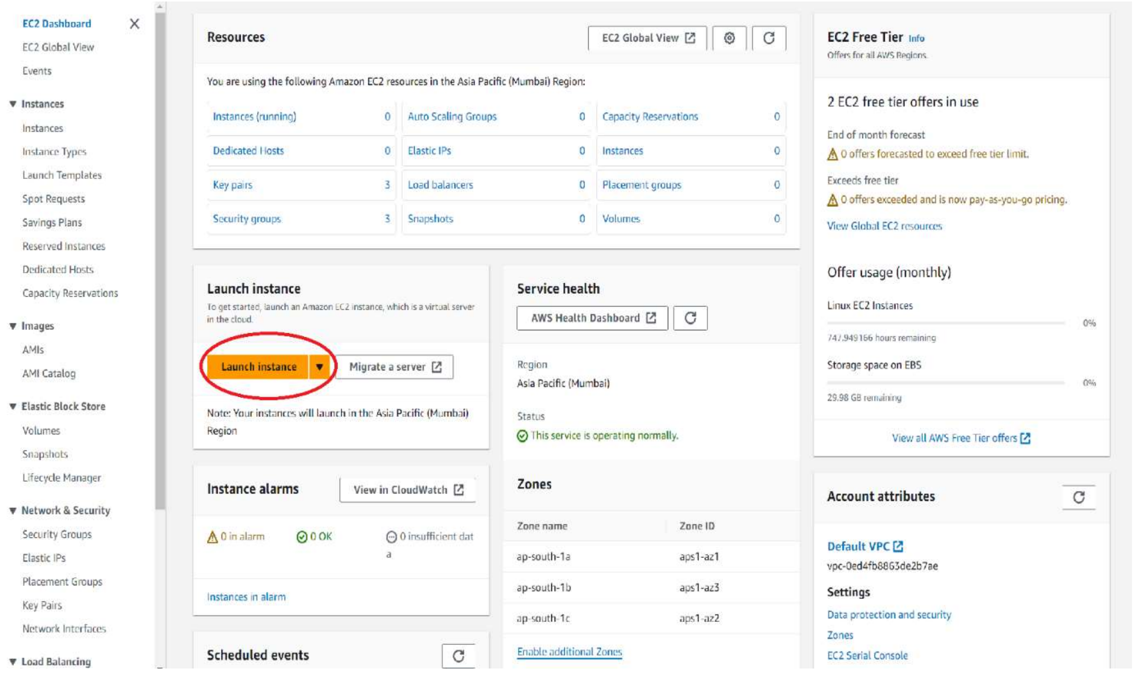
<u>Tasks To Be Performed:</u>

1. Create an EFS and connect it to 3 different EC2 instances. Make sure that all instances have different operating systems. For instance, Ubuntu, Red Hat Linux and Amazon Linux 2.

<u>Solution:</u>

1) Login to the management console and go to the EC2 dashboard. Click on launch instances.



Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running)	0	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	0
Key pairs	3	Load balancers	0	Placement groups	0
Security groups	3	Snapshots	0	Volumes	0

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the Asia Pacific (Mumbai) Region

Service health

Region: Asia Pacific (Mumbai)

Status: This service is operating normally.

Instance alarms: 0 in alarm, 0 OK, 0 insufficient data

Zones

Zone name	Zone ID
ap-south-1a	aps1-az1
ap-south-1b	aps1-az3
ap-south-1c	aps1-az2

EC2 Free Tier Info

Offers for all AWS Regions.

2 EC2 free tier offers in use

End of month forecast: 0 offers forecasted to exceed free tier limit.

Exceeds free tier: 0 offers exceeded and is now pay-as-you-go pricing.

Offer usage (monthly)

Linux EC2 Instances 0% — 747.949166 hours remaining

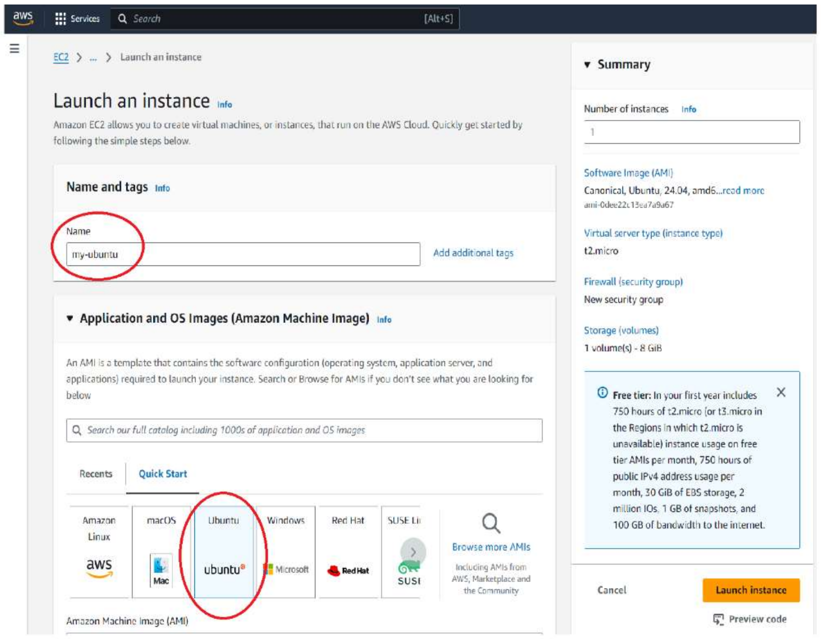
Storage space on EBS 0% — 29.98 GB remaining

Account attributes

Default VPC: vpc-0ed4fb8863de2b7ae

Settings

2) We need to launch 3 instances with different OS. So give name of your instance and choose the AMI. The first instance I have done is my-ubuntu as shown below.



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: my-ubuntu

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Summary

Number of instances Info: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more ami-0dee22c13ea7a9a67

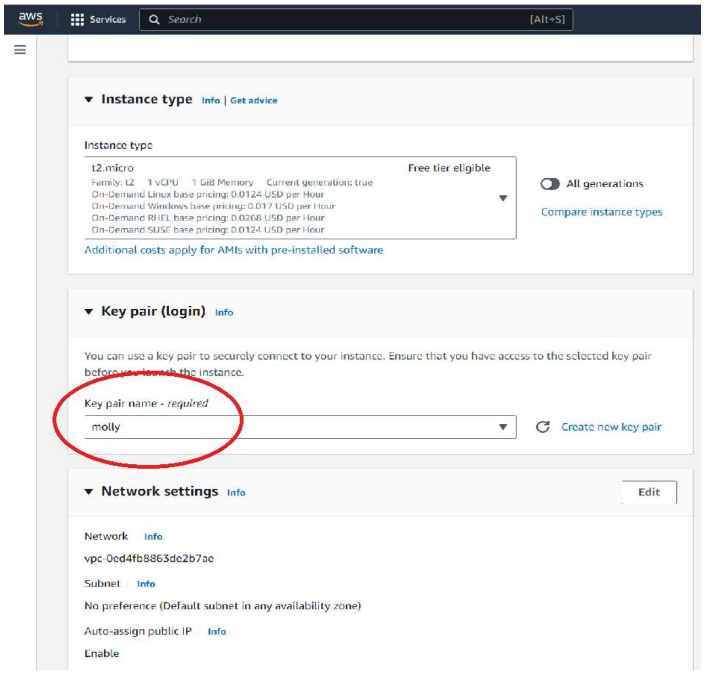
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

3) Choose the instance type=t2.micro (free tier eligible) , Choose the key pair login or you can create a new one. Choose default vpc or you can create your own. I have chosen t2.micro, molly.rsa key pair and default VPC.



Instance type Info | Get advice

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0268 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour

All generations
Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*
molly
Create new key pair

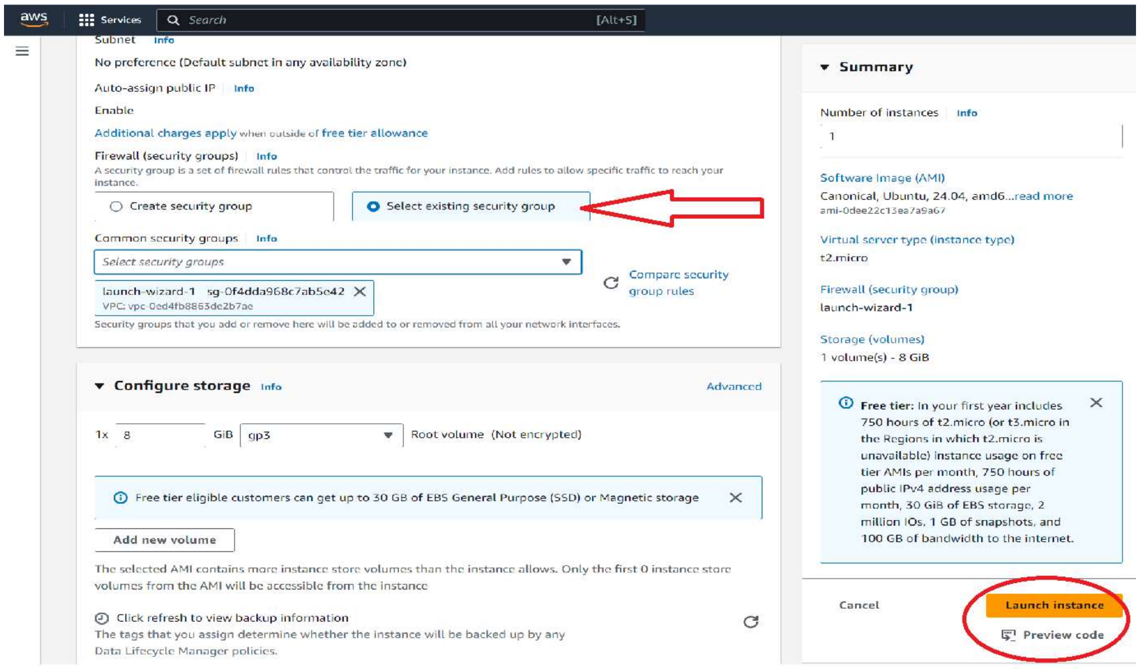
Network settings Info — Edit

Network Info
vpc-0ed4fb8863de2b7ae

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

4) I have chosen my existing security group for my-ubuntu. We can create a new one also . I went with the default storage. And then click launch instance. My ubuntu instance is created.



Subnet Info
No preference (Default subnet in any availability zone)

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]

Common security groups Info
Select security groups
launch-wizard-1 sg-0f4dda968c7ab5e42
VPC: vpc-0ed4fb8863de2b7ae
Compare security group rules
Security groups that you add or remove here will be added to or removed from all your network interfaces.

Configure storage Info — Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

Summary

Number of instances Info
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0dee22c13ea7a9a67

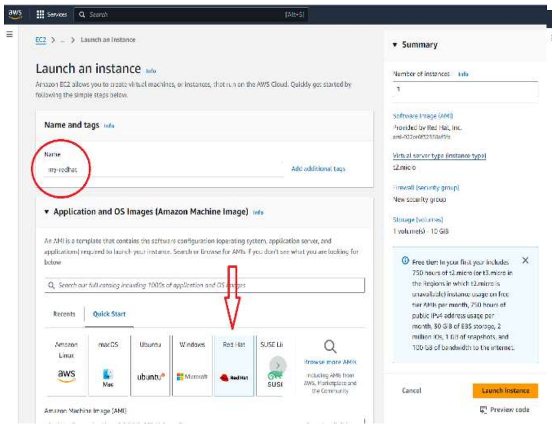
Virtual server type (instance type)
t2.micro

Firewall (security group)
launch-wizard-1

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

5) So I created 2 more instances with different os (Red hat linux and amazon linux 2) . The steps are same but only the name and AMI is different as shown below.



EC2 > ... > Launch an instance

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: my-redhat

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Quick Start: Red Hat

Summary

Number of instances: 1

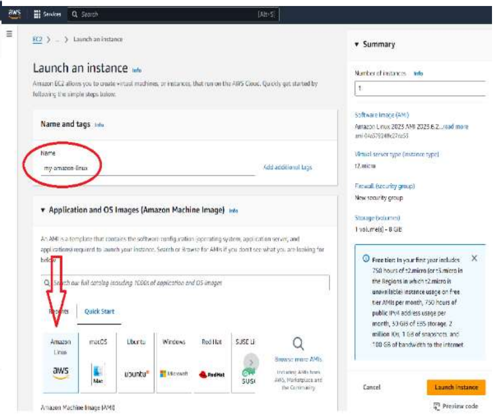
Software Image (AMI): Provided by Red Hat, Inc.

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 10 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.



EC2 > ... > Launch an instance

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: my-amazon-linux

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Quick Start: Amazon Linux

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.6.2...read more

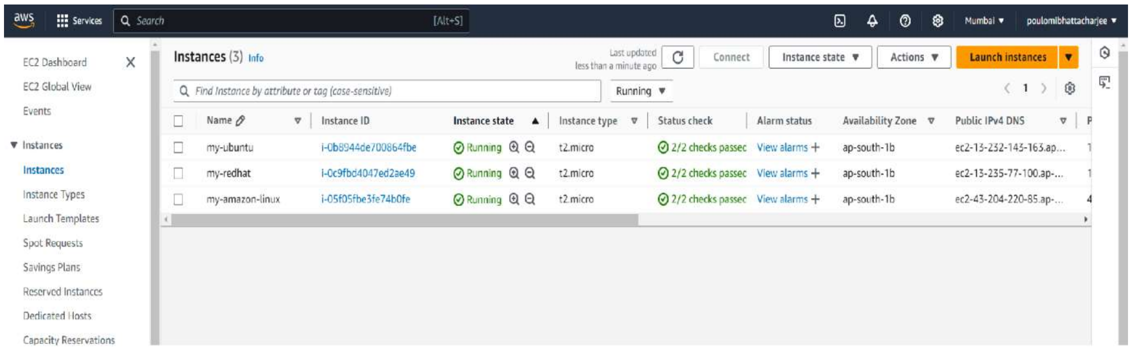
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet

6) So now my 3 instances are created and in running state .

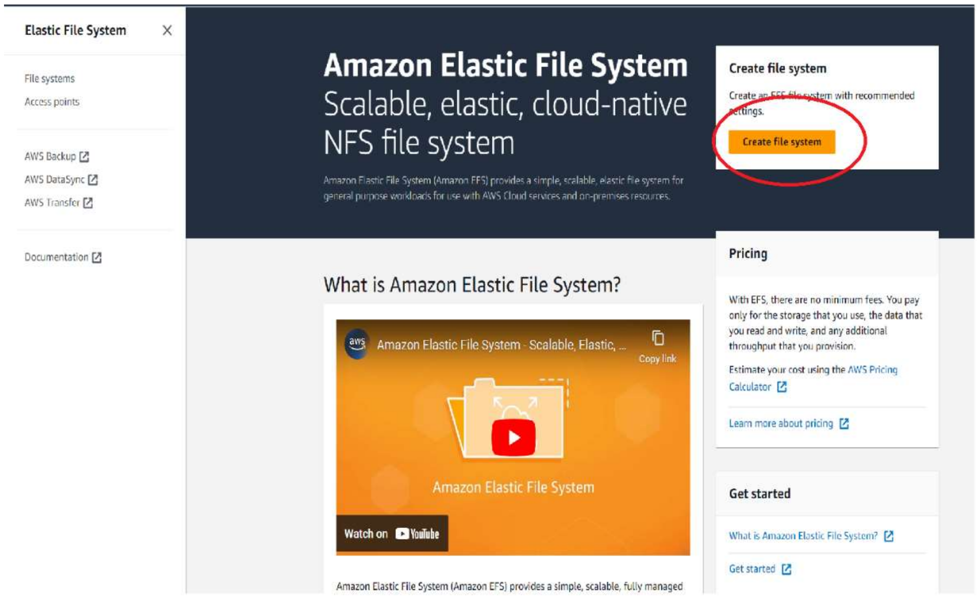


Instances (3) Info

Last updated less than a minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
my-ubuntu	i-0b8944de700864fbe	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-232-145-163.ap...
my-redhat	i-0c9fbd4047ed2ae49	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-235-77-100.ap-...
my-amazon-linux	i-05f05fbe3fe74b0fe	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-43-204-220-85.ap-...

7) Now go to EFS console and click create file system.



Amazon Elastic File System

Scalable, elastic, cloud-native NFS file system

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, elastic file system for general purpose workloads for use with AWS Cloud services and on-premises resources.

Create file system

Create an EFS file system with recommended settings.

Pricing

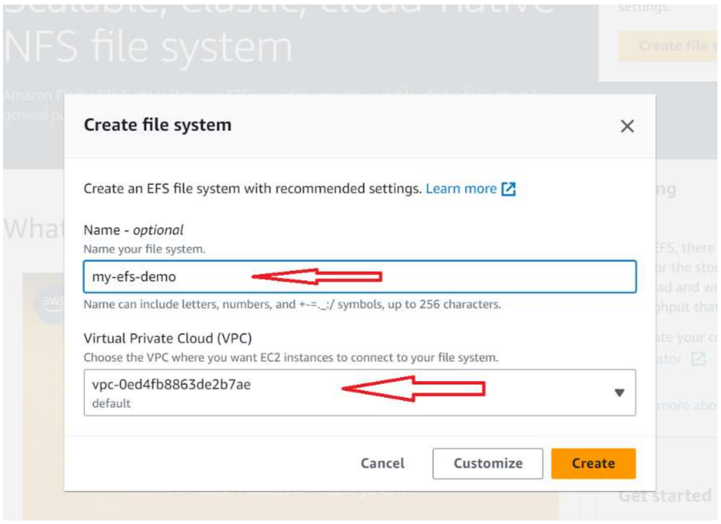
With EFS, there are no minimum fees. You pay only for the storage that you use, the data that you read and write, and any additional throughput that you provision.

Estimate your cost using the AWS Pricing Calculator

What is Amazon Elastic File System?

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed

8) Choose the name of your EFS and choose the VPC . Then click either create or you can customize it.



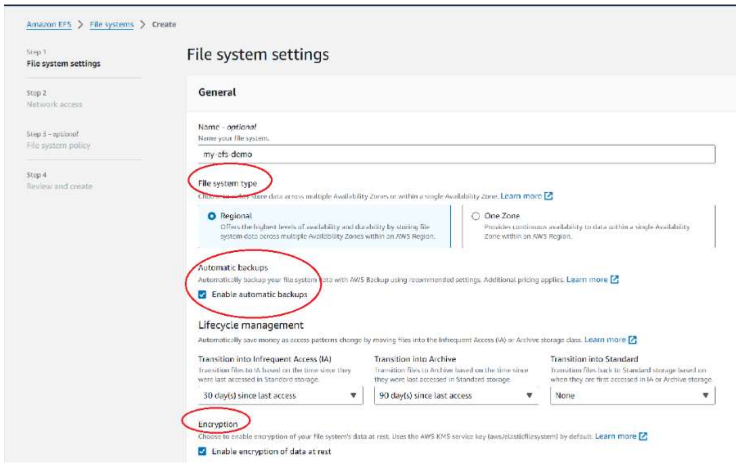
Create file system

Create an EFS file system with recommended settings. Learn more

Name - *optional*
Name your file system.
my-efs-demo
Name can include letters, numbers, and +-=._:/ symbols, up to 256 characters.

Virtual Private Cloud (VPC)
Choose the VPC where you want EC2 instances to connect to your file system.
vpc-0ed4fb8863de2b7ae
default

9) For customize we can choose file system type: whether it will be regional or one zone. We can enable/disable automatic backup as per our need. We can enable/disable encryption . we can decide lifecycle transitions.



File system settings

General

Name - *optional*
Name your file system.
my-efs-demo

File system type
Choose to either store data across multiple Availability Zones or within a single Availability Zone. Learn more

- [✓ Regional] Offers the highest levels of availability and durability by storing file system data across multiple Availability Zones within an AWS Region.
- One Zone: Provides continuous availability to data within a single Availability Zone within an AWS Region.

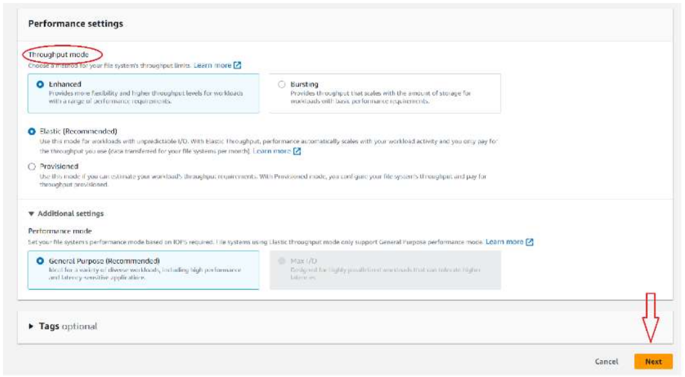
Automatic backups
Automatically backup your file system data with AWS Backup using recommended settings. Additional pricing applies. Learn more
☑ Enable automatic backups

Lifecycle management
Automatically save money as access patterns change by moving files into the Infrequent Access (IA) or Archive storage class. Learn more

Transition into Infrequent Access (IA)	Transition into Archive	Transition into Standard
Transition files to IA based on the time since they were last accessed in Standard storage.	Transition files to Archive based on the time since they were last accessed in Standard storage.	Transition files back to Standard storage based on when they are first accessed in IA or Archive storage.
30 day(s) since last access	90 day(s) since last access	None

Encryption
Choose to enable encryption of your file system's data at rest. Uses the AWS KMS service key (aws/elasticfilesystem) by default. Learn more
☑ Enable encryption of data at rest

10) Choose the throughput options enhanced / bursting as per your need. Click on next.



Performance settings

Throughput mode
Choose a method for your file system's throughput limits. Learn more

- [✓ Enhanced] Provides more flexibility and higher throughput levels for workloads with a range of performance requirements.
- Bursting: Provides throughput that scales with the amount of storage for workloads with basic performance requirements.

- [✓ Elastic (Recommended)] Use this mode for workloads with unpredictable I/O. With Elastic Throughput, performance automatically scales with your workload activity and you only pay for the throughput you use (data transferred for your file systems per month). Learn more
- Provisioned: Use this mode if you can estimate your workload's throughput requirements. With Provisioned mode, you configure your file system's throughput and pay for throughput provisioned.

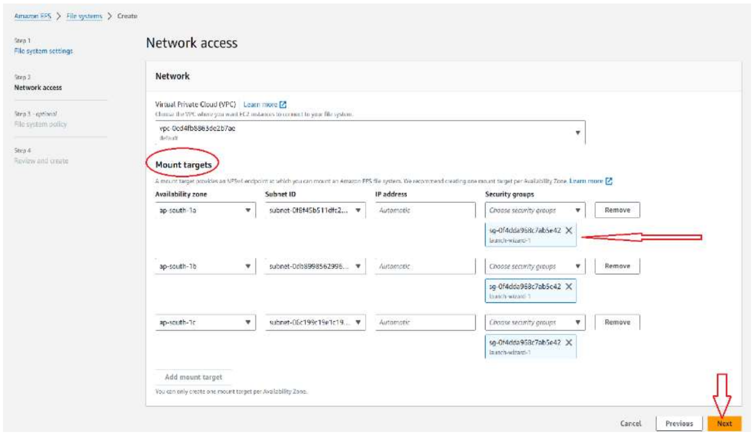
▼ Additional settings

Performance mode
Set your file system's performance mode based on IOPS required. File systems using Elastic throughput mode only support General Purpose performance mode. Learn more

- [✓ General Purpose (Recommended)] Ideal for a variety of diverse workloads, including high performance and latency-sensitive applications.
- Max I/O: Designed for highly parallelized workloads that can tolerate higher latencies.

11) Under network access we can choose the VPC where we want our EFS or go with the default one. A mount target in every AZ is created where we can mount our EFS. Here we can choose the security group associate with each mount target point. I have chosen the same security

group “ launch wizard 1” for all the mount targets. We can remove mount targets if we don’t need. Click next to proceed.


Amazon EFS > File systems > Create

Step 1 File system settings
Step 2 Network access
Step 3 - optional File system policy
Step 4 Review and create

Network access

Network

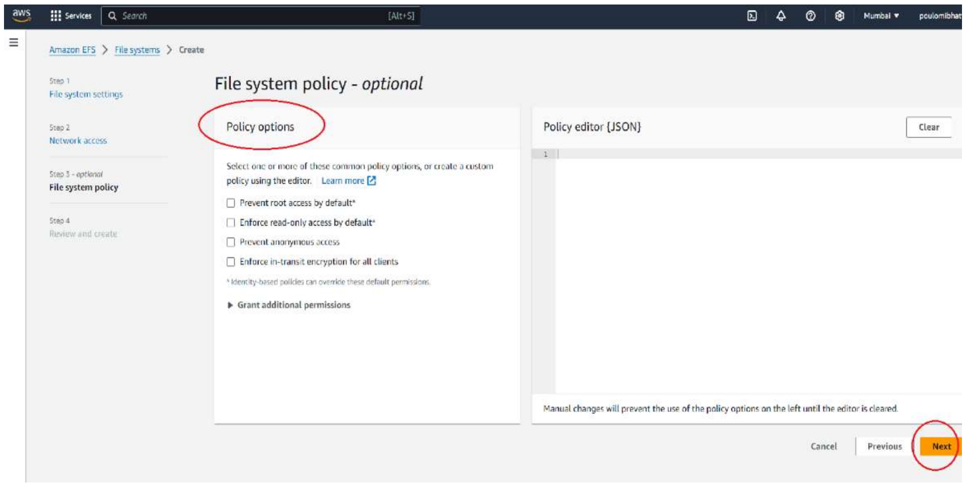
Virtual Private Cloud (VPC) Learn more
Choose the VPC where you want EC2 instances to connect to your file system.
vpc-0ed4fb8863de2b7ae
default

Mount targets
A mount target provides an NFSv4 endpoint at which you can mount an Amazon EFS file system. We recommend creating one mount target per Availability Zone. Learn more

Availability zone	Subnet ID	IP address	Security groups	
ap-south-1a	subnet-0f6f45b511dfc2...	Automatic	Choose security groups sg-0f4dda968c7ab5e42 launch-wizard-1	Remove
ap-south-1b	subnet-0db8998562996...	Automatic	Choose security groups sg-0f4dda968c7ab5e42 launch-wizard-1	Remove
ap-south-1c	subnet-06c199c19e1c19...	Automatic	Choose security groups sg-0f4dda968c7ab5e42 launch-wizard-1	Remove

Add mount target
You can only create one mount target per Availability Zone.

12) File system policy is optional . Here we can manage who can access and what permission is granted. Its like an extra security . I went wth default settings. click next.



Amazon EFS > File systems > Create

Step 1 File system settings
Step 2 Network access
Step 3 - optional File system policy
Step 4 Review and create

File system policy - *optional*

Policy options

Select one or more of these common policy options, or create a custom policy using the editor. Learn more

- [ ] Prevent root access by default*
- [ ] Enforce read-only access by default*
- [ ] Prevent anonymous access
- [ ] Enforce in-transit encryption for all clients

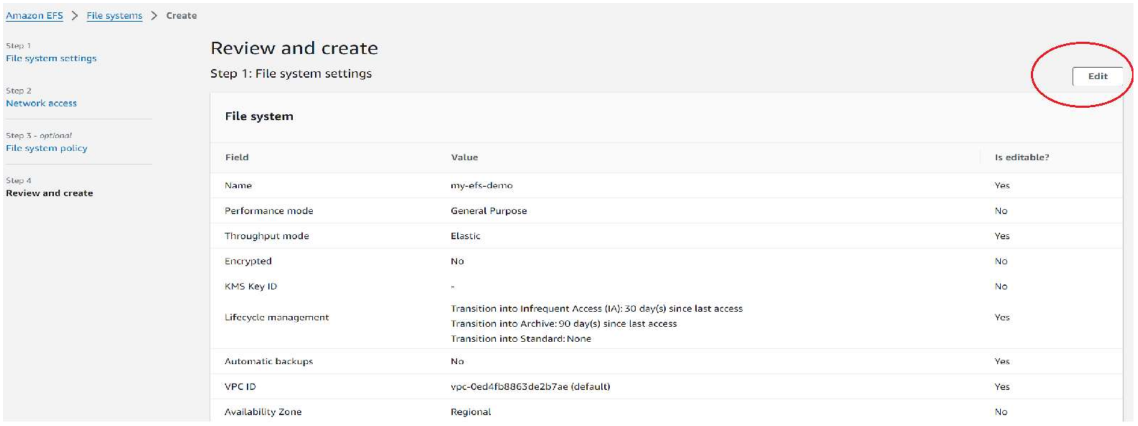
* Identity-based policies can override these default permissions.

▶ Grant additional permissions

Policy editor (JSON)

Manual changes will prevent the use of the policy options on the left until the editor is cleared.

13) Finally we can review and if any changes we can edit it. Click create.


Amazon EFS > File systems > Create

Step 1 File system settings
Step 2 Network access
Step 3 - optional File system policy
Step 4 Review and create

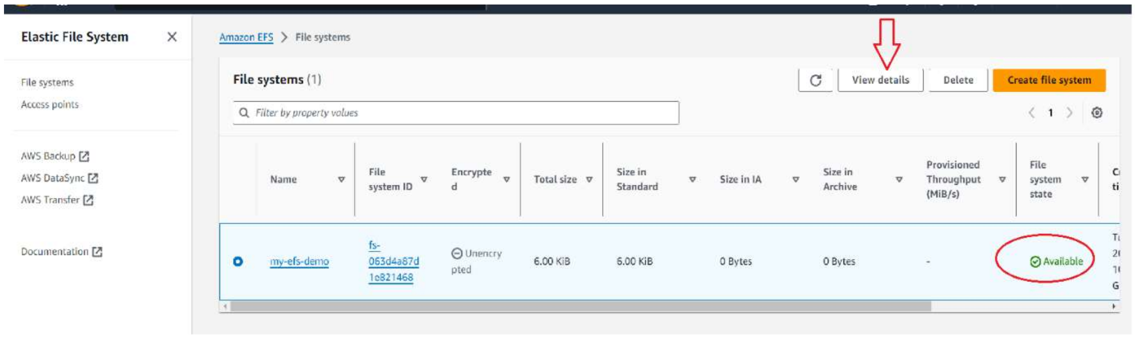
Review and create

Step 1: File system settings

File system

Field	Value	Is editable?
Name	my-efs-demo	Yes
Performance mode	General Purpose	No
Throughput mode	Elastic	Yes
Encrypted	No	No
KMS Key ID	-	No
Lifecycle management	Transition into Infrequent Access (IA): 30 day(s) since last access Transition into Archive: 90 day(s) since last access Transition into Standard: None	Yes
Automatic backups	No	Yes
VPC ID	vpc-0ed4fb8863de2b7ae (default)	Yes
Availability Zone	Regional	No

14) So my EFS is created and available. Click on view details to attach EFS with instances.



Elastic File System

File systems
Access points

AWS Backup
AWS DataSync
AWS Transfer

Documentation

Amazon EFS > File systems

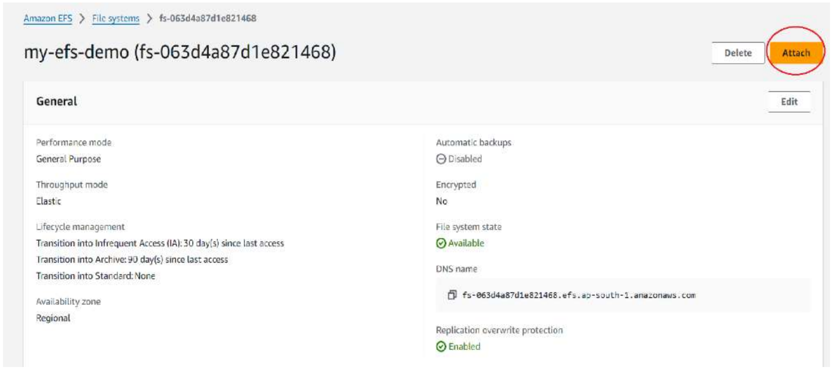
File systems (1)

View details | Delete | Create file system

Filter by property values

	Name	File system ID	Encrypted	Total size	Size in Standard	Size in IA	Size in Archive	Provisioned Throughput (MiB/s)	File system state
●	my-efs-demo	fs-063d4a87d1e821468	Unencrypted	6.00 KiB	6.00 KiB	0 Bytes	0 Bytes	-	Available

15) We can view all the details and there is attach option. Click on attach.



Amazon EFS > File systems > fs-063d4a87d1e821468

my-efs-demo (fs-063d4a87d1e821468)

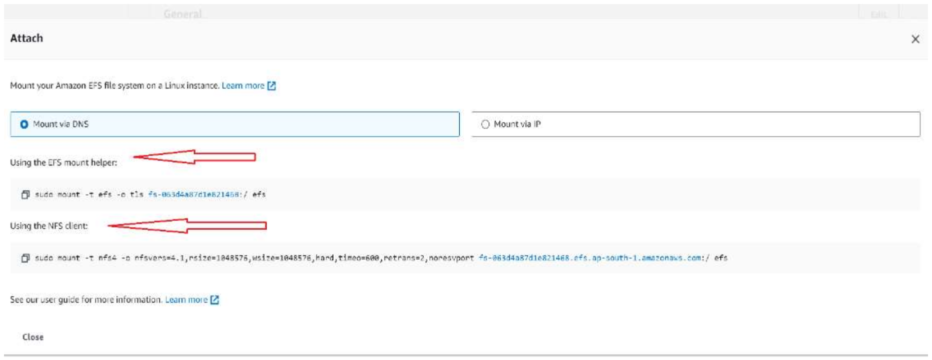
Delete | Attach

General

Edit

Performance mode: General Purpose	Automatic backups: Disabled
Throughput mode: Elastic	Encrypted: No
Lifecycle management: Transition into Infrequent Access (IA): 30 day(s) since last access; Transition into Archive: 90 day(s) since last access; Transition into Standard: None	File system state: Available
Availability zone: Regional	DNS name: fs-063d4a87d1e821468.efs.ap-south-1.amazonaws.com
	Replication overwrite protection: Enabled

16) So we can mount our EFS using nfs client or Efs mount helper.



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-063d4a87d1e821468:/ efs
```

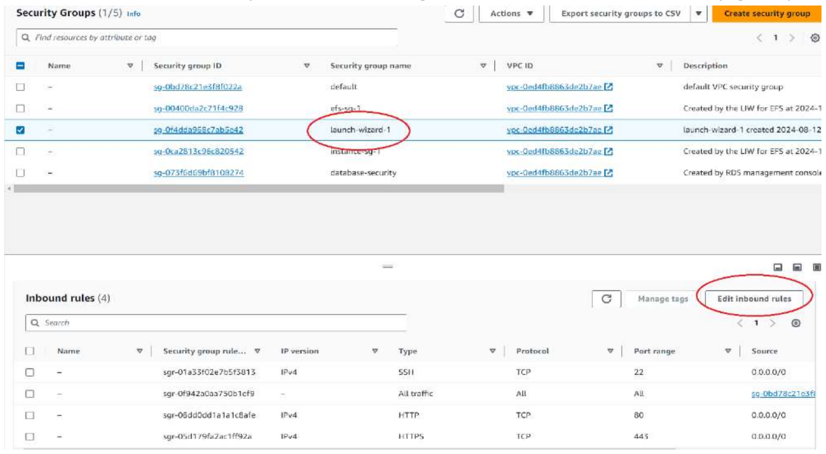
Using the NFS client:

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-063d4a87d1e821468.efs.ap-south-1.amazonaws.com:/ efs
```

See our user guide for more information. Learn more

Close

17) Before mounting we need to go to security group of the mount target and in inbound rule we have to allow NFS protocol coming from our instance security group.



Security Groups (1/5) Info

Actions | Export security groups to CSV | Create security group

Find resources by attribute or tag

	Name	Security group ID	Security group name	VPC ID	Description
☐	–	sg-0bd78c21e3f8f022a	default	vpc-0ed4fb8863de2b7ae	default VPC security group
☐	–	sg-00400da2c71f4c928	efs-sg-1	vpc-0ed4fb8863de2b7ae	Created by the LIW for EFS at 2024-1
☑	–	sg-0f4dda968c7ab5c42	launch-wizard-1	vpc-0ed4fb8863de2b7ae	launch-wizard-1 created 2024-08-12
☐	–	sg-0ca2813c96c820542	instance-sg-1	vpc-0ed4fb8863de2b7ae	Created by the LIW for EFS at 2024-1
☐	–	sg-073f6d69bf8108274	database-security	vpc-0ed4fb8863de2b7ae	Created by RDS management console

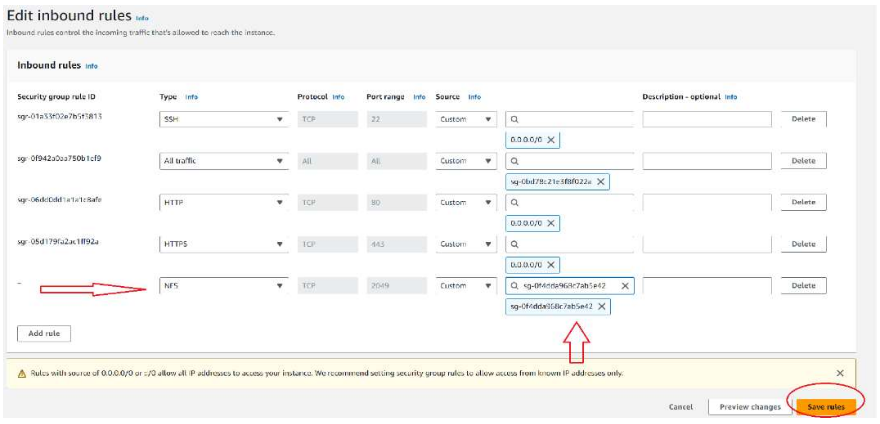
Inbound rules (4)

Manage tags | Edit inbound rules

Search

	Name	Security group rule...	IP version	Type	Protocol	Port range	Source
☐	–	sgr-01a33f02e7b5f3813	IPv4	SSH	TCP	22	0.0.0.0/0
☐	–	sgr-0f942a0aa750b1cf9	–	All traffic	All	All	sg-0bd78c21e3f
☐	–	sgr-06dd0dd1a1a1c8afe	IPv4	HTTP	TCP	80	0.0.0.0/0
☐	–	sgr-05d179fa2ac1ff92a	IPv4	HTTPS	TCP	443	0.0.0.0/0

18) Click save rule. Inbound security group rules successfully modified on security group (sg-0f4dda968c7ab5e42 | launch-wizard-1)



Edit inbound rules

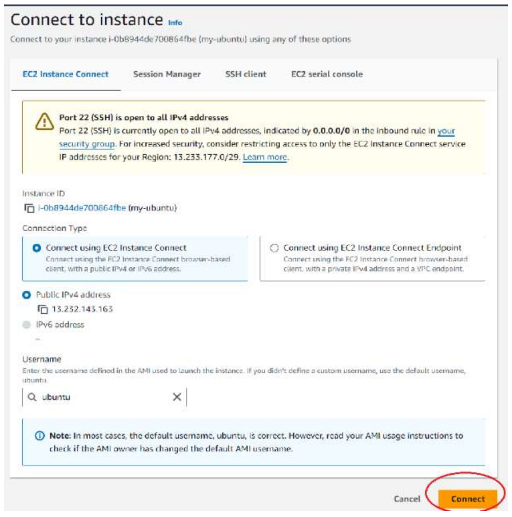
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-01a33f02e7b5f3813	SSH	TCP	22	Custom	0.0.0.0/0		Delete
sgr-0f942a0aa750b1cf9	All traffic	All	All	Custom	sg-0bd78c21e3f8f022a		Delete
sgr-06dd0dd1a1a1c8afe	HTTP	TCP	80	Custom	0.0.0.0/0		Delete
sgr-05d179fa2ac1ff92a	HTTPS	TCP	443	Custom	0.0.0.0/0		Delete
–	NFS	TCP	2049	Custom	sg-0f4dda968c7ab5e42		Delete

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

19) Connect all the 3 instances one by one. I have connected using instance connect.



Connect to instance

Connect to your instance i-0b8944de700864fbe (my-ubuntu) using any of these options

Port 22 (SSH) is open to all IPv4 addresses
Port 22 (SSH) is currently open to all IPv4 addresses, indicated by **0.0.0.0/0** in the inbound rule in your security group. For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 13.233.177.0/29. Learn more.

Instance ID
i-0b8944de700864fbe (my-ubuntu)

Connection Type
[✓ Connect using EC2 Instance Connect] Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.
Connect using EC2 Instance Connect Endpoint — Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

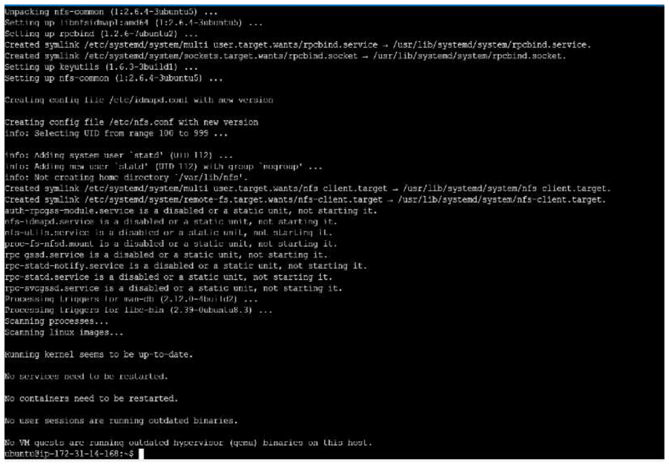
[✓ Public IPv4 address] 13.232.143.163
IPv6 address –

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.
ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

20) For ubuntu :

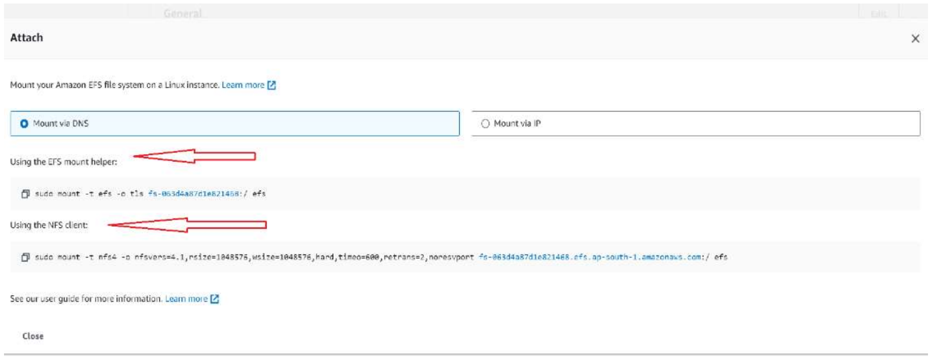
a) After connecting we have to download nfs client. If you're using an Ubuntu Amazon EC2 AMI, install the NFS client with the following command.
 sudo apt-get -y install nfs-common.



```
Unpacking nfs-common (1:2.6.4-3ubuntu5) ...
Setting up libnfsidmap1:amd64 (1:2.6.4-3ubuntu5) ...
Setting up rpcbind (1.2.6-7ubuntu2) ...
Created symlink /etc/systemd/system/multi user.target.wants/rpcbind.service → /usr/lib/systemd/system/rpcbind.service.
Created symlink /etc/systemd/system/sockets.target.wants/rpcbind.socket → /usr/lib/systemd/system/rpcbind.socket.
Setting up keyutils (1.6.3-3build1) ...
Setting up nfs-common (1:2.6.4-3ubuntu5) ...

Creating config file /etc/idmapd.conf with new version

Creating config file /etc/nfs.conf with new version
info: Selecting UID from range 100 to 999 ...

info: Adding system user `statd' (UID 112) ...
info: Adding new user `statd' (UID 112) with group `nogroup' ...
info: Not creating home directory `/var/lib/nfs'.
Created symlink /etc/systemd/system/multi user.target.wants/nfs client.target → /usr/lib/systemd/system/nfs client.target.
Created symlink /etc/systemd/system/remote-fs.target.wants/nfs-client.target → /usr/lib/systemd/system/nfs-client.target.
auth-rpcgss-module.service is a disabled or a static unit, not starting it.
nfs-idmapd.service is a disabled or a static unit, not starting it.
nfs-utils.service is a disabled or a static unit, not starting it.
proc-fs-nfsd.mount is a disabled or a static unit, not starting it.
rpc gssd.service is a disabled or a static unit, not starting it.
rpc-statd-notify.service is a disabled or a static unit, not starting it.
rpc-statd.service is a disabled or a static unit, not starting it.
rpc-svcgssd.service is a disabled or a static unit, not starting it.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-14-168:~$
```

b) Then make a directory where we need to mount our efs.
   Sudo mkdir /efsdemo   *** efsdemo : name of the mount point

c) Now using nfs client we can attach the efs.



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-063d4a87d1e821468:/ efs
```

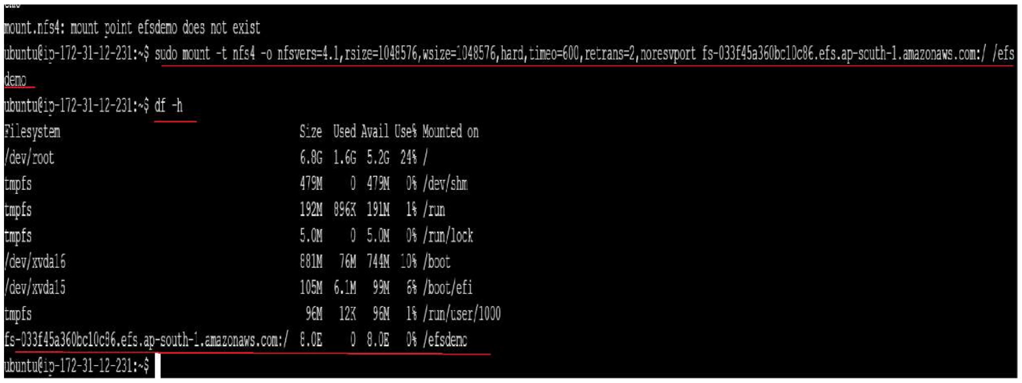
Using the NFS client:

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-063d4a87d1e821468.efs.ap-south-1.amazonaws.com:/ efs
```

See our user guide for more information. Learn more

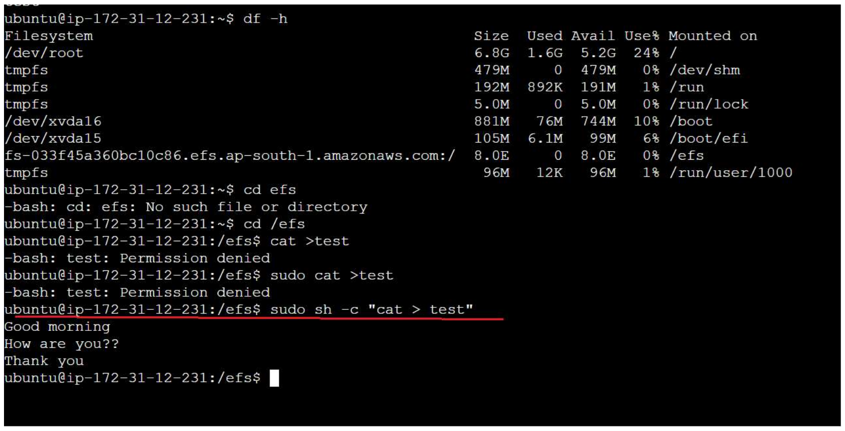
Close

d) While using the nfs client command change the name of the default mountpoint with the name we have chosen. For me: efsdemo.
   To verify whether efs is mounted use df -h command and we can see that it is mounted.



```
mount.nfs4: mount point efsdemo does not exist
ubuntu@ip-172-31-12-231:~$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-033f45a360bc10c86.efs.ap-south-1.amazonaws.com:/ /efs
demo
ubuntu@ip-172-31-12-231:~$ df -h
Filesystem                                          Size  Used Avail Use% Mounted on
/dev/root                                           6.8G  1.6G  5.2G  24% /
tmpfs                                               479M     0  479M   0% /dev/shm
tmpfs                                               192M  896K  191M   1% /run
tmpfs                                               5.0M     0  5.0M   0% /run/lock
/dev/xvda16                                         881M   76M  744M  10% /boot
/dev/xvda15                                         105M  6.1M   99M   6% /boot/efi
tmpfs                                                96M   12K   96M   1% /run/user/1000
fs-033f45a360bc10c86.efs.ap-south-1.amazonaws.com:/  8.0E     0  8.0E   0% /efsdemo
ubuntu@ip-172-31-12-231:~$
```

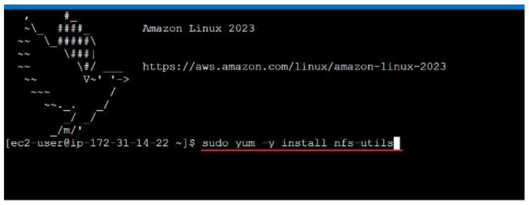
e) Now create a file and put some contents in it. I have created a test file using cat command.



```
ubuntu@ip-172-31-12-231:~$ df -h
Filesystem                                          Size  Used Avail Use% Mounted on
/dev/root                                           6.8G  1.6G  5.2G  24% /
tmpfs                                               479M     0  479M   0% /dev/shm
tmpfs                                               192M  892K  191M   1% /run
tmpfs                                               5.0M     0  5.0M   0% /run/lock
/dev/xvda16                                         881M   76M  744M  10% /boot
/dev/xvda15                                         105M  6.1M   99M   6% /boot/efi
fs-033f45a360bc10c86.efs.ap-south-1.amazonaws.com:/  8.0E     0  8.0E   0% /efs
tmpfs                                                96M   12K   96M   1% /run/user/1000
ubuntu@ip-172-31-12-231:~$ cd efs
-bash: cd: efs: No such file or directory
ubuntu@ip-172-31-12-231:~$ cd /efs
ubuntu@ip-172-31-12-231:/efs$ cat >test
-bash: test: Permission denied
ubuntu@ip-172-31-12-231:/efs$ sudo cat >test
-bash: test: Permission denied
ubuntu@ip-172-31-12-231:/efs$ sudo sh -c "cat > test"
Good morning
How are you??
Thank you
ubuntu@ip-172-31-12-231:/efs$
```

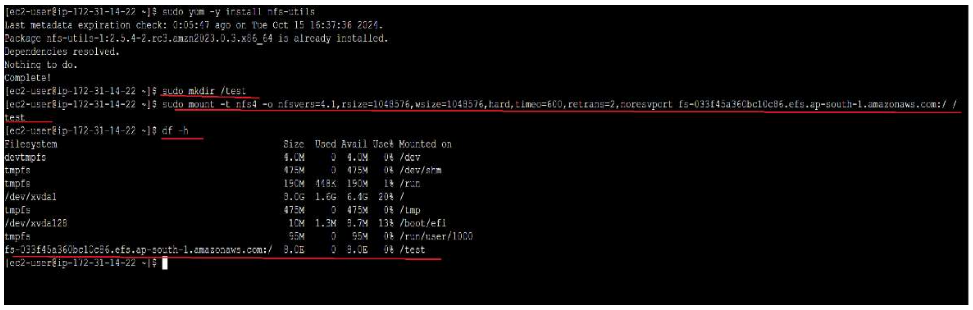
21) For Amazon linux 2 :

a) After connecting we have to download nfs client. If you're using Amazon linux 2 EC2 AMI, install the NFS client with the following command.
   sudo yum -y install nfs-utils.



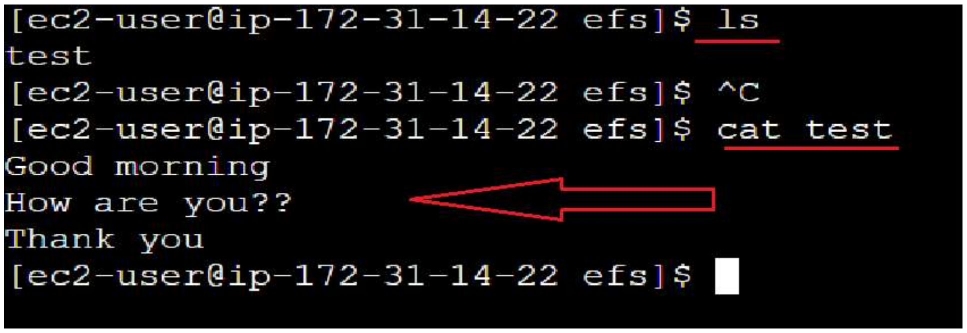
```
       ,     #_
       ~\_  ####_        Amazon Linux 2023
      ~~  \_#####\
      ~~     \###|
      ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
       ~~       V~' '->
        ~~~         /
          ~~._.   _/
             _/ _/
           _/m/'
[ec2-user@ip-172-31-14-22 ~]$ sudo yum -y install nfs-utils
```

b) Same steps we have to follow as we have done in ubuntu. Make a directory with a different name and mount efs using the nfs client. To verify use df -h command.



```
[ec2-user@ip-172-31-14-22 ~]$ sudo yum -y install nfs-utils
Last metadata expiration check: 0:05:47 ago on Tue Oct 15 16:37:36 2024.
Package nfs-utils-1:2.5.4-2.rc3.amzn2023.0.3.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-14-22 ~]$ sudo mkdir /test
[ec2-user@ip-172-31-14-22 ~]$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-033f45a360bc10c86.efs.ap-south-1.amazonaws.com:/ /
test
[ec2-user@ip-172-31-14-22 ~]$ df -h
Filesystem                                          Size  Used Avail Use% Mounted on
devtmpfs                                            4.0M     0  4.0M   0% /dev
tmpfs                                               475M     0  475M   0% /dev/shm
tmpfs                                               190M  448K  190M   1% /run
/dev/xvda1                                          8.0G  1.6G  6.4G  20% /
tmpfs                                               475M     0  475M   0% /tmp
/dev/xvda128                                         10M  1.3M  8.7M  13% /boot/efi
tmpfs                                                95M     0   95M   0% /run/user/1000
fs-033f45a360bc10c86.efs.ap-south-1.amazonaws.com:/  8.0E     0  8.0E   0% /test
[ec2-user@ip-172-31-14-22 ~]$
```

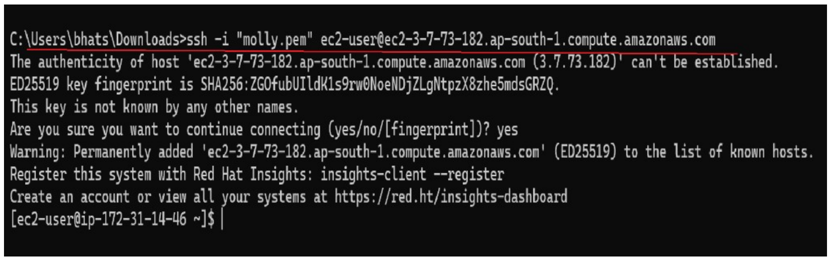
c) We can access the test file we created using ubuntu ec2 from amazon linux ec2. This shows file can be shared easily.



```
[ec2-user@ip-172-31-14-22 efs]$ ls
test
[ec2-user@ip-172-31-14-22 efs]$ ^C
[ec2-user@ip-172-31-14-22 efs]$ cat test
Good morning
How are you??
Thank you
[ec2-user@ip-172-31-14-22 efs]$
```

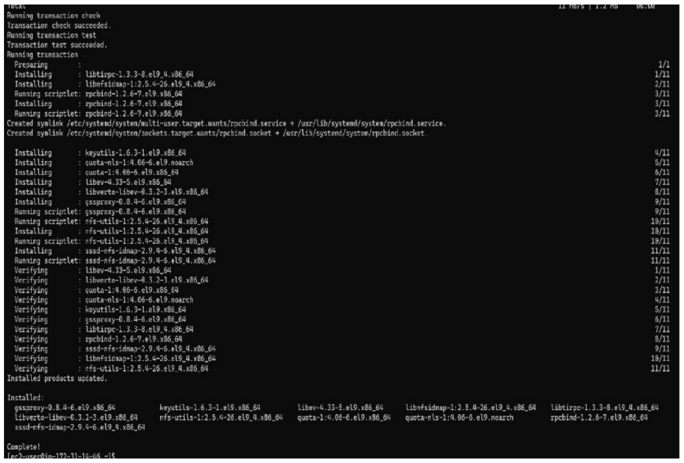
22) For Red hat linux :

a) we have to connect Red hat linux ec2 using ssh client. Open your terminal and locate the location of the private key. Paste the ssh client command associated with your red hat ec2.



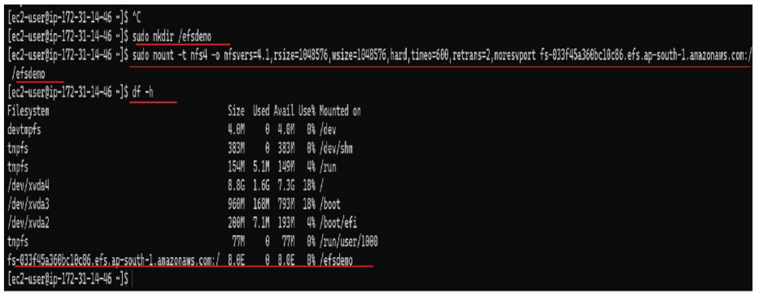
```
C:\Users\bhats\Downloads>ssh -i "molly.pem" ec2-user@ec2-3-7-73-182.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-3-7-73-182.ap-south-1.compute.amazonaws.com (3.7.73.182)' can't be established.
ED25519 key fingerprint is SHA256:ZGOfubUIldK1s9rw0NoeNDjZLgNtpzX8zhe5mdsGRZQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-7-73-182.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-172-31-14-46 ~]$ |
```

b) After connecting we have to download nfs client. If you're using Red hat linux EC2 AMI, install the NFS client with the following command.
sudo yum -y install nfs-utils.



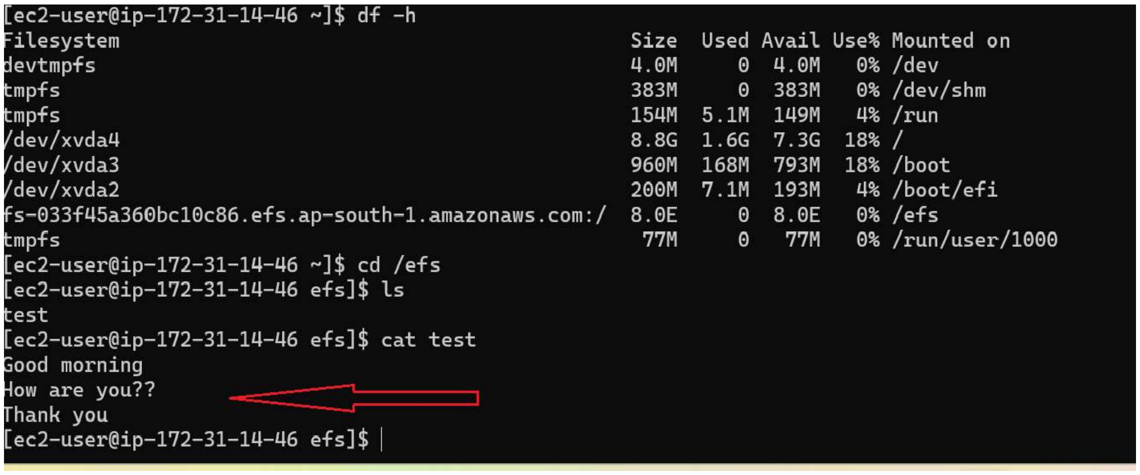
```
Total                                                                                          11 MB/s | 1.2 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                  1/1
  Installing       : libtirpc-1.3.3-8.el9_4.x86_64                                                                   1/11
  Installing       : libnfsidmap-1:2.5.4-26.el9_4.x86_64                                                             2/11
  Running scriptlet: rpcbind-1.2.6-7.el9.x86_64                                                                      3/11
  Installing       : rpcbind-1.2.6-7.el9.x86_64                                                                      3/11
  Running scriptlet: rpcbind-1.2.6-7.el9.x86_64                                                                      3/11
Created symlink /etc/systemd/system/multi-user.target.wants/rpcbind.service → /usr/lib/systemd/system/rpcbind.service.
Created symlink /etc/systemd/system/sockets.target.wants/rpcbind.socket → /usr/lib/systemd/system/rpcbind.socket.

  Installing       : keyutils-1.6.3-1.el9.x86_64                                                                     4/11
  Installing       : quota-nls-1:4.06-6.el9.noarch                                                                   5/11
  Installing       : quota-1:4.06-6.el9.x86_64                                                                       6/11
  Installing       : libev-4.33-5.el9.x86_64                                                                         7/11
  Installing       : libverto-libev-0.3.2-3.el9.x86_64                                                               8/11
  Installing       : gssproxy-0.8.4-6.el9.x86_64                                                                     9/11
  Running scriptlet: gssproxy-0.8.4-6.el9.x86_64                                                                     9/11
  Running scriptlet: nfs-utils-1:2.5.4-26.el9_4.x86_64                                                              10/11
  Installing       : nfs-utils-1:2.5.4-26.el9_4.x86_64                                                              10/11
  Running scriptlet: nfs-utils-1:2.5.4-26.el9_4.x86_64                                                              10/11
  Installing       : sssd-nfs-idmap-2.9.4-6.el9_4.x86_64                                                            11/11
  Running scriptlet: sssd-nfs-idmap-2.9.4-6.el9_4.x86_64                                                            11/11
  Verifying        : libev-4.33-5.el9.x86_64                                                                         1/11
  Verifying        : libverto-libev-0.3.2-3.el9.x86_64                                                               2/11
  Verifying        : quota-1:4.06-6.el9.x86_64                                                                       3/11
  Verifying        : quota-nls-1:4.06-6.el9.noarch                                                                   4/11
  Verifying        : keyutils-1.6.3-1.el9.x86_64                                                                     5/11
  Verifying        : gssproxy-0.8.4-6.el9.x86_64                                                                     6/11
  Verifying        : libtirpc-1.3.3-8.el9_4.x86_64                                                                   7/11
  Verifying        : rpcbind-1.2.6-7.el9.x86_64                                                                      8/11
  Verifying        : sssd-nfs-idmap-2.9.4-6.el9_4.x86_64                                                             9/11
  Verifying        : libnfsidmap-1:2.5.4-26.el9_4.x86_64                                                            10/11
  Verifying        : nfs-utils-1:2.5.4-26.el9_4.x86_64                                                              11/11
Installed products updated.

Installed:
  gssproxy-0.8.4-6.el9.x86_64          keyutils-1.6.3-1.el9.x86_64          libev-4.33-5.el9.x86_64     libnfsidmap-1:2.5.4-26.el9_4.x86_64   libtirpc-1.3.3-8.el9_4.x86_64
  libverto-libev-0.3.2-3.el9.x86_64    nfs-utils-1:2.5.4-26.el9_4.x86_64    quota-1:4.06-6.el9.x86_64   quota-nls-1:4.06-6.el9.noarch         rpcbind-1.2.6-7.el9.x86_64
  sssd-nfs-idmap-2.9.4-6.el9_4.x86_64

Complete!
[ec2-user@ip-172-31-14-46 ~]$
```

c) Same steps we have to follow as we have done in ubuntu. Make a directory and mount efs using the nfs client. To verify use df -h command.



```
[ec2-user@ip-172-31-14-46 ~]$ ^C
[ec2-user@ip-172-31-14-46 ~]$ sudo mkdir /efsdemo
[ec2-user@ip-172-31-14-46 ~]$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-033f45a360bc10c86.efs.ap-south-1.amazonaws.com:/
 /efsdemo
[ec2-user@ip-172-31-14-46 ~]$ df -h
Filesystem                                         Size  Used Avail Use% Mounted on
devtmpfs                                           4.0M     0  4.0M   0% /dev
tmpfs                                              383M     0  383M   0% /dev/shm
tmpfs                                              154M  5.1M  149M   4% /run
/dev/xvda4                                         8.8G  1.6G  7.3G  18% /
/dev/xvda3                                         960M  168M  793M  18% /boot
/dev/xvda2                                         200M  7.1M  193M   4% /boot/efi
tmpfs                                               77M     0   77M   0% /run/user/1000
fs-033f45a360bc10c86.efs.ap-south-1.amazonaws.com:/  8.0E     0  8.0E   0% /efsdemo
[ec2-user@ip-172-31-14-46 ~]$
```

d) We can access the test file we created using ubuntu ec2 from Red hat linux. This shows file can be shared easily among all the three instances of different os.



```
[ec2-user@ip-172-31-14-46 ~]$ df -h
Filesystem                                           Size  Used Avail Use% Mounted on
devtmpfs                                             4.0M     0  4.0M   0% /dev
tmpfs                                                383M     0  383M   0% /dev/shm
tmpfs                                                154M  5.1M  149M   4% /run
/dev/xvda4                                           8.8G  1.6G  7.3G  18% /
/dev/xvda3                                           960M  168M  793M  18% /boot
/dev/xvda2                                           200M  7.1M  193M   4% /boot/efi
fs-033f45a360bc10c86.efs.ap-south-1.amazonaws.com:/  8.0E     0  8.0E   0% /efs
tmpfs                                                 77M     0   77M   0% /run/user/1000
[ec2-user@ip-172-31-14-46 ~]$ cd /efs
[ec2-user@ip-172-31-14-46 efs]$ ls
test
[ec2-user@ip-172-31-14-46 efs]$ cat test
Good morning
How are you??
Thank you
[ec2-user@ip-172-31-14-46 efs]$
```

So finally all my ec2 instances can share data using EFS.