AWS-S3

Simple Storage Service

CONTENTS

Object Storage Vs. Block Storage

• Object storage is highly scalable and customizable, but not always fast. • Overall, object storage is typically used for large volumes of unstructured data • Example:- AWS S3	• Block storage is fast, but usually more expensive than object storage. • while block storage works best with transactional data and small files that need to be retrieved often. • AWS EBS

What is S3

- **Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance.**
- **Customers of all sizes and industries can use Amazon S3 to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.**
- **Amazon S3 provides management features so that you can optimize, organize, and configure access to your data to meet your specific business, organizational, and compliance requirements.**

Features of S3

- **Amazon S3 offers a range of storage classes designed for different use cases.**
- **For example, you can store mission-critical production data in S3 Standard for frequent access,**
- **save costs by storing infrequently accessed data in S3 Standard-IA or S3 One Zone-IA.**
- **and archive data at the lowest costs in S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, and S3 Glacier Deep Archive.**
- **You can store data with changing or unknown access patterns in S3 Intelligent-Tiering, which optimizes storage costs by automatically moving your data between four access tiers when your access patterns change.**

Amazon S3 Storage Classes (Definitely an Interview Question)

Documentation:- https://aws.amazon.com/s3/storage-classes/

	S3 Standard	S3 Intelligent-Tiering	S3 Standard-IA	S3 One Zone-IA	S3 Glacier	S3 Glacier Deep Archive
Designed for durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Designed for availability	99.99%	99.9%	99.9%	99.5%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99.9%	99.9%
Availability Zones	≥3	≥3	≥3	1	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128KB	128KB	40KB	40KB
Minimum storage duration charge	N/A	30 days	30 days	30 days	90 days	180 days
Retrieval fee	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved

Buckets and Objects

- **You can get started with Amazon S3 by working with buckets and objects. A *bucket* is a container for objects. An *object* is a file and any metadata that describes that file.**
- **To store an object in Amazon S3, you create a bucket and then upload the object to the bucket. When the object is in the bucket, you can open it, download it, and move it. When you no longer need an object or a bucket, you can clean up your resources.**

Bucket Access Control List (ACL)

Upload an Object - Console

After creating a bucket in Amazon S3, you're ready to upload an object to the bucket. An object can be any kind of file: a text file, a photo, a video, and so on.

To upload an object to a 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 Add files.**
5. **Choose a file to upload, and then choose Open.**
6. **Choose Upload.**

CLI S3api important commands (Depreciated but important)

- aws s3api list-buckets --query "Buckets[].Name"
- aws s3api create-bucket --bucket my-bucket --region us-east-1
- aws s3api delete-bucket --bucket my-bucket --region us-east-1
- aws s3api delete-object --bucket my-bucket --key test.txt
- aws s3api get-bucket-acl --bucket my-bucket

Newer S3 CLI Bucket specific commands

https://docs.aws.amazon.com/cli/latest/userguide/cli-services-s3-commands.html

https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3/index.html

- **aws s3 cp Copies a local file or S3 object to another location locally or in S3.**
- **aws s3 mb Creates an S3 bucket.**
- **aws s3 mv The mv command copies the file to the specified destination then deletes the source object or file.**
- **aws s3 ls List S3 objects and common prefixes under a prefix or all S3 buckets.**
- **aws s3 rb Deletes an empty S3 bucket.**
- **aws s3 rm Deletes an empty S3 bucket**
- **aws s3 sync Recursively copies new and updated files from the source directory to the destination.**

Examples of CLI commands

Example 1: Create a bucket

The following `mb` command creates a bucket. In this example, the user makes the bucket `mybucket`. The bucket is created in the region specified in the user's configuration file:

aws s3 mb s3://mybucket

Example 1: Listing all buckets owned by the owner

aws s3 ls

Example 1: Listing all objects in a bucket.

aws s3 ls s3://mybucket

Examples of CLI commands

Example 1: Copying a local file to S3

The following `cp` command copies a single file to a specified bucket and key:

aws s3 cp test.txt s3://mybucket/test2.txt

Example 1: Move a local file to the specified bucket

The following `mv` command moves a single file to a specified bucket and key.

aws s3 mv test.txt s3://mybucket/test2.txt

Example 1: Delete an S3 object

The following `rm` command deletes a single s3 object:

aws s3 rm s3://mybucket/test2.txt

Upload an Object - CLI

1. Create a Bucket
    a. aws s3 mb <target> [--options]
2. List Buckets and Objects
    a. aws s3 ls <target> [--options]
3. Copy Object to Bucket
    a. aws s3 cp s3://bucket-name/example s3://my-bucket/
    b. aws s3 cp filename.txt s3://bucket-name
    c. aws s3 cp s3://bucket-name/filename.txt ./
    d. echo "hello world" | aws s3 cp - s3://bucket-name/filename.txt
4. Delete a Bucket

S3 block public access

- **The Amazon S3 Block Public Access feature provides settings for access points, buckets, and accounts to help you manage public access to Amazon S3 resources. By default, new buckets, access points, and objects don't allow public access.**
- **However, users can modify bucket policies, access point policies, or object permissions to allow public access. S3 Block Public Access settings override these policies and permissions so that you can limit public access to these resources.**
- **With S3 Block Public Access, account administrators and bucket owners can easily set up centralized controls to limit public access to their Amazon S3 resources that are enforced regardless of how the resources are created.**

Bucket Policies

- **A bucket policy is a resource-based policy that you can use to grant access permissions to your Amazon S3 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 that are owned by the bucket owner. These permissions do not apply to objects that are owned by other AWS accounts.**
- **Bucket policies use JSON-based IAM policy language. You can use bucket policies to add or deny permissions for the objects in a bucket. Bucket policies can allow or deny requests based on the elements in the policy. These elements include the requester, S3 actions, resources, and aspects or conditions of the request (such as the IP address that's used to make the request).**

Adding a Bucket Policy through Console-

https://docs.aws.amazon.com/AmazonS3/latest/userguide/add-bucket-policy.html

Example of a Bucket Policy

The following example policy requires every object that is written to the bucket to be encrypted with server-side encryption using AWS Key Management Service (AWS KMS) keys (SSE-KMS). If the object isn't encrypted with SSE-KMS, the request will be denied.

```
{
"Version": "2012-10-17",
"Id": "PutObjPolicy",
"Statement": [{
  "Sid": "DenyObjectsThatAreNotSSEKMS",
  "Principal": "*",
  "Effect": "Deny",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
  "Condition": {
    "Null": {
      "s3:x-amz-server-side-encryption-aws-kms-key-id": "true"
    }
  }
}]
}
```

Amazon S3 Access Points

- **Amazon S3 access points simplify data access for any AWS service or customer application that stores data in S3. Access points are named network endpoints that are attached to buckets that you can use to perform S3 object operations, such as GetObject and PutObject.**
- **Each access point has distinct permissions and network controls that S3 applies for any request that is made through that access point.**
- **Each access point enforces a customized access point policy that works in conjunction with the bucket policy that is attached to the underlying bucket. You can configure any access point to accept requests only from a virtual private cloud (VPC) to restrict Amazon S3 data access to a private network. You can also configure custom block public access settings for each access point.**

Creating an Access Point for a Bucket

The following example command creates an access point named *example-ap* for the bucket *DOC-EXAMPLE-BUCKET* in the account *111122223333*. To create the access point, you send a request to Amazon S3 that specifies the following:

- **The access point name. For information about naming rules, see Rules for naming Amazon S3 access points.**
- **The name of the bucket that you want to associate the access point with.**
- **The account ID for the AWS account that owns the bucket.**

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --bucket DOC-EXAMPLE-BUCKET
```

When you're creating an access point by using a bucket in a different AWS account, include the --bucket-account-idparameter. The following example command creates an access point in the AWS account *111122223333*, using the bucket*DOC-EXAMPLE-BUCKET2*, which is in the AWS account *444455556666*.

aws s3control create-access-point --name example-ap --account-id 111122223333 --bucket DOC-EXAMPLE-BUCKET --bucket-account-id 444455556666

Using an Access Point

Access point ARNs use the format arn:aws:s3:*region*:*account-id*:accesspoint/*resource*.

For example:

- **arn:aws:s3:us-west-2:123456789012:accesspoint/test represents the access point named test, owned by account123456789012 in Region us-west-2.**
- **arn:aws:s3:us-west-2:123456789012:accesspoint/* represents all access points under account 123456789012 in Region us-west-2.**

Encrypting a Bucket

- **You can set the default encryption behavior on an Amazon S3 bucket so that all objects are encrypted when they are stored in the bucket. The objects are encrypted using server-side encryption with either Amazon S3-managed keys (SSE-S3) or AWS Key Management Service (AWS KMS) keys.**
- **Default encryption works with all existing and new Amazon S3 buckets. Without default encryption, to encrypt all objects stored in a bucket, you must include encryption information with every object storage request. You must also set up an Amazon S3 bucket policy to reject storage requests that don't include encryption information.**

Encrypt a Bucket Using CLI

https://docs.aws.amazon.com/AmazonS3/latest/userguide/default-bucket-encryption.html

Security Best Practices-S3

Security best practices

- Use AWS Identity and Access Management (IAM) to control access to your AWS resources, including your instances. You can create IAM users and groups under your AWS account, assign security credentials to each, and control the access that each has to resources and services in AWS. For more information, see Identity and access management for Amazon EC2.
- Restrict access by only allowing trusted hosts or networks to access ports on your instance. For example, you can restrict SSH access by restricting incoming traffic on port 22. For more information, see Amazon EC2 security groups for Linux instances.
- Review the rules in your security groups regularly, and ensure that you apply the principle of *least privilege*—only open up permissions that you require. You can also create different security groups to deal with instances that have different security requirements. Consider creating a bastion security group that allows external logins, and keep the remainder of your instances in a group that does not allow external logins.
- Disable password-based logins for instances launched from your AMI. Passwords can be found or cracked, and are a security risk. For more information, see Disable password-based remote logins for root. For more information about sharing AMIs safely, see Shared AMIs.

Assignment

- **Write a bash script to create a s3 bucket. The bucket name should be created using a random function where you have no idea of the name of the bucket. Use some commands to retrieve the bucket name and cp a script file to it. Display the contents of the bucket and delete it.**
- **Write a bash script to modify the access control policy of an existing bucket.**
- **Write a bash script to create a bucket and host a static website in it using nginx. Is the website accessible through curl and through browser?**
- **Using AWS CLI create an Ec2 instance. Use some command to display the publicdns address of the ec2 instance. Using the publicdns ssh into the instance through the script. Write another remote script that should be copied to the ec2 instance. This script should contain steps to create a bucket and attach an access point to it.**