Operations on buckets

StorageGRID

NetApp
September 04, 2024

Table of Contents

Operations on buckets

The StorageGRID system supports a maximum of 1,000 buckets for each S3 tenant account.

Bucket name restrictions follow the AWS US Standard region restrictions, but you should further restrict them to DNS naming conventions in order to support S3 virtual hosted-style requests.

Amazon Web Services (AWS) Documentation: Bucket Restrictions and Limitations

Configure S3 API endpoint domain names

The GET Bucket (List Objects) and GET Bucket versions operations support StorageGRID consistency controls.

You can check whether updates to last access time are enabled or disabled for individual buckets.

The following table describes how StorageGRID implements S3 REST API bucket operations. To perform any of these operations, the necessary access credentials must be provided for the account.

Operation	Implementation
DELETE Bucket	Implemented with all Amazon S3 REST API behavior.
DELETE Bucket cors	This operation deletes the CORS configuration for the bucket.
DELETE Bucket encryption	This operation deletes the default encryption from the bucket. Existing encrypted objects remain encrypted, but any new objects added to the bucket are not encrypted.
DELETE Bucket lifecycle	This operation deletes the lifecycle configuration from the bucket.
DELETE Bucket policy	This operation deletes the policy attached to the bucket.
DELETE Bucket replication	This operation deletes the replication configuration attached to the bucket.
DELETE Bucket tagging	This operation uses the `tagging` subresource to remove all tags from a bucket.

Operation	Implementation
GET Bucket (List Objects), version 1 and version 2	This operation returns some or all (up to 1,000) of the objects in a bucket. The Storage Class for objects can have either of two values, even if the object was ingested with the `REDUCED_REDUNDANCY` storage class option: • `STANDARD`, which indicates the object is stored in a storage pool consisting of Storage Nodes. • `GLACIER`, which indicates that the object has been moved to the external bucket specified by the Cloud Storage Pool. If the bucket contains large numbers of deleted keys that have the same prefix, the response might include some `CommonPrefixes` that do not contain keys.
GET Bucket acl	This operation returns a positive response and the ID, DisplayName, and Permission of the bucket owner, indicating that the owner has full access to the bucket.
GET Bucket cors	This operation returns the `cors` configuration for the bucket.
GET Bucket encryption	This operation returns the default encryption configuration for the bucket.
GET Bucket lifecycle	This operation returns the lifecycle configuration for the bucket.
GET Bucket location	This operation returns the region that was set using the `LocationConstraint` element in the PUT Bucket request. If the bucket's region is `us-east-1`, an empty string is returned for the region.
GET Bucket notification	This operation returns the notification configuration attached to the bucket.
GET Bucket Object versions	With READ access on a bucket, this operation with the `versions` subresource lists metadata of all of the versions of objects in the bucket.
GET Bucket policy	This operation returns the policy attached to the bucket.
GET Bucket replication	This operation returns the replication configuration attached to the bucket.
GET Bucket tagging	This operation uses the `tagging` subresource to return all tags for a bucket.
GET Bucket versioning	This implementation uses the `versioning` subresource to return the versioning state of a bucket. • *blank*: Versioning has never been enabled (bucket is "Unversioned") • Enabled: Versioning is enabled • Suspended: Versioning was previously enabled and is suspended

Operation	Implementation
GET Object Lock Configuration	This operation returns the bucket default retention mode and default retention period, if configured. See GET Object Lock Configuration for detailed information.
HEAD Bucket	This operation determines if a bucket exists and you have permission to access it. This operation returns: • `x-ntap-sg-bucket-id`: The UUID of the bucket in UUID format. • `x-ntap-sg-trace-id`: The unique trace ID of the associated request.
PUT Bucket	This operation creates a new bucket. By creating the bucket, you become the bucket owner. • Bucket names must comply with the following rules: ◦ Must be unique across each StorageGRID system (not just unique within the tenant account). ◦ Must be DNS compliant. ◦ Must contain at least 3 and no more than 63 characters. ◦ Can be a series of one or more labels, with adjacent labels separated by a period. Each label must start and end with a lowercase letter or a number and can only use lowercase letters, numbers, and hyphens. ◦ Must not look like a text-formatted IP address. ◦ Should not use periods in virtual hosted style requests. Periods will cause problems with server wildcard certificate verification. • By default, buckets are created in the `us-east-1` region; however, you can use the `LocationConstraint` request element in the request body to specify a different region. When using the `LocationConstraint` element, you must specify the exact name of a region that has been defined using the Grid Manager or the Grid Management API. Contact your system administrator if you do not know the region name you should use. **Note**: An error will occur if your PUT Bucket request uses a region that has not been defined in StorageGRID. • You can include the `x-amz-bucket-object-lock-enabled` request header to create a bucket with S3 Object Lock enabled. See Use S3 Object Lock. You must enable S3 Object Lock when you create the bucket. You cannot add or disable S3 Object Lock after a bucket is created. S3 Object Lock requires bucket versioning, which is enabled automatically when you create the bucket.

Operation	Implementation
PUT Bucket cors	This operation sets the CORS configuration for a bucket so that the bucket can service cross-origin requests. Cross-origin resource sharing (CORS) is a security mechanism that allows client web applications in one domain to access resources in a different domain. For example, suppose you use an S3 bucket named `images` to store graphics. By setting the CORS configuration for the `images` bucket, you can allow the images in that bucket to be displayed on the website `http://www.example.com`.
PUT Bucket encryption	This operation sets the default encryption state of an existing bucket. When bucket-level encryption is enabled, any new objects added to the bucket are encrypted.StorageGRID supports server-side encryption with StorageGRID-managed keys. When specifying the server-side encryption configuration rule, set the `SSEAlgorithm` parameter to `AES256`, and do not use the `KMSMasterKeyID` parameter. Bucket default encryption configuration is ignored if the object upload request already specifies encryption (that is, if the request includes the `x-amz-server-side-encryption-*` request header).
PUT Bucket lifecycle	This operation creates a new lifecycle configuration for the bucket or replaces an existing lifecycle configuration. StorageGRID supports up to 1,000 lifecycle rules in a lifecycle configuration. Each rule can include the following XML elements: • Expiration (Days, Date) • NoncurrentVersionExpiration (NoncurrentDays) • Filter (Prefix, Tag) • Status • ID StorageGRID does not support these actions: • AbortIncompleteMultipartUpload • ExpiredObjectDeleteMarker • Transition To understand how the Expiration action in a bucket lifecycle interacts with ILM placement instructions, see “How ILM operates throughout an object’s life” in the instructions for managing objects with information lifecycle management. **Note**: Bucket lifecycle configuration can be used with buckets that have S3 Object Lock enabled, but bucket lifecycle configuration is not supported for legacy Compliant buckets.

Operation	Implementation

PUT Bucket notification	This operation configures notifications for the bucket using the notification configuration XML included in the request body. You should be aware of the following implementation details:
	
	• StorageGRID supports Simple Notification Service (SNS) topics as destinations. Simple Queue Service (SQS) or Amazon Lambda endpoints are not supported.
	• The destination for notifications must be specified as the URN of an StorageGRID endpoint. Endpoints can be created using the Tenant Manager or the Tenant Management API.
	
	The endpoint must exist for notification configuration to succeed. If the endpoint does not exist, a `400 Bad Request` error is returned with the code `InvalidArgument`.
	
	• You cannot configure a notification for the following event types. These event types are **not** supported.
	
	◦ `s3:ReducedRedundancyLostObject`
	
	◦ `s3:ObjectRestore:Completed`
	
	• Event notifications sent from StorageGRID use the standard JSON format except that they do not include some keys and use specific values for others, as shown in the following listing:
	• **eventSource**
	
	`sgws:s3`
	
	• **awsRegion**
	
	not included
	
	• **x-amz-id-2**
	
	not included
	
	• **arn**
	
	`urn:sgws:s3:::bucket_name`

PUT Bucket policy	This operation sets the policy attached to the bucket.

Operation	Implementation
PUT Bucket replication	This operation configures StorageGRID CloudMirror replication for the bucket using the replication configuration XML provided in the request body. For CloudMirror replication, you should be aware of the following implementation details: • StorageGRID only supports V1 of the replication configuration. This means that StorageGRID does not support the use of the `Filter` element for rules, and follows V1 conventions for deletion of object versions. For details, see the Amazon S3 documentation on replication configuration. • Bucket replication can be configured on versioned or unversioned buckets. • You can specify a different destination bucket in each rule of the replication configuration XML. A source bucket can replicate to more than one destination bucket. • Destination buckets must be specified as the URN of StorageGRID endpoints as specified in the Tenant Manager or the Tenant Management API. The endpoint must exist for replication configuration to succeed. If the endpoint does not exist, the request fails as a `400 Bad Request`. The error message states: `Unable to save the replication policy. The specified endpoint URN does not exist: URN.` • You do not need to specify a `Role` in the configuration XML. This value is not used by StorageGRID and will be ignored if submitted. • If you omit the storage class from the configuration XML, StorageGRID uses the `STANDARD` storage class by default. • If you delete an object from the source bucket or you delete the source bucket itself, the cross-region replication behavior is as follows: ◦ If you delete the object or bucket before it has been replicated, the object/bucket is not replicated and you are not notified. ◦ If you delete the object or bucket after it has been replicated, StorageGRID follows standard Amazon S3 delete behavior for V1 of cross-region replication.
PUT Bucket tagging	This operation uses the `tagging` subresource to add or update a set of tags for a bucket. When adding bucket tags, be aware of the following limitations: • Both StorageGRID and Amazon S3 support up to 50 tags for each bucket. • Tags associated with a bucket must have unique tag keys. A tag key can be up to 128 Unicode characters in length. • Tag values can be up to 256 Unicode characters in length. • Key and values are case sensitive.

Operation	Implementation
PUT Bucket versioning	This implementation uses the `versioning` subresource to set the versioning state of an existing bucket. You can set the versioning state with one of the following values: • Enabled: Enables versioning for the objects in the bucket. All objects added to the bucket receive a unique version ID. • Suspended: Disables versioning for the objects in the bucket. All objects added to the bucket receive the version ID `null`.
PUT Object Lock Configuration	This operation configures or removes the bucket default retention mode and default retention period. If the default retention period is modified, the retain-until-date of existing object versions remains the same and is not recalculated using the new default retention period. See PUT Object Lock Configuration for detailed information.

Related information

Consistency controls

GET Bucket last access time request

Bucket and group access policies

S3 operations tracked in audit logs

Manage objects with ILM

Use tenant account

Create S3 lifecycle configuration

You can create an S3 lifecycle configuration to control when specific objects are deleted from the StorageGRID system.

The simple example in this section illustrates how an S3 lifecycle configuration can control when certain objects are deleted (expired) from specific S3 buckets. The example in this section is for illustration purposes only. For complete details on creating S3 lifecycle configurations, see Amazon Simple Storage Service Developer Guide: Object lifecycle management. Note that StorageGRID only supports Expiration actions; it does not support Transition actions.

What lifecycle configuration is

A lifecycle configuration is a set of rules that are applied to the objects in specific S3 buckets. Each rule specifies which objects are affected and when those objects will expire (on a specific date or after some number of days).

StorageGRID supports up to 1,000 lifecycle rules in a lifecycle configuration. Each rule can include the

following XML elements:

- Expiration: Delete an object when a specified date is reached or when a specified number of days is reached, starting from when the object was ingested.
- NoncurrentVersionExpiration: Delete an object when a specified number of days is reached, starting from when the object became noncurrent.
- Filter (Prefix, Tag)
- Status
- ID

If you apply a lifecycle configuration to a bucket, the lifecycle settings for the bucket always override StorageGRID ILM settings. StorageGRID uses the Expiration settings for the bucket, not ILM, to determine whether to delete or retain specific objects.

As a result, an object might be removed from the grid even though the placement instructions in an ILM rule still apply to the object. Or, an object might be retained on the grid even after any ILM placement instructions for the object have lapsed. For details, see How ILM operates throughout an object's life.

> Bucket lifecycle configuration can be used with buckets that have S3 Object Lock enabled, but bucket lifecycle configuration is not supported for legacy Compliant buckets.

StorageGRID supports the use of the following bucket operations to manage lifecycle configurations:

- DELETE Bucket lifecycle
- GET Bucket lifecycle
- PUT Bucket lifecycle

Create lifecycle configuration

As the first step in creating a lifecycle configuration, you create a JSON file that includes one or more rules. For example, this JSON file includes three rules, as follows:

1. Rule 1 applies only to objects that match the prefix `category1/` and that have a `key2` value of `tag2`. The `Expiration` parameter specifies that objects matching the filter will expire at midnight on 22 August 2020.
2. Rule 2 applies only to objects that match the prefix `category2/`. The `Expiration` parameter specifies that objects matching the filter will expire 100 days after they are ingested.

   > Rules that specify a number of days are relative to when the object was ingested. If the current date exceeds the ingest date plus the number of days, some objects might be removed from the bucket as soon as the lifecycle configuration is applied.

3. Rule 3 applies only to objects that match the prefix `category3/`. The `Expiration` parameter specifies that any noncurrent versions of matching objects will expire 50 days after they become noncurrent.

```
{
    "Rules": [
        {
            "ID": "rule1",
            "Filter": {
                "And": {
                    "Prefix": "category1/",
                    "Tags": [
                        {
                            "Key": "key2",
                            "Value": "tag2"
                        }
                    ]
                }
            },
            "Expiration": {
                "Date": "2020-08-22T00:00:00Z"
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule2",
            "Filter": {
                "Prefix": "category2/"
            },
            "Expiration": {
                "Days": 100
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule3",
            "Filter": {
                "Prefix": "category3/"
            },
            "NoncurrentVersionExpiration": {
                "NoncurrentDays": 50
            },
            "Status": "Enabled"
        }
    ]
}
```

Apply lifecycle configuration to bucket

After you have created the lifecycle configuration file, you apply it to a bucket by issuing a PUT Bucket lifecycle request.

This request applies the lifecycle configuration in the example file to objects in a bucket named `testbucket`.

```
aws s3api --endpoint-url <StorageGRID endpoint> put-bucket-lifecycle-
configuration
--bucket testbucket --lifecycle-configuration file://bktjson.json
```

To validate that a lifecycle configuration was successfully applied to the bucket, issue a GET Bucket lifecycle request. For example:

```
aws s3api --endpoint-url <StorageGRID endpoint> get-bucket-lifecycle-
configuration
 --bucket testbucket
```

A successful response lists the lifecycle configuration you just applied.

Validate that bucket lifecycle expiration applies to object

You can determine if an expiration rule in the lifecycle configuration applies to a specific object when issuing a PUT Object, HEAD Object, or GET Object request. If a rule applies, the response includes an `Expiration` parameter that indicates when the object expires and which expiration rule was matched.

> Because bucket lifecycle overrides ILM, the `expiry-date` shown is the actual date the object will be deleted. For details, see How object retention is determined.

For example, this PUT Object request was issued on 22 Jun 2020 and places an object in the `testbucket` bucket.

```
aws s3api --endpoint-url <StorageGRID endpoint> put-object
--bucket testbucket --key obj2test2 --body bktjson.json
```

The success response indicates that the object will expire in 100 days (01 Oct 2020) and that it matched Rule 2 of the lifecycle configuration.

```
{
      *"Expiration": "expiry-date=\"Thu, 01 Oct 2020 09:07:49 GMT\", rule-
id=\"rule2\"",
      "ETag": "\"9762f8a803bc34f5340579d4446076f7\""
}
```

For example, this HEAD Object request was used to get metadata for the same object in the testbucket bucket.

```
aws s3api --endpoint-url <StorageGRID endpoint> head-object
--bucket testbucket --key obj2test2
```

The success response includes the object's metadata and indicates that the object will expire in 100 days and that it matched Rule 2.

```
{
      "AcceptRanges": "bytes",
      *"Expiration": "expiry-date=\"Thu, 01 Oct 2020 09:07:48 GMT\", rule-
id=\"rule2\"",
      "LastModified": "2020-06-23T09:07:48+00:00",
      "ContentLength": 921,
      "ETag": "\"9762f8a803bc34f5340579d4446076f7\""
      "ContentType": "binary/octet-stream",
      "Metadata": {}
}
```

Use S3 Object Lock default bucket retention

If a bucket has S3 Object Lock enabled, you can specify a default retention mode and default retention period that is applied to each object added to the bucket.

- S3 Object Lock can be enabled or disabled for a bucket during bucket creation.
- If S3 Object Lock is enabled for a bucket, you can configure default retention for the bucket.
- Default retention configuration specifies:
  - Default retention mode: StorageGRID supports only "COMPLIANCE" mode.
  - Default retention period in days or years.

GET Object Lock Configuration

The GET Object Lock Configuration request allows you to determine if Object Lock is enabled for a bucket and, if it is enabled, see if there is a default retention mode and retention period configured for the bucket.

When new object versions are ingested to the bucket, the default retention mode is applied if `x-amz-object-lock-mode` is not specified. The default retention period is used to calculate the retain-until-date if `x-amz-object-lock-retain-until-date` is not specified.

You must have the s3:GetBucketObjectLockConfiguration permission, or be account root, to complete this operation.

Request example

```
GET /bucket?object-lock HTTP/1.1
Host: host
Accept-Encoding: identity
User-Agent: aws-cli/1.18.106 Python/3.8.2 Linux/4.4.0-18362-Microsoft
botocore/1.17.29
x-amz-date: date
x-amz-content-sha256: authorization string
Authorization: authorization string
```

Response example

```
HTTP/1.1 200 OK
x-amz-id-2:
iVmcB7OXXJRkRH1FiVq1151/T24gRfpwpuZrEG11Bb9ImOMAAe98oxSpXlknabA0LTvBYJpSIX
k=
x-amz-request-id: B34E94CACB2CEF6D
Date: Fri, 04 Sep 2020 22:47:09 GMT
Transfer-Encoding: chunked
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<ObjectLockConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <ObjectLockEnabled>Enabled</ObjectLockEnabled>
    <Rule>
        <DefaultRetention>
            <Mode>COMPLIANCE</Mode>
            <Years>6</Years>
        </DefaultRetention>
    </Rule>
</ObjectLockConfiguration>
```

PUT Object Lock Configuration

The PUT Object Lock Configuration request allows you to modify the default retention mode and default retention period for a bucket that has Object Lock enabled. You can also remove previously configured default retention settings.

When new object versions are ingested to the bucket, the default retention mode is applied if `x-amz-object-lock-mode` is not specified. The default retention period is used to calculate the retain-until-date if `x-amz-object-lock-retain-until-date` is not specified.

If the default retention period is modified after ingest of an object version, the retain-until-date of the object version remains the same and is not recalculated using the new default retention period.

You must have the s3:PutBucketObjectLockConfiguration permission, or be account root, to complete this operation.

The `Content-MD5` request header must be specified in the PUT request.

Request example

```
PUT /bucket?object-lock HTTP/1.1
Accept-Encoding: identity
Content-Length: 308
Host: host
Content-MD5: request header
User-Agent: s3sign/1.0.0 requests/2.24.0 python/3.8.2
X-Amz-Date: date
X-Amz-Content-SHA256: authorization string
Authorization: authorization string

<ObjectLockConfiguration>
    <ObjectLockEnabled>Enabled</ObjectLockEnabled>
    <Rule>
        <DefaultRetention>
            <Mode>COMPLIANCE</Mode>
            <Years>6</Years>
        </DefaultRetention>
    </Rule>
</ObjectLockConfiguration>
```

Custom operations on buckets

The StorageGRID system supports custom bucket operations that are added on to the S3 REST API and are specific to the system.

The following table lists the custom bucket operations supported by StorageGRID.

Operation	Description	For more information
GET Bucket consistency	Returns the consistency level being applied to a particular bucket.	GET Bucket consistency request
PUT Bucket consistency	Sets the consistency level applied to a particular bucket.	PUT Bucket consistency request
GET Bucket last access time	Returns whether last access time updates are enabled or disabled for a particular bucket.	GET Bucket last access time request
PUT Bucket last access time	Allows you to enable or disable last access time updates for a particular bucket.	PUT Bucket last access time request

Operation	Description	For more information
DELETE Bucket metadata notification configuration	Deletes the metadata notification configuration XML associated with a particular bucket.	DELETE Bucket metadata notification configuration request
GET Bucket metadata notification configuration	Returns the metadata notification configuration XML associated with a particular bucket.	GET Bucket metadata notification configuration request
PUT Bucket metadata notification configuration	Configures the metadata notification service for a bucket.	PUT Bucket metadata notification configuration request
PUT Bucket with compliance settings	Deprecated and not supported: You can no longer create new buckets with Compliance enabled.	Deprecated: PUT Bucket with compliance settings
GET Bucket compliance	Deprecated but supported: Returns the compliance settings currently in effect for an existing legacy Compliant bucket.	Deprecated: GET Bucket compliance request
PUT Bucket compliance	Deprecated but supported: Allows you to modify the compliance settings for an existing legacy Compliant bucket.	Deprecated: PUT Bucket compliance request

Related information

S3 operations tracked in the audit logs

Copyright information

Trademark information