CLOUD

"The cloud" refers to servers that are accessed over the Internet, and the software and databases that run on those servers. Cloud servers are located in data center all over the world. By using cloud computing, users and companies do not have to manage physical servers themselves or run software applications on their own machines.

CLOUD STORAGE

Cloud storage is a virtual locker where we can remotely stash any data. When we upload a file to a cloud-based server like Google Drive, OneDrive, or iCloud that file gets copied over the Internet into a data server that is **cloud-based** actual physical space where companies store files on multiple hard drives. Most companies have hundreds of the ese servers known as 'server farms' spanning across multiple locations. So, if our data gets somehow lost we will not lose our

Features of Cloud Storage System:

Security is one of the major components and using cloud computing you can secure all over the networks. The key features of cloud computing are as follows.

- It has a greater availability of resources.
- Easy maintenance is one of the key benefits of using Cloud computing.
- Cloud computing has a Large Network Access.
- It has an automatic system.

TYPES OF CLOUD STORAGE

- Block-Based Storage System
- File-Based Storage System
- Object-Based Storage System

Let's discuss it one by one as follows.

1. Block-Based Storage System –

- Hard drives are block-based storage systems. Your operating system like Windows or Linux actually sees a hard disk drive. So, it sees a drive on which you can create a volume, and then you can partition that volume and format them.
- For example, If a system has 1000 GB of volume, then we can partition it into 800 GB and 200 GB for local C and local D drives respectively.
- Remember with a block-based storage system, your computer would see a drive, and then you can create volumes and partitions.

2. File-Based Storage System –

- In this, you are actually connecting through a . You are going over a network, and then you can access the network-attached storage server (NAS). NAS devices are file-based storage systems.
- This storage server is another computing device that has another disk in it. It is already created a file system so that it's already formatted its partitions, and it will share its file systems over the network. Here, you can actually map the drive to its network location.
- In this, like the previous one, there is no need to partition and format the volume by the user. It's already done in file-based storage systems. So, the operating system sees a file system that is mapped to a local drive letter.

3. Object-Based Storage System –

- In this, a user uploads objects using a web browser and uploads an object to a container i.e., Object Storage Container. This uses the HTTP Protocols with the rest of the APIs (for example: GET, PUT, POST, SELECT, DELETE).
- For example, when you connect to any website, you need to download some images, text, or anything that the website contains. For that, it is a code HTTP GET request. If you want to review any product then you can use PUT and POST requests.
- Also, there is no hierarchy of objects in the container. Every file is on the same level in an Object-Based storage system.

S3

S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can use Amazon S3 to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.

Amazon 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.

S3 durability and redundency

Amazon S3 provides the most durable storage in the cloud. Based on its unique architecture, S3 is designed to exceed 99.999999999% (11 nines) data durability.

S3 stores data redundantly across a minimum of 3 Availability Zones by default, providing built-in resilience against widespread disaster.

S3 Buckets

A bucket is a container for objects stored in Amazon S3. You can store any number of objects in a bucket and can have up to 100 buckets in your account.

When you create a bucket, you enter a bucket name and choose the AWS Region where the bucket will reside. After you create a bucket, you cannot change the name of the bucket or its Region. Bucket names must follow the Bucket naming rules

Buckets also:

- Organize the Amazon S3 namespace at the highest level.
- Identify the account responsible for storage and data transfer charges.
- Provide access control options, such as bucket policies, access control lists (ACLs), and S3 Access Points, that you can use to manage access to your Amazon S3 resources.
- Serve as the unit of aggregation for usage reporting.

S3 UPLOADING DOWNLOADING

S3 uploading

When you upload a file to Amazon S3, it is stored as an S3 *object*. Objects consist of the file data and metadata that describes the object. You can have an unlimited number of objects in a bucket. Before you can upload files to an Amazon S3 bucket, you need write permissions for the bucket. For more information about access permissions.

You can upload any file type—images, backups, data, movies, and so on—into an S3 bucket. The maximum size of a file that you can upload by using the Amazon S3 console is 160 GB. To upload a file larger than 160 GB, use the AWS Command Line Interface (AWS CLI), AWS SDKs, or Amazon S3 REST API.

If you upload an object with a key name that already exists in a versioning-enabled bucket, Amazon S3 creates another version of the object instead of replacing the existing object.

Depending on the size of the data that you're uploading, Amazon S3 offers the following options:

Upload an object in a single operation by using the AWS SDKs, REST API, or AWS CLI – With a single PUT operation, you can upload a single object up to 5 GB in size.

Upload a single object by using the Amazon S3 console – With the Amazon S3 console, you can upload a single object up to 160 GB in size.

Upload an object in parts by using the AWS SDKs, REST API, or AWS CLI – Using the multipart upload API operation, you can upload a single large object, up to 5 TB in size.

The multipart upload API operation is designed to improve the upload experience for larger objects. You can upload an object in parts. These object parts can be uploaded independently, in any order, and in parallel. You can use a multipart upload for objects from 5 MB to 5 TB in size.

When you upload an object, the object is automatically encrypted using server-side encryption with Amazon S3 managed keys (SSE-S3) by default. When you download it, the object is decrypted.

When you're uploading an object, if you want to use a different type of default encryption, you can also specify server-side encryption with AWS Key Management Service (AWS KMS) keys (SSE-KMS) in your S3 PUT requests or set the default encryption configuration in the destination bucket to use SS objects

S3 Downloding

This section explains how to download objects from an Amazon S3 bucket. With Amazon S3, you can store objects in one or more buckets, and each single object can be up to 5 TB in size. Any Amazon S3 object that is not archived is accessible in real time. Archived objects, however, must be restored before they can be downloaded.

You can download a single object by using the Amazon S3 console, AWS Command Line Interface (AWS CLI), AWS SDKs, or Amazon S3 REST API. To download an object from S3 without writing any code or running any commands, use the S3 console.

To download multiple objects, use AWS CloudShell, the AWS CLI, or the AWS SDKs.

If you need to download part of an object, you use extra parameters with the AWS CLI or REST API to specify only the bytes that you want to download..

If you need to download an object that you don't own, ask the object owner to generate a presigned URL that allows you to download the object.

When you download objects outside of the AWS network, data-transfer fees apply. Data transfer within the AWS network is free within the same AWS Region, but you will be charged for any GET requests.

E-KMS to encrypt your data.

Policies and permissions in Amazon S3

This page provides an overview of bucket and user policies in Amazon S3 and describes the basic elements of a policy. Each listed element links to more details about that element and examples of how to use it.

In its most basic sense, a policy contains the following elements:

- Resource – The Amazon S3 bucket, object, access point, or job that the policy applies to. Use the Amazon Resource Name (ARN) of the bucket, object, access point, or job to identify the resource.

An example for bucket-level operations:

- "Resource": "arn:aws:s3:::*bucket_name*".

Examples for object-level operations:

- "Resource": "arn:aws:s3:::*bucket_name/**" for all objects in the bucket.

- "Resource": "arn:aws:s3:::*bucket_name/prefix/**" for objects under a certain prefix in the bucket..

- Actions – For each resource, Amazon S3 supports a set of operations. You identify resource operations that you will allow (or deny) by using action keywords.

For example, the s3:ListBucket permission allows the user to use the Amazon S3 operation.Effect – What the effect will be when the user requests the specific action—this can be either *allow* or *deny*.

If you do not explicitly grant access to (allow) a resource, access is implicitly denied. You can also explicitly deny access to a resource. You might do this to make sure that a user can't access the resource, even if a different policy grants access.

- Principal – The account or user who is allowed access to the actions and resources in the statement. In a bucket policy, the principal is the user, account, service, or other entity that is the recipient of this permission.
- Condition – Conditions for when a policy is in effect. You can use AWS-wide keys and Amazon S3-specific keys to specify conditions in an Amazon S3 access policy.

Objects versioning

Objects are the fundamental entities stored in Amazon S3. Objects consist of object data and metadata. The metadata is a set of name-value pairs that describe the object. These pairs 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 that the object is stored.

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. The combination of a bucket, object key, and optionally, version ID (if S3 Versioning is enabled for the bucket) uniquely identify each object. So you can think of Amazon S3 as a basic data map between "bucket + key + version" and the object itself.

Every object in Amazon S3 can be uniquely addressed through the combination of the web service endpoint, bucket name, key, and optionally, a version. For example, in the URL https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/photos/puppy.jpg, DOC-EXAMPLE-BUCKET is the name of the bucket and photos/puppy.jpg is the key.

Versioning

You can use S3 Versioning to keep multiple variants of an object in the same bucket. With S3 Versioning, you can preserve, retrieve, and restore every version of every object stored in your buckets. You can easily recover from both unintended user actions and application failures.

Version ID

When you enable S3 Versioning in a bucket, Amazon S3 generates a unique version ID for each object added to the bucket. Objects that already existed in the bucket at the time that you enable versioning have a version ID of null. If you modify these (or any other) objects with other operations, such as CopyObject and PutObject, the new objects get a unique version ID.

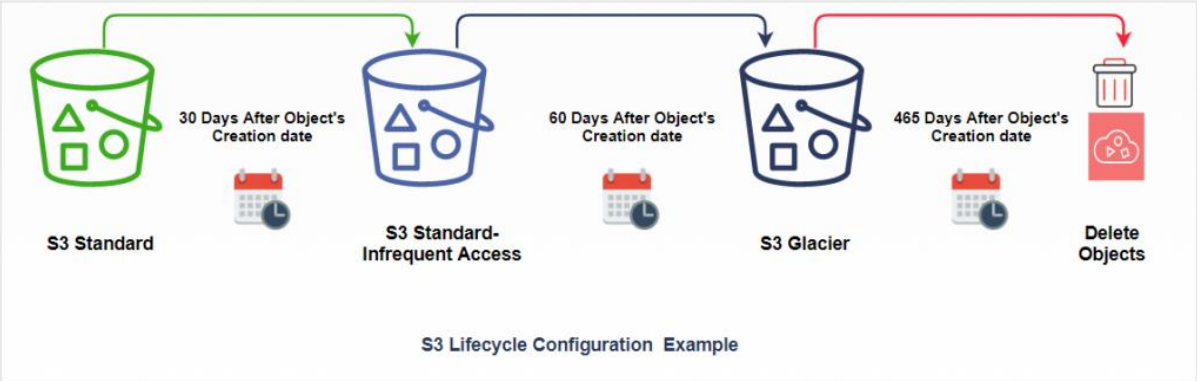
S3 LIFE CYCLE POLICY

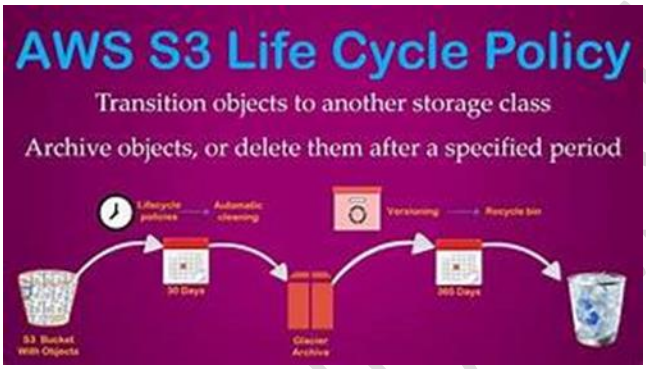
Lifecycle policy allow you to automatically review objects within your s3 buckets and have them moved to glacier or have the objects deleted from s3

To manage your objects so that they are stored cost effectively throughout their lifecycle, create an *Amazon S3 Lifecycle configuration*. An Amazon S3 Lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. There are two types of actions:

- **Transition actions** – These actions define when objects transition to another storage class. For example, you might choose to transition objects to the S3 Standard-IA storage class 30 days after creating them, or archive objects to the S3 Glacier Flexible Retrieval storage class one year after creating them.
- **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





COMPONENTS OF S3 LIFE CYCLE

ID element.

status element.

filter element.

elements to describe lifecycle actions.

STORAGE GATEWAY

AWS Storage Gateway is a service that connects an on-premises software appliance with cloud-based storage to provide seamless and secure integration between your on-premises IT

Depending on your use case, Storage Gateway provides three types of storage interfaces for your on-premises applications: file, volume, and tape.

The Amazon S3 File Gateway enables you to store and retrieve objects in Amazon Simple Storage Service (S3) using file protocols such as Network File System (NFS) and Server Message Block (SMB). Objects written through S3 File Gateway can be directly accessed in S3.

The Amazon FSx File Gateway enables you to store and retrieve files in Amazon FSx for Windows File Server using the SMB protocol. Files written through Amazon FSx File Gateway are directly accessible in Amazon FSx for Windows File Server.

The Volume Gateway provides block storage to your on-premises applications using iSCSI connectivity. Data on the volumes is stored in Amazon S3 and you can take point-in-time copies of volumes that are stored in AWS as Amazon EBS snapshots. You can also take copies of volumes and manage their retention using AWS Backup. You can restore EBS snapshots to a Volume Gateway volume or an EBS volume.

The Tape Gateway provides your backup application with an iSCSI virtual tape library (VTL) interface, consisting of a virtual media changer, virtual tape drives, and virtual tapes. Virtual tapes are stored in Amazon S3 and can be archived to Amazon S3 Glacier or Amazon S3 Glacier Deep Archive.

Storage Gateway supports four key hybrid cloud use cases – (1) move backups and archives to the cloud, (2) reduce on-premises storage with cloud-backed file shares, (3) provide on-premises applications low-latency access to data stored in AWS, and (4) data lake access for pre and post You have two touchpoints to use the service: the AWS Management Console and the gateway appliance.

You use the AWS Management Console to download the gateway virtual appliance or launch it as an EC2 instance, configure storage, and manage and monitor the service. The gateway connects your applications to AWS storage by providing standard storage interfaces. It provides transparent caching, efficient data transfer, and integration with AWS monitoring and security services.launch it as an EC2 instant.

IMPORT EXPORT

You can import data that's been stored using Amazon Simple Storage Service into a table on an RDS for PostgreSQL DB instance. To do this, you first install the RDS for PostgreSQL aws_s3 extension. This extension provides the functions that you use to import data from an Amazon S3 bucket. A *bucket* is an Amazon S3 container for objects and files. The data can be in a comma-separate value (CSV) file, a text file, or a compressed (gzip) file. Following, you can learn how to install the extension and how to import data from Amazon S3 into a table.

Your database must be running PostgreSQL version 10.7 or higher to import from Amazon S3 into RDS for PostgreSQL.

If you don't have data stored on Amazon S3, you need to first create a bucket and store the data. For more information, see the following topics in the *Amazon Simple Storage Service User Guide*.

EXPORT

In the following sections, you'll find information about your export delivery.

- **Export S3 parent directory structure:** How export data is structured in the S3 directory to which your export is delivered to.
- **Export refreshing:** How often your export updates in your S3 directory.
- **Export overwriting and create new:** How your export delivery changes with overwriting and creates new delivery preferences.
- **Export data file names and chunks:** How the export files (gzip/csv or Parquet) are named.

S3 TRANSFER ACCELARETION

Amazon S3 Transfer Acceleration can speed up content transfers to and from Amazon S3 by as much as 50-500% for long-distance transfer of larger objects. Customers who have either web or mobile applications with widespread users or applications hosted far away from their S3 bucket can experience long and variable Acceleration (S3TA) reduces the variability in Internet routing, congestion and speeds that can affect transfers, and logically shortens the distance to S3 for remote applications. S3TA improves transfer performance by routing traffic through Amazon CloudFront's globally distributed Edge Locations and over AWS backbone networks, and by using network protocol optimizations. You can turn on S3TA with a few clicks in the S3 console, and test its benefits from your location with . With S3TA, you pay only for transfers that are accelerated.

GLACIER STORAGE

The Amazon S3 Glacier storage classes are purpose-built for data archiving, providing you with the highest performance, most retrieval flexibility, and the lowest cost archive storage in the cloud. All S3 Glacier storage classes provide virtually unlimited scalability and are designed for 99.999999999% (11 nines) of data durability. The S3 Glacier storage classes deliver options for the fastest access to your archive data and the lowest-cost archive storage in the cloud.

You can choose from three archive storage classes optimized for different access patterns and storage duration. For archive data that needs immediate access, such as medical images, news media assets, or genomics data, choose the S3 Glacier Instant Retrieval storage class, an archive storage class that delivers the lowest cost storage with milliseconds retrieval. For archive data that does not require immediate access but needs the flexibility to retrieve large sets of data at no cost, such as backup or disaster recovery use cases, choose S3 Glacier Flexible Retrieval, with retrieval in minutes or free bulk retrievals in 5-12 hours. To save even more on long-lived archive storage such as compliance archives and digital media preservation, choose S3 Glacier Deep Archive, the lowest cost storage in the cloud with data retrieval within twelve hours.