

# S3 and Glacier Storage User's Guide

- [Introduction](#)
- [Bucket Naming Convention](#)
- [Bucket Lifecycle and Glacier Storage](#)
- [S3 Bucket Access](#)
  - [Access from AWS EC2 Instances](#)
  - [Access from Other Locations](#)
- [S3 Commands](#)
- [S3 Command Examples](#)
- [Summary Notes](#)
- [References](#)

## Introduction

This document describes the current set up and design of CDL S3 and Glacier storage in AWS. S3 and Glacier are object based storage is **NOT** mounted like a filesystem and accessible via `aws cli` and `api` commands. There are two types of buckets for each program, one with versioning enabled and one without. Versioning provides the ability to restore objects in case of accidental deletion, however there is a storage cost if there are multiple versions of the object. Developers may choose which bucket to use based on application requirements. Within a bucket, "folders" are established for individual services. A "folder" is simply a prefix to the key of an object stored in a bucket.

## Bucket Naming Convention

### CDL AWS Buckets - Versioning Not Enabled

Each "non-versioning" bucket will have the following format of program and environment:

*Format:*

`<program>-s3-<environment>`

*Example:*

`d2d-s3-dev, d2d-s3-prd, d2d-s3-stg, ias-s3-dev, ias-s3-prd, ias-s3-stg, mdg-s3-dev, mdg-s3-prd, mdg-s3-stg, pub-s3-dev, pub-s3-prd, pub-s3-stg, uc3-s3-dev, uc3-s3-prd, uc3-s3-stg, web-s3-dev, web-s3-prd, web-s3-stg`

### CDL AWS Buckets - Versioning Enabled

Each "versioned enabled" bucket will have a suffix of "ver" in the following format of program and environment:

*Format:*

`<program>-s3ver-<environment>`

*Example:*

`d2d-s3ver-dev, d2d-s3ver-prd, d2d-s3ver-stg, ias-s3ver-dev, ias-s3ver-prd, ias-s3ver-stg, mdg-s3ver-dev, mdg-s3ver-prd, mdg-s3ver-stg, pub-s3ver-dev, pub-s3ver-prd, pub-s3ver-stg, uc3-s3ver-dev, uc3-s3ver-prd, uc3-s3ver-stg, web-s3ver-dev, web-s3ver-prd, web-s3ver-stg`

## Bucket Lifecycle and Glacier Storage

Each bucket has one or more "service folders," each with a sub-folder named `glacier`. These folders will be created by IAS at the time the request is made. The `glacier` sub-folder will have the "lifecycle rule" that specifies any object in the `glacier` sub-folder will be moved to Glacier storage, by default the same day (0 days). **Note: Be advised that any folder created by an end-user named "glacier" or "Glacier" does not mean that it will be automatically archived into Glacier storage.**

*Format:*

`<S3Bucket>/<service>`  
`<S3Bucket>/<service>/glacier`

*Example:*

`d2d-s3-prd/ht/`  
`d2d-s3-prd/ht/glacier`

Glacier is suitable for long term archival storage. Developers may choose to place objects in the `glacier` folder or not, depending on application requirements. Reference: <http://docs.aws.amazon.com/AmazonS3/latest/dev/object-archival.html>

## S3 Bucket Access

### Access from AWS EC2 Instances

This is the standard use case and requires no special credentials. Every CDL EC2 instance is associated with an IAM system role. A bucket policy is attached to the IAM role for system(s) that require access to a specific bucket/folder. Any user logged in to that system can access resources based on this policy (from a personal account as well as a role account).

### Access from Other Locations

An IAM user will be created in the CDL AWS CDL account with credentials to access the S3 Bucket/<service> folders. IAS will download the credentials of the IAM user and install them in the system role account on the instance or external host. An .aws directory in the role account home directory contains the configuration and keys to access the <S3 Bucket>/<folder>.

## S3 Commands

Buckets may be accessed using the AWS S3 CLI commands or one of the AWS SDKs.

- S3 commands: <http://docs.aws.amazon.com/cli/latest/reference/s3/index.html>
- S3 API commands: <http://docs.aws.amazon.com/cli/latest/reference/s3api/index.html>
- SDKs for supported languages (Java, Python, Ruby, others): <https://aws.amazon.com/tools/>

The AWS CLI is pre-installed on Amazon Linux. If it is not available on your system, follow the directions here to install it: [AWS CLI COMMANDS](#).

## S3 Command Examples

- To upload an object a.k.a. file to your S3 Bucket/folder:

#### S3 Commands, upload and object

```
.  
aws s3 cp <filename> s3://<bucketname>/<foldername>/  
.
```

- To sync up a directory to your S3 Bucket/folder. Note: This uploads new and/or updated objects to S3.

#### S3 Commands, sync a directory

```
.  
aws s3 sync <local_directory> s3://<bucketname>/<foldername>/  
.
```

- To list the contents of a S3 Bucket/folder:

#### S3 Commands, list folder content

```
.  
aws s3 ls s3://<bucketname>/<foldername>/  
.
```

- To delete a file from a bucket/folder:

#### S3 Command, delete a file

```
. aws s3 rm s3://<bucketname>/<foldername>/filename .
```

- To recursively upload multiple objects in a directory S3 Bucket/folder/glacier: (Note: This is significantly faster than uploading objects one at a time.)

### S3 Commands, recursively upload objects

```
aws s3 cp <directory> s3://<bucketname>/<foldername>/glacier/ --recursive
```

- To list objects in an S3 Bucket/folder/glacier:

### S3 Commands, list objects

```
aws s3api list-objects --bucket <bucketname> --prefix <folder>/glacier/
```

or

```
aws s3 ls s3://<bucketname>/<folder>/glacier/
```

- To restore an object from S3 Bucket/folder/glacier for 3 days:

**Note:** You can only restore an object and not a prefix aka directory. One idea is to create a "for loop" of your listed objects and restore each object.

### S3 Commands, restore an object with time constraint

```
aws s3api restore-object --bucket <bucketname> --key <foldername>/glacier/<filename> --restore-request Days=3
```

- Command to see if the S3 object is restored and available for copy and output.
- Note: A restore from glacier can take up to few hours to days. The "ongoing-request" output means that the object is not available.

### S3 Commands, check restored object

```
aws s3api head-object --bucket <bucketname> --key <foldername>/glacier/<filename>
```

Output:

```
{
  "Restore": "ongoing-request=\false\", expiry-date=\Sun, 13 Aug 2017 00:00:00 GMT\"",
  ...
  "StorageClass": "GLACIER",
  "Metadata": {}
}
```

- Command to copy an object from your S3 Bucket/folder/glacier to current directory once it's restored:

**Note:** the "dot" at the end of syntax.

### S3 Commands, copy restored object to directory

```
aws s3 cp s3://<bucketname>/<foldername>/glacier/<filename> .
```

- Command to copy all objects from your S3 Bucket/folder to current directory:

**Note:** the "dot" in the syntax.

### S3 Commands, copy all objects to current directory

```
aws s3 cp s3://<bucketname>/<foldername>/ . --recursive
```

- Command to move file from S3 Bucket/folder to Glacier storage.

### S3 Commands, move file to glacier

```
aws s3 mv s3://<bucketname>/<foldername>/<file_name> s3://<bucketname>/<foldername>/glacier/
```

- Command to create a subfolder in the S3 bucket. **Note:** you cannot easily create an empty folder(prefix) because prefixes do not get created if there is no corresponding object for it. This is one method to create the folder(prefix).

### S3 Commands, create a subfolder

1. Create an empty file in your local directory.
2. `aws s3 cp <emptyfile> s3://<bucketname>/<foldername>/<new-sub-folder>/`

- Working with versioned buckets

### S3 Commands, version s3 buckets

List all versions of objects in a bucket:  
`aws s3api list-object-versions --bucket <bucket> --prefix <folder>/`

Retrieve a specific version:  
`aws s3api get-object --bucket <bucket> --key <folder>/<filename> --version-id <version> local-file`

- Deleting objects

### S3 Commands, delete object

Delete an object:  
`aws s3api delete-object --bucket <bucket> --key <folder>/<filename>`

Delete a specific version of an object:  
`aws s3api delete-object --bucket <bucket> --key <folder>/<filename> --version-id <version>`

To permanently delete an object in a versioned bucket, you must explicitly delete all versions of that object.

- Checking capacity of your bucket or prefix

### S3 Commands, delete object

```
aws s3 ls s3://<bucketname> --recursive | grep -v -E "(Bucket: |Prefix: |LastWriteTime|^$|--)" |awk 'BEGIN {total=0} {total+=$3}END{print total/1024/1024" MB"}'
```

```
aws s3 ls s3://<bucketname>/<prefix>/ --recursive | grep -v -E "(Bucket: |Prefix: |LastWriteTime|^$|--)" |awk 'BEGIN {total=0}{total+=$3}END{print total/1024/1024" MB"}'
```

- List capacity of each object in a bucket and prefix

### S3 Commands, delete object

```
aws s3api list-objects --bucket <bucketname> --prefix <prefix>/ --query 'Contents[].{Key: Key, Size: Size}'
```

## Summary Notes

- "Empty" directories cannot be stored into S3.
- "Symbolic links" cannot be stored into S3, but instead upload the data referenced by the link.
- "Time stamps" are not preserved.
- The role account is responsible for any deletion of objects in a bucket. To fully delete an object in a versioned bucket, all versions of the object must be deleted.
- IAS cannot recover objects that have been deleted by the role account, whether in a versioned bucket or not. S3 is durable, redundant storage, but there are no backups.
- .

## References

- <http://docs.aws.amazon.com/cli/latest/reference/s3/index.html>
- <http://docs.aws.amazon.com/cli/latest/reference/s3api/index.html>
- <http://aws.amazon.com/documentation/s3/>
- <https://aws.amazon.com/tools/>