What is Amazon S3?

Amazon S3 is a web service that is used to store and retrieve unlimited data anywhere over the internet. It is similar to Google Drive and is probably the best storage option under AWS. It is mostly used for-

- Static web content and media.
- Hosting entire static websites.
- Data storage for large-scale analytics.
- Backup and archival of critical data.
- Disaster recovery solutions for business continuity.

AWS Storage Services

The need for storage is increasing day by day. To fulfill that need Amazon offers a total of five main storage options; they are:

- AWS Import/Export
- Amazon Glacier.
- AWS Storage Gateway.
- Amazon Elastic Block Store (Amazon EBS)
- Amazon Simple Storage Service (Amazon S3).

What is an Amazon S3 Bucket?

Amazon S3 has two basic entities called Object and Bucket, where objects are stored inside buckets. By default, one can create 100 buckets per account. In case of more bucket demands, one can submit the request to increase the limit. Bucket names should be globally unique irrespective of the region. Every bucket has its data and descriptive metadata.

Let’s have a look at the basic concepts of Amazon S3.

How Does Amazon S3 Work?

Amazon S3 offers an object storage service where each object is stored as a file name With a given ID number and metadata. Unlike file and block cloud storage, Amazon S3 allows a developer to access an object via a REST API.

There are two types of metadata in S3

- System Defined
- User-Defined

The system defined is used for maintaining things such as creation date, size, last modified, etc. whereas the user-defined system is used to assign key values to the data that the user uploads. Key-value helps users to organize objects and allows easy retrieval. S3 allows users to upload, store and download files having sizes up to five terabytes.

What are the Important Features of Amazon S3?:

Write, read, and delete unlimited objects containing from 1 byte to 5 terabytes of data.

- Each object is stored in a bucket and accessed via a unique, user-assigned key.
- Objects stored by the user at a specific region never leave the location unless he/she transfer it out.
- Objects can be made private or public, and rights can be granted to specific users.
- Uses standards-based REST and SOAP interfaces that can work with any Internet-development toolkit.
- The default download protocol is HTTP. AWS CLI and SDK operate on HTTPS connections by default.
- Provides functionality to divide data by buckets, supervise and control spend, and automatically archive data to even lower-cost storage options for better manageability of data through its lifetime.

Data Consistency Models

S3 in Amazon provides high availability and durability solutions by replicating the data of one bucket in multiple data centers. As told earlier, the Amazon S3 bucket never leaves its position until a user moves it or deletes it. Consistency is an important part of data storage; it ensures that every change committed to a system should be visible to all the participants. S3 has two types of consistency models-

- read-after-write consistency
- Eventual consistency

1) **Read-after-write consistency:** It enables the visibility of a newly created object to all clients without any delays. Similarly, there are read-after-delete and read-after-update. In read-after-update, the user can edit or make changes to an already existing object whereas read-after-delete guarantees that reading a deleted file or object will fail for all clients.
2) **Eventual consistency:** there is a time lag between the changes made in the data to the point where all participants can see it. It might not be visible immediately, but eventually, it appears.

What are the Storage Classes in Amazon S3?

S3 Standard

Standard storage class is a default storage class in S3 that stores the user's data across multiple devices. It provides 99.99% availability and 99.999999999% durability. It can save the loss of two facilities concurrently and provide low latency and high throughput performance.

S3 Standard IA (infrequently accessed)

This class is used when data is not accessed frequently but demands rapid access when needed. It is also designed to sustain the loss of two facilities concurrently and provide low latency and high throughput performance.

S3 one zone - infrequent access

This storage class is used by the user when data is accessed less frequently, but when it requires fast access when needed. It cost 20% less than the standard IA storage class because it stores data in a single availability zone, unlike all other storage classes. It is cost-effective storage and a good choice for storing backup data.

S3 Glacier

It is the cheapest storage class in Amazon S3 where you can store immense data at a lower rate as compared to other storage classes but can be used for archives only. Let us see what the three types of models offered by S3 Glacier are

- **Expedited:** data is stored only for a few minutes
- **Standard:** Retrieval time of standard model is 3 to 5 hours
- **Bulk:** Retrieval time of bulk model is 5 to 12 hours

Amazon S3 Object Lifecycle Management

Every user has to pay a monthly monitoring and automation fee for storing objects in the S3 buckets. The rate charged for that depends on the object's size, storage class, and duration of storage. Proper object lifecycle management and configuration are very necessary if you want to get a cost-effective deal. With lifecycle configuration rules, users can tell Amazon S3 to place data in a less expensive storage class, or archive or delete them permanently. Let us find out when one should use a lifecycle configuration.

If a user uploads periodic logs to a bucket, the application might require that data for a week or a month. After that, a user might want to delete them.

Some data is accessed frequently for a specific period of time. After that, they are rarely needed. At that point, the user would like to archive it for some particular time and then delete it permanently. Amazon S3 provides a set of API operations to manage lifecycle configuration on a bucket. Check out the operations below-

- PUT Bucket lifecycle
- GET Bucket lifecycle
- DELETE Bucket lifecycle

What is Object Versioning?

Object Versioning is one of the most salient features of Amazon S3 and is used to keep multiple versions of data at the same time in the bucket. It is used to avoid accidental or unplanned overwrite and deletion of data. Object versioning is not a defaulted feature, but the user has to enable it.

Once it is enabled, a user cannot delete any object directly. All versions of the data reside in the bucket, and a delete marker is introduced in the bucket which becomes the current version. Now, to delete the object, the user needs to remove that delete marker also. Note that existing objects in your bucket are not affected by this operation; only future requests behavior will change.

How to Secure Your Data Using Amazon S3 Encryption?

A user cannot afford to lose his/her data stored on the cloud. S3 is enriched with great features, and one of them is default encryption. Protecting data while in transition mode (as it travels to or from Amazon S3) or stored on disks in Amazon S3, a user needs to set default encryption on a bucket. There are two ways in which you can encrypt the data - client-side encryption and server-side encryption.

In client-side encryption, a user encrypts the data using the KMS (key management service) and then transfers it to the S3. In this case, S3 cannot see the raw data. In server-side encryption, the user transfers the data to S3 where it is encrypted. When the user retrieves data, AWS decrypts the data and sends the raw data back.

How to Get Started with Amazon S3?

STEP-1: Create an S3 Bucket

A bucket can be created using AWS Command Line Interface or logging into the AWS Management Console. By default, 100 buckets can be created but can be extended with a request. Go to the Amazon S3 console and click "Create bucket". Follow the bucket naming rule to give the globally unique name to the bucket and click and "create". Also, choose the configure option and set permission as per your need.

STEP-2: Configure Options

Here, you will be given various configure options that you can select to enable a particular set of features on a bucket such as

- **Versioning:** Enabling this feature will help you track each version of the file and make it easier to recover the file after accidental deletion.
- **Server Access Logging:** It helps in carrying out all activities and requests from one bucket to another bucket.
- **Tags:** It is easy to search the resources with the tags. Therefore, tag the bucket with the key and name.
- **Object-Level Logging:** Activating this feature will help you record each and every action of objects in the bucket.
- **Default Encryption:** Enabling this feature will allow AWS to encrypt data in the bucket and protect it from being accessed by unauthorized people.
  STEP- 3: Set Permissions

By default, permission is private, which can be changed through AWS Management Console Permission or bucket policy. While granting permissions to read, write and delete, be selective and avoid keeping buckets open to the public.

Why Use AWS S3 Transfer Acceleration?

Amazon S3 Transfer acceleration promotes fast and secure data transfer from client to S3 bucket. You may need to use this for various possible reasons such as-

- If you have customers all over the world and they upload their data to a centralized bucket
- If you have to send terabytes of data daily across the continents
- If you are not able to use available internet bandwidth when uploading data to Amazon S3

If you are facing such needs, you should definitely start using Amazon S3 transfer acceleration. Let us find out how can you enable this feature and make the most out of it. Enable transfer acceleration on a bucket by

- Using the Amazon S3 console
- Using the REST API PUT Bucket Accelerate option
- Through AWS CLI and AWS SDKs

After enabling it, you can transfer the data from the bucket through the names of the S3 accelerate endpoint domain.

What Are the Benefits of Amazon S3?

- Industry-leading performance, scalability, availability, and durability

S3

It is the best and cost-effective cloud storage platform that handles the fluctuating storage demands of the user. It is enriched with amazing features and offers data durability along with scalability, availability, and industry-leading performance.

- Unmatched security compliance and audit capabilities

  Its encryption feature ensures data protection from unauthorized access. It is the only object storage service that gives a block option to public access at the bucket or at the account level. AWS also has various auditing capabilities to handle and maintain access requests for S3 resources.

- Wide range of cost-effective storage classes

  AWS gives you several options to store and move your data to a lower-cost storage class as per access patterns. Preferring this platform will help save costs without sacrificing the performance of operation done on data as per the requirement.

- Easily manage data and access control

  With Amazon S3, you can easily manage access, cost, and data protection. AWS lambda helps you to log activities, automate workflows without any additional infrastructure. Use this cloud storage platform to manage data operations with specific permissions for your application.

- Query-in-place services for analytics

  Analyze data stored in AWS data warehouse and S3 resources through standard SQL expressions. Also, improve query performance by retrieving the needed set of data instead of the entire object. Do use this data storage platform to manage your data properly and perform effective operations on data.

How do I control the right of entry to an S3 bucket?

Following are some of the now not unusual place location strategies for coping with getting proper access to an S3 bucket:

- **S3 Access Points:** S3 provides predefined buckets to a particular application. You can get proper access to elements that we are able to use to control get proper access to S3 datasets.

- **S3 Bucket Policies:** To get proper access to S3 you can config the S3 bucket. We can also configure permissions at the bucket diploma which are the most effective exercise for gadgets inner a bucket.

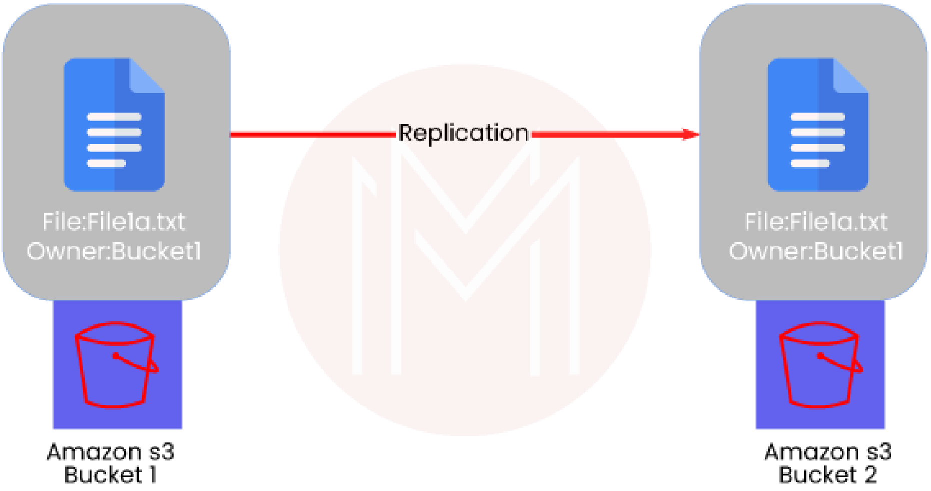
S3

- **ACL:** You may utilize Access Control List (ACL) to control get proper access to S3 assets and gadgets inside a bucket.
- **IAM:** We can use AWS Identity and Access Management (IAM) Groups, Roles, and Users to control get proper access to S3 assets.

5. What Is AWS S3 Replication?

When we need to replicate gadgets asynchronously, we use S3 replication at a few degrees within-side the AWS S3 buckets. This can be auto stretch, actually in minimum budget.

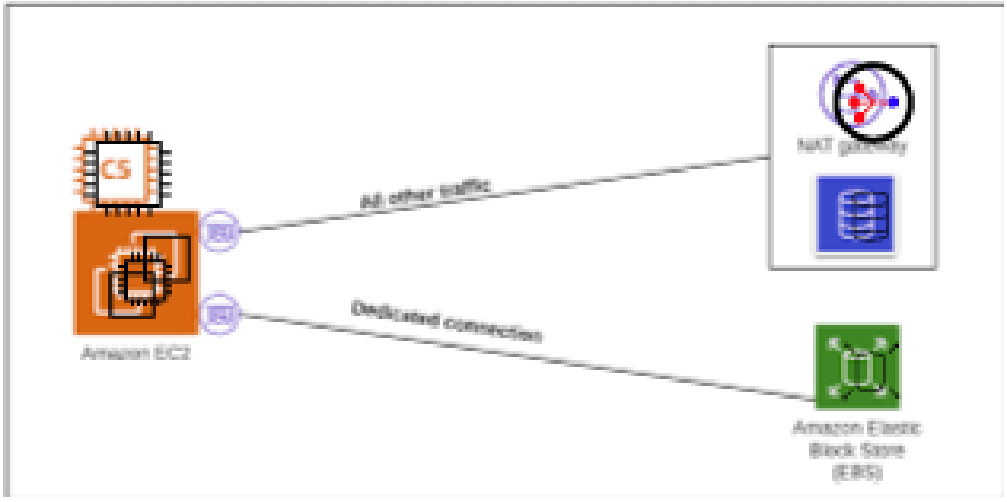
We have given data related to records sovereignty and agency goals, it offers us the authority to manipulate those data.



6. How does Elastic Block Store work?

Elastic Block Store (EBS) is a slab-diploma cache of Amazon Web Services (AWS).

It helps to find out the stored data, because of this the records will be saved on EBS irrespective of the recognition of EC2 instances. With ECE instances you can have a lot of space.



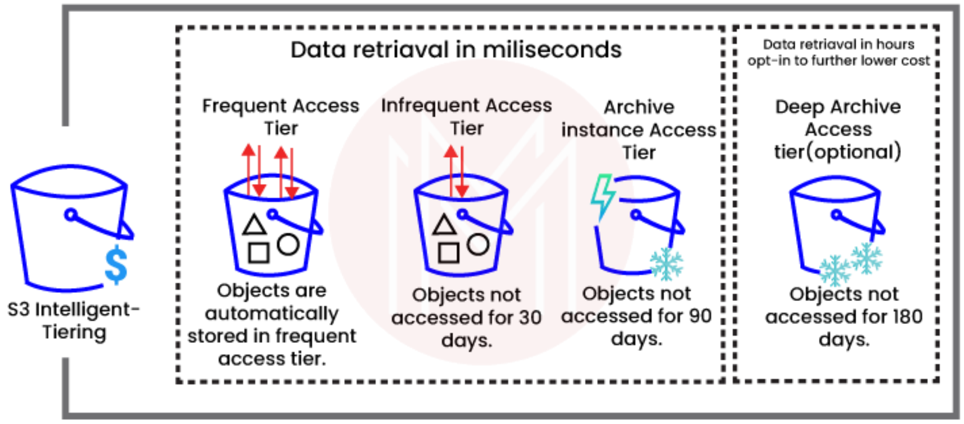
7. Write down the Differences Between S3 And EBS?

The differences between Amazon S3 and EBS are given below:

S3	EBS
Entity can be stored.	EC2 Instances file management tool.
Data security is high.	Data security is fewer.
Data centres using Redundancy.	Redundancy can be used by only one data centre.

8. What do you know about S3 Intelligent Tier?

One of the S3 storage facilities that let customers hold costs via manner of the technique of transferring gadgets as consistent with the get proper of access to the volume amongst S3 now not unusual place to get proper of access to and IA.


S3 Intelligent-Tiering
Data retrieval in miliseconds
Frequent Access Tier
Objects are automatically stored in frequent access tier.
Infrequent Access Tier
Objects not accessed for 30 days.
Archive instance Access Tier
Objects not accessed for 90 days.
Data retrieval in hours opt-in to further lower cost
Deep Archive Access tier(optional)
Objects not accessed for 180 days.

How to restrict a third party user to access all the buckets and list only a bucket having an access

To restrict a third-party user from viewing all your S3 buckets and only allowing them to access a specific bucket (or set of objects within that bucket), you need to implement bucket policies or IAM policies that define granular permissions.

Here’s how you can achieve this:

1. **Use a Bucket Policy to Restrict Access**

You can attach a bucket policy to the specific S3 bucket that allows access to only the third-party user and denies access to other buckets. Here’s an example of a bucket policy that grants the third party (with a specific AWS account ID) permission to access only `my-bucket`.

Example Bucket Policy:

S3

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::THIRD_PARTY_ACCOUNT_ID:root"
      },
      "Action": [
        "s3:ListBucket",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::my-bucket",
        "arn:aws:s3:::my-bucket/*"
      ]
    }
  ]
}
```

Explanation:

- **Denying List of All Buckets:**
  - The first statement denies the `s3:ListAllMyBuckets` action, which restricts the third party from seeing all buckets in your account.

S3

  - `"Effect": "Deny"` is applied to any entity (via `"Principal": "*"`), preventing the listing of all S3 buckets.

- **Allowing Specific Access:**

  - The second statement allows the third party (replace `THIRD_PARTY_ACCOUNT_ID` with their actual account ID) to list the contents of `my-bucket` and get objects within that bucket.

  - `"Resource": "arn:aws:s3:::my-bucket"` allows them to list the bucket.

  - `"Resource": "arn:aws:s3:::my-bucket/*"` grants access to the objects within the bucket.

2. **Use an IAM Policy for the Third Party**

If the third party uses an IAM user or role within your AWS account, you can also attach an IAM policy to the user/role that restricts access to only the specific bucket.

Example IAM Policy for Restricting Access:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::my-bucket",
```

S3

        "arn:aws:s3:::my-bucket/*"
      ]
    }
  ]
}
~~~

Explanation:

- **Deny Access to All Buckets:**

  - This policy first denies the `s3:ListAllMyBuckets` permission, which means the user cannot see any buckets except the ones explicitly allowed.

- **Allow Access to a Specific Bucket:**

  - The second statement allows the IAM user or role to list the objects in the specific bucket (`my-bucket`) and access its contents.

3. **Using an External ID for Cross-Account Access**

If the third party is from a different AWS account, you should use a cross-account role with an external ID to enhance security.

Here's an example of a trust policy that you would attach to an IAM role in your AWS account, which allows a specific third-party AWS account to assume the role and access the S3 bucket:

Trust Policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::THIRD_PARTY_ACCOUNT_ID:root"
~~~

```
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "EXTERNAL_ID_VALUE"
        }
      }
    }
  ]
}
```

In this case, the third party can assume the role using the external ID and gain access to the resources in the bucket as defined in the role's permission policy.

Steps to Set Up:

1. **Set up an IAM role in your AWS account** that the third party can assume.

2. **Attach the bucket policy or IAM policy** to restrict access to only the specified bucket.

3. If it’s a cross-account scenario, ensure the third party assumes the role using the external ID (recommended for security).

By following these steps, you can effectively restrict access to your other buckets and grant third-party users access to only the necessary S3 bucket.