
Amazon S3

❖ AMAZON S3 (SIMPLE STORAGE SERVICE):

- S3 is easy to use **Object Storage**, with a simple web service interface to store and retrieve any amount of data from anywhere on the web.
- Objects are synced across all AZ's for extremely **High Available** and **Durability** (**99.999999999%**).
- S3 provides management features so that you can optimize, organize, and configure access to your data to meet your specific business, organizational, and compliance requirements.
- Objects are stored in **Buckets**.

BUCKET:

- Buckets are the main storage container for **Objects**.
- Each bucket must have a **Unique Name Space** across ALL of AWS.
- There is **Unlimited Storage**. **100 Buckets** per an account.
- With Amazon S3 bucket**:**
  - Serving content to customers.
  - Sharing Data with Ec2.

OBJECTS:

- **An Objects** are static files contain **Meta-Data** information.
- All objects are **Private** by default.
- When the object is in the bucket, you can open it, download it, and move it.
- Objects can**:**
  - Be as small as 0 bytes and as large as 5Tb.
  - Be made publicly available via a **URL**.

KEYS:

- An object key (or key name) is the unique identifier for an object within a bucket. Every object in a bucket has exactly one key.
- S3 as a basic data map between **"bucket + key + version"** and the object itself.

➢ S3 STORAGE CLASSES:

- Amazon S3 offers a range of storage classes designed for different use cases.
- Each object must be assigned a storage class type, which determines objects availability, durability, cost.

STORAGE CLASSES FOR FREQUENTLY ACCESSED OBJECTS:

- These classes provide immediate access in milliseconds to store data.

S3 STANDARD:

- The default storage class. If you don't specify the storage class when you upload an object, Amazon S3 assigns the S3 Standard storage class.

REDUCED REDUNDANCY(RRS):

- The RRS storage class is designed for noncritical, reproducible data that can be stored with less redundancy than the S3 Standard storage class.

STORAGE CLASS FOR AUTOMATICALLY OPTIMIZING DATA WITH CHANGING OR UNKNOWN ACCESS PATTERNS:

- **S3 Intelligent-Tiering** is an Amazon S3 storage class designed to optimize storage costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead.
- It is the only cloud storage that delivers automatic cost savings by moving data on a granular object level b/w access tiers when access patterns change.

STORAGE CLASSES FOR INFREQUENTLY ACCESSED OBJECTS:

- S3 Standard-IA and S3 One Zone-IA storage classes are designed for long-lived and infrequently accessed data.
- S3 Standard-IA (IA stands for infrequent access) and S3 One Zone-IA objects are available for millisecond access (similar to the S3 Standard storage class).

S3 STANDARD-IA:

- Amazon S3 stores the object data redundantly across multiple geographically separated Availability Zones.

S3 ONE ZONE-IA:

- Amazon S3 stores the object data in only one Availability Zone, which makes it less expensive than S3 Standard-IA.

S3 STANDARD-IA:

- Amazon S3 stores the object data redundantly across multiple geographically separated Availability Zones.

S3 ONE ZONE-IA:

- Amazon S3 stores the object data in only one Availability Zone, which makes it less expensive than S3 Standard-IA.

STORAGE CLASSES FOR ARCHIVING OBJECTS:

- The S3 Glacier storage classes are designed for low-cost data archiving.
- These storage classes offer the same durability and resiliency as the S3 Standard and S3 Standard-IA storage classes.

S3 GLACIER INSTANT RETRIEVAL:

- Use for archiving data that is rarely accessed and requires milliseconds retrieval.
- It offers a cost savings compared to the S3 Standard-IA storage class, with the same latency and throughput performance as the S3 Standard-IA storage class.

S3 GLACIER FLEXIBLE RETRIEVAL:

- Use for archives where portions of the data might need to be **retrieved in minutes.**
- A minimum storage duration period of **90 days** and can be accessed in as little as **1-5 minutes** using expedited retrieval.

S3 GLACIER DEEP ARCHIVE:

- Use for archiving data that rarely needs to be accessed.
- A minimum storage duration period of 180 days and a default retrieval time of 12 hours.
- S3 Glacier Deep Archive is the lowest cost storage option in AWS.

➢ S3 DATA CONSISTENCY MODEL:

- S3 Data Consistency provides strong **read-after-write** consistency for GET, PUT and DELETE requests of objects in the S3 bucket in all AWS Regions.
- When dealing with S3, we use HTTP requests. The most common requests used are:
- **GET:** Which is used to read data from S3
- **PUT:** Which is used to store the data in the request to S3.

- **DELETE:** Which is obvious from the name, it is used to delete data.
- S3 supports the following data consistency models:

READ-AFTER-WRITE:

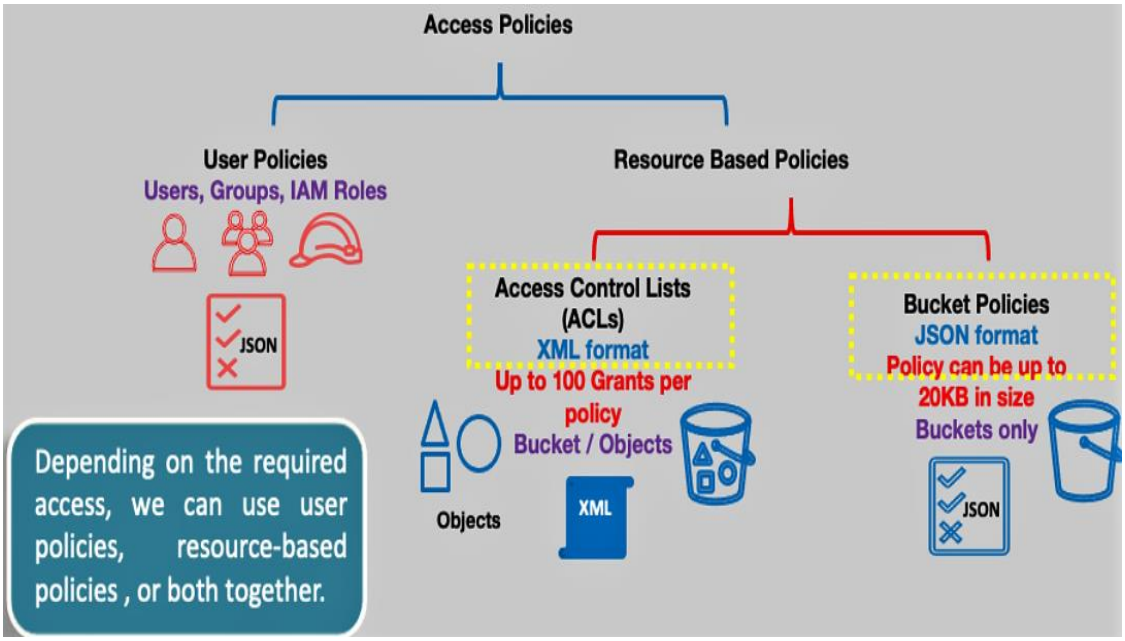
- It is an immediate or strong consistency of PUTs of new objects.
- A PUT is an HTTP request to store the enclosed data (with the request).

EVENTUAL CONSISTENCY:

- It is for overwrite PUTs and DELETEs (for changes/updates to existing Objects in S3)

➢ S3 ACCESS MANAGEMENT & POLICIES:

- Amazon S3 offers access policy options broadly categorized as resource-based policies and user policies.
- S3 resources are either bucket and objects.
- Buckets and objects are many sub resources each, including Access Control List (ACL) where each object or bucket has one.
- An access policy is the policy that provides grants to use or request actions against resources.
- An access policy defines and controls, through permissions, who has access to what.



UNDERSTANDING RESOURCE OWNERSHIP:

- Objects and buckets are private by default.
- Only the respective resource owner has "Full Control" access to the resource by default.
- The resource owner of the bucket is the account that created the bucket.
- The owner of the object is the account that upload ed the object, even if the bucket is owned by another account.

S3 ACCESS CONTROL LIST (ACL):

- S3 access control lists (ACLs) enable you to manage access to buckets and objects.
- AWS recommends using S3 bucket policies or IAM policies for access control.
- S3 ACLs is a legacy access control mechanism that predates IAM.
- An S3 ACL is a sub-resource that's attached to every S3 bucket and object.
- It defines which AWS accounts or groups are granted access and the type of access. When you create a bucket or an object, Amazon S3 creates a default ACL that grants the resource owner full control over the resource.

OBJECT ACL's:

- These are only used when it is required to provide permissions on objects in a bucket.
- This is the case of cross account permissions to upload objects in a bucket owned by another account.
- However, the bucket owner can do the following:
  - Deny access to any object in their bucket – including those they not own.
  - Delete any object in their bucket – including those they do not own.
  - Archive/restore any object in their bucket including those they don't own

BUCKET ACL's:

- Can provide permission to accounts or pre-defined groups.
- Can provide cross-account permissions to other accounts.
- Provide the WRITE permission over the bucket for LOG Delivery Groups to write access logs in the bucket.

DISADVANTAGES OF ACL's:

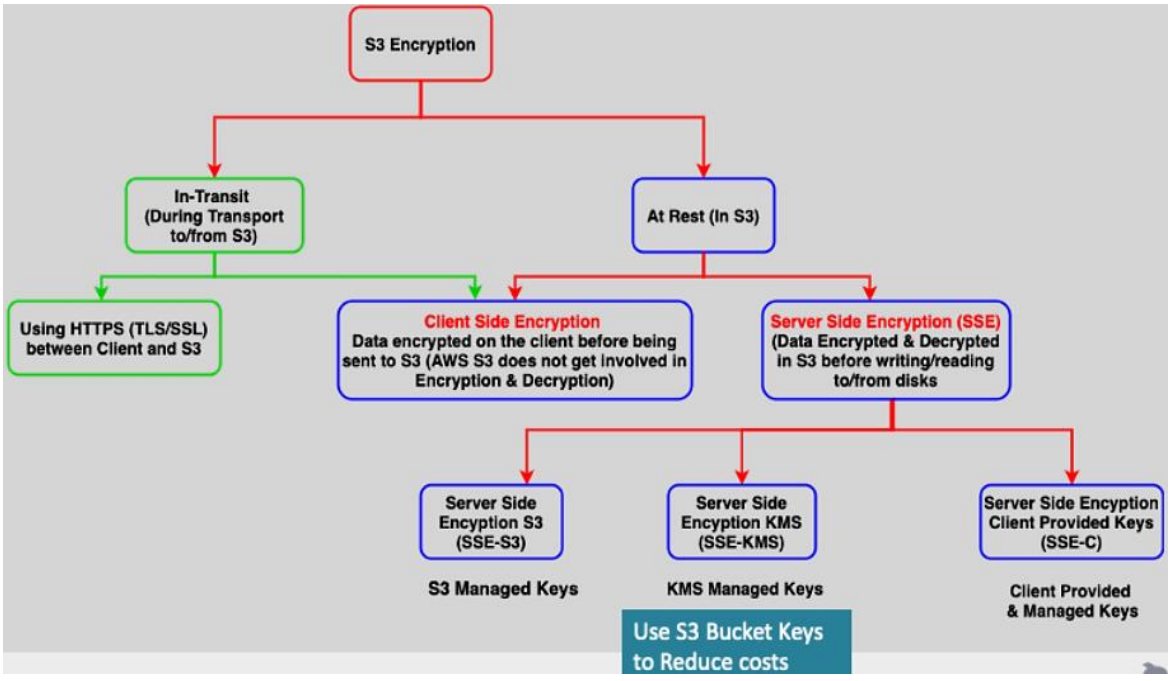
- We can not define condition statements like in IAM policies.
- ACL's do not support the full set of operations on S3.

➢ S3 SECURITY:

- Amazon S3 offers flexible security features to block unauthorized users from accessing your data.
- S3 allows you to block public access to all of your objects at the bucket or the account level with S3 Block Public Access.
- Store your data in S3 and secure it from unauthorized access with encryption features and access management tools.
- S3 Security Features are:
  - S3 Block Public Access
  - S3 Object Ownership
  - IAM Access Analyzer for S3
  - AWS PrivateLink for S3
  - Encryption
- Bucket access controlled by:
  - Bucket policies
  - Access Control Lists
- Object access controlled by ACL.

S3 ENCRYPTION:

- Encryption is one of the most basic requirements for ensuring data privacy, especially for end-to-end protection of data transmitted across networks.
- Plain text is encrypted using an encryption algorithm and an encryption key.



- Amazon S3 supports both server-side encryption (with three key management options) and client-side encryption for data uploads.
- Use S3 Inventory to check the encryption status of your S3 objects (see storage management for more information on S3 Inventory).
- Encryption or data protection in S3 is about encrypting data **in-transit** or **at rest** in S3

IN-TRANSIT: (CLIENT-SIDE-ENCRYPTION)

- It can be done either by client-side encryption, where the client encrypts data before sending it to S3, which means data is encrypted in Transit. Then when the data arrives at S3, S3 will save it to disk, which means it will be encrypted at rest as well.
- Or we can do encryption In-Transit using TLS/SSL or HTTPS sessions between the client and S3 endpoints.

AT REST (IN S3): (SERVER-SIDE-ENCRYPTION)

- Server-side encryption protects data at rest. Amazon S3 encrypts each object with a unique key.
- SSE is divided into the following:

  SSE-S3:
  - In this, the key material and the key will be provided by AWS itself to encrypt the objects in the S3 bucket. S3 Managed Keys-check box implemented, object-level keys, master key rotation.

  SSE-KMS:
  - In this, key material and the key will be generated in AWS KMS service to encrypt the objects in S3 bucket.
  - Use of envelope to retrieve private key, keys under customer control

  SSE-C:
  - In this, the key will be provided by the customer and Amazon S3 manages the encryption and decryption process while uploading/downloading the objects into the S3 bucket.

S3 DEFAULT BUCKET ENCRYPTION:

- To set the default encryption behaviour on an Amazon S3 bucket, all objects are encrypted when they are stored in the bucket.
- The objects are encrypted using server-side encryption with either Amazon S3-managed keys or AWS Key Management Service keys.
- Default encryption works with all existing and new Amazon S3 buckets.

➢ BUCKET PROPERTIES:

VERSIONING:

- To keep multiple variants of an object in same bucket.
- With S3 Versioning feature to preserve, retrieve, and restore every version of every object stored in your buckets.
- Versioning-enabled buckets can help you recover objects from accidental deletion or overwrite.
- Once enabled, can only be suspended, not disabled.
- Buckets can be in one of three states:
  - Unversioned (the default)
  - Versioning-enabled
  - Versioning-suspended

TAGS:

- A tag is a key-value pair that represents a label that you assign to a bucket.
- With AWS cost allocation, you can use bucket tags to annotate billing for your use of a bucket.

DEFAULT ENCRYPTION:

- Enabling default encryption provides you with automatic server-side encryption.
- S3 encrypts an object before saving it to a disk and decrypts the object when you download it.

SERVER ACCESS LOGGING:

- It provides detailed records for the requests that are made to a bucket.
- These logs can be useful for many applications for security purpose or access auditing.
- To specify the source bucket and the target bucket where the logs will be delivered.
- By default, Amazon S3 doesn't collect server access logs.

CLOUDTRAIL DATA EVENTS:

- CloudTrail is the AWS API **Auditing Service**.
- Use CloudTrail to log data events. By default, trails don't log data events.
- Additional charges apply for data events.

EVENT NOTIFICATIONS:

- Receive notifications when specific events occur in your bucket.
- Notifications typically deliver events in seconds but sometimes take a minute or longer.

TRANSFER ACCELERATION:

- Enable fast, easy, and secure transfers of files over long distances between your client and an S3 bucket.
- Offloads object transfer (in/out) from bucket region location to AWS **Edge Location.**

OBJECT LOCK:

- Use S3 Object Lock to prevent an object from being deleted or overwritten for a fixed amount of time or indefinitely.

REQUESTER PAYS:

- In general, bucket owners pay for storage and data transfer (download) charges.
- Configure the bucket to be a “requester pays” bucket if the bucket owner would like to share large data sets but does not want to incur the read/download charges.
- Bucket owner continuous to pay for storage, but the requester pays for data transfer.
- Anonymous access to requester pays buckets is not allowed. The requester must be authenticated for billing purposes.

STATIC WEBSITE HOSTING:

- S3 buckets can be used to host static content (not dynamic) websites.
- Content can be HTML pages, images, videos, client-side scripts such as java scripts.
- The web site S3 end point URL has the format http://bucket-name.S3-website-AWS-Region.amzon.com
- Requires **no virtual machine/instance**.
- S3 hosted static web sites can enable redirection for the whole domain, pages within a domain, or specific objects.

➢ S3 MANAGEMENT:

LIFECYCLE RULES:

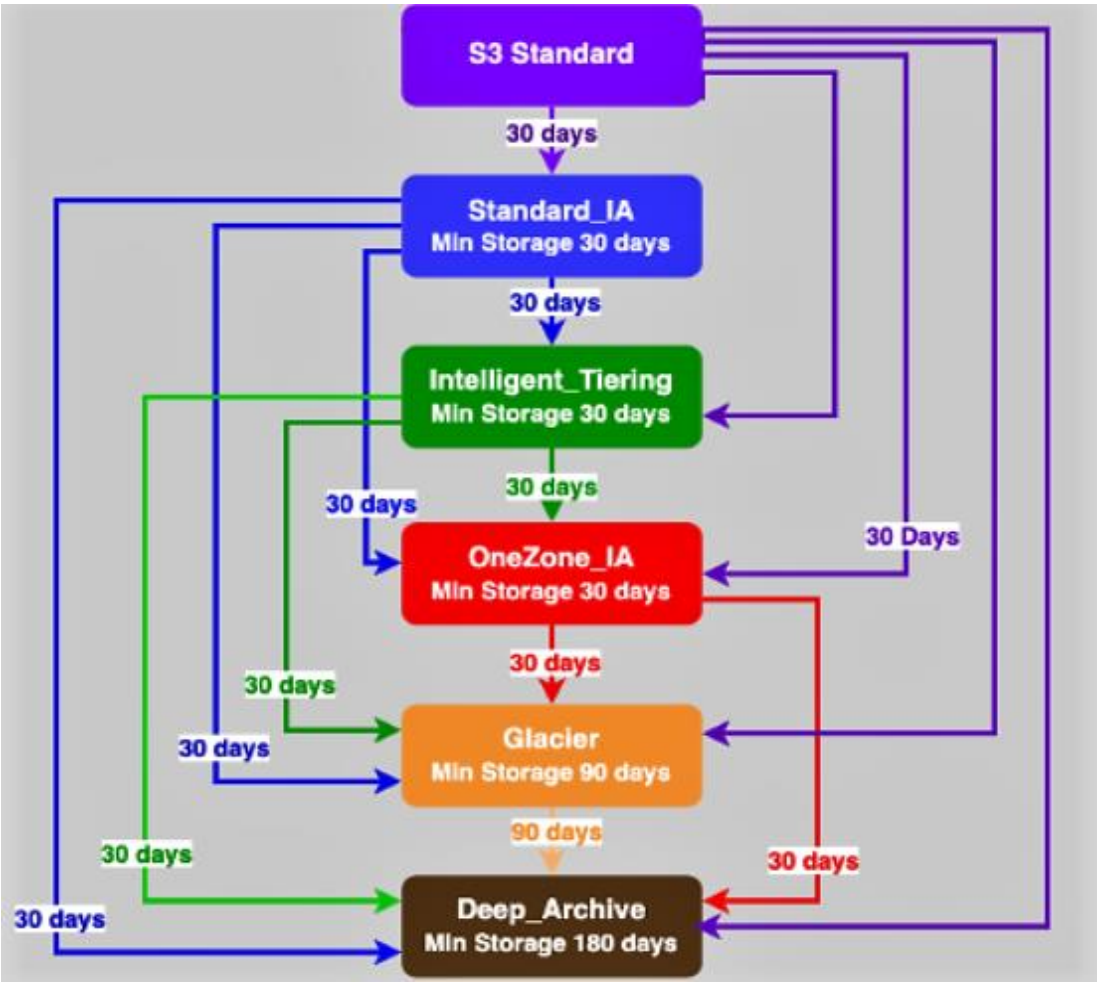
- An object lifecycle policy is a set of rules that automate the **migration** of an objects storage class to a different storage class, based on specific **Time Intervals**.
- Minimum time interval **30 days**.
- There are two types of actions:

TRANSITION ACTIONS:

- These actions define when objects transition to another storage class.

EXPIRATION ACTIONS:

- These actions define when objects expire. Amazon S3 deletes expired objects on your behalf.
- Lifecycle expiration costs depend on when you choose to expire objects.



REPLICATION RULES:

- **Automatic** and **Asynchronous** copying of objects across buckets in different AWS regions or accounts.
- We can replicate all objects, or a subset based on a prefix.
- The data pre-fixing in the bucket is not copied, we can use aws s3 sync command to copy it.
- Replication between buckets require versioning to be enabled.
- TLS/SSL is used to encrypt replication data in-transit.

➢ S3 ACCESS POINTS:

- Access points are unique hostnames that customers create to enforce distinct permissions and network controls for any request made through the access point.
- These are named network endpoints that are attached to buckets that you can use to perform S3 object operations, such as **GetObject** and **PutObject**.