S3:-

**What is Amazon S3?**

Amazon S3 (Simple Storage Service) is a highly scalable, secure, and durable object storage service offered by AWS. It allows you to store and retrieve any amount of data, at any time, from anywhere on the web. S3 is commonly used for storing backups, static files, media content, and large datasets.

**Why Use Amazon S3?**

1. **Scalability**: Automatically scales to handle large volumes of data and requests without user intervention.
2. **Durability**: Provides 99.999999999% (11 9s) durability by redundantly storing data across multiple devices and facilities.
3. **Accessibility**: Enables access to stored data via secure web protocols (HTTP, HTTPS).
4. **Cost-Effective**: Pay only for the storage and requests you use, with various storage tiers for cost optimization.
5. **Integration**: Works seamlessly with AWS services like CloudFront, EC2, RDS, Athena, and more.

**Benefits of Amazon S3**

1. **Data Availability**:
   * Provides high availability, ensuring your data is always accessible.
2. **Data Security**:
   * Supports encryption at rest and in transit.
   * Allows fine-grained access control through IAM, bucket policies, and ACLs.
3. **Global Reach**:
   * With multiple AWS Regions, you can store data closer to users for reduced latency.
4. **Versioning and Replication**:
   * Versioning safeguards against accidental deletions.
   * Replication ensures data redundancy across regions.
5. **Lifecycle Management**:
   * Automates transitioning of data to cost-effective storage classes or deletion.

**How to Create an S3 Bucket**

**Steps to Create an S3 Bucket:**

1. **Log in to AWS Console**:
   * Navigate to the AWS Management Console and select **S3** from the list of services.
2. **Create a New Bucket**:
   * Click on the **"Create bucket"** button.
3. **Configure Basic Settings**:
   * **Bucket Name**: Enter a globally unique bucket name (e.g., my-example-bucket).
   * **Region**: Choose the AWS Region where you want to store the data.
4. **Enable or Disable Versioning** (Optional):
   * Turn on versioning if you want to maintain multiple versions of objects.
5. **Set Permissions**:
   * Specify public or private access to the bucket.
   * Use **Block Public Access** settings to restrict access if required.
6. **Additional Settings** (Optional):
   * Enable encryption to secure your data at rest.
   * Add tags or set up logging for monitoring.
7. **Create Bucket**:
   * Click the **"Create bucket"** button to finish.

**How to Use the Bucket**

1. **Upload Files**:
   * Use the S3 console or SDKs (AWS CLI, Boto3) to upload files to your bucket.
2. **Access Files**:
   * Use the object's URL or integrate with other AWS services to access data.
3. **Set Permissions**:
   * Define who can access the data using bucket policies, IAM roles, or ACLs.
4. **Integrate with AWS Services**:
   * Use with CloudFront for content delivery or Athena for querying data.

A screenshot of a computer

AI-generated content may be incorrect.

Policy:-

A screenshot of a computer

AI-generated content may be incorrect.

Adding Static website:-(home.html,error.html)

A screenshot of a computer

AI-generated content may be incorrect.

Allow public access:-

A screenshot of a computer

AI-generated content may be incorrect.

Add policy to access s3(GetObject):-

A screenshot of a computer

AI-generated content may be incorrect.

**What is Versioning in S3?**

Versioning in Amazon S3 (Simple Storage Service) is a feature that allows you to maintain multiple versions of an object within a bucket. When versioning is enabled, S3 keeps all versions of an object, even if it's updated or deleted.

**What Does S3 Versioning Do?**

1. **Keeps a History of Changes**: Every time an object is modified or deleted, S3 saves the previous version instead of overwriting or permanently deleting it.
2. **Object Restoration**: Allows you to restore a previous version of an object if the current one is accidentally modified or deleted.
3. **Facilitates Object Deletion Tracking**: Even if an object is deleted, its previous versions remain accessible unless explicitly removed.

**Benefits of Versioning**

1. **Data Protection**:
   * Prevents accidental overwrites or deletions by maintaining older versions.
   * Enables recovery of lost or corrupted data.
2. **Backup and Recovery**:
   * Acts as a backup system for objects, ensuring that you can roll back to an earlier version if needed.
3. **Auditing and Compliance**:
   * Maintains a history of changes for regulatory compliance and auditing purposes.
4. **Support for MFA Delete**:
   * Enhances security by requiring multi-factor authentication to delete versions, preventing unauthorized deletions.

**How Versioning Works**

* When versioning is enabled in a bucket:
  + Each new object upload receives a **unique version ID**.
  + Existing objects are not overwritten; they are stored as older versions.
  + A delete operation only creates a **delete marker**, and the previous versions remain intact.

**Use Cases**

* **Data Recovery**: Recover accidentally deleted or modified data.
* **Application Backups**: Keep historical backups for application logs, configurations, or data files.
* **Audit Trails**: Maintain an immutable history for compliance or debugging.

**Enabling Versioning in S3**

1. Go to the **Amazon S3 Console**.
2. Select the bucket for which you want to enable versioning.
3. Navigate to the **Properties** tab.
4. Under **Bucket Versioning**, click **Edit** and enable it.

**Important Notes**

* **Cost**: Versioning may increase storage costs since all versions of an object are retained.
* **Lifecycle Policies**: Use lifecycle rules to manage versions (e.g., automatically delete older versions to control costs).

**What is Replication in S3?**

Replication in Amazon S3 is a feature that automatically copies objects from one bucket (source bucket) to another bucket (destination bucket). This process works between buckets in the same AWS account or across different AWS accounts and can span different regions.

**Types of S3 Replication**

1. **Cross-Region Replication (CRR)**: Copies objects to a bucket in a different AWS Region for disaster recovery or data distribution.
2. **Same-Region Replication (SRR)**: Copies objects to another bucket in the same AWS Region, often for compliance or backup purposes.

**Benefits of S3 Replication**

* **Disaster Recovery**: Ensures data is available in another region in case of a regional failure.
* **Performance**: Provides localized data access in different regions to reduce latency.
* **Compliance**: Meets regulatory requirements for data redundancy across locations.

**Steps to Create Cross-Region Replication in S3**

Here’s how to create a replication rule for copying objects between two buckets in different AWS Regions:

**1. Prerequisites**

1. **Enable Versioning**:
   * Both source and destination buckets must have versioning enabled.
2. **Destination Bucket Policy**:
   * Ensure the destination bucket has a policy allowing replication from the source bucket.
3. **IAM Role**:
   * Create or use an IAM role with permissions to replicate objects from the source bucket to the destination bucket.

**2. Set Up Cross-Region Replication**

1. **Navigate to Source Bucket**:
   * Open the Amazon S3 console.
   * Go to the source bucket where you want to set up replication.
2. **Enable Replication**:
   * In the source bucket's **Management** tab, select **Replication rules** and click **Create replication rule**.
3. **Define Rule Scope**:
   * Name the replication rule.
   * Choose whether to replicate the entire bucket or specific objects based on prefix or tags.
4. **Choose Destination**:
   * Select the destination bucket in another region.
   * You can choose a bucket in your account or another AWS account.
5. **Assign Permissions**:
   * Either allow S3 to create a new IAM role or select an existing role with replication permissions.
6. **Additional Options** (Optional):
   * **Replication Time Control (RTC)**: Ensure replication happens within a specified time frame (costs extra).
   * **Delete Marker Replication**: Choose whether to replicate delete markers.
7. **Review and Save**:
   * Review the configuration and click **Save rule**.

**Types of S3 Storage Classes**

Amazon S3 offers various storage classes designed for specific use cases, balancing cost, durability, and data retrieval needs. Here's a summary of each:

**1. S3 Standard**

* **Purpose**: Frequently accessed data.
* **Features**:
  + High durability (99.999999999% or "11 9s").
  + High availability and low latency.
  + Supports multiple availability zones.
* **Cost**: Highest among S3 classes due to its performance.
* **Use Cases**:
  + Hosting dynamic websites.
  + Storing frequently accessed application data.

**2. S3 Standard-IA (Infrequent Access)**

* **Purpose**: Infrequently accessed but still critical data.
* **Features**:
  + Lower storage cost than Standard.
  + Higher retrieval cost.
  + Supports multiple availability zones.
* **Use Cases**:
  + Backups.
  + Disaster recovery files.

**3. S3 One-Zone-IA**

* **Purpose**: Infrequently accessed data that doesn’t require multiple availability zone replication.
* **Features**:
  + Lower cost than Standard-IA.
  + Data is stored in a single availability zone.
  + Reduced durability compared to multi-zone options.
* **Use Cases**:
  + Secondary backups.
  + Data easily replaceable if lost.

**4. S3 Glacier Instant Retrieval**

* **Purpose**: Archive storage with instant access.
* **Features**:
  + Cost-effective for archival storage.
  + Millisecond retrieval.
  + Slightly higher cost than other Glacier classes.
* **Use Cases**:
  + Media archives.
  + Long-term storage with occasional access.

**5. S3 Glacier Flexible Retrieval (formerly Glacier)**

* **Purpose**: Long-term archival with flexible retrieval options.
* **Features**:
  + Lowest storage cost among retrievable archives.
  + Retrieval options: expedited, standard, or bulk (ranging from minutes to hours).
* **Use Cases**:
  + Historical data storage.
  + Regulatory archives.

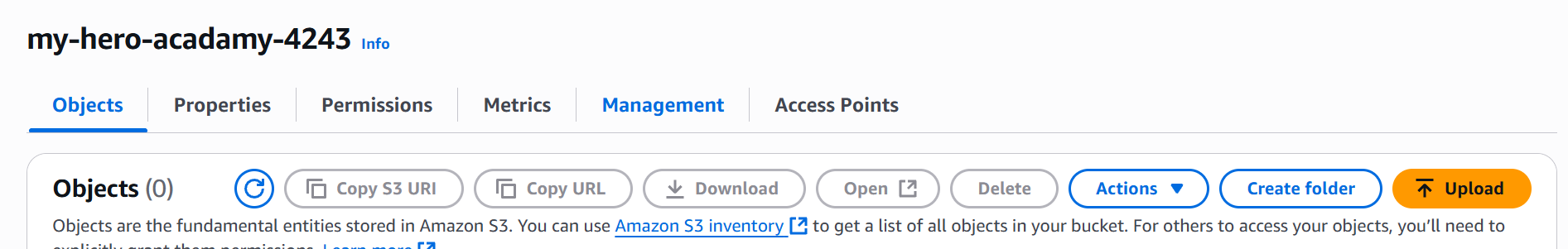
**6. S3 Glacier Deep Archive**

* **Purpose**: Lowest-cost storage for data accessed once or twice a year.
* **Features**:
  + Lowest storage cost in S3.
  + Retrieval time ranges from 12–48 hours.
* **Use Cases**:
  + Compliance archives.
  + Rarely accessed data with minimal urgency.

**7. S3 Intelligent-Tiering**

* **Purpose**: Automatically optimizes storage costs by moving data between access tiers based on usage.
* **Features**:
  + No retrieval fees.
  + Monitors access patterns.
  + Automatically transitions data to cost-effective tiers (e.g., Standard, IA, Glacier).
* **Use Cases**:
  + Data with unpredictable access patterns.
  + Cost-conscious applications.

CRETATING LIFECYCLE RULE:-



A screenshot of a computer

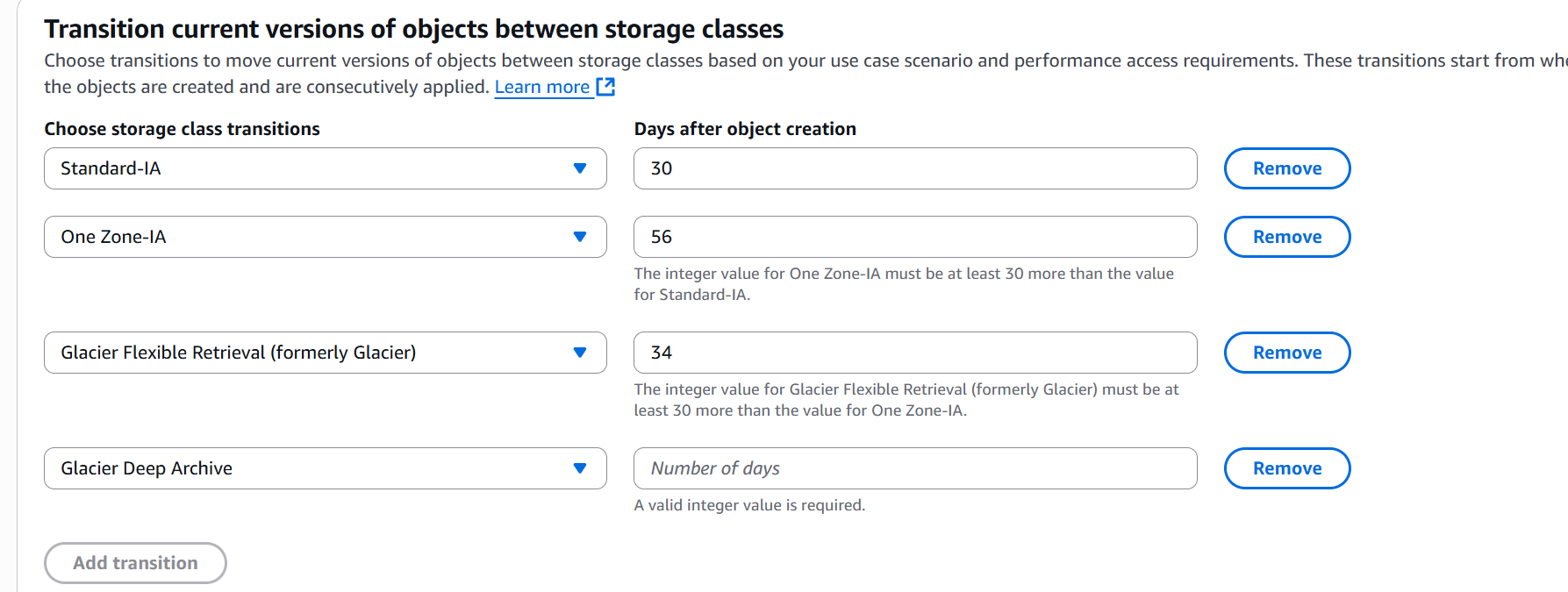
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



Creating a **lifecycle policy** in Amazon S3 involves setting rules to manage the lifecycle of objects in a bucket, such as transitioning them to different storage classes or expiring them after a set period. Follow these steps to create an S3 lifecycle policy:

**1. Sign in to AWS Management Console**

* Navigate to the **Amazon S3 service** in the AWS Management Console.

**2. Select the Bucket**

* In the **Buckets** list, click the name of the bucket where you want to configure the lifecycle policy.

**3. Go to Management Tab**

* In the bucket details page, click on the **Management** tab.
* Scroll down to the **Lifecycle rules** section.

**4. Create a Lifecycle Rule**

* Click on **Create lifecycle rule**.

**5. Configure the Rule**

* **Rule Name:**
  + Enter a descriptive name for your lifecycle rule (e.g., "ArchiveOldLogs").
* **Rule Scope:**
  + Apply to the whole bucket or specify a **prefix** or **tags** to apply the rule to specific objects.

**6. Add Lifecycle Actions**

You can choose from the following lifecycle actions:

**a. Transition to a Different Storage Class**

* Choose when objects should move to another storage class (e.g., **S3 Standard-IA**, **Glacier**, or **Deep Archive**).
* Specify the number of days after object creation when the transition should occur.

**b. Expire Objects**

* Specify the number of days after object creation for the objects to be permanently deleted.

**c. Clean Up Incomplete Multipart Uploads**

* Set a time period after which incomplete multipart uploads are automatically deleted.

**7. Define Rule Priority (Optional)**

* If you have multiple lifecycle rules, you can assign a **priority** to determine the order in which they are applied.

**8. Review and Save the Rule**

* Review your lifecycle configuration.
* Click **Create rule** to save it.