Hammerspace Administration Guide

Version 5.2 · Published 2026-11-20

For the latest version of this document, see docs.hammerspace.com/home/downloads

HAMMERSPACE

Table of Contents

About This Documentation 1

Part 1 — Introduction **2**

1. Simplifying Data Orchestration and Management 3
2. User Interfaces 4
   - 2.1. Using the Admin CLI 4
     - 2.1.1. Getting Help in the Admin CLI 5
     - 2.1.2. Common Options 6
   - 2.2. Navigating the Management GUI 6
   - 2.3. Recommended Browsers 7

Part 2 — Monitoring the Cluster **9**

3. Monitoring Cluster Health 10
   - 3.1. About the Prometheus Exporters 10
   - 3.2. Enabling or Disabling Prometheus Exporters 10
   - 3.3. Prometheus Exporter Ports 11
   - 3.4. Storing and Viewing Prometheus Metrics 11
   - 3.5. Monitoring S3 Activity 12
4. Monitoring Storage Health 13
5. Monitoring Storage Volumes 15
6. Monitoring Tasks and Events 16
   - 6.1. Viewing Events 16
   - 6.2. Configuring Event Forwarding 16
   - 6.3. Configuring External Syslog 17
     - 6.3.1. Disable and Clear Syslog Configuration 18
   - 6.4. Examples of Event Notification Emails 18
   - 6.5. Configuring Email Notifications 20
     - 6.5.1. Viewing the List of Notification Rules 21
     - 6.5.2. Viewing the Details of a Notification Rule 22
     - 6.5.3. Disabling Email Events 22
   - 6.6. Configuring Share-Capacity Notifications 23
     - 6.6.1. Example Share-Capacity Alerts 23
   - 6.7. Viewing Tasks 24

Part 3 — Managing the Cluster **26**

7. Client Data-Access Management 27
   - 7.1. Auditing Access to Data 27
     - 7.1.1. How to Configure Event Forwarding for Auditing Events 27
     - 7.1.2. Configuring a Share to Send Auditing Events 28
       - From a Windows Client Using the Microsoft Management Console 28
     - 7.1.3. Syslog Examples 30
   - 7.2. Kerberos Support for NFS 30
     - 7.2.1. Limitations 30
     - 7.2.2. NFS Protocol Support 31
     - 7.2.3. Active Directory Configuration Requirements 31
     - 7.2.4. Hammerspace Cluster Configuration with Kerberos 32

Enabling Kerberos Security on NFS Export Options 32
Using the Management GUI 32
Automatic Kerberos Export Options on the Root Share 34
Enabling the NFSv4.1 Service 35
Registering NFS Principal Names 35
In the Admin CLI 35
In the Management GUI 35
7.2.5. DNS Requirements 35
Joining Active Directory 36
7.2.6. Client Configuration 36
Mounting a Share Using Kerberos Security 38
7.3. NFS 38
7.3.1. NFSv4 Client Unique-Id Requirement 38
7.3.2. NFS Version-Specific Mount Examples 39
7.3.3. Supported NFS Protocols 39
7.3.4. NFSv4.1 ACL Passthrough on Portal Mounts 40
7.3.5. Enabling NFS LOCALIO on Linux Clients 41
7.4. Multi-Protocol Support 42
7.4.1. Multi-Protocol File Locking 42
7.5. S3 43
7.5.1. S3 Metadata 43
7.5.2. S3 Terminology 43
7.5.3. S3 Limitations 44
7.5.4. Multi-Tenancy 45
7.5.5. Managing S3 Servers 45
7.5.6. Transport Security 45
7.5.7. X.509 Server Certificate 45
7.5.8. Basic Workflow for Configuring S3 46
7.5.9. Preparing for S3 Configuration 46
7.5.10. Creating an S3 Server Using the GUI 46
7.5.11. Deleting an S3 Server 48
7.5.12. Managing S3 Buckets 49
Deleting an S3 Bucket 49
Using the Management GUI 50
Missing Bucket and Bucket Container Events 50
7.5.13. Managing S3 Users 51
Adding an Access Key to an S3 Server 51
Adding Multiple Local Users at the Same Time 53
Removing Access to an S3 Server 53
7.5.14. S3 API Overview 54
7.5.15. S3 Object Keys and the File System Namespace 55
How Object Keys Map to Files and Directories 56
Object and Prefix Name Conflicts 57
Keys That End in a Slash 59
How a Folder Appears in Listings 59

When a Folder Disappears 60
Listing and Lookup 60
Deleting Objects and Prefixes 61
Guidance for Applications and Test Suites 62
7.6. SMB 62
7.6.1. Active Directory Integration 62
Joining an Active Directory Domain 62
Leaving an Active Directory Domain 64
7.6.2. User Mapping Between Windows and Linux 65
Default Behavior 65
Mapping the UID and GID Within the Same Active Directory Domain 65
Cross-Domain Mapping 67
Creating Mappings Between Domains 67
Selecting a Preferred Domain 67
Removing a Preferred Domain 67
Applying a Preferred Domain to a Share 67
Reloading Updated Domain-Mapping Rules 67
Reviewing Cross-Domain Mappings 68
7.6.3. Client Access to Data Using SMB 68
macOS Client Access 68
Windows Client Access 70
7.6.4. Load Balancing 70
7.6.5. Mapping a Share 70
7.6.6. Mapping a Drive Letter Using the Windows CMD 71
7.6.7. SMB Multi-channel 72
Requirements 72
Enabling SMB Multi-channel with the GUI 72
7.6.8. Viewing Mapped Shares Using the Windows CMD or PowerShell 73
7.6.9. Unmounting a Share Using the Windows CMD 73
7.6.10. Supported SMB Versions 74
7.6.11. Creating Hidden SMB Shares 74
7.6.12. Setting the Browsable Option 74
Using the GUI 74
Using the Admin CLI 75
7.6.13. Setting Aliases for SMB 75
Adding Aliases to an Existing Share 76
Listing SMB Aliases 78
Removing Aliases 78
Additional Details 78
7.6.14. Allowing Access-Based Enumeration 79
Enabling ABE for an Entire Share Using the GUI 80
Enabling ABE for a Sub-folder Using the GUI 81
Enabling ABE Using Objective Expressions 82
7.6.15. Using the Administrators Group 83
Viewing Existing Members of the Administrators Group 83

Adding a New Group to the Administrators Group 83
Adding an Individual User to the Administrators Group 83
Removing Members of the Administrators Group 83
7.6.16. Windows Previous Versions 84
7.6.17. Using Microsoft Management Console 85
Managing Share-Level Permissions 85
Using the Microsoft Management Console to Set Share Permissions 85
Viewing and Closing Files 87
Viewing and Closing Sessions 89
7.7. Active Directory Permissions 90
7.7.1. Two Accounts Are Involved 90
7.7.2. Minimum Permissions: Joining with a New Computer Account 90
7.7.3. Additional Permissions by Capability 91
7.7.4. What Each Capability Means in Practice 92
7.7.5. Recommended Delegation for Most Sites 93
7.7.6. Troubleshooting 93
8. Data Management 95
8.1. At the Share Level 96
8.2. At the Directory Level 96
8.3. Anti-virus Scanning 97
8.3.1. Requirements 97
8.3.2. How It Works 97
8.3.3. Limitations 98
8.3.4. Virus Definition Updates 98
8.3.5. Performance 98
8.3.6. Adding an ICAP Server 99
8.3.7. Identifying Infected Files 99
8.3.8. Locating Infected Files by Querying Data 100
8.3.9. Remediation 101
8.3.10. Reporting 101
8.3.11. Viewing Only Top Files 102
8.3.12. Viewing Summary of Virus-Scanning Operations on a Share 102
8.3.13. Viewing Threat Status of an Individual File 103
8.3.14. Viewing Threat Status of a Share 103
8.3.15. Virus-Scanning Objective 105
8.4. Cross-Site Versioning 105
8.4.1. Configuring Cross-Site Versioning Using the Admin CLI 105
8.4.2. Configuring Cross-Site Versioning Using the GUI 106
8.5. Data Availability and Durability 106
8.5.1. The Availability Number-of-Nines Objective 107
Best Practices 107
8.5.2. Volume Availability - Default Values 107
8.5.3. Mirroring Data Between NFS Volumes 107
8.5.4. Using the Durability Number-of-Nines Objective 108
8.5.5. Volume Durability – Default Values 108

8.6. Data Profiler 108
8.6.1. Data Mobility 109
8.6.2. Mobility History Chord Diagrams 110
8.6.3. Mobility History Lists 112
8.6.4. Alignment 113
Alignment Reporting 113
Resolving Alignment Issues 115
8.7. File Clones and File Snapshots 115
8.8. File Versioning 115
8.8.1. Configuring File Versioning Using Objectives 115
8.8.2. Configuring File Versioning in the GUI 116
8.8.3. Locating Versioned Files 117
8.8.4. Restoring a Previous Version of a File 118
8.8.5. Protecting Files from Deletion 120
How Undelete Works 120
8.8.6. Configuring Undelete Using the GUI 120
8.8.7. Configuring Undelete Using the CLI 122
8.8.8. Recovering Deleted File Using the GUI 123
8.9. Managing Labels 124
8.9.1. Creating Labels 125
8.9.2. Listing Labels 126
8.9.3. Renaming Labels 126
8.9.4. Changing the Label Hierarchy 127
Move Coast Live Oak and Black Oak to Large Oaks 128
Move Shrub Oak to Small Oaks 128
List the Labels After the Change 128
8.9.5. Removing Labels 129
8.9.6. Using the Hs Command with Labels 129
Adding a Label to a File Using the Hs Command 130
Listing All Files and Their Paths with a Particular Label 130
Listing All Files with a Particular Label 130
8.10. Managing Shares 130
8.10.1. Creating a Share 131
8.10.2. Deleting a Share 133
Shares Referenced by SMB Autohome Entries 134
Using the Admin CLI 134
8.10.3. Changing the Default File Placement Behavior 135
8.11. Max Share Size Enforcement 135
8.11.1. How the Share Size Limit Is Detected 136
8.11.2. NFS Clients on a Full Share 136
NFSv4.2 Clients 136
NFSv3 Clients 137
8.11.3. SMB Clients on a Full Share 137
8.11.4. Best Practices for Max Share Size 137
8.12. Referral Shares 137

8.12.1. REST API Rules 138
8.12.2. Client Access and Transparency 138
8.12.3. Limitations 138
8.12.4. Creating a Referral Share 139
Using the Admin CLI 139
Using the Management GUI 139
8.12.5. Updating a Referral Share 139
Updating the Referral Path Using the Admin CLI 139
8.12.6. Deleting a Referral Share 139
Using the Hammerspace Admin CLI 139
Using the REST API 140
8.12.7. Additional Tasks 140
Adding or Removing Referral Addresses Using the Admin CLI 140
Replacing All Referral Addresses 140
8.13. Snapshots 140
8.13.1. Accessing Snapshots from Clients 140
Linux or macOS NFS Client 141
Windows Client 141
8.13.2. Creating an Instant Snapshot 142
8.13.3. Custom Schedules and Retention Policies 142
Custom Scheduling Examples for the Admin CLI 143
Creating a Custom Snapshot Schedule 143
Viewing Snapshot Schedules 143
Custom Retention Examples for the Admin CLI 143
Creating a Snapshot Retention-Time Policy 143
Creating a Policy for Number of Snapshot Copies to Retain 143
Creating a Share-Snapshot Schedule and Retention Policy 143
8.13.4. Deleting Snapshots Using the Management GUI 144
8.13.5. Listing Snapshots Using the Admin CLI 144
8.13.6. Restoring a Snapshot 144
8.13.7. Restoring Individual Files Using the CLI 145
8.13.8. Saved-Snapshot Location 146
8.13.9. Snapshot Schedule 146
Example Snapshot Schedules 146
Creating Snapshot Schedule in the GUI 146
Creating Snapshots Using the CLI 147
8.14. Using Objectives 147
8.14.1. Default Objectives 148
8.14.2. Predefined Objectives 148
8.14.3. Editing an Objective 150
8.14.4. Advanced Objectives 151
Common Operators 151
Common Functions for Advanced Expressions 151
8.14.5. Examples of Advanced Objectives 153
Tiering of Data from File Storage to Archive (Cloud/Object) 153

Using the Admin CLI 153
Move Snapshots to Cloud/Object Storage 153
Keep Most Recent Snapshot on Original Location 154
Keeping Live Tree on File Volumes 154
Keeping File Clones on Cloud/Object Storage 154
Placing PDF Files on Cloud/Object Storage 154
Moving Data in the Archive Directory to Cloud/Object Storage 154
Moving Undelete Files to Cloud or Object Storage 155
9. Storage Management 156
9.1. Adding Storage Capacity 156
9.1.1. Adding a Storage System 156
9.1.2. Adding a Storage Volume 156
9.1.3. Resizing a 3rd Party NAS Volume 159
9.2. Adding Storage to DSX 159
9.2.1. Part 1: Adding Storage 159
9.2.2. Part 2: Refreshing the DSX Configuration 159
9.2.3. Part 3: Creating a New Logical Volume 160
9.2.4. Part 4: Registering the New Volume 161
9.3. Capacity Management 162
9.3.1. Viewing Total Capacity 163
9.3.2. Addressing Capacity Issues 163
9.4. Removing Storage 163
9.4.1. Part 1: Identifying the Volume 163
9.4.2. Part 2: Deleting the Logical Volume (CLI Only) 164
10. User Management 166
10.1. Account Auto Locking 166
10.2. Adding Local Management User Using the GUI 167
10.3. Applying Network Restrictions 168
10.4. Clearing Network Settings 168
10.5. Deleting a User 168
10.6. Editing a User 169
10.7. Enterprise Sign On 169
10.7.1. Limitations 169
10.7.2. Setting Up Enterprise Sign On 169
10.7.3. Securing the Connection to the Identity Provider 171
10.8. Local Management Users 172
10.9. Login Policies 172
10.10. User Access to the Management Interfaces 172
10.11. User Roles 172
10.11.1. Management Roles 172
10.11.2. Data-Access Roles 173
11. Hammerspace System Management 174
11.1. Server Certificate 174
11.1.1. Installing a Certificate Using the Management GUI 175
11.1.2. Viewing and Downloading the Installed Certificate 176

11.1.3. Installing a New Certificate Using the Admin CLI 176
11.1.4. Resetting the Installed Certificate 179
11.2. System Backup and Restore 180
11.2.1. System Backup 180
Creating a System Backup Schedule 181
Creating a Scheduled Backup 181
Checking the Backup Schedule 181
Changing or Removing a Schedule and/or Changing the Backup Location 182
Listing System Backups 182
11.2.2. Restoring System from Backup 183
11.3. Upgrading the Product 184
Appendices **185**
Appendix A: Technology Preview 186
A.1. Connecting Storage to a DSX Using NVMe-oF 186
A.1.1. Adding an NVMe-oF Enclosure and Subsystems 187
Configuring the Logical Volumes 192
A.1.2. Admin CLI Command: nvmeof-config 194
Examples 195
A.1.3. Admin CLI Workflow to Connect NVMe-oF Enclosures 196
A.1.4. Additional Admin CLI Examples 201
Listing Enclosures and Connections 201
Adding Targets 203
Changing an Enclosure's Connection Settings 204
Connecting Subsystems 205
Disconnecting Subsystems 206
Removing an NVMe-oF Enclosure 206
Creating a Logical Volume 207
A.1.5. NVMe-oF Path Health and Recovery 208
Path Health Events 208
How the Events Behave 209
Responding to a Path Health Event 210
SNMP Traps 210
Connection State Is Not Path Health 210
Controller Loss Timeout 212
Why -1 Is Recommended 212
Checking the Current Value 212
Setting the Value 213
A.1.6. NVMe-oF Limitations and Operating Guidance 213
Presenting Storage to DSX Nodes 213
The DSX Host NQN 214
Connecting and Disconnecting 214
A.1.7. Upgrading a Cluster That Uses NVMe-oF Storage 214
Before You Upgrade 215
After You Upgrade 215
Appendix B: Event Listing 216

Appendix C: Hammerspace Terminology 229
Appendix D: Additional Documentation Resources 232

About This Documentation

This guide describes how to administer and manage a Hammerspace system after it has been installed and licensed. It covers the management interfaces, infrastructure monitoring, client data-access protocols (NFS, SMB, S3), data management features, storage management, user administration, and system maintenance tasks such as backup, restore, and upgrades.

Technology preview features — including NVMe-oF storage connectivity — are documented in a dedicated section.

This guide is intended for system administrators responsible for the day-to-day operation of a Hammerspace deployment. Readers should be familiar with general storage and networking concepts. For initial installation and licensing, see the *Hammerspace Installation and Licensing Guide*.

To open a support case, use the Hammerspace Support Hub. To download Hammerspace software and licenses, use the Hammerspace Licensing and Delivery Portal.



Do not install additional or third-party software, agents, or packages directly on Hammerspace Anvil (metadata) or DSX (data) nodes. These nodes run a managed appliance software stack; anything installed outside that stack is unsupported and is removed during a software upgrade. Such software can also interfere with node operation. If you need monitoring or integration with an external tool, contact Hammerspace Support for a supported approach.

© 2026 - Hammerspace

Part 1 — Introduction

The Hammerspace product (the Product) is a software-defined platform that provides a simple approach to manage, migrate, and present data across on-premises and hybrid-cloud data centers. The documentation in this *Admin Guide* will enable administrators to effectively maintain, monitor, and troubleshoot the components of your Product infrastructure.

© 2026 - Hammerspace

Chapter 1. Simplifying Data Orchestration and Management

Hammerspace simplifies data management for system administrators in several ways:

- **Unified data access:** Hammerspace provides a global view of files across different storage systems, simplifying access for users and administrators. Users can access data using standard protocols without needing additional client software, which streamlines workflows.
- **Centralized metadata management:** By assimilating metadata from existing storage, Hammerspace creates a unified control plane. This separation of metadata from the underlying infrastructure allows administrators to manage data without disrupting user access.
- **Objectives-based automation:** Hammerspace uses *objectives* to automate data orchestration. Administrators can set objectives to control the most granular details of data placement and behavior, enabling the system to automatically manage data according to these rules.
- **Efficient data operations:** Hammerspace streamlines various data operations through its GUI and CLI, including share management, objective management, storage and volume management, snapshots, and networking. For example, administrators can easily create, modify, and delete shares; apply objectives to manage data placement; and manage storage volumes and snapshots.

© 2026 - Hammerspace

Chapter 2. User Interfaces

You can manage Hammerspace clusters through the command line, GUI, and REST API, although API functionality is limited. All functionalities are available through the CLI and GUI. Some of the procedures in this guide are documented using the GUI and others are documented using the Admin CLI. The method documented is generally based on what we think is the simplest way to achieve the result.

This section includes the following subsections:

- Using the Admin CLI
- Navigating the Management GUI
- Recommended Browsers

2.1. Using the Admin CLI

Hammerspace provides a custom, administration Command Line Interface (*Admin CLI*, or CLI) with a restricted shell and a set of administration-specific commands.

The Admin CLI shell and commands provide auto completion and built-in help. The Admin CLI does not provide access to bash, other shells, or common Linux commands. The CLI provides a history function that is accessible by using the up and down arrow keys.

The CLI can be accessed by logging on as **admin** on the physical or remote console of a bare-metal server, on the virtual console of a VM, or by SSH to the IP address, hostname, or FQDN of the management IP of the cluster.

```
ssh admin@<server-ip-address>
```

In cloud deployments, the username and authentication protocol are specific to the cloud.

Table 1. Supported Clouds

Platform	Admin username
ISO-based installs (bare-metal, VMware, Hyper-V, KVM)	admin
Amazon AWS	ec2-user using Key Pair or IAM
Microsoft Azure	admin
Google Cloud	admin



The password for the cluster is set during installation.

© 2026 - Hammerspace

2.1.1. Getting Help in the Admin CLI

The following tips will help you use the *Admin CLI* more easily:

- Typing a question mark alone on the command line will display a list of all available commands and a brief description of each one. For example, **admin@anvil> ?**
- Typing part of a command name followed by **Tab** or **Space** will list possible matching commands. If there is only one match, pressing **Tab** or **Space** will complete the command.
- Pressing **Enter** after typing part of a command will list possible matches if there are more than one, or complete and execute the command if there is only one match.
- Typing part of a command followed by **?** will list possible matches along with a brief description of each matching command.
- The **help** command gives general help for using the CLI and editing commands.

Hammerspace Admin CLI: Output of help command

```
admin@anvil> help

CONTEXT SENSITIVE HELP

[?] - Display context sensitive help. This is either a list of possible

command completions with summaries, or the full syntax of the

current command. A subsequent repeat of this key, when a command

has been resolved, will display a detailed reference.

AUTO-COMPLETION

The following keys both perform auto-completion for the current command line.

If the command prefix is not unique then the bell will ring and a subsequent repeat of the key will
display possible completions.

[enter] - Auto-completes, syntax-checks then executes a command. If there is

a syntax error then offending part of the command line will be

highlighted and explained.

[space] or [tab] - Auto-completes, or if the command is already resolved inserts a space.

USEFUL KEYS

[CTRL-A] - Move to the start of the line

[CTRL-E] - Move to the end of the line.
```


© 2026 - Hammerspace

```
[CTRL-C] - Delete and abort the current line

[CTRL-D] - Delete the character to the right on the insertion point.

[CTRL-K] - Delete all the characters to the right of the insertion point.

[CTRL-U] - Delete the whole line.

[backspace] - Delete the character to the left of the insertion point.
```

- Hammerspace commands include built-in, context-sensitive help. Typing the command name followed by **?** or `--help` will list options and their descriptions.
- You can even use **?** or `--help` after entering some parameters. Using **?** in this way will return you to the partially completed command line. Using `--help` will return to a blank command prompt, but clicking the up arrow will retrieve the partially completed command.
- The CLI gives hints as to syntax errors when typing or editing commands, especially when complete or partially complete with options and parameters. If you are unable to type a space where one is needed, that means that there is a syntax error somewhere else in the command, such as a space missing earlier in the parameters. Values for parameters containing spaces and some other special characters must escape the space with a backslash (****) before each space, or by putting the whole parameter in single or double quotes.

2.1.2. Common Options

- `--name` is the management name of an managed object, such as a share, volume name, storage system name, or objective. When creating or adding most objects, the name is arbitrary in that it does not have to match the object's actual name. For example, you can name a storage system something other than its host name or FQDN. For some objects, the name can be changed arbitrarily however some objects such as product node names cannot be changed after adding them.
- `--id` is a UUID for an object.
- `--internal-id` is a short, sequentially assigned numeric identifier for an object. These are useful when copying and pasting a lengthy UUID is not possible, such as in a remote desktop or virtual console session.
- Commands that operate on an management object require that the **name**, **id**, or `internal-id` option be specified.

2.2. Navigating the Management GUI

The Management Graphical Interface (GUI) is accessible by entering the management cluster IP address or DNS name into a supported web browser.

Hammerspace includes the following management categories and pages. You can access each of these pages by clicking the icon on the left navigation bar.

Table 2. Navigation Pages

© 2026 - Hammerspace

Navigation Menu	Description
Dashboard	Summarized information about data or infrastructure, as well as Quick Links to areas of interest.
Mobility	Data mobility visualization
Data	View and create shares, add storage systems, volumes, S3 servers and buckets.
Objectives	View and create objectives that define how data is placed and moved. Profile the cost of data placement by age.
Infrastructure	View detailed performance and alignment data for storage volumes, volume groups, and storage systems.
Administration	View clusters and nodes, S3 servers, licensing, SNMP configuration, Active Directory integration; manage users, download support logs, and update software.

The following general administrative areas are accessible from the status bar at the top of each page:

Table 3. Status bar functions

Task	Description
Getting Started	Common tasks for configuring the product.
Events	View events.
Tasks	View the status and progress of tasks.
Users	Logout; view user details; change the user password; API documentation, view information about Hammerspace version.
Errors	The triple-bar icon at the top right of the page displays a list of recent errors from UI components.
Expand menu	The triple bar icon at the top left of the page expands the left side panel.

2.3. Recommended Browsers

The following browsers are supported for accessing the Hammerspace GUI.

Table 4. Supported browsers

Browser	OS	Version
Chrome	Linux, macOS, and Windows	Version 115 and later
Firefox	Linux, macOS, and Windows	Version 110 and later

© 2026 - Hammerspace

Browser	OS	Version
Safari	macOS	Version 17 and later

© 2026 - Hammerspace

Part 2 — Monitoring the Cluster

The following sections describe the ways to monitor and maintain the Hammerspace product (Product) infrastructure.

Monitoring the Product includes the following tasks:

- Monitoring the health of the cluster
- Monitoring the health of DSX storage node
- Monitoring the health of DSX portal node

Events are sent automatically when node health changes.



It is important to configure the desired event channel to make sure nothing is missed.

The GUI also shows the health of the cluster. You can go to the **Administration** › **System** page to check for overall cluster health.

© 2026 - Hammerspace

Chapter 3. Monitoring Cluster Health

The local drives on a product node are automatically monitored for space utilization. There is a WARNING event at 70% and an EXCEEDED event at 90%. This is done by a space monitoring service, which checks all the local filesystems (e.g. **/, /boot, /var** etc) every two minutes. This runs on All Anvil and DSX nodes.

It is worth noting that the Prometheus exporters will also collect this information and it can be viewed graphically with Grafana.

Here is an example of the event text:

Example of event text

```
____
ID: d0719f4c-b1ca-11ef-9bd4-30e171639c80

Created: 2024-12-03 23:03:31 UTC

Modified: 2024-12-03 23:03:31 UTC

Type: DISK_USAGE_LIMIT_EXCEEDED

Description: Disk usage limit EXCEEDED on dsx1.example.com:/boot with only 106M available.

Severity: ALERT
____
```

3.1. About the Prometheus Exporters

Hammerspace ships integrated Prometheus exporters. They export data about the system, the resource utilization of the system, and many counters related to the data services Hammerspace provides — particularly file-system protocols and mobility. The exporters are disabled by default.

Hammerspace does not provide a Prometheus server. You supply your own, and configure it to scrape the Hammerspace cluster periodically to collect the counters.

To visualize the data, Hammerspace publishes dashboards for Grafana. These dashboards read from a Prometheus data source and require no further configuration once that source is in place.

3.2. Enabling or Disabling Prometheus Exporters

Use the following command and options to either enable or disable Prometheus exporters in Hammerspace.

1. Log in to the Admin CLI.
2. Run the following command to enable Prometheus exporters:

   Command:

© 2026 - Hammerspace

```
cluster-config --prometheus-exporters-enable
```

Expected output (the first lines of the cluster configuration; the **Prometheus exporters** field shows the new state):

```
ID:                                 f976411d-c581-4d56-a48a-2d8c70b13b60
Name:                               AEEPQ7Z5Z3487K
State:                              Standalone
Management IPs:                     [192.0.2.10/20]
Data IPs:                           [192.0.2.10/20]
Cluster floating IPs:               [192.0.2.10/20]
Since:                              2026-09-23 21:17:03 UTC
Created:                            2026-09-23 21:16:42 UTC
Timezone:                           UTC
GFS participant max suspected time: 30 minutes
Metered billing:                    Disabled
Prometheus exporters:               Enabled
Evaluation expiration:              2026-10-23 21:16:42 UTC
<Additional output removed for brevity>
```

Alternatively, to disable the exporters, use the same command but with the `prometheus-exporters-disable` option:

```
cluster-config --prometheus-exporters-disable
```

Either command prints the cluster configuration, and the **Prometheus exporters** field in that output shows the new state. To confirm the current state later, run `cluster-config` with no options and check the same field.

3.3. Prometheus Exporter Ports

Remote monitoring with Prometheus requires network connectivity between the Prometheus server and the Hammerspace Anvil and DSX nodes. The exporters listen on the following TCP ports:

Node	Protocol	Ports
Anvil	TCP	9100 – 9103
DSX	TCP	9100 and 9105



Port 9104 is not used by the Hammerspace exporters.

3.4. Storing and Viewing Prometheus Metrics

The Hammerspace exporters expose metrics; they do not store them. To retain and visualize metrics, run a Prometheus server that scrapes the cluster, and point Grafana at that server as a data source.

© 2026 - Hammerspace

Hammerspace publishes a set of Grafana dashboards for this purpose:

- Hammerspace Grafana Dashboards — downloading, installing, and configuring the dashboards and the related components.

The following third-party documentation is not published by Hammerspace, but is useful when setting up the surrounding components:

- Monitoring Linux host metrics with the Node Exporter — installing Prometheus exporters on Linux nodes.
- DCGM Exporter — using Prometheus and Grafana with NVIDIA GPU nodes.

3.5. Monitoring S3 Activity

When an S3 server is created and used, each DSX node keeps an in-memory summary of the S3 requests it serves. This summary is read directly from the node and is separate from the Prometheus exporters.

To view the S3 summary metrics for a single DSX node:

1. Obtain the hostname or IP address of the DSX node, using the `node-list` command or the **Infrastructure > Storage Systems** page in the Management GUI.

   ```
   node-list
   ```

2. In a supported web browser with network access to the DSX node, go to `<dsx-node>/s3/metrics`, where `<dsx-node>` is that node's hostname or IP address. For example:

   ```
   https://gfs-dads1.company.internal/s3/metrics
   http://10.20.43.201/s3/metrics
   ```

The summary reports, for each S3 action that has been requested:

- the number of times the action was requested;
- the client-side error count (HTTP 400–499) and the server-side error count;
- the server-side duration of the action, grouped into predefined ranges;
- the slowest request for that action since the S3 service last started, with the client IP address that request came from.

Distributions differ from node to node, depending on each node's hardware and its share of the S3 workload.



These counters are held in memory and accumulate only while the S3 service is running on that node. They are cleared whenever the S3 service restarts — during a software update, for example, or when the node reboots.

© 2026 - Hammerspace

Chapter 4. Monitoring Storage Health

Hammerspace enables administrators to efficiently and intelligently manage all storage resources under management. The health of storage volumes can be quickly assessed from the **Storage Systems** tab in the **Infrastructure** page of the GUI:

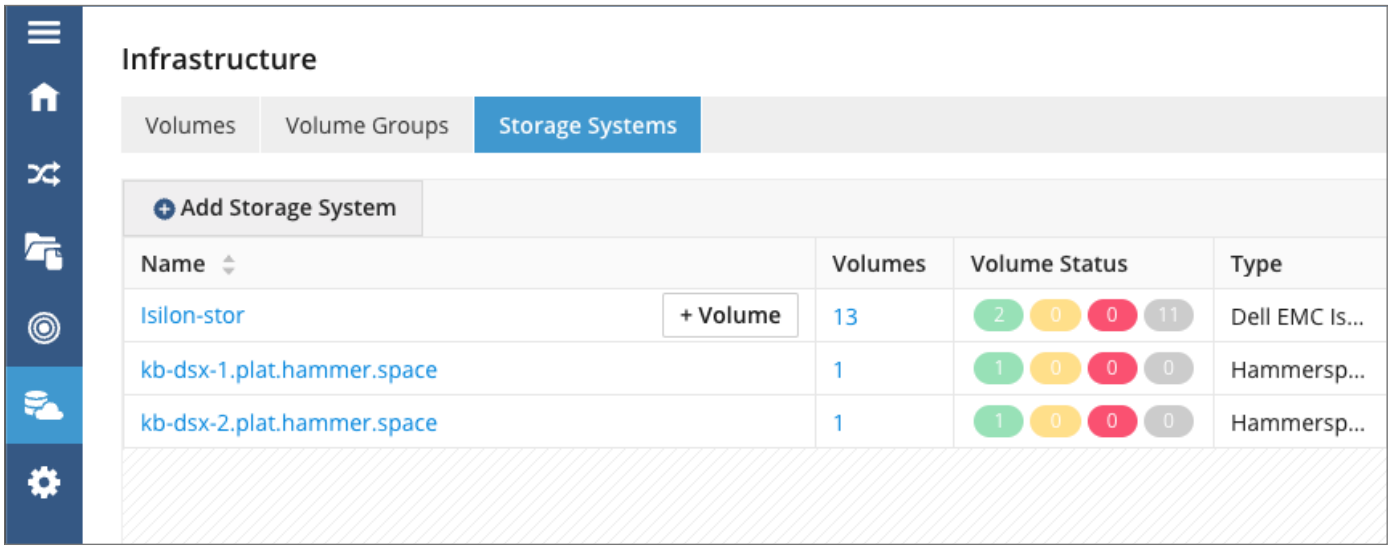


Figure 1. Viewing storage-system status

Volume-status icon colors indicate the following health status:

- Healthy volumes are indicated by a green icon.
- Degraded volumes are indicated by a yellow icon. A volume is degraded if the latency increases to levels that might cause data to be non-aligned with their objectives.
- Unhealthy volumes are indicated by a red icon. A volume is unhealthy when Hammerspace and clients cannot access it and/or when data cannot be written to it.
- Unregistered volumes are indicated by a gray icon.

Clicking on a storage system or volume name in the **Infrastructure** view provides information about capacity, performance, and alignment, which can be useful for troubleshooting and resolving issues.

© 2026 - Hammerspace

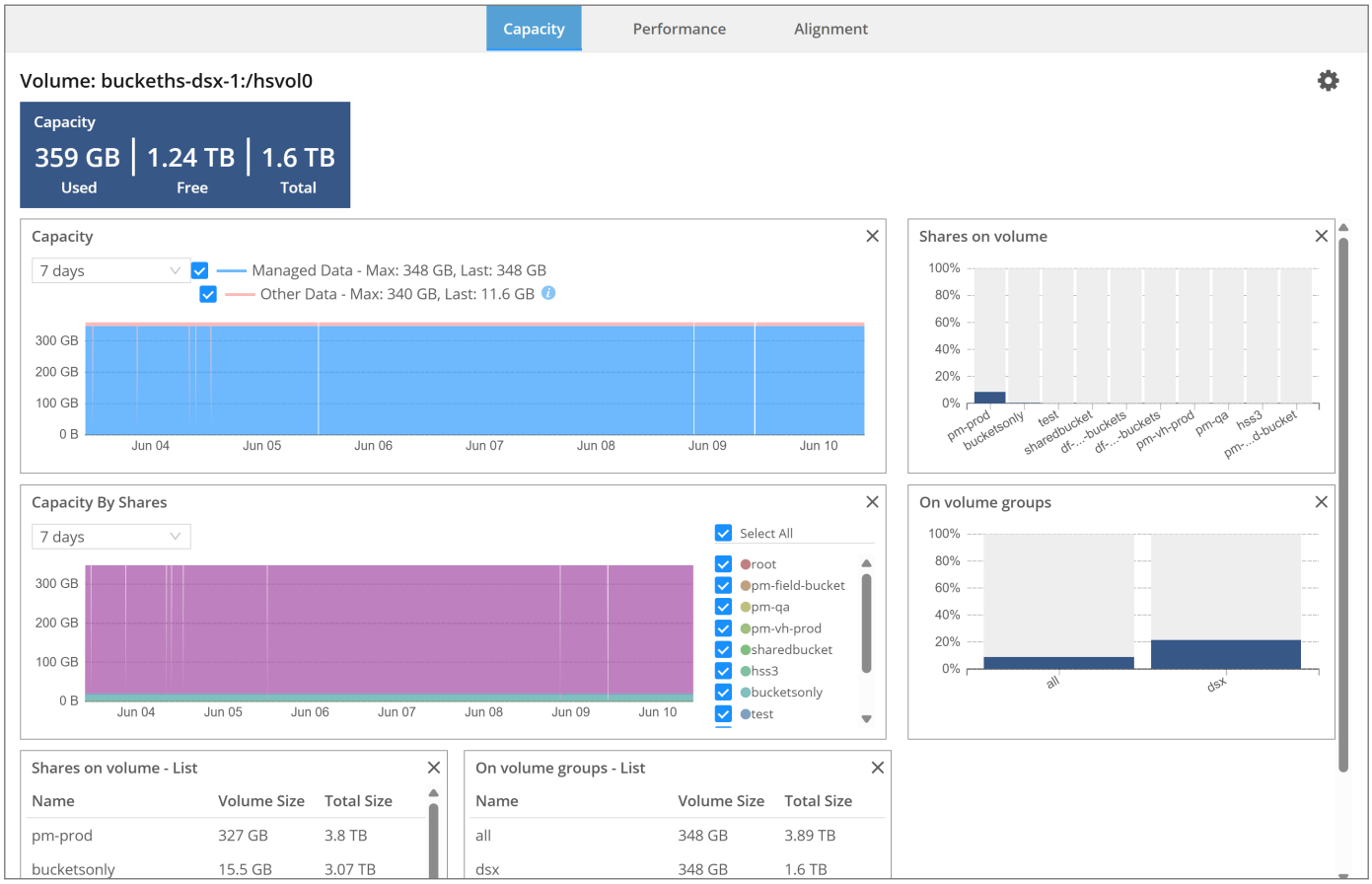


Figure 2. Viewing infrastructure volume capacity

© 2026 - Hammerspace

Chapter 5. Monitoring Storage Volumes

Hammerspace has built-in, automatic health checks that periodically check the health of connected storage systems and volumes, as well as permissions and connectivity. This check is also executed when a storage volume is added to the configuration.

The storage configuration test for NAS storage volumes checks for all the required NFS operations that are required for a read-write storage volume. Permissions (chmod) changes, ownership (chown) as well as root squashing. If an error is detected, the storage configuration test will automatically place that volume into a suspect state and generate an alert.

The storage configuration test for NAS also includes operations such as creating a directory and creating files. This ensures that files can be created with the appropriate permissions and identities.

The storage health check for object storage volumes will update a will ensure connectivity to the object storage volume periodically. Part of managing your object storage volume is ensuring connectivity and correct permissions to read and write to the bucket. This is checked by uploading an object into the bucket from each site. If any error is encountered, an event will be generated to notify the administrator that it needs attention.

© 2026 - Hammerspace

Chapter 6. Monitoring Tasks and Events

6.1. Viewing Events

You can view an events log in the Management GUI.

1. Select **Events** at the top of the browser window. The GUI displays the events list, in chronological order, from the most recent event.

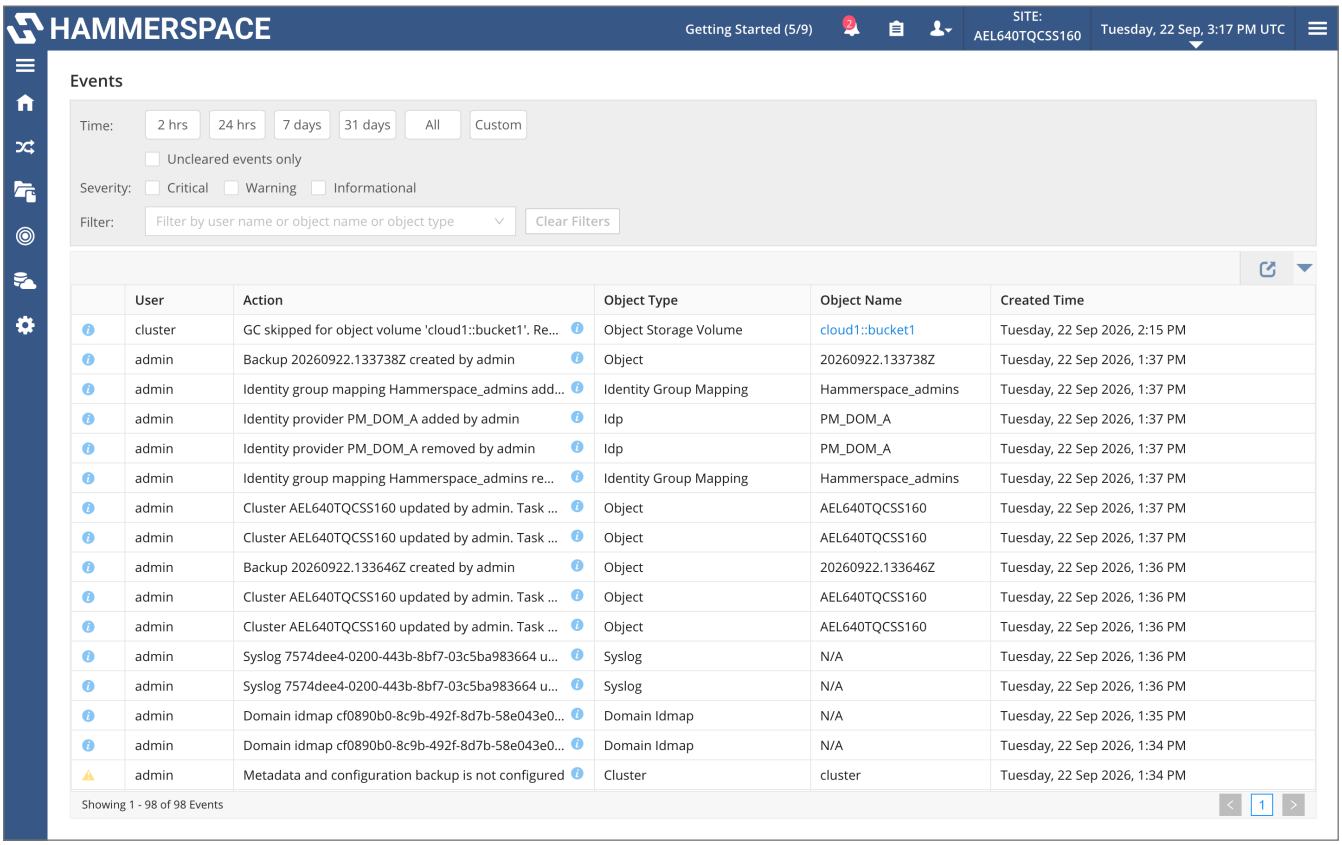


Figure 3. Viewing the Events list

2. Click **View All Events** to choose how far back to view the events.
3. Optionally, use the **Filter** window on the left side of the page to select the events you want to view. You can choose a **Time** frame, one or more types of events from the **Type** list, and one or more severities of events from the **Severity** list:
    - **Informational** — Shows information about a non-critical event, such as, "node added."
    - **Warning** — Shows information about a warning, such as, "volume is 90% full."
    - **Critical** — Shows information about a critical event, such as, "out of space".

6.2. Configuring Event Forwarding

In Hammerspace, events can be forwarded to other sources using syslog or emails. The emails contain the same event information that is shown in the Management GUI or CLI; however, the formatting of the event is

© 2026 - Hammerspace

slightly different.

Events are sent immediately and not on a schedule; nor are they batched together, so ordinary event types might send several events as they happen, several times; while in the product GUI, those events may just be shown once because they are repeated, and the date of the event reflects when it last happened.

For syslog, you can forward product events and alerts and send file system audit events over a syslog stream. The file-system audit events are only available over syslog and not available over any other event mechanism.

6.3. Configuring External Syslog

The system can send events to external syslog server(s) to aid in troubleshooting and log analysis. It is supported to send product events as well as file system auditing events via syslog. The RELP protocol transport for syslog is supported for file system auditing events. See Auditing Access to Data for additional information.

Syslog messages can be configured to be sent via TCP and UDP to an IP address or a hostname. If using a hostname, DNS must be configured.

Perform the following steps to configure and enable sending events to a syslog server:

1. SSH into the management CLI as an admin user.

   Command:

   ```
   # ssh admin@<cluster-management-address>
   ```

2. Enter the IP address, port, protocol, and event type to use for each server. Use `--server` for each syslog server you want to add. The example below adds two servers; the first server only receives the events, while the second server receives both events and file system auditing events.

   Command:

   ```
   syslog-config --server 192.0.2.50,514,tcp,event --server 192.0.2.51,514,tcp,event|filesystem --enable
   ```

3. Check the configuration.

   Check syslog configuration

   Command:

   ```
   syslog-config --view
   ```

   Expected output:

   ```
   Enabled:                true
   ```


© 2026 - Hammerspace

```
Servers:
                        Address:                192.0.2.50
                        Port:                   514
                        Transport:              TCP
                        Message types:          [Event]

                        Address:                192.0.2.51
                        Port:                   514
                        Transport:              TCP
                        Message types:          [Event, Filesystem]
```

4. To edit the configuration, repeat step 2 with the updated configuration. In this example, we removed the first server.

   Command:

   ```
   syslog-config --server 192.0.2.51,514,tcp,event|filesystem --enable
   ```

5. Check the configuration.

   Check syslog configuration

   Command:

   ```
   syslog-config --view
   ```

   Expected output:

   ```
   Enabled:                true
   Servers:
                           Address:                192.0.2.51
                           Port:                   514
                           Transport:              TCP
                           Message types:          [Event, Filesystem]
   ```

6.3.1. Disable and Clear Syslog Configuration

The disable option will clear the syslog configuration and stop events from being propagated via syslog:

```
syslog-config --disable
```

6.4. Examples of Event Notification Emails

Share-create event notification example:

© 2026 - Hammerspace

Share-create event notification text

```
<pdevent>

<uuid>5d52f2f9-670f-42d5-a007-1a0f65e2c88c</uuid>

<timestamp_utc>2025-06-19 16:21:40</timestamp_utc>

<timestamp_utc_offset>+0000</timestamp_utc_offset>

<systemId>7d170b64-c472-453d-a627-94e9f8f363c7</systemId>

<severity>INFORMATIONAL</severity>

<code>ADDED</code>

<description>share Apps added by admin</description>

<user>admin</user>

<originator>Uoid [uuid=621bccf9-cecb-414d-8f2b-7323d2110bd3, objectType=SHARE]</originator>

<technicianatwork>No</technicianatwork>

</pdevent>
```

Critical-event notification example:

Critical-event notification text

```
<pdevent>

<uuid>9e7e6124-599f-4ce2-b480-99986ff374c4</uuid>

<timestamp_utc>2025-06-19 16:51:26</timestamp_utc>

<timestamp_utc_offset>+0000</timestamp_utc_offset>

<systemId>7d170b64-c472-453d-a627-94e9f8f363c7</systemId>

<severity>CRITICAL</severity>

<code>MISSING_STORAGE_VOLUME</code>

<description>No writable backend storage volumes. At least 1 writable volume must be added before using
the namespace.</description>

<user>admin</user>

<originator>Uoid [uuid=7d170b64-c472-453d-a627-94e9f8f363c7, objectType=CLUSTER]</originator>
```


© 2026 - Hammerspace

```
<technicianatwork>No</technicianatwork>

</pdevent>
```

6.5. Configuring Email Notifications

Events can be sent via email. The event payload is in XML format. See example notification below.

Enabling email notifications requires the following actions, which are described in greater detail in this section:

Part 1: Configuring the email server

Part 2: Creating a local user within the cluster

Part 3: Adding notification rules

Part 1: Configuring the email server

This step must be performed using the Admin CLI. The configuration can be set and updated with the same command `email-config`. A single email server can be configured.

Command:

```
email-config --from-address admin@mycompany.com --host my.mail.server --port 25 --ssl
```

Optionally, a user and password can be configured if the email server requires credentials.

Part 2: Creating a local user

Continue using the CLI with the `user-create` command or switch into the GUI and navigate to **Administration › Users**. Only the viewer role is required for users created for the purpose of viewing notifications.

A local user with a correct email address must be created for every email destination where notifications are sent. For example, emails can be sent to different addresses depending on the severity of the event. The `--password` option is required; passwords must be 8–20 characters with at least one lowercase letter, one uppercase letter, one digit, and one non-alphanumeric character.

Creating users for email events

Command:

```
user-create --email info@example.com --mgmt-role-name viewer --username info_events --password <password>
```

Expected output:

```
Name:                   info_events
```


© 2026 - Hammerspace

```
ID:                     8e38eb44-fdb7-4a58-8f7e-93c230f717f8
UID:                    1304
GID:                    100
Email:                  info@example.com
Enabled:                true
Management role:        viewer
```

Command:

```
user-create --email alert@example.com --mgmt-role-name viewer --username alert_events --password
<password>
```

Expected output:

```
Name:                   alert_events
ID:                     ecc55da8-86f4-4fc6-9652-c07847e5a70a
UID:                    1305
GID:                    100
Email:                  alert@example.com
Enabled:                true
Management role:        viewer
```

Part 3: Adding notification rules

Create one or more notification event rules that determine what type of notification is sent to where. A notification event rule can be configured with multiple users but only a single threshold. To receive notifications on all thresholds, an event rule must be configured for each threshold.

The thresholds are: INFORMATIONAL | NOTICE | WARNING | ERROR | CRITICAL | ALERT | EMERGENCY.

Example of sending INFORMATIONAL events to a single destination:

```
notification-rule-create --name info_events --format XML --threshold informational --users info_events
```

Example of sending ALERT events to two destinations:

```
notification-rule-create --name alert_events --format XML --threshold alert --users
info_events,alert_events
```

6.5.1. Viewing the List of Notification Rules

To list all the configured notification rules, use the `notification-rule-list` command. The following example shows the command and output:

Viewing list of notification rules

© 2026 - Hammerspace

Command:

```
notification-rule-list
```

Expected output:

```
total 2
ID:                     bbfc6e61-6c35-46f0-889f-10f2a91be116
Name:                   info_events
Threshold:              INFORMATIONAL
Format:                 xml

ID:                     e3c2ad7f-5c7b-42da-a7f1-e241c017f0eb
Name:                   alert_events
Threshold:              ALERT
Format:                 xml
```

6.5.2. Viewing the Details of a Notification Rule

To view the details of a notification rule, use the `--name` option. The following example shows the command and output:

Viewing details of a notification rule

Command:

```
notification-rule-list --name info_events
```

Expected output:

```
ID:                     bbfc6e61-6c35-46f0-889f-10f2a91be116
Name:                   info_events
Threshold:              INFORMATIONAL
Format:                 xml
Users:
                        [Name: info_events, ID: 8e38eb44-fdb7-4a58-8f7e-93c230f717f8, Enabled: true]
```

6.5.3. Disabling Email Events

To disable email events, remove the notification rule(s) for the relevant email events. The following example shows the command and output:

Disabling email events

```
notification-rule-remove --name info_events
```


© 2026 - Hammerspace

```
Success
```

6.6. Configuring Share-Capacity Notifications

When configuring a share size, the system will automatically send a notification when it reaches the defined threshold. The threshold is set or edited using the share-create and share-edit dialogs, respectively.

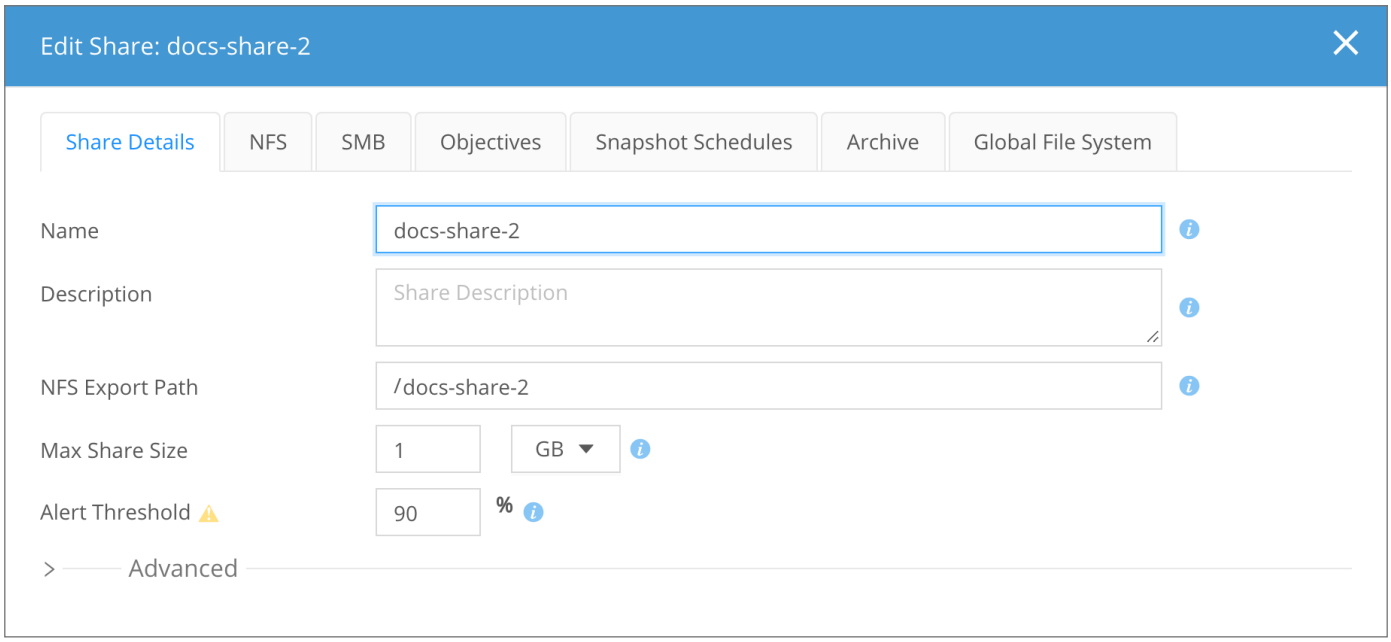


Figure 4. Setting share size and alert threshold

Share capacity events are sent out with a threshold WARNING across all configured event destinations.

Two events report share capacity:

- `SHARE_QUOTA_WARNING` (WARNING) is raised when usage reaches the Alert Threshold.
- `SHARE_QUOTA_EXCEEDED` (ALERT) is raised when usage reaches 100 percent of the Max Share Size. The warning event is cleared at the same time.

When usage falls below 100 percent but is still at or above the Alert Threshold, the exceeded event is cleared and the warning event is raised again. Both events are cleared when usage falls below the Alert Threshold.

A share can hold more data than its Max Share Size. For how the limit is enforced and what clients see when a share is full, see Max Share Size Enforcement.

6.6.1. Example Share-Capacity Alerts

	User	Action	Type	Impacted	Time
⚠	cluster	Share quota is running out, crossed 50%	share	Apps	Monday, 19 Jul, 5:19 PM

Figure 5. Warning notification when capacity threshold is reached

Share-capacity notification in the CLI

© 2026 - Hammerspace

The following example shows the command and output:

Share-capacity WARNING notification

Command:

```
event-list --type SHARE_QUOTA_WARNING
```

Expected output:

```
total 1
ID:                     4dbc2ecc-576d-4466-8823-9c0ec58c8135
Created:                2026-09-21 23:48:35 UTC
Modified:               2026-09-21 23:48:35 UTC
Type:                   SHARE_QUOTA_WARNING
Description:            Share quota is running out, crossed 90%
Severity:               WARNING
Source type:            Share
Source ID:              7be8926e-4e81-464b-86e0-6c0701df9a47
Source name:            docs-share-2
Cleared:                false
```

6.7. Viewing Tasks

Administrators can view task activity via the task log. Complete the following steps to view the task log:

1. Click the **Tasks** (clipboard) icon at the top of the browser window. The Management GUI displays the current running task list.
2. Click **View All Tasks** to open the Tasks page, which lists your own tasks (**My Tasks**) and every user's tasks (**All Tasks**).

© 2026 - Hammerspace

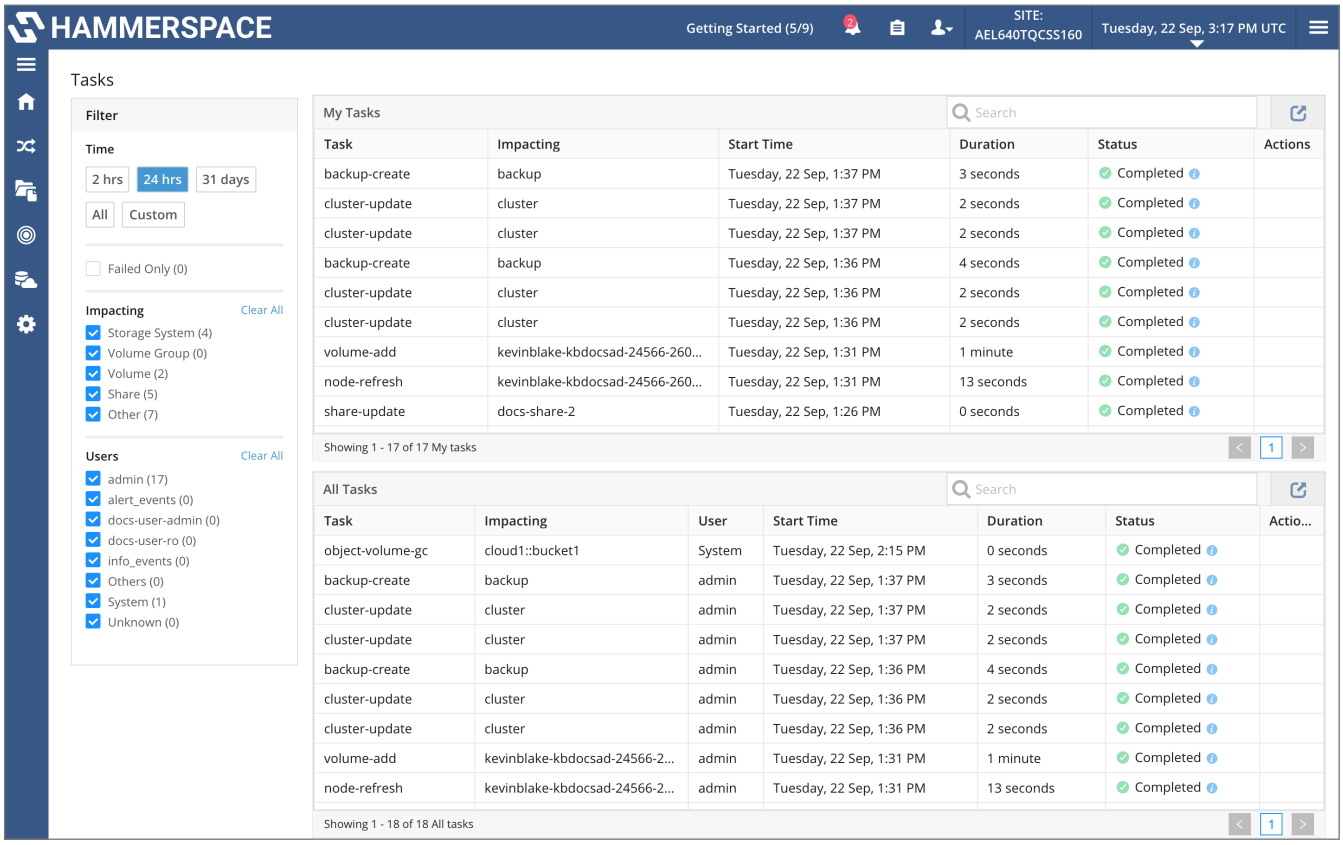


Figure 6. Viewing task activity in the task log

The task list displays the following information about activity in chronological order, from most recent to least:

- Task: The name of the task that occurred.
- Impacting: What the task impacted.
- User: The user who started the task (All Tasks list only).
- Start Time: The time the task was initiated.
- Status: The status of the process. Options are:
  - **Completed**. Tasks that have completed. Completed tasks show 0 minutes to completion.
  - **In Progress**. Tasks that are currently running. In-progress tasks show the expected time to completion.
  - **Failed**. Tasks that have exited with a failed status.

Hovering over the `[i]` icon may display additional information about the task.

© 2026 - Hammerspace

Part 3 — Managing the Cluster

The following sections describe common management and maintenance tasks for the Hammerspace product (Product) infrastructure, including systems, storage, data, and users.

© 2026 - Hammerspace

Chapter 7. Client Data-Access Management

Hammerspace presents the same data to clients over multiple access protocols, including NFS, SMB, and S3, with every DSX node in the cluster exporting all shares. This section describes how clients connect to and access data, the protocols Hammerspace supports, and the security and identity-mapping features that protect that access — including Kerberos for NFS, multi-protocol access to the same data, and auditing of file-access events to an external syslog server.

This section includes the following subsections:

- Auditing Access to Data
- Kerberos Support for NFS
- NFS
- Multi-Protocol Support
- S3
- SMB

7.1. Auditing Access to Data

Hammerspace supports sending file-access auditing events to an external syslog server. These audit events include information such as the type of operation, time stamp, path, inode identifier, user/group, and client.



The client details are currently NFSv4.2-specific. This means that for data accessed using a DSX Portal (SMB, NFS v3, NFS 4.1), only the DSX will appear as a client.

Example of audit event

Sample audit event

```
2025-06-06T15:53:41+00:00 tf-anvil2 filesystem AUDIT [audit_access='dD'
audit_access_type='SUCCESSFUL_ACCESS'] [share='Home' path='./mydata/folder1'] [siteid=2 snapid=0
inode=1026

33 inode_64=2251799813787881] [op='RMDIR'] by [uid='S-1-5-21-3011133496-64911139-1301028031-500', gid='S-
1-5-21-3011133496-64911139-1301028031-512'] at ['Fri 2025-06-06 15:53:41 U

TC'] from [client=4]
```

7.1.1. How to Configure Event Forwarding for Auditing Events

Syslog is used to send events from the cluster. The following example shows how to configure syslog for file system auditing events:

Syslog configuration

Command:

© 2026 - Hammerspace

```
syslog-config --server 192.0.2.50,514,tcp,filesystem --enable
```

Expected output:

```
Enabled:                true
Servers:
                        Address:                192.0.2.50
                        Port:                   514
                        Transport:              TCP
                        Message types:          [Filesystem]
```

7.1.2. Configuring a Share to Send Auditing Events

Auditing information is configured on a client using the built-in tools.

From a Windows Client Using the Microsoft Management Console

The following example procedure shows how to enable auditing for delete and XX operations on the Home share.

1. Open the Microsoft Management Console and connect to the cluster.

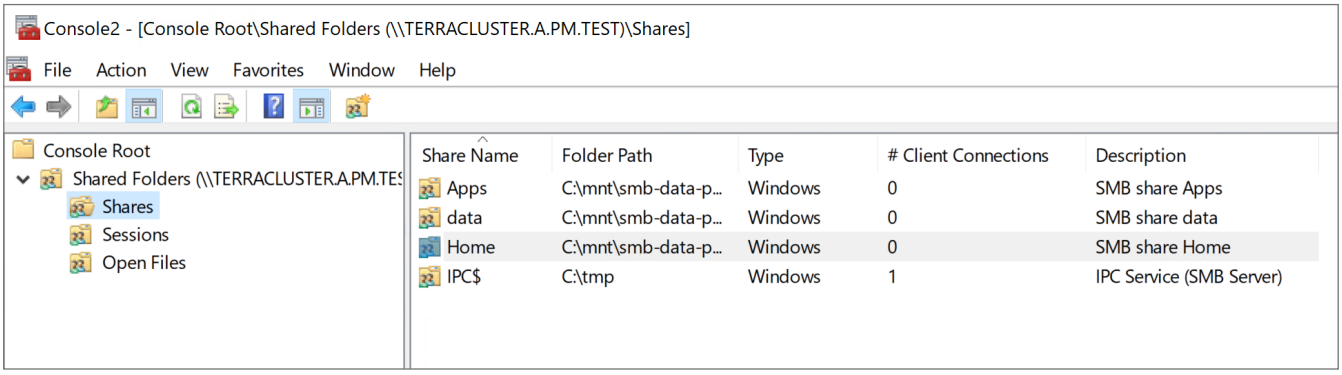


Figure 7. Connect to the cluster

2. Select **Properties** for the Home share and navigate to the **Security** › **Advanced** tab and select **Auditing**.

© 2026 - Hammerspace

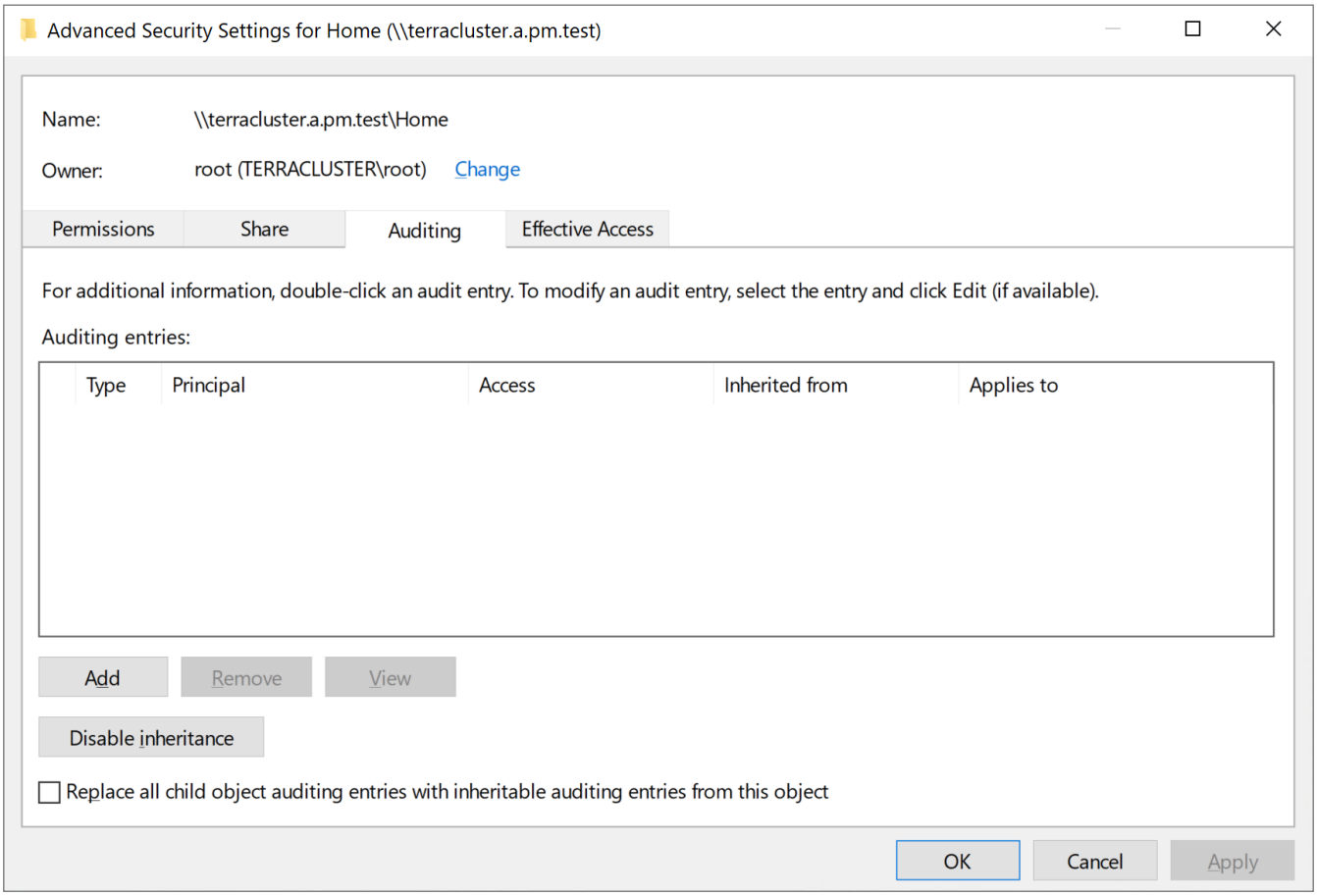


Figure 8. Advanced security properties

3. Click **Add**.
   a. Select **Everyone** for the principal. This will audit file access for all users.
   b. Select **All** for type. This will audit successful and failed attempts. Note that if you only want to audit failed attempts, then select failed.

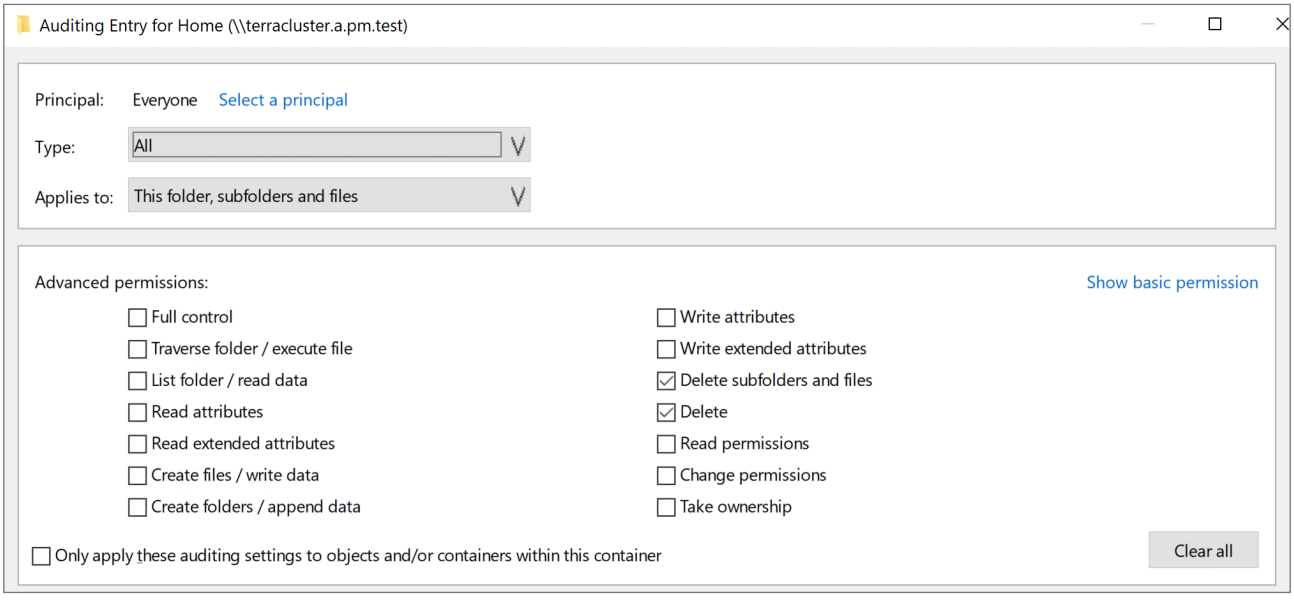


Figure 9. Select auditing specifics

© 2026 - Hammerspace

c. Select the operations to audit. In this example, only delete operations are audited.

4. Auditing for delete operations has now been configured. Events like this will be logged to the syslog stream.

7.1.3. Syslog Examples

File deletion

Syslog file-delete example

```
2025-06-06T15:53:41+00:00 tf-anvil2 filesystem AUDIT [audit_access='dD'
audit_access_type='SUCCESSFUL_ACCESS'] [share='Home' path='./mydata/folder1/file.txt'] [siteid=2 snapid=0
inode=102636 inode_64=2251799813787884] [op='UNLINK'] by [uid='S-1-5-21-3011133496-64911139-1301028031-
500', gid='S-1-5-21-3011133496-64911139-1301028031-512'] at ['Fri 2025-06-06 15:53:41 UTC'] from
[client=4]
```

Directory deletion

Syslog directory-delete example

```
2025-06-06T15:53:41+00:00 tf-anvil2 filesystem AUDIT [audit_access='dD'
audit_access_type='SUCCESSFUL_ACCESS'] [share='Home' path='./mydata/folder1'] [siteid=2 snapid=0
inode=102633 inode_64=2251799813787881] [op='RMDIR'] by [uid='S-1-5-21-3011133496-64911139-1301028031-
500', gid='S-1-5-21-3011133496-64911139-1301028031-512'] at ['Fri 2025-06-06 15:53:41 UTC'] from
[client=4]
```

7.2. Kerberos Support for NFS

Hammerspace supports using Kerberos as a frontend-only security enhancement for NFSv3 and NFSv4.1 and later shares. With a correctly configured Active Directory (AD), Hammerspace shares can be configured with Kerberos security options.

7.2.1. Limitations

- Mounting of shares using IP addresses is not supported.



Although it still WORKS in some circumstances, the mount is not kerberized. This may be because the client needs an SPN to request a Kerberos ticket from the KDC. With an IP address, the SPN it builds (**nfs/*<ip>***) is unlikely to match a real entry in the KDC. If the SPN doesn't match an entry in the KDC, Kerberos auth will fail.

- Currently, Hammerspace only supports Microsoft Active Directory Key Distribution Centers (KDCs).
- Kerberos connections only work with the shares as exposed by the data portals. As such, the data-portal feature *will not work with NFS version 4.2*.
- Kerberos is a frontend-only security enhancement and is not global. Thus, no communication between the data portal and the Anvil (and all other storage nodes) uses Kerberos.

© 2026 - Hammerspace

7.2.2. NFS Protocol Support

NFS Version and Security Modes:

- Kerberos security enhancements for Hammerspace shares work best with NFSv4.1 via data portals.
  - NFS 4.1 data portals are recommended for Kerberos and may not be enabled by default. You can enable data portals with NFSv4.1 using **dp-update --enable** followed by one of the following parameters: [`--data-portal-id, --node-name, or --data-portal-type`]. For more information, use **dp-update --help**
  - While NFSv3 can be used with Kerberos, the full security benefits are only realized with NFSv4 or later deployments. However, it is important to note that data portals will not work with NFS4.2.
- When Kerberos is enabled, export rules for shares must specify one of the following security methods: **krb5** (Kerberos V5 authentication), **krb5i** (Kerberos V5 authentication with integrity checking using checksums), or **krb5p** (Kerberos V5 authentication with privacy service).

7.2.3. Active Directory Configuration Requirements

Many Kerberos integration problems encountered in the field are due to misconfigurations of Active Directory. Setting up an Active Directory to use NFS is well documented online; however, there are several details you should verify to ensure the smooth integration with Hammerspace:

- Your Active Directory is configured to use Kerberos.
- To use the Active Directory, you must join Hammerspace to it with the following options set:
  - NFS Kerberos Service Principal Names (SPNs) are required. They can be created automatically in the Admin CLI using `ad-config --nfs-spns-create`. However, if your environment is complex, you can also manually add them to your Active Directory.



When using Microsoft KDCs, note that the first 15 characters of SPNs must be unique across Storage Virtual Machines (SVMs) within a realm or domain.

- After running the command, you can go to **Administration > Active Directory** in the Management GUI and in the **Kerberos NFS Service Principal Names** dropdown, move the SPNs in the Required SPNs box to the Registered SPNs box, using the double-arrow button.
- DNS records are required for all DSX (Hammerspace) nodes and client machines using Kerberos. Hammerspace recommends using **ad-config --dns-auto-register-floating-ips-enable** to enable automatic registration of portal floating IPs with Active Directory's DNS.
  - Client machines require accurate hostnames and DNS configurations.



This is a critical requirement. Hammerspace currently has no way to know whether this is missing, making it a potentially tricky issue to troubleshoot.

- You can do it manually: The DNS records should look like: **<SMB_SERVER_NAME>.<AD_DOMAIN>**. By default, the SMB server name is the same as the cluster name, so an example DNS record for a DSX would look like: **AE2N4M25AG5E3X.win.ad.test**
- The hostnames of the clients that will use Kerberos to mount NFS shares must be resolvable in Active

© 2026 - Hammerspace

Directory. This is something you must do independently of Hammerspace, in your enterprise infrastructure.

- The clients must be joined to the Active Directory domain.
- Not required, but very likely desired: The RFC2307 or RFC2307BIS schema is in use by the Active Directory domain and configured in Hammerspace. Without this, you will not be able to perform ID mapping.
- You must add Kerberos security options to any Hammerspace shares you want to use with Kerberos.



For the specific Active Directory permissions behind these requirements — including what your AD team needs to delegate for DNS registration and service principal names — see Active Directory Permissions.

When specifying the organizational unit (OU) in which to create the Hammerspace computer account, provide the OU path as a distinguished name, for example:

```
OU=Storage,OU=Servers,DC=corp,DC=example,DC=com
```

If no OU is specified, the computer account is created in the default Computers container.

7.2.4. Hammerspace Cluster Configuration with Kerberos

Enabling Kerberos Security on NFS Export Options

Using the Management GUI

You can enable Kerberos security options as part of the following workflows:

- Creating or editing a Hammerspace share
- Adding or editing an NFS export option on a share

1. Go to **Data › Shares** and click **Create Share**. Or click on the edit icon for an existing share to edit it.

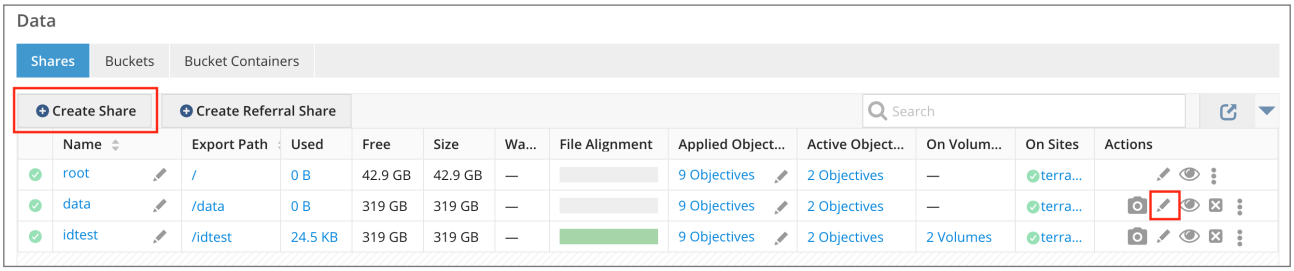


Figure 10. Create a new share or edit the configuration of an existing share

2. Provide the share details, as needed.
3. Click the **NFS** tab. If you are adding an export option, continue with the next step. If editing an existing export option, click the pencil icon for that export option and go to Step 4c to select Kerberos security options.
4. Click **Add Export Option** or the pencil icon, as appropriate, and enter the requested details.

© 2026 - Hammerspace

a. Enter the client specification. This can be one of the following:

- **Single host:** A single host is specified using either a DNS entry name, a dotted-decimal IPv4 address or an IPv6 address. The IPv6 address must not be enclosed in square braces.
- **IP network:** An IP network or subnet is specified using an IP address and a netmask in the format address/netmask. The address is specified either as a dotted-decimal IPv4 address, or as an IPv6 address. The netmask, which must be contiguous, is specified either in dotted-decimal IPv4 format or as a contiguous mask length.
- **Wildcards:** The wildcard operators " **and '?', as well as a square-bracket enclosed class of characters, may be used to match multiple hostnames. The** " will match all characters, including dots, and so may be used to specify an entire subdomain or a class of machines within that subdomain. e.g. `*.my.local.domain` will match any machine in the subdomain my.local.domain, including `www.this.machine.is.in.my.local.domain`.

b. Grant the appropriate access by checking the radio button for **Read-Only** or **Read/Write**.

c. In the Security section, select the appropriate security mode for the share. You do not have to keep UNIX enabled unless you are comfortable with users accessing the system in this manner. Typically, users tend to opt for one approach or the other: strong security with Kerberos or weak security with UNIX settings. The Kerberos security options include

- **Krb5 - Kerberos v5 authentication:** Provides cryptographic proof of a user's identity in each RPC request. This provides strong verification of user identity when accessing data on the server. Note that additional configuration, beyond adding this mount option, is required to enable Kerberos security.



Very few people use basic Kerberos v5 because it is trivial to hack in a man-in-the-middle attack.

- **Krb5i - Kerberos v5 authentication with integrity check:** Provides a cryptographically strong guarantee that the data in each RPC request has not been tampered with.



krb5i makes it less easy for the attacker to modify the payload of the RPC call, but leaves the RPC call and the reply vulnerable to interception if your network is insecure.

- **Krb5p - Kerberos v5 authentication, integrity checking, and privacy protection:** Encrypts every RPC request to prevent data exposure during network transit. Expect a performance impact when using integrity checking or encryption.



krb5p is the only option that offers complete over-the-wire security without the help of some third-party encryption (e.g., TLS).

5. Select whether to **root squash**. Checking the box specifies that requests from the UNIX UID/GID 0, or from the Windows administrator account, are mapped to the anonymous UID/GID.
6. Secure port is checked by default. Unselecting this option disables the *requirement* that NFS client connections must originate from a privileged source port < 1024.
7. After making your selections, click **Add** if you are adding a new export, or **Update** if you are editing an existing export. A bar at the bottom of the window shows the progress of the change and will show

© 2026 - Hammerspace

when it is complete.

8. If you are in the process of creating a share, continue with the configuration. If editing an existing share or export, click **Close**.

After you have finished configuring the share, you can view the export configuration, which displays the Kerberos security flavor in the **Exports** list.

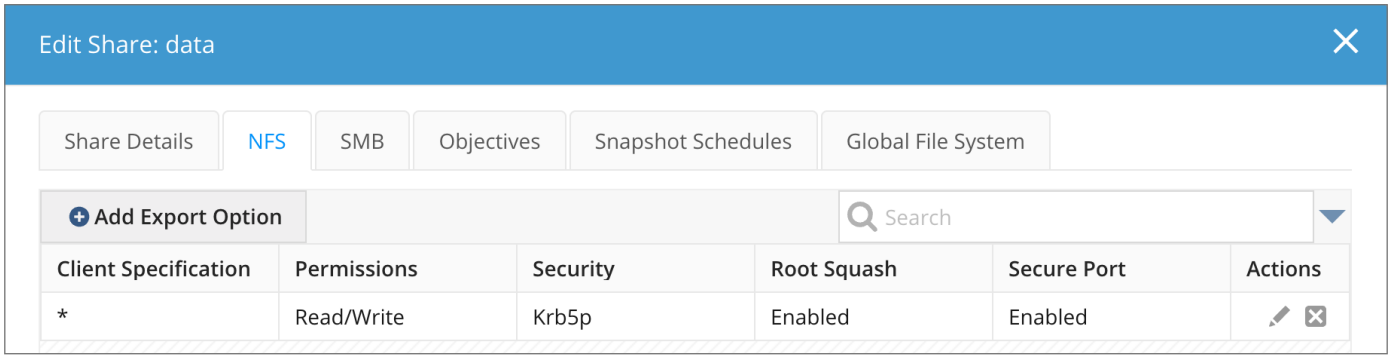


Figure 11. Exports list

Automatic Kerberos Export Options on the Root Share

NFS clients must be able to traverse the root share to reach a kerberized share. For this reason, when you add Kerberos security types to the export options of any share, Hammerspace automatically ensures the root share also permits Kerberos.

When you add Kerberos security types to a share, Hammerspace adjusts the root share as follows:

- If the root share already has a Kerberos security type, nothing changes.
- Otherwise, if the root share has an export option with a wildcard subnet (*), Hammerspace adds the Kerberos security types to that existing export option. This applies both to the default root-share export option and to a customized one — for example, a root share made read-only, or with a changed `root-squash` setting. Existing security flavors are preserved. For example, the default option `*,rw,no-root-squash,sec=sys` becomes:

```
*,rw,no-root-squash,sec=sys:krb5:krb5i:krb5p
```

- If the root share has no wildcard-subnet export option, Hammerspace creates one. The created export option is read-only with root-squash enabled, and does not include the `sys` security flavor:

```
*,ro,root-squash,sec=krb5:krb5i:krb5p
```

When you remove the Kerberos security types from the last non-root share that has them, Hammerspace reverses the change ("dekerberize"):

- If the root share has no Kerberos security types, nothing changes.
- If the root share has no wildcard-subnet export option, nothing changes.

© 2026 - Hammerspace

- Otherwise, Hammerspace removes all Kerberos security types from the wildcard export option, leaving any non-Kerberos security flavors in place.

A share can have only one export option per subnet, so there is never more than one wildcard-subnet export option to reconcile.

Enabling the NFSv4.1 Service

To enable NFSv4.1 support via the DSX data portals, use the following Admin CLI command:

```
dp-update --enable --data-portal-id <ID>
```

You can also use:

- **node-name <node-name>**
- **data-portal-type <type>**

This command activates the NFSv4.1 service on the specified data portal. It is important because NFSv4.1 is not enabled by default in some deployments.

Registering NFS Principal Names

In the Admin CLI

Use the following command to create NFS SPNs in Active Directory automatically:

Command:

```
ad-config --nfs-spns-create
```

In the Management GUI

Navigate to **Administration** › **Active Directory** and move SPNs from the **Required SPNs** box to the **Registered SPNs** box using the double-arrow button.



If SPNs fail to register because the computer account lacks the necessary Active Directory permission, see Active Directory Permissions.

7.2.5. DNS Requirements

Hostnames used in SPNs must resolve correctly via DNS. Use the following command to ensure floating IPs are registered in AD DNS.

Command:

```
ad-config --dns-auto-register-floating-ips-enable
```


© 2026 - Hammerspace

Automatic DNS registration can also be enabled from the Management GUI when joining the domain. See Joining an Active Directory Domain for the GUI procedure, the DNS names involved, and the records to create.

If you'd rather choose which addresses are registered instead of registering all floating IPs automatically, use `--dns-configured-ips` (or `--dns-configured-ip-add` / `--dns-configured-ip-remove` to manage the list incrementally). See the `ad-config` CLI reference for the full option list.

Joining Active Directory

Joining Active Directory is covered in depth in the section, Joining an Active Directory Domain.

7.2.6. Client Configuration

Client requirements for using Kerberos mount options with Active Directory:

- Client **krb5.conf** should be properly configured with settings such as **default_realm**, **dns_lookup_realm** set to **true**, and **dns_lookup_kdc** set to **true**, pointing to your AD KDC.

  See the following example of a properly configured **krb5.conf** file for reference:

  Example of krb5.conf

```
[libdefaults]

default_realm = WIN.AD.TEST

       dns_lookup_realm = true

       dns_lookup_kdc = true

       rdns = false

       ticket_lifetime = 24h

       forwardable = true

udp_preference_limit = 0

[realms]

WIN.AD.TEST = {

       kdc = win2012r21.win.ad.test

       admin_server = win2012r21.win.ad.test

}

[domain_realm]
```


© 2026 - Hammerspace

```
        .win.ad.test = WIN.AD.TEST

        win.ad.test = WIN.AD.TEST
```

- Client systems should be joined to the Active Directory domain (e.g., using **realm join -U *<username>* *<AD_domain>***).
- Client **idmapd.conf** and **sssd.conf** files must be properly configured to map to your AD domain for identity management.

  See the following examples of properly configured **idmapd.conf** and **sssd.conf** files for reference:

  Example of idmapd.conf

```
[General]

        Verbosity = 3

        Domain = win.ad.test

[Translation]

        Method = sss

[Mapping]

        Nobody-User = nobody

        Nobody-Group = nobody
```

  Example of sssd.conf

```
[sssd]

domains = win.ad.test

config_file_version = 2

services = nss, pam

[domain/win.ad.test]

ad_domain = win.ad.test

krb5_realm = WIN.AD.TEST

realmd_tags = manages-system joined-with-adcli

cache_credentials = True
```


© 2026 - Hammerspace

```
id_provider = ad

krb5_store_password_if_offline = True

default_shell = /bin/bash

ldap_id_mapping = False

use_fully_qualified_names = True

fallback_homedir = /home/%u@%d

access_provider = ad
```



SSD can introduce delays if it cannot reach a domain. You can set **ad_enabled_domains** in **sssd.conf** so that SSD ignores any domains not listed. This could potentially help in resolving slow mount or directory listing.

Mounting a Share Using Kerberos Security

It is required to use the FQDN for the hostname when mounting using Kerberos.



The first mount of a share using NFSv4.1 with Kerberos may take slightly longer than later mounts, because the initial mount coordinates across the client, data portal, DSX, Anvil, and Active Directory domain controller. Subsequent mounts complete quickly.

A first-mount delay of several seconds is not expected and usually points to a configuration problem. When ID mapping is enabled and one or more trusted Active Directory domains is offline or unreachable, `sssd` repeatedly retries and times out against those domains. List only the online domains in `ad_enabled_domains` in `sssd.conf` (with the RFC2307 or RFC2307BIS schema in use) to keep `sssd` from stalling on unreachable domains; once it is configured correctly, the delay should no longer occur.

7.3. NFS

Hammerspace provides data access over NFS to a broad set of client operating systems, including Linux, Windows, UNIX, and macOS.



NFSv4.2 has only been tested with Linux clients. More information about which distributions are tested is available in the *Hammerspace 5.2 Hardware and Software Compatibility List*.

7.3.1. NFSv4 Client Unique-Id Requirement

The NFSv4 protocol (applies to both 4.1 and 4.2) requires each client to have a unique ID. By default, the Linux NFS client uses an ID string that contains the local node name. Therefore, if all node names are unique, no further action is required. Failure to use unique IDs on nodes will result in unreliable NFSv4 operations.

If node names are not unique, or if clients were cloned from the same image, assign each client its own NFSv4

© 2026 - Hammerspace

ID:

1. Write a unique ID to a file in `/etc/modprobe.d/`. The following command generates a new UUID for this client:

   ```
   # echo "options nfs nfs4_unique_id=$(uuidgen)" > /etc/modprobe.d/nfs4-unique-id.conf
   ```

   Every client needs a different value. Run the command on each client rather than copying the file between clients, and do not reuse the file from a cloned image.

2. Load the new ID. Either reboot the client, or unmount all NFS file systems, reload the `nfs` kernel module, and remount. The following command assumes that all current NFS mounts are defined in `/etc/fstab`:

   ```
   # umount -a -t nfs; modprobe -r nfs; mount -a -t nfs
   ```

3. Optional: confirm that the client is using the new ID:

   ```
   # cat /sys/module/nfs/parameters/nfs4_unique_id
   ```

For more details, refer to https://www.kernel.org/doc/Documentation/filesystems/nfs/nfs.txt.

7.3.2. NFS Version-Specific Mount Examples

If you are not familiar with NFS mount options, read the NFS **man** page for the client operating system. Here are a few examples that will work for most Linux clients:

NFSv4.2 example:

```
# mount -t nfs -o vers=4.2 <DSX DATA IP>:<EXPORT PATH> <MOUNT POINT>
```

NFSv4.1 example:

```
# mount -t nfs -o vers=4.1,nolock <DSX DATA IP>:<EXPORT PATH> <MOUNT POINT>
```

NFSv3 example:

```
# mount -t nfs -o vers=3,nolock <DSX DATA IP>:<EXPORT PATH> <MOUNT POINT>
```

7.3.3. Supported NFS Protocols

© 2026 - Hammerspace

NFS version	Status
Version 3	Supported
Version 4.1	Supported [IMPORTANT] When NFS 4.1 support is enabled, NFS 4.2 clients must mount the shares from the Anvil and not the DSX nodes.

Version 4.2	• Supported by qualified clients. See the *Hammerspace Hardware and Software Compatibility List* for specifics. If a client tries to access and it is not on the qualified list, it will transparently be re-directed to NFSv3 protocol version, and the mount will succeed.
	• NFS 4.2 clients must have network access to all storage volumes where data may be stored.
	• For RHEL 7.x & CentOS 7.x clients, make sure the client flexfiles timeout is adjusted due to an incorrect default value.
	• Add the following single line to `/etc/modprobe.d/hs_flex.conf` and reboot the client:
	`options nfs_layout_flexfiles dataserver_retrans=0 dataserver_timeo=600`
	• Mount kernel check bypass option:
	For NFS clients that cannot mount due to the kernel output not matching the approved list, there is a way to bypass the strict kernel check. It is recommended to only use this option when the kernel is 5.15 or later. Note that the bypass option is only supported when mounting against the Anvil IP. Example:
	`mount -t nfs -o vers=4.2,port=20492 <ANVIL IP>:<EXPORT PATH> <MOUNT POINT>`

7.3.4. NFSv4.1 ACL Passthrough on Portal Mounts

On an NFS 4.1 client mounted against a Hammerspace portal, you can read and write the full NFSv4 ACL of a file or directory using the standard `nfs4_getfacl` and `nfs4_setfacl` commands (from the `nfs4-acl-tools` package).

```
nfs4_setfacl --dacl -a A::4001:RWX /mnt/share41/test
nfs4_getfacl --dacl /mnt/share41/test
```

These commands default to the older NFSv4.0 ACL attribute. To work with the full NFSv4.1 ACL on a portal mount, pass `--dacl` (discretionary ACL, for access) or `--sacl` (system ACL, for auditing).

© 2026 - Hammerspace

Either consistently use the `--dacl`/`--sacl` flags or consistently never use them. Mixing the two modes produces inconsistent results, because the default operates on the NFSv4.0 ACL attribute while `--dacl`/`--sacl` operate on the separate NFSv4.1 attributes. (The `getrichacl` and `setrichacl` tools default to NFSv4.1 and do not need the equivalent flags.)

Enterprise Linux 8 clients (for example, RHEL 8 and CentOS 8) lack the separate NFSv4.1 ACL extended attributes (`system.nfs4_dacl` and `system.nfs4_sacl`); on these clients, `--dacl`/`--sacl` fail with `Failed to instantiate ACL`, and you can manage only NFSv4.0 ACLs. Use a client distribution qualified for Hammerspace 5.3 (see the Hardware and Software Compatibility List) for full NFSv4.1 ACL support.

7.3.5. Enabling NFS LOCALIO on Linux Clients

You can enable the LOCALIO auxiliary RPC protocol on RHEL 10 or on any other Linux kernel that has LOCALIO support enabled. To do this, add a file to `/etc/modprobe.d/`, for example `/etc/modprobe.d/localio.conf`, that contains the following line:

Enabling LOCALIO in localio.conf

```
options nfs localio_enabled=Y
```

Alternatively, add `nfs.localio_enabled=Y` to the kernel command line. For example, on a system that uses `grubby`:

Command:

```
# grubby --args=nfs.localio_enabled=Y --update-kernel=ALL
```

Whenever a client connects over NFSv4.2 to **protod**, and **protod** does not have a cached entry for it, **protod** will initiate an `EXCHANGE_ID` request to a potential server on the client. If the client accepts the request and replies, it will send back the **so_major_id** in the reply.

The following example shows the server owners that **protod** has recorded:

Example server owner list

Command:

```
$ hs-server-owner-list
```

Expected output:

```
Volume ID    Client ID    Owner ID
337          6            "dsx1.example.com"
395          9            "dsx2.example.com"
421          11           "dsx3.example.com"
477          18           "dsx5.example.com"
```


© 2026 - Hammerspace

```
482          138          "dsx4.example.com"
502          20           "dsx6.example.com"
```

Each of the following clients would be considered to have local I/O to `dsx6.example.com`:

Example showing clients with LOCALIO enabled

Command:

```
$ hs-client-owner-list | grep dsx6
```

Expected output:

```
165          "Linux NFSv4.2 21d1ba73-b292-48b8-bfbf-e37ead97eb8d/dsx6.example.com"
140          "Linux NFSv4.2 cc8011c0-ab65-4df1-8576-8ab97291ea01/dsx6.example.com"
19           "Linux NFSv4.2 dsx6.example.com"
166          "Linux NFSv4.2 21d1ba73-b292-48b8-bfbf-e37ead97eb8d/dsx6.example.com"
139          "Linux NFSv4.2 cc8011c0-ab65-4df1-8576-8ab97291ea01/dsx6.example.com"
20           "Linux NFSv4.2 dsx6.example.com"
```



Not shown in the example above is that 19 and 20 probably correspond to the port 20491 and 20492 connections for case-sensitive and case-insensitive mounts.

Then, for each client that connects, **protod** scans the list of server owners, and if the **so_major_id** is a substring at the end of the **co_ownerid**, the client is considered to have localized storage.

The following sections describe how to set up the NFSv4 server and client identifiers on a Tier 0 architecture so that **protod** can discover the co-location of the client and data server.

To configure the NFSv4 client owner on Linux, refer to the following article: https://docs.kernel.org/filesystems/nfs/client-identifier.html.

Based on the Linux kernel documentation and implementation details, the Linux NFSv4 server likely constructs the server owner name (also called "server scope") using the system's hostname by default.

7.4. Multi-Protocol Support

The Hammerspace product supports NFS, SMB, and S3 protocols using a scale-out architecture in which every DSX node exports all shares.

7.4.1. Multi-Protocol File Locking

Locking is supported across SMB and NFS v4.2 protocols. S3 does not support file locking.

© 2026 - Hammerspace

7.5. S3

Hammerspace supports using the S3 protocol for data access. The Product provides an S3 service as part of the platform. S3 Servers and buckets within each server can then be configured to allow access to data over the S3 protocol.

The S3 protocol is available over the selected ports for an S3 server. Currently, ports 80 (http) and 443 (https) are supported.

7.5.1. S3 Metadata

The S3 service on each DSX node stores two kinds of metadata with the objects it serves:

System metadata

Standard S3 object properties handled by the service itself, such as object size, last-modified time, entity tag (ETag), and content type.

Product custom metadata

User-defined key-value pairs supplied by the S3 client as request headers with the `x-amz-meta-` prefix. These are stored with the object and returned when the object is read. On a copy, whether they are carried over or replaced is controlled by the `x-amz-metadata-directive` header.

7.5.2. S3 Terminology

Table 5. S3 terms and definitions

Term	Definition
Access key	A key used to access and authenticate with the S3 service
Anvil	A component in the Hammerspace platform that provides a global namespace and metadata management.
Bucket	A container for storing data on the S3 server
Bucket container	A top-level location that allows clients to create buckets within it
Delimiter	A character that an S3 listing request uses to group object keys into common prefixes. Hammerspace supports only `/`, which corresponds to the directory separator.
DSX (Data Services node)	A Product node that provides data access and management services. The S3 service runs on DSX nodes.
FQDN (Fully Qualified Domain Name)	A unique identifier for the S3 server used by clients to connect.
Local user	A local user specific to this Hammerspace site.

© 2026 - Hammerspace

Term	Definition
Object key	The name of an object in a bucket. In Hammerspace, a key is a path: each part of the key before a `/` is a directory, and the part after the last `/`, if any, is a file.
Objective	A policy that defines how data should be placed and managed within the Product system.
Prefix	The leading part of an object key. In Hammerspace, a prefix that ends in `/` corresponds to a directory in the bucket.
S3 (Simple Storage Service)	An object storage service
S3 client	Any client that speaks the S3 API, such as a command-line tool, an SDK, or an application with S3 support.
S3 Server	A server that provides S3 storage services.

7.5.3. S3 Limitations

The following tested limitations currently apply to the S3 service:

Table 6. S3 service minimum and maximums

S3 Service Limitations	S3 Service Limitations
Max number of objects	2^64 (18,446,744,073,709,551,616) [1]
Max number of S3 servers	2048
Max number of buckets	20,000 per server
Max number of Access Keys per S3 server	10,000
Minimum object size	0 bytes
Max object size (with multi-part upload)	10,000 * multi-part size [2]
Max multi-part size (recommended)	128 MB [3]
Max length of each part of an object key between slashes	255 bytes [4]

`Footnotes:`

[1] Actual max number is limited by the size of the metadata drives.

[2] 10,000 chunks is a hard coded limit and the max size is determined by the chunk size. Max size with 128 MB chunk size is ~1.2 TB object size.

[3] The multi-part chunk size is set by the S3 client.

[4] Each part of an object key is stored as a file or directory name. The full key, together with the path of the

© 2026 - Hammerspace

bucket's directory in its share, can be at most 4,096 bytes. See S3 Object Keys and the File System Namespace.

For how object keys map to files and directories, and how this affects S3 behavior compared with AWS, see S3 Object Keys and the File System Namespace.

7.5.4. Multi-Tenancy

Hammerspace enables multi-tenancy with S3 in several ways. By combining these methods, you can create a secure and isolated environment for each S3 tenant.

- **S3 Server**: Each tenant connects to their own S3 server. There is no ability for a tenant to discover other S3 servers.
- **Buckets:** Each tenant can be assigned their own bucket or set of buckets. This ensures that data is separated, and tenants cannot access each other's data.
- **Access Control:** Access keys can be used to restrict access to buckets. Each tenant is assigned a unique access key that grants them access to their own buckets.
- **Objectives:** You can use objectives to control data placement and access for each tenant. This allows you to set different policies for each tenant, such as where their data is stored and who has access to it.
- **FQDN (Fully Qualified Domain Name):** Each S3 server is configured with a unique name/endpoint.

7.5.5. Managing S3 Servers

The system supports running many S3 servers, each with a unique configuration. The servers are hosted on all the nodes running the S3 service. This allows each S3 server to be a scale-out entity.

7.5.6. Transport Security

An S3 server can be connected using HTTP or HTTPS.

- It is recommended to use HTTPS to ensure transport-level security between the S3 client and the S3 server. HTTPS connectivity is supported on port 443.
- HTTPS connectivity is only supported with TLS (Transport Layer Security) version 1.2 and 1.3. If possible, it is recommended to use TLS 1.3 on your client. Support for 1.0 and 1.1 has been removed because those versions are no longer considered safe.
- The encryption ciphers have also been reduced to comply with the latest security errata.

7.5.7. X.509 Server Certificate

By default, Hammerspace comes with a self-generated security certificate; however, this will often generate "untrusted certificate" warnings, and some S3 clients may not be able to bypass this warning.

To avoid untrusted-certificate warnings, install a certificate in Hammerspace using the `cluster-config` command.

For more information on managing server certificates, refer to the Server Certificate section.

© 2026 - Hammerspace

7.5.8. Basic Workflow for Configuring S3

The following major tasks describe a generic workflow for configuring S3 data access.

1. Enable the S3 Service.
2. Set up FQDN in DNS.
3. Create an S3 server.
4. Manage users and access to the S3 server.
5. Set up buckets.
6. Configure the S3 client to connect to the S3 server.

7.5.9. Preparing for S3 Configuration

Several key data points should be determined before configuring the S3 server.

- Each S3 server is typically addressed using an endpoint name, which is represented by a DNS record. The endpoint name determines which S3 server the traffic is routed to. We recommend that this be configured before setting up the S3 server. Note that you can only access the default S3 server if using an IP address as the endpoint.
- S3 buckets have two methods for data access: traditional path-based or virtual-host style. It is important to know that the virtual-host style requires additional DNS entries for each bucket accessed using virtual-host addressing.
- Determine how buckets should be managed. The Product supports managing buckets with the administrative API, as well as creating and deleting buckets via the S3 API, or a combination of both.
- If sharing existing data (an entire share or a subfolder within a share), determine the type of access that should be provided via S3. Permissions are standard across all protocols; however, access can be limited via S3 using ACLs. Because a bucket is a directory, each object key is a path to a file in it, and some S3 behaviors differ from AWS as a result. See S3 Object Keys and the File System Namespace.

7.5.10. Creating an S3 Server Using the GUI

Creating an S3 server is required as part of the workflow to access data over the S3 protocol. The workflow to create an S3 Server can be used to create the S3 Server, configure buckets/bucket containers, and add the access keys.



The S3 server can be created using the Admin CLI, Management GUI, or REST API. In *Revision 1* of this documentation, only the GUI method is described. Future revisions will include procedures for creating an S3 server using the CLI or the REST API.

In the GUI, an S3 server can be created as part of the `add-bucket` workflow or the **create-S3-server** workflow. In either workflow, the S3 server options are the same.

© 2026 - Hammerspace

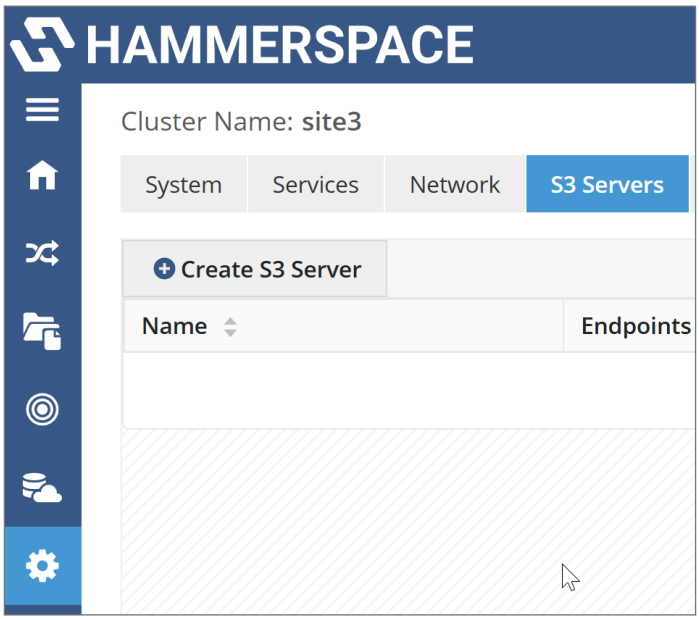


Figure 12. Click the Create S3 Server button

Navigate to **Administration › S3 Servers** tab and click **Create S3 Server**.

1. On the S3 Server page of the wizard, add the required information.

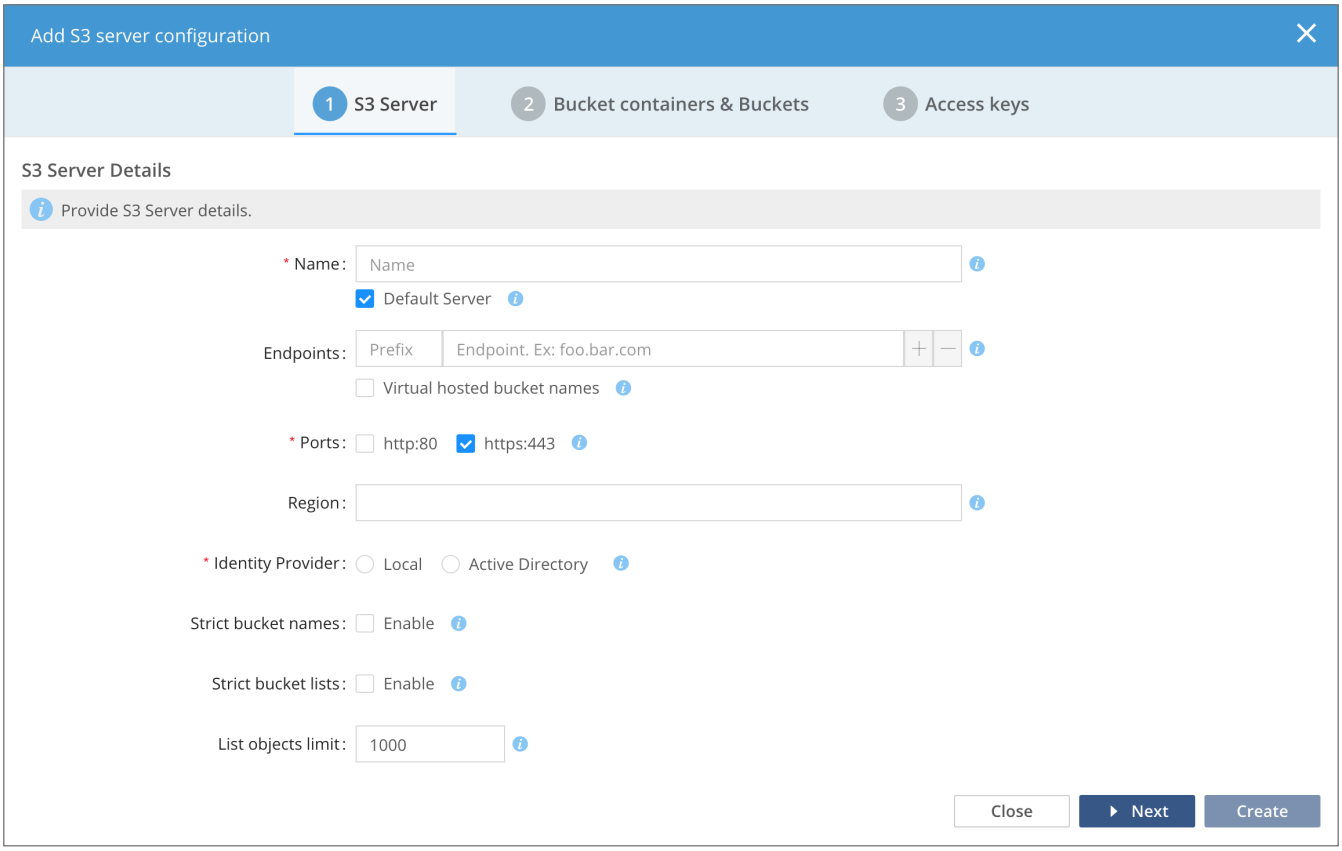


Figure 13. Add server information

Where:

© 2026 - Hammerspace

- **Name:** The management name for the S3 server.
- **Endpoints:** One or more DNS endpoints with prefixes. The endpoint identifies the S3 server and provides a way for clients to connect to different S3 servers. The endpoint must be unique to this S3 server. If no endpoint is configured, the default server option must be checked.
- **Prefix:** Allows virtual-hosted style bucket access. The prefix must be the same for all endpoints.
- **Virtual Hosted Bucket Names:** Check to enable virtual hosted style bucket access. A prefix is required in the endpoint address if this option is checked. DNS needs to be configured for all buckets that will be addressed using virtual-hosted style bucket names.
  Example: **https://bucket-name.<Prefix>.foo.bar.com/**
- **Ports:** Select the HTTP and/or HTTPS protocol. Both can be selected simultaneously.
- **Region:** Default region for buckets created in this server. This field is optional, but can be used when applications are expecting a region. us-east is the default region, which is not configured.
- **Identity Provider:** Every S3 user requires a file system identity, it can be configured to be Local to the server or from Active Directory. If the bucket contents is shared with NFS/SMB protocols, ensure the same identity provider is used.
- **Strict Bucket Names:** Enables strict bucket naming to conform with Amazon S3 bucket naming rules.
- **Strict Bucket Lists:** When enabled, only return the buckets owned by the user.
- **List Objects Limit:** Sets the default number of keys returned by the **ListObjects** or **ListObjectsV2** API call. Note that the ListObjectsV2 API is allowed to override this value.
- **S3 access permissions:** Default access permissions for objects created in buckets. Note that this setting can be overridden with the bucket access permissions setting.

2. Click **Create** to create and configure the S3 server.



The server is not usable until a bucket or bucket container is configured and access keys are set up. This can be done by continuing the wizard or by navigating to the **Buckets or Bucket Container** tab in the GUI.

7.5.11. Deleting an S3 Server

Deleting an S3 server does not delete any actual data; it only deletes the server configuration. The server configuration includes the bucket, bucket container, and access key information.

Care must be taken when deleting an S3 server. The delete process allows for the deletion of an S3 Server with configured buckets; there is no method to undo a deletion.

© 2026 - Hammerspace

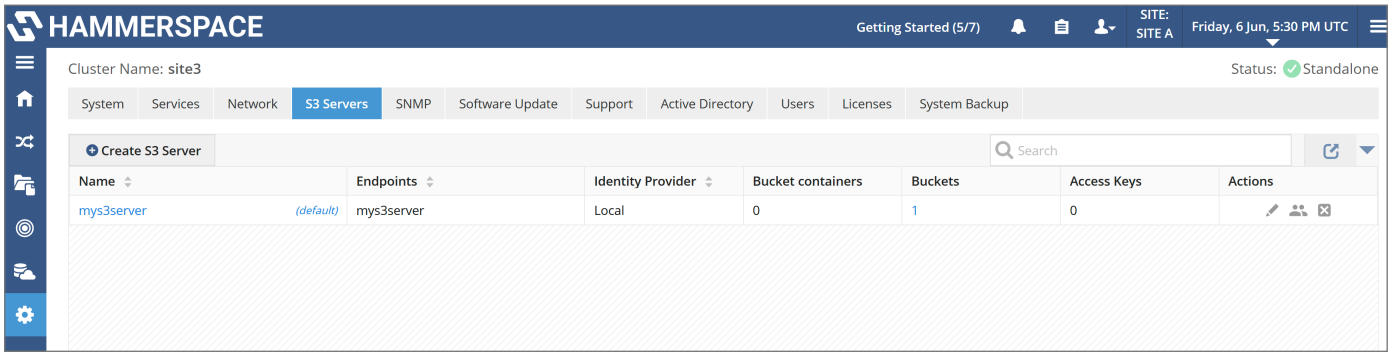


Figure 14. S3 server view

In the Management GUI, navigate to **Administration › S3 Servers**.

1. Click the **Delete** (**X**) icon to delete the S3 server.



This will delete the entire configuration of this S3 server. However, it will NOT delete any of the data. It is required to manually type the name of the S3 server to minimize accidental operations.

Confirm the deletion by clicking Delete S3 server configuration.

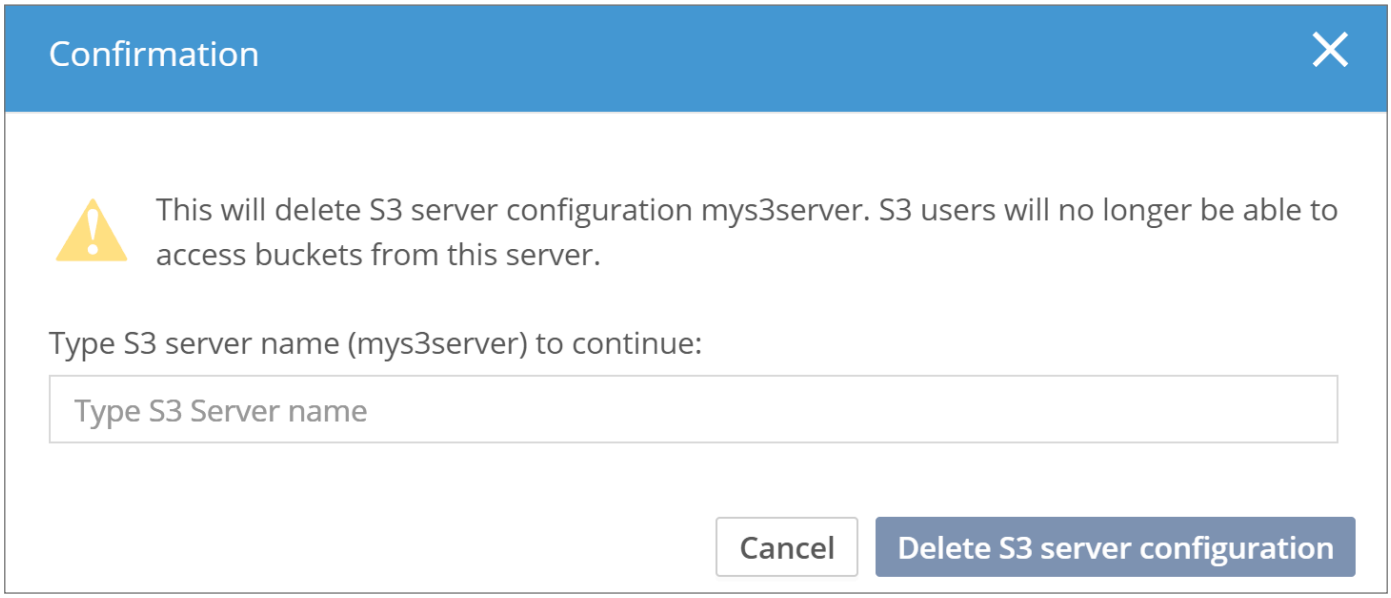


Figure 15. Confirmation of S3 server deletion

7.5.12. Managing S3 Buckets

Deleting an S3 Bucket

S3 buckets can be deleted using the S3 API, as well as from the Management API.

When deleting buckets using the S3 API, the access credentials must have access to delete the bucket, and only empty buckets can be deleted using the S3 API.

© 2026 - Hammerspace

Using the Management GUI

Buckets can be deleted using the product GUI. As part of the workflow, the data can also be deleted from the bucket. This requires the same level of role access as when deleting a share.

Use the following procedure to delete a bucket from an S3 Server:

1. Navigate to the bucket listing. This can be done via the S3 server tab or the **Buckets** tab under the Data section.

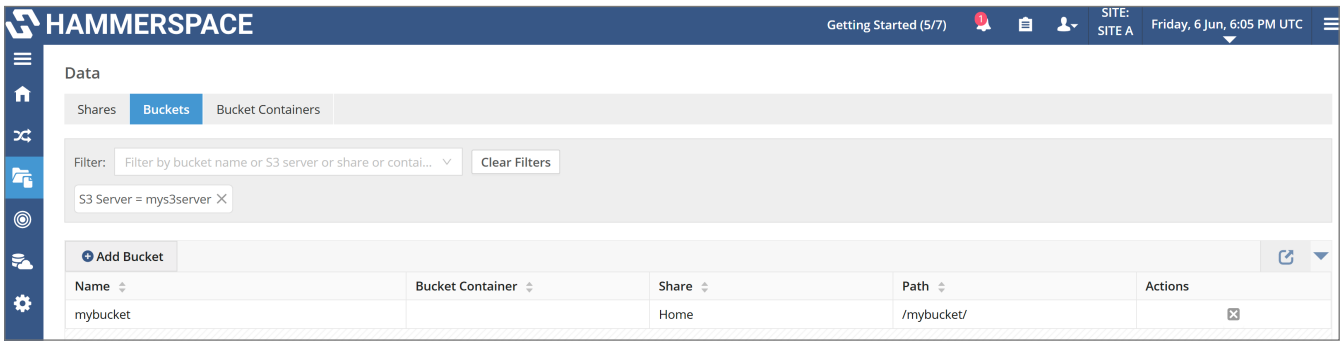


Figure 16. Buckets list

2. Click the **Delete** icon (**X**) under the **Actions** column. This opens the confirmation dialog.
3. Decide whether to delete the data as well as the configuration.

   The dialog offers a **Delete data from bucket** option:

   - Leave it cleared to remove only the bucket configuration. The objects remain in the share, reachable over NFS and SMB, and over S3 if the bucket is reconfigured.
   - Select it to delete the bucket's data along with its configuration. This requires the same level of role access as deleting a share, and there is no method to undo it.

4. Click **Delete bucket configuration** to complete the deletion.

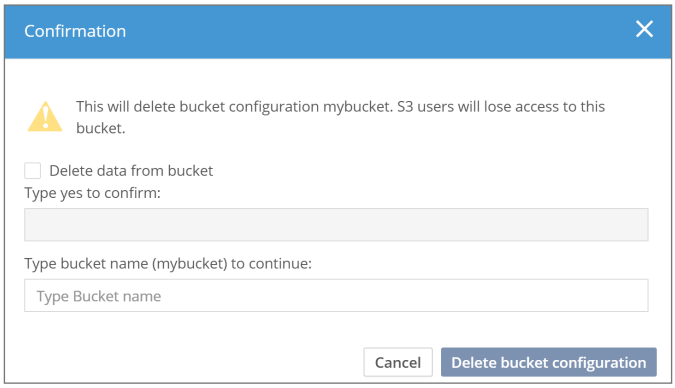


Figure 17. S3 bucket-delete confirmation

Missing Bucket and Bucket Container Events

Buckets and bucket containers created through Hammerspace (the management GUI or the management API) are recorded in the system configuration; those created directly through the S3 API are not. Because each is

© 2026 - Hammerspace

backed by a directory in a share, it can also be reached over NFS or SMB. If the directory for a Hammerspace-created bucket or bucket container is removed over a non-S3 protocol, it is gone on disk but Hammerspace still lists it. The next time an S3 client uses it, the S3 service detects the missing directory and reports an event: status `NoSuchBucket` (a missing bucket) or `NoSuchContainer` (a missing bucket container), recorded with the `S3` source name and listed by `event-list` and in the GUI.



To avoid notification floods, the S3 service reports this only once per S3 service restart. After the first event, further missing buckets are not reported until the S3 service restarts. The event therefore means that *at least one* item is missing; it is not a complete inventory.

This behavior is not specific to replication. In a global filesystem where the same buckets are configured at each site, each site detects and reports the condition independently. With a shared object-storage volume, the site that currently hosts the volume reports it.

When you see one of these events:

- Note the bucket name or path identified in the event.
- Determine whether the directory was removed intentionally. For a bucket or bucket container created through the GUI or the management API, removing the directory over NFS or SMB does not remove it from the Hammerspace configuration, so the two are now out of sync.
- Resolve the mismatch by recreating the bucket or bucket container, or by deleting its Hammerspace configuration.
- Re-check your full bucket configuration, because a single event does not necessarily correspond to a single missing item.

7.5.13. Managing S3 Users

Access to S3 requires credentials, which are provided using the Access and Secret key approach. Each S3 server manages its own set of credentials, and one set of credentials cannot be used with another S3 server unless the same credentials are added to that server.

The credentials map to an S3 user. The S3 user maps to an identity in the file system. This identity can be either a local user or originate from a centralized source, such as Active Directory. When creating the S3 server, the identity source is configured, and mixing multiple sources is not supported.

Adding an Access Key to an S3 Server

Use the following procedure to add an access key to an S3 server:

1. In the Management GUI, navigate to **Administration** › **S3 Servers** tab.
2. For the server you want to manage, click either the **Edit** (pencil) icon and then the **Access keys** tab, or the **View Users** icon.
3. Click **Add Access Key**.

© 2026 - Hammerspace

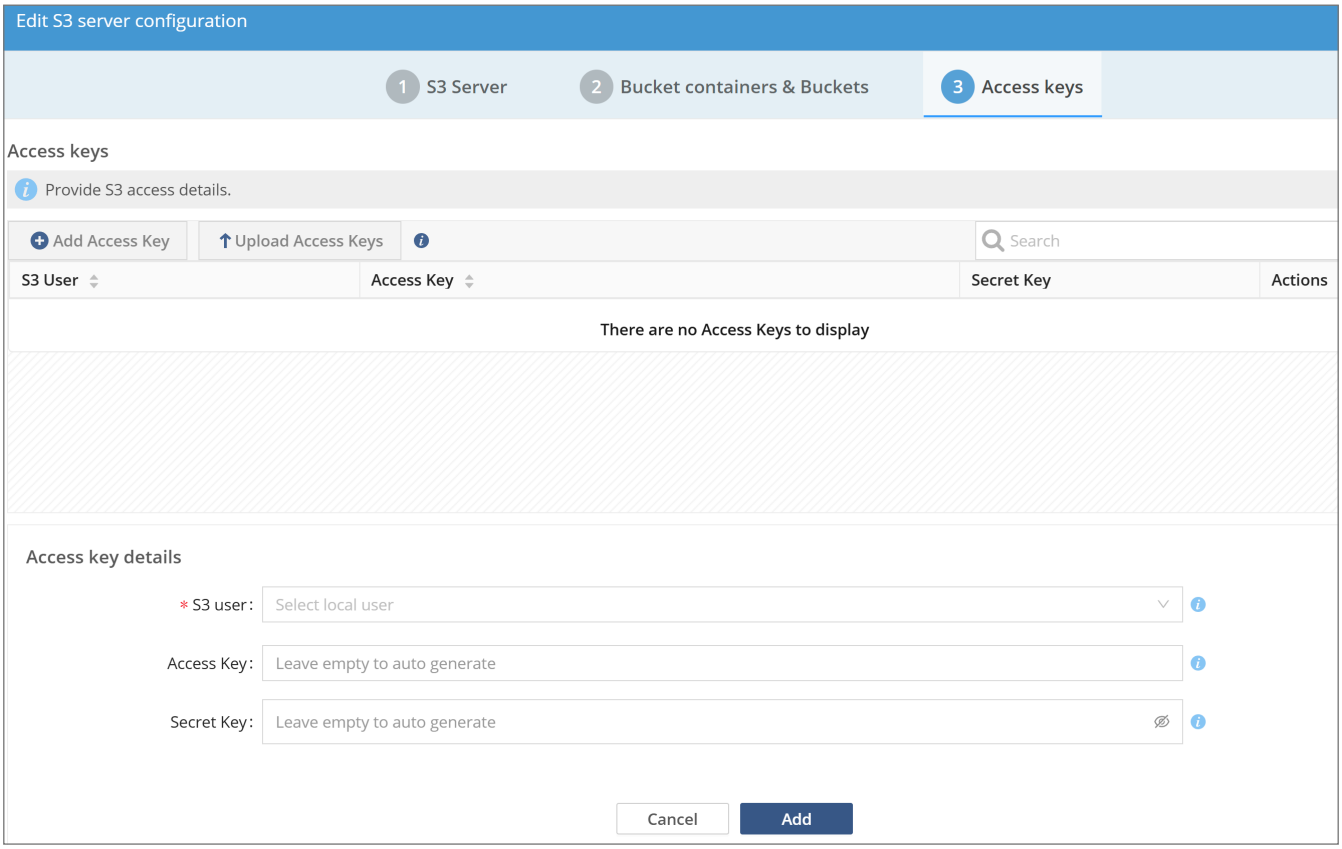


Figure 18. Example of adding an access key for a local user

Additional details:

- The **S3 user** field must match a locally created user if using local users. If none exists, navigate to **Administration › Users** and add a local user for **data access**.
- If using Active Directory, the username will be validated, and an error will be shown if it cannot be found.
- The Access and Secret key will be autogenerated if left empty. Note that the access key needs to be unique within an S3 server.

4. Press **Add** to update the configuration. The secret key will not be visible again after this step and must either be downloaded or manually copied to a secure location.

© 2026 - Hammerspace

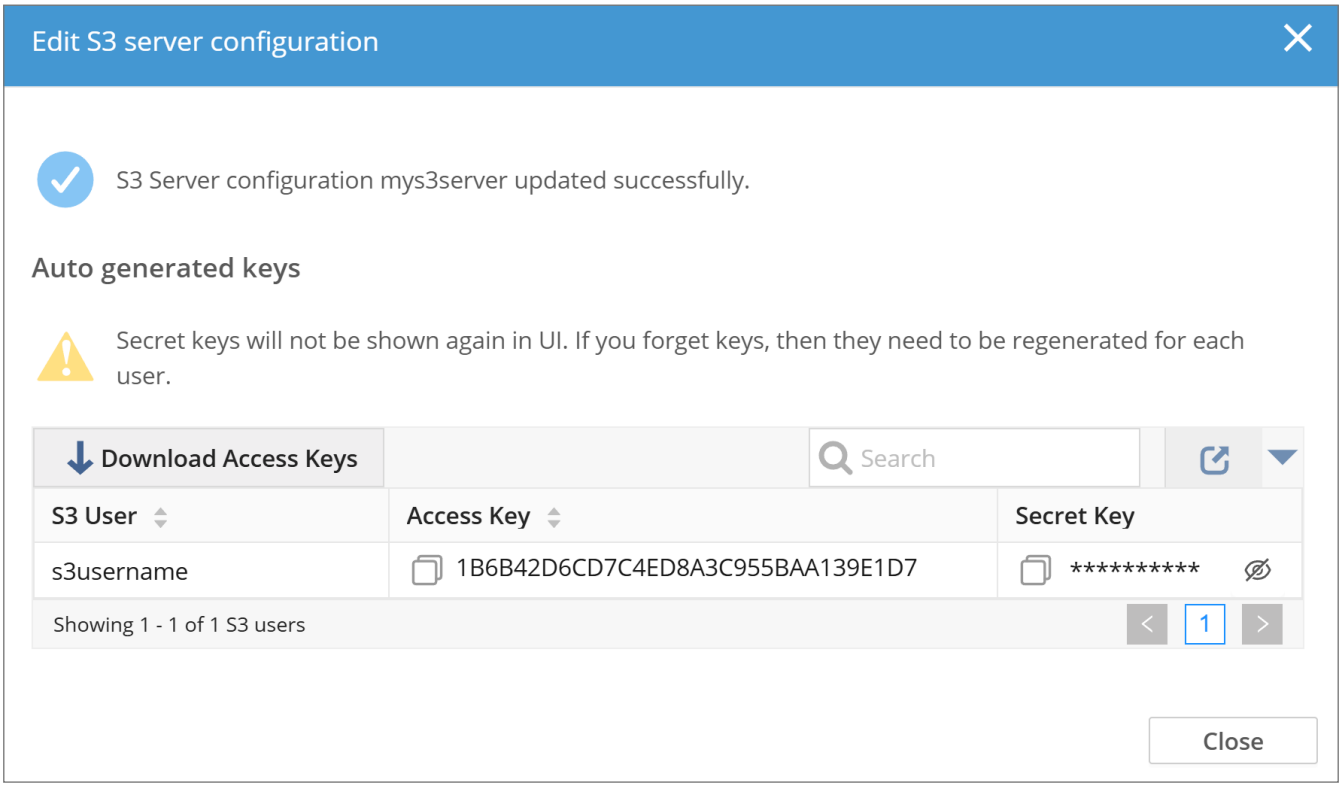


Figure 19. Adding an access key to an S3 server

Adding Multiple Local Users at the Same Time

The product also supports uploading a `.csv` file with columns for username, access, and secret key. All three are required to successfully import a user. A sample CSV can be downloaded by clicking the information (i) icon.

Example CSV with a sample entry

```
s3user,accessKey,secretKey
testUser,testUserAccessKey,testUserSecretKey
```

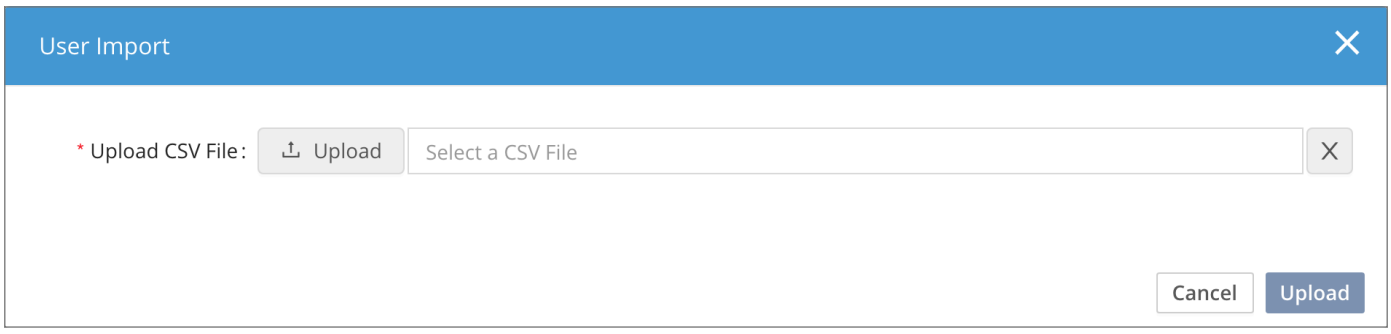


Figure 20. Uploading a CSV for S3 users

Removing Access to an S3 Server

To remove access to an S3 server, a user can be either deleted or disabled (in Active Directory, for example), and they will no longer be able to authenticate and connect.

© 2026 - Hammerspace

If the user is removed from the S3 server, it will only impact access to that S3 server, not any other S3 servers. Removing a user will not remove any objects/data in the file system.

7.5.14. S3 API Overview

Hammerspace supports the S3 API operations listed below. Addressing and authentication support both path-style and virtual-hosted bucket addressing, and presigned URLs using AWS Signature Version 4 query parameters.

Table 7. Bucket operations

Operation	Notes
CreateBucket	
DeleteBucket	
HeadBucket	
ListBuckets	
GetBucketLocation	
GetBucketAcl / PutBucketAcl	
GetBucketCors / PutBucketCors / DeleteBucketCors	
GetBucketOwnershipControls / PutBucketOwnershipControls / DeleteBucketOwnershipControls	
GetBucketPolicyStatus	
GetBucketVersioning	
PutBucketVersioning	Cannot be used to disable snapshots or versioning.
GetBucketIntelligentTieringConfiguration	Always returns `STANDARD`.
DeleteBucketIntelligentTieringConfiguration	Always returns `200 OK`.
GetBucketPolicy / DeleteBucketPolicy	
GetPublicAccessBlock / DeletePublicAccessBlock	
GetBucketLogging	
GetBucketAccelerateConfiguration	

Table 8. Object operations

© 2026 - Hammerspace

Operation	Notes
GetObject / PutObject / CopyObject	
HeadObject	Returns `200 OK` for a key that names a directory. See Listing and Lookup.
DeleteObject / DeleteObjects	A DeleteObjects entry that names a directory deletes the directory and everything in it. DeleteObject has no effect on a directory that contains objects. See Deleting Objects and Prefixes.
ListObjects / ListObjectsV2	Only `/` is supported as a delimiter. See Listing and Lookup.
ListObjectVersions	
GetObjectAcl / PutObjectAcl	
GetObjectTagging / PutObjectTagging / DeleteObjectTagging	
GetObjectLockConfiguration / PutObjectLockConfiguration	
GetObjectLegalHold / PutObjectLegalHold	

Table 9. Multipart upload operations

Operation	Notes
CreateMultipartUpload	
UploadPart / UploadPartCopy	
ListMultipartUploads / ListParts	
CompleteMultipartUpload / AbortMultipartUpload	

Operations that do not appear in these tables are not supported.

Because each bucket is a directory in a share, some operations on object keys that contain `/` behave differently than on AWS. See S3 Object Keys and the File System Namespace.

7.5.15. S3 Object Keys and the File System Namespace

In AWS S3, a bucket is a flat namespace. An object key is an opaque string, and the `/` character in a key means nothing to the service. Clients and consoles treat `/` as a folder separator only by convention.

A Hammerspace S3 bucket is a directory in a Hammerspace share, and every object in it is a file. Hammerspace treats each `/` in an object key as a directory boundary: each part of the key before a `/` is a directory, and the part after the last `/`, if any, is a file. Because of this, the same data is available as files over NFS and SMB and as objects over S3, with no copying or conversion. This is a deliberate design choice, and S3

© 2026 - Hammerspace

services that store objects in a file system generally share it.

A name in a file system can be a file or a directory, but not both. As a result, some sequences of S3 requests that succeed on AWS behave differently on Hammerspace. These differences matter most for applications and tools that:

- Use the same name as both an object and a prefix.
- Create zero-byte "folder" objects whose keys end in `/`.
- Delete data by prefix.
- Run S3 conformance test suites.

The following sections describe each difference and the exact response Hammerspace returns.

How Object Keys Map to Files and Directories

When an S3 client writes an object, Hammerspace stores it as a file at the path given by its key, beneath the bucket's directory. Hammerspace creates any directories in the path that do not already exist.

Table 10. How object keys are stored

Object key	Stored as
`q1.csv`	File `q1.csv` in the bucket's directory
`reports/2024/q1.csv`	Directory `reports`, directory `reports/2024` within it, and file `q1.csv` within that
`logs/`	Directory `logs`, with no file (see Keys That End in a Slash)

© 2026 - Hammerspace

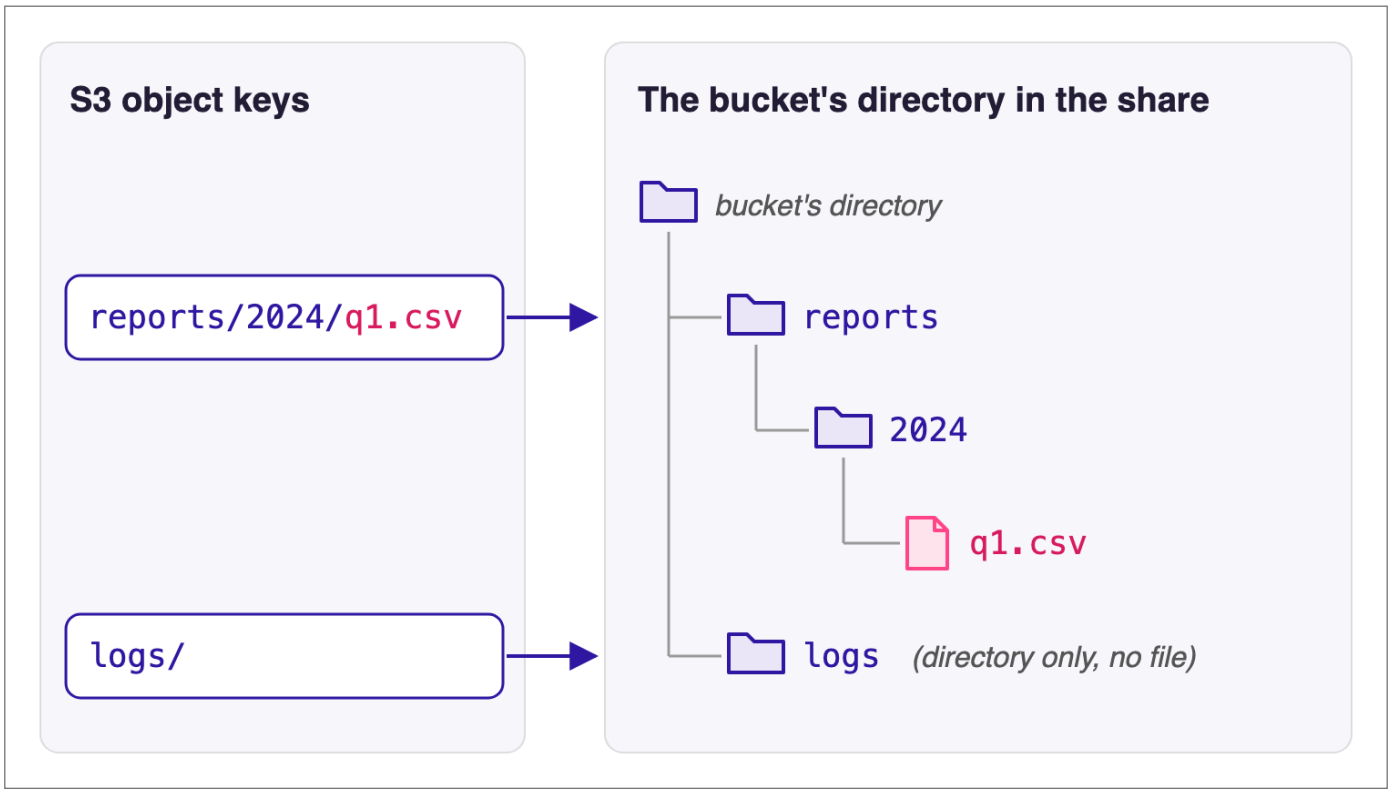


Figure 21. How the keys `reports/2024/q1.csv` and `logs/` are stored in the bucket's directory

Because keys are stored as paths, Hammerspace normalizes them the way a file system normalizes a path:

- Repeated slashes collapse into one. The keys `data//file.txt` and `data/file.txt` refer to the same object.
- A leading slash is dropped. The keys `/data/file.txt` and `data/file.txt` refer to the same object.

On AWS, each of these keys is a distinct object.

Each part of a key between slashes becomes a file or directory name, so it can be at most 255 bytes long. The full key, together with the path of the bucket's directory in its share, can be at most 4,096 bytes long. Hammerspace does not apply the AWS limit of 1,024 bytes to a whole key. A request that uses a longer key or key part fails, and the error it returns depends on the operation.

An object key must be valid UTF-8.

Object and Prefix Name Conflicts

On AWS, one bucket can hold an object named `reports` and an object named `reports/2024/q1.csv` at the same time. On Hammerspace, that would make `reports` both a file and a directory, which a file system cannot hold. Whichever of the two writes comes second fails.

Table 11. Object and prefix conflicts

© 2026 - Hammerspace

Requests, in order	Why the second write fails	Response to the second write
PutObject `reports`, then PutObject `reports/2024/q1.csv`	`reports` is a file, so Hammerspace cannot create a directory with that name	Depends on the upload type. See the list below.
PutObject `reports/2024/q1.csv`, then PutObject `reports`	`reports` is a directory that contains objects, so Hammerspace cannot create a file with that name. If the directory is empty, the object replaces it. See Keys That End in a Slash.	`409 Conflict`, error code `ConditionalRequestConflict`, for standard and chunked uploads.

When an object already exists at a name and a client writes a key that uses that name as a prefix, the response depends on how the client uploads the data:

- **Standard upload** (for example, the AWS CLI `put-object` command): `409 Conflict`, error code `ConditionalRequestConflict`.
- **Chunked (streaming) upload**, where the request's `x-amz-content-sha256` header begins with `STREAMING-`. This is the default for the AWS SDK for Java. Hammerspace returns HTTP status `429`, error code `SlowDown`. S3 SDKs treat `SlowDown` as a temporary error and retry, so the client retries the upload several times before it reports a failure. If an application reports repeated `SlowDown` errors for a single key while the cluster is otherwise responsive, check whether the key conflicts with an existing object.

A multipart upload reports the conflict when it completes: CompleteMultipartUpload returns `409 Conflict`, error code `ConditionalRequestConflict`.

For example, with the object `t/obj` in the bucket, a standard PutObject request for the key `t/obj/child` fails with this response:

Request:

```
PUT /<bucket-name>/t/obj/child
```

Response:

```
HTTP 409 Conflict
<?xml version="1.0" ?>
<Error>
  <Code>ConditionalRequestConflict</Code>
  <Message>A conflicting operation occurred. If using PutObject you can retry the request. If using
multipart upload you should initiate another CreateMultipartUpload request and re-upload each
part.</Message>
  <Resource>/<bucket-name>/t/obj/child</Resource>
  <RequestId><request-id></RequestId>
</Error>
```


© 2026 - Hammerspace

On a standard upload, an application can detect a conflict from the HTTP status `409`. On a chunked upload, the conflict appears as repeated `SlowDown` errors.

The same conflict occurs when one of the names is created over another protocol. For example, if a user creates a directory named `reports` in the bucket's directory over NFS or SMB, a later PutObject for the key `reports` fails in the same way.

Keys That End in a Slash

Some S3 tools create an empty "folder" by writing a zero-byte object whose key ends in `/`, such as `logs/`. On AWS, that is an ordinary object that can also hold data. On Hammerspace, a PutObject request for a key that ends in `/` creates a directory instead:

- Hammerspace creates the directory. It does not store an object. If the request has a body, Hammerspace discards it.
- If the directory already exists, the request succeeds and nothing changes.
- If the name is already in use by an object (for example, the object `logs` exists and the request is for `logs/`), the request fails with `409 Conflict`, error code `ConditionalRequestConflict`.

How a Folder Appears in Listings

Hammerspace lists a directory as a key only while the directory is empty. Once it contains an object, a listing shows the objects instead.

Table 12. How the folder logs/ appears in ListObjects and ListObjectsV2

State	Listing without a delimiter	Listing with delimiter `/`
`logs/` is empty	The zero-byte key `logs/`	The common prefix `logs/`
`logs/` contains the object `logs/app.log`	The key `logs/app.log` only	The common prefix `logs/`

On AWS, a listing without a delimiter would return both `logs/` and `logs/app.log`, because the folder is an object in its own right.

For example, after a PutObject request for the key `t/m/`, a ListObjectsV2 request with prefix `t/` and no delimiter returns the empty folder in this `Contents` element:

Request:

```
GET /<bucket-name>?list-type=2&prefix=t%2F
```

Response (excerpt):

```
<Contents>
  <Key>t/m/</Key>
```


© 2026 - Hammerspace

```
  <LastModified><timestamp></LastModified>
  <ETag><etag></ETag>
  <Size>0</Size>
  <StorageClass>STANDARD</StorageClass>
  <Owner>
    <ID><owner-id></ID>
    <DisplayName><user-name></DisplayName>
  </Owner>
</Contents>
```

When a Folder Disappears

When an S3 client deletes an object, Hammerspace also removes the object's parent directory if it is now empty, and continues up the path until it reaches a directory that is not empty. As a result, a folder created from a key that ends in `/` disappears when the last object in it is deleted. On AWS, the `logs/` object remains until it is deleted explicitly.

This behavior is not configurable. If an application needs the folder to remain, it must create the folder again after it deletes the folder's last object.

An empty folder is also replaced by an object with the same name. For example, if `logs/` is an empty folder, a PutObject request for the key `logs` removes the folder and stores the object. If the folder contains objects, the request fails; see Object and Prefix Name Conflicts.

Listing and Lookup

A listing request with a prefix and the delimiter `/` maps directly onto a directory. For example, a ListObjectsV2 request with prefix `reports/` and delimiter `/` returns the objects in the directory `reports` as keys, and returns each of its subdirectories as a common prefix. A directory that contains objects never appears as a key itself; only its objects do. For how an empty directory appears, see Keys That End in a Slash.

The only supported delimiter is `/`. If a listing request specifies any other delimiter, Hammerspace treats it as `/`, and the response reports `/` as the delimiter. An application that groups keys with a different delimiter, such as `|`, receives results grouped by `/` instead.

In a ListObjects (version 1) response to a request with a prefix, each common prefix omits the request's prefix. For example, for prefix `reports/`, the subdirectory `reports/2024/` appears as `2024/`. ListObjectsV2 reports the full prefix.

A HeadObject request for a key that names a directory (for example, `reports` when the object `reports/2024/q1.csv` exists) returns `200 OK`. On AWS, the same request returns `404 Not Found`, because no object has that key. Tools that use HeadObject to tell an object from a folder can therefore treat the directory as an object. A GetObject request for the same key returns `404 Not Found`, error code `NoSuchKey`.

For example, with the object `t/d/b` in the bucket, HeadObject and GetObject requests for the key `t/d` receive these responses:

Request:

© 2026 - Hammerspace

```
HEAD /<bucket-name>/t/d
```

Response:

```
HTTP 200 OK
```

Request:

```
GET /<bucket-name>/t/d
```

Response:

```
HTTP 404 Not Found
<?xml version="1.0" ?>
<Error>
  <Code>NoSuchKey</Code>
  <Message>The specified key does not exist</Message>
  <Resource>/<bucket-name>/t/d</Resource>
  <RequestId><request-id></RequestId>
</Error>
```

Deleting Objects and Prefixes

On AWS, DeleteObject and DeleteObjects both delete only the object whose key matches exactly. On Hammerspace, a key can name a directory, and the two operations handle a directory differently.

Table 13. Deleting a key that names a directory

Operation	Directory is empty	Directory contains objects
DeleteObject	Removes the directory	Returns `204 No Content` and deletes nothing
DeleteObjects (batch delete)	Removes the directory	Deletes the directory and everything in it. See the warning below.

A DeleteObjects request that lists a key naming a directory deletes that directory and every object and subdirectory beneath it. For example, a DeleteObjects entry for `reports` or `reports/` deletes `reports/2024/q1.csv` and every other object whose key begins with `reports/`. Copies of the objects in existing snapshots are not deleted. On AWS, the same request deletes only an object named exactly `reports` or `reports/`, and leaves the other objects in place.

Before you send a DeleteObjects request, check that no entry in it names a prefix that you intend to keep.

© 2026 - Hammerspace

For how Hammerspace removes empty parent directories after a delete, see Keys That End in a Slash.

Guidance for Applications and Test Suites

When you port an S3 application to Hammerspace, or when an S3 conformance test fails, check the following points:

- Do not use one name as both an object and a prefix. Choose a layout in which every name is either an object or a folder. For example, store a summary as `reports/summary.csv` rather than as an object named `reports` next to the prefix `reports/`.
- When PutObject returns `409`, check for a naming conflict before retrying. A naming conflict is not a temporary error: retrying does not succeed until the conflicting object or directory is removed. The message in a `ConditionalRequestConflict` response says that you can retry the request; that does not apply to a naming conflict. In Hammerspace 5.2, a chunked upload reports the same conflict as `SlowDown`, so check for a naming conflict when one key keeps failing with `SlowDown`.
- Do not store data in keys that end in `/`. Hammerspace discards the body of such a request, removes the folder when its last object is deleted, and replaces an empty folder with an object of the same name.
- Use `/` as the only delimiter in listing requests.
- Use DeleteObjects only with the exact keys of objects. An entry that names a folder deletes everything in it.
- Do not rely on keys that differ only by repeated or leading slashes. Hammerspace stores them as the same object.

Some S3 conformance test suites write an object and then write another object beneath it as a prefix, or write data to a key that ends in `/`. On Hammerspace, those tests fail because of the namespace model described in this section, and not because the operation under test is unsupported. The conflict is usually incidental to the test's layout. Treat these failures as expected differences, or change the test's key names so that no name is used as both an object and a prefix.

7.6. SMB

SMB clients connect to any of the DSX nodes in the cluster. All SMB shares are available from all DSX nodes. This should ideally be through the floating IP/name registered in DNS.

7.6.1. Active Directory Integration

Hammerspace integrates with Active Directory (AD) to enable authentication and User ID mapping.



Ports 88, 135, 389, and 445 must be open.

Joining an Active Directory Domain

Complete the following steps to join an Active Directory domain:

1. Go to the **Administration** page of the GUI and select the **Active Directory** tab.
2. Enter the **Domain Name** using Fully Qualified Domain Name (IP can also be used).

© 2026 - Hammerspace

3. Set the **SMB server name**. The SMB server name is used to register a computer account in Active Directory. To change the SMB server name, leave and rejoin Active Directory with the new name. This is the name that SMB clients use to navigate to with the ***\\<SMB SERVER NAME.domain name>*** address.
4. Enter **Active Directory account credentials** with the privilege to *add workstations* to a domain. In this case, the SMB server computer account will be registered in the AD server. Ongoing DNS registration is then handled by that computer account rather than by these credentials; for the exact permissions involved, see Active Directory Permissions.
5. *Optionally*, enter an **Organizational Unit**. Note that the credentials from Step 4 must have permissions to write into the Organizational Unit location.
6. Auto select AD servers or manually select from a list of discovered servers. Auto select works well when AD Sites and Services are set up correctly.
7. Select schema used to map Active Directory Security Identifiers (SIDs) to User IDs (UID) and Groups (GID):
   a. Disabled: No cross-protocol user mapping between NFS and SMB
   b. RFC2307: Enable cross-protocol user mapping between NFS and SMB using the RFC2307 schema
   c. RFC2307BIS: Enable cross-protocol user mapping between NFS and SMB using the RFC2307BIS schema

   Refer to the **User Mapping** section for more details.

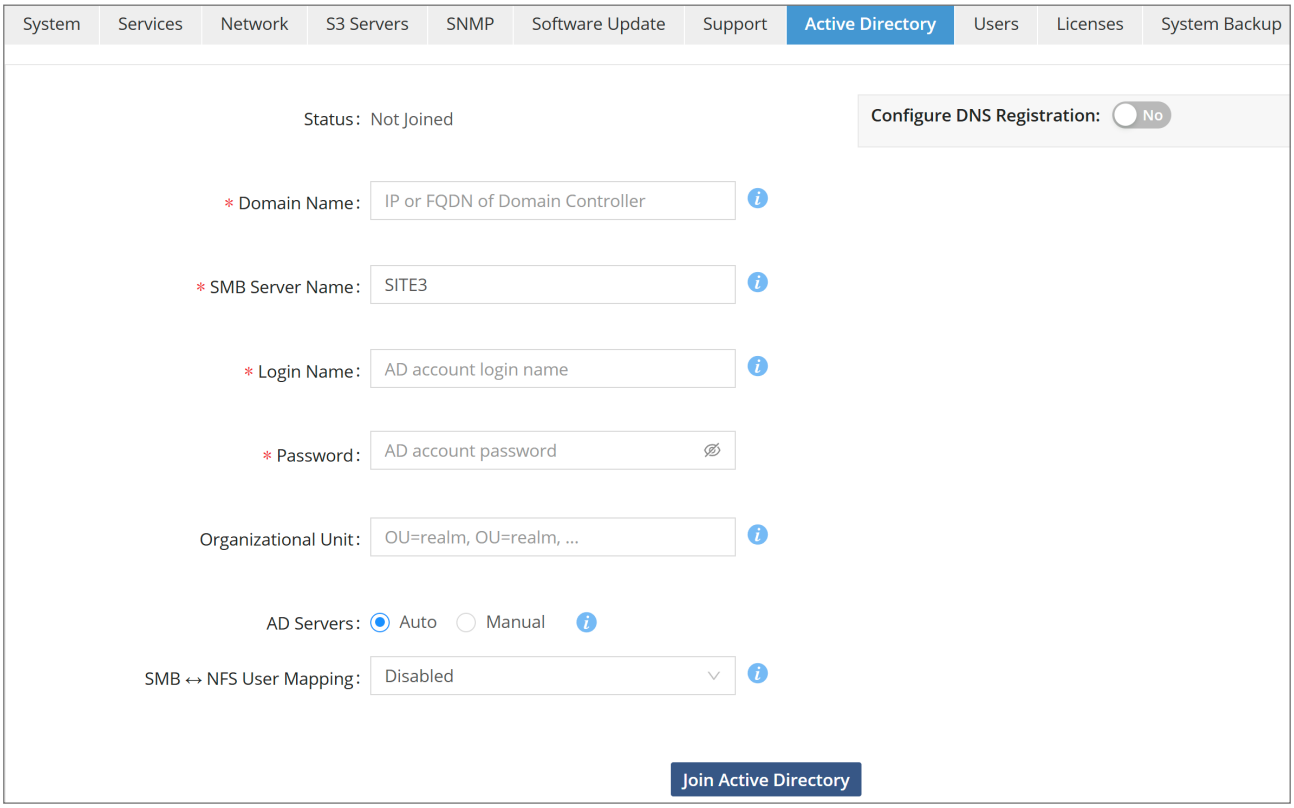


Figure 22. Joining an Active Directory domain

8. At this step, sufficient information has been provided to join Active Directory; however, manual DNS registration of the SMB server name is required unless proceeding to the next step.

© 2026 - Hammerspace

Leaving an Active Directory Domain

Complete the following steps to leave a domain:



After leaving an Active Directory domain, SMB clients cannot connect to the SMB shares. This operation will interrupt data access and must be executed carefully. Also, if S3 servers are configured to use AD, users will no longer be able to connect to those S3 servers.

1. Go to the **Administration** page of the GUI and select the **Active Directory** tab.

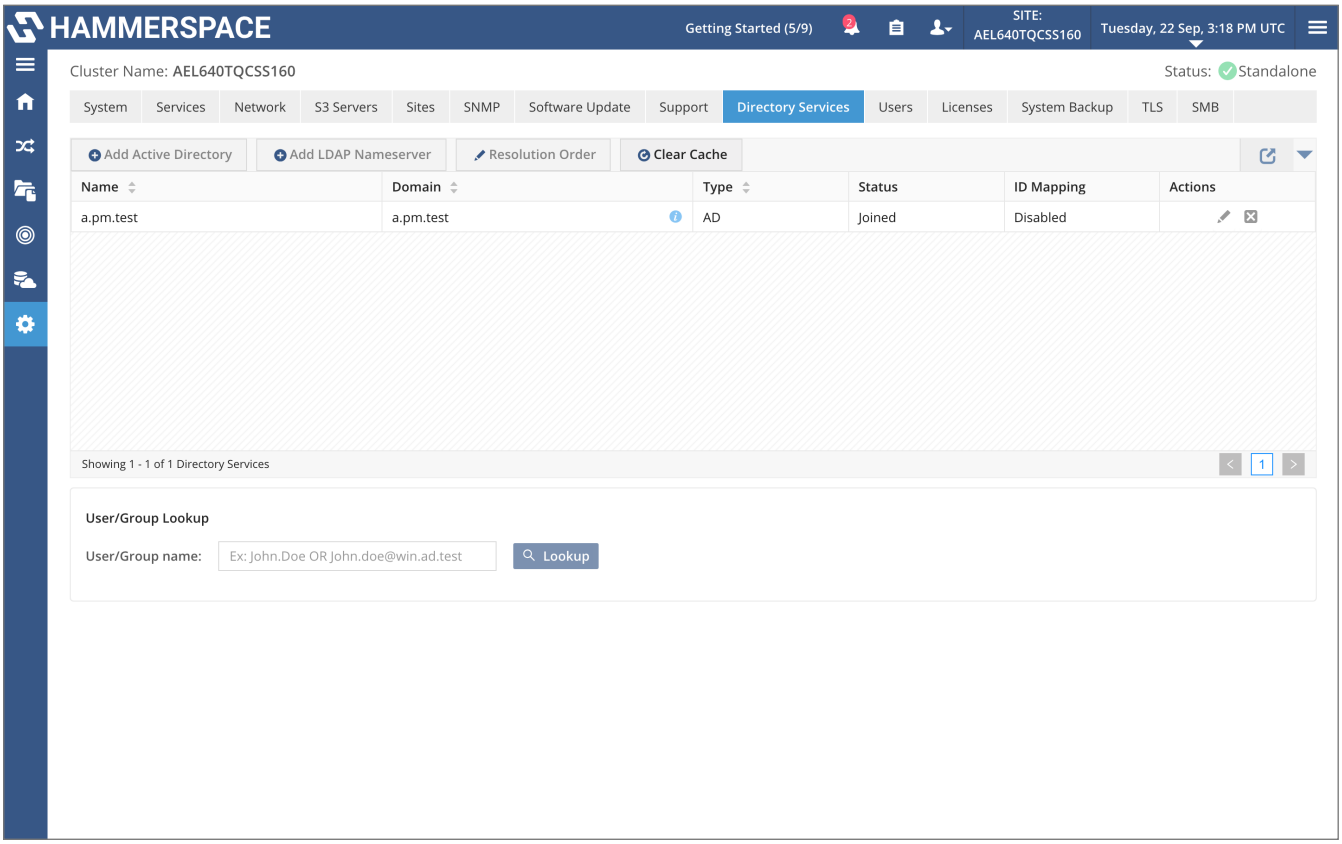


Figure 23. Leaving an Active Directory domain

2. Click **Leave Active Directory** to initiate the task.
3. A pop-up will appear, allowing you to optionally remove the Computer Account and DNS entries if they were created by the product.

© 2026 - Hammerspace

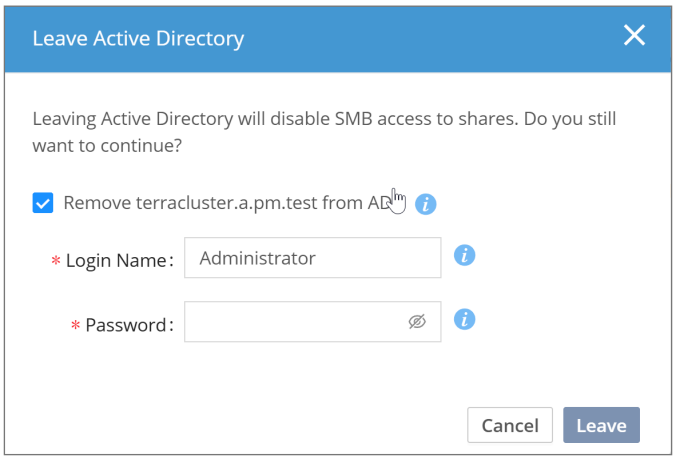


Figure 24. Leave Active Directory - Remove Computer Account

7.6.2. User Mapping Between Windows and Linux

Anvil stores a unified permission structure in its file system, enabling tight integration between Windows access control lists (ACLs) and NFS permissions. There is no option to choose what is stored, or to prioritize one over the other. The ACLs are mapped to NFS permissions whenever needed.

Anvil uses Active Directory as the source for mapping between the two environments. Anvil will read Windows credentials from Active Directory and fields such as uidNumber and gidNumber for the purpose of mapping between Windows identities and Linux. For Group Membership, when files are created, Anvil maps to the Primary Group assigned to the user in Active Directory.

A common UID/GID is required for user mapping to work; in other words, if your user has UID 5555 on the Linux client, 5555 must be entered in the uidNumber field in Active Directory to enable successful mapping.

The client computer should be connected to the same directory service, Active Directory being the most common. For Windows, this is standard; however, for Linux, this may not always be the case. Please follow the instructions for the respective Linux platform to join it to Active Directory. It is recommended (but not required) that the Linux environment is connected to Active Directory; Linux environments can be configured with other sources that provide identity to the user.

Default Behavior

The default behavior, without having user mapping configured, will make the environments look disconnected. For example, when saving a file from Windows, the default user, UID/GID 65534/65534, is used. This typically maps to nfsnobody on most Linux systems and is the owner of files and directories. This happens because the file was not saved with a UID/GID that is recognizable by Linux.

Mapping the UID and GID Within the Same Active Directory Domain

In Active Directory, each user and group object can also be configured with a UID (User ID) and GID (Group ID) mapping. This allows Linux users with their own UIDs/GIDs to work on files they own and, perhaps most importantly, be part of groups in Active Directory.

They are called *uidNumber* and *gidNumber* in the Active Directory object, as shown in the following screenshots:

© 2026 - Hammerspace

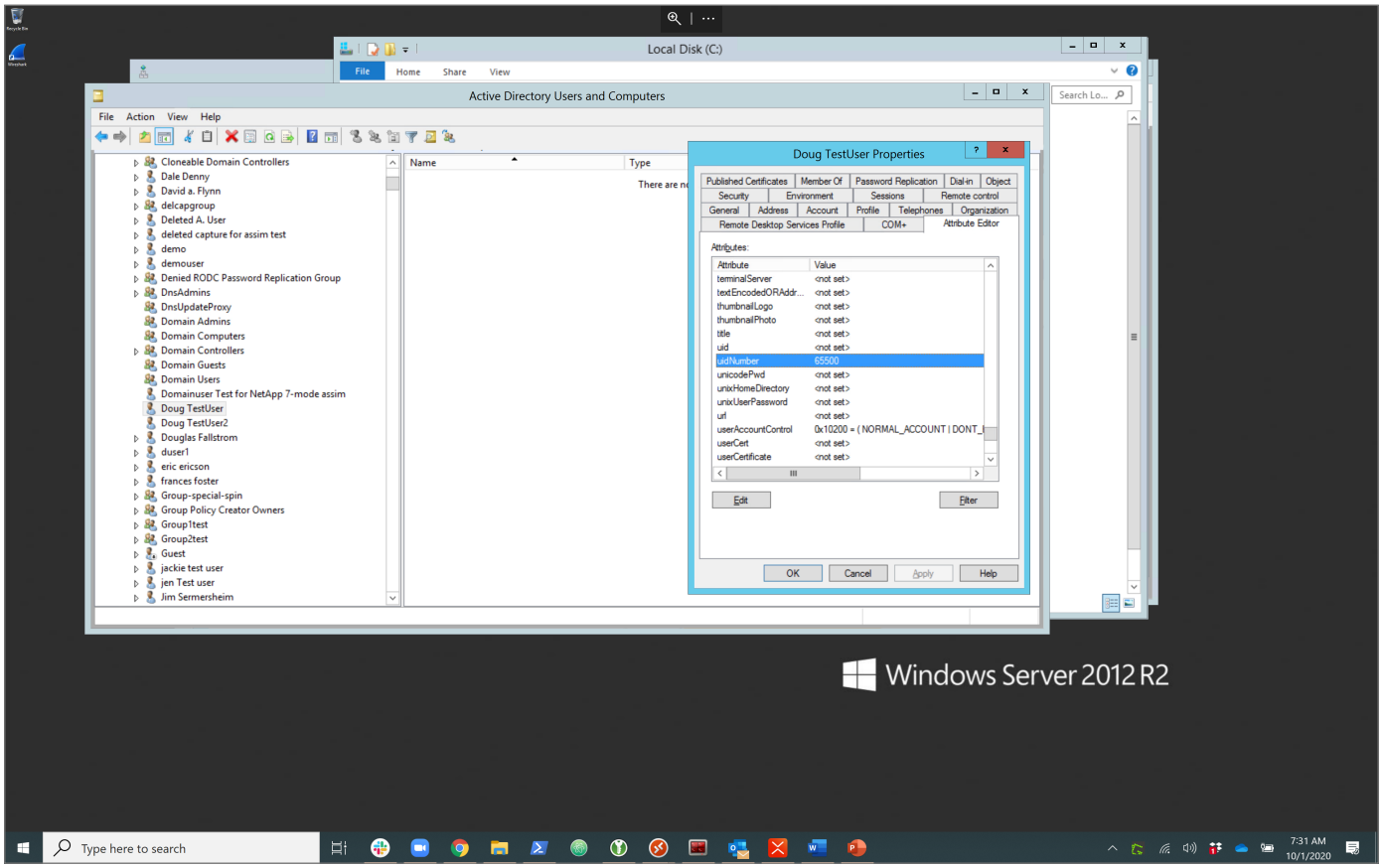


Figure 25. The uidNumber field in Active Directory

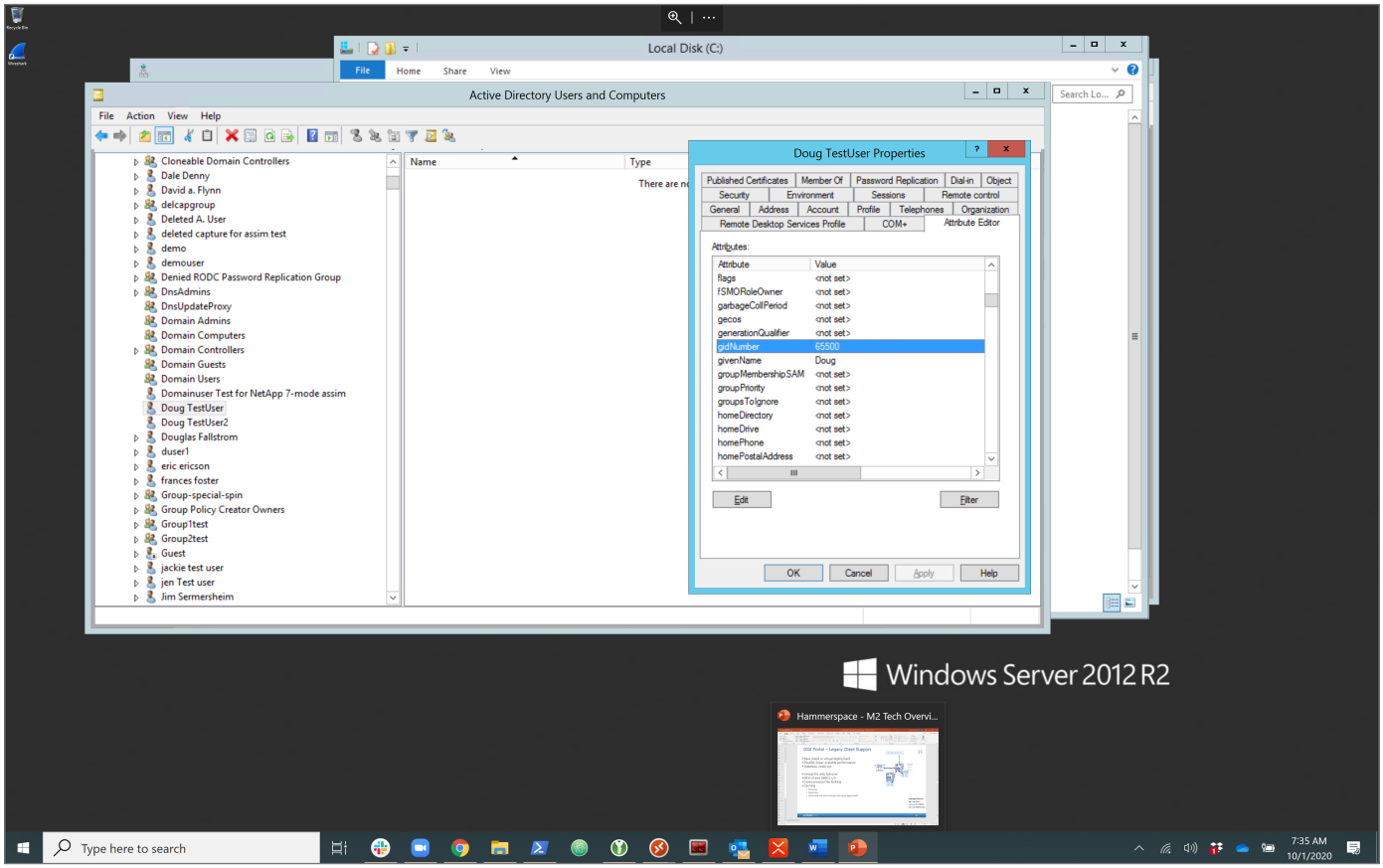


Figure 26. The gidNumber field in Active Directory

© 2026 - Hammerspace

If these parameters do not exist, consult with your Active Directory administrators regarding how to create them.

Cross-Domain Mapping

Hammerspace provides a mapping service that bridges separate Windows and Linux domains, enabling users with accounts in both domains to securely access their files. This is typically only used when the domains are separate. For user mapping within the same domain, please see the User Mapping section.

The following examples demonstrate how to set up cross-domain mapping.



Cross-domain mapping is not supported without joining Active Directory. Join Active Directory by following the steps in the Joining an Active Directory section.

Creating Mappings Between Domains

This example creates a bidirectional map between the **lin.ad.test** and **win.ad.test** domains. Repeat this action as needed to create additional cross-domain mapping.

```
domain-idmap-add --from lin.ad.test --to win.ad.test --attribute TestLNXWindowsAccountName
--bidirectional
```

Selecting a Preferred Domain

When creating shares that will be accessed by users from different domains use the `--preferred-domain` option to select the preferred domain for the share:

```
share-create --export-option *,rw,no-root-squash --preferred-domain win.ad.test --path /win1 --create
-path --name win1
```

Removing a Preferred Domain

Use the following command to remove a preferred domain from an existing share:

```
share-update --name win1 --preferred-domain-clear
```

Applying a Preferred Domain to a Share

Use the following command to apply a preferred domain to an existing share:

```
share-update --name win1 --preferred-domain lin.ad.test
```

Reloading Updated Domain-Mapping Rules

Although the domain mapping rules are updated periodically, you can use the following command to

© 2026 - Hammerspace

immediately reload the rules:

```
domain-idmap-reload
```

Reviewing Cross-Domain Mappings

Use the following command to review the cross-domain mappings:

Reviewing cross-domain mappings

Command:

```
domain-idmap-list
```

Expected output:

```
total 1
ID:                     cf0890b0-8c9b-492f-8d7b-58e043e05e19
From:                   lin.ad.test
Inherit from:           false
To:                     win.ad.test
Inherit to:             false
Attribute:              TestLNXWindowsAccountName
Bidirectional:          true
Order:                  1
```

7.6.3. Client Access to Data Using SMB

macOS Client Access

You can mount a share over SMB from macOS using the following steps:

1. Open **Finder**, click **Go,** and then click **Connect to Server.**

© 2026 - Hammerspace

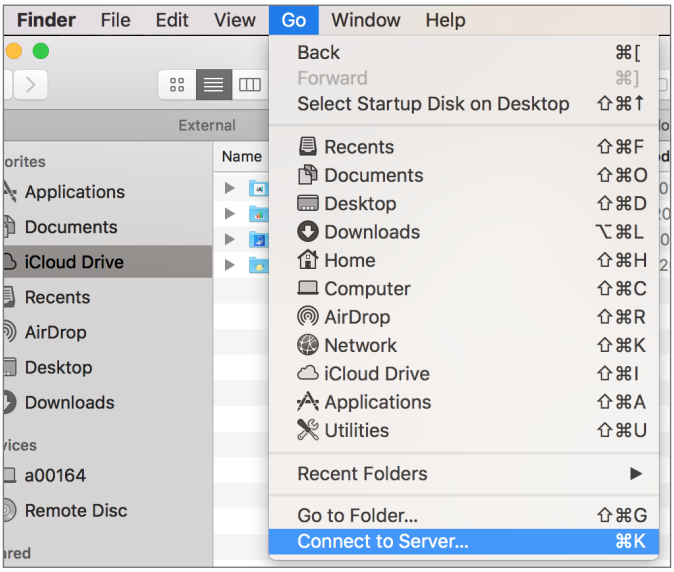


Figure 27. Location of Connect to Server control

2. Enter **smb://DSX DATA IP** or **DNS name /sharename**.

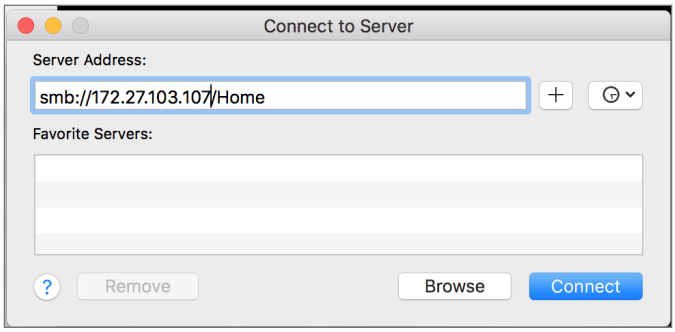


Figure 28. Enter server address to connect to server

3. Enter the credentials. The credentials are from Active Directory; users are not managed locally on the Anvil.

© 2026 - Hammerspace

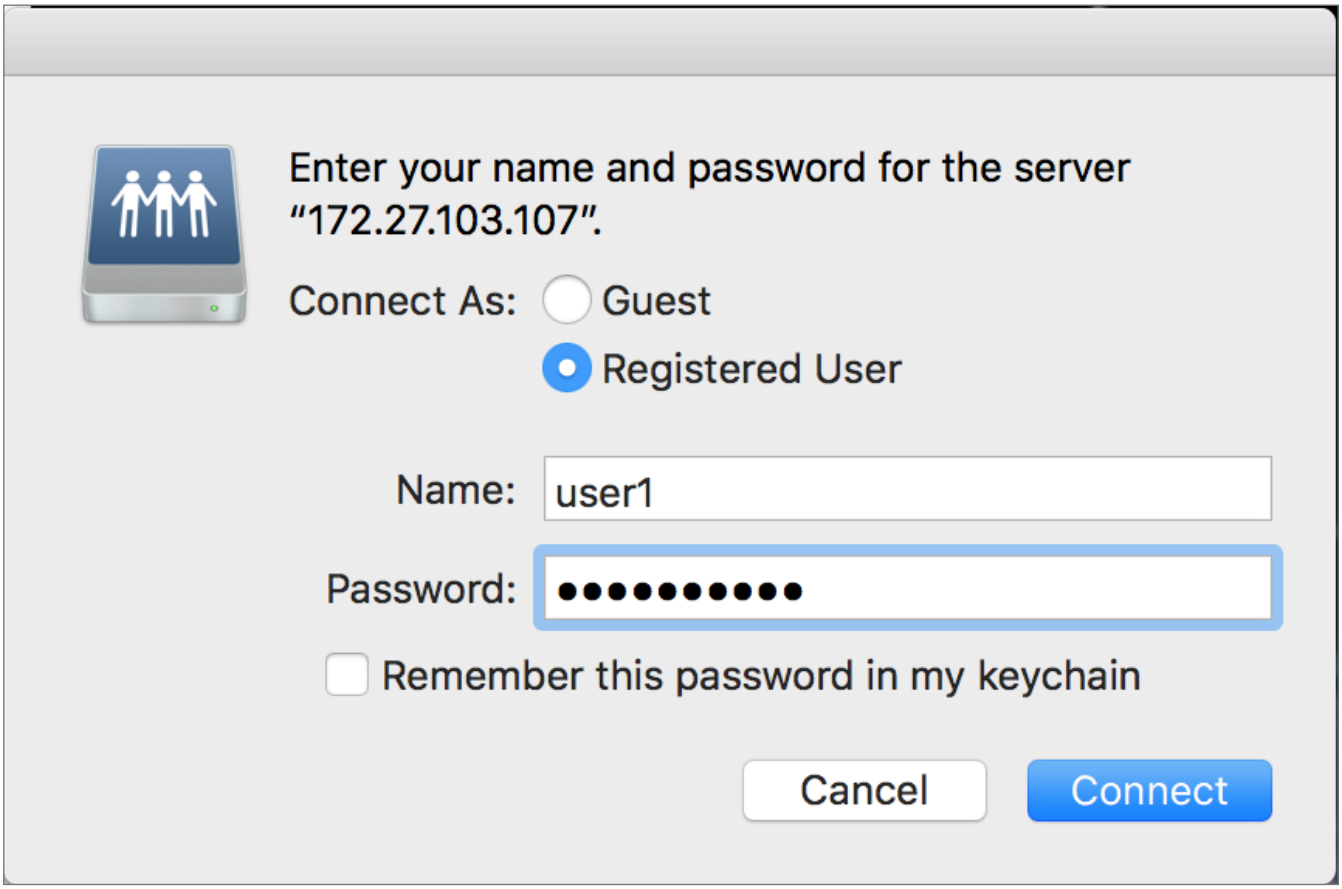


Figure 29. Enter Active Directory credentials

Windows Client Access

A share can be mounted over SMB from a Windows client by mapping a “Network Drive”. It is recommended to use the Fully Qualified Domain Name (FQDN) of the SMB Server when mapping or navigating the namespace.



When using the IP address of a DSX, SMB does not use Kerberos authentication. To use Kerberos authentication, you must access the share using the SMB Server Name that you entered when Anvil was installed. To do this, you must register the SMB Server Name in Active Directory and point it to the DSX data IP(s)/Floating IPs.

7.6.4. Load Balancing

When using multiple DSX nodes, it is recommended to include all the floating IPs in DNS. DNS servers typically rotate IP addresses in response to client requests. This will provide connections to be load-balanced across DSX nodes when clients connect to the SMB server name.

7.6.5. Mapping a Share

You can map shares using the Microsoft Management Console.

© 2026 - Hammerspace

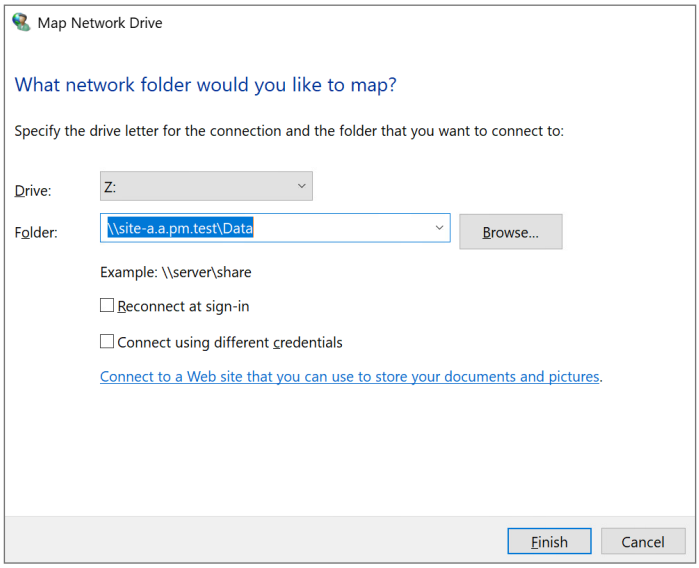


Figure 30. Mapping a network drive

The share is now mapped as a drive letter.

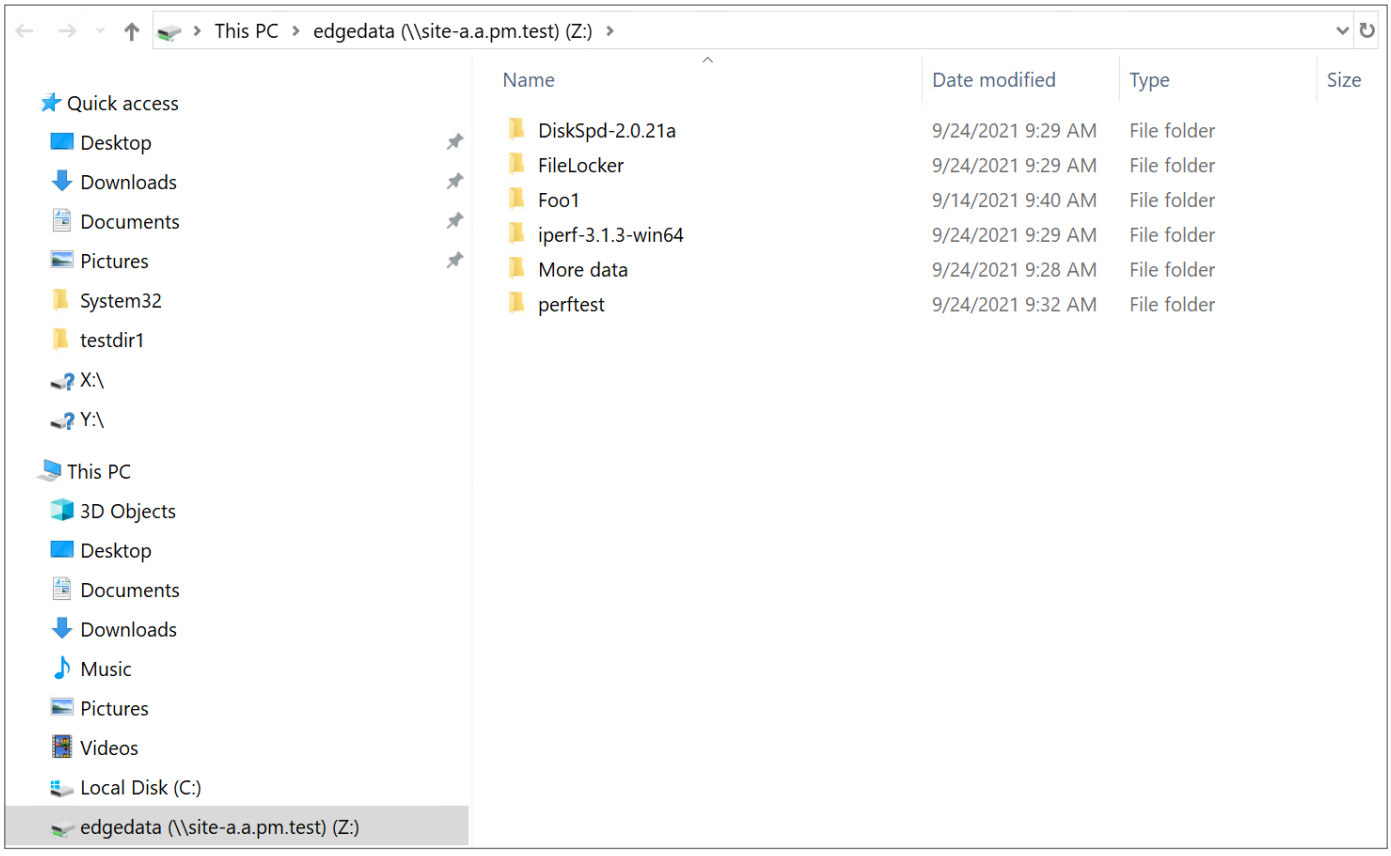


Figure 31. Share is mapped as a drive letter

7.6.6. Mapping a Drive Letter Using the Windows CMD

The following example shows the command and output:

Mapping a drive letter using Windows CLI

Command:

```
C:\Users> net use Z:\\site-a.a.pm.test\edgedata
```

Expected outcome

```
The command completed successfully.
```

7.6.7. SMB Multi-channel

SMB Multichannel, a feature included with Windows Server 2012 R2 and Windows Server 2012 and part of the Server Message Block (SMB) 3.0 protocol, improves the network performance and availability of file servers.

SMB Multichannel enables file servers to use multiple network connections simultaneously. It facilitates aggregation of network bandwidth and network fault tolerance when multiple paths are available between the SMB 3.0 client and the SMB 3.0 server.

This capability enables server applications to utilize the full bandwidth of the available network and makes them more resilient to network failures.

Requirements

- All the IP addresses for a multi-channel session must be on the same DSX node.
- The number of DSX nodes does not matter.



SMB Multichannel is a technology preview in Hammerspace 5.2. Enabling it requires assistance from Hammerspace Support.

SMB Multichannel lets an SMB 3.x client use multiple TCP connections — across multiple NICs, or multiple CPU cores on a single high-bandwidth NIC — for one SMB session, improving throughput and resiliency.

- Hammerspace advertises the Multichannel capability; the client decides whether and how to open additional connections. Client support and behavior are defined by Microsoft: SMB Multichannel is supported on Windows 8 and later and Windows Server 2012 and later. See Microsoft's documentation for the full client list and behavior: Microsoft SMB Multichannel.
- Multiple NICs are not required — a single high-bandwidth NIC benefits because the client opens multiple connections across CPU cores (RSS).
- For multi-NIC configurations, the DSX must present independent interfaces. SMB Multichannel over bonded interfaces is not supported.
- Enabling the feature alone does not guarantee higher performance; results depend on the client, NIC count/speed, and workload.

Enabling SMB Multi-channel with the GUI

Multichannel is enabled by default. This setting can be verified or changed in the Management GUI.

© 2026 - Hammerspace

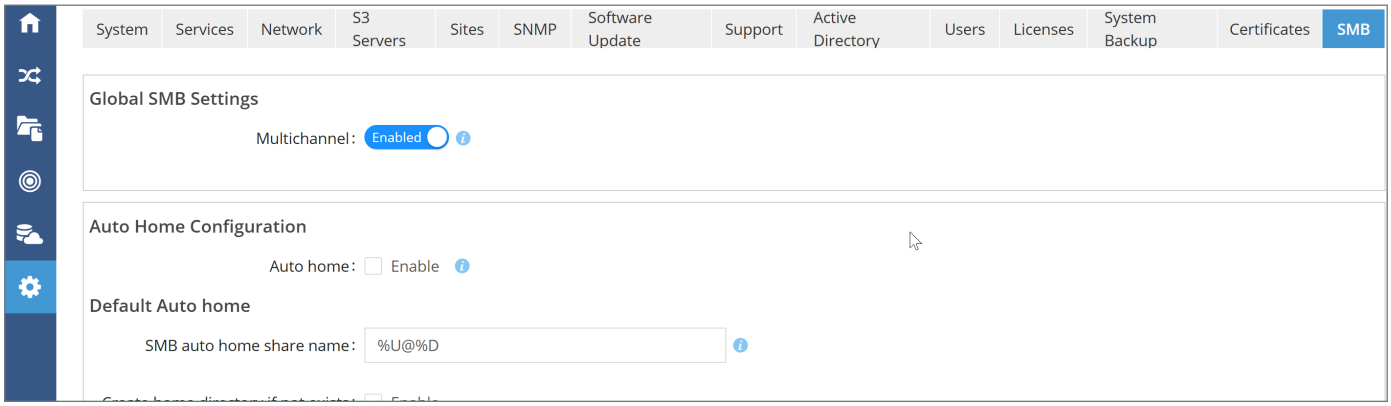


Figure 32. SMB Multi-Channel setting in the Management GUI

7.6.8. Viewing Mapped Shares Using the Windows CMD or PowerShell

Viewing mapped shares: Windows CLI and PowerShell

```
C:\Users> net use z: \\site-a.a.pm.test\edgedata

The command completed successfully.

PS C:\Users> net use

New connections will not be remembered.

Status Local Remote Network

-------------------------------------------------------------------------------

z: \\site-a.a.pm.test\edgedata

NFS Network

The command completed successfully.
```

7.6.9. Unmounting a Share Using the Windows CMD

Unmounting a share using Windows PowerShell

Command:

```
C:\Users> net use z: /delete
```

Expected outcome:

```
Z: was deleted successfully.
```


© 2026 - Hammerspace

7.6.10. Supported SMB Versions

The SMB client automatically negotiates the SMB protocol version and typically requires no management. The SMB server will automatically use the highest version offered by the client, but it will not negotiate below the minimum defined version. The table below represents the product defaults.

Table 14. Supported SMB protocol versions

SMB version	Status
SMB version 1	Supported but disabled by default
SMB version 2	Supported
SMB version 3	Supported
SMB version 3.1 and later	Supported

Do not use `smb-config --min-smb-client` to change the minimum SMB dialect. The option does not take effect. See the `smb-config` entry in the CLI Reference.



Hammerspace strongly recommends using SMB version 2 and later for Microsoft Windows SMB and Apple macOS.

7.6.11. Creating Hidden SMB Shares

To create hidden SMB shares and aliases, simply add a dollar sign $ at the end of the share name when creating the share. This will hide the share when browsing the network.



Hidden shares are sometimes referred to as *administrative shares*.

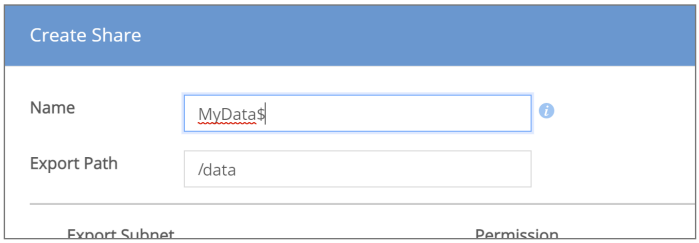


Figure 33. Creating a hidden share

7.6.12. Setting the Browsable Option

An alternative to naming hidden shares with a dollar sign (**$**) is to enable the *browsable* option. You can allow the browsable option on a share and SMB alias on a granular level.

Using the GUI

1. Navigate to **Data** and locate the share you want to hide.
2. Click the **Edit** icon on the same row as the share.
3. Select the **SMB** tab and click on the **Edit** icon for the share.

© 2026 - Hammerspace

4. Click the **Browsable** check box so that it displays a check mark, and then click **Update**.

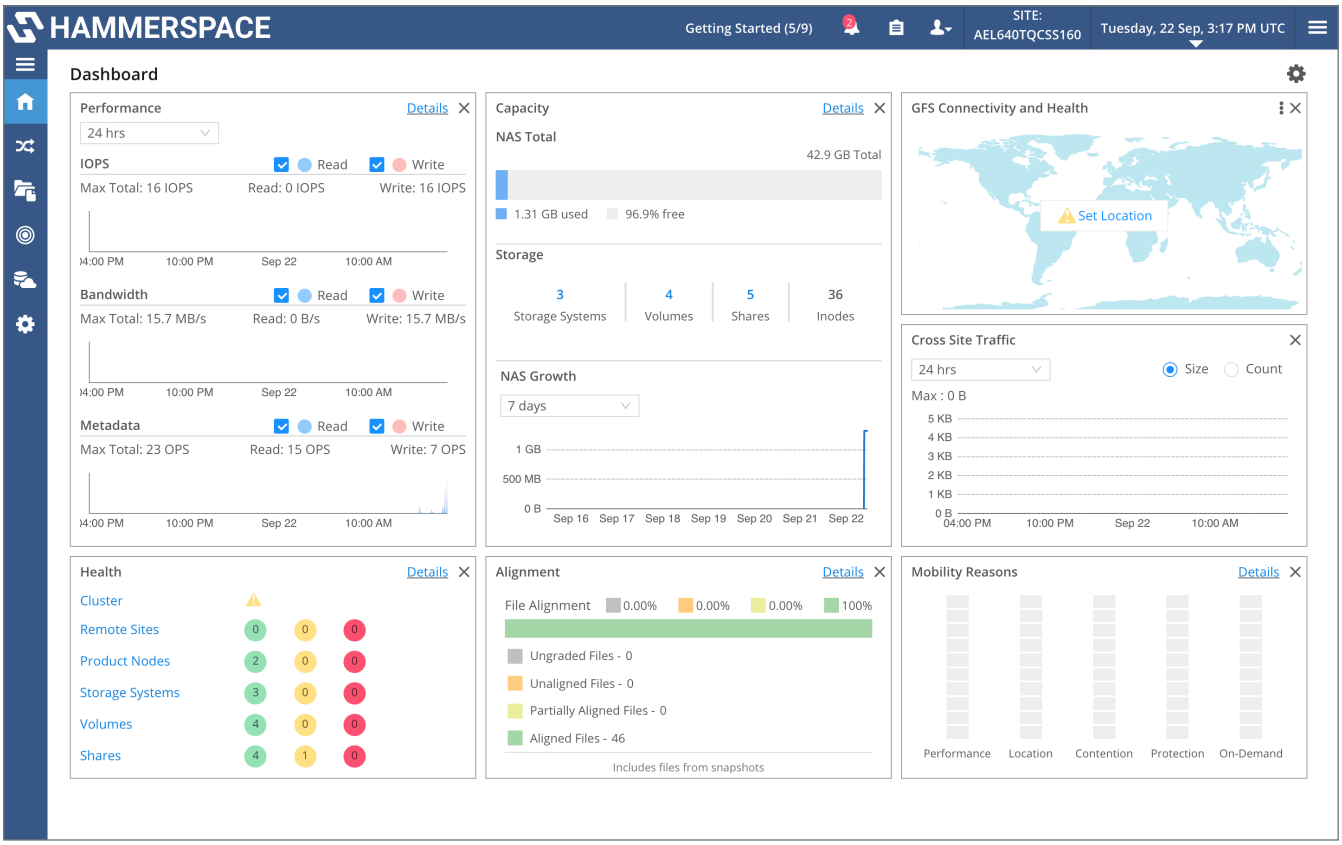


Figure 34. Making the SMB alias browsable or not browsable



The browsable checkbox has three states: **Inherited**, **checked**, and **unchecked**. If the browsable checkbox is solid blue, the share is already browsable; however, you can still make this SMB alias not browsable within the otherwise browsable share.

Using the Admin CLI

The browsable option can also be configured using the `share-update` command.

```
share-update --name <share-name> --smb-browsable-<on|off>
```

7.6.13. Setting Aliases for SMB

SMB aliases can be used to configure additional SMB shares on the system. Typically, the SMB alias is a sub-folder of an existing share, but it can also represent the entire share with a different SMB name. An SMB Alias does not have its own ACL or permissions; it will have the same ACL as either the directory it is pointing to or of the root of the share if it is pointing to `/`.

To add an SMB alias, open the **Edit Share** workflow and navigate to the SMB tab. Click on **Add Alias** and fill in the required details.

© 2026 - Hammerspace

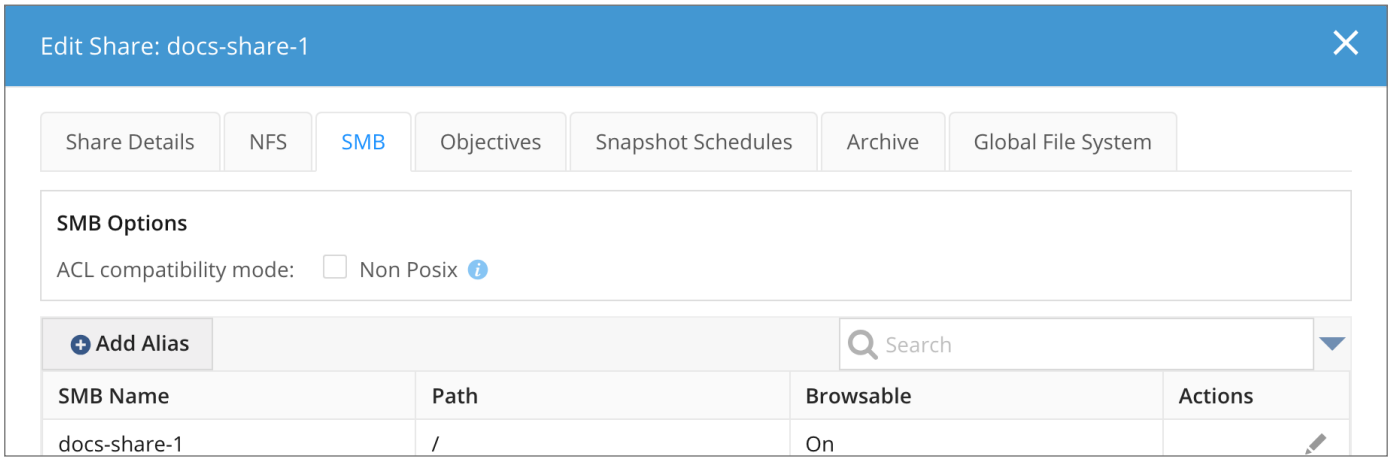


Figure 35. Add an SMB alias

You can manage SMB aliases using the Admin CLI. After a share is created, `share-update` can be used to add, update, remove, and clear aliases.

```
share-update --smb-alias <SMB name>,<PATH within the share>
```

```
share-update --smb-alias-clear
```

Adding Aliases to an Existing Share

You can use the `share-update` command to add or remove aliases on an existing share using the `--smb-alias`, `--smb-alias-add`, `--smb-alias-delete`, and `--smb-alias-clear` options.

1. Start by creating a share:

   ```
   share-create --name Prod --path /prod
   ```

   When browsing the SMB endpoint, the single share is displayed:

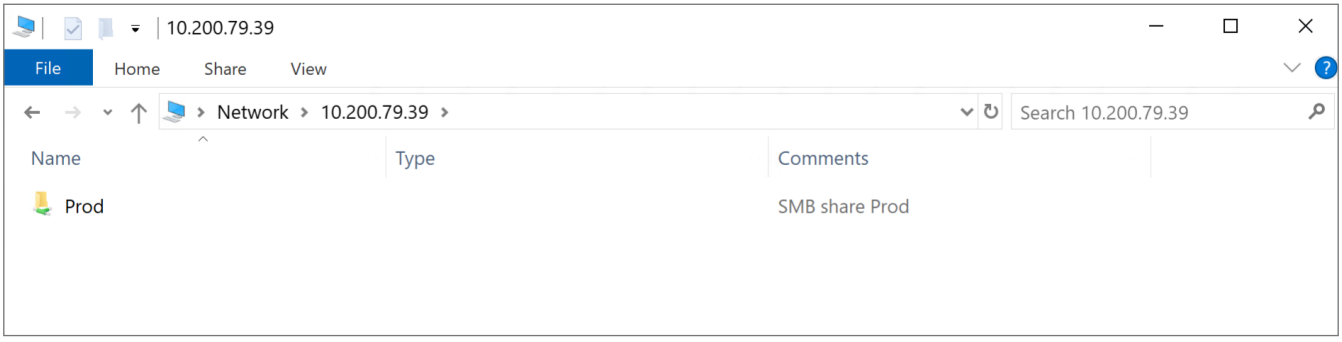
   

   Figure 36. SMB endpoint displaying a single share

2. Now create additional SMB aliases using the `share-update --smb-alias` option. SMB clients can mount these aliases directly. The following examples point back to the root of the share:

© 2026 - Hammerspace

a. Create two additional SMB shares by specifying the option twice:

```
share-update --name Prod --path /prod --smb-alias Production,/ --smb-alias ProdStuff,/
```

When browsing the SMB endpoint, you can see that the two additional shares have been created:

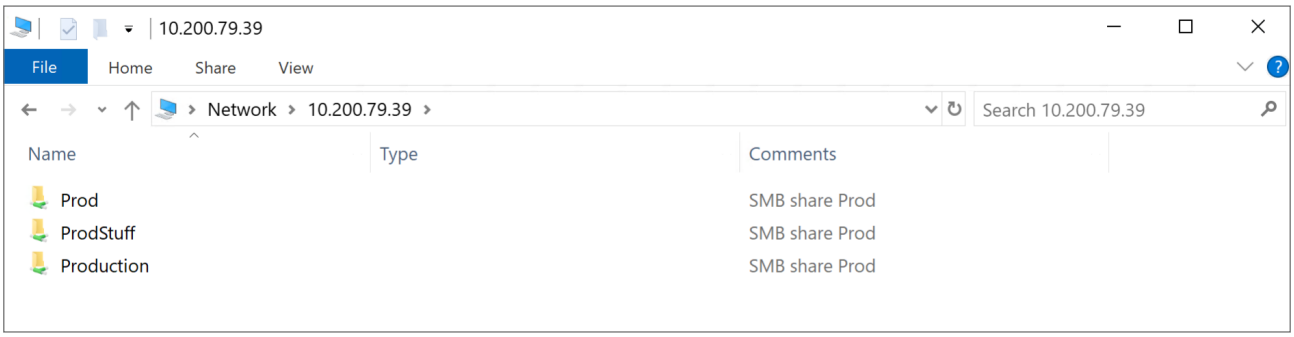


Figure 37. Shares visible in an SMB endpoint

b. Optionally, to add an SMB endpoint with spaces, enclose the information using quotation marks, `" "`. The same applies when using spaces in the path:

```
share-update --name Prod --path /prod --smb-alias "My Prod Data,/"
```

When browsing the SMB endpoint, you can see that the SMB endpoint with spaces has been created:

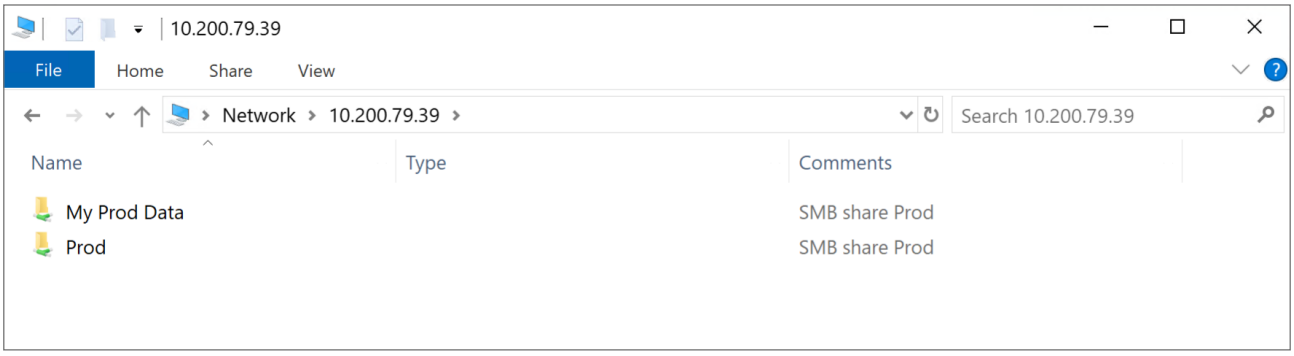


Figure 38. Share names with spaces

3. You can also specify a path within the share. Note that all paths used must be created in advance; otherwise, the `share-update` command will fail.

   The following command adds three SMB endpoints to an existing share, each pointing to a different subdirectory within the share.

```
share-update --name Prod --smb-alias Prod-Data,/data --smb-alias Binaries,/bin --smb-alias
UserData,/space/userdata
```



All paths start with a forward slash (`/`). This is a relative reference within the share, since `/` is the root of the share itself.

© 2026 - Hammerspace

When browsing the SMB endpoint, you can see that all the endpoints have been created:

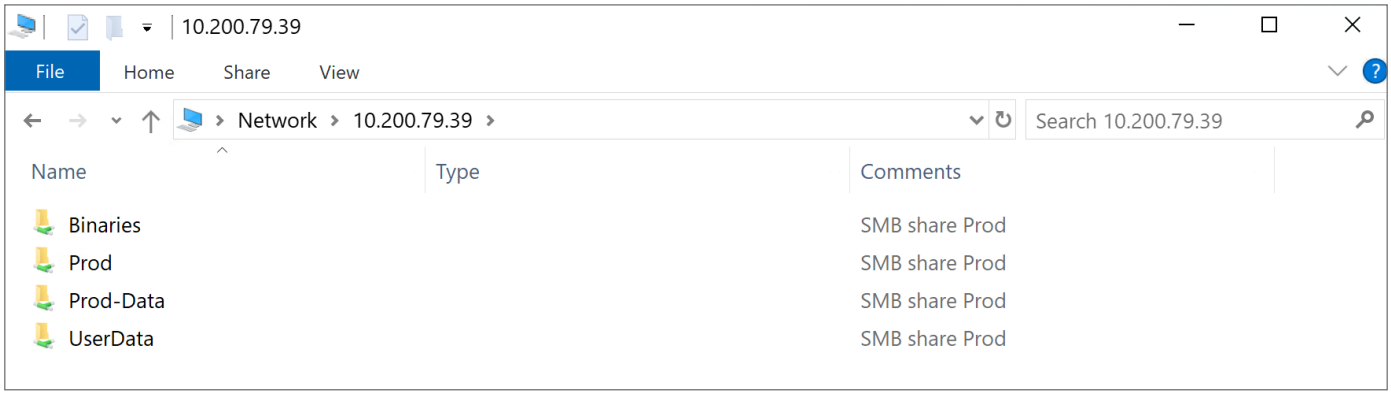


Figure 39. Share edited with three additional SMB endpoints

Listing SMB Aliases

Use `share-list` to list SMB aliases for a share.

```
share-list --name <share name>
```

If a share has an alias configured, there will be a new field called **SMB Aliases** with all the entries specified for that share. Each entry is separated by a space.

Removing Aliases

All SMB aliases can be removed by using the `share-update` command with the `--smb-alias-delete` option. Note that this will disconnect users who are using the SMB endpoint being deleted.

Deleting an SMB alias will not delete the share itself, nor will it delete any data or folders associated with it.

```
share-update --share-name Prod --smb-alias-delete Prod-Data
```

To remove all aliases for a share, execute `smb-alias-clear` without any options.



This will disconnect users who are using those SMB endpoints and should be used with caution.

```
share-update --share-name Prod --smb-alias-clear
```

Additional Details

- SMB alias names must be unique in the system; the same SMB alias cannot be used for other shares within the same cluster.
- SMB aliases do not support share ACLs as they are not shares themselves. If the alias is pointing to the root of the share, then it will have the same share ACL as the share itself.

© 2026 - Hammerspace

- SMB aliases are not replicated or created on remote sites when configuring a global file system. If your goal is to have the same SMB aliases on other sites, they must be added manually using the `share-update` command.
- A maximum of 100 SMB aliases can be created per share.
- The maximum SMB alias length is the same as the maximum length of an SMB share name.
- $ indicates it is a hidden share. When creating an alias ending with a $ sign, the share will not be visible when browsing the SMB endpoint. The following example adds a hidden SMB share to the existing list:

```
share-update --name Prod --smb-alias --smb-alias-add HiddenProd$,/
```

When viewing the share from the SMB endpoint, the hidden SMB share is not visible:

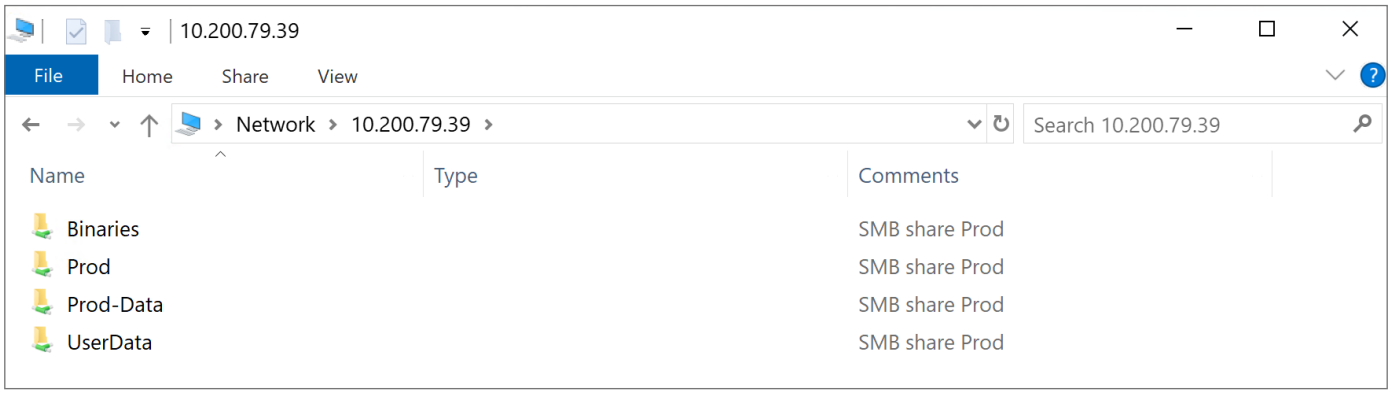


Figure 40. Hidden SMB share not visible

However, it can be mounted by specifying the name directly:

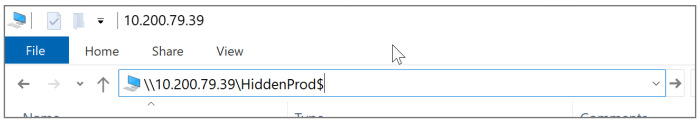


Figure 41. Hidden SMB share mountable by entering the name directly

7.6.14. Allowing Access-Based Enumeration

Access-based Enumeration (ABE) allows for files and folders to be hidden when users do not have list or read permissions to access the objects. Proper management and care must be taken to ensure that inheritance does not unintentionally open or lock down directories deep within the share.

ABE can be enabled at the share/directory and file-granular levels using objectives. The access-based-enumeration objective is pre-created and can be applied easily through the GUI.

The objective can be applied at the share level, with applicability, meaning it cannot be removed further down in the share.



If the goal is to enforce ABE for all folders and files *in an entire share*, it is always recommended to use a share-level objective and applicability.

© 2026 - Hammerspace

ABE can also be applied using regular objective expressions at any level within the share. It can be applied at the file level or a sub-directory level, using either the management interface (UI/CLI) or the **hstk** toolkit.

ABE is applied by Hammerspace regardless of protocol: when user mapping is properly configured, directory listings returned over NFS are filtered in the same way as over SMB. ABE is a filter on directory listings, not an access control. It decides which names appear when a user lists a directory. File and directory permissions still control what each user can open, read, or change.

ABE gives each user a correct view over NFS only when each client is used by one user. NFS clients, including Linux and macOS clients, cache directory listings for each mount rather than for each user. When several users on one client list the same directory, the client can show later users the listing it received for the first user:

- A user can see the names of files and directories that ABE should hide from them. Their permissions still prevent them from opening those files and directories.
- A user can be missing files and directories they do have access to.



The wrong listing stays until something on the client invalidates its cache, such as a write to that directory. This affects every NFS version, including NFSv3 access through a DSX node: with the `do-not-cache` objective, the DSX node does not cache directory listings, but the client still caches the listing it receives. The `do-not-cache` objective is still required with ABE. NFS mount options such as `noac` and `lookupcache=none` do not prevent the client from caching listings.

To give each user a correct view, do one of the following:

- Give each user their own client, with only that user working on it and its own NFS mount. The client can be a physical machine, a virtual machine, or a container.
- Have the users access the data over SMB, where ABE filters the listing for each user.



When applying the ABE objective, it is also necessary to use the `do-not-cache` objective. Otherwise, caching on the DSX node may occasionally defeat the ABE functionality.

Enabling ABE for an Entire Share Using the GUI

1. In the Management GUI, navigate to the share listing and press the edit icon for the share.

© 2026 - Hammerspace

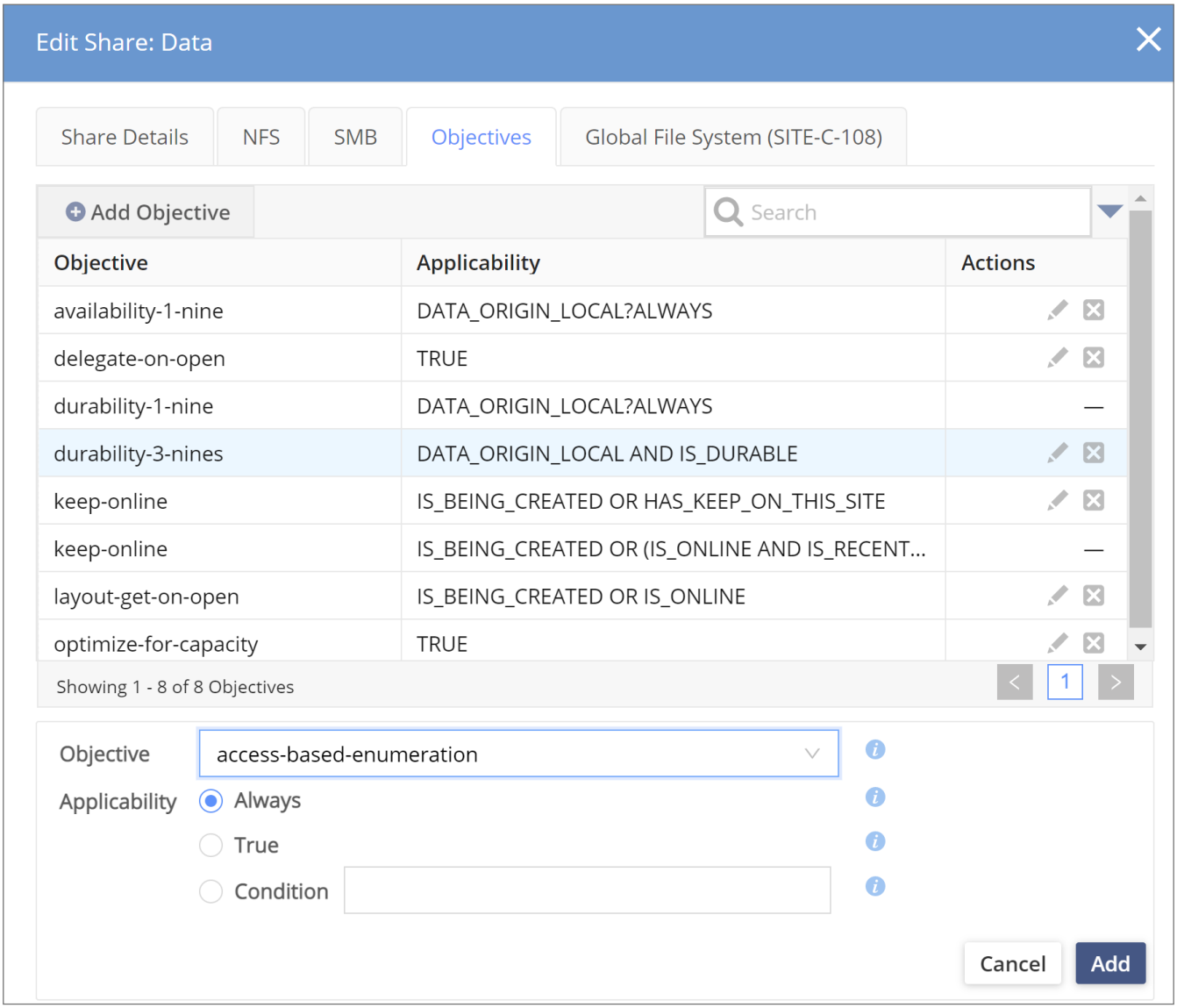


Figure 42. Share-level objective for access-based enumeration (ABE)

Click **Add Objective** under the Objectives tab. Just like any objective in the system, it can be added using a selection mechanism or to all the data. **Always** ensures that it cannot be disabled farther down in the share while **True** could be over-ruled by other objectives within the share.

Enabling ABE for a Sub-folder Using the GUI

Navigate to the file browser in the GUI and select the folder to apply the `access-based-enumeration` objective. Note that if this is configured on existing data, it may take some time to be effective. The time depends on how many files are pre-existing in the sub-directory where the objective is applied.

© 2026 - Hammerspace

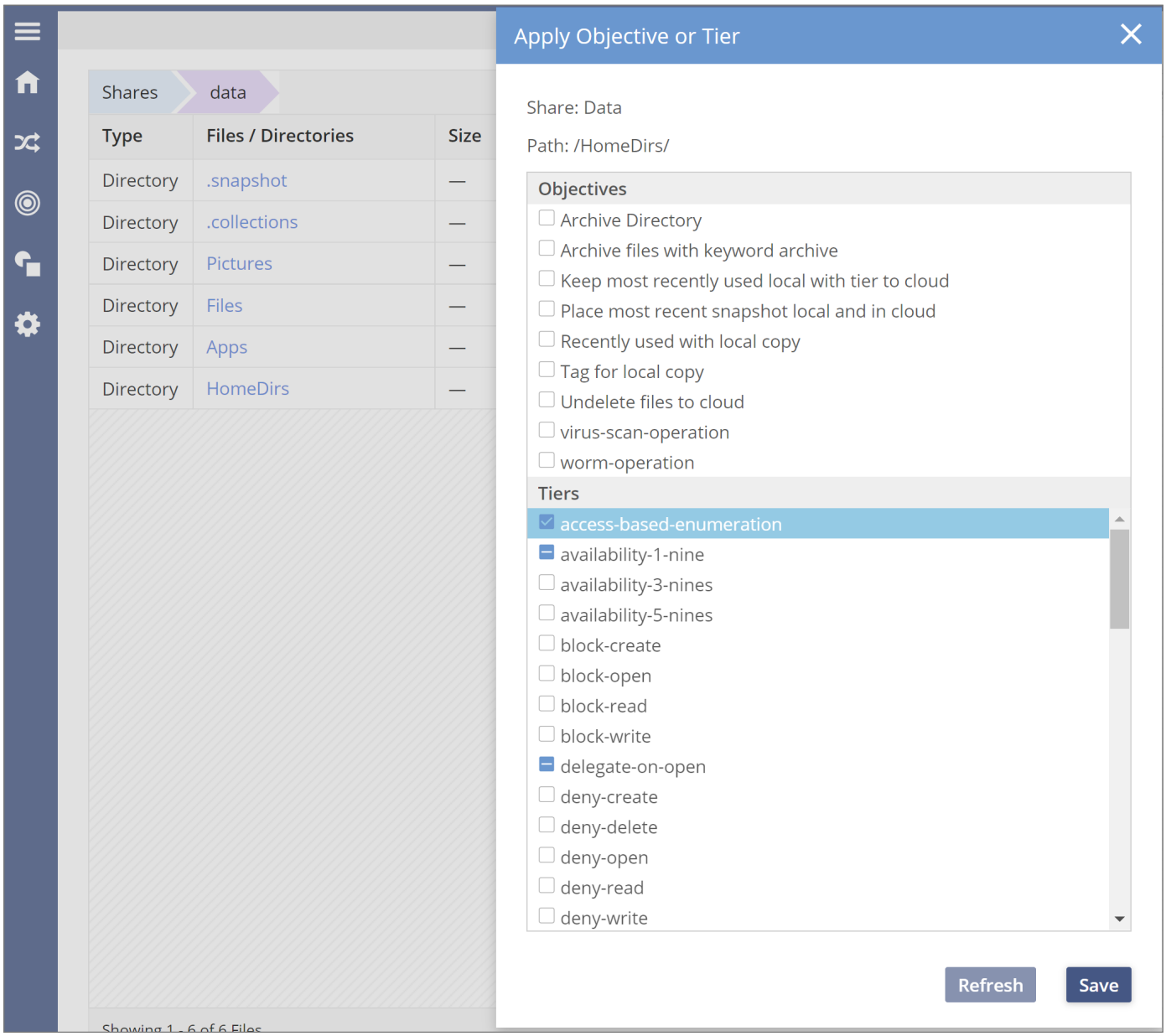


Figure 43. Subfolder objective for access-based enumeration (ABE)

Enabling ABE Using Objective Expressions

It is also possible to enable ABE using an expression as part of the objective. You can do this with the Hammerspace toolkit (**hstk**) in the Admin CLI.

This example uses the **hstk** utility (from a Linux or Windows client) to enable ABE when a file or folder has the label **"hidemyfile"** within the sub-directory **HomeDirs**. This enables users to enable ABE without requesting that an administrator enable it for the entire share.

```
# hs objective add -e 'HAS_LABEL(\"hidemyfile\")' access-based-enumeration HomeDirs/
```


© 2026 - Hammerspace

7.6.15. Using the Administrators Group

The purpose of the built-in Administrators group is to give a set of administrators elevated privileges to configure or fix issues with permissions.

When joined to Active Directory, the group Domain Admins is automatically added to the built-in group.

Individual users or other groups can be added to the Administrators group and receive the same privileges as Domain Admins. This should be done with caution.

Viewing Existing Members of the Administrators Group

Viewing Admin group members

```
builtin-group-user-list --group Administrators

Domain Admins
```

Adding a New Group to the Administrators Group

Adding new group to Admin group

Command:

```
builtin-group-user-add --group Administrators --username PMHSADMINS@a.pm.test
```

Expected output:

```
success
```

Adding an Individual User to the Administrators Group

Adding an individual user to Admins group

Command:

```
builtin-group-user-add --group Administrators --username pmuser10@a.pm.test
```

Expected output:

```
success
```

Removing Members of the Administrators Group

Note that `@<domain>` must be added to the group or username; otherwise, it will not be found.

Removing members of Admins group

© 2026 - Hammerspace

Command:

```
builtin-group-user-remove --group Administrators --username pmuser10@a.pm.test
```

Expected output:

```
success
```

7.6.16. Windows Previous Versions

The following capabilities are enabled when using Windows Previous Versions:

- Users can self-service data recovery using Windows Previous Versions functionality. Hammerspace will display snapshots, versioning, undelete, and log-xfer files as versions in a file.
- Users can right-click on a file or folder and restore or navigate to a previous point in time.

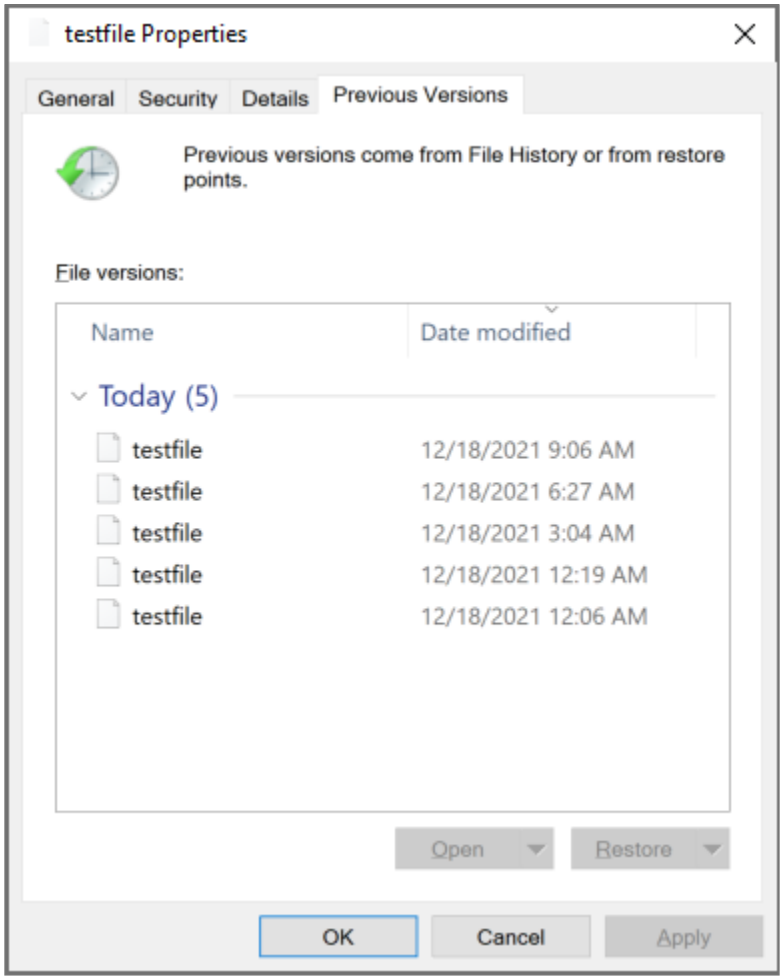


Figure 44. Windows file properties: Previous Versions tab



It is not recommended to restore the entire share using Windows Previous Versions as it will take longer than using the built-in Product functionality of share-restore. For very large

© 2026 - Hammerspace

> restore operations, it is recommended to use the Product interface for faster recovery.

7.6.17. Using Microsoft Management Console

Microsoft Management Console (MMC) can be used to manage functionality using native Windows tools. It allows customers to manage common operations directly from their Windows workstations.

The following operations are supported with MMC:

- Manage (change and view) share level permissions
- View open files
- Close open files
- View sessions
- Close sessions

Managing Share-Level Permissions

Hammerspace recommends using Microsoft management tools to view or change the share-level permissions on shares.

The default permissions of a share are set to allow *everyone* to read and write to the share. It is recommended that this be configured to meet the customer's best practices.

Using the Microsoft Management Console to Set Share Permissions

1. Open up MMC on a Windows client and Add or Remove **Snap-ins** › **Shared Folders**.

© 2026 - Hammerspace

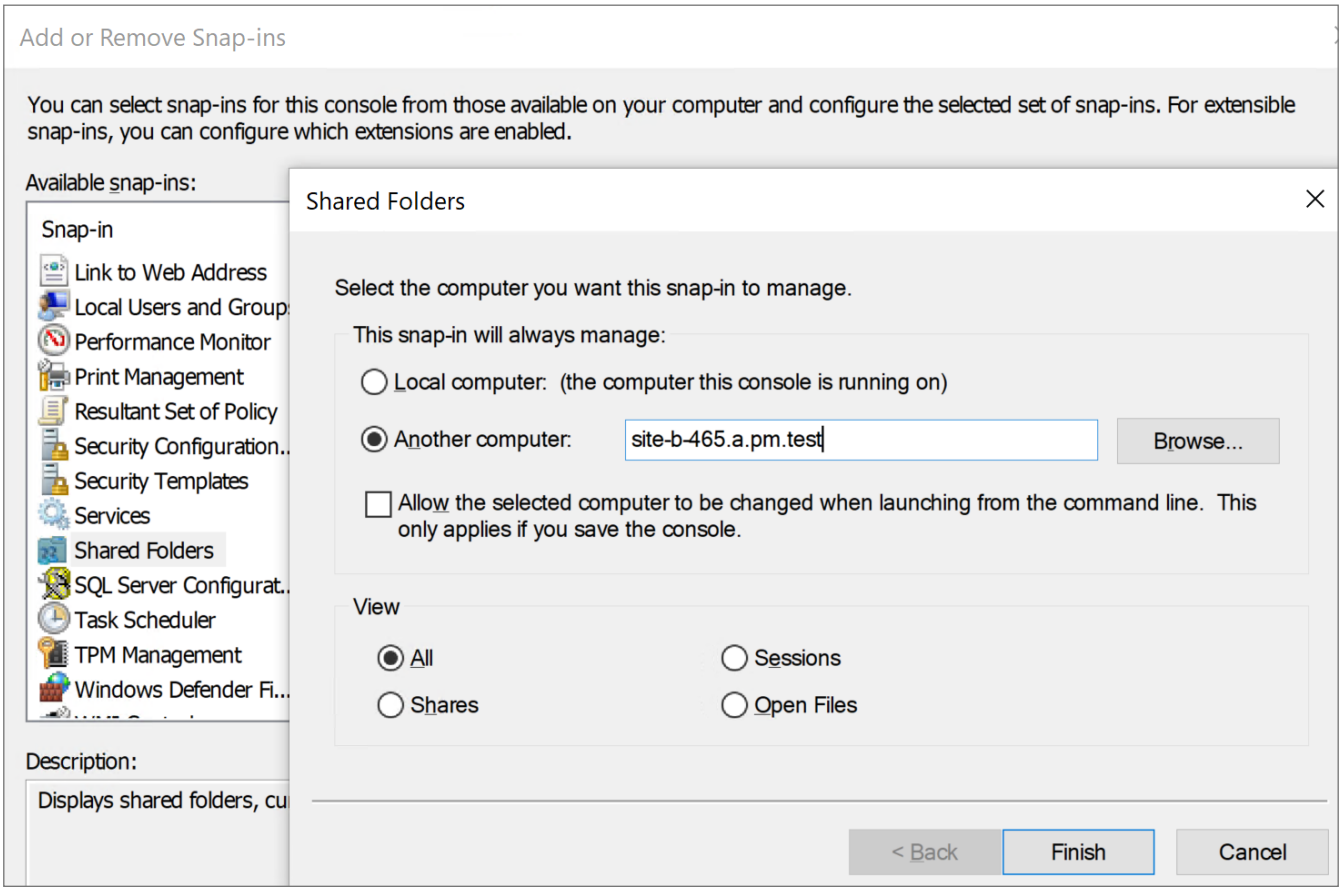


Figure 45. Connect to SMB server name (FQDN)

2. Type in the FQDN of the DNS-registered endpoint for the SMB server name.
3. Once connected, expand **Shared Folders** › **Shares** to see the shares.

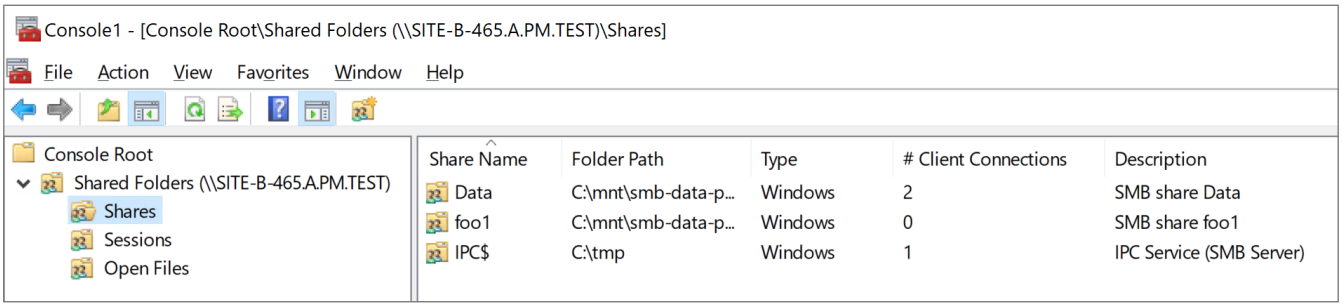


Figure 46. Share listing in Microsoft Management Console (MMC)

4. Right-click on the share and select **Properties** › **Share Permissions** tab.

© 2026 - Hammerspace

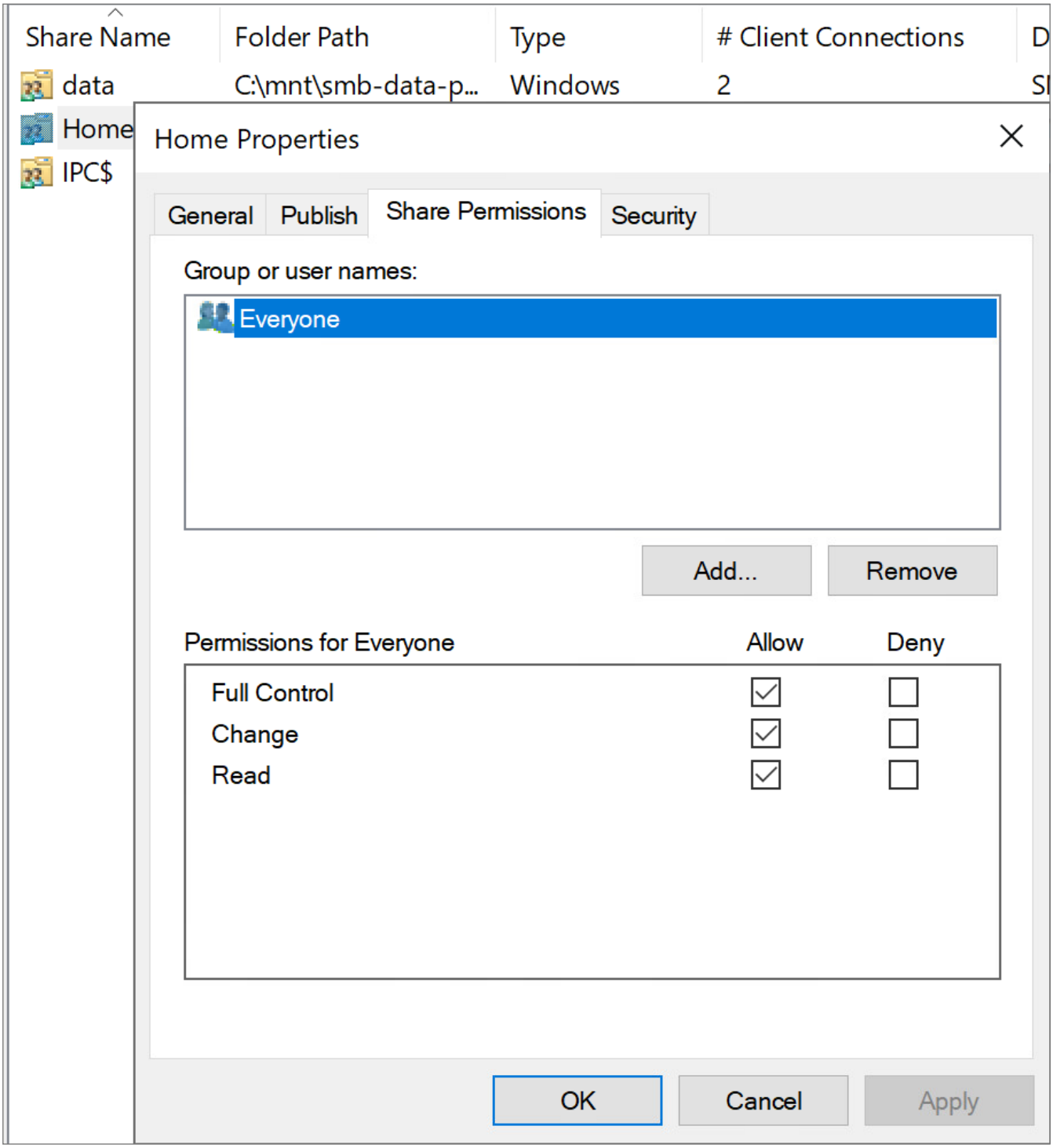


Figure 47. Share permissions

5. The default permissions are open for everyone. Configure the settings according to desired best practices.

Viewing and Closing Files

At times, it may be required to help resolve client issues by closing open files. This can be done directly from the MMC.

1. Open the MMC on a Windows client and under File, select **Add/Remove Snap-in**.

© 2026 - Hammerspace

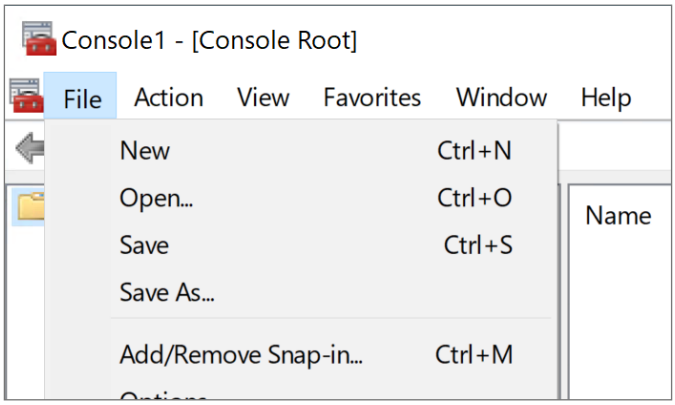


Figure 48. Location of Add/Remove Snap-in control

2. Type in the hostname of the DNS-registered DSX endpoint or the IP address of one of the DSX nodes.

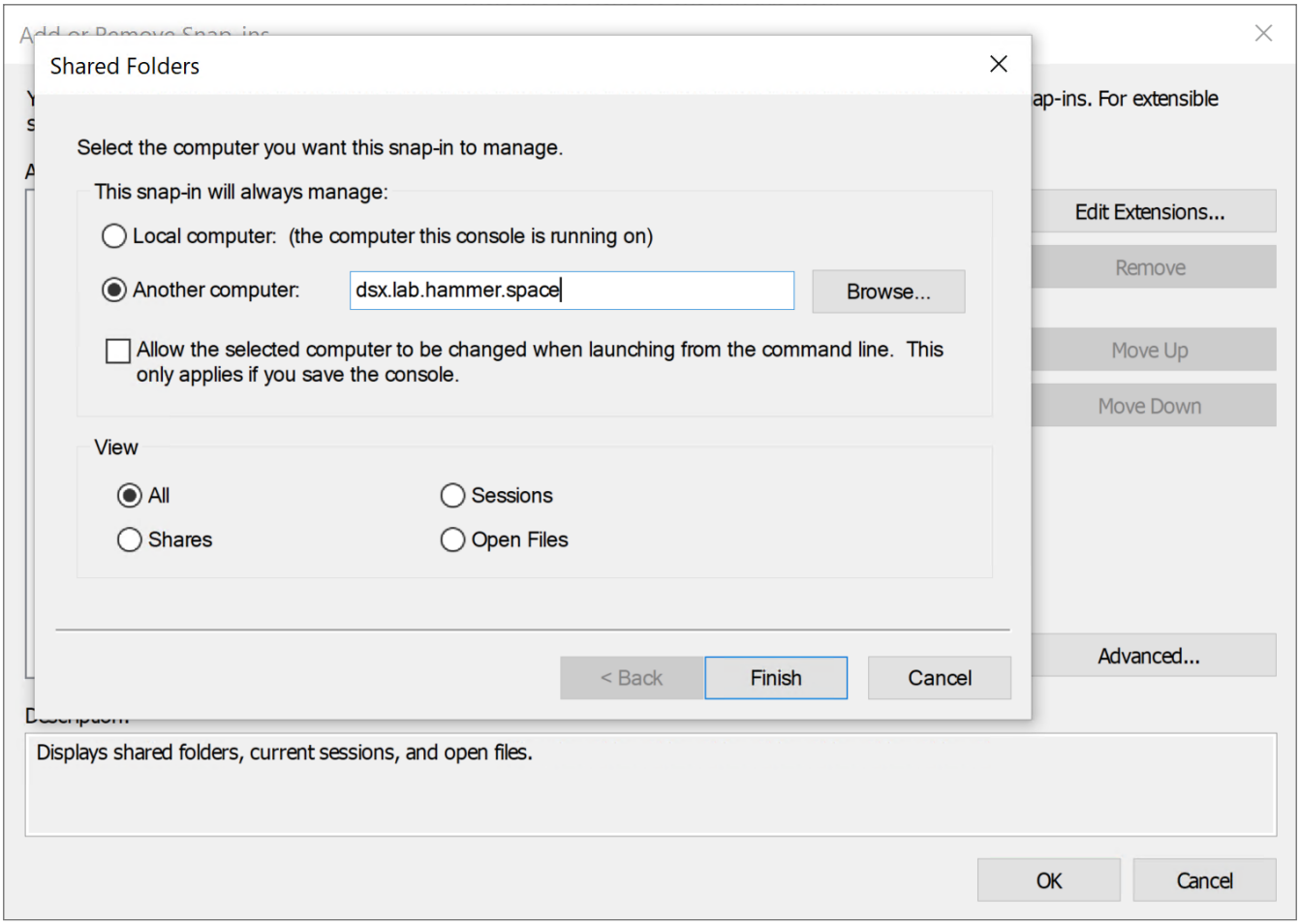


Figure 49. Associating Snap-in with the DSX node

3. Once connected, expand **Shared Folders › Open Files** to view which files are currently open. The example below highlights a file currently open.

© 2026 - Hammerspace

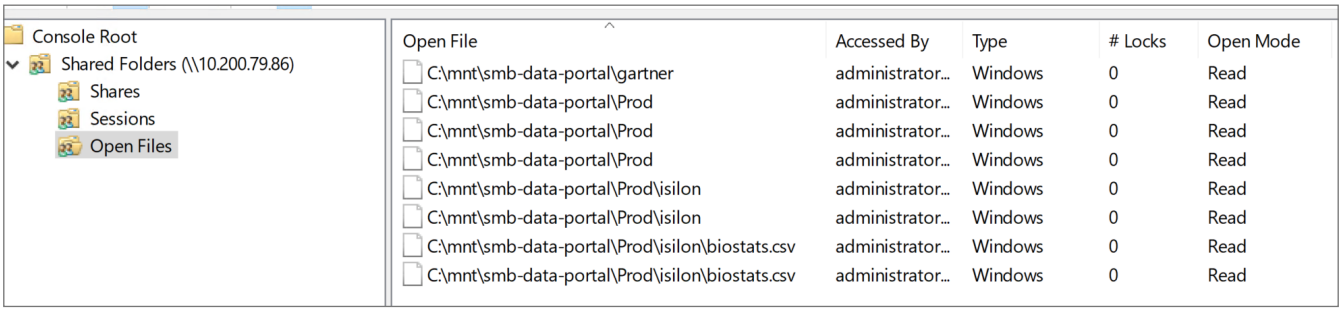


Figure 50. Viewing open files

4. Right-click on the file and select **Close Open File** to close the file.

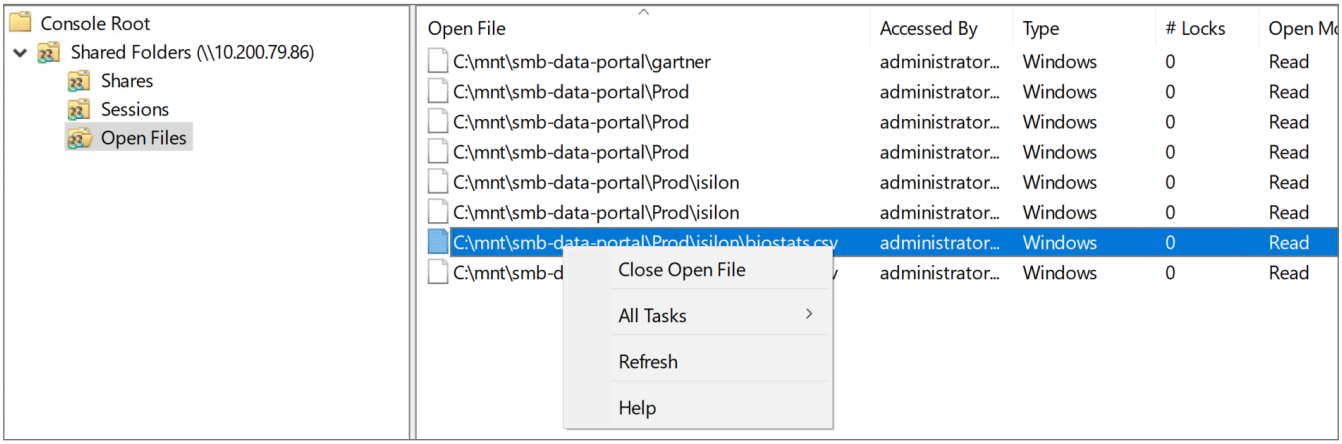


Figure 51. Closing an open file

Viewing and Closing Sessions

At times, it may be necessary to assist in resolving client issues by closing their sessions. This can be done directly from the MMC.

1. Follow the instructions in **Viewing Open Files** to connect to Hammerspace.
2. Once connected, expand **Shared Folders** › **Sessions** to see the clients that are currently connected.

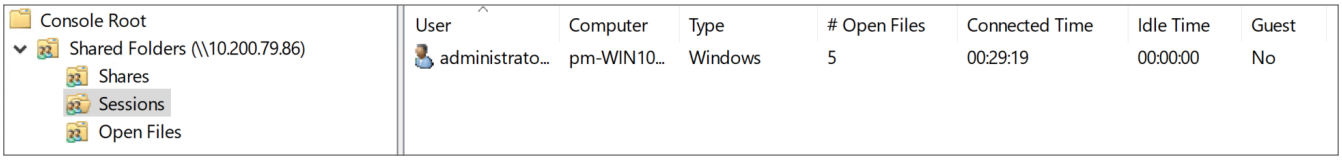


Figure 52. Viewing open sessions

3. Right-click on a session and select **Close Session** to close the session.

© 2026 - Hammerspace

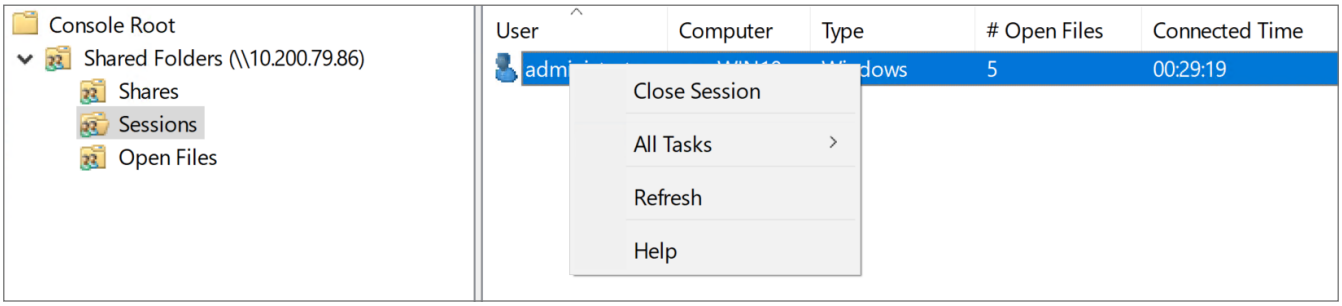


Figure 53. Closing an open session

7.7. Active Directory Permissions

Audience: Active Directory administrators who must delegate the rights that Hammerspace needs, and Hammerspace administrators who need to tell their AD team what to grant.

Hammerspace does not require Domain Admin to join Active Directory. This page lists the minimum permissions for a basic join, then the additional permissions needed for each optional capability, so that a site can grant only what it actually uses. Permissions are named as they appear in the Active Directory Delegation of Control Wizard and the Advanced Security Settings dialog, so they can be followed directly by an AD administrator.

7.7.1. Two Accounts Are Involved

Understanding this distinction makes the rest of this page straightforward. Most delegation mistakes come from granting a permission to the wrong one.

Account	What It Is	When It Is Used
Join account	The Active Directory user name and password an operator supplies when joining, leaving, or re-saving the Active Directory configuration.	Only during those administrative operations. Hammerspace does not store it for ongoing use.
Computer account	The computer object Hammerspace creates in Active Directory to represent the cluster.	Continuously, after the join, for routine work such as DNS registration, SPN maintenance, and Kerberos ticket handling.

Some permissions below are held by the join account and some by the computer account. The **Held By** column identifies which.

7.7.2. Minimum Permissions: Joining with a New Computer Account

This is the baseline. It covers a first-time join where Hammerspace creates the computer account itself, and it is enough for the join to complete, for Kerberos to be configured, and for the standard service principal names to be registered.

© 2026 - Hammerspace

Permission	Scope	Held By	Why It Is Needed
Ability to authenticate and read the directory	Domain	Join account	Hammerspace validates the supplied credentials and reads basic domain information before making any change.
Create Computer objects	The organizational unit specified during the join; if none is specified, the default `Computers` container	Join account	Hammerspace creates the computer object that represents the cluster.
(Automatic) Full control of the object it just created	The new computer object	Join account	Active Directory grants the creator of an object control over it. This is why a first-time join needs so little explicitly delegated.

Why the baseline is this small. Because the join account creates the computer object, Active Directory automatically gives it control of that object. Setting the account password, recording the DNS host name, and registering the initial service principal names are all covered by that automatic control. As soon as the computer object is created by someone or something else, that automatic control no longer applies, and permissions must be delegated explicitly. That is the single idea behind most of the rows in the next table.

Alternative without delegation. Active Directory's default "Add workstations to the domain" user right also allows a join, but the computer object is placed in the default `Computers` container and the number of objects an account may create is capped by the domain's machine account quota. Sites that need the object in a specific organizational unit, or that will re-join over time, should use the delegation above instead.

7.7.3. Additional Permissions by Capability

Each row is additive on top of the baseline. Grant only the rows matching the capabilities the site will use.

Capability	Additional Permissions Required	Held By	Scope
Re-join when the computer account already exists **and** Hammerspace created it with the same join account	None	Join account	Existing computer object
Re-join when the computer account already exists and was **created by a different account**, or pre-created by the AD team	Reset Password; Validated write to DNS host name; Validated write to service principal name; Write Account Restrictions	Join account	Descendant computer objects in the organizational unit

© 2026 - Hammerspace

Capability	Additional Permissions Required	Held By	Scope
Leave the domain **and delete** the computer account	Delete Computer objects	Join account	Computer object, or the organizational unit containing it
Leave the domain **and keep** the computer account	None in the normal case. Modify Permissions is needed only if credentials are supplied and the computer account is missing the SPN permission described below.	Join account	Computer object
Register and update **DNS records**	Zone configured for secure dynamic updates; permission to create records in the zone (granted to authenticated computers by default); write access to any record of the same name that already exists	**Computer account**	The Active Directory–integrated DNS zone
Maintain **service principal names that match the computer account name**	None	**Computer account**	Its own computer object

7.7.4. What Each Capability Means in Practice

Re-joining an existing computer account. Active Directory will not let one account change another account's password without knowing the current password. Hammerspace does not know the existing computer account's password, so it must be able to force a password reset. This is the Reset Password permission. It applies whenever the computer object was not created by the join account being used — most commonly when the AD team pre-creates the object in a controlled organizational unit, or when a different administrator performed the original join. The accompanying validated-write permissions let Hammerspace update the object's DNS host name and service principal names, which the join account would otherwise have received automatically as the object's creator.

Because a site usually cannot predict whether a re-join will one day be needed, granting this row up front is recommended. Without it, the join fails with an access-denied error, and the AD team must delete or reset the existing object before the join can succeed.

Leaving the domain. Leaving can either delete the computer account or leave it in place for a later re-join. Deleting it requires the delete permission. Keeping it requires nothing extra in the normal case. The one exception is a leave where administrator credentials are supplied and the computer account is missing the service principal name permission; Hammerspace then tries to restore that permission first so it can clean up all the service principal names.

DNS registration. After the join, Hammerspace registers DNS records for the SMB server name, the Anvil DNS name, and the configured or floating IP addresses. These registrations are made by the **computer account**, not by the administrator account used during the join, so the permissions apply to the computer account and to the DNS zone rather than to the organizational unit. In a zone using secure dynamic updates, an authenticated computer may create a new record and becomes its owner. If a record with the same name already exists and

© 2026 - Hammerspace

is owned by a different account, the update is refused even when every organizational unit permission is correct. Pre-created DNS records must therefore grant the Hammerspace computer account write access. This is one of the most commonly missed requirements.

Hammerspace recommends registering floating IPs, rather than static IPs, in these DNS records, so that a DSX node failure does not leave the name unreachable. An option in the Active Directory configuration keeps these DNS records in sync automatically as floating IPs are added or removed; when it is not selected, the DNS administrator must update the records manually after any such change.

Service principal names. NFS with Kerberos requires service principal names in Active Directory. Those that match the cluster's computer account name are written by the computer account using a permission Active Directory grants to every computer for its own account, so they need no configuration.

7.7.5. Recommended Delegation for Most Sites

Granting the baseline plus the pre-created and delete rows covers every join, re-join, and leave scenario without granting broad directory rights. On the organizational unit that will hold the Hammerspace computer object:

1. Create Computer objects
2. Delete Computer objects
3. On descendant computer objects: Reset Password, Validated write to DNS host name, Validated write to service principal name, Write Account Restrictions
4. On descendant computer objects: Modify Permissions — only if the site wants Hammerspace to be able to repair the service principal name permission later
5. In the Active Directory–integrated DNS zone: dynamic update rights for the SMB server name, the Anvil DNS name, and the cluster's IP addresses



"Modify Permissions" appears in the Windows UI as **Modify permissions** in the advanced permission list for an object.

7.7.6. Troubleshooting

Symptom	Likely Cause
Join fails with access denied, and the computer object already exists in Active Directory	Reset Password not delegated on the existing object
Join fails with access denied, and no computer object exists	Create Computer objects not delegated on the target organizational unit, or the machine account quota is exhausted
Join succeeds, but leaving with the option to delete the computer account reports an error	Delete Computer objects not delegated

© 2026 - Hammerspace

Symptom	Likely Cause
Join succeeds, but a DNS name does not resolve to the cluster	A record with that name already exists in the zone and is owned by another account, or the zone does not accept secure dynamic updates
NFS Kerberos mounts fail, and the system reports that the SPN permission is required	The computer account lacks permission to write service principal names that do not match its own name

© 2026 - Hammerspace

Chapter 8. Data Management

The Hammerspace product enables the protection and availability of share data using objectives. Objectives are the policies that enable granular control over data placement, retention, versioning, and many other parts of the data lifecycle. Objectives can be applied to shares, directories, or files.

- **Data availability and protection objectives:** These objectives allow admins to specify the desired levels of data availability and durability. Each storage volume is assigned an availability and durability value. The system will place data on volumes that meet those values or create multiple file instances to meet the requested number of nines.
- **File versioning, undelete, and site versioning objectives:** These objectives provide more granular control over data recoverability. They use the same methods as Product snapshots but are applied as an objective rather than a share option.
- **Anti-virus scanning objectives:** The Product can scan files for viruses, preventing infected files from being opened.
- **Access-based enumeration:** This objective hides files and folders that users do not have permission to access.

The Product supports a few kinds of objectives, which enable increasingly granular orchestration of data:

- **Default objectives** are automatically applied to shares when new shares are created.
- **Predefined objectives** are preconfigured in the Product and can quickly and easily be applied to your shares, directories, and files using the GUI.
- **Conditional objectives** can be constructed using the GUI and combine the simplicity of the predefined objectives with the ability to selectively implement the objective.
- **Advanced objectives** enable granularity and precision using advanced expressions.

Objectives can be applied using the GUI, CLI, and *Hammerspace toolkit* (**hstk**), which is available on GitHub. This section will cover applying objectives using the GUI and CLI. For more information about **hstk**, see the GitHub page: https://github.com/hammer-space/hstk.

Objectives can be applied at the share, directory, and file levels. The objective is evaluated the same way, regardless of which tool you use to apply it.

This section includes the following subsections:

- Anti-Virus Scanning
- Cross-Site Versioning
- Data Availability and Durability
- Data Profiler
- File Clones and File Snapshots
- File Versioning
- Managing Labels

© 2026 - Hammerspace

- Managing Shares
- Referral Shares
- Snapshots
- Using Objectives

8.1. At the Share Level

When objectives are applied at the share level, i.e., they are applied during a share-create or share-edit operation, every file and directory in the share automatically inherits the objective(s) of the share.

This can be especially important for objectives that deliver solutions for WORM, Anti-virus scanning, Access-based enumeration, versioning, and undelete. When objectives are applied at the share-level, with the applicability of **Always**, it ensures that the objectives are always evaluated when appropriate.

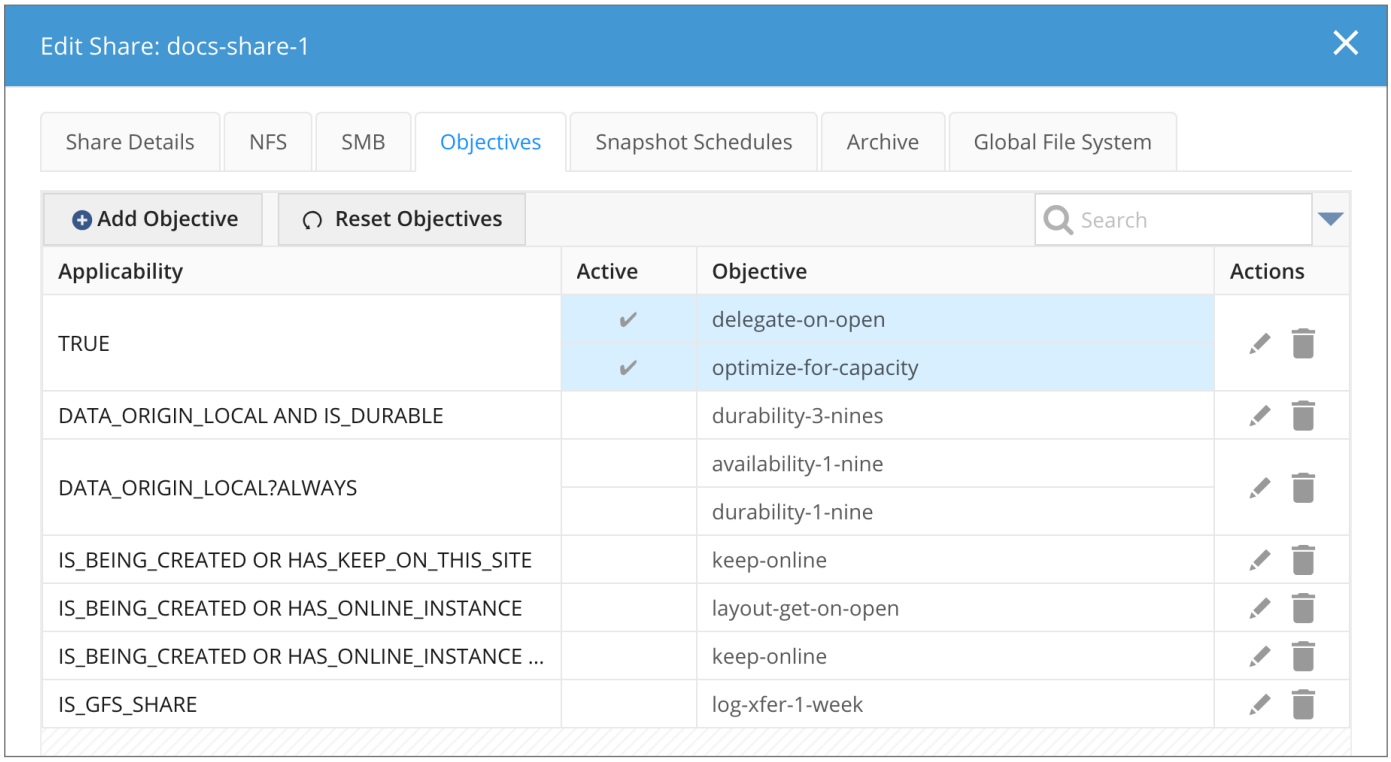


Figure 54. Editing share-level objectives

8.2. At the Directory Level

When compared to share-level objectives, directory-level objectives can appear very similar; however, there is a major distinction: a directory-level objective can be removed farther down in the directory tree. This would make it potentially unsuitable for operations such as WORM or virus scanning.

Directory objectives offer the most flexible data management ability in the system. They can be applied to any directory and, by default, all the data below that directory will inherit the directory objective. Since they can be applied using the Hammerspace toolkit, `hstk`, end-users can self-manage their data and directly impact where data is placed.

© 2026 - Hammerspace

Directory objectives can also be applied at the root of the share (on `/`), assuming the user has the appropriate permission to modify the root of the share. It is not recommended to apply objectives at the root of a share (for manageability reasons). Instead, use share-level objectives to achieve exactly the same results.

You can edit objectives for a directory by navigating to a share **Data** › **Shares** > *<share-name>*, selecting the **Files** tab, then clicking the edit icon for the directory in the **Applied Objectives** column.

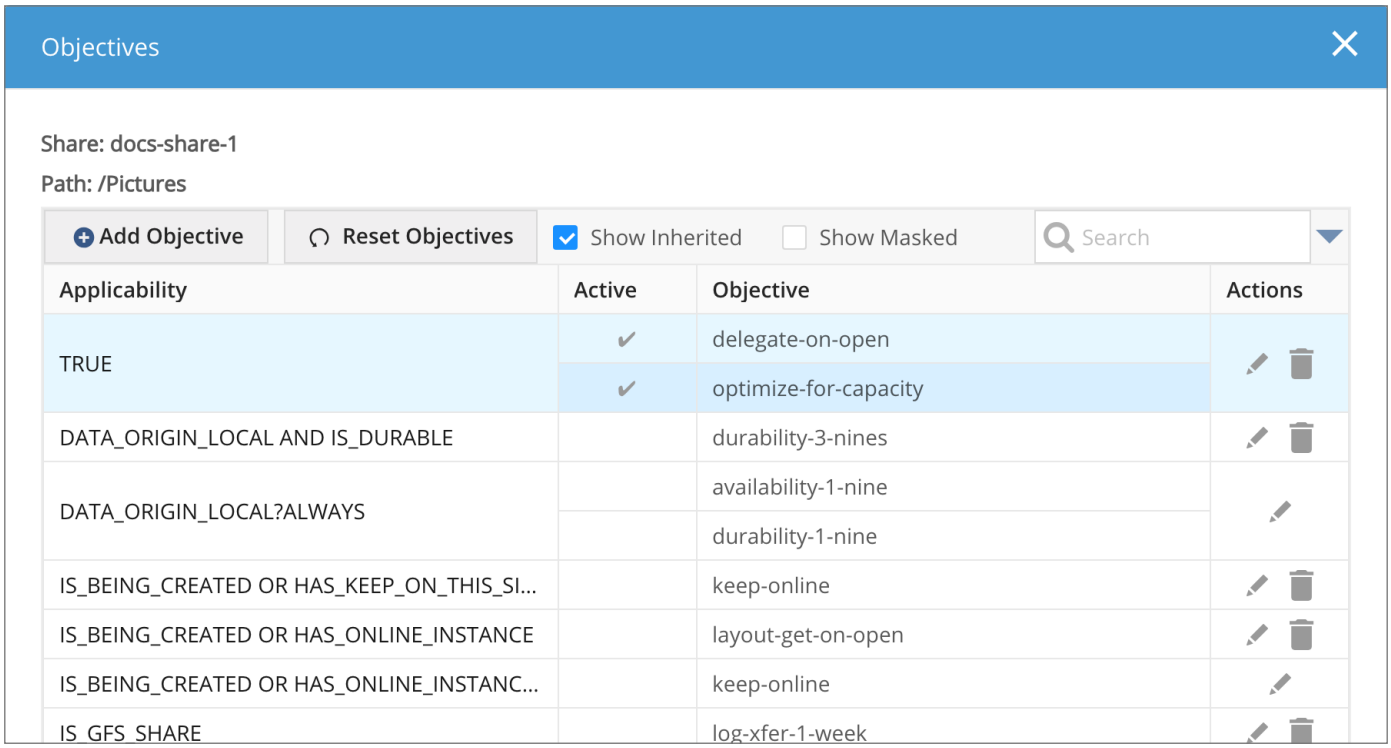


Figure 55. Editing directory-level objectives

8.3. Anti-virus Scanning

Scanning files for anti-viruses is supported for on-access and background scanning. ICAP protocol support on the anti-virus servers is required for the anti-virus scanning functionality.

The Product will scan files on access and prevent files from being opened if a virus is detected.

8.3.1. Requirements

- One or more ICAP-enabled virus scanning servers is configured
- Virus scanning objectives are configured on the share

8.3.2. How It Works

Virus scanning is enabled using objectives.

When a file is written to the share, it is considered an UNSCANNED file. The system will allow for the file to be written and closed without forcing a scan at the time of writing; however, files cannot be opened again without first being scanned.

© 2026 - Hammerspace

The scanning occurs when the file is opened or in the background, as part of the background sweep of the share. The results of a scan operation are either NON-THREAT or THREAT. If the file is considered a THREAT, it cannot be opened by a client across any protocol. Files that are considered a threat can be deleted by a user with delete permissions/ACLs to that file.

Virus scanning can be enabled on a share with existing data. When virus scanning is enabled on a share after data has been stored in the share, it may take several minutes before the virus scan objective is effective on individual files.



Hammerspace recommends that virus scanning functionality be set at the root of the share; however, it can be applied on top of any directory in a share.

8.3.3. Limitations

The following limitations apply to virus scanning:

- There are no scheduled virus scans. Files are scanned when opened and/or automatically in the background *based on the objective applied*.
- Automated repair of files is not supported.
- Files will not open when a virus is detected; however, there is no quarantine location where files are moved if a virus is detected. *The file is effectively quarantined in place.*
- There is no GUI support for virus scan-specific reporting. This only works in the administrator command line.
- If all virus scanning servers are down, the system currently will allow for data access, flagging the file as non-compliant and will scan the file once the servers are back up again. This behavior is currently not configurable.

8.3.4. Virus Definition Updates

Virus definition version details are provided via the ICAP protocol. The system will store metadata related to which version of the virus definitions was used to last scan the file. When a file is opened, it is automatically determined if the file needs to be scanned again based on the version of the virus definitions.

8.3.5. Performance

When a file is scanned, it is sent to one of the configured ICAP servers. This may incur a performance impact, which depends on how long the scan takes. The scan can take longer for larger files or if the ICAP servers are too busy to handle the request immediately.



Hammerspace recommends providing at least **two** ICAP servers to ensure a resilient architecture. If multiple ICAP servers are configured, scanning requests will automatically be distributed across available and healthy ICAP servers. For large and highly active environments, we recommend configuring as many ICAP servers as DSX nodes.

© 2026 - Hammerspace

8.3.6. Adding an ICAP Server

You can add an ICAP server(s) using the `antivirus-add` command in the Admin CLI.

Adding an ICAP server

Command:

```
antivirus-add --endpoint icap://icap1.example.com:1344/avscan --name av1 --type CLAM_AV
```

Expected output:

```
Name: av1
Type: Clam AntiVirus
Internal ID: 1073741900
ID: b8d6ac67-6fd0-486e-ab74-2c8abda4cb16
Endpoint: icap://icap1.example.com:1344/avscan
Oper state: Up
Admin state: Up
```

Both secure (port: 1345) and regular (port: 1344) ports are supported. McAfee Anti-Virus and ClamAV have been tested with the Product.

If adding multiple servers, simply run the command for each server.

The ICAP endpoints are automatically added to the virus scanning volume group and if objectives are applied, will automatically be used as they come online.

8.3.7. Identifying Infected Files

1. Start by mounting the share on an NFS client that has the Hammerspace toolkit (hstk) installed. It can be mounted over NFS v3 or v4.2. The total number of files on this share is 4,237 files.

   Mounting NFS client with hstk installed

   Command:

   ```
   # cd _<mount point>_
   # find . -type f -print | wc -l
   ```

   Expected output:

   ```
   4237
   ```

2. Change directory in `.collections/threat` and find out how many files are considered a threat.

   Viewing number of threat files

© 2026 - Hammerspace

Command:

```
# cd .collections/threat

# find . -type f -print | wc -l
```

Expected outcome:

```
7
```

3. Now you can simply identify the files with the `find` command, a recursive `ls`, or your preferred method. This example uses the `--exec` option to find files.



It may be helpful to take a share snapshot before this step to store the threat files for further investigation.



DO NOT use `rm -rf` in this directory, because that would also delete directories that are likely to store non-threat files.

Identifying the threat files using --exec option

```
# find . -type f -print -exec rm {} \;

./dir1/2/eicar.com

./dir1/3/eicar.com

./dir1/eicarcom2.zip

./dir1/eicar_com.zip

./dir1/eicar.com.txt

./dir1/eicar.com

./dir1/virus2
```

8.3.8. Locating Infected Files by Querying Data

Table 15. Locating infected files by querying the data

```
cd <mount-point/path>

hs eval -e 'IS_FILE and attributes.virus_scan==virus_scan_state("THREAT")?path' . --recursive

"./dir1/3/eicar.com"

"./dir1/2/eicar.com"

"./dir1/eicarcom2.zip"

"./dir1/eicar_com.zip"

"./dir1/virus2"

"./dir1/eicar.com.txt"

"./dir1/eicar.com"
```

8.3.9. Remediation

Hammerspace currently does not support auto-remediation. The deletion of infected files is the responsibility of the administrator or user. Several threat reports can be generated (see Reporting section) to show whether infected files exist in a share.

There is also a new threat collection tool that sequesters files considered threats. In every directory in the share, the user can navigate to `.collections/threat`. This collection will only show infected files. This collection makes it easy to identify and safely delete only infected files.



Although the threat collection directory is available in the GUI, files must be deleted using the CLI from an NFS client.

In this example, we identify infected files and delete them. This example assumes that the user deleting the files has file permissions to do so.

8.3.10. Reporting

The GUI reports how many files and how much data are being sent to the anti-virus servers. This information is part of the mobility graph. The administrator can differentiate between on-access mobility and background scanning activity.

Navigate to the Mobility graph and from the Filters (top left), **Volume Group** dropdown, select `virus-scanners`. The data can also be filtered further by selecting a share.

In the example below, you can see that most of the anti-virus scanning is performed as a background activity and that only 12 files have been scanned on-demand.

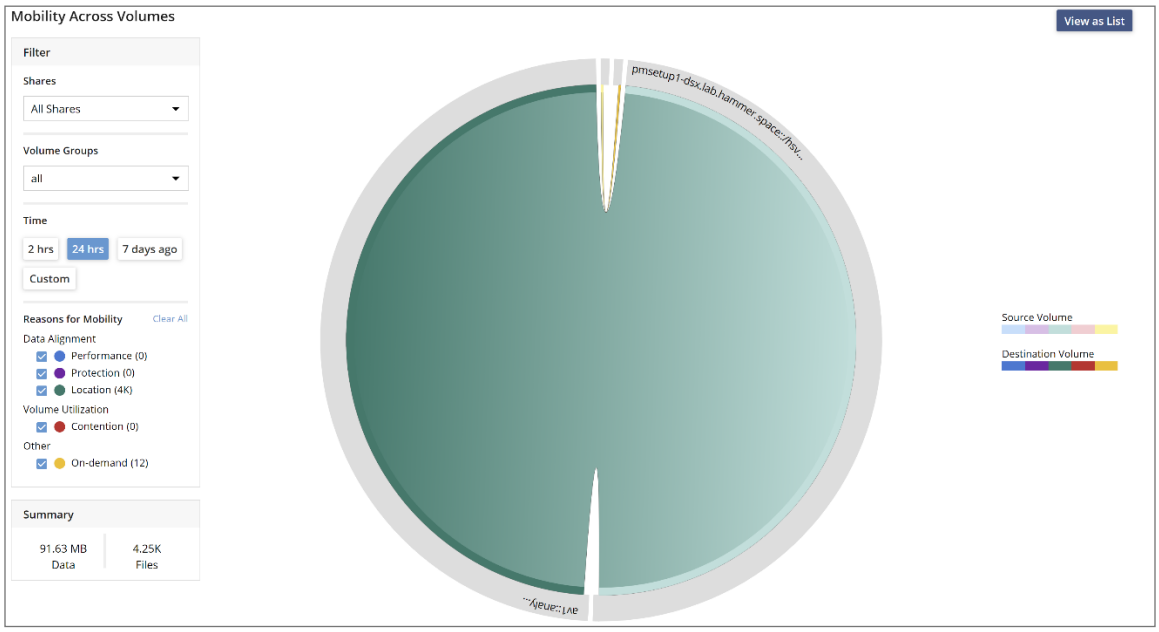


Figure 56. Mobility to anti-virus servers

Reporting of detailed virus scan operations is available from the CLI, using the **hstk** (https://github.com/hammer-space/hstk), and it requires an NFS mount. When the toolkit is installed and the share is mounted on the client, more granular reports can be generated.

The system maintains in-memory statistics and this information can be gathered at any time and is available instantly.

8.3.11. Viewing Only Top Files

Using hstk to generate virus scanning report of top files

Command:

```
hs collsum . threat --collation top-files
```

Expected outcome

```
|KEY = {4.096 KBYTES, "./dir1/virus2"};
|KEY = {4.096 KBYTES, "./dir1/eicar_com.zip"};
|KEY = {4.096 KBYTES, "./dir1/eicarcom2.zip"};
|KEY = {4.096 KBYTES, "./dir1/eicar.com.txt"};
|KEY = {4.096 KBYTES, "./dir1/eicar.com"};
|KEY = {4.096 KBYTES, "./dir1/3/eicar.com"};
|KEY = {4.096 KBYTES, "./dir1/2/eicar.com"}}}
```

8.3.12. Viewing Summary of Virus-Scanning Operations on a Share

Using hstk to summarize all virus scanning operations for a share since the system was started

Command:

```
hs collsum _<PATH TO MOUNTED SHARE>_ all --collation by-virus-scan
```

Expected outcome:

```
INDEXED_TABLE++{++SUMMATION('BY-VIRUS-SCAN'), INDEXED_TABLE++{++
VIRUS_SCAN_STATE('UNSCANNED'), ++{++0 FILES, 0 BYTES, 0 OPERATIONS / SECOND};
VIRUS_SCAN_STATE('NON-THREAT'), ++{++4.23 KFILES, 101.39 MBYTES, 1 OPERATION / SECOND};
VIRUS_SCAN_STATE('THREAT'), ++{++5 FILES, 20.48 KBYTES, 0 OPERATIONS / SECOND}}}
```

8.3.13. Viewing Threat Status of an Individual File

Individual files can also be queried. This is calculated when the command is run.

HSTK: Check individual file status

```
hs eval -e "attributes.virus_scan" virus2

VIRUS_SCAN_STATE('THREAT')
```

8.3.14. Viewing Threat Status of a Share

The share-level report can be generated anywhere in the tree. It includes a top-1000 list of files that are considered a threat.

HSTK: Generate share-level report from anywhere in the directory tree

```
cd <mount>/apps/sample
hs collsum . threat
INDEXED_TABLE{
SUMMATION('BASIC'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND};
SUMMATION('BY-VOLUME'), INDEXED_TABLE{
STORAGE_VOLUME('dsx1.example.com::/hsvol0'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND};
STORAGE_VOLUME('av1::analyzer'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND}};
SUMMATION('BY-ALIGNMENT'), INDEXED_TABLE{
ALIGNMENT('SEVERELY MISALIGNED'), {0 FILES, 0 BYTES, 0 OPERATIONS / SECOND};
ALIGNMENT('ALIGNED'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND}};
SUMMATION('BY-ACTIVE-OBJECTIVE'), INDEXED_TABLE{
SLO('keep-online'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND};
SLO('optimize-for-capacity'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND};
SLO('delegate-on-open'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND};
SLO('layout-get-on-open'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND};
SLO('durability-1-nine'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND};
SLO('durability-3-nines'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND};
SLO('availability-1-nine'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND};
SLO('compress-on-object'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND};
SLO('content-based-chunk-on-object'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND};
SLO('sync-metadata'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND};
```


© 2026 - Hammerspace

```
SLO('deny-open'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND};
SLO('place-on-Virus-Scanners'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND}};
SUMMATION('BY-HEAT'), INDEXED_TABLE{
TEMPERATURE_LEVEL('1 TO 5 MINUTES OLD'), {1 FILE, 4.096 KBYTES, 0 OPERATIONS / SECOND};
TEMPERATURE_LEVEL('UNDER 1 MINUTE OLD'), {6 FILES, 24.576 KBYTES, 0 OPERATIONS / SECOND}};
SUMMATION('BY-ACCESS-AGE'), INDEXED_TABLE{
TIMESPAN_LEVEL('UNDER 1 MINUTE OLD'), {5 FILES, 20.48 KBYTES, 0 OPERATIONS / SECOND};
TIMESPAN_LEVEL('1 TO 5 MINUTES OLD'), {2 FILES, 8.192 KBYTES, 0 OPERATIONS / SECOND}};
SUMMATION('BY-MODIFY-AGE'), INDEXED_TABLE{
TIMESPAN_LEVEL('UNDER 1 MINUTE OLD'), {1 FILE, 4.096 KBYTES, 0 OPERATIONS / SECOND};
TIMESPAN_LEVEL('6 MONTHS TO 1 YEAR OLD'), {6 FILES, 24.576 KBYTES, 0 OPERATIONS / SECOND}};
SUMMATION('BY-CHANGE-AGE'), INDEXED_TABLE{
TIMESPAN_LEVEL('UNDER 1 MINUTE OLD'), {6 FILES, 24.576 KBYTES, 0 OPERATIONS / SECOND};
TIMESPAN_LEVEL('1 TO 5 MINUTES OLD'), {1 FILE, 4.096 KBYTES, 0 OPERATIONS / SECOND}};
SUMMATION('BY-CREATE-AGE'), INDEXED_TABLE{
TIMESPAN_LEVEL('UNDER 1 MINUTE OLD'), {6 FILES, 24.576 KBYTES, 0 OPERATIONS / SECOND};
TIMESPAN_LEVEL('1 TO 5 MINUTES OLD'), {1 FILE, 4.096 KBYTES, 0 OPERATIONS / SECOND}};
SUMMATION('BY-SPACE-USED'), INDEXED_TABLE{SIZE_LEVEL('4 TO 32 KBYTES'), {7 FILES, 28.672 KBYTES, 0
OPERATIONS / SECOND}};
SUMMATION('BY-ERRORS'), INDEXED_TABLE{0, {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND}};
SUMMATION('BY-TYPE'), INDEXED_TABLE{ITEM_TYPE('FILE'), {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND}};
SUMMATION('BY-VERSION'), INDEXED_TABLE{1, {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND}};
SUMMATION('BY-MIME'), INDEXED_TABLE{"/", {7 FILES, 28.672 KBYTES, 0 OPERATIONS / SECOND}};
SUMMATION('BY-VIRUS-SCAN'), INDEXED_TABLE{VIRUS_SCAN_STATE('THREAT'), {7 FILES, 28.672 KBYTES, 0
OPERATIONS / SECOND}};
SUMMATION('TOP-FILES'), TOP1000_TABLE{
|KEY = {4.096 KBYTES, "./dir1/virus2"};
|KEY = {4.096 KBYTES, "./dir1/eicar_com.zip"};
|KEY = {4.096 KBYTES, "./dir1/eicarcom2.zip"};
|KEY = {4.096 KBYTES, "./dir1/eicar.com.txt"};
|KEY = {4.096 KBYTES, "./dir1/eicar.com"};
|KEY = {4.096 KBYTES, "./dir1/3/eicar.com"};
|KEY = {4.096 KBYTES, "./dir1/2/eicar.com"}}}
```

Note that the entire output can be quite long. It can be reduced by adding `--collation <summation category>`.

The summation categories are

- BASIC
- BY-VOLUME
- BY-ALIGNMENT
- BY-ACTIVE-OBJECTIVE
- BY-HEAT
- BY-ACCESS-AGE
- BY-MODIFY-AGE
- BY-CHANGE-AGE

© 2026 - Hammerspace

- BY-CREATE-AGE
- BY-SPACE-USED
- BY-ERRORS
- BY-TYPE
- BY-VERSION
- BY-MIME
- BY-VIRUS-SCAN
- TOP-FILES

8.3.15. Virus-Scanning Objective

Here is an example objective, added as an advanced objective that will enable virus scanning. Note that this objective is only in effect if

- a virus-scanner server is set up and available.
- the file has not yet been scanned.
- the file is larger than 2 MB.

Example of an advanced virus scanning objective

```
IF VOLUME_GROUPS[|NAME="Virus-Scanners"].NUMBER_UP AND (IS_BEING_CREATED OR
ATTRIBUTES.VIRUS_SCAN==VIRUS_SCAN_STATE("unscanned") AND SIZE<2*MBYTES AND IS_ONLINE) THEN
{SLO('place-on-Virus-Scanners'),SLO('block-open')} ELSE IF
ATTRIBUTES.VIRUS_SCAN==VIRUS_SCAN_STATE("threat") THEN {SLO('place-on-Virus-Scanners'),SLO('deny-
open')} ELSE {SLO('place-on-Virus-Scanners')}
```

8.4. Cross-Site Versioning

Cross-site versioning is a feature of Hammerspace Global File System (GFS) that allows you to keep a record of file changes when a file is modified in one site and then saved in a different site. The last site to modify the file is considered the owner of the file.

The Global File System is a core capability of Hammerspace platform that allows users to access files from different locations as if they were all stored in the same place.

Cross-site versioning is configured using the `log-xfer` objective. When you create a log-xfer objective, you can specify how long you want to keep a version of a file after it is saved in a different site. The options are 1 hour, 1 day, 1 week, or 1 month.

8.4.1. Configuring Cross-Site Versioning Using the Admin CLI

Enter the following command to set the log-xfer objective for a share using the Admin CLI.

© 2026 - Hammerspace

```
share-objective-add --name <share name> --objective log-xfer-1-<duration>
```

For example, to keep a version of a cross-versioned file for one week after it is saved in a different site, you would use the following command:

```
share-objective-add --name <share name> --objective log-xfer-1-week
```

8.4.2. Configuring Cross-Site Versioning Using the GUI

Use the following procedure to configure the cross-file versioning for a share using the GUI.

1. Enter the cluster IP address in your browser to open the GUI.
2. Select **Data** from the left-side panel and see the **Shares** tab.
3. Locate the share you want to configure and click the pencil icon for that share in the **Applied Objectives** column.
4. In the popup, click **Add Objective** and in the **Filter**, type "*log-xfer*" to see the options available for cross-file versioning.
5. From the available objectives, select a duration for how long you want to keep the cross-site versioned file and click **Apply**. The options are for an hour, day, week, or month.
6. In the **Objectives** tab for the share, you can see the new log-xfer objective and verify that its applicability is TRUE, and it is Active. Click **Close** to finish.

8.5. Data Availability and Durability

This section describes objectives related to data availability and protection.

The system treats availability and durability in a very similar fashion, but there are some subtle differences. One way to think of it is protection against temporary vs permanent loss of data access.

Availability represents temporary loss of (access to) data, while durability represents permanent loss of data. As such, durability should always be greater than availability, since data that is permanently lost is, by definition, inaccessible.

One subtle way the difference comes into play is that if you want to temporarily take a storage node offline, you can lower its configured availability to match the time you expect to keep it offline. A short service window may mean you don't lower the availability at all. A longer one means you lower it, possibly even all the way to zero. By lowering the availability, data that is resident there, and with availability objectives not otherwise met, will need to be moved (or multiply instantiated) onto other volumes.

In other words, to service storage, you lower availability. To decommission storage, the system lowers durability to zero.

This is why you need two measures to express the temporary vs. permanent nature of the loss.

© 2026 - Hammerspace

8.5.1. The Availability Number-of-Nines Objective

Three availability objectives enable the admin to specify the number of nines of availability the data should have. Each storage volume is assigned an availability value when it is added. Data will be placed either on a storage volume that meets the availability number, or multiple file instances may be created to meet the requested number of nines.



If you configure a share with an availability objective that causes multiple file instances to be created, and the volume containing some or all those files becomes marked as SUSPECTED (unavailable) within the cluster, the cluster will re-silver the impacted files after 30 minutes by default. If the volume SUSPECTED status later clears, the cluster will automatically remove any excess file instances.

The default availability objectives allow you to specify either 1, 3, or 5 nines. Custom objectives can be created that allow for up to 11 nines in the GUI and up to 20 nines in the Admin CLI.

Best Practices

- Use availability objectives to control how many file instances are created, so that if a storage volume becomes unavailable, a file can still be read from another volume. The availability objective will always try to place the file instances on different failure domains (meaning different hosts) to ensure this happens.
- Double-check your volume availability and durability values! While Hammerspace does apply defaults, not all volumes are created equal. For example, a self-hosted object storage volume on a standalone S3 server is likely less durable and available than one managed by AWS, Google, or Microsoft Azure; or a DSX volume backed by a RAID is better than one that is not. To edit the settings, go to the **Infrastructure** page and click the pencil icon to the right of the volume name. In the Volume Properties **Details** tab, scroll down to the **Capabilities** section and edit the **Availability** and **Durability** values as needed.
- The volume durability value should always be larger than the availability value.

8.5.2. Volume Availability - Default Values

Volumes are assigned the following default availability values that can be changed if required:

- All DSX or remote NFS (online) volumes: 2 nines
- All Object storage (offline) volumes: 4 nines



Based on these default values, the share default objectives will only create one copy of every file.

8.5.3. Mirroring Data Between NFS Volumes

You can mirror data across NFS volumes using the `availability-3-nines` objective. This objective creates two copies of your data on different storage volumes. This will protect your data if one of the storage volumes fails.

Complete the following steps to apply the availability-3-nines objective to a share in your environment.

1. Enter the cluster IP in your browser to open the Management GUI.

© 2026 - Hammerspace

2. Select **Data** from the left-side panel and see the **Shares** tab.
3. Locate the share you want to configure and click the pencil icon for that share in the **Applied Objectives** column.
4. You can apply the availability-3-nines objective to the share or directories or files within the share.
5. *Optional:* Navigate within the share to the directory or file that you want to mirror, and click the pencil icon in the **Applied Objectives** column for that object.
6. In the popup, click **Add Objective** and in the Filter, type “availability” and press `Enter`.
7. Click the box for availability-3 nines and click **Apply**.
8. In the Objectives tab for the share or path, you can see the new `availability-3-nines` objective and verify that its applicability is TRUE, and it is Active. Click **Close** to finish.

8.5.4. Using the Durability Number-of-Nines Objective

Three durability objectives enable the admin to specify how many nines of durability the data should have. Each storage volume is assigned a durability value. Data will be placed either on a storage volume that meets the durability number (minimum is 1) or multiple file instances may be created to meet the requested number of nines.



If you configure a share with a durability objective that causes multiple file instances to be created, the cluster will only resilver the impacted files if the volume is decommissioned or failed. Decommissioning a volume gracefully moves the file instances it contains to other volumes, while failing a volume marks all file instances that it contains as permanently lost. Volumes are typically only failed if their underlying storage has failed.

The default durability objectives allow you to specify one, three, or five nines. Custom objectives can be created that allow for up to 11 nines in the GUI and up to 20 nines in the Admin CLI.



Durability and availability are calculated separately. Based on the cluster volume configuration, each may require a different number of file instances to be created.

8.5.5. Volume Durability – Default Values

Volumes are assigned the following default durability values that can be changed if required:

- All DSX or remote NFS (online) volumes: three nines
- All Object storage (offline) volumes: nine nines



Based on these default values, the share default objectives will only create one instance of every file.

8.6. Data Profiler

The data profiler analyses data in a share and gives the user the ability to build a tiering model to determine if savings can be achieved by moving data to different tiers or even into the cloud. The data profiler gives the user a before and after view of costs to help determine the business value of tiering. The following data points can

© 2026 - Hammerspace

be seen in the data profiler:

- Total cost of the analyzed storage environment
- Cost savings vs. previous or reference configuration
- Cost per storage tier
- Amount of capacity per tier
- Number of files per tier

The profiler can run on data managed by Hammerspace, and it does not affect the data or client data access. Data can also be imported from other NAS storage systems by leveraging Read-Only volume assimilation to build a virtual share. See the Assimilation Overview to learn more.



Product documentation for Hammerspace 5.1 is available via PDF download on the Hammerspace Documentation Version 5.1.x page of the *Hammerspace License and Delivery Portal*.

Complete the following steps to run the Data Profiler:

1. Open the Management GUI and select **Objectives** from the left-side navigation menu, and then click the **Data Profiler** tab.
2. Select a share from the **Select Share** pull-down list.
3. From the **Data** pull-down list, select whether you want to analyze the data using Access Time (atime), Change Time (ctime), or Modify Time (mtime).
4. Click the **Refresh** button to populate the graph. For large data sets, this operation may take some time.



The moment the Refresh button is clicked, the job is submitted and will be executed as a background task. Navigating away from the page will not cancel or affect the background task.

5. Once the analysis is complete, enter the **$/GB** (US-dollar increment per Gigabyte) for the current storage in the **Cost** field at the bottom of the page, and then click **Set as Baseline at the top of the page**. This will generate a basis for the current cost of the data.
6. Click on the slider bar to create one or more objectives, set the **$ / GB** for each objective and slide from left to right to change the level of inactivity. The number of files, the capacity used, and the cost are dynamically updated.

The baseline and the cost(s) can be updated several times to help analyze various scenarios.

8.6.1. Data Mobility

Hammerspace non-disruptively moves data between storage volumes to maintain alignment with objectives. Data may be moved to maintain performance and/or protection requirements, to balance load and avoid contention; or to follow a simple placement rule defined by an administrator.

The Alignment column in the Management GUI dashboard displays current alignment status and the reason for current data mobility events.

© 2026 - Hammerspace

The Product provides a historical view of data mobility in the form of either a chord diagram or a list that shows where data moved to and from, along with the reason for the data movement.

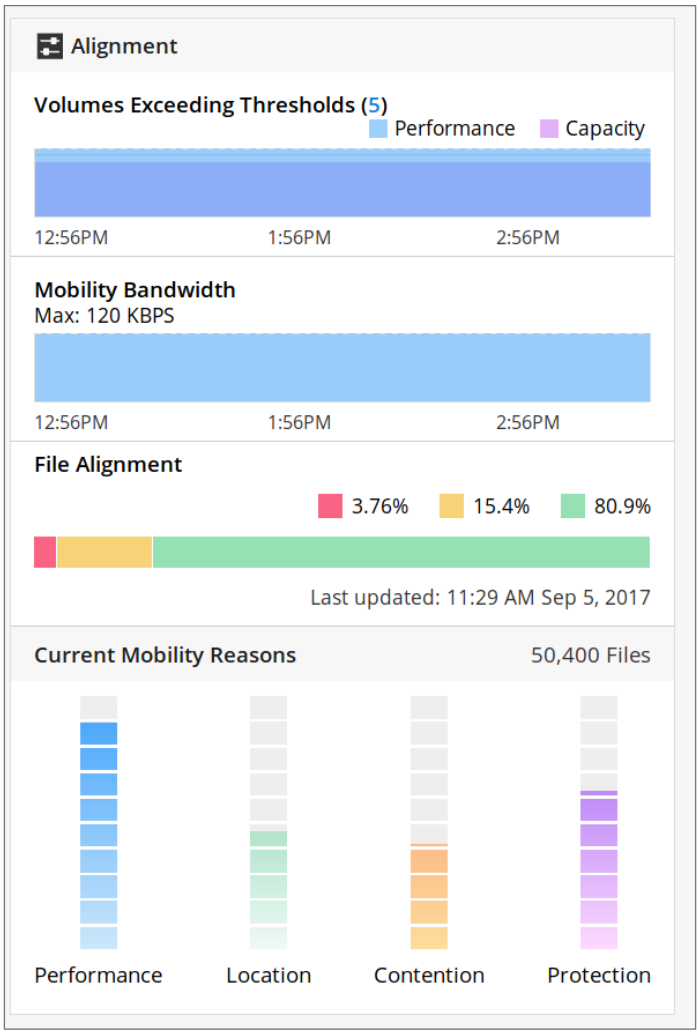


Figure 57. File alignment monitoring

8.6.2. Mobility History Chord Diagrams

To view mobility history as a chord diagram, select **Mobility** from the left side panel:

© 2026 - Hammerspace

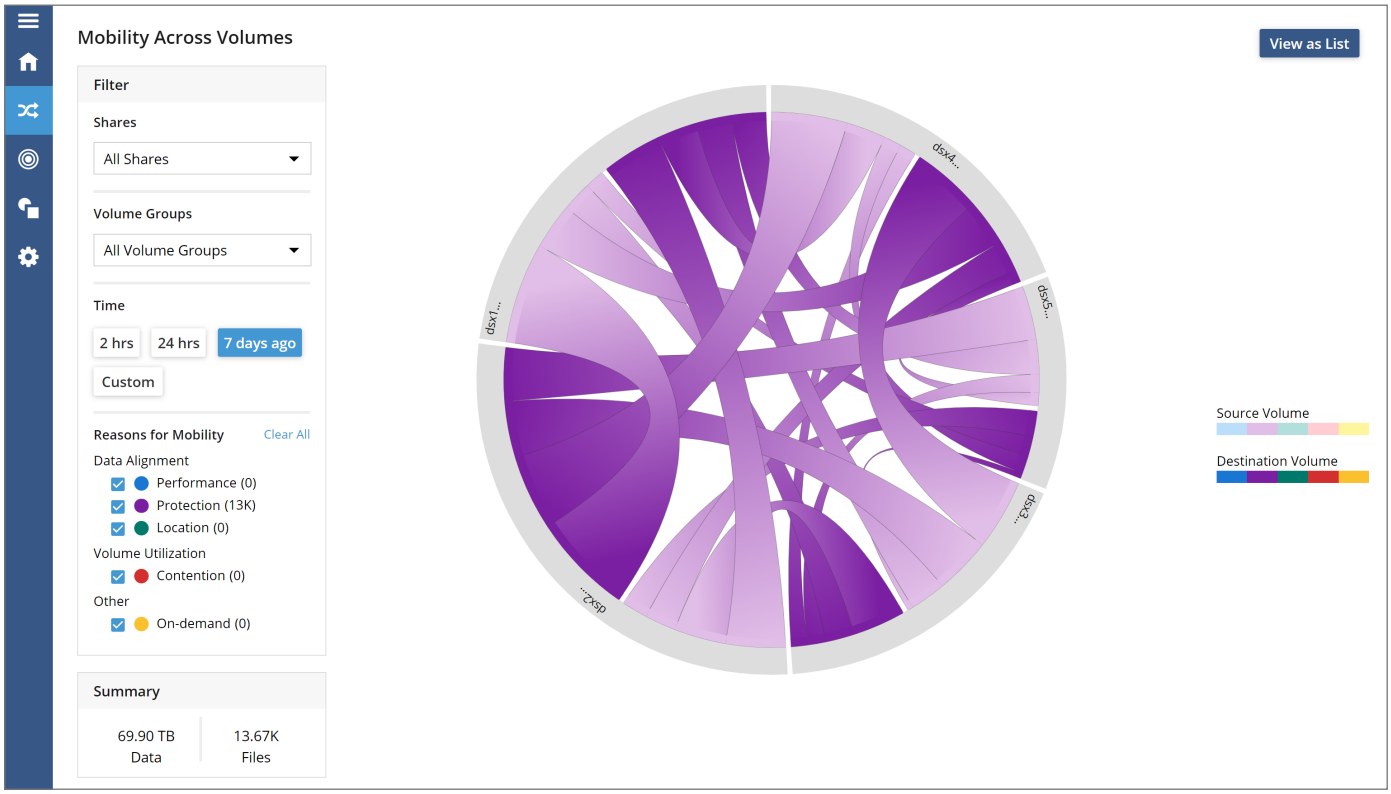


Figure 58. Mobility diagram example

The reasons for data movement are shown in color-coded ribbons. The width of each ribbon represents the relative amount of data moved, and the color is graded from light to dark to help visualize source and destination volumes.

The view can be filtered by share, volume group, mobility reason(s), and by time range.

© 2026 - Hammerspace

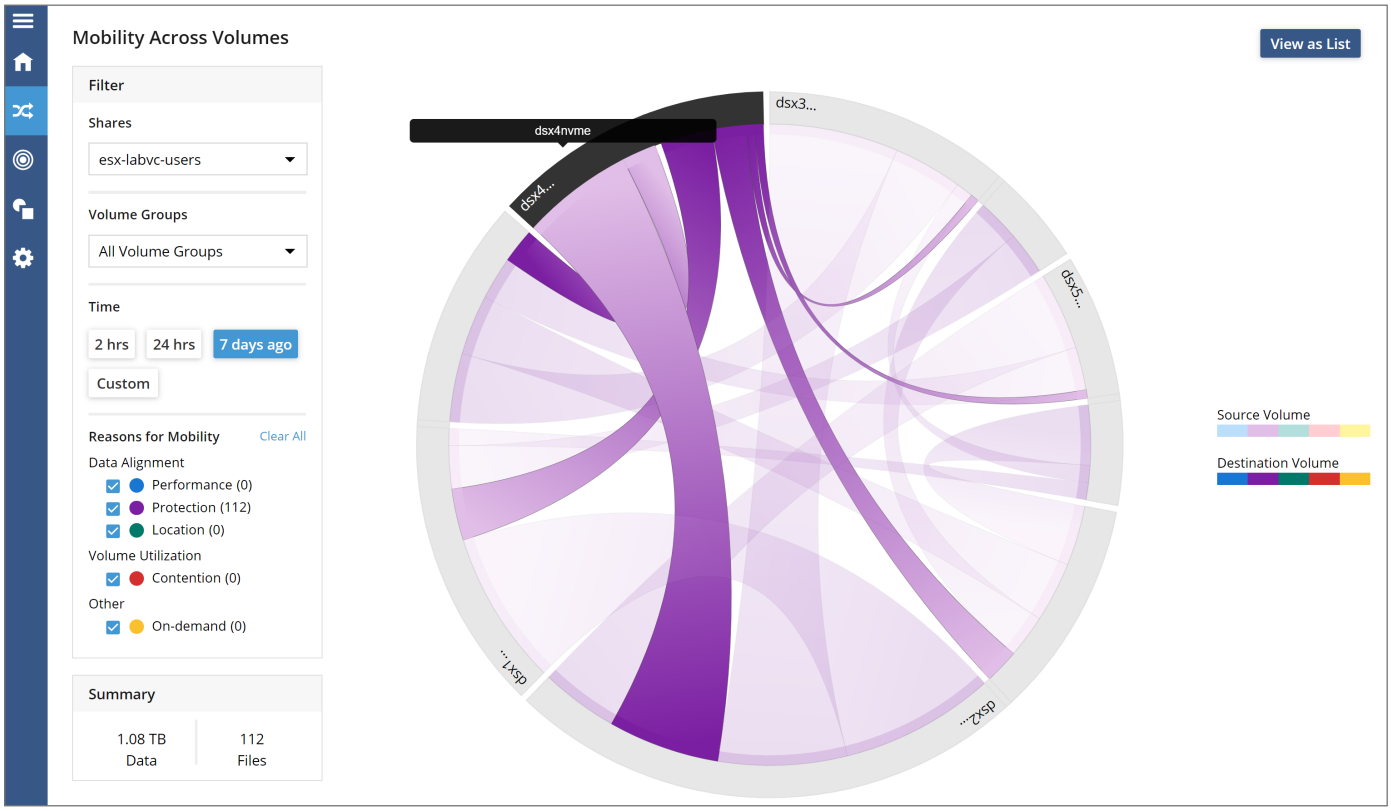


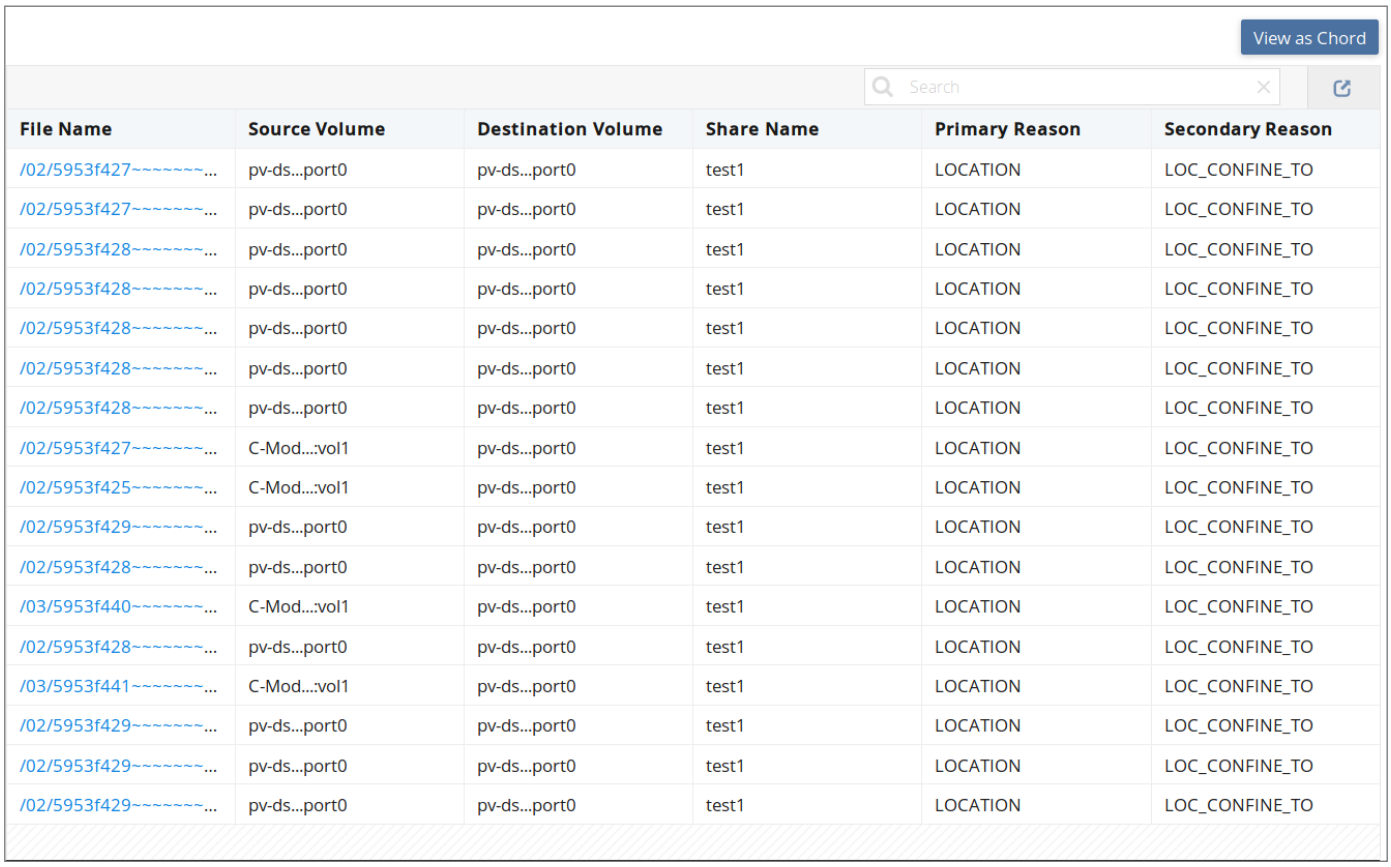
Figure 59. Hover over an entry to filter view to a single volume

Left-clicking a ribbon lists the files it represents.

8.6.3. Mobility History Lists

To view mobility history as a list, select **Mobility** from the left side panel and click **View as a list**:

© 2026 - Hammerspace



File Name	Source Volume	Destination Volume	Share Name	Primary Reason	Secondary Reason
/02/5953f427~~~~~~~...	pv-ds...port0	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f427~~~~~~~...	pv-ds...port0	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f428~~~~~~~...	pv-ds...port0	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f428~~~~~~~...	pv-ds...port0	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f428~~~~~~~...	pv-ds...port0	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f428~~~~~~~...	pv-ds...port0	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f428~~~~~~~...	pv-ds...port0	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f427~~~~~~~...	C-Mod...:vol1	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f425~~~~~~~...	C-Mod...:vol1	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f429~~~~~~~...	pv-ds...port0	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f428~~~~~~~...	pv-ds...port0	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/03/5953f440~~~~~~~...	C-Mod...:vol1	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f428~~~~~~~...	pv-ds...port0	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/03/5953f441~~~~~~~...	C-Mod...:vol1	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f429~~~~~~~...	pv-ds...port0	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f429~~~~~~~...	pv-ds...port0	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO
/02/5953f429~~~~~~~...	pv-ds...port0	pv-ds...port0	test1	LOCATION	LOC_CONFINE_TO

Figure 60. Viewing mobility history as a list

The list shows which files were moved, the reasons for the move, and where they were moved to and from.

The list can be exported to a CSV file by clicking on the arrow in the top-right corner of the table.

8.6.4. Alignment

Hammerspace enables administrators and data owners to visualize the data alignment of objectives, helping them determine whether the objectives set for data can be met with the current resources and configuration.

Alignment Reporting

For a share

The following procedure shows how to view alignment information for a share.

1. Select **Data** from the left side panel, and select the **Shares** tab.
2. Click on the desired share; this opens the dashboard for the share.
3. Select the **Alignment** tab. This page displays current and historical information about the alignment of all files and directories on the share.

For a file or directory

The following procedure shows how to see the current alignment status of specific directories or files.

1. Select **Data** from the left side panel, and select the **Shares** tab.

© 2026 - Hammerspace

2. Click on the desired share; this opens the dashboard for the share.
3. Select the **Files** tab. This shows the top-level directories and files on the share.
    a. You can click on a directory to see the files contained in it and the current alignment status for each file.
    b. You can click on a file name to view details about the file, including the alignment status for individual objectives applied to it.
    c. Alignment status for the file is shown on the **File Alignment** tile. The colour of the circle on that tile indicates the alignment status.
    d. To expand the **File Activity** chart, click on the ⤢ icon. This chart reports I/O activity, not alignment.

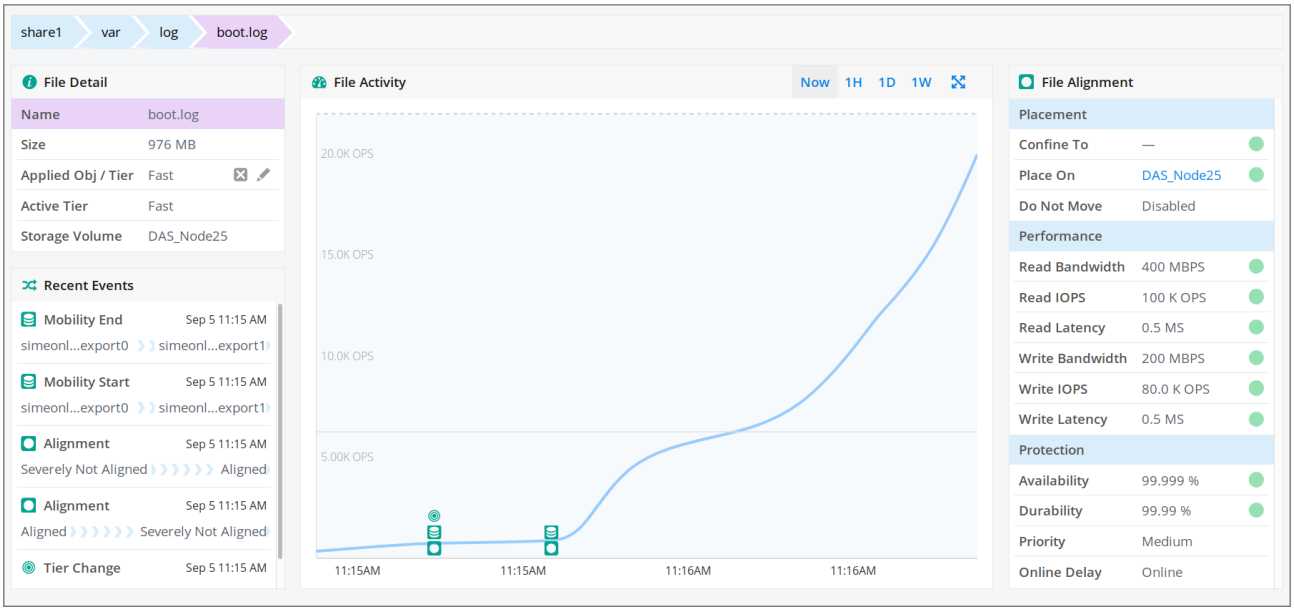


Figure 61. Viewing file IOPs over time

The activity chart shows the current I/O by default. To see I/O activity for the last hour, click on **1H**, for the last day click on **1D**, and for the last week click **1W**.

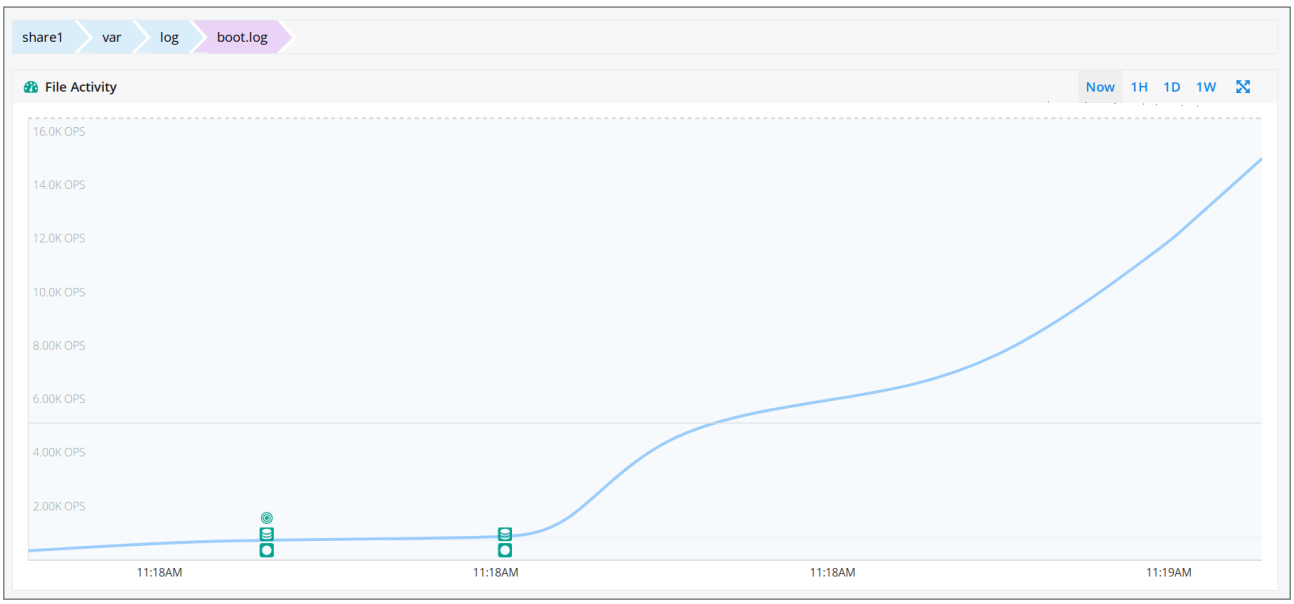


Figure 62. Selecting different look-back periods for file IOPs history

© 2026 - Hammerspace

Resolving Alignment Issues

Under certain conditions, such as when backend volumes are inaccessible, the Product may not be able to maintain alignment with objectives. In these instances, the Hammerspace software makes automated decisions about data placement.

Administrators can take the following actions to resolve compliance issues:

- **Add storage**. Once storage capacity is added, the Product will automatically redistribute workloads across all storage, including the new storage, as required by objectives.
- **Modify Objectives**. Administrators can change the Objective requirements for data that is currently competing, or that might soon compete for resources.

8.7. File Clones and File Snapshots

The Product also supports individual file cloning, which is called file snapshot in the GUI. Despite the snapshot name, this operation creates a clone of the selected file, and the clone is placed in a virtual sub-directory called `.fsnapshot/<FILENAME>/<TIMESTAMP>`.

You can also set up a file cloning schedule and retention policy to protect files on a regular basis.

Cloned files can be recovered manually using the GUI or by accessing the `.fsnapshot` directory using the NFS protocol.



Do not confuse file clones with share snapshots. Share snapshots are the recommended approach to protect more than one file in a share. File clones do not maintain special VERSION metadata, further differentiating them from share snapshots. File clones will not show up as a file version using Windows Previous Versions.

For an example objective that moves file clones to cloud/object storage, see *Keeping file clones (snapshots) on cloud/object storage*.

8.8. File Versioning

The file versioning functionality creates versions of files when changes to file data occur. Although file versioning is like snapshots as both generate copies of data, it has a fundamentally different triggering mechanism. Whereas a schedule triggers snapshots, file versions are triggered by changes to file data. File versions are created when changes to a file are saved.

8.8.1. Configuring File Versioning Using Objectives

File versioning is configured using objectives. The versioning configuration can be applied to an entire share, a directory, or using a selection mechanism, whereby it is triggered by metadata. New file versions are stored in the following format:

`<filename>[#M=<YYYYMMDD.HHMMSS.SSSSS #S=n-nn>].<suffix>`

For example, if we create a file named `testfile.txt` in the engineering share, a subsequent version will look as

© 2026 - Hammerspace

follows:

```
testfile[#M=20211028.224602.46490 #S=0-88].txt
```



For the system to delete expired versioning files, the share must be configured with a snapshot schedule or as part of a global file system.

8.8.2. Configuring File Versioning in the GUI

This procedure outlines the steps to configure file versioning in the Product Management GUI. By following these steps, you can set a file versioning objective at the share, folder, or file level with customizable retention periods, ensuring data availability and easy recovery from accidental changes or deletions.

1. Enter the cluster IP in your browser to open the Hammerspace GUI.
2. Select **Data** from the left-side panel and see the **Shares** tab.
3. Locate the share you want to configure and click the pencil icon for that share in the **Applied Objectives** column.

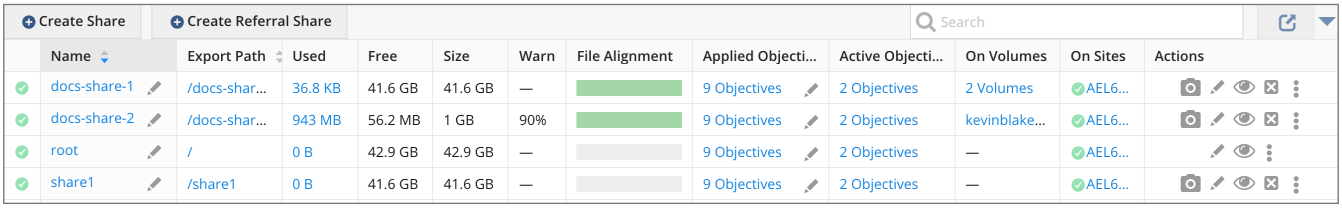


Figure 63. Editing a share

4. In the popup, click **Add Objective**, then in the **Filter** box type "*versioning*" and press `Enter` to see the options available for file versioning.
5. Check the box for a version retention period that best suits your requirements and click **Apply**.

© 2026 - Hammerspace

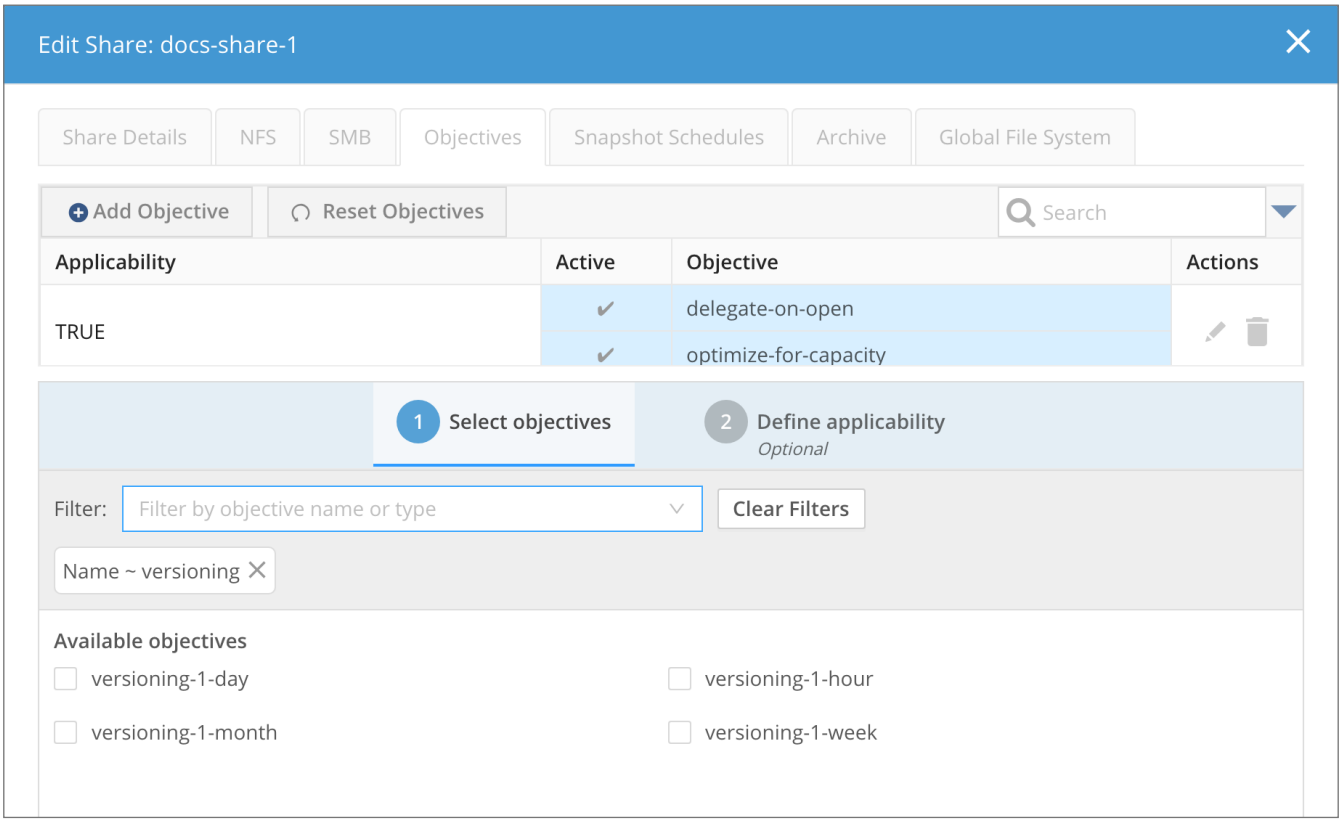


Figure 64. Filtering a set of objectives

6. You can see the objective is set and active. Click **Close** to exit.

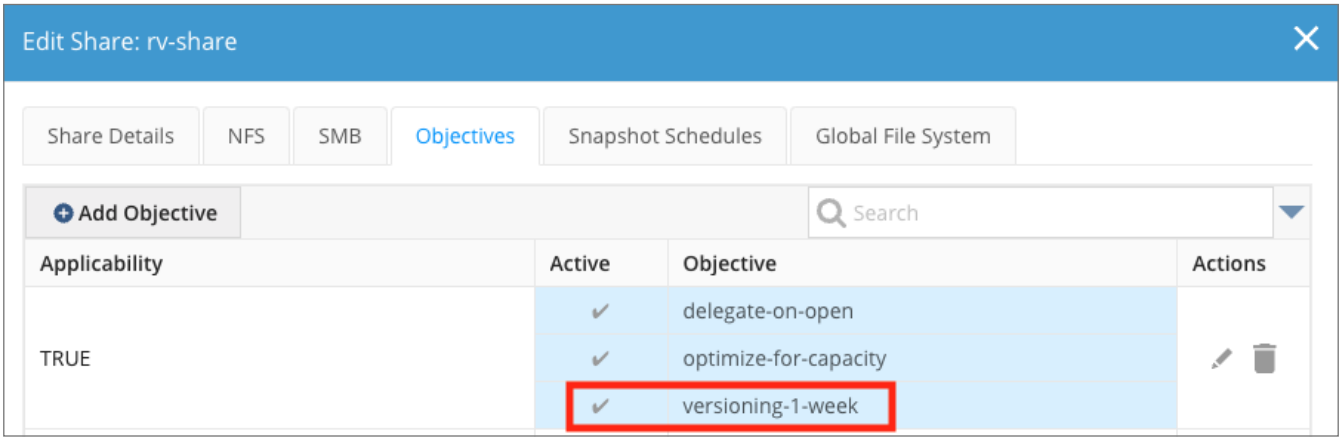


Figure 65. Verifying an objective is active

In the example above, all the directories and files within the share are protected with file versioning and new versions are retained for one week. File versioning protection can be set for a file, folder, or share for the specified duration.

8.8.3. Locating Versioned Files

Find versioned files using the following steps in the GUI:

1. Open the Management GUI and click on **Data** in the left side panel.

© 2026 - Hammerspace

2. In the **Shares** tab, which is active by default, click on the name of the share with the files you want to view.
3. At the top of the **Dashboard** page for the share, click the **Files** tab.
4. In the **Files/Directories** column, click on the `.snapshot` directory.
5. In the **Name** column, click on the **current** directory.
6. View the available directories and files with active backups.

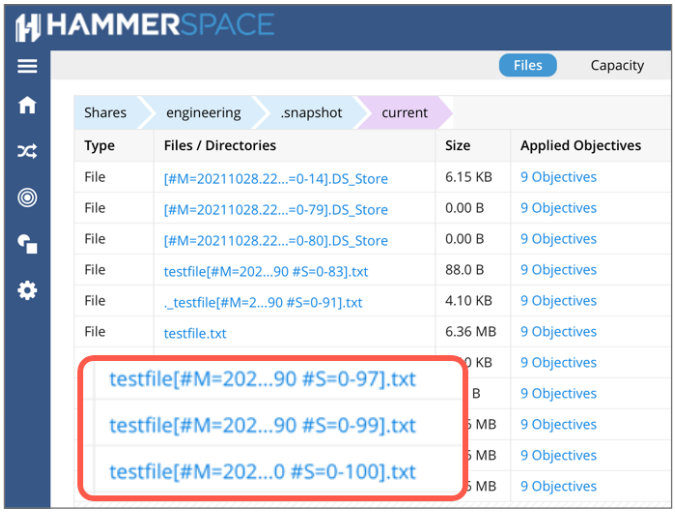


Figure 66. Viewing the list of available backups

8.8.4. Restoring a Previous Version of a File

The Product provides a simple and efficient way to restore files through the GUI. This procedure will guide you through the steps to quickly recover a file from an earlier version.

1. Open the Management GUI and select **Data** from the left-side panel. By default, you will see information in the **Shares** tab.
2. Click on the share on which you want to restore a file version. This opens the dashboard for the share.
3. Click **Files** at the top of the page and navigate to the menu:.snapshot[current] directory.
4. Versioned files are easily and quickly restored by selecting the **Restore** icon to the right of the version you want to recover, as shown below:

© 2026 - Hammerspace

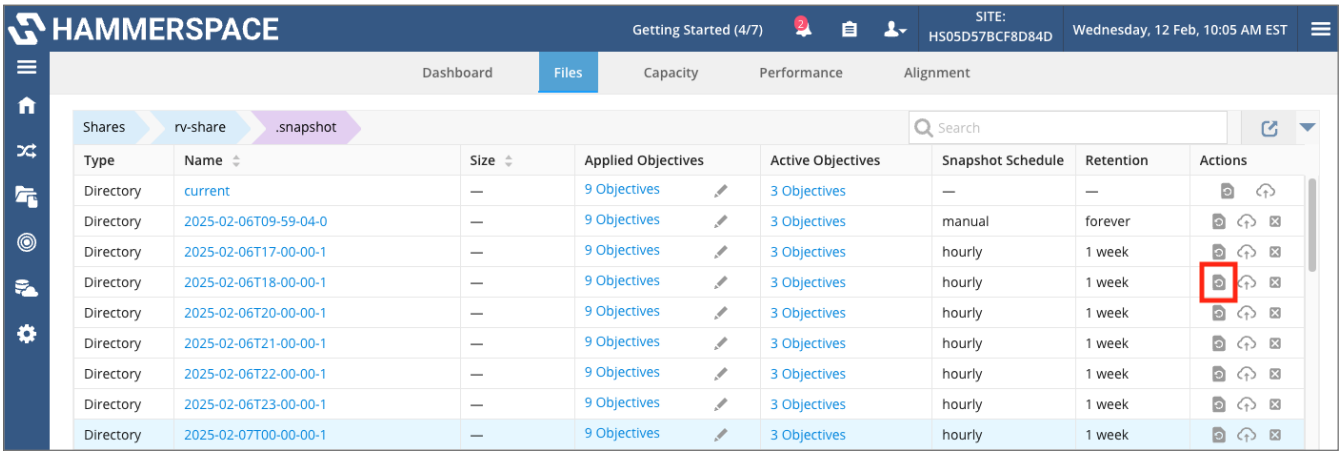


Figure 67. Location of Restore icon

5. In the **Restore Snapshot** popup, select to restore files to their original location, or specify a different location within the share. You can also select to overwrite the contents of the restoration target directory or to create a new target directory for the restored data.
6. Click **Restore** and you will see a notification in the GUI that the restore operation is in progress.
7. You can click the **Tasks** icon to confirm that the restore operation is successful.

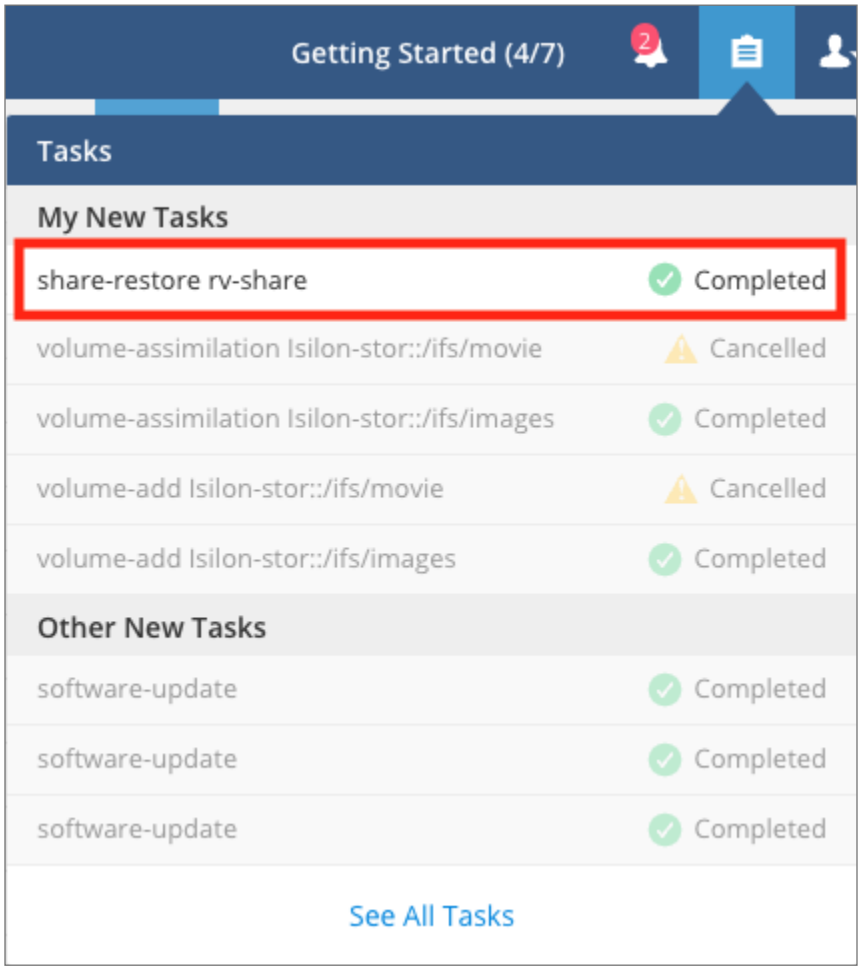


Figure 68. Viewing task-completion status

© 2026 - Hammerspace

8.8.5. Protecting Files from Deletion

The *Undelete* objective enables you to protect file data from deletion for a specified period. Granular options enable protection from accidental or intentional deletion, from a single file up to entire shares. Undelete can also be applied to Multipurpose Internet Mail Extension (MIME) types or through custom metadata tagging and descriptors.

How Undelete Works

When an undelete objective (for example, `undelete-1-day`) is active and a file is deleted:

- The deleted file immediately appears in the share's `.snapshot/current` directory, preserving its original path.
- The file remains recoverable there until the objective's retention period — an hour, a day, a week, or a month — has elapsed.
- After the retention period expires, the file is eligible for cleanup. The storage it occupies is freed when the system next processes the share — through a snapshot delete or, on global file system shares, through replication.



The expiration of an undelete retention period by itself does not reclaim space. For the system to delete expired undelete files, the share must be configured with a snapshot schedule that includes a retention (delete) policy, or be part of a global file system. If you enable an undelete objective on a share that has neither, files that pass their retention period continue to consume storage capacity indefinitely.

When you apply an undelete objective to a share, also configure a snapshot schedule with a retention policy on that share. See Snapshots.



Undelete files captured inside a share snapshot remain available for as long as the snapshot is retained, even past the undelete retention period.

Deleting a share removes **all** data in that share, including files protected by an undelete objective and files protected by WORM. Share deletion does not honor any undelete retention or WORM expiration policies. Before deleting a share, confirm it contains no data you need to keep.

8.8.6. Configuring Undelete Using the GUI

Complete the following steps to enable undelete protection in the GUI. By configuring undelete protection, you can safeguard important data against accidental deletion by retaining deleted files for a specified duration—ranging from an hour to a month—allowing for quick and easy recovery when needed.

1. Open the GUI and select **Data** from the left side panel. By default you will see information in the **Shares** tab.
2. Click on the share from which you want to restore a file version. This opens the dashboard for the share.
3. Click **Files** at the top of the page.
4. Locate the file or directory you want to protect with undelete and click the pencil icon in the **Applied Objectives** column, as shown below.

© 2026 - Hammerspace

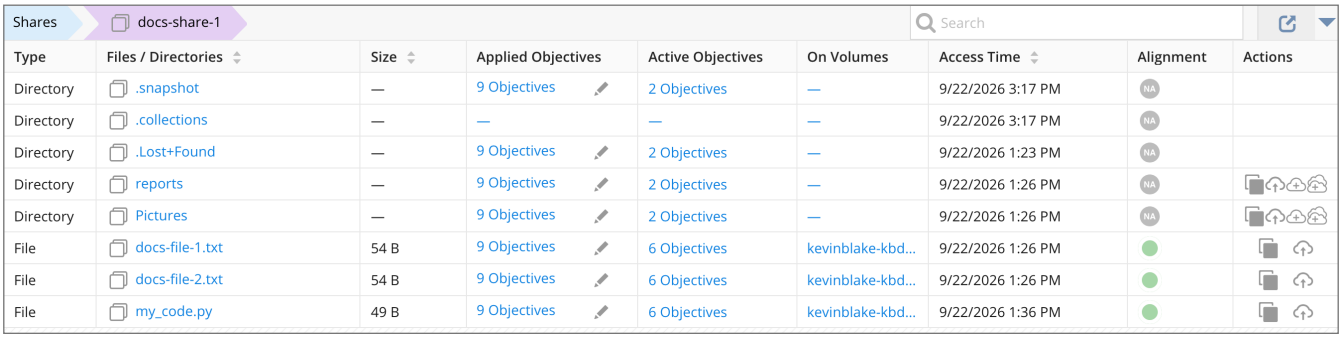


Figure 69. Protecting a directory with undelete

5. In the popup, click **Add Objective**, then in the **Filter** box type "**undelete**" and press `Enter` to see the options available.
6. Check the box for a undelete duration that best suits your requirements and click **Apply**.

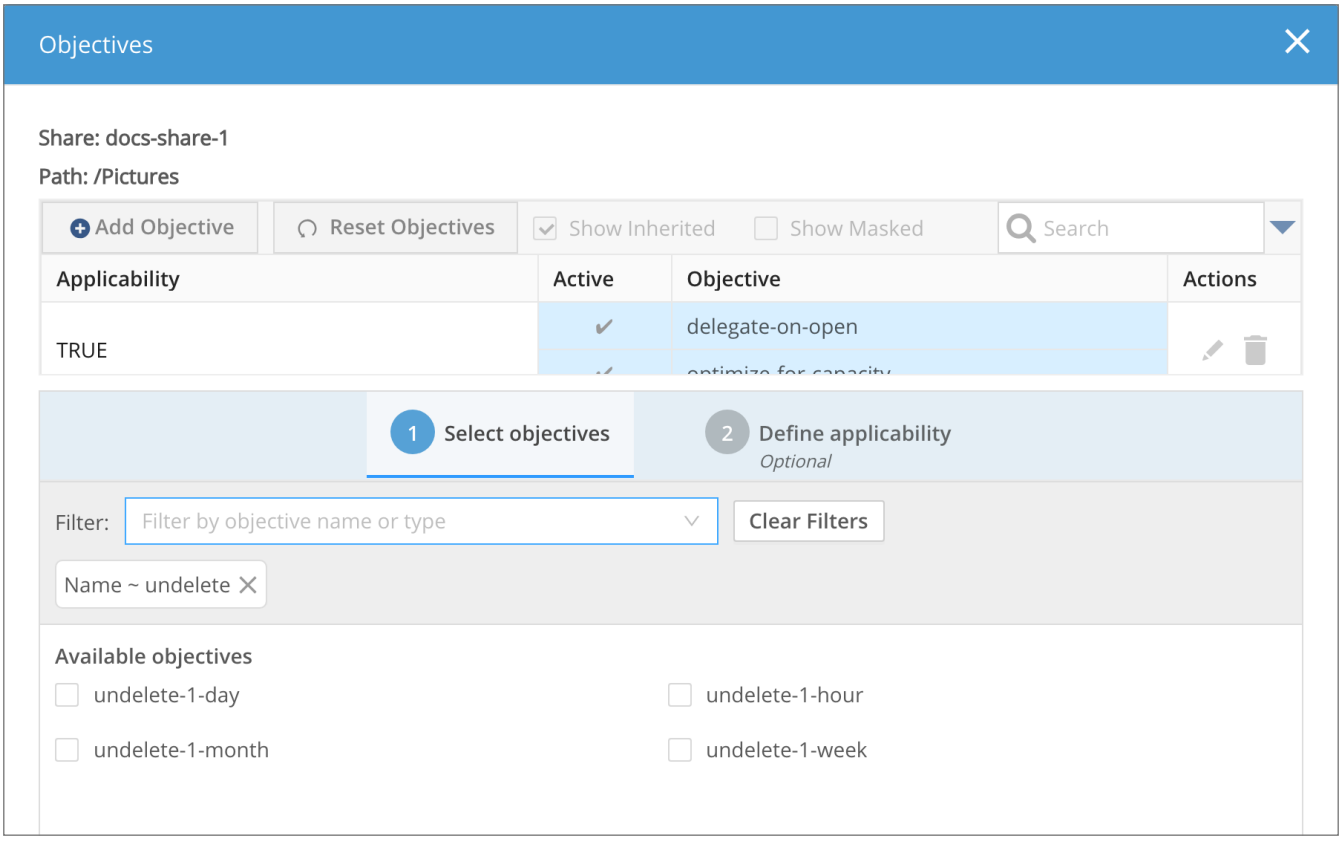


Figure 70. Selecting the term for undelete protection

7. You can now see the objective is set and active. Click **Close** to exit.

© 2026 - Hammerspace

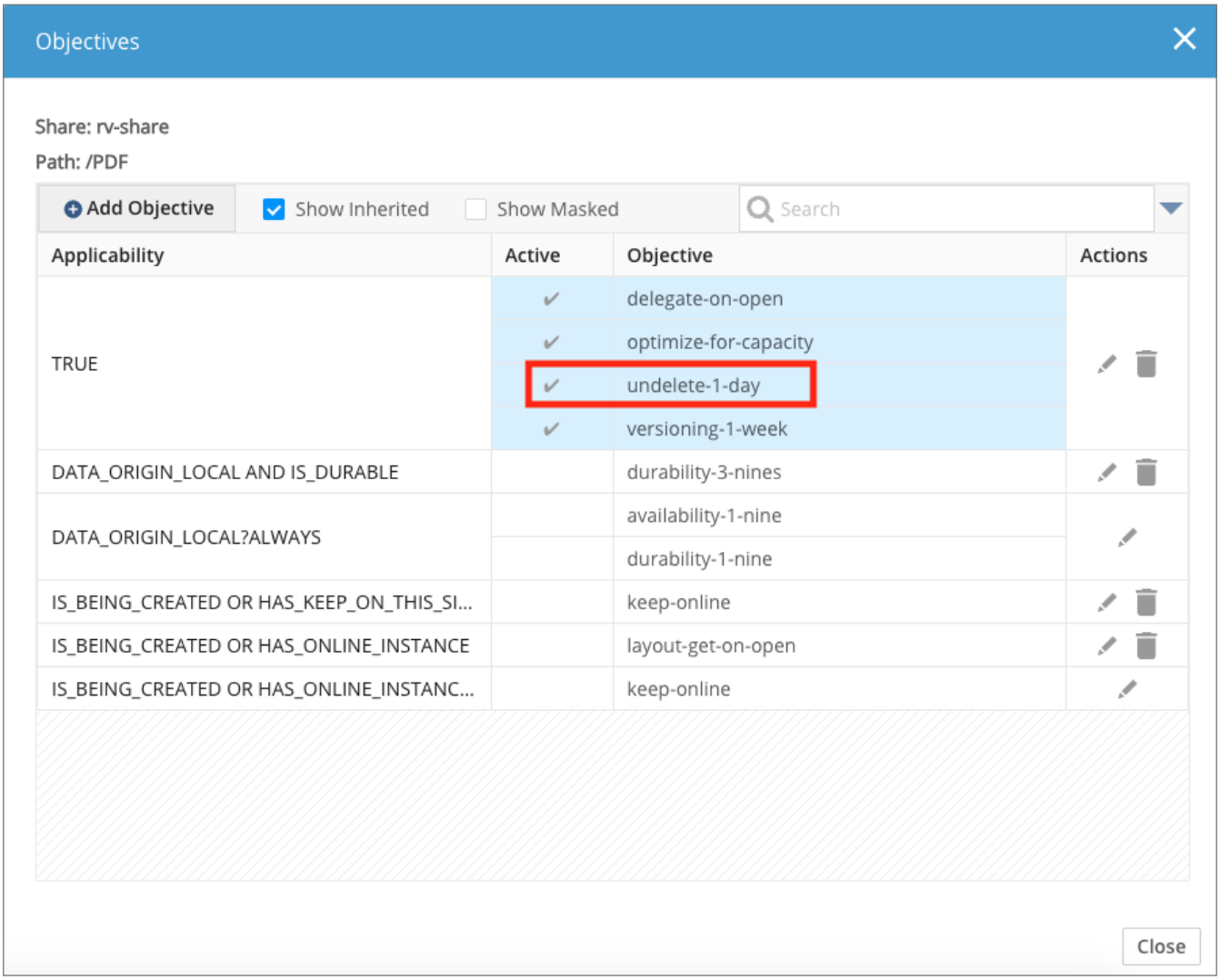


Figure 71. Verifying the objective is active

8.8.7. Configuring Undelete Using the CLI

Deleted share data can be recovered instantly from the ***<share-name>*/.snapshot/current** directory in a share. Files protected with undelete will have the following format:

Undelete file-name format

```
<filename>[D=<YYYYMMDD.HHMMSS.SSSSS>].<suffix>
```

This procedure explains how to configure undelete protection using the Admin CLI. By following these steps, administrators can set undelete objectives for specific shares, folders, and files, ensuring that deleted files are retained for a designated period—ranging from an hour to a month—providing an added layer of data protection and recovery.

1. Log into the Admin CLI via SSH and enter the associated password:

© 2026 - Hammerspace

Log in as admin

```
ssh admin@<hostname|ip address>
```

2. Configure the undelete objective using the following syntax:

Configure undelete

```
share-objective-add --name <share name> --objective undelete-<duration>
```

3. You can set the duration to an hour, a day, a week, or a month. For example, setting an undelete objective for the engineering share for a week would look this way:

Set undelete duration

```
share-objective-add --name engineering --objective undelete-1-week
```

8.8.8. Recovering Deleted File Using the GUI

This procedure guides you through the process of restoring files protected by the undelete feature in the GUI. By following these steps, you can locate and recover deleted files from the `.snapshot/current` directory, ensuring quick restoration to their original location.

1. Open the GUI and select **Data** from the left side panel. By default, you will see information in the **Shares** tab.
2. Click the share where you want to restore a file version. This opens the dashboard for the share.
3. Click **Files** at the top of the page and navigate to the `.snapshot/current` directory.

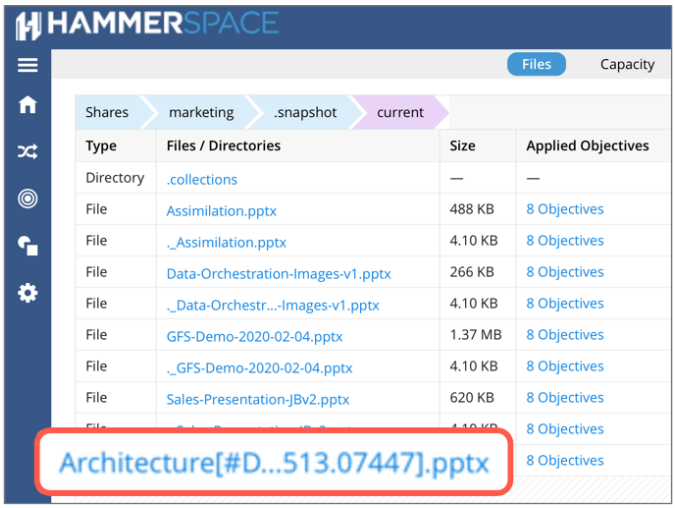


Figure 72. Locating a protected file

4. Locate the undelete-protected file. Notice that a timestamp has been inserted between the filename and its suffix. Mousing over the filename will reveal the entire string, which in this example is `Architecture[#D=20211030.1513.07447].pptx`.

© 2026 - Hammerspace

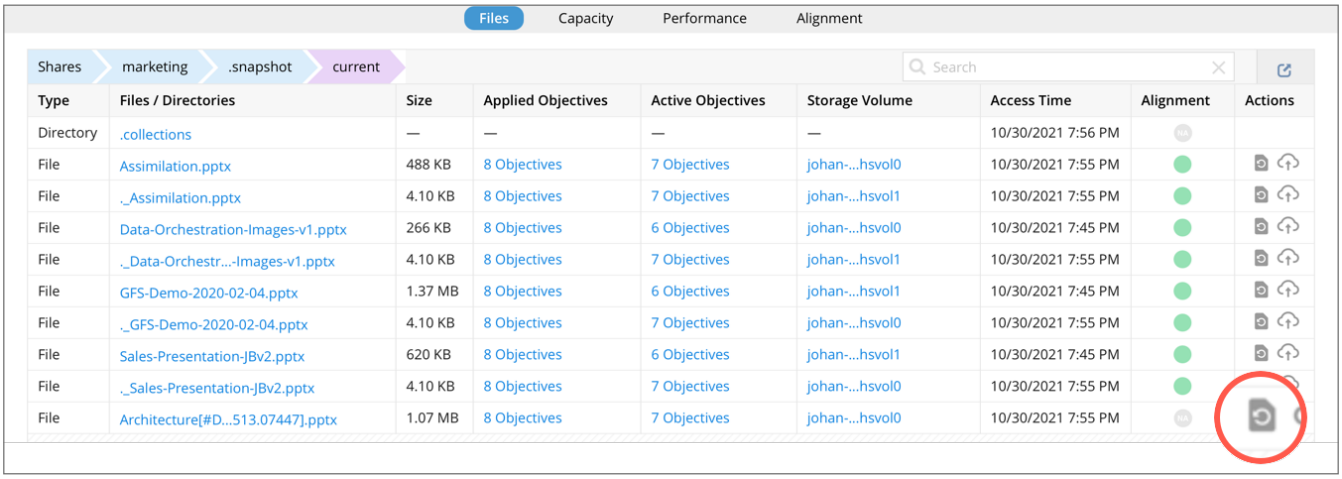


Figure 73. Restoring the file

5. Click on the **Restore** icon in the **Actions** column to begin the file recovery.
6. When the pop-up window opens, click the **Restore** button to finalize the recovery of the file to its original location.

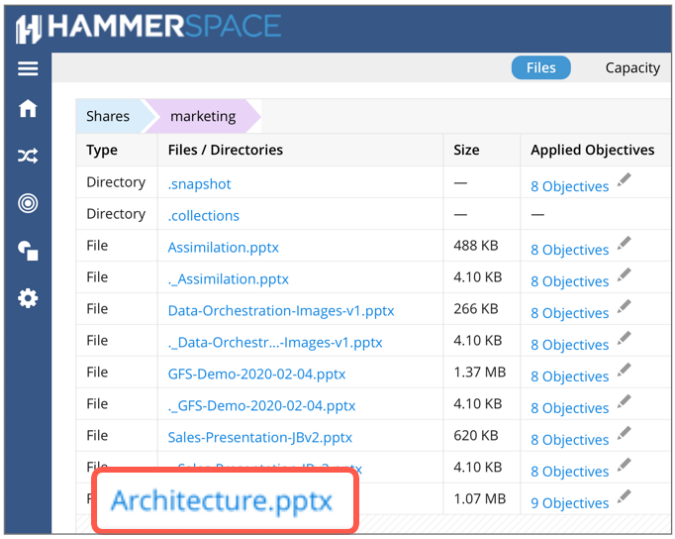


Figure 74. Locating the recovered file

You can navigate back to the original location to find the recovered file.

8.9. Managing Labels

The Product provides the Hammerspace toolkit, `hstk`, open-source utility to help manage data on shares. The utility is available on GitHub: https://github.com/hammer-space/hstk. This section explains how to set up labels for use with the `hstk` utility.

In addition to file-system metadata, the Product supports custom metadata, such as labels. Labels are schema-driven tags with an optional hierarchical layout. Labels can be used to easily tag data in a pre-configured, organized manner with constrained vocabularies, thereby avoiding scenarios in which words are entered with slight misspellings.

Labels can be set on a directory (including share-root) and files. When labels are applied to directories, they automatically inherit further down the directory tree.

Example of label hierarchy

```
Tree
Oak
Coast live oak
Black oak
Shrub oak
Fir
- ...
```

8.9.1. Creating Labels

Labels are managed using the Admin CLI or with REST API calls.

To create a label, use the `label-create` CLI command.

Create a label.

```
label-create --name <label>
```

You can create a label hierarchy using the `--implied-labels` option.

Create a label hierarchy.

```
label-create --name <label> --implied-labels <parent label>
```

A label can be associated with multiple parent labels in the hierarchy.

Associate label with parent labels.

```
label-create --name <label> --implied-labels <parent1, parent2 ...>
```

To create the hierarchy from the example above, use the following command sequence:

Admin CLI Command	Output
label-create --name Tree	Tree
label-create --name Oak --implied-labels Tree	Tree • Oak

© 2026 - Hammerspace

Admin CLI Command	Output
label-create --name Oak --implied-labels "Coast live oak" label-create --name Oak --implied-labels "Black oak" label-create --name Oak --implied-labels "Shrub oak"	Tree • Oak ◦ Coast live oak ◦ Black oak ◦ Shrub oak

8.9.2. Listing Labels

Configured labels in the system can be listed using `label-list`.

Admin CLI

Command:

```
label-list
```

Expected output:

```
total 5
Name:                   Black oak
Implied labels:         [Oak]
Internal ID:            4

Name:                   Coast live oak
Implied labels:         [Oak]
Internal ID:            3

Name:                   Oak
Implied labels:         [Tree]
Internal ID:            2

Name:                   Shrub Oak
Implied labels:         [Oak]
Internal ID:            5

Name:                   Tree
Internal ID:            1
```

8.9.3. Renaming Labels

Labels can be renamed using `label-update`.

© 2026 - Hammerspace

Admin CLI

```
label-update --name <current name> --new-name <new name>
```

Note that renaming a label does not change the relationship with any other labels. Any other label with the update label in the implied-labels list will now reference the new name automatically.

8.9.4. Changing the Label Hierarchy

The implied labels can be updated.

Admin CLI

Command:

```
label-update --name <label> --implied-labels <list of comma separated labels>
```

The implied labels list replaces the existing list. To retain existing implied labels, ensure that they are included in the updated list.

Using the example above, insert a "Large Oaks" and "Small Oaks" label and move existing labels into one of the two choices.

Admin CLI: Inserting implied labels

Command:

```
label-create --name "Large Oaks" --implied-labels Oak

label-create --name "Small Oaks" --implied-labels Oak
```

Then verify the labels with `label-list`:

Command:

```
label-list
```

Expected output:

```
total 7
Name:                   Black oak
Implied labels:         [Oak]
Internal ID:            4

Name:                   Coast live oak
Implied labels:         [Oak]
Internal ID:            3
```


© 2026 - Hammerspace

```
Name:                   Large Oaks
Implied labels:         [Oak]
Internal ID:            6

Name:                   Oak
Implied labels:         [Tree]
Internal ID:            2

Name:                   Shrub Oak
Implied labels:         [Oak]
Internal ID:            5

Name:                   Small Oaks
Implied labels:         [Oak]
Internal ID:            7

Name:                   Tree
Internal ID:            1
```

Move Coast Live Oak and Black Oak to Large Oaks

Command:

```
label-update --name "Coast live oak" --implied-labels "Large Oaks"

label-update --name "Black oak" --implied-labels "Large Oaks"
```

Move Shrub Oak to Small Oaks

Command:

```
label-update --name "Shrub Oak" --implied-labels "Small Oaks"
```

List the Labels After the Change

Command:

```
label-list
```

Expected output:

```
total 7
Name:                   Black oak
Implied labels:         [Large Oaks]
```


© 2026 - Hammerspace

```
Internal ID:            4

Name:                   Coast live oak
Implied labels:         [Large Oaks]
Internal ID:            3

Name:                   Large Oaks
Implied labels:         [Oak]
Internal ID:            6

Name:                   Oak
Implied labels:         [Tree]
Internal ID:            2

Name:                   Shrub Oak
Implied labels:         [Small Oaks]
Internal ID:            5

Name:                   Small Oaks
Implied labels:         [Oak]
Internal ID:            7

Name:                   Tree
Internal ID:            1
```

8.9.5. Removing Labels

When a label is no longer needed, it can be removed using the `label-delete` command.

Admin CLI

```
label-delete --name <label>
```



If you delete a label, its reference will be removed from any other labels that includes it as an implied label. `label-delete` will not delete any other labels.



Deleting labels should be done with care as labels can also be used in objectives to determine file placement and behavior. Removing labels does not update or change any Objectives in the system.

8.9.6. Using the Hs Command with Labels

With labels created in the system, the Hammerspace toolkit (**hstk**) can be used to assign labels to files and directories as metadata. This type of metadata only lives within the file system itself and does not change the contents of the file. The metadata is also replicated when used together with the Global File System feature.

For a detailed understanding of all the capabilities of the **hs** command, please see the **hs** help pages. The **hs** command can be used from Linux or Windows, over both NFS and SMB protocols.

© 2026 - Hammerspace

Adding a Label to a File Using the Hs Command

hstk on a client

```
hs label add <label name> <file and/or directory>
```

Listing All Files and Their Paths with a Particular Label

List all the files containing the "Black oak" label and print the PATH within the share.

hstk on a client

Command:

```
hs eval -e 'HAS_LABEL("Black oak")?PATH' . --recursive
```

Expected output:

```
"./Pictures/2020/Apr/DCS2522.jpg"
"./Pictures/2020/Dec/DCS3555.jpg"
"./Pictures/2021/May/DCS5524.jpg"
```

Listing All Files with a Particular Label

This label was not manually set on any files or directories but rather it is inherited through the hierarchy of the label definitions.

hstk on a client

Command:

```
hs eval -e 'HAS_LABEL("Oak")?PATH' . --recursive
```

Expected output:

```
"./Pictures/2020/Apr/DCS2521.jpg"
"./Pictures/2020/Apr/DCS2522.jpg"
"./Pictures/2020/Dec/DCS3555.jpg"
"./Pictures/2021/May/DCS5524.jpg"
"./Pictures/2021/May/DCS5534.jpg"
```

8.10. Managing Shares

A share in Hammerspace is similar in concept to an export in a traditional NAS filer in that NFS and/or SMB clients with permission can mount and access the data within the share.

© 2026 - Hammerspace

The Product supports multiple shares, making it easy to manage data and securely isolate data between clients and applications.

The default size of a share is limited only by the capacity of the file-backed storage. Shares automatically grow as new file storage capacity is added. Shares can also be set to a logical size that is unrelated to the physical space in the system.

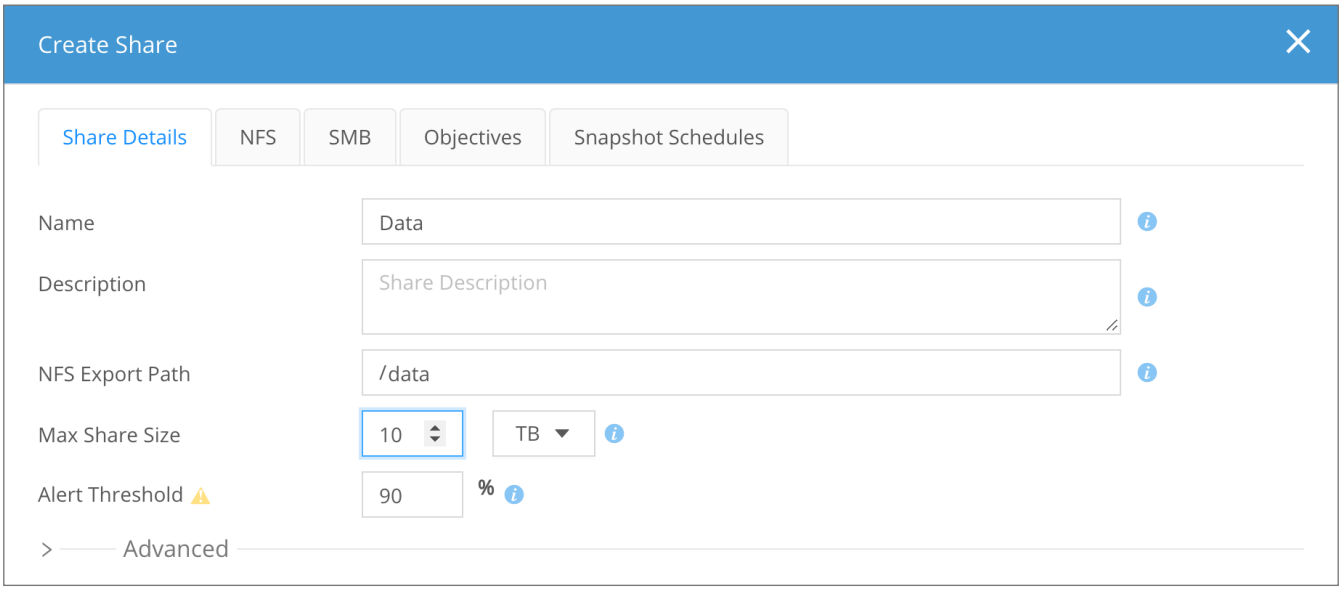
The file-backing storage is a common resource pool across all shares unless objectives have been applied to control where data is located.

Administrators can apply objectives to shares, directories, and/or files within a share to ensure that data in the share is stored on storage that meets its objectives.

8.10.1. Creating a Share

Complete the following steps to create a share:

1. Select **Data** from the left side panel. By default you will see the **Shares** tab.
2. On the Shares tab, click **Create Share**.



3. On the **Share Details** tab, enter the following information about the share:
   - **Name** that will identify the share for SMB clients. The name represents the SMB share name, which is case insensitive. The name is also used to reference and manage the share within Hammerspace GUI, CLI, and REST API.
   - **Description** (Optional) free text describing the share.
   - **NFS Export Path** for the share in the namespace for use with NFS. The path is filled in from the name as you type it (for example, a share named `Data` gets the export path `/Data`); edit it if you want a different path. If the path does not already exist, it is automatically created.
   - **Max Share Size** (Optional) maximum size for the share. This is the capacity limit that an NFS or SMB client would see when using the share. Max Share Size is a soft limit: Hammerspace enforces it after

© 2026 - Hammerspace

the fact, so the amount of data in the share can exceed it. See Max Share Size Enforcement. The share size can be increased or decreased later by editing the share. The size cannot be decreased below the actual amount of data stored in the share.

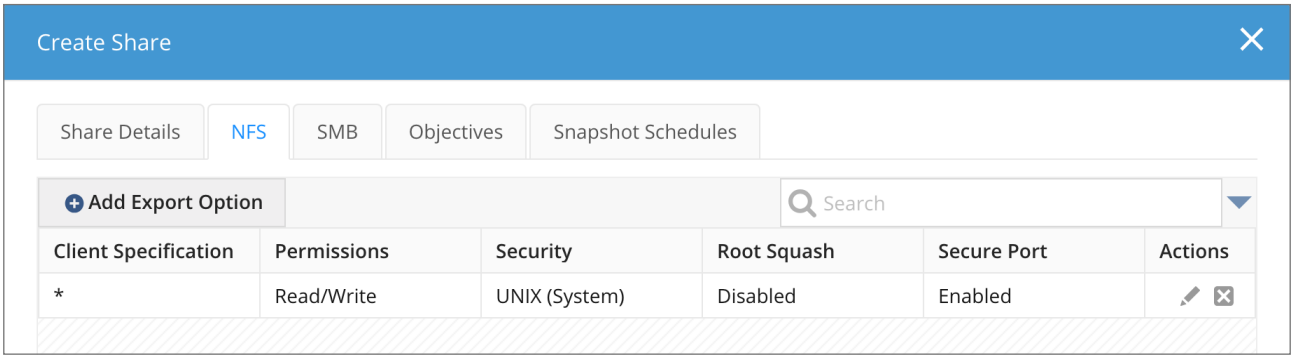
The default share size is the total physical capacity across file-based storage volumes and does not include cloud/object volume capacity. The setting for max share size is not constrained by available capacity and can be set to the desired size.

- **Alert Threshold** (Optional) triggers warning events for share usage when the specified percentage of max share size is reached. Note that this field is only visible when a Max Share Size is specified. For what happens when usage reaches the limit, see Max Share Size Enforcement.

The **Advanced** section, collapsed by default, holds additional settings that are not required to create a share.

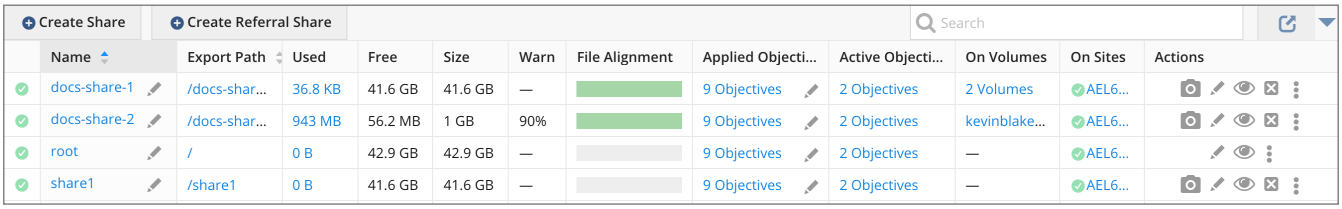
4. (Optional) To control which NFS clients can access the share, select the **NFS** tab. The tab is unavailable until a name and NFS export path have been entered. By default the share has one export option that grants Read/Write access to any client (*). Click **+ Add Export Option** to add an export option, and enter the following:
   - **Client Specification**: the clients the option applies to, as an IP address, a CIDR subnet, a hostname, a subdomain, or a netgroup (`@netgroup`).
   - **Permissions**: **Read-Only** or **Read/Write**.
   - **Security**: the NFS security flavor, `UNIX (System)`, `Krb5`, `Krb5i`, or `Krb5p`.
   - **Root Squash**: select the checkbox to enable root squash.
     Root squash reduces the access rights of remote root users (superusers). Enabling it blocks remote root users from certain operations that are determined by which client they are using.
   - **Secure Port**: whether clients must connect from a privileged (secure) port.

   Repeat for each additional client specification. These settings only affect NFS access.



5. Click **Create**. Shares are created as a background operation. Once the operation is complete, the share will appear in the list on the Shares tab.

© 2026 - Hammerspace



SMB aliases create additional SMB shares on the system from an existing share. See Setting Aliases for SMB for details.

The Objectives tab is empty until after the share has been created.

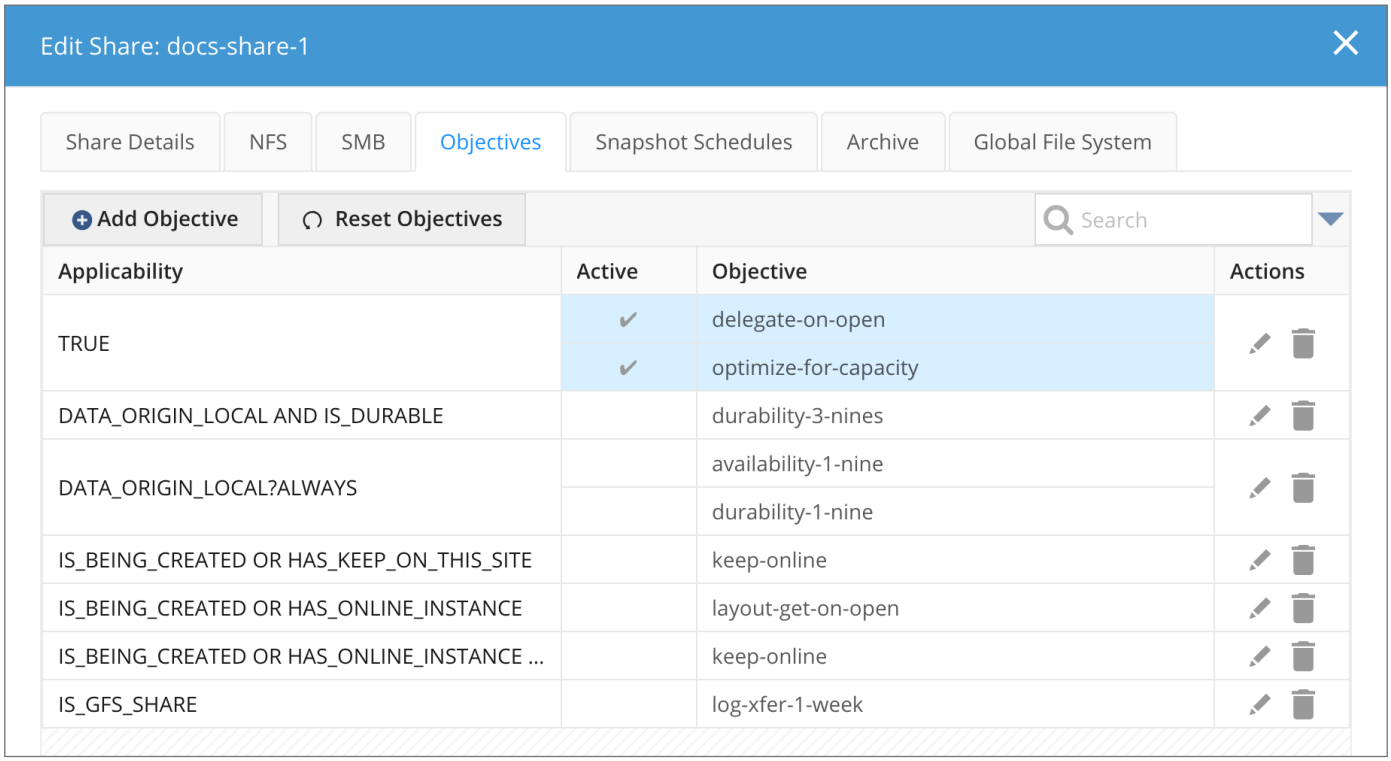


Figure 75. Objectives tab

Once a share is created, the default objectives are applied. The default objectives do not need to be modified. Instead, it is a **best practice** to create and add new objectives using **Add Objective**.

8.10.2. Deleting a Share

You can delete a share and its contents using the Admin CLI or the Management GUI.

When a share is deleted, all the data for that share is also deleted, including files protected by an undelete objective and files protected by WORM. Share deletion does not honor any undelete retention or WORM expiration policies. Before deleting a share, confirm it contains no data you need to keep. For more information, see File Versioning.

- You can only delete a share with a confirmation prompt.
- By default, the share delete will not be executed for 24 hours.

© 2026 - Hammerspace

- If you have a share snapshot schedule, the share delete is not allowed.

Shares Referenced by SMB Autohome Entries

You cannot delete a share while any SMB autohome entry references it. The delete fails with an error saying that the share is referenced by (or used by) SMB autohome entries. Disabling SMB autohomes does not release the share: the entries still exist, and you must remove them.

To delete a share that SMB autohome entries reference:

1. List the SMB configuration, and note the **ID** of each entry under **Autohome Entries** whose **Share** field names the share you want to delete:

   Command:

   ```
   smb-config --view
   ```

2. Delete each of those entries by ID, including the default entry if it references the share:

   Command:

   ```
   smb-config --autohome-delete --id <entry-id>
   ```

3. Delete the share as usual, with `share-delete`.

You can also delete the entries in the Management GUI, from the **User Auto Homes** table in the SMB settings.

Using the Admin CLI

Admin CLI: share-delete options

Command:

```
share-delete --help
```

Expected output:

```
Usage: share-delete [options]
Delete a share

  Options:
    --async         Causes the command to be executed asynchronously. A
                    reference task identifier will be returned
    --delay         Postpone share space reclaim for period of time (for example
                    24h). Mutually exclusive with "--now"
    --delete-path   Delete the share path directory when the share is purged.
                    Directory must be empty in order to be deleted
```


© 2026 - Hammerspace

```
  --help          Display this help and exit
* --id            The ID of the share to delete. Required unless any of the
                  following is specified:
                  --internal-id, --name
* --internal-id   The internal ID of the share to delete. Required unless any
                  of the following is specified:
                  --name, --id
* --name          The name of the share to delete. Required unless any of the
                  following is specified:
                  --internal-id, --id
  --no-timeout    When running a task-based command synchronously, monitor for
                  task completion indefinitely
  --now           Share space will be reclaimed immediately. Mutually
                  exclusive with "--delay"
```

8.10.3. Changing the Default File Placement Behavior

By default, directories and files inherit the system's default objectives, which automatically distribute files across all available file-based storage volumes, such as NAS exports and DSX volumes. This behavior can be easily modified by selecting additional objectives for the share.

> An objective can be applied to a share, directory, or file.

To apply additional objectives to a share, follow these steps:

1. Select **Shares** from the left side panel.
2. Locate the share either by scrolling through the list or by using the search function.
3. In the **Applied Objectives** column, click the edit (▯) icon. Alternatively, click the Edit pen for the share and navigate to the Objectives tab.
4. In the **Objectives** tab, click **Add Objective**.

   > It is recommended to create a new objective rather than modifying predefined objectives.

5. Select the objective(s) you wish to add to the share and then click the **Define Applicability** tab to further refine the objective.
6. The system will automatically begin realigning data based on the additional objective(s) you selected.

8.11. Max Share Size Enforcement

Max Share Size is a soft limit on the amount of data in a share. Hammerspace enforces the limit after the fact: it refuses new writes once it has recorded that the share holds more data than the limit allows. Because of this, the amount of data in a share can exceed its Max Share Size.

This section describes how the limit is detected, what NFS and SMB clients see when a share is full, and how to plan for it. To set the limit, see Creating a Share.

> The GUI calls this setting **Max Share Size**. The Admin CLI and the system events call the

© 2026 - Hammerspace

> same setting the share *quota*: the `--size` option of the `share-create` and `share-update` commands is described as a quota, and the events are named `SHARE_QUOTA_WARNING` and `SHARE_QUOTA_EXCEEDED`.

Max Share Size is separate from the capacity of the storage volumes behind the share. A share can run out of storage capacity before it reaches its Max Share Size. See Capacity Management.

8.11.1. How the Share Size Limit Is Detected

Clients read and write file data directly on the storage volumes. The data does not pass through the Anvil, the metadata server that manages the share. For this reason, Hammerspace cannot count each write against the limit as it happens. Instead, it keeps a record of the space each share has used, updates that record as usage is reported, and compares the recorded usage with Max Share Size.

When the recorded usage is greater than Max Share Size, Hammerspace treats the share as full and does the following:

- Asks NFS clients that hold permission to write directly to storage (a pNFS *write layout*) for files in the share to return that permission. The request is advisory. A client that is already writing a file is not forced to stop.
- Refuses new requests to write to files in the share. What the client sees depends on the protocol, as the following sections describe.

Requests to read files in a full share are not refused.

Because the limit is checked against recorded usage, the share can already be over its limit by the time new writes are refused.

The share accepts new writes again when its recorded usage is no longer greater than its Max Share Size, either because data was removed or because the limit was raised or cleared.

8.11.2. NFS Clients on a Full Share

When a share is full, Hammerspace reports the condition to NFS clients as a no-space error. An application on a Linux client sees `No space left on device` (`ENOSPC`). Hammerspace does not return a quota-exceeded error (`EDQUOT`) for Max Share Size.

NFSv4.2 Clients

An NFSv4.2 client writes to storage directly, using a write layout that it requests from the Anvil. On a full share:

- A request for a new write layout is refused with the no-space error. Opening a file for writing is not refused by itself. The error is returned when the client requests the layout it needs to write.
- A request for a read layout is not refused, so files in the share can still be read.
- A client that already holds a write layout for a file is asked to return it, but is not forced to. Such a client can continue to write to that file.
- Creating a new file can leave an empty file behind. The file name is created before the write is refused.

© 2026 - Hammerspace

NFSv3 Clients

NFSv3 clients receive the same no-space error. An NFSv3 client usually learns of the error when it sends a write that must be committed to storage immediately, or when it commits a series of earlier writes. The application sees the error from a synchronous write, or when it calls `fsync()`.

8.11.3. SMB Clients on a Full Share

In Hammerspace 5.2 and 5.3, Max Share Size does not stop an SMB client from creating new files in a full share. SMB clients can continue to create and write new files, so the amount of data in the share keeps growing past the limit.



Do not rely on Max Share Size to cap the amount of data that SMB clients store in a share. Set an Alert Threshold and act on the share-capacity events before the share reaches its limit.

8.11.4. Best Practices for Max Share Size

- Treat Max Share Size as a soft limit. If the data in a share must never exceed a fixed amount, set the limit below that amount to leave room for data written before the limit is enforced.
- Set an **Alert Threshold** when you set a Max Share Size, so that a `SHARE_QUOTA_WARNING` event is raised while there is still room to act. See Configuring Share-Capacity Notifications.
- Send share-capacity events to the people who can act on them. A `SHARE_QUOTA_EXCEEDED` event means that the share has reached its limit and that NFS clients are being refused new writes, or soon will be.
- When a share reaches its limit, remove data from the share or raise the limit. To change the limit, edit the share in the GUI, or use the `--size` option of the `share-update` command. To remove the limit, use `--size -clear`.
- For shares used by SMB clients, keep usage below the limit. SMB clients are not stopped at the limit in the same way as NFS clients. See SMB Clients on a Full Share.

8.12. Referral Shares

A referral share is a type of share that does not serve data itself but redirects clients to another system's export path. Instead of managing files locally within Hammerspace, the share acts as a pointer to an external NFS export, typically another Hammerspace cluster.

A referral share consists of

- **Referral or target path:** The export path of the target share (e.g., `/remote/target`). Referred to as "referral path" in the CLI output and "target path" in the Management GUI.
- **Referral or target address:** One or more IPv4, IPv6, or hostnames where the target share is hosted. It is recommended to use the data cluster IP of the target Hammerspace cluster. Referred to as "referral address" in the CLI and "target address" in the Management GUI.
- **isReferral or referral flag:** Marks the share as a referral, letting the system and GUI distinguish it from normal shares. Referred to as "isReferral" in the CLI and "referral" in the Management GUI.

© 2026 - Hammerspace

Users can create referral shares through either the Hammerspace GUI, Admin CLI, or the REST API.

This section includes the following subsections:

- REST API Rules
- Client Access and Transparency
- Limitations
- Creating a Referral Share
- Updating a Referral Share
- Deleting a Referral Share
- Additional Tasks

8.12.1. REST API Rules

Use the existing /shares endpoints with referral fields populated.

- On creation, both **referral.path** and at least one **referral.address** are required.
- On update, you can modify **referral.path** or adjust addresses.

Validation rules include:

- The referral path must begin with `/`.
- At least one valid referral address (IPv4, IPv6, or hostname) must be provided.
- Local addresses cannot match floating IPs or Anvil static IPs.
- You cannot convert a normal share into a referral share (or vice versa).
- You cannot blank out the path or remove all addresses.

8.12.2. Client Access and Transparency

To clients, a referral share appears like any other NFSv4 export. When mounted, the NFS client is redirected to the external system specified by the referral path and addresses.

This redirection is *transparent to the end-user application*. The only visible difference is that data is served from the referred-to system (target) rather than the local system.

8.12.3. Limitations

- **Conversion restrictions**: You cannot change share types (from normal to referral or vice versa) after they have been created.
- **Path/address immutability:** You cannot clear the referral path or leave addresses empty.
- **IP restrictions:** *Referral addresses* should not be floating IPs or Anvil static IPs.
- **Not validated for reachability:** The system does not check if the referral target is a valid target.

© 2026 - Hammerspace

- **Excluded from some operations:**
  - Cannot be an assimilation destination
  - No objectives, GFS participation, or snapshot schedules can be applied to the referral shares
  - Many file system-related statistics are skipped for referral shares

8.12.4. Creating a Referral Share

Using the Admin CLI

Command:

```
share-create --name referral-share --path /test/referral \
--referral-path /remote/target \
--referral-address 192.168.1.100 \
--referral-address 192.168.1.101
```

Using the Management GUI

Creating a referral share works like creating a normal share:

1. In the **Shares** view, next to the **Create Share** button, select **Create Referral Share**. The wizard presents three tabs:
   - **Share Details** – Specify name, export path, and enable the Referral Share option.
   - **NFS** – Enter the NFS specification.
   - **SMB** – Optional settings if SMB export integration is needed.
2. Click **Create** to finish.

8.12.5. Updating a Referral Share

Updating the Referral Path Using the Admin CLI

Command:

```
share-update --name referral-share --referral-path /new/remote/path
```

8.12.6. Deleting a Referral Share

Referral shares are deleted the same way as regular shares.

Using the Hammerspace Admin CLI

Command:

© 2026 - Hammerspace

```
share-delete --name referral-share
```

Using the REST API

Use the `/shares/{id}` REST API DELETE endpoint.

When deleted, the referral mapping is removed. Since the data resides on an external system, deleting a referral share does not delete underlying data on the target system.

8.12.7. Additional Tasks

Adding or Removing Referral Addresses Using the Admin CLI

Admin CLI

```
share-update --name referral-share --referral-address-add 10.0.0.1
share-update --name referral-share --referral-address-remove 192.168.1.101
```

Replacing All Referral Addresses

Command:

```
share-update --name referral-share --referral-address 10.0.0.1 --referral-address 10.0.0.2
```

8.13. Snapshots

Hammerspace supports share-level snapshots to protect data in a share. Share snapshots are nondisruptive to client reads and writes. Snapshots can be manually triggered from the GUI, CLI, or REST API, or scheduled to happen automatically. Snapshot schedules also have retention rules. Each share is limited to 4,095 snapshots.

A snapshot consumes no extra data space until existing data in the share is modified. Then only the modified files use additional space. Each snapshot will use a small amount of metadata space for the snapshot itself and for every file modified after the snapshot was taken.



Offloading older snapshot data from primary storage into cloud/object storage is recommended to reduce the capacity impact of retaining many snapshots. When data is offloaded to cloud/object storage, it is globally deduplicated and compressed, creating a highly space-efficient backup. See the section *titled "Move snapshots to cloud/object storage"* for an example.

8.13.1. Accessing Snapshots from Clients

Hammerspace enables users with permission to view snapshot files directly in their client and perform self-service recovery by copying data from a snapshot.

© 2026 - Hammerspace

Linux or macOS NFS Client

The following example shows how to access a recovery snapshot from a Linux or macOS NFS client:

1. From a Linux client that has mounted a Hammerspace share, a user accidentally modifies their latest software project called **my_code.py**.

   Linux client user error

   Command:

   ```
   echo "my bad" > my_code.py
   ```

2. Access file snapshots in the `.snapshot` directory.

   Listing the snapshot directory

   Command:

   ```
   ls .snapshot/
   ```

   Expected output:

   ```
   2026-09-21T23-46-49-0
   2026-09-21T23-46-56-0
   2026-09-21T23-47-02-0
   current
   ```

3. Copy the file from the snapshot back to the live tree under a new name, then move it into place:

   Restore a file from a snapshot

   Command:

   ```
   cp .snapshot/<snapshot>/my_code.py my_code.py.restored
   mv my_code.py.restored my_code.py
   ```

   where `<snapshot>` is one of the directories listed in the previous step.

   The client sees the live file and its snapshot copy as the same file, so copying the snapshot copy directly over the live file fails with "are the same file", and redirecting `cat` output into the live file empties it. Always copy to a new name first.

Windows Client

Windows users can browse and recover files directly by right clicking on the file in Windows Explorer and selecting **Previous Versions**.

© 2026 - Hammerspace

8.13.2. Creating an Instant Snapshot

Snapshots capture a point-in-time view of your data, allowing you to protect against accidental deletions or modifications. By scheduling and setting retention policies, you can ensure long-term data availability and easy recovery when needed. This procedure provides steps for creating snapshots using the GUI.

1. Open the GUI and select **Data** from the left side panel. By default you will see information in the **Shares** tab.
2. Identify the share you want to snapshot and click on the **Snapshot** icon in the **Actions** column to the right of that share. In the Snapshot Now popup, click **Snapshot**.

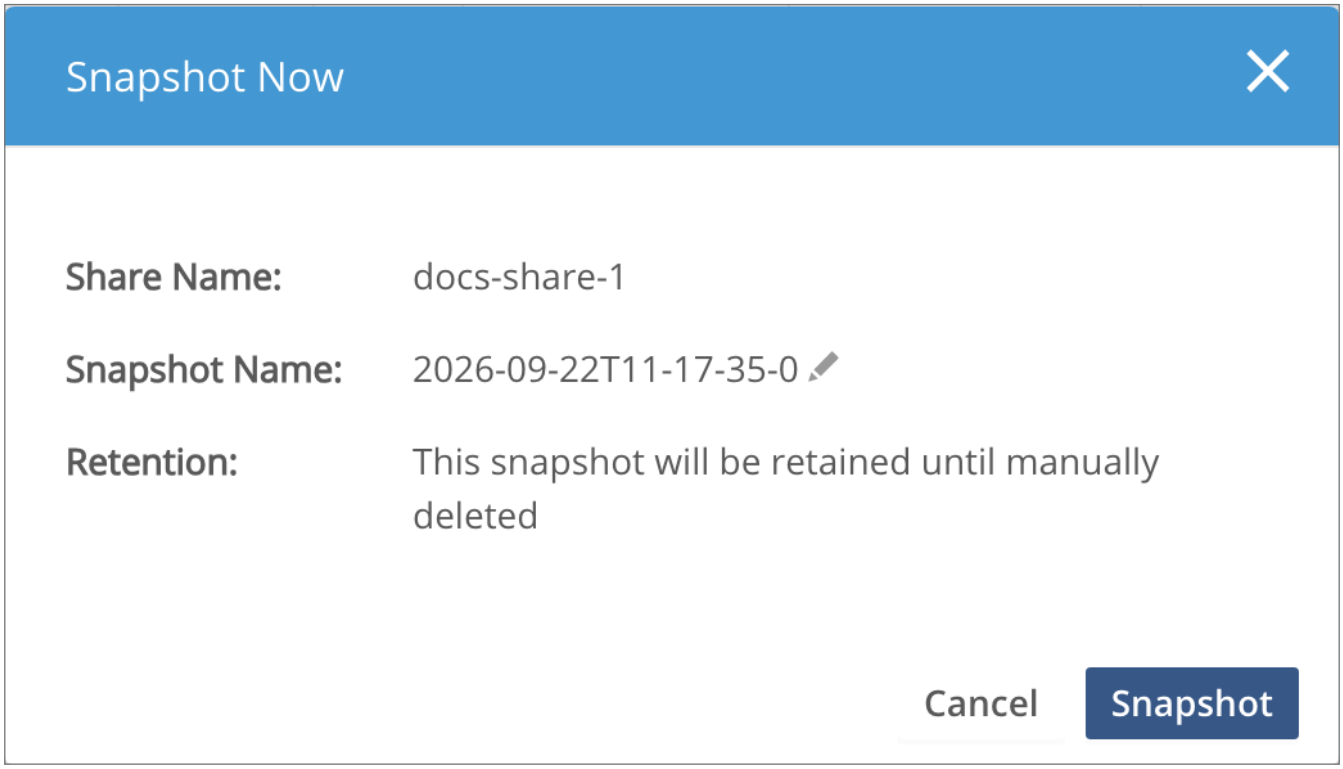


Figure 76. On-demand snapshot popup

You will see a message verifying that the snapshot was created successfully.

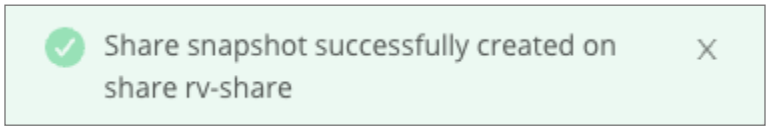


Figure 77. Snapshot successfully created

When no longer needed, the snapshot must be manually deleted.

8.13.3. Custom Schedules and Retention Policies

You can define custom schedules and retention policies to use when creating a share snapshot. Schedules can be very flexible and be defined to run a certain Dates, Days of the Week, Hours, Minutes, etc. The following examples show how to define schedules and retention policies and then create a share snapshot with a schedule and retention policy.

© 2026 - Hammerspace

Schedules can also be applied to tasks other than snapshots.

Snapshots are quick and space efficient, but they can consume large amounts of storage over time because blocks are not released when files are deleted if the files are in a snapshot.

Retention policies applied to a snapshot task cause the snapshot to be automatically deleted either at a certain age, or after a certain number of newer snapshots are created.

Snapshots can also be offloaded into cloud/object storage to remove any space pressure from primary storage.

Custom Scheduling Examples for the Admin CLI

Creating a Custom Snapshot Schedule

Admin CLI

```
schedule-create --day-of-week MON-FRI --hour 12 --minute 30 --name my-schedule
```

Viewing Snapshot Schedules

Admin CLI

Command:

```
schedule-list
```

Custom Retention Examples for the Admin CLI

Creating a Snapshot Retention-Time Policy

Admin CLI

```
snapshot-retention-create --retention-time 14:00:00 --name fortnight
```

Creating a Policy for Number of Snapshot Copies to Retain

Admin CLI

```
snapshot-retention-create --number-of-copies 5 --name my-retain
```

Creating a Share-Snapshot Schedule and Retention Policy

Admin CLI

```
share-snapshot-create --share-name test --schedule-name my-schedule --retention-name my-retain
```


© 2026 - Hammerspace

8.13.4. Deleting Snapshots Using the Management GUI

Over time, saved snapshots take up a lot of storage space so it is recommended that you remove them from your snapshot stores regularly. Use the following steps to delete a snapshot from storage using the GUI.

1. Open the GUI and select **Data** from the left side panel. By default, you will see information in the **Shares** tab.
2. Click the name of share that contains the snapshot(s) to be deleted.
3. Click the Files tab and then click the `.snapshot` directory.
4. Locate the snapshot(s) you want to delete from the saved snapshot files.

> Do not delete the **current** file.

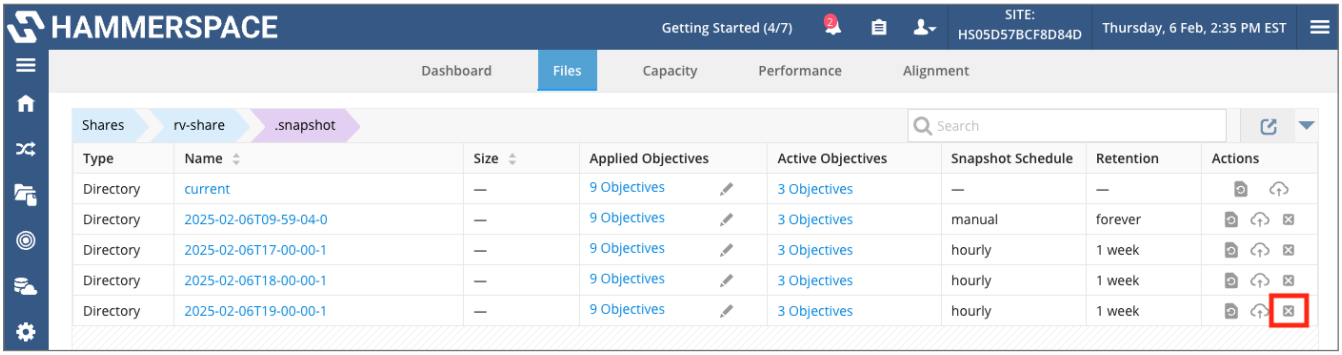


Figure 78. Location of the Delete icon

5. Click on the delete icon to the far right of the row for the snapshot to be deleted.
6. Confirm you want to delete the snapshot file by clicking **Delete**. The file is immediately deleted.

8.13.5. Listing Snapshots Using the Admin CLI

You can list the snapshots of a share:

Admin CLI: List snapshots of a share

```
share-snapshot-list --share-name test
```

8.13.6. Restoring a Snapshot

You can restore a snapshot using the GUI:

1. Open the GUI and select **Data** from the left side panel. By default you will see information in the **Shares** tab.
2. Click on the share name and select the **Files** tab.
3. Click on the .**snapshot** directory and find the snapshot you want to restore, then click the **restore** icon to the far right of the row for that snapshot.

© 2026 - Hammerspace

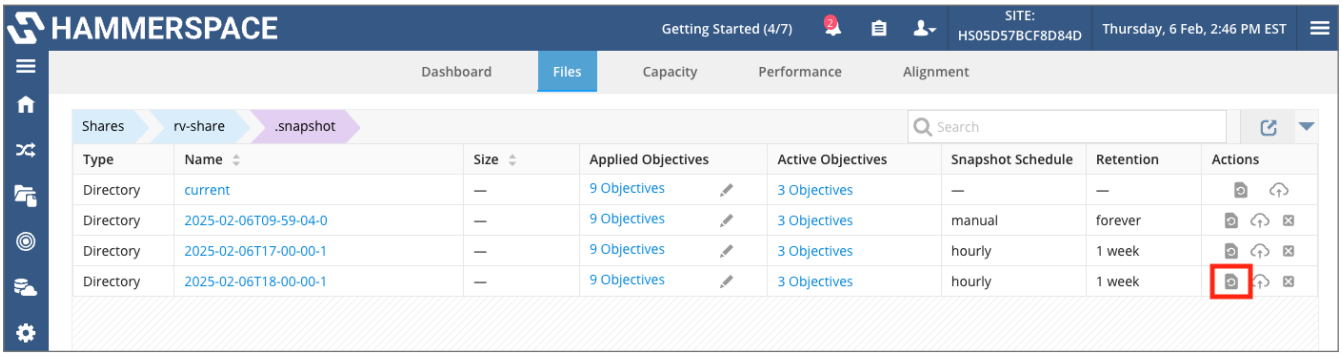


Figure 79. Location of the Restore icon

4. In the Restore Snapshot popup, select to restore files to their original location, or specify a different location within the share. You can also select to overwrite the contents of the restoration target directory or to create a new target directory for the restored data.
5. Click **Restore** and you will see a notification in the GUI that the restore operation is in progress. You can click the **Tasks** icon to confirm that the restore operation is successful.

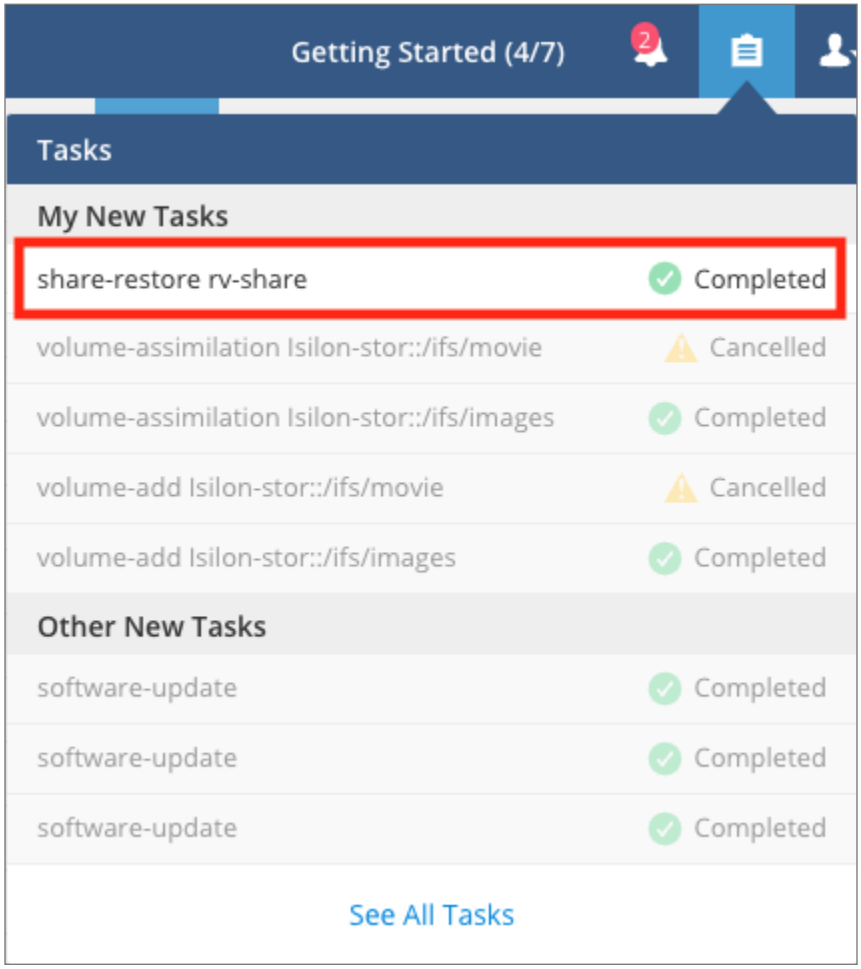


Figure 80. Viewing completed task status

8.13.7. Restoring Individual Files Using the CLI

You can restore files from a snapshot of a share.

© 2026 - Hammerspace

Restore files from a share snapshot.

```
share-snapshot-restore --share-name test --filename /VM2/.vmdk* --snapshot-name test-2025-01-25T16-59-27.371Z-0
```

8.13.8. Saved-Snapshot Location

Share snapshots are available in a virtual `.snapshot` folder, and by using Windows Previous Versions. The `.snapshot` folder is not visible over SMB data access. The `.snapshot` folder is available in every directory in the share, representing the data from that directory and its contents.

8.13.9. Snapshot Schedule

The Product provides pre-created schedules for hourly, daily, and weekly snapshots. In addition, administrators can define custom schedules to fit their precise needs.

Example Snapshot Schedules

- An hourly snapshot schedule with a retention of seven days.
- A daily snapshot schedule with a retention of seven years (to meet compliance rules).
- A weekly snapshot schedule with a retention of 12 weeks.

Creating Snapshot Schedule in the GUI

You can set a snapshot schedule for a share, folder, or file. When the snapshot is created for a share or folder, the snapshot captures everything in the share or folder. This procedure shows how to set a snapshot schedule.

1. Open the GUI and select **Data** from the left side panel. By default you will see information in the **Shares** tab.

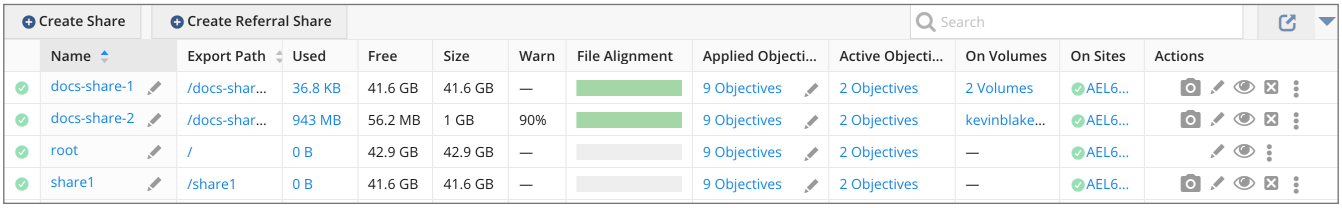


Figure 81. Editing a share

2. Identify the share to snapshot and click the **Edit** icon.
3. In the Edit Share popup, select the **Snapshot Schedules** tab.
4. Click **New Schedule**.

© 2026 - Hammerspace

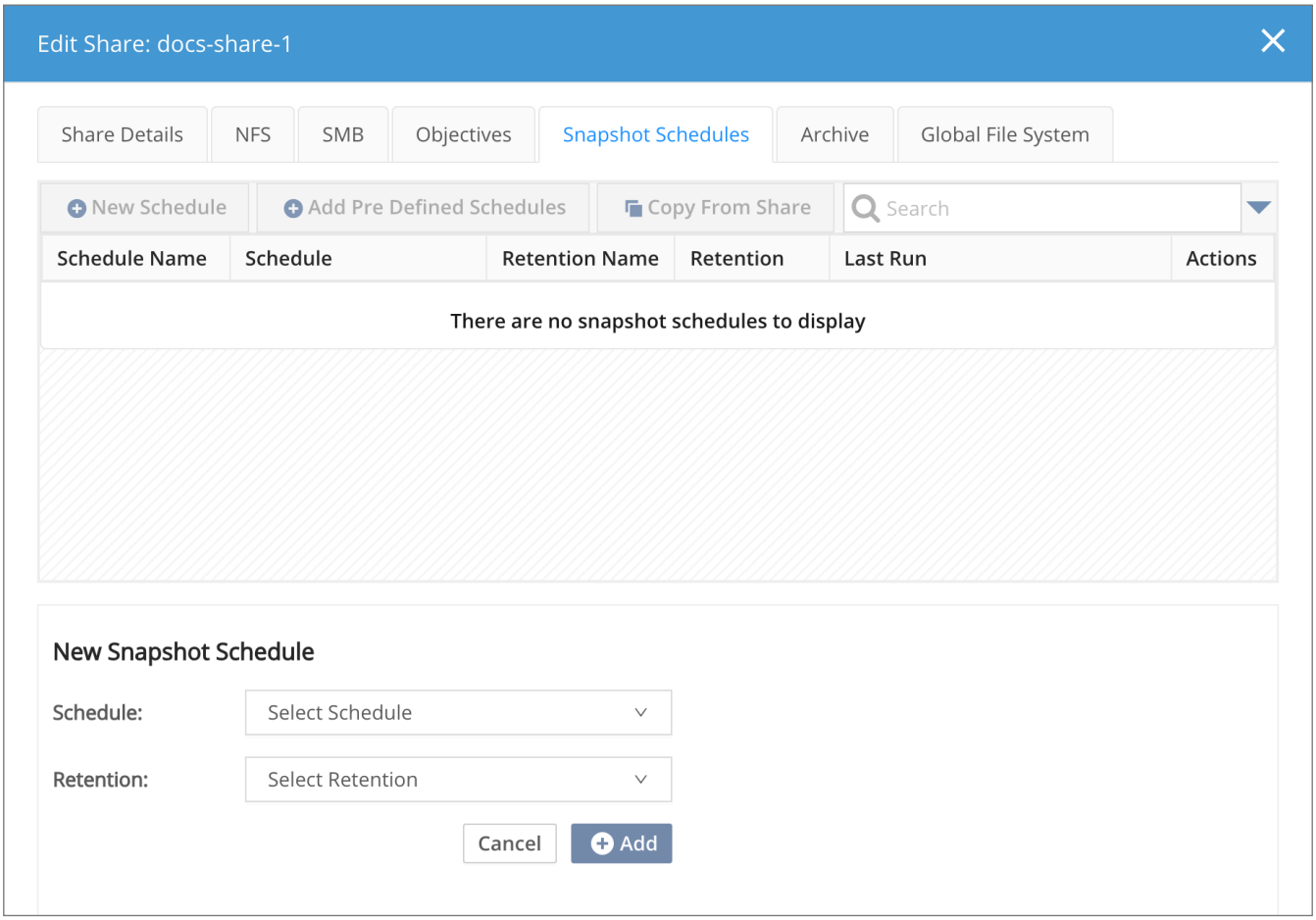


Figure 82. Adding a new snapshot schedule

5. In the **New Snapshot Schedule** section of the popup, select a frequency schedule and a retention length for the snapshots, then click **Add**. You will see the new snapshot schedule in the Snapshot Schedules tab of the popup.
6. Click **Close** to finish.

Creating Snapshots Using the CLI

You can create a snapshot of a share instantly:

Admin CLI: Create instant snapshot

```
share-snapshot-create --share-name test --now
```

8.14. Using Objectives

Default objectives confer a baseline level of protection and are applied to each share when it is created. In addition to, or instead of, the default objectives, other protection options can be selected when shares are created.

The *create-share* workflow makes selecting additional protection easy. Following the creation of a share, objectives for the share can be changed at any time by modifying the Applied Objectives in **Data › Shares** >

© 2026 - Hammerspace

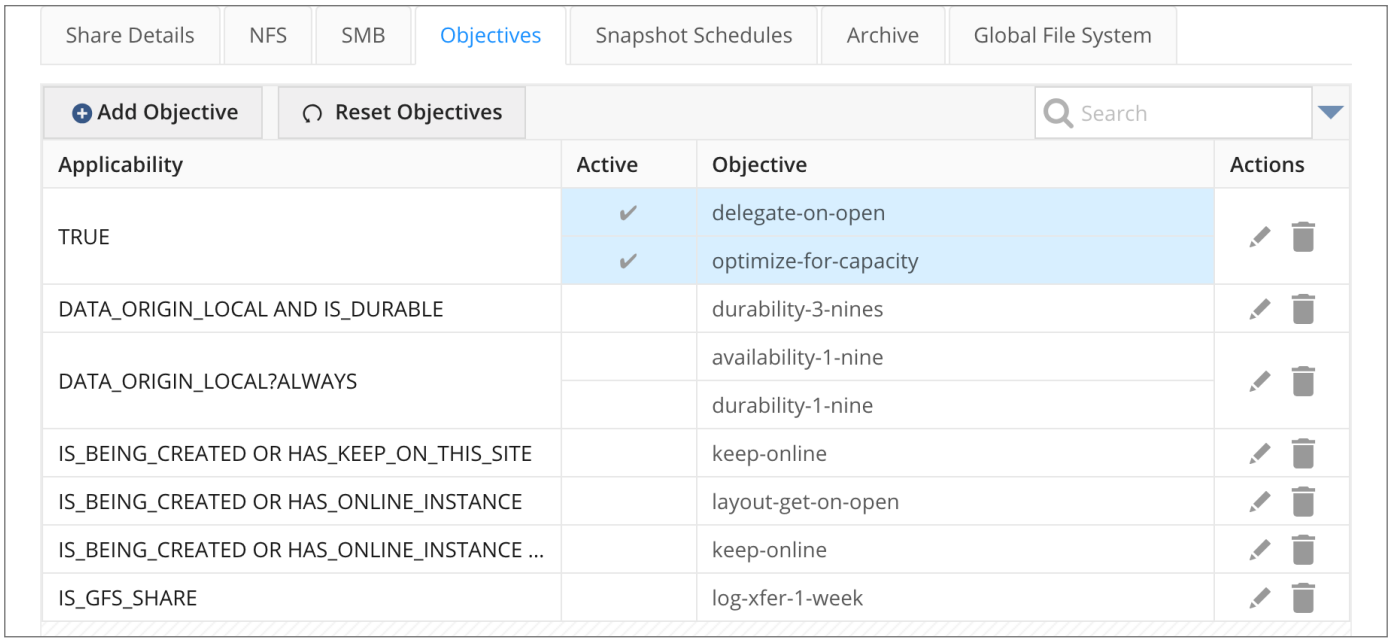
<share-name>.

If the default objectives are not sufficient, a predefined objective might offer the required additional protection. If a predefined objective is still not adequate, any number of custom and advanced objectives can be created.

Furthermore, any number of objectives can be applied to a share or a directory to meet the different demands of data placement.

8.14.1. Default Objectives

The Product applies the following default objectives to new shares. You can click on the pencil icon to modify the defaults:



Applicability	Active	Objective	Actions
TRUE	✔	delegate-on-open	
	✔	optimize-for-capacity	
DATA_ORIGIN_LOCAL AND IS_DURABLE		durability-3-nines	
DATA_ORIGIN_LOCAL?ALWAYS		availability-1-nine	
		durability-1-nine	
IS_BEING_CREATED OR HAS_KEEP_ON_THIS_SITE		keep-online	
IS_BEING_CREATED OR HAS_ONLINE_INSTANCE		layout-get-on-open	
IS_BEING_CREATED OR HAS_ONLINE_INSTANCE ...		keep-online	
IS_GFS_SHARE		log-xfer-1-week	

Figure 83. Default objectives following installation

8.14.2. Predefined Objectives

Predefined objectives are those that are available and ready to use on the system. You can view the full list of predefined objectives in the Admin CLI using `objective-list` command. The following bullets describe the predefined objectives.

- **Access-based-enumeration:** Access-based enumeration limits which files and folders users see when they list a directory. It filters directory listings and does not control access. Requires that the do-not-cache objective is also applied; Hammerspace does not apply it automatically. On an NFS client that several users share, such as a Linux or macOS client, per-user filtering is not reliable; see Allowing Access-Based Enumeration.
- **ACL-compatibility-mode-non-posix:** When applied, disables UNIX/POSIX mode class constraints; the mode behaves like an ACL. Commonly used with Windows or multi-protocol (Windows & UNIX) environments.
- **Availability-#-nines:** The four availability objectives allow the admin user to specify how many nines of availability the data should have. Each storage volume is assigned an availability value. Data will be placed either on a storage volume that meets the availability number, or multiple backend copies of the file may

© 2026 - Hammerspace

get created to meet the requested number of nines.

- **Block-create, -open, -read, -write:** Blocks an operation. Block objectives can cause situations where the client “may” appear hung because no progress is being made. Typically used in situations where an external process such as virus-scanning must be executed before access is given.
- **Confine-to-object-volumes**: Confines the data to object storage volumes.
- **Confine-to-shared-object-volumes**: Confines the data to shared object storage volumes.
- **Default-objective:** This ensures that data can be placed on any volume in the system.
- **Delegate-on-open:** Controls the delegate behavior for NFSv4.2 protocol. Leaving it on generally improves performance when files are opened and written to. If using NFSv3 or SMB, this indirectly impacts the behavior because the DSX node will be holding the layout instead of the client.
- **Deny-create / delete / open / read / write:** Denies the operation, returning an error code to the application.
- **Do-not-cache:** Do not cache metadata or data on clients. Requires the client to support the NFSv4.2 FATTR4_UNCACHEABLE attribute.
- **Do-not-move:** This allows the administrator to simulate what files are impacted by an objective but will not actually move the file.
- **Durability-#-nines:** There are four durability objectives. This allows the admin to specify how many nines of durability the data should have. Each storage volume is assigned a durability value. Data will be placed either on a storage volume that meets the Durability number (minimum is 1) or multiple backend copies of the file may get created to meet the requested number of nines.
- **Exclude-from-object-volumes:** Avoid placing data on volumes of type "object".
- **Exclude-from-shared-object-volumes:** Avoid placing data on volumes of type "shared object".
- **Exclude-from-virus-scanners:** This will prevent data from being sent to iCAP-configured virus scanners.
- **Exclude-from-*<volume name>*:** This will prevent data from being placed on a specific storage system, volume name, or tier.
- **Keep-online:** This will place data on any file-based volume, including DSX-based volumes and third-party NAS volumes in the system. Keep-online is also automatically assigned to data that is modified and/or created.
- **Layout-deny-on-error:** Controls the layout behavior when errors occur on the file storage, return EIO when encountering storage errors. Typical workloads do not need to modify this option.
- **Layout-get-on-open:** Controls the layout behavior for NFSv4.2 protocol. Leaving it on generally improves the performance when files are opened and then written to. If using NFSv3 or SMB, this in-directly impacts the behavior as the DSX node will be holding the layout vs. the client.
- **Log-xfer-*:** Used with the global file system. Preserves a version of the file when site ownership changes. Retention example: log-xfer-1-day. This will retain the version for 1-day. If a share snapshot is taken, the version will be retained, as part of the snapshot, for as long as the snapshot exists.
- **No-atime:** Removes the need to record access time.
- **Optimize-for-capacity:** Do not maintain instances on local storage when no objectives require a local copy. The file data will be in the cloud, on object storage or on other sites.
- **Performance-fast / -medium / -slow:** Pre-defined tiers for placing data on different performance tiers.

© 2026 - Hammerspace

- **Place-on-object-volumes:** Place data on volumes of type "object".
- **Place-on-shared-object-volumes:** Place data on volumes of type "shared object".
- **Place-on-virus-scanners:** Align data to iCAP-configured virus scanners. It is recommended to use the objective virus-scan-operation for a complete workflow.
- **Undelete-*:** Turns on the undelete feature. Undelete saves the file for a pre-determined amount of time (driven by the objective, 1 hour, 1 day, 1 week or 1 month) in the directory, `.snapshot/current`. Note that undelete files are also retained as part of share snapshots and will be retained for as long as the snapshot is retained.
- **Versioning-*:** Enables file versioning. File versions are created after no activity have happened on a file for 4 minutes. Versioned files are retained for the pre-determined amount specified as part of the objective name or for as long as the snapshot that includes the file is retained.
- **Virus-scan-operation**: Scan data using available virus scanners.
- **Worm-operation**: Enables WORM functionality when used together with a WORM expiration date.

Setting a `worm_expire_date` relative to creation time (for example, `expression(CREATE_TIME + 10 minutes)`) together with the `worm-operation` objective works as expected over NFSv4.2. Over NFSv3 and SMB, file creation and the first write are not atomic, so a newly created file can be write-denied before the application writes its content. Use NFSv4.2 mounts for workflows that create files under an active WORM objective.

Deleting a share removes all data in that share, including files protected by `worm-operation` or an undelete objective. Share deletion does not honor WORM expiration or undelete retention. See File Versioning.

On global file system (GFS) shares, the system automatically applies the `log-xfer-1-week` objective — a removable, share-level objective with applicability `IS_GFS_SHARE` — to new GFS shares, and to existing GFS shares the first time the storage manager starts after an upgrade to 5.2. This preserves a version of files when site ownership changes, preventing replication data loss. Administrators may remove it.

8.14.3. Editing an Objective

Objectives apply to the data on a share on the Hammerspace site.

1. Select **Objectives** from the left navigation panel. The Objectives tab opens by default.
2. Click the **Edit** icon in the same row as the objective you want to edit.

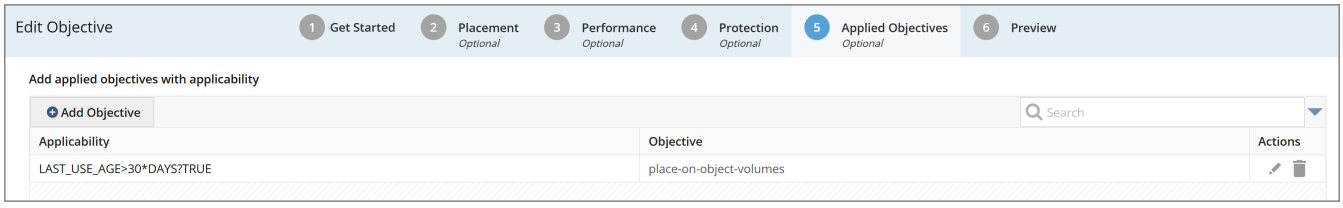


Figure 84. Editing an objective

3. Walk through the edit Objectives wizard to make changes.

© 2026 - Hammerspace

4. Click **Save Objective**. The system will automatically detect the changes and align data to the updated rules.

8.14.4. Advanced Objectives

Advanced objectives use logic operators such as **AND, OR,** and **>=** against values such as filename, extension, size, access age, and more to create powerful matching expressions.

When writing advanced expressions using either the built-in editor or the Admin CLI command `objective-create`, refrain from writing extensive and complicated multi-line **IF-THEN-ELSE** to match the desired formula. It can be difficult to maintain objectives over time when they are too complicated.

The simpler approach is to write several one-line expressions and then apply all of them to the file/directory or share. Hammerscript will evaluate all expressions applied to a file and will automatically manage any potential conflicts.

There is no performance penalty for IO if files or directories have complicated expressions.

Common Operators

- `&&` (AND, true if all arguments are true)
- `||` (OR, true if any argument is true)
- `!` (NOT, false if the argument is true)
- `&&!` (AND NOT, true if first argument is true and second is argument false)
- `||!` (OR NOT, true if first argument is true or second is argument false)
- `==` or `=` (equal to)
- `!=` or `<>` (not equal to)
- `>` (greater than)
- `>=` (greater than or equal to)
- `<`(less than)
- `⇐` (less than or equal to)
- `&` (Concatenation)

Common Functions for Advanced Expressions

The expression operators are case insensitive; the expression names below, and in the examples that follow, use upper case for readability. Various units can be expressed in easy-to-use English, such as MINUTES, HOURS, WEEKS, and so forth, for time. The same applies for capacity where both GB (gigabytes) and GiB (gibibytes) variants are available.

- `ACCESS_AGE`: Time since the most recent READ of the file. Reads could theoretically be cached on the client and not updated in Anvil metadata. Also see LAST_USE_AGE.
- `ACCESS_TIME`: POSIX atime

© 2026 - Hammerspace

- `ACTIVITY`: current average IOPS
- `CHANGE_TIME`: POSIX ctime
- `CREATE_TIME`: Create time. Commonly used with SMB protocol but also set when files are created via NFS.
- `LAST_USE_AGE`: Includes reads, writes AND opens
- `FNMATCH` (<TEXT>, NAME|PATH): Match a string with input such as PATH or NAME



The FNMATCH operand is not an available option in the Management GUI dropdown. Instead, by selecting FILE_NAME or PATH, the application adds the FNMATCH operand to the expression in the editor. This is by design to enable admins to use FNMATCH using more familiar parameters.

- `MATCH_EXTENSION`(<TEXT>, NAME): Match the extension of the filename
- `OWNER_GROUP`: owner group ID
- `MODIFY_TIME`: POSIX mtime
- `NAME`: file name
- `NOW`: current date and time
- `OWNER`: owner ID
- `PARENT_SHARE`: Share name the file or directory belongs to
- `PATH`: directory path within the share
- `SIZE`: file size
- `SPACE_USED`: actual space consumed by file
- `IS_LIVE`: Is the file the live version or not. True means the file is in the live tree and not part of a snapshot
- `IS_OPEN`: Does the file have an outstanding layout, i.e. has someone opened the file
- `IS_SNAP`: is the file in a snapshot?
- `IS_UNDELETE`: Is the file an undelete file? Only true of undelete files in special `.snapshot/CURRENT` directory
- `VERSION_NEWEST`: most recent snapshot in which this file exists
- `VERSION_OLDEST`: oldest snapshot in which this file exists
- `VERSION`: snapshot version where this file resides
- `VERSION_AGE`: How old is the file compared to the live tree. Age of 0 seconds indicates it is the live version of the file.
- `VERSIONS_TOTAL`: Number of snapshots the file is a part of.

The Web UI "define applicability" builder uses operand names that differ from the expression-language function names. Use this mapping:

© 2026 - Hammerspace

Web UI operand	Hammerscript equivalent
PATH	`FNMATCH("<pattern>", PATH)`
FILE_NAME	`FNMATCH("<pattern>", NAME)`
GROUP	`OWNER_GROUP`

8.14.5. Examples of Advanced Objectives

The following example objectives are created using the `objective-create` command in the Admin CLI:

- When creating the advanced expressions, the `\` character must be used before every `[SPACE]`. When using the **FNMATCH** function, `\` must be used before `()` characters.

Every expression has a destination for data and is expressed using `SLO(<name of destination>)`. The destination can be a configured tier, an added storage system, an added storage volume, a volume group, and so forth.



The expression operators are case insensitive; upper case is used for readability in the examples in this section.

Tiering of Data from File Storage to Archive (Cloud/Object)

This example will match any file in the live tree and move files to cloud/object storage when it has not been used for more than four weeks.

Using the Admin CLI

Admin CLI

Command:

```
objective-create --name "Old files to Object Storage" --applied-objective place-on-object-
volumes,IS_LIVE&&LAST_USE_AGE>4WEEKS
```

Move Snapshots to Cloud/Object Storage

This objective will move snapshots to cloud/object storage. This will free up space on the file storage volumes and increase the resiliency of the snapshot data.

It is recommended to combine this objective with the next example objective to keep the most recent snapshot on file volumes and cloud/object storage for complete protection of the snapshot data.

Admin CLI

```
objective-create --name "Snapshots to cloud" --applied-objective place-on-object-volumes,VERSION>=2
```


© 2026 - Hammerspace

Keep Most Recent Snapshot on Original Location

Only modified files in a snapshot take actual space on file storage and the most efficient method to store that file is on the original file storage volume. However, it is also important for resiliency reasons to make sure that the snapshot is available outside of the original volume and this objective will create a copy of that file on cloud/object storage.

Admin CLI

```
objective-create --name "Keep most recent snapshot on File" --applied-objective keep-online,VERSION=2
```

Keeping Live Tree on File Volumes

This objective keeps the live tree on file volumes except file clones (also known as file snapshots in the GUI). The `IS_LIVE` expression makes sure that this objective does not apply to any data in .snapshot directory.

Admin CLI

```
objective-create --name "Keep live tree on File" --applied-objective keep-online,
IS_LIVE&&!FNMATCH("/.fsnapshot/",PATH)
```

Keeping File Clones on Cloud/Object Storage

This objective will move file clones (also known as file snapshots in the GUI) to cloud/object volumes. Files that are cloned using the UI or CLI are placed in a sub-directory called `.fsnapshot/<FILENAME>/DATE-TIME`.

Admin CLI

```
objective-create --name "Snapshots in cloud" --expression IF\ FNMATCH\("/.fsnapshot/",PATH\)\ THEN\
{SLO('place-on-object-volumes')}
```

Placing PDF Files on Cloud/Object Storage

This objective will move files with the `.pdf` extension to cloud/object storage. This move will happen after five minutes after the last use of the file, as per the objective. If the file is opened, then it will be moved back to a file volume and within five minutes of last being used, it will again be moved. The subsequent move will only upload data if the file was modified, since a copy of the file is already in the cloud/object storage.

Admin CLI

```
objective-create --name "Place PDF files in cloud" --expression IF\
MATCH_EXTENSION\("pdf",NAME\)&&LAST_USE_AGE>5*MINUTES\ THEN\ {SLO('place-on-object-volumes')}
```

Moving Data in the Archive Directory to Cloud/Object Storage

This objective will move any data placed in or below a directory called archive to the cloud. It is recommended to apply this objective to the root of a share, as the archive directory can then be placed anywhere in the share, and this objective will apply.

© 2026 - Hammerspace

Admin CLI

```
smart-objective-create --name "Archive directory in cloud" --expression IF\ FNMATCH\(“/archive/”,PATH\)\
THEN\ {SLO('place-on-object-volumes')}
```

Moving Undelete Files to Cloud or Object Storage

The Hammerspace file system supports the ability to protect data against deletion using an *undelete* objective. This objective can be applied to any file or directory in the file system, and it is available in the UI. This example objective moves the undelete copy of the file to the cloud/object storage to free up capacity on primary storage.

Admin CLI: Advanced objective example

```
objective-create --name "Undelete files in cloud" --applied-objective place-on-object-volumes,IS_UNDELETE
```


© 2026 - Hammerspace

Chapter 9. Storage Management

Hammerspace lets administrators add, monitor, and remove storage capacity across NAS and object storage systems without disrupting client access to data. Using the GUI, you can add new storage systems and volumes, mix storage types to meet changing performance and capacity needs, track available and used capacity to plan ahead, and non-disruptively decommission volumes when they are no longer needed.

This section includes the following subsections:

- Adding Storage Capacity
- Adding Storage to DSX
- Capacity Management
- Removing Storage

9.1. Adding Storage Capacity

Using the GUI, you can easily add your stored data to Hammerspace by adding storage systems and volumes on which it is stored. Add a new NAS or Object storage system, or add volumes from storage systems that are already recognized by the cluster.

9.1.1. Adding a Storage System

Complete the following steps to add a data storage system to Hammerspace:

1. In the GUI, go to the **Infrastructure** page and click the **Storage Systems** tab.
2. Click **Add Storage System**. The Add Storage System wizard opens.
3. Provide the requested details:
    a. Provide a name for the storage system.
    b. Select NAS for the storage type. It might already be selected by default.
    c. Provide the storage type and then the access details.
4. Click **Add System**.

9.1.2. Adding a Storage Volume

Complete the following steps to add a storage volume:

1. In the GUI, go to the **Infrastructure** page. The **Volumes** tab is visible by default.
2. In the **Volumes** tab, click **Add Volume**. The Add Volumes wizard opens.
3. In the **Selections** tab, select the storage system that contains the volumes you wish to add. Check the box next to the storage system and click **Next Step** to advance. Optionally, you can click on the name of the system and select specific volumes to permit in the following step.
4. In the **Data** tab, specify access permission and whether to assimilate the existing data on the volume. This

© 2026 - Hammerspace

action will take some time and is not reversible. Make the appropriate selections and click **Next Step**.



If the data on the volume has already been assimilated on another Anvil, then if you assimilate or add it in read/write mode, previous metadata will be deleted, and the data will no longer be accessible.

5. In the **Capacity** tab, you can adjust capacities and mobility thresholds for each of the volumes to be added. If capacity is already set on the volume(s) or system, you will not be able to change these settings. Click **Next Step**.
6. In the **Capability** tab, you can leave the Auto settings or adjust the durability and availability requirements for the volume(s). Changing the Auto setting to a specific *number-of-nines* allows you to then add those settings to an objective and/or skip the performance test of the settings.
7. In the **IP** tab, view the discovered IPs for the volume and edit additional and excluded IPs as needed. Click **Next Step**.
8. In the **Review & Add** tab, review the configuration. If the configuration settings are correct, click **Add Volume**.

The GUI will take you back to the **Infrastructure** › **Volumes** tab while the system performs the operation. You can view the status of the operation by clicking on the **Tasks** tab at the top of the page.

© 2026 - Hammerspace

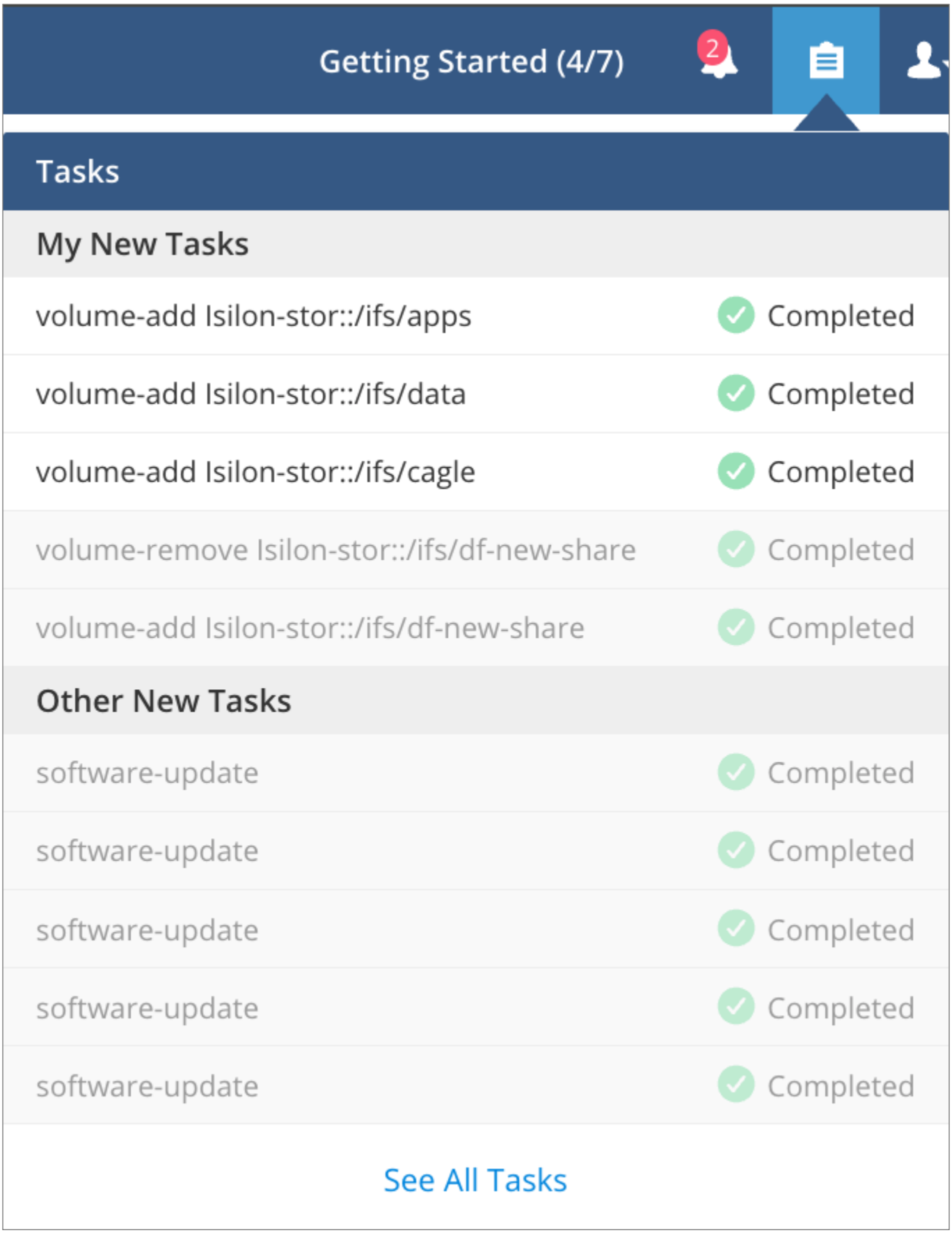


Figure 85. View operation status in the Tasks dropdown

© 2026 - Hammerspace

9.1.3. Resizing a 3rd Party NAS Volume

Resizing a 3rd party NAS volume is as simple as resizing the NAS volume as needed, and either manually refreshing or waiting for the system to automatically refresh the capacity information.

9.2. Adding Storage to DSX

You can add storage to a DSX node after the initial installation is complete. Each of these steps are detailed in the sections that follow:

1. Add storage [External task]
2. Refresh the configuration [Hammerspace task]
3. Create a logical volume [Hammerspace task]
4. Register the new volume [Hammerspace task]

9.2.1. Part 1: Adding Storage

This step is done outside of the Hammerspace interface. This example has added storage using the VMware vCenter interface. The DSX node automatically detects this storage and adds the new block devices at the operating system level. Additional steps, described in Steps 2 and 3 below, are required before the new storage is usable.

It is also possible to use a Fibre Channel storage area network (FC SAN) or an Internet Small Computer Systems Interface storage area network (iSCSI SAN) to add new storage; however, multi-pathing options are not available. In a VMware environment, it is recommended that the hypervisor present the storage to the virtual machine so that it is configured to support multi-pathing.

9.2.2. Part 2: Refreshing the DSX Configuration

1. On Anvil, log in as administrator and execute `node-refresh` on the DSX where you added new storage.

   Admin CLI

   Command:

   ```
   node-refresh --name dsx.example.com --rescan
   ```

   Expected output:

   ```
   Name:                       dsx.example.com
   Type:                       Product
   Product node type:          DSX
   Internal ID:                1073741832
   ID:                         406901cd-c185-5a5d-93b1-8ad3e91a5809
   HW state:                   OK
   Node state:                 MANAGED
   ```


© 2026 - Hammerspace

```
Node mode:                    ONLINE
<Additional output removed for brevity>
```

- `node-list`: Identify the correct DSX.
- `node-refresh` options:
  - `--id`: ID of the DSX node.
  - `--rescan`: Forces the DSX to discover new block devices. Does not affect existing, already discovered devices. This is a required option when adding new storage.

9.2.3. Part 3: Creating a New Logical Volume

1. Identify the added block device(s) first. In this example, the device added is `/dev/sdb`:

   Admin CLI

   Command:

   ```
   drive-list --node-name dsx.example.com
   ```

   Expected output:

   ```
   total 6
   Type:                   Hdd
   ID:                     11967fd2-1caa-4f3e-b81e-59d8e6b62a02
   Name:                   /dev/dm-1
   State:                  OK

   Type:                   Hdd
   ID:                     1dc3944b-4b10-40f7-8155-1eee83d73852
   Name:                   /dev/sda
   State:                  OK

   Type:                   Hdd
   ID:                     3f827138-aa01-4db2-bd40-404c69b68669
   Name:                   /dev/sdc
   State:                  OK

   Type:                   Hdd
   ID:                     6c301217-cf61-45f5-918a-517bd21db199
   Name:                   /dev/sdb
   State:                  OK

   Type:                   Hdd
   ID:                     b38b67e2-6634-44a1-a7a5-372223a27506
   Name:                   /dev/dm-0
   State:                  OK
   ```


© 2026 - Hammerspace

```
Type:                   Hdd
ID:                     bf256e29-b1a6-4b78-a4ce-199dc8ca43da
Name:                   /dev/dm-2
State:                  OK
```

2. Create a logical volume using the new device path with the `--device-path` option.

   Admin CLI

   Command:

   ```
   logical-volume-create --node-name dsx.example.com --device-path /dev/sdb
   ```

   Expected output:

   ```
   Type:                   Logical Volume
   ID:                     42a939fe-7ee7-4208-b950-c781077fc88c
   Name:                   /hsvol0
   Exported path:          /hsvol0
   Addresses:
                           [IP: 192.0.2.21/20, Port: 3049, NetId: tcp, NodeNum: 0]
   Node:                   dsx.example.com
   Usage:                  DS
   Reserved:               false
   Filesystem type:        XFS
   Capacity:               [Total: 21.4GB]
   Hardware:
                           [Type: Hdd, ID: fd037f43-3880-475b-a8bc-6852a0d8c817, Name: /dev/sdb, State:
   OK]
   ```

9.2.4. Part 4: Registering the New Volume

This step creates a new file system on the logical volume and adds the new volume into the available storage configuration. Depending on how the system objectives are configured, it is possible that data will be placed on the new volume as soon as this step is complete.



The logical-volume-name is found in the output from the previous command.

Add the volume

Command:

```
volume-add --logical-volume-name /hsvol0 --node-name dsx.example.com
```

Expected output:

```
ID:                     eddc2054-ddba-4315-b15c-d0afd2a90eaa
```


© 2026 - Hammerspace

```
Name:                   dsx.example.com::/hsvol0
Internal ID:            15
Discovered addresses:
                        [IP: 192.0.2.21/20, Port: 3049, NetId: tcp, NodeNum: 0]
Effective IPs:
                        [IP: 192.0.2.21/20, Port: 3049, NetId: tcp, NodeNum: 0]
Path:                   /hsvol0
State:                  OK
Access type:            Read Write
Node:                   dsx.example.com
Oper state:             Up
Admin state:            Up
Capacity:               [Total: 21.4GB, Used: 225MB (1%), Free: 21.2GB]
Capabilities:           Max read IOPs:            35782
                        Max write IOPs:           772
                        Min read latency:         0.144167064 milliseconds
                        Min write latency:        3.6410911550000002 milliseconds
                        Max read bandwidth:       2255200 KB/s
                        Max write bandwidth:      148365 KB/s
                        NFS read request size:    1.0 MiB
                        NFS write request size:   1.0 MiB
                        Tolerable read latency:   100000 milliseconds
                        Tolerable write latency:  100000 milliseconds
                        High threshold:           75%
                        Low threshold:            60%
                        Availability:             99%
                        Availability (effective): 99%
                        Availability drop:        Disabled
                        Durability:               99.9%
                        Durability (effective):   99.9%
                        Clone status:             Functional
                        Online delay:             Online
Created:                2026-09-22 00:35:03 UTC
Modified:               2026-09-22 00:35:03 UTC
Root file handle:       0100060064472081d8e54c58b65cdb219cdccf3f
Locations:
                        [Type: Node, ID: 406901cd-c185-5a5d-93b1-8ad3e91a5809, Name: dsx.example.com]
                        [Type: Storage Volume, ID: eddc2054-ddba-4315-b15c-d0afd2a90eaa, Name:
dsx.example.com::/hsvol0]
                        [Type: Volume Group, ID: c368268b-6a9b-4b40-99ef-5cc6f59b55ff, Name: all]
Random writes:          Enabled
```

The new volume will be viewable in the UI after a short wait as the system refreshes.

9.3. Capacity Management

Administrators can increase performance and capacity of any storage type by adding storage capacity of the same type, or of any other type that best meets an organization's changing needs. Administrators can easily view available resources, including total available storage capacity, total used capacity, individual volumes

© 2026 - Hammerspace

available, and used capacities. This makes it easier to estimate when capacity will run out, and to proactively take action to address capacity issues.

9.3.1. Viewing Total Capacity

You can view the capacity of your storage resources.

1. To view total capacity, open the GUI and go to the **Dashboard**.
2. In the **Capacity** panel, **Storage** section, you can view the number of storage systems, volumes, shares, and files. The **NAS Total** section shows how much of the total capacity is used and free. The **NAS Growth** section shows the history of capacity usage for the past 7 days.

9.3.2. Addressing Capacity Issues

Capacity issues can be quickly addressed by adding storage. Adding new storage and moving data to different storage volumes are non-disruptive operations.

9.4. Removing Storage

Removing storage volumes from the environment is a non-disruptive task and can be done at any time. It is important to follow these steps to ensure continued access to the data.

9.4.1. Part 1: Identifying the Volume

1. Go to **Infrastructure** › **Volumes** tab in the GUI. Locate the volume you want to decommission and press the **Decommission** icon under the **Actions** column.

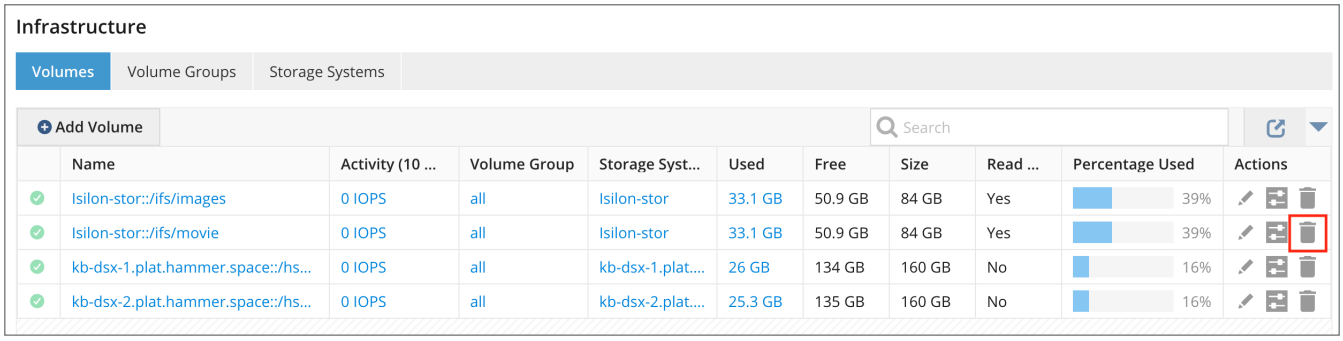


Figure 86. Volume decommission icon

2. In the **Decommission Volume** dialog, set the options. Hammerspace moves the volume's data to other volumes as your objectives direct (objectives are the rules that decide where each file's data may be placed).
   - **Replace with** (optional, not selected by default): select one or more volumes, volume groups, or nodes to take this volume's place in the objectives and volume groups that name it. With the references replaced, a rule that placed data on this volume places it on the replacement, and a rule that excluded data from this volume excludes it from the replacement.

     You can make the same replacement from the Admin CLI. Run `volume-decommission` (or, to

© 2026 - Hammerspace

decommission and remove in one step, `volume-remove`) with `--replace-with`, for example:

```
volume-decommission --name <VOLUME-NAME> --replace-with volume-group:<VOLUME-GROUP-NAME>
```

For an object volume, use `object-volume-decommission` or `object-volume-remove`. Each replacement is `<type>:<name>`, where the type is `volume`, `object-volume`, `volume-group`, or `node`; separate several with commas. The Admin CLI makes the replacement before the decommission starts. See the *Hammerspace Command Line Reference*.

Whichever method you use, review your objectives and volume groups after the decommission starts to confirm that they name the replacement, and edit any that do not.

- **Remove volume after decommission** (selected by default): Hammerspace removes the volume as soon as decommissioning finishes. If you clear it, remove the volume yourself afterward, as described in the last step of this procedure.
- **Force delete, even if it will impact access to data** (object volumes only): allows the decommission to go ahead even when a global file system (GFS) depends on the object volume.

3. Type `Yes` in the **Type Yes to continue** field (it is not case-sensitive) and click **Decommission**. If **Replace with** is selected, **Decommission** stays unavailable until you choose at least one replacement.

   Hammerspace moves the data off the volume without interrupting client access. Files move only to volumes that their objectives allow. If those volumes do not have enough free capacity, the decommission does not complete until capacity becomes available. Adding capacity helps only if the files' objectives allow data on the new volume; for example, adding an object volume helps only if the objectives allow data on object storage.

4. Follow progress in the **Tasks** view. While the decommission runs, the volume's **Actions** column in the **Volumes** view shows a **Cancel** button. When **Remove volume after decommission** is selected, the task is named `volume-remove` (`object-volume-remove` for an object volume).
5. If the volume was not removed as part of the decommission, remove it now. In the **Volumes** view, the volume's **Actions** column shows a **Remove** button once decommissioning is complete. Click **Remove**, then click **Remove** in the **Remove Volume** dialog. A decommissioned volume cannot be used to store data, nor can it be added back until it has been removed.

9.4.2. Part 2: Deleting the Logical Volume (CLI Only)

A logical volume is the Hammerspace object that represents a storage device on a node; the storage volume you removed in Part 1 was built on it. Deleting the logical volume requires the Admin CLI, and it is required before you remove the physical storage from the node.



A logical volume that a storage volume still uses cannot be deleted; the command fails with an error naming the storage volume that uses it. Decommission and remove the storage volume (Part 1) first.

1. Log in to the Admin CLI and list the node's logical volumes:

© 2026 - Hammerspace

```
logical-volume-list --node-name <NODE-NAME>
```

Find the logical volume that backed the storage volume you removed (the **Name** and **Exported path** columns help you match it), and note its **ID**.

2. Delete the logical volume by its ID:

```
logical-volume-delete --id <LOGICAL-VOLUME-ID>
```

If the node cannot finish releasing the device, the logical volume is still deleted, but the command reports a warning such as `The logical volume is busy and could not be unmounted. A node reboot is required to complete cleanup.` Reboot the node before you reuse or physically remove the device.

© 2026 - Hammerspace

Chapter 10. User Management

You can control administrative user access to the Hammerspace product (Product) Management GUI, Admin CLI, and REST API through user management. This includes creating and editing local management users, assigning roles, applying login policies and network restrictions, configuring enterprise sign-on, and managing account auto-locking.

This section includes the following subsections:

- Account Auto Locking
- Adding Local Management User Using the GUI
- Applying Network Restrictions
- Clearing Network Settings
- Deleting a User
- Editing a User
- Enterprise Sign On
- Local Management Users
- Login Policies
- User Access to the Management Interfaces
- User Roles

10.1. Account Auto Locking

The Product can be configured to automatically lock an account after a set number of consecutive login failures. By default, the account lockout is disabled. The lockout functionality only applies to local users.

Admin CLI: Enable account lockout after 5 failures

Command:

```
login-policy-config --lock-after-failures 5
```

Expected output:

```
Allowed networks:            All
Lock after login failures:   5
```

The lock-failure interval and the lockout time can also be configured.

© 2026 - Hammerspace

10.2. Adding Local Management User Using the GUI

Local management users are not required when using the enterprise sign-on feature.

Complete the following steps in the Management GUI to add a local management user:

1. Select **Administration › Users**.
2. Click **Create User**.

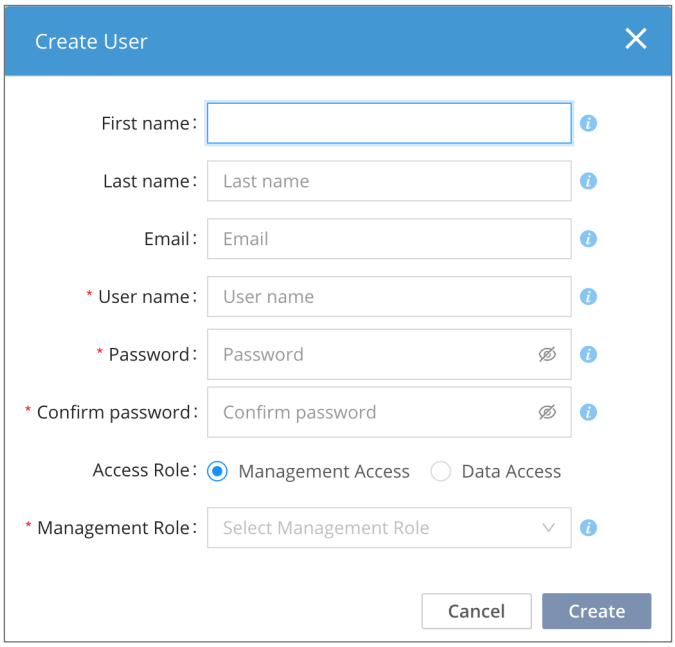


Figure 87. Entering information for a new management user

3. Enter the following information for the user:

Table 16. New user details

Information required for new, local management user	
First name	First name of the user
Last name	Last name of the user
Email	The Product will send alerts and notifications to this address. To enable email notifications, the SMTP server must be configured using the CLI.
User name	Username
Password	Passwords must be 8-20 characters long and contain at least one of each: lowercase letter, uppercase letter, digit, and non-alpha numeric.
Confirm password	Enter the password again.
Access Role	Select Management Access.

© 2026 - Hammerspace

Management Role	Select the desired role.

4. Click **Create**.

10.3. Applying Network Restrictions

The Product can be configured to only allow management access from a limited set of networks. For multiple networks, use a space-separated list.

Admin CLI: Only allow access from 10.50.100.0/24 network

Command:

```
login-policy-config --allowed-networks 10.50.100.0/24
```

Expected output:

```
Allowed networks: 10.50.100.0/24
Lock after login failures: Disabled
```

10.4. Clearing Network Settings

Admin CLI: Clear allowed networks settings

Command:

```
login-policy-config --allowed-networks-clear
```

Expected output:

```
Allowed networks:            All
Lock after login failures:   Disabled
```

10.5. Deleting a User

Complete the following steps to delete a user:

1. Select **Administration** from the left side panel, then the **Users** tab.
2. Click **Remove**.
3. Confirm by clicking **Delete.**

© 2026 - Hammerspace

10.6. Editing a User

Complete the following steps to edit a user's information:

1. Select **Administration** from the left-side panel, then the **Users** tab.
2. Click the edit icon next to the name of the user you want to modify**.**
3. Make the desired changes and click **Save**.

10.7. Enterprise Sign On

Hammerspace can be configured to use Active Directory to control management access.

This functionality is specific to the management of the Hammerspace infrastructure and does not impact the customer's regular data authentication for SMB, NFS, or S3 protocols.

10.7.1. Limitations

- All configuration is done using the Admin CLI or REST API. There is no GUI support.
- If you configure multiple IDP sources and use a group that exists in both IDP domains, then users from that group from both IDP domains will be able to log in.
- AD users must have their **UserPrincipalName** populated with their `<username>@<domain>`.



Active Directory users are subject to the login rules and password expiration of their Active Directory environment. Local Product rules, such as the number of failed logins, are only enforced for local users and are not applied to Active Directory users.

10.7.2. Setting Up Enterprise Sign On

Configuring the system to accept authentication against external identity providers involves a two-step process. The first step is to add the identity provider, and the second step is to configure which group of that identity provider should be used.

Example scenario

If you want to configure Hammerspace to use the group "Hammerspace Admins" from your Active Directory server. Members of the group should be assigned the **admin** role when they log in using the GUI or CLI.

Example-scenario workflow

This procedure provides instructions for how to add Active Directory as an identity provider in Hammerspace. Once Active Directory is added, users can log in to the GUI using their Active Directory credentials.

1. Add Active Directory as an identity provider.

   Admin CLI

   Command:

© 2026 - Hammerspace

```
idp-add --name PM_DOM_A --domain a.pm.test --type AD --servers ad0.a.pm.test
```

Expected output:

```
ID:                     84697f77-4a5d-45f4-a687-935519c48081
Name:                   PM_DOM_A
Type:                   Active Directory
Domain:                 a.pm.test
Servers:                ad0.a.pm.test
```



Identity providers can be viewed, updated, and removed with the `idp-list`, `idp-update`, and `idp-remove` commands.

2. Configure group-to-role mapping. Add an identity-group mapping for members of the *Hammerspace Admins* group to be recognized with the admin-level group.

   Admin CLI

   Command:

   ```
   identity-group-mapping-create --name Hammerspace_admins --group "Hammerspace Admins" --role-name
   admin
   ```

   Expected output:

   ```
   ID:                     aa2d6e74-d0c2-478c-b57b-14be93c7f395
   Name:                   Hammerspace_admins
   Group:                  Hammerspace Admins
   Management role:        admin
   ```

3. Log in to the GUI. Members of the *Hammerspace Admins* group can now log in to the Management GUI with their credentials.

© 2026 - Hammerspace

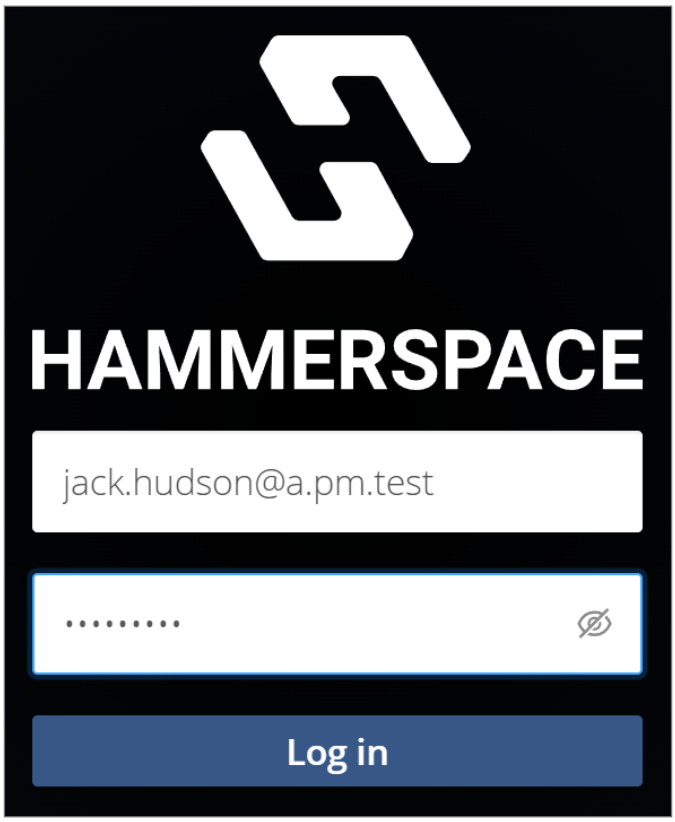


Figure 88. Example login screen with domain address

The username must include the domain to ensure it is not authenticated against local users.

10.7.3. Securing the Connection to the Identity Provider

Starting with Hammerspace 5.3, the connection that Hammerspace uses to check management sign-ins against an Active Directory identity provider can be encrypted. Set the encryption with the `--connection-security` option of `idp-add`, or change it later with `idp-update`:

- `TLS`: LDAPS, encrypted from the start of the connection (port 636). `SSL` is accepted as a synonym.
- `STARTTLS`: the connection starts on port 389 and is then upgraded to TLS.
- `NONE`: not encrypted.

Hammerspace does not validate the server's certificate unless you turn validation on with `--validate-server-certs-enable`.

To encrypt the connection and validate the server's certificate:

1. Add the CA certificate that signed the Active Directory server's certificate to the cluster's trusted certificates, in the Management GUI under **Administration › TLS › Trusted Certificates** or with the `cert-add` command. See Add a Trusted CA Certificate Using the Management GUI.
2. Set the connection security and turn on certificate validation for the identity provider. For example:

© 2026 - Hammerspace

```
idp-update --name <idp-name> --connection-security TLS --validate-server-certs-enable
```



- After an upgrade to 5.3, existing identity providers use connection security `NONE`, with certificate validation not set. Configure each one explicitly.
- To turn validation off for an existing identity provider, use `idp-update --validate-server-certs-disable`. `idp-add` has no option to turn it off.
- Hammerspace does not watch the Active Directory server for later changes to its TLS configuration. If that configuration changes, update the identity provider to match.

10.8. Local Management Users

Local management users are strictly limited to managing the Product and are not permitted to manage data access. By default, one management user is created at installation, named **admin**.



The admin-user password is set during the installation process. If you have lost your admin password, please contact Support for assistance with resetting it.

10.9. Login Policies

Additional security options can be configured to further secure management access to the Hammerspace infrastructure.



Login policies do not affect data access.

10.10. User Access to the Management Interfaces

Access to the GUI, CLI, and API requires authentication by either a locally managed user or enterprise sign-on. Access to functionality within the Product is determined based on the role to which the user is mapped.



Configuring management access for users does not affect data access or cross-protocol User ID mapping.

10.11. User Roles

A user is always tied to a role. There are three default types of management user roles and two types of data access roles.

10.11.1. Management Roles

- **Admin** has full access to all management functionalities.
- **Data Owner** is limited to editing and setting objectives.
- **Viewer** is a read-only role that can view the dashboard.

© 2026 - Hammerspace

10.11.2. Data-Access Roles

- **SMB** role allows a local user to connect to the cluster using SMB. The SMB role is different from an Active Directory role, and *it is not recommended to use this role when Active Directory is available*.
- **The S3** role allows the creation of local users with identities in the file system. This should only be used when configuring an S3 server that is not authenticating to Active Directory.

Additional management roles can be created in the Admin CLI using the `role-create` command.

© 2026 - Hammerspace

Chapter 11. Hammerspace System Management

System management covers the administrative tasks that keep a Hammerspace cluster secure and current. This includes installing a customer-specific X.509 server certificate to replace the default self-signed certificate, backing up and restoring Anvil metadata and system configuration to protect against node loss, and upgrading the Product.

This section includes the following subsections:

- Server Certificate
- System Backup and Restore
- Upgrading the Product

11.1. Server Certificate

By default, Hammerspace ships with a self-signed certificate. This will often cause warnings when clients are connecting to the Management GUI over HTTPS because the certificate does not match what is installed in their browser. The resolution is for the customer to add their certificate to the installation.

Administrators can add a customer-specific **X.509** server certificate. A new certificate can be used to prevent browser security warnings during regular logins to the Management GUI. A new certificate would also be used in other external communication as appropriate.

The following additional information applies to certificates:

- The same certificate is used across all HTTPS connections in Hammerspace, including the S3 endpoint on port 8443.
- By default, without a customer-installed certificate, security warnings will appear on the S3 client.
- The default certificate is self-generated:
  - Key: RSA 3072 bits
  - Algorithm: SHA384withRSA
- The self-generated certificate expires 824 days after it is created. Hammerspace does not warn you before it expires. Track its expiration date, or install your own certificate.
- The self-generated certificate carries only the cluster IP address as a subject alternative name. Browsers and clients that connect by a host name show a warning even after they trust the certificate. To connect by name without a warning, install a certificate that includes the name.

You can install a certificate, view or download the installed certificate, and reset to a new self-generated certificate, either in the Management GUI or with the `cluster-config` command in the Admin CLI.



A new or reset certificate takes effect only after the primary Anvil is rebooted or, in a high-availability (HA) cluster, after a failover to the other Anvil. The Management GUI and the

© 2026 - Hammerspace

> Admin CLI both show the reboot instruction, with the name of the primary Anvil, when the change is accepted. Plan the reboot or failover for a maintenance window.

11.1.1. Installing a Certificate Using the Management GUI

Before you start, have the following files:

- The server certificate in PEM format, optionally followed by its intermediate CA certificates and root CA certificate, in one file. Accepted file extensions are `.pem`, `.crt`, and `.cer`.
- The certificate's RSA private key in PKCS#1 or PKCS#8 PEM format, in a file with the extension `.key`, `.pem`, `.crt`, or `.cer`. The key can be an encrypted PKCS#8 key. If it is, you need its passphrase.

1. Log in to the Management GUI as an administrator.
2. Navigate to **Administration** › **Certificates**. The **Cluster Certificate Details** section shows the installed certificate.
3. Click **Upload New Certificate**.
4. In the **Upload New Certificate** dialog, under **Private Key:**, select how you will provide the private key:
   - **File**: In **Private Key File:**, select the key file. If the key is encrypted, enter its passphrase in **Key File Passphrase:**. Leave the passphrase empty if the key is not encrypted.

     Do not type anything in **Key File Passphrase:** for an unencrypted key, even if you delete it again. Once the field has held a value, the upload of an unencrypted key fails with the `Bad certificate chain or private key` message. Close the dialog and open it again.

   - **Plain text**: Paste the PEM-encoded private key, including its `-----BEGIN` and `-----END` lines. The key must not be encrypted; the passphrase field is available only with **File**.
5. In **Certificate:**, select the certificate file. If you click **Upload** without one, the dialog shows `Please select a certificate file.` under the field.
6. Click **Upload**.

   When the certificate is accepted, a dialog shows `Certificate uploaded successfully.` and instructs you to reboot the primary Anvil, by name, for the new certificate to take effect. For a few seconds after you close the dialog, the **Web Server Certificate** tab shows `CERTIFICATE UNDER CONSTRUCTION` with empty fields and 1970 dates; it then shows the uploaded certificate's details. The certificate that the GUI and REST API serve does not change until the reboot (or, in an HA cluster, a failover).

1. Click **Close**.
2. Reboot the primary Anvil during a maintenance window.
3. After the reboot, check that the details section shows the new certificate's **Subject** and **Validity**, then reload the Management GUI in your browser and confirm that it presents the new certificate.

If the key is in an unsupported format, if the passphrase is wrong, or if a passphrase is given for a key that is not encrypted, the upload fails with the message `Bad certificate chain or private key. Ensure the private key format is valid and the passphrase is correct if the key is encrypted.` A key that does

© 2026 - Hammerspace

not match the certificate fails with `Private key is invalid for specified certificate.`

11.1.2. Viewing and Downloading the Installed Certificate

Navigate to **Administration › Certificates**. The **Cluster Certificate Details** section shows the installed certificate's version, serial number, signature algorithm, issuer, subject, validity dates, and fingerprints.

To save the certificate chain to a file, click **Download Certificate**. The browser saves a file named `certificate-chain.crt`.

In the Admin CLI, `cluster-config --view` shows the same details under **Server Certificate chain**.

11.1.3. Installing a New Certificate Using the Admin CLI

Log in to the Anvil as an admin-level role and install the new certificate with `cluster-config --server-certificate-chain`. It is essential that the entire certificate is copied exactly as it appears. Use `" "` to ensure that the system correctly interprets the input text.

The `--server-private-key` option must also be used. If you omit it, the Admin CLI prompts for the private key; the prompt works only in an interactive session. The `--server-private-key` option cannot be used without `--server-certificate-chain`.



The `--server-private-key` option accepts only an unencrypted RSA private key in PKCS#1 or PKCS#8 PEM format. To install a certificate whose private key is encrypted, use the Management GUI and provide the key as a file with its passphrase. See the Installing a Certificate Using the Management GUI section of this page.



Type or paste the command in an interactive Admin CLI session. The quoted certificate and key span several lines, and the CLI only continues a quoted value across lines in an interactive session; input that is piped or scripted (for example, `ssh` with a command argument) is read one line at a time and fails with syntax errors.

Admin CLI

Command:

```
cluster-config --server-certificate-chain "-----BEGIN CERTIFICATE-----
<server certificate, followed by any intermediate and root CA certificates>
-----END CERTIFICATE-----" --server-private-key "-----BEGIN RSA PRIVATE KEY-----
<private key>
-----END RSA PRIVATE KEY-----"
```

The command prints the cluster configuration (the output of `cluster-view`) and ends by confirming that the certificate was accepted, with the command to reboot the primary Anvil. Reboot the primary Anvil at this time to activate the certificate. On a high-availability cluster, a failover to the other Anvil also activates it.

Expected output:

```
ID:                                 0ec84e65-0258-45bf-8d29-85db81588f5f
Name:                               AE742ZCK22YBB0
State:                              Standalone
Management IPs:                     [192.0.2.10/20]
Data IPs:                           [192.0.2.10/20]
Cluster floating IPs:               [192.0.2.10/20]
Since:                              2026-09-21 23:51:44 UTC
Created:                            2026-09-21 23:51:24 UTC
Timezone:                           UTC
GFS participant max suspected time: 30 minutes
Metered billing:                    Disabled
Prometheus exporters:               Disabled
Evaluation expiration:              2026-10-21 23:51:24 UTC
Online license activation support:  true
NAS volume capacity:                [Total: 42.9GB, Used: 1.3GB, Free: 41.6GB]
Share space (quota):                [Total: 1GB, Used: 943.7MB, Free: 56.2MB]
EULA accepted:                      2026-09-22 00:26:56 UTC
Metadata servers:
                        [Object type: Anvil, Node name: anvil.example.com, Role: Primary, Admin state:
Up, Oper state: Up]
Server Certificate chain:
                        Version:                3
                        Serial:                 803038211491614656512525949954669386929210083
(0x1681844070edeea0ba30832cad5548120d61ce1)
                        Algorithm:              SHA256withRSA
                        Issuer:                 CN=Example Docs Root CA, O=Example Docs, C=US
                        Subject:                CN=anvil.example.com, O=Example Docs, C=US
                        Validity:               Not Before:             2026-09-22 00:39:31 UTC
                                                Not After:              2028-12-25 00:39:31 UTC
                        Fingerprints:           SHA1:
2c:9f:b9:30:10:1a:a1:73:be:41:7f:38:89:d8:50:e8:83:fb:c:e1
                                                MD5:
de:a3:c2:37:ca:3d:25:4c:af:86:da:68:ff:ea:f6:cd
                                                CRC32:                  0xfcad84d8

                        Version:                3
                        Serial:                 17057174381434182781035685110594655819563670115 8
(0x1de0b39728beeb751870a62aebdfa3b55c0d07e6)
                        Algorithm:              SHA256withRSA
                        Issuer:                 CN=Example Docs Root CA, O=Example Docs, C=US
                        Subject:                CN=Example Docs Root CA, O=Example Docs, C=US
                        Validity:               Not Before:             2026-09-22 00:39:31 UTC
                                                Not After:              2036-09-19 00:39:31 UTC
                        Fingerprints:           SHA1:
bb:e3:b6:b8:3:28:8d:5a:28:c5:5c:30:a5:d3:d8:57:69:33:61:98
                                                MD5:
41:e9:65:dd:9:41:ca:53:68:1d:68:d:7a:b6:82:21
                                                CRC32:                  0x39c9096a
```


© 2026 - Hammerspace

The certificate fingerprint can be viewed using the `cluster-view` command.

Admin CLI

Command:

```
cluster-config --view
```

Expected output:

```
ID:                                 0ec84e65-0258-45bf-8d29-85db81588f5f
Name:                               AE742ZCK22YBB0
State:                              Standalone
Management IPs:                     [192.0.2.10/20]
Data IPs:                           [192.0.2.10/20]
Cluster floating IPs:               [192.0.2.10/20]
Since:                              2026-09-21 23:51:44 UTC
Created:                            2026-09-21 23:51:24 UTC
Timezone:                           UTC
GFS participant max suspected time: 30 minutes
Metered billing:                    Disabled
Prometheus exporters:               Disabled
Evaluation expiration:              2026-10-21 23:51:24 UTC
Online license activation support:  true
NAS volume capacity:                [Total: 42.9GB, Used: 1.3GB, Free: 41.6GB]
Share space (quota):                [Total: 1GB, Used: 943.7MB, Free: 56.2MB]
EULA accepted:                      2026-09-22 00:26:56 UTC
Metadata servers:
                        [Object type: Anvil, Node name: anvil.example.com, Role: Primary, Admin state:
Up, Oper state: Up]
Server Certificate chain:
                        Version:                3
                        Serial:                 803038211491614656512525949954669386929210083
3
(0x1681844070edeea0ba30832cad5548120d61ce1)
                        Algorithm:              SHA256withRSA
                        Issuer:                 CN=Example Docs Root CA, O=Example Docs, C=US
                        Subject:                CN=anvil.example.com, O=Example Docs, C=US
                        Validity:               Not Before:             2026-09-22 00:39:31 UTC
                                                Not After:              2028-12-25 00:39:31 UTC
                        Fingerprints:           SHA1:
2c:9f:b9:30:10:1a:a1:73:be:41:7f:38:89:d8:50:e8:83:fb:c:e1
                                                MD5:
de:a3:c2:37:ca:3d:25:4c:af:86:da:68:ff:ea:f6:cd
                                                CRC32:                  0xfcad84d8

                        Version:                3
                        Serial:                 170571743814341827810356851105946558195636701158
(0x1de0b39728beeb751870a62aebdfa3b55c0d07e6)
                        Algorithm:              SHA256withRSA
```


© 2026 - Hammerspace

```
                        Issuer:                 CN=Example Docs Root CA, O=Example Docs, C=US
                        Subject:                CN=Example Docs Root CA, O=Example Docs, C=US
                        Validity:               Not Before:             2026-09-22 00:39:31 UTC
                                                Not After:              2036-09-19 00:39:31 UTC
                        Fingerprints:           SHA1:
bb:e3:b6:b8:3:28:8d:5a:28:c5:5c:30:a5:d3:d8:57:69:33:61:98
                                                MD5:
41:e9:65:dd:9:41:ca:53:68:1d:68:d:7a:b6:82:21
                                                CRC32:                  0x39c9096a
```

11.1.4. Resetting the Installed Certificate

The system can be reset to use a self-generated certificate, effectively removing any manually installed certificate. The new self-generated certificate is valid for 824 days.

To reset the certificate in the Management GUI:

1. Navigate to **Administration** › **Certificates**.
2. Click **Reset Certificate**.
3. The confirmation dialog states that the current certificate will be replaced with a fresh internally generated self-signed certificate, and that a reboot of the primary Anvil is required. Click **Yes**.
4. Reboot the primary Anvil during a maintenance window. On a high-availability cluster, a failover to the other Anvil also activates the new certificate.

To reset the certificate in the Admin CLI:

Admin CLI

Command:

```
cluster-config --server-certificate-reset
```

Expected output:

```
ID:                                 0ec84e65-0258-45bf-8d29-85db81588f5f
Name:                               AE742ZCK22YBB0
State:                              Standalone
Management IPs:                     [192.0.2.10/20]
Data IPs:                           [192.0.2.10/20]
Cluster floating IPs:               [192.0.2.10/20]
Since:                              2026-09-21 23:51:44 UTC
Created:                            2026-09-21 23:51:24 UTC
Timezone:                           UTC
GFS participant max suspected time: 30 minutes
Metered billing:                    Disabled
Prometheus exporters:               Disabled
Evaluation expiration:              2026-10-21 23:51:24 UTC
```


© 2026 - Hammerspace

```
Online license activation support:  true
NAS volume capacity:                [Total: 42.9GB, Used: 1.3GB, Free: 41.6GB]
Share space (quota):                [Total: 1GB, Used: 943.7MB, Free: 56.2MB]
EULA accepted:                      2026-09-22 00:26:56 UTC
Metadata servers:
                        [Object type: Anvil, Node name: anvil.example.com, Role: Primary, Admin state:
Up, Oper state: Up]
Server Certificate chain:
                        Version:                3
                        Serial:                 8711156112346315821 (0x78e43e549b5ffc2d)
                        Algorithm:              SHA384withRSA
                        Issuer:                 O=Hammerspace, C=US, CN=Hammerspace-97013624
                        Subject:                O=Hammerspace, C=US, CN=Hammerspace-97013624
                        Validity:               Not Before:             2026-09-22 00:45:25 UTC
                                                Not After:              2028-12-24 00:45:25 UTC
                        Fingerprints:           SHA1:
d:eb:32:9c:e:3e:ca:2a:89:ff:b:f:5d:4:de:d4:c4:9a:b5:d9
                                                MD5:
18:6b:5b:8c:e:53:99:4d:e0:5a:ef:c3:b9:c5:ba:e8
                                                CRC32:                  0x780535db
```

11.2. System Backup and Restore

Backup and recovery protect against the loss of the Anvil nodes. The **backup** command takes a point in time backup of metadata and system configuration. The Restore command restores the metadata and configuration from a backup.

If recovering from a disaster, re-install the Anvil nodes with the same version used to take the backup and then use the restore command.

It is only supported to run Restore from freshly installed Anvil nodes.



If restoring a system with Global File System shares, additional steps must be taken for recovery of those shares. Contact Hammerspace Support for assistance.



System Backup and Restore does NOT backup actual user data.

11.2.1. System Backup

To do a backup or restore operation, login to the command line as an admin user. The `system-backup-config` command enables a backup to be scheduled or taken immediately.

A backup from an Anvil cluster is stored in a unique sub-directory under the specified path using the directory naming of anvil-<UUID>.

Several versions of the backups are automatically stored in the destination directory. The backup directory name is date-time to ensure they are unique.

Each backup is a full backup and is independent of the other backups. A total of five backups are stored in one

 © 2026 - Hammerspace

location. When the number of backups exceeds five, the system will periodically remove the oldest backup.

The following example provides the IP address of the storage node and the NFSv3 export path to back up to. The `--now` option takes the backup immediately.

Admin CLI

Command:

```
system-backup-config --ip 192.0.2.52 --path /backup --now
```

Expected output:

```
success
```

Creating a System Backup Schedule

The example below runs at 10 minutes past each hour.

Admin CLI

```
schedule-create --minute 10 --name backup-schedule
```

Creating a Scheduled Backup

Admin CLI

Command:

```
system-backup-config --ip 192.0.2.52 --path /backup --schedule-name backup-schedule
```

Expected output:

```
Export path:            /backup
IP:                     192.0.2.52/32
Schedule name:          backup-schedule
Schedule expression:    10 * * * *
Schedule description:   At 10 minutes past the hour
ID:                     74917d09-1ea4-4244-aba9-b7efca5cd474
```

Checking the Backup Schedule

Admin CLI

Command:

© 2026 - Hammerspace

```
system-backup-config --view
```

Expected output:

```
Export path:            /backup
IP:                     192.0.2.52/32
Schedule name:          backup-schedule
Schedule expression:    10 * * * *
Schedule description:   At 10 minutes past the hour
ID:                     ed57a683-de7b-4ccb-b163-30641e314759
```

Changing or Removing a Schedule and/or Changing the Backup Location

This example changes the schedule from hourly to daily.

Admin CLI

Command:

```
system-backup-config --ip 192.0.2.52 --path /backup --schedule-name backup-schedule-daily
```

Expected output:

```
Export path:            /backup
IP:                     192.0.2.52/32
Schedule name:          backup-schedule-daily
Schedule expression:    59 23 * * *
Schedule description:   At 11:59 PM
ID:                     c72cc474-086a-4cbe-8205-7188263692dc
```

Listing System Backups

The `system-backup-list` command can be used to list available backups.

Admin CLI

Command:

```
system-backup-list --ip 192.0.2.52 --path /backup
```

Expected output:

```
total 1
Cluster UUID:           07da6cde-cd13-4ca8-9511-6cd1180d6fe6
Available Backups:
```


© 2026 - Hammerspace

```
Name:                   20260921.155348Z
Time:                   Mon Sep 21 15:53:50 UTC 2026
Cluster name:           AE54099FEVN3BN
Management IP:          192.0.2.10
Site name:              AE54099FEVN3BN
Version:                5.2.14-1266
Size:                   3.5MB

Name:                   20260921.155308Z
Time:                   Mon Sep 21 15:53:10 UTC 2026
Cluster name:           AE54099FEVN3BN
Management IP:          192.0.2.10
Site name:              AE54099FEVN3BN
Version:                5.2.14-1266
Size:                   3.5MB
```



If the path for the `system-backup-list` command does not contain any backups of the current system, then all backups that are in that location will be shown.

11.2.2. Restoring System from Backup

The `backup-restore` command restores the system configuration and metadata. It does not restore or modify any user data.

A metadata backup must be restored to the same Anvil configuration as where the backup was taken. For example, a backup of an HA Anvil configuration must be restored to a HA installation.

The duration of a restore operation depends on the size of the backup. Most configurations are restored within 30 minutes.

The following example provides the IP address of the storage node and the export path of the backup directory to restore from.

Admin CLI

Command:

```
system-backup-restore --ip 192.0.2.52 --path /backup
```

Expected output:

```
backup restore started
```



It is only supported to run a restore operation from a freshly installed system that has not yet been used.

© 2026 - Hammerspace

11.3. Upgrading the Product

The procedure for upgrading the Product is described in detail in the *Hammerspace Installation and Licensing Guide*, which is available on the *Hammerspace License and Delivery Portal*, Hammerspace Documentation downloads page.

If any DSX node uses NVMe-oF storage, also read Upgrading a Cluster That Uses NVMe-oF Storage before you start.

© 2026 - Hammerspace

Appendices

© 2026 - Hammerspace

Appendix A: Technology Preview

This chapter describes features that are provided in the Hammerspace product (Product) for testing purposes but are not supported for production use.

This section includes the following subsections:

- Connecting Storage to a DSX Using NVMe-oF

A.1. Connecting Storage to a DSX Using NVMe-oF

Hammerspace supports NVMe over Fabric (NVMe-oF), enabling NVMe drives to be added to a DSX node. This extends the storage capacity of the DSX node by providing a managed, reboot-persistent way to mount and incorporate networked NVMe block devices from external arrays into the cluster's usable volume pool. Hammerspace supports NVMe-oF over TCP.

- **Configuration and management:**
  - Configuration is performed on a per-DSX node basis. Multiple NVMe-oF enclosures (targets) can be added to a single DSX node.
  - The Admin CLI command `nvmeof-config` handles all NVMe-oF operations, including adding enclosures, connecting subsystems, and changing each enclosure's connection settings.
  - The system uses NVMe Qualified Names (NQNs) for identification. Each DSX node has a host NQN that identifies it to the storage array, and each subsystem (which contains namespaces/drives) has its own NQN. The host NQN is an identifier, not a credential. Enter the DSX host NQN into the storage array's host access configuration so that the array presents only the intended namespaces to that DSX.
  - The configuration workflow involves discovering targets and then connecting to discovered namespaces to expose them as block devices.



- **Logical volume creation:** Once the external storage devices (namespaces) are connected and discovered by the DSX, administrators can create logical volumes on them (i.e., format them with a file system) and then add them to the Hammerspace cluster as regular volumes.
- **Multipathing and redundancy:**
  - The system supports multipathing, which is automatically enabled if multiple IP address targets are provided for a single NVMe enclosure. Provide every target address when you add the enclosure; a target is identified by its address, and a target added to an enclosure after its subsystems are connected does not become a path to them.
  - Multipathing enhances redundancy (if one IP network fails, the system can use the other) and helps distribute load across multiple paths.
  - Configurable parameters include the I/O policy for multipathing. The supported policies are `queue-depth` (the default), `round-robin`, and `numa`.

© 2026 - Hammerspace

- **Configuration persistence and boot ordering:** Connections to NVMe-oF namespaces persist across reboots. On each boot, Hammerspace reconnects the configured subsystems after the network is up and before file systems are mounted, so the block devices are present when the mounts occur. No additional configuration is required.
- **Health monitoring:** A DSX raises two NVMe-oF events to System Manager, both per path. **NVMe-oF path degraded** is raised at Warning severity when a path has been stuck trying to reconnect. **NVMe-oF path lost** is raised at Warning severity while the subsystem still has at least one working path, and at Critical severity once the subsystem has no working paths left. Both clear automatically when the path recovers, and both are debounced, so a brief interruption does not raise an event. By default, reconnection attempts are unlimited. For the event names, what each severity means, and the matching SNMP traps, see NVMe-oF Path Health and Recovery.
- Use the `nvmeof-config` command to view configured enclosures and the state of their connections. When given no subcommand, it defaults to listing.
- When connecting a subsystem within an enclosure, all of the namespaces within that subsystem are connected.
- The same is true for disconnecting from a subsystem; all namespaces of that subsystem are also disconnected from the DSX. However, even when a subsystem is disconnected, the namespaces remain visible in Hammerspace inventory until the Hammerspace administrator reconciles the inventory.

To remove disconnected namespaces from the inventory, refresh the DSX node:

Command:

```
node-refresh --name <dsx-node-name> --reconcile-components
```

This section includes the following subsections:

- Adding an NVMe-oF Enclosure and Subsystems
- Admin CLI Command: nvmeof-config
- Admin CLI Workflow to Connect NVMe-oF Enclosures
- Additional Admin CLI examples
- NVMe-oF Path Health and Recovery
- NVMe-oF Limitations and Operating Guidance
- Upgrading a Cluster That Uses NVMe-oF Storage

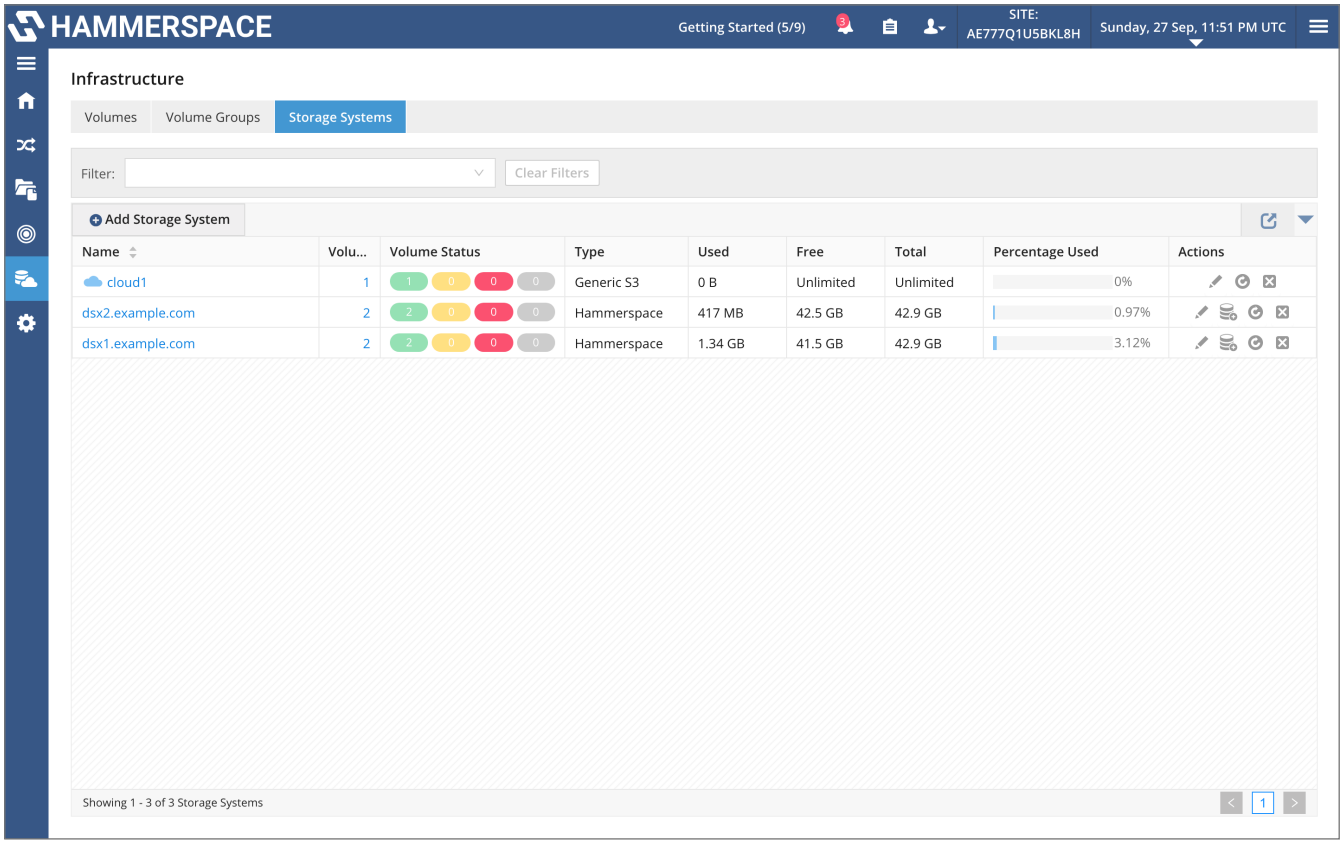
A.1.1. Adding an NVMe-oF Enclosure and Subsystems

Add an NVMe-oF enclosure to a DSX node using the Management GUI.

1. Log in to the Management GUI and open the **Infrastructure** page.

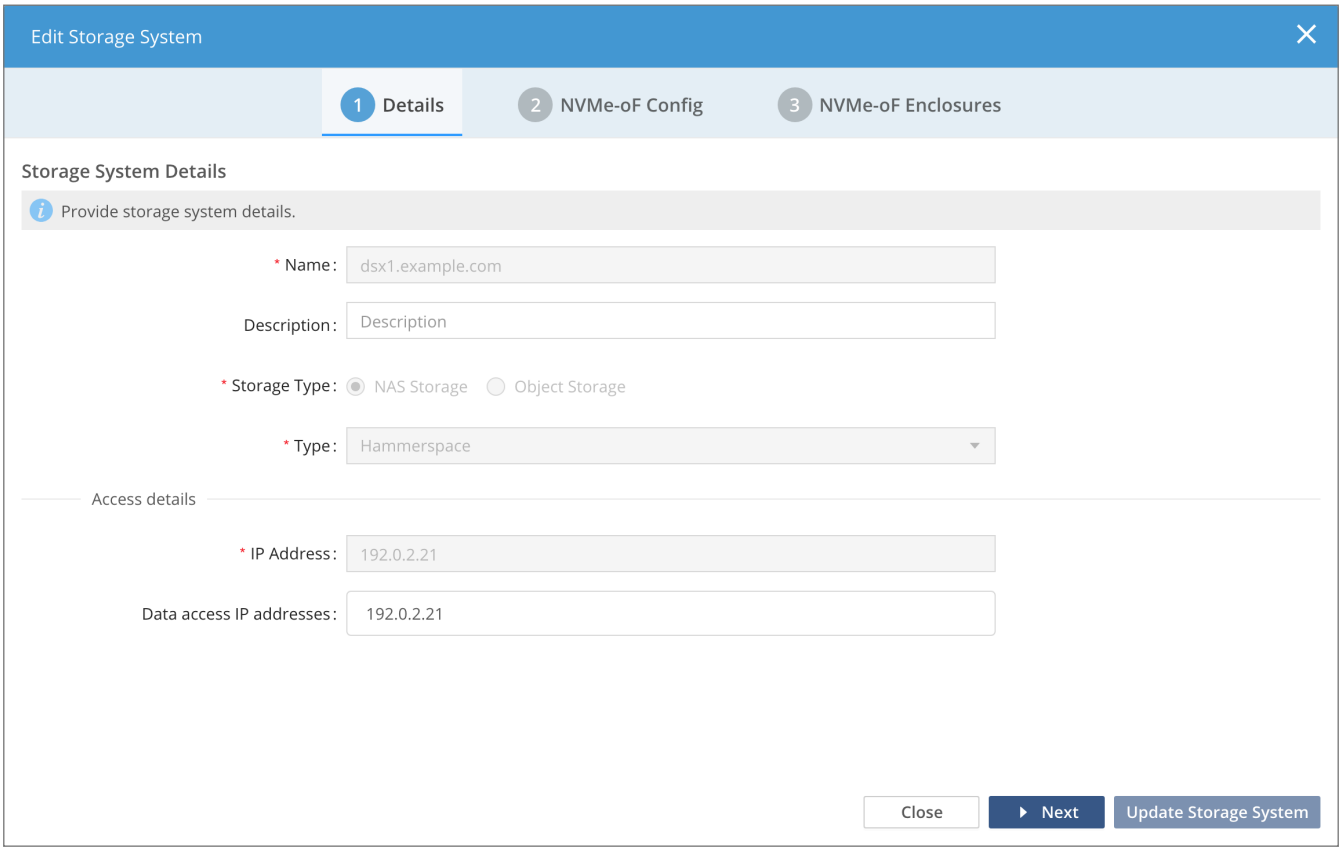
© 2026 - Hammerspace

2. Click the **Storage Systems** tab and identify the DSX node to which you will connect the NVMe device.



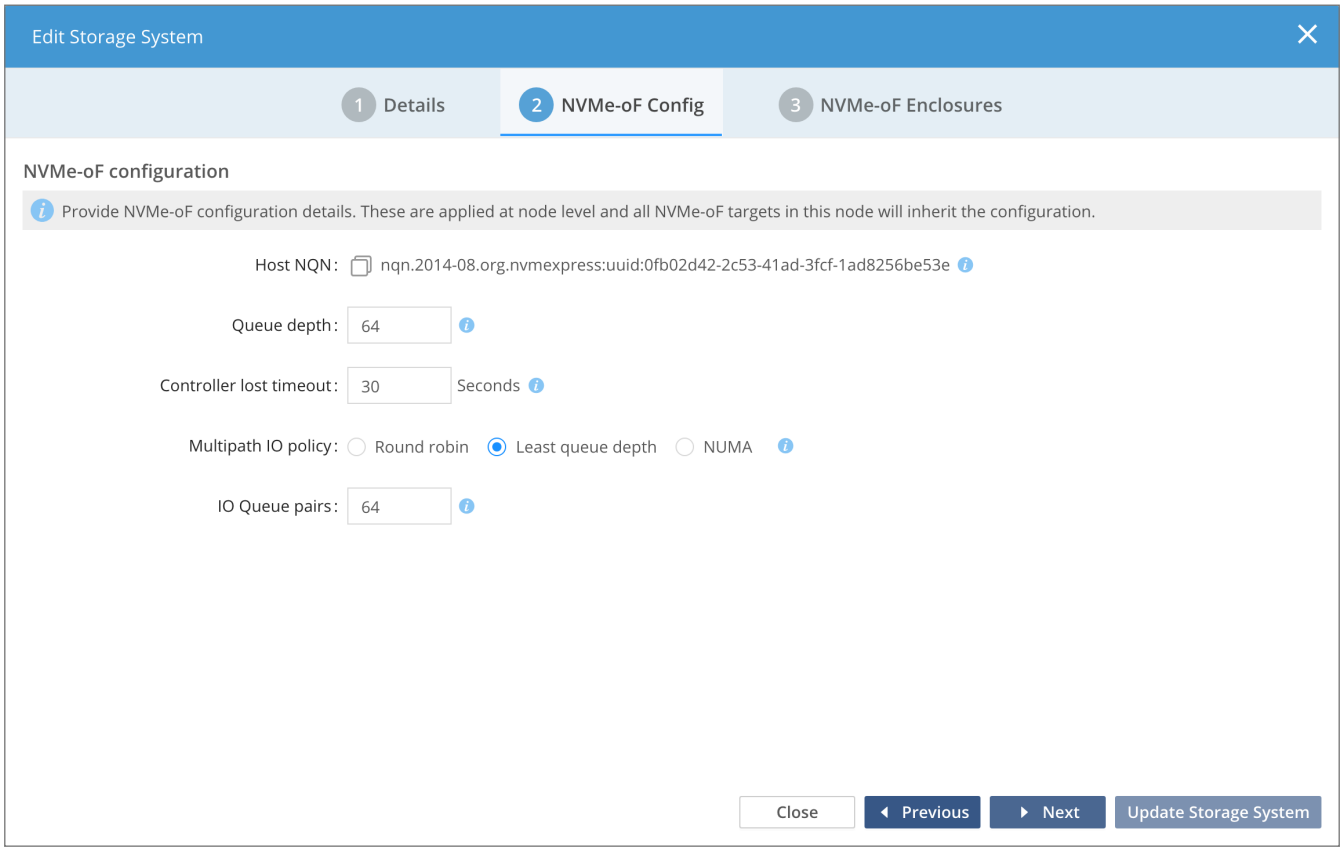
3. In the **Actions** column, click the pencil icon to edit the node. The Edit Storage System wizard opens with the **Details** tab visible. Note that the information in the Details tab was configured during the DSX node installation and cannot be changed here.

© 2026 - Hammerspace



4. Click **Next**.
5. On the **NVMe-oF Config** tab, the **Host NQN** field shows the DSX node's host NQN, which you add to the array's host access configuration. The connection settings on this tab (**Queue depth**, **Controller lost timeout**, **Multipath IO policy**, and **IO Queue pairs**) apply to every NVMe-oF enclosure on this DSX node. For a brief explanation of each setting, click its information icon. Click **Next** to continue.

© 2026 - Hammerspace



Before the node has an enclosure, the tab suggests a queue depth of 24. An enclosure added from the Admin CLI uses the product default of 64. Once the node has an enclosure, the tab shows that enclosure's values.

6. In the **NVMe-oF Enclosures** tab, click the **Add NVMe-oF Enclosure** button to add details for an enclosure. Mandatory fields are marked with a red asterisk. You can add multiple enclosures to the DSX; however, each must be added separately using the **Add NVMe-oF Enclosure** button.
   a. Provide a name for the enclosure.
   b. Optionally, enter a description.
   c. In **Target details**, enter the host name or IP address of the array's discovery target. The port defaults to 8009 and the transport to TCP.
      i. Add every address that the array presents the subsystems on by clicking the **+** button for each one. If you add more than one, the **Multipath** box is selected automatically; you cannot select it yourself. Add all of the addresses now: an address added to the enclosure later does not become a path to subsystems that are already connected.
      ii. When you add the required details, the cluster searches for subsystems on the target. The search takes about 2½ minutes; the **Subsystems** panel lists the subsystems when it finishes. Click **Refresh** in the **Subsystems** panel to search again.
      iii. Select the check box next to each subsystem to connect, or the **Subsystems** check box to select them all. No subsystem is selected until you select it.

© 2026 - Hammerspace

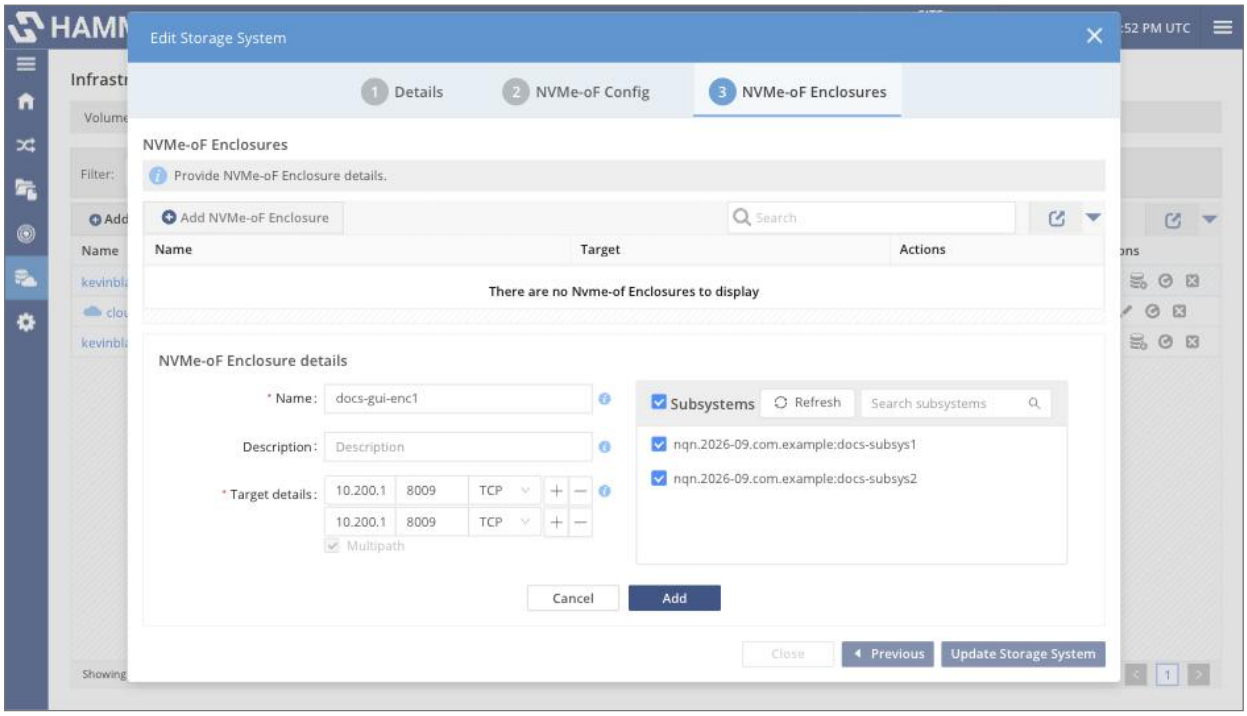


Figure 89. An enclosure with two target addresses and two subsystems selected

7. After configuring the enclosure details and which subsystems to connect to, click **Add**. The enclosure appears in the table; **+1** after the target means that the enclosure has one more target address.

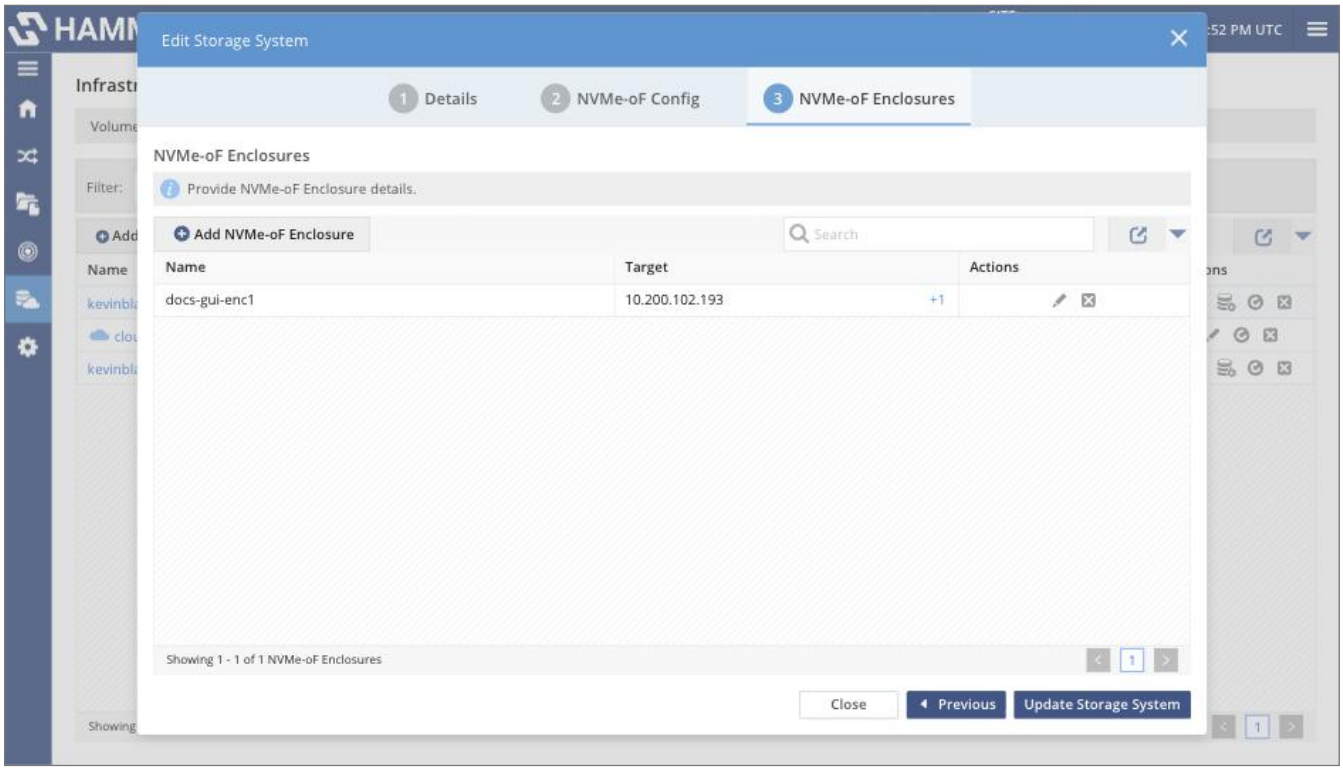
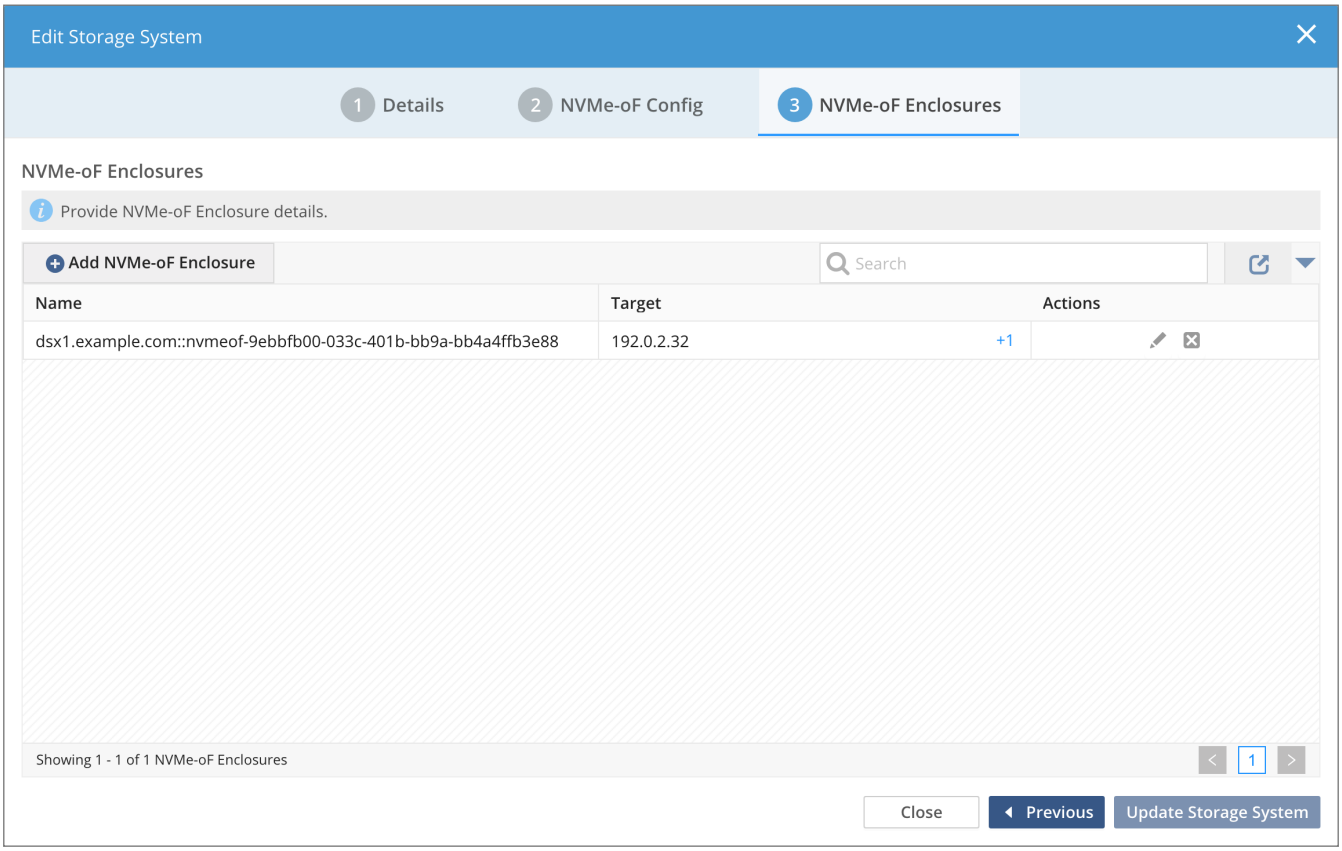


Figure 90. The new enclosure in the NVMe-oF Enclosures table

Click **Update Storage System** for Hammerspace to accept the device. Adding the enclosure and refreshing the storage system takes a few minutes. The number of block devices in the **Logical Volumes** section of the result includes the DSX node's own disks.

© 2026 - Hammerspace



8. Connect the subsystems. Hammerspace adds the enclosure with its subsystems discovered but not connected, whatever you selected. On the **NVMe-oF Enclosures** tab, click the pencil icon of the new enclosure, wait for the **Subsystems** panel to list the subsystems again (the search runs again and takes about 2½ minutes), select the ones to connect, click **Update**, and then click **Update Storage System** again. Connecting takes about a minute. Allow about 8 minutes for the whole procedure with two subsystems.

   To check, run `nvmeof-config --list --full` in the Admin CLI: each connected subsystem shows `State: CONNECTED` on its target addresses and its namespaces.

After the refresh completes, the popup displays the new enclosures and confirms that the storage system was updated successfully. You will also see the available block devices. Next, click Configure Logical Volumes to configure the file systems on the blank block devices.

Configuring the Logical Volumes

This procedure assumes you have just configured and connected NVMe-oF enclosures and subsystems, and that you are in the popup.

1. Click **Configure Logical Volumes**. This opens the Configure Logical Volumes wizard. The **Block Devices** table shows all of the block devices recognized by the system, with their **Device Path**, **State**, **Force Initialize** box, **Logical Volume** (if one exists), **Serial No**, and **Storage System**. The NVMe-oF namespaces you connected appear as `/dev/nvme⋯` devices, with a **Serial No**. The table also lists the DSX node's own devices: its disks (`/dev/sd⋯`, which already have a logical volume) and device-mapper devices (`/dev/dm-⋯`, no serial number). Select only the `/dev/nvme⋯` rows.

© 2026 - Hammerspace

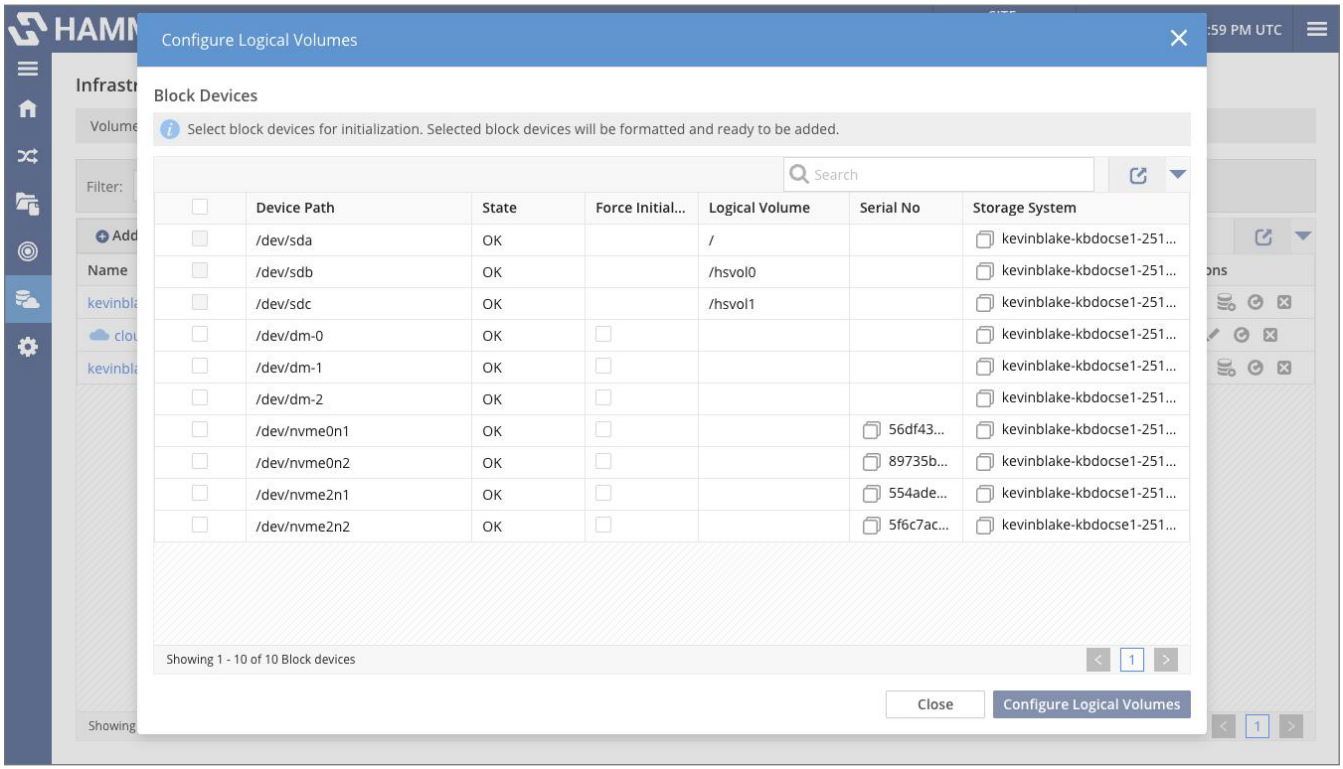


Figure 91. The Block Devices table

2. Check the box next to each NVMe drive you added earlier, then click **Configure Logical Volumes**.

   If a drive belongs to a subsystem that is also connected to another DSX node, the operation fails for that drive with **Error creating logical volume for** `<device-path>`. Present each subsystem to one DSX node only.

+

> As a safety feature, Hammerspace will fail the operation to configure logical volumes if it detects that a drive already contains customer data. To use a drive that was previously written and contains data that is safe to delete, you can click the box in the same row as the drive, under the Force Initialize column, to allow Hammerspace to configure the drive.

+ Without Force Initialize, such a drive fails with **Error creating logical volume for** `<device-path>` and no reason. This includes a drive whose logical volume was deleted, because deleting a logical volume leaves its file system on the drive. The **Force Initialize** box appears only on rows that have no logical volume.

+ .Force Initialize selected for a drive that was used before image::admin-configuring-the-logical-volumes-image3.png[]

1. Click **Configure Logical Volumes**. A popup shows the progress of the operation as a logical volume is created for each drive. Creating logical volumes on two drives took about 20 seconds.

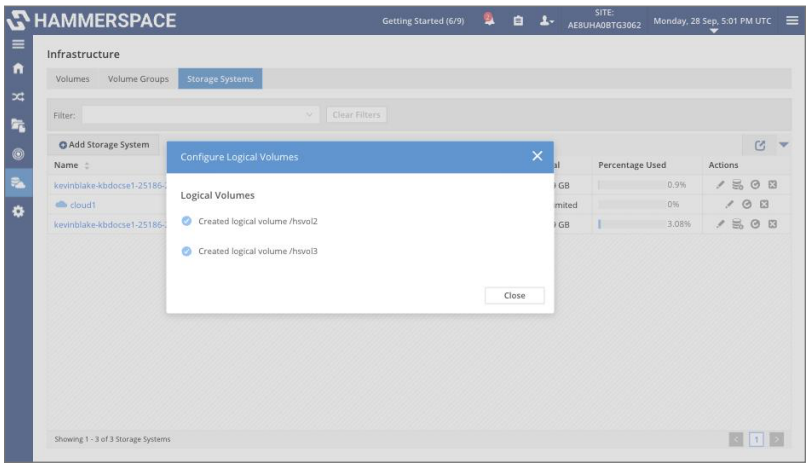


Figure 92. Logical volumes created

After the first logical volume is created, a **+ Volume** button appears beside the DSX node's name on the **Storage Systems** tab.

2. After the operation has finished successfully, go to the **Infrastructure** page > **Volumes** tab, and click **Add Volume**.
3. In the Add Volumes popup, the **Selections** tab shows previously added volumes with grayed, already-active check boxes and the recently added logical volumes for each drive. Check the box next to each new logical volume and click **Next Step**.
4. In the **Data** tab for each volume, you can select to restrict data access to Read Only and to assimilate the data on the volume. Once you have made these selections, click **Next Step**.
5. In the **Capacity** tab, you can adjust the default capacity threshold parameters as needed. Click **Next Step**.
6. In the **Capability** tab, the default, auto-created objectives for durability and availability are already selected for each volume. Make the selections and click **Next Step**.

> Auto-created objectives are the best option in most cases; however, you can unselect the auto-created objective and specify durability and availability objectives manually. You can also select whether to perform a performance test on the volumes using the selected objective.

7. In the **IP** tab, you will see the discovered IPs, and you can add additional IPs and exclude IPs for each volume as needed. Make your selections and click **Next Step**.
8. In the **Review & Add** tab, you can review volume details and click **Add Volumes** to add them to your DSX infrastructure.

As the operation progresses, you can view the status in the **Tasks** view.

A.1.2. Admin CLI Command: nvmeof-config

NVMe-oF actions use the `nvmeof-config` command and its options. The command is designed to enable users to do multiple tasks at once, for example, adding an enclosure and connecting to certain subsystems, or connecting to some subsystems while disconnecting from others.

Usage: `nvmeof-config [options]`

Command:

```
nvmeof-config --help
```

Expected output:

```
Usage: nvmeof-config [options]
Configure and view NVMe-oF enclosures and configurations

  Options:
    --connect-all        Connect all targets of the specified NVMe-oF enclosure
    --connect-nqn        Subsystem NQN(s) to connect
    --ctrl-loss-tmo      Number of seconds to wait for a controller to reconnect
                         before giving up (-1 retries indefinitely)
    --disconnect-all     Disconnect all targets of the specified NVMe-oF
                         enclosure
    --disconnect-nqn     Subsystem NQN(s) to disconnect
    --dsx-id             The ID of the DSX node when adding an NVMe-oF enclosure
    --dsx-internal-id    The internal ID the DSX when adding an NVMe-oF
                         enclosure
    --dsx-name           The name of the DSX node when adding an NVMe-oF
                         enclosure
    --enclosure-add      Add an NVMe-oF enclosure to a DSX
    --enclosure-id       UUID of enclosure to operate on
    --enclosure-name     The name of the enclosure to be operated on
    --enclosure-remove   Remove an NVMe-oF enclosure by name or Id
    --full               Print extra information for each element
    --help               Display this help and exit
    --io-policy          Controller I/O policy. Possible values: ROUND_ROBIN |
                         NUMA | QUEUE_DEPTH.
    --io-queue-pairs     Number of request/completion queue pairs
    --list               List all NVMe-oF enclosures across all DSXs
    --queue-depth        Maximum number of I/O queue entries
    --target             The target(s) and optionally port and transport type
                         when adding or updating an enclosure
                         Examples:
                         --target 10.10.1.1
                         --target nvme1.local
                         --target 10.10.1.1,8009
                         --target nvme2.local,1234,TCP
    --target-port        Network target port
    --transport          Network transport type. Possible values: TCP | RDMA.
```

Examples

`--target 10.10.1.1`

© 2026 - Hammerspace

`--target nvme1.local`

`--target 10.10.1.1,8009`

`--target nvme2.local,1234,TCP`

A.1.3. Admin CLI Workflow to Connect NVMe-oF Enclosures

Connect an NVMe-oF enclosure to a DSX node by using the Admin CLI on the Anvil, the Hammerspace metadata and management node.

1. Find the name of the DSX node that will own the enclosure:

   Command:

   ```
   node-list
   ```

   In the output, the DSX nodes show `Product node type: DSX`.

2. Read the DSX node's host NQN, and add it to the array's host access configuration so that the array presents the intended namespaces to that node. The host NQN identifies the DSX node to the array. It is not a password.

   Command:

   ```
   node-list --name <dsx-node-name>
   ```

   The host NQN is on the line `NVMe-oF host NQN:`. In the Management GUI, it is the **Host NQN** field on the **NVMe-oF Config** tab that opens when you edit the DSX node on the **Storage Systems** tab of the **Infrastructure** page.

   Expected output (trimmed):

   ```
   Name:                          dsx.example.com
   Type:                          Product
   Product node type:             DSX
   Internal ID:                   1073741832
   ID:                            4cdae232-a7de-501a-ad2a-1e90457a0f98
   HW state:                      OK
   Node state:                    MANAGED
   Node mode:                     ONLINE
   Management IP:                 192.0.2.21/20
   SW version:                    5.2.14-1266
   S3 auth signing type:          Default
   Vendor:                        VMware, Inc.
   Product name:                  VMware Virtual Platform
   Serial number:                 8d8c7b0f9c31
   ```


© 2026 - Hammerspace

```
Firmware version:              None
NVMe-oF host NQN:              nqn.2014-08.org.nvmexpress:uuid:63422d42-6bd4-a136-5617-8d8c7b0f9c31
Created:                       2026-09-27 23:23:16 UTC
Modified:                      2026-09-27 23:42:38 UTC
Components:
                         Type:                 Hdd
                         ID:                   5f2e0a0f-323c-4172-82cc-7ecb0d592698
                         Name:                 /dev/sdc
                         State:                OK

<Additional output removed for brevity>
```

3. Add the enclosure to the DSX node, with the array's discovery address as the target. To connect all of the subsystems that the DSX node discovers at the same time, add `--connect-all`:

   Command:

   ```
   nvmeof-config --enclosure-add --dsx-name <dsx-node-name> --target <target-address> --connect-all
   ```

   Give every address that the array presents the subsystems on, by repeating `--target` for each one, when you add the enclosure. Each address becomes a path to each subsystem found there, and multipathing is enabled when a subsystem has more than one. An address that you add to the enclosure later is not connected to subsystems that are already connected; see NVMe-oF Limitations and Operating Guidance. A target is identified by its address; giving the same address again with a different port does not add a target.

   You can use `--dsx-id <dsx-node-id>` instead of `--dsx-name`. Leave out `--connect-all` to add the enclosure without connecting anything yet. The command discovers and connects before it returns, which can take a few minutes; its output shows every subsystem found, with `State: CONNECTED` on each address that was connected.

   Expected output (two subsystems, two targets, trimmed):

   ```
   Type:                    Nvme-of Enclosure
   ID:                      b157dd67-abf8-4327-97ba-b51ac236fc5a
   Name:                    dsx.example.com::nvmeof-b157dd67-abf8-4327-97ba-b51ac236fc5a
   Node:                    dsx.example.com
   Node host NQN:           nqn.2014-08.org.nvmexpress:uuid:63422d42-6bd4-a136-5617-8d8c7b0f9c31
   NVMe-oF config:          Queue depth:             64
                            IO queue pairs:          64
                            I/O policy:              QUEUE_DEPTH
                            Transport:               TCP
                            Target port:             8009
                            Controller loss timeout: -1
   Discovery targets:
                            Address:                 192.0.2.31
                            Port:                    8009
                            Transport:               TCP
   ```


© 2026 - Hammerspace

```
                        Address:                192.0.2.32
                        Port:                   8009
                        Transport:              TCP
Subsystems:
                        Type:                   Nvme-of Subsystem
                        ID:                     40b94f39-4bb2-5fd5-a353-f3afaac1d095
                        Subsystem NQN:          nqn.2026-09.com.example:docs-subsys2
                        Namespaces:
                                                [Namespace ID: 1, Size: 4.2GB, Sector size: 4096,
Model: Docs Lab NVMe Target, Serial: DOCSLAB00000002, Namespace unique ID: bef19a80-60dd-b2bd-30a3-
4965b79d3e20, Connected: true]
                                                [Namespace ID: 2, Size: 4.2GB, Sector size: 4096,
Model: Docs Lab NVMe Target, Serial: DOCSLAB00000002, Namespace unique ID: bef19a80-60dd-b2bd-30a3-
4965b79d3e20-2, Connected: true]
                        Transport addresses:
                                                [Address: 192.0.2.32, Port: 8009, Transport: TCP,
State: CONNECTED]
                                                [Address: 192.0.2.31, Port: 8009, Transport: TCP,
State: CONNECTED]

                        Type:                   Nvme-of Subsystem
                        ID:                     8e1b8ba6-a027-5539-98e9-0bcee62a28c5
                        Subsystem NQN:          nqn.2026-09.com.example:docs-subsys1
                        Namespaces:
                                                [Namespace ID: 1, Size: 4.2GB, Sector size: 4096,
Model: Docs Lab NVMe Target, Serial: DOCSLAB00000001, Namespace unique ID: c9eedb06-110b-552b-beaa-
1cd7e38ae394, Connected: true]
                                                [Namespace ID: 2, Size: 4.2GB, Sector size: 4096,
Model: Docs Lab NVMe Target, Serial: DOCSLAB00000001, Namespace unique ID: c9eedb06-110b-552b-beaa-
1cd7e38ae394-2, Connected: true]
                        Transport addresses:
                                                [Address: 192.0.2.32, Port: 8009, Transport: TCP,
State: CONNECTED]
                                                [Address: 192.0.2.31, Port: 8009, Transport: TCP,
State: CONNECTED]
```

On builds earlier than 5.2.12-P3, `--connect-all` given together with `--target` connects nothing. On those builds, add the enclosure first, and then connect the subsystems as a separate command (step 5).

4. Find the enclosure's name. An enclosure added from the Admin CLI is named after the DSX node, followed by `::nvmeof-` and a unique ID. This is so even if you give `--enclosure-name` with `--enclosure-add`: the name you give is ignored.

   Command:

   ```
   nvmeof-config --list
   ```

   Expected output:

```
total 1
Type:                   Nvme-of Enclosure
ID:                     b157dd67-abf8-4327-97ba-b51ac236fc5a
Name:                   dsx.example.com::nvmeof-b157dd67-abf8-4327-97ba-b51ac236fc5a
Node:                   dsx.example.com
Node host NQN:          nqn.2014-08.org.nvmexpress:uuid:63422d42-6bd4-a136-5617-8d8c7b0f9c31
```

5. If you did not use `--connect-all`, connect the subsystems you want:

   Command:

   ```
   nvmeof-config --enclosure-name <enclosure-name> --connect-nqn <subsystem-nqn>
   ```

   Connecting a subsystem that the DSX node did not discover succeeds but connects nothing.

   Run this command after the enclosure has been added. If you give `--connect-nqn` with `--enclosure-add` instead, and another enclosure on the DSX node already uses the same target address, the subsystem is discovered but not connected (`State: DISCOVERED`); run this command to connect it.

6. Find the device path of each connected namespace. The DSX node's drive list shows every connected namespace as an **Nvme-of Drive** with its device path.

   Command:

   ```
   drive-list --node-name <dsx-node-name>
   ```

   Expected output (trimmed to the NVMe-oF drives):

   ```
   total 10
   Type:                   Hdd
   ID:                     5f2e0a0f-323c-4172-82cc-7ecb0d592698
   Name:                   /dev/sdc
   State:                  OK

   Type:                   Hdd
   ID:                     87de74f9-d96f-4c6d-8c2b-f0e520872076
   Name:                   /dev/dm-1
   State:                  OK

   Type:                   Hdd
   ID:                     90b0f905-07f6-4469-9d16-9b4df73994d4
   Name:                   /dev/dm-2
   State:                  OK

   Type:                   Hdd
   ID:                     960b144f-33ac-4c82-8336-f038d47f5c01
   ```


© 2026 - Hammerspace

```
Name:                   /dev/dm-0
State:                  OK

Type:                   Hdd
ID:                     e17fa6e5-abb1-4d8a-a831-44f6951f60b9
Name:                   /dev/sdb
State:                  OK

Type:                   Hdd
ID:                     e553917a-c252-4b26-a910-785dc31c1015
Name:                   /dev/sda
State:                  OK

Type:                   Nvme-of Drive
ID:                     29f32ef8-896b-5689-9b17-96a6bcbf0e2a
Name:                   /dev/nvme0n1
State:                  OK
Capacity:               4.2GB
Serial number:          29f32ef8-896b-5689-9b17-96a6bcbf0e2a

Type:                   Nvme-of Drive
ID:                     4412c29a-ffd7-5961-a4c0-61a8b509e0c5
Name:                   /dev/nvme2n1
State:                  OK
Capacity:               4.2GB
Serial number:          4412c29a-ffd7-5961-a4c0-61a8b509e0c5

Type:                   Nvme-of Drive
ID:                     863571b4-59f2-5e51-9e30-4c2534ee9981
Name:                   /dev/nvme2n2
State:                  OK
Capacity:               4.2GB
Serial number:          863571b4-59f2-5e51-9e30-4c2534ee9981

Type:                   Nvme-of Drive
ID:                     88073fd2-9dcf-5fab-952a-08abbabcab63
Name:                   /dev/nvme0n2
State:                  OK
Capacity:               4.2GB
Serial number:          88073fd2-9dcf-5fab-952a-08abbabcab63
```

Read the device path right before you use it. Device paths such as `/dev/nvme0n1` are assigned by the DSX node's kernel and can change when a subsystem is disconnected and reconnected.

7. Create a logical volume on a namespace, and then add it to Hammerspace as a storage volume. The logical volume's name is in the output of the first command.

   Command:

© 2026 - Hammerspace

```
logical-volume-create --node-name <dsx-node-name> --device-path <device-path>
```

Expected output:

```
Type:                   Logical Volume
ID:                     eea406dd-ddb8-4416-886f-61f6ebcf09ce
Name:                   /hsvol2
Exported path:          /hsvol2
Addresses:
                        [IP: 192.0.2.21/20, Port: 3049, NetId: tcp, NodeNum: 0]
Node:                   dsx.example.com
Usage:                  DS
Reserved:               false
Filesystem type:        XFS
Capacity:               [Total: 4.2GB]
Hardware:
                        [Type: Nvme-of Drive, ID: 29f32ef8-896b-5689-9b17-96a6bcbf0e2a, Name:
/dev/nvme0n1, State: OK]
```

```
volume-add --node-name <dsx-node-name> --logical-volume-name <logical-volume-name> --name <volume-
name>
```

The Management GUI does both in one flow; see Adding an NVMe-oF Enclosure and Subsystems.

8. To remove the namespaces of a disconnected or removed subsystem from the DSX node's inventory, refresh the node:

   Command:

   ```
   node-refresh --name <dsx-node-name> --reconcile-components
   ```

A.1.4. Additional Admin CLI Examples

Every example on this page uses the `nvmeof-config` command. Settings, connections, and disconnections apply to one enclosure at a time, so each command names the enclosure with `--enclosure-name` or `--enclosure-id`. To find an enclosure's name or ID, list the enclosures first.

Listing Enclosures and Connections

To list every NVMe-oF enclosure on every DSX node, with the DSX node and its host NQN, run:

Command:

```
nvmeof-config --list
```


© 2026 - Hammerspace

Running `nvmeof-config` with no options does the same.

To include each enclosure's connection settings, discovery targets, and subsystems, with their namespaces and target addresses, add `--full`:

Command:

```
nvmeof-config --list --full
```

The **State** of a target address shows whether Hammerspace connected the subsystem, not whether the path is working. See NVMe-oF Path Health and Recovery.

Expected output of `nvmeof-config --list` (one enclosure) and of `--list --full`:

```
total 1
Type:                   Nvme-of Enclosure
ID:                     b157dd67-abf8-4327-97ba-b51ac236fc5a
Name:                   dsx.example.com::nvmeof-b157dd67-abf8-4327-97ba-b51ac236fc5a
Node:                   dsx.example.com
Node host NQN:          nqn.2014-08.org.nvmexpress:uuid:63422d42-6bd4-a136-5617-8d8c7b0f9c31
```

```
total 1
Type:                   Nvme-of Enclosure
ID:                     c32efdee-d04e-4626-9c8d-a807e35d2579
Name:                   dsx.example.com::nvmeof-c32efdee-d04e-4626-9c8d-a807e35d2579
Node:                   dsx.example.com
Node host NQN:          nqn.2014-08.org.nvmexpress:uuid:63422d42-6bd4-a136-5617-8d8c7b0f9c31
NVMe-oF config:         Queue depth:             64
                        IO queue pairs:          64
                        I/O policy:              QUEUE_DEPTH
                        Transport:               TCP
                        Target port:             8009
                        Controller loss timeout: -1
Discovery targets:
                        Address:                 192.0.2.31
                        Port:                    8009
                        Transport:               TCP

                        Address:                 192.0.2.32
                        Port:                    8009
                        Transport:               TCP
Subsystems:
                        Type:                    Nvme-of Subsystem
                        ID:                      8e1b8ba6-a027-5539-98e9-0bcee62a28c5
                        Subsystem NQN:           nqn.2026-09.com.example:docs-subsys1
                        Namespaces:
                                                 [Namespace ID: 2, Size: 4.2GB, Sector size: 4096,
```


© 2026 - Hammerspace

```
Model: Docs Lab NVMe Target, Serial: DOCSLAB00000001, Namespace unique ID: c9eedb06-110b-552b-beaa-
1cd7e38ae394, Connected: true]
                                                [Namespace ID: 1, Size: 4.2GB, Sector size: 4096,
Model: Docs Lab NVMe Target, Serial: DOCSLAB00000001, Namespace unique ID: c9eedb06-110b-552b-beaa-
1cd7e38ae394-2, Connected: true]
                        Transport addresses:
                                                [Address: 192.0.2.31, Port: 8009, Transport: TCP,
State: CONNECTED]
                                                [Address: 192.0.2.32, Port: 8009, Transport: TCP,
State: CONNECTED]

                        Type:                   Nvme-of Subsystem
                        ID:                     40b94f39-4bb2-5fd5-a353-f3afaac1d095
                        Subsystem NQN:          nqn.2026-09.com.example:docs-subsys2
                        Namespaces:
                                                [Namespace ID: 1, Size: 4.2GB, Sector size: 4096,
Model: Docs Lab NVMe Target, Serial: DOCSLAB00000002, Namespace unique ID: bef19a80-60dd-b2bd-30a3-
4965b79d3e20, Connected: false]
                                                [Namespace ID: 2, Size: 4.2GB, Sector size: 4096,
Model: Docs Lab NVMe Target, Serial: DOCSLAB00000002, Namespace unique ID: bef19a80-60dd-b2bd-30a3-
4965b79d3e20-2, Connected: false]
                        Transport addresses:
                                                [Address: 192.0.2.31, Port: 8009, Transport: TCP,
State: DISCOVERED]
                                                [Address: 192.0.2.32, Port: 8009, Transport: TCP,
State: DISCOVERED]
```

The **Namespace unique ID** and the **Serial number** that `drive-list` shows for an NVMe-oF drive are identifiers that Hammerspace derives for each namespace. They are not the array's serial number or the namespace's NGUID.

Adding Targets

A target is an address where the DSX node discovers the array's subsystems, optionally with a port and a transport. When you add a target to an enclosure, Hammerspace discovers the enclosure's subsystems again at that address.

The targets you give on the command line replace the enclosure's current list of targets. To add a target, list every target the enclosure should keep as well as the new one:

Command:

```
nvmeof-config --enclosure-name <enclosure-name> --target <existing-target-address> --target <new-target-
address>
```

Adding a target does not add a path to subsystems that are already connected. The listing shows the new address as `CONNECTED` for those subsystems, but the DSX node has not connected to it. To connect them on the new address as well, disconnect and reconnect them (`--disconnect-all` and then `--connect-all`), which is

 © 2026 - Hammerspace

possible only while none of their namespaces has a logical volume. Give every address when you add the enclosure instead.

Removing a target from the list does not disconnect subsystems that are already connected.

Each `--target` value is an IP address or host name, optionally followed by a port and a transport, separated by commas. For example: `--target 10.10.1.1`, `--target 10.10.1.1,8009`, or `--target nvme2.local,1234,TCP`. The default discovery port is 8009 and the default transport is TCP. Hammerspace identifies a target by its address and transport: the same address with a different port is the same target, so it is not added and the port is not changed.

Changing an Enclosure's Connection Settings

Each enclosure has its own connection settings. The Admin CLI changes them for one enclosure at a time. The **NVMe-oF Config** tab of the Management GUI's Edit Storage System wizard sets them for the whole DSX node: a change there updates every enclosure on that node.

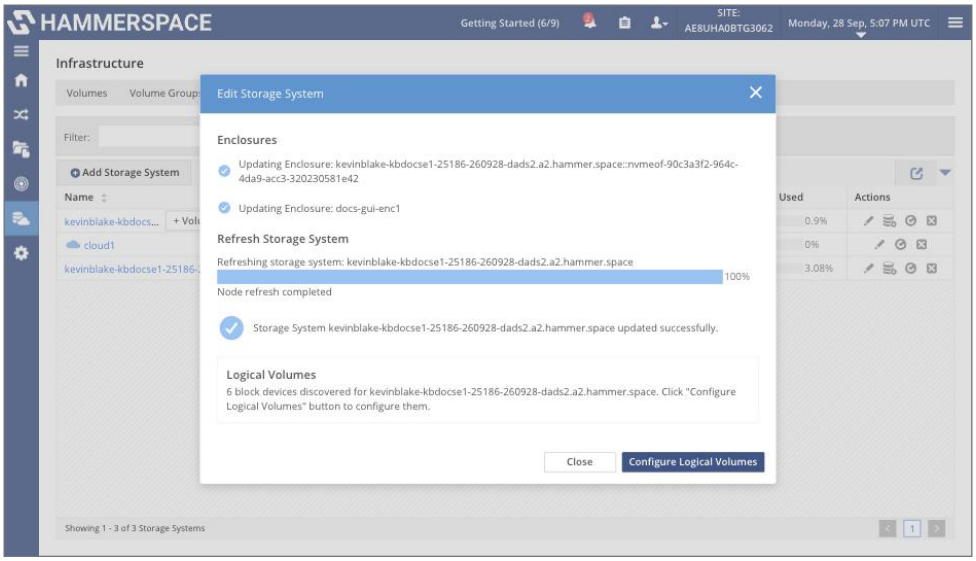


Figure 93. A queue-depth change from the GUI updates both enclosures on the node

To change one or more settings for one enclosure, name the enclosure and give the new values:

Command:

```
nvmeof-config --enclosure-name <enclosure-name> --io-policy ROUND_ROBIN
```

Command:

```
nvmeof-config --enclosure-name <enclosure-name> --io-queue-pairs 32
```

The settings and their permitted values are:

Table 17. NVMe-oF connection settings

© 2026 - Hammerspace

Option	Values
`--io-policy`	`QUEUE_DEPTH` (the default), `ROUND_ROBIN`, or `NUMA`
`--io-queue-pairs`	1 to 128. The default is 64. The DSX node opens at most one I/O queue per CPU core to each controller, so a value above the node's core count has no further effect.
`--ctrl-loss-tmo`	0 to 3600 seconds from the Admin CLI, which rejects `-1`. Set `-1`, the default, in the Management GUI. See Controller Loss Timeout.
`--queue-depth`	1 to 1024. The default is 64. This release stores the value but does not apply it to the connection.
`--target-port`	The discovery port used for targets that you add later without a port. The default is 8009.
`--transport`	The transport used for targets that you add later without a transport: `TCP`, the default. TCP is the only supported transport.

A value outside its range is rejected before anything changes, for example `queueDepth must be between 1 and 1024`. The policy and transport names are accepted in any letter case.

Changing any setting except the I/O policy disconnects and reconnects the controllers of the enclosure's connected subsystems. Changing only the I/O policy does not reconnect anything.

On 5.2.12-P5, 5.2.13, and later, volumes stay mounted through the reconnect, but they may briefly show the **Suspected** operational state. On earlier builds, any change can leave volumes unusable until the DSX node restarts. See Controller Loss Timeout.

Connecting Subsystems

Connecting a subsystem connects all of its namespaces to the DSX node that owns the enclosure. The subsystem must be one that the enclosure discovered. `nvmeof-config --list --full` shows the enclosure's subsystems.

To connect one subsystem, run:

Command:

```
nvmeof-config --enclosure-name <enclosure-name> --connect-nqn <subsystem-nqn>
```

To connect every subsystem that the enclosure discovered, run:

Command:

```
nvmeof-config --enclosure-name <enclosure-name> --connect-all
```


© 2026 - Hammerspace

If the NQN is not one that the enclosure discovered, the command succeeds but connects nothing. Check the result with `nvmeof-config --list --full`.

You can connect some subsystems and disconnect others in one command by combining `--connect-nqn` and `--disconnect-nqn`. You cannot combine `--connect-all` or `--disconnect-all` with any other connect or disconnect option.

Connect each subsystem to one DSX node only. If the array presents a subsystem to two DSX nodes, Hammerspace lets both connect it, but it refuses to create a logical volume on a namespace of a subsystem that is already connected to another node: `One or more of the specified NVMe-oF subsystems is already connected to another node`. Use the array's host access configuration to present each namespace to one DSX node.

You do not need to refresh the DSX node after connecting; the connected namespaces appear in `drive-list --node-name <dsx-node-name>` as soon as the command returns.

Disconnecting Subsystems

Disconnecting a subsystem disconnects all of its namespaces from the DSX node. You cannot disconnect a subsystem while any of its namespaces has a logical volume. Remove the Hammerspace volume and delete the logical volume first.

To disconnect one subsystem, run:

Command:

```
nvmeof-config --enclosure-name <enclosure-name> --disconnect-nqn <subsystem-nqn>
```

To disconnect every subsystem of the enclosure, run:

Command:

```
nvmeof-config --enclosure-name <enclosure-name> --disconnect-all
```

Disconnecting through Hammerspace normally does not raise an NVMe-oF path event.

Removing an NVMe-oF Enclosure

Removing an enclosure disconnects its connected subsystems and removes the enclosure from the DSX node. You cannot remove an enclosure while any of its subsystems' namespaces has a logical volume.

Command:

```
nvmeof-config --enclosure-remove --enclosure-name <enclosure-name>
```

Expected output:

© 2026 - Hammerspace

```
success
```

While a logical volume exists on one of the enclosure's namespaces, the command fails.

Error:

```
nvmeof-config: Unable to disconnect NVMe-oF subsystem(s), one or more subsystems have active volumes.
Subsystem(s)=[nqn.2026-09.com.example:docs-subsys2].
```

The removed enclosure's namespaces stay in the DSX node's inventory until you refresh the node with `node-refresh --name <dsx-node-name> --reconcile-components`.

Creating a Logical Volume

After a subsystem is connected, create a logical volume on each namespace you want to use, and then add the logical volume to Hammerspace as a volume. The Management GUI shows the DSX node's block devices and creates the logical volumes for you. See Adding an NVMe-oF Enclosure and Subsystems.

Creating a logical volume formats the device and erases any data on it. Never create a logical volume on a device whose data you need.

To create one from the Admin CLI, find the namespace's device path in the DSX node's drive list, where each connected namespace is an **Nvme-of Drive**, and then create the logical volume on it. Read the path right before you use it: the kernel assigns device paths such as `/dev/nvme0n1`, and they can change when a subsystem is disconnected and reconnected.

Command:

```
drive-list --node-name <dsx-node-name>
```

```
logical-volume-create --node-name <dsx-node-name> --device-path <device-path>
```

Expected output:

```
Type:                   Logical Volume
ID:                     eea406dd-ddb8-4416-886f-61f6ebcf09ce
Name:                   /hsvol2
Exported path:          /hsvol2
Addresses:
                        [IP: 192.0.2.21/20, Port: 3049, NetId: tcp, NodeNum: 0]
Node:                   dsx.example.com
Usage:                  DS
Reserved:               false
Filesystem type:        XFS
Capacity:               [Total: 4.2GB]
```


© 2026 - Hammerspace

```
Hardware:
                          [Type: Nvme-of Drive, ID: 29f32ef8-896b-5689-9b17-96a6bcbf0e2a, Name:
/dev/nvme0n1, State: OK]
```

The logical volume is not yet a Hammerspace storage volume. Add it with `volume-add --node-name <dsx-node-name> --logical-volume-name <logical-volume-name> --name <volume-name>`; see Admin CLI Workflow to Connect NVMe-oF Enclosures.

A.1.5. NVMe-oF Path Health and Recovery

Starting with Hammerspace 5.2.14, each DSX node monitors every path to each NVMe-oF subsystem that it is connected to. A path is one connection from the DSX node to one target address of the subsystem. A subsystem that the array presents on two target addresses has two paths to each DSX node that connects to it.

When a path stops working, the DSX node raises an event and keeps trying to reconnect the path. When the target answers again, the DSX node reconnects the path and clears the event. You do not need to reconnect the path yourself.

Path Health Events

Table 18. NVMe-oF path health events

Event	Severity	Meaning
`NVMEOF_PATH_DEGRADED`	Warning	A path has been trying to reconnect for a while, and the subsystem still has at least one working path. I/O continues on the working paths with reduced redundancy.
`NVMEOF_PATH_LOST`	Warning	The connection for a path no longer exists, and the subsystem still has at least one working path. I/O continues on the working paths with reduced redundancy. This happens, for example, when the enclosure's controller loss timeout is a finite value and has expired, or when a path did not connect when the DSX node started.
`NVMEOF_PATH_LOST`	Critical	The subsystem has had no working path from this DSX node for a short time, whether its connections are still trying to reconnect or no longer exist. I/O to volumes on the subsystem's namespaces waits until a path is restored. If the enclosure's controller loss timeout is a finite value and it expires first, that I/O fails instead; see Controller Loss Timeout.

Each event names the path it is about, in this order: the subsystem NQN, the target address, the target port, the transport, and the host name of the DSX node. One event is raised for each path of each subsystem, so a subsystem with two paths on a target that goes down raises two events. For example:

```
NVMe-oF path lost: nqn=<subsystem-nqn>, traddr=<target-address>, trsvcid=<target-port>, transport=tcp,
node=<dsx-host-name>
```


© 2026 - Hammerspace

The source of both events is the DSX node that lost the path. In the **Events** list, the **Source type** column shows **Node**.

Expected output of `event-list --type NVMEOF_PATH_DEGRADED` with one path of a two-path target down:

```
total 2
ID:                     18e0d25a-bad1-11f1-9c79-005056ad0a7b
Created:                2026-09-28 00:11:05 UTC
Modified:               2026-09-28 00:11:05 UTC
Type:                   NVMEOF_PATH_DEGRADED
Description:            NVMe-oF path degraded (controller stuck connecting): nqn=nqn.2026-
09.com.example:docs-subsys2, traddr=192.0.2.31, trsvcid=8009, transport=tcp, node=dsx.example.com
Severity:               WARNING
Source type:            Node
Source ID:              4cdae232-a7de-501a-ad2a-1e90457a0f98
Source name:            dsx.example.com
Cleared:                false

ID:                     190e525c-bad1-11f1-9c79-005056ad0a7b
Created:                2026-09-28 00:11:05 UTC
Modified:               2026-09-28 00:11:05 UTC
Type:                   NVMEOF_PATH_DEGRADED
Description:            NVMe-oF path degraded (controller stuck connecting): nqn=nqn.2026-
09.com.example:docs-subsys1, traddr=192.0.2.31, trsvcid=8009, transport=tcp, node=dsx.example.com
Severity:               WARNING
Source type:            Node
Source ID:              4cdae232-a7de-501a-ad2a-1e90457a0f98
Source name:            dsx.example.com
Cleared:                false
```

How the Events Behave

- A brief interruption does not raise an event. The DSX node waits to see whether a path recovers on its own before it reports it.
- When the subsystem's last working path fails, the events change to Critical. A `NVMEOF_PATH_LOST` event at Warning rises to Critical in place. A `NVMEOF_PATH_DEGRADED` event is cleared and replaced by a `NVMEOF_PATH_LOST` event at Critical for the same path.
- When any path to the subsystem works again, the events for the paths that are still down step back down to Warning: a `NVMEOF_PATH_LOST` event for a path whose connection no longer exists returns to Warning in place, and a Critical event for a path that is still trying to reconnect is cleared and replaced by a `NVMEOF_PATH_DEGRADED` event at Warning.
- When the path works again, the event clears automatically, usually within ten seconds.
- Disconnecting a subsystem or removing an enclosure through Hammerspace normally does not raise these events, and any event that is active for those paths is cleared.
- The DSX node monitors only subsystems that were connected through Hammerspace, by using the Management GUI or the `nvmeof-config` command.

© 2026 - Hammerspace

Responding to a Path Health Event

For a Warning event, check the network path, the DSX node's network interface, and the array port for the target address named in the event. Data remains available through the subsystem's other paths.

For a Critical event, treat the problem as urgent and restore connectivity between the DSX node and the array. As soon as any path to the subsystem answers again, the DSX node reconnects it and the event clears.

Do not disconnect and reconnect the subsystem to recover a path. Hammerspace reconnects the path automatically, and you cannot disconnect a subsystem whose namespaces have logical volumes.

SNMP Traps

If SNMP trap forwarding is configured, each event is also sent as an SNMP trap:

- `NvmeofPathDegraded`, notification 128
- `NvmeofPathLost`, notification 129

Each trap carries the event type, severity, and description, along with the event's other standard fields.

Both traps are defined in the Hammerspace MIB (`HAMMERSPACE-MIB`), which you can download from the Management GUI: go to **Administration › SNMP** and click **Download MIB File**. Their full OIDs are `1.3.6.1.4.1.47219.1.3.0.128` (`NvmeofPathDegraded`) and `1.3.6.1.4.1.47219.1.3.0.129` (`NvmeofPathLost`).

Connection State Is Not Path Health

The `nvmeof-config --list --full` output shows a **State** for each target address of each subsystem:

`DISCOVERED`
: The subsystem was found at this address, but Hammerspace has not connected it.

`CONNECTED`
: Hammerspace connected the subsystem.

The State field records what Hammerspace was asked to do. It does not show whether a path is working right now. A path that has gone down still shows `CONNECTED`, and so does every address of a subsystem that is connected on any of them. The `Connected` value of each namespace and the `State` of each drive in `drive-list` are the same kind of record.

To find out whether paths are healthy, use the NVMe-oF path health events described at the start of this page.

Expected output of `nvmeof-config --list --full` with one subsystem connected and one only discovered:

```
total 1
Type:                   Nvme-of Enclosure
ID:                     c32efdee-d04e-4626-9c8d-a807e35d2579
Name:                   dsx.example.com::nvmeof-c32efdee-d04e-4626-9c8d-a807e35d2579
Node:                   dsx.example.com
```


© 2026 - Hammerspace

```
Node host NQN:           nqn.2014-08.org.nvmexpress:uuid:63422d42-6bd4-a136-5617-8d8c7b0f9c31
NVMe-oF config:          Queue depth:            64
                         IO queue pairs:         64
                         I/O policy:             QUEUE_DEPTH
                         Transport:              TCP
                         Target port:            8009
                         Controller loss timeout: -1
Discovery targets:
                         Address:                192.0.2.31
                         Port:                   8009
                         Transport:              TCP

                         Address:                192.0.2.32
                         Port:                   8009
                         Transport:              TCP
Subsystems:
                         Type:                   Nvme-of Subsystem
                         ID:                     8e1b8ba6-a027-5539-98e9-0bcee62a28c5
                         Subsystem NQN:          nqn.2026-09.com.example:docs-subsys1
                         Namespaces:
                                                 [Namespace ID: 2, Size: 4.2GB, Sector size: 4096,
Model: Docs Lab NVMe Target, Serial: DOCSLAB00000001, Namespace unique ID: c9eedb06-110b-552b-beaa-
1cd7e38ae394, Connected: true]
                                                 [Namespace ID: 1, Size: 4.2GB, Sector size: 4096,
Model: Docs Lab NVMe Target, Serial: DOCSLAB00000001, Namespace unique ID: c9eedb06-110b-552b-beaa-
1cd7e38ae394-2, Connected: true]
                         Transport addresses:
                                                 [Address: 192.0.2.31, Port: 8009, Transport: TCP,
State: CONNECTED]
                                                 [Address: 192.0.2.32, Port: 8009, Transport: TCP,
State: CONNECTED]

                         Type:                   Nvme-of Subsystem
                         ID:                     40b94f39-4bb2-5fd5-a353-f3afaac1d095
                         Subsystem NQN:          nqn.2026-09.com.example:docs-subsys2
                         Namespaces:
                                                 [Namespace ID: 1, Size: 4.2GB, Sector size: 4096,
Model: Docs Lab NVMe Target, Serial: DOCSLAB00000002, Namespace unique ID: bef19a80-60dd-b2bd-30a3-
4965b79d3e20, Connected: false]
                                                 [Namespace ID: 2, Size: 4.2GB, Sector size: 4096,
Model: Docs Lab NVMe Target, Serial: DOCSLAB00000002, Namespace unique ID: bef19a80-60dd-b2bd-30a3-
4965b79d3e20-2, Connected: false]
                         Transport addresses:
                                                 [Address: 192.0.2.31, Port: 8009, Transport: TCP,
State: DISCOVERED]
                                                 [Address: 192.0.2.32, Port: 8009, Transport: TCP,
State: DISCOVERED]
```


© 2026 - Hammerspace

Controller Loss Timeout

The controller loss timeout sets how long a DSX node keeps trying to reconnect a lost connection to an NVMe-oF controller before the node's kernel gives up and removes the controller. It is one of the connection settings of an enclosure. An enclosure is Hammerspace's record of one NVMe-oF array as a DSX node sees it: the array's discovery addresses, the subsystems found there, and the settings used to connect to them. Each enclosure has its own value.

The value is either `-1` or a number of seconds from `0` to `3600`. The value `-1` means that the DSX node never gives up trying to reconnect, and it is the recommended setting. Do not set `0`: with `0`, the kernel removes the controller as soon as a path fails.

Enclosures added on Hammerspace 5.2.12-P4, 5.2.13, or later use `-1` by default. Enclosures added on earlier builds keep the value they were created with, even after a software update. Builds earlier than 5.2.12-P4 do not accept `-1`.

Why -1 Is Recommended

Starting with Hammerspace 5.2.14, Hammerspace reconnects a lost path automatically when the target answers again, whatever the timeout is set to. The timeout still matters, because it decides what happens to the block device while the path is down:

- With `-1`, the NVMe block device stays attached while the DSX node keeps trying to reconnect. If every path to a subsystem is down, I/O to volumes on that subsystem waits until a path returns, and then completes. The storage volume stays online.
- With a finite value, the kernel removes the controller when the timeout expires. If that was the subsystem's last working path, the storage volume on the affected namespace goes to the **Suspected** operational state, I/O that is waiting fails, and the file system on the volume stops serving data. When a path returns, Hammerspace reconnects the path, but the volume stays **Suspected**. Contact Hammerspace Support to recover the volume.

Checking the Current Value

Command:

```
nvmeof-config --list --full
```

In the output, each enclosure's NVMe-oF configuration shows its **Controller loss timeout**.

Expected output (one enclosure, trimmed to the configuration block):

```
total 1
Type:                   Nvme-of Enclosure
ID:                     16a6c135-04c4-4e9e-8dac-f7eb4566b5e3
Name:                   dsx.example.com::nvmeof-16a6c135-04c4-4e9e-8dac-f7eb4566b5e3
Node:                   dsx.example.com
Node host NQN:          nqn.2014-08.org.nvmexpress:uuid:63422d42-6bd4-a136-5617-8d8c7b0f9c31
NVMe-oF config:         Queue depth:            64
```


© 2026 - Hammerspace

```
                        IO queue pairs:          64
                        I/O policy:              QUEUE_DEPTH
                        Transport:               TCP
                        Target port:             8009
                        Controller loss timeout: -1
<Additional output removed for brevity>
```

Setting the Value

Set the timeout in the Management GUI. The **Controller lost timeout** field is on the **NVMe-oF Config** tab that opens when you click the pencil icon of the DSX node on the **Storage Systems** tab of the **Infrastructure** page. Enter the value and click **Update Storage System**. A value saved there applies to every NVMe-oF enclosure on that DSX node, not just one; the tab shows the values of the node's first enclosure.

The Admin CLI accepts only a value from `0` to `3600` for `--ctrl-loss-tmo`; it rejects `-1` with a syntax error. To set a finite timeout on one enclosure, replace `<enclosure-name>` with the enclosure's name as shown by `nvmeof-config --list`. You can use `--enclosure-id <enclosure-uuid>` instead.

Command:

```
nvmeof-config --ctrl-loss-tmo <seconds> --enclosure-name <enclosure-name>
```

Once you have set a finite value from the Admin CLI, use the Management GUI to return to `-1`.

Changing the timeout, or any other connection setting except the I/O policy, disconnects and reconnects the controllers of the enclosure's connected subsystems, because these settings cannot be changed on a live connection. Changing only the I/O policy does not reconnect anything.

On 5.2.12-P5, 5.2.13, and later, volumes stay mounted through the reconnect, but they may briefly show the **Suspected** operational state. Make the change in a maintenance window anyway.

On builds earlier than 5.2.12-P5 and 5.2.13, changing any NVMe-oF setting on an enclosure that has volumes, including the I/O policy, can leave those volumes unusable until the DSX node is restarted. Update to 5.2.12-P5, 5.2.13, or later before you change a setting.

A.1.6. NVMe-oF Limitations and Operating Guidance

Presenting Storage to DSX Nodes

- Several DSX nodes can connect to the same array. Add a separate enclosure on each DSX node. Each of those enclosures lists the same subsystems and namespaces.
- Only one logical volume can use a given namespace. Once one DSX node has created a logical volume on a namespace, creating another on it fails on every DSX node with `One or more of the specified NVMe-oF subsystems is already connected to another node`. That message means the namespace is already in use by a logical volume.
- Hammerspace cannot control hosts outside the cluster. If the array presents a namespace to any other

© 2026 - Hammerspace

host, that host can write to it and corrupt the file system. Use the array's host access configuration to present namespaces only to the DSX nodes' host NQNs.

- To change an enclosure's target addresses, run `nvmeof-config --target`. The addresses you give replace the enclosure's list, so include every address you want to keep, including addresses that are already connected. Any address you leave out is removed from the enclosure. Example: `nvmeof-config --enclosure-name <enclosure-name> --target 10.0.0.10 --target 10.0.0.11`
- Give the enclosure every address when you add it. A subsystem that is not yet connected uses all of the enclosure's addresses when you connect it. A subsystem that is already connected does not use an address added later, even though the enclosure lists that address as `CONNECTED`. Removing an address does not disconnect existing paths on it. To change the addresses of a subsystem that is already connected, contact Hammerspace Support.
- Do not present a clone or snapshot of a namespace to the same DSX node as the original. Two namespaces on one DSX node must not carry the same file system UUID; if they do, Hammerspace cannot tell them apart and may fail to mount a volume on either, with `Filesystem UUID <uuid> matches multiple NVMe-oF devices`. A clone presented to a different DSX node is not affected.
- If a connection to a subsystem fails with a general error, check that the array presents that subsystem to the DSX node's host NQN. The error does not say that access was refused.

The DSX Host NQN

- Each DSX node generates its host NQN once, when it is first configured, and you cannot change it. Never copy one DSX node's host NQN to another host or edit it. Two hosts that present the same NQN are not detected and are not supported.
- Reinstalling a DSX node gives it a new host NQN. Update the array's host access configuration with the new NQN before you reconnect the node's enclosures.

To read a DSX node's host NQN, see Admin CLI Workflow to Connect NVMe-oF Enclosures.

Connecting and Disconnecting

- Hammerspace connects and disconnects whole subsystems. You cannot disconnect a single namespace.
- You cannot disconnect a subsystem while any of its namespaces has a logical volume. Remove the Hammerspace volume and delete the logical volume first. Otherwise the command fails with the message `Unable to disconnect NVMe-oF subsystem(s), one or more subsystems have active volumes.`
- Right after a subsystem is connected, a namespace can take a short time to appear. If a namespace you expect is missing, repeat the operation.
- Changing any connection setting except the I/O policy briefly disconnects and reconnects the enclosure's controllers. Volumes remain, but they may briefly show the **Suspected** operational state. Make the change in a maintenance window. See Controller Loss Timeout.

A.1.7. Upgrading a Cluster That Uses NVMe-oF Storage

If any DSX node in the cluster uses NVMe-oF storage, follow this guidance in addition to the software update procedure in Upgrading Hammerspace in the *Hammerspace Installation and Licensing Guide*.

© 2026 - Hammerspace

Before You Upgrade

1. Make sure that every NVMe-oF path is healthy. If the cluster runs 5.2.14 or later, list the NVMe-oF path events that have not been cleared. In the Management GUI, open **Events**, select **Uncleared events only**, and look in the **Action** column for **NVMe-oF path degraded** and **NVMe-oF path lost**. From the Admin CLI, run `event-list --type NVMEOF_PATH_DEGRADED --uncleared`, and then the same command with `--type NVMEOF_PATH_LOST`. Do not start the upgrade while either event is active; restore the path first. Earlier builds do not raise these events, so ask Hammerspace Support to check path health before you upgrade.

   Do not use the **State** shown by `nvmeof-config --list --full` for this check. It shows `CONNECTED` for a path that is down. See NVMe-oF Path Health and Recovery.

2. Upgrade to 5.2.14P3 or later. These builds include fixes for clusters that use NVMe-oF storage. Upgrade directly to the build you choose; there is no benefit in stopping at intermediate builds.
3. Plan to have out-of-band access to every node during the upgrade, for example through the server's BMC or through vCenter for a virtual machine. A DSX node that uses NVMe-oF storage can stop responding while it restarts at the end of the upgrade, and then it needs a reset from outside the operating system.
4. Do not change the NVMe-oF configuration while the upgrade is running. That includes adding or removing enclosures, connecting or disconnecting subsystems, and changing connection settings. Some nodes run the new release and some the old one until the upgrade finishes.

After You Upgrade

1. Check that every volume on NVMe-oF storage is online, and that no `NVMEOF_PATH_DEGRADED` or `NVMEOF_PATH_LOST` event is active for a path you expect to work.
2. If a volume on NVMe-oF storage is missing or offline, contact Hammerspace Support.

Never re-create a missing volume to recover it. Without **Force Initialize** (`logical-volume-create --force` in the Admin CLI), the operation fails because the device already holds a file system; with it, the operation erases the data on the device. Contact Hammerspace Support instead.

1. Check the controller loss timeout of each enclosure. Enclosures keep the value they had before the upgrade. See Controller Loss Timeout.

© 2026 - Hammerspace

Appendix B: Event Listing

The following table list all of the events that can be triggered on the system. The numbers in the Message column are simply placeholders for other strings that may be passed (things like entity names).

Table 19. Hammerspace events

Event Name	Type	Message
NONE	DEBUG	None
CREATED	INFORMATIONAL	{0} {1} created by {2}{3}
ADDED	INFORMATIONAL	{0} {1} added by {2}{3}
CANCELLED	INFORMATIONAL	{0} {1} cancelled by {2}{3}
UPDATED	INFORMATIONAL	{0} {1} updated by {2}{3}
DELETED	INFORMATIONAL	{0} {1} deleted by {2}{3}
REMOVED	INFORMATIONAL	{0} {1} removed by {2}{3}
CREATE_FAILED	ERROR	Failed to create {0} {1} by {2}{3}
ADD_FAILED	ERROR	Failed to add {0} {1} by {2}{3}
UPDATE_FAILED	ERROR	Failed to update {0} {1} by {2}{3}
DELETE_FAILED	ERROR	Failed to delete {0} {1} by {2}{3}
REMOVE_FAILED	ERROR	Failed to remove {0} {1} by {2}{3}
NO_AVAILABLE_STORAGE_ON_NODE	ERROR	Node {0} has no available storage
COMPONENT_MISSING	NOTICE	Component {0} {1} is missing in the node {2}. If this is expected, node refresh may be used to reconcile the missing component
SOFTWARE_VERSION_CHANGED	INFORMATIONAL	Software version has changed from {0} to {1} on node {2}
SOFTWARE_UPDATE_STARTED	INFORMATIONAL	Updating {0}to {1}
SOFTWARE_UPDATE_SUCCESS	INFORMATIONAL	Updated {0}to {1}
SOFTWARE_UPDATE_FAILED		Failed to update {0}to {1}
LICENSE_STATE_CHANGED	INFORMATIONAL	The license state changed
LICENSE_ACTIVATED	INFORMATIONAL	License activated successfully
LICENSE_ACTIVATION_FAILED	ERROR	License activation failed{0}

© 2026 - Hammerspace

EVAL_ABOUT_TO_EXPIRE	WARNING	Your evaluation will end within {0} days, please contact support
EVAL_EXPIRED	ERROR	A license is required to operate beyond {0} days after installation. Protocol access has been disabled. Please contact support to resolve the situation
LICENSE_ABOUT_TO_EXPIRE	WARNING	Your {2} license is expiring within {1} days, in order to maintain product functionality and not lose protocol access, please contact support
LICENSE_EXPIRED	ERROR	Your {1} license has expired. The product functionality has been reduced, you can no longer access your data over supported protocols. Please contact support immediately to re-enable access
EXPIRED_LICENSE_IN_GRACE	WARNING	Your expired {3} license has entered a {1} day grace period. There are {2} days remaining before you can no longer access your data over supported protocols. Please contact support to obtain a new license
METERED_LICENSE_IN_GRACE		Your metered license with activation ID {0} which is reporting usage to {1} has entered a {2} day grace period due to successive failures to communicate with the license server. There are {3} days remaining before you can no longer access your data over supported protocols. If communication with the license server is re-established within this period your system will automatically return to normal operation. If the condition persists, please contact support
LICENSE_IN_GRACE_ABOUT_TO_EXPIRE	WARNING	Your expired {3} license has {1} days remaining in a {2} day grace period. Please contact support to obtain a new license

© 2026 - Hammerspace

LICENSE_ABOUT_TO_ENTER_GRACE	WARNING	Your {3} license will enter a grace period in {1} days. After that, you will have {2} days to renew your license
LICENSING_MISCONFIGURED		Your licensing is improperly configured. When using capacity-based licensing you need either a File license or both a File and Object license. System protocols are being disabled. Please contact support.
BORROWING_LICENSE_EXPIRED		Your {1} license has expired. All used {1} storage is now counted against your licensed {2}
BORROWING_LICENSE_ABOUT_TO_EXPIRE		Your {1} license is expiring soon, after which time all used {3} storage will be counted against your licensed {2}
ACTUAL_ALLOCATED_CAPACITY_EXCEEDED		You have exceeded the {1}TB of licensed {2}. You are borrowing {3} from {4}
ALLOCATED_CAPACITY_EXCEEDED	ERROR	You have exceeded the {1}TB of licensed {3} and have entered a grace period. You have {2} days remaining to reduce storage usage
ALLOCATED_CAPACITY_APPROACHING_LIMIT	WARNING	You are currently using {1}% of the {2}TB of licensed {3}
OVER_CAPACITY_GRACE_ABOUT_TO_EXPIRE	WARNING	The over-capacity grace period is nearing expiration. You have {2} days remaining to reduce storage usage
OVER_CAPACITY_LICENSE_VIOLATION		You have exceeded the {1}TB of licensed {3} for over {2} days. System protocols are being disabled
USER_PASSWORD_CHANGED	INFORMATIONAL	The user password changed
MGMT_STARTED	INFORMATIONAL	Management started
MGMT_STARTED_FROM_RESTORE	INFORMATIONAL	Management started from restore
MGMT_GENERIC_EVENT		{0}

© 2026 - Hammerspace

NODE_MODE_ISSUE	NOTICE	Node "{0}" mode changed to OFFLINE. {1}
CLUSTER_FAILOVER	NOTICE	{0} is taking over as the primary Anvil
CLUSTER_SPLIT_BRAIN	ERROR	Critical error {0} in the Anvil cluster state has been identified. The Anvil can not continue to serve, please call support.
CLUSTER_REPLICATION_EVENT	INFORMATIONAL	Cluster replication state changed to {0}
DD_CLUSTER_PLANNED_SHUTDOWN	INFORMATIONAL	Anvil cluster shut down. Reason: {0}
DD_CLUSTER_PLANNED_SHUTDOWN_FAILED	ERROR	Failed to shut down Anvil cluster
NETWORK_LDAP_UNREACHABLE	ERROR	LDAP is unreachable
NETWORK_NTP_UNREACHABLE	ERROR	NTP is unreachable
VOLUME_STATE_CHANGED	INFORMATIONAL	{0} "{1}" changed state to "{3}"
VOLUME_DECOMMISSION_EVENT	INFORMATIONAL	{0} {1} decommissioned
VOLUME_ASSIMILATION_EVENT	INFORMATIONAL	Volume assimilation event
VOLUME_CONFIG_TEST_FAILED	ERROR	Volume configuration test failed from "{0}" on node "{1}" to "{2}" on volume "{3}". Output: {4}
VOLUME_CONFIG_TEST_ALL_FAILED	ERROR	Volume configuration test failed from node "{0}" to volume "{1}" over all IP addresses. Output: {2}
VOLUME_CLONE_TEST_FAILED	ERROR	Volume offloaded clone check failed. Reason: {0}
HW_NIC_LINK_EVENT	NOTICE	NIC event: NIC {0} {1}
HW_DRIVE_FAILED	ERROR	Physical drive {0} failed
HW_RAID_EVENT	NOTICE	RAID event occurred
HW_FAN_FAILED	ERROR	Fan failed
HW_PSU_FAILED	ERROR	Power supply failed
HW_FRU_EVENT	NOTICE	{0} hardware device {1}
VASA_VAAI_PROVIDER_EVENT	INFORMATIONAL	VASA VAAI provider event

© 2026 - Hammerspace

PORTAL_EXPORT_FAILED	ERROR	Data Portal "{0}" encountered problems while trying to export path "{1}"
PORTAL_UNEXPORT_FAILED	ERROR	Data Portal "{0}" encountered problems while trying to unexport path "{1}"
SUPPORT_CALL_HOME_EVENT	INFORMATIONAL	Support call home event
SHARE_QUOTA_WARNING	WARNING	Share quota is running out, crossed {0}%
SHARE_QUOTA_EXCEEDED	ALERT	Share quota exceeded
SHARE_RESTORED	INFORMATIONAL	Share {0} restored from snapshot {1}
SHARE_RESTORE_FAILED	ERROR	Failed to restore share {0} from snapshot {1}
SHARE_SNAPSHOT_CREATED	INFORMATIONAL	Created snapshot {0} for share {1}
SHARE_SNAPSHOT_CREATE_FAILED	ERROR	Failed to create snapshot {0} for share {1} due to error {2}
SHARE_SNAPSHOT_DELETED	INFORMATIONAL	Deleted snapshot {0} for share {1}
SHARE_SNAPSHOT_DELETE_FAILED	ERROR	Failed to delete snapshot {0} for share {1} due to error {2}
SHARE_CLONED	INFORMATIONAL	Share {0} cloned
SHARE_CLONE_FAILED	ERROR	Share {0} clone failed
ASSIMILATION_SUCCESS	INFORMATIONAL	Assimilation of {0} {1} by {2} completed successfully{3}
ASSIMILATION_FAILED	ERROR	Failed to assimilate data from {0} {1} by {2}{3}
ASSIMILATION_STOPPED	ERROR	Assimilation from {0} {1} by {2} has been stopped for user remediation{3}
ASSIMILATION_CANCELLED	INFORMATIONAL	Assimilation from {0} {1} by {2} has been cancelled{3}
READ_LATENCY_THRESHOLD_CROSSED	WARNING	{0} {1} exceeded read latency threshold
WRITE_LATENCY_THRESHOLD_CROSSED	WARNING	{0} {1} exceeded write latency threshold
SHARE_PRUNING	INFORMATIONAL	Share {0} pruned

© 2026 - Hammerspace

SHARE_PRUNING_BLOCKED	ERROR	Share {0} pruning blocked. Last Pruned:{1}
VOLUME_EVICTION_FAILED	ERROR	Attempt to evict files from Volume {0} failed
INVALID_COMB_STRUCTURE	ERROR	volume {0} Invalid COMB Structure
NO_REGISTERED_MOVER	ERROR	No registered mover in Data Manager
NO_REGISTERED_STORAGE_VOLUME	ERROR	No registered Storage Volume in Data Manager
PDDM_EXITED	ERROR	Data Manager Exited upon detecting inconsistency in data or configuration
PDDM_KILLED_DI	ERROR	Mover with ID {1} terminated
MODELER_SHARE_SWEEP_COMPLETED	INFORMATIONAL	Modeler share sweep of share {0} completed
PDDM_RESTART	INFORMATIONAL	Data Manager Restarted
COLLECTIONS_ENTER	INFORMATIONAL	Inode {1} entered collection {2}
COLLECTIONS_LEAVE	INFORMATIONAL	Inode {1} exited collection {2}
DISK_USAGE_LIMIT_WARNING	WARNING	Disk usage near limit on {0}:{1} with only {2} available.
DISK_USAGE_LIMIT_EXCEEDED	ERROR	Disk usage limit EXCEEDED on {0}:{1} with only {2} available.
SALT_NOT_FUNCTIONAL	ERROR	Salt is not functional. Reason: {0}
MISSING_DATA_MOVER	ERROR	No data mover detected. Please add a DSX node to the installation.
MISSING_CLOUD_MOVER	ERROR	No cloud mover detected. Please add a DSX node to the installation.
MISSING_STORAGE_VOLUME	ERROR	No writable backend storage volumes. At least 1 writable volume must be added before using the namespace.
DATA_COPY_TO_OBJECT_SUCCESS	INFORMATIONAL	Data copy to object from "{0}" in share "{1}" to bucket "{2}" and path "{3}" at "{4}" succeeded

© 2026 - Hammerspace

DATA_COPY_TO_OBJECT_FAILED	ERROR	Data copy to object from "{0}" in share "{1}" to bucket "{2}" and path "{3}" at "{4}" failed
BACKUP_NOT_SCHEDULED	WARNING	Metadata and configuration backup is not configured
BACKUP_FAILED	ERROR	Metadata and configuration backup to {0}:{1} failed. Schedule name: {2}.
BACKUP_COMPLETED	INFORMATIONAL	Metadata and configuration backup to {0}:{1} completed. Schedule name: {2}.
BACKUP_CREATION_FAILED	ERROR	Could not start metadata and configuration backup to {0}:{1}. Schedule name: {2}.
CTDB_NODE_FAILURE	ERROR	CTDB node {0}@{1} status is {2}
CTDB_NOT_RESPONDING	ERROR	CTDB service is not responding: {0}. Please contact support to resolve the situation.
CTDB_IN_RECOVERY	WARNING	CTDB service is in recovery mode. Please contact support to resolve the situation.
KMS_OPERATION_CHECK_FAILED	ERROR	KMS operation check failed. Reason: {0}
GFS_PARTICIPANT_ADDED	INFORMATIONAL	GFS participant added to share "{0}" for site "{1}"
GFS_PARTICIPANT_ADD_FAILED	ERROR	Failed to add GFS participant to share "{0}" for site "{1}". Reason: {2}
GFS_PARTICIPANT_REMOVED	INFORMATIONAL	GFS participant removed from share "{0}" for site "{1}"
GFS_PARTICIPANT_REMOVE_FAILED	ERROR	Failed to remove GFS participant from share "{0}" for site "{1}". Reason: {2}
REPLICATION_PORT_CHECK_FAILED	ERROR	Share replication port check failed. Details: {0}
GFS_PARTICIPANT_OPER_STATE_CHANGED		GFS participant operational state changed to state "{0}". Reason: {1}

© 2026 - Hammerspace

METADATA_REPLICATION_ISSUE		Metadata replication issue for share "{0}" and site "{1}". Reason: {2}
DATA_REPLICATION_ISSUE		Data replication issue for share "{0}" and site "{1}". Reason: {2}
REPLICATION_LATENCY_THRESHOLD_EXCEEDED		Replication {1} latency of {2} seconds exceeds the threshold for participant {0}
MISSING_SHARED_OBJECT_STORAGE_VOLUME		No healthy object storage volumes for share replication between share "{0}" and site "{1}".
CAPACITY_HEARTBEAT_DISABLED		Capacity heartbeat reporting disabled while operating under a capacity-based license
MOVER_CONNECTED	INFORMATIONAL	Mover "{0}" (uuid: "{1}") connected from ip address "{2}"
MOVER_UNKNOWN	WARNING	Unknown mover (uuid: "{1}") attempted to connect from ip address "{2}"
MOVER_BAD_IP		Mover "{0}" (uuid: "{1}") connected from an invalid data-channel ip address "{2}"
MOVER_DROPPED	ERROR	Dropped mover "{0}" (uuid: "{1}") due to heartbeat failure
MOVER_AT_CAPACITY		Mover "{0}" (uuid: "{1}") at "{2}" is over-utilized. Data mobility performance may be suboptimal. If this persists, consider adding memory or CPU resources to this mover, or adding another DSX
CAPACITY_THRESHOLD_PASSED	WARNING	'{1}" has exceeded its {2} capacity threshold of {3}% ({4}/{5} used). {7,choice,0#The volume will be less likely to be the target of new data and data mobility.
SHARE_EVENTS_PROCESSING		Event processing in progress for share "{0}". Volume removal may be affected until all events are processed. Events processed: {1}/{2}

© 2026 - Hammerspace

SHARE_EVENTS_PROCESSED		Event processing completed for share "{0}"
SHARE_EVENT_PROCESSING_BLOCKED		Event processing blocked for share "{0}". Please contact support
FAILURE_REPORTING_LICENSED_USAGE		Failure reporting usage for activation ID {0} to license server at {1}
INVALID_ADDITIONAL_ADDRESS		Additional IP address(es) mis-configured. Cause: {1}
KEYTAB_MISMATCH		Kerberos keytab file on Node "{0}" is out of sync, and attempts to re-sync it have failed. Please contact support.
AD_WITHOUT_NTP		When Active Directory is joined, NTP should also be configured
NTP_NOT_SYNCHRONIZED		The clock between "{0}" and "{1}" is not synchronized. Check for correct configuration and networking issues
AD_DC_OUT_OF_TIME_SYNC		Active Directory Domain Controllers out of time sync with product. Product time: "{0}". Out of sync DCs: {1}
AD_COMPUTER_ISSUE		Problem found with Active Directory computer account "{0}". Cause: {1}. Either remedy in Active Directory or leave and re-join Active Directory.
AD_CACHE_FLUSH_SUCCESS		Active Directory cache flush successful.
AD_CACHE_FLUSH_FAILED		Active Directory cache flush failed. Reason: {0}"
IDMAPD_SYNC_ISSUE	INFORMATIONAL	{0}
REPLICATION_WITHOUT_NTP		The site "{0}" at {1} is participating in one or more global filesystems and needs NTP configured

© 2026 - Hammerspace

REPLICATION_CLOCK_OUT_OF_SYNC		The site "{0}" at {1} is participating in one or more global filesystems and its clock differs by {2} seconds with this site"s clock. Ensure that both sites are properly configured with NTP.
AD_SERVER_ISSUE		Problem found with Active Directory server value: {0}. Either clear the list of AD servers, or update it to contain only valid and reachable servers.
SERVICE_OPER_STATE_ISSUE	NOTICE	{0} "{1}" operational state changed to "{2}". {3}
OBJECT_VOLUME_IO_TEST_FAILED		Object volume I/O test failed from node "{0}" to volume "{1}". Cause: {2}
OBJECT_VOLUME_IO_TEST_ALL_FAILED		Object volume I/O test failed from all nodes to volume "{0}". See individual failures for causes.
OBJECT_VOLUME_RESERVATION_OLD	NOTICE	An old reservation exists on object volume "{0}" for site "{1}". The last refresh of the reservation was at {2}. The site at {3} {4,choice,1#does not respond
DNS_ISSUE	ERROR	Domain name resolution {0}. {1}
CLUSTER_DEGRADED	ERROR	HA cluster state degraded. Reason: {0}
OBJECT_VOLUME_GC_COMPLETED	INFORMATIONAL	{3}
OBJECT_VOLUME_GC_FAILED	ERROR	{3}
OBJECT_VOLUME_GC_CANCELLED	INFORMATIONAL	{3}
OBJECT_VOLUME_GC_BLOCKED		GC is blocked for volume {0} due to {1}
ECGROUP_ARRAY	INFORMATIONAL	ECGROUP array : {0}
ECGROUP_MOUNTPOINT	INFORMATIONAL	ECGROUP mountpoint : {0}
ECGROUP_STORAGE	INFORMATIONAL	ECGROUP storage : {0}

© 2026 - Hammerspace

OSV_QUORUM_IN_FLUX		Object volume quorum undergoing changes. Content database should resolve the issue as it discovers the site that is joining or leaving.
OSV_HMDB_REPLICATION_OUT_OF_SYNC		Object volume content database replication out of sync. Garbage collection disabled. Please contact support.
OSV_SITE_TO_SITE_CONN_FAILED		Network connection failure to Object volume content database at remote site anvil {2}:9298 (TCP). Check network configuration: {3}
OSV_CSP_CONN_FAILED		Object volume content database cloud service provider connection failure: {0}
PROMETHEUS_DEGRADED		Prometheus exporters in degraded state on node {0}
OSV_HMDB_INSUFFICIENT_GC_MEMORY		The system is unable to initiate Object Volume Garbage Collection. Contact support for additional help: Insufficient allocated memory for Object volume content database; {0} bytes required, {1} bytes available; filter false positive rate: {2}
QUORUM_DEVICE_CONFIGURED		Quorum Device configured
QUORUM_DEVICE_UNCONFIGURED		Quorum Device unconfigured
QUORUM_DEVICE_UNHEALTHY		Quorum Device is unhealthy with respect to Anvil {0}
MISSING_COMB_STRUCTURE	ERROR	Storage volume is missing a comb structure for MDSI "{0}" ({1}). Will re-attempt creation.
KERBEROS_SYNC_FAILURE		Failed to synchronize Kerberos configuration to node {0}
KERBEROS_SHARE_WITH_UNJOINED_AD		Share "{0}" has Kerberos security options, but AD is not joined
NFS_LAYOUT_ERROR_STALE	ERROR	Data I/O error for share "{1}": {2} recent attempts to access data with stale file handle. {3}

© 2026 - Hammerspace

NFS_LAYOUT_ERROR_NXIO	ERROR	Data I/O error for share "{1}": {2} recent attempts to access an unavailable volume. {3}
NFS_LAYOUT_ERROR_IO	ERROR	Data I/O error for share "{1}": {2} recent attempts to access files whose underlying instances could not be found. {3}
NFS_LAYOUT_ERROR_ACCESS	ERROR	Data I/O error for share "{1}": {2} recent denied volume access attempts. {3}
S3_MULTIPART_UPLOAD_ABORTED		{0} S3 multipart {1,choice,1#upload
OSV_HMDB_DEGRADED_BLOOM_FILTER_ALLOCATED		The system is unable to initiate Object Volume Garbage Collection at full efficiency. Contact support for additional help: Insufficient allocated memory for Object volume content database; {2} bytes required, {3} bytes available; filter false positive rate: {0}, preferred rate: {1}
KERBEROS_CLUSTER_UNHEALTHY		Cluster misconfigured for Kerberos data portal access: {0}
KERBEROS_SHARE_WITH_NO_NFS_SPNS		Share "{0}" has Kerberos security options, but NFS SPNs are not configured
CLUSTER_MISCONFIGURED		A quorum device must be configured, or a DSX node must be added in order to maintain high availability under all scenarios.
NoSuchBucket	NOTICE	S3 service: a Hammerspace-created bucket directory is missing on disk (for example, removed over NFS/SMB). Reported once per S3 service restart; audit message that does not clear.
NoSuchContainer	NOTICE	S3 service: a Hammerspace-created bucket container directory is missing on disk (for example, removed over NFS/SMB). Reported once per S3 service restart; audit message that does not clear.

© 2026 - Hammerspace

NVMEOF_PATH_DEGRADED	WARNING	NVMe-oF path degraded (controller stuck connecting): {0}, {1}, {2}, {3}, {4}
NVMEOF_PATH_LOST	WARNING or CRITICAL	NVMe-oF path lost: {0}, {1}, {2}, {3}, {4}

© 2026 - Hammerspace

Appendix C: Hammerspace Terminology

The following list of terms may be helpful to users of this documentation.

Table 20. Hammerspace terminology

Term	Definition
Alignment	Used to report if a file is currently adhering to all, some, or none of the currently applied objectives.
Aligned	Indicates that all applicable objectives have been met for a file.
Anvil	A Hammerspace metadata server. Responsible for managing all the metadata for the Global File System, and hosts the API and Management GUI services.
Assimilation	The ingestion of all existing file and folder metadata required to serve data using the new Hammerspace global namespace. Can be Read-Write or Read-Only.
Attribute	A type of custom metadata that can be applied and values set by the user using the **hstk**.
Cloud Mover	A DSX service responsible for uploading and downloading data from cloud and object storage.
Collections	Subsets of files determined by a defined query of metadata updated on each pass of the sweeper process. Accessible in any directory in a Hammerspace share in .collections.
Condition	Rules used to specify when an objective will be applied. If no condition is specified, the objective will be applied as TRUE
Data Orchestration	The movement of data based on Hammerspace objectives.
DSX (Data Services nodes)	The Hammerspace data services node. Includes functional capabilities like Portal, Mover, Cloud Mover, and Store.
DSX Store	A DSX service that makes block storage available as file storage volumes using XFS.
Global file share	A share added to more than one Hammerspace cluster. Each cluster has the same view of the share contents.

© 2026 - Hammerspace

Hammerscript	The proprietary scripting language used to define and work with metadata, modeled after spreadsheet formula syntax. Used for queries, objectives, and reporting.
Hammerspace Toolkit (HSTK)	A collection of Python-based extensions for command-line interfaces (Linux Bash, Windows CMD/PowerShell) used to query and manipulate file system metadata. See HSTK GitHub for more information.
Instance	A file placed on Hammerspace will have at least one instance written to the underlying storage; more if the objectives require it.
Label	A type of custom metadata; a simple string that uses an administrator-defined schema.
Metadata	Data that provides information about other data. Hammerspace manages POSIX, Internal, and Custom metadata.
Mover	A stateless DSX service that moves files non-disruptively between file storage volumes.
Objective	Rules that define the intent for managing files. Uses metadata criteria to orchestrate data access and placement.
Offline file	A file considered offline when it is only available in an object storage volume.
Online file	A file considered online when it is in either a DSX volume or a connected NFS storage volume.
Partially aligned	Indicates that a file adheres to at least some of the applicable objectives.
Portal	A DSX service enabling protocol access for NFS, SMB, and S3 clients.
Retention policy	Specified length of time to keep a file system snapshot or file clone.
Schedule	Specified time and frequency to take a snapshot or make a file clone.
Shadow namespace	Provides hidden access to Hammerspace metadata and functionality through a mounted file system, using ?. extensions.

© 2026 - Hammerspace

Share	A file system presented for file access by clients via NFS, SMB, and/or S3 protocols. Objectives determine where the file instances are placed.
Site	Refers to a specific Hammerspace cluster, which may or may not be at another physical location.
SLO (Service-level objective)	A target value or range of values measured by service-level indicators. Often refers to an objective to place file instances on specific storage.
SMB (Server Message Block)	A network communication protocol for providing shared access to files, printers, and serial ports between nodes on a network.
Snapshot	A record of the state of the file system at a specific point in time.
Tag	A type of custom metadata; a key-value pair that does not use a predefined schema.
Unaligned	Indicates that a file does not currently adhere to any of the applicable objectives.
Volume	A storage container used by Hammerspace to store instances of files. Can be DSX storage drives, NFS exports, or object storage buckets.
Volume groups	Used to group storage systems, volumes, or other volume groups for easier targeting in objectives.

© 2026 - Hammerspace

Appendix D: Additional Documentation Resources

Every Hammerspace title is available at docs.hammerspace.com: the product user guides, the solution guides, the compliance and security documents, and the Hammerspace Knowledge Base. Read any title online, or download it as a PDF from the title's own pages or from the Downloads page.

The following titles may be of special interest to users of this guide:

- Assimilation Overview [Solution Guide]: An overview of a typical data migration with Hammerspace using assimilation and data orchestration.
- Objectives Guide [Solution Guide]: How to configure Hammerspace objectives.
- Hammerscript and the Hammerspace Toolkit Guide [Solution Guide]: An introduction to the concepts of Hammerspace metadata, and the tools to manipulate metadata and put it to work to achieve business objectives for data.
- S3 Guide [Solution Guide]: This document covers the creation and management of S3 servers, bucket containers, and buckets, as well as user access control and security considerations, ensuring seamless multi-protocol data access and efficient data management.

To download Hammerspace software and licenses, use the Hammerspace Licensing and Delivery Portal. To open a support case, use the Hammerspace Support Hub.

© 2026 - Hammerspace