S3- ASSIGNMENT 1: SUBMITTED BY KALAISELVAN

1. What is S3?

- Amazon S3 (Simple Storage Service) is a cloud-based storage service provided by Amazon Web Services (AWS). It allows users to store and retrieve large amounts of data securely and reliably.
- S3 uses buckets to organize data and offers features like **encryption, access control, versioning, and lifecycle management.**
    - **Encryption**: Converting data into a secure and unreadable form to protect it from unauthorized access.
    - **Access Control:** Mechanisms and policies that regulate authorized access to resources and actions within a system.
    - **Versioning:** Maintaining multiple versions of data to track changes and provide a history of modifications.
    - **Lifecycle Management:** Managing data throughout its lifecycle, automating actions based on predefined rules and criteria.
- It is highly durable and has a **99.999999999**% durability guarantee.
- S3 is commonly used for backup, data archiving, content distribution, data lakes, and hosting static websites.
    - **Backup**: Creating copies of data to protect against loss or corruption, focusing on data integrity and recovery.
    - **Data Archiving:** Long-term retention of data with historical or legal value, moving infrequently accessed data to reduce storage costs.
    - **Content Distribution:** Delivering digital content efficiently to end-users using content delivery networks (CDNs) for better performance.
    - **Data Lakes:** Centralized repositories for storing raw and unstructured data, facilitating advanced analytics and data exploration.
    - **Hosting Static Websites:** Serving fixed content directly to web browsers, suitable for displaying information that doesn't change frequently.
- Overall, it is a popular and widely used cloud storage solution.

2. What are the storage services available in AWS?

- **Amazon S3:** Scalable object storage service for storing and retrieving any amount of data, used for backup, archiving, content distribution, data lakes, and hosting static websites.

- **Amazon EBS:** Block-level storage volumes attached to EC2 instances, providing durable and persistent storage for applications requiring low-latency and high-performance storage.
- **Amazon EFS:** Scalable file storage service with shared access via NFS, suitable for applications needing shared file access and scalability across multiple EC2 instances.
- **Amazon Glacier:** Low-cost archival storage service for long-term data retention of infrequently accessed data, ideal for compliance and regulatory purposes.
- **AWS Storage Gateway:** Hybrid storage service enabling integration between on-premises applications and AWS cloud storage, extending on-premises storage infrastructure to the cloud.
- **Amazon FSx**: Fully managed file systems for Windows and Lustre workloads, offering high-performance and scalable file storage optimized for specific use cases.
- **AWS Snow Family**: Physical devices (Snowcone, Snowball, Snowmobile) for secure and efficient data transfer in and out of AWS, particularly useful for large-scale data migration.

3. Differentiate between S3, EBS & EFS?

	Amazon S3	**Amazon EBS**	**Amazon EFS**
Storage Type	S3 uses an object storage model where data is stored as objects in buckets.	EBS provides block-level storage volumes, similar to virtual hard drives	EFS offers a shared file system accessible by multiple EC2 instances.
Access Scope	S3 is accessed over HTTP/HTTPS, making it accessible from anywhere on the web. **Through**: Web system interface	EBS volumes can be attached to a single EC2 instance and accessed at the block level. **Through**: file system interface	EFS uses the Network File System (NFS) protocol for file-level access **Through**: Web and file system interface
Use Cases	S3 is commonly used for backup and restore, data archiving, content distribution, data lakes, and hosting static websites.	EBS is suitable for applications that require low-latency, high-performance storage, such as databases or transactional workloads.	EFS is designed for applications that require shared access to files, such as content management systems, web serving, and data analytics.

Features	Encryption, access control, versioning, and lifecycle management	EBS supports features like snapshots for backup and restore, volume resizing, and transferring volumes between EC2 instances.	EFS automatically scales storage capacity based on demand and can be accessed by EC2 instances in multiple Availability Zones.
speed	Slower than EBS and EFS	Faster than S3 and EFS	Faster than S3, slower than EBS

4. What is size of S3 Bucket?

In Amazon S3, the capacity of an S3 bucket refers to the total amount of data that can be stored in the bucket. S3 buckets have virtually unlimited capacity, allowing you to store large amounts of data. You can store any number of objects (files) in an S3 bucket, and each object can range in size from 0 bytes to 5 terabytes (TB).

5. Explain the Storage classes in S3?

S3 Standard is the default storage class. It provides high availability and durability, with low latency and high throughput. This is a good choice for data that is accessed frequently.

S3 Intelligent-Tiering automatically moves data between two storage tiers, S3 Standard and S3 Standard-IA, based on access patterns. This can help you save money on storage costs for data that is not accessed frequently.

S3 Standard-IA and S3 One Zone-IA are designed for less frequently accessed data. They offer lower storage costs than S3 Standard, but with slightly longer retrieval latency.

S3 Glacier Instant Retrieval is designed for data that needs to be accessed quickly, even though it is not accessed frequently. It offers the lowest retrieval latency of all the S3 storage classes, but it also has the highest storage costs.

S3 Glacier Flexible Retrieval (formerly S3 Glacier) is designed for rarely accessed data that does not need to be accessed quickly. It offers the lowest storage costs of all the S3 storage classes, but it also has the longest retrieval latency.

Amazon S3 Glacier Deep Archive is designed for long-term archive and digital preservation. It offers the lowest storage costs of all the S3 storage classes, but it also has the longest retrieval latency.

Storage Class	Availability	Durability	Latency	Throughput	Frequency of Access	Cost
S3 Standard	99.99%	100.00%	Low	High	Frequently	High
S3 Intelligent-Tiering	99.90%	100.00%	Low	High	Frequently to Infrequently	Medium
S3 Standard-IA	99.90%	100.00%	Slightly longer	High	Infrequently	Low
S3 One Zone-IA	99.50%	100.00%	Slightly longer	High	Infrequently	Low
S3 Glacier Instant Retrieval	99.99%	100.00%	Lowest	Low	Infrequently to Rarely	High
S3 Glacier Flexible Retrieval	99.99%	100.00%	Long	Low	Rarely	Very Low
Amazon S3 Glacier Deep Archive	99.99%	100.00%	Longest	Low	Rarely	Very Very Low

*** Throughput is a measure of how many units of information a system can process in a given amount of time.**

*** latency refers to the time it takes to access or retrieve data from the storage service.**

6. What are the life cycle rules of S3?

These rules help optimize storage costs, performance, and compliance by automatically moving objects between different storage classes or deleting them when they reach a certain age.

Here are the key components of lifecycle rules in S3:

Transition actions: Transition actions define when and how objects should be moved between different storage classes. There are two types of transitions:

a. Transition to Glacier or Glacier Deep Archive: Objects can be automatically transitioned from the current storage class to the S3 Glacier or S3 Glacier Deep Archive storage classes. This is useful for archiving infrequently accessed data while reducing storage costs.

b. Transition to other storage classes: Objects can be moved from the current storage class to other S3 storage classes, such as S3 Intelligent-Tiering, S3 Standard-IA, or S3 One Zone-IA. This enables optimizing storage costs based on access patterns.

Expiration actions: Expiration actions define when objects should be deleted from your bucket. You can specify a time period after which objects should be automatically deleted. This is helpful for managing data retention policies or ensuring the removal of temporary or outdated data.

Filtering rules: Filtering rules define the scope of objects to which the lifecycle rule applies. You can define filters based on object key names, object tags, or object age. Filtering allows you to apply lifecycle actions only to specific objects that meet the specified criteria.

7. How you will you encrypt the objects in S3?

To encrypt objects in Amazon S3, you can choose from different options:

Server-Side Encryption with Amazon S3 Managed Keys (SSE-S3): S3 automatically encrypts objects using AES-256 encryption. No key management required.

Server-Side Encryption with AWS Key Management Service (SSE-KMS): S3 encrypts objects using AWS KMS customer master keys. Offers key management control and features like key rotation.

Server-Side Encryption with Customer-Provided Keys (SSE-C): You provide your own encryption keys for S3 to encrypt objects. You are responsible for key management.

Client-Side Encryption: Objects are encrypted on the client side before uploading to S3. You manage encryption and key management processes.

Regardless of the encryption method, objects in S3 remain encrypted at rest, and data transfers are secured using SSL/TLS encryption. Consider security needs, key management preferences, and compliance requirements to choose the appropriate encryption method.

8. How will you host static website in S3? Explain the steps in detail?

Steps on how to host a static website in S3:

Step 1: Create an S3 bucket. The bucket name must be unique across all of AWS. You can use the same name as your domain name, if you have one.

Step 2: Enable static website hosting. In the S3 console, go to the Properties tab for your bucket and select the "Static website hosting" option. Enter the index document name (typically index.html) and the error document name (typically 404.html).

Step 3: Upload your website files to the bucket. You can drag and drop files into the bucket in the S3 console, or you can use the AWS CLI or SDK.

Step 4: Set Permissions for Public Access

- Select the uploaded files and click on "Actions" > "Make public" to grant public read access to the objects.

- Alternatively, you can set the permissions manually for each file or configure the bucket policy to allow public access.

Step 5: Open a web browser and enter the S3 bucket's website endpoint.

Step 6: The static website should now be accessible via the provided URL or custom domain.

9. What is object lock in S3?

Amazon S3 Object Lock is a feature that allows users and businesses to store files in a highly secure, tamper-proof way. It is used for situations in which businesses must be able to prove that data has not been modified or destroyed after it was written, and it relies on a model known as write once, read many (WORM).

To use S3 Object Lock, you need to:

- Create a new bucket with Object Lock enabled.
- (Optional) Configure a default retention period for objects placed in the bucket.
- Place the objects that you want to lock in the bucket.
- Apply a retention period, a legal hold, or both, to the objects that you want to protect.
- Object Lock can help prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. You can use Object Lock to help meet regulatory requirements that require WORM storage or to simply add another layer of protection against object changes and deletion.

10. Explain Bucket level properties of S3?

The bucket-level properties in Amazon S3:

Bucket Name: Must be globally unique.

Access Control: Policies for controlling bucket and object access.

Logging: Detailed records of requests made to the bucket.

Versioning: Preservation of multiple versions of objects.

Lifecycle Management: Automated object transition and management based on criteria.

Cross-Region Replication: Replicating objects across regions for redundancy and compliance.

Static Website Hosting: Hosting static websites with index and error documents.

Notification Configuration: Configuring event notifications for specific bucket events.

Requester Pays: Requiring requesters to pay for data transfer and requests.

Object Lock: Preventing deletion or modification of objects for a specified retention period.

These bucket-level properties allow you to control access, logging, versioning, replication, hosting, notifications, cost allocation, and data immutability for your S3 buckets.

11. Explain Object level properties of S3?

The object-level properties in Amazon S3:

Key: Unique identifier for an object within a bucket.
Metadata: Additional information about the object in key-value pairs.
Access Control: Permissions defining who can access the object.
Storage Class: Determines durability, availability, performance, and cost.
Encryption: Options for encrypting the object's contents.
Version ID: Unique identifier for each version of an object (if versioning is enabled).
Retention Period: Object Lock feature to enforce retention for compliance.
Expiration: Setting an expiration date or time for automatic object deletion.
Restore Options: Retrieval options for objects stored in Glacier or Glacier Deep Archive.
These properties provide control over object behavior, security, lifecycle management, and compliance in S3. They allow customization of settings for each object to ensure efficient and secure data management.

12. What is Versioning in S3 ? What is use of Versioning?

Versioning in Amazon S3 is a feature that allows you to keep multiple versions of an object in the same bucket. You can use this feature to preserve, retrieve, and restore every version of every object stored in your buckets. Versioning-enabled buckets can help you recover objects from accidental deletion or overwrite.
For example, if you enable versioning on a bucket and then upload a new version of an object that already exists in the bucket, Amazon S3 automatically stores all versions of that object. You can then retrieve any version of the object that you want.
You can also use versioning to protect your data from being deleted or overwritten by mistake. If you delete an object from a versioning-enabled bucket, Amazon S3 does not delete the object but instead adds a delete marker to indicate that the object was deleted. You can still retrieve the object by specifying its version ID.

13. What is max size of single object in S3?

The maximum size of a single object in Amazon S3 (Simple Storage Service) is 5 terabytes (TB). This means you can store individual files or objects in S3, and each object can be up to 5 TB in size.

14. What is Bucket policy and how it works?

A bucket policy in Amazon S3 is a JSON-based access control mechanism for managing access to S3 buckets.

It defines permissions and restrictions for various operations on the bucket and its objects.

Bucket policies are written in JSON format and consist of statements that specify permissions and conditions.

Principals represent entities (such as IAM users, roles, AWS accounts, or services) to which the policy applies.

Actions define the specific operations or actions that are allowed or denied.

Resources refer to the S3 buckets or objects to which the actions in the policy apply.

Conditions are optional statements that can be added to the policy to further control access based on specific conditions.

When a request is made, the bucket policy is evaluated to determine whether the requester is allowed or denied the requested action.

It's important to carefully craft and review bucket policies to balance access, security, and compliance.

Always refer to the official AWS documentation for the most up-to-date information on bucket policies and their functionality.

15. What is glacier?

- Amazon Glacier is a low-cost, long-term storage service offered by AWS.
- It is designed for data archiving, backups, and long-term storage needs.
- Glacier provides a durable, secure, and cost-effective storage solution.
- It works in conjunction with Amazon S3 using S3 lifecycle policies.
- With Glacier, you can store data for extended periods, ranging from months to decades.
- It is particularly suitable for archival and backup purposes.
- Glacier is a reliable and economical cloud storage option within the AWS infrastructure.

16. What is S3 Cross Region Replication?

S3 Cross-Region Replication (CRR) - Key Points

Replication Process:

Copies objects from a source bucket in one AWS region to a destination bucket in another region.

Region-to-Region Replication:

Enables replication across different geographic locations globally.

Automatic Replication:

Replicates new and updated objects automatically in real-time or at configurable intervals.

Object-Level Replication:

Replicates individual objects while preserving metadata, permissions, and properties.

Synchronization:

Maintains synchronization between the source and destination objects, ensuring changes are replicated.

Versioning and Deletion:

Supports replication of all versions or only the latest version of an object.

Configurable deletion rules control object deletion in the destination bucket.

Permissions and Encryption:

Replication includes the access control list (ACL) of source objects.

Supports encryption of objects in transit using Amazon S3 server-side encryption or AWS Key Management Service (KMS) keys.

Monitoring and Metrics:

Provides monitoring and logging capabilities to track replication status and view replication metrics.

Helps troubleshoot any replication issues.

Use Cases:

Achieving regulatory compliance by storing data in specific regions.

Ensuring high availability by maintaining redundant copies in different regions.

Enhancing data durability and disaster recovery capabilities.

Cost Considerations:

Enabling Cross-Region Replication incurs additional costs for data transfer and storage in the destination region.

Refer to AWS documentation for up-to-date pricing details and considerations.

17. How will you access the s3 bucket from one aws account to another aws account?

There are two ways to access an S3 bucket from another AWS account:

Using IAM roles

Using VPC endpoints

Using IAM roles

- Create an IAM role in the account that owns the S3 bucket.
- Grant the role permissions to perform the S3 operations that you need.
- In the role's trust policy, grant a role or user in the other account permissions to assume the role.
- In the other account, create a user or role that can assume the role in the first account.
- Attach the role to the user or role in the other account.
- Use the temporary credentials that are generated when you assume the role to access the S3 bucket in the first account.

Using VPC endpoints

- Create a VPC endpoint for S3 in the account that owns the S3 bucket.
- Create a VPC in the other account that is connected to the VPC endpoint.
- In the other account, create a user or role that can access the VPC endpoint.
- Use the user or role to access the S3 bucket in the first account through the VPC endpoint.

Step-by-step procedure

Here are the step-by-step procedures for accessing an S3 bucket from another AWS account using IAM roles and VPC endpoints:

Using IAM roles:

Go to the IAM console in the account that owns the S3 bucket.

Click **Roles.**

Click Create role.

Select the AWS service option and select S3.

Click Next: Permissions.

Select the Attach permissions policies option and select the **AmazonS3FullAccess policy.**

Click Next: Tags.

(Optional) Add tags to the role.

Click Next: Review.

Enter a name for the role and click Create.

Go to the IAM console in the other account.

Click **Users.**

Click Create user.

Enter a username and select the Programmatic access option.

Click Next: Permissions.

Select the Attach existing policies directly option and select the AWSS3FullAccess policy.

Click Next: Tags.

(Optional) Add tags to the user.

Click Next: Review.

Enter a password for the user and click Create.

The user will be sent an email with their temporary credentials.

Use the temporary credentials to access the S3 bucket in the first account.

(https://www.youtube.com/watch?v=isY0MbSWwvM)

Using VPC endpoints:

Go to the VPC console in the account that owns the S3 bucket.

Click Endpoints.

Click Create endpoint.

Select the S3 service.

Select the VPC endpoint type as Gateway.

Select the VPC that you want to connect to the S3 bucket.

Click Create endpoint.

Go to the VPC console in the other account.

Click Endpoints.

Click Attach endpoint.

Select the S3 endpoint that you created in the previous step.

Click Attach endpoint.

The VPC endpoint will be attached to the VPC.

You can now access the S3 bucket in the first account through the VPC endpoint.

18. How AWS s3 intelligent tiering works?

AWS S3 Intelligent-Tiering is a storage class within Amazon S3 that automatically optimizes the cost of storing data by moving objects between two access tiers: frequent access and infrequent access. It uses machine learning algorithms to analyze object access patterns and determines the most appropriate storage tier for each object.

Here's an overview of how AWS S3 Intelligent-Tiering works:

1.Storage Tiers:

Frequent Access Tier: This tier is designed for objects that are accessed frequently.

Infrequent Access Tier: This tier is for objects that are accessed less frequently.

2.Initial Placement:

When you enable Intelligent-Tiering for a bucket, all newly uploaded objects are initially placed in the frequent access tier.

3.Access Monitoring:

AWS monitors access patterns to determine the frequency of object access.

4.Automatic Transitions:

Based on the access patterns, objects that have not been accessed for a specified duration are automatically transitioned from the frequent access tier to the infrequent access tier.

5.Cost Optimization:

By moving objects to the appropriate tier, Intelligent-Tiering optimizes storage costs. Objects that are frequently accessed remain in the frequent access tier, while less-accessed objects are moved to the infrequent access tier.

6.Performance and Durability:

Both tiers provide the same performance and durability guarantees as the standard S3 storage class.

7.Seamless Access:

Regardless of the storage tier, objects can be accessed transparently using the same S3 API and access controls.

8.Billing:

Intelligent-Tiering pricing is based on a combination of the storage capacity and the number of monthly object transitions between tiers.

9.Optimization Recommendations:

AWS provides recommendations to help you optimize costs further, such as identifying objects that could be archived using S3 Glacier for long-term storage.

AWS S3 Intelligent-Tiering simplifies the process of managing storage costs by automatically moving objects between access tiers based on their access patterns. It ensures that frequently accessed data remains readily available while minimizing costs for infrequently accessed data.

19. Explain the use case of pre-signed URL in S3?

Pre-signed URLs in Amazon S3 allow temporary and secure access to specific objects in an S3 bucket. Here's a summary of their use case and benefits:

- Securely share S3 objects without granting direct access or making them public.
- Enforce fine-grained access control by specifying permissions and expiration time.
- Grant temporary access for downloading, uploading, or performing actions on S3 objects.
- Integrate S3 with third-party applications by providing pre-signed URLs.
- Maintain object security through authentication and authorization.

- Serve private content to authenticated users for a limited duration.
- Simplify authentication by leveraging S3's pre-signed URLs.
- Perform time-limited actions like object copying or invoking S3 APIs.
- Provides flexibility, security, and controlled access to S3 objects.

20. What is CORS in S3 & How will you configure it?

CORS stands for Cross-Origin Resource Sharing. It is a mechanism that allows web applications to make requests to resources from other domains. By default, browsers restrict cross-origin requests, but CORS allows you to specify which domains are allowed to make requests to your S3 bucket.

To configure CORS for your S3 bucket, you need to add a CORS configuration to the bucket's permissions. The CORS configuration is a JSON document that specifies which origins are allowed to make requests to your bucket, the HTTP methods that are allowed, and other CORS-related settings.

Here is an example of a CORS configuration:

```
JSON
{
 "corsConfiguration": {
  "allowedOrigins": ["*"],
  "allowedMethods": ["GET", "PUT", "POST", "DELETE"],
  "allowedHeaders": ["*"],
  "maxAgeSeconds": 3000
 }
}
```

The ***allowedOrigins*** element specifies the list of origins that are allowed to make requests to your bucket. The ***allowedMethods*** element specifies the list of HTTP methods that are allowed. The ***allowedHeaders*** element specifies the list of headers that are allowed in the requests. The ***maxAgeSeconds*** element specifies the amount of time that the browser will cache the CORS configuration.

To add a CORS configuration to your S3 bucket, you can use the AWS Console, the AWS CLI, or the AWS SDKs.

Here are the steps on how to add a CORS configuration to your S3 bucket using the AWS Console:

I. Go to the Amazon S3 console.
II. Select the bucket that you want to configure CORS for.

III. Click the Permissions tab.

IV. Scroll down to the Cross-origin resource sharing (CORS) section.

V. Click Edit.

VI. In the CORS configuration editor text box, type or copy and paste the CORS configuration.

VII. Click Save changes.

21. Why S3 is a global Service? Why not Regional?

S3 is a global service because it is designed to provide high availability and low latency for users all over the world. By having S3 buckets distributed across multiple regions, AWS can ensure that users can access their data even if there is an outage in one region.

Regional services are designed to provide high availability and low latency for users in a specific region. However, regional services can be more vulnerable to outages than global services because they are not as distributed.

There are a few reasons why AWS chose to make S3 a global service:

Availability. By distributing S3 buckets across multiple regions, AWS can ensure that users can access their data even if there is an outage in one region.

Latency. By having S3 buckets in multiple regions, AWS can reduce the latency for users who are located far away from the region where their data is stored.

Cost. By sharing the infrastructure for S3 across multiple regions, AWS can reduce the cost of the service.

There are some downsides to having a global service like S3. For example, it can be more difficult to manage a global service than a regional service. Additionally, global services can be more expensive than regional services.

However, the benefits of having a global service like S3 outweigh the downsides. For most users, the availability, latency, and cost benefits of a global service like S3 make it the best option.

22. S3 is Global Service, While creating buckets its showing regions why so?

Even though Amazon S3 is a global service, you need to create your buckets in a specific region. This is because S3 buckets are physically located in data centers in different regions around the world. When you create a bucket, you are choosing the region where the bucket will be physically located.

There are a few reasons why you need to choose a region when you create an S3 bucket:

Availability: By choosing a region that is close to your users, you can improve the availability of your data. For example, if you have users in Europe, you might want to create your buckets in the Europe (Ireland) or Europe (Frankfurt) regions.

Latency: By choosing a region that is close to your users, you can improve the latency of your data. Latency is the time it takes for data to travel from your bucket to your users.

Cost: The cost of storing data in S3 varies by region. By choosing a region with lower storage costs, you can save money on your S3 bill.

Once you have created a bucket, you can access it from anywhere in the world. However, the data in the bucket will be physically located in the region where you created the bucket.

23. **What is ACL in S3, explain in detail**?

ACL (Access Control List) in Amazon S3 is a mechanism used to manage access permissions for S3 buckets and objects. It defines who can perform actions on resources and grants permissions to AWS accounts or predefined groups.

Key points about ACL in S3:

- ACL is a JSON-based policy that controls access to S3 buckets and objects.It can be set at the bucket-level or object-level.
- Bucket-level ACL applies to the entire bucket, while object-level ACL applies to individual objects.
- ACL uses permission grants to define access.
- Grantees can be AWS accounts (identified by their unique ID), email addresses, or predefined groups.
- Permission options include READ, WRITE, READ_ACP, WRITE_ACP, and FULL_CONTROL.
- READ allows reading or listing contents, WRITE allows writing or replacing objects, READ_ACP allows reading ACL, WRITE_ACP allows modifying ACL, and FULL_CONTROL grants all permissions.
- Default ACLs can be set for objects within a bucket, automatically applying to new objects.
- ACLs can be combined with IAM policies for more advanced access control.
- Best practices include following the principle of least privilege, regularly reviewing and auditing ACLs, and considering the use of IAM policies for finer control.

ACLs in Amazon S3 provide a secure and controlled environment for managing access to buckets and objects, ensuring appropriate permissions are granted to users and groups while adhering to access management best practices.

Note:

The permission options and their corresponding commands/apis:

READ:

Allows reading (downloading) the object or listing the contents of a bucket.

Command/API: s3:GetObject, s3:ListBucket

WRITE:

Allows writing (uploading or replacing) the object.

Command/API: s3:PutObject, s3:DeleteObject

READ_ACP:

Allows reading the object's ACL.

Command/API: s3:GetObjectAcl

WRITE_ACP:

Allows modifying the object's ACL.

Command/API: s3:PutObjectAcl

FULL_CONTROL:

Grants all permissions for the object or bucket.

Command/API: All S3 actions can be granted by using the wildcard s3:*.

24. **Write a CLI Command to create s3 bucket & Delete S3 bucket?**

The CLI commands to create and delete an S3 bucket:

Create S3 Bucket:

To create an S3 bucket, you can use the ***aws s3api create-bucket*** command. Here's an example command: aws s3api create-bucket --bucket <bucket-name> --region <region>

aws s3api create-bucket --bucket my-example-bucket --region us-east-1

Delete S3 Bucket:

To delete an S3 bucket, you can use the ***aws s3api delete-bucket*** command. Here's an example command: aws s3api delete-bucket --bucket <bucket-name>

aws s3api delete-bucket --bucket my-example-bucket

Note: Please note that to successfully delete a bucket, it must be empty. If the bucket contains any objects, you need to delete them first using the aws s3 rm command before deleting the bucket itself.

25. **Write a CLI Command to copy the files from local to S3 Bucket & S3 to local machine?**

To copy files from your local machine to an Amazon S3 bucket or from an S3 bucket to your local machine, you can use the AWS Command Line Interface (CLI). The AWS CLI provides a set of commands to interact with various AWS services, including S3. Here are the CLI commands to perform the copy operations:

Copy files from local machine to S3 bucket:

aws s3 cp <local-file-path> s3://<bucket-name>/<destination-prefix>/

aws s3 cp myfile.txt s3://my-bucket/my-folder/

Copy files from S3 bucket to local machine:

aws s3 cp s3://<bucket-name>/<object-key> <local-destination-path>

aws s3 cp s3://my-bucket/my-folder/myfile.txt ~/Downloads/

To copy a directory contains files:

To copy a directory (including all files and subdirectories) from your local machine to an Amazon S3 bucket using the AWS CLI, you can use the sync command. The sync command is specifically designed for synchronizing directories between a local location and an S3 bucket. Here's the CLI command to perform the sync operation:

From local to s3

aws s3 sync <local-directory-path> s3://<bucket-name>/<destination-prefix>/

aws s3 sync C:\Users\kalai\Desktop\s3upload s3://myfamily27-06-2023/