Hammerspace Administration Guide

Version 5.3 · Published 2026-10-16

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 22
     - 6.5.2. Viewing the Details of a Notification Rule 22
     - 6.5.3. Disabling Email Events 23
   - 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 33

Enabling Kerberos Security on NFS Export Options 33
Using the Management GUI 33
Automatic Kerberos Export Options on the Root Share 35
Enabling the NFSv4.1 Service 36
Registering NFS Principal Names 36
In the Admin CLI 36
In the Management GUI 36
7.2.5. DNS Requirements 37
Joining Active Directory 37
7.2.6. Client Configuration 37
Mounting a Share Using Kerberos Security 39
Mounting a Share over NFSv4.2 with Kerberos 40
7.3. Configuring an LDAP Name Service 41
7.3.1. Prerequisites 42
7.3.2. Important Information 42
7.3.3. Limitations 43
7.3.4. Best Practices 44
7.3.5. Securing LDAP Connections with LDAPS or StartTLS 45
What Hammerspace Checks 45
Trusting the LDAP Server Certificate 46
Choosing the Server Address 46
What You See When the Certificate Is Wrong 46
7.3.6. How Hammerspace Manages SSSD on the Anvils 47
Configuration Files 47
Configuration States 47
When a Sync Fails 48
7.3.7. Procedures 48
Adding an LDAP Name Service 49
Using the GUI 49
Using the Admin CLI 51
Verifying the Configuration (User/Group Lookup) 52
Using the GUI 52
Using the Admin CLI 54
Testing the Connection to a Name Service 55
Using the GUI 55
Using the Admin CLI 56
Viewing Configured Name Services and Resolution Order 56
Using the GUI 56
Using the Admin CLI 57
Changing the Directory Services Resolution Order 58
Using the GUI 58
Using the Admin CLI 59
Updating an LDAP Name Service 60
Using the GUI 60
Using the Admin CLI 62

Removing a Name Service 63
Using the GUI 63
Using the Admin CLI 63
7.3.8. Admin CLI Reference: name-service-config 63
7.3.9. REST API Reference 66
7.3.10. Troubleshooting 68
7.4. NFS 70
7.4.1. NFSv4 Client Unique-Id Requirement 70
7.4.2. NFS Version-Specific Mount Examples 71
7.4.3. Supported NFS Protocols 71
7.4.4. NFSv4.1 ACL Passthrough on Portal Mounts 72
7.4.5. Enabling NFS LOCALIO on Linux Clients 73
7.5. Multi-Protocol Support 74
7.5.1. Multi-Protocol File Locking 74
7.6. S3 74
7.6.1. S3 Metadata 75
7.6.2. S3 Terminology 75
7.6.3. S3 Limitations 76
7.6.4. Multi-Tenancy 76
7.6.5. Managing S3 Servers 77
7.6.6. Transport Security 77
7.6.7. X.509 Server Certificate 77
7.6.8. Basic Workflow for Configuring S3 77
7.6.9. Preparing for S3 Configuration 78
7.6.10. Creating an S3 Server Using the GUI 78
7.6.11. Deleting an S3 Server 80
7.6.12. Managing S3 Buckets 81
Deleting an S3 Bucket 81
Using the Management GUI 82
Missing Bucket and Bucket Container Events 82
7.6.13. Managing S3 Users 83
Adding an Access Key to an S3 Server 83
Adding Multiple Local Users at the Same Time 85
Removing Access to an S3 Server 85
7.6.14. S3 API Overview 86
7.6.15. S3 Object Keys and the File System Namespace 88
How Object Keys Map to Files and Directories 88
Object and Prefix Name Conflicts 89
Keys That End in a Slash 91
How a Folder Appears in Listings 91
When a Folder Disappears 92
Listing and Lookup 92
Deleting Objects and Prefixes 93
Guidance for Applications and Test Suites 94
7.7. SMB 94

7.7.1. Active Directory Integration 94
Joining an Active Directory Domain 94
Leaving an Active Directory Domain 96
7.7.2. User Mapping Between Windows and Linux 96
Default Behavior 97
Mapping the UID and GID Within the Same Active Directory Domain 97
Cross-Domain Mapping 99
Creating Mappings Between Domains 99
Selecting a Preferred Domain 99
Removing a Preferred Domain 99
Applying a Preferred Domain to a Share 99
Reloading Updated Domain-Mapping Rules 99
Reviewing Cross-Domain Mappings 100
7.7.3. Client Access to Data Using SMB 100
macOS Client Access 100
Windows Client Access 102
7.7.4. Load Balancing 102
7.7.5. Mapping a Share 102
7.7.6. Mapping a Drive Letter Using the Windows CMD 103
7.7.7. SMB Multi-channel 104
Requirements 104
Enabling SMB Multi-channel with the GUI 104
7.7.8. Viewing Mapped Shares Using the Windows CMD or PowerShell 105
7.7.9. Unmounting a Share Using the Windows CMD 105
7.7.10. Supported SMB Versions 106
7.7.11. Creating Hidden SMB Shares 106
7.7.12. Setting the Browsable Option 106
Using the GUI 106
Using the Admin CLI 107
7.7.13. Setting Aliases for SMB 107
Adding Aliases to an Existing Share 108
Listing SMB Aliases 110
Removing Aliases 110
Additional Details 110
7.7.14. Allowing Access-Based Enumeration 111
Enabling ABE for an Entire Share Using the GUI 112
Enabling ABE for a Sub-folder Using the GUI 113
Enabling ABE Using Objective Expressions 113
7.7.15. Using the Administrators Group 114
Viewing Existing Members of the Administrators Group 114
Adding a New Group to the Administrators Group 114
Adding an Individual User to the Administrators Group 114
Removing Members of the Administrators Group 115
7.7.16. Windows Previous Versions 115
7.7.17. Using Microsoft Management Console 116

Managing Share-Level Permissions 117
Using the Microsoft Management Console to Set Share Permissions 117
Viewing and Closing Files 119
Viewing and Closing Sessions 121
7.8. Active Directory Permissions 122
7.8.1. Two Accounts Are Involved 122
7.8.2. Minimum Permissions: Joining with a New Computer Account 122
7.8.3. Additional Permissions by Capability 123
7.8.4. What Each Capability Means in Practice 124
7.8.5. Recommended Delegation for Most Sites 125
7.8.6. Troubleshooting 126
8. Data Management 127
8.1. At the Share Level 128
8.2. At the Directory Level 128
8.3. Anti-virus Scanning 129
8.3.1. Requirements 129
8.3.2. How It Works 129
8.3.3. Limitations 130
8.3.4. Virus Definition Updates 130
8.3.5. Performance 130
8.3.6. Adding an ICAP Server 131
8.3.7. Identifying Infected Files 131
8.3.8. Locating Infected Files by Querying Data 132
8.3.9. Remediation 133
8.3.10. Reporting 133
8.3.11. Viewing Only Top Files 134
8.3.12. Viewing Summary of Virus-Scanning Operations on a Share 134
8.3.13. Viewing Threat Status of an Individual File 135
8.3.14. Viewing Threat Status of a Share 135
8.3.15. Virus-Scanning Objective 137
8.4. Cross-Site Versioning 137
8.4.1. Configuring Cross-Site Versioning Using the Admin CLI 137
8.4.2. Configuring Cross-Site Versioning Using the GUI 138
8.5. Data Availability and Durability 138
8.5.1. The Availability Number-of-Nines Objective 139
Best Practices 139
8.5.2. Volume Availability - Default Values 139
8.5.3. Mirroring Data Between NFS Volumes 139
8.5.4. Using the Durability Number-of-Nines Objective 140
8.5.5. Volume Durability – Default Values 140
8.6. Data Profiler 140
8.6.1. Data Mobility 141
8.6.2. Mobility History Chord Diagrams 142
8.6.3. Mobility History Lists 144
8.6.4. Alignment 145

Alignment Reporting 145
Resolving Alignment Issues 147
8.7. File Clones and File Snapshots 147
8.8. File Versioning 148
8.8.1. Configuring File Versioning Using Objectives 148
8.8.2. Configuring File Versioning in the GUI 148
8.8.3. Locating Versioned Files 150
8.8.4. Restoring a Previous Version of a File 151
8.8.5. Protecting Files from Deletion 152
How Undelete Works 152
8.8.6. Configuring Undelete Using the GUI 153
8.8.7. Configuring Undelete Using the CLI 155
8.8.8. Recovering Deleted File Using the GUI 156
8.9. Managing Labels 157
8.9.1. Creating Labels 158
8.9.2. Listing Labels 159
8.9.3. Renaming Labels 159
8.9.4. Changing the Label Hierarchy 160
Move Coast Live Oak and Black Oak to Large Oaks 161
Move Shrub Oak to Small Oaks 161
List the Labels After the Change 161
8.9.5. Removing Labels 162
8.9.6. Using the Hs Command with Labels 162
Adding a Label to a File Using the Hs Command 163
Listing All Files and Their Paths with a Particular Label 163
Listing All Files with a Particular Label 163
8.10. Managing Shares 163
8.10.1. Creating a Share 164
8.10.2. Deleting a Share 166
What Happens When You Delete a Share 166
Shares Referenced by SMB Autohome Entries 167
Using the Admin CLI 167
8.10.3. Changing the Default File Placement Behavior 168
8.11. Referral Shares 169
8.11.1. REST API Rules 169
8.11.2. Client Access and Transparency 170
8.11.3. Limitations 170
8.11.4. Creating a Referral Share 170
Using the Admin CLI 170
Using the Management GUI 170
8.11.5. Updating a Referral Share 171
Updating the Referral Path Using the Admin CLI 171
8.11.6. Deleting a Referral Share 171
Using the Hammerspace Admin CLI 171
Using the REST API 171

8.11.7. Additional Tasks 171
Adding or Removing Referral Addresses Using the Admin CLI 171
Replacing All Referral Addresses 171
8.12. Snapshots 172
8.12.1. Accessing Snapshots from Clients 172
Linux or macOS NFS Client 172
Windows Client 173
8.12.2. Creating an Instant Snapshot 173
8.12.3. Custom Schedules and Retention Policies 174
Custom Scheduling Examples for the Admin CLI 175
Creating a Custom Snapshot Schedule 175
Viewing Snapshot Schedules 175
Custom Retention Examples for the Admin CLI 175
Creating a Snapshot Retention-Time Policy 175
Creating a Policy for Number of Snapshot Copies to Retain 175
Creating a Share-Snapshot Schedule and Retention Policy 175
8.12.4. Deleting Snapshots Using the Management GUI 175
8.12.5. Listing Snapshots Using the Admin CLI 176
8.12.6. Restoring a Snapshot 176
8.12.7. Restoring Individual Files Using the CLI 177
8.12.8. Saved-Snapshot Location 177
8.12.9. Snapshot Schedule 178
Example Snapshot Schedules 178
Creating Snapshot Schedule in the GUI 178
Creating Snapshots Using the CLI 179
8.13. Using Objectives 179
8.13.1. Default Objectives 180
8.13.2. Predefined Objectives 180
8.13.3. Editing an Objective 182
8.13.4. Advanced Objectives 183
Common Operators 183
Common Functions for Advanced Expressions 183
8.13.5. Examples of Advanced Objectives 185
Tiering of Data from File Storage to Archive (Cloud/Object) 185
Using the Admin CLI 185
Move Snapshots to Cloud/Object Storage 185
Keep Most Recent Snapshot on Original Location 186
Keeping Live Tree on File Volumes 186
Keeping File Clones on Cloud/Object Storage 186
Placing PDF Files on Cloud/Object Storage 186
Moving Data in the Archive Directory to Cloud/Object Storage 186
Moving Undelete Files to Cloud or Object Storage 187
9. Storage Management 188
9.1. Adding Storage Capacity 188
9.1.1. Adding a Storage System 188

9.1.2. Adding a Storage Volume 188
9.1.3. Resizing a 3rd Party NAS Volume 191
9.2. Adding Storage to DSX 191
9.2.1. Part 1: Adding Storage 191
9.2.2. Part 2: Refreshing the DSX Configuration 191
9.2.3. Part 3: Creating a New Logical Volume 192
9.2.4. Part 4: Registering the New Volume 193
9.3. Connecting Storage to a DSX Using NVMe-oF 194
9.3.1. Adding an NVMe-oF Enclosure and Subsystems 196
Configuring the Logical Volumes 201
9.3.2. Admin CLI Command: nvmeof-config 203
Examples 204
9.3.3. Admin CLI Workflow to Connect NVMe-oF Enclosures 205
9.3.4. Additional Admin CLI Examples 210
Listing Enclosures and Connections 210
Adding Targets 213
Changing an Enclosure's Connection Settings 214
Connecting Subsystems 215
Disconnecting Subsystems 216
Removing an NVMe-oF Enclosure 216
Creating a Logical Volume 217
9.3.5. NVMe-oF Path Health and Recovery 218
Path Health Events 218
How the Events Behave 219
Responding to a Path Health Event 219
SNMP Traps 220
Connection State Is Not Path Health 220
Controller Loss Timeout 222
Why -1 Is Recommended 223
Checking the Current Value 223
Setting the Value 224
9.3.6. NVMe-oF Limitations and Operating Guidance 224
Presenting Storage to DSX Nodes 224
The DSX Host NQN 225
Connecting and Disconnecting 225
9.3.7. Upgrading a Cluster That Uses NVMe-oF Storage 225
Before You Upgrade 225
After You Upgrade 226
9.4. Support for AWS S3 Glacier Instant Retrieval (GIR) 226
9.4.1. When to Use GIR 227
9.4.2. Cost Considerations 228
Additional Billing Rules 228
9.4.3. Cost Observability and Reporting Limitations 229
9.4.4. How Hammerspace Manages GIR 229
GIR Is a Volume-Level Designation 230

9.4.5. Prerequisites 230
9.4.6. Adding an AWS S3 Storage System 230
Using the Web UI 231
Using the Admin CLI 231
9.4.7. Adding a GIR Object Storage Volume 232
Using the Web UI 232
Using the Admin CLI 232
Default Settings for GIR OSVs 233
Verifying the OSV Was Added Correctly 233
9.4.8. The Archive-Object-Volumes Volume Group 234
9.4.9. Placing Data on GIR — Configuring Share Archive Rules 235
Adding an Archive Rule Using the Web UI 235
Adding a Placement Objective Using the Admin CLI 236
Placement Objective Reference 236
9.4.10. Mobility Behavior 237
How GIR Contributes to Availability and Durability Scoring 238
9.4.11. Replication Behavior 238
9.4.12. Shared and Mixed-Version Configurations 238
Adding a Shared GIR OSV 238
Storage Class Validation on Shared OSVs 239
Mixed-Version Deployments (Sites with and Without Storage Class Support) 239
Legacy OSV Identity Files 240
9.4.13. Verifying Your GIR Configuration 240
9.4.14. Limitations and Constraints 240
9.4.15. Upgrade Considerations 241
9.4.16. Troubleshooting 241
9.4.17. CLI Reference: object-volume-add 242
9.4.18. FAQ 244
9.5. Capacity Management 245
9.5.1. Viewing Total Capacity 246
9.5.2. Addressing Capacity Issues 246
9.6. Removing Storage 246
9.6.1. Part 1: Identifying the Volume 246
What Happens During a NAS Volume Decommission 247
9.6.2. Part 2: Deleting the Logical Volume (CLI Only) 248
10. User Management 249
10.1. Account Auto Locking 249
10.2. Adding Local Management User Using the GUI 250
10.3. Applying Network Restrictions 251
10.4. Clearing Network Settings 251
10.5. Deleting a User 251
10.6. Editing a User 252
10.7. Enterprise Sign On 252
10.7.1. Limitations 252
10.7.2. Setting Up Enterprise Sign On 252

10.7.3. Securing the Connection to the Identity Provider 254
10.8. Local Management Users 255
10.9. Login Policies 255
10.10. User Access to the Management Interfaces 255
10.11. User Roles 255
10.11.1. Management Roles 255
10.11.2. Data-Access Roles 256
11. Hammerspace System Management 257
11.1. Server Certificate 257
11.1.1. Installing a Certificate Using the Management GUI 258
11.1.2. Viewing and Downloading the Installed Certificate 259
11.1.3. Installing a New Certificate Using the Admin CLI 259
11.1.4. Resetting the Installed Certificate 262
11.2. System Backup and Restore 263
11.2.1. System Backup 263
Creating a System Backup Schedule 264
Creating a Scheduled Backup 264
Checking the Backup Schedule 264
Changing or Removing a Schedule and/or Changing the Backup Location 265
Listing System Backups 265
11.2.2. Restoring System from Backup 266
11.3. Upgrading the Product 266
Appendices **267**
Appendix A: Technology Preview 268
A.1. Using TLS for End-to-End Encryption 268
A.1.1. Important Considerations 269
A.1.2. Limitations in Hammerspace 5.3 270
A.1.3. Prerequisites 271
Hammerspace Cluster 271
Linux Clients 271
Third-Party Backend Storage 271
Backup and External Storage Servers 272
A.1.4. Workflow Overview 272
Enabling TLS on an Existing Cluster 272
Enabling TLS on a New Cluster (Recommended) 273
A.1.5. Enable TLS Using the Management GUI 273
A.1.6. Remount Clients 275
Remounting an Enterprise Linux 10.1 (or Later) Client 275
Prerequisites 275
Procedure 276
Verify the TLS Connection 277
Remounting an Ubuntu Client 277
Prerequisites 278
Procedure 279
Verify the TLS Connection 280

A.1.7. External Movers (DI and Cloud Mover Containers) 280
Supplying PKI Material to External Movers 280
Creating the tlshd Configuration File 281
Adding the Required Trust Anchors 281
Restarting an External Mover 281
A.1.8. Disable NFS over TLS 282
A.1.9. Certificate Lifetimes and Expiration 282
Certificate Lifetimes 282
Checking Certificate Expiration Dates 283
When a Certificate Expires 284
Renewing Certificates 285
A.1.10. Troubleshoot TLS Issues 286
TLS-related Events 286
Client Cannot Mount After TLS Is Enabled 286
File I/O Hangs on a TLS Mount 288
Backups Fail with "access denied by server" 288
Diagnosing TLS Connectivity to a Storage Node 289
External Mover Refuses TLS-Required Mobilities 289
Local tlshd.conf Edits Lost After a Software Update 289
Spurious TLS-unhealthy Alert During a Software Update 290
A.1.11. Manage TLS from the Command Line 290
Enable or Disable TLS 290
Manage Certificates and CAs 291
List the System Root CA 292
List All Registered Certificates 292
Add a Trusted CA Certificate Using the Management GUI 293
Add a Certificate or CA to the Trust Store 293
Sign a CSR with the Hammerspace System CA 294
Create a Self-Signed Certificate for Labs and POCs 295
Remove a Certificate from the Trust Store 295
A.1.12. Reference: Client Requirements 295
A.1.13. Reference: Supported Third-Party Backend Storage 296
A.1.14. Reference: Certificate Trust Model 296
Appendix B: Event Listing 298
Appendix C: Hammerspace Terminology 311
Appendix D: Additional Documentation Resources 314

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 TLS end-to-end encryption — 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

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:                                aeeadb7f-7258-41e3-aad9-bc57ec176769
Name:                              AEL640TQCSS160
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:                             2026-09-22 13:11:34 UTC
Created:                           2026-09-22 13:11:15 UTC
Timezone:                          UTC
NFS transport policy:              TLS forbidden
GFS participant max suspected time: 30 minutes
Metered billing:                   Disabled
Prometheus exporters:              Enabled
<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:

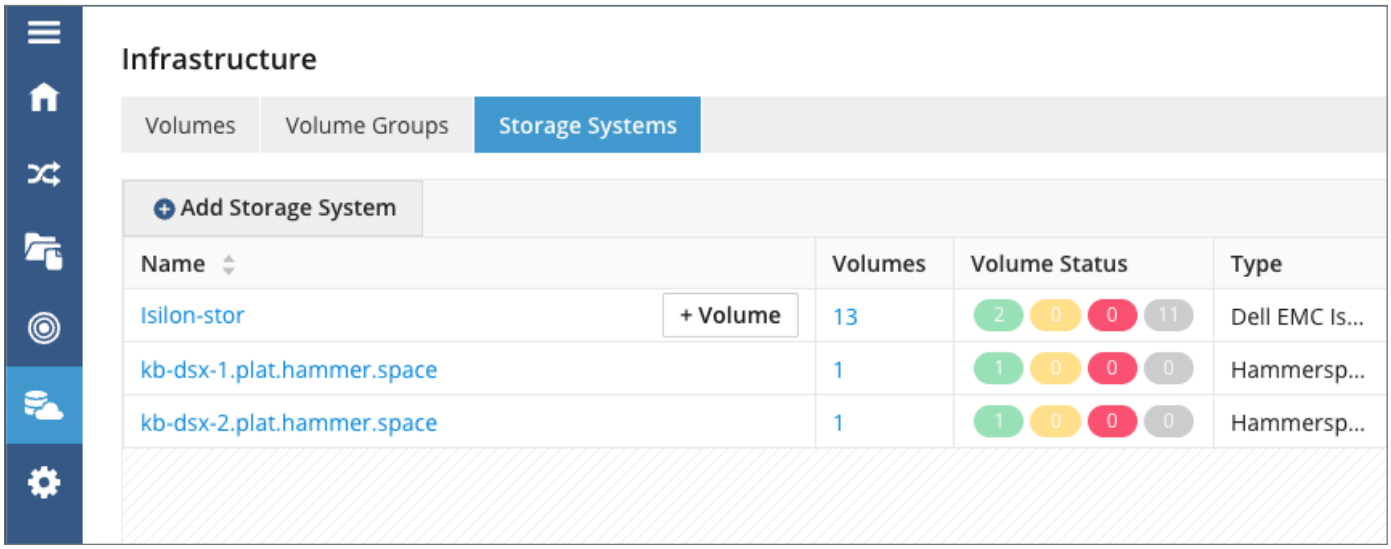


Figure 1. Viewing storage-system status

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

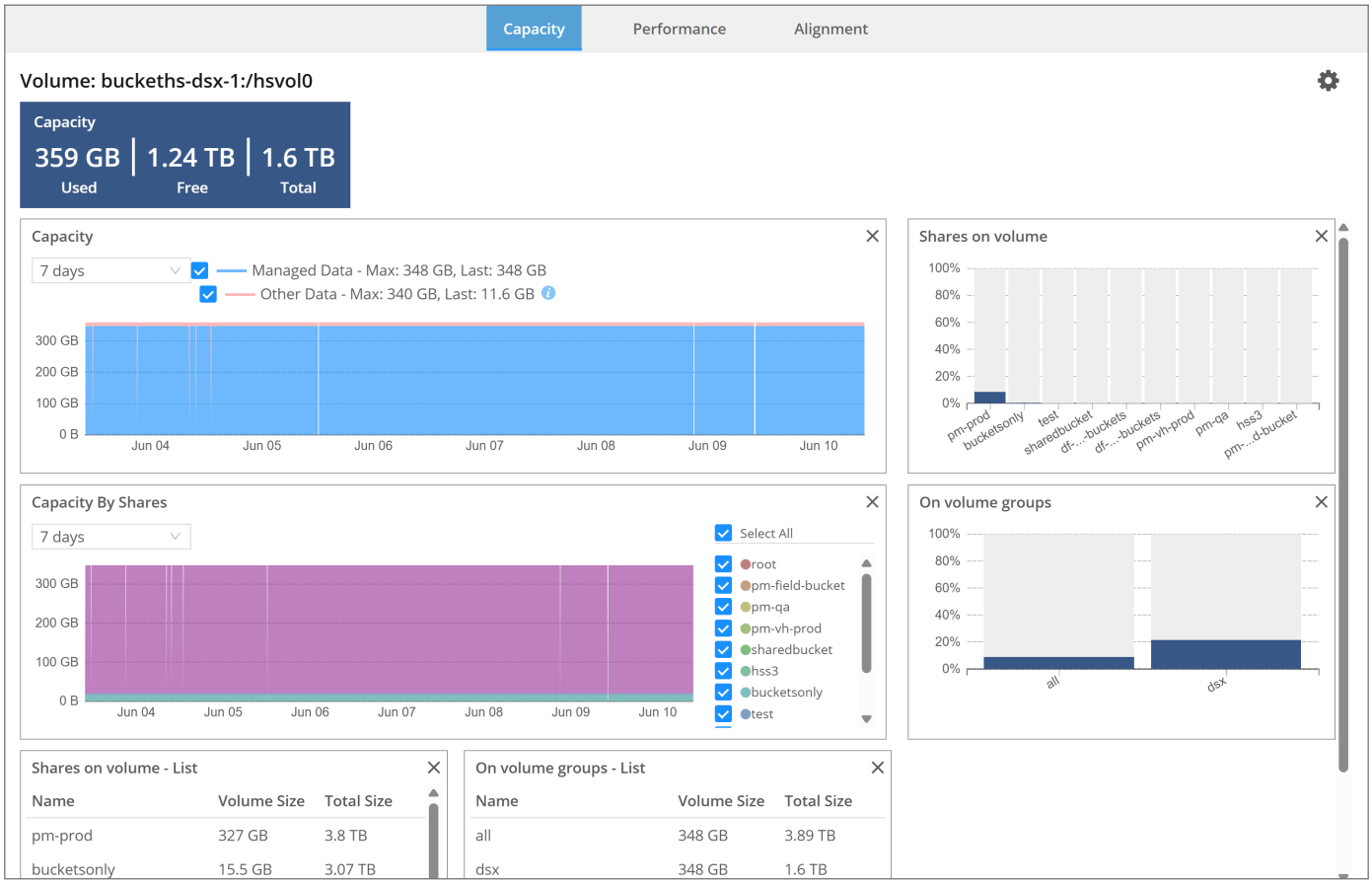


Figure 2. Viewing infrastructure volume capacity

© 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.

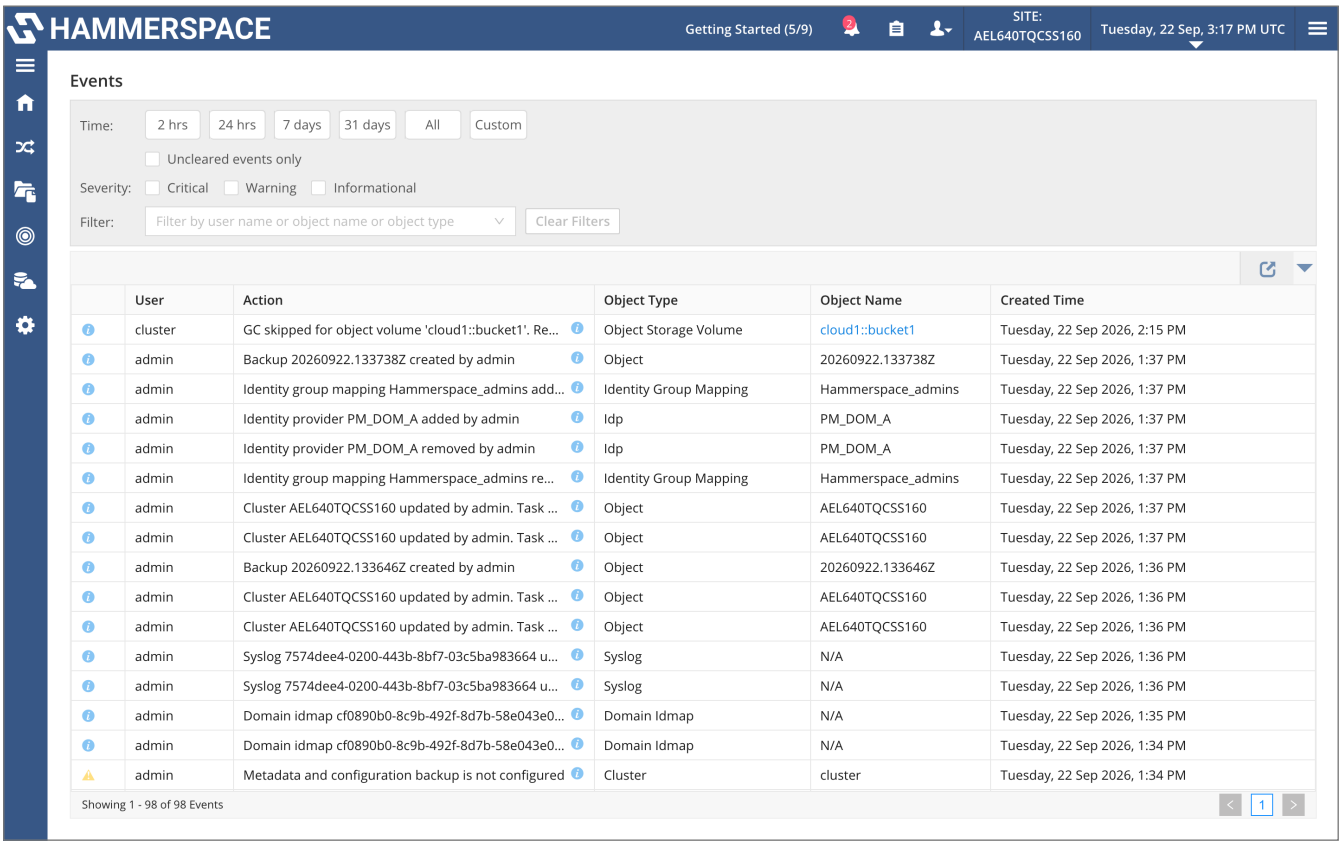


Figure 3. Viewing the Events list

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 587 --connection-security
STARTTLS
```



`--connection-security` replaces the deprecated `--ssl`/`--no-ssl` options. Existing configurations migrate automatically on upgrade: a prior `--ssl` becomes `TLS`; unset or `--no-ssl` becomes `NONE`. Update any scripts that still use `--ssl`/`--no-ssl`. Selecting `TLS` for a server that expects STARTTLS (or the reverse) causes the connection to fail.

The same `--connection-security` option applies to `idp-add` and `idp-update` when configuring an Active Directory identity provider. See Securing the Connection to the Identity Provider.

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.

© 2026 - Hammerspace

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
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:

© 2026 - Hammerspace

```
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

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:
```


© 2026 - Hammerspace

```
                    [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

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.

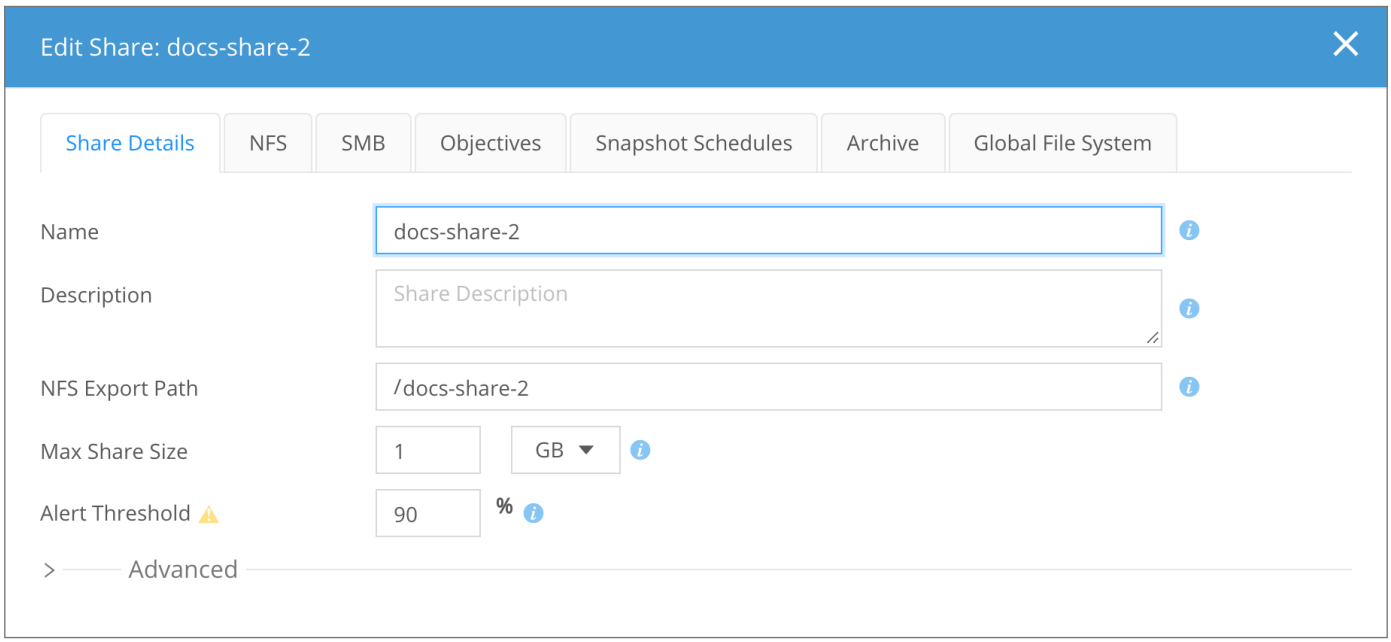


Figure 4. Setting share size and alert threshold

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

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

The following example shows the command and output:

© 2026 - Hammerspace

Share-capacity WARNING notification

Command:

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

Expected output:

```
total 1
ID:                     375bdb12-3e52-4629-b536-1ca2b306bad7
Created:                2026-09-22 13:28:30 UTC
Modified:               2026-09-22 13:28:30 UTC
Type:                   SHARE_QUOTA_WARNING
Description:            Share 'docs-share-2' quota is running out, crossed 90%
Severity:               WARNING
Source type:            Share
Source ID:              8b7c54c1-fd27-4342-8421-348891951bd8
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

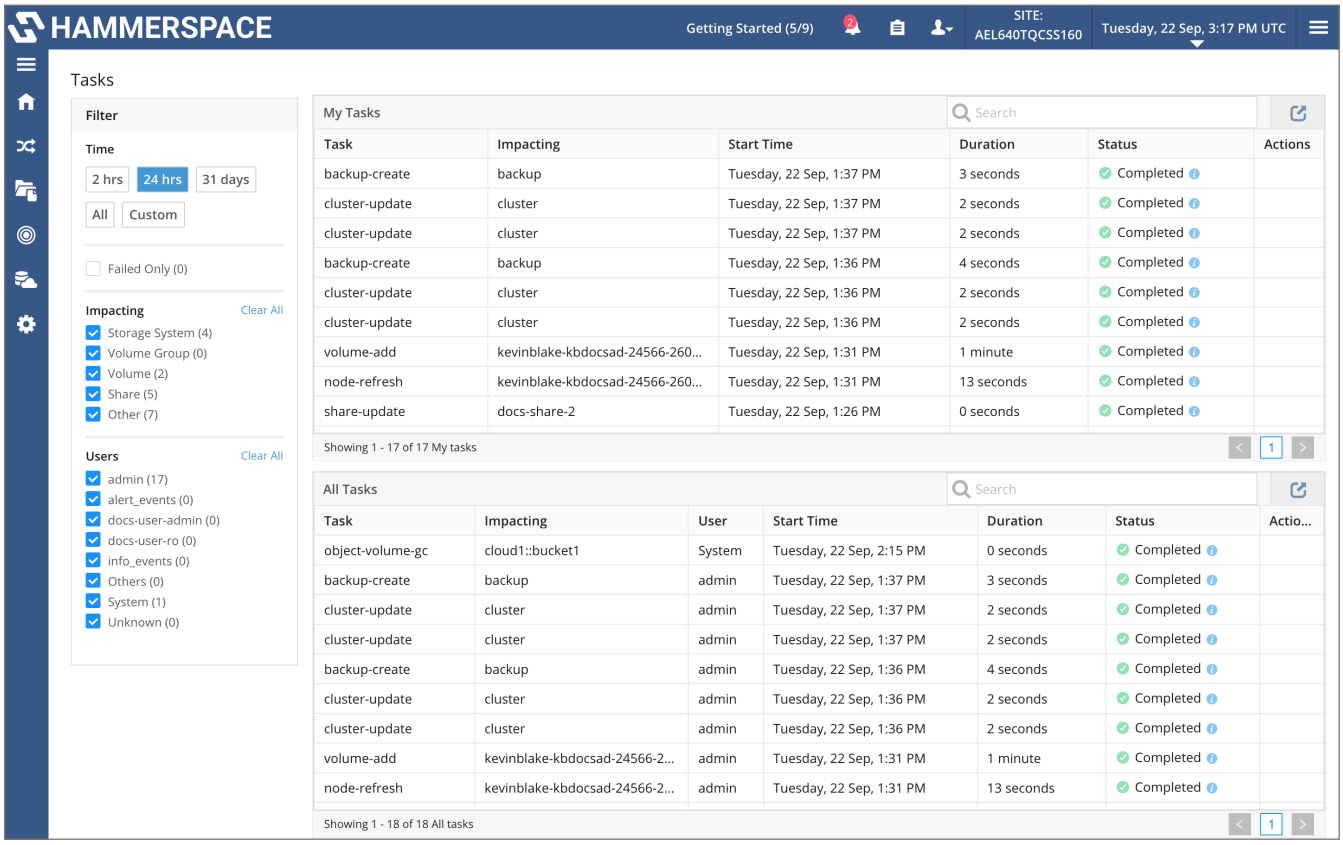


Figure 6. Viewing task activity in the task log

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.
- Duration: How long the task has been running, or how long it took to finish.
- 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.

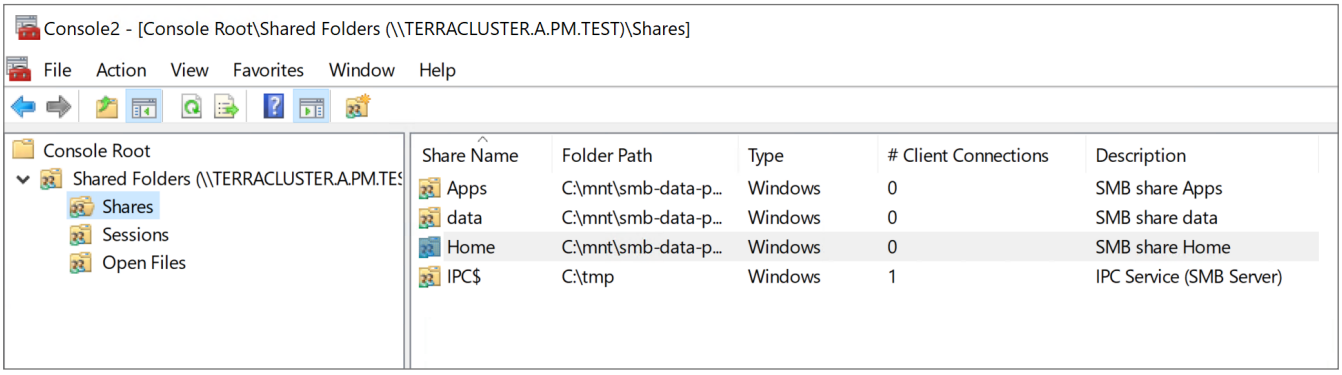


Figure 7. Connect to the cluster

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

© 2026 - Hammerspace

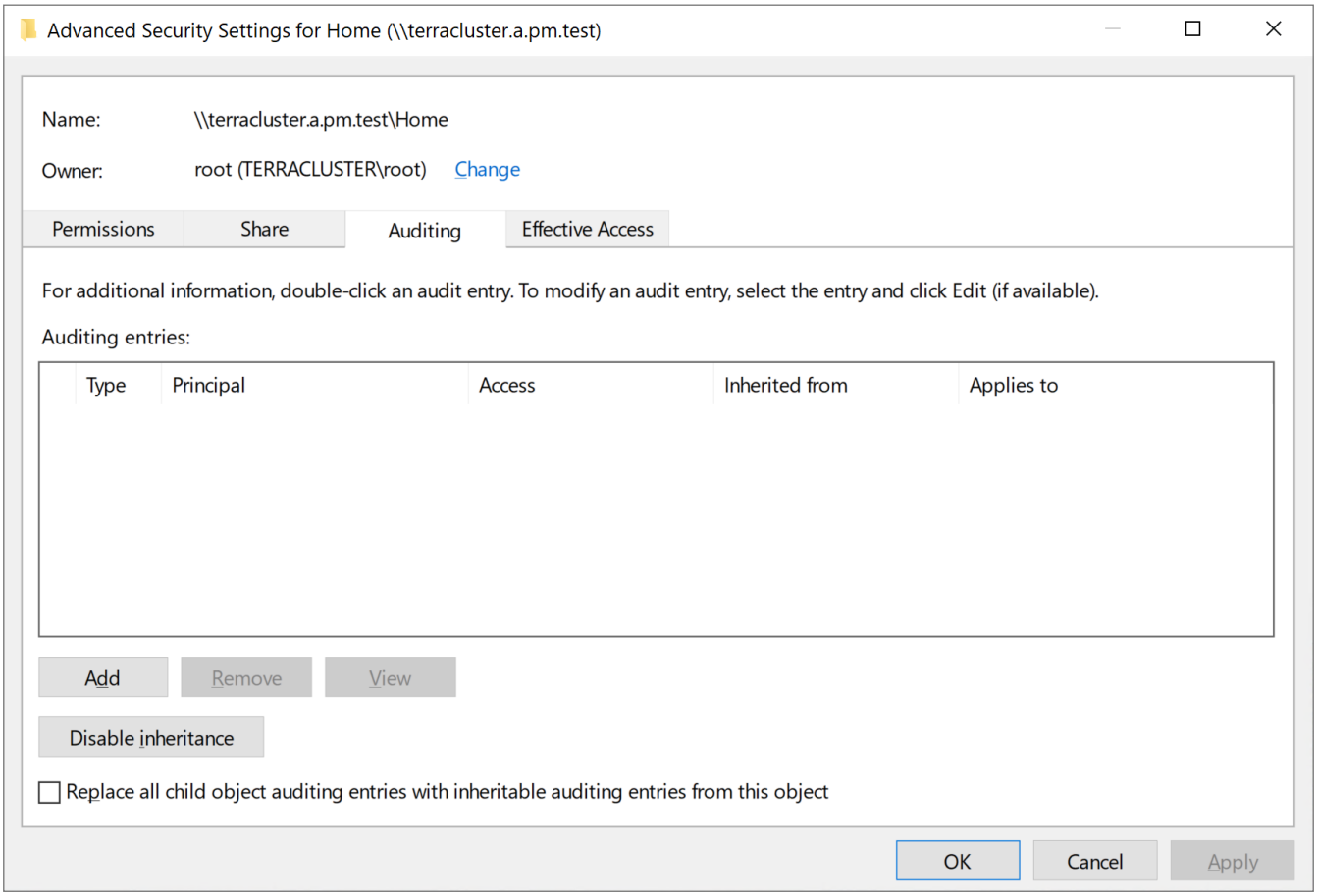


Figure 8. Advanced security properties

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.

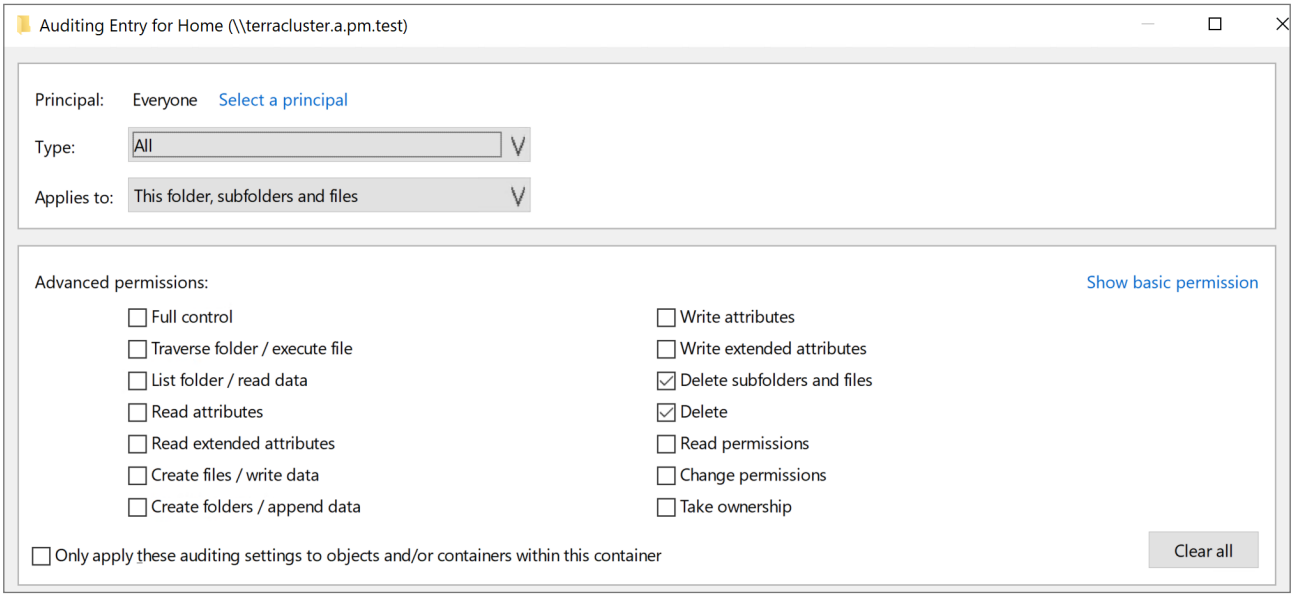


Figure 9. Select auditing specifics

© 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

- Mount by DNS hostname. A mount by IP address uses Kerberos only when the client can resolve the address to the server's DNS hostname and `rpc.gssd` is running on the client; otherwise Kerberos authentication fails.
- Currently, Hammerspace only supports Microsoft Active Directory Key Distribution Centers (KDCs).
- Kerberos over NFSv3 and NFSv4.1 works only with shares as exposed by the data portals, and supports **krb5**, **krb5i**, and **krb5p**.
- Kerberos over NFSv4.2 works through the Anvil, not through the data portals, and supports **krb5** only. **krb5i** and **krb5p** are not supported over NFSv4.2.
- The Portal DNS name must resolve in DNS, even on a site that uses only NFSv4.2. If it does not resolve, large numbers of concurrent NFSv4.2 Kerberos mounts can fail.
- A share whose export rule lists **sys** together with Kerberos flavors also accepts AUTH_SYS mounts. To require Kerberos, remove **sys** from the export rule.

© 2026 - Hammerspace

- 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.

7.2.2. NFS Protocol Support

NFS Version and Security Modes:

- Kerberos security is available two ways. NFSv4.1 through data portals supports **krb5**, **krb5i**, and **krb5p**. NFSv4.2 through the Anvil supports **krb5** only. Use NFSv4.1 through data portals when you need integrity checking (**krb5i**) or privacy (**krb5p**).
  - NFSv4.2 Kerberos mounts go to the Anvil and do not use a data portal. The procedure is in Mounting a Share Using Kerberos Security, later in this section.
  - NFS 4.1 data portals 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).
- Kerberos over NFSv4.2 supports **krb5** only. **krb5i** and **krb5p** are not supported over NFSv4.2.
- Kerberos mounts use a different DNS name depending on the NFS version. NFSv4.2 Kerberos mounts use the **Anvil DNS name**; NFSv3 and NFSv4.1 Kerberos mounts use the **Portal DNS name**. Both names must be registered in DNS. See Joining an Active Directory Domain for the two names, their defaults, and the records to create.

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 are created automatically when Hammerspace joins the domain. To re-create them while the system remains joined, use `ad-config --recreate-nfs-spns`. 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.

- To check the SPNs in the Management GUI, go to **Administration › Directory Services**, click the **Edit** action for the Active Directory domain, and open the **Kerberos** tab, which lists **Missing SPNs** and **Registered SPNs**. The GUI cannot register missing SPNs; use the Admin CLI command above.

© 2026 - Hammerspace

- 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 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.

> Kerberos-secured NFS with cross-protocol user mapping requires the Active Directory domain to be configured with the RFC2307 (or RFC2307BIS) schema. If you enable Kerberos on a share while the domain is not configured for RFC2307(bis), Hammerspace still allows the configuration, but UID/GID user mapping does not function and Hammerspace raises the informational event **Kerberos user-mapping is nonfunctional** (`KERBEROS_USER_MAPPING_NONFUNCTIONAL`) for the share. The event clears when the AD LDAP schema is set to RFC2307 or RFC2307BIS.

> When the RFC2307 or RFC2307BIS schema is in use, Hammerspace relies on the Active Directory domain controller to resolve SIDs to the UID/GID values used for cross-protocol (NFS/SMB) ID mapping. This resolution can fail if the domain controller has the RC4_HMAC_MD5 Kerberos encryption type disabled — for example, by a group policy that enforces an AES-only policy.
>
> If a group policy disables RC4_HMAC_MD5 on the domain controller, Hammerspace may be unable to map domain users and groups to their UID/GID. Affected users resolve to `nobody`, and shares that rely on ID mapping become inaccessible to them.

To confirm the symptom, run `hs-ctl cred-info`. When ID mapping has failed, the user and group resolve to `nobody`:

```
user: UID:123477 (...) - nobody
group: GID:10001 (...) - nobody
```

A working configuration resolves to the domain principal instead — for example, `adusertest10006@ad.local` and `Domain Users@ad.local`.

To remediate, re-enable the RC4_HMAC_MD5 encryption type on the domain controller and apply the change:

© 2026 - Hammerspace

```
gpupdate /force
```

ID mapping then resolves correctly.



Disabling RC4_HMAC_MD5 is a common hardening measure, so re-enabling it domain-wide may not be acceptable in your environment. If you cannot re-enable it, contact Hammerspace Support for the supported workaround.

- 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.

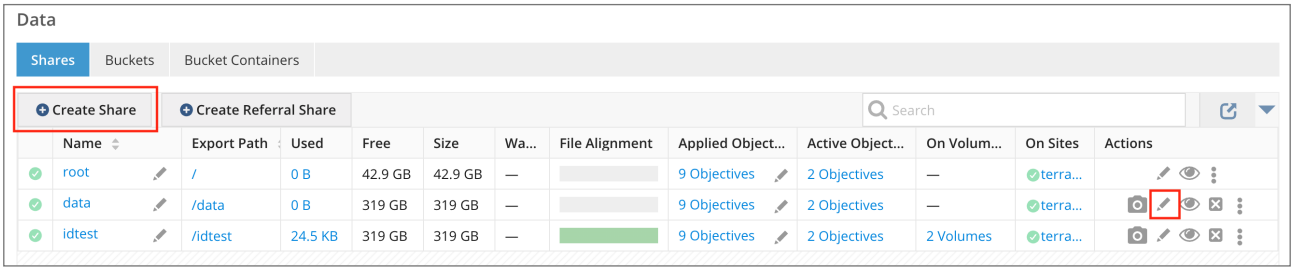


Figure 10. Create a new share or edit the configuration of an existing share

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.

© 2026 - Hammerspace

4. Click **Add Export Option** or the pencil icon, as appropriate, and enter the requested details.
   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.

© 2026 - Hammerspace

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 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.

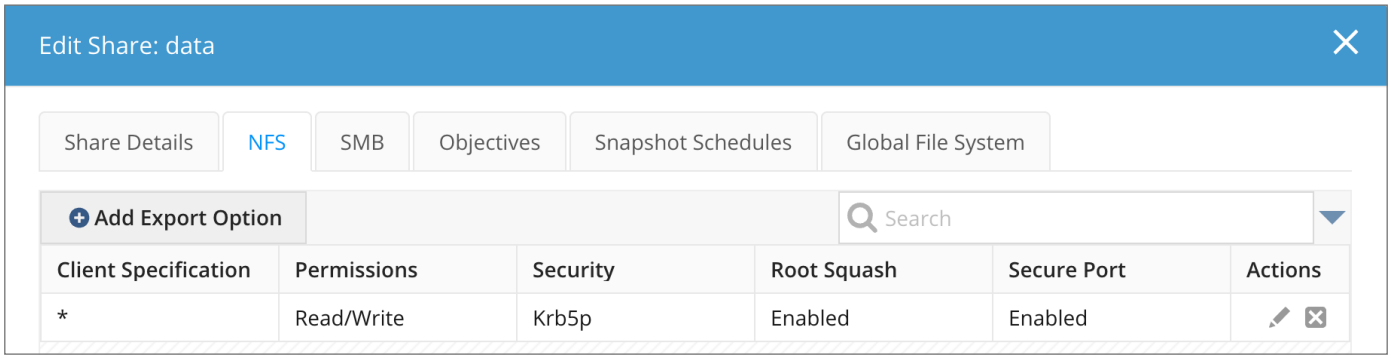


Figure 11. Exports list

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"):

© 2026 - Hammerspace

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

NFS SPNs are created automatically when Hammerspace joins the Active Directory domain. To re-create them while the system remains joined to the domain, use the following command. The password is the one used to join the domain.

Command:

```
ad-config --recreate-nfs-spns --password <password-used-to-join-the-domain>
```

In the Management GUI

In Hammerspace 5.3, the Management GUI cannot register missing NFS SPNs, because it has no field for the password used to join the domain. The **Kerberos** tab of the Active Directory entry under **Administration › Directory Services** lists **Missing SPNs** and **Registered SPNs**. To register missing SPNs, use the Admin CLI command above, or leave and rejoin the domain.



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

© 2026 - Hammerspace

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
```

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.

A join from the CLI (`ad-config --join`) does not turn on floating-IP registration. Hammerspace recommends turning it on. You can include `--dns-auto-register-floating-ips-enable` in the `ad-config --join` command itself; a separate command is not needed.

When the Portal DNS name is served only by static DSX data IP addresses, floating-IP registration does not cover it. Add those addresses with `--dns-configured-ip-add` so that the Portal DNS name is registered.

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
  ```


© 2026 - Hammerspace

```
        forwardable = true

udp_preference_limit = 0

[realms]

WIN.AD.TEST = {

        kdc = win2012r21.win.ad.test

        admin_server = win2012r21.win.ad.test

}

[domain_realm]

        .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]
```


© 2026 - Hammerspace

```
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

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.

These requirements also apply to clients that mount shares over NFSv4.2 with Kerberos.

For NFSv4.2 Kerberos mounts to map users and groups to UIDs and GIDs, the Active Directory domain must use the RFC2307 or RFC2307bis schema, selected under **SMB ↔ NFS User Mapping** when the cluster joins the domain. Either schema works.

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.

Mounting a Share over NFSv4.2 with Kerberos

Starting with Hammerspace 5.3, Linux clients can mount a share over NFSv4.2 with Kerberos authentication. NFSv4.2 Kerberos mounts go to the Anvil, not to a data portal, and support the **krb5** security flavor only. A mount that requests **krb5i** or **krb5p** over NFSv4.2 fails with `access denied by server`. When you need integrity checking or privacy, mount over NFSv4.1 through a data portal instead.

Before you mount, make sure that:

- The cluster is joined to the Active Directory domain, and the Anvil DNS name is registered in DNS. The Anvil DNS hostname must resolve to the cluster floating IP address on the active Anvil node, with a matching PTR record. Without the PTR record, the mount fails with `access denied by server`. See Joining an Active Directory Domain.
- The share's export rule includes **krb5**. Kerberos flavors are not enabled in export rules by default. If the export rule does not include **krb5**, a mount with `sec=krb5` fails.
- The client is configured as described in Client Configuration.
- The clocks of the client, the cluster, and the Active Directory domain controller agree to within 5 minutes, the default Kerberos clock-skew tolerance in Active Directory.

Mount by the Anvil DNS hostname. The client finds the Kerberos service principal (`nfs/<hostname>@<REALM>`) by looking up the server's address in reverse DNS, which is why the PTR record must return the Anvil DNS hostname.

You can also mount by the cluster IP address, provided the client can resolve that address to the Anvil DNS hostname and `rpc.gssd` is running on the client. These are client configuration requirements. Mounting by the Anvil DNS hostname is the usual form.

To mount a share over NFSv4.2 with Kerberos:

1. As the user who will access the share, obtain a Kerberos ticket.

   Command:

   ```
   $ kinit <username>@<REALM>
   ```

2. Mount the share with `vers=4.2` and `sec=krb5`, using the Anvil DNS hostname.

   Command:

   ```
   $ sudo mount -t nfs -o vers=4.2,sec=krb5 <anvil-dns-hostname>.<domain>:/<share> /<mountpoint>
   ```


© 2026 - Hammerspace

3. Confirm that the mount uses NFSv4.2 and Kerberos.

   Command:

   ```
   $ mount | grep /<mountpoint>
   ```

   Expected output:

   ```
   anvil-dns.ad.example.com:/docs-krb5 on /mnt/docs-krb5 type nfs4
   (rw,relatime,vers=4.2,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=k
   rb5,clientaddr=192.0.2.31,local_lock=none,addr=192.0.2.10)
   ```

   The options for the mount include `vers=4.2` and `sec=krb5`.

An NFSv4.2 Kerberos mount works again after the active Anvil node reboots and the cluster returns to a healthy state; you do not need to mount the share again. Files written over it can be read over NFSv3, over NFSv4.1 with Kerberos, and over SMB.

When the user's Kerberos ticket expires, NFS operations on the mount fail with a "Key has expired" error. Run `kinit` again to obtain a new ticket.

7.3. Configuring an LDAP Name Service

Hammerspace 5.3 and later can use one or more existing LDAP directories to resolve Unix user and group identities (UIDs and GIDs) on the NFS data path. Each directory you configure is called an *LDAP name service*. You can add a name service from the GUI, the Admin CLI (`name-service-config`), or the REST API. When you configure more than one, you set the order in which Hammerspace queries them.

After you add an LDAP name service, users in that directory can access files on NFSv4.2 mounts with the permissions their UID and GID grant.

Hammerspace does the lookups through SSSD (the System Security Services Daemon) on the Anvil nodes — the metadata servers of the cluster. Hammerspace writes and manages the SSSD configuration for you; you do not configure SSSD yourself.

In Hammerspace 5.3, an LDAP name service and Active Directory cannot be configured on the same cluster. See Important Information.



This feature is distinct from the identity provider (IdP) used to log in to the Hammerspace management GUI, Admin CLI, and API. If you want LDAP for both NFS identity resolution and administrator login, configure the directory twice — once in **Administration › Directory Services** (covered here) and once in **Administration › Authentication**. The two configurations are independent.

This section includes the following subsections:

© 2026 - Hammerspace

- Prerequisites
- Important Information
- Limitations
- Best Practices
- Securing LDAP Connections with LDAPS or StartTLS
- How Hammerspace Manages SSSD on the Anvils
- Procedures
- Admin CLI Reference: name-service-config
- REST API Reference
- Troubleshooting

7.3.1. Prerequisites

- Your organization has an operational LDAP directory. You know its domain name (for example, `corp.example.com`) and the IP address or DNS name of at least one LDAP server.
- The LDAP server's root DSE lists a naming context that matches the domain name in domain-component form. For domain `corp.example.com`, the server must serve `dc=corp,dc=example,dc=com`. Hammerspace checks this during the connection test, whatever search base you configure.
- If your LDAP server does not allow anonymous searches, you have a Bind DN and Bind Secret (password) for an account with read access to the user and group entries.
- User and group entries use the RFC 2307 or RFC 2307bis schema. User entries must have `uidNumber` and `gidNumber` attributes, and Hammerspace matches user names against the `uid` attribute.
- The Anvil nodes can reach the LDAP server. If you specify a port, open that port. If you do not specify a port, Hammerspace tries port 389 and then port 636 for each transport mode, so allow both.
- If you plan to use LDAPS or StartTLS, the CA certificate that issued the LDAP server's certificate is in the Hammerspace trust store before you add the name service, and the server certificate lists the address you will configure in its Subject Alternative Name (SAN) extension. See Securing LDAP Connections with LDAPS or StartTLS.
- The cluster is not joined to Active Directory. In Hammerspace 5.3, LDAP name services and Active Directory cannot be configured together.

7.3.2. Important Information

Identity mapping only — not administrator authentication. The LDAP name service resolves Unix user and group identities (UID and GID) for the NFS data path. It does not authenticate administrators logging in to the Hammerspace GUI, Admin CLI, or API.

Hammerspace manages SSSD for you. When you add, update, reorder, or remove a name service, Hammerspace regenerates the SSSD configuration and distributes it to every Anvil. Do not edit SSSD files on an Anvil; Hammerspace overwrites them. For how the configuration is owned, and why a missing `/etc/sssd/sssd.conf` can be normal, see How Hammerspace Manages SSSD on the Anvils.

© 2026 - Hammerspace

Checking lookups from the Anvil. Standard Linux commands such as `getent passwd` and `id`, run in a shell on an Anvil, do not resolve LDAP users. Hammerspace resolves names for NFS inside its own services, not through the Anvil's system name-service switch. To check that identities resolve, use **User/Group Lookup** in the GUI or the `--resolve-user`, `--resolve-group`, and `--resolve-user-groups` options of `name-service-config`. On NFS client machines, `getent` and `id` work as usual.

LDAPS and StartTLS certificates are checked differently by the connection test and at run time. The connection test that runs when you add or update a name service checks only that the server certificate names the address you configured. Lookups made after the name service is saved use SSSD, which also requires the certificate to chain to a CA that the Anvil trusts. A name service can therefore pass the connection test and still fail every lookup. Add the issuing CA to the Hammerspace trust store before you add the name service. See Securing LDAP Connections with LDAPS or StartTLS.

LDAP and Active Directory are mutually exclusive in 5.3. A cluster that has joined Active Directory cannot use LDAP name services, and a cluster that has an LDAP name service cannot join Active Directory. To switch, remove every LDAP name service before joining Active Directory, or leave Active Directory before adding an LDAP name service. In the GUI, **Add LDAP Nameserver** is unavailable while Active Directory is joined, and **Add Active Directory** is unavailable while any directory service is configured.

The bind secret is stored obfuscated. If you configure a Bind DN and Bind Secret, Hammerspace stores the secret in obfuscated form in its database and in the SSSD configuration on the Anvils, never returns it in GUI, CLI, or API output, and replaces it with `REDACTED` in support bundles. When you pass the secret to `name-service-config` with `--bind-secret`, it is part of the command line, so use an account whose only permission is to read the directory.

Resolution order matters when more than one domain is configured. SSSD queries the domains in the configured order. If the same unqualified user or group name exists in more than one domain, the entry from the first domain in the order is used. Qualify names (`user@domain`) or configure the order carefully if your domains have overlapping names.

Health monitoring can take up to 24 hours to report a failure. Once a day, Hammerspace tests the connection to every name service. A service that fails this test changes to operational state `DOWN`, and a `NAME_SERVICE_UNHEALTHY` event is raised. Every 10 minutes, Hammerspace re-tests services that are not `UP`, and returns a service to `UP` (clearing the event) when its test passes. A server that stops responding can therefore show `UP` for up to 24 hours. To check a name service immediately, see Testing the Connection to a Name Service.

7.3.3. Limitations

- **NFSv4.2 only.** LDAP identity mapping is supported on the NFSv4.2 data path only. NFSv3, NFSv4.1, SMB, and S3 are not supported.
- **Cannot coexist with Active Directory in 5.3.** LDAP name services and Active Directory cannot be configured on the same cluster at the same time.
- **RFC 2307 and RFC 2307bis schemas only.** LDAP entries must include `uidNumber` and `gidNumber` attributes. Other schemas are not supported.
- **Anonymous and simple bind only.** Kerberos (GSSAPI) and other SASL bind methods are not supported.

© 2026 - Hammerspace

- **The domain name cannot be changed.** After a name service is created, its domain name is fixed. To use a different domain name, remove the name service and add it again.
- **One domain per name service, and one name service per domain.** Two name services cannot use the same domain name, even with different search bases.
- **One server address when the connection test runs.** Adding or updating a name service with more than one server address fails when the connection test runs. The GUI always runs the connection test, so in the GUI a name service can have only one address. To configure more than one address, use the Admin CLI with `--no-connection-test`, `--transport-mode`, and a port on every address. See Adding an LDAP Name Service.
- **Certificate checking cannot be relaxed.** SSSD always requires a trusted LDAP server certificate for LDAPS and StartTLS. This is not configurable.
- **Cache tuning is not configurable.** SSSD cache timeouts and refresh intervals are not exposed for configuration in 5.3.
- **SSSD log verbosity is not configurable.** SSSD debug levels cannot be set through the GUI, Admin CLI, or API in 5.3.
- **There is no LDAP cache flush.** In 5.3, **Clear Cache** in **Administration › Directory Services** clears Hammerspace's own identity-mapping caches but does not expire entries that SSSD has cached from LDAP. Changes made in the LDAP directory are seen when SSSD's cached entries expire.
- **`getent` and `id` do not resolve LDAP users on the Anvil.** See Important Information.

If a configuration change fails to apply, lookups stop until Hammerspace retries. In 5.3, if SSSD cannot be restarted with a new configuration on the primary Anvil, Hammerspace removes `/etc/sssd/sssd.conf` from that Anvil and raises an `SSSD_CONFIG_SYNC_FAILED` event. LDAP lookups fail until the next attempt, which Hammerspace makes every 2 minutes, succeeds and the event clears. If you see lookup failures immediately after adding, updating, or removing a name service, check for this event.

7.3.4. Best Practices

Name service resolution order. The resolution order affects both performance and which entry is used when a name exists in more than one domain. Put the domain that holds most of the commonly used names first, rather than simply the closest server. If you have duplicate names across domains, configure the order so that the entry you want is found first, or have users and applications use qualified names.

If several servers hold most of the same names, identify the fastest connection and order the name services accordingly. If you experience identity-lookup delays, contact Hammerspace Support for troubleshooting assistance.

Use LDAPS or StartTLS, and configure the server by the name on its certificate. Plain LDAP sends lookups, and the bind secret if you use one, in clear text. With LDAPS or StartTLS, configure the server address exactly as it appears in the certificate's Subject Alternative Name (SAN) extension — usually the server's DNS name. See Securing LDAP Connections with LDAPS or StartTLS.

Set the transport mode explicitly in production. If you leave the transport mode unset, the connection test chooses the first mode that works, trying StartTLS, then LDAPS, then plain LDAP. If the secure modes are blocked or misconfigured — including when the server certificate does not name the address you configured — the test falls back to plain LDAP if the server accepts it, and the name service is saved unencrypted. Setting the

 © 2026 - Hammerspace

transport mode makes the add or update fail instead.

Use a read-only bind account. If your directory requires a Bind DN, use a dedicated account that can only read the user and group entries.

Verify before you rely on the configuration. After adding or changing a name service, look up a known user and group with **User/Group Lookup** or `--resolve-user` before users mount shares. The lookup exercises the same SSSD path that NFS uses; the connection test does not.

7.3.5. Securing LDAP Connections with LDAPS or StartTLS

Hammerspace can connect to an LDAP server in one of three transport modes:

Transport mode	What it does
`LDAP`	Plain LDAP. Nothing is encrypted, including the bind secret if you use one. No certificate is involved.
`LDAPS`	LDAP over TLS from the start of the connection (`ldaps://`), conventionally on port 636.
`STARTTLS`	Opens a plain LDAP connection (`ldap://`), conventionally on port 389, and upgrades it to TLS before any directory data is exchanged.

With LDAPS or StartTLS, the LDAP server's certificate must be acceptable to two different checks, described below. Most failures with a secure transport come from satisfying the first check but not the second.

What Hammerspace Checks

The connection test checks the server's identity only. The connection test runs when you add or update a name service, when you click **Test Connection**, when you run `name-service-config --test-connect`, and in the scheduled health checks. It accepts the certificate if a Subject Alternative Name (SAN) in it matches the server address you configured:

- A DNS name in the SAN must match a configured host name. A wildcard such as `*.example.com` matches one level only (`ldap1.example.com`, but not `ldap1.east.example.com`).
- An IP address in the SAN must match a configured IP address.
- The certificate's Common Name (CN) is not considered. A certificate without a matching SAN fails the test.

The connection test does not check the certificate chain, the signature, or the validity dates. A self-signed or expired certificate passes it as long as the SAN matches.

SSSD checks the whole certificate at run time. Lookups made after the name service is saved — by NFS and by **User/Group Lookup** — go through SSSD on the Anvil. SSSD requires the certificate to be valid and to chain to a CA certificate in the Anvil's system trust store, and this requirement cannot be relaxed. Hammerspace fills the Anvil's system trust store from its own trust store, which you manage with `cert-add` or the **Trusted Certificates** tab.

Because the two checks differ, a name service can pass the connection test, show operational state `UP`, and still

fail every lookup. The scheduled health checks use the connection test too, so they do not detect this.

Trusting the LDAP Server Certificate

Add the certificate of the CA that issued the LDAP server's certificate to the Hammerspace trust store **before** you add the name service.

1. Obtain the issuing CA certificate in PEM format. Use the root CA certificate where possible. The root CA alone is enough when the LDAP server sends its intermediate CA certificates along with its own certificate, which is the usual configuration.
2. Add it to the Hammerspace trust store in one of these ways:
   - In the GUI, on the **Trusted Certificates** tab of **Administration › TLS**. See Add a Trusted CA Certificate Using the Management GUI. In Hammerspace 5.3, the GUI adds root CA certificates only.
   - In the Admin CLI, with `cert-add --pem "<pem_content>"`. To add an intermediate CA certificate, put the intermediate certificate and its root together in `<pem_content>` (intermediate first) and add `--trust-intermediate-ca`; an intermediate given on its own is rejected, even when its root is already in the trust store. If the LDAP server uses a self-signed certificate that is not a CA certificate, add that certificate itself with `--trust-self-signed-certificate`. See Manage Certificates and CAs.
3. Add the name service, with the server address set to a name or IP address that appears in the server certificate's SAN. See Adding an LDAP Name Service.
4. Verify with **User/Group Lookup** or `name-service-config --resolve-user`. A successful connection test alone does not prove that the certificate is trusted.

Hammerspace copies the trust store to every node and adds its certificates to each node's operating-system trust store. You cannot add the LDAP server's own certificate when it is issued by a CA (a CA-signed leaf certificate); `cert-add` rejects it with `Certificate is not a CA or self-signed`. Add the issuing CA instead.



If you add a CA certificate to fix a name service that already exists, you do not need to change or re-add the name service: SSSD picks up the new CA on its own, and lookups succeed within about a minute. Check with **User/Group Lookup**.

Choosing the Server Address

The address you configure is used both for the connection-test identity check and by SSSD, so:

- Configure the server by the DNS name in its certificate's SAN, and make sure the Anvil nodes can resolve that name in DNS.
- Configure the server by IP address only if the certificate carries that IP address in its SAN.

What You See When the Certificate Is Wrong

The certificate does not name the configured address. The secure transport modes fail the connection test. What happens next depends on whether you set a transport mode:

- If you set `LDAPS` or `STARTTLS`, the add or update fails with `Failed to connect to LDAP server '<domain>' using address '<address>' with transport mode <mode>`. The message does not give the reason.

© 2026 - Hammerspace

- If you left the transport mode unset and the server also accepts plain LDAP, the test falls back to `LDAP`, and the name service is added with transport mode `LDAP` — unencrypted. The failed secure attempts are listed with the result (in the GUI, click **Show Details**; in the Admin CLI, under `Connection attempts:`), and a SAN mismatch is reported there as `Server identity mismatch: expected <address>, but certificate SANs were: <SAN list>`. Check the **Transport mode** of the name service after adding it.

The fallback happens only when the SAN does not match. When the SAN matches but the certificate is not trusted, the test keeps the secure transport mode (`STARTTLS` when the server offers it) and the name service is added — and its lookups then fail as described next.

Reconfigure the name service with an address that the certificate names, or have the certificate reissued with the address in its SAN.

The certificate is not trusted, has expired, or has a broken chain. The connection test passes and the name service shows `UP`, but lookups fail: **User/Group Lookup** and `--resolve-user` report `User '<user>@<domain>' not found`, with no error text, and NFS users are not mapped to their UIDs and GIDs. Users and groups that SSSD has already cached from earlier lookups keep resolving until their cache entries expire (the SSSD default is 90 minutes), so the symptom is often that new users are not found while existing users still work. Add the issuing CA to the trust store as described above.

7.3.6. How Hammerspace Manages SSSD on the Anvils

Hammerspace writes the SSSD configuration on the Anvils itself, from the name services and Active Directory settings you configure. This section explains what the configuration looks like in each state, so that you can tell a normal state from a fault when you or Hammerspace Support inspect an Anvil.

Configuration Files

Hammerspace manages two files in `/etc/sssd` on each Anvil:

- `/etc/sssd/sssd.conf` — the SSSD configuration. When Hammerspace generates it from your LDAP name services, its first line is `# hammerspace-managed-by = nameservice`. It has one `[domain/<domain name>]` section per name service, listed in resolution order. The file is readable by `root` only, and a bind secret in it is obfuscated.
- `/etc/sssd/target-sssd-state` — contains `enabled` or `disabled`, the state Hammerspace wants SSSD to be in.

Do not edit either file. Hammerspace rewrites `sssd.conf` whenever the configuration changes and also once a day, and a manual change is lost.

Configuration States

What is configured	/etc/sssd/sssd.conf	SSSD
One or more LDAP name services	Generated by Hammerspace, with the `hammerspace-managed-by` line	Running on the primary Anvil

© 2026 - Hammerspace

What is configured	/etc/sssd/sssd.conf	SSSD
No LDAP name service; Active Directory joined with the RFC2307 or RFC2307BIS schema	Written by the Active Directory integration (`id_provider = ad`), without the `hammerspace-managed-by` line	Running
No LDAP name service, and Active Directory not joined or joined without an ID-mapping schema	Absent, or left from an earlier Active Directory join (see below)	Stopped and disabled

An absent `sssd.conf` is correct in the last state; it does not by itself indicate a fault. When you remove the last LDAP name service, Hammerspace removes the file and stops SSSD.

In Hammerspace 5.3, SSSD runs only on the primary Anvil while LDAP name services are configured. The other Anvils receive the same `sssd.conf`, but SSSD is not started on them, so finding SSSD stopped on a secondary Anvil is expected. SSSD is started on the new primary Anvil after a failover, and lookups resume as soon as the management service is available there, typically within a few minutes.

After you leave Active Directory, `sssd.conf` can still contain the Active Directory domain's section. This is expected in 5.3: SSSD is stopped when you leave, and the file is replaced when you add an LDAP name service.

When a Sync Fails

If Hammerspace cannot copy the configuration to an Anvil, or cannot start SSSD with it, it raises an `SSSD_CONFIG_SYNC_FAILED` event naming that Anvil and retries every 2 minutes. The event clears when a retry succeeds. See Limitations for what happens to lookups meanwhile.

7.3.7. Procedures

The following procedures describe how to add, verify, test, view, reorder, update, and remove LDAP name services for NFS identity resolution. Use them to configure one or more LDAP domains, confirm that user and group lookups resolve correctly, and adjust the resolution order as your environment changes.

This section includes the following subsections:

- Adding an LDAP Name Service
- Verifying the Configuration (User/Group Lookup)
- Testing the Connection to a Name Service
- Viewing Configured Name Services and Resolution Order
- Changing the Directory Services Resolution Order
- Updating an LDAP Name Service
- Removing a Name Service

© 2026 - Hammerspace

Adding an LDAP Name Service

When you add a name service, Hammerspace runs a connection test before saving it: it connects to the server, binds with the Bind DN if you gave one, checks that the server's root DSE lists a naming context for the domain, and searches the search base. If you do not specify a transport mode or port, the test tries each transport mode in the order StartTLS, LDAPS, plain LDAP, and for each mode tries port 389 and then port 636. It keeps the first combination that works and saves it with the name service. If no combination works, the name service is not added.

If the name service will use LDAPS or StartTLS, add the LDAP server's issuing CA certificate to the Hammerspace trust store first. See Securing LDAP Connections with LDAPS or StartTLS.

Using the GUI

1. Go to **Administration** › **Directory Services**. The table shows any configured directory services.
2. Click **Add LDAP Nameserver**. The **Add LDAP Name server** dialog opens on the **Details** tab.

   Add LDAP Nameserver is unavailable while the cluster is joined to Active Directory. Likewise, **Add Active Directory** is unavailable while an LDAP name service is configured.

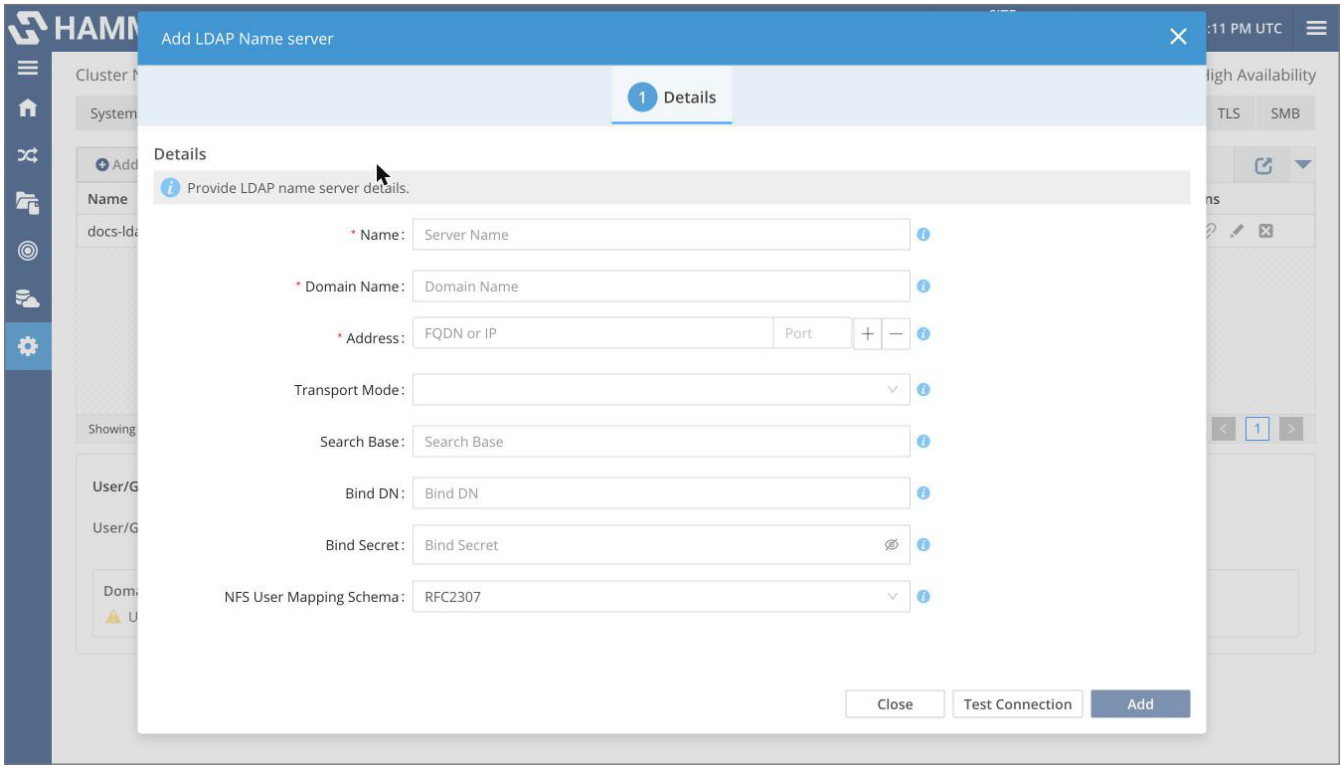


Figure 12. The Add LDAP Name server dialog

3. Enter the name service details. Only **Name**, **Domain Name**, and **Address** are required. **Add** becomes available when these three are filled in.

   Name

   A management label for the name service. It does not need to match the domain name, must be unique (case-insensitive), and can be up to 255 characters. It cannot be changed later in the GUI.

© 2026 - Hammerspace

Domain Name

The LDAP domain, for example `corp.example.com`. It must be the directory's own suffix in dotted form (`corp.example.com` for a directory whose naming context is `dc=corp,dc=example,dc=com`), whatever the field's tooltip example suggests; any other value fails the connection test. Until you edit this field, it is filled in from **Name**. It must be unique (case-insensitive) and cannot be changed later.

Address

The IP address or DNS name of the LDAP server, and optionally its port (1–65535) in the second box of the same row. If you leave the port empty, the connection test finds it. For LDAPS or StartTLS, enter an address that appears in the server certificate's Subject Alternative Name. Enter one address only; see Limitations.

Transport Mode

Optional. `LDAP`, `LDAPS`, or `STARTTLS`. If you leave it empty, the connection test finds the first mode that works.

Search Base

Optional. The base DN for user and group searches, for example `ou=people,dc=corp,dc=example,dc=com`. If you leave it empty, the root of the domain (`dc=corp,dc=example,dc=com`) is used.

Bind DN

Optional. The distinguished name of the account used to search the directory. Leave it empty for anonymous access.

Bind Secret

The password for the Bind DN. Enter a Bind Secret if and only if you enter a Bind DN.

NFS User Mapping Schema

`RFC2307` (default) or `RFC2307BIS`, to match the schema your directory uses for group membership.

4. Optionally, click **Test Connection**. The result appears at the bottom of the dialog, for example `Success: Search (LDAP) base='dc=corp,dc=example,dc=com' on ldap1.corp.example.com at 10.0.0.15:389 was successful`. The test does not fill in **Transport Mode** or the port; Hammerspace records them when you add the name service.
5. Click **Add**. Hammerspace runs the connection test and, if it succeeds, adds the name service.

   When the name service is added, the dialog shows `<domain> added successfully.` If some connection attempts failed before one succeeded, click **Show Details** to see each attempt. If every attempt failed, the dialog shows `Failed to add <domain>.` and the reason; click **Previous** to correct the details.

6. Click **Close**.

The new name service appears in the Directory Services table with status `UP`. Verify it as described in Verifying the Configuration (User/Group Lookup).

© 2026 - Hammerspace

You can also start from the **Add LDAP NameServer** item in the **Getting Started** menu, which opens the same dialog. The item is unavailable while the cluster is joined to Active Directory.

Using the Admin CLI

Add a name service with the minimum required information — a management name, the domain name, and one server address. The connection test determines the transport mode and port:

```
name-service-config --add --name <management-name> --domain <domain-name> --address <IP-or-hostname>
```

Expected output:

```
ID:                     6a62676e-29d1-4591-a923-b02d1b6dfcfc
Internal ID:            65
Name:                   docs-ldap
Type:                   LDAP
Domain:                 example.com
Resolution order:       0
Oper state:             Up
Addresses:              [192.0.2.33:389]
Transport mode:         STARTTLS
Schema:                 RFC2307
Search base:            dc=example,dc=com
```

If you omit `--domain`, the management name is used as the domain name, and it must then be a valid domain name.

If some connection attempts fail before one succeeds, the command output ends with `Connection attempts:` followed by one line per attempt. Failures listed there before a success are expected.

Add a name service with an explicit transport mode and port:

```
name-service-config --add --name <management-name> --domain <domain-name> --address <IP-or-hostname>:636
--transport-mode LDAPS
```

Add a name service that binds with a Bind DN. You must give `--bind-dn` and `--bind-secret` together; the Admin CLI does not prompt for the secret:

```
name-service-config --add --name <management-name> --domain <domain-name> --address <IP-or-hostname>
--bind-dn <bind-dn> --bind-secret <bind-secret>
```

When you add a name service, if you give `--bind-dn` a plain account name instead of a distinguished name, Hammerspace expands it to `cn=<name>,<domain in dc= form>`, for example `cn=reader,dc=corp,dc=example,dc=com`.

© 2026 - Hammerspace

Add a name service that uses the RFC 2307bis schema and a specific search base:

```
name-service-config --add --name <management-name> --domain <domain-name> --address <IP-or-hostname>
--schema RFC2307BIS --search-base ou=people,dc=corp,dc=example,dc=com
```

To give a name service more than one server address, repeat `--address` once per server; do not separate addresses with commas. More than one address works only when the connection test is skipped, so also give `--no-connection-test`, `--transport-mode`, and a port on every address:

```
name-service-config --add --name <management-name> --domain <domain-name> --address
ldap1.corp.example.com:389 --address ldap2.corp.example.com:389 --transport-mode STARTTLS --no-connection
-test
```

Because nothing is tested when you skip the connection test, test the name service and look up a user afterward:

```
name-service-config --test-connect <management-name>
```

```
name-service-config --resolve-user <username>
```

Verifying the Configuration (User/Group Lookup)

After adding or changing a name service, verify that users and groups resolve correctly before relying on the configuration for file access. A lookup goes through SSSD on the Anvil, the same path NFS uses, so it also confirms that an LDAPS or StartTLS certificate is trusted.

You can look up a name in two forms:

- **Unqualified**, for example `jsmith`. Hammerspace looks the name up in every configured domain and reports a result for each one.
- **Qualified**, for example `jsmith@corp.example.com`. Hammerspace looks the name up only in that domain. If no name service is configured for the domain, the lookup fails with `Domain '<domain>' specified by user name '<name>' could not be found`.

A lookup reports what each domain returns; it does not apply the resolution order. When NFS maps an unqualified name that exists in more than one domain, the domain that comes first in the resolution order is used.

In Hammerspace 5.3, if the cluster is joined to Active Directory with an ID-mapping schema instead of using LDAP name services, the lookup queries Active Directory.

Using the GUI

1. Go to **Administration** › **Directory Services**. The **User/Group Lookup** section is below the table.

© 2026 - Hammerspace

2. In **User/Group name**, enter a user or group name, unqualified (`jsmith`) or qualified (`jsmith@corp.example.com`), and click **Lookup**.

   The GUI looks the name up both as a user and as a group.

3. Review the results. Each domain that was searched is shown under a **Domain:** heading:

   - If the name was found, the result shows **Name** (the name as it was looked up in that domain, for example `jsmith@corp.example.com`), **UID**, **GID**, and **Groups**. In **Groups**, the user's primary group can appear as its GID number or by name, and can be listed twice. A dash means the value is empty; a group has a dash for **UID**.

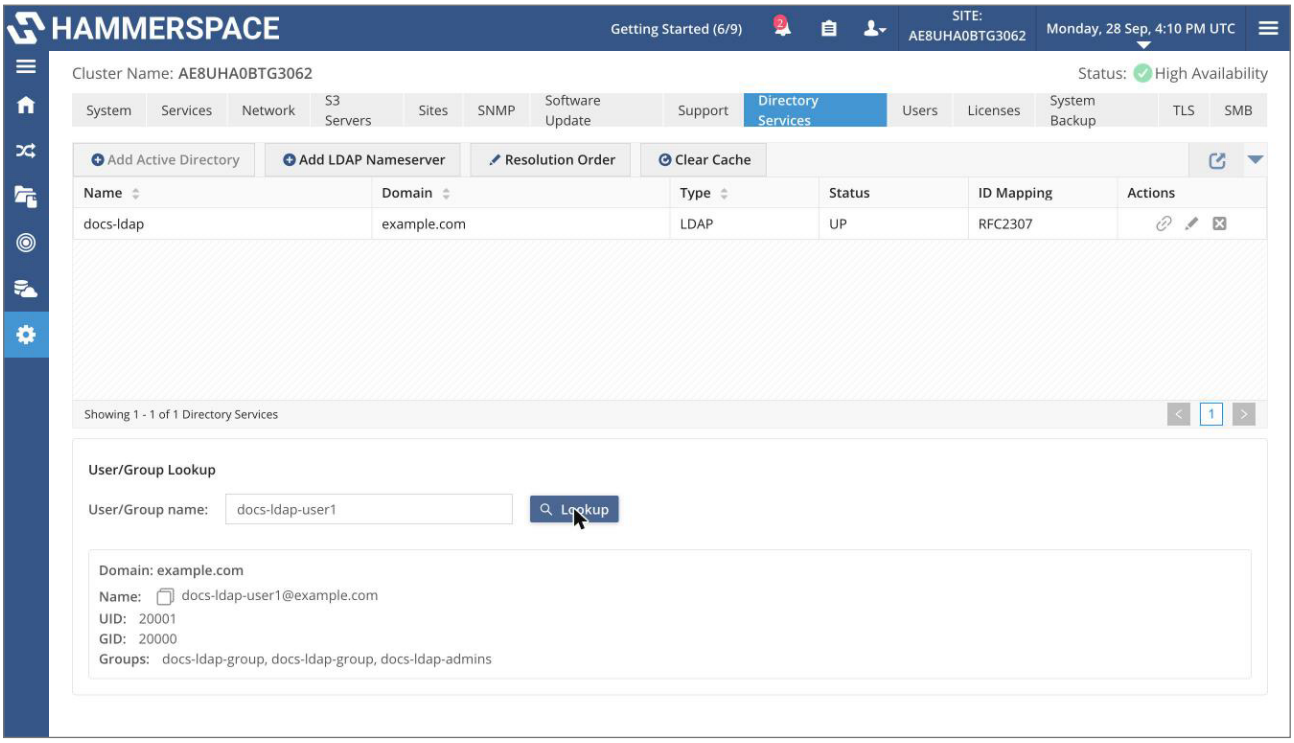


Figure 13. A user found in an LDAP name service

   - If the name was not found in that domain, the result shows `User/Group '<name>' not found`.

© 2026 - Hammerspace

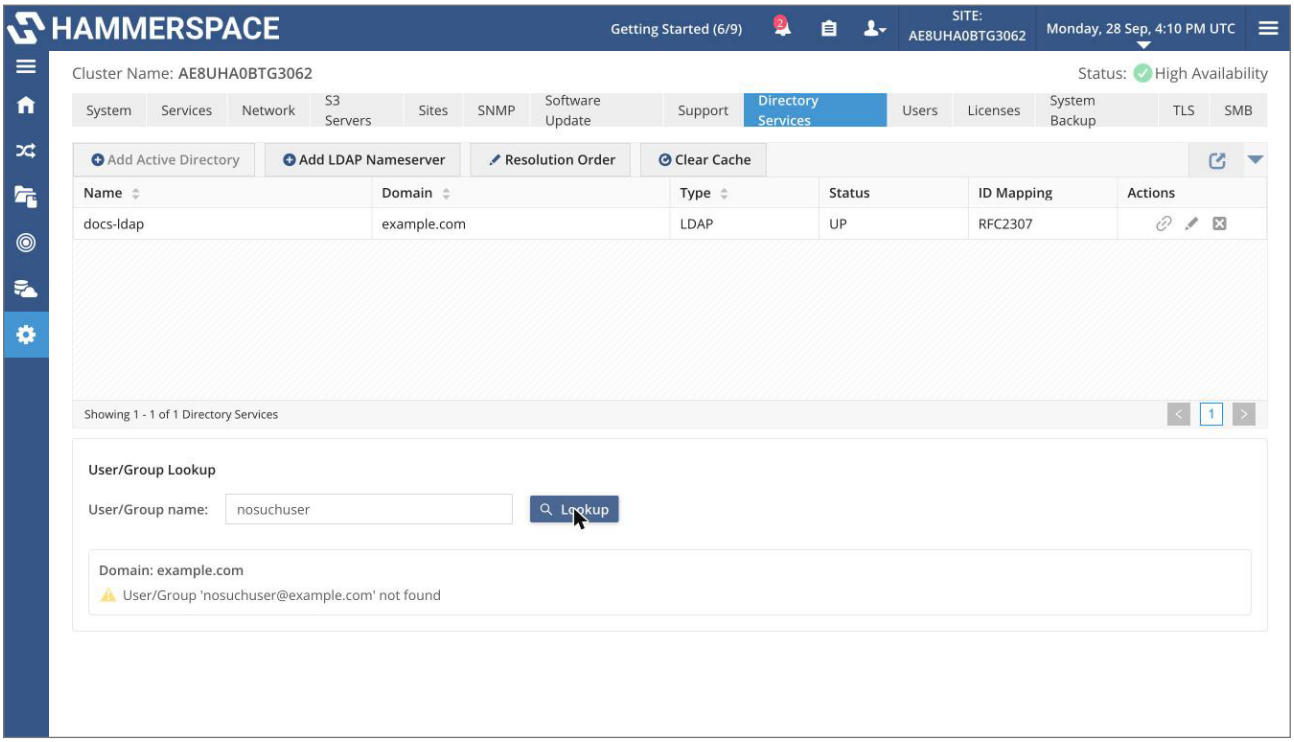


Figure 14. A name that was not found

- If no name service is configured, the lookup fails with `No Name Services are configured, cannot attempt to resolve user '<name>'`.

Using the Admin CLI

Look up a user in every configured domain (unqualified):

```
name-service-config --resolve-user <username>
```

Look up a user in one domain (qualified):

```
name-service-config --resolve-user <username>@<domain>
```

Look up a group:

```
name-service-config --resolve-group <groupname>
```

List the groups a user belongs to:

```
name-service-config --resolve-user-groups <username>
```

For each domain searched, the output shows that name service's details followed by a **Domain search result** line. For a user who is found, that line reads like the following example.

© 2026 - Hammerspace

Expected output:

```
Domain search result: User 'jsmith@corp.example.com' resolved successfully: name=jsmith, uidNumber=10001,
gidNumber=10001, gecos=John Smith
```

The result also lists `memberof=` when the directory returns group memberships for the user; `--resolve-user-groups` lists the user's groups by name as `groups=`. A name that does not exist in a domain is reported as `User '<name>@<domain>' not found` (or `Group …`) for that domain.

Testing the Connection to a Name Service

You can rerun the connection test for a name service that is already configured, for example after a network or server change, instead of waiting for the scheduled health check. The test uses the saved address, port, transport mode, search base, and Bind DN and secret. It does not change the name service or its **Status**. For a name service with more than one address, the test passes if any one address works.

The connection test does not check whether an LDAPS or StartTLS certificate is trusted; see Securing LDAP Connections with LDAPS or StartTLS. To check the full lookup path, also look up a user as described in Verifying the Configuration (User/Group Lookup).

Using the GUI

1. Go to **Administration** › **Directory Services**.
2. In the **Actions** column of the LDAP name service, click the **Test Connection** icon (the first icon).

   The **Test Connection** dialog shows the result, for example `Success: Search (STARTTLS) base='dc=corp,dc=example,dc=com' on ldap1.corp.example.com at 10.0.0.15:389 was successful`, or the reason the test failed. When the address is an IP address, the message has no `at <IP address>` part.

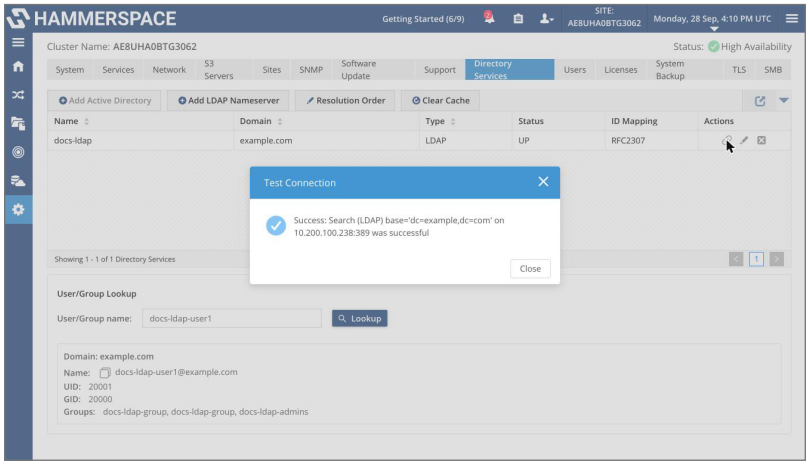


Figure 15. A successful connection test

   For a name service with more than one address, the message names only the first address that answered.

3. Click **Close**.

Using the Admin CLI

Test one name service by its management name, UUID, or internal ID:

```
name-service-config --test-connect <name>
```

Expected output (the name service's details; there is no separate success line):

```
ID:                     83be2e59-2417-4e6d-bf52-083329bde415
Internal ID:            53
Name:                   docs-ldap
Type:                   LDAP
Domain:                 example.com
Resolution order:       0
Oper state:             Up
Addresses:              [192.0.2.33:389]
Transport mode:         STARTTLS
Schema:                 RFC2307
Search base:            dc=example,dc=com
```

To test several, give a comma-separated list. All identifiers in the list must be of the same type (all names, all UUIDs, or all internal IDs):

```
name-service-config --test-connect <name1>,<name2>
```

If every test fails, the command fails with one line per address and transport mode tried, for example `Failed to connect to LDAP server 'corp.example.com' using address 'ldap1.corp.example.com' with transport mode STARTTLS`.

Viewing Configured Name Services and Resolution Order

Using the GUI

Go to **Administration** › **Directory Services**. The table lists each configured directory service with its **Name**, **Domain**, **Type** (`LDAP` or `AD`), **Status** (`UP` or `DOWN` for an LDAP name service), and **ID Mapping** schema. To see the resolution order, click **Resolution Order**; the dialog lists the configured name services in the order they are queried. Click **Cancel** to close it without changes.

© 2026 - Hammerspace

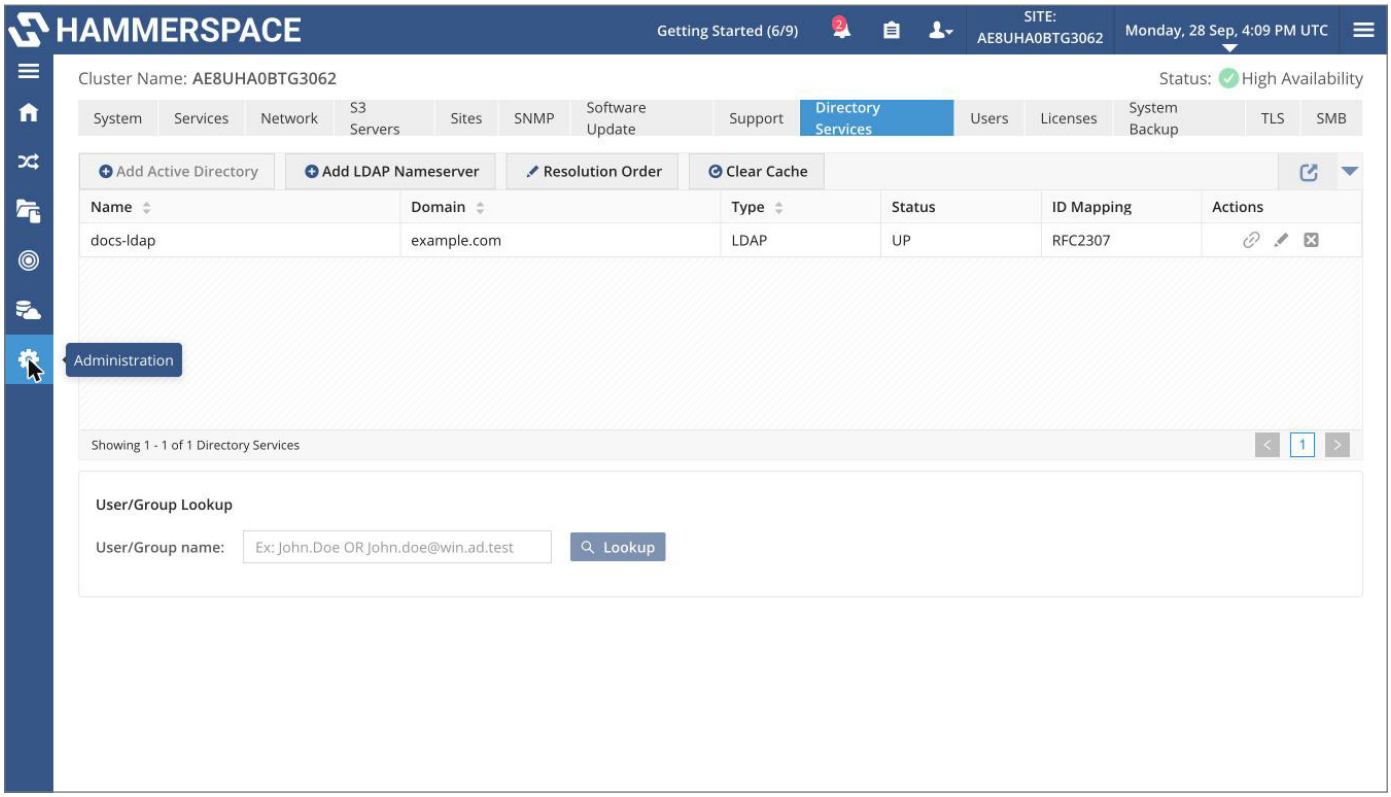


Figure 16. The Directory Services page with one LDAP name service

The **Actions** column has three icons for an LDAP name service: **Test Connection**, **Edit**, and **Delete**.

Using the Admin CLI

```
name-service-config --list
```

Expected output:

```
total 1
ID:                     1b22171b-aaa0-444e-a866-c30bc0ab26c3
Internal ID:            46
Name:                   docs-ldap
Type:                   LDAP
Domain:                 example.com
Resolution order:       0
Oper state:             Up
Addresses:              [192.0.2.33:389]
Transport mode:         STARTTLS
Schema:                 RFC2307
Search base:            dc=example,dc=com
```

Running `name-service-config` with no operation also lists the name services.

The list shows the name services in resolution order. Each entry shows **ID** (UUID), **Internal ID**, **Name**, **Type**,

© 2026 - Hammerspace

Domain, **Resolution order** (starting at `0` for the name service that is queried first), **Oper state** (`UP` or `DOWN`), **Addresses**, **Transport mode**, **Schema**, **Search base**, and **Bind DN**. The bind secret is never shown.

To show one name service, add `--name`, `--id`, or `--internal-id`:

```
name-service-config --list --name <name>
```

Changing the Directory Services Resolution Order

Hammerspace resolves unqualified user and group names against the LDAP name services in the configured order. By default, the order is the order in which the name services were added; when you remove a name service, the ones after it move up. You can change the order at any time. The change is distributed to the Anvils like any other configuration change.



See Best Practices for guidance on choosing an effective resolution order.

Using the GUI

1. Go to **Administration** › **Directory Services** and click **Resolution Order**. The **Update Resolution Order** dialog lists the configured directory services.
2. Select a name service and use the arrow buttons to move it up or down.
3. Click **Update**.

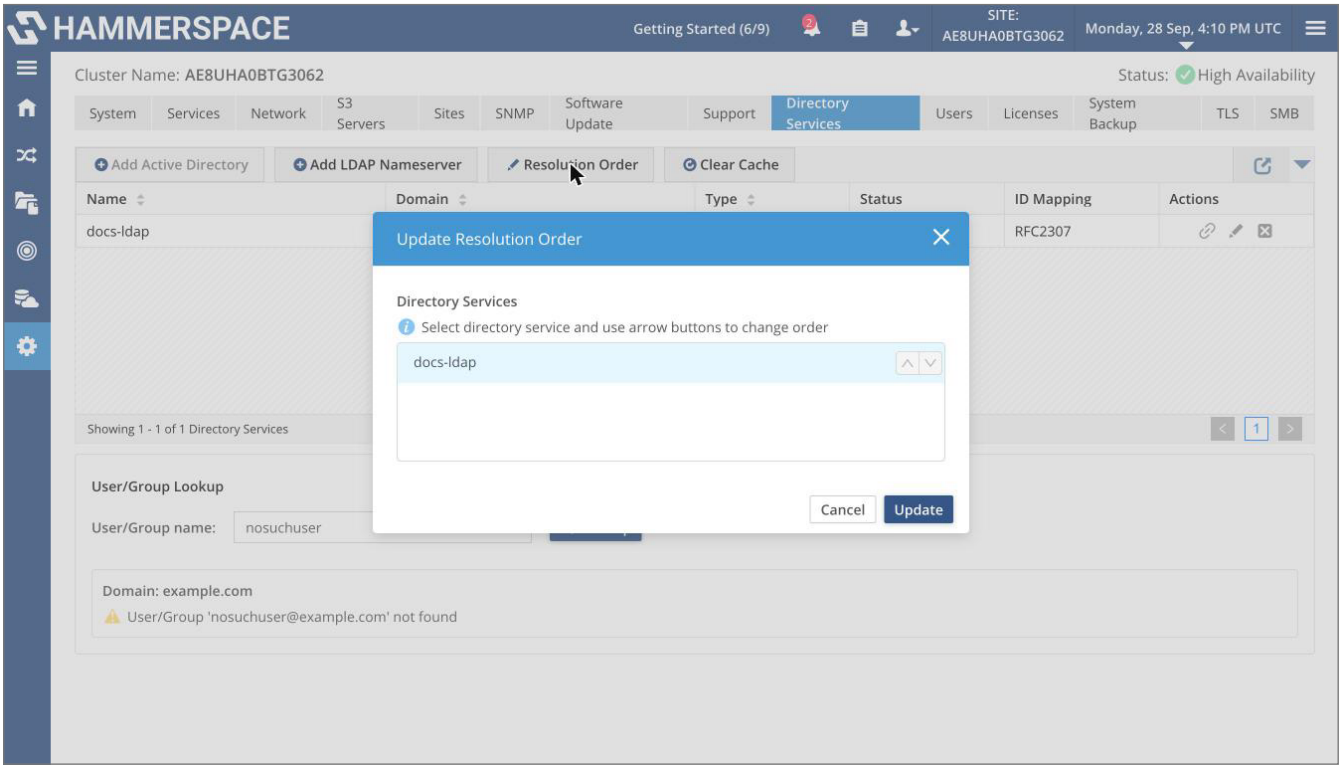


Figure 17. The Update Resolution Order dialog

The button is available only when at least one LDAP name service is configured.

© 2026 - Hammerspace

Using the Admin CLI

First, list the configured name services to get their names, UUIDs, or internal IDs:

```
name-service-config --list
```

Expected output:

```
total 2
ID:                     c01613ab-b84c-422b-bd04-c0ab15fb0ce9
Internal ID:            59
Name:                   docs-ldap
Type:                   LDAP
Domain:                 example.com
Resolution order:       0
Oper state:             Up
Addresses:              [192.0.2.33:389]
Transport mode:         STARTTLS
Schema:                 RFC2307
Search base:            dc=example,dc=com

ID:                     772eff01-f9c3-4c06-92a9-3ab0be347e97
Internal ID:            60
Name:                   docs-ldap-2
Type:                   LDAP
Domain:                 example.org
Resolution order:       1
Oper state:             Up
Addresses:              [192.0.2.33:389]
Transport mode:         STARTTLS
Schema:                 RFC2307
Search base:            dc=example,dc=org
```

Set the order by name (comma-separated, no spaces):

```
name-service-config --resolution-order <first-name>,<second-name>,<third-name>
```

Expected output (the list in its new order):

```
ID:                     772eff01-f9c3-4c06-92a9-3ab0be347e97
Internal ID:            60
Name:                   docs-ldap-2
Type:                   LDAP
Domain:                 example.org
Resolution order:       0
Oper state:             Up
Addresses:              [192.0.2.33:389]
```


© 2026 - Hammerspace

```
Transport mode:          STARTTLS
Schema:                  RFC2307
Search base:             dc=example,dc=org

ID:                      c01613ab-b84c-422b-bd04-c0ab15fb0ce9
Internal ID:             59
Name:                    docs-ldap
Type:                    LDAP
Domain:                  example.com
Resolution order:        1
Oper state:              Up
Addresses:               [192.0.2.33:389]
Transport mode:          STARTTLS
Schema:                  RFC2307
Search base:             dc=example,dc=com
```

Set the order by internal ID:

```
name-service-config --resolution-order 2,1,3
```

All identifiers in the list must be of the same type — names, UUIDs, or internal IDs; otherwise the command fails with `All domain identifiers must be of the same type.` Name services that you do not list are placed after the listed ones, in no guaranteed order, so list every name service when the order of all of them matters.

Updating an LDAP Name Service

You can change a name service's server address, transport mode, schema, search base, and bind credentials. In the Admin CLI, you can also change its management name.



The domain name cannot be changed after the name service is created. To use a different domain name, remove the name service and add it again.

Hammerspace reruns the connection test on an update only if the update changes the server addresses or ports, the transport mode, or the search base. A change to only the Bind DN, the Bind Secret, or the schema is saved without a connection test. After such a change, test the name service and look up a user to confirm that it works; see Testing the Connection to a Name Service.

Re-enter the Bind Secret with every update. In Hammerspace 5.3, if the name service uses a Bind DN, every update must include the Bind Secret again, even when you are not changing it. Otherwise the update fails with `If one of 'bindDn' or 'bindSecret' is specified, both must be specified.`

Using the GUI

1. Go to **Administration** › **Directory Services**.
2. In the **Actions** column of the LDAP name service, click the **Edit** icon (the second icon). The **Edit LDAP Name server** dialog opens. **Name** and **Domain Name** cannot be changed.

© 2026 - Hammerspace

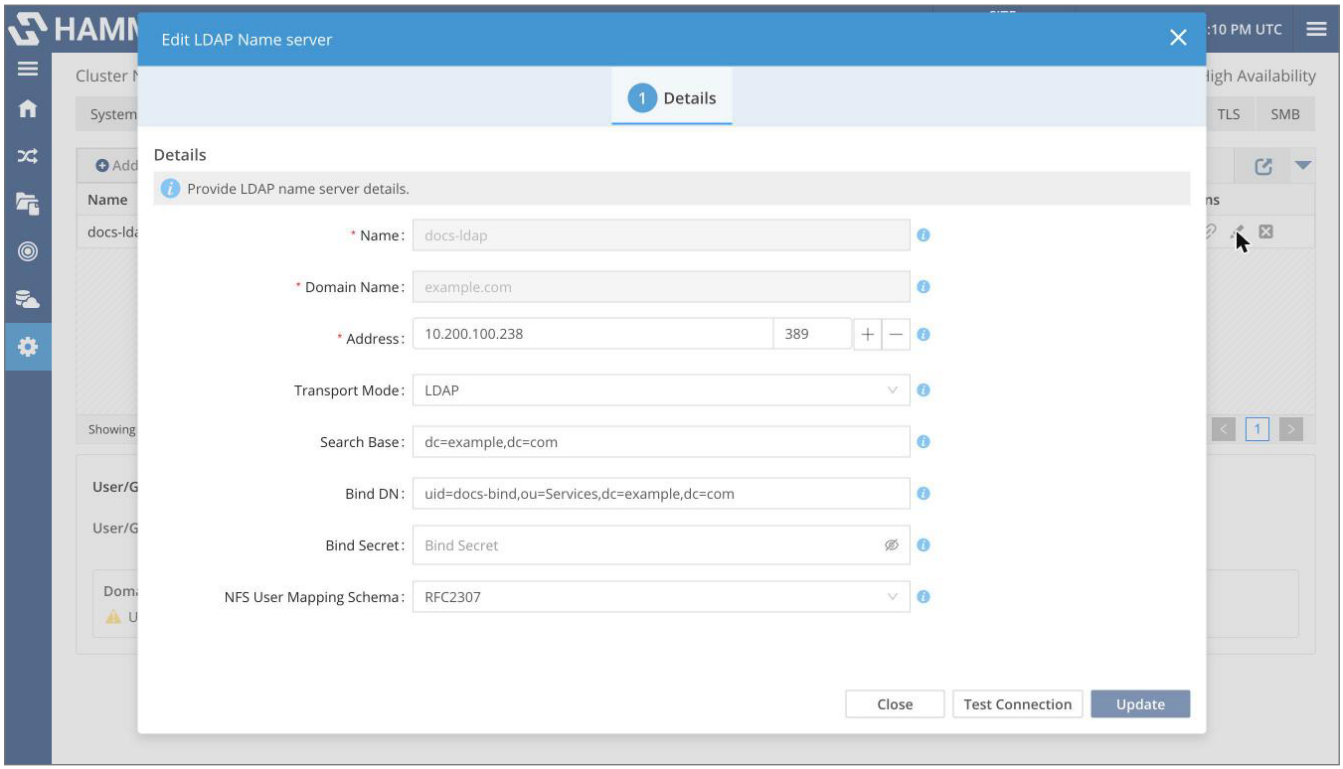


Figure 18. The Edit LDAP Name server dialog

3. Change the fields you need. If the name service has a **Bind DN**, enter the **Bind Secret** again; the field is always empty when the dialog opens. If you leave it empty, the update fails with `Failed to update <domain>. Details: If one of 'bindDn' or 'bindSecret' is specified, both must be specified`.

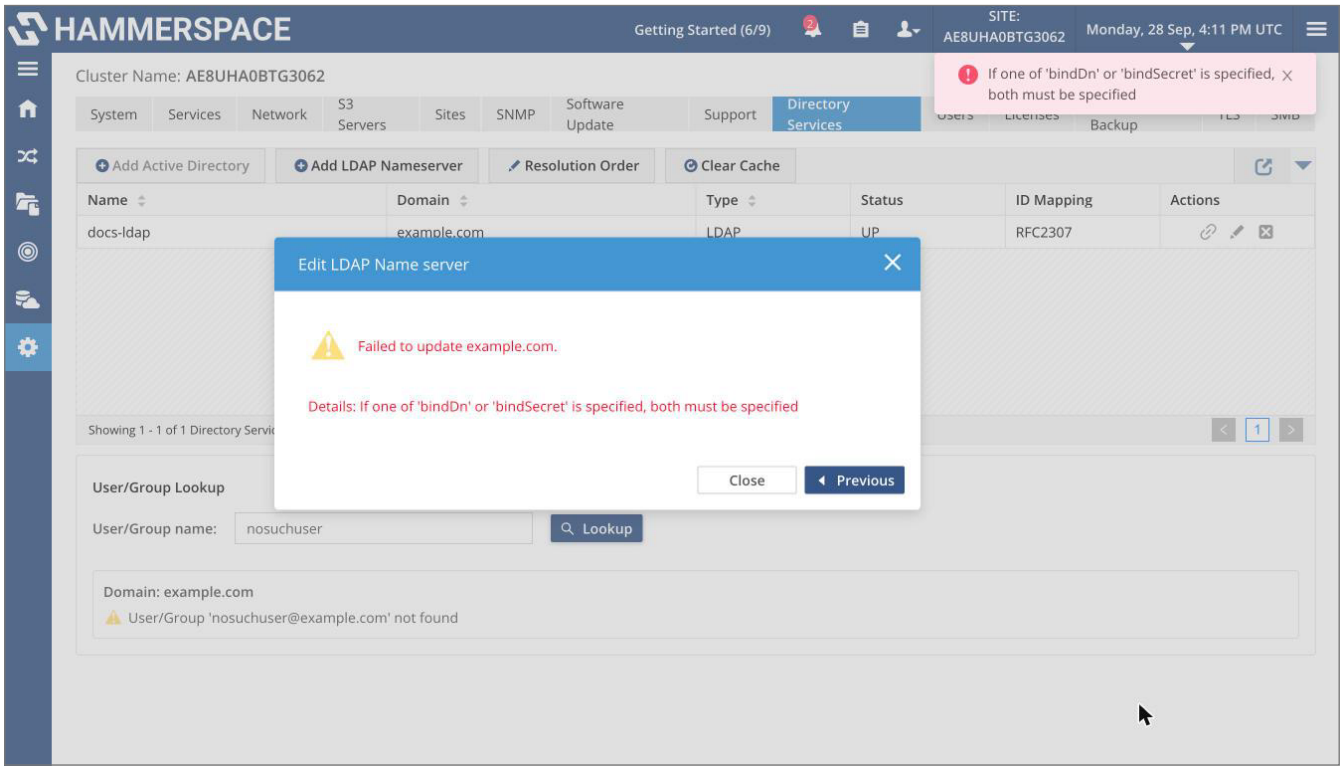


Figure 19. The update fails when the Bind Secret is not entered again

© 2026 - Hammerspace

4. Optionally, click **Test Connection**.
5. Click **Update**. The button is available once you have changed a field.

   The dialog shows `<domain> updated successfully.`, or `Failed to update <domain>.` and the reason.
6. Click **Close**.

Using the Admin CLI

Identify the name service with `--name`, `--id`, or `--internal-id`. Options you do not give keep their current values.

If the name service uses a Bind DN, add `--bind-secret <bind-secret>` to each of the following commands except the one that uses `--bind-clear`.

Replace the server address. The addresses you give replace all existing addresses:

```
name-service-config --update --name <name> --address <new-IP-or-hostname>
```

Change the bind credentials. To change only the secret, give `--bind-secret` alone; to change the Bind DN, give `--bind-dn` and `--bind-secret` together:

```
name-service-config --update --name <name> --bind-dn <bind-dn> --bind-secret <bind-secret>
```

Remove the bind credentials and use anonymous access:

```
name-service-config --update --name <name> --bind-clear
```

Clear the saved transport mode so that the connection test determines it again:

```
name-service-config --update --name <name> --transport-mode-reset
```

Reset the search base to the root of the domain (for example, `dc=corp,dc=example,dc=com`):

```
name-service-config --update --name <name> --search-base-reset
```

To set an empty search base instead, use `--search-base ""`.

Rename the name service (management name only):

```
name-service-config --update --name <old-name> --new-name <new-name>
```

`--transport-mode-reset` cannot be combined with `--no-connection-test`, because a name service saved

© 2026 - Hammerspace

without a test must have a transport mode.

Removing a Name Service

When you remove an LDAP name service, users in its domain can no longer be mapped to their UIDs and GIDs, so they lose access to shares that depend on that mapping. Hammerspace regenerates the SSSD configuration without the domain.

If you remove the last LDAP name service, Hammerspace also removes the SSSD configuration and stops SSSD. See How Hammerspace Manages SSSD on the Anvils.

Using the GUI

1. Go to **Administration** › **Directory Services**.
2. In the **Actions** column of the LDAP name service, click the **Delete** icon (the third icon).
3. In the **Delete Directory Service** dialog, confirm that you want to remove the name service for the domain shown.

Using the Admin CLI

Identify the name service with `--name`, `--id`, or `--internal-id`:

```
name-service-config --remove --name <name>
```

Expected output:

```
success
```

7.3.8. Admin CLI Reference: name-service-config

When called without an operation, `name-service-config` lists the configured name services, as with `--list`. Operations cannot be combined in one command.

Option	Used with	Required?	Notes
`--add`			Add a new LDAP name service. Requires `--name` and `--address`.
`--list`			List the configured name services in resolution order, or one name service when given `--name`, `--id`, or `--internal-id`.
`--update`			Update a configured name service. Requires `--name`, `--id`, or `--internal-id`.

© 2026 - Hammerspace

Option	Used with	Required?	Notes
`--remove`			Remove a configured name service. Requires `--name`, `--id`, or `--internal-id`.
`--name <name>`	`--add`, `--list`, `--update`, `--remove`	Required for `--add`	The management name of the name service. Maximum 255 characters; must be unique (case-insensitive).
`--id <uuid>`	`--list`, `--update`, `--remove`	Optional	UUID of the name service. Not allowed with `--add`.
`--internal-id <n>`	`--list`, `--update`, `--remove`	Optional	Integer ID of the name service. Not allowed with `--add`.
`--new-name <name>`	`--update`	Optional	Rename the name service (management name).
`--domain <domain>`	`--add`	Optional	The domain name, for example `corp.example.com`. If omitted, the `--name` value is used as the domain name and must be a valid domain name. Must be unique (case-insensitive); cannot be changed later.
`--address <host[:port]>`	`--add`, `--update`	Required for `--add`	IP address or DNS name of the LDAP server, with an optional port (1–65535). Repeat `--address` once per server; do not separate addresses with commas. On `--update`, the addresses given replace all existing addresses. If a port is omitted, the connection test determines it. More than one address works only with `--no-connection-test`.
`--transport-mode <mode>`	`--add`, `--update`	Optional	`LDAP`, `LDAPS`, or `STARTTLS` (not case-sensitive). If omitted, the connection test tries `STARTTLS`, then `LDAPS`, then `LDAP`, each on port 389 and then 636 when no port is given, and keeps the first that works. Required with `--no-connection-test`.

© 2026 - Hammerspace

Option	Used with	Required?	Notes
`--transport-mode-reset`	`--update`	Optional	Clear the transport mode so that the connection test determines it again. Cannot be used with `--no-connection-test`.
`--schema <schema>`	`--add`, `--update`	Optional	`RFC2307` (default) or `RFC2307BIS`.
`--search-base <base>`	`--add`, `--update`	Optional	Base DN for searches, for example `ou=people,dc=corp,dc=example,dc=com`. Defaults to the root of the domain. Pass `""` to set an empty search base.
`--search-base-reset`	`--update`	Optional	Reset the search base to the root of the domain (for example, `dc=corp,dc=example,dc=com`).
`--bind-dn <dn>`	`--add`, `--update`	Optional	Bind distinguished name for authenticated access. On `--add`, a plain account name is expanded to `cn=<name>,<domain in dc= form>`; on `--update`, give a full DN. Must be given with `--bind-secret`.
`--bind-secret <secret>`	`--add`, `--update`	Optional	Password for the Bind DN. Must be given with `--bind-dn` when adding a name service. On `--update`, give it alone to change only the secret. The Admin CLI does not prompt for it. In 5.3, also required on every `--update` of a name service that has a Bind DN, except with `--bind-clear`.
`--bind-clear`	`--update`	Optional	Remove the Bind DN and Bind Secret, so that the name service uses anonymous access.

© 2026 - Hammerspace

Option	Used with	Required?	Notes
`--no-connection-test`	`--add`, `--update`	Optional	Skip the connection test. Requires `--transport-mode` and a port on every `--address`.
`--resolution-order <list>`			Set the resolution order. Give a comma-separated list of names, UUIDs, or internal IDs, all of the same type. Name services not listed are placed after the listed ones, in no guaranteed order.
`--resolve-user <user>`			Look up a user through SSSD. An unqualified name (`jsmith`) is looked up in every configured domain, with a result for each; a qualified name (`jsmith@corp.example.com`) only in that domain, and an unconfigured domain returns an error.
`--resolve-group <group>`			Look up a group. Qualified and unqualified names behave as for `--resolve-user`.
`--resolve-user-groups <user>`			List the groups a user belongs to. Qualified and unqualified names behave as for `--resolve-user`.
`--test-connect <list>`			Run the connection test for one or more configured name services, using their saved settings. Give a name, UUID, or internal ID, or a comma-separated list of identifiers of the same type.

7.3.9. REST API Reference

Base URL: `https://<anvil-IP>:8443/mgmt/v1.2/rest/name-services/`

© 2026 - Hammerspace

Resource	Method	Description	Response Codes
`/name-services/`	GET	List all configured name services in resolution order.	200
`/name-services/<identifier>`	GET	Get a configured name service by UUID, internal ID, or name.	200, 404
`/name-services/`	POST	Add a new LDAP name service. The connection test runs unless the query parameter `skipConnectionTest=true` is given, in which case `ldapTransport` and a port on every network endpoint are required.	200, 400, 409
`/name-services/<identifier>`	PUT	Update a configured LDAP name service. Accepts `skipConnectionTest=true`, as for POST.	200, 400, 404, 409
`/name-services/<identifier>`	DELETE	Remove a configured name service.	204, 404
`/name-services/reorder`	PUT	Change the resolution order. The body is the list of name services in the new order.	200, 400
`/name-services/resolveUser?user-identifier=<user>`	GET	Look up a user. Returns one entry per domain searched, each with an `sssdQueryResult` whose `status` is `SUCCESS`, `NOT_FOUND`, or `FAILURE`.	200, 400
`/name-services/resolveGroup?group-identifier=<group>`	GET	Look up a group. Results as for `resolveUser`.	200, 400
`/name-services/resolveUserGroups?user-groups-identifier=<user>`	GET	List the groups a user belongs to. Results as for `resolveUser`.	200, 400
`/name-services/connectionTest`	POST	Test connectivity for a list of name services without saving them. They do not need to be configured. Returns 400 if every test fails; if only some fail, returns 200 with the header `X-Partial-Success: true` and the errors in each returned entry. In 5.3, include at most one name service that is not yet configured in each request.	200, 400

A lookup of a name that does not exist returns 200, with `NOT_FOUND` in the result for each domain. A qualified name whose domain is not configured, or a lookup when no name service is configured, returns 400.

When an add or update succeeds after some connection attempts failed, the response is 200 with `X-Partial-`

© 2026 - Hammerspace

`Success: true`, and the returned name service lists the failed attempts in `errors` and the successful one in `successMessage`.

Enumerated values in request bodies, such as `ldapTransport` and `ldapSchema`, are case-sensitive in 5.3. Use `LDAP`, `LDAPS`, or `STARTTLS`, and `RFC2307` or `RFC2307BIS`.

7.3.10. Troubleshooting

A user receives a permission-denied error on an NFSv4.2 mount.

The user's identity may not be resolving. Look up the user with **User/Group Lookup** or `--resolve-user` and check that the UID and GID are what you expect. Also check that the name service's status is `UP` in the Directory Services table. A status of `UP` shows only that the last connection test passed; the lookup is the real check.

User/Group Lookup returns "not found" for a known user.

Possible causes:

- The user is outside the configured search base. Check the **Search Base**.
- The user entry does not have `uidNumber` and `gidNumber`, or the user name is not in the `uid` attribute.
- The schema setting (RFC 2307 or RFC 2307bis) does not match the directory.
- The name service uses LDAPS or StartTLS and the server certificate is not trusted by the Anvils. The connection test and the `UP` status do not detect this, and users that were looked up before the problem started may still resolve from the SSSD cache. See Securing LDAP Connections with LDAPS or StartTLS.
- The LDAP server is unreachable from the Anvils. Test the connection; see Testing the Connection to a Name Service.
- The change was made in the directory recently and SSSD still has the old entry cached. Wait for the cached entry to expire. The **Clear Cache** button on the Directory Services page clears the Active Directory cache only, not the LDAP cache.

A qualified lookup fails with "Domain '<domain>' specified by user name '<name>' could not be found".

The domain in the qualified name (`user@domain`) does not match the domain name of any configured name service. Check the domain names in the Directory Services table. Domain names are matched without regard to case.

Adding a name service fails with "Failed to connect to LDAP server".

Every combination of address and transport mode that was tried failed. The message has one line per combination, naming the domain, the address, and the transport mode (or "any transport mode" and "and with all common LDAP ports" when those were probed); it does not give the reason. Check the following causes, testing from an NFS client or another host on the Anvils' network with standard LDAP tools where you can:

- The server is unreachable on the port, or the transport mode is not enabled on the server.
- No naming context on the server matches the domain name. The domain `corp.example.com` requires the naming context `dc=corp,dc=example,dc=com` in the server's root DSE, even if you set a narrower search

base. The GUI's **Domain Name** tooltip example (`ldap.test.com`) does not change this rule.

- The search base does not exist, or the Bind DN cannot read it.
- The Bind DN or Bind Secret is wrong.
- For LDAPS or StartTLS, the certificate does not list the configured address in its Subject Alternative Name. See Securing LDAP Connections with LDAPS or StartTLS.

When some attempts fail but one succeeds, the name service is added, and the failed attempts are listed with their reasons (in the GUI, click **Show Details**; in the Admin CLI, under `Connection attempts:`).

Adding or updating a name service with more than one address fails.

A name service with more than one address can be saved only when the connection test is skipped. In the Admin CLI the failure is reported only as `name-service-config: unexpected error`. See Limitations.

Updating a name service fails with "If one of 'bindDn' or 'bindSecret' is specified, both must be specified".

The name service uses a Bind DN, and the update did not include the Bind Secret. In 5.3, include the Bind Secret with every update of such a name service (in the GUI, enter it again in **Bind Secret**; in the Admin CLI, add `--bind-secret`), unless the update removes the credentials with `--bind-clear`. Giving `--bind-dn` without `--bind-secret` fails with the same message.

Test Connection shows the port with a comma, for example `10.0.0.15:1,389`.

The GUI formats port numbers of 1000 and above with a thousands separator. The port in use is 1389; no action is needed.

A name service shows status `DOWN`.

The last connection test to the name service failed. Check:

- Network connectivity from the Anvils to the LDAP server.
- That the LDAP service is running on the server.
- That the saved address, port, and transport mode are still correct.

Hammerspace retests a `DOWN` name service every 10 minutes and changes it back to `UP` when the test passes. It tests `UP` name services only once every 24 hours, so a failure can take up to a day to show as `DOWN`. To test immediately, see Testing the Connection to a Name Service.

A `NAME_SERVICE_UNHEALTHY` event appears.

The event reads `Name Service '<name>' (domain '<domain>') is unhealthy` and corresponds to a name service with status `DOWN`; see the previous entry. It clears automatically when the name service passes a connection test.

An `SSSD_CONFIG_SYNC_FAILED` event appears.

The event reads `Failed to synchronize SSSD configuration to node '<Anvil>'`. Hammerspace could not

 © 2026 - Hammerspace

apply the SSSD configuration on that Anvil, or could not start SSSD with it. Check that the Anvil is online and reachable from the other Anvil. Hammerspace retries every 2 minutes and clears the event when it succeeds. While the event is active, LDAP lookups can fail; see Limitations.

`/etc/sssd/sssd.conf` is missing on an Anvil.

This is normal when no LDAP name service is configured and Active Directory is not joined with an ID-mapping schema. See How Hammerspace Manages SSSD on the Anvils.

Joining Active Directory fails with `ad-join: Cannot join Active Directory while an LDAP Name Service is configured.`

LDAP name services and Active Directory are mutually exclusive in Hammerspace 5.3. Remove all LDAP name services before joining Active Directory. In the GUI and the API, the same message appears without the `ad-join:` prefix.

Adding a name service fails with "Cannot configure an LDAP Name Service when the system is joined to an Active Directory domain".

The cluster is joined to Active Directory. In 5.3, leave Active Directory before adding an LDAP name service.

Error messages appear when adding a name service with the Admin CLI.

When the connection test tries transport modes and ports to find one that works, it reports each combination that fails. This is expected as long as the output ends with a successful attempt. If no attempt succeeds, the name service is not added; check the address, port, and transport mode.

7.4. 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*.

7.4.1. 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 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
```


© 2026 - Hammerspace



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.

7.4.2. 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>
```

7.4.3. Supported NFS Protocols

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.

© 2026 - Hammerspace

NFS version	Status
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>`

7.4.4. 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).



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

© 2026 - Hammerspace

> 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.

7.4.5. 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"
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

© 2026 - Hammerspace

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.

7.5. 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.

7.5.1. Multi-Protocol File Locking

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

7.6. 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.

© 2026 - Hammerspace

7.6.1. 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.

7.6.2. 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.
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

© 2026 - Hammerspace

Term	Definition
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.

7.6.3. 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 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.

7.6.4. 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.

© 2026 - Hammerspace

- **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.

7.6.5. 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.

7.6.6. 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.

7.6.7. 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.

7.6.8. 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.

© 2026 - Hammerspace

5. Set up buckets.
6. Configure the S3 client to connect to the S3 server.

7.6.9. 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.

7.6.10. 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

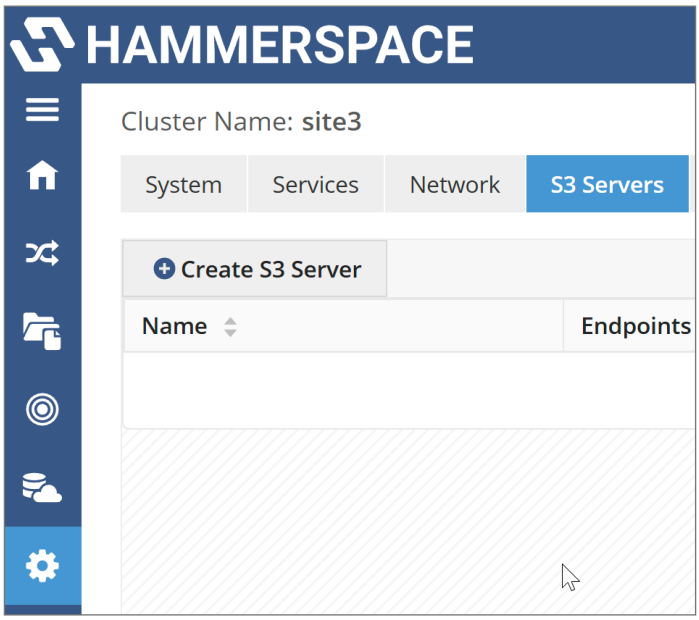


Figure 20. Click the Create S3 Server button

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

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

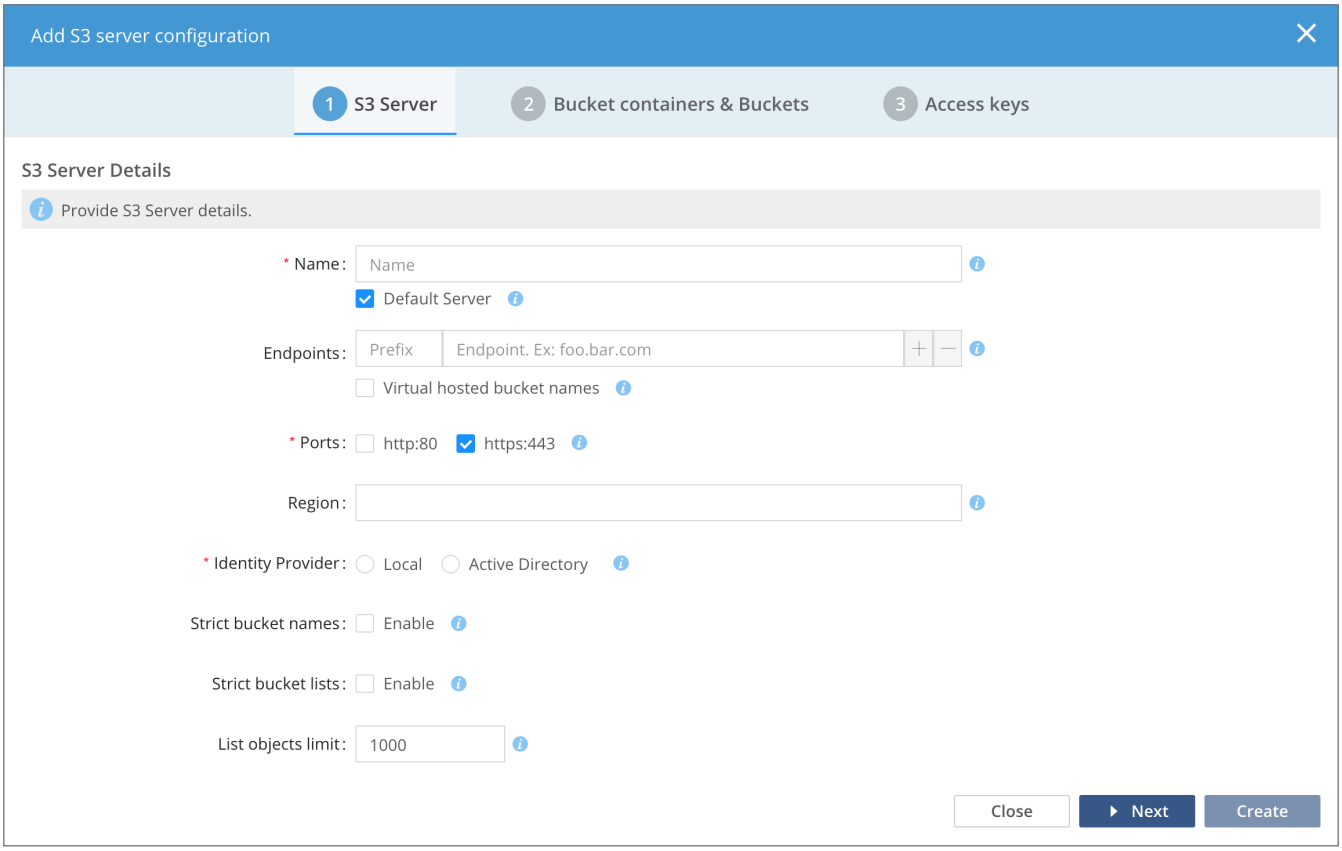


Figure 21. Add server information

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.

7.6.11. 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

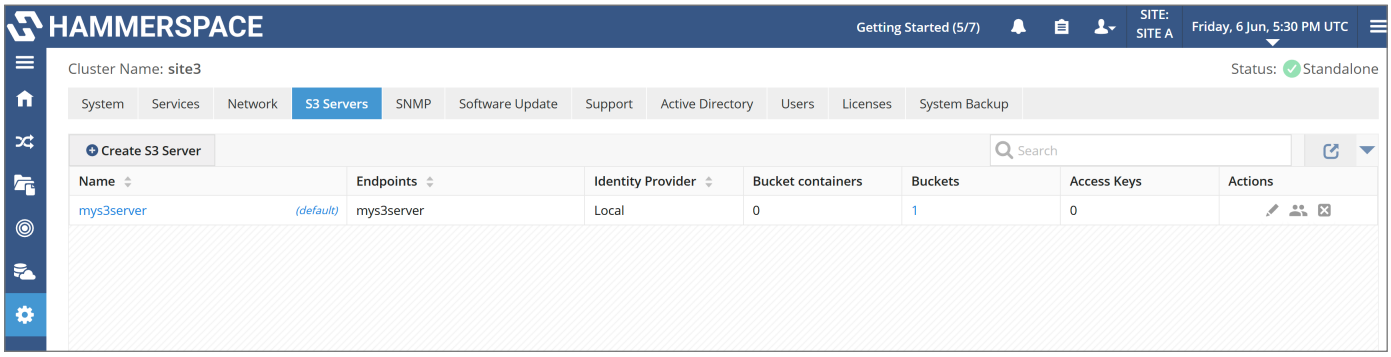


Figure 22. S3 server view

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.

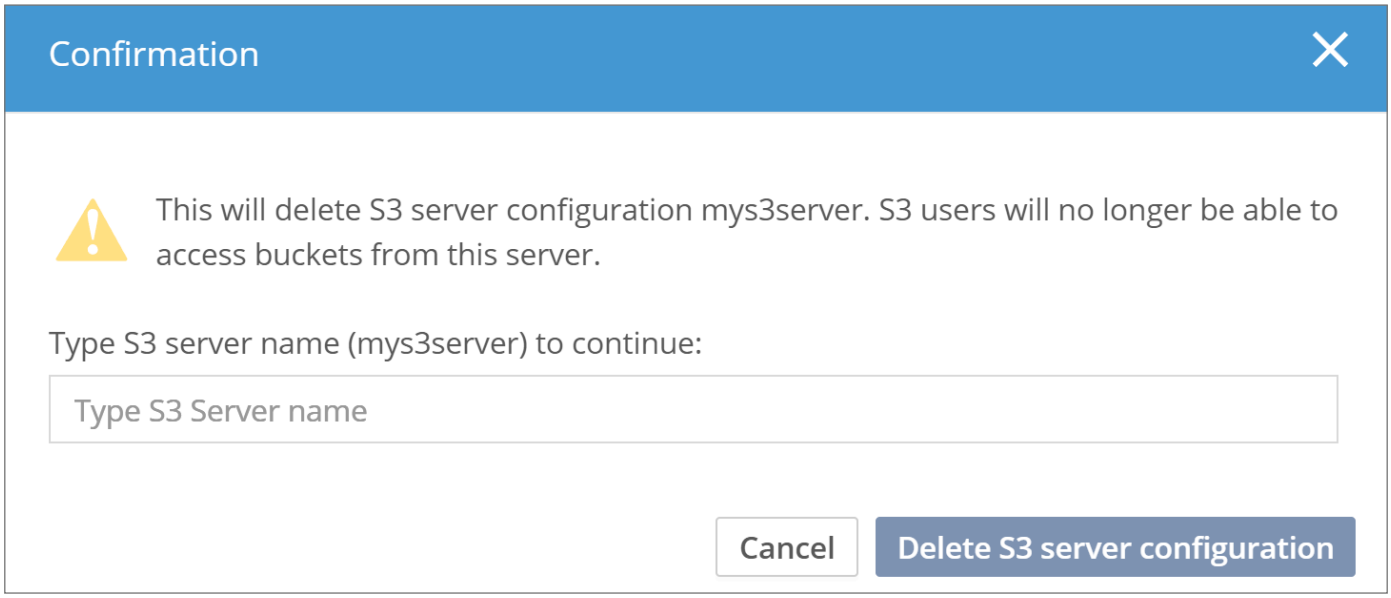


Figure 23. Confirmation of S3 server deletion

7.6.12. 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.

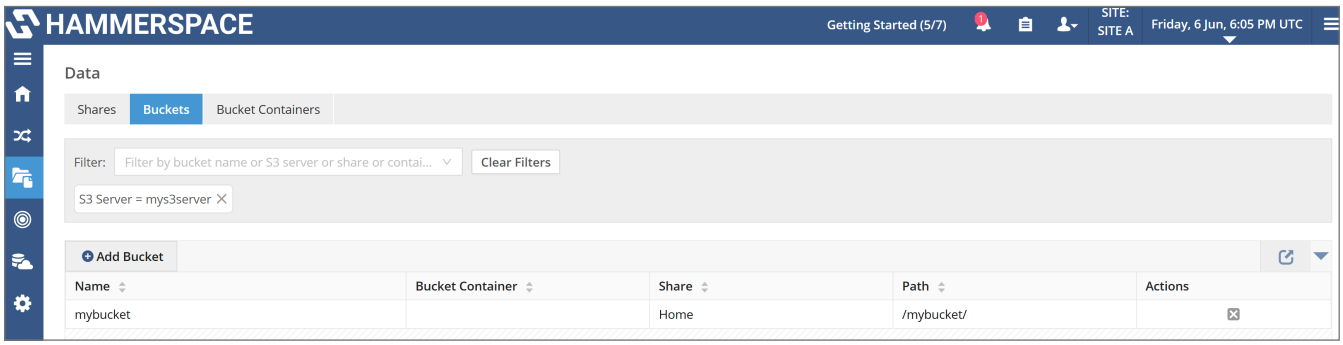


Figure 24. Buckets list

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.

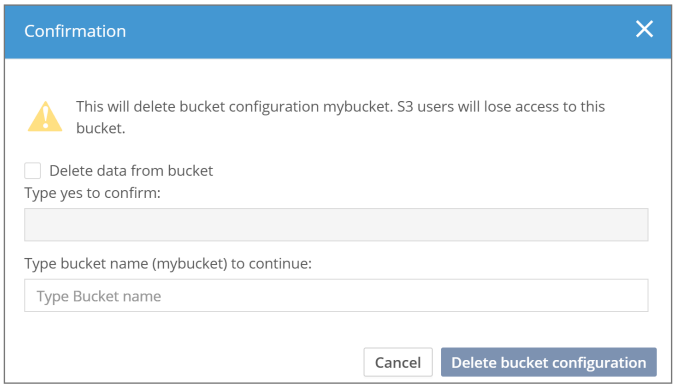


Figure 25. S3 bucket-delete confirmation

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.

7.6.13. 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

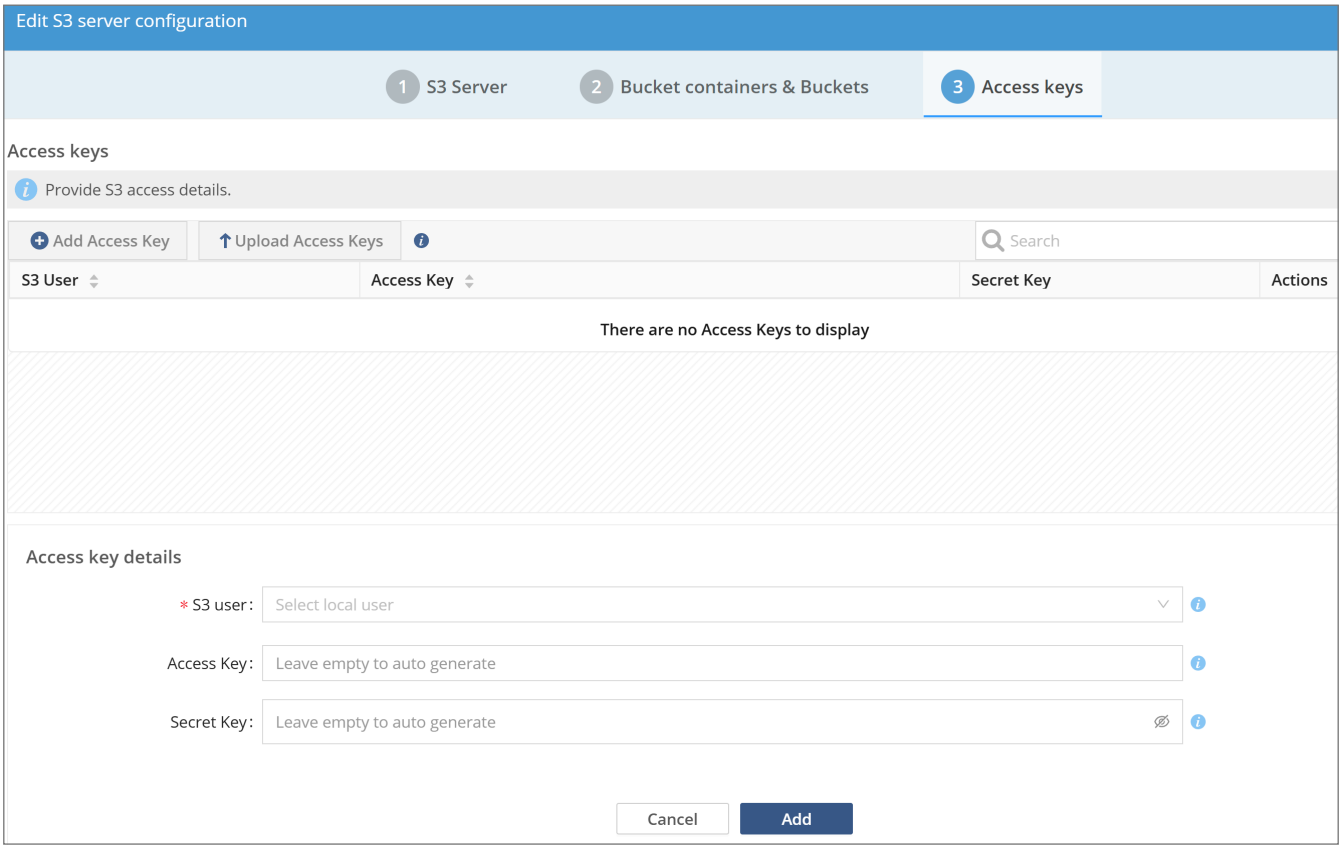


Figure 26. Example of adding an access key for a local user

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

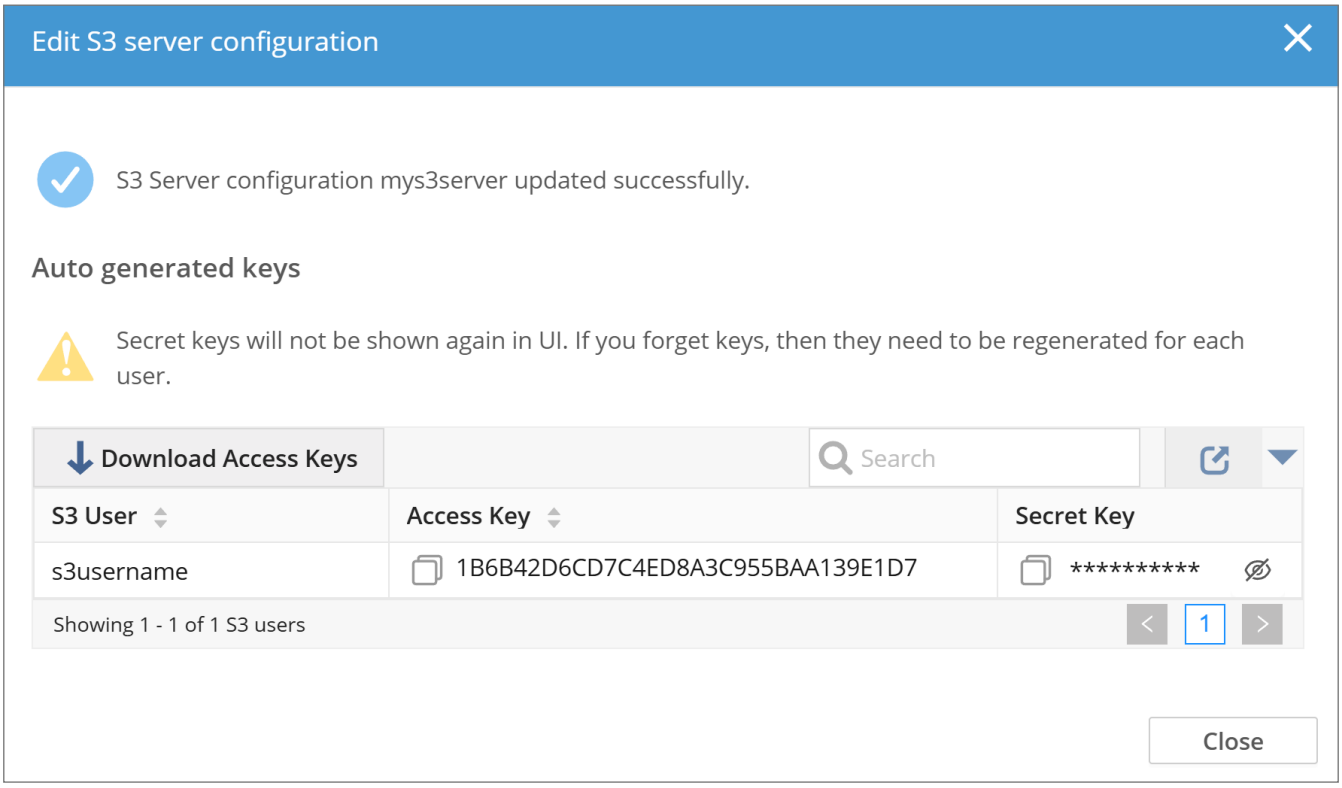


Figure 27. Adding an access key to an S3 server

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
```

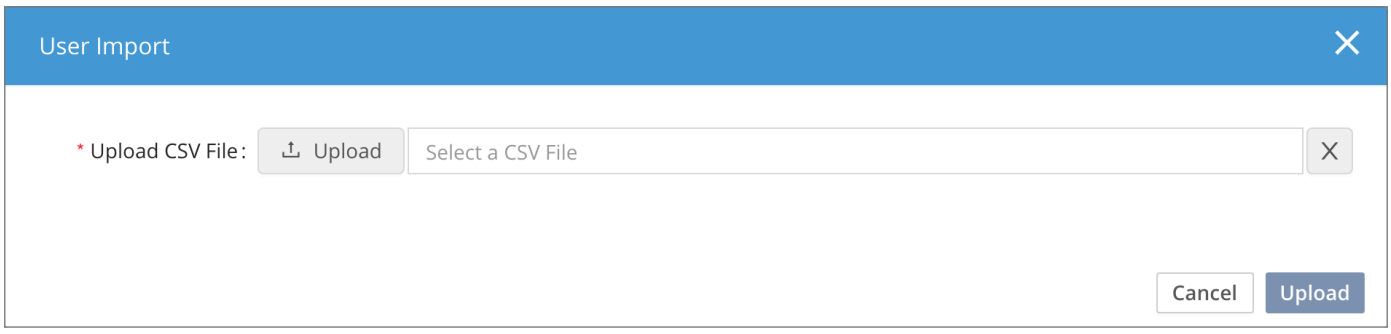


Figure 28. Uploading a CSV for S3 users

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.

7.6.14. 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	
GetBucketTagging / PutBucketTagging / DeleteBucketTagging	Added in 5.3.
GetBucketOwnershipControls / PutBucketOwnershipControls / DeleteBucketOwnershipControls	
GetBucketPolicyStatus	
GetBucketVersioning	
PutBucketVersioning	Cannot be used to disable snapshots or versioning.
GetBucketIntelligentTieringConfiguration	Always returns `STANDARD`.
ListBucketIntelligentTieringConfiguration	Always returns `STANDARD`.
DeleteBucketIntelligentTieringConfiguration	Always returns `200 OK`.
GetBucketPolicy / DeleteBucketPolicy	
GetPublicAccessBlock / PutPublicAccessBlock / DeletePublicAccessBlock	
GetBucketLogging	

© 2026 - Hammerspace

Operation	Notes
GetBucketAccelerateConfiguration	
GetBucketLifecycleConfiguration / PutBucketLifecycleConfiguration / DeleteBucketLifecycle	Used in conjunction with Hammerspace objectives. `PutBucketLifecycleConfiguration` supports `NoncurrentDays` only.

Table 8. Object operations

Operation	Notes
GetObject / PutObject / CopyObject	
PostObject	Added in 5.3.
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 `/`

© 2026 - Hammerspace

> behave differently than on AWS. See S3 Object Keys and the File System Namespace.

7.6.15. 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 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.

For what happens when a bucket's directory is removed over NFS or SMB, see the Missing Bucket and Bucket Container Events section of this page.

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

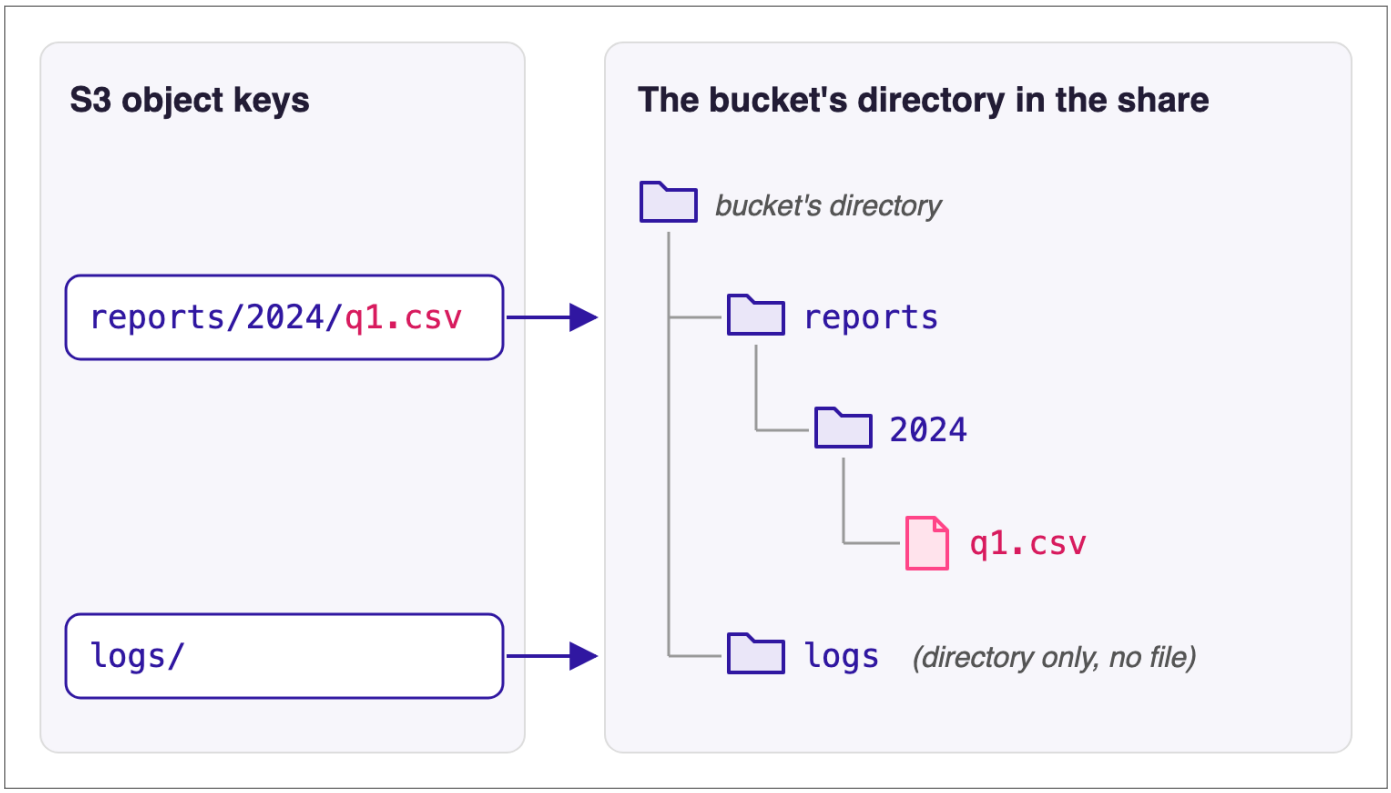


Figure 29. How the keys `reports/2024/q1.csv` and `logs/` are stored in the bucket's directory

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-` (the default for the AWS SDK for Java): `409 Conflict`. The error code is `ConditionalRequestConflict` when the existing object is the key's immediate parent, and `KeyAlreadyExists` when it is higher up the path.

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>
```

Because the error code depends on the upload type, an application should detect a conflict from the HTTP

status `409`, and should not rely on a single error code. When a `409` response to PutObject is caused by a naming conflict, retrying the same request does not succeed until the conflicting object or directory is removed.

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>
```

ListObjectVersions reports an empty directory the same way: as a key in a listing without a delimiter, and as a common prefix in a listing with delimiter `/`.

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

© 2026 - Hammerspace

these responses:

Request:

```
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.

© 2026 - Hammerspace

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

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.
- 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.

7.7. 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.

7.7.1. 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:

> In Hammerspace 5.3, joining Active Directory for Kerberos requires two DNS names — an

© 2026 - Hammerspace

> *Anvil DNS name* and a *Portal DNS name* — each backed by a DNS record that resolves to the correct address. The **Anvil DNS name** is used by NFS v4.2 clients that mount with Kerberos security (`sec=krb5`). If the Anvil DNS name cannot be resolved, NFS v4.2 Kerberos mounts hang and eventually time out or fail. Before mounting with Kerberos, confirm that both names resolve correctly (forward and reverse) from your clients and domain controllers.

The two names exist because Kerberos identifies a service by name. A client connecting to a data portal and a client connecting to the Anvil are talking to different services, so each needs its own DNS entry and its own service principal name.

Name	Resolves to	Used by
Anvil DNS name	The cluster floating IP address on the active Anvil node	NFS v4.2, and other Kerberos-sensitive operations. This name is new in 5.3.
Portal DNS name	Portal floating IP addresses, or static DSX data IP addresses, depending on how the deployment is designed	NFSv3, NFSv4.1, and SMB. This is the cluster's original name, which earlier releases called the SMB server name or cluster NetBIOS name, and it is the name of the cluster's computer account in Active Directory.

By default, the Anvil DNS name is the Portal DNS name with `-anvil` appended. On a cluster named `filesrv01` in domain `example.com`, the defaults are `filesrv01.example.com` for the portal and `filesrv01-anvil.example.com` for the Anvil.

Where more than one address backs the Portal DNS name, clients are distributed across those addresses by round-robin DNS. Create PTR records for the relevant addresses as well, so that reverse lookups resolve.



> In Hammerspace 5.3, a cluster cannot join Active Directory while an LDAP name service is configured, and **Add Active Directory** is unavailable while any directory service is listed. Remove all LDAP name services first. See Configuring an LDAP Name Service.

1. Go to **Administration** › **Directory Services** and click **Add Active Directory**.
2. On the **Details** tab:
   a. Enter the **Domain Name** as a fully qualified domain name (an IP address can also be used).
   b. Enter the **Portal DNS Name (SMB | NFS 3 | NFS 4.1)**. This name is used to register the cluster's computer account in Active Directory, and SMB clients use it to reach the cluster at **\\\<*portal DNS name.domain name*>**. To change it, leave and rejoin Active Directory with the new name.
   c. Check the **Anvil DNS Name (NFS 4.2)**. It is filled in as the Portal DNS name with `-anvil` appended; change it if your DNS design requires a different name. The two names must be different.
   d. Enter the **Login Name** and **Password** of an Active Directory account with the privilege to *add workstations* to the domain. Ongoing DNS registration is then handled by the computer account rather than by these credentials; for the exact permissions involved, see Active Directory Permissions.

© 2026 - Hammerspace

e. *Optionally*, enter an **Organizational Unit**. The account entered above must have permission to write into the Organizational Unit location.

f. For **AD Servers**, select **Auto** to use discovered servers, or **Manual** to enter them yourself. Auto works well when AD Sites and Services are set up correctly.

g. Under **SMB ↔ NFS User Mapping**, select the schema used to map Active Directory Security Identifiers (SIDs) to User IDs (UID) and Group IDs (GID): **Disabled** (no cross-protocol mapping), **RFC2307**, or **RFC2307BIS**. Refer to the **User Mapping** section for more details.

3. On the **DNS Registration** tab, choose the IP addresses to register in DNS for the two names:
   - Select **Register and continually manage Floating IPs** (recommended) to have Hammerspace register the floating IPs and keep the DNS host records up to date as floating IPs are added, removed, or go offline. This option is not selected by default.
   - *Optionally*, select static IP addresses. Registering static IPs is not recommended.
4. Review the **Kerberos** tab. Hammerspace registers the NFS service principal names (SPNs) automatically when it joins the domain; see Registering NFS Principal Names.
5. Click **Add**.

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 **Administration › Directory Services**.
2. In the row for the Active Directory domain, click the **Delete** action. The **Leave Active Directory** dialog opens.
3. *Optionally*, select **Remove computer accounts** to remove the Anvil and Portal computer accounts from Active Directory. To remove them, enter the **Login Name** and **Password** of an Active Directory account with the privilege to remove computers from the domain. The password is required only for removing computer accounts.
4. Click **Leave**.



If you leave without entering the password used to join the domain, the NFS service principal names (SPNs) registered for the Anvil DNS name remain in Active Directory. Ask your Active Directory administrator to remove them.

7.7.2. 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

© 2026 - Hammerspace

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

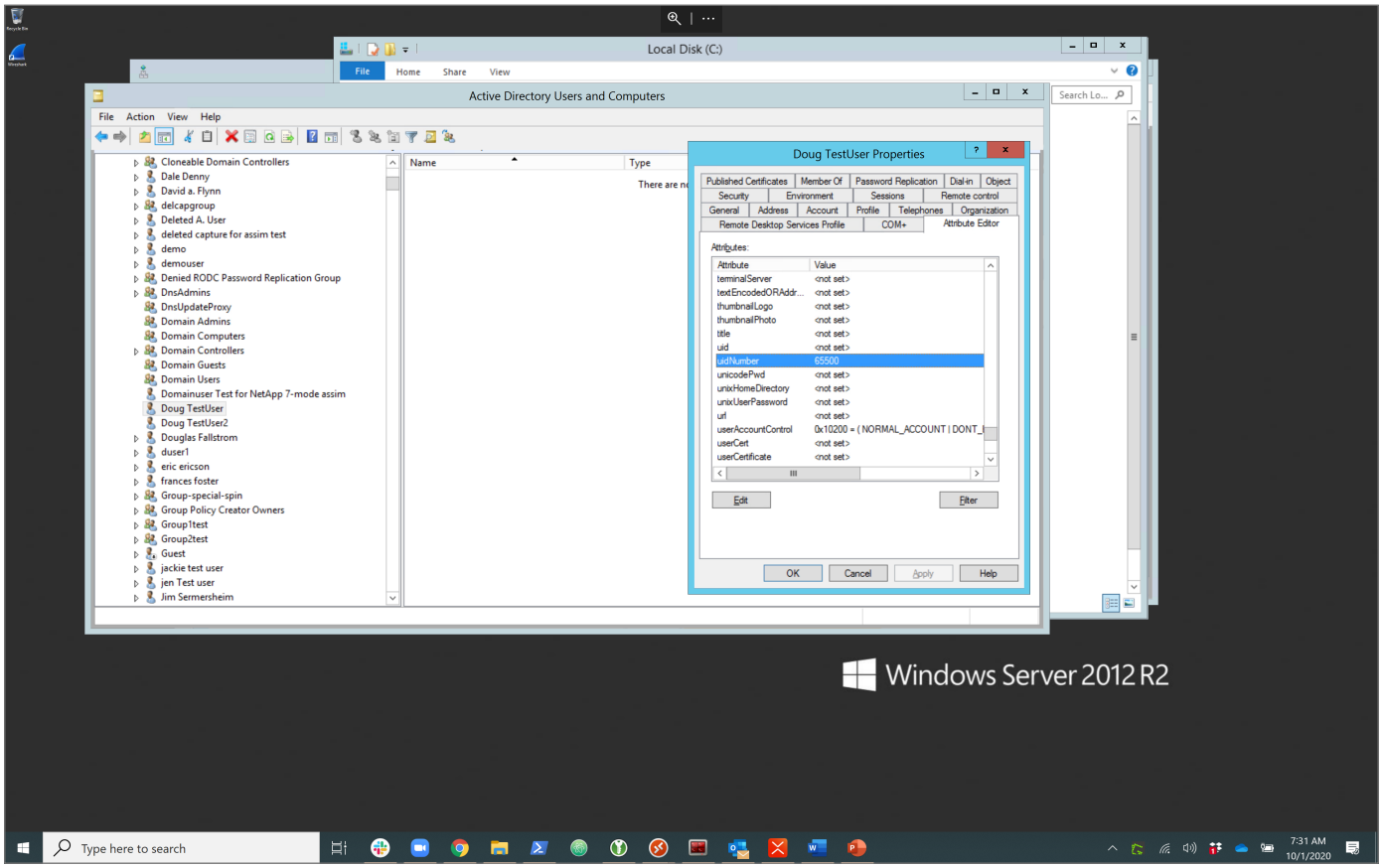


Figure 30. The uidNumber field in Active Directory

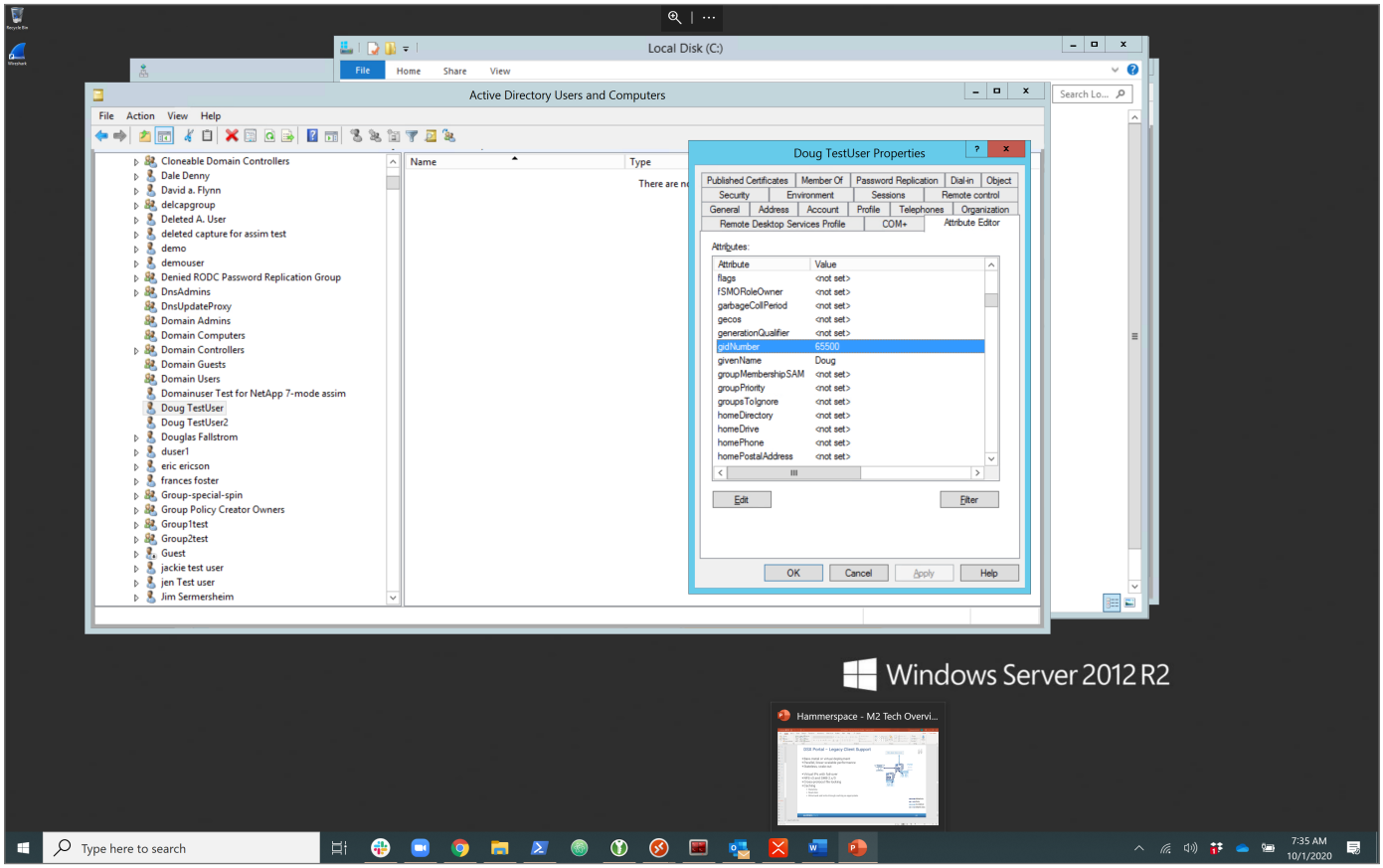


Figure 31. The gidNumber field in Active Directory

© 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
```

7.7.3. 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

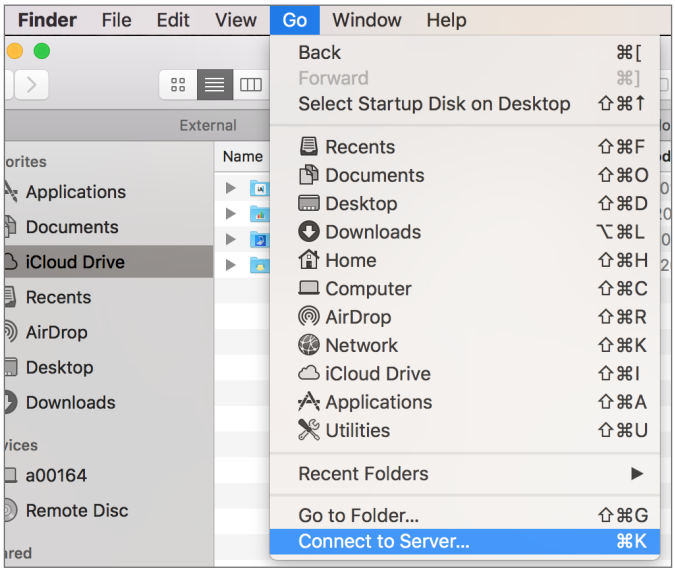


Figure 32. Location of Connect to Server control

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

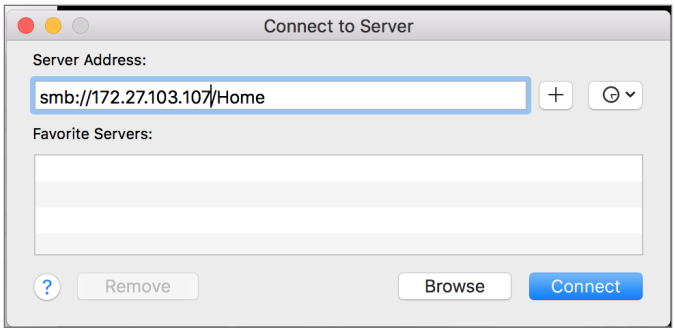


Figure 33. Enter server address to connect to server

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

© 2026 - Hammerspace

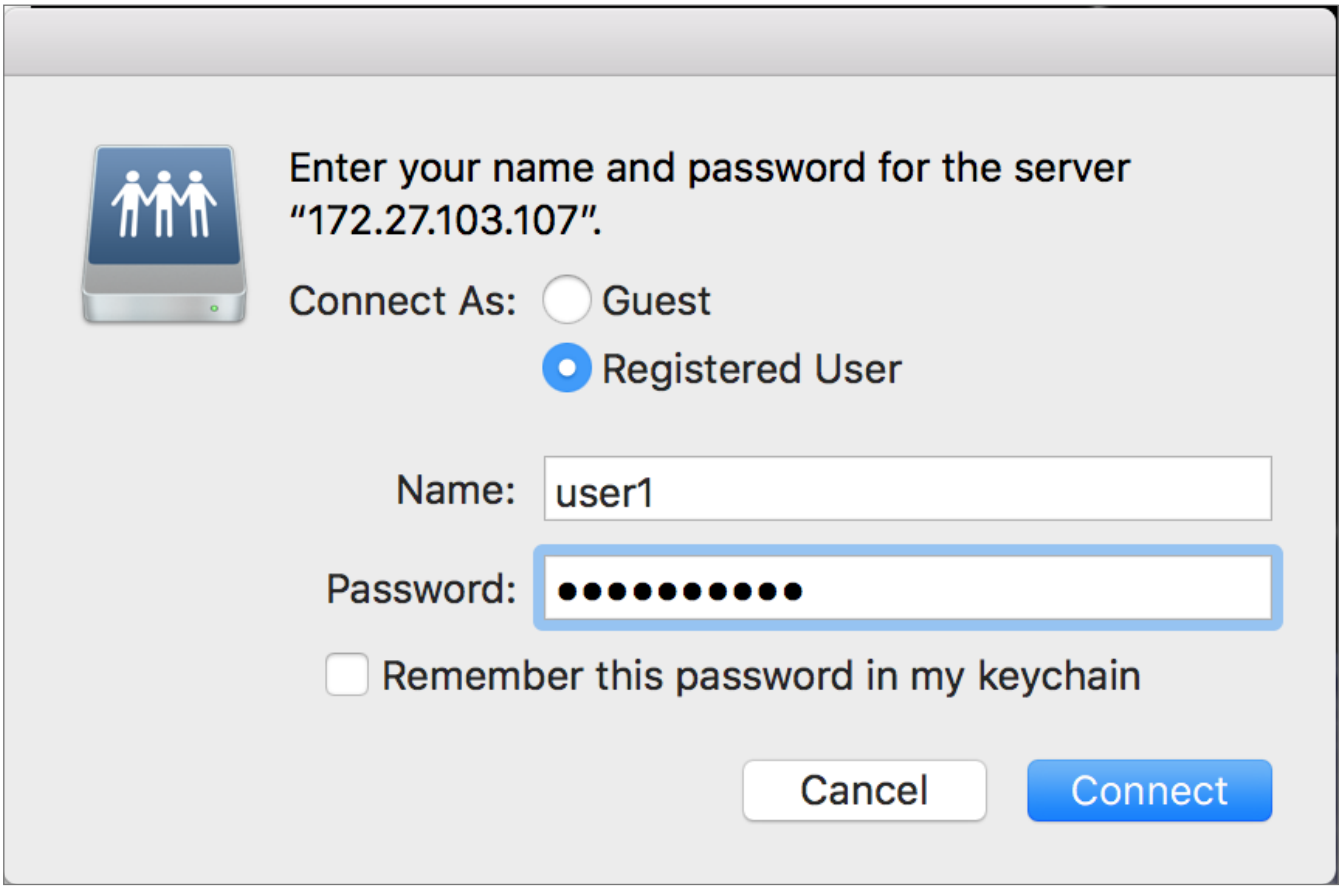


Figure 34. Enter Active Directory credentials

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.

7.7.4. 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.

7.7.5. Mapping a Share

You can map shares using the Microsoft Management Console.

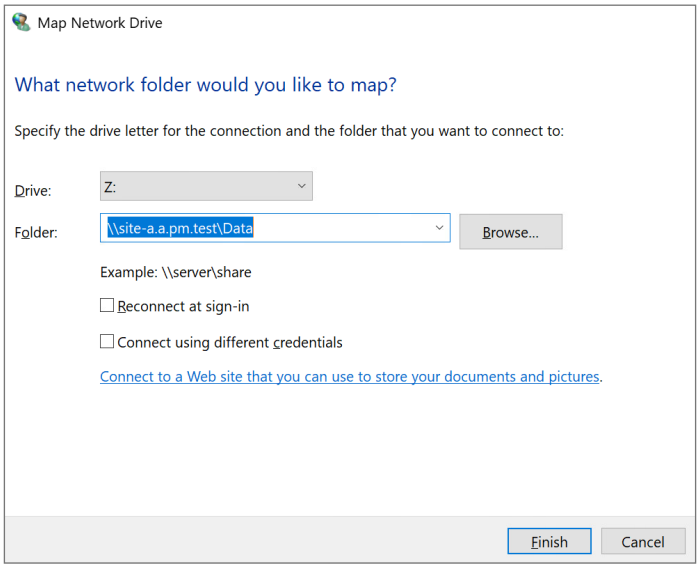


Figure 35. Mapping a network drive

The share is now mapped as a drive letter.

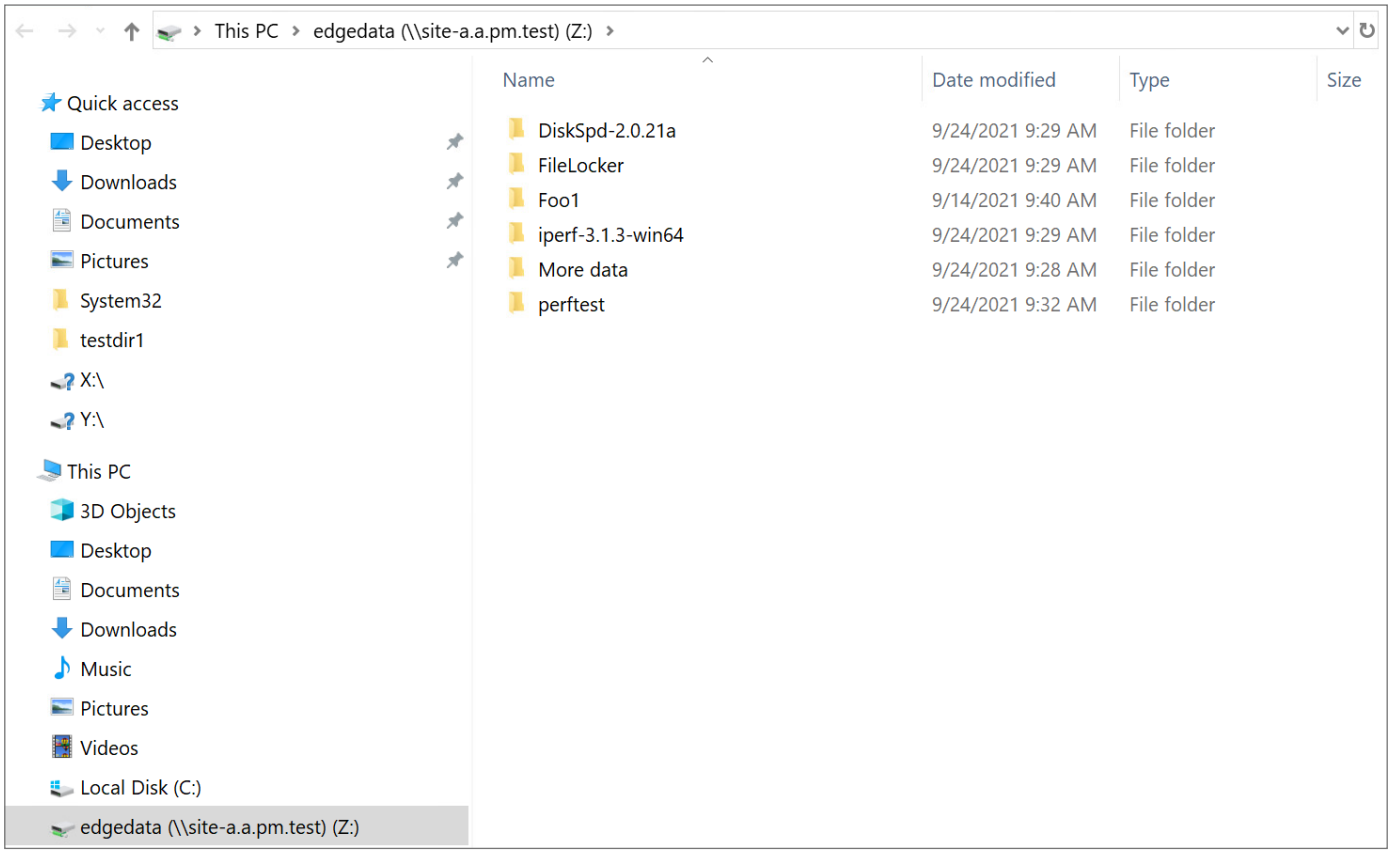


Figure 36. Share is mapped as a drive letter

7.7.6. Mapping a Drive Letter Using the Windows CMD

The following example shows the command and output:

Mapping a drive letter using Windows CLI

© 2026 - Hammerspace

Command:

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

Expected outcome

```
The command completed successfully.
```

7.7.7. 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

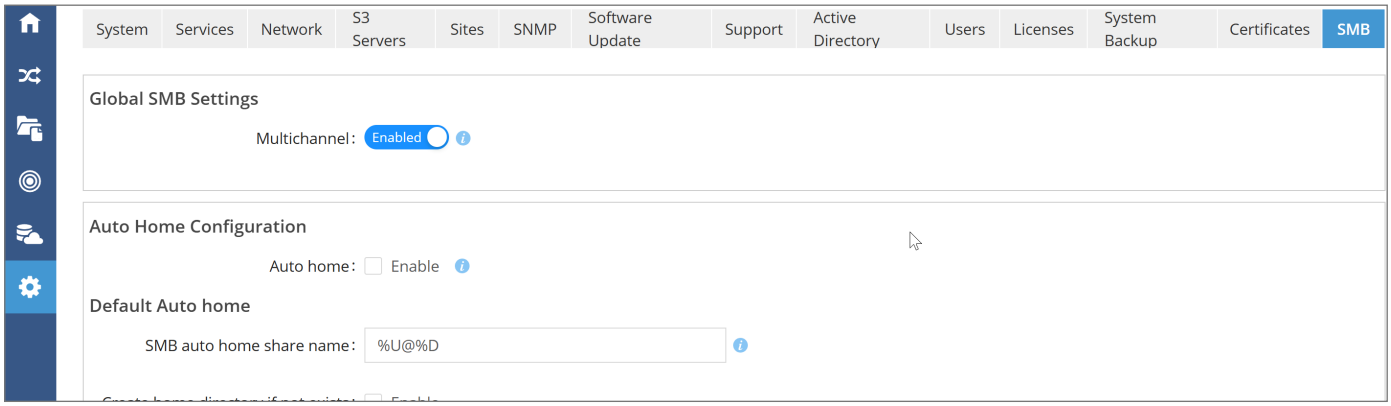


Figure 37. SMB Multi-Channel setting in the Management GUI

7.7.8. 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.
```

7.7.9. 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

7.7.10. 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.

7.7.11. 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*.

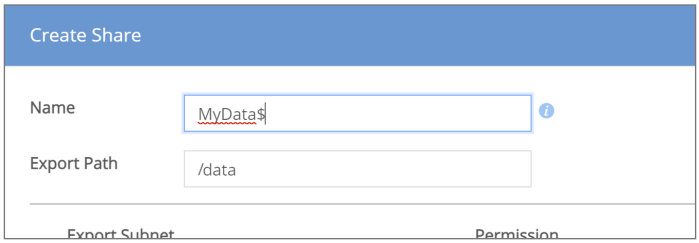


Figure 38. Creating a hidden share

7.7.12. 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**.

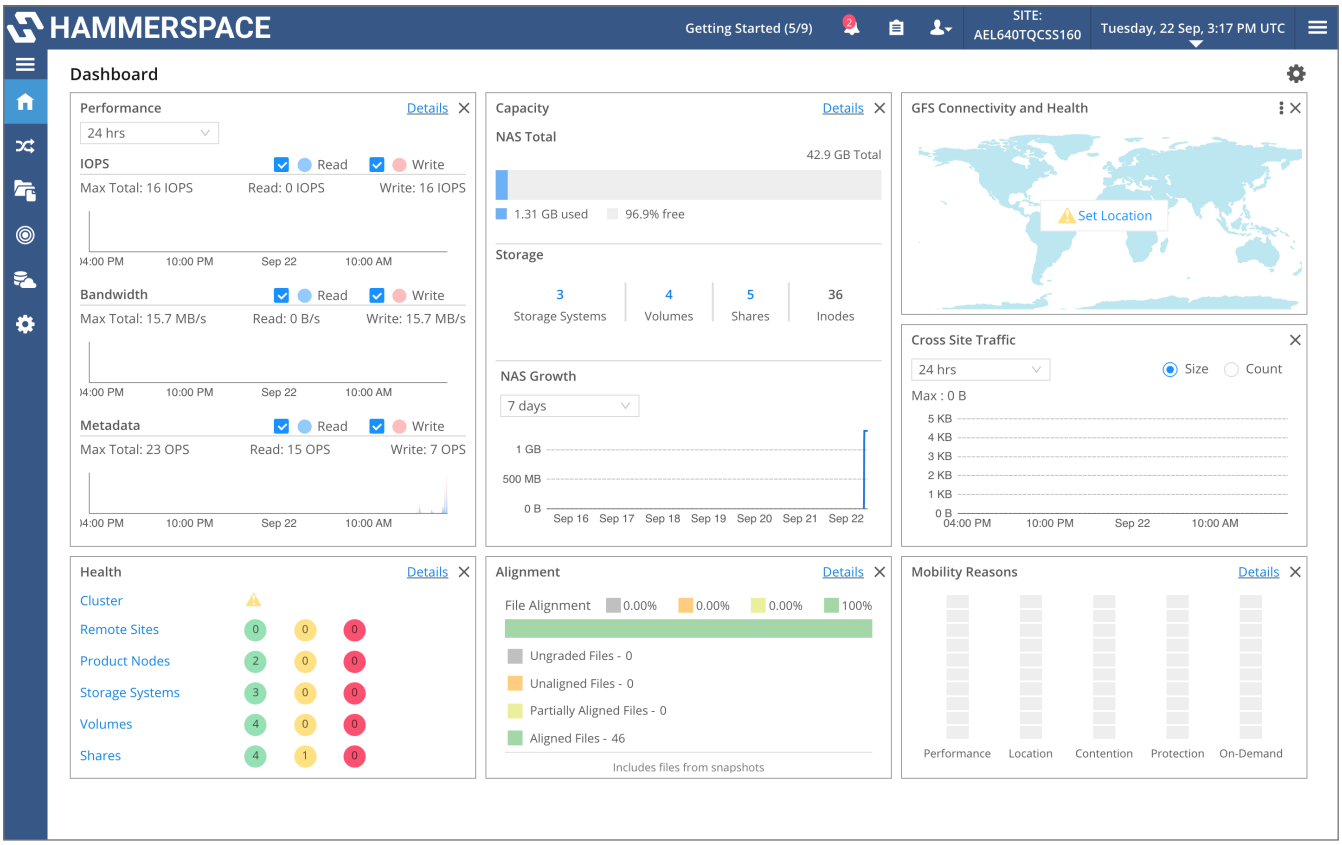


Figure 39. Making the SMB alias browsable or not browsable



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>
```

7.7.13. 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

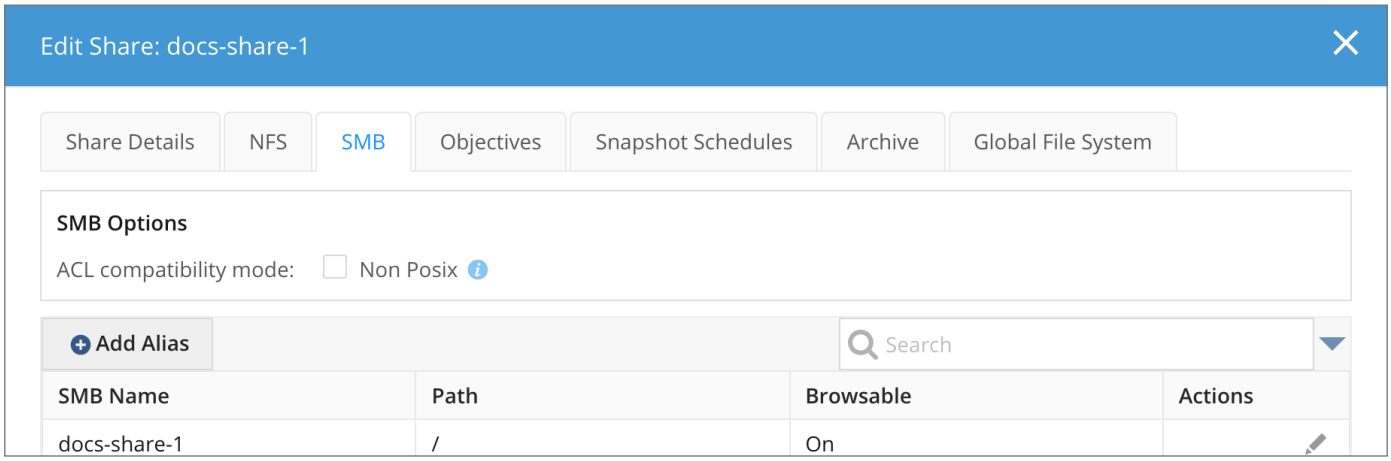


Figure 40. Add an SMB alias

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:

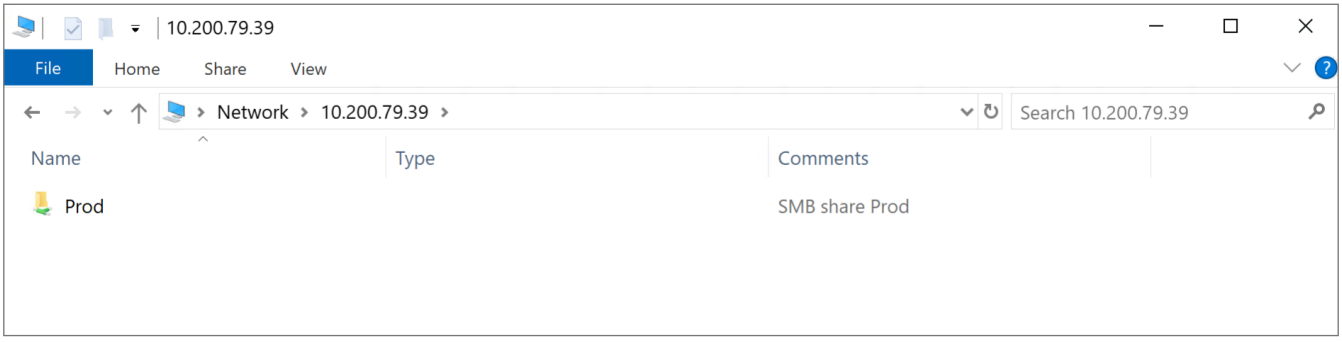
   

   Figure 41. SMB endpoint displaying a single share

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:

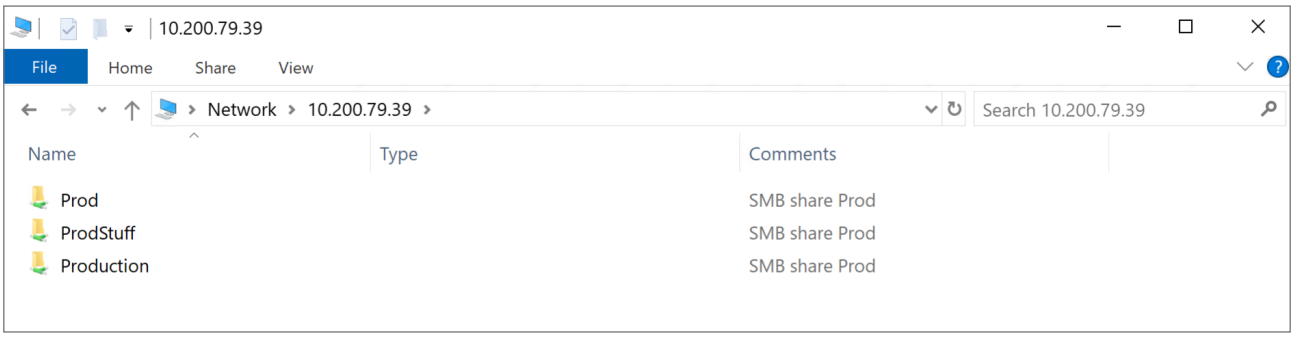


Figure 42. Shares visible in an SMB endpoint

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:

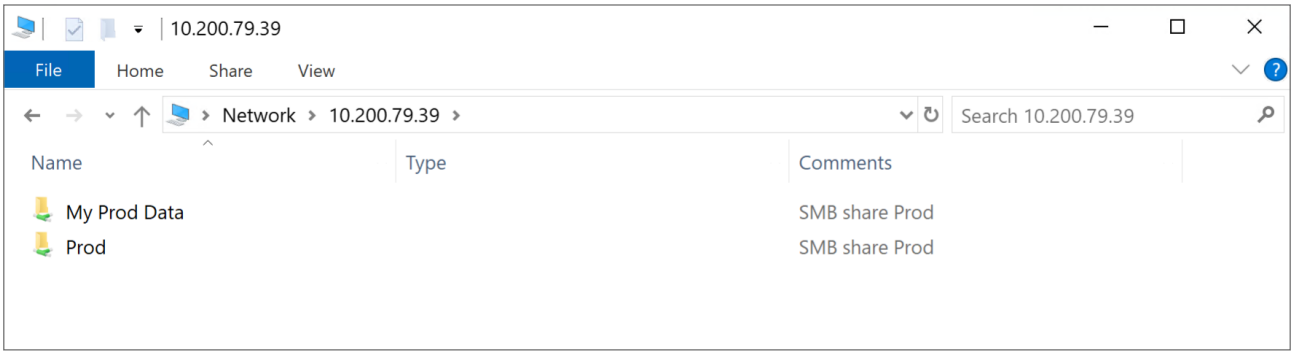


Figure 43. Share names with spaces

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:

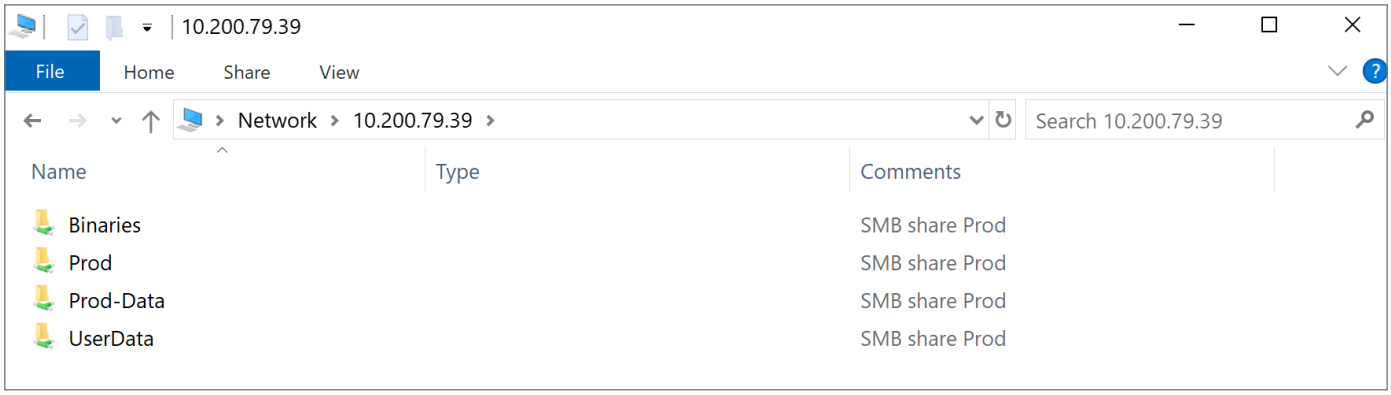


Figure 44. Share edited with three additional SMB endpoints

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:

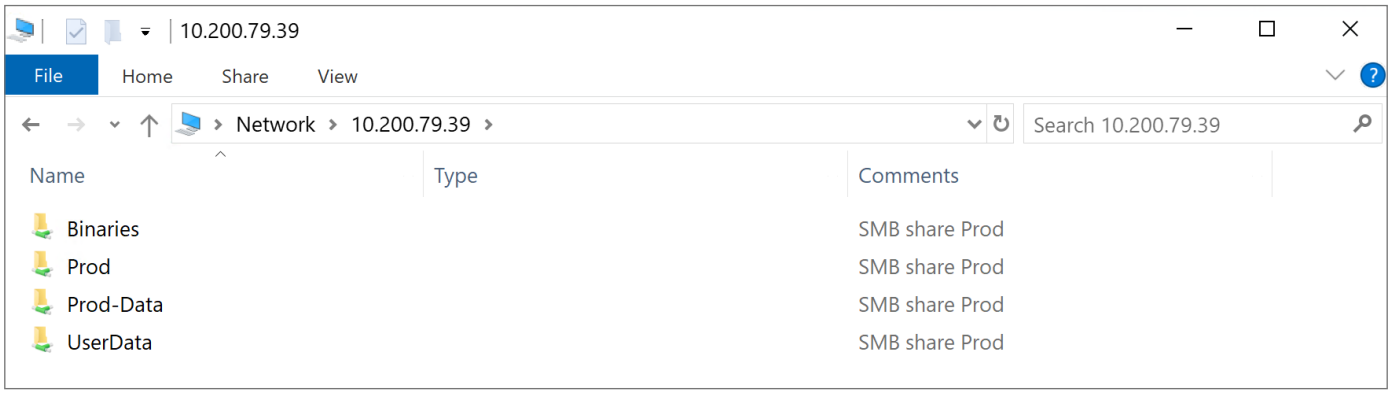


Figure 45. Hidden SMB share not visible

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

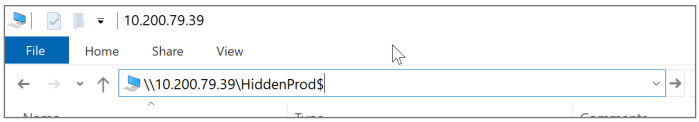


Figure 46. Hidden SMB share mountable by entering the name directly

7.7.14. 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 also applicable across protocols. When user mapping is properly configured, an NFS mount is also subject to the same ABE behavior. Users without read permission on NFS will not be able to see the folders or files.



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.

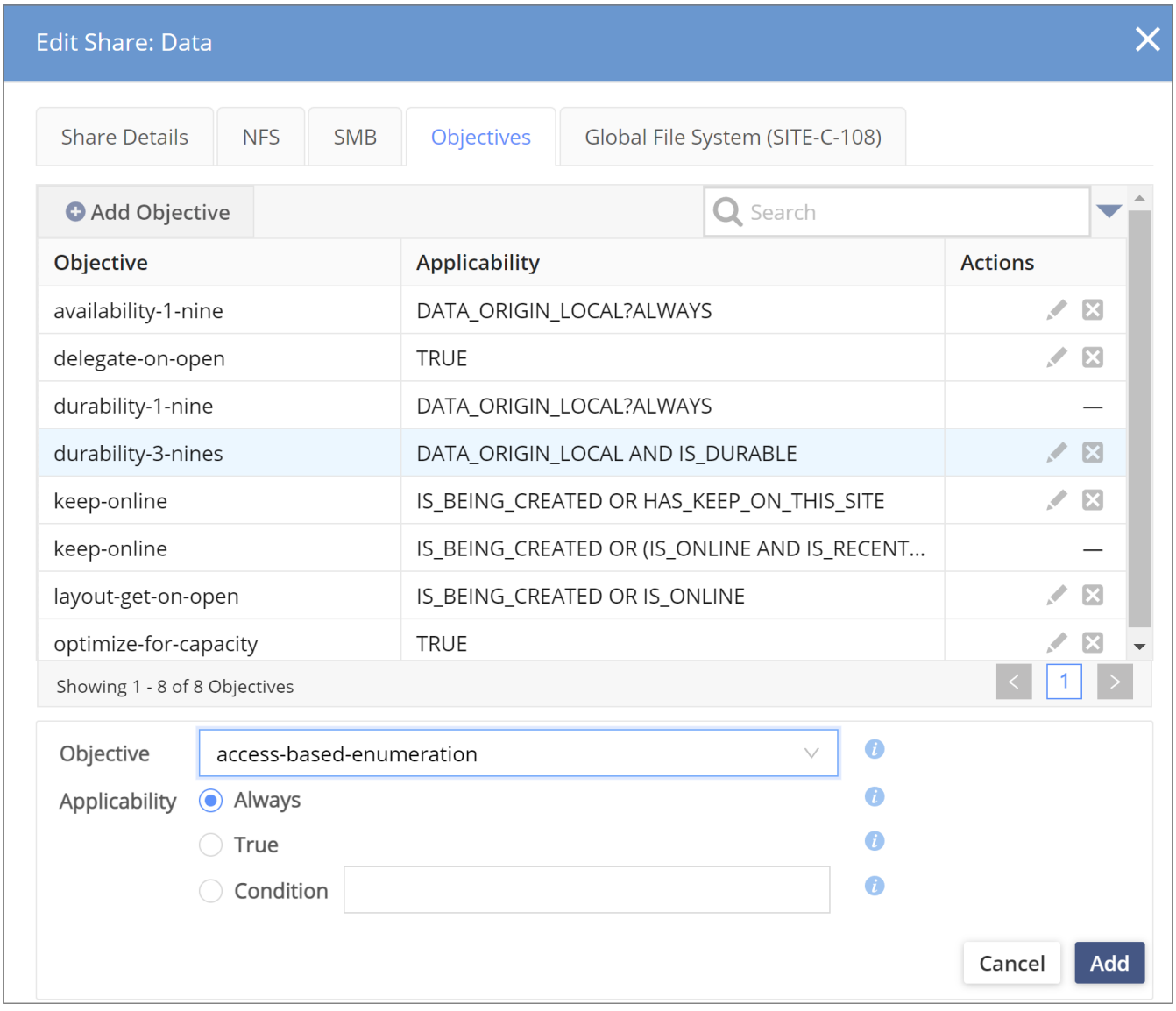


Figure 47. Share-level objective for access-based enumeration (ABE)

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.

© 2026 - Hammerspace

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.

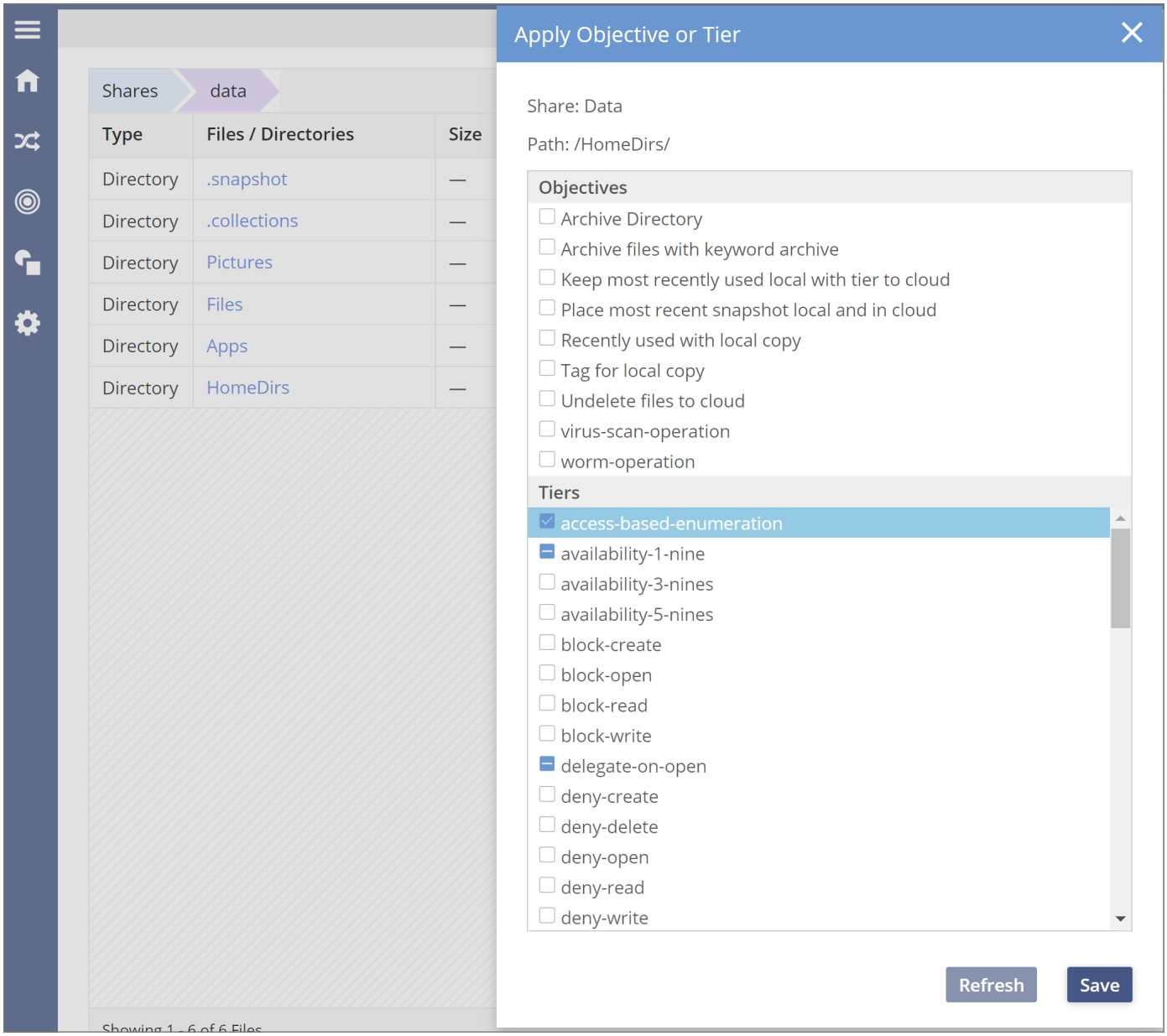


Figure 48. Subfolder objective for access-based enumeration (ABE)

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.

© 2026 - Hammerspace

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

7.7.15. 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
```


© 2026 - Hammerspace

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

Command:

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

Expected output:

```
success
```

7.7.16. 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.

© 2026 - Hammerspace

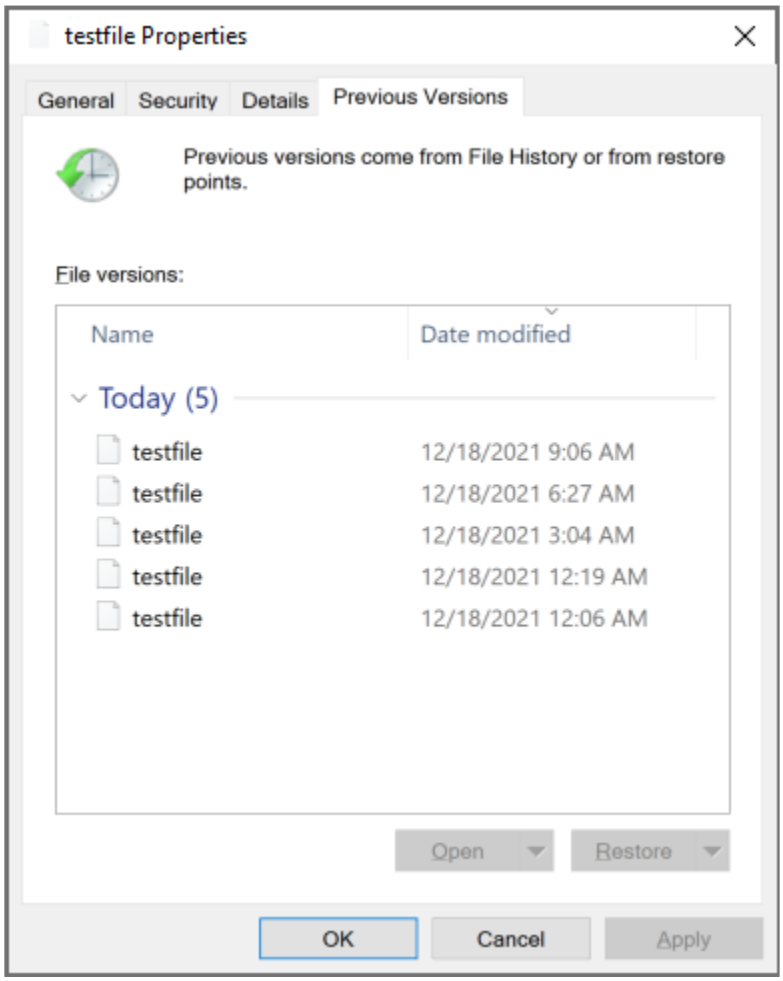


Figure 49. Windows file properties: Previous Versions tab



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

7.7.17. 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

© 2026 - Hammerspace

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**.

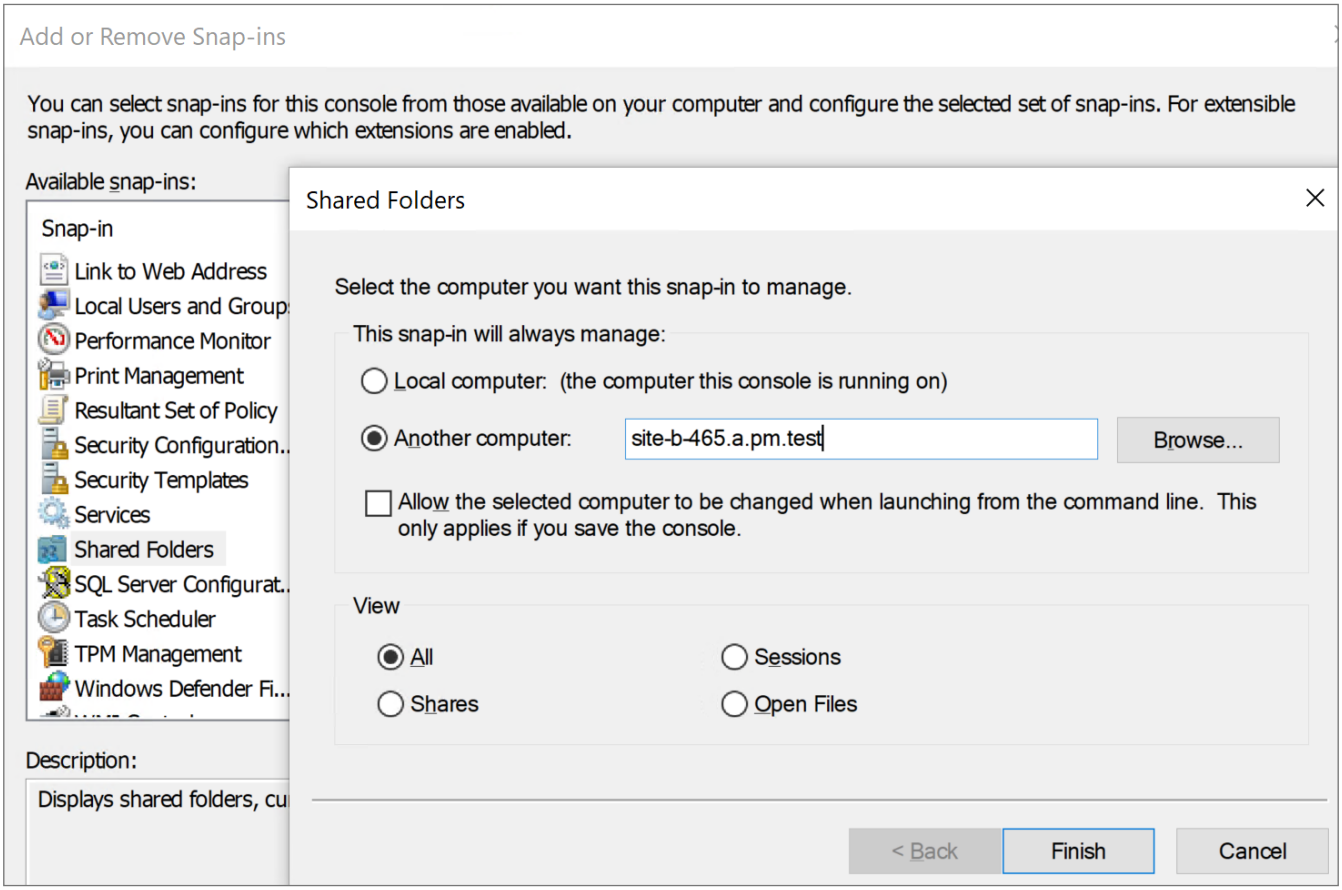


Figure 50. Connect to SMB server name (FQDN)

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.

© 2026 - Hammerspace

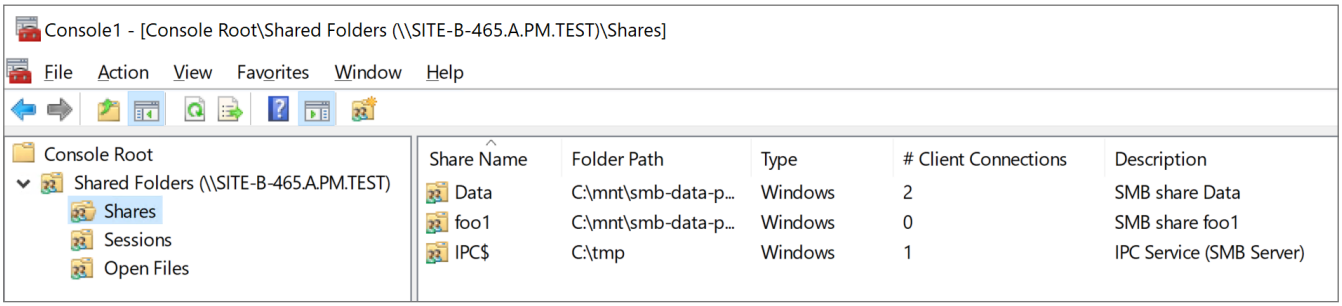


Figure 51. Share listing in Microsoft Management Console (MMC)

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

© 2026 - Hammerspace

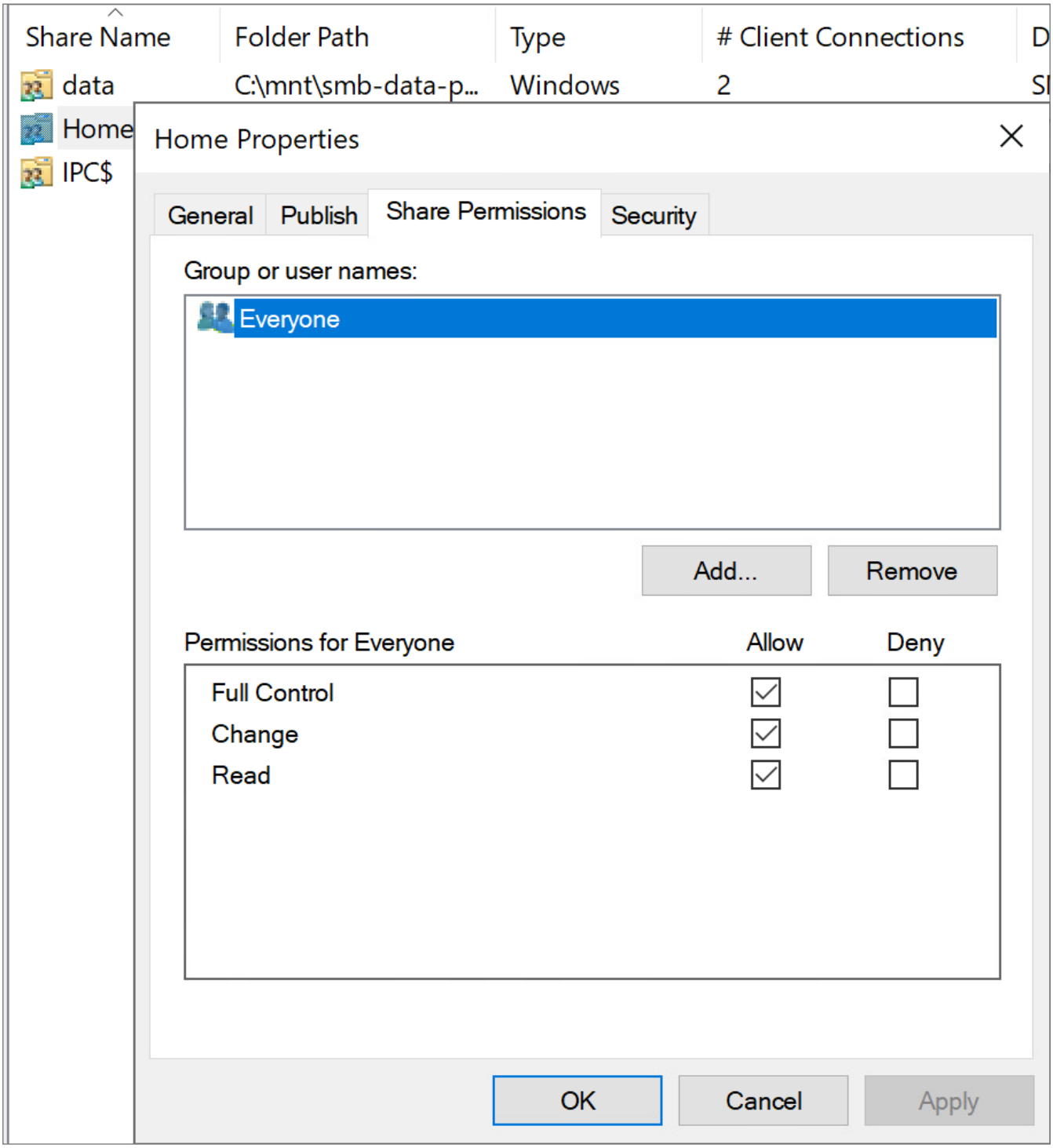


Figure 52. Share permissions

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

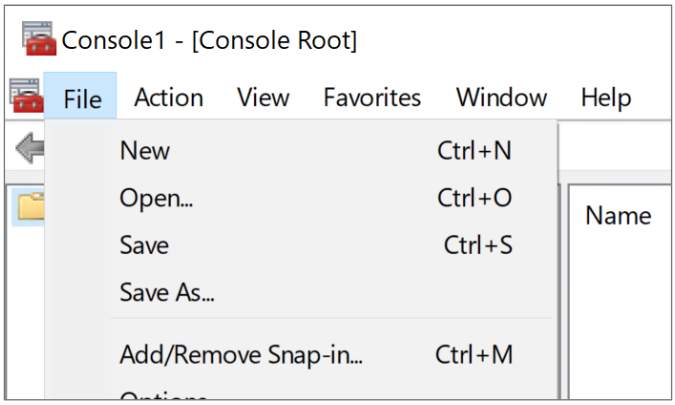


Figure 53. Location of Add/Remove Snap-in control

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

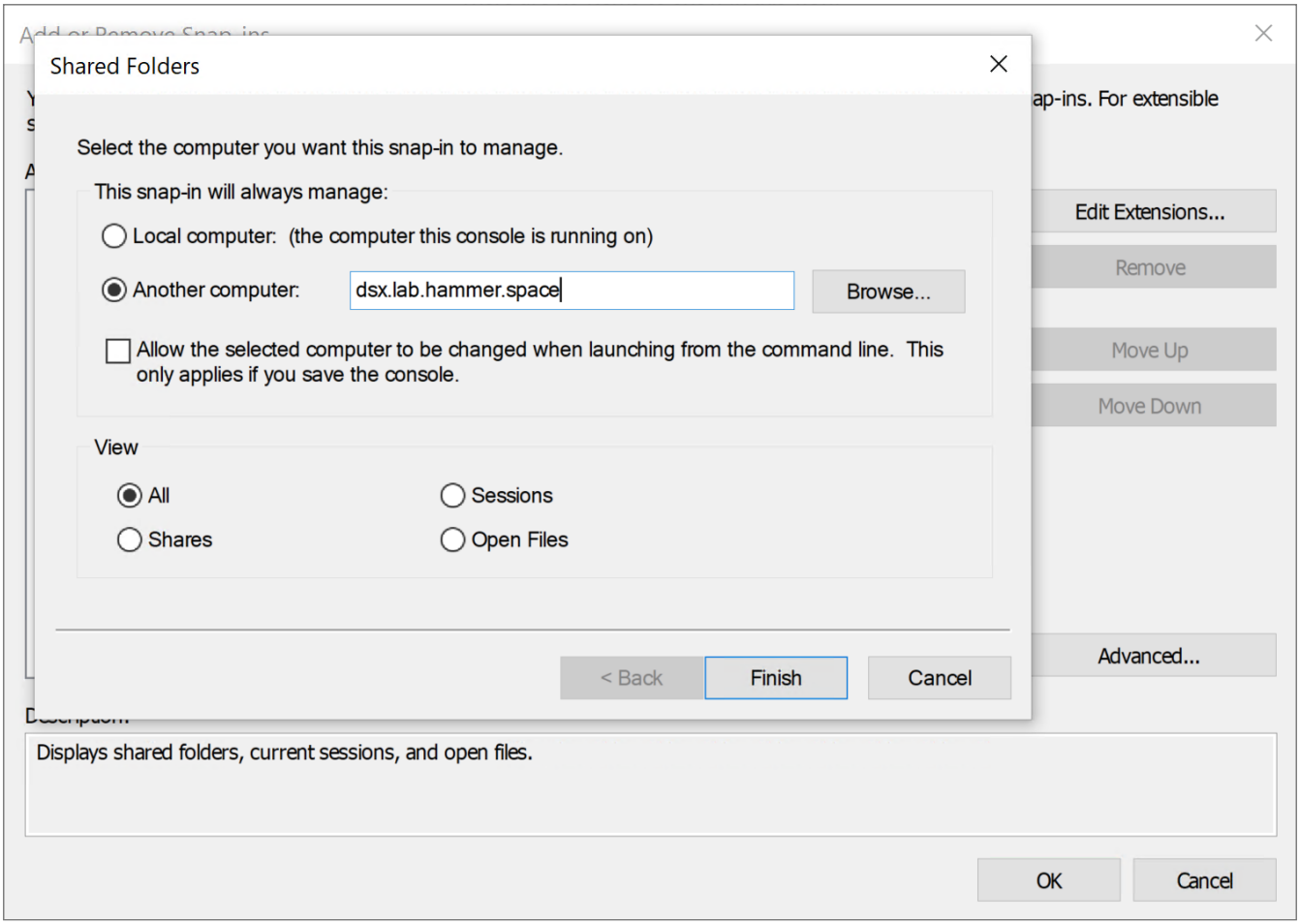


Figure 54. Associating Snap-in with the DSX node

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

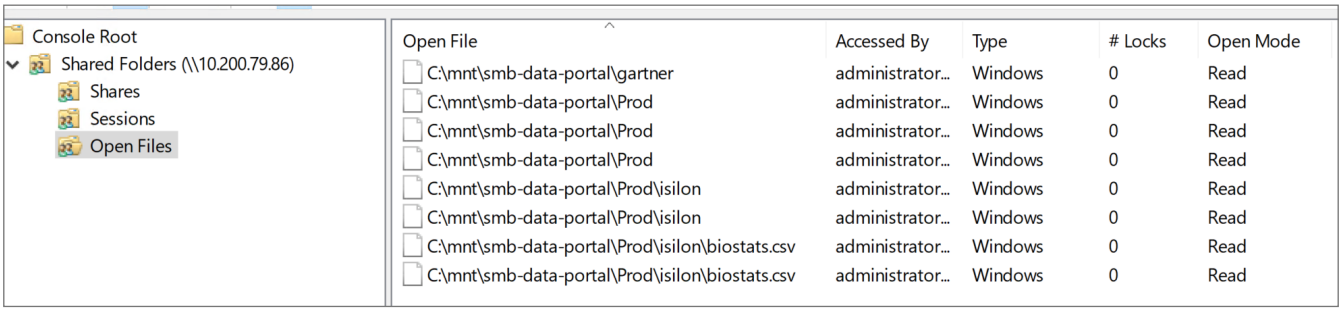


Figure 55. Viewing open files

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

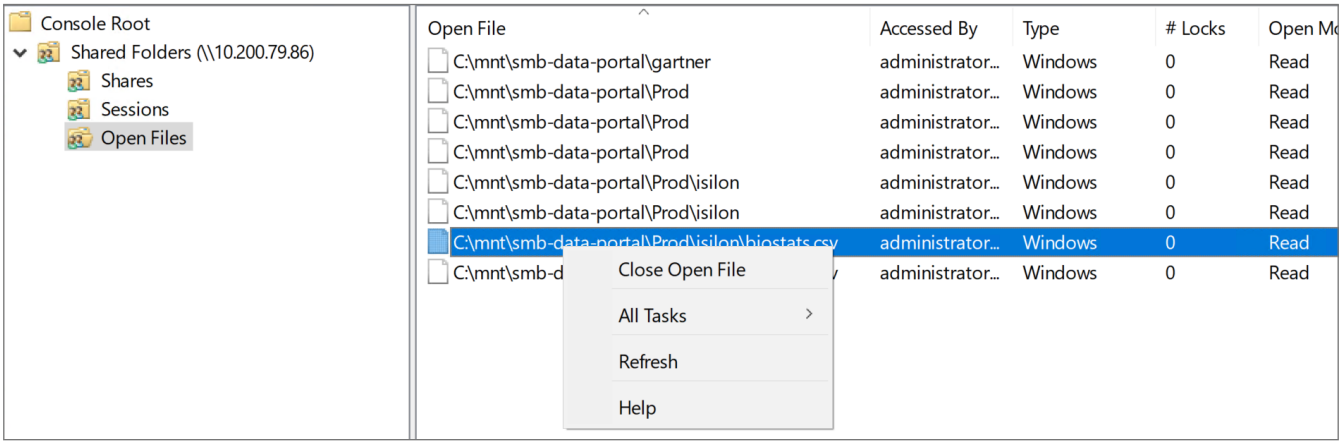


Figure 56. Closing an open file

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.

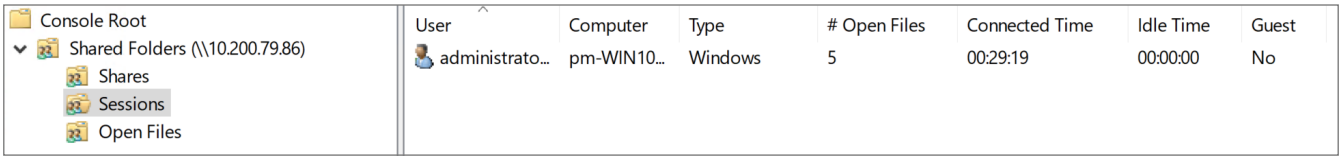


Figure 57. Viewing open sessions

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

© 2026 - Hammerspace

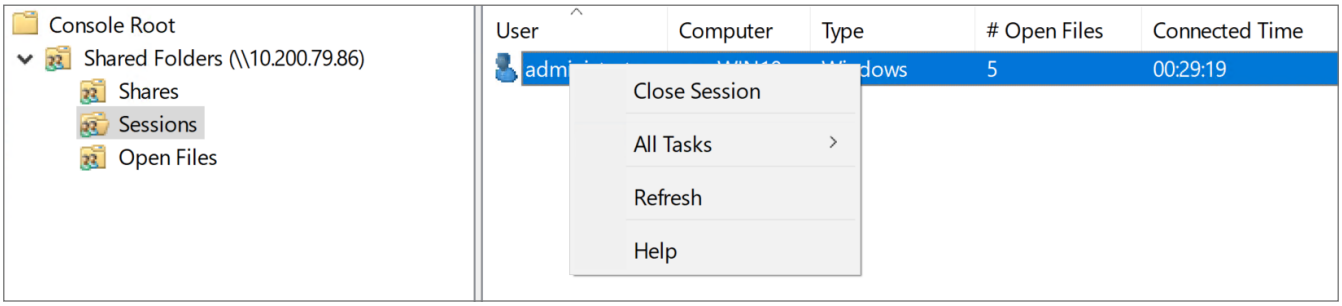


Figure 58. Closing an open session

7.8. 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.

7.8.1. 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.

7.8.2. 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.

7.8.3. 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
Maintain **service principal names that use a different name**, such as the Anvil DNS name used by NFS clients	Permission to write non-validated service principal names on the computer account. Granted automatically during a normal join. To grant or repair it later, the join account additionally needs Modify Permissions.	Computer account holds it; join account grants it	Computer object

7.8.4. 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

© 2026 - Hammerspace

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 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.

Service principal names built from a different name — the Anvil DNS name that NFS clients use — require an additional permission on the computer account. A normal Hammerspace join grants that permission automatically, so most sites never need to think about it. It becomes relevant when the computer object was created outside Hammerspace, or when the permission is later removed by AD policy or cleanup. In those cases Hammerspace raises the `KERBEROS_REQUIRED_NFS_SPNS_MISSING` event, and an administrator clears it by re-creating the NFS SPNs from the Active Directory configuration in the GUI, or by saving the Active Directory configuration again with credentials.

Note that restoring the permission does not by itself recreate the missing service principal names. After granting it, an administrator must run the recreate NFS SPNs operation, or change the Anvil DNS name, to rebuild them.

7.8.5. 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

© 2026 - Hammerspace

> permission list for an object.

7.8.6. 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
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
The SPN permission warning is cleared, but NFS Kerberos mounts still fail	The permission was restored but the service principal names were not rebuilt; run the recreate NFS SPNs operation

© 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.

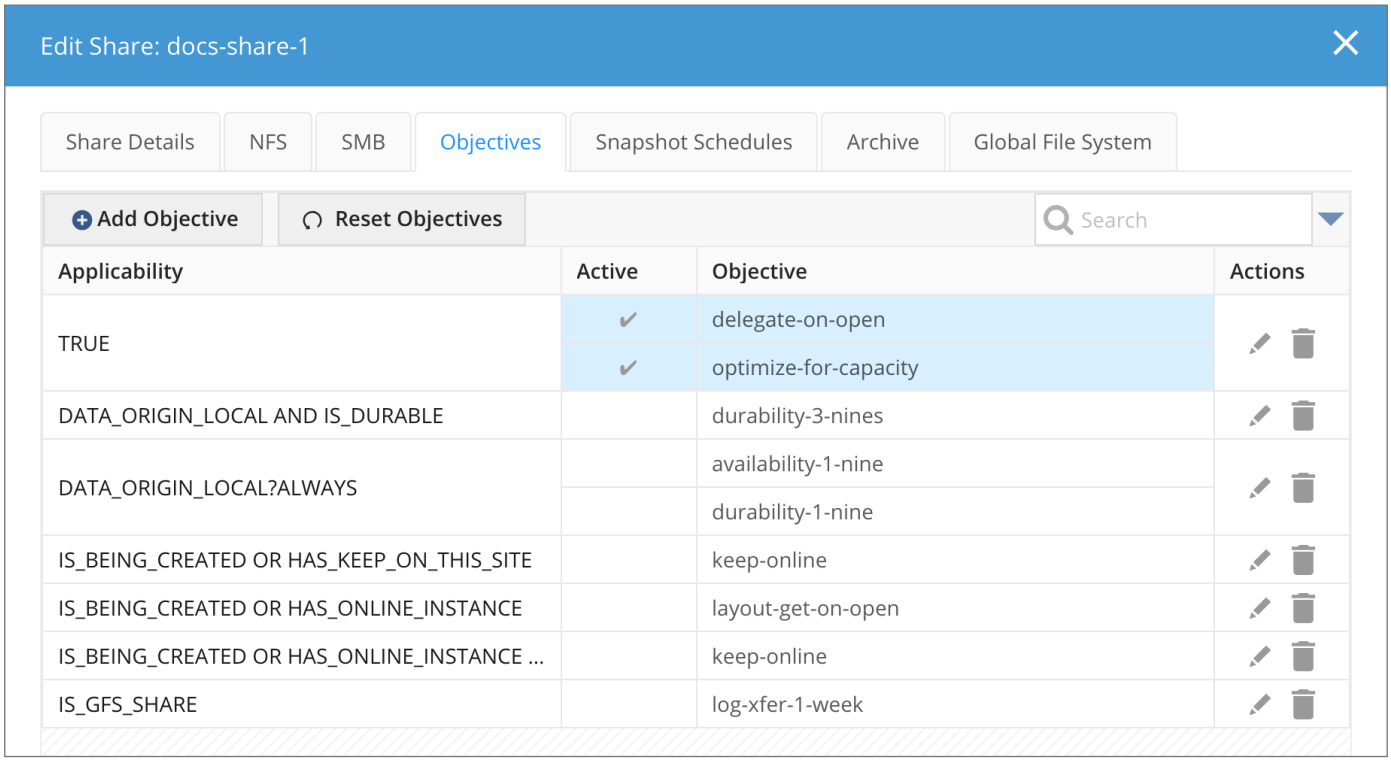


Figure 59. Editing share-level objectives

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.

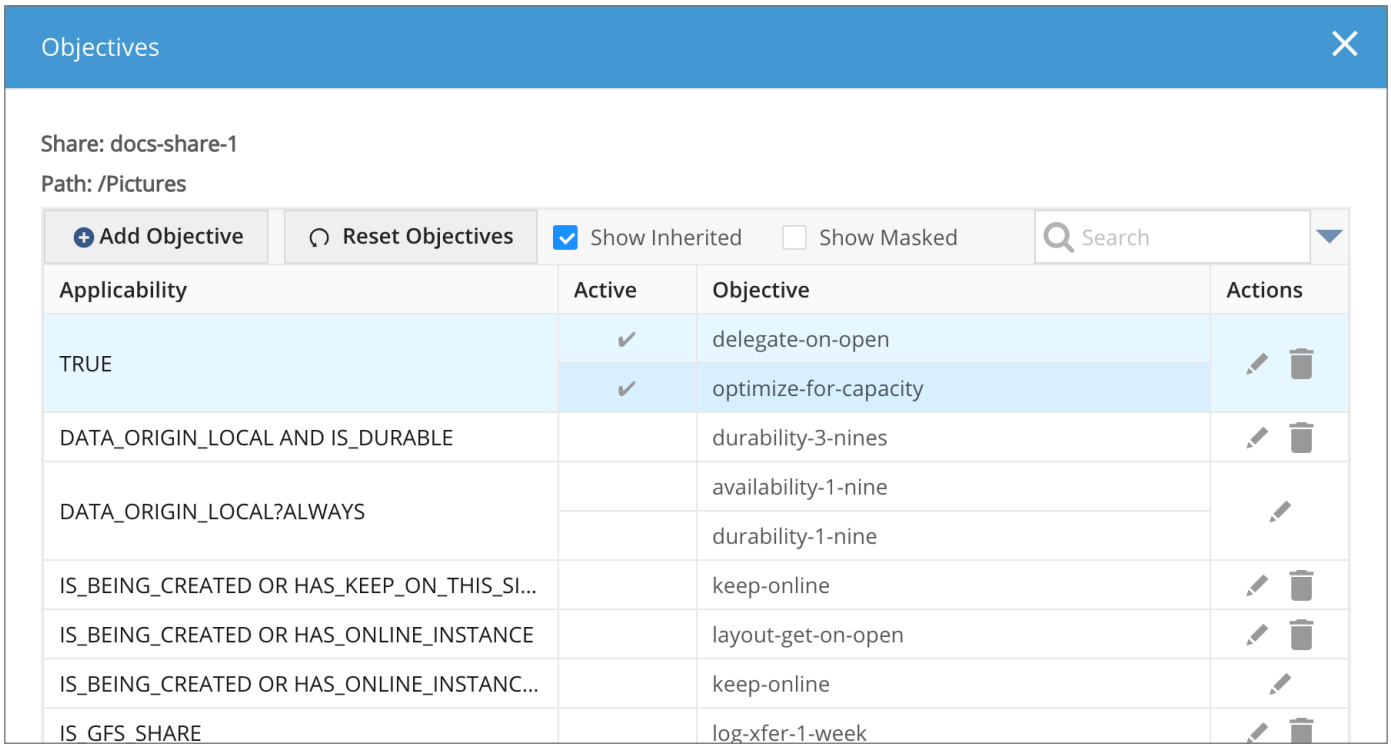


Figure 60. Editing directory-level objectives

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

© 2026 - Hammerspace

```
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.

© 2026 - Hammerspace

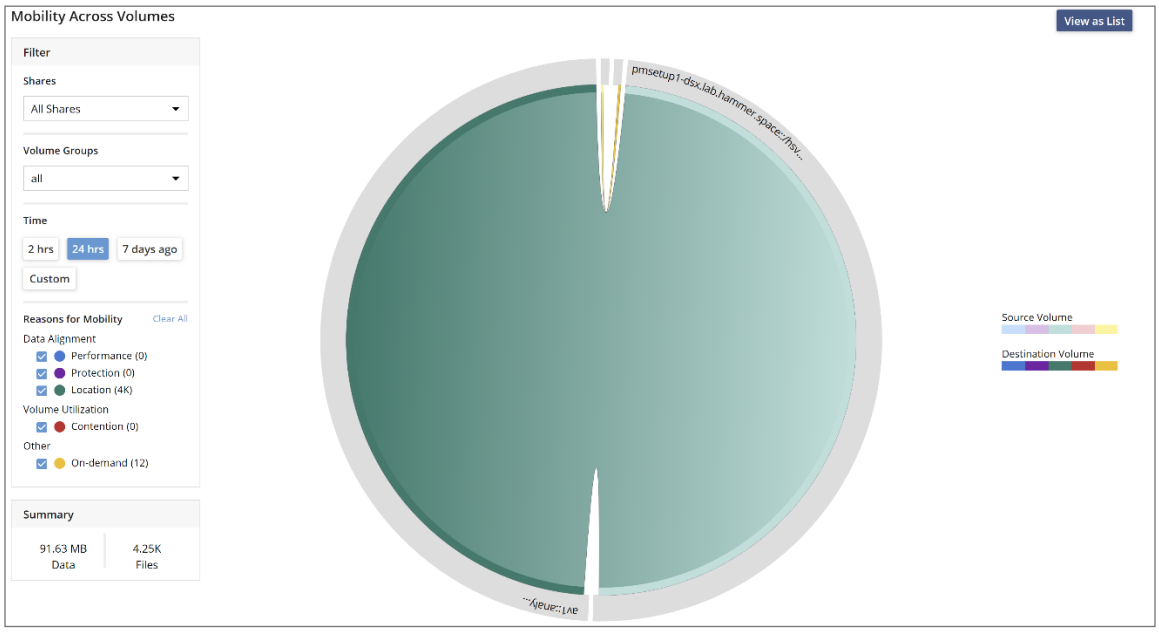


Figure 61. Mobility to anti-virus servers

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 {kb-assimilation-url}[{kb-assimilation-title}] 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.

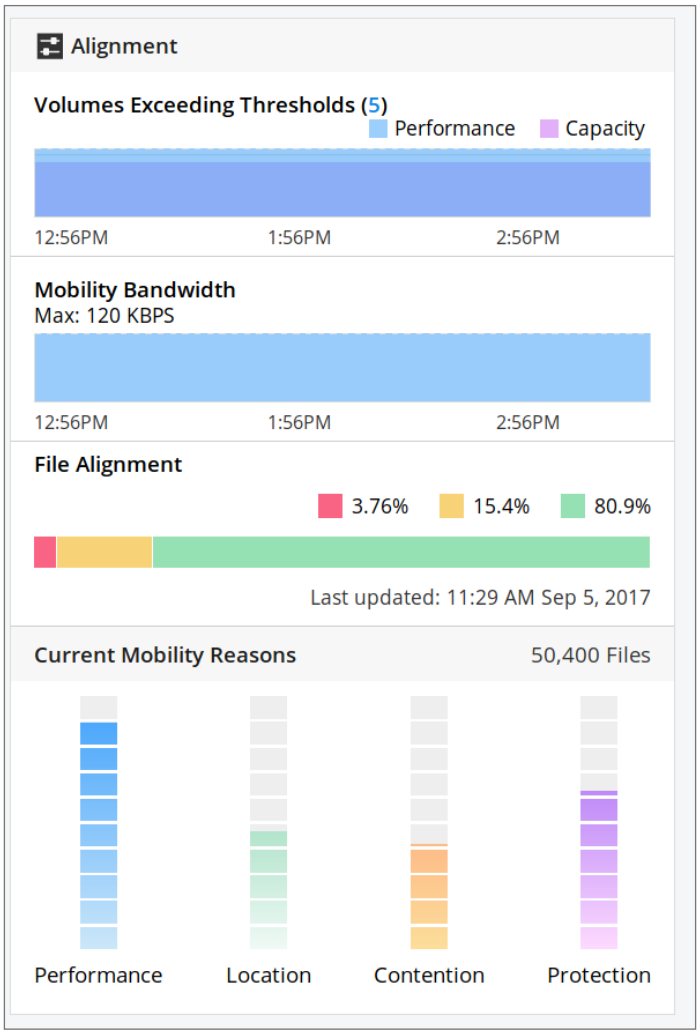


Figure 62. File alignment monitoring

8.6.2. Mobility History Chord Diagrams

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

© 2026 - Hammerspace

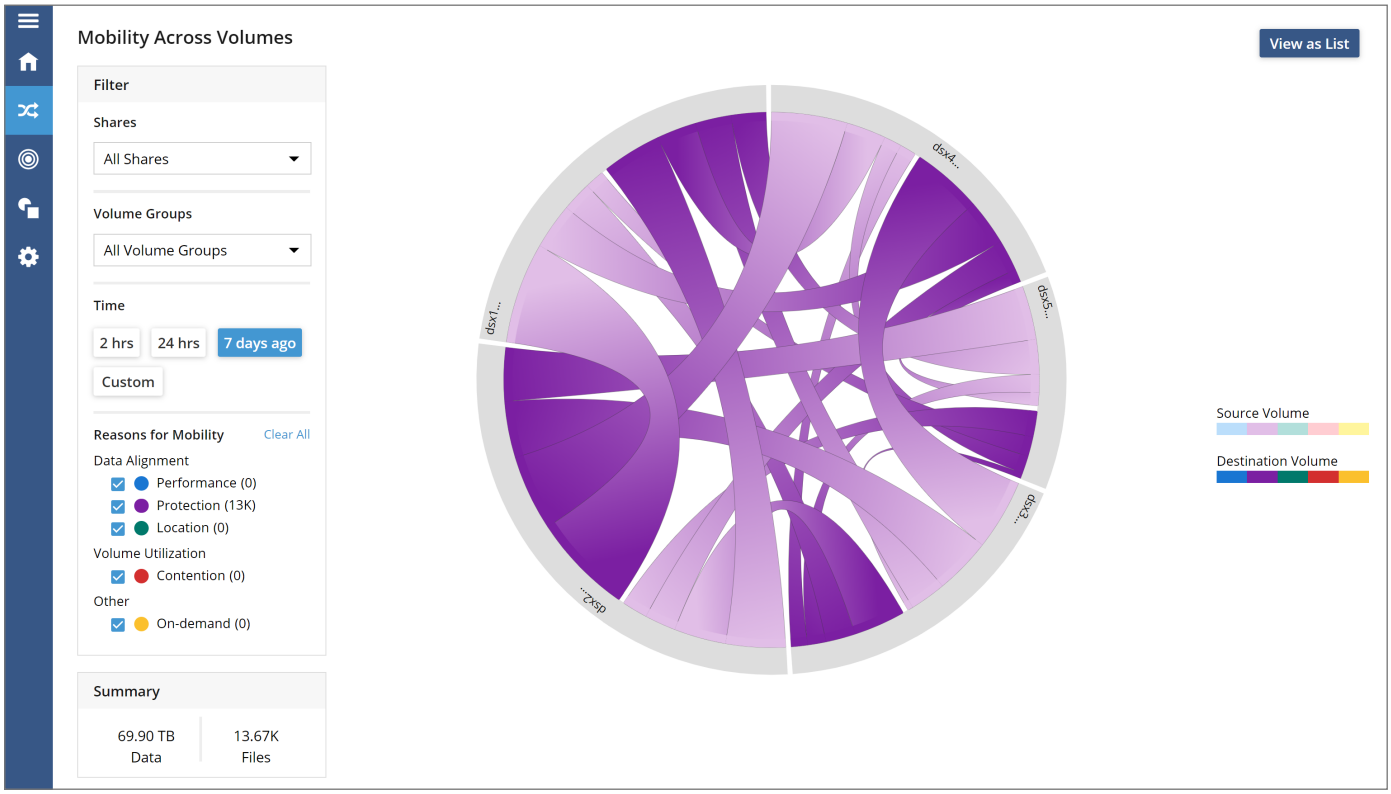


Figure 63. Mobility diagram example

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

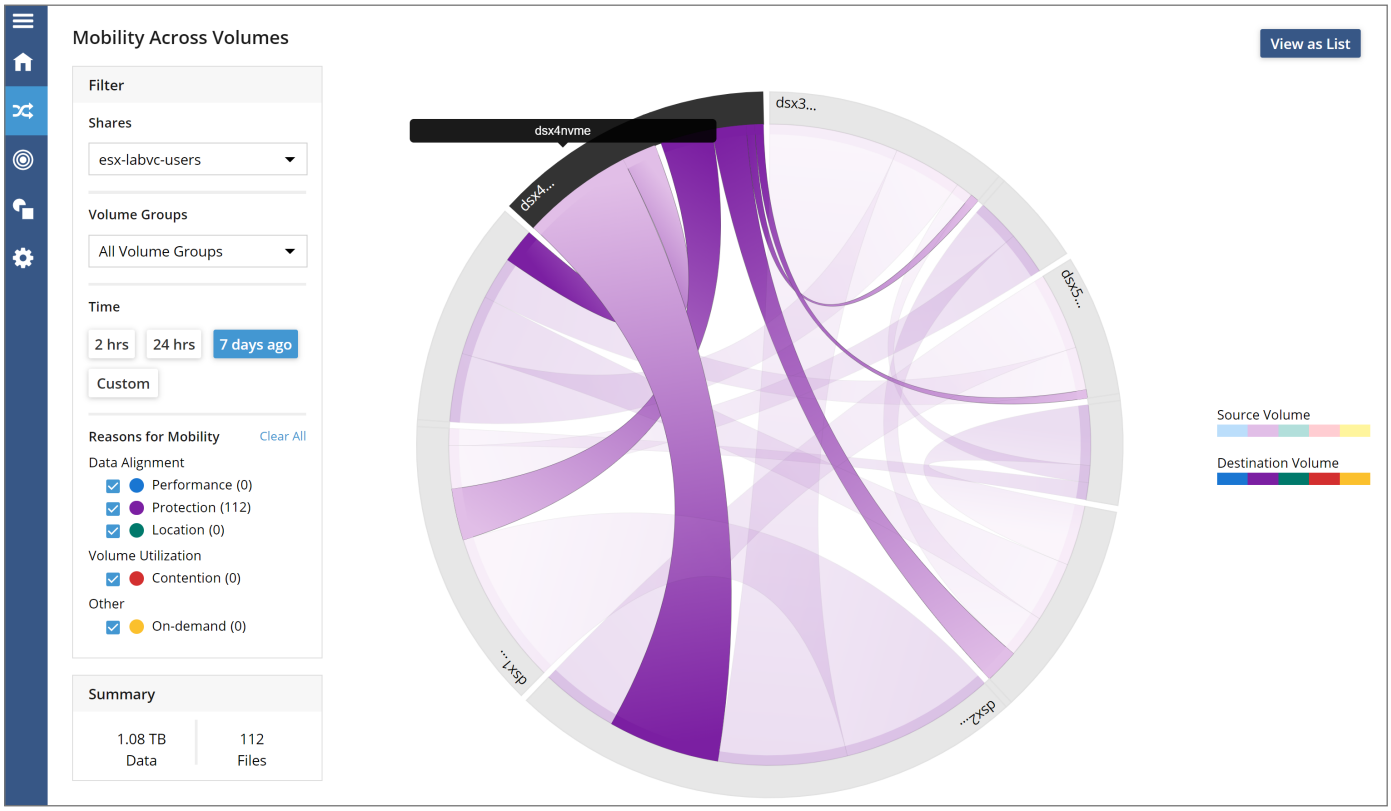


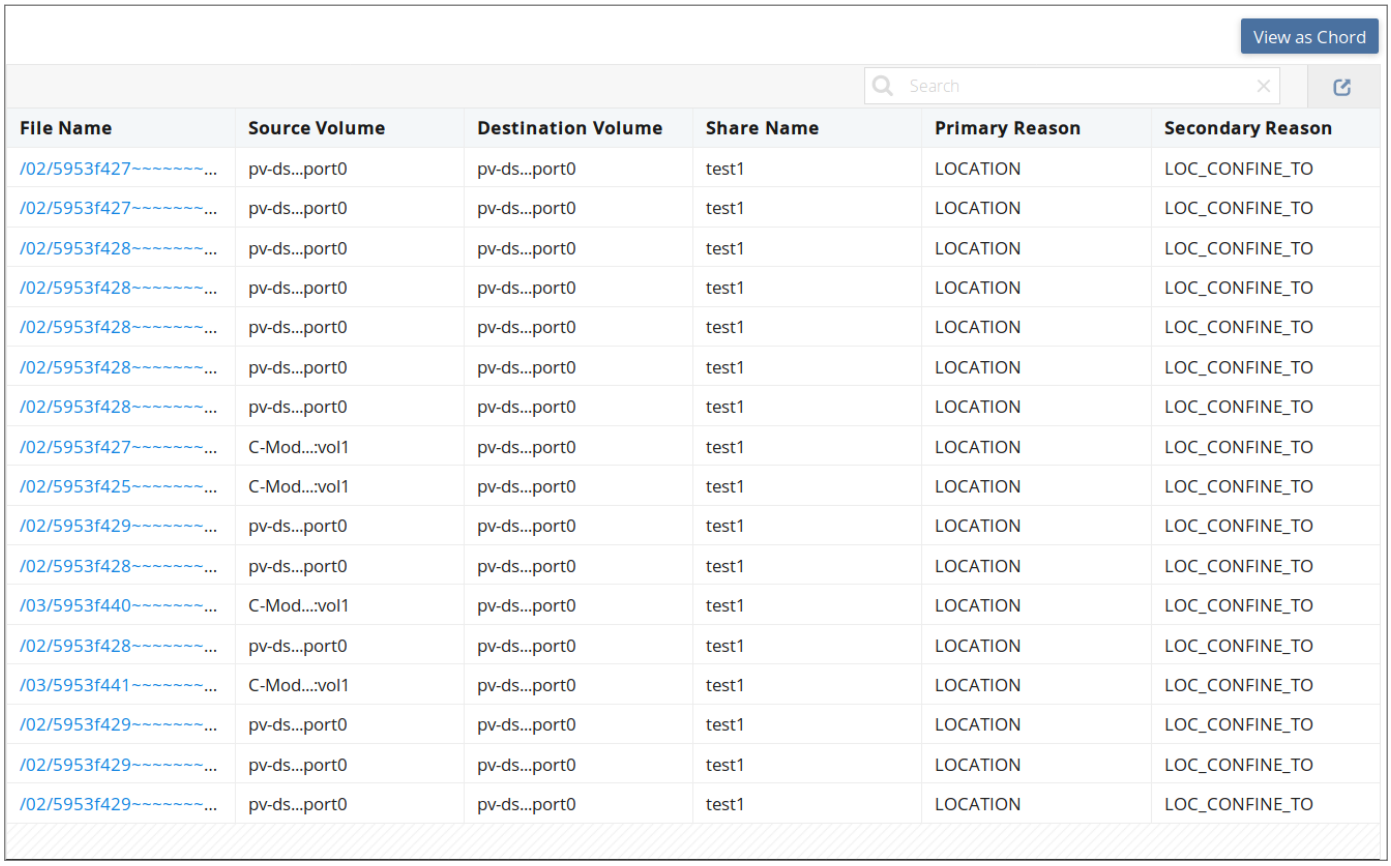
Figure 64. Hover over an entry to filter view to a single volume

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 65. 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.

Mobility "View as a list" shows [Unknown] filenames after upgrade. After upgrading to Hammerspace 5.3, the Mobility **View as a list** view may show **[Unknown]** in the filename column for files whose mobility activity occurred before the upgrade. This affects only the displayed mobility history; data mobility itself is unaffected. Mobility activity recorded after the upgrade displays filenames correctly, and the older entries clear automatically as the pre-upgrade history ages out (approximately seven days). No action is required, and there is no manual refresh to update the older entries sooner.

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.

© 2026 - Hammerspace

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.
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.

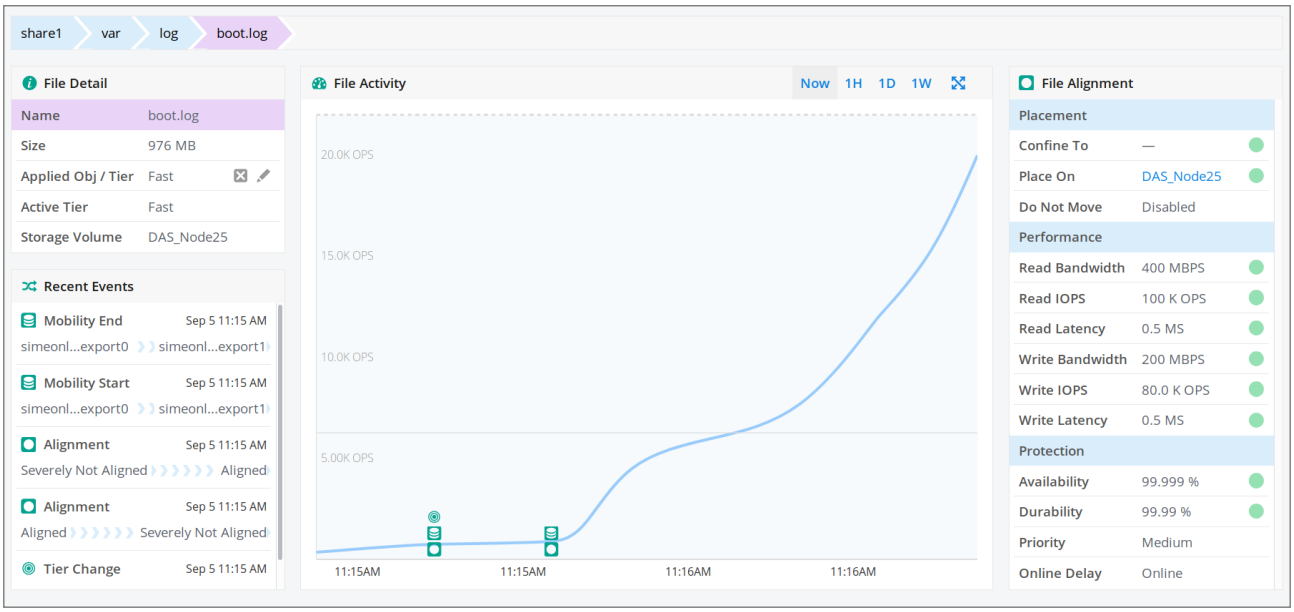


Figure 66. Viewing file IOPs over time

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**.

© 2026 - Hammerspace

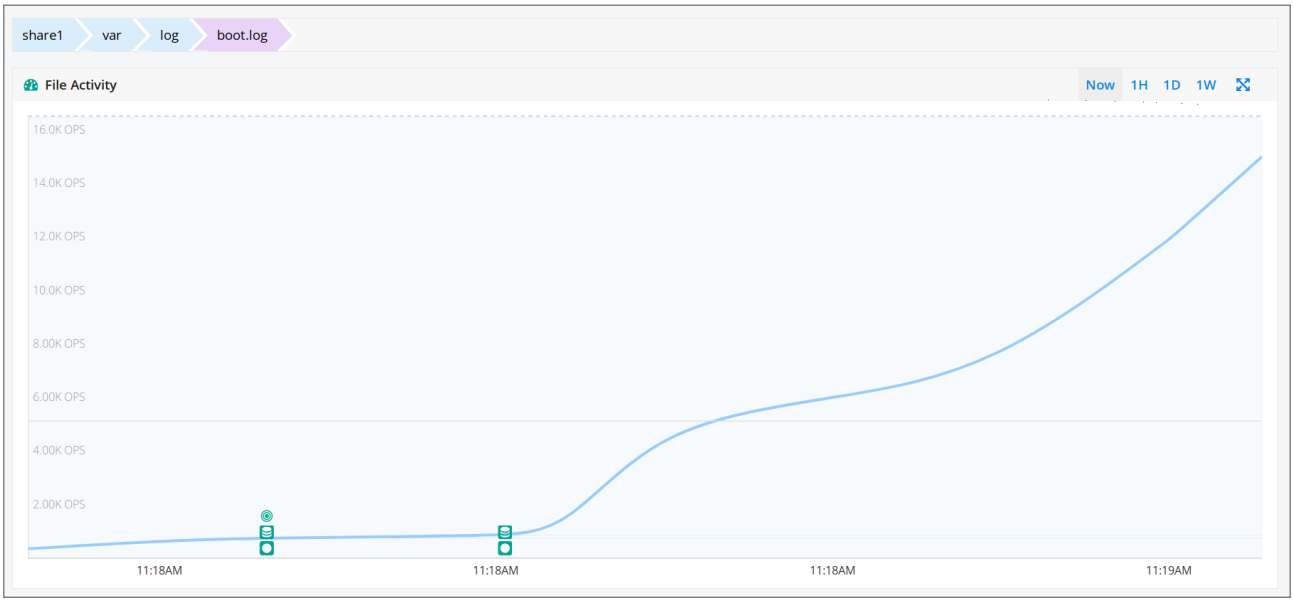


Figure 67. Selecting different look-back periods for file IOPs history

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.



The file counts shown in the management GUI's alignment statistics include files in snapshots. The `hs usage alignment /hs/<share>` command does not count files under the share's `.snapshot` directory, so the GUI can report several times more files than the CLI for the same share. This is expected behavior.

To reconcile the two figures, run the alignment command against the snapshot path:

```
hs usage alignment /hs/<share>/.snapshot
```

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.

© 2026 - Hammerspace

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 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.

© 2026 - Hammerspace

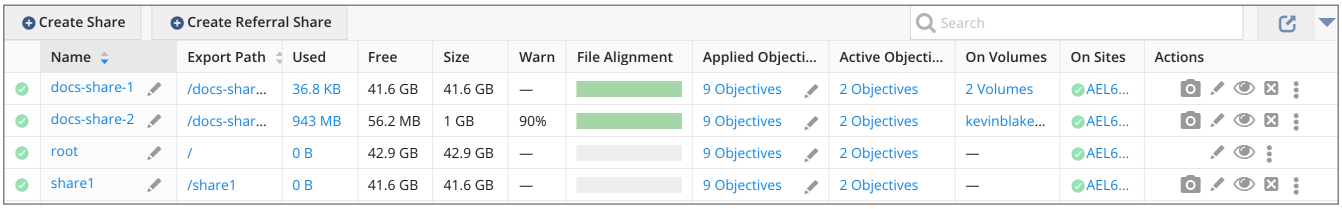


Figure 68. Editing a share

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**.

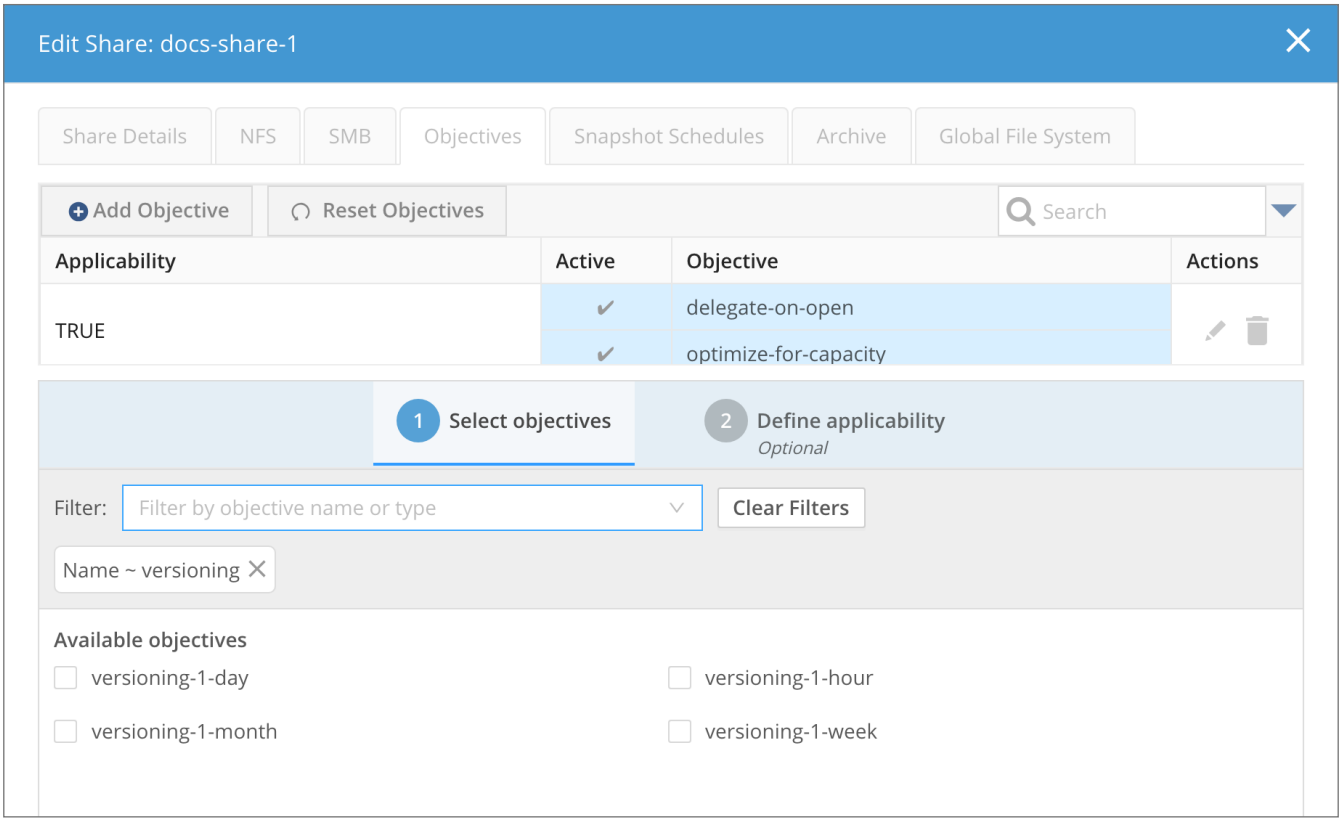


Figure 69. Filtering a set of objectives

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

© 2026 - Hammerspace

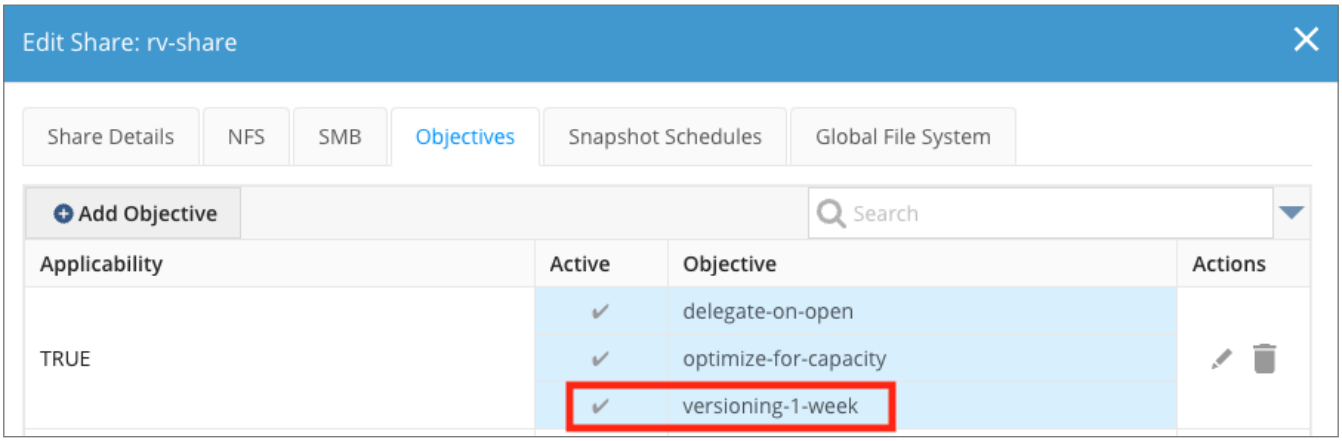


Figure 70. Verifying an objective is active

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.
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.

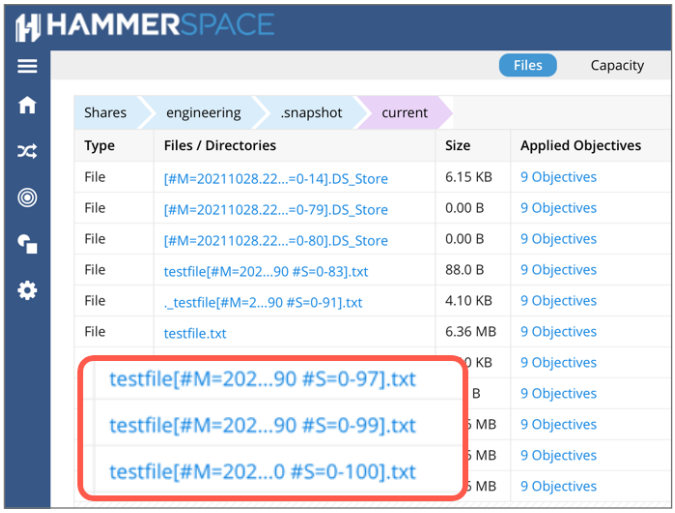


Figure 71. Viewing the list of available backups

© 2026 - Hammerspace

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:

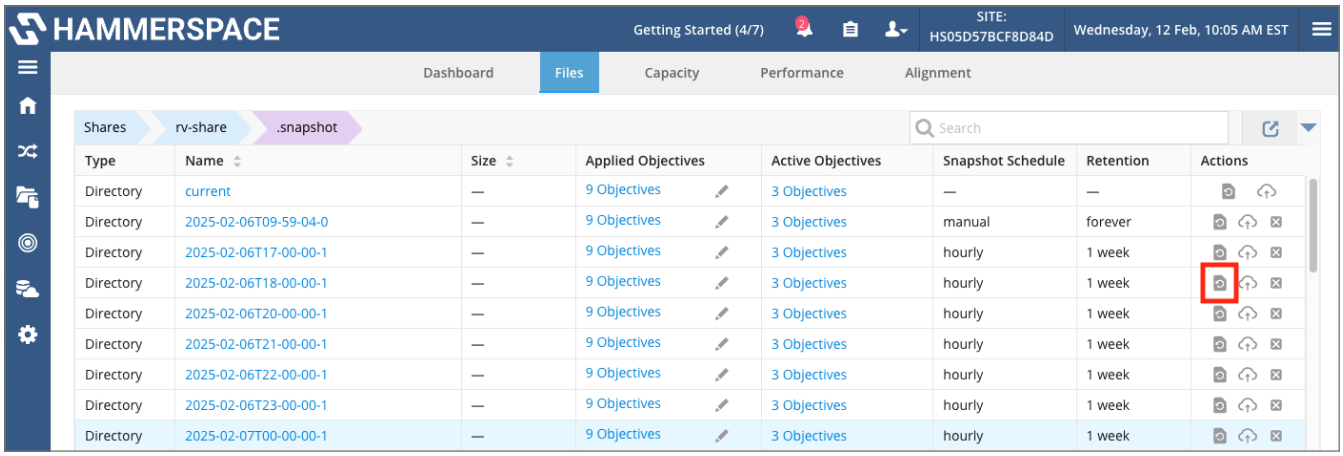


Figure 72. Location of Restore icon

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.

© 2026 - Hammerspace

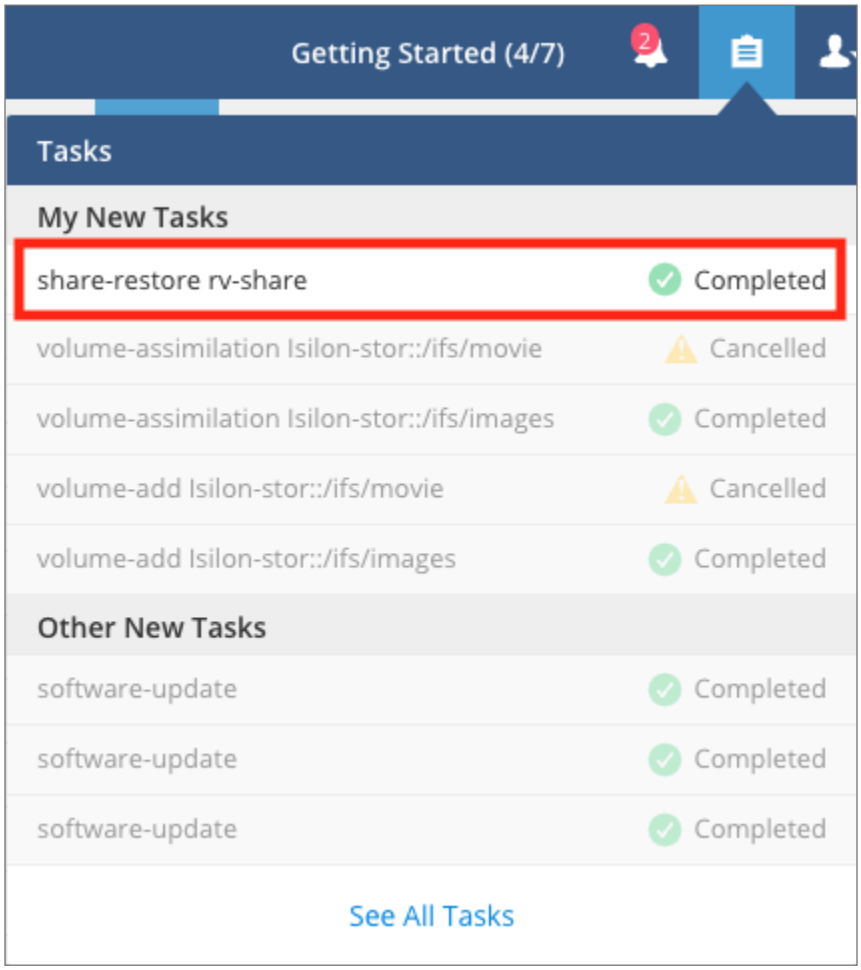


Figure 73. Viewing task-completion status

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

© 2026 - Hammerspace

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.

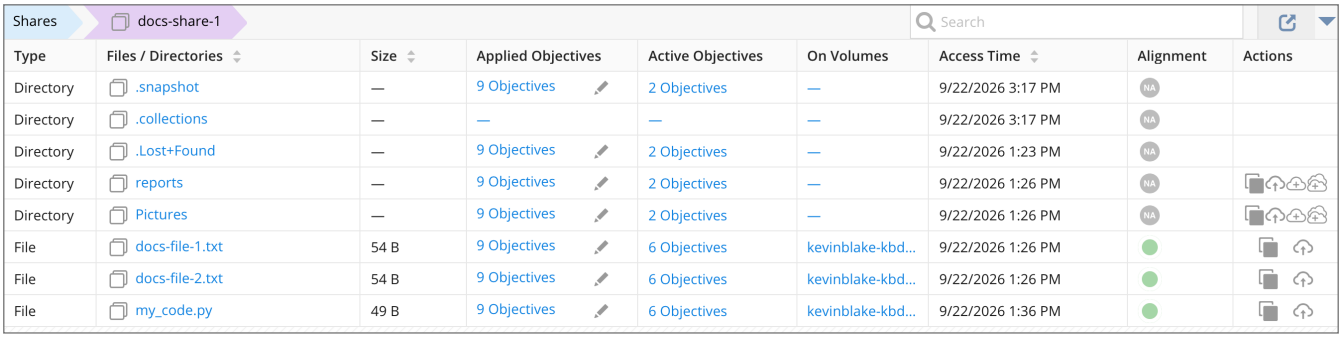


Figure 74. Protecting a directory with undelete

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**.

© 2026 - Hammerspace

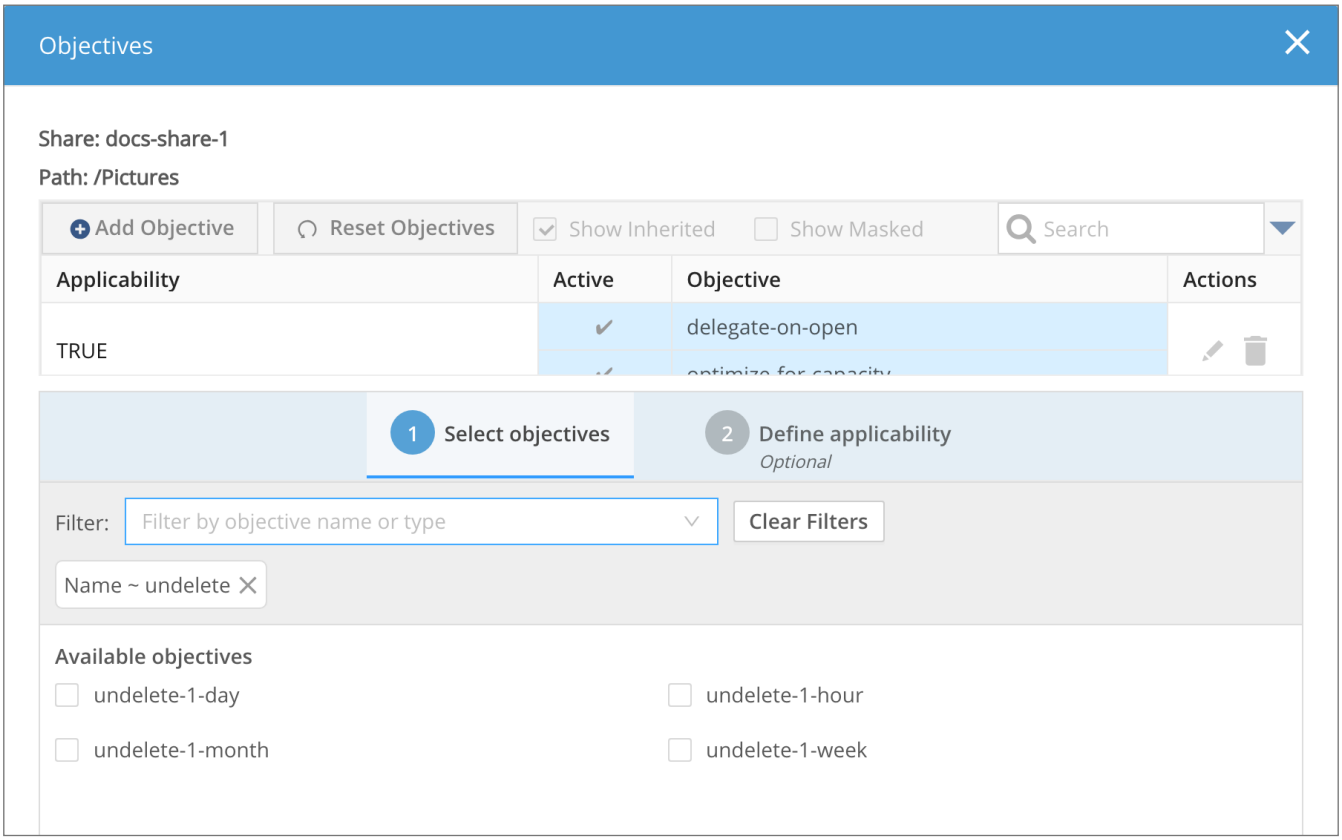


Figure 75. Selecting the term for undelete protection

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

© 2026 - Hammerspace

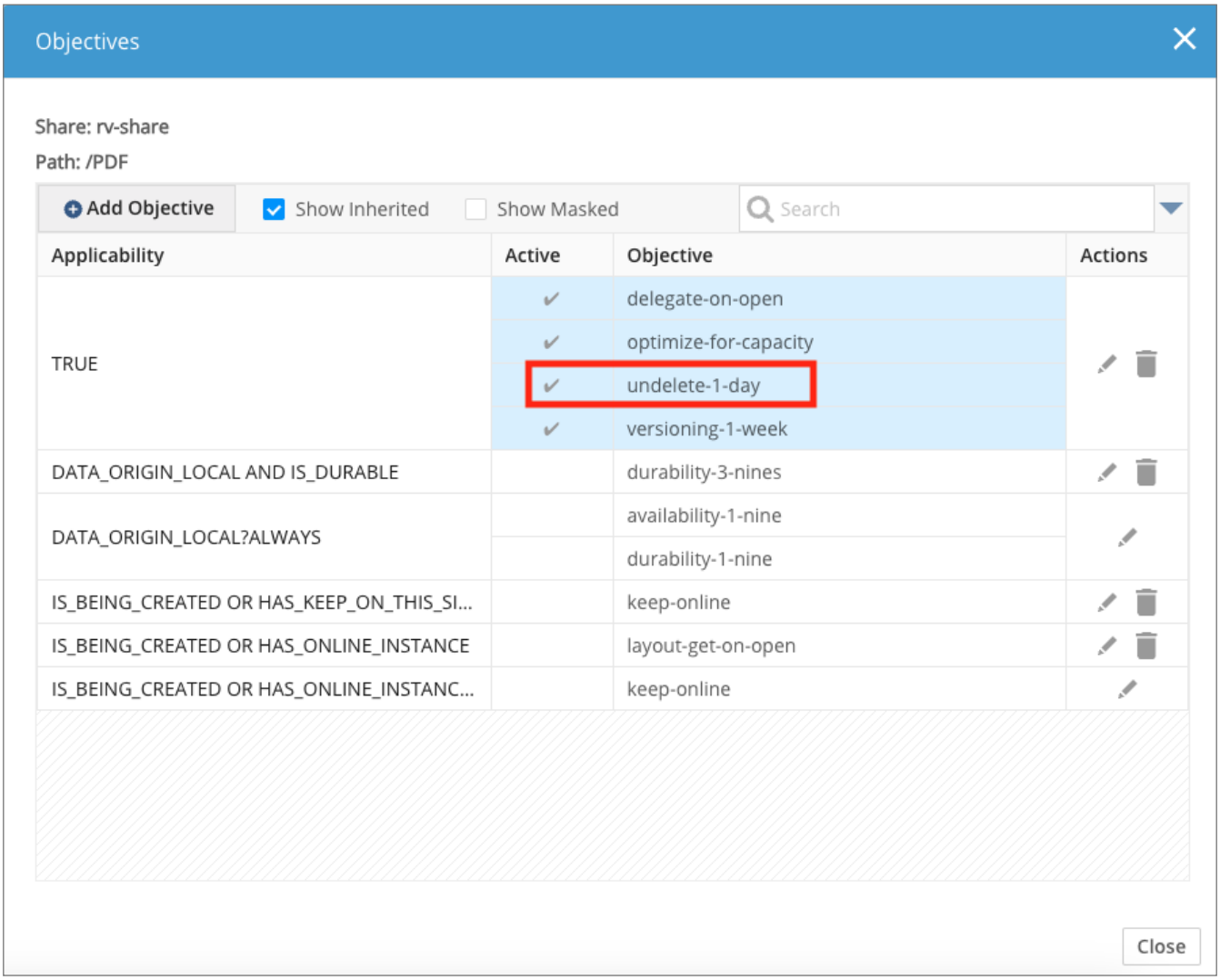


Figure 76. Verifying the objective is active

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.

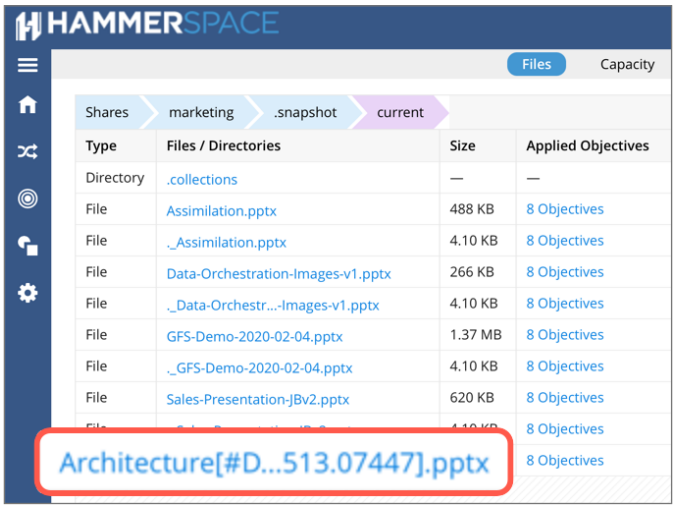


Figure 77. Locating a protected file

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

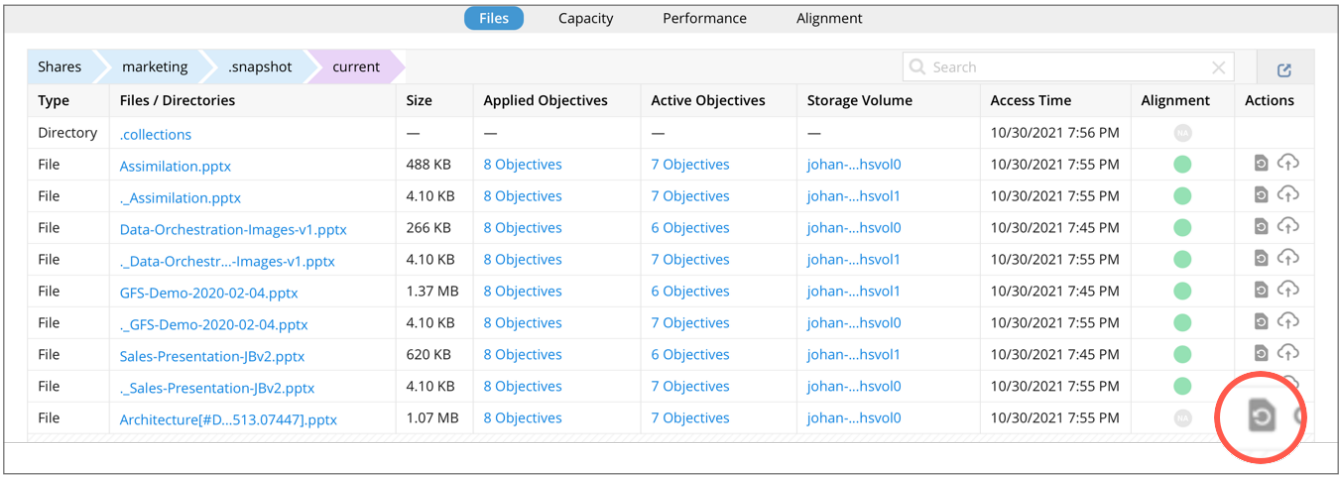


Figure 78. Restoring the file

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.

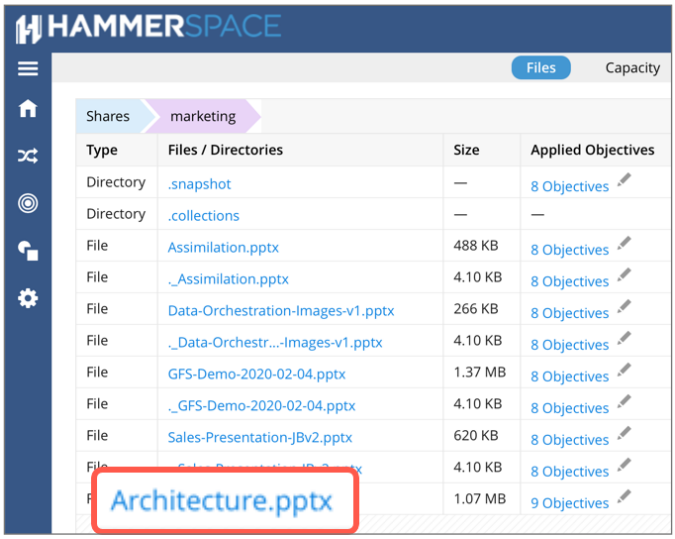


Figure 79. Locating the recovered file

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.

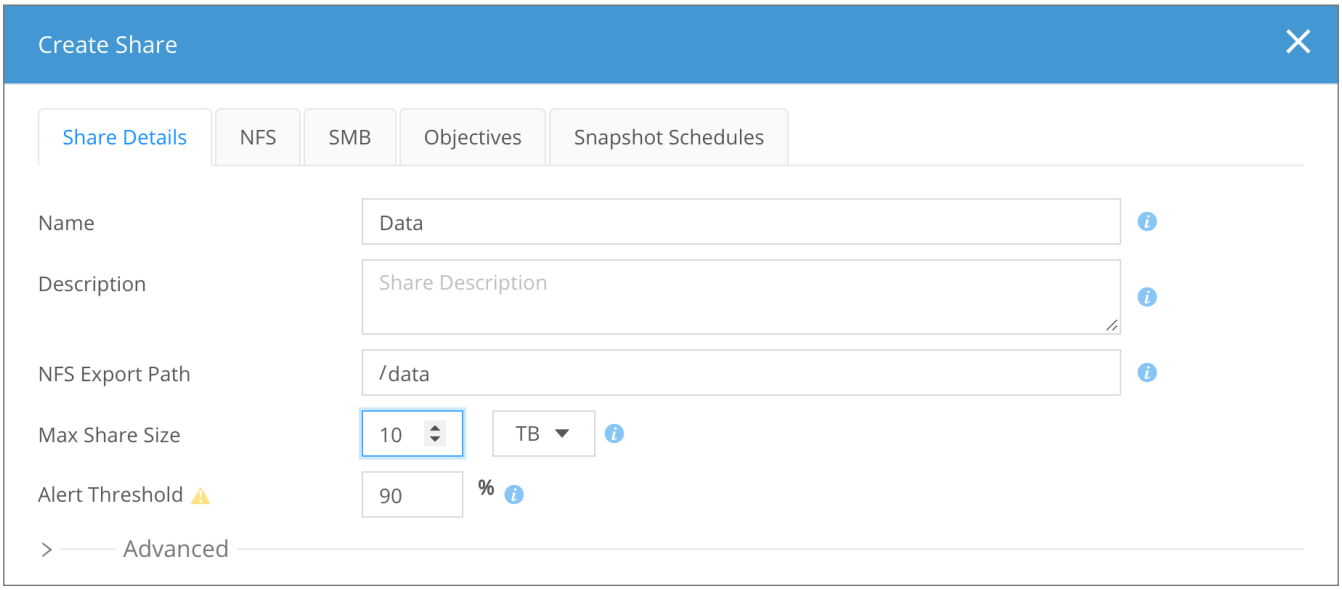
The file-backing storage is a common resource pool across all shares unless objectives have been applied to control where data is located.

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. This limit cannot be exceeded. The share size can be increased

© 2026 - Hammerspace

or decreased later by editing the share. The size cannot be decreased below the actual amount of data stored in the share.

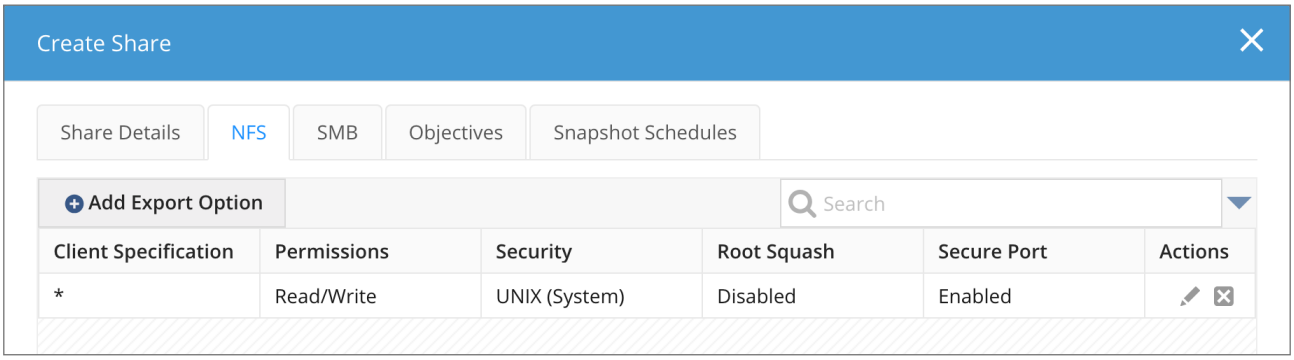
The default share size is the total physical capacity across file-based storage volumes and does not include cloud/object volume capacity. The setting for max share size is not constrained by available capacity and can be set to the desired size.

- **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.

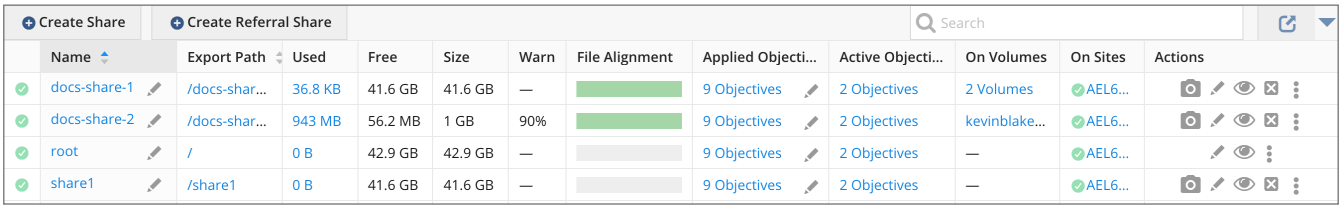
The **Advanced** section, collapsed by default, holds additional settings that are not required to create a share.

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.

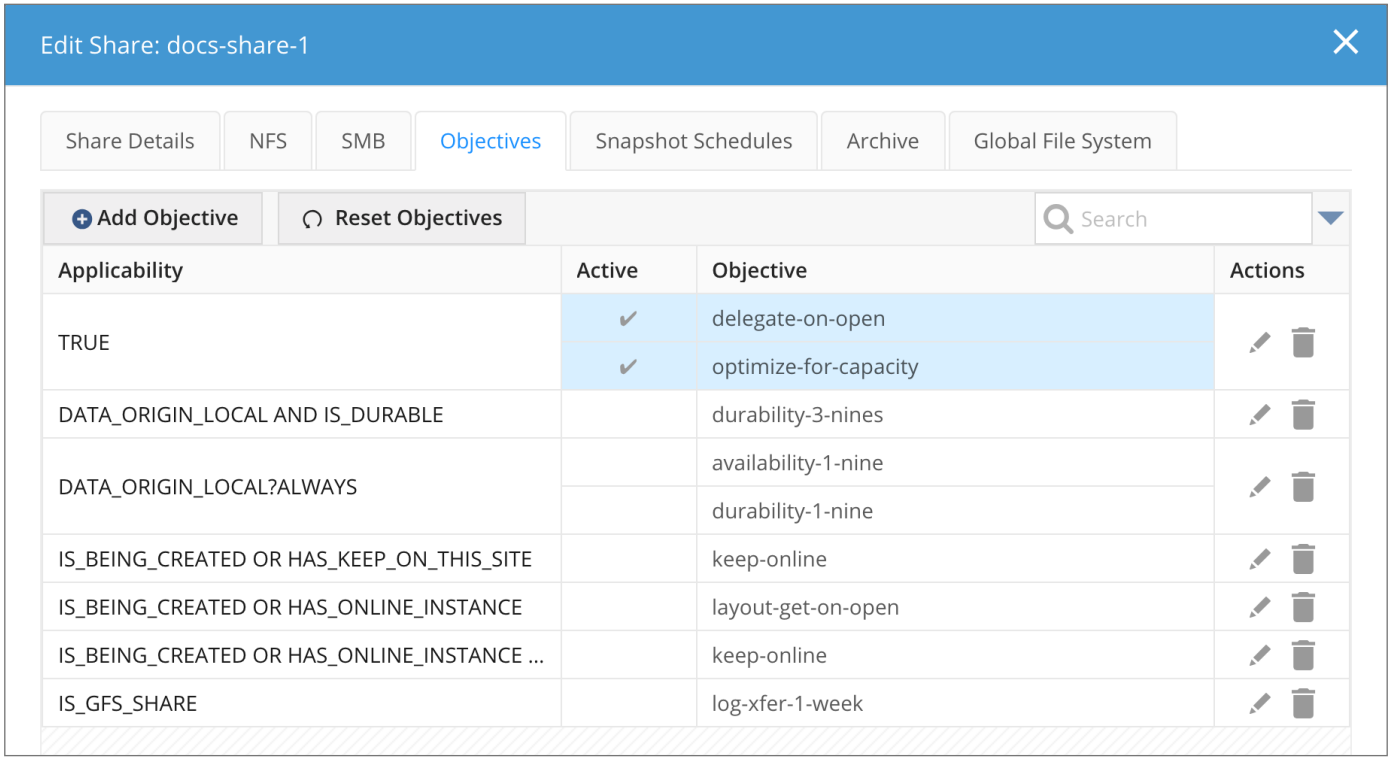


Figure 80. Objectives tab

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.
- If you have a share snapshot schedule, the share delete is not allowed.

What Happens When You Delete a Share

Deleting a share happens in two stages.

The initial delete unexports and unmounts the share and schedules it for purge. The delay before the purge runs

© 2026 - Hammerspace

— 24 hours by default — is the recovery window: until the purge runs, the deletion can still be undone. Undelete and WORM protection on files in the share do not extend this window or block the purge; see File Versioning.

Scheduling the deletion does not cancel data mobility for the share. Mobility jobs that are queued, delayed, or already running continue during the delay window.

When the purge runs, the Data Mobility Engine removes the share's queued and delayed jobs and aborts any mobility state machines still running for it.



This cleanup matters when you are also decommissioning a volume. Mobility left over from a deleted share can hold a volume at the instance-removal stage, because the volume cannot finish while instances are still moving. If a decommission appears stuck there, check whether a deleted share is still waiting for its purge.

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
```


© 2026 - Hammerspace

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
    --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.

© 2026 - Hammerspace

6. The system will automatically begin realigning data based on the additional objective(s) you selected.

8.11. 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.

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

8.11.1. 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).

© 2026 - Hammerspace

- You cannot blank out the path or remove all addresses.

8.11.2. 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.

8.11.3. 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.
- **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

8.11.4. 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.

© 2026 - Hammerspace

2. Click **Create** to finish.

8.11.5. Updating a Referral Share

Updating the Referral Path Using the Admin CLI

Command:

```
share-update --name referral-share --referral-path /new/remote/path
```

8.11.6. Deleting a Referral Share

Referral shares are deleted the same way as regular shares.

Using the Hammerspace Admin CLI

Command:

```
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.

8.11.7. 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
```


© 2026 - Hammerspace

8.12. 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.

8.12.1. 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.

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:

   ```
   2026-09-22T13-26-17-0
   2026-09-22T13-26-23-0
   2026-09-22T13-26-29-0
   ```


© 2026 - Hammerspace

```
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**.

8.12.2. 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**.

© 2026 - Hammerspace

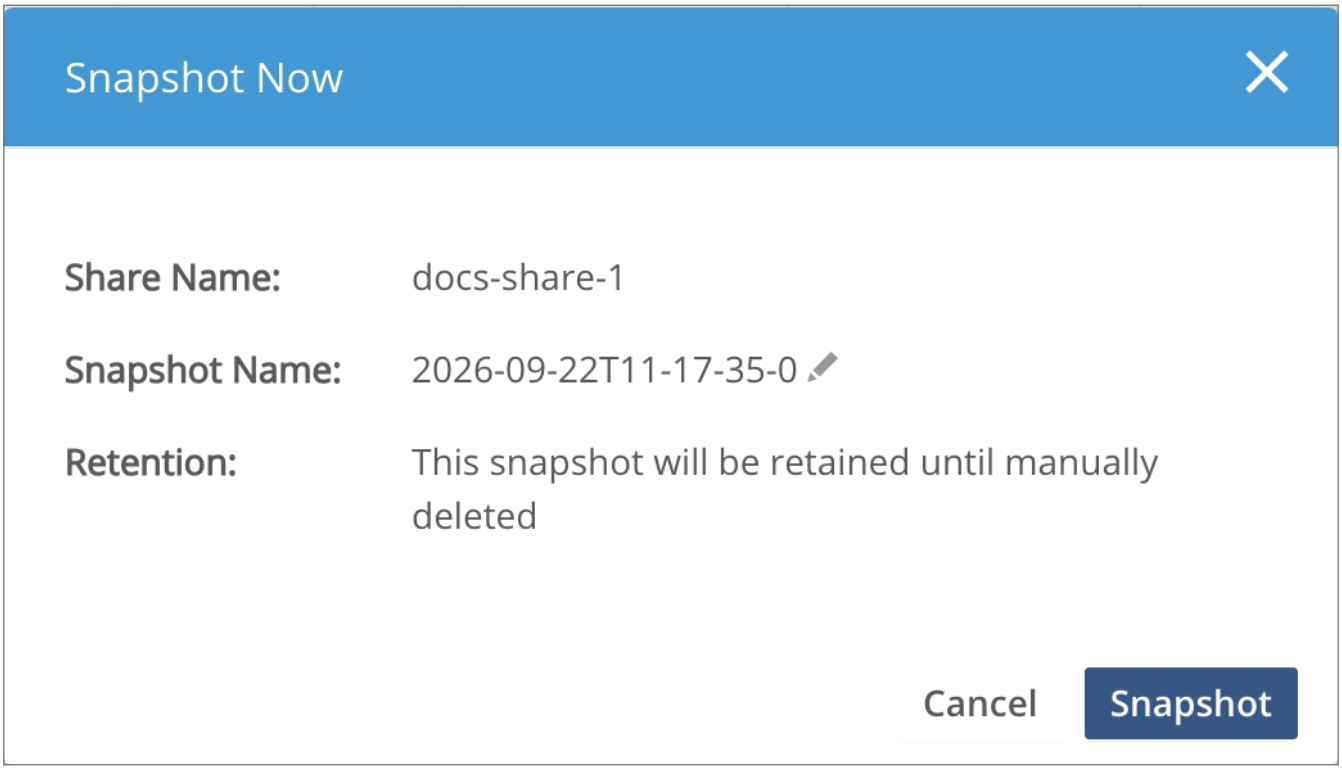


Figure 81. On-demand snapshot popup

You will see a message verifying that the snapshot was created successfully.

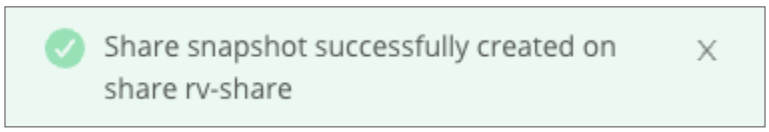


Figure 82. Snapshot successfully created

When no longer needed, the snapshot must be manually deleted.

8.12.3. 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.

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.

© 2026 - Hammerspace

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
```

8.12.4. 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.

© 2026 - Hammerspace

Do not delete the **current** file.

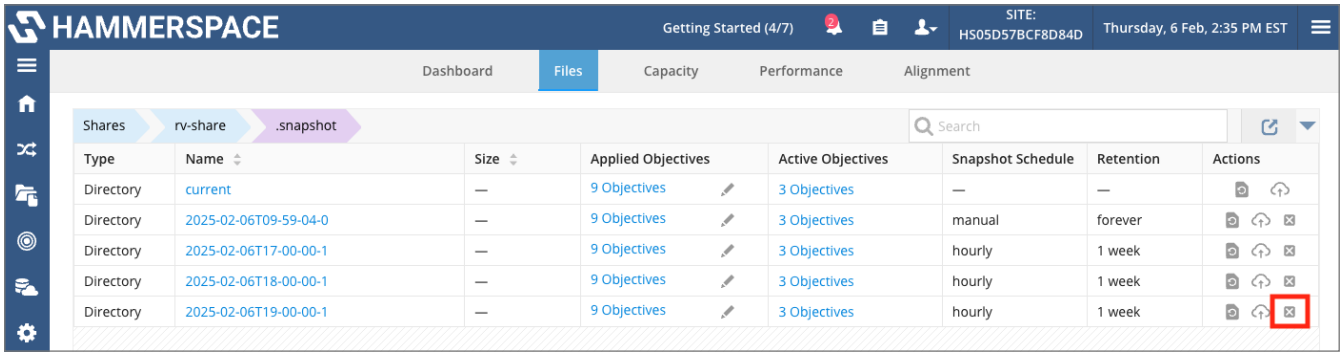


Figure 83. Location of the Delete icon

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.

8.12.5. 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
```

8.12.6. 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.

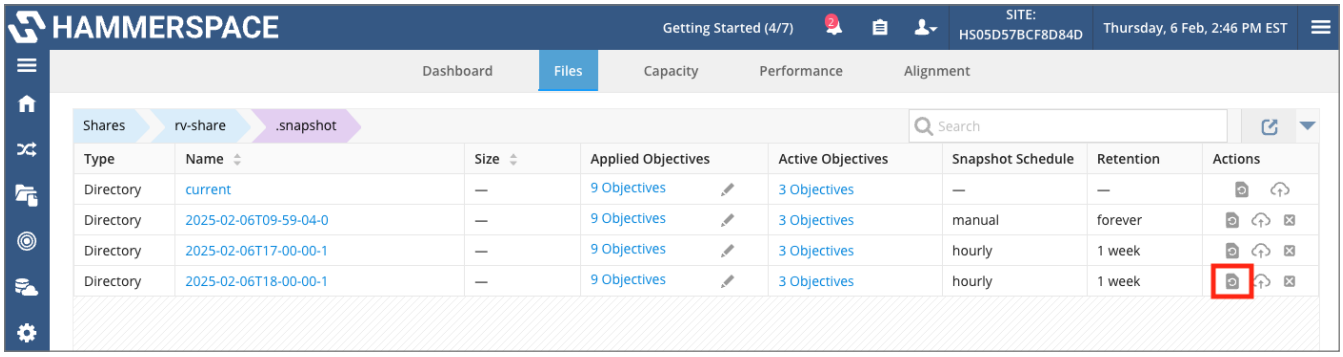


Figure 84. Location of the Restore icon

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.

© 2026 - Hammerspace

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.

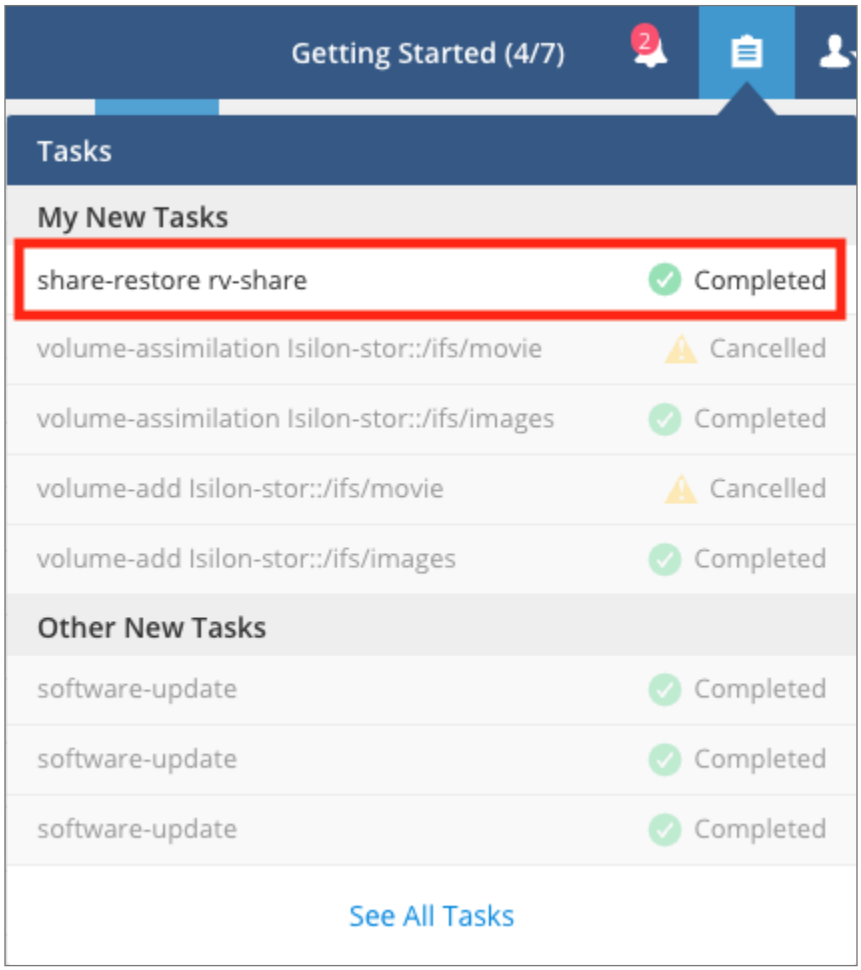


Figure 85. Viewing completed task status

8.12.7. Restoring Individual Files Using the CLI

You can restore files from a snapshot of a share.

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
```

8.12.8. 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.

© 2026 - Hammerspace

8.12.9. 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.

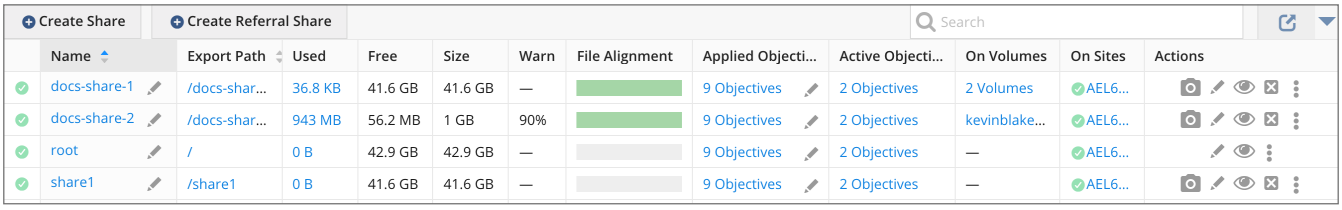


Figure 86. Editing a share

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

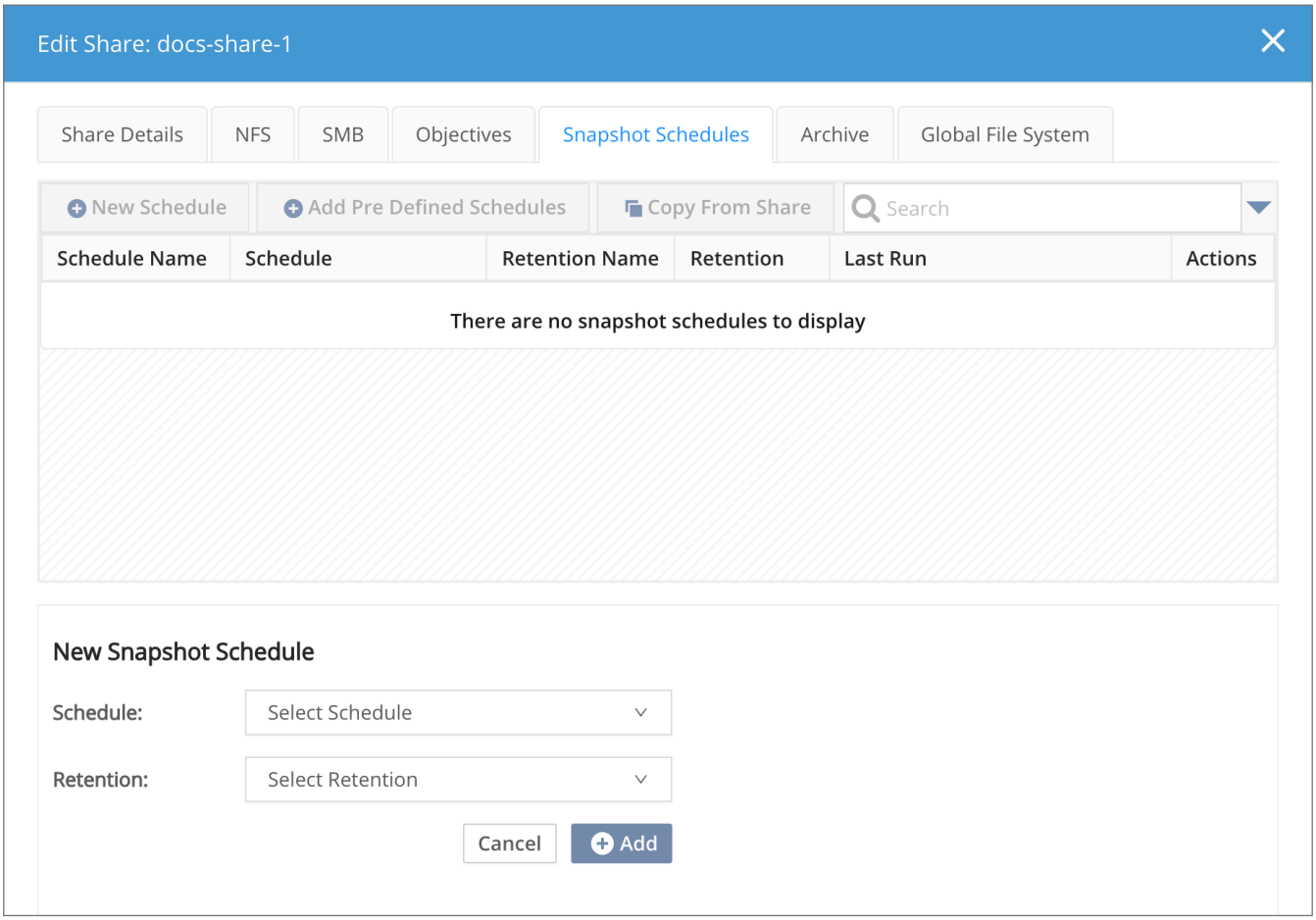


Figure 87. Adding a new snapshot schedule

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
```

8.13. 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.

8.13.1. Default Objectives

The Product applies the following default objectives to new shares. You can click on the pencil icon to modify the defaults:

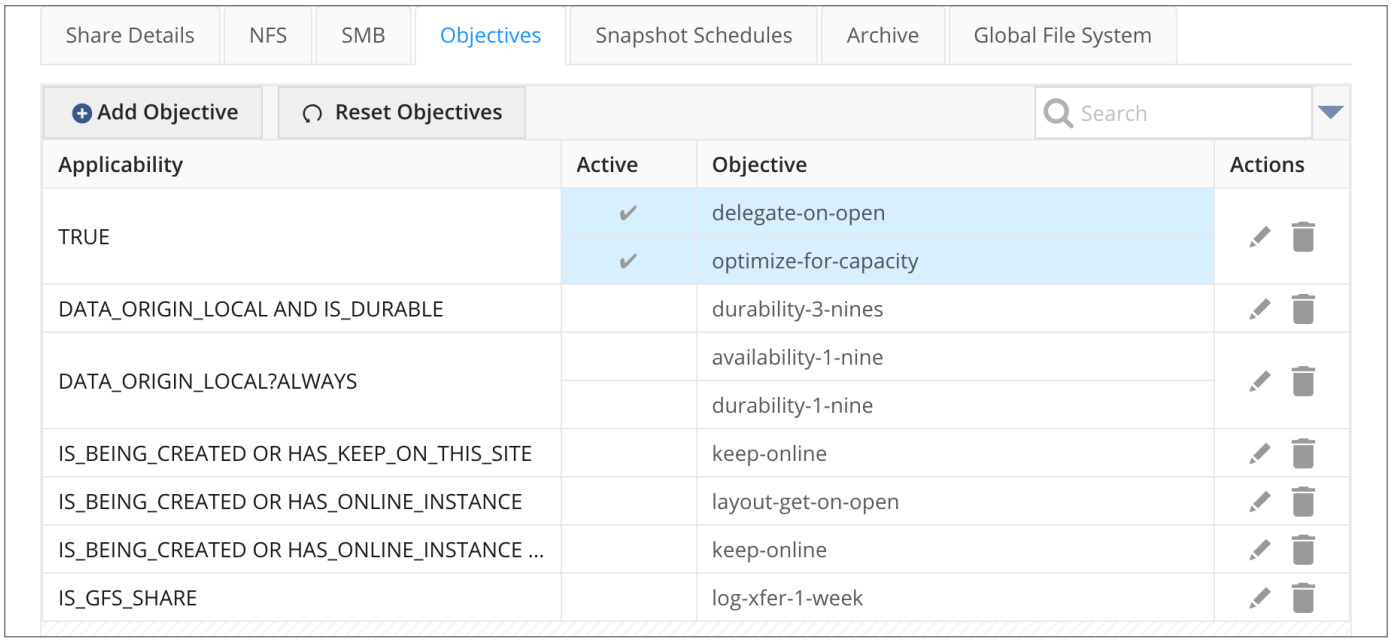


Figure 88. Default objectives following installation

8.13.2. 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 what users can see when browsing files and folders. This applies to all protocols. Requires that do-not-cache objective is also applied.
- **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 get created to meet the requested number of nines.

© 2026 - Hammerspace

- **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.
- **Place-on-object-volumes:** Place data on volumes of type "object".

© 2026 - Hammerspace

- **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.

8.13.3. 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.

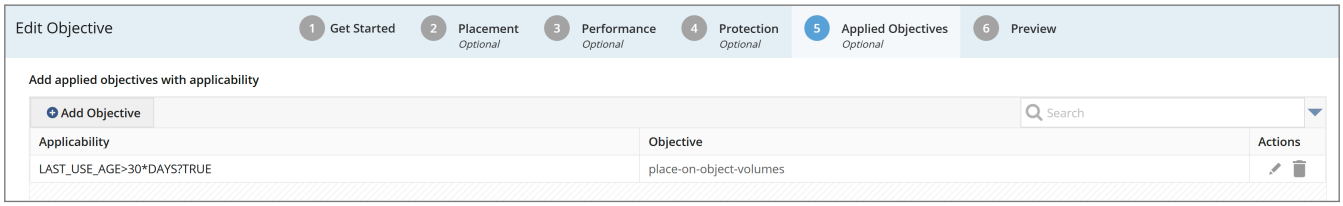


Figure 89. Editing an objective

3. Walk through the edit Objectives wizard to make changes.
4. Click **Save Objective**. The system will automatically detect the changes and align data to the updated rules.

© 2026 - Hammerspace

8.13.4. 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
- `ACTIVITY`: current average IOPS

© 2026 - Hammerspace

- `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:

Web UI operand	Hammerscript equivalent
PATH	`FNMATCH("<pattern>", PATH)`

© 2026 - Hammerspace

Web UI operand	Hammerscript equivalent
FILE_NAME	`FNMATCH("<pattern>", NAME)`
GROUP	`OWNER_GROUP`

8.13.5. 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
- Connecting Storage to a DSX Using NVMe-oF
- 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.

© 2026 - Hammerspace

4. In the **Data** tab, specify access permission and whether to assimilate the existing data on the volume. This 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

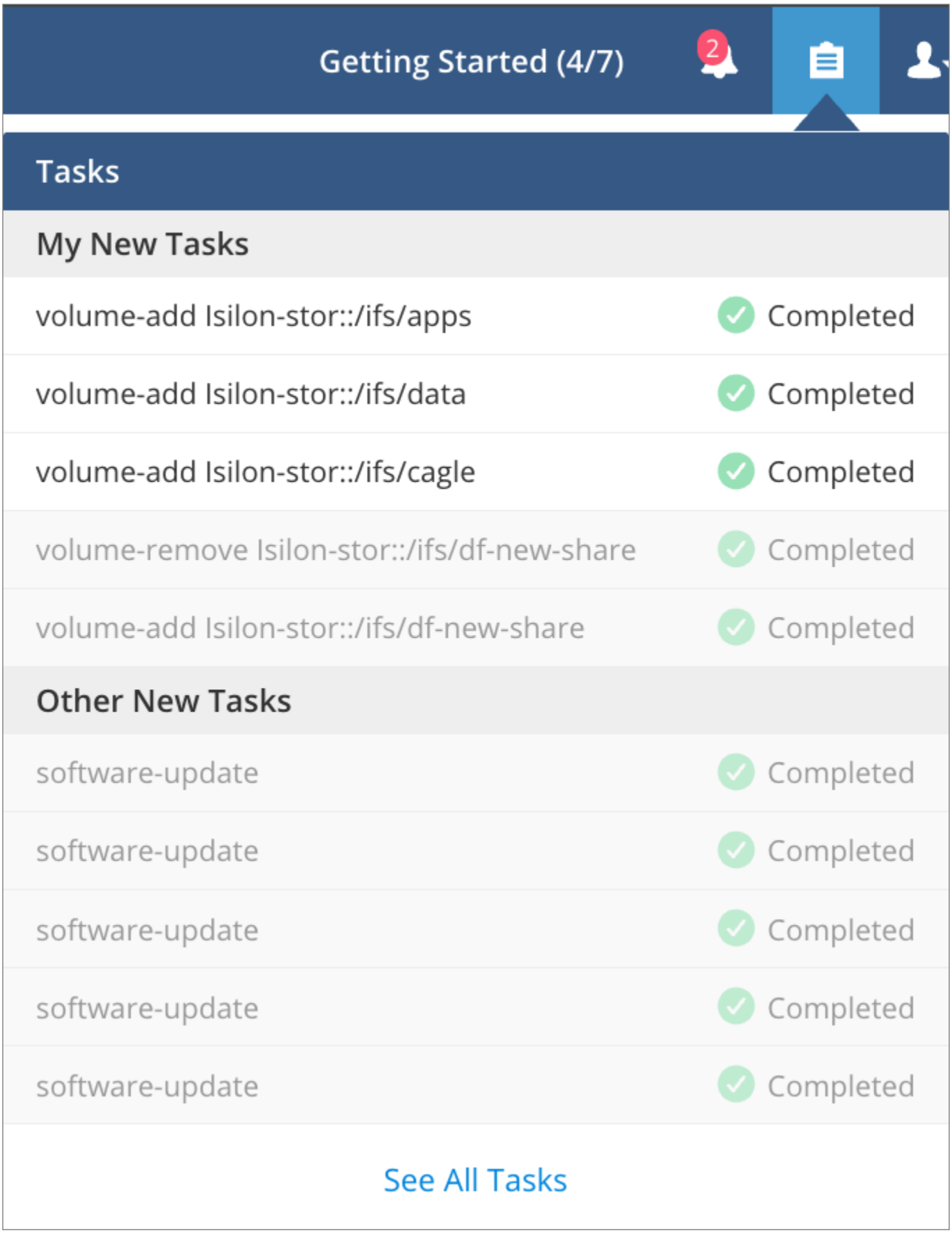


Figure 90. View operation status in the Tasks dropdown

© 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:                       Hammerspace
   Product node type:          DSX
   Internal ID:                1073741833
   ID:                         6444d356-0670-5281-a795-5845396f50f1
   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:                     0ab5f946-f754-45d2-b689-f48dab222fdf
   Name:                   /dev/dm-0
   State:                  OK

   Type:                   Hdd
   ID:                     2fe8580b-e8d0-4304-8162-61e1599e0987
   Name:                   /dev/sda
   State:                  OK

   Type:                   Hdd
   ID:                     73595b5c-5d9f-4bcd-b892-15eeb3c1a08d
   Name:                   /dev/dm-2
   State:                  OK

   Type:                   Hdd
   ID:                     99944f5b-91db-4695-8bae-160f502c6263
   Name:                   /dev/dm-1
   State:                  OK

   Type:                   Hdd
   ID:                     b052f260-b3e5-4a7a-a368-2b17eeedc7a4
   Name:                   /dev/sdc
   State:                  OK
   ```


© 2026 - Hammerspace

```
Type:                   Hdd
ID:                     fd037f43-3880-475b-a8bc-6852a0d8c817
Name:                   /dev/sdb
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:                     8a62660e-f9d9-484a-a5d7-ab23e200a0c9
```


© 2026 - Hammerspace

```
Name:                   dsx.example.com::/hsvol0
Internal ID:            19
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: 216.1MB (1%), Free: 21.2GB]
Capabilities:           Max read IOPs:            10557
                        Max write IOPs:           2506
                        Min read latency:         0.328342 milliseconds
                        Min write latency:        1.874412 milliseconds
                        Max read bandwidth:       1061642 KB/s
                        Max write bandwidth:      177658 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 13:31:42 UTC
Modified:               2026-09-22 13:31:42 UTC
Root file handle:       0100060042a939fe7ee74208b950c781077fc88c
Locations:
                        [Type: Node, ID: 6444d356-0670-5281-a795-5845396f50f1, Name: dsx.example.com]
                        [Type: Volume Group, ID: 275f5acb-f7a3-4010-9127-d5a9f12fca02, Name: all]
                        [Type: Storage Volume, ID: 8a62660e-f9d9-484a-a5d7-ab23e200a0c9, Name:
dsx.example.com::/hsvol0]
Random writes:          Enabled
```

The new volume will be viewable in the UI after a short wait as the system refreshes.

9.3. 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.

© 2026 - Hammerspace

NVMe-oF storage is supported with the arrays listed under NVMe over Fabrics in the *Hammerspace Hardware and Software Compatibility List*. Before you connect an array, read NVMe-oF Limitations and Operating Guidance.

- **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`.
- **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.

© 2026 - Hammerspace

- 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
```



NFS DIRECT and NFSD DIRECT are enabled by default on every DSX node. A DSX turns them on during node setup, and upgrading to this release turns them on as part of the upgrade. They are enabled on DSX nodes only, not on Anvil nodes.

If you need these settings changed, contact Hammerspace Support. Every software update turns them on again on every DSX node, even if they were turned off, so plan to contact Support again after each update if you need them off.

The read and write sides cannot disagree about direct I/O. Turning one of them off while the other is set to direct causes the kernel to downgrade the other rather than leave them mismatched. This behavior is deliberate.

This section includes the following subsections:

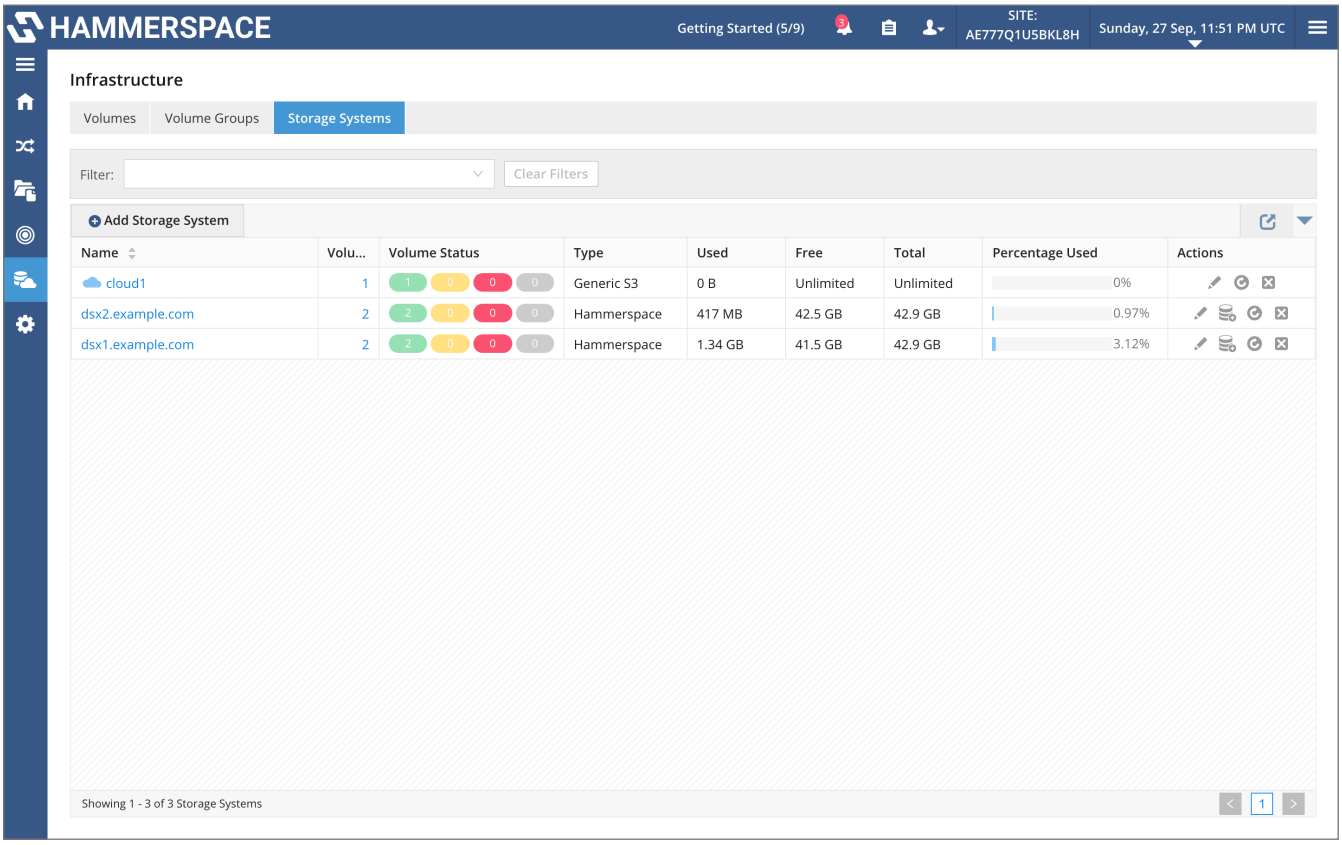
- Adding an NVMe-oF Enclosure and Subsystems
- Admin CLI Command: nvmeof-config
- Admin CLI Workflow to Connect NVMe-oF Enclosures
- Additional Admin CLI examples
- NVMe-oF Path Health and Recovery
- NVMe-oF Limitations and Operating Guidance
- Upgrading a Cluster That Uses NVMe-oF Storage

9.3.1. Adding an NVMe-oF Enclosure and Subsystems

Add an NVMe-oF enclosure to a DSX node using the Management GUI.

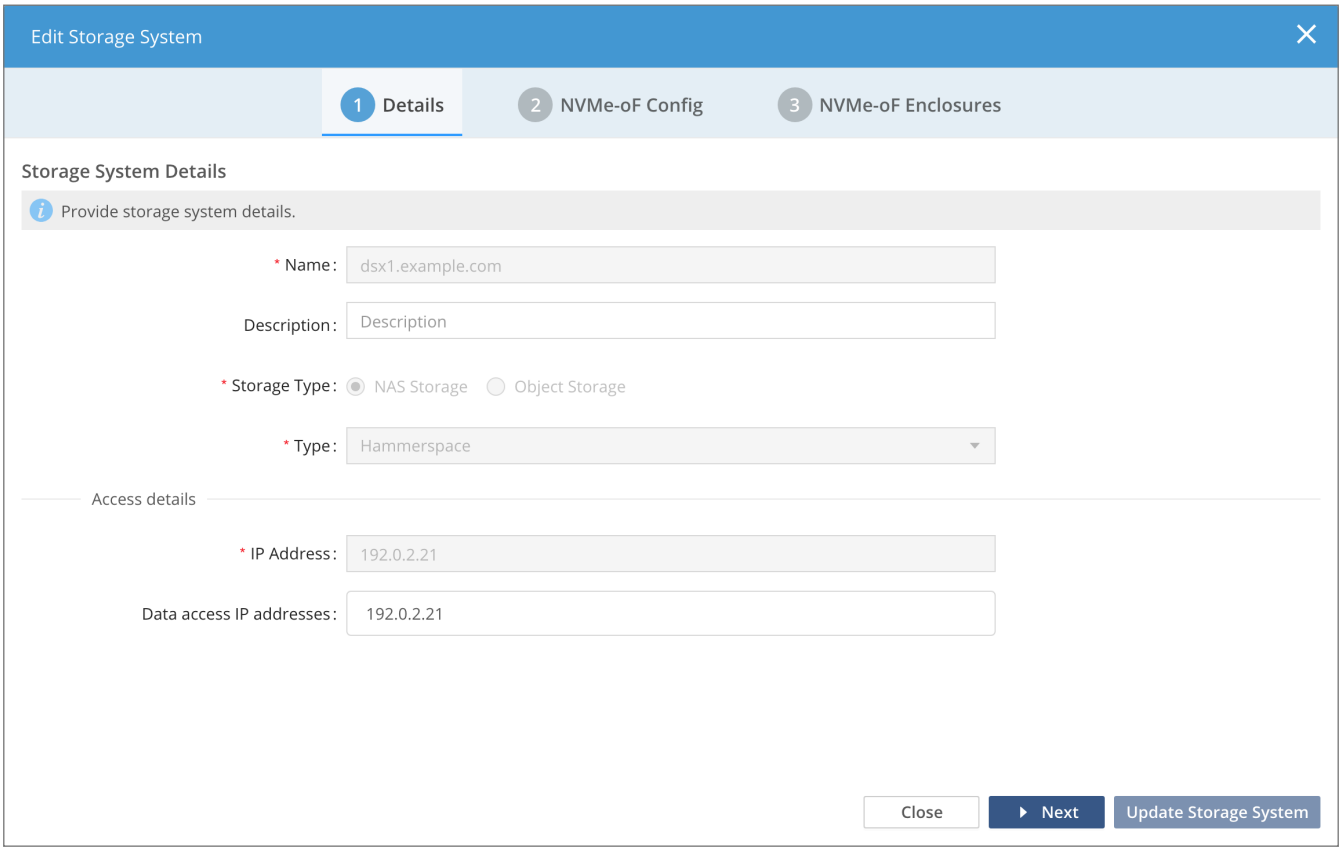
1. Log in to the Management GUI and open the **Infrastructure** page.
2. Click the **Storage Systems** tab and identify the DSX node to which you will connect the NVMe device.

© 2026 - Hammerspace



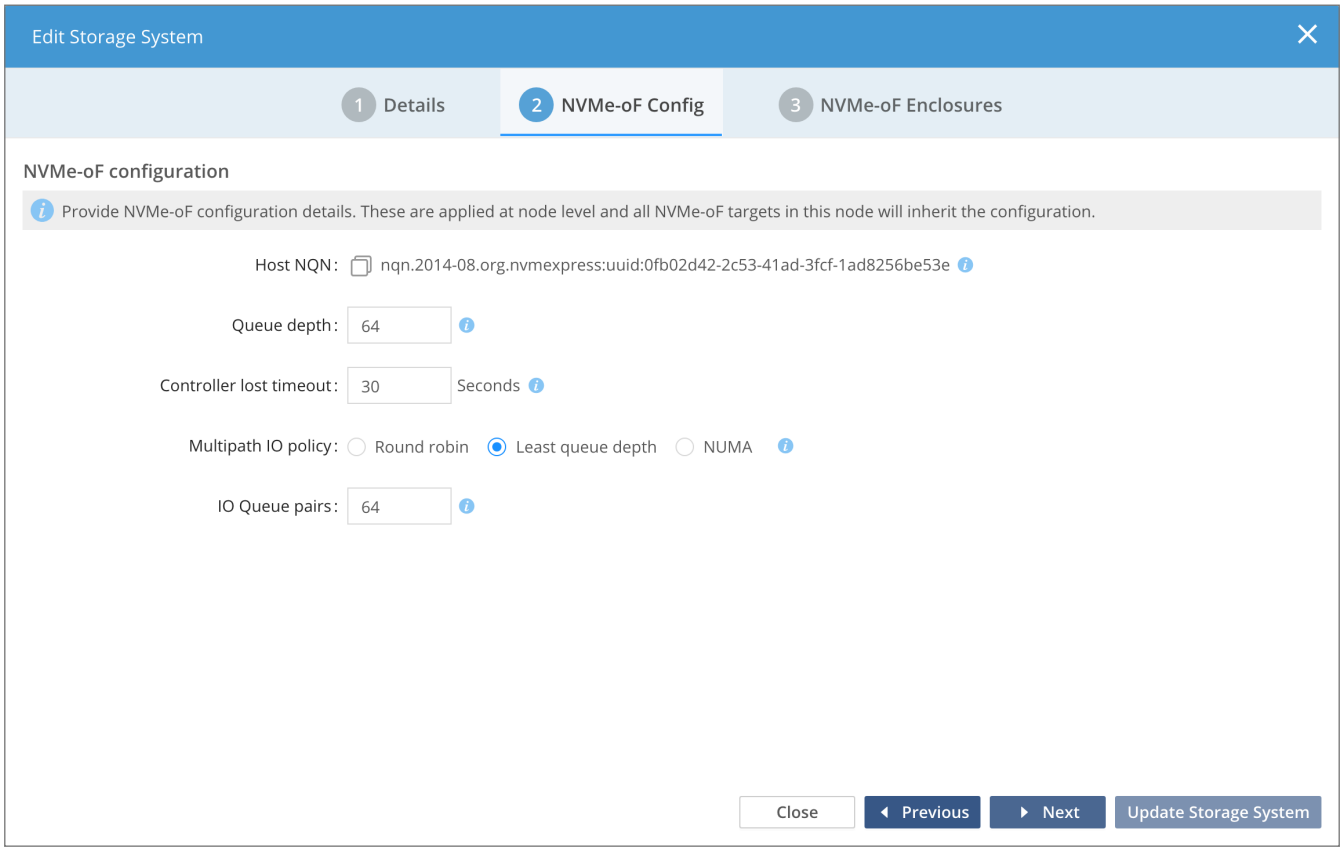
3. In the **Actions** column, click the pencil icon to edit the node. The Edit Storage System wizard opens with the **Details** tab visible. Note that the information in the Details tab was configured during the DSX node installation and cannot be changed here.

© 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

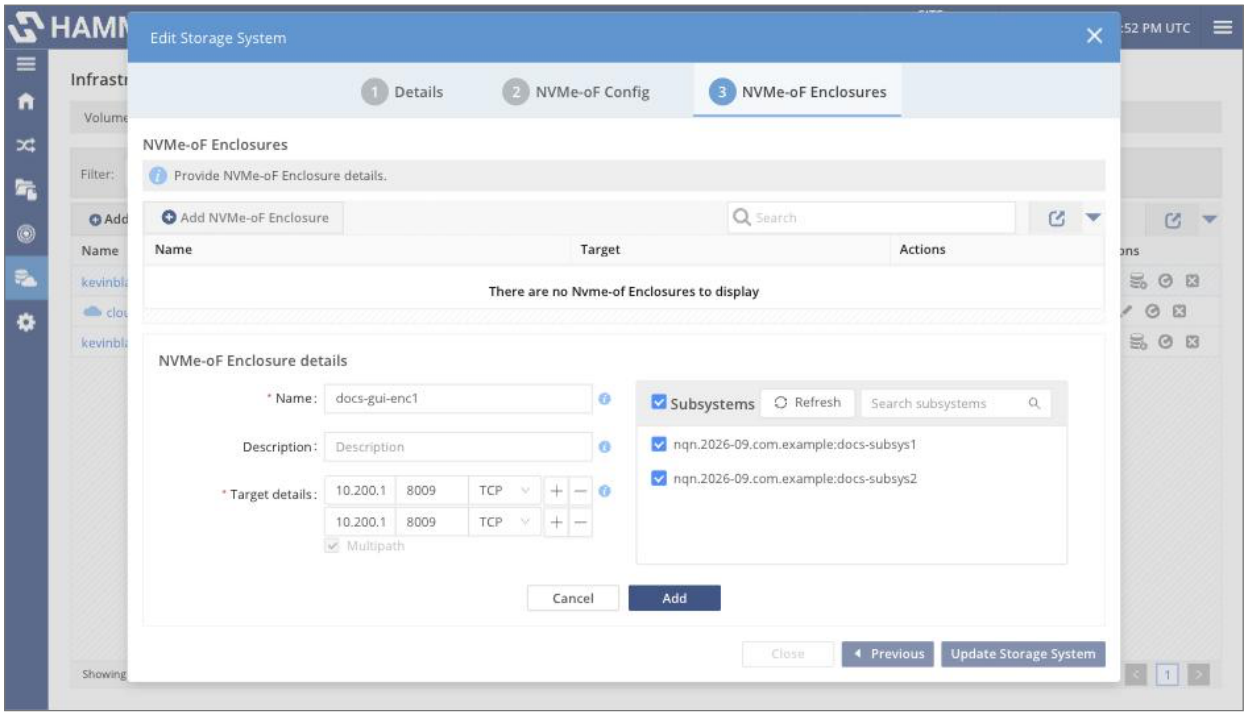


Figure 91. An enclosure with two target addresses and two subsystems selected

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.

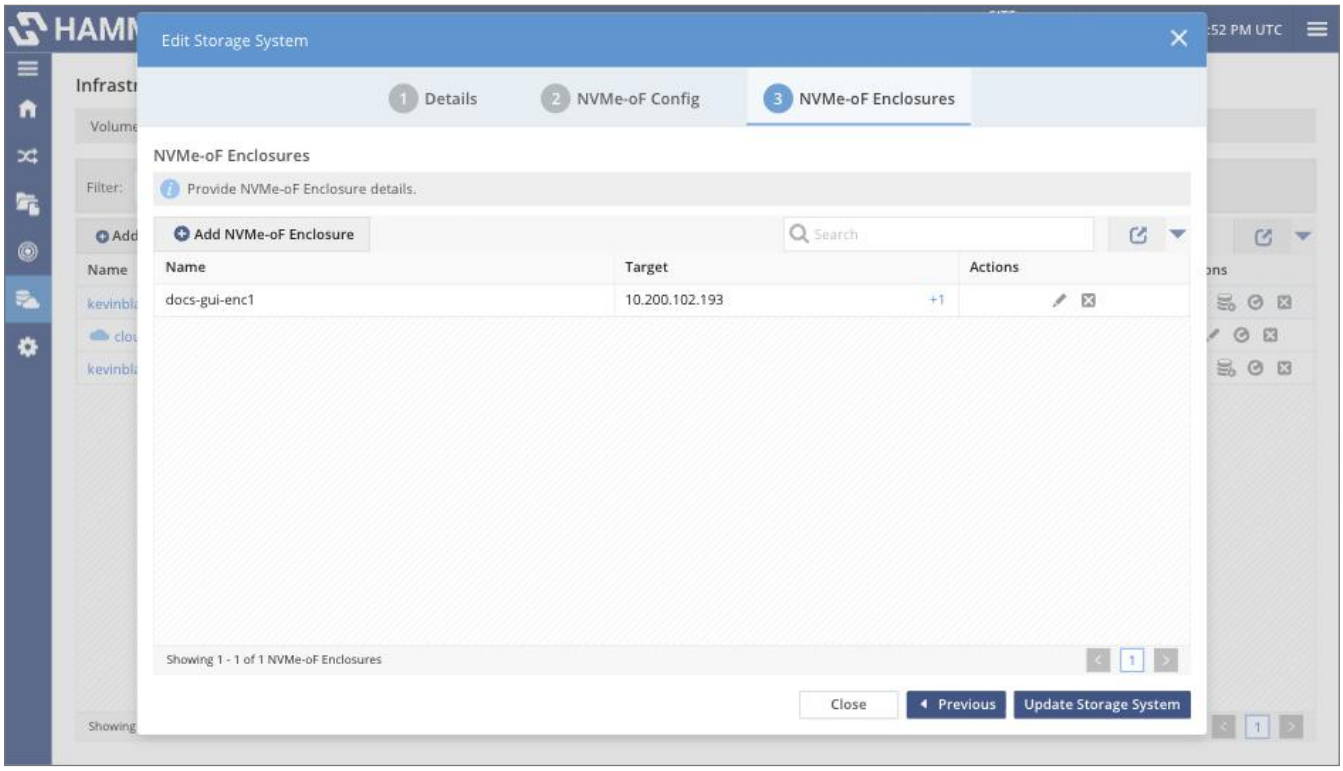
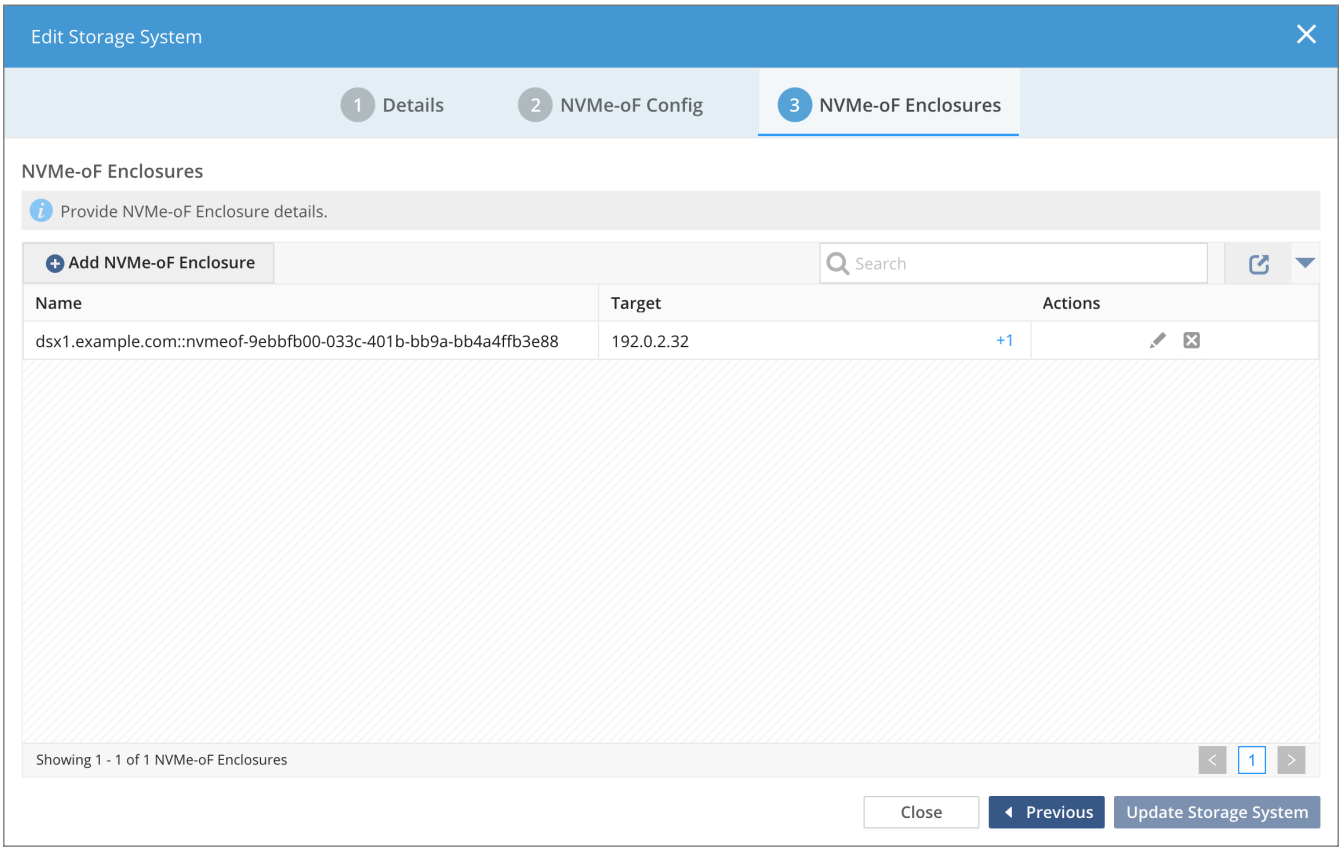


Figure 92. The new enclosure in the NVMe-oF Enclosures table

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

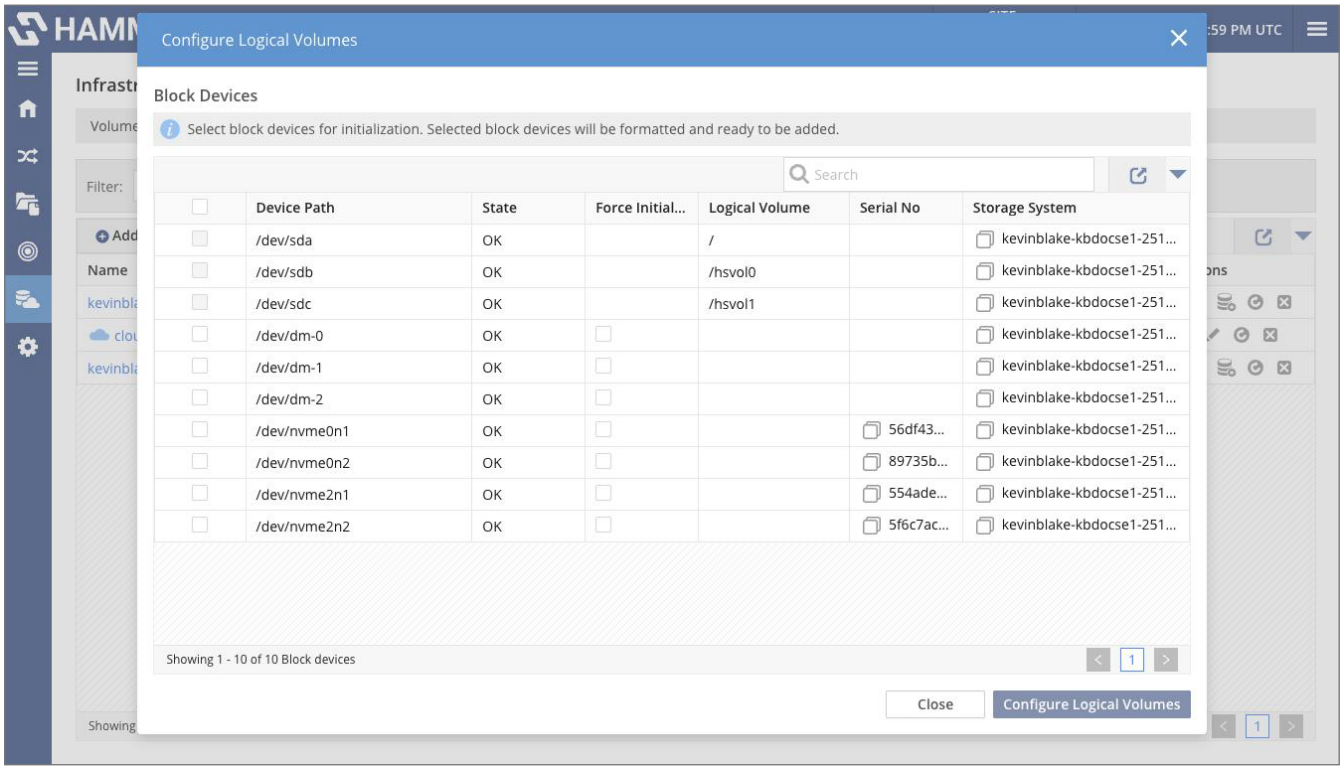


Figure 93. The Block Devices table

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.

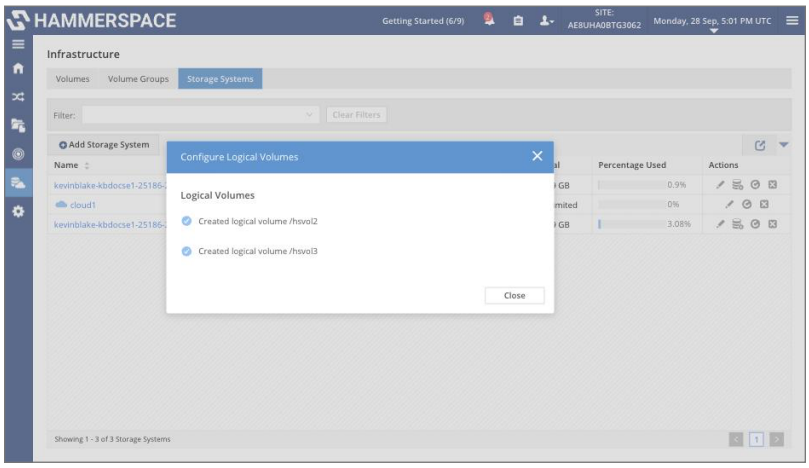


Figure 94. Logical volumes created

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.

9.3.2. 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 uses the special no-timeout value)
    --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`

9.3.3. 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:                          dsx1.example.com
   Type:                          Hammerspace
   Product node type:             DSX
   Internal ID:                   1073741839
   ID:                            7887b279-adb3-57d2-a8ca-341eca2c5ca1
   HW state:                      OK
   Node state:                    MANAGED
   Node mode:                     ONLINE
   Management IP:                 192.0.2.21/20
   SW version:                    5.3.1-1209
   S3 auth signing type:          Default
   Vendor:                        VMware, Inc.
   Product name:                  VMware Virtual Platform
   Serial number:                 1ad8256be53e
   ```


© 2026 - Hammerspace

```
Firmware version:              None
NVMe-oF host NQN:              nqn.2014-08.org.nvmexpress:uuid:0fb02d42-2c53-41ad-3fcf-1ad8256be53e
Created:                       2026-09-27 18:25:22 UTC
Modified:                      2026-09-27 23:40:40 UTC
Components:
                         Type:                 Nic
                         ID:                   892631b6-4d15-577b-96fd-c6bd2bca5b02
                         State:                OK

                         Type:                 Hdd
<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:                      5f2f3932-2c3b-45cd-a27e-f93e8c4f472d
   Name:                    dsx1.example.com::nvmeof-5f2f3932-2c3b-45cd-a27e-f93e8c4f472d
   Node:                    dsx1.example.com
   Node host NQN:           nqn.2014-08.org.nvmexpress:uuid:0fb02d42-2c53-41ad-3fcf-1ad8256be53e
   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:                 192.0.2.32
                            Port:                    8009
                            Transport:               TCP
   ```


© 2026 - Hammerspace

```
                        Address:                192.0.2.31
                        Port:                   8009
                        Transport:              TCP
Subsystems:
                        Type:                   Nvme-of Subsystem
                        ID:                     7c4b48fa-ce56-5ff2-936f-e68170bb6966
                        Subsystem NQN:          nqn.2026-09.com.example:docs-subsys2
                        Namespaces:
                                                [Namespace: nvme0n1, Namespace ID: 2, Size: 4.2GB,
Sector size: 4096, Model: Docs Lab NVMe Target, Serial: DOCSLAB00000002, Namespace unique ID:
bef19a80-60dd-b2bd-30a3-4965b79d3e20, Connected: true]
                                                [Namespace: nvme0n2, Namespace ID: 1, 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:                     9d588e3f-d951-5d41-b918-d2e557bf8edc
                        Subsystem NQN:          nqn.2026-09.com.example:docs-subsys1
                        Namespaces:
                                                [Namespace: nvme2n1, 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: nvme2n2, 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]
```

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
   ```


© 2026 - Hammerspace

```
ID:                     5f2f3932-2c3b-45cd-a27e-f93e8c4f472d
Name:                   dsx1.example.com::nvmeof-5f2f3932-2c3b-45cd-a27e-f93e8c4f472d
Node:                   dsx1.example.com
Node host NQN:          nqn.2014-08.org.nvmexpress:uuid:0fb02d42-2c53-41ad-3fcf-1ad8256be53e
```

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:                   Nvme-of Drive
   ID:                     77620eed-1e30-5c4f-891a-aa4d3dc9718d
   Name:                   /dev/nvme2n2
   State:                  OK
   Capacity:               4.2GB
   Serial number:          77620eed-1e30-5c4f-891a-aa4d3dc9718d

   Type:                   Nvme-of Drive
   ID:                     a3874636-842f-5503-8f81-a6bfdb7f6458
   Name:                   /dev/nvme2n1
   State:                  OK
   Capacity:               4.2GB
   Serial number:          a3874636-842f-5503-8f81-a6bfdb7f6458

   Type:                   Nvme-of Drive
   ID:                     23770f46-3435-554d-9288-fbfeaeb498c6
   Name:                   /dev/nvme0n1
   State:                  OK
   Capacity:               4.2GB
   Serial number:          23770f46-3435-554d-9288-fbfeaeb498c6
   ```


© 2026 - Hammerspace

```
Type:                   Nvme-of Drive
ID:                     fb91b1e9-8a1b-5eb8-9f44-7c7908708621
Name:                   /dev/nvme0n2
State:                  OK
Capacity:               4.2GB
Serial number:          fb91b1e9-8a1b-5eb8-9f44-7c7908708621

Type:                   Hdd
ID:                     2dcbd93d-9bb6-4263-af16-436a3fcc16d7
Name:                   /dev/dm-2
State:                  OK

Type:                   Hdd
ID:                     1c92fad9-da28-4e99-b77f-e8dd28824329
Name:                   /dev/dm-1
State:                  OK

Type:                   Hdd
ID:                     aac8ac10-17fa-4cda-988b-23f65485801f
Name:                   /dev/dm-0
State:                  OK

Type:                   Hdd
ID:                     fd780ea7-e109-4184-947a-bd24b3943920
Name:                   /dev/sdc
State:                  OK

Type:                   Hdd
ID:                     a8fa53db-dd9a-4e9e-9ae4-6125f4e4b530
Name:                   /dev/sdb
State:                  OK

Type:                   Hdd
ID:                     a10ebcbf-2130-44ea-a8d8-c231a801a6bb
Name:                   /dev/sda
State:                  OK
```

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:

   ```
   logical-volume-create --node-name <dsx-node-name> --device-path <device-path>
   ```

   Expected output:

© 2026 - Hammerspace

```
Type:                   Logical Volume
ID:                     2b53f6f1-1d21-49f3-aede-637f8ddd806c
Name:                   /hsvol2
Exported path:          /hsvol2
Addresses:
                        [IP: 192.0.2.21/20, Port: 3049, NetId: tcp, NodeNum: 0]
Node:                   dsx1.example.com
Usage:                  DS
Reserved:               false
Filesystem type:        XFS
Capacity:               [Total: 4.2GB]
Hardware:
                        [Type: Nvme-of Drive, ID: 77620eed-1e30-5c4f-891a-aa4d3dc9718d, Name:
/dev/nvme2n2, 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
   ```

9.3.4. 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
```

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`:

© 2026 - Hammerspace

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:                     5f2f3932-2c3b-45cd-a27e-f93e8c4f472d
Name:                   dsx1.example.com::nvmeof-5f2f3932-2c3b-45cd-a27e-f93e8c4f472d
Node:                   dsx1.example.com
Node host NQN:          nqn.2014-08.org.nvmexpress:uuid:0fb02d42-2c53-41ad-3fcf-1ad8256be53e
```

```
total 2
Type:                   Nvme-of Enclosure
ID:                     9ebbfb00-033c-401b-bb9a-bb4a4ffb3e88
Name:                   dsx1.example.com::nvmeof-9ebbfb00-033c-401b-bb9a-bb4a4ffb3e88
Node:                   dsx1.example.com
Node host NQN:          nqn.2014-08.org.nvmexpress:uuid:0fb02d42-2c53-41ad-3fcf-1ad8256be53e
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:                 192.0.2.32
                        Port:                    8009
                        Transport:               TCP

                        Address:                 192.0.2.31
                        Port:                    8009
                        Transport:               TCP
Subsystems:
                        Type:                    Nvme-of Subsystem
                        ID:                      9d588e3f-d951-5d41-b918-d2e557bf8edc
                        Subsystem NQN:           nqn.2026-09.com.example:docs-subsys1
                        Namespaces:
                                                 [Namespace: nvme0n1, 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: nvme0n2, 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]
```


© 2026 - Hammerspace

```
                        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:                     7c4b48fa-ce56-5ff2-936f-e68170bb6966
                        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]

Type:                   Nvme-of Enclosure
ID:                     7d53462a-d5d2-4b53-9264-e9883b03a8dd
Name:                   dsx1.example.com::nvmeof-7d53462a-d5d2-4b53-9264-e9883b03a8dd
Node:                   dsx1.example.com
Node host NQN:          nqn.2014-08.org.nvmexpress:uuid:0fb02d42-2c53-41ad-3fcf-1ad8256be53e
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:                192.0.2.32
                        Port:                   8009
                        Transport:              TCP

                        Address:                192.0.2.31
                        Port:                   8009
                        Transport:              TCP
Subsystems:
                        Type:                   Nvme-of Subsystem
                        ID:                     9d588e3f-d951-5d41-b918-d2e557bf8edc
                        Subsystem NQN:          nqn.2026-09.com.example:docs-subsys1
                        Namespaces:
                                                [Namespace: nvme0n1, 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: nvme0n2, 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: 192.0.2.31, Port: 8009, Transport: TCP,
State: CONNECTED]
                                                [Address: 192.0.2.32, Port: 8009, Transport: TCP,
State: CONNECTED]

                        Type:                   Nvme-of Subsystem
                        ID:                     7c4b48fa-ce56-5ff2-936f-e68170bb6966
                        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: 192.0.2.32, Port: 8009, Transport: TCP,
State: DISCOVERED]
                                                [Address: 192.0.2.31, 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 possible only while none of their namespaces has a logical volume. Give every address when you add the enclosure instead.

© 2026 - Hammerspace

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.

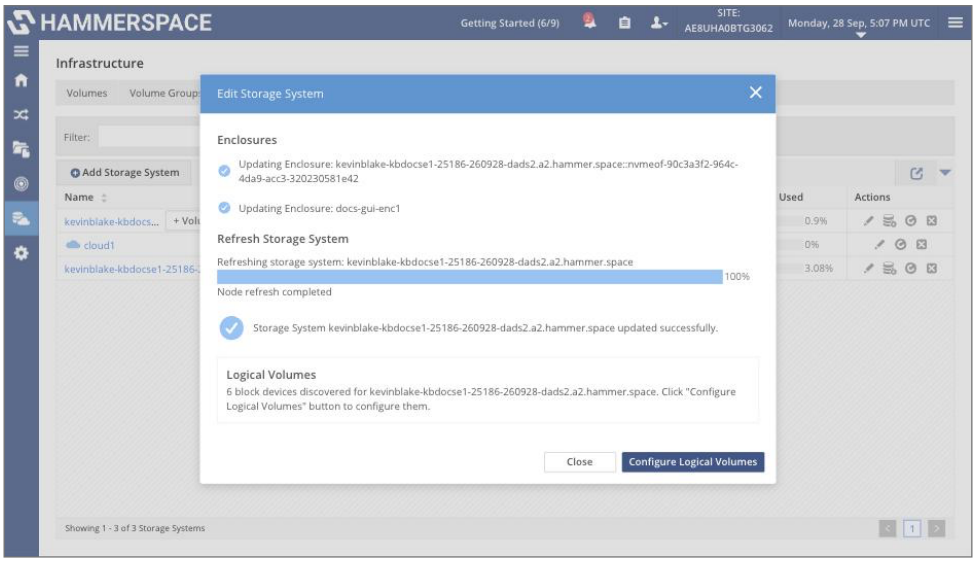


Figure 95. A queue-depth change from the GUI updates both enclosures on the node

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 16. NVMe-oF connection settings

Option	Values
`--io-policy`	`QUEUE_DEPTH` (the default), `ROUND_ROBIN`, or `NUMA`

© 2026 - Hammerspace

Option	Values
`--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.
`--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) or `RDMA`.

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.

Volumes stay mounted through the reconnect. Make the change in a maintenance window anyway.

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
```

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

© 2026 - Hammerspace

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:

```
success
```

While a logical volume exists on one of the enclosure's namespaces, the command fails.

© 2026 - Hammerspace

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:                     2b53f6f1-1d21-49f3-aede-637f8ddd806c
Name:                   /hsvol2
Exported path:          /hsvol2
Addresses:
                        [IP: 192.0.2.21/20, Port: 3049, NetId: tcp, NodeNum: 0]
Node:                   dsx1.example.com
Usage:                  DS
Reserved:               false
Filesystem type:        XFS
Capacity:               [Total: 4.2GB]
Hardware:
                        [Type: Nvme-of Drive, ID: 77620eed-1e30-5c4f-891a-aa4d3dc9718d, Name:
/dev/nvme2n2, State: OK]
```


© 2026 - Hammerspace

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.

9.3.5. NVMe-oF Path Health and Recovery

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 17. 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>
```

The source of both events is the NVMe-oF subsystem, identified by its NQN. In the **Events** list, the **Source type** column shows **Nvme-of Subsystem**. If Hammerspace cannot match the path to a subsystem that it manages, the source is the DSX node that lost the path.

Expected output of `event-list --type NVMEOF_PATH_DEGRADED` with one path of a two-path target down:

© 2026 - Hammerspace

```
total 2
ID:                     3c5265f0-bad1-11f1-8660-005056adc30c
Created:                2026-09-28 00:12:04 UTC
Modified:               2026-09-28 00:12:04 UTC
Type:                   NVMEOF_PATH_DEGRADED
Description:            NVMe-oF path degraded (controller stuck connecting): nqn=nqn.2026-
09.com.example:docs-subsys2, traddr=192.0.2.32, trsvcid=8009, transport=tcp, node=dsx1.example.com
Severity:               WARNING
Source type:            Nvme-of Subsystem
Source ID:              7c4b48fa-ce56-5ff2-936f-e68170bb6966
Source name:            nqn.2026-09.com.example:docs-subsys2
Cleared:                false

ID:                     3c7f2d10-bad1-11f1-8660-005056adc30c
Created:                2026-09-28 00:12:05 UTC
Modified:               2026-09-28 00:12:05 UTC
Type:                   NVMEOF_PATH_DEGRADED
Description:            NVMe-oF path degraded (controller stuck connecting): nqn=nqn.2026-
09.com.example:docs-subsys1, traddr=192.0.2.32, trsvcid=8009, transport=tcp, node=dsx1.example.com
Severity:               WARNING
Source type:            Nvme-of Subsystem
Source ID:              9d588e3f-d951-5d41-b918-d2e557bf8edc
Source name:            nqn.2026-09.com.example:docs-subsys1
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.

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.

© 2026 - Hammerspace

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 2
Type:                   Nvme-of Enclosure
ID:                     9ebbfb00-033c-401b-bb9a-bb4a4ffb3e88
Name:                   dsx1.example.com::nvmeof-9ebbfb00-033c-401b-bb9a-bb4a4ffb3e88
Node:                   dsx1.example.com
Node host NQN:          nqn.2014-08.org.nvmexpress:uuid:0fb02d42-2c53-41ad-3fcf-1ad8256be53e
NVMe-oF config:         Queue depth:            64
                        IO queue pairs:         64
                        I/O policy:             QUEUE_DEPTH
                        Transport:              TCP
```


© 2026 - Hammerspace

```
                        Target port:            8009
                        Controller loss timeout: -1
Discovery targets:
                        Address:                192.0.2.32
                        Port:                   8009
                        Transport:              TCP

                        Address:                192.0.2.31
                        Port:                   8009
                        Transport:              TCP
Subsystems:
                        Type:                   Nvme-of Subsystem
                        ID:                     9d588e3f-d951-5d41-b918-d2e557bf8edc
                        Subsystem NQN:          nqn.2026-09.com.example:docs-subsys1
                        Namespaces:
                                                [Namespace: nvme0n1, 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: nvme0n2, 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]

                        Type:                   Nvme-of Subsystem
                        ID:                     7c4b48fa-ce56-5ff2-936f-e68170bb6966
                        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]

Type:                   Nvme-of Enclosure
ID:                     7d53462a-d5d2-4b53-9264-e9883b03a8dd
Name:                   dsx1.example.com::nvmeof-7d53462a-d5d2-4b53-9264-e9883b03a8dd
Node:                   dsx1.example.com
Node host NQN:          nqn.2014-08.org.nvmexpress:uuid:0fb02d42-2c53-41ad-3fcf-1ad8256be53e
NVMe-oF config:         Queue depth:            64
                        IO queue pairs:         64
```


© 2026 - Hammerspace

```
                        I/O policy:              QUEUE_DEPTH
                        Transport:               TCP
                        Target port:             8009
                        Controller loss timeout: -1
Discovery targets:
                        Address:                 192.0.2.32
                        Port:                    8009
                        Transport:               TCP

                        Address:                 192.0.2.31
                        Port:                    8009
                        Transport:               TCP
Subsystems:
                        Type:                    Nvme-of Subsystem
                        ID:                      9d588e3f-d951-5d41-b918-d2e557bf8edc
                        Subsystem NQN:           nqn.2026-09.com.example:docs-subsys1
                        Namespaces:
                                                 [Namespace: nvme0n1, 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: nvme0n2, 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: 192.0.2.31, Port: 8009, Transport: TCP,
State: CONNECTED]
                                                 [Address: 192.0.2.32, Port: 8009, Transport: TCP,
State: CONNECTED]

                        Type:                    Nvme-of Subsystem
                        ID:                      7c4b48fa-ce56-5ff2-936f-e68170bb6966
                        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: 192.0.2.32, Port: 8009, Transport: TCP,
State: DISCOVERED]
                                                 [Address: 192.0.2.31, Port: 8009, Transport: TCP,
State: DISCOVERED]
```

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

© 2026 - Hammerspace

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 this release use `-1` by default. Enclosures that were added before the cluster was upgraded keep the value they had, so check them after upgrading.

Why -1 Is Recommended

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 it and remounts the file system; the volume returns to **Up** within a few minutes without any action from you.

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 2
Type:                   Nvme-of Enclosure
ID:                     9ebbfb00-033c-401b-bb9a-bb4a4ffb3e88
Name:                   dsx1.example.com::nvmeof-9ebbfb00-033c-401b-bb9a-bb4a4ffb3e88
Node:                   dsx1.example.com
Node host NQN:          nqn.2014-08.org.nvmexpress:uuid:0fb02d42-2c53-41ad-3fcf-1ad8256be53e
NVMe-oF config:         Queue depth:             64
                        IO queue pairs:          64
                        I/O policy:              QUEUE_DEPTH
                        Transport:               TCP
                        Target port:             8009
                        Controller loss timeout: -1
<Additional output removed for brevity>
```


© 2026 - Hammerspace

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.

Volumes stay mounted through the reconnect. Make the change in a maintenance window anyway.

9.3.6. NVMe-oF Limitations and Operating Guidance

Presenting Storage to DSX Nodes

- Present each namespace to exactly one DSX node. Use the array's host access configuration, with each DSX node's host NQN, so that no other host can see the namespace. Hammerspace lets a second DSX node connect a subsystem that another node already uses, and refuses only the creation of a logical volume on it (`One or more of the specified NVMe-oF subsystems is already connected to another node`); it cannot stop a host outside Hammerspace from writing to a namespace. Two hosts writing to one namespace corrupt its file system.
- Give the enclosure every address that the array presents its subsystems on when you add it. Hammerspace connects a subsystem to the addresses that are known when the subsystem is connected. An address added to the enclosure later is listed, but the connected subsystems are not connected to it, even though the listing shows the new address as `CONNECTED`. To add a path to a subsystem that is already connected, disconnect and reconnect the subsystem, which you can do only while none of its namespaces has a logical volume.
- Do not present a clone or snapshot of a namespace to the same DSX node as the original. Namespaces that carry the same file system UUID are not supported, and Hammerspace may be unable to mount a volume on either of them.
- 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.

© 2026 - Hammerspace

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 other than the I/O policy reconnects the enclosure's controllers. See Controller Loss Timeout.

9.3.7. 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*.

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 5.3.1 or later. These builds include fixes for clusters that use NVMe-oF storage.
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.

© 2026 - Hammerspace

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.

   Do not re-create the logical volume, and do not use **Force Initialize** in the **Configure Logical Volumes** wizard or `logical-volume-create --force`. Both format the device and erase the data on it. The data on the array is usually intact, and Hammerspace Support can restore the volume without reformatting it.

3. If you upgraded from 5.2, clear any NVMe-oF path event that was raised before the upgrade. Such an event does not clear automatically after the upgrade, because 5.2 attributed it to the DSX node and this release attributes it to the NVMe-oF subsystem.
   a. First make sure the paths are healthy (step 1).
   b. List the path events that have not been cleared:

      Command:

      ```
      event-list --type NVMEOF_PATH_LOST --uncleared
      ```

      Repeat the command with `--type NVMEOF_PATH_DEGRADED`.

   c. For each event whose source is a DSX node, not an NVMe-oF subsystem, clear it by its ID:

      Command:

      ```
      event-update --clear --id <event-id>
      ```

      In the listing, an event raised before the upgrade shows `Source type: Node` and the DSX node's name as its source; an event raised after the upgrade shows `Source type: Nvme-of Subsystem` and the subsystem's NQN.

1. Check the controller loss timeout of each enclosure. Enclosures keep the value they had before the upgrade. See Controller Loss Timeout.

   The upgrade also turns on NFS DIRECT and NFSD DIRECT on every DSX node. See Connecting Storage to a DSX Using NVMe-oF.

9.4. Support for AWS S3 Glacier Instant Retrieval (GIR)

Starting in Hammerspace 5.3, administrators can designate an Object Storage Volume (OSV) as AWS S3 Glacier Instant Retrieval (GIR) class. GIR is an AWS archive storage tier that costs roughly 5.75 times less per GB per month than S3 Standard, while still providing millisecond read latency — unlike deeper archive tiers such as Glacier Flexible Retrieval or Glacier Deep Archive, which require hours of restore time before data is accessible.

© 2026 - Hammerspace

GIR enables organizations to significantly reduce cloud storage spend on data that must be retained for compliance, reference, or archival purposes — data that must be immediately accessible upon request but is rarely or never part of active workflows.

Storage system support: Amazon S3 is the only object storage type that supports GIR in this release. The `--storage-class` CLI parameter and the **Archive Type** field in the GUI are not available for Azure, GCP, or Hammerspace S3 storage nodes.

This section includes the following subsections:

- When to Use GIR
- Cost Considerations
- Cost Observability and Reporting Limitations
- How Hammerspace Manages GIR
- Prerequisites
- Adding an AWS S3 Storage System
- Adding a GIR Object Storage Volume
- The archive-object-volumes Volume Group
- Placing Data on GIR — Configuring Share Archive Rules
- Mobility Behavior
- Replication Behavior
- Shared and mixed-version Configurations
- Verifying Your GIR Configuration
- Limitations and Constraints
- Upgrade Considerations
- Troubleshooting
- CLI Reference: object-volume-add
- FAQ

9.4.1. When to Use GIR

GIR is appropriate when you need to:

- Retain large volumes of data for compliance, archival, or reference workflows
- Keep that data immediately accessible (millisecond reads) without restoring from a deeper archive tier
- Reduce monthly storage costs significantly compared to S3 Standard

Typical workloads include media asset libraries, healthcare information archiving, legal information archiving, scientific datasets, and any long-lived, infrequently accessed data that must remain instantly retrievable.

GIR is **not** appropriate for:

© 2026 - Hammerspace

- Active working sets that are frequently read or written
- Data that is read often enough that egress costs would exceed the storage savings
- Files that are very small (under 128 KB) in large quantities — AWS bills each object at a minimum of 128 KB
- Data with a short lifecycle — AWS charges a minimum of 90 days of storage per object regardless of when it is deleted
- Data used as a replication transfer conduit between Hammerspace sites

9.4.2. Cost Considerations

GIR's pricing model differs significantly from S3 Standard, and these differences drive nearly all of Hammerspace's GIR behavior. Understanding the cost model before deploying GIR is strongly recommended.



The figures below are representative values intended to illustrate the relative cost differences between storage classes. Refer to the AWS S3 Pricing page for current pricing in your AWS region.

Cost dimension	S3 Standard	S3 Glacier Instant Retrieval
Storage per GB / month	$0.023	$0.004 (~5.75× cheaper)
`PUT` requests per 1,000	$0.005	$0.02 (4× more expensive)
`GET` requests per 1,000	$0.0004	$0.02
Per-GB retrieval	none	$0.03



Reading from GIR incurs retrieval and request charges that Standard storage does not. How much more a given read costs depends on the region, the object sizes, and the number of requests, so no single ratio holds generally — work it out for your own access pattern using the AWS S3 Pricing page.

Hammerspace deliberately constrains GIR reads to limit this exposure, but any read activity, including mobility jobs, drives measurable charges.

Additional Billing Rules

Minimum object size: 128 KB. AWS bills a minimum of 128 KB per stored object regardless of actual file size. A 10-byte file stored on a GIR OSV is billed at the 128 KB rate. For workloads with large numbers of small files, this can make GIR more expensive than Standard, not less.

Minimum retention period: 90 days. Objects deleted before day 90 are still charged for the remainder of the 90-day window. Hammerspace applies a 90-day minimum time-to-live (TTL) at the Cloud Mover layer to align its own garbage collection behavior with this billing rule. Deleting a file from a Hammerspace share does not trigger an immediate charge, but AWS continues to bill for storage for up to 90 days.

Object overwrite before 90 days. If a file stored on a GIR OSV is overwritten before its 90-day window has elapsed, the previous version is billed for the remainder of its window and the new version starts a fresh 90-day period.

© 2026 - Hammerspace

Cross-region retrieval. Cross-region `GET` requests incur an additional ~$0.02/GB transfer charge on top of the standard GIR retrieval fee.

9.4.3. Cost Observability and Reporting Limitations



These are known limitations of the initial GIR release. Improved cost observability is planned for a future release.

Hammerspace does not currently integrate with the AWS billing or CloudWatch Cost APIs. Before deploying GIR, be aware of the following:

No in-product cost estimation. Hammerspace cannot calculate what placing a given amount of data on GIR will cost. Use the AWS Pricing Calculator for storage projections before committing data to GIR.

Capacity reporting discrepancies. Because of the 90-day minimum billing window, the storage capacity that Hammerspace reports as "in use" on a GIR OSV will not always match what AWS bills for the same volume. Data that has been deleted from the share can continue to appear as a billing charge in AWS for up to 90 days.

No per-volume upload/download counters. Usage counters and customer-defined rate thresholds — for example, "alert me if more than X GB is uploaded to this OSV per week" — are not part of this release. These are planned as a future enhancement.

Until those enhancements are available, monitor AWS billing and CloudWatch directly for cost visibility on GIR OSVs.

9.4.4. How Hammerspace Manages GIR

Hammerspace applies a set of default policies to GIR volumes specifically designed to prevent unexpected egress and storage charges. The table below summarizes this behavior; each item is discussed in more detail in later sections.

Area	GIR behavior
OSV storage class	Set at creation; **immutable for the life of the OSV**
Default Volume Group	Automatically added to `archive-object-volumes`
Placement	Data is placed on a GIR OSV **only when an explicit** `place-on` **or** `confine-to` **objective targets it**
Availability/durability objectives	GIR does not receive new copies to satisfy generic availability or durability goals alone
Replication transfer conduit	**Excluded by default**
Mobility — copy source	Used **only as a last resort** when no other copy exists
Watermark-based evacuation	**Not applied** to GIR OSVs
COW split	Neither the base nor the clone is left with only a GIR instance unless unavoidable

© 2026 - Hammerspace

Area	GIR behavior
Garbage collection	Runs **once per day** on GIR volumes (less frequent than standard OSVs)
Minimum TTL	90-day minimum TTL applied at the Cloud Mover to match AWS billing

GIR Is a Volume-Level Designation

The GIR storage class applies at the OSV level, not at the share or file level. Every file that Hammerspace places on a GIR OSV is uploaded to AWS with the `x-amz-storage-class: GLACIER_IR` request header. It is not possible to mix GIR and Standard storage within a single OSV.



Hammerspace only writes **data blobs in the Merkle tree** (under the `Hammerspace_v2/` prefix) as GIR. OSV management files — the identity file (`hs_osv_details`, in the bucket root), the site reservation file, and I/O test files — are always stored as Standard class. The S3 bucket itself carries no special GIR designation; it is the Hammerspace Cloud Mover that applies the storage class per blob.

9.4.5. Prerequisites

Before adding a GIR Object Storage Volume, you need:

1. **An AWS account** with an S3 bucket available. The bucket must be empty — it cannot contain existing data when added to Hammerspace.
2. **AWS access credentials** (an access key ID and secret access key) for the service account Hammerspace will use to connect to S3.
3. **IAM permissions** on the AWS service account that allow the required S3 operations. GIR needs no permissions beyond those of any AWS S3 volume; see the list in Adding an AWS S3 Storage System.
4. **The AWS S3 storage system is registered in Hammerspace.** If your organization already uses S3 in Hammerspace, the storage system may already be present. If not, add it first using the procedure in the next section.

9.4.6. Adding an AWS S3 Storage System



This procedure is required the first time you connect an AWS S3 storage system to your Hammerspace cluster. If an AWS S3 storage system is already registered in **Infrastructure › Storage Systems**, skip to Adding a GIR Object Storage Volume.

This procedure applies to all AWS S3 object storage use cases, not only GIR.

The AWS service account whose access key you enter needs these S3 permissions on the buckets that Hammerspace uses:

- `s3:ListAllMyBuckets`
- `s3:ListBucket`
- `s3:GetObject`

© 2026 - Hammerspace

- `s3:PutObject`
- `s3:DeleteObject`
- `s3:AbortMultipartUpload`
- `s3:GetBucketLocation`, when Hammerspace must find each bucket's AWS region itself (the storage system's **Endpoint** is set to **Auto Select**)

GIR volumes need no permissions beyond these.

Using the Web UI

1. Go to **Infrastructure** › **Storage Systems**, then click **Add Storage System**.
2. In the Add Storage System dialog:
   - Enter a **Name** for the storage system.
   - For **Storage Type**, select **Object Storage**.
   - For **Type**, select **Amazon S3**.

   Glacier Instant Retrieval is supported only on Amazon S3 storage systems. If you select Azure, GCP, or Hammerspace S3, the GIR storage class option will not be available when you later add a volume.

3. Scroll to the **Access Details** section and enter the **Access Key** (access key ID) and **Secret Key** (secret access key) for the AWS service account. Hammerspace uses these credentials for all subsequent S3 operations on this storage system.

   The AWS region configured on the storage system is used for volumes added on it.

4. Configure any optional settings as needed, then click **Add Storage System**.
5. Hammerspace connects to AWS and validates the credentials. When the storage system is added successfully, the dialog displays **Storage System <name> created successfully**. You can click the button in the dialog to proceed directly to adding volumes, or close the dialog and manage the storage system later from **Infrastructure** › **Storage Systems**.

Using the Admin CLI

```
object-storage-add \
  --name <storage-system-name> \
  --type amazon_s3 \
  --access-id <ACCESS_KEY_ID> \
  --secret <SECRET_ACCESS_KEY> \
  --async
```

After the command completes, verify that the storage system appears in the list:

© 2026 - Hammerspace

```
node-list --name <storage-system-name>
```

9.4.7. Adding a GIR Object Storage Volume

Once an AWS S3 storage system is registered in Hammerspace, you can add an Object Storage Volume with the GIR storage class. The GIR storage class is supported only on AWS S3; a generic S3 endpoint accepts the Standard storage class and rejects GIR.



The storage class of an OSV is **set at creation and cannot be changed afterward**. Plan your OSV layout carefully before adding volumes.

Use a new, empty S3 bucket for a GIR OSV. Adding a bucket that already contains data is not a supported configuration, and an existing volume cannot be converted to GIR.

Using the Web UI

1. Go to **Infrastructure** › **Storage Systems** and click the **+ Volume** button next to the AWS S3 storage system.
2. In the Add Volumes wizard, select the S3 bucket or buckets to add as volumes.
3. On the **Data** tab, locate the **Archive Type** field. Leave **Archive Type** unselected to use Standard storage, or select **AWS Glacier Instant Retrieval** to create a GIR volume.



When **Archive Type** is unselected, the volume is added as a regular Standard-class OSV and the **Review & Add** screen displays a dash for the field. Only volumes explicitly set to **AWS Glacier Instant Retrieval** are managed as GIR.

4. Review and adjust the capacity, availability, durability, and IP settings as needed, or accept the defaults.
5. On the **Review & Add** screen, confirm that **Archive Type** shows **AWS Glacier Instant Retrieval** before submitting. A dash means the volume will be created as Standard.
6. Click **Add Volume**.

Using the Admin CLI

```
object-volume-add \
  --node-name <aws-storage-system-name> \
  --logical-volume-name <bucket-name> \
  --storage-class AWS_GLACIER_INSTANT_RETRIEVAL \
  --create-placement-objectives
```

The `--storage-class` parameter accepts the following values (not case-sensitive):

- `STANDARD` — default if omitted; adds a standard-class OSV
- `AWS_GLACIER_INSTANT_RETRIEVAL` — adds a GIR-class OSV

The `--create-placement-objectives` flag automatically creates `place-on`, `confine-to`, and `exclude-from`

© 2026 - Hammerspace

objectives for the new OSV. Using this flag is recommended when adding a GIR OSV so the objectives are ready when you configure share archive rules.



`--storage-class` is rejected with an error on Azure, GCP, and Hammerspace S3 storage nodes. GIR is AWS S3 only.

Default Settings for GIR OSVs

A new GIR OSV receives the same default settings as a standard object storage volume:

Setting	Default
High / Low capacity threshold	75% / 60%
Availability	99.99%
Durability	99.9999999%
Online delay	5 minutes

You can adjust these when adding the volume or accept the defaults. Note that although the capacity thresholds are set on GIR OSVs, watermark-based data evacuation is not applied to them (see Mobility behavior).

Verifying the OSV Was Added Correctly

After adding the volume, confirm the configuration with `object-volume-list --name`, naming the volume as `<storage-system-name>::<bucket-name>`. (The plain `object-volume-list` summary does not show the storage class or the volume groups.)

Command:

```
object-volume-list --name <aws-storage-system-name>::<bucket-name>
```

In the output, look for the `Storage class` line and the `archive-object-volumes` entry under `Locations`:

Expected output:

```
ID:                     d4385b5f-e24d-4759-bc62-5b97a1023e49
Name:                   aws-s3::docs-gir-bucket
Internal ID:            268435473
State:                  OK
Node:                   aws-s3
Oper state:             Up
Admin state:            Up
Capabilities:           High threshold:           75%
                        Low threshold:            60%
                        Availability:             99.99%
                        Availability (effective): 99.99%
```


© 2026 - Hammerspace

```
                        Availability drop:        Disabled
                        Durability:               99.9999999%
                        Durability (effective):   99.9999999%
                        Online delay:             5 minutes
Total capacity:         Unlimited
Physical used:          0B
Logical used:           0B
Physical free:          Unlimited
Naming type:            Hash named
Compression type:       No compression
Chunking type:          Fixed-size
Storage class:          AWS Glacier Instant Retrieval
Encrypt using KMS:      None. No Kms Configured.
Created:                2026-09-22 13:23:45 UTC
Modified:               2026-09-22 13:23:45 UTC
Locations:
                        [Type: Volume Group, ID: 4bc06571-70ea-4f20-b9df-dfd3d92c7ce6, Name: archive-
object-volumes]
                        [Type: Node, ID: fe49a56c-aa46-48ca-b762-1fc5f33d08dc, Name: aws-s3]
                        [Type: Object Storage Volume, ID: d4385b5f-e24d-4759-bc62-5b97a1023e49, Name:
aws-s3::docs-gir-bucket]
                        [Type: Volume Group, ID: 275f5acb-f7a3-4010-9127-d5a9f12fca02, Name: all]
Logical volume:         Type:                     Object Store Logical Volume
                        ID:                       fb1d5485-48aa-48c3-9081-48ed007284ac
                        Name:                     docs-gir-bucket
                        Ownership:                Not shared
GC enabled:             Yes
```

Standard OSVs show `Storage class: Standard`. If the GIR OSV does not appear in `archive-object-volumes`, see the troubleshooting section.

To confirm that data uploaded to the OSV is being correctly tagged in S3, use the AWS CLI after placing a file on the volume:

Command:

```
aws s3api head-object \
  --bucket <bucket-name> \
  --key Hammerspace_v2/<object-path>
```

The response should include `"StorageClass": "GLACIER_IR"`.

9.4.8. The Archive-Object-Volumes Volume Group

Hammerspace provides a built-in, system-managed Volume Group named `archive-object-volumes`. The group is present by default, with an empty membership expression until a GIR OSV is added. It has the following characteristics:

© 2026 - Hammerspace

- All GIR OSVs are **added automatically**. You do not need to add them manually.
- **Standard OSVs cannot be added** to `archive-object-volumes`.
- GIR OSVs **cannot be manually removed** from the group. A GIR OSV leaves the group only when the underlying OSV is deleted.
- The Volume Group **cannot be renamed or deleted**.
- Three placement objectives are **built in**: `place-on-archive-object-volumes`, `confine-to-archive-object-volumes`, and `exclude-from-archive-object-volumes`.

The separation between `archive-object-volumes` and the standard object-volumes and shared-object-volumes groups is intentional. It prevents general-purpose placement objectives — such as availability or durability rules that target object storage broadly — from inadvertently landing data on a GIR OSV and incurring unexpected costs.

Confirm that the Volume Group exists and contains your GIR OSV:

```
volume-group-list --name archive-object-volumes
```

Confirm that the placement objectives were created:

```
objective-list
```

Look for objectives named `place-on-archive-object-volumes`, `confine-to-archive-object-volumes`, and `exclude-from-archive-object-volumes` in the output.

9.4.9. Placing Data on GIR — Configuring Share Archive Rules

Hammerspace will not place data on a GIR OSV automatically. Every share that should have data archived to GIR must have an explicit placement objective configured.

The most administrator-friendly way to set this up is through the **Archive** tab in the Edit Share workflow, which lets you define conditions that determine which files on the share are eligible for GIR archival.

Adding an Archive Rule Using the Web UI

1. Go to **Data** › **Shares** and click the **edit** icon next to the share.
2. Select the **Archive** tab.
3. Click **Add Archive Rule** to build a rule from scratch, or **Add Rule from Template** to start from a predefined pattern.

   Available templates include:

   - File not accessible (Data only)
   - File not accessible (Data and Snapshot)

© 2026 - Hammerspace

- Has Label
- Has Tag

4. Set the **archive condition**. For time-based conditions (for example, files not accessed for more than 3 months), use the time and unit fields on the main tab. Set the rule's time threshold using the **Days**, **Months**, or **Years** units. For more complex expressions, use the **Advanced** tab to compose a custom condition such as `Last_use_age > 90 days`.
5. In the rule's placement dropdown, select the `place-on` or `confine-to` objective that was created for your GIR volume (for example, `place-on-archive-object-volumes`).
6. Click **Save**. The rule appears in the rules table on the Archive tab and is also visible as a placement objective on the share in both the UI and the CLI.



Rules added through the Archive tab in the GUI are also visible and manageable through the Admin CLI. To inspect the objectives on a share from the CLI, use `share-list` or `objective-list`.

Adding a Placement Objective Using the Admin CLI

You can also apply placement objectives to a share directly from the CLI without going through the Archive tab:

```
share-objective-add \
  --name <share-name> \
  --objective place-on-archive-object-volumes
```

To apply the objective only when a condition is met, add `--applicability` with an objective expression. For example, to place files on archive object volumes only after they have gone unused for 90 days:

```
share-objective-add \
  --name <share-name> \
  --objective place-on-archive-object-volumes \
  --applicability "LAST_USE_AGE>90*DAYS"
```

When `--applicability` is omitted, the objective always applies. For the expression syntax, see Common Functions for Advanced Expressions.

Placement Objective Reference

Objective	Effect
`place-on-archive-object-volumes`	Directs Hammerspace to place a copy of the file on the GIR OSV. Without this or `confine-to`, Hammerspace will never upload to GIR — even if availability or durability objectives could otherwise be satisfied by doing so.

© 2026 - Hammerspace

Objective	Effect
`confine-to-archive-object-volumes`	Forces GIR to be the **only** instance of the file. Triggers removal of all NAS and Standard OSV copies. Use with care: the data then exists only on GIR, and any subsequent read will incur GIR retrieval costs.
`exclude-from-archive-object-volumes`	Prevents new copies from being placed on GIR, and moves any existing GIR copies off. Applying this to data that is already on a GIR OSV triggers an egress read, so apply with awareness of the cost.
`keep-online`	Maintains a NAS copy alongside an existing GIR instance so data can be read from NAS without incurring GIR retrieval costs. **`keep-online` by itself does not place a copy on GIR** — to keep both a NAS copy and a GIR copy, apply `place-on-archive-object-volumes` together with `keep-online`.



Smart (attribute-based) place-on rules are supported. For example, you can archive any file with `last_use_age > 3 months` to the `archive-object-volumes` Volume Group, and Hammerspace will apply the rule only to files that meet the condition.

9.4.10. Mobility Behavior

Hammerspace's Data Mobility Engine (DME) treats GIR volumes as a last-resort copy source. DME prefers a usable Standard source whenever one is available. A GIR OSV becomes eligible as a copy source only when no usable Standard source is available; a Standard copy that exists but is not usable does not prevent DME from falling back to GIR.

This behavior is deliberate. Reading from a GIR OSV can incur retrieval and request charges that Standard storage does not incur. Charges vary by AWS region and by usage pattern; consult Amazon S3 pricing for current rates. DME is designed to avoid incurring these charges during routine mobility operations.

Key practical effects:

- A `keep-online` objective on a share that has a GIR instance creates a NAS copy from GIR, incurring the retrieval cost once. Later reads use that NAS copy for as long as it remains available and retained. If that NAS copy is later removed or becomes unavailable, another GIR retrieval can be required.
- Multiple non-GIR copies are always preferred over GIR for COPY and clone operations.
- For COW (copy-on-write) splits, Hammerspace prefers a Standard archive copy when both Standard and GIR copies exist, and uses GIR when it is the only archive choice. When a file has both NAS and GIR copies, the live side retains the NAS copy while the snapshot side can remain GIR-only.
- `optimize-for-capacity` can cause instance drops, including archive and GIR instances, when no objective requires that instance. For example, a GIR instance may be dropped if no `place-on` objective or other objective depends on it.
- DME does not create new copies on a GIR volume solely because other volumes are under capacity pressure. However, if a GIR volume is itself over its configured capacity threshold, DME can raise the egress capacity cost and may move files off the GIR volume. Configure the GIR volume with unlimited capacity, or with thresholds that effectively disable capacity pressure, if you do not want capacity-driven evictions from

© 2026 - Hammerspace

a GIR volume.

How GIR Contributes to Availability and Durability Scoring

GIR OSVs are assigned the same nominal availability and durability values as Standard OSVs. The Hammerspace defaults are 99.99% (four nines) availability, 99.9999999% (nine nines) durability, and a five-minute online delay. These are configurable Hammerspace defaults — they are not fixed limits, and they are not AWS service guarantees. For AWS's published characteristics of each storage class, see Amazon S3 storage classes.

The rules that determine what you see are:

- Existing GIR instances contribute to alignment using their configured Hammerspace availability and durability values.
- The DME does not create a new GIR instance solely to satisfy a generic availability or durability objective.
- An explicit placement objective can still place data on GIR.

The practical result: if a share has a four-nines availability objective and the only available volume is a GIR OSV, the system does not upload to GIR to satisfy the objective, and the share remains misaligned. To place data on GIR, use an explicit `place-on` or `confine-to` objective that targets `archive-object-volumes` or a specific GIR OSV.

9.4.11. Replication Behavior

Hammerspace uses Shared Object Storage Volumes (SOSVs) as conduits for cross-site replication. The system may reuse data already present on a shared OSV, or create data for the transfer. Data created only for replication is removed through normal retention and garbage-collection processing; it is not necessarily deleted immediately after the transfer completes. GIR is therefore a poor default conduit choice, because AWS applies a 90-day minimum storage duration and per-GB retrieval charges to GIR data. See Amazon S3 pricing.

GIR OSVs are therefore **excluded from the replication transfer conduit pool by default**. This applies even if a GIR SOSV is the only shared OSV available. If your deployment requires replication and only GIR shared volumes are reachable, contact Hammerspace Support. Hammerspace Support can enable GIR as a last-resort conduit; Standard conduits remain preferred whenever an active one is available.

9.4.12. Shared and Mixed-Version Configurations

GIR OSVs can be marked as shared (Shared Object Storage Volume, or SOSV), allowing multiple Hammerspace sites to use the same GIR S3 bucket.

Adding a Shared GIR OSV

Use the `--shared` flag when adding the OSV:

```
object-volume-add \
  --node-name <aws-storage-system-name> \
  --logical-volume-name <bucket-name> \
```


© 2026 - Hammerspace

```
  --storage-class AWS_GLACIER_INSTANT_RETRIEVAL \
  --shared \
  --create-placement-objectives
```

In the GUI, check the **Shared** option in the Add Volumes wizard.

Storage Class Validation on Shared OSVs

When a GIR-aware site adds a shared OSV that is already in use by another site, Hammerspace reads the OSV identity file (`hs_osv_details`) from the bucket root and validates both the recorded storage class and the recorded feature set. Matching storage classes alone is not sufficient.

Storage class. If the recorded class matches the class being requested, the OSV is added. If it does not, the operation fails with:

```
Cannot add shared Object Storage Volume: storage class mismatch. Bucket configured with <existing_class>,
requested <new_class>.
```

This prevents a GIR bucket from being added as Standard on one site and GIR on another.

Features and feature versions. The identity file also records the features in use on the volume and the version of each. Adding a GIR OSV requires the GIR feature at version 1 or later. A site rejects the volume in either of these cases:

- The identity file records a feature this site does not recognize, or records a feature at a version higher than this site supports. This happens when another site is running a newer release. The operation fails with an instruction to upgrade this site before adding the volume.
- The requested storage class needs a feature that the other sites sharing the volume do not support. The operation fails, and all sites must remove the OSV and re-add it with the feature enabled.

Mixed-Version Deployments (Sites with and Without Storage Class Support)

GIR is supported beginning with Hammerspace 5.3. All sites sharing a GIR SOSV must run a release that supports GIR.

Do not add or re-add a shared GIR SOSV from a Hammerspace 5.2 site. A 5.2 site does not recognize the GIR storage class recorded in the identity file and can treat the volume as Standard. That bypasses the GIR-aware source-selection and conduit protections, so data placement from the 5.2 site may not respect the GIR cost protections.

A 5.2 site accepts the OSV without error, because it has no awareness of storage class and ignores the newer fields in the identity file. There is no product enforcement preventing this. Between GIR-aware sites, every site must configure the shared OSV with the same storage class, and the feature validation described in Storage Class Validation on Shared OSVs applies.

© 2026 - Hammerspace

Legacy OSV Identity Files

Buckets created before storage class support was introduced have no storage class and no feature list in their identity file. A GIR-aware site that reads such a file treats the OSV as Standard class with no features enabled.

9.4.13. Verifying Your GIR Configuration

Use the following checks to confirm that a GIR OSV is correctly configured.

What to verify	Command
Storage class on the OSV	`object-volume-list` — look for `Storage class: AWS Glacier Instant Retrieval`
OSV membership in `archive-object-volumes`	`volume-group-list --name archive-object-volumes`
Placement objectives were created	`objective-list` — look for `*-archive-object-volumes` objectives
S3-side storage class tag	`aws s3api head-object --bucket <bucket> --key Hammerspace_v2/<path>` — look for `"StorageClass": "GLACIER_IR"`
Storage class in the database	`runuser -l postgres -c 'psql -d director -c "SELECT uuid,storageclass FROM objectstoragevolume;"'`
OSV identity file in the bucket	Inspect `hs_osv_details` in the bucket root — look for the storage class `AWS_GLACIER_INSTANT_RETRIEVAL`
DME recognition of the class	Search DME logs for `AWS_GLACIER_INSTANT_RETRIEVAL`
Archive rules visible in CLI	`objective-list` or `share-list` — rules added via GUI should appear here

9.4.14. Limitations and Constraints

#	Limitation
1	**The GIR storage class is supported only on AWS S3.** GIR is not available on Azure, GCP, or Hammerspace S3 storage nodes. The `--storage-class` option and the GUI **Archive Type** field are not available on those node types. Generic S3 endpoints accept the Standard storage class and reject GIR.
2	**Storage class is set when an OSV is created and cannot be changed later.** Existing Standard OSVs cannot be converted to GIR, and existing GIR OSVs cannot be converted to Standard. To use a different storage class, create a new OSV using a new, empty bucket and migrate data by updating placement objectives.
3	**Explicit placement is required to add a new GIR copy.** Availability and durability objectives alone do not cause Hammerspace to place new data on GIR. Use a `place-on` or `confine-to` objective that targets `archive-object-volumes` or a specific GIR OSV. Existing GIR copies can remain, subject to snapshot, copy-on-write, and retention behavior.

© 2026 - Hammerspace

#	Limitation
4	**All sites sharing a GIR SOSV must run a release that supports GIR.** GIR is supported beginning with Hammerspace 5.3. Do not add or re-add a shared GIR SOSV from a 5.2 site: that site does not recognize the GIR storage class and can treat the volume as Standard. Between GIR-aware sites, every site must configure the shared OSV with the same storage class, and a mismatch causes the add to fail with a storage class mismatch error.
5	**No impact or change to the Data Copy to Object feature.** GIR does not interact with or modify the behavior of that feature.
6	**No cost reporting or AWS billing integration.** Hammerspace cannot show projected or historical GIR costs. Capacity reported by Hammerspace may not match AWS billing, because AWS applies a minimum billable object size and a minimum storage duration.
7	**No GIR-specific usage or cost alerts.** Hammerspace does not provide GIR-specific upload and download counters, cost projections, or billing-alert thresholds. Normal object-volume capacity thresholds remain available on GIR OSVs.
8	**AWS billing minimums apply.** AWS bills each GIR object for at least 128 KB and applies a 90-day minimum storage duration. Deleting, overwriting, or transitioning an object sooner can incur a prorated charge for the remaining duration. Hammerspace delays cleanup of GIR data to reduce early-deletion exposure, but cannot guarantee that these AWS charges are avoided.

9.4.15. Upgrade Considerations

Existing OSVs are unaffected. Object Storage Volumes created before storage class support was introduced have no storage class field and are treated as Standard class. No migration or reconfiguration is required on upgrade.

The `archive-object-volumes` Volume Group is present by default. The Volume Group is created with the system, with an empty membership expression. Adding a GIR OSV does not create the group; the OSV becomes a member of the group that is already there. If no GIR OSV exists, the group remains present and empty.

Mixed-version clusters. Sites that do not support storage class can participate in a cluster that uses a GIR SOSV but will not enforce GIR mobility and replication guardrails. If you require uniform GIR behavior across all sites, upgrade all sites to a release that supports storage class before introducing GIR SOSVs.

OSV identity file versioning. The OSV identity file (`hs_osv_details`, in the bucket root) is updated to record the storage class. The file also includes a feature version indicator that allows newer Hammerspace versions to refuse to join a shared OSV when the bucket's identity file was created by a version too far ahead of the local site's version. This safeguard prevents version incompatibility errors from silently degrading GIR behavior.

9.4.16. Troubleshooting

Data is not being placed on the GIR OSV. Check whether the share has an explicit `place-on-archive-object-volumes` or `confine-to-archive-object-volumes` objective. Availability and durability objectives alone will not drive data to GIR. Confirm with `objective-list` that the objective exists and is applied to the correct share.

© 2026 - Hammerspace

The GIR OSV is not appearing in the `archive-object-volumes` Volume Group. Verify that the OSV was actually created with `--storage-class AWS_GLACIER_INSTANT_RETRIEVAL`. Use `object-volume-list` and check the Storage class field. If the field shows Standard, the OSV was not added as GIR. Delete the OSV and re-add it with the correct storage class.

"Invalid storage-class" error on the CLI. The value must be one of the enum tokens, `STANDARD` or `AWS_GLACIER_INSTANT_RETRIEVAL`. The token is **not** case-sensitive — for example, `AWS_glacier_instant_retrieval` is accepted — but the spaced display form 'AWS Glacier Instant Retrieval' is rejected because it is not a valid token.

Storage class parameter rejected on a non-AWS storage node. GIR is supported only on Amazon S3 storage systems. The `--storage-class` parameter is not valid for Azure, GCP, or Hammerspace S3 nodes.

Storage class mismatch error when adding a shared OSV. Two storage-class-aware sites are attempting to share an OSV with conflicting storage class designations. The second site's add request does not match the storage class recorded in the OSV identity file (`hs_osv_details`). Both sites must add the SOSV with the same storage class.

Unexpected AWS billing for data that has already been deleted from the share. This is expected behavior caused by the 90-day AWS minimum retention period. Hammerspace applies a 90-day minimum TTL at the Cloud Mover layer, but AWS billing for recently deleted objects continues through the end of the 90-day window. Use AWS billing tools directly to monitor charges.

Hammerspace-reported capacity on the GIR OSV does not match AWS billing. This is a known limitation. Because of the 90-day minimum billing window, recently deleted data may still be billed by AWS after it no longer appears in Hammerspace capacity reports. These figures cannot be reconciled within the product in this release.

A GIR OSV appears to be used as a replication conduit. This should not occur: GIR OSVs are excluded from the replication transfer conduit pool by default. If you observe replication traffic on a GIR volume, contact Hammerspace Support.

Cannot change the storage class on an existing OSV. By design. Storage class is immutable. Create a new OSV with the desired class and use placement objectives to migrate data.

Setting a lower total capacity on an OSV is rejected. You cannot set the total capacity of an object storage volume, GIR or otherwise, below the amount of data already stored on it. The update is refused with the error `The specified total capacity '<value>' must be greater than current used bytes '<used>'.` Choose a value above the current used capacity, or move data off the volume first.

New GIR copies are not being created to satisfy an availability objective. By design. Availability and durability objectives alone do not cause Hammerspace to create a new GIR copy. Apply an explicit `place-on` objective that targets `archive-object-volumes` or a specific GIR OSV. Existing GIR copies are still considered when Hammerspace evaluates the current data layout.

9.4.17. CLI Reference: object-volume-add

The `object-volume-add` command accepts the following options. The GIR-specific addition is `--storage-class`.

© 2026 - Hammerspace

Two selectors are required:

- One node selector: `--node-name` or `--node-id`.
- One logical-volume selector: `--logical-volume-name` or `--logical-volume-id`.

The following options are mutually exclusive: `--gc-enable` and `--gc-disable`; `--use-high-compression` and `--no-compression`.

Option	Description
`--access-key`	Access key for this volume. If omitted, the credentials configured on the storage system are used. Supplying `--access-key` without `--secret-key` prompts for the secret key; supplying `--secret-key` requires `--access-key`.
`--availability`	Volume availability capability, expressed as the number of nines (0–20).
`--create-placement-objectives`	Creates default `place-on`, `confine-to`, and `exclude-from` objectives for the new volume. Recommended when adding a GIR OSV.
`--durability`	Volume durability capability, expressed as the number of nines (0–20).
`--gc-disable`	Disables coordinated garbage collection. Shared object volumes only. Mutually exclusive with `--gc-enable`.
`--gc-enable`	Enables coordinated garbage collection. Shared object volumes only. Mutually exclusive with `--gc-disable`.
`--help`	Displays the help text and exits.
`--use-high-compression`	Enables high compression when storing data. Compatible with GIR OSVs. Mutually exclusive with `--no-compression`.
`--high-threshold`	Percentage (0–100) of capacity that triggers data evacuation. Defaults to 75%. Specifying this option does not change the `--low-threshold` default.
`--logical-volume-name`	Object logical volume (bucket) name. Required unless `--logical-volume-id` is supplied.
`--logical-volume-id`	Object logical volume (bucket) ID. Required unless `--logical-volume-name` is supplied.
`--low-threshold`	Target percentage (0–100) at which evacuation stops. Defaults to 60% when omitted, and specifying `--high-threshold` does not change that default. To use a different pair, supply both options explicitly — for example, `--high-threshold 90` with `--low-threshold 72`.
`--max-suspected-time`	Time a volume stays in SUSPECTED state before becoming UNAVAILABLE. For example: 10seconds, 1minute.
`--name`	Display name for the OSV. A default is assigned if omitted.

© 2026 - Hammerspace

Option	Description
`--native`	Stores data as non-chunked, unencrypted, and uncompressed. Compatible with GIR OSVs.
`--no-compression`	Disables compression. Mutually exclusive with `--use-high-compression`.
`--no-encryption`	Disables encryption.
`--node-name`	Name of the storage node (storage system) to add the volume on. Required unless `--node-id` is supplied.
`--node-id`	ID of the storage node. Required unless `--node-name` is supplied.
`--online-delay`	Maximum time (in nanoseconds) allowed to restore a file from this volume.
`--realign-on-protection-drop-disabled`	Prevents aggressive realignment when availability or durability is reduced.
`--secret-key`	Secret key for this volume. If omitted, the credentials configured on the storage system are used. Requires `--access-key`.
`--shared`	Marks the volume as a Shared Object Storage Volume (SOSV).
`--storage-class`	Storage class for cost-optimized tiers. Defaults to `STANDARD` if omitted. Cannot be changed after OSV creation. Valid values: `STANDARD`, `AWS_GLACIER_INSTANT_RETRIEVAL` (not case-sensitive). AWS S3 only.
`--total-capacity`	Total data capacity allowed on this volume. Defaults to practically unlimited if omitted.

9.4.18. FAQ

Will Hammerspace ever upload data to a GIR OSV without an explicit objective? For normal data placement, no. Availability and durability objectives alone do not cause Hammerspace to create a new GIR copy. To place data on GIR, use a `place-on` or `confine-to` objective that targets the `archive-object-volumes` Volume Group or a specific GIR OSV. Existing GIR copies can remain or be reused through retention, snapshots, and copy-on-write behavior. Replication has a separate, Support-controlled exception that can use GIR as a last-resort conduit when no Standard conduit is usable; see Replication Behavior.

Can I change a Standard OSV to a GIR OSV (or vice versa) later? No. Storage class is set at creation and is immutable. To move data between storage classes, add a new OSV with the target class, apply placement objectives that target the new OSV, and allow mobility to complete the migration.

Can I enable GIR on a bucket that already contains data? No. Adding a bucket that already contains data as a GIR OSV is not a supported configuration. Create the GIR OSV using a new, empty bucket. To migrate Hammerspace-managed data from a Standard OSV, add the GIR OSV, apply a `place-on` objective for it to the relevant shares, verify that mobility completes, then remove the old placement.

What happens when I delete a file that lives only on a GIR OSV before 90 days have elapsed? The file is removed from the Hammerspace namespace immediately. Its underlying S3 data is deleted later, through Hammerspace retention and garbage-collection processing; this is not an exact 90-day timer. Separately, AWS

© 2026 - Hammerspace

applies a 90-day minimum storage duration and can charge for the remaining duration when GIR data is deleted, overwritten, or transitioned earlier.

Why aren't new GIR copies being created when I raise my availability requirement? By design. Availability and durability objectives alone do not cause Hammerspace to create a new GIR copy. To place data on GIR, apply an explicit `place-on` objective that targets `archive-object-volumes` or a specific GIR OSV. Existing GIR copies can still be considered when Hammerspace evaluates the current data layout.

Is GIR storage compatible with Hammerspace compression? Yes. The `--use-high-compression` option is supported on GIR OSVs. Compression can reduce the amount of data stored and retrieved, but the actual savings depend on data compressibility, object layout, request count, and current AWS pricing. A smaller upload does not by itself reduce a PUT charge, because AWS charges PUT requests per request rather than by size.

Is the `--native` (non-chunked) mode supported on GIR OSVs? Yes. Adding a GIR OSV with `--native` stores data as non-chunked, unencrypted, and uncompressed blobs. This is verified in QA.

Can I see what AWS will charge me for a GIR OSV from inside Hammerspace? No. Hammerspace does not integrate with AWS cost APIs in this release. Use the AWS Pricing Calculator for projections, and AWS Billing and Cost Management tools such as Cost Explorer or cost and usage reports to analyze charges. CloudWatch can provide S3 storage and request metrics, but not the actual charges for a specific Hammerspace OSV.

What happens to pre-existing non-Hammerspace data in a bucket when I add it as an OSV? Hammerspace works exclusively within its own path prefix inside the bucket (`Hammerspace_v2/`). Pre-existing objects outside that prefix are not accessed, modified, or deleted by Hammerspace. However, adding a bucket that already contains data is not a supported configuration for GIR OSVs. Use an empty bucket.

Does GIR apply to the whole share, to individual volumes, or to individual files? GIR is a **volume-level** designation. Every file that Hammerspace places on a GIR OSV is stored as GIR. You control *which* files are placed on a GIR OSV by configuring placement objectives on the share — for example, a rule that archives files not accessed in 90 days. The GIR designation itself is not applied per-file or per-share; it is an attribute of the OSV.

Can a single share have some files on Standard OSVs and others on GIR OSVs? Yes. You can configure a share with a `keep-online` objective (for NAS or Standard cloud availability) and a `place-on-archive-object-volumes` objective with an age-based condition. Hammerspace will keep active files on Standard storage and archive inactive files to the GIR OSV over time.

9.5. 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 available, and used capacities. This makes it easier to estimate when capacity will run out, and to proactively take action to address capacity issues.

© 2026 - Hammerspace

9.5.1. 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.

9.5.2. 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.

9.6. 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.

9.6.1. 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.

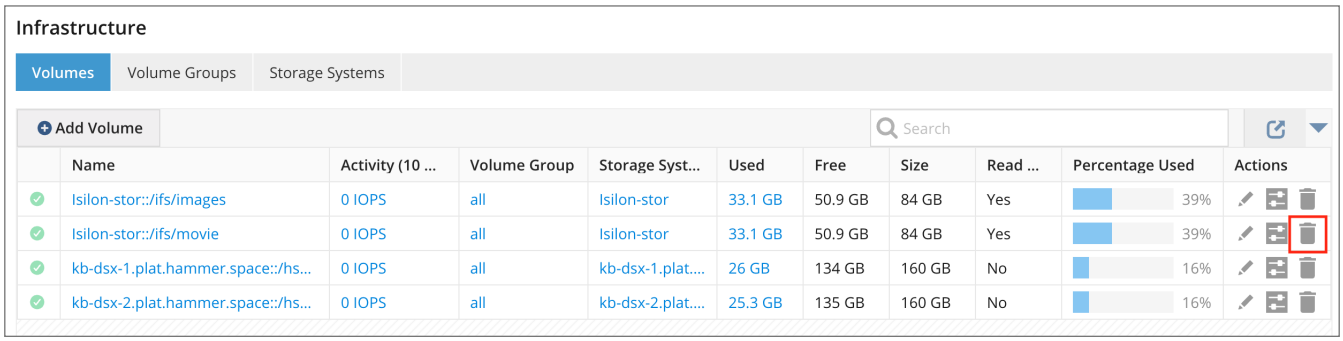


Figure 96. Volume decommission icon

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 decommission and remove in one step, `volume-remove`) with `--replace-with`, for example:

© 2026 - Hammerspace

```
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). If you cleared that option, the task is named `volume-decommission` (`object-volume-decommission` 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.

What Happens During a NAS Volume Decommission

For a NAS volume, Hammerspace works through an ordered sequence of stages. Understanding the order helps you tell normal progress from a stall:

1. **DME quiesce** — no new data placement or ingress on the volume.
2. **Envoy quiesce**.
3. **PDFS quiesce**.
4. **DME regrade**.
5. **Instance drain**, followed by a stronger `safeToClean` check.

© 2026 - Hammerspace

6. **Cleaning**.
7. Final transition to `DECOMMISSIONED`.

Keep the following in mind while the task runs:

- Acknowledgements from the affected components must match the current modification count and reach 100%. Stale, missing, or partial acknowledgements do not advance the task.
- System Management waits five minutes before republishing. Repeated timeouts identify which component is blocking, but the task keeps waiting rather than failing.
- The meaningful progress indicators are the workflow stage, the task status, and the "N instances remaining" count. The task percentage on its own does not prove that the decommission finished safely.
- Insufficient replacement capacity, or a DME readiness condition that has not cleared, can leave the task waiting at instance removal.
- You can cancel through instance removal. Once cleaning or finalization has begun, the task cannot be cancelled.

Do not physically disconnect or delete the backing storage until the task has completed successfully and the volume shows a state of `DECOMMISSIONED` with a workflow stage of `NONE`.

9.6.2. 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:

   ```
   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**.

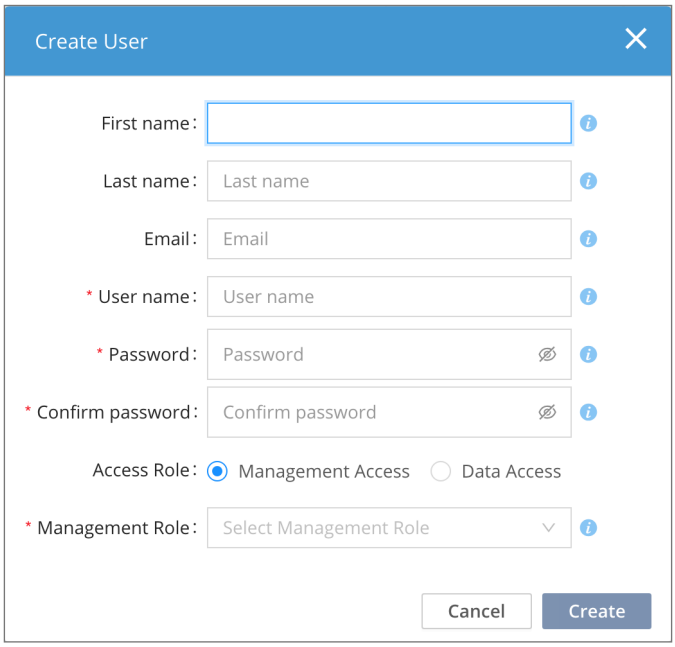


Figure 97. Entering information for a new management user

3. Enter the following information for the user:

Table 18. 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:                           8590b98a-acaf-4f35-baed-024f0fa217c2
Name:                         PM_DOM_A
Type:                         Active Directory
Domain:                       a.pm.test
Servers:                      ad0.a.pm.test
Connection security:          None
Validate server certificates: Disabled
```



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

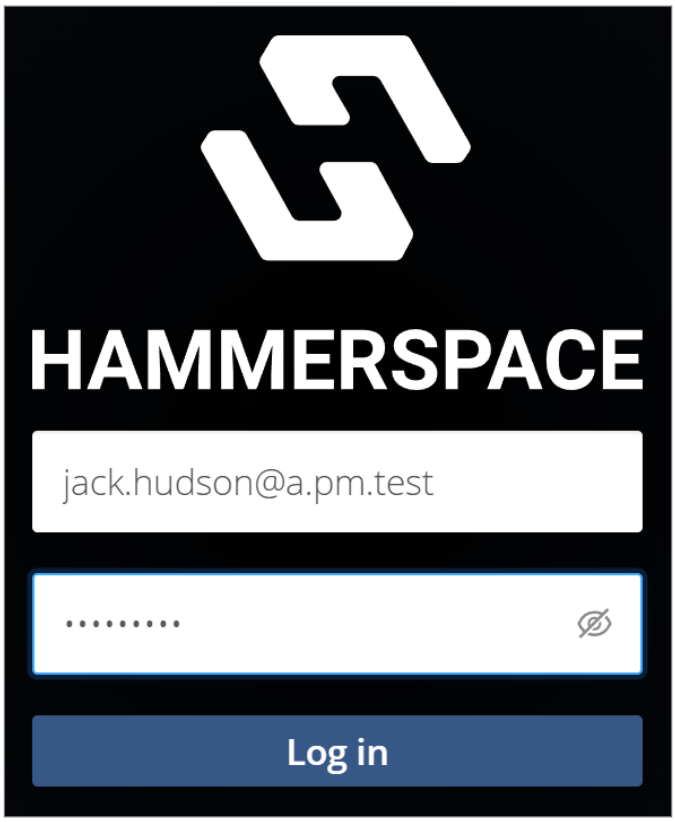


Figure 98. Example login screen with domain address

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 › TLS** and select the **Web Server Certificate** tab. The **Web Server Certificate Details** section shows the installed certificate.

   The page title is **NFS Transport Layer Security**, but the web server certificate is not used by NFS over TLS.

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.

© 2026 - Hammerspace

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 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** › **TLS** and select the **Web Server Certificate** tab. The **Web Server 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.

In Hammerspace 5.3, the certificate options cannot be combined with `--nfs-tls-required` or `--nfs-tls-forbidden` in the same `cluster-config` command. Run them as separate commands.



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-----"
```


© 2026 - Hammerspace

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:                                 aeeadb7f-7258-41e3-aad9-bc57ec176769
Name:                               AEL640TQCSS160
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:                              2026-09-22 13:11:34 UTC
Created:                            2026-09-22 13:11:15 UTC
Timezone:                           UTC
NFS transport policy:               TLS forbidden
GFS participant max suspected time: 30 minutes
Metered billing:                    Disabled
Prometheus exporters:               Disabled
Evaluation expiration:              2026-10-22 13:11:15 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 13:29:06 UTC
Metadata servers:
                         [Object type: Anvil, Node name: anvil.example.com, Role: Primary, Admin state:
Up, Oper state: Up]

Certificate reconfiguration accepted.

*** NOTE: Please reboot with 'node-mode-change --name anvil.example.com --mode REBOOT' to activate. ***
```

The certificate fingerprint can be viewed using the `cluster-view` command.

Admin CLI

Command:

```
cluster-config --view
```

Expected output:

```
ID:                                 aeeadb7f-7258-41e3-aad9-bc57ec176769
Name:                               AEL640TQCSS160
State:                              Standalone
Management IPs:                     [192.0.2.10/20]
Data IPs:                           [192.0.2.10/20]
Cluster floating IPs:               [192.0.2.10/20]
```


© 2026 - Hammerspace

```
Since:                              2026-09-22 13:11:34 UTC
Created:                            2026-09-22 13:11:15 UTC
Timezone:                           UTC
NFS transport policy:               TLS forbidden
GFS participant max suspected time: 30 minutes
Metered billing:                    Disabled
Prometheus exporters:               Disabled
Evaluation expiration:              2026-10-22 13:11:15 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 13:29:06 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:                  C=US,O=Example Docs,CN=Example Docs Root CA
                        Subject:                 C=US,O=Example Docs,CN=anvil.example.com
                        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
                        Subject key identifier:
C3:51:D1:2B:54:35:4B:AE:EF:DC:40:C8:77:81:4E:96:90:AB:15:F3
                        Authority key identifier:
CA:8F:17:8A:55:6D:E7:CA:0C:05:EF:38:12:3A:BE:3D:2D:07:8A:70
                        Subject Alternative Name: DNS Name: [anvil.example.com, *.example.com]

                        Version:                 3
                        Serial:                  170571743814341827810356851105946558195636701158
(0x1de0b39728beeb751870a62aebdfa3b55c0d07e6)
                        Algorithm:               SHA256withRSA
                        Issuer:                  C=US,O=Example Docs,CN=Example Docs Root CA
                        Subject:                 C=US,O=Example Docs,CN=Example Docs Root CA
                        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
                        Subject key identifier:
CA:8F:17:8A:55:6D:E7:CA:0C:05:EF:38:12:3A:BE:3D:2D:07:8A:70
```

```
                        Authority key identifier:
CA:8F:17:8A:55:6D:E7:CA:0C:05:EF:38:12:3A:BE:3D:2D:07:8A:70
```

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 › TLS** and select the **Web Server Certificate** tab.
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:                                aeeadb7f-7258-41e3-aad9-bc57ec176769
Name:                              AEL640TQCSS160
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:                             2026-09-22 13:11:34 UTC
Created:                           2026-09-22 13:11:15 UTC
Timezone:                          UTC
NFS transport policy:              TLS forbidden
GFS participant max suspected time: 30 minutes
Metered billing:                   Disabled
Prometheus exporters:              Disabled
Evaluation expiration:             2026-10-22 13:11:15 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 13:29:06 UTC
Metadata servers:
                        [Object type: Anvil, Node name: anvil.example.com, Role: Primary, Admin state:
```


© 2026 - Hammerspace

```
Up, Oper state: Up]

Certificate reconfiguration accepted.

*** NOTE: Please reboot with 'node-mode-change --name anvil.example.com --mode REBOOT' to activate. ***
```

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.

Restoring a system from backup is not supported when the system has an object-storage volume flagged as *shared* — a configuration used in Global File System deployments that replicate to another site. Recovery halts with the error `recovery cannot be performed on systems with OSVs flagged 'shared'; exiting recovery process`. Do not attempt backup-based recovery on a system that has a shared object-storage volume; contact Hammerspace Support to discuss options.



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 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:

© 2026 - Hammerspace

```
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. Timezone: UTC
ID:                     08267237-438d-4148-917d-7110b844edda
```

Checking the Backup Schedule

Admin CLI

Command:

```
system-backup-config --view
```

Expected output:

```
Export path:            /backup
IP:                     192.0.2.52/32
Schedule name:          backup-schedule
Schedule expression:    10 * * * *
```


© 2026 - Hammerspace

```
Schedule description:    At 10 minutes past the hour. Timezone: UTC
ID:                      42c8fe6b-c276-43d9-a5f7-c6b63cab9ded
```

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. Timezone: UTC
ID:                      84662115-c1fb-492c-8259-4d159f92f096
```

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:            aeeadb7f-7258-41e3-aad9-bc57ec176769
Available Backups:
                         Name:                   20260922.133646Z
                         Time:                   Tue Sep 22 13:36:48 UTC 2026
                         Cluster name:           AEL640TQCSS160
                         Management IP:          192.0.2.10
                         Site name:              AEL640TQCSS160
                         Version:                5.3.2-234
                         Size:                   3.6MB
```



If the path for the `system-backup-list` command does not contain any backups of the

© 2026 - Hammerspace

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.

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:

- Using TLS for end-to-end Encryption

A.1. Using TLS for End-to-End Encryption

Hammerspace 5.3 supports end-to-end, data-path encryption using NFS over mutual Transport Layer Security (mTLS) as a technology preview. With TLS enabled, NFS protocol traffic between clients, the Hammerspace cluster, and backend storage systems is encrypted and mutually authenticated. SMB and S3 client traffic is not covered.

This feature implements IETF RFC 9289, which defines RPC-with-TLS for NFS. It provides wire-level confidentiality and mutual authentication: both the client and the server present X.509 certificates to verify each other's identity before exchanging any data. This protects against eavesdropping and man-in-the-middle attacks on NFS traffic — threats that standard NFS has no native defense against.

TLS secures the **transport channel** and authenticates the **host**, not the user. User identity still comes from the share's RPC security flavor — AUTH_SYS, or Kerberos where it is configured. Per-user certificate authentication is outside the scope of this release.

NFS over TLS is an **all-or-nothing** solution. When enabled, every NFS connection in the environment — from clients to the cluster, from the cluster to backend storage, and between internal cluster components — must use TLS. Partial or per-share TLS configurations are not supported. Within a global file system, the enable wizard recommends enabling TLS on every GFS site, but does not enforce it.

This section includes the following subsections:

- Important Considerations
- Limitations in Hammerspace 5.3
- Prerequisites
- Workflow Overview
- Enable TLS Using the Management GUI
- Remount Clients
- External Movers (DI and Cloud Mover Containers)
- Disable NFS over TLS
- Certificate Lifetimes and Expiration
- Troubleshoot TLS Issues
- Manage TLS from the Command Line

© 2026 - Hammerspace

- Reference: Client Requirements
- Reference: Supported third-party Backend Storage
- Reference: Certificate Trust Model

To encrypt the connection that Hammerspace uses for management sign-ins against Active Directory, see Securing the Connection to the Identity Provider. That setting is separate from NFS over TLS.

A.1.1. Important Considerations

Review the following before planning your TLS deployment.



This feature is supported as a **technology preview** and is not supported for use in production environments.

All-or-nothing enforcement. TLS is either required cluster-wide or forbidden cluster-wide. There is no mixed mode. All connected backend storage servers, Hammerspace cluster nodes, and clients must have TLS enabled when TLS is turned on. TLS can later be disabled.

Backup servers must also be TLS-enabled. When TLS is required, every external NFS storage server *and every backup server* the cluster touches must also be TLS-enabled. A non-TLS backup target **fails** once the cluster policy is set to require TLS, with the following error:

```
mount.nfs: access denied by server while mounting <ip>:/backup, error code: 32
```

This is a hard failure, not a degraded mode: when the cluster policy requires TLS, mounting a non-TLS backup target is denied. Identify and remediate any non-TLS backup servers before enabling TLS.



Best-effort handling applies only as a documented nuance for backend storage that genuinely cannot participate in mutual TLS. For such storage, backup operations to that device may not be fully encrypted. This is the exception, not the default — when the cluster policy requires TLS, a reachable non-TLS backup target is denied as shown above.

Enabling TLS is disruptive. Enabling or disabling TLS requires a planned maintenance window. The change first reconfigures and restarts the metadata server on the Anvil, then reconfigures each Anvil and DSX node. While it runs, NFS I/O over existing mounts can fail with `access denied by server`. The process temporarily takes the cluster, clients, and backend storage volumes offline for normal data operations. After TLS is enabled, all NFS clients must be remounted.

RDMA is not available while TLS is required. TLS for NFS runs over TCP only. If clients mount over NFS/RDMA, plan to move them to TCP before you enable TLS.

Certificate management is shared. Hammerspace automatically generates an internal root Certificate Authority (CA) and issues signed X.509 certificates to each cluster node. The customer's certificate authority (CA) remains responsible for the full lifecycle of externally-managed certificates, including monitoring expiration, issuing new certificates, and handling revocation. Hammerspace does not alert you before any certificate expires and does not renew certificates automatically, including the node certificates it issues, which are valid for 365 days. Check the expiration dates before you enable TLS. See Certificate Lifetimes and Expiration.

© 2026 - Hammerspace

Referral shares are not supported. TLS cannot be enabled on clusters that have NFS referral shares configured.

Performance overhead. TLS adds latency to every NFS operation. Metadata-intensive workloads near the cluster's saturation point see the largest effect. Large-file throughput with TLS has not been measured. Test with your workload before enabling TLS in production.

SMB and S3 client access is unaffected. From the client's perspective, TLS applies to NFS data paths only — SMB and S3 are configured independently, and clients using those protocols are functionally unaffected by whether NFS TLS is enabled or disabled. Internally, however, the data and metadata that the SMB and S3 services exchange with the cluster's portals travel over NFS and are therefore subject to the cluster's NFS TLS policy; that internal path carries the same performance overhead and is affected by any TLS misconfiguration.

Third-party storage support is limited. In Hammerspace 5.3, NFS over TLS has been validated with Linux NFS servers only. Before you plan a deployment that uses other backend storage, confirm NFS over TLS support with the storage vendor. When TLS is required, backend storage that cannot complete a mutual TLS handshake becomes unavailable. See Reference: Supported Third-Party Backend Storage.

Private key storage. Node private keys are stored unencrypted on the local drive to allow automated startup. These files are protected by the node's filesystem permissions.

A.1.2. Limitations in Hammerspace 5.3

The following constraints apply to the technology preview of NFS over TLS in Hammerspace 5.3.

- **Technology preview only** — not validated or supported for production environments.
- **All-or-nothing** — no mixed TLS/non-TLS cluster configurations; no per-share TLS.
- **No RDMA while TLS is required** — RPC-with-TLS (RFC 9289) runs over TCP only. While TLS is required, client access over NFS/RDMA fails. Clients that mount over RDMA must remount over TCP with `xprtsec=mtls`.
- **Referral shares are not supported** — NFS referral shares are not supported in a TLS-enabled cluster.
- **Certificate revocation not implemented** — Hammerspace does not support certificate revocation lists (CRL) or OCSP. To stop trusting a certificate that you uploaded, remove it from the trust store.
- **Removing a node does not revoke its certificate** — A node removed from the cluster keeps its certificate, its private key, and the Hammerspace root CA certificate, and the certificate remains valid until it expires. A host holding them can authenticate to the cluster as a Hammerspace node. Before a removed node leaves your control, reinstall it or wipe its system disks. If you cannot, contact Hammerspace Support. Every 5.3 node has a certificate whether or not TLS is enabled, so a node removed while TLS is off can still authenticate to the cluster if TLS is enabled later.
- **No certificate renewal or expiration alerts** — Hammerspace does not renew certificates automatically and raises no event before a certificate expires. Administrators cannot renew node certificates or replace the system CA. See Certificate Lifetimes and Expiration.
- **Cipher suites are fixed and cannot be configured** — `pd-protod` and `pd-di` do not use system defaults. They use TLS 1.3 with AES-128-GCM, AES-256-GCM, and ChaCha20-Poly1305. Only the kernel side — DSX `knfsd` and clients, through `tlshd` — goes through GnuTLS. There is no UI or CLI to customize cipher suites.

 © 2026 - Hammerspace

- **No per-user certificate authentication** — the transport is secured and the host is authenticated; user identity comes from the share's RPC security flavor.
- **No SmartNIC TLS offload** — TLS processing runs on host CPUs.
- **External movers receive no system-issued certificates** — and no PKI health monitoring or `PKI_*` events. See External Movers (DI and Cloud Mover Containers).
- **Third-party storage support is limited** — see Reference: Supported Third-Party Backend Storage.
- **Strict client OS requirements** — see Reference: Client Requirements.

A.1.3. Prerequisites

Before enabling TLS, verify that all components in your environment meet the following requirements.

Hammerspace Cluster

- Hammerspace 5.3 or later.
- The cluster must not have NFS referral shares configured.
- All cluster nodes must be reachable and healthy before starting the enablement process.

Linux Clients

Each Linux client that will mount NFS shares over TLS must meet the following requirements:

Component	Minimum version
Linux kernel	6.12
`nfs-utils`	2.5.4
`ktls-utils`	1.0.0-rc1
Enterprise Linux distribution	RHEL 10.1, Rocky Linux 10.1, or later

In addition:

- **Mutual (two-way) trust must be established between the client and Hammerspace.** The client must present an X.509 certificate that Hammerspace trusts — either issued directly to the client or signed by a certificate authority that Hammerspace trusts — and the Hammerspace root CA certificate (or the customer's own CA certificate) must be installed in the client's trust store. For how to establish this trust, see Reference: Certificate Trust Model.
- The `tlshd` service (part of `ktls-utils`) must be installed and configured on each client. `tlshd` manages the TLS handshake for the Linux NFS client and kernel.

Third-Party Backend Storage

- Backend storage must support NFS over TLS with mutual authentication. When TLS is required, Hammerspace mounts every backend storage volume with TLS, and a volume on storage that does not

© 2026 - Hammerspace

support it becomes unavailable. Enablement does not check the volumes for you: the **Volumes** step of the Enable TLS wizard shows their status, but does not stop you from continuing.

- Confirm TLS support with your storage vendor before proceeding. See Reference: Supported Third-Party Backend Storage for known-supported systems.

Backup and External Storage Servers

When TLS is required, every external NFS storage server *and every backup server* the cluster touches must also be TLS-enabled. A non-TLS backup target fails to mount once the cluster policy requires TLS, with the error:

```
mount.nfs: access denied by server while mounting <ip>:/backup, error code: 32
```

Verify TLS capability on all backup servers — not just primary backend storage — before starting the enablement process.

A.1.4. Workflow Overview

Enabling TLS on an Existing Cluster

Use this workflow when enabling TLS on a Hammerspace cluster that is already configured with backend storage and clients.

This workflow is fully disruptive to normal platform operations. Schedule this procedure during a maintenance window. While the operation is in progress, the Hammerspace cluster, clients, and backend storage volumes will be offline for normal data operations.

Prepare the storage systems and clients before you enable TLS. When TLS is enabled, Hammerspace refuses NFS connections that do not use TLS and mounts every backend storage volume with TLS. A backend storage volume whose storage system cannot complete the mutual TLS handshake becomes unavailable, and client mounts that do not use TLS stop working.

1. Prepare the backend storage systems. On each storage system that requires it, install the Hammerspace cluster root CA certificate, so that the storage system trusts Hammerspace. To download the certificate, go to **Administration › TLS**, select the **Cluster Root CA** tab, and click **Download Certificate**. Also obtain the root CA certificate that signed each storage system's own certificate; you add it to Hammerspace in step 3. See Reference: Supported Third-Party Backend Storage.
2. Prepare each Linux client. Make sure the client meets Reference: Client Requirements. Remount Clients gives the commands for each supported Linux distribution; do not remount the clients yet. On each client:
   - Install a client certificate and its private key. The certificate must be signed by a CA that Hammerspace will trust in step 3 (for example, your organization's CA), or signed by the cluster itself with `cert-csr-sign`. See Manage Certificates and CAs.
   - Add the Hammerspace cluster root CA certificate to the client's trust store.
   - Configure `tlshd`.
3. Add the root CAs to Hammerspace. Add the root CA certificate of every CA that signed a client certificate or

© 2026 - Hammerspace

a storage system certificate. Hammerspace trusts a client because its certificate chains to a CA in the Hammerspace trust store; you do not upload individual client certificates.

In the Enable TLS wizard, add root CA certificates on the **Certificates** tab. The wizard accepts root CA certificates only. To trust an intermediate CA, add it with the Admin CLI command `cert-add --trust-intermediate-ca` before you run the wizard, or include the intermediate certificate in each client's certificate file. See Manage Certificates and CAs and Reference: Certificate Trust Model.

Certificates that you add in the wizard are uploaded only when you click **Enable TLS** on its last tab. If you close the wizard before then, they are discarded. If you use the Management GUI, steps 3 and 4 are therefore one pass through the wizard.

4. Enable TLS. Complete the Enable TLS wizard and click **Enable TLS** on its last tab (**GFS**), or run `cluster-config --nfs-tls-required`. See Enable TLS Using the Management GUI or Enable or Disable TLS.
5. Remount each client. Unmount each Hammerspace share and mount it again with the `xprtsec=mtls` mount option. See Remount Clients.

If this cluster is part of a global file system (GFS), the wizard's **GFS** tab recommends enabling TLS on every site in the GFS configuration: enable it on this cluster first, then on the remote sites.

Enabling TLS on a New Cluster (Recommended)

If you are deploying a new Hammerspace cluster that will use TLS from the start, complete setup in this order to minimize disruption:

1. Install Hammerspace. Do not add third-party backing stores yet. See the *Hammerspace 5.3 Installation and Licensing Guide*.
2. Enable TLS using the procedure in this document.
3. Configure third-party backing stores and clients. See the *Hammerspace 5.3 Configuration Guide*.
4. Add or assimilate data from the configured backing stores. See the Assimilation Overview.

A.1.5. Enable TLS Using the Management GUI

This procedure is fully disruptive to normal platform operations. After TLS is enabled on the cluster, all clients must be remounted before data access can resume. Schedule this procedure during a planned maintenance window.

1. Log in to the Hammerspace Management GUI as an admin user.
2. Navigate to **Administration** › **TLS**.
3. Click the **NFS Transport Layer Security** toggle to open the Enable TLS wizard.
4. In the **Details** tab, review the information about tech preview support status and the impact of enabling TLS. Select the checkbox to confirm that you understand the risks and want to proceed, then click **Next**.
5. In the **Certificates** tab, review the table of trusted certificates. Hammerspace must trust the root CA that issued each client's and each storage server's certificate. In this step you can add root CA certificates only. Add intermediate CA certificates and individual self-signed certificates with `cert-add`, before or after the

© 2026 - Hammerspace

wizard.

a. Click **Add Trusted Certificate**.

b. In **Certificates File**, click **Upload** and select a **self-signed root CA certificate**. This step rejects intermediate CA certificates, CA-signed client certificates, and self-signed certificates that are not CA certificates — an intermediate CA certificate is refused with `A certificate was skipped because of error: Certificate is an intermediate CA and trustIntermediateCa is not true`, although the step's own hint text mentions intermediate certificates.

c. Click **Add** to add the certificate to the wizard's pending list. The certificate displays **Local (Not yet uploaded)** until the wizard completes.

d. Repeat for each root CA certificate that Hammerspace must trust.

When all certificates have been added, click **Next**.



Closing the wizard before completing the configuration removes any pending certificate from the list.

The table shows a delete icon on every certificate, including the Hammerspace System CA and the node certificates. Delete only certificates that you uploaded.

6. In the **Volumes** tab, review the list of storage volumes on which TLS will be enabled. All volumes must support TLS — fix or remove any volume shown as failed before continuing. Click **Next**.



You can download the Hammerspace cluster CA certificate from this tab (**Download Cluster Certificate**; the same button is on the **Certificates** tab). Install this certificate in the trust store of any backend storage system that needs to authenticate the Hammerspace cluster. Hammerspace does not use an intermediate CA, so you must add either the Hammerspace root CA certificate or the node certificate directly to the storage system's trust store.

7. In the **Mounts** tab, review the list of NFS mounts that will be affected by the TLS policy change. The list shows the cluster mounts and each DSX node's mounts for every share, and can span several pages. Click **Next**.

8. In the **Clients** tab, review the client IP addresses, ports, and protocols. The list includes the clients that currently have the cluster mounted; pNFS clients are listed as **NFS 4.2 Clients with outstanding layouts**. Each of these clients must be remounted after TLS is enabled. Click **Next**.

9. In the **GFS** tab, review the site names and management addresses for each remote site in your global file system deployment. This is the final step of the wizard, so there is no **Next** action here.

10. Click **Enable TLS** to begin the enablement process.

The cluster performs a series of background operations to transition all components to TLS. You can monitor progress in the Enable TLS dialog.

Enabling TLS is a cluster-wide operation. It can take a few minutes or more than 30 minutes: on each DSX node, Hammerspace restarts the node's data-portal mounts one after another, and a mount that is slow to come back is retried for up to 10 minutes. In testing, the change took from about 2½ minutes to about 35 minutes on clusters with one or two DSX nodes. Monitor progress in the Enable TLS dialog; the change is complete when

 © 2026 - Hammerspace

NFS Transport Layer Security shows **Enabled**.



While the change runs, transient `VOLUME_CLONE_TEST_FAILED` alerts ("Permission denied") on the storage volumes, a short `Down` state of the DSX data portals, and `HW_FRU_EVENT` notices (`Added hardware device /dev/sd⋯`, several per DSX node) are expected and clear on their own. Clients that are still mounted without TLS lose access about a minute after you click **Enable TLS**, when the metadata server restarts and before the change completes (`Permission denied`; the client kernel logs `RPC: server <address> requires stronger authentication`). Clients can mount with `xprtsec=mtls` from that point.

After the process finishes, proceed to Remount Clients.

A.1.6. Remount Clients

After TLS is enabled on the Hammerspace cluster, each NFS client must be fully unmounted and remounted with TLS mount options. A standard `mount -o remount` is not sufficient — switching from plaintext NFS to TLS requires a complete unmount and remount due to the fundamental change in the network transport layer.

Before remounting, confirm that certificate trust is configured in both directions: the client must trust the Hammerspace cluster's CA certificate, and Hammerspace must trust the client's certificate. See Reference: Certificate Trust Model. A client that mounts without both sides trusting each other fails the TLS handshake.

This section includes the following subsections:

- Remounting an Enterprise Linux 10.1 (or Later) Client
- Remounting an Ubuntu Client

Remounting an Enterprise Linux 10.1 (or Later) Client

These instructions apply to Red Hat Enterprise Linux 10.1, Rocky Linux 10.1, and compatible distributions.

On Enterprise Linux 10.1 (kernel 6.12.0-124), a mount that uses `xprtsec=mtls` succeeds and directory listings work, but every file read and write hangs. The client's pNFS data connections to the DSX nodes (port 3049) do not use TLS, and the DSX nodes refuse them. Until you run a kernel that fixes this, disable the client's pNFS flexible-files layout driver before you remount, as described under Prerequisites on this page. File I/O then goes through the TLS connection to the Anvil.

Prerequisites

Verify the following before remounting:

- The `nfs-utils` and `ktls-utils` packages are installed.
- The `tlshd` service is enabled and running:

```
sudo systemctl enable --now tlshd.service
```


© 2026 - Hammerspace

- The Hammerspace cluster CA certificate is trusted by the client. This is the issuer of every Anvil and DSX certificate (**Administration › TLS › Cluster Root CA › Download Certificate**); the client needs it whichever CA issued the client's own certificate:

```
sudo trust anchor /path/to/hammerspace-ca.crt
```

- The client has a valid X.509 certificate and private key, and `tlshd` is configured to use them. If the client certificate was issued by an intermediate CA, put the intermediate CA certificate after the client certificate in the `x509.certificate` file, or add the intermediate CA to the Hammerspace trust store with `cert-add --trust-intermediate-ca`. The `tlshd` configuration file at `/etc/tlshd.conf` must reference the correct certificate paths:

```
[authenticate.client]
x509.truststore= /etc/pki/tls/certs/ca-bundle.crt
x509.certificate= /etc/pki/hs-tls/nfs-pki/cert.pem
x509.private_key= /etc/pki/hs-tls/nfs-pki/key.pem
```



In `ktls-utils` 1.3.0 and later, the default configuration file is `/etc/tlshd/config`. If `/etc/tlshd.conf` exists, `tlshd` reads it instead. Edit whichever file your distribution uses, and restart `tlshd` afterward.

- On kernel 6.12.0-124 (Enterprise Linux 10.1), the pNFS flexible-files layout driver is disabled, so that file I/O uses the TLS connection to the Anvil instead of unencrypted connections to the DSX nodes. Create a file named `/etc/modprobe.d/hammerspace-no-pnfs.conf` that contains these two lines, then reboot the client:

```
blacklist nfs_layout_flexfiles
install nfs_layout_flexfiles /bin/false
```

After you remount, `grep pnfs= /proc/self/mountstats` shows `pnfs=not configured` for each Hammerspace mount. Reboot rather than unloading the module: a client that already hung on file I/O keeps the module in use until it restarts.

Procedure

1. If the share is currently mounted, stop any processes using it, then unmount:

```
sudo umount /mnt/your_mount_point
```



If the unmount fails because the mount point is busy, use `fuser -m /mnt/your_mount_point` to identify and stop the active processes before retrying.

2. Remount the share with the `xprtsec=mtls` option to require mutual TLS authentication:

© 2026 - Hammerspace

```
sudo mount -o xprtsec=mtls <cluster_ip>:/<export_path> /mnt/your_mount_point
```

3. To ensure the share remounts automatically with TLS after a system reboot, add or update the entry in `/etc/fstab`:

```
<cluster_ip>:/<export_path>  /mnt/your_mount_point  nfs  xprtsec=mtls,_netdev  0  0
```

   - `xprtsec=mtls` — Requires mutual TLS authentication for the NFS connection.
   - `_netdev` — Ensures the network is available before the system attempts to mount the share at boot time.

Verify the TLS Connection

After remounting, verify that the connection is using TLS:

- Check the connection status. Connections to the Anvil use port 2049. With a pNFS mount (NFSv4.2), data connections go directly to DSX nodes on port 3049. Check both:

```
sudo ss -tni | grep -E -A1 ':(2049|3049)\b'
```

  A connection that uses TLS shows `tcp-ulp-tls` in its details, for example `tcp-ulp-tls version: 1.3 cipher: aes-gcm-256`. Run the command as root: on some kernels the TLS details are omitted for other users.

  Data connections to port 3049 open when files are read or written, so run the check during file I/O. On kernel 6.12, `ss` shows the TLS details to non-root users as well. If a port 3049 connection does not show `tcp-ulp-tls`, or file I/O hangs, see Troubleshoot TLS Issues. A client whose pNFS layout driver is disabled (see Prerequisites on this page) has no port 3049 connections: all traffic uses the TLS connection on port 2049.

- Check the NFS mount statistics. Confirm that `xprtsec=mtls` appears in the reported mount options:

```
nfsstat -m
```

Remounting an Ubuntu Client



NFS over TLS requires **Linux kernel 6.12 or later**. Ubuntu system administrators should verify their kernel version before proceeding:

```
uname -r
```

Ubuntu 25.04 ships kernel 6.14 and meets this requirement out of the box. Ubuntu 24.04 LTS ships kernel 6.8 by default; administrators on 24.04 LTS should verify whether their system is running the Hardware Enablement

© 2026 - Hammerspace

(HWE) kernel, which tracks more recent upstream releases:

```
dpkg -l linux-image-generic-hwe-24.04
```

These instructions apply to Ubuntu 25.04 and later, and to Ubuntu 24.04 LTS with a kernel that meets the ≥ 6.12 requirement.

Prerequisites

Verify the following before remounting:

1. Install the required packages:

   ```
   sudo apt install nfs-common ktls-utils
   ```

   `nfs-common` provides the Ubuntu equivalent of `nfs-utils` (the `mount.nfs` and related utilities). `ktls-utils` provides `tlshd`, the TLS handshake daemon required by the Linux kernel NFS client.

   > Check the installed version with `tlshd --version` (or `journalctl -u tlshd`, which logs `Built from ktls-utils <version>`). Ubuntu 24.04 LTS packages `ktls-utils` 0.9, which is below the minimum in Reference: Client Requirements. With 0.9, `tlshd` can only verify the Hammerspace server certificate when the server address reverse-resolves to a name in that certificate, so mounts fail with `gnutls: Error in the certificate. (-43)` in the `tlshd` journal. Install a current `ktls-utils` release (built from source if your distribution does not package one) before continuing.

1. Install the Hammerspace cluster CA certificate in the Ubuntu trust store. This is the certificate that **Administration › TLS › Cluster Root CA › Download Certificate** provides; it is the issuer of every Anvil and DSX certificate, so the client must trust it whichever CA issued the client's own certificate:

   ```
   sudo cp hammerspace-ca.crt /usr/local/share/ca-certificates/hammerspace-ca.crt
   sudo update-ca-certificates
   ```

   The `update-ca-certificates` command adds the certificate to `/etc/ssl/certs/ca-certificates.crt`, which `tlshd` uses as its default trust store.

2. Configure `tlshd` with the client certificate and private key. Edit `/etc/tlshd.conf` and populate the `[authenticate.client]` section:

   ```
   [authenticate.client]
   x509.truststore= /etc/ssl/certs/ca-certificates.crt
   x509.certificate= /etc/ssl/hs-tls/cert.pem
   x509.private_key= /etc/ssl/private/hs-tls/key.pem
   ```

   > In `ktls-utils` 1.3.0 and later, the default configuration file is `/etc/tlshd/config`. If

© 2026 - Hammerspace

> `/etc/tlshd.conf` exists, `tlshd` reads it instead and logs `Please relocate /etc/tlshd.conf to /etc/tlshd/config`. Edit whichever file your distribution uses, and restart `tlshd` afterward.

Replace the certificate and key paths with the actual locations of your client certificate and private key. If the client certificate was issued by an intermediate CA, put the intermediate CA certificate after the client certificate in the same file (`x509.certificate`), or add the intermediate CA to the Hammerspace trust store with `cert-add --trust-intermediate-ca`; Hammerspace must be able to build the chain from what the client presents to a CA it trusts. Restrict access to the private key file:

```
sudo chmod 600 /etc/ssl/private/hs-tls/key.pem
```

3. Enable and start the `tlshd` service:

```
sudo systemctl enable --now tlshd.service
```

4. Verify that `tlshd` is running without errors:

```
sudo journalctl -u tlshd --no-pager -n 20
```

Confirm there are no errors related to certificate loading or TLS handshake failures before proceeding.

Procedure

1. If the share is currently mounted, stop any processes using it, then unmount:

```
sudo umount /mnt/your_mount_point
```

> If the unmount fails because the mount point is busy, use `fuser -m /mnt/your_mount_point` to identify and stop the active processes before retrying.

2. Remount the share with the `xprtsec=mtls` option to require mutual TLS authentication:

```
sudo mount -o xprtsec=mtls <cluster_ip>:/<export_path> /mnt/your_mount_point
```

3. To ensure the share remounts automatically with TLS after a system reboot, add or update the entry in `/etc/fstab`:

```
<cluster_ip>:/<export_path>  /mnt/your_mount_point  nfs  xprtsec=mtls,_netdev  0  0
```

   - `xprtsec=mtls` — Requires mutual TLS authentication for the NFS connection.

© 2026 - Hammerspace

- `_netdev` — Ensures the network is available before the system attempts to mount the share at boot time.

Verify the TLS Connection

After remounting, verify that the connection is using TLS:

- Check the connection status. Connections to the Anvil use port 2049. With a pNFS mount (NFSv4.2), data connections go directly to DSX nodes on port 3049. Check both:

  ```
  sudo ss -tni | grep -E -A1 ':(2049|3049)\b'
  ```

  A connection that uses TLS shows `tcp-ulp-tls` in its details, for example `tcp-ulp-tls version: 1.3 cipher: aes-gcm-256`. Run the command as root: on some kernels the TLS details are omitted for other users.

- Check the NFS mount statistics. Confirm that `xprtsec=mtls` appears in the reported mount options:

  ```
  nfsstat -m
  ```

- Review `tlshd` logs to confirm the TLS handshake completed successfully:

  ```
  sudo journalctl -u tlshd --no-pager -n 20
  ```

A.1.7. External Movers (DI and Cloud Mover Containers)

External movers — the Data Instantiator (`pd-di`) and Cloud Mover containers — move data between Hammerspace and external storage. To participate in a TLS-required cluster, an external mover must present its own PKI material.

External movers are handled differently from Anvil and DSX nodes. The cluster does not issue them certificates, does not monitor their PKI health, and does not restart them when TLS is enabled on the cluster. Each of those is an administrator responsibility.

Supplying PKI Material to External Movers

The cluster issues certificates only to Anvil and DSX nodes. Nothing is distributed to a mover container.

Supply the mover's certificate and key yourself. The certificate must be signed by a certificate authority the cluster trusts, uploaded to the cluster as a trusted CA, and must carry the mover's IP addresses as subject alternative names.

External movers receive no PKI health checking and raise no `PKI_*` events, so a mover with missing or invalid PKI material is not reported by cluster monitoring.

The Cloud Mover checks that the certificate presented by each Hammerspace node or storage server it

 © 2026 - Hammerspace

connects to includes the IP address it connects to as a subject alternative name. Certificates that Hammerspace issues to Anvil and DSX nodes include their IP addresses. For third-party storage, make sure the storage server's certificate does too.

Creating the tlshd Configuration File

`pd-di` and the Cloud Mover do not use `tlshd`. They perform their own TLS handshakes and read `/etc/tlshd.conf` only to locate the certificate, key, and trust store. Create the file whether or not `tlshd` is installed in the container. `tlshd` matters only if something in the container performs kernel NFS mounts.

Create `/etc/tlshd.conf` with the following two sections — one for the container acting as a TLS client, one for it acting as a TLS server — and point each at the mover's PKI files:

```
[authenticate.client]
x509.truststore = /etc/pki/tls/certs/ca-bundle.crt
x509.certificate = /etc/pki/hs-tls/nfs-pki/cert.pem
x509.private_key = /etc/pki/hs-tls/nfs-pki/key.pem

[authenticate.server]
x509.truststore = /etc/pki/tls/certs/ca-bundle.crt
x509.certificate = /etc/pki/hs-tls/nfs-pki/cert.pem
x509.private_key = /etc/pki/hs-tls/nfs-pki/key.pem
```

Set all three keys in both sections. `pd-di` does not report a key that is missing or misspelled. It treats the key as not configured and still logs that it is TLS-capable, but its PKI material is incomplete, and TLS connections that need the missing file fail.

The private key must be an unencrypted PEM file; `pd-di` cannot read an encrypted key. Restrict access to the key file, for example with mode `600`, as Hammerspace does on its own nodes.

Adding the Required Trust Anchors

The container must trust the Hammerspace root CA, and the certificate authority of any third-party storage it moves data to or from.

1. Copy `/etc/pki/hs-tls/cacerts/hs-root-cacert.pem` from any Anvil node, together with the CA certificates of any third-party storage involved, into `/etc/pki/ca-trust/source/anchors/` inside the container.
2. Run `update-ca-trust extract`.



Pointing `x509.truststore` at a PEM file is not sufficient. The Cloud Mover ignores `x509.truststore` and reads the Java trust store that `update-ca-trust extract` generates.

Restarting an External Mover

`pd-di` reads `/etc/tlshd.conf` and checks the PKI files only at startup. If they are missing or invalid when it starts, the mover remains TLS-incapable until it is restarted. It logs `DI is TLS-incapable (PKI material unavailable)` and refuses mobilities to TLS-required storage, which DME then reschedules to another DI.

© 2026 - Hammerspace

Where possible, put the PKI files in place and restart the mover container before enabling TLS on the cluster.

On DSX nodes the cluster restarts `pd-di` when TLS is enabled. Nothing does this for an external mover, so the restart is manual.

Restart `pd-di` again after renewing its certificate or adding a trusted CA, because its outbound TLS context is loaded once at startup.

A.1.8. Disable NFS over TLS

You can disable TLS encryption at any time. Disabling TLS is also a disruptive operation that requires remounting all clients.

Disabling TLS requires a maintenance window. After TLS is disabled, all clients must be remounted without TLS mount options before data access can resume.

1. Log in to the Management GUI and navigate to **Administration › TLS**.
2. Click the **NFS Transport Layer Security** switch. The **Disable TLS** dialog warns that all clients with TLS connections will be interrupted and may need to be remounted. Click **Disable TLS**.

   Disabling TLS is a cluster-wide operation. It usually takes 1–3 minutes; on a cluster where enabling TLS took about 30 minutes, disabling took under 2 minutes. Allow up to about 30 minutes on a large cluster.

1. After the process completes, remount each client using your standard NFS mount options, without the `xprtsec` option (or with `xprtsec=none`):

   Remount clients

   ```
   sudo umount /mnt/your_mount_point
   sudo mount <cluster_ip>:/<export_path> /mnt/your_mount_point
   ```

2. Update `/etc/fstab` to remove the `xprtsec=mtls` option from any persistent mount entries. After TLS is disabled, a mount that still requests `xprtsec=mtls` is refused (`mount.nfs: access denied by server`).

A.1.9. Certificate Lifetimes and Expiration

Every certificate that NFS over TLS relies on has a fixed lifetime. Hammerspace does not warn you before a certificate expires, and does not renew any certificate automatically, so you must track expiration dates yourself and act before the earliest one expires.

The most important dates are those of the node certificates. Hammerspace issues one to every Anvil and DSX node, they are valid for one year, and they exist even on a cluster that has never enabled TLS. When a node certificate expires on a cluster that requires TLS, new TLS connections to and from that node fail.

Certificate Lifetimes

Table 19. Certificate lifetimes

© 2026 - Hammerspace

Certificate	Lifetime	When it is issued, and how it is replaced
Hammerspace system CA (`SYSTEM_CA`)	10 years (3,650 days)	Created once, when the cluster first runs Hammerspace 5.3. Hammerspace 5.3 provides no way to replace it.
Node certificate (`SYSTEM_ISSUED`), one per Anvil and DSX node	365 days	Issued when the cluster first runs Hammerspace 5.3, whether by installation or by upgrade, and when a node is added to the cluster. Issued whether or not TLS is enabled. See Renewing Certificates.
Certificate signed from a CSR (`CSR_SIGNED_BY_CA`)	365 days	Issued when you sign a certificate signing request with `cert-csr-sign` or on the **Signed Certificates** tab. Supports client authentication only. Replace it by signing a new CSR.
Certificate or CA that you upload to the trust store (`UPLOADED_ROOT_CA`, `UPLOADED_INTERMEDIATE_CA`, `UPLOADED_PINNED_CERT`)	Set by the issuer	Managed by your certificate authority. Replace it by adding the new certificate and removing the old one.

Enabling or disabling TLS does not issue new node certificates. A node's existing certificate is used, so its expiration date is counted from the day it was issued, not from the day TLS was enabled. On a cluster that was upgraded to 5.3 and switched to require TLS later, the node certificates can expire well under a year after TLS was enabled.

Hammerspace does not reissue a node certificate as it approaches expiration.

Each certificate's **Not Before** date is set 30 days before the date it was issued, to allow for clock differences between systems. A node certificate's **Not After** date is therefore 395 days after its **Not Before** date.

The certificate that the Management GUI and REST API present over HTTPS is separate from NFS over TLS and has its own lifetime. See Server Certificate.

Checking Certificate Expiration Dates

Check the expiration dates of the node certificates as soon as the cluster is running Hammerspace 5.3, and again before you enable TLS. Record the earliest **Not After** date and plan to act well before it.

To list the node certificates with their validity dates, run the following command in the Admin CLI:

List node certificates

Command:

```
cert-list --origin SYSTEM_ISSUED --full
```

Expected output (one node's entry; the listing has one entry per node):

```
total 4
```


© 2026 - Hammerspace

```
ID:                     da5cc4d0-fe89-4605-b231-00e748a522d5
Type:                   Certificate
Origin:                 System issued
Usages:                 [TLS client, Service to service trust, TLS server]
Enabled:                Yes
Certificate:
                        Version:      3
                        Serial:       821442837118459164991035504149124979154724649 3
(0x17059035d28d8e89268e285ef547982e60c879d)
                        Algorithm:    SHA512withRSA
                        Issuer:       O=Hammerspace, OU=CLUSTER, OU=789c4318-baf9-44f8-b43c-
fb5923741c38, CN=Hammerspace System CA
                        Subject:      O=Hammerspace, OU=NODE, CN=b3282a04-a030-5816-b9ea-6c0452e8980e
                        Validity:
                          Not Before: 2026-08-28 18:36:49 UTC
                          Not After:  2027-09-27 18:36:49 UTC
                        Fingerprints:
                          SHA1:       3a:48:da:2:22:c2:84:77:8:9d:ad:cb:5f:60:66:6c:b4:86:85:e3
                          MD5:        5e:b3:e0:b9:50:eb:96:32:42:12:62:3a:d1:f2:18:1c
                          CRC32:      0x735b89cf

Parent:                 ID:                     80836289-5308-49eb-94d9-0c1c15ffee68
                        Type:                   Certificate
<Additional output removed for brevity>
```

The output includes one certificate for each Anvil and DSX node. For each certificate, the **Not After** line under **Validity** is the expiration date, in UTC. The certificate's **Subject** contains `OU=NODE` and `CN=<node ID>`, where `<node ID>` is the node's ID. To match an ID to a node name, compare it with the **ID** column of the `node-list` command.

To check the other certificates, run `cert-list --full` without the `--origin` option. The system CA, uploaded certificates, and certificates signed from a CSR with the `--track` option are listed with the same **Validity** fields. A certificate signed without `--track` is not kept by Hammerspace, so check its expiration date where it is installed.

In the Management GUI, go to **Administration › TLS** and select the **Trusted Certificates** tab. The tab lists the node certificates, the system CA, and the certificates you have uploaded, each with an **Expiration Date**. A node certificate's **Issued To** value begins with `CN=<node ID>,OU=NODE`; node certificates and the system CA have no delete action. The **Signed Certificates** tab lists the certificates signed from a CSR, and the **Cluster Root CA** tab shows the system CA's validity dates.

When a Certificate Expires

Every NFS over TLS connection begins with a TLS handshake in which both sides present a certificate and each side checks that the other's certificate is signed by a trusted CA and has not expired. A handshake that presents an expired certificate fails.

What an expired certificate breaks depends on which certificate it is:

© 2026 - Hammerspace

- **A node certificate, while the cluster requires TLS.** New TLS connections to and from the node fail. Clients cannot mount through that node, and connections between the node and the Anvil, other DSX nodes, and backend storage cannot be re-established. When every node certificate was issued on the same day, as on a cluster upgraded to 5.3, they all expire on the same day.
- **A node certificate, while TLS is forbidden.** NFS traffic does not use the certificate, so data access continues. The certificate must be replaced before you enable TLS.
- **A certificate signed from a CSR, or a certificate issued by a CA you uploaded.** The client, storage system, or external mover that presents it can no longer complete a TLS handshake with the cluster.

A Linux client reports a failed handshake, whatever the cause, as `access denied by server`. For how to tell an expired certificate from other causes, see Troubleshoot TLS Issues.

Hammerspace checks each node's certificate periodically. When it finds that a node certificate has expired, it raises the `PKI_NODE_CHECK_FAILED` event for that node, at Critical severity while TLS is required and at Warning severity while TLS is forbidden. The event is raised only after the certificate has expired. No event is raised before expiration, and no event is raised for CSR-signed or uploaded certificates.

Renewing Certificates

Table 20. How to replace each certificate before it expires

Certificate	How to replace it
Node certificate	Hammerspace 5.3 does not provide a way for administrators to renew node certificates. Contact Hammerspace Support well before the earliest node certificate expires.
Certificate signed from a CSR (clients only)	Generate a new CSR on the client, sign it with `cert-csr-sign` or on the **Signed Certificates** tab, and install the new certificate in place of the old one. If the old certificate was signed with `--track` or on the **Signed Certificates** tab, remove it from that tab or with `cert-remove`. Certificates signed from a CSR support client authentication only, so storage servers need a certificate from their own CA. See Manage Certificates and CAs.
Certificate or CA that you uploaded	Add the replacement before the old one expires: a root CA with `cert-add` or on the **Trusted Certificates** tab, and an intermediate CA or pinned certificate with `cert-add`, then remove the old one with `cert-remove`. Remove the old CA only after every client, storage system, and external mover that presents a certificate issued by it has a replacement.
System CA	Hammerspace 5.3 does not provide a way to replace the system CA. It is valid for 10 years from the day the cluster first ran Hammerspace 5.3.



A Data Instantiator in an external mover container reads its certificate and trusted CAs only when it starts. After you replace a mover's certificate or add a CA to its trust store, restart the mover. See External Movers (DI and Cloud Mover Containers).

© 2026 - Hammerspace

A.1.10. Troubleshoot TLS Issues

TLS-related Events

Hammerspace generates the following events when TLS configuration problems are detected. These events are visible in the Management GUI under **Events** and through any configured alert channels (SNMP, email, syslog).

Table 21. TLS events

Event	Description
`NFS_TLS_CONFIGURATION_UNHEALTHY`	Raised per node. The node's applied TLS policy — or `pd-protod's policy on the primary Anvil — does not match the cluster policy and the automatic re-apply did not correct it, or the check could not run.
`PKI_CA_CREATION_FAILED`	The Hammerspace system CA could not be created during initialization.
`PKI_TRUST_SYNC_FAILED`	CA trust material failed to distribute to one or more cluster nodes.
`PKI_NODE_CERT_CREATION_FAILED`	A per-node X.509 certificate could not be issued.
`PKI_NODE_CERT_SYNC_FAILED`	A per-node certificate could not be distributed to its target node.
`PKI_NODE_CHECK_FAILED`	The periodic health check of the node's certificate failed. The check covers the CA match, the certificate chain, expiry, and whether the key matches the certificate. It is raised at Critical severity while TLS is enabled and at Warning severity while TLS is off. The event's details name the check that failed. Hammerspace Support can run `hs-nfs-pki-check` on the node for more detail, and its output is captured in support bundles.



While TLS is being enabled, each DSX node raises `HW_FRU_EVENT` notices such as `Added hardware device /dev/sda` as it re-announces its own disks. These notices are expected and need no action.

Client Cannot Mount After TLS Is Enabled

If a client cannot mount an NFS share after TLS has been enabled on the cluster, verify the following:

1. Mounts made without TLS before the change are unmounted. A plaintext mount that is still in place loses access when TLS is enabled, and the client kernel logs `RPC: server <address> requires stronger authentication` about once a second until you unmount it:

© 2026 - Hammerspace

```
sudo umount /mnt/your_mount_point
```

2. The mount uses the `xprtsec=mtls` option:

```
mount -o xprtsec=mtls <server>:/<export> /<mountpoint>
```

3. The client mounts by an address that appears in the server's certificate. Hammerspace-issued certificates carry IP addresses only — node, cluster, and portal floating IPs — plus the internal name `data-cluster`. Mounting by a customer DNS name fails the handshake.
4. The client's `ktls-utils` is a current release. With `ktls-utils` 0.9 (the version Ubuntu 24.04 LTS packages) the mount fails with `gnutls: Error in the certificate. (-43)` in the `tlshd` journal (preceded by `Name or service not known`) unless the server address reverse-resolves to a name in the server certificate.
5. The `tlshd.service` is running on the client:

```
systemctl status tlshd.service
```

6. The Hammerspace root CA certificate (or your organization's CA certificate) is installed in the client's trust store. See Remount Clients for the trust anchor command.
7. The client's `/etc/tlshd.conf` names a certificate and a private key. The `ktls-utils` package ships a default configuration file, so a missing file is the less likely fault. The common fault is an `[authenticate.client]` section with no `x509.certificate` and no `x509.private_key`: the client then presents no certificate and the cluster rejects the handshake.
8. The client certificate is valid and has not expired. Check whatever path `x509.certificate` under `[authenticate.client]` in the client's `/etc/tlshd.conf` points at:

```
openssl x509 -in <path-from-tlshd.conf> -noout -dates
```



`/etc/pki/hs-tls/nfs-pki/cert.pem` is the certificate path on Hammerspace Anvil and DSX nodes. A client uses whatever path its own `tlshd.conf` specifies.

9. Restart `tlshd` after editing `/etc/tlshd.conf` or replacing certificate files:

```
systemctl restart tlshd.service
```

10. The client meets the minimum Linux kernel and package version requirements. See Reference: Client Requirements.
11. Hammerspace trusts the client's certificate. A client certificate that Hammerspace does not trust fails the handshake even when every item above is correct: the mount reports `mount.nfs: access denied by server` although the client's `tlshd` journal says `Handshake with ⋯ was successful`, because the server closes the connection after the handshake. A certificate issued by an intermediate CA needs the

intermediate certificate either in the client's certificate file or in the Hammerspace trust store. See Reference: Certificate Trust Model for which certificates each side must trust, and Remount Clients for configuring client certificate trust.

File I/O Hangs on a TLS Mount

Symptom: A client mounts a share with `xprtsec=mtls` and can list directories, but reads and writes never finish. Commands such as `dd` or `cp` hang, and `dmesg` shows no errors. Small writes that appear to succeed leave files of 0 bytes on the share.

Cause: The client's pNFS data connections to the DSX nodes (port 3049) do not use TLS, so the DSX nodes refuse them and the client retries indefinitely. This happens on Enterprise Linux 10.1 (kernel 6.12.0-124).

To confirm, run these commands on the client while the I/O is hung:

```
sudo ss -tn state time-wait '( dport = :3049 )' | wc -l
grep -E 'pnfs=|LAYOUTGET' /proc/self/mountstats
```

Hundreds of `TIME-WAIT` connections to port 3049, `pnfs=LAYOUT_FLEX_FILES`, and a `LAYOUTGET` line with a large error count indicate this problem.

Resolution:

1. Disable the pNFS flexible-files layout driver on the client, as described in the prerequisites of Remounting an Enterprise Linux 10.1 (or Later) Client.
2. Reboot the client. The hung mounts keep the driver loaded, so unmounting alone does not clear the problem.
3. Remount the share with `xprtsec=mtls`, and check that `grep pnfs= /proc/self/mountstats` shows `pnfs=not configured`.

File I/O then goes through the client's TLS connection to the Anvil.

Backups Fail with "access denied by server"

If a backup mount fails after TLS is required on the cluster with the error:

```
mount.nfs: access denied by server while mounting <ip>:/backup, error code: 32
```

the TLS handshake failed. The Linux client reports every failed TLS handshake with this message, so the cause is not necessarily that the backup server has TLS disabled.

The handshake is mutual, so check both directions:

- The backup server must trust the Hammerspace root certificate authority. The Anvil presents its node certificate to the backup server.

© 2026 - Hammerspace

- The cluster must trust the backup server's certificate. That certificate must also carry the IP address configured for the backup target as a subject alternative name.

Diagnosing TLS Connectivity to a Storage Node

Hammerspace provides a diagnostic utility, `volume_config_test`, that tests TLS connectivity to a specific storage node. Run it on an Anvil or DSX node.

```
volume_config_test --mtls --ip <node_ip> --rootfh <root_filehandle> --node-uuid <node_uuid>
```

`--mtls`

Uses the client certificate from the `/etc/tlshd.conf` of the node you run the command on.

`--node-uuid`

The UUID of the node you run the command **on**, not the UUID of the storage node being tested. The tool uses it to name the test directory it creates on the volume.

`--port`

Required only for a DSX volume, which uses port 3049. Third-party NAS uses the default of 2049, so omit `--port` for third-party NAS.

`--src-ip`

Optional.

External Mover Refuses TLS-Required Mobilities

If mobilities to TLS-required storage are not running on a Data Instantiator, check the `pd-di` log for:

```
DI is TLS-incapable (PKI material unavailable)
```

This means `pd-di` did not find valid PKI material when it started, so it refuses mobilities to TLS-required storage. DME reschedules those mobilities to another DI.

`pd-di` checks its PKI files only at startup, so putting the files in place is not enough on its own — the mover must be restarted. See External Movers (DI and Cloud Mover Containers).

Local tlshd.conf Edits Lost After a Software Update

Every software update copies the shipped `tlshd.conf` over `/etc/tlshd.conf` on Anvil and DSX nodes. Any local edits to that file are lost.

This most often shows up as `tlshd` debug logging that stops working after an update. Reapply local changes to `/etc/tlshd.conf` after each software update, and restart `tlshd` on the node afterwards.

© 2026 - Hammerspace

Spurious TLS-unhealthy Alert During a Software Update

You may see an `NFS_TLS_CONFIGURATION_UNHEALTHY` event fire during a cluster software update, with an accompanying error code 4057 ("prohibited during software update"), even if TLS was never configured on the cluster. This is a known false positive. It clears on its own once the update completes and does not indicate an actual TLS problem.

A.1.11. Manage TLS from the Command Line

The Hammerspace Admin CLI provides commands for all TLS certificate management tasks. Connect to the cluster using SSH:

```
ssh admin@<cluster_management_ip>
```

This section includes the following subsections:

- Enable or Disable TLS
- Manage Certificates and CAs

Enable or Disable TLS

Enabling or disabling TLS from the CLI carries the same disruption and client remount requirements as doing so from the GUI. See Workflow Overview before running either command.

TLS is enabled and disabled cluster-wide using the `cluster-config` command.

To **enable** TLS and require mutual authentication on all NFS connections:

```
cluster-config --nfs-tls-required
```

To **disable** TLS and revert all NFS connections to plaintext:

```
cluster-config --nfs-tls-forbidden
```

Both commands start a background task named `cluster-update` and, by default, wait up to 5 minutes for it to finish:

- If the task finishes within 5 minutes, the command prints the cluster configuration.
- If the task is still running after 5 minutes, the command exits with the error `cluster-config: Task-based operation still running after 5 minutes. To monitor progress, use: task-list --id <task-id>`. The task keeps running.
- If the task fails, stops, or is cancelled, the command prints `Task failed.` followed by the task's details.

To make the command wait for the task however long it takes, add `--no-timeout`:

© 2026 - Hammerspace

```
cluster-config --nfs-tls-required --no-timeout
```

To follow a task that is still running, run `task-list --id <task-id>`, or `task-list --name cluster-update` to list the cluster update tasks. The change is complete when the task's status is `COMPLETED` and its status message is `Updated: [nfsTransportPolicy -> TLS_REQUIRED_WITH_MUTUAL_AUTH]` (or `Updated: [nfsTransportPolicy -> TLS_FORBIDDEN]`).

A policy change can take a few minutes or more than 30 minutes. Hammerspace configures the Anvil and DSX nodes one at a time, and on each DSX node it restarts the node's internal mounts of the Hammerspace file system (the mounts that its NFS, SMB, and S3 data services use) one after another. A mount that is slow to come back is retried for up to 10 minutes. In testing, enabling TLS took from about 2½ minutes to about 35 minutes on clusters with one or two DSX nodes.

While the change runs, the task's status message shows its progress, for example `Updating NFS transport policy: Configuring metadata server`, `Updating NFS transport policy: Restarting metadata server`, and `Updating NFS transport policy: Configuring product nodes`. Hammerspace makes up to three attempts to configure any node that fails before the task fails.

To return immediately without waiting, add `--async`, and monitor the task separately:

```
cluster-config --nfs-tls-required --async
```

The command returns at once with `Process started. To monitor progress, use: task-list --id <task-id>`.

To check the current TLS policy, run `cluster-config --view` (or `cluster-config` with no options) and find the **NFS transport policy** line:

```
cluster-config --view
```

The line reads `TLS required with mutual auth` when TLS is enabled and `TLS forbidden` when it is disabled. While a policy change is still running, the line still shows the previous policy; it changes only after the `cluster-update` task completes.



The `cluster-config` command also includes options for managing the **management plane** HTTPS certificate used by the GUI and REST API (`--server-certificate-chain`, `--server-private-key`, `--server-certificate-reset`). These options are unrelated to NFS over TLS and are not covered in this document. Run them separately: `cluster-config` does not accept them in the same command as `--nfs-tls-required` or `--nfs-tls-forbidden`.

Manage Certificates and CAs

The following commands manage the certificates and certificate authorities used for NFS TLS mutual authentication.

© 2026 - Hammerspace

List the System Root CA

Display the Hammerspace-generated root CA certificate(s). The active CA is used to sign all internal node certificates:

```
cert-ca-list
```

To list a specific CA by role:

```
cert-ca-list --role SYSTEM_ACTIVE
cert-ca-list --role SYSTEM_OLD
```

`SYSTEM_ACTIVE` is the CA that signs node certificates and certificates signed from a CSR. The `SYSTEM_OLD` role is reserved for a CA that has been replaced. Hammerspace 5.3 provides no way to replace the system CA, so `cert-ca-list --role SYSTEM_OLD` normally returns nothing.

List All Registered Certificates

- Display all certificates known to the system:

  Display certificates

  ```
  cert-list
  ```

- For detailed output including validity dates and subject information:

  Display detailed output

  ```
  cert-list --full
  ```

- To filter by how a certificate was added to the system, use the `--origin` option:

  Certificate origin

  ```
  cert-list --origin <origin>
  ```

  Table 22. Origin values

Origin value	Description
`SYSTEM_CA`	The Hammerspace root CA certificate, generated automatically at cluster startup.
`SYSTEM_ISSUED`	Per-node X.509 certificates signed by the Hammerspace system CA and distributed to each product node.

© 2026 - Hammerspace

`CSR_SIGNED_BY_CA`	Certificates generated by signing a user-supplied CSR with the Hammerspace system CA (see `cert-csr-sign`).
`UPLOADED_ROOT_CA`	A self-signed CA certificate uploaded by an administrator to serve as a trust anchor, extending the trust store.
`UPLOADED_INTERMEDIATE_CA`	A CA-signed (non-self-signed) intermediate CA certificate uploaded by an administrator with `--trust-intermediate-ca`.
`UPLOADED_PINNED_CERT`	An individual self-signed certificate uploaded by an administrator and pinned directly to the trust store.

- To retrieve a specific certificate by ID:

  Retrieve certificate using ID

  ```
  cert-list --id <certificate_id>
  ```

Add a Trusted CA Certificate Using the Management GUI

Add a CA certificate to the Hammerspace trust store when Hammerspace must trust certificates that your organization's CA issued: to clients and storage servers that connect with NFS over TLS, to external movers, or to an LDAP or Active Directory server that Hammerspace connects to over LDAPS or StartTLS. You do not need to add the Hammerspace system CA; it is trusted automatically.

1. Navigate to **Administration** › **TLS** and select the **Trusted Certificates** tab.
2. Click **Add Trusted Certificate**.
3. In **Certificates File**, select the PEM certificate file. Accepted file extensions are `.pem`, `.crt`, `.cer`, and `.csr`.
4. Click **Add**.

   Each certificate that is added is reported as `Certificate <serial number> added successfully.` A certificate that cannot be added is listed with the reason.

In Hammerspace 5.3, the Management GUI adds root (self-signed) CA certificates only. To add an intermediate CA certificate or a self-signed certificate that is not a CA, use the `cert-add` command described in the Add a Certificate or CA to the Trust Store section of this page.

The certificate is listed on the **Trusted Certificates** tab, with its expiration date. To remove it later, use the delete action in its row and confirm in the **Delete Trusted Certificate** dialog, or use `cert-remove`.

Add a Certificate or CA to the Trust Store

Upload an external CA certificate, a certificate chain, or an individual client or storage certificate to the Hammerspace trust store. The `--pem` option accepts PEM-format content:

© 2026 - Hammerspace

```
cert-add --pem "<pem_content>"
```

By default, `cert-add` accepts only root CA certificates. Use the following options when adding other certificate types:

```
# Add an intermediate CA certificate (the PEM holds the intermediate certificate followed by its root)
cert-add --pem "<pem_content>" --trust-intermediate-ca

# Add a self-signed non-CA certificate (e.g. a pinned server cert)
cert-add --pem "<pem_content>" --trust-self-signed-certificate
```



An intermediate CA certificate is verified against its issuer in the same `<pem_content>`, so give the intermediate certificate and its root together, intermediate first. An intermediate given on its own is rejected with `Certificate signature verification failed`, even when its root is already in the trust store. If the root is already there, it is reported as `Certificate already exists` and the intermediate is still added.



You do not need to add the Hammerspace system root CA using `cert-add` — it is generated and distributed automatically. Use `cert-add` for certificates issued by your organization's CA or by external parties.



`cert-add` accepts CA certificates (self-signed root CAs, or CA-signed intermediate CAs with `--trust-intermediate-ca`) and self-signed non-CA certificates (with `--trust-self-signed-certificate`). A CA-signed leaf (end-entity) certificate — one that is neither a CA itself nor self-signed — is rejected: `Certificate is not a CA or self-signed.`

Sign a CSR with the Hammerspace System CA

If a client or storage system does not have its own certificate authority, you can sign its Certificate Signing Request (CSR) using the Hammerspace system CA. Hammerspace signs the request and, with `--track`, keeps the signed certificate so that you can download it for the client or storage system to install. A certificate signed by the system CA is trusted by the cluster without any further trust-store change.



Include at least one IP address in the Subject Alternative Name (SAN) field of the CSR. `cert-csr-sign` does not enforce this — it signs the CSR regardless of SAN contents — but clients and Hammerspace nodes performing TLS validation generally match against the SAN, so a certificate without an IP SAN may fail validation elsewhere even though it signs successfully.

```
cert-csr-sign --pem "<csr_pem_content>"
```

To add the signed certificate to the Hammerspace certificate inventory for expiration tracking:

```
cert-csr-sign --pem "<csr_pem_content>" --track
```


© 2026 - Hammerspace

The Admin CLI prints the signed certificate's details (issuer, subject, validity, fingerprints) but not the certificate itself. Always use `--track`: it adds the signed certificate to the **Signed Certificates** tab, from which you download it (it is also returned as `certPem` by `GET /pki-certificates` in the REST API). A certificate signed without `--track` cannot be retrieved afterwards.

The certificate's expiration date is shown on the **Signed Certificates** tab. A certificate signed from a CSR is valid for 365 days. Hammerspace does not alert you before it expires. See Certificate Lifetimes and Expiration.

Create a Self-Signed Certificate for Labs and POCs

Remove a Certificate from the Trust Store

Remove an uploaded CA certificate or pinned certificate from the trust store. First retrieve the certificate ID using `cert-list`, then pass it to `cert-remove`:

```
cert-list --full
cert-remove --id <certificate_id>
```

`SYSTEM_CA` and `SYSTEM_ISSUED` certificates are managed by Hammerspace and cannot be removed. `cert-remove` refuses them with the message `Cannot delete certificate with origin: <origin>`.

A.1.12. Reference: Client Requirements

Table 23. Client requirement for TLS

Component	Minimum version	Notes
Linux kernel	6.12	Required for RFC 9289 support via the kernel NFS client and knfsd with `tlshd`.
`nfs-utils`	2.5.4	TLS-aware NFS userspace utilities.
`ktls-utils`	1.0.0-rc1	Provides `tlshd`, the TLS handshake daemon required by the kernel NFS client.
RHEL / Rocky Linux	10.1	Required for NFSv4.2 over TLS. NFSv3 over TLS may work on some earlier distributions with the correct kernel and packages.
Ubuntu	TBD	Minimum version to be confirmed.

Supported NFS versions over TLS: NFSv4.1 and NFSv4.2 to the cluster (the Anvil), and NFSv3 and NFSv4.2 to DSX data-portal addresses. The Anvil does not serve NFSv4.0 or NFSv3, with or without TLS. With NFSv3, the

© 2026 - Hammerspace

MOUNT, NLM (locking), and NSM protocols are not encrypted.

Supported TLS versions: TLS 1.2 and TLS 1.3. TLS 1.3 is preferred when both the client and server support it. TLS 1.2 is available for environments with clients that do not yet support TLS 1.3.

A.1.13. Reference: Supported Third-Party Backend Storage

Hammerspace 5.3 has been validated with NFS over TLS against Linux NFS servers only.

Vendor	Product	NFS version	TLS version
Linux	Any Linux NFS server whose kernel and `ktls-utils` meet the minimums in Reference: Client Requirements	NFSv3, NFSv4.1, NFSv4.2	TLS 1.2, TLS 1.3

Before you plan a TLS deployment that uses other backend storage, confirm with the storage vendor that the product and release you run support NFS over TLS (RFC 9289) with mutual authentication, and test it.

When TLS is required, Hammerspace mounts every backend storage volume with TLS, and a volume whose storage cannot complete a mutual TLS handshake becomes unavailable. This also applies to backup servers. Verify TLS support with all storage vendors **and backup-server vendors** before starting a TLS deployment.

A.1.14. Reference: Certificate Trust Model

Hammerspace uses X.509 certificates for mutual authentication between all NFS endpoints. Understanding the trust model helps administrators plan their certificate strategy.

Hammerspace system CA. At startup, Hammerspace generates a self-signed root CA certificate for the cluster. This CA is used to sign per-node certificates for all product nodes. The system CA has a validity period of 10 years. Administrators can download the system CA certificate to install as a trust anchor on clients and storage systems. During enablement, click **Download Cluster Certificate** on the **Certificates** or **Volumes** step of the Enable TLS wizard. At any other time, go to **Administration › TLS**, select the **Cluster Root CA** tab, and click **Download Certificate**.

Per-node certificates. Each product node receives a unique X.509 certificate signed by the system CA. Certificates include the node's IP addresses in the Subject Alternative Name (SAN) field. Per-node certificates are valid for 365 days from issue. A node's certificate is issued when the cluster first runs Hammerspace 5.3 or when the node is added, whether or not TLS is enabled. Hammerspace re-issues a node's certificate when the node's IP addresses change or the system CA changes; it does not renew certificates as they approach expiry. See Limitations in Hammerspace 5.3.

Uploaded CA certificates. Administrators can upload their organization's CA certificate (or certificate chain) to extend Hammerspace's trust store. Uploaded CAs are used for verifying incoming client and storage certificates — they are **not** used to sign internal Hammerspace node certificates.

Signing user-supplied CSRs. Clients and storage systems that do not have their own CA can submit a Certificate Signing Request (CSR) to Hammerspace. Hammerspace signs the CSR with its system CA and returns the signed certificate. The CSR must include at least one IP address in the SAN field.

© 2026 - Hammerspace



A certificate that Hammerspace issues from a user-supplied CSR supports client authentication (`clientAuth`) only. Even when the CSR requests both `clientAuth` and `serverAuth`, the returned certificate contains `clientAuth`.

These certificates cannot be used for a storage-server endpoint that must present a server-authentication certificate. For those endpoints, obtain a certificate from a CA that can issue `serverAuth`.



Customers managing multiple Hammerspace clusters without an external CA can designate one cluster as the master signing authority. Upload the master cluster's root CA certificate to all other clusters to establish a unified trust chain across the environment.

© 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 24. 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	ERROR	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
NAME_SERVICE_UNHEALTHY	WARNING	Name Service "{1}" (domain "{2}") is unhealthy
SSSD_CONFIG_SYNC_FAILED	WARNING	Failed to synchronize SSSD configuration to node "{1}"
KERBEROS_REQUIRED_NFS_SPNS_MISSING	ALERT	NFS SPNs required for Kerberos are missing. Use the AD configuration to re-create NFS SPNs or visit the Active Directory page in the GUI.
KERBEROS_USER_MAPPING_NON FUNCTIONAL	INFORMATIONAL	Share "{0}" is configured for Kerberos, but AD LDAP Schema is not configured for user-mapping. Set the AD LDAP Schema to RFC2307 or RFC2307bis to enable user-mapping.

© 2026 - Hammerspace

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.
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. For more in-depth definitions of terms relevant to Hammerspace Global File Systems and data orchestration, see the Dictionary of Terms on the Hammerspace web site.

Table 25. 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:

- {kb-assimilation-url}[{kb-assimilation-title}] [Solution Guide]: An overview of a typical data migration with Hammerspace using assimilation and data orchestration.
- {kb-objectives-url}[{kb-objectives-title}] [Solution Guide]: How to configure Hammerspace objectives.
- {kb-hstk-url}[{kb-hstk-title}] [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