Use Immutable Object Storage

Immutable Object Storage preserves electronic records and maintains data integrity. Retention policies ensure that data is stored in a WORM (Write-Once-Read-Many), non-erasable and non-rewritable manner. This policy is enforced until the end of a retention period and the removal of any legal holds.

This feature can be used by any user that needs long-term data retention in their environment, including but not limited to organizations in the following industries:

* Financial
* Healthcare
* Media content archives
* Anyone looking to prevent privileged modification or deletion of objects or documents

Retention policies can also be used by organizations that deal with financial records management, such as broker-dealer transactions, and might need to store data in a non-rewritable and non-erasable format.

Immutable Object Storage is available in certain regions only, see [Integrated Services](https://cloud.ibm.com/docs/services/cloud-object-storage/basics?topic=cloud-object-storage-service-availability#service-availability) for details. It also requires a Standard pricing plan. See [pricing](https://www.ibm.com/cloud/object-storage) for details.

It isn't possible to use Aspera high-speed transfer with buckets with a retention policy.

**Terminology and usage**

Retention period

The duration of time an object must be stored in the IBM Cloud Object Storage bucket.

Retention policy

A retention policy is enabled at the IBM Cloud Object Storage bucket level. Minimum, maximum and default retention period are defined by this policy and apply to all objects in the bucket.

Minimum retention period is the minimum duration of time an object must be kept unmodified in the bucket.

Maximum retention period is the maximum duration of time an object can be kept unmodified in the bucket.

If an object is stored in the bucket without specifying a custom retention period, the default retention period is used. The minimum retention period must be less than or equal to the default retention period, which in turn must be less than or equal to the maximum retention period.

A maximum retention period of 1000 years can be specified for the objects.

To create a retention policy on a bucket, you need Manager role. See [Bucket permissions](https://cloud.ibm.com/docs/services/cloud-object-storage/iam?topic=cloud-object-storage-iam-bucket-permissions) for more details.

Legal hold

Certain objects might need to be prevented from modification after a retention period expires. An example is an incomplete legal review, where records might need to be accessible for an extended duration beyond the retention period originally set. A legal hold flag can then be applied at the object level.

Legal holds can be applied to objects during initial uploads or after an object is written.

Note: A maximum of 100 legal holds can be applied per object.

Indefinite retention

Allows the user to set the object to be stored indefinitely until a new retention period is applied. This is set at a per object level.

Event-based retention

Immutable Object Storage allows users to set indefinite retention on the object if they are unsure of the final duration of the retention period, or want to use event-based retention. Once set to indefinite, user applications can then can change the object retention to a finite value later. For example, a company has a policy of retaining employee records for three years after the employee leaves the company. When an employee joins the company, the records that are associated with that employee can be indefinitely retained. When the employee leaves the company, the indefinite retention is converted to a finite value of three years from the current time, as defined by company policy. The object is then protected for three years after the retention period change. A user or third-party application can change the retention period from indefinite to finite retention that uses an SDK or REST API.

Permanent retention

Permanent retention can only be enabled at a IBM Cloud Object Storage bucket level with retention policy enabled and users are able to select the permanent retention period option during object uploads. Once enabled, this process can't be reversed and objects uploaded that use a permanent retention period **cannot be deleted**. It's the responsibility of the users to validate at their end if there's a legitimate need to **permanently** store objects by using Object Storage buckets with a retention policy.

When using Immutable Object Storage, you are responsible for ensuring that your IBM Cloud Account is kept in good standing per IBM Cloud policies and guidelines for as long as the data is subject to a retention policy. Refer to IBM Cloud Service terms for more information.

**Immutable Object Storage and considerations for various regulations**

When using immutable Object Storage, it is the client's responsibility to check for and ensure whether any of the feature capabilities that are discussed can be used to satisfy and comply with the key rules around electronic records storage and retention that is generally governed by:

* [Securities and Exchange Commission (SEC) Rule 17a-4(f)](https://www.ecfr.gov/cgi-bin/text-idx?SID=b6b7a79d18d000a733725e88d333ddb5&mc=true&node=pt17.4.240&rgn=div5#se17.4.240_117a_64),
* [Financial Industry Regulatory Authority (FINRA) Rule 4511(c)](https://www.finra.org/rules-guidance/rulebooks/finra-rules/4511), and
* [Commodity Futures Trading Commission (CFTC) Rule 1.31(c)-(d)](https://www.ecfr.gov/cgi-bin/text-idx?SID=2404f765a6f79e0b7fcf05b6844046cb&mc=true&node=se17.1.1_131&rgn=div8)

To assist clients in making informed decisions, IBM engaged Cohasset Associates Inc. to conduct an independent assessment of IBM’s Immutable Object Storage. Review [Cohasset Associates Inc.’s report](https://cos-docs.s3.us-east-2.amazonaws.com/6b294113-4d27-46cf-a460-b767bb20fcad.pdf) that provides details on the assessment of the Immutable Object Storage feature of IBM Cloud Object Storage.

Audit of access and transactions

Access log data for Immutable Object Storage to review changes to retention parameters, object retention period, and application of legal holds is available on a case-by-case basis by opening a customer service ticket.

**Using the console**

Retention policies can be added to new or existing empty buckets, and cannot be removed. For a new bucket, ensure that you are creating the bucket in a [supported region](https://cloud.ibm.com/docs/services/cloud-object-storage/basics?topic=cloud-object-storage-service-availability#service-availability), and then choose the **Add retention policy** option. For an existing bucket, ensure that it has no objects and then navigate to configuration settings and click the **Create policy** button below the bucket retention policy section. In either case, set a minimum, maximum, and default retention periods.

**Using the REST API, Libraries, and SDKs**

Several new APIs have been introduced to the IBM Cloud Object Storage SDKs to provide support for applications working with retention policies. Select a language (HTTP, Java, JavaScript, or Python) at the beginning of this page to view examples that use the appropriate Object Storage SDK.

All code examples assume the existence of a client object that is called cos that can call the different methods. For details on creating clients, see the specific SDK guides.

All date values used to set retention periods are Greenwich mean time. A Content-MD5 header is required to ensure data integrity, and is automatically sent when using an SDK.

Add a retention policy on an existing bucket

This implementation of the PUT operation uses the protection query parameter to set the retention parameters for an existing bucket. This operation allows you to set or change the minimum, default, and maximum retention period. This operation also allows you to change the protection state of the bucket.

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 bucket's default retention value is given to an object unless an object-specific value is provided when the object is created. 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 minimum and maximum supported values for the retention period settings MinimumRetention, DefaultRetention, and MaximumRetention are a minimum of 0 days and a maximum of 365243 days (1000 years).

A Content-MD5 header is required. This operation does not make use of extra query parameters.

For more information about endpoints, see [Endpoints and storage locations](https://cloud.ibm.com/docs/services/cloud-object-storage?topic=cloud-object-storage-endpoints#endpoints)

**Syntax**

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

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

**Example request**

PUT /example-bucket?protection= HTTP/1.1

Authorization: {authorization-string}

x-amz-date: 20181011T190354Z

x-amz-content-sha256: 2938f51643d63c864fdbea618fe71b13579570a86f39da2837c922bae68d72df

Content-MD5: GQmpTNpruOyK6YrxHnpj7g==

Content-Type: text/plain

Host: 67.228.254.193

Content-Length: 299

<ProtectionConfiguration>

<Status>Retention</Status>

<MinimumRetention>

<Days>100</Days>

</MinimumRetention>

<MaximumRetention>

<Days>10000</Days>

</MaximumRetention>

<DefaultRetention>

<Days>2555</Days>

</DefaultRetention>

</ProtectionConfiguration>

**Example response**

HTTP/1.1 200 OK

Date: Wed, 5 Oct 2018 15:39:38 GMT

X-Clv-Request-Id: 7afca6d8-e209-4519-8f2c-1af3f1540b42

Accept-Ranges: bytes

Server: Cleversafe/3.14.1

X-Clv-S3-Version: 2.5

x-amz-request-id: 7afca6d8-e209-4519-8f2c-1af3f1540b42

Content-Length: 0

Check retention policy on a bucket

This implementation of a GET operation fetches the retention parameters for an existing bucket.

**Syntax**

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

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

**Example request**

GET /example-bucket?protection= HTTP/1.1

Authorization: {authorization-string}

x-amz-date: 20181011T190354Z

Content-Type: text/plain

Host: 67.228.254.193

**Example response**

HTTP/1.1 200 OK

Date: Wed, 5 Oct 2018 15:39:38 GMT

X-Clv-Request-Id: 7afca6d8-e209-4519-8f2c-1af3f1540b42

Accept-Ranges: bytes

Server: Cleversafe/3.13.1

X-Clv-S3-Version: 2.5

x-amz-request-id: 7afca6d8-e209-4519-8f2c-1af3f1540b42

Content-Length: 299

<ProtectionConfiguration>

<Status>Retention</Status>

<MinimumRetention>

<Days>100</Days>

</MinimumRetention>

<MaximumRetention>

<Days>10000</Days>

</MaximumRetention>

<DefaultRetention>

<Days>2555</Days>

</DefaultRetention>

</ProtectionConfiguration>

If there is no protection configuration on the bucket, the server responds with disabled status instead.

<ProtectionConfiguration>

<Status>Disabled</Status>

</ProtectionConfiguration>

Upload an object to a bucket with retention policy

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, and if an object is under retention any overwrites will fail.

Objects in buckets with retention policy that are no longer under retention (retention period has expired and the object doesn't 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.

A Content-MD5 header is required.

These headers apply to POST object and multipart upload requests as well. If uploading an object in multiple parts, each part requires a Content-MD5 header.

| **Value** | **Type** | **Description** |
| --- | --- | --- |
| Retention-Period | Non-negative integer (seconds) | Retention period to store on the object in seconds. The object can be neither overwritten nor deleted until the amount of time that is 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. Zero (0) is a legal value assuming the bucket's minimum retention period is also 0. |
| Retention-expiration-date | Date (ISO 8601 Format) | Date on which it is 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. Supported ISO 8601 format is [YYYY]-[MM]-[DD]T[hh]:[mm]:[ss]Z or [YYYY][MM][DD]T[hh][mm][ss]Z (for example, 2020-11-28T03:10:01Z or 20201128T031001Z are both valid). |
| Retention-legal-hold-id | String | 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. |

Add or remove a legal hold to or from an object

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

The object can support 100 legal holds:

* A legal hold identifier is a string of maximum length 64 characters and a minimum length of one 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 is 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 that 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 presence of a retention period header is required, otherwise a 400 error is returned.

**Syntax**

POST https://{endpoint}/{bucket-name}/{object-name}?legalHold # path style

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

**Example request**

POST /BucketName/ObjectName?legalHold&add=legalHoldID HTTP/1.1

Host: myBucket.mydsNet.corp.com

Date: Fri, 8 Dec 2018 17:50:00 GMT

Authorization: authorization string

Content-Type: text/plain

**Example response**

HTTP/1.1 200 OK

Date: Fri, 8 Dec 2018 17:51:00 GMT

Connection: close

Extend the retention period of an object

This implementation of the POST operation uses the extendRetention query parameter 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 is set in one of three ways:

* additional time from the current value (Additional-Retention-Period or similar method)
* new extension period in seconds (Extend-Retention-From-Current-Time or similar method)
* new retention expiry date of the object (New-Retention-Expiration-Date or similar method)

The current retention period that is stored in the object metadata is either increased by the given extra time or replaced with the new value, depending on the parameter that is set in the 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 updated retention period is greater than the current retention period.

Supported ISO 8601 format for New-Retention-Expiration-Date is [YYYY]-[MM]-[DD]T[hh]:[mm]:[ss]Z or [YYYY][MM][DD]T[hh][mm][ss]Z (for example, 2020-11-28T03:10:01Z or 20201128T031001Z are both valid).

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.

**Syntax**

POST https://{endpoint}/{bucket-name}/{object-name}?extendRetention= # path style

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

**Example request**

POST /BucketName/ObjectName?extendRetention HTTP/1.1

Host: myBucket.mydsNet.corp.com

Date: Fri, 8 Dec 2018 17:50:00GMT

Authorization: authorization string

Content-Type: text/plain

Additional-Retention-Period: 31470552

**Example response**

HTTP/1.1 200 OK

Date: Fri, 8 Dec 2018 17:50:00GMT

Connection: close

List legal holds on an object

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

This operation returns:

* Object creation date
* Object retention period in seconds
* Calculated retention expiration date based on the period and creation date
* List of legal holds
* Legal hold identifier
* Timestamp when legal hold was applied

If there are no legal holds on the object, an empty LegalHoldSet is returned. If there is no retention period that is specified on the object, a 404 error is returned.

**Syntax**

GET https://{endpoint}/{bucket-name}/{object-name}?legalHold= # path style

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

**Example request**

GET /BucketName/ObjectName?legalHold HTTP/1.1

Host: myBucket.mydsNet.corp.com

Date: Fri, 8 Dec 2018 17:50:00 GMT

Authorization: {authorization-string}

Content-Type: text/plain

**Example response**