S3 Multitenancy Guide

Published 2026-11-20

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

HAMMERSPACE

Table of Contents

About This Documentation 1
1. Architecture 2
1.1. Hammerspace Big Picture 2
1.2. Hammerspace Multitenant S3 Building Blocks 3
1.2.1. Tenants 3
1.2.2. S3 Clients 4
1.2.3. Load Balancer 4
1.2.4. Anvil Nodes 4
1.2.5. DSX Nodes 4
1.2.6. S3 Service 4
1.2.7. S3 Server 4
1.2.8. Storage Volumes 5
1.2.9. Data Orchestration 5
2. Tenant Isolation Topics 6
2.1. Administration 6
2.2. S3 Client Access 6
2.3. S3 Client Authentication 7
2.4. Traffic Isolation 8
2.5. Storage Isolation 9
2.5.1. How Hammerspace Stores Data 10
2.5.2. Storage Capacity Quotas 12
3. Deployment Examples 13
3.1. Prepare for S3 Configuration 13
3.2. Prerequisites 13
4. Example 1: Two Tenants, Shared Resources 15
4.1. Hammerspace Architecture 15
4.2. Configuration Steps 16
5. Example 2: Two Tenants, Isolated Resources 33
5.1. Hammerspace Architecture 33
5.2. Configuration Steps 33
Appendices **48**
Appendix A: Appendix: Determining Umask Values 49
A.1. Unix Permissions - Octal Values 49
A.2. Creating an S3 Umask Value for a Local Identity Provider 49

About This Documentation

Hammerspace is a sophisticated, highly flexible software-defined data platform. It has many capabilities that can be configured to meet a wide range of use-case requirements. These include the ability to act as an object storage target for clients using the S3 protocol.

This S3 support is native, meaning S3 is truly an equal peer with the other client protocols, NFS and SMB. No gateway, shim, cache, or translation layer is required. S3 services are also architected to support multitenancy, meaning that a single Hammerspace implementation can provide object storage services to multiple user populations that are separate from - and unaware of - each other.

This solution guide focuses on the multitenant capabilities of Hammerspace's S3 object storage services. After reviewing the architecture and various tenant isolation techniques, several step-by-step configuration examples are provided.



Although the screenshots shown represent the v5.2 GUI, this guide applies to Hammerspace versions 5.1 and later.

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

Chapter 1. Architecture

To understand how S3 multitenancy works in Hammerspace, it is necessary to understand a little bit about Hammerspace's overall architecture. Keep in mind this document is focused on S3, so not every facet of Hammerspace is covered. For a more thorough treatment of Hammerspace's capabilities, refer to the Hammerspace Technology White Paper on hammerspace.com and the product documentation.

1.1. Hammerspace Big Picture

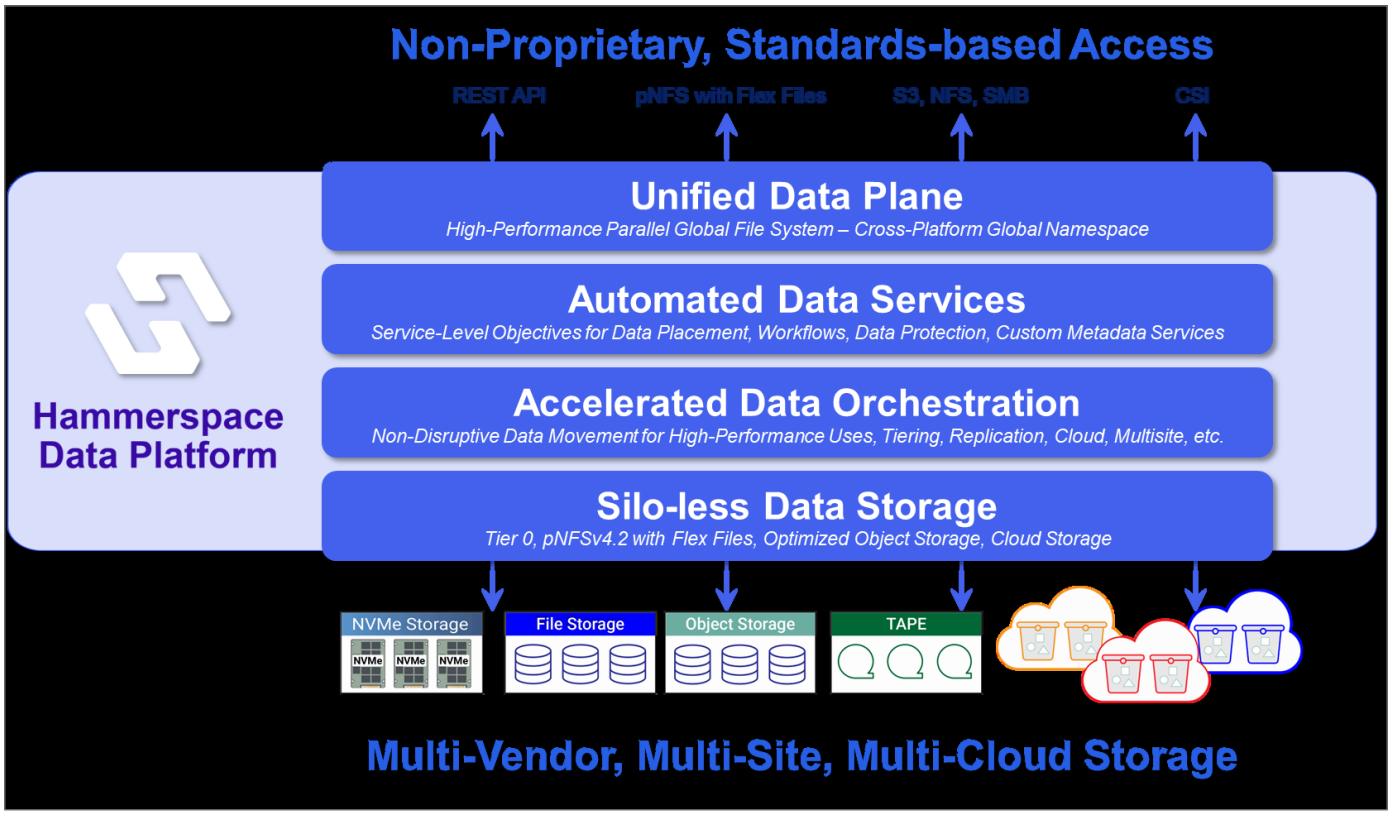


Figure 1. Hammerspace unifies storage into a global namespace

Hammerspace is software that unifies storage into a global namespace, acting as an abstraction layer that separates data presentation from storage. Users and applications see standard NFS exports, SMB shares, S3 buckets, etc., but the data stored via these interfaces is not tied to any particular storage system. Data orchestration makes data within the global namespace mobile across storage systems, types, sites, and clouds, transparently.

Creating a unified data plane in this way streamlines administration by centralizing services like data protection, archiving, and security. Storage silos are eliminated, and storage migrations are non-events, occurring completely without disruption to users and applications. Data on preexisting storage isn't left behind, either. It can be rapidly assimilated into the Hammerspace global namespace while leaving it in place. Access control, share permissions, anti-virus scanning and WORM are managed from one place, not fragmented across multiple systems.

When using Hammerspace as a multi-tenant object store, this means:

© 2026 - Hammerspace

- Storage freedom of choice - Use any mix of storage systems and change them at any time, without disruption
- Performance scales up and out
- Built-in data protection and tiering capabilities, which reduce risk and help control costs
- Multi-site and multi-cloud support. Backing storage can be any storage, anywhere

Next, let's look in more detail at how these capabilities are implemented.

1.2. Hammerspace Multitenant S3 Building Blocks

The figure below shows the components of a Hammerspace installation providing S3-based object storage services to clients across multiple tenants. The subsections that follow explain the functions of each component. Hammerspace software runs on bare-metal or virtual infrastructure, including all popular public clouds, which means the Anvil and DSX nodes shown may be physical servers, virtual machines (VMs), or cloud compute instances.

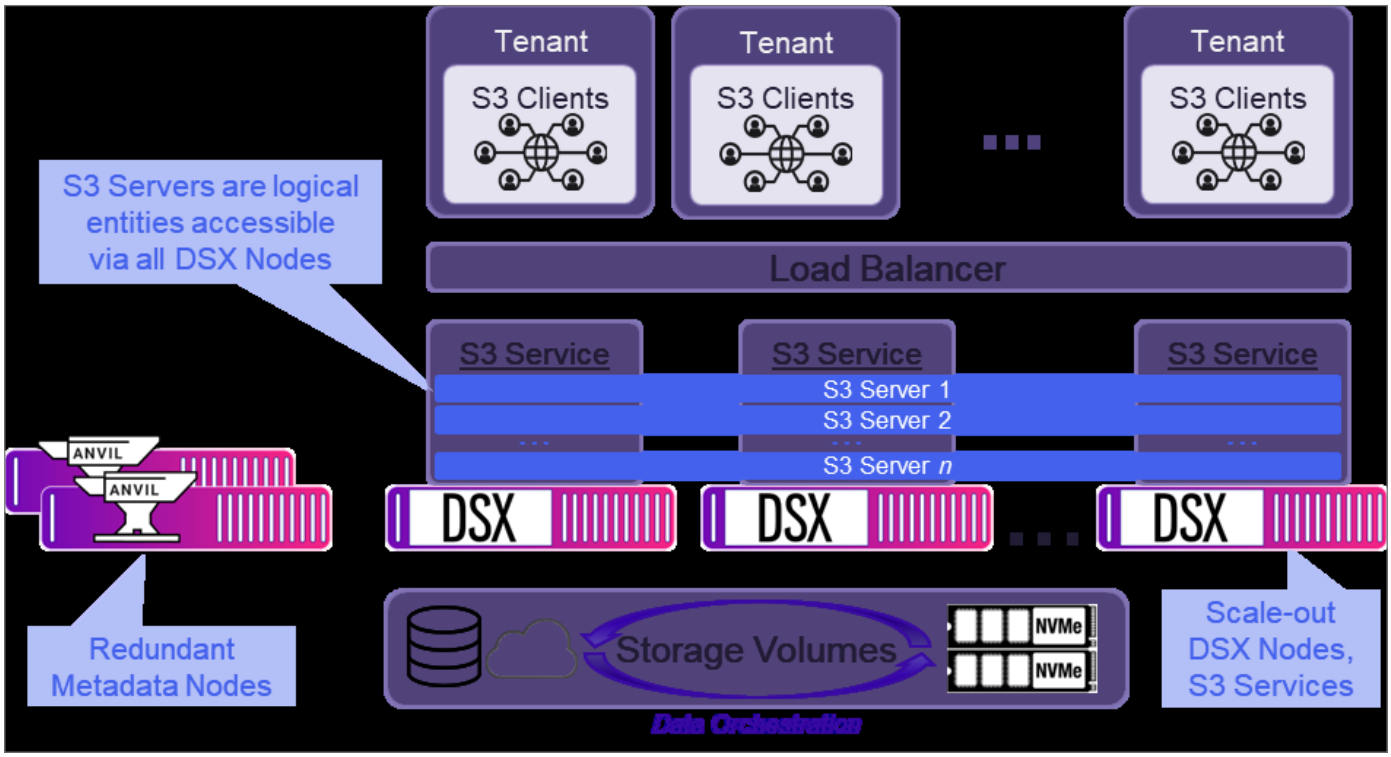


Figure 2. Components of a multitenant Hammerspace S3 installation

1.2.1. Tenants

Tenants are the entities that are securely sharing the S3 services provided by Hammerspace. A tenant may correspond to an organization, a group within an organization, or an application. A tenant may have one or more S3 clients associated with it.

© 2026 - Hammerspace

1.2.2. S3 Clients

S3 clients are the users and applications that consume the S3 storage services provided by Hammerspace. They interact with Hammerspace S3 services just as they would with any other S3-compatible storage service, authenticating with access key / secret key pairs and issuing GET, PUT, and other S3 commands. For the full list of supported S3 commands, refer to the Hammerspace Administration Guide.

1.2.3. Load Balancer

To scale performance and provide resilience against failures, S3 services are configured in a redundant, scale-out fashion across multiple nodes. Incoming S3 client requests are spread across all nodes using a customer-provided load balancer.

Load balancers can be static, such as simple round-robin DNS, or dynamic. Dynamic load balancing can get fancy, but in its simplest form, it detects when a server is down and routes traffic to other servers instead. Either static or dynamic load balancing can be used in a multitenant Hammerspace S3 environment, but each approach has pros and cons. The load-balancing strategy is covered in the tenant-isolation section.

1.2.4. Anvil Nodes

In every Hammerspace site, a redundant set of Anvil nodes is responsible for all metadata. Anvils also schedule and monitor data mobility activity, host the UI, and handle other cluster administrative functions. If a Hammerspace installation spans multiple sites and/or clouds, each site has its own Anvil nodes. Anvils are never in the data path.

1.2.5. DSX Nodes

DSX, short for Data Services eXtension nodes, can serve several storage and data-related roles in a Hammerspace cluster. They host client protocol services, including S3, and they execute data orchestration tasks under the direction of the Anvils. DSX nodes may also host block storage volumes. When more performance is needed for DSX-related functions, additional DSX nodes (up to 60) are deployed to scale out. This enables very high-performance, high-volume use cases.

1.2.6. S3 Service

Every DSX node contains an independent, stateless S3 service that may be enabled or disabled. When enabled, that node can service S3 requests from clients. A load balancer distributes S3 requests across S3-enabled DSX nodes.

1.2.7. S3 Server

S3 servers are the endpoints for S3 client connections to a Hammerspace cluster. Similar to Hammerspace SMB and NFS shares, S3 servers are logical entities, not tied to specific DSX nodes. Every S3 server is accessible through every DSX node with an active S3 service. A Hammerspace cluster may host up to 2048 S3 servers as of the time of this writing. For current information on product limits, refer to the Release Notes.

When Hammerspace is used to store objects, the S3 server is the fundamental unit of multitenancy. Each tenant is assigned one or more S3 servers, each of which has its own:

© 2026 - Hammerspace

- Hostname (FQDN)
- Bucket containers (optional - the default location for new buckets created via S3)
- Buckets
- Access control
- Objectives (explained below)
- Configuration parameters, such as whether users may create buckets, whether path-style or virtual-host access is used, and more

1.2.8. Storage Volumes

When a storage entity is configured into Hammerspace, it becomes a Hammerspace storage volume, reserved exclusively for Hammerspace use. Any or all of the following types of storage may be attached to Hammerspace as storage volumes:

- Block storage located within or attached to DSX nodes, including SAN-attached storage
- File storage, including NFS exports from commercial NAS appliances or simple Linux storage servers
- Object storage systems
- Cloud storage targets

In an S3 object store, storage volumes are where object data resides.

1.2.9. Data Orchestration

Data orchestration in Hammerspace means controlling data behavior based on declarative policies known as objectives. Among other functions, objectives control:

- Data Placement - which storage volume or volumes house data
- Data Protection - the level of availability and durability to be maintained
- Data Security - functions such as WORM, anti-virus scanning, etc.
- Data Mobility - which data is shared between sites, when data should move between storage tiers, etc.

For an S3 object store, data orchestration might be used to perform functions such as tiering colder data from NVMe storage volumes to HDD storage volumes, maintaining a secondary instance of all objects at another site or in a public cloud for DR protection, or simply controlling which storage volumes house data for specific buckets or S3 servers.

© 2026 - Hammerspace

Chapter 2. Tenant Isolation Topics

Due to the way Hammerspace's S3 capabilities are architected, various levels of tenant isolation are possible. Some resources are always shared among all tenants, some may be shared or isolated, and some are always isolated.

2.1. Administration

Administration of a Hammerspace cluster is centralized, even when the system hosts multiple S3 tenants. The Hammerspace administrator (e.g., an employee at the service provider running the cluster) has full access to manage the system, including the ability to create and manage S3 servers. Hammerspace admins also have visibility into share-specific and cluster-wide metrics, such as storage consumption and performance. Administrative logins do not have access to data, but administrators can delete buckets, including the data in those buckets.

S3 clients have no administrative access to the system. If the administrator configures an S3 server to allow it, S3 clients may create and delete buckets using the appropriate S3 commands. Otherwise, clients are restricted to working within administrator-created buckets.

If fully separate administrative domains are necessary - for example, if a service provider requires separate administrators for different tenant populations - then separate Hammerspace clusters may be deployed to meet this requirement.

2.2. S3 Client Access

Each S3 Server configured in a Hammerspace multi-tenant configuration must be assigned one or more unique fully-qualified domain names (FQDNs) as endpoints. Assigning FQDNs to S3 servers is technically optional, but in a multi-tenant configuration, it is required.

S3 clients make S3 requests using the FQDN, such as `s2.lab.com`. The local DNS resolves the FQDN to a DSX IP address, and the request is routed to the S3 service on that DSX. The S3 service inspects the host header on the request, and routes it to the S3 server with the matching FQDN.

© 2026 - Hammerspace

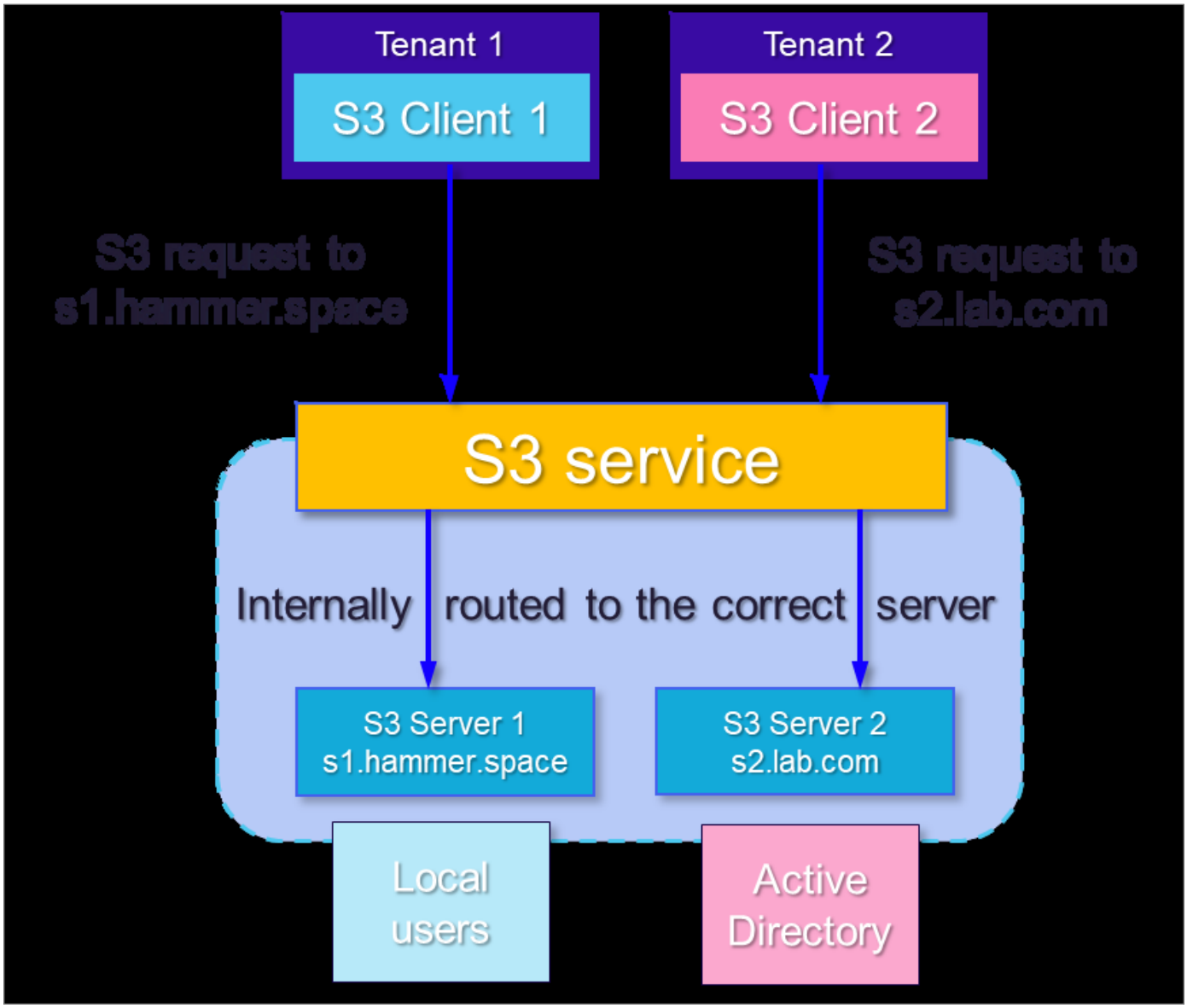


Figure 3. S3 requests are routed to the S3 server with the matching FQDN

As explained in the next section, even if an S3 client knows or guesses the FQDN of another tenant's S3 server, it cannot gain access to the data stored there. To prevent unauthorized access, however, it is recommended to use a separate DNS server for each tenant. This way, S3 clients can only resolve the names of S3 servers assigned to their tenant.

2.3. S3 Client Authentication

S3 clients are authenticated using access key / secret key pairs, which are independently defined for each S3 server. An access key / secret key pair defined for a client on one S3 server does not provide access to other S3 servers; an S3 client may have access key / secret key pairs on multiple S3 servers if desired.

The authentication source may be independently defined for each S3 server, with options for local users (a user database defined in Hammerspace) or Microsoft Active Directory (AD). Each access key / secret key pair defined for an S3 server is mapped to a user.

© 2026 - Hammerspace

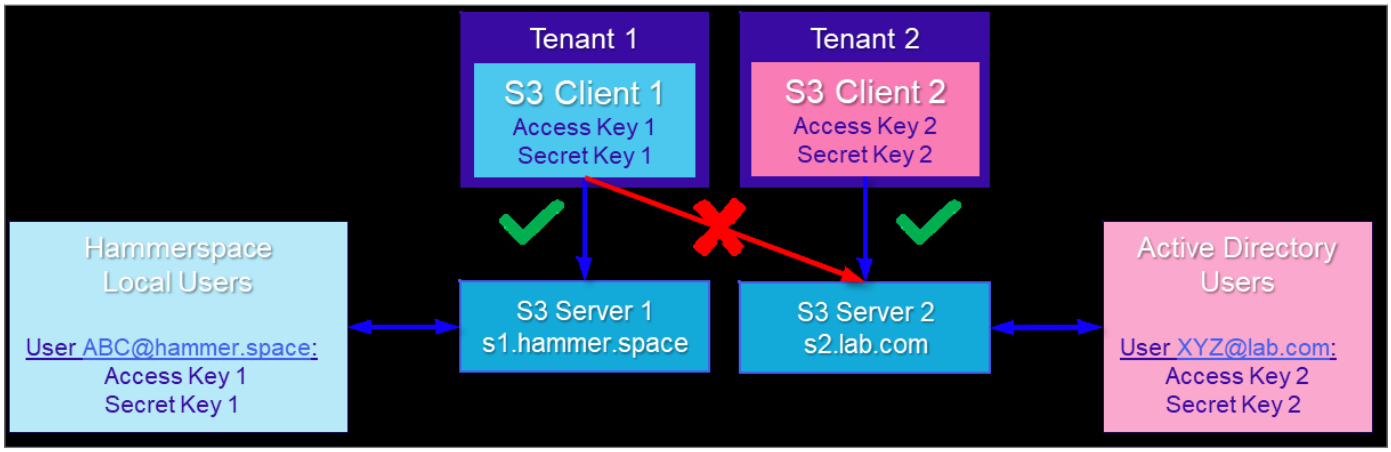


Figure 4. Access key / secret key pairs are defined independently for each S3 server

To remove client access to specific S3 resources, the Hammerspace administrator deletes the relevant access key/secret key pairs. To disable a client's access entirely, such as when an individual leaves the tenant's organization, the Hammerspace administrator needs only to disable the associated user in AD, or, in the case of local users, delete the local user or delete all of the user's access key/secret key pairs.

Hammerspace currently supports only a single AD Forest or other grouping of domains that are linked with trust relationships. This simplifies administration by enabling centralized management of user credentials across all tenants.

2.4. Traffic Isolation

The usual configuration for Hammerspace S3 (see the diagram in the Architecture section) is to enable the S3 service on every DSX node and use a simple static load balancing mechanism like a round-robin DNS to spread incoming traffic across the DSXs. For this configuration floating IP (FIP) addresses are configured in Hammerspace to maintain connectivity in the event a DSX node fails or is taken out of service. The best practice in this case is to create one DNS 'A' record for each DSX FIP, for each S3 server.

This simple configuration will result in traffic for all S3 servers (and thus all tenants) being spread across all DSX nodes. This is adequate in most cases, but carries a small risk of "noisy neighbor" effects, where a tenant generating unexpectedly high traffic may negatively impact the performance experienced by other tenants.

Assigning specific DSX nodes to particular tenants or groups of tenants can mitigate any noisy neighbor risk. Dedicated DSX nodes can also enable different business models and cost allocation approaches, since it's clear which tenant is using each DSX.

© 2026 - Hammerspace

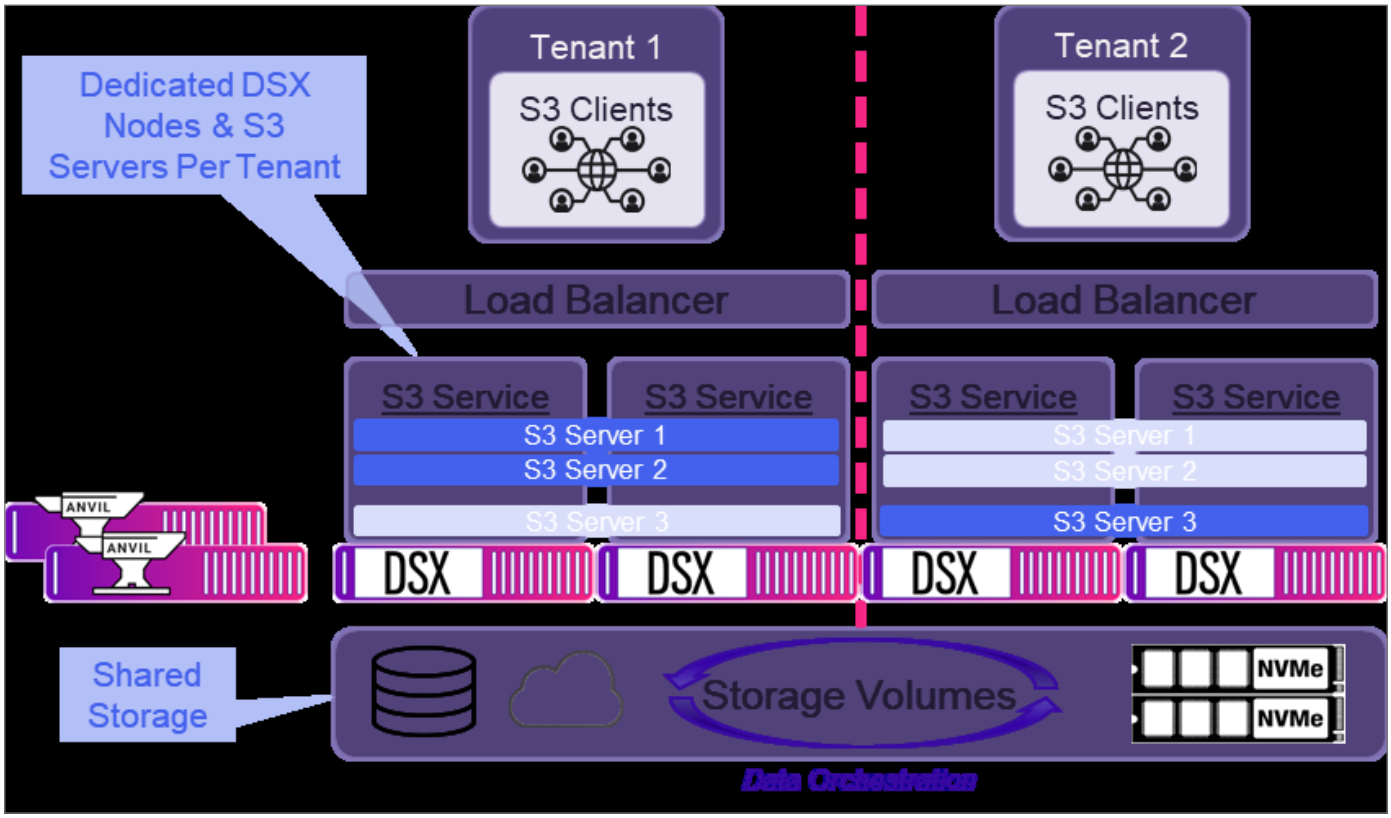


Figure 5. Dedicated DSX nodes per tenant require dynamic load balancing

Restricting DSXs to specific tenants requires the use of dynamic load balancing. In this case floating IPs are not used, just the static IPs of the DSX nodes. The load balancer allocates requests for each S3 server's FQDN across the DSX node IPs assigned to that S3 server. The responsibility to detect an out of service DSX and route requests around it shifts from Hammerspace to the load balancer or other upstream customer infrastructure.



Even though all S3 servers are still represented on all DSX nodes, the load balancer configuration will restrict requests to specific DSXs.

Keep in mind that isolating DSXs per tenant results in a less flexible configuration that is more complex to configure and manage. Sharing all DSXs among all tenants is recommended unless isolation is absolutely necessary.

2.5. Storage Isolation

Storage for a Hammerspace multitenant S3 implementation can likewise be shared or dedicated. By default all storage volumes are shared by all tenants, and this is the simplest approach. As with sharing DSX nodes, however, sharing storage can lead to noisy neighbor problems. Isolating each tenant to their own pool of storage guarantees per-tenant storage performance, and it also enables the administrator to set per-tenant capacity quotas.

Configuring isolated storage for each tenant involves the configuration of the S3 servers, the structure of the underlying shares, and some objectives.

© 2026 - Hammerspace

2.5.1. How Hammerspace Stores Data

Within Hammerspace, all data is visible and accessible in a unified namespace via any protocol. "File" and "Object" are simply different lenses through which data may be viewed and accessed. All data, regardless of the protocol used to write it into Hammerspace, is ultimately stored as files in a file system.

The key to the "dual citizenship" of data across both file and object realms is the principle that one object = one file and vice versa. Each object written via S3 becomes a file in the Hammerspace file system. This is true even for multipart uploads, where the individual object parts are written into a hidden temporary directory and then assembled into a complete file.

Similarly, when an existing share or directory containing files is exposed as an S3 bucket, the files appear as objects with names that reflect the file name, and if located in a subdirectory, the path. Buckets created via S3 become directories in the file system, located at the specified bucket root. Object names with embedded slashes will cause the corresponding directory structure to be created on the file system. The figure below illustrates these concepts.

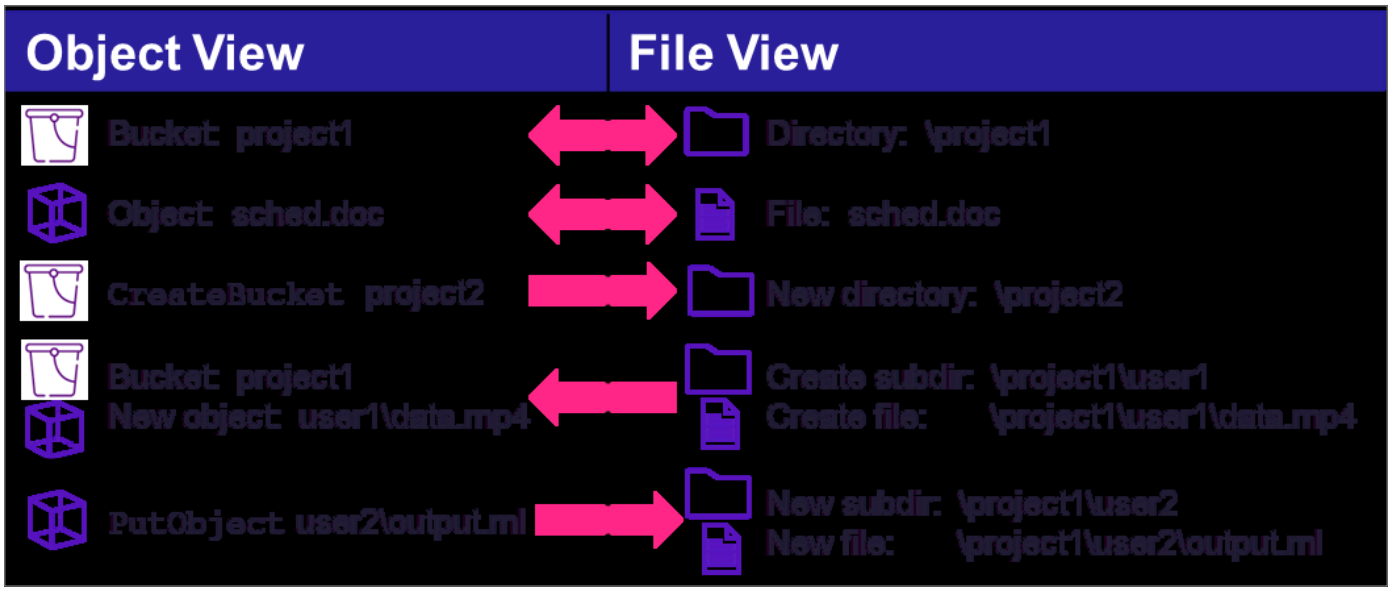


Figure 6. One object = one file: buckets and objects map to directories and files

When an object is stored into a Hammerspace S3 bucket, that object is stored in the corresponding Hammerspace share that was defined at the time the S3 server was created. Like S3 servers, Hammerspace shares are virtual entities. Within the share, each object is stored as a single file on the filesystem of one or more of the underlying Hammerspace storage volumes. The exact storage volume or volumes that house the object data is determined based on the configured Hammerspace objectives.

© 2026 - Hammerspace

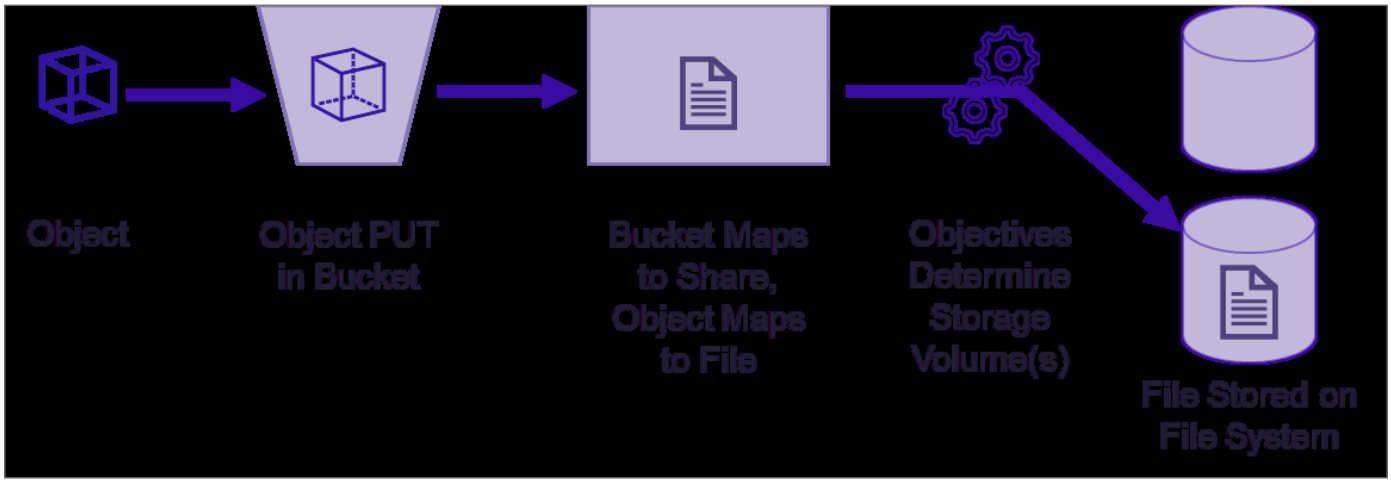


Figure 7. Objects are stored as files on storage volumes selected by objectives

Isolating a tenant's data to a specific set of storage hardware, then, involves configuring a Hammerspace share for the tenant's S3 server(s), and configuring the objectives on that share to place data on the storage volumes representing the tenant's physical storage devices. The diagram below shows an example configuration using both dedicated DSX nodes and dedicated storage for each tenant.

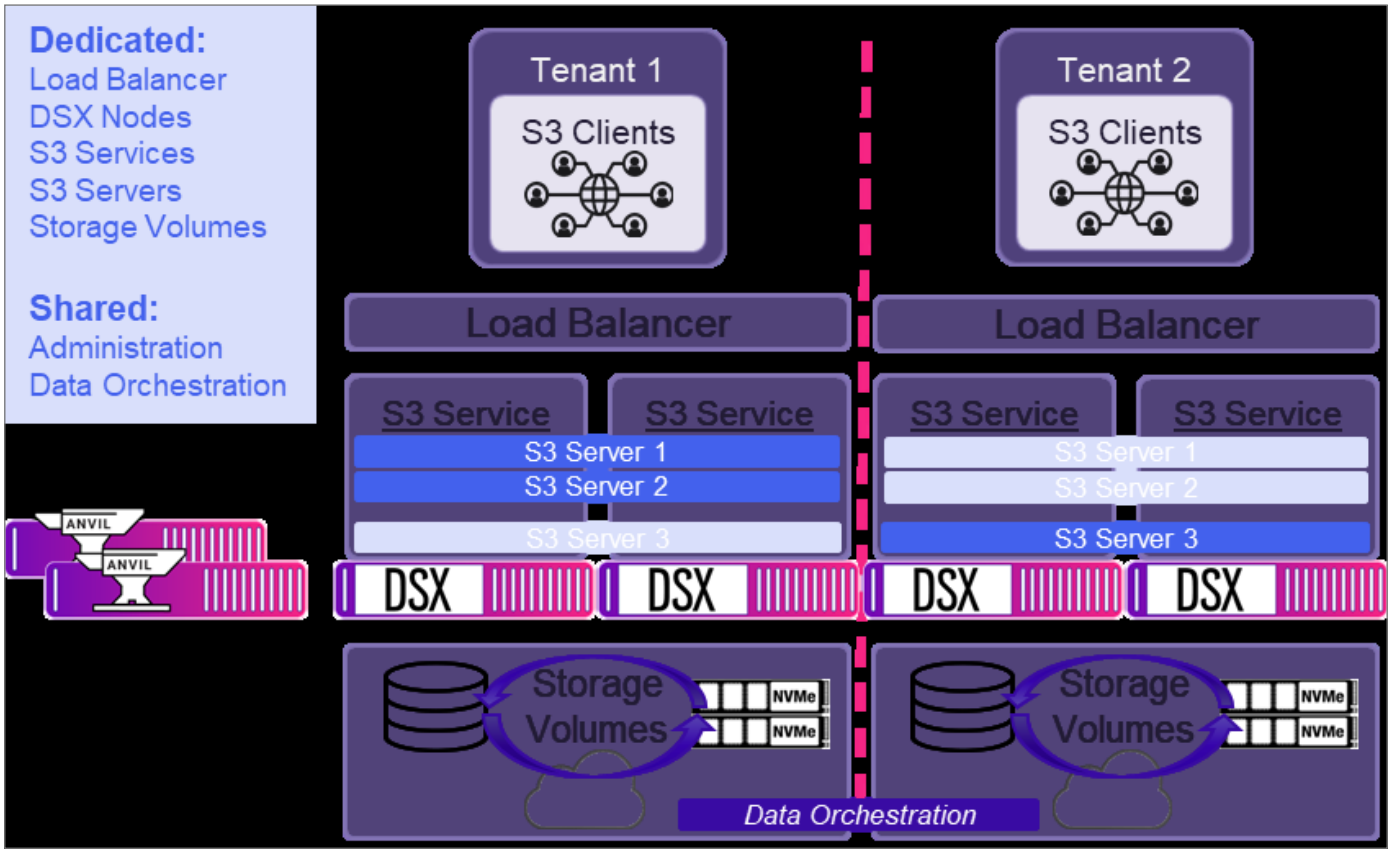


Figure 8. Example configuration with dedicated DSX nodes and dedicated storage per tenant

Remember that objectives can also drive more sophisticated data orchestration tasks, such as tiering across storage types, data protection, and more. The principles of isolating tenant storage remain the same, however; objectives control data placement on storage volumes mapped to tenant-specific storage resources.

© 2026 - Hammerspace

2.5.2. Storage Capacity Quotas

Hammerspace storage capacity quotas are defined at the share level. A configurable warning threshold triggers an alert to the Hammerspace administrator when it is exceeded. If an S3 client exceeds the configured capacity limit, they will receive either a `507 Insufficient Storage` error or a `403 Forbidden` error.

© 2026 - Hammerspace

Chapter 3. Deployment Examples

To illustrate how to create multitenant S3 configurations in Hammerspace, this section contains two fictional but realistic deployment examples. The first example has two tenants configured to use shared resources, and the second example has two tenants with isolated resources. Not every option is shown, so this document should be used in combination with other applicable documentation as needed.

After listing the prerequisites and pre-installation preparatory procedures all multitenant S3 Hammerspace installations have in common, step-by-step instructions are provided for configuring each example configuration. For simplicity, these examples use the GUI, but all of the configuration shown can be completed using the CLI or with automation leveraging the Hammerspace REST API.

3.1. Prepare for S3 Configuration

Several decisions need to be made before configuring Hammerspace S3 services. The major ones are listed below. For complete information on all of the S3 configuration options, refer to the Hammerspace Administration Guide.

- S3 Server Endpoint Names: Each S3 server in a multitenant configuration is addressed using an endpoint name, which is an FQDN. The endpoint name specified by the client is used by the receiving DSX to route the request to the appropriate S3 server process.
- Data Access Method: S3 buckets in Hammerspace may be addressed two ways: path-based (e.g., `s3.meche.ivy.edu/mybucket`) or virtual-host style (e.g., `mybucket.s3.meche.ivy.edu`). Virtual-host style requires DNS entries for each bucket.
- Identity Provider: Hammerspace can connect to Active Directory or use a local user database. Connecting Hammerspace to AD requires the appropriate AD administrative credentials. Local users are configured in the Hammerspace admin GUI.
- Strict Bucket Names: If maintaining 100% compatibility with Amazon AWS is required, Strict Bucket Names should be selected. This option restricts bucket names to those that conform with AWS S3 bucket naming rules.
- Strict Bucket Lists: Should S3 clients be able to see all buckets on an S3 server using a 'list buckets' command, or should their view be restricted to only the buckets they own?
- Bucket Management: Should S3 clients be able to create buckets using the appropriate S3 commands, or will they be restricted to using pre-created buckets? If the former, a 'Bucket Container' will need to be created. If the latter, the administrator must create the buckets in advance. If creating buckets will be allowed, will deleting buckets also be allowed?
- S3 Access Permissions: What default permissions should be assigned to objects? Options are Private, Group, and Public. Note that this default can be overridden by the bucket-level access permissions settings, which can be unique for each bucket and bucket container.

3.2. Prerequisites

- Configure Storage: Ensure the storage that will be used with Hammerspace is installed and configured. Refer to the Hammerspace Configuration Guide for instructions.

© 2026 - Hammerspace

- Install Hammerspace Software: Install the appropriate number of Anvil nodes and DSX nodes. For details, refer to the Hammerspace Installation and Licensing Guide.
- Install Quorum Server: If required, install the lightweight quorum service on a Linux server (physical or VM). Complete requirements and installation instructions for the quorum server are contained in the Hammerspace Installation and Licensing Guide.
- Configure DNS: When using floating DSX IP addresses, create one DNS 'A' record for each DSX FIP for each S3 server.
- Configure Load Balancer: Load balancing services are customer-provided and should be configured to spread traffic for S3 endpoints across DSX nodes as appropriate for the configuration.
- Identify S3 Clients: Identify the individuals and applications that will be using each S3 server. All clients using a particular S3 server must have a corresponding user account in the same identity provider - either AD or Hammerspace local.

© 2026 - Hammerspace

Chapter 4. Example 1: Two Tenants, Shared Resources

This example will configure a single Hammerspace instance for two tenants, in this case two engineering departments at a university. For simplicity and ease of administration by the university's central IT organization, all resources are shared. Servers are bare metal.

4.1. Hammerspace Architecture

A pair of Anvil nodes are deployed for redundancy, with a lightweight quorum service running elsewhere in the customer infrastructure. The quorum service provides extra protection against 'split brain' if a network partition occurs and is a best practice when using two Anvils.

Three DSX nodes are used, each with the S3 service enabled. This is the minimum quantity of DSX nodes recommended as it enables one to be taken out of service for maintenance while still retaining protection against a DSX failure. In this installation, the DSXs host S3 services and also act as data movers for data orchestration. If these three are not sufficient to handle the load, it is a simple matter to deploy additional DSXs.

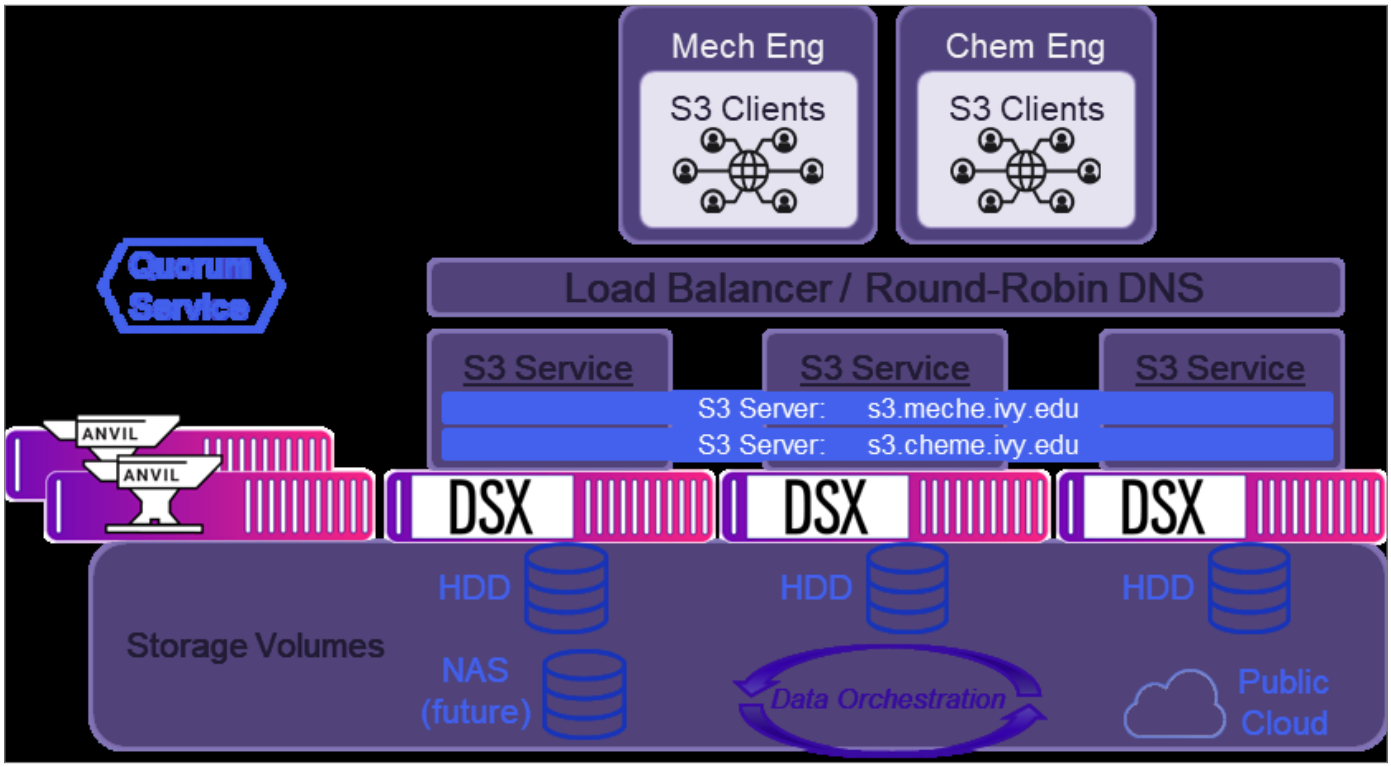


Figure 9. Example 1 architecture: two tenants sharing all resources

Back-end storage consists of HDD-based JBODs attached to each DSX. The university is also considering repurposing some older NAS systems to provide additional capacity. These may be easily added later. For offsite DR protection, a copy of each object is maintained in inexpensive public cloud storage.

© 2026 - Hammerspace

4.2. Configuration Steps

The process to set up this example configuration consists of ten steps.

1. Create Floating IPs: On the Administration page, Network tab, click **Add Floating IP** to add a floating IP.

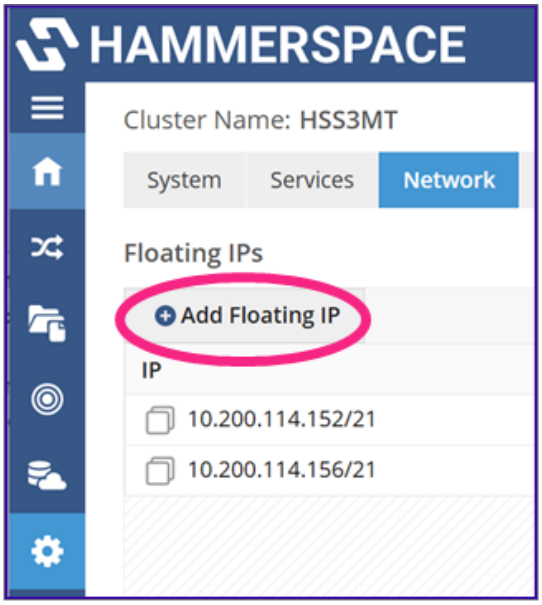


Figure 10. The Add Floating IP button on the Network tab

Fill in the IP and subnet mask, and select Portal for the Type, and click **Add**.

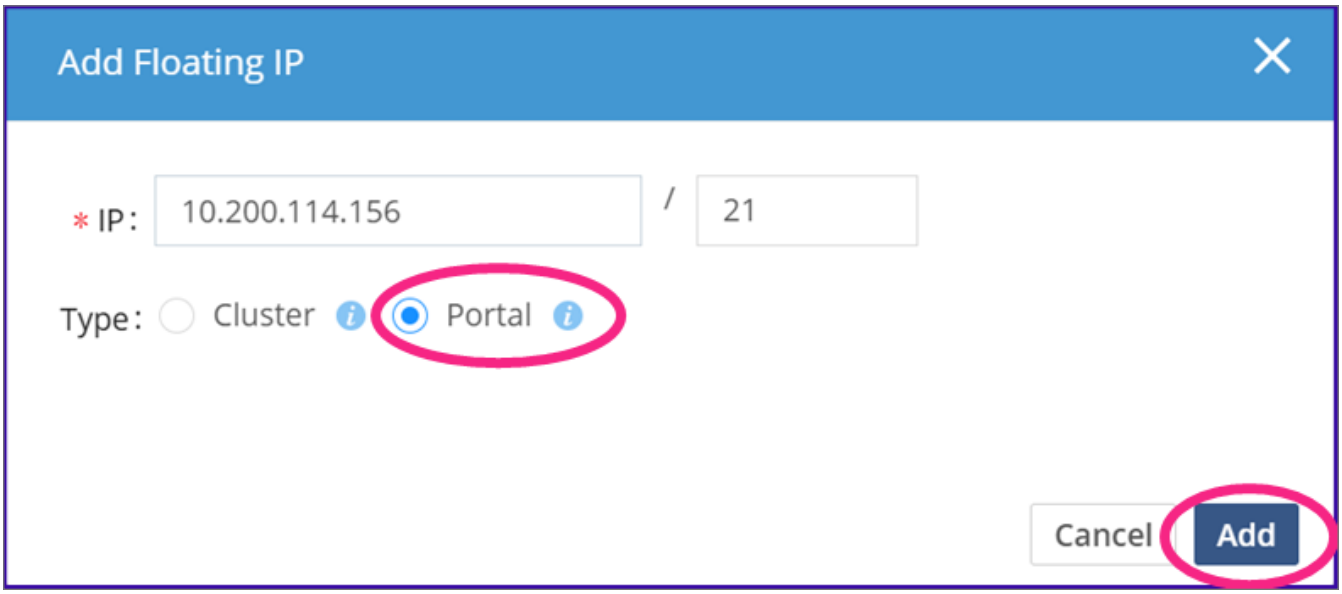


Figure 11. Adding a floating IP with Portal as the type

Follow the same process to add two more floating IPs. Note that one is assigned to each DSX. If a DSX goes down for any reason, its floating IPs will be reassigned to a surviving DSX node.

© 2026 - Hammerspace

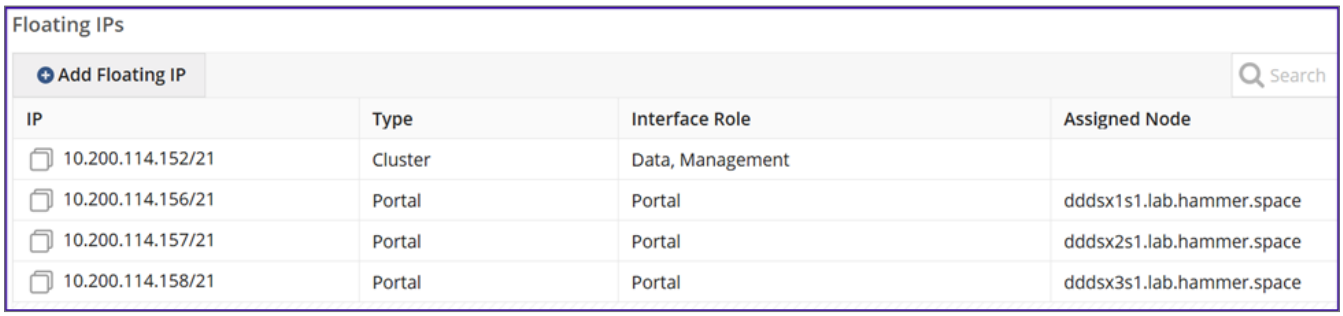


Figure 12. One floating IP assigned to each DSX node

2. Enable S3 Services on all DSXs: Still on the Administration page, switch to the Services tab and enable the S3 service on all three DSXs.

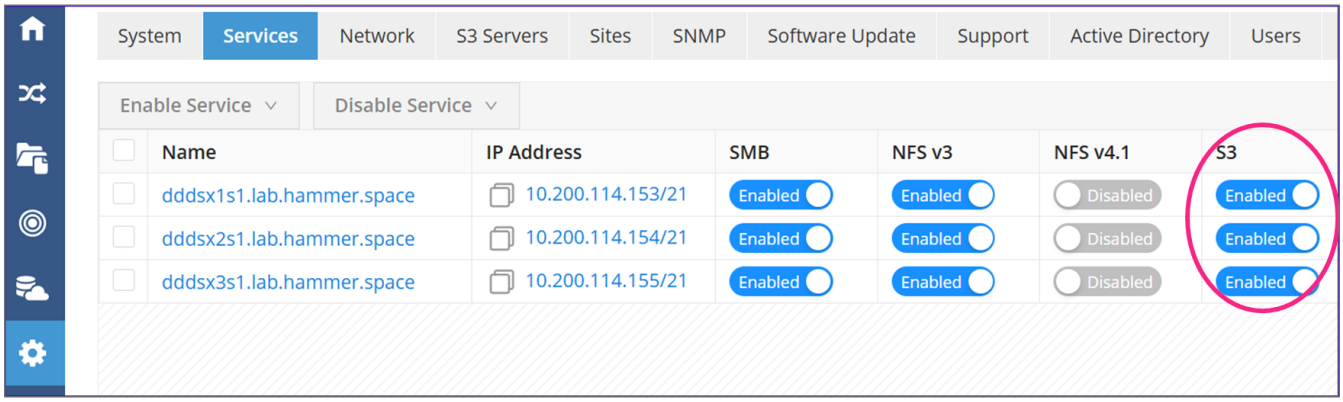


Figure 13. The S3 service enabled on all three DSX nodes

3. Create Users: If using Active Directory, go to the Active Directory tab to configure Hammerspace to connect to AD. Full instructions are contained in the Hammerspace Administration Guide. The instructions below are for creating local users.

   Still on the Administration page, switch to the Users tab and use the **Create Users** button to add at least one user for each tenant. Select the S3 Data Access Role.

© 2026 - Hammerspace

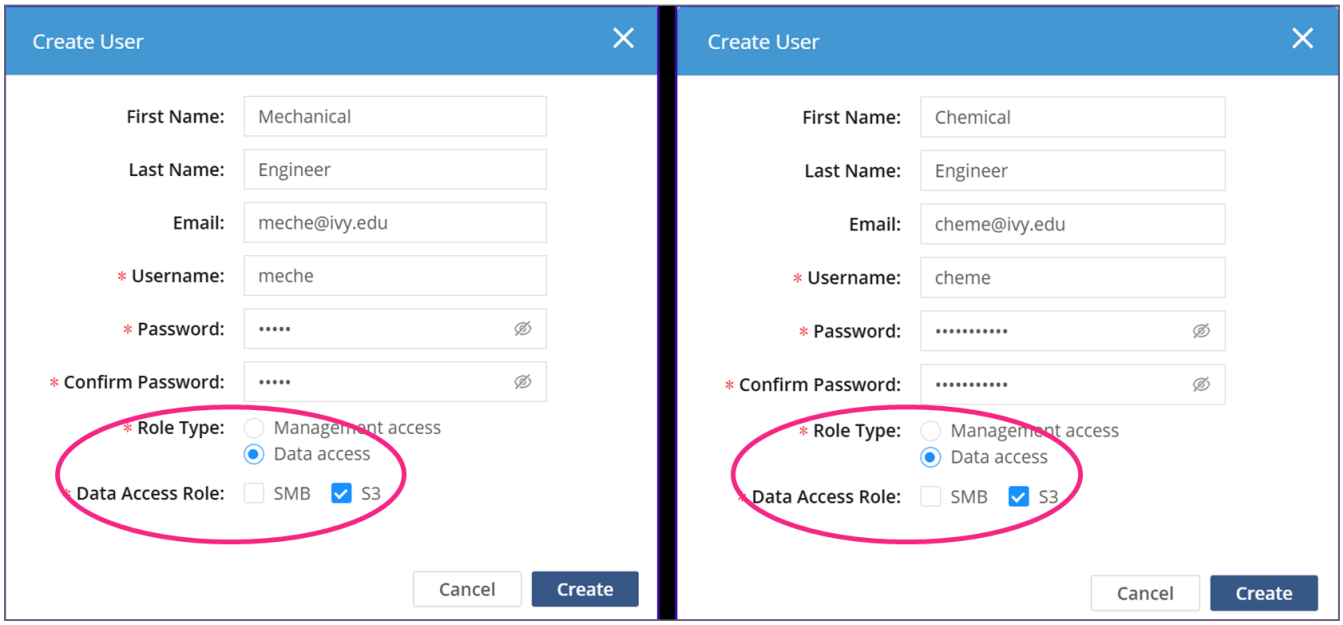


Figure 14. Creating a local user with the S3 Data Access role

4. Create a Volume Group: The local DSX disk volumes will have already been added to Hammerspace as storage volumes as part of the DSX installation process. Since all storage volumes will be shared among all tenants, configuring a volume group containing all of these storage volumes will simplify other configuration.

   On the Infrastructure page, Volume Groups tab, click **Create Volume Group**. Fill in the details in Step 1, then skip to Step 4.

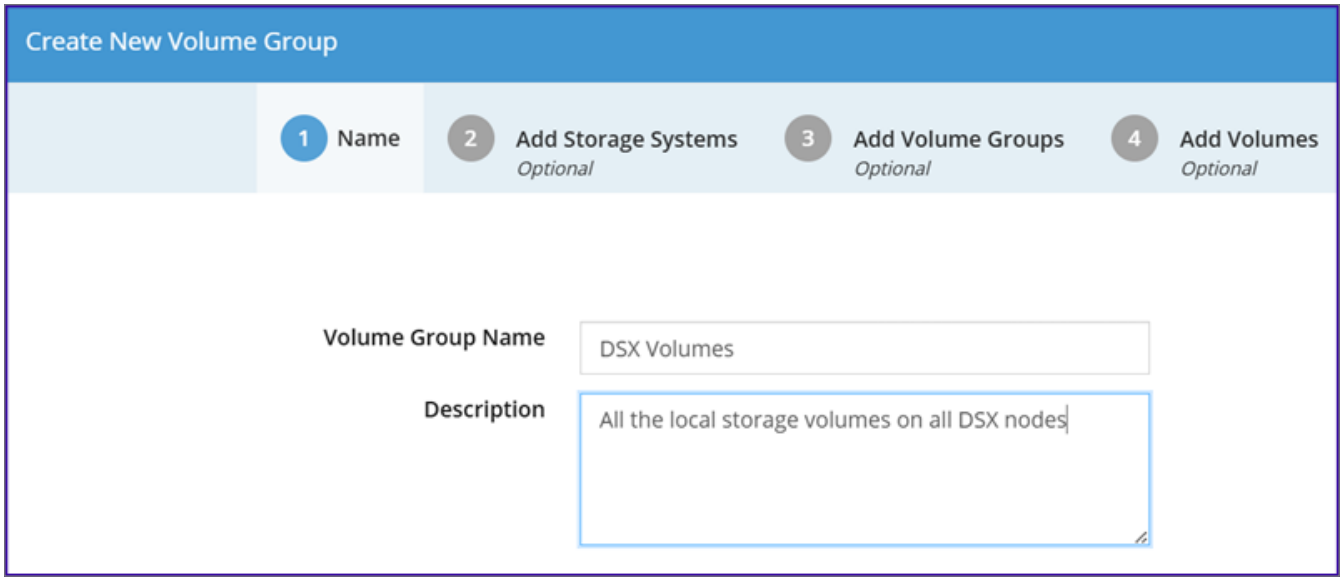


Figure 15. Volume group details in Step 1 of the wizard

In Step 4, select all of the DSX storage volumes and click **Next**. On Step 5, click the button to confirm and create the Volume Group.

© 2026 - Hammerspace

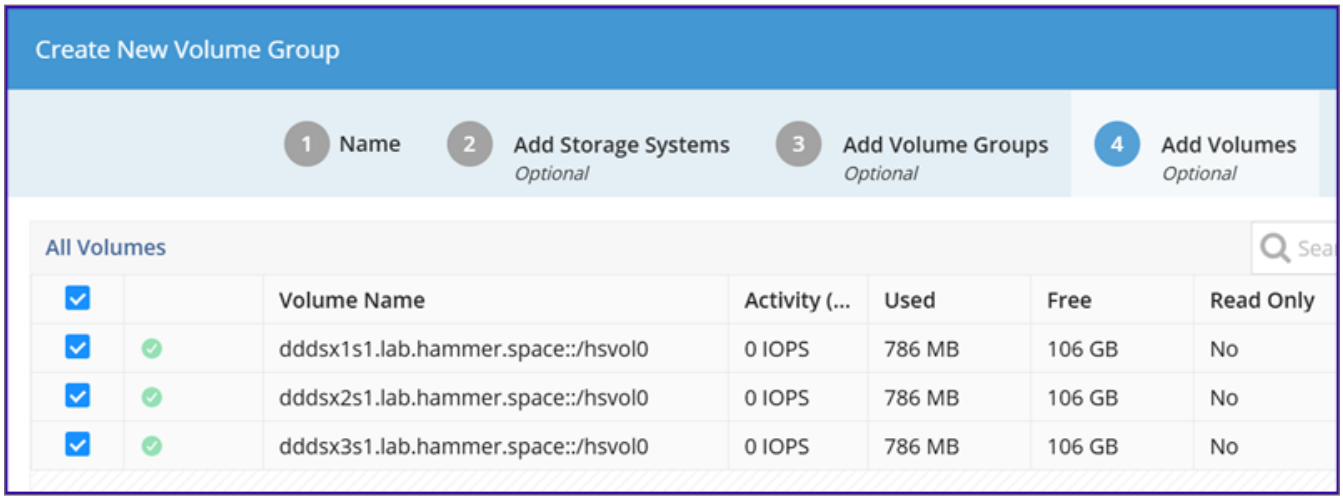


Figure 16. Selecting all DSX storage volumes in Step 4

Back on the Infrastructure page, you will see the new Volume Group in the list. If more DSX storage volumes for S3 use are added in the future, adding them to this Volume Group is all that needs to be done.

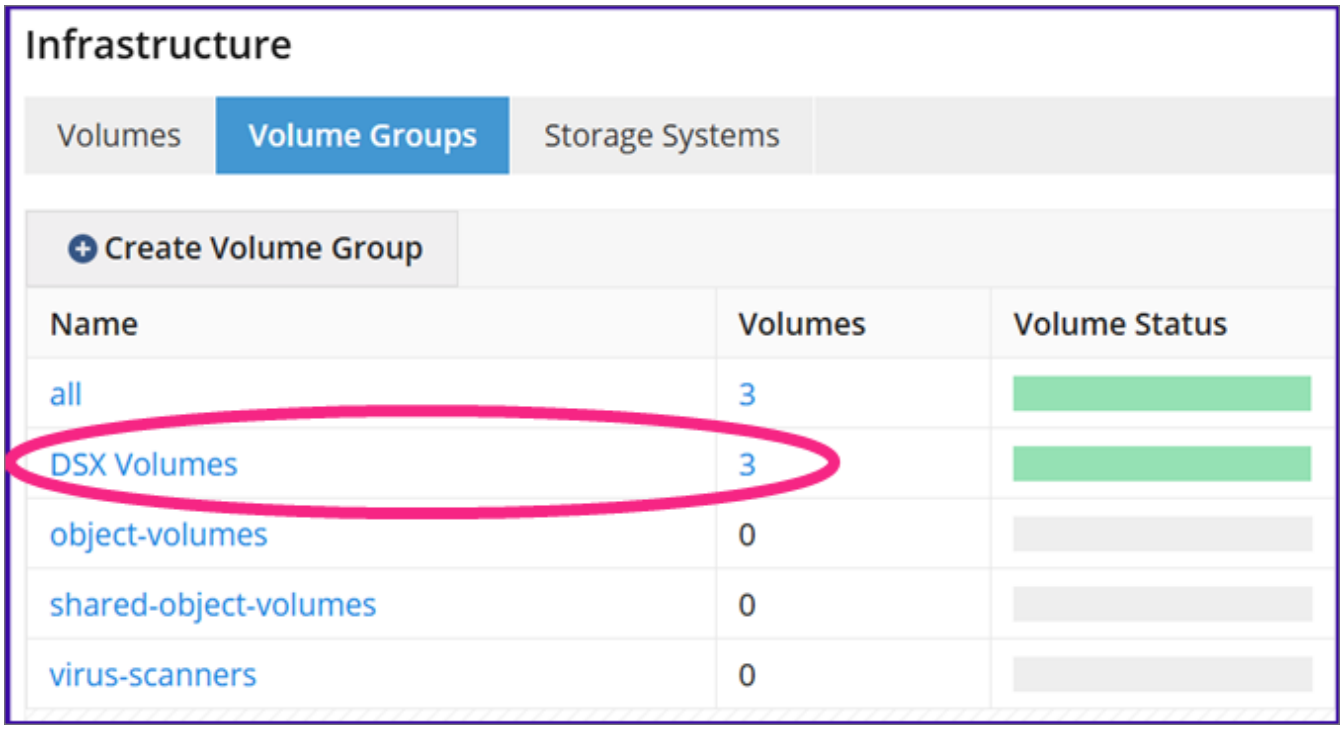


Figure 17. The new volume group in the Volume Groups list

5. Create a Cloud Volume: This example uses an AWS cloud bucket to store DR copies of objects. Before that can happen, the AWS bucket needs to be configured into Hammerspace as a storage volume.

   On the Infrastructure page, Storage Systems tab, click **Add Storage System**.

© 2026 - Hammerspace

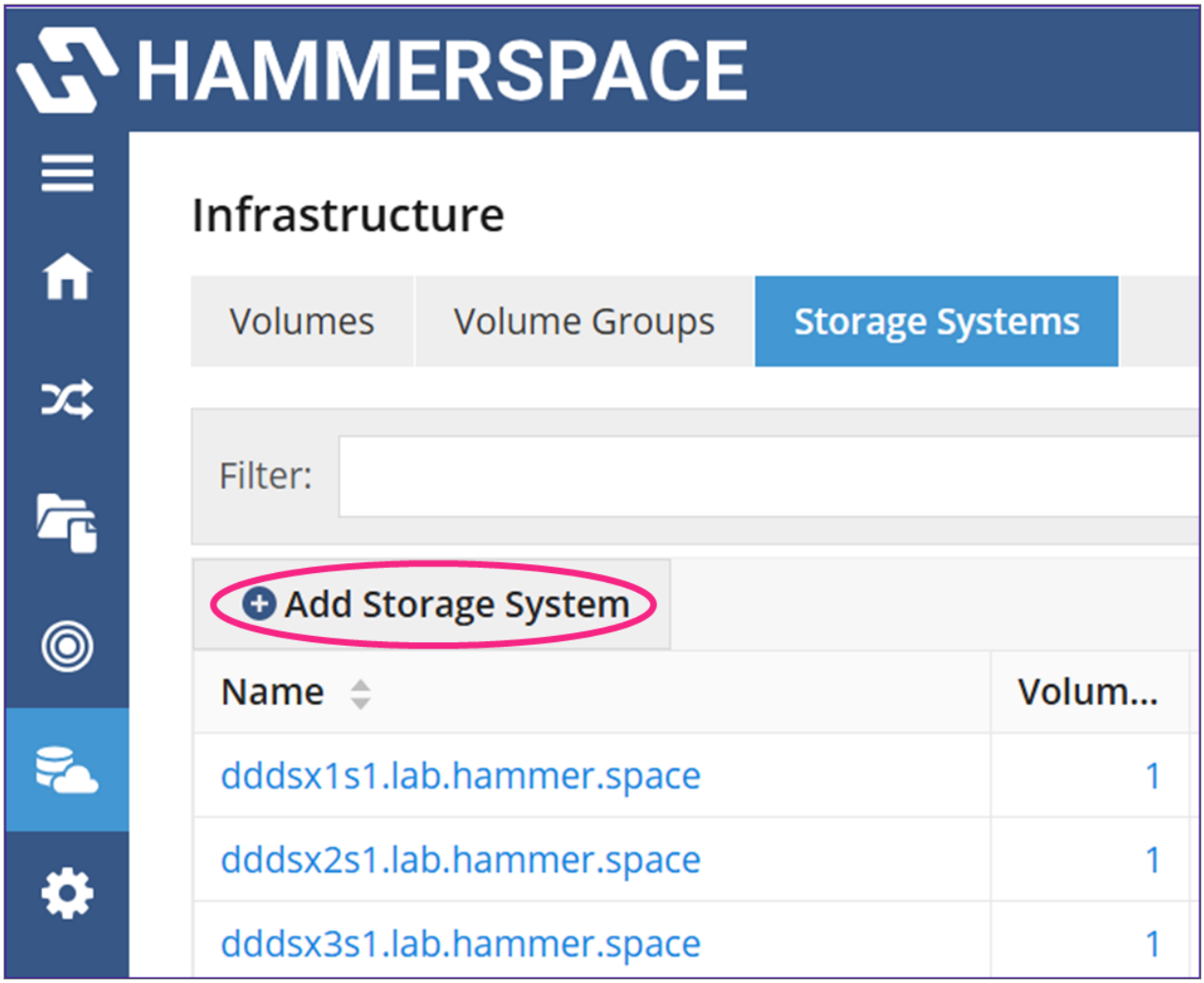


Figure 18. The Add Storage System button on the Storage Systems tab

Fill in the details, selecting Object Storage and Amazon S3 for the storage type, using the correct Access Key / Secret Key credential pair and selecting the correct region.

© 2026 - Hammerspace

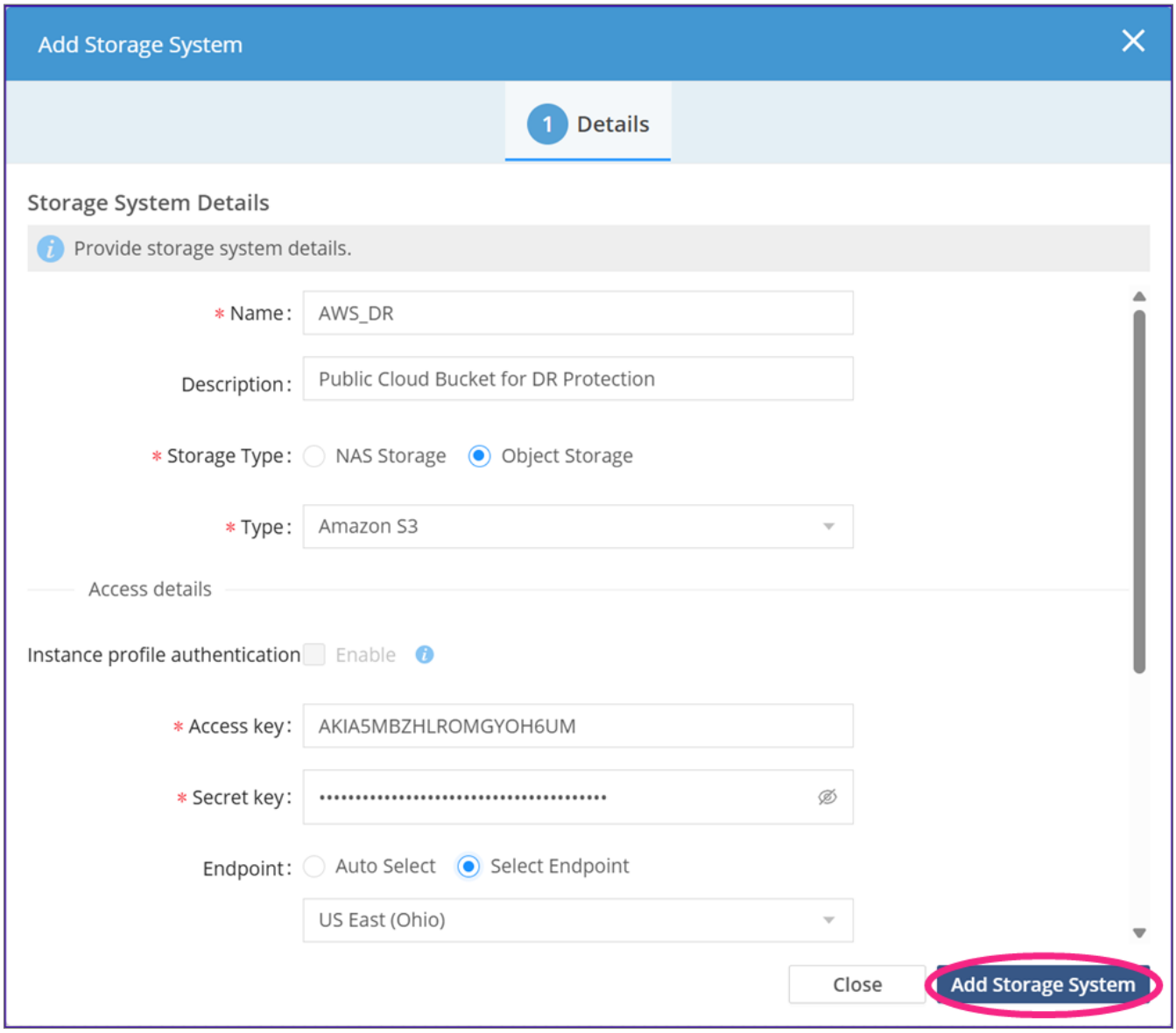


Figure 19. Adding an Amazon S3 storage system

Click **Add Storage System** to continue.

The following dialog will appear once the storage system is successfully configured. Note at this point we have simply connected to the AWS account with the supplied credentials. No bucket has been selected yet. Click **Add Volumes** to continue.

© 2026 - Hammerspace

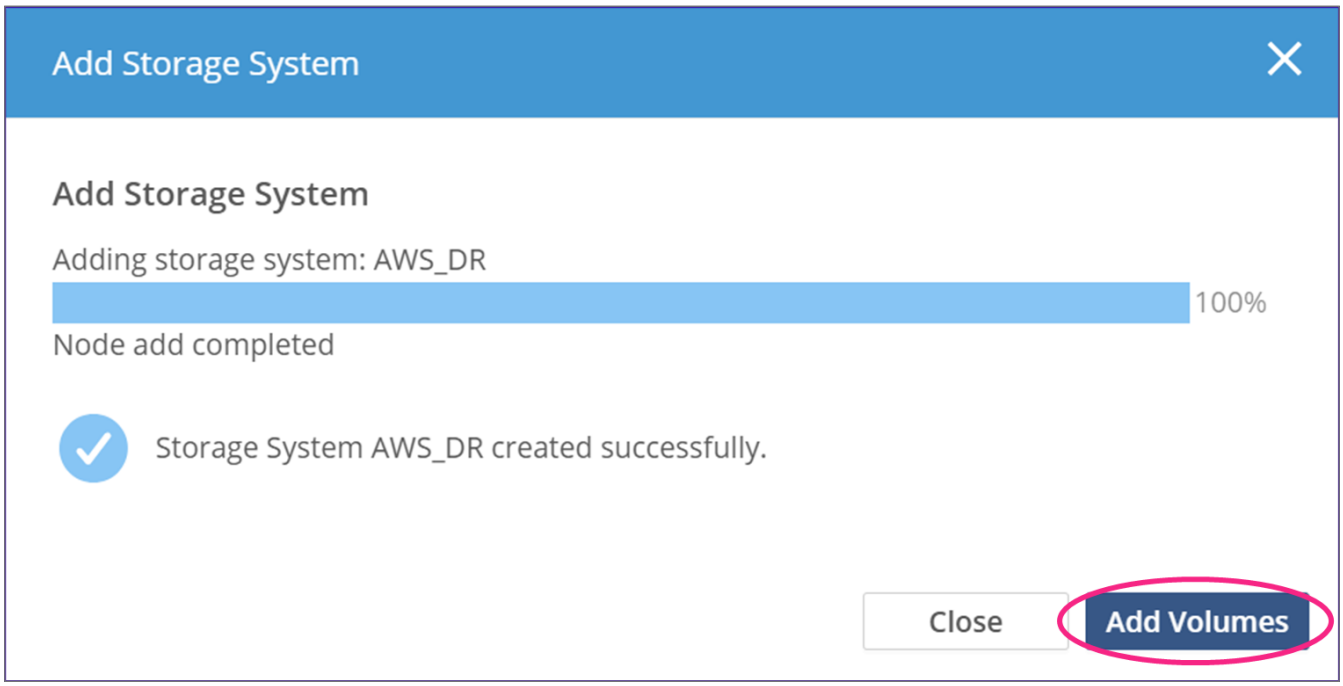


Figure 20. The storage system is configured; no bucket selected yet

Click the checkbox next to the appropriate bucket to select it. Bucket names appear in the 'Name' column following the storage system name. In this example, the bucket name is "ivydr."

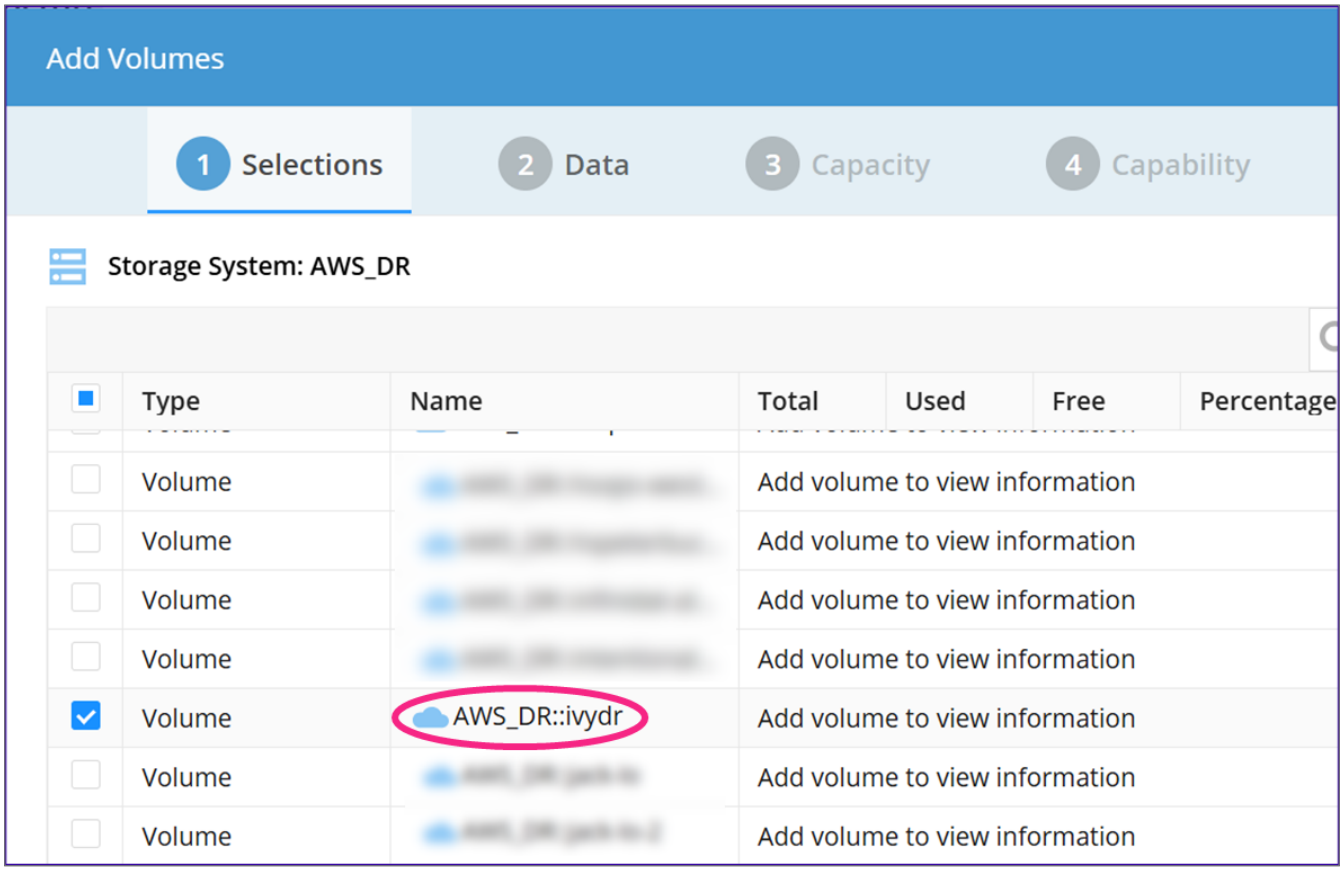


Figure 21. Selecting the ivydr bucket

In the next step, un-check the Shared Volume checkbox, as this bucket will not be used for sharing data

© 2026 - Hammerspace

between multiple sites. Then click **Next Step**.

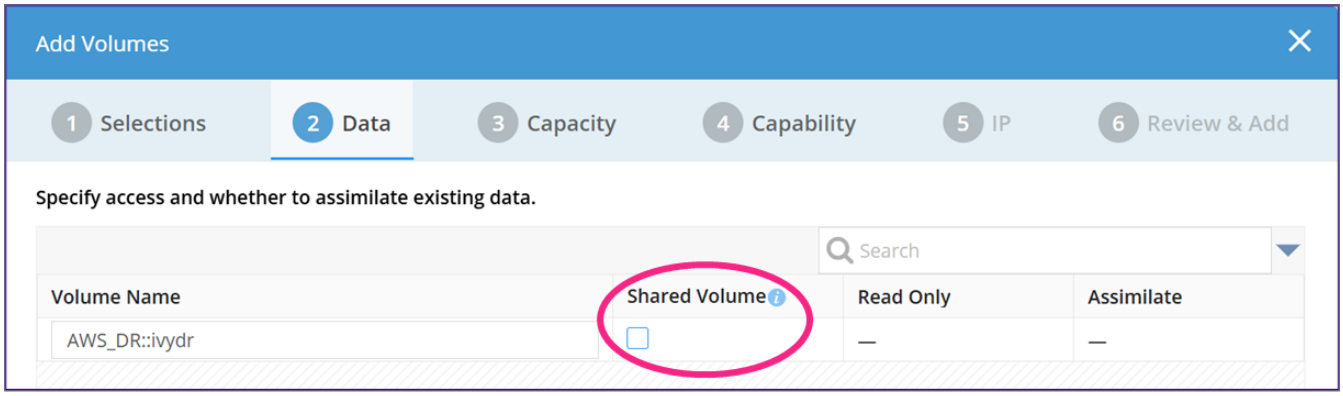


Figure 22. Un-checking the Shared Volume checkbox

In step 3, leave the Capacity limit at 'Auto' to ensure all objects are protected, regardless of how large the capacity grows, or place a limit on the amount of data Hammerspace can store in the bucket. Enable Compression to minimize cloud storage charges. Objects stored in cloud buckets are automatically deduplicated. Click **Next Step** to proceed.

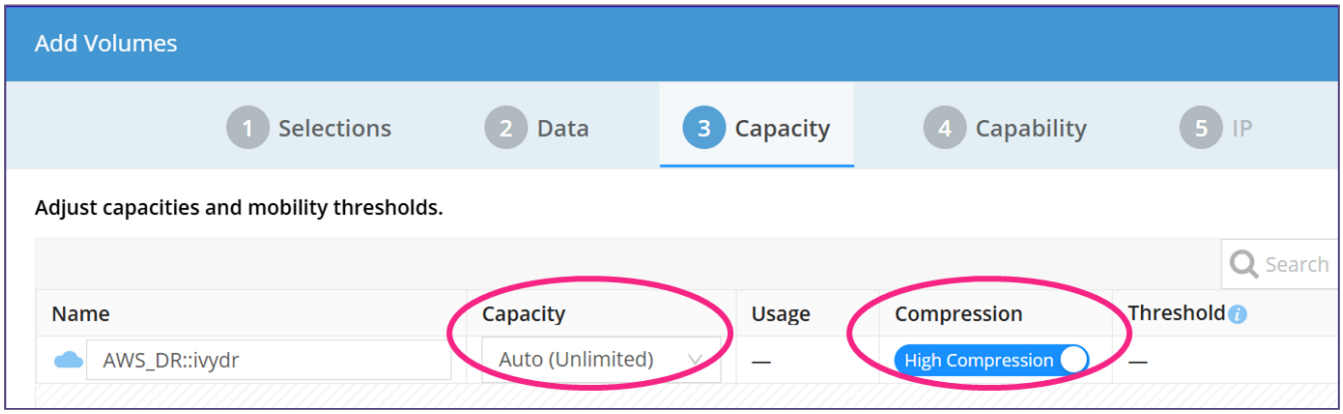


Figure 23. Capacity and compression settings for the cloud volume

On steps 4 and 5, simply click **Next Step** to proceed, leaving all values at their defaults. On step 6, click **Add Volume** to add the bucket as a storage volume.

© 2026 - Hammerspace

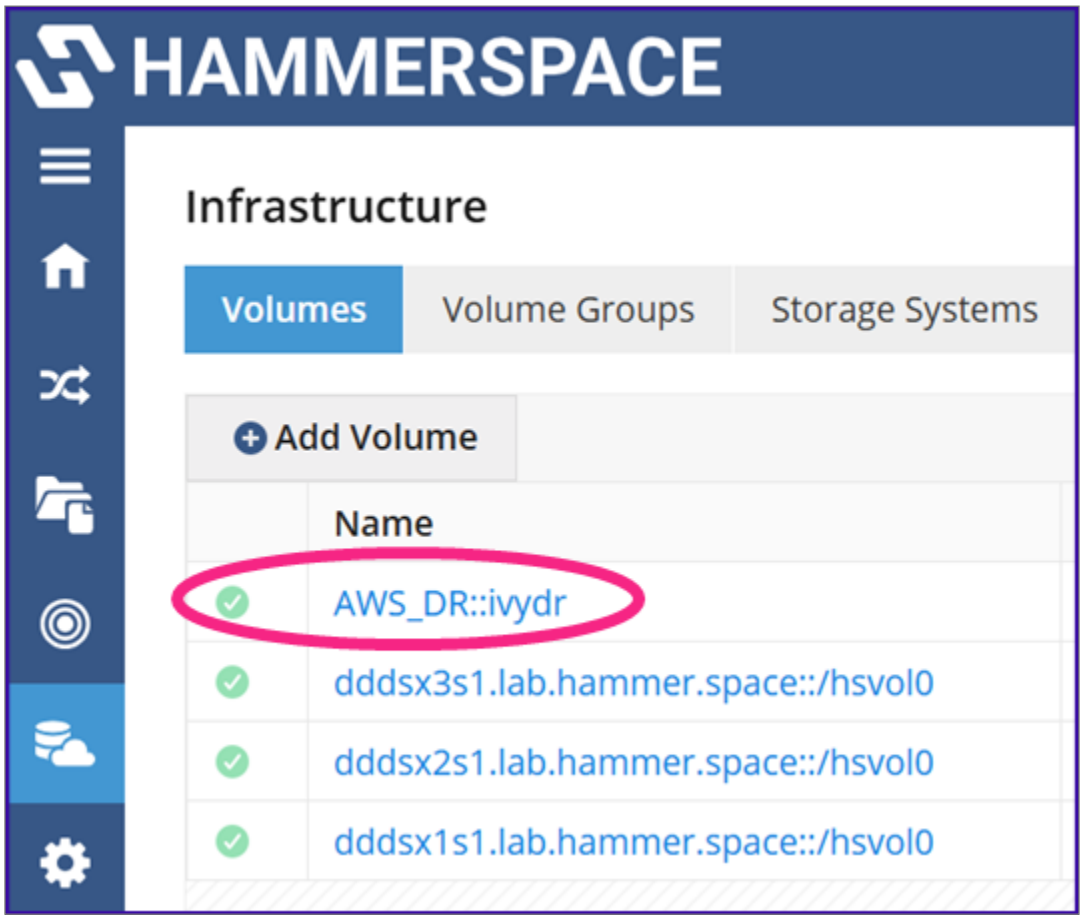


Figure 24. Confirming the new cloud volume

The new volume will appear in the list of volumes on the Volume tab of the Infrastructure page.

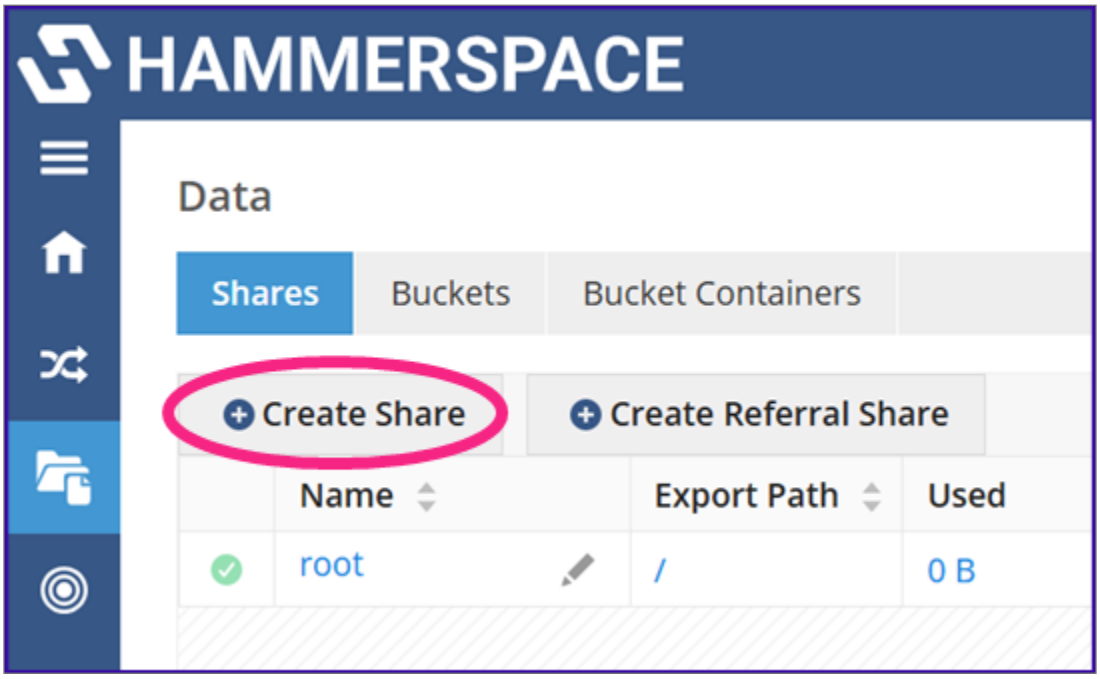


Figure 25. The cloud volume in the volumes list

6. Create a Share: Hammerspace S3 servers require a target share, so before creating any S3 servers, one or

© 2026 - Hammerspace

more shares must be configured. On the Data page, Shares tab, click **Create Share** to begin the process.

Give the share a name and description, and if desired set a limit on the maximum size of the share (a share quota).

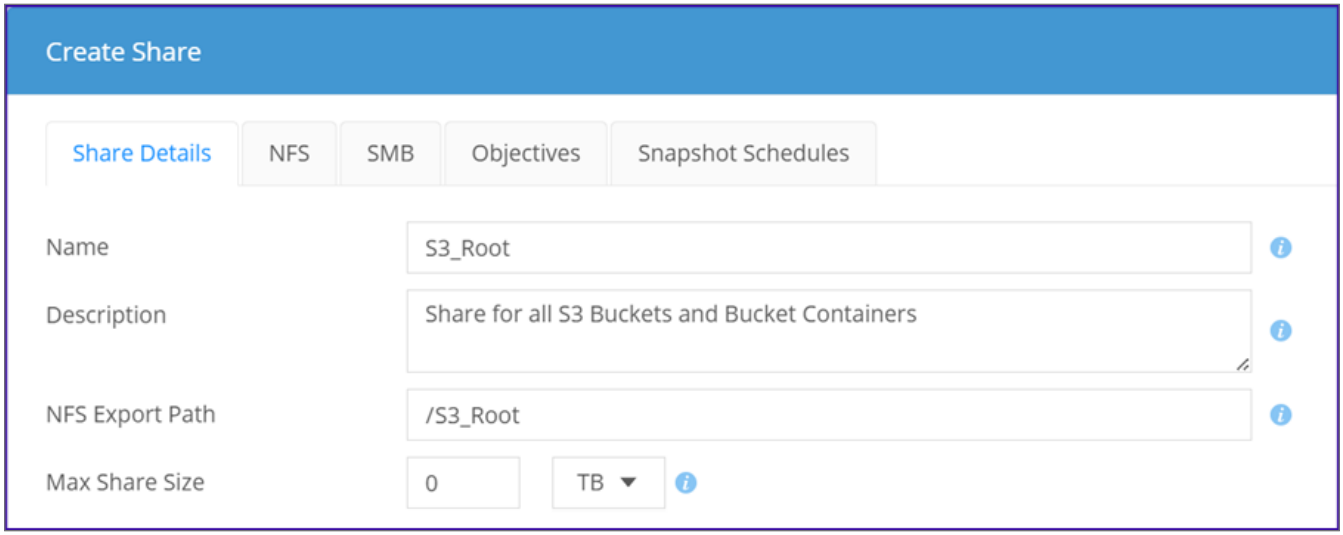


Figure 26. Naming the share and setting an optional share quota

If desired, set up a snapshot schedule for data protection using the Snapshot Schedules tab. The settings on the other tabs may be left at their defaults. Objectives will be customized after the share is created. Click **Create** to create the share. After a few moments it will appear in the shares list.

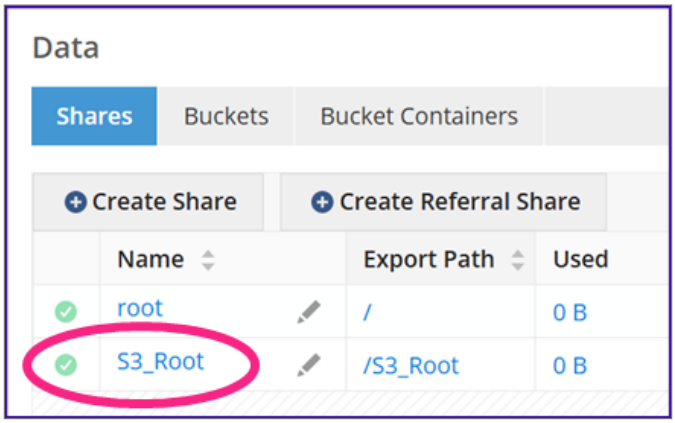


Figure 27. The S3_Root share in the shares list

7. Configure Share Objectives: Shares, like S3 servers, are logical entities. Objectives are used to control the storage volumes on which data written to the share will be stored. In this case, S3 servers are writing into the share, so the objectives determine which storage volumes will house incoming S3 objects from each S3 server.

   To edit the objectives for the S3 share created in the previous step, click the pencil icon in the Applied Objectives column.

© 2026 - Hammerspace

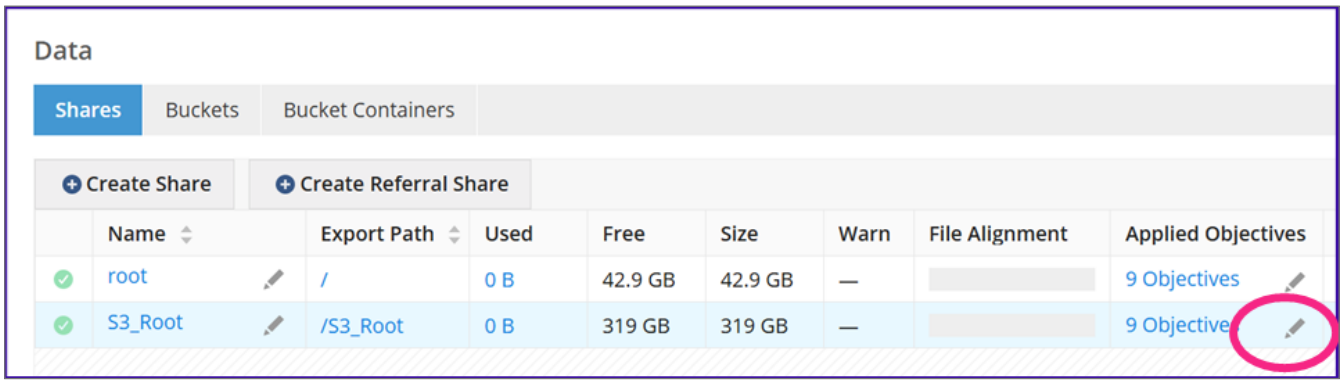


Figure 28. The pencil icon in the Applied Objectives column

On the Applicability: TRUE row of the table, click the pencil icon to edit those objectives.

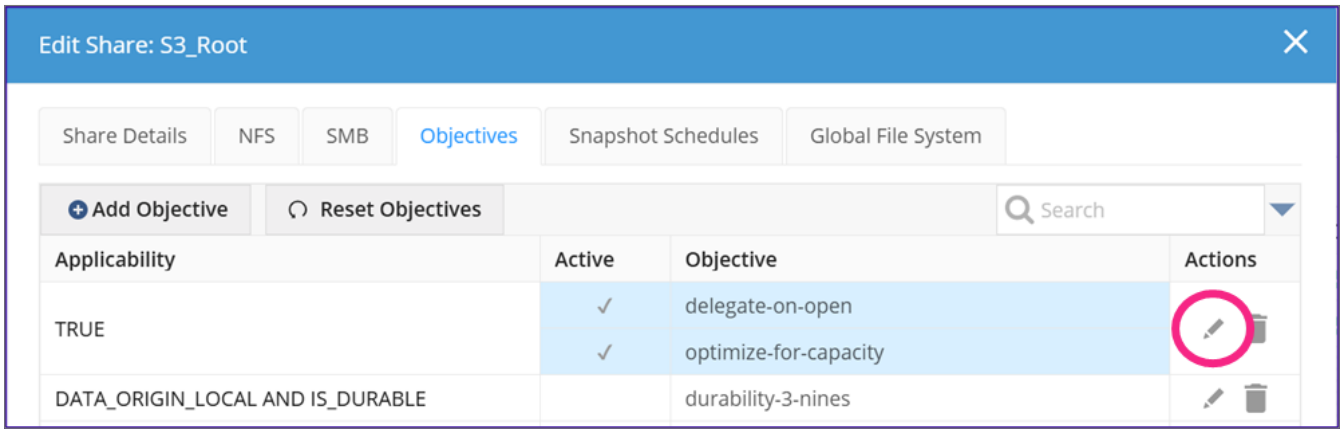


Figure 29. The pencil icon on the Applicability: TRUE row

In the Select Objectives section, un-check optimize-for-capacity to remove that objective. In this configuration, all objects will be written to the AWS cloud as a backup vs. using the cloud as a capacity tier.

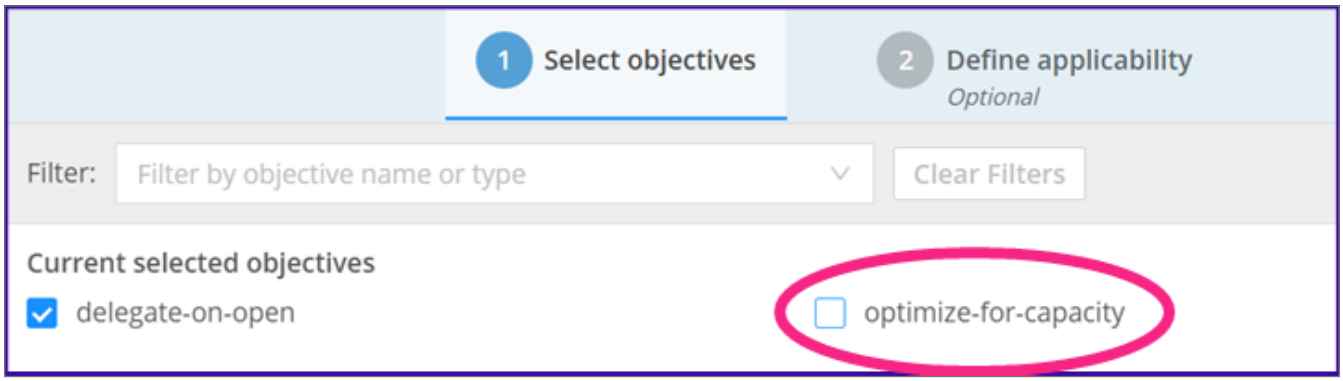


Figure 30. Removing the optimize-for-capacity objective

Scroll down and check the boxes next to the place-on-DSX volumes and place-on-object-volumes objectives. Note that place-on-object-volumes is not the same as place-on-shared-object-volumes! The two selected objectives ensure that one instance of each object will be placed on local DSX storage, and one will be placed in the cloud for DR.

© 2026 - Hammerspace

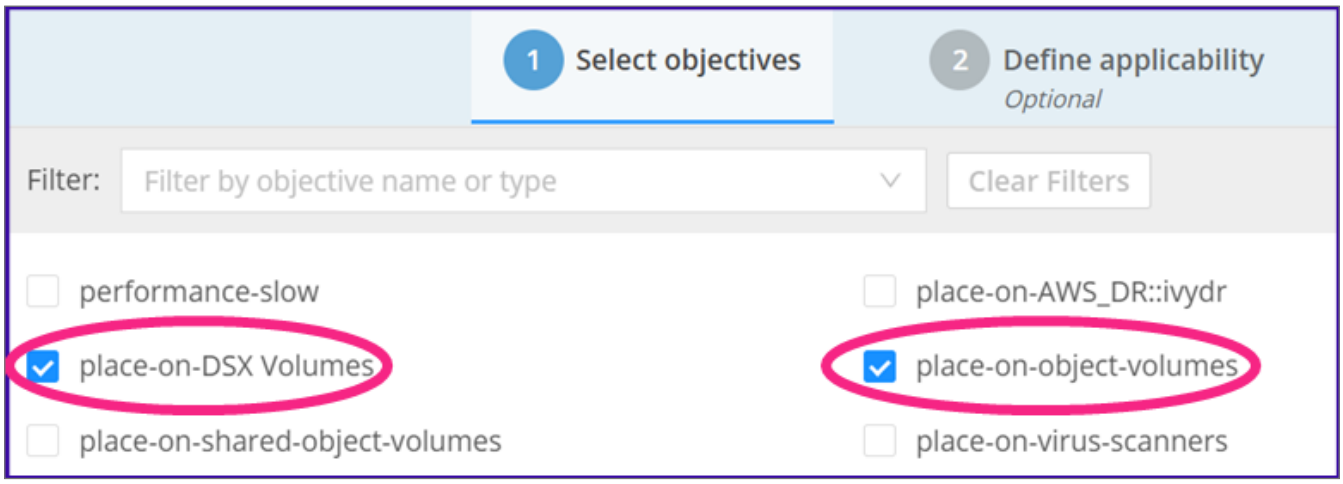


Figure 31. Selecting the place-on objectives

Click **Apply** to apply the changes. The Applicability: TRUE row in the table should now look like this:

Applicability	Active	Objective	Actions
TRUE	✓	delegate-on-open	
	✓	place-on-DSX Volumes	
	✓	place-on-object-volumes	

Figure 32. The Applicability: TRUE row after applying the objectives

Click **Close** to finalize the changes.

8. Configure S3 Servers: Now that the groundwork has been laid, the two S3 servers can be created. On the Administration page, S3 Servers tab, click **Create S3 Server**.

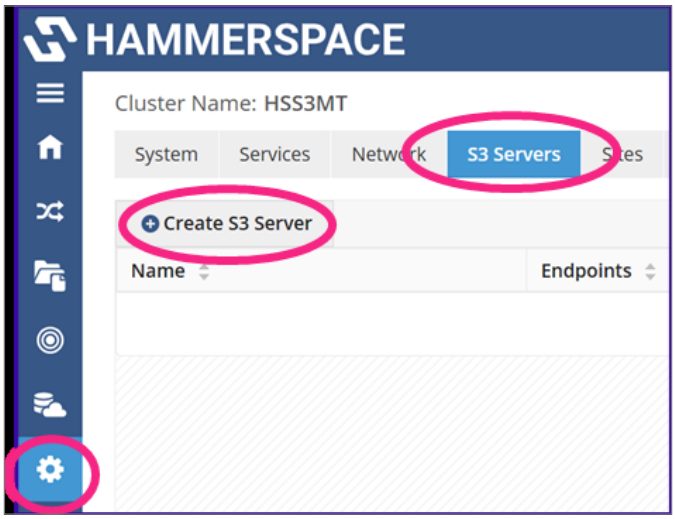


Figure 33. The Create S3 Server button on the S3 Servers tab

Give the S3 server an administrative name of MechE Dept S3. Un-check the Default Server checkbox, and ignore the warning. If a client attempts to connect to this server without specifying an endpoint, or using a non-existent endpoint, we want the attempt to be rejected. Specify an endpoint name of `s3.meche.ivy.edu`. This is how S3 clients will connect in our example. For the Identity Provider, select Local.

© 2026 - Hammerspace

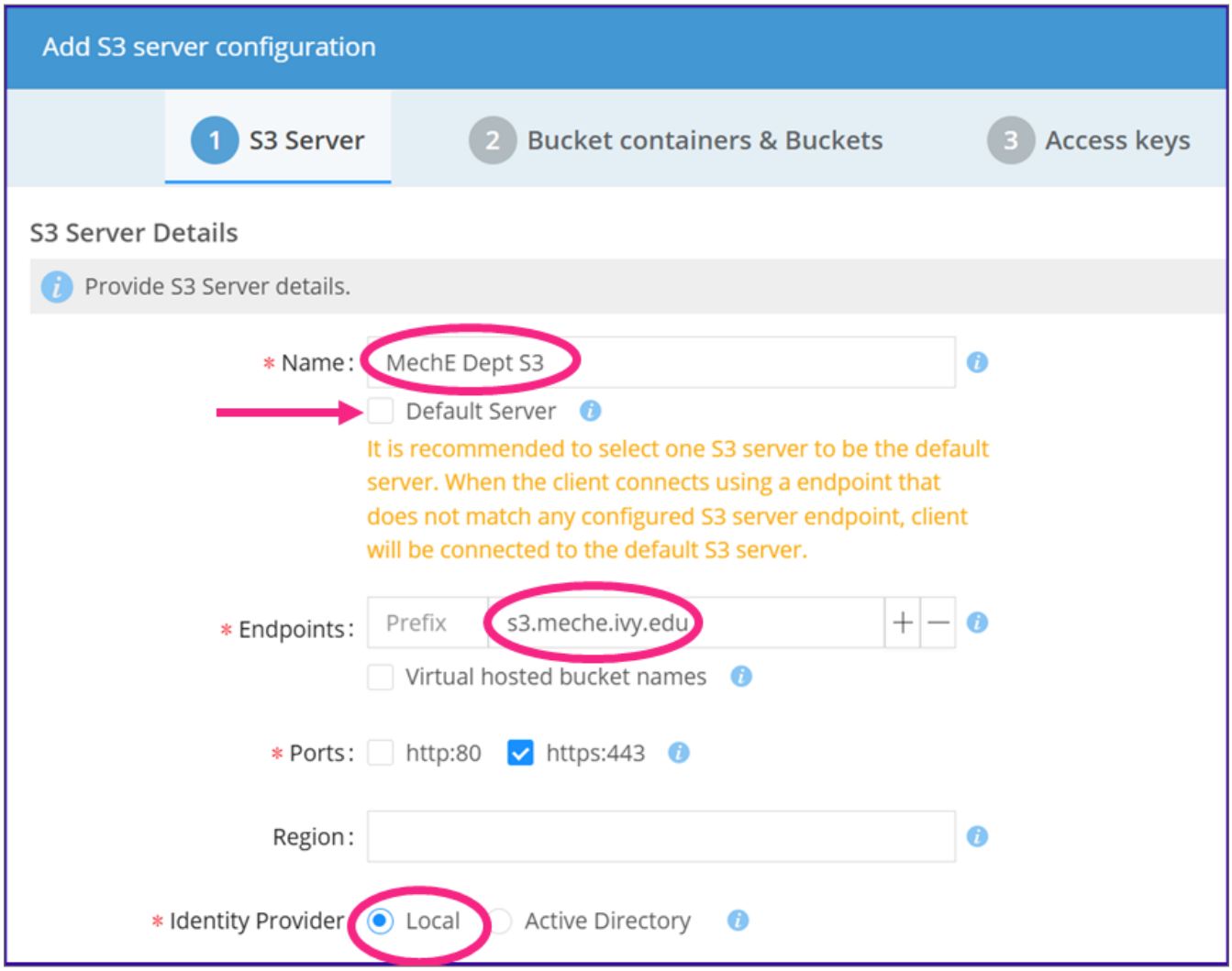


Figure 34. Creating the MechE Dept S3 server

Scroll down and set the remaining options as desired. The tooltips that appear when hovering over the blue information icons explain what they do. The most important is the Umask.

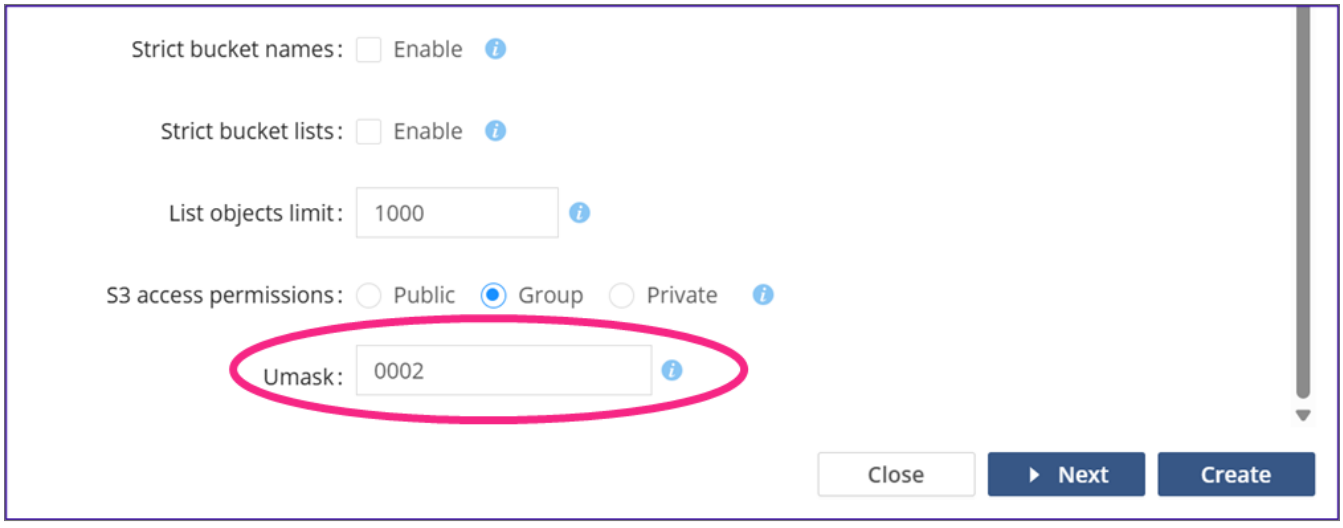


Figure 35. The remaining S3 server options

© 2026 - Hammerspace

The Umask determines the access permissions for objects when Local users are used, not AD. The default setting (0002) is equivalent to "Group." For a complete explanation of how Umask works, see the Appendix. For this use case the default is appropriate.

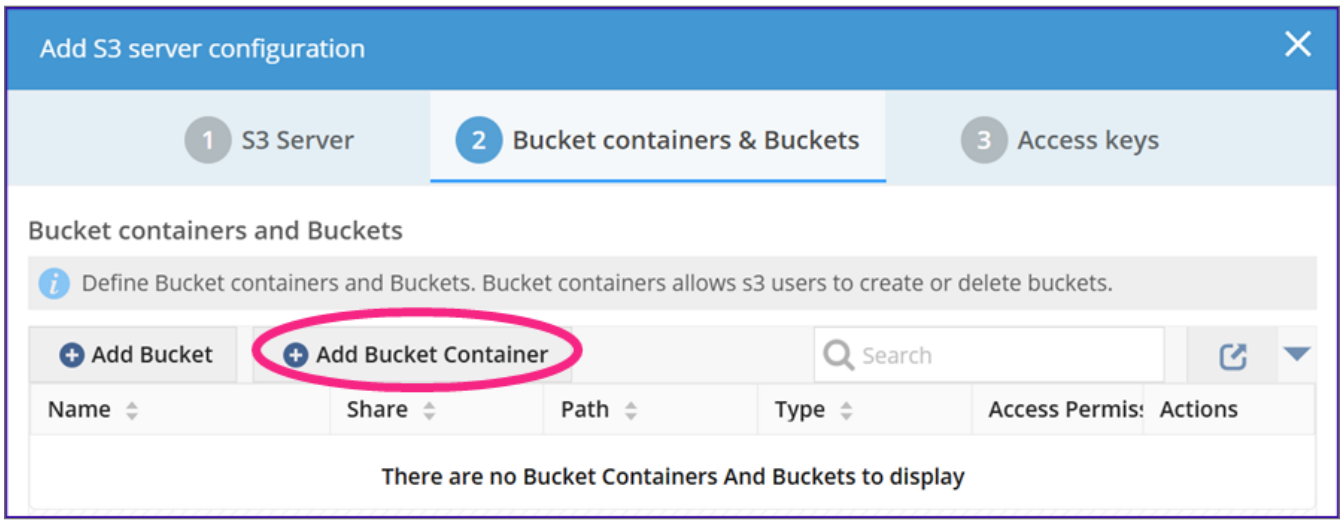


Figure 36. The Umask setting

Click **Next** to continue to the Bucket Container and Buckets step.

9. Create the Bucket Container: In this example we need S3 clients to be able to create buckets using S3 API commands. This requires a bucket container to be configured, which is the location where created buckets will live. Click **Add Bucket Container** to begin.

   On the next screen, fill in the Bucket container name Mech E Dept Buckets, choose the share previously created (S3_Root), and specify the bucket container path as `/MechE`. The directory `/S3_Root/MechE` will be created and represent the bucket container in the filesystem. When a bucket is created using S3 API commands, a corresponding subdirectory will be created in `/S3_Root/MechE`. For example, creation of a bucket named "Bucket1" will result in the directory `/S3_Root/MechE/Bucket1` being created. Objects stored in Bucket1 will become files in the Bucket1 directory.

   The rest of the options can remain at their defaults. Note however that the S3 server-level default S3 access permissions could be overridden here, and the change would apply to all buckets created in this bucket container. The ability for clients to create and delete S3 buckets may also be enabled or disabled here.

© 2026 - Hammerspace

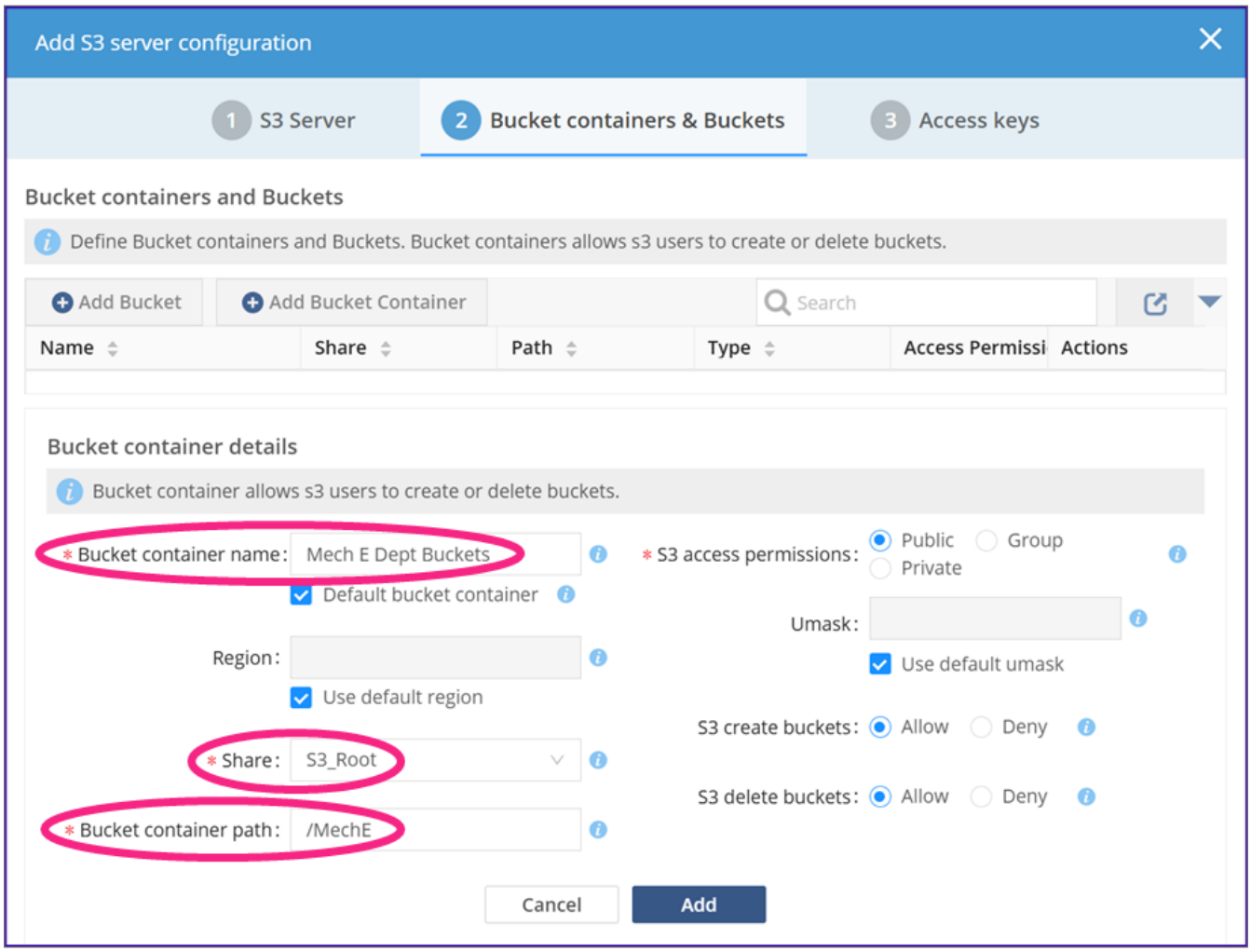


Figure 37. Adding the bucket container

Click **Add** to add the bucket container, which will be displayed in the table.

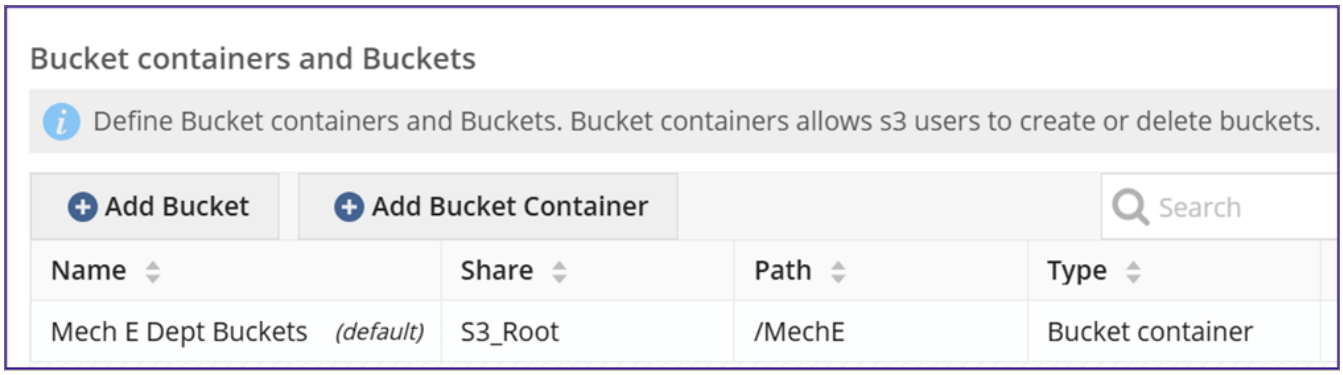


Figure 38. The bucket container displayed in the table

Click **Next** to continue to the next step, adding access keys.

10. Assign S3 Credentials: Users were created previously, in step 3. These users need access keys and secret keys to be able to authenticate to their S3 server. The assignment of these credentials happens in this step. Begin by clicking **Add Access Key**.

© 2026 - Hammerspace

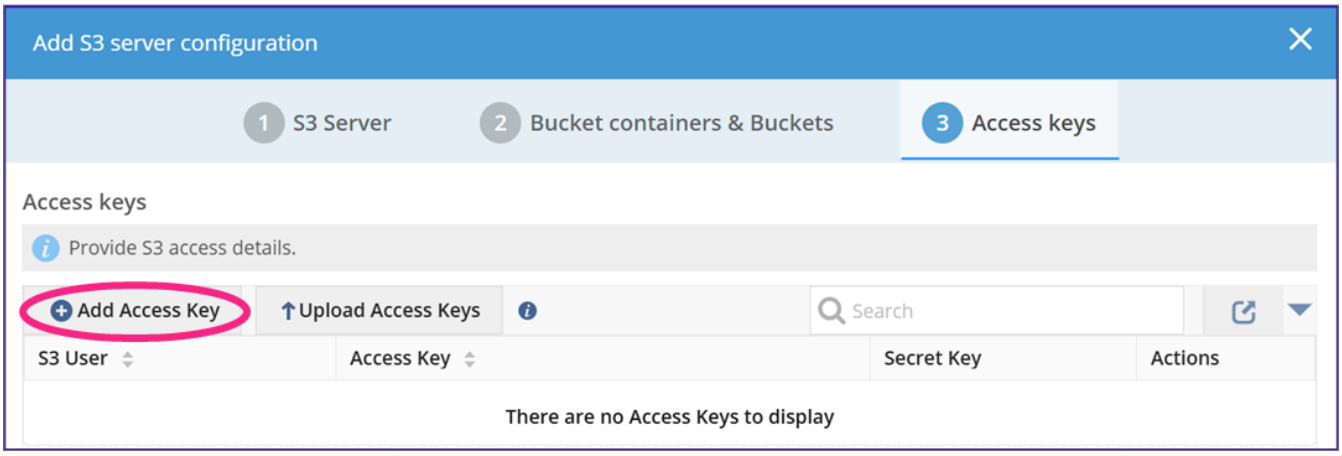


Figure 39. The Add Access Key button

On the next screen, choose the user 'meche' from the dropdown, and either leave the Access Key and Secret Key fields blank to have them automatically generated, or manually enter values.

Click **Add** to add the credentials, which will return you to the Access Keys step.

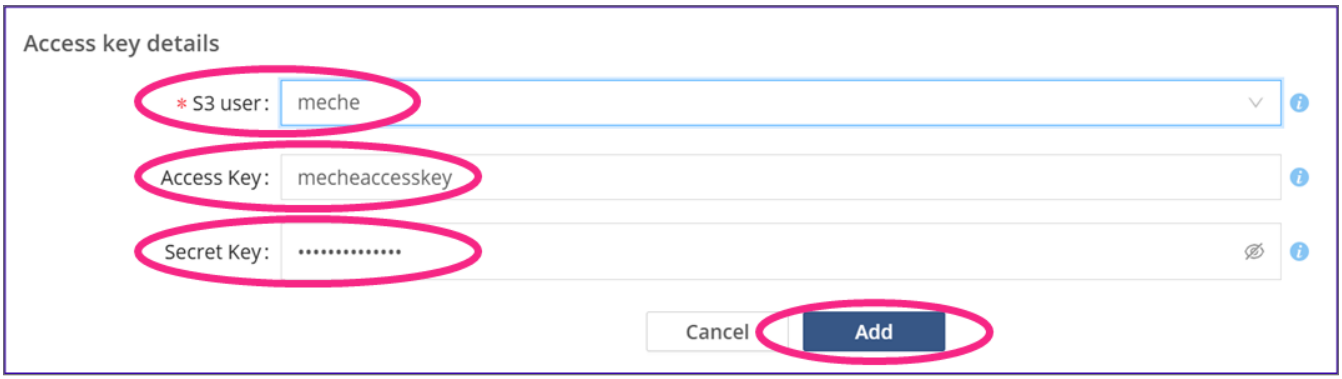


Figure 40. Adding an access key for the meche user

To finish creating the S3 server, including the bucket container and access keys, click **Create**. You can always edit the server later to add more buckets, bucket containers, or access keys, or to change the access permissions or other options.

Repeat steps 8, 9, and 10 to create another S3 server for the Chemical Engineering Dept.

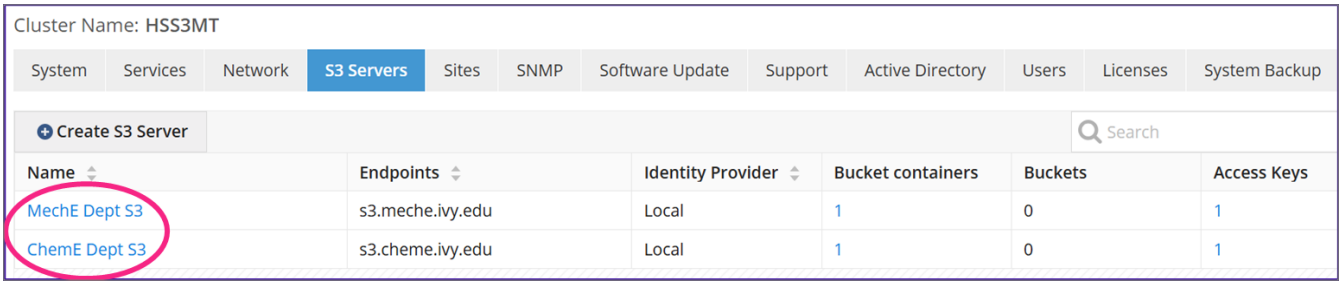


Figure 41. The two S3 servers, one per tenant

That completes the configuration for Example 1. Users 'meche' and 'cheme' can now use their S3 access key / secret key pairs and any S3 client to create buckets and store and retrieve objects from the Hammerspace S3 servers. Hammerspace will automatically create a secondary instance of every object in

© 2026 - Hammerspace

AWS for DR protection.

© 2026 - Hammerspace

Chapter 5. Example 2: Two Tenants, Isolated Resources

This example will configure a single Hammerspace instance for two tenants representing two completely independent corporate customers of a service provider. The tenants in this case must be kept as isolated from each other as possible. A service provider will configure, manage, and monitor the system.

5.1. Hammerspace Architecture

Three Anvil nodes are deployed, one in each of the three power domains at the service provider facility. Four DSX nodes are used, each with the S3 service enabled. If additional horsepower is needed, additional DSX nodes are easily deployed.

Back-end storage consists of Linux Storage Servers (LSSs) which are simply commodity Linux servers with attached HDD storage. The storage is exported using NFSv3 and will be configured as Hammerspace storage volumes. Objectives will ensure object data is maintained with the level of availability and durability specified in each tenant's contract.

5.2. Configuration Steps

The process to set up this example configuration consists of nine steps.

1. Enable S3 Services on all DSXs: On the GUI Administration page, switch to the Services tab and enable the S3 service on all four DSXs.

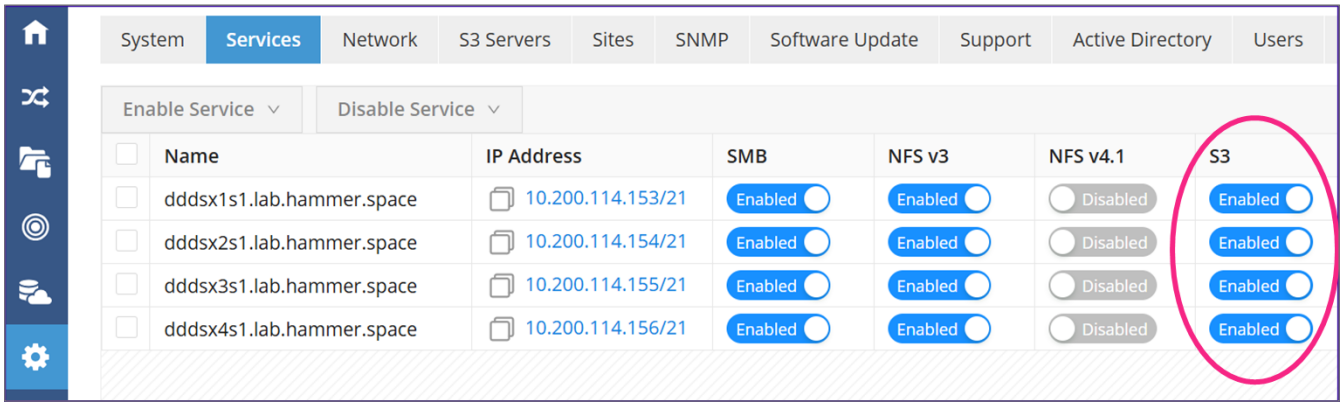


Figure 42. The S3 service enabled on all four DSX nodes

2. Create Users: If using Active Directory, go to the Active Directory tab to configure Hammerspace to connect to AD. Full instructions are contained in the Hammerspace Administration Guide. The instructions below are for creating local users.

   Still on the Administration page, switch to the Users tab and use the **Create Users** button to add at least one user for each tenant. Select the S3 Data Access Role.

© 2026 - Hammerspace

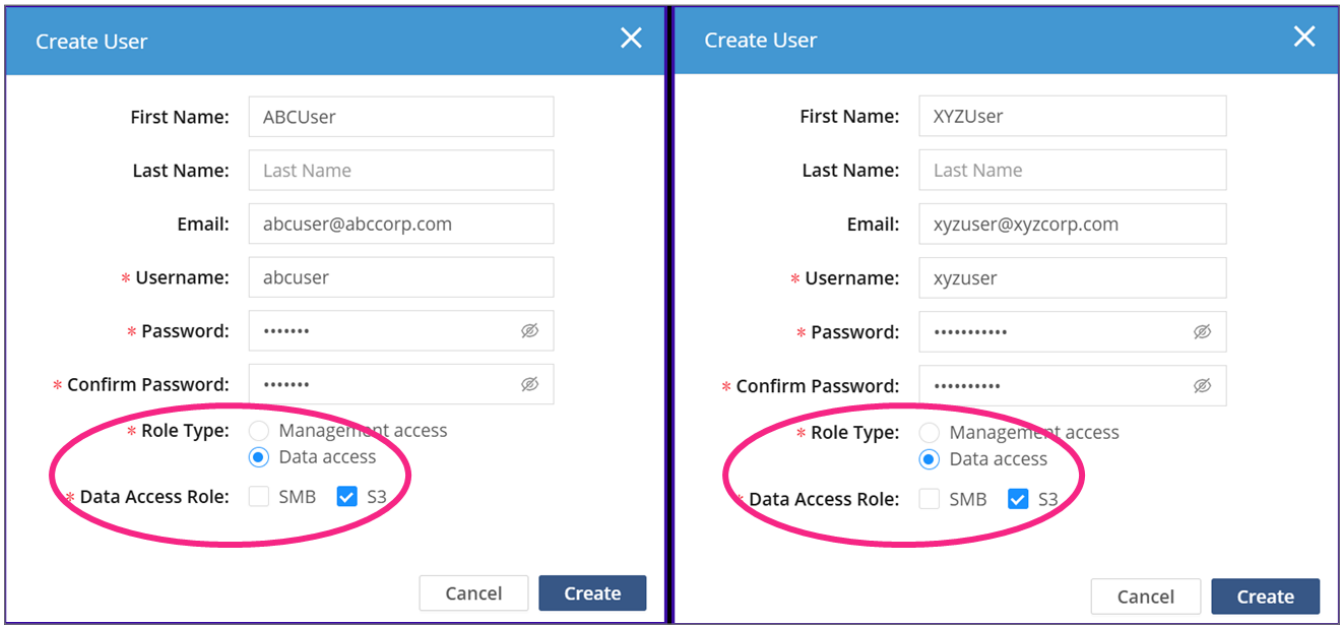


Figure 43. Creating a local user for each tenant with the S3 Data Access role

3. Add Storage Volumes: The NFSv3 exports on our Linux Storage Servers (LSSs) need to be configured as Hammerspace Storage Volumes. Go to the Infrastructure page, Storage Systems tab and click **Add Storage System**.

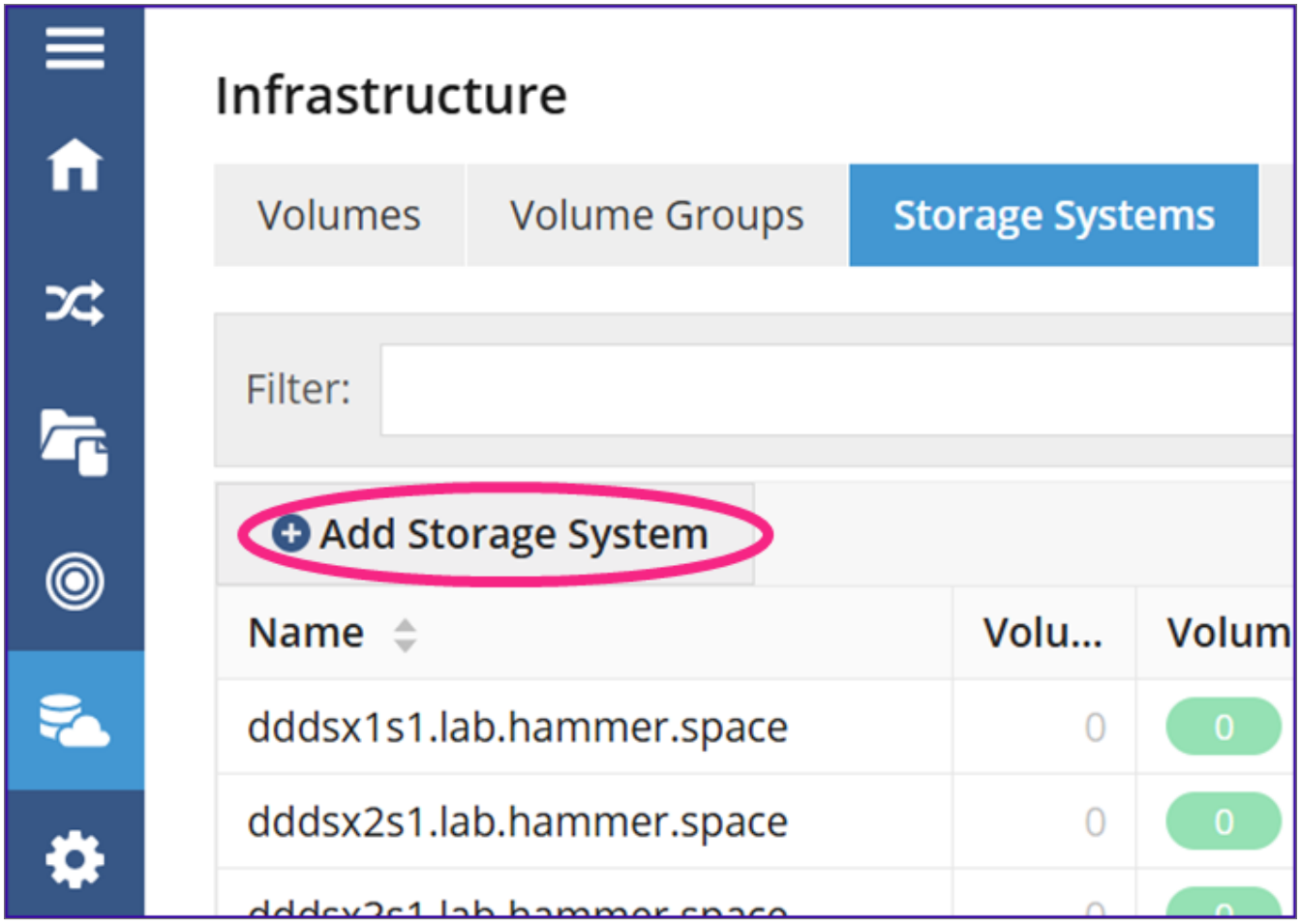


Figure 44. The Add Storage System button on the Storage Systems tab

© 2026 - Hammerspace

Fill in the fields, choosing NAS Storage as the storage type, and Other NFS for the NAS type. When finished, click **Add Storage System**.

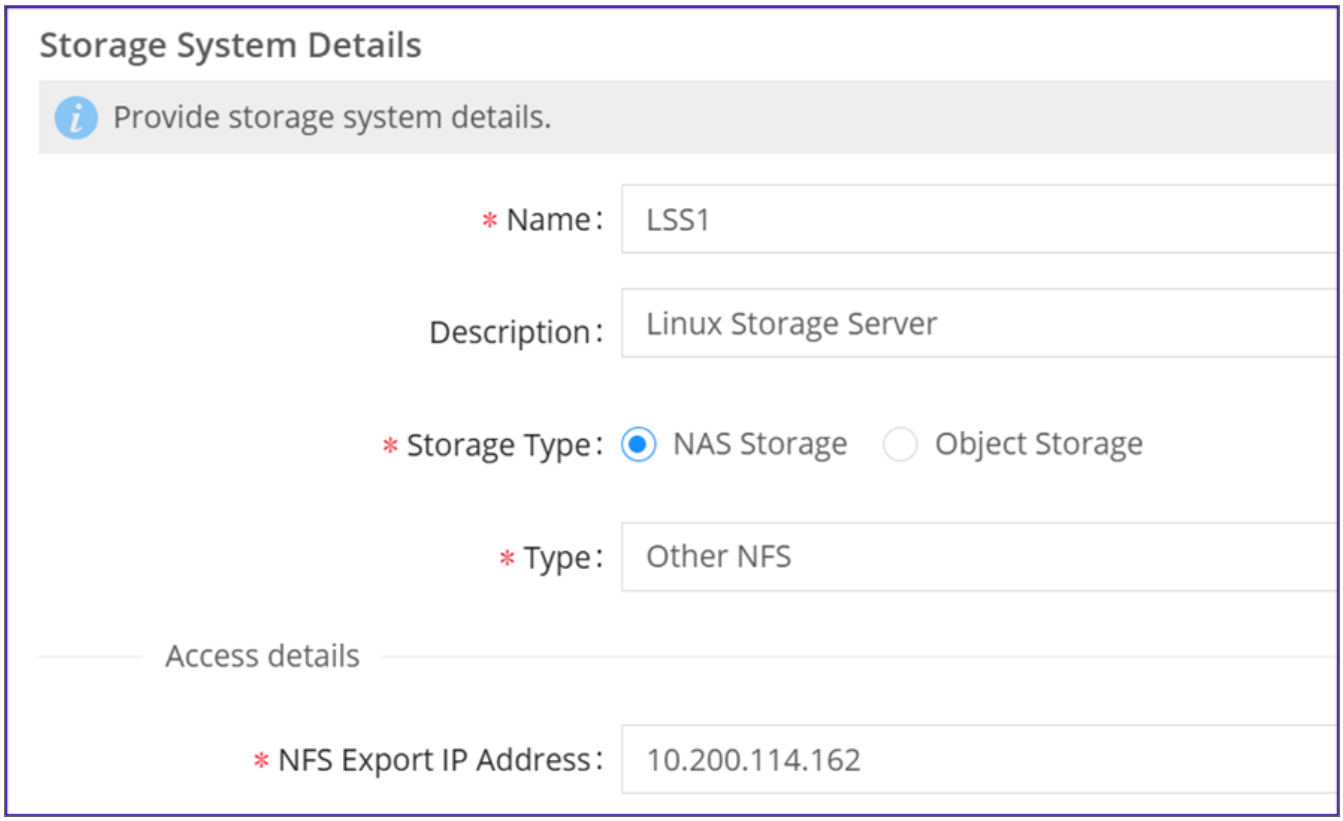


Figure 45. Adding a Linux Storage Server as a NAS storage system

At the completion dialog, click **Add Volumes**.

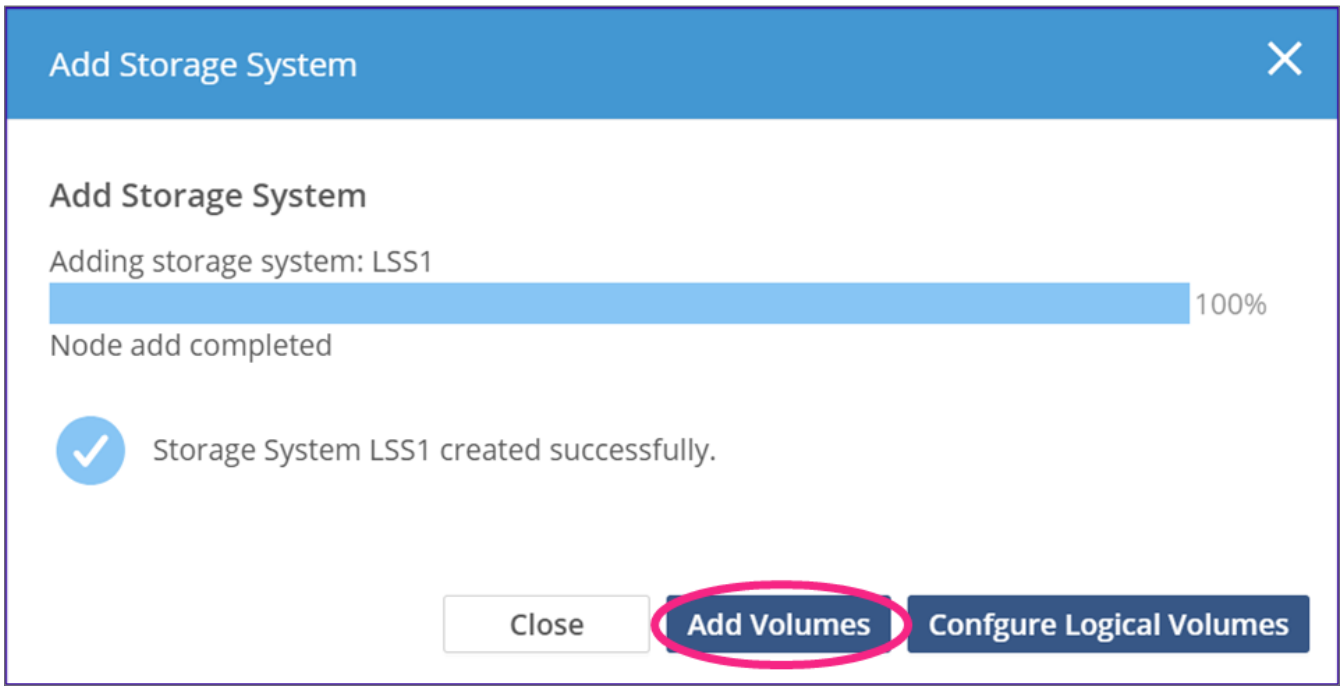


Figure 46. The completion dialog

© 2026 - Hammerspace

The next screen shows a list of exported volumes on this storage system, only one in this example. Check the checkbox next to the volume name and click **Next Step**.

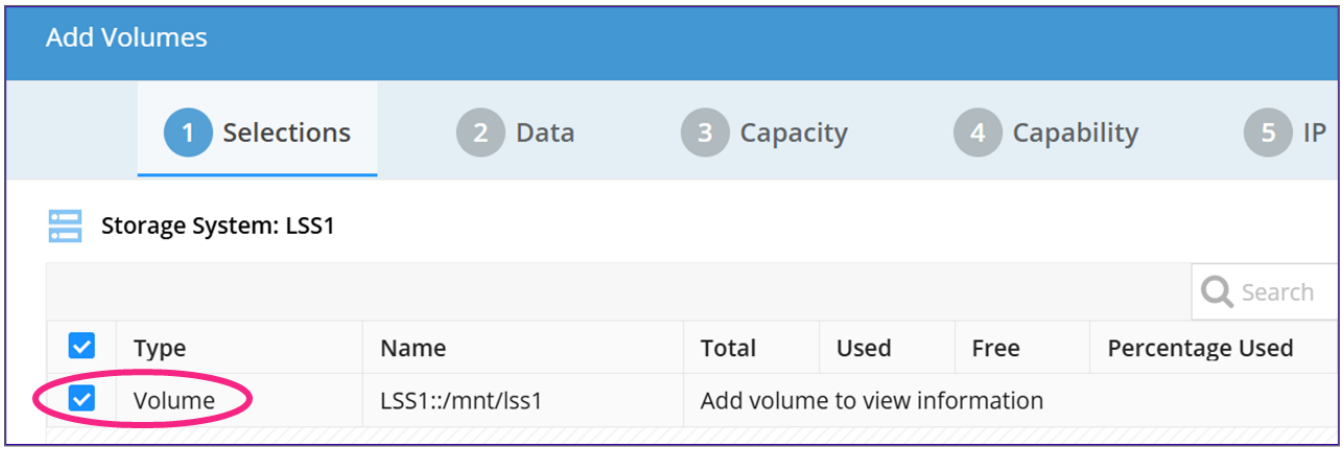


Figure 47. Selecting the exported volume

Since this share is empty and we don't need to customize it, continue clicking **Next Step** until you get to the last step, Review & Add. Consult the Hammerspace Administration Guide if you are curious about the various options you see in the intermediate steps.

On step 6, Review & Add, click **Add Volume** to add the volume to Hammerspace. Click **Add Storage System** again and repeat the steps to add the remaining three LSSs and their storage volumes. You will then see all LSSs and their stats shown on the Storage Systems tab.

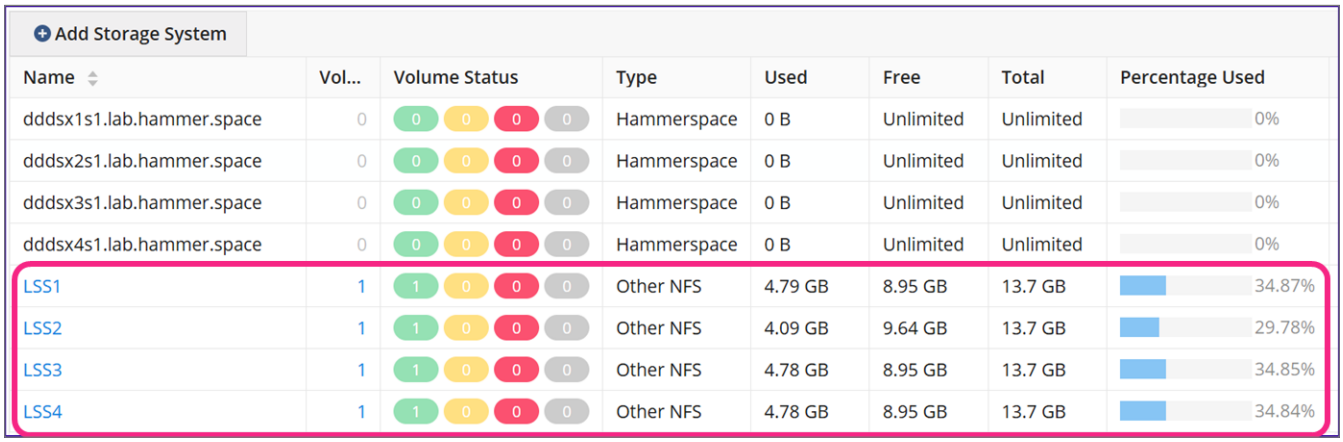


Figure 48. All four LSSs shown on the Storage Systems tab

4. Create Volume Groups: Configuring a volume group for each tenant will simplify the configuration of Objectives and make it easier to add more LSSs and storage volumes for each tenant later.

   On the Infrastructure page, Volume Groups tab, click **Create Volume Group**. Fill in the details in Step 1 as shown, then skip to Step 4.

© 2026 - Hammerspace

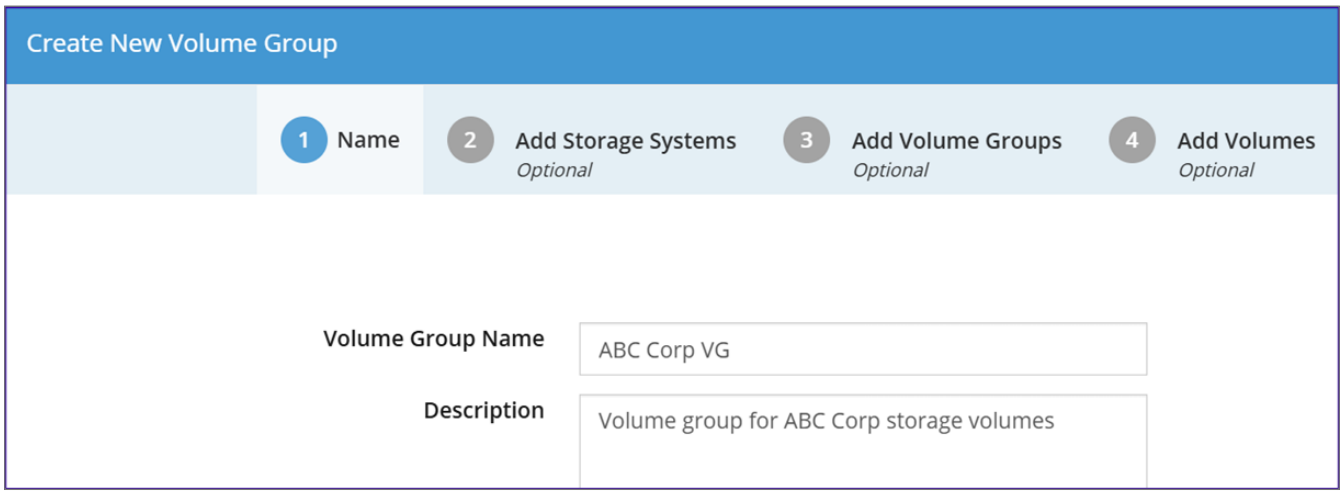


Figure 49. Volume group details in Step 1 of the wizard

In Step 4, select the first two storage volumes and click **Next**.

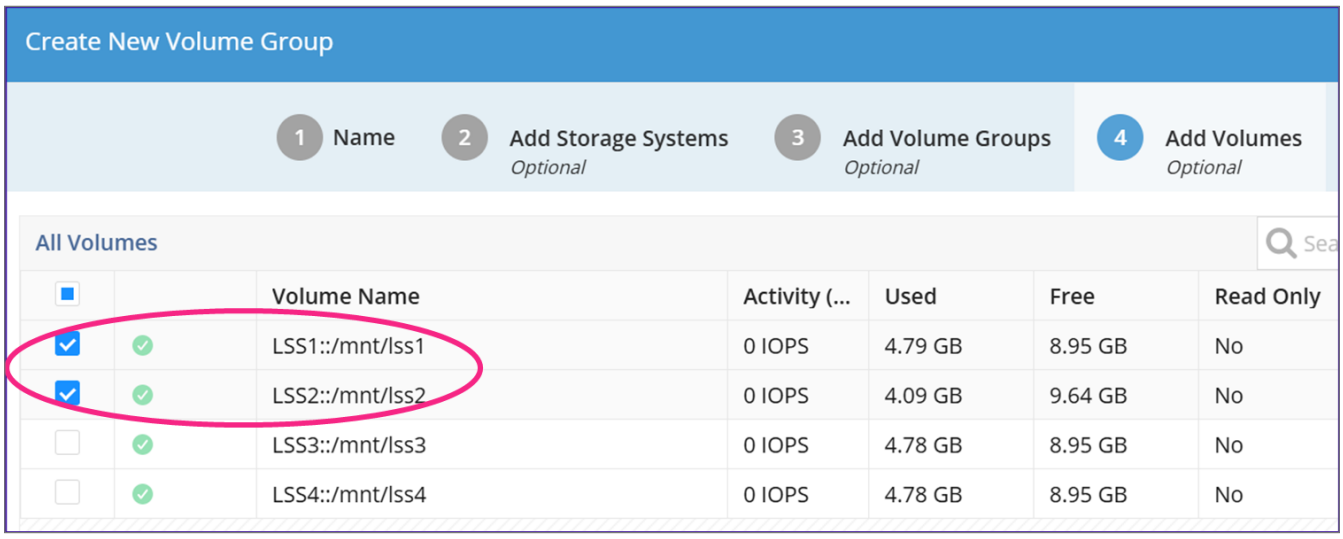


Figure 50. Selecting the first two storage volumes in Step 4

On Step 5, click the button to confirm and create the Volume Group. Repeat the steps and create a second volume group for XYZ Corp using the remaining two volumes.

© 2026 - Hammerspace

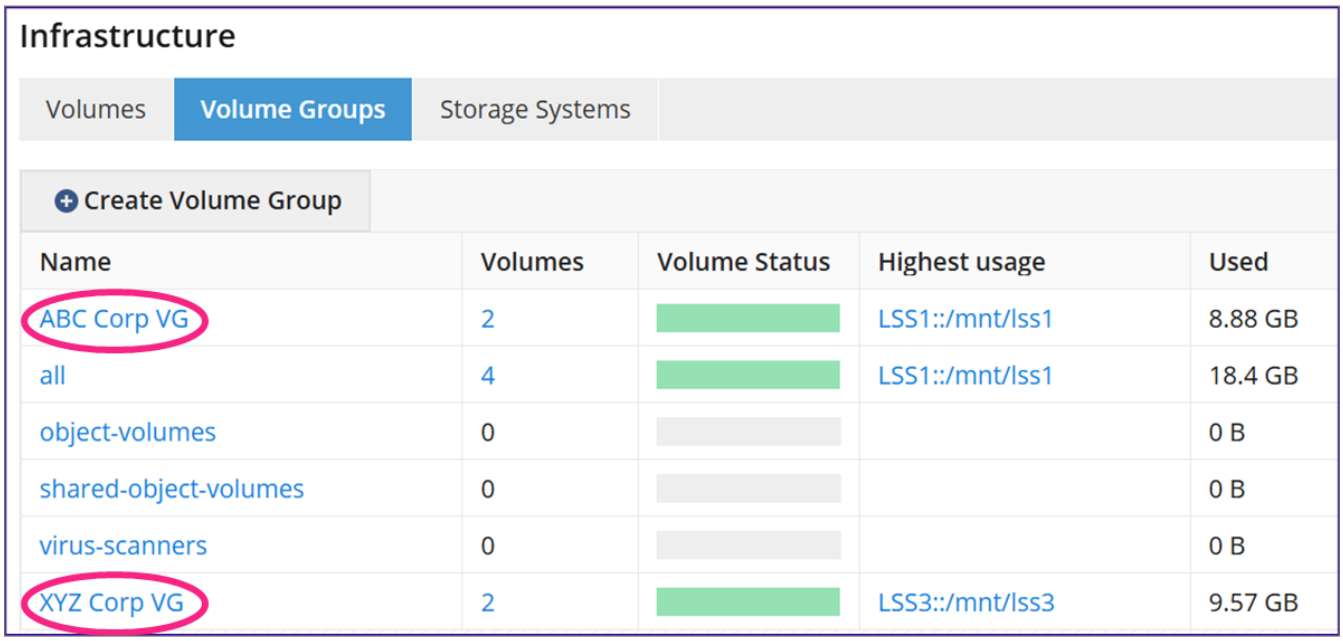


Figure 51. Confirming the volume group

When more storage volumes are added for a tenant, adding them to the appropriate Volume Group is all that needs to be done.

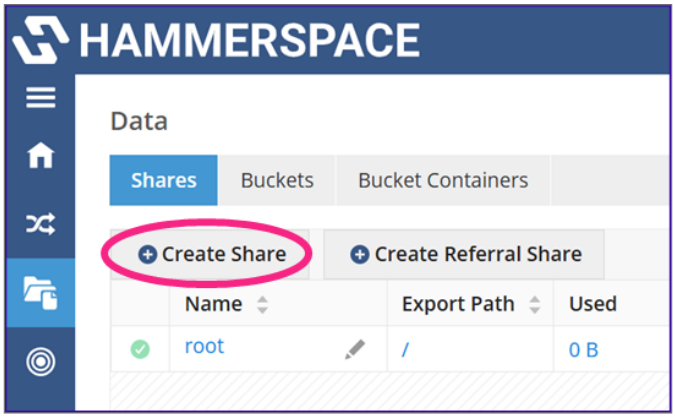


Figure 52. One volume group per tenant

5. Create Shares: Hammerspace S3 servers require a target share, so before creating any S3 servers, shares must be configured. For this example, it makes sense to create one share per tenant. On the Data page, Shares tab, click **Create Share** to begin the process.

   Give the share a name and description, and if desired set a limit on the maximum size of the share (a share quota).

© 2026 - Hammerspace

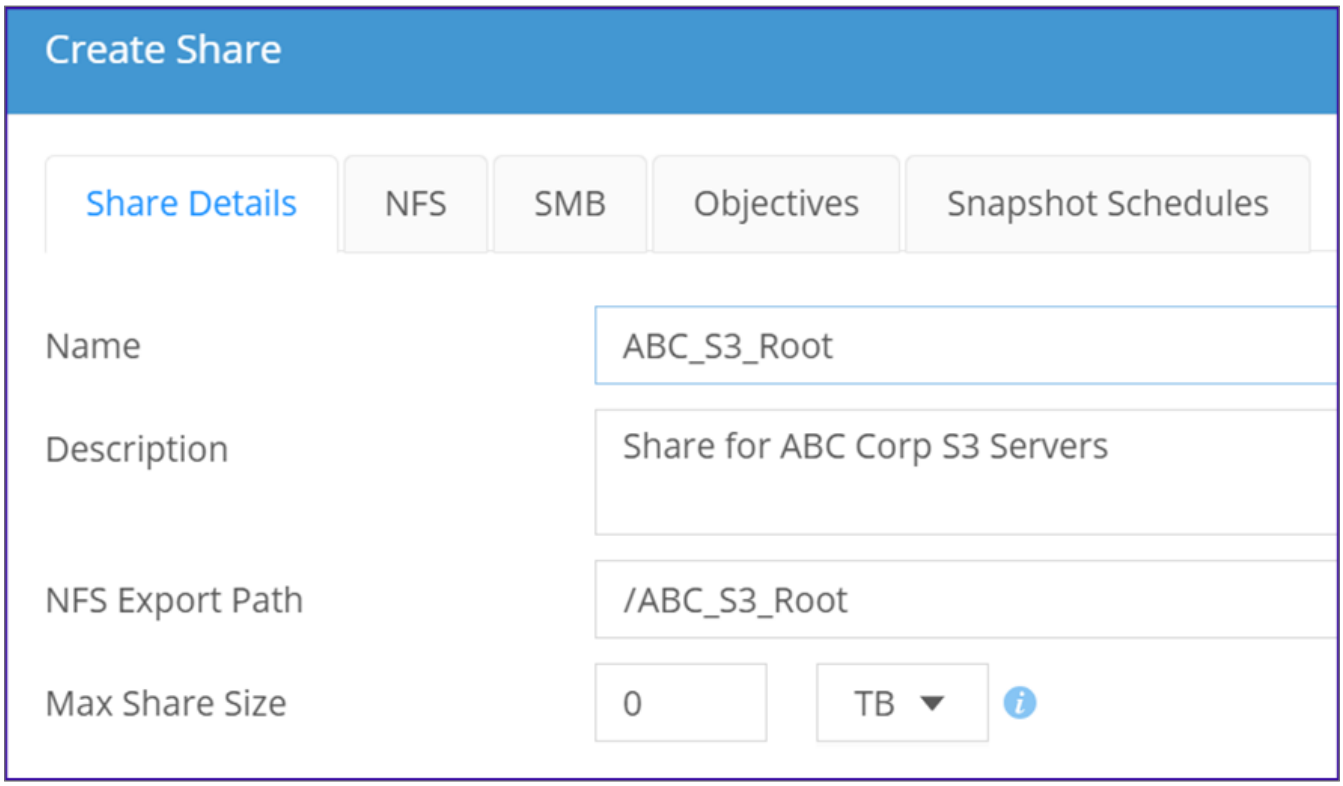


Figure 53. Naming the tenant share and setting an optional share quota

If desired, set up a snapshot schedule for data protection using the Snapshot Schedules tab. The settings on the other tabs may be left at their defaults. Objectives will be customized after the share is created. Click **Create** to create the share. After a few moments it will appear in the shares list. Follow the same process to create a share for the second tenant.

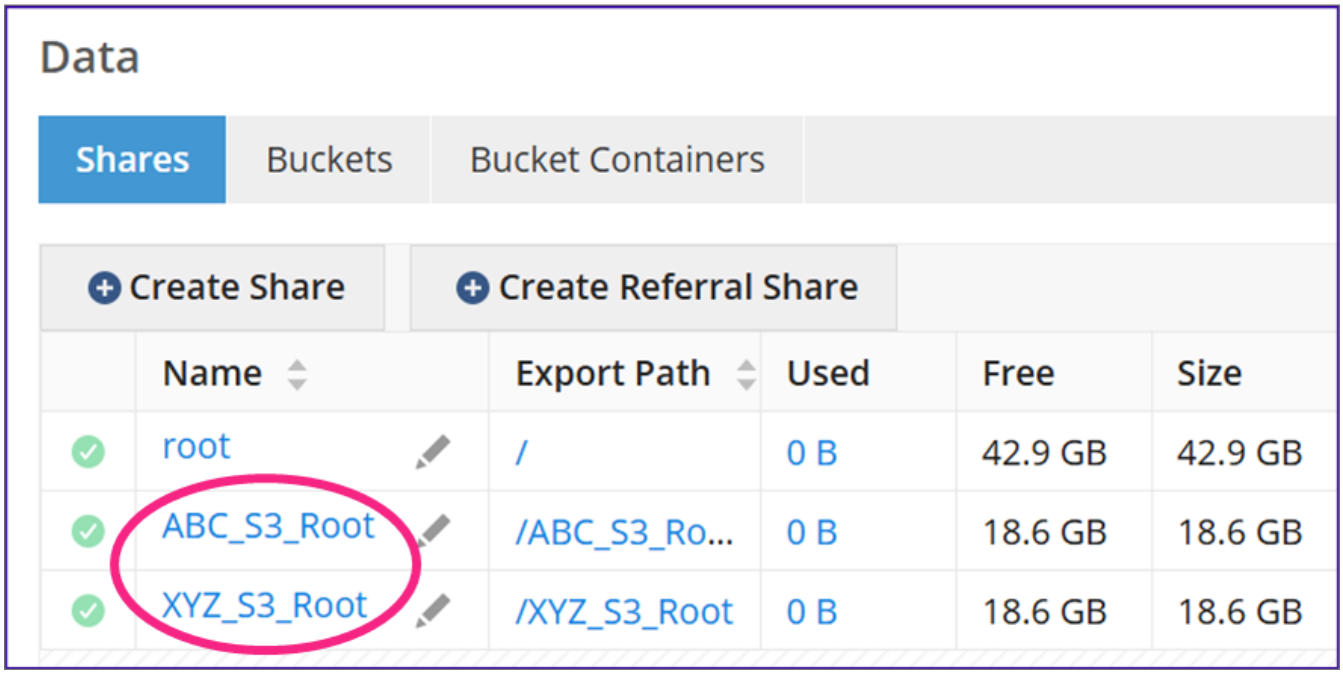


Figure 54. One share per tenant in the shares list

6. Configure Share Objectives: Shares, like S3 servers, are logical entities. Objectives are used to control the storage volumes on which data written to the share will be stored. In this case, S3 servers are writing into

© 2026 - Hammerspace

the share, so the objectives determine which storage volumes will house incoming S3 objects from each S3 server.

To edit the objectives for the S3 shares created in the previous step, click the pencil icon in the Applied Objectives column.

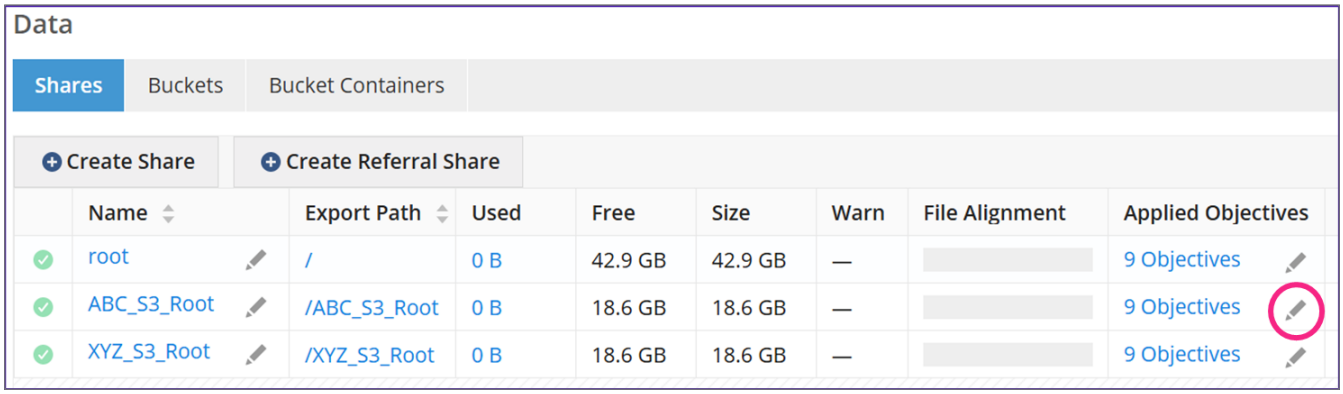


Figure 55. The pencil icon in the Applied Objectives column

On the Applicability: TRUE row of the table, click the pencil icon to edit those objectives.

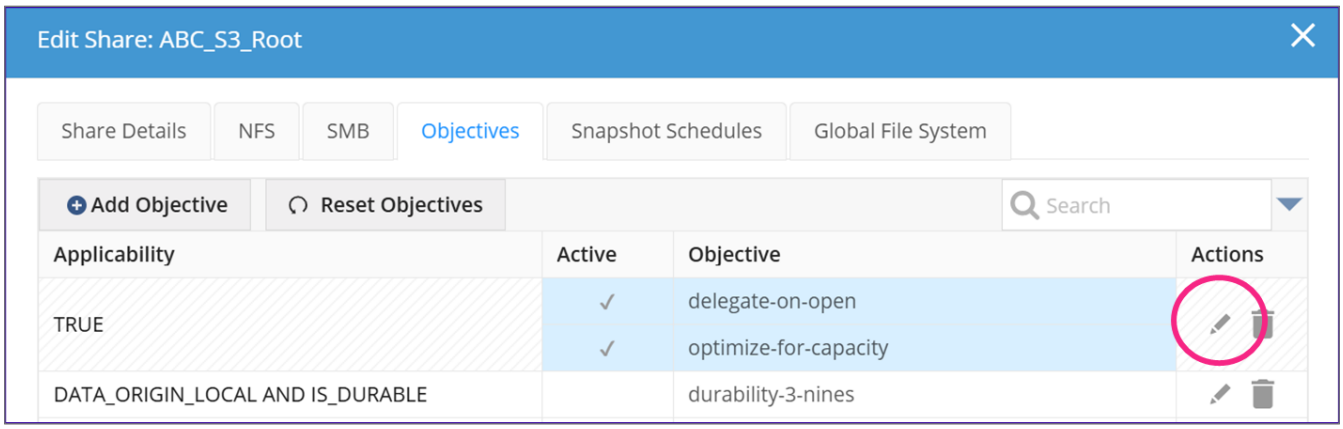


Figure 56. The pencil icon on the Applicability: TRUE row

Scroll down and in the Available Objectives section check the box next to the confine-to-ABC Corp VG objective. This directs Hammerspace to store all data that is written into the ABC_S3_Root share only on storage volumes that are members of the ABC Corp VG volume group. Also click the checkbox for the availability-3-nines objective and the durability-5-nines objective (not shown). This will ensure that Hammerspace maintains two instances of each file on different hosts for data protection.



This is a very basic example. Hammerspace Objectives may be configured to enable nearly any desired data protection scheme imaginable. Consult the Objectives Guide for more information regarding objectives.

© 2026 - Hammerspace

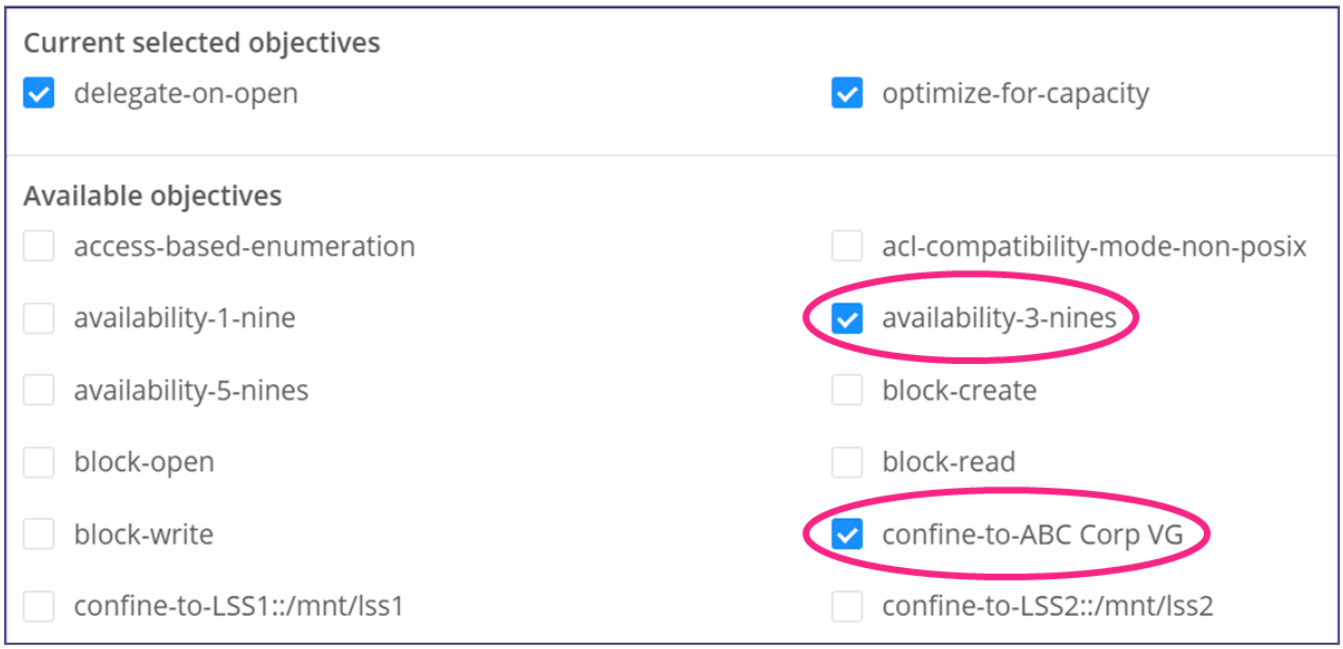


Figure 57. Selecting the confine-to objective for the tenant's volume group

Click **Apply** to apply the changes. The Applicability: TRUE row in the table should now look like this:

Applicability	Active	Objective	Actions
TRUE	✓	availability-3-nines	
	✓	confine-to-XYZ Corp VG	
	✓	delegate-on-open	
	✓	durability-5-nines	
	✓	optimize-for-capacity	

Figure 58. The Applicability: TRUE row after applying the objectives

Click **Close** to finalize the changes, and repeat the steps with the XYZ_S3_Root share, using the confine-to-XYZ Corp VG objective.

7. Configure S3 Servers: Now that the groundwork has been laid, the two S3 servers can be created. On the Administration page, S3 Servers tab, click **Create S3 Server**.

© 2026 - Hammerspace

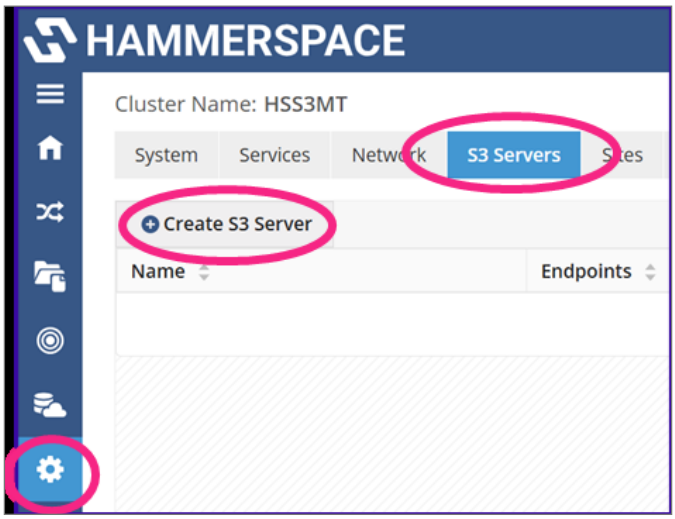


Figure 59. The Create S3 Server button on the S3 Servers tab

Give the S3 server an administrative name of ABC Corp S3. Un-check the Default Server checkbox, and ignore the warning. If a client attempts to connect to this server without specifying an endpoint, or using a non-existent endpoint, we want the attempt to be rejected. Specify an endpoint name of `s3.abccorp.com`. This is how S3 clients will connect in our example. For the Identity Provider, select Local.

© 2026 - Hammerspace

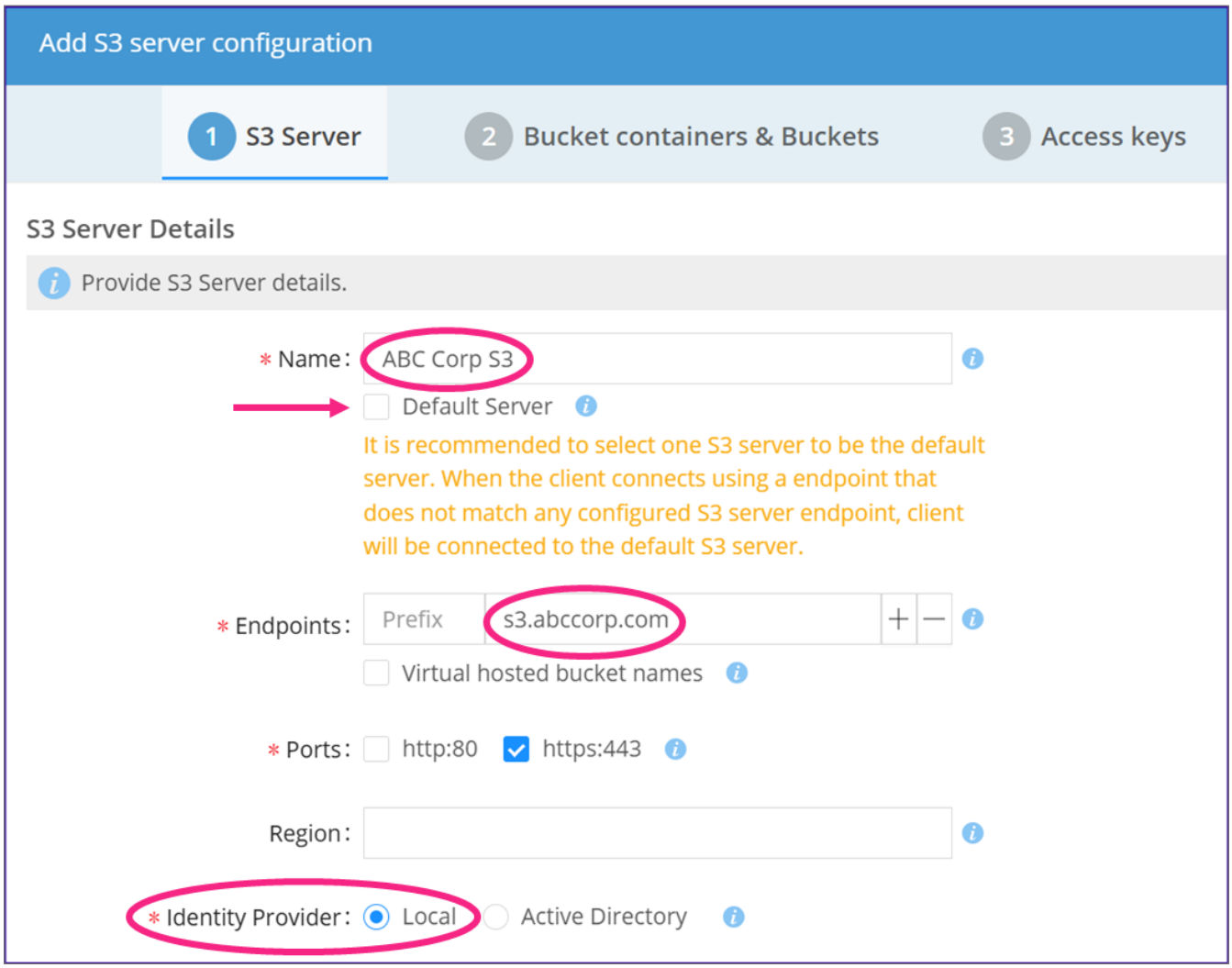


Figure 60. Creating the ABC Corp S3 server

Scroll down and set the remaining options as desired. The tooltips that appear when hovering over the blue information icons explain what they do. The most important is the Umask.

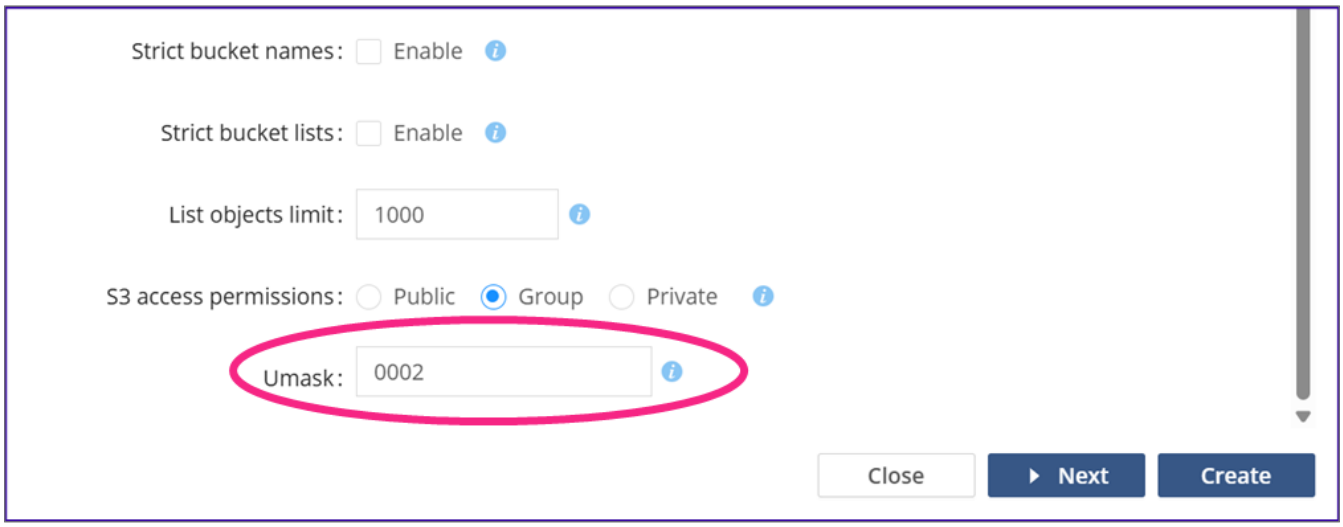


Figure 61. The remaining S3 server options

© 2026 - Hammerspace

The Umask determines the access permissions for objects when Local users are used, not AD. The default setting (0002) is equivalent to "Group." For a complete explanation of how Umask works, see the Appendix. For this use case the default is appropriate.

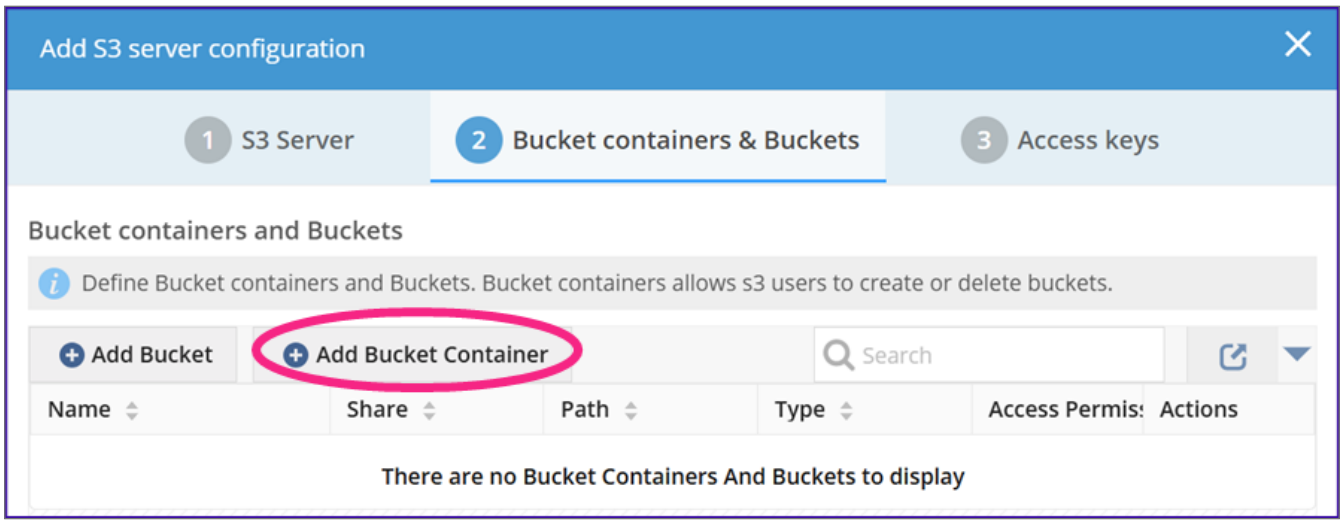


Figure 62. The Umask setting

Click **Next** to continue to the Bucket Container and Buckets step.

8. Create the Bucket Containers: In this example we need S3 clients to be able to create buckets using S3 API commands. This requires a bucket container to be configured, which is the location where created buckets will live. Click **Add Bucket Container** to begin.

   On the next screen, fill in the Bucket container name ABC Corp Buckets, choose the share previously created (ABC_S3_Root), and specify the bucket container path as `/Buckets`. The directory `/ABC_S3_Root/Buckets` will be created and represent the bucket container in the filesystem. When a bucket is created using S3 API commands, a corresponding subdirectory will be created in `/ABC_S3_Root/Buckets`. For example, creation of a bucket named "Bucket1" will result in the directory `/ABC_S3_Root/Buckets/Bucket1` being created. Objects stored in Bucket1 will become files in the Bucket1 directory.

   Check the Default bucket container checkbox. The rest of the options can remain at their defaults. Note however that the S3 server-level default S3 access permissions could be overridden here, and the change would apply to all buckets created in this bucket container. The ability for clients to create and delete S3 buckets may also be enabled or disabled here.

© 2026 - Hammerspace

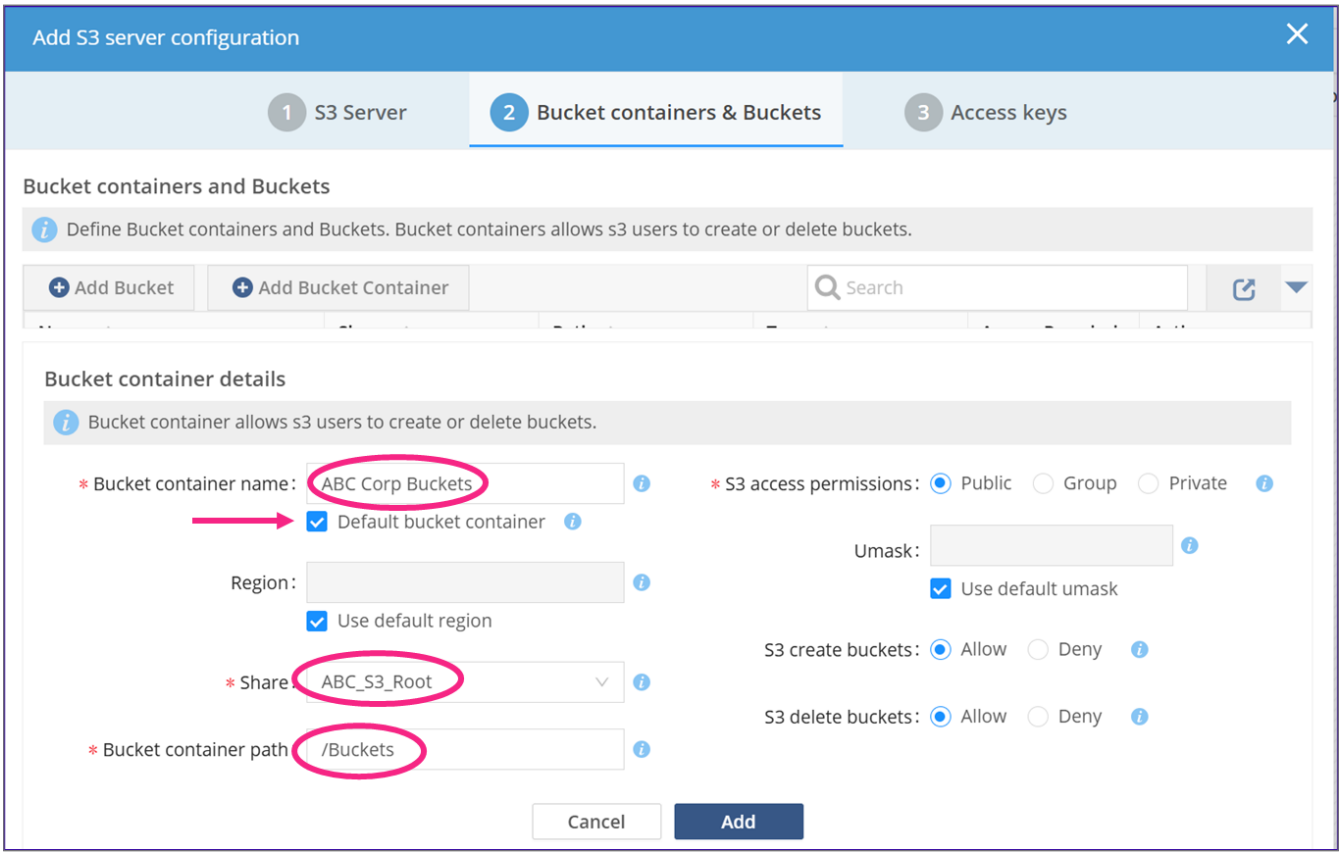


Figure 63. Adding the bucket container

Click **Add** to add the bucket container, which will be displayed in the table.

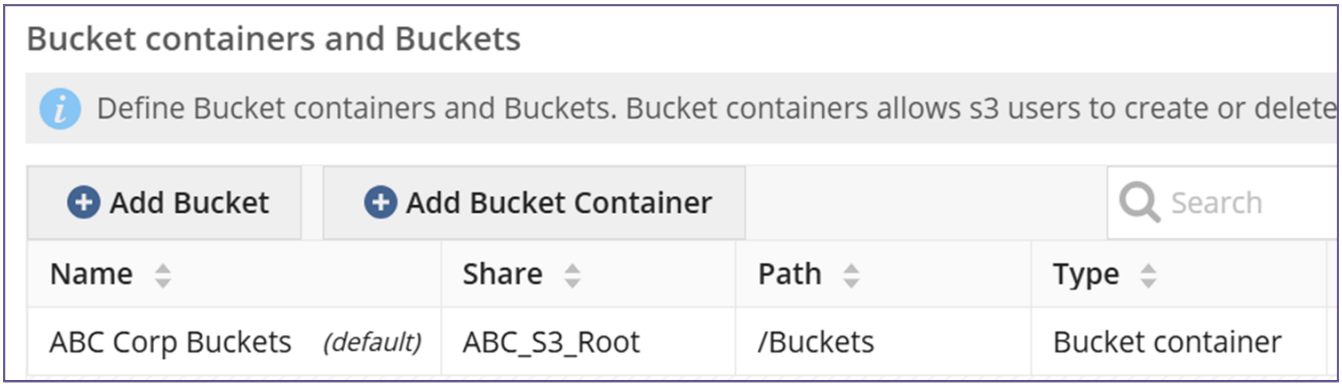


Figure 64. The bucket container displayed in the table

Click **Next** to continue to the next step, adding access keys.

9. Assign S3 Credentials: Users were created previously, in step 2. These users need access keys and secret keys to be able to authenticate to their S3 server. The assignment of these credentials happens in this step. Begin by clicking **Add Access Key**.

© 2026 - Hammerspace

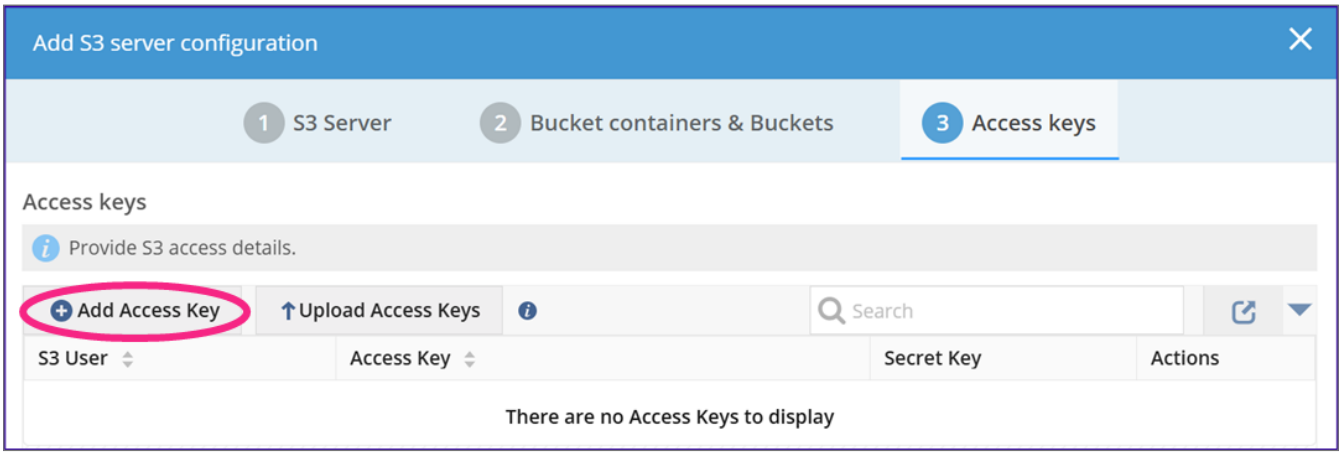


Figure 65. The Add Access Key button

On the next screen, choose the user 'abcuser' from the dropdown, and either leave the Access Key and Secret Key fields blank to have them automatically generated, or manually enter values.

Click **Add** to add the credentials, which will return you to the Access Keys step.

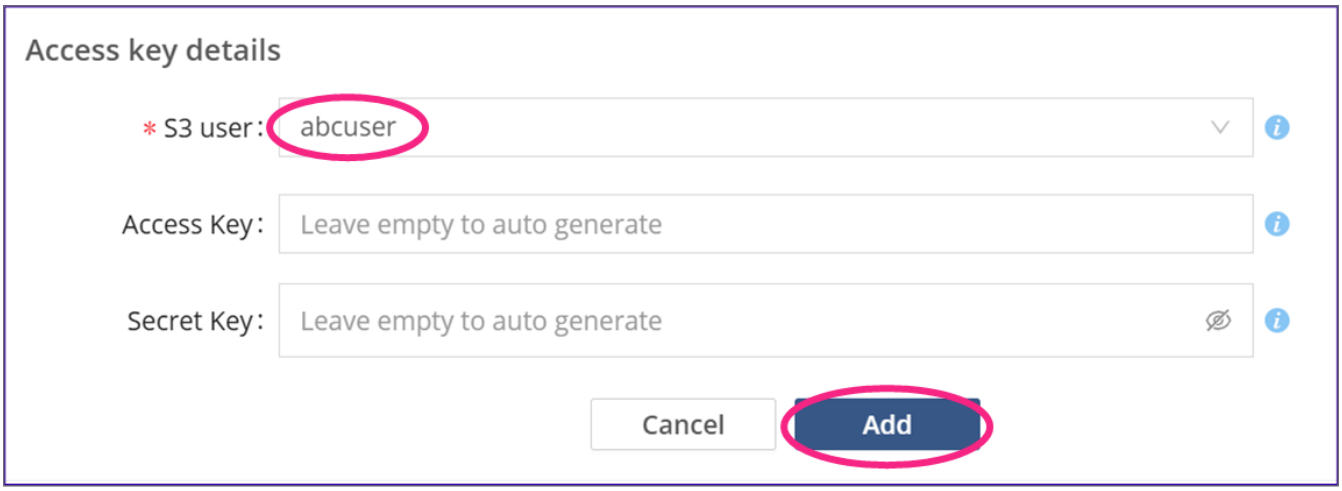


Figure 66. Adding an access key for the abcuser user

To finish creating the S3 server, including the bucket container and access keys, click **Create**. You can always edit the server later to add more buckets, bucket containers, or access keys, or to change the access permissions or other options.

Repeat steps 7, 8, and 9 to create another S3 server for tenant XYZ Corp.

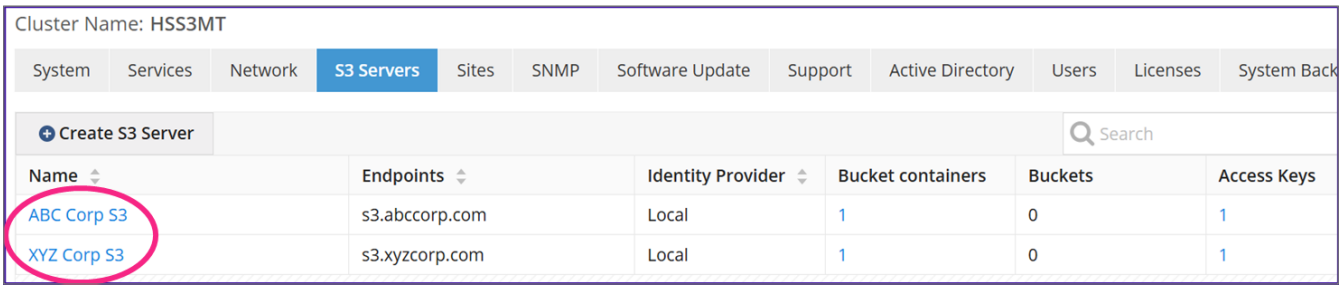


Figure 67. The two isolated tenant S3 servers

© 2026 - Hammerspace

That completes the configuration for Example 2. The two tenants, ABC Corp and XYZ Corp, access separate S3 servers that each use isolated back-end storage. Basic data protection is provided by objectives that ensure two instances of every object exist in separate storage servers. Administration at the service provider is centralized, and the underlying storage infrastructure may be scaled or modified at any time with zero disruption to users.

© 2026 - Hammerspace

Appendices

© 2026 - Hammerspace

Appendix A: Appendix: Determining Umask Values

This appendix describes Umask values and how to calculate them. Umasks are used to define access permissions in Hammerspace when local users are employed instead of Active Directory.

A.1. Unix Permissions - Octal Values

The following table shows the conversion of Unix permissions from their binary value to an octal value, which is needed to determine the desired Umask setting.

Permissions	Octal Value	Binary Value	Description
`---`	0	000	No permissions
`--x`	1	001	Execute
`-w-`	2	010	Write
`-wx`	3	011	Write and execute
`r--`	4	100	Read
`r-x`	5	101	Read and execute
`rw-`	6	110	Read and write
`rwx`	7	111	Read, write, and execute

To determine the Umask value, simply determine what must be subtracted from 7 to reach the desired permission set. For example, to grant only read permissions for new files/folders (octal value 4 above), use a Umask value of 3, since 7 - 3 = 4.

A.2. Creating an S3 Umask Value for a Local Identity Provider

A Hammerspace S3 server that uses the local identity provider relies on a Umask setting to determine the permissions for new files and folders that are created using the S3 protocol.

The default setting is 0002, which can be explained as follows:

- The first value is 0, which sets the sticky bit to off. The sticky bit is a special file permission flag in Unix-like systems that, when set on a directory, prevents users (other than the owner or root) from deleting or moving files within that directory, even if they have write access.
- The second value sets the permissions for the file owner; a 0 (7 - 0 = octal permission value of 7) means that the owner will have full access to all files and folders that they create (read/write for files, read/write/execute for folders).

© 2026 - Hammerspace

- The third value sets the permissions for the groups that the file owner is a member of; a 0 (7 - 0 = octal permission value of 7) means that members of those groups will have full access to all files and folders the owner creates (read/write for files, read/write/execute for folders).
- The fourth value sets the permissions for all other users; a 2 (7 - 2 = octal permission value of 5, the Hammerspace default) means that all other users will have only read/execute permissions for folders and read permissions for files.



- A Umask of 0 (octal permission value of 7) grants read/write/execute permissions to folders and read/write permissions to files; this is expected.
- Hammerspace does not alter the existing permissions of any files or folders that were or are created by non-S3 clients.
- Umask is only used with S3 servers that use a local identity provider, not those that use Active Directory.

© 2026 - Hammerspace