**From:** Vernekar, Shantaram [CCC-OT NE]   
**Sent:** Tuesday, August 7, 2018 9:33 PM  
**To:** Shantaram Vernekar ([shantaram\_vernekar@persistent.com](mailto:shantaram_vernekar@persistent.com)) <[shantaram\_vernekar@persistent.com](mailto:shantaram_vernekar@persistent.com)>; Vernekar, Shantaram [CCC-OT NE] <[sv16217@imcnam.ssmb.com](mailto:sv16217@imcnam.ssmb.com)>  
**Subject:** AWS-S3

While creating a new bucket below information is needed:

Name and region - Bucket is created in a given region – Not global

Server access logging

* Provides detailed records for the requests that are made to your bucket

Versioning

Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite. For example:

* If you delete an object, instead of removing it permanently, Amazon S3 inserts a delete marker, which becomes the current object version. You can always restore the previous version.
* If you overwrite an object, it results in a new object version in the bucket. You can always restore the previous version.
* Objects stored in your bucket before you set the versioning state have a version ID of null. When you enable versioning, existing objects in your bucket do not change. What changes is how Amazon S3 handles the objects in future requests.
* The bucket owner (or any user with appropriate permissions) can suspend versioning to stop accruing object versions. When you suspend versioning, existing objects in your bucket do not change. What changes is how Amazon S3 handles objects in future requests.

**MFA Delete**

You can optionally add another layer of security by configuring a bucket to enable MFA (multi-factor authentication) Delete, which requires additional authentication for either of the following operations:

* Change the versioning state of your bucket
* Permanently delete an object version

MFA Delete requires two forms of authentication together:

* Your security credentials
* The concatenation of a valid serial number, a space, and the six-digit code displayed on an approved authentication device

A DELETE request has the following use cases:

* When versioning is enabled, a simple DELETE cannot permanently delete an object.

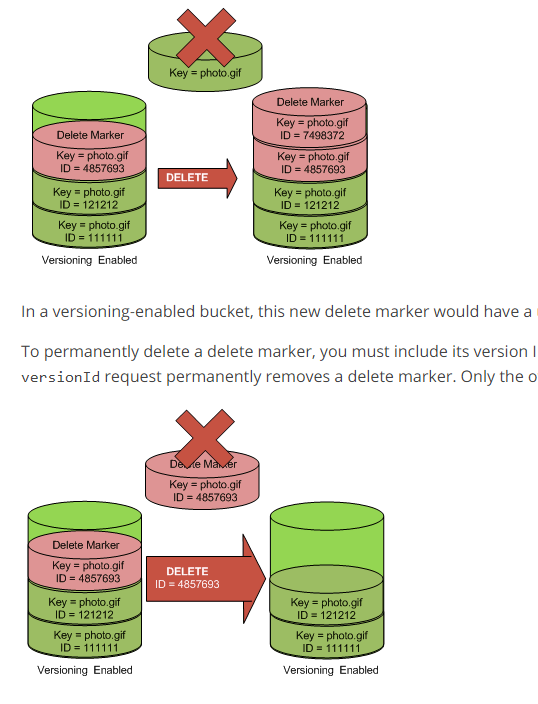
Instead, Amazon S3 inserts a delete marker in the bucket, and that marker becomes the current version of the object with a new ID. When you try to GET an object whose current version is a delete marker, Amazon S3 behaves as though the object has been deleted (even though it has not been erased) and returns a 404 error.

The following figure shows that a simple DELETE does not actually remove the specified object. Instead, Amazon S3 inserts a delete marker.



**To permanently remove a delete marker**

1. Set versionId to the ID of the version to the delete marker you want to remove.
2. Send a DELETE Object versionId request.



# Restoring Previous Versions

One of the value propositions of versioning is the ability to retrieve previous versions of an object. There are two approaches to doing so:

* Copy a previous version of the object into the same bucket

The copied object becomes the current version of that object and all object versions are preserved.

* Permanently delete the current version of the object

When you delete the current object version, you, in effect, turn the previous version into the current version of that object.

Tags – cost allocation

Object access logging

* Records object level API activity by using cloud trail data events

Encryption

SSE – Data encryption is done at server side

                While the data is saved, it is encrypted..

                While downloading data is decrypted..

**Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)**

**Key managed by s3, along with encryption and decryption**

**Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)**

            Key managed by KMS, **along with encryption and decyption**

**Use Server-Side Encryption with Customer-Provided Keys (SSE-C)**

**Key managed by client, along with encryption and decyption**

CSE – Data encryption and decryption is done at client side.

* Use an AWS KMS-managed customer master key
* Use a client-side master key

CloudWatch Request Matrics

Users – from current account and from another account..

Permissions – Public or not..

S3 log delivery group – write access

Amazon S3 uses a special log delivery account, called the Log Delivery group, to write access logs.

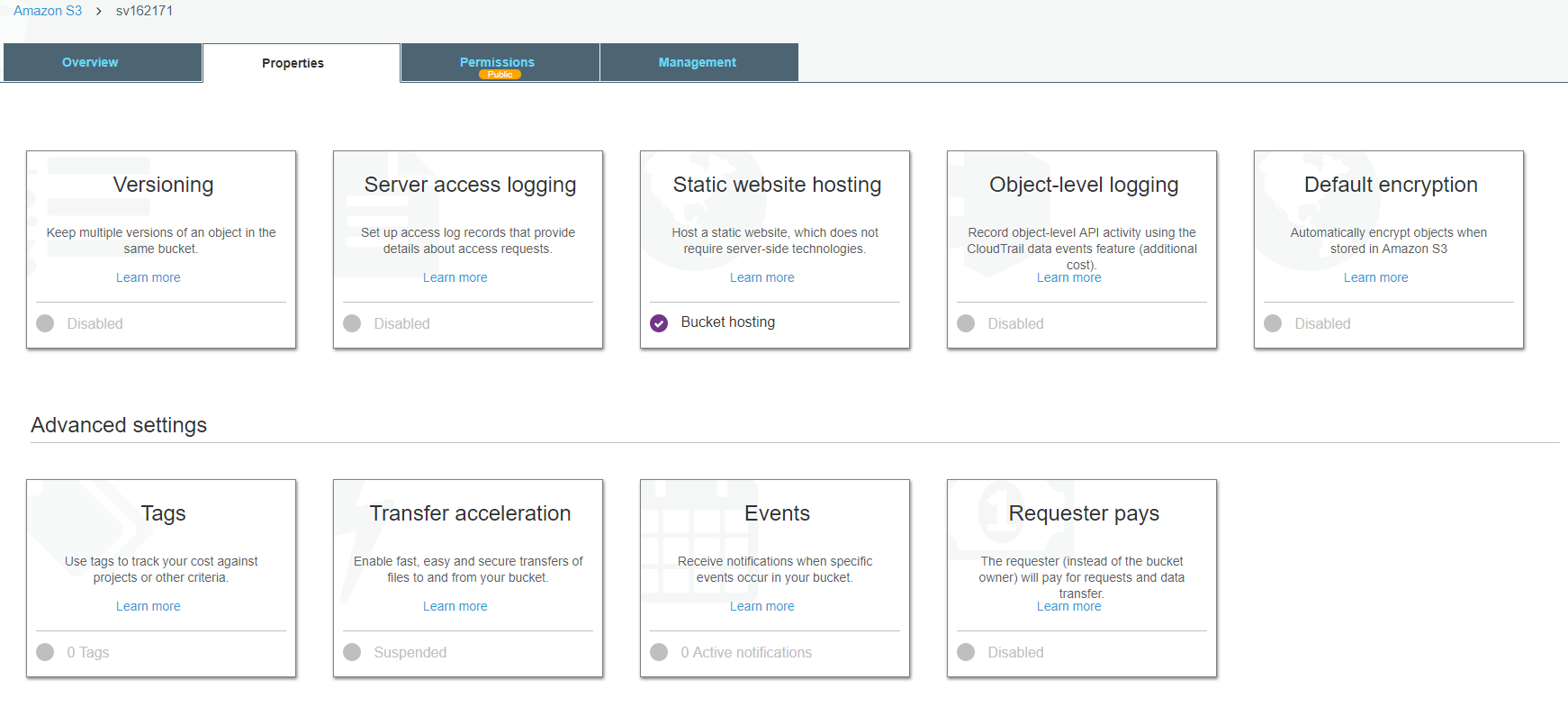
You must grant the Log Delivery group write permission on the target bucket by adding a grant entry in the bucket's access control list (ACL). If you use the Amazon S3 console to enable logging on a bucket, the console both enables logging on the source bucket and updates the ACL on the target bucket to grant write permission to the Log Delivery group.

ACL

Bucket Policy

CORS configuration

Bucket Properties



**Static web site hosting**

You can host a static website on Amazon Simple Storage Service (Amazon S3). On a static website, individual webpages include static content

The website is then available at the AWS Region-specific website endpoint of the bucket, which is in one of the following formats:

S3 website endpoint:

<bucket-name>.s3-website-<AWS-region>.amazonaws.com

<bucket-name>.s3-website.<AWS-region>.amazonaws.com

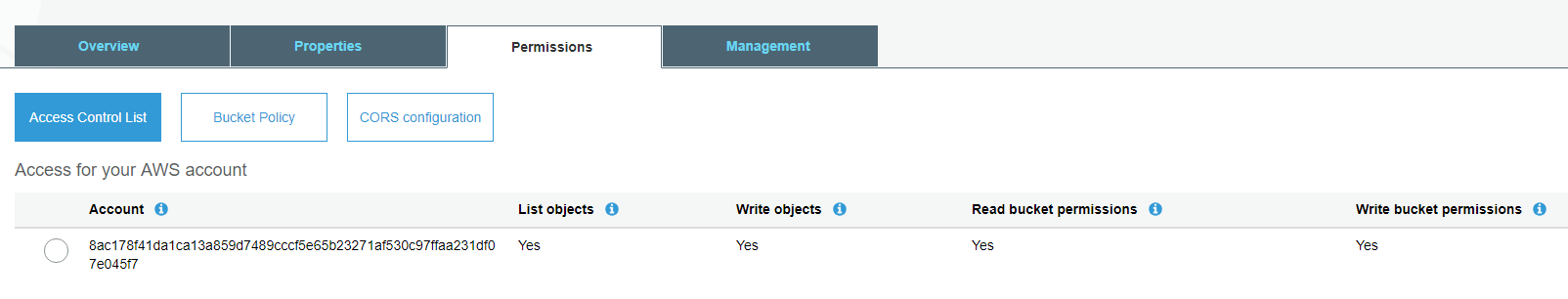
Transfer acceleration

Events

**Permission**

**ACL –**

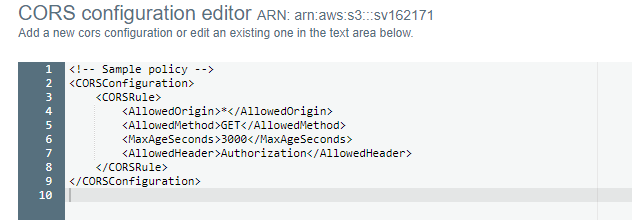
Which user has read & write permission to objects and bucket permission



**Cross origin resource sharing (CORS)**

Cross-origin resource sharing (CORS) defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. With CORS support, you can build rich client-side web applications with Amazon S3 and selectively allow cross-origin access to your Amazon S3 resources.

Default:



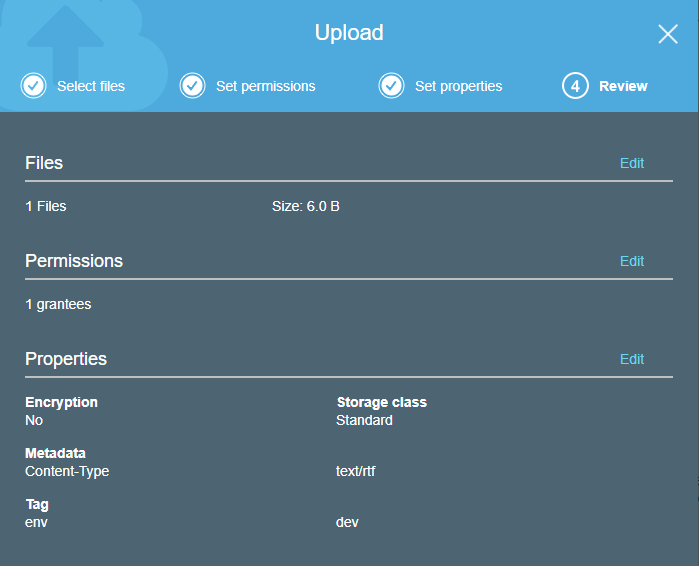
* The first rule allows cross-origin PUT, POST, and DELETE requests from the <http://www.example1.com> origin. The rule also allows all headers in a preflight OPTIONS request through the Access-Control-Request-Headers header. In response to preflight OPTIONS requests, Amazon S3 returns requested headers.
* The second rule allows the same cross-origin requests as the first rule, but the rule applies to another origin, <http://www.example2.com>.
* The third rule allows cross-origin GET requests from all origins. The \* wildcard character refers to all origins.



**Object level properties –**

**storage type is object level**

**meta data is object level**



Read and write consistency

**Consistency – Read after write consistency**

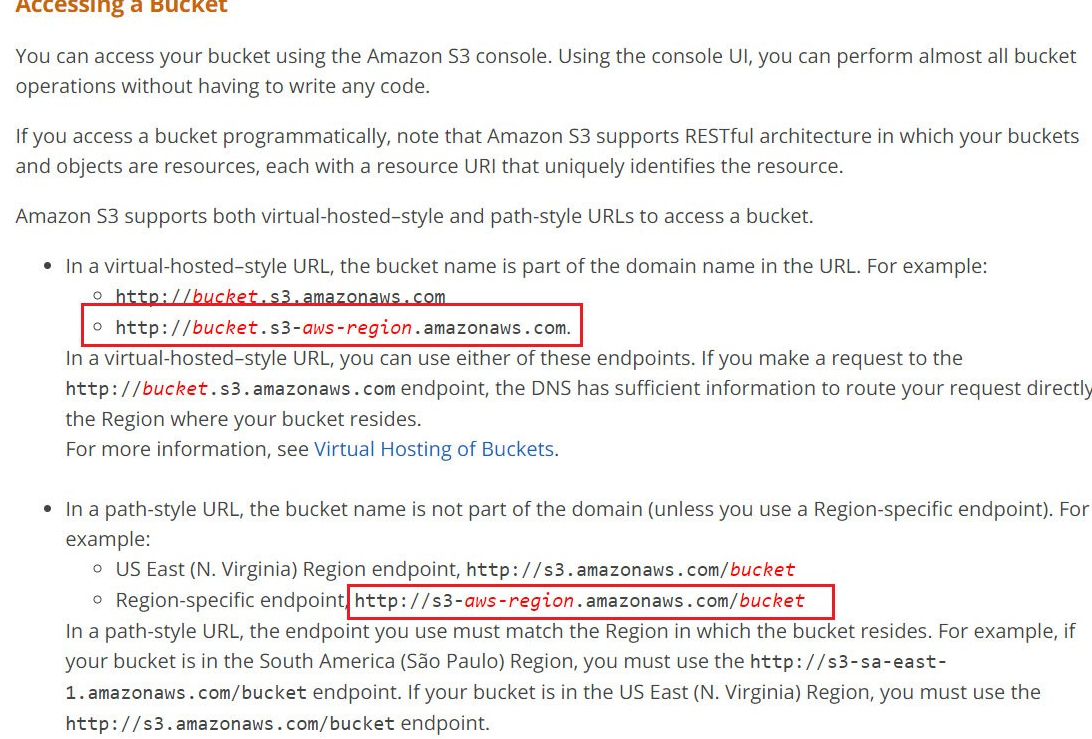
Delete – Eventual consistency

Modify – eventual consistency

**Accessing bucket**

Virtual hosted style URL – https://<bucket>.s3.awsregion.amazonaws.com

Path style URL -  <https://s3.awsregion.amazonaws.com/bucket>



Object deletion:

**Specify the version ID to delete for S3 to permanently delete the object.. Otherwise file is not delete and a delete marker flag is added to that file**

S3:

Amazon S3 provides developers and IT teams with secure, durable, and highly-scalable cloud storage.

Amazon S3 is easy-to-use *object storage* with a simple web service interface that you can use to store and retrieve any amount of data from anywhere on theweb.

Amazon S3 also allows you to pay only for the storage you actually use, which eliminates the capacity planning and capacity constraints associated with traditional storage.

Common use cases for Amazon S3 storage include:

* Backup and archive for on-premises or cloud data
* Content, media, and software storage and distribution
* Big data analytics
* Static website hosting
* Cloud-native mobile and Internet application hosting
* Disaster recovery

Glacier

optimized for data archiving and long-term backup at extremely low cost.

Amazon Glacier is suitable for “cold data,” which is data that is rarely accessed and for which a retrieval time of three to five hours is acceptable.

Block Storage v/s file storage

Block Storage – Example EBS (Elastic Block Storage)

Block storage operates at a lower level—the raw storage device level

Data is managed as set of numbered, fixed-size blocks

Data is accessed over a network in the form of a Storage Area Network (SAN) using protocols such as iSCSI or Fibre Channel

File Storage – Example EFS (Elastic File system)

            File storage operates at a higherlevel—the operating system level

Data is managed as a named hierarchy of files and folders

Uses Network Attached Storage (NAS) file server or “filer” for file storage, using protocols such as Common Internet File System (CIFS) or Network File System (NFS

Both storage is very closelyassociated with the server and the operating system that is using the storage

Amazon S3 storage is independent of a server and is accessed over the Internet

Data is managed as objects using an Application Program Interface (API) built on standard HTTP verbs

Objects reside in containers called *buckets*,

Each object is identified by a unique user-specified key (filename).

Buckets are a simple flat folder with no file system hierarchy. That is, you can have multiple buckets, but you can’t have a sub-bucket within a bucket. Each bucket can hold an unlimited number of objects.

You can’t “mount” a bucket, “open” an object, install an operating system on Amazon S3, or run a database on it.

**Highly-durable and highly-scalable**

Amazon S3 objects are automatically replicated on multiple devices in multiple facilities within a region.

The same with scalability—if your request rate grows steadily, Amazon S3 automatically partitions buckets to support very high request rates and simultaneous access by many clients.

**Buckets**

A *bucket* is a container (web folder) for objects (files) stored in Amazon S3

Name is globally unique

No file and folder hierarchy

Can contain upto 63 characters, best practive to have domain name in it

Objects can range in size from 0 bytes up to 5TB

Each object contains both data and metadata.

Metadata – System meta data - The metadata associated with an Amazon S3 object is a set of name/value pairs that describe the object (date last modified, object size, MD5 digest, and HTTP Content-Type)

                        User meta data – optional (can be specified at object creation)

KEYS (FileName)  
Every object stored in an 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. The combination of bucket, key, and optional version ID uniquely identifies an Amazon S3 object

**Durability and Availability**

Will my data still be there in the future?”

Can I access my data right now?”

Amazon S3 standard storage is designed for 99.999999999% durability and 99.99% availability of objects over a given year. For example, if you store 10,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every

10,000,000 years. Amazon S3 achieves high durability by automatically storing data redundantly on multiple devices in multiple facilities within a region. It is designed to sustain the concurrent loss of data in two facilities without loss of user data

**Data Consistency**

Amazon S3 is an *eventually consistent* system. Because your data is automatically replicated across multiple servers and locations within a region, changes in your data may take some time to propagate to all locations. As a result, there are some situations where information that you read immediately after an update may return stale data

**Access Control**

Secure

Using ACL (Access control list) and

bucket policies, IAM (Identity and access management)

Similar to IAM except

They are associated with the bucket resource instead of an IAM principal.

They include an explicit reference to the IAM principal in the policy. This principal can be associated with a different AWS account, so Amazon S3 bucket policies allow you to assign cross-account access to Amazon S3 resources.

Using an Amazon S3 bucket policy, you can specify who can access the bucket, from where (by Classless Inter-Domain Routing [CIDR] block or IP address), and during what time of day

**Static Website Hosting**

common use case for Amazon S3 storage is *static website* hosting.

Many websites, particularly micro-sites, don’t need the services of a full web server

That don’t need server side processing

Can have client side processing

**Prefixes and Delimiters**

Amazon S3 uses a flat structure in a bucket, it supports the use of *prefix* and *delimiter* parameters when listing key names. This feature lets you organize, browse, and retrieve the objects within a bucket hierarchically

The REST API, wrapper SDKs, AWS CLI, and the Amazon Management Console all support the use of delimiters and prefixes. This feature lets you logically organize new data and easily maintain the hierarchical folder-and-file structure of existing data uploaded or backed up from traditional file systems

**Storage Classes**

S3 Standard

            offers high durability, high availability, low latency, and high performance object storage for general purpose use. Because it delivers low first-byte latency and high throughput, Standard is well-suited for short-term or long-term storage of frequently accessed data

S3 Standard – Infrequent access (Standard-IA)

            offers the same durability, low but is designed for long-lived, less frequently accessed data

S3 Reduced Redundancy storage (RRS)

            offers slightly lower durability (4 nines) than Standard or Standard-IA at a reduced cost. It is most appropriate for derived data that can be easily reproduced, such as image thumbnails

Glacier –

secure, durable, and extremely low-cost cloud storage for data that does not require real-time access, such as archives and longterm backups.

            Can be restored using S3 API. .Takes 3 to 5 hours.. Glacier object is copied to Amazon S3 RRS. Note that the restore simply creates a copy in Amazon S3 RRS; the original data object remains in Amazon Glacier until explicitly deleted.

**Object Lifecycle Management**

data has a natural lifecycle, starting out as “hot” (frequently accessed) data, moving to “warm” (less frequently accessed) data as it ages, and ending its life as “cold” (long-term backup or archive) data before eventual deletion.

Using Amazon S3 lifecycle configuration rules, you can significantly reduce your storage costs by automatically transitioning data from one storage class to another or even automatically deleting data after a period of time. For example, the lifecycle rules for

backup data might be

Store backup data initially in Amazon S3 Standard.

After 30 days, transition to Amazon Standard-IA.

After 90 days, transition to Amazon Glacier.

After 3 years, delete

**Logging**

Logging can be enabled for audit, disabled by default

Event Notifications

Amazon S3 event notifications are set up at the bucket level

Amazon S3 can 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.

**Pre-Signed URLs**

object owner can optionally share objects with others by creating a *presigned URL*, using their own security credentials to grant time-limited permission todownload the objects

When you create a pre-signed URL for your object, you must provide your security credentials and specify a bucket name, an object key, the HTTP method (GET to download the object), and an expiration date and time. The pre-signed

URLs are valid only for the specified duration. This is particularly useful to protect against “content scraping” of web content such as media files stored in Amazon S3

* Sharing data limited time

**Multipart Upload**

Provides multiupoad API to upload large objects

Objects are uploaded as set of parts, better network utilization (parallel transfers)

Can pause and restart

APIS should be used if object size > 100 MB

**Range GETs**

Allows to download a portion of object

Add HTTP range header, specify range of bytes of object

Useful while downloading large objects, poor-ntwork, download known portion of large amazon glacier..

**Cross-Region Replication**

*Cross-region replication* is a feature of Amazon S3 that allows you to asynchronously replicate all new objects in the source bucket in one AWS region to a target bucket in another region.

Cross-region replication is commonly used to reduce the latency required to access objects in Amazon S3 by placing objects closer to a set of users or to meet requirements to store backup data at a certain distance from the original source data

If turned on in an existing bucket, cross-region replication will only replicate new objects. Existing objects will not be replicated and must be copied to the new bucket via a separate command.

Note : If you are using Amazon S3 in a GET-intensive mode, such as a static website hosting, for best performance you should consider using an Amazon CloudFront distribution as a caching layer in front of your Amazon S3 bucket

**Encryption**

Data in flight – encrypted using SSL (S3 secure socket layer)

Data at rest –

**Server side encryption (SSE)**

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

Uses 256 bit AES (Advanced Encryption system)

**SSE-S3 (AWS-Managed Keys)**

AWS handles the key management and key protection for Amazon S3

Every object is encrypted with a unique key

The actual object key itself is then further encrypted by a separate master key.

A new master key is issued at least monthly, with AWS rotating the keys. Encrypted data, encryption keys, and master keys are all stored separately on secure hosts, further enhancing protection.

**SSE-KMS (AWS KMS Keys)**

Amazon handles your key management and protection for Amazon S3, but where you manage the keys

Using SSE-KMS, there are separate permissions for using the master key, which provide protection against unauthorized access to your objects stored in Amazon S3 and an additional layer of control.

AWS KMS also provides auditing, so you can see who used your key to access which object and when they tried to access this object. AWS KMS also allows you to view any failed attempts to access data from users who did not have permission to decrypt the data

***Client-Side Encryption***

Encrypt data on the client before sending it to Amazon S3.

Use an AWS KMS-managed customer master key.

Use a client-side master key.

When using client-side encryption, you retain end-to-end control of the encryption process, including management of the encryption keys.

For maximum simplicity and ease of use, use server-side encryption with AWS-managed keys (SSE-S3 or SSE-KMS).

**Versioning**

Versioning is turned on at the bucket level

Amazon S3 versioning helps protects your data against accidental or malicious deletion by keeping multiple versions of each object in the bucket, identified by a unique version ID.

Versioning allows you to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. If a user makes an accidental change or even maliciously deletes an object in your S3 bucket, you can restore the object to its original state simply by referencing the version ID in addition to the bucket and object key. Once enabled, versioning cannot be removed from a bucket; it can only be suspended.

**MFA Delete**

*MFA Delete* adds another layer of data protection on top of bucket versioning. MFA Delete requires additional authentication in order to permanently delete an object version or change the versioning state of a bucket. In addition to your normal security credentials, MFA Delete requires an authentication code (a temporary, one-time password

**Amazon Glacier**

Amazon Glacier is an extremely low-cost storage service that provides durable, secure,and flexible storage for data archiving and online backup.

To keep costs low, Amazon Glacier is designed for infrequently accessed data where a retrieval time of three to five hours is acceptable.

Data stored in TAR(tape archive) and zip

data is stored in *archives NOT object* . An archive can contain up to 40TB of data

***Vaults*** are containers for archives

**Vault locks** – policy to control write access to a vault.

You can retrieve up to 5% of your data stored in Amazon Glacier for free each month, calculated on a daily prorated basis

Glacier v/s S3

Glacier supports 40TB archives versus 5TB objects in Amazon S3.

Archives in Amazon Glacier are **identified by system-generated archive IDs,** while Amazon S3 lets you use “**friendly” key names.**

**Amazon Glacier archives are automatically encrypted**, while encryption at rest is optional in Amazon S3

Each archive is assigned a unique archive ID at the time of creation, and **the content of the archive is immutable**, meaning that after an archive is created it cannot be updated.