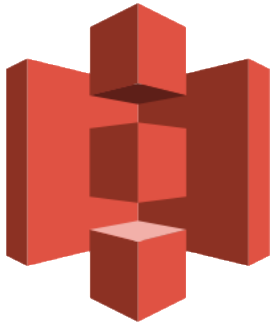

AWS – S3

What is S3?

- Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance.
- This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.
- Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements.
- Amazon S3 is designed for 99.999999999% (11 9's) of durability, and stores data for millions of applications for companies all around the world.

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
First byte latency	milliseconds	millseconds	milliseconds	milliseconds	select minutes or hours	select hours
Storage type	Object	Object	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes	Yes	Yes

S3 Consistency Models

- Amazon S3 provides read-after-write consistency for PUTS of new objects in your S3 bucket in all Regions with one caveat. The caveat is that if you make a HEAD or GET request to the key name (to find if the object exists) before creating the object, Amazon S3 provides eventual consistency for read-after-write.
- Amazon S3 offers eventual consistency for overwrite PUTS and DELETES in all Regions.
- Updates to a single key are atomic. For example, if you PUT to an existing key, a subsequent read might return the old data or the updated data, but it never returns corrupted or partial data.
- Amazon S3 achieves high availability by replicating data across multiple servers within AWS data centres.

Buckets & Objects

- To upload your data (photos, videos, documents etc.) to Amazon S3, you must first create an S3 bucket in one of the AWS Regions. You can then upload any number of objects to the bucket
- For example, you can create a bucket and upload objects using the Amazon S3 API. You can also use the Amazon S3 console to perform these operations. The console uses the Amazon S3 APIs to send requests to Amazon S3.
- Amazon S3 is a simple key, value store designed to store as many objects as you want. You store these objects in one or more buckets.

Objects

An object consists of the following:

- **Key** - The name that you assign to an object. You use the object key to retrieve the object.
- **Version ID** - Within a bucket, a key and version ID uniquely identify an object. The version ID is a string that Amazon S3 generates when you add an object to a bucket.
- **Value** - The content that you are storing. An object value can be any sequence of bytes. Objects can range in size from zero to 5 TB.
- **Metadata** - A set of name-value pairs with which you can store information regarding the object.
- **Access Control Information** - You can control access to the objects you store in Amazon S3. Amazon S3 supports both the resource-based access control, such as an access control list (ACL) and bucket policies, and user-based access control.

Bucket/Object Access Control

- An **S3 ACL** is a sub-resource that's attached to every **S3** bucket and object. It defines which AWS accounts or groups are granted access and the type of access. When you create a bucket or an object, Amazon **S3** creates a default **ACL** that grants the resource owner full control over the resource
- **S3 bucket policies**, on the other hand, are attached only to S3 buckets. S3 bucket policies specify what actions are allowed or denied for which principals on the bucket that the bucket policy is attached to (e.g. allow user Alice to PUT but not DELETE objects in the bucket).
- S3 access can be limits through **IAM policies** too by specifying what kind of actions would one like the user to do to what resource(bucket).

S3 IAM Policy - Example

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect": "Allow",
    "Action": "s3:*",
    "Resource": ["arn:aws:s3:::my_bucket",
                 "arn:aws:s3:::my_bucket/*"]
    }
  ]
}
```

The user, group or role to whom this policy is attached will have full access to that particular bucket and objects within that bucket.

S3 Bucket Policy - Example

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": ["arn:aws:iam::111122223333:user/Alice",
                "arn:aws:iam::111122223333:root"]
      },
      "Action": "s3:*",
      "Resource": ["arn:aws:s3:::my_bucket",
                   "arn:aws:s3:::my_bucket/*"]
    }
  ]
}
```

This bucket policy gives access to user Alice under account number 1111122223333 and to the person who owns the account to work with the bucket called my_bucket and all objects within it.

Demo

- Basics of S3 – Bucket, Object
- Create a bucket
- Upload an object
- Choosing appropriate region
- Viewing object properties
- Viewing bucket properties
- View/Modify bucket permissions
- View/Modify object permissions
- Understand that S3 is object based storage and not block based storage