EFS

What is EFS in AWS?

* Amazon EFS (Elastic File System) is a fully managed file storage service in AWS that allows you to share file data between multiple compute instances, including EC2, ECS, EKS, and even on-premises servers.
* EFS is a file-level, fully managed, storage provided by AWS [(Amazon Web Services)](https://www.geeksforgeeks.org/introduction-to-amazon-web-services/) that can be accessed by multiple EC2 instances concurrently.
* Amazon EFS stores each file system object in multiple availability zones ([AZs](https://www.techtarget.com/searchaws/definition/availability-zones)); an IT professional can access each file system from different AZs in the region it is located.
* The service includes file system access facilities, such as data consistency and file locks.

Features of AWS EFS

This document outlines the key features of Amazon Elastic File System (EFS), a fully managed, scalable, and highly available file storage service provided by AWS. It covers aspects such as management, scalability, availability, performance, access control, cost-effectiveness, and security features.

• **Fully Managed Service:** EFS is a fully managed service, meaning that AWS handles the infrastructure management, including hardware provisioning, patching, and backups.

• **Scalability:** EFS automatically scales your file storage capacity up or down as you add or remove files, eliminating the need for manual intervention.

• **High Availability and Durability:** EFS is designed to provide high availability and durability by storing data across multiple Availability Zones (AZs) within a region.

• **Performance Modes:** EFS offers two performance modes: General Purpose and Max I/O. The General Purpose mode is optimized for latency-sensitive use cases, while the Max I/O mode is designed for high throughput and large-scale workloads, making it suitable for big data applications.

• **Access Control:** EFS integrates with AWS Identity and Access Management (IAM) to provide fine-grained access control. Users can set permissions at the file and directory level, ensuring that only authorized users and applications can access sensitive data.

• **Serverless:** It's a "set-and-forget" service. AWS handles the underlying infrastructure, patching, and maintenance.

• **Cost-Effective:** You pay only for the storage you use. Offers different storage classes (Standard, Infrequent Access) to optimize costs based on access patterns.

• **Security:**

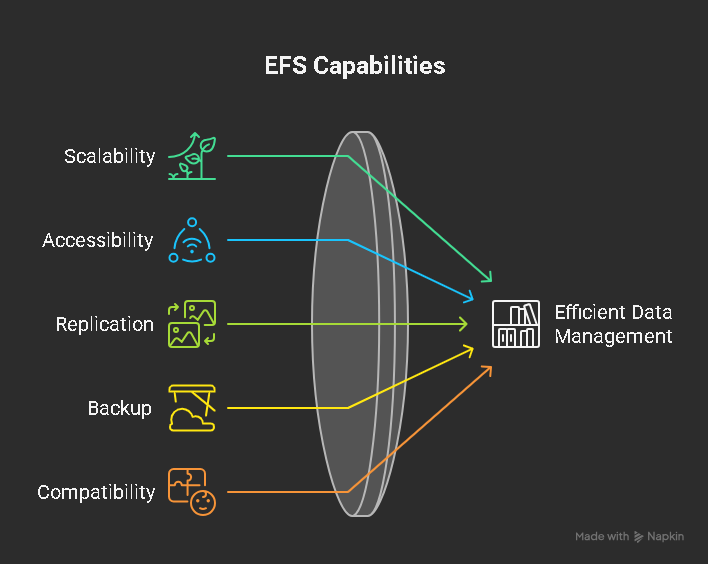
*\*Encryption:\*\* Supports encryption of data at rest (using AWS Key Management Service - KMS) and in transit.*

*\*Access Control:\*\* Integrates with AWS Identity and Access Management (IAM) and Virtual Private Cloud (VPC) security groups to control network and file-level access. Supports POSIX permissions*



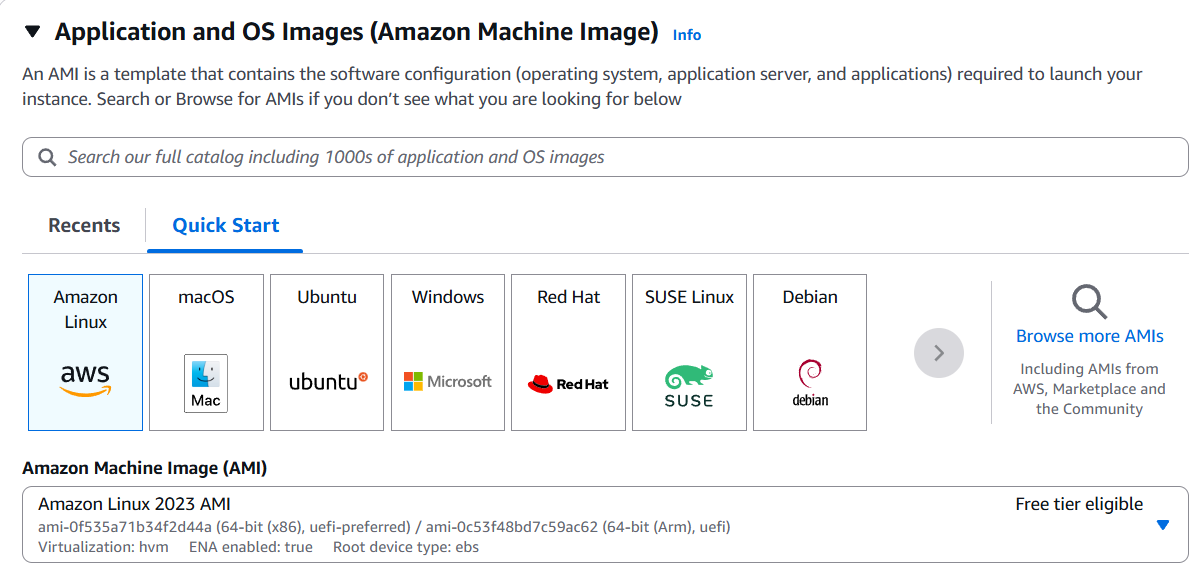
**Advantages of AWS Elastic File System (EFS):-**

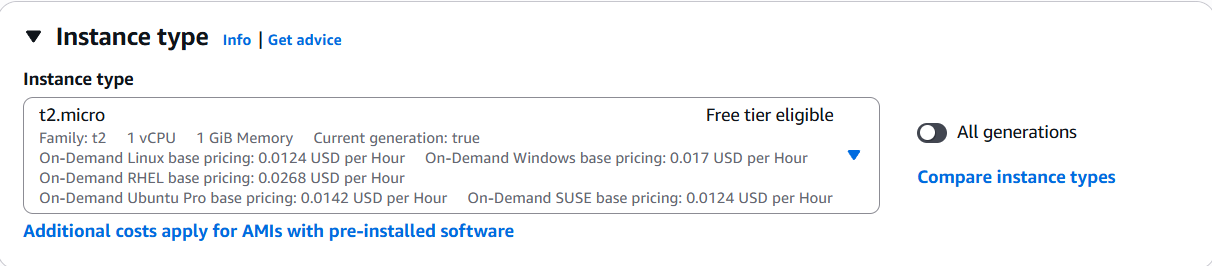
1. EFS is scalable and elastic where you can scale the EFS depending on the data that is going to be stored. The scaling is done automatically.
2. Multiple instances in AWS can access the EFS simultaneously. It makes it easy to share data across instances.
3. Within the same region, the EFS replicates the data to the multiple availability zones.
4. EFS allows you to take a backup of data from time to time so if there is any data loss you can always have backup.
5. EFS supports a wide range of POSIX file system features, making it compatible with many Linux-based applications



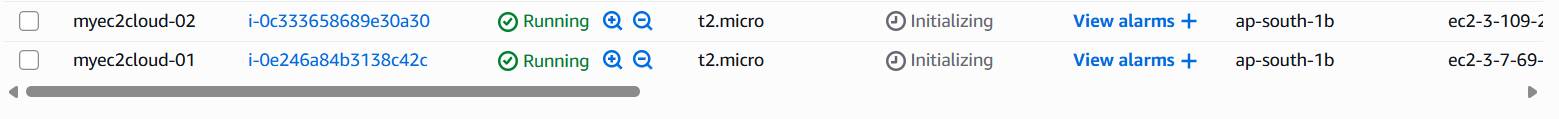
Steps to create and mount **EFS file system (Amazon EFS) to an Amazon Elastic Compute Cloud (Amazon EC2) instance:-**

**Task 1: Launching two EC2 Instances**

1. First of all, login to your aws account then make sure you are in the **Asia pacific Mumbai** Region.
2. Navigate to the search bar at the top, then search for **EC2**.
3. Next click on **launch instance**.
4. Name: Enter ***MyEC2***
5. For Amazon Machine Image (AMI): Search for **Amazon Linux 2 AMI** in the search box and click on the **Select** button.
6. **For Instance, Type:** select ***t2.micro***



1. For Key pair: Select **Create a new key pair** Button
   1. Type a name for your key pair
   2. Key pair type: **RSA**
   3. Private key file format: **.ppk**
2. Select **Create key pair** Button.
3. In Network Settings Click on **Edit**:
4. Auto-assign public IP: **Enable**
5. Select **Create new Security group**
6. Security Group Name: Enter ***EFS Security Group***
   1. To add **SSH:**
      1. Choose Type: **SSH**
      2. Source: **Anywhere**
   2. For **NFS:**
      1. Click on **Add security group rule**
      2. Choose Type: **NFS**
      3. Source: **Anywhere**
7. Keep Rest thing Default and Click on **Launch Instance** Button.
8. Select **View all Instances** to View Instance you Created
9. **Launch Status:** Your instance is now launching. Click on the instance ID and wait until the initialization status changes to **Running.**



1. Take note of the IPv4 Public IP Addresses of the EC2 instances and save them for later.

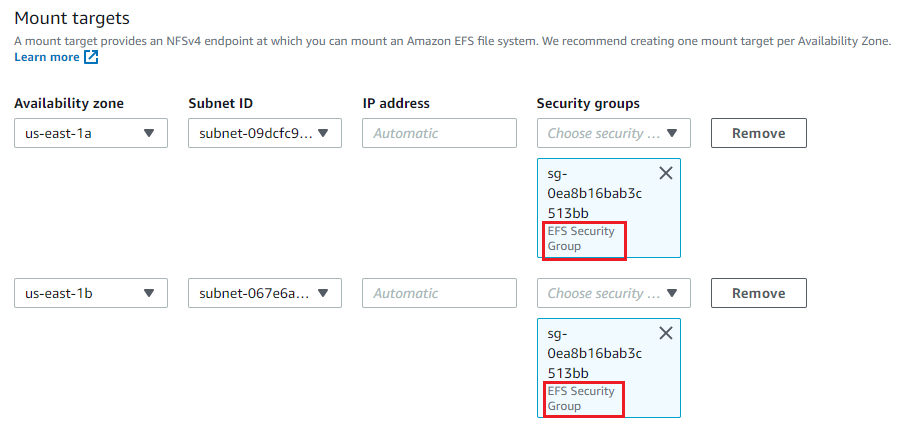
**Task 2: Creating an Elastic File System**

1. Now search for EFS in the search bar and then click on EFS.
2. Click on **Create file system**
3. Click on **Customize** button.

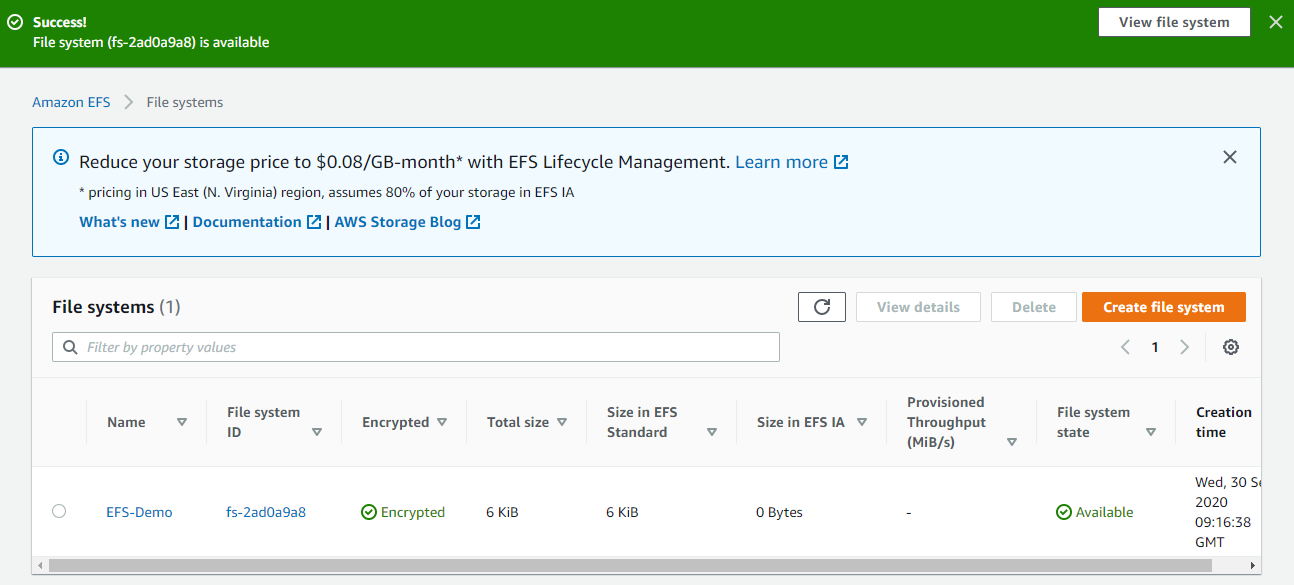
Enter the details below, Type the **Name** as ***EFS-Demo*** and make sure **default** **VPC** and **default regional** options are selected.

1. Uncheck the option of **Enable automated backups**
2. Leave everything by default and click on the **Next**button present below.
3. **Network Access**:
   * Choose the same VPC you selected while launching the EC2 instance (leave as default).

* **Mount Targets**
  + Select all the Availability Zones, and in the Security Groups, select **EFS Security Group** instead of the default value.
  + Make sure you remove the default security group and select the EFS Security Group, otherwise you will get an error in further steps.
  + Click on **Next**button



1. Click on **Next**button
2. **Review and Create**: Review the configuration thenclick on **Create**button to proceed.
3. Now the file system is created, next **mount** your EC2 Instance with the EFS File system.



**Task 3: Mount the File System to MyEC2-1 Instance**

1. Select the MyEC2-1 Instance and copy the IPv4 Public IP.
2. Run the updates using the following command:

* **Sudo su**



1. Install the NFS client as amazon-efs-utils.



1. Create a directory by the name **efs**



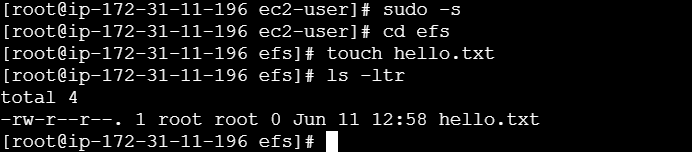
1. We have to mount the file system in this directory.
2. Navigate to the AWS console and click on the created file system. On the top-right corner, click on **Attach**
3. Copy the command of **Using the EFS mount helper.**



1. To display information for all currently-mounted file systems, we'll use the command bellow:
2. df -h



12. Create a directory in our current location:



**Task 4: Mount the File System to MyEC2-2 Instance**

1. Select the MyEC2-2 Instance and copy the IPv4 Public IP.
2. SSH into the EC2 Instance.
3. Switch to root user.
4. Run the updates using the following command:



1. Install the NFS client as amazon-efs-utils.



1. Create a directory with the name **efs**



1. We have to mount the file system in this directory.
2. To do so, navigate to the AWS console and click on the created file system. On the top-right corner, click on **Attach**

* Copy the command of **Using the EFS mount helper**
* To display information for all currently mounted file systems, we'll use the command

**df -h**

**Task 6: Testing the File System**

1. SSH into both instances in a side-by-side view on your machine, if possible.
2. Switch to root user

sudo-s

1. Navigate to the **efs directory in both the servers**using the command

cdefs

1. Create a file in any one server.

touchhello.txt

1. Check the file using the command

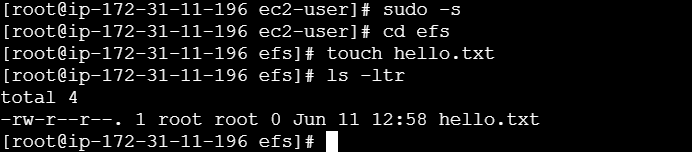
ls-ltr

1. Now go to the other server and give the command

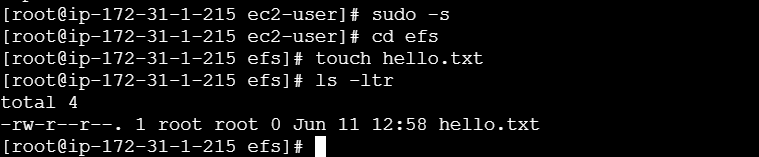
cdefs

1. You can see the file created on this server as well. This proves that our EFS is working.
2. You can try creating files (**touch**command) or directories (**mkdir**command) on other servers to continue to grow the EFS implementation.

**TERMINAL-01**



**TERMINAL-02**



**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.

How is Amazon EFS Different Than Amazon S3ata and another is to store dynamic data.

|  |  |
| --- | --- |
| **Elastic File System** | **S3** |
| 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. |

**Limitations of AWS Elastic File System(EFS): -**

There are a few limitations to consider when using AWS Elastic File System (EFS).

1. EFS only supports the Network File System (NFS) protocol, so it can only be mounted and accessed by devices that support NFS.
2. EFS has a maximum file size of 47.9 TB.
3. EFS has a maximum throughput of 1000 MB/s per file system and a maximum of 16,000 IOPS per file system.
4. EFS has a maximum number of files and directories that can be created within a single file system, which is determined by the size of the file system. For example, a 1 TB file system can support up to about 20 million files and directories.
5. EFS is only available in certain regions, and it is not possible to migrate data between regions.

