How StorageGRID manages data

StorageGRID 11.7

NetApp
April 10, 2024

Table of Contents

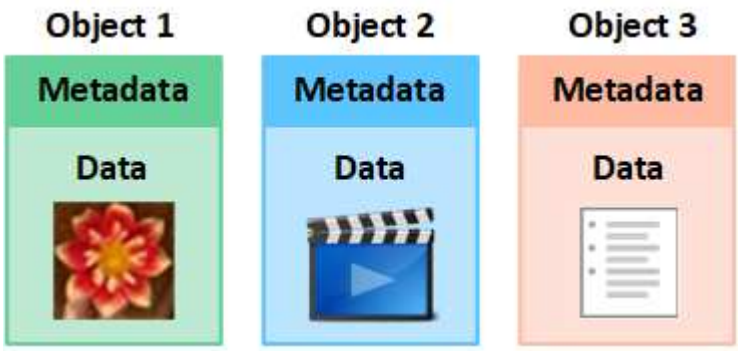
How StorageGRID manages data

What is an object

With object storage, the unit of storage is an object, rather than a file or a block. Unlike the tree-like hierarchy of a file system or block storage, object storage organizes data in a flat, unstructured layout.

Object storage decouples the physical location of the data from the method used to store and retrieve that data.

Each object in an object-based storage system has two parts: object data and object metadata.



What is object data?

Object data might be anything; for example, a photograph, a movie, or a medical record.

What is object metadata?

Object metadata is any information that describes an object. StorageGRID uses object metadata to track the locations of all objects across the grid and to manage each object's lifecycle over time.

Object metadata includes information such as the following:

- System metadata, including a unique ID for each object (UUID), the object name, the name of the S3 bucket or Swift 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.
- The current storage location of each object copy or erasure-coded fragment.
- Any user metadata associated with the object.

Object metadata is customizable and expandable, making it flexible for applications to use.

For detailed information about how and where StorageGRID stores object metadata, go to Manage object metadata storage.

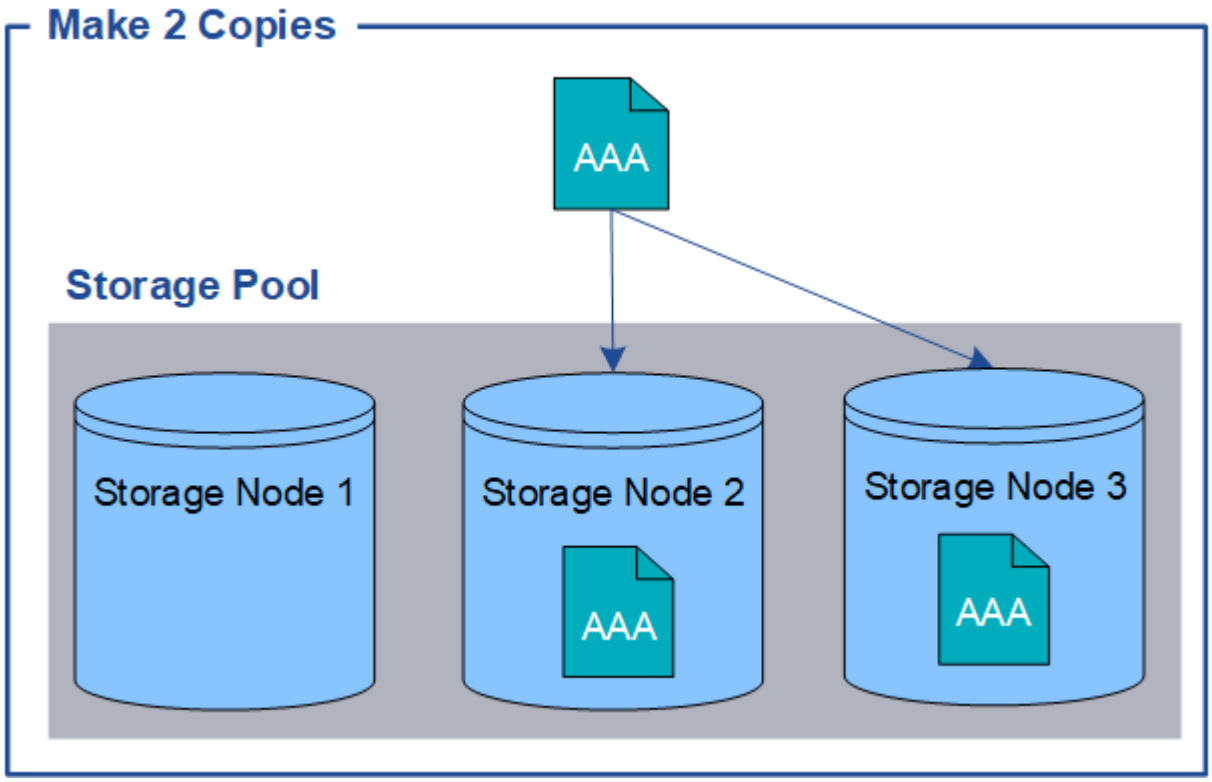
How is object data protected?

The StorageGRID system provides you with two mechanisms to protect object data from loss: replication and erasure coding.

Replication

When StorageGRID matches objects to an information lifecycle management (ILM) rule that is configured to create replicated copies, the system creates exact copies of object data and stores them on Storage Nodes, Archive Nodes, or Cloud Storage Pools. ILM rules dictate the number of copies made, where those copies are stored, and for how long they are retained by the system. If a copy is lost, for example, as a result of the loss of a Storage Node, the object is still available if a copy of it exists elsewhere in the StorageGRID system.

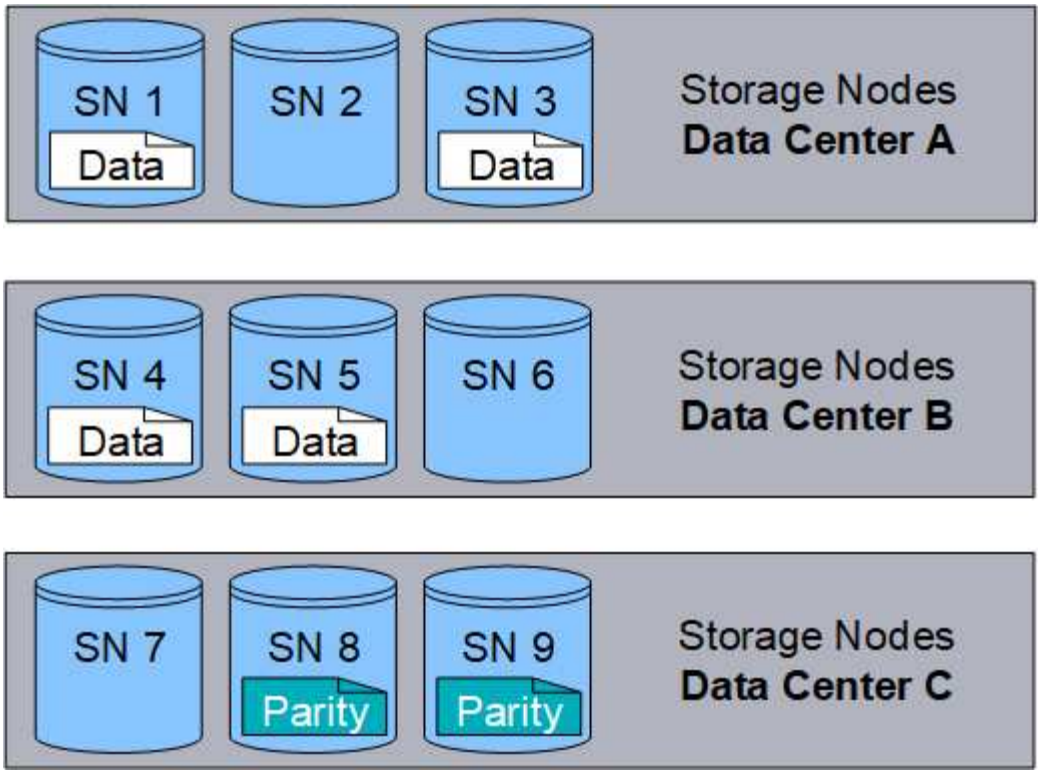
In the following example, the Make 2 Copies rule specifies that two replicated copies of each object be placed in a storage pool that contains three Storage Nodes.



Erasure coding

When StorageGRID matches objects to an ILM rule that is configured to create erasure-coded copies, it slices object data into data fragments, computes additional parity fragments, and stores each fragment 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. ILM rules and erasure coding profiles determine the erasure coding scheme used.

The following example illustrates the use of erasure coding 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 Storage Node across three data centers to provide data protection for node failures or site loss.



Related information

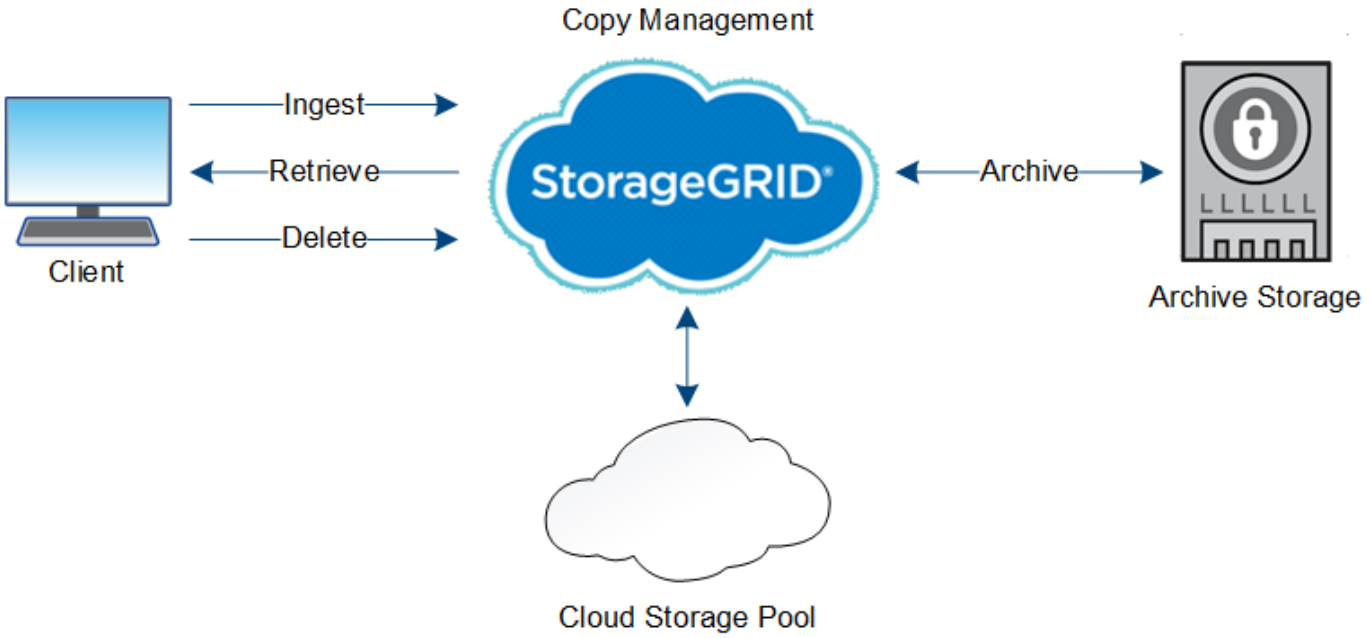
- Manage objects with ILM
- Use information lifecycle management

The life of an object

An object's life consists of various stages. Each stage represents the operations that occur with the object.

The life of an object includes the operations of ingest, copy management, retrieve, and delete.

- **Ingest**: The process of an S3 or Swift client application saving an object over HTTP to the StorageGRID system. At this stage, the StorageGRID system begins to manage the object.
- **Copy management**: The process of managing replicated and erasure-coded copies in StorageGRID, as described by the ILM rules in the active ILM policy. During the copy management stage, StorageGRID protects object data from loss by creating and maintaining the specified number and type of object copies on Storage Nodes, in a Cloud Storage Pool, or on Archive Node.
- **Retrieve**: The process of a client application accessing an object stored by the StorageGRID system. The client reads the object, which is retrieved from a Storage Node, Cloud Storage Pool, or Archive Node.
- **Delete**: The process of removing all object copies from the grid. Objects can be deleted either as a result of the client application sending a delete request to the StorageGRID system, or as a result of an automatic process that StorageGRID performs when the object's lifetime expires.



Related information

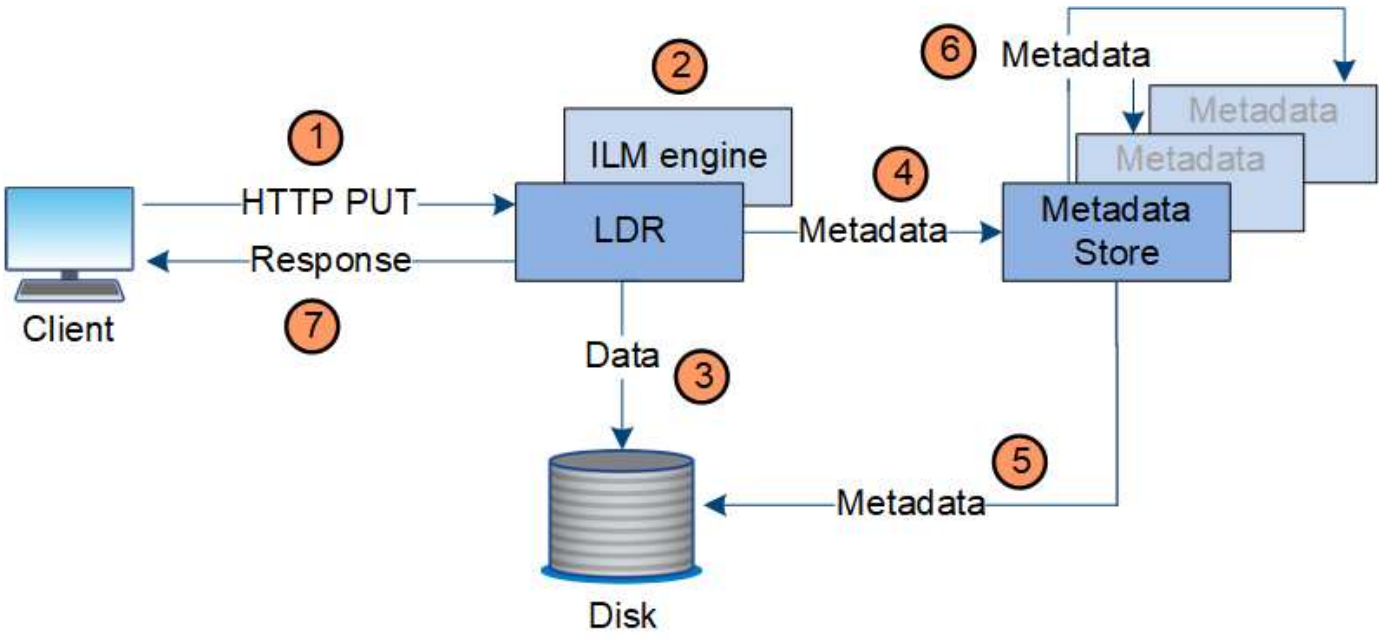
- Manage objects with ILM
- Use information lifecycle management

Ingest data flow

An ingest, or save, operation consists of a defined data flow between the client and the StorageGRID system.

Data flow

When a client ingests an object to the StorageGRID system, the LDR service on Storage Nodes processes the request and stores the metadata and data to disk.



1. The client application creates the object and sends it to the StorageGRID system through an HTTP PUT request.
2. The object is evaluated against the system's ILM policy.
3. The LDR service saves the object data as a replicated copy or as an erasure coded copy. (The diagram shows a simplified version of storing a replicated copy to disk.)
4. The LDR service sends the object metadata to the metadata store.
5. The metadata store saves the object metadata to disk.
6. The metadata store propagates copies of object metadata to other Storage Nodes. These copies are also saved to disk.
7. The LDR service returns an HTTP 200 OK response to the client to acknowledge that the object has been ingested.

Copy management

Object data is managed by the active ILM policy and its ILM rules. ILM rules make replicated or erasure coded copies to protect object data from loss.

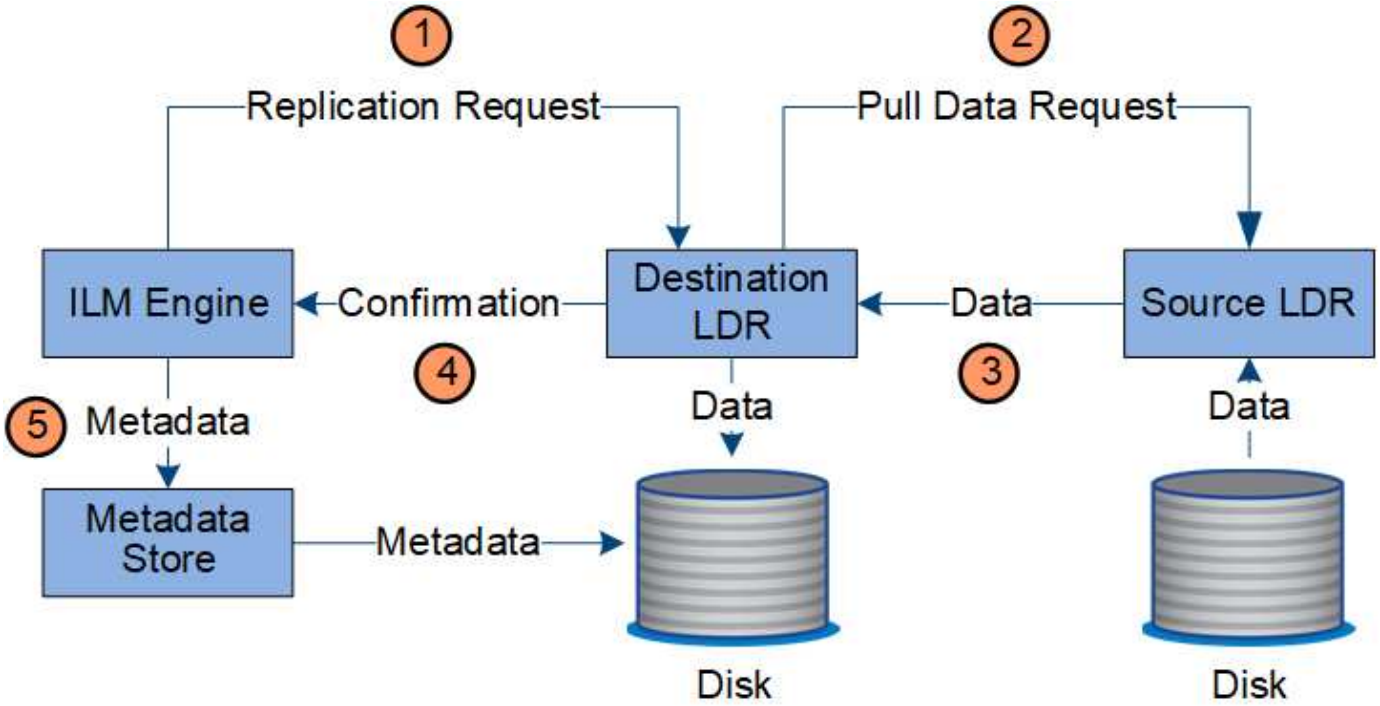
Different types or locations of object copies might be required at different times in the object's life. ILM rules are periodically evaluated to ensure that objects are placed as required.

Object data is managed by the LDR service.

Content protection: replication

If an ILM rule's content placement instructions require replicated copies of object data, copies are made and stored to disk by the Storage Nodes that make up the configured storage pool.

The ILM engine in the LDR service controls replication and ensures that the correct number of copies are stored in the correct locations and for the correct amount of time.

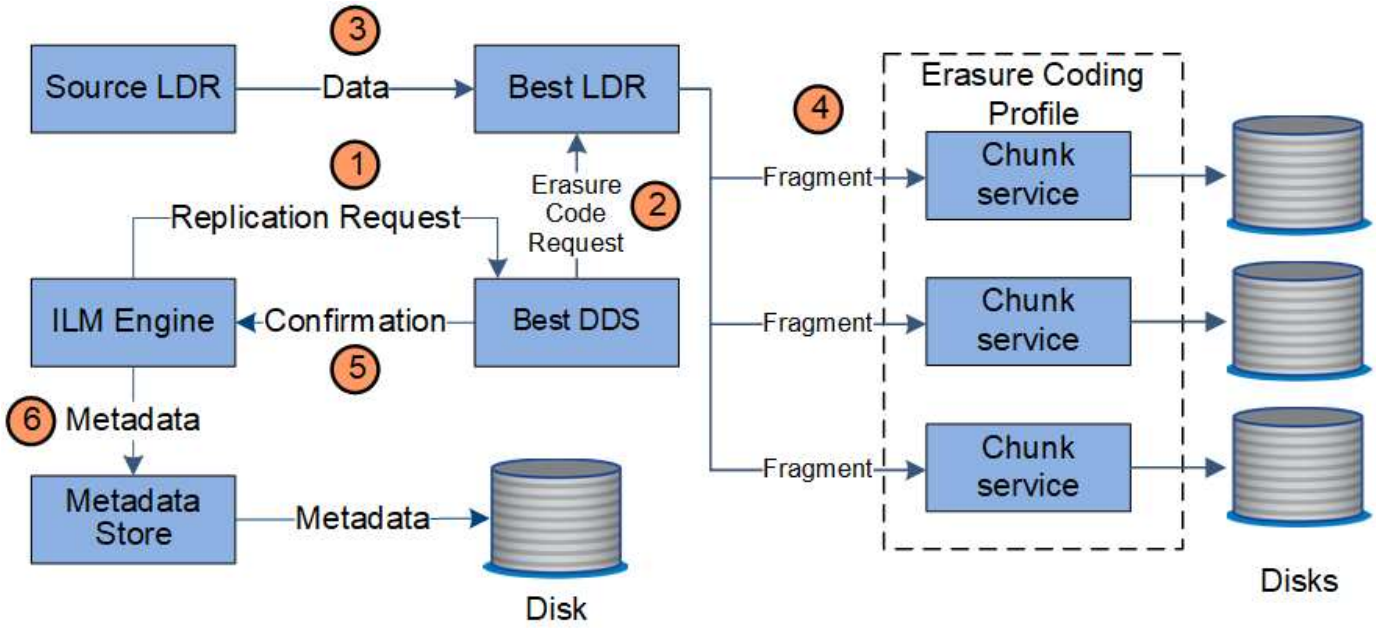


1. The ILM engine queries the ADC service to determine the best destination LDR service within the storage pool specified by the ILM rule. It then sends that LDR service a command to initiate replication.
2. The destination LDR service queries the ADC service for the best source location. It then sends a replication request to the source LDR service.
3. The source LDR service sends a copy to the destination LDR service.
4. The destination LDR service notifies the ILM engine that the object data has been stored.
5. The ILM engine updates the metadata store with object location metadata.

Content protection: erasure coding

If an ILM rule includes instructions to make erasure coded copies of object data, the applicable erasure coding scheme breaks object data into data and parity fragments and distributes these fragments across the Storage Nodes configured in the erasure coding profile.

The ILM engine, which is a component of the LDR service, controls erasure coding and ensures that the erasure coding profile is applied to object data.

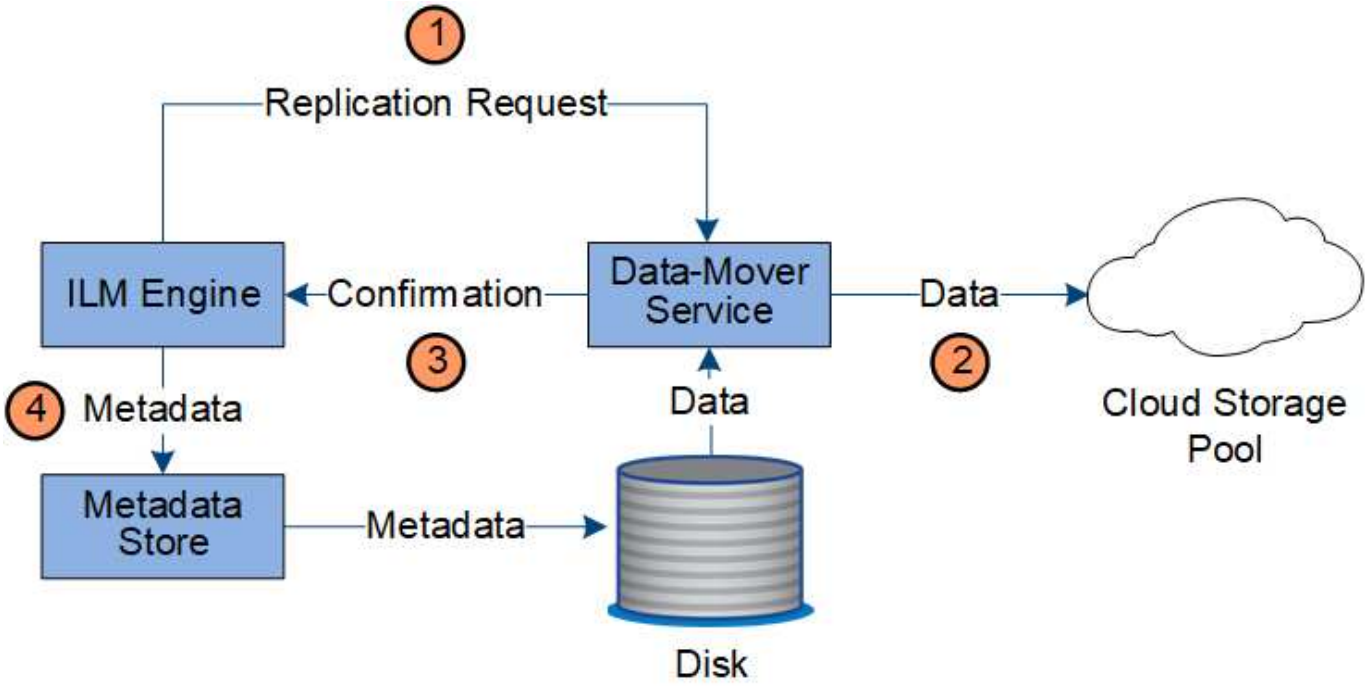


1. The ILM engine queries the ADC service to determine which DDS service can best perform the erasure coding operation. Once determined, the ILM engine sends an "initiate" request to that service.
2. The DDS service instructs an LDR to erasure code the object data.
3. The source LDR service sends a copy to the LDR service selected for erasure coding.
4. Once broken into the appropriate number of parity and data fragments, the LDR service distributes these fragments across the Storage Nodes (Chunk services) that make up the erasure coding profile's storage pool.
5. The LDR service notifies the ILM engine, confirming that object data is successfully distributed.
6. The ILM engine updates the metadata store with object location metadata.

Content protection: Cloud Storage Pool

If an ILM rule's content placement instructions require that a replicated copy of object data is stored on a Cloud Storage Pool, object data is duplicated to the external S3 bucket or Azure Blob storage container that was specified for the Cloud Storage Pool.

The ILM engine, which is a component of the LDR service, and the Data Mover service control the movement of objects to the Cloud Storage Pool.



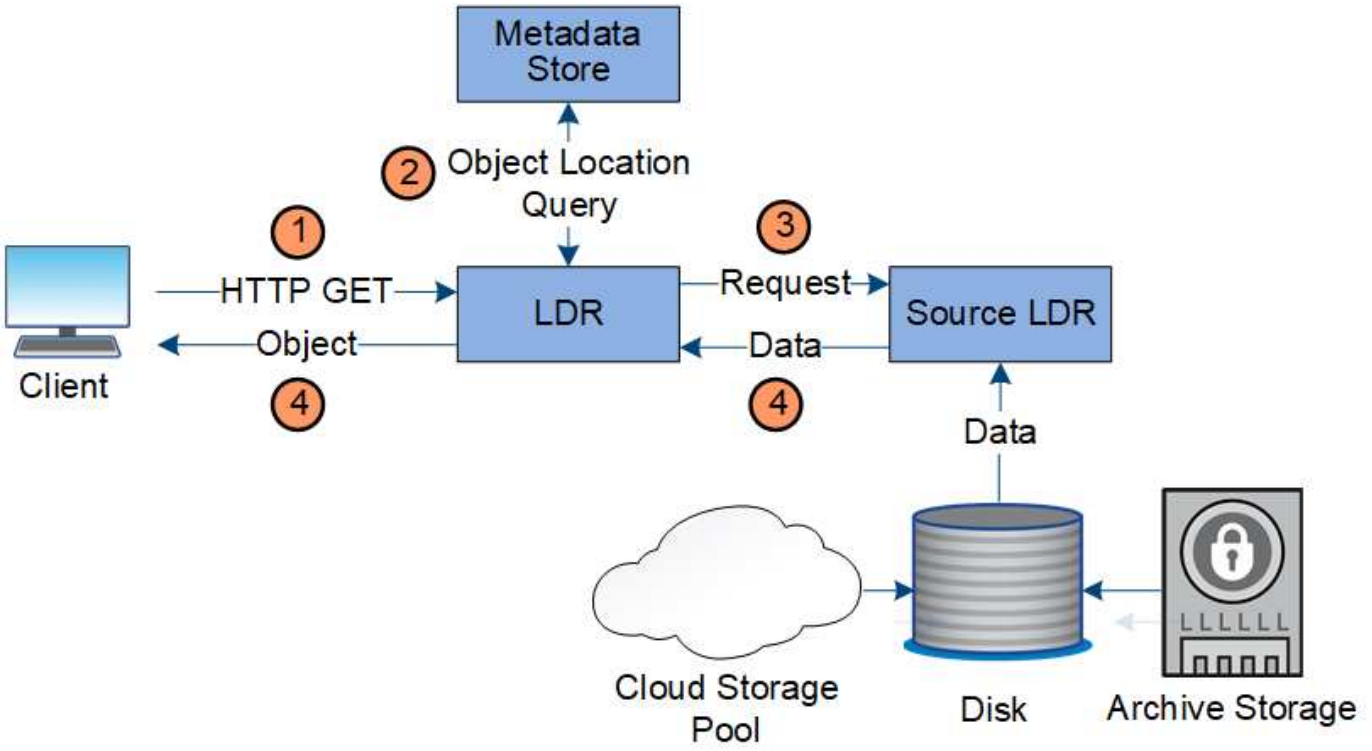
1. The ILM engine selects a Data Mover service to replicate to the Cloud Storage Pool.
2. The Data Mover service sends the object data to the Cloud Storage Pool.
3. The Data Mover service notifies the ILM engine that the object data has been stored.
4. The ILM engine updates the metadata store with object location metadata.

Retrieve data flow

A retrieve operation consists of a defined data flow between the StorageGRID system and the client. The system uses attributes to track the retrieval of the object from a Storage Node or, if necessary, a Cloud Storage Pool or Archive Node.

The Storage Node's LDR service queries the metadata store for the location of the object data and retrieves it from the source LDR service. Preferentially, retrieval is from a Storage Node. If the object is not available on a Storage Node, the retrieval request is directed to a Cloud Storage Pool or to an Archive Node.

If the only object copy is on AWS Glacier storage or the Azure Archive tier, the client application must issue an S3 POST Object restore request to restore a retrievable copy to the Cloud Storage Pool.



1. The LDR service receives a retrieval request from the client application.
2. The LDR service queries the metadata store for the object data location and metadata.
3. LDR service forwards the retrieval request to the source LDR service.
4. The source LDR service returns the object data from the queried LDR service and the system returns the object to the client application.

Delete data flow

All object copies are removed from the StorageGRID system when a client performs a delete operation or when the object's lifetime expires, triggering its automatic removal. There is a defined data flow for object deletion.

Deletion hierarchy

StorageGRID provides several methods for controlling when objects are retained or deleted. Objects can be deleted by client request or automatically. StorageGRID always prioritizes any S3 Object Lock settings over client delete requests, which are prioritized over S3 bucket lifecycle and ILM placement instructions.

- **S3 Object Lock**: If the global S3 Object Lock setting is enabled for the grid, S3 clients 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.

    - If S3 clients apply a default retain-until-date to the bucket, they don’t need to specify a retain-until-date for each object.

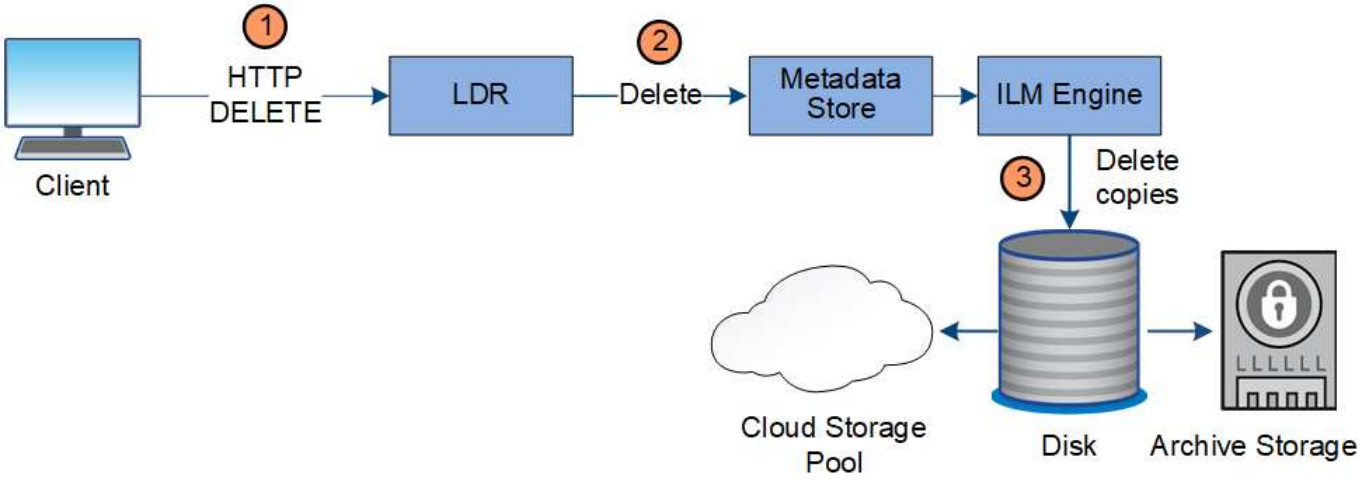
- **Client delete request**: An S3 or Swift client can issue a delete object request. When a client deletes an object, all copies of the object are removed from the StorageGRID system.
- **Delete objects in bucket**: Tenant Manager users can use this option to permanently remove all copies of the objects and object versions in selected buckets from the StorageGRID system.
- **S3 bucket lifecycle**: S3 clients can add a lifecycle configuration to their buckets that specifies an Expiration action. If a bucket lifecycle exists, StorageGRID automatically deletes all copies of an object when the date or number of days specified in the Expiration action are met, unless the client deletes the object first.
- **ILM placement instructions**: Assuming that the bucket does not have S3 Object Lock enabled and that there is no bucket lifecycle, StorageGRID automatically deletes an object when the last time period in the ILM rule ends and there are no further placements specified for the object.

> 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.

How S3 delete markers are deleted

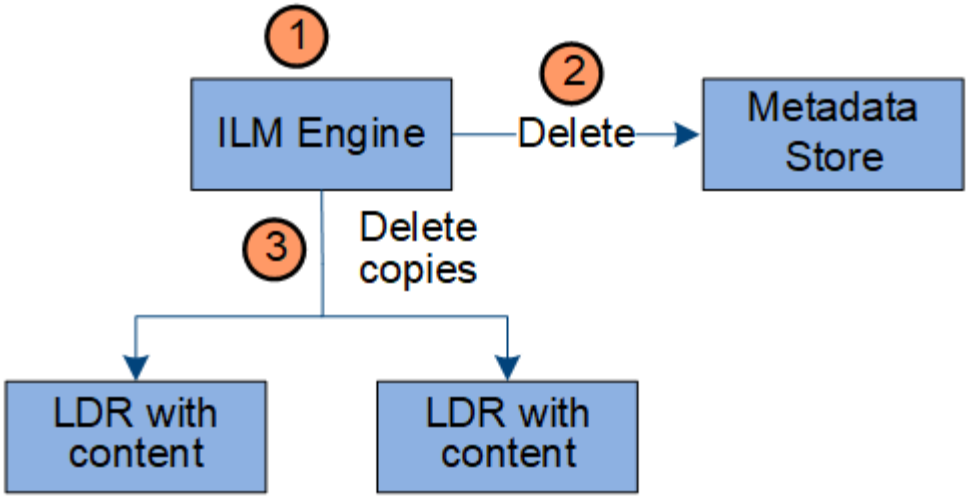
When a versioned object is deleted, StorageGRID creates a delete marker as the current version of the object. To remove the zero-byte delete marker from the bucket, the S3 client must explicitly delete the object version. Delete markers aren’t removed by ILM, bucket lifecycle rules, or Delete objects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

Data flow for client deletes



1. The LDR service receives a delete request from the client application.
2. The LDR service updates the metadata store so the object looks deleted to client requests, and instructs the ILM engine to remove all copies of object data.
3. The object is removed from the system. The metadata store is updated to remove object metadata.

Data flow for ILM deletes



1. The ILM engine determines that the object needs to be deleted.
2. The ILM engine notifies the metadata store. The metadata store updates object metadata so that the object looks deleted to client requests.
3. The ILM engine removes all copies of the object. The metadata store is updated to remove object metadata.

Use information lifecycle management

You use information lifecycle management (ILM) to control the placement, duration, and ingest behavior for all objects in your StorageGRID system. ILM rules determine how StorageGRID stores objects over time. You configure one or more ILM rules and then add them to an ILM policy.

A grid has only one active policy at a time. A policy can contain multiple rules.

ILM rules define:

- Which objects should be stored. A rule can apply to all objects, or you can specify filters to identify which objects a rule applies to. For example, a rule can apply only to objects associated with certain tenant accounts, specific S3 buckets or Swift containers, or specific metadata values.
- The storage type and location. Objects can be stored on Storage Nodes, in Cloud Storage Pools, or on Archive Nodes.
- The type of object copies made. Copies can be replicated or erasure coded.
- For replicated copies, the number of copies made.
- For erasure coded copies, the erasure-coding scheme used.
- The changes over time to an object's storage location and type of copies.
- How object data is protected as objects are ingested into the grid (synchronous placement or dual commit).

Note that 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.

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.

How an ILM policy evaluates objects

The active ILM policy for your StorageGRID system controls the placement, duration, and ingest behavior 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:

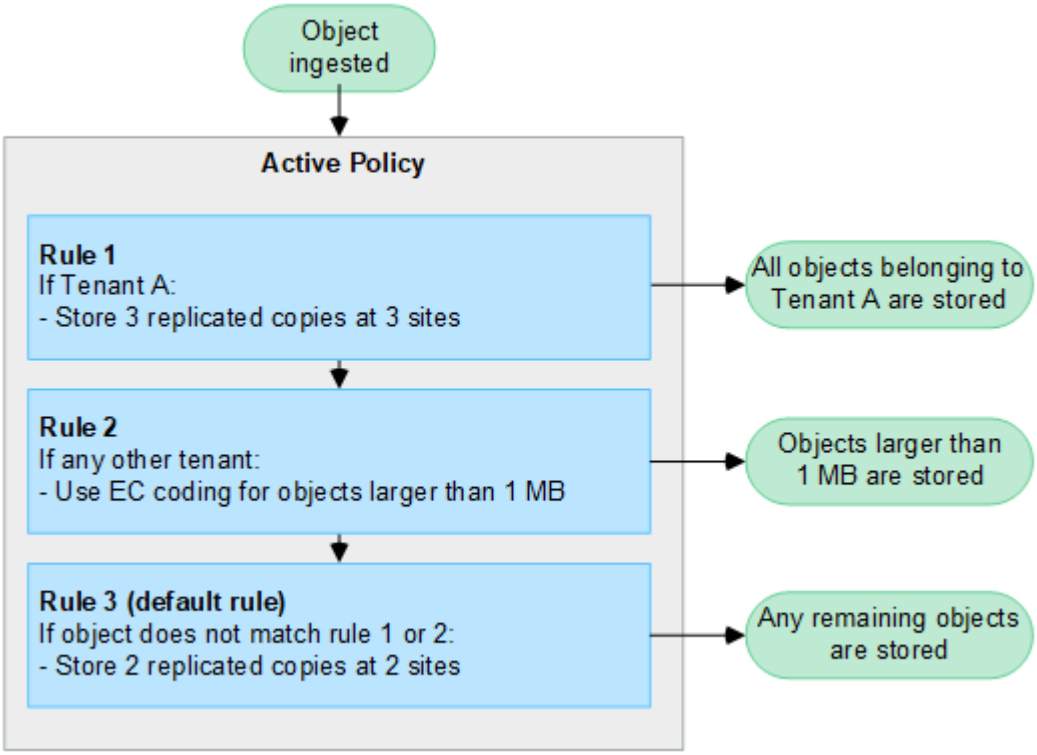
1. If the filters for the first rule in the policy match an object, the object is ingested according to that rule’s ingest behavior and stored according to that rule’s placement instructions.
2. If the filters for the first rule don’t match the object, the object is evaluated against each subsequent 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 and can’t use any filters. It must apply to all tenants, all buckets, and all object versions.

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).



Related information

- Manage objects with ILM

Copyright information

Trademark information