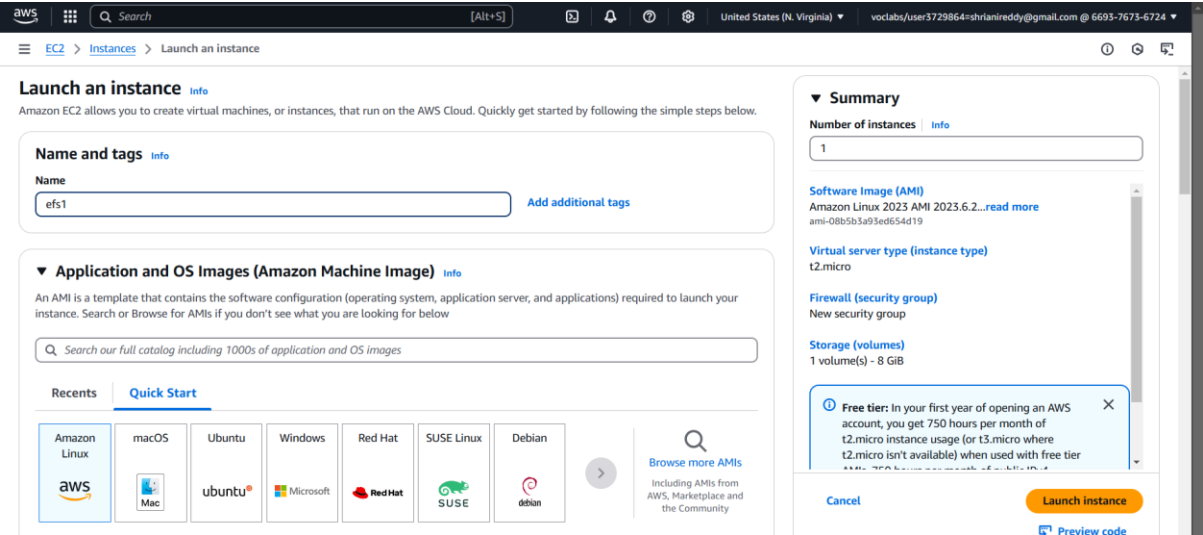EFS

Amazon **Elastic File System (EFS)** is a **fully managed, scalable, and shared file storage** service for **Linux-based** EC2 instances. It allows multiple instances to access the same file system **simultaneously**, making it ideal for applications that require shared storage.

Navigate to the AWS console and create a ec2 instance, select LINUX OS

Let AMI and architecture be default

Select t2.micro instance type.

Create a ppk key-pair

Edit the Network settings and create a security group

▼ **Network settings** Info

VPC - *required* | Info

| vpc-001d4b527e6a58a9b | (default) ▼ | ↻ |
| 172.31.0.0/16 | | |

Subnet | Info

| No preference ▼ | ↻ Create new subnet ⧉ |

Auto-assign public IP | Info

| Enable ▼ |

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

| ● Create security group | ○ Select existing security group |

Security group name - *required*

| launch-wizard-5 |

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required* | Info

| launch-wizard-5 created 2025-03-13T04:51:55.635Z |

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)   Remove

Select NFS type and add CIDR

| 0.0.0.0/0 ✕ |

▼ Security group rule 4 (TCP, 2049, 0.0.0.0/0)   Remove

Type | Info

| NFS ▼ |

Protocol | Info

| TCP |

Port range | Info

| 2049 |

Source type | Info

| Custom ▼ |

Source | Info

| 🔍 Add CIDR, prefix list or security group |

| 0.0.0.0/0 ✕ |

Description – *optional* | Info

| e.g. SSH for admin desktop |

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.   ✕

Add security group rule

Let the storage settings be default

Launch the instance

Create another instance and name it, select the LINUX operating system

Select the AMI, Architecture and Instance type

Select the existing key-pair and select the security group that you created in the previous instance

Configure the storage system

Launch the instance

Successfully initiated launch of instance (i-0a4bd425b47129e8a)

The 2 instances should be in 2 different zones of the same region

☐	efs2	i-0a4bd425b47129e8a	⊘ Running ⊕ ⊖	t2.micro	⊘ 2/2 checks passed	View alarms +	us-east-1c	ec2-3-89
☐	efs1	i-0514c2b2a83992124	⊘ Running ⊕ ⊖	t2.micro	⊘ 2/2 checks passed	View alarms +	us-east-1d	ec2-54-2

Connect to efs1, and open putty, copy the host id and paste it

Give the path to the ppk key-pair for authentication
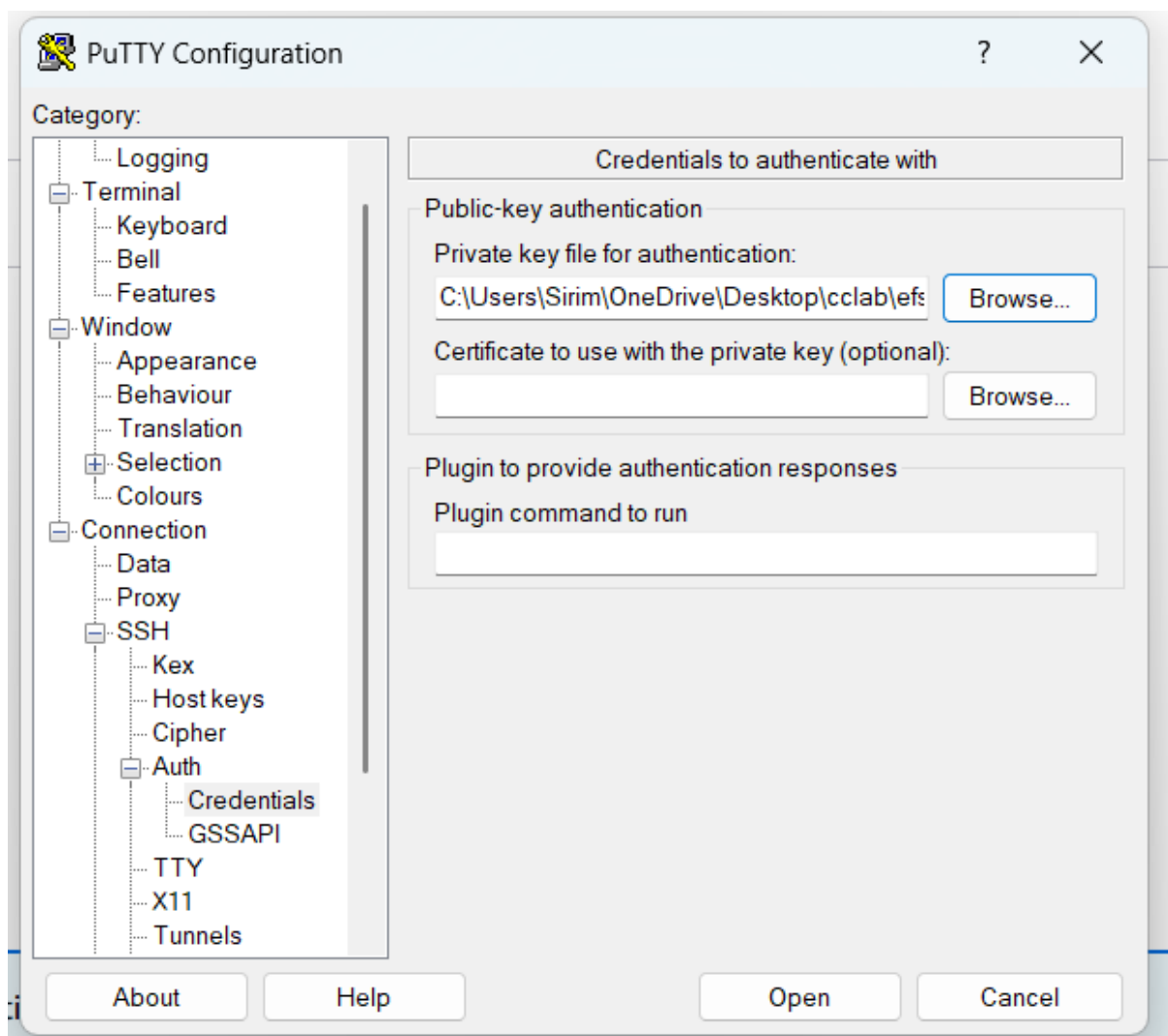
Connect to the instance.

Similarly open the putty and connect to efs2 with putty by giving its host id and the key pair ppk

```
ec2-user@ip-172-31-86-106:~                              —    □    X
Using username "ec2-user".
Authenticating with public key "efs-putty"
      #_
  ~\_  ####_            Amazon Linux 2023
 ~~  \_#####\
 ~~      \###|
 ~~       \#/  ___     https://aws.amazon.com/linux/amazon-linux-2023
  ~~       V~' '->
   ~~~         /
     ~~._.   _/
        _/ _/
       _/m/'
[ec2-user@ip-172-31-86-106 ~]$ █
```

Put the putty terminals side by side

Run the following commands

sudo su

mkdir efs
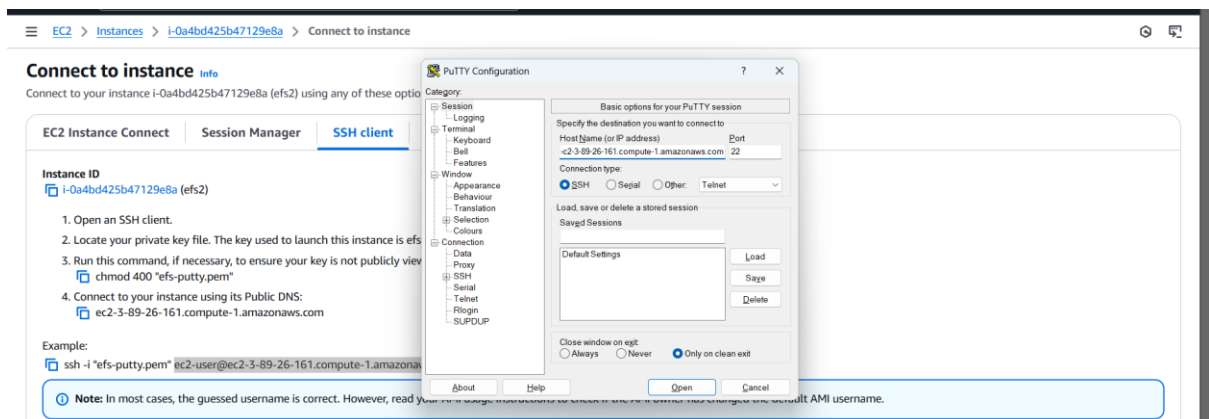
Yum install -y amazon-efs-utils

Open your AWS console and go to the EFS services

Create a file system

Navigate to the EFS that you created

Navigate to network settings and select the security group that you created for the instances in those zones

us-east-1c	subnet-09fa5ff8d5d61f3b	172.31.88.66	Choose security groups ▼	Remove
			sg-06308802cf8665ecc ✕ launch-wizard-5	
us-east-1d	subnet-02e43ba98628d77	172.31.26.72	Choose security groups ▼	Remove
			sg-06308802cf8665ecc ✕ launch-wizard-5	

Save the changes

⊘ **Success!**
Submitted all mount target changes successfully for file system (fs-0f7fd7a2f123464dd)   ✕   Ⓖ

Amazon EFS > File systems > fs-0f7fd7a2f123464dd

EFS-ONE (fs-0f7fd7a2f123464dd)   Delete   **Attach**

Click on attach and copy the NFS client command

Elastic File System   ✕   Amazon EFS > File systems > fs-0f7fd7a2f123464dd   Ⓖ

Attach   ✕

Mount your Amazon EFS file system on a Linux instance. Learn more ↗

● Mount via DNS   ○ Mount via IP

Using the EFS mount helper:

```
sudo mount -t efs -o tls fs-0f7fd7a2f123464dd:/ efs
```

Using the NFS client:

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-0f7fd7a2f123464dd.efs.us-east-1.amazonaws.com:/ efs
```

See our user guide for more information. Learn more ↗

Close

Paste the command in both the terminals and run

```
Complete!
[root@ip-172-31-24-164 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=10485
76,hard,timeo=600,retrans=2,noresvport fs-0f7fd7a2f123464dd.efs.us-east-1.amazonaws.com:/ efs
[root@ip-172-31-24-164 ec2-user]# cd efs
[root@ip-172-31-24-164 efs]# touch f1
[root@ip-172-31-24-164 efs]# ls
f1  f2
[root@ip-172-31-24-164 efs]#
```

```
[root@ip-172-31-86-106 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=10485
76,hard,timeo=600,retrans=2,noresvport fs-0f7fd7a2f123464dd.efs.us-east-1.amazonaws.com:/ efs
[root@ip-172-31-86-106 ec2-user]# cd efs
[root@ip-172-31-86-106 efs]# ls
f1
[root@ip-172-31-86-106 efs]# touch f2
[root@ip-172-31-86-106 efs]# []
```

Create files in one console those must be reflected in the other terminal too.

```
[root@ip-172-31-24-164 efs]# git clone https://github.com/manognavadla/AWSExample.git
Cloning into 'AWSExample'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
[root@ip-172-31-24-164 efs]# ls
AWSExample  f1  f2
[root@ip-172-31-24-164 efs]#
```

```
[root@ip-172-31-86-106 efs]# git clone https://github.com/manognavadla/sesample.git
bash: git: command not found
[root@ip-172-31-86-106 efs]# ls
AWSExample  f1  f2
[root@ip-172-31-86-106 efs]#
[root@ip-172-31-86-106 efs]#
[root@ip-172-31-86-106 efs]#
[root@ip-172-31-86-106 efs]#
[root@ip-172-31-86-106 efs]#
[root@ip-172-31-86-106 efs]#
```

If you clone the git repository it will be shared to the other instance too

```
[root@ip-172-31-24-164 efs]# git clone https://github.com/manognavadla/AWSExample.git
Cloning into 'AWSExample'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
[root@ip-172-31-24-164 efs]# ls
AWSExample  f1  f2
[root@ip-172-31-24-164 efs]#
```

```
[root@ip-172-31-86-106 efs]# git clone https://github.com/manognavadla/sesample.git
bash: git: command not found
[root@ip-172-31-86-106 efs]# ls
AWSExample  f1  f2
[root@ip-172-31-86-106 efs]#
[root@ip-172-31-86-106 efs]#
[root@ip-172-31-86-106 efs]#
[root@ip-172-31-86-106 efs]#
[root@ip-172-31-86-106 efs]#
[root@ip-172-31-86-106 efs]#
```