IBM Cloud Object Storage System™
Version 3.14.3

Cloud Storage Object API 2.5 Development

This edition applies to IBM Cloud Object Storage System and is valid until replaced by new editions.

Contents

Document information

Intended purpose and audience

The Cloud Storage Object (CSO) Application Programming Interface (API) enables application developers to use existing Amazon Simple Storage Service (S3) applications to access object vaults on a system. It is a REST API available as part of the Dispersed Storage Object (DSO) API Family. The CSO API supports the most commonly used subset of Amazon S3 API operations. This document highlights differences between these two APIs. It should be used in combination with the Amazon Simple Storage Service API Reference Version 2006-03-01.

Chapter 1. Overview

IBM Cloud Object Storage System™ APIs

The CSO API is automatically deployed on any Accesser node. Any object vault that is deployed to an Accesser node can be through any API in the API Family.

Table 1. Dispersed Object API List

REST API	Write Mode	Listing	Store User Metadata	Compatibility
Simple Object HTTP	Write-by-ID	Recovery Only		
Cloud Storage Object	Write-by-Name	Prefix-based Name Search and Recovery Listing	Yes	Amazon S3 v2006-03-01
OpenStack Object Storage				OpenStack v1.1

API versus ClevOS versioning

Important: IBM Cloud Object Storage System™ APIs follow a different version convention than ClevOS. They do not:

- Reconcile with ClevOS versions.
- Increment with every new ClevOS release automatically.

Minor versions add some new functions and are compatible with an earlier version within a major version.

Major versions are major changes to the API and cannot be compatible with an earlier version.

Write modes

A client specifies an object name when it performs a write operation by using the CSO API. It is known as *write-by-name*. Objects that are written by name can also read by that name.

Deployment and client configuration

When an object vault is deployed to an Accesser node, it is automatically made available through each of the other supported REST APIs (Swift, SOH).

Clients can always access the CSO API via the following URL scheme:

```
[source,http] ---- http://{accesser-ip}/s3/{vault-name}/{object-name} ----
```

Table 2. URL Parameters for CSO REST calls

Parameter	Explanation
accesser-ip	The IP address or hostname of the Accesser node. The actual address might differ for client applications if a proxy or load balancer is being used.
s3	The literal that indicates the CSO API.
vault-name	The vault name that is being accessed. A vault is synonymous with a bucket in the S3 API.
object-name	The full object name that is being accessed. It is omitted for bucket operations such as listing.

If the CSO API is configured as the default API for an Accesser node, clients can also access the API via the following URL scheme:

```
[source,http] ---- http://{accesser-ip}/{vault-name}/{object-name} ----
```

Access through HTTPS is recommended, as some operations are supported only through HTTPS.

Note: See the Vaults section of the *Manager Administration Guide* for more information on deploying an object vault and about SSL and PKI access through an Accesser node.

If an S3 Virtual Host Suffix is configured for an Accesser node, clients can also access the API via the following URL scheme:

```
[source,http] ---- http://{bucket-name}.{s3-virtual-host-suffix}/{object-name} ----
```

Note: See the 'Create Access Pool' section and the 'Edit Access Pool' section of the *Manager Administration Guide* for more information on configuring virtual host-style access through an Accesser node.

Chapter 2. API variations

Differences between S3 and CSO APIs

The following information highlights some functional differences between S3 API and CSO API.

Table 3. Differences between S3 and CSO APIs

Feature	S3	CSO
Object Size Limitations	5 TB	No explicit limit for single request uploads. Objects that are uploaded via multipart upload, part size and part count limits are enforced.
Retained Version Count Limitations	No explicit limit.	A maximum of 1000 retained versions are allowed per object.
Vault (Bucket) Granular ACL		Users who are configured in the Manager Web Interface can be granted **read/write**, **read-only**, or **no-access** permissions to any vault. These settings apply to the entire vault.
Vault (Bucket) Granular Data Reliability	Allows a storage class to be configured for each object. All objects that are stored in any vault share reliability characteristics.	Vault reliability characteristics are determined at vault creation time.
Traditional Authentication Mechanisms	Uses a custom HTTP scheme based on a keyed-HMAC.	In addition to Access Key authentication, these authentication methods are also supported: • HTTP Basic over HTTP and HTTPS • PKI over HTTPS • Anonymous **Note:** For more information on configuring and authentication on a system, see the *Manager Administration Guide*.
Separated Audit and Logging Functions		Accesser node collects both access logs and audit trail information but does not expose it through the API.
Encryption and Cryptographic Integrity		• An Object Vault can be configured to store information in an encrypted form. • It must be configured at the vault/bucket level through the System Manager. • These settings cannot be viewed or edited through the API. • Request signing is also supported. Non-cryptographic • MD5 checksums are calculated and stored with objects.
Lifecycle Configuration		Does not support policy-based migration of data to alternative storage classes, nor does it support automatic expiration, deletion of object data, or archiving of data.
Website Hosting		Does not support static website hosting.

Table 3. Differences between S3 and CSO APIs (continued)

Feature	S3	CSO
Vault (Bucket) Location Constraints	Allows buckets to be created with specific location constraints.	• Can configure a system to allow data in one vault to be in a separate geographical location from data on another vault. • It is configured when vaults are created in the Manager Web Interface.
Hard Quota Function	Does not support quotas for buckets.	A hard quota can be configured for an object vault. HTTP status code **507** (Insufficient Storage) is returned for a write request that would cause a hard quota to be exceeded.

Chapter 3. API reference

API Overview

IBM Cloud Object Storage supports the most commonly used subset of Amazon S3 API operations and includes several IBM extensions, listed in the following tables:

Service operation:

Account operation	Note
GET account	Used to retrieve a list of all buckets created by the requesting account.

Bucket operations:

Bucket operation	Note
DELETE Bucket	Deletes an empty bucket.
DELETE Bucket CORS	Deletes any cross-origin resource sharing configuration set on a bucket.
GET Bucket	Lists objects contained in a bucket. Limited to listing 1,000 objects at once.
GET Bucket ACL	Retrieves the access control list for a bucket.
GET Bucket CORS	Retrieves any cross-origin resource sharing configuration set on a bucket.
GET Bucket Protection	Retrieves the protection configuration for a bucket.
HEAD Bucket	Retrieves a bucket's headers.
GET multipart uploads	Lists multipart uploads that have not completed or been canceled.
PUT Bucket	Buckets have naming restrictions. Systems in Vault Mode have a vault limit that also limits the number of buckets.
PUT Bucket ACL	Creates an access control list for a bucket.
PUT Bucket CORS	Creates a cross-origin resource sharing configuration for a bucket.
PUT Bucket Protection	Add protection to a bucket.
GET Bucket Tagging	(Vault Mode only)
GET Bucket Object Versions	(Vault Mode only)
GET Bucket Versioning	(Vault Mode only)
PUT Bucket Tagging	(Vault Mode only)
PUT Bucket Versioning	(Vault Mode only)

Object operations:

Object operation	Note
DELETE Object	Deletes an object from a bucket.
Delete Multiple Objects (POST)	Deletes multiple objects from a bucket.

Object operation	Note
GET Object	Retrieves an object from a bucket.
GET Object ACL	Retrieves an object's access control list.
GET Object Legal Hold	Retrieves a list of legal holds on and the retention period of an object.
HEAD Object	Retrieves the metadata of an object.
OPTIONS Object	Checks CORS configuration to see if a specific request can be sent.
POST Object	Adds an object to a bucket using HTML forms.
POST Object Legal Hold	Add or remove a single legal hold from an object.
POST Object Extend Retention	Extends the retention period of an object.
PUT Object	Adds an object to a bucket.
PUT Object ACL	Creates an access control list for an object.
PUT Object (Copy)	Creates a copy of an object.
Initiate Multipart Upload (POST)	Creates an upload ID for a given set of parts to be uploaded.
Upload Part (PUT)	Uploads a part of an object associated with an upload ID.
Upload Part (Copy) (PUT)	Uploads a part of an existing object associated with an upload ID.
Complete Multipart Upload (POST)	Assembles an object from parts associated with an upload ID.
Abort Multipart Upload (DELETE)	Aborts upload and deletes outstanding parts associated with an upload ID.
List Parts (GET)	Returns a list of parts associated with an upload ID

The following operations are *not supported* in the IBM COS implementation of the S3 API:

- DELETE Bucket analytics
- DELETE Bucket inventory
- DELETE Bucket lifecycle
- DELETE Bucket metrics
- DELETE Bucket policy
- DELETE Bucket replication
- DELETE Bucket website
- GET Bucket accelerate
- GET Bucket analytics
- GET Bucket inventory
- GET Bucket lifecycle
- GET Bucket location
- GET Bucket logging
- GET Bucket metrics
- GET Bucket notification
- GET Bucket policy
- GET Bucket replication
- GET Bucket requestPayment

- GET Bucket website
- List Bucket Analytics Configurations
- List Bucket Inventory Configurations
- List Bucket Metrics Configurations
- PUT Bucket accelerate
- PUT Bucket analytics
- PUT Bucket inventory
- PUT Bucket lifecycle
- PUT Bucket logging
- PUT Bucket metrics
- PUT Bucket notification
- PUT Bucket policy
- PUT Bucket replication
- PUT Bucket requestPayment
- PUT Bucket website
- DELETE Object tagging
- GET Object tagging
- GET Object torrent
- POST Object restore
- PUT Object tagging

An IBM COS System may be configured in either 'Vault' or 'Container' mode. In 'Vault' mode, a system can contain up to 1,000 'Vaults' (commonly referred to as 'buckets' in the S3 API). In 'Container' mode, each of the 1,000 'Container Vaults' themselves can hold any number of buckets, each in turn containing any number of objects. Some operations, such as tagging and versioning, are not available in 'Container' mode.

Common headers

Common Response Headers

The following table describes common response headers.

Header	Note
Content-Length	The length of the request body in bytes.
Connection	Indicates whether the connection is open or closed.
ETag	MD5 hash value of the request.
Date	Timestamp of the request.
Server	Name of the responding server.
X-Clv-Request-Id	Unique identifier generated per request.

Error codes

Error Code	Description	HTTP Status Code
AccessDenied	Access Denied. Authentication or authorization failed for the request.	403 Forbidden

Error Code	Description	HTTP Status Code
AccessDenied	Access Denied - Server-Side Encryption with Customer-Provided Keys is not enabled for this vault.	403 Forbidden
BadDigest	The Content-MD5 you specified did not match what we received	400 Bad Request
BadRequest	Operations on protected vaults are not supported on this interface	400 Bad Request
BucketAlreadyExists	The requested bucket name is not available. The bucket namespace is shared by all users of the system. Please select a different name and try again.	409 Conflict
BucketNotEmpty	The bucket you tried to delete is not empty.	409 Conflict
CredentialsNotSupported	This request does not support credentials.	400 Bad Request
EntityTooSmall	Your proposed upload is smaller than the minimum allowed object size.	400 Bad Request
EntityTooLarge	Your proposed upload exceeds the maximum allowed object size.	400 Bad Request
IllegalVersioningConfigurationException	Indicates that the versioning configuration specified in the request is invalid.	400 Bad Request
IncompleteBody	You did not provide the number of bytes specified by the Content-Length HTTP header.	400 Bad Request
IncorrectNumberOfFilesInPostRequest	POST requires exactly one file upload per request.	400 Bad Request
InlineDataTooLarge	Inline data exceeds the maximum allowed size.	400 Bad Request
InternalError	We encountered an internal error. Please try again.	500 Internal Server Error
InvalidAccessKeyId	The AWS access key Id you provided does not exist in our records.	403 Forbidden
InvalidArgument	Invalid Argument	400 Bad Request
InvalidArgument	Requests specifying Server-Side Encryption with Customer-Provided Keys must be made over a secure connection.	400 Bad Request
InvalidArgument	Requests specifying Server-Side Encryption with Customer-Provided Keys must provide an appropriate secret key.	400 Bad Request
InvalidArgument	Requests specifying Server-Side Encryption with Customer-Provided Keys must provide the client calculated MD5 of the secret key.	400 Bad Request
InvalidArgument	The MD5 hash of the secret key was improperly encoded. The MD5 hash must be Base64 encoded.	400 Bad Request
InvalidArgument	The calculated MD5 hash of the key did not match the hash that was provided.	400 Bad Request
InvalidArgument	Maximum retention is outside the range defined for the service	400 Bad Request
InvalidArgument	Minimum retention is outside the range defined for the service	400 Bad Request

Error Code	Description	HTTP Status Code
InvalidArgument	Default retention is outside the range defined by Minimum Retention and Maximum Retention	400 Bad Request
InvalidArgument	The retention period of an object can only be extended up to the bucket maximum retention period from the time of the request	400 Bad Request
InvalidArgument	The total retention period of an object must be greater than or equal to the minimum retention for the bucket. If permanent retention is enabled for the bucket, the retention period can be set to -2	400 Bad Request
InvalidArgument	New retention expiration date must be greater than current time	400 Bad Request
InvalidArgument	The target bucket does not have permanent retention enabled	400 Bad Request
InvalidArgument	Legal hold ID cannot be null	400 Bad Request
InvalidArgument	The object is under permanent retention, thus the retention period cannot be modified	400 Bad Request
InvalidBucketName	The specified bucket is not valid.	400 Bad Request
InvalidBucketState	The request is not valid with the current state of the bucket.	409 Conflict
InvalidBucketState	Requested status change is not allowed for the bucket. Status requested: {status_specified_in_request}	409 Conflict
InvalidBucketState	Protection cannot be enabled on a bucket with versioning enabled	409 Conflict
InvalidBucketState	Protection cannot be enabled on a bucket with indexing disabled	409 Conflict
InvalidBucketState	Indexing cannot be disabled for a Protected bucket	409 Conflict
InvalidBucketState	Versioning cannot be enabled for a Protected bucket	409 Conflict
InvalidDigest	The Content-MD5 you specified is not valid.	400 Bad Request
InvalidEncryptionAlgorithmError	The Encryption request you specified is not valid. Supported value: AES256.	400 Bad Request
InvalidLocationConstraint	The specified location constraint is not valid. For more information about regions, see How to Select a Region for Your Buckets.	400 Bad Request
InvalidProtectionState	The target bucket is not protected	403 Forbidden
InvalidObjectState	The operation is not valid for the current state of the object.	403 Forbidden
InvalidPart	One or more of the specified parts could not be found. The part might not have been uploaded, or the specified entity tag might not have matched the part's entity tag.	400 Bad Request
InvalidPartOrder	The list of parts was not in ascending order. Parts list must specified in order by part number.	400 Bad Request

Error Code	Description	HTTP Status Code
InvalidRange	The requested range cannot be satisfied.	416 Requested Range Not Satisfiable
InvalidRequest	Please use AWS4-HMAC-SHA256.	400 Bad Request
InvalidRequest	The object was stored using a form of Server-Side Encryption. The correct parameters must be provided to retrieve the object.	400 Bad Request
InvalidRequest	The encryption parameters are not applicable to this object.	400 Bad Request
InvalidRequest	The object was stored using a form of Server-Side Encryption. The correct parameters must be provided to retrieve the object.	400 Bad Request
InvalidRequest	Operations on Protected Vaults not supported on this interface	400 Bad Request
InvalidRequest	Request contains too many retention headers	400 Bad Request
InvalidRequest	*Retention-Period* or *Retention-Expiration-Date* is outside the retention range defined for the bucket	400 Bad Request
InvalidRequest	Request contains duplicate copies of the same header: <*header name*>	400 Bad Request
InvalidRequest	The *Retention-Directive* is set to **COPY** and the Retention period of the source object is outside the range configured for the destination bucket	400 Bad Request
InvalidRequest	The *Retention-Directive* is set to **COPY** and the request includes one or more retention or Legal Hold headers	400 Bad Request
InvalidRequest	Request does not have a request header	400 Bad Request
InvalidRequest	Request signature type is not valid for protected objects or protected vaults.	400 Bad Request
InvalidRequest	Protection Feature is Disabled	400 Bad Request
InvalidRequest	Cannot add and remove legal hold in the same request	400 Bad Request
InvalidRequest	The action "add" or "remove" must be specified	400 Bad Request
InvalidRequest	The Service does not support permanent retention	400 Bad Request
InvalidRequestForLegalReasons	The object is protected	451 Unavailable For Legal Reasons
InvalidRequestForLegalReasons	New retention period is less than the old retention period	451 Unavailable For Legal Reasons
InvalidSecurity	The provided security credentials are not valid.	403 Forbidden
InvalidURI	Couldn't parse the specified URI.	400 Bad Request
KeyTooLong	Your key is too long.	400 Bad Request
LegalHoldTooLong	The legal hold identifier exceeded the maximum legal hold identifier length	400 Bad Request

Error Code	Description	HTTP Status Code
LegalHoldTooShort	The legal hold identifier is below the legal hold identifier minimum length	400 Bad Request
LegalHoldDuplicate	The legal hold could not be added because a legal hold with the identifier already exists	409 Conflict
LegalHoldIDInvalid	The legal hold identifier contains unsupported characters	400 Bad Request
LegalHoldTooMany	The object already has the maximum number of legal holds	400 Bad Request
LegalHoldNotFound	The legal hold identifier being removed was not found on the object	404 Not Found
MalformedACLError	The XML you provided was not well-formed or did not validate against our published schema.	400 Bad Request
MalformedPOSTRequest	The body of your POST request is not well-formed multipart/form-data.	400 Bad Request
MalformedXML	The XML provided was not well-formed or did not validate against the published schema.	400 Bad Request
MaxMessageLengthExceeded	Your request was too big.	400 Bad Request
MaxPostPreDataLengthExceededError	Your POST request fields preceding the upload file were too large.	400 Bad Request
MetadataTooLarge	Your metadata headers exceed the maximum allowed metadata size.	400 Bad Request
MethodNotAllowed	The specified method is not allowed against this resource.	405 Method Not Allowed
MethodNotAllowed	Operation not supported for protected buckets	405 Method Not Allowed
MissingContentLength	You must provide the Content-Length HTTP header.	411 Length Required
MissingDigest	Missing required content hash for this request: Content-MD5 or x-amz-content-sha256	400 Bad Request
MissingRequestBodyError	Request body is empty.	400 Bad Request
NoSuchBucket	The specified bucket does not exist.	404 Not Found
NoSuchKey	The specified key does not exist.	404 Not Found
NoSuchObject	The object does not exist	404 Not Found
NoSuchUpload	The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.	404 Not Found
NoSuchVersion	Indicates that the version ID specified in the request does not match an existing version.	404 Not Found
NotImplemented	A header you provided implies functionality that is not implemented.	501 Not Implemented
NotImplementedForMirrorSetProtection	The API does not support the setting of protection for mirror destination.	501 Not Implemented

Error Code	Description	HTTP Status Code
OperationAborted	A conflicting conditional operation is currently in progress against this resource. Try again.	409 Conflict
PreconditionFailed	At least one of the preconditions you specified did not hold.	412 Precondition
ProtectedBucketNotEmpty	The bucket cannot be deleted because it is a protected bucket and is not empty.	451 Unavailable for Legal Reasons
Redirect	Temporary redirect.	307 Moved Temporarily
RequestIsNotMultiPartContent	Bucket POST must be of the enclosure-type multipart/form-data.	400 Bad Request
RequestTimeout	Your socket connection to the server was not read from or written to within the timeout period.	400 Bad Request
RequestTimeTooSkewed	The difference between the request time and the server's time is too large.	403 Forbidden
SignatureDoesNotMatch	The request signature we calculated does not match the signature you provided. Check your AWS secret access key and signing method. For more information, see REST Authentication and SOAP Authentication for details.	403 Forbidden
ServiceUnavailable	Reduce your request rate.	503 Service Unavailable
SlowDown	Reduce your request rate.	503 Slow Down
TemporaryRedirect	You are being redirected to the bucket while DNS updates.	307 Moved Temporarily
TooManyBuckets	You have attempted to create more buckets than allowed.	400 Bad Request
UnexpectedContent	This request does not support content.	400 Bad Request
UnresolvableGrantByEmailAddress	The email address you provided does not match any account on record.	400 Bad Request
UserKeyMustBeSpecified	The bucket POST must contain the specified field name. If it is specified, check the order of the fields.	400 Bad Request
VaultQuotaExceeded	The capacity used on the target vault has exceeded a hard quota	507 Insufficient Storage

Service operations

List buckets belonging to an account

A `GET` issued to the endpoint root returns a list of buckets owned by the requesting account. This operation does not make use of operation specific headers, query parameters, or payload elements.

Syntax

```
GET https://{endpoint}/
```

Sample request

```
GET 10.132.11.153/ HTTP/1.1
Content-Type: text/plain
Host: 67.228.254.193
X-Amz-Date: 20160822T030815Z
Authorization: {authorization-string}
```

Sample response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Owner>
        <ID>{access-key}</ID>
        <DisplayName>{access-key}</DisplayName>
    </Owner>
    <Buckets>
        <Bucket>
            <Name>bucket-27200-lwx4cfvcue</Name>
            <CreationDate>2016-08-18T14:21:36.593Z</CreationDate>
        </Bucket>
        <Bucket>
            <Name>bucket-27590-drqmydpfdv</Name>
            <CreationDate>2016-08-18T14:22:32.366Z</CreationDate>
        </Bucket>
        <Bucket>
            <Name>bucket-27852-290jtb0n2y</Name>
            <CreationDate>2016-08-18T14:23:03.141Z</CreationDate>
        </Bucket>
        <Bucket>
            <Name>bucket-28731-k0o1gde2rm</Name>
            <CreationDate>2016-08-18T14:25:09.599Z</CreationDate>
        </Bucket>
    </Buckets>
</ListAllMyBucketsResult>
```

List buckets with pagination and extended metadata

This implementation of the **GET account** operation uses the *pagination* subresource to paginate a list of all the buckets owned by the requesting account, and includes the location constraint associated with each bucket.

The behavior in vault mode is the following:

- Pagination and limiting container are not supported, resulting in **IsTruncated** always being false and **MaxKeys** always being 2147483647.
- The request parameters **marker**, **max-keys**, and **prefix** are ignored.
- Location constraint associated with each bucket is an empty string.

In container mode, if the region code is set in the COS Manager, then the **LocationConstraint** element is set as region code. Otherwise, the default behavior occurs.

This operation does not make use of operation specific headers or payload elements

Syntax

```
GET https://{endpoint}/?pagination
```

Optional query parameters

Table 4. Optional query parameters

Request Parameter	Type	Description
marker	String	Specifies the object from where the listing should begin, in UTF–8 binary order. This parameter is ignored if pagination has not been requested
max-keys	Integer	Restricts the number of objects to display in the response. Default and maximum is 1,000. This parameter is ignored if pagination has not be requested
prefix	String	Constrains response to object names beginning with prefix. This parameter is ignored if pagination has not be requested

Operation-specific response parameters

Table 5. Operation-specific response parameters

Response Parameter	Style	Type	Description
LocationConstraint	body	String	Specifies the region where the bucket resides. In vault mode, this parameter is an empty string. In container mode, this value is obtained from the region set for the container vault. If the region is not set, this value is obtained from the provisioning code set for the vault.

Sample request

```
GET 10.132.11.153/?pagination&marker=quota HTTP/1.1
```

Sample response

```
HTTP/1.1 200 OK
Content-Length: 342
Content-Type: application/xml
Accept-Ranges: bytes
X-Clv-Request-Id: b338cafc-1f50-45c3-9655-c20b3fe216bf
Server: Cleversafe/3.9.1-jenkins-smc_feature_brewing_build-80
x-amz-request-id: 1735e66d-9e75-409f-b24e-0a974e015460
Date: Fri, 18 Mar 2016 00:56:10 GMT

<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <Owner>
    <ID>bcaf1ffd86f461ca5fb16fd081034f</ID>
    <DisplayName>bcaf1ffd86f461ca5fb16fd081034f</DisplayName>
  </Owner>
  <Prefix><Prefix/>
  <Marker>quota</Marker>
  <IsTruncated>false</IsTruncated>
  <MaxKeys>1000</MaxKeys>
  <Buckets>
    <Bucket>
      <Name>quotes</Name>
      <LocationConstraint>EU</LocationConstraint>
      <CreationDate>2006-02-03T16:45:09.000Z</CreationDate>
    </Bucket>
    <Bucket>
      <Name>samples</Name>
      <LocationConstraint>EU</LocationConstraint>
```

```
      <CreationDate>2006-02-03T16:41:58.000Z</CreationDate>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>
```

Operations on buckets

Create a new bucket

A PUT issued to the endpoint root followed by a string will create a bucket using that string for a name.

. Bucket names must be unique. Bucket names might be required to be DNS-compliant depending on settings in the COS Manager; if so, names must be 3 - 63 characters long and must be made of lowercase letters, numbers, and dashes. Bucket names must begin and end with a lowercase letter or number. Bucket names that resemble IP addresses are not allowed.

In Vault Mode, the number of buckets you can create is limited by the vault limit of the system.

This operation does not use operation-specific headers or query parameters.

Attention: When in Vault Mode, or when in Container Mode with multiple access pools in the system, wait 90 seconds before you write to a newly created bucket.

Syntax

```
PUT https://{endpoint}/{bucket-name} # path style
PUT https://{bucket-name}.{endpoint} # virtual host style
```

Optional payload elements

Note:
If an XML block specifying a LocationConstraint is provided, it must correspond with a provisioning code set in the COS Manager. If the request payload is left empty, the default provisioning code is used.

```
<CreateBucketConfiguration>
  <LocationConstraint>{provisioning-code}</LocationConstraint>
</CreateBucketConfiguration>
```

Sample request

This is an example of creating a new bucket that is called 'images'.

```
PUT /images HTTP/1.1
Content-Type: text/plain
Host: 67.228.254.193
X-Amz-Date: 20160821T052842Z
Authorization:{authorization-string}
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 24 Aug 2016 17:45:25 GMT
X-Clv-Request-Id: dca204eb-72b5-4e2a-a142-808d2a5c2a87
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.115
X-Clv-S3-Version: 2.5
x-amz-request-id: dca204eb-72b5-4e2a-a142-808d2a5c2a87
Content-Length: 0
```

Retrieve a bucket's headers

A HEAD issued to a bucket resource will return the headers for that bucket. This operation does not make use of operation specific headers, query parameters, or payload elements.

Syntax

```
HEAD https://{endpoint}/{bucket-name} # path style
HEAD https://{bucket-name}.{endpoint} # virtual host style
```

Sample request

This is an example of fetching the headers for the 'images' bucket.

```
HEAD /images HTTP/1.1
Content-Type: text/plain
Host: 67.228.254.193
X-Amz-Date: 20160821T052842Z
Authorization:{authorization-string}
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 24 Aug 2016 17:46:35 GMT
X-Clv-Request-Id: 0c2832e3-3c51-4ea6-96a3-cd8482aca08a
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.115
X-Clv-S3-Version: 2.5
x-amz-request-id: 0c2832e3-3c51-4ea6-96a3-cd8482aca08a
Content-Length: 0
```

Delete a bucket

A `DELETE` issued to an empty bucket resource deletes the bucket. After deleting a bucket the name is reserved by the system for 10 minutes then released for re-use. *Only empty buckets can be deleted.* This operation does not make use of operation specific headers, query parameters, or payload elements.

Syntax

```
DELETE https://{endpoint}/{bucket-name} # path style
DELETE https://{bucket-name}.{endpoint} # virtual host style
```

Sample request

```
DELETE /images HTTP/1.1
Host: 67.228.254.193
x-amz-date: 20160822T064812Z
Authorization: {authorization-string}
```

The server responds with `204 No Content`.

If a non-empty bucket is requested for deletion, the server responds with `409 Conflict`.

Sample request

```
DELETE /example HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20160825T174049Z
Host: 67.228.254.193
```

Sample response

```
<Error>
  <Code>BucketNotEmpty</Code>
  <Message>The bucket you tried to delete is not empty.</Message>
  <Resource>/example-bucket/</Resource>
  <RequestId>9d2bbc00-2827-4210-b40a-8107863f4386</RequestId>
  <httpStatusCode>409</httpStatusCode>
</Error>
```

List objects in a given bucket

A `GET` request addressed to a bucket returns a list of objects, limited to 1,000 at a time and returned in non-lexographical order. The `StorageClass` value that is returned in the response is the storage class set in the COS Manager in container mode, or the default value if no storage class is set or in vault mode.

Syntax

```
GET https://{endpoint}/{bucket-name} # path style
GET https://{bucket-name}.{endpoint} # virtual host style
```

Optional request header

Table 6. Request header

Name	Description	Required
`Mirror-Destination`	This header is applicable for listing of buckets in a protected mirror. The **`Mirror-Destination`** header specifies from which vault of the mirror to read. By default, if no explicit vault is specified, then the listing request will attempt to read from both vaults and provide a listing response that combines the list of objects on each component vault to the mirror excluding duplicates (object resides on both vaults in the mirror). If the **`Mirror-Destination`** header is specified and includes a valid Vault Identifier, the data returned will be from the Vault with the ID that matches what was provided in the header. The **`Mirror-Destination`** header is applicable only to protected mirrors, and the header is ignored otherwise. A failure to read from the specified vault will result in an error back to the HTTP client. **Type** String **Default** None **Constraints** {Valid Vault Identifier}	No

Optional query parameters

Name	Type	Description
`prefix`	string	Constrains response to object names beginning with `prefix`.
`delimiter`	string	Groups objects between the `prefix` and the `delimiter`.
`encoding-type`	string	If unicode characters that are not supported by XML are used in an object name, this parameter can be set to `url` to properly encode the response.
`max-keys`	string	Restricts the number of objects to display in the response. Default and maximum is 1,000.
`marker`	string	Specifies the object from where the listing should begin, in UTF–8 binary order.

Sample request

This request lists the objects inside the “example” bucket.

```
GET /example-bucket HTTP/1.1
Content-Type: text/plain
Host: 67.228.254.193
X-Amz-Date: 20160822T225156Z
Authorization: {authorization-string}
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 24 Aug 2016 17:36:24 GMT
X-Clv-Request-Id: 9f39ff2e-55d1-461b-a6f1-2d0b75138861
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.115
X-Clv-S3-Version: 2.5
x-amz-request-id: 9f39ff2e-55d1-461b-a6f1-2d0b75138861
Content-Type: application/xml
Content-Length: 909
```

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>example-bucket</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <Delimiter/>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>object-1</Key>
    <LastModified>2016-08-25T17:38:38.549Z</LastModified>
    <ETag>"0cbc6611f5540bd0809a388dc95a615b"</ETag>
    <Size>4</Size>
    <Owner>
      <ID>{access-key}</ID>
      <DisplayName>{username}</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>soldier-bee</Key>
    <LastModified>2016-08-25T17:49:06.006Z</LastModified>
    <ETag>"37d4c94839ee181a2224d6242176c4b5"</ETag>
    <Size>11</Size>
    <Owner>
      <ID>{access-key}</ID>
      <DisplayName>{username}</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>worker-bee</Key>
    <LastModified>2016-08-25T17:46:53.288Z</LastModified>
    <ETag>"d34d8aada2996fc42e6948b926513907"</ETag>
    <Size>467</Size>
    <Owner>
      <ID>{access-key}</ID>
      <DisplayName>{username}</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
</ListBucketResult>
```

List objects in a given bucket (v2)

A GET request addressed to a bucket with the query parameter **list-type=2** returns a list of objects, limited to 1,000 at a time, returned in non-lexigraphical order and allows for additional flexibility in

listing. The `StorageClass` value that is returned in the response is a default value as storage class operations are not implemented in the system. This operation does not make use of operation specific headers or payload elements.

Syntax

```
GET https://{endpoint}/{bucket-name}?list-type=2 # path style
GET https://{bucket-name}.{endpoint}?list-type=2 # virtual host style
```

Optional request header

Table 7. Request header

Name	Description	Required
`Mirror-Destination`	This header is applicable for listing of buckets in a protected mirror. The **`Mirror-Destination`** header specifies from which vault of the mirror to read. By default, if no explicit vault is specified, then the listing request will attempt to read from both vaults and provide a listing response that combines the list of objects on each component vault to the mirror excluding duplicates (object resides on both vaults in the mirror). If the **`Mirror-Destination`** header is specified and includes a valid Vault Identifier, the data returned will be from the Vault with the ID that matches what was provided in the header. The **`Mirror-Destination`** header is applicable only to protected mirrors, and the header is ignored otherwise. A failure to read from the specified vault will result in an error back to the HTTP client. **Type** String **Default** None **Constraints** {Valid Vault Identifier}	No

Optional query parameters

Name	Type	Description
`prefix`	string	Constrains response to object names beginning with `prefix`.
`delimiter`	string	Groups objects between the `prefix` and the `delimiter`.
`max-keys`	string	Restricts the number of objects to display in the response. Default and maximum is 1,000.
`continuation-token`	string	If the total number of objects exceeds the value of `max-keys`, a `NextContinuationToken` is returned. Passing the value of the `NextContinuationToken` in a subsequent listing request continues listing from that point.
`fetch-owner`	string	Owner information is not passed by default. If owner information is needed, this value must be set to `true`.

Name	Type	Description
\start-after	string	Sets the starting point o list objects in lexigraphical order. Ignored if continuation-token is also present.

Sample request

This request lists the objects inside the “example” bucket.

```
GET /example-bucket?list-type=2 HTTP/1.1
Content-Type: text/plain
Host: 67.228.254.193
X-Amz-Date: 20160822T225156Z
Authorization: {authorization-string}
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 24 Aug 2016 17:36:24 GMT
X-Clv-Request-Id: 9f39ff2e-55d1-461b-a6f1-2d0b75138861
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.115
X-Clv-S3-Version: 2.5
x-amz-request-id: 9f39ff2e-55d1-461b-a6f1-2d0b75138861
Content-Type: application/xml
Content-Length: 909
```

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>apiary</Name>
  <Prefix/>

  <MaxKeys>1000</MaxKeys>
  <KeyCount>205</KeyCount>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>drone-bee</Key>
    <LastModified>2016-08-25T17:38:38.549Z</LastModified>
    <ETag>"0cbc6611f5540bd0809a388dc95a615b"</ETag>
    <Size>4</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>soldier-bee</Key>
    <LastModified>2016-08-25T17:49:06.006Z</LastModified>
    <ETag>"37d4c94839ee181a2224d6242176c4b5"</ETag>
    <Size>11</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>worker-bee</Key>
    <LastModified>2016-08-25T17:46:53.288Z</LastModified>
    <ETag>"d34d8aada2996fc42e6948b926513907"</ETag>
    <Size>467</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    ...
  </Contents>
  ...
</ListBucketResult>

<Name>bucket</Name>
    <Prefix/>
    <KeyCount>205</KeyCount>
    <MaxKeys>1000</MaxKeys>
```

```
    <IsTruncated>false</IsTruncated>
    <Contents>
        <Key>my-image.jpg</Key>
        <LastModified>2009-10-12T17:50:30.000Z</LastModified>
        <ETag>"fba9dede5f27731c9771645a39863325"</ETag>
        <Size>434234</Size>
        <StorageClass>STANDARD</StorageClass>
    </Contents>
    <Contents>
       ...
    </Contents>
    ...
</ListBucketResult>
```

Create an access control list for a bucket

A `PUT` issued to a bucket with the necessary query parameter creates or replaces an access control list (ACL) for that bucket. Access control lists allow for granting different sets of permissions to different storage accounts using the account's ID, or by using a pre-made ACL.

ACLs can use pre-made permissions sets (or 'canned ACLs') or be customized in the body of the request. Pre-made ACLs are specified using the `x-amz-acl` header and custom ACLs are specified using XML in the request payload. Only one method (header or payload) can be used in a single request.

This operation does not make use of additional operation specific query parameters.

ACL grantees can be specified using any of the following methods:

Method	Description	Example
Canonical ID	User account UUID	`43a89ab8-a5e9-44bf-9671-d23a8729b2e0`
Email Address	Username of user account as set in COS Manager	`user1`
URI	Used for pre-defined groups. COS supports the All Users Group for bucket ACLs and the All Users Group and Authenticated Users URIs for Object ACLs. All other predefined groups are unsupported.	`http://acs.amazonaws.com/groups/global/AllUsers` or `http://acs.amazonaws.com/groups/global/AuthenticatedUsers`

The assigned permissions behave as follows:

Permission	When granted on a bucket	When granted on an object
READ	Allows grantee to list and read all objects in bucket	Allows grantee to read object data and metadata
WRITE	Allows grantee to create, overwrite and delete any object in bucket. Cannot be granted independently from READ permission.	N/A
READ_ACP	This permission does not exist for buckets; default setting is FULL_CONTROL	Allows grantee to read object ACL
WRITE_ACP	Default setting is FULL_CONTROL	Allows grantee to write ACL for applicable object

Permission	When granted on a bucket	When granted on an object
FULL_CONTROL	Allows grantee READ, WRITE, READ_ACP and WRITE_ACP permissions on bucket	Allows grantee READ, READ_ACP and WRITE_ACP permissions on object

Note: The `READ_ACP`, `WRITE_ACP`, and `FULL_CONTROL` permissions are implied by the bucket "own" permission. When any of these permissions are assigned to a grantee in a bucket ACL, that grantee will be granted the bucket "own" permission.

The following canned ACLs are supported by IBM COS. Values not listed below are not supported.

Canned ACL	Applies to	Notes
private	Bucket and object	When set on a bucket, the requestor is interpreted as the bucket owner.
public-read	Bucket and object	When set on a bucket, the requestor is interpreted as the bucket owner.
public-read-write	Bucket and object	When set on a bucket, the requestor is interpreted as the bucket owner.
authenticated-read	Bucket and object	Supported when set on an object only. Not supported as a bucket ACL.

Syntax

```
PUT https://{endpoint}/{bucket-name}?acl= # path style
PUT https://{bucket-name}.{endpoint}?acl= # virtual host style
```

Sample request of a basic pre-made ACL

This is an example of specifying a pre-made ACL to allow for `public-read` access to the "example" bucket. This allows any storage account to view the bucket's contents and ACL details.

```
PUT /example?acl= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161011T190354Z
x-amz-acl: public-read
Host: 67.228.254.193
```

Sample response

```
HTTP/1.1 200 OK
Date: Tue, 4 Oct 2016 19:03:55 GMT
X-Clv-Request-Id: 73d3cd4a-ff1d-4ac9-b9bb-43529b11356a
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.129
X-Clv-S3-Version: 2.5
x-amz-request-id: 73d3cd4a-ff1d-4ac9-b9bb-43529b11356a
Content-Length: 0
```

Sample request Custom ACL

This is an example of specifying a custom ACL to allow for another user using their username to view the ACL for the "example" bucket, but not to list objects stored inside the bucket. A third account is given full access to the same bucket as another element of the same ACL. All authenticated users of the system can list objects in the bucket.

```
PUT /example?acl= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161011T190354Z
Host: 67.228.254.193
```

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>{owner-storage-account-uuid}</ID>
    <DisplayName>OwnerDisplayName</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="AmazonCustomerByEmail">
        <ID>{username}</ID>
        <DisplayName>Grantee1DisplayName</DisplayName>
      </Grantee>
      <Permission>READ_ACP</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>{second-grantee-storage-account-uuid}</ID>
        <DisplayName>Grantee2DisplayName</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <ID>http://acs.amazonaws.com/groups/global/AuthenticatedUsers</ID>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Sample response

```
HTTP/1.1 200 OK
Date: Tue, 4 Oct 2016 19:03:55 GMT
X-Clv-Request-Id: 73d3cd4a-ff1d-4ac9-b9bb-43529b11356a
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.129
X-Clv-S3-Version: 2.5
x-amz-request-id: 73d3cd4a-ff1d-4ac9-b9bb-43529b11356a
```

Retrieve the access control list for a bucket

A `GET` issued to a bucket with the proper parameters retrieves the ACL for a bucket. This operation does not make use of operation specific headers, additional query parameters, or payload elements.

Syntax

```
GET https://{endpoint}/{bucket-name}?acl= # path style
GET https://{bucket-name}.{endpoint}?acl= # virtual host style
```

Optional request header

Table 8. Request header

Name	Description	Required
Mirror-Destination	This header is applicable for listing of buckets in a protected mirror. The **Mirror-Destination** header specifies from which vault of the mirror to read. By default, if no explicit vault is specified, then the listing request will attempt to read from both vaults and provide a listing response that combines the list of objects on each component vault to the mirror excluding duplicates (object resides on both vaults in the mirror). If the **Mirror-Destination** header is specified and includes a valid Vault Identifier, the data returned will be from the Vault with the ID that matches what was provided in the header. The **Mirror-Destination** header is applicable only to protected mirrors, and the header is ignored otherwise. A failure to read from the specified vault will result in an error back to the HTTP client. **Type** String **Default** None **Constraints** {Valid Vault Identifier}	No

Sample request

This is an example of retrieving a bucket ACL.

```
GET /example?acl= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161011T190354Z
Host: 67.228.254.193
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 5 Oct 2016 14:14:34 GMT
X-Clv-Request-Id: eb57e60e-d84e-4237-b18a-be9c2bb0deb8
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.129
X-Clv-S3-Version: 2.5
x-amz-request-id: eb57e60e-d84e-4237-b18a-be9c2bb0deb8
Content-Type: application/xml
Content-Length: 550
```

```
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>{owner-storage-account-uuid}</ID>
    <DisplayName>OwnerDisplayName</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>{owner-storage-account-uuid}</ID>
        <DisplayName>OwnerDisplayName</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Create a cross-origin resource sharing configuration for a bucket

A PUT issued to a bucket with the proper parameters creates or replaces a cross-origin resource sharing (CORS) configuration for a bucket. Note that in addition to a SHA256 hash of the body, a Content-MD5 header is required as well. This operation does not make use of operation specific headers or additional query parameters.

Syntax

```
PUT https://{endpoint}/{bucket-name}?cors= # path style
PUT https://{bucket-name}.{endpoint}?cors= # virtual host style
```

Optional payload elements

In the XML block defining the key CORS elements (AllowedOrigin and AllowedMethod) there are two optional elements that can be optionally specified.

Element	Description
MaxAgeSeconds	Time in seconds that the browser will cache the response to the pre-flight OPTIONS request for the specified resource.
ExposeHeader	Defines specific headers that will be exposed to external applications.

Sample request

This is an example of adding a CORS configuration that allows requests from www.ibm.com to issue GET, PUT, and POST requests to the bucket.

```
GET /example-bucket?cors= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161011T190354Z
x-amz-content-sha256: 2938f51643d63c864fdbea618fe71b13579570a86f39da2837c922bae68d72df
Content-MD5: GQmpTNpruOyK6YrxHnpj7g==
Content-Type: text/plain
Host: 67.228.254.193
Content-Length: 237
```

```
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.ibm.com</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
  </CORSRule>
</CORSConfiguration>
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 5 Oct 2016 15:39:38 GMT
X-Clv-Request-Id: 7afca6d8-e209-4519-8f2c-1af3f1540b42
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.129
X-Clv-S3-Version: 2.5
x-amz-request-id: 7afca6d8-e209-4519-8f2c-1af3f1540b42
Content-Length: 0
```

List any cross-origin resource sharing configuration for a bucket

A GET issued to a bucket with the proper parameters retrieves information about cross-origin resource sharing (CORS) configuration for a bucket. This operation does not make use of operation specific headers, additional query parameters, or payload elements.

Syntax

```
GET https://{endpoint}/{bucket-name}?cors= # path style
GET https://{bucket-name}.{endpoint}?cors= # virtual host style
```

Optional request header

Table 9. Request header

Name	Description	Required
Mirror-Destination	This header is applicable for listing of buckets in a protected mirror. The **Mirror-Destination** header specifies from which vault of the mirror to read. By default, if no explicit vault is specified, then the listing request will attempt to read from both vaults and provide a listing response that combines the list of objects on each component vault to the mirror excluding duplicates (object resides on both vaults in the mirror). If the **Mirror-Destination** header is specified and includes a valid Vault Identifier, the data returned will be from the Vault with the ID that matches what was provided in the header. The **Mirror-Destination** header is applicable only to protected mirrors, and the header is ignored otherwise. A failure to read from the specified vault will result in an error back to the HTTP client. **Type** String **Default** None **Constraints** {Valid Vault Identifier}	No

Sample request

This is an example of listing a CORS configuration on the “example” bucket.

```
GET /example-bucket?cors= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161011T190354Z
Host: 67.228.254.193
```

Sample response No CORS configuration set

```
HTTP/1.1 200 OK
Date: Wed, 5 Oct 2016 15:20:30 GMT
X-Clv-Request-Id: 0b69bce1-8420-4f93-a04a-35d7542799e6
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.129
X-Clv-S3-Version: 2.5
x-amz-request-id: 0b69bce1-8420-4f93-a04a-35d7542799e6
Content-Type: application/xml
Content-Length: 123

<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/"/>
```

Delete any cross-origin resource sharing configuration for a bucket

A DELETE issued to a bucket with the proper parameters removes cross-origin resource sharing (CORS) configuration for a bucket. This operation does not make use of operation specific headers, additional query parameters, or payload elements.

Syntax

```
DELETE https://{endpoint}/{bucket-name}?cors= # path style
DELETE https://{bucket-name}.{endpoint}?cors= # virtual host style
```

Sample request

This is an example of deleting a CORS configuration for a bucket.

```
GET /example-bucket?cors= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161011T190354Z
Host: 67.228.254.193
```

The server responds with `204 No Content`.

Set up a bucket for versioning

A `PUT` issued to a bucket with the proper parameters enables or suspends the versioning of objects stored in the bucket. Objects are limited to 1,000 versions. Each object uploaded will automatically be assigned a unique version ID, which is shown in response headers as `x-amz-version-id`. In the event of multiple simultaneous object writes, all objects are stored as seperate versions. Any `GET` or `HEAD` request against an object resource can then be modified with the `?versionId=` query parameter to specify a version. This operation does not make use of operation specific headers, additional query parameters, or payload elements.

Syntax

```
PUT https://{endpoint}/{bucket-name}?versioning= # path style
PUT https://{bucket-name}.{endpoint}?versioning= # virtual host style
```

Required payload

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>{Enabled | Disabled}</Status>
</VersioningConfiguration>
```

Sample request

This is an example of enabling versioning on a bucket.

```
PUT /example-bucket?versioning= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161011T190354Z
Host: 67.228.254.193
x-amz-content-sha256: f4edcce17375eba17a48c19e272604dfcfb6723c04eb18bc4aa4ce8567c196a2
Content-Type: text/plain; charset=utf-8
Content-Length: 124

<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 11 Oct 2016 15:22:27 GMT
X-Clv-Request-Id: 9fa96daa-9f37-42ee-ab79-0bcda049c671
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.129
X-Clv-S3-Version: 2.5
x-amz-request-id: 9fa96daa-9f37-42ee-ab79-0bcda049c671
```

Check if versioning is enabled on a bucket

A `GET` request addressed to a bucket with the proper parameter returns an indication of whether the bucket has versioning enabled, disabled, or never enabled. This operation does not make use of operation specific headers, additional query parameters, or payload elements.

The StorageClass value that is returned in the response is the storage class set in the COS Manager in container mode, or it is the default value if no storage class is set or if the COS Manager is in vault mode.

Syntax

```
GET https://{endpoint}/{bucket-name}?versioning= # path style
GET https://{bucket-name}.{endpoint}?versioning= # virtual host style
```

Sample request

This request lists the objects inside the "example" bucket.

```
GET /example-bucket?versioning= HTTP/1.1
Host: 67.228.254.193
X-Amz-Date: 20160822T225156Z
Authorization: {authorization-string}
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 22 Aug 2016 17:36:24 GMT
X-Clv-Request-Id: 9f39ff2e-55d1-461b-a6f1-2d0b75138861
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.115
X-Clv-S3-Version: 2.5
Content-Type: application/xml
Content-Length: 127
```

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

List versioned objects in a bucket

A GET request addressed to a bucket returns a list of versioned objects, limited to 1,000 at a time and returned in non-lexographical order. The StorageClass value that is returned in the response is a default value as storage class operations are not implemented in COS. This operation does not make use of operation specific headers or payload elements.

Syntax

```
GET https://{endpoint}/{bucket-name}?versions= # path style
GET https://{bucket-name}.{endpoint}?versions= # virtual host style
```

Optional query parameters

Name	Type	Description
prefix	string	Constrains response to object names beginning with prefix.
delimiter	string	Groups objects between the prefix and the delimiter.
encoding-type	string	If unicode characters that are not supported by XML are used in an object name, this parameter can be set to url to properly encode the response.
max-keys	string	Restricts the number of objects to display in the response. Default and maximum is 1,000.

Name	Type	Description
key-marker	string	Specifies the object from where the listing should begin, in UTF–8 binary order.
version-id-marker	string	Specifies the object from where the listing should begin, in UTF–8 binary order.

Sample request

This request lists the objects inside the "example" bucket.

```
GET /example-bucket?versions= HTTP/1.1
Content-Type: text/plain
Host: 67.228.254.193
X-Amz-Date: 20160822T225156Z
Authorization: {authorization-string}
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 24 Aug 2016 17:36:24 GMT
X-Clv-Request-Id: 9f39ff2e-55d1-461b-a6f1-2d0b75138861
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.115
X-Clv-S3-Version: 2.5
x-amz-request-id: 9f39ff2e-55d1-461b-a6f1-2d0b75138861
Content-Type: application/xml
Content-Length: 909
```

```
<ListVersionsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
    <Name>example-bucket</Name>
    </Prefix>
    <KeyMarker/>
    <VersionIdMarker/>
    <MaxKeys>5</MaxKeys>
    <IsTruncated>false</IsTruncated>
    <Version>
        <Key>my-file-1</Key>
        <VersionId>3/L4kqtJl40Nr8X8gdRQBpUMLUo</VersionId>
        <IsLatest>true</IsLatest>
         <LastModified>2016-10-12T17:50:30.000Z</LastModified>
        <ETag>"fba9dede5f27731c9771645a39863328"</ETag>
        <Size>4623462</Size>
        <StorageClass>STANDARD</StorageClass>
        <Owner>
            <ID>{access-key}</ID>
            <DisplayName>{username}</DisplayName>
        </Owner>
    </Version>
    <DeleteMarker>
        <Key>my-file-2</Key>
        <VersionId>03jpff543dhffds434rfdsFDN943fdsFkdmqnh892</VersionId>
        <IsLatest>true</IsLatest>
        <LastModified>2016-11-12T17:50:30.000Z</LastModified>
        <Owner>
            <ID>{access-key}</ID>
            <DisplayName>{username}</DisplayName>
        </Owner>
    </DeleteMarker>
    <Version>
        <Key>my-file-2</Key>
        <VersionId>QUpfdndhfd8438MNFDN93jdnJFkdmqnh893</VersionId>
        <IsLatest>false</IsLatest>
        <LastModified>2016-10-10T17:50:30.000Z</LastModified>
```

```
        <ETag>"0cbc6611f5540bd0809a388dc95a615b"</ETag>
        <Size>32452346</Size>
        <StorageClass>STANDARD</StorageClass>
        <Owner>
            <ID>{access-key}</ID>
            <DisplayName>{username}</DisplayName>
        </Owner>
    </Version>
    <DeleteMarker>
        <Key>my-file-3</Key>
        <VersionId>03jpff543dhffds434rfdsFDN943fdsFkdmqnh892</VersionId>
        <IsLatest>true</IsLatest>
        <LastModified>2009-10-15T17:50:30.000Z</LastModified>
        <Owner>
            <ID>{access-key}</ID>
            <DisplayName>{username}</DisplayName>
        </Owner>
    </DeleteMarker>
    <Version>
        <Key>my-file-3</Key>
        <VersionId>UIORUnfndfhnw89493jJFJ</VersionId>
        <IsLatest>false</IsLatest>
        <LastModified>2016-10-11T12:50:30.000Z</LastModified>
        <ETag>"d34d8aada2996fc42e6948b926513907"</ETag>
        <Size>45</Size>
        <StorageClass>STANDARD</StorageClass>
        <Owner>
            <ID>{access-key}</ID>
            <DisplayName>{username}</DisplayName>
        </Owner>
     </Version>
</ListVersionsResult>
```

Add protection to a bucket

This implementation of the **PUT** operation uses the *protection* subresource to set the retention parameters for an existing bucket. This operation allows you to set or change the minimum, default, and maximum retention period. It is not supported for protected mirrors.

First, you must create an empty bucket with indexing on and, if supported, versioning off. Then, you can enable the protection state **Retention** on the bucket. If the bucket is not empty, indexing is off, or versioning is on, then a 400 error is returned. The following diagram shows the legal state transition.

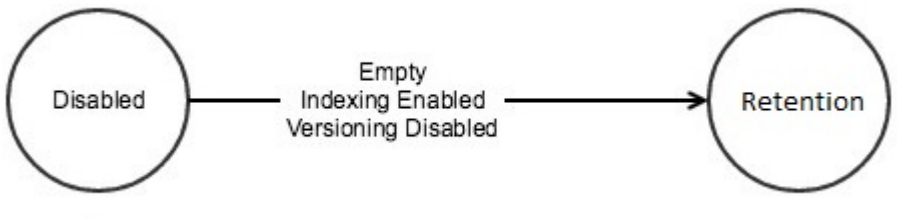

Figure 1. PUT bucket protection - state transition

Note: To support backwards compatibility with ClevOS 3.12.0, this request accepts the value `Compliance` for the `Status` parameter in addition to `Retention`. The protection state `Retention` and `Compliance` are used interchangeably.

Objects written to a protected bucket cannot be deleted until the protection period has expired and all legal holds on the object are removed. The default retention value of a protected bucket is given to an object written to that bucket if the object write request did not specify a retention period. Objects in

protected buckets that are no longer under retention (retention period has expired and the object does not have any legal holds), when overwritten, will again come under retention. The new retention period can be provided as part of the object overwrite request or the default retention period of the bucket will be given to the object. For more information on protected objects, see "Upload an object to a protected bucket" on page 45.

The supported values for the retention period settings **MinimumRetention**, **DefaultRetention**, and **MaximumRetention** are bound by the system minimum and system maximum retention values. **DefaultRetention** is bounded by the **MinimumRetention** and **MaximumRetention** included in the request. In Vault Mode, a `PUT Bucket?protection` request that does not contain all the three retention periods will be accepted and the missing parameters will be replaced with the System Level parameters. In Container Mode, all the retention parameters must be included with each `PUT Bucket?protection` request. Otherwise, an error will be returned to the user.. If you want to change one of the three values, but maintain non-default values for the other two, then all three must be specified. In Container Mode, all required parameters must be specified when creating or modifying a bucket policy.

The system has maximum and minimum values for the bucket's minimum and maximum retention period. If the request exceeds the system level default settings, an error is returned. A system administrator can adjust the system limits as well as the system defaults. For more information, see Configuring vault protection

The storage account user making a **POST /bucket?protection** request must have **FULL_CONTROL** permissions for this object. For more information, see "Create an ACL for an object" on page 64.

AWS Signature V4 is required for this operation. It is recommended that protection headers are included in the signature and that the **x-amz-content-sha256** header is set to `STREAMING-AWS4-HMAC-SHA256-PAYLOAD` (chunked upload) or the payload checksum (single chunk upload with signed payload). It is recommended that users do not use **UNSIGNED-PAYLOAD** in the V4 signature calculation. If a **x-amz-content-sha256** header is not included in the V4 signature, then a `Content-MD5` header is required for this operation.

This operation does not make use of additional query parameters.

Permanent retention

In addition to setting minimum, maximum and default retention periods for a bucket, you can allow permanent retention of objects in a protected bucket by setting the optional `EnablePermanentRetention` parameter to `true`. Permanent retention, when enabled for a bucket, allows objects to be written for permanent storage into that bucket. Once written, such objects cannot be deleted.

Note: Permanent Retention can be enabled on a bucket only if the system is enabled for Permanent retention. For more information, see Configuring vault protection.

Requests

Syntax

```
PUT https://{endpoint}/{bucket-name}?protection= # path style
PUT https://{bucket-name}.{endpoint}?protection= # virtual host style
```

Payload elements

```
<ProtectionConfiguration>
  <Status>Retention</Status>
  <MinimumRetention>
    <Days>100</Days>
  </MinimumRetention>
  <MaximumRetention>
    <Days>10000</Days>
```

```
  </MaximumRetention>
  <DefaultRetention>
    <Days>2555</Days>
  </DefaultRetention>
  <EnablePermanentRetention>true</EnablePermanentRetention>
</ProtectionConfiguration>
```

Table 10. Payload elements

Name	Description	Required
ProtectionConfiguration	Container for setting retention state. **Type** Container **Children** **Status**, **MinimumRetention**, **DefaultRetention**, **MaximumRetention**, **EnablePermanentRetention** **Ancestor** None	Yes
Status	Sets retention state on the bucket. If this field is not specified the state is not changed. **Type** Container **Valid Values** **Retention** **Note:** To support backwards compatibility with ClevOS 3.12.0, this request accepts the value `Compliance` for the `Status` parameter in addition to `Retention`. The protection state `Retention` and `Compliance` are used interchangeably. **Ancestor** **ProtectionConfiguration**	Yes
MinimumRetention	Minimum retention period for an object. If a write request of an object specifies a shorter retention period, the object write request fails. **Default value (days) if not specified** Vault Mode: System Minimum Duration Container Mode: None, the value must be specified. **Minimum value (days)** System Minimum Duration **Maximum value (days)** System Maximum Duration **Type** Container **Children** **Days** **Ancestor** **ProtectionConfiguration** **Constraints** **MinimumRetention** must be greater than or equal to the System Minimum Duration configured by the service administrator.	Yes

Table 10. Payload elements (continued)

Name	Description	Required
DefaultRetention	Default retention period for an object if an object write request does not specify a retention period. If a write request of an object does not specify a retention period, then the bucket default retention period is assigned to the object. A value of -2 indicates that objects which do not specify a retention period during object creation are permanently retained. **Default value (days) if not specified** Vault Mode: System Default Retention Duration Container Mode: None, the value must be specified. **Minimum value (days)** System Minimum Duration **Maximum value (days)** System Maximum Duration **Type** Container **Children** **Days** **Ancestor** **ProtectionConfiguration** **Constraints** **DefaultRetention** must be greater than or equal to **MinimumRetention** and less than or equal to **MaximumRetention**	Yes
MaximumRetention	Maximum retention period for an object. If an object write request specifies a longer retention period than the maximum retention period, then the request fails. **Default value (days) if not specified** Vault Mode: System Maximum Duration Container Mode: None, the value must be specified. **Minimum value (days)** System Minimum Duration **Maximum value (days)** System Maximum Duration **Type** Container **Children** **Days** **Ancestor** **ProtectionConfiguration** **Constraints** **MaximumRetention** must be less than or equal to the System Maximum Duration configured by the service administrator	No

Table 10. Payload elements (continued)

Name	Description	Required
Days	Specifies a retention period in days. **Type** Container **Valid values** Non-negative integer or -2 **Children** **Days** **Ancestor** **MinimumRetention**, **DefaultRetention**, **MaximumRetention** **Constraint** Must be specified if one of its ancestors is specified.	Yes (if parent is specified)
EnablePermanentRetention	Specifies whether this bucket can support permanent retention of objects. This field can only be set to `true` if permanent retention is enabled for the system. Once set to `true`, this field must be included in subsequent `PUT bucket?protection` requests and must be set to `true`, or the request will be rejected. **Default value if not specified** False **Type** Boolean **Ancestor** **ProtectionConfiguration** **Constraints** This field can only be set to `true` if the service administrator has enabled permanent retention for the service.	No

Examples

Sample request

This is an example of modifying an existing, empty bucket called "images" to have the Retention protection configuration and the ability to retain objects permanently enabled.

```
PUT /images?protection HTTP/1.1
Host: myBucket.mydsNet.corp.com
Date: Wed, 8 Feb 2017 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: <Length>

<ProtectionConfiguration>
  <Status>Retention</Status>
  <MinimumRetention>
    <Days>100</Days>
  </MinimumRetention>
  <MaximumRetention>
    <Days>10000</Days>
  </MaximumRetention>
  <DefaultRetention>
    <Days>2555</Days>
  </DefaultRetention>
  <EnablePermanentRetention>true</EnablePermanentRetention>
</ProtectionConfiguration>
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 8 Feb 2017 17:51:00 GMT
Connection: close
```

Add tags to a bucket

A PUT issued to a bucket with the proper parameters adds tags to a bucket. This operation does not make use of operation specific headers or additional query parameters.

Syntax

```
PUT https://{endpoint}/{bucket-name}?tagging= # path style
PUT https://{bucket-name}.{endpoint}?tagging= # virtual host style
```

Required payload

```
<Tagging>
  <TagSet>
     <Tag>
       <Key>{tag-key}</Key>
       <Value>{tag-value}</Value>
     </Tag>
  </TagSet>
</Tagging>
```

Sample request

This is an example of adding tags on a bucket.

```
PUT /example-bucket?tagging= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161011T190354Z
Host: 67.228.254.193
x-amz-content-sha256: f4edcce17375eba17a48c19e272604dfcfb6723c04eb18bc4aa4ce8567c196a2
Content-Type: text/plain; charset=utf-8
Content-Length: 124
```

```
<Tagging>
  <TagSet>
     <Tag>
       <Key>Team</Key>
       <Value>Marketing</Value>
     </Tag>
     <Tag>
       <Key>Color</Key>
       <Value>Blue</Value>
     </Tag>
  </TagSet>
</Tagging>
```

The server responds with `204 No Content`.

Delete tags from a bucket

A DELETE issued to a bucket with the proper parameters removes tags to a bucket. This operation does not make use of operation specific headers or additional query parameters.

Syntax

```
DELETE https://{endpoint}/{bucket-name}?tagging= # path style
DELETE https://{bucket-name}.{endpoint}?tagging= # virtual host style
```

Sample request

This is an example of deleting tags from a bucket.

```
DELETE /example-bucket?tagging= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161011T190354Z
Host: 67.228.254.193
```

The server responds with `204 No Content`.

List canceled/incomplete multipart uploads for a bucket

A `GET` issued to a bucket with the proper parameters retrieves information about any canceled or incomplete multipart uploads for a bucket.

Syntax

```
GET https://{endpoint}/{bucket-name}?uploads= # path style
GET https://{bucket-name}.{endpoint}?uploads= # virtual host style
```

Optional request header

Table 11. Request header

Name	Description	Required
`Mirror-Destination`	This header is applicable for listing of buckets in a protected mirror. The **`Mirror-Destination`** header specifies from which vault of the mirror to read. By default, if no explicit vault is specified, then the listing request will attempt to read from both vaults and provide a listing response that combines the list of objects on each component vault to the mirror excluding duplicates (object resides on both vaults in the mirror). If the **`Mirror-Destination`** header is specified and includes a valid Vault Identifier, the data returned will be from the Vault with the ID that matches what was provided in the header. The **`Mirror-Destination`** header is applicable only to protected mirrors, and the header is ignored otherwise. A failure to read from the specified vault will result in an error back to the HTTP client. **Type** String **Default** None **Constraints** {Valid Vault Identifier}	No

Optional additional query parameters

Name	Type	Description
`prefix`	string	Constrains response to object names beginning with {`prefix`}.
`delimiter`	string	Groups objects between the `prefix` and the `delimiter`.
`encoding-type`	string	If unicode characters that are not supported by XML are used in an object name, this parameter can be set to `url` to properly encode the response.
`max-uploads`	integer	Restricts the number of objects to display in the response. Default and maximum is 1,000.

Name	Type	Description
key-marker	string	Specifies from where the listing should begin.
upload-id-marker	string	Ignored if key-marker is not specified, otherwise sets a point at which to begin listing parts above upload-id-marker.

Sample request

This is an example of retrieving all current canceled and incomplete multipart uploads.

```
GET /example-bucket?uploads= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161011T190354Z
Host: 67.228.254.193
```

Sample response (no multipart uploads in progress)

```
HTTP/1.1 200 OK
Date: Wed, 5 Oct 2016 15:22:27 GMT
X-Clv-Request-Id: 9fa96daa-9f37-42ee-ab79-0bcda049c671
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.129
X-Clv-S3-Version: 2.5
x-amz-request-id: 9fa96daa-9f37-42ee-ab79-0bcda049c671
Content-Type: application/xml
Content-Length: 374
```

```
<ListMultipartUploadsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Bucket>example-bucket</Bucket>
  <KeyMarker/>
  <UploadIdMarker/>
  <NextKeyMarker>multipart-object-123</NextKeyMarker>
  <NextUploadIdMarker>0000015a-df89-51d0-2790-dee1ac994053</NextUploadIdMarker>
  <MaxUploads>1000</MaxUploads>
  <IsTruncated>false</IsTruncated>
  <Upload>
    <Key>file</Key>
    <UploadId>0000015a-d92a-bc4a-c312-8c1c2a0e89db</UploadId>
    <Initiator>
      <ID>d4d11b981e6e489486a945d640d41c4d</ID>
      <DisplayName>d4d11b981e6e489486a945d640d41c4d</DisplayName>
    </Initiator>
    <Owner>
      <ID>d4d11b981e6e489486a945d640d41c4d</ID>
      <DisplayName>d4d11b981e6e489486a945d640d41c4d</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
    <Initiated>2017-03-16T22:09:01.002Z</Initiated>
  </Upload>
  <Upload>
    <Key>multipart-object-123</Key>
    <UploadId>0000015a-df89-51d0-2790-dee1ac994053</UploadId>
    <Initiator>
      <ID>d4d11b981e6e489486a945d640d41c4d</ID>
      <DisplayName>d4d11b981e6e489486a945d640d41c4d</DisplayName>
    </Initiator>
    <Owner>
      <ID>d4d11b981e6e489486a945d640d41c4d</ID>
      <DisplayName>d4d11b981e6e489486a945d640d41c4d</DisplayName>
    </Owner>
```

```
    <StorageClass>STANDARD</StorageClass>
    <Initiated>2017-03-18T03:50:02.960Z</Initiated>
  </Upload>
</ListMultipartUploadsResult>
```

List the protection configuration for a bucket

This implementation of the **GET** operation uses the *protection* subresource to return the protection configuration for the bucket.

Note: To support backwards compatibility with ClevOS 3.12.0, the response returns the protection state `Compliance` for all buckets that were created with a `Status` of either `Retention` or `Compliance`. The protection state `Retention` and `Compliance` are used interchangeably.

This operation does not make use of operation specific headers, query parameters, or payload elements.

Requests

Syntax

```
GET https://{endpoint}/{bucket-name}?protection= # path style
GET https://{bucket-name}.{endpoint}?protection= # virtual host style
```

Responses

Response elements

Table 12. GET bucket protection - response elements

Name	Description	Required
ProtectionConfiguration	Container for describing retention state of the bucket. **Type** Container **Children** **Status**, **MinimumRetention**, **DefaultRetention**, **MaximumRetention**, **EnablePermanentRetention** **Ancestor** None	Yes
Status	Status of protection for the bucket. **Type** Enum **Valid Values** **Disabled** | **Compliance** **Note:** To support backwards compatibility with ClevOS 3.12.0, the response returns the protection state `Compliance` for all buckets that were created with a `Status` of either `Retention` or `Compliance`. The protection state `Retention` and `Compliance` are used interchangeably. **Ancestor** **ProtectionConfiguration**	Yes
MinimumRetention	Minimum retention period for an object. If retention is not enabled this element is not returned. **Type** Container **Children** **Days** **Ancestor** **ProtectionConfiguration**	

Table 12. GET bucket protection - response elements (continued)

Name	Description	Required
DefaultRetention	Default retention period for an object. If retention is not enabled this element is not returned. A value of -2 indicates that the default retention period is set to permanent retention. **Type** Container **Children** **Days** **Ancestor** **ProtectionConfiguration**	
MaximumRetention	Maximum retention period for an object. If retention is not enabled this element is not returned. **Type** Container **Children** **Days** **Ancestor** **ProtectionConfiguration**	
Days	Specifies a retention period of time in days. **Type** Non-negative integer or -2 **Ancestor** **MinimumRetention**, **DefaultRetention**, **MaximumRetention**	Yes (if parent is specified)
EnablePermanentRetention	Specifies whether this bucket supports permanent retention of objects. This field returns true if permanent retention is enabled for this bucket, and thus for the system. **Default value if not specified** False **Type** Boolean **Ancestor** **ProtectionConfiguration** **Constraints** This field will only be included if the bucket has permanent retention enabled.	Yes, if permanent retention is enabled for this bucket

Examples

Sample request

This request lists the protection configuration for the "MyRetentionBucket" bucket

```
GET /MyRetentionBucket?protection HTTP/1.1
Host: myBucket.mydsNet.corp.com
Date: Wed, 8 Feb 2017 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
```

Sample response

This response lists the configuration for a bucket with Protection enabled.

```
HTTP/1.1 200 OK
Date: Wed, 8 Feb 2017 17:51:00 GMT
Connection: close

<ProtectionConfiguration>
  <Status>Compliance</Status>
  <MinimumRetention>
    <Days>100</Days>
  </MinimumRetention>
  <MaximumRetention>
    <Days>10000</Days>
  </MaximumRetention>
  <DefaultRetention>
    <Days>2555</Days>
  </DefaultRetention>
  <EnablePermanentRetention>true</EnablePermanentRetention>
</ProtectionConfiguration>
```

This response lists the configuration for a bucket with Protection disabled.

```
HTTP/1.1 200 OK
Date: Sat, 11 Feb 2017 12:00:00 GMT
Connection: close

<ProtectionConfiguration>
  <Status>Disabled</Status>
</ProtectionConfiguration>
```

Operations on objects

Upload an object

A `PUT` given a path to an object uploads the request body as an object. A SHA256 hash of the object is a required header. All objects are limited to 5TB in size. This operation does not make use of operation specific query parameters, or payload elements. If versioning is enabled on the bucket, objects will be versioned up to 1,000 times per object.

Syntax

```
PUT https://{endpoint}/{bucket-name}/{object-name} # path style
PUT https://{bucket-name}.{endpoint}/{object-name} # virtual host style
```

Specific headers for SSE-C

The following headers are available for buckets using Server Side Encryption with Customer-Provided Keys (SSE-C) enabled. Any request using SSE-C headers must be sent using SSL. Note that `ETag` values in response headers are *not* the MD5 hash of the object, but a randomly generated 32-byte hexadecimal string. Each version of an object can have a unique customer key. For more information on how to enable SSE-C, see the Manager Administration Guide.

Header	Type	Description
`x-amz-server-side-encryption-customer-algorithm`	string	This header is used to specify the algorithm and key size to use with the encryption key stored in `x-amz-server-side-encryption-customer-key` header. This value must be set to the string `AES256`.

Header	Type	Description
x-amz-server-side-encryption-customer-key	string	This header is used to transport the base 64 encoded byte string representation of the AES 256 key used in the server side encryption process.
x-amz-server-side-encryption-customer-key-MD5	string	This header is used to transport the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. The object store will use this value to validate the key passes in the x-amz-server-side-encryption-customer-key has not been corrupted during transport and encoding process. The digest must be calculated on the key BEFORE the key is base 64 encoded.

Sample request

```
PUT /example-bucket/queen-bee HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20160825T183001Z
x-amz-content-sha256: 309721641329cf441f3fa16ef996cf24a2505f91be3e752ac9411688e3435429
Content-Type: text/plain; charset=utf-8
Host: 67.228.254.193

Content-Length: 533

 The 'queen' bee is developed from larvae selected by worker bees and fed a
 substance referred to as 'royal jelly'. After a short while the 'queen' is
 the mother of nearly every bee in the hive, and the colony will fight
 fiercely to protect her.
```

Sample response

```
HTTP/1.1 200 OK
Date: Thu, 25 Aug 2016 18:30:02 GMT
X-Clv-Request-Id: 9f0ca49a-ae13-4d2d-925b-117b157cf5c3
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.121
X-Clv-S3-Version: 2.5
x-amz-request-id: 9f0ca49a-ae13-4d2d-925b-117b157cf5c3
ETag: "3ca744fa96cb95e92081708887f63de5"
Content-Length: 0
```

Sample request using SSE-C

```
PUT /example-bucket/queen-bee HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20160825T183001Z
x-amz-content-sha256: 309721641329cf441f3fa16ef996cf24a2505f91be3e752ac9411688e3435429
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key: MjRCRTJCQTNDQjdFOTkyMzY0NjZEN0NBMDhGQTBGRUQwNzFBMjEwMkQyNjU4MjNEOEMy
ODU5MkQxQ0ZEMkQ1OQ==
x-amz-server-side-encryption-customer-key-MD5: HBbrEt+ZH5iIfDNeBju03w==
Content-Type: text/plain; charset=utf-8
Host: 67.228.254.193

Content-Length: 533
```

```
The 'queen' bee is developed from larvae selected by worker bees and fed a
substance referred to as 'royal jelly'. After a short while the 'queen' is
the mother of nearly every bee in the hive, and the colony will fight
fiercely to protect her.
```

Sample response

```
HTTP/1.1 200 OK
Date: Thu, 25 Aug 2016 18:30:02 GMT
X-Clv-Request-Id: 9f0ca49a-ae13-4d2d-925b-117b157cf5c3
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.121
X-Clv-S3-Version: 2.5
x-amz-request-id: 9f0ca49a-ae13-4d2d-925b-117b157cf5c3
ETag: "3ca744fa96cb95e92081708887f63de5"
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key-MD5: HBbrEt+ZH5iIfDNeBju03w==
Content-Length: 0
```

Upload an object using HTML forms

A `POST` adds an object to a specified bucket using HTML forms. `POST` is an alternate form of `PUT` that enables browser-based uploads as a way of putting objects in buckets. Parameters that are passed to `PUT` through HTTP Headers are instead passed as form fields to `POST` in the multipart/form-data encoded message body. All objects are limited to 5TB in size. This operation does not make use of operation specific query parameters. If versioning is enabled on the bucket, objects will be versioned up to 1,000 times per object.

Syntax

```
POST https://{endpoint}/{bucket-name}/{object-name} # path style
POST https://{bucket-name}.{endpoint}/{object-name} # virtual host style
```

Form fields

Table 13. Form fields

Name	Description	Required
`AWSAccessKeyId`	The AWS access key ID of the owner of the bucket who grants an Anonymous user access for a request that satisfies the set of constraints in the policy. Type: String Default: None	Yes, if a policy document is included with the request
`acl`	Specifies an access control list. If an invalid access control list is specified, an error is generated. Type: String Default: Read Valid Values: `READ` | `WRITE` | `READ_ACP` | `WRITE_ACP` | `FULL_CONTROL`	No
`file`	File or text content. The file or text content must be the last field in the form. You cannot upload more than one file at a time. Type: File or text content Default: None	Yes

Table 13. Form fields (continued)

Name	Description	Required
`key`	The name of the uploaded key. To use the file name provided by the user, use the *${filename}* variable. For example, if the user Jerry uploads the file `mouse.jpg` and you specify `/user/jerry/${filename}`, the key name will be `/user/jerry/mouse.jpg`. Type: String Default: None	Yes
`policy`	Security Policy describing what is permitted in the request. Requests without a security policy are considered anonymous and work only on anonymously writable buckets. Type: String Default: None	Yes, if the bucket is not publicly writable
`success_action_status`	The status code returned to the client upon successful upload if `success_action_redirect` is not specified. Accepts the values 200, 201, or 204 (default). If the value is set to 200 or 204, the API returns an empty document with a 200 or 204 status code. If the value is set to 201, the API returns an XML document with a 201 status code. If the value is not set or if it is set to an invalid value, the API returns an empty document with a 204 status code. Type: String Default: None Note Some versions of the Adobe Flash player do not properly handle HTTP responses with an empty body. To support uploads through Adobe Flash, we recommend setting `success_action_status` to 201.	No
`x-amz-meta-*`	Headers starting with this prefix are user-defined metadata. Each one is stored and returned as a set of key-value pairs. The API doesn't validate or interpret user-defined metadata. For more information, see PUT Object. Type: String Default: None	No

Specific headers for SSE-C

The following headers are available for buckets using Server Side Encryption with Customer-Provided Keys (SSE-C) enabled. Any request using SSE-C headers must be sent using SSL. Note that `ETag` values in

response headers are *not* the MD5 hash of the object, but a randomly generated 32-byte hexadecimal string. Each version of an object can have a unique customer key. For more information on how to enable SSE-C, see the Manager Administration Guide.

Header	Type	Description
x-amz-server-side-encryption-customer-algorithm	string	This header is used to specify the algorithm and key size to use with the encryption key stored in x-amz-server-side-encryption-customer-key header. This value must be set to the string AES256.
x-amz-server-side-encryption-customer-key	string	This header is used to transport the base 64 encoded byte string representation of the AES 256 key used in the server side encryption process.
x-amz-server-side-encryption-customer-key-MD5	string	This header is used to transport the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. The object store will use this value to validate the key passes in the x-amz-server-side-encryption-customer-key has not been corrupted during transport and encoding process. The digest must be calculated on the key BEFORE the key is base 64 encoded.

Sample request

```
POST /example-bucket HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20160825T183001Z
x-amz-content-sha256: 309721641329cf441f3fa16ef996cf24a2505f91be3e752ac9411688e3435429
Content-Type: text/plain; charset=utf-8
Host: 67.228.254.193

Content-Type: multipart/form-data; boundary=----------------------------2393619733680
Content-Length: 639

----------------------------2393619733680
Content-Disposition: form-data; name="key"

uploads/${filename}
----------------------------2393619733680
Content-Disposition: form-data; name="Content-Type"

text/plain
----------------------------2393619733680
Content-Disposition: form-data; name="file"; filename="queen-bee.txt"
Content-Type: text/plain

The 'queen' bee is developed from larvae selected by worker bees and fed a
 substance referred to as 'royal jelly'. After a short while the 'queen' is
 the mother of nearly every bee in the hive, and the colony will fight
 fiercely to protect her.
----------------------------2393619733680--
```

Sample response

```
HTTP/1.1 204 No Content
Date: Thu, 25 Aug 2016 18:30:02 GMT
X-Clv-Request-Id: 9f0ca49a-ae13-4d2d-925b-117b157cf5c3
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.121
X-Clv-S3-Version: 2.5
x-amz-request-id: 9f0ca49a-ae13-4d2d-925b-117b157cf5c3
ETag: "3ca744fa96cb95e92081708887f63de5"
Content-Length: 0
```

Upload an object to a protected bucket

This enhancement of the **PUT** operation adds three new request headers: two for specifying the retention period in different ways, and one for adding a single legal hold to the new object. New errors are defined for illegal values for the new headers. If an object is under retention, it cannot be overwritten or deleted.

Objects in protected buckets that are no longer under retention (retention period has expired and the object does not have any legal holds), when overwritten, will again come under retention. The new retention period can be provided as part of the object overwrite request or the default retention time of the bucket will be given to the object.

The storage account user making a PUT Object with Retention Header request must have **WRITE_ACP** permissions for this object. For more information, see "Create an ACL for an object" on page 64.

AWS Signature V4 is required for this operation. It is recommended that protection headers are included in the signature and that the **x-amz-content-sha256** header is set to `STREAMING-AWS4-HMAC-SHA256-PAYLOAD` (chunked upload) or the payload checksum (single chunk upload with signed payload). It is recommended that users do not use **UNSIGNED-PAYLOAD** in the V4 signature calculation. If a **x-amz-content-sha256** header is not included in the V4 signature, then a `Content-MD5` header is required for this operation.

Requests

Syntax

```
PUT /BucketName/ObjectName HTTP/1.1
Host: myBucket.mydsNet.corp.com
Date: Wed, 8Feb 201717:50:00GMT
Authorization: {authorization-string}
Content-Type: text/plain
Retention-Period: 220752000
Retention-Legal-Hold-Id: SomeLegalHold2012
```

Note: The syntax only shows the new request headers.

Request headers

Table 14. Protection request headers

Name	Description	Required
Content-MD5	The base64-encoded 128-bit MD5 digest of the message (without the headers) according to RFC 1864. This header is used as a message integrity check to verify that the data is the same data that was originally sent. This header ensures no corruption is written to a retention bucket during transmission, which cannot be fixed after write. **Type** String **Default** None **Constraints** None	Yes, if protection configuration for target vault is **retention** and the V4 Signature is not present (pre-signed URL, POST Object) or the sha256 of content is not included in the signature. Otherwise, not required.
Retention-Period	Retention period to store on the object in seconds. The object can be neither overwritten nor deleted until the amount of time specified in the retention period has elapsed. If both *Retention-Period* and *Retention-Expiration-Date* are specified, a 400 error is returned. If neither is specified, the bucket's **DefaultRetention** period will be used. A retention period of -1 indicates indefinite retention for the object. A retention period of -1 can only be specified at the initial object creation. An object with a retention period of -1 cannot be overwritten or deleted. The indefinite retention period for an object can be changed to a finite value at any time with the retention extension operation. Once an object has been given a positive value for the retention period, that object cannot be given a retention period of -1. A retention period of -2 indicates permanent retention for the object. In order to specify -2, permanent retention must be enabled on the bucket in which this object resides. Once an object is permanently retained, the object, and thus the bucket which contains the object, cannot be deleted. 0 is a legal value assuming the bucket's minimum retention period is also 0. **Type** Non-negative integer (in seconds) or -1 or -2 **Constraints** **Retention-Period** must be greater than or equal to the bucket **MinimumRetention** and less than or equal to the bucket **MaximumRetention**	No

Table 14. Protection request headers (continued)

Name	Description	Required
Retention-Expiration-Date	Date on which it is possible to delete or modify the object. You can only specify this or the *Retention-Period* header. If both are specified a 400 error will be returned. If neither is specified the bucket's **DefaultRetention** period will be used. This header should be used to calculate a retention period in seconds and then stored in that manner. **Type** Date (ISO 8601 Format) **Constraints** The **Retention-Expiration-Date** must be greater than or equal to (current time + bucket **MinimumRetention**) and less than or equal to (current time + bucket **MaximumRetention**)	No
Retention-Legal-Hold-ID	A single legal hold to apply to the object. A legal hold is a Y character long string. The object cannot be overwritten or deleted until all legal holds associated with the object are removed. **Type** String	No

Specific headers for SSE-C

The following headers are available for buckets using Server Side Encryption with Customer-Provided Keys (SSE-C) enabled. Any request using SSE-C headers must be sent using SSL. Note that ETag values in response headers are *not* the MD5 hash of the object, but a randomly generated 32-byte hexadecimal string. Each version of an object can have a unique customer key. For more information on how to enable SSE-C, see the Manager Administration Guide.

Attention: SSE-C headers can be used to write or write objects from a protected bucket. However, it should be noted that SSE-C keys cannot be rotated for objects in a protected bucket.

Table 15. Specific headers for SSE-C

Header	Type	Description
x-amz-server-side-encryption-customer-algorithm	string	This header is used to specify the algorithm and key size to use with the encryption key stored in x-amz-server-side-encryption-customer-key header. This value must be set to the string AES256.
x-amz-server-side-encryption-customer-key	string	This header is used to transport the base 64 encoded byte string representation of the AES 256 key used in the server side encryption process.
x-amz-server-side-encryption-customer-key-MD5	string	This header is used to transport the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. The object store will use this value to validate the key passes in the x-amz-server-side-encryption-customer-key has not been corrupted during transport and encoding process. The digest must be calculated on the key BEFORE the key is base 64 encoded.

Upload an object to a protected bucket using HTML webforms

Specify retention periods and add a single legal hold to a protected object using webforms.

This enhancement of the **POST** operation adds three new form fields to the submitted webform: two for specifying the retention period in different ways, and one for adding a single legal hold to the new protected object. Errors are defined for illegal values for the headers, and if an object is under retention any overwrites are failed.

Protection headers included in a **POST** object request are ignored. Bucket retention values or, if specified in the webform, values in webform fields are used instead.

Related reference:

"Delete an object" on page 57
A DELETE given a path to an object deletes an object. This operation does not make use of operation specific query parameters (besides versioning), headers, or payload elements.

"Get the headers of an object" on page 50
A HEAD given a path to an object retrieves that object's headers. This operation does not make use of operation specific query parameters (besides versioning) or payload elements.

Requests

Syntax

POST Object

```
POST /BucketName/ObjectName HTTP/1.1
Host: myBucket.mydsNet.corp.com
Date: Wed, 8Feb 201717:50:00GMT
Authorization: authorization string
Content-Type: text/plain
Retention-Period: 220752000
Retention-Legal-Hold-Id: SomeLegalHold2012
```

Note: The syntax shows just the new headers.

Request Parameters

This implementation of the operation does not allow for these fields to be specified in the request headers. If these items are specified in the request headers, then they will be ignored.

Request Headers

Table 16. POST Object (webforms) - request headers

Name	Description	Required
Content-MD5	The base64-encoded 128-bit MD5 digest of the message (without the headers) according to RFC 1864. This header is used as a message integrity check to verify that the data is the same data that was originally sent. This form field is required if the bucket's protection level/state is **retention**. We require this to ensure no corruption during transmission that would otherwise be unfixable after the object is written into a retention bucket. **Type** String **Default** none **Constraints** none	Yes, if protection level is **retention**

Table 16. POST Object (webforms) - request headers (continued)

Name	Description	Required
Retention-Period	Retention period to store on the object in seconds. The object can be neither overwritten nor deleted until the amount of time specified in the retention period has elapsed. If this field and **Retention-Expiration-Date** are specified a 400 error is returned. If neither is specified the bucket's **DefaultRetention** period will be used. A retention period of -1 indicates indefinite retention for the object. A retention period of -1 can only be specified at the initial object creation. An object with a retention period of -1 cannot be overwritten or deleted. The indefinite retention period for an object can be changed to a finite value at any time with the retention extension operation. Once an object has been given a positive value for the retention period, that object cannot be given a retention period of -1. A retention period of -2 indicates permanent retention for the object. In order to specify -2, permanent retention must be enabled on the bucket in which this object resides. Once an object is permanently retained, the object, and thus the bucket which contains the object, cannot be deleted. 0 is a legal value assuming the bucket's minimum retention period is also 0. **Type** Non-negative integer (in seconds) or -1 or -2 **Constraints** **Retention-Period** must be greater than or equal to the bucket **MinimumRetention** and less than or equal to the bucket **MaximumRetention**	No
Retention-Expiration-Date	Date on which it will be legal to delete or modify the object. Clients may only specify this or the **Retention-Period** header. If both are specified a 400 error will be returned. If neither is specified the bucket's **DefaultRetention** period will be used. This header should be used to calculate a retention period in seconds and then stored in that manner. **Type** Date (ISO 8601 Format) **Constraints** The **Retention-Expiration-Date** must be greater than or equal to (current time + bucket **MinimumRetention**) and less than or equal to (current time + bucket **MaximumRetention**)	No
Retention-Legal-Hold-ID	A single legal hold to apply to the object. A legal hold is a **Y** character long string. A legal hold can only consist of US Alpha Numeric Characters (a-z, A-Z, 0-9) and the following symbols: ! _ . * ' ( ) - **Type** String	No

Request Elements

This enhancement to the operation does not use request elements.

Responses

Response Headers

This enhancement to the operation uses only response headers that are common to most responses.

Response Elements

This enhancement to the operation does not change any response elements.

Get the headers of an object

A `HEAD` given a path to an object retrieves that object's headers. This operation does not make use of operation specific query parameters (besides versioning) or payload elements.

Syntax

```
HEAD https://{endpoint}/{bucket-name}/{object-name} # path style
HEAD https://{bucket-name}.{endpoint}/{object-name} # virtual host style
```

Optional headers

Header	Type	Description
`range`	string	Returns the bytes of an object within the specified range.
`Mirror-Destination`	string	This header is applicable for listing of buckets in a protected mirror. The **`Mirror-Destination`** header specifies from which vault of the mirror to read. By default, if no explicit vault is specified, then the listing request will attempt to read from both vaults and provide a listing response that combines the list of objects on each component vault to the mirror excluding duplicates (object resides on both vaults in the mirror). If the **`Mirror-Destination`** header is specified and includes a valid Vault Identifier, the data returned will be from the Vault with the ID that matches what was provided in the header. The **`Mirror-Destination`** header is applicable only to protected mirrors, and the header is ignored otherwise. A failure to read from the specified vault will result in an error back to the HTTP client. **Type** String **Default** None **Constraints** {Valid Vault Identifier}

Specific headers for SSE-C

The following headers are available for buckets using Server Side Encryption with Customer-Provided Keys (SSE-C) enabled. Anyrequest using SSE-C headers must be sent using SSL. Note that `ETag` values in response headers are *not* the MD5 hash of the object, but a randomly generated 32-byte hexadecimal string. Each version of an object can have a unique customer key. For more information on how to enable SSE-C, see the Manager Administration Guide.

Header	Type	Description
x-amz-server-side-encryption-customer-algorithm	string	This header is used to specify the algorithm and key size to use with the encryption key stored in x-amz-server-side-encryption-customer-key header. This value must be set to the string AES256.
x-amz-server-side-encryption-customer-key	string	This header is used to transport the base 64 encoded byte string representation of the AES 256 key used in the server side encryption process.
x-amz-server-side-encryption-customer-key-MD5	string	This header is used to transport the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. The object store will use this value to validate the key passes in the x-amz-server-side-encryption-customer-key has not been corrupted during transport and encoding process. The digest must be calculated on the key BEFORE the key is base 64 encoded.

Sample request

```
HEAD /example-bucket/soldier-bee HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20160825T183244Z
Host: 67.228.254.193
```

Sample response

```
HTTP/1.1 200 OK
Date: Thu, 25 Aug 2016 18:32:44 GMT
X-Clv-Request-Id: da214d69-1999-4461-a130-81ba33c484a6
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.121
X-Clv-S3-Version: 2.5
x-amz-request-id: da214d69-1999-4461-a130-81ba33c484a6
ETag: "37d4c94839ee181a2224d6242176c4b5"
Content-Type: text/plain; charset=UTF-8
Last-Modified: Thu, 25 Aug 2016 17:49:06 GMT
Content-Length: 11
```

Get the headers of a protected object

A HEAD given a path to a protected object retrieves that object's headers. This operation does not make use of operation specific query parameters (besides versioning) or payload elements.

Optional headers

Header	Type	Description
range	string	Returns the bytes of an object within the specified range.

Header	Type	Description
Mirror-Destination	string	This header is applicable for listing of buckets in a protected mirror. The **Mirror-Destination** header specifies from which vault of the mirror to read. By default, if no explicit vault is specified, then the listing request will attempt to read from both vaults and provide a listing response that combines the list of objects on each component vault to the mirror excluding duplicates (object resides on both vaults in the mirror). If the **Mirror-Destination** header is specified and includes a valid Vault Identifier, the data returned will be from the Vault with the ID that matches what was provided in the header. The **Mirror-Destination** header is applicable only to protected mirrors, and the header is ignored otherwise. A failure to read from the specified vault will result in an error back to the HTTP client. **Type** String **Default** None **Constraints** {Valid Vault Identifier}

Specific headers for SSE-C

The following headers are available for buckets using Server Side Encryption with Customer-Provided Keys (SSE-C) enabled. Anyrequest using SSE-C headers must be sent using SSL. Note that `ETag` values in response headers are *not* the MD5 hash of the object, but a randomly generated 32-byte hexadecimal string. Each version of an object can have a unique customer key. For more information on how to enable SSE-C, see the Manager Administration Guide.

Attention: SSE-C headers can be used to write or write objects from a protected bucket. However, it should be noted that SSE-C keys cannot be rotated for objects in a protected bucket.

Header	Type	Description
`x-amz-server-side-encryption-customer-algorithm`	string	This header is used to specify the algorithm and key size to use with the encryption key stored in `x-amz-server-side-encryption-customer-key` header. This value must be set to the string `AES256`.
`x-amz-server-side-encryption-customer-key`	string	This header is used to transport the base 64 encoded byte string representation of the AES 256 key used in the server side encryption process.

Header	Type	Description
`x-amz-server-side-encryption-customer-key-MD5`	string	This header is used to transport the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. The object store will use this value to validate the key passes in the `x-amz-server-side-encryption-customer-key` has not been corrupted during transport and encoding process. The digest must be calculated on the key BEFORE the key is base 64 encoded.

Response headers

This enhancement to the operation add these new headers.

Table 17. Response headers

Name	Description	Required
`Retention-Period`	Retention period of the object in seconds. The object can be neither overwritten or deleted until the amount of time specified in the retention period has elapsed. If there is no retention period on the object this header is not returned. A retention period of `-1` indicates indefinite retention for the object. A retention period of `-1` can only be specified at the initial object creation. An object with a retention period of `-1` cannot be overwritten or deleted. The indefinite retention period for an object can be changed to a finite value at any time with the retention extension operation. Once an object has been given a positive value for the retention period, that object cannot be given a retention period of `-1`. A retention period of `-2` indicates permanent retention for the object. In order to specify `-2`, permanent retention must be enabled on the bucket in which this object resides. Once an object is permanently retained, the object, and thus the bucket which contains the object, cannot be deleted. **Type** Non-negative integer (in seconds) or `-1` or `-2`	No
`Retention-Legal-Hold-Count`	Returns the count of legal holds on the object. **Type** Non-negative integer	No
`Retention-Expiration-Date`	Computed date on which the retention period will expire. Calculated from object last-modified-time + retention period. If there is no retention period on the object, or the retention period is set to indefinite or permanent, this header is not returned. **Type** Date (ISO 8601 Format)	No

Download an object

A `GET` given a path to an object downloads the object. This operation does not make use of operation specific query parameters (besides versioning) or payload elements.

Syntax

```
GET https://{endpoint}/{bucket-name}/{object-name} # path style
GET https://{bucket-name}.{endpoint}/{object-name} # virtual host style
```

Optional headers

Header	Type	Description
range	string	Returns the bytes of an object within the specified range.
x-amz-storage-class	string	Return the storage class if set in the COS Manager in container mode.

Specific headers for SSE-C

The following headers are available for buckets using Server Side Encryption with Customer-Provided Keys (SSE-C) enabled. Any request using SSE-C headers must be sent using SSL. Note that ETag values in response headers are *not* the MD5 hash of the object, but a randomly generated 32-byte hexadecimal string. Each version of an object can have a unique customer key. For more information on how to enable SSE-C, see the Manager Administration Guide.

Header	Type	Description
x-amz-server-side-encryption-customer-algorithm	string	This header is used to specify the algorithm and key size to use with the encryption key stored in x-amz-server-side-encryption-customer-key header. This value must be set to the string AES256.
x-amz-server-side-encryption-customer-key	string	This header is used to transport the base 64 encoded byte string representation of the AES 256 key used in the server side encryption process.
x-amz-server-side-encryption-customer-key-MD5	string	This header is used to transport the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. The object store will use this value to validate the key passes in the x-amz-server-side-encryption-customer-key has not been corrupted during transport and encoding process. The digest must be calculated on the key BEFORE the key is base 64 encoded.

Sample request

```
GET /example-bucket/worker-bee HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20160825T183244Z
Host: 67.228.254.193
```

Sample response

```
HTTP/1.1 200 OK
Date: Thu, 25 Aug 2016 18:34:25 GMT
X-Clv-Request-Id: 116dcd6b-215d-4a81-bd30-30291fa38f93
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.121
X-Clv-S3-Version: 2.5
ETag: "d34d8aada2996fc42e6948b926513907"
Content-Type: text/plain; charset=UTF-8
Last-Modified: Thu, 25 Aug 2016 17:46:53 GMT
```

```
Content-Length: 467

 Female bees that are not fortunate enough to be selected to be the 'queen'
 while they were still larvae become known as 'worker' bees. These bees lack
 the ability to reproduce and instead ensure that the hive functions smoothly,
 acting almost as a single organism in fulfilling their purpose.
```

Download a protected object

This enhancement of the **GET** operations adds new response headers.

The storage account user making this request must have certain permissions for this object. In Vault Mode, if restrictive ACL is enabled then the storage account user must have READ_ACP permissions for the object. If restrictive ACL is not enabled, the storage account user must have READ_ACP permissions for the bucket. In Container Mode, the storage account user must have READ_ACP permissions on the object. For more information, see "Create an ACL for an object" on page 64.

Optional request header

Header	Type	Description
range	string	Returns the bytes of an object within the specified range.
Mirror-Destination	string	This header is applicable for listing of buckets in a protected mirror. The **Mirror-Destination** header specifies from which vault of the mirror to read. By default, if no explicit vault is specified, then the listing request will attempt to read from both vaults and provide a listing response that combines the list of objects on each component vault to the mirror excluding duplicates (object resides on both vaults in the mirror). If the **Mirror-Destination** header is specified and includes a valid Vault Identifier, the data returned will be from the Vault with the ID that matches what was provided in the header. The **Mirror-Destination** header is applicable only to protected mirrors, and the header is ignored otherwise. A failure to read from the specified vault will result in an error back to the HTTP client. **Type** String **Default** None **Constraints** {Valid Vault Identifier}

Specific headers for SSE-C

The following headers are available for buckets using Server Side Encryption with Customer-Provided Keys (SSE-C) enabled. Any request using SSE-C headers must be sent using SSL. Note that ETag values in response headers are *not* the MD5 hash of the object, but a randomly generated 32-byte hexadecimal string. Each version of an object can have a unique customer key. For more information on how to enable SSE-C, see the Manager Administration Guide.

Attention: SSE-C headers can be used to write or write objects from a protected bucket. However, it should be noted that SSE-C keys cannot be rotated for objects in a protected bucket.

Header	Type	Description
x-amz-server-side-encryption-customer-algorithm	string	This header is used to specify the algorithm and key size to use with the encryption key stored in x-amz-server-side-encryption-customer-key header. This value must be set to the string AES256.
x-amz-server-side-encryption-customer-key	string	This header is used to transport the base 64 encoded byte string representation of the AES 256 key used in the server side encryption process.
x-amz-server-side-encryption-customer-key-MD5	string	This header is used to transport the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. The object store will use this value to validate the key passes in the x-amz-server-side-encryption-customer-key has not been corrupted during transport and encoding process. The digest must be calculated on the key BEFORE the key is base 64 encoded.

Responses

Response headers

This enhancement to the operation add these new headers.

Table 18. Response headers

Name	Description	Required
Retention-Period	Retention period of the object in seconds. The object can be neither overwritten or deleted until the amount of time specified in the retention period has elapsed. If there is no retention period on the object this header is not returned. A retention period of -2 indicates permanent retention for the object. In order to specify -2, permanent retention must be enabled on the bucket in which this object resides. Once an object is permanently retained, the object, and thus the bucket which contains the object, cannot be deleted. **Type** Non-negative integer (in seconds) or -1 or -2	No
Retention-Legal-Hold-Count	Returns the count of legal holds on the object. **Type** Non-negative integer	No
Retention-Expiration-Date	Computed date on which the retention period will expire. Calculated from object last-modified-time + retention period. If there is no retention period on the object, or the retention period is set to indefinite or permanent, this header is not returned. **Type** Date (ISO 8601 Format)	No

Examples

Sample request

```
GET /BucketName/ObjectName HTTP/1.1
Host: myBucket.mydsNet.corp.com
Date: Sat, 11 Feb 2017 17:09:00 GMT
Authorization: {authorization-string}
```

Sample response

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Sat, 11 Feb 2017 17:10:00 GMT
Last-Modified: Thu, 2 Sep 2016 21:33:08 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Retention-Period: 220752000
Retention-Expiration-Date: Fri, 1 Sep 2023 21:33:08 GMT
Retention-Legal-Hold-Count: 1
Content-Length: 434234
[434234 bytes of object data]
```

Delete an object

A `DELETE` given a path to an object deletes an object. This operation does not make use of operation specific query parameters (besides versioning), headers, or payload elements.

Syntax

```
DELETE https://{endpoint}/{bucket-name}/{object-name} # path style
DELETE https://{bucket-name}.{endpoint}/{object-name} # virtual host style
```

Sample request

```
DELETE /example-bucket/soldier-bee HTTP/1.1
Authorization: {authorization-string}
Host: 67.228.254.193
```

Sample response

```
HTTP/1.1 204 No Content
Date: Thu, 25 Aug 2016 17:44:57 GMT
X-Clv-Request-Id: 8ff4dc32-a6f0-447f-86cf-427b564d5855
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.121
X-Clv-S3-Version: 2.5
```

Delete a protected object

This enhancement of the DELETE operation adds a new error status. If an object is protected, it cannot be deleted and a `451 Unavailable for Legal Reasons` is returned.

AWS Signature V4 is required for this operation.

Deleting multiple objects

A `POST` given a path to an bucket and proper parameters will delete a specified set of objects. This requires a `Content-MD5` header in addition to the `x-amz-content-sha256` header. This operation does not make use of operation specific query parameters (besides versioning), headers, or payload elements.

Syntax

```
POST https://{endpoint}/{bucket-name}/{object-name}?delete= # path style
POST https://{bucket-name}.{endpoint}/{object-name}?delete= # virtual host style
```

Sample request

```
POST /example?delete= HTTP/1.1
Authorization: {authorization-string}
Host: 67.228.254.193
x-amz-date: 20161205T231624Z
x-amz-content-sha256: 3ade096cd9471017539ede10c4d8aa05a1ecd015a16f4f090e9fcee92a816cf4
Content-MD5: zhi+TmIAhD2U3GfoYayyTQ==
Content-Type: text/plain; charset=utf-8

<?xml version="1.0" encoding="UTF-8"?>
<Delete>
    <Object>
         <Key>surplus-bee</Key>
    </Object>
    <Object>
         <Key>unnecessary-bee</Key>
    </Object>
</Delete>
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 30 Nov 2016 18:54:53 GMT
X-Clv-Request-Id: a6232735-c3b7-4c13-a7b2-cd40c4728d51
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.137
X-Clv-S3-Version: 2.5
x-amz-request-id: a6232735-c3b7-4c13-a7b2-cd40c4728d51
Content-Type: application/xml
Content-Length: 207

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<DeleteResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Deleted>
         <Key>surplus-bee</Key>
    </Deleted>
    <Deleted>
         <Key>unnecessary-bee</Key>
    </Deleted>
</DeleteResult>
```

Copy an object

A `PUT` given a path to a new object creates a new copy of another object specified by the `x-amz-copy-source` header. Unless otherwise altered the metadata remains the same, although any ACL is reset to `private` for the account creating the copy. This operation does not make use of operation specific query parameters (besides versioning) or payload elements.

Syntax

```
PUT https://{endpoint}/{bucket-name}/{object-name} # path style
PUT https://{bucket-name}.{endpoint}/{object-name} # virtual host style
```

Optional headers

Header	Type	Description
`x-amz-metadata-directive`	string (`COPY` or `REPLACE`)	`REPLACE` will overwrite original metadata with new metadata that is provided.
`x-amz-copy-source-if-match`	string (`ETag`)	Creates a copy if the specified `ETag` matches the source object.
`x-amz-copy-source-if-none-match`	string (`ETag`)	Creates a copy if the specified `ETag` is different from the source object.

Header	Type	Description
x-amz-copy-source-if-unmodified-since	string (timestamp)	Creates a copy if the the source object has not been modified since the specified date. Date must be a valid HTTP date (e.g. Wed, 30 Nov 2016 20:21:38 GMT).
x-amz-copy-source-if-modified-since	string (timestamp)	Creates a copy if the source object has been modified since the specified date. Date must be a valid HTTP date (e.g. Wed, 30 Nov 2016 20:21:38 GMT).

Specific headers for SSE-C

The following headers are available for objects being copied into buckets that have Server Side Encryption with Customer-Provided Keys (SSE-C) enabled. Any PUT request using SSE-C headers must be sent using SSL. Note that ETag values in response headers are *not* the MD5 hash of the object, but a randomly generated 32-byte hexadecimal string. For more information on how to enable SSE-C, see the Manager Administration Guide.

The specific SSE-C headers used to initially upload objects are required if the copy operation will encrypt the copy of the data at the target destination. If the original/source object was encrypted using SSE-C, the specific headers used for copying objects will need to be present to decrypt the object source. Copies of objects do not need to be encrypted with the same key.

Attention: SSE-C headers can be used to write or write objects from a protected bucket. However, it should be noted that SSE-C keys cannot be rotated for objects in a protected bucket.

Header	Type	Description
x-amz-copy-source-server-side-encryption-customer-algorithm	string	This header is used to specify the algorithm and key size to use with the encryption key stored in x-amz-copy-source-server-side-encryption-customer-key header. This value must be set to the string AES256.
x-amz-copy-source-server-side-encryption-customer-key	string	This header is used to transport the base 64 encoded byte string representation of the AES 256 key used in the server side encryption process.
x-amz-copy-source-server-side-encryption-customer-key-MD5	string	This header is used to transport the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. The object store will use this value to validate the key passes in the x-amz-copy-source-server-side-encryption-customer-key has not been corrupted during transport and encoding process. The digest must be calculated on the key BEFORE the key is base 64 encoded.

Sample request

This basic example takes the `bee` object from the `garden` bucket, and creates a copy in the `example` bucket with the new key `wild-bee`.

```
PUT /example-bucket/wild-bee HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161130T195251Z
x-amz-copy-source: /garden/bee
Host: 67.228.254.193
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 30 Nov 2016 19:52:52 GMT
X-Clv-Request-Id: 72992a90-8f86-433f-b1a4-7b1b33714bed
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.137
X-Clv-S3-Version: 2.5
x-amz-request-id: 72992a90-8f86-433f-b1a4-7b1b33714bed
ETag: "853aab195ce770b0dfb294a4e9467e62"
Content-Type: application/xml
Content-Length: 240

<CopyObjectResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <LastModified>2016-11-30T19:52:53.125Z</LastModified>
  <ETag>"853aab195ce770b0dfb294a4e9467e62"</ETag>
</CopyObjectResult>
```

Copy a protected object or copy an object to a protected bucket

This extension to the **PUT Object (Copy)** operation allows copying an object into another vault and changing the Protection settings on the copy.

If the destination vault is not protected, then the Protection headers of the object are not copied. If the destination vault is protected, then there three options: copy the existing protected state of the object (assuming the Protection period is not outside the range of the destination vaults min and max Protection period), apply new Protection settings, or allow the bucket's defaults to apply. If the source object is indefinitely retained (a retention period of `-1`), the retention period is copied as `-1` to the target protected bucket, irrespective of the minimum retention of the target bucket.

Note: The *Content MD-5* header is not required for a **PUT-COPY** request on a protected object.

The request header *Retention-Directive* has two values:

- **COPY**: The *Retention-Period* and *Retention-Legal-Hold-ID* state are copied from the source object
- **REPLACE**: You specify the *Retention-Period* and *Retention-Legal-Hold-ID* headers. If they are not specified, the destination bucket's defaults apply.

When copying an object, the new object's creation date is set to the time of the copy operation. When copying the legal holds for an object, the time of the copy will be used for the timestamp on each legal hold.

If copying from a non-protected bucket to a protected bucket, and *Retention-Directive* is set to **COPY**, then the operation is treated as a **REPLACE** with no other headers specified, and the destination object inherits the bucket settings.

A *Retention-Directive* of **REPLACE** with a retention period of `-1` is valid, as long as the object does not already exist as protected object on the target.

The **PUT-COPY** operation uses the fastest vault as the source of the copy if the source is a protected mirror.

Objects in protected buckets that are no longer under retention (retention period has expired and the object does not have any legal holds), when overwritten, will again come under retention. The new retention period can be provided as part of the object overwrite request or the default retention time of the bucket will be given to the object.

AWS Signature V4 is required for this operation. It is recommended that protection headers are included in the signature and that the **x-amz-content-sha256** header is set to STREAMING-AWS4-HMAC-SHA256-PAYLOAD (chunked upload) or the payload checksum (single chunk upload with signed payload).

Requests

Request headers

Table 19. Request headers

Name	Description	Required
x-amz-copy-source	Copy the **sourceObject** from this location and write it to the specified **/{BucketName}/{ObjectName}**	Yes
Retention-Directive	This header controls how the Protection state of the source object is copied to the destination object. If copied, the retention period and all legal holds are copied onto the new object. The legal hold date's is set to the date of the copy. If replaced, then you can specify the *Retention-Period* or *Retention-Expiration-Date*, and *Retention-Legal-Hold-ID* headers as defined in "Upload an object to a protected bucket" on page 45. If you do not specify one or both headers, the bucket's defaults are applied to the copy of the object. Since replace is the default value, if you copy without specifying any retention related headers, the destination bucket's defaults are used and the existing Protection state is not copied. **Type** Enum **Default** **REPLACE** **Valid values** **COPY** | **REPLACE** **Constraints** If the source objects retention period is outside the minimum and maximum range for the destination vault, a 400 is returned.	No

Table 19. Request headers (continued)

Name	Description	Required
Retention-Period	Retention period to store on the object in seconds. The object can be neither overwritten nor deleted until the amount of time specified in the retention period has elapsed. If this field and *Retention-Expiration-Date* are specified, a 400 error is returned. If neither is specified the bucket's **DefaultRetention** period is used. A retention period of -1 indicates indefinite retention for the object. A retention period of -1 can only be specified at the initial object creation. An object with a retention period of -1 cannot be overwritten or deleted. The indefinite retention period for an object can be changed to a finite value at any time with the retention extension operation. Once an object has been given a positive value for the retention period, that object cannot be given a retention period of -1. A retention period of -2 indicates permanent retention for the object. In order to specify -2, permanent retention must be enabled on the bucket in which this object resides. Once an object is permanently retained, the object, and thus the bucket which contains the object, cannot be deleted. 0 is a legal value assuming the bucket's minimum retention period is also 0. **Type** Non-negative integer (in seconds) or -1 or -2	No
Retention-Expiration-Date	Date on which it will be legal to delete or modify the object. You can only specify this or the *Retention-Period* header. If both are specified, a 400 error is returned. If neither is specified, the bucket's **DefaultRetention** period is used. This header should be used to calculate a retention period in seconds and then stored in that manner. **Type** Date (ISO 8601 Format)	No
Retention-Legal-Hold-ID	A single legal hold to apply to the object. A legal hold is a Y character long string. **Type** String	No

Specific headers for SSE-C

The following headers are available for objects being copied into buckets that have Server Side Encryption with Customer-Provided Keys (SSE-C) enabled. Any PUT request using SSE-C headers must be sent using SSL. Note that ETag values in response headers are *not* the MD5 hash of the object, but a randomly generated 32-byte hexadecimal string. For more information on how to enable SSE-C, see the Manager Administration Guide.

The specific SSE-C headers used to initially upload objects are required if the copy operation will encrypt the copy of the data at the target destination. If the original/source object was encrypted using SSE-C, the specific headers used for copying objects will need to be present to decrypt the object source. Copies of objects do not need to be encrypted with the same key.

Attention: SSE-C headers can be used to write or write objects from a protected bucket. However, it should be noted that SSE-C keys cannot be rotated for objects in a protected bucket.

Header	Type	Description
x-amz-copy-source-server-side-encryption-customer-algorithm	string	This header is used to specify the algorithm and key size to use with the encryption key stored in x-amz-copy-source-server-side-encryption-customer-key header. This value must be set to the string AES256.
x-amz-copy-source-server-side-encryption-customer-key	string	This header is used to transport the base 64 encoded byte string representation of the AES 256 key used in the server side encryption process.
x-amz-copy-source-server-side-encryption-customer-key-MD5	string	This header is used to transport the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. The object store will use this value to validate the key passes in the x-amz-copy-source-server-side-encryption-customer-key has not been corrupted during transport and encoding process. The digest must be calculated on the key BEFORE the key is base 64 encoded.

Retrieve an object's ACL

A GET given a path to an object given the parameter ?acl= retrieves the access control list for the object. This operation does not make use of operation specific headers, additional query parameters (besides versioning) or payload elements.

Syntax

```
GET https://{endpoint}/{bucket-name}/{object-name}?acl= # path style
GET https://{bucket-name}.{endpoint}/{object-name}?acl= # virtual host style
```

Sample request

```
GET /example-bucket/queen-bee?acl= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161207T155945Z
Host: 67.228.254.193
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 07 Dec 2016 15:59:46 GMT
X-Clv-Request-Id: 78541562-29bf-4800-9eb3-0c360f0a037a
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.137
X-Clv-S3-Version: 2.5
x-amz-request-id: 78541562-29bf-4800-9eb3-0c360f0a037a
Content-Type: application/xml
Content-Length: 550

<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>{owner-storage-account-uuid}</ID>
    <DisplayName>{owner-storage-account-uuid}</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
```

```
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>{owner-storage-account-uuid}</ID>
        <DisplayName>{owner-storage-account-uuid}</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Create an ACL for an object

A `PUT` issued to an object with the proper parameters creates an access control list (ACL) for that object. Access control lists allow for granting different sets of permissions to different storage accounts using the account's ID, or by using a pre-made ACL.

ACL grantees can be specified using any of the following methods:

Method	Description	Example
Canonical ID	User account UUID	`43a89ab8-a5e9-44bf-9671-d23a8729b2e0`
Email Address	Username of user account as set in COS Manager	`user1`
URI	Used for pre-defined groups. COS supports the All Users Group for bucket ACLs and the All Users Group and Authenticated Users URIs for Object ACLs. All other predefined groups are unsupported.	`http://acs.amazonaws.com/groups/global/AllUsers` or `http://acs.amazonaws.com/groups/global/AuthenticatedUsers`

The assigned permissions behave as follows:

Permission	When granted on a bucket	When granted on an object
READ	Allows grantee to list and read all objects in bucket	Allows grantee to read object data and metadata
WRITE	Allows grantee to create, overwrite and delete any object in bucket. Cannot be granted independently from READ permission.	N/A
READ_ACP	This permission does not exist for buckets; default setting is FULL_CONTROL	Allows grantee to read object ACL and object legal hold
WRITE_ACP	Default setting is FULL_CONTROL	Allows grantee to write ACL and legal hold for applicable object
FULL_CONTROL	Allows grantee READ, WRITE, READ_ACP and WRITE_ACP permissions on bucket	Allows grantee READ, READ_ACP and WRITE_ACP permissions on object

The following canned ACLs are supported by IBM COS. Values not listed below are not supported.

Canned ACL	Applies to	Notes
private	Bucket and object	When set on a bucket, the requestor is interpreted as the bucket owner.
public-read	Bucket and object	When set on a bucket, the requestor is interpreted as the bucket owner.

Canned ACL	Applies to	Notes
public-read-write	Bucket and object	When set on a bucket, the requestor is interpreted as the bucket owner.
authenticated-read	Bucket and object	Supported when set on an object only. Not supported as a bucket ACL.

Syntax

```
PUT https://{endpoint}/{bucket-name}/{object-name}?acl= # path style
PUT https://{bucket-name}.{endpoint}/{object-name}?acl= # virtual host style
```

Sample request (canned ACL)

```
PUT /example-bucket/queen-bee?acl= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161207T162842Z
x-amz-acl: public-read
Host: 67.228.254.193
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 07 Dec 2016 16:28:42 GMT
X-Clv-Request-Id: b8dea44f-af20-466d-83ec-2a8563f1617b
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.137
X-Clv-S3-Version: 2.5
x-amz-request-id: b8dea44f-af20-466d-83ec-2a8563f1617b
Content-Length: 0
```

Sample request (canned ACL in header)

It is also possible to assign a canned ACL directly when uploading an object by passing the `x-amz-acl` header and a canned ACL value. This example makes the `queen-bee` object publicly and anonymously accessible.

```
PUT /example-bucket/queen-bee HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161207T162842Z
x-amz-acl: public-read
Host: 67.228.254.193
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 07 Dec 2016 16:28:42 GMT
X-Clv-Request-Id: b8dea44f-af20-466d-83ec-2a8563f1617b
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.137
X-Clv-S3-Version: 2.5
x-amz-request-id: b8dea44f-af20-466d-83ec-2a8563f1617b
Content-Length: 0
```

Sample request (custom ACL)

This is an example of specifying a custom ACL to allow for another account to view the ACL for the “queen-bee” object, but not to access object itself. Additionally, a third account is given full access to the same object as another element of the same ACL.

```
PUT /example-bucket/queen-bee?acl= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20161207T163315Z
Content-Type: text/plain
Host: 67.228.254.193
Content-Length: 564

<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>{owner-storage-account-uuid}</ID>
    <DisplayName>OwnerDisplayName</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>{first-grantee-storage-account-uuid}</ID>
        <DisplayName>Grantee1DisplayName</DisplayName>
      </Grantee>
      <Permission>READ_ACP</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>{second-grantee-storage-account-uuid}</ID>
        <DisplayName>Grantee2DisplayName</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 07 Dec 2016 17:11:51 GMT
X-Clv-Request-Id: ef02ea42-6fa6-4cc4-bec4-c59bc3fcc9f7
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.137
X-Clv-S3-Version: 2.5
x-amz-request-id: ef02ea42-6fa6-4cc4-bec4-c59bc3fcc9f7
Content-Length: 0
```

Check an object's CORS configuration

An `OPTIONS` given a path to an object along with an origin and request type checks to see if that object is accessible from that origin using that request type. Unlike all other requests, an `OPTIONS` request does not require the `authorization` or `x-amx-date` headers. For protected mirrors, the `OPTIONS` uses the CORS configuration from the fastest vault of the protected mirror.

Syntax

```
OPTIONS https://{endpoint}/{bucket-name}/{object-name} # path style
OPTIONS https://{bucket-name}.{endpoint}/{object-name} # virtual host style
```

Sample request

```
OPTIONS /example-bucket/queen-bee HTTP/1.1
Access-Control-Request-Method: PUT
Origin: http://ibm.com
Host: 67.228.254.193
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 07 Dec 2016 16:23:14 GMT
X-Clv-Request-Id: 9a2ae3e1-76dd-4eec-a8f2-1a7f60f63483
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.137
```

```
X-Clv-S3-Version: 2.5
x-amz-request-id: 9a2ae3e1-76dd-4eec-a8f2-1a7f60f63483
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: PUT
Access-Control-Allow-Credentials: true
Vary: Origin, Access-Control-Request-Headers, Access-Control-Allow-Methods
Content-Length: 0
```

Add or remove a legal hold to or from a protected object

This implementation of the POST operation uses the **legalHold** subresource and **add** and **remove** query parameters to add or remove a single legal hold from a protected object in a protected vault.

- The object can support 100 legal holds
- A legal hold identifier is a string of maximum length 64 characters and a minimum length of 1 character. Valid characters are letters, numbers, !, _, ., *, ', (,), and -.
- If the addition of the given legal hold exceeds 100 total legal holds on the object, the new legal hold will not be added, a 400 error will be returned.
- If an identifier is too long it will not be added to the object and a 400 error is returned.
- If an identifier contains invalid characters, it will not be added to the object and a 400 error is returned.
- If an identifier is already in use on an object, the existing legal hold is not modified and the response indicates the identifier was already in use with a 409 error.
- If an object does not have retention period metadata, a 400 error is returned and adding or removing a legal hold is not allowed.

The legal hold identifiers are stored in the object metadata along with the timestamp of when they are added to the object. The presence of any legal hold identifiers prevents the modification or deletion of the object data, even if the retention period has expired. The object must be in a protected bucket.

The storage account user making a **POST ?legalHold** request must have **WRITE_ACP** permissions for this object. For more information, see “Create an ACL for an object” on page 64.

AWS Signature V4 is required for this operation.

This operation does not make use of operation specific payload elements.

Requests

Syntax

```
POST https://{endpoint}/{bucket-name}/{object-name}?legalHold&add={legal-hold-ID}= #path style
POST https://{bucket-name}.{endpoint}/{object-name}?legalHold&add={legal-hold-ID}= # virtual host style

POST https://{endpoint}/{bucket-name}/{object-name}?legalHold&remove={legal-hold-ID}= #path style
POST https://{bucket-name}.{endpoint}/{object-name}?legalHold&remove={legal-hold-ID}= # virtual host style
```

Request parameters

Table 20. Request parameters

Name	Description	Required
legalHold	Subresource for adding or removing legal hold identifiers from an object.	Yes

Table 20. Request parameters (continued)

Name	Description	Required
add	Client specified identifier for a legal hold, this is a string of at most 64 characters. Valid characters are letters, numbers, !, _, ., *, ', (,), and -. This legal hold identifier will be added to the object's existing set of legal hold identifiers. Only a single **legalHold** may be added to the object at a time. **add** and **remove** cannot be both specified. **Type** String **Length** <=64	Yes (unless **remove** is specified)
remove	Client specified identifier for a legal hold, this is a string of at most 64 characters. Valid characters are letters, numbers, !, _, ., *, ', (,), and -. This identifier will be removed from the object if present. Only a single **legalHold** may be removed from the object at a time. **add** and **remove** can not be both specified. **Type** String **Length** <=64	Yes (unless **add** is specified)

Examples

Sample request

Post Object Legal Hold

```
POST /BucketName/ObjectName?legalHold&add=legalHoldID HTTP/1.1
Host: myBucket.mydsNet.corp.com
Date: Wed, 8 Feb 2017 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
```

Sample Response

POST Object Legal Hold Response

```
HTTP/1.1 200 OK
Date: Wed, 8 Feb 2017 17:51:00 GMT
Connection: close
```

Extend the retention period of a protected object

This implementation of the **POST** operation uses the **extendRetention** sub-resource to extend the retention period of a protected object in a protected bucket.

- The retention period of an object can only be extended. It cannot be decreased from the currently configured value.
- The retention expansion value can be set in one of four ways:
  - Additional time from the current value
  - New retention period in seconds
  - New retention expiry date of the object
  - Extend retention period from current time
  - The total retention period of the object, after the retention extension, must be less than or equal to the System Maximum Duration

The current retention period stored in the object metadata is either increased by the given additional time or replaced with the new value, depending on the parameter that is set in the **POST /bucket/**

object?extendRetention request. In all cases, the extend retention parameter is checked against the current retention period and the extended parameter is only accepted if the following conditions are true:

- The retention period cannot be extended into the past.
- The retention period cannot be reduced.
- The retention extension must be less than the maximum retention period of the bucket.

The storage account user making a **POST /bucket/object?extendRetention** request must have the following permissions:

- Vault Mode
  - Restrictive ACLs disabled: WRITE_ACP permissions for this bucket
  - Reistritive ACLs enabled: WRITE_ACP permissions for this object
- Container Mode: WRITE_ACP permissions for this object

For more information, see "Create an ACL for an object" on page 64.

Requests

Syntax

POST Object Extend Retention

```
POST /BucketName/ObjectName?extendRetention HTTP/1.1
Additional-Retention-Period: 31470552
```

Request parameters

This implementation of the operation does not use request parameters.

Request headers

Table 21. Post Object extend retention - request headers

Name	Description	Required
Additional-Retention-Period	Additional time, in seconds, to add to the existing retention period for the object. If this field and *New-Retention-Time*, *New-Retention-Expiration-Date*, or *Extend-Retention-From-Current-Time* are specified, a 400 error will be returned. If none of the request headers are specified, a 400 error is returned. The retention period of an object can be extended up to the bucket *MaximumRetention* time from the time of the request. Thus, `[(object creation time)+ (old Retention Period) + (Additional-Retention-Period)] - (time of extension request) <= (MaximumRetention)` for the bucket. A retention period of `-2` indicates permanent retention for the object. In order to specify `-2`, permanent retention must be enabled on the bucket in which this object resides. Once an object is permanently retained, the object, and thus the bucket which contains the object, cannot be deleted. **Type** Non-negative integer (in seconds) or `-2` **Constraints** The total retention period of an object, after the retention extension, must be less than the System Maximum Duration period configured by the service administrator.	Yes, if no other headers in this table are specified. Only one header from this table can be specified at a time.

Table 21. Post Object extend retention - request headers (continued)

Name	Description	Required
New-Retention-Period	Retention period, in seconds, to use for the object in place of the existing retention period stored for the object. If this value is less than the existing value stored for the object, a 400 error is returned. If this field and *Additional-Retention-Period* or *New-Retention-Expiration-Date* or *Extend-Retention-From-Current-Time* are specified, a 400 error is returned. If none of the request headers are specified, a 400 error is returned. The retention period of an object can be extended up to the bucket *MaximumRetention* time from the time of the request. Thus, `[(object creation time) + (New-Retention-Period)] - (time of extension request) <= (MaximumRetention)` for the bucket. A retention period of `-2` indicates permanent retention for the object. In order to specify `-2`, permanent retention must be enabled on the bucket in which this object resides. Once an object is permanently retained, the object, and thus the bucket which contains the object, cannot be deleted. **Type** Non-negative integer (in seconds) or `-2` **Constraints** The total retention period of an object, after the retention extension, must be less than the system maximum retention period configured by the service administrator.	Yes, if no other headers in this table are specified. Only one header from this table can be specified at a time.
New-Retention-Expiration-Date	A new retention date to use for the object in place of the existing retention date. If this value is less than the existing value stored for the object, a 400 error is returned. If this field and *Additional-Retention-Period* or *New-Retention-Period* or *Extend-Retention-From-Current-Time* are specified, a 400 error is returned. If none of the request headers are specified, a 400 error is returned. The retention period of an object can be extended up to the bucket *MaximumRetention* time from the time of the request. Thus, `(New-Retention-Expiration-Date) - (time of extension request) <= (MaximumRetention)` for the bucket. **Type** Date (ISO 8601 Format) **Constraints** The total retention period of an object, after the retention extension, must be less than the system maximum retention period configured by the service administrator.	Yes, if no other headers in this table are specified. Only one header from this table can be specified at a time.

Table 21. Post Object extend retention - request headers (continued)

Name	Description	Required
Extend-Retention-From-Current-Time	Retention period, in seconds. The retention period is enforced from object the creation time until the current time plus the value specified in this header. This value has to be within the ranges defined for the bucket. If this field and *Additional-Retention-Period*, *New-Retention-Period*, or *New-Retention-Expiration-Date* are specified, a 400 error is returned. If none of the request headers are specified, a 400 error is returned to the user. The retention period of an object may be extended up to the bucket *MaximumRetention* time from the time of the request. Thus, (*Extend-Retention-From-Current-Time*) <= (*MaximumRetention*) for the bucket. A retention period of -2 indicates permanent retention for the object. In order to specify -2, permanent retention must be enabled on the bucket in which this object resides. Once an object is permanently retained, the object, and thus the bucket which contains the object, cannot be deleted. **Type** Non-negative integer (in seconds) or -2 **Constraints** The total retention period of an object, after the retention extension, must be less than the system maximum retention period configured by the service administrator.	Yes, if no other headers in this table are specified. Only one header from this table can be specified at a time.

Request elements

This implementation of the operation does not use request elements.

Responses

Response headers

This implementation of the operation uses only response headers that are common to most responses.

Response Elements

This implementation of the operation does not use response elements

Examples

Sample request

Post Object Extend Retention

```
POST /BucketName/ObjectName?extendRetention HTTP/1.1
Host: myBucket.mydsNet.corp.com
Date: Wed, 8Feb 201717:50:00GMT
Authorization: authorization string
Content-Type: text/plain
Additional-Retention-Period: 31470552
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 8Feb 201717:51:00GMT
Connection: close
```

List legal holds on a protected object

This implementation of the GET operation uses the **legalHold** sub-resource to return the list of legal holds on an object and related retention state in an XML response body.

- Object creation date
- Object retention period in seconds (our chosen unit of time for S3 API retention periods)
- Calculated retention expiration date based on the period and creation date
- List of legal holds
  - Legal hold identifier
  - Time stamp when legal hold was applied

If there are no legal holds on the object, an empty **LegalHoldSet** is returned.

If the object's retention period is indefinite or permanent, the retention expiration date returned in the response is not applicable. However, to maintain backwards compatibility, a date is returned. This date can be ignored, and is calculated as follows: Object Expiration Date = Object Creation Date + System Maximum Retention.

If there is no retention period specified on the object, a 404 error is returned.

The storage account user making a **GET ?legalHold** request must have **READ_ACP** permissions for this object. For more information, see "Create an ACL for an object" on page 64.

This operation does not make use of operation specific query parameters, headers, or payload elements.

Requests

Syntax

```
GET https://{endpoint}/{bucket-name}/{object-name}?legalHold
GET https://{bucket-name}.{endpoint}/{object-name}?legalHold
```

Optional request header

Table 22. Request header

Name	Description	Required
Mirror-Destination	This header is applicable for listing of buckets in a protected mirror. The **Mirror-Destination** header specifies from which vault of the mirror to read. By default, if no explicit vault is specified, then the listing request will attempt to read from both vaults and provide a listing response that combines the list of objects on each component vault to the mirror excluding duplicates (object resides on both vaults in the mirror). If the **Mirror-Destination** header is specified and includes a valid Vault Identifier, the data returned will be from the Vault with the ID that matches what was provided in the header. The **Mirror-Destination** header is applicable only to protected mirrors, and the header is ignored otherwise. A failure to read from the specified vault will result in an error back to the HTTP client. **Type** String **Default** None **Constraints** {Valid Vault Identifier}	No

Responses

Response elements

Table 23. Response elements

Name	Description	Required
RetentionState	Container for retention and legal hold state. **Type** Container **Children** **CreateTime**, **RetentionPeriod**, **LegalHoldSet**, **RetentionExpirationDate**	Yes
CreateTime	Date of object creation. **Type** Date (ISO 8601 Format) **Ancestor** **RetentionState**	Yes
RetentionPeriod	Retention period in seconds. **Type** Non-negative integer or -1 or -2 **Ancestor** **RetentionState**	Yes
RetentionExpirationDate	Date on which the retention period will expire. If **RetentionPeriod** is set to -1 or -2, this parameter is not applicable, but does return a date to maintain backwards compatibility (RetentionExpirationDate = Object Creation Date + System Maximum Retention). **Type** Date (ISO 8601 Format) **Ancestor** **RetentionState**	Yes
LegalHoldSet	Container to hold information about all the legal holds on an object. Can have many children. The set is specified in the response, but may be empty if there are no legal holds. **Type** Container **Ancestor** **RetentionState** **Children** **LegalHold**	Yes
LegalHold	Container to hold information about a single legal hold. **Type** Container **Ancestor** **ID**, **Date** **Children** **LegalHold**	No
ID	User-specified identifier for a legal hold. **Type** String **Ancestor** **LegalHold**	No

Table 23. Response elements (continued)

Name	Description	Required
Date	Date the legal hold was established on the object. **Type** Date (ISO 8601 Format) **Ancestor** **LegalHold**	No

Examples

Sample request

```
GET /BucketName/ObjectName?legalHold HTTP/1.1
Host: myBucket.mydsNet.corp.com
Date: Wed, 8 Feb 2017 17:50:00 GMT
Authorization: {authorization-string}
Content-Type: text/plain
```

Sample response

```
HTTP/1.1 200 OK
Date: Wed, 8 Feb 2017 17:51:00 GMT
Connection: close
<?xml version="1.0" encoding="UTF-8"?>
<RetentionState>
<CreateTime>Thu, 2 Sep 2016 21:33:08 GMT</CreateTime>
<RetentionPeriod>220752000</RetentionPeriod>
<RetentionExpirationDate>Fri, 1 Sep 2023 21:33:08
GMT</RetentionExpirationDate>
<LegalHoldSet>
<LegalHold>
<ID>SomeLegalHoldID</ID>
<Date>Thu, 15 Sep 2016 23:13:18 GMT</Date>
</LegalHold>
<LegalHold>
...
</LegalHold>
</LegalHoldSet>
</RetentionState>
```

Uploading objects in multiple parts

When working with larger objects, multipart upload operations are recommended to write objects into IBM COS. An upload of a single object can be performed as a set of parts and these parts can be uploaded independently in any order and in parallel. Upon upload completion, IBM COS then presents all parts as a single object. This provides many benefits: network interruptions do not cause large uploads to fail, uploads can be paused and restarted over time, and objects can be uploaded as they are being created.

Multipart uploads are only available for objects larger than 5MB. For objects smaller than 50GB, 500 parts sized 20MB to 100MB is recommended for optimum performance. For larger objects, part size can be increased without significant performance impact. Multipart uploads are limited to no more than 10,000 parts of 5GB each and a maximum object size of 5TB.

Due to the additional complexity involved, it is recommended that developers make use of S3 API libraries that provide multipart upload support.

Incomplete multipart uploads do persist until the object is deleted or the multipart upload is aborted with `AbortIncompleteMultipartUpload`. If an incomplete multipart upload is not aborted, the partial upload continues to use resources. Interfaces should be designed with this point in mind, and clean up incomplete multipart uploads.

There are three phases to uploading an object in multiple parts:

1. The upload is initiated and an `UploadId` is created.
2. Individual parts are uploaded specifying their sequential part numbers and the `UploadId` for the object.
3. When all parts are finished uploading, the upload is completed by sending a request with the `UploadId` and an XML block that lists each part number and it's respective `Etag` value.

Initiate a multipart upload

A `POST` issued to an object with the query parameter `upload` creates a new `UploadId` value, which is then be referenced by each part of the object being uploaded.

Syntax

```
POST https://{endpoint}/{bucket-name}/{object-name}?uploads= # path style
POST https://{bucket-name}.{endpoint}/{object-name}?uploads= # virtual host style
```

Specific headers for SSE-C

The following headers are available for buckets using Server Side Encryption with Customer-Provided Keys (SSE-C) enabled. Any request using SSE-C headers must be sent using SSL. Note that `ETag` values in response headers are *not* the MD5 hash of the object, but a randomly generated 32-byte hexadecimal string. Each version of an object can have a unique customer key. For more information on how to enable SSE-C, see the Manager Administration Guide. *These headers must be identical to those provided for each part of the multipart upload.*

Header	Type	Description
`x-amz-server-side-encryption-customer-algorithm`	string	This header is used to specify the algorithm and key size to use with the encryption key stored in `x-amz-server-side-encryption-customer-key` header. This value must be set to the string `AES256`.
`x-amz-server-side-encryption-customer-key`	string	This header is used to transport the base 64 encoded byte string representation of the AES 256 key used in the server side encryption process.
`x-amz-server-side-encryption-customer-key-MD5`	string	This header is used to transport the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. The object store will use this value to validate the key passes in the `x-amz-server-side-encryption-customer-key` has not been corrupted during transport and encoding process. The digest must be calculated on the key BEFORE the key is base 64 encoded.

Sample request

```
POST /some-bucket/multipart-object-123?uploads= HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20170303T203411Z
Host: 67.228.254.193
```

Sample response

```
HTTP/1.1 200 OK
Date: Fri, 03 Mar 2017 20:34:12 GMT
X-Clv-Request-Id: 258fdd5a-f9be-40f0-990f-5f4225e0c8e5
Accept-Ranges: bytes
Server: Cleversafe/3.9.1.114
X-Clv-S3-Version: 2.5
Content-Type: application/xml
Content-Length: 276
```

```
<InitiateMultipartUploadResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Bucket>some-bucket</Bucket>
  <Key>multipart-object-123</Key>
  <UploadId>0000015a-95e1-4326-654e-a1b57887784f</UploadId>
</InitiateMultipartUploadResult>
```

Upload a part

A `PUT` request issued to an object with query parameters `partNumber` and `uploadId` will upload one part of an object. The parts may be uploaded serially or in parallel, but must be numbered in order.

Syntax

```
PUT https://{endpoint}/{bucket-name}/{object-name}?partNumber={sequential-integer}&uploadId={uploadId}= # path style
PUT https://{bucket-name}.{endpoint}/{object-name}?partNumber={sequential-integer}&uploadId={uploadId}= # virtual host styl
```

Specific headers for SSE-C

The following headers are available for buckets using Server Side Encryption with Customer-Provided Keys (SSE-C) enabled. Any request using SSE-C headers must be sent using SSL. Note that `ETag` values in response headers are *not* the MD5 hash of the object, but a randomly generated 32-byte hexadecimal string. Each version of an object can have a unique customer key. For more information on how to enable SSE-C, see the Manager Administration Guide. *These headers must be identical to those provided when the multipart operation was initiated.*

Header	Type	Description
`x-amz-server-side-encryption-customer-algorithm`	string	This header is used to specify the algorithm and key size to use with the encryption key stored in `x-amz-server-side-encryption-customer-key` header. This value must be set to the string `AES256`.
`x-amz-server-side-encryption-customer-key`	string	This header is used to transport the base 64 encoded byte string representation of the AES 256 key used in the server side encryption process.
`x-amz-server-side-encryption-customer-key-MD5`	string	This header is used to transport the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. The object store will use this value to validate the key passes in the `x-amz-server-side-encryption-customer-key` has not been corrupted during transport and encoding process. The digest must be calculated on the key BEFORE the key is base 64 encoded.

Sample request

```
PUT /some-bucket/multipart-object-123?partNumber=1&uploadId=0000015a-df89-51d0-2790-dee1ac994053 HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20170318T035641Z
Content-Type: application/pdf
Host: 67.228.254.193
Content-Length: 13374550
```

Sample response

```
HTTP/1.1 200 OK
Date: Sat, 18 Mar 2017 03:56:41 GMT
X-Clv-Request-Id: 17ba921d-1c27-4f31-8396-2e6588be5c6d
Accept-Ranges: bytes
Server: Cleversafe/3.9.1.114
X-Clv-S3-Version: 2.5
ETag: "7417ca8d45a71b692168f0419c17fe2f"
Content-Length: 0
```

Upload a part for protected objects

This enhancement of the **UPLOAD PART** requires that a Content MD-5 header is included with each part that is uploaded if the V4 Signature does not include the sha256 of the content. However, the Content MD-5 header is not required for **UPLOAD PART-COPY** requests. The Content MD-5 header and V4 Signature signing are not required for **INITIATE MULTI PART UPLOAD** requests.

If an **UPLOAD PART**/ **UPLOAD PART-COPY**/ **INITIATE MULTI PART UPLOAD** request is sent with retention headers present in it, they are ignored and the normal operation continues.

Objects in protected buckets that are no longer under retention (retention period has expired and the object does not have any legal holds), when overwritten, will again come under retention. The new retention period can be provided as part of the object overwrite request or the default retention time of the bucket will be given to the object.

The UPLOAD PART-COPY uses the fastest vault as the source of the copy if the source is a protected mirror.

AWS Signature V4 is required for this operation. It is recommended that protection headers are included in the signature and that the **x-amz-content-sha256** header is set to STREAMING-AWS4-HMAC-SHA256-PAYLOAD (chunked upload) or the payload checksum (single chunk upload with signed payload). This operation does not make use of additional query parameters or payload elements.

Requests

Request headers

This implementation of the operation requires that the Content MD-5 header or V4 signing are included with each part that is uploaded.

Table 24. Request headers

Name	Description	Required
Content-MD5	The base64-encoded 128-bit MD5 digest of the message (without the headers) according to RFC 1864. This header is used as a message integrity check to verify that the data is the same data that was originally sent. It is required to ensure no corruption during transmission that would otherwise be unfixable after the object is written into a retention bucket. **Type** String **Default** None **Constraints** None	Yes, if protection configuration is **retention** and the V4 Signature is not present (Pre-signed URL, POST Object) or the sha256 of content is not included in the signature. Otherwise, not required.

Specific headers for SSE-C

The following headers are available for buckets using Server Side Encryption with Customer-Provided Keys (SSE-C) enabled. Any request using SSE-C headers must be sent using SSL. Note that `ETag` values in response headers are *not* the MD5 hash of the object, but a randomly generated 32-byte hexadecimal string. Each version of an object can have a unique customer key. For more information on how to enable SSE-C, see the Manager Administration Guide. *These headers must be identical to those provided when the multipart operation was initiated.*

Attention: SSE-C headers can be used to write or write objects from a protected bucket. However, it should be noted that SSE-C keys cannot be rotated for objects in a protected bucket.

Header	Type	Description
`x-amz-server-side-encryption-customer-algorithm`	string	This header is used to specify the algorithm and key size to use with the encryption key stored in `x-amz-server-side-encryption-customer-key` header. This value must be set to the string `AES256`.
`x-amz-server-side-encryption-customer-key`	string	This header is used to transport the base 64 encoded byte string representation of the AES 256 key used in the server side encryption process.
`x-amz-server-side-encryption-customer-key-MD5`	string	This header is used to transport the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. The object store will use this value to validate the key passes in the `x-amz-server-side-encryption-customer-key` has not been corrupted during transport and encoding process. The digest must be calculated on the key BEFORE the key is base 64 encoded.

Complete a multipart upload

A POST request issued to an object with query parameter uploadId and the appropriate XML block in the body will complete a multipart upload.

Syntax

```
POST https://{endpoint}/{bucket-name}/{object-name}?uploadId={uploadId}= # path style
POST https://{bucket-name}.{endpoint}/{object-name}?uploadId={uploadId}= # virtual host style
<CompleteMultipartUpload>
  <Part>
    <PartNumber>{sequential part number}</PartNumber>
    <ETag>{ETag value from part upload response header}</ETag>
  </Part>
</CompleteMultipartUpload>
```

Sample request

```
POST /some-bucket/multipart-object-123?uploadId=0000015a-df89-51d0-2790-dee1ac994053 HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20170318T035641Z
Content-Type: text/plain; charset=utf-8
Host: 67.228.254.193
Content-Length: 257
<CompleteMultipartUpload>
  <Part>
    <PartNumber>1</PartNumber>
    <ETag>"7417ca8d45a71b692168f0419c17fe2f"</ETag>
  </Part>
  <Part>
    <PartNumber>2</PartNumber>
    <ETag>"7417ca8d45a71b692168f0419c17fe2f"</ETag>
  </Part>
</CompleteMultipartUpload>
```

Sample response

```
HTTP/1.1 200 OK
Date: Fri, 03 Mar 2017 19:18:44 GMT
X-Clv-Request-Id: c8be10e7-94c4-4c03-9960-6f242b42424d
Accept-Ranges: bytes
Server: Cleversafe/3.9.1.114
X-Clv-S3-Version: 2.5
ETag: "765ba3df36cf24e49f67fc6f689dfc6e-2"
Content-Type: application/xml
Content-Length: 364
<CompleteMultipartUploadResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Location>http://67.228.254.193/example-bucket/multipart-object-123</Location>
  <Bucket>some-bucket</Bucket>
  <Key>multipart-object-123</Key>
  <ETag>"765ba3df36cf24e49f67fc6f689dfc6e-2"</ETag>
</CompleteMultipartUploadResult>
```

Complete a multipart upload for protected objects

This enhancement of the **Complete Multipart Upload** operation adds three new request headers: two for specifying the retention period in different ways, and one for adding a single legal hold to the new object. This operation returns the same errors as the **PUT/POST Object** operation.

Objects in protected buckets that are no longer under retention (retention period has expired and the object does not have any legal holds), when overwritten, will again come under retention. The new retention period can be provided as part of the object overwrite request or the default retention time of the bucket will be given to the object.

AWS Signature V4 is required for this operation. It is recommended that protection headers are included in the signature and that the **x-amz-content-sha256** header is set to STREAMING-AWS4-HMAC-SHA256-PAYLOAD (chunked upload) or the payload checksum (single chunk upload with signed payload). It is recommended that users do not use **UNSIGNED-PAYLOAD** in the V4 signature calculation. If a **x-amz-content-sha256** header is not included in the V4 signature, then a Content-MD5 header is required for this operation.

This operation does not make use of operation specific query parameters or payload elements.

Requests

Syntax

```
POST /BucketName/ObjectName?uloadId=uploadId HTTP/1.1
Host: myBucket.mydsNet.corp.com
Date: Wed, 8Feb 201717:50:00GMT
Authorization: authorization string
Content-Type: text/plain
Retention-Period: 220752000
Retention-Legal-Hold-Id: SomeLegalHold2012
<CompleteMultipartUpload>
  <Part>
    <PartNumber>PartNumber</PartNumber>
    <ETag>ETag</ETAG>
  </Part>
  ...
</CompleteMultiPartUpload>
```

Note: The syntax above only shows the new request headers

Request headers

Table 25. Protection request headers

Name	Description	Required
Content-MD5	The base64-encoded 128-bit MD5 digest of the message (without the headers) according to RFC 1864. This header is used as a message integrity check to verify that the data is the same data that was originally sent. This header ensures no corruption is written to a retention bucket during transmission, which cannot be fixed after write. For the **Complete Upload** request, the Content MD-5 is calculated against the parts list provided in the request. **Type** String **Default** None **Constraints** None	Yes, if protection level is **retention** and the V4 Signature is not present (IAM, Pre-signed URL, POST Object) or the sha256 of content is not included in the signature. Otherwise, not required.

Table 25. Protection request headers (continued)

Name	Description	Required
Retention-Period	Retention period to store on the object in seconds. The object can be neither overwritten nor deleted until the amount of time specified in the retention period has elapsed. If both *Retention-Period* and *Retention-Expiration-Date* are specified, a 400 error is returned. If neither is specified, the bucket's **DefaultRetention** period will be used. A retention period of -1 indicates indefinite retention for the object. A retention period of -1 can only be specified at the initial object creation. An object with a retention period of -1 cannot be overwritten or deleted. The indefinite retention period for an object can be changed to a finite value at any time with the retention extension operation. Once an object has been given a positive value for the retention period, that object cannot be given a retention period of -1. A retention period of -2 indicates permanent retention for the object. In order to specify -2, permanent retention must be enabled on the bucket in which this object resides. Once an object is permanently retained, the object, and thus the bucket which contains the object, cannot be deleted. 0 is a legal value assuming the bucket's minimum retention period is also 0. **Type** Non-negative integer (in seconds) or -1 or -2 **Constraints** **Retention-Period** must be greater than or equal to the bucket **MinimumRetention** and less than or equal to the bucket **MaximumRetention**	No
Retention-Expiration-Date	Date on which it will be legal to delete or modify the object. You can only specify this or the *Retention-Period* header. If both are specified a 400 error will be returned. If neither is specified the bucket's **DefaultRetention** period will be used. This header should be used to calculate a retention period in seconds and then stored in that manner. **Type** Date (ISO 8601 Format) **Constraints** The **Retention-Expiration-Date** must be greater than or equal to (current time + bucket **MinimumRetention**) and less than or equal to (current time + bucket **MaximumRetention**)	No
Retention-Legal-Hold-ID	A single legal hold to apply to the object. A legal hold is a Y character long string. **Type** String	No

Abort incomplete multipart uploads

A DELETE request issued to an object with query parameter uploadId will delete all unfinished parts of a multipart upload.

Sample request

```
DELETE /some-bucket/multipart-object-123?uploadId=0000015a-df89-51d0-2790-dee1ac994053 HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20170318T035641Z
Host: 67.228.254.193
```

Sample response

```
HTTP/1.1 204 No Content
Date: Thu, 16 Mar 2017 22:07:48 GMT
X-Clv-Request-Id: 06d67542-6a3f-4616-be25-fc4dbdf242ad
Accept-Ranges: bytes
Server: Cleversafe/3.9.1.114
X-Clv-S3-Version: 2.5
```

Notices

This information was developed for products and services offered in the US. This material might be available from IBM® in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Accesser®, Cleversafe®, ClevOS™, Dispersed Storage®, dsNet®, IBM Cloud Object Storage Accesser®, IBM Cloud Object Storage Dedicated™, IBM Cloud Object Storage Insight™, IBM Cloud Object Storage Manager™, IBM Cloud Object Storage Slicestor®, IBM Cloud Object Storage Standard™, IBM Cloud Object Storage System™, IBM Cloud Object Storage Vault™, SecureSlice™, and Slicestor® are trademarks or registered trademarks of Cleversafe, an IBM Company and/or International Business Machines Corp.

Other product and service names might be trademarks of IBM or other companies.

Homologation statement

This product may not be certified in your country for connection by any means whatsoever to interfaces of public telecommunications networks. Further certification may be required by law prior to making any such connection. Contact an IBM representative or reseller for any questions.

Printed in USA