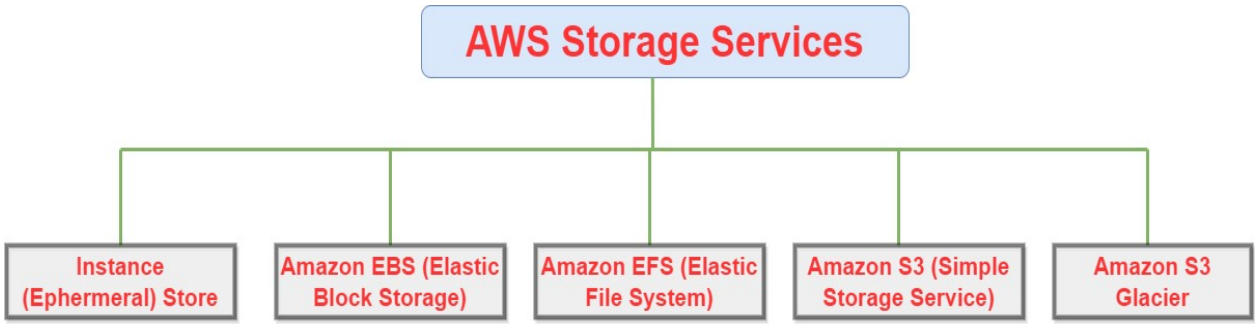
AWS Storage Services

Storage is another AWS core service category. Some of the AWS storage services include: instance store (ephemeral storage), Amazon EBS, Amazon EFS, Amazon S3, and Amazon S3 Glacier.



- Instance store (ephemeral storage) - It is temporary storage that is added to your Amazon EC2 instance.
- Amazon EBS (Elastic Block Storage) - It is persistent, mountable storage that can be mounted as a device to an Amazon EC2 instance. Amazon EBS volumes and Amazon EC2 instances must be in the same Availability Zone. Amazon EBS volume can be attached to single EC2 instance only.
- Amazon EFS (Elastic File System) – It is a shared file system that multiple Amazon EC2 instances can mount at the same time.
- Amazon S3 (Simple Storage Service) - It is persistent storage where each file becomes an object, stored in containers called Buckets. Each object in bucket is available through a Uniform Resource Locator (URL); it can be accessed from anywhere.
- Amazon S3 Glacier is for cold storage for data that is not accessed frequently.

AWS S3 – Introduction

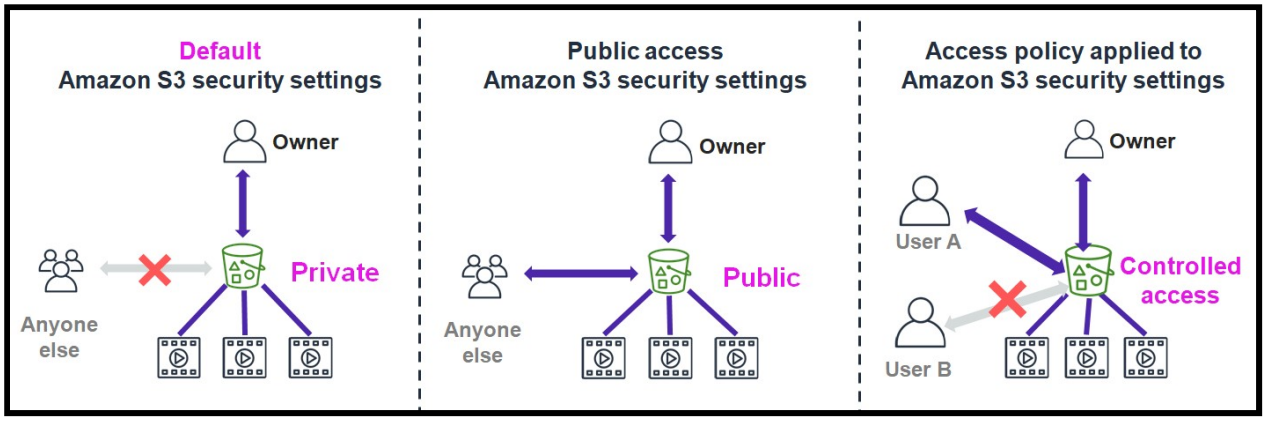
- Amazon S3 offers the ability to store large amounts of data.
- Amazon S3 is object storage that is built to store and retrieve any amount of data from anywhere.
- Developers can use Amazon S3 as an object storage solution for backup files, Static Website Hosting, Repository for Internet of Things (IoT) device data and big data etc.
- Amazon S3 is object-level storage, which means that if you want to change a part of a file, you must make the change and then re-upload the entire modified file.
- It enables you to store virtually unlimited amounts of data.
- Amazon S3 also provides low-latency access to the data over the internet by Hypertext Transfer Protocol (HTTP) or Secure HTTP (HTTPS), so you can retrieve data anytime from anywhere.
- You can access Amazon S3 through
    - ✓ the web-based AWS Management Console;
    - ✓ AWS CLI (Command Language Interface)
    - ✓ programmatically through the API and SDKs
- The essential components of Amazon S3 are buckets and objects.

S3 Buckets and Objects

- A bucket is a container for objects that are stored in Amazon S3.
- You can store any number of objects in a bucket and can have up to 100 buckets in your account.
- Data/files are stored as objects. You place objects in a bucket, which you create.
- Objects can be almost any data file, such as images, videos, or server logs.
- Individual objects cannot be larger than 5 TB.
- When you create an S3 bucket, you specify a name for it. You must follow specific rules when you choose a bucket name:
    - ✓ An S3 bucket name must be globally unique
    - ✓ Additionally, bucket names –
        - Must be at least 3 characters and no more than 63 characters.
        - Can contain lowercase letters, numbers, and hyphens (-).

- Must not contain uppercase characters or underscores (_).

- After you create an S3 bucket, you can't change the bucket name, so be careful when choosing the name.
- When you create S3 creates buckets, you must specify the Region. When you choose a Region, you should consider latency, cost, or regulatory requirements that could affect the data in the bucket.
- After you create a bucket, you cannot change its Region.
- Granular access to bucket and objects – you control over who can access by using AWS Identity and Access Management (IAM) policies, Amazon S3 bucket policies, and even per-object access control lists.
- By default, all S3 buckets are private and can be accessed only by users who are explicitly granted access.
- Here are three different general approaches to configuring access to objects in an S3 bucket.



Hands On: S3 Buckets and Objects

Task 1: Create a Bucket

Task 2: Upload an object to your Bucket

Task 3: Delete the objects

Task 4: Emptying your Bucket

Task 4: Delete the Buckets

Task 1: Create a Bucket

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Choose Create bucket.
3. The Create bucket wizard opens.
4. In Bucket name, enter a DNS-compliant name for your bucket. After you create the bucket, you cannot change its name.
5. In Region, choose the AWS Region where you want the bucket to reside.
6. Under Object Ownership, to disable or enable ACLs and control ownership of objects uploaded in your bucket.
7. In Bucket settings for Block Public Access, choose the Block Public Access settings that you want to apply to the bucket.
8. (Optional) Under Bucket Versioning, you can choose if you wish to keep variants of objects in your bucket.
9. (Optional) Under Tags, you can choose to add tags to your bucket. Tags are key-value pairs used to categorize storage.
10. (Optional) Under Default encryption, you can choose to configure your bucket to use server-side encryption.
11. (Optional) You can enable/disable S3 Object Lock.
12. Choose Create bucket.Congratulations! You have created your first bucket in Amazon S3.

Task 2: Upload an object to your Bucket

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the Buckets list, **choose** the name of the bucket that you want to upload your object to.
3. On the Objects tab for your bucket, choose **Upload**.
4. Under Files and folders, choose **Addfiles**.
5. Choose a file to upload, and then choose **Open**.
6. Choose **Upload**.You've successfully uploaded an object to your bucket.

Task 3: Delete the objects

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the Buckets list, choose the name of the bucket that you want to delete an object from.
3. Select the check box to the left of the names of the objects that you want to delete.
4. Choose Actions and choose Delete from the list of options that appears.Alternatively, choose Delete from the options in the upper right.
5. Type **permanently delete** if asked to confirm that you want to delete these objects.
6. Choose Delete objects in the bottom right and Amazon S3 deletes the specified objects.

Task 4: Emptying your Bucket

If you plan to delete your bucket, you must first empty your bucket, which deletes all the objects in the bucket.

1. In the Buckets list, select the bucket that you want to empty, and then choose **Empty**.
2. To confirm that you want to empty the bucket and delete all the objects in it, in Empty bucket, type **permanentlydelete**.
3. To empty the bucket and delete all the objects in it, and choose **Empty**.
4. To return to your bucket list, choose **Exit**.

Task 5: Delete Bucket

After you empty your bucket or delete all the objects from your bucket, you can delete your bucket.

1. To delete a bucket, in the Buckets list, select the bucket.
2. Choose **Delete**.
3. To confirm deletion, type the name of the bucket.
4. To delete your bucket, choose **Delete bucket**.

Hands-on Assignments

1. Perform the following Tasks:
    a. Create a bucket
    b. Upload objects in it.
    c. Access any specific object using object URL

d. If access is denied, demonstrate why you are unable to access.
e. Make necessary bucket configuration changes to make the all objects public.

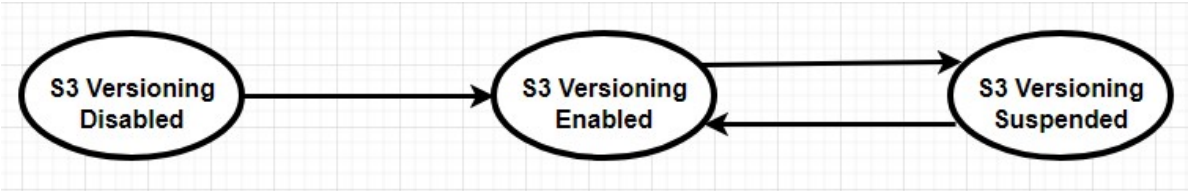
2. Perform the following Tasks:
    a. Create a bucket
    b. Upload two objects in it.
    c. One object should be accessible to everyone and other one should be private
    d. Try to access both the objects using object URL and observe the results.

3. Perform the following Tasks:
    a. Create a bucket
    b. Upload two objects in it.
    c. Delete the object
    d. Demonstrate the difference between **Empty Bucket**and **Delete Bucket** option.

AWS S3 Versioning

- ✓ Amazon S3 Versioning feature provides an additional level of protection.
- ✓ Versioning is a method of keeping multiple variants of an object in the same bucket.
- ✓ It provides a way to recover data if an application fails, or when customers accidentally overwrite or delete objects.
- ✓ You can use versioning to preserve, retrieve, and restore every version of every object stored in an S3 bucket.
- ✓ 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.
- ✓ Overwriting an object results in a new object version in the bucket. You can always restore the previous version.
- ✓ Buckets can be in one of three states:
    - Unversioned(the default, versioning-disabled),
    - versioning-enabled, or
    - versioning-suspended.

- ✓ After you enable versioning for a bucket, you can never change it to an un-versioned state. You can, however, suspend versioning on that bucket.



Hands-on: AWS S3 Versioning

Task 1: Enable or Suspend versioning on an S3 Bucket.

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the Buckets list, choose the name of the bucket that you want to enable versioning for.
3. Choose Properties.
4. Under Bucket Versioning, choose Edit.
5. Choose Suspend or Enable, and then choose Save changes.

Hands-on Assignments

1. Perform the following Tasks:
    a. Create a bucket and Upload two objects in it.
    b. Check whether bucket versioning is enabled or not.
    c. Enable bucket versioning
    d. Check whether bucket versioning is enabled or not.
    e. Upload same object multiple times and observer version ID's of each object.
    f. Try to delete the object. Does it delete the object??
    g. Permanently delete the object
2. Perform the following Tasks:
    a. Create a bucket
    b. Upload two objects in it.
    c. Check whether bucket versioning is enabled or not.
    d. If disabled, enable it.

e. Upload another different object
f. Observe the Version Id's of Newly created object
g. Observe the Version ID of initially uploaded two objects.

3. Perform the following Tasks:
    a. Create a versioning enabled bucket
    b. Upload two objects (a.txt and b.txt) in it.
    c. Upload a.txt object again
    d. Observe the Version Id's of all objects
    e. Disable bucket versioning and write down your observation
    f. Is it possible to suspend versioning? If Yes, Suspend the versioning.
    g. Upload the a.txt object again
    h. Write down your observations.

AWS S3 Encryption

- data encryption is an essential tool to protect digital data.
- Data encryption takes data that is in readable form and encodes.
- Encrypted data is unreadable to anyone who does not have access to the secret key that can be used to decode it.
- Thus, even if an attacker gains access to your data, they cannot make sense of it.
- You have two primary options for encrypt data stored in Amazon S3:
    - ✓ Server-side Encryption and
    - ✓ Client-side Encryption.
- When you set the Default encryption option on a bucket, it enables server-side encryption.

Server-side Encryption	**Client-side Encryption**
✓ Amazon S3 encrypts your object before it saves the object to disk. ✓ Amazon S3 will decrypt it when you download the object.	✓ You encrypt the data on the client side before you upload it to Amazon S3. ✓ You manage the encryption/decryption process, the encryption keys, and related tools.

Hands-on: AWS S3 Encryption

Task 1: Enable Default Encryption on an S3 Bucket.

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the Buckets list, **choose** the name of the bucket that you want.
3. Choose **Properties**.
4. Under Default encryption, choose **Edit**.
5. To enable or disable server-side encryption, choose **Enable or Disable**.
6. To enable server-side encryption using an Amazon S3-managed key, under Encryption key type, choose Amazon S3 key (SSE-S3).
7. To enable server-side encryption using an AWS KMS key, follow these steps:
    A. Under Encryption key type, choose AWS Key Management Service key (SSE-KMS).
    B. Under AWS KMS key choose one of the following:
        - AWS managed key
        - Choose from your KMS root keys, and choose your KMS root key.
        - Enter KMS root key ARN, and enter your AWS KMS key ARN.
8. To use S3 Bucket Keys, under **Bucket Key**, choose **Enable**.
9. Choose Save changes.

Hands-on Assignments

1. Perform the following Tasks:
    a. Create a bucket and Enable server side encryption
    b. Upload two objects in it.

2. Perform the following Tasks:
    a. Create a bucket
    b. Upload client side encrypted two objects in it.

AWS S3 Security and Bucket Policies

To protect your data in Amazon S3, by default, users only have access to the S3 resources they create. You can grant access to other users by using one or a combination of the following access management features:

- **AWS IAM:**AWS Identity and Access Management (IAM) to create users and manage their respective access;
- **ACL:**Access Control Lists (ACLs) to make individual objects accessible to authorized users;
- **Bucket policies**: Bucket policies to configure permissions for all objects within a single S3 bucket;
- **Object Lock:** Object Lock can help prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely.
- **Object Ownership:** S3 Object Ownership is an Amazon S3 bucket-level setting that you can use to disable access control lists (ACLs) and take ownership of every object in your bucket, simplifying access management for data stored in Amazon S3.
- **Encryption:** Amazon S3 supports both server-side encryption and client-side encryption for data uploads.

AWS S3 Bucket Policies

- You can create and configure bucket policies to grant access permissions to your bucket and the objects in it.
- Only the bucket owner can associate a policy with a bucket.
- The permissions attached to the bucket apply to all of the objects in the bucket.
- In your bucket policy, you can use wildcard characters to grant permissions to a subset of objects.
- For example, you can control access to groups of objects that begin with a common prefix or end with a given extension, such as .html.

Hands On: S3 Bucket Policies

Task : Create or edit a bucket policy

13. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
14. In the Buckets list, choose the name of the bucket that you want to create a bucket policy for or whose bucket policy you want to edit.
15. Choose Permissions.
16. Under Bucket policy, choose Edit. This opens the Edit bucket policy page.
17. On the Edit bucket policy page, choose Policy generator to generate a policy automatically, or edit the JSON in the Policy section.
18. If you choose Policy generator, the AWS Policy Generator opens in a new window:
    - ✓ On the AWS Policy Generator page, in Select Type of Policy, choose S3 Bucket Policy.
    - ✓ Add a statement by entering the information in the provided fields, and then choose Add Statement. Repeat for as many statements as you would like to add..
    - ✓ After you finish adding statements, choose Generate Policy.
    - ✓ Copy the generated policy text, choose Close, and return to the Edit bucket policy page in the Amazon S3 console.
19. In the Policy box, edit the existing policy or paste the bucket policy from the Policy generator. Make sure to resolve security warnings, errors, general warnings, and suggestions before you save your policy.
20. Choose Save changes.

AWS S3 Websites

- ✓ You can use Amazon S3 to host a static website.
- ✓ On a static website, individual webpages include static content. They might also contain client-side scripts.By contrast, a dynamic website relies on server-side processing, including server-side scripts, such as PHP, JSP, or ASP.NET.
- ✓ Amazon S3 does not support server-side scripting, but AWS has other resources (such as EC2) for hosting dynamic websites.

- ✓ When you configure a bucket as a static website, if you want your website to be public, you can grant public read access. To make your bucket publicly readable, you must disable block public access settings for the bucket and write a bucket policy that grants public read access.
- ✓ No servers are needed, and you can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web.
- ✓ Hosting a static website involves following tasks:
    - ✓ Create a bucket in Amazon S3
    - ✓ Upload content to your bucket
    - ✓ Enable access to the bucket objects

Hands-on: AWS S3 -Static Website Hosting

Step 1: Create a S3 Bucket.

- Choose Create bucket
- Enter Bucket name and choose region. An S3 bucket name is globally unique.
- Public access to buckets is blocked by default.
- In the Object Ownership section, select ACLs enabled, then verify Bucket owner preferred is selected.
- Clear Block all public access, then select the box that states I acknowledge that the current settings may result in this bucket and the objects within becoming public.
- Choose Create bucket.

Step 2: Upload objects to your S3 Bucket.

- Open the bucket created and upload objects.
- Choose Upload
- Choose Add files
- Locate and select all website file that you have developed.
- Choose Upload Choose Close.

Step 3: Enabling access to the objects

- Objects that are stored in Amazon S3 are private by default. You need to public all the objects uploaded to bucket. Select all objects.

- In the Actions menu, choose Make public via ACL.
- Choose Make public

Step 4: Enable Static Website Hosting

- Move to the Properties Tab. Scroll to the Static website hosting panel.
- Choose Edit
- Configure the following settings:
  - ✓ Static web hosting: Enable
  - ✓ Hosting type: Host a static website
  - ✓ Index document: index.html
  - ✓ Error document: error.html
- Choose Save changes
- In the Static website hosting panel, choose the link under Bucket website endpoint.
- Open browser and paste it in address bar. Tour website has been hosted and now it is accessible.

Hands on Assignment:

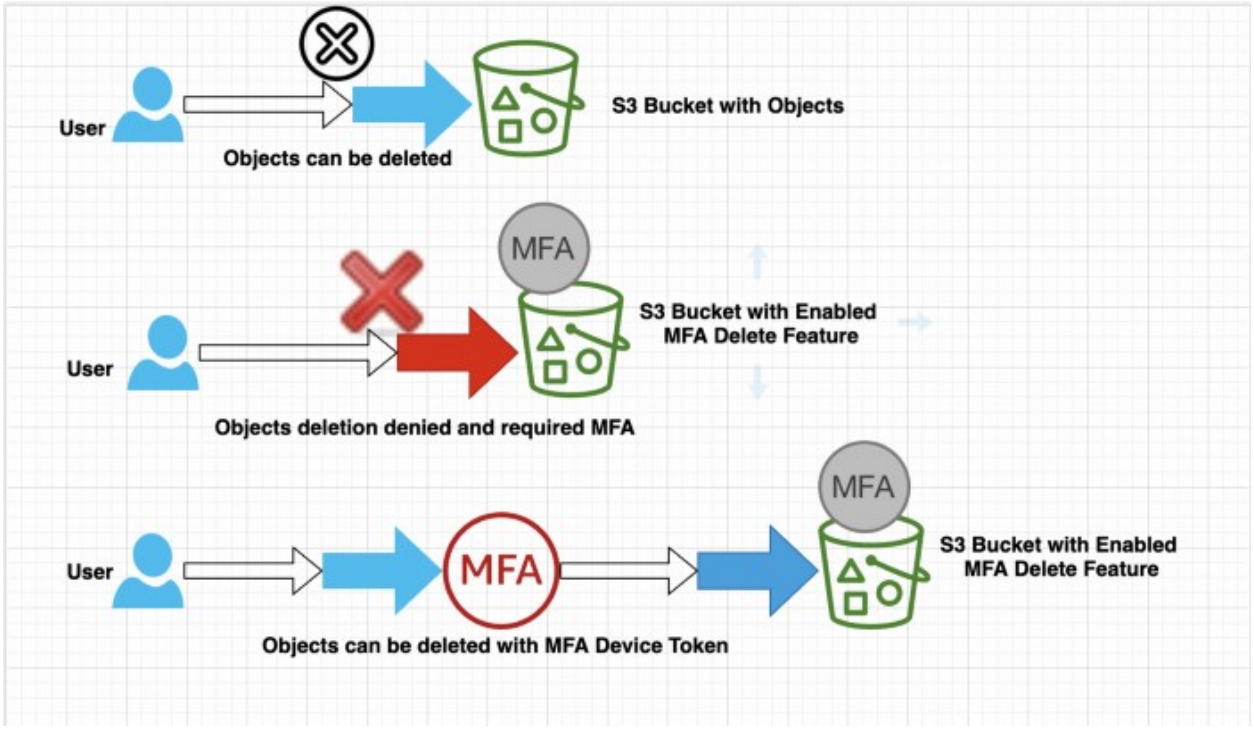
Develop Simple HTML web page containing **"WELCOME TO KARNATAKA LMS"** and host it using AWS S3 service.

AWS S3 MFA Delete

- When working with S3 Versioning in Amazon S3 buckets, you can optionally add another layer of security by configuring a bucket to enable *MFA (multi-factor authentication) delete*.
- When you do this, the bucket owner must include two forms of authentication in any request to delete a version or change the versioning state of the bucket.
- MFA delete requires additional authentication for either of the following operations:
  - ✓ Changing the versioning state of your bucket
  - ✓ Permanently deleting an object version

- MFA delete requires two forms of authentication together:
  - ✓ Your security credentials
  - ✓ Six-digit MFA code displayed on an approved authentication device
- MFA delete thus provides added security if, for example, your security credentials are compromised.
- Steps to Enable MFA Delete Feature
  - ✓ Create S3 bucket
  - ✓ Make sure you have Root User Account Keys for CLI access
  - ✓ Configure AWS CLI with root account credentials
  - ✓ List and Verify Versioning enabled for the Bucket
  - ✓ List the Virtual MFA Devices for Root Account
  - ✓ Enable MFA Delete on Bucket
  - ✓ Test MFA Delete

- Following figure illustrates MFA Delete feature:



Hands-on: AWS S3 MFA Delete

Task: Enable MFA Delete on Bucket

1. Go to the **AWS CLI Interface**
2. Authenticate yourself
3. Type the below command to enable MFA Delete

```
aws s3api put-bucket-versioning --bucket DOC-EXAMPLE-BUCKET1 --versioning-configuration Status=Enabled,MFADelete=Enabled --mfa "MFA Code"
```

4. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
5. In the Buckets list, choose the name of the bucket that contains the object that you want to delete.
6. Try to delete the bucket, choose delete
7. It should ask for MFA Code, if so, it indicates that your bucket and its object are MFA delete Protected.

Hands-on Assignments

1. Perform the following Tasks:
    a. Create a bucket and Enable MFA Delete
    b. Upload objects
    c. Try to delete the objects and bucket
    d. Give the reason If you are enable to delete.
    e. Disable MFA Delete
    f. Now try to delete the objects

AWS S3 Default Encryption

- When you set the Default encryption option on a bucket, it enables server-side encryption.
- Amazon S3 encrypts your object before it saves the object to disk.
- Amazon S3 will decrypt it when you download the object.

Hands-on: AWS S3 Default Encryption

Task: Enable Default Encryption on an S3 Bucket.

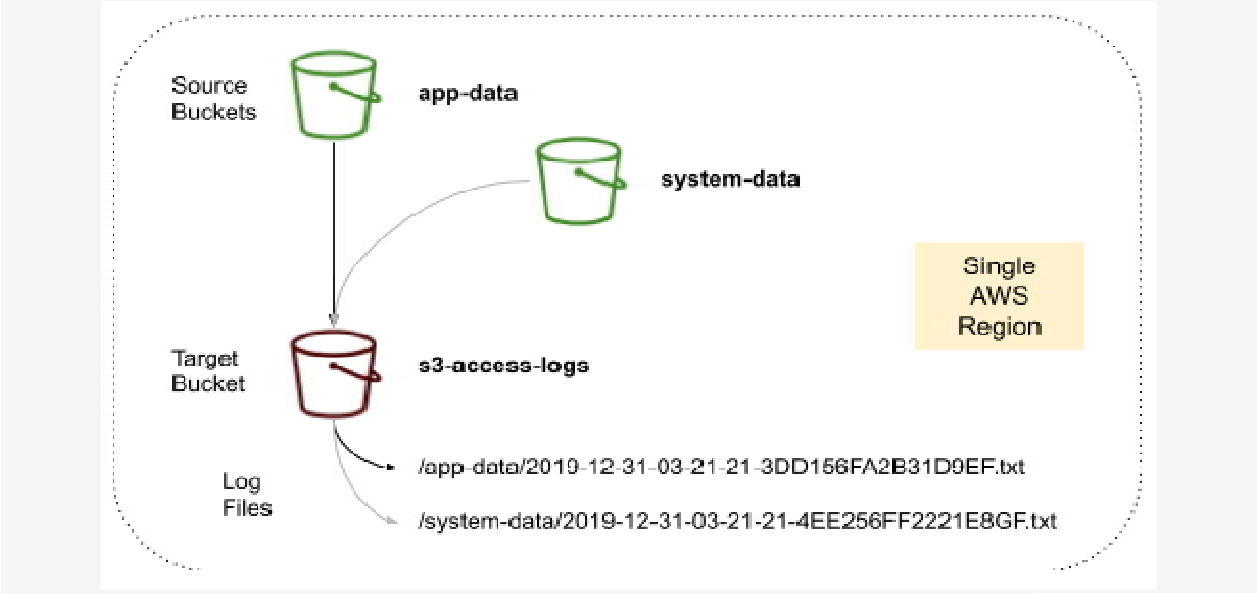
- Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
- In the Buckets list, **choose** the name of the bucket that you want.
- Choose **Properties**.
- Under Default encryption, choose **Edit**.
- Choose **Enable**.
- To enable server-side encryption using an Amazon S3-managed key, under Encryption key type, choose Amazon S3 key (SSE-S3).
- To enable server-side encryption using an AWS KMS key, follow these steps:
    a. Under Encryption key type, choose AWS Key Management Service key (SSE-KMS).
    b. Under AWS KMS key choose one of the following:
        i. AWS managed key
        ii. Choose from your KMS root keys, and choose your KMS root key.
        iii. Enter KMS root key ARN, and enter your AWS KMS key ARN.
- To use S3 Bucket Keys, under **Bucket Key**, choose **Enable**.
- Choose Save changes.

AWS S3 Access Logs

- S3 Access Logs captures information on all requests made to a bucket, such as PUT, GET, and DELETE actions.
- Bucket access logging is a recommended security best practice that can help teams with upholding compliance standards or identifying unauthorized access to your data.
- Basic Terms:
    a. Source Bucket: The S3 bucket to monitor.
    b. Target Bucket: The S3 bucket that will receive S3 access logs from source buckets.

c. Access Logs: Information on requests made to your buckets.

- S3 bucket access logging is configured on the source bucket by specifying a target bucket and prefix where access logs will be delivered.
- It's important to note that target buckets must live in the same region and account as the source buckets.
- Following figure illustrates S3 Access logging:



Hands-on: AWS S3 Access Logs

Task: Enable S3 Access logging

- Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
- In the **Buckets** list, choose the name of the bucket that you want to enable server access logging for.
- Choose **Properties**.
- In the **Server access logging** section, choose **Edit**.
- Under **Server access logging**, select **Enable**.
- For **Target bucket**, enter the name of the bucket that you want to receive the log record objects.The target bucket must be in the same Region as the source bucket.
- Choose **Save changes**.

Hands-on Assignments

1. Perform the following Tasks:
    a) Create two bucket (Source and Target)
    b) Upload two objects in Source Bucket.
    c) Enable Server Access logging on source bucket to store log entries in destination bucket
    d) Perform some activities on source bucket like upload object, delete object etc.
    e) Observe whether log entry files are created in Destination bucket

AWS S3 Replication

- S3 Replication enables automatic copying of objects across Amazon S3 buckets.
- We can replicate objects to a single destination bucket or to multiple destination buckets.
- The destination buckets can be in different AWS Regions or within the same Region as the source bucket.
- Here are the two S3 storage replication options:
    - **Cross-Region Replication (CRR)**—copies S3 objects across multiple Regions.
    - **Same-Region Replication (SRR)**—copies S3 objects between buckets in different availability zones (AZs), which are in the Region.
- To enable SRR or CRR, you add a replication configuration to your source bucket which includes:
    - The destination buckets – The bucket or buckets where you want Amazon S3 to replicate the objects.
    - The objects that you want to replicate

Hands-on: AWS S3 Replication

Task: Configuration of S3 Replication

- Go to the AWS S3 management console, sign in to your account, and select the name of the source bucket.
- Go to the **Management** tab in the menu, and choose the **Replication** option. Next, choose **Add rule.**

- Under the **Set resource** configuration, choose the **Entire bucket** option.
- Under the **Set destination** configuration, choose the **Buckets in this account** option. If you wish to replicate to another account, select this option and specify a bucket policy for the destination.
- To change the storage class of the object after replication, go to the **Destination options** configuration, and select a different storage class for the destination objects.
- You can also set the replication time.

Hands-on Assignments

1. Perform the following Tasks:
    a. Create two versioning disabled buckets (Source and Destination)
    b. Upload two objects in source bucket
    c. Create replication rule to copy all objects in source bucket to destination bucket.
    d. Now Observe destination bucket whether objects are replicated here from source bucket.
2. Perform the following Tasks:
    a. Create two versioning enabled buckets (Source and Destination)
    b. Upload two objects (one text file and one image file) in source bucket
    c. Create replication rule to copy all text file objects in source bucket to destination bucket.
    d. Now Observe destination bucket whether only text file objects are replicated here from source bucket.
3. Perform the following Tasks:
    a. Create three versioning enabled buckets (Source, D1 and D2)
    b. Upload two objects (one text file and one image file) in source bucket
    c. Create replication rule to copy all file objects in source bucket to D1 bucket.
    d. Create replication rule to copy all text file objects in source bucket to D2 bucket.
    e. Now Observe D1 bucket whether all objects are replicated here from source bucket.

f. Now Observe D2 bucket whether only text file objects are replicated here from source bucket.

AWS S3 Pre-signed URL

- All objects and buckets are private by default.
- You can use pre-signed URLs to generate a URL that can be used to access your Amazon S3 buckets.
- Pre-signed URL can be used to share the objects.
- Pre-signed URL allow your customers/users to access objects or upload objects to buckets without AWS security credentials or permissions.
- When you create a pre-signed URL, you associate it with a specific action (Share/upload object).
- When you create a pre-signed URL for your object, you must provide your security credentials and then specify a bucket name, an object key, an HTTP method (GET to download the object), and an expiration date and time.
- Anyone who receives the pre-signed URL can then access the object.
- For example, if you have a video in your bucket and both the bucket and the object are private, you can share the video with others by generating a pre-signed URL.
- Because pre-signed URLs grant access to your Amazon S3 buckets to whoever has the URL, we recommend that you protect them appropriately.
- The URL will expire and no longer work when it reaches its expiration time.

Hands-on: AWS S3 Pre-signed URL

Task: Generate S3 Pre-signed URL for your object.

- Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
- In the Buckets list, choose the name of the bucket that contains the object that you want a pre-signed URL for.
- In the Objects list, select the object that you want to create a pre-signed URL for.
- On the Actions menu, choose Share with a pre-signed URL.
- Specify how long you want the pre-signed URL to be valid.

- Choose Create pre-signed URL.
- When a confirmation appears, the URL is automatically copied to your clipboard. You will see a button to copy the pre-signed URL if you need to copy it again.

Hands-on Assignments

1. Perform the following Tasks:
    f) Create a bucket
    g) Upload two objects in it.
    h) Access any specific object using object URL
    i) Are you able to access the object?
    j) What options we have to make the object accessible??
2. Perform the following Tasks:
    a) Create a bucket
    b) Upload two objects (a.txt and b.txt) in it.
    c) Generate Pre-signed URL for object a.txt and it should be accessible only for 2 minutes and observe it.
    d) Generate Pre-signed URL for object b.txt and it should beaccessible only for 5 minutes and observe it.

AWS S3 Storage Classes + Glacier

Amazon S3 offers following storage classes that are designed for different use cases.

1. Amazon S3 Standard – Amazon S3 Standard is designed for frequently accessed data. Because it delivers low latency and high throughput.
2. Amazon S3 Intelligent – Tiering – The Amazon S3 Intelligent - Tiering storage class is designed to optimize costs by automatically moving data to the most cost - effective

access tier, moves the objects that have not been accessed for 30 consecutive days to the infrequent access tier. If an object in the infrequent access tier is accessed, it is automatically moved back to the frequent access tier. No additional fees when objects are moved between access tiers.

3. Amazon S3 Standard - Infrequent Access (Amazon S3 Standard-IA) – The Amazon S3 Standard-IA storage class is used for data that is accessed less frequently, but requires rapid access when needed.
4. Amazon S3 One Zone - Infrequent Access (Amazon S3 One Zone - IA) – Amazon S3 One Zone- IA is for data that is accessed less frequently, but requires rapid access when needed. Unlike other Amazon S3 storage classes, which store data in a minimum of three Availability Zones, Amazon S3 One Zone - IA stores data in a single Availability Zone and it costs less than Amazon S3 Standard - IA.
5. Amazon S3 Glacier – Amazon S3 Glacier is a secure, durable, and low-cost storage class for data archiving.
6. Amazon S3 Glacier Deep Archive – Amazon S3 Glacier Deep Archive is the lowest - cost storage class for Amazon S3. It supports long - term retention and digital preservation for data that might be accessed once or twice in a year.

Hands On: S3 Storage Classes

Task 1: Upload object in a specific storage Class

Task 2: Change the Storage class of object

Task 1: Upload Object in a specific storage class

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the Buckets list, **choose** the name of the bucket that you want to upload your object to.
3. On the Objects tab for your bucket, choose **Upload**.
4. Under Files and folders, choose **Addfiles**.
5. Choose a file to upload, and then choose **Open**.
6. Under Properties, Choose the storage class.
7. Choose **Upload**.You've successfully uploaded an object to your bucket.

Task 2: Change the storage class of object

7. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
8. In the Buckets list, choose the name of the bucket that you want to change the storage class of an object.
9. Select the check box to the left of the names of the objects that you want to change storage class.
10. Choose Actions and choose edit Storage class.
11. Select the storage class to which you want to change.
12. Choose Save Changes.

Hands-on Assignments

1. Perform the following Tasks:
    a. Create a bucket
    b. Upload object (a.txt) in S3 Standard Storage Class.
    c. Upload object (b.txt) in S3 Standard IA Storage Class.
    d. Upload object (c.txt) in S3 Glacier Storage Class.
    e. Change the storage class of object (b.txt) to S3 Standard Storage Class

AWS S3 Lifecycle Rules

- You should automate the lifecycle of the data that you store in Amazon S3.
- By using lifecycle policies, you can transmit data at regular intervals between different Amazon S3 storage types.
- This automation reduces your overall cost, because you pay less for data as it becomes less important with time.
- In addition to setting lifecycle rules per object, you can also set lifecycle rules per bucket.
- Consider an example of a lifecycle policy that moves data as it ages from Amazon S3 Standard to Amazon S3 Standard – Infrequent Access, and finally, into Amazon S3 Glacier before it is deleted.
- Suppose that a user uploads a video to your application and your application generates a thumbnail preview of the video. This video preview is stored to Amazon S3 Standard, because it is likely that the user wants to access it right away.

- Your usage data indicates that most thumbnail previews are not accessed after 30 days. Your lifecycle policy takes these previews and moves them to Amazon S3 – Infrequent Access after 30 days. After another 30 days' elapse, the preview is unlikely to be accessed again. The preview is then moved to Amazon S3 Glacier, where it remains for 1 year. After 1 year, the preview is deleted.
- The important thing is that the lifecycle policy manages all this movement automatically.

Hands-on: AWS S3 Lifecycle Rules

Task: Configuration of S3 Lifecycle

- Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
- In the Buckets list, choose the name of the bucket that you want to create a lifecycle rule for.
- Choose the Management tab, and choose Create lifecycle rule.
- In Lifecycle rule name, enter a name for your rule.The name must be unique within the bucket.
- Choose the scope of the lifecycle rule:
    - ✓ To apply this lifecycle rule to *all objects with a specific prefix or tag*, choose Limit the scope to specific prefixes or tags.
    - ✓ To apply this lifecycle rule to *all objects in the bucket*, choose This rule applies to *all* objects in the bucket, and choose I acknowledge that this rule applies to all objects in the bucket.
- To filter a rule by object size, you can check Specify minimum object size, Specify maximum object size, or both options.
- Under Lifecycle rule actions, choose the actions that you want your lifecycle rule to perform:
    - ✓ Transition *current* versions of objects between storage classes
    - ✓ Transition *previous* versions of objects between storage classes
    - ✓ Expire *current* versions of objects
    - ✓ Permanently delete *previous* versions of objects

- ✓ Delete expired delete markers or incomplete multipart uploads

Depending on the actions that you choose, different options appear.

- To transition *current* versions of objects between storage classes, under Transition current versions of objects between storage classes:
  - ✓ In Storage class transitions, choose the storage class to transition to:
    - Standard-IA
    - Intelligent-Tiering
    - One Zone-IA
    - S3 Glacier Flexible Retrieval
    - Glacier Deep Archive
  - ✓ In Days after object creation, enter the number of days after creation to transition the object.
- To transition *non-current* versions of objects between storage classes, under **Transition non-current versions of objects between storage classes**:
  - ✓ In **Storage class transitions**, choose the storage class to transition to:
    - Standard-IA
    - Intelligent-Tiering
    - One Zone-IA
    - S3 Glacier Flexible Retrieval
    - Glacier Deep Archive
  - ✓ In **Days after object becomes non-current**, enter the number of days after creation to transition the object.
- To expire *current* versions of objects, under **Expire previous versions of objects**, in **Number of days after object creation**, enter the number of days.
- To permanently delete previous versions of objects, under **Permanently delete noncurrent versions of objects**, in **Days after objects become noncurrent**, enter the number of days. You can optionally specify the number of newer versions to retain by entering a value under **Number of newer versions to retain**.
- Under **Delete expired delete markers or incomplete multipart uploads**, choose **Delete expired object delete markers** and **Delete incomplete multipart uploads**. Then, enter

the number of days after the multipart upload initiation that you want to end and clean up incomplete multipart uploads.

- Choose **Create rule**.

Hands-on Assignments

4. Perform the following Tasks:
    a. Create a bucket
    b. Upload object (a.txt) in S3 Standard Storage Class.
    c. Add Lifecycle Configuration rule for the transition of object (a.txt) as follows.
        i. Move object to S3 Standard IA Storage Class after 30 days
        ii. Move object to S3 Glacier Storage Class after 90 days
        iii. Move object to S3 Glacier Storage Class after 150 days

AWS Athena

- AWS Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL.
- AWS Athena is commonly used for querying and analysis of data stored in S3, especially for log analysis, data exploration, etc.
- It allows you to directly run SQL queries on data stored in Amazon S3 without the need for infrastructure management.

Here are some key features and components of AWS Athena:

- Serverless Query Service: Athena is a serverless service, which means there is no infrastructure to manage, and you pay only for the queries you run.
- Integration with Amazon S3: Athena integrates with Amazon S3, allowing you to run queries directly on data stored in S3 buckets.
- Standard SQL Support: Athena supports standard SQL, making it familiar to anyone with SQL querying experience.

AWS Snow Family

- The AWS Snow Family is a collection of physical devices offered by Amazon Web Services (AWS) designed to help customers move large amounts of data offline securely.
- It provides a secure, reliable, and efficient method for transferring large datasets to the AWS cloud.
- These devices are useful in situations where it might not be practical or feasible to move data directly over the internet.
- The Snow Family devices are tamper-resistant and use encryption to secure data during transit.
- They include built-in tracking and allow you to monitor the status of data transfers using the AWS Management Console.
- The Snow Family includes the following devices:
    - ✓ AWS Snowcone:** Snowcone is the smallest and most portable member of the Snow Family. It is rugged and secure, and it can transfer small amount of data.
    - ✓ AWS Snowball:** Snowball is a larger data transport solution that comes in two options: the 50 TB Snowball and the 80 TB Snowball Edge. These devices are designed for large-scale data migrations securely.
    - ✓ AWS Snowmobile: Snowmobile is an exabyte-scale data transfer service that can transfer up to 100 PB per transfer. It is a secure data truck designed to move extremely large amounts of data to AWS. It is typically used for massive data center migrations.

Amazon FSx

Amazon FSx is a fully managed file storage service that makes it easy to launch and run file systems.

Key features and benefits of Amazon FSx include:

- Fully Managed Service: Amazon FSx is a fully managed service that automates time-consuming administration tasks such as hardware provisioning, software configuration, patching, and backups.

- Compatibility: It offers compatibility with popular file systems like Lustre, Windows File Server, etc.
- High Performance: Amazon FSx provides high performance and throughput, making it suitable for a wide range of workloads.
- Integration with AWS Services: Amazon FSx integrates with other AWS services, allowing you to use it as shared storage for your EC2 instances and containers.
- Data Durability and Security: Amazon FSx offers data durability and security through automated backups, data encryption at rest, etc.

Amazon Storage Gateway

- AWS Storage Gateway is a hybrid cloud storage service that provides secure interface between on-premises environments and AWS cloud storage.
- It enables you to securely connect your on-premises applications to AWS storage services, including Amazon S3, Amazon S3 Glacier, Amazon EBS, and Amazon EFS.

AWS Storage Gateway offers three types of storage interfaces:

- File Gateway: This gateway provides a file interface, which allows you to store and retrieve objects in Amazon S3 using standard file protocols.
- Volume Gateway: Volume Gateway provides block storage volumes. It offers two configurations: stored volumes, which store your primary data locally while asynchronously backing up the data to AWS, and cached volumes, which store your primary data in Amazon S3 while retaining frequently accessed data locally.
- Tape Gateway: Tape Gateway provides a virtual tape infrastructure, allowing you to store virtual tapes in AWS.