White Paper

SMB FILE MIGRATION TO EMC ISILON

Guidance for optimal data migration of SMB workflows

Abstract

This paper provides technical information and recommendations to help you migrate data from a single SMB protocol workflow on another NAS vendor to an EMC Isilon storage cluster. It includes the best practices for planning, setting up, and executing the migration.

September 2013

EMC²

Part Number H12212

Table of contents

Introduction

This white paper outlines the recommended approach for migrating single protocol SMB data from other NAS storage systems to an EMC® Isilon® storage cluster. The paper includes best practices on Isilon cluster configuration, tool selection, and host setup to optimize a data migration. The paper also includes best practices to optimize performance, management, and support. Although this paper addresses a single SMB protocol data migration, the approach and many of the best practices can be used as a foundation for other types of data migration.

Much of the relevant information for planning, provisioning, and supporting end-user directories on an Isilon storage cluster is available through white papers and guides from EMC Isilon at http://support.emc.com. As such, this guide avoids duplicating content by including only the information that pertains to setting up and operating an Isilon cluster as a destination for an SMB data migration.

Assumptions

This guide assumes that the source data is only accessed through a single protocol SMB workflow. The authentication and authorization for the source data is handled by Microsoft® Active Directory, and users and groups have membership and access in both the source and destination storage systems. No translation or remediation of these security protocols is required during the data migration.

Directories have unique access control list (ACL) permissions that restrict access to the intended user. Additional access is provided for administrative support and backup services as necessary.

This document focuses on the data migration of SMB accessed files; it does not specifically address the migration to an Isilon cluster of SMB shares, local users and groups, or any other SMB configurations from another network-attached storage (NAS) system.

Audience

This guide is intended for experienced system and storage administrators who are familiar with file services and network storage administration.

This guide assumes you have a working knowledge of the following:

- NAS systems
- The SMB storage protocol, as appropriate for the specific migration requirements
- The Isilon scale-out storage architecture and the Isilon OneFS® operating system
- Additional Isilon features, including SmartConnect™, SmartPools® policy management, SnapshotIQ™, and SmartQuotas™
- File-system management concepts and practices, including provisioning, permissions, and performance optimization
- Integration practices for connecting and establishing authentication relationships with Microsoft Active Directory

While this guide is intended to provide a consolidated reference point to migrate data to an Isilon storage cluster, it is not intended to be the authoritative source of information on the technologies and features used to provide and support a file-services platform. In the event that additional services are required, EMC IT Services are available to assist in streamlining data migrations, reducing risk, and minimizing impact.

Prerequisites

Some of the features that are described or recommended in this document may require separate per-node licensing from EMC Isilon. For more information, please contact your EMC Isilon representative.

The challenge of data migration

The migration of a storage system's data and all the existing user access permissions is a complex process. Moving the data while limiting downtime and protecting the data can be challenging. As you execute a migration, the key requirements are being able to access the data at all times and ensuring data integrity against loss or corruption.

It's critical to understand that a data migration must be considered a unique project. Few environments are the same and, as a result, each migration should be considered a unique event. No pre-existing approach will necessarily be appropriate for all migrations. But that's not to say that common approaches cannot be used after you evaluate and understand the requirements of a specific migration. The goal of this white paper is to introduce the recommended approach to designing and executing an SMB data migration to an Isilon cluster.

Understanding the sequence of a migration project is critical to its success. The ability to predict and manage the time required to execute the data movement is paramount—it may be the single biggest factor that will affect the project's success.

Maintaining data availability throughout the lifecycle of the data migration project is also critical. Little of today's data can go for days without being available. In order to maintain data availability, you will need a strategy to maintain access to data throughout the migration.

Risk management

It is not uncommon to have a number of challenges or perceived problems that are seen as blocking issues or barriers to executing the migration. With sufficient planning and testing, the perceived risks can be addressed and managed successfully.

Common risks associated with data migrations include the following:

- Amount of data and the time required to move it
- Performance of the existing data solution and the network during the migration
- Maintaining access to the data throughout the migration
- Changes to the data permission models
- Challenges with moving client connections
- Maintaining a consistent security model after the migration

- Execution of the actual cutover event

This paper helps you understand these perceived risks and develop a data migration methodology to manage the risks, while implementing Isilon best practices to facilitate and optimize the migration.

Data integrity

Data integrity is critical. The data must be moved exactly as it is, and any modification to the data during migration may impact the availability of data and the success of the migration. The goal of the project is to ensure the data is successfully moved and its integrity is not compromised during movement. In most cases this includes the migration of all relevant file metadata as well as the underlying data blocks.

Data availability

Any migration activity will require a transition or cutover from the existing source systems to the new destination systems. This cutover will require a window of time when the data is unavailable. Minimizing this time is the goal of all migrations and is often determined by the type and amount of data. A number of migration strategies can be employed to limit the unavailability of data during this time. In any cutover, data clients will also need to be redirected toward the new storage target to ensure that they can continue to access their data once the data has been moved.

Project phases and methodology overview

A data migration project should be broken into distinct phases. The goal of the project phases is to develop a robust, repeatable migration strategy that aids the execution and leads to a successful migration cutover.

Discovery and planning phase

The goal of the discovery and planning phase is to determine the following and design a migration methodology and plan that will allow you to execute the project:

- Qualify the project
- Identify the migration scope
- Understand expectations
- Identify risks
- Define the timeline
- Identify all migration requirements (e.g., rollback plan)

During this phase, a detailed review of the existing source environment and data is undertaken, after which a plan to migrate the data to the new target Isilon environment can be developed. The planning phase should be completed and validated before you start the other project phases. The key aspects of the planning phase include discovering the infrastructure, the data, and the cluster.

Infrastructure discovery

This is where the infrastructure of the existing storage system, network architecture, and the network path between the source data and the Isilon cluster are evaluated.

Data discovery

This is where you analyze the data and workflows that you plan to migrate, and the way they map to the target end-state on the Isilon cluster.

Isilon cluster configuration design and discovery

This is where activities such as the design of the Isilon network, disk pools, shares, and authentication can affect the migration design. The configuration of the cluster is critical to the execution of the migration.

The output of the discovery phase feeds into the migration design and drives the execution of the project.

Migration approach and requirements

The analysis of the data that you collected during the discovery phase drives the migration requirements and the migration plan. The migration requirements break down into subcategories:

What—What are you migrating?

- Will you migrate all the data or a subset of the data?
- Will you replicate the existing data as is or transform it during migration?
- Will you copy the data but implement a new security model?
- Will you take a hybrid approach?

How—How are you going to migrate the data, security, workflows?

- What tools will you use to copy the data and for security?
- What is your cutover strategy; how will client connections be moved?
- If the data has a rapid rate of change, how will you accommodate it?
- Do you have data that is static, and that can be moved without impact?
- Will you initiate full data copies and follow-up incremental copies to gather recently updated data?
- How do client currently access this data currently (method x/y/z)?
- How will you limit access to the old data and redirect during the cutover?

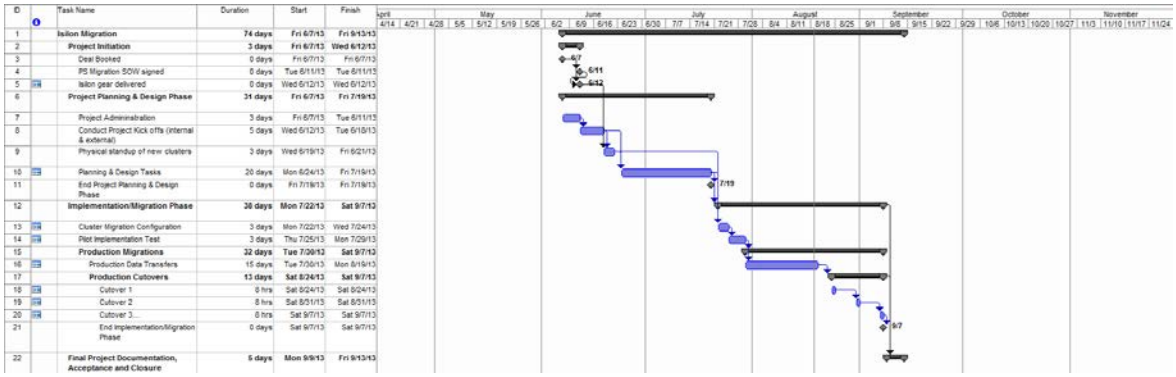
When—When are you implementing the cutover?

- Will it be a single mass event?
- Will you do several large cutover events?
- Will it be a series of smaller cutovers sustained over a larger timeframe?
- Will it be a rolling migration?

Having clearly defined these requirements, a migration methodology can be developed to address them.

Migration methodology

Analysis of the migration requirements leads to the development of a migration methodology. The migration methodology follows a waterfall methodology, with phases in general occurring on completion of the prior phase. Although the preparation for upcoming phases can occur before prior phases are complete, the execution is defined by the completion of its dependent phase.



ID	Task Name	Duration	Start	Finish
1	**Isilon Migration**	**74 days**	**Fri 6/7/13**	**Fri 9/13/13**
2	**Project Initiation**	**3 days**	**Fri 6/7/13**	**Wed 6/12/13**
3	Deal Booked	0 days	Fri 6/7/13	Fri 6/7/13
4	PS Migration SOW signed	0 days	Tue 6/11/13	Tue 6/11/13
5	Isilon gear delivered	0 days	Wed 6/12/13	Wed 6/12/13
6	**Project Planning & Design Phase**	**31 days**	**Fri 6/7/13**	**Fri 7/19/13**
7	Project Administration	3 days	Fri 6/7/13	Tue 6/11/13
8	Conduct Project Kick offs (internal & external)	5 days	Wed 6/12/13	Tue 6/18/13
9	Physical standup of new clusters	3 days	Wed 6/19/13	Fri 6/21/13
10	Planning & Design Tasks	20 days	Mon 6/24/13	Fri 7/19/13
11	End Project Planning & Design Phase	0 days	Fri 7/19/13	Fri 7/19/13
12	**Implementation/Migration Phase**	**38 days**	**Mon 7/22/13**	**Sat 9/7/13**
13	Cluster Migration Configuration	3 days	Mon 7/22/13	Wed 7/24/13
14	Pilot Implementation Test	3 days	Thu 7/25/13	Mon 7/29/13
15	**Production Migrations**	**32 days**	**Tue 7/30/13**	**Sat 9/7/13**
16	Production Data Transfers	15 days	Tue 7/30/13	Mon 8/19/13
17	**Production Cutovers**	**13 days**	**Sat 8/24/13**	**Sat 9/7/13**
18	Cutover 1	0 hrs	Sat 8/24/13	Sat 8/24/13
19	Cutover 2	0 hrs	Sat 8/31/13	Sat 8/31/13
20	Cutover 3...	0 hrs	Sat 9/7/13	Sat 9/7/13
21	End Implementation/Migration Phase	0 days	Sat 9/7/13	Sat 9/7/13
22	**Final Project Documentation, Acceptance and Closure**	**5 days**	**Mon 9/9/13**	**Fri 9/13/13**

Figure 1: Sample migration plan

The plan addresses how all aspects of the migration are achieved: sequencing, tools, timing, communication, and implementation. After you develop a plan, a proof of concept can help you evaluate the approach and test the phases of the plan.

Toolset selection

After you finish the discovery phase and develop a methodology, you can select a toolset.

Testing the migration methodology

After you develop a migration methodology, you must review, validate, and test the migration plan. A test migration is usually run on a subset of the data. Running a test migration is also invaluable in estimating the performance and timing of a migration.

Data migration testing

Testing the actual data movement process and execution is the first phase in the testing of the overall methodology. The data migration testing determines whether the proposed methodology meets the requirements and accomplishes the goals of the project.

Role of data migration testing:

- Validates the tool selection; does the tool do what you want it to? Does it copy the data and attributes?
- Validates the data transfer; is the data moved as expected?
- Validates that the permissions are copied over; are they correct, functional, and operational?

- Benchmarks the data-transfer performance; how long does it take to run full and incremental transfers?
- Tests the new data; is it available? Are the read-write settings correct? Does the new workflow work?
- Gives you the option to experiment with different methods, tools, and flags
- Lets you tune the process to achieve the best results

The testing should give you confidence that the data will be accessible and available after all the data and users are cut over to the new system.

User acceptance testing

Before you execute the full migration and cutover, user acceptance testing (UAT) should be undertaken against the new storage system and a sample of the migrated data. UAT validates that the data is ready for cutover:

- Data is accessible; users and applications can access the data correctly
- Permission models are correct; the required security is applied to the migrated data
- Workflows are operational; no issues with the data's use

Cutover methodology testing

Cutover methodology testing helps determine how you will move client connections—and how the clients will respond to the cutover. Through testing, you can gauge how long it takes to move the connections, what kinds of issues may occur, and how to troubleshoot the issues. Testing the cutover strategy thoroughly provides feedback on how to execute the final cutover.

Rollback strategy testing

You should also test your rollback methodology. The rollback testing should validate that your plan to failback or abort a migration works, so that you are prepared in case there are issues during the cutover. Make sure that access to the data on the old system can be restored quickly and efficiently without affecting users.

Executing the migration

Having completed and validated all your methodology and processes, you can move on to the main migration.

Core migration phases:

- Data transfer: all the data is migrated from the old system to the new system
- Cutover: connections and clients are moved to the data on the new storage
- Acceptance: the new data source is ratified
- Rollback: only if required
- Steady-state and repeat: migration phase considered complete but additional separation migrations may occur
- Post-migration monitoring: the new system and data is monitor following the cutover

Data transfer

A standard approach is to first execute a full-data-copy to move all the initially identified data, and then to follow it up with a series of incremental copies, which only move data that changed after the initial full copies had run. This gives you the most flexibility in executing the cutover, as the additional incremental copies will be substantially shorter to execute than the initial large data copies.

Cutover

After you have migrated the data, the act of moving clients from the old storage to the new storage will occur during a cutover event.

High-level cutover plan:

- Old source: remove write access to ensure clients are unable to write any new data
- Execute a final incremental copy to migrate any remaining data from the old system to the new system
- Test new target data and connectivity; selective UAT
- GO or NO GO: decision to move forward with cutover event
- Update the client-to-storage connection mechanisms: DNS, DFS, VIPs, etc.
- Monitor the new storage system: monitor load and connections as the clients cut over
- Validate clients: review and validate that clients can successfully connect and operate
- Cutover complete

Acceptance

Having migrated the data and client connections over to the new storage solution, the storage availability and workflow acceptance of the new data and storage solution must be validated.

Once you begin writing new data to the new storage system, the ease with which you can roll back to the old storage system diminishes. If you were to roll back to the old system, the changed data would need to be copied back to the old environment. Unless this newly written data can be discarded, re-written, and manually reconciled, it is strongly suggested that any rollback be executed before any large-scale writing of data has occurred to the new storage.

Rollback

Make sure you have a fully tested rollback plan in place. A rollback may be needed for a variety of reasons:

- Client connectivity or storage name resolution issues develop following cutover
- Final incremental migration not completed in outage window, so not all data is migrated
- Unplanned IT outage or issue occurring at the same time
- Data access on the new storage is invalid and workflows are affected

The goal of a rollback plan is to quickly restore the access to the old data storage solution. Assuming the cutover was executed correctly, restoring the prior access should be straightforward and should be able to be implemented with minimal additional disruption. The primary goal would be to restore access within the cutover window so no additional downtime and interruption to data is introduced. It is critical to have a tested rollback plan that can be used if an issue with the cutover occurs.

Repetition

After you validate the data transfer through cutover and client acceptance, most migration projects consist of multiple migration cycles. The methodology can be executed again on different sets of data in migration waves that encompass the entire project.

Post-migration

Following the migration cutover, it is important to monitor both the new storage system and the old storage system. You should observe that client connections are moving to the new system and that active data connections are no longer getting initiated on the old storage. Can clients connect and work with the new storage systems without issue? As connection counts increase on the new storage system, you should monitor the load and performance and make performance adjustments as needed.

You should be monitoring the following items during and after the cutover:

- New system: system load and performance, number of connections, movement of users, security, and performance
- Old system: are users still connecting to it? Are there legacy connections to it from old applications?

You should have a transition plan of what to do with the old storage. Some common approaches include:

- Keep it around for a while but with administrator access only
- Provide read-only access for users
- Mothball the system while the new systems transitions
- Decommission it

Single protocol SMB data migration

Although this paper addresses a single SMB protocol data migration, the approach and many of the best practices can be used as a foundation for other types of data migration.

Challenges of single protocol SMB data migration

Moving large amounts of data presents a number of challenges:

- It may be difficult to perform without downtime; most large Microsoft Windows® environments rely on the data being available at all times.

- It often has complex security implementations; most large Windows environments rely on the extensive use of ACLs to enforce security. ACLs can be challenging to migrate and validate after migration.
- A large number of shares may need to be migrated. You must move not only the data but also the Windows shares and share permissions. This introduces a second type of migration that must be undertaken during the project.
- It may include a large number of differently connected clients that require cutover and validation; often Windows clients connect differently to their file shares, mapped drives, startup scripts, DFS, and DNS. All the different connection types need to be identified, cut over, and validated.
- It may have a high rate of change; large Windows environments often contain many concurrently connected clients. You must account for the rapid rate of change of data during cutover.

Data-specific considerations

When you design a migration strategy, you should plan how your data will appear after it has been migrated. The metadata of files, in particular, can add complexity to a migration. With Windows files, you should identify the metadata that you want to migrate with the data. The following metadata can affect your migration strategy:

- Security identifiers, ACL structure, all ACEs on a file or directory
- Security inheritance
- File properties such as access time, created time, modified times, owners
- File attributes such as read-only or archive (an Isilon cluster does not support compressed and encrypted)
- Extended proprietary file attributes that are in use; not supported on Isilon
- Local users and groups; are these defined on the files?
- Deduplication in use, archive stub files, Mac OS X resource forks present

How does the data need to appear post-migration:

- Direct replication of all data and attributes
- Move the data, then make updates, fix problems, change the security, etc.
- Migrate just the data and implement an entirely new security model

Migration requirements and customer data collection

Before you can plan your migration, you must collect requirements.

Requirements gathering

The data migration planning begins with identifying the data that you want to move from the old storage system to the new storage system. Here is what you need to document:

- Current state—what is the current state of:
  - Source infrastructure
    - Existing storage platforms

        - Network design and implementation
        - Name resolution infrastructure: DNS, DFS, global namespace
        - Servers/clients/OS/applications
    - Source infrastructure configurations
        - Volumes
        - Shares
        - Access
        - Authentication
    - Source data
        - Logical structure: data layout, directory depth
        - Physical structure: total size, min/max/average file size
    - Source data security
        - Current security model and how file access is enforced
        - Local users and groups
        - POSIX permissions
        - Windows users and groups from Active Directory
        - LDAP users and groups
- Target state—What will be the target state:
    - Target infrastructure: Isilon cluster
        - Network configuration
    - Target configurations:
        - Directory layout and structure
        - Shares
        - Access and authentication model
    - Target data
        - Logical structure—same as source or new
        - Physical structure—same as source or new
    - Target data security
        - Same as current security model
        - Migrate and change the security model
        - Move the data and implement a new security model

How to gather the data

- Interview stakeholders
- Gather documents: network diagrams, run books, infrastructure, and application details
- List of shares

- Storage reports, etc.
- Review ACLs and share permissions

Current infrastructure and data analysis

Start the migration design phase by collecting the data needed to develop the migration requirements.

Best practice: Standardize your documentation

Create and utilize a standardized data collection and migration-planning document along with a standard target configuration guide. By using a structured document to gather and collect all your source data and information, you can identify your migration requirements, which will lead to clear migration design decisions.

Why: To simplify and consolidate migration planning.

You need to collect the following information:

- How much data: actual file data, not compressed or deduplicated
- How many directories and files; identify the directory trees and quantity
- What is the directory structure: shallow and flat, wide and deep
- How many shares; do you see share name collisions or reuse on multiple source hosts
- What is the use of these shares: home directories, application or group use
- How ACLs are applied to source; individual or at the group-level
- How many source locations; single source system or multiple
- How clients access data; protocols and how they resolve storage names
- Rate of change of files; how often and where are files changing
- Networking architecture; source systems and network between it and the Isilon cluster
- Source system load; understand the load the source storage is under

Determine migration methodology

After you collect information on the source system, the data, and the infrastructure, you are ready to develop a migration methodology.

Logical migration design

By analyzing the structure and layout of the source data, you can make logical migration design decisions—structuring the migration into distinct executable units. A goal of the migration methodology is to identify logical boundaries to facilitate the cutover of your clients and workflows.

Some logical migration boundaries are as follows:

- Hosts/Filers/Servers/Arrays
- Volumes
- Shares
- Directories—users or groups

Best practice: Define migration boundaries

Identify well-defined data structures to migrate and cut over—for example, entire shares or volumes. Be aware of the size of data inside a migration boundary, as the size of the data affects the outage window required to complete a cutover.

Why: To organize the migration into segment waves, making the migration easier to manage.

After you segment the logical boundaries into distinct migration phases, you can address other elements of your workflow, such as metadata, that you need to migrate.

File attributes and security

Most data migration also includes the migration of files' metadata: ownership, access times, create time, and security descriptors. Before you can execute your migration, determine how you plan on handling metadata and file security.

Common migration approaches:

- Migrate data file as-is (no change to ACLs).
- Migrate data and ACLs but also fix on destination (recalibrate the ACLs).
- Migrate data-only; create new ACLs on the destination, or create a new security design.
- Migrate away from an existing security model and implement a new model. It is recommended that you use a central authentication scheme on Isilon. For example, if the NAS system that you are migrating from uses several directory services, you should consider consolidating the directory services into a single directory service for the new NAS system.

Best practice: Understand the data attributes of to be migrated data

Be data aware: Identify any DOS attributes, non-standard extended file attributes, and non-standard permissions that are not supported by an Isilon cluster. Also, identify your local users or groups and have a plan to deal with them.

Why: Before you execute the migration, you may need to take additional steps to prepare the data for migration so it will be available on the new storage system.

Migration sequencing

The execution of a migration will likely require multiple iterations of the data transfer. Source data is constantly changing—until the source data is locked in a read-only state or access is denied. Once access to the source data is removed, the final data transfer can take place. Otherwise, there may be differences between the data on the source and the target.

The recommended approach for a data migration is to use a multi-step migration. A multi-step migration consists of an initial “full” or “level 0” data copy. The initial data copy is followed by a series of “incrementals” that update only new or changed data. The initial data copy moves an entire copy of the source data. It can often take a long time to execute because all the data must be assessed and transferred over the network to the migration target.

After the initial copy completes, additional differential transfers copy only the data that has changed since the initial “full” was executed. Additionally, any data that’s deleted on the source will also be deleted on the target through the incremental process. The size of an incremental copy is affected by the rate of change of the source data.

You should run multiple, over-the-top incremental copies to guarantee the integrity and consistency of data that encounters issues during the initial full copy. Incremental copies will also keep the two data sources in sync with each other and require less catch-up on final copy.

A final incremental copy should always be executed as part of the migration cutover plan to ensure all the latest data is on the new target storage.

Best practice: Run initial full copies followed by incremental copies

Always execute a final incremental copy during cutover to ensure the latest data from the source is migrated.

Why: Executing multiple migration passes will ensure that all the data is transferred and that the latest version of the files will be stored on the target storage system.

Type of migration

You must determine how the migration will be executed. There are two possibilities: an indirect execution from a host and a direct execution from an Isilon cluster. With host-based migration, an intermediary host executes a copy process between the source system to the target system through the host, as shown in the following figure:

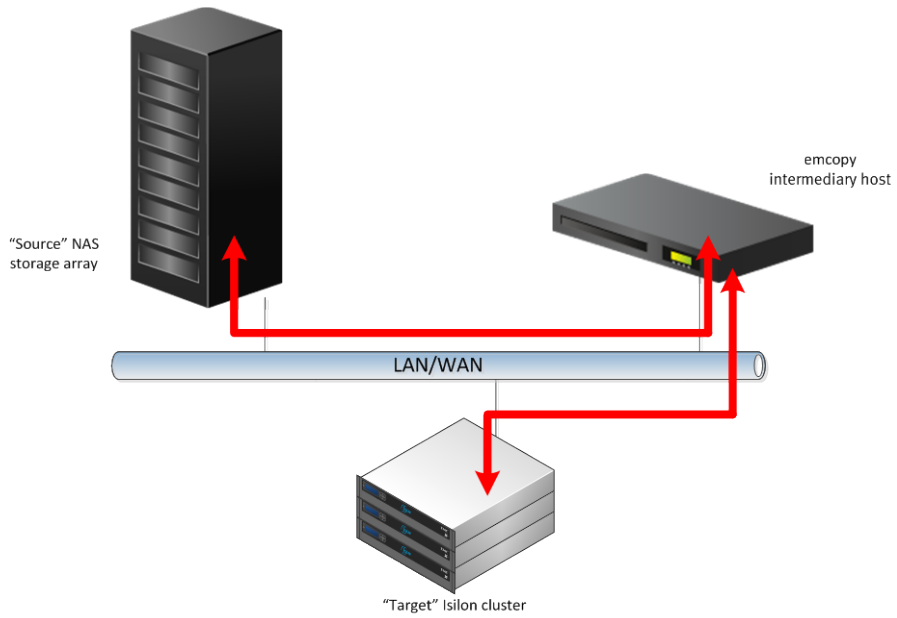


Figure 2: Host-based migration

With a host-based migration, data is transferred through an intermediary host on route to the Isilon cluster.

If the source system is supported, the Isilon cluster can execute a direct source to Isilon data copy by using the `isi_vol_copy` command over NDMP or by using rsync.

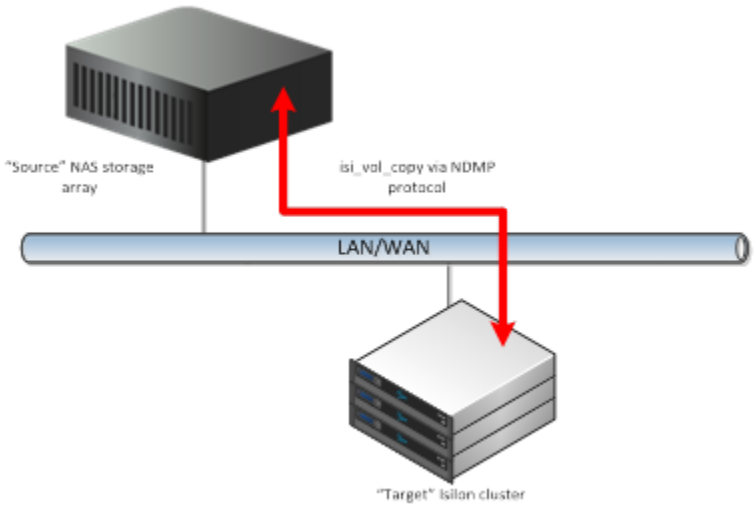


Figure 3: Isilon-based migration

With an Isilon-based migration, data is pushed or pulled directly to the Isilon cluster.

Host-based migrations

A host-based approach might be selected for a number of reasons:

- Source system does not support Isilon-based migration: `isi_vol_copy` not supported
- Connectivity is restricted: storage on different networks, a host may bridge the networks
- Flexibility in execution: separate the execution from administration of the storage systems
- Security restrictions: can be used to limit access to systems

In a host-based migration, the toolset executing the migration makes a connection to the source and to the target system and then copies the data through the host. For the purpose of this paper, the two primary host-based tools are EMCOPY, by EMC, and Robocopy, by Microsoft.

Best practice: Select a suitable host

Select a suitable host to run the migration. **Why:** Because all the data will move through the host, incorrect sizing may lead to a bottleneck or an interruption in the migration. Using multiple hosts may facilitate multi-streamed migrations, in which you can maximize network usage and the Isilon nodes by executing multiple migrations concurrently.

Some common considerations include:

- Adequate resources to execute the migrations; CPU, RAM, Network
- 10 Gigabit network infrastructure where possible
- Connectivity between host and the source and target storage systems
- Availability; the host is stable and reliable, no reboots or downtime
- Dedicated host; not running a lot of other parallel workloads and restricted user access

The migration host needs to be optimized for the migration workload and network throughput, as it will send and receive all the data to be migrated.

Isilon-based migrations

If the source system is capable of supporting an Isilon-based migration by using `isi_vol_copy` or by direct access with rsync, the connectivity exists and the migration methodology supports utilizing this approach, and it may be a more efficient technique. The main advantage of the direct approach is there is no need for an intermediary host to execute the process or for the data to traverse the host.

Type of Isilon-based migration

There are two primary types of Isilon-based migration:

- NDM-based with `isi_vol_copy`

- rsync-based: use the UNIX rsync tool to connect and either push or pull data to the Isilon target

Known source systems that support Isilon-based NDMP migrations:

NetApp®

Isilon Requirements: Isilon OneFS 6.5.5.6 or later/newer

NetApp Requirements: Data ONTAP 7.x or Data ONTAP 8.x operating in 7-mode

It is anticipated that additional source systems will be supported in future releases of Isilon OneFS.

It's critical to evaluate the migration methodology against the selected approach to determine if the method selected will facilitate the migration goals.

Best practice: Evaluate migration approach

Select the migration method that meets your specific migration requirements, provides cutover flexibility, and optimizes data throughput.

Why: The selected approach will impact the migration schedule and planning.

Having identified the migration approach, the selection of the appropriate migration tool can occur.

Migration tool selection and use

The data migration requirements will help define the tool selected to execute the data migration.

Tool selection

A number of tools are available and will work; any file copy that can connect over SMB to the source and target storage can be used to move data between the systems. It is recommended that you use a tool that can be automated and provides robust functionality—a tool that can copy attributes, security, logging, etc.

The common SMB data copy tools are as follows:

Tool	Advantage	Disadvantage
EMCOPY	• EMC preferred tool • Lots of available switches • Multithreaded copies • Can perform file hashing on source and target	• Unable to remove some DOS attributes • Limited error reporting
`isi_vol_copy`	• Included with Isilon OneFS • Pulls across all users and	• Only supported against specific source storage

	permissions • Supports both SMB and NFS protocols. • Utilizes NDMP • Direct migration source-target	systems (NetApp systems running OnTap 6.5 and above) • Limited error reporting
Robocopy	• Free, included with Microsoft Windows • Widely available and used • Lots of available switches	• Limited error reporting • No vendor support • No alerting
SecureCopy	• Scheduling and logging • Widely used	• Not free

Table 1: Summary of SMB copy tools

Tool versions

It is important to understand that different tools may behave differently on different hosts; it is strongly suggested that you test tool versions and observer the behavior.

Best practice: Use the correct version of the tool

EMCOPY and Robocopy are available in 32-bit and 64-bit versions. You must use the correct version for the host that is running the tool.

Why: Using the correct version of the tool will optimize throughput and performance.

Best practice: Use the latest version of tool

It is recommended that you always utilize the latest versions of the chosen file copy tool.

Why: Performance is optimized and they often have newer features.

Migration tools

The following is an overview of the use of the primary SMB migration tools that can be used in Isilon data migrations:

EMCOPY

Overview:

The EMCOPY tool provides a method to copy files, directories, and subdirectories from SMB shares to other SMB shares with the security and attributes intact. The tool was developed by EMC and is available to all EMC customers with active support contracts with EMC.

Usage:

```
emcopy.exe <source> <destination> [options]
```

Features:

- Ability to copy file data, Windows ACLs, ownership, time-stamp information
- Ability to retry file copies
- Supports the Windows Backup Operators, back up right to backup files that it does not have access to
- Supports multithreading
- Ability to perform enhanced file comparisons

For a full list of EMCOPY features and switches, run the following command:

```
emcopy.exe /?
```

Sample EMCOPY command:

```
emcopy64.exe S:\users T:\users /ignoredhsm /o /sd /s /d /th 64 /c /purge
/r:1 /w:2  /log:<path_to_log.txt>
```

Best practice: Starting EMCOPY switches

Suggested initial EMCOPY switches:

`/o =`	Copy owner security information
`/d =`	Copy only the source files when its last modification time is greater than the destination file's modification time
Or	
`/de =`	Copy the source file when its last modification time is not equal to the destination file's modification time or when files size are different
`/sd =`	Do not copy the file content if any error occurs on the security descriptor
`/s =`	Copy all files in the subdirectories
`/purge =`	Remove extra files and directories from the destination tree
	The directories or files including denied rights for delete for the user of EMCOPY are not removed
`/log:`	`<path_to_log.txt>` = Set log filename to the path; erase the existing file
`/w:2 =`	Specify the time in seconds to wait between two retries, 30 by default
`/r:1 =`	Specify the number of retries, 100 by default
`/c =`	Continue after the number of retries
`/th n =`	Number of working threads: by default 64 working threads are created;
	working thread count must be in the range of 1 to 256

`/ignoredhsm =`	Remove special DHSM ACE, reset offline attribute when file content is fully rehydrated

Why: It is suggested that you start with a baseline of switches and test the copy, validate the results and behavior of the copy, and make the appropriate adjustments to the EMCOPY switches. No one default set of switches will work for all migrations.

You should become familiar with all the EMCOPY switches and their use. The following highlights a few possible options that you should be familiar with. It is important to recognize that each migration will require different switches due to the unique requirements of each dataset.

A few useful switches to be aware of include the following:

`/d vs. /de`

`/d` — Copy only the source file when its last modification time is greater than the destination file's modification time.

`/de` — Copy the source file when its last modification time is not equal to the destination file's modification time or when the file size is different.

There may be times when a file's modification date is not sufficient to determine that the file has changed. Using the `/de` switch will also look at the file's size. This switch should only be used on incremental copies because you gain no benefit with it on full copies.

`/secforce vs. /secfix`

`/secfix` — Fix the NT security properties on existing destination files or directories.

`/secforce` — Forces EMCOPY to overwrite, instead of merge, security information.

If the source and target permissions are different for some reason, you can force-replace the target permissions with the source permissions. A `/secfix` merges the permissions, which may not be what you want.

`/purge`

A `/purge` will remove all extra files and directories from the destination tree. The directories or files including denied rights for delete for the user of EMCOPY are not removed. It should be used to delete any files on the target that have since been removed on the target since the last run.

`/sdd`

Can be used to synchronize target dates to source dates.

`/cm <bin|md5>`

`bin` — Content of source and target file are compared after the copy.

In evaluation mode, contents are checked if other criteria are verified. Activation of this mode could reduce performance.

`md5` — an MD5 signature of the source file is computed during the copy.

The target file is read after the copy and an MD5 signature is computed. Both MD5 signatures are then compared. Activation of this mode could reduce performance.

`/stream`

Copy all data streams of files and directories. Can be useful to copy files that include alternate data streams (ADS).

`/tee`

Is useful to push the output of the copy job to the console during testing to review what is occurring and view errors. You should not use this switch while running real copy jobs as it can impact the performance of the migration.

`/xf` and `/xd`

`/xf file` [file ...]

Can be used to exclude files with the specified names, paths, or wildcard characters, if needed, and system files or temporary files.

`/xd dir` [dir ...]

Can be used to exclude directories with the specified names, paths, or wildcard characters, if needed.

Example: `/xd .etc lost+found` — skip the .etc and lost+found directories that are created in the top level of Celerra file systems.

`/ignoredhsm`

Will remove the special dynamic hierarchical storage management (DHSM) ACE added by some archiving solutions using stub files; it will reset the offline attribute when file content is fully rehydrated.

Best practice: Know the EMCOPY switches

Understand all the EMCOPY switches, as well as when and how to use them.

Why: Different migrations will require the use of different switches to meet the requirement of the data copy and the final state in which the data is needed.

Best practice: Using the `/th` correctly; the working threads flag

When using the `/th n` flag, test and validate behavior with different settings. Benchmark performance and tune the copy accordingly.

Why: You may not always see increased performance with higher values based on the workflows and the nature of the copy execution.

Best practice: Use mapped drives

Use mapped drives from the migration host to the source and target systems.

Why: Provides persistent connection to storage systems, simplifies connection strings, and allows alternate credentials to be used.

Robocopy

Overview:

Robocopy is a file and directory copy tool created by Microsoft. It is freely available in all versions of Microsoft Windows. The tool provides a number of options to copy file data and security attributes.

Usage:

```
robocopy <source> <destination> [options]
```

Features:

- Ability to copy file data, Windows ACLs, ownership, time-stamp information
- Ability to retry file copies
- Supports the Windows Backup Operators, backup right to backup files that it does not have access to
- Newer version supports multithreaded copies

For a full list of Robocopy features and switches, run the following command:

```
robocopy /?
```

Sample Robocopy command:

```
robocopy64.exe S:\ps T:\ps /zb /e /r:1 /w:1 /copy:datos /tee /purge
/mt:16 /log:<path_to_log.txt>
```

Best practice: Robocopy switches

`/zb =`	Use restartable mode; if access denied, you should use backup mode
`/e =`	Copy subdirectories, including empty directories
`/r:1 =`	Retry once on a failed copy
`/w:1 =`	Wait one second before retrying
`/tee =`	Write status output to the console window and to a log file; used for testing
`/purge =`	Deletes destination files and directories that no longer exist in the source
`/log =`	Log to the log file that you specify
`/copy:datso =`	Copy data, attributes, time-stamp, security, and owner
`/mt n =`	Perform multithreaded copies with n threads (default 8); n must be at least 1 and not greater than 128; this option is incompatible with the `/IPG` and `/EFSRAW` options

You should become familiar with all the Robocopy switches and their use. The following highlights a few possible options that you should be familiar with. The switches that you select depend on the unique requirements of your dataset. For more information, see the Robocopy main page.

`/purge`

A `/purge` removes extra files and directories from the destination tree. You should use this switch to delete files on the target that have been removed on the target since the last run.

`/tee`

This switch is useful during testing to push the output of the copy job to the console so you can review the output. You should not use this switch while running real copy jobs, as it can impact the performance of the migration.

Best practice: Know the Robocopy switches

Understand all the Robocopy switches and when and how to use them.

Why: Different migrations will require the use of different switches to meet the requirement of the data copy and the final state of the data.

Best practice: Use mapped drives

Use mapped drives from the migration host to the source and target systems.

Why: Provides persistent connection to storage systems, simplifies connection strings, and allows alternate credentials to be used.

Best practice: When using `/mt` log only to logfile

When using `/mt n` with a high number of threads, redirect output using the `/log` option for better performance.

Why: Higher throughput is observed when not teeing the output to the screen; do not use `/tee`.

Best practice: Using the `/mt` — the working threads flag

When using the `/mt` flag, test and validate behavior with different settings. Benchmark performance and tune the copy accordingly.

Why: You may not always see increased performance with higher values based on the workflows and the nature of the copy execution.

Isilon-based migrations — `isi_vol_copy`

Overview:

`Isi_vol_copy` is a native Isilon OneFS tool that supports data migration through the use of the NDMP protocol. The tool allows the cluster to mimic the behavior of a backup target, and allows the data to be copied directly from the source system to the Isilon cluster.

Usage:

```
isi_vol_copy <src_filer>:<src_dir> <dest_dir>
                  [-sa user: | user:password]
                  [-sport ndmp_src_port]
                  [-full | -incr]
                  [-dhost dest_ip_addr]
                  [-maxino maxino]
                  [-h]
```

Features:

- Utilizes native NDMP functionality and connectivity
- Supports full and incremental backup levels
- Migrates data and all security, attribute information
- Will restore the set of permissions/ACLs that existed on the source data
- Will migrate NFS and SMB source data
- Does not impact or interact with client data access
- Dedicated data transfer pipe between source and cluster
- Starting with 7.0.2, it supports the Backup Restartable Extension, so full backups can be interrupted and restarted from a checkpoint

Sample `isi_vol_copy` command:

```
isi_vol_copy  <source_IP>:/<source> /ifs/ -sa<ndmpuser>:<ndmppassword> -
full
```

Best practice: `isi_vol_copy` use — data access on target

Do not touch the data on the target Isilon cluster until after the `isi_vol_copy` has completed.

Why: This would create problems and you may have to redo a full copy.

Best practice: `isi_vol_copy` use: multiple copies to same target

Do not execute multiple `isi_vol_copy` going to the same target. In other words, don't have all your `isi_vol_copy` going only to `/ifs/data`.

Why: Creates problems for the copy process and may require remediation.

Best practice: `isi_vol_copy` use — load monitoring

`isi_vol_copy` is optimized to stream as much data as possible across a network, so you should always monitor the load on the source and target systems for potential impact.

Why: Since `isi_vol_copy` is optimized to stream as much data possible, don't overwhelm

older source systems and create potential link saturation or disk problems.

Migration preparation

After you finish planning the migration and selecting the tools, you can prepare the source and target systems for the migration.

Infrastructure and environment setup

Network connectivity

Since all the data in the migration will traverse the network, you should optimize the network infrastructure and connectivity between the source system and target.

Common recommendations include the following:

- Maximize network; 10 Gbps is preferred to 1 Gbps, optimize end to end
- Limit hops and latency between source and target
- Isolate migration traffic
- Limit potential network bottlenecks; routers, firewalls, IDS, and shared network infrastructure

Best practice: Optimize the network for the migration traffic

Optimize the migration network path by limiting other production traffic from this network and limiting network devices the traffic traverse (firewalls, IDS, etc.). Ideally, look to create a dedicated, private migration network that can be optimized for only the migration traffic.

Why: Separating the migration traffic from other network traffic will allow for maximum throughput and reduce the potential impact to existing production traffic by limiting network saturation.

Migration account

For the migration data to be copied from source to target, the tool accessing the data must be able to access the source and target data.

Commonly used migration accounts:

- Accounts that are members of the Windows Domain Administrators Group
- Accounts that are members of the Windows Backup Operators Group
- User accounts created explicitly for the execution of `isi_vol_copy`

The account used to connect to the source and target storage systems depends on the security model implemented in the environment.

Best practice: Use a specific migration account to execute migration tasks

Use a specific migration account or account with group membership that has the required access to all source and target data.

Why: Using a dedicated account will allow for oversight and management of the migration data access. It will also allow for the separation of migration tasks and users from other production accounts.

Source host preparation

The source data storage system should be prepared and optimized for the migration.

Best practice: Control access during cutover

Create source-specific migration shares, create a hidden share, or restrict access only to an administrative account that will be used during migration and after cutover.

Why: Allows you to separate migration events from normal production access—for example, you can disallow one but allow the other. This can be used during post-cutover to continue to allow access to the migration account but deny read-write to normal users. This prevents updates to the data during data cutover and post-migration, while continuing to allow administrative access.

Migration host preparation—source and target access

The migration host should be prepared and optimized for running the migration copies.

- Limit workload and access to optimize throughput
- Restrict access and reduce service issues with the host
- Prepare all migration jobs as scripts
- Test and validate network throughput

Best practice: Use mapped drives from the migration host

Map source and target network drives from the migration host.

Why: Doing so simplifies the execution of copy jobs. Alternate credentials can be used if needed to access the source or target systems.

Isilon cluster preparation configuration

All primary setup and configuration of the Isilon cluster should be completed before you begin to migrate data. The configuration includes but is not limited to the following:

- Authentication provider integration—all authentication providers are online and fully operational
- Access zone and RBAC setup—complete any zone and RBAC setup
- Networking design and setup—complete the setup and implementation of the network configuration

- SmartPools—complete the implementation of any SmartPools policies to limit post-migration work
- SyncIQ®—prepare any existing SyncIQ policies to operate alongside any data migration events
- SnapshotIQ—prepare any Snapshot policies to operate alongside any data migration events
- SmartLock®—execute all preliminary SmartLock work prior to migration

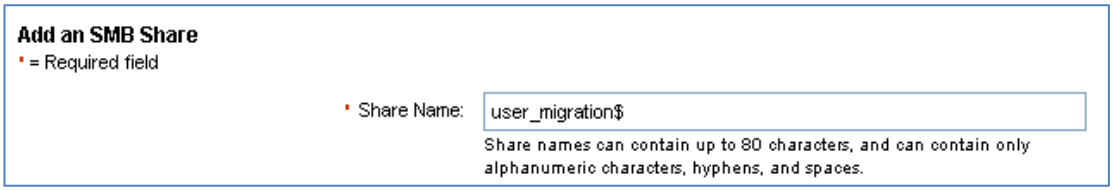
It is suggested that you use an Isilon migration share to execute all migrations against. Using a dedicated administrative migration share with the appropriate access and share configuration can facilitate the migration without impacting workflows or data permissions.

The following best practices are recommended for the creation of an administrative migration share.

Best practice: Create a dedicated SMB migration share

Create a dedicated migration target–specific hidden $share

Why: A hidden $(dollar sign) share will hide the share from users browsing the cluster's SMB shares



Best practice: Using `/ifs` as the administrative share will give flexibility

Creating the migration share as `/ifs` provides the most flexibility.

Why: Having a hidden share at the root of OneFS will provide access to the entire file system for all migrations.

When you create a share, it is critical to understand the impact of the permissions settings on the directory ACLs. Selecting "Applying Windows Default ACLs" will update the ACLs on the folder selected with a default set of ACLs. Applying default ACLs can have a major impact on users and workflows. Make sure you clearly understand the impact of this setting on pre-existing directories. In most cases, selecting "Do not change the existing permissions" is the recommended selection, because it will leave the existing permissions on a directory intact.

Best practice: Do not change permissions on creation of the administrative share

When creating the administrative share, always select "Do not change existing permissions."

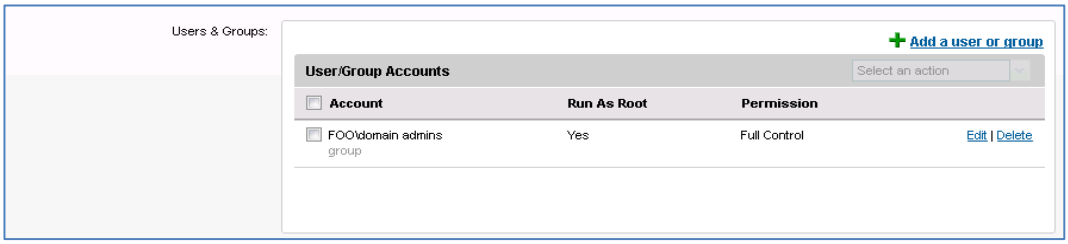
Why: By selecting "Do not change existing permissions" you will not make any changes to the directories' existing ACLs, which could impact cluster operations. You can always edit the directories' ACLs after share creation.

EMC Isilon stresses the potential impact of "Applying Windows Default ACLs" to `/ifs`; be sure you understand the consequence of this action.

Best practice: Set restrictive share permissions on the administrative share

Limit access to the administrative migration share to only administrative and migration-specific accounts. Use a Windows Domain Administrator or a Windows Backup Operator account for the account executing the migration, because it will, in most cases, facilitate access to the source data.

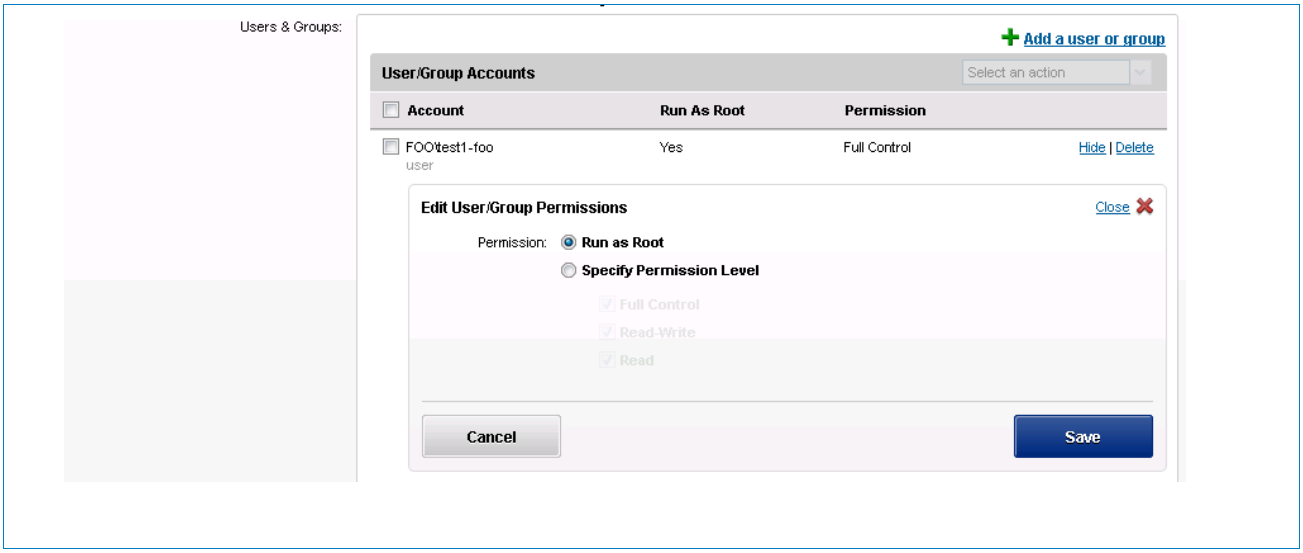
Why: This will limit data access and prevent issues around unauthorized use.



Best practice: Enable run-as-root on the administrative share

Enable run-as-root access on the administrative migration hidden share.

Why: Enabling run-as-root access on an Isilon share facilitates the ability of the connected account to have unrestricted access to the share and directory structure regardless of the actual permissions set on a directory or file. This is useful, because it allows the account executing the migration to read and write data to the target directories without having to make modifications to the directory access control list (ACL). Using the run-as-root feature allows you to bypass having to create migration-specific access control entries (ACEs) on the directory structure to migrate data.



An alternative approach would be to rename the default `/ifs` share to `/ifs$` and then make changes to the share permissions and allow "run as root" on this share if the `/ifs` share is not in use. Using a dedicated `migration$` share will allow for administrative isolation of the share used for access and potentially different security to be enforced.

Having created the administrative migration share, validate access to it from the migration host system. It is suggested that you map a drive from the host system to this `migration$` share.

EMC Isilon also suggests that you pre-create all the client-facing SMB shares on the cluster before planning for the data migration and cutover. Pre-creating the shares lets you test data access, authentication, and workflows. The migration tools can be used to create the base directory structure for share creation prior to actual data copies occurring.

Best practice: Create the directory structure for SMB share creation

Build the primary top-level folder structures:

Robocopy: use the `/LEV:n` — only copy the top n levels of the source directory tree

EMCOPY: `/lev:n` — specify the level of directory depth

Why: This will facilitate the creation of the user's share prior to data migration.

Best practice: Create the SMB shares

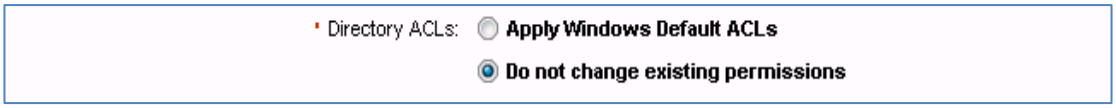
Create the new Isilon SMB shares prior to data migration.

Why: This will allow the creation and setup of the shares and share permissions prior to data migration and cutover for testing and access validation.

Best practice: Do not change existing permissions on directory when creating share

Again, when creating the client shares, select "Do not change existing permissions." Otherwise, the share creation process will modify any explicitly defined ACLs that were added during the initial directory setup.

Why: Selecting "Apply Windows Default ACLs" will overwrite explicitly defined ACLs and lay down a default set of ACLs, which may be in conflict and require remediation before the data migration can continue.



Best practice: Create the correct SMB share permissions

Setup the correct share permissions on the newly created user shares.

Why: Setting the correct share permissions will allow you to test and validate workflows when test migrations are undertaken.

Note: The migration methodology may include adding an explicit "Deny" share permission on users so they cannot write data to these shares until the cutover has been executed.

Best practice: Review and validate the Isilon cluster ACL policies

Before migrating data and permissions to the cluster, review that the cluster ACL policies are set correctly and are appropriate for the migration.

Why: Reviewing the ACL policies will ensure the cluster is appropriately configured for the migration and operation.

A detailed review of Isilon cluster ACL policies is beyond the scope of this paper; for additional information, see http://support.emc.com.

Additional Isilon cluster considerations

The following are some additional Isilon cluster considerations that may need to be addressed prior to and during a data migration.

Production or pre-production cluster

An important consideration around planning and executing a data migration is the current status of the Isilon cluster. Is the cluster in production or will the migration mark the initial cutover to active production traffic? Our primary goal is to lessen any impact on a production cluster during migration activities, so suitable steps should be taken to address these concerns.

Common factors to be aware of while migrating to a cluster:

- Administratively destructive actions

- Saturation of network links
- Cluster load and ingest, impact on production workflows

Access zones and role-based access control (RBAC)

If the cluster uses an Isilon access zone or RBAC, the migration methodology may need to be adjusted to accommodate this configuration.

SmartConnect or direct node connections

The current status of the cluster may dictate that you look to optimize and segregate migration traffic within the cluster's network configuration.

- Use SmartConnect to auto-balance traffic
- Separate migration traffic from existing production traffic; use a direct node or separate the SmartConnect zone connection for migration traffic

If you use SmartConnect, you should validate and optimize the configuration before the transfer of data across the network.

SyncIQ considerations

If the data to be migrated will be replicated to a secondary cluster through a SyncIQ policy, additional planning should be undertaken to address the impact of the data migration and its interaction with active SyncIQ policies.

- Pause active SyncIQ policies if they include migration paths
- Schedule SyncIQ jobs to run outside of data copy windows
- Utilize SmartConnect zones for copying and SyncIQ replication

SmartPools

Any SmartPools data policies should be in place prior to data migration, or additional cluster overhead maybe required to move data within the cluster post-migration.

SnapshotIQ

Any active SnapshotIQ policies should be analyzed for impact during the data migration.

Antivirus integration

Review and disable any active antivirus scanning policies that may be running against the target data.

Best practice: Disable antivirus scanning

Disable active antivirus scanning on migrated data during initial full and incremental copies.

Why: The large influx of data associated with the migration can place an excessive load on the antivirus scanning architecture and create a slowdown and potential bottleneck for the inbound data.

Migration approach—testing and proof of concept

Having developed the migration approach, selected the toolset, and prepared the infrastructure for the data migration, you can proceed with your initial testing of the methodology. The goal of the testing here is to validate the outcome—is data migrated, are the permissions moved, and are the time-stamps moved? The testing phase also allows us to tune and modify the migration approach to optimize all parts of the migration.

Recommended testing approach:

- Run full copy—benchmark and monitor
- Review and validate—potentially look at tuning or tweaking the methodology and re-run
- Run incremental copy—benchmark and monitor
- Review and validate—potentially look at tuning or tweaking the methodology and re-run
- Continue to run incrementals—continue to monitor

EMC Isilon recommends that you test different copy methodologies to tune and optimize the throughput while meeting your migration requirements.

Best practice: Execute multiple test migrations to validate the methodology

It is recommended that you execute multiple migration tests on smaller subsets of different data.

Why: Since different data will tend to have different properties and access profiles, it's important to test all data types and determine if the migration methodology will need to be modified for different data sets.

Critical areas to evaluate and monitor during data migration testing are as follows:

- Network performance—throughput, saturation, impact
- Time to execute a full data copy—will allow for refinement of project plans
- Time needed to execute an incremental copy after X days of data change—will help define cutover windows
- Cluster load, source load, host load—will help tune and refine the migration methodology

Best practice: Test all phases of the migration methodology

Execute all steps in the migration methodology to identify the time involved and to verify that the proposed methodology fulfills all the migration requirements.

Why: To identify issues with the methodology before executing production migrations and cutovers.

Data validation

After you migrate the data, you must validate the data and the file attributes. Verify the following aspects of the data:

- File data copied correctly; data is intact and integrity is maintained
- File security, ownership, and attributes migrated correctly
- File time-stamps are correct

Validate that the files have an ACL. Running the Isilon `ls -al` command on the cluster shows a plus sign beside the POSIX mode bits for every file that includes an ACL:

```
vcluster-mav1-1# ls -al
total 48
drwxrwx--- +   3 FOO\test1-foo  FOO\domain users  143 Jun 14 14:23 .
drwxrwxrwx    23 root           wheel              603 Jul 10 16:11 ..
-rwxrwx--- +   1 FOO\test1-foo  FOO\domain users    6 Jun 10 16:46 1.txt
-rwxrwx--- +   1 FOO\test1-foo  FOO\domain users    0 Jun 10 11:02 2.txt
-rwxrwx--- +   1 FOO\test1-foo  FOO\domain users    0 Jun 10 11:02 3.txt
-rwxrwx--- +   1 FOO\test1-foo  FOO\domain users    0 Jun 10 11:02 4.txt
-rwxrwx--- +   1 FOO\test1-foo  FOO\domain users    0 Jun 10 11:02 5.txt
```

Next, review the access control entries on a file by running the `ls -led` followed by a file name:

```
vcluster-mav1-1# ls -led 1.txt
-rwxrwx--- +  1 FOO\test1-foo  FOO\domain users  6 Jun 10 16:46 1.txt
 OWNER: user:FOO\test1-foo
 GROUP: group:FOO\domain users
 CONTROL:dacl_auto_inherited
 0: user:FOO\test1-foo allow inherited file_gen_all,inherited_ace
 1: group:FOO\domain admins allow inherited file_gen_all,inherited_ace
 2: group:FOO\testgrp1-foo allow inherited file_gen_read,file_gen_execute,inherited_ace
```

You should also validate the data. Common methods include the following:

- File size compares
- Checksum/File hash compares—MD5
- Tools—windiff, dumpACL
- Audit and review directory structures

Having reviewed the data attributes directly, it is critical to validate that the data works in client workflows.

Performance

A migration often moves a large amount of data. You must ensure that your migration methodology, toolset, and environment are optimized for performance and throughput to work within the migration timeline. The common areas to focus on when evaluating performance are as follows:

- Identify bottlenecks—attempt to identify the worst performing component
  - Disable antivirus scanning processes on target and/or source file systems during initial migration copies to minimize CPU impact on client access and its potential impact on elongating copy times.

    - Check WAN bandwidth physical (circuit limitations) and concurrency impact on other systems that are replicating data (SAN, backup, etc.) over a shared link. This could have an effect on replication performance for SyncIQ jobs that need to be run to completion before certain cutovers can be conducted.
- Monitor timing of execution—the time of day and day of week tests were executed versus performance
- Collect metrics on the data copies, network throughput, source, host, and target systems—evaluate the copy as a whole

Best practice: Time the incremental copies

Benchmark the incremental copies by timing how long they take to execute, so you can plan and orchestrate the cutover phases appropriately.

Why: Knowing how long an incremental copy will take will likely determine the length of time required to execute a cutover and will help determine the data outage window.

User acceptance testing—data and workflow testing

The final step of migration data testing is the user acceptance test (UAT), in which the data is tested for integrity with existing workflows. We suggest using test workflows, as this data should only be considered test data and may be removed by later migration steps.

Best practice: Check workflows with test migrated data

Review all workflows on test migrated data.

Why: It is critical to validate that newly migrated data can be integrated into workflows (e.g., user home director or group share access, etc.) at cutover time without issues. By testing the workflows, you ensure that cutovers occur without incident.

Begin migration execution

After you complete all the testing and validation, you can begin to move into the production migration phases. All the information obtained from testing and tuning should be used to modify and optimize the overall methodology, so that the production migrations are as clean and quick as possible.

The migration execution phases are as follows:

- Execute the initial full copy
- Execute incremental copies to keep the new storage up to date and as close to production as possible
- Based on performance, multiple migrations may be executed simultaneously, if supported and if the infrastructure can support the load

Best practice: Continue to run incremental copies

Continue to run incremental, even if the cutovers are not scheduled.

Why: This will keep the source and target data closer in sync and require less data transfer during the final pre-cutover copy.

Depending on the size of the data migrations, the initial full copies may take a while to execute. During this time, you can prepare for the final cutover events.

Pre-cutover preparation

After you start to migrate data, you can begin to prepare your cutover events.

Best practice: Create a detailed migration plan

Create a detailed migration plan with all the specific steps and the timing of the migration's execution, detailing how the migration will be executed.

Why: This document will dictate the commands and the work being executed. The plan controls the entire migration from start to finish. All roles, tasks, and responsibilities are defined.

Best practice: Create a cutover document

Create a cutover document that defines the high-level cutover tasks, responsibilities, and timing. The document should outline the phases and sequence in which tasks are executed.

Why: This document will outline the sequence of events that need to occur during a cutover. It can be used to track and monitor the progress of the cutover.

Best practice: Create a schedule and define outage windows

Have a well-defined cutover schedule and outage window.

Why: The schedule helps execute the migration cutover. The outage window is the time you have clearly designated in which access to storage will be unavailable and that you can make storage system changes without impact to clients.

Best practice: Create a communication plan

Have a communication plan.

Why: This communication plan will clearly outline protocols to keep all users up to date on the status of a migration, and allow storage administrators to stay focused on the execution of the cutover and not be distracted by requests for information from end users.

Best practice: Prepare the DNS name resolution infrastructure for cutover

Lower the DNS TTL.

Why: This will facilitate the cutover of clients using DNS name resolution.

Additional pre-cutover preparation steps often include the following:

- Prepare the DFS namespace
- Create CNAMES in DNS
- Update scripts used by clients for storage connections
- Prepare clients and applications

Cutover event

Having migrated the data and prepared the environment for cutover, the actual final migration event can occur.

In general, the high-level cutover sequence looks something like the following:

- Initiate migration cutover window—communicate the event
- Restrict access or make source data read-only—prevent new writes to the old data source
- Execute a final incremental—copy all final data to new storage
- Validate final incremental—validate that the source data is ready for cutover
- Final testing executed—the final cutover testing is completed
- Go or no go call on full cutover—decide if the migration should continue or roll back
- Update connection and name resolution protocols: DNS, CNAMES, DFS, scripts
- Flip new storage to read/write—enable writes to the new storage
- Continue testing and user acceptance—continue to test and monitor as production traffic moves over
- Execute the redirection of clients to the new storage—initiate client redirection process
- Monitor—assess the cutover, new storage, and clients

On executing a cutover event, the following best practices are recommended:

Best practice: Follow the cutover schedule

Follow a cutover schedule.

Why: By following a well-defined schedule, the migration can be monitored and controlled. It is suggested that you execute cutovers during off-hours or when the number of active connections is low.

Best practice: Test the migrated cutover data

Prepare a number of data and workflow tests to execute against the migrated data. Have a number of well-defined production use cases, data tests, and test users available to conduct post-cutover testing and review.

Why: Having a well-defined use case and users to validate the migration cutover will help in

the decision to continue with the cutover.

Best practice: Monitor clients and application during migrations

Monitor client and application connections to the new storage during cutover.

Why: To validate your cutover methodology is working as defined and clients are moving and connecting to the new storage.

Best practice: Develop a client connection remediation plan

Have a client remediation methodology plan in place and ready to execute against clients that exhibit any issues connecting to the new data targets.

Why: A well-defined strategy to handle client connection issues, including a dedicated support line, email address, or an IT desk, will make you aware of issues as they arise and help you resolve them more quickly.

The go or no-go decision

During the migration cutover window, a critical point will be reached. This threshold determines whether you continue with the cutover or abort the cutover and roll back to the existing storage.

Common abort cutover situations:

- Final incremental does not complete in the outage window
- Cutover methodology fails; client not connecting correctly
- Security issues with new storage system
- Workflow issues post-cutover
- Load and availability problems
- Other unknown issues

Best practice: Clearly define your cutover criteria

It is critical to have a series of cutover criteria that clearly define when a migration will continue or be aborted and rolled back.

Why: The criteria remove uncertainty, help with decision-making, and dictate the action to take.

Once you begin to write data to the new storage system, reverting to the old system becomes much more complicated, because you now need to reconcile data with the original storage system.

Rollback

If a decision to abort the cutover is taken, a well-defined rollback plan should have been developed and tested, so that you can restore data access as quickly as possible.

Rollback plan:

- Prevent any new writes to the new storage
- Move client connections back to the old storage
- Enable writes to the old storage

Best practice: Develop a rollback plan

Have a clearly defined rollback strategy that is easy to implement and can restore user access to data quickly and cleanly. Make sure the plan is tested.

Why: A rollback plan will quickly help you restore client data access in the event of a failure during a migration cutover event.

If any data has already been written to the new storage and a rollback is executed, then steps to remediate this data must be taken to restore this new data back to the original storage.

Common strategies for reconciling data during a rollback are as follows:

- Manually reconciled—identify and manually move any data from the new storage to the old storage
- A reverse incremental—have migration-type jobs run in a reverse direction to update the old storage
- Data just discarded—consider the data as non-critical and be comfortable with not reconciling it
- Data is rewritten from the client or application to the old storage—allow applications and clients to rewrite the data

The goal of any rollback strategy is to limit the impact to end users and restore data access as seamlessly as possible. It is for this reason that your migration cutover criteria should be well-defined and that the rollback strategy should have been thoroughly tested in the event that you have to use it.

Migration event complete

After you successfully complete a cutover, you should continue to monitor the new storage system.

Best practice: Monitor the new storage post-cutover

Continue to monitor the cutover storage after the cutover event for any new issue resulting from the cutover.

Why: Production load and workflow may be unpredictable; close monitoring should be undertaken to rectify any post-migration issues.

Steady state

Repetition

Most data migrations will constitute multiple cutover events. Having developed a well-structured migration methodology, these additional cutover events should be run with the same plan and strategy.

Lessons learned

After you complete a migration, you should assess the successes and failures of the methodology. If an additional migration needs to be performed, the lessons learned will allow you to refine the process.

Ask yourself the following questions:

- What worked during the migration and cutover?
- What did not work during the migration and cutover?
- Can the migration methodology be modified or optimized?

Conclusion

The goal of this document is to supply you with solid guidance for conducting an SMB single protocol file migration from a NAS system to an Isilon cluster. The guidance is based on a comprehensive set of industry lessons learned and best practices on the technical aspects and process of data migration. As stated in the beginning of this document, this paper does not aim to be an exhaustive authoritative source on the subject of SMB single protocol migrations, but rather a comprehensive reference document that covers the key areas to help ensure your success.

Appendix: Sample migration use case

This appendix provides a sample high-level overview of how to collect information for and plan a migration of SMB home directories. Keep in mind that the information in this section provides only a skeleton of some of the information that you would want to collect and an overview of the strategy that you would want to define for your migration. The use case that follows answers the following question at a high level: What are the recommendations and best practices as well as supported Isilon configurations to migrate home directories from a sample Celerra storage system to an Isilon cluster?

Source configuration and data	Sample directory structure
Single source system — Celerra NS80, 2DM Total data - 25TB Max file size – 4GB Min file size – 1GB Avg file size – 256KB File count: 2000000 5 top-level shares that users access User1 [users A-E] User2 [users F-K] User3 [users L-P] User4 [users Q-U] User5 [users V-Z] 3000 users' home directories; each user has a single home directory under a higher level share	\users \User1 \adamsj001 \... \... \adamss001 \agatek001 \.. \User2 \.. \..

Table 2: Source data

Share permissions	Windows ACLs
Domain Administrators – Full Control Domain Users – Modify	Users(X): Domain Administrators – Full Control, Inheritable Users\<username> Read/Write/Modify

Table 3: Shares and permissions

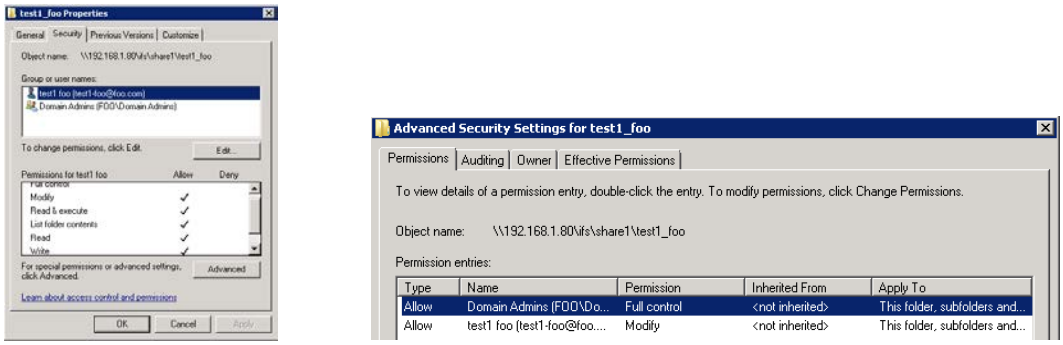


Figure 4: Windows ACLs

Additional source information	Additional Isilon information
All Active Directory – single domain	Isilon 3x400 – 100TB
Source system – 2x 1 Gb/s	OneFS 7.0.x
Each user has a defined mapped H:\	2x1gbs + 2x10gbs
No firewalls, IDS or QoS	SmartQuotas
Monthly Full –NDMP accelerators	SnapshotIQ
Antivirus scanning in place	
DART 5.6	
Same data center as Isilon	
No deduplication or offline files	
DNS – 5 CNAMEs, 1 per user share	
No routing or VLAN restrictions	

Table 4: Additional source data information

Requirements

- All data and permissions moved as is, no changes; move all existing ACLs
- Five cutover events; 12 hour window—Saturday 8:00PM through Sunday 8:00AM
- One migration per weekend
- Each user has a defined quota; quota limits to be replicated on Isilon

Migration project assumptions (including but not limited to)
• Customer will have approved change controls submitted for any migration activity • Migration plan and design will have been reviewed and approved by the customer prior to the start of cutovers • Any recommended array OS upgrades necessary for the migration will be applied before migration cutover activity • Source NAS and target Isilon systems are in a known good state prior to conducting migrations

Strategy

- Host-based migration; dedicated VM-Windows Server 2008r2
- Conduct pilot migration: validate methodology, document performance metrics, refine and tune EMCOPY switches
- Develop backout plan, review with customer
- Execute five migration phases, one per user share
- Use a DNS update methodology to redirect clients
- Execute initial full copies, followed by nightly incrementals
- During cutover; make source file system =R/O
- Reduce DNS TTLs in advance of cutover windows
- Develop detailed project timeline and cutover schedule with customer
- Validate change rates and time to execute incrementals
- Develop client communication; customer should provide dedicated cutover IT support desk/personnel
- Complete a full backup of all file systems that are to be migrated before cutover

Source system configuration

- Create dedicated **userSourceMigration$** administrative share, Domain Administrator
- Migration will be executed in context of Domain Administrator, they have full access to all data

Isilon configuration

- Create dedicated **userMigration$** admin share, Domain Administrator—run-as-root
- Pre-create SMB shares with identical share permissions
- Disable snapshots on data until cutover complete
- Disable AV scanning

Toolset selection

- use EMCOPY64

```
emcopy64.exe S:\users T:\users /ignoredhsm /o /sd /s /d /th 64 /c
/purge /r:1 /w:2  /log:<path_to_log.txt>
```

Migration testing

- Map source and target drives from the migration host
- Use a small set of test users to validate full data access (data to be discarded following test)

UAT

- Verify that data and permissions on Isilon are replicated following copies
- Review user access
- Use a small set of test users to validate full data access during migration event
- Verify that a user can read/write to a file, create a new file
- Monitor Isilon performance and client connections

Cutover plan

- Make source file systems read-only
- Execute final incremental copies
- Update DNS
- Initiate user logoff and re-logon

Rollback plan

- Reverse DNS update
- Make old source file systems read-write
- Remove any connections to Isilon
- Any data that was written to Isilon is considered lost, no reverse or reconciliation will be done

Exit criteria

- DNS resolves to new target storage and shares
- Clients can successfully read/write data to user home directories
- No connection issues

Post-cutover

- Customer meeting to review and triage the migration
- Documentation of migration protocol
- Lessons learned discussion with both the internal and customer team