

Uploading an Object to a Bucket



Service Status

The Object Storage Unit (OSU) service is now **END OF SUPPORT AND MAINTENANCE**. For more information, see [End-of-Life Policy](#).

- [EN](#)
- [FR](#)

You can upload a local object to a bucket to store it and create a backup in the Cloud, and possibly share it with other users.

As the storage capacity is only restricted by the storage platform itself, objects are not limited in size. You can upload an object whose size is from 1 byte to 5 GiB in a single command. However, it is recommended to upload objects of 100 MiB or more in several smaller parts in order to upload them faster. For more information, see [Creating a Multipart Upload](#).

You can upload objects in your buckets or in a bucket another user shared with you. Other users can also upload objects in your bucket if you granted them the appropriate permissions. For more information, see [Access Control List \(ACL\) Reference](#).

You cannot change the key of an object once it is uploaded in the bucket. If you upload an object with the same key as another object already contained in the bucket, this object you upload replaces the former object. If the bucket versioning is enabled, the former object is conserved as a previous version. For more information, see [Enabling or Disabling Bucket Versioning](#).

By default, the objects you upload are private. However, you can share the objects that you uploaded in your bucket with other users setting an Access Control List (ACL). For more information, see [Access Control List \(ACL\) Reference](#).

This action also returns an ETag element that you need to create a multipart-upload. For more information, see [Creating a Multipart Upload](#).



Ensure not to set public read permissions for your sensitive data.

This feature is not available from Cockpit. This documentation only describes the procedure using AWS CLI.

- To upload a local object to a bucket, use the **put-object** command following this syntax:

Request sample

```
$ aws s3api put-object \
  --profile YOUR_PROFILE \
  --bucket BUCKET \
  --key OBJECT \
  --body usr/bin/YOUR_OBJECT \
  --acl private \
  --content-length 42 \
  --grant-full-control "id=USER_ID, id=USER_ID" \
  --grant-read "id=USER_ID, id=USER_ID" \
  --grant-read-acp "id=USER_ID, id=USER_ID" \
  --grant-write-acp "id=USER_ID, id=USER_ID" \
  --endpoint ENDPOINT
```

This command contains the following attributes that you need to specify:

- (optional) `profile`: The named profile you want to use, created when configuring AWS CLI. For more information, see [Installing and Configuring AWS CLI](#).
- `bucket`: The bucket in which you want to upload the object.
- `key`: The name you want to give to your object in the bucket.

Related Pages

- [Copying an Object to a Bucket](#)
- [Listing the Objects of a Bucket](#)
- [Removing Objects from a Bucket](#)

Tutorial: Setting Up a Bucket with Objects

[Previous Step:](#)

(optional) [Configuring a Bucket ACL](#)

[Next Step:](#)

(optional) [Configuring an Object ACL](#)

Tutorial: Copying an OMI Across Different Regions (API)

[Previous Step:](#)

[Creating a Manifest File for an Image Export Task](#)

[Next Step:](#)

[Configuring a Pre-signed URL](#)



You can use the same prefix in the keys of several objects in the following format: `prefix/key1`, `prefix/key2`, and so on.

This creates a virtual architecture in your bucket and enables you to retrieve your objects faster when listing them. For more information, see [Listing the Objects of a Bucket](#).

- `body`: The path to the object on your machine.



- When specifying new permissions, all the previous permissions are replaced. Therefore, you need to specify both the existing permissions that you want to keep (including for yourself) and the new permissions that you want to give in a single command.
- If you are the owner of the bucket, you can lose your own permissions but not the ability to manage the ACL itself.

For more information about existing permissions, see [Getting Information About a Bucket ACL](#) and [Getting Information About an Object ACL](#).

- (optional) `acl`: The permissions you grant for your object (`private` | `public-read` | `public-read-write` | `authenticated-read` | `bucket-owner-read` | `bucket-owner-full-control`).
- (optional) `content-length`: If the size cannot be defined automatically, the size of the object, in bytes.
- (optional) `grant-full-control`: One or more IDs of users to whom you grant the `full-control` permission.
- (optional) `grant-read`: One or more IDs of users to whom you grant the `read` permission.
- (optional) `grant-read-acp`: One or more IDs of users to whom you grant the `read-acp` permission.
- (optional) `grant-write-acp`: One or more IDs of users to whom you grant the `write-acp` permission.



- When using OOS, you need to specify S3 user IDs. You can retrieve S3 user IDs via the [Listing Your Buckets](#) and [Listing the Objects of a Bucket](#) methods using the `oos` endpoint.
- When using OSU, you need to specify OUTSCALE account IDs.
- In both cases, you can also specify user email addresses using the `emailaddress=name@domain.com` format.

- `endpoint`: The endpoint corresponding to the service (`oos` or `osu`) and Region you want to send the request to, in the following format: `https://<SERVICE>.<REGION>.outscale.com`

The `put-object` command returns the following element:

- `ETag`: The ETag of your object, that you need to use the **multipart upload** commands. For more information, see [Creating a Multipart Upload](#).

Result sample

```
{
  "ETag": "\"fae0000b0d0000fababdd00e0a000000\""
}
```

The object is uploaded to the bucket.



- You can also perform this action using an s3 command following this syntax:

```
$ aws s3 cp --profile YOUR_PROFILE LOCAL_PATH  
/OBJECT_TO_UPLOAD s3://YOUR_BUCKET --endpoint ENDPOINT
```

- You can also manage your object storage resources using s3cmd commands. For more information, see [s3cmd](#).

AWS™ and **Amazon Web Services™** are trademarks of Amazon Technologies, Inc or its affiliates in the United States and/or other countries.