Amazon
Simple Storage Service (S3)

Basic Terms that we should be aware before we step into S3 bucket learning modules

1. Buckets:

- A bucket is a container for storing objects (files). Each bucket must have a globally unique name and can contain an unlimited number of objects.

2. Objects:

- Objects are the fundamental entities stored in S3. Each object consists of data (the file) and metadata (information about the file). Objects are identified within a bucket by a unique key (or name).

3. Keys:

- A key is the unique identifier for an object within a bucket. It is essentially the object's filename within the bucket.

4. Regions:

- AWS S3 stores data in multiple locations known as regions. Each region is a separate geographic area, and choosing the right region is crucial for optimizing performance and ensuring compliance with regulations.

5. Versioning:

- Versioning allows you to keep multiple versions of an object in the same bucket, providing a way to preserve, retrieve, and restore every version of every object stored.

6. Storage Classes:

- S3 offers different storage classes that determine the cost and availability of your data:
    - **S3 Standard**: High durability, availability, and performance for frequently accessed data.
    - **S3 Intelligent-Tiering**: Automatically moves data to the most cost-effective access tier.
    - **S3 Standard-IA** (Infrequent Access): For data that is accessed less frequently, but requires rapid access when needed.
    - **S3 Glacier** and **S3 Glacier Deep Archive**: For long-term archive data with varying retrieval times.

7. Access Control:

- S3 provides multiple mechanisms to control who can access your data:
    - **Bucket Policies**: Define permissions on a bucket-wide level.
    - **Access Control Lists (ACLs)**: Provide more granular control at the object level.
    - **IAM Policies**: Attach policies to AWS IAM users, groups, or roles to control their access to S3 resources.

8. Data Transfer:

- AWS S3 supports different methods for data transfer, including:
    - **S3 Transfer Acceleration**: Uses Amazon CloudFront's globally distributed edge locations to accelerate data transfer.
    - **Multipart Upload**: Allows you to upload large objects as a series of parts, which can be uploaded independently.

9. Lifecycle Policies:

- These policies automatically transition objects between different storage classes or delete them after a specified period, helping to manage costs.

10. Cross-Region Replication (CRR):

- Allows you to replicate objects from one bucket to another in a different AWS region, providing disaster recovery and data sovereignty.

11. Event Notifications:

- S3 can send notifications when certain events happen in your bucket, like object creation or deletion. These notifications can trigger AWS Lambda functions, send messages to an SNS topic, or post messages to an SQS queue.

12. Encryption:

- S3 supports various encryption methods to protect your data, including:
    - **Server-Side Encryption (SSE)**: AWS handles the encryption and decryption process.
    - **Client-Side Encryption**: You encrypt data before uploading it to S3.

13. Object Lock:

- S3 Object Lock allows you to store objects using a write-once-read-many (WORM) models, helping to prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely.

14. Requester Pays:

- A feature where the requester rather than the bucket owner pays for data access charges.

Let's start

Amazon Simple Storage Service (S3) is a scalable, high-performance object storage service that provides industry-leading durability, availability, security, and performance. S3 is designed to store and retrieve any amount of data from anywhere on the web, making it ideal for a wide range of use cases, including backup and restore, disaster recovery, data archiving, big data analytics, content distribution, and more.

Key Parameters and Notes

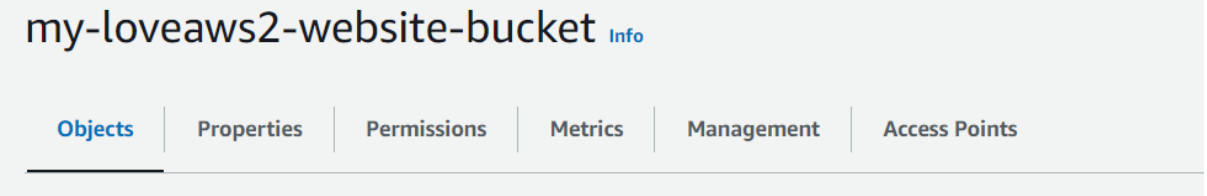
1. **Scalability**:
    - **Virtually Unlimited Storage**: S3 scales to accommodate any amount of data without requiring you to provision or manage storage.
    - **Automatic Scaling**: S3 automatically scales to handle any level of demand, from a few requests per day to millions per second.
2. **Durability and Availability**:
    - **11 9's of Durability (99.999999999%)**: S3 is designed for 99.999999999% durability, meaning your data is highly protected and unlikely to be lost.
    - **High Availability**: S3 provides 99.99% availability over a given year, ensuring that your data is accessible when you need it.

3. **Global Infrastructure:**
    - **Regional Data Centers:** S3 is deployed in regions around the world, enabling you to store data close to your customers for lower latency and compliance.
    - **Cross-Region Replication:** Automatically replicate data across regions to enhance availability and disaster recovery capabilities.
4. **Security:**
    - **Encryption:** S3 offers various encryption options, including Server-Side Encryption (SSE) with S3-Managed Keys (SSE-S3), AWS Key Management Service (SSE-KMS), and Customer-Provided Keys (SSE-C).
    - **Access Control:** S3 integrates with AWS Identity and Access Management (IAM) and supports bucket policies, Access Control Lists (ACLs), and encryption to ensure secure data access and management.
    - **Compliance:** S3 is compliant with various industry standards, including HIPAA, PCI-DSS, and GDPR.
5. **Data Management Features:**
    - **Versioning:** Enables you to keep multiple versions of an object, protecting against accidental deletion or overwriting.
    - **Lifecycle Policies:** Automate data management by transitioning objects to different storage classes or deleting them after a specific period.
    - **Intelligent-Tiering:** S3 automatically moves data between two access tiers (frequent and infrequent access) based on changing access patterns, optimizing cost without performance impact.
6. **Cost-Effectiveness:**
    - **Pay-As-You-Go Pricing:** S3 pricing is based on the amount of data stored, data transfer, and the number of requests, ensuring you only pay for what you use.
    - **Storage Classes:** Multiple storage classes (Standard, Intelligent-Tiering, Standard-IA, Glacier, etc.) allow you to optimize costs based on how frequently you access your data.

7. **Data Transfer and Integration**:
    - **Multipart Upload**: S3 supports uploading large files in parts, which can be uploaded independently, improving efficiency and reliability.
    - **Integration with AWS Services**: S3 integrates seamlessly with a wide range of AWS services such as Lambda, Snowball, CloudFront, and Data Pipeline, enabling you to build scalable and robust architectures.
8. **Event Notifications**:
    - **Triggers**: S3 can trigger notifications to AWS Lambda, SNS, or SQS when specific events occur, such as object creation or deletion, enabling event-driven workflows.
9. **Performance**:
    - **High Throughput and Low Latency**: S3 is designed for high performance with low latency, ensuring fast data access and retrieval.
    - **S3 Transfer Acceleration**: Speeds up data transfers by routing data through Amazon CloudFront edge locations.
10. **Use Cases**:
    - **Data Lake**: S3 serves as a foundational building block for data lakes, enabling you to store and analyze vast amounts of data.
    - **Backup and Restore**: S3's durability and availability make it an ideal solution for backing up critical data.
    - **Content Distribution**: S3 is commonly used to store and distribute content like media files, software, and websites.
    - **Big Data Analytics**: S3 integrates with AWS analytics services, making it easier to process and analyze large datasets.

For reference – you created a bucket & it has different tabs underneath it →

Objects, Properties, Permissions, Metrics, Management & Access Points.



Now – we will learn about each tab and its options

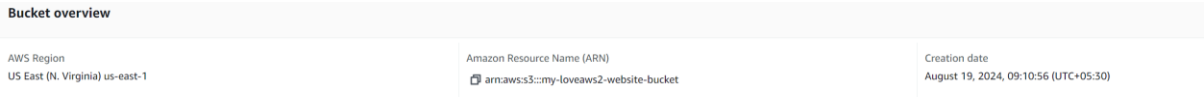
Tab: Objects

Objects → as we know the definition earlier, this is where all your files appear, there are no options in the Tab but you can select the Object and click on actions button to modify the objects properties.

Tab: Properties

Properties →

Option 01: Bucket Overview – the information of the bucket can be seen over this option.



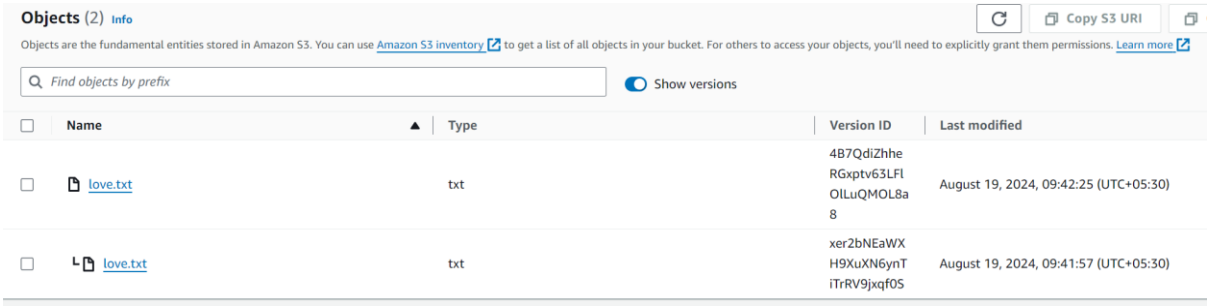
Option 02: Bucket Versioning (It has multi-factor authentication (MFA) delete option as well)

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

Practical: 01

The point needs to keep in mind

The "L" next to an old version of a file indicates a delete marker, signifying that the file has been logically deleted but still exists in a versioned state. Understanding delete markers helps in managing object deletions and restorations in S3 buckets with versioning enabled.



Now other Option - ***Multi-factor authentication (MFA) delete*** , what it denotes – from the UI , this is always in Disable Mode

Multi-Factor Authentication (MFA) Delete is an additional security feature provided by Amazon S3 to protect your objects and bucket configurations from accidental or malicious deletions. It ensures that any delete operations on your objects or bucket configurations require additional authentication, specifically through MFA.

MFA Delete can only be enabled or disabled using the AWS CLI or SDKs. It is not possible to enable MFA Delete directly from the AWS Management Console UI.

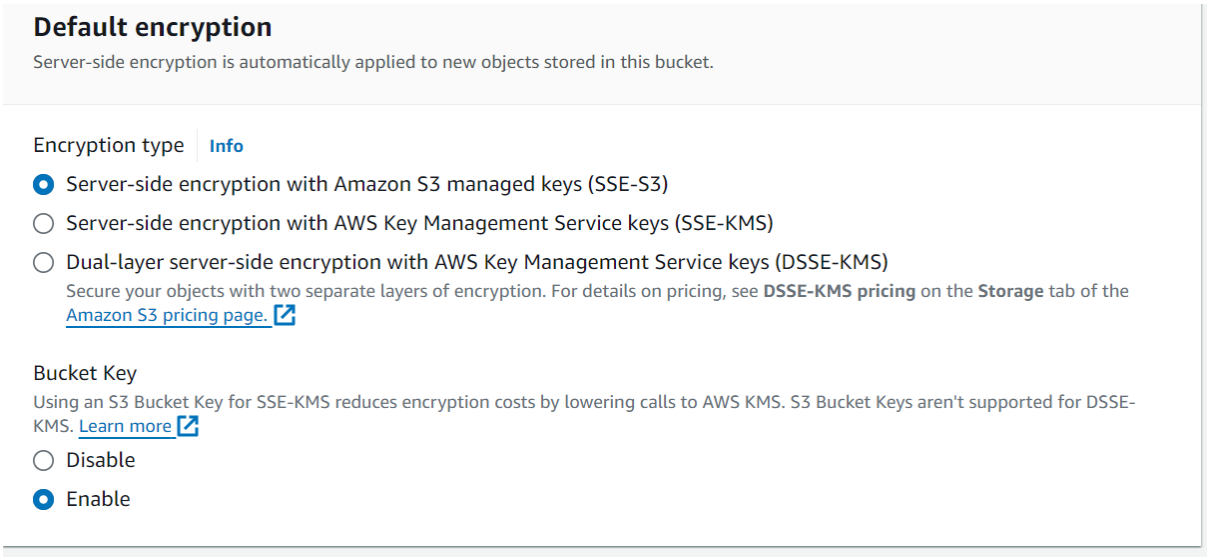
Now we move onto next option:

Option 03 → Tags: Unique naming Identifiers as we have been seeing from the AWS Day 00 class.

Amazon
Simple Storage Service (S3)

Option 04: Default encryption →

Default Encryption in AWS S3 is a feature that automatically encrypts all new objects uploaded to an S3 bucket. It ensures that your data is encrypted at rest by default, providing an additional layer of security without requiring you to manually encrypt each object.



As per above Image – there are 3 types

1. Server-Side Encryption with S3-Managed Keys (SSE-S3)
2. Server-Side Encryption with AWS Key Management Service (SSE-KMS)

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

3. Dual-Layer Server-Side Encryption with AWS Key Management Service (DSSE-KMS)

Server-Side Encryption: Encrypts data at rest within S3. Includes SSE-S3, SSE-KMS, and SSE-C.

Client-Side Encryption: Encrypts data before it is uploaded to S3. Includes encryption with AWS KMS or customer-provided keys, as well as custom encryption methods.

Amazon
Simple Storage Service (S3)

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

- **Encryption Method**: Uses keys managed by S3.
- **Details**: AWS S3 handles the encryption and decryption of data with its own keys.
- **Use Case**: Suitable for most use cases where simplicity is desired, and you don't need control over the encryption keys.

Suitability:

- **Server-Side**: Suitable for general use cases where simplicity is preferred and you don't need to manage encryption keys.
- **Not for Client-Side**: This is a server-side encryption method and does not apply to client-side encryption.

Parameters

- AES-256 encryption & HTTP or HTTPS can be used

- Must set header: `"x-amz-server-side-encryption": "AES256"`

Server-Side Encryption with AWS Key Management Service (SSE-KMS)

- **Encryption Method**: Uses keys managed by AWS Key Management Service (KMS).
- **Details**: Provides more control over encryption keys. You can create, manage, and control access to the keys used for encryption.
- **Use Case**: Ideal for use cases requiring more granular control over keys and compliance with strict regulatory requirements. Supports features like key rotation and access policies.

Suitability:

- **Server-Side**: Ideal for use cases where additional control over encryption keys and auditing is required.
- **Client-Side**: Can be used in conjunction with client-side encryption if data is first encrypted on the client side before uploading. AWS SDKs support this integration.

Parameters

- HTTP or HTTPS can be used
- KMS provides control over who has access to what keys as well as audit trails
- Envelope encryption is used in this case. The user uploading the file must have kms:GenerateDataKey permission.
- Headers to be set

  "x-amz-server-side-encryption": "aws:kms"

  “x-amz-server-side-encryption-aws-kms-key-id”

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

- **Encryption Method**: Uses keys provided by the customer.
- **Details**: You provide your own encryption keys. AWS S3 does not store or manage the keys.
- **Use Case**: Best for scenarios where you want to manage encryption keys entirely on your own. Requires handling key management and security yourself.

  Suitability:

- **Server-Side:** Useful if you need to manage your own keys but still want S3 to handle encryption and decryption.
- **Client-Side:** Not typically used for client-side encryption. It’s a server-side method where you provide the key during interaction.

  Parameters

- Keys managed by the client
- Client sends the key in HTTPS headers for encryption/decryption (S3 discards the key after the operation)
- **If the client loses the key, they cannot decrypt the object**
- **HTTPS must be used** as key (secret) is being transferred

Amazon
Simple Storage Service (S3)

Dual-Layer Server-Side Encryption with AWS Key Management Service (DSSE-KMS)

Dual-Layer Server-Side Encryption with AWS Key Management Service (DSSE-KMS) is a specific encryption approach used by Amazon S3 that adds an additional layer of security by combining two encryption mechanisms:

- **Server-Side Encryption with AWS Key Management Service (SSE-KMS)**: This method encrypts your data using AWS KMS keys. It provides enhanced control over encryption keys and allows you to manage key policies and perform key rotation.
- **Additional Encryption Layer**: DSSE-KMS adds another layer of encryption on top of the data encrypted by SSE-KMS. This involves encrypting the data again using another key or mechanism, adding a second layer of encryption to further protect the data.
- This one tilt towards **Server-Side Encryption**

Enforcing Encryption:

- **Default encryption**: encrypt the files using the default encryption (specify the encryption for the file while uploading to override the default)
- Bucket policy can be used to force a specific type of encryption on the objects uploaded to S3 </aside>

Encryption in Transit

- Enforce HTTPS connection by creating an S3 bucket policy that denies incoming request where **SecureTransport** is **false**

Amazon
Simple Storage Service (S3)

Option 05 → Intelligent-Tiering Archive configurations

Intelligent-Tiering Archive Configurations provide a way to optimize storage costs for data with unpredictable access patterns by automatically transitioning data between access tiers, including a lower-cost Archive Access Tier for long-term storage. This approach helps in managing storage costs efficiently while retaining data for compliance or archival needs.

Imagine you have a large dataset that includes both frequently accessed and rarely accessed data. With Intelligent-Tiering:

- **Frequent Access**: Your frequently accessed data remains in the Frequent Access tier for low latency and quick retrieval.
- **Infrequent Access**: Data accessed less often moves to the Infrequent Access tier, reducing storage costs.
- **Archive Access**: Data that becomes even less frequently accessed, such as old records or historical data, transitions to the Archive Access tier for long-term, cost-effective storage.

Summary

- 99.9% availability
- Moves objects automatically between Access Tiers based on usage
- Small monthly monitoring and auto-tiering fee
- No retrieval charges

Option 06: Server access logging

Server Access Logging in Amazon S3 captures comprehensive details about requests made to your S3 buckets, including requester information, request type, response status, and more. This information is crucial for monitoring, auditing, troubleshooting, and optimizing the use of your S3 resources.

Amazon
Simple Storage Service (S3)

Option 07: AWS CloudTrail Data Events

AWS CloudTrail Data Events provide in-depth visibility into the actions performed on your AWS resources, offering critical insights for security, compliance, troubleshooting, and performance management. By capturing detailed logs of data operations, CloudTrail helps you maintain a secure and well-managed AWS environment.

Below ones can be captured specific to AWS S3

Amazon S3 Data Events:

- **Object-Level Operations:**
  - **GET:** Accessing an object in S3.
  - **PUT:** Uploading or updating an object in S3.
  - **DELETE:** Removing an object from S3.
  - **LIST:** Listing objects in a bucket.
  - **COPY:** Copying an object within S3 or between buckets.
- **Event Details:**
  - **Object Key:** The name of the object involved in the request.
  - **Bucket Name:** The name of the bucket containing the object.
  - **Requester:** The IAM user or role making the request.
  - **Request IP Address:** The IP address from which the request originated.
  - **Response Status:** The HTTP status code returned by S3.

Option 08: Event Notifications

Event notifications in AWS are essential for real-time alerts, automated responses, security monitoring, application management, and cost management. By configuring and using event notifications effectively, you can enhance operational efficiency, maintain security, and ensure smooth running of your AWS resources and applications.

Amazon S3 Event Notifications:

- **Events:** Notifications for object-level changes such as PUT, POST, COPY, DELETE, and others.
- **Use Cases:** Trigger workflows or Lambda functions for processing files, or send notifications when files are uploaded.

Option 09: Amazon EventBridge

Integrating Amazon EventBridge with S3 provides enhanced event routing capabilities, flexible processing, and the ability to build complex, decoupled architectures. It enables you to set up sophisticated event-driven workflows, automate tasks, and integrate with various AWS services, improving overall system efficiency and responsiveness.

Option 10: Transfer acceleration

Amazon S3 Transfer Acceleration enhances data transfer speeds by routing traffic through Amazon CloudFront's global edge network. It's particularly useful for improving performance when dealing with large files or users located far from your S3 bucket's region. By enabling Transfer Acceleration, you can optimize your data transfer times and provide a better user experience for global users.

When you enable Transfer Acceleration on an S3 bucket, data is routed to the nearest CloudFront edge location, where it is then forwarded to your S3 bucket.

Your Location: India

S3 Bucket Location: US East (N. Virginia)

Without Transfer Acceleration:

You would upload files directly to the S3 bucket in US East, which involves data traveling over the public internet all the way from India to the US. The transfer speed could be limited by the internet conditions, such as latency, packet loss, and available bandwidth.

With Transfer Acceleration:

Instead of sending the data directly to the US, your data is first sent to a nearby AWS edge location, like the one in Mumbai. The edge location receives the data quickly due to its proximity.

The data is then transferred from the edge location to the S3 bucket in the US over AWS's optimized global network, leading to a faster and more efficient transfer.

Option 11: Object Lock

Amazon S3 Object Lock is a feature that allows you to prevent objects in your S3 bucket from being deleted or overwritten for a specified retention period. It is primarily used to enforce data retention policies, ensuring that data cannot be altered or removed before the retention period expires. This feature is especially important for compliance and data protection purposes.

Governance Mode:

- **Purpose**: Allows users with special permissions to delete or overwrite objects, but prevents accidental deletions or modifications by ordinary users.
- **Use Case**: Useful in scenarios where you need to protect data from accidental changes or deletions but still need some flexibility for authorized users.

Compliance Mode:

- **Purpose**: Prevents all users, including those with administrative permissions, from deleting or overwriting objects until the retention period expires.
- **Use Case**: Essential for regulatory compliance requirements, where data must be retained unaltered for a specific period (e.g., legal or financial records).

Key Points:

- **Non-Editable After Creation**: Once Object Lock is enabled on a bucket, it cannot be disabled or modified. Ensure you configure it correctly during bucket creation.
- **Costs**: Object Lock does not incur additional costs beyond the standard S3 storage charges, but consider any potential impacts on your data management and retention policies.

Option 12: Requester pays

Requestor Pays is a feature in Amazon S3 that allows you to configure your S3 bucket so that the requester, rather than the bucket owner, is responsible for paying the data transfer costs associated with accessing the objects in the bucket. This can be useful in scenarios where you want to share data publicly or with specific users but shift the cost burden to those accessing the data.

Option 13: Static Website Hosting

Static Website Hosting in Amazon S3 is a powerful and cost-effective solution for serving static content to users. It simplifies website deployment, provides high availability and durability, and scales automatically with traffic. By using S3 for static website hosting, you can efficiently deliver your website content while minimizing infrastructure management and costs.

Amazon
Simple Storage Service (S3)

- Host static websites (may contain **client-side scripts**) and have them accessible on the public internet over **HTTP only** (for HTTPS, use CloudFront with S3 bucket as the origin)
- The website URL will be either of the following:
    - <bucket-name>.s3-website-<region>.amazonaws.com
    - <bucket-name>.s3-website.<region>.amazonaws.com
- If you get a 403 Forbidden error, make sure the bucket policy allows public reads
- For cross-origin access to the S3 bucket, we need to enable CORS on the bucket

We will do a practical site building anyway ..!!

Amazon
Simple Storage Service (S3)

Next Tab: Permissions Tab

Option 01: Permission Overview

Usual overview tab for the information showcase

Option 02: Bucket Public Access

Bucket Public Access settings in Amazon S3 are crucial for managing and securing your data. They help prevent unauthorized public access, protect sensitive information, and ensure compliance with data privacy regulations. By configuring and managing these settings properly, you can maintain control over who can access your S3 data and reduce the risk of accidental data exposure.

Object ACLs (Access Control Lists): Individual objects have their own ACLs that can specify permissions for different users or groups. Object ownership affects how these ACLs are applied and managed.

Types of Bucket Public Access Settings

1. **Block Public Access Settings:**
    - **Block All Public Access:** Prevents any public access to the bucket and its objects, regardless of the bucket policy or object ACLs. This is the most restrictive setting and ensures that data is kept private.
    - **Block Public Access to Buckets and Objects Granted through ACLs:** Prevents public access that might be granted through object-level ACLs.
    - **Block Public Access to Buckets and Objects Granted through Policies:** Prevents public access that might be granted through bucket policies.
    - **Block Public Access to Buckets and Objects Granted through ACLs (Explicit):** Prevents public access through explicit object ACL settings.
2. **Public Access Block Configuration:**
    - **Effective Policy:** The settings you choose apply at the account or bucket level, and they take precedence over any bucket policies or ACL settings that might allow public access.

Amazon
Simple Storage Service (S3)

Option 03: Bucket Policy

Bucket policies in Amazon S3 are a powerful tool for managing access to your S3 buckets and objects. They provide granular control over permissions, help enforce security and compliance, and integrate with other AWS services. By using bucket policies effectively, you can ensure that your data is secure and access is managed according to your needs and policies.

Reference

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::my-bucket/*"
    }
  ]
}
```

Option 04: Object Ownership

Object Ownership in Amazon S3 is a crucial aspect of managing access and control over the objects stored in a bucket. It determines who has control over the objects and how permissions are granted and enforced. By understanding and properly configuring object ownership, you can effectively manage access, ensure security, and maintain control over your S3 data.

Owner and ACLs:

- **Bucket Owner**: When objects are uploaded to an S3 bucket, the bucket owner is typically the owner of those objects. This means the bucket owner has full control over the objects, including managing access permissions.
- **Object ACLs (Access Control Lists)**: Individual objects have their own ACLs that can specify permissions for different users or groups. Object ownership affects how these ACLs are applied and managed.

Option 05: Access Control List (ACL)

An Access Control List (ACL) is a set of rules that define who can access specific S3 resources (buckets or objects) and what actions they can perform (e.g., read, write, delete).

Scope:

- **Bucket ACLs**: Control access to the bucket itself, including who can list the contents or perform other bucket-level operations.
- **Object ACLs**: Control access to individual objects within a bucket, determining who can read, write, or delete the objects.

Grants specify the permissions assigned to a user or group. Each grant includes the following components:

- **Grantee**: The entity (user, group, or service) to whom the permissions are granted.
- **Permission**: The specific action(s) allowed (e.g., READ, WRITE, FULL_CONTROL).

Reference

```
<AccessControlPolicy>
  <Owner>
    <ID>example-owner-id</ID>
    <DisplayName>example-owner-display-name</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUs
        <ID>example-canonical-user-id</ID>
        <DisplayName>example-user-display-name</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Owner:

- <ID>: The unique identifier of the owner (typically the AWS account ID).
- <DisplayName>: The human-readable name of the owner.

AccessControlList:

- **Grants**: Each <Grant> element specifies a grantee (an entity that is granted permissions) and the type of permission.
  - **Grantee**: This can be an AWS account (CanonicalUser), a group (e.g., AllUsers, AuthenticatedUsers), or an email address.
    - **CanonicalUser**: Refers to a specific AWS account.
    - **Group**: Refers to predefined groups in S3, like AllUsers (everyone) or AuthenticatedUsers (only authenticated AWS users).
    - **URI**: The URI for predefined groups.
  - **Permission**: Defines the level of access granted. Possible values include:
    - FULL_CONTROL: Grantee has full control over the object or bucket.
    - READ: Grantee can read the object or bucket.
    - WRITE: Grantee can write to the bucket (this does not apply to objects).
    - READ_ACP: Grantee can read the ACL.
    - WRITE_ACP: Grantee can write to the ACL.

Option 06: Cross-origin resource sharing (CORS)

Cross-Origin Resource Sharing (CORS) is an essential feature for managing how resources in S3 can be accessed from different domains. It allows web applications to interact with S3 resources while enforcing security controls to prevent unauthorized access. By configuring CORS properly, you ensure that your resources are accessible to legitimate users and applications while maintaining security and compliance.

Next tab: Metrics

Option 01: Bucket Metrics

Bucket metrics are essential for monitoring, managing, and optimizing your Amazon S3 storage. They provide valuable insights into performance, usage, and costs, helping you ensure that your S3 buckets are operating efficiently and effectively. By leveraging these metrics, you can make informed decisions, optimize resource usage, and maintain the reliability of your storage solutions.

How to Access Bucket Metrics

1. **Amazon CloudWatch:**
    - **Metrics:** Amazon S3 integrates with Amazon CloudWatch to provide detailed metrics and dashboards. You can access metrics through the CloudWatch console or API.
    - **Alarms:** Set up CloudWatch alarms to get notifications based on specific thresholds or conditions for your S3 metrics.
2. **S3 Management Console:**
    - **Metrics Tab:** Access bucket metrics through the S3 Management Console under the "Metrics" tab for each bucket. This provides a visual representation of key metrics.
3. **AWS CLI and SDKs:**
    - **API Calls:** Use the AWS CLI or SDKs to programmatically access and manage metrics data. This is useful for integrating metrics into custom applications or scripts.

Types of Bucket Metrics

1. **Request Metrics:**
    - **Total Request Count:** Number of requests made to your bucket (e.g., GET, PUT, DELETE).
    - **Request Rate:** Frequency of requests over time, useful for understanding traffic patterns.
    - **Request Latency:** Time taken to process requests, important for evaluating performance.

2. **Data Metrics**:
    - **Total Data Size**: Amount of data stored in the bucket.
    - **Data Transfer**: Amount of data transferred in and out of the bucket, relevant for understanding network usage and costs.
3. **Error Metrics**:
    - **4xx Errors**: Client-side errors indicating issues with request formatting or permissions.
    - **5xx Errors**: Server-side errors indicating problems with S3 services or processing.
4. **Storage Metrics**:
    - **Storage Utilization**: Amount of storage used by your objects, important for monitoring growth and managing storage capacity.
    - **Storage Class Distribution**: Breakdown of objects by storage class (e.g., Standard, Infrequent Access), useful for cost management.
5. **Event Metrics**:
    - **Event Notifications**: Metrics related to events such as object creation, deletion, and modification, providing insights into how objects are being managed.

Option 02: Storage Class Analysis

Storage Class Analysis is a tool that analyses the access patterns of your S3 objects over time. It helps you determine which objects are frequently accessed and which are not, enabling you to make decisions about transitioning objects to more cost-effective storage classes.

How It Works:

- **Data Collection**: Storage Class Analysis collects data on how often your objects are accessed and the type of operations performed (e.g., GET, PUT).
- **Data Aggregation**: It aggregates this data over a specified period, helping you identify patterns and trends in object access.
- **Analysis**: Based on the collected data, it provides recommendations on transitioning objects to different storage classes, such as moving infrequently accessed objects to the S3 Infrequent Access (IA) or Glacier storage classes.

Option 03: Replication Metrics

Replication in AWS S3 refers to the automatic copying of objects from one S3 bucket (the source bucket) to another bucket (the destination bucket) in the same or different AWS region. This feature is used to ensure data redundancy, improve data availability, and meet compliance requirements.

Replication Metrics refer to the data and statistics related to the replication of objects from a source S3 bucket to a destination bucket, either within the same region (intra-region replication) or across regions (cross-region replication). These metrics track the success, progress, and performance of replication operations.

Types of Replication Metrics:

- **Replication Status**: Metrics that indicate whether the replication process for an object was successful or encountered errors.
- **Replication Time**: Metrics that track the time taken for replication to complete, providing insights into the efficiency of the replication process.
- **Replication Lag**: Metrics showing the delay between the time an object is written to the source bucket and the time it appears in the destination bucket.
- **Replication Errors**: Metrics related to any errors or issues encountered during the replication process, such as failed replication attempts or permission issues.

How to Access Replication Metrics

1. **Amazon CloudWatch**:
    - **Metrics Dashboard**: Amazon S3 integrates with Amazon CloudWatch to provide detailed replication metrics. You can access these metrics through the CloudWatch console or API.
    - **Alarms and Notifications**: Set up CloudWatch alarms to receive notifications based on specific thresholds or conditions related to replication metrics, such as replication lag or error rates.
2. **S3 Management Console**:
    - **Replication Metrics**: Access replication metrics through the S3 Management Console under the "Metrics" tab for your bucket. This provides a visual representation of key metrics related to replication.

3. **AWS CLI and SDKs:**
    - **API Calls:** Use the AWS CLI or SDKs to programmatically access and manage replication metrics. This is useful for integrating replication monitoring into custom applications or scripts.

Next tab: Management

Option 01: Life cycle Rules

Lifecycle Rules are configurations that automate the management of objects in an S3 bucket by specifying actions to be performed on objects as they age or based on specific conditions. These actions include transitioning objects to different storage classes, archiving them, or deleting them.

Types of Lifecycle Actions:

- **Transition Actions:** Move objects from one storage class to another (e.g., from S3 Standard to S3 Infrequent Access or S3 Glacier) based on age or other criteria.
- **Expiration Actions:** Permanently delete objects after a specified period or based on a date. This helps manage the lifecycle of objects and prevent unnecessary storage costs.
- **Abort Incomplete Multipart Upload Actions:** Automatically abort multipart uploads that are incomplete after a specified number of days, freeing up space and avoiding costs associated with incomplete uploads.

Option 02: Replication Rules

Replication Rules are policies that define how and when objects in a source S3 bucket should be replicated to a destination bucket. These rules include criteria for selecting which objects to replicate, the destination bucket, and the replication behaviour.

Types of Replications:

- **Cross-Region Replication (CRR):** Replicates objects from a source bucket in one AWS region to a destination bucket in a different AWS region. This provides geographic redundancy and helps with disaster recovery.
- **Same-Region Replication (SRR):** Replicates objects within the same AWS region from a source bucket to a destination bucket. This is useful for data replication within a region for compliance, data processing, or data backup.

Option 03: Inventory Configurations

Inventory Configurations are settings that define how and when Amazon S3 generates reports on the objects in your buckets. These reports contain metadata about the objects and are delivered to a specified destination bucket in CSV or ORC (Optimized Row Columnar) format.

Components of Inventory Configurations:

- **Inventory Name**: A unique name for the inventory configuration.
- **Destination Bucket**: The S3 bucket where inventory reports will be delivered.
- **Report Format**: The format of the reports (CSV or ORC).
- **Frequency**: How often inventory reports are generated (daily or weekly).
- **Contents**: The specific metadata included in the reports (e.g., object size, storage class, last modified date).
- **Filter**: Optional filters to include only specific objects based on prefixes or tags.
- **Additional Fields**: Optional fields to include in the reports, such as replication status or encryption status.

Amazon
Simple Storage Service (S3)

Next tab: Access Points

Access Points

Access Points are network endpoints associated with S3 buckets that provide a customized way to access data. Each access point has its own DNS name and access policy, enabling you to manage access to your bucket in a more flexible manner.

Components of Access Points:

- **Access Point Name**: A unique name for the access point.
- **Access Point ARN**: The Amazon Resource Name (ARN) that identifies the access point.
- **Access Point Policy**: A policy attached to the access point that specifies permissions for accessing data.
- **Network Configuration**: Configuration options such as VPC (Virtual Private Cloud) settings to control network access.
- **Custom Alias**: An endpoint URL that can be used to access the data in the associated bucket.

Amazon
Simple Storage Service (S3)

AWS S3 Storage Classes

S3 Standard:

- **Use Case**: Frequently accessed data.
- **Durability**: 99.999999999% (11 9's) durability.
- **Availability**: 99.99% availability over a given year.
- **Cost**: Higher cost compared to other storage classes, but offers the best performance.

S3 Intelligent-Tiering:

- **Use Case**: Data with unknown or changing access patterns.
- **Durability**: 99.999999999% durability.
- **Availability**: 99.9% availability over a given year.
- **Cost**: Automatically moves data between two access tiers (frequent and infrequent) based on access patterns. There's a small monitoring and automation fee.

S3 Standard-IA (Infrequent Access):

- **Use Case**: Data that is less frequently accessed but needs to be quickly retrievable.
- **Durability**: 99.999999999% durability.
- **Availability**: 99.9% availability over a given year.
- **Cost**: Lower storage cost compared to S3 Standard but higher retrieval cost.

S3 One Zone-IA:

- **Use Case**: Data that is infrequently accessed and does not require multiple availability zone resilience.
- **Durability**: 99.999999999% durability within a single availability zone.
- **Availability**: 99.5% availability over a given year.
- **Cost**: Lower cost than S3 Standard-IA, but data is stored in a single availability zone.

S3 Glacier:

- **Use Case**: Data archiving and long-term backup.
- **Durability**: 99.999999999% durability.
- **Availability**: 99.99% availability over a given year.
- **Cost**: Very low storage cost, with retrieval times ranging from minutes to hours.

S3 Glacier Instant Retrieval & Flexible Retrieval

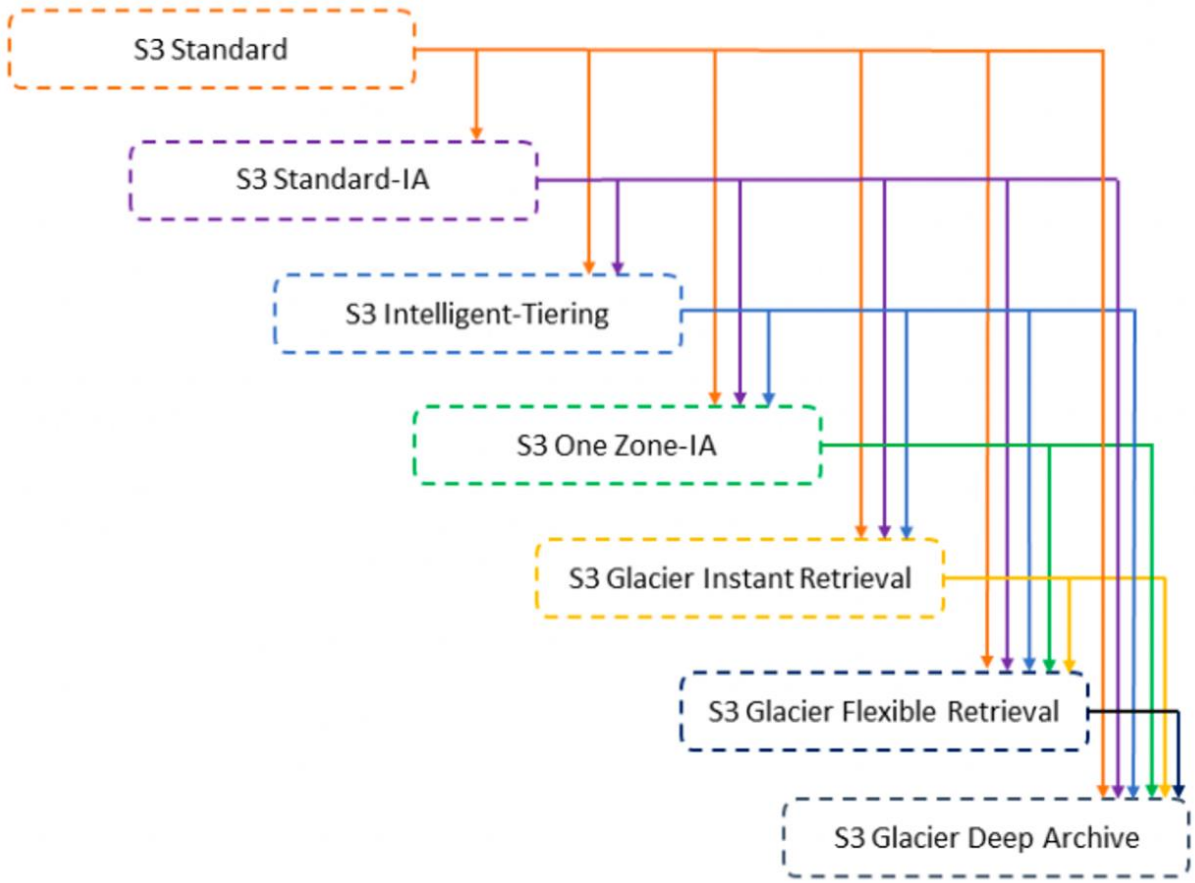
Instant - Ideal for data that needs to be accessed occasionally but must be retrieved instantly. It provides millisecond retrieval times.

Flexible - Designed for long-term archiving where retrieval time is less critical. It offers flexible retrieval options with a focus on cost efficiency.

S3 Glacier Deep Archive:

- **Use Case**: Long-term data archiving that is rarely accessed.
- **Durability**: 99.999999999% durability.
- **Availability**: 99.9% availability over a given year.
- **Cost**: Lowest storage cost, with retrieval times ranging from 12 to 48 hours.

Transition can only happen in the downward direction



Replication

- **Asynchronous replication**
- Objects are replicated with the **same version ID** and tags
- Supports **cross-region** and **cross-account** replication
- **Versioning must be enabled for source and destination buckets**
- For DELETE operations:
    - Replicate delete markers from source to target (optional)
    - Permanent deletes are not replicated
- There is no chaining of replication. So, if bucket 1 has replication into bucket 2, which has replication into bucket 3. Then objects created in bucket 1 are not replicated to bucket 3.
- Lifecycle actions are not replicated
- Can be configured at the S3 bucket level, prefix level, or object level using S3 object tags

Pre-signed URL

- Pre-signed URLs for S3 have temporary access token as query string parameters which allow anyone with the URL to temporarily access the resource before the URL expires (default 1h)
- Pre-signed URLs inherit the permission of the user who generated it
- Uses:
    - Allow only logged-in users to download a premium video
    - Allow users to upload files to a precise location in the bucket

Amazon
Simple Storage Service (S3)

S3 Numerical Questions

https://lisireddy.medium.com/aws-s3-numerical-questions-6562ed74639e

S3 Scenario based Questions

https://lisireddy.medium.com/aws-s3-scenario-based-questions-13eef3910d66

Wish you the best …! Happy Learning ..!

Yours' Love (@lisireddy across all the platforms)