



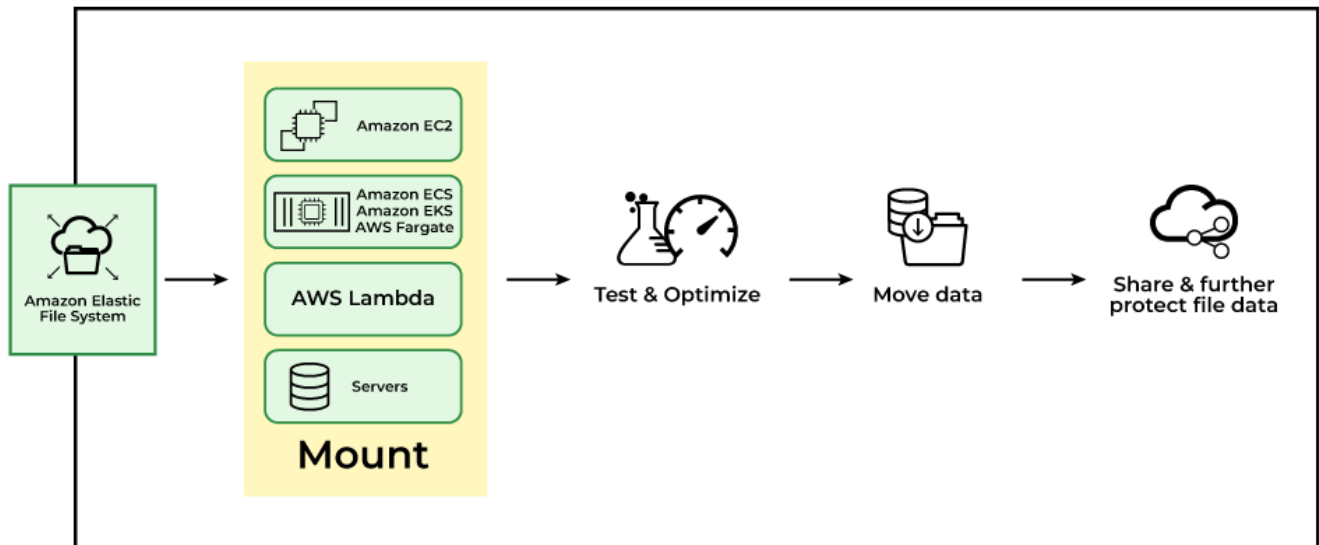
JIDNYASA

CORPORATE TRAINING
CENTRE

NOTES
AWS-EFS

AWS Elastic File System(EFS):

- A fully-managed file storage service that makes it easy to set up and scale file storage in the Amazon Cloud.
- Serverless, fully elastic file storage that lets you share file data without provisioning or managing storage capacity and performance.
- Amazon EFS file systems can automatically scale from gigabytes to petabytes of data without needing to provision storage.
- Tens, hundreds, or even thousands of compute instances can access single Amazon EFS file system at the same time, and Amazon EFS provides consistent performance to each compute instance.
- Amazon EFS is designed to be highly durable and highly available.
- With Amazon EFS, there is no minimum fee or setup costs, and you pay only for what you use.



Features of EFS:

- The service manages all the file storage infrastructure for you, avoiding the complexity of deploying, patching, and maintaining complex file system configurations.
- EFS supports the Network File System version 4 protocol (NFS4).
- You can mount EFS filesystems onto EC2 instances running Linux or MacOS Big Sur.
- **You cannot mount EFS with Windows, Windows are not supported by EFS.**
- Multiple Amazon EC2 instances can access an EFS file system at the same time, providing a **common data source** for workloads and applications running on more than one instance or server.
- EFS file systems store data and metadata across multiple Availability Zones in an AWS Region.
- EFS file systems can grow to petabyte scale, drive high levels of throughput, and allow massively parallel access from EC2 instances to your data.
- EFS enables you to control access to your file systems through Portable Operating System Interface (POSIX) permissions.
- You can schedule automatic incremental backups of your EFS file system using the EFS-to-EFS Backup solution.
- Amazon EFS Infrequent Access (EFS IA) is a new storage class for Amazon EFS that is cost-optimized for files that are accessed less frequently. Customers can use EFS IA by creating a new file system and enabling Lifecycle Management. With Lifecycle Management enabled, EFS automatically will move files that have not been accessed for 30 days from the Standard storage class to the Infrequent Access storage class. To further lower your costs in exchange for durability, you can use the EFS IA-One Zone storage class.

Use cases of EFS:

1. **Secured file sharing:** You can share your files in every secured manner and in a faster and easier way and also ensures consistency across the system.
2. **Web Hosting:** Well suited for web servers where multiple web servers can access the file system and can store the data EFS also scales whenever the data incoming is increased.
3. **Modernize application development:** You can share the data from the AWS resources like ECS, EKS, and any serverless web applications in an efficient manner and without more management required.
4. **Machine Learning and AI Workloads:** EFS is well suited for large data AI applications where multiple instances and containers will access the same data improving collaboration and reducing data duplication.

When should I use Amazon EFS

Before you are going to use any particular service for your application you need to analyse the cost and requirements of the data that you're going to save.

1. **Shared File Storage:** If the multiple EC2-Instances have to access the same data. EFS management of shared data and ensures consistency across instances.
2. **Scalability:** EFS can increase and decrease its storage capacity depending on the incoming data. If you don't have an idea how much data is going to come to the store then you can use the Amazon EFS.
3. **Simplified Data Sharing:** If different applications want the same data to use in a collaborative manner then you can choose the Amazon EFS. EFS can share large datasets across a group of instances.
4. **Use with Serverless Applications:** Amazon EFS is well suited for the service like serverless computing services like some of the examples AWS lambda, EFS, and so on.
5. **Pay-as-You-go-Model:** If your application is having unpredictable storage growth then there is no need of paying upfront or no need of any prior commitments. You pay only for the storage that you are going to use.

AWS offers cloud storage services to support a wide range of storage workloads.

- **EFS** is a file storage service for use with Amazon compute (EC2, containers, serverless) and on-premises servers. EFS provides a file system interface, file system access semantics (such as strong consistency and file locking), and concurrently accessible storage for up to thousands of EC2 instances.
- **Amazon EBS** is a block-level storage service for use with EC2. EBS can deliver performance for workloads that require the lowest-latency access to data from a single EC2 instance.
- **Amazon S3** is an object storage service. S3 makes data available through an internet API that can be accessed anywhere.

How do I access a file system from an EC2 instance?

To access your file system, mount the file system on an EC2 Linux-based instance using the standard Linux mount command and the file system's DNS name.

Once mounted, you can work with the files and directories in your file system like you would with a local file system. EFS uses the Network File System version 4 (NFS v4) protocol.

Amazon EFS supports one to thousands of Amazon Elastic Compute Cloud (EC2) instances connecting to a file system concurrently.

Storage classes in EFS:

Amazon EFS offers three storage classes:

- **EFS Standard:** This is the default storage class for EFS. The user is only charged for the amount of storage used. This is recommended for storing frequently accessed files. Data that is frequently accessed tends to have higher performance needs, so EFS provides an SSD-powered EFS Standard class designed to deliver sub-millisecond latencies.
- **EFS Infrequent Access (IA):** Cheaper storage space. Recommended for rarely accessed files. Increased latency when reading or writing files. The user is charged not only for the storage of files but also

charged for read and write operations. EFS Infrequent Access (IA), designed for data accessed only a few times a quarter. EFS IA offers an up to 95% lower cost than EFS Standard for infrequently accessed data.

- **EFS Archive:** EFS Archive, designed for data accessed a few times a year or less. Providing a more cost-optimized experience for even colder data, EFS Archive offers an up to 50% lower cost than EFS Infrequent Access, with a higher request charge when that data is accessed. EFS Archive is optimized for and supported on EFS Regional file systems using EFS's default Elastic Throughput mode.

Difference between S3 & EBS & EFS

How is Amazon EFS Different than Amazon S3?

- Amazon EFS (Elastic file system) and S3 (Simple Storage Service) are two different storage services provided by Amazon web services with two different purposes one storage is to store static data and another is to store dynamic data.

Elastic File System	Simple Storage Service
EFS can be accessed by multiple EC2-instance at the same time which can analyse the data and can use the data combined.	S3 is an object storage that is mainly used to store and retrieve static data the data is stored in the form of objects.
If any changes are made by one instance to the data it is visible to the other instances immediately.	When you perform the read and write operations on the data you will always get the most updated version of the data only.
Widely used for the scenarios like data sharing with multiple instances.	Commonly used for backup, restoring, and hosting static content.

How is Amazon EFS Different Than Amazon EBS?

- Amazon EFS (Elastic file system) and Amazon EBS (Elastic Block Store) are two different services provided by Amazon web services for different use cases.

Elastic File System	Elastic Block Store
If the application is required to share the shared file access then you can choose the Elastic file system.	If the application requires separate storage then you can use the block store.
Multiple instances can access the storage system at once.	A single instance only can access the storage system at once.
Deliveries the aggregate throughputs to thousands of clients simultaneously.	Highly available with low latency.

Different Performance Modes in EFS

General-purpose:

- Offers low latency.
- Supports a maximum of 7000 IOPS.
- As a cloud watch metric, you can view the amount of IOPS your architecture uses and can switch to Max IOPS if required.

Max I/O:

- This is recommended when EFS needs over 7000 IOPS
- Theoretically, this mode has an unlimited I/O speed.

Performance mode

Set your file system's performance mode based on IOPS required. File systems using One Zone Storage classes only support General Purpose performance mode. [Learn more](#)

- ☒ **General Purpose (Recommended)**
Ideal for a variety of diverse workloads, including high performance and latency-sensitive applications

- ☐ **Max I/O**
Designed for highly parallelized workloads that can tolerate higher latencies

EFS Practical Steps:

Part 1: Create webserver

→ First of all, create 3 EC2 instances in 3 different availability zones (ap-south-1a, ap-south-1b, ap-south-1c)
→ Create 1 security group for SSH & HTTP port (Port 22 & port 80) and one Security group for NFS (Port 2049) and attach both the Security Group to all EC2 instances → Install & start httpd service on all three EC2 instances.

Part 2: Create EFS

→ Now let's create an EFS → Go to EFS management console → Click on 'Create file system' → Name your file system 'test-efs' → For Virtual Private Cloud (VPC) option select the VPC in which you have launched 3 instances in Part 1 → Then click on 'Customize' → Keep file system type as 'Regional' → keep all other options default and then click on 'Next' → For network make sure you have selected correct VPC once again → In mount target you can see 3 Mount target as per availability zones → now remove the default security group and select security group that we have created for NFS (port 2049) for all 3 mount targets → Click on next → Now Review all the configuration and click on 'create'.

Part 3: Mount EFS on each EC2 instance

→ Now go to EC2 console and SSH into server → make sure nfs util is installed on linux if not installed then run below command to install nfs client on linux machine

```
$ sudo yum install nfs-utils -y
```

→ Now let's mount the EFS with our httpd server's html directory, for that go to EFS console and open the EFS we created → Click on 'Attach' → Select option 'Mount via DNS' → Select mount link 'Using the NFS client' → Copy and paste command in Linux console with our own mount directory, so your command will look like below:

```
$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport fs-0d9cdeff306e124fb.efs.ap-south-1.amazonaws.com:/ /var/www/html
```

Eg:

```
$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport fs-0d9cdeff306e124fb.efs.ap-south-1.amazonaws.com:/ /var/www/html
```

→ Use `df -h` Command to verify if mounting is successful → mount EFS on other ec2 instances as well → Now create index.html page on first server inside /var/www/html directory

Test:

Hit public IP of all ec2 instances and confirm if you are able to mount and use EFS successfully.

Command for Unmounting-

```
$ umount /mount-directory-name
```

Command to check mounted volumes-

```
$ df -h
```