# S3

## TOC



## Overview

S3 (or Simple Storage Services) was one of the first services offered by AWS, as a result it features heavily in the exam. It is a object store unlike EBS which is block storage. Object storage is where you store a file as apposed to block storage where you might install a OS for example. There is no limit to the amount of storage that you can use.

## Key Facts

### Object Sizes

Objects can be between 0 bytes and 5TB.

### Storage Classes of S3

There are four storage classes in S3. They are:

- Standard
- IA or infrequent access
- One-Zone IA (Infrequent Access)
- Glacier

Each class has a different availability and durability rating.

## Availability

Each storage class has been designed for a different availability level. They are:

| Storage Class | Availability |
|---|---|
| Standard | 99.99% |
| IA | 99.9% |
| One-Zone IA | 99.5% |
| Glacier | N/A |

## Durability

Each storage class has the same durability level:

| Storage Class | Durability |
|---|---|
| Standard | 99.999999999% |
| IA | 99.999999999% |
| One-Zone IA | 99.999999999%* |
| Glacier | 99.999999999% |

However, as noted in the AWS documentation, because S3 one zone-IA is only storage in one AWS availability zone, if that AZ is destroyed or if the region is off-line the data will be lost.

## Consistency Models

There are two consistency models used when uploading content to a S3 bucket. For new PUT requests the consistency model used is read after write consistency, meaning that once the file is uploaded you can read it immediately. For overwrite PUT and DELETE requests it is eventual consistency, meaning that if you upload the file and then try to access it you may not get the updated copy.

When uploading files to a S3 bucket you should get a 200 code to signal a successful upload.

## Bucket URLs

Bucket URLs have the same format across all regions, however there are two ways that they can be addressed. They can either be addressed virtual-hosted-style or path-style. This is a virtual-hosted-style address:

```
http://bucketname.s3.amazonaws.com
OR
http://bucketname.s3-eu-west-1.amazonaws.com
```

If you don't specify the region in a virtual-hosted-style link the DNS has enough information to pass it onto the correct region.

This is a path-type address:

```
http://s3.amazonaws.com/bucketname (US East, N. Virginia)
http://s3-eu-west-1.amazonaws.com/bucketname
```

As you can see, you must specify the region that your bucket is in if using the path-style address. The only exception to this is if your bucket is in US East North Virginia, where you do not specify the region.

# S3 Storage Classes

## Standard

- Standard storage has a availability of 99.99%.
- The data is resilient in the event of one AZ destruction.
- Has low latency and high throughput performance.

## Infrequent Access or IA

- IA has a availability of 99.9% and is resilient in the event of one AZ destruction.
- Has lower latency and high throughput performance.
- IA is set the object level and can exist in the same bucket as S3 standard.
- Has a lower storage and per GB retrieval fee

## One Zone-Infrequent Access

- One Zone-IA has a availability of 99.5%.
- Has a low latency and high throughput performance.
- Is stored in a single AZ, meaning it is not resilient if the AZ goes down.
- Set at the object level, meaning it can reside with standard and IA objects in the same bucket.

## Glacier

- Used for data archiving.
- Very cheap to store data, however there is a cost to retrieve data.
- Can take several hours to get the data back out.

# Metadata

The metadata of a S3 object is made up of different components:

- A key - The name of the object.
- The value - The bytes which make up the file.
- Version ID - This is used for versioning.
- Metadata - The data describing the object, for example when you uploaded it.
- Sub-resources - Access control lists (ACLs) and torrents.

## How S3 Is charged

S3 is charged in three ways:

- Charged by GB stored.
- Per request to S3.
- Storage management e.g. tagging of the object, moving data between regions and transfer acceleration

## Transfer Acceleration

Transfer acceleration allows for fast, easy and secure transfer of files over a long distance. It does this by using Cloudfront.

Transfer acceleration is when a user uploads a file to their nearest AWS edge, which is then sent over from the edge to the bucket.

For example, a user may have a bucket in London but a user in Sydney, The user in Sydney would upload their file to their nearest edge which then passes the data back to the bucket in London.

## Cloudfront

Cloudfront is a content delivery network or CDN. It speeds up the delivery of content to users by caching the data at many edge locations around the world. Edge locations are separate to AZs or regions. They are normally located in places that do not have any AZs.

When setting up Cloudfront you can specify the data origin. This can be either a EC2 instance, S3 bucket or an elastic load balancer with EC2 instances behind it. You can also have a origin which isn't a AWS service.

If it is the first time that a edge location is asked for data and it is not in its cache, the edge location will go to the origin, get the data and send it to the user whilst also caching it for next time. After the first time, each subsequent user will get the cached version. This will result in fast load times.

There are two types of distribution that can be setup, web distribution and RTMP distribution. Web distributions are used for websites (both static and dynamic) and RTMP distributions are used for media streaming.

Edge locations can be used to write information back to the origin.

## Security

By default, buckets are set to private when created. By doing this, no one else can read the content of the bucket unless you specify it. You can do this by using either a bucket policy or ACL.

You can configure access log for buckets which will give you information about the object and how its been accessed.

The data can also be encrypted. When uploading objects, this can be done over SSL/TLS. You can also use either client-side or server-side encryption for the data. Server-side is used to encrypt the data at rest.