

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.

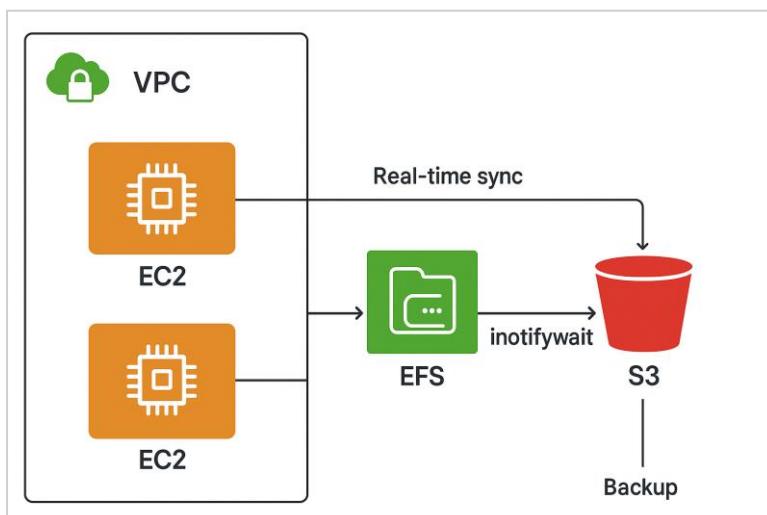
Objectives

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

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with 'Virtual private cloud' expanded, showing options like Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, and NAT gateways. The main area is titled 'Your VPCs' and shows two entries: 'Project vpc' (selected) and 'ydc-0ec5c79a56277cafe'. A message 'Select a VPC above' is displayed below the table. At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information for 2025, Amazon Web Services, Inc. or its affiliates, and links for Privacy, Terms, and Cookie preferences.

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

The screenshot shows the AWS EC2 Instances dashboard. On the left, there's a sidebar with 'Instances' expanded, showing options like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager (New), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes). The main area is titled 'Instances' and shows two entries: 'Web Server 1' and 'Web Server 2', both in the 'Running' state. Below the table, there's a message 'Select an instance'. At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information for 2025, Amazon Web Services, Inc. or its affiliates, and links for Privacy, Terms, and Cookie preferences.

Step: 2 - Create IAM Role

Create IAM role with Amazon S3 Full access permission

The screenshot shows the AWS IAM console with the 'Roles' page selected. A green banner at the top right says 'Role project created.' Below it, the 'Permissions' tab is active. Under 'Permissions policies', there is one policy listed: 'AmazonS3FullAccess'. The 'Attached entities' section shows 5 entities. There are tabs for 'Trust relationships', 'Tags', 'Last Accessed', and 'Revoke sessions'. At the bottom, there are links for 'CloudShell', 'Feedback', and 'Console Mobile App'.

Attach the Role to two EC2 Server already created.

The screenshot shows the AWS EC2 Instances page. It lists two instances: 'Web Server 1' and 'Web Server 2'. A context menu is open over 'Web Server 1', with the 'Modify IAM role' option highlighted. Other options in the menu include 'Instance diagnostics', 'Instance settings', 'Networking', 'Security', 'Image and templates', 'Storage', and 'Monitor and troubleshoot'. The main pane shows details for 'Web Server 1'.

The screenshot shows the 'Modify IAM role' dialog box. At the top, it says 'Attach an IAM role to your instance.' Below that is a table with 'Instance ID' (i-0504f1b40127299b3) and 'IAM role' (a dropdown menu). The dropdown menu has 'project' selected. At the bottom right are 'Cancel' and 'Update IAM role' buttons.

Step: 3 - Create & Mount EFS

1.Create an EFS File System in the same VPC

The screenshot shows the AWS EFS console interface. On the left, there's a sidebar with 'Elastic File System' selected, followed by 'File systems' and 'Access points'. Below these are links for 'AWS Backup', 'AWS DataSync', and 'AWS Transfer'. At the bottom of the sidebar is a link to 'Documentation'. The main content area is titled 'pg efs (fs-0aeb7fa47613e9f6e)'. It has a 'General' tab selected. Under 'Amazon resource name (ARN)', it shows 'arn:aws:elasticfilesystem:us-east-1:59885070641:file-system/fs-0aeb7fa47613e9f6e'. There are sections for 'Performance mode' (set to 'General Purpose'), 'Throughput mode' (set to 'Elastic'), 'Lifecycle management' (with options for IA, Archive, and Standard), 'Availability zone' (set to 'Regional'), 'Automatic backups' (enabled), 'Encrypted' (status 81de39d1-7091-4afb-bff5-2cc2ac2e2118), 'File system state' (available), 'DNS name' (fs-0aeb7fa47613e9f6e.efs.us-east-1.amazonaws.com), and 'Replication overwrite protection' (enabled). At the top right, there are 'Delete' and 'Attach' buttons, and an 'Edit' button. The bottom of the screen includes standard AWS navigation links like CloudShell, Feedback, and Console Mobile App, along with copyright information for 2025 and links to Privacy, Terms, and Cookie preferences.

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.

The screenshot shows the AWS CloudWatch Terminal interface. The terminal window displays the output of a yum install command for the 'amazon-efs-utils' package. The output shows the package being downloaded and installed, with transaction statistics: 1.8 MB/s download speed, 61 kB size, and 00:00 duration. The package version is 3.22.1.0-4.amzn2023.x86_64. After the install, a user runs a script to sync the local EFS mount point with an S3 bucket. The terminal shows the user's session on an EC2 instance with IP i-0504f1b40127299b3 (Web Server 1), with public and private IP addresses listed. The bottom of the screen includes standard AWS navigation links like CloudShell, Feedback, and Console Mobile App, along with copyright information for 2025 and links to Privacy, Terms, and Cookie preferences.

Make the folder Writable:

The screenshot shows the AWS CloudShell interface. The terminal window displays the following command execution:

```
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 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
```

Below the terminal, the EC2 instance details are shown:

i-0504f1b40127299b3 (Web Server 1)
Public IPs: 54.242.247.105 Private IPs: 10.0.0.101

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

Step 4: - Create S3 Bucket

Bucket: s3:// l1-s3-2025

The screenshot shows the AWS S3 Buckets page. A new bucket named "l1-s3-2025" is listed under the "Buckets" section.

l1-s3-2025 info

Objects (0) Actions Create folder Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix Show versions

Name	Type	Last modified	Size	Storage class
No objects	You don't have any objects in this bucket.			

Upload

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

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
             ├─29290 /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)

Public IPs: 54.242.247.105 Private IPs: 10.0.0.101

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

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
             ├─29290 /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)

Public IPs: 54.242.247.105 Private IPs: 10.0.0.101

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

Step: 6 - Create inotify sync script

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

- Create the script file:

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

The screenshot shows the AWS CloudShell interface. At the top, there's a navigation bar with icons for AWS Lambda, CloudWatch Metrics, and CloudWatch Logs, followed by a search bar and account information ('Account ID: 5988-5070-6410' and 'United States (N. Virginia)'). Below the search bar is a terminal window titled 'GNU nano 8.3'. The script content is:

```
#!/bin/bash
inotifywait -m -r -e create,modify,delete,move /share/projects |
while read path action file; do
    aws s3 sync /share/projects s3://ll-s3-2025 --delete
done
```

The terminal has a menu bar at the bottom with options like Help, Write Out, Where Is, Cut, Paste, Execute, Location, Go To Line, Undo, Redo, Set Mark, and Copy. At the very bottom of the terminal window, it says 'i-0504f1b40127299b3 (Web Server 1)' and 'PublicIPs: 54.242.247.105 PrivateIPs: 10.0.0.101'.

Make it Executable:

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

This screenshot shows the AWS CloudShell terminal after running the chmod command. The terminal output includes the command itself and the creation of a systemd service file:

```
[ec2-user@ip-10-0-0-101 ~]$ sudo chmod +x /usr/local/bin/efs-s3-sync.sh
[ec2-user@ip-10-0-0-101 ~]$ sudo tee /etc/systemd/system/efs-s3-sync.service > /dev/null << 'EOF'
[Unit]
Description=EFS to S3 Real-time Sync Service
After=network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/efs-s3-sync.sh
Restart=always

[Install]
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
```

At the bottom of the terminal, it says 'i-0504f1b40127299b3 (Web Server 1)' and 'PublicIPs: 54.242.247.105 PrivateIPs: 10.0.0.101'.

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`

```

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.
● 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)

```

● 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)

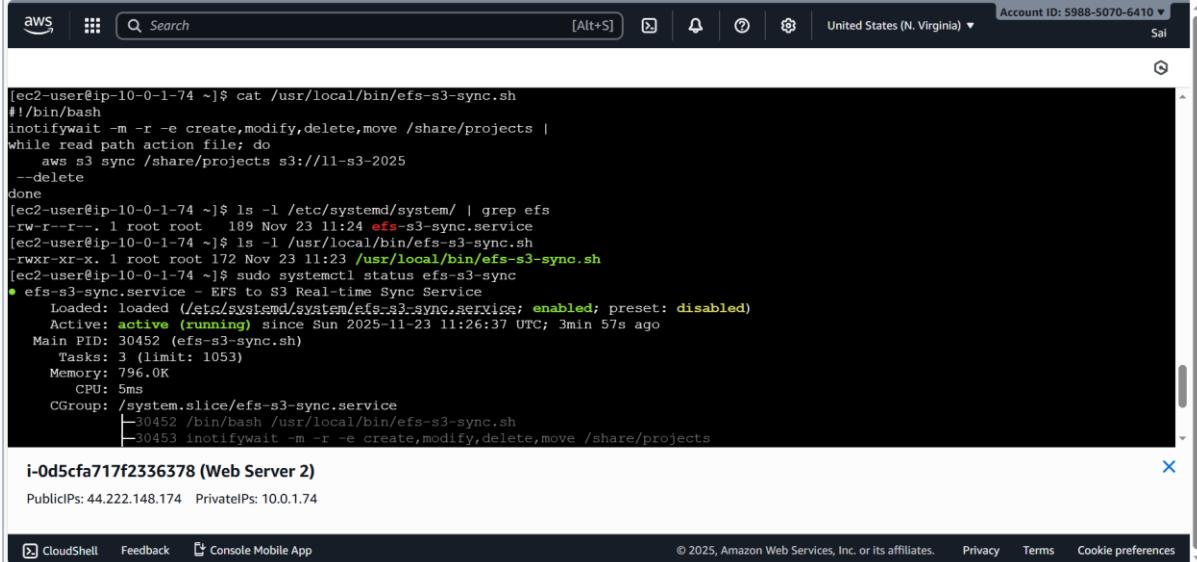
PublicIPs: 44.222.148.174 PrivateIPs: 10.0.1.74

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
```



```
[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
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

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

Enable auto-start at boot

- sudo systemctl enable efs-s3-sync

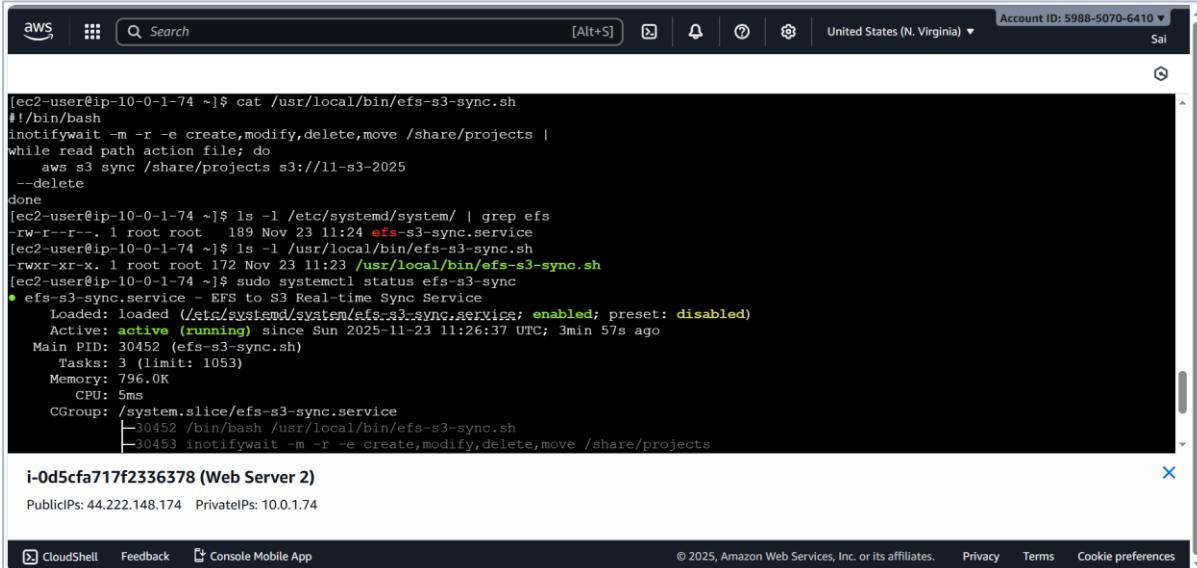
Start service now

- sudo systemctl start efs-s3-sync

Check if running

- systemctl status efs-s3-sync

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



```
[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
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

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

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

The screenshot shows the AWS S3 console interface. At the top, the navigation bar includes the AWS logo, a search bar, and account information (Account ID: 5988-5070-6410). Below the navigation bar, the path is shown as 'Amazon S3 > Buckets > l1-s3-2025'. The 'Objects' tab is selected, and the main content area displays a table of objects. The table has columns for Name, Type, Last modified, Size, and Storage class. There are four entries:

Name	Type	Last modified	Size	Storage class
ec2-2.txt	txt	November 23, 2025, 17:01:20 (UTC+05:30)	24.0 B	Standard
test1.txt	txt	November 23, 2025, 16:50:50 (UTC+05:30)	24.0 B	Standard
test2.txt	txt	November 23, 2025, 16:50:50 (UTC+05:30)	6.0 B	Standard
testfile.txt	txt	November 23, 2025, 16:55:38 (UTC+05:30)	6.0 B	Standard

At the bottom of the page, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information (© 2025, Amazon Web Services, Inc. or its affiliates.) and links for Privacy, Terms, and Cookie preferences.