

# Your Personal Amazon S3 Cheat Sheet

For many organizations and startups, Amazon S3 is their first step into the world of cloud adoption. That's why it's considered an important service to know inside and out for anyone taking an AWS certification exam.

In this article, we will give you an AWS certification cheat sheet for Amazon S3. Not only will this cheat sheet introduce you to many popular Amazon S3 features, it will also show you how to make configuration changes by accessing Amazon S3 through the Amazon S3 CLI.

## Amazon S3 Basics

---

When it comes to AWS storage in the cloud, every company has its own reasons for turning to Amazon S3. Enterprises might need it to scale, for backup & DR, archiving, hybrid storage or for analytics. For other organizations, including startups, it's the website hosting, software delivery version management, or content management (CDN) that brings them to AWS. In all of these cases, Amazon S3 is considered the AWS storage service of choice.

Note: The following commands cover the Amazon S3 CLI, which can be installed on Windows, Linux, MAC and Unix. Amazon S3 can also be accessed through the AWS Console, SDK, or RESTful APIs.

To start, let's look at the two most basic elements of Amazon S3: buckets and objects.

### Buckets

A bucket is a top-level container where you store your files (which are known as objects in Amazon S3 jargon). The bucket name has to be unique across all AWS accounts and all AWS regions.

- **Cross-To create bucket in a specified region:**

```
aws s3 mb s3://bucketname --region us-east-1
```

- **To list all buckets:**

```
aws s3 ls
```

- **To list everything inside a bucket:**

```
aws s3 ls s3://bucketname
```

## Objects

An object is the basic unit of storage in Amazon S3. Any item or file stored in Amazon S3 is known as an object. Each object will have a unique key to identify it along with its content and metadata. A single object can be between 0 byte to 5 TB.

There are 3 primary storage classes of Amazon S3 objects, and each serves a different use case as well as differing in durability and cost:

- 1. Standard:** Generic and default storage for Amazon S3 that gives up to 11 9s of durability.
- 2. Infrequent Access:** For less-frequently accessed objects this storage format is the ideal choice. It gives the same durability as standard storage, but it is cheaper in storage costs. It should be noted that data transfer costs for Infrequent Access objects are a little higher than with other classes.
- 3. Reduced Redundancy:** Less durable (99.99%) but a cheaper option compared to Standard storage.

In addition, the above three storage classes, Amazon S3's lifecycle allows you to change the object class to **Amazon Glacier**, which is a long term archival storage.

It's very cheap compared to Standard storage but it can take anywhere from 1–5 minutes to 5–12 hours for data to be retrieved.

Some useful commands to work with objects:

- You can upload or copy a file from your local machine to Amazon S3. That file will then be treated as an object in Amazon S3:
- To recursively copy files under local directory to Amazon S3 but exclude files with a specific extension:

```
aws s3 cp test.txt s3://bucketname/test2.txt
```

```
aws s3 cp myDir s3://bucketname/ --recursive --exclude  
"* .jpg"
```

## Amazon S3 Bucket Features



In this section we'll take a look at some of the core and advanced features of Amazon S3 buckets.

### 1. Versioning

Versioning allows you to maintain older copies of an object when that object is modified. AWS supports multiple versions of individual objects. The main use for versioning is to keep objects safe from accidental deletion.

- To enable bucket versioning:

```
aws s3api put-bucket-versioning --bucket bucketname --versioning-configuration  
Status=Enabled
```

## 2. Static Website Hosting

One of the marquee use cases for Amazon S3 is static website hosting. The static files can have client-side scripting (such as Angular, AJAX, etc.) to process dynamic content at the backend (such as Amazon EC2, AWS Lambda, etc). Static website hosting allows you to map your domain to a static website URL with the Amazon Route 53 DNS service.

- To set up static website hosting:

```
aws s3 website s3://bucketname/ --index-document index.html --error-document error.html
```

## 3. Bucket Logging

When you want to get reports on bucket access, such as object names, requester, bucket name, request time, request action or more, you should enable bucket logging. Bucket logging creates log files in the Amazon S3 bucket.

- To set up bucket logging:

```
aws s3api put-bucket-logging --bucket MyBucket --bucket-logging-status file://logging.json
```

## 4. Tagging Amazon S3 Buckets and Objects

Tags are useful for billing segregation as well for distribution of control using Identity and Access Management (IAM).

- To tag a bucket:

```
aws s3api put-bucket-tagging --bucket bucketname --tagging  
'TagSet=[{Key=organization,Value=sales}]'
```

## 5. Amazon S3 Transfer Acceleration

Amazon S3 transfer acceleration allows for faster transfer of objects using Amazon CloudFront. Although it saves time and improves performance, you should know that transfer acceleration also increases transfer costs. Transfer acceleration is ideal for when you want a faster upload to a central bucket from around the globe or when you have large amount of content in GBs to upload.

- To enable transfer acceleration:

```
aws s3api put-bucket-tagging --bucket bucketname --accelerate-configuration Enabled
```

## 6. Amazon S3 Inventory Configuration

Amazon S3 inventory configuration allows users to download a comma-separated values (CSV) flat-file of objects and their corresponding metadata on a daily or weekly basis. This is useful when you want to execute a process or run analyses based on inventory of that data..

- To set up an Amazon S3 inventory:

```
aws s3api put-bucket-inventory-configuration --bucket bucketname --id 123 --inventory-configuration 'Destination={S3BucketDestination={AccountId=string,Bucket=string,Format=string,Prefix=string}},IsEnabled=boolean,Filter={Prefix=string},Id=string,IncludedObjectVersions=string,OptionalFields=string,string,Schedule={Frequency=string}'
```

## 7. Lifecycle Configurations

Amazon S3 allows you to change the storage class of an object with Amazon S3 lifecycle configuration. This is helpful when you have objects stored for long durations and you want to save on AWS storage costs by migrating them to the Infrequent Access storage class or archive them on Amazon Glacier. It's all about automation: This feature allows you to set up rules that will move the object to a cheaper storage class without manual intervention. In addition to that, lifecycle configuration also allows you to set rules that automatically delete objects which are no longer required. This is ideal for log files, backup data and other files which you want to store for certain amount of time, but don't need once new versions are available.

- To put lifecycle configuration on a bucket:

```
aws s3api put-bucket-lifecycle-configuration --bucket bucketname --lifecycle-configuration file://lifecycle.json
```

## 8. Bucket Policy

Bucket policy allows the user to define access rights for objects at the bucket level instead of setting an ACL at the individual object level. To set a bucket policy, you can either download a sample policy or create your own from scratch.

- To download a sample Amazon S3 bucket policy, run the following command::

```
aws s3api get-bucket-policy --bucket mybucket --query Policy --output text > policy.json  
'TagSet=[{Key=organization,Value=sales}]'
```

Once downloaded, modify the policy .json as required (such as bucket name, policy rights etc). The last step is to put the modified policy into effect back on the Amazon S3 bucket.

- You can do that by running this command:

```
aws s3api put-bucket-policy --bucket mybucket --policy file://policy.json
```

## 9. Bucket Analytics Configuration

Amazon S3 bucket analysis helps identify whether you are storing objects in the right storage class or not. It helps to identify storage access patterns. For example, objects that aren't accessed very often will be recommended a move to the Infrequent Access storage class; objects which are rarely accessed at all may be recommended archiving in Glacier.

- To run bucket analysis:

```
aws s3api put-bucket-analytics-configuration --bucket bucketname --id 123 --analytics-configuration file://analytics.json
```

## 10. Bucket Metrics Configurations

Amazon S3 offers storage and request metrics. Request metrics are available at every-minute frequency.

- To put a request metrics configuration on a bucket:

```
aws s3api put-bucket-metrics-configuration --bucket bucketname --id 123 --metrics-configuration file://metrics.json
```

## 11. CORS Configuration for Buckets

If you are creating static website hosting with a rich client UI, you will have to configure Cross-Origin Resource Sharing (CORS) at the bucket level. CORS allows you to have a client application hosted on one domain in order to access an application that is hosted on another domain.

- To put CORS on bucket:

```
aws s3api put-bucket-cors --bucket bucketname --cors-configuration file://cors.json
```

Note: The cors.json file will be a json document which will specify the rules for CORS. To get to know its structure and see an example, visit [this link](#).

## 12. Set Bucket Notifications

Amazon S3 bucket notifications allow you to receive notifications when certain events (such as an upload or an object modification) take place.

- To put bucket notifications on a bucket:

```
aws s3api put-bucket-notification --bucket bucketname --notification-configuration file://notification.json
```

## 13. Cross-Region Replication

Bucket replication creates a replica of an object in a separate bucket. This is useful for DR since it allows a user to replicate data in separate regions.

- You can do it by running this command:

```
aws s3api put-bucket-replication --bucket bucketname --replication-configuration file://  
replication.json
```

## 14. Requester Pays Buckets

Generally, the owner of a bucket will be the person who pays for that bucket's storage as well as for data transmission. With the Requester Pays Bucket feature, owners can configure their bucket so that the requester pays for the data transmission costs. The owner still pays for storage.

- To set up a Requester Pays Bucket:

```
aws s3api put-bucket-request-payment --bucket bucketname --request-payment-configuration  
Payer=Requester
```

## 15. Multipart Upload

Amazon S3 multipart upload allows users to upload large objects in separate parts, in any order, as a way to create a faster data upload.

- To create multipart upload for the key "multipart/01" in the bucket "bucketname":

```
aws s3api create-multipart-upload --bucket bucketname --key 'multipart/01'
```

# Object Configuration & Services



This section will discuss different configurations and services that can be applied to Amazon S3 objects.

## 1. Amazon S3 Bittorrent Protocol

Amazon S3 supports the BitTorrent protocol, which is a peer-to-peer protocol for a fast and cost-effective option for downloading objects from Amazon S3. This is useful when you have a number of people downloading the same large file. Peer-to-peer sharing allows costs to be optimized.

- To get object using the Bittorrent protocol:

```
aws s3api get-object-torrent --bucket bucketname --key large-video-file.mp4 large-video-file.  
torrent
```

## 2. Amazon S3 Encryption

Amazon S3 supports both server-side and client-side encryption. Client-side encryption is managed by the user while Amazon S3 provides AWS 256-bit server-side encryption. With server-side encryption, objects are encrypted before they are stored in AWS data centers and decrypted by Amazon S3 before they are delivered back to the user.

- To encrypt an object:

```
aws s3 cp --sse s3://bucketname/objectname
```

## 3. Pre-Signed Amazon S3 URLs

All Amazon S3 objects and buckets are private by default. If a user wants to allow other accounts or customers without AWS credentials to upload objects to the user's bucket, that can be achieved with pre-signed URLs.

- To generate a pre-signed URL with an expiry time in seconds:

```
aws s3 presign s3://bucketname/test.txt --expires-in 4800
```

## Conclusion

Amazon S3 is one of the most important services on AWS, so knowing it well can come in handy during an examination. Some prominent topics for certifications are storage classes, consistency model, ACL and policy, performance and lifecycle models.

This Amazon S3 cheat sheet was created to give you an edge in an exam. However, it is highly recommended that you refer to the Amazon S3 FAQs for a full refresher course on Amazon S3 topics before you head into an exam.

It's also helpful to practice hands-on using the AWS web console and AWS CLI before you put your skills to the test, in order to be completely sure you're ready for all AWS storage and Amazon S3-related questions you'll run into on your AWS exam.