**Amazon S3(Simple storage service):**

* Amazon **S3** is a massively scalable object **storage** service (Blob storage)
* You can use Amazon S3 to store and retrieve **infinite** amount of data at any time, from anywhere.
* Independent of a server and is accessed over the Internet.
* Objects(files) reside in containers called buckets and is identified by a unique user-specified key (filename).
* Buckets are a simple folder.
* Each bucket can hold an unlimited number of objects.
* **highly durable** and **highly scalable** object storage
* Amazon s3 is **only for storing the data**, not to install O.S or run a database on it
* Available 24\*7\*365

**Bucket:**

* A container (web folder) for objects (files) stored in Amazon S3.
* Bucket names are **global**
* **unique** across all AWS accounts, much like domain names (DNS)
* can create and use multiple buckets up to **100 per account** by default.

**AWS region:**

* You control the location of your data; data in an Amazon S3 bucket is stored in that region unless **you explicitly copy it to another bucket located in a different region**.

**Objects:**

- Objects are **files** stored in Amazon S3 buckets.

* Size from **0 bytes – 5 TB**
* consists of **data** (the file itself) and **metadata** (data about the file).
* S3 object is a set of **name/value pairs**
* Two types of metadata:

1. **system metadata** (created and used by Amazon S3 itself e.g. the date last modified, object size,)
2. **user metadata** (optional, and it can only be specified at the time an object is created.)

**Keys:**

* S3 bucket is identified by a **unique identifier** called a key(filename).
* Keys must be unique within a single bucket, but different buckets can contain objects with the same key.

**Object URL:**

E.g.:

<http://mybucket.s3.amazonaws.com/jack.doc>

my bucket is the S3 bucket name, and jack.doc is the key or filename

E.g.:

D: Movies/English/2016/abc.m3

^ ^ ^ ^

Path=>s3 key / Parent=>s3Bucket/ folder/File=>s3 object

**S3 Properties:**

1. **Versioning:**

* Versioning allows you to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket.
* Protects you from unintended overwrites and deletions
* **Archive objects** so you have access to previous versions.
* By default, versioning is disabled you must **explicitly enable versioning** on your bucket.
* Each object in your bucket has a version ID regardless of whether you have enabled versioning.
* **Not enabled** versioning, S3 sets the value of the **version ID to null**.
* If you have **enabled versioning**, Amazon S3 **assigns a unique version ID** value for the object.
* When you enable versioning on a bucket, **objects already stored in the bucket are unchanged**.
* versioning **cannot be remove**d from a bucket; it can only be **suspended**

**MFA delete**: This adds another layer of data protection on top of bucket versioning.

Requires additional authentication in order to permanently delete an object version or change

the versioning state of a bucket.

1. **Server access logging:**

* Provides detailed records for the requests that are made to your bucket.
* Delivers access logs for a **source bucket to a target bucket** that you choose.

1. **Static website hosting**

* Can host a static website
* A static website might also contain client-side scripts.
* Amazon S3 does not support server-side scripting such as PHP, JSP, or ASP.NET

1. **Object level logging**

- **CloudTrail** supports logging Amazon **S3 object-level API operations** such as GetObject, DeleteObject, and PutObject. These events are called data events

**5.Events:**

- Enable Amazon S3 bucket events to send a notification message to a destination whenever the events occur.

- **Publish notifications** when new objects are **created** (by a **PUT, POST, COPY,** or multipart upload completion), when objects are **removed** (by a **DELETE**), or when Amazon S3 detects that an RRS object was lost.

- Notification messages can be sent through either Amazon Simple Notification Service **(Amazon SNS)** or Amazon Simple Queue Service **(Amazon SQS)** or delivered directly to **AWS Lambda** to **invoke AWS Lambda functions.**

1. **Transfer acceleration:**

- Fast, easy, and secure transfers of files between your client and an S3 bucket over long distances.

E.g.: Enable T.A, move the file to edge location, Amazon takes care to move the file to s3 bucket with 500GB/sec of speed

**6. Requester Pays:**

**- Bucket owners pay** for all Amazon S3 storage and **data transfer costs** associated with their bucket

-  A bucket owner, however, can configure a bucket to be a Requester Pays bucket.

- With Requester Pays buckets, the **requester pays the cost of the request and the data downloaded** from the bucket **instead of the bucket owner**.

**7.Tags:**

-  A tag is a **key-value pair** that represents a label that you assign to a bucket

- AWS organizes your resource costs on your cost allocation report using tags

 Two types of cost allocation tag:

1.AWS-generated tag: **AWS** **defines, creates, and applies** the AWS generated createdBy tag for you after an Amazon S3 CreateBucket event.

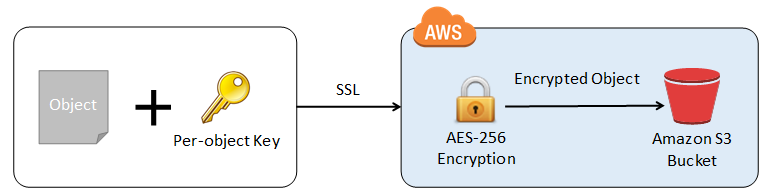
2.user-defined tags: **You define, create, and apply** user-defined tags to your S3 bucket.

**S3 Encryption:**

* S3 supports both **client side and server-side** encryption.

**Client-side Encryption:**

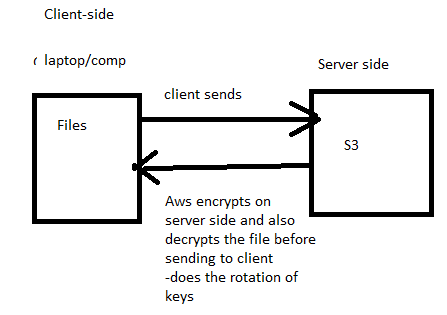
* **Encrypting the data** on the client side of your application **before sending it to AmazonS3**
* When using client-side encryption, **you have end-to-end control** of the encryption process, **including management of the encryption keys**.
* i.e., for example., you choose the algorithm and you choose the certificate(key)
* Use a client-side master key
* Use an AWS KMS-managed customer master key



The above shows both client and server-side encryption

**Server-side Encryption:**

* Amazon S3 encrypts your data at the object level as it writes it to disks in its data centers and decrypts it for you when you access it.
* Server side encryption (SSE) performed by Amazon s3 uses 256-bit Advanced Encryption Standard (AES) and AWS Key Management Service (Amazon KMS)



-