Simple Storage Service (S3)

- Introduction
- Object Storage versus Traditional Block and File Storage
- Amazon Simple Storage Service (Amazon S3) Basics
- Buckets and Objects
- Amazon S3 Operations
- Storage Classes
- Durability and Availability
- Data Consistency
- Static Website Hosting
- CORS - Explained
- Amazon S3 Advanced Features

Simple Storage Service (S3)

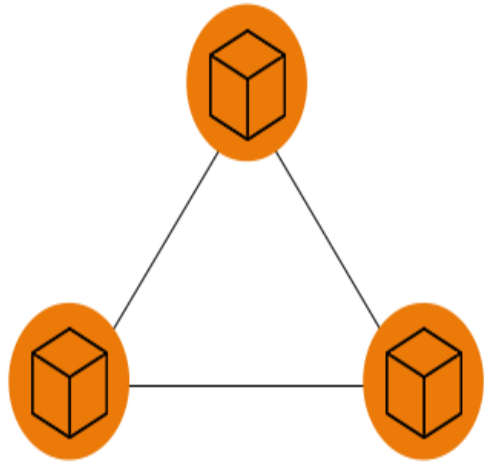
- Object Lifecycle Management
- Encryption
- Versioning
- MFA Delete
- Pre-Signed URLs
- Multipart Upload
- Range GETs
- Cross-Region Replication
- Logging
- Event Notifications
- Best Practices, Patterns, and Performance

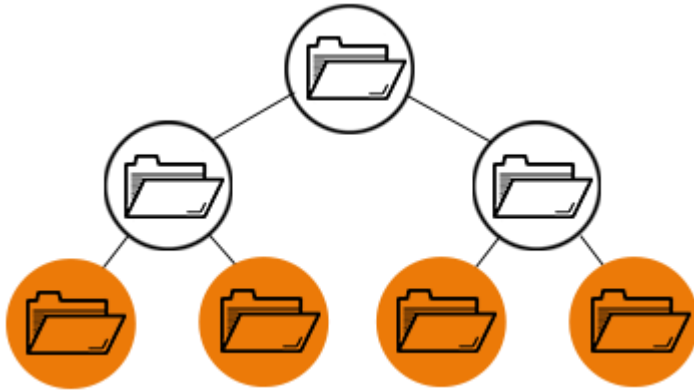
S3-Introduction

- 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 the web.
- Amazon S3 is one of first services introduced by AWS
- 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

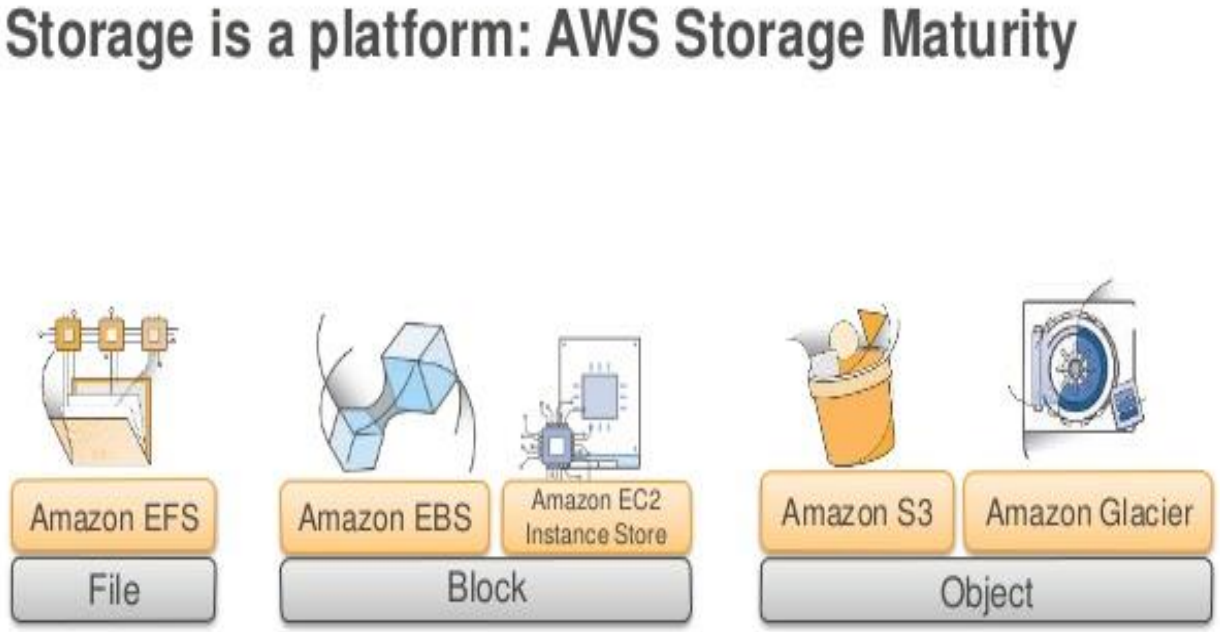
Object Storage vs Traditional Block and File Storage

- **Block storage** operates at a lower level—the raw storage device level—and manages data as a set of numbered, fixed-size blocks.
- Each block of data is given a unique identifier, which allows a storage system to place the smaller pieces of data wherever is most convenient
- **File storage** operates at a higher level—the operating system level—and manages data as a named hierarchy of files and folders
- Used protocols such as Common Internet File System (CIFS) or Network File System (NFS)





- Block or File storage is very closely associated with the server and the operating system that is using the storage.
- Object storage is independent of a server and is accessed over the Internet. Instead of managing data as blocks or files using SCSI, CIFS, or NFS protocols, data is managed as objects using an Application Program Interface (API)
- Each Amazon S3 object contains both data and metadata.



Amazon Simple Storage Service (Amazon S3) Basics

- Amazon S3 is a simple key, value object store designed for the Internet
- S3 provides unlimited storage space and works on the pay as you use model. Service rates gets cheaper as the usage volume increases
- S3 offers an extremely durable, highly available, and infinitely scalable data storage infrastructure at very low costs.
- S3 is an Object level storage (not a Block level storage) and cannot be used to host OS or dynamic websites

Buckets and Objects

- Amazon S3 allows to store objects (files) in “**buckets**” (directories)
- Buckets must have a **globally unique name**
- Buckets are defined at the **region level**
- **100 buckets** (soft limit) and maximum of 1000 buckets can be created in each of AWS account
- Naming convention
  - No uppercase
  - No underscore
  - 3-63 characters long
  - Not an IP
  - Must start with lowercase letter or number

Objects

- Objects (files) have a Key
- The **key** is the FULL path:
  - s3://**my-bucket**/my_file.txt
  - s3://**my-bucket**/my_folder1/another_folder/my_file.txt
- The key is composed of **prefix + object name**
  - s3://my-bucket/my_folder1/another_folder/my_file.txt
- There's no concept of "directories" within buckets

(although the UI will trick you to think otherwise)

- Just keys with very long names that contain slashes ("/")
- **Metadata** is the data about the data and is a set of name-value pairs that describe the object for e.g. content-type, size, last modified. Custom metadata can also be specified at the time the object is stored.

Amazon S3 Operations

- The Amazon S3 API is intentionally simple, with only a handful of common operations. They include:
  - Create/delete a bucket
  - Write an object
  - Read an object
  - Delete an object
  - List keys in a bucket

- The native interface for Amazon S3 is a REST API. With the REST interface, you use standard HTTP or **HTTPS** requests to create and delete buckets, list keys, and read and write objects.
- SDK, AWS Command Line Interface (CLI), and the AWS Management Console.

Durability and Availability

- **Durability** addresses the question, “Will my data still be there in the future?”
- **Availability** addresses the question, “Can I access my data right now?”
- Amazon S3 is designed to provide both very high durability and very high availability for your data.
- 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.

Storage Classes

- Amazon S3 offers a range of storage classes suitable for various use cases.
- https://aws.amazon.com/s3/storage-classes/

Data Consistency

- Amazon S3 is an eventually consistent system.
- Your data is automatically replicated across multiple servers and locations within a region.
- Read after write consistency for PUTS of new objects
  - As soon as a new object is written, we can retrieve it
  - This is true, except if we did a GET before to see if the object existed
- Eventual Consistency for DELETES and PUTS of existing objects
  - If we read an object after updating, we might get the older version
  - If we delete an object, we might still be able to retrieve it for a short time

Static Website Hosting

- To configure an Amazon S3 bucket for static website hosting:

1. Create a bucket with the same name as the desired website hostname.
2. Upload the static files to the bucket.
3. Make all the files public.
4. Enable static website hosting for the bucket. This includes specifying an Index document and an Error document.
5. The website will now be available at the S3 website URL: <bucket-name>.s3-website-<AWS-region>.amazonaws.com.
6. Create a friendly DNS name in your own domain for the website using a DNS CNAME, or an
7. Amazon Route 53 alias that resolves to the Amazon S3 website URL.
8. The website will now be available at your website domain name.

DEMO

CORS - Explained

- An **origin** is a scheme (protocol), host (domain) and port
- E.g.: https://www.mainorigin.com (implied port is 443 for HTTPS, 80 for HTTP)
- CORS means **Cross-Origin Resource Sharing**
- Web Browser based mechanism to allow requests to other origins while visiting the main origin
- Same origin: http://www.mainorigin.com/app1 & http://www.mainorigin.com/app2
- Different origins: http://www.mainorigin.com & http://www.otherorigin.com
- The requests won't be fulfilled unless the other origin allows for requests, using CORS Headers (ex: **Access-Control-Allow-Origin**)

CORS – S3

- If a client does a cross-origin request on our S3 bucket, we need to enable the correct CORS headers
- It’s a popular exam question
- You can allow for a specific origin or for * (all origins)
- With the Cross-Origin Resource Sharing (CORS) policy enabled

Object Lifecycle Management

- S3 Object lifecycle can be managed by using a lifecycle configuration, which defines how S3 manages objects during their lifetime.
- Lifecycle configuration enables simplification of object lifecycle management, for e.g. moving of less frequently access objects, backup or archival of data for several years or permanent deletion of objects,
- S3 controls all transitions automatically
- Lifecycle Management rules applied to an bucket are applicable to all the existing objects in the bucket as well as the ones that will be added anew
- S3 Object lifecycle management allows 2 types of behavior
  - **Transition** in which the storage class for the objects change
  - **Expiration** where the objects expire and are permanently deleted

  DEMO

Encryption – S3

- There are 4 methods of encrypting objects in S3
  - SSE-S3: encrypts S3 objects using keys handled & managed by AWS
  - SSE-KMS: leverage AWS Key Management Service to manage encryption keys
  - SSE-C: when you want to manage your own encryption keys
  - Client Side Encryption

- It’s important to understand which ones are adapted to which situation for the exam

SSE-S3

- Encryption using keys handled & managed by Amazon S3
- Every object is encrypted with a unique key
- Object is encrypted server side
- AWS rotating the keys
- AES-256 encryption type
- Must set header: "x-amz-server-side-encryption": "AES256"

SSE-KMS

- Encryption using keys handled & managed by KMS
- KMS Advantages: user control + audit trail
- Object is encrypted server side
- Must set header: “x-amz-server-side-encryption”: “aws:kms”
- AWS KMS also allows you to view any failed attempts to access data from users who did not have permission to decrypt the data

SSE-C

- Server-side encryption using data keys fully managed by the customer outside of AWS
- Amazon S3 does not store the encryption key you provide
- HTTPS must be used
- Encryption key must provided in HTTP headers, for every HTTP request made
- AWS will do the encryption/decryption of your objects while you maintain full control of the keys used to encrypt/decrypt the objects in Amazon S3.

Client Side Encryption

- Client library such as the Amazon S3 Encryption Client
- Clients must encrypt data themselves before sending to S3
- Clients must decrypt data themselves when retrieving from S3
- Customer fully manages the keys and encryption cycle
- You have the following two options for using data encryption keys:
  - Use an AWS KMS-managed customer master key.
  - Use a client-side master key.

Amazon S3 - Versioning

- You can version your files in Amazon S3
- It is enabled at the bucket level
- Same key overwrite will increment the “version”: 1, 2, 3….
- It is best practice to version your buckets
  - Protect against unintended deletes (ability to restore a version)
  - Easy roll back to previous version
- Notes:
  - Any file that is not versioned prior to enabling versioning will have version “null”
  - Suspending versioning does not delete the previous versions
- DEMO

Amazon S3 - 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.
- MFA Delete requires an authentication code generated by a hardware or virtual Multi-Factor Authentication (MFA) device
- MFA Delete can only be enabled by the root account.

Pre-Signed URLs

- By default, all objects are private — meaning only the bucket account owner initially has access to the object
- If you want a user to have access to a specific bucket or objects without making them public, you can provide the user with the appropriate permissions using an IAM policy
- When you create a presigned URL for your object, you must provide your security credentials and specify a bucket name, an object key, the HTTP method, and an expiration date and time.
- The presigned URLs are valid only for the specified duration.
- By default, all objects are private — meaning only the bucket account owner initially has access to the object

Multipart Upload

- Uploading or coping large objects as a set of parts, which generally gives better network utilization (through parallel transfers)
- Is used to upload an object (objects) in parts
  - Parts are uploaded independently and in parallel, in any order
- It is recommended for objects sizes of 100MB or larger
  - However, you can use for object size 5 MB ~ 5 TB
  - You must use it for Objects larger than 5GB
- This is done through the S3 Multipart upload API
- Multipart upload is a three-step process:
  - Initiation
  - Uploading the parts
  - Completion
- After all of the parts are uploaded, Amazon S3 assembles the parts in order to create an object.

Range GETs

- It is possible to download (GET) only a portion of an object by using Range GET.
- Using the Range HTTP header in the GET request specify a range of bytes of the object.

Cross-Region Replication

- Cross-region replication allows you to asynchronously replicate all new objects across region
- Any metadata and ACLs associated with the object are also part of the replication.
- To enable cross-region replication, versioning must be turned on for both source and destination buckets
- IAM policy should updated to give Amazon S3 permission to replicate objects on your behalf.

Amazon S3 - Logging

- Amazon S3 server access logs can be enable to track requests to your Amazon S3 bucket
- Logging is **off** by default
- When you enable logging for a bucket, you must choose where the logs will be stored (**the target bucket**).
- Logs include information such as:
  - Requestor account and IP address
  - Bucket name
  - Request time
  - Action (GET, PUT, LIST, and so forth)
  - Response status or error code

S3 Transfer Acceleration

- 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.
- Is used to accelerate object uploads/downloads to S3 buckets from users over long distances
- Transfer acceleration is as secure as direct upload to S3 over the internet
- It utilizes AWS CloudFront edge location nearest to the upload source (user), once data arrives at the edge location, it gets routed to the destination S3 bucket over an optimized network path.
- Using Transfer acceleration incurs a charge
  - AWS checks for speed enhancements, and if no enhancement is provided, client does not get charged for using Transfer Acceleration
- You can use multipart uploads with Transfer Acceleration
- Transfer Acceleration is HIPAA Compliant
- http://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparsion.html

Best Practices

- Block ‘Public Access’ to AWS account level for data protection
- Block ‘Public Access’ to the S3 bucket
- Amazon S3 buckets should have ‘Default Encryption’ feature enabled
- AWS S3 buckets should have ‘server access logging’ enabled to track access requests
- Buckets should have the ‘MFA Delete’ feature enabled
- Ensure AWS S3 ‘object versioning’ is enabled for an additional level of data protection
- AWS S3 ‘object versioning’ need to be enabled for an additional level of data protection
- S3 buckets to use ‘Transfer Acceleration’ feature for faster data transfers