



# **Managing S3 tenant accounts**

StorageGRID

NetApp

October 03, 2025

This PDF was generated from <https://docs.netapp.com/us-en/storagegrid-115/tenant/creating-your-own-s3-access-keys.html> on October 03, 2025. Always check [docs.netapp.com](https://docs.netapp.com) for the latest.

# Table of Contents

Managing S3 tenant accounts .....	1
Managing S3 access keys .....	1
Creating your own S3 access keys .....	1
Viewing your S3 access keys .....	3
Deleting your own S3 access keys .....	4
Creating another user's S3 access keys .....	5
Viewing another user's S3 access keys .....	8
Deleting another user's S3 access keys .....	10
Managing S3 buckets .....	11
Using S3 Object Lock .....	11
Creating an S3 bucket .....	15
Viewing S3 bucket details .....	18
Changing the consistency level .....	20
Enabling or disabling last access time updates .....	22
Configuring Cross-Origin Resource Sharing (CORS) .....	24
Deleting an S3 bucket .....	26

# Managing S3 tenant accounts

You can use the Tenant Manager to manage S3 access keys and to create and manage S3 buckets.

- [Managing S3 access keys](#)
- [Managing S3 buckets](#)

## Managing S3 access keys

Each user of an S3 tenant account must have an access key to store and retrieve objects in the StorageGRID system. An access key consists of an access key ID and a secret access key.

### About this task

S3 access keys can be managed as follows:

- Users who have the **Manage Your Own S3 Credentials** permission can create or remove their own S3 access keys.
- Users who have the **Root Access** permission can manage the access keys for the S3 root account and all other users. Root access keys provide full access to all buckets and objects for the tenant unless explicitly disabled by a bucket policy.

StorageGRID supports Signature Version 2 and Signature Version 4 authentication. Cross-account access is not permitted unless explicitly enabled by a bucket policy.

## Creating your own S3 access keys

If you are using an S3 tenant and you have the appropriate permission, you can create your own S3 access keys. You must have an access key to access your buckets and objects in the S3 tenant account.

### What you'll need

- You must be signed in to the Tenant Manager using a supported browser.
- You must have the **Manage Your Own S3 Credentials** permission.

### About this task

You can create one or more S3 access keys that allow you to create and manage buckets for your tenant account. After you create a new access key, update the application with your new access key ID and secret access key. For security, do not create more keys than you need, and delete the keys you are not using. If you have only one key and it is about to expire, create a new key before the old one expires, and then delete the old one.

Each key can have a specific expiration time or no expiration. Follow these guidelines for expiration time:

- Set an expiration time for your keys to limit your access to a certain time period. Setting a short expiration time can help reduce your risk if your access key ID and secret access key are accidentally exposed. Expired keys are removed automatically.
- If the security risk in your environment is low and you do not need to periodically create new keys, you do

not have to set an expiration time for your keys. If you decide later to create new keys, delete the old keys manually.

**i** The S3 buckets and objects belonging to your account can be accessed using the access key ID and secret access key displayed for your account in the Tenant Manager. For this reason, protect access keys as you would a password. Rotate access keys on a regular basis, remove any unused keys from your account, and never share them with other users.

## Steps

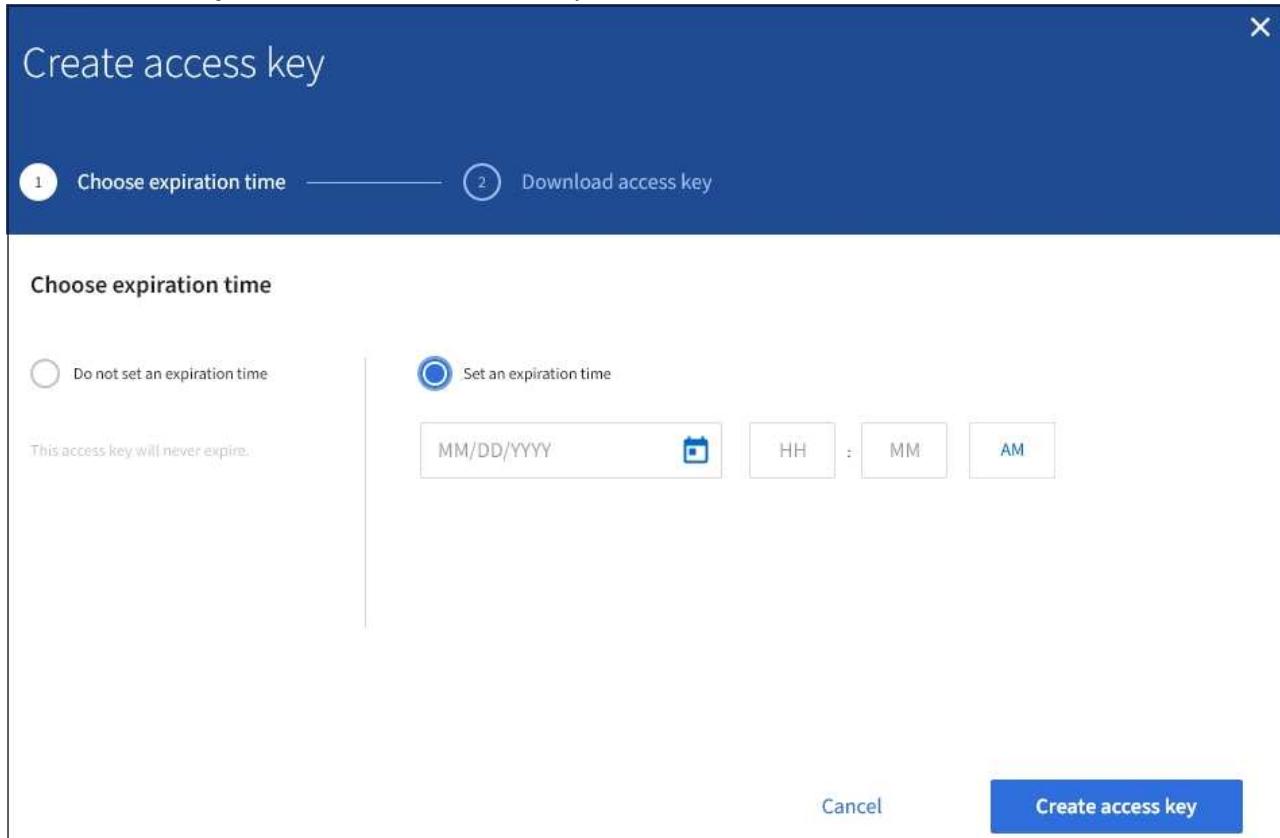
1. Select **STORAGE (S3) > My access keys**.

The My access keys page appears and lists any existing access keys.

2. Select **Create key**.

3. Do one of the following:

- Select **Do not set an expiration time** to create a key that will not expire. (Default)
- Select **Set an expiration time**, and set the expiration date and time.



4. Select **Create access key**.

The Download access key dialog box appears, listing your access key ID and secret access key.

5. Copy the access key ID and the secret access key to a safe location, or select **Download .csv** to save a spreadsheet file containing the access key ID and secret access key.



Do not close this dialog box until you have copied or downloaded this information.

Create access key X

1 Choose expiration time 2 Download access key

**Download access key**

To save the keys for future reference, select **Download .csv**, or copy and paste the values to another location.

**Info** You will not be able to view the Access key ID or Secret access key after you close this dialog.

**Access key ID**  
SHTBQKDRVHQ34YKKUAWX Copy

**Secret access key**  
UGu9+XeACtnOWQYFdbzmgmgVXXDvCkSOzT1OsZ9K Copy

Download .csv Finish

6. Select **Finish**.

The new key is listed on the My access keys page. Changes might take up to 15 minutes to take effect because of caching.

#### Related information

[Tenant management permissions](#)

## Viewing your S3 access keys

If you are using an S3 tenant and you have the appropriate permission, you can view a list of your S3 access keys. You can sort the list by expiration time, so you can determine which keys will expire soon. As needed, you can create new keys or delete keys that you are no longer using.

#### What you'll need

- You must be signed in to the Tenant Manager using a supported browser.
- You must have the [Manage Your Own S3 Credentials](#) permission.

**i** The S3 buckets and objects belonging to your account can be accessed using the access key ID and secret access key displayed for your account in the Tenant Manager. For this reason, protect access keys as you would a password. Rotate access keys on a regular basis, remove any unused keys from your account, and never share them with other users.

#### Steps

1. Select **STORAGE (S3) > My access keys**.

The My access keys page appears and lists any existing access keys.

My access keys

Manage your personal S3 access keys. If a key will expire soon, you can create a new key and delete the one it is replacing.

4 keys

Create key

Delete key

<input type="checkbox"/>	Access key ID	Expiration time
<input type="checkbox"/>	*****OTLS	2020-11-23 12:00:00 MST
<input type="checkbox"/>	*****0M45	2020-12-01 19:00:00 MST
<input type="checkbox"/>	*****69QJ	None
<input type="checkbox"/>	*****3R8P	None

2. Sort the keys by **Expiration time** or **Access key ID**.

3. As needed, create new keys and manually delete keys that you are no longer using.

If you create new keys before the existing keys expire, you can begin using the new keys without temporarily losing access to the objects in the account.

Expired keys are removed automatically.

#### Related information

[Creating your own S3 access keys](#)

[Deleting your own S3 access keys](#)

### Deleting your own S3 access keys

If you are using an S3 tenant and you have the appropriate permission, you can delete your own S3 access keys. After an access key is deleted, it can no longer be used to

access the objects and buckets in the tenant account.

## What you'll need

- You must be signed in to the Tenant Manager using a supported browser.
- You must have the Manage Your Own S3 Credentials permission.



The S3 buckets and objects belonging to your account can be accessed using the access key ID and secret access key displayed for your account in the Tenant Manager. For this reason, protect access keys as you would a password. Rotate access keys on a regular basis, remove any unused keys from your account, and never share them with other users.

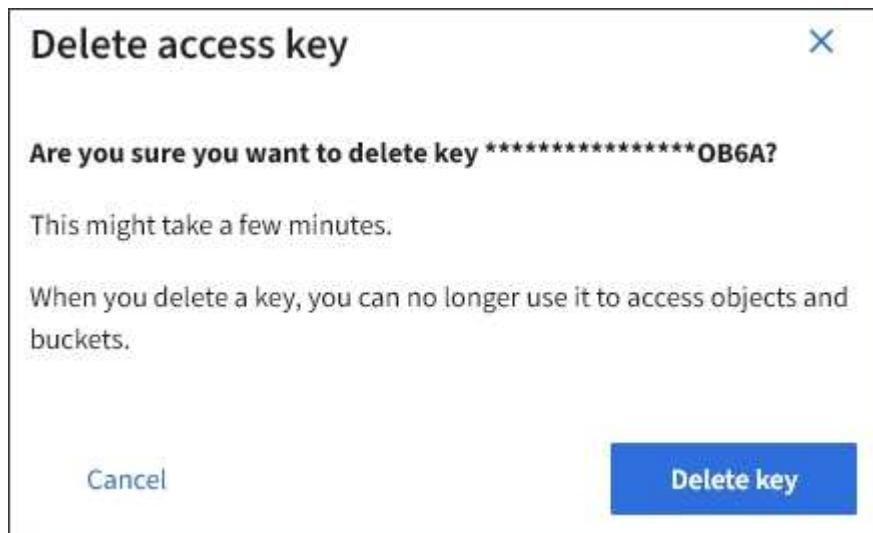
## Steps

1. Select **STORAGE (S3) > My access keys**.

The My access keys page appears and lists any existing access keys.

2. Select the check box for each access key you want to remove.
3. Select **Delete key**.

A confirmation dialog box appears.



4. Select **Delete key**.

A confirmation message appears in the upper right corner of the page. Changes might take up to 15 minutes to take effect because of caching.

## Related information

[Tenant management permissions](#)

## Creating another user's S3 access keys

If you are using an S3 tenant and you have the appropriate permission, you can create S3 access keys for other users, such as applications that need access to buckets and objects.

## What you'll need

- You must be signed in to the Tenant Manager using a supported browser.
- You must have the Root Access permission.

## About this task

You can create one or more S3 access keys for other users so they can create and manage buckets for their tenant account. After you create a new access key, update the application with the new access key ID and secret access key. For security, do not create more keys than the user needs, and delete the keys that are not being used. If you have only one key and it is about to expire, create a new key before the old one expires, and then delete the old one.

Each key can have a specific expiration time or no expiration. Follow these guidelines for expiration time:

- Set an expiration time for the keys to limit the user's access to a certain time period. Setting a short expiration time can help reduce risk if the access key ID and secret access key are accidentally exposed. Expired keys are removed automatically.
- If the security risk in your environment is low and you do not need to periodically create new keys, you do not have to set an expiration time for the keys. If you decide later to create new keys, delete the old keys manually.

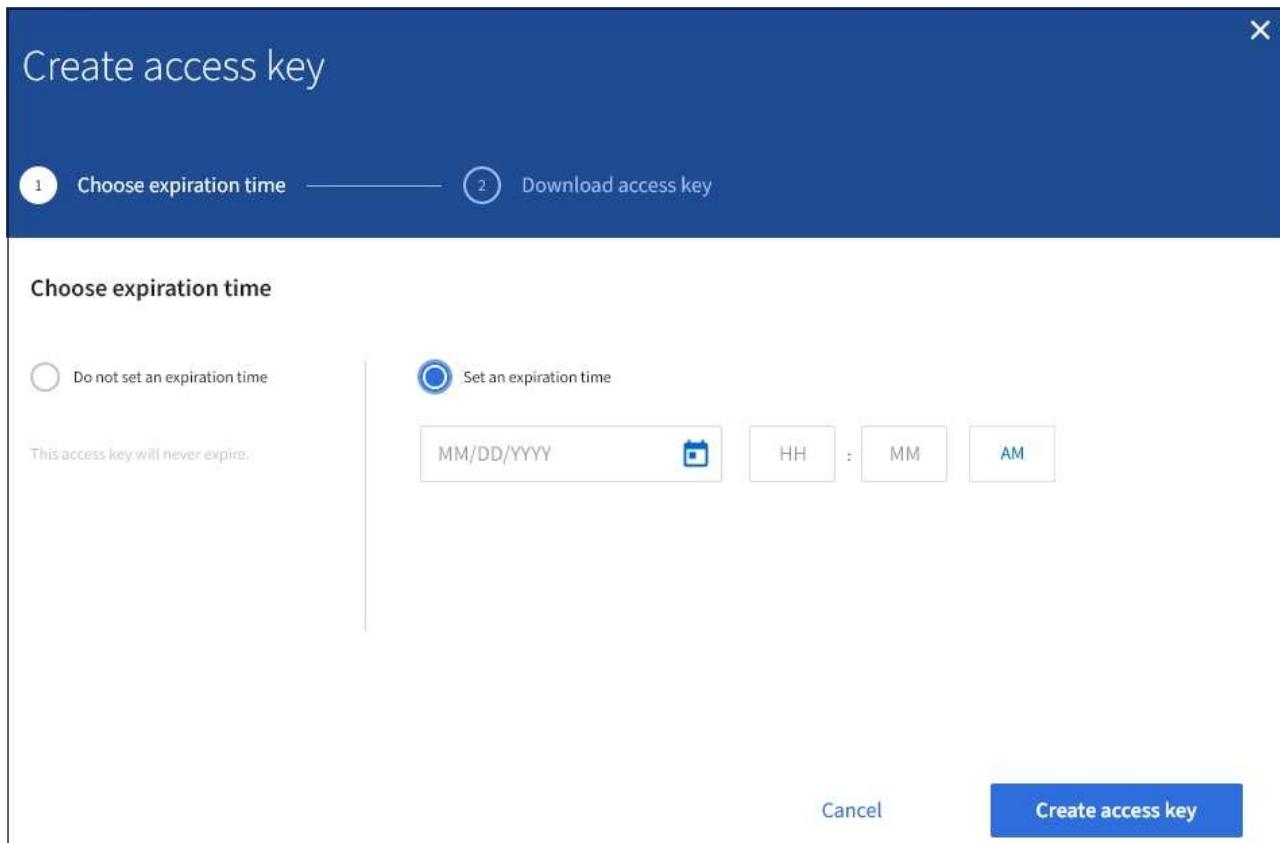
 The S3 buckets and objects belonging to a user can be accessed using the access key ID and secret access key displayed for that user in the Tenant Manager. For this reason, protect access keys as you would a password. Rotate access keys on a regular basis, remove any unused keys from the account, and never share them with other users.

## Steps

1. Select **ACCESS MANAGEMENT > Users**.
2. Select the user whose S3 access keys you want to manage.

The user detail page appears.

3. Select **Access keys**, then select **Create key**.
4. Do one of the following:
  - Select **Do not set an expiration time** to create a key that does not expire. (Default)
  - Select **Set an expiration time**, and set the expiration date and time.



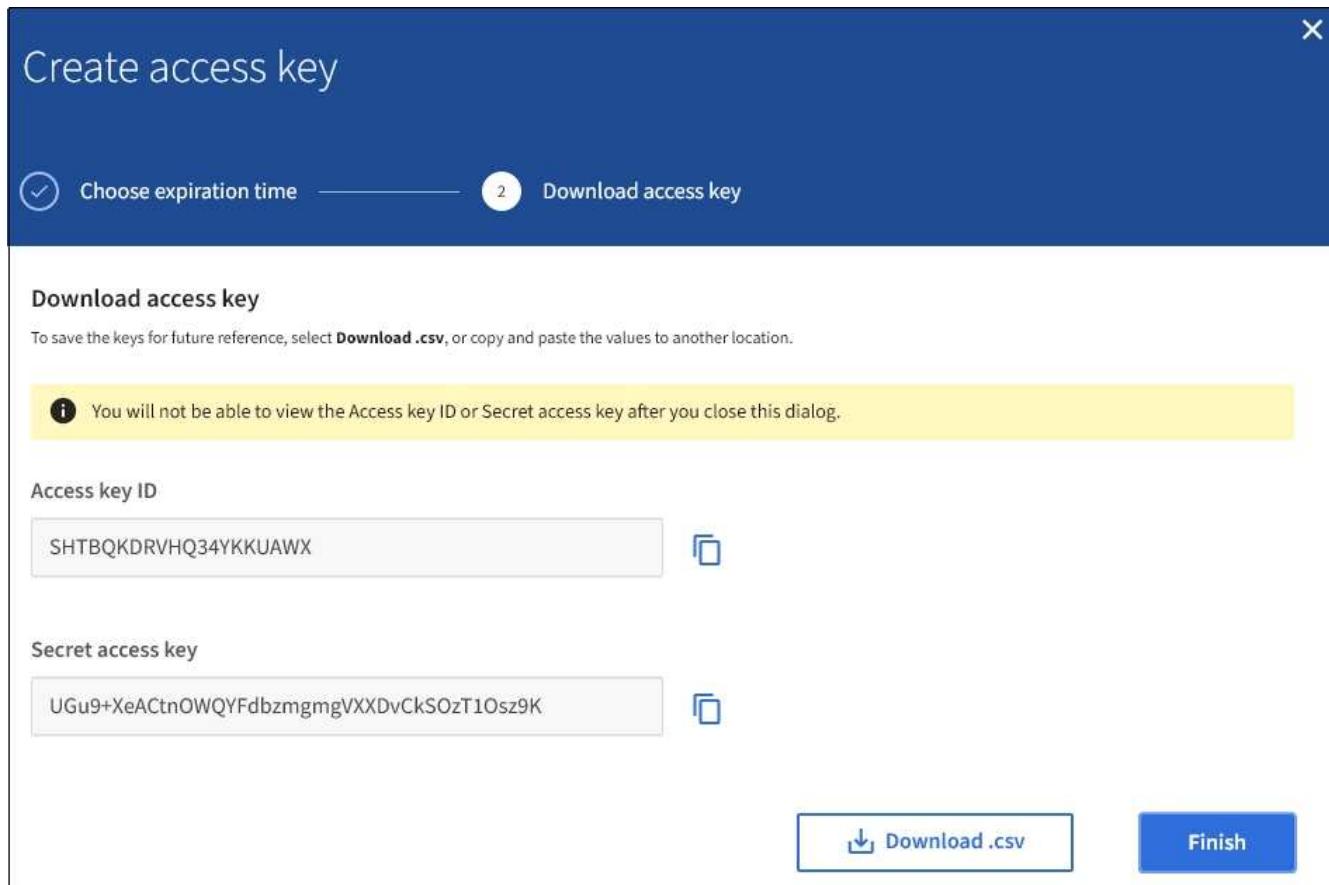
5. Select **Create access key**.

The Download access key dialog box appears, listing the access key ID and secret access key.

6. Copy the access key ID and the secret access key to a safe location, or select **Download .csv** to save a spreadsheet file containing the access key ID and secret access key.



Do not close this dialog box until you have copied or downloaded this information.



7. Select **Finish**.

The new key is listed on the Access keys tab of the user details page. Changes might take up to 15 minutes to take effect because of caching.

#### Related information

[Tenant management permissions](#)

## Viewing another user's S3 access keys

If you are using an S3 tenant and you have appropriate permissions, you can view another user's S3 access keys. You can sort the list by expiration time so you can determine which keys will expire soon. As needed, you can create new keys and delete keys that are no longer in use.

#### What you'll need

- You must be signed in to the Tenant Manager using a supported browser.
- You must have the Root Access permission.



The S3 buckets and objects belonging to a user can be accessed using the access key ID and secret access key displayed for that user in the Tenant Manager. For this reason, protect access keys as you would a password. Rotate access keys on a regular basis, remove any unused keys from the account, and never share them with other users.

#### Steps

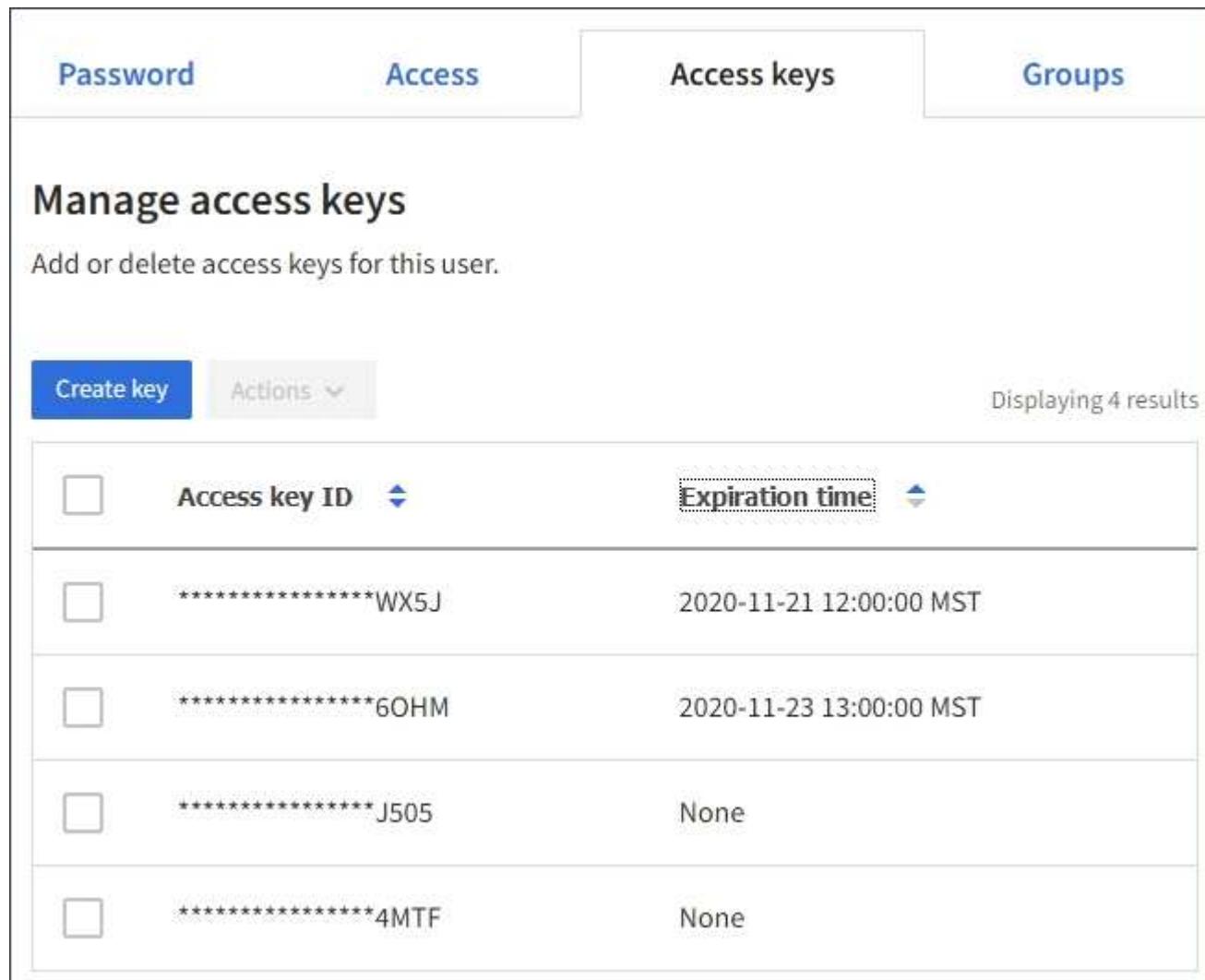
1. Select **ACCESS MANAGEMENT > Users**.

The Users page appears and lists the existing users.

2. Select the user whose S3 access keys you want to view.

The User details page appears.

3. Select **Access keys**.



The screenshot shows a user interface for managing access keys. At the top, there are tabs: 'Password', 'Access', 'Access keys' (which is the active tab), and 'Groups'. Below the tabs, the title 'Manage access keys' is displayed, followed by the sub-instruction 'Add or delete access keys for this user.' On the left, there are buttons for 'Create key' and 'Actions' with a dropdown arrow. On the right, it says 'Displaying 4 results'. The main area is a table with two columns: 'Access key ID' and 'Expiration time'. The table contains four rows, each with a checkbox and a key ID. The key IDs are partially obscured by asterisks. The expiration times are listed to the right of the key IDs.

Access key ID	Expiration time
*****WX5J	2020-11-21 12:00:00 MST
*****60HM	2020-11-23 13:00:00 MST
*****J505	None
*****4MTF	None

4. Sort the keys by **Expiration time** or **Access key ID**.

5. As needed, create new keys and manually delete keys that are no longer in use.

If you create new keys before the existing keys expire, the user can begin using the new keys without temporarily losing access to the objects in the account.

Expired keys are removed automatically.

#### Related information

[Creating another user's S3 access keys](#)

[Deleting another user's S3 access keys](#)

## Deleting another user's S3 access keys

If you are using an S3 tenant and you have appropriate permissions, you can delete another user's S3 access keys. After an access key is deleted, it can no longer be used to access the objects and buckets in the tenant account.

### What you'll need

- You must be signed in to the Tenant Manager using a supported browser.
- You must have the Root Access permission.



The S3 buckets and objects belonging to a user can be accessed using the access key ID and secret access key displayed for that user in the Tenant Manager. For this reason, protect access keys as you would a password. Rotate access keys on a regular basis, remove any unused keys from the account, and never share them with other users.

### Steps

1. Select **ACCESS MANAGEMENT > Users**.

The Users page appears and lists the existing users.

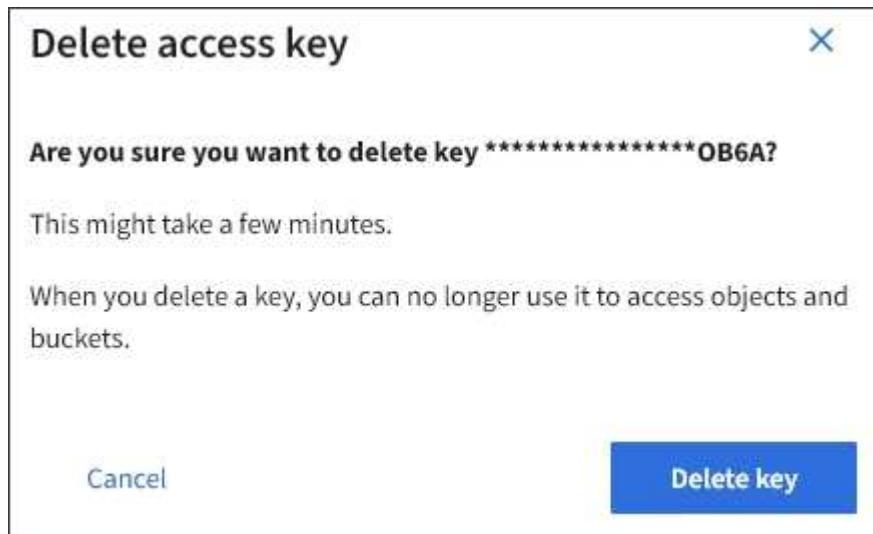
2. Select the user whose S3 access keys you want to manage.

The User details page appears.

3. Select **Access keys**, and then select the check box for each access key you want to delete.

4. Select **Actions > Delete selected key**.

A confirmation dialog box appears.



5. Select **Delete key**.

A confirmation message appears in the upper right corner of the page. Changes might take up to 15 minutes to take effect because of caching.

### Related information

## Managing S3 buckets

If you are using an S3 tenant with the appropriate permissions, you can create, view, and delete S3 buckets, update consistency level settings, configure Cross-Origin Resource Sharing (CORS), enable and disable last access time update settings, and manage S3 platform services.

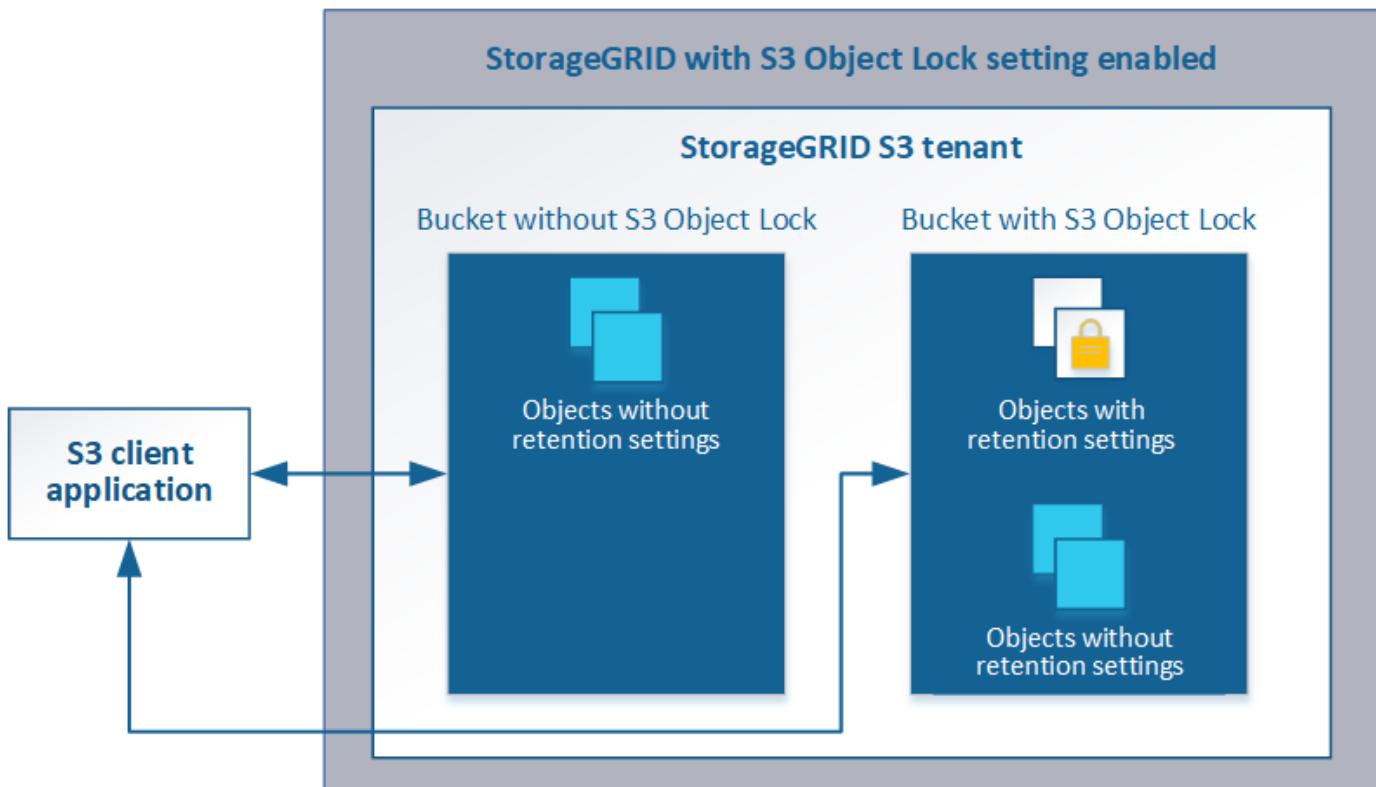
### Using S3 Object Lock

You can use the S3 Object Lock feature in StorageGRID if your objects must comply with regulatory requirements for retention.

#### What is S3 Object Lock?

The StorageGRID S3 Object Lock feature is an object-protection solution that is equivalent to S3 Object Lock in Amazon Simple Storage Service (Amazon S3).

As shown in the figure, when the global S3 Object Lock setting is enabled for a StorageGRID system, an S3 tenant account can create buckets with or without S3 Object Lock enabled. If a bucket has S3 Object Lock enabled, S3 client applications can optionally specify retention settings for any object version in that bucket. An object version must have retention settings specified to be protected by S3 Object Lock.



The StorageGRID S3 Object Lock feature provides a single retention mode that is equivalent to the Amazon S3 compliance mode. By default, a protected object version cannot be overwritten or deleted by any user. The StorageGRID S3 Object Lock feature does not support a governance mode, and it does not allow users with special permissions to bypass retention settings or to delete protected objects.

If a bucket has S3 Object Lock enabled, the S3 client application can optionally specify either or both of the following object-level retention settings when creating or updating an object:

- **Retain-until-date:** If an object version's retain-until-date is in the future, the object can be retrieved, but it cannot be modified or deleted. As required, an object's retain-until-date can be increased, but this date cannot be decreased.
- **Legal hold:** Applying a legal hold to an object version immediately locks that object. For example, you might need to put a legal hold on an object that is related to an investigation or legal dispute. A legal hold has no expiration date, but remains in place until it is explicitly removed. Legal holds are independent of the retain-until-date.

For details on these settings, go to “using S3 object lock” in [S3 REST API supported operations and limitations](#).

### Managing legacy Compliant buckets

The S3 Object Lock feature replaces the Compliance feature that was available in previous StorageGRID versions. If you created compliant buckets using a previous version of StorageGRID, you can continue to manage the settings of these buckets; however, you can no longer create new compliant buckets. For instructions, see the NetApp Knowledge Base article.

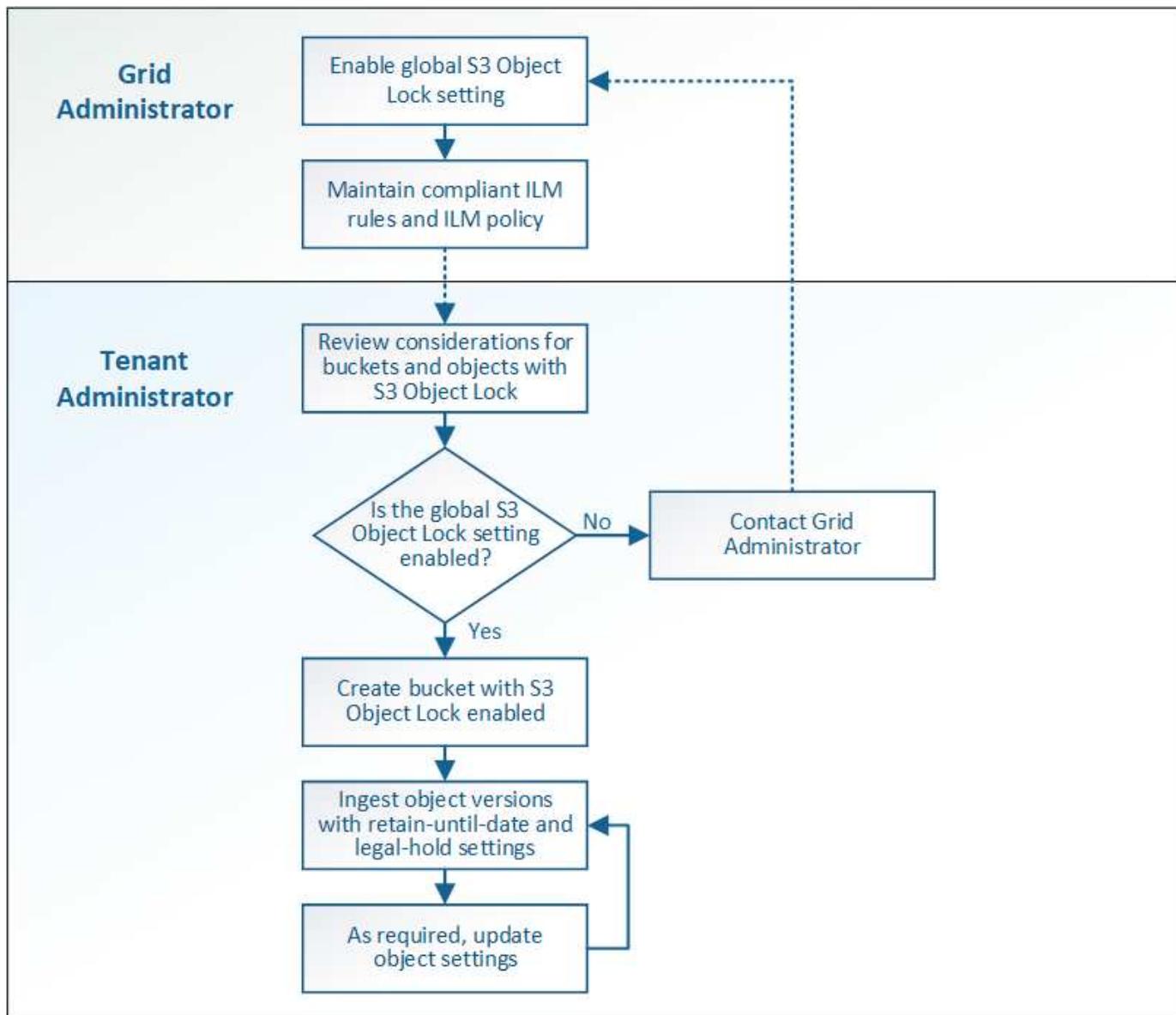
[NetApp Knowledge Base: How to manage legacy Compliant buckets in StorageGRID 11.5](#)

### S3 Object Lock workflow

The workflow diagram shows the high-level steps for using the S3 Object Lock feature in StorageGRID.

Before you can create buckets with S3 Object Lock enabled, the grid administrator must enable the global S3 Object Lock setting for the entire StorageGRID system. The grid administrator must also ensure that the information lifecycle management (ILM) policy is “compliant”; it must meet the requirements of buckets with S3 Object Lock enabled. For details, contact your grid administrator or see the instructions for managing objects with information lifecycle management.

After the global S3 Object Lock setting has been enabled, you can create buckets with S3 Object Lock enabled. You can then use the S3 client application to optionally specify retention settings for each object version.



## Related information

[Manage objects with ILM](#)

## Requirements for S3 Object Lock

Before enabling S3 Object Lock for a bucket, review the requirements for S3 Object Lock buckets and objects and the lifecycle of objects in buckets with S3 Object Lock enabled.

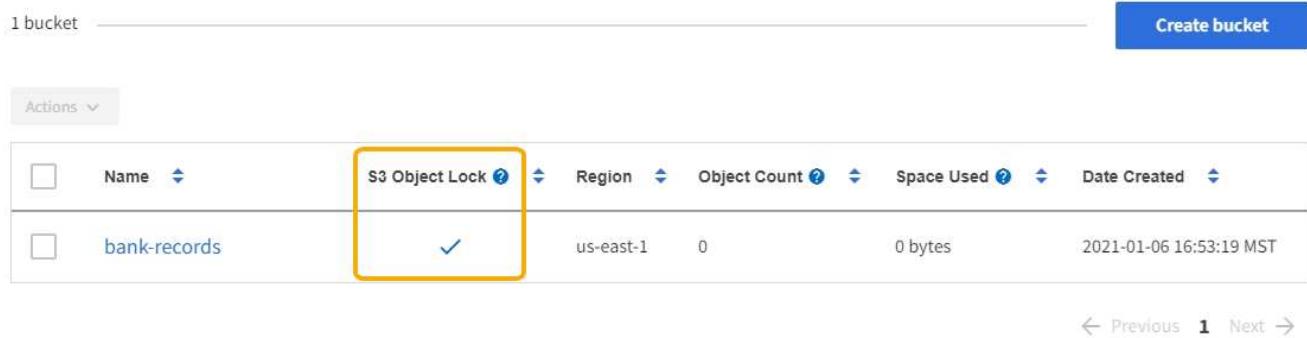
### Requirements for buckets with S3 Object Lock enabled

- If the global S3 Object Lock setting is enabled for the StorageGRID system, you can use the Tenant Manager, the Tenant Management API, or the S3 REST API to create buckets with S3 Object Lock enabled.

This example from the Tenant Manager shows a bucket with S3 Object Lock enabled.

# Buckets

Create buckets and manage bucket settings.



The screenshot shows the AWS S3 Buckets list page. At the top, it says "1 bucket" and has a "Create bucket" button. Below is a table with the following data:

Actions	Name	S3 Object Lock	Region	Object Count	Space Used	Date Created
<input type="checkbox"/>	bank-records	✓	us-east-1	0	0 bytes	2021-01-06 16:53:19 MST

At the bottom, there are navigation links: "Previous" and "Next".

- If you plan to use S3 Object Lock, you must enable S3 Object Lock when you create the bucket. You cannot enable S3 Object Lock for an existing bucket.
- Bucket versioning is required with S3 Object Lock. When S3 Object Lock is enabled for a bucket, StorageGRID automatically enables versioning for that bucket.
- After you create a bucket with S3 Object Lock enabled, you cannot disable S3 Object Lock or suspend versioning for that bucket.
- An StorageGRID bucket that has S3 Object Lock enabled does not have a default retention period. Instead, the S3 client application can optionally specify a retention date and legal hold setting for each object version that is added to that bucket.
- Bucket lifecycle configuration is supported for S3 Object Lifecycle buckets.
- CloudMirror replication is not supported for buckets with S3 Object Lock enabled.

## Requirements for objects in buckets with S3 Object Lock enabled

- The S3 client application must specify retention settings for each object that needs to be protected by S3 Object Lock.
- You can increase the retain-until-date for an object version, but you can never decrease this value.
- If you are notified of a pending legal action or regulatory investigation, you can preserve relevant information by placing a legal hold on an object version. When an object version is under a legal hold, that object cannot be deleted from StorageGRID, even if it has reached its retain-until-date. As soon as the legal hold is lifted, the object version can be deleted if the retain-until-date has been reached.
- S3 Object Lock requires the use of versioned buckets. Retention settings apply to individual object versions. An object version can have both a retain-until-date and a legal hold setting, one but not the other, or neither. Specifying a retain-until-date or a legal hold setting for an object protects only the version specified in the request. You can create new versions of the object, while the previous version of the object remains locked.

## Lifecycle of objects in buckets with S3 Object Lock enabled

Each object that is saved in a bucket with S3 Object Lock enabled goes through three stages:

### 1. Object ingest

- When adding an object version to a bucket with S3 Object Lock enabled, the S3 client application can optionally specify retention settings for the object (retain-until-date, legal hold, or both). StorageGRID then generates metadata for that object, which includes a unique object identifier (UUID) and the ingest

date and time.

- After an object version with retention settings is ingested, its data and S3 user-defined metadata cannot be modified.
- StorageGRID stores the object metadata independently of the object data. It maintains three copies of all object metadata at each site.

## 2. Object retention

- Multiple copies of the object are stored by StorageGRID. The exact number and type of copies and the storage locations are determined by the compliant rules in the active ILM policy.

## 3. Object deletion

- An object can be deleted when its retain-until-date is reached.
- An object that is under a legal hold cannot be deleted.

# Creating an S3 bucket

You can use the Tenant Manager to create S3 buckets for object data. When you create a bucket, you must specify the bucket's name and region. If the global S3 Object Lock setting is enabled for the StorageGRID system, you can optionally enable S3 Object Lock for the bucket.

## What you'll need

- You must be signed in to the Tenant Manager using a supported browser.
- You must belong to a user group that has the Manage All Buckets or the Root Access permission. These permissions override the permissions settings in group or bucket policies.
- If you plan to create a bucket with S3 Object Lock, the global S3 Object Lock setting must have been enabled for the StorageGRID system and you must have reviewed the requirements for S3 Object Lock buckets and objects.

## Using S3 Object Lock

## Steps

### 1. Select **STORAGE (S3) > Buckets**.

The Buckets page appears and lists any buckets that have already been created.

# Buckets

Create buckets and manage bucket settings.

0 buckets

[Create bucket](#)

Actions ▾

Name	S3 Object Lock	Region	Object Count	Space Used	Date Created
No buckets found					
<a href="#">Create bucket</a>					

## 2. Select **Create bucket**.

The Create bucket wizard appears.

### Create bucket

**Enter bucket details**  
Enter the bucket's name and select the bucket's region.

Bucket name [?](#)

Region [?](#)  
 ▾

[Cancel](#) [Create bucket](#)



If the global S3 Object Lock setting is enabled, Create bucket includes a second step for managing S3 Object Lock for the bucket.

## 3. Enter a unique name for the bucket.



You cannot change the bucket name after creating the bucket.

Bucket names must comply with these rules:

- Must be unique across each StorageGRID system (not just unique within the tenant account).
- Must be DNS compliant.
- Must contain at least 3 and no more than 63 characters.
- Can be a series of one or more labels, with adjacent labels separated by a period. Each label must start and end with a lowercase letter or a number and can only use lowercase letters, numbers, and hyphens.
- Must not look like a text-formatted IP address.
- Should not use periods in virtual hosted style requests. Periods will cause problems with server wildcard certificate verification.



See the Amazon Web Services (AWS) Documentation for more information.

#### 4. Select the region for this bucket.

Your StorageGRID administrator manages the available regions. A bucket's region can affect the data-protection policy applied to objects. By default, all buckets are created in the `us-east-1` region.



You cannot change the region after creating the bucket.

#### 5. Select **Create bucket** or **Continue**.

- If the global S3 Object Lock setting is not enabled, select **Create bucket**. The bucket is created and added to the table on the Buckets page.
- If the global S3 Object Lock setting is enabled, select **Continue**. Step 2, Manage S3 Object Lock, appears.

**Create bucket**

1 **Enter details** ————— 2 **Manage S3 Object Lock** Optional

**Manage S3 Object Lock** (This step is optional)

S3 Object Lock allows you to specify retention and legal hold settings for the objects ingested into a bucket. If you want to use S3 Object Lock, you must enable this setting when you create the bucket. You cannot add or disable S3 Object Lock after a bucket is created.

If S3 Object Lock is enabled, bucket versioning is required and will be enabled automatically.

**Enable S3 Object Lock**

**Previous** **Create bucket**

#### 6. Optionally, select the check box to enable S3 Object Lock for this bucket.

S3 Object Lock must be enabled for the bucket before an S3 client application can specify retain-until-date and legal hold settings for the objects added to the bucket.



You cannot enable or disable S3 Object Lock after creating the bucket.



If you enable S3 Object Lock for a bucket, bucket versioning is enabled automatically.

## 7. Select **Create bucket**.

The bucket is created and added to the table on the Buckets page.

### Related information

[Manage objects with ILM](#)

[Understanding the Tenant Management API](#)

[Use S3](#)

## Viewing S3 bucket details

You can view a list of the buckets and bucket settings in your tenant account.

### What you'll need

- You must be signed in to the Tenant Manager using a supported browser.

### Steps

#### 1. Select **STORAGE (S3) > Buckets**.

The Buckets page appears and lists all buckets for the tenant account.

Actions	Name	S3 Object Lock	Region	Object Count	Space Used	Date Created
<input type="checkbox"/>	bucket-01	✓	us-east-1	0	0 bytes	2020-11-04 14:16:59 MST
<input type="checkbox"/>	bucket-02		us-east-1	0	0 bytes	2020-11-04 14:17:14 MST

#### 2. Review the information for each bucket.

As required, you can sort the information by any column, or you can page forward and back through the list.

- Name: The bucket's unique name, which cannot be changed.
- S3 Object Lock: Whether S3 Object Lock is enabled for this bucket.

This column is not displayed if the global S3 Object lock setting is disabled. This column also shows information for any legacy Compliant buckets.

- Region: The bucket's region, which cannot be changed.
- Object Count: The number of objects in this bucket.
- Space Used: The logical size of all objects in this bucket. The logical size does not include the actual space required for replicated or erasure-coded copies or for object metadata.
- Date Created: The date and time the bucket was created.



The Object Count and Space Used values displayed are estimates. These estimates are affected by the timing of ingests, network connectivity, and node status.

3. To view and manage the settings for a bucket, select the bucket name.

The bucket details page appears.

This page allows you to view and edit the settings for bucket options, bucket access, and platform services.

See the instructions for configuring each setting or platform service.

The screenshot shows the AWS Bucket Details page for 'bucket-02'. The top navigation bar shows 'Buckets > bucket-02'. The main section is titled 'Overview' with the following details:

Name:	bucket-02
Region:	us-east-1
S3 Object Lock:	Disabled
Date created:	2020-11-04 14:51:59 MST

Below the overview, there are three tabs: 'Bucket options' (selected), 'Bucket access', and 'Platform services'. Under 'Bucket options', there are two settings:

Consistency level	Read-after-new-write
Last access time updates	Disabled

## Related information

[Changing the consistency level](#)

[Enabling or disabling last access time updates](#)

[Configuring Cross-Origin Resource Sharing \(CORS\)](#)

[Configuring CloudMirror replication](#)

[Configuring event notifications](#)

[Configuring the search integration service](#)

## Changing the consistency level

If you are using an S3 tenant, you can use the Tenant Manager or the Tenant Management API to change the consistency control for operations performed on the objects in S3 buckets.

### What you'll need

- You must be signed in to the Tenant Manager using a supported browser.
- You must belong to a user group that has the Manage All Buckets or the Root Access permission. These permissions override the permissions settings in group or bucket policies.

### About this task

Consistency level makes a trade-off between the availability of the objects and the consistency of those objects across different Storage Nodes and sites. In general, you should use the **Read-after-new-write** consistency level for your buckets. If the **Read-after-new-write** consistency level does not meet the client application's requirements, you can change the consistency level by setting the bucket consistency level or by using the **Consistency-Control** header. The **Consistency-Control** header overrides the bucket consistency level.



When you change a bucket's consistency level, only those objects that are ingested after the change are guaranteed to meet the revised level.

### Steps

1. Select **STORAGE (S3) > Buckets**.
2. Select the bucket name from the list.

The bucket details page appears.

3. Select **Bucket options > Consistency level**.

Bucket options
Bucket access
Platform services

Consistency level
Read-after-new-write (default)
^

Change the consistency control for operations performed on the objects in the bucket. Consistency level makes a trade-off between the availability of the objects and the consistency of those objects across different Storage Nodes and sites.

In general, use the **Read-after-new-write** consistency level for your buckets. Then, if objects do not meet availability or consistency requirements, change the client application's behavior, or set the `Consistency-Control` header for an individual API request, which overrides the bucket setting.

- All  
Provides the highest guarantee of consistency. All nodes receive the data immediately, or the request will fail.
- Strong-global  
Guarantees read-after-write consistency for all client requests across all sites.
- Strong-site  
Guarantees read-after-write consistency for all client requests within a site.
- Read-after-new-write (default)  
Provides read-after-write consistency for new objects and eventual consistency for object updates. Offers high availability, and data protection guarantees.  
Note: If your application attempts HEAD operations on keys that do not exist, set the Consistency Level to **Available**, unless you require AWS S3 consistency guarantees. Otherwise, a high number of 500 Internal Server errors can result if one or more Storage Nodes are unavailable.
- Available  
Behaves the same as the **Read-after-new-write** consistency level, but only provides eventual consistency for HEAD operations. Offers higher availability for HEAD operations than **Read-after-new-write** if Storage Nodes are unavailable. Differs from AWS S3 consistency guarantees for HEAD operations only.

Save changes

4. Select a consistency level for operations performed on the objects in this bucket.

Consistency level	Description
All	All nodes receive the data immediately, or the request will fail.
Strong-global	Guarantees read-after-write consistency for all client requests across all sites.

Consistency level	Description
Strong-site	Guarantees read-after-write consistency for all client requests within a site.
Read-after-new-write (Default)	<p>Provides read-after-write consistency for new objects and eventual consistency for object updates. Offers high availability, and data protection guarantees. Matches Amazon S3 consistency guarantees.</p> <p><b>Note:</b> If your application attempts HEAD operations on keys that do not exist, set the Consistency Level to <b>Available</b>, unless you require Amazon S3 consistency guarantees. Otherwise, a high number of 500 Internal Server errors can result if one or more Storage Nodes are unavailable.</p>
Available (eventual consistency for HEAD operations)	Behaves the same as the <b>Read-after-new-write</b> consistency level, but only provides eventual consistency for HEAD operations. Offers higher availability for HEAD operations than <b>Read-after-new-write</b> if Storage Nodes are unavailable. Differs from Amazon S3 consistency guarantees for HEAD operations only.

## 5. Select **Save changes**.

### Related information

[Tenant management permissions](#)

## Enabling or disabling last access time updates

When grid administrators create the information lifecycle management (ILM) rules for a StorageGRID system, they can optionally specify that an object's last access time be used to determine whether to move that object to a different storage location. If you are using an S3 tenant, you can take advantage of such rules by enabling last access time updates for the objects in an S3 bucket.

These instructions only apply to StorageGRID systems that include at least one ILM rule that uses the **Last Access Time** option in its placement instructions. You can ignore these instructions if your StorageGRID system does not include such a rule.

### What you'll need

- You must be signed in to the Tenant Manager using a supported browser.
- You must belong to a user group that has the Manage All Buckets or the Root Access permission. These permissions override the permissions settings in group or bucket policies.

**Last Access Time** is one of the options available for the **Reference Time** placement instruction for an ILM rule. Setting the Reference Time for a rule to Last Access Time lets grid administrators specify that objects be placed in certain storage locations based on when those objects were last retrieved (read or viewed).

For example, to ensure that recently viewed objects remain on faster storage, a grid administrator can create an ILM rule specifying the following:

- Objects that have been retrieved in the past month should remain on local Storage Nodes.
- Objects that have not been retrieved in the past month should be moved to an off-site location.



See the instructions for managing objects with information lifecycle management.

By default, updates to last access time are disabled. If your StorageGRID system includes an ILM rule that uses the **Last Access Time** option and you want this option to apply to objects in this bucket, you must enable updates to last access time for the S3 buckets specified in that rule.



Updating the last access time when an object is retrieved can reduce StorageGRID performance, especially for small objects.

A performance impact occurs with last access time updates because StorageGRID must perform these additional steps every time objects are retrieved:

- Update the objects with new timestamps
- Add the objects to the ILM queue, so they can be reevaluated against current ILM rules and policy

The table summarizes the behavior applied to all objects in the bucket when last access time is disabled or enabled.

Type of request	Behavior if last access time is disabled (default)		Behavior if last access time is enabled	
	Last access time updated?	Object added to ILM evaluation queue?	Last access time updated?	Object added to ILM evaluation queue?
Request to retrieve an object, its access control list, or its metadata	No	No	Yes	Yes
Request to update an object's metadata	Yes	Yes	Yes	Yes
Request to copy an object from one bucket to another	<ul style="list-style-type: none"><li>• No, for the source copy</li><li>• Yes, for the destination copy</li></ul>	<ul style="list-style-type: none"><li>• No, for the source copy</li><li>• Yes, for the destination copy</li></ul>	<ul style="list-style-type: none"><li>• Yes, for the source copy</li><li>• Yes, for the destination copy</li></ul>	<ul style="list-style-type: none"><li>• Yes, for the source copy</li><li>• Yes, for the destination copy</li></ul>
Request to complete a multipart upload	Yes, for the assembled object	Yes, for the assembled object	Yes, for the assembled object	Yes, for the assembled object

## Steps

1. Select **STORAGE (S3) > Buckets**.
2. Select the bucket name from the list.

The bucket details page appears.

3. Select **Bucket options > Last access time updates**.
4. Select the appropriate radio button to enable or disable last access time updates.

Bucket options Bucket access Platform services

Consistency level Read-after-new-write

Last access time updates Disabled

Enable or disable last access time updates for the objects in this bucket.

When last access time updates are disabled, the following behavior applies to objects in the bucket:

- Requests to retrieve an object, its access control list, or its metadata do not update the object's last access time. The object is not added to ILM evaluation queues.
- Requests to update an object's metadata update the object's last access time. The object is added to ILM evaluation queues.
- Requests to copy an object from one bucket to another do not update the last access time for the source copy and do not add the source object to the ILM evaluation queue. However, the last access time is updated for the destination copy, and the destination object is added to ILM evaluation queues.
- A request to complete a multipart upload causes the last access time for the assembled object to be updated. The new object is added to ILM evaluation queues.

**i** Updating the last access time when an object is retrieved can reduce performance, especially for small objects.

Enable last access time updates when retrieving an object

Disable last access time updates when retrieving an object

Save changes

5. Select **Save changes**.

#### Related information

[Tenant management permissions](#)

[Manage objects with ILM](#)

## Configuring Cross-Origin Resource Sharing (CORS)

You can configure Cross-Origin Resource Sharing (CORS) for an S3 bucket if you want that bucket and objects in that bucket to be accessible to web applications in other domains.

#### What you'll need

- You must be signed in to the Tenant Manager using a supported browser.
- You must belong to a user group that has the Manage All Buckets or the Root Access permission. These permissions override the permissions settings in group or bucket policies.

#### About this task

Cross-Origin Resource Sharing (CORS) is a security mechanism that allows client web applications in one domain to access resources in a different domain. For example, suppose you use an S3 bucket named Images to store graphics. By configuring CORS for the Images bucket, you can allow the images in that bucket to be displayed on the website <http://www.example.com>.

#### Steps

1. Use a text editor to create the XML required to enable CORS.

This example shows the XML used to enable CORS for an S3 bucket. This XML allows any domain to send GET requests to the bucket, but it only allows the `http://www.example.com` domain to send POST and DELETE requests. All request headers are allowed.

```
<CORSConfiguration
  xmlns="http://s3.amazonaws.com/doc/2020-10-22/">
  <CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
  </CORSRule>
  <CORSRule>
    <AllowedOrigin>http://www.example.com</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
  </CORSRule>
</CORSConfiguration>
```

For more information about the CORS configuration XML, see [Amazon Web Services \(AWS\) Documentation: Amazon Simple Storage Service Developer Guide](#).

2. In the Tenant Manager, select **STORAGE (S3) > Buckets**.

3. Select the bucket name from the list.

The bucket details page appears.

4. Select **Bucket access > Cross-Origin Resource Sharing (CORS)**.
5. Select the **Enable CORS** check box.
6. Paste the CORS configuration XML into the text box, and select **Save changes**.

The screenshot shows the 'Bucket access' tab selected in the AWS S3 Bucket settings. Under the 'Cross-Origin Resource Sharing (CORS)' section, the status is 'Disabled'. A description explains that CORS allows bucket and objects to be accessible to web applications in other domains. A checked checkbox labeled 'Enable CORS' is present, and a 'Clear' button is available. A text input box contains the following XML configuration:

```
<corsConfiguration xmlns="http://s3.amazonaws.com/doc/2020-10-22/">
  <corsRule>
    <allowedOrigin>*</allowedOrigin>
    <allowedMethod>GET</allowedMethod>
    <allowedHeader>*</allowedHeader>
  </corsRule>
  <corsRule>
    <allowedOrigin>http://www.example.com</allowedOrigin>
    <allowedMethod>GET</allowedMethod>
    <allowedMethod>POST</allowedMethod>
    <allowedMethod>DELETE</allowedMethod>
  </corsRule>
</corsConfiguration>
```

A 'Save changes' button is located at the bottom right of the text input area.

7. To modify the CORS setting for the bucket, update the CORS configuration XML in the text box or select **Clear** to start over. Then select **Save changes**.
8. To disable CORS for the bucket, unselect the **Enable CORS** check box, and then select **Save changes**.

## Deleting an S3 bucket

You can use the Tenant Manager to delete an S3 bucket that is empty.

### What you'll need

- You must be signed in to the Tenant Manager using a supported browser.
- You must belong to a user group that has the Manage All Buckets or the Root Access permission. These permissions override the permissions settings in group or bucket policies.

### About this task

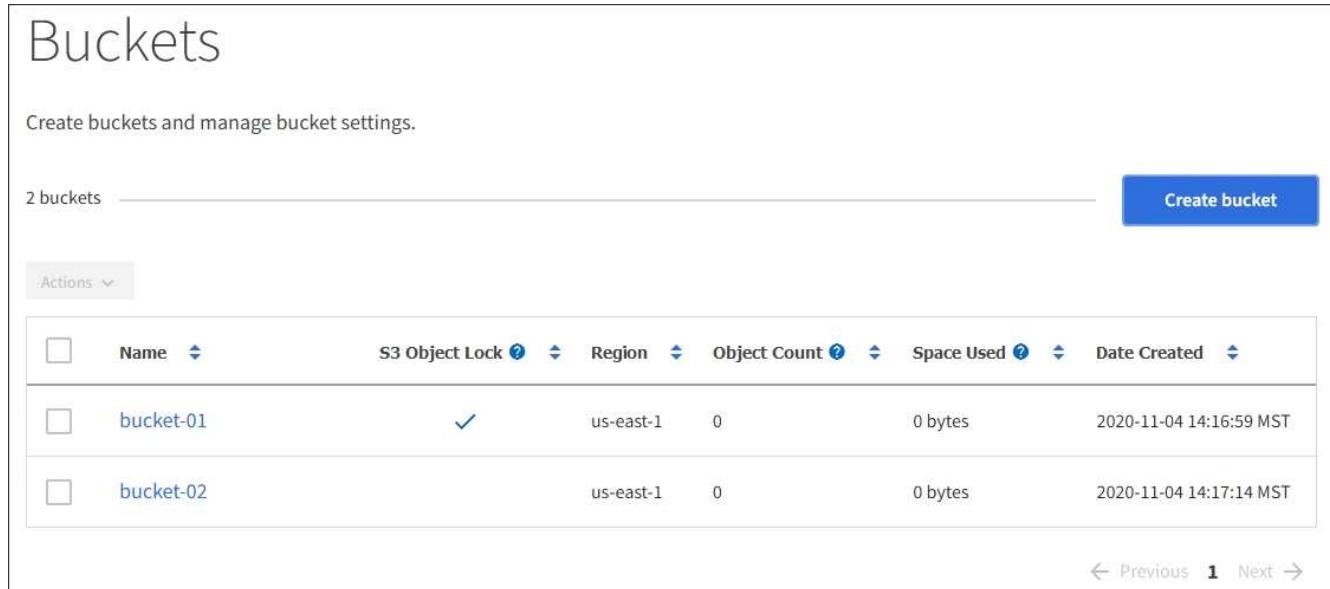
These instructions describe how to delete an S3 bucket using the Tenant Manager. You can also delete S3 buckets using the Tenant Management API or the S3 REST API.

You cannot delete an S3 bucket if it contains objects or noncurrent object versions. For information about how S3 versioned objects are deleted, see the instructions for managing objects with information lifecycle management.

## Steps

1. Select **STORAGE (S3) > Buckets**.

The Buckets page appears and shows all existing S3 buckets.



Buckets

Create buckets and manage bucket settings.

2 buckets

Create bucket

Actions

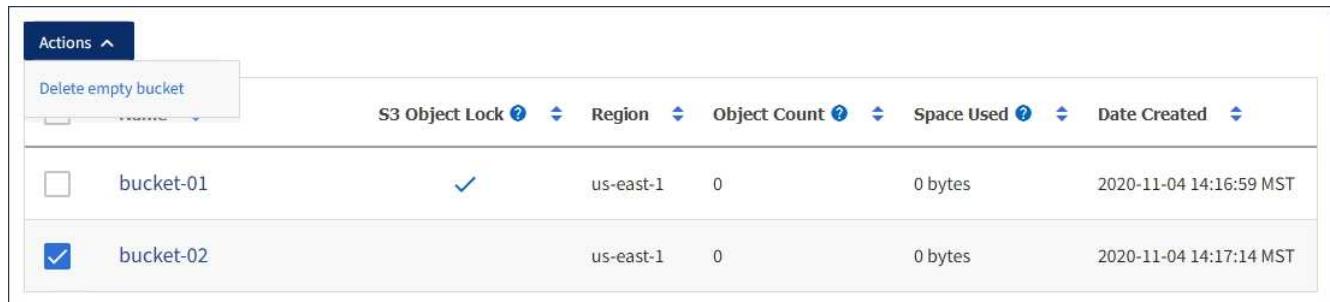
Name	S3 Object Lock	Region	Object Count	Space Used	Date Created
bucket-01	✓	us-east-1	0	0 bytes	2020-11-04 14:16:59 MST
bucket-02		us-east-1	0	0 bytes	2020-11-04 14:17:14 MST

← Previous 1 Next →

2. Select the check box for the empty bucket you want to delete.

The Actions menu is enabled.

3. From the Actions menu, select **Delete empty bucket**.

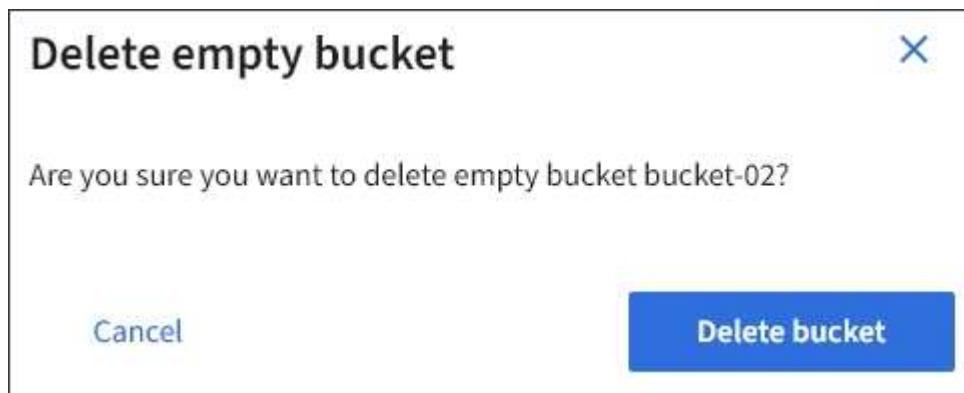


Actions

Delete empty bucket

Name	S3 Object Lock	Region	Object Count	Space Used	Date Created
bucket-01	✓	us-east-1	0	0 bytes	2020-11-04 14:16:59 MST
bucket-02	✓	us-east-1	0	0 bytes	2020-11-04 14:17:14 MST

A confirmation message appears.



Delete empty bucket

Are you sure you want to delete empty bucket bucket-02?

Cancel

Delete bucket

4. If you are sure you want to delete the bucket, select **Delete bucket**.

StorageGRID confirms that the bucket is empty and then deletes the bucket. This operation might take a few minutes.

If the bucket is not empty, an error message appears. You must delete all objects before you can delete the bucket.



Unable to delete the bucket because it is not empty. You must delete all objects before you can delete this bucket.

## Related information

[Manage objects with ILM](#)

## Copyright information

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

**LIMITED RIGHTS LEGEND:** Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

## Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.