful replication job are not available on the target cluster.

11.3 Failback

Users continue to read and write to the target cluster while the source cluster is repaired. Once the source cluster becomes available again, the administrator decides when to revert client I/O back to it. To achieve this, the administrator initiates a SyncIQ failback, which synchronizes any incremental changes made to the target cluster during failover back to the source. When complete, the administrator redirects client I/O back to the original cluster again.

Failback may occur almost immediately, in the event of a functional test, or more likely, after some elapsed time during which the issue which prompted the failover can be resolved. Updates to the dataset while in the failover state will almost certainly have occurred. Therefore, the failback process must include propagation of these back to the source.

Failback consists of three phases. Each phase should complete before proceeding.

11.3.1 Resync-prep

Run the preparation phase (resync-prep) on the source cluster to prepare it to receive intervening changes from the target cluster. This phase creates a read-only replication domain with the following steps:

- The last known good snapshot is restored on the source cluster.
- A SyncIQ policy is created on the target policy appended with ‘_mirror’. This policy is used to failback the dataset with any modification that has occurred since the last snapshot on the source cluster. During this phase, clients are still connected to the target.

11.3.2 Mirror policy

Run the mirror policy created in the previous step to sync the most recent data to the source cluster.

11.3.3 Verify

Verify that the failback has completed, via the replication policy report, and redirect clients back to the source cluster again. At this time, the target cluster is automatically relegated back to its role as a target.
11.4 Allow-writes compared to break association

Once a SyncIQ policy is configured between a source and target cluster, an association is formed between the two clusters. OneFS associates a policy with its specified target directory by placing a cookie on the source cluster when the job runs for the first time. The cookie allows the association to persist, even if the target cluster’s name or IP address is modified. SyncIQ provides two options for making a target cluster writable after a policy is configured between the two clusters. The first option is to ‘Allow-Writes’, as stated previously in this section. The second option to make the target cluster writeable, is to break a target association.

If the target association is broken, the target dataset will become writable, and the policy must be reset before the policy can run again. A full or differential replication will occur the next time the policy runs. During this full resynchronization, SyncIQ creates a new association between the source and its specified target.

In order to perform a Break Association, from the target cluster’s CLI, execute the following command:

```bash
isi sync target break -policy=[Policy Name]
```

**Note:** Practice caution prior to issuing a policy break command. Ensure the repercussions are understood as explained in this section.

To perform this from the target cluster’s web interface, select Data Protection > SyncIQ and select the “Local Targets” tab. Then click “More” under the “Actions” column for the appropriate policy, and click “Break Association”, as displayed in Figure 27.

![Figure 27: Break association from web interface](image)

On the contrary, the ‘Allow-Writes’ option does not result in a full or differential replication to occur after the policy is active again, as the policy is not reset.

Typically, breaking an association is useful to temporary test scenarios or if a policy has become obsolete for various reasons. Allowing writes is useful for failover and failback scenarios. Typical applications of both options are listed in Table 1.

<table>
<thead>
<tr>
<th>Allow-writes</th>
<th>Break association</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failover and failback</td>
<td>Temporary test environments</td>
</tr>
<tr>
<td>Temporarily allowing writes on a target cluster, while the source is restored</td>
<td>Obsolete SyncIQ policies</td>
</tr>
<tr>
<td>Once the source cluster is brought up, it does not require a full or differential replication, depending on the policy</td>
<td>Data migrations</td>
</tr>
<tr>
<td></td>
<td>Once the source cluster is brought up, it requires a full or differential replication</td>
</tr>
</tbody>
</table>
As with any major IT implementation, it is recommended to test all functions in a lab environment, rather than a production environment to understand how each function performs.
12 Superna Eyeglass DR Edition

Many SyncIQ failover and failback functions can be automated with additional features through Superna Eyeglass® DR Edition. Superna provides software that integrates with PowerScale, delivering disaster recovery automation, security, and configuration management. Dell EMC sells Superna software as a Select partner.

Superna’s DR Edition supports PowerScale SyncIQ by automating the failover process. Without Superna the failover process requires manual administrator intervention. Complexity is minimized with DR Edition as it provides one-button failover, but also updates Active Directory, DNS, and client data access, as illustrated in Figure 28.

Once DR Edition is configured it continually monitors the PowerScale cluster for DR readiness through auditing, SyncIQ configuration, and several other cluster metrics. The monitoring process includes alerts and steps to rectify discovered issues. In addition to alerts, DR edition also provides options for DR testing, which is highly recommended, ensuring IT administrators are prepared for DR events. The DR testing can be configured to run on a schedule. For example, depending on the IT requirements, DR testing can be configured to run on a nightly basis, ensuring DR readiness.

As DR Edition collects data, it provides continuous reports on RPO compliance, ensuring data on the target cluster is current and relevant.

### 13 SyncIQ and CloudPools

OneFS SyncIQ and CloudPools features are designed to work together seamlessly. CloudPools tiers data to a cloud provider. The cloud provider could be Dell EMC’s Elastic Cloud Storage (ECS), a public, private, or hosted cloud. As data is tiered to a cloud provider, a small file is retained on the cluster, referred to as a SmartLink, containing the relevant metadata to retrieve the file at a later point. A file that is tiered to the cloud, cannot be retrieved without the SmartLink file. For more information about CloudPools, refer to [PowerScale OneFS CloudPools Administration Guide](#) or the [PowerScale CloudPools and ECS Solution Guide](#).

If a directory on the source cluster is configured for data replication to a target cluster containing the SmartLink files, the SmartLink files are also replicated to the target cluster.

**Note:** Although configuration to a cloud provider exists on the source and target clusters, it is important to understand that only a single cluster may have **read and write access** to the cloud provider. Both the source and target cluster have read access, but only a single cluster may have read and write access.

During normal operation, the source cluster has read-write access to the cloud provider, while the target cluster is read-only, as illustrated in Figure 29.

![Diagram](#)

**Figure 29** PowerScale SyncIQ and CloudPools with ECS

#### 13.1 CloudPools failover and failback implications

SyncIQ provides a seamless failover experience for clients. The experience does not change if CloudPools is configured. After a failover to the target cluster, clients continue accessing the data stored at the cloud provider without interruption to the existing workflow. The target cluster has read-only access to the specified cloud provider, as clients request files stored in the cloud the target cluster retrieves these files with the SmartLinks and delivers them in the same method the source cluster did.

However, if the files are modified those changes are not propagated to the cloud provider. Instead, any changes to the cloud tiered files are stored locally in the target cluster’s cache. When the failback is complete to the source cluster, the new changes to the cloud tiered files are sent to the source cluster. The source cluster then propagates the changes to the cloud provider.

If a failover is permanent, or for an extended period of time, the target cluster requires read-write access to the cloud provider. The read-write status is updated through the `isi cloud access` command. For more information about this command, refer to the administration and solution guide referenced above.
13.2 **Target cluster SyncIQ and CloudPools configuration**

Irrespective of when CloudPools is configured on the source cluster, the cloud provider account information, CloudPools, and filepool policy are automatically configured on the target cluster.

13.2.1 **CloudPools configured prior to a SyncIQ policy**

Configuring CloudPools prior to creating a SyncIQ policy is a supported option. When the SyncIQ policy runs for the first time it checks if the specified source directory contains SmartLink files.

If SmartLink files are found in the source directory, on the target cluster SyncIQ performs the following:

- Configures the cloud storage account and CloudPools matching the source cluster configuration
- Configures the file pool policy matching the source cluster configuration

Although the target cluster is configured for the same cloud provider using CloudPools, it only has read access to the provider.

13.2.2 **CloudPools configured after a SyncIQ policy**

An existing SyncIQ policy also supports the replication of SmartLink files. If the SyncIQ policy is already configured and active, the source directory could be updated to work with CloudPools. After the CloudPools configuration is complete, the following SyncIQ job detects the SmartLink files on the source.

In this case, once the SmartLink files are detected in the source directory, on the target cluster SyncIQ performs the following:

- Configures the cloud storage account and CloudPools matching the source cluster configuration
- Configures the file pool policy matching the source cluster configuration

Although the target cluster is configured for the same cloud provider using CloudPools, it only has read access to the provider.

---

**Note:** As a best practice, prior to configuring CloudPools on a source cluster directory, temporarily disable the associated SyncIQ policy. After updating the source cluster directory for CloudPools, enable the SyncIQ policy, allowing the next job to detect the SmartLink files and configure the target cluster accordingly.
14 SyncIQ security

By default, SyncIQ starts replication to a target PowerScale cluster specified without any configuration necessary on the target cluster. The replication policy is configured on the source cluster only, and if network connectivity is available through the front-end ports, the replication policy is initiated.

Depending on the network architecture hierarchy and where the PowerScale clusters are placed in the hierarchy, this could be a concern. For instance, a cluster could receive many replication policies from a source cluster that could overwhelm its resources. In environments where several PowerScale clusters are active, an administrator may inadvertently specify the IP address of another cluster rather than the intended target cluster.

Securing a PowerScale cluster from unauthorized replication of data is performed through two available options. As a best practice and per DSA-2020-039, Dell EMC PowerScale OneFS Security Update for a SyncIQ Vulnerability, enabling SyncIQ encryption, is recommended, preventing man-in-the-middle attacks and alleviating security concerns. SyncIQ encryption was introduced in OneFS 8.2.

SyncIQ is disabled by default on greenfield OneFS release 9.1 clusters. Once SyncIQ is enabled, the global encryption flag is enabled, requiring all SyncIQ policies to be encrypted. For PowerScale clusters upgraded to OneFS 9.1, the global encryption flag is also enabled. However, the global encryption flag is not enabled on PowerScale clusters upgraded to OneFS 9.1 with an existing SyncIQ policy.

As an alternative for PowerScale clusters running a release prior to OneFS 8.2, a SyncIQ pre-shared key (PSK) can be configured, protecting a cluster from unauthorized replication policies without the PSK. For more information about SyncIQ PSK, refer to Section 14.2, SyncIQ pre-shared key.

14.1 SyncIQ encryption

OneFS release 8.2 introduced over-the-wire, end-to-end encryption for SyncIQ data replication, protecting and securing in-flight data between clusters. A global setting is available enforcing encryption on all incoming and outgoing SyncIQ policies.

Note: Prior to enabling SyncIQ encryption on a production cluster, test in a lab environment that mimics the production environment. Encryption adds minimal overhead to the transmission, but it may impact a production workflow depending on the network bandwidth, cluster resources, workflow, and policy configuration. Only after successfully testing encryption in a lab environment and collecting satisfactory measurements, may the production cluster be considered for implementing SyncIQ encryption.
SyncIQ provides encryption through the use of X.509 certificates paired with TLS version 1.2 and OpenSSL version 1.0.2o. The certificates are stored and managed in the source and target cluster’s certificate stores, as illustrated in Figure 30. Encryption between clusters is enforced by each cluster, storing its certificate and its peer’s certificate. Therefore, the source cluster is required to store the target cluster’s certificate, and vice versa. Storing the peer’s certificate essentially creates a white list of approved clusters for data replication. SyncIQ encryption also supports certificate revocation through the use of an external OCSP responder.

**Note:** Both the source and target cluster must be upgraded and committed to OneFS release 8.2 or newer, prior to enabling SyncIQ encryption.

### 14.1.1 Configuring SyncIQ encryption

OneFS release 9.1 introduced support for SyncIQ configuration through the WebUI. For releases prior to OneFS 9.1, SyncIQ encryption configuration is available through the CLI only. To configure SyncIQ encryption between a source and target cluster, perform the following:

1. Utilizing publicly available tools, create X.509 certificates for the source and target cluster. This process results in a certificate authority certificate, source certificate, source private key, target certificate, and target private key.

   Certain certificate authorities do not generate the public and private key pairs. In that case, the public and private key pairs must be manually generated with a Certificate Signing Request (CSR), requiring a manually generated CSR. To manually generate the CSR file, as an example, use the following:

   ```bash
   openssl req -new -newkey rsa:2048 -keyout <src_key> -out <src_csr>
   ```

   Next, provide the CSR file for each cluster to the certificate authority, and signed certificates are returned.

   **Note:** The certificates should be configured for use in TLS connections with client authentication enabled. They must be signed by a certificate authority and be able to act as both a client and a server certificate. Certificate extensions are neither required nor recommended, as this results in additional restrictions and may cause SyncIQ policies to fail.

   The procedure explained in this step with the certificate authority is the recommended process. Alternatively, for environments where a Certificate Authority is not available, a self-signed certificate
SyncIQ security

can be used for SyncIQ encryption. To configure SyncIQ encryption with a self-signed certificate, refer to Appendix B, SyncIQ encryption with self-signed certificates.

2. Add the certificates created in step 1 to the source cluster certificate store, using the following commands:

   ```shell
   isi sync cert server import <src_cert_id> <src_key>
   isi sync cert peer import <tgt_cert_id> --name=[Specify a certificate name]
   isi cert authority import <ca_cert_id> --name=[Specify the authority name]
   ```

   Alternatively, upload the source server certificate through the WebUI, under Data Protection > SyncIQ > Settings. Under the “Server Certificates” section, click “Add certificate” and upload the source cluster certificate, as displayed in Figure 31.

   ![Source cluster certificates](Image)

   **Figure 31** Source cluster certificates

3. This step may be skipped if encryption is configured through the WebUI. On the source cluster, activate the SyncIQ cluster certificate from Step 1, using the following command:

   ```shell
   isi sync settings modify --cluster-certificate-id=<src_cert_id>
   ```

4. Add the certificates created in step 1 to the target cluster certificate store, using the following commands:

   ```shell
   isi sync cert server import <tgt_cert_id> <tgt_key>
   isi sync cert peer import <src_cert_id> --name=[Specify a certificate name]
   isi cert authority import <ca_cert_id> --name=[Specify the authority name]
   ```

   Alternatively, upload the target server certificate through the WebUI, under Data Protection > SyncIQ > Certificates. Under the “Import target cluster certificates” section, click “Add certificate” and upload the target cluster certificate, as displayed in Figure 32.

   ![Target cluster certificates](Image)

   **Figure 32** Target cluster certificates

5. This step may be skipped if encryption is configured through the WebUI. On the target cluster, activate the SyncIQ cluster certificate from step 1, using the following command:

   ```shell
   isi sync settings modify --cluster-certificate-id=<tgt_cert_id>
   ```

6. A global option is available, requiring that all incoming and outgoing SyncIQ policies are encrypted.

   **Note:** Executing this command impacts existing SyncIQ policies that may not have encryption enabled. Only execute this command once all existing policies have encryption enabled. Otherwise, existing policies that do not have encryption enabled will fail.
To enable this, execute the following command:

`isi sync settings modify --encryption-required=True`

7. To modify an existing SyncIQ policy for encryption, proceed to Step 0. Otherwise, to create a new encrypted SyncIQ policy, on the source cluster, use the following command:

`isi sync policies create <SyncIQ Policy Name> sync <Source Cluster Directory> <Target Cluster IP Address> <Target Cluster Directory> --target-certificate-id=<tgt_cert_id>`

Alternatively, from the WebUI, navigate to **Data Protection > SyncIQ > Policies**, and click “Create a SyncIQ policy.” Specify the policy name, source cluster directory, target cluster IP address, target cluster directory, and target cluster certificate, as illustrated in Figure 33.

![Figure 33](image)

8. To modify an existing SyncIQ policy for encryption, on the source cluster, use the following command:

`isi sync policies modify <pol_name> --target-certificate-id=<tgt_cert_id>`

Alternatively, from the WebUI, navigate to **Data Protection > SyncIQ > Policies**, select an existing SyncIQ policy, and click “View / Edit” on the right. From the “Target certificate” drop-down, select the appropriate target cluster certificate.

14.1.2 Other optional commands

SyncIQ provides an option to require a policy to use a specified SSL cipher suite. To update a policy and enforce a specific SSL suite, use the following command:

`isi sync policies modify <pol_name> --encryption-cipher-list=<suite>`

A target cluster may be updated to check the revocation status of incoming certificates with the following command:

`isi sync settings modify --ocsp-address=<OCSP IP Address> --ocsp-issuer-certificate-id=<ca_cert_id>`
SyncIQ security

By default, the encrypted connection is renegotiated on a cluster every eight hours. This value may be updated with the following command:

   isi sync settings modify --renegotiation-period=<Specify time period in hours>

14.1.3 Troubleshooting
As with other SyncIQ policies, errors are documented in the SyncIQ reports. The same applies to SyncIQ encryption as the reason for failure is listed in the report. For instance, if the job failed due to a TLS authentication failure, the error message from the TLS library is provided in the report.

Additionally, for a TLS authentication failure, a detailed log is available in the /var/log/messages directory on the source and target clusters. The log includes the error code and reason for failure, depth at which the failure occurred in the certificate chain, the certificate ID, and the subject name of the certificate that caused the failure.

14.2 SyncIQ pre-shared key
A SyncIQ pre-shared key (PSK) is only configured on the target cluster and limits policies from source clusters if they do not have the PSK configured in the SyncIQ policy.

**Note:** A SyncIQ PSK is only recommended for environments where SyncIQ encryption may not be configured. These environments include clusters running an OneFS version prior to OneFS 8.2 or other environmental factors. For more information about configuring SyncIQ encryption, refer to Section 14.1, SyncIQ encryption.

SmartLock Compliance mode clusters do not support SyncIQ PSK. For clusters in SmartLock Compliance mode, upgrading to OneFS 8.2 or later is recommended and configuring SyncIQ encryption. SmartLock Enterprise mode clusters support SyncIQ PSK.

To configure a SyncIQ pre-shared key (PSK) on a target cluster, perform the following:

1. Ensure SyncIQ jobs are not running. Configuring the PSK will cause all jobs replicating to the target cluster to fail. Prior to proceeding with the SyncIQ PSK configuration, either wait for SyncIQ jobs to complete or cancel running jobs. To manually cancel a SyncIQ job:

   isi sync jobs cancel <policy-name>

   Alternatively, to cancel all SyncIQ jobs:

   isi sync jobs cancel --all

2. Create a file named ‘passwd’, under /ifs/.ifsvar/modules/tsm/. If the file does not already exist, create it with ACLs to limit access:

   touch /ifs/.ifsvar/modules/tsm/passwd
   chmod 700 /ifs/.ifsvar/modules/tsm/passwd

3. In the ‘passwd’ file, specify a single text string limited to 255 characters as the target cluster’s SyncIQ PSK. The PSK must be the only line in the file and cannot contain any spaces or tab characters. Enter the PSK using the ‘vi’ or other utility. As a best practice, ensure this PSK is unique to this system only, ensuring further security.

   vi /ifs/.ifsvar/modules/tsm/passwd
4. After saving the PSK in the ‘passwd’ file, confirm the PSK entry:

```bash
cat /ifs/.ifsvar/modules/tsx/passwd
```

5. Once the PSK is configured on the target cluster, policies on the source cluster must be modified through the CLI

   For OneFS 8.0 and later, use the following command:

   ```bash
  isi sync policies modify [Policy Name] --set-password --password=[Target Cluster PSK specified in ‘passwd’ file]
   ```

   For OneFS 7.1.x or 7.2.x, use the following command:

   ```bash
   isi sync policies modify [Policy Name] --password [Target Cluster PSK specified in ‘passwd’ file]
   ```

   For OneFS 7.0.x and earlier, use the following command:

   ```bash
   isi sync policy modify [Policy Name] --passwd=[Target Cluster PSK specified in ‘passwd’ file]
   ```

   Once the policies on the source cluster are updated, the source cluster does not require any additional configuration. To confirm if the PSK is configured on a source cluster policy, view the policy using `isi sync policies view`, and check the ‘Password Set’ field. A ‘Yes’ should be listed.

   To resume a stopped SyncIQ job, use the following command: `isi sync jobs start [policy-name]`

   If a target cluster has a PSK in place for SyncIQ and the source cluster policy is not configured with the PSK using the `--set-password` flag, the policy will fail. An error is listed under the report, stating ‘Authentication with target failed’, as displayed in Figure 34.

   ![View SyncIQ report details](image)

   **Figure 34**  SyncIQ authentication failed with target cluster

   Optionally, to de-configure the SyncIQ PSK, remove the ‘passwd’ file on the target cluster. Next, modify all policies on the source cluster.

   For OneFS 8.0 and later, use the following command:
isi sync policies modify [policy-name] --set-password --password="<null>"

For OneFS 7.1.x or 7.2.x, use the following command:

isi sync policies modify [policy-name] --password ""

For OneFS 8.0 and later, use the following command:

isi sync policy modify [policy-name] --passwd=""
15 **SyncIQ bandwidth reservations**

Prior to OneFS 8.2, a global bandwidth configuration was available, impacting all SyncIQ policies. The global reservation is then split amongst the running policies. For more information about configuring the SyncIQ global bandwidth reservation, refer to section 8, SyncIQ performance rules.

OneFS 8.2 introduces an option to configure bandwidth reservations on a per policy basis, providing granularity for each policy. The global bandwidth reservation available in previous releases continues in OneFS 8.2. However, this is applied as a combined limit of the policies, allowing for a reservation configuration per policy, as illustrated in Figure 35. As bandwidth reservations are configured, consider the global bandwidth policy which may have an associated schedule.

![Figure 35  SyncIQ bandwidth reservation](image)

**Note:** As bandwidth reservations are configured, it is important to consider that SyncIQ calculates bandwidth based on the bandwidth rule, rather than the actual network bandwidth or throughput available.

15.1 **Bandwidth reservation configuration**

The first step in configuring a per policy bandwidth reservation is to configure a global bandwidth performance rule, as explained in section 8, SyncIQ performance rules. From the CLI, the global bandwidth reservation is configured using the `isi sync rules` command.

Once a global bandwidth reservation is configured, a per policy bandwidth reservation is configured for new policies using the following command:

```
isi sync policy create --bandwidth-reservation=[bits per second]
```

Once a global bandwidth reservation is configured, a per policy bandwidth reservation is configured for existing policies using the following command:

```
isi sync modify create --bandwidth-reservation=[bits per second]
```
15.2 Bandwidth reserve

If a bandwidth reservation is not created for a policy, the bandwidth reserve is applied. The bandwidth reserve is specified as a global configuration parameter, as a percentage of the global configured bandwidth or an absolute limit in bits per second.

Note: If a bandwidth reservation is not configured in OneFS 8.2 for a specific policy, the default bandwidth reserve is 1% of the global configured bandwidth. The default is set at this level to encourage administrators to configure the bandwidth reservation per policy. For clusters upgrading from a previous release to OneFS 8.2, it is important to note that any existing policies default to the 1% bandwidth reservation, assuming a global bandwidth reserve is not configured.

In the case where a bandwidth reservation is not configured for a policy, the bandwidth reserve is applied if sufficient bandwidth is not available. To configure a bandwidth reservation percentage, use the following command:

```
sync settings modify --bandwidth-reservation-reserve-percentage=[% of global bandwidth reservation]
```

To configure a bandwidth reservation in bits per second rather than a percentage, use the following command:

```
isist sync settings modify --bandwidth-reservation-reserve-percentage=[bits per second]
```

Further, to clear a configured bandwidth reserve, use the following command:

```
isist sync settings modify --clear-bandwidth-reservation-reserve
```

15.3 Bandwidth reservation scenarios

How a bandwidth reservation is applied to a policy varies depending on two factors, the global bandwidth rule and the number of policies running at once. These two factors lead to two possible scenarios.

Under the first scenario, more bandwidth is available than all the running policies. In this case, the available bandwidth is split evenly across all running policies, the same as the pre-OneFS 8.2 behavior.

In the second scenario, the global configured bandwidth is less than the sum of the per policy configured bandwidth for the running policies. Therefore, SyncIQ is unable to provide all the policies the requested bandwidth. Under this scenario, an even split occurs of bandwidth across all running policies, until the requested reservation is met. The even split ensures the policies that have the lowest reservation meet their reservation before the policies that have larger reservations, preventing starvation across the policies.
15.3.1 Bandwidth reservation example 1: insufficient bandwidth

In this example, the total requested bandwidth of running policies is more than the global bandwidth reservation. For example, with a global bandwidth rule of 30 Mb and 3 policies running at the same time, consider the following:

- Policy 1 has a bandwidth reservation of 20 Mb
- Policy 2 has a bandwidth reservation of 40 Mb
- Policy 3 has a bandwidth reservation of 60 Mb

In this scenario, enough bandwidth is not available for each policy to meet its reservation. Therefore, each policy is allocated 10 Mb, as illustrated in Figure 36.

![Bandwidth Reservation vs Allocated](image)

Figure 36 Insufficient bandwidth example 1
15.3.2 Bandwidth reservation example 2: insufficient bandwidth

In this example, the total requested bandwidth of running policies is more than the global bandwidth reservation. However, ample bandwidth is available for some of the policies to meet their reservation. For example, with a global bandwidth rule of 80 Mb and 3 policies running at the same time, consider the following:

- Policy 1 has a bandwidth reservation of 20 Mb
- Policy 2 has a bandwidth reservation of 40 Mb
- Policy 3 has a bandwidth reservation of 60 Mb

In this scenario, enough bandwidth is not available for each policy to meet its reservation, but enough is available for Policy 1. Therefore, Policy 1 is allocated its full reservation of 20 Mb, but Policy 2 and 3 are allocated a split of the remaining bandwidth of 30 Mb each, as illustrated in Figure 37.

![Bandwidth Reservation vs Allocated](image)

Figure 37 Insufficient bandwidth example 2
15.3.3 Bandwidth reservation example 3: extra bandwidth available

In this example, the total requested bandwidth of running policies is less than the global bandwidth reservation, allowing additional bandwidth to be granted to policies. For instance, with a global bandwidth rule of 80 Mb and 3 policies running at the same time, consider the following:

- Policy 1 has a bandwidth reservation of 10 Mb
- Policy 2 has a bandwidth reservation of 20 Mb
- Policy 3 has a bandwidth reservation of 30 Mb

In this scenario, enough bandwidth is available for each policy to meet its reservation, but additional bandwidth is available that is not granted. Therefore, Policy 3 is allocated its full reservation of 30 Mb, but Policy 2 and 3 are allocated 25 Mb each, as additional bandwidth is available, as illustrated in Figure 38.

![Bandwidth Reservation vs Allocated](image)

Figure 38 Extra bandwidth example 3
16 Monitoring, alerting, reporting, and optimizing performance

SyncIQ allows administrators to monitor the status of policies and replication jobs with real-time performance indicators and resource utilization. Administrators can determine how different policy settings affect job execution and impact performance on the cluster. In addition, every job execution produces a comprehensive report that can be reviewed for troubleshooting and performance analysis. The real-time reports provide information about the amount of data replicated and the effectiveness of those jobs, enabling resources to be tuned accordingly. For more information about SyncIQ tuning, refer to Section 8, SyncIQ performance rules.

In addition to including cluster-wide performance monitoring tools, such as the isi statistics command or the PowerScale InsightIQ software module, SyncIQ includes module-specific performance monitoring tools. For information about isi statistics and InsightIQ, refer to the PowerScale OneFS 8.2.1 CLI Administration Guide and the PowerScale InsightIQ 4.1 User Guide.

16.1 Policy job monitoring

For high-level job monitoring, use the SyncIQ Summary page where job duration and total dataset statistics are available. The Summary page includes currently running jobs, as well as reports on completed jobs. For more information about a particular job, click the “View Details” link to review job-specific datasets and performance statistics. Use the Reports page to select a specific policy that was run within a specific period and completed with a specific job status.

![SyncIQ Job report details](image)

Figure 39  SyncIQ Job report details

In addition to the Summary and Reports pages, the Alerts page displays SyncIQ specific alerts extracted from the general-purpose cluster Alerts system.
16.2 Performance monitoring

For performance tuning purposes, use the WebUI Cluster Overview performance reporting pages, providing network and CPU utilization rates via real-time or historical graphs. The graphs display both cluster-wide performance and per-node performance. These limits are cluster-wide and are shared across simultaneous running jobs.

Comprehensive resource utilization cluster statistics are available using PowerScale’s InsightIQ multi-cluster reporting and trending analytics suite.

16.3 Alerts

In addition to the dashboard of alerts presented above, errors are also reported in the following log:

/var/log/isi_migrate.log

For information about RPO alerts, refer to Section 6.3.2.1, RPO alerts.

16.4 Reporting

As SyncIQ jobs are running, report data is written at phase changes and checkpoints. The report files are located at the following location:

/ifs/.ifsvar/modules/tsm/sched/reports/<syncpolicyid>/report[-timestamp].gc
16.5 Optimizing SyncIQ performance

The recommended approach for measuring and optimizing performance is as follows:

- Establish reference network performance using common tools such as Secure Copy (SCP) or NFS copy from cluster to cluster. This provides a baseline for a single thread data transfer over the existing network.
- After creating a policy and before running the policy for the first time, use the policy assessment option to see how long it takes to scan the source cluster dataset with default settings.
- Use file rate throttling to roughly control how much CPU and disk I/O SyncIQ consumes while jobs are running through the day.
- Remember that “target aware synchronizations” are much more CPU-intensive than regular baseline replication but they potentially yield much less network traffic if both source and cluster datasets are already seeded with similar data.
- Use IP address pools to control which nodes participate in a replication job and to avoid contention with other workflows accessing the cluster through those nodes.
- Use network throttling to control how much network bandwidth SyncIQ can consume through the day.

16.5.1 Workers and performance scalability

For releases prior to OneFS 8.0, the number of primary and secondary workers is calculated between both clusters based on two factors. First, the lowest number of nodes between the two clusters is considered. The lowest number of nodes is then multiplied by the number of workers per node, which is a configurable value. The default value for workers per node is three. SyncIQ randomly distributes workers across the cluster with each node having at least one worker. If the number of workers is less than the number of nodes, then all nodes will not participate in the replication. An example calculation is illustrated in Figure 41.

![Figure 41: Calculating primary and secondary workers for release prior to OneFS 8.0](image-url)
In OneFS 8.0, the limits have increased to provide additional scalability and capability in line with cluster sizes and higher performing nodes that are available. The maximum number of workers and the maximum number of workers per policy both scale as the number of nodes in the cluster increases. The defaults should be changed only with the guidance of PowerScale Technical Support.

- A maximum of 1,000 configured policies and 50 concurrent jobs are now available.
- Maximum workers per cluster is determined by the total number of virtual cores in the node’s CPUs. The default is 4 * [total virtual cores in the cluster]
- Maximum workers per policy is determined by the total number of nodes in the cluster. The default is 8 * [total nodes in the cluster]
- Instead of a static number of workers as in previous releases, workers are dynamically allocated to policies, based on the size of the cluster and the number of running policies. Workers from the pool are assigned to a policy when it starts, and the number of workers on a policy will change over time as individual policies start and stop. The goal is that each running policy always has an equal number (+/- 1) of the available workers assigned.
- Maximum number of target workers remains unchanged at 100 per node

**Note:** The source and target cluster must have the same number of workers, as each set of source and target workers create a TCP session. Any inconsistency in the number of workers results in failed sessions. As stated above, the maximum number of target workers is 100 per node, implying the total number of source workers is also 100 per node.

**Note:** The following example is provided for understanding how a node’s CPU type impact worker count, how workers are distributed across policies, and how SyncIQ works on a higher level. The actual number of workers is calculated dynamically by OneFS based on the node type. The calculations in the example are not a tuning recommendation and are merely for illustration. If the worker counts require adjustment, contact PowerScale Technical Support, as the number of virtual cores, nodes, and other factors are considered prior to making changes.

As an example, consider a 4-node cluster, with 4 cores per node. Therefore, there are 16 total cores in the cluster. Following the previous rules:

- Maximum workers on the cluster = 4 * 16 = 64 workers
- Maximum workers per policy = 8 * 4 = 32

When the first policy starts, it will be assigned 32 workers (out of the maximum 64). A second policy starting will also be assigned 32 workers. The maximum number of workers per policy has been determined previously as 32, and there are now a total of 64 workers – the maximum for this cluster. When a third policy starts, assuming the first two policies are still running, the maximum of 64 workers are redistributed evenly, so that 21 workers are assigned to the third policy, and the first two policies have their number of workers reduced from 32 to 21 and 22 respectively, as 64 does not split into 3 evenly. Therefore, there are 3 policies running, each with 21 or 22 workers, keeping the cluster maximum number of workers at 64. Similarly, a fourth policy starting would result in all four policies having 16 workers. When one of the policies completes, the reallocation again ensures the workers are distributed evenly amongst the remaining running policies.

**Note:** Any reallocation of workers on a policy occurs gradually to reduce thrashing when policies are starting and stopping frequently.
16.5.2 Specifying a maximum number of concurrent SyncIQ jobs
Administrators may want to specify a limit for the number of concurrent SyncIQ jobs running. Limiting the number is particularly useful during peak cluster usage and client activity. Forcing a limit on cluster resources for SyncIQ ensures that clients do not experience any performance degradation.

**Note:** Consider all factors prior to limiting the number of concurrent SyncIQ jobs, as policies may take more time to complete, impacting RPO and RTO times. As with any significant cluster update, testing in a lab environment is recommended prior to a production cluster update. Additionally, a production cluster should be updated gradually, minimizing impact and allowing measurements of the impacts.

To limit the maximum number of concurrent SyncIQ jobs, perform the following steps from the OneFS CLI:

1. **Modify** `/ifs/.ifsvar/modules/tsm/config/siq-conf.gc` using a text editor.
2. Change the following line to represent the maximum number of concurrent jobs for the cluster:
   ```plaintext```
   `scheduler.max_concurrent_jobs`
   ```plaintext```
3. **Restart** SyncIQ services by executing the following command:
   ```shell```
   `isi sync settings modify --service off;sleep5; isi sync settings modify --service on`
   ```shell```

16.5.3 Performance tuning for OneFS 8.X releases
OneFS 8.0 introduced an updated SyncIQ algorithm taking advantage of all available cluster resources, improving overall job run times significantly. SyncIQ is exceptionally efficient in network data scaling and utilizes 2 MB TCP windows, considering WAN latency while delivering maximum performance.

**Note:** The steps and processes mentioned in this section may significantly impact RPO times and client workflow. Prior to updating a production cluster, test all updates in a lab environment that mimics the production environment. Only after successful lab trials, should the production cluster be considered for an update. As a best practice, gradually implement changes and closely monitor the production cluster after any significant updates.

SyncIQ achieves maximum performance by utilizing all available cluster resources. If available, SyncIQ consumes the following:

- All available CPU bandwidth
- Worker global pool – Default compute is based on node count and total cluster size. As explained in the previous section
- All available Bandwidth

As SyncIQ consumes cluster resources, this may impact current workflows depending on the environment and available resources. If data replication is impacting other workflows, consider tuning SyncIQ as a baseline by updating the following:

- Limit CPU to 33% per node
- Limit workers to 33% of global – Factoring in lower performance nodes
- Configure bandwidth rules – For example, limit to 10 GB during business hours and 20 GB during off-hours

For information about updating the variables above, refer to Section 8, SyncIQ performance rules. Once the baseline is configured, gradually increase each parameter and collect measurements, ensuring workflows are not impacted. Additionally, consider modifying the maximum number of SyncIQ jobs, as explained in section 16.5.2, Specifying a maximum number of concurrent SyncIQ jobs.
**Note:** The baseline variables provided above are only for guidance and not a one size fits all metric. Every environment varies. Carefully consider cluster resources and workflow, while finding the intersection of workflow impacts with SyncIQ performance.
17 Administration

SyncIQ utilizes options provided by OneFS for access control and using the Platform API. For more information about the PowerScale SDK, reference the following community page: https://community.emc.com/docs/DOC-48273

17.1 Role-based access control

Role-based access control (RBAC) divides up the powers of the “root” and “administrator” users into more granular privileges and allows assignment of these to specific roles. For example, data protection administrators can be assigned full access to SyncIQ configuration and control, but only read-only access to other cluster functionality. SyncIQ administrative access is assigned via the ISL_PRIV_SYNCIQ privilege. RBAC is fully integrated with the SyncIQ CLI, WebUI and Platform API.

17.2 OneFS platform API

The OneFS Platform API provides a RESTful programmatic interface to SyncIQ, allowing automated control of cluster replication. The Platform API is integrated with RBAC, as described above, providing a granular authentication framework for secure, remote SyncIQ administration via scripting languages.
18 SyncIQ replication and data reduction

Data reduction includes deduplication and compression. This section covers how SyncIQ interacts with each of PowerScale’s data reduction processes.

18.1 SmartDedupe

When deduplicated files are replicated to another PowerScale cluster via SyncIQ, the deduplicated files are rehydrated back to their original size, since they no longer share blocks on the target PowerScale cluster. Once replication is complete, SmartDedupe can run on the target cluster, providing the same space efficiency benefits as the source cluster.

Shadow stores are not transferred to target clusters or backup devices. Hence, deduplicated files do not consume less space than non-deduplicated files when they are replicated or backed up. To avoid running out of space on target clusters or tape devices, it is essential to verify that the total amount of storage space saved, and storage space consumed does not exceed the available space on the target cluster or tape device. To reduce the amount of storage space consumed on a target PowerScale cluster, configure deduplication for the target directories of the replication policies. Although this deduplicates data on the target directory, it does not allow SyncIQ to transfer shadow stores. Deduplication is still performed post-replication, via a deduplication job running on the target cluster.

18.2 Isilon F810, H5600 and PowerScale nodes

The Isilon F810 and H5600 platforms provide in-line compression and deduplication. OneFS 9.0 introduces the PowerScale nodes with in-line compression and deduplication.

For source clusters that contain F810, H5600, or PowerScale nodes, during SyncIQ replication, the source data is rehydrated, decompressed and transferred uncompressed to the target cluster. If the target cluster consists of F810, H5600, or PowerScale nodes, the replication data goes through the same in-line compression and deduplication as any other data that is written to these platforms.
16 TiB large file support and SyncIQ implications

OneFS 8.2.2 introduces a feature for large file support, permitting the maximum allowable file size in a PowerScale cluster to increase four-fold from 4 tebibytes previously, to 16 tebibytes. For more information about the large file support feature for 16 TiB, refer to the Dell EMC PowerScale OneFS Best Practices whitepaper. It is critical to note the implications of this feature, as once it is enabled, it cannot be disabled.

If large file support is enabled for 16 TiB on a source cluster, all SyncIQ policies only connect with target clusters that also have large file support enabled, as illustrated in Figure 42. Otherwise, SyncIQ policies fail when establishing a connection with a cluster without large file support. However, a source cluster without large file support enabled, with a maximum file size of 4 TiB, successfully establishes SyncIQ policies to a target cluster with large file support enabled, with a maximum file size of 16 TiB, as illustrated in Figure 42.

These unique requirements apply to new and existing SyncIQ policies. Furthermore, workflow impacts for existing SyncIQ policies are possible, if for any reason, the target cluster does not have resources for the 16 TiB feature.

Figure 42  16 TiB large file support and SyncIQ implications
OneFS version compatibility

It is recommended to have the same OneFS version and patches on both the source and target cluster. However, this is not always possible in some environments due to varying factors.

Note: As a best practice, upgrade the target cluster before upgrading the source cluster to ensure no interruptions to replication jobs occur as part of the upgrade process.

OneFS 8.2.2 introduces large file support for 16 TiB files. Once enabled, a source or target cluster only establishes connections with clusters that have the 16 TiB feature enabled. For more information about the 16 TiB feature, refer to Section 19, 16 TiB large file support and SyncIQ.

If the source and target cluster are running different versions of OneFS, to confirm SyncIQ compatibility refer to Table 2.

Table 2  SyncIQ OneFS version compatibility

<table>
<thead>
<tr>
<th>Source Cluster OneFS Version</th>
<th>Target Cluster OneFS Version</th>
<th>7.2.x</th>
<th>8.x and 9.0</th>
<th>8.2.2 and 9.0 with 16 TiB feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2.x</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>8.x and 9.0</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>8.2.2 and 9.0 with 16 TiB feature</td>
<td></td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
</tbody>
</table>
SmartLock compatibility

Data replication is a crucial requirement for many WORM protected data sets. OneFS provides WORM functionality through SmartLock, which is compatible with SyncIQ for data replication.

For SyncIQ and SmartLock environments it is essential to ensure all node clocks are synchronized. Therefore, it is recommended to have all nodes on the source and target clusters configured with Network Time Protocol (NTP) Peer Mode. If Compliance SmartLock is required, all source and target nodes must be configured in NTP Peer Mode prior to configuring the compliance clock.

Replicating data from a source SmartLock directory to a target SmartLock directory, ensures all metadata related to the retention date and commit status persists on the target. On the contrary, replicating from a SmartLock directory to a non-SmartLock directory causes all metadata relating to the retention date and commit status to be lost.

It is recommended to have the source and target directory in the same compliance mode. In many environments, it may not be possible to have the source and target directories in the same compliance mode. Depending on the source and target directory types, SyncIQ may be compatible. However, to confirm if the source and target directories are compatible with SmartLock, refer to Table 3.

<table>
<thead>
<tr>
<th>Source directory type</th>
<th>Target directory type</th>
<th>SyncIQ source-to-target compatibility</th>
<th>Failback allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-SmartLock</td>
<td>Non-SmartLock</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Non-SmartLock</td>
<td>Enterprise SmartLock</td>
<td>Yes</td>
<td>Yes, unless files are committed to a WORM state on the target cluster.</td>
</tr>
<tr>
<td>Non-SmartLock</td>
<td>Compliance SmartLock</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Enterprise SmartLock</td>
<td>Non-SmartLock</td>
<td>Yes, replication type is allowed. However, retention will not be enforced.</td>
<td>Yes, however, files will not have WORM status.</td>
</tr>
<tr>
<td>Enterprise SmartLock</td>
<td>Enterprise SmartLock</td>
<td>Yes</td>
<td>Yes, any newly committed WORM files will be included.</td>
</tr>
<tr>
<td>Enterprise SmartLock</td>
<td>Compliance SmartLock</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Compliance SmartLock</td>
<td>Non-WORM</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Compliance SmartLock</td>
<td>Enterprise SmartLock</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Compliance SmartLock</td>
<td>Compliance SmartLock</td>
<td>Yes</td>
<td>Yes, any newly committed WORM files will be included.</td>
</tr>
</tbody>
</table>
21.1 **Compliance mode**

Replicating data with SyncIQ from a source cluster configured for SmartLock compliance directories to a target cluster is only supported if the target cluster is running in SmartLock compliance mode. The source and target directories of the replication policy must be root paths of SmartLock compliance directories on the source and target cluster. Replicating data from a compliance directory to a non-compliance directory is not supported, causing the replication job to fail.

21.2 **Failover and failback with SmartLock**

OneFS 8.0 introduced support for failover and failback functions of ‘Enterprise Mode’ directories. OneFS 8.0.1 introduced support for failover and failback of ‘Compliance Mode’ directories, delivering automated disaster recovery for the financial services SEC-17a4 regulatory compliance. Refer to Table 3, to confirm if failback is supported, depending on the source and target directory types.

21.3 **SmartLock and SyncIQ security**

As a best practice, securing a SmartLock cluster is recommended using either SyncIQ encryption or a pre-shared key. Configuring encryption is preferred. However, for environments where this isn't possible, the pre-shared key (PSK) is recommended.

SmartLock Compliance mode clusters do not support SyncIQ PSK. For clusters in SmartLock Compliance mode, upgrading to OneFS 8.2 or later is recommended and configuring SyncIQ encryption. SmartLock Enterprise mode clusters support SyncIQ PSK.

For more information about configuring SyncIQ encryption or a PSK, refer to Section 14, SyncIQ security.
Conclusion

SyncIQ implements scale-out asynchronous replication for PowerScale clusters, providing scalable replication performance, easy failover and failback, and dramatically improving recovery objectives. SyncIQ design, combined with tight integration with OneFS, native storage tiering, point-in-time snapshots, retention, and leading backup solutions, makes SyncIQ a powerful, flexible, and easy-to-manage solution for disaster recovery, business continuity, disk-to-disk backup, and remote archive.
Failover and failback steps

This section provides detailed steps to complete a SyncIQ Failover and Failback.

**Note:** Ensure the steps provided in this section are followed in **sequential order in its entirety**. If the steps in this section are not followed sequentially in its entirety data could be lost and become unrecoverable.

### A.1 Assumptions

- In order to failover to an associated cluster, a SyncIQ policy must exist between the source and target cluster, as explained in Section 6, Configuring a SyncIQ policy.
- If the policy is configured, it must have successfully run at least once.
- All system configuration is complete on the target cluster, emulating the source cluster’s configuration. This includes licensing, access zones, SmartConnect, shares, authentication providers, etc.
- This section does not consider any network or other environmental changes required. During disaster readiness testing, ensure all other environmental steps are documented and shared with administrators.

**Note:** As a best practice, configure DNS to require a single forwarding change only. During an outage, this minimizes downtime and simplifies the failover process.

### A.2 Failover

Once a policy is configured and run successfully, a failover may be initiated with the following steps:

1. **If the source cluster is online**, stop all writes impacting the directory path of the replication policy, limiting any new data from being written on the cluster. In large environments it may be difficult to stop all clients from writing data, it may be easier to stop SMB, NFS and FTP services on the source cluster.

   To stop services on the source cluster, execute the following commands:

   ```
   Source-cluster# isi services smb disable
   Source-cluster# isi services nfs disable
   Source-cluster# isi services vsftpd disable
   ```

2. **If the source cluster is online**, ensure any scheduled policies on the source cluster do not replicate data during the failover. Place the policies in manual mode by executing the following command:

   ```
   Source-cluster# isi sync policies modify [Policy Name] -schedule ""
   ```

3. **If the source cluster is online**, run the associated policy manually by executing the following command:

   ```
   Source-cluster# isi sync policies jobs start [Policy Name]
   ```

   **Ensure** the policy completes prior to proceeding to the next step.
4. On the target cluster, from the web interface click **Data Protection > SyncIQ > Local Targets**. From the “Local Targets” tab, scroll to the appropriate policy and select **More > Allow Writes**. To perform this from the CLI, execute the following command:

```
Target-cluster# isi sync recovery allow-write --policy-name=[Policy Name]
```

At this point, the target cluster is now accessible and writable.

Clients must now be **re-directed** to the target cluster to continue accessing the file system. Make any necessary network, DNS, and environmental updates. Depending on the DNS configuration, a single DNS update only changing the forwarding is sufficient.

### A.3 Failback

Once the failover is complete, and the source cluster is operational, the failback process may commence with the following steps:

1. On the source cluster, click **Data Protection > SyncIQ > Policies**. In the **SyncIQ Policies** list, for the associated replication policy, click **More > Resync-prep**. Alternatively, from the source cluster CLI, execute the following command:

```
Source-cluster# isi sync recovery resync-prep [Policy Name]
```

To check the current status of the resync-prep, with duration, transfer, and throughput, execute the following command:

```
Source-cluster# isi sync jobs reports list
```

This action causes SyncIQ to create a mirror policy for the replication policy on the target cluster. The mirror policy is placed under **Data Protection > SyncIQ > Local Targets** on the target cluster.

SyncIQ names mirror policies according to the following pattern:

```
<replication-policy-name>_mirror
```

2. Before beginning the failback process, prevent clients from accessing the target cluster. In large environments it may be difficult to stop all clients from writing data, it may be easier to stop SMB, NFS and FTP services on the source cluster.

To stop services on the target cluster, execute the following commands:

```
Target-cluster# isi services smb disable
Target-cluster# isi services nfs disable
Target-cluster# isi services vsftpd disable
```

3. On the target cluster, click **Data Protection > SyncIQ > Policies**. In the **SyncIQ Policies** list, for the mirror policy, click **More > Start Job**. Alternatively, to start the mirror policy from the CLI, execute the following command:

```
Target-cluster# isi sync jobs start -policy-name=[Mirror Policy Name]
```

If required, the mirror policy on the target cluster may be modified to specify a schedule for the policy to run.
Failover and failback steps

**Note:** Prior to proceeding to the next step, ensure the mirror policy completes successfully, otherwise data may be lost and unrecoverable.

4. On the source cluster, click **Data Protection > SyncIQ > Local Targets**. In the **SyncIQ Local Targets** list, for the mirror policy, select **More > Allow Writes**. Alternatively, to perform the allow writes from the CLI, execute the following command:

   ```bash
   Source-cluster# isi sync recovery allow-write --policy-name=[Policy Name]
   ```

5. On the target cluster, click **Data Protection > SyncIQ > Policies**. For the appropriate mirror policy in the **SyncIQ Policies** list, click **More > Resync-prep**. Alternatively, to perform the resync-prep from the CLI, execute the following command:

   ```bash
   Target-cluster# isi sync recovery resync-prep [Policy Name]
   ```

   This places the target cluster back into read-only mode and ensures that the data sets are consistent on both the source and target clusters.

**A.3.1 Finalizing the failback**

Redirect clients to begin accessing their data on the source cluster. Although not required, it is safe to remove a mirror policy after failback has completed successfully.
SyncIQ encryption was introduced in OneFS 8.2. For more information about SyncIQ encryption, refer to Section 14.1, SyncIQ encryption. Configuring SyncIQ encryption with self-signed certificates is only suggested for specific environments that may not have access to a Certificate Authority. Utilizing a Certificate Authority for configuring SyncIQ encryption is the best practice, as explained in Section 14.1.1, Configuring SyncIQ encryption.

To configure SyncIQ encryption utilizing self-signed certificates, perform the following:

**B.1 Generate keys**

1. **On the source cluster, generate keys:**
   a. mkdir /ifs/data/[Specify a directory]
   b. cd /ifs/data/[New directory from Step a]
   c. openssl req -newkey rsa:2048 -keyout source_cluster_key.key -x509 -days [Number of days the certificate is valid for] -out source_cluster_cert.pem

   Record and take note of the passphrase created at this step.

2. **On the target cluster, generate keys:**
   a. mkdir /ifs/data/[Specify a directory]
   b. cd /ifs/data/[New directory from Step a]
   c. openssl req -newkey rsa:2048 -keyout target_cluster_key.key -x509 -days [Number of days the certificate is valid for] -out target_cluster_cert.pem

   Record and take note of the passphrase created at this step.

**B.2 Import keys and apply SyncIQ settings**

1. **On the source cluster, import the target cluster’s certificate:**
   a. scp root@[target cluster IP]:/ifs/data/[Directory specified in Section B.1]/target_cluster_cert.pem /ifs/data/[Directory specified in Section B.1]/

2. For the source and target cluster to successfully SSL handshake, add the target cluster’s self-signed certificate to the certificate authority list on the source cluster, ensuring the source cluster trusts the target cluster’s signature. On the source cluster:
   a. isi cert auth import ./target_cluster_cert.pem --name TargetCluster_Self-Signed

3. **On the source cluster, define the target cluster as a SyncIQ peer:**
SyncIQ encryption with self-signed certificates

a. isi sync cert peer import ./target_cluster_cert.pem --name=[Specify a descriptive certificate name]

4. On the source cluster, provide SyncIQ with the source cluster server certificate and private key:
   a. isi sync cert server import ./source_cluster_cert.pem ./source_cluster_key.key --name=[Specify a name for the source server certificate] --certificate-key-password [Passphrase for the private key created in Section B.1, Step 1.c]

5. To apply the cluster certificate with SyncIQ, the full certificate ID is required. On the source cluster, retrieve the truncated certificate ID for the server certificate:
   a. isi sync cert server list
      Make a note of the appropriate truncated certificate ID, from the 'ID' column. On the source cluster, retrieve the full certificate ID, using the truncated certificate ID from Step a:
   b. isi sync cert server view [truncated certificate ID from Step a]
      Make a note of the full certificate ID, from the 'ID' field.

6. On the source cluster, apply the full certificate ID from the previous step as the cluster certificate:
   a. isi sync settings modify --cluster-certificate-id=[full certificate ID from the previous step]

7. A global option is available, requiring that all incoming and outgoing SyncIQ policies are encrypted.

    Note: Executing this command impacts existing SyncIQ policies that may not have encryption enabled. Only execute this command once all existing policies have encryption enabled. Otherwise, existing policies that do not have encryption enabled will fail.

    On the source cluster, require encryption globally for all SyncIQ policies:
    a. isi sync settings modify --encryption-required=true

8. On the target cluster, import the source cluster's certificate:
   a. scp root@[source cluster IP]:/ifs/data/[Directory specified in Section B.1]/source_cluster_cert.pem /ifs/data/[Directory specified in Section B.1]/

9. For the target and source cluster to successfully SSL handshake, add the source cluster's self-signed certificate to the certificate authority list on the target cluster, ensuring the target cluster trusts the source cluster's signature. On the target cluster:
   a. isi cert auth import ./source_cluster_cert.pem --name SourceCluster_Self-Signed

10. On the target cluster, define the source cluster as a SyncIQ peer:
SyncIQ encryption with self-signed certificates

**B.3 Create an encrypted SyncIQ policy**

1. On the source cluster, find the truncated certificate ID of the target cluster, also known as the SyncIQ peer:
   a. `isi sync certificates peer list`

   Make a note of the appropriate truncated certificate ID, from the ‘ID’ column. On the source cluster, retrieve the full certificate ID, using the truncated certificate ID from Step a:
   b. `isi sync certificates peer view [truncated certificate ID from Step a]`

   Make a note of the full certificate ID, from the ‘ID’ field.

2. On the source cluster, create an encrypted SyncIQ policy, using the following command:
SyncIQ encryption with self-signed certificates

a. isi sync pol create [SyncIQ Policy Name] sync [Source Cluster Directory] [Target Cluster IP Address] [Target Cluster Directory] --target-certificate-id=[full certificate ID from the previous step]

B.4 Modify an existing SyncIQ policy for encryption

1. On the source cluster, find the truncated certificate ID of the target cluster, also known as the SyncIQ peer:
   a. isi sync certificates peer list

   Make a note of the appropriate truncated certificate ID, from the ‘ID’ column. On the source cluster, retrieve the full certificate ID, using the truncated certificate ID from Step a:

   b. isi sync certificates peer view [truncated certificate ID from Step a]

   Make a note of the full certificate ID, from the ‘ID’ field.

2. To modify an existing SyncIQ policy for encryption, on the source cluster, use the following command:

   isi sync policies modify <pol_name> --target-certificate-id=<full certificate ID from the previous step>

B.5 Additional SyncIQ information and optional commands

For more information about SyncIQ encryption and optional commands, refer to Section 14.1, SyncIQ encryption.
Configuring cascaded replication

The steps provided in this appendix provide an example of configuring cascaded replication. For this example, the steps configure the implementation in Figure 43.

1. On ‘Cluster A’, configure replication from ‘Cluster A’ to ‘Cluster B’ based on when a snapshot of the source directory is taken:

   ```bash
   IsilonClusterA# isi sync policies create --name=pol_a_b sync --source-root-path=/ifs/data/pol_a_b --target-host=[Cluster B IP Address] --target-path=/ifs/data/pol_a_b --schedule=when-snapshot-taken
   ```

   Modify policy ‘pol_a_b’ to specify a snapshot naming pattern and archive snapshots:

   ```bash
   IsilonClusterA# isi sync policies modify pol_a_b --target-snapshot-pattern=SIQ_%{SrcCluster}-%{PolicyName}-%Y-%m-%d-%H-%M-%S --target-snapshot-archive=true
   ```

   Confirm policy ‘pol_a_b’ is configured for when a snapshot is taken, the snapshot is archived, and the naming pattern is specified:

   ```bash
   IsilonClusterA# isi sync policies view pol_a_b
   <Output truncated - Confirm the fields listed below>
   .
   .
   Target Snapshot Archive: Yes
   Target Snapshot Pattern: SIQ_%{SrcCluster}-%{PolicyName}-%Y-%m-%d-%H-%M-%S
   Target Snapshot Expiration: Never
   Schedule: when-snapshot-taken
   ```

2. On ‘Cluster B’, configure replication from ‘Cluster B’ to ‘Cluster C’ based on when a snapshot of the source directory is taken:

   ```bash
   IsilonClusterB# isi sync policies create --name=pol_b_c sync --source-root-path=/ifs/data/pol_a_b --target-host=[Cluster C IP Address] --target-path=/ifs/data/pol_a_b --schedule=when-snapshot-taken
   ```

   Modify policy ‘pol_b_c’ to specify a snapshot naming pattern and archive snapshots:
IsilonClusterB# isi sync policies modify pol_b_c --target-snapshot-pattern=SIQ_%{SrcCluster}-%{PolicyName}-%Y-%m-%d-%H-%M-%S --target-snapshot-archive=true

Confirm policy 'pol_b_c' is configured for when a snapshot is taken, the snapshot is archived, and the naming pattern is specified:

IsilonClusterB# isi sync policies view pol_b_c
<Output truncated - Confirm the fields listed below>

Target Snapshot Archive: Yes
Target Snapshot Pattern: SIQ_%{SrcCluster}-%{PolicyName}-%Y-%m-%d-%H-%M-%S
Target Snapshot Expiration: Never
Schedule: when-snapshot-taken

3. On 'Cluster C', configure replication from 'Cluster C' to 'Cluster D' based on when a snapshot of the source directory is taken:

IsilonClusterC# isi sync policies create --name=pol_c_d sync --source-root-path=/ifs/data/pol_a_b --target-host=[Cluster D IP Address] --target-path=/ifs/data/pol_a_b --schedule=when-snapshot-taken

Modify policy 'pol_c_d' to specify a snapshot naming pattern and archive snapshots:

IsilonClusterC# isi sync policies modify pol_c_d --target-snapshot-pattern=SIQ_%{SrcCluster}-%{PolicyName}-%Y-%m-%d-%H-%M-%S --target-snapshot-archive=true

Confirm policy ‘pol_c_d’ is configured for when a snapshot is taken, the snapshot is archived, and the naming pattern is specified:

IsilonClusterC# isi sync policies view pol_c_d
<Output truncated - Confirm the fields listed below>

Target Snapshot Archive: Yes
Target Snapshot Pattern: SIQ_%{SrcCluster}-%{PolicyName}-%Y-%m-%d-%H-%M-%S
Target Snapshot Expiration: Never
Schedule: when-snapshot-taken
D Configuring custom replication

The steps provided in this appendix provide an example of configuring a custom replication, combining the cascaded and one-to-many topologies. For this example, the steps configure the implementation in Figure 44.

1. On ‘Cluster A’, configure replication from ‘Cluster A’ to ‘Cluster B’ based on when a snapshot of the source directory is taken:

   ```
   IsilonClusterA# isi sync policies create --name=pol_a_b sync --source-root-path=/ifs/data/pol_a_b --target-host=[Cluster B IP Address] --target-path=/ifs/data/pol_a_b --schedule=when-snapshot-taken
   ```

   Modify policy ‘pol_a_b’ to specify a snapshot naming pattern and archive snapshots:

   ```
   IsilonClusterA# isi sync policies modify pol_a_b --target-snapshot-pattern=SIQ_%{SrcCluster}-%{PolicyName}-%Y-%m-%d_%H-%M-%S --target-snapshot-archive=true
   ```

   Confirm policy ‘pol_a_b’ is configured for when a snapshot is taken, the snapshot is archived, and the naming pattern is specified:

   ```
   IsilonClusterA# isi sync policies view pol_a_b
   <Output truncated – Confirm the fields listed below>
   .
   .
   .
   Target Snapshot Archive: Yes
   Target Snapshot Pattern: SIQ_%{SrcCluster}-%{PolicyName}-%Y-%m-%d_%H-%M-%S
   Target Snapshot Expiration: Never
   ```

Figure 44 Cascaded and one-to-many replication
2. On ‘Cluster B’, configure replication from ‘Cluster B’ to ‘Cluster C’ based on when a snapshot of the source directory is taken:

   IsilonClusterB# isi sync policies create --name=pol_b_c sync --source=root-path=/ifs/data/pol_a_b --target-host=[Cluster C IP Address] --target-path=/ifs/data/pol_a_b --schedule=when-snapshot-taken

   Modify policy ‘pol_b_c’ to specify a snapshot naming pattern and archive snapshots:

   IsilonClusterB# isi sync policies modify pol_b_c --target-snapshot-pattern=SIQ_%{SrcCluster}-%{PolicyName}-%Y-%m-%d-%H-%M-%S --target-snapshot-archive=true

   Confirm policy ‘pol_b_c’ is configured for when a snapshot is taken, the snapshot is archived, and the naming pattern is specified:

   IsilonClusterB# isi sync policies view pol_b_c
   <Output truncated - Confirm the fields listed below>
   .
   .
   .
   Target Snapshot Archive: Yes
   Target Snapshot Pattern: SIQ_%{SrcCluster}-%{PolicyName}-%Y-%m-%d-%H-%M-%S
   Target Snapshot Expiration: Never
   Schedule: when-snapshot-taken

3. On ‘Cluster B’, configure replication from ‘Cluster B’ to ‘Cluster D’ based on when a snapshot of the source directory is taken:

   IsilonClusterB# isi sync policies create --name=pol_b_d sync --source=root-path=/ifs/data/pol_a_b --target-host=[Cluster D IP Address] --target-path=/ifs/data/pol_a_b --schedule=when-snapshot-taken

   Modify policy ‘pol_b_d’ to specify a snapshot naming pattern and archive snapshots:

   IsilonClusterB# isi sync policies modify pol_b_d --target-snapshot-pattern=SIQ_%{SrcCluster}-%{PolicyName}-%Y-%m-%d-%H-%M-%S --target-snapshot-archive=true

   Confirm policy ‘pol_b_d’ is configured for when a snapshot is taken, the snapshot is archived, and the naming pattern is specified:

   IsilonClusterB# isi sync policies view pol_b_d
   <Output truncated - Confirm the fields listed below>
   .
   .
   .
   Target Snapshot Archive: Yes
   Target Snapshot Pattern: SIQ_%{SrcCluster}-%{PolicyName}-%Y-%m-%d-%H-%M-%S
   Target Snapshot Expiration: Never
   Schedule: when-snapshot-taken

4. On ‘Cluster B’, configure replication from ‘Cluster B’ to ‘Cluster E’ based on when a snapshot of the source directory is taken:
Configuring custom replication

IsilonClusterB# isi sync policies create --name=pol_b_e sync --source-root-path=/ifs/data/pol_a_b --target-host=[Cluster E IP Address] --target-path=/ifs/data/pol_a_b --schedule=when-snapshot-taken

Modify policy ‘pol_b_e’ to specify a snapshot naming pattern and archive snapshots:

IsilonClusterB# isi sync policies modify pol_b_e --target-snapshot-pattern=SIQ_%%{SrcCluster}-%{PolicyName}-%Y-%m-%d-%H-%M-%S --target-snapshot-archive=true

Confirm policy ‘pol_b_e’ is configured for when a snapshot is taken, the snapshot is archived, and the naming pattern is specified:

IsilonClusterB# isi sync policies view pol_b_e
<Output truncated – Confirm the fields listed below>
  .
  .
  .
Target Snapshot Archive: Yes
Target Snapshot Pattern: SIQ_%%{SrcCluster}-%{PolicyName}-%Y-%m-%d-%H-%M-%S
Target Snapshot Expiration: Never
Schedule: when-snapshot-taken
Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

Storage technical documents and videos provide expertise that helps to ensure customer success on Dell EMC storage platforms.

E.1 Related resources

OneFS 9.0 Documentation - PowerScale Info Hub
PowerScale Network Design Considerations
Superna Eyeglass
High Availability and Data Protection with Dell EMC PowerScale Scale-Out NAS
PowerScale OneFS 9.0 CLI Administration Guide
OneFS 9.0 Web Administration Guide
PowerScale InsightIQ 4.1 User Guide
OneFS 9.0 Security Configuration Guide
PowerScale OneFS CloudPools Administration Guide
PowerScale CloudPools and ECS Solution Guide
Dell EMC PowerScale OneFS Best Practices