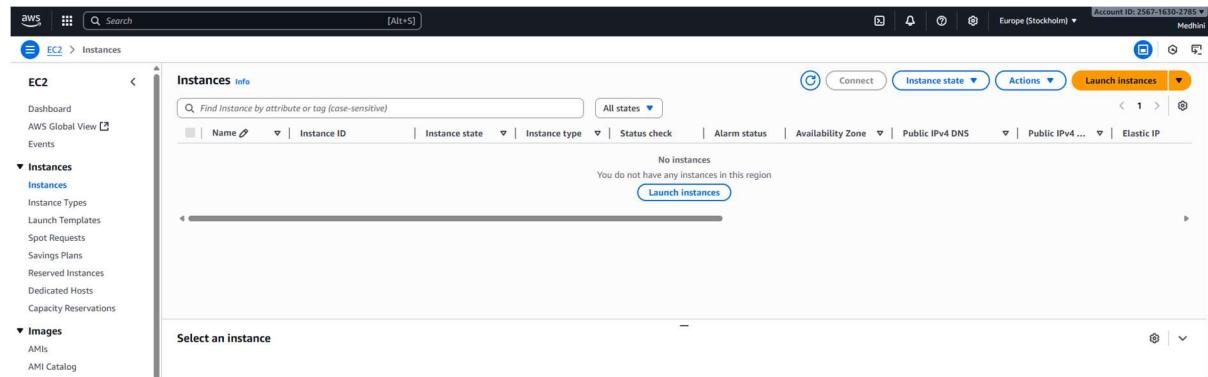
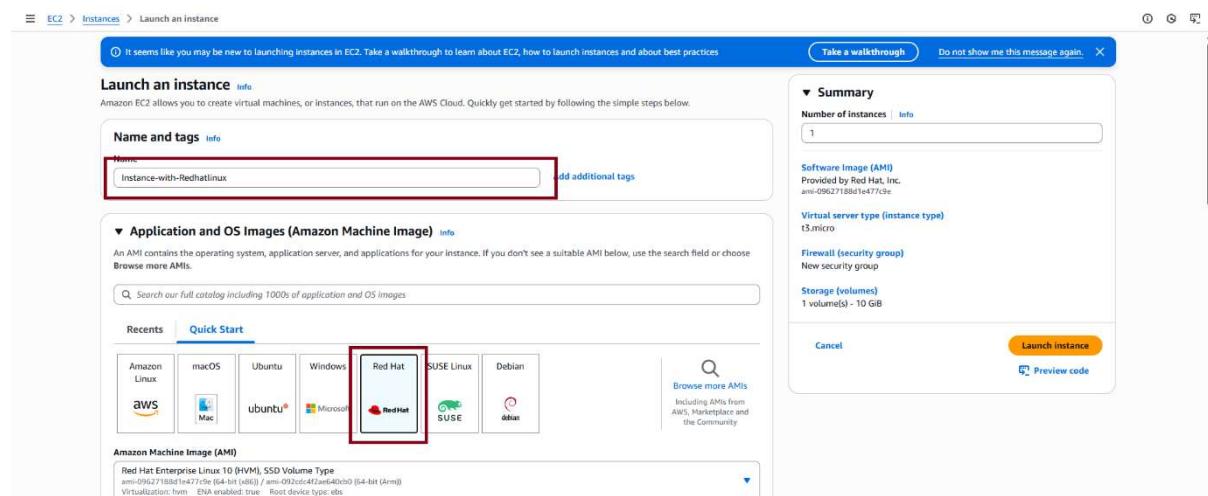


# EC2 Instance with Red Hat Linux Operating System

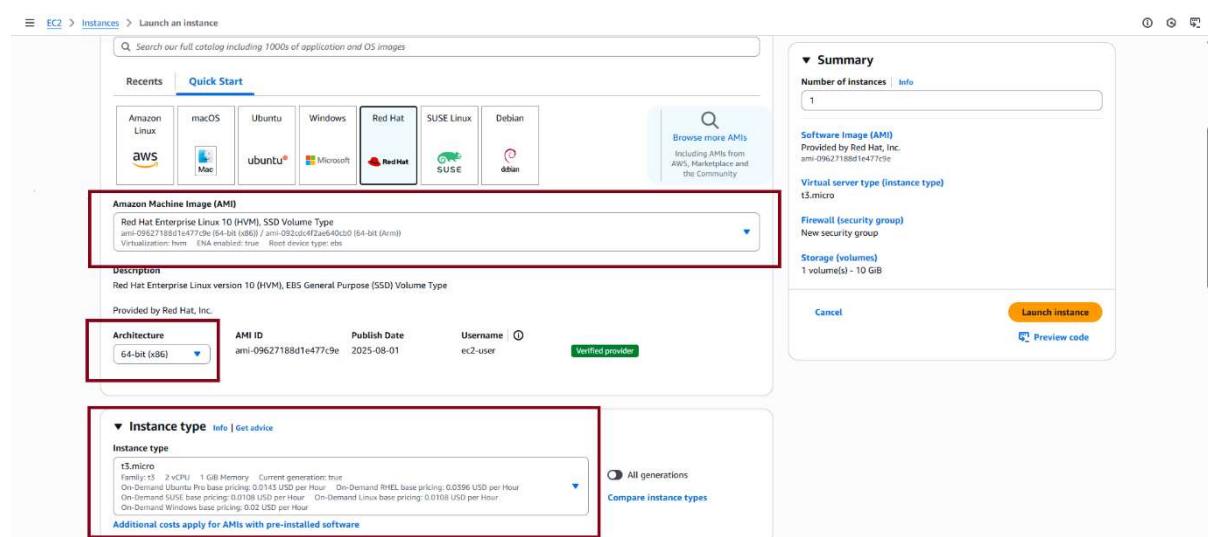
Step 1: Login to your account, click on Instances



Step 2: Give a name for an instance and select Operating system as RED Hat

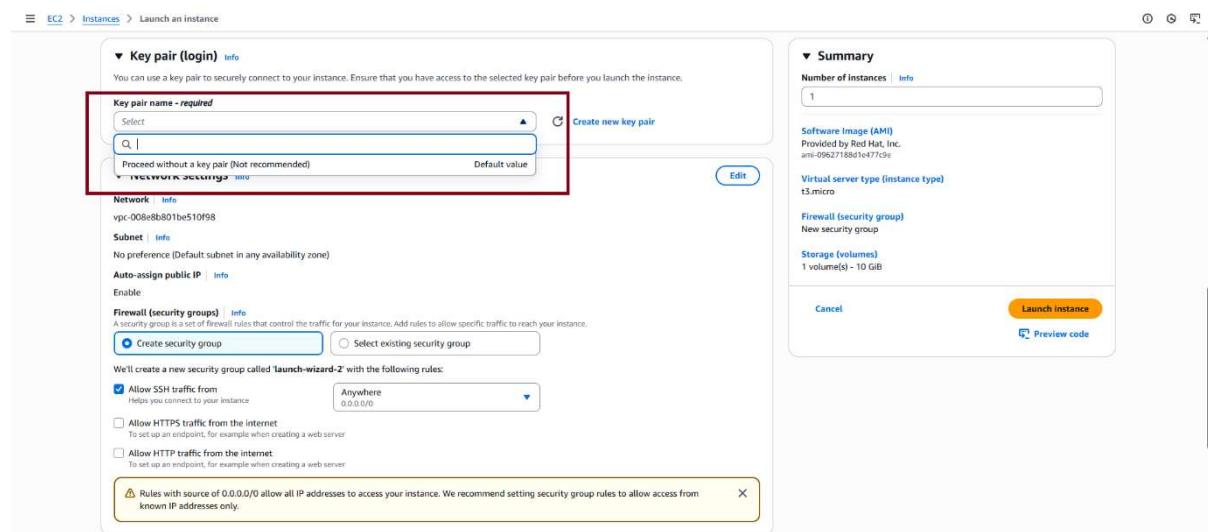


Step 3: Select AMI, choose storage type and instance type as shown in below picture



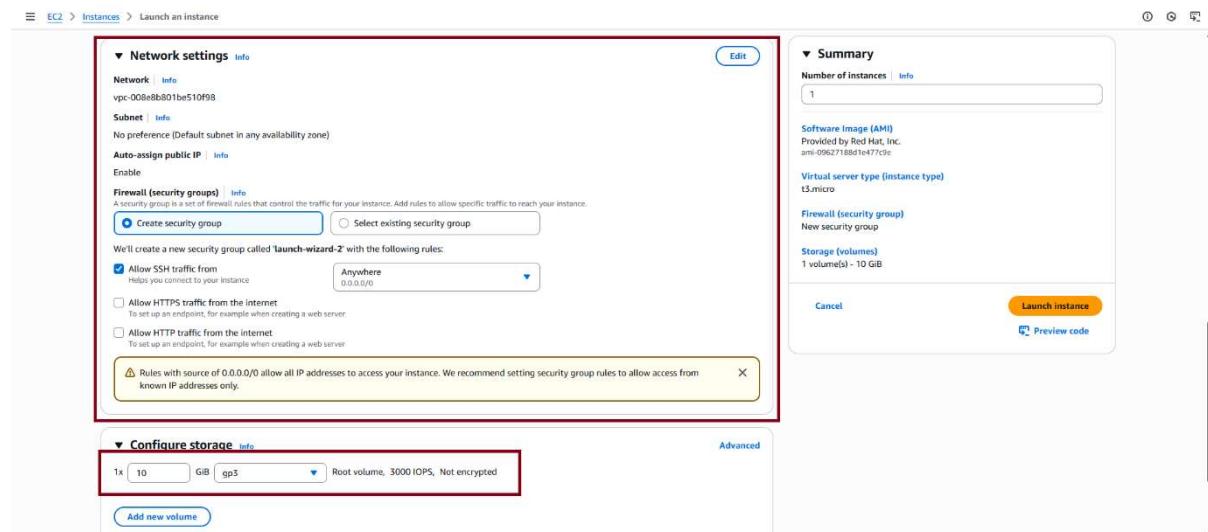
# Assignment 1 – EC2 AND EFS

## Step 4: Select Keypair as shown in below picture



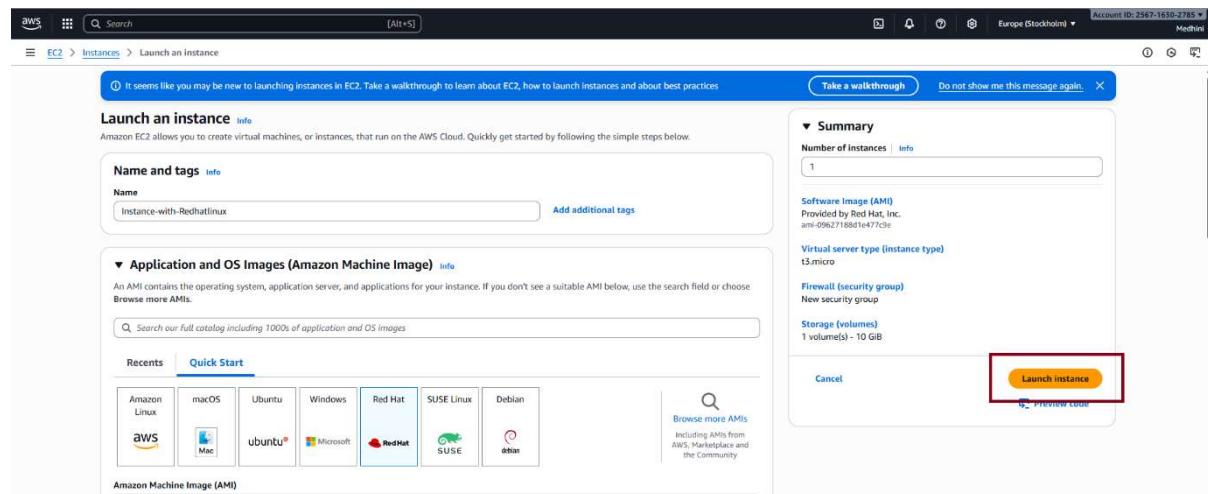
The screenshot shows the 'Launch an instance' wizard, Step 4: Set Key Pair. The 'Key pair name - required' input field is highlighted with a red box. The field contains 'Select' and a dropdown arrow. To its right is a 'Create new key pair' button. Below this section are 'Network settings' and 'Firewall (security groups)' sections.

## Step 5: Select Network Settings and configure storage as shown in below figure



The screenshot shows the 'Launch an instance' wizard, Step 5: Configure Storage. The 'Configure storage' section is highlighted with a red box. It shows a dropdown menu set to '1x 10 GB gp3' and a note: 'Root volume, 3000 IOPS, Not encrypted'. Below this is an 'Add new volume' button. To the right is a 'Summary' section showing instance details like Software Image (AMI), Virtual server type (t3.micro), and Storage (1 volume(s) - 10 GiB).

## Step 6: Click on Launch Instance as shown in picture



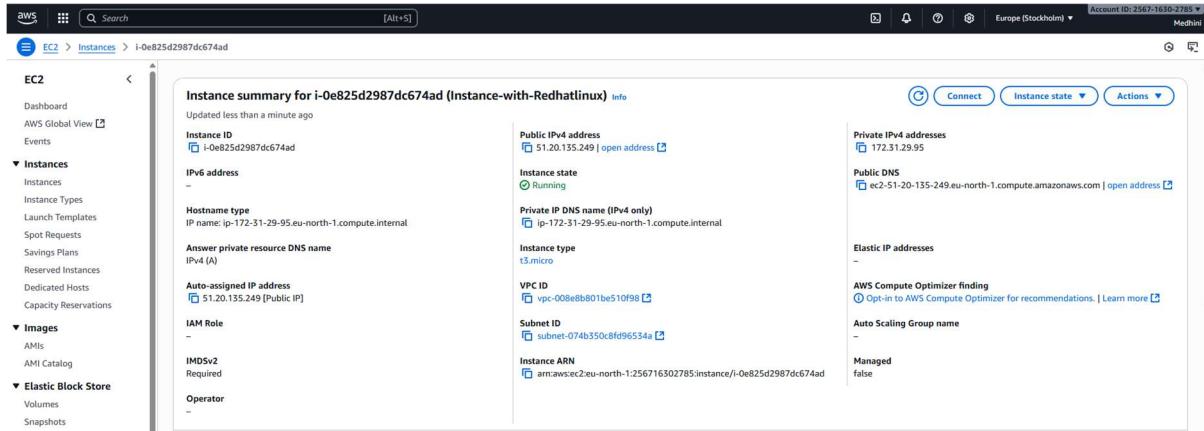
The screenshot shows the 'Launch an instance' wizard, Step 6: Final Review. The 'Launch instance' button is highlighted with a red box. The summary section on the right shows the selected instance type (t3.micro), AMI (Red Hat Enterprise Linux 10 (HVM)), and storage (1 volume(s) - 10 GiB). A message at the top left says, 'It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices.'

## Assignment 1 – EC2 AND EFS

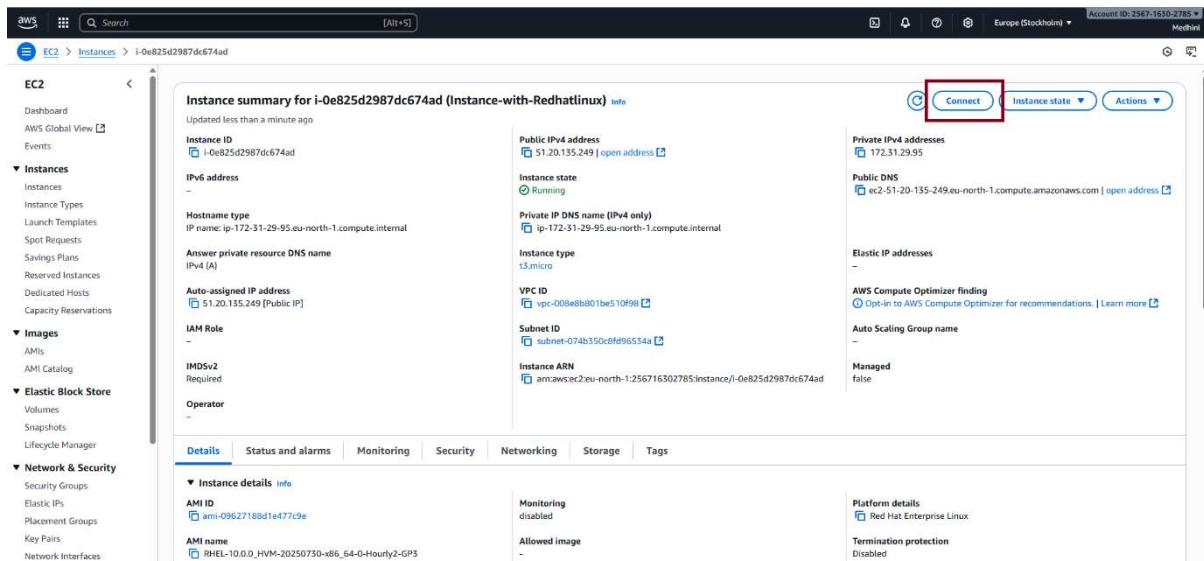
Step 7: You will see this Green bar where your instance is created and click on the marked area



Step 8: The below picture shows output of EC2 instance created with Redhat operating system

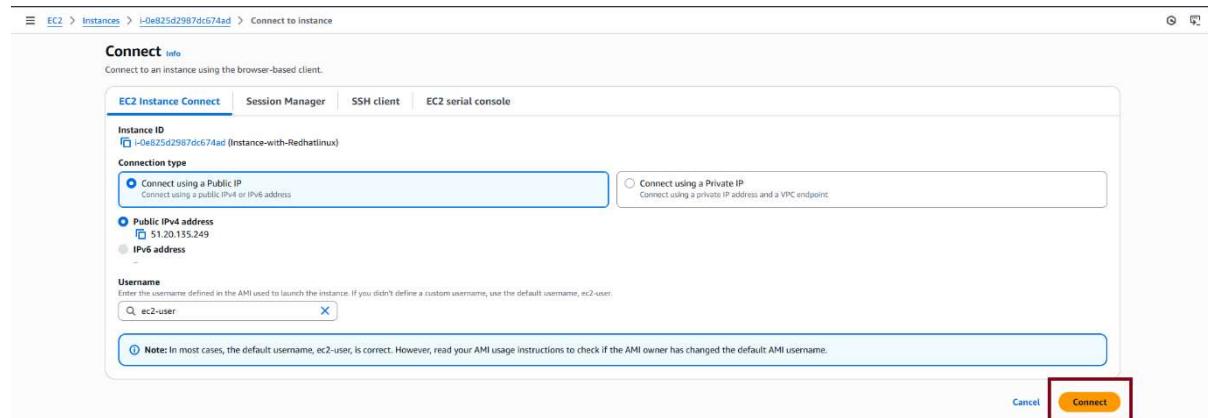


Step 9: Click on Connect as shown in below figure

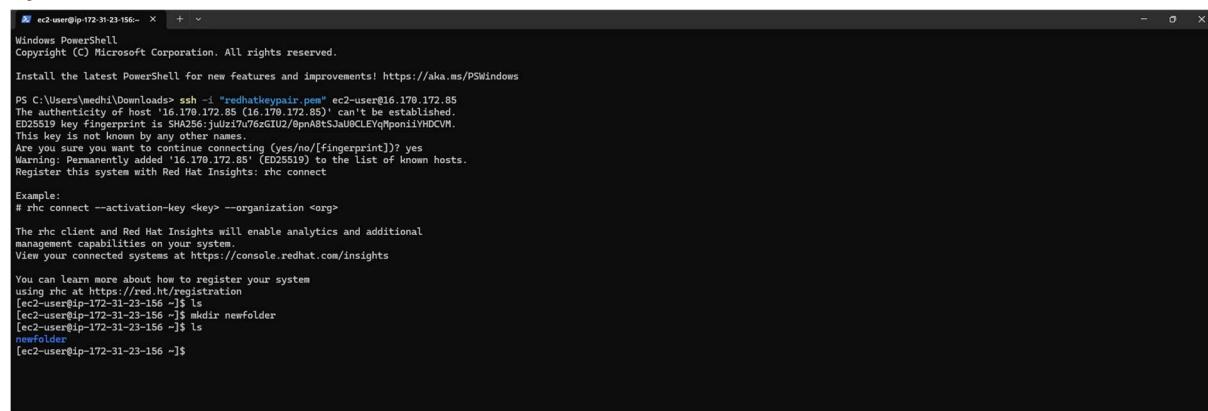


## Assignment 1 – EC2 AND EFS

Step 10: Click on connect it will redirect to new tab

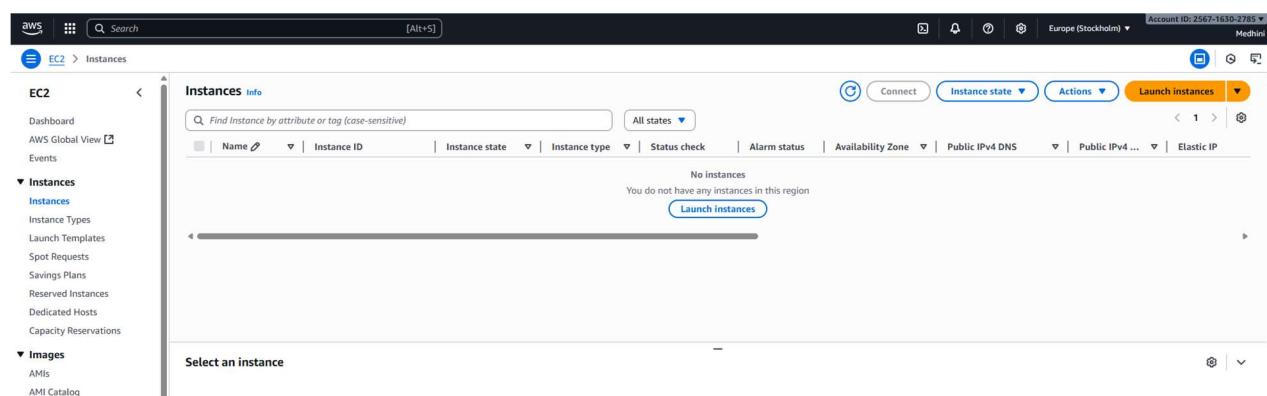


Step 11: The below picture shows the EC2 instance running on REDHAT linux operating system



## EC2 Instance with AWS Linux Operating System

Step 1: Login to your account, click on **Instances**



Step 2: Give a name for an instance

# Assignment 1 – EC2 AND EFS

The screenshot shows the 'Launch an Instance' wizard on the AWS EC2 console. The current step is 'Name and tags'. A text input field contains 'Instance-with-Linux'. To the right, there's a 'Add additional tags' button and a 'Summary' section.

**Summary**  
Number of instances: 1  
Software Image (AMI): Amazon Linux 2023 AMI 2023.9.2...  
Virtual server type (instance type): t3.micro  
Firewall (security group): New security group  
Storage (volumes): 1 volume(s) - 8 GiB

**Launch instance** | **Preview code**

## Step 3: Select Operating systems and storage under Architecture

The screenshot shows the 'Launch an Instance' wizard on the AWS EC2 console. The current step is 'Architecture'. It shows the selection of 'Amazon Linux' and 'Amazon Linux 2023 kernel-6.1 AMI'. Below this, there's a detailed description of the selected AMI and a table showing architecture details.

**Description**  
Amazon Linux 2023 kernel-6.1 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Architecture	Boot mode	AMI ID	Publish Date	Username	Verified provider
64-bit (x86)	uefi-preferred	ami-04c08fd8aa14af291	2025-09-25	ec2-user	

**Launch instance** | **Preview code**

## Step 4: Select instance type and Keypair

The screenshot shows the 'Launch an Instance' wizard on the AWS EC2 console. The current steps are 'Instance type' and 'Key pair (login)'. In the 'Instance type' step, 't3.micro' is selected. In the 'Key pair (login)' step, a dropdown menu shows 'Select' and a search bar with 'Q |'. There are also buttons for 'Create new key pair' and 'Edit'.

**Summary**  
Number of instances: 1  
Software Image (AMI): Amazon Linux 2023 AMI 2023.9.2...  
Virtual server type (instance type): t3.micro  
Firewall (security group): New security group  
Storage (volumes): 1 volume(s) - 8 GiB

**Launch instance** | **Preview code**

## Step 5: Select Network settings

# Assignment 1 – EC2 AND EFS

The screenshot shows the 'Network settings' step of the EC2 instance launch wizard. It includes fields for VPC (vpc-008e8bb801be510f98), Subnet (No preference), Auto-assign public IP (Enabled), and Firewall (security groups) (Create new security group). A note says 'We'll create a new security group called "launch-wizard-1" with the following rules:' followed by a dropdown menu showing 'Allow SSH traffic from Anywhere (0.0.0.0/0)'.

## Step 6: Configure storage

The screenshot shows the 'Configure storage' step of the EC2 instance launch wizard. It displays a single 8 GiB gp3 root volume. Below it are buttons for 'Add new volume' and 'Click refresh to view backup information'.

## Step 7: Click on Launch Instance

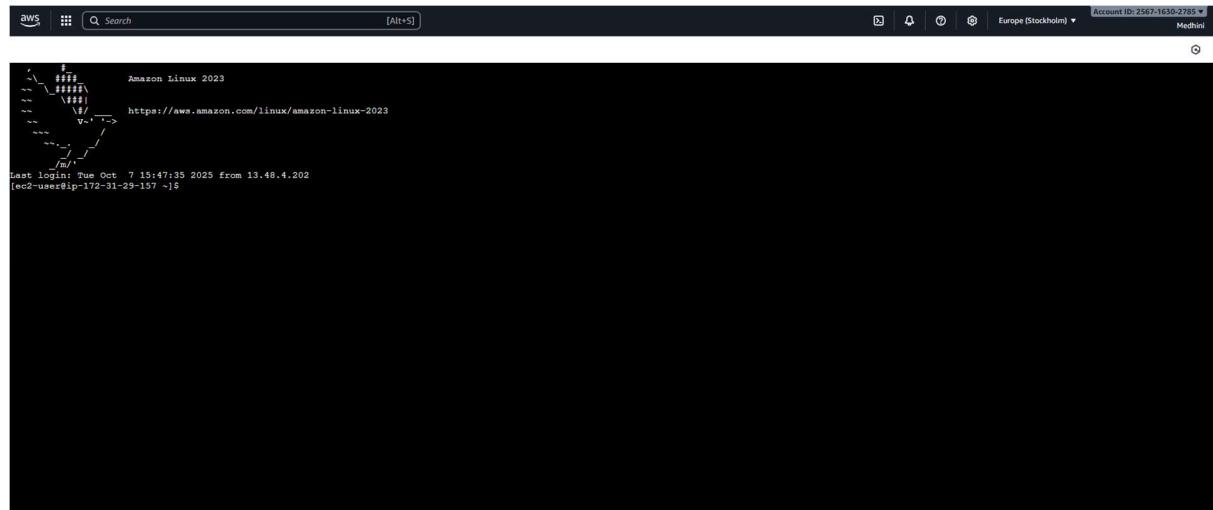
The screenshot shows the 'Summary' step of the EC2 instance launch wizard. It lists the Software Image (AMI) as Amazon Linux 2023.9.2, Virtual server type as t3.micro, and Storage (volumes) as 1 volume(s) - 8 GiB. The 'Launch instance' button is prominently displayed.

## Step 8: The Instance overview and Click on connect it will redirect to new tab as shown in figure below

The screenshot shows the EC2 Instances Overview page for the instance i-07f52c72cb8d50dea. It provides a detailed summary including Public IP (51.21.222.162), Instance ID (i-07f52c72cb8d50dea), Instance state (Running), and various identifiers like VPC ID, Subnet ID, and Instance ARN. On the right side, there are buttons for 'Connect', 'Instance state', and 'Actions'.

## Assignment 1 – EC2 AND EFS

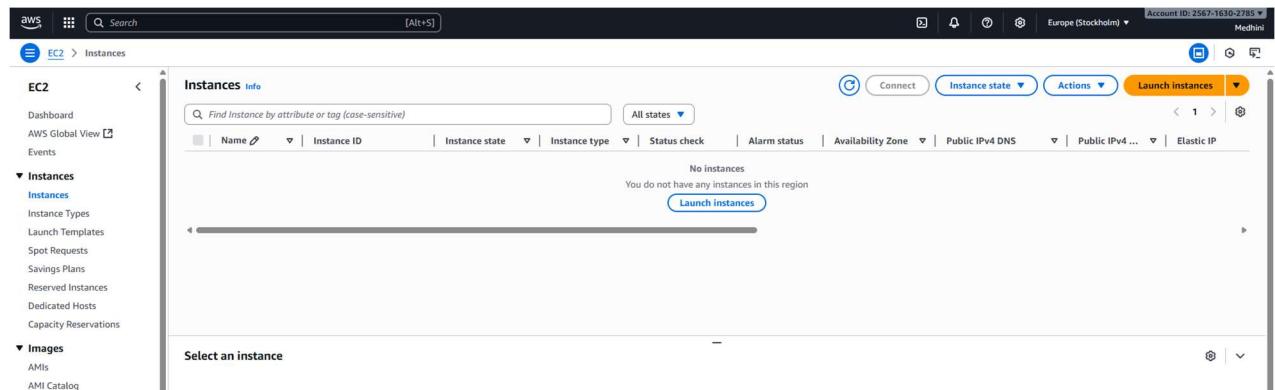
Step 9: The instance is running in AWS Linux is shown in below picture



```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
Last Logon: Tue Oct  7 15:47:35 2025 from 13.48.4.202
[ec2-user@ip-172-31-29-157 ~]$
```

## EC2 Instance with Ubuntu Operating System

Step 1: Login to your account, click on **Instances**



## Assignment 1 – EC2 AND EFS

Step 2: Give a name for an instance and choose operating system as Ubuntu as shown in below picture

The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' step, the instance name is set to 'Instance-with-Ubuntu'. The 'Application and OS Images (Amazon Machine Image)' section is expanded, showing various AMI categories like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. Under the 'Ubuntu' category, the 'Ubuntu Server 24.04 LTS (HVM), SSD Volume Type' AMI is selected. The 'Virtual server type (instance type)' is set to 't3.micro'. The 'Storage (volumes)' section indicates 1 volume(s) - 8 GiB. At the bottom right, there is a 'Launch instance' button.

Step 3: Choose AMI, Storage under Architecture and Instance type

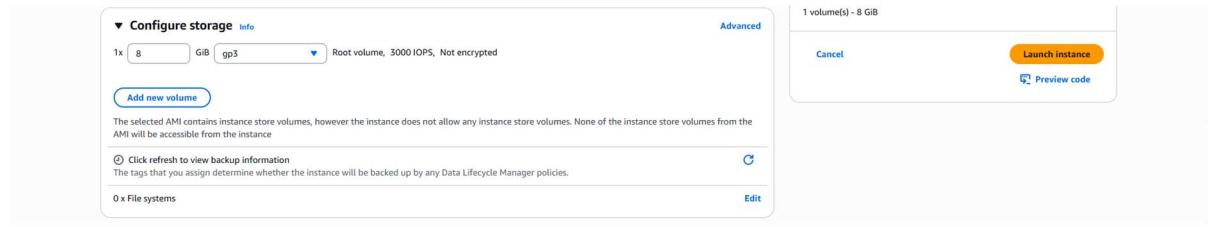
The screenshot shows the 'Amazon Machine Image (AMI)' and 'Instance type' steps of the wizard. In the 'Amazon Machine Image (AMI)' step, the 'Ubuntu Server 24.04 LTS (HVM), SSD Volume Type' AMI is selected. In the 'Instance type' step, the 't3.micro' instance type is chosen. Both steps show the same configuration: 1 volume(s) - 8 GiB, New security group, and a 'Launch instance' button at the bottom right.

Step 4: Select Keypair and Network settings

The screenshot shows the 'Key pair (login)' and 'Network settings' steps. In the 'Key pair (login)' step, it is noted that no key pair is selected. In the 'Network settings' step, the 'Network' section shows 'vpc-008e8bb801be510f98'. The 'Subnet' section indicates 'No preference (Default subnet in any availability zone)'. The 'Auto-assign public IP' section is set to 'Enabled'. The 'Firewall (security groups)' section shows a new security group named 'launch-wizard-3' being created. It includes rules for 'Allow SSH traffic from Anywhere' and 'Allow HTTP traffic from the internet'. A note at the bottom states: '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.' Both steps show the same configuration: 1 volume(s) - 8 GiB, New security group, and a 'Launch instance' button at the bottom right.

Step 5: Configure storage and click on **Lunch Instance**

## Assignment 1 – EC2 AND EFS



Step 6: The below figure shows the Instance created with Ubuntu Operating Systems

Step 7: Click on connect it will redirect you to OS

## Assignment 1 – EC2 AND EFS

Step 9: The below picture shows the operating system is running

```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Tue Oct 7 15:34:51 UTC 2025

System load: 0.19      Temperature:          -273.1 °C
Usage of /: 25.6% of 6.71GB  Processes:           115
Memory usage: 22%        Users logged in:       0
Swap usage: 0%          IPv4 address for ena5: 172.31.30.95

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-30-95:~$
```

i-01f418a027ba83076 (instance-with-Ubuntu)

## Create EFS(Elastic File System)

Step 1: Login to your account and search EFS and click on EFS

```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Tue Oct 7 15:34:51 UTC 2025

System load: 0.19      Temperature:          -273.1 °C
Usage of /: 25.6% of 6.71GB  Processes:           115
Memory usage: 22%        Users logged in:       0
Swap usage: 0%          IPv4 address for ena5: 172.31.30.95

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-30-95:~$
```

Step 2: Click on Create File System as shown in below picture

The screenshot shows the AWS EFS console. On the left, there's a sidebar with 'Elastic File System' selected. The main area has a heading 'Amazon Elastic File System' with the subtext 'Scalable, elastic, cloud-native NFS file system'. Below this is a section titled 'What is Amazon Elastic File System?' featuring a small image of a file system icon. To the right, there's a 'Create file system' button with a red border, and another section titled 'Pricing' with some text and links.

## Assignment 1 – EC2 AND EFS

Step 3: EFS has been created as shown in below picture

The screenshot shows the AWS EFS console under the 'File systems' section. A green success message at the top states: 'Success! File system (fs-0fb43165b7e7f7e) is available.' Below this, a table lists the created file system: 'EFS-Connect' (fs-0fb43165b7e7f7e), which is encrypted, has a total size of 6.00 KiB, and is in the 'Available' state. The table includes columns for Name, File system ID, Encrypted, Total size, Size in Standard, Size in IA, Size in Archive, Provisioned Throughput (MiB/s), File system state, Creation time, Availability Zone, and Replication overwrites protection.

Step 4: Click on created EFS, there will be overview of the created EFS

The screenshot shows the detailed view of the EFS file system 'EFS-Connect'. It includes sections for General (Amazon resource name (ARN), Performance mode (General Purpose), Throughput mode (Elastic), Lifecycle management (Transition into Infrequent Access (90 days) since last access, Transition into Archive (90 days) since last access, Transition into Standard None), Availability zone (Regional)), Automatic backups (Enabled), Encrypted (fs-0fb43165b7e7f7e), File system state (Available), DNS name (fs-0fb43165b7e7f7e.efs.eu-north-1.amazonaws.com), and Replication overwrite protection (Enabled). Below this, a 'Network' tab displays network interface details across three availability zones (eu-north-1a, eu-north-1b, eu-north-1c) with their respective mount target IDs, subnet IDs, VPC IDs, and security group associations.

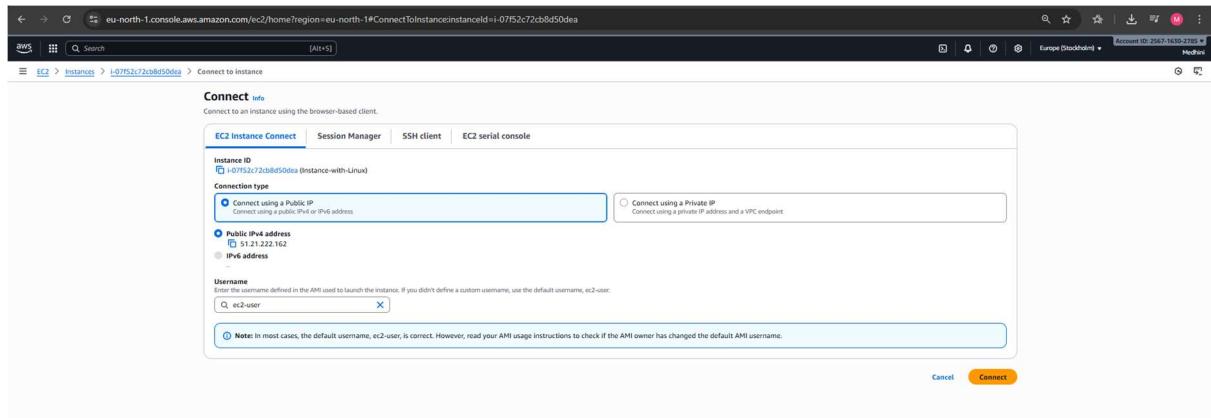
## Connect AWS Linux Instance to EFS

Step 1: Select AWS instance and click on connect

The screenshot shows the AWS EC2 Instances page. It lists three instances: 'Instance-with-Linux' (running, t3.micro, eu-north-1a, Public IP: ec2-51-21-222-162.eu...), 'Instance-with-Ubuntu' (running, t3.micro, eu-north-1a, Public IP: ec2-13-51-108-105.eu...), and 'RedHatInstance' (running, t3.micro, eu-north-1a, Public IP: ec2-16-170-172-85.eu...). The 'Instance-with-Linux' instance is selected. At the top right, there are buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'.

## Assignment 1 – EC2 AND EFS

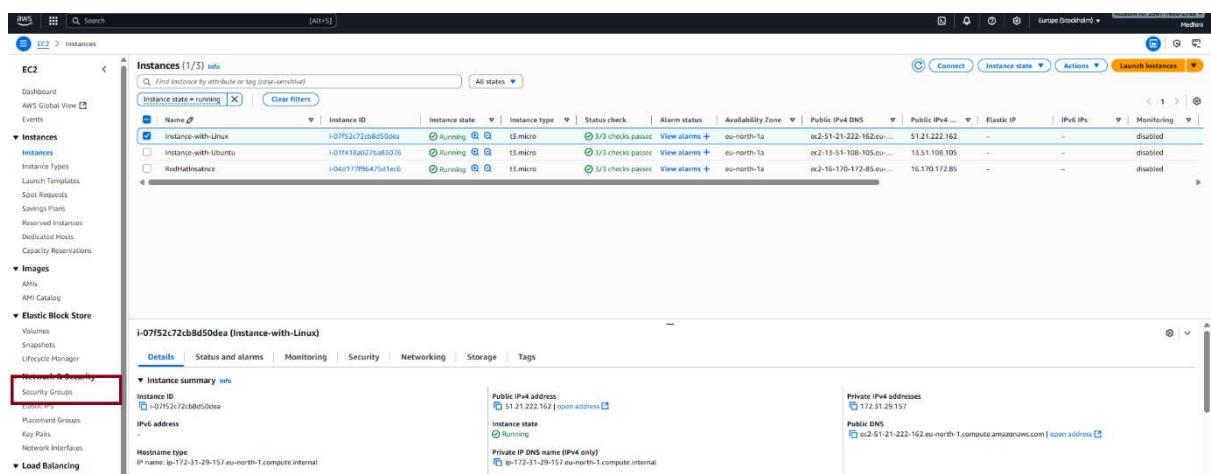
Step 2: You will redirect to below section, now click on Connect



Step 3: The OS will be running



Step 6: Click on Security Groups marked on below picture



## Assignment 1 – EC2 AND EFS

Step 6: Go to EFS and click On “NETWORK” and then click on “Manage”

Availability zone (AZ-ID)	Mount target ID	Subnet ID	VPC ID	Mount target state	IPv4 address	IPv6 address	Network interface ID	Security groups
eu-north-1a (eun1-az1)	fsmt-Oaad0311927cbe0bf	subnet-074b350c8fd96534a	vpc-008eb801be510f98	Available	172.31.29.206	N/A	eni-00e97899046e091fd	sg-09e96aa05b1dd6a61 (default)
eu-north-1b (eun1-az2)	fsmt-07a152af0143b5e7c	subnet-09849ad046eff0a6b	vpc-008eb801be510f98	Available	172.31.40.234	N/A	eni-00daf47a113bb5f4a	sg-09e96aa05b1dd6a61 (default)
eu-north-1c (eun1-az3)	fsmt-0061dc132b0f14f35	subnet-0e868b76b7bd7d71e	vpc-008eb801be510f98	Available	172.31.6.119	N/A	eni-0sf2a83a098727c19	sg-09e96aa05b1dd6a61 (default)

Step 7: Go to Security groups and click on “Edit Inbound Rules”

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-0e9125e78adca692c	IPv4	SSH	TCP	22	0.0.0.0/0	
sgr-064042396eec8667c	IPv4	NFS	TCP	2049	0.0.0.0/0	

## Assignment 1 – EC2 AND EFS

Step 8: Add new and Select NFS and then select “Anywhere ipv4”

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0e9125e78adca692c	SSH	TCP	22	Custom	0.0.0.0/0
sgr-064042396eec8667c	NFS	TCP	2049	Anywhere...	0.0.0.0/0

Add rule

⚠ Rules of 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.

Cancel Preview changes Save rules

Step 9: Then Go do OS running command window and paste “sudo yum install -y amazon-efs-utils” code to manually install  
then create a folder and add “sudo mount -t efs -o tls fs-0f0fb43165b7e7f7e:/ <folder name>”

### Amazon Linux 2 / Amazon Linux 2023

#### Step 1: Install required packages

# For Amazon Linux 2

sudo yum install -y amazon-efs-utils

# For Amazon Linux 2023

sudo dnf install -y amazon-efs-utils

#### Step 2: Create a mount point

sudo mkdir -p /mnt/efs

#### Step 3: Mount the EFS

sudo mount -t efs fs-xxxxxxxx:/ /mnt/efs

#### Optional: Add to /etc/fstab

echo "fs-xxxxxxxx:/ /mnt/efs efs defaults,\_netdev 0 0" | sudo tee -a /etc/fstab

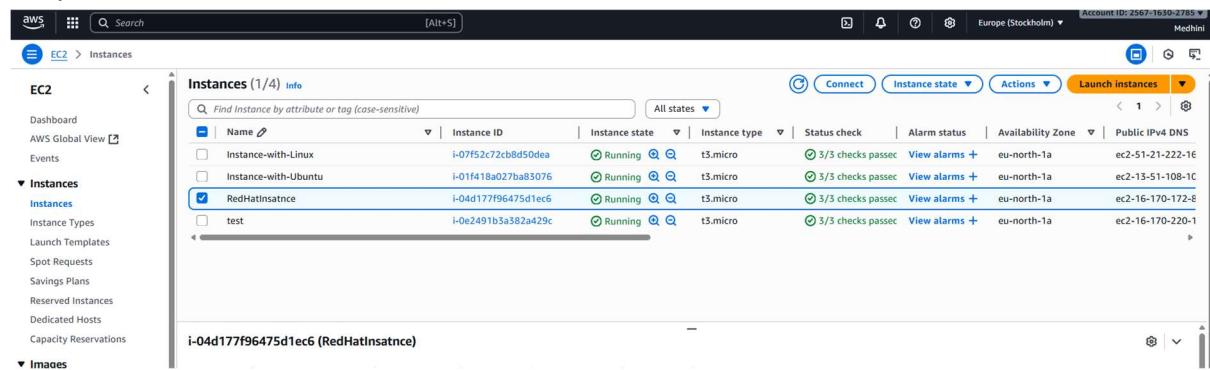
### The AWS Linux – Instance is mounted to EFS

## Assignment 1 – EC2 AND EFS

```
[ec2-user@ip-172-31-29-100 ~]$ sudo mount -t efs fs-0f0fb43165b7e7f7e:/ /mnt/efs
[ec2-user@ip-172-31-29-100 ~]$ df -h
Filesystem      Size   Used  Available  Mounted on
/dev/efs        500M    4.0M     496M  /mnt/efs
tmpfs          453M     0  453M  /dev/shm
tmpfs          181M  444K  181M  /run
/dev/nvme0n1p1  8.0G  1.6G  6.4G  208 /
tmpfs          453M     0  453M  /tmp
/dev/nvme0n1p2B 10M  1.3M  8.7M  138 /boot/efi
tmpfs          93M     0  93M  /run/user/1000
[ec2-user@ip-172-31-29-100 ~]$
```

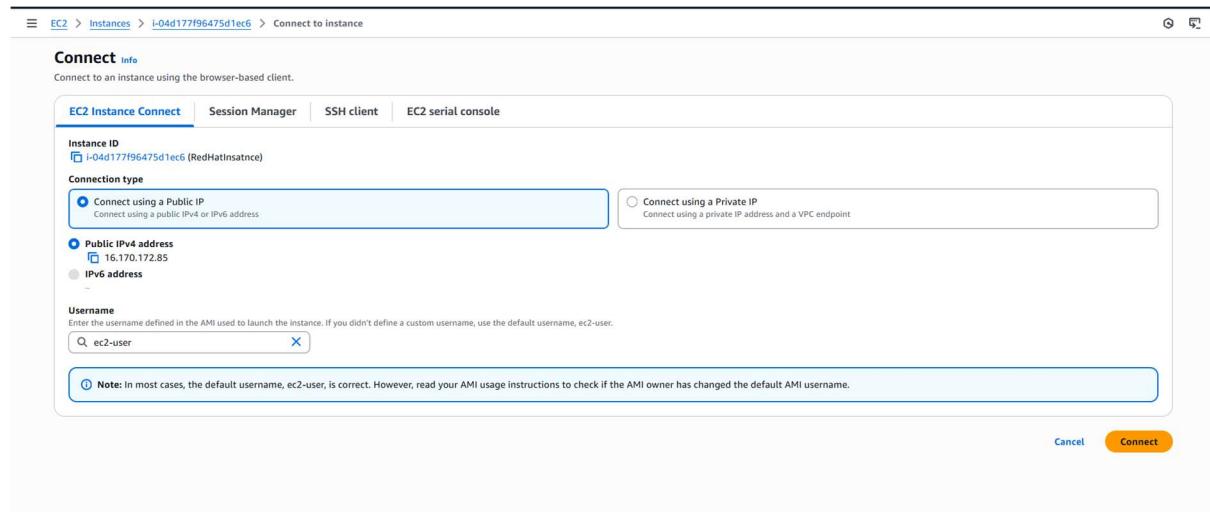
## RedHat EFS Mount

Step 1: Select instance and click on connect



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Instance-with-Linux	i-07f52c72cb8d50dea	Running	t3.micro	5/5 checks passed	<a href="#">View alarms</a>	eu-north-1a	ec2-51-21-222-1E
Instance-with-Ubuntu	i-01f418a027ba85076	Running	t3.micro	5/5 checks passed	<a href="#">View alarms</a>	eu-north-1a	ec2-13-51-108-1C
<b>RedHatsatnce</b>	i-04d177f96475d1ec6	Running	t3.micro	5/5 checks passed	<a href="#">View alarms</a>	eu-north-1a	ec2-16-170-172-8
test	i-0e2491b5a382a429c	Running	t3.micro	5/5 checks passed	<a href="#">View alarms</a>	eu-north-1a	ec2-16-170-220-1

Step 2: Click on Connect



Step 1: Install required packages

1. sudo yum install -y amazon-efs-utils
2. sudo yum install -y nfs-utils

Step 2: Create a mount point

## Assignment 1 – EC2 AND EFS

sudo mkdir -p /mnt/efs

### Step 3: Mount the EFS

Replace `fs-xxxxxxx` with your EFS ID and `<region>` with your AWS region.

sudo mount -t efs fs-xxxxxxx:/ /mnt/efs

Using the EFS mount helper:

```
sudo mount -t efs -o tls fs-0f0fb43165b7e7f7e:/ efs
```

Using the NFS client:

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,norespport fs-0f0fb43165b7e7f7e.efs.eu-north-1.amazonaws.com:/ efs
```

The RedHat OS – Instance has been successfully mounted on EFS

```
Installing : quota-nls-1:4.09-9.el10.noarch          4/10
Installing : quota-1:4.09-9.el10.x86_64           5/10
Installing : libev-4.33-14.el10.x86_64            6/10
Installing : libverto-libev-0.3-2.el10.x86_64       7/10
Running scriptlet: gssproxy-0.9.2-10.el10.x86_64    8/10
Installing : gssproxy-0.9.2-10.el10.x86_64           9/10
Running scriptlet: gssproxy-0.9.2-10.el10.x86_64       8/10
Running scriptlet: nfs-utils-1:2.8.2-3.el10.x86_64     9/10
Installing : nfs-utils-1:2.8.2-3.el10.x86_64           9/10
Running scriptlet: nfs-utils-1:2.8.2-3.el10.x86_64       9/10
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'.

Warning: The unit file, source configuration file or drop-ins of gssproxy.service changed on disk. Run 'systemctl daemon-reload' to reload units.
Warning: The unit file, source configuration file or drop-ins of gssproxy.service changed on disk. Run 'systemctl daemon-reload' to reload units.

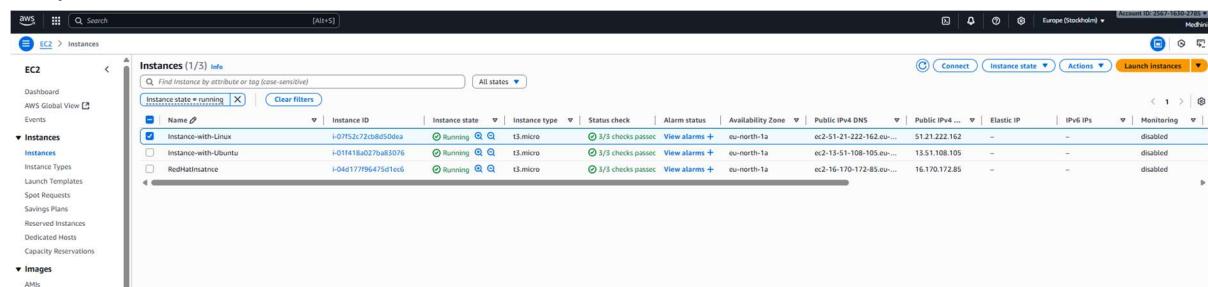
Installing : ssdd-nfs-idmap-2.10.2-3.el10.x86_64        10/10
Running scriptlet: ssdd-nfs-idmap-2.10.2-3.el10.x86_64      10/10
Installed products updated.

Installed:
gssproxy-0.9.2-10.el10.x86_64          libev-4.33-14.el10.x86_64          libnfsidmap-1:2.8.2-3.el10.x86_64
libtirpc-1.3.5-1.el10.x86_64           libverto-libev-0.3-2.el10.x86_64          nfs-utils-1:2.8.2-3.el10.x86_64
quota-nls-1:4.09-9.el10.noarch         quota-1:4.09-9.el10.x86_64           rpcbind-1.2.7-3.el10.x86_64
ssdd-nfs-idmap-2.10.2-3.el10.x86_64

Complete!
[ec2-user@ip-172-31-23-156 ~]$ sudo mkdir -p /mnt/efs
[ec2-user@ip-172-31-23-156 ~]$ sudo mount -t efs -o tls fs-0f0fb43165b7e7f7e:/ /mnt/efs
mount: /mnt/efs: unknown filesystem type 'efs'.
Try 'fsck' or 'e2fsck' after failed mount system call.
[ec2-user@ip-172-31-23-156 ~]$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,norespport fs-0f0fb43165b7e7f7e.efs.eu-north-1.amazonaws.com:/ /mnt/efs
[ec2-user@ip-172-31-23-156 ~]$ df -h
Filesystem      Size  Used  Avail Avail% Mounted on
/dev/nvme0n1p3   9.8G  8.1G  1.8G  83% /
devtmpfs        4.0M  0  4.0M  0% /dev
tmpfs          100M  0  100M  0% /dev/shm
efivars          128K  0  128K  0% /efi/firmware/efi/efivars
tmpfs          1.0M  0  1.0M  0% /run/credentials/systemd-journal.service
/dev/nvme0n1p2   200M  8.4M  192M  5% /boot/efi
tmpfs          1.0M  0  1.0M  0% /run/credentials/serial-getty@ttyS0.service
tmpfs          1.0M  0  1.0M  0% /run/credentials/getty@tty1.service
tmpfs          91M  4.8K  91M  1% /run/user/1000
fs-0f0fb43165b7e.efs.eu-north-1.amazonaws.com:  8.0E  0  8.0E  0% /mnt/efs
[ec2-user@ip-172-31-23-156 ~]$
```

## Ubuntu EFS Mount

Step 1: Select AWS instance and click on connect

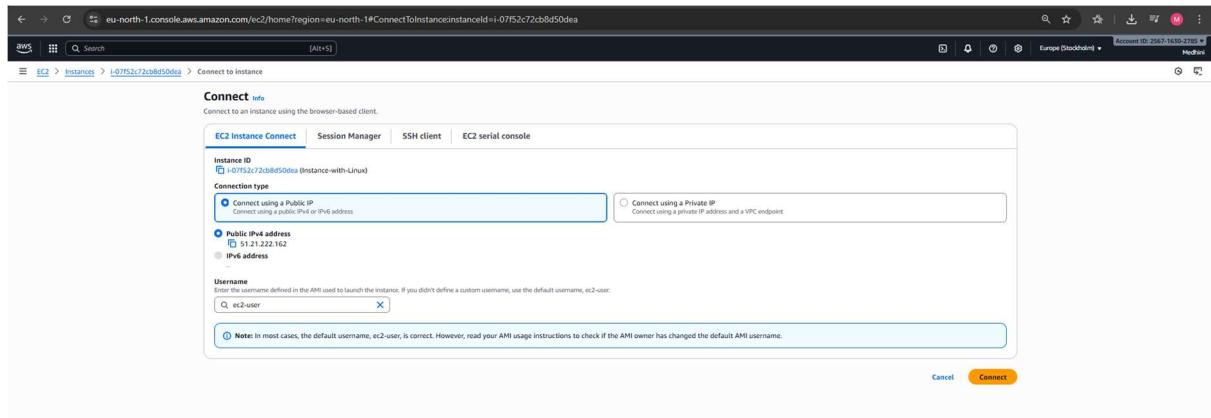


The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with 'EC2' selected. The main area displays a table of instances. There are three instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPv6 IPs	Monitoring
Instance-with-Linux	i-07f52c720bd5d0ea	Running	t3.micro	3/3 checks passed	View alarms +	eu-north-1a	ec2-51-21-222-162.eu... 51.21.222.162	-	-	-	disabled
Instance-with-Ubuntu	i-011418a027ba81076	Running	t3.micro	3/3 checks passed	View alarms +	eu-north-1a	ec2-13-51-108-105.eu... 13.51.108.105	-	-	-	disabled
RedHatInstance	i-04d17796475d1ec6	Running	t3.micro	3/3 checks passed	View alarms +	eu-north-1a	ec2-16-170-172-85.eu... 16.170.172.85	-	-	-	disabled

# Assignment 1 – EC2 AND EFS

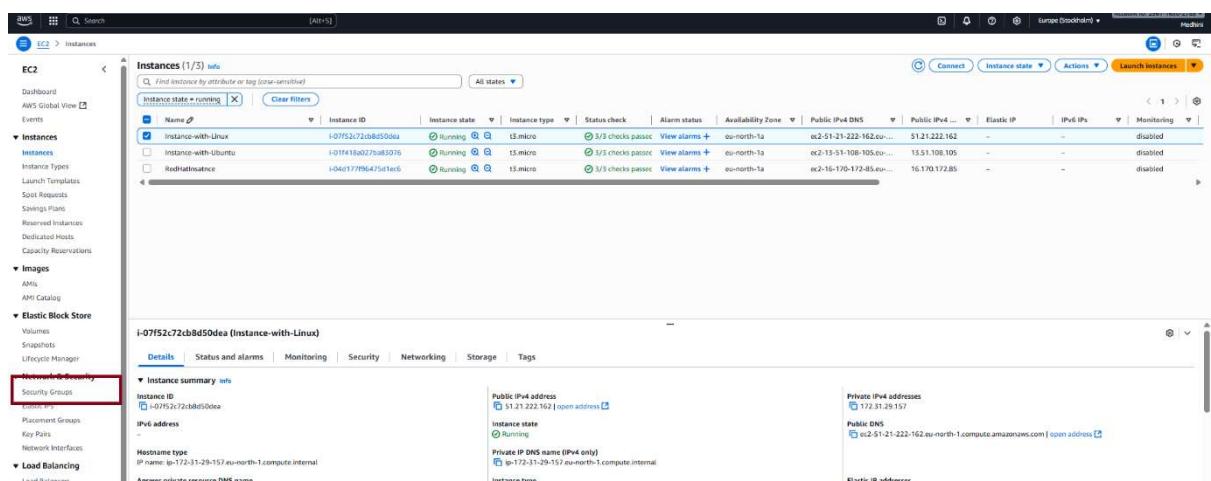
Step 2: You will redirect to below section, now click on Connect



Step 3: The OS will be running



Step 6: Click on Security Groups marked on below picture



## Assignment 1 – EC2 AND EFS

Step 6: Go to EFS and click On “NETWORK” and then click on “Manage”

Availability zone (AZ-ID)	Mount target ID	Subnet ID	VPC ID	Mount target state	IPv4 address	IPv6 address	Network interface ID	Security groups
eu-north-1a (eun1-az1)	fsmt-Oaad0311927cbe0bf	subnet-074b350c8fd96534a	vpc-008eb801be510f98	Available	172.31.29.206	N/A	eni-00e97899046e091fd	sg-09e96aa05b1dd6a61 (default)
eu-north-1b (eun1-az2)	fsmt-07a152af0143b5e7c	subnet-09849ad046eff0a6b	vpc-008eb801be510f98	Available	172.31.40.234	N/A	eni-00daf47a113bb5f4a	sg-09e96aa05b1dd6a61 (default)
eu-north-1c (eun1-az3)	fsmt-0061dc132b0f14f35	subnet-0e868b76b7bd7d71e	vpc-008eb801be510f98	Available	172.31.6.119	N/A	eni-0sf2a83a098727c19	sg-09e96aa05b1dd6a61 (default)

Step 7: Go to Security groups and click on “Edit Inbound Rules”

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-0e9125e78adca692c	IPv4	SSH	TCP	22	0.0.0.0/0	
sgr-064042396eec8667c	IPv4	NFS	TCP	2049	0.0.0.0/0	

## Assignment 1 – EC2 AND EFS

### Step 8: Add new and Select NFS and then select “Anywhere ipv4”

The screenshot shows the 'Edit inbound rules' page for a security group. There are two rules listed:

- SSH**: Type: SSH, Protocol: TCP, Port range: 22, Source: Custom (0.0.0.0/0), Description: optional.
- NFS**: Type: NFS, Protocol: TCP, Port range: 2049, Source: Anywhere..., Description: optional.

A warning message at the bottom states: "⚠ 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." Buttons for 'Cancel', 'Preview changes', and 'Save rules' are at the bottom right.

### Step 9: Use Below code to mount on Ubuntu

```
sudo apt install -y nfs-common
```

```
sudo mkdir -p /mnt/efs
```

```
sudo mount -t nfs4 -o nfsvers=4.1 fs-xxxxxxxx.efs.<region>.amazonaws.com:/ /mnt/efs
```

```
df -h (This last one is to verify the mounted ec2)
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-30-95:~$ sudo mkdir -p /mnt/efs
ubuntu@ip-172-31-30-95:~$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-0f0fb43165b7e7f7e.efs.eu-north-1.amazonaws.com:/ /mnt/efs
ubuntu@ip-172-31-30-95:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/efivars    128M   36K  120K  3%   /sys/firmware/efi/efivars
/dev/nvme0n1p16 861M  87M  794M  1%   /boot/efi
/dev/nvme0n1p15 105M  63M  99M  6%   /boot/efi
tmpfs          5.0M   0B  5.0M  0%   /run/lock
tmpfs          5.0M   0B  5.0M  0%   /run
tmpfs          5.0M   0B  5.0M  0%   /run/user/1000
fs-0f0fb43165b7e7f7e.efs.eu-north-1.amazonaws.com:/  8.0E   0  8.0E  0%  /mnt/efs
ubuntu@ip-172-31-30-95:~$
```