AWS Storage Services Project

Contents

[Amazon Simple Storage Service(S3) 2](#_Toc194920642)

[Project 1: Creating a Simple Storage Service(S3) Bucket 2](#_Toc194920643)

[Let’s get started with creating AWS S3 bucket. 2](#_Toc194920644)

[Bucket type 4](#_Toc194920645)

[Learnings 14](#_Toc194920646)

[Elastic File System (EFS) 15](#_Toc194920647)

[Project 2: Creating EFS file system 15](#_Toc194920648)

[Let’s create first EFS File system 15](#_Toc194920649)

[Storage Strategies best practices 26](#_Toc194920650)

[Learning 27](#_Toc194920651)

# Amazon Simple Storage Service(S3)

## Project 1: Creating a Simple Storage Service(S3) Bucket

Simple Storage Service or **S3 is an *Object Storage* system**, this allows you to store files, videos, audios, logs and all other types of files in this storage system. It **stores files in structured and unstructured format**, object storage manages data as objects within buckets. **Each object includes data, metadata, and a unique identifier**, allowing for efficient data retrieval and management at any scale.

### Let’s get started with creating AWS S3 bucket.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1 Go to AWS Console and search S3 in search bar.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 2 Hit Create Bucket

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3 Give a name to your bucket. I have named it as ‘images999mihiraws’. It cannot have upper case letters and must be unique in global namespace.

### Bucket type

Amazon S3 supports two types of buckets. You can't change the bucket type after you have created the bucket.

* **General purpose buckets** – A general purpose bucket is the default Amazon S3 bucket type. General purpose buckets are recommended for the majority of use cases in Amazon S3. These buckets support most Amazon S3 storage classes and all Amazon S3 features.

General purpose buckets have a flat storage structure instead of a hierarchical structure like you might see in a file system. However, the Amazon S3 console supports the folder concept as a means of grouping objects by using a shared name prefix for objects in the same folder. A general purpose bucket's name must be globally unique and follow a specific set of bucket naming rules.

When you create a general purpose bucket, you specify the AWS Region where you want Amazon S3 to create your bucket. The objects inside of your buckets are stored across a minimum of three Availability Zones (AZs) in the specified AWS Region.

* **Directory bucket** – A directory bucket is an Amazon S3 bucket type that is used for workloads or performance-critical applications that require consistent single-digit millisecond latency. Directory buckets organize data hierarchically into directories, and can elastically scale performance to support hundreds of thousands of transactions per second (TPS). Directory buckets support only the S3 Express One Zone storage class and a limited set of Amazon S3 features.

You can create a directory bucket in a supported Availability Zone (AZ) that you choose. Unlike general purpose buckets, directory buckets exist only in a single Availability Zone in a single AWS Region. For optimal performance, we recommend that you co-locate your directory bucket and your computer resources in the same Availability Zone.

Directory bucket names must be unique within the chosen AZ, and the AZ ID is automatically included in the directory bucket name's suffix.

Ref: AWS Website itself when you click on Info.

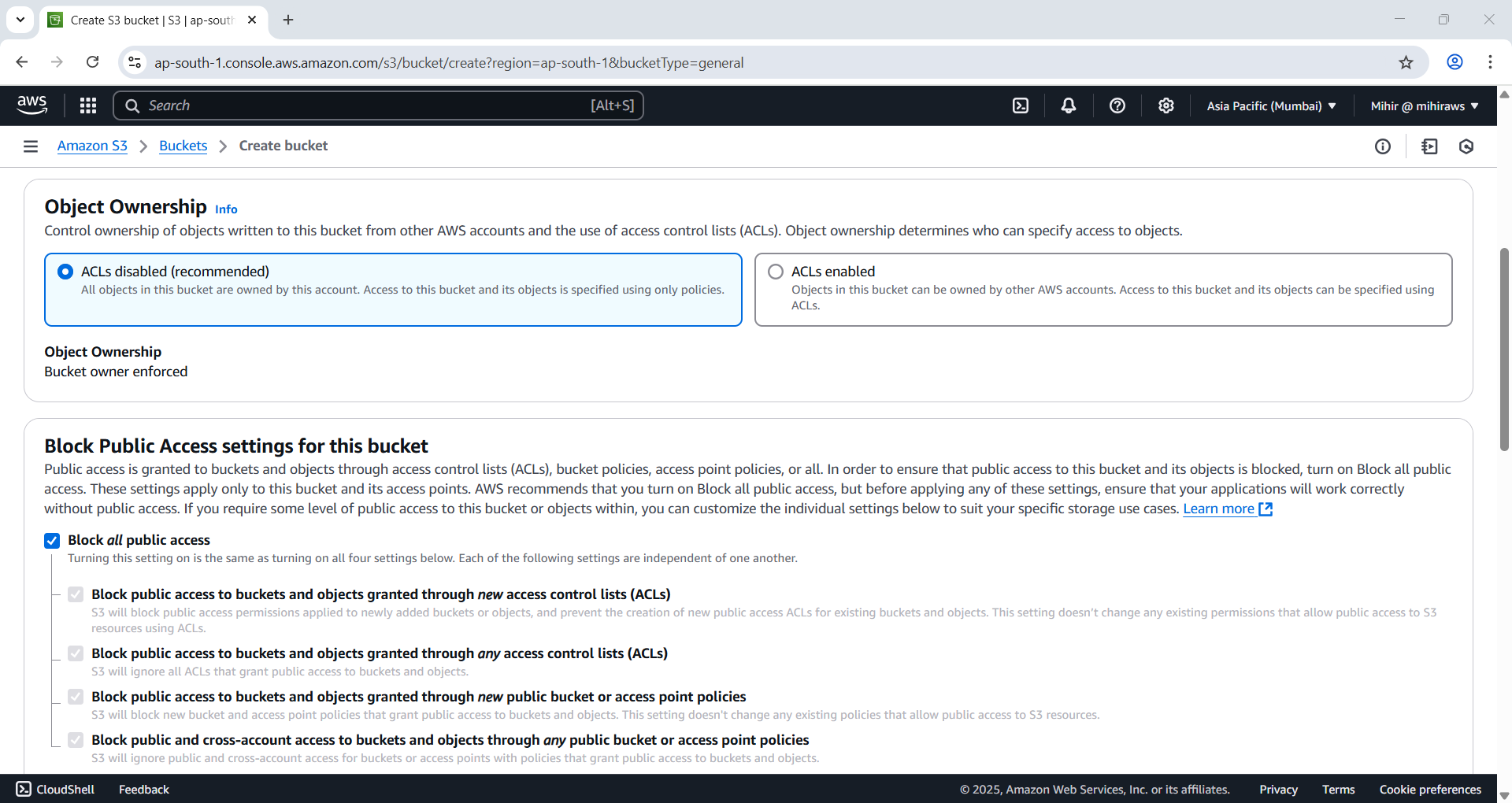


Figure 4 For now leave everything to default.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

Figure 5 I'm adding a tag. Best Practices is to add tags for reference purpose.

An orange rectangular object with black text

AI-generated content may be incorrect.

Figure 6 Hit 'Create Bucket'.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 7 Successfully created AWS S3 bucket.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 8 Hit 'Create Folder’ button.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 9 I have named my folder 'Images\_Porject 1'. Leave rest to default.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 10 Folder created successfully.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 11 Hit the upload button to ad files to this bucket. You need to first open the bucket otherwise file will be uploaded otuside the folder.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 12 Added the file to upload.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 13 File uploaded to the folder we created.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 14. Successfully uploaded files to the folder.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 15 If you click the file information about it is displayed.

**By default, all files uploaded to the bucket are private and owned by the creator, meaning only the owner can view, edit, or download them.** However, there may be instances where you want to configure the bucket for public access—such as when everyone needs to view images on a website.

To achieve this, we can configure the bucket to allow public access to the files. This should make the object URL accessible to the public. Unfortunately, in this case, we are receiving an "Access Denied" message when attempting to access the file.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 16 By default accessed is denied.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 17 Need to go to 'Permissions' tab at 'Bucket' level.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 18 Hit 'Edit'.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 19 Find service 'S3'. And search action 'GetObject'.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 20 Copy ARN and change principal to \*

A screenshot of a computer

AI-generated content may be incorrect.

Figure 21 By disabling "block public access..." last option it bypasses the block policy set by default by allowing access to S3 objecrs and buckets if policies are aloowing it.

A computer screen shot of a cube

AI-generated content may be incorrect.

Figure 22 Going back to the bucket and opening the object uri it gives access to the S3 objects/files.

Credit: Image taken from unsplash.com by [BoliviaInteligente](https://unsplash.com/photos/graphical-user-interface--nDhdxxOvgc).

### Learnings

I learned how to create a S3 bucket, make a folder and add files to it. Apart from uploading files to S3, also learned to add policy at bucket level, and making objects available at public domain. We can host static websites, add file of any type to S3 object storage. If this service is paired with CloudFront and Content Delivery Network service can efficiently deliver content to users with low latency.

A kind thank you to the author **[Zoumana Keita](https://www.datacamp.com/portfolio/keitazoumana" \t "_self)**.

Reference:

<https://www.datacamp.com/tutorial/aws-s3-efs-tutorial>

# Elastic File System (EFS)

## Project 2: Creating EFS file system

Elastic File System or EFS is a service offered by AWS **that allows shared filed system with other EC2 instances and supports simultaneous access from multiple instances**. EFS **auto scales** **when the load increases** and allows fast file access from multiple EC2 instances. EFS is highly recommended **with applications that require a shared file system** or if multiple users or system **need access and modification of content.** It is a file storage type.

A real-world example of using Amazon EFS is in a **video production company** that works with large video files. Imagine a company that produces films or online video content.

In this scenario, different departments (such as editing, animation, sound, and special effects) need to access and work on the same video files. Each department might be working from separate servers, but they all need to access the same footage and collaborate in real time.

By using Amazon EFS, all of the video editors, animators, and other team members can mount the same shared file system. This allows them to:

* **Collaborate seamlessly**: Multiple editors can work on different parts of the video simultaneously without having to duplicate files.
* **Ensure file consistency**: Since EFS is shared across all instances, any changes made to the video files are immediately available to everyone, ensuring all team members are working with the latest version.
* **Scalability**: As the video files grow larger with new footage and edits, EFS automatically scales to handle the increased storage needs without manual intervention.

In this case, EFS enables efficient collaboration, smooth workflows, and eliminates bottlenecks from having to transfer large files between different departments.

Let’s create first EFS File system

A screenshot of a computer

AI-generated content may be incorrect.

Figure 23 First We create new Security Group for our EC2 Instance.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 24 Make sure outbound rule has first rule with 'All Traffic' selected and hit ‘Create Security Group'.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 25 Security Group successfully created.

A computer screen shot of a computer

AI-generated content may be incorrect.

Figure 26 Open EFS page from search bar. Hit ‘Create file system’.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 27 I have named by EFS system as 'MyFirstEFSFileSystem', next hit 'Customize' button.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 28 Change security of all the availability zones to the one we we just created. Hit ‘Next’ button twice until you reach Review tab.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 29 Carefully review.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 30 Hit 'Create' button. and Go and create an EC2 Instance.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 31 Successfully created EFS file system.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 32 I go to my EC2 instance page and launch both the instances.

*I already created both the instance earlier for some other projects and we are going to start both here. If you don’t have any you can create your own.*

A screenshot of a computer

AI-generated content may be incorrect.

Figure 33 Instances successfully started.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 34 I SSHed into both of instances.

Finally, we are going to mount EFS File system we created on both instances.

We leverage this command**: sudo yum install -y amazon-efs-utils**

This command installs EFS mount helper, this needs to be executed on all instances.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 35 Successfully installed the mount helper

A screenshot of a computer

AI-generated content may be incorrect.

Figure 36 Created efs folder in root directory.

We will now mount it into ‘efs’ directory created with the help of following command.

Command: **sudo mount -t efs [systemID] /efs**

* The command starts with sudo, which allows the rest of the command to be executed with root privileges.
* mount is a command used to mount filesystems.
* -t efs specifies the type of the filesystem to be mounted, in this case, Amazon EFS (Elastic File System).
* fs-088faeea6372efa83:/ is the identifier of the EFS filesystem to be mounted.
* /efs is the directory where the EFS filesystem will be mounted.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 37 The reason it isn't allowing is because security group doesn't allow inbound traffic from to EFS file system.

In order to fix this, we need to edit our security group.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 38 We will edit the inbound rules.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 39 Select types as 'NFS' and Source as ‘Anywhere’.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 40 Successfully added new inbound rule.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 41 No error message means we did it correct.

Now we will create a folder under efs directory and place a file and access it from another instance.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 42 If you look closely file created in one instance is accessible in another.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 43 Created a file in instace 2 and accessing it from instance 1.

### Storage Strategies best practices

Efficient data management relies on implementing effective storage strategies, which focus on optimizing both performance and costs. These strategies are particularly important when working with Amazon’s Simple Storage Service (S3) and Elastic File System (EFS).

Performance optimization in S3 and EFS involves choosing the right storage classes and throughput options to improve speed and operational efficiency.

Cost optimization, on the other hand, is achieved by utilizing various storage tiers based on how often data is accessed and leveraging AWS monitoring tools for cost control. By employing these strategies, organizations can achieve both efficient data management and cost savings in cloud storage.

### Learning

By leveraging performance and cost optimization strategies, businesses can ensure that they manage their cloud storage resources efficiently while controlling expenses. EFS is a shared file system and with attaching security group we can gain or remove access to EFS file systems. It is an efficient solution where multiple systems or users can read/write to same files.

I learned 2 services S3 and EFS. Both services are necessary and must take a strategy to apply both as per the need of the organization. While S3 is good for storing data whether in structured or unstructured format, EFS can be utilized in places where files need to be accessed by multiple users or systems.

**Amazon S3 (Simple Storage Service)** and **Amazon EFS (Elastic File System)** are both scalable storage solutions, but they serve different purposes. Here are real-world use cases for both:

**Amazon S3:**

1. **Backup and Archiving**: S3 is ideal for storing large amounts of data such as backup files, old records, or archived data due to its durability and low-cost storage options.
2. **Web Hosting**: Static websites (e.g., HTML, CSS, JavaScript) are commonly stored in S3 for fast, scalable, and secure hosting.
3. **Big Data Analytics**: S3 is used to store vast amounts of raw data for analytics or machine learning projects, where the data can be easily accessed by tools like AWS Athena, Redshift, or SageMaker.

**Amazon EFS:**

1. **Shared File Storage**: EFS is used for applications that require a shared file system across multiple EC2 instances, such as content management systems, media workflows, or enterprise applications.
2. **Web Applications**: EFS provides a shared storage backend for dynamic web applications that need to store and retrieve files across multiple servers.
3. **DevOps and CI/CD**: EFS is used to share configuration files, logs, or application code among development and testing environments.

In short, S3 is great for object storage like backups and static assets, while EFS is used for shared file systems requiring concurrent access from multiple EC2 instances.

A kind thank you to the author **[Zoumana Keita](https://www.datacamp.com/portfolio/keitazoumana" \t "_self)**.

Reference:

<https://www.datacamp.com/tutorial/aws-s3-efs-tutorial>