S3 Notes

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers scalability, data availability, security and performance.

S3 is Object-based - you can upload files but cannot install OS/DB.

S3 is a global service, not specific to region. But, you can have buckets in individual regions.

Files can be from 0 Bytes to 5TB

Files are stored in Buckets

S3 is a universal namespace, meaning the names must be unique globally.

When you upload a file to S3, you will receive a HTTP 200 code if the upload was successful.

S3 Provides MFA Delete to protect objects

AWS S3 Data Consistency Model:

Read after Write Consistency:

- S3 provides Read-after-Write consistency for PUTS of new objects
- For a PUT request, S3 synchronously stores data across multiple facilities before returning SUCCESS
- A process writes a new object to S3 and will be immediately able to read the Object i.e. PUT 200 -> GET 200
- A process writes a new object to S3 and immediately lists keys within its bucket. Until the change is fully propagated, the object might not appear in the list.
- However, if a HEAD or GET request to a key name is made before the object is created, then create the object shortly after that, a subsequent GET might not return the object due to eventual consistency. i.e. GET 404 -> PUT 200 -> GET 404- Eventual Consistency for overwrite PUTS and DELETES(if you update an existing file or delete a file and read it immediately, you may get the older version, or you may not. It take some time to reflect the changes)

Eventual Consistency for PUTS and DELETES:

- S3 provides Eventual Consistency for overwrite PUTS and DELETES in all regions.
- For updates and deletes to Objects, the changes are eventually reflected and not available immediately i.e. PUT 200 -> PUT 200 -> GET 200 (might be older version) OR DELETE 200 -> GET 200
- if a process replaces an existing object and immediately attempts to read it, S3 might return the prior data till the change is fully propagated
- if a process deletes an existing object and immediately attempts to read it, S3 might return the deleted data until the deletion is fully propagated

Storage Classes:

- S3 Standard:

- Designed for 99.999999999% i.e. 11 9’s Durability of objects across AZs
- Designed for 99.99% availability over a given year
- STANDARD is the default storage class, if none specified during upload
- Ideal for performance-sensitive use cases and frequently accessed data

- S3 Standard is appropriate for a wide variety of use cases, including cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics.

- **S3 – IA (Infrequently Accessed):**

- S3 Standard-Infrequent Access storage class is optimized for long-lived and less frequently accessed data. for e.g. for backups and older data where access is limited, but the use case still demands high performance

- Data stored redundantly across multiple geographically separated AZs and are resilient to the loss of an Availability Zone.
- S3 charges a retrieval fee for these objects, so they are most suitable for infrequently accessed data.

- **S3 One Zone – IA:**

- S3 One Zone-Infrequent Access storage classes are designed for long-lived and infrequently accessed data
- Ideal when the data can be recreated if the AZ fails, and for object replicas when setting cross-region replication (CRR).
- Stores the object data in only one AZ, which makes it less expensive than Standard-Infrequent Access
- Data is not resilient to the physical loss of the AZ resulting from disasters, such as earthquakes and floods.
- One Zone-Infrequent Access storage class is as durable as Standard-Infrequent Access, but it is less available and less resilient.

- **S3 - Intelligent Tiering:**

- S3 Intelligent Tiering storage class is designed to optimize storage costs by automatically moving data to the most cost-effective storage access tier, without performance impact or operational overhead.
- Ideal to optimize storage costs automatically for long-lived data when access patterns are unknown or unpredictable.
- There are no separate retrieval fees when using the Intelligent Tiering storage class. If an object in the infrequent access tier is accessed, it is automatically moved back to the frequent access tier..

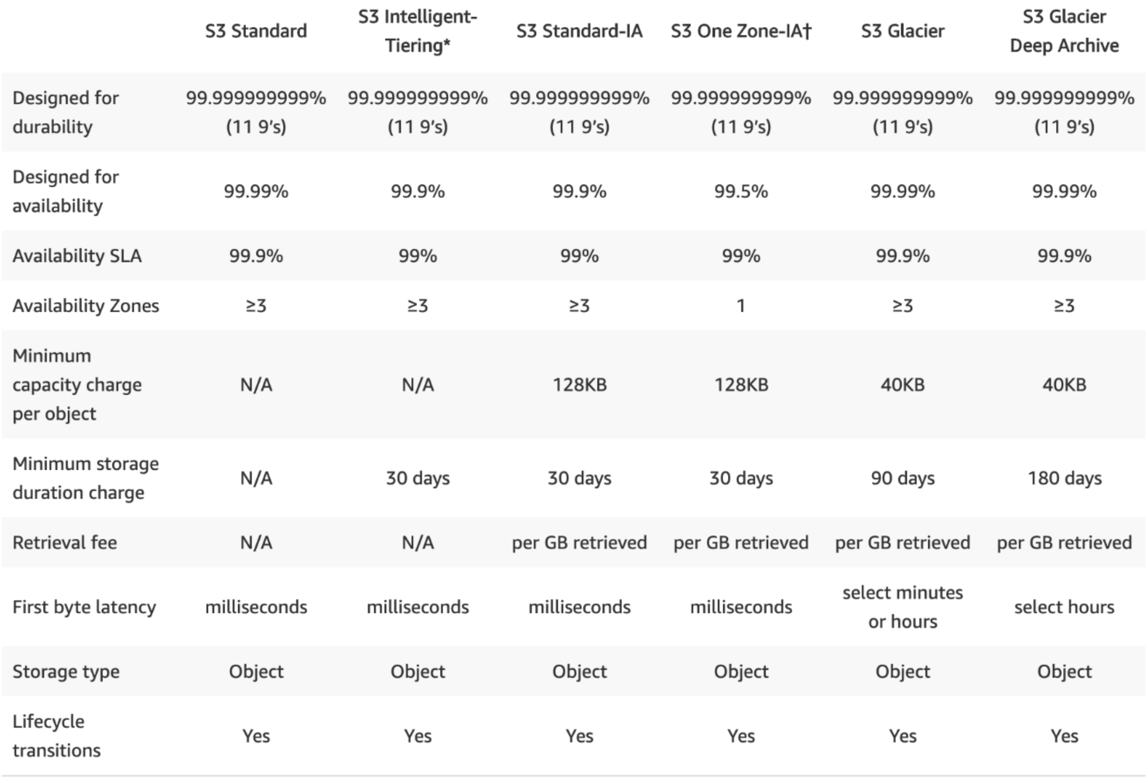
- **S3 Glacier:**

- GLACIER storage class is suitable for low cost data archiving where data access is infrequent and retrieval time of minutes to hours is acceptable.
- Storage class has a minimum storage duration period of 90 days

S3 Glacier Deep Archive:

- Glacier Deep Archive storage class provides lowest cost data archiving where data access is infrequent and retrieval time of hours is acceptable.
- Has a minimum storage duration period of 180 days and can be accessed in at a default retrieval time of 12 hours.

- Supports long-term retention and digital preservation for data that may be accessed once or twice in a year



	S3 Standard	S3 Intelligent-Tiering*	S3 Standard-IA	S3 One Zone-IA†	S3 Glacier	S3 Glacier Deep Archive
Designed for durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Designed for availability	99.99%	99.9%	99.9%	99.5%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99.9%	99.9%
Availability Zones	≥3	≥3	≥3	1	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128KB	128KB	40KB	40KB
Minimum storage duration charge	N/A	30 days	30 days	30 days	90 days	180 days
Retrieval fee	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds	select minutes or hours	select hours
Storage type	Object	Object	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes	Yes	Yes

S3 Object Versioning:

- S3 Object Versioning can be used to protect from unintended overwrites and deletions
- Versioning helps to keep multiple variants of an object in the same bucket and can be used to preserve, retrieve, and restore every version of every object stored in the S3 bucket.
- As Versioning maintains multiple copies of the same objects as whole and charges accrue for multiple versions for e.g. for a 1GB file with 5 copies with minor differences would consume 5GB of S3 storage space and you would be charged for the same.
- Versioning is not enabled by default and has to be explicitly enabled for each bucket
- Versioning once enabled, cannot be disabled and can only be suspended
- Versioning enabled on a bucket applies to all the objects within the bucket

AWS S3 Data Protection:

- Amazon S3 provides a S3 data protection using highly durable storage infrastructure designed for mission-critical and primary data storage.
- Objects are redundantly stored on multiple devices across multiple facilities in an S3 region.
- S3 PUT and PUT Object copy operations synchronously store the data across multiple facilities before returning SUCCESS.
- Once the objects are stored, S3 maintains its durability by quickly detecting and repairing any lost redundancy.

S3 Notes

Encryption in Transit is achieved by:

SSL/TLS

Encryption At Rest (Server Side) is achieved by:

S3 Managed Keys - SSE-S3

AWS Key Management Service, Managed Keys - SSE-KMS

Server Side Encryption with Customer provided Keys - SSE-C

Client Side Encryption

Data in-transit

S3 allows protection of data in-transit by enabling communication via SSL or using client-side encryption

Data at Rest

- S3 supports both client side encryption and server side encryption for protecting data at rest
- Using Server-Side Encryption, S3 encrypts the object before saving it on disks in its data centers and decrypt it when the objects are downloaded
- Using Client-Side Encryption, data is encrypted at client-side and uploaded to S3. In this case, the encryption process, the encryption keys, and related tools are managed by the user.

Server-Side Encryption

- Server-side encryption is about data encryption at rest
- Server-side encryption encrypts only the object data. Any object metadata is not encrypted.
- S3 handles the encryption (as it writes to disks) and decryption (when objects are accessed) of the data objects

Server-Side Encryption with S3-Managed Keys (SSE-S3):

- Each object is encrypted with a unique data key employing strong multi-factor encryption.
- SSE-S3 encrypts the data key with a master key that is regularly rotated.
- S3 server-side encryption uses one of the strongest block ciphers available, 256-bit Advanced Encryption Standard (AES-256), to encrypt the data.

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

- SSE-KMS is similar to SSE-S3, but it uses AWS Key management Services (KMS) which provides additional benefits along with additional charges
- KMS is a service that combines secure, highly available hardware and software to provide a key management system scaled for the cloud.
- KMS uses customer master keys (CMKs) to encrypt the S3 objects.

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

- Encryption keys can be managed and provided by the Customer and S3 manages the encryption, as it writes to disks, and decryption, when you access the objects
- When you upload an object, the encryption key is provided as a part of the request and S3 uses that encryption key to apply AES-256 encryption to the data and removes the encryption key from memory.

- When you download an object, the same encryption key should be provided as a part of the request. S3 first verifies the encryption key and if matches decrypts the object before returning back to you
- SSE-C request must be done through HTTPS and S3 will reject any requests made over HTTP when using SSE-C.

Client-Side Encryption

Client-side encryption refers to encrypting data before sending it to Amazon S3 and decrypting the data after downloading it