# Manage objects with ILM

StorageGRID 11.7

NetApp
April 12, 2024

# Table of Contents

# Manage objects with ILM

## Manage objects with ILM: Overview

You manage the objects in a StorageGRID system by configuring an information lifecycle management (ILM) policy that consists of one or more ILM rules. The ILM rules instruct StorageGRID how to create and distribute copies of object data and how to manage those copies over time.

### About these instructions

Designing and implementing ILM rules and the ILM policy requires careful planning. You must understand your operational requirements, the topology of your StorageGRID system, your object protection needs, and the available storage types. Then, you must determine how you want different types of objects to be copied, distributed, and stored.

Use these instructions to:

- Learn about StorageGRID ILM, including how ILM operates throughout an object’s life.
- Learn how to configure storage pools, Cloud Storage Pools, and ILM rules.
- Learn how to create, simulate, and activate an ILM policy that will protect object data across one or more sites.
- Learn how to manage objects with S3 Object Lock, which helps to ensure that objects in specific S3 buckets aren’t deleted or overwritten for a specified amount of time.

### Learn more

To learn more, review these videos:

- Video: Information lifecycle management rules in StorageGRID 11.7.

- Video: Information lifecycle management policies in StorageGRID 11.7

Information lifecycle management policies in StorageGRID 11.7
Getting Started
NetApp

# ILM and object lifecycle

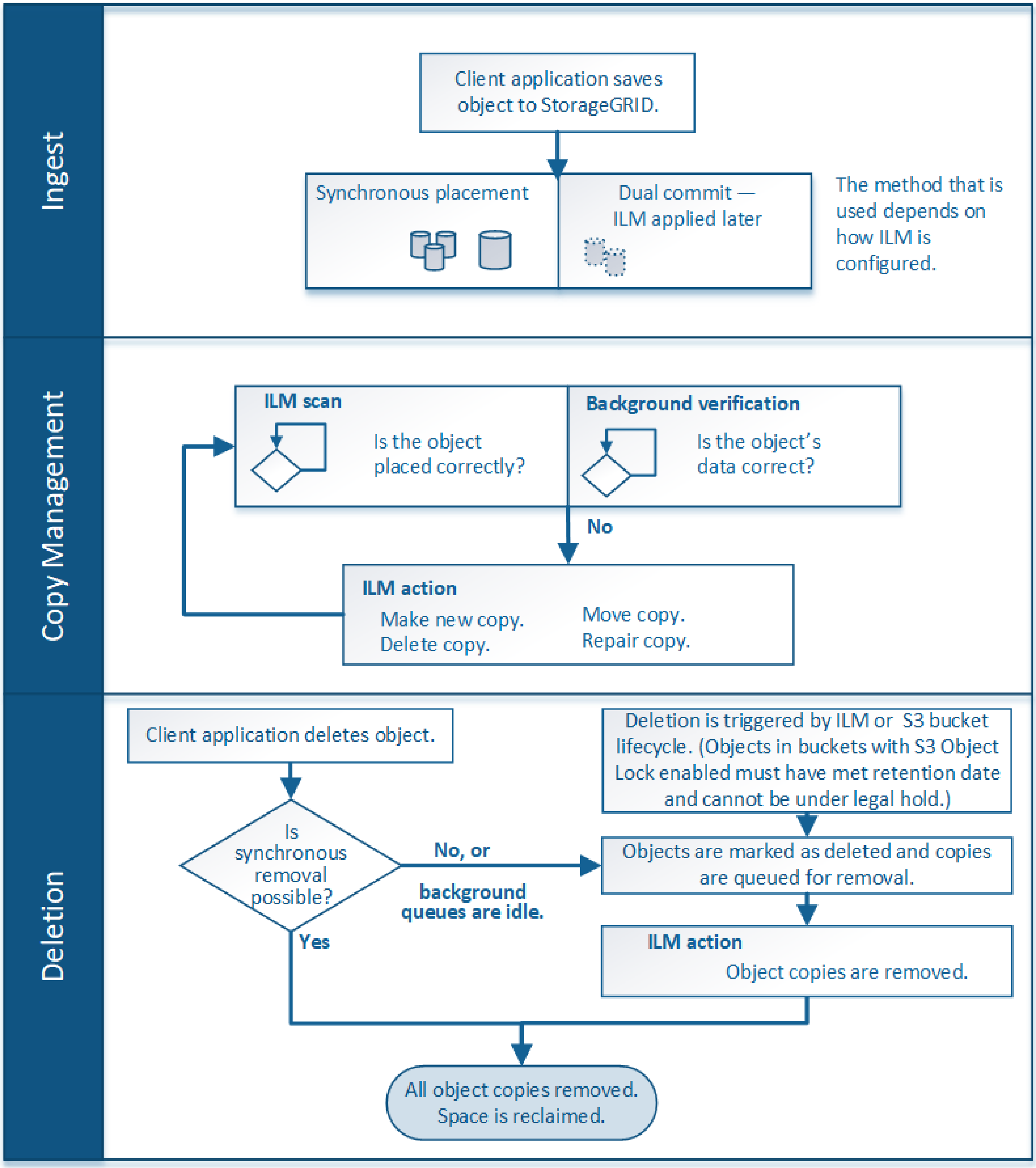
## How ILM operates throughout an object’s life

Understanding how StorageGRID uses ILM to manage objects during every stage of their life can help you design a more effective policy.

- **Ingest**: Ingest begins when an S3 or Swift client application establishes a connection to save an object to the StorageGRID system, and is complete when StorageGRID returns an “ingest successful” message to the client. Object data is protected during ingest either by applying ILM instructions immediately (synchronous placement) or by creating interim copies and applying ILM later (dual commit), depending on how the ILM requirements were specified.
- **Copy management**: After creating the number and type of object copies that are specified in the ILM’s placement instructions, StorageGRID manages object locations and protects objects against loss.
  - ILM scanning and evaluation: StorageGRID continuously scans the list of objects stored in the grid and checks if the current copies meet ILM requirements. When different types, numbers, or locations of object copies are required, StorageGRID creates, deletes, or moves copies as needed.
  - Background verification: StorageGRID continuously performs background verification to check the integrity of object data. If a problem is found, StorageGRID automatically creates a new object copy or a replacement erasure-coded object fragment in a location that meets current ILM requirements. See Verify object integrity.
- **Object deletion**: Management of an object ends when all copies are removed from the StorageGRID system. Objects can be removed as a result of a delete request by a client, or as a result of deletion by ILM or deletion caused by the expiration of an S3 bucket lifecycle.

> ⓘ Objects in a bucket that has S3 Object Lock enabled can’t be deleted if they are under a legal hold or if a retain-until-date has been specified but not yet met.

The diagram summarizes how ILM operates throughout an object’s lifecycle.


Ingest
Client application saves object to StorageGRID.
Synchronous placement
Dual commit — ILM applied later
The method that is used depends on how ILM is configured.
Copy Management
ILM scan
Is the object placed correctly?
Background verification
Is the object's data correct?
No
ILM action
Make new copy.
Delete copy.
Move copy.
Repair copy.
Deletion
Client application deletes object.
Deletion is triggered by ILM or S3 bucket lifecycle. (Objects in buckets with S3 Object Lock enabled must have met retention date and cannot be under legal hold.)
Is synchronous removal possible?
No, or
background queues are idle.
Objects are marked as deleted and copies are queued for removal.
ILM action
Object copies are removed.
Yes
All object copies removed. Space is reclaimed.

## How objects are ingested

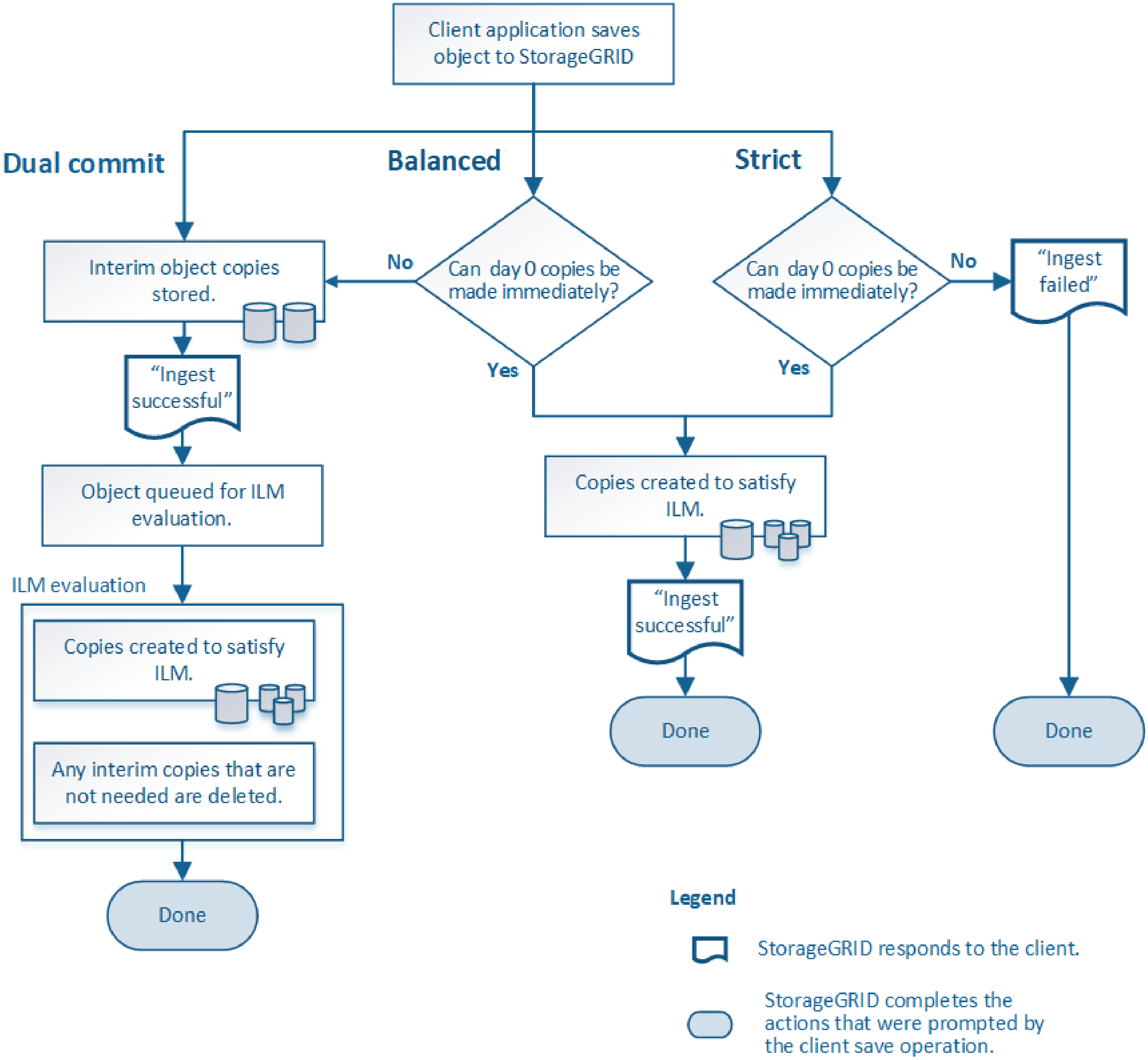
### Ingest options

When you create an ILM rule, you specify one of three options for protecting objects at ingest: Dual commit, Strict, or Balanced.

Depending on your choice, StorageGRID makes interim copies and queues the objects for ILM evaluation later, or it uses synchronous placement and immediately makes copies to meet ILM requirements.

#### Flowchart of ingest options

The flowchart shows what happens when objects are matched by an ILM rule that uses each of the three ingest options.



Client application saves object to StorageGRID
Dual commit
Balanced
Strict
Interim object copies stored.
No
Can day 0 copies be made immediately?
Can day 0 copies be made immediately?
No
"Ingest failed"
"Ingest successful"
Yes
Yes
Object queued for ILM evaluation.
Copies created to satisfy ILM.
ILM evaluation
"Ingest successful"
Copies created to satisfy ILM.
Any interim copies that are not needed are deleted.
Done
Done
Done
Legend
StorageGRID responds to the client.
StorageGRID completes the actions that were prompted by the client save operation.

#### Dual commit

When you select the Dual commit option, StorageGRID immediately makes interim object copies on two different Storage Nodes and returns an "ingest successful" message to the client. The object is queued for ILM evaluation, and copies that meet the rule's placement instructions are made later.

### When to use the Dual commit option

Use the Dual commit option in either of these cases:

- You are using multi-site ILM rules and client ingest latency is your primary consideration. When using Dual commit, you must ensure your grid can perform the additional work of creating and removing the dual-commit copies if they don't satisfy ILM. Specifically:
  - The load on the grid must be low enough to prevent an ILM backlog.
  - The grid must have excess hardware resources (IOPS, CPU, memory, network bandwidth, and so on).
- You are using multi-site ILM rules and the WAN connection between the sites usually has high latency or limited bandwidth. In this scenario, using the Dual commit option can help prevent client timeouts. Before choosing the Dual commit option, you should test the client application with realistic workloads.

#### Strict

When you select the Strict option, StorageGRID uses synchronous placement on ingest and immediately makes all object copies specified in the rule's placement instructions. Ingest fails if StorageGRID can't create all copies, for example, because a required storage location is temporarily unavailable. The client must retry the operation.

### When to use the Strict option

Use the Strict option if you have an operational or regulatory requirement to immediately store objects only in the locations outlined in the ILM rule. For example, to satisfy a regulatory requirement, you might need to use the Strict option and a Location Constraint advanced filter to guarantee that objects are never stored at certain data centers.

See Example 5: ILM rules and policy for Strict ingest behavior.

#### Balanced (default)

When you select the Balanced option, StorageGRID also uses synchronous placement on ingest and immediately makes all copies specified in the rule's placement instructions. In contrast with the Strict option, if StorageGRID can't immediately make all copies, it uses Dual commit instead.

### When to use the Balanced option

Use the Balanced option to achieve the best combination of data protection, grid performance, and ingest success. Balanced is the default option in the Create ILM rule wizard.

### Advantages, disadvantages, and limitations of the ingest options

Understanding the advantages and disadvantages of each of the three options for protecting data at ingest (Balanced, Strict, or Dual commit) can help you decide which one to select for an ILM rule.

For an overview of ingest options, see Ingest options.

#### Advantages of the Balanced and Strict options

When compared to Dual commit, which creates interim copies during ingest, the two synchronous placement options can provide the following advantages:

- **Better data security**: Object data is immediately protected as specified in the ILM rule's placement instructions, which can be configured to protect against a wide variety of failure conditions, including the failure of more than one storage location. Dual commit can only protect against the loss of a single local copy.

- **More efficient grid operation**: Each object is processed only once, as it is ingested. Because the StorageGRID system does not need to track or delete interim copies, there is less processing load and less database space is consumed.
- **(Balanced) Recommended**: The Balanced option provides optimal ILM efficiency. Using the Balanced option is recommended unless Strict ingest behavior is required or the grid meets all of the criteria for using Dual commit.
- **(Strict) Certainty about object locations**: The Strict option guarantees that objects are immediately stored according to the placement instructions in the ILM rule.

#### Disadvantages of the Balanced and Strict options

When compared to Dual commit, the Balanced and Strict options have some disadvantages:

- **Longer client ingests**: Client ingest latencies might be longer. When you use the Balanced or Strict options, an “ingest successful” message is not returned to the client until all erasure-coded fragments or replicated copies are created and stored. However, object data will most likely reach its final placement much faster.
- **(Strict) Higher rates of ingest failure**: With the Strict option, ingest fails whenever StorageGRID can’t immediately make all copies specified in the ILM rule. You might see high rates of ingest failure if a required storage location is temporarily offline or if network issues cause delays in copying objects between sites.
- **(Strict) S3 multipart upload placements might not be as expected in some circumstances**: With Strict, you expect objects either to be placed as described by the ILM rule or for ingest to fail. However, with an S3 multipart upload, ILM is evaluated for each part of the object as it is ingested, and for the object as a whole when the multipart upload completes. In the following circumstances this might result in placements that are different than you expect:
  - **If ILM changes while an S3 multipart upload is in progress**: Because each part is placed according to the rule that is active when the part is ingested, some parts of the object might not meet current ILM requirements when the multipart upload completes. In these cases, ingest of the object does not fail. Instead, any part that is not placed correctly is queued for ILM re-evaluation and is moved to the correct location later.
  - **When ILM rules filter on size**: When evaluating ILM for a part, StorageGRID filters on the size of the part, not the size of the object. This means that parts of an object can be stored in locations that don’t meet ILM requirements for the object as a whole. For example, if a rule specifies that all objects 10 GB or larger are stored at DC1 while all smaller objects are stored at DC2, at ingest each 1 GB part of a 10-part multipart upload is stored at DC2. When ILM is evaluated for the object, all parts of the object are moved to DC1.
- **(Strict) Ingest does not fail when object tags or metadata are updated and newly required placements cannot be made**: With Strict, you expect objects either to be placed as described by the ILM rule or for ingest to fail. However, when you update metadata or tags for an object that is already stored in the grid, the object is not re-ingested. This means that any changes to object placement that are triggered by the update aren’t made immediately. Placement changes are made when ILM is re-evaluated by normal background ILM processes. If required placement changes can’t be made (for example, because a newly required location is unavailable), the updated object retains its current placement until the placement changes are possible.

#### Limitations on object placements with the Balanced and Strict options

The Balanced or Strict options can’t be used for ILM rules that have any of these placement instructions:

- Placement in a Cloud Storage Pool at day 0.

- Placement in an Archive Node at day 0.
- Placements in a Cloud Storage Pool or an Archive Node when the rule has a User defined creation time as its Reference time.

These restrictions exist because StorageGRID can’t synchronously make copies to a Cloud Storage Pool or an Archive Node, and a User defined creation time could resolve to the present.

#### How ILM rules and consistency controls interact to affect data protection

Both your ILM rule and your choice of consistency control affect how objects are protected. These settings can interact.

For example, the ingest behavior selected for an ILM rule affects the initial placement of object copies, while the consistency control used when an object is stored affects the initial placement of object metadata. Because StorageGRID requires access to both an object’s data and metadata to fulfill client requests, selecting matching levels of protection for the consistency level and ingest behavior can provide better initial data protection and more predictable system responses.

Here is a brief summary of the consistency controls that are available in StorageGRID:

- **all**: All nodes receive object metadata immediately or the request will fail.
- **strong-global**: Object metadata is immediately distributed to all sites. Guarantees read-after-write consistency for all client requests across all sites.
- **strong-site**: Object metadata is immediately distributed to other nodes at the site. Guarantees read-after-write consistency for all client requests within a site.
- **read-after-new-write**: Provides read-after-write consistency for new objects and eventual consistency for object updates. Offers high availability and data protection guarantees. Recommended for most cases.
- **available**: Provides eventual consistency for both new objects and object updates. For S3 buckets, use only as required (for example, for a bucket that contains log values that are rarely read, or for HEAD or GET operations on keys that don’t exist). Not supported for S3 FabricPool buckets.

ⓘ Before selecting a consistency level, read the full description of consistency controls in the instructions for Use S3 REST API. You should understand the benefits and limitations before changing the default value.

#### Example of how the consistency control and ILM rule can interact

Suppose you have a two-site grid with the following ILM rule and the following consistency level setting:

- **ILM rule**: Create two object copies, one at the local site and one at a remote site. The Strict ingest behavior is selected.
- **Consistency level**: “strong-global” (Object metadata is immediately distributed to all sites.)

When a client stores an object to the grid, StorageGRID makes both object copies and distributes metadata to both sites before returning success to the client.

The object is fully protected against loss at the time of the ingest successful message. For example, if the local site is lost shortly after ingest, copies of both the object data and the object metadata still exist at the remote site. The object is fully retrievable.

If you instead used the same ILM rule and the “strong-site” consistency level, the client might receive a success message after object data is replicated to the remote site but before object metadata is distributed there. In this case, the level of protection of object metadata does not match the level of protection for object data. If the local site is lost shortly after ingest, object metadata is lost. The object can’t be retrieved.

The inter-relationship between consistency levels and ILM rules can be complex. Contact NetApp if you require assistance.

**Related information**

- Example 5: ILM rules and policy for Strict ingest behavior

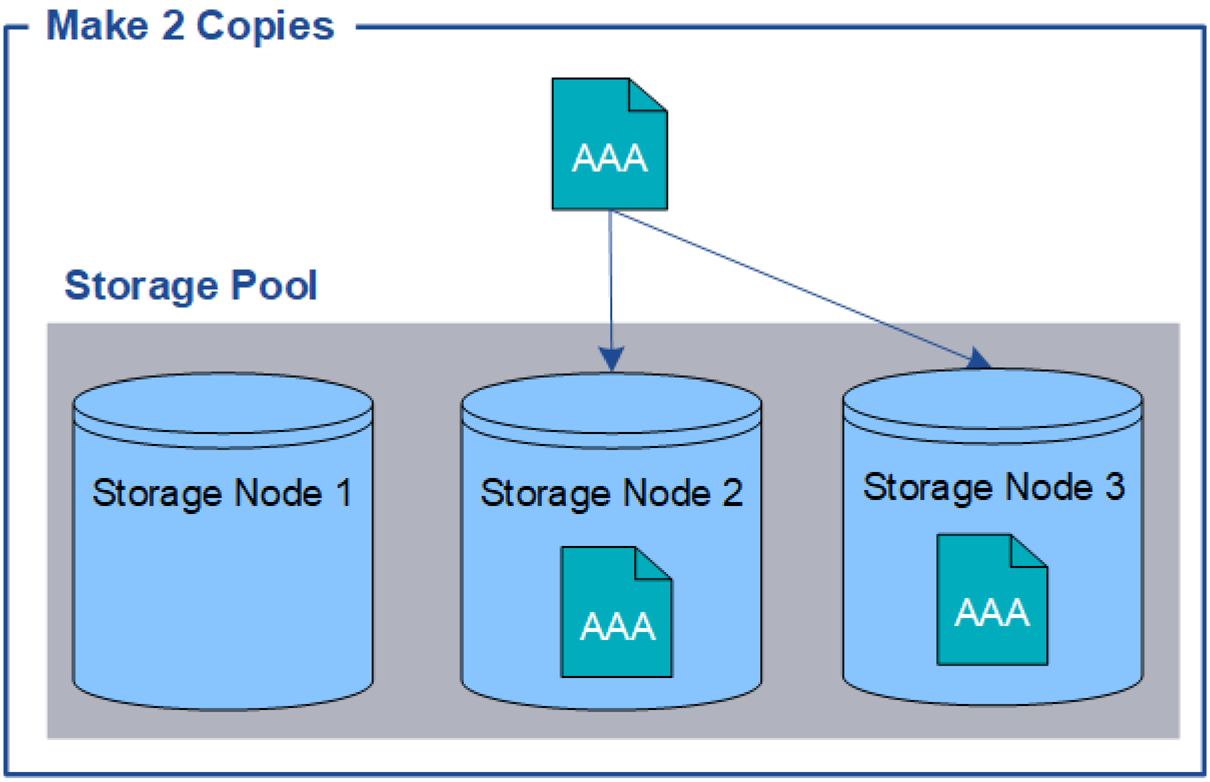
## How objects are stored (replication or erasure coding)

### What is replication?

Replication is one of two methods used by StorageGRID to store object data. When objects match an ILM rule that uses replication, the system creates exact copies of object data and stores the copies on Storage Nodes or Archive Nodes.

When you configure an ILM rule to create replicated copies, you specify how many copies should be created, where those copies should be placed, and how long the copies should be stored at each location.

In the following example, the ILM rule specifies that two replicated copies of each object be placed in a storage pool that contains three Storage Nodes.



When StorageGRID matches objects to this rule, it creates two copies of the object, placing each copy on a different Storage Node in the storage pool. The two copies might be placed on any two of the three available Storage Nodes. In this case, the rule placed object copies on Storage Nodes 2 and 3. Because there are two copies, the object can be retrieved if any of the nodes in the storage pool fails.

StorageGRID can store only one replicated copy of an object on any given Storage Node. If your grid includes three Storage Nodes and you create a 4-copy ILM rule, only three copies will be made—one copy for each Storage Node. The **ILM placement unachievable** alert is triggered to indicate that the ILM rule could not be completely applied.

**Related information**

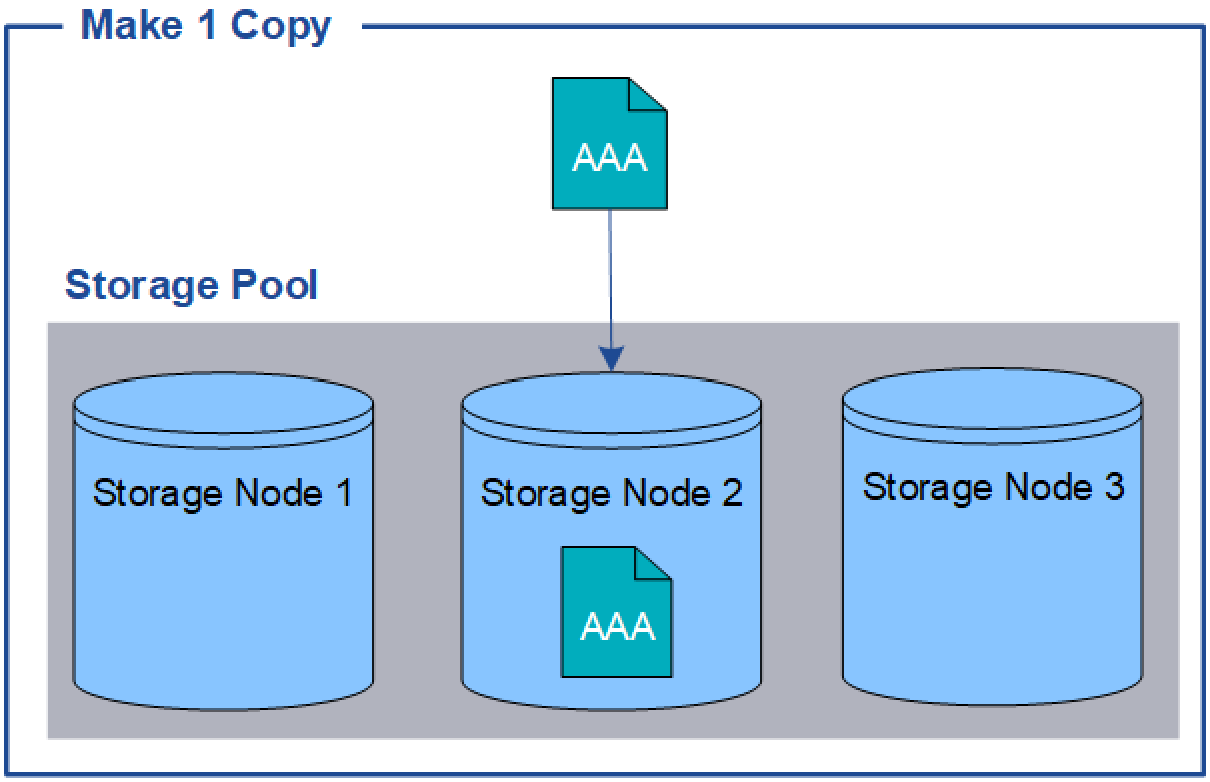
- What is erasure coding?
- What is a storage pool?
- Enable site-loss protection using replication and erasure coding

### Why you should not use single-copy replication

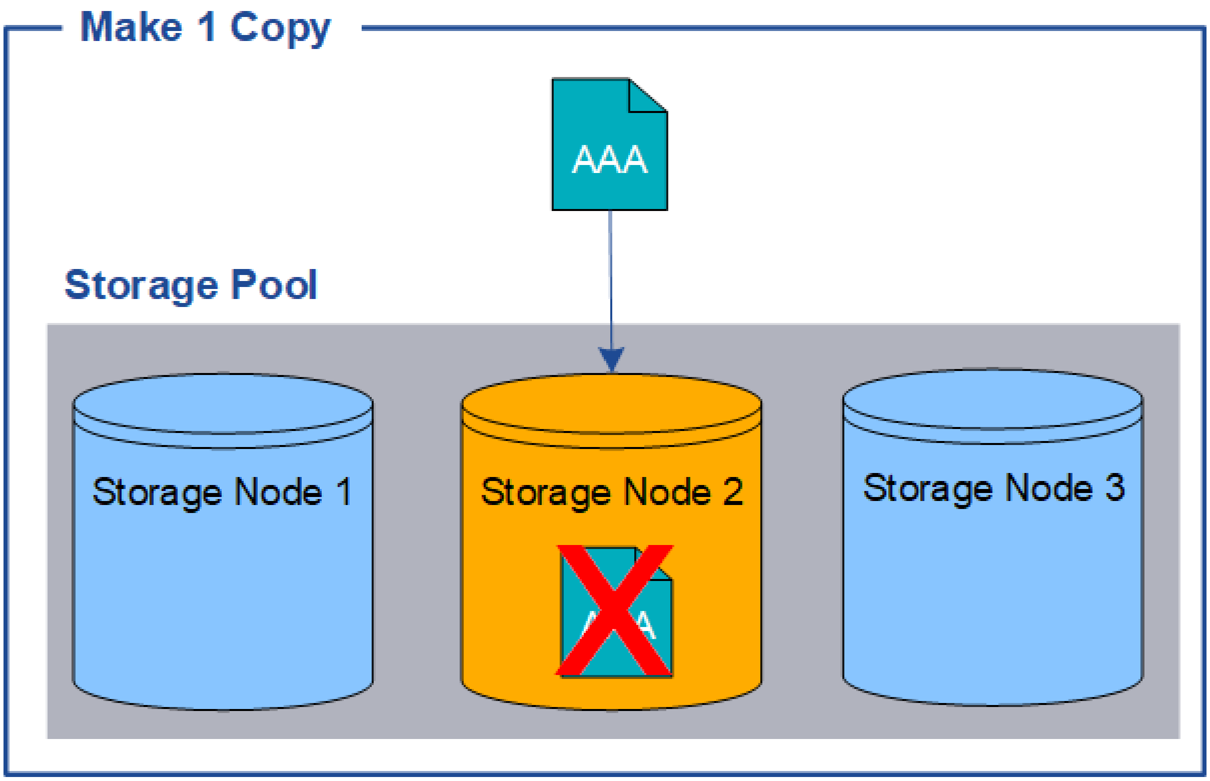
When creating an ILM rule to create replicated copies, you should always specify at least two copies for any time period in the placement instructions.

Don’t use an ILM rule that creates only one replicated copy for any time period. If only one replicated copy of an object exists, that object is lost if a Storage Node fails or has a significant error. You also temporarily lose access to the object during maintenance procedures such as upgrades.

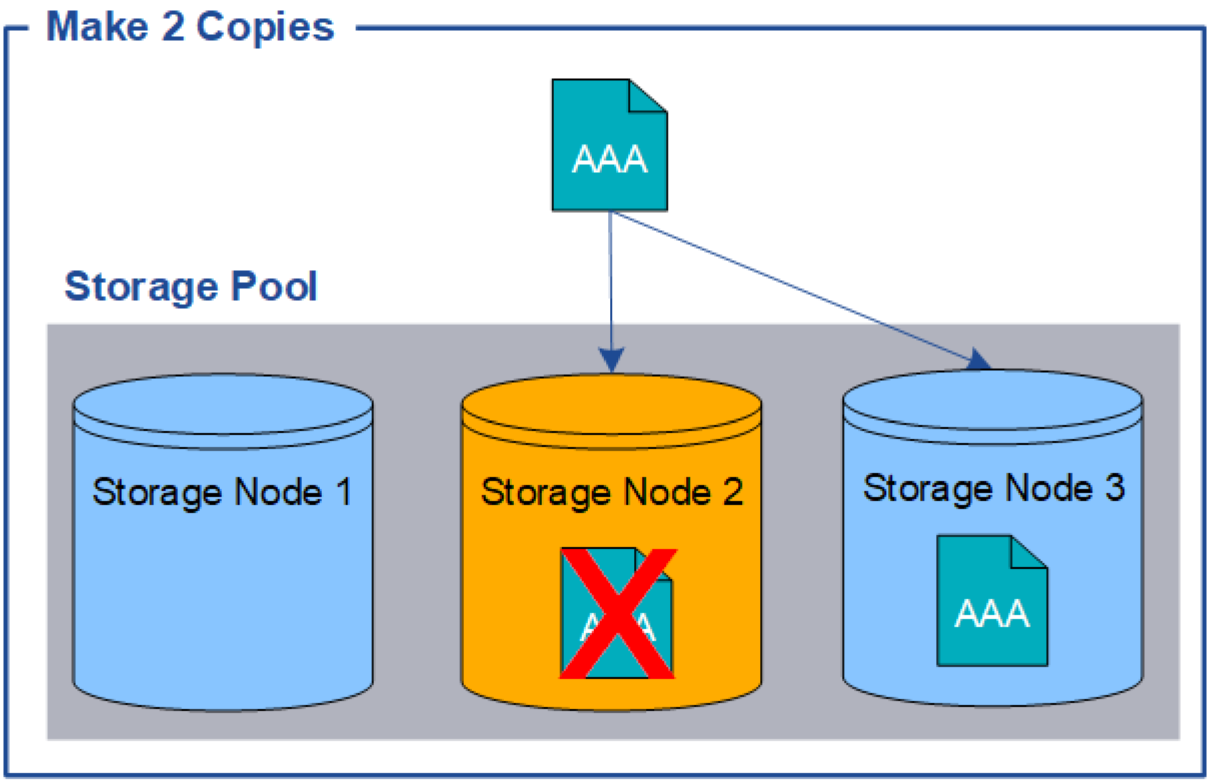
In the following example, the Make 1 Copy ILM rule specifies that one replicated copy of an object be placed in a storage pool that contains three Storage Nodes. When an object is ingested that matches this rule, StorageGRID places a single copy on only one Storage Node.



When an ILM rule creates only one replicated copy of an object, the object becomes inaccessible when the Storage Node is unavailable. In this example, you will temporarily lose access to object AAA whenever Storage Node 2 is offline, such as during an upgrade or other maintenance procedure. You will lose object AAA entirely if Storage Node 2 fails.



To avoid losing object data, you should always make at least two copies of all objects you want to protect with replication. If two or more copies exist, you can still access the object if one Storage Node fails or goes offline.



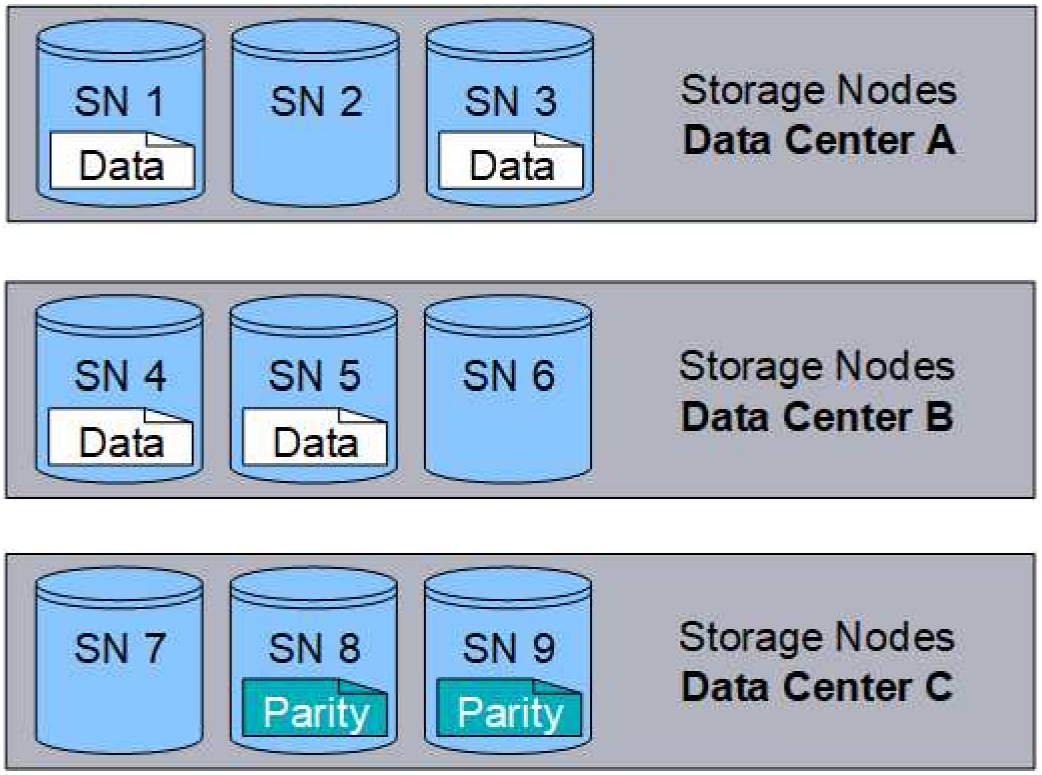
### What is erasure coding?

Erasure coding is one of two methods StorageGRID uses to store object data. When objects match an ILM rule that uses erasure coding, those objects are sliced into data fragments, additional parity fragments are computed, and each fragment is stored on a different Storage Node.

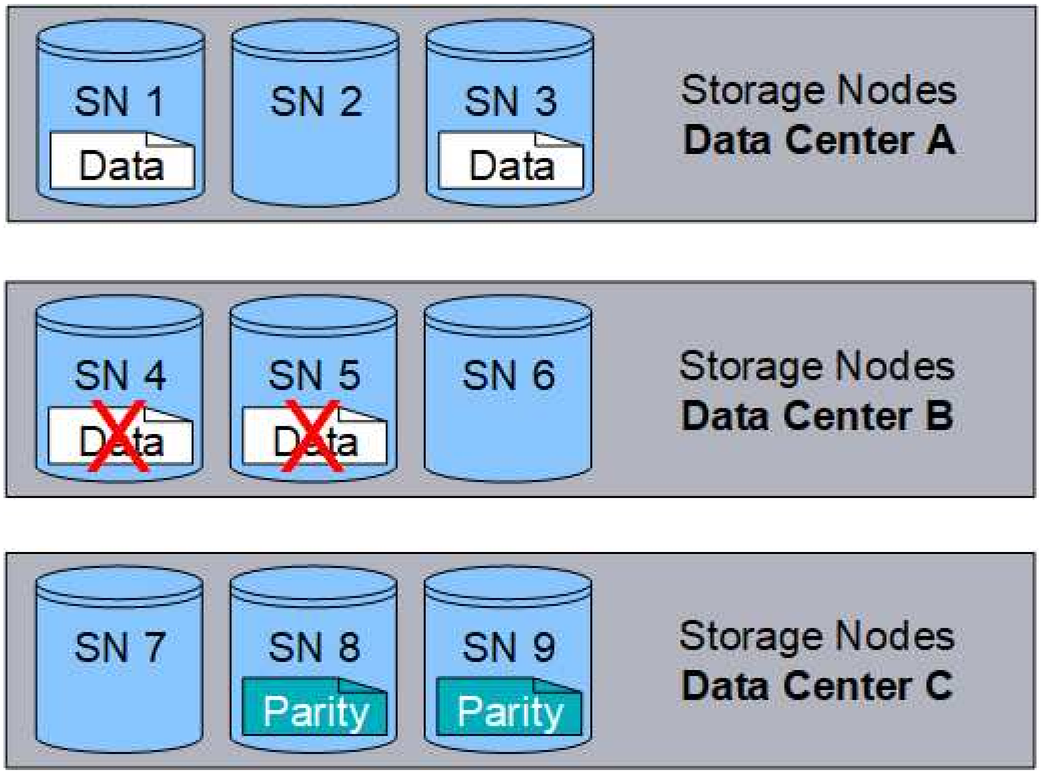
When an object is accessed, it is reassembled using the stored fragments. If a data or a parity fragment becomes corrupt or lost, the erasure-coding algorithm can recreate that fragment using a subset of the remaining data and parity fragments.

As you create ILM rules, StorageGRID creates erasure coding profiles that support those rules. You can view a list of erasure coding profiles, rename an erasure coding profile, or deactivate an erasure coding profile if it is not currently used in any ILM rules.

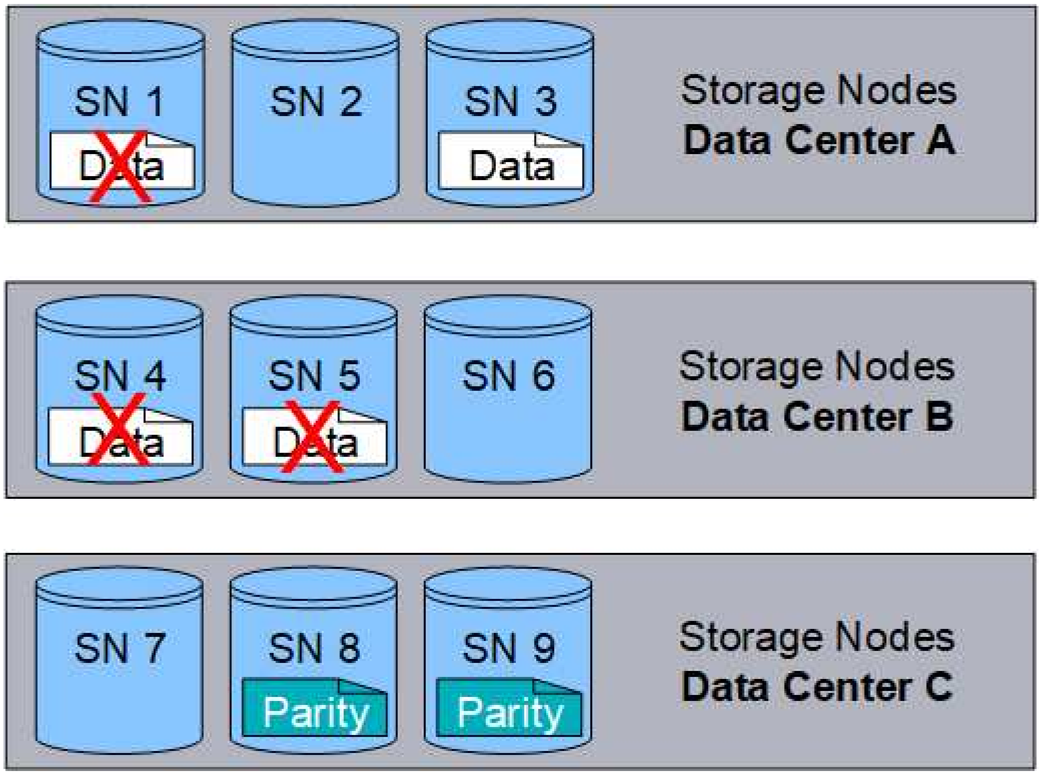
The following example illustrates the use of an erasure-coding algorithm on an object's data. In this example, the ILM rule uses a 4+2 erasure-coding scheme. Each object is sliced into four equal data fragments, and two parity fragments are computed from the object data. Each of the six fragments is stored on a different node across three data center sites to provide data protection for node failures or site loss.



The 4+2 erasure-coding scheme can be configured in various ways. For example, you can configure a single-site storage pool that contains six Storage Nodes. For site-loss protection, you can use a storage pool containing three sites with three Storage Nodes at each site. An object can be retrieved as long as any four of the six fragments (data or parity) remain available. Up to two fragments can be lost without loss of the object data. If an entire site is lost, the object can still be retrieved or repaired, as long as all of the other fragments remain accessible.



If more than two Storage Nodes are lost, the object is not retrievable.



**Related information**

- What is replication?
- What is a storage pool?
- What are erasure coding schemes?

- Rename an erasure coding profile
- Deactivate an erasure coding profile

### What are erasure coding schemes?

## Erasure-coding schemes control how many data fragments and how many parity fragments are created for each object.

When you configure the erasure coding profile for an ILM rule, you select an available erasure-coding scheme based on how many Storage Nodes and sites make up the storage pool you plan to use.

The StorageGRID system uses the Reed-Solomon erasure-coding algorithm. The algorithm slices an object into `k` data fragments and computes `m` parity fragments. The `k + m = n` fragments are spread across `n` Storage Nodes to provide data protection. An object can sustain up to `m` lost or corrupt fragments. To retrieve or repair an object, `k` fragments are needed.

When selecting the storage pool to use for a rule that will create an erasure-coded copy, use the following guidelines for storage pools:

- The storage pool must include three or more sites, or exactly one site.

  > You can't use erasure coding if the storage pool includes two sites.

  - Erasure-coding schemes for storage pools containing three or more sites
  - Erasure-coding schemes for one-site storage pools
- Don't use a storage pool that includes the default site, All Sites.
- The storage pool should include at least *`k+m`* `+1` Storage Nodes.

  The minimum number of Storage Nodes required is `k+m`. However, having at least one additional Storage Node can help prevent ingest failures or ILM backlogs if a required Storage Node is temporarily unavailable.

The storage overhead of an erasure-coding scheme is calculated by dividing the number of parity fragments (`m`) by the number of data fragments (`k`). You can use the storage overhead to calculate how much disk space each erasure-coded object requires:

*`disk space = object size + (object size * storage overhead)`*

For example, if you store a 10 MB object using the 4+2 scheme (which has 50% storage overhead), the object consumes 15 MB of grid storage. If you store the same 10 MB object using the 6+2 scheme (which has 33% storage overhead), the object consumes approximately 13.3 MB.

Select the erasure-coding scheme with the lowest total value of `k+m` that meets your needs. Erasure-coding schemes with a lower number of fragments are overall more computationally efficient, as fewer fragments are created and distributed (or retrieved) per object, can show better performance due to the larger fragment size, and can require fewer nodes be added in an expansion when more storage is required. (For information about planning a storage expansion, see the instructions for expanding StorageGRID.)

#### Erasure-coding schemes for storage pools containing three or more sites

The following table describes the erasure-coding schemes currently supported by StorageGRID for storage pools that include three or more sites. All of these schemes provide site-loss protection. One site can be lost, and the object will still be accessible.

For erasure-coding schemes that provide site-loss protection, the recommended number of Storage Nodes in the storage pool exceeds `k+m +1` because each site requires a minimum of three Storage Nodes.

| **Erasure-coding scheme (*k*+*m*)** | **Minimum number of deployed sites** | **Recommended number of Storage Nodes at each site** | **Total recommended number of Storage Nodes** | **Site loss protection?** | **Storage overhead** |
|---|---|---|---|---|---|
| 4+2 | 3 | 3 | 9 | Yes | 50% |
| 6+2 | 4 | 3 | 12 | Yes | 33% |
| 8+2 | 5 | 3 | 15 | Yes | 25% |
| 6+3 | 3 | 4 | 12 | Yes | 50% |
| 9+3 | 4 | 4 | 16 | Yes | 33% |
| 2+1 | 3 | 3 | 9 | Yes | 50% |
| 4+1 | 5 | 3 | 15 | Yes | 25% |
| 6+1 | 7 | 3 | 21 | Yes | 17% |
| 7+5 | 3 | 5 | 15 | Yes | 71% |

StorageGRID requires a minimum of three Storage Nodes per site. To use the 7+5 scheme, each site requires a minimum of four Storage Nodes. Using five Storage Nodes per site is recommended.

When selecting an erasure-coding scheme that provides site protection, balance the relative importance of the following factors:

- **Number of fragments**: Performance and expansion flexibility are generally better when the total number of fragments is lower.
- **Fault tolerance**: Fault tolerance is increased by having more parity segments (that is, when `m` has a higher value.)
- **Network traffic**: When recovering from failures, using a scheme with more fragments (that is, a higher total for `k+m`) creates more network traffic.
- **Storage overhead**: Schemes with higher overhead require more storage space per object.

For example, when deciding between a 4+2 scheme and 6+3 scheme (which both have 50% storage overhead), select the 6+3 scheme if additional fault tolerance is required. Select the 4+2 scheme if network resources are constrained. If all other factors are equal, select 4+2 because it has a lower total number of fragments.

If you are unsure of which scheme to use, select 4+2 or 6+3, or contact technical support.

#### Erasure-coding schemes for one-site storage pools

A one-site storage pool supports all of the erasure-coding schemes defined for three or more sites, provided that the site has enough Storage Nodes.

The minimum number of Storage Nodes required is `k+m`, but a storage pool with `k+m +1` Storage Nodes is recommended. For example, the 2+1 erasure-coding scheme requires a storage pool with a minimum of three Storage Nodes, but four Storage Nodes is recommended.

| Erasure-coding scheme (*k+m*) | Minimum number of Storage Nodes | Recommended number of Storage Nodes | Storage overhead |
|---|---|---|---|
| 4+2 | 6 | 7 | 50% |
| 6+2 | 8 | 9 | 33% |
| 8+2 | 10 | 11 | 25% |
| 6+3 | 9 | 10 | 50% |
| 9+3 | 12 | 13 | 33% |
| 2+1 | 3 | 4 | 50% |
| 4+1 | 5 | 6 | 25% |
| 6+1 | 7 | 8 | 17% |
| 7+5 | 12 | 13 | 71% |

### Advantages, disadvantages, and requirements for erasure coding

Before deciding whether to use replication or erasure coding to protect object data from loss, you should understand the advantages, disadvantages, and the requirements for erasure coding.

#### Advantages of erasure coding

When compared to replication, erasure coding offers improved reliability, availability, and storage efficiency.

- **Reliability**: Reliability is gauged in terms of fault tolerance—that is, the number of simultaneous failures that can be sustained without loss of data. With replication, multiple identical copies are stored on different nodes and across sites. With erasure coding, an object is encoded into data and parity fragments and distributed across many nodes and sites. This dispersal provides both site and node failure protection. When compared to replication, erasure coding provides improved reliability at comparable storage costs.
- **Availability**: Availability can be defined as the ability to retrieve objects if Storage Nodes fail or become inaccessible. When compared to replication, erasure coding provides increased availability at comparable

storage costs.

- **Storage efficiency**: For similar levels of availability and reliability, objects protected through erasure coding consume less disk space than the same objects would if protected through replication. For example, a 10 MB object that is replicated to two sites consumes 20 MB of disk space (two copies), while an object that is erasure coded across three sites with a 6+3 erasure-coding scheme only consumes 15 MB of disk space.

> ⓘ Disk space for erasure-coded objects is calculated as the object size plus the storage overhead. The storage overhead percentage is the number of parity fragments divided by the number of data fragments.

#### Disadvantages of erasure coding

When compared to replication, erasure coding has the following disadvantages:

- An increased number of Storage Nodes and sites is recommended, depending on the erasure coding scheme. In contrast, if you replicate object data, you need only one Storage Node for each copy. See Erasure coding schemes for storage pools containing three or more sites and Erasure coding schemes for one-site storage pools.
- Increased cost and complexity of storage expansions. To expand a deployment that uses replication, you add storage capacity in every location where object copies are made. To expand a deployment that uses erasure coding, you must consider both the erasure-coding scheme in use and how full existing Storage Nodes are. For example, if you wait until existing nodes are 100% full, you must add at least `k+m` Storage Nodes, but if you expand when existing nodes are 70% full, you can add two nodes per site and still maximize usable storage capacity. For more information, see Add storage capacity for erasure-coded objects.
- There are increased retrieval latencies when you use erasure coding across geographically distributed sites. The object fragments for an object that is erasure coded and distributed across remote sites take longer to retrieve over WAN connections than an object that is replicated and available locally (the same site to which the client connects).
- When you use erasure coding across geographically distributed sites, there is higher WAN network traffic usage for retrievals and repairs, especially for frequently retrieved objects or for object repairs over WAN network connections.
- When you use erasure coding across sites, the maximum object throughput declines sharply as network latency between sites increases. This decrease is due to the corresponding decrease in TCP network throughput, which affects how quickly the StorageGRID system can store and retrieve object fragments.
- Higher usage of compute resources.

#### When to use erasure coding

Erasure coding is best suited for the following requirements:

- Objects greater than 1 MB in size.

> ⓘ Erasure coding is best suited for objects greater than 1 MB. Don’t use erasure coding for objects smaller than 200 KB to avoid the overhead of managing very small erasure-coded fragments.

- Long-term or cold storage for infrequently retrieved content.
- High data availability and reliability.

- Protection against complete site and node failures.
- Storage efficiency.
- Single-site deployments that require efficient data protection with only a single erasure-coded copy rather than multiple replicated copies.
- Multiple-site deployments where the inter-site latency is less than 100 ms.

## How object retention is determined

StorageGRID provides options for both grid administrators and individual tenant users to specify how long to store objects. In general, any retention instructions provided by a tenant user take precedence over the retention instructions provided by the grid administrator.

### How tenant users control object retention

Tenant users have three primary ways to control how long their objects are stored in StorageGRID:

- If the global S3 Object Lock setting is enabled for the grid, S3 tenant users can create buckets with S3 Object Lock enabled and then use the S3 REST API to specify retain-until-date and legal hold settings for each object version added to that bucket.
  - An object version that is under a legal hold can’t be deleted by any method.
  - Before an object version’s retain-until-date is reached, that version can’t be deleted by any method.
  - Objects in buckets with S3 Object Lock enabled are retained by ILM “forever.” However, after its retain-until-date is reached, an object version can be deleted by a client request or the expiration of the bucket lifecycle. See Manage objects with S3 Object Lock.
- S3 tenant users can add a lifecycle configuration to their buckets that specifies an Expiration action. If a bucket lifecycle exists, StorageGRID stores an object until the date or number of days specified in the Expiration action are met, unless the client deletes the object first. See Create S3 lifecycle configuration.
- An S3 or Swift client can issue a delete object request. StorageGRID always prioritizes client delete requests over S3 bucket lifecycle or ILM when determining whether to delete or retain an object.

### How grid administrators control object retention

Grid administrators use ILM placement instructions to control how long objects are stored. When objects are matched by an ILM rule, StorageGRID stores those objects until the last time period in the ILM rule has elapsed. Objects are retained indefinitely if “forever” is specified for the placement instructions.

Regardless of who controls how long objects are retained, ILM settings control what types of object copies (replicated or erasure coded) are stored and where the copies are located (Storage Nodes, Cloud Storage Pools, or Archive Nodes).

### How S3 bucket lifecycle and ILM interact

The Expiration action in an S3 bucket lifecycle always overrides ILM settings. As a result, an object might be retained on the grid even after any ILM instructions for placing the object have lapsed.

### Examples for object retention

To better understand the interactions between S3 Object Lock, bucket lifecycle settings, client delete requests, and ILM, consider the following examples.

#### Example 1: S3 bucket lifecycle keeps objects longer than ILM

**ILM**

Store two copies for 1 year (365 days)

**Bucket lifecycle**

Expire objects in 2 years (730 days)

**Result**

StorageGRID stores the object for 730 days. StorageGRID uses the bucket lifecycle settings to determine whether to delete or retain an object.

> ⓘ If the bucket lifecycle specifies that objects should be kept longer than specified by ILM, StorageGRID continues to use the ILM placement instructions when determining the number and type of copies to store. In this example, two copies of the object will continue to be stored in StorageGRID from days 366 to 730.

#### Example 2: S3 bucket lifecycle expires objects before ILM

**ILM**

Store two copies for 2 years (730 days)

**Bucket lifecycle**

Expire objects in 1 year (365 days)

**Result**

StorageGRID deletes both copies of the object after day 365.

#### Example 3: Client delete overrides bucket lifecycle and ILM

**ILM**

Store two copies on Storage Nodes “forever”

**Bucket lifecycle**

Expire objects in 2 years (730 days)

**Client delete request**

Issued on day 400

**Result**

StorageGRID deletes both copies of the object on day 400 in response to the client delete request.

#### Example 4: S3 Object Lock overrides client delete request

**S3 Object Lock**

Retain-until-date for an object version is 2026-03-31. A legal hold is not in effect.

**Compliant ILM rule**

Store two copies on Storage Nodes “forever.”

**Client delete request**

Issued on 2024-03-31.

**Result**

StorageGRID will not delete the object version because the retain-until-date is still 2 years away.

## How objects are deleted

StorageGRID can delete objects either in direct response to a client request or automatically as a result of the expiration of an S3 bucket lifecycle or the requirements of the ILM policy. Understanding the different ways that objects can be deleted and how StorageGRID handles delete requests can help you manage objects more effectively.

StorageGRID can use one of two methods to delete objects:

- Synchronous deletion: When StorageGRID receives a client delete request, all object copies are removed immediately. The client is informed that deletion was successful after the copies have been removed.
- Objects are queued for deletion: When StorageGRID receives a delete request, the object is queued for deletion and the client is informed immediately that deletion was successful. Object copies are removed later by background ILM processing.

When deleting objects, StorageGRID uses the method that optimizes delete performance, minimizes potential delete backlogs, and frees space most quickly.

The table summarizes when StorageGRID uses each method.

| Method of performing deletion | When used |
|---|---|
| Objects are queued for deletion | When **any** of the following conditions are true:<br>• Automatic object deletion has been triggered by one of the following events:<br>◦ The expiration date or number of days in the lifecycle configuration for an S3 bucket is reached.<br>◦ The last time period specified in an ILM rule elapses.<br>**Note:** Objects in a bucket that has S3 Object Lock enabled can’t be deleted if they are under a legal hold or if a retain-until-date has been specified but not yet met.<br>• An S3 or Swift client requests deletion and one or more of these conditions is true:<br>◦ Copies can’t be deleted within 30 seconds because, for example, an object location is temporarily unavailable.<br>◦ Background deletion queues are idle. |

| Method of performing deletion | When used |
|---|---|
| Objects are removed immediately (synchronous deletion) | When an S3 or Swift client makes a delete request and **all** of the following conditions are met:<br>• All copies can be removed within 30 seconds.<br>• Background deletion queues contain objects to process. |

When S3 or Swift clients make delete requests, StorageGRID begins by adding objects to the delete queue. It then switches to performing synchronous deletion. Making sure that the background deletion queue has objects to process allows StorageGRID to process deletes more efficiently, especially for low concurrency clients, while helping to prevent client delete backlogs.

### Time required to delete objects

The way that StorageGRID deletes objects can affect how the system appears to perform:

- When StorageGRID performs synchronous deletion, it can take StorageGRID up to 30 seconds to return a result to the client. This means that deletion can appear to be happening more slowly, even though copies are actually being removed more quickly than they are when StorageGRID queues objects for deletion.
- If you are closely monitoring delete performance during a bulk delete, you might notice that the deletion rate appears to slow after a certain number of objects have been deleted. This change occurs when StorageGRID shifts from queuing objects for deletion to performing synchronous deletion. The apparent reduction in the deletion rate does not mean that object copies are being removed more slowly. On the contrary, it indicates that on average, space is now being freed more quickly.

If you are deleting large numbers of objects and your priority is to free space quickly, consider using a client request to delete objects rather than deleting them using ILM or other methods. In general, space is freed more quickly when deletion is performed by clients because StorageGRID can use synchronous deletion.

The amount of time required to free space after an object is deleted depends on several factors:

- Whether object copies are synchronously removed or are queued for removal later (for client delete requests).
- Other factors such as the number of objects in the grid or the availability of grid resources when object copies are queued for removal (for both client deletes and other methods).

### How S3 versioned objects are deleted

When versioning is enabled for an S3 bucket, StorageGRID follows Amazon S3 behavior when responding to delete requests, whether those requests come from an S3 client, the expiration of an S3 bucket lifecycle, or the requirements of the ILM policy.

When objects are versioned, object delete requests don’t delete the current version of the object and don’t free space. Instead, an object delete request creates a delete marker as the current version of the object, which makes the previous version of the object “noncurrent.”

Even though the object has not been removed, StorageGRID behaves as though the current version of the object is no longer available. Requests to that object return 404 NotFound. However, because noncurrent object data has not been removed, requests that specify a noncurrent version of the object can succeed.

To free space when deleting versioned objects, use one of the following:

- **S3 client request**: Specify the object version ID in the S3 DELETE Object request (`DELETE /object?versionId=ID`). Keep in mind that this request only removes object copies for the specified version (the other versions are still taking up space).
- **Bucket lifecycle**: Use the `NoncurrentVersionExpiration` action in the bucket lifecycle configuration. When the number of NoncurrentDays specified is met, StorageGRID permanently removes all copies of noncurrent object versions. These object versions can't be recovered.

  The `NewerNoncurrentVersions` action in the bucket lifecycle configuration specifies the number of noncurrent versions retained in a versioned S3 bucket. If there are more noncurrent versions than `NewerNoncurrentVersions` specifies, StorageGRID removes the older versions once the NoncurrentDays value has elapsed. The `NewerNoncurrentVersions` threshold overrides lifecycle rules provided by ILM, meaning that a noncurrent object with a version within the `NewerNoncurrentVersions` threshold is retained if ILM requests its deletion.
- **ILM**: Clone the active policy and add two ILM rules to the new proposed policy:
  - First rule: Use “Noncurrent time” as the Reference time to match the noncurrent versions of the object. In Step 1 (Enter details) of the Create an ILM rule wizard, select **Yes** for the question, “Apply this rule to older object versions only (in S3 buckets with versioning enabled)?”
  - Second rule: Use **Ingest time** to match the current version. The “Noncurrent time” rule must appear in the policy above the **Ingest time** rule.

#### How S3 delete markers are deleted

When a versioned object is deleted, StorageGRID creates a delete marker as the current version of the object. To remove the zero-byte delete marker from the bucket, the S3 client must explicitly delete the object version. Delete markers aren't removed by ILM, bucket lifecycle rules, or Delete objects in bucket operations.

### Related information

- Use S3 REST API
- Example 4: ILM rules and policy for S3 versioned objects

# Create and assign storage grades

Storage grades identify the type of storage used by a Storage Node. You can create storage grades if you want ILM rules to place certain objects on certain Storage Nodes.

### Before you begin

- You are signed in to the Grid Manager using a supported web browser.
- You have specific access permissions.

### About this task

When you first install StorageGRID, the **Default** storage grade is automatically assigned to every Storage Node in your system. As required, you can optionally define custom storage grades and assign them to different Storage Nodes.

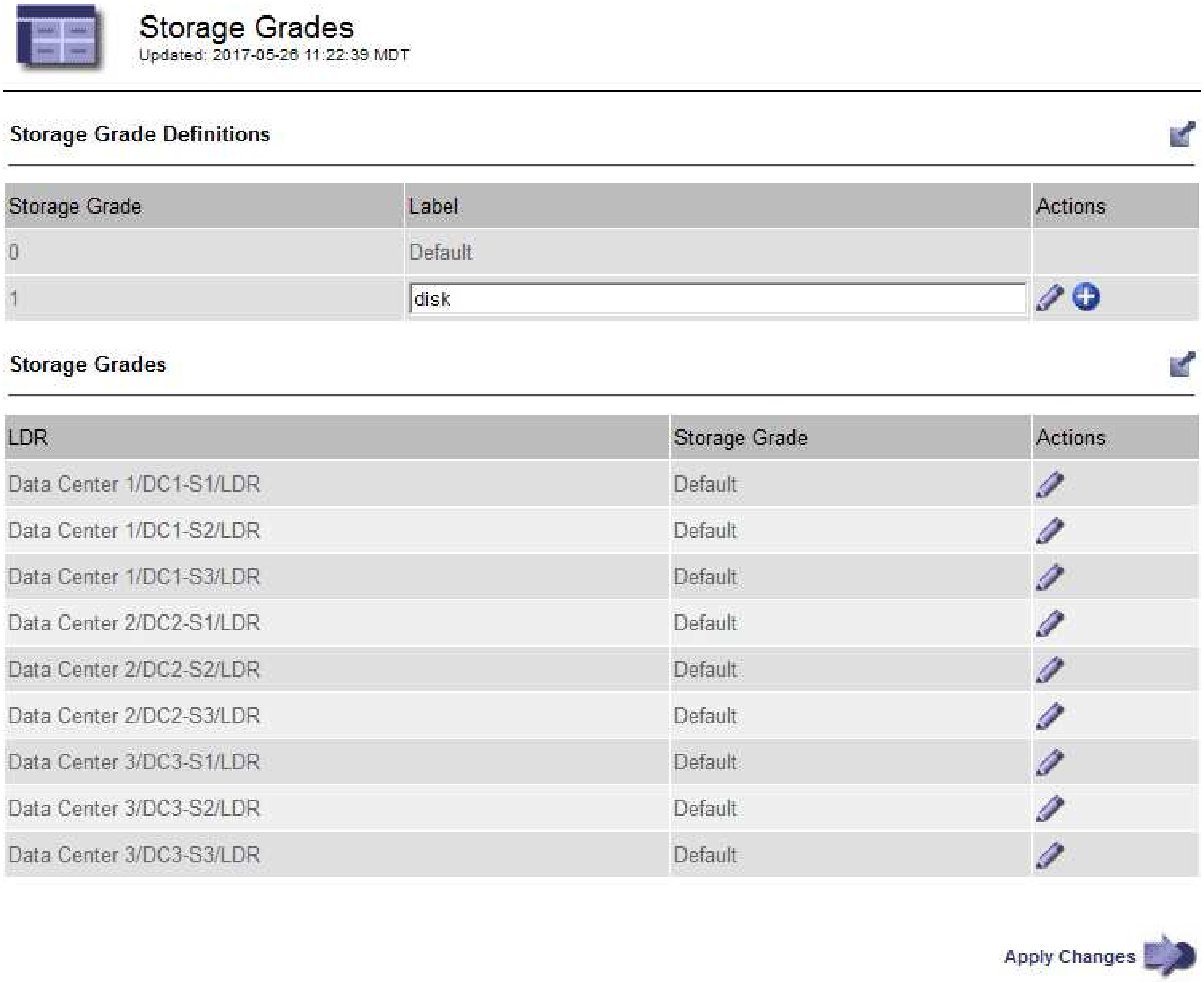
Using custom storage grades allows you to create ILM storage pools that contain only a specific type of Storage Node. For example, you might want certain objects to be stored on your fastest Storage Nodes, such as StorageGRID all-flash storage appliances.

If storage grade is not a concern (for example, all Storage Nodes are identical), you can skip this procedure and use the **includes all storage grades** selection for the storage grade when you create storage pools. Using this selection ensures that the storage pool will include every Storage Node at the site, regardless of its storage grade.

> Don’t create more storage grades than necessary. For example, don’t create a storage grade for each Storage Node. Instead, assign each storage grade to two or more nodes. Storage grades assigned to only one node can cause ILM backlogs if that node becomes unavailable.

**Steps**

1. Select **ILM** > **Storage grades**.
2. Define custom storage grades:
   a. For each custom storage grade you want to add, select **Insert** ⊕ to add a row.
   b. Enter a descriptive label.



Storage Grades

Updated: 2017-05-26 11:22:39 MDT

Storage Grade Definitions

| Storage Grade | Label | Actions |
|---|---|---|
| 0 | Default | |
| 1 | disk | ✎ ⊕ |

Storage Grades

| LDR | Storage Grade | Actions |
|---|---|---|
| Data Center 1/DC1-S1/LDR | Default | ✎ |
| Data Center 1/DC1-S2/LDR | Default | ✎ |
| Data Center 1/DC1-S3/LDR | Default | ✎ |
| Data Center 2/DC2-S1/LDR | Default | ✎ |
| Data Center 2/DC2-S2/LDR | Default | ✎ |
| Data Center 2/DC2-S3/LDR | Default | ✎ |
| Data Center 3/DC3-S1/LDR | Default | ✎ |
| Data Center 3/DC3-S2/LDR | Default | ✎ |
| Data Center 3/DC3-S3/LDR | Default | ✎ |

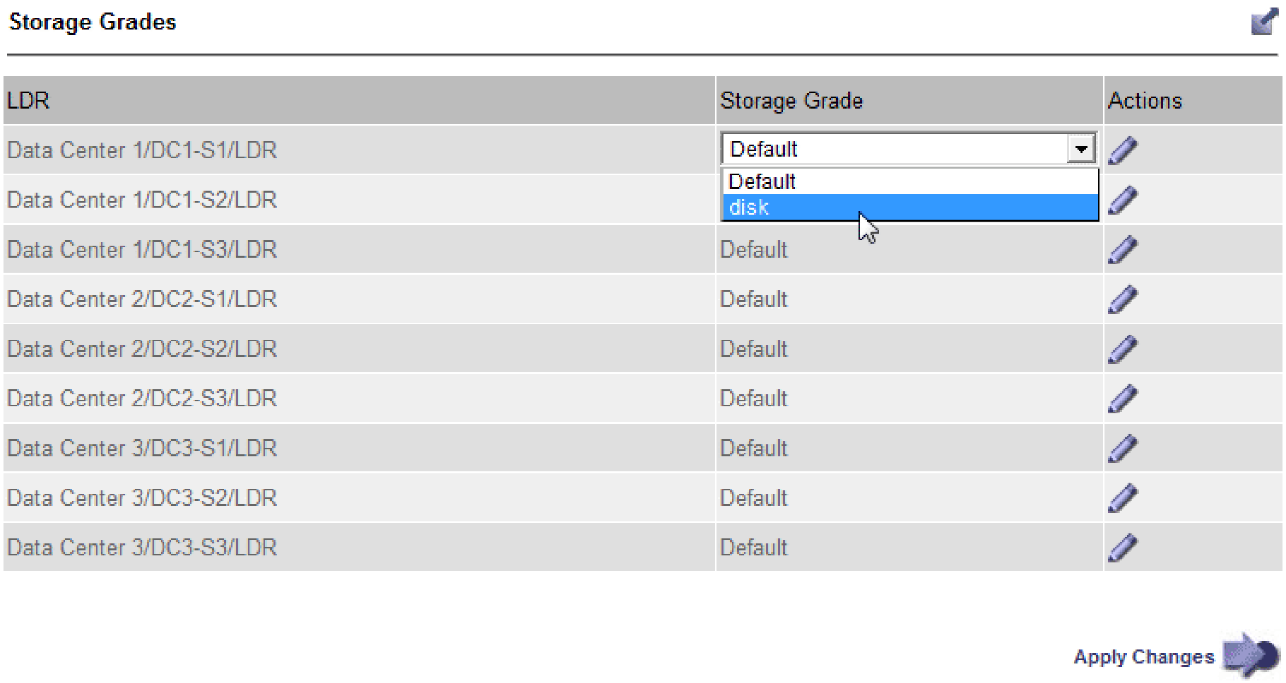
   c. Select **Apply Changes**.
   d. Optionally, if you need to modify a saved label, select **Edit** ✎ and select **Apply Changes**.

   > You can’t delete storage grades.

3. Assign new storage grades to Storage Nodes:

a. Locate the Storage Node in the LDR list, and select its **Edit** icon ✎.

b. Select the appropriate storage grade from the list.



Storage Grades

| LDR | Storage Grade | Actions |
|---|---|---|
| Data Center 1/DC1-S1/LDR | Default | ✎ |
| Data Center 1/DC1-S2/LDR |  | ✎ |
| Data Center 1/DC1-S3/LDR | Default | ✎ |
| Data Center 2/DC2-S1/LDR | Default | ✎ |
| Data Center 2/DC2-S2/LDR | Default | ✎ |
| Data Center 2/DC2-S3/LDR | Default | ✎ |
| Data Center 3/DC3-S1/LDR | Default | ✎ |
| Data Center 3/DC3-S2/LDR | Default | ✎ |
| Data Center 3/DC3-S3/LDR | Default | ✎ |

Assign a storage grade to a given Storage Node only once. A Storage Node recovered from failure maintains the previously assigned storage grade. Don't change this assignment after the ILM policy is activated. If the assignment is changed, data is stored based on the new storage grade.

c. Select **Apply Changes**.

# Use storage pools

## What is a storage pool?

A storage pool is a logical grouping of Storage Nodes or Archive Nodes.

When you install StorageGRID, one storage pool per site is automatically created. You can configure additional storage pools as needed for your storage requirements.

Support for Archive Nodes (for both archiving to the cloud using the S3 API and archiving to tape using TSM middleware) is deprecated and will be removed in a future release. Moving objects from an Archive Node to an external archival storage system has been replaced by ILM Cloud Storage Pools, which offer more functionality.

See Use Cloud Storage Pools.

Storage pools have two attributes:

- **Storage grade**: For Storage Nodes, the relative performance of backing storage.
- **Site**: The data center where objects will be stored.

Storage pools are used in ILM rules to determine where object data is stored and the type of storage used. When you configure ILM rules for replication, you select one or more storage pools that include either Storage Nodes or Archive Nodes. When you create erasure coding profiles, you select a storage pool that includes Storage Nodes.

## Guidelines for creating storage pools

Configure and use storage pools to protect against data loss by distributing data across multiple sites. Replicated copies and erasure-coded copies require different storage pool configurations.

See Examples of enabling site-loss protection using replication and erasure coding.

### Guidelines for all storage pools

- Keep storage pool configurations as simple as possible. Don’t create more storage pools than necessary.
- Create storage pools with as many nodes as possible. Each storage pool should contain two or more nodes. A storage pool with insufficient nodes can cause ILM backlogs if a node becomes unavailable.
- Avoid creating or using storage pools that overlap (contain one or more of the same nodes). If storage pools overlap, more than one copy of object data might be saved on the same node.
- In general, don’t use the All Storage Nodes storage pool (StorageGRID 11.6 and earlier) or the All Sites site. These items are automatically updated to include any new sites you add in an expansion, which might not be the behavior you want.

### Guidelines for storage pools used for replicated copies

- For site-loss protection using replication, specify one or more site-specific storage pools in the placement instructions for each ILM rule.

  One storage pool is automatically created for each site during StorageGRID installation.

  Using a storage pool for each site ensures that replicated object copies are placed exactly where you expect (for example, one copy of every object at each site for site-loss protection).

- If you add a site in an expansion, create a new storage pool that contains only the new site. Then, update ILM rules to control which objects are stored on the new site.
- If the number of copies is less than the number of storage pools, the system distributes the copies to balance disk usage among the pools.
- If the storage pools overlap (contain the same Storage Nodes), all copies of the object might be saved at only one site. You must ensure that the selected storage pools don’t contain the same Storage Nodes.

### Guidelines for storage pools used for erasure-coded copies

- For site-loss protection using erasure coding, create storage pools that consist of at least three sites. If a storage pool includes only two sites, you can’t use that storage pool for erasure coding. No erasure-coding schemes are available for a storage pool that has two sites.
- The number of Storage Nodes and sites contained in the storage pool determine which erasure-coding schemes are available.
- If possible, a storage pool should include more than the minimum number of Storage Nodes required for the erasure-coding scheme you select. For example, if you use a 6+3 erasure-coding scheme, you must have at least nine Storage Nodes. However, having at least one additional Storage Node per site is

recommended.

- Distribute Storage Nodes across sites as evenly as possible. For example, to support a 6+3 erasure-coding scheme, configure a storage pool that includes at least three Storage Nodes at three sites.
- If you have high throughput requirements, using a storage pool that includes multiple sites is not recommended if the network latency between sites is greater than 100 ms. As latency increases, the rate at which StorageGRID can create, place, and retrieve object fragments decreases sharply due to the decrease in TCP network throughput.

  The decrease in throughput affects the maximum achievable rates of object ingest and retrieval (when Balanced or Strict are selected as the ingest behavior) or could lead to ILM queue backlogs (when Dual commit is selected as the ingest behavior). See ILM rule ingest behavior.

  > ⓘ If your grid includes only one site, you are prevented from using the All Storage Nodes storage pool (StorageGRID 11.6 and earlier) or the All Sites default site in an erasure coding profile. This behavior prevents the profile from becoming invalid if a second site is added.

- You can’t use Archive Nodes for erasure-coded data.

### Guidelines for storage pools used for archived copies

> ! Support for Archive Nodes (for both archiving to the cloud using the S3 API and archiving to tape using TSM middleware) is deprecated and will be removed in a future release. Moving objects from an Archive Node to an external archival storage system has been replaced by ILM Cloud Storage Pools, which offer more functionality.
>
> See Migrate objects to a Cloud Storage Pool.
>
> In addition, you should remove Archive Nodes from the active ILM policy in StorageGRID 11.7 or earlier. Removing object data stored on Archive Nodes will simplify future upgrades. See Working with ILM rules and ILM policies.

- You can’t create a storage pool that includes both Storage Nodes and Archive Nodes. Archived copies require a storage pool that only includes Archive Nodes.
- When using a storage pool that includes Archive Nodes, you should also maintain at least one replicated or erasure-coded copy on a storage pool that includes Storage Nodes.
- If the global S3 Object Lock setting is enabled and you are creating a compliant ILM rule, you can’t use a storage pool that includes Archive Nodes. See the instructions for managing objects with S3 Object Lock.
- If an Archive Node’s Target Type is Cloud Tiering - Simple Storage Service (S3), the Archive Node must be in its own storage pool.

## Enable site-loss protection

If your StorageGRID deployment includes more than one site, you can use replication and erasure coding with appropriately configured storage pools to enable site-loss protection.

Replication and erasure coding require different storage pool configurations:

- To use replication for site-loss protection, use the site-specific storage pools that are automatically created during StorageGRID installation. Then create ILM rules with placement instructions that specify multiple storage pools so that one copy of each object will be placed at each site.

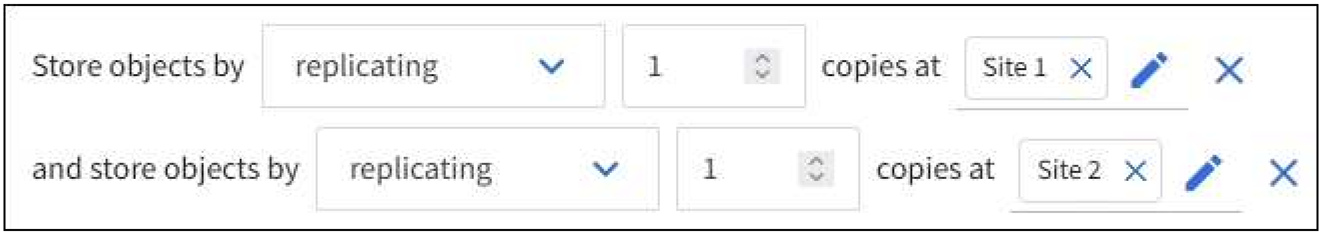
- To use erasure coding for site-loss protection, create storage pools that consist of multiple sites. Then create ILM rules that use one storage pool consisting of multiple sites and any available erasure-coding schema.

### Replication example

By default, one storage pool is created for each site during StorageGRID installation. Having storage pools that consist of only one site enables you to configure ILM rules that use replication for site-loss protection. In this example:

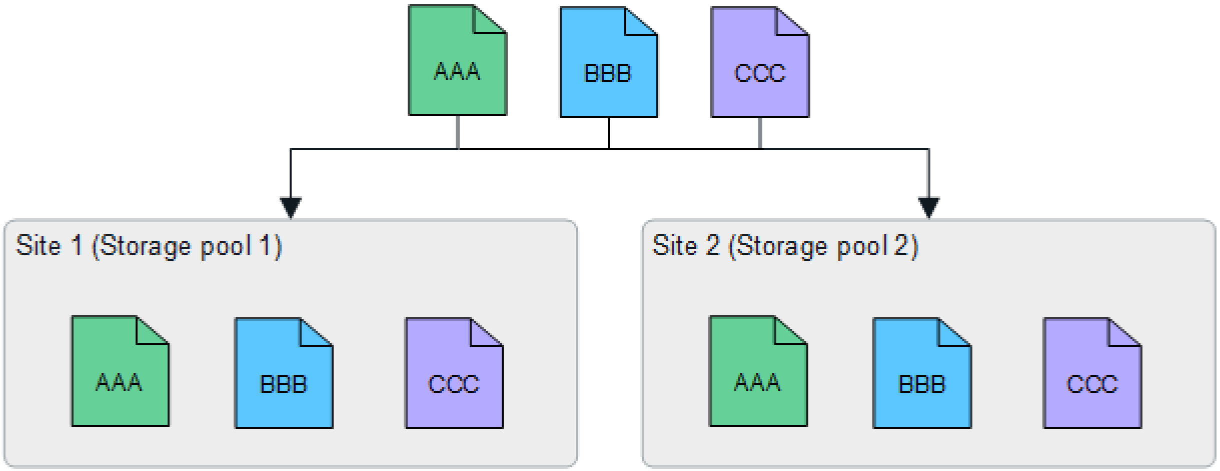
- Storage pool 1 contains Site 1
- Storage pool 2 contains Site 2
- The ILM rule contains two placements:
  - Store objects by replicating 1 copy at Site 1
  - Store objects by replicating 1 copy at Site 2

ILM rule placements:



Store objects by replicating 1 copies at Site 1

and store objects by replicating 1 copies at Site 2



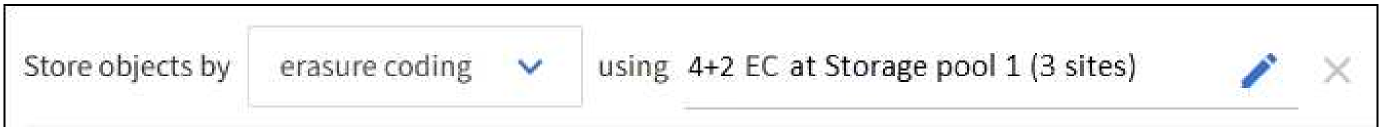
If one site is lost, copies of the objects are available at the other site.

### Erasure coding example

Having storage pools that consist of more than one site per storage pool enables you to configure ILM rules that use erasure coding for site-loss protection. In this example:

- Storage pool 1 contains Sites 1 through 3
- The ILM rule contains one placement: Store objects by erasure coding using a 4+2 EC scheme at Storage pool 1, which contains three sites

ILM rule placements:



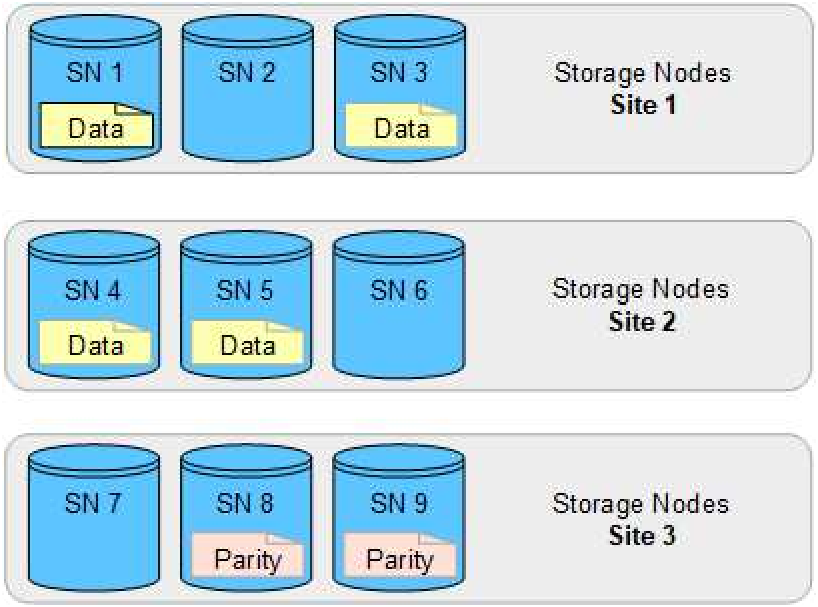
Store objects by erasure coding using 4+2 EC at Storage pool 1 (3 sites)

In this example:

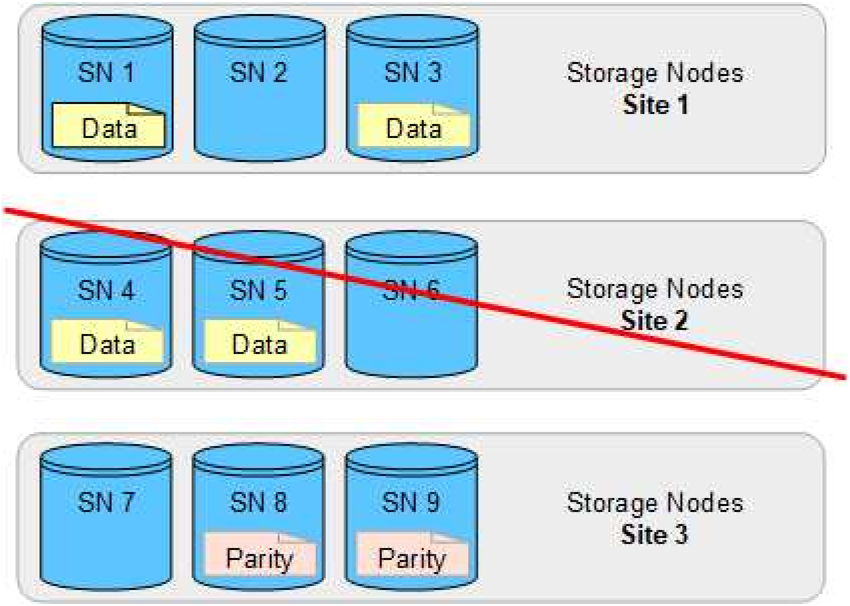
- The ILM rule uses a 4+2 erasure-coding scheme.
- Each object is sliced into four equal data fragments, and two parity fragments are computed from the object data.
- Each of the six fragments is stored on a different node across three data center sites to provide data protection for node failures or site loss.

Erasure coding is allowed in storage pools containing any number of sites *except* two sites.

ILM rule using 4+2 erasure-coding scheme:



If one site is lost, data can still be recovered:



## Create a storage pool

You create storage pools to determine where the StorageGRID system stores object data and the type of storage used. Each storage pool includes one or more sites and one or more storage grades.

> When you install StorageGRID 11.7 on a new grid, storage pools are automatically created for each site to reduce the number of steps required to create new ILM rules. However, during upgrades to StorageGRID 11.7, storage pools aren’t created for each site.

If you want to create Cloud Storage Pools to store object data outside of your StorageGRID system, see the information about using Cloud Storage Pools.

### Before you begin

- You are signed in to the Grid Manager using a supported web browser.
- You have specific access permissions.
- You have reviewed the guidelines for creating storage pools.

### About this task

Storage pools determine where object data is stored. The number of storage pools you need depends on the number of sites in your grid and on the types of copies you want: replicated or erasure-coded.

- For replication and single-site erasure coding, create a storage pool for each site. For example, if you want to store replicated object copies at three sites, create three storage pools.
- For erasure coding at three or more sites, create one storage pool that includes an entry for each site. For example, if you want to erasure code objects across three sites, create one storage pool.

  > Don’t include the All Sites site in a storage pool that will be used in an erasure coding profile. Instead, add a separate entry to the storage pool for each site that will store erasure coded data. See this step for an example.

- If you have more than one storage grade, don’t create a storage pool that includes different storage grades at a single site. See the Guidelines for creating storage pools.

**Steps**

1. Select **ILM** > **Storage pools**.

   The Storage pools tab lists all defined storage pools.

   > ⓘ For new installations of StorageGRID 11.6 or earlier, the All Storage Nodes storage pool is automatically updated whenever you add new data center sites. Don’t use this pool in ILM rules.

2. To create a new storage pool, select **Create**.
3. Enter a unique name for the storage pool. Use a name that will be easy to identify when you configure erasure coding profiles and ILM rules.
4. From the **Site** drop-down list, select a site for this storage pool.

   When you select a site, the number of Storage Nodes and Archive Nodes in the table are automatically updated.

   In general, don’t use the All Sites site in any storage pool. ILM rules that use an All Sites storage pool place objects at any available site, giving you less control of object placement. Also, an All Sites storage pool uses the Storage Nodes at a new site immediately, which might not be the behavior you expect.

5. From the **Storage grade** drop-down list, select the type of storage that will be used if an ILM rule uses this storage pool.

   The storage grade, includes all storage grades, includes all Storage Nodes at the selected site. The default Archive Nodes storage grade includes all Archive Nodes at the selected site. If you created additional storage grades for the Storage Nodes in your grid, they are listed in the drop-down.

6.  If you want to use the storage pool in a multi-site erasure coding profile, select **Add more nodes** to add an entry for each site to the storage pool.

   > ⓘ You are prevented from creating duplicate entries or from creating a storage pool that includes both the Archive Nodes storage grade and any storage grade that contains Storage Nodes.
   >
   > You are warned if you add more than one entry with different storage grades for a site.

   To remove an entry, select the delete icon ✕.

7. When you are satisfied with your selections, select **Save**.

   The new storage pool is added to the list.

## View storage pool details

You can view the details of a storage pool to determine where the storage pool is used and to see which nodes and storage grades are included.

**Before you begin**

- You are signed in to the Grid Manager using a supported web browser.
- You have specific access permissions.

**Steps**

1. Select **ILM** > **Storage pools**.

   The Storage pools table includes the following information for each storage pool that includes Storage Nodes:

   - **Name**: The unique display name of the storage pool.
   - **Node count**: The number of nodes in the storage pool.
   - **Storage usage**: The percentage of the total usable space that has been used for object data on this node. This value does not include object metadata.
   - **Total capacity**: The size of the storage pool, which equals the total amount of usable space for object data for all nodes in the storage pool.
   - **ILM usage**: How the storage pool is currently being used. A storage pool might be unused or it might be used in one or more ILM rules, erasure coding profiles, or both.

   > You can't remove a storage pool if it is being used.

2. To view details about a specific storage pool, select its name.

   The details page for the storage pool appears.

3. View the **Nodes** tab to learn about the Storage Nodes or Archive Nodes included in the storage pool.

   The table includes the following information for each node:

   - Node name
   - Site name
   - Storage grade
   - Storage usage (%): The percentage of the total usable space for object data that has been used for the Storage Node. This field is not visible for Archive Node pools.

   > The same Storage usage (%) value is also shown in the Storage Used - Object Data chart for each Storage Node (select **NODES** > ***Storage Node*** > **Storage**).

4. Select the **ILM usage** tab to determine if the storage pool is currently being used in any ILM rules or erasure coding profiles.
5. Optionally, go to the **ILM rules page** to learn about and manage any rules that use the storage pool.

   See the instructions for working with ILM rules.

## Edit storage pool

You can edit a storage pool to change its name or to update sites and storage grades.

**Before you begin**

- You are signed in to the Grid Manager using a supported web browser.
- You have specific access permissions.
- You have reviewed the guidelines for creating storage pools.
- If you plan to edit a storage pool that is used by a rule in the active ILM policy, you have considered how your changes will affect object data placement.

### About this task

If you are adding a new site or storage grade to a storage pool that is used in the active ILM policy, be aware that the Storage Nodes in the new site or storage grade will not be used automatically. To force StorageGRID to use a new site or storage grade, you must activate a new ILM policy after saving the edited storage pool.

### Steps

1. Select **ILM** > **Storage pools**.
2. Select the checkbox for the storage pool you want to edit.

   You can't edit the All Storage Nodes storage pool (StorageGRID 11.6 and earlier).

3. Select **Edit**.
4. As required, change the storage pool name.
5. As required, select other sites and storage grades.

   > You are prevented from changing the site or storage grade if the storage pool is used in an erasure coding profile and the change would cause the erasure-coding scheme to become invalid. For example, if a storage pool used in a erasure coding profile currently includes a storage grade with only one site, you are prevented from using a storage grade with two sites because the change would make the erasure-coding scheme invalid.

6. Select **Save**.

### After you finish

If you added a new site or storage grade to a storage pool used in the active ILM policy, activate a new ILM policy to force StorageGRID to use the new site or storage grade. For example, clone your existing ILM policy and then activate the clone. See Work with ILM rules and ILM policies.

## Remove a storage pool

You can remove a storage pool that is not being used.

### Before you begin

- You are signed in to the Grid Manager using a supported web browser.
- You have the required access permissions.

### Steps

1. Select **ILM** > **Storage pools**.
2. Look at the ILM usage column in the table to determine whether you can remove the storage pool.

   You can't remove a storage pool if it is being used in an ILM rule or in an erasure coding profile. As required, select ***storage pool name*** > **ILM usage** to determine where the storage pool is used.

3. If the storage pool you want to remove is not being used, select the checkbox.
4. Select **Remove**.
5. Select **OK**.

# Use Cloud Storage Pools

## What is a Cloud Storage Pool?

A Cloud Storage Pool lets you use ILM to move object data outside of your StorageGRID system. For example, you might want to move infrequently accessed objects to lower-cost cloud storage, such as Amazon S3 Glacier, S3 Glacier Deep Archive, Google Cloud, or the Archive access tier in Microsoft Azure Blob storage. Or, you might want to maintain a cloud backup of StorageGRID objects to enhance disaster recovery.

From an ILM perspective, a Cloud Storage Pool is similar to a storage pool. To store objects in either location, you select the pool when creating the placement instructions for an ILM rule. However, while storage pools consist of Storage Nodes or Archive Nodes within the StorageGRID system, a Cloud Storage Pool consists of an external bucket (S3) or container (Azure Blob storage).

> Moving objects from an Archive Node to an external archival storage system through the S3 API is deprecated and has been replaced by ILM Cloud Storage Pools, which offer more functionality. If you are currently using an Archive Node with the Cloud Tiering - Simple Storage Service (S3) option, migrate your objects to a Cloud Storage Pool instead.

The table compares storage pools to Cloud Storage Pools and shows the high-level similarities and differences.

| | Storage pool | Cloud Storage Pool |
|---|---|---|
| How is it created? | Using the **ILM** > **Storage pools** option in Grid Manager. | Using the **ILM** > **Storage pools** > **Cloud Storage Pools** option in Grid Manager.<br>You must set up the external bucket or container before you can create the Cloud Storage Pool. |
| How many pools can you create? | Unlimited. | Up to 10. |

|  | Storage pool | Cloud Storage Pool |
|---|---|---|
| Where are objects stored? | On one or more Storage Nodes or Archive Nodes within StorageGRID. | In Amazon S3 bucket, Azure Blob storage container, or Google Cloud that is external to the StorageGRID system.<br>If the Cloud Storage Pool is an Amazon S3 bucket:<br>• You can optionally configure a bucket lifecycle to transition objects to low-cost, long-term storage, such as Amazon S3 Glacier or S3 Glacier Deep Archive. The external storage system must support the Glacier storage class and the S3 POST Object restore API.<br>• You can create Cloud Storage Pools for use with AWS Commercial Cloud Services (C2S), which supports the AWS Secret Region.<br>If the Cloud Storage Pool is an Azure Blob storage container, StorageGRID transitions the object to the Archive tier.<br>**Note:** In general, don’t configure Azure Blob storage lifecycle management for the container used for a Cloud Storage Pool. POST Object restore operations on objects in the Cloud Storage Pool can be affected by the configured lifecycle. |
| What controls object placement? | An ILM rule in the active ILM policy. | An ILM rule in the active ILM policy. |
| Which data protection method is used? | Replication or erasure coding. | Replication. |
| How many copies of each object are allowed? | Multiple. | One copy in the Cloud Storage Pool and, optionally, one or more copies in StorageGRID.<br>**Note:** You can’t store an object in more than one Cloud Storage Pool at any given time. |
| What are the advantages? | Objects are quickly accessible at any time. | Low-cost storage. |
|  |  | **Note**: FabricPool data can’t be tiered to Cloud Storage Pools. Objects with S3 Object Lock enabled can’t be placed in Cloud Storage Pools. |

## Lifecycle of a Cloud Storage Pool object

Before implementing Cloud Storage Pools, review the lifecycle of objects that are stored in each type of Cloud Storage Pool.

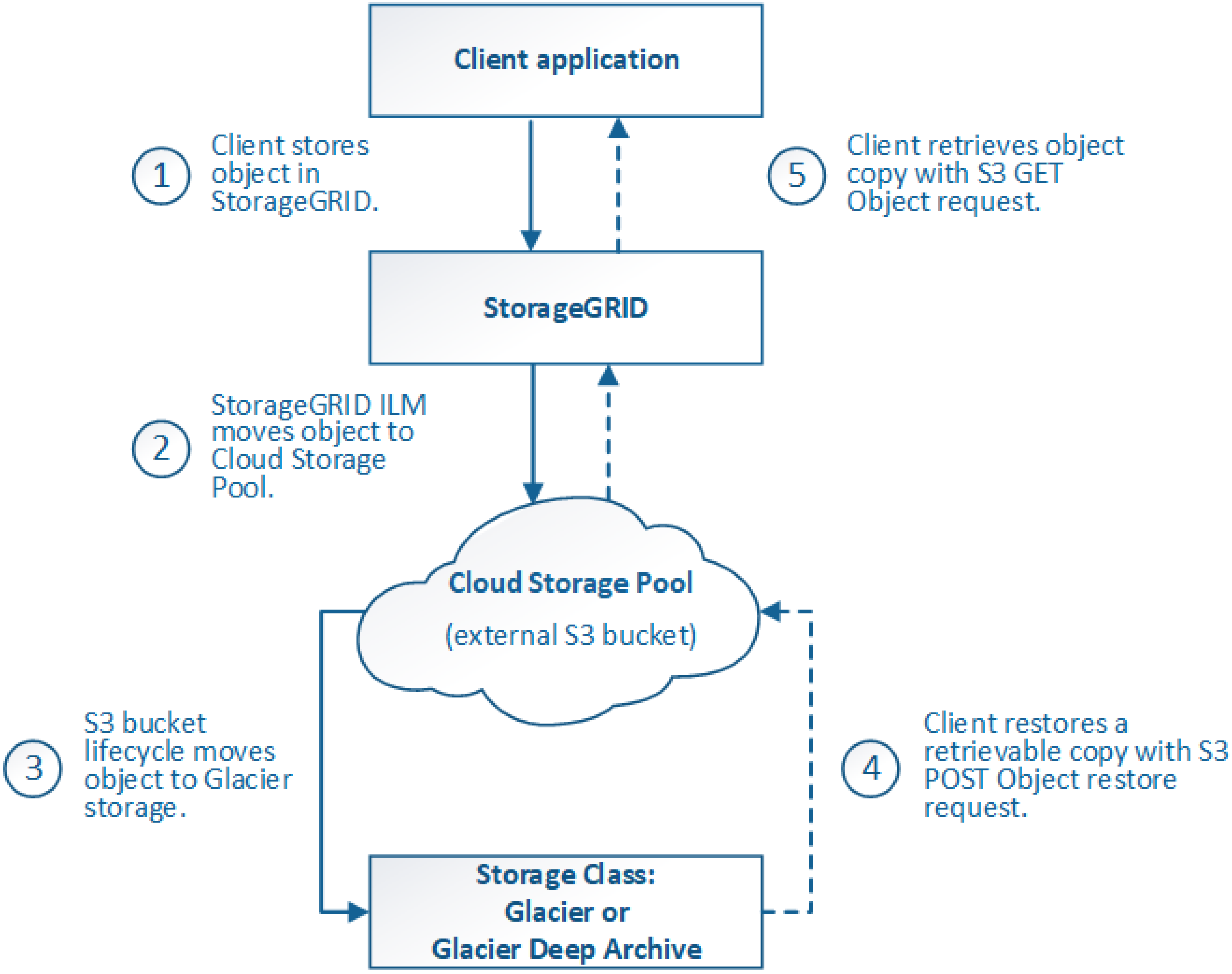
- S3: Lifecycle of a Cloud Storage Pool object
- Azure: Lifecycle of a Cloud Storage Pool object

### S3: Lifecycle of a Cloud Storage Pool object

The figure shows the lifecycle stages of an object that is stored in an S3 Cloud Storage Pool.

ⓘ In the figure and explanations, “Glacier” refers to both the Glacier storage class and the Glacier Deep Archive storage class, with one exception: the Glacier Deep Archive storage class does not support the Expedited restore tier. Only Bulk or Standard retrieval is supported.

ⓘ The Google Cloud Platform (GCP) supports object retrieval from long-term storage without requiring a POST Restore operation.


Client application
1 Client stores object in StorageGRID.
5 Client retrieves object copy with S3 GET Object request.
StorageGRID
2 StorageGRID ILM moves object to Cloud Storage Pool.
Cloud Storage Pool
(external S3 bucket)
3 S3 bucket lifecycle moves object to Glacier storage.
4 Client restores a retrievable copy with S3 POST Object restore request.
Storage Class:
Glacier or
Glacier Deep Archive

1. **Object stored in StorageGRID**

To start the lifecycle, a client application stores an object in StorageGRID.

2. **Object moved to S3 Cloud Storage Pool**
   - When the object is matched by an ILM rule that uses an S3 Cloud Storage Pool as its placement location, StorageGRID moves the object to the external S3 bucket specified by the Cloud Storage Pool.
   - When the object has been moved to the S3 Cloud Storage Pool, the client application can retrieve it using an S3 GET Object request from StorageGRID, unless the object has been transitioned to Glacier storage.
3. **Object transitioned to Glacier (non-retrievable state)**
   - Optionally, the object can be transitioned to Glacier storage. For example, the external S3 bucket might use lifecycle configuration to transition an object to Glacier storage immediately or after some number of days.

     > If you want to transition objects, you must create a lifecycle configuration for the external S3 bucket, and you must use a storage solution that implements the Glacier storage class and supports the S3 POST Object restore API.

     > Don’t use Cloud Storage Pools for objects that have been ingested by Swift clients. Swift does not support POST Object restore requests, so StorageGRID will not be able to retrieve any Swift objects that have been transitioned to S3 Glacier storage. Issuing a Swift GET object request to retrieve these objects will fail (403 Forbidden).

   - During the transition, the client application can use an S3 HEAD Object request to monitor the object’s status.
4. **Object restored from Glacier storage**

   If an object has been transitioned to Glacier storage, the client application can issue an S3 POST Object restore request to restore a retrievable copy to the S3 Cloud Storage Pool. The request specifies how many days the copy should be available in the Cloud Storage Pool and the data-access tier to use for the restore operation (Expedited, Standard, or Bulk). When the expiration date of the retrievable copy is reached, the copy is automatically returned to a non-retrievable state.

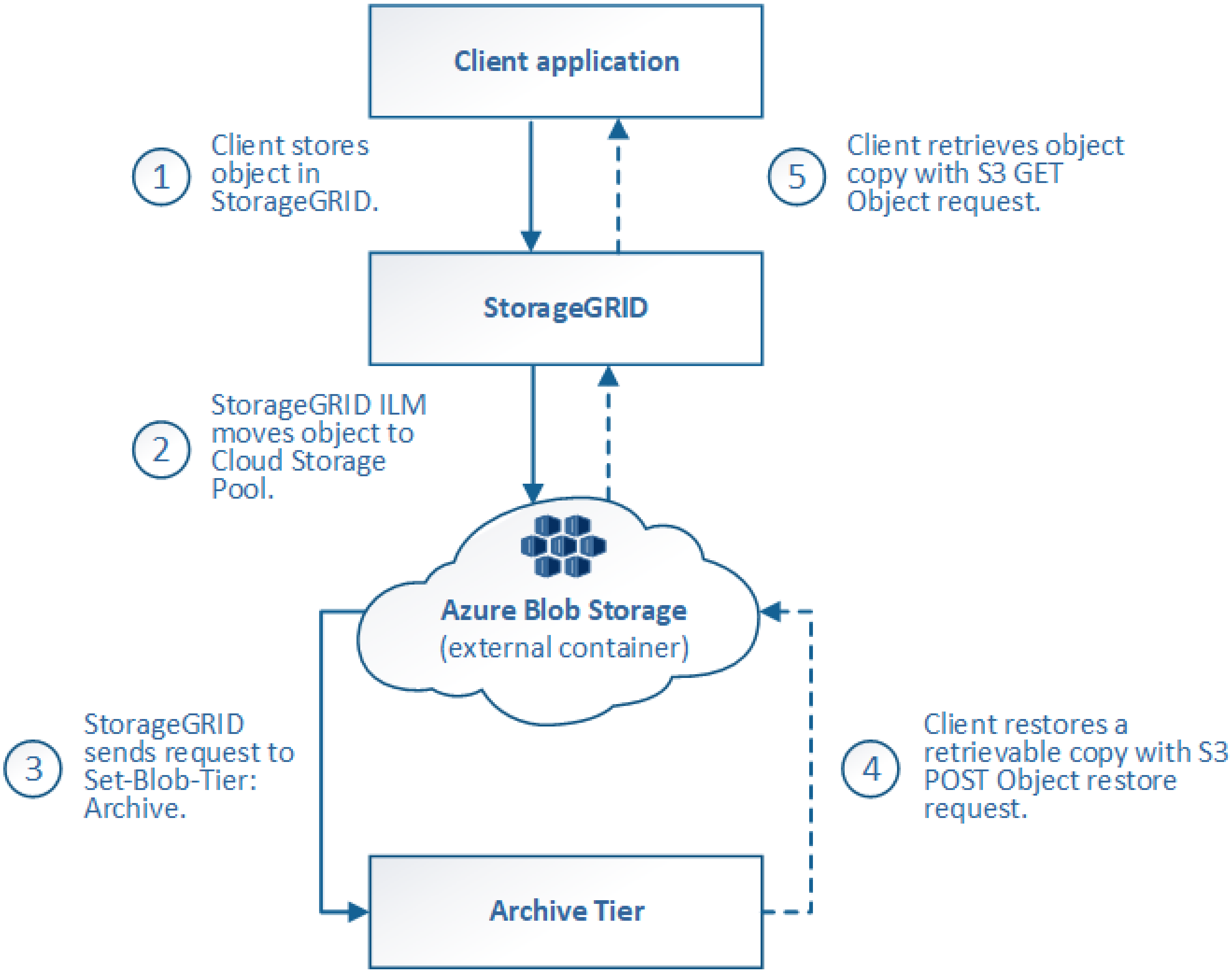
   > If one or more copies of the object also exist on Storage Nodes within StorageGRID, there is no need to restore the object from Glacier by issuing a POST Object restore request. Instead, the local copy can be retrieved directly, using a GET Object request.

5. **Object retrieved**

   Once an object has been restored, the client application can issue a GET Object request to retrieve the restored object.

### Azure: Lifecycle of a Cloud Storage Pool object

The figure shows the lifecycle stages of an object that is stored in an Azure Cloud Storage Pool.


Client application
1
Client stores object in StorageGRID.
5
Client retrieves object copy with S3 GET Object request.
StorageGRID
2
StorageGRID ILM moves object to Cloud Storage Pool.
Azure Blob Storage
(external container)
3
StorageGRID sends request to Set-Blob-Tier: Archive.
4
Client restores a retrievable copy with S3 POST Object restore request.
Archive Tier

1. **Object stored in StorageGRID**

   To start the lifecycle, a client application stores an object in StorageGRID.

2. **Object moved to Azure Cloud Storage Pool**

   When the object is matched by an ILM rule that uses an Azure Cloud Storage Pool as its placement location, StorageGRID moves the object to the external Azure Blob storage container specified by the Cloud Storage Pool

   > ⓘ Don’t use Cloud Storage Pools for objects that have been ingested by Swift clients. Swift does not support POST Object restore requests, so StorageGRID will not be able to retrieve any Swift objects that have been transitioned to the Azure Blob storage Archive tier. Issuing a Swift GET object request to retrieve these objects will fail (403 Forbidden).

3. **Object transitioned to Archive tier (non-retrievable state)**

   Immediately after moving the object to the Azure Cloud Storage Pool, StorageGRID automatically transitions the object to the Azure Blob storage Archive tier.

4. **Object restored from Archive tier**

   If an object has been transitioned to the Archive tier, the client application can issue an S3 POST Object restore request to restore a retrievable copy to the Azure Cloud Storage Pool.

When StorageGRID receives the POST Object Restore, it temporarily transitions the object to the Azure Blob storage Cool tier. As soon as the expiration date in the POST Object restore request is reached, StorageGRID transitions the object back to the Archive tier.

> ⓘ If one or more copies of the object also exist on Storage Nodes within StorageGRID, there is no need to restore the object from the Archive access tier by issuing a POST Object restore request. Instead, the local copy can be retrieved directly, using a GET Object request.

5. **Object retrieved**

   Once an object has been restored to the Azure Cloud Storage Pool, the client application can issue a GET Object request to retrieve the restored object.

**Related information**

Use S3 REST API

## When to use Cloud Storage Pools

Using Cloud Storage Pools, you can back up or tier data to an external location. Additionally, you can back up or tier data to more than one cloud.

### Back up StorageGRID data to external location

You can use a Cloud Storage Pool to back up StorageGRID objects to an external location.

If the copies in StorageGRID are inaccessible, the object data in the Cloud Storage Pool can be used to serve client requests. However, you might need to issue S3 POST Object restore request to access the backup object copy in the Cloud Storage Pool.

The object data in a Cloud Storage Pool can also be used to recover data lost from StorageGRID because of a storage volume or Storage Node failure. If the only remaining copy of an object is in a Cloud Storage Pool, StorageGRID temporarily restores the object and creates a new copy on the recovered Storage Node.

To implement a backup solution:

1. Create a single Cloud Storage Pool.
2. Configure an ILM rule that simultaneously stores object copies on Storage Nodes (as replicated or erasure-coded copies) and a single object copy in the Cloud Storage Pool.
3. Add the rule to your ILM policy. Then, simulate and activate the policy.

### Tier data from StorageGRID to external location

You can use a Cloud Storage Pool to store objects outside of the StorageGRID system. For example, suppose you have a large number of objects that you need to retain, but you expect to access those objects rarely, if ever. You can use a Cloud Storage Pool to tier the objects to lower-cost storage and to free up space in StorageGRID.

To implement a tiering solution:

1. Create a single Cloud Storage Pool.

2. Configure an ILM rule that moves rarely used objects from Storage Nodes to the Cloud Storage Pool.
3. Add the rule to your ILM policy. Then, simulate and activate the policy.

### Maintain multiple cloud endpoints

You can configure multiple Cloud Storage Pool endpoints if you want to tier or back up object data to more than one cloud. The filters in your ILM rules let you specify which objects are stored in each Cloud Storage Pool. For example, you might want to store objects from some tenants or buckets in Amazon S3 Glacier and objects from other tenants or buckets in Azure Blob storage. Or, you might want to move data between Amazon S3 Glacier and Azure Blob storage.

> When using multiple Cloud Storage Pool endpoints, keep in mind that an object can be stored in only one Cloud Storage Pool at a time.

To implement multiple cloud endpoints:

1. Create up to 10 Cloud Storage Pools.
2. Configure ILM rules to store the appropriate object data at the appropriate time in each Cloud Storage Pool. For example, store objects from bucket A in Cloud Storage Pool A, and store objects from bucket B in Cloud Storage Pool B. Or, store objects in Cloud Storage Pool A for some amount of time and then move them to Cloud Storage Pool B.
3. Add the rules to your ILM policy. Then, simulate and activate the policy.

## Considerations for Cloud Storage Pools

If you plan to use a Cloud Storage Pool to move objects out of the StorageGRID system, you must review the considerations for configuring and using Cloud Storage Pools.

### General considerations

- In general, cloud archival storage, such as Amazon S3 Glacier or Azure Blob storage, is an inexpensive place to store object data. However, the costs to retrieve data from cloud archival storage are relatively high. To achieve the lowest overall cost, you must consider when and how often you will access the objects in the Cloud Storage Pool. Using a Cloud Storage Pool is recommended only for content that you expect to access infrequently.
- Don’t use Cloud Storage Pools for objects that have been ingested by Swift clients. Swift does not support POST Object restore requests, so StorageGRID will not be able to retrieve any Swift objects that have been transitioned to S3 Glacier storage or the Azure Blob storage Archive tier. Issuing a Swift GET object request to retrieve these objects will fail (403 Forbidden).
- Using Cloud Storage Pools with FabricPool is not supported because of the added latency to retrieve an object from the Cloud Storage Pool target.
- Objects with S3 Object Lock enabled can’t be placed in Cloud Storage Pools.
- If the destination S3 bucket for a Cloud Storage Pool has S3 Object Lock enabled, the attempt to configure bucket replication (PutBucketReplication) will fail with an AccessDenied error.

### Considerations for the ports used for Cloud Storage Pools

To ensure that the ILM rules can move objects to and from the specified Cloud Storage Pool, you must configure the network or networks that contain your system’s Storage Nodes. You must ensure that the following ports can communicate with the Cloud Storage Pool.

By default, Cloud Storage Pools use the following ports:

- **80**: For endpoint URIs that begin with http
- **443**: For endpoint URIs that begin with https

You can specify a different port when you create or edit a Cloud Storage Pool.

If you use a non-transparent proxy server, you must also configure a Storage proxy to allow messages to be sent to external endpoints, such as an endpoint on the internet.

### Considerations for costs

Access to storage in the cloud using a Cloud Storage Pool requires network connectivity to the cloud. You must consider the cost of the network infrastructure you will use to access the cloud and provision it appropriately, based on the amount of data you expect to move between StorageGRID and the cloud using the Cloud Storage Pool.

When StorageGRID connects to the external Cloud Storage Pool endpoint, it issues various requests to monitor connectivity and to ensure it can perform the required operations. While some additional costs will be associated with these requests, the cost of monitoring a Cloud Storage Pool should only be a small fraction of the overall cost of storing objects in S3 or Azure.

More significant costs might be incurred if you need to move objects from an external Cloud Storage Pool endpoint back to StorageGRID. Objects might be moved back to StorageGRID in either of these cases:

- The only copy of the object is in a Cloud Storage Pool and you decide to store the object in StorageGRID instead. In this case, you reconfigure your ILM rules and policy. When ILM evaluation occurs, StorageGRID issues multiple requests to retrieve the object from the Cloud Storage Pool. StorageGRID then creates the specified number of replicated or erasure-coded copies locally. After the object is moved back to StorageGRID, the copy in the Cloud Storage Pool is deleted.
- Objects are lost because of Storage Node failure. If the only remaining copy of an object is in a Cloud Storage Pool, StorageGRID temporarily restores the object and creates a new copy on the recovered Storage Node.

> ⓘ When objects are moved back to StorageGRID from a Cloud Storage Pool, StorageGRID issues multiple requests to the Cloud Storage Pool endpoint for each object. Before moving large numbers of objects, contact technical support for help in estimating the time frame and associated costs.

### S3: Permissions required for the Cloud Storage Pool bucket

The bucket policy for the external S3 bucket used for a Cloud Storage Pool must grant StorageGRID permission to move an object to the bucket, get an object’s status, restore an object from Glacier storage when required, and more. Ideally, StorageGRID should have full-control access to the bucket (`s3:*`); however, if this is not possible, the bucket policy must grant the following S3 permissions to StorageGRID:

- `s3:AbortMultipartUpload`
- `s3:DeleteObject`
- `s3:GetObject`
- `s3:ListBucket`
- `s3:ListBucketMultipartUploads`

- `s3:ListMultipartUploadParts`
- `s3:PutObject`
- `s3:RestoreObject`

### S3: Considerations for the external bucket’s lifecycle

The movement of objects between StorageGRID and the external S3 bucket specified in the Cloud Storage Pool is controlled by ILM rules and the active ILM policy in StorageGRID. In contrast, the transition of objects from the external S3 bucket specified in the Cloud Storage Pool to Amazon S3 Glacier or S3 Glacier Deep Archive (or to a storage solution that implements the Glacier storage class) is controlled by that bucket’s lifecycle configuration.

If you want to transition objects from the Cloud Storage Pool, you must create the appropriate lifecycle configuration on the external S3 bucket, and you must use a storage solution that implements the Glacier storage class and supports the S3 POST Object restore API.

For example, suppose you want all objects that are moved from StorageGRID to the Cloud Storage Pool to be transitioned to Amazon S3 Glacier storage immediately. You would create a lifecycle configuration on the external S3 bucket that specifies a single action (**Transition**) as follows:

```
<LifecycleConfiguration>
  <Rule>
    <ID>Transition Rule</ID>
    <Filter>
       <Prefix></Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

This rule would transition all bucket objects to Amazon S3 Glacier on the day they were created (that is, on the day they were moved from StorageGRID to the Cloud Storage Pool).

> When configuring the external bucket’s lifecycle, never use **Expiration** actions to define when objects expire. Expiration actions cause the external storage system to delete expired objects. If you later attempt to access an expired object from StorageGRID, the deleted object will not be found.

If you want to transition objects in the Cloud Storage Pool to S3 Glacier Deep Archive (instead of to Amazon S3 Glacier), specify `<StorageClass>DEEP_ARCHIVE</StorageClass>` in the bucket lifecycle. However, be aware that you can’t use the `Expedited` tier to restore objects from S3 Glacier Deep Archive.

#### Azure: Considerations for Access tier

When you configure an Azure storage account, you can set the default Access tier to Hot or Cool. When creating a storage account for use with a Cloud Storage Pool, you should use the Hot tier as the default tier. Even though StorageGRID immediately sets the tier to Archive when it moves objects to the Cloud Storage Pool, using a default setting of Hot ensures that you will not be charged an early deletion fee for objects removed from the Cool tier before the 30-day minimum.

#### Azure: Lifecycle management not supported

Don’t use Azure Blob storage lifecycle management for the container used with a Cloud Storage Pool. The lifecycle operations might interfere with Cloud Storage Pool operations.

**Related information**

- Create a Cloud Storage Pool

## Compare Cloud Storage Pools and CloudMirror replication

As you begin using Cloud Storage Pools, it might be helpful to understand the similarities and differences between Cloud Storage Pools and the StorageGRID CloudMirror replication service.

| | Cloud Storage Pool | CloudMirror replication service |
|---|---|---|
| What is the primary purpose? | Acts as an archive target. The object copy in the Cloud Storage Pool can be the only copy of the object, or it can be an additional copy. That is, instead of keeping two copies onsite, you can keep one copy within StorageGRID and send a copy to the Cloud Storage Pool. | Enables a tenant to automatically replicate objects from a bucket in StorageGRID (source) to an external S3 bucket (destination). Creates an independent copy of an object in an independent S3 infrastructure. |
| How is it set up? | Defined in the same way as storage pools, using the Grid Manager or the Grid Management API. Can be selected as the placement location in an ILM rule. While a storage pool consists of a group of Storage Nodes, a Cloud Storage Pool is defined using a remote S3 or Azure endpoint (IP address, credentials, and so on). | A tenant user configures CloudMirror replication by defining a CloudMirror endpoint (IP address, credentials, and so on) using the Tenant Manager or the S3 API. After the CloudMirror endpoint is set up, any bucket owned by that tenant account can be configured to point to the CloudMirror endpoint. |
| Who is responsible for setting it up? | Typically, a grid administrator | Typically, a tenant user |
| What is the destination? | • Any compatible S3 infrastructure (including Amazon S3)<br>• Azure Blob Archive tier<br>• Google Cloud Platform (GCP) | • Any compatible S3 infrastructure (including Amazon S3)<br>• Google Cloud Platform (GCP) |

| | Cloud Storage Pool | CloudMirror replication service |
|---|---|---|
| What causes objects to be moved to the destination? | One or more ILM rules in the active ILM policy. The ILM rules define which objects StorageGRID moves to the Cloud Storage Pool and when the objects are moved. | The act of ingesting a new object into a source bucket that has been configured with a CloudMirror endpoint. Objects that existed in the source bucket before the bucket was configured with the CloudMirror endpoint aren't replicated, unless they are modified. |
| How are objects retrieved? | Applications must make requests to StorageGRID to retrieve objects that have been moved to a Cloud Storage Pool. If the only copy of an object has been transitioned to archival storage, StorageGRID manages the process of restoring the object so it can be retrieved. | Because the mirrored copy in the destination bucket is an independent copy, applications can retrieve the object by making requests either to StorageGRID or to the S3 destination. For example, suppose you use CloudMirror replication to mirror objects to a partner organization. The partner can use its own applications to read or update objects directly from the S3 destination. Using StorageGRID is not required. |
| Can you read from the destination directly? | No. Objects moved to a Cloud Storage Pool are managed by StorageGRID. Read requests must be directed to StorageGRID (and StorageGRID will be responsible for retrieval from Cloud Storage Pool). | Yes, because the mirrored copy is an independent copy. |
| What happens if an object is deleted from the source? | The object is also deleted from the Cloud Storage Pool. | The delete action is not replicated. A deleted object no longer exists in the StorageGRID bucket, but it continues to exist in the destination bucket. Similarly, objects in the destination bucket can be deleted without affecting the source. |
| How do you access objects after a disaster (StorageGRID system not operational)? | Failed StorageGRID nodes must be recovered. During this process, copies of replicated objects might be restored using the copies in the Cloud Storage Pool. | The object copies in the CloudMirror destination are independent of StorageGRID, so they can be accessed directly before the StorageGRID nodes are recovered. |

## Create a Cloud Storage Pool

A Cloud Storage Pool specifies a single external Amazon S3 bucket or other S3-compatible provider, or Azure Blob storage container.

When you create a Cloud Storage Pool, you specify the name and location of the external bucket or container that StorageGRID will use to store objects, the cloud provider type (Amazon S3/GCP or Azure Blob storage), and the information StorageGRID needs to access the external bucket or container.

StorageGRID validates the Cloud Storage Pool as soon as you save it, so you must ensure that the bucket or container specified in the Cloud Storage Pool exists and is reachable.

**Before you begin**

- You are signed in to the Grid Manager using a supported web browser.
- You have the required access permissions.
- You have reviewed the considerations for Cloud Storage Pools.
- The external bucket or container referenced by the Cloud Storage Pool already exists, and you know its name and location.
- To access the bucket or container, you have the following information for the authentication type you will choose:

**S3 access key**

*For the external S3 bucket*

- The access key ID for the account that owns the external bucket.
- The associated secret access key.

Alternatively, you can specify Anonymous for the authentication type.

**C2S access portal**

*For Commercial Cloud Services (C2S) S3 service*

You have the following:

- Complete URL that StorageGRID will use to obtain temporary credentials from the C2S access portal (CAP) server, including all the required and optional API parameters assigned to your C2S account.
- Server CA certificate issued by an appropriate Government Certificate Authority (CA). StorageGRID uses this certificate to verify the identity of the CAP server. The server CA certificate must use PEM encoding.
- Client certificate issued by an appropriate Government Certificate Authority (CA). StorageGRID uses this certificate to identity itself to the CAP server. The client certificate must use PEM encoding and must have been granted access to your C2S account.
- PEM-encoded private key for the client certificate.
- Passphrase for decrypting the private key for the client certificate, if it is encrypted.

> ⓘ If the client certificate will be encrypted, use the traditional format for the encryption. PKCS #8 encrypted format is not supported.

**Azure Blob storage**

*For the external container*

- Uniform Resource Identifier (URI) used to access the Blob Storage container.
- Name of the storage account and the account key. You can use the Azure portal to find these values.

**Steps**

1. Select **ILM** > **Storage pools** > **Cloud Storage Pools**.
2. Select **Create**, then enter the following information:

| Field | Description |
|---|---|
| Cloud Storage Pool name | A name that briefly describes the Cloud Storage Pool and its purpose. Use a name that will be easy to identify when you configure ILM rules. |
| Provider type | Which cloud provider you will use for this Cloud Storage Pool:<br>• **Amazon S3/GCP**: Select this option for an Amazon S3, Commercial Cloud Services (C2S) S3, Google Cloud Platform (GCP), or other S3-compatible provider.<br>• **Azure Blob Storage** |
| Bucket or container | The name of the external S3 bucket or Azure container. You can't change this value after the Cloud Storage Pool is saved. |

3. Based on your Provider type selection, enter the Service endpoint information.

**Amazon S3/GCP**

a. For the protocol, select either HTTPS or HTTP.

> ⓘ Don’t use HTTP connections for sensitive data.

b. Enter the hostname. Example:

`s3-`*`aws-region`*`.amazonaws.com`

c. Select the URL style:

| Option | Description |
|---|---|
| Auto-detect | Attempt to automatically detect which URL style to use, based on the information provided. For example, if you specify an IP address, StorageGRID will use a path-style URL. Select this option only if you don’t know which specific style to use. |
| Virtual-hosted-style | Use a virtual-hosted-style URL to access the bucket. Virtual-hosted-style URLs include the bucket name as part of the domain name. Example: `https://bucket-name.s3.company.com/key-name` |
| Path-style | Use a path-style URL to access the bucket. Path-style URLs include the bucket name at the end. Example: `https://s3.company.com/bucket-name/key-name`<br>**Note:** The path-style URL option is not recommended and will be deprecated in a future release of StorageGRID. |

d. Optionally, enter the port number, or use the default port: 443 for HTTPS or 80 for HTTP.

**Azure Blob Storage**

a. Using one of the following formats, enter the URI for the service endpoint.

- `https://host:port`
- `http://host:port`

Example: `https://`*`myaccount`*`.blob.core.windows.net:443`

If you don’t specify a port, by default port 443 is used for HTTPS and port 80 is used for HTTP.

4. Select **Continue**. Then select the authentication type and enter the required information for the Cloud Storage Pool endpoint:

**Access key**

*For Amazon S3/GCP provider type only*

a. For **Access key ID**, enter the access key ID for the account that owns the external bucket.
b. For **Secret access key**, enter the secret access key.

**CAP (C2S access portal)**

*For Commercial Cloud Services (C2S) S3 service*

a. For **Temporary credentials URL**, enter the complete URL that StorageGRID will use to obtain temporary credentials from the CAP server, including all the required and optional API parameters assigned to your C2S account.
b. For **Server CA certificate**, select **Browse**, and upload the PEM-encoded CA certificate that StorageGRID will use to verify the CAP server.
c. For **Client certificate**, select **Browse**, and upload the PEM-encoded certificate that StorageGRID will use to identify itself to the CAP server.
d. For **Client private key**, select **Browse**, and upload the PEM-encoded private key for the client certificate.
e. If the client private key is encrypted, enter the passphrase for decrypting the client private key. Otherwise, leave the **Client private key passphrase** field blank.

**Azure Blob Storage**

a. For **Account name**, enter the name of the Blob storage account that owns the external service container.
b. For **Account key**, enter the secret key for the Blob storage account.

**Anonymous**

No additional information is required.

5. Select **Continue**. Then choose the type of server verification you want to use:

| Option | Description |
|---|---|
| Use root CA certificates in Storage Node OS | Use the Grid CA certificates installed on the operating system to secure connections. |
| Use custom CA certificate | Use a custom CA certificate. Select **Browse**, and upload the PEM-encoded certificate. |
| Do not verify certificate | The certificate used for the TLS connection is not verified. |

6. Select **Save**.

When you save a Cloud Storage Pool, StorageGRID does the following:

- Validates that the bucket or container and the service endpoint exist and that they can be reached using the credentials that you specified.

   - Writes a marker file to the bucket or container to identify it as a Cloud Storage Pool. Never remove this file, which is named `x-ntap-sgws-cloud-pool-uuid`.

     If Cloud Storage Pool validation fails, you receive an error message that explains why validation failed. For example, an error might be reported if there is a certificate error or if the bucket or container you specified does not already exist.

7. If an error occurs, see the instructions for troubleshooting Cloud Storage Pools, resolve any issues, and then try saving the Cloud Storage Pool again.

## Edit a Cloud Storage Pool

You can edit a Cloud Storage Pool to change its name, service endpoint, or other details; however, you can't change the S3 bucket or Azure container for a Cloud Storage Pool.

**Before you begin**

- You are signed in to the Grid Manager using a supported web browser.
- You have specific access permissions.
- You have reviewed the considerations for Cloud Storage Pools.

**Steps**

1. Select **ILM** > **Storage pools** > **Cloud Storage Pools**.

   The Cloud Storage Pools table lists the existing Cloud Storage Pools.

2. Select the checkbox for the Cloud Storage Pool you want to edit.
3. Select **Actions** > **Edit**.
4. As required, change the display name, service endpoint, authentication credentials, or certificate validation method.

   > ⓘ You can't change the provider type or the S3 bucket or Azure container for a Cloud Storage Pool.

   If you previously uploaded a server or client certificate, you can select **Certificate details** to review the certificate that is currently in use.

5. Select **Save**.

   When you save a Cloud Storage Pool, StorageGRID validates that the bucket or container and the service endpoint exist, and that they can be reached using the credentials that you specified.

   If Cloud Storage Pool validation fails, an error message is displayed. For example, an error might be reported if there is a certificate error.

   See the instructions for troubleshooting Cloud Storage Pools, resolve the issue, and then try saving the Cloud Storage Pool again.

## Remove a Cloud Storage Pool

You can remove a Cloud Storage Pool if it not used in an ILM rule and it does not contain object data.

**Before you begin**

- You are signed in to the Grid Manager using a supported web browser.
- You have the required access permissions.

### If needed, use ILM to move object data

If the Cloud Storage Pool you want to remove contains object data, you must use ILM to move the data to a different location. For example, you can move the data to Storage Nodes on your grid or to a different Cloud Storage Pool.

**Steps**

1. Select **ILM** > **Storage pools** > **Cloud Storage Pools**.
2. Look at the ILM usage column in the table to determine whether you can remove the Cloud Storage Pool.

   You can't remove a Cloud Storage Pool if it is being used in an ILM rule or in an erasure coding profile.

3. If the Cloud Storage Pool is being used, select ***cloud storage pool name*** > **ILM usage**.
4. Clone each ILM rule that currently places objects in the Cloud Storage Pool you want to remove.
5. Determine where you want to move the existing objects managed by each rule you cloned.

   You can use one or more storage pools or a different Cloud Storage Pool.

6. Edit each of the rules you cloned.

   For Step 2 of the Create ILM rule wizard, select the new location from the **copies at** field.

7. Create a new proposed ILM policy and replace each of the old rules with a cloned rule.
8. Activate the new policy.
9. Wait for ILM to remove objects from the Cloud Storage Pool and place them in the new location.

### Delete Cloud Storage Pool

When the Cloud Storage Pool is empty and not used in any ILM rules, you can delete it.

**Before you begin**

- You have removed any ILM rules that might have used the pool.
- You have confirmed that the S3 bucket or Azure container does not contain any objects.

  An error occurs if you attempt to remove a Cloud Storage Pool if it contains objects. See Troubleshoot Cloud Storage Pools.

  > ⓘ When you create a Cloud Storage Pool, StorageGRID writes a marker file to the bucket or container to identify it as a Cloud Storage Pool. Don't remove this file, which is named `x-ntap-sgws-cloud-pool-uuid`.

**Steps**

1. Select **ILM** > **Storage pools** > **Cloud Storage Pools**.

2. If the ILM usage column indicates that Cloud Storage Pool is not being used, select the checkbox.
3. Select **Actions** > **Remove**.
4. Select **OK**.

## Troubleshoot Cloud Storage Pools

Use these troubleshooting steps to help resolve errors you might encounter when creating, editing, or deleting a Cloud Storage Pool.

### Determine if an error has occurred

StorageGRID performs a simple health check on every Cloud Storage Pool once a minute to ensure that the Cloud Storage Pool can be accessed and that it is functioning correctly. If the health check detects an issue, a message is shown in the Last error column of the Cloud Storage Pools table on the Storage pools page.

The table shows the most recent error detected for each Cloud Storage Pool and indicates how long ago the error occurred.

In addition, a **Cloud Storage Pool connectivity error** alert is triggered if the health check detects that one or more new Cloud Storage Pool errors have occurred within the past 5 minutes. If you receive an email notification for this alert, go to the Storage pools page (select **ILM** > **Storage pools**), review the error messages in the Last error column, and refer to the troubleshooting guidelines below.

### Check if an error has been resolved

After resolving any underlying issues, you can determine if the error has been resolved. From the Cloud Storage Pool page, select the endpoint, and select **Clear error**. A confirmation message indicates that StorageGRID has cleared the error for the Cloud Storage Pool.

If the underlying problem has been resolved, the error message is no longer displayed. However, if the underlying problem has not been fixed (or if a different error is encountered), the error message will be shown in the Last error column within a few minutes.

### Error: This Cloud Storage Pool contains unexpected content

You might encounter this error when you try to create, edit, or delete a Cloud Storage Pool. This error occurs if the bucket or container includes the `x-ntap-sgws-cloud-pool-uuid` marker file, but that file does not have the expected UUID.

Typically, you will only see this error if you are creating a new Cloud Storage Pool and another instance of StorageGRID is already using the same Cloud Storage Pool.

Try these steps to correct the issue:

- Check to make sure that no one in your organization is also using this Cloud Storage Pool.
- Delete the `x-ntap-sgws-cloud-pool-uuid` file and try configuring the Cloud Storage Pool again.

### Error: Could not create or update Cloud Storage Pool. Error from endpoint

You might encounter this error when you try to create or edit a Cloud Storage Pool. This error indicates that some kind of connectivity or configuration issue is preventing StorageGRID from writing to the Cloud Storage Pool.

To correct the issue, review the error message from the endpoint.

- If the error message contains `Get url: EOF`, check that the service endpoint used for the Cloud Storage Pool does not use HTTP for a container or bucket that requires HTTPS.
- If the error message contains `Get url: net/http: request canceled while waiting for connection`, verify that the network configuration allows Storage Nodes to access the service endpoint used for the Cloud Storage Pool.
- For all other endpoint error messages, try one or more of the following:
  - Create an external container or bucket with the same name you entered for the Cloud Storage Pool, and try to save the new Cloud Storage Pool again.
  - Correct the container or bucket name you specified for the Cloud Storage Pool, and try to save the new Cloud Storage Pool again.

#### Error: Failed to parse CA certificate

You might encounter this error when you try to create or edit a Cloud Storage Pool. The error occurs if StorageGRID could not parse the certificate you entered when configuring the Cloud Storage Pool.

To correct the issue, check the CA certificate you provided for issues.

#### Error: A Cloud Storage Pool with this ID was not found

You might encounter this error when you try to edit or delete a Cloud Storage Pool. This error occurs if the endpoint returns a 404 response, which can mean either of the following:

- The credentials used for the Cloud Storage Pool don't have read permission for the bucket.
- The bucket used for the Cloud Storage Pool does not include the `x-ntap-sgws-cloud-pool-uuid` marker file.

Try one or more of these steps to correct the issue:

- Check that the user associated with the configured Access Key has the requisite permissions.
- Edit the Cloud Storage Pool with credentials that have the requisite permissions.
- If the permissions are correct, contact support.

#### Error: Could not check the content of the Cloud Storage Pool. Error from endpoint

You might encounter this error when you try to delete a Cloud Storage Pool. This error indicates that some kind of connectivity or configuration issue is preventing StorageGRID from reading the contents of the Cloud Storage Pool bucket.

To correct the issue, review the error message from the endpoint.

#### Error: Objects have already been placed in this bucket

You might encounter this error when you try to delete a Cloud Storage Pool. You can't delete a Cloud Storage Pool if it contains data that was moved there by ILM, data that was in the bucket before you configured the Cloud Storage Pool, or data that was put in the bucket by some other source after the Cloud Storage Pool was created.

Try one or more of these steps to correct the issue:

- Follow the instructions for moving objects back to StorageGRID in “Lifecycle of a Cloud Storage Pool object.”
- If you are certain the remaining objects were not placed in the Cloud Storage Pool by ILM, manually delete the objects from the bucket.

> ⓘ Never manually delete objects from a Cloud Storage Pool that might have been placed there by ILM. If you later attempt to access a manually deleted object from StorageGRID, the deleted object will not be found.

**Error: Proxy encountered an external error while trying to reach the Cloud Storage Pool**

You might encounter this error if you have configured a non-transparent Storage proxy between Storage Nodes and the external S3 endpoint used for the Cloud Storage Pool. This error occurs if the external proxy server can’t reach the Cloud Storage Pool endpoint. For example, the DNS server might not be able to resolve the hostname or there might be an external networking issue.

Try one or more of these steps to correct the issue:

- Check the settings for the Cloud Storage Pool (**ILM** > **Storage pools**).
- Check the networking configuration of the Storage proxy server.

**Related information**

Lifecycle of a Cloud Storage Pool object

# Manage erasure coding profiles

You can rename an erasure coding profile if needed. You can deactivate an erasure coding profile if it is not currently used in any ILM rules.

## Rename an erasure coding profile

You might want to rename an erasure coding profile to make it more obvious what the profile does.

**Before you begin**

- You are signed in to the Grid Manager using a supported web browser.
- You have the required access permissions.

**Steps**

1. Select **ILM** > **Erasure coding**.
2. Select the profile you want to rename.
3. Select **Rename**.
4. Enter a unique name for the erasure coding profile.

   The erasure coding profile name is appended to the storage pool name in the placement instruction for an ILM rule.

   > ⓘ Erasure coding profile names must be unique. A validation error occurs if you use the name of an existing profile, even if that profile has been deactivated.

5. Select **Save**.

## Deactivate an erasure coding profile

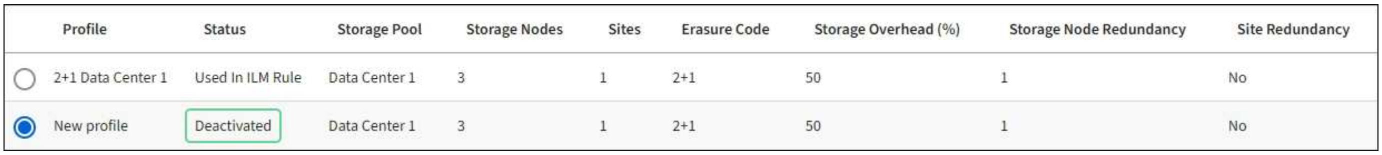
You can deactivate an erasure coding profile if you no longer plan to use it and if the profile is not currently used in any ILM rules.

### Before you begin

- You are signed in to the Grid Manager using a supported web browser.
- You have the required access permissions.
- You have confirmed that no erasure coded data repair operations or decommission procedures are in process. An error message is returned if you attempt to deactivate an erasure coding profile while either of these operations are in progress.

### About this task

When you deactivate an erasure coding profile, the profile still appears on the Erasure Coding Profiles page, but its status is **Deactivated**.



| | Profile | Status | Storage Pool | Storage Nodes | Sites | Erasure Code | Storage Overhead (%) | Storage Node Redundancy | Site Redundancy |
|---|---|---|---|---|---|---|---|---|---|
| ○ | 2+1 Data Center 1 | Used In ILM Rule | Data Center 1 | 3 | 1 | 2+1 | 50 | 1 | No |
| ● | New profile | Deactivated | Data Center 1 | 3 | 1 | 2+1 | 50 | 1 | No |

You can no longer use an erasure coding profile that has been deactivated. A deactivated profile is not shown when you create the placement instructions for an ILM rule. You can’t reactivate a deactivated profile.

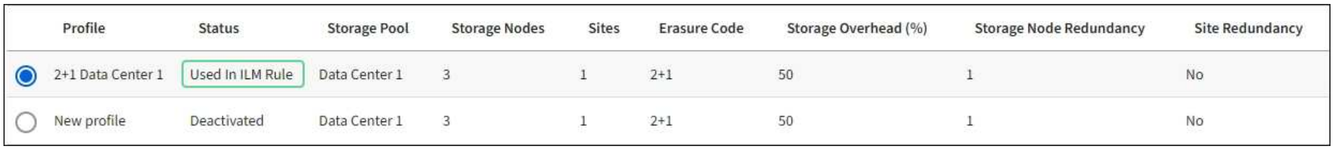
StorageGRID prevents you from deactivating an erasure coding profile if either of the following is true:

- The erasure coding profile is currently used in an ILM rule.
- The erasure coding profile is no longer used in any ILM rules, but object data and parity fragments for the profile still exist.

### Steps

1. Select **ILM** > **Erasure Coding**.
2. Review the **Status** column to confirm that the erasure coding profile you want to deactivate is not used in any ILM rules.

   You can’t deactivate an erasure coding profile if it is used in any ILM rule. In the example, the **2+1 Data Center 1** profile is used in at least one ILM rule.



| | Profile | Status | Storage Pool | Storage Nodes | Sites | Erasure Code | Storage Overhead (%) | Storage Node Redundancy | Site Redundancy |
|---|---|---|---|---|---|---|---|---|---|
| ● | 2+1 Data Center 1 | Used In ILM Rule | Data Center 1 | 3 | 1 | 2+1 | 50 | 1 | No |
| ○ | New profile | Deactivated | Data Center 1 | 3 | 1 | 2+1 | 50 | 1 | No |

3. If the profile is used in an ILM rule, follow these steps:
   a. Select **ILM** > **Rules**.
   b. Select each rule and review the retention diagram to determine if the rule uses the erasure coding profile you want to deactivate.
   c. If the ILM rule uses the erasure coding profile you want to deactivate, determine if the rule is used in

either the active ILM policy or a proposed policy.

d. Complete the additional steps in the table, based on where the erasure coding profile is used.

| **Where has the profile been used?** | **Additional steps to perform before deactivating the profile** | **Refer to these additional instructions** |
|---|---|---|
| Never used in any ILM rule | No additional steps required. Continue with this procedure. | *None* |
| In an ILM rule that has never been used in any ILM policy | 1. Edit or delete all affected ILM rules. If you edit the rule, remove all placements that use the erasure coding profile.<br>2. Continue with this procedure. | Work with ILM rules and ILM policies |
| In an ILM rule that is currently in the active ILM policy | 1. Clone the active policy.<br>2. Remove the ILM rule that uses the erasure coding profile.<br>3. Add one or more new ILM rules to ensure objects are protected.<br>4. Save, simulate, and activate the new policy.<br>5. Wait for the new policy to be applied and for existing objects to be moved to new locations based on the new rules you added.<br>**Note:** Depending on the number of objects and the size of your StorageGRID system, it might take weeks or even months for ILM operations to move the objects to new locations, based on the new ILM rules.<br>While you can safely attempt to deactivate an erasure coding profile while it is still associated with data, the deactivation operation will fail. An error message will inform you if the profile is not yet ready to be deactivated.<br>6. Edit or delete the rule you removed from the policy. If you edit the rule, remove all placements that use the erasure coding profile.<br>7. Continue with this procedure. | Create an ILM policy<br>Work with ILM rules and ILM policies |

| Where has the profile been used? | Additional steps to perform before deactivating the profile | Refer to these additional instructions |
|---|---|---|
| In an ILM rule that is currently in a proposed ILM policy | 1. Edit the proposed policy.<br>2. Remove the ILM rule that uses the erasure coding profile.<br>3. Add one or more new ILM rules to ensure all objects are protected.<br>4. Save the proposed policy.<br>5. Edit or delete the rule you removed from the policy. If you edit the rule, remove all placements that use the erasure coding profile.<br>6. Continue with this procedure. | Create an ILM policy<br>Work with ILM rules and ILM policies |
| In an ILM rule that is in a historical ILM policy | 1. Edit or delete the rule. If you edit the rule, remove all placements that use the erasure coding profile. (The rule will now appear as a historical rule in the historical policy.)<br>2. Continue with this procedure. | Work with ILM rules and ILM policies |

e. Refresh the Erasure Coding Profiles page to ensure that the profile is not used in an ILM rule.

4. If the profile is not used in an ILM rule, select the radio button and select **Deactivate**.

   The Deactivate EC Profile dialog box appears.

5. If you are sure you want to deactivate the profile, select **Deactivate**.
   - If StorageGRID is able to deactivate the erasure coding profile, its status is **Deactivated**. You can no longer select this profile for any ILM rule.
   - If StorageGRID is not able to deactivate the profile, an error message appears. For example, an error message appears if object data is still associated with this profile. You might need to wait several weeks before trying the deactivation process again.

# Configure regions (optional and S3 only)

ILM rules can filter objects based on the regions where S3 buckets are created, allowing you to store objects from different regions in different storage locations.

If you want to use an S3 bucket region as a filter in a rule, you must first create the regions that can be used by the buckets in your system.

> You can’t change the region for a bucket after the bucket has been created.

**Before you begin**

- You are signed in to the Grid Manager using a supported web browser.

- You have specific access permissions.

### About this task

When creating an S3 bucket, you can specify that the bucket be created in a specific region. Specifying a region allows the bucket to be geographically close to its users, which can help optimize latency, minimize costs, and address regulatory requirements.

When you create an ILM rule, you might want to use the region associated with an S3 bucket as an advanced filter. For example, you can design a rule that applies only to objects in S3 buckets created in the us-west-2 region. You can then specify that copies of those objects be placed on Storage Nodes at a data center site within that region to optimize latency.

When configuring regions, follow these guidelines:

- By default, all buckets are considered to belong to the us-east-1 region.
- You must create the regions using the Grid Manager before you can specify a non-default region when creating buckets using the Tenant Manager or Tenant Management API or with the LocationConstraint request element for S3 PUT Bucket API requests. An error occurs if a PUT Bucket request uses a region that has not been defined in StorageGRID.
- You must use the exact region name when you create the S3 bucket. Region names are case sensitive. Valid characters are numbers, letters, and hyphens.

> ⓘ EU is not considered to be an alias for eu-west-1. If you want to use the EU or eu-west-1 region, you must use the exact name.

- You can’t delete or modify a region if it is currently used within the active ILM policy or the proposed ILM policy.
- If the region used as the advanced filter in an ILM rule is invalid, it is still possible to add that rule to the proposed policy. However, an error occurs if you attempt to save or activate the proposed policy.

  An invalid region can result if you use a region as an advanced filter in an ILM rule but you later delete that region, or if you use the Grid Management API to create a rule and specify a region that you have not defined.

- If you delete a region after using it to create an S3 bucket, you will need to re-add the region if you ever want to use the Location Constraint advanced filter to find objects in that bucket.

### Steps

1. Select **ILM** > **Regions**.

   The Regions page appears, with the currently defined regions listed. **Region 1** shows the default region, `us-east-1`, which can’t be modified or removed.

2. To add a region:
   a. Select the insert icon ➕ to the right of the last entry.
   b. Enter the name of a region that you want to use when creating S3 buckets.

      You must use this exact region name as the LocationConstraint request element when you create the corresponding S3 bucket.

3. To remove an unused region, select the delete icon ✕.

An error message appears if you attempt to remove a region that is currently used in the active policy or the proposed policy.

4. When you are done making changes, select **Save**.

   You can now select these regions from the Advanced filters section in step 1 of the Create ILM rule wizard. See Use advanced filters in ILM rules.

# Create ILM rule

## Create an ILM rule: Overview

To manage objects, you create a set of information lifecycle management (ILM) rules and organize them into an ILM policy.

Every object ingested into the system is evaluated against the active policy. When a rule in the policy matches an object's metadata, the instructions in the rule determine what actions StorageGRID takes to copy and store that object.

> ⓘ Object metadata is not managed by ILM rules. Instead, object metadata is stored in a Cassandra database in what is known as a metadata store. Three copies of object metadata are automatically maintained at each site to protect the data from loss.

### Elements of an ILM rule

An ILM rule has three elements:

- **Filtering criteria**: A rule's basic and advanced filters define which objects the rule applies to. If an object matches all filters, StorageGRID applies the rule and creates the object copies specified in the rule's placement instructions.
- **Placement instructions**: A rule's placement instructions define the number, type, and location of object copies. Each rule can include a sequence of placement instructions to change the number, type, and location of object copies over time. When the time period for one placement expires, the instructions in the next placement are automatically applied by the next ILM evaluation.
- **Ingest behavior**: A rule's ingest behavior allows you to choose how the objects filtered by the rule are protected as they are ingested (when an S3 or Swift client saves an object to the grid).

### ILM rule filtering

When you create an ILM rule, you specify filters to identify which objects the rule applies to.

In the simplest case, a rule might not use any filters. Any rule that does not use filters applies to all objects, so it must be the last (default) rule in an ILM policy. The default rule provides storage instructions for objects that don't match the filters in another rule.

- Basic filters allow you to apply different rules to large, distinct groups of objects. These filters allow you to apply a rule to specific tenant accounts, specific S3 buckets or Swift containers, or both.

  Basic filters give you a simple way to apply different rules to large numbers of objects. For example, your company's financial records might need to be stored to meet regulatory requirements, while data from the marketing department might need to be stored to facilitate daily operations. After creating separate tenant accounts for each department or after segregating data from the different departments into separate S3 buckets, you can easily create one rule that applies to all financial records and a second rule that applies to all marketing data.

- Advanced filters give you granular control. You can create filters to select objects based on the following object properties:
    - Ingest time
    - Last access time
    - All or part of the object name (Key)
    - Location constraint (S3 only)
    - Object size
    - User metadata
    - Object tag (S3 only)

You can filter objects on very specific criteria. For example, objects stored by a hospital's imaging department might be used frequently when they are less than 30 days old and infrequently afterwards, while objects that contain patient visit information might need to be copied to the billing department at the health network's headquarters. You can create filters that identify each type of object based on object name, size, S3 object tags, or any other relevant criteria, and then create separate rules to store each set of objects appropriately.

You can combine filters as needed in a single rule. For example, the marketing department might want to store large image files differently than their vendor records, while the Human Resources department might need to store personnel records in a specific geography and policy information centrally. In this case you can create rules that filter by tenant account to segregate the records from each department, while using filters in each rule to identify the specific type of objects that the rule applies to.

### ILM rule placement instructions

Placement instructions determine where, when, and how object data is stored. An ILM rule can include one or more placement instructions. Each placement instruction applies to a single period of time.

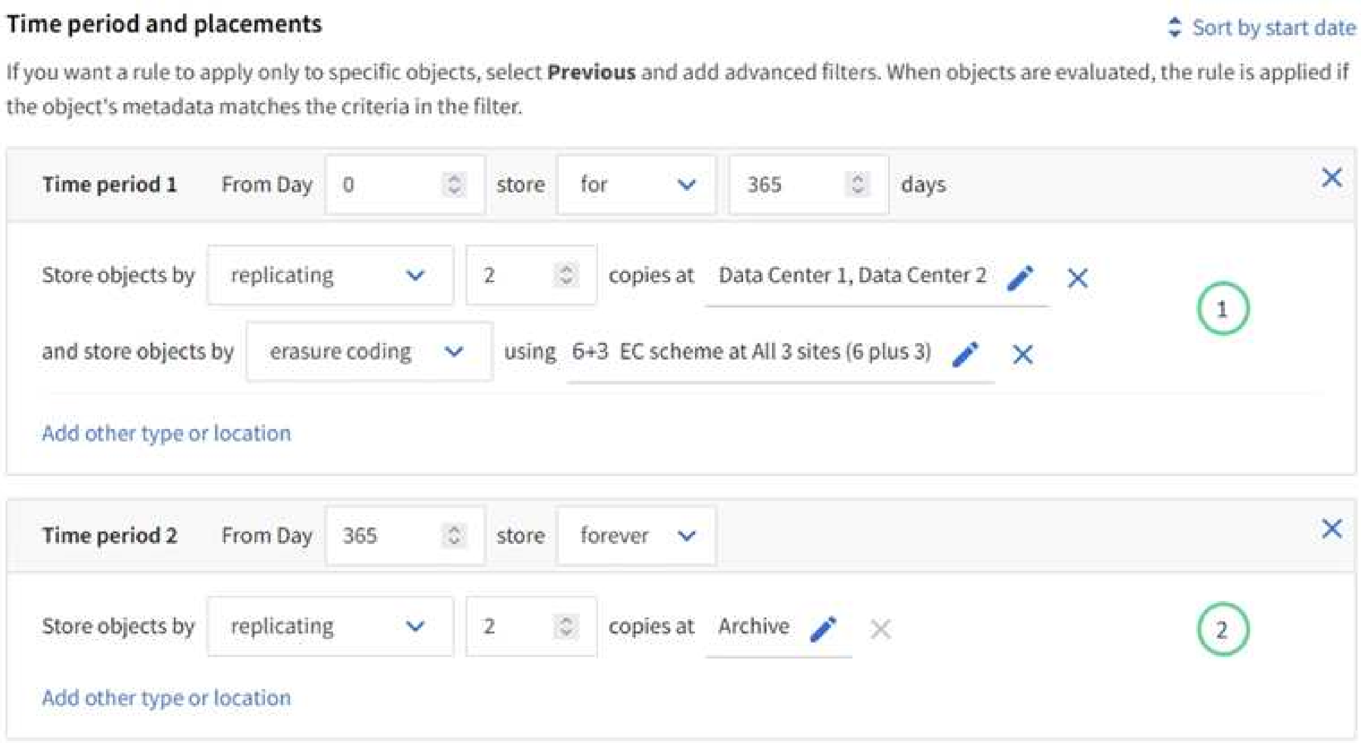
When you create placement instructions:

- You start by specifying the reference time, which determines when the placement instructions start. The reference time might be when an object is ingested, when an object is accessed, when a versioned object becomes noncurrent, or a user-defined time.
- Next, you specify when the placement will apply, relative to the reference time. For example, a placement might start on day 0 and continue for 365 days, relative to when the object was ingested.
- Finally, you specify the type of copies (replication or erasure coding) and the location where the copies are stored. For example, you might want to store two replicated copies at two different sites.

Each rule can define multiple placements for a single time period and different placements for different time periods.

- To place objects in multiple locations during a single time period, select **Add other type or location** to add more than one line for that time period.
- To place objects in different locations in different time periods, select **Add another time period** to add the next time period. Then, specify one or more lines within the time period.

The example shows two placement instructions on the Define placements page of the Create ILM rule wizard.



Time period and placements

Sort by start date

If you want a rule to apply only to specific objects, select **Previous** and add advanced filters. When objects are evaluated, the rule is applied if the object's metadata matches the criteria in the filter.

Time period 1 From Day 0 store for 365 days

Store objects by replicating 2 copies at Data Center 1, Data Center 2 ①

and store objects by erasure coding using 6+3 EC scheme at All 3 sites (6 plus 3)

Add other type or location

Time period 2 From Day 365 store forever

Store objects by replicating 2 copies at Archive ②

Add other type or location

The first placement instruction ① has two lines for the first year:

- The first line creates two replicated object copies at two data center sites.
- The second line creates a 6+3 erasure-coded copy using three data center sites.

The second placement instruction ② creates two archived copies after one year and keeps those copies forever.

When you define the set of placement instructions for a rule, you must ensure that at least one placement instruction begins at day 0, that there are no gaps between the time periods you have defined, and that the final placement instruction continues either forever or until you no longer require any object copies.

As each time period in the rule expires, the content placement instructions for the next time period are applied. New object copies are created and any unneeded copies are deleted.

### ILM rule ingest behavior

Ingest behavior controls whether object copies are immediately placed according to the instructions in the rule, or if interim copies are made and the placement instructions are applied later. The following ingest behaviors are available for ILM rules:

- **Balanced**: StorageGRID attempts to make all copies specified in the ILM rule at ingest; if this is not possible, interim copies are made and success is returned to the client. The copies specified in the ILM rule are made when possible.
- **Strict**: All copies specified in the ILM rule must be made before success is returned to the client.
- **Dual commit**: StorageGRID immediately makes interim copies of the object and returns success to the client. Copies specified in the ILM rule are made when possible.

#### Related information

- Ingest options
- Advantages, disadvantages, and limitations of the ingest options
- How consistency controls and ILM rules interact to affect data protection

#### Example ILM rule

As an example, an ILM rule could specify the following:

- Apply only to the objects belonging to Tenant A.
- Make two replicated copies of those objects and store each copy at a different site.
- Retain the two copies “forever,” which means that StorageGRID will not automatically delete them. Instead, StorageGRID will retain these objects until they are deleted by a client delete request or by the expiration of a bucket lifecycle.
- Use the Balanced option for ingest behavior: the two-site placement instruction is applied as soon as Tenant A saves an object to StorageGRID, unless it is not possible to immediately make both required copies.

  For example, if Site 2 is unreachable when Tenant A saves an object, StorageGRID will make two interim copies on Storage Nodes at Site 1. As soon as Site 2 becomes available, StorageGRID will make the required copy at that site.

#### Related information

- What is a storage pool?
- What is a Cloud Storage Pool?

## Access the Create an ILM rule wizard

ILM rules allow you to manage the placement of object data over time. To create an ILM rule, you use the Create an ILM rule wizard.

> If you want to create the default ILM rule for a policy, follow the instructions for creating a default ILM rule instead.

#### Before you begin

- You are signed in to the Grid Manager using a supported web browser.
- You have specific access permissions.
- If you want to specify which tenant accounts this rule applies to, you have the Tenant accounts permission or you know the account ID for each account.
- If you want the rule to filter objects on last access time metadata, Last access time updates must be enabled by bucket for S3 or by container for Swift.
- You have configured any Cloud Storage Pools you plan to use. See Create Cloud Storage Pool.
- You are familiar with the ingest options.
- If you need to create a compliant rule for use with S3 Object Lock, you are familiar with the requirements for S3 Object Lock.
- Optionally, you have watched the video: Video: Information lifecycle management rules in StorageGRID 11.7.

### About this task

When creating ILM rules:

- Consider the StorageGRID system’s topology and storage configurations.
- Consider what types of object copies you want to make (replicated or erasure coded) and the number of copies of each object that are required.
- Determine what types of object metadata are used in the applications that connect to the StorageGRID system. ILM rules filter objects based on their metadata.
- Consider where you want object copies to be placed over time.
- Decide which ingest option to use (Balanced, Strict, or Dual commit).

### Steps

1. Select **ILM** > **Rules**.

   Based on the number of sites in the grid, the Make 2 Copies rule or the 1 Copy Per Site rule is shown in the list of rules.

   > If the global S3 Object Lock setting has been enabled for the StorageGRID system, the summary table includes a **Compliant** column, and the details for the selected rule include a **Compliant** field.

2. Select **Create**. Step 1 (Enter details) of the Create an ILM rule wizard appears.

## Step 1 of 3: Enter details

The **Enter details** step of the Create an ILM rule wizard allows you to enter a name and description for the rule and to define filters for the rule.

Entering a description and defining filters for the rule are optional.

### About this task

When evaluating an object against an ILM rule, StorageGRID compares the object metadata to the rule’s filters. If the object metadata matches all filters, StorageGRID uses the rule to place the object. You can design a rule to apply to all objects, or you can specify basic filters, such as one or more tenant accounts or bucket names, or advanced filters, such as the object’s size or user metadata.

### Steps

1. Enter a unique name for the rule in the **Name** field.
2. Optionally, enter a short description for the rule in the **Description** field.

   You should describe the rule’s purpose or function so you can recognize the rule later.

3. Optionally, select one or more S3 or Swift tenant accounts to which this rule applies. If this rule applies to all tenants, leave this field blank.

   If you don't have either the Root access permission or the Tenant accounts permission, you can't select tenants from the list. Instead, enter the tenant ID or enter multiple IDs as a comma-delimited string.

4. Optionally, specify the S3 buckets or Swift containers to which this rule applies.

   If **matches all** is selected (default), the rule applies to all S3 buckets or Swift containers.

5. For S3 tenants, optionally select **Yes** to apply the rule only to older object versions in S3 buckets that have versioning enabled.

   If you select **Yes**, "Noncurrent time" will be automatically selected for Reference time in Step 2 of the Create an ILM rule wizard.

   > ⓘ Noncurrent time applies only to S3 objects in versioning-enabled buckets. See Operations on buckets, PUT Bucket versioning and Manage objects with S3 Object Lock.

   You can use this option to reduce the storage impact of versioned objects by filtering for noncurrent object versions. See Example 4: ILM rules and policy for S3 versioned objects.

6. Optionally, select **Add an advanced filter** to specify additional filters.

   If you don't configure advanced filtering, the rule applies to all objects that match the basic filters. For more information about advanced filtering, see Use advanced filters in ILM rules and Specify multiple metadata types and values.

7. Select **Continue**. Step 2 (Define placements) of the Create an ILM rule wizard appears.

### Use advanced filters in ILM rules

Advanced filtering allows you to create ILM rules that apply only to specific objects based on their metadata. When you set up advanced filtering for a rule, you select the type of metadata you want to match, select an operator, and specify a metadata value. When objects are evaluated, the ILM rule is applied only to those objects that have metadata matching the advanced filter.

The table shows the types of metadata you can specify in advanced filters, the operators you can use for each type of metadata, and the metadata values expected.

| Metadata type | Supported operators | Metadata value |
|---|---|---|
| Ingest time | • is<br>• is not<br>• is before<br>• is on or before<br>• is after<br>• is on or after | Time and date the object was ingested.<br>**Note:** To avoid resource issues when activating an new ILM policy, you can use the Ingest time advanced filter in any rule that might change the location of large numbers of existing objects. Set Ingest time to be greater than or equal to the approximate time when the new policy will go into effect to ensure that existing objects aren't moved unnecessarily. |

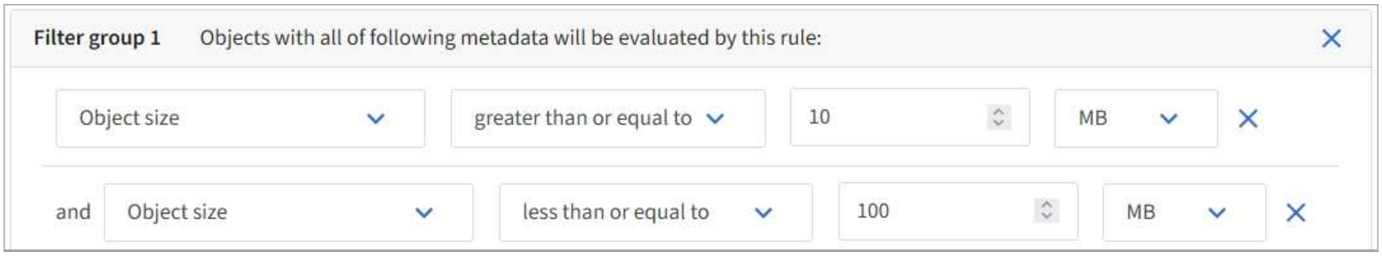
| Metadata type | Supported operators | Metadata value |
|---|---|---|
| Key | • equals<br>• does not equal<br>• contains<br>• does not contain<br>• starts with<br>• does not start with<br>• ends with<br>• does not end with | All or part of a unique S3 or Swift object key.<br>For example, you might want to match objects that end with `.txt` or start with `test-object/`. |
| Last access time | • is<br>• is not<br>• is before<br>• is on or before<br>• is after<br>• is on or after | Time and date the object was last retrieved (read or viewed).<br>**Note:** If you plan to use last access time as an advanced filter, Last access time updates must be enabled for the S3 bucket or Swift container. |
| Location constraint (S3 only) | • equals<br>• does not equal | The region where an S3 bucket was created. Use **ILM** > **Regions** to define the regions that are shown.<br>**Note:** A value of us-east-1 will match objects in buckets created in the us-east-1 region as well as objects in buckets that have no region specified. See Configure regions (optional and S3 only). |
| Object size | • equals<br>• does not equal<br>• less than<br>• less than or equal to<br>• greater than<br>• greater than or equal to | The object's size.<br>Erasure coding is best suited for objects greater than 1 MB. Don't use erasure coding for objects smaller than 200 KB to avoid the overhead of managing very small erasure-coded fragments.<br>**Note:** To filter on object sizes smaller than 1 MB, enter a decimal value. Your browser type and locale settings control whether you need to use a period or a comma as the decimal separator. |

| Metadata type | Supported operators | Metadata value |
|---|---|---|
| User metadata | • contains<br>• ends with<br>• equals<br>• exists<br>• does not contain<br>• does not end with<br>• does not equal<br>• does not exist<br>• does not start with<br>• starts with | Key-value pair, where **User metadata name** is the key and **Metadata value** is the value.<br>For example, to filter on objects that have user metadata of `color=blue`, specify `color` for **User metadata name**, `equals` for the operator, and `blue` for **Metadata value**.<br>**Note:** User-metadata names aren't case sensitive; user-metadata values are case sensitive. |
| Object tag (S3 only) | • contains<br>• ends with<br>• equals<br>• exists<br>• does not contain<br>• does not end with<br>• does not equal<br>• does not exist<br>• does not start with<br>• starts with | Key-value pair, where **Object tag name** is the key and **Object tag value** is the value.<br>For example, to filter on objects that have an object tag of `Image=True`, specify `Image` for **Object tag name**, `equals` for the operator, and `True` for **Object tag value**.<br>**Note:** Object tag names and object tag values are case sensitive. You must enter these items exactly as they were defined for the object. |

### Specify multiple metadata types and values

When you define advanced filtering, you can specify multiple types of metadata and multiple metadata values. For example, if you want a rule to match objects between 10 MB and 100 MB in size, you would select the **Object size** metadata type and specify two metadata values.

- The first metadata value specifies objects greater than or equal to 10 MB.
- The second metadata value specifies objects less than or equal to 100 MB.

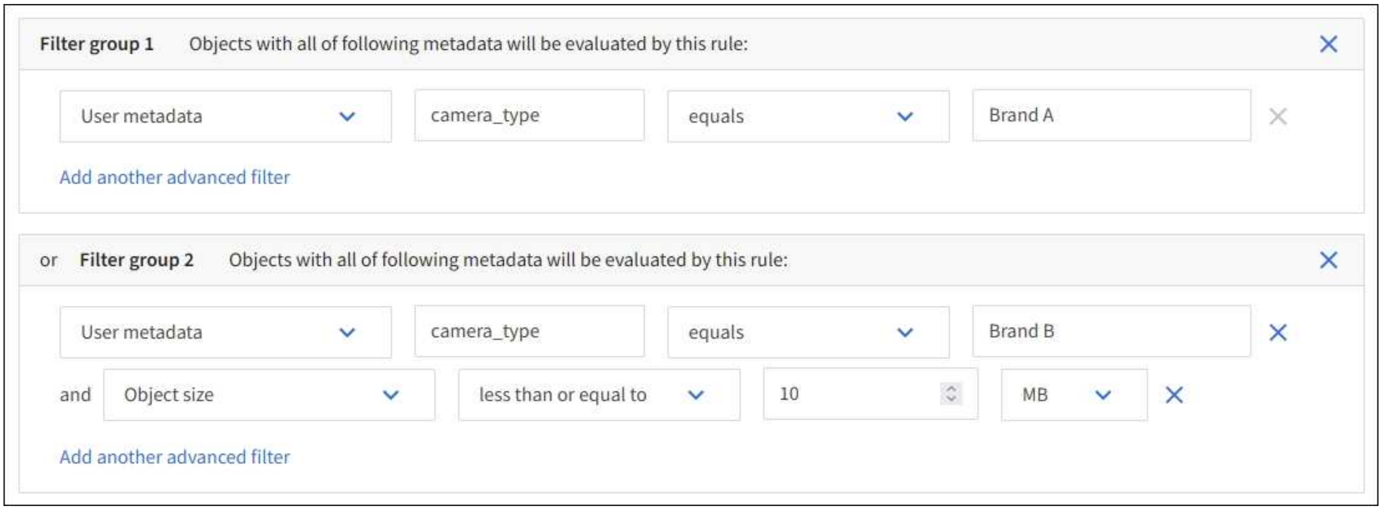


**Filter group 1** Objects with all of following metadata will be evaluated by this rule:

Object size | greater than or equal to | 10 | MB

and Object size | less than or equal to | 100 | MB

Using multiple entries allows you to have precise control over which objects are matched. In the following example, the rule applies to objects that have a Brand A or Brand B as the value of the camera_type user metadata. However, the rule only applies to those Brand B objects that are smaller than 10 MB.



**Filter group 1** Objects with all of following metadata will be evaluated by this rule:

User metadata | camera_type | equals | Brand A

Add another advanced filter

or **Filter group 2** Objects with all of following metadata will be evaluated by this rule:

User metadata | camera_type | equals | Brand B

and Object size | less than or equal to | 10 | MB

Add another advanced filter

## Step 2 of 3: Define placements

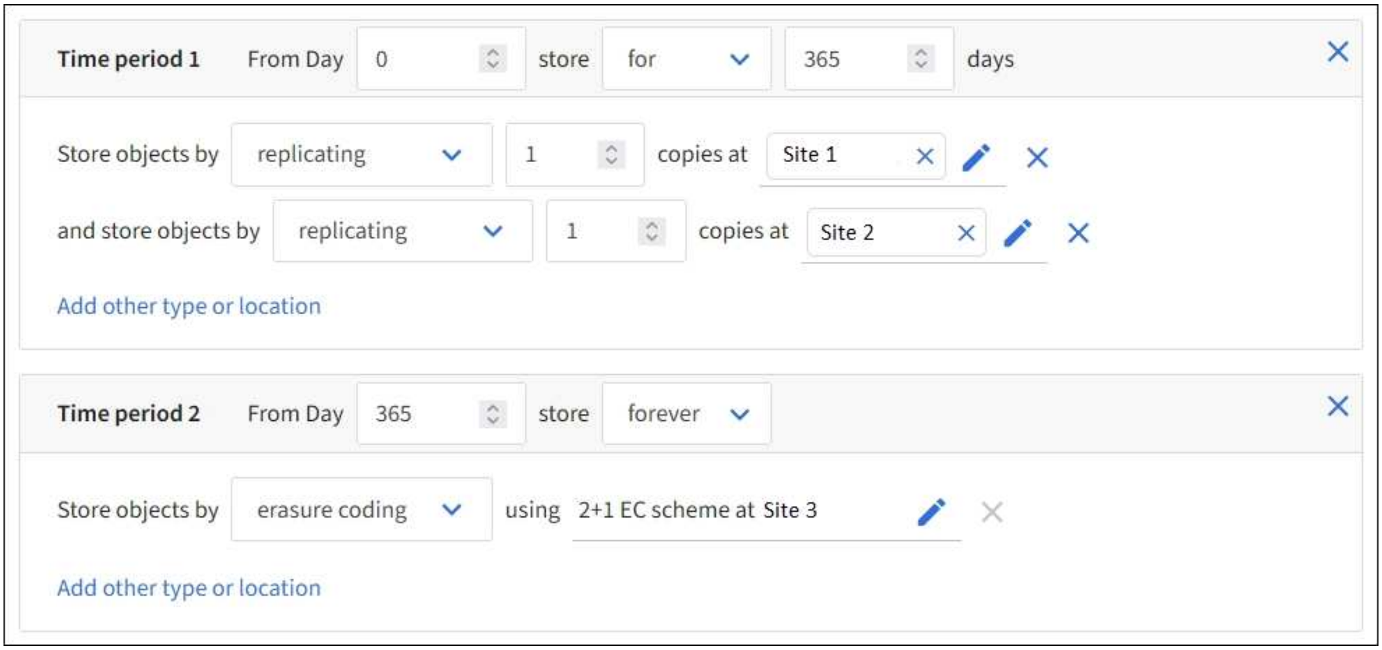
The **Define placements** step of the Create ILM Rule wizard allows you to define the placement instructions that determine how long objects are stored, the type of copies (replicated or erasure coded), the storage location, and the number of copies.

### About this task

An ILM rule can include one or more placement instructions. Each placement instruction applies to a single period of time. When you use more than one instruction, the time periods must be contiguous, and at least one instruction must start on day 0. The instructions can continue either forever, or until you no longer require any object copies.

Each placement instruction can have multiple lines if you want to create different types of copies or use different locations during that time period.

In this example, the ILM rule stores one replicated copy in Site 1 and one replicated copy in Site 2 for the first year. After one year, a 2+1 erasure-coded copy is made and saved at only one site.



**Time period 1** From Day 0 store for 365 days

Store objects by replicating 1 copies at Site 1

and store objects by replicating 1 copies at Site 2

Add other type or location

**Time period 2** From Day 365 store forever

Store objects by erasure coding using 2+1 EC scheme at Site 3

Add other type or location

**Steps**

1. For **Reference time**, select the type of time to use when calculating the start time for a placement instruction.

| Option | Description |
|---|---|
| Ingest time | The time when the object was ingested. |
| Last access time | The time when the object was last retrieved (read or viewed).<br>**Note:** To use this option, updates to Last access time must be enabled for the S3 bucket or Swift container. See Use Last access time in ILM rules. |
| User defined creation time | A time specified in user-defined metadata. |
| Noncurrent time | “Noncurrent time” is automatically selected if you selected **Yes** for the question, “Apply this rule to older object versions only (in S3 buckets with versioning enabled)?” in Step 1 of the Create an ILM rule wizard. |

ⓘ If you want to create a compliant rule, you must select **Ingest time**. See Manage objects with S3 Object Lock.

2. In the **Time period and placements** section, enter a starting time and a duration for the first time period.

   For example, you might want to specify where to store objects for the first year (*From day 0 store for 365 days*). At least one instruction must start at day 0.

3. If you want to create replicated copies:

   a. From the **Store objects by** drop-down list, select **replicating**.

   b. Select the number of copies you want to make.

      A warning appears if you change the number of copies to 1. An ILM rule that creates only one replicated copy for any time period puts data at risk of permanent loss. See Why you should not use single-copy replication.

      To avoid the risk, do one or more of the following:

      - Increase the number of copies for the time period.
      - Add copies to other storage pools or to a Cloud Storage Pool.
      - Select **erasure coding** instead of **replicating**.

        You can safely ignore this warning if this rule already creates multiple copies for all time periods.

   c. In the **copies at** field, select the storage pools you want to add.

      **If you specify only one storage pool**, be aware that StorageGRID can store only one replicated copy of an object on any given Storage Node. If your grid includes three Storage Nodes and you select 4 as the number of copies, only three copies will be made—one copy for each Storage Node.

ⓘ The **ILM placement unachievable** alert is triggered to indicate that the ILM rule could not be completely applied.

**If you specify more than one storage pool**, keep these rules in mind:

- The number of copies can't be greater than the number of storage pools.
- If the number of copies equals the number of storage pools, one copy of the object is stored in each storage pool.
- If the number of copies is less than the number of storage pools, one copy is stored at the ingest site, and then the system distributes the remaining copies to keep disk usage among the pools balanced, while ensuring that no site gets more than one copy of an object.
- If the storage pools overlap (contain the same Storage Nodes), all copies of the object might be saved at only one site. For this reason, don't specify the All Storage Nodes storage pool (StorageGRID 11.6 and earlier) and another storage pool.

4. If you want to create an erasure-coded copy:

   a. From the **Store objects by** drop-down list, select **erasure coding**.

   ⓘ Erasure coding is best suited for objects greater than 1 MB. Don't use erasure coding for objects smaller than 200 KB to avoid the overhead of managing very small erasure-coded fragments.

   b. If you didn't add an Object size filter for a value greater than 0.2 MB, select **Previous** to return to Step 1. Then, select **Add an advanced filter** and set an **Object size** filter to any value greater than 0.2 MB.

   c. Select the storage pool you want to add and the erasure-coding scheme you want to use.

   The storage location for an erasure-coded copy includes the name of the erasure coding scheme, followed by the name of the storage pool.

5. Optionally:

   a. Select **Add other type or location** to create additional copies at different locations.

   b. Select **Add another time period** to add different time periods.

   ⓘ Objects are automatically deleted at the end of the final time period unless the final time period ends with **forever**.

6. If you want to store objects in a Cloud Storage Pool:

   a. In the **Store objects by** drop-down list, select **replicating**.

   b. Select the **copies at** field, then select a Cloud Storage Pool.

   When using Cloud Storage Pools, keep these rules in mind:

   - You can't select more than one Cloud Storage Pool in a single placement instruction. Similarly, you can't select a Cloud Storage Pool and a storage pool in the same placement instruction.
   - You can store only one copy of an object in any given Cloud Storage Pool. An error message appears if you set **Copies** to 2 or more.
   - You can't store more than one object copy in any Cloud Storage Pool at the same time. An error message appears if multiple placements that use a Cloud Storage Pool have overlapping dates or if multiple lines in the same placement use a Cloud Storage Pool.

- You can store an object in a Cloud Storage Pool at the same time that object is being stored as replicated or erasure coded copies in StorageGRID. However, you must include more than one line in the placement instruction for the time period, so you can specify the number and types of copies for each location.

7. In the Retention diagram, confirm your placement instructions.

   Each line in the diagram shows where and when object copies will be placed. The color of a line represents the type of copy:

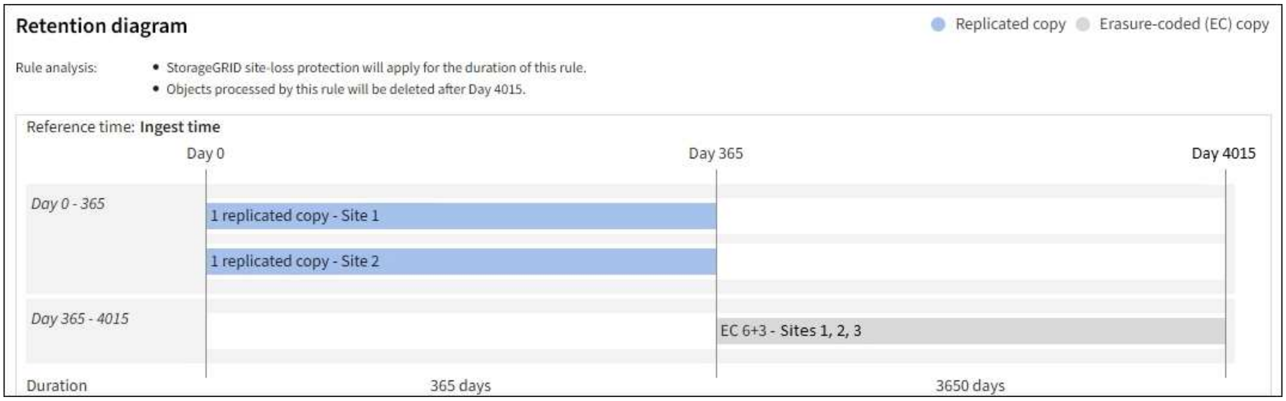
| | |
|---|---|
| 🟦 | Replicated copy |
| ⬜ | Erasure-coded copy |
| 🟨 | Cloud Storage Pool copy |

   In this example, the ILM rule stores one replicated copy in Site 1 and one replicated copy in Site 2 for the first year. After one year and for an additional 10 years, a 6+3 erasure-coded copy will be saved at three sites. After 11 years total, the objects will be deleted from StorageGRID.

   The Rule analysis section of the Retention diagram states:

   - StorageGRID site-loss protection will apply for the duration of this rule.
   - Objects processed by this rule will be deleted after Day 4015.

   See Enable site-loss protection.



Retention diagram

● Replicated copy ● Erasure-coded (EC) copy

Rule analysis:

- StorageGRID site-loss protection will apply for the duration of this rule.
- Objects processed by this rule will be deleted after Day 4015.

Reference time: **Ingest time**

8. Select **Continue**. Step 3 (Select ingest behavior) of the Create an ILM rule wizard appears.

## Use Last access time in ILM rules

You can use Last access time as the reference time in an ILM rule. For example, you might want to leave objects that have been viewed in the last three months on local Storage Nodes, while moving objects that have not been viewed as recently to an off-site location. You can also use Last access time as an advanced filter if you want an ILM rule to apply only to objects that were last accessed on a specific date.

**About this task**

Before using Last access time in an ILM rule, review the following considerations:

- When using Last access time as a reference time, be aware that changing the Last access time for an object does not trigger an immediate ILM evaluation. Instead, the object's placements are assessed and the object is moved as required when background ILM evaluates the object. This could take two weeks or more after the object is accessed.

  Take this latency into account when creating ILM rules based on Last access time and avoid placements that use short time periods (less than one month).

- When using Last access time as an advanced filter or as a reference time, you must enable last access time updates for S3 buckets. You can use the Tenant Manager or the Tenant Management API.

  ⓘ Last access time updates are always enabled for Swift containers, but are disabled by default for S3 buckets.

  ⓘ Be aware that enabling last access time updates can reduce performance, especially in systems with small objects. The performance impact occurs because StorageGRID must update the objects with new timestamps every time the objects are retrieved.

The following table summarizes whether the Last access time is updated for all objects in the bucket for different types of requests.

| Type of request | Whether Last access time is updated when last access time updates are disabled | Whether Last access time is updated when last access time updates are enabled |
|---|---|---|
| Request to retrieve an object, its access control list, or its metadata | No | Yes |
| Request to update an object's metadata | Yes | Yes |
| Request to copy an object from one bucket to another | • No, for the source copy<br>• Yes, for the destination copy | • Yes, for the source copy<br>• Yes, for the destination copy |
| Request to complete a multipart upload | Yes, for the assembled object | Yes, for the assembled object |

## Step 3 of 3: Select ingest behavior

The **Select ingest behavior** step of the Create ILM Rule wizard allows you to choose how the objects filtered by this rule are protected as they are ingested.

### About this task

StorageGRID can make interim copies and queue the objects for ILM evaluation later, or it can make copies to meet the rule's placement instructions immediately.

### Steps

1. Select the ingest behavior to use.

   For more information, see Advantages, disadvantages, and limitations of the ingest options.

   > You can't use the Balanced or Strict option if the rule uses one of these placements:
   >
   > - A Cloud Storage Pool at day 0
   > - An Archive Node at day 0
   > - A Cloud Storage Pool or an Archive Node when the rule uses a User defined creation time as a Reference time
   >
   > See Example 5: ILM rules and policy for Strict ingest behavior.

2. Select **Create**.

   The ILM rule is created. The rule does not become active until it is added to an ILM policy and that policy is activated.

   To view the details of the rule, select the rule's name on the ILM rules page.

## Create a default ILM rule

Before creating an ILM policy, you must create a default rule to place any objects not matched by another rule in the policy. The default rule can't use any filters. It must apply to all tenants, all buckets, and all object versions.

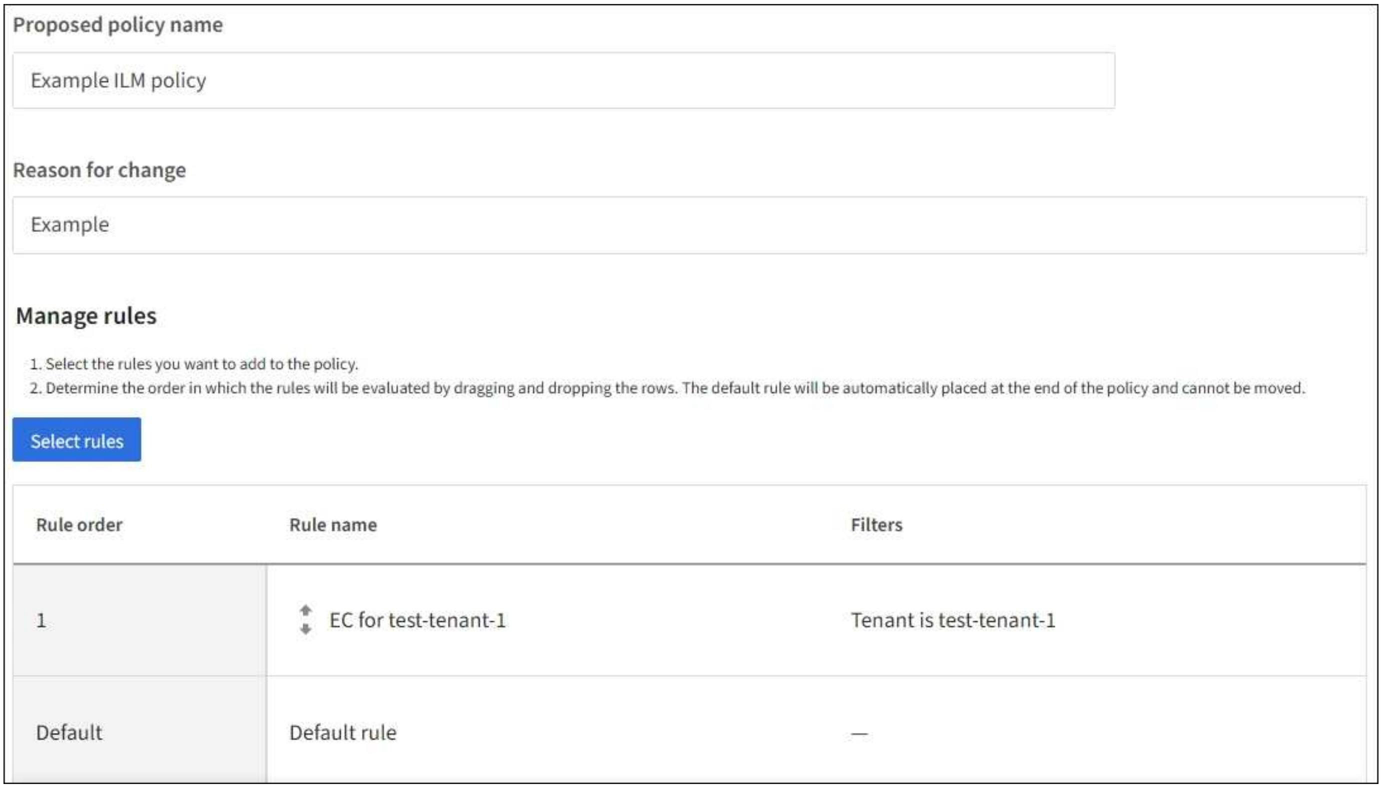
### Before you begin

- You are signed in to the Grid Manager using a supported web browser.
- You have specific access permissions.

### About this task

The default rule is the last rule to be evaluated in an ILM policy, so it can't use any filters. The placement instructions for the default rule are applied to any objects that aren't matched by another rule in the policy.

In this example policy, the first rule applies only to objects belonging to test-tenant-1. The default rule, which is last, applies to objects belonging to all other tenant accounts.



Proposed policy name

Example ILM policy

Reason for change

Example

Manage rules

1. Select the rules you want to add to the policy.
2. Determine the order in which the rules will be evaluated by dragging and dropping the rows. The default rule will be automatically placed at the end of the policy and cannot be moved.

| Rule order | Rule name | Filters |
|---|---|---|
| 1 | ↕ EC for test-tenant-1 | Tenant is test-tenant-1 |
| Default | Default rule | — |

When you create the default rule, keep these requirements in mind:

- The default rule is automatically placed as the last rule in the policy.
- The default rule can’t use any basic or advanced filters.
- The default rule must apply to all object versions.
- The default rule should create replicated copies.

> ⓘ Don’t use a rule that creates erasure-coded copies as the default rule for a policy. Erasure-coding rules should use an advanced filter to prevent smaller objects from being erasure coded.

- In general, the default rule should retain objects forever.
- If you are using (or you plan to enable) the global S3 Object Lock setting, the default rule for the active or proposed policy must be compliant.

**Steps**

1. Select **ILM** > **Rules**.
2. Select **Create**.

   Step 1 (Enter details) of the Create ILM rule wizard appears.
3. Enter a unique name for the rule in the **Rule name** field.
4. Optionally, enter a short description for the rule in the **Description** field.
5. Leave the **Tenant accounts** field blank.

   The default rule must apply to all tenant accounts.

6. Leave the Bucket name drop-down selection as **matches all**.

   The default rule must apply to all S3 buckets and Swift containers.

7. Keep the default answer, **No**, for the question, “Apply this rule to older object versions only (in S3 buckets with versioning enabled)?”
8. Don’t add advanced filters.

   The default rule can’t specify any filters.

9. Select **Next**.

   Step 2 (Define placements) appears.

10. For Reference time, select any option.

    If you kept the default answer, **No**, for the question, “Apply this rule to older object versions only?” Noncurrent time will not be included in the pull-down list. The default rule must apply all object versions.

11. Specify the placement instructions for the default rule.
    - The default rule should retain objects forever. A warning appears when you activate a new policy if the default rule does not retain objects forever. You must confirm this is the behavior you expect.
    - The default rule should create replicated copies.

      > Don’t use a rule that creates erasure-coded copies as the default rule for a policy. Erasure-coding rules should include the **Object size (MB) greater than 0.2** advanced filter to prevent smaller objects from being erasure coded.

    - If you are using (or you plan to enable) the global S3 Object Lock setting, the default rule must be compliant:
      - It must create at least two replicated object copies or one erasure-coded copy.
      - These copies must exist on Storage Nodes for the entire duration of each line in the placement instructions.
      - Object copies can’t be saved in a Cloud Storage Pool.
      - Object copies can’t be saved on Archive Nodes.
      - At least one line of the placement instructions must start at day 0, using Ingest time as the reference time.
      - At least one line of the placement instructions must be “forever.”
12. Look at the Retention diagram to confirm your placement instructions.
13. Select **Continue**.

    Step 3 (Select ingest behavior) appears.

14. Select the ingest option to use, and select **Create**.

# Create ILM policy

## Create an ILM policy: Overview

An information lifecycle management (ILM) policy is an ordered set of ILM rules that determines how the StorageGRID system manages object data over time.

When you create an ILM policy, you start by selecting and arranging the ILM rules. Then, you verify the behavior of your proposed policy by simulating it against previously ingested objects. When you are satisfied that the proposed policy is functioning as intended, you can activate it to create the active policy.

> ! An ILM policy that has been incorrectly configured can result in unrecoverable data loss. Before activating an ILM policy, carefully review the ILM policy and its ILM rules, and then simulate the ILM policy. Always confirm that the ILM policy will work as intended.

### Default ILM policy

When you install StorageGRID and add sites, a default ILM policy is automatically created. If your grid contains one site, the default policy contains a default rule that replicates two copies of each object at that site. If your grid contains more than one site, the default rule replicates one copy of each object at each site.

If the default policy does not meet your storage requirements, you can create your own rules and policy. See What an ILM rule is and Creating a proposed ILM policy.

### How does an ILM policy evaluate objects?

The active ILM policy for your StorageGRID system controls the placement, duration, and data protection of all objects.

When clients save objects to StorageGRID, the objects are evaluated against the ordered set of ILM rules in the active policy, as follows:

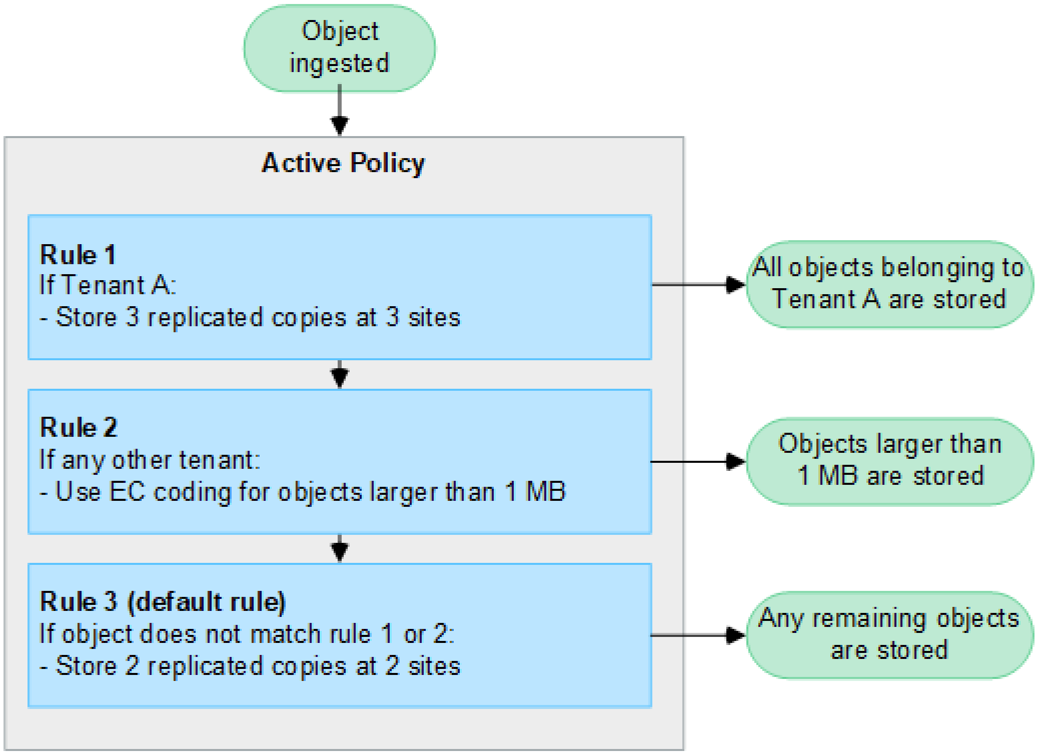
1. If the filters for the first rule in the policy match an object, the object is ingested according to that rule's ingest behavior and stored according to that rule's placement instructions.
2. If the filters for the first rule don't match the object, the object is evaluated against each subsequent rule in the policy until a match is made.
3. If no rules match an object, the ingest behavior and placement instructions for the default rule in the policy are applied. The default rule is the last rule in a policy. The default rule must apply to all tenants, all buckets, and all object versions and can't use any advanced filters.

### Example ILM policy

As an example, an ILM policy could contain three ILM rules that specify the following:

- **Rule 1: Replicated copies for Tenant A**
  - Match all objects belonging to Tenant A.
  - Store these objects as three replicated copies at three sites.
  - Objects belonging to other tenants aren't matched by Rule 1, so they are evaluated against Rule 2.
- **Rule 2: Erasure coding for objects greater than 1 MB**
  - Match all objects from other tenants, but only if they are greater than 1 MB. These larger objects are stored using 6+3 erasure coding at three sites.
  - Does not match objects 1 MB or smaller, so these objects are evaluated against Rule 3.

- **Rule 3: 2 copies 2 data centers** (default)
  - Is the last and default rule in the policy. Does not use filters.
  - Make two replicated copies of all objects not matched by Rule 1 or Rule 2 (objects not belonging to Tenant A that are 1 MB or smaller).


Object ingested
Active Policy
Rule 1
If Tenant A:
- Store 3 replicated copies at 3 sites
Rule 2
If any other tenant:
- Use EC coding for objects larger than 1 MB
Rule 3 (default rule)
If object does not match rule 1 or 2:
- Store 2 replicated copies at 2 sites
All objects belonging to Tenant A are stored
Objects larger than 1 MB are stored
Any remaining objects are stored

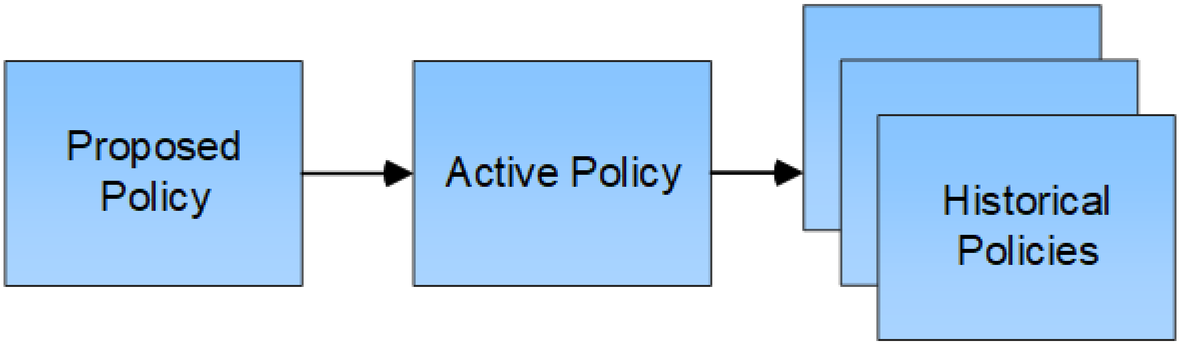
**What are proposed, active, and historical policies?**

Every StorageGRID system must have one active ILM policy. A StorageGRID system might also have one proposed ILM policy and any number of historical policies.

When you first create an ILM policy, you create a proposed policy by selecting one or more ILM rules and arranging them in a specific order. After you have simulated the proposed policy to confirm its behavior, you activate it to create the active policy.

When you activate a new ILM policy, StorageGRID uses that policy to manage all objects, including existing objects and newly ingested objects. Existing objects might be moved to new locations when the ILM rules in the new policy are implemented.

Activating the proposed policy causes the previously active policy to become a historical policy. Historical ILM policies can't be deleted.



### Considerations for creating an ILM policy

- Only use the system-provided policy, Baseline 2 copies policy, in test systems. For StorageGRID 11.6 and earlier, the Make 2 Copies rule in this policy uses the All Storage Nodes storage pool, which contains all sites. If your StorageGRID system has more than one site, two copies of an object might be placed on the same site.

> ⓘ The All Storage Nodes storage pool is automatically created during the installation of StorageGRID 11.6 and earlier. If you upgrade to a later version of StorageGRID, the All Storage Nodes pool will still exist. If you install StorageGRID 11.7 or later as a new installation, the All Storage Nodes pool is not created.

- When designing a new policy, consider all of the different types of objects that might be ingested into your grid. Make sure the policy includes rules to match and place these objects as required.
- Keep the ILM policy as simple as possible. This avoids potentially dangerous situations where object data is not protected as intended when changes are made to the StorageGRID system over time.
- Make sure that the rules in the policy are in the correct order. When the policy is activated, new and existing objects are evaluated by the rules in the order listed, starting at the top. For example, if the first rule in a policy matches an object, that object will not be evaluated by any other rule.
- The last rule in every ILM policy is the default ILM rule, which can't use any filters. If an object has not been matched by another rule, the default rule controls where that object is placed and for how long it is retained.
- Before activating a new policy, review any changes that the policy is making to the placement of existing objects. Changing an existing object's location might result in temporary resource issues when the new placements are evaluated and implemented.

## Create a proposed ILM policy

You can create a proposed ILM policy from scratch, or you can clone the current active policy if you want to start with the same set of rules.

Before creating your own policy, verify that the default ILM policy does not meet your storage requirements.

> ⓘ If the global S3 Object Lock setting has been enabled, you must ensure that the ILM policy is compliant with the requirements of buckets that have S3 Object Lock enabled. In this section, follow the instructions that mention having S3 Object Lock enabled.

### Before you begin

- You are signed in to the Grid Manager using a supported web browser.
- You have the required access permissions.
- You have created ILM rules based on whether S3 Object Lock is enabled.

**S3 Object Lock not enabled**

- You have created the ILM rules you want to add to the proposed policy. As required, you can save a proposed policy, create additional rules, and then edit the proposed policy to add the new rules.
- You have created a default ILM rule that does not contain any filters.

**S3 Object Lock enabled**

- The global S3 Object Lock setting is already enabled for the StorageGRID system.
- You have created the compliant and non-compliant ILM rules you want to add to the proposed policy. As required, you can save a proposed policy, create additional rules, and then edit the proposed policy to add the new rules.
- You have created a default ILM rule for the policy that is compliant.

- Optionally, you have watched the video: Video: Information lifecycle management policies in StorageGRID 11.7

Information lifecycle management policies in StorageGRID 11.7
Getting Started
NetApp

See also Create an ILM policy: Overview.

**About this task**

Typical reasons for creating a proposed ILM policy include:

- You added a new site and need to use new ILM rules to place objects at that site.
- You are decommissioning a site and you need to remove all ILM rules that refer to the site.
- You added a new tenant that has special data protection requirements.
- You started to use a Cloud Storage Pool.

> Only use the system-provided policy, Baseline 2 copies policy, in test systems. For StorageGRID 11.6 and earlier, the default rule in this policy uses the All Storage Nodes storage pool, which contains all sites. If your StorageGRID system has more than one site, two copies of an object might be placed on the same site.

**Steps**

1. Select **ILM** > **Policies**.

   If the global S3 Object Lock setting is enabled, the ILM policies page indicates which ILM rules are compliant.

2. Determine how you want to create the proposed ILM policy.

**Start from scratch**

a. If a proposed ILM policy currently exists, select **Proposed policy** > **Actions** > **Remove**.

   You can’t create a new proposed policy if a proposed policy already exists.

b. Select **Create proposed policy** > **Create new policy**.

**Start with rules from active policy**

a. If a proposed ILM policy currently exists, select **Proposed policy** > **Actions** > **Remove**.

   You can’t clone the active policy if a proposed policy already exists.

b. Select **Create proposed policy** > **Clone active policy**.

**Edit existing proposed policy**

a. Select **Proposed policy** > **Actions** > **Edit**.

3. In the **Proposed policy name** field, enter a unique name for the proposed policy.
4. In the **Reason for change** field, enter the reason you are creating a new proposed policy.
5. To add rules to the policy, select **Select rules**. Select a rule name to view the settings for that rule.

   ⓘ Periodically, the list of rules is automatically updated to reflect additions or removals. If a rule is removed after you select it, an error message appears.

   If you are cloning a policy:

   - The rules used by the policy you are cloning are selected.
   - If the policy you are cloning used any rules with no filters that were not the default rule, you are prompted to remove all but one of those rules.
   - If the default rule used a filter, you are prompted to select a new default rule.
   - If the default rule was not the last rule, you can move the rule to the end of the new policy.

**S3 Object Lock not enabled**

a. Select one default rule for the proposed policy. To create a new default rule, select **ILM rules page**.

   The default rule applies to any objects that don't match another rule in the policy. The default rule can't use any filters and is always evaluated last.

   > Don't use the Make 2 Copies rule as the default rule for a policy. The Make 2 Copies rule uses a single storage pool, All Storage Nodes, which contains all sites. If your StorageGRID system has more than one site, two copies of an object might be placed on the same site.

**S3 Object Lock enabled**

a. Select one default rule for the proposed policy. To create a new default rule, select **ILM rules page**.

   The list of rules contains only the rules that are compliant and don't use any filters.

   > Don't use the Make 2 Copies rule as the default rule for a policy. The Make 2 Copies rule uses a single storage pool, All Storage Nodes, which contains all sites. If you use this rule, multiple copies of an object might be placed on the same site.

b. If you need a different "default" rule for objects in non-compliant S3 buckets, select **Include a rule without filters for non-compliant S3 buckets**, and select one non-compliant rule that does not use a filter.

   For example, you might want to use a Cloud Storage Pool to store objects in buckets that don't have S3 Object Lock enabled.

   > You can only select one non-compliant rule that does not use a filter.

See also Example 7: Compliant ILM policy for S3 Object Lock.

6. When you are done selecting the default rule, select **Continue**.
7. For the Other rules step, select any other rules you want to add to the policy. These rules use at least one filter (tenant account, bucket name, advanced filter, or the Noncurrent reference time). Then select **Select**.

   The Create a proposed policy window now lists the rules you selected. The default rule is at the end, with the other rules above it.

   If S3 Object Lock is enabled and you also selected a non-compliant "default" rule, that rule is added as the second-to-last rule in the policy.

   > A warning appears if any rule does not retain objects forever. When you activate this policy, you must confirm that you want StorageGRID to delete objects when the placement instructions for the default rule elapse (unless a bucket lifecycle keeps the objects for a longer time period).

8. Drag the rows for the non-default rules to determine the order in which these rules will be evaluated.

   You can't move the default rule. If S3 Object Lock is enabled, you also can't move the non-compliant "default" rule if one was selected.

> You must confirm that the ILM rules are in the correct order. When the policy is activated, new and existing objects are evaluated by the rules in the order listed, starting at the top.

9. As required, select **Select rules** to add or remove rules.
10. When you are done, select **Save**.
11. Go to Simulate an ILM policy. You should always simulate a proposed policy before activating it to ensure it works as expected.

## Simulate an ILM policy

Simulate a proposed policy on test objects before activating the policy and applying it to your production data. The simulation window provides a standalone environment that is safe for testing policies before they are activated and applied to data in the production environment.

### Before you begin

- You are signed in to the Grid Manager using a supported web browser.
- You have the required access permissions.
- You know the S3 bucket/object-key or the Swift container/object-name for each object you want to test.

### About this task

Carefully select the objects you want the proposed policy to test. To simulate a policy thoroughly, you should test at least one object for each filter in each rule.

For example, if a policy includes one rule to match objects in bucket A and another rule to match objects in bucket B, you must select at least one object from bucket A and one object from bucket B to test the policy thoroughly. You must also select at least one object from another bucket to test the default rule.

When simulating a policy, the following considerations apply:

- After you make changes to a policy, save the proposed policy. Then, simulate the behavior of the saved proposed policy.
- When you simulate a policy, the ILM rules in the policy filter the test objects, so you can see which rule was applied to each object. However, no object copies are made and no objects are placed. Running a simulation does not modify your data, rules, or the policy in any way.
- The Simulate proposed policy window retains the objects you tested until you select either **Clear all** or the remove icon ✕ for each object in the Simulation results list.
- Simulation returns the name of the matched rule. To determine which storage pool or erasure coding profile is in effect, select the name of the rule to go to the rule details page, where you can view the retention diagram and other details about the rule.
- If S3 versioning is enabled, you can enter the version ID for the version of the object you want to use for the simulation.

### Steps

1. Create a proposed policy.
2. Using an S3 or Swift client or the experimental S3 Console, which is available in Tenant Manager for each

tenant, ingest the objects required to test each rule.

3. On the ILM policy page, Proposed policy tab, select **Simulate**.
4. In the **Object** field, enter the S3 `bucket/object-key` or the Swift `container/object-name` for a test object. For example, `bucket-01/filename.png`.
5. Optionally, enter a version ID for the object in the **Version ID** field.
6. Select **Simulate**.
7. In the Simulation results section, confirm that each object was matched by the correct rule.

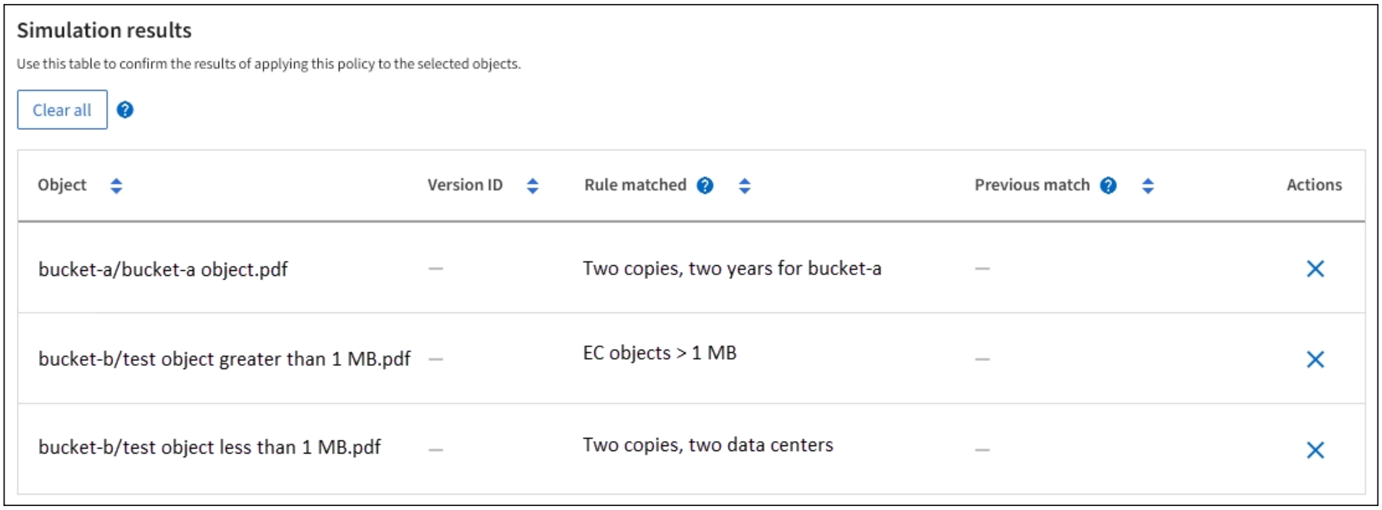
### Example 1: Verify rules when simulating a proposed ILM policy

This example describes how to verify rules when simulating a proposed policy.

In this example, the **Example ILM policy** is being simulated against the ingested objects in two buckets. The policy includes three rules, as follows:

- The first rule, **Two copies, two years for bucket-a**, applies only to objects in bucket-a.
- The second rule, **EC objects > 1 MB**, applies to all buckets but filters on objects greater than 1 MB.
- The third rule, **Two copies, two data centers**, is the default rule. It does not include any filters and does not use the Noncurrent reference time.

After simulating the policy, confirm that each object was matched by the correct rule.



Simulation results

Use this table to confirm the results of applying this policy to the selected objects.

| Object | Version ID | Rule matched | Previous match | Actions |
|---|---|---|---|---|
| bucket-a/bucket-a object.pdf | — | Two copies, two years for bucket-a | — | × |
| bucket-b/test object greater than 1 MB.pdf | — | EC objects > 1 MB | — | × |
| bucket-b/test object less than 1 MB.pdf | — | Two copies, two data centers | — | × |

In this example:

- `bucket-a/bucket-a object.pdf` correctly matched the first rule, which filters on objects in `bucket-a`.
- `bucket-b/test object greater than 1 MB.pdf` is in `bucket-b`, so it did not match the first rule. Instead, it was correctly matched by the second rule, which filters on objects greater than 1 MB.
- `bucket-b/test object less than 1 MB.pdf` did not match the filters in the first two rules, so it will be placed by the default rule, which includes no filters.

### Example 2: Reorder rules when simulating a proposed ILM policy

This example shows how you can reorder rules to change the results when simulating a policy.

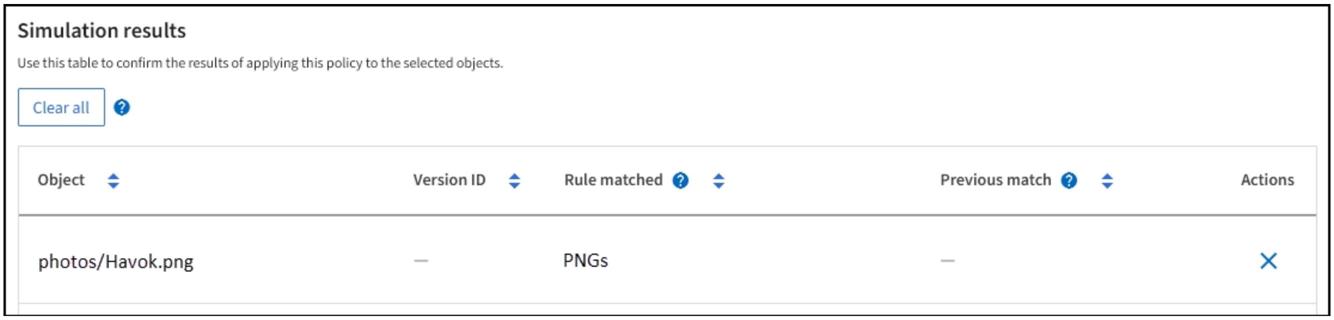
In this example, the **Demo** policy is being simulated. This policy, which is intended to find objects that have series=x-men user metadata, includes three rules, as follows:

- The first rule, **PNGs**, filters for key names that end in `.png`.
- The second rule, **X-men**, applies only to objects for Tenant A and filters for `series=x-men` user metadata.
- The last rule, **Two copies two data centers**, is the default rule, which matches any objects that don’t match the first two rules.

**Steps**

1. After adding the rules and saving the policy, select **Simulate**.
2. In the **Object** field, enter the S3 bucket/object-key or the Swift container/object-name for a test object, and select **Simulate**.

   The Simulation results appear, showing that the `Havok.png` object was matched by the **PNGs** rule.

   Simulation results

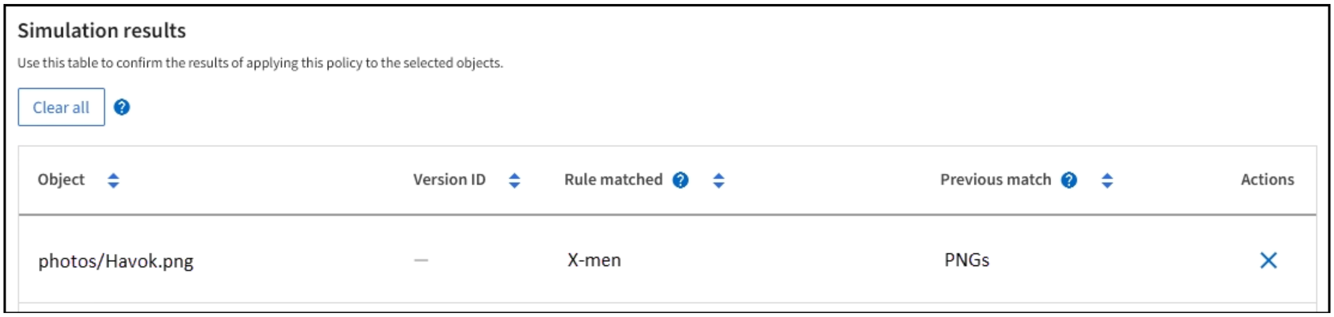
   Use this table to confirm the results of applying this policy to the selected objects.

   | Object | Version ID | Rule matched | Previous match | Actions |
   |---|---|---|---|---|
   | photos/Havok.png | — | PNGs | — | ✕ |

   However, `Havok.png` was meant to test the **X-men** rule.

3. To resolve the issue, reorder the rules.
   a. Select **Finish** to close the Simulate ILM Policy window.
   b. Select **Actions** > **Edit** to edit the policy.
   c. Drag the **X-men** rule to the top of the list.
   d. Select **Save**.
4. Select **Simulate**.

   The objects you previously tested are re-evaluated against the updated policy, and the new simulation results are shown. In the example, the Rule matched column shows that the `Havok.png` object now matches the X-men metadata rule, as expected. The Previous match column shows that the PNGs rule matched the object in the previous simulation.

   Simulation results

   Use this table to confirm the results of applying this policy to the selected objects.

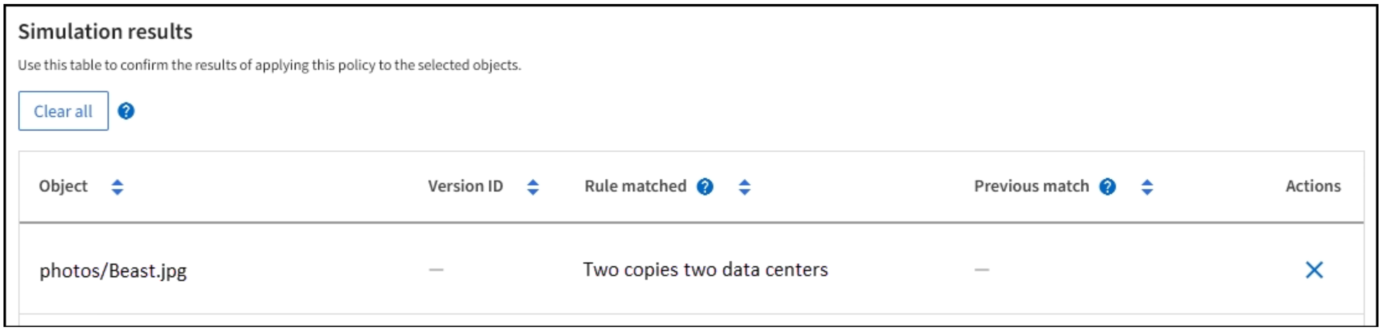
   | Object | Version ID | Rule matched | Previous match | Actions |
   |---|---|---|---|---|
   | photos/Havok.png | — | X-men | PNGs | ✕ |

If you stay on the Proposed policy tab, you can re-simulate a policy after making changes without needing to re-enter the names of the test objects.

### Example 3: Correct a rule when simulating a proposed ILM policy

This example shows how to simulate a policy, correct a rule in the policy, and continue the simulation.

In this example, the **Demo** policy is being simulated. This policy is intended to find objects that have `series=x-men` user metadata. However, unexpected results occurred when simulating this policy against the `Beast.jpg` object. Instead of matching the X-men metadata rule, the object matched the default rule, Two copies two data centers.



Simulation results

Use this table to confirm the results of applying this policy to the selected objects.

| Object | Version ID | Rule matched | Previous match | Actions |
|---|---|---|---|---|
| photos/Beast.jpg | — | Two copies two data centers | — | ✕ |

When a test object is not matched by the expected rule in the policy, you must examine each rule in the policy and correct any errors.

**Steps**

1. Select **Finish** to close the Simulate policy dialog. On the Proposed policy tab, select **Retention diagram**. Then select **Expand all** or **View details** for each rule as needed.
2. Review the rule's tenant account, reference time, and filtering criteria.

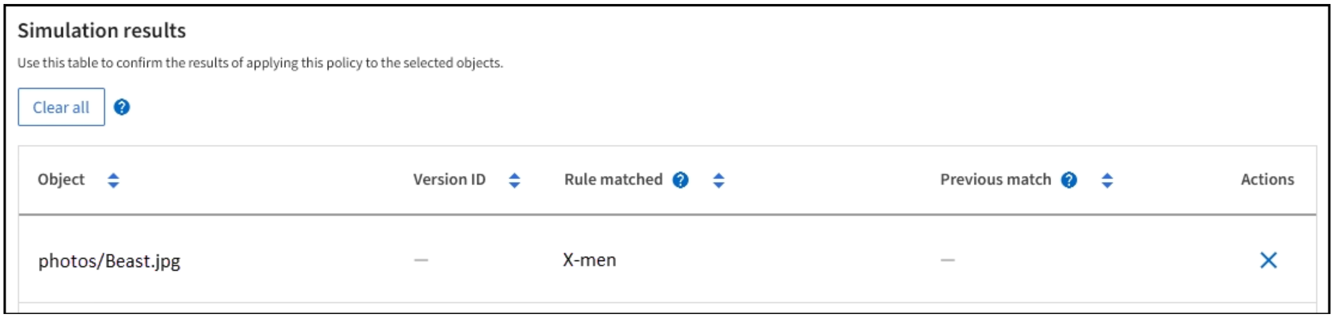
   As an example, suppose the metadata for the X-men rule was entered as “x-men01” instead of “x-men.”

3. To resolve the error, correct the rule as follows:
   - If the rule is part of the proposed policy, you can either clone the rule or remove the rule from the policy and then edit it.
   - If the rule is part of the active policy, you must clone the rule. You can’t edit or remove a rule from the active policy.

| Option | Steps |
|---|---|
| Clone the rule | a. Select **ILM** > **Rules**.<br>b. Select the incorrect rule, and select **Clone**.<br>c. Enter a name for the new rule, then change the incorrect information and select **Create**.<br>d. Select **ILM** > **Policies** > **Proposed policy**.<br>e. Select **Actions** > **Edit**.<br>f. Select **Select rules**, then select **Continue** to accept the same default rule.<br>g. In the Select other rules step, select the checkbox for the new rule, clear the checkbox for the original rule, and select **Select**.<br>h. If necessary, reorder the rules by dragging the new rule to the correct location.<br>i. Select **Save**. |
| Edit the rule | a. Select **ILM** > **Policies** > **Proposed policy** and remove the rule you want to edit.<br>b. Select **ILM** > **Rules**.<br>c. Select the rule you want to edit and select **Edit**. Or select the checkbox for the rule and select **Actions** > **Edit**.<br>d. Change the incorrect information for each part of the wizard, then select **Update**.<br>e. Select **ILM** > **Policies** > **Proposed policy**.<br>f. Select **Actions** > **Edit**.<br>g. Select **Select rules**, then select **Continue** to accept the same default rule.<br>h. In the Select other rules dialog box, select the checkbox for the corrected rule, select **Select**, then select **Save**.<br>i. Drag the rows for the non-default rules to determine the order in which these rules will be evaluated. |

4. Perform the simulation again.

   In this example, the corrected X-men rule now matches the `Beast.jpg` object based on the `series=x-men` user metadata, as expected.



Simulation results

Use this table to confirm the results of applying this policy to the selected objects.

| Object | Version ID | Rule matched | Previous match | Actions |
|---|---|---|---|---|
| photos/Beast.jpg | — | X-men | — | ✕ |

## Activate the ILM policy

After you add ILM rules to a proposed ILM policy, simulate the policy, and confirm it behaves as you expect, you are ready to activate the proposed policy.

### Before you begin

- You are signed in to the Grid Manager using a supported web browser.
- You have specific access permissions.
- You have saved and simulated the proposed ILM policy.

> Errors in an ILM policy can cause unrecoverable data loss. Carefully review and simulate the policy before activating it to confirm that it will work as intended.
> When you activate a new ILM policy, StorageGRID uses it to manage all objects, including existing objects and newly ingested objects. Before activating a new ILM policy, review any changes to the placement of existing replicated and erasure-coded objects. Changing an existing object's location might result in temporary resource issues when the new placements are evaluated and implemented.

### About this task

When you activate an ILM policy, the system distributes the new policy to all nodes. However, the new active policy might not actually take effect until all grid nodes are available to receive the new policy. In some cases, the system waits to implement a new active policy to ensure that grid objects aren't accidentally removed.

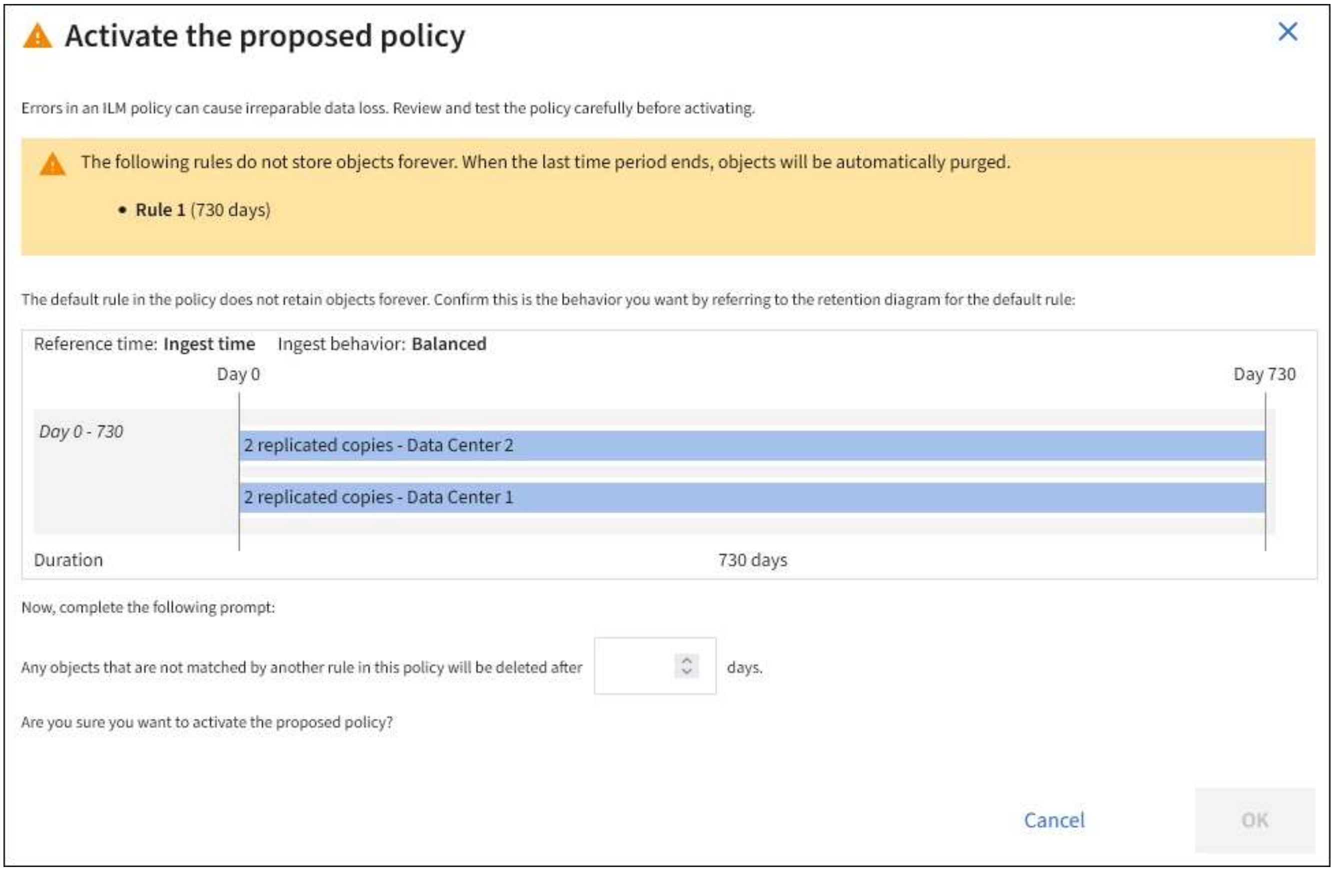
- If you make policy changes that increase data redundancy or durability, those changes are implemented immediately. For example, if you activate a new policy that includes a three-copies rule instead of a two-copies rule, that policy will be implemented right away because it increases data redundancy.
- If you make policy changes that could decrease data redundancy or durability, those changes will not be implemented until all grid nodes are available. For example, if you activate a new policy that uses a two-copies rule instead of a three-copies rule, the new policy will appear in the Active policy tab but it will not take effect until all nodes are online and available.

### Steps

1. When you are ready to activate a proposed policy, select **ILM policies** > **Proposed policy**, then select **Activate**.

   A warning message is displayed, prompting you to confirm that you want to activate the proposed policy.

   A prompt appears in the warning message if the default rule does not retain objects forever. In this example, the retention diagram shows that the default rule will delete objects after 730 days (2 years). You must type **730** in the text box to acknowledge that any objects not matched by another rule in the policy will be removed from StorageGRID after 730 days.



Activate the proposed policy

Errors in an ILM policy can cause irreparable data loss. Review and test the policy carefully before activating.

The following rules do not store objects forever. When the last time period ends, objects will be automatically purged.

- **Rule 1** (730 days)

The default rule in the policy does not retain objects forever. Confirm this is the behavior you want by referring to the retention diagram for the default rule:

Reference time: **Ingest time** Ingest behavior: **Balanced**

Day 0
Day 730
Day 0 - 730
2 replicated copies - Data Center 2
2 replicated copies - Data Center 1
Duration
730 days

Now, complete the following prompt:

Any objects that are not matched by another rule in this policy will be deleted after ______ days.

Are you sure you want to activate the proposed policy?

2. Select **OK**.

### Result

When a new ILM policy has been activated:

- The policy is shown on the Active policy tab. The Start date entry indicates the date and time the policy was activated.
- The previously active policy appears in the Policy history tab. The Start date and End date entries indicate when the policy became active and when it was no longer in effect.

### Related information

Example 6: Changing an ILM policy

## Verify an ILM policy with object metadata lookup

After you have activated an ILM policy, you should ingest representative test objects into the StorageGRID system. You should then perform an object metadata lookup to confirm that copies are being made as intended and placed in the correct locations.

### Before you begin

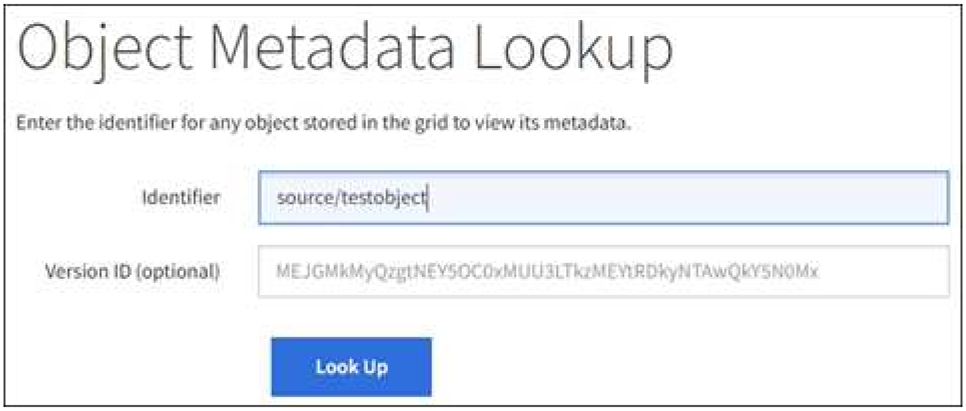
- You have an object identifier, which can be one of:
  - **UUID**: The object's Universally Unique Identifier. Enter the UUID in all uppercase.
  - **CBID**: The object's unique identifier within StorageGRID. You can obtain an object's CBID from the audit log. Enter the CBID in all uppercase.
  - **S3 bucket and object key**: When an object is ingested through the S3 interface, the client application

uses a bucket and object key combination to store and identify the object. If the S3 bucket is versioned and you want to look up a specific version of an S3 object using the bucket and object key, you have the **version ID**.

   - **Swift container and object name**: When an object is ingested through the Swift interface, the client application uses a container and object name combination to store and identify the object.

**Steps**

1. Ingest the object.
2. Select **ILM** > **Object metadata lookup**.
3. Type the object's identifier in the **Identifier** field. You can enter a UUID, CBID, S3 bucket/object-key, or Swift container/object-name.
4. Optionally, enter a version ID for the object (S3 only).



Object Metadata Lookup

Enter the identifier for any object stored in the grid to view its metadata.

Identifier: source/testobject

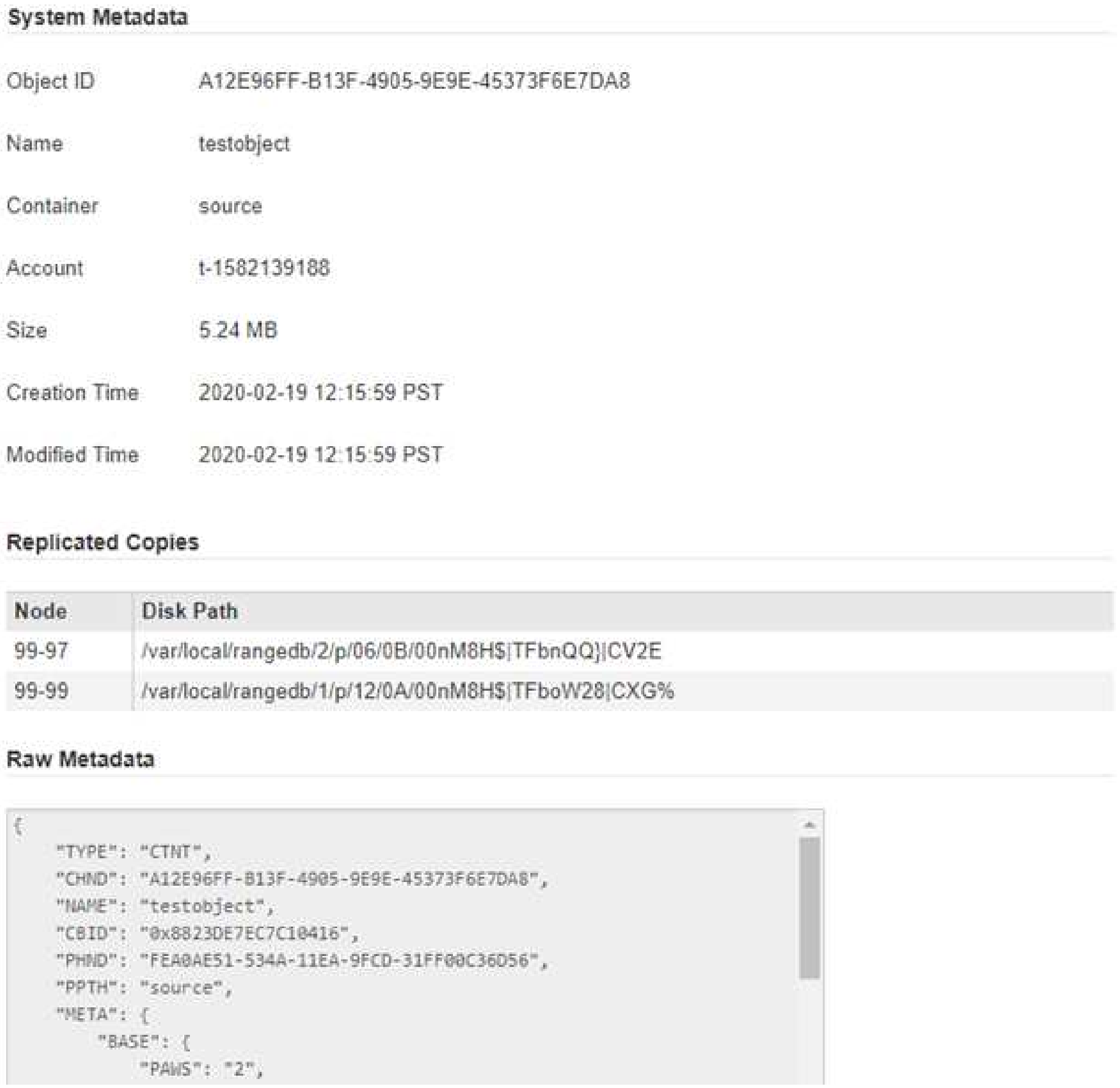
Version ID (optional): MEJGMkMyQzgtNEY5OC0xMUU3LTkzMEYtRDkyNTAwQkY5N0Mx

5. Select **Look Up**.

   The object metadata lookup results appear. This page lists the following types of information:

   - System metadata, including the object ID (UUID), the object name, the name of the container, the tenant account name or ID, the logical size of the object, the date and time the object was first created, and the date and time the object was last modified.
   - Any custom user metadata key-value pairs associated with the object.
   - For S3 objects, any object tag key-value pairs associated with the object.
   - For replicated object copies, the current storage location of each copy.
   - For erasure-coded object copies, the current storage location of each fragment.
   - For object copies in a Cloud Storage Pool, the location of the object, including the name of the external bucket and the object's unique identifier.
   - For segmented objects and multipart objects, a list of object segments including segment identifiers and data sizes. For objects with more than 100 segments, only the first 100 segments are shown.
   - All object metadata in the unprocessed, internal storage format. This raw metadata includes internal system metadata that is not guaranteed to persist from release to release.

     The following example shows the object metadata lookup results for an S3 test object that is stored as two replicated copies.



**System Metadata**

| | |
|---|---|
| Object ID | A12E96FF-B13F-4905-9E9E-45373F6E7DA8 |
| Name | testobject |
| Container | source |
| Account | t-1582139188 |
| Size | 5.24 MB |
| Creation Time | 2020-02-19 12:15:59 PST |
| Modified Time | 2020-02-19 12:15:59 PST |

**Replicated Copies**

| Node | Disk Path |
|---|---|
| 99-97 | /var/local/rangedb/2/p/06/0B/00nM8H$\|TFbnQQ}\|CV2E |
| 99-99 | /var/local/rangedb/1/p/12/0A/00nM8H$\|TFboW28\|CXG% |

**Raw Metadata**

```
{
    "TYPE": "CTNT",
    "CHND": "A12E96FF-B13F-4905-9E9E-45373F6E7DA8",
    "NAME": "testobject",
    "CBID": "0x8823DE7EC7C10416",
    "PHND": "FEA0AE51-534A-11EA-9FCD-31FF00C36D56",
    "PPTH": "source",
    "META": {
        "BASE": {
            "PAWS": "2",
```

6. Confirm that the object is stored in the correct location or locations and that it is the correct type of copy.

> If the Audit option is enabled, you can also monitor the audit log for the ORLM Object Rules Met message. The ORLM audit message can provide you with more information about the status of the ILM evaluation process, but it can't give you information about the correctness of the object data's placement or the completeness of the ILM policy. You must evaluate this yourself. For details, see Review audit logs.

**Related information**

- Use S3 REST API
- Use Swift REST API

# Work with ILM policies and ILM rules

As your storage requirements change, you might need put a different policy in place or modify the ILM rules associated with the policy. You can view ILM metrics to determine system performance.

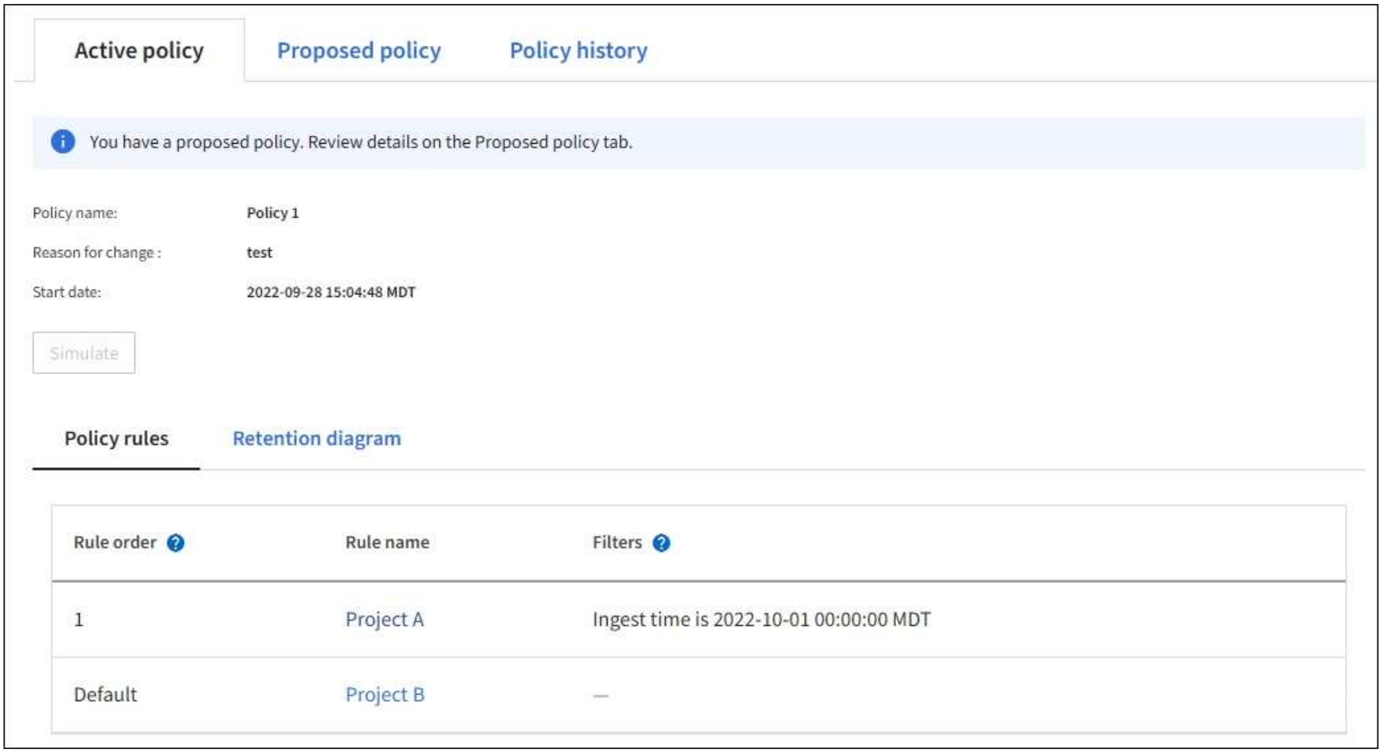
**Before you begin**

- You are signed in to the Grid Manager using a supported web browser.
- You have specific access permissions.

## View ILM policies

To view active, proposed, and historical ILM policies:

1. Select **ILM** > **Policies**.
2. As needed, select **Active policy**, **Proposed policy**, or **Policy history** to view the details for each. In each tab, you can select **Policy rules** and **Retention diagram**.



You have a proposed policy. Review details on the Proposed policy tab.

Policy name: Policy 1

Reason for change : test

Start date: 2022-09-28 15:04:48 MDT

| Rule order | Rule name | Filters |
|---|---|---|
| 1 | Project A | Ingest time is 2022-10-01 00:00:00 MDT |
| Default | Project B | — |

## Clone a historical ILM policy

To clone a historical ILM policy:

1. Select **ILM** > **Policies** > **Policy history**.
2. Remove the proposed policy if one exists.
3. Select the radio button for the policy you want to clone, then select **Clone historical policy**.
4. Complete the required details by following the instructions in Create proposed ILM policy.

An ILM policy that has been incorrectly configured can result in unrecoverable data loss. Before activating an ILM policy, carefully review the ILM policy and its ILM rules, and then simulate the ILM policy. Always confirm that the ILM policy will work as intended.

## Remove the proposed ILM policy

To remove the proposed policy:

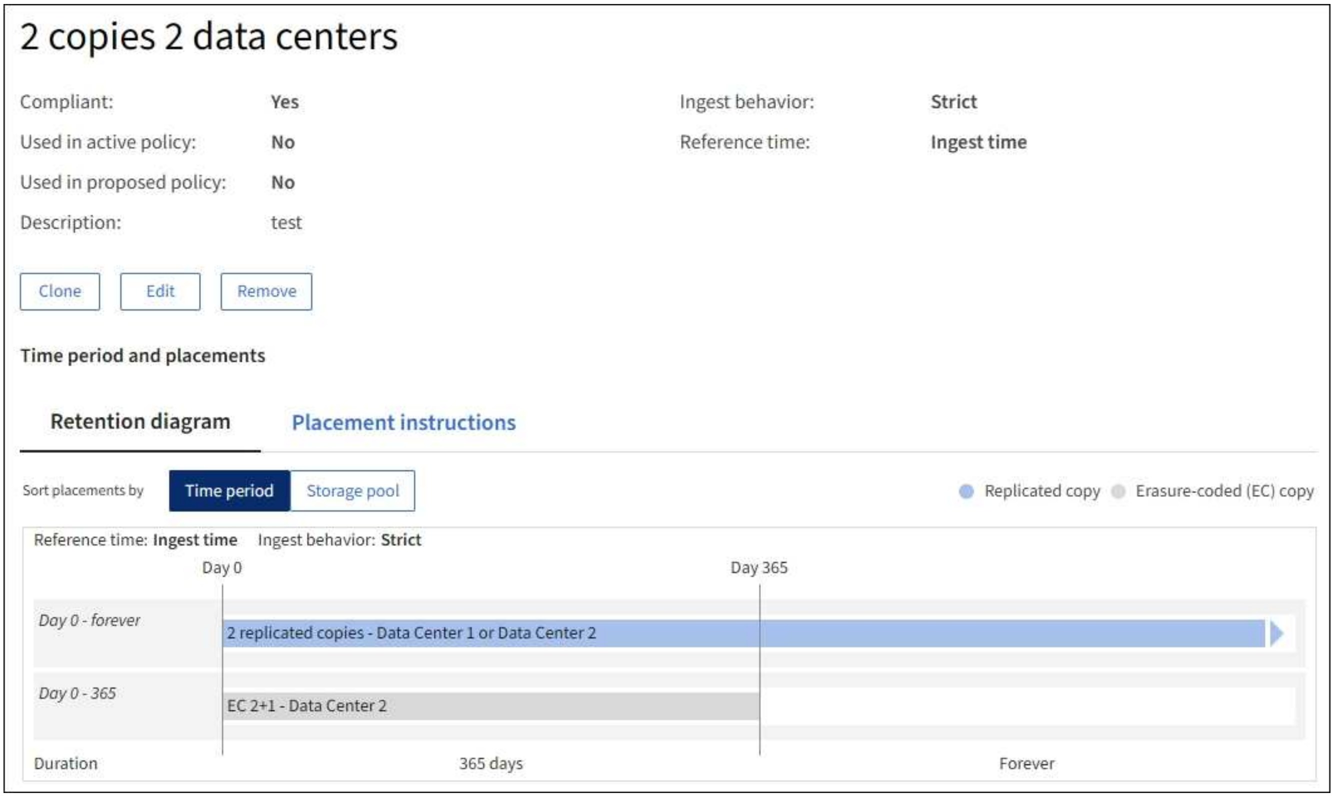
1. Select **ILM** > **Policies** > **Proposed policy**.
2. Select **Actions** > **Remove**.

The proposed policy and the Proposed policy tab are removed.

## View ILM rule details

To view the details for an ILM rule, including the retention diagram and placement instructions for the rule:

1. Select **ILM** > **Rules**.
2. Select the rule whose details you want to view. Example:



2 copies 2 data centers

Compliant: Yes
Used in active policy: No
Used in proposed policy: No
Description: test
Ingest behavior: Strict
Reference time: Ingest time

Time period and placements

Reference time: Ingest time   Ingest behavior: Strict

Additionally, you can use the details page to clone, edit, or remove a rule.

## Clone an ILM rule

You can’t edit a rule if it is being used in the proposed ILM policy or the active ILM policy. Instead, you can clone a rule and make any required changes to the cloned copy. Then, if required, you can remove the original rule from the proposed policy and replace it with the modified version. You can’t clone an ILM rule if it was created using StorageGRID version 10.2 or earlier.

Before adding a cloned rule to the active ILM policy, be aware that a change to an object’s placement instructions might cause an increased load on the system.

### Steps

1. Select **ILM** > **Rules**.
2. Select the checkbox for the rule you want to clone, then select **Clone**. Alternatively, select the rule name, then select **Clone** from the rule details page.

3. Update the cloned rule by following the steps for editing an ILM rule and using advanced filters in ILM rules.

   When cloning an ILM rule, you must enter a new name.

## Edit an ILM rule

You might need to edit an ILM rule to change a filter or placement instruction.

You can't edit a rule if it is being used in the active ILM policy or the proposed ILM policy. Instead, you can clone these rules and make any required changes to the cloned copy. You also can't edit the system-provided rule, Make 2 Copies.

> ⓘ Before adding an edited rule to the active ILM policy, be aware that a change to an object's placement instructions might cause an increased load on the system.

**Steps**

1. Select **ILM** > **Rules**.
2. Confirm that the rule you want to edit is not used in the active ILM policy or the proposed ILM policy.
3. If the rule you want to edit is not in use, select the checkbox for the rule and select **Actions** > **Edit**. Alternatively, select the name of the rule, then select **Edit** on the rule details page.
4. Complete the pages of the Edit ILM rule wizard. As necessary, follow the steps for creating an ILM rule and using advanced filters in ILM rules.

   When editing an ILM rule, you can't change its name.

   > ⓘ If you edit a rule that is used in a historical policy, the ⊗ icon appears for the rule when you view the policy, which indicates that the rule has become a historical rule.

## Remove an ILM rule

To keep the list of current ILM rules manageable, remove any ILM rules that you aren't likely to use.

**Steps**

To remove an ILM rule that is currently used in the active policy or in the proposed policy:

1. Clone the active policy or edit the proposed policy.
2. Remove the ILM rule from the policy.
3. Save, simulate, and activate the new policy to make sure objects are protected as expected.

To remove an ILM rule that is not currently used:

1. Select **ILM** > **Rules**.
2. Confirm that the rule you want to remove is not used in the active policy or the proposed policy.
3. If the rule you want to remove is not in use, select the rule and select **Remove**. You can select multiple rules and remove all of them at the same time.
4. Select **Yes** to confirm that you want to remove the ILM rule.

The ILM rule is removed.

> ⓘ If you remove a rule that is used in a historical policy, the ⊗ icon appears for the rule when you view the policy, which indicates that the rule has become a historical rule.

## View ILM metrics

You can view metrics for ILM, such as the number of objects in the queue and the evaluation rate. You can monitor these metrics to determine system performance. A large queue or evaluation rate might indicate that the system is not able to keep up with the ingest rate, the load from the client applications is excessive, or that some abnormal condition exists.

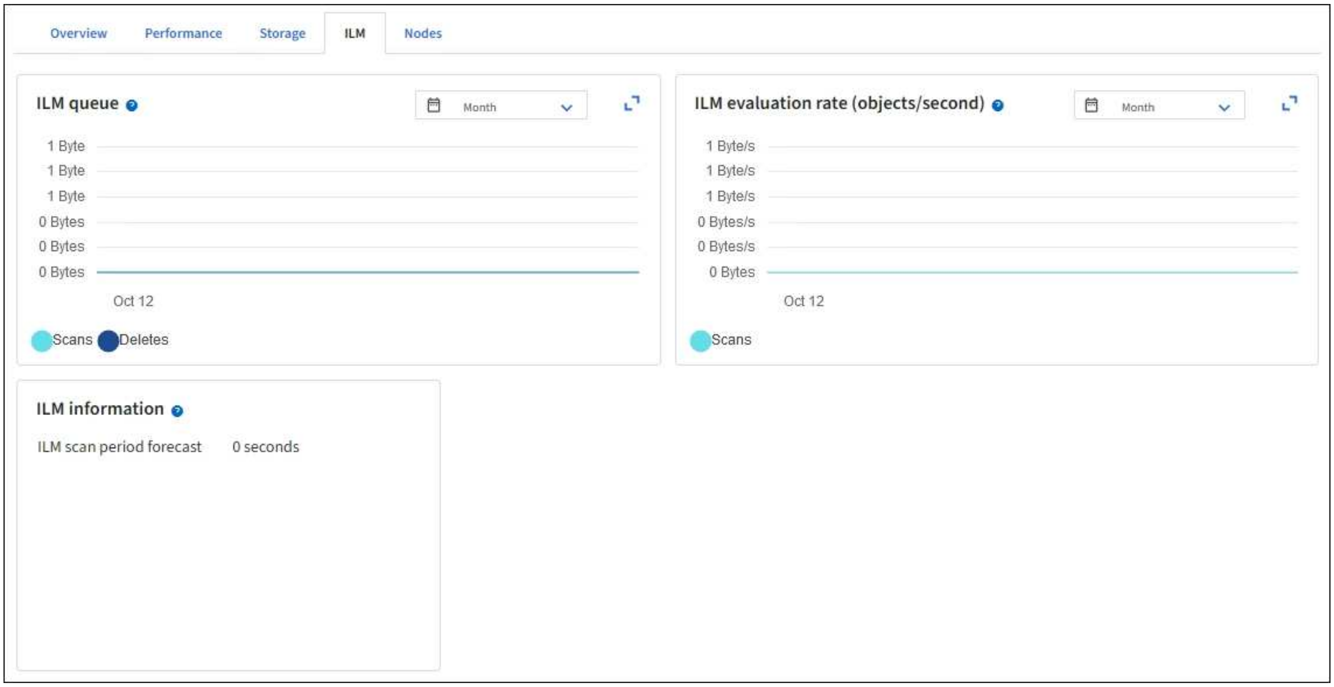
**Steps**

1. Select **Dashboard** > **ILM**.

   > ⓘ Because the dashboard can be customized, the ILM tab might not be available.

2. Monitor the metrics on the ILM tab.

   You can select the question mark ? to see a description of the items on the ILM tab.



ILM scan period forecast 0 seconds

# Use S3 Object Lock

## Manage objects with S3 Object Lock

As a grid administrator, you can enable S3 Object Lock for your StorageGRID system and implement a compliant ILM policy to help ensure that objects in specific S3 buckets aren’t deleted or overwritten for a specified amount of time.

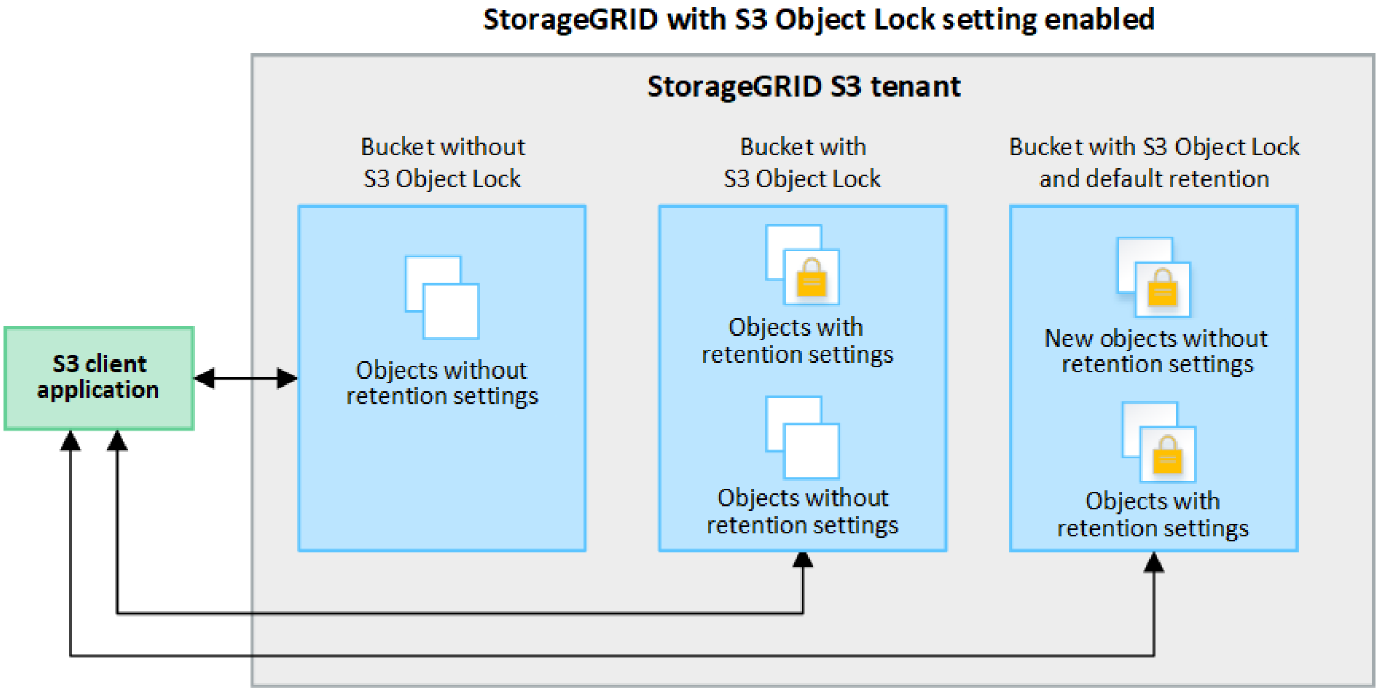
### What is S3 Object Lock?

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, bucket versioning is required and is enabled automatically.

If a bucket has S3 Object Lock enabled, S3 client applications can optionally specify retention settings for any object version saved to that bucket.

In addition, a bucket that has S3 Object Lock enabled can optionally have a default retention mode and retention period. The default settings apply only to objects that are added to the bucket without their own retention settings.


StorageGRID with S3 Object Lock setting enabled
StorageGRID S3 tenant
Bucket without S3 Object Lock
Bucket with S3 Object Lock
Bucket with S3 Object Lock and default retention
S3 client application
Objects without retention settings
Objects with retention settings
Objects without retention settings
New objects without retention settings
Objects with retention settings

#### Retention modes

The StorageGRID S3 Object Lock feature supports two retention modes to apply different levels of protection to objects. These modes are equivalent to the Amazon S3 retention modes.

- In compliance mode:
  - The object can’t be deleted until its retain-until-date is reached.
  - The object’s retain-until-date can be increased, but it can’t be decreased.
  - The object’s retain-until-date can’t be removed until that date is reached.
- In governance mode:
  - Users with special permission can use a bypass header in requests to modify certain retention settings.
  - These users can delete an object version before its retain-until-date is reached.
  - These users can increase, decrease, or remove an object’s retain-until-date.

#### Retention settings for object versions

If a bucket is created with S3 Object Lock enabled, users can use the S3 client application to optionally specify the following retention settings for each object that is added to the bucket:

- **Retention mode**: Either compliance or governance.
- **Retain-until-date**: If an object version's retain-until-date is in the future, the object can be retrieved, but it can't be deleted.
- **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.

  > ⓘ If an object is under a legal hold, no one can delete the object, regardless of its retention mode.

  For details on the object settings, see Use S3 REST API to configure S3 Object Lock.

#### Default retention setting for buckets

If a bucket is created with S3 Object Lock enabled, users can optionally specify the following default settings for the bucket:

- **Default retention mode**: Either compliance or governance.
- **Default retention period**: How long new object versions added to this bucket should be retained, starting from the day they are added.

The default bucket settings apply only to new objects that don't have their own retention settings. Existing bucket objects aren't affected when you add or change these default settings.

See Create an S3 bucket and Update S3 Object Lock default retention.

### Comparing S3 Object Lock to legacy Compliance

The S3 Object Lock replaces the Compliance feature that was available in earlier StorageGRID versions. Because the S3 Object Lock feature conforms to Amazon S3 requirements, it deprecates the proprietary StorageGRID Compliance feature, which is now referred to as "legacy Compliance."

> ⓘ The global Compliance setting is deprecated. If you enabled this setting using a previous version of StorageGRID, the S3 Object Lock setting is enabled automatically. You can continue to use StorageGRID to manage the settings of existing compliant buckets; however, you can't create new compliant buckets. For details, see NetApp Knowledge Base: How to manage legacy Compliant buckets in StorageGRID 11.5.

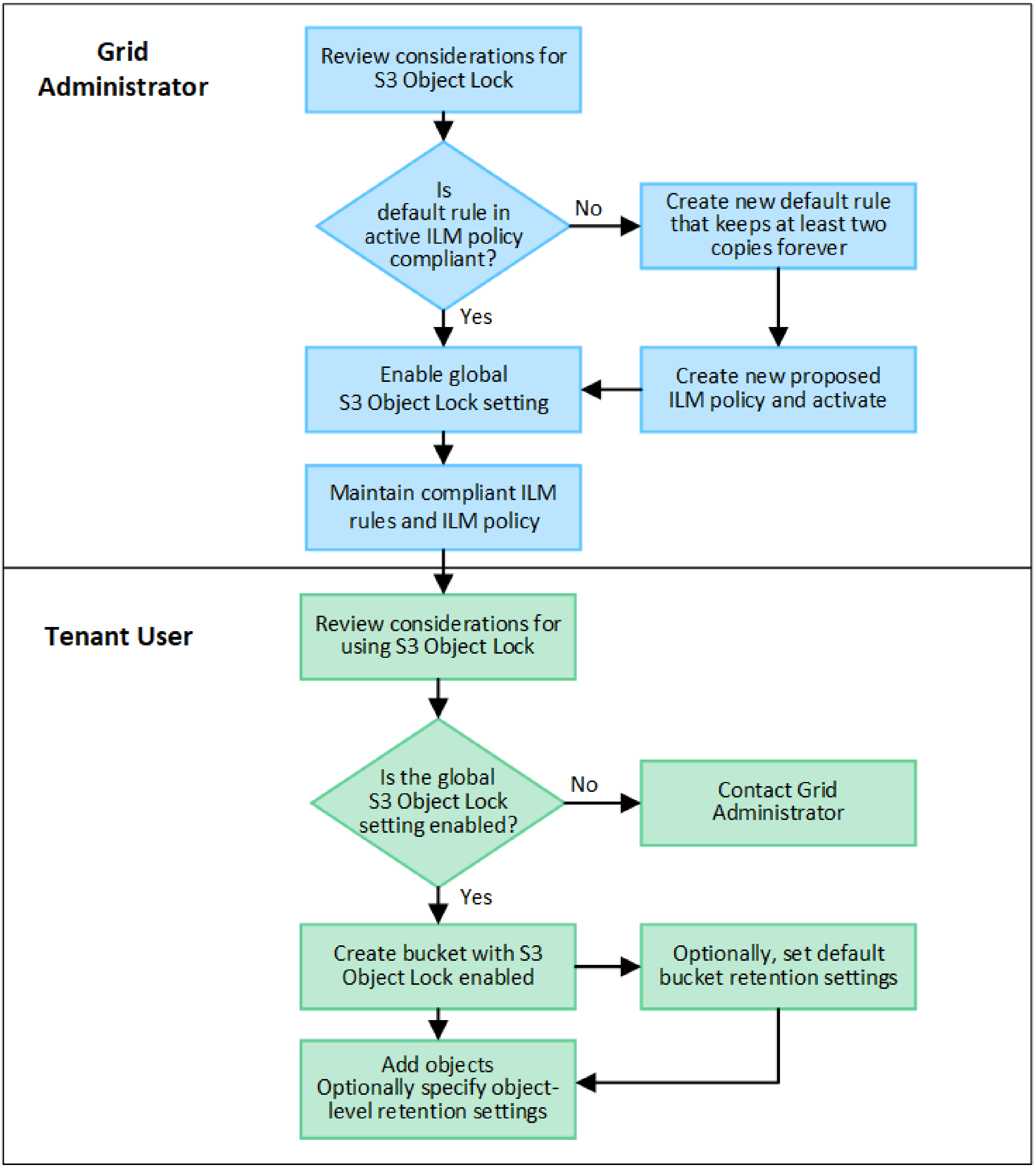
If you used the legacy Compliance feature in a previous version of StorageGRID, refer to the following table to learn how it compares to the S3 Object Lock feature in StorageGRID.

|  | **S3 Object Lock** | **Compliance (legacy)** |
|---|---|---|
| How is the feature enabled globally? | From the Grid Manager, select **CONFIGURATION** > **System** > **S3 Object Lock**. | No longer supported. |
| How is the feature enabled for a bucket? | Users must enable S3 Object Lock when creating a new bucket using the Tenant Manager, the Tenant Management API, or the S3 REST API. | No longer supported. |
| Is bucket versioning supported? | Yes. Bucket versioning is required and is enabled automatically when S3 Object Lock is enabled for the bucket. | No. |
| How is object retention set? | Users can set a retain-until-date for each object version, or they can set a default retention period for each bucket. | Users must set a retention period for the entire bucket. The retention period applies to all objects in the bucket. |
| Can the retention period be changed? | • In compliance mode, the retain-until-date for an object version can be increased but never decreased.<br>• In governance mode, users with special permissions can decrease or even remove an object's retention settings. | A bucket's retention period can be increased but never decreased. |
| Where is legal hold controlled? | Users can place a legal hold or lift a legal hold for any object version in the bucket. | A legal hold is placed on the bucket and affects all objects in the bucket. |
| When can objects be deleted? | • In compliance mode, an object version can be deleted after the retain-until-date is reached, assuming the object is not under legal hold.<br>• In governance mode, users with special permissions can delete an object before its retain-until-date is reached, assuming the object is not under legal hold. | An object can be deleted after the retention period expires, assuming the bucket is not under legal hold. Objects can be deleted automatically or manually. |
| Is bucket lifecycle configuration supported? | Yes | No |

### Workflow for S3 Object Lock

As a grid administrator, you must coordinate closely with tenant users to ensure that the objects are protected in a manner that satisfies their retention requirements.

The workflow diagram shows the high-level steps for using S3 Object Lock. These steps are performed by the grid administrator and by tenant users.


Grid Administrator
Review considerations for S3 Object Lock
Is default rule in active ILM policy compliant?
No
Create new default rule that keeps at least two copies forever
Yes
Enable global S3 Object Lock setting
Create new proposed ILM policy and activate
Maintain compliant ILM rules and ILM policy
Tenant User
Review considerations for using S3 Object Lock
Is the global S3 Object Lock setting enabled?
No
Contact Grid Administrator
Yes
Create bucket with S3 Object Lock enabled
Optionally, set default bucket retention settings
Add objects Optionally specify object-level retention settings

### Grid admin tasks

As the workflow diagram shows, a grid administrator must perform two high-level tasks before S3 tenant users can use S3 Object Lock:

1. Create at least one compliant ILM rule and make that rule the default rule in the active ILM policy.
2. Enable the global S3 Object Lock setting for the entire StorageGRID system.

### Tenant user tasks

After the global S3 Object Lock setting has been enabled, tenants can perform these tasks:

1. Create buckets that have S3 Object Lock enabled.
2. Optionally, specify default retention settings for the bucket. Any default bucket settings are applied only to new objects that don't have their own retention settings.
3. Add objects to those buckets and optionally specify object-level retention periods and legal hold settings.
4. As required, update default retention for the bucket or update the retention period or the legal hold setting for an individual object.

## Requirements for S3 Object Lock

You must review the requirements for enabling the global S3 Object Lock setting, the requirements for creating compliant ILM rules and ILM policies, and the restrictions StorageGRID places on buckets and objects that use S3 Object Lock.

### Requirements for using the global S3 Object Lock setting

- You must enable the global S3 Object Lock setting using the Grid Manager or the Grid Management API before any S3 tenant can create a bucket with S3 Object Lock enabled.
- Enabling the global S3 Object Lock setting allows all S3 tenant accounts to create buckets with S3 Object Lock enabled.
- After you enable the global S3 Object Lock setting, you can't disable the setting.
- You can't enable the global S3 Object Lock unless the default rule in the active ILM policy is *compliant* (that is, the default rule must comply with the requirements of buckets with S3 Object Lock enabled).
- When the global S3 Object Lock setting is enabled, you can't create a new proposed ILM policy or activate an existing proposed ILM policy unless the default rule in the policy is compliant. After the global S3 Object Lock setting has been enabled, the ILM rules and ILM policies pages indicate which ILM rules are compliant.

### Requirements for compliant ILM rules

If you want to enable the global S3 Object Lock setting, you must ensure that the default rule in your active ILM policy is compliant. A compliant rule satisfies the requirements of both buckets with S3 Object Lock enabled and any existing buckets that have legacy Compliance enabled:

- It must create at least two replicated object copies or one erasure-coded copy.
- These copies must exist on Storage Nodes for the entire duration of each line in the placement instructions.
- Object copies can't be saved in a Cloud Storage Pool.

- Object copies can’t be saved on Archive Nodes.
- At least one line of the placement instructions must start at day 0, using **Ingest time** as the reference time.
- At least one line of the placement instructions must be “forever.”

### Requirements for active and proposed ILM policies

When the global S3 Object Lock setting is enabled, active and proposed ILM policies can include both compliant and non-compliant rules.

- The default rule in the active or any proposed ILM policy must be compliant.
- Non-compliant rules only apply to objects in buckets that don’t have S3 Object Lock enabled or that don’t have the legacy Compliance feature enabled.
- Compliant rules can apply to objects in any bucket; S3 Object Lock or legacy Compliance does not need to be enabled for the bucket.

A compliant ILM policy might include these three rules:

1. A compliant rule that creates erasure-coded copies of the objects in a specific bucket with S3 Object Lock enabled. The EC copies are stored on Storage Nodes from day 0 to forever.
2. A non-compliant rule that creates two replicated object copies on Storage Nodes for a year and then moves one object copy to Archive Nodes and stores that copy forever. This rule only applies to buckets that don’t have S3 Object Lock or legacy Compliance enabled because it stores only one object copy forever and it uses Archive Nodes.
3. A default, compliant rule that creates two replicated object copies on Storage Nodes from day 0 to forever. This rule applies to any object in any bucket that was not filtered out by the first two rules.

### 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.
- If you plan to use S3 Object Lock, you must enable S3 Object Lock when you create the bucket. You can’t enable S3 Object Lock for an existing bucket.
- When S3 Object Lock is enabled for a bucket, StorageGRID automatically enables versioning for that bucket. You can’t disable S3 Object Lock or suspend versioning for the bucket.
- Optionally, you can specify a default retention mode and retention period for each bucket using the Tenant Manager, the Tenant Management API, or the S3 REST API. The bucket’s default retention settings apply only to new objects added to the bucket that don’t have their own retention settings. You can override these default settings by specifying a retention mode and retain-until-date for each object version when it is uploaded.
- Bucket lifecycle configuration is supported for buckets with S3 Object Lock enabled.
- CloudMirror replication is not supported for buckets with S3 Object Lock enabled.

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

- To protect an object version, you can specify default retention settings for the bucket, or you can specify retention settings for each object version. Object-level retention settings can be specified using the S3 client application or the S3 REST API.
- 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 these stages:

1. **Object ingest**

   When an object version is added to bucket that has S3 Object Lock enabled, retention settings are applied as follows:

   - If retention settings are specified for the object, the object-level settings are applied. Any default bucket settings are ignored.
   - If no retention settings are specified for the object, the default bucket settings are applied, if they exist.
   - If no retention settings are specified for the object or the bucket, the object is not protected by S3 Object Lock.

   If retention settings are applied, both the object and any S3 user-defined metadata are protected.

2. **Object retention and deletion**

   Multiple copies of each protected object are stored by StorageGRID for the specified retention period. The exact number and type of object copies and the storage locations are determined by the compliant rules in the active ILM policy. Whether a protected object can be deleted before its retain-until-date is reached depends on its retention mode.

   - If an object is under a legal hold, no one can delete the object, regardless of its retention mode.

**Related information**

- Create an S3 bucket
- Update S3 Object Lock default retention
- Use S3 REST API to configure S3 Object Lock
- Example 7: Compliant ILM policy for S3 Object Lock

## Enable S3 Object Lock globally

If an S3 tenant account needs to comply with regulatory requirements when saving object data, you must enable S3 Object Lock for your entire StorageGRID system. Enabling the global S3 Object Lock setting allows any S3 tenant user to create and manage buckets and objects with S3 Object Lock.

**Before you begin**

- You have the Root access permission.
- You are signed in to the Grid Manager using a supported web browser.
- You have reviewed the S3 Object Lock workflow, and you understand the considerations.
- You have confirmed that the default rule in the active ILM policy is compliant. See Create a default ILM rule for details.

### About this task

A grid administrator must enable the global S3 Object Lock setting to allow tenant users to create new buckets that have S3 Object Lock enabled. After this setting is enabled, it can’t be disabled.

> The global Compliance setting is deprecated. If you enabled this setting using a previous version of StorageGRID, the S3 Object Lock setting is enabled automatically. You can continue to use StorageGRID to manage the settings of existing compliant buckets; however, you can’t create new compliant buckets. For details, see NetApp Knowledge Base: How to manage legacy Compliant buckets in StorageGRID 11.5.

### Steps

1. Select **CONFIGURATION** > **System** > **S3 Object Lock**.

   The S3 Object Lock Settings page appears.

2. Select **Enable S3 Object Lock**.
3. Select **Apply**.

   A confirmation dialog box appears and reminds you that you can’t disable S3 Object Lock after it is enabled.

4. If you are sure you want to permanently enable S3 Object Lock for your entire system, select **OK**.

   When you select **OK**:

   - If the default rule in the active ILM policy is compliant, S3 Object Lock is now enabled for the entire grid and can’t be disabled.
   - If the default rule is not compliant, an error appears. You must create and activate a new ILM policy that includes a compliant rule as its default rule. Select **OK**. Then, create a new proposed policy, simulate it, and activate it. See Create ILM policy for instructions.

### After you finish

After you enable the global S3 Object Lock setting, you might want to create a new ILM policy. After the setting is enabled, the ILM policy can optionally include both a compliant default rule and a non-compliant default rule. For example, you might want to use a non-compliant rule that does not have filters for objects in buckets that don’t have S3 Object Lock enabled.

## Resolve consistency errors when updating the S3 Object Lock or legacy Compliance configuration

If a data center site or multiple Storage Nodes at a site become unavailable, you might need to help S3 tenant users apply changes to the S3 Object Lock or legacy Compliance configuration.

Tenant users who have buckets with S3 Object Lock (or legacy Compliance) enabled can change certain settings. For example, a tenant user using S3 Object Lock might need to put an object version under legal hold.

When a tenant user updates the settings for an S3 bucket or an object version, StorageGRID attempts to immediately update the bucket or object metadata across the grid. If the system is unable to update the metadata because a data center site or multiple Storage Nodes are unavailable, it returns an error:

```
503: Service Unavailable
Unable to update compliance settings because the settings can't be
consistently applied on enough storage services. Contact your grid
administrator for assistance.
```

To resolve this error, follow these steps:

1. Attempt to make all Storage Nodes or sites available again as soon as possible.
2. If you are unable to make enough of the Storage Nodes at each site available, contact technical support, who can help you recover nodes and ensure that changes are consistently applied across the grid.
3. Once the underlying issue has been resolved, remind the tenant user to retry their configuration changes.

**Related information**

- Use a tenant account
- Use S3 REST API
- Recover and maintain

# Example ILM rules and policies

## Example 1: ILM rules and policy for object storage

You can use the following example rules and policy as a starting point when defining an ILM policy to meet your object protection and retention requirements.

> ! The following ILM rules and policy are only examples. There are many ways to configure ILM rules. Before activating a new policy, simulate the proposed policy to confirm it will work as intended to protect content from loss.

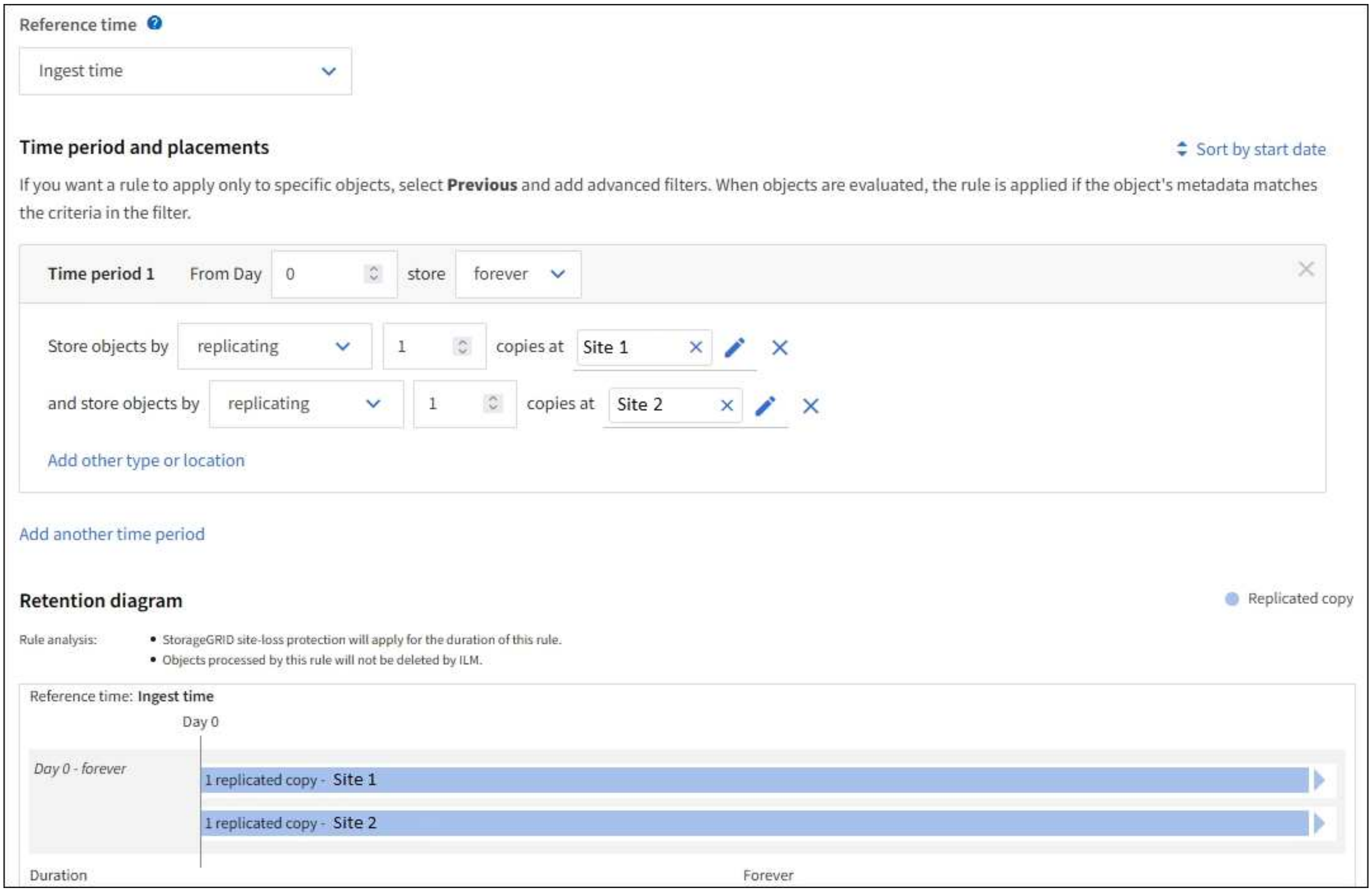
### ILM rule 1 for example 1: Copy object data to two sites

This example ILM rule copies object data to storage pools in two sites.

| Rule definition | Example value |
|---|---|
| One-site storage pools | Two storage pools, each containing different sites, named Site 1 and Site 2. |
| Rule name | Two Copies Two Sites |
| Reference time | Ingest time |
| Placements | On Day 0 to forever, keep one replicated copy at Site 1 and one replicated copy at Site 2. |

The Rule analysis section of the Retention diagram states:

- StorageGRID site-loss protection will apply for the duration of this rule.
- Objects processed by this rule will not be deleted by ILM.



Reference time

Ingest time

Time period and placements

Sort by start date

If you want a rule to apply only to specific objects, select **Previous** and add advanced filters. When objects are evaluated, the rule is applied if the object's metadata matches the criteria in the filter.

Time period 1 From Day 0 store forever

Store objects by replicating 1 copies at Site 1

and store objects by replicating 1 copies at Site 2

Add other type or location

Add another time period

Retention diagram

● Replicated copy

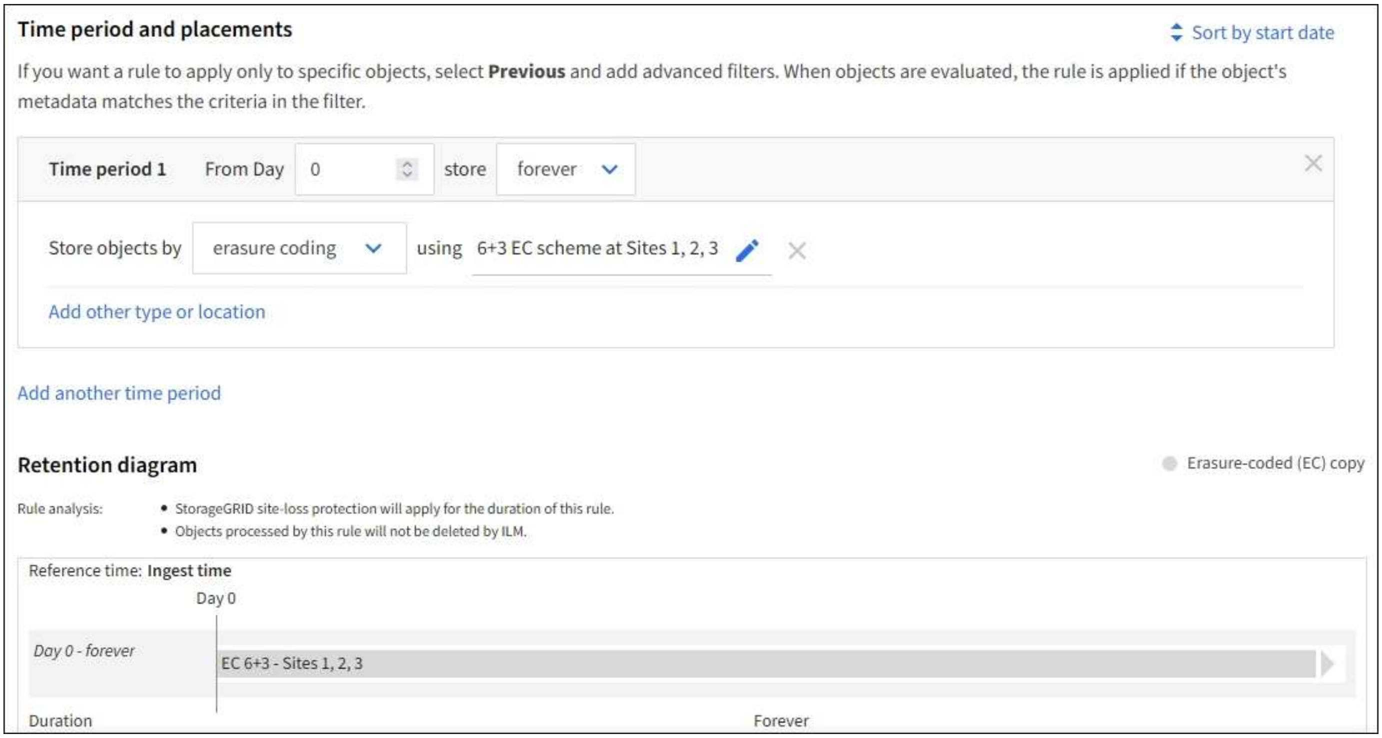
Rule analysis:
- StorageGRID site-loss protection will apply for the duration of this rule.
- Objects processed by this rule will not be deleted by ILM.

Reference time: Ingest time

### ILM rule 2 for example 1: Erasure coding profile with bucket matching

This example ILM rule uses an erasure coding profile and an S3 bucket to determine where and how long the object is stored.

| Rule definition | Example value |
|---|---|
| Storage pool with multiple sites | • One storage pool across three sites (Sites 1, 2, 3)<br>• Use 6+3 erasure-coding scheme |
| Rule name | S3 Bucket finance-records |
| Reference time | Ingest time |
| Placements | For objects in the S3 bucket named finance-records, create one erasure-coded copy in the pool specified by the erasure coding profile. Keep this copy forever. |



**Time period and placements**

If you want a rule to apply only to specific objects, select **Previous** and add advanced filters. When objects are evaluated, the rule is applied if the object's metadata matches the criteria in the filter.

Time period 1 From Day 0 store forever

Store objects by erasure coding using 6+3 EC scheme at Sites 1, 2, 3

**Retention diagram**

Rule analysis:
- StorageGRID site-loss protection will apply for the duration of this rule.
- Objects processed by this rule will not be deleted by ILM.

Reference time: **Ingest time**

Day 0 - forever: EC 6+3 - Sites 1, 2, 3

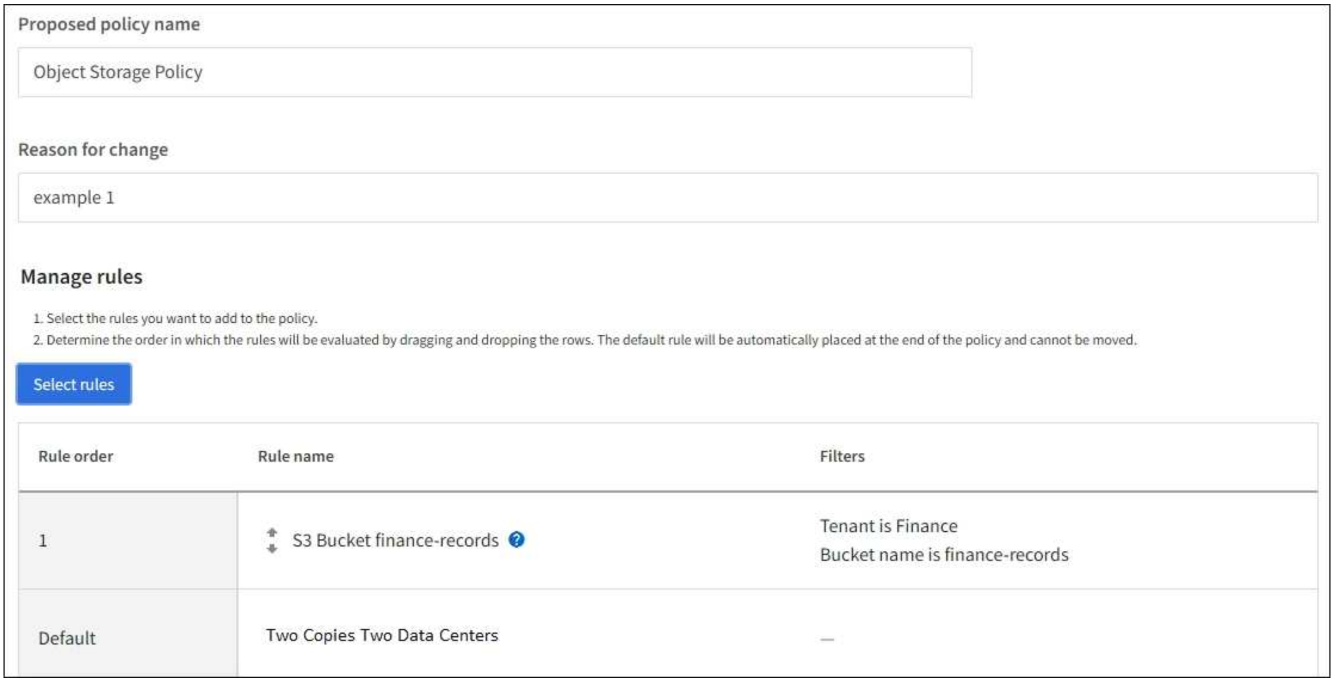
Duration: Forever

### ILM policy for example 1

In practice, most ILM policies are simple, even though the StorageGRID system allows you to design sophisticated and complex ILM policies.

A typical ILM policy for a multi-site grid might include ILM rules such as the following:

- At ingest, store all objects belonging to the S3 bucket named `finance-records` in a storage pool that contains three sites. Use 6+3 erasure coding.
- If an object does not match the first ILM rule, use the policy's default ILM rule, Two Copies Two Data Centers, to store one copy of that object in Site 1, and one copy in Site 2.



Proposed policy name: Object Storage Policy

Reason for change: example 1

**Manage rules**

1. Select the rules you want to add to the policy.
2. Determine the order in which the rules will be evaluated by dragging and dropping the rows. The default rule will be automatically placed at the end of the policy and cannot be moved.

| Rule order | Rule name | Filters |
|---|---|---|
| 1 | S3 Bucket finance-records | Tenant is Finance<br>Bucket name is finance-records |
| Default | Two Copies Two Data Centers | — |

**Related information**

- Create an ILM policy: Overview
- Create a proposed ILM policy

## Example 2: ILM rules and policy for EC object size filtering

You can use the following example rules and policy as starting points to define an ILM policy that filters by object size to meet recommended EC requirements.

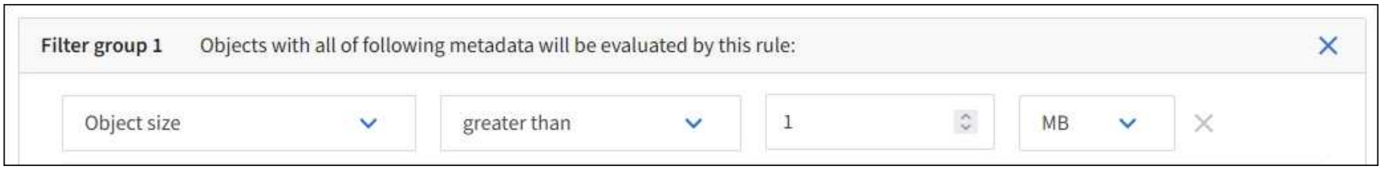
> The following ILM rules and policy are only examples. There are many ways to configure ILM rules. Before activating a new policy, simulate the proposed policy to confirm it will work as intended to protect content from loss.

### ILM rule 1 for example 2: Use EC for objects greater than 1 MB

This example ILM rule erasure codes objects that are greater than 1 MB.

> Erasure coding is best suited for objects greater than 1 MB. Don’t use erasure coding for objects smaller than 200 KB to avoid the overhead of managing very small erasure-coded fragments.

| Rule definition | Example value |
|---|---|
| Rule name | EC Only Objects > 1 MB |
| Reference time | Ingest time |
| Advanced filter for Object size | Object size greater than 1 MB |
| Placements | Create a 2+1 erasure-coded copy using three sites |



**Filter group 1** Objects with all of following metadata will be evaluated by this rule:

Object size | greater than | 1 | MB

### ILM rule 2 for example 2: Two replicated copies

This example ILM rule creates two replicated copies and does not filter by object size. This rule is the default rule for the policy. Because the first rule filters out all objects greater than 1 MB, this rule only applies to objects that are 1 MB or smaller.

| Rule definition | Example value |
|---|---|
| Rule name | Two Replicated Copies |
| Reference time | Ingest time |

| Rule definition | Example value |
|---|---|
| Advanced filter for Object size | None |
| Placements | On Day 0 to forever, keep one replicated copy at Site 1 and one replicated copy at Site 2. |

#### ILM policy for example 2: Use EC for objects greater than 1 MB

This example ILM policy includes two ILM rules:

- The first rule erasure codes all objects that are greater than 1 MB.
- The second (default) ILM rule creates two replicated copies. Because objects greater than 1 MB have been filtered out by rule 1, rule 2 only applies to objects that are 1 MB or smaller.

### Example 3: ILM rules and policy for better protection for image files

You can use the following example rules and policy to ensure that images greater than 1 MB are erasure coded and that two copies are made of smaller images.

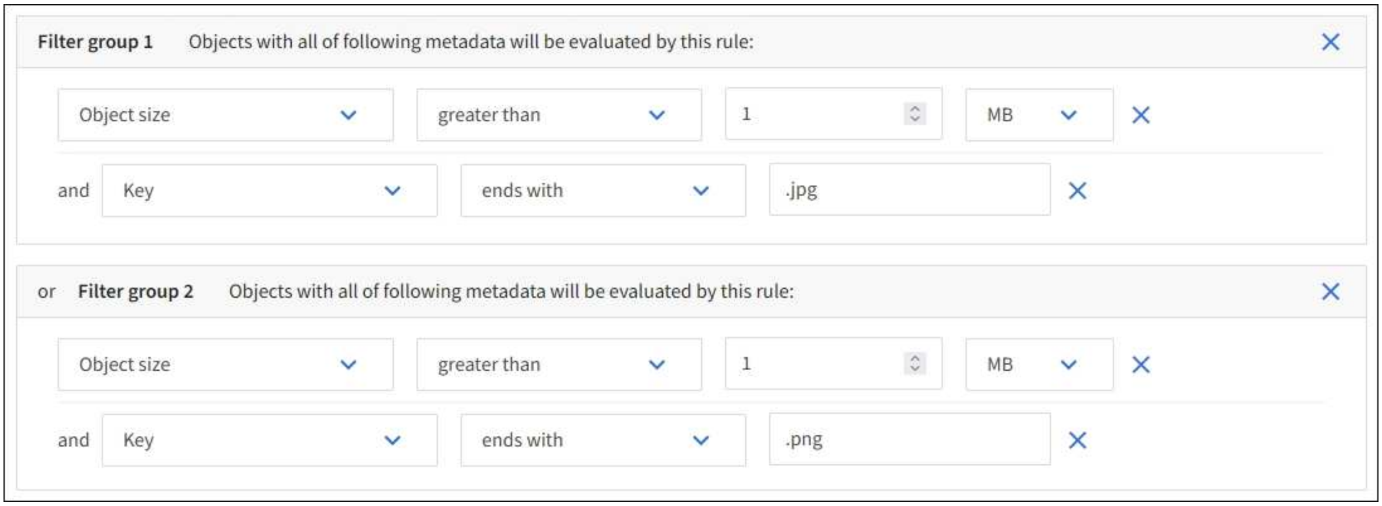
> ! The following ILM rules and policy are only examples. There are many ways to configure ILM rules. Before activating a new policy, simulate the proposed policy to confirm it will work as intended to protect content from loss.

#### ILM rule 1 for example 3: Use EC for image files greater than 1 MB

This example ILM rule uses advanced filtering to erasure code all image files greater than 1 MB.

> i Erasure coding is best suited for objects greater than 1 MB. Don’t use erasure coding for objects smaller than 200 KB to avoid the overhead of managing very small erasure-coded fragments.

| Rule definition | Example value |
|---|---|
| Rule name | EC Image Files > 1 MB |
| Reference time | Ingest time |
| Advanced filter for Object size | Object size greater than 1 MB |
| Advanced filters for Key | • Ends with .jpg<br>• Ends with .png |
| Placements | Create a 2+1 erasure-coded copy using three sites |



**Filter group 1** Objects with all of following metadata will be evaluated by this rule:

Object size | greater than | 1 | MB

and Key | ends with | .jpg

or **Filter group 2** Objects with all of following metadata will be evaluated by this rule:

Object size | greater than | 1 | MB

and Key | ends with | .png

Because this rule is configured as the first rule in the policy, the erasure-coding placement instruction only applies to .jpg and .png files that are greater than 1 MB.

### ILM rule 2 for example 3: Create 2 replicated copies for all remaining image files

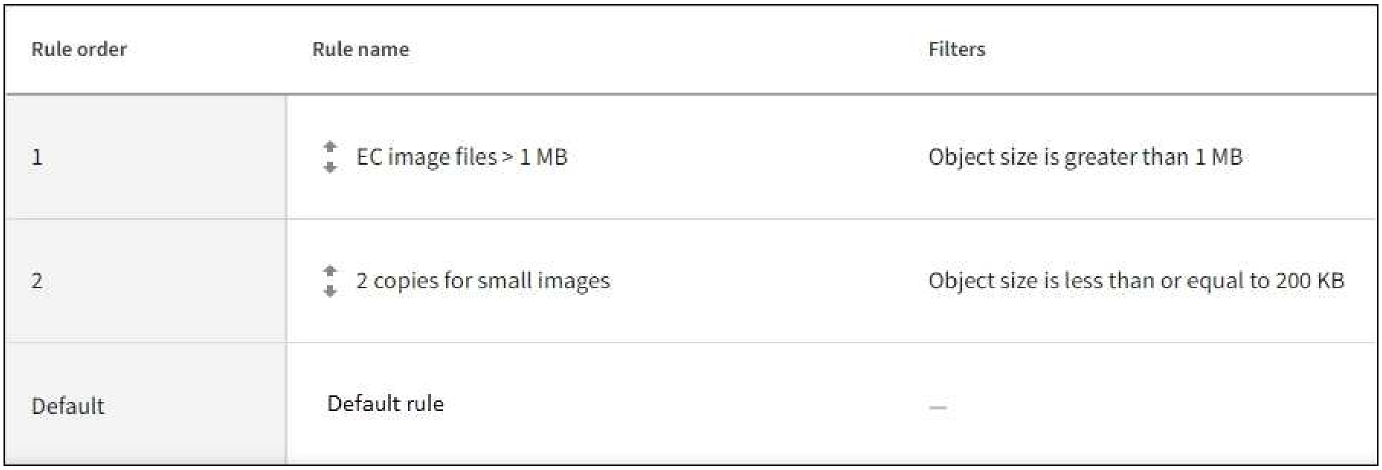
This example ILM rule uses advanced filtering to specify that smaller image files be replicated. Because the first rule in the policy has already matched image files greater than 1 MB, this rule applies to image files that are 1 MB or smaller.

| Rule definition | Example value |
|---|---|
| Rule name | 2 Copies for Image Files |
| Reference time | Ingest time |
| Advanced filters for Key | • Ends with .jpg<br>• Ends with .png |
| Placements | Create 2 replicated copies in two storage pools |

### ILM policy for example 3: Better protection for image files

This example ILM policy includes three rules:

- The first rule erasure codes all image files greater than 1 MB.
- The second rule creates two copies of any remaining image files (that is, images that are 1 MB or smaller).
- The default rule applies to all remaining objects (that is, any non-image files).



| Rule order | Rule name | Filters |
|---|---|---|
| 1 | ↕ EC image files > 1 MB | Object size is greater than 1 MB |
| 2 | ↕ 2 copies for small images | Object size is less than or equal to 200 KB |
| Default | Default rule | — |

## Example 4: ILM rules and policy for S3 versioned objects

If you have an S3 bucket with versioning enabled, you can manage the noncurrent object versions by including rules in your ILM policy that use “Noncurrent time” as the reference time.

> If you specify a limited retention time for objects, those objects will be deleted permanently after the time period is reached. Make sure you understand how long the objects will be retained.

As this example shows, you can control the amount of storage used by versioned objects by using different placement instructions for noncurrent object versions.

> The following ILM rules and policy are only examples. There are many ways to configure ILM rules. Before activating a new policy, simulate the proposed policy to confirm it will work as intended to protect content from loss.

> To perform ILM policy simulation on a noncurrent version of an object, you must know the object version’s UUID or CBID. To find the UUID and CBID, use object metadata lookup while the object is still current.

**Related information**

- How objects are deleted

### ILM rule 1 for example 4: Save three copies for 10 years

This example ILM rule stores a copy of each object at three sites for 10 years.

This rule applies to all objects, whether or not they are versioned.

| Rule definition | Example value |
|---|---|
| Storage pools | Three storage pools, each consisting of different data centers, named Site 1, Site 2, and Site 3. |
| Rule name | Three Copies Ten Years |

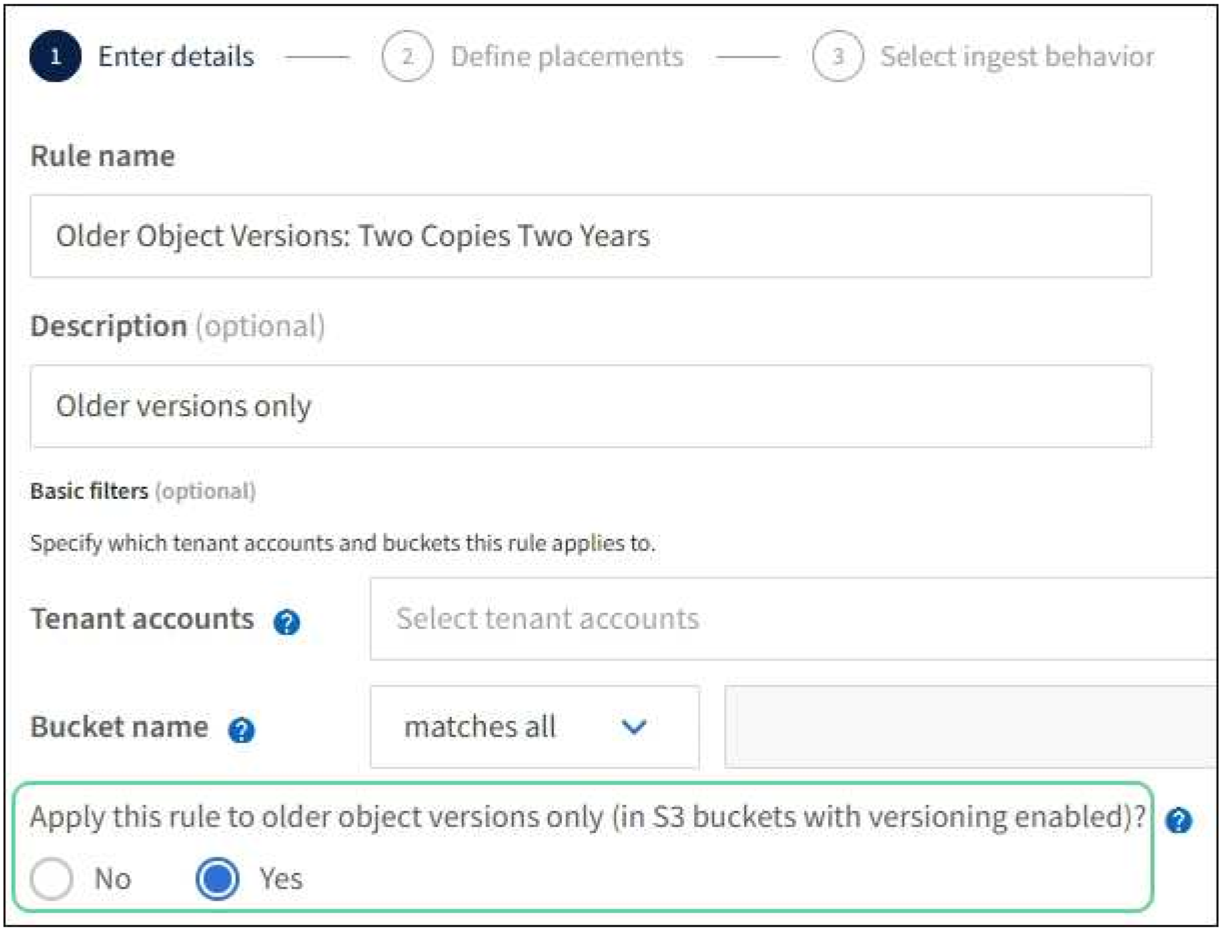
| Rule definition | Example value |
|---|---|
| Reference time | Ingest time |
| Placements | On Day 0, keep three replicated copies for 10 years (3,652 days), one in Site 1, one in Site 2, and one in Site 3. At the end of 10 years, delete all copies of the object. |

**ILM rule 2 for example 4: Save two copies of noncurrent versions for 2 years**

This example ILM rule stores two copies of the noncurrent versions of an S3 versioned object for 2 years.

Because ILM rule 1 applies to all versions of the object, you must create another rule to filter out any noncurrent versions.

To create a rule that uses “Noncurrent time” as the reference time, select **Yes** for the question, “Apply this rule to older object versions only (in S3 buckets with versioning enabled)?” in Step 1 (Enter details) of the Create an ILM rule wizard. When you select **Yes**, *Noncurrent time* is automatically selected for the reference time, and you can’t select a different reference time.



Rule name

Older Object Versions: Two Copies Two Years

Description (optional)

Older versions only

Basic filters (optional)

Specify which tenant accounts and buckets this rule applies to.

Tenant accounts: Select tenant accounts

Bucket name: matches all

Apply this rule to older object versions only (in S3 buckets with versioning enabled)?

○ No ● Yes

In this example, only two copies of the noncurrent versions are stored, and those copies will be stored for two years.

| Rule definition | Example value |
|---|---|
| Storage Pools | Two storage pools, each at different data centers, Site 1 and Site 2. |
| Rule name | Noncurrent Versions: Two Copies Two Years |
| Reference time | Noncurrent time<br>Automatically selected when you select **Yes** for the question, "Apply this rule to older object versions only (in S3 buckets with versioning enabled)?" in the Create an ILM rule wizard. |
| Placements | On Day 0 relative to noncurrent time (that is, starting from the day the object version becomes the noncurrent version), keep two replicated copies of the noncurrent object versions for 2 years (730 days), one in Site 1 and one in Site 2. At the end of 2 years, delete the noncurrent versions. |

### ILM policy for example 4: S3 versioned objects

If you want to manage older versions of an object differently than the current version, rules that use "Noncurrent time" as the reference time must appear in the ILM policy before rules that apply to the current object version.

An ILM policy for S3 versioned objects might include ILM rules such as the following:

- Keep any older (noncurrent) versions of each object for 2 years, starting from the day the version became noncurrent.

  > The "Noncurrent time" rules must appear in the policy before the rules that apply to the current object version. Otherwise, the noncurrent object versions will never be matched by the "Noncurrent time" rule.

- At ingest, create three replicated copies and store one copy at each of three sites. Keep copies of the current object version for 10 years.

When you simulate the example policy, you would expect test objects to be evaluated as follows:

- Any noncurrent object versions would be matched by the first rule. If a noncurrent object version is older than 2 years, it is permanently deleted by ILM (all copies of the noncurrent version removed from the grid).

  > To simulate noncurrent object versions, you must use that version's UUID or CBID. While the object is still current, you can use object metadata lookup to find its UUID and CBID.

- The current object version would be matched by the second rule. When the current object version has been stored for 10 years, the ILM process adds a delete marker as the current version of the object, and it makes the previous object version "noncurrent". The next time ILM evaluation occurs, this noncurrent version is matched by the first rule. As a result, the copy at Site 3 is purged and the two copies at Site 1 and Site 2 are stored for 2 more years.

## Example 5: ILM rules and policy for Strict ingest behavior

You can use a location filter and the Strict ingest behavior in a rule to prevent objects from being saved at a particular data center location.

In this example, a Paris-based tenant does not want to store some objects outside of the EU because of regulatory concerns. Other objects, including all objects from other tenant accounts, can be stored at either the Paris data center or the US data center.

> ! The following ILM rules and policy are only examples. There are many ways to configure ILM rules. Before activating a new policy, simulate the proposed policy to confirm it will work as intended to protect content from loss.

**Related information**

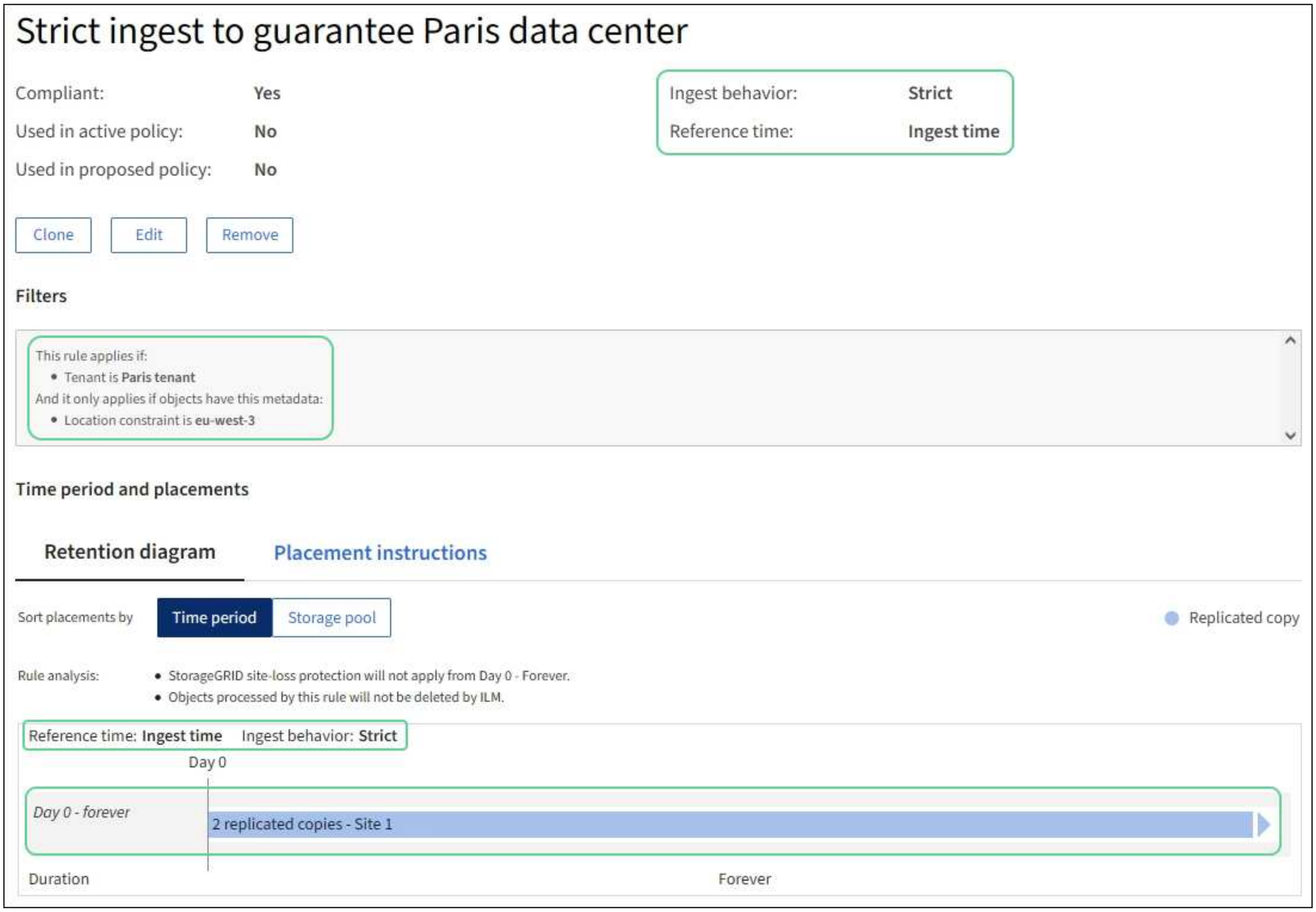
- Ingest options
- Create ILM rule: Select ingest behavior

### ILM rule 1 for example 5: Strict ingest to guarantee Paris data center

This example ILM rule uses the Strict ingest behavior to guarantee that objects saved by a Paris-based tenant to S3 buckets with the region set to eu-west-3 region (Paris) are never stored at the US data center.

This rule applies to objects that belong to the Paris tenant and that have the S3 bucket region set to eu-west-3 (Paris).

| Rule definition | Example value |
|---|---|
| Tenant account | Paris tenant |
| Advanced filter | Location constraint equals eu-west-3 |
| Storage pools | Site 1 (Paris) |
| Rule name | Strict ingest to guarantee Paris data center |
| Reference time | Ingest time |
| Placements | On Day 0, keep two replicated copies forever in Site 1 (Paris) |
| Ingest behavior | Strict. Always use this rule’s placements on ingest. Ingest fails if it is not possible to store two copies of the object at the Paris data center. |



Strict ingest to guarantee Paris data center

Compliant: Yes

Used in active policy: No

Used in proposed policy: No

Ingest behavior: Strict

Reference time: Ingest time

Filters

This rule applies if:

- Tenant is **Paris tenant**

And it only applies if objects have this metadata:

- Location constraint is **eu-west-3**

Time period and placements

Sort placements by: Time period

Rule analysis:

- StorageGRID site-loss protection will not apply from Day 0 - Forever.
- Objects processed by this rule will not be deleted by ILM.

Reference time: **Ingest time** Ingest behavior: **Strict**

*Day 0 - forever*: 2 replicated copies - Site 1

### ILM rule 2 for example 5: Balanced ingest for other objects

This example ILM rule uses the Balanced ingest behavior to provide optimum ILM efficiency for any objects not matched by the first rule. Two copies of all objects matched by this rule will be stored—one at the US data center and one at the Paris data center. If the rule can’t be satisfied immediately, interim copies are stored at any available location.

This rule applies to objects that belong to any tenant and any region.

| Rule definition | Example value |
|---|---|
| Tenant account | Ignore |
| Advanced filter | *Not specified* |
| Storage pools | Site 1 (Paris) and Site 2 (US) |
| Rule name | 2 Copies 2 Data Centers |
| Reference time | Ingest time |
| Placements | On Day 0, keep two replicated copies forever at two data centers |

| Rule definition | Example value |
|---|---|
| Ingest behavior | Balanced. Objects that match this rule are placed according to the rule's placement instructions if possible. Otherwise, interim copies are made at any available location. |

#### ILM policy for example 5: Combining ingest behaviors

The example ILM policy includes two rules that have different ingest behaviors.

An ILM policy that uses two different ingest behaviors might include ILM rules such as the following:

- Store objects that belong to the Paris tenant and that have the S3 bucket region set to eu-west-3 (Paris) only in the Paris data center. Fail ingest if the Paris data center is not available.
- Store all other objects (including those that belong to the Paris tenant but that have a different bucket region) in both the US data center and the Paris data center. Make interim copies in any available location if the placement instruction can't be satisfied.

When you simulate the example policy, you expect test objects to be evaluated as follows:

- Any objects that belong to the Paris tenant and that have the S3 bucket region set to eu-west-3 are matched by the first rule and are stored at the Paris data center. Because the first rule uses Strict ingest, these objects are never stored at the US data center. If the Storage Nodes at the Paris data center aren't available, ingest fails.
- All other objects are matched by the second rule, including objects that belong to the Paris tenant and that don't have the S3 bucket region set to eu-west-3. One copy of each object is saved at each data center. However, because the second rule uses Balanced ingest, if one data center is unavailable, two interim copies are saved at any available location.

## Example 6: Change an ILM policy

If your data protection needs to be changed or you add new sites, you can create and activate a new ILM policy.

Before changing a policy, you must understand how changes in ILM placements can temporarily affect the overall performance of a StorageGRID system.

In this example, a new StorageGRID site has been added in an expansion, and a new active ILM policy needs to be implemented to store data at the new site. To implement a new active policy, first create a proposed policy by either cloning an existing policy *or* starting from scratch. Afterward, you must simulate and then activate the new policy.

> ! The following ILM rules and policy are only examples. There are many ways to configure ILM rules. Before activating a new policy, simulate the proposed policy to confirm it will work as intended to protect content from loss.

#### How changing an ILM policy affects performance

When you activate a new ILM policy, the performance of your StorageGRID system might be temporarily affected, especially if the placement instructions in the new policy require many existing objects to be moved to new locations.

When you activate a new ILM policy, StorageGRID uses it to manage all objects, including existing objects and newly ingested objects. Before activating a new ILM policy, review any changes to the placement of existing replicated and erasure-coded objects. Changing an existing object's location might result in temporary resource issues when the new placements are evaluated and implemented.

To ensure a new ILM policy does not affect the placement of existing replicated and erasure-coded objects, you can create an ILM rule with an ingest time filter. For example, **Ingest time** ***is on or after <date and time>***, so that the new rule applies only to objects ingested on or after the date and time specified.

The types of ILM policy changes that can temporarily affect StorageGRID performance include the following:

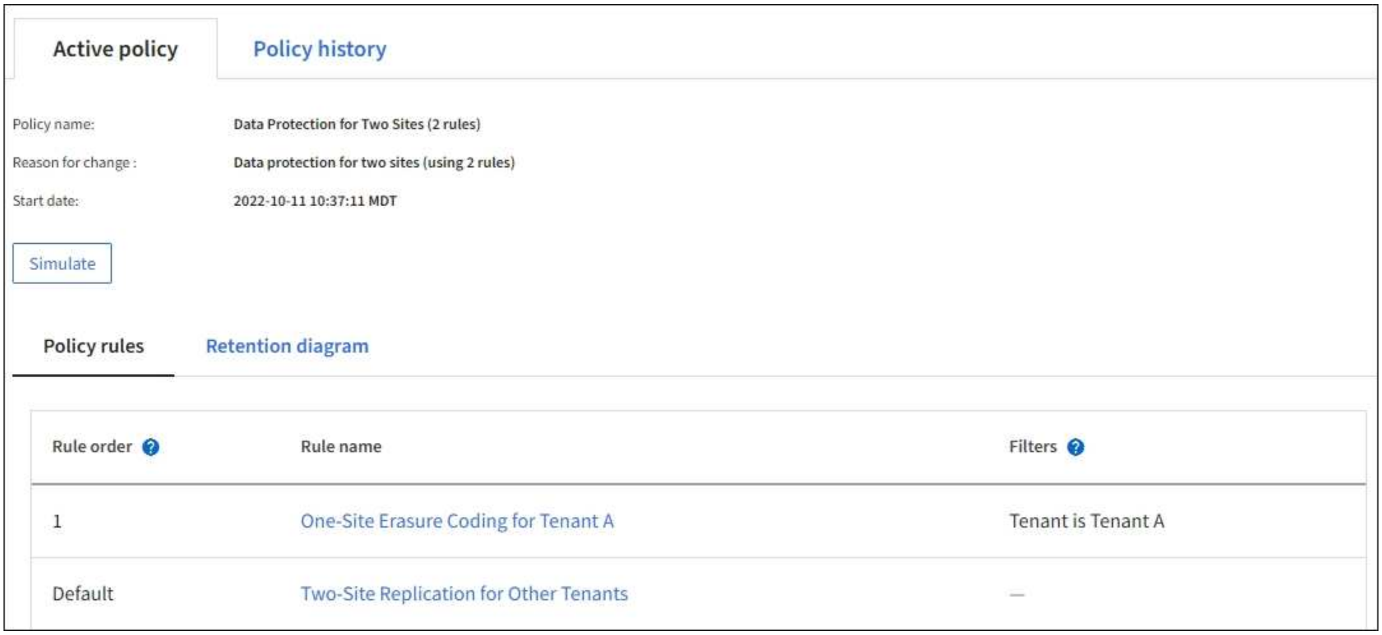
- Applying a different erasure coding profile to existing erasure-coded objects.

  > StorageGRID considers each erasure coding profile to be unique and does not reuse erasure-coding fragments when a new profile is used.

- Changing the type of copies required for existing objects; for example, converting a large percentage of replicated objects to erasure-coded objects.
- Moving copies of existing objects to a completely different location; for example, moving a large number of objects to or from a Cloud Storage Pool or to or from a remote site.

### Active ILM policy for example 6: Data protection at two sites

In this example, the active ILM policy was initially designed for a two-site StorageGRID system and uses two ILM rules.



Policy name: Data Protection for Two Sites (2 rules)
Reason for change : Data protection for two sites (using 2 rules)
Start date: 2022-10-11 10:37:11 MDT

| Rule order | Rule name | Filters |
|---|---|---|
| 1 | One-Site Erasure Coding for Tenant A | Tenant is Tenant A |
| Default | Two-Site Replication for Other Tenants | — |

In this ILM policy, objects belonging to Tenant A are protected by 2+1 erasure coding at a single site, while objects belonging to all other tenants are protected across two sites using 2-copy replication.

> The first rule in this example uses an advanced filter to ensure that erasure coding is not used for small objects. Any of Tenant A's objects that are smaller than 1 MB will be protected by the default rule, which uses replication.

#### Rule 1: One-site erasure coding for Tenant A

| Rule definition | Example value |
|---|---|
| Rule name | One-Site Erasure Coding for Tenant A |
| Tenant Account | Tenant A |
| Storage Pool | Site 1 |
| Placements | 2+1 erasure coding in Site 1 from day 0 to forever |

#### Rule 2: Two-site replication for other tenants

| Rule definition | Example value |
|---|---|
| Rule name | Two-Site Replication for Other Tenants |
| Tenant Account | Ignore |
| Storage Pools | Site 1 and Site 2 |
| Placements | Two replicated copies from day 0 to forever: one copy at Site 1 and one copy at Site 2. |

### Proposed ILM policy for example 6: Data protection at three sites

In this example, the ILM policy is being replaced with a new policy for a three-site StorageGRID system.

After performing an expansion to add the new site, the grid administrator created two new storage pools: a storage pool for Site 3 and a storage pool containing all three sites (not the same as the All Storage Nodes default storage pool). Then, the administrator created two new ILM rules and a new proposed ILM policy, which is designed to protect data at all three sites.

When this new ILM policy is activated, objects belonging to Tenant A will be protected by 2+1 erasure coding at three sites, while objects belonging to other tenants (and smaller objects belonging to Tenant A) will be protected across three sites using 3-copy replication.

#### Rule 1: Three-site erasure coding for Tenant A

| Rule definition | Example value |
|---|---|
| Rule name | Three-Site Erasure Coding for Tenant A |
| Tenant Account | Tenant A |
| Storage Pool | All 3 Sites (includes Site 1, Site 2, and Site 3) |
| Placements | 2+1 erasure coding in All 3 Sites from day 0 to forever |

#### Rule 2: Three-site replication for other tenants

| Rule definition | Example value |
|---|---|
| Rule name | Three-Site Replication for Other Tenants |
| Tenant Account | Ignore |
| Storage Pools | Site 1, Site 2, and Site 3 |
| Placements | Three replicated copies from day 0 to forever: one copy at Site 1, one copy at Site 2, and one copy at Site 3. |

### Activating the proposed ILM policy for example 6

When you activate a new proposed ILM policy, existing objects might be moved to new locations or new object copies might be created for existing objects, based on the placement instructions in any new or updated rules.

> Errors in an ILM policy can cause unrecoverable data loss. Carefully review and simulate the policy before activating it to confirm that it will work as intended.

> When you activate a new ILM policy, StorageGRID uses it to manage all objects, including existing objects and newly ingested objects. Before activating a new ILM policy, review any changes to the placement of existing replicated and erasure-coded objects. Changing an existing object’s location might result in temporary resource issues when the new placements are evaluated and implemented.

#### What happens when erasure-coding instructions change

In the currently active ILM policy for this example, objects belonging to Tenant A are protected using 2+1 erasure coding at Site 1. In the new proposed ILM policy, objects belonging to Tenant A will be protected using 2+1 erasure coding at Sites 1, 2, and 3.

When the new ILM policy is activated, the following ILM operations occur:

- New objects ingested by Tenant A are split into two data fragments and one parity fragment is added. Then, each of the three fragments is stored at a different site.
- The existing objects belonging to Tenant A are re-evaluated during the ongoing ILM scanning process. Because the ILM placement instructions use a new erasure coding profile, entirely new erasure-coded fragments are created and distributed to the three sites.

  > The existing 2+1 fragments at Site 1 aren’t reused. StorageGRID considers each erasure coding profile to be unique and does not reuse erasure-coding fragments when a new profile is used.

#### What happens when replication instructions change

In the currently active ILM policy for this example, objects belonging other tenants are protected using two replicated copies in storage pools at Sites 1 and 2. In the new proposed ILM policy, objects belonging to other tenants will be protected using three replicated copies in storage pools at Sites 1, 2, and 3.

When the new ILM policy is activated, the following ILM operations occur:

- When any tenant other than Tenant A ingests a new object, StorageGRID creates three copies and saves one copy at each site.
- Existing objects belonging to these other tenants are re-evaluated during the ongoing ILM scanning process. Because the existing object copies at Site 1 and Site 2 continue to satisfy the replication requirements of the new ILM rule, StorageGRID only needs to create one new copy of the object for Site 3.

#### Performance impact of activating this policy

When the proposed ILM policy in this example is activated, the overall performance of this StorageGRID system will be temporarily affected. Higher than normal levels of grid resources will be required to create new erasure-coded fragments for Tenant A’s existing objects and new replicated copies at Site 3 for other tenants' existing objects.

As a result of the ILM policy change, client read and write requests might temporarily experience higher than normal latencies. Latencies will return to normal levels after the placement instructions are fully implemented across the grid.

To avoid resource issues when activating a new ILM policy, you can use the Ingest time advanced filter in any rule that might change the location of large numbers of existing objects. Set Ingest time to be greater than or equal to the approximate time when the new policy will go into effect to ensure that existing objects aren’t moved unnecessarily.

ⓘ Contact technical support if you need to slow or increase the rate at which objects are processed after an ILM policy change.

## Example 7: Compliant ILM policy for S3 Object Lock

You can use the S3 bucket, ILM rules, and ILM policy in this example as a starting point when defining an ILM policy to meet the object protection and retention requirements for objects in buckets with S3 Object Lock enabled.

ⓘ If you used the legacy Compliance feature in previous StorageGRID releases, you can also use this example to help manage any existing buckets that have the legacy Compliance feature enabled.

❗ The following ILM rules and policy are only examples. There are many ways to configure ILM rules. Before activating a new policy, simulate the proposed policy to confirm it will work as intended to protect content from loss.

#### Related information

- Manage objects with S3 Object Lock
- Create an ILM policy

### Bucket and objects for S3 Object Lock example

In this example, an S3 tenant account named Bank of ABC has used the Tenant Manager to create a bucket with S3 Object Lock enabled to store critical bank records.

| Bucket definition | Example value |
|---|---|
| Tenant Account Name | Bank of ABC |
| Bucket Name | bank-records |
| Bucket Region | us-east-1 (default) |

Each object and object version that is added to the bank-records bucket will use the following values for `retain-until-date` and `legal hold` settings.

| Setting for each object | Example value |
|---|---|
| `retain-until-date` | `"2030-12-30T23:59:59Z"` (December 30, 2030)<br>Each object version has its own `retain-until-date` setting. This setting can be increased, but not decreased. |
| `legal hold` | `"OFF"` (Not in effect)<br>A legal hold can be placed or lifted on any object version at any time during the retention period. If an object is under a legal hold, the object can't be deleted even if the `retain-until-date` has been reached. |

### ILM rule 1 for S3 Object Lock example: Erasure coding profile with bucket matching

This example ILM rule applies only to the S3 tenant account named Bank of ABC. It matches any object in the `bank-records` bucket and then uses erasure coding to store the object on Storage Nodes at three data center sites using a 6+3 erasure coding profile. This rule satisfies the requirements of buckets with S3 Object Lock enabled: a copy is kept on Storage Nodes from day 0 to forever, using Ingest time as the reference time.

| Rule definition | Example value |
|---|---|
| Rule name | Compliant Rule: EC Objects in bank-records Bucket - Bank of ABC |
| Tenant Account | Bank of ABC |
| Bucket Name | `bank-records` |
| Advanced filter | Object Size (MB) greater than 1<br>**Note:** This filter ensures that erasure coding is not used for objects 1 MB or smaller. |

| Rule definition | Example value |
|---|---|
| Reference time | Ingest time |

| Rule definition | Example value |
|---|---|
| Placements | From day 0 store forever |
| Erasure Coding Profile | • Create an erasure-coded copy on Storage Nodes at three data center sites<br>• Uses 6+3 erasure-coding scheme |

### ILM rule 2 for S3 Object Lock example: Non-compliant rule

This example ILM rule initially stores two replicated object copies on Storage Nodes. After one year, it stores one copy on a Cloud Storage Pool forever. Because this rule uses a Cloud Storage Pool, it is not compliant and will not apply to the objects in buckets with S3 Object Lock enabled.

| Rule definition | Example value |
|---|---|
| Rule name | Non-compliant rule: Use Cloud Storage Pool |
| Tenant accounts | Not specified |
| Bucket name | Not specified, but will only apply to buckets that don't have S3 Object Lock (or the legacy Compliance feature) enabled. |
| Advanced filter | Not specified |

| Rule definition | Example value |
|---|---|
| Reference time | Ingest time |
| Placements | • On Day 0, keep two replicated copies on Storage Nodes in Data Center 1 and Data Center 2 for 365 days<br>• After 1 year, keep one replicated copy in a Cloud Storage Pool forever |

### ILM rule 3 for S3 Object Lock example: Default rule

This example ILM rule copies object data to storage pools in two data centers. This compliant rule is designed to be the default rule in the ILM policy. It does not include any filters, does not use the Noncurrent reference time, and satisfies the requirements of buckets with S3 Object Lock enabled: two object copies are kept on Storage Nodes from day 0 to forever, using Ingest as the reference time.

| Rule definition | Example value |
|---|---|
| Rule name | Default compliant rule: Two Copies Two Data Centers |
| Tenant account | Not specified |

| Rule definition | Example value |
|---|---|
| Bucket name | Not specified |
| Advanced filter | Not specified |

| Rule definition | Example value |
|---|---|
| Reference time | Ingest time |
| Placements | From Day 0 to forever, keep two replicated copies—one on Storage Nodes in Data Center 1 and one on Storage Nodes in Data Center 2. |

### Compliant ILM policy for S3 Object Lock example

To create an ILM policy that will effectively protect all objects in your system, including those in buckets with S3 Object Lock enabled, you must select ILM rules that satisfy the storage requirements for all objects. Then, you must simulate and activate the proposed policy.

#### Add rules to the policy

In this example, the ILM policy includes three ILM rules, in the following order:

1. A compliant rule that uses erasure coding to protect objects greater than 1 MB in a specific bucket with S3 Object Lock enabled. The objects are stored on Storage Nodes from day 0 to forever.
2. A non-compliant rule that creates two replicated object copies on Storage Nodes for a year and then moves one object copy to a Cloud Storage Pool forever. This rule does not apply to buckets with S3 Object Lock enabled because it uses a Cloud Storage Pool.
3. The default compliant rule that creates two replicated object copies on Storage Nodes from day 0 to forever.

#### Simulate the proposed policy

After you have added rules in your proposed policy, chosen a default compliant rule, and arranged the other rules, you should simulate the policy by testing objects from the bucket with S3 Object Lock enabled and from other buckets. For example, when you simulate the example policy, you would expect test objects to be evaluated as follows:

- The first rule will only match test objects that are greater than 1 MB in the bucket bank-records for the Bank of ABC tenant.
- The second rule will match all objects in all non-compliant buckets for all other tenant accounts.
- The default rule will match these objects:
  - Objects 1 MB or smaller in the bucket bank-records for the Bank of ABC tenant.
  - Objects in any other bucket that has S3 Object Lock enabled for all other tenant accounts.

#### Activate the policy

When you are completely satisfied that the new policy protects object data as expected, you can activate it.

## Copyright information

## Trademark information