Amazon S3

- Amazon S3 is one of the main building blocks of AWS
- It's advertised as "infinitely scaling" storage
- Many websites use Amazon S3 as a backbone
- Many AWS services use Amazon S3 as an integration as well

Use Cases:

- Backup and storage
- Disaster Recovery
- Archive
- Hybrid Cloud storage
- Application hosting
- Media hosting
- Data lakes & big data analytics
- Software delivery
- Static website

Buckets:

- Amazon S3 allows people to store objects (files) in "buckets" (directories)
- Buckets must have a globally unique name (across all regions all accounts)
- Buckets are defined at the region level
- S3 looks like a global service but buckets are created in a region

Objects

- Objects (files) have a Key
- Just keys with very long names that contain slashes ("/")

- Object values are the content of the body:
    - Max. Object Size is 5TB (5000GB)
    - If uploading more than 5GB, must use “multi-part upload”

- Metadata (list of text key / value pairs – system or user metadata)
- Tags (Unicode key / value pair – up to 10) – useful for security / lifecycle
- Version ID (if versioning is enabled

Security of S3

User-Based:

- IAM Policies – which API calls should be allowed for a specific user from IAM

Resource-Based:

- Bucket Policies – bucket wide rules from the S3 console - allows cross account
- Object Access Control List (ACL) – finer grain (can be disabled)
- Bucket Access Control List (ACL) – less common (can be disabled)

Note: an IAM principal can access an S3 object if

- The user IAM permissions ALLOW it OR the resource policy ALLOWS it
- AND there’s no explicit DENY

Encryption: encrypt objects in Amazon S3 using encryption keys

Bucket Policies:

JSON based policies:

- Resources: buckets and objects
- Effect: Allow / Deny
- Actions: Set of API to Allow or Deny
- Principal: The account or user to apply the policy to

Use S3 bucket for policy to:

- Grant public access to the bucket
- Force objects to be encrypted at upload
- Grant access to another account (Cross Account)

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "publicRead",
            "Effect": "Allow",
            "principal": "*",
            "Action": [
                "s3:GetObject"
            ],

            "resource": [
             "arn:aws:s3:::samplebucket/*"
            ]
        }
    ]
}
```

Static Website Hosting:

- S3 can host static websites and have them accessible on the Internet
- If you get a 403 Forbidden error, make sure the bucket policy allows public reads

Task:

- Install Visual Studio code
- Create a file “index.html” and “!” and press tab
- Add some content and images in a folder
- Upload all the files in the folder to S3
- Enable static website hosting in Bucket Properties

Versioning:

- You can version your files in Amazon S3
- It is enabled at the bucket level
- Same key overwrite will change the “version”: 1, 2, 3....
- It is best practice to version your buckets
- Protect against unintended deletes (ability to restore a version) • Easy roll back to previous version

Note:

- Any file that is not versioned prior to enabling versioning will have version “null”
- Suspending versioning does not delete the previous versions

Replication:

- Must enable Versioning in source and destination buckets
- Cross-Region Replication (CRR)
- Same-Region Replication (SRR)
- Buckets can be in different AWS accounts
- Copying is asynchronous
- Must give proper IAM permissions to S3

Use cases:

- CRR – compliance, lower latency access, replication across accounts
- SRR – log aggregation, live replication between production and test account