AWS S3

Section introduction

- 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
- We’ll have a step-by-step approach to S3

Amazon S3 Use cases

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

Real-world examples

- **Nasdaq** stores 7 years of data into **S3 Glacier** for long-term archival
- **Sysco** runs analytics on its data stored in **S3** to gain business insights

Amazon S3 - Buckets

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

Naming convention

- No uppercase letters, no underscores
- Bucket names must be 3–63 characters long
- Bucket name must not be formatted as an IP address
- Must start with a lowercase letter or number
- Must **not** start with the prefix `xn--`
- Must **not** end with the suffix `-s3alias`

Amazon S3 - Objects

- Objects (files) have a **Key**
- The **key** is the **full path**:
  - `s3://my-bucket/my_file.txt`

  - `s3://my-bucket/my_folder/another_folder/my_file.txt`

- The key is composed of **prefix** + **object name**

  - Example: `s3://my-bucket/my_folder/another_folder/my_file.txt`

- There’s no concept of ”directories” within buckets
  (although the UI might make you think otherwise)

- It's just keys with very long names that contain slashes (`/`)

Amazon S3 – Objects (cont.)

- Object values are the content of the body:

  - Max. object size is **5 TB (5000 GB)**
  - If uploading more than **5 GB**, must use **multi-part upload**

- **Metadata**: list of text key/value pairs (system or user metadata)

- **Tags**: Unicode key/value pairs (up to 10) — useful for **security** and **lifecycle policies**

- **Version ID**: present if **versioning** is enabled

Amazon S3 – Security

User-Based

- **IAM Policies** – define which API calls are allowed for a specific IAM user

Resource-Based

- **Bucket Policies** – bucket-wide rules from the S3 console (allows cross-account access)
- **Object Access Control List (ACL)** – fine-grained control (can be disabled)
- **Bucket Access Control List (ACL)** – less common (can be disabled)

Note

An **IAM principal** can access an S3 object **if**:

- The IAM permissions allow it **OR** the resource policy allows it
- **AND** there is no explicit **DENY**

Encryption

- Encrypt objects in Amazon S3 using encryption keys

S3 Bucket Policies

JSON-based policies

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

Use S3 bucket policy to:

- Grant public access to the bucket

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

Example: Public Access – Use Bucket Policy

An **anonymous website visitor** can access content in an **S3 bucket**
if the **S3 Bucket Policy allows public access**.

Key Point

- Public access is enabled via a bucket policy that explicitly allows access to everyone (`Principal: "*"`)
- No authentication is required for access

Use case

- Hosting a static website with public assets (e.g., HTML, images, CSS)

Example policy snippet

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::my-public-bucket/*"
    }
  ]
}
```

Example: User Access to S3 – IAM permissions

An **IAM User** can access an **S3 bucket**
if they have an **IAM Policy** that explicitly allows access.

Key Point

- The IAM Policy attached to the user defines which S3 actions are allowed or denied
- Access is **user-based**, controlled via IAM, not via the bucket policy

Example use case

- Granting internal developers or employees fine-grained access to specific S3 buckets

Example policy snippet

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

Example: EC2 instance access – Use IAM Roles

An **EC2 instance** can access an **S3 bucket**
if it has an **EC2 Instance Role** attached with the correct **IAM permissions**.

Key Point

- The EC2 Instance Role acts as an identity for the instance
- The permissions defined in the role's policy determine what the instance can access
- No need to manage access keys manually inside the EC2 instance

Use case

- Allowing applications running on EC2 to access S3 securely and with least privilege

Example policy snippet (attached to the role)

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

Advanced: Cross-Account Access – Use Bucket Policy

An **IAM User from another AWS account** can access an **S3 Bucket**
if the **Bucket Policy allows cross-account access**.

Key Point

- Cross-account access is granted by specifying the external account or user in the `Principal` field of the bucket policy
- This is a **resource-based policy** applied directly to the bucket

Example use case

- Sharing logs, data, or backups between accounts in a multi-account AWS setup

Example bucket policy snippet

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/ExternalUser"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::example-bucket/*"
    }
  ]
}
```

Bucket settings for Block Public Access

Options available:

- **Block all public access**: On
- **Block public access to buckets and objects granted through new access control lists (ACLs)**: On
- **Block public access to buckets and objects granted through any access control lists (ACLs)**: On
- **Block public access to buckets and objects granted through new public bucket or access point policies**: On
- **Block public and cross-account access to buckets and objects through any public bucket or access point policies**: On

Notes

- These settings were created to **prevent company data leaks**
- If you know your bucket should **never be public**, keep these **enabled**
- Settings can be applied at the **account level**

Amazon S3 – Static Website Hosting

- S3 can host static websites and make them accessible on the Internet

The website URL will be (depending on the region):

- `http://bucket-name.s3-website-aws-region.amazonaws.com`
  OR
- `http://bucket-name.s3-website.aws-region.amazonaws.com`

Important

- If you get a **403 Forbidden** error, make sure the **bucket policy allows public reads**

Example

For a bucket named `demo-bucket` in region `us-west-2` :

- `http://demo-bucket.s3-website-us-west-2.amazonaws.com`

Amazon S3 – Versioning

- You can version your files in Amazon S3
- Versioning is **enabled at the bucket level**
- Overwriting an object with the same key generates a new **version** (e.g., 1, 2, 3...)

Best Practices

- It's recommended to enable versioning:
  - Protects against unintended deletes (you can restore a previous version)
  - Enables easy rollback to a previous version

Notes

- Any file that is not versioned before versioning is enabled will have version `"null"`
- Suspending versioning **does not delete previous versions**

Amazon S3 – Replication (CRR & SRR)

- Must **enable Versioning** in both source and destination buckets
- **Cross-Region Replication (CRR)**
- **Same-Region Replication (SRR)**
- Buckets can belong to **different AWS accounts**
- Copying is **asynchronous**
- Must grant proper **IAM permissions** to S3 to perform replication

Use Cases

- **CRR (Cross-Region Replication)**:
  - Compliance
  - Lower latency access from other regions
  - Replication across AWS accounts
- **SRR (Same-Region Replication)**:
  - Log aggregation
  - Live replication between production and test accounts

Amazon S3 – Replication (Notes)

- After you enable replication, **only new objects are replicated**
- Optionally, you can replicate existing objects using **S3 Batch Replication**
  - Replicates existing objects and objects that failed replication

For DELETE operations

- Can **replicate delete markers** from source to target (optional setting)
- Deletions with a **version ID** are *not* replicated (to avoid malicious deletes)

No “chaining” of replication

- If bucket 1 replicates to bucket 2, and bucket 2 replicates to bucket 3,
  → Objects created in bucket 1 are **not** replicated to bucket 3

S3 Storage Classes

- **Amazon S3 Standard** – General purpose storage
- **Amazon S3 Standard-Infrequent Access (IA)** – For infrequently accessed data
- **Amazon S3 One Zone-Infrequent Access** – Lower-cost option for infrequent access in a single AZ
- **Amazon S3 Glacier Instant Retrieval** – Low-cost archive with milliseconds access
- **Amazon S3 Glacier Flexible Retrieval** – Archive with flexible access times (minutes to hours)
- **Amazon S3 Glacier Deep Archive** – Lowest-cost archive, access time in hours
- **Amazon S3 Intelligent Tiering** – Automatically moves data between access tiers based on usage

Note

- Objects can move between classes **manually** or using **S3 Lifecycle configurations**

S3 Durability and Availability

Durability

- High durability: **99.999999999% (11 nines)** across multiple Availability Zones (AZ)
- Example: if you store **10,000,000 objects**, you can expect to lose **one object every 10,000 years**
- Durability is the **same for all S3 storage classes**

Availability

- Measures how readily accessible the service is
- **Varies by storage class**
- Example: **S3 Standard** offers **99.99% availability**
  → This means it may be unavailable for up to **53 minutes per year**

S3 Standard – General Purpose

- **99.99% Availability**
- Used for **frequently accessed data**
- Provides **low latency** and **high throughput**
- Can sustain **2 concurrent facility failures**

Use Cases

- Big Data analytics
- Mobile & gaming applications
- Content distribution

S3 Storage Classes – Infrequent Access

- Designed for data accessed less frequently, but still requiring rapid access
- Lower cost compared to S3 Standard

Amazon S3 Standard-Infrequent Access (S3 Standard-IA)

- **Availability**: 99.9%
- **Use cases**: Disaster Recovery, backups

Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA)

- **Durability**: 99.999999999% (in a single AZ)
- **Availability**: 99.5%
- **Note**: Data is lost if the Availability Zone is destroyed

- **Use cases**: Secondary backup copies of on-premises data, or data that can be recreated

Amazon S3 Glacier Storage Classes

- Low-cost object storage meant for archiving / backup
- Pricing: includes cost for storage + object retrieval

Amazon S3 Glacier Instant Retrieval

- Millisecond retrieval, ideal for data accessed once a quarter
- Minimum storage duration: **90 days**

Amazon S3 Glacier Flexible Retrieval (formerly Amazon S3 Glacier)

- Retrieval options:
  - Expedited: **1 to 5 minutes**
  - Standard: **3 to 5 hours**
  - Bulk: **5 to 12 hours** (free)
- Minimum storage duration: **90 days**

Amazon S3 Glacier Deep Archive – for long-term storage

- Retrieval options:
  - Standard: **12 hours**
  - Bulk: **48 hours**
- Minimum storage duration: **180 days**

S3 Intelligent-Tiering

- Small monthly monitoring and auto-tiering fee
- Moves objects automatically between Access Tiers based on usage
- No retrieval charges in S3 Intelligent-Tiering

Access Tiers:

- **Frequent Access tier (automatic)**: default tier
- **Infrequent Access tier (automatic)**: objects not accessed for 30 days
- **Archive Instant Access tier (automatic)**: objects not accessed for 90 days
- **Archive Access tier (optional)**: configurable from 90 to 700+ days
- **Deep Archive Access tier (optional)**: configurable from 180 to 700+ days

S3 Storage Classes Comparison

Feature	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Glacier Deep Archive
Durability	99.999999999% (11 9's)						
Availability	99.99%	99.9%	99.9%	99.5%	99.9%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99%	99.9%	99.9%

Availability Zones	>= 3	>= 3	>= 3	1	>= 3	>= 3	>= 3
Min. Storage Duration	None	None	30 Days	30 Days	90 Days	90 Days	180 Days
Min. Billable Object Size	None	None	128 KB	128 KB	128 KB	40 KB	40 KB
Retrieval Fee	None	None	Per GB	Per GB	Per GB	Per GB	Per GB

📎 [AWS Storage Classes Documentation]()

S3 Storage Classes – Price Comparison (us-east-1)

Feature	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Gla De Arc
Storage Cost (per GB)	$0.023	$0.0025 – $0.023	$0.0125	$0.01	$0.004	$0.0036	$0.0
Retrieval Cost (per 1000 requests)	**GET**: $0.0004 **POST**: $0.005	**GET**: $0.0004 **POST**: $0.005	**GET**: $0.001 **POST**: $0.01	**GET**: $0.001 **POST**: $0.01	**GET**: $0.01 **POST**: $0.02	**GET**: $0.0004 **POST**: $0.03 **Expedited**: $10 **Standard**: $0.05 **Bulk**: free	**GET**: $0.0 **POS** $0.0 **Stan** $0.1 **Bulk** $0.0
Retrieval Time	Instantaneous	Instantaneous				Expedited (1–5 mins) Standard (3–5 hours) Bulk (5–12 hours)	Stan (12h) Bulk
Monitoring Cost	–	$0.0025	–	–	–	–	–

📎 [AWS S3 Pricing]()