Veritas Access Appliance

Object Access API Guide

Version 8.1

Table of Contents

About the Object Access Server

Veritas Object Access provides an implementation of the simple object access service. This section explains the details of the RESTful APIs that are supported for the object access service. The supported APIs are compatible with Amazon S3 with only mentioned parameters.

Amazon S3 supports a wide range of services and APIs. The Veritas Access object access server does not support all of Amazon's APIs. Any API that is not documented in this guide should be treated as a non-supported API. There can be exceptions or differences in API behavior.

- User management
- Simple storage service protocol implementation for Amazon S3. The S3 service mandates signing of every request by using access and secret keys as specified by Amazon AWS.

You can use RESTful APIs or the user helper script to create the initial access and secret keys by using the object access user management APIs. After creating the initial AWS access and secret keys, you can use these keys for S3 service object access.

About the object access user management APIs

User authentication on the Veritas Access server is done using authentication services such as NIS, LDAP, and Active Directory(AD). For authentication to work correctly, the Veritas Access cluster must be configured with the correct authentication service and user password authentication needs to work. The user and identity management APIs depend on the correct working of the background authentication services.

All of the APIs must be called in HTTP POST requests. APIs are called by providing specific URI headers. The requests and responses are encoded in XML format.

Common Error Response

In case of errors, the following XML response is returned to the REST client.

- Response content type is "application/xml"
- RequestID is a unique ID generated per request
- Response content type is XML with following format.

```
<?xml version="1.0" encoding="UTF-8"?>
<ErrorResponse xmlns="http://iam.amazonaws.com/doc/2010-05-08/">
  <Error>
    <Type>Sender</Type>
    <Code>AccessDenied</Code>
    <Message>Access Denied.</Message>
  </Error>
  <RequestId>8c94eebd-52c8-4a69-b96e-926af9f791a5</RequestId>
</ErrorResponse>
```

Create access and secret keys for a given user

Use this API to create access and secret keys for a given user. Users can create a maximum of two keys. The secret key created in this API never gets displayed again, so users need to note down the access and secret keys in a secure and accessible location.

Request Parameters:

- HTTP URI Parameters - Parameter 'VRTSAction' Service request to execute, "CreateAccessKey". Parameter 'UserName' Valid username for authentication
- HTTP Header 'VRTSPassword'- Password for user. Password should be url percent encoded.

Example Request

```
http://admin.accessclus1:4567/?VRTSAction=CreateAccessKey
&UserName=user1
&Version=2010-05-08

VRTSPassword: urlencode(password)
```

Example Response

```
<?xml version="1.0" encoding="UTF-8"?>
<CreateAccessKeyResponse xmlns="http://iam.amazonaws.com/doc/2010-05-08/">
  <CreateAccessKeyResult>
    <AccessKey>
      <UserName>user1</UserName>
      <AccessKeyId>YTEzYjdhZGZkMzcwMmE</AccessKeyId>
      <Status>Active</Status>
      <SecretAccessKey>ZGJjYWJjNTZkOWRjYTkwODU4OWMyM2Y5YjI1ODE</SecretAccessKey>
    </AccessKey>
  </CreateAccessKeyResult>
  <ResponseMetadata>
    <RequestId>e101ac91-8a0b-4251-afa2-a6ee0411c59b</RequestId>
  </ResponseMetadata>
</CreateAccessKeyResponse>
```

Possible Error Response

- **Success** HTTP status code 200.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchEntity** Request was rejected because referenced entity does not exist. HTTP status code 404.
- **LimitExceeded** The request was rejected because it attempted to create/delete resources beyond the current account limits. HTTP status code 409.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Delete access and secret key for User

Delete access key. Users must provide a valid access key, username, and password. Use this API to delete the access key. Users must provide a valid access key, username, and password.

Request Parameters

- HTTP URI Parameters - Parameter 'VRTSAction' Service request to execute, "DeleteAccessKey"
  Parameter 'UserName' Valid username for authentication
  Parameter 'AccessKeyId' Access key ID to delete.
- HTTP Header 'VRTSPassword'- Password for user. Password should be url percent encoded.

Example Request

```
http://admin.accessclus1:4567/?VRTSAction=DeleteAccessKey
&UserName=user1
&Version=2010-05-08
&AccessKeyId=ZjhmZjM4ODEwN2ZhZGQ

VRTSPassword: urlencode(password)
```

Example Response

```
<DeleteAccessKeyResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08/">
  <ResponseMetadata>
    <RequestId>7a62c49f-347e-4fc4-9331-6e8eEXAMPLE</RequestId>
  </ResponseMetadata>
</DeleteAccessKeyResponse>
```

Possible Error Response

- **Success** HTTP status code 200.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchEntity** Request was rejected because referenced entity does not exist. HTTP status code 404.

- **InternalError** Request failed because of an internal server error. HTTP status code 500.

List access keys for user

Use this API to list access keys created by the user. Users must provide a valid username and password.

Request Parameters

- HTTP URI Parameters - Parameter 'VRTSAction' Service request to execute "ListAccessKeys" Parameter 'UserName' Valid username for authentication
- HTTP Header parameter 'VRTSPassword': Password for user. Password should be url percent encoded.

Example Request

```
http://admin.accessclus1:4567/?VRTSAction=ListAccessKeys
&UserName=user1
&Version=2010-05-08

VRTSPassword: urlencode(password)
```

Example Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ListAccessKeysResponse xmlns="http://iam.amazonaws.com/doc/2010-05-08/">
  <ListAccessKeysResult>
    <UserName>support</UserName>
    <AccessKeyMetadata>
      <member>
        <UserName>support</UserName>
    <AccessKeyId>YTEzYjdhZGZkMzcwMmE</AccessKeyId>
    <Status>Active</Status>
      </member>
      <member>
        <UserName>support</UserName>
    <AccessKeyId>YWU4NGRiYzRjNjQ5NGZ</AccessKeyId>
    <Status>Active</Status>
      </member>
    </AccessKeyMetadata>
      <IsTruncated>false</IsTruncated>
  </ListAccessKeysResult>
  <ResponseMetadata>
    <RequestId>c9639b1c-eeda-11e6-8a40-0026b97fab39</RequestId>
  </ResponseMetadata>
</ListAccessKeysResponse>
```

Possible Error response

- **Success** HTTP status code 200.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchEntity** Request was rejected because referenced entity does not exist. In this case, no access keys were created for the specified user. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Object Access simple storage service (S3) APIs

Common Error Response

In case of error following xml response is returned to REST client.

- Response content type will be "application/xml"
- RequestId is unique id generated per request
- Response content will be xml of following format.

```
<?xml version="1.0" encoding="UTF-8"?>
<ErrorResponse xmlns="http://iam.amazonaws.com/doc/2010-05-08/">
  <Error>
    <Type>Sender</Type>
    <Code>AccessDenied</Code>
    <Resource>/bucket1/file.txt</Resource>
    <Message>Access Denied.</Message>
  </Error>
  <RequestId>8c94eebd-52c8-4a69-b96e-926af9f791a5</RequestId>
</ErrorResponse>
```

Operations on Buckets

The following operations can be performed on the buckets:

Create Bucket (PUT Bucket)

Description - The Put Bucket API creates a new bucket. You are not permitted to create buckets using anonymous requests. You become the bucket owner when you create a bucket. The bucket name should comply with DNS naming conventions.
You can set the canned ACL, grant list, or the detailed ACL using the ACL request headers. You can create a Lock enabled bucket using x-amz-bucket-object-lock-enabled header. For more information on ACL, refer to ACL.

Request Syntax -

```
PUT / HTTP/1.1
Host: Bucket.s3.accesscluster:8143
x-amz-acl: ACL
x-amz-grant-full-control: GrantFullControl
x-amz-grant-read: GrantRead
x-amz-grant-read-acp: GrantReadACP
x-amz-grant-write: GrantWrite
x-amz-grant-write-acp: GrantWriteACP
x-amz-bucket-object-lock-enabled: ObjectLockEnabledForBucket
```

Request Parameters -

Bucket

Name of the bucket to be created.
Required: Yes
Type: String

Request Headers -

x-amz-acl

Set canned ACL on the bucket.
Type: String
Valid Values: private | public-read | public-read-write | authenticated-read

x-amz-bucket-object-lock-enabled

Specify whether the bucket should be object lock enabled or not.
Type: String
Valid Values: True | False

x-amz-grant-full-control

Allow read, write, read ACP and write ACP permissions on the bucket to grantee.
Type: String

x-amz-grant-read

Allow grantee to list objects in the bucket.
Type: String

x-amz-grant-read-acp

Allow grantee to read the bucket ACL.
Type: String

x-amz-grant-write

Allow grantee to put/delete objects.
Type: String

x-amz-grant-write-acp

Allow grantee to write the ACL for bucket.
Type: String

Response Syntax -

```
HTTP/1.1 200
```

Security Considerations - Only a caller with a valid access and secret key can create a bucket on the server.

Possible Error Response -

- **Success** HTTP status code 200.
- **InvalidArgument** Invalid Argument. HTTP status code 400.
- **InvalidBucketName** The specified bucket is not valid. HTTP status code 400.
- **AccessDenied** Access Denied. HTTP status code 403.
- **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. HTTP status code 409.
- **BucketAlreadyOwnedByYou** Your previous request to create the named bucket succeeded and you already own it. HTTP status code 409.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Delete Bucket

Description - The DELETE Bucket API deletes the bucket. All objects including all object versions and delete markers in the bucket must be deleted before the bucket itself can be deleted.

Request Syntax -

```
DELETE / HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameters -

Bucket

Name of the bucket to be deleted.
Required: Yes
Type: String

Response Syntax -

```
HTTP/1.1 204
```

Security Considerations - Only the bucket owner is allowed to perform DELETE operation.

Possible Error response -

- **Success** HTTP status code 204.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **BucketNotEmpty** The bucket you tried to delete is not empty. HTTP status code 409.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

GET Bucket ACL

Description - The Get Bucket Access Control List (ACL) API returns the access control list of the bucket.

Request Syntax -

```
GET /?acl HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameters -

Bucket

Bucket name for which you want to obtain ACL details.
Required: Yes
Type: String

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy>
  <Owner>
    <DisplayName>string</DisplayName>
    <ID>string</ID>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee>
        <DisplayName>string</DisplayName>
        <ID>string</ID>
        <xsi:type>string</xsi:type>
        <URI>string</URI>
      </Grantee>
      <Permission>string</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Response Body -

AccessControlPolicy

Root level tag for the AccessControlPolicy parameters.
Required: Yes

Grants

A list of grants.

Grantee

Details of the user which has permission.

DisplayName

Name of the user.

ID

UID of the user.

xsi:type

Type of grantee. Valid Values: CanonicalUser | Group

URI

URI of grantee group.

Permission

Specifies permission given to grantee. Valid Values: FULL_CONTROL | WRITE | WRITE_ACP | READ | READ_ACP

Owner

Container for the bucket owner's display name and ID.

DisplayName

Name of the owner.

ID

UID of the owner.

Security Considerations - For the Get Bucket ACL to succeed, the caller should have READ ACP permission on the bucket. For more information on ACL, refer to ACL.

Possible Error Response -

- **Success** HTTP status code 200.
- **AccessDenied** Access Denied. HTTP status code 403.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

GET Bucket Location

Description - The GET Bucket Location API returns the bucket's region using LocationConstraint of that object.

Veritas Object Access server returns an empty string as the location constraint. This empty string represents US East as the location for AWS S3 clients.

Request Syntax -

```
GET /?location HTTP/1.1
Host: Bucket.s3.accessscluster:8143
```

Request Parameters -

Bucket

The name of the bucket.
Required: Yes
Type: String

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<LocationConstraint>
  <LocationConstraint>string</LocationConstraint>
</LocationConstraint>
```

Response Body -

LocationConstraint

Root level tag for the LocationConstraint parameters.
Required: Yes

LocationConstraint

Veritas Access returns null location constraint. Some clients consider it as 'us-east-1'.

Security Considerations - Caller must have read permissions on the bucket for the Get Bucket Location API.

Possible Error Response -

- **Success** HTTP status code 200.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Get Bucket Versioning

Description - Get Bucket Versioning API returns the versioning state of a bucket. By default, versioning is always enabled on Object Lock enabled buckets. Enabling versioning on Object Lock disabled buckets is not supported.

Request Syntax -

```
GET /?versioning HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameters -

Bucket

Bucket name for which you want to get the versioning information.
Required: Yes
Type: String

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<VersioningConfiguration>
  <Status>string</Status>
</VersioningConfiguration>
```

Response Body -

VersioningConfiguration

Root level tag for the VersioningConfiguration parameters.
Required: Yes

Status

Versioning status of bucket.
Valid Values: Enabled

Security Considerations - The Get Bucket Versioning API is successful only if the caller has read permission.

Possible Error response -

- **Success** HTTP status code 200.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Get Object Lock Configuration

Description - Get Object Lock Configuration API gets the Object Lock configuration for a bucket. If default lock configuration is set on bucket, every new object will be applied with that default lock configuration. Existing objects will remain intact, they will not be modified for default configuration.

Request Syntax -

```
GET /?object-lock HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameters -

Bucket

Bucket name for which you want to get object lock configuration.
Required: Yes
Type: String

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ObjectLockConfiguration>
  <ObjectLockEnabled>string</ObjectLockEnabled>
  <Rule>
    <DefaultRetention>
      <Days>integer</Days>
      <Mode>string</Mode>
      <Years>integer</Years>
    </DefaultRetention>
  </Rule>
</ObjectLockConfiguration>
```

Response Body -

ObjectLockConfiguration

Root level tag for the ObjectLockConfiguration parameters.
Required: Yes

ObjectLockEnabled

It shows the Object Lock Configuration status of bucket.
Type: String
Valid Values: Enabled

Rule

It shows default Object Lock mode and period.

DefaultRetention

Specifies Default Lock configuration mode and period. The DefaultRetention require both retention mode and period. Retention period either be in days or years.

Days

Specifies number of days for which you want the default retention period to last on new objects.
Type: Integer

Mode

The default Object Lock retention mode, you want to set on new objects.
Valid Values: GOVERNANCE | COMPLIANCE

Years

Specifies number of years for which you want the default retention period to last on new objects. Maximum value can be 60 years.
Type: Integer

Security Considerations - For the Get Object Lock Configuration to succeed, the caller should have READ ACP permission on the bucket.

Possible Error response -

- **Success** HTTP status code 200.
- **ObjectLockConfigurationNotFoundError** Object Lock configuration does not exist for this bucket. HTTP status code 400.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Head Bucket

Description - The Head Bucket API is used to determine if a bucket exists or not. The operation returns a 200 OK if the bucket exists and the user has permissions to access it.

Request Syntax -

```
HEAD / HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameters -

Bucket

The name of the bucket.
Required: Yes
Type: String

Response Syntax -

```
HTTP/1.1 200
```

Security Considerations - Caller must have read permissions on the bucket for the Head Bucket API.

Possible Error Response -

- **Success** HTTP status code 200.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

List Buckets

Description - The LIST Buckets API lists all the buckets owned by user.

Request Syntax -

```
GET / HTTP/1.1
```

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult>
   <Buckets>
      <Bucket>
         <CreationDate>timestamp</CreationDate>
         <Name>string</Name>
      </Bucket>
   </Buckets>
   <Owner>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
   </Owner>
</ListAllMyBucketsResult>
```

Response Body -

ListAllMyBucketsResult

Root level tag for all bucket results.
Required: Yes

Buckets

The list of buckets owned by the user authenticated for the request.

Bucket

Information of the bucket.

CreationDate

Bucket creation date and time.

Name

Name of the bucket.

Owner

The owner of buckets.

DisplayName

Name of the owner.

ID

UID of owner.

Security Considerations - Any authenticated user should be able to list buckets by that user.

Possible Error Response -

- **Success** HTTP status code 200.
- **InvalidSecurity** The provided security credentials are not valid. HTTP status code 403.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

List Multipart Uploads

Description - The List Multipart Uploads API lists in-progress multipart uploads. An in-progress multipart upload is a multipart upload that has been initiated using the Create Multipart Upload request, but has not yet been completed or aborted. This operation returns a maximum of 1,000 multipart uploads in the response.

Request Syntax -

```
GET /?uploads&delimiter=Delimiter&key-
marker=KeyMarker&maxuploads=MaxUploads&prefix=Prefix&upload-id-marker=UploadIdMarker
HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameters -

Bucket

Name of the bucket on which the multipart upload was initiated.
Required: Yes
Type: String

delimiter

A delimiter is a character used to group keys. It rolls up the keys that contain the same character between the prefix and the first occurrence of the delimiter into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response. Each rolled-up result counts as only one return against the MaxKeys value. Veritas Access supports only "/" string as delimiter. For any other strings, Object Access server will return S3_InvalidArgument with HTTP_BAD_REQUEST error code.
Type: String

key-marker

The marker is the point where Object Access should begin listing objects. Veritas Access expects the key-marker which was returned by server in last request. Using object key as marker is not supported. The value of "NextMarker" of response should be used in request as marker.
Type: String

max-uploads

Limits the number of multipart uploads returned in the response. By default, the action returns up to 1,000 uploads.
Type: Integer

prefix

Limits the response to uploads that begin with the specified prefix.
Type: String

upload-id-marker

With key-marker, you can specify an upload-id-marker after which listing should begin. If key-marker is not specified, upload-id-marker will be ignored.
Type: String

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListMultipartUploadsResult>
  <Bucket>string</Bucket>
  <KeyMarker>string</KeyMarker>
  <UploadIdMarker>string</UploadIdMarker>
  <NextKeyMarker>string</NextKeyMarker>
  <Prefix>string</Prefix>
  <Delimiter>string</Delimiter>
  <NextUploadIdMarker>string</NextUploadIdMarker>
  <MaxUploads>integer</MaxUploads>
  <IsTruncated>boolean</IsTruncated>
  <Upload>
    <Initiated>timestamp</Initiated>
    <Initiator>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
    </Initiator>
    <Key>string</Key>
    <Owner>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
    </Owner>
    <StorageClass>string</StorageClass>
    <UploadId>string</UploadId>
  </Upload>
  ...
  <CommonPrefixes>
    <Prefix>string</Prefix>
  </CommonPrefixes>
  ...
</ListMultipartUploadsResult>
```

Response Body -

ListMultipartUploadsResult

Root level tag for the ListMultipartUploadsResult parameters.
Required: Yes

Bucket

Name of the bucket on which the multipart upload was initiated.

CommonPrefixes

When determining the number of returns, all the keys (up to 1,000) rolled into a common prefix count as one. CommonPrefixes contains all keys between Prefix and the next occurrence of the string specified by the delimiter.

Delimiter

Delimiter value passed in request.

IsTruncated

A flag indicating whether all the results satisfying the search criteria were returned by Access S3.

KeyMarker

Veritas Access expects the key-marker which was returned by server in last request. Using object key as key-marker is not supported. The value of "NextKeyMarker" of response should be used in request as key-marker.

MaxUploads

Limits the number of multipart uploads returned in the response. By default, the action returns up to 1,000 uploads.

NextKeyMarker

When the response is truncated, you can use this value as marker in subsequent request to get next set of objects.

NextUploadIdMarker

When the response is truncated, you can use this value as marker in subsequent request to get next set of objects.

Prefix

Limits the response to keys that begin with the specified prefix.

Upload

Information related to particular multipart upload. Response can contain zero or multiple uploads.

Initiated

This is the time and date when the multipart upload was initiated.
Type: Timestamp

Initiator

Information about user who initiated multipart upload.

DisplayName

Name of the initiator.

ID

UID of the initiator.

Key

Object name for which multipart upload was initiated.

Owner

Information about user who uploaded part of object.

DisplayName

Name of the user who uploaded part of object.

ID

UID of the user who uploaded part of object.

StorageClass

Storage class of the uploaded part.

UploadId

Upload ID that identifies the multipart upload.

UploadIdMarker

The value of upload-id-marker passed in the request.

Security Considerations - Caller should have read permission on bucket for List Multipart Uploads API.

Possible Error Response -

- **Success** HTTP status code 200.
- **InvalidArgument** Invalid Argument. HTTP status code 400.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

List Objects (GET Bucket Version 1)

Description - The GET bucket API returns a list of all the objects in a bucket. You can use the request parameters as selection criteria to return a subset of the objects in a bucket. The API returns objects with the latest version when versioning is enabled on the bucket. A 200 OK response can contain valid or invalid XML. Ensure that you design your application to parse the contents of the response and handle it appropriately.

The Keys in the response are not in lexicographical order. Application should not assume lexicographical order of response keys.

Request Syntax -

```
GET /?delimiter=Delimiter&marker=Marker&maxkeys=MaxKeys&prefix=Prefix HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameters -

Bucket

Name of the bucket containing the objects.
Required: Yes
Type: String

delimiter

A delimiter is a character used to group keys. It rolls up the keys that contain the same character between the prefix and the first occurrence of the delimiter into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response. Each rolled-up result counts as only one return against the MaxKeys value. Veritas Access supports only "/" string as delimiter. For any other strings, Object Access server will return S3_InvalidArgument with HTTP_BAD_REQUEST error code.
Type: String

marker

The marker is the point where Object Access should begin listing objects. Veritas Access expects the marker which was returned by server in last request. Using object key as marker is not supported. The value of "NextMarker" of response should be used in request as marker.
Type: String

max-keys

Limits the number of keys returned in the response. By default, the action returns up to 1,000 key names.
Type: Integer

prefix

Limits the response to keys that begin with the specified prefix.
Type: String

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult>
  <IsTruncated>boolean</IsTruncated>
  <Marker>string</Marker>
  <NextMarker>string</NextMarker>
  <Contents>
    <ETag>string</ETag>
    <Key>string</Key>
    <LastModified>timestamp</LastModified>
    <Owner>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
    </Owner>
    <Size>integer</Size>
    <StorageClass>string</StorageClass>
  </Contents>
  ...
  <Name>string</Name>
  <Prefix>string</Prefix>
  <Delimiter>string</Delimiter>
  <MaxKeys>integer</MaxKeys>
  <CommonPrefixes>
  <Prefix>string</Prefix>
  </CommonPrefixes>
  ...
</ListBucketResult>
```

Response Body -

ListBucketResult

Root level tag for the ListBucketResult parameters.
Required: Yes

CommonPrefixes

When determining the number of returns, all the keys (up to 1,000) rolled into a common prefix count as one. CommonPrefixes contains all keys between the prefix and the next occurrence of the string specified by the delimiter.

Contents

Metadata about each object that is returned.

ETag

MD5 digest of the object.

Key

Name of the object.

LastModified

Last modification date and time of the object.

Owner

Information of object owner.

DisplayName

Name of the object owner.

ID

UID of the object owner.

Size

Size of the object.

StorageClass

Storage class of the object.

Delimiter

Delimiter value passed in request.

IsTruncated

A flag indicating whether all the results satisfying the search criteria were returned by Access S3.

Marker

Indicates where listing begins in the bucket. Marker is included in response only if its passed in request.

MaxKeys

The maximum number of objects that can be returned in the response body.

Name

The name of the bucket.

NextMarker

When the response is truncated, you can use this value as the marker in the subsequent request to get the next set of objects.

Prefix

List the objects that begin with specified prefix.

Security Considerations - Caller must have read permissions on the bucket for the GET Bucket API.

Possible Error Response -

- **Success** HTTP status code 200.
- **InvalidArgument** Invalid Argument. HTTP status code 400.

- **AccessDenied** Access Denied. HTTP status code 403.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

List Objects (GET Bucket) Version 2

Description - The GET bucket API returns a list of all the objects in a bucket. You can use the request parameters as a selection criteria to return a subset of the objects in a bucket. The API returns objects with the latest version when versioning is enabled on the bucket. A 200 OK response can contain valid or invalid XML. Make sure to design your application to parse the contents of the response and handle it appropriately.

The keys in the response are not in lexicographical order. Application should not assume lexicographical order of response keys.

Request Syntax -

```
GET /?list-type=2&continuation-token=ContinuationToken&delimiter=Delimiter&max-
keys=MaxKeys&prefix=Prefix
StartAfter HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameter -

Bucket

Name of the bucket containing the objects.
Required: Yes
Type: String

continuation-token

Continuation-token is the point from where you want Object Access to start listing objects. Veritas Access expects the continuation-token which was returned by the server in last request. Using object key as continuation-token is not supported. The value of "NextContinuationToken" of response should be used in request as marker.
Type: String

delimiter

A delimiter is a character used to group keys. It rolls up the keys that contain the same character between the prefix and the first occurrence of the delimiter into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response. Each rolled-up result counts as only one return against the MaxKeys value. Veritas Access supports only "/" string as delimiter. For any other strings, Object Access server will return S3_InvalidArgument with HTTP_BAD_REQUEST error code.
Type: String

max-keys

Limits the number of keys returned in the response. By default, the action returns up to 1,000 key names.

Type: Integer

prefix

Limits the response to keys that begin with the specified prefix.
Type: String

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult>
  <IsTruncated>boolean</IsTruncated>
  <Contents>
    <ETag>string</ETag>
    <Key>string</Key>
    <LastModified>timestamp</LastModified>
    <Owner>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
    </Owner>
    <Size>integer</Size>
    <StorageClass>string</StorageClass>
  </Contents>
  ...
  <Name>string</Name>
  <Prefix>string</Prefix>
  <Delimiter>string</Delimiter>
  <MaxKeys>integer</MaxKeys>
  <CommonPrefixes>
    <Prefix>string</Prefix>
  </CommonPrefixes>
  ...
  <KeyCount>integer</KeyCount>
  <ContinuationToken>string</ContinuationToken>
  <NextContinuationToken>string</NextContinuationToken>
</ListBucketResult>
```

Response Body -

ListBucketResult

Root level tag for the ListBucketResult parameters. Required: Yes

CommonPrefixes

When determining the number of returns, all the keys (up to 1,000) rolled into a common prefix count as one. CommonPrefixes contains all keys between the prefix and the next occurrence of the string specified by the delimiter.

Contents

Metadata about each object that is returned.

ETag

MD5 digest of the object.

Key

Name of the object.

LastModified

Last modification date and time of the object.

Owner

Information of object owner.

DisplayName

Name of the object owner.

ID

UID of the object owner.

Size

Size of the object.

StorageClass

Storage class of the object.

ContinuationToken

The ContinuationToken is the point from where you want Object Access to start listing objects. Veritas Access expects ContinuationToken which was returned by server in last request. Using object key as ContinuationToken is not supported. The value of "NextContinuationToken" of response should be used in request as ContinuationToken.

Delimiter

Delimiter value passed in request.

IsTruncated

A flag indicating whether all the results satisfying the search criteria were returned by Access S3.

KeyCount

The number of objects returned in the response body.

MaxKeys

The maximum number of objects that can be returned in the response body.

Name

The name of the bucket.

NextContinuationToken

When the response is truncated, you can use this value as ContinuationToken in subsequent

request to get next set of objects.

Prefix

List the objects that begin with specified prefix.

Security Considerations - Caller must have read permissions on the bucket for the GET Bucket API.

Possible Error Response -

- **Success** HTTP status code 200.
- **InvalidArgument** Invalid Argument. HTTP status code 400.
- **AccessDenied** Access Denied. HTTP status code 403.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

List Object Versions

Description - The ListObjectVersions API returns metadata about all versions of the objects in a bucket. You can also use request parameters as selection criteria to return metadata about a subset of all the object versions.

The keys in the response are not in lexicographical order. Application should not assume lexicographical order of response keys. If number of versions are more than one page, then IsLatest may not be true for any version. Veritas Access recommend to use this API with 1000 maxkeys and object name as prefix, to list all object versions in one request.

Request Syntax -

```
GET /?versions&delimiter=Delimiter&key-
marker=KeyMarker&maxkeys=MaxKeys&prefix=Prefix&version-id-marker=VersionIdMarker
HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameter -

Bucket

Name of the bucket containing the objects.
Required: Yes
Type: String

delimiter

A delimiter is a character used to group keys. It rolls up the keys that contain the same character between the prefix and the first occurrence of the delimiter into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response. Each rolled-up result counts as only one return against the MaxKeys value. Veritas Access supports only "/" string as delimiter. For any other strings, Object Access server

will return S3_InvalidArgument with HTTP_BAD_REQUEST error code.
Type: String

key-marker

The key-marker is the point where Object Access should begin listing objects. Veritas Access expects the key-marker which was returned by server in last request. Using object key as key-marker is not supported. The value of "NextKeyMarker" of response should be used in request as marker.
Type: String

max-keys

Limits the number of keys returned in the response. By default, the action returns up to 1,000 key names.
Type: Integer

prefix

Limits the response to keys that begin with the specified prefix.
Type: String

version-id-marker

The version-id-marker is passed only with key-marker. This the the point to start listing of object versions.
Type: String

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult>
  <IsTruncated>boolean</IsTruncated>
  <KeyMarker>string</KeyMarker>
  <VersionIdMarker>string</VersionIdMarker>
  <NextKeyMarker>string</NextKeyMarker>
  <NextVersionIdMarker>string</NextVersionIdMarker>
  <Version>
    <ETag>string</ETag>
    <IsLatest>boolean</IsLatest>
    <Key>string</Key>
    <LastModified>timestamp</LastModified>
    <Owner>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
    </Owner>
    <Size>integer</Size>
    <StorageClass>string</StorageClass>
    <VersionId>string</VersionId>
  </Version>
  ...
  <DeleteMarker>
    <IsLatest>boolean</IsLatest>
    <Key>string</Key>
    <LastModified>timestamp</LastModified>
    <Owner>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
    </Owner>
    <VersionId>string</VersionId>
  </DeleteMarker>
  ...
  <Name>string</Name>
  <Prefix>string</Prefix>
  <Delimiter>string</Delimiter>
  <MaxKeys>integer</MaxKeys>
  <CommonPrefixes>
    <Prefix>string</Prefix>
  </CommonPrefixes>
  ...
</ListVersionsResult>
```

Response Body -

ListVersionsResult

Root level tag for the ListVersionsResult parameters.
Required: Yes

CommonPrefixes

When determining the number of returns, all the keys (up to 1,000) rolled into a common prefix count as one. CommonPrefixes contains all keys between the prefix and the next occurrence of the string specified by the delimiter.

DeleteMarker

Metadata about each delete marker. Response can have zero or more delete markers.

IsLatest

Specify if the object is latest.
Type: boolean

Key

Name of the delete marker.

LastModified

Last modification date and time of the delete marker. Type: Timestamp

Owner

Information of delete marker owner.

DisplayName

Name of the delete marker owner.

ID

UID of the delete marker owner.

VersionId

Specify version id of the delete marker.

Delimiter

Delimiter value passed in request.

IsTruncated

A flag indicating whether all the results satisfying the search criteria were returned by Access S3.

KeyMarker

Veritas Access expects the key-marker which was returned by server in last request. Using object key as key-marker is not supported. The value of "NextKeyMarker" of response should be used in request as key-marker.

MaxKeys

The maximum number of objects that can be returned in the response body.

Name

Name of the bucket containing the objects.

NextKeyMarker

When the response is truncated, you can use this value as marker in subsequent request to get next set of objects.

NextVersionIdMarker

When the response is truncated, you can use this value as marker in subsequent request to get next set of objects.

Prefix

Limits the response to keys that begin with the specified prefix.

Version

Metadata about object version.

ETag

MD5 digest of the object.

IsLatest

Specify if the object is latest.
Type: boolean

Key

Name of the object.

LastModified

Last modification date and time of the object.

Owner

Information of object owner.

DisplayName

Name of the object owner.

ID

UID of the object owner.

Size

Size of the object.

StorageClass

Storage class of the object.

VersionId

Specify version id of the object.

VersionIdMarker

Veritas Access expects version-id-marker which was returned by server in last request. Using object version id as version-id-marker is not supported. The value of "NextVersionIdMarker"

of response should be used in request as version-id-marker.

Security Considerations - Caller must have read permissions on the bucket for the GET Bucket API.

Possible Error Response -

- **Success** HTTP status code 200.
- **InvalidArgument** Invalid Argument. HTTP status code 400.
- **AccessDenied** Access Denied. HTTP status code 403.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Put Bucket ACL

Description - Put Bucket ACL sets permission on the existing bucket by using an Access Control List (ACL). You can use one of two ways to set the bucket permission:

1. Specify the ACL in the request body.
2. Specify the permission using request headers. If you specify ACL in the request headers, you must use either a canned ACL or the list of x-amz-grant-<read|write|read-acp|write-acp|full-control>. You cannot use both in the same request. For more information on ACL, refer to ACL.

Request Syntax -

```
PUT /?acl HTTP/1.1
Host: Bucket.s3.accesscluster:8143
x-amz-acl: ACL
Content-MD5: ContentMD5
x-amz-grant-full-control: GrantFullControl
x-amz-grant-read: GrantRead
x-amz-grant-read-acp: GrantReadACP
x-amz-grant-write: GrantWrite
x-amz-grant-write-acp: GrantWriteACP
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <AccessControlList>
    <Grant>
      <Grantee>
        <DisplayName>string</DisplayName>
        <ID>string</ID>
        <xsi:type>string</xsi:type>
        <URI>string</URI>
      </Grantee>
      <Permission>string</Permission>
    </Grant>
  </AccessControlList>
  <Owner>
    <DisplayName>string</DisplayName>
    <ID>string</ID>
  </Owner>
</AccessControlPolicy>
```

Request Parameter -

Bucket

Bucket name on which you want to set ACL.
Required: Yes
Type: String

Request Headers -

Content-MD5

Should be used for message integrity check.

x-amz-acl

Set canned ACL on the bucket.
Type: String
Valid Values: private | public-read | public-read-write | authenticated-read

x-amz-grant-full-control

Allow read, write, read ACP and write ACP permissions on the bucket to grantee.
Type: String

x-amz-grant-read

Allow grantee to list objects in the bucket.
Type: String

x-amz-grant-read-acp

Allow grantee to read the bucket ACL.
Type: String

x-amz-grant-write

Allow grantee to put/delete objects.
Type: String

x-amz-grant-write-acp

Allow grantee to write the ACL for bucket.
Type: String

Request Body -

AccessControlPolicy

Root level tag for the AccessControlPolicy parameters.
Required: Yes

Grants

A list of grants. Grantee type can be ‘CanonicalUser’ or 'Group'. While specifying type ‘CanonicalUser’, the user name of grantee and UID of user id should be specified.

Grantee

Name of the user which has permission.

DisplayName

Name of the user.

ID

UID of the user.

xsi:type

Type of grantee. Valid Values: CanonicalUser | Group

URI

URI/GID of grantee group.

Permission

Specifies permission given to grantee. Valid Values: FULL_CONTROL | WRITE | WRITE_ACP | READ | READ_ACP

Owner

Container for the bucket owner’s display name and ID. The ‘DisplayName’ should be the username of the grantee which should resolve to user id in server’s authentication method (passwd/NIS/LDAP/AD).

Required: No

DisplayName

Name of the owner.

ID

UID of owner.

Response Syntax -

```
HTTP/1.1 200
```

Security Considerations - Caller should have write-acp permission for the Put Bucket ACL API.

Possible Error response -

- **Success** HTTP status code 200.
- **BadDigest** The Content-MD5 you specified did not match with what was received. HTTP status code 400.
- **InvalidRequest** Specifying both Canned ACLs and Header Grants is not allowed. HTTP status code 400.
- **UnexpectedContent** This request does not support content. HTTP status code 400.
- **InvalidArgument** Invalid group URI or Invalid user id or maximum number of access control entries per ACL exceeded. HTTP status code 400.
- **MalformedACLError** The XML you provided was not well-formed or did not validate against our published schema. HTTP status code 400.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Put Object Lock Configuration

Description - The PUT Object Lock Configuration API places an object lock configuration on the specified bucket. The rule specified in the object lock configuration is applied by default to every new object placed in the specified bucket.

Request Syntax -

```
PUT /?object-lock HTTP/1.1
Host: Bucket.s3.accesscluster:8143
Content-MD5: ContentMD5
<?xml version="1.0" encoding="UTF-8"?>
<ObjectLockConfiguration>
   <ObjectLockEnabled>string</ObjectLockEnabled>
   <Rule>
      <DefaultRetention>
         <Days>integer</Days>
         <Mode>string</Mode>
         <Years>integer</Years>
      </DefaultRetention>
   </Rule>
</ObjectLockConfiguration>
```

Request Parameters -

Bucket

The name of bucket for which you want to create object lock configuration.
Required: Yes

Request Body -

ObjectLockConfiguration

Root level tag for applying or removing lock configuration on bucket.
Required: Yes

ObjectLockEnabled

Shows object lock is enabled on a bucket.
Value: Enabled

Rule

To specify rule used for applying object lock configuration for specified object.

DefaultRetention

Specify Default Lock configuration mode and period. The DefaultRetention require both retention mode and period. Retention period either be in days or years.

Days

Specify number of days for which you want the default retention period to last on new objects.
Type: Integer

Mode

The default Object Lock retention mode, you want to set on new objects.
Type: String
Valid Values: GOVERNANCE | COMPLIANCE

Years

Specify number of years for which you want the default retention period to last on new objects.
Type: Integer
Valid Range: 1-60

Response Syntax -

```
HTTP/1.1 200
```

Security Considerations - Caller should have write-acp permission for the Put Object Lock Configuration API.

Possible Error response -

- **Success** HTTP status code 200.
- **MissingRequestBodyError** Request body is empty. HTTP status code 400.
- **MalformedXML** The XML provided was not well formed. HTTP status code 400.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InvalidBucketState** The request is not valid with the current state of the bucket. HTTP status code 409.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Operations on Objects

Abort Multipart Upload

Description - The Abort Multipart Upload API aborts a multipart upload. After a multipart upload is aborted, no additional parts can be uploaded using that upload ID. The storage consumed by any previously uploaded parts will be freed.

Request Syntax -

```
DELETE /Key+?uploadId=UploadId HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameters -

Bucket

The name of bucket for which multipart upload was run.
Required: Yes
Type: String

Key

The name of the object for which multipart upload was initiated.
Required: Yes

Type: String

uploadId

Upload ID of multipart upload.
Required: Yes
Type: String
Valid range: 1-10000

Response Syntax -

```
HTTP/1.1 200
```

Security Considerations - By default, the bucket owner and the initiator of the multipart upload are allowed to perform this action.

Possible Error response -

- **Success** HTTP status code 204.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchUpload** The uploadId or Key might be invalid. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Complete Multipart upload

Description - The Complete Multipart Upload API completes a multipart upload by assembling previously uploaded parts. The part numbers must be in increments of 1. For example, the sequence (1, 3, 5, 7) will not work with the Veritas Object Access complete multipart API.

Request Syntax -

```
POST /Key+?uploadId=UploadId HTTP/1.1
Host: Bucket.s3.accesscluster:8143
<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUpload xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
   <Part>
      <ETag>string</ETag>
      <PartNumber>integer</PartNumber>
   </Part>
   ...
</CompleteMultipartUpload>
```

Request Parameters -

Bucket

The name of bucket on which Multipart Upload was initiated.
Required: Yes
Type: String

Key

The name of the destination object.
Required: Yes
Type: String

uploadId

ID for the initiated multipart upload.
Required: Yes
Valid range: 1-10000

Request Body -

CompleteMultipartUpload

Root level tag for the CompleteMultipartUpload parameters.
Required: Yes

Part

List of parts to create final object. It contains ETag and PartNumber.

ETag

ETag of the uploaded part.

PartNumber

PartNumber of the uploaded part.

Response Syntax -

```
HTTP/1.1 200
x-amz-version-id: VersionId
<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUploadResult>
   <Bucket>string</Bucket>
   <Key>string</Key>
   <ETag>string</ETag>
</CompleteMultipartUploadResult>
```

Response Headers -

x-amz-version-id

Version Id of the created object.

Response Body -

CompleteMultipartUploadResult

Root level tag for the CompleteMultipartUploadResult parameters.
Required: Yes

Bucket

The name of the Bucket.

Key

The name of the created object.

ETag

MD5 digest of the object.

Security Considerations - By default, the bucket owner and the initiator of the multipart upload are allowed to perform this action.

Possible Error response -

- **Success** HTTP status code 200.
- **BadDigest** The Content-MD5 you specified did not match what was received. HTTP status code 400.
- **MalformedXML** The XML you provided was not well-formed or did not validate against our published schema. HTTP status code 400.
- **InvalidArgument** Invalid Argument. Part number must be an integer between 1 and 10000, inclusive. HTTP status code 400.
- **InvalidPartOrder** The list of parts was not in ascending order. The parts list must specified in order of the part number. HTTP status code 400.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchUpload** The uploadId or Key might be invalid. HTTP status code 404.
- **NoSuchKey** The specified key does not exist. HTTP status code 404.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Copy Object

Description - The Copy Object API creates a copy of an object that is already stored in the Veritas Access S3 server.

```
The object created by this method is visible to all clients immediately after
the completion for the PUT operation. The server works on the following
consistency guarantees: -
```

- For (any number of) an ongoing download, if the PUT on the same object completes successfully, the ongoing request gets data and metadata of old object only.
- Any new GET object request after a successful PUT operation will get the latest object.
- Requests made from different nodes of cluster does not change the above semantics.

For information on ACL, refer to ACL.

Request Syntax -

```
PUT /Key+ HTTP/1.1
Host: Bucket.s3.accesscluster:8143
x-amz-acl: ACL
Content-Type: ContentType
x-amz-copy-source: CopySource
x-amz-copy-source-if-match: CopySourceIfMatch
x-amz-copy-source-if-modified-since: CopySourceIfModifiedSince
x-amz-copy-source-if-none-match: CopySourceIfNoneMatch
x-amz-copy-source-if-unmodified-since: CopySourceIfUnmodifiedSince
x-amz-grant-full-control: GrantFullControl
x-amz-grant-read: GrantRead
x-amz-grant-read-acp: GrantReadACP
x-amz-grant-write-acp: GrantWriteACP
x-amz-metadata-directive: MetadataDirective
x-amz-storage-class: StorageClass
x-amz-object-lock-mode: ObjectLockMode
x-amz-object-lock-retain-until-date: ObjectLockRetainUntilDate
```

Request Parameters -

Bucket

The name of bucket for which you want to Put Copy object.
Required: Yes
Type: String

Key

The name of the destination object.
Required: Yes
Type: String

Request Headers -

x-amz-acl

The ACL to be applied on the object.
Type: String
Valid Values: private | public-read | public-read-write | authenticated-read | bucket-owner-read | bucket-owner-full-control

Content-Type

Describes the format of content.

x-amz-copy-source

Specifies the source object for copy operation. You can specify this as bucket/object or bucket/dir/object. If you want to copy specific version of object, append ?versionId=<version-id> to the value (for example: bucket/object?versionId=3).
Type: String

x-amz-copy-source-if-match

Copies the source object if its entity tag (Etag) matches with specified value.

Type: String

x-amz-copy-source-if-modified-since

Copies the source object if it has been modified since specified time.
Type: Timestamp

x-amz-copy-source-if-none-match

Copies the source object if its entity tag (Etag) is different than specified value.
Type: String

x-amz-copy-source-if-unmodified-since

Copies the source object if it has not been modified since specified time.
Type: Timestamp

x-amz-grant-full-control

Grant READ, READ_ACP and WRITE_ACP permission on the object.
Type: String

x-amz-grant-read

Grant READ permission on the object.
Type: String

x-amz-grant-read-acp

Grant READ_ACP permission on the object.
Type: String

x-amz-grant-write-acp

Grant WRITE_ACP permission on the object.
Type: String

x-amz-metadata-directive

Specifies whether to copy metadata of source object or to replace it with provided metadata.
Type: String
Valid Values: COPY | REPLACE

x-amz-storage-class

Default value is STANDARD.
Type: String
Valid Values: STANDARD | REDUCED_REDUNDANCY | STANDARD_IA

x-amz-object-lock-mode

The Lock mode to be applied on object.
Type: String
Values: GOVERNANCE|COMPLIANCE

x-amz-object-lock-retain-until-date

Timestamp for object retention value.
Type: String

Type: Timestamp

Response Syntax -

```
HTTP/1.1 200
x-amz-copy-source-version-id: CopySourceVersionId
x-amz-version-id: VersionId
<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
   <ETag>string</ETag>
   <LastModified>timestamp</LastModified>
</CopyObjectResult>
```

Response Headers -

x-amz-copy-source-version-id

VersionId of source object.

x-amz-version-id

Version Id of newly created object.

Response Body -

CopyObjectResult

Root level tag for the CopyObjectResult parameters.
Required: Yes

ETag

Returns the ETag of new object.

LastModified

Creation time of new object.
Type: Timestamp

Security Considerations - User should have read permission on the source object to perform the Put Copy operation. Public user is not allowed to perform put object copy operation.

Possible Error response -

- **Success** HTTP status code 200.
- **InvalidArgument** Invalid Argument or Invalid version id specified. HTTP status code 400.
- **InvalidRequest** Invalid Request or The source of a copy request may not specifically refer to a delete marker by version id. HTTP status code 400.
- **InvalidStorageClass** The storage class you specified is not valid. HTTP status code 400.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **NoSuchKey** The specified key does not exist. HTTP status code 404.

- **MissingContentLength** Content-Length HTTP header missing in request. HTTP status code 411.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Create Multipart upload

Description - The Create Multipart Upload API initiates a multipart upload and returns an upload ID. This upload ID is used to associate all the parts in the specific multipart upload.

For information on ACL, refer to ACL.

Request Syntax -

```
POST /{Key+}?uploads HTTP/1.1
Host: Bucket.s3.accesscluster:8143
x-amz-acl: ACL
Content-Type: ContentType
x-amz-grant-full-control: GrantFullControl
x-amz-grant-read: GrantRead
x-amz-grant-read-acp: GrantReadACP
x-amz-grant-write-acp: GrantWriteACP
x-amz-storage-class: StorageClass
x-amz-object-lock-mode: ObjectLockMode
x-amz-object-lock-retain-until-date: ObjectLockRetainUntilDate
```

Request Parameters -

Bucket

The name of bucket for which you want initiate multipart upload.
Required: Yes
Type: String

Key

The name of the destination object.
Required: Yes
Type: String

Request Headers -

x-amz-acl

The ACL to be applied on the object.
Type: String
Valid Values: private | public-read | public-read-write | authenticated-read | bucket-owner-read | bucket-owner-full-control

Content-Type

Describes the format of content.

x-amz-grant-full-control

Grant READ, READ_ACP and WRITE_ACP permission on the object.

Type: String

x-amz-grant-read

Grant READ permission on the object.
Type: String

x-amz-grant-read-acp

Grant READ_ACP permission on the object.
Type: String

x-amz-grant-write-acp

Grant WRITE_ACP permission on the object.
Type: String

x-amz-storage-class

Default value is STANDARD.
Type: String
Valid Values: STANDARD | REDUCED_REDUNDANCY | STANDARD_IA

x-amz-object-lock-mode

The Lock mode to be applied on object.
Type: String
Values: GOVERNANCE|COMPLIANCE

x-amz-object-lock-retain-until-date

Timestamp for object retention value.
Type: Timestamp

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<InitiateMultipartUploadResult>
   <Bucket>string</Bucket>
   <Key>string</Key>
   <UploadId>string</UploadId>
</InitiateMultipartUploadResult>
```

Response Body -

InitiateMultipartUploadResult

Root level tag for the InitiateMultipartUploadResult parameters.
Required: Yes

Bucket

The name of the bucket.

Key

Object name for which multipart upload initiated.

UploadId

ID for the initiated multipart upload.

Security Considerations - User should have write permission on the bucket to perform Create Multipart Upload.

Possible Error response -

- **Success** HTTP status code 200.
- **InvalidArgument** Invalid Argument. HTTP status code 400.
- **InvalidRequest** Invalid Request HTTP status code 400.
- **InvalidStorageClass** The storage class you specified is not valid. HTTP status code 400.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Delete Object

Description - The Delete Object API deletes the specified object in the bucket for a non-versioned bucket. If versioning is enabled on the bucket and VersionId is passed, the specified version of the object is deleted. If versioning is enabled on the bucket and VersionId is not passed, a DeleteMarker is created for the object.

Request Syntax -

```
DELETE /Key+?versionId=VersionId HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameter -

Bucket

The name of the bucket containing object.
Required: Yes
Type: String

Key

The name of the object you want to delete.
Required: Yes
Type: String

versionId

The version ID of the Object.
Type: String

Response Syntax -

```
HTTP/1.1 204
x-amz-delete-marker: DeleteMarker
x-amz-version-id: VersionId
```

Response Headers -:

x-amz-delete-marker

Specifies if the deleted object is a delete marker or not.

x-amz-version-id

Specifies Version Id of deleted object.

Security Considerations - Caller should have write permissions on the bucket to delete an object.

Possible Error response -

- **Success** HTTP status code 204.
- **InvalidArgument** Invalid Argument. Invalid version id specified. HTTP status code 400.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchKey** The specified key does not exist. HTTP status code 404.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Delete Objects

Description - Use to delete multiple objects from bucket using single request.

Request Syntax -

```
POST /?delete HTTP/1.1
Host: Bucket.s3.accessscluster:8143
<?xml version="1.0" encoding="UTF-8"?>
<Delete xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
   <Object>
      <Key>string</Key>
      <VersionId>string</VersionId>
   </Object>
   ...
   <Quiet>boolean</Quiet>
</Delete>
```

Request Body -

Delete

Root level tag for the Delete parameters.
Required: Yes

Object

The Objects to be deleted. It contain Key and VersionId. Required: Yes

Quiet

To enable quiet mode for the request.

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<DeleteResult>
   <Deleted>
      <DeleteMarker>boolean</DeleteMarker>
      <DeleteMarkerVersionId>string</DeleteMarkerVersionId>
      <Key>string</Key>
      <VersionId>string</VersionId>
   </Deleted>
   ...
   <Error>
      <Code>string</Code>
      <Key>string</Key>
      <Message>string</Message>
      <VersionId>string</VersionId>
   </Error>
   ...
</DeleteResult>
```

Response Body -

DeleteResult

Root level tag for the DeleteResult parameters.
Required: Yes

Deleted

Information of objects which are successfully deleted.

DeleteMarker

Specifies if deleted object was a delete marker or not.

DeleteMarkerVersionId

Specifies versionId of the deleted delete marker.

Key

Object name of the deleted object.

VersionId

VersionId of the deleted object.

Error

Information of the objects which failed to be deleted.

Code

Error code of the error occurred while deleting object.

Key

Object name or delete marker name.

Message

Error message.

VersionId

VersionId of the object or delete marker for which error occurred.

Security Considerations - Caller should have write permissions on the bucket to delete an object.

Possible Error response -

- **Success** HTTP status code 200.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Get Object

Description - The Get Object API retrieves objects from an Access S3 bucket.

- Simple Get object operation supports object download of up to 100 MB in size. For larger object download, use the range-based Get Object API.

Request Syntax -

```
GET /Key+?versionId=VersionId HTTP/1.1
Host: Bucket.s3.accesscluster:8143
If-Match: IfMatch
If-Modified-Since: IfModifiedSince
If-None-Match: IfNoneMatch
If-Unmodified-Since: IfUnmodifiedSince
Range: Range
```

Request Parameters -

Bucket

Bucket name containing the object.
Required: Yes
Type: String

Key

Name of the object to read.
Required: Yes
Type: String

versionId

Version Id of object.
Type: String

Request Headers -

If-Match

Return the object only if entity tag (ETag) of object matches with specified ETag.
Type: String

If-Modified-Since

Return the object only if it has been modified since mentioned time.
Type: Timestamp

If-None-Match

Return the object only if entity tag (ETag) of object does not matches with specified ETag.
Type: String

If-Unmodified-Since

Return the object only if its not modified since mentioned time.
Type: String

Range

Returns specified range bytes of object.
Type: Integer

Response Syntax -

```
HTTP/1.1 200
x-amz-delete-marker: DeleteMarker
accept-ranges: AcceptRanges
Last-Modified: LastModified
Content-Length: ContentLength
ETag: ETag
x-amz-version-id: VersionId
Content-Range: ContentRange
Content-Type: ContentType
x-amz-storage-class: StorageClass
x-amz-object-lock-mode: ObjectLockMode
x-amz-object-lock-retain-until-date: ObjectLockRetainUntilDate

Body
```

Response Headers -

x-amz-delete-marker

Specifies the object return is a delete marker or not. If the object is not a delete marker, then this header does not get added in response.

Last-Modified

It is the object last modified time.

Content-Length

Returned body size in bytes.

ETag

Specifies md5sum of returned object.

x-amz-version-id

Specifies version Id of returned object.

Content-Range

The range of object returned in response.

x-amz-storage-class

Specifies storage class of returned object.

x-amz-object-lock-mode

Specifies the object lock mode of returned object. Header gets added only if Object Lock is set on object.

x-amz-object-lock-retain-until-date

Specifies the date and time when object's Object Lock will expire. Header gets added only if Object Lock is set on object.

Security Considerations - Caller should have read permissions on the object for the Get object API.

Possible Error response -

- **Success** HTTP status code 200.
- **InvalidArgument** Invalid Argument. Invalid version id specified. HTTP status code 400.
- **EntityTooLarge** Your proposed upload exceeds the maximum allowed object size. HTTP status code 400.
- **InvalidObjectState** The operation is not valid for the object's storage class. HTTP status code 403.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchKey** The specified key does not exist. HTTP status code 404.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Get Object Acl

Description - Returns access control list (ACL) of an object. For information on ACL, refer to ACL.

Request Syntax -

```
GET /{Key+}?acl&versionId=VersionId HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameter -

Bucket

The name of the bucket containing object.
Required: Yes
Type: String

Key

The name of the object for which you want to get ACL.
Required: Yes
Type: String

versionId

The version ID of the object.
Type: String

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy>
   <Owner>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
   </Owner>
   <AccessControlList>
      <Grant>
         <Grantee>
            <DisplayName>string</DisplayName>
            <ID>string</ID>
            <xsi:type>string</xsi:type>
            <URI>string</URI>
         </Grantee>
         <Permission>string</Permission>
      </Grant>
   </AccessControlList>
</AccessControlPolicy>
```

Response Body -

AccessControlPolicy

Root level tag for the AccessControlPolicy parameters.
Required: Yes

Grants

A list of grants. While specifying 'CanonicalUser', the user name of grantee and UID of user id needs to specified.
Required: No

Owner

Container for the bucket owner's display name and ID. The 'DisplayName' should be the username of the grantee which should resolve to user id in server's authentication method (passwd/NIS/LDAP/AD).
Required: No

Security Considerations - User should have read-acp permission on the bucket to perform Get object ACL.

Possible Error response -

- **Success** HTTP status code 200.
- **InvalidArgument** Invalid Argument. Invalid version id specified. HTTP status code 400.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchKey** The specified key does not exist. HTTP status code 404.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Get Object Retention

Description - The Get Object Retention API retrieves an object's retention settings.

Request Syntax -

```
GET /{Key+}?retention&versionId=VersionId HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameter -

Bucket

The name of the bucket containing object.
Required: Yes
Type: String

Key

The name of the object for which you want to get Object Lock Configuration.
Required: Yes
Type: String

versionId

The version ID of the Object.
Type: String

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<Retention>
   <Mode>string</Mode>
   <RetainUntilDate>timestamp</RetainUntilDate>
</Retention>
```

Response Body -:

Retention

Root level tag for the Retention parameters.
Required: Yes

Mode

Specifies the retention mode of the object. Values: GOVERNANCE|COMPLIANCE

RetainUntilDate

The date and time at which Object Lock retention will expire. Type: timestamp

Security Considerations - User should have read permission on the source object to perform Get object retention operation.

Possible Error response -

- **Success** HTTP status code 200.
- **InvalidRequest** Bucket is missing Object Lock Configuration. HTTP status code 400.
- **InvalidArgument** Invalid Argument. Invalid version id specified. HTTP status code 400.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchKey** The specified key does not exist. HTTP status code 404.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Head Object

Description - The Head Object API retrieves metadata from an object without returning the object itself. This operation is used when you are only interested in an object's metadata.

Request Syntax -

```
HEAD /Key+?versionId=VersionId HTTP/1.1
Host: Bucket.s3.accesscluster:8143
If-Match: IfMatch
If-Modified-Since: IfModifiedSince
If-None-Match: IfNoneMatch
If-Unmodified-Since: IfUnmodifiedSince
```

Request Parameters -

Bucket

Bucket name containing the object.
Required: Yes
Type: String

Key

Name of the object to read.
Required: Yes
Type: String

versionId

Version Id of object.
Type: String

Request Headers -

If-Match

Return the object only if entity tag (ETag) of object matches with specified ETag.
Type: String

If-Modified-Since

Return the object only if it has been modified since mentioned time.
Type: String

If-None-Match

Return the object only if entity tag (ETag) of object does not matches with specified ETag.
Type: String

If-Unmodified-Since

Return the object only if its not modified since mentioned time.
Type: String

Response Syntax -

```
HTTP/1.1 200
x-amz-delete-marker: DeleteMarker
accept-ranges: AcceptRanges
Last-Modified: LastModified
Content-Length: ContentLength
ETag: ETag
x-amz-version-id: VersionId
Content-Range: ContentRange
Content-Type: ContentType
x-amz-storage-class: StorageClass
x-amz-object-lock-mode: ObjectLockMode
x-amz-object-lock-retain-until-date: ObjectLockRetainUntilDate
```

Response Headers -

x-amz-delete-marker

Specifies whether the object returned is a delete marker or not. If the object is not a delete marker, then this header does not get added in response.

Last-Modified

It is the object last modified time.

Content-Length

Returned body size in bytes.

ETag

Specifies md5sum of returned object.

x-amz-version-id

Specifies version Id of returned object.

Content-Range

The range of object returned in response.

x-amz-storage-class

Specifies storage class of returned object.

x-amz-object-lock-mode

Specifies the object lock mode of returned object. Header gets added only if Object Lock is set on object.

x-amz-object-lock-retain-until-date

Specifies the date and time when object's Object Lock will expire. Header gets added only if Object Lock is set on object.

Security Considerations - Caller should have read permission on object for head object API.

Possible Error response -

- **Success** HTTP status code 200.
- **InvalidArgument** Invalid Argument. Invalid version id specified. HTTP status code 400.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchKey** The specified key does not exist. HTTP status code 404.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

List Parts

Description - The List Part Upload API lists in progress multipart uploads for a specific upload ID. It shows the parts which were initiated but are not completed or aborted yet.

Request Syntax -

```
GET /Key+?max-parts=MaxParts&part-number-marker=PartNumberMarker&uploadId=UploadId
HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameters -

Bucket

The name of bucket for which Initiate Multipart Upload was run.
Required: Yes
Type: String

Key

The name of the object for which multipart upload was initiated.
Required: Yes
Type: String

max-parts

Maximum number of parts returned in response.
Type: Integer

part-number-marker

Part number from where listing will start.
Type: String

uploadId

Upload ID of multipart upload.
Required: Yes
Type: String
Valid range: 1-10000

Response Syntax -

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListPartsResult>
   <Bucket>string</Bucket>
   <Key>string</Key>
   <UploadId>string</UploadId>
   <PartNumberMarker>integer</PartNumberMarker>
   <NextPartNumberMarker>integer</NextPartNumberMarker>
   <MaxParts>integer</MaxParts>
   <IsTruncated>boolean</IsTruncated>
   <Part>
      <ETag>string</ETag>
      <LastModified>timestamp</LastModified>
      <PartNumber>integer</PartNumber>
      <Size>integer</Size>
   </Part>
   ...
   <Owner>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
   </Owner>
</ListPartsResult>
```

Response Body -

ListPartsResult

Root level tag for the ListPartsResult parameters.
Required: Yes

Bucket

The name of the bucket.

Key

The name of key for which multipart upload was started.

UploadId

ID for the initiated multipart upload.

PartNumberMarker

Marks the last part in response, in case the list was truncated.

NextPartNumberMarker

Marks the next part in response, in case the list was truncated.

MaxParts

The maximum number of parts that can be returned in the response body.

IsTruncated

A flag indicating whether all the results satisfying the search criteria were returned by Access

S3 or not.

Part

Contains the elements related to a particular part.
Type: Array

Owner

Contains elements related to object owner.
Type: Array

Security Considerations - Caller should have read permission on bucket for List Part API.

Possible Error response -

- **Success** HTTP status code 200.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Put Object

Description - The Put Object API adds an object to a bucket. If bucket is versioning enabled, then Put Object API returns the VersionId of the object. The version ID returned by veritas Access should be opaque to application, Application should not make any assumption about version ID.

For information on ACL, refer to ACL.

Request Syntax -

```
PUT /Key+ HTTP/1.1
Host: Bucket.s3.accesscluster:8143
Content-Length: ContentLength
Content-MD5: ContentMD5
Content-Type: ContentType
x-amz-acl: ACL
x-amz-grant-full-control: GrantFullControl
x-amz-grant-read: GrantRead
x-amz-grant-read-acp: GrantReadACP
x-amz-grant-write-acp: GrantWriteACP
x-amz-storage-class: StorageClass
x-amz-object-lock-mode: ObjectLockMode
x-amz-object-lock-retain-until-date: ObjectLockRetainUntilDate

Body
```

Request Parameters -

Bucket

The name of bucket for which you want to PUT object.
Required: Yes

Type: String

Key

The name of the object which is being PUT.
Required: Yes
Type: String

Request Headers -

Content-Type

Describes the format of content.

x-amz-acl

The ACL to be applied on the object.
Type: String

x-amz-grant-full-control

Grant READ, READ_ACP and WRITE_ACP permission on the object.
Type: String

x-amz-grant-read

Grant READ permission on the object.
Type: String

x-amz-grant-read-acp

Grant READ_ACP permission on the object.
Type: String

x-amz-grant-write-acp

Grant WRITE_ACP permission on the object.
Type: String

x-amz-storage-class

Default value is STANDARD.
Type: String
Valid Values: STANDARD | REDUCED_REDUNDANCY | STANDARD_IA

x-amz-object-lock-mode

The Lock mode to be applied on object.
Type: String
Values: GOVERNANCE|COMPLIANCE

x-amz-object-lock-retain-until-date

Timestamp for object retention value.
Type: Timestamp

Response Syntax -

```
HTTP/1.1 200
ETag: ETag
x-amz-version-id: VersionId
```

Response Headers -

x-amz-version-id

The version-id of the object PUT in the bucket.

Security Considerations - User should have write permission on the bucket to perform PUT object.

Possible Error response -

- **Success** HTTP status code 200.
- **EntityTooLarge** The object size exceeded maximum allowed size. HTTP status code 400.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **MissingContentLength** Content-Length HTTP header missing in request. HTTP status code 411.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Put Object Acl

Description - The Put Object ACL API sets access control list permissions on an existing or new S3 object. For information on ACL, refer to ACL.

Request Syntax -

```
PUT /{Key+}?acl&versionId=VersionId HTTP/1.1
Host: Bucket.s3.accesscluster:8143
x-amz-acl: ACL
Content-MD5: ContentMD5
x-amz-grant-full-control: GrantFullControl
x-amz-grant-read: GrantRead
x-amz-grant-read-acp: GrantReadACP
x-amz-grant-write: GrantWrite
x-amz-grant-write-acp: GrantWriteACP
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy>
   <AccessControlList>
      <Grant>
         <Grantee>
            <DisplayName>string</DisplayName>
            <ID>string</ID>
            <xsi:type>string</xsi:type>
            <URI>string</URI>
         </Grantee>
         <Permission>string</Permission>
      </Grant>
   </AccessControlList>
   <Owner>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
   </Owner>
</AccessControlPolicy>
```

Request Parameter -

Bucket

Bucket name on which you want to set ACL.
Required: Yes
Type: String

Key

The name of the object.
Required: Yes
Type: String

versionId

The version ID of the key.
Type: String

Request Headers -

x-amz-acl

Set canned ACL on the bucket.
Type: String
Valid Values: private | public-read | public-read-write | authenticated-read

x-amz-grant-full-control

Allow read, write, read ACP and write ACP permissions on the bucket to grantee.
Type: String

x-amz-grant-read

Allow grantee to list objects in the bucket.
Type: String

x-amz-grant-read-acp

Allow grantee to read the bucket ACL.
Type: String

x-amz-grant-write

Allow grantee to put/delete objects.
Type: String

x-amz-grant-write-acp

Allow grantee to write the ACL for bucket.
Type: String

Request Body -

AccessControlPolicy

Root level tag for the AccessControlPolicy parameters.
Required: Yes

Grants

A list of grants. While specifying 'CanonicalUser', the user name of grantee and UID of user id needs to specified.
Required: No

Owner

Container for the bucket owner's display name and ID. The 'DisplayName' should be the username of the grantee which should resolve to user id in server's authentication method (passwd/NIS/LDAP/AD).
Required: No

Response Syntax -

```
HTTP/1.1 200
```

Security Considerations - User should have write-acp permission on the bucket to perform Put object ACL.

Possible Error response -

- **Success** HTTP status code 200.
- **BadDigest** The Content-MD5 you specified did not match what we received. HTTP status code

400.

- **InvalidRequest** Specifying both Canned ACLs and Header Grants is not allowed. HTTP status code 400.
- **UnexpectedContent** This request does not support content. HTTP status code 400.
- **InvalidArgument** Invalid group URI or Invalid user id or Max number of access control entries per ACL exceeded or Invalid VersionId specified. HTTP status code 400.
- **MalformedACLError** The XML you provided was not well-formed or did not validate against our published schema. HTTP status code 400.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Put Object Retention

Description - The Put Object Retention API places retention configuration on an object.

Request Syntax -

```
PUT /{Key+}?retention&versionId=VersionId HTTP/1.1
Host: Bucket.s3.accesscluster:8143
Content-MD5: ContentMD5
<?xml version="1.0" encoding="UTF-8"?>
<Retention>
   <Mode>string</Mode>
   <RetainUntilDate>timestamp</RetainUntilDate>
</Retention>
```

Request Parameters -

Bucket

The name of the bucket in which object is present.
Required: Yes
Type: String

Key

The name of the object on which retention needs to be applied.
Required: Yes
Type: String

versionId

The version ID of the key on which we need to apply retention.
Type: String

Request Body -

Retention

Root level tag for applying retention parameters.

Mode

The lock mode to be applied on the object.
Values: GOVERNANCE|COMPLIANCE

RetainUntilDate

Timestamp for retention time for object.

Response Syntax -

```
HTTP/1.1 200
```

Security Considerations - User should have write permission on the bucket to perform Put object retention.

Possible Error response -

- **Success** HTTP status code 200.
- **InvalidArgument** HTTP status code 400.
- **MalformedXML** The XML provided was not well formed. HTTP status code 400.
- **InvalidRequest** The request is invalid. HTTP status code 400.
- **AccessDenied** The retention value is less than current value. HTTP status code 403.
- **NoSuchKey** The key does not exist. HTTP status code 404.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **MethodNotAllowed** The method is not allowed for the resource. HTTP status code 405.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Upload part

Description - The Upload Part API uploads a part in a multipart upload. There is no minimum allowable part size for a multipart upload.

Request Syntax -

```
PUT /Key+?partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: Bucket.s3.accesscluster:8143
```

Request Parameters -

Bucket

The name of bucket for which multipart upload was run.
Required: Yes
Type: String

Key

The name of the object for which multipart upload was initiated.

Required: Yes
Type: String

partNumber

Number of part which is being uploaded.
Required: Yes
Type: String

uploadId

Upload ID of multipart upload.
Required: Yes
Type: String
Valid range: 1-10000

Response Syntax -

```
HTTP/1.1 200
```

Security Considerations - User should have write permission on the bucket to perform Put object.

Possible Error response -

- **Success** HTTP status code 200.
- **InvalidArgument** HTTP status code 400.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchUpload** The uploadId or Key might be invalid. HTTP status code 404.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Upload part copy

Description - The Upload part copy API uploads a part by copying data from an existing object as data source. You specify the data source by adding the request header x-amz-copy-source in your request and a byte range by adding the request header x-amz-copy-source-range in your request. There is no minimum allowable part size for a multipart upload.

Request Syntax -

```
PUT /Key+?partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: Bucket.s3.accesscluster:8143
x-amz-copy-source: CopySource
x-amz-copy-source-if-match: CopySourceIfMatch
x-amz-copy-source-if-modified-since: CopySourceIfModifiedSince
x-amz-copy-source-if-none-match: CopySourceIfNoneMatch
x-amz-copy-source-if-unmodified-since: CopySourceIfUnmodifiedSince
x-amz-copy-source-range: CopySourceRange
```

Request Headers -

x-amz-copy-source

Specify the data source to be copied.
Type: String

x-amz-copy-source-if-match

Copy source if Etag matches.
Type: String

x-amz-copy-source-if-modified-since

Copy source if it has been modified since specified time.
Type: Timestamp

x-amz-copy-source-if-none-match

Copy source if Etag does not match.
Type: String

x-amz-copy-source-if-unmodified-since

Copy source if it is unmodified since specified time.
Type: Timestamp

x-amz-copy-source-range

Byte range to be copied.
Type: Integer

Request Parameters -

Bucket

Name of the bucket for which multipart upload was run.
Required: Yes

Key

The name of the object for which multipart upload was initiated.
Required: Yes

partNumber

Number of part which is being copied.
Required: Yes

uploadId

Upload ID of multipart upload.
Required: Yes
Valid range: 1-10000

Response Syntax -

```
HTTP/1.1 200
x-amz-copy-source-version-id: CopySourceVersionId
```

Response Headers -

x-amz-copy-source-version-id

The version of source object that was copied.

Security Considerations - User should have read permission on the source object to perform upload part copy operation.

Possible Error response -

- **Success** HTTP status code 200.
- **InvalidArgument** HTTP status code 400.
- **InvalidRequest** HTTP status code 400.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchUpload** The uploadId or Key might be invalid. HTTP status code 404.
- **NoSuchKey** The specified key does not exist. HTTP status code 404.
- **NoSuchBucket** The specified bucket does not exist. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

Data Types supported by Veritas Object Access server

Timestamp

The timeStamp is the UTC time in ISO 8601 format (for example, 20220406T000000Z (Wed, 06 Apr 2022 00:00:00 GMT)).

Access Control Lists (ACL) in Veritas Object Access server

The access control list enables users to manage and share buckets or objects in the Veritas Object Access server. Every bucket and Object has an associated ACL. This ACL defines which user/group will get what type of access for given resources.

What resource I can grant permissions

- Bucket
- Objects in buckets

What type of Permissions I can grant

- **Read**: This permission allows user to read content of resource.
- **Write**: This permission allows user to write new/delete existing content of resource.
- **Read ACP**: This permission allows user to read contents of ACL set on resource.
- **Write ACP**: This permission allows user to edit contents of ACL set on resource.
- **Full Control**: This permission allows users to perform all above operations.

To whom I can grant permissions

- Users (By using Canonical user ID)
- Predefined Groups (By using group URI), Following predefined groups are supported:
  - Authenticated Users group - This group is represented by URI, http://acs.amazonaws.com/groups/global/AuthenticatedUsers. This group represents all users who are requesting a resource or action by using a valid access and secret key. The request must be authenticated or signed.
  - All Users group - This group is represented by URI, http://acs.amazonaws.com/groups/global/AllUsers All non-signed/authenticated requests are considered as an anonymous request. To get access for an anonymous request, the resource must have required permissions for AllUsers group. **Veritas Object Access server maps 'all user group' to user id 'nobody'**.
  - Log Delivery group - This group is represented by URI, http://acs.amazonaws.com/groups/s3/LogDelivery Permission set of this group will be ignored for now, Since Veritas Object Access server do not support bucket logging.

How permissions work on bucket

- **Read**: This permission allows caller to list objects in bucket and Head bucket.
- **Write**: This permission allows caller create/delete objects in bucket.
- **Read-ACP**: This permission allows caller read bucket ACL by using Get Bucket ACL API.
- **Write-ACP**: This permission allows caller set new set of ACLs on bucket.

- **Full Control**: This permission allows caller to list object, list ACL, create/delete objects and set new ACLs on bucket.

How permissions work on object

- **Read**: This permission allows caller to download (GET + HEAD) object.
- **Write**: This permission is ignored for object.
- **Read-ACP**: This permission allows caller read object ACL by using Get Object ACL API.
- **Write-ACP**: This permission allows caller set new set of ACLs on object.
- **Full Control**: This permission allows caller to GET/HEAD object, list ACL and set new ACLs on object.

How to set permissions on resources

There are 3 ways to set permissions on resource. You can use any of either three ways to set permission on new resource or existing resources. The resource can be object or it can be bucket.

- Setting ACL using request header x-amz-acl i.e. "Canned ACL"
- Setting ACL using Grant header
- Setting ACL using complete details in request body.

The section below explains each approach in more detail.

Setting ACL using request header x-amz-acl i.e. "Canned ACL"

A user can set an ACL on a resource by using the request header x-amz-acl. This request header covers a set of permissions. The following are the list of the Canned ACLs supported by the Veritas Object Access Server. You can use this header in the create bucket/object call.

- private (Applies to:Bucket and Object) With this ACL, only the owner has full control. If there is no ACL set on the object or if the object is created by some other protocol, this default ACL will be set by the server.
- public-read (*Applies to: Bucket and Object*) The resource owner has full control and S3 All users group has read access.
- public-read-write (*Applies to: Bucket and Object*) The resource owner has full control and S3 All users group has read-write access. Setting this ACL on a bucket is not recommended.
- aws-exec-read: This ACL is not supported by the Veritas Object Access Server and will be ignored.
- authenticated-read (*Applies to: Bucket and Object*): The resource owner has full control and S3 AuthenticatedUsers group has read access.
- bucket-owner-read (*Applies to: Object*): The resource owner has full control and the bucket owner has read access.
- bucket-owner-full-control (*Applies to: Object*) The resource owner has full control and the bucket owner has full control.

- log-delivery-write: This ACL is not supported by the Veritas Object Access Server and will be ignored.

Setting ACL using Grant headers

User can set ACL in PUT request by using the following grant headers. These headers are supported with PUT Object, create bucket API.

- x-amz-grant-read
- x-amz-grant-write
- x-amz-grant-read-acp
- x-amz-grant-write-acp
- x-amz-grant-full-control

The Grantee container element contains

DisplayName

The user name.
Type: String

ID

ID of the user.
Type: String

Type

The type of the grantee.
Type: String

URI

Grantee Group URI.
Type: String

Following are examples of grant header, x-amz-grant-write: uri="url",id="123"

The valid input for uri is, "http://acs.amazonaws.com/groups/global/AuthenticatedUsers", "http://acs.amazonaws.com/groups/global/AllUsers", "http://acs.amazonaws.com/groups/s3/LogDelivery"

The valid input for id is number which represents UID of grantee. The users email address is not supported with Grant headers.

Setting ACL using complete details in request body. The ACL can be set by sending data in request body itself. User can send correct xml in PutObjectACL and PutBucketACL reqeust.

Presigned URLs

All Buckets and Objects are by default private. However, you can share object using presigned URL. Presigned URL is generated using credentials of Bucket or Object owner and can be shared with intended customers/users. When you create presigned URL, you associate it with specific action. When the URL reaches its expiration date, it will no longer work.
Veritas Object Access server supports presigned URLs generated using only Signature version 4. Signature version 2 is not supported.

- **Sharing Object**: By creating a presigned URL, object owners may share objects with others using their own security credentials. And provide permission to customers/users to download objects for a limited time.
  In addition to your security credentials, you must specify a bucket name, an object key, an HTTP method (GET to download the object), and an expiration date and time when you create the presigned URL for your object.
- **Uploading Object**: If you want your customer to be able to upload objects of your bucket without requiring AWS credentials or permissions, you can use the presigned URLs.
  In addition to your security credentials, you must specify a bucket name, an object key, an HTTP method (PUT to uploading objects), and an expiration date and time when you create the presigned URL for your object.
- **Deleting an Object**: If you want your customer to be able to delete object of your bucket without requiring AWS credentials or permissions, you can use the presigned URLs.
  In addition to your security credentials, you must specify a bucket name, an object key, an HTTP method (Delete to delete the objects), and an expiration date and time when you create the presigned URL for your object.

Example of Presigned URL -

```
http://s3.accesscluster:8143/bucket/object?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=NDljYzcwOGVmYWNiMzk%2F20220%2Fus-east-2%2Fs3%2Faws4_request&X-Amz-
Date=20220330T083002Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-
Amzgnature=76901e5a7a28bfefde252db166e6f203a0b5c3a64a38e491728b55318e824dd9
```