WHITE PAPER

Titan T6000

PowerScale: OneFS S3 API Guide

CRS TITAN

POWERED BY DELL Technologies

Technical White Paper

Dell EMC PowerScale: OneFS S3 API Guide

Abstract

This document provides technical details about Dell EMC™ PowerScale™ OneFS™ S3 API compatibility to help incorporate applications with OneFS S3. This document also includes the supported quest parameter details for each OneFS S3 API.

October 2021

H18293.2

Revisions

Date	Description
June 2020	Initial release
May 2021	Update with OneFS S3 ETag implementation
October 2021	Update with OneFS 9.3.0 S3 enhancements: MPU, DeleteObjects, Chunked Upload and non-slash delimiter support for ListObjects/ ListObjectsV2

Acknowledgments

Author: Lieven Lin

Support: Miles Ohlrich, Takafumi Yonekura

The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Table of contents

Executive summary

Amazon Simple Storage Service (Amazon S3) provides a set of APIs that enable applications to access object storage. With the popularity of S3 object storage, more storage vendors are implementing their own object-storage engines with S3-compatible APIs. However, some Amazon S3 APIs are designed specifically for Amazon Web Services (AWS), with example services including life cycle management, charging, and analytics. Third-party S3-compatible storage usually implements a subset of Amazon S3 official APIs to enable object access.

Starting with Dell EMC™ OneFS™ version 9.0, Dell EMC PowerScale™ introduces the capability of data access by exposing the compatible S3 APIs. To facilitate administration and application deployment with the OneFS S3 implementation, this document shows how OneFS S3 API compatibility compares with the latest version of the Amazon S3 API.

1 OneFS S3 API compatibility overview

The Amazon S3 API was originally developed as the data-access interface of Amazon S3. As applications were developed using the S3 API, it became a common standard for object storage. This document refers to the S3 API for object storage as the S3 protocol. This provides a consistent nomenclature along with other NAS protocols regarding the OneFS file service.

Starting with OneFS 9.0, PowerScale OneFS supports the Amazon S3 protocol with OneFS S3, an object-storage interface that is compatible with the Amazon S3 API. OneFS S3 enables access to file-based data that is stored on OneFS clusters as objects.

This section compares OneFS S3 API compatibility with the Amazon S3 official APIs. OneFS supports both path-style requests and virtual hosted-style requests. It uses port 9020 for HTTP and port 9021 for HTTPS by default.

Table 1 lists the latest versions of AWS S3 APIs (sourced from Amazon S3 API documentation), and shows the compatibility with OneFS S3. See section 2 and section 3 for details about each OneFS supported S3 API. The APIs are sorted alphabetically as in the Amazon S3 API documentation.

Table 1 OneFS AWS S3 API compatibility

API names	Apply to	Supported by OneFS
AbortMultipartUpload	Object	Yes
CompleteMultipartUpload	Object	Yes
CopyObject	Object	Yes
CreateBucket	Bucket	Yes
CreateMultipartUpload	Object	Yes
DeleteBucket	Bucket	Yes
DeleteBucketAnalyticsConfiguration	Bucket	No
DeleteBucketCors	Bucket	No
DeleteBucketEncryption	Bucket	No
DeleteBucketInventoryConfiguration	Bucket	No
DeleteBucketLifecycle	Bucket	No
DeleteBucketMetricsConfiguration	Bucket	No
DeleteBucketPolicy	Bucket	No
DeleteBucketReplication	Bucket	No
DeleteBucketTagging	Bucket	No
DeleteBucketWebsite	Bucket	No
DeleteObject	Object	Yes
DeleteObjects	Object	Yes
DeleteObjectTagging	Object	No
DeletePublicAccessBlock	Bucket	No
GetBucketAccelerateConfiguration	Bucket	No

API names	Apply to	Supported by OneFS
GetBucketAcl	Bucket	Yes
GetBucketAnalyticsConfiguration	Bucket	No
GetBucketCors	Bucket	No
GetBucketEncryption	Bucket	No
GetBucketInventoryConfiguration	Bucket	No
GetBucketLifecycle	Bucket	No
GetBucketLifecycleConfiguration	Bucket	No
GetBucketLocation	Bucket	Yes
GetBucketLogging	Bucket	No
GetBucketMetricsConfiguration	Bucket	No
GetBucketNotification	Bucket	No
GetBucketNotificationConfiguration	Bucket	No
GetBucketPolicy	Bucket	No
GetBucketPolicyStatus	Bucket	No
GetBucketReplication	Bucket	No
GetBucketRequestPayment	Bucket	No
GetBucketTagging	Bucket	No
GetBucketVersioning	Bucket	No
GetBucketWebsite	Bucket	No
GetObject	Object	Yes
GetObjectAcl	Object	Yes
GetObjectLegalHold	Object	No
GetObjectLockConfiguration	Object	No
GetObjectRetention	Object	No
GetObjectTagging	Object	No
GetObjectTorrent	Object	No
GetPublicAccessBlock	Bucket	No
HeadBucket	Bucket	Yes
HeadObject	Object	Yes
ListBucketAnalyticsConfigurations	Bucket	No
ListBucketInventoryConfigurations	Bucket	No
ListBucketMetricsConfigurations	Bucket	No
ListBuckets	Bucket	Yes
ListMultipartUploads	Bucket	Yes
ListObjects	Bucket	Yes

API names	Apply to	Supported by OneFS
ListObjectsV2	Bucket	Yes
ListObjectVersions	Bucket	No
ListParts	Object	Yes
PutBucketAccelerateConfiguration	Bucket	No
PutBucketAcl	Bucket	Yes
PutBucketAnalyticsConfiguration	Bucket	No
PutBucketCors	Bucket	No
PutBucketEncryption	Bucket	No
PutBucketInventoryConfiguration	Bucket	No
PutBucketLifecycle	Bucket	No
PutBucketLifecycleConfiguration	Bucket	No
PutBucketLogging	Bucket	No
PutBucketMetricsConfiguration	Bucket	No
PutBucketNotification	Bucket	No
PutBucketNotificationConfiguration	Bucket	No
PutBucketPolicy	Bucket	No
PutBucketReplication	Bucket	No
PutBucketRequestPayment	Bucket	No
PutBucketTagging	Bucket	No
PutBucketVersioning	Bucket	No
PutBucketWebsite	Bucket	No
PutObject	Object	Yes
PutObjectAcl	Object	Yes
PutObjectLegalHold	Object	No
PutObjectLockConfiguration	Object	No
PutObjectRetention	Object	No
PutObjectTagging	Object	No
PutPublicAccessBlock	Bucket	No
RestoreObject	Object	No
SelectObjectContent	Object	No
UploadPart	Object	Yes
UploadPartCopy	Object	Yes

2 OneFS S3 supported bucket APIs

Many request parameters of Amazon S3 APIs are designed for Amazon S3 features, and OneFS does not support these parameters in its supported APIs. This section introduces the request parameters of bucket S3 APIs that are supported by OneFS. For more information about request and response examples, see the documentation AWS S3 API Reference. OneFS supports the following bucket APIs:

- CreateBucket
- DeleteBucket
- GetBucketAcl
- GetBucketLocation
- HeadBucket
- ListBuckets
- ListMultipartUploads
- ListObjects
- ListObjectsV2
- PutBucketAcl

2.1 CreateBucket

Table 2 shows the details about the OneFS S3 CreateBucket API request parameters and request body. If not specified, all unsupported request parameters and the request body are silently ignored by OneFS without an error code being returned to client.

Table 2 OneFS S3 CreateBucket API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the bucket to create. Bucket names must consist of characters, including lowercase letters (a-z), numbers (0-9), or dashes (-). Bucket names must start or end with a lowercase letter (a-z) or number (0-9). Bucket names must be 3–63 characters in length.
x-isi-path	Yes	This is an optional OneFS extended header where a OneFS directory can be specified as the bucket path. OneFS creates the directory implicitly if the directory does not exist. See appendix A.1 for an example.
x-amz-acl	Yes	This is the canned ACL to apply to the bucket. Valid values: private, public-read, public-read-write, authenticated-read
x-amz-bucket-object-lock-enabled	No	N/A
x-amz-grant-full-control	Yes	This allows the grantee the read, write, read ACP, and write ACP permissions on the bucket.

Request parameters/body	Supported	Description
x-amz-grant-read	Yes	This allows the grantee to list the objects in the bucket.
x-amz-grant-read-acp	Yes	This allows the grantee to read the bucket ACL.
x-amz-grant-write	Yes	This allows the grantee to create, overwrite, and delete any object in the bucket.
x-amz-grant-write-acp	Yes	This allows the grantee to write the ACL for the applicable bucket.
CreateBucketConfiguration	No	N/A
LocationConstraint	No	N/A. OneFS sets the bucket location to empty string.

Note: Amazon S3 allows specifying a grantee using the following headers: email address, id, uri in the x-amz-grant-read, x-amz-grant-write, x-amz-grant-read-acp, x-amz-grant-write-acp, and x-amz-grant-full-control. OneFS does not support the email address to specify a grantee, and 501 NotImplemented code is returned.

2.2 DeleteBucket

Table 3 shows the details about the OneFS S3 DeleteBucket API request parameters and request body. When a bucket is deleted, OneFS only removes the bucket information while preserving the data under the bucket. If not specified, all unsupported request parameters and the request body are silently ignored by OneFS without an error code being returned to client.

Table 3 OneFS S3 DeleteBucket API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This specifies the bucket being deleted.
x-isi-force-delete	Yes	This is an optional OneFS extended header. By default, all objects (including all OneFS files and directories) in the bucket must be deleted before the bucket itself can be deleted. With the x-isi-force-delete extension, users can delete the bucket while preserving the data under the bucket. See appendix A.2 for an example.

2.3 GetBucketAcl

Table 4 shows the details about the OneFS S3 GetBucketAcl API request parameters and request body. If not specified, all unsupported request parameters and request body are silently ignored by OneFS without an error code being returned to client.

Table 4 OneFS S3 GetBucketAcl API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This specifies the S3 bucket whose ACL is being requested.

2.4 GetBucketLocation

Table 5 shows the details about the OneFS S3 GetBucketLocation API request parameters and request body. If not specified, all unsupported request parameters and request body are silently ignored by OneFS without an error code being returned to client.

Table 5 OneFS S3 GetBucketLocation API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the bucket location being requested. OneFS always returns an empty string.

2.5 HeadBucket

This API determines if a bucket exists and if permission is granted to access it. Table 6 shows the details about the OneFS S3 HeadBucket API request parameters and request body.

Table 6 OneFS S3 HeadBucket API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the bucket.

2.6 ListBuckets

This API returns a list of all buckets that are owned by the authenticated sender of the request. There are no request parameters and request body required for this API.

2.7 ListMultipartUploads

This API lists in-progress multipart uploads of a OneFS bucket. Table 7 shows the details about the OneFS S3 ListMultipartUploads API request parameters and request body. If not specified, all unsupported request parameters and request body are silently ignored by OneFS without an error code being returned to client.

Table 7 OneFS S3 ListMultipartUploads API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the bucket to which the multipart upload is initiated.
delimiter	No	N/A
encoding-type	No	N/A
key-marker	No	N/A
max-uploads	Yes	This sets the maximum number of multipart uploads in the response, ranging from 1–1,000.
prefix	Partial support	This returns entries that match with given prefix.
upload-id-marker	Yes	This specifies the multipart upload after which the listing should begin. When a response is truncated, set this parameter value to the value of NextUploadIdMarker in the response to continue listing the rest of multipart uploads.

2.8 ListObjects

This API returns some or all (up to 1,000) of the objects in a bucket. Table 8 shows the details about the OneFS S3 ListObjects API request parameters and request body. If not specified, all unsupported request parameters and request body are silently ignored by OneFS without an error code being returned to client.

Table 8 OneFS S3 ListObjects API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the bucket that contains the objects.
delimiter	Yes	A delimiter is a character used to group keys. Starting from OneFS 9.3.0, OneFS supports the non-slash characters as delimiter. Note that only delimiters ending with '/' or without '/' are supported.
encoding-type	No	N/A
marker	Yes	This specifies the key to start with when listing objects in a bucket. When a response is truncated, set this parameter value to the value of NextMarker, in the response to continue listing the rest of objects.

max-keys	Yes	This sets the maximum number of keys returned in the response, ranging from 1–1,000.
prefix	Yes	This limits the response to keys that begin with the specified prefix.
x-amz-request-payer	No	N/A

2.9 ListObjectsV2

This API returns some or all (up to 1,000) of the objects in a bucket. Table 9 shows the details about the OneFS S3 ListObjectsV2 API request parameters and request body. If not specified, all unsupported request parameters and request body are silently ignored by OneFS without an error code being returned to client.

Table 9 OneFS S3 ListObjectsV2 API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the bucket containing the objects.
continuation-token	Yes	This specifies that the list is being continued on this bucket with this token.
delimiter	Yes	The delimiter is a character that is used to group keys. Starting from OneFS 9.3.0, OneFS supports the non-slash characters as delimiter. Note that only delimiters ending with '/' or without '/' are supported.
encoding-type	No	N/A
fetch-owner	Yes	The owner field is not present in listV2 by default. If needing to return the owner field with each key in the result, set the fetch owner field to **true**.
max-keys	Yes	This sets the maximum number of keys that are returned in the response, ranging from 1–1,000.
prefix	Yes	This limits the response to keys that begin with the specified prefix.
start-after	Yes	This key specifies where OneFS starts the listing from. It can be any key in the bucket.
x-amz-request-payer	No	N/A

2.10 PutBucketAcl

This API sets the permissions on an existing bucket using access control lists (ACLs). Table 10 shows the details about the OneFS S3 PutBucketACL API request parameters and request body. If not specified, all unsupported request parameters and request body are silently ignored by OneFS without an error code being returned to client.

Table 10 OneFS S3 PutBucketACL API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the bucket containing the objects.
Content-MD5	Yes	This is the base64-encoded, 128-bit MD5 digest of the data.
x-amz-acl	Yes	This is the canned ACL to apply to the bucket. Valid values: private | public-read | public-read-write | authenticated-read
x-amz-grant-full-control	Yes	This allows the grantee the read, write, read ACP, and write ACP permissions on the bucket.
x-amz-grant-read	Yes	This allows the grantee to list the objects in the bucket.
x-amz-grant-read-acp	Yes	This allows the grantee to read the bucket ACL.
x-amz-grant-write	Yes	This allows the grantee to create, overwrite, and delete any object in the bucket.
x-amz-grant-write-acp	Yes	This allows the grantee to write the ACL for the applicable bucket.

Note: Amazon S3 allows specifying a grantee using the following headers: email address, id, uri in the x-amz-grant-read, x-amz-grant-write, x-amz-grant-read-acp, x-amz-grant-write-acp, and x-amz-grant-full-control. OneFS does not support an email address to specify a grantee, and a 501 NotImplemented code is returned.

3 OneFS S3 supported object APIs

Many request parameters of Amazon S3 APIs are designed for Amazon S3 features, and OneFS does not support these parameters in its supported APIs. This section introduces the request parameters of supported OneFS S3 APIs on objects. For the request and response examples, see the Amazon S3 API Reference. OneFS supports the following bucket APIs:

- AbortMultipartUpload
- CompleteMultipartUpload
- CopyObject
- CreateMultipartUpload
- DeleteObject
- DeleteObjects
- GetObject
- GetObjectAcl
- HeadObject
- ListParts
- PutObject
- PutObjectAcl
- UploadPart
- UploadPartCopy

3.1 AbortMultipartUpload

Table 11 shows the details about the OneFS S3 AbortMultipartUpload API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.

Table 11 OneFS S3 AbortMultipartUpload API request parameters

Request parameters/body	**Supported**	**Description**
Bucket	Yes	Required: Yes This is the bucket name to which the multipart upload was taking place.
Key	Yes	Required: Yes This is the key of the object for which the multipart upload was initiated.
uploadId	Yes	Required: Yes This is the upload ID that identifies the multipart upload.
x-amz-request-payer	No	N/A

3.2 CompleteMultipartUpload

This API completes a multipart upload by assembling previously uploaded parts. Table 12 shows the details about the OneFS S3 CompleteMultipartUpload API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.

Table 12 OneFS S3 CompleteMultipartUpload API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the bucket name to which the multipart upload was initiated.
Key	Yes	Required: Yes This is the key of the object for which the multipart upload was initiated.
uploadId	Yes	Required: Yes This is the upload ID that identifies the multipart upload.
x-amz-request-payer	No	N/A

3.3 CopyObject

This API creates a copy of an object that is already stored in OneFS. Table 13 shows the details about the OneFS S3 CopyObject API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.

Table 13 OneFS S3 CopyObject API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the destination bucket.
Cache-Control	No	N/A
Content-Disposition	No	N/A
Content-Encoding	No	N/A
Content-Language	No	N/A
Content-Type	No	N/A. OneFS ignores it and set to binary/octet-stream only.
Expires	No	N/A
Key	Yes	Required: Yes This is the key of the destination object.

Request parameters/body	Supported	Description
x-amz-acl	Yes	This is the canned ACL to apply to the object. OneFS does not support aws-exec-read and log-delivery-write. Valid values: private | public-read | public-read-write | authenticated-read | bucket-owner-read | bucket-owner-full-control
x-amz-copy-source	Yes	Required: Yes This is the name of the source bucket and key name of the source object
x-amz-copy-source-if-match	Yes	This copies the object if its ETag matches the specified tag.
x-amz-copy-source-if-modified-since	Yes	This copies the object if it has been modified since the specified time.
x-amz-copy-source-if-none-match	Yes	This copies the object if its ETag is different than the specified ETag.
x-amz-copy-source-if-unmodified-since	Yes	This copies the object if it has not been modified since the specified time.
x-amz-copy-source-server-side-encryption-customer-algorithm	No	N/A
x-amz-copy-source-server-side-encryption-customer-key	No	N/A
x-amz-copy-source-server-side-encryption-customer-key-MD5	No	N/A
x-amz-grant-full-control	Yes	This gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
x-amz-grant-read	Yes	This allows the grantee to read the object data and its metadata.
x-amz-grant-read-acp	Yes	This allows the grantee to read the object ACL.
x-amz-grant-write-acp	Yes	This allows the grantee to write the ACL for the applicable object.
x-amz-metadata-directive	Yes	This specifies whether the metadata is copied from the source object or replaced with metadata that is provided in the request. Valid values: COPY | REPLACE
x-amz-object-lock-legal-hold	No	N/A
x-amz-object-lock-mode	No	N/A
x-amz-object-lock-retain-until-date	No	N/A

Request parameters/body	Supported	Description
x-amz-request-payer	No	N/A
x-amz-server-side-encryption	No	N/A
x-amz-server-side-encryption-aws-kms-key-id	No	N/A
x-amz-server-side-encryption-context	No	N/A
x-amz-server-side-encryption-customer-algorithm	No	N/A
x-amz-server-side-encryption-customer-key	No	N/A
x-amz-server-side-encryption-customer-key-MD5	No	N/A
x-amz-storage-class	No	N/A. OneFS ignores it and sets it to STANDARD only.
x-amz-tagging	No	N/A
x-amz-tagging-directive	No	N/A
x-amz-website-redirect-location	No	N/A

Note: Amazon S3 allows specifying a grantee using the following headers: email address, id, uri in the x-amz-grant-read, x-amz-grant-read-acp, x-amz-grant-write-acp, and x-amz-grant-full-control. OneFS does not support an email address to specify a grantee, and a 501 NotImplemented code is returned.

3.4 CreateMultipartUpload

This API Initiates a multipart upload and returns an upload ID. Table 14 shows the details about the OneFS S3 CreateMultipartUpload API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.

Table 14 OneFS S3 CreateMultipartUpload API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the destination bucket.
Cache-Control	No	N/A
Content-Disposition	No	N/A
Content-Encoding	No	N/A
Content-Language	No	N/A

Request parameters/body	Supported	Description
Content-Type	No	N/A. OneFS ignores it and sets it to binary/octet-stream only.
Expires	No	N/A
Key	Yes	Required: Yes This is the key of the destination object.
x-amz-acl	Yes	This is the standard ACL to apply to the object. OneFS does not support aws-exec-read and log-delivery-write. Valid values: private | public-read | public-read-write | authenticated-read | bucket-owner-read | bucket-owner-full-control
x-amz-grant-full-control	Yes	This gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
x-amz-grant-read	Yes	This allows the grantee to read the object data and its metadata.
x-amz-grant-read-acp	Yes	This allows the grantee to read the object ACL.
x-amz-grant-write-acp	Yes	This allows the grantee to write the ACL for the applicable object.
x-amz-object-lock-legal-hold	No	N/A
x-amz-object-lock-mode	No	N/A
x-amz-object-lock-retain-until-date	No	N/A
x-amz-request-payer	No	N/A
x-amz-server-side-encryption	No	N/A
x-amz-server-side-encryption-aws-kms-key-id	No	N/A
x-amz-server-side-encryption-context	No	N/A
x-amz-server-side-encryption-customer-algorithm	No	N/A
x-amz-server-side-encryption-customer-key	No	N/A
x-amz-server-side-encryption-customer-key-MD5	No	N/A
x-amz-storage-class	No	N/A. OneFS ignores it and sets it to STANDARD only.
x-amz-tagging	No	N/A
x-amz-website-redirect-location	No	N/A

Note: Amazon S3 allows specifying a grantee using the following headers: email address, id, uri in the x-amz-grant-read, x-amz-grant-read-acp, x-amz-grant-write-acp, and x-amz-grant-full-control. OneFS does not support an email address to specify a grantee, and a 501 NotImplemented code is returned.

3.5 DeleteObject

This API deletes a specific object in a bucket. Table 15 shows the details about the OneFS S3 DeleteObject API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.

Table 15 OneFS S3 DeleteObject API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the bucket containing the object.
Key	Yes	Required: Yes This is the key name of the object to delete.
versionId	No	N/A
x-amz-bypass-governance-retention	No	N/A
x-amz-mfa	No	N/A
x-amz-request-payer	No	N/A

3.6 DeleteObjects

This API allows you to delete multiple objects from a bucket using a single HTTP request. Table 16Table 15 shows the details about the OneFS S3 DeleteObjects API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.

Table 16 OneFS S3 DeleteObjects API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the bucket containing the object.
x-amz-expected-bucket-owner	No	N/A
x-amz-bypass-governance-retention	No	N/A
x-amz-mfa	No	N/A
x-amz-request-payer	No	N/A

3.7 GetObject

This API retrieves objects from OneFS through the S3 protocol. If read permission is granted to the **nobody** user in OneFS, a client can retrieve the object without using an authorization header. Table 17 shows the details about the OneFS S3 GetObject API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.

Table 17 OneFS S3 GetObject API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the bucket name containing the object.
If-Match	Yes	This returns the object only if its ETag is the same as the one specified. Otherwise, it returns a 412 code (precondition failed).
If-Modified-Since	Yes	This returns the object only if it has been modified since the specified time. Otherwise, it returns a 304 code (not modified).
If-None-Match	Yes	This returns the object only if its ETag is different from the one specified. Otherwise, it returns a 304 code (not modified).
If-Unmodified-Since	Yes	This returns the object only if it has not been modified since the specified time. Otherwise, it returns a 412 code (precondition failed).
Key	Yes	Required: Yes This is the key of the object to get.

Request parameters/body	Supported	Description
partNumber	No	N/A. A 510 NotImplemented code is returned.
Range	Yes	This gets the specified range bytes of an object. See article RFC2616 for more information about the HTTP Range header.
response-cache-control	No	N/A
response-content-disposition	No	N/A
response-content-encoding	No	N/A
response-content-language	No	N/A
response-content-type	No	N/A. OneFS ignores it and sets it to binary/octet-stream only.
response-expires	No	N/A
versionId	No	N/A
x-amz-request-payer	No	N/A
x-amz-server-side-encryption-customer-algorithm	No	N/A
x-amz-server-side-encryption-customer-key	No	N/A
x-amz-server-side-encryption-customer-key-MD5	No	N/A

3.8 GetObjectAcl

This API returns the access control list (ACL) of an object. Table 18 shows the details about the OneFS S3 GetObjectAcl API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.

Table 18 OneFS S3 GetObjectAcl API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the bucket name that contains the object for which to get the ACL information.
Key	Yes	Required: Yes This is the key of the object for which to get the ACL information.
versionId	No	N/A
x-amz-request-payer	No	N/A

3.9 HeadObject

This API retrieves metadata from an object without returning the object data. Table 19 shows the details about the OneFS S3 HeadObject API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.

Table 19 OneFS S3 HeadObject API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the bucket name containing the object.
If-Match	Yes	This returns the object only if its ETag is the same as the one specified. Otherwise, this returns a 412 code (precondition failed).
If-Modified-Since	Yes	This returns the object only if it has been modified since the specified time. Otherwise, this returns a 304 code (not modified).
If-None-Match	Yes	This returns the object only if its ETag is different from the one specified. Otherwise, this returns return a 304 code (not modified).
If-Unmodified-Since	Yes	This returns the object only if it has not been modified since the specified time. Otherwise, this returns a 412 code (precondition failed).
Key	Yes	Required: Yes This is the key of the object to get.
partNumber	No	N/A. A 510 NotImplemented code is returned.
Range	Yes	This gets the specified range bytes of an object. See article RFC2616 for more information about the HTTP Range header.
versionId	No	N/A
x-amz-request-payer	No	N/A
x-amz-server-side-encryption-customer-algorithm	No	N/A
x-amz-server-side-encryption-customer-key	No	N/A
x-amz-server-side-encryption-customer-key-MD5	No	N/A

3.10 ListParts

This API lists the parts that have been uploaded for a specific multipart upload. Table 20 shows the details about the OneFS S3 ListParts API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.

Table 20 OneFS S3 ListParts API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the bucket to which the parts are being uploaded.
Key	Yes	Required: Yes This is the object key for which the multipart upload was initiated.
max-parts		This sets the maximum number of parts to return, from 1–1,000.
part-number-marker	Yes	This specifies the part after which listing should begin. Only parts with higher part numbers are listed. While the response is truncated, the value can be set to the value of NextPartNumberMarker from response.
uploadId	Yes	Required: Yes This is the upload ID that identifies the multipart upload whose parts are being listed.
x-amz-request-payer	No	N/A

3.11 PutObject

Table 21 shows the details about the OneFS S3 PutObject API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.

Table 21 OneFS S3 PutObject API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the destination bucket.
Cache-Control	No	N/A
Content-Disposition	No	N/A
Content-Encoding	No	N/A
Content-Language	No	N/A
Content-Length	Yes	This is the size of the body in bytes.
Content-MD5	Yes	This is the base64-encoded, 128-bit MD5 digest of the data. Please refer to section 3.15 for details.
Content-Type	No	OneFS ignores it and sets it to binary/octet-stream only by default.
Expires	No	N/A

Request parameters/body	Supported	Description
Key		Required: Yes This is the key of the destination object.
x-amz-acl	Yes	This is the standard ACL to apply to the object. OneFS does not support aws-exec-read and log-delivery-write. Valid values: private | public-read | public-read-write | authenticated-read | bucket-owner-read | bucket-owner-full-control
x-amz-grant-full-control	Yes	This gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
x-amz-grant-read	Yes	This allows the grantee to read the object data and its metadata.
x-amz-grant-read-acp	Yes	This allows the grantee to read the object ACL.
x-amz-grant-write-acp	Yes	This allows the grantee to write the ACL for the applicable object.
x-amz-object-lock-legal-hold	No	N/A
x-amz-object-lock-mode	No	N/A
x-amz-object-lock-retain-until-date	No	N/A
x-amz-request-payer	No	N/A
x-amz-server-side-encryption	No	N/A
x-amz-server-side-encryption-aws-kms-key-id	No	N/A
x-amz-server-side-encryption-context	No	N/A
x-amz-server-side-encryption-customer-algorithm	No	N/A
x-amz-server-side-encryption-customer-key	No	N/A
x-amz-server-side-encryption-customer-key-MD5	No	N/A
x-amz-storage-class	No	N/A. OneFS ignores it and sets it to STANDARD only.
x-amz-tagging	No	N/A
x-amz-website-redirect-location	No	N/A

Note: Amazon S3 allows specifying a grantee using the following headers: email address, id, uri in the x-amz-grant-read, x-amz-grant-read-acp, x-amz-grant-write-acp, and x-amz-grant-full-control. OneFS does not support an email address to specify a grantee, and a 501 NotImplemented code is returned.

3.12 PutObjectAcl

This API sets the permissions on an existing object using ACLs. Table 22 shows the details about the OneFS S3 PutObjectACL API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.

Table 22 OneFS S3 PutObjectACL API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the bucket containing the objects.
Content-MD5	Yes	This is the base64-encoded 128-bit MD5 digest of the data. Please refer to section 3.15 for details.
Key	Yes	Required: Yes This is the key name of the object to set the ACL.
versionId	No	N/A
x-amz-acl	Yes	This is the standard ACL to apply to the object. OneFS does not support aws-exec-read and log-delivery-write. Valid values: private | public-read | public-read-write | authenticated-read | bucket-owner-read | bucket-owner-full-control
x-amz-grant-full-control	Yes	This gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
x-amz-grant-read	Yes	This allows the grantee to read the object data and its metadata.
x-amz-grant-read-acp	Yes	This allows the grantee to read the object ACL.
x-amz-grant-write-acp	Yes	This allows the grantee to write the ACL for the applicable object.

Note: Amazon S3 allows specifying a grantee using the following headers: email address, id, uri in the x-amz-grant-readx-amz-grant-read-acp, x-amz-grant-write-acp, and x-amz-grant-full-control. OneFS does not support an email address to specify a grantee, and a 501 NotImplemented code is returned.

3.13 UploadPart

Table 23 This API uploads a part in a multipart upload. Each part must be at least 5 MB in size, except for the last part. The maximum size of each part is 5 GB. Table 23 shows the details about the OneFS S3 UploadPart API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.OneFS S3 UploadPart API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the bucket containing the objects.
Content-Length	Yes	This is the size of the body in bytes.
Content-MD5	Yes	This is the base64-encoded, 128-bit MD5 digest of the data. See section 3.15 for details.
Key	Yes	Required: Yes This is the object key for which the multipart upload is initiated.
partNumber	Yes	Required: Yes This is the part number of the part being uploaded. This is a positive integer between 1–10,000.
uploadId	Yes	Required: Yes This is the upload ID that identifies the multipart upload whose part is being uploaded.
x-amz-request-payer	No	N/A
x-amz-server-side-encryption-customer-algorithm	No	N/A
x-amz-server-side-encryption-customer-key	No	N/A
x-amz-server-side-encryption-customer-key-MD5	No	N/A

3.14 UploadPartCopy

This API uploads a part by copying data from an existing object as the data source. Each part must be at least 5 MB in size, except for the last part. The maximum size of each part is 5 GB. Table 24 shows the details about the OneFS S3 UploadPartCopy API request parameters and request body. If not specified, OneFS silently ignores all unsupported request parameters and the request body, without returning an error code to the client.

Table 24 OneFS S3 UploadPartCopy API request parameters

Request parameters/body	Supported	Description
Bucket	Yes	Required: Yes This is the name of the bucket containing the objects.
Key	Yes	Required: Yes This is the object key for which the multipart upload is initiated.
partNumber	Yes	Required: Yes This is the part number of the part being uploaded. This is a positive integer between 1– 10,000.
uploadId	Yes	Required: Yes This is the upload ID that identifies the multipart upload whose part is being copied.
x-amz-copy-source	Yes	Required: Yes This is the name of the source bucket and key name of the source object.
x-amz-copy-source-if-match	Yes	This copies the object if its ETag matches the specified tag.
x-amz-copy-source-if-modified-since	Yes	This copies the object if it has been modified since the specified time.
x-amz-copy-source-if-none-match	Yes	This copies the object if its ETag is different than the specified ETag.
x-amz-copy-source-if-unmodified-since	Yes	This copies the object if it has not been modified since the specified time.
x-amz-copy-source-range	Yes	This is the range of bytes to copy from the source object.
x-amz-copy-source-server-side-encryption-customer-algorithm	No	N/A
x-amz-copy-source-server-side-encryption-customer-key	No	N/A
x-amz-copy-source-server-side-encryption-customer-key-MD5	No	N/A

Request parameters/body	Supported	Description
x-amz-request-payer	No	N/A
x-amz-server-side-encryption-customer-algorithm	No	N/A
x-amz-server-side-encryption-customer-key	No	N/A
x-amz-server-side-encryption-customer-key-MD5	No	N/A

3.15 Additional features

3.15.1 OneFS S3 ETag

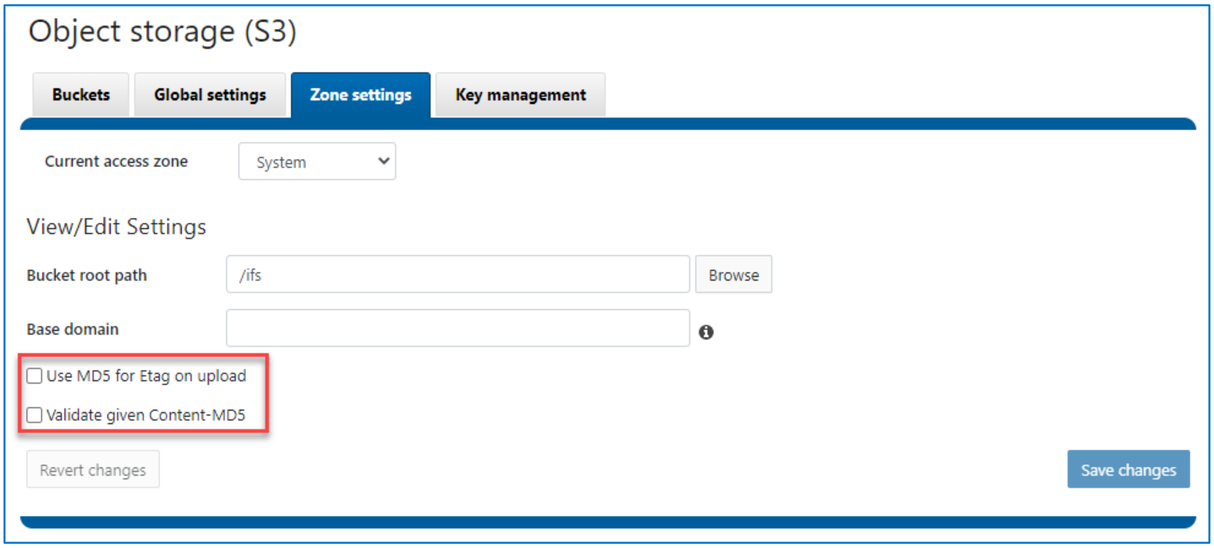
AWS S3 may use an MD5 Checksum as an ETag value for objects. This value may be specified in the HTTP Header “Content-MD5”. In OneFS 9.0 and OneFS 9.1, OneFS uses the MD5 value from client as an ETag directly instead of calculating it by itself. If the MD5 is not specified in client request, OneFS generates a unique string for that file as an ETag in response. This behavior is different from AWS S3.

Most S3 applications do not send the MD5 value in their requests, thus, OneFS generates a unique string for that file as an ETag in response. This behavior causes many issues with applications that rely on the ETag value. Therefore, starting from OneFS 9.2, OneFS introduces two new options to allow administrators to specify if the MD5 should be calculated and verified.

These two options are under the S3 zone settings. You can configure them using CLI **isi s3 settings zone modify --use-md5-for-etag=true/false --validate-content-md5=true/false** or WebUI, shown as Figure 1.

```
# isi s3 settings zone view
                   Root Path: /ifs
                 Base Domain:
           Object ACL Policy: replace
Bucket Directory Create Mode: 0777
            Use Md5 For Etag: No
```

```
          Validate Content Md5: No
```



Object storage (S3)

Current access zone: System

View/Edit Settings

Bucket root path: /ifs

Base domain:

□ Use MD5 for Etag on upload

□ Validate given Content-MD5

Figure 1 S3 ETag options

Table 25 shows the different behavior by using the two new options.

Table 25 OneFS S3 ETag behavior

	--use-md5-for-etag=false	**--use-md5-for-etag=true**
--validate-content-md5=false	This is the default value. If “Content-MD5” exists in client request, OneFS uses it directly as the ETag without validation and checking the BASE64 encoding format. If “Content-MD5” does not exist in client request, OneFS generates a unique string for that file as the ETag.	If “Content-MD5” exists in client request and its value is properly encoded as BASE64 format, OneFS uses it as the ETag without validation. If “Content-MD5” does not exist in client request, OneFS calculates the MD5 value as the ETag.
--validate-content-md5=true	If “Content-MD5” exists in client request and its value is properly encoded as BASE64 format, OneFS calculates the MD5 value and compare with the MD5 value from client request, if matched, uses it as the ETag. Otherwise, an error is returned to client. If “Content-MD5” does not exist in client request, OneFS generates a unique string for that file as the ETag.	If “Content-MD5” exists in client request, OneFS calculates the MD5 value and compare with the MD5 value from client request, if matched, uses it as the ETag. Otherwise, an error is returned to client. If “Content-MD5” does not exist in client request, OneFS calculates the MD5 value as the ETag.

Note: Objects created with a multipart upload request do not use MD5 value as ETag.

3.15.2 Presigned URLs

A presigned URL gives you access to the object identified in the URL, OneFS supports presigned URLs to allow users to access objects without needing credentials. Please refer to AWS S3 Presigned URLs for more details.

3.15.3 Chunked upload

There are two types of uploading options when authenticating requests using the **Authorization** header of AWS Signature Version 4:

- Transfer payload in a single chunk
- Transfer payload in multiple chunks (chunked upload)

Starting with OneFS 9.3.0, the chunked upload is introduced. With chunked upload, you can break up your payload into chunks. These can be fixed or variable-size chunks. By uploading data in chunks, you avoid reading the entire payload to calculate the signature. Instead, for the first chunk, you calculate a seed signature that uses only the request headers. The second chunk contains the signature for the first chunk, and each subsequent chunk contains the signature for the chunk that precedes it. At the end of the upload, you send a final chunk with 0 bytes of data that contains the signature of the last chunk of the payload. You can refer to AWS S3 Chunked Upload for more details.

4 OneFS S3 limitations

This section addresses the limitations of the OneFS S3 implementation. It is recommended to take these limitations into consideration when designing an application to work with OneFS S3.

Table 26 shows the details about OneFS S3 limitations.

Table 26 OneFS S3 limitations

Feature	Limitation
Object versioning	OneFS does not support Amazon S3 object versioning. It is recommended to use OneFS SnapshotIQ as an alternative method.
Object lock	OneFS does not support Amazon S3 object lock. It is recommended to use OneFS SmartLock as an alternative solution to meet compliance requirements.
Object key delimiter	Starting from OneFS 9.3.0, OneFS supports the following delimiter format: • Delimiter contains only one single slash(“/”) character by default. • Delimiter contains only one single slash(“/”) at the end, such as “abc/”. • Delimiter does not contain any slash(“/”), such as “abcd”. Therefore, “/abcd” and “ab/cd” are not valid delimiters.
Number of buckets	There are 1,000 buckets per user and 40,000 per cluster.
User-defined metadata	• The key of user-defined metadata consists of a maximum of 200 Bytes of UTF-8 encoded, case-sensitive, alphanumeric characters, the period (.), and the underscore (_) characters. • The value of user-defined metadata is not more than 1,024 bytes. • At most, 128 user-metadata are allowed for an object. • The total size of the HTTP header (including user-defined metadata and other headers) may not exceed 8 KB in OneFS. The total size of user-defined metadata is limited and varies depending on the actual HTTP request.
Bucket traversal	An S3 user must traverse or execute permissions on all elements of the path to reach an object in the OneFS directory hierarchy.
Object size	An object may not exceed 16 TiB, which is aligned with the OneFS file system.

A OneFS S3 extended-request parameter examples

A.1 x-isi-path extended header

When creating a bucket through S3, OneFS provides an optional S3 extension to specify a OneFS directory as a bucket path. The following is an example of adding the extended header through boto3.

```
import boto3
HOST=<ip/fqdn>  # Your SmartConnect name or cluster IP goes here
USERNAME='1_local_user01_accid'  # Your access ID
USERKEY='mWQbXkadl2CR_x2_WRj4tYu_d11j'   # Your secret key
URL = 'http://{}:9020'.format(HOST)
s3 = boto3.resource('s3')
session = boto3.Session()
s3client =
session.client(service_name='s3',aws_access_key_id=USERNAME,aws_secret_access_ke
y=USERKEY,endpoint_url=URL,use_ssl=False,verify=False)
bucket_name='bucket01'
bucket _path = "/ifs/data/s3buckets/{}".format(bucket_name)
headers = {'x-isi-path': bucket_path}

# pass in custom headers before CreateBucket call
add_headers = (lambda **kwargs: kwargs['params']['headers'].update(headers))
s3client.meta.events.register('before-call.s3.CreateBucket', add_headers)

s3client.create_bucket(Bucket=bkt_name)
```

A.2 x-isi-force-delete extended header

By default, a bucket cannot be deleted if it is not empty. With the x-isi-force-delete extension, a bucket can be deleted from OneFS while preserving the data under the bucket. This option is effective if it is contained in a request. A value does not have to be assigned for this header, and an empty string is acceptable. The following is an example of adding the extended header through boto3.

```
import boto3
HOST=<ip/fqdn>  # Your SmartConnect name or cluster IP goes here
USERNAME='1_local_user01_accid'  # Your access ID
USERKEY='mWQbXkadl2CR_x2_WRj4tYu_d11j'   # Your secret key
URL = 'http://{}:9020'.format(HOST)
s3 = boto3.resource('s3')
session = boto3.Session()
s3client =
session.client(service_name='s3',aws_access_key_id=USERNAME,aws_secret_access_ke
y=USERKEY,endpoint_url=URL,use_ssl=False,verify=False)
bucket_name='bucket01'
headers = {'x-isi-force-delete': ''}

# pass in custom headers before DeleteBucket call
```

```
add_headers = (lambda **kwargs: kwargs['params']['headers'].update(headers))
s3client.meta.events.register('before-call.s3.DeleteBucket', add_headers)

s3client.delete_bucket(Bucket=bkt_name)
```

B Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

Storage technical documents and videos provide expertise that helps to ensure customer success on Dell Technologies storage platforms.

B.1 Related resources

- AWS S3 API Reference
- Boto3 Documentation