

L1 – Project

Centralized File Sharing and Backup System

Content

1. Introduction
2. Objectives
3. Services Used
4. Architecture
5. Project Implementation details
6. Output Screenshot

Name: SAYELAKSHMI S

Introduction

The objective of this project is to design and implement a centralized file-sharing system on AWS using two EC2 instances connected to a shared Amazon EFS file system. Any file created, modified, or deleted on either EC2 instance is instantly updated in EFS, ensuring seamless collaboration. Additionally, a real-time sync mechanism using *inotifywait* continuously monitors the EFS directory and synchronizes all changes to an S3 bucket. This provides both high availability and secure backup storage for long-term retention.

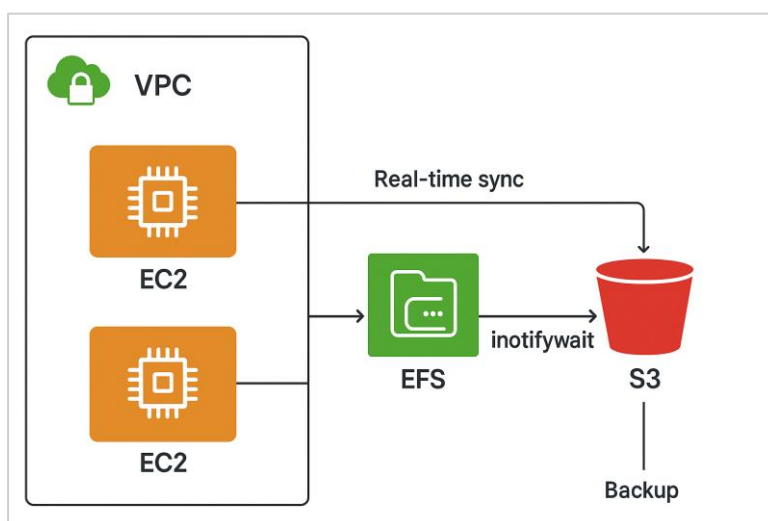
Objectives

The object of this project is to design and implement an AWS architecture where two EC2 instances, hosted in a custom VPC, share an Amazon EFS file system. Any file or directory created or modified inside the mount point **/ share/projects** on either instance must be **automatically** uploaded to an **Amazon S3 Bucket within Seconds**.

AWS Services Used

- Amazon EC2 – two Linux servers
- Amazon VPC – custom private network
- Amazon EFS – shared storage mounted on both EC2
- Amazon S3 – target storage bucket for sync
- AWS IAM – role with S3 Full Access for EC2 instances

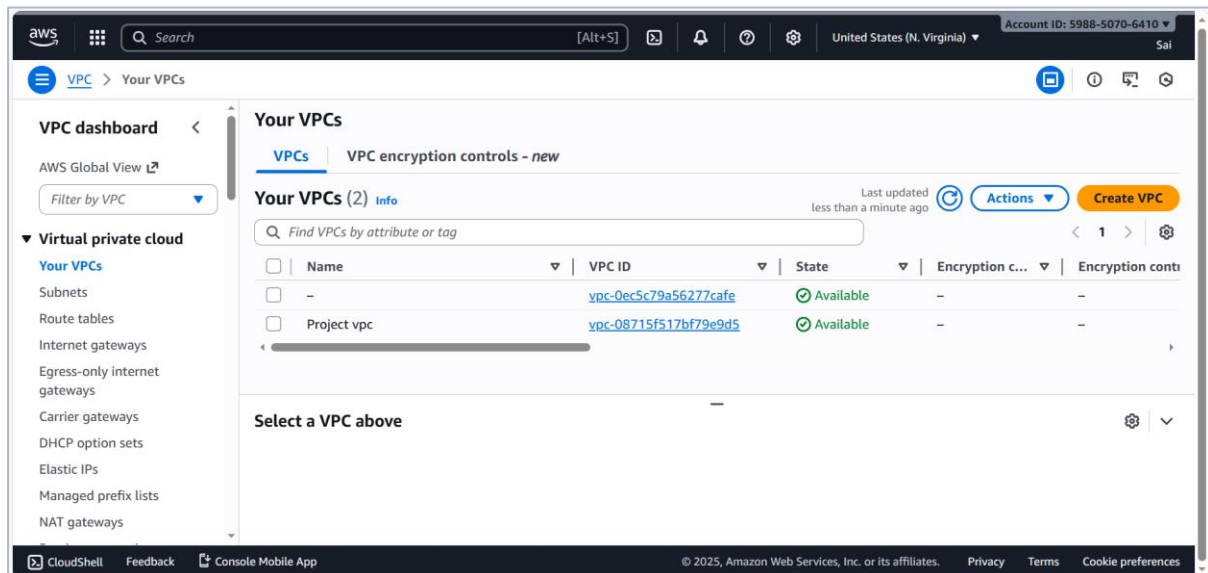
Architecture Diagram



Project Implementation setup:

Step -1 - Setup VPC & EC2 Instances.

1.Created a Custom VPC

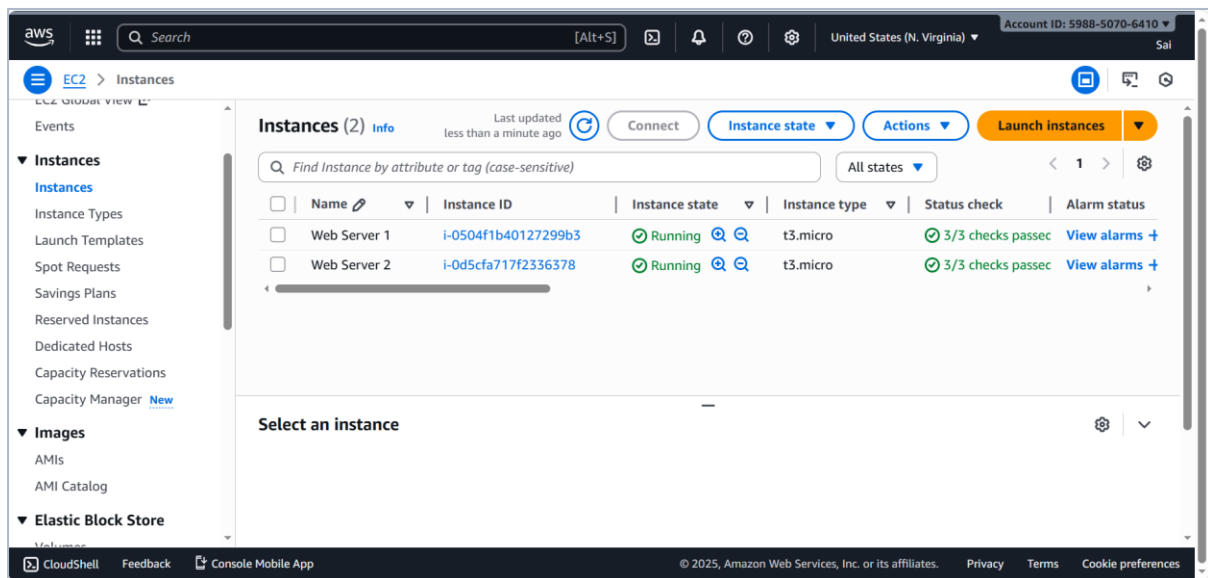


Launch **TWO EC2** instances in this VPC.

Choose AMI: Amazon Linux

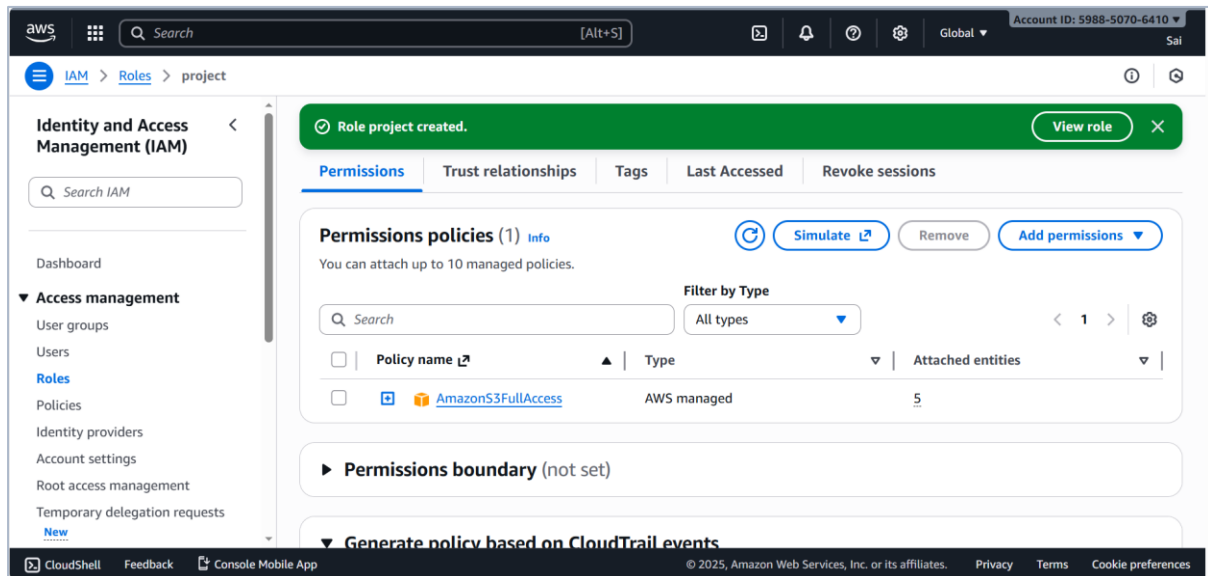
Created Web Server 1 in Subnet -1

And Web Server 2 in subnet 2

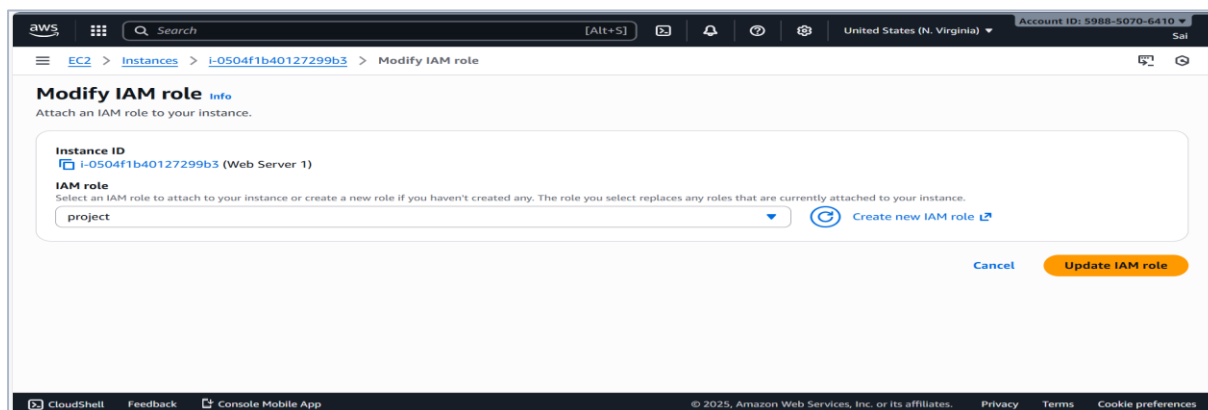
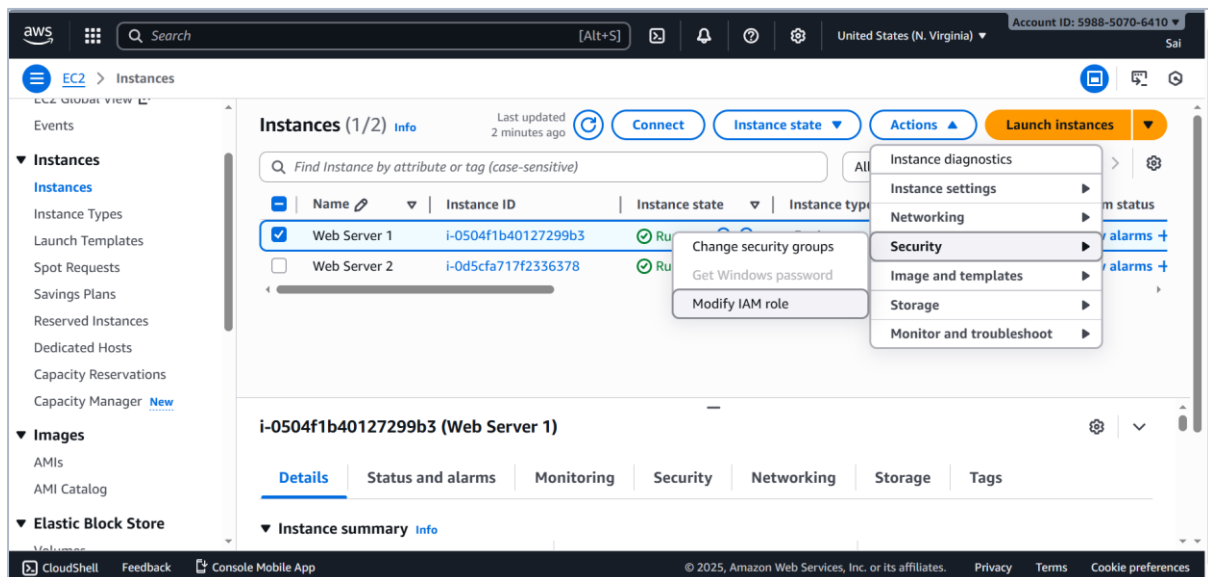


Step: 2 - Create IAM Role

Create IAM role with Amazon S3 Full access permission

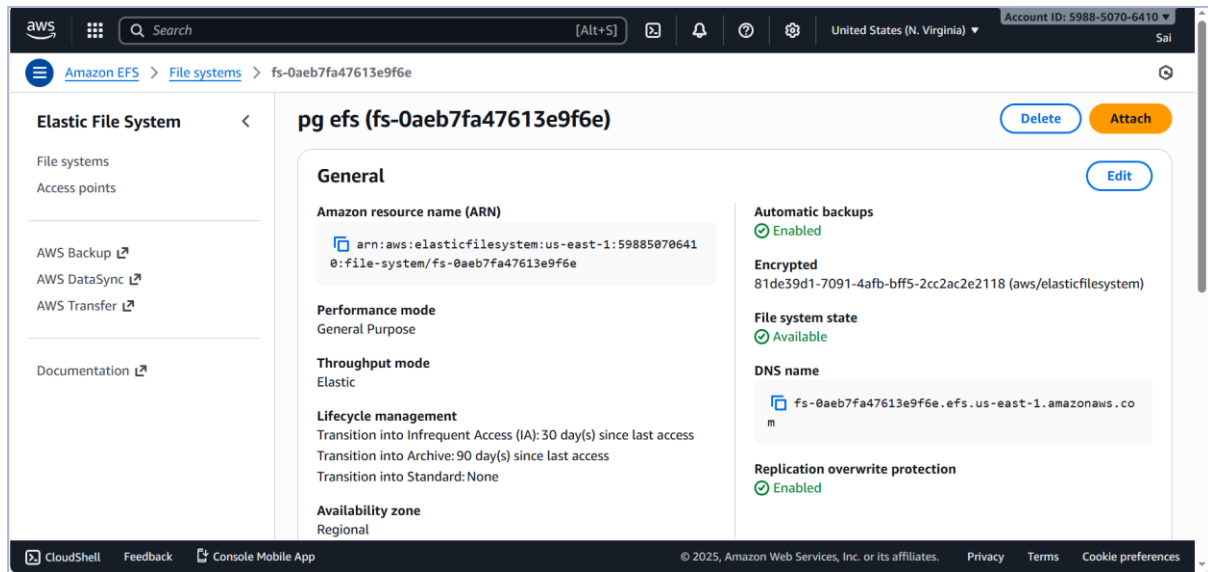


Attach the Role to two EC2 Server already created.



Step: 3 - Create & Mount EFS

1. Create an EFS File System in the same VPC



Create a mount target in the same subnet as yours EC2

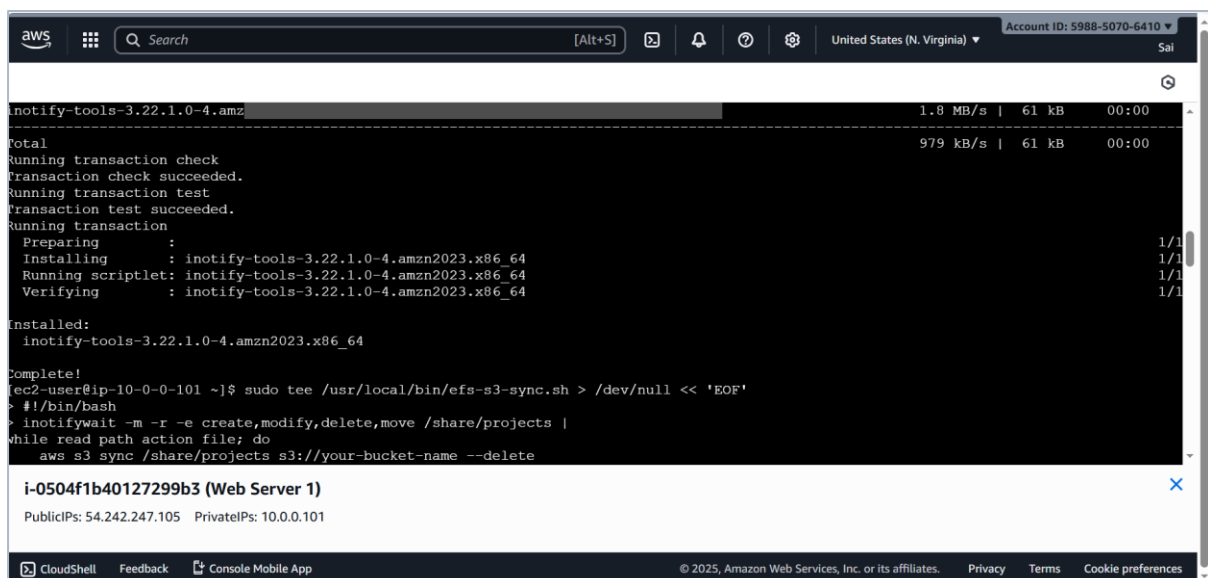
Mount EFS on both EC2 servers:

```
sudo yum install -y amazon-efs-utils
```

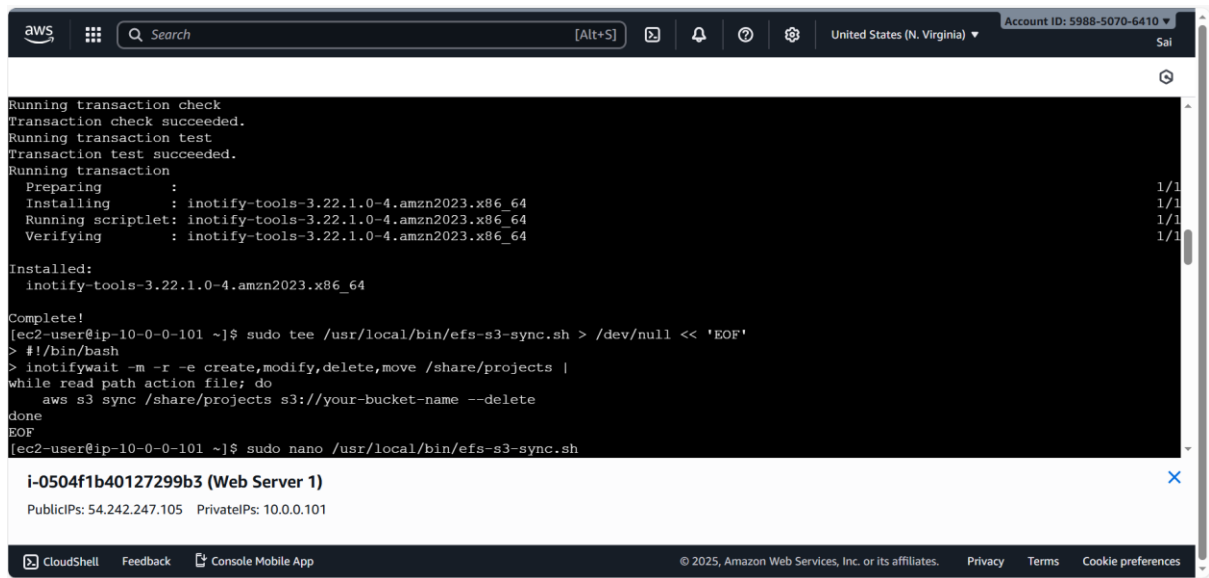
```
sudo mkdir -p /share/projects
```

```
sudo mount -t efs fs-0aeb7fa47613e9f6e:/ /share/projects
```

/share/projects will be our centralized folder.



Make the folder Writable:



```
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      : inotify-tools-3.22.1.0-4.amzn2023.x86_64 1/1
  Installing     : inotify-tools-3.22.1.0-4.amzn2023.x86_64 1/1
  Running scriptlet: inotify-tools-3.22.1.0-4.amzn2023.x86_64 1/1
  Verifying      : inotify-tools-3.22.1.0-4.amzn2023.x86_64 1/1

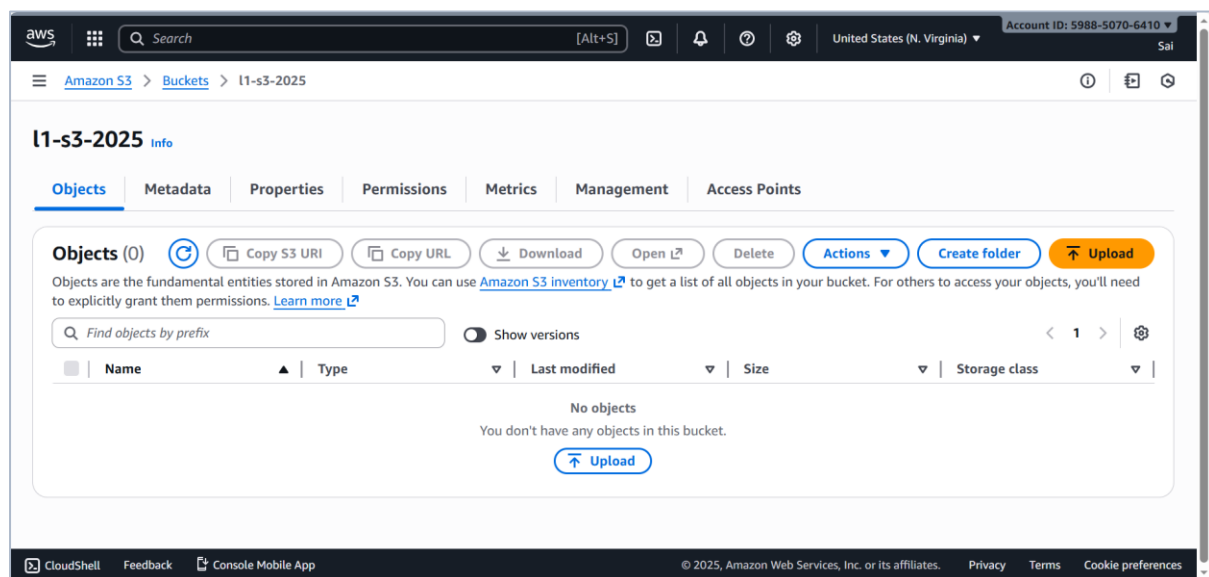
Installed:
  inotify-tools-3.22.1.0-4.amzn2023.x86_64

Complete!
[ec2-user@ip-10-0-0-101 ~]$ sudo tee /usr/local/bin/efs-s3-sync.sh > /dev/null << 'EOF'
> #!/bin/bash
> inotifywait -m -r -e create,modify,delete,move /share/projects |
while read path action file; do
  aws s3 sync /share/projects s3://your-bucket-name --delete
done
EOF
[ec2-user@ip-10-0-0-101 ~]$ sudo nano /usr/local/bin/efs-s3-sync.sh
```

i-0504f1b40127299b3 (Web Server 1)
PublicIPs: 54.242.247.105 PrivateIPs: 10.0.0.101

Step 4: - Create S3 Bucket

Bucket: s3:// l1-s3-2025



Make sure EC2 IAM Role can access it.

Step:5 - Install Dependencies

Sudo yum install -y inotify-tools

```
aws s3 sync /share/projects s3://l1-s3-2025 --delete
done

[ec2-user@ip-10-0-0-101 ~]$ sudo systemctl restart efs-s3-sync
[ec2-user@ip-10-0-0-101 ~]$ sudo systemctl status efs-s3-sync
• efs-s3-sync.service - EFS to S3 Real-time Sync Service
   Loaded: loaded (/etc/systemd/system/efs-s3-sync.service; enabled; preset: disabled)
   Active: active (running) since Sun 2025-11-23 11:19:45 UTC; 11s ago
     Main PID: 29284 (efs-s3-sync.sh)
       Tasks: 3 (limit: 1012)
      Memory: 808.0K
         CPU: 3ms
    CGroup: /system.slice/efs-s3-sync.service
            └─29284 /bin/bash /usr/local/bin/efs-s3-sync.sh
              └─29288 inotifywait -m -r -e create,modify,delete,move /share/projects
                └─29289 /bin/bash /usr/local/bin/efs-s3-sync.sh

Nov 23 11:19:45 ip-10-0-0-101.ec2.internal systemd[1]: Started efs-s3-sync.service - EFS to S3 Real-time Sync Service.
Nov 23 11:19:45 ip-10-0-0-101.ec2.internal efs-s3-sync.sh[29288]: Setting up watches. Beware: since -r was given, this may take a while!
Nov 23 11:19:46 ip-10-0-0-101.ec2.internal efs-s3-sync.sh[29288]: Watches established.
[ec2-user@ip-10-0-0-101 ~]$ echo "hello from Web Server 1" > /share/projects/test1.txt
```

i-0504f1b40127299b3 (Web Server 1)
PublicIPs: 54.242.247.105 PrivateIPs: 10.0.0.101

Ensure AWS CLI works with our IAM role:

```
aws s3 ls s3://l1-s3-2025
```

```
[ec2-user@ip-10-0-0-101 ~]$ sudo systemctl status efs-s3-sync
• efs-s3-sync.service - EFS to S3 Real-time Sync Service
   Loaded: loaded (/etc/systemd/system/efs-s3-sync.service; enabled; preset: disabled)
   Active: active (running) since Sun 2025-11-23 11:19:45 UTC; 11s ago
     Main PID: 29284 (efs-s3-sync.sh)
       Tasks: 3 (limit: 1012)
      Memory: 808.0K
         CPU: 3ms
    CGroup: /system.slice/efs-s3-sync.service
            └─29284 /bin/bash /usr/local/bin/efs-s3-sync.sh
              └─29288 inotifywait -m -r -e create,modify,delete,move /share/projects
                └─29289 /bin/bash /usr/local/bin/efs-s3-sync.sh

Nov 23 11:19:45 ip-10-0-0-101.ec2.internal systemd[1]: Started efs-s3-sync.service - EFS to S3 Real-time Sync Service.
Nov 23 11:19:45 ip-10-0-0-101.ec2.internal efs-s3-sync.sh[29288]: Setting up watches. Beware: since -r was given, this may take a while!
Nov 23 11:19:46 ip-10-0-0-101.ec2.internal efs-s3-sync.sh[29288]: Watches established.
[ec2-user@ip-10-0-0-101 ~]$ echo "hello from Web Server 1" > /share/projects/test1.txt
[ec2-user@ip-10-0-0-101 ~]$ aws s3 ls s3://l1-s3-2025
2025-11-23 11:20:50      24 test1.txt
2025-11-23 11:20:50       6 test2.txt
2025-11-23 11:20:50      15 testfile.txt
[ec2-user@ip-10-0-0-101 ~]$
```

i-0504f1b40127299b3 (Web Server 1)
PublicIPs: 54.242.247.105 PrivateIPs: 10.0.0.101

Step: 6 - Create inotify sync script

Implement a near real-time sync mechanism using inotify-based monitoring

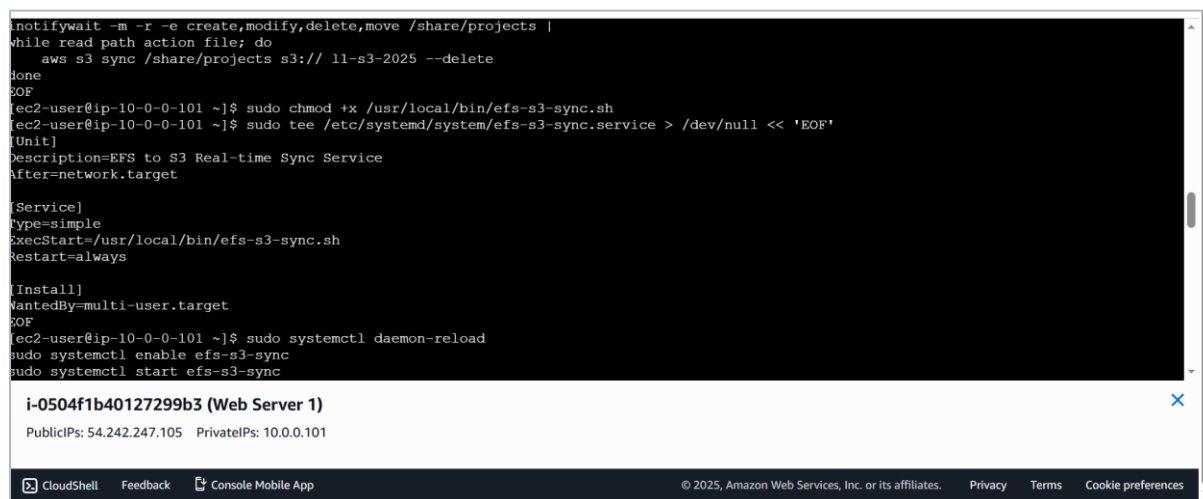
1. Create the script file:

```
/usr/local/bin/efs-s3-sync.sh
```



Make it Executable:

`chmod +x /home/ec2-user/efs-s3-watch.sh`



Step:7 - Run the Script in Background

This script that syncs EFS → S3

- `/usr/local/bin/efs-s3-sync.sh`

Runs the script in background after you log out

- `/etc/systemd/system/efs-s3-sync.service`

Check that it's running

- `systemctl status efs-s3-sync`


```
aws
[Search] [Alt+S] [Notifications] [Help] [Settings] United States (N. Virginia) Account ID: 5988-5070-6410 Sai

WantedBy=multi-user.target
EOF
[ec2-user@ip-10-0-0-101 ~]$ sudo systemctl daemon-reload
sudo systemctl enable efs-s3-sync
sudo systemctl start efs-s3-sync
Created symlink /etc/systemd/system/multi-user.target.wants/efs-s3-sync.service → /etc/systemd/system/efs-s3-sync.service.
[ec2-user@ip-10-0-0-101 ~]$ systemctl status efs-s3-sync
● efs-s3-sync.service - EFS to S3 Real-time Sync Service
   Loaded: loaded (/etc/systemd/system/efs-s3-sync.service; enabled; preset: disabled)
   Active: active (running) since Sun 2025-11-23 11:10:36 UTC; 14s ago
     Main PID: 28371 (efs-s3-sync.sh)
        Tasks: 3 (limit: 1012)
       Memory: 824.0K
          CPU: 5ms
      CGroup: /system.slice/efs-s3-sync.service
              └─28371 /bin/bash /usr/local/bin/efs-s3-sync.sh
                 └─28372 inotifywait -m -r -e create,modify,delete,move /share/projects
                    └─28373 /bin/bash /usr/local/bin/efs-s3-sync.sh

Nov 23 11:10:36 ip-10-0-0-101.ec2.internal systemd[1]: Started efs-s3-sync.service - EFS to S3 Real-time Sync Service.
Nov 23 11:10:36 ip-10-0-0-101.ec2.internal efs-s3-sync.sh[28372]: Setting up watches. Beware: since -r was given, this may take a while!
Nov 23 11:10:36 ip-10-0-0-101.ec2.internal efs-s3-sync.sh[28372]: Watches established.

i-0504f1b40127299b3 (Web Server 1)
PublicIPs: 54.242.247.105 PrivateIPs: 10.0.0.101

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Step:8 - Test Automatic Sync

1.Create a file in EFS:

```
echo "hello from Web Server 1" > /share/projects/test1.txt
```

2. Wait a few seconds.

3. Check S3:

```
aws s3 ls s3://l1-s3-2025
```

- File should appear automatically.
- Check log: cat /var/log/efs-s3-sync.log

Step: 9 - Same command run into the Web Server 2 (EC2-2)

```
aws
[Search] [Alt+S] [Notifications] [Help] [Settings] United States (N. Virginia) Account ID: 5988-5070-6410 Sai

● efs-s3-sync.service - EFS to S3 Real-time Sync Service
   Loaded: loaded (/etc/systemd/system/efs-s3-sync.service; enabled; preset: disabled)
   Active: active (running) since Sun 2025-11-23 11:26:37 UTC; 3min 57s ago
     Main PID: 30452 (efs-s3-sync.sh)
        Tasks: 3 (limit: 1053)
       Memory: 796.0K
          CPU: 5ms
      CGroup: /system.slice/efs-s3-sync.service
              └─30452 /bin/bash /usr/local/bin/efs-s3-sync.sh
                 └─30453 inotifywait -m -r -e create,modify,delete,move /share/projects
                    └─30454 /bin/bash /usr/local/bin/efs-s3-sync.sh

Nov 23 11:26:37 ip-10-0-1-74.ec2.internal systemd[1]: Started efs-s3-sync.service - EFS to S3 Real-time Sync Service.
Nov 23 11:26:37 ip-10-0-1-74.ec2.internal efs-s3-sync.sh[30453]: Setting up watches. Beware: since -r was given, this may take a while!
Nov 23 11:26:37 ip-10-0-1-74.ec2.internal efs-s3-sync.sh[30453]: Watches established.
[ec2-user@ip-10-0-1-74 ~]$ echo "hello-from-Web Server 2" > /share/projects/ec2-2.txt
[ec2-user@ip-10-0-1-74 ~]$ aws s3 ls s3://l1-s3-2025
2025-11-23 11:31:20      24 ec2-2.txt
2025-11-23 11:20:50      24 test1.txt
2025-11-23 11:20:50       6 test2.txt
2025-11-23 11:25:38       6 testfile.txt
[ec2-user@ip-10-0-1-74 ~]$

i-0d5cfa717f2336378 (Web Server 2)
PublicIPs: 44.222.148.174 PrivateIPs: 10.0.1.74

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Step:10 - Make Script Start on Boot (systemd service)

1. Create a systemd service file:

ls -l /etc/systemd/system/ | grep efs

The screenshot shows a terminal window in AWS CloudShell. The user has created a script at `/usr/local/bin/efs-s3-sync.sh` and a systemd service file at `/etc/systemd/system/efs-s3-sync.service`. The service is currently active and running. The terminal output is as follows:

```
[ec2-user@ip-10-0-1-74 ~]$ cat /usr/local/bin/efs-s3-sync.sh
#!/bin/bash
inotifywait -m -r -e create,modify,delete,move /share/projects |
while read path action file; do
    aws s3 sync /share/projects s3://11-s3-2025
    --delete
done
[ec2-user@ip-10-0-1-74 ~]$ ls -l /etc/systemd/system/ | grep efs
-rw-r--r--. 1 root root 189 Nov 23 11:24 efs-s3-sync.service
[ec2-user@ip-10-0-1-74 ~]$ ls -l /usr/local/bin/efs-s3-sync.sh
-rwxr-xr-x. 1 root root 172 Nov 23 11:23 /usr/local/bin/efs-s3-sync.sh
[ec2-user@ip-10-0-1-74 ~]$ sudo systemctl status efs-s3-sync
● efs-s3-sync.service - EFS to S3 Real-time Sync Service
   Loaded: loaded (/etc/systemd/system/efs-s3-sync.service; enabled; preset: disabled)
   Active: active (running) since Sun 2025-11-23 11:26:37 UTC; 3min 57s ago
     Main PID: 30452 (efs-s3-sync.sh)
        Tasks: 3 (limit: 1053)
      Memory: 796.0K
         CPU: 5ms
    CGroup: /system.slice/efs-s3-sync.service
            └─30452 /bin/bash /usr/local/bin/efs-s3-sync.sh
              └─30453 inotifywait -m -r -e create,modify,delete,move /share/projects
```

Enable auto-start at boot

- `sudo systemctl enable efs-s3-sync`

Start service now

- `sudo systemctl start efs-s3-sync`

Check if running

- `systemctl status efs-s3-sync`

Now the service is active (running), Now test file sync.

This screenshot is identical to the one above, showing the same terminal output where the `efs-s3-sync.service` is active and running.

Result - The Instances automatically uploaded to an Amazon S3 Bucket within seconds.

