**S3**

## S3 Concept

The S3 concept revolves around Bucket and Object. Generally, a file has a metadata, but in this case there are numerous metadata on a file which is why it’s called an object. S3 is an object-based storage. Metadata includes name of the object, size and date. The object cannot be independent but has to be within a bucket. The bucket is a container for an object. There can be hundreds of buckets in each Amazon account and within each bucket there can be hundreds of objects. Each object can vary from a size of 1 Byte to 5 Terabytes.

A key in the S3 concept is a unique identifier for an object within a bucket. Every object in a bucket has exactly one key. Each object would have an Access Control List which helps us know whether the object can be shared across the internet.

**S3 Features**

* Storing objects up to 5 TB in size
* Read and write entire object
* Every object has a unique developer assign key
* Collect objects into bucket
* Every object has a unique URL
* Intrinsic redundancy
* Full control of access rights
* Eventual consistency data model
* Supports versioning

## S3 Concept

The S3 concept revolves around Bucket and Object. Generally, a file has a metadata, but in this case there are numerous metadata on a file which is why it’s called an object. S3 is an object-based storage. Metadata includes name of the object, size and date. The object cannot be independent but has to be within a bucket. The bucket is a container for an object. There can be hundreds of buckets in each Amazon account and within each bucket there can be hundreds of objects. Each object can vary from a size of 1 Byte to 5 Terabytes.

A key in the S3 concept is a unique identifier for an object within a bucket. Every object in a bucket has exactly one key. Each object would have an Access Control List which helps us know whether the object can be shared across the internet.

**S3 Functionality**

* The user can write, read and delete objects of the size range 1 byte to 5 terabytes each. The number of objects that can be stored is unlimited. The objects can also be retrieved with a unique, developer-assigned key.
* The authentication mechanisms are provided to ensure that data is kept secure from unauthorized access. Objects can be made private or public, and right can be granted to specific users.
* Uses standards based on REST and SOAP (https) interfaces designed to work with any Internet-developed toolkit.
* It also supports the BitTorrent Protocol.
* S3 Reduced Redundancy Storage gives lower durability with lower costs.

## S3 Bucket Policy

Bucket policies provide centralized, access control to buckets. The bucket access is given in two ways namely Bucket Policy and Bucket Control list. In Bucket Policy, the user can give customized access to particular user, account and business hours. Access control lists (ACL) can only add (grant) permissions on individual objects and policies.

Each bucket and object in Amazon S3 has an ACL that defines its access control policy. The user can give 100 grants per ACL after that the user can adopt the Bucket Policy.

On storage classes : <https://www.msp360.com/resources/blog/amazon-s3-storage-classes-guide/>

## Amazon S3 Encryption Types

AWS has several offerings in the data encryption space. In addition to the Amazon S3 encryption offerings discussed here, Amazon Elastic Block Store ([AWS EBS](https://cloud.netapp.com/blog/ebs-volumes-5-lesser-known-functions)) encryption options are also available.

### **SSE Data Encryption**

Within Amazon S3, Server Side Encryption (SSE) is the simplest data encryption option available. SSE encryption manages the heavy lifting of encryption on the AWS side, and falls into two types: SSE-S3 and SSE-C.

The SSE-S3 option lets AWS manage the key for you, which requires that you trust them with that information. With SSE-S3, you don’t have access to see or encrypt data using the key directly, but you can be assured that the raw data you own is encrypted at rest by AWS’s standard processes.

The SSE-C option similarly manages encryption and decryption of your data for you, but uses a key provided by you (the customer) and passed in to AWS with each request to encrypt or decrypt. AWS does not store your key with this method, so you are responsible for its safe keeping.

### **S3 Client-Side Data Encryption**

[S3 Client-Side Encryption](https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html) puts all the responsibility for the encryption heavy lifting onto the user. Rather than allowing AWS to encrypt your data, you perform the encryption within your own data center and upload the encrypted data directly to AWS.

S3 Client-Side Encryption also comes in two options: server-side master key storage, and client-side master key storage.

In server-side master key storage, you can store your master key server-side in the AWS KMS (Key Management Service) service, and AWS will provide sophisticated key management software to manage sub-keys based on the master key that is used to encrypt your data.

In client-side master key storage, your master keys aren’t stored on AWS’s servers, and you take full responsibility for the encryption. Using this second approach is potentially the most secure, as your keys and data are never seen by Amazon servers in an unencrypted state. However, the level of security that you can achieve with this method depends on the integrity of your own processes and technology rather than AWS’s.