**What is Simple Storage Service(S3)**

Before we move into S3, we should understand storage services in AWS.

AWS offers Storage classes.

1. Block level storage ===🡺 Amazon EBS, Instance store

2. File level storage ===🡺 Amazon EFS, Amazon FSx

3. Object level storage ===🡺 Amazon S3, Amazon S3 Glacier

**Simple Storage Service(S3)**

Amazon Simple Storage Service(S3) is a storage for the internet.

Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web.

The main concepts in S3 are Buckets and Objects.

Bucket is nothing but like a container where we store objects.

Object is nothing but a file, it may be an image file, pdf file or doc file.

Objects consist of object data and metadata. The data portion is opaque to Amazon S3. The metadata is a set of name-value pairs that describe the object. These include some default metadata, such as the date last modified, and standard HTTP metadata, such as Content-Type. You can also specify custom metadata at the time the object is stored.

Let’s say we have an app, in our app user has a profile with profile picture. So, user will upload/select an image file and set it as profile picture. We have to store these image file somewhere. So, we will store these image files in amazon S3 bucket as each image file is an object.

For example, if the object named “photos/puppy.jpg” is stored in the “awsexamplebucket1” bucket in the US West (Oregon) Region, then it is addressable using the URL “https://awsexamplebucket1.s3.us-west-2.amazonaws.com/photos/puppy.jpg”.

While creating the S3 bucket we can configure it in which AWS Region they should be create. Depending on the Region the cost will be vary.

If we create a S3 bucket in a Region which is far away from our application/user then the latency will increase.

So, these two factors (cost, latency) should be considered while selecting a Region for creating a S3 bucket.

To optimize latency, minimize costs, or address regulatory requirements, choose any AWS Region that is geographically close to you. For example, if you reside in Europe, you might find it advantageous to create buckets in the Europe (Ireland) or Europe (Frankfurt) Regions.

We can also configure a bucket so that every time an object is added to it, Amazon S3 generates a unique version ID and assigns it to the object. This is called versioning.

To use versioning, we just need to enable it on object in AWS.

**Storage classes in S3**

Amazon S3 offers a range of storage classes designed for different use cases. These include.

1. Amazon S3 STANDARD # For general-purpose storage of frequently accessed data

2. Amazon S3 STANDARD\_IA # For long-lived, but less frequently accessed data

3. S3 Glacier  # For long-term archive (i.e. data accessed occasionally)

**Bucket policies**

Bucket policies provide centralized access control to buckets and objects based on a variety of conditions, including Amazon S3 operations (read, write, GetObject, GetObjectVersion, DeleteObject, or DeleteBucket. etc.), requesters(users), resources (EC2, Lambda, etc.), and aspects of the request (for example, IP address).

This means we can specify what access (read/write) to give to which user/recourse like EC2, Lambda, etc. from which network/ IP address.

The permissions attached to a bucket apply to all of the bucket's objects that are owned by the bucket owner account.

Unlike access control lists which can only give/define allow permissions on individual objects, policies can either add or deny permissions across all objects within a bucket.

We can use wildcards on Amazon Resource Names (ARNs) and other values. Wildcards are like regular expression operators (\*, ^, $, etc.). then we can control access to groups of objects that begin with a common prefix or end with a given extension, such as *.html* at a time.

**Note:** Only the bucket owner is allowed to associate a policy with a bucket.

We can use AWS Identity and Access Management (IAM) to manage access to our Amazon S3 resources.

For example, we can use IAM with Amazon S3 to control the type of access a user or group of users has to specific parts of an Amazon S3 bucket our AWS account owns.

**\*Note:** A bucket is owned by the AWS account that created it. Bucket ownership is not transferable.

**Versioning:**

You can use *versioning* to keep multiple versions of an object in the same bucket.

Let say we have a doc file with 3 lines of data in it, we have uploaded it into S3 bucket. Now we made changes to file by adding 2 lines of data.

Now, if we enabled versioning on the bucket, when we uploaded the same file with new changes this won’t delete the existing old file, instead it will create a new object having updated data for the file with new version id.

**Managing public access to buckets:**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both.

To help us manage public access to Amazon S3 resources, Amazon S3 provides *block public access* settings while creating a bucket.

Amazon S3 block public access settings can override ACLs and bucket policies so that you can enforce uniform limits on public access to these resources.

You can apply block public access settings to individual buckets or to all buckets in your account.

To help ensure that all of your Amazon S3 buckets and objects have their public access blocked, we recommend that you turn on all four settings for block public access for your account. These settings block public access for all current and future buckets.

Before applying these settings, verify that your applications will work correctly without public access. If you require some level of public access to your buckets or objects, for example to host a static website as described at [Hosting a static website on Amazon S3](https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html), you can customize the individual settings to suit your storage use cases.

**Note:** You can enable block public access settings only for access points, buckets, and AWS accounts. Amazon S3 doesn't support block public access settings on a per-object basis.

**Note:** When you apply block public access settings to an account, the settings apply to all AWS Regions globally. The settings might not take effect in all Regions immediately or simultaneously, but they eventually propagate to all Regions.