

File Sharing with AWS EFS - Hands-On Practice Using NFS Protocol

What is EFS?

- EFS stands for **Elastic File System**.
- It is a **region-specific** managed storage service in AWS.
- EFS stores data as **files** and is designed for **synchronizing data between two or more Linux EC2 instances**.
- It supports only **Linux** operating systems.
- EFS works using the **NFS (Network File System)** protocol on port **204G**.
- EFS does not provide bootable drives.

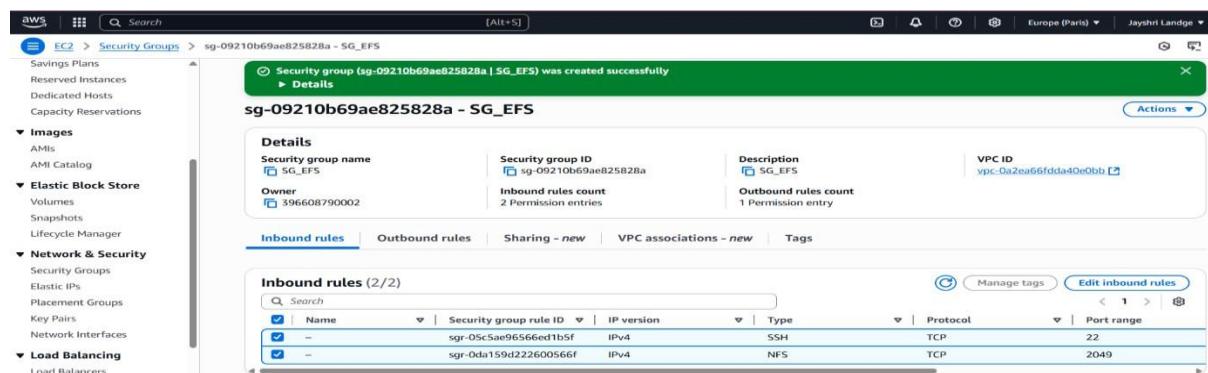
EFS vs. EBS – Key Differences

Feature	EBS	EFS
Full Form	Elastic Block Store	Elastic File System
Scope	Zone-specific	Region-specific
Data Format	Block storage	File storage
OS Support	Linux C Windows	Only Linux
Bootable Volumes	Yes	No
Protocol / Port	Not Required	NFS (Port 2049)
Purpose	Add extra volume	Sync shared directories

Step-by-Step EFS Hands-On Practice

Step 1: Create Security Group for EFS Access

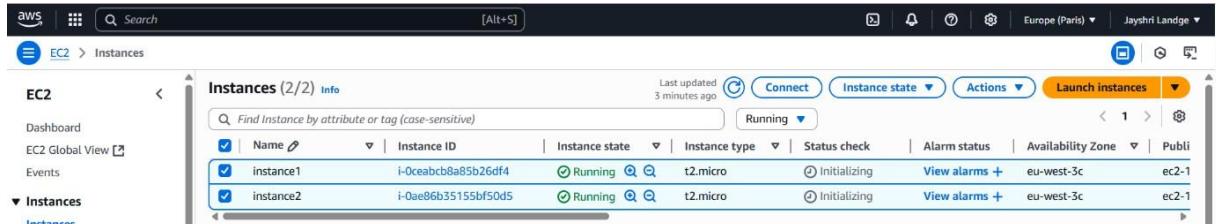
- Open ports 22 (SSH) and 204G (NFS).
- Apply this security group to both EC2 instances and EFS.



Screenshot 1: Security Group with ports 22 & 204G

Step 2: Launch Two Linux EC2 Instances

- Use Amazon Linux AMIs.
- Assign the same security group to both instances.



Screenshot 2: EC2 instance configuration

Step 3: Connect to Both Instances

- Use PuTTY or SSH to connect both instances in separate terminal sessions.

```
root@ip-172-31-41-166:~  
[ec2-user@ip-172-31-41-166 ~]$ ssh -i /path/to/keypair.pem ec2-user@  
[ec2-user@ip-172-31-41-166 ~]$  
  
[ec2-user@ip-172-31-40-145 ~]$ ssh -i /path/to/keypair.pem ec2-user@  
[ec2-user@ip-172-31-40-145 ~]$  
  
[ec2-user@ip-172-31-41-166 ~]$ sudo su -  
[root@ip-172-31-41-166 ~]#  
  
[ec2-user@ip-172-31-40-145 ~]$ sudo su -  
[root@ip-172-31-40-145 ~]#
```

```
root@ip-172-31-40-145:~  
[ec2-user@ip-172-31-40-145 ~]$ ssh -i /path/to/keypair.pem ec2-user@  
[ec2-user@ip-172-31-40-145 ~]$  
  
[ec2-user@ip-172-31-41-166 ~]$ ssh -i /path/to/keypair.pem ec2-user@  
[ec2-user@ip-172-31-41-166 ~]$  
  
[ec2-user@ip-172-31-40-145 ~]$ sudo su -  
[root@ip-172-31-40-145 ~]#  
  
[ec2-user@ip-172-31-41-166 ~]$ sudo su -  
[root@ip-172-31-41-166 ~]#
```

Screenshot 3: SSH terminal access to both instances

Step 4: Create and Attach EFS File System

- Go to **EFS service** in AWS Console.
- Click **Create File System**
- Choose VPC, and attach the **same security group (22 s 204G)**
- Once created, click **Attach**
- Copy the **mount command** provided

The screenshot shows three sequential steps in the AWS EFS console:

- Step 1: File Systems List** - Shows a single file system named "File_System_2" with ID "fs-05c6aa7c24e03c9c6". It is encrypted, has a total size of 6.00 KiB, and is available.
- Step 2: File System Details** - Shows the "File_System_2" details page. Key information includes:
 - Amazon resource name (ARN): arn:aws:elasticfilesystem:eu-west-3:356608790002:file-system/fs-05c6aa7c24e03c9c6
 - Performance mode: General Purpose
 - Throughput mode: Elastic
 - Last access timestamp: Transition into Infrequent Access (IA) 30 days(s) since last access
 - File system state: Available
 - DNS name: fs-05c6aa7c24e03c9c6.efs.eu-west-3.amazonaws.com
 - Replication overwrite protection: Enabled
- Step 3: Attach Step** - Shows the "Attach" dialog box with two options:
 - Mount via DNS
 - Mount via IPBelow the dialog, there are instructions for using the EFS mount helper or NFS client, and a link to the user guide.

Screenshot 4: EFS Creation and Attach screen

Step 5: Mount EFS on Both Instances

On both instances, run the following:

```
sudo mkdir /efs  
sudo mount -t nfs4 -o  
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-  
05c6aa7c24e03c9c6.efs.eu-west-3.amazonaws.com:/ /efs
```

```
[root@ip-172-31-41-58 ~]# mkdir /efs  
[root@ip-172-31-41-58 ~]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-05c6aa7c24e03c9c6.efs.eu-west-3.amazonaws.com:/ /efs
```

```
[root@ip-172-31-32-252 ~]# mkdir /efs  
[root@ip-172-31-32-252 ~]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-05c6aa7c24e03c9c6.efs.eu-west-3.amazonaws.com:/ /efs
```

Screenshot 5: Directory mounting output

Step 6: Sync and Test Shared Data

- On Instance 1:

```
cd /efs  
touch file1.txt file2.txt file3.txt  
cat > file1  
HELLO ALL  
cat > file2  
WELCOME TO EFS  
cat > file3  
(ELASTIC FILE SYSTEM)
```

```
mkdir Rohan
```

- On Instance 2:

```
cd /efs  
ls # You should see all data that is present in instance 1 or vice versa
```

```
[root@ip-172-31-41-58 efs]# touch file1 file2 file3  
[root@ip-172-31-41-58 efs]# cat > file1  
HELLO ALL  
[root@ip-172-31-41-58 efs]# cat > file2  
WELCOME TO EFS  
[root@ip-172-31-41-58 efs]# cat > file3  
(ELASTIC FILE SYSTEM)  
[root@ip-172-31-41-58 efs]# mkdir /Rohan  
[root@ip-172-31-41-58 efs]# ls  
file1 file2 file3  
[root@ip-172-31-41-58 efs]# mkdir Rohan  
[root@ip-172-31-41-58 efs]# ls  
file1 file2 file3 Rohan
```

```
[root@ip-172-31-32-252 ~]# cd /efs  
[root@ip-172-31-32-252 efs]# ls  
file1 file2 file3 Rohan  
[root@ip-172-31-32-252 efs]# cat file1  
HELLO ALL  
[root@ip-172-31-32-252 efs]# cat file2  
WELCOME TO EFS  
[root@ip-172-31-32-252 efs]# cat file3  
(ELASTIC FILE SYSTEM)  
[root@ip-172-31-32-252 efs]# 
```

Screenshot 6: File sync across instances

Conclusion

This hands-on implementation of AWS EFS demonstrates how to:

- Create and configure a shared file system
- Attach the same EFS to multiple Linux instances
- Use NFS protocol to mount and sync data
- Understand the real-world use case of **centralized file sharing** in AWS

EFS is a powerful solution for applications needing shared access to file data across instances in the same region.