# Dell PowerStore: File Capabilities

July 2022

H18155.5

White Paper


## Abstract

This document discusses the features, functionality, and protocols supported by the Dell PowerStore file architecture.



Dell Technologies

## Copyright


Published in the USA July 2022 H18155.5.

# Contents

# Executive summary

## Overview

Dell PowerStore offers a native file solution that is designed for the modern data center. The file system architecture is highly scalable, efficient, performance-focused, and flexible. PowerStore also includes a rich supporting feature set, enabling the ability to support a wide array of use cases such as departmental shares or home directories. These file capabilities are integrated, so no extra hardware, software, or licenses are required. File management, monitoring, and provisioning capabilities are handled through the simple and intuitive HTML5-based PowerStore Manager.

## Audience

This document is intended for IT administrators, storage architects, partners, and Dell Technologies employees. This audience also includes any individuals who may evaluate, acquire, manage, operate, or design a Dell networked storage environment using PowerStore systems.

## Revisions

| Date | Description |
|---|---|
| April 2020 | Initial release: PowerStoreOS 1.0 |
| July 2020 | Minor updates |
| September 2020 | PowerStoreOS 1.0.2 updates |
| April 2021 | PowerStoreOS 2.0 updates |
| November 2021 | Minor updates; template update |
| July 2022 | PowerStoreOS 3.0 updates |

## We value your feedback

Dell Technologies and the authors of this document welcome your feedback on this document. Contact the Dell Technologies team by email.

**Author:** Wei Chen

**Note**: For links to other documentation for this topic, see the PowerStore Info Hub.

# Introduction

## PowerStore overview

PowerStore achieves new levels of operational simplicity and agility. It uses a container-based microservices architecture, advanced storage technologies, and integrated machine learning to unlock the power of your data. PowerStore is a versatile platform with a performance-centric design that delivers multidimensional scale, always-on data reduction, and support for next-generation media.

PowerStore brings the simplicity of public cloud to on-premises infrastructure, streamlining operations with an integrated machine-learning engine and seamless automation. It also offers predictive analytics to easily monitor, analyze, and troubleshoot the environment. PowerStore is highly adaptable, providing the flexibility to host specialized workloads

directly on the appliance and modernize infrastructure without disruption. It also offers investment protection through flexible payment solutions and data-in-place upgrades.

## PowerStore file capabilities

PowerStore features a native file solution that is highly scalable, efficient, performance-focused, and flexible. This design enables accessing data over file protocols such as Server Message Block (SMB), Network File System (NFS), File Transfer Protocol (FTP), and SSH File Transfer Protocol (SFTP).

PowerStore uses virtualized NAS servers to enable access to file systems, provide data separation, and act as the basis for multitenancy. File systems can be accessed through a wide range of protocols and can take advantage of advanced protocol features. Services such as anti-virus, scheduled snapshots, Network Data Management Protocol (NDMP) backups, and replication ensure the data on the file systems is well protected.

PowerStore file is available natively on PowerStore T model appliances, which are designed as true unified storage systems. There are no extra pieces of software, hardware, or licenses required to enable this functionality. All file management, monitoring, and provisioning capabilities are available in the HTML5-based PowerStore Manager.

## Terminology

The following table provides definitions for some of the terms that are used in this document.

**Table 1. Terminology**

| Term | Definition |
|---|---|
| File system | A storage resource that can be accessed through file sharing protocols such as SMB or NFS. |
| Network-Attached Storage (NAS) server | A virtualized network-attached storage server that uses the SMB, NFS, FTP, and SFTP protocols to catalog, organize, and transfer files within file system shares and exports. A NAS server, the basis for multitenancy, must be created before you can create file-level storage resources. NAS servers are responsible for the configuration parameters on the set of file systems that it serves. |
| Network File System (NFS) | An access protocol that enables users to access files and folders on a network. NFS is typically used by Linux/UNIX hosts. |
| PowerStore Manager | An HTML5 user interface used to manage PowerStore systems. |
| Server Message Block (SMB) | An access protocol that allows remote file data access from clients to hosts on a network. SMB is typically used in Microsoft Windows environments. |
| Snapshot | A point-in-time view of data stored on a storage resource. A user can recover files from a snapshot or restore a storage resource from a snapshot. |

# Unified appliance

When running through the Initial Configuration Wizard (ICW) on a PowerStore T model appliance, you can choose to configure it either as a unified or block-optimized appliance. Selecting Unified enables file and block functionality while selecting Block-Optimized only enables block functionality. This selection determines the resource allocation on the appliance. This selection can only be made during initial configuration and cannot be changed without reinitializing. This selection is not displayed on PowerStore X model appliances since they do not support file.

To enable file functionality on the appliance, select **Unified**. If there is a chance that file may be required, it is recommended to choose this option. If a unified configuration is selected, the NAS installation is started automatically after the cluster creation successfully completes.

In order to complete the NAS installation process, a communication channel between the two nodes is required. Starting with PowerStoreOS 1.0.2, this communication is done through an internal backplane interconnect. Previously, this communication traveled through the first two ports of the 4-port card from one node, through the top-of-rack switches, and through the first two ports of the 4-port card on the second node. If the switches are not properly configured, it results in a NAS installation failure. With PowerStoreOS 1.0.2, the NAS installation completes successfully even if the switches are not configured properly. This behavior allows the administrator to bring the system online and address the switch configuration later.

The PowerStore 500 can be ordered without a 4-port card. However, in order to support file functionality, the 4-port card must be ordered. On PowerStore 500 systems that do not have a 4-port card installed, the Unified option is disabled and the only option available is Block Optimized. In addition to not supporting file, PowerStore 500 systems without a 4-port card also do not support clustering.

If file functionality is not required, the Block Optimized selection provides slightly higher block IOPS potential. The following figure shows the storage configuration options on a PowerStore T model appliance.

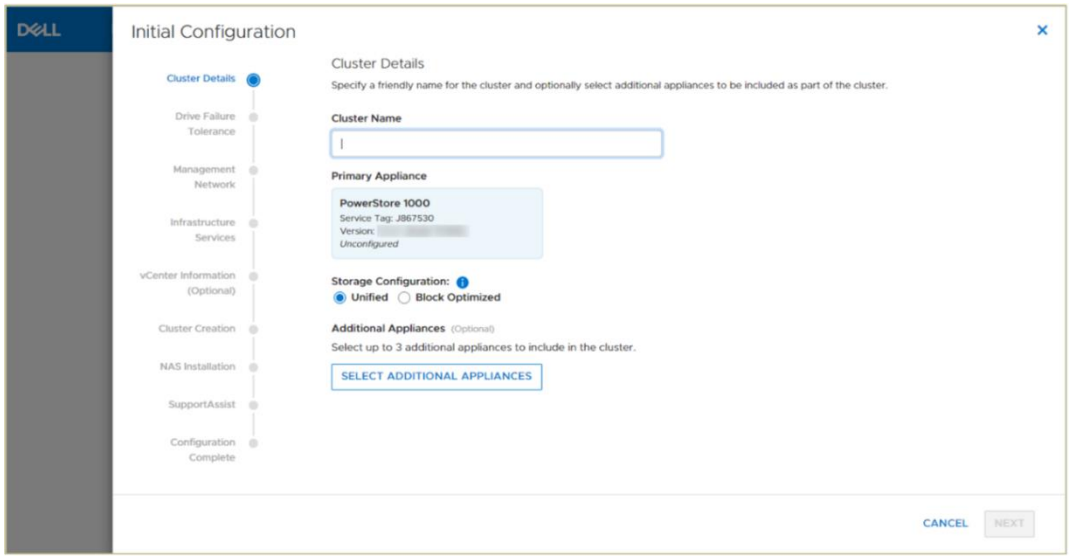


**Figure 1. Storage configuration**

After submitting the required data in the ICW, the cluster creation and file installation process begins. Starting with PowerStoreOS 3.0, these operations are split into two separate phases. The first is the core services initialization phase which includes all core services required to make the cluster operational. The second phase consists of non-core and optional services, such as file. If any errors are detected during the non-core services phase, this behavior enables troubleshooting and addressing the issue while maintaining the healthy state of the cluster. The create cluster and file services initialization step of the ICW is shown in Figure 2.

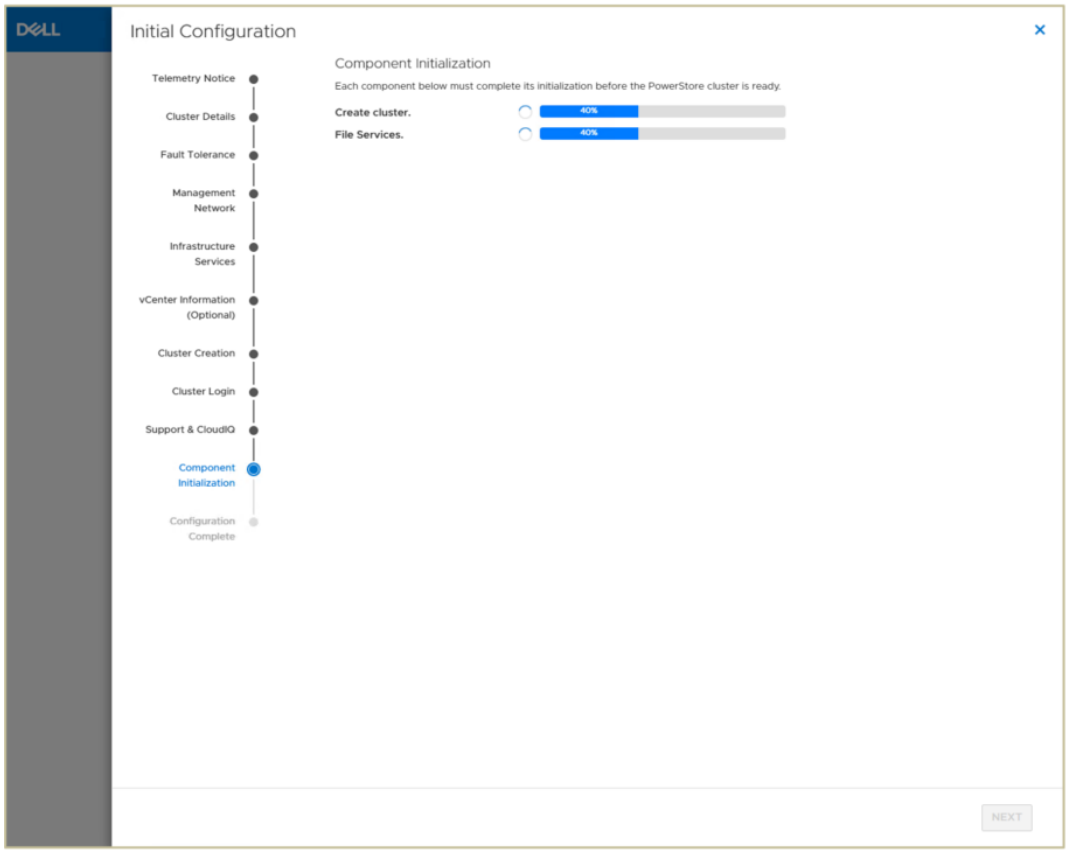


**Figure 2. Create cluster and file services initialization**

In a multi-appliance cluster configuration, file functionality is only available on the primary or first appliance in the cluster. The remaining appliances in the cluster are configured as Block-Optimized, and only the capacity on the first appliance can be used for file systems. The capacity on the other appliances within the same cluster can be used for volumes and VMware vSphere Virtual Volumes (vVols), but not for file systems.

Within the first appliance, both nodes are used for file. This configuration creates a fully redundant and active/active architecture where both nodes are used to serve file data. This design enables the ability to load balance across both nodes and ensure high availability in a failover.

# NAS servers

## Introduction

PowerStore file uses virtualized file servers that are called NAS servers. A NAS server contains the configuration, interfaces, and environmental information that is used to facilitate access to the file systems. This includes services such as Domain Name System (DNS), Lightweight Directory Access Protocol (LDAP), Network Information Service (NIS), protocols, anti-virus, NDMP, and so on.

## Multitenancy

NAS servers can be used to enforce multitenancy. This is useful when hosting multiple tenants on a single system, such as for service providers. Since each NAS server has its own independent configuration, it can be tailored to the requirements of each tenant

without impacting the other NAS servers on the same appliance. Also, each NAS server is logically separated from each other, and clients that have access to one NAS server do not inherently have access to the file systems on the other NAS servers. File systems are assigned to a NAS server upon creation and cannot be moved between NAS servers.

## High availability

New NAS servers are automatically assigned on a round-robin basis across the available nodes. The preferred node acts as a marker to indicate the node that the NAS server should be running on, based on this algorithm. Once provisioned, the preferred node for a NAS server never changes. The current node indicates the node that the NAS server is running on. Changing the current node moves the NAS server to a different node, which can be used for load-balancing purposes. When a NAS server is moved to a new node, all file systems on the NAS server are moved along with it. The following figure shows the current and preferred node columns for a NAS server.

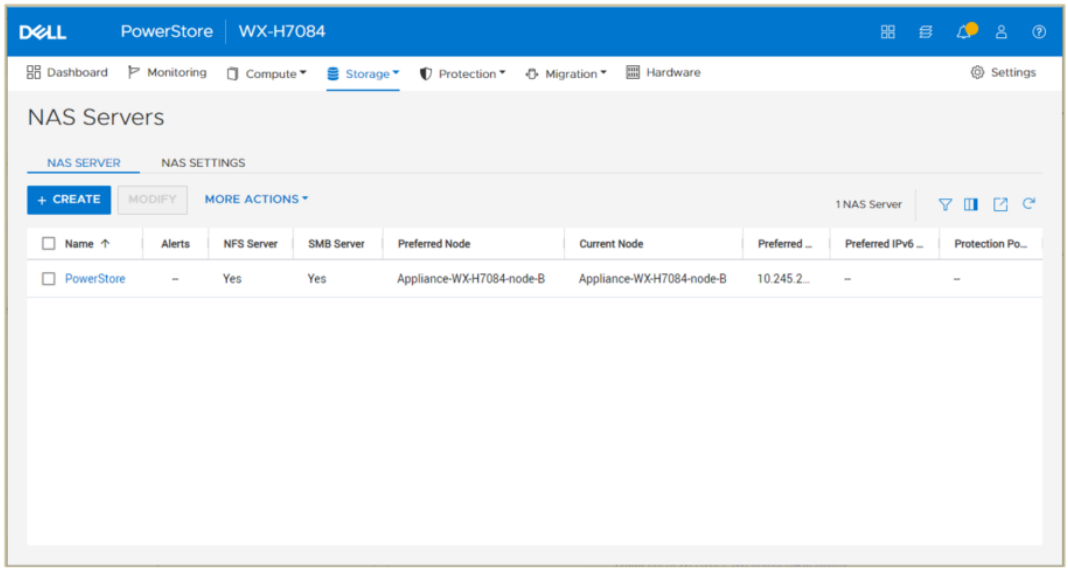


**Figure 3. Current and preferred node**

In a PowerStore node failure, the NAS servers automatically fail over to the surviving node. This process generally completes within 30s to avoid host timeouts. Once the failed node is recovered, failing back the NAS servers to return to a balanced configuration is a manual process.

NAS servers are also automatically moved to the peer node and back during the upgrade process. After the upgrade is complete, the NAS servers return to the node they were assigned to at the beginning of the upgrade.

## Interfaces

Each NAS server supports up to 50 production and 10 backup interfaces. Production interfaces are used for client connectivity over FTP, SFTP, NFS, and SMB. Backup interfaces support NFS-only access, which can be used for backup purposes.

Ping from one of the NAS server interfaces can be used for troubleshooting purposes. The system designates one interface as the preferred interface, which is used for outgoing communication to external services. Also, custom host and network routes can be configured on a per-interface basis.

## Link Aggregation

The system configures all NAS server interfaces on the first two bonded ports on the four-port card, by default. Since both bonded ports are used for file traffic, link loss on a single port does not impact connectivity to the NAS server. For maximum bandwidth on these ports, it is recommended to configure Virtual Link Trunking interconnect (VLTi) with Link Aggregation Control Protocol (LACP) or equivalent technology on the switch. The embedded module that holds the four-port card is shown in the following figure.

**Figure 4. Embedded module with four-port card**

Starting with PowerStoreOS 3.0, user-defined link aggregations can be configured for file interfaces. This allows you to create custom bonds on two to four ports. The bonds can span the 4-port card and IO modules, but they must have the same speed, duplex, and MTU settings.

To configure a bond, the switch must be configured properly. The ports that will be part of the bond on node A should be configured into a port channel. The same configuration should be mirrored for the ports on node B. If the switch is not in a consistent configuration with the storage system, PowerStore Manager displays an alert.

Afterwards, in PowerStore Manager, navigate to **Hardware → Ports**. Select the ports on either node A or node B that should be part of the bond, then click **Link Aggregation → Aggregate Link,** as shown in Figure 5**.**

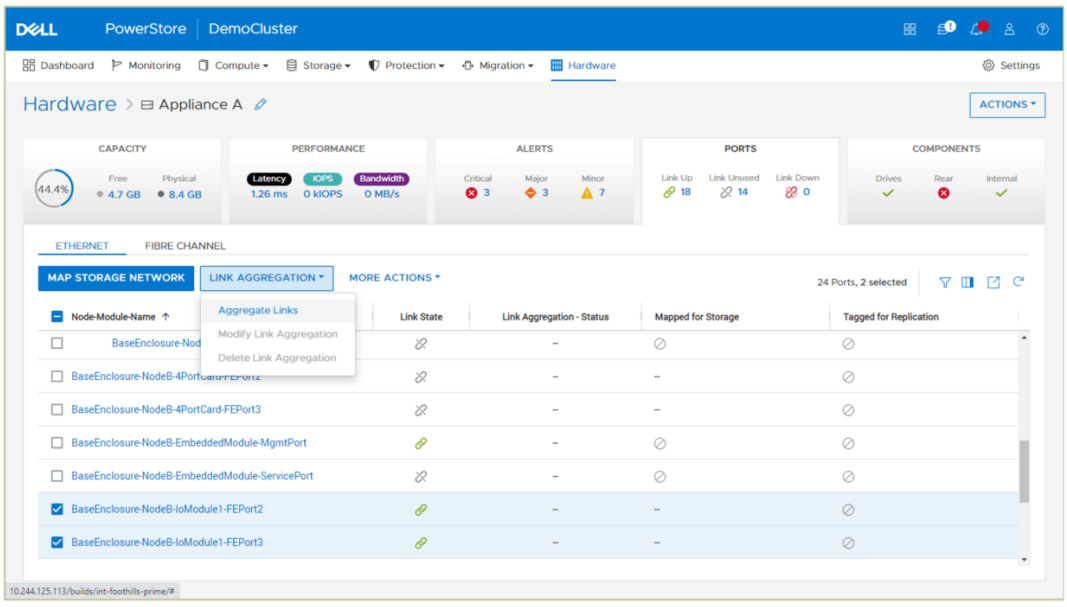


**Figure 5. Aggregate links**

Regardless of the node from which the ports were selected, the bond is created on both nodes automatically. A name for the bond is generated automatically and cannot be changed. An optional description can be added to the bond and can be changed at any time. The aggregate that was created is shown in Figure 6.

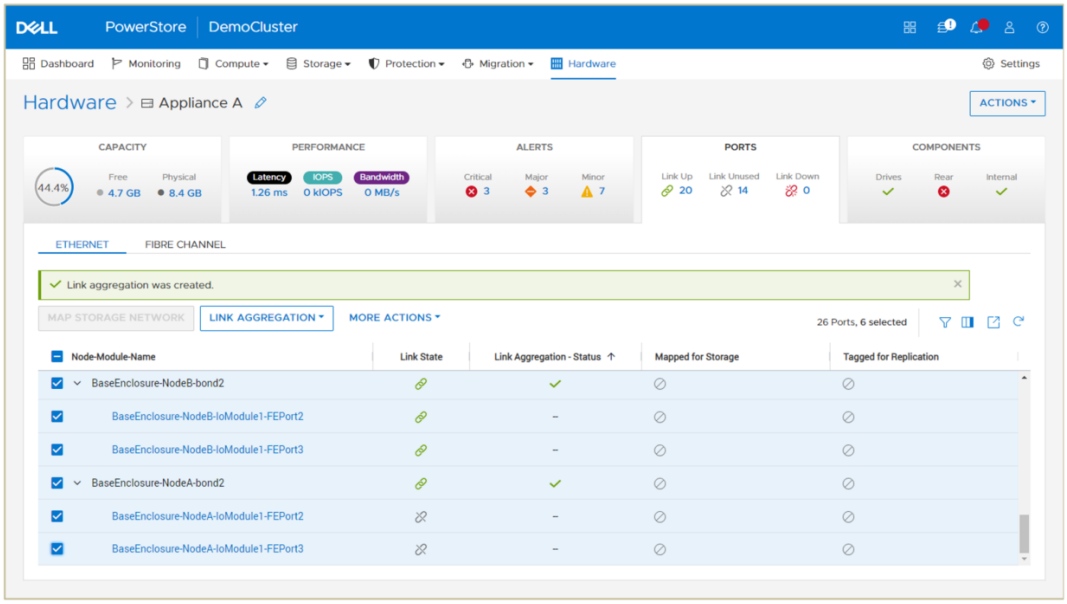


**Figure 6. Aggregated bond2 on both nodes**

Existing bonds can be expanded to add more members to it. To do this, select the bond along with additional individual ports and click **Aggregate Link**. Bonds can also be deleted if they are no longer needed.

After the link aggregation is created, it can be used to create a file interface on a NAS server, as shown in Figure 7.

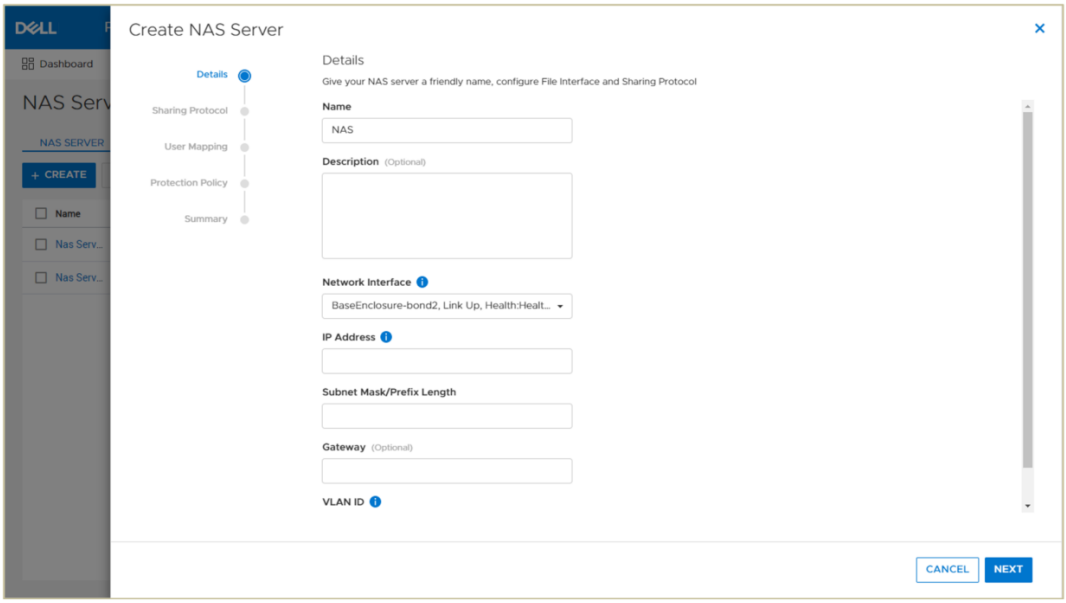


**Figure 7. Selecting a bond for a NAS server interface**

### VLAN

Each NAS server interface can be configured on a specific VLAN. Interfaces that reside on different VLANs can be created, even if they are within the same NAS server. Each IP address must be unique, even if they reside on different VLANs. You cannot create a NAS server interface that resides on the same VLAN as the storage VLAN, which is used for iSCSI and replication connectivity.

### NAS server parameters

NAS server parameters are used for controlling the behavior and advanced tuning of file features. Management of parameters is only available in the CLI by the service user using the svc_nas_tools command. All parameter changes are preserved through node reboots and failovers.

When viewing the details or changing a parameter, view the output to see more information about the parameter including the granularity when it takes effect. Some parameters are applied at a NAS server granularity while others are global. The global parameters are applied at the system level by specifying **ALL** instead of a NAS server name. Also, some parameters require a reboot to take effect.

The following figure shows an example of a NAS server parameter.

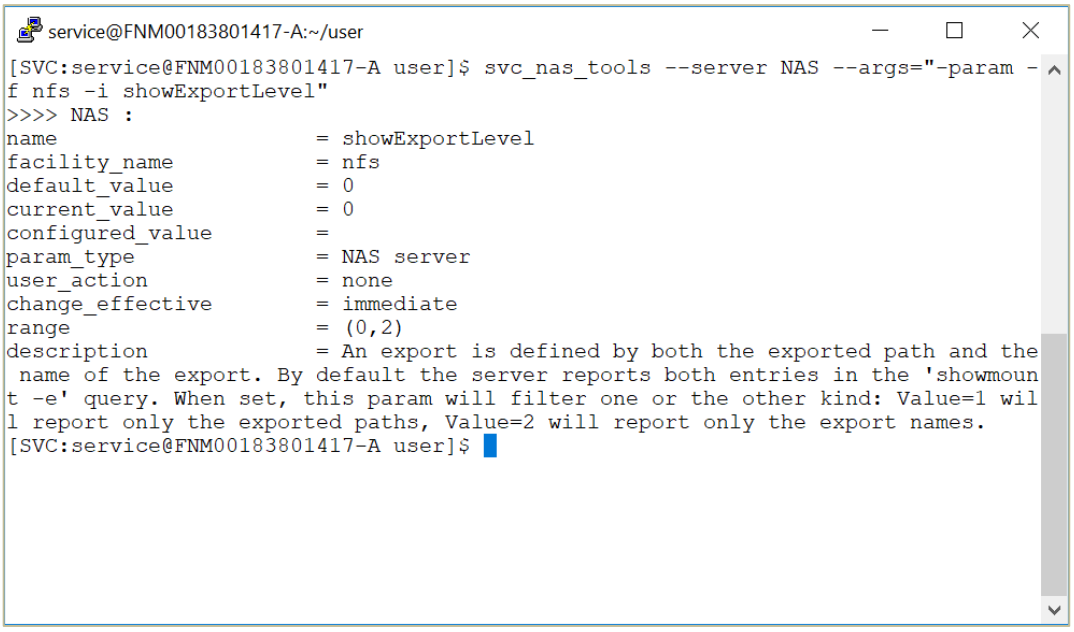


**Figure 8. NAS server parameters**

For more information about the available NAS server parameters and how to configure them, see the document *Dell PowerStore Service Scripts Guide* on Dell.com/powerstoredocs.

# Protocols

### Introduction

PowerStore file supports a wide range of protocols including SMB, NFS, FTP, and SFTP. Since these protocols are enabled at the NAS server level, each NAS server can be customized to allow only the specific protocols that are being used. Each NAS server can

be configured to support one or more protocols and these options can be changed at any time.

## SMB

PowerStore file supports SMB1 through 3.1.1. SMB3 enhancements such as continuous availability, offload copy, protocol encryption, multichannel, and shared VHDX in Hyper-V are supported on PowerStore.

The SMB option on the NAS server enables or disables SMB connectivity to the file systems. The SMB version that is negotiated depends on the client operating system:

- CIFS: Windows NT 4.0
- SMB1: Windows 2000, Windows XP, Windows Server 2003, and Windows Server 2003 R2
- SMB2: Windows Vista (SP1 or later) and Windows Server 2008
- SMB2.1: Windows 7 and Windows Server 2008 R2
- SMB3.0: Windows 8 and Windows Server 2012
- SMB3.02: Windows 8.1 and Windows Server 2012 R2
- SMB3.1.1: Windows 10 and Windows Server 2016 and Windows Server 2019

Due to the age of the protocol and potential security vulnerabilities, client access using SMB1 is disabled by default. If client access using SMB1 is required, it can be enabled by modifying the **cifs.smb1.disabled** parameter. Using SMB2 at a minimum is recommended as it provides security enhancements and increases efficiency compared to SMB1.

NAS servers use SMB2 to communicate with the domain controllers for operations such as authentication, SID lookups, Group Policies, and so on. If SMB2 is not available, the NAS server attempts to use SMB1 as a backup option. This means that any domain controllers that are running older operating systems that only support SMB1 can continue to function.

When enabling SMB support on a NAS server, the SMB server can either be stand-alone or Active Directory (AD) domain-joined. Domain-joined NAS servers require DNS to be configured, but this configuration is optional for stand-alone SMB servers. Domain-joined NAS servers are placed in the **CN=Computers** container, by default. When joining an SMB server to the domain, the computer object can be configured to be stored in a different OU location in the advanced settings.

Support for advanced SMB protocol options is also available. Table 2 shows a list of SMB protocol options, where they are configured, and the default setting for each option.

**Table 2. SMB options**

| Protocol options | Level | Default |
|---|---|---|
| Sync Writes Enabled | File system | Disabled |
| Oplocks Enabled | File system | Enabled |
| Notify on Write Enabled | File system | Disabled |

| Protocol options | Level | Default |
|---|---|---|
| Notify on Access Enabled | File system | Disabled |
| Continuous Availability | Share | Disabled |
| Protocol Encryption | Share | Disabled |
| Access-Based Enumeration | Share | Disabled |
| Branch Cache Enabled | Share | Disabled |
| Offline Availability | Share | None |
| UMASK (Multiprotocol) | Share | 022 |

## Sync writes

Synchronous writes enable the storage system to perform immediate synchronous writes for storage operations, regardless of how the SMB protocol performs write operations. Enabling synchronous writes operations allow you to store and access database files (for example, MySQL) on storage system SMB shares. This option guarantees that any write to the share is done synchronously and reduces the chances of data loss or file corruption in various failure scenarios, for example, loss of power. If SMB3 Continuous Availability (CA) is enabled, all write operations are automatically synced to satisfy the requirements for CA. This option can have a big impact on performance. It is not recommended unless you intend to use Windows file systems to provide storage for database applications.

## Oplocks

Opportunistic file locks (oplocks) allow SMB clients to buffer file data locally before sending it to a server. SMB clients can then work with files locally and periodically communicate changes to the storage system rather than having to communicate every operation over the network to the storage system. Unless your application handles critical data or has specific requirements that make this mode or operation unfeasible, leaving the oplocks enabled is recommended.

The following oplocks implementations are supported:

- **Level II Oplocks**: This informs a client that multiple clients are currently accessing a file, but no client has yet modified it. A level II oplock lets the client perform read operations and file-attribute fetches by using cached or read-ahead local information. All other file access requests must be sent to the server.
- **Exclusive Oplocks (SMB2 only)**: This informs a client that it is the only client opening the file. An exclusive oplock lets a client perform all file operations by using cached or read-ahead information until it closes the file, at which time the server must be updated with any changes that are made to the state of the file (contents and attributes).
- **Batch Oplocks**: This informs a client that it is the only client opening the file. A batch oplock lets a client perform all file operations by using cached or read-ahead information (including opens and closes). The server can keep a file opened for a client even though the local process on the client machine has closed the file. This mechanism curtails the amount of network traffic by letting clients skip the extraneous close and open requests.

This option only applies to client access over SMB1 since oplocks are always enabled for client access over SMB2 and SMB3. However, disabling this option also invalidates the SMB2.1 file and directory lease feature. Leasing serves the same purpose as oplocks, but provides greater flexibility and enhancements, increasing performance and reducing network utilization.

- **Read-caching lease**: This allows caching reads and can be shared by multiple clients.
- **Write-caching lease**: This allows caching writes and is exclusive to only one client.
- **Handle-caching lease**: This allows caching handles and can be shared by multiple clients.

### Notify on write or access enabled

This option enables notifications when a file system is written to or accessed. Applications that run on Windows platforms, and use the Win32 API, can register with the SMB server to be notified of file and directory content changes, such as file creation, modify, or rename. For example, this feature can indicate when a display must be refreshed (Windows Explorer) or when the cache must be refreshed (Microsoft Internet Information Server), without having to constantly poll the SMB server.

### Continuous availability

Continuous availability is a share-level SMB3 feature. In a client or storage processor failure, CA allows persistent access to file systems without loss of the session state. This ability is useful for critical applications such as Microsoft Hyper-V or SQL, where constant availability to files is of the upmost importance. SMB3 uses persistent handles to enable the NAS server to save specific metadata that is associated to an open handle on disk. In a node failure, applications accessing open file content are not affected if the NAS server and file system failover to the peer node completes within the timeout of the application. This action results in clients transparently reconnecting to the peer node after the NAS server failover without affecting client access to files.

Continuous availability is also available on the client side, which is independent from storage CA. Client CA transparently preserves access in a node failure within a client application cluster. When a failure of one node in the cluster occurs, the application is moved to the other node and reopens its content on the share from that node using its originally assigned ApplicationID without an interruption in access. The CA option on the share does not need to be enabled in order to use client CA.

SMB 3.1.1 adds a reliability enhancement for Continuous Availability for Hyper-V Cluster Client Failover by adding an ApplicationInstanceVersion tag in addition to the ApplicationID. The ApplicationInstanceVersion tag is incremented each time that an application is restarted on a new node within the cluster. In situations where network access is lost, but storage access remains available, the application may be restarted on a new node without the cluster knowing due to the lack of network access. The ApplicationInstanceVersion tag enables the storage system to easily identify which node in the cluster is the correct owner of the application. The storage system can safely close any locks that were opened with a lower ApplicationInstanceVersion number, which allows the application to restart without any conflicts.

## Protocol encryption

Protocol encryption is a share-level SMB3 feature, which provides in-flight data encryption between SMB3 clients and the NAS server. The client or NAS server encrypts the data before sending it to the destination. It is then decrypted upon reaching its destination, whether that is the NAS server or SMB client. The protocol encryption is enforced at user session level, ensuring the whole SMB traffic is encrypted once the user session is established.

The following setting can be configured in the NAS server registry:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanServer\Parameters\RejectUnencryptedAccess: Determines if clients that do not support encryption (pre-SMB3.0) have access to the share.

- 1 (default): Returns access denied to pre-SMB3.0 clients that do not support encryption
- 0: Allows pre-SMB3.0 clients to access the share without encryption

SMB 3.1.1 also provides improved security and encryption traffic performance for SMB3 by changing the encryption algorithm from AES-CCM-128 to AES-GCM-128. This change improves performance under certain conditions such as large file transfers. In addition, this improves security against man-in-the-middle attacks.

## Access-based enumeration

Access-based enumeration is a share-level option that restricts the display of files and folders based on the access privileges of the user attempting to view them. Without access-based enumeration, all users can view all files and folders within a directory. However, they cannot open or view these files and folders without the appropriate access privileges. When access-based enumeration is enabled on a share, users are only able to see files or folders for which they have at read access or above.

For example, without access-based enumeration, a user could see all files in a directory, regardless of whether they can open them. However, with access-based enumeration, the inaccessible files are hidden from the user view. Administrator users are always able to see all files and folders, even when access-based enumeration is enabled on a share.

## BranchCache

BranchCache is a share-level option that allows users to access data that is stored on a remote NAS server locally over the LAN without being required to traverse the WAN to access the NAS server. This ability is useful in a remote or branch office environment, where branch offices are required to access data stored on PowerStore at the main office. BranchCache allows this data to be cached locally at the branch, either by a designated Windows BranchCache server or distributed across Windows clients. This ability can reduce WAN bandwidth that is used by many clients constantly and repeatedly traversing the WAN for the same data.

With BranchCache enabled, the client uses the WAN to retrieve the hash of the file from the NAS server at the main office. The client searches the local file cache to look for a file with a matching hash. If all or some of the data is available locally, either on the designated Windows BranchCache server or another Windows client system, the data is retrieved locally. The data is validated using a hash function to ensure that the file is the

same. Any data that is not cached locally is retrieved from the NAS server over the WAN, and then cached locally for future requests. BranchCache works best for data that does not change often, allowing files to be cached for longer periods of time at the branch offices.

### Offline availability

Offline availability is a share-level option that allows administrators to determine if and how files and programs in a share are available when offline. This ability allows users to access shares on a server even when they are not connected to the network by storing a version of the share in a local cache on the client system. For offline availability to function, it must be configured on both the share and the individual client systems accessing the share.

SMB shares support four options for offline availability:

- **None (Default)**: No files or programs from the share are available offline. Client systems cannot cache any content from this share for offline access.
- **Manual**: Only files and programs that the users specify are available offline. Nothing is cached without the user requesting it.
- **Programs**: All files and programs that users open from the share are automatically available offline. However, executable files that have been previously cached locally are run from the cached copy rather than the copy on the share, even when the share is available. This option is useful for reducing network traffic and performance overhead.
- **Documents**: All files and programs that users open from the share are automatically available offline. Whenever a user accesses a file or program from a share, that content is automatically cached to be available to that user in offline mode. All files that are opened continue to be cached and available for offline access until the cache becomes full or the user deletes files from the cache. Cached content continues to sync with the version on the server. Files and programs that have not been opened are not available offline.

### DFS

PowerStore also supports the Microsoft Distributed File System (DFS) namespace. This ability enables the administrator to present shares from multiple file systems through a single mapped share. PowerStore SMB servers can be configured as a stand-alone DFS root node or as a leaf node on an Active Directory DFS root. DFS-R (replication) is not supported on PowerStore SMB servers.

### File extension filtering

File extension filtering enables restricting specific file extensions from being stored on an SMB share. This feature is useful for home directories to prevent users from storing unnecessary data on their share, such as .mp3 files. This feature only works on SMB and does not filter file extensions when writing over NFS. Users can also rename file extensions to bypass this filter, such as changing from .mp3 to .m3p, unless these are also explicitly blocked. When users attempt to copy a file with a blocked extension, they receive an access denied error, as shown in the following figure.

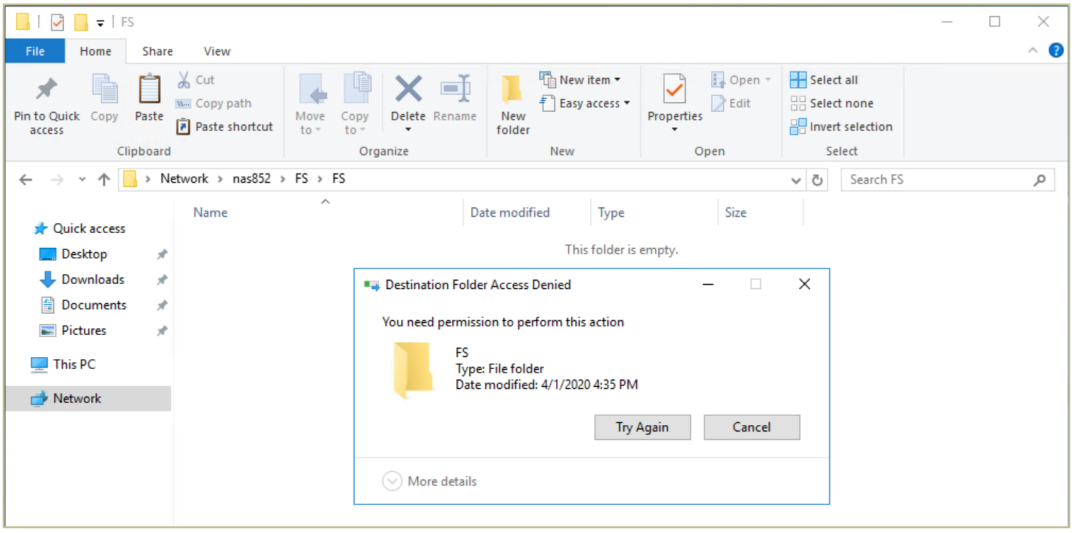


**Figure 9. Access denied due to filtered file**

To configure file extension filtering, go to the **\\<SMB_Server>\c$\.etc\.filefilter** directory as an administrator. To configure a filter, create an empty file using the naming convention **extension@sharename**. For example, to filter .mp3 files on the FS1 share, create a file named **mp3@FS1**. To enable the filter on all shares on the SMB server, create the file with only the extension such as **mp3**. You can configure multiple filters by creating additional files. Each SMB server has its own independent file extension filtering configuration.

After configuring a file extension filter, you can permit exceptions for specific users or groups. This action is done by changing the ACL on the filter file to provide Full Control privileges to the users or groups that should be excluded. For example, if the Music group is provided Full Control permissions on the **mp3** filter file, then users in the Music group can store .mp3 files on the share, while others cannot. Exceptions can be configured independently for each file filter being created.

The following figure shows an example configuration of the file extension filtering.

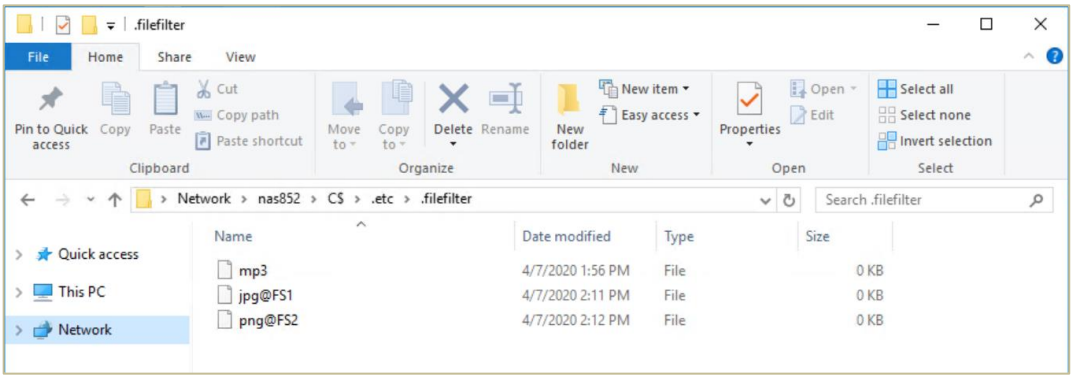


**Figure 10. File extension filtering configuration**

## NFS

PowerStore supports NFSv3 through NFSv4.1 and as Secure NFS. Each NAS server has options to enable NFSv3 and NFSv4 independently. Support for advanced NFS protocol

options is also available. Table 3 shows a list of NFS protocol options, where they are configured, and the default setting for each option.

**Table 3. NFS options**

| Protocol options | Level | Default |
|---|---|---|
| Secure NFS (with Kerberos) | NAS server | Disabled |
| Minimum Security | Share | Sys |
| Default Host Access | Share | No Access |

## NFSv4

NFSv4 is a version of the NFS protocol that differs considerably from previous implementations. Unlike NFSv3, this version is a stateful protocol, meaning that it maintains a session state and does not treat each request as an independent transaction without the need for additional preexisting information. This behavior is like Windows environments with SMB. NFSv4 brings support for several new features including NFS ACLs that expand on the existing mode-bit-based access control in previous versions of the protocol.

While PowerStore fully supports most of the NFSv4 and v4.1 functionality described in the relevant RFCs, directory delegation and parallel NFS are not supported.

To configure NFSv4, you must first enable NFSv4 on the NAS server, create a file system, and an NFS export. Then, the file system can be mounted on the host using the NFSv4 mount option.

## Secure NFS

Traditionally, NFS is not the most secure protocol, because it trusts the client to authenticate users and build user credentials and send these in clear text over the network. With the introduction of secure NFS, Kerberos can be used to secure data transmissions through user authentication and data signing through encryption. Kerberos is a well-known, strong authentication protocol where a single key distribution center, or KDC, is trusted rather than each individual client. There are three different modes available on PowerStore:

- **Kerberos**: Use Kerberos for authentication only
- **Kerberos With Integrity**: Use Kerberos for authentication and include a hash to ensure data integrity
- **Kerberos With Encryption**: Use Kerberos for authentication, include a hash, and encrypt the data in-flight

To enable secure NFS, the following must be configured:

- DNS must be configured on the NAS server.
- A UNIX Directory Service (UDS) such as NIS, LDAP, or Local Files must be enabled.
- A Kerberos realm must exist.

If an Active Directory domain joined SMB server existed on the NAS server, that Kerberos realm may be leveraged. Otherwise, a custom realm can be configured for use in PowerStore Manager. LDAP over SSL (LDAPS) is used for Secure NFS to avoid weaknesses in the security chain. Although NFSv3 is supported with Secure NFS, it is preferable to use NFSv4 to maximize security.

### Minimum security

The minimum security setting determines the type of security that is enforced on the NFS export. The default setting of Sys uses client-provided UNIX UIDs and GIDs for NFS authentication. If Secure NFS is enabled on the NAS server, the Kerberos options become available. See above for more information about the Secure NFS with Kerberos options.

### Default host access

The default host access option determines the access permissions for all hosts that attempt to connect to the NFS export. The available options are:

- No Access (Default)
- Read/Write
- Read-Only
- Read/Write, allow Root
- Read-Only, allow Root

The allow root options are the equivalent to **no_root_squash** on UNIX systems. This means if the user has root access on the client, they are also granted root access to the NFS export.

For hosts that need different access than the default, they can be configured by adding hostnames, IP addresses, or subnets to the override list with one of the access options above. Multiple entries can also be added simultaneously in a comma-separated format. Table 4 shows the supported options when configuring NFS host access.

**Table 4. NFS host access**

| Name | Example | Notes |
|---|---|---|
| Hostname | host1.dell.com | Hostname should be defined in the local hosts file, NIS, LDAP, or DNS. |
| IPv4 or IPv6 Address | 10.10.10.10<br>fd00:c6:a8:1::1 | |
| Subnet | 10.10.10.0/255.255.255.0<br>10.10.10.0/24 | IP address/netmask or IP address/prefix |

Host access can also be configured by uploading a CSV file with a list of hosts and their respective access levels. PowerStore Manager provides a template with examples on the formatting and syntax for this file. This template can be downloaded from the system, edited, and then imported. When multiple NFS exports that require the same access configuration are configured, the same file can be imported multiple times and across multiple clusters as well. Once the file is imported, the newly imported hosts are

appended to the access list. The following figure shows the host access configuration on an NFS export.

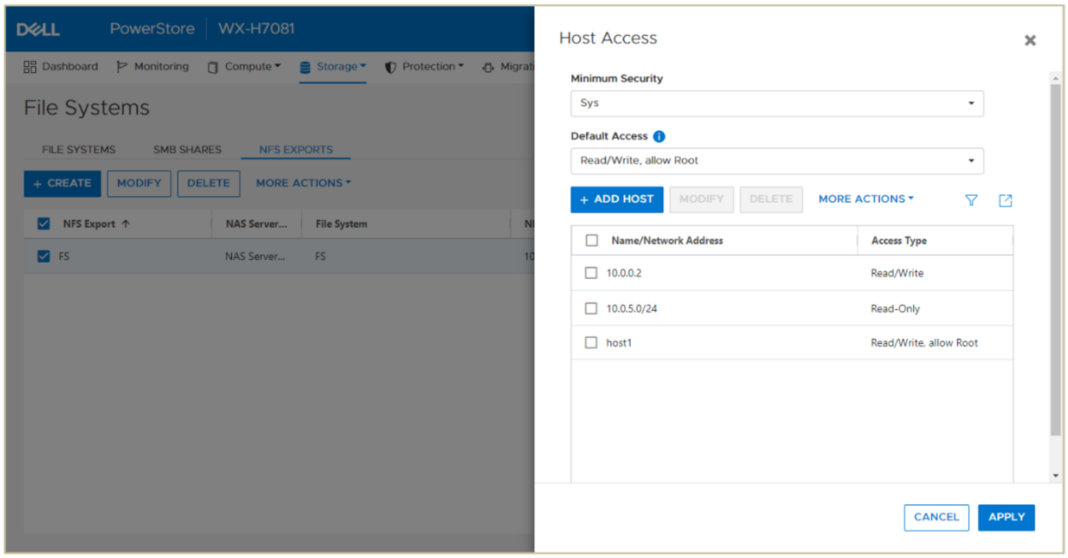


**Figure 11. NFS host access configuration**

## FTP and SFTP

NAS servers and file systems also support access for FTP and SFTP. SFTP is more secure since, unlike FTP, it does not transmit usernames and passwords in clear text. FTP and SFTP access can be enabled or disabled individually at the NAS server level. Only active mode FTP and SFTP connections are supported.

Administrators can control the types of user accounts that can access files over FTP and SFTP, such as SMB, UNIX, or anonymous users. A home directory restriction option limits access to only the user home directory on the file system. If this option is disabled, a default home directory can be specified. Users that have a home directory that is not defined or accessible are placed in the default home directory instead.

FTP and SFTP can track and record connections and access for the NAS server. The audit logging settings also allow administrators to define the audit log file directory and the maximum size of audit log files.

A welcome message and a message of the day can be displayed when users connect to the FTP or SFTP server. The welcome message is displayed before the client authenticates. The message of the day is only displayed after a client authenticates successfully.

For more granular control over access, FTP and SFTP support defining access control lists. Access can either be allowed or denied for a user-defined list of users, groups, and hosts to restrict FTP or SFTP access to only the necessary users. However, users, groups, or hosts with restricted access to FTP or SFTP can still access the NAS server and file systems over SMB or NFS as allowed by the ACLs or host access configurations for those protocols. Table 5 provides a list of FTP and SFTP protocol options.

**Table 5. FTP and SFTP options**

| Protocol options | Default |
|---|---|
| Enable FTP | Disabled |
| Enable SFTP | Disabled |
| Allow SMB Users Access to the FTP/SFTP server | Enabled |
| Allow UNIX Users Access to the FTP/SFTP server | Enabled |
| Allow anonymous Users Access to the FTP server | Disabled |
| Home Directory Restriction | Enabled |
| Default Home Directory | / |
| Enable FTP/SFTP Auditing | Disabled |
| Directory of Audit Files | /.etc/log |
| Maximum Size of Audit Files | 512 KB |
| Welcome Message and Message of the Day | Empty |
| Access Control List | Empty |

FTP and SFTP access can be authenticated using the same methods as NFS or SMB. Once authentication is complete, access is then considered to be the same as SMB or NFS for security and permissions purposes. The method of authentication that is used depends on the format that is used for the username. If **domain@user or domain\user** is used, SMB authentication is used. For any other single username format, NFS authentication is used. SMB authentication uses the Windows domain controller while NFS authentication uses the UDS or local files.

In order to use local files for FTP and SFTP access, the **passwd** file must include an encrypted password for the user. This password is only used for FTP and SFTP access. The local passwd file uses the same format and syntax as a standard UNIX system, and it can be used to generate the passwd file. On a UNIX system, use **useradd <user>** to add a new user and **passwd <user>** to set the password for that user. Then, copy the hashed password from the **/etc/shadow** file, add it to the second field in the **/etc/passwd** file, and upload it to the NAS server.

# Multiprotocol

When a NAS server has both the SMB and NFS protocols enabled, multiprotocol access is automatically enabled. Multiprotocol access enables accessing a single file system using the SMB and NFS protocols simultaneously.

Windows and UNIX have inherent differences in how they handle things such as authentication, identification, permissions, locking, and so on. Some examples of these differences are displayed in the following table.

**Table 6. UNIX and Windows differences**

| Name | UNIX | Windows |
|---|---|---|
| Authentication Provided By | Client or UDS (for Secure NFS) | Domain controller or Local Group Database |
| Username Length | 8 characters | 20+ characters |
| Identifiers | User ID (UID) and Group ID (GID) | Security Identifier (SID) |
| Group Concept | Simple User/Group Concept | Allows Nested Groups |
| File/Directory Ownership | Requires User AND Group Owner | Can be owned by a User or Group |
| Permissions | NFSv3 – Mode Bits<br>NFSv4 – ACLs | ACLs |
| File Locking | NFSv3 – Advisory<br>NFSv4 – Advisory or Mandatory | Mandatory |

Due to these inherent differences in the protocols, some configuration is required to maintain seamless access and enforce security across both protocols.

## User mapping

To understand how multiprotocol user mapping works, it is important first to understand how single protocol user mapping works for both SMB and NFS.

- SMB
  - When an SMB user connects to a share, they are identified by their SID.
  - Active Directory is used to resolve their SID to a human-readable username and conversely.

    Example: S-1-5-21-1553607022-1141325308-60145995-1789 ⇔ DELL\Tom
- NFS
  - When an NFS user connects to an export, they are identified by their UID and primary GID.
  - The UNIX Directory Service (UDS) or Local Files are used to resolve their UID to a human-readable username and conversely.

    Example: UID=1000 ⇔ Tom
  - The UNIX Directory Service (UDS) or Local Files are used to resolve their primary GID to a human-readable group name and conversely.

    Example: GID=1000 ⇔ Users

Ultimately, the goal of the multiprotocol-mapping process is to create a mapping between the Windows SID and the UNIX UID. Once the UID is known, it is possible to find its associated primary GID. To accomplish this task, use the usernames to bridge the two protocols together. The following shows an example of a complete multiprotocol mapping between the SID ⇔ UID and primary GID.

- Multiprotocol
  - An SMB user connects to a share and is identified by their SID.
  - Active Directory resolves their SID to a human-readable username.
  - The Windows username is converted to a UNIX username.
  - UDS or Local Files are used to resolve the UNIX username to their UID.
  - UDS or Local Files are used to resolve the UID primary GID.

    Example: S-1-5-21-155360702… ⇔ DELL\Tom ⇔ Tom ⇔ UID=1000 → GID=1000

In a multiprotocol configuration, the end-to-end mapping between the SID, SMB Name, UNIX Name, UID, and primary GID is crucial. Both Windows and UNIX resolvers must be available to provide their respective mappings, which are joined to create this end-to-end mapping. This mapping provides the ability for a Windows user to be matched to a UNIX user, and conversely, in order to enforce file security when the other protocol is used for access. This cross-protocol mapping is principally done by matching usernames between the protocols, but each protocol also requires a method to map their respective usernames to their IDs.

If the user mapping is not properly configured, users may be denied access to the file system, obtain access to files that they should not have access to, or be prevented from accessing their own files. This mapping enables the system to identify when the same user is trying to access their files, regardless of the access protocol.

Table 7 shows the components that are involved in the user-mapping process and a short description of their purpose.

**Table 7. User-mapping components**

| Name | Service | Description |
|---|---|---|
| Windows resolvers (SMB) | Local Group Database (LGDB) or domain controller (DC) | LGDB is used for local users<br>DC is used to resolve:<br>Windows account name ⇔ SID |
| UNIX Directory Service (NFS) | LDAP/NIS, Local Files, or Both | Used to resolve:<br>UNIX account name ⇔ UID and primary GID<br>UNIX group name ⇔ GID |
| Secure Mapping Cache | Secmap Cache | A local cache that contains all the mappings on a NAS server. The following mappings are tracked:<br>SID ⇔ usernames ⇔ UID |
| ntxmap | NTXMAP | Used for advanced name translations between protocols |

## SMB mapping – domain users

In a multiprotocol configuration, it is recommended to join the SMB server to an Active Directory domain for resolving SIDs to and from Windows usernames. When connecting

to a multiprotocol file system, domain users go through the user mapping process to create a mapping from the Windows SID to the UNIX UID and primary GID.

### SMB mapping – local users

Because local users on an SMB server are intended for SMB-only access, they are not mapped using this process. Because standalone SMB servers only support local users, they would not have the necessary mappings for a proper multiprotocol configuration.

If a local user connects to a multiprotocol file system over SMB, the LGDB is searched and used to resolve the SID to a Windows username. The local user is mapped to a dedicated UID range, starting with 2151678452 as the local Administrator user. The UID then increments with each additional local user.

Due to this, the UID of the local user on the file system is unlikely to match the UID configured on the UNIX client. From the NAS server's perspective, they are being tracked as two different users. This results in the same user having inconsistent permissions across different protocols. There are some potential workarounds for this:

- Manually configure UIDs to ensure that they are consistent with the local SMB server
  - To do this, create all the local users on an SMB server, determine the UIDs of the local users, and then configure the UNIX clients to use those UIDs
- If security is not a concern, using open permissions could be another option
  - If the files are accessible to everyone, then there is no need to maintain consistent permissions across protocols

### NFS mapping

In a multiprotocol configuration, it is recommended to enable a service for resolving UID and GIDs to and from usernames. The available options are:

- UNIX Directory Service (UDS): LDAP or NIS
- Local files

Although multiprotocol can be used without any of these services, the NAS server would not be able to create the end-to-end mapping described previously. This means when the same user attempts to access files using a different protocol, they may encounter permissions issues.

For more information about how to configure local files, NIS, or LDAP, see the respective sections in this document.

### Secure mapping cache

Secure mapping cache (secmap) is a cache that contains the mappings of users that have previously connected to the NAS server. This includes the SID, username, and UID for each user. Since secmap is a cache, it only stores mappings that are generated by the standard mapping mechanisms. It is not a resolution service and does not generate any mappings of its own.

Once a user mapping is stored in secmap, the NAS server leverages this local cache for future mapping lookups. Only new users connecting to the NAS server must rely on external services to resolve their mappings.

Under normal circumstances, secmap is persistent and does not need to be managed. However, in specific situations, it may be necessary to edit secmap such as when a user UID changes. In this case, the cached entry in secmap is no longer accurate and can be updated or deleted using the **svc_nas_cifssupport** command. If the entry is deleted, the user is treated as a new user the next time they connect, so their new mapping is resolved and stored in secmap.

### Ntxmap

Ntxmap is an optional local file that is used to provide name translations between protocols. The multiprotocol user-mapping workflow that is described previously assumes that users have the same usernames across both protocols. However, this may not be the case in all environments. In environments where usernames are different across protocols, ntxmap is required in order to translate usernames from one protocol to another. For example, if a user has a Windows account that is named **DELL\Tom** and a UNIX account that is named **Thomas**, the system cannot assume that these are the same user.

In addition to one-to-one mappings, ntxmap can also be used to provide advanced name translations. Ntxmap can be used to convert multiple usernames to a single username. For example, all the users in the Windows ENG domain can be mapped to a single UNIX user named **enguser.** Another option that is available is to provide name conversions. For example, all the UNIX users can be mapped to **DELL\unix-<username>.**

Ntxmap only provides username translations between protocols and does not provide any ID to username mappings. In environments where usernames are always the same and have a one-to-one mapping between protocols, ntxmap is not required.

### Automatic mapping for unmapped Windows accounts

Since the PowerStore file system is UNIX-based, all data that is written must be associated with a valid UID and primary GID. NFS users have a UID and primary GID natively available. However, SMB users must have a mapping that converts their native SID to a UID and primary GID. A reverse mapping from UID to SID is not always required since this is only necessary if Windows permissions are enforced (Windows Access Policy).

This feature enables the ability to automatically generate and assign a unique UID to Windows users that do not have a UID mapping. This feature enables access to the share for unmapped users, instead of denying them access. Since each user has a unique UID, UID-based feature such as user quotas can still properly track the consumption of each individual user.

This option is enabled by default on SMB-only and multiprotocol NAS servers. If this feature is enabled, the ability to configure default accounts is disabled. Since each UID is automatically assigned by the system, this should only be used in environments where the actual UID of these users is not critical. In environments where administrators want to control the UID assignments, this feature should be disabled. If this feature is disabled

and there are no other mapping methods available for unmapped users, they are denied access to the share.

This feature generates 32-bit UIDs with the most significant bit set to prevent conflicts with UIDs defined by the administrator in the UDS/local files. The range of UIDs generated by this feature is between 2147483649 (0x80000001) and 2151677951 (0x803FFFFF). The automatic UID is only assigned if the user does not already have a UID configured in the UDS/local files.

If the UDS or local files are updated to configure a UID after one is already assigned by this feature, the new entry in the UDS or local files is ignored. If you would like to use the entry in UDS/local files, you must delete the entry from secmap cache**.**

### Default accounts for unmapped users

This feature allows administrators to designate a specific Windows and/or UNIX account to serve as the mapping destination for unmapped users. This feature enables access to the share for unmapped users, instead of denying them access. For example, in an environment where many users only have Windows accounts, a default UNIX user may be designed to allow access for these unmapped users to the multiprotocol file system.

With default accounts enabled, the UID and primary GID of the default UNIX user are used if an unmapped Windows user attempts to access the file system through NFS. Similarly, the credentials of the default Windows user are used when an unmapped UNIX user attempts to access the file system through SMB.

Although multiple users could be writing to the file system as the default user, this user is still considered a single user since they share the same UID. This causes user quota calculations to be inaccurate. Also, the UNIX account may have ownership of files from many different Windows users.

The default UNIX user can be configured as a username or as a numerical UID and primary GID value. If a username is specified, the username must be resolvable to a UID and primary GID through the UDS or local files for the mapping process.

However, configuring the default UNIX user using a numerical UID and primary GID value does not require the user to have an entry in the UDS or local files. This is because all the information needed to create the mapping between the SID to UID and primary GID is available. The specified UID must be in the 32-bit range and follow this format**:** **@uid=<UID>,gid=<GID>**@. For example, if you want to configure a default UNIX user with a UID 1000 and primary GID of 2000, enter @**uid=1000,gid=2000**@.

This option is disabled by default on SMB-only and multiprotocol NAS servers. If this feature is enabled, the ability to enable automatic mapping is disabled. If this feature is disabled and there are no other mapping methods available for unmapped users; they are denied access to the share.

### Mapping process

Figure 12 shows the process that is used to resolve a Windows user (SID) to a UNIX user (UID and primary GID). In a multiprotocol configuration, local users on the SMB Server can still be used for SMB-only access, so they are not mapped to a UNIX user.

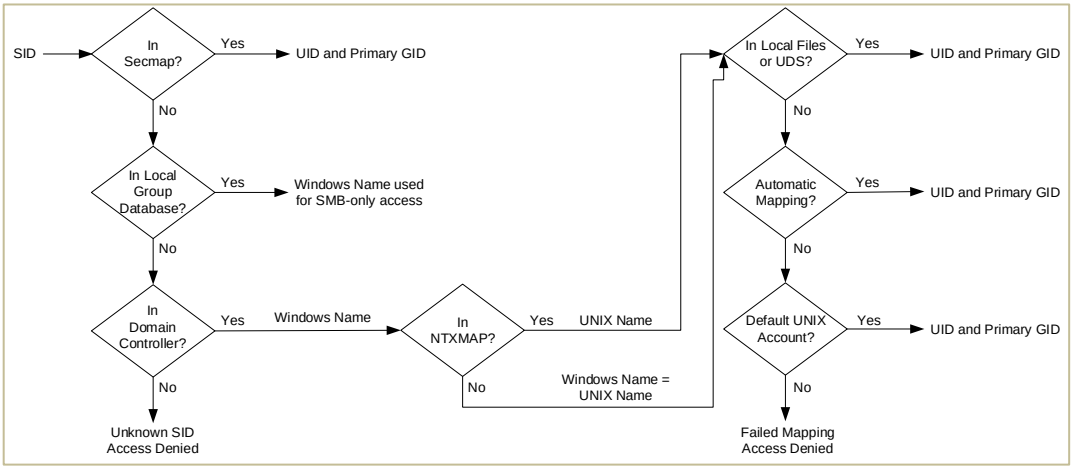


**Figure 12. SID to UID and primary GID mapping**

1. Secmap is searched for the SID. If the SID is found, the UID and primary GID mapping is resolved.
2. If the SID is not in secmap, the Windows username must be found.
    a. The local group database (LGDB) is searched for the SID to determine if this is a local user. If the SID is found, the name is `SMB_SERVER\USER`. Since this is a local user for SMB-only access, no UNIX mapping is required.
    b. If SID is not found in the LGDB, the DC is searched for the SID. If the SID is found in the domain, the name is `DOMAIN\USER`.
    c. If the SID is not resolvable, access is denied. This failed mapping added to the persistent secmap database.
3. If the default UNIX account is not used, the Windows name is translated to the UNIX name.
    a. If the Windows name is found in NTXMAP, that entry is used as the UNIX name.
    b. If the Windows name is not found in NTXMAP or if NTXMAP is disabled, the Windows name is used as the UNIX name.
4. The local files or UDS is searched for the UNIX name to find the UID and primary GID.
    a. If the UNIX name is found, the UID and primary GID mapping is resolved. This successful mapping is added to the persistent secmap database.
    b. If the UNIX name is not found, but the automatic mapping for unmapped Windows accounts feature is enabled, the UID is automatically assigned. This successful mapping is added to the persistent secmap database.
    c. If the UNIX name is not found, but a default UNIX account is configured, the UID and primary GID are mapped to that of the default UNIX account. This failed mapping added to the persistent secmap database.
    d. If the UNIX name is not resolvable, access is denied. This failed mapping is added to the persistent secmap database.

Mappings that do not result in a permanent UID are considered failed mappings – 2c, 4c, and 4d. Users with failed mappings are added to the secmap database with 4294967294 as their UID. This indicates that the mapping process must be retried the next time that the user connects. If a mapping is defined for these users later, they can convert in to successfully mapped users upon connecting. The secmap database is then updated accordingly with the permanent mapping. This means that these users must go through the mapping process each time they connect until they have a permanent mapping defined.

Figure 13 shows the process that is used to resolve a UNIX user (UID) to a Windows user (SID). This process is only needed if the access policy is set to Windows. This process is different compared to the UNIX UID that is always required, regardless of the access policy, since the UID is also used for file ownership and quota management purposes.

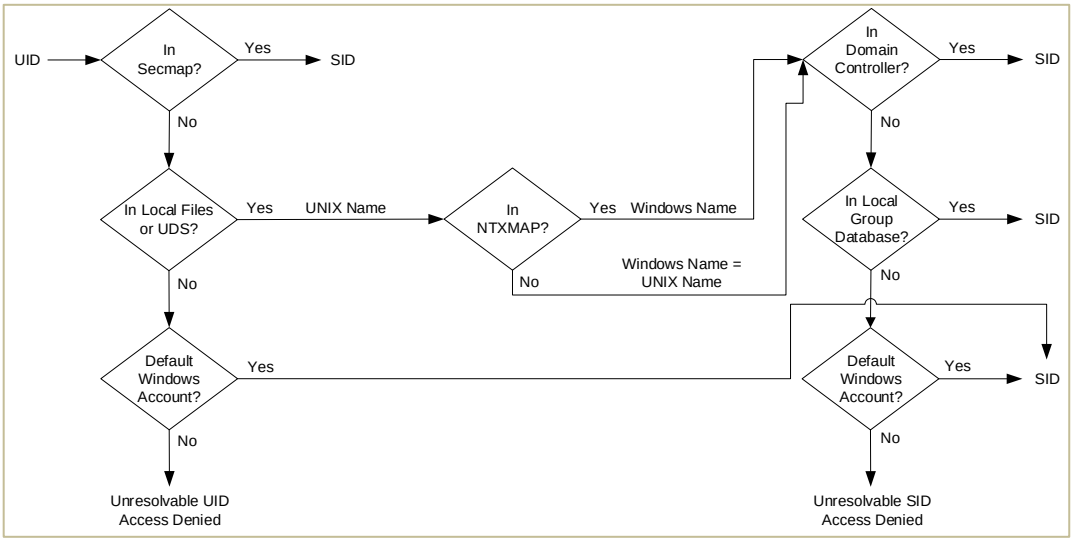


**Figure 13. UID to SID mapping**

1. Secmap is searched for the UID. If the UID is found, the SID mapping is resolved.
2. If the UID is not in secmap, the UNIX username must be found.
    a. Local files or UDS is searched for the UID. If the UID is found, the UNIX name is determined.
    b. If the UID is not found, but a default Windows account is configured, the UID is mapped to the default Windows account. If it does not exist, the default Windows user is added to the persistent secmap database.
    c. If the UID is not resolvable, access is denied.
3. If the default Windows account is not used, the UNIX name is translated to the Windows name.
    a. If the UNIX name is found in NTXMAP, that entry is used as the Windows name.
    b. If the UNIX name is not found in NTXMAP or if NTXMAP is disabled, the UNIX name is used as the Windows name.
4. The DC or LGDB is searched for the Windows name to find the SID.

a. The Windows name is searched in the DC. If the Windows name is found, the SID mapping is resolved. This successful mapping is added to the persistent secmap database.
b. If the Windows name contains a period (.) and the part of the name following the last period (.) matches an SMB server name, the LGDB of that SMB Server is searched. If the Windows name is found, the SID mapping is resolved. This successful mapping is added to the persistent secmap database.
c. If the Windows name is not found, but a default Windows account is configured, the SID is mapped to that of the default Windows account. If it does not exist, the default Windows user is added to the persistent secmap database.
d. If the Windows name is not resolvable, access is denied.

## Permissions

Security is also handled differently between SMB and NFS. For NFS, the authentication credentials are provided by the client or, for secure NFS, built from the UDS. User rights are determined by the NFSv3 mode bits or NFSv4 ACLs, and the UID/GIDs are used for identification purposes. There are no privileges associated with UNIX security.

For SMB, the credentials are built from the domain controller (DC) and local group database (LGDB) of the SMB server. User rights are determined by the ACL, and the SID is used for identification purposes. Windows security includes privileges such as TakeOwnership**,** Backup**,** and Restore that are granted by the LGDB or group policy object (GPO).

When configuring or managing a multiprotocol NAS environment, there are additional configuration options at the NAS-server and file-system levels that are related to how the permissions are handled between SMB and NFS users that are accessing file-system data. These options are shown in Table 8.

**Table 8. Multiprotocol permission options**

| Option | Level | Default |
|---|---|---|
| Access policy | File system | Native |
| UMASK (SMB) | Share | 022 |

## Access policy

The access policy is used to define how security is enforced on a multiprotocol file system. The default setting of Native maintains two separate sets of permissions for each file and the protocol that is used to access the file determines which set of permissions are checked. If SMB is used, the ACLs are checked. If NFS is used, the NFSv3 mode bits or NFSv4 ACL are checked.

If the multiprotocol environment is heavily weighted toward users of one type or another, setting the access policy to one of the other values may be desirable. The Windows setting only checks the ACLs and completely ignores the NFSv3 mode bits and NFSv4 ACL while the UNIX policy does the opposite.

Table 9 describes the available policies that can be configured at the file system level.

**Table 9. Access policy details**

| Access policy | Description |
|---|---|
| Native (default) | Manages access for each protocol separately with its own native security<br>Security for NFS shares uses the UNIX mode bits or NFSv4 ACL<br>Security for SMB shares uses the SMB Access Control List (ACL)<br>The two sets of permissions are independent and there is no synchronization between them<br>NFSv3 UNIX mode bits or NFSv4 ACL permission changes are synchronized to each other, but SMB ACL is not changed<br>SMB ACL permission changes do not change the NFSv3 UNIX mode bits or NFSv4 ACL |
| Windows | Uses the SMB ACL for both protocols<br>Upon request for NFS access, the Windows credential that is built from the DC/LGDB is used to check the ACL for permissions<br>NFSv3 UNIX mode bits or NFSv4 ACLs are updated when SMB ACL permissions are changed<br>NFSv3 UNIX mode bits or NFSv4 ACL permission changes are denied |
| UNIX | Uses the NFSv3 UNIX mode bits or NFSv4 ACL for both protocols<br>Upon request for SMB access, the UNIX credential that is built from the UDS/local files is used to check the NFSv3 mode bits or NFSv4 ACL for permissions<br>SMB ACL permissions are updated when NFSv3 UNIX mode bits or NFSv4 ACLs are changed<br>SMB ACL permission changes are allowed, to avoid causing disruption, but these permissions are not maintained |

Figure 14 shows how to configure the access policy on a file system.

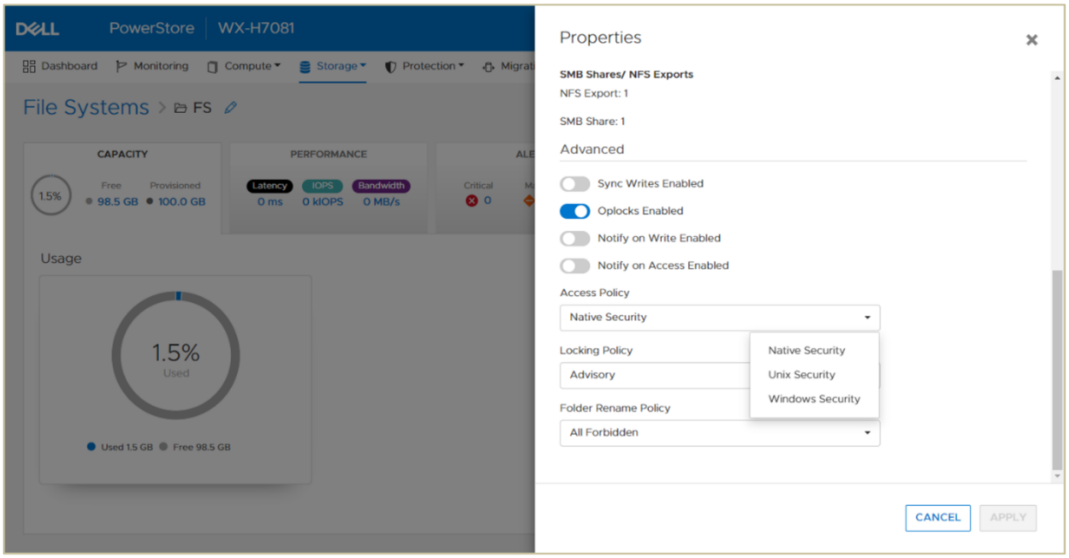


**Figure 14. Access policy**

### UMASK

For files created on NFS, the SMB ACLs are determined by ACL inheritance. For files created on SMB, the NFSv3 UNIX permissions are determined by the UMASK setting. The UMASK is a bitmask that controls the default UNIX permissions for newly created files and folders. This setting is only applied to new files and folders that are created on SMB on multiprotocol file systems.

The UMASK setting determines which UNIX permissions are excluded for new files and directories. By default, new files have 666 `(-rw-rw-rw-)` permissions while new directories have 777 `(drwxrwxrwx)` permissions. If the UMASK is set to the default value of 022, this means that new files have 644 `(-rw-r--r--)` permissions and new directories have 755 `(drwxr-xr-x)` permissions instead. Note that if NFSv4 ACL inheritance is present on the directory, this takes precedence over the UMASK setting.

This behavior is only used to determine the UNIX permissions when creating files. If permissions are changed on an existing file, the behavior depends on the configured access policy.

Figure 15 shows how to configure the UMASK setting on an SMB share.

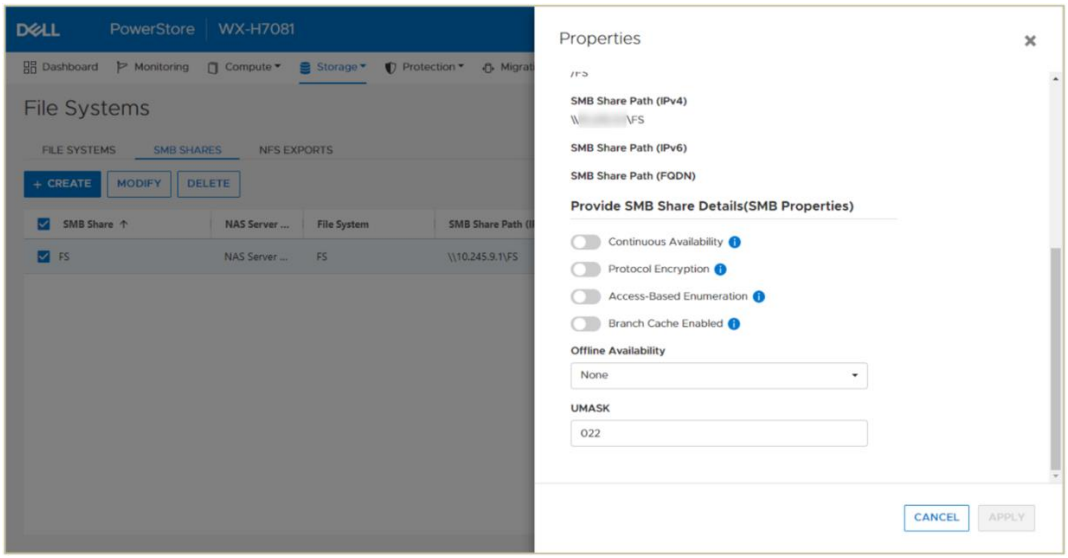


**Figure 15. UMASK setting**

## Multiprotocol policies

The locking and folder rename policies can be configured on multiprotocol file systems. These settings allow the administrator to control the behavior since locking and folder renaming behave differently depending on the protocol.

### Locking policy

Range locks allow hosts to lock a byte range of a file. These locks can be shared locks (denies writes) or exclusive locks (denies reads and writes). Each protocol implements either mandatory or advisory locking. For mandatory locks, any I/O to the locked range is denied. For advisory locks, it is the responsibility of the client to check for a lock and even if a lock is detected, it can disregard it and perform I/O anyway. Table 10 shows the locking semantics and mechanisms for NFSv3, NFSv4, and SMB.

**Table 10. Locking mechanisms**

| Protocol | Advisory/mandatory | Mechanism |
|---|---|---|
| NFSv3 | Advisory | Separate protocol (NLM) |
| NFSv4 | Advisory or mandatory (default) | Embedded in the protocol |
| SMB | Mandatory | Embedded in the protocol |

Due to the differences in the protocol specifications, the locking policy can be configured to control the behavior on multiprotocol file systems. The protocol that is used and the locking policy setting determines whether a lock prevents I/O:

- **Mandatory**: All I/O must honor SMB and NFSv4 range locks. NFSv3 range locks never prevent IO since they are always advisory due to protocol specification.
- **Advisory (default)**: NFSv3/v4/FTP I/O bypasses all range locks. SMB bypasses NFSv4 range locks, but always honors SMB range locks due to protocol specification. Any lock requests continue to report lock conflicts.

Table 11 also shows which locks are honored for each protocol and locking policy setting in a chart format.

**Table 11. Honored locks**

| Protocol | Mandatory (default) | Advisory |
|---|---|---|
| NFSv3 | SMB + NFSv4 | None |
| NFSv4 | SMB + NFSv4 | None |
| FTP | SMB + NFSv4 | None |
| SMB | SMB + NFSv4 | SMB |

Figure 16 shows how to configure the locking policy on a file system.

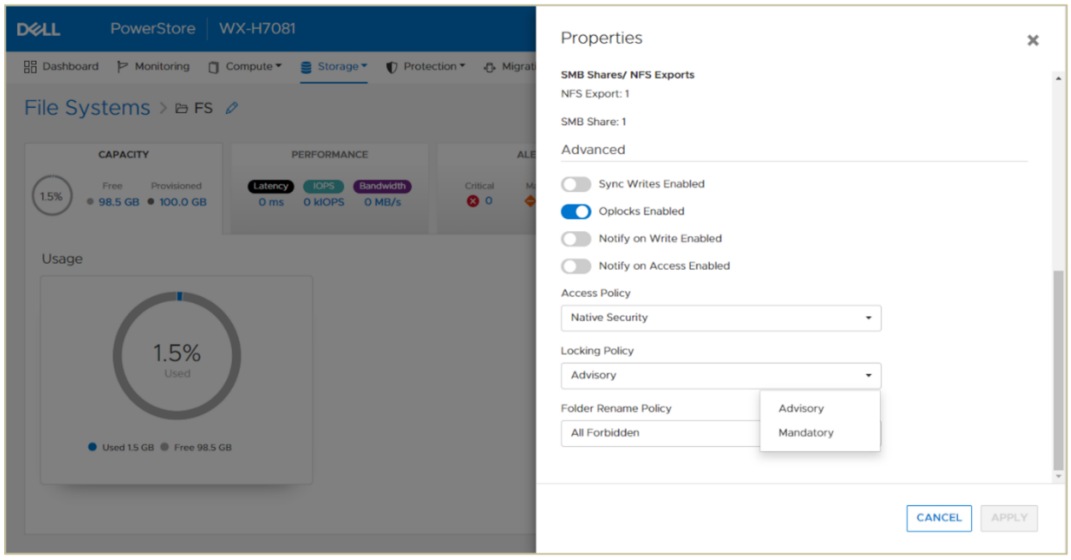


**Figure 16. Locking policy**

### Folder rename policy

According to the SMB protocol specifications, renaming any directory that is in the path of an open file is prohibited. For example, if **C:\Folder1\Folder2\Folder3\File1.txt** is opened by an SMB client, other clients are prevented from renaming any of the folders in the path leading up to **File1.txt**.

Clients using NFS or FTP do not have the same restriction. This is because SMB opens the entire path but NFS and FTP leverage file handles instead. Due to the differences between protocols, the folder rename policy allows the storage administrator to configure the behavior on multiprotocol file systems. The folder rename policy settings are only invoked when attempting to rename a folder in a path of an open file. Renaming folders that do not have any open files in the path are always allowed.

The folder rename policy can be configured to:

- **All Allowed**: All protocols can rename folders in the path of an open file without restrictions
- **SMB Forbidden**: SMB protocol renames of a folder in the path of an open file are prohibited, but other protocols are allowed
- **All Forbidden (default)**: No protocols can rename folders in the path of an open file

Figure 17 shows how to configure the folder rename on a file system.

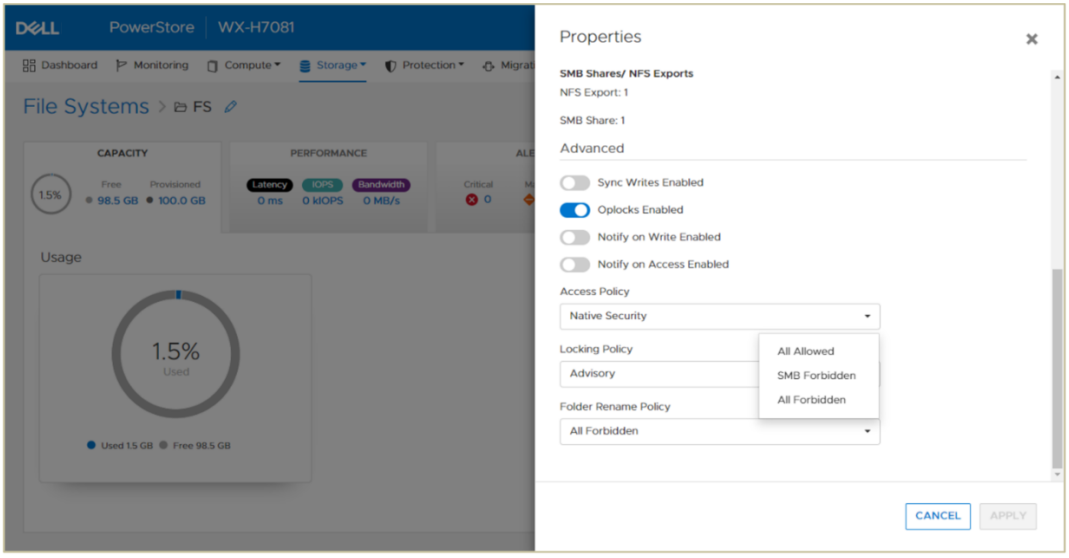


**Figure 17. Folder rename policy**

# Naming and directory services

PowerStore supports the following naming and directory services:

- DNS - A service that is used to provide translations between hostnames and IP addresses
- UDS (LDAP/NIS) – Services that provide a centralized user directory for username and ID resolution

- Local files – Individual files used to provide username and ID resolution

DNS is a commonly used service that relieves the need to remember IP addresses of individual components within the data center. Instead, human-friendly names can be used instead for connectivity and access.

A UDS should be configured if the environment is large and requires consistent UID and GID mappings across multiple NAS servers. The mappings can be managed from the centralized UDS server and propagated outward.

Local passwd and group files can be used in smaller environments since mappings are configured by uploading them to the NAS server. These local passwd and group files share the same syntax as UNIX environments. This means that the same files that are configured on hosts can be leveraged to provide mappings to the NAS server.

Once the UDS or local files are configured, you must specify the service in the UNIX Directory Service Search Order for it to be used. By default, this setting is set to **None** which means that no directory services are searched. It is possible to enable local files in addition to a UDS. If both are configured, there is an option to choose whether the local files, UDS, or both are used. This can be configured either during NAS server creation or by modifying the NAS server afterwards, as shown in Figure 18.

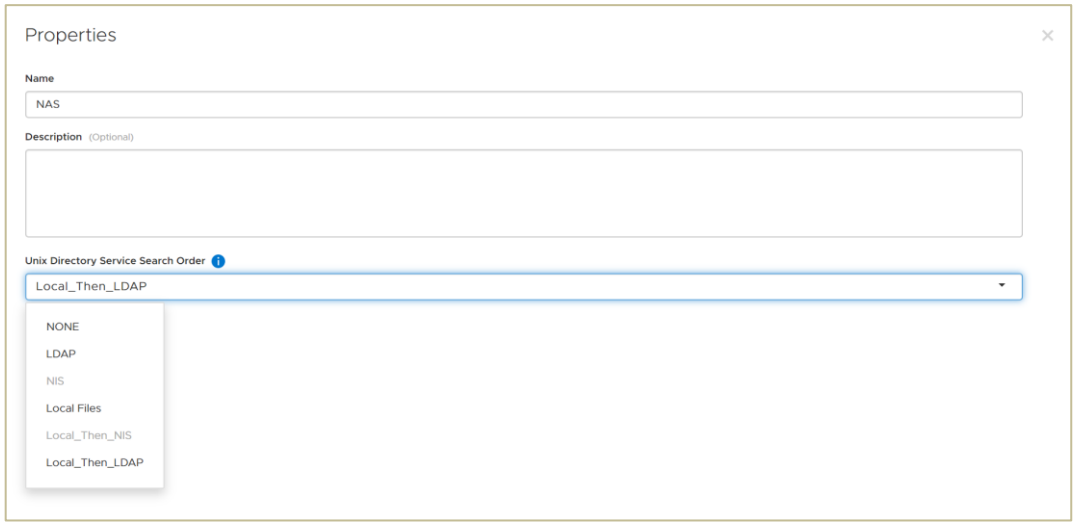


**Figure 18. UNIX Directory Service Search Order**

## DNS

DNS is required when configuring domain-joined SMB servers, Secure NFS, and proper multiprotocol configurations. When configuring DNS, provide the following information:

- UDP (default) or TCP protocol
- Domain
- IP address of DNS servers

NAS servers accept between one to three DNS servers. If multiple DNS servers are supplied, they can be moved up or down in the priority list.

## LDAP

LDAP can be used for the UDS. LDAP supports anonymous, simple, or Kerberos authentication. There are also options to configure the LDAP schema, enable LDAP secure (using SSL) to encrypt the LDAP traffic, and configure the Certification Authority (CA) certificate for authentication. The description and syntax for all the LDAP configuration settings are shown in Table 12.

**Table 12. LDAP configuration settings**

| | Anonymous | Simple (iPlanet or OpenLDAP) | Simple (AD LDAP or IDMU) | Kerberos |
|---|---|---|---|---|
| Server | LDAP Server IPs or Hostnames | | | |
| Port | LDAP Server Port Number - Default: 389 / SSL: 636 | | | |
| Base DN | Base DN in LDAP notation format. For example, if using svt.lab.com, the Base DN would be DC=svt,DC=lab,DC=com | | The Base DN is the same as the Fully Qualified Domain Name. For example, svt.lab.com | Base DN in LDAP notation format. For example, if using svt.lab.com, the Base DN would be DC=svt,DC=lab,DC=com |
| Profile DN (optional) | Profile DN for the iPlanet or OpenLDAP server | | | |
| Bind DN (simple only) | | User account in LDAP notation format. For example, cn=administrator,cn=users,dc=svt,dc=lab,dc=com | | |
| Bind DN Password (simple only) | | User account password | | |
| Notes | Active Directory (AD) is not supported with Anonymous LDAP authentication | | | See below for Kerberos authentication options |

There are two methods for configuring Kerberos:

- **Authenticate to the SMB domain**: Authenticate using the SMB server account or authenticate with other credentials.
- **Configure a custom realm**: Point to any type of Kerberos realm (Windows, MIT, or Heimdal). With this option, the NAS server uses the custom Kerberos realm that is defined in the Kerberos subsection of the NAS server Security tab. AD authentication is not used when you choose this option. If using NFS secure with a custom realm, you must upload a keytab file.

The LDAP configuration must adhere to either the IDMU, RFC 2307 or RFC2307bis schemas. See the RFC for a list of what is required for each schema. You can verify the current schema configuration by using the Retrieve Current Schema link on the LDAP page to retrieve the ldap.conf file, edit it, and upload a new version.

All containers that are specified in the ldap.conf file must point to a location that is valid and exists in the LDAP configuration, including ones that may not be in use, such as netgroup and host. If any entries are removed from this file, the NAS server automatically sets them to a default value based on the Base DN, which may result in lookup issues. Consult with your domain administrator to get the proper values for each container. Figure 19 shows an example of a valid LDAP schema for IDMU.

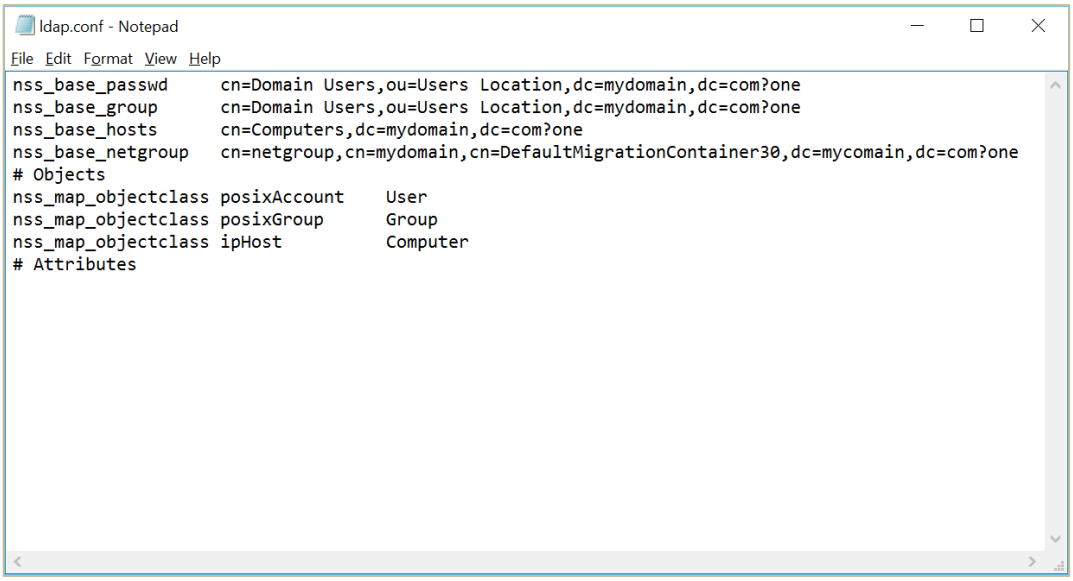


**Figure 19. LDAP conf file**

## NIS

NIS can be used for the UDS. To configure NIS, provide the following information:

- NIS domain
- IP addresses for NIS servers

If multiple NIS servers are supplied, they can be moved up or down in the priority list.

## Local files

Local passwd and group files can be used to resolve IDs and usernames. You can download the current version of the local files from the NAS server, which also provides syntax, examples, and additional details. After the files are edited with the user details, they can be uploaded back to the NAS server.

The passwd file uses the same format and syntax as UNIX-based operating systems so an existing file from a host could also be leveraged for the NAS server. Figure 20 shows the syntax and an example entry in a passwd file. The comment, home directory, and shell can be empty since they are not used by the NAS server. The NAS server is only interested in the username, hashed password (used for FTP authentication), UID, and primary GID.

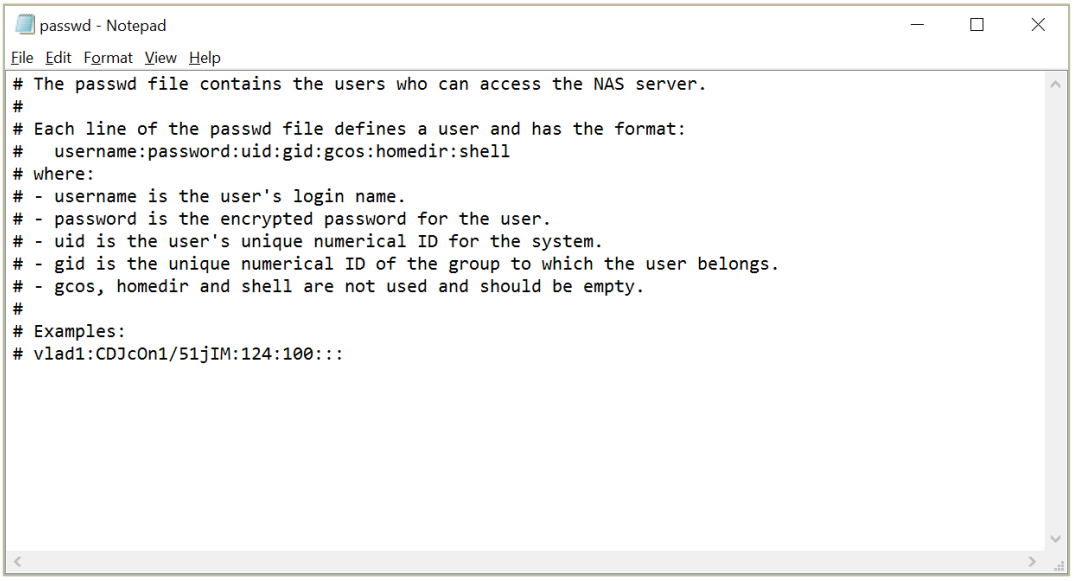


**Figure 20. Local passwd file**

# File systems

## Introduction

PowerStore file leverages a 64-bit file system that is highly scalable, efficient, performant, and flexible. The PowerStore file system is mature and robust, enabling it to be used in many of the traditional NAS use cases.

## Scalability

PowerStore file systems can accommodate large amounts of data, directories, and files. Table 13 shows several of the scalability attributes and limits of file system.

**Table 13. File system scalability**

| File system attribute | Limit |
|---|---|
| Maximum File System Size | 256 TB |
| Subdirectories per Directory | ~10 million |
| Files per File System | ~32 billion |
| Filenames per Directory | ~10 million |
| ACL IDs | 4 million |
| Timestamp Granularity | 1 nanosecond |

## Storage efficiency

All file systems are thinly provisioned and always have compression and deduplication enabled. With thin file systems, only 1.5 GB is allocated upfront for metadata, regardless of how large the file system is. As capacity is consumed on the file system, additional capacity is allocated on demand. This continuously happens until the specified file system size is reached and the file system becomes full.

Compression and deduplication help reduce the total cost of ownership and increase the efficiency of the system by reducing the amount of physical capacity that is needed to

store the data. Savings are not only limited to the file system itself, but also to its snapshots and thin clones. Compression and deduplication occur in line between the system cache and the backend drives. The compression task is offloaded to a dedicated chip on the node, which frees up CPU cycles.

## Performance

PowerStore file systems are tuned and optimized for high performance across all use cases. In addition, platform components such as Non-Volatile Memory Express (NVMe) drives and high-speed connectivity options enable the system to maintain low response times while servicing large workloads. For more information about performance best practices when configuring file systems, see the document *Dell PowerStore: Best Practices Guide*.

## Shrink and extend

PowerStore file systems provide increased flexibility by providing the ability to shrink and extend file systems as needed. Shrink and extend operations are used to resize the file system and update the capacity that is seen by the client. Extend operations do not change how much capacity is allocated to the file system. However, shrink operations may be able to reclaim unused space depending on how much capacity is allocated to the file system and the presence of snapshots or thin clones.

If the advertised file system size is too small or full, extending it allows additional data to be written to the file system. If the advertised file system size is too large, shrinking it limits the amount of data that can be written to the file system. For shrink and extend, the minimum value is equal to the used size of the file system and the maximum value is 256 TB. You cannot shrink the file system to less than the used size, as this would cause the client to see the file system as more than 100% full.

Figure 21 shows the file system properties page in PowerStore Manager, where you can shrink or extend a file system.

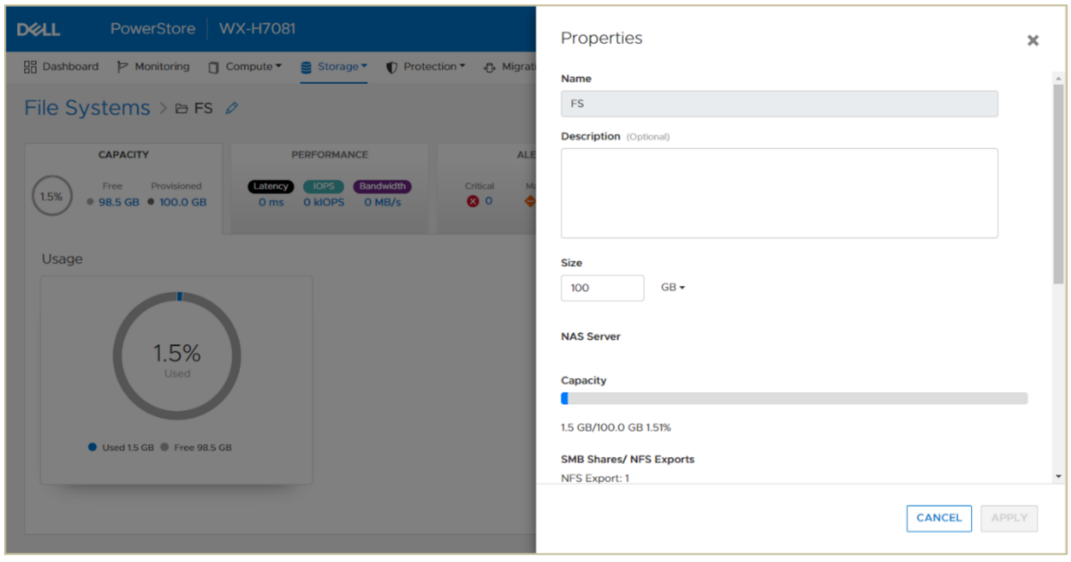


**Figure 21. File system shrink and extend**

## File system types

### General file systems

All PowerStore releases support General file systems. General file systems should be used for all file use cases except for VMware NFS datastores.

### VMware file systems

Starting with PowerStoreOS 3.0, an option to create a VMware file system is added. VMware file systems are designed and optimized specifically to be used as VMware NFS datastores. VMware file systems support AppSync for VMware NFS, Virtual Storage Integrator (VSI), hardware acceleration, and VM awareness in PowerStore Manager. The file system type selection in the file system provisioning wizard appears in Figure 22.

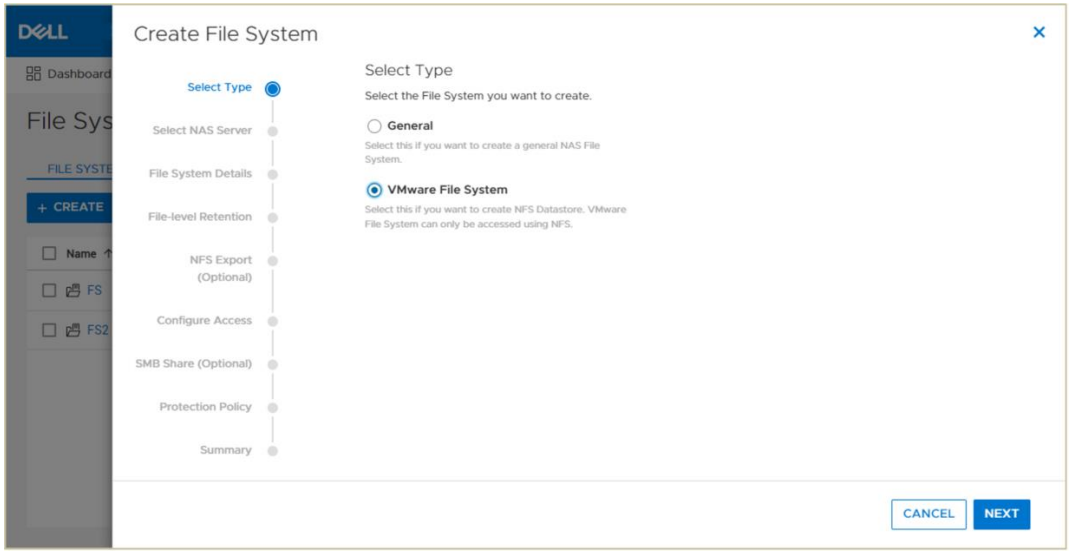


**Figure 22. File system type**

### Host IO size

When provisioning a VMware file system, the host IO size can be configured. To maximize performance, configure the host IO size to match the application IO size. The available options are 8K, 16K, 32K, and 64K. For general VMware NFS datastores, use the default setting of 8K. The file system type and host IO size settings are specified during creation and cannot be changed afterwards.

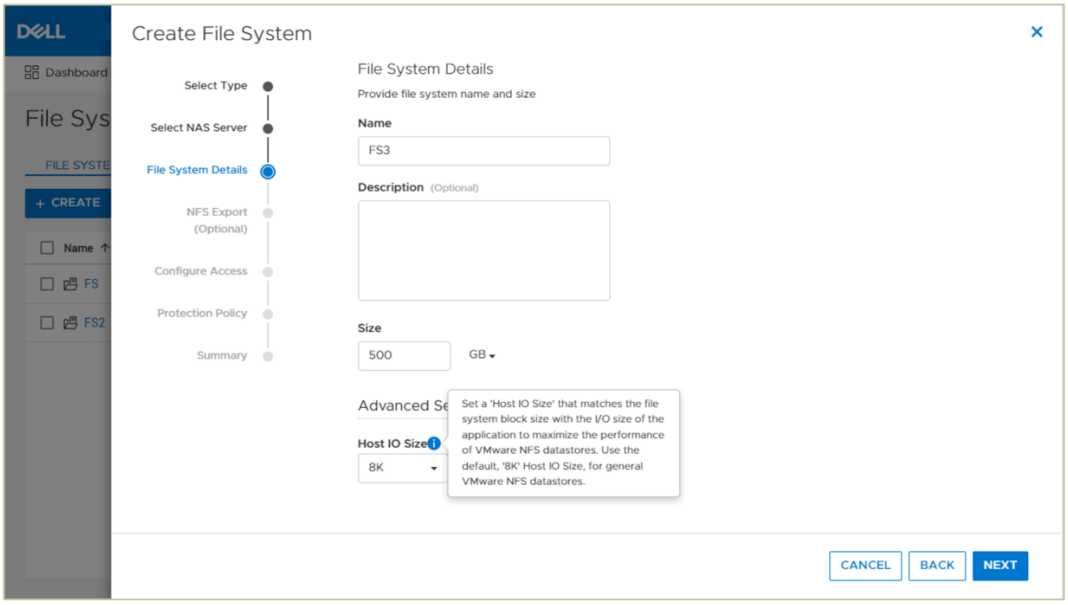


**Figure 23. Host IO size**

For existing file systems, you can identify the file system type by navigating to the **Storage → File Systems** page in PowerStore Manager and reviewing the **Config Type** column. In addition, you can view the configured host IO size by clicking the **Modify** button, as shown in Figure 24.

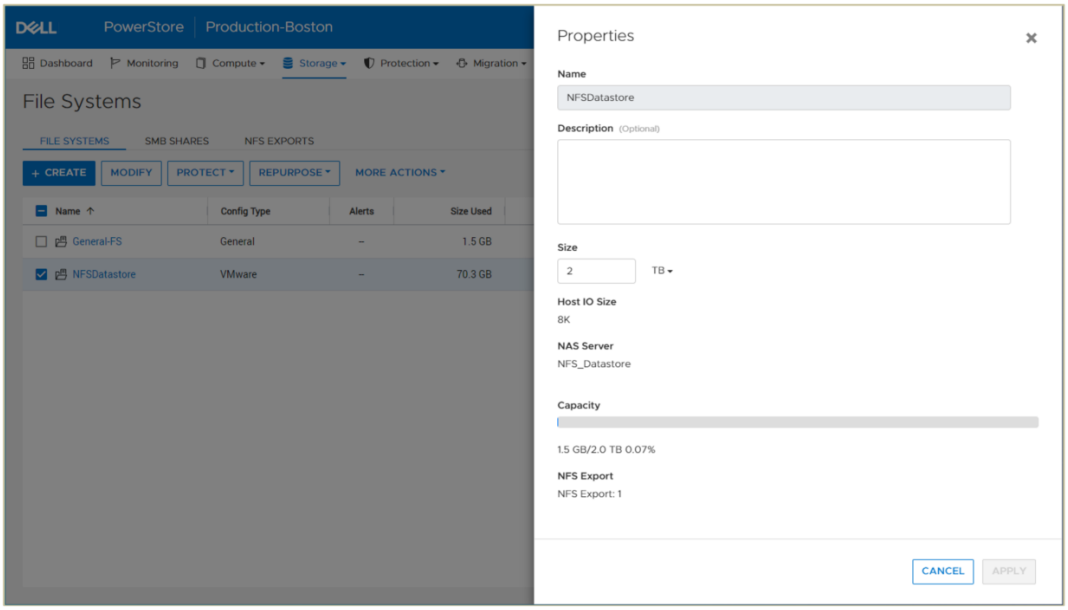


**Figure 24. Config Type column**

PowerStore monitors the IOs that are sent to a VMware file system over the previous 24-hour period. If more than 50% of the IOs are a different size than the configured host IO size, or are not aligned properly, PowerStore Manager displays a warning. If there are less than 86,400 IOs received over the 24-hour period (1 IOPS), this check is skipped.

## vStorage APIs for Array Integration (VAAI)

VMware file systems support vStorage APIs for Array Integration (VAAI). These are a set of storage primitives that enable offloading storage operations from the host to the storage system. To leverage VAAI, the PowerStore NAS VAAI plugin must be installed on the ESXi host. The following VAAI primitives are supported:

- Fast File Clone – Enables the creation of virtual machine snapshots to be offloaded to the array
- Full File Clone – Enables the offloading of virtual-disk cloning to the array
- Reserve Space – Enables provisioning virtual disks using the Thick Lazy and Eager Zeroed options over NFS
- Extended Statistics – Provides additional capacity utilization information

## VM awareness

PowerStore Manager provides insight into the VMware environment for VMware file systems. The name of the datastore in vSphere is displayed in PowerStore Manager, enabling administrators to correlate the resources easily, as shown in Figure 25. In PowerStore Manager, navigate to the **Storage → File Systems** page and enable the **Datastore** column.

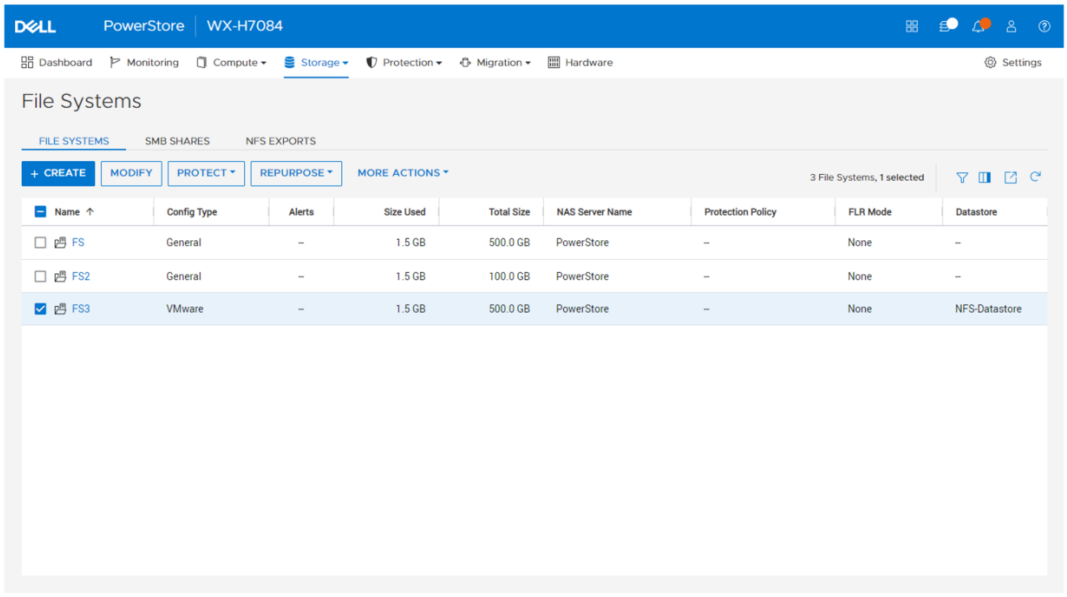


**Figure 25. VMware file system and associated datastore**

In addition, administrators can view the VMs that are stored on this datastore, as shown in Figure 26. In PowerStore Manager, navigate to the **Storage → File Systems → file system properties → Virtual Machines** page.

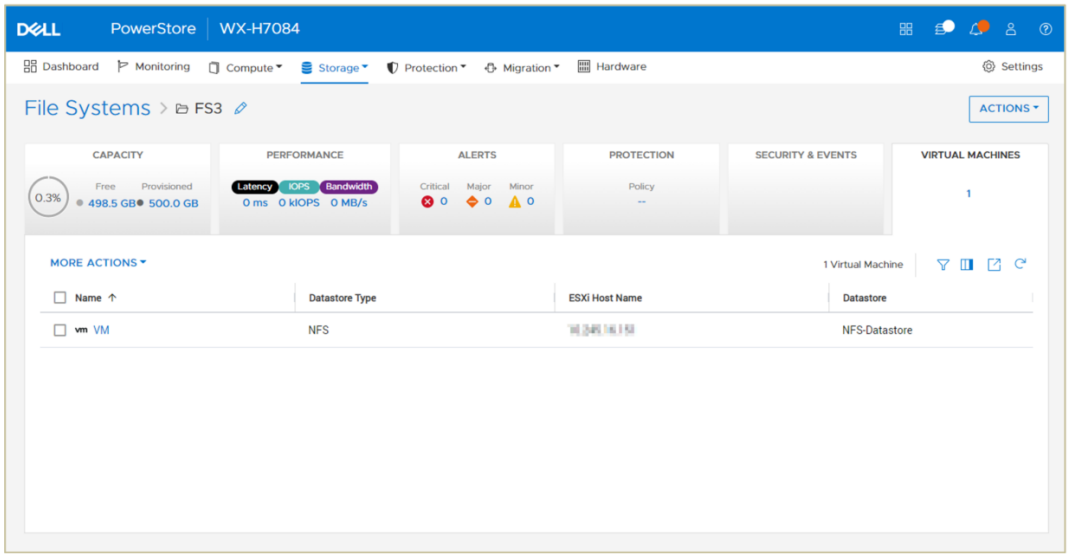


**Figure 26. Virtual machines on a VMware NFS datastore**

### Performance

VMware file systems are optimized for performance in VMware environments. On VMware file systems, metadata updates occur asynchronously because they are not critical for VMware NFS datastores. This behavior enables improved efficiency and performance on the file system.

### VMware file system interoperability

VMware file systems do not support quotas and File-Level Retention (FLR), because these features are not required for VMware NFS datastores.

# Supporting file features

## Introduction

PowerStore also includes a rich set of supporting file features to ensure that the data is secure, protected, and easily monitored.

## Quotas

To regulate file system storage consumption, PowerStore includes quota support to allow administrators to place limits on the amount of space that can be consumed. These simple but flexible quotas can easily be configured through any of the available management interfaces. PowerStore supports user quotas, quota trees, and user quotas on tree quotas. All three types of quotas can co-exist on the same file system and may be used in conjunction to achieve finer grained control over storage usage. Quotas are supported on General file systems. Quotas are not available on VMware file systems because they are not necessary for NFS datastores.

### User quotas

User quotas are set at a file system level and limit the amount of space a user can consume on a file system. Quotas are disabled by default, but can be enabled in the quota properties page dialog box along with the default user quota settings. The default quota limits are applied automatically to all users who access the file system. However,

the default limits can be overridden for specific users by creating a new user quota entry in PowerStore Manager.

Because all unspecified users are subject to the default quota settings, there is no ability to delete a user quota. Instead, a user quota can be set to 0 to allow unlimited access. Alternatively, a user quota can be set to inherit the default limits.

## Tree quotas

Quota trees limit the maximum size of a directory on a file system. Unlike user quotas, which are applied and tracked on a user-by-user basis, quota trees are applied to directories within the file system. Quota trees can be applied on new or existing directories.

If an administrator specifies a nonexistent directory when configuring a new quota tree, the directory is automatically created as part of quota configuration. However, an administrator can also specify an existing file system directory with existing data when creating a quota tree, allowing the ability to implement quotas on existing file system and directory structures after they have already been in production. If a tree quota is deleted, the directory itself remains intact and all files continue to be available.

Quota trees cannot be nested within a single directory. For example, if a quota tree is created on /directory1, another quota tree cannot be created on /directory1/subdirectory1. However, it is possible to have quota trees on /directory2, /directory3, and so on.

In PowerStoreOS 1.0, the quota grace period setting applies to all user quotas and tree quotas within the file system. Starting with PowerStoreOS 2.0, this setting only applies to user quotas because each tree quota can have its own individual grace period setting. Newly created tree quotas have a default grace period setting of seven days, which can be customized during creation or afterwards. A file system containing multiple tree quotas with different grace periods configured is shown in Figure 27.

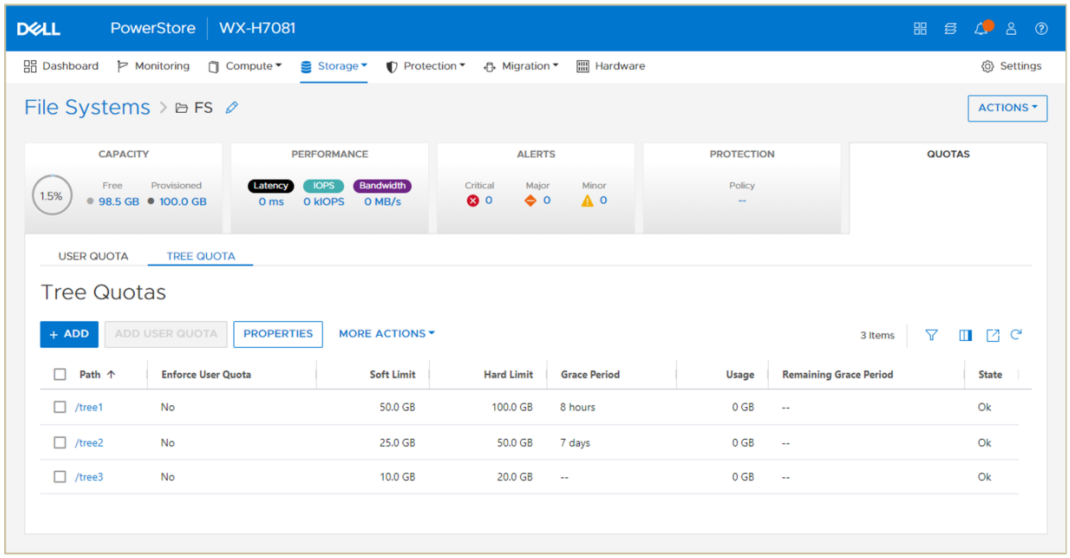


**Figure 27. Multiple tree quotas with different grace periods**

### User quotas on tree quotas

When a quota tree is created, it is also possible to create additional user quotas within that specific directory by choosing to enforce user quotas. When multiple limits apply, users are bound by the limit that they reach first. As an example, a single user may be bound by the following limits on a file system:

- File system user quota: 25 GB

  This user has a limit of 25 GB across the entire file system.

- Tree quota (/directory1): 100 GB

  Data from all users in this directory may not exceed 100 GB.

- User quota on tree quota (/directory1): 10 GB

  This user cannot consume more than 10 GB on this directory.

### Quota limits

All quotas consist of three major parameters that determine the amount of space that can be consumed on a file system and define the behavior when a limit is being approached or exceeded. These parameters are:

- Soft limit (GB)
- Grace period (time)
- Hard limit (GB)

The soft limit is a capacity threshold that triggers the grace period timer to start. For as long as the soft limit is exceeded, the grace period continues to count down. If the soft limit remains exceeded long enough for the grace period to expire, no new data can be added by the user or to the directory. The grace period has a minimum value of one minute and a maximum value of unlimited. However, if enough data is removed from the file system or directory to reduce the utilization below the soft limit before the grace period expires, data can continue to be written normally. Administrators can also allow users to continue writing data by increasing the value of the soft limit.

A hard limit is also set for each quota configured. Upon reaching a hard limit, no new data can be added to by the user or to the directory. When this happens, the quota must be increased, or data must be removed from the file system before additional data can be written.

Suppose the following user quotas are configured on a file system:

- Soft limit: 20 GB
- Grace period: 1 day
- Hard limit: 25 GB

The user copies data onto the file system, and after some time the user has stored 16 GB of files on the file system. Because the user has not reached their quota limits, the user is still able to add more data to the file system unimpeded. Figure 28 shows a file system under normal operation.

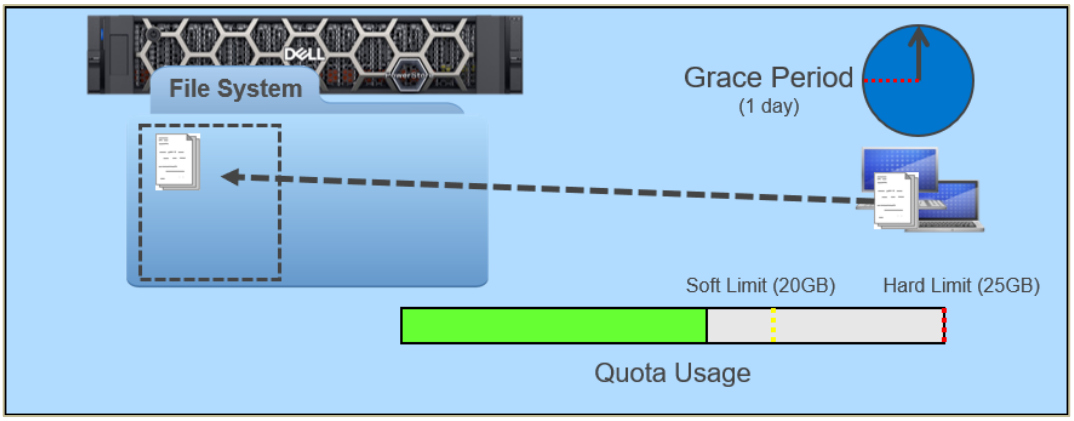


**Figure 28. Normal operation**

The user then continues to add more data to the file system, crossing the 20 GB soft limit. The user is still able to add additional data to the file system, but the grace period of one day begins. If the user does not remove data from the file system prior to the expiration of the grace period, they can no longer add data to the file system. They must remove enough data for the usage to fall below the soft limit first. Figure 29 shows a user that has crossed the soft limit and reached the end of their grace period.

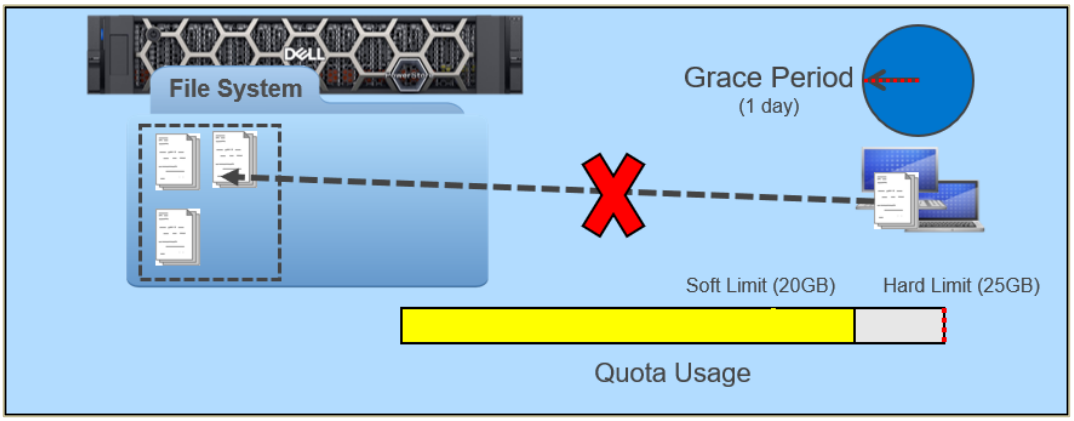


**Figure 29. Grace period reached**

If the user continues writing to and using additional space from the file system despite passing the soft limit, they may eventually reach the hard limit. When this happens, the user can no longer add data to the file system unless they remove some data first. Figure 30 shows a user that has crossed the soft limit and reached the hard limit.

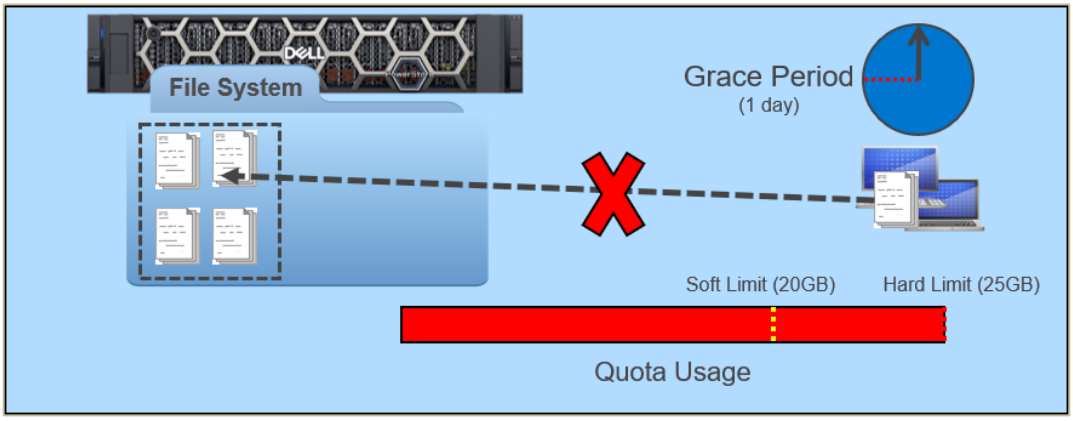


**Figure 30. Hard limit reached**

## Snapshots

PowerStore features pointer-based snapshots. These can be used for restoring individual files or the entire file system back to a previous point in time. Since these snapshots leverage redirect-on-write on technology, no additional capacity is consumed when the snapshot is first created. Capacity only starts to be consumed as data is written to the file system and changes are tracked.

Snapshots can be taken manually or by the integrated scheduler. Manual snapshots can be created at any time on the file system properties page. Scheduled snapshots can be configured by creating a protection policy with one or more snapshot rules and applying the policy to the file system.

When a snapshot is created, it can be configured to have no automatic deletion or retention until a specific date and time. If retention is set, the snapshot is automatically deleted upon reaching the retention date. This does not prevent the snapshot from being deleted before the retention date.

All snapshots that are created by a protection policy must have a retention date. The maximum retention date varies depending on how often the snapshots are scheduled. Each file system supports up to 256 snapshots so the combination of the snapshot frequency and retention cannot be configured to exceed this number.

For file snapshots, there are two access types available:

- Protocol (default) – Enables the snapshot to be shared and mounted like a file system
- Snapshot – Makes the snapshot available for self-service restores

Both protocol and snapshot type snapshots are read-only. In order to access the data on a Protocol snapshot, a share must be created. Once the share is created, the snapshot can be mounted on the host and accessed as if it were a read-only file system.

Snapshot type snapshots have integration with Windows and UNIX systems to enable self-service restores. On UNIX systems, users can access snapshot data by going to the .snapshot directory. On Windows systems, users can access snapshot data using the Previous Versions tab. Figure 31 shows how to access snapshot data through Previous Versions.

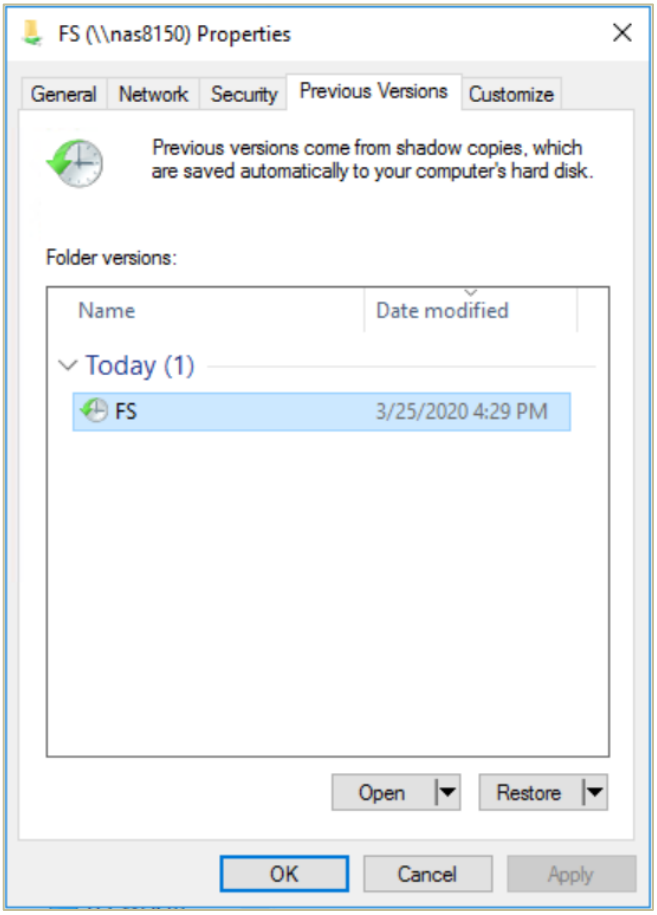


**Figure 31. Accessing snapshot data through Previous Versions**

Both protocol and snapshot type snapshots can be used for individual file and folder restores by copying the data off the snapshot back to the file system. Also, both types of snapshots can be used to restore the entire file system back to that point in time. When restoring the entire file system, a backup snapshot of the current file system data is taken by default.

Snapshots can be refreshed at any time. Refreshing the snapshot overwrites the contents of the snapshot with the data that is currently on the file system. Refreshing a snapshot only updates the data contents of the snapshot so the snapshot properties do not change.

For more information about snapshots, see the document *Dell PowerStore: Snapshots and Thin Clones*.

## Thin clones

### File system thin clones

Thin clones are pointer-based copies of the file system that can be written to. They enable file systems to be repurposed for copy data management use cases such as testing, development, or analytics. Thin clones can be created based on the current file system or any snapshot.

Since these are pointer-based clones, they continue to share blocks with the file system and snapshots. This means that no additional capacity is consumed when the thin clone is first created. Capacity only starts to be consumed as data is written to the file system or

thin clone and changes are tracked. Any changes to the data on the thin clone or the file system do not affect each other.

Although blocks are shared, there are no dependencies between objects within the family. For example, a file system that is used as the source for a thin clone can be deleted without impacting the thin clone. Once a thin clone is created, it is treated as if it is a file system. This means thin clones can also be created by using a thin clone as the source.

Since they are treated as independent resources, a thin clone can have its own set of snapshots and protection policy applied. When a thin clone is created, it inherits the source file system protection policy, and this can be changed afterwards. A thin clone does not inherit the snapshots, SMB shares, or NFS exports from the source.

## NAS server thin clones

Starting with PowerStoreOS 3.0, NAS servers can be cloned to create a NAS server with the same configuration. The only settings that are not copied are ones that would cause conflicts, such as the network interfaces and joining the SMB server to the domain. To enable access to the newly cloned NAS server, a new interface needs to be added to the clone. If a domain-joined SMB server is needed, enter a new SMB Computer Name, Domain Username, and Password to join it to the domain.

When cloning a NAS server, you can optionally choose any file systems that you want to also clone onto the new NAS server. Any file systems included in the NAS server clone operation have their SMB shares and NFS exports cloned as well, but snapshots are not cloned.

This enables the ability easily to configure new NAS servers for use cases such as test/dev and analytics, without impacting the production NAS server. A NAS server clone can also be used to access data on a replication destination. Cloning a NAS server with one file system is shown in Figure 32.

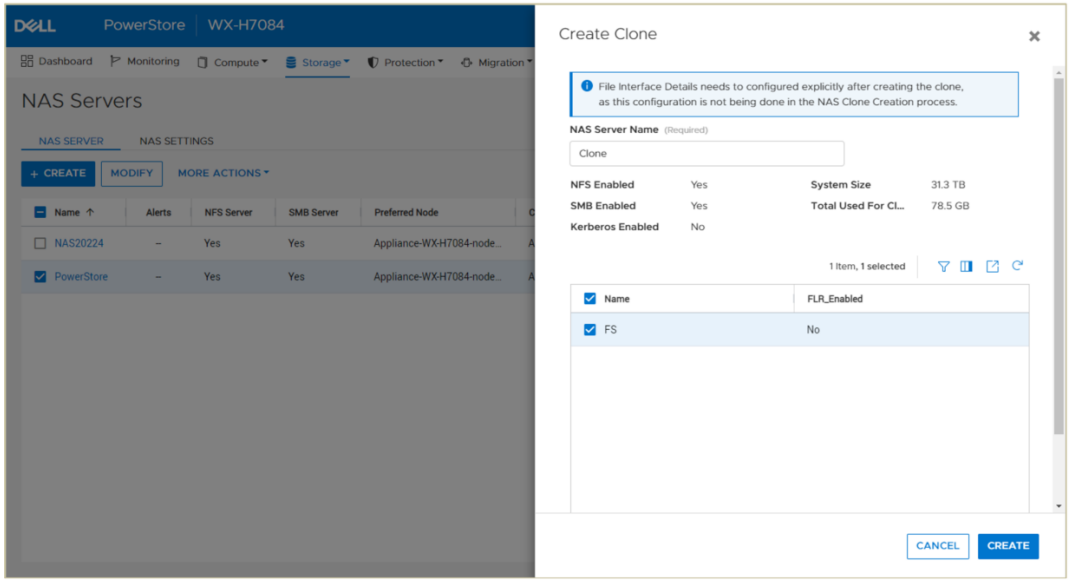


**Figure 32. NAS server clone**

For more information about thin clones, see the document *Dell PowerStore: Snapshots and Thin Clones*.

## Replication

Starting with PowerStoreOS 3.0, asynchronous file replication is available. Asynchronous replication can be used to protect against a storage system outage by creating a copy of data to a remote system. Replication is a software feature that synchronizes data to a remote system within the same site or to a different location. Replicating data helps to provide data redundancy and safeguards against storage system failures at the main production site. Having a remote disaster recovery (DR) site protects against system and site-wide outages. It also provides a remote location that can resume production and minimize downtime due to a disaster. The PowerStore platform offers many data-protection solutions that can meet disaster recovery needs in various environments.

The asynchronous replication for PowerStore is designed to have minimal impact on host I/O latency. Host writes are acknowledged when they are saved to the local storage resource, and no additional writes are needed for change tracking. Because write operations are not immediately replicated to a destination resource, all writes are tracked on the source. This data is replicated during the next synchronization. With protection policies, asynchronous replication uses the concept of a recovery point objective (RPO). The RPO is the acceptable amount of data, measured in units of time, that may be lost due to an outage. This delta of time also affects the amount of data that must be replicated during the next synchronization. It also reflects the amount of potential data loss in a disaster scenario.

For more information about file asynchronous replication, see the document *Dell PowerStore: Replication Technologies*.

## NDMP

PowerStore file supports three-way Network Data Management Protocol (NDMP) backups, allowing administrators to protect file systems by backing up to a tape library or other backup device. In an NDMP configuration, there are three primary components:

- Primary system – Source system to be backed up, such as PowerStore
- Data Management Application (DMA) – Backup application that orchestrates the backup sessions, such as Dell NetWorker
- Secondary system – The backup target, such as Data Domain

Three-way NDMP transfers both the metadata and backup data over the network. The metadata travels from the primary system to the DMA. The data travels from the primary system to the DMA and then finally to the secondary system. An example of a three-way NDMP configuration is shown in Figure 33.

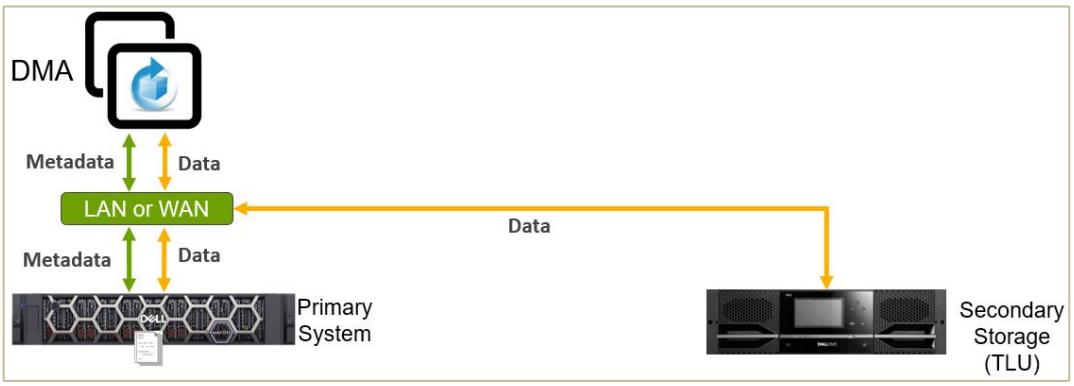


**Figure 33. Three-way NDMP backup**

PowerStore supports taking NDMP full backups, incremental backups, restores, and tape cloning. Both dump and tar backups are supported but volume-based backups (VBB) are not supported. The DMA can specify the parameters below when running an NDMP backup.

It is recommended to enable all these parameters when running an NDMP backup:

- HIST: Allows the backup application to request the file history from the storage system
- UPDATE: Allows the backup application to request the file history for incremental backups
- DIRECT: Enables the ability to restore a single file from a backup
- SNAPSURE: Allows the backup application to request a snapshot of the file system for backup purposes

## File-Level Retention

Starting with PowerStoreOS 3.0, File-Level Retention (FLR) is available. FLR is a feature that is used to protect file data from deletion or modification until a specified retention date. This functionality is also known as Write-Once, Read-Many (WORM).

### FLR modes

PowerStore supports FLR-Enterprise (FLR-E) and FLR-Compliance (FLR-C). FLR-C has additional restrictions and is designed for companies that need to comply with federal regulations. A comparison of FLR-E and FLR-C is shown in Table 14.

**Table 14. FLR-E and FLR-C**

| Name | FLR-Enterprise (FLR-E) | FLR-Compliance (FLR-C) |
|---|---|---|
| Functionality | Prevents file modification and deletion by users and administrators through NAS protocols such as SMB, NFS, and FTP | |
| Deleting a file system with locked files | Allowed (warning is displayed) | Not allowed |
| Factory reset (destroys all data) | Allowed | |

| Name | FLR-Enterprise (FLR-E) | FLR-Compliance (FLR-C) |
|---|---|---|
| Infinite retention period behavior | Soft - A file locked with infinite retention can be reduced to a specific time later | Hard - A file locked with infinite retention can never be reduced (a FLR-C file system that has a file locked with infinite retention can never be deleted) |
| Data integrity check | Not available | Available (see below for more information) |
| Restoring file system from a snapshot | Allowed | Not allowed |
| Meets requirements of SEC rule 17a-4(f) | No | Yes |

FLR-C includes a data integrity check as required by SEC rule 17a-4(f). When data is written, it is read back by the storage system to ensure that it has not changed during the write process. If the data does not match, the comparison is retried two more times. If there is still a mismatch, an error is reported. Files that are already locked do not have any additional write verification because they cannot be written to anymore. The write verification functionality is disabled by default because it can have a performance impact due to the additional overhead. It is important to enable this if it is required for compliance reasons. It can be enabled by changing the NAS server param `FLRCompliance.writeverify` from `0` to `1`.

The FLR mode is set when creating a General file system and cannot be changed afterwards. FLR is not available on VMware file systems because file locking is not necessary for NFS datastores. The available FLR modes are Off (default), Enterprise, and Compliance.

### Retention settings

If FLR is enabled, the following retention periods can be configured:

- Minimum - Specifies the shortest period for which files can be locked
- Default - Used when a file is locked, and a retention period is not specified
- Maximum - Specifies the longest period that files can be locked

The minimum, maximum, and default retention periods can be changed afterwards but any updates do not apply to any files that are already locked. The FLR step of the file system provisioning wizard is shown in Figure 34.

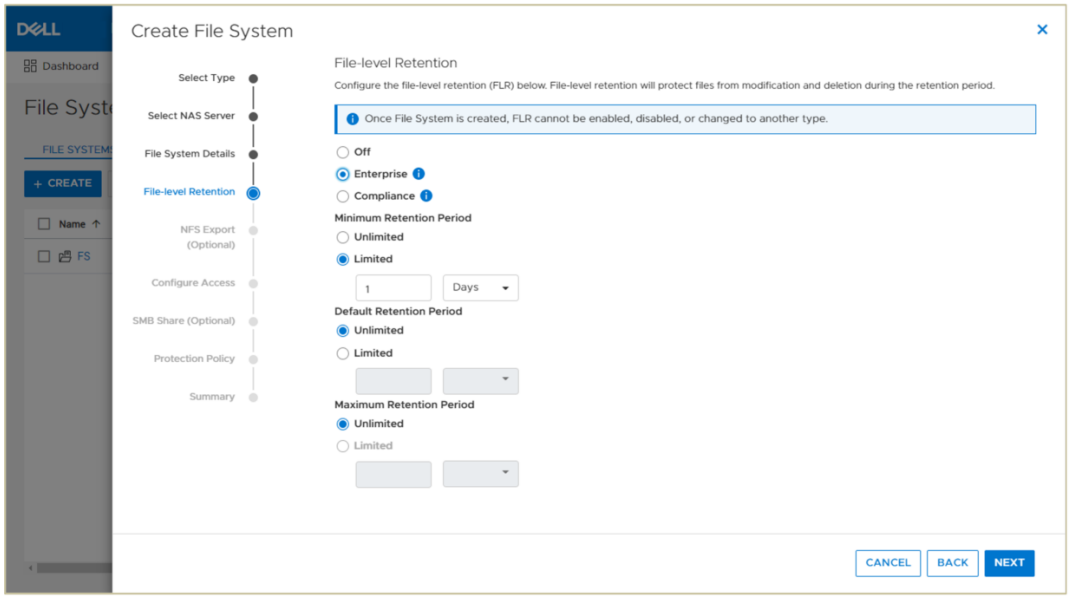


**Figure 34. File Level Retention configuration**

FLR also has built-in Auto-Lock and Auto-Delete functionality that can optionally be enabled for automation purposes. These settings can be configured on an FLR-enabled file system after it is provisioned, as shown in Figure 35. They are both disabled by default and can be modified at any time.

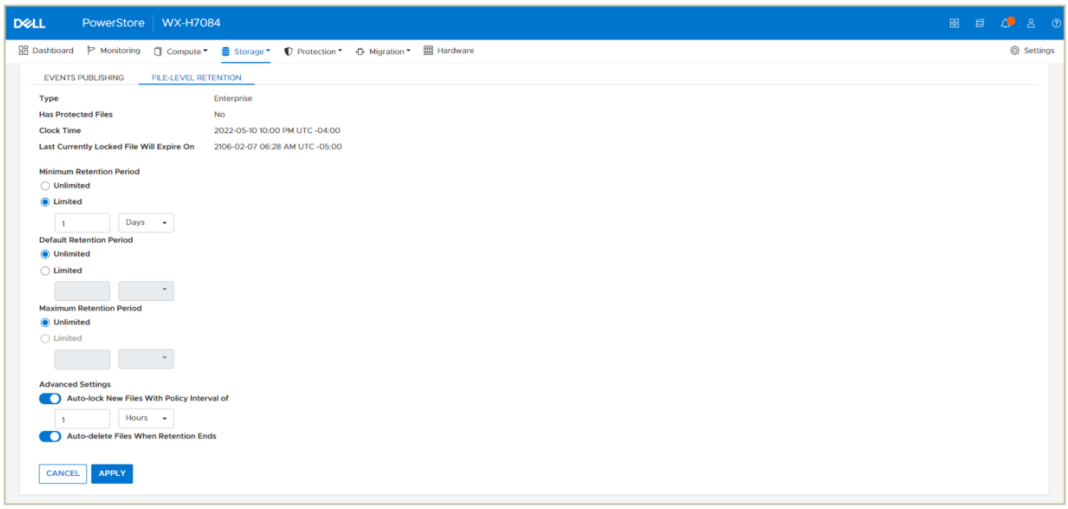


**Figure 35. File Level Retention properties**

- Auto-Lock - Files are automatically locked if they are not modified for a user-specified period (Policy Interval)
  - Automatically locked files use the default retention period
  - Files in append-only mode are also subject to automatic locking
- Policy Interval - Specifies how long to wait after files are modified before they are automatically locked
  - Specifies how long to wait after files are modified before they are automatically locked

  - Default: 1 Hour; Minimum: 1 Minute, Maximum, 365 Days
- Auto-Delete - Automatically deletes locked files after their retention date has expired
  - Weekly process that scans the file system to search for expired files
  - The first scan is initiated seven days after the feature is enabled

### FLR file states

In an FLR-enabled file system, files can be in one of four states:

- Not locked
  - Initial state of a new file
  - Treated in the same manner as a file in a non-FLR file system (can be modified, deleted, renamed, moved, and so on)
- Locked
  - Cannot be modified or deleted until the retention date has passed
  - Files can be manually locked by a user or automatically locked by the system or FLR Toolkit
  - A locked file can have its retention period extended, but not shortened
- Append only
  - Existing data cannot be modified or deleted
  - New data can be added to the end of the file
  - Useful for log files
  - Can be locked later
- Expired
  - File that was previously locked, but the retention date has passed
  - An expired file can only be re-locked or deleted from the file system, it cannot be changed to append-only (unless it is empty)
  - Data in expired files cannot be modified

### Locking files on Linux

The process to set a retention date and lock a file depends on the client operating system. Linux natively includes commands to perform these operations.

- Users can set the retention period using the `touch` command to set the last access time of the file to a future date and time
  - `root@vm:~# touch -at 202201141200 FLRtest01.txt`
- Users can lock the file by changing the access permissions to read-only using the `chmod` command
  - `root@vm:~# chmod -w FLRtest01.txt`

### Locking files on Windows

Windows does not offer a native UI or CLI option and requires using the Windows API `SetFileTime` function instead. The Dell FLR Toolkit is a Windows application that presents the `SetFileTime` function in a user-friendly manner, enabling administrators to manage files on an FLR-enabled file system.

The FLR Toolkit includes the following user interfaces and tools:

- FLR Explorer – GUI that can be used to set retention periods, lock files, run queries, and generate reports
- FLRApply – CLI options for setting retention periods and locking files
- FLR Monitor Service – Service that monitors folders in FLR-enabled file systems and acts on them, based on a user-configured policy
- Windows Explorer Enhancements – Adds FLR options to the Windows Explorer right-click and Properties menus

To set up the FLR Toolkit, enable the Distributed Hierarchical Storage Management (DHSM) API on the NAS server, specify the credentials that the toolkit will use, and install the toolkit on a Windows client.

### Creating append-only files

To create an append-only file, create an empty file, remove write permissions, and then re-apply write permissions to the file. To lock an append-only file later, simply set the retention date and remove write permissions again. This can be done with native Linux commands or by using the FLR Toolkit.

### FLR interoperability

Operations such as thin clones and replication on an FLR-enabled file system maintain the same FLR mode. For example, if the source file system has FLR-E enabled, the clone or replication destination file system also has FLR-E enabled, and this cannot be changed.

### FLR activity log

When an FLR-enabled file system is created, an FLR_Logs directory is automatically created on the root of the file system to store all FLR-related activity. This includes operations such as files getting locked successfully, attempted changes to locked files, and retention settings updates. Individual details of each event, such as the user, timestamp, file information, and results, are also included. Files in the activity log are identified by their inode number.

The log itself is an append-only file. When the size reaches 10MB, it is locked with the maximum retention period and a new log is created. The following is an example of the FLR activity log:

```
root@vm:~# cd /mnt/FLR_Logs/

root@vm:/mnt/FLR_Logs/# ls

flrLog20220126193040

root@vm:/mnt/FLR_Logs/# cat flrLog20220126193040
```

```
Wed Jan 26 19:30:40 2022 : Activity log file created

Wed Jan 26 19:30:40 2022 : Initial fs rp range: max = infinite,
default = 1D, min = 0D

Wed Jan 26 19:36:27 2022 : Set auto lock feature: oldVal = disable
: newVal = enable : Passed

Wed Jan 26 19:36:27 2022 : Set auto lock policy Interval: oldVal =
3600 : newVal = 60 : Passed

Wed Jan 26 19:40:58 2022 : Worm commit clean file : Inode No =
9459 : Generation No = 1643225480 : Uid = 0 : Gid = 1 : FileMode =
444 : FileSize = 14 : RP = Thu Jan 27 19:40:58 2022 : Passed
```

## CAVA

Common Anti-Virus Agent (CAVA) provides an anti-virus solution to SMB clients by using third-party anti-virus software to identify and eliminate known viruses before they infect files on the storage system. Windows clients require this to reduce the chance of storing infected files on the file system and protects them if they happen to open an infected file. This anti-virus solution consists of a combination of the PowerStore, Common Event Enabler (CEE) CAVA agent, and a third-party anti-virus engine. CAVA is enabled on a per NAS server basis.

PowerStore monitors events and triggers the anti-virus engine to initiate a scan when necessary. Some of the possible event triggers include file renames, modifications, and first reads. While a file is being scanned, access to the file from any SMB client is temporarily blocked. The CAVA solution is for clients running the SMB protocol only. If clients use the NFS or FTP protocols to create, modify, or move files, the CAVA solution does not scan these files for viruses.

CAVA can be customized depending on your specific needs. It can scan specific file extensions, exclude specific file extensions, configure the maximum file size to be scanned, configure the behavior if the anti-virus server goes offline, and more. To ensure that file scanning is maintained if an anti-virus server goes offline or cannot be reached, you should configure at least two CAVA servers.

Figure 36 shows the anti-virus configuration page in PowerStore Manager.

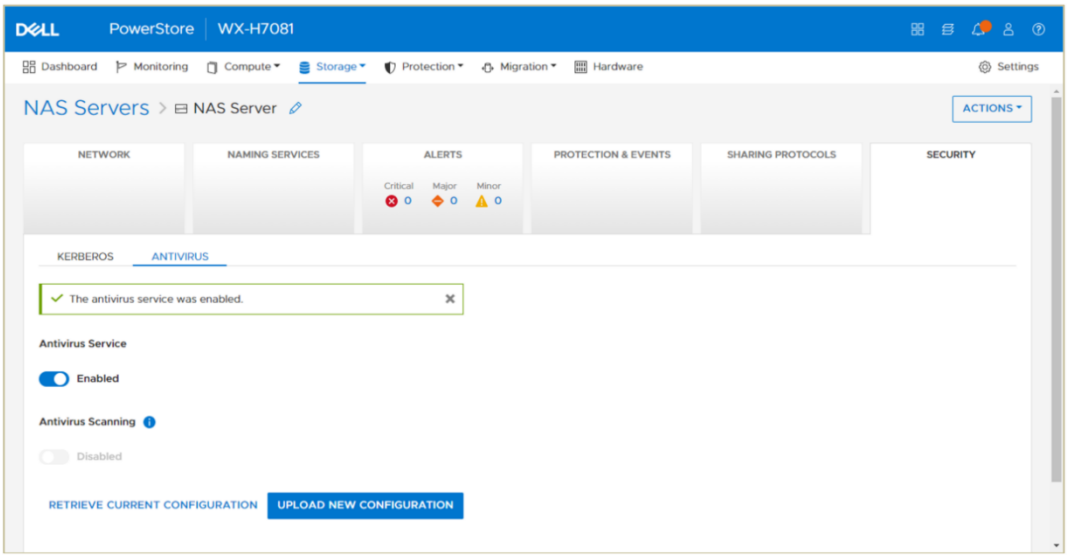


**Figure 36. Anti-virus configuration**

For a list of supported AV engines or for more information about how to configure CAVA, see the documents *Dell PowerStore Simple Support Matrix* and *Dell PowerStore Configuring SMB* on Dell.com/powerstoredocs.

## CEPA

Starting with PowerStoreOS 3.0, Common Event Publishing Agent (CEPA) is available. CEPA delivers SMB and NFS file and directory event notifications to a server, enabling them to be parsed and controlled by third-party applications. This can be used for use cases such as detecting ransomware, managing user access, configuring quotas, and providing storage analytics. The event notification solution consists of a combination of the PowerStore, Common Event Enabler (CEE) CEPA software, and a third-party application.

CEPA includes the following facilities that can be leveraged by third-party applications:

- Auditing – Enables file auditing for operations such as create, open, delete, close, rename, and ACL updates
- Centralized Quota Management (CQM) – Enables managing quotas across multiple storage systems
- VCAPS – Notifies search and indexing appliances when it is time to re-scan files

### Publishing Pools

To configure CEPA, create a Publishing Pool and Events Publisher on the **Storage → NAS Servers → NAS Settings** page in PowerStore Manager. The publishing pool specifies which events should trigger notifications and to which servers they should be sent. There can be up to five CEPA servers and they can be specified by IPv4 address, IPv6 address, or FQDN. The available events fall under three categories:

- Pre-Events – When an operation is requested, the NAS server sends a notification and waits for approval before allowing the operation to occur
- Post-Events – NAS server sends a notification after an operation occurs

- Post-Error-Events – NAS server sends a notification if an operation generates an error

Figure 37 shows the publishing pool configuration page.

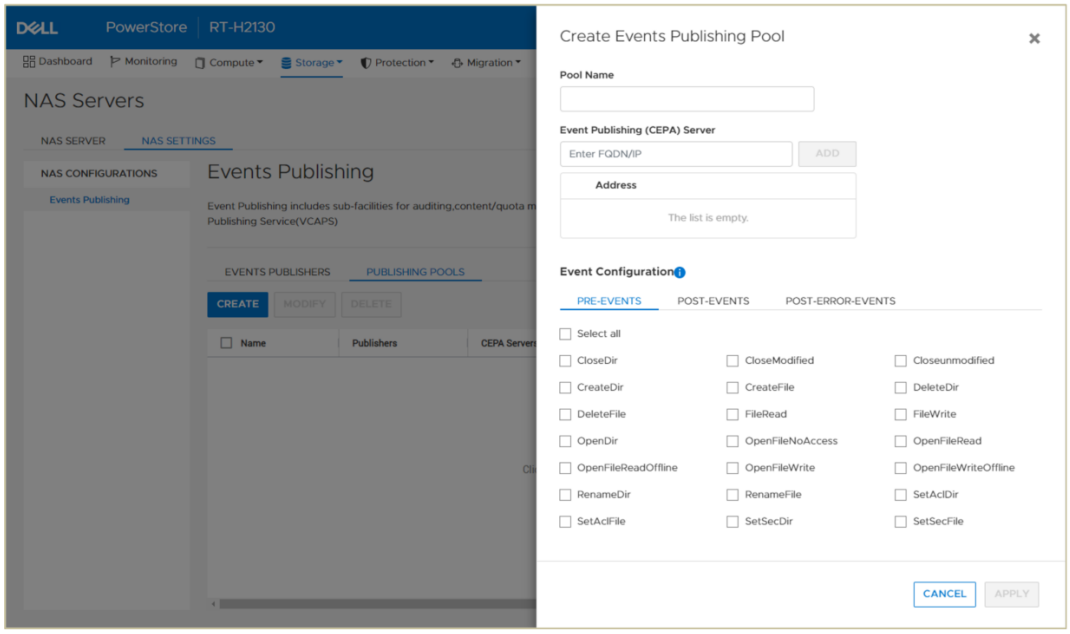


**Figure 37. Publishing pool configuration**

## Events publisher

The events publisher specifies one to three publishing pools and enables configuration of advanced settings.

- Pre-Events Failure Policy – Determines the pre-event behavior if PowerStore cannot reach CEPA Server
  - Ignore (default) – Consider pre-events acknowledged when CEPA servers are offline
  - Deny – Deny user access when a corresponding pre-event request to CEPA servers failed
- Post-Events Failure Policy – Determines the post-event behavior if PowerStore cannot reach CEPA Server
  - Ignore – Continue and tolerate lost events
  - Accumulate (default) – Continue and persist lost events in an internal buffer
  - Guarantee – Persist lost events, deny file systems access when the buffer is full
  - Deny – Deny access to file systems when CEPA servers are offline
- Connectivity and protocol settings
  - HTTP and Port – HTTP and 12228, by default
  - Microsoft RPC and Accounts – Enabled and SMB, by default
  - Heartbeat and Timeout – 10 sec and 1000ms, by default

Figure 38 shows the events publisher configuration page.

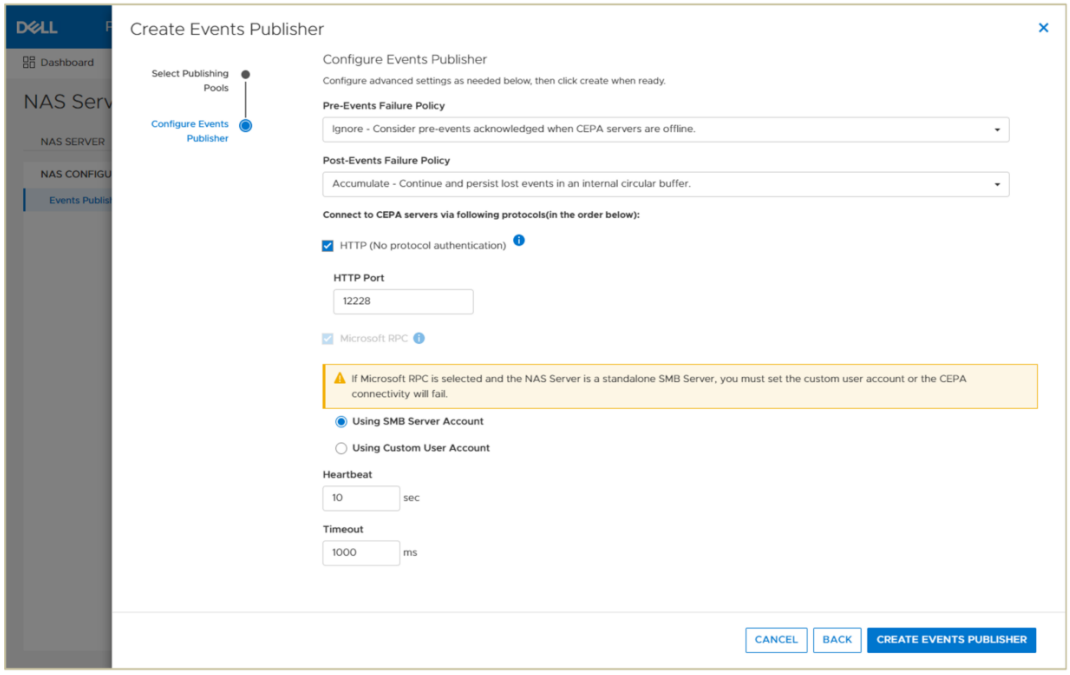


**Figure 38. Events publisher configuration**

## Enabling events publishing

When an events publisher is created, events publishing can be enabled on a NAS server. Multiple NAS servers can use the same events publisher. This can be done by navigating to the **NAS server properties → Security & Events → Events Publishing** page. When enabling events publishing, there is also an option to **Enable for all existing file systems under the NAS server** for **SMB** and **NFS**, as shown in Figure 39.

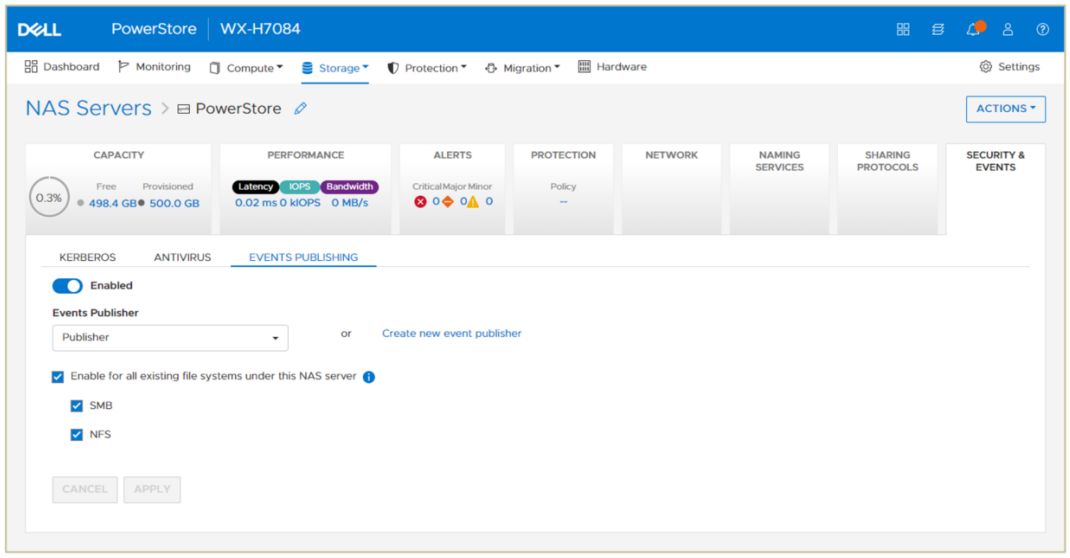


**Figure 39. Enabling events publisher on a NAS server**

A NAS server can contain a mix of file systems with and without events publishing enabled. When events publishing is enabled on the NAS server, it can be enabled or

disabled on each individual file system. This can be done by navigating to the **Storage → File Systems → file system properties → Security & Events** page, as shown in Figure 40.

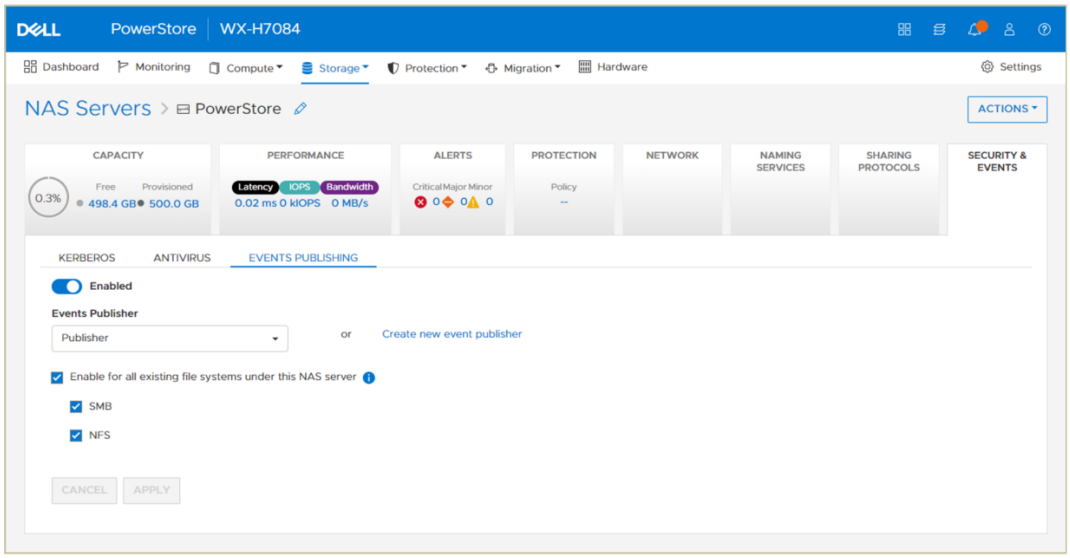


**Figure 40. Events publishing on a file system**

For a list of supported third party applications or for more information about how to configure CEPA, see the documents *Dell PowerStore Simple Support Matrix, Dell PowerStore Configuring SMB, and Dell PowerStore Configuring NFS* on Dell.com/powerstoredocs.

## MMC snap-in

The Microsoft Management Console (MMC) snap-in enables management of the file features and functions directly from a Windows client. Features that can be configured using the snap-in include:

- AntiVirus: Enables configuring anti-virus parameters such as file extensions to scan or exclude, maximum file size to scan, retry timeouts, and more.
- Audit Policy: Determines which security events are logged in the SMB security log. You can log successful attempts, failed attempts, both, or neither. These events can be viewed in the Windows Event Viewer.
- User Rights Assignment: Manages the privileges that users and groups have on the SMB server.
- HomeDir (home directories): Configures settings for the home directories feature. This can be used for options such as automatically creating home directories if they do not exist.

Figure 41 shows the audit policy management screen using the MMC snap-in.

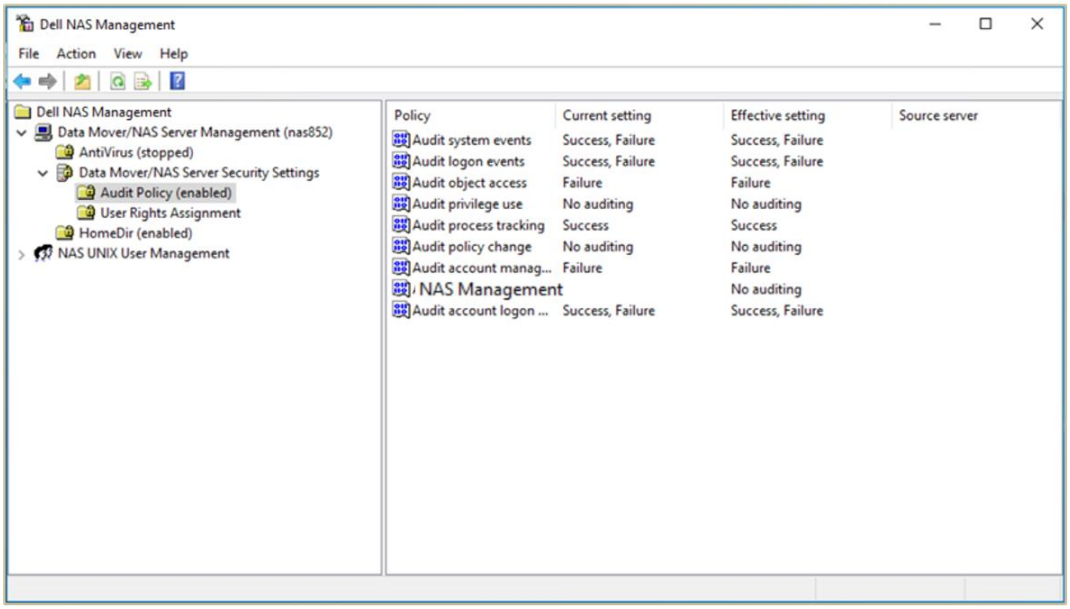


**Figure 41. MMC snap-in**

# Metrics

In PowerStoreOS 1.0, performance metrics display the overall backend performance, which includes both file and block workloads. Starting with PowerStoreOS 2.0, the ability to view file-specific performance metrics is also available. The overall backend performance metrics are displayed in the “Overall” tab, and file metrics are displayed in the “File” tab.

File metrics are available at the node, appliance, and cluster level:

- Node – 20-second granularity
  - Node-level SMB metrics – 5-second granularity
  - Node-level NFS metrics – 5-second granularity
- Appliance – 20-second granularity
- Cluster – 5-second granularity

The available metrics are:

- Read, write, and total IOPS
- Read, write, and total bandwidth
- Read, write, and average latency
- Read, write, and average IO size

Figure 42 shows the file metrics page, displaying the node-level SMB protocol metrics on Node B.

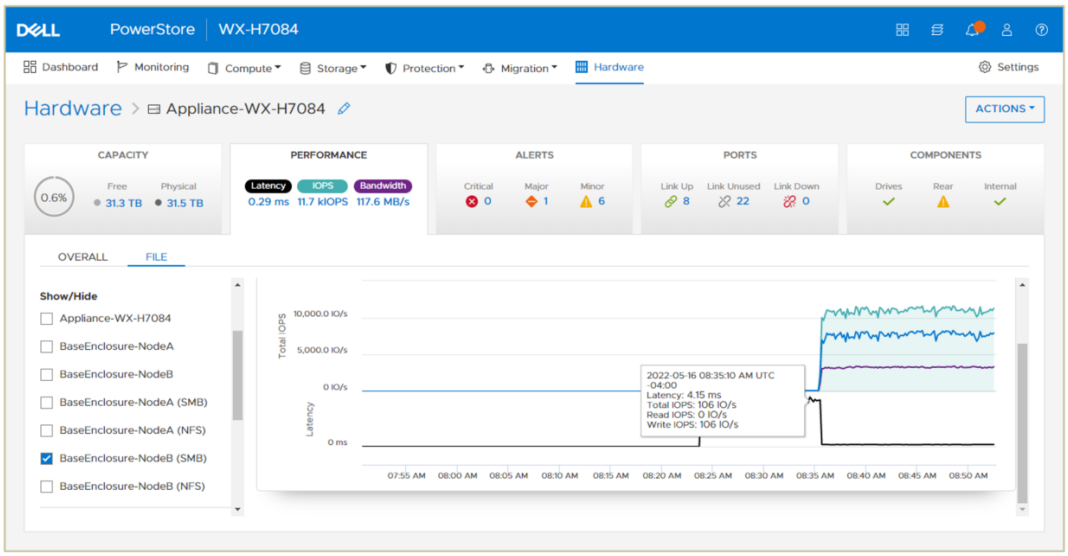


**Figure 42. SMB protocol metrics on Node B**

Starting with PowerStoreOS 3.0, additional capacity and performance metrics for file systems and NAS servers are available. As shown in Figure 43, the following file system capacity metrics are available at a 5-minute granularity:

- Thin savings
- Snap savings
- Snap/thin clone space
- File system family unique data

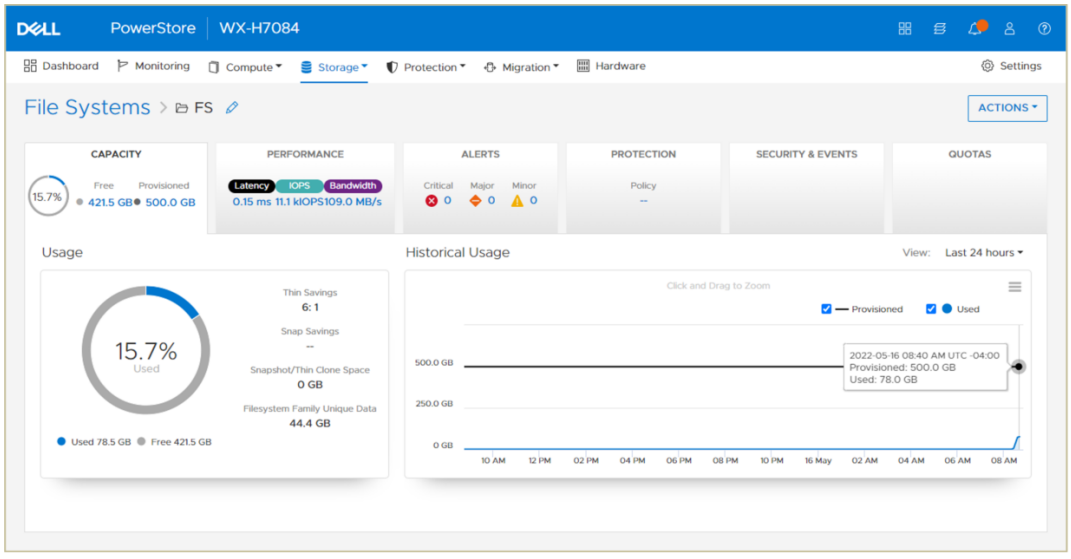


**Figure 43. File system capacity metrics**

NAS server metrics enable administrators to view aggregated data for all file systems that reside on the NAS server. As shown in Figure 44, the following NAS server capacity metrics are available at a 5-minute granularity:

- Size used

- Size provisioned

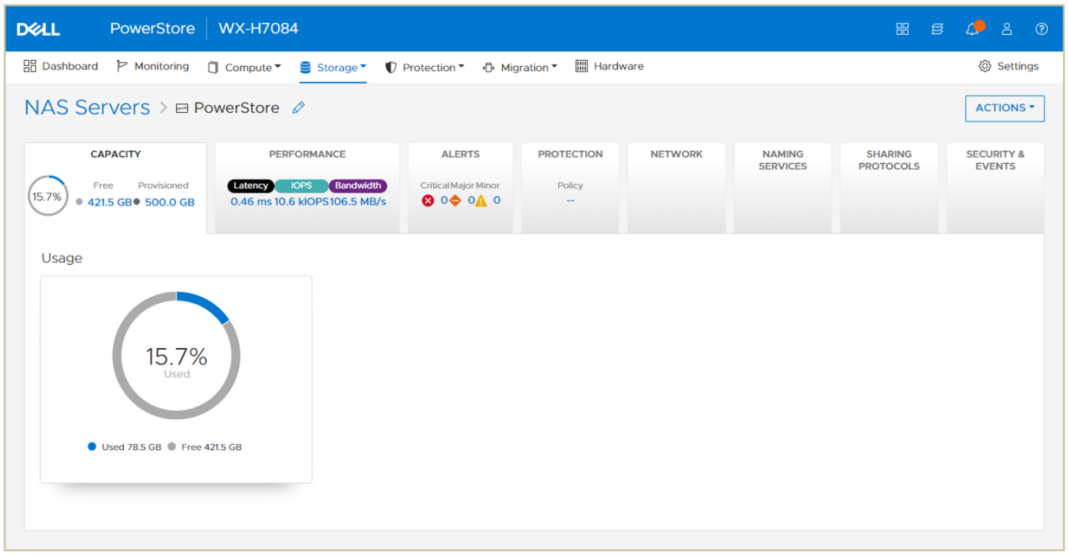


**Figure 44. NAS server capacity metrics**

As shown in Figure 45, the following NAS server performance metrics are available at a 20-second granularity:

- Read, write, and average latency
- Read, write, and average IOPS
- Read, write, and average bandwidth
- Read, write, and average IO size

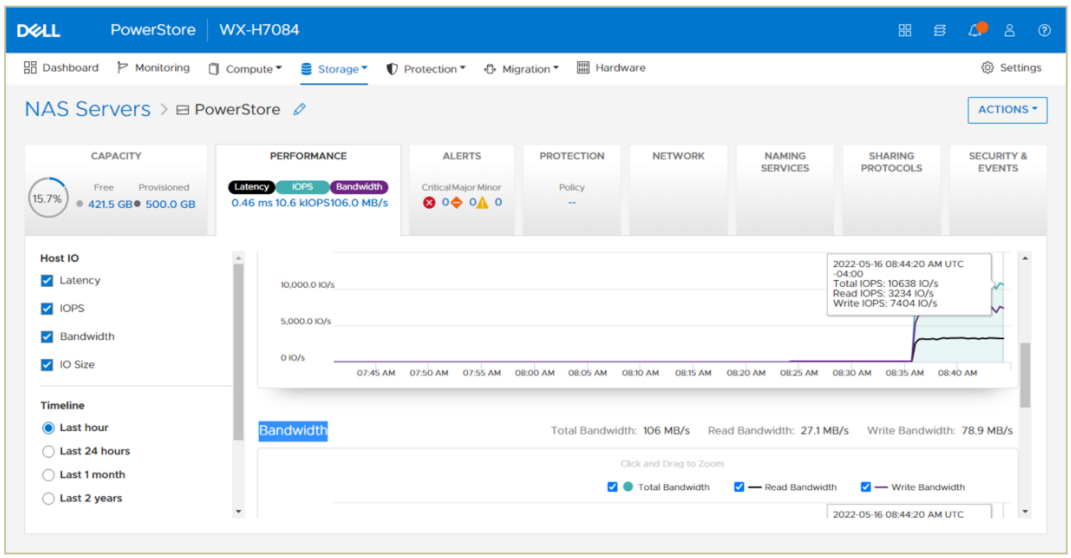


**Figure 45. NAS server performance metrics**

# Migration

### Dell Select Datadobi DobiMigrate

Datadobi, a Dell Select partner, offers the migration software DobiMigrate to perform file system migrations to the PowerStore platform. DobiMigrate is compatible with many source storage systems, including Dell storage systems and a set of third-party storage arrays. For more details, see the DobiMigrate support matrix.

DobiMigrate is run on a hypervisor supporting OVA deployment (such as VMware ESXi) or installed on a Red Hat Enterprise Linux or CentOS Linux host through an RPM. It supports NFS, SMB, and basic multiprotocol migration, with host machines known as proxies running DobiMigrate software to handle the data transfer of the migration. Management of migration sessions using DobiMigrate is performed through an intuitive UI that provides status and reporting options through each step of the migration operation.

Figure 46 shows a configuration diagram from the Datadobi document NAS and Object Migration Software for Modern Data Centers.

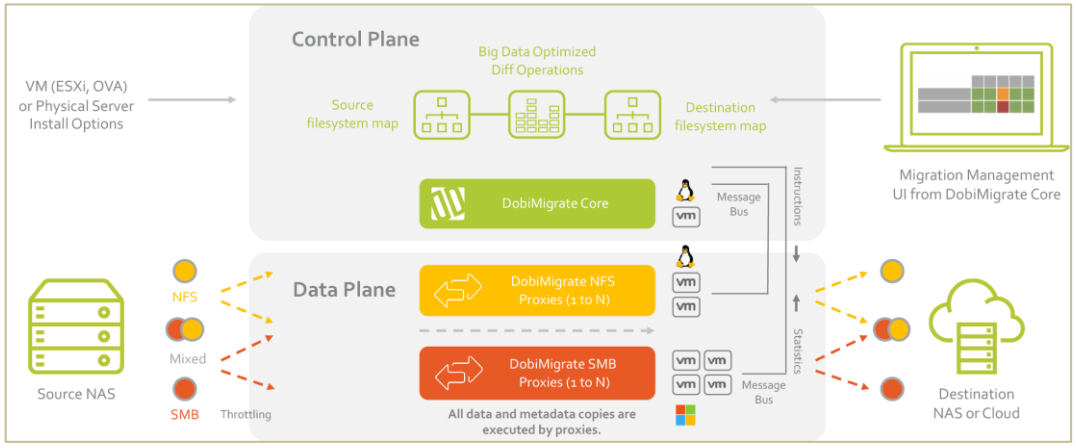


**Figure 46. DobiMigrate configuration**

More information about Datadobi DobiMigrate can be found on the Datadobi site. Information about Datadobi and its integration points with Dell storage can be found on Dell Support.

### Native file import

Starting with PowerStoreOS 3.0, native file import is available. This provides the ability to import file storage resources from VNX2, and enables migrating a Virtual Data Mover (VDM) along with its associated NFS or SMB file systems to PowerStore. The creation, monitoring, and management of the migration session is all handled by PowerStore and has a similar user experience to native block import.

For more information about native file import, see the document *Dell PowerStore: Migration Technologies* and *Dell PowerStore Importing External Storage to PowerStore* on Dell.com/powerstoredocs.

# Conclusion

With the native file capabilities available on PowerStore, administrators can easily implement a highly scalable, efficient, performant, and flexible solution that is designed for the modern data center. The rich supporting feature set and mature architecture provides the ability to support a wide array of use cases. Because these file capabilities are integrated, no additional hardware, software, or licenses are required in order to leverage this functionality. Because all file management, monitoring, and provisioning capabilities are available in the HTML5-based PowerStore Manager, administration is quick and simple. PowerStore file provides great value to environments that leverage block, file, or a mixture of both.

# Appendix: Technical support and resources

The Dell Technologies Info Hub > Storage site provides expertise that helps to ensure customer success with Dell storage platforms.

Dell.com/powerstoredocs provides detailed documentation about how to install, configure, and manage Dell PowerStore systems.