

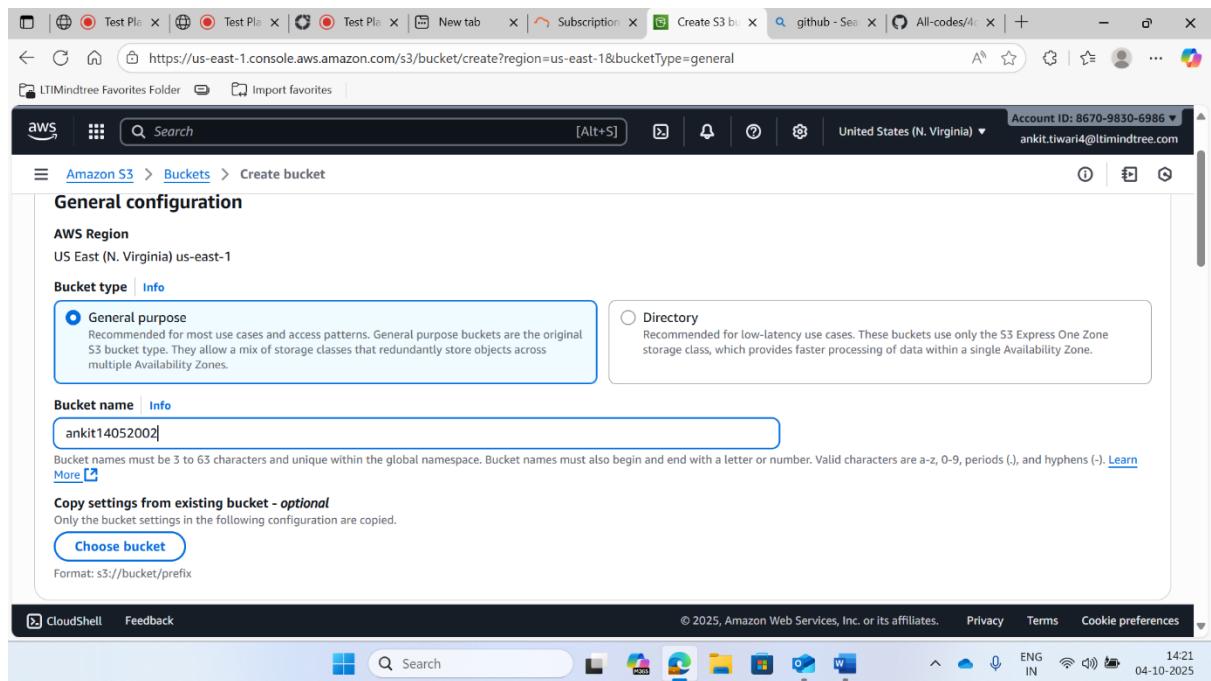
Milestone Assessment 1

Name: Ankit Tiwari

Q1>Launch an AWS S3 bucket with Uniq name. and upload some objects. And this S3 bucket should be reachable on Linux host from where I can upload the object in the AWS S3 bucket.

Ans->

->Creating an s3 bucket name: ankit14052002



->Bucket created in N. virginia region

General purpose buckets (1) [Info](#)

[Create bucket](#)

Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
ankit14052002	US East (N. Virginia) us-east-1	October 4, 2025, 14:23:24 (UTC+05:30)

Account snapshot [Info](#)
Updated daily

[View dashboard](#)

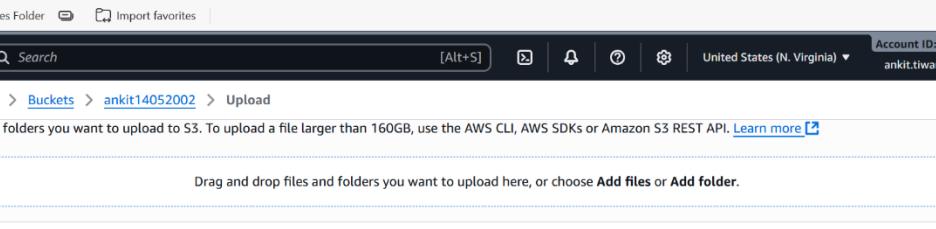
Storage Lens provides visibility into storage usage and activity trends.

External access summary - [new](#)
Updated daily

[Info](#)

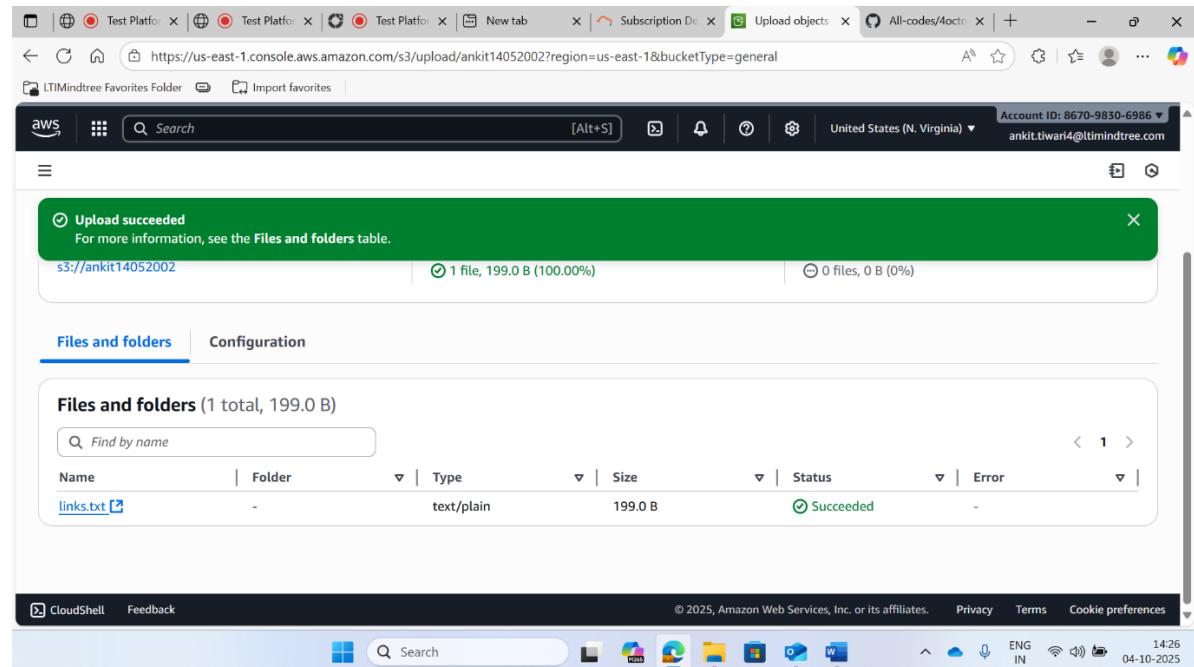
External access findings help you identify bucket permissions that allow public access or access from other AWS accounts.

->Adding files to the bucket



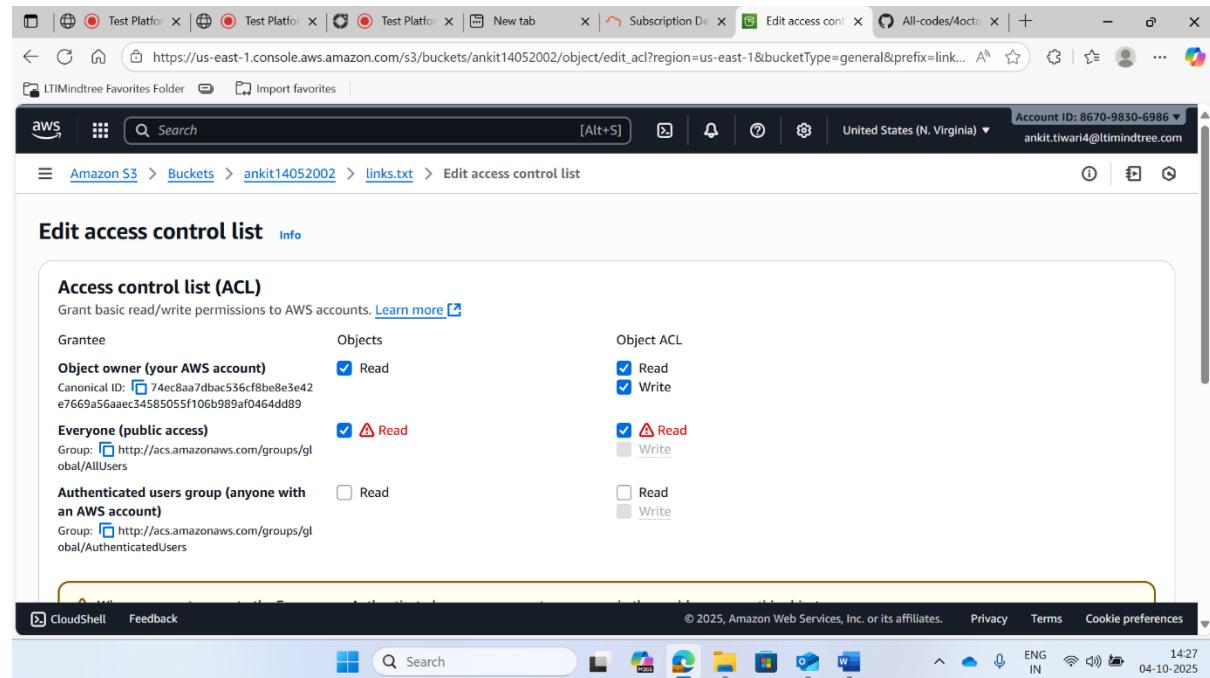
The screenshot shows the AWS S3 'Upload objects' interface. The URL in the address bar is <https://us-east-1.console.aws.amazon.com/s3/upload/ankit14052002?region=us-east-1&bucketType=general>. The top navigation bar includes tabs for 'Test Platform' (x3), 'New tab', 'Subscription De...', 'Upload objects', and 'All-codes/40cto...'. The AWS logo is in the top-left corner, and the top-right corner shows 'Account ID: 8670-9830-6986' and the email 'ankit.tiwari4@litimindtree.com'. The main content area shows the 'Amazon S3 > Buckets > ankit14052002 > Upload' path. A message encourages adding files and folders, with a link to 'Learn more'. A large blue dashed box allows for dragging and dropping files. Below this, a table lists 'Files and folders (1 total, 199.0 B)'. The table has columns for 'Name', 'Folder', 'Type', and 'Size'. A single file 'links.txt' is listed as a text/plain file of size 199.0 B. Buttons for 'Remove', 'Add files', and 'Add folder' are available. The 'Destination' section shows the URL 's3://ankit14052002'. The bottom navigation bar includes 'CloudShell', 'Feedback', '© 2025, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'. The bottom-right corner shows system icons for battery, signal, and date/time (14:26, 04-10-2025).

File successfully added



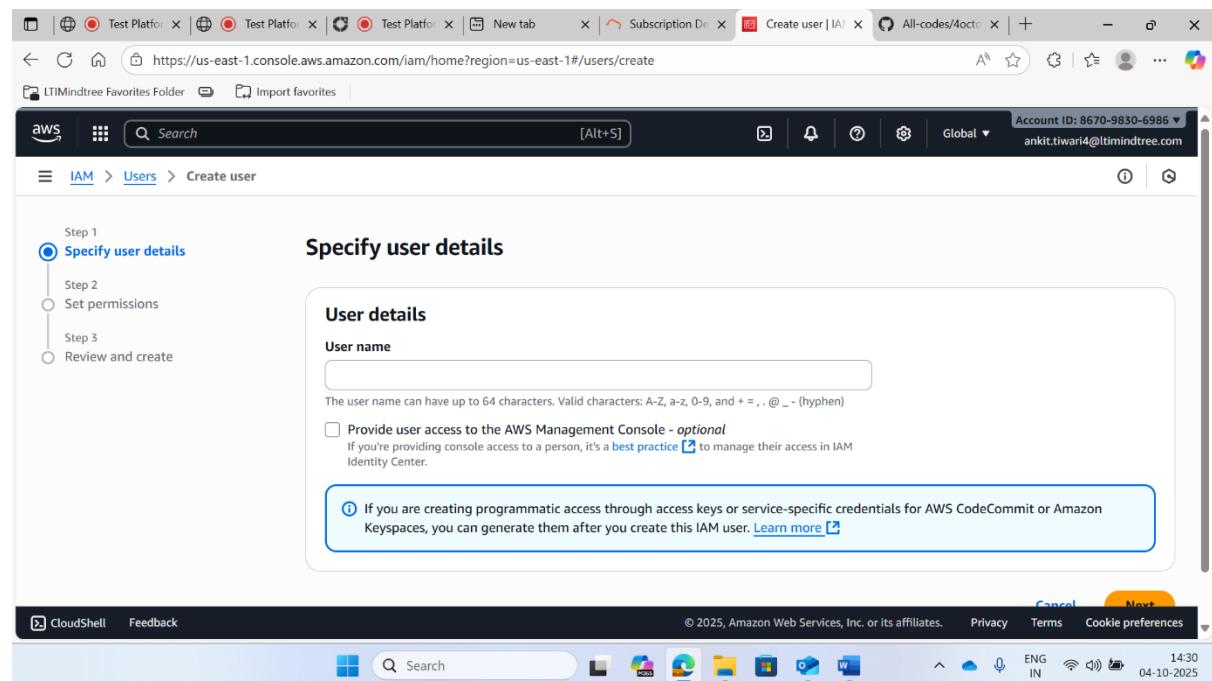
The screenshot shows the AWS S3 console interface. At the top, a green success message box displays "Upload succeeded" and "For more information, see the Files and folders table." Below this, a summary bar shows "1 file, 199.0 B (100.00%)" and "0 files, 0 B (0%)". The main area is titled "Files and folders (1 total, 199.0 B)" and lists a single file named "links.txt" with a size of 199.0 B, a type of text/plain, and a status of "Succeeded". The interface includes a search bar, a toolbar with various icons, and a navigation bar at the bottom.

→ Giving public access to the file so that it should be readable



The screenshot shows the "Edit access control list (ACL)" page for the "links.txt" file. The "Access control list (ACL)" section allows granting basic read/write permissions to AWS accounts. It shows three entries: "Object owner (your AWS account)" with "Read" and "Write" checked; "Everyone (public access)" with "Read" checked and "Write" unchecked; and "Authenticated users group (anyone with an AWS account)" with "Read" unchecked and "Write" unchecked. The interface includes a search bar, a toolbar with various icons, and a navigation bar at the bottom.

-> Creating a user in IAM to generate access key id and secret access key



Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

User details

User name

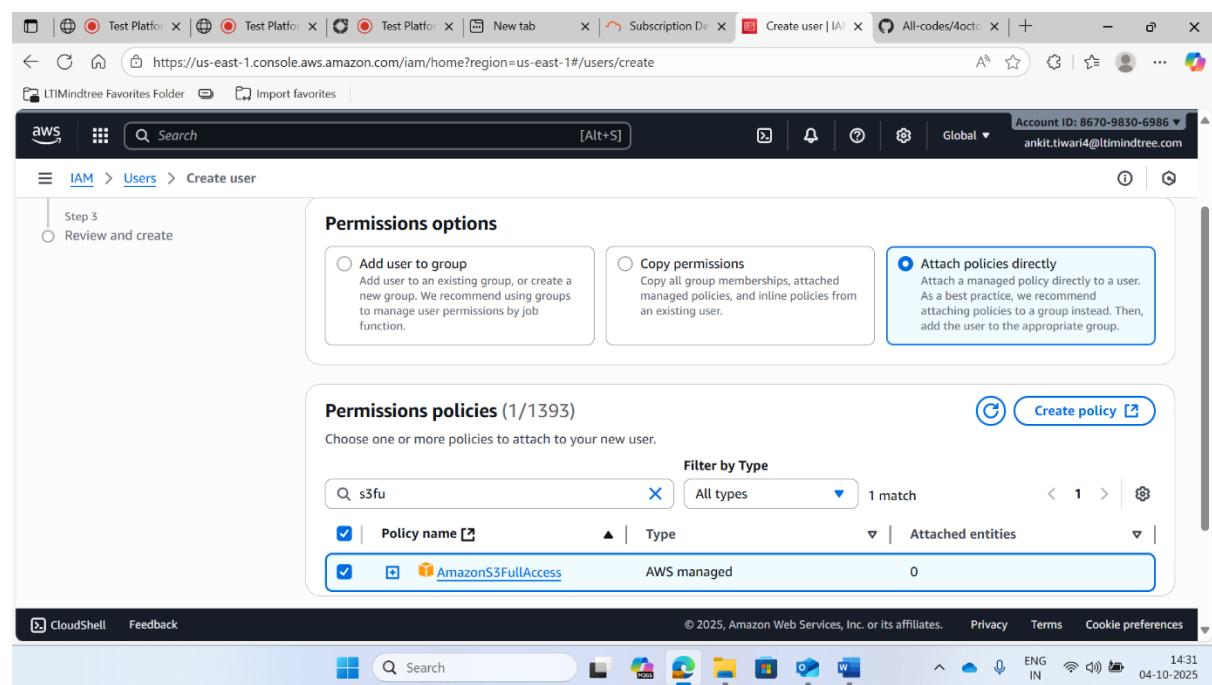
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, ., @, -, (hyphen)

Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 14:30 IN 04-10-2025

→ Giving policies to user



Step 3
Review and create

Permissions options

Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1393)

Choose one or more policies to attach to your new user.

Filter by Type

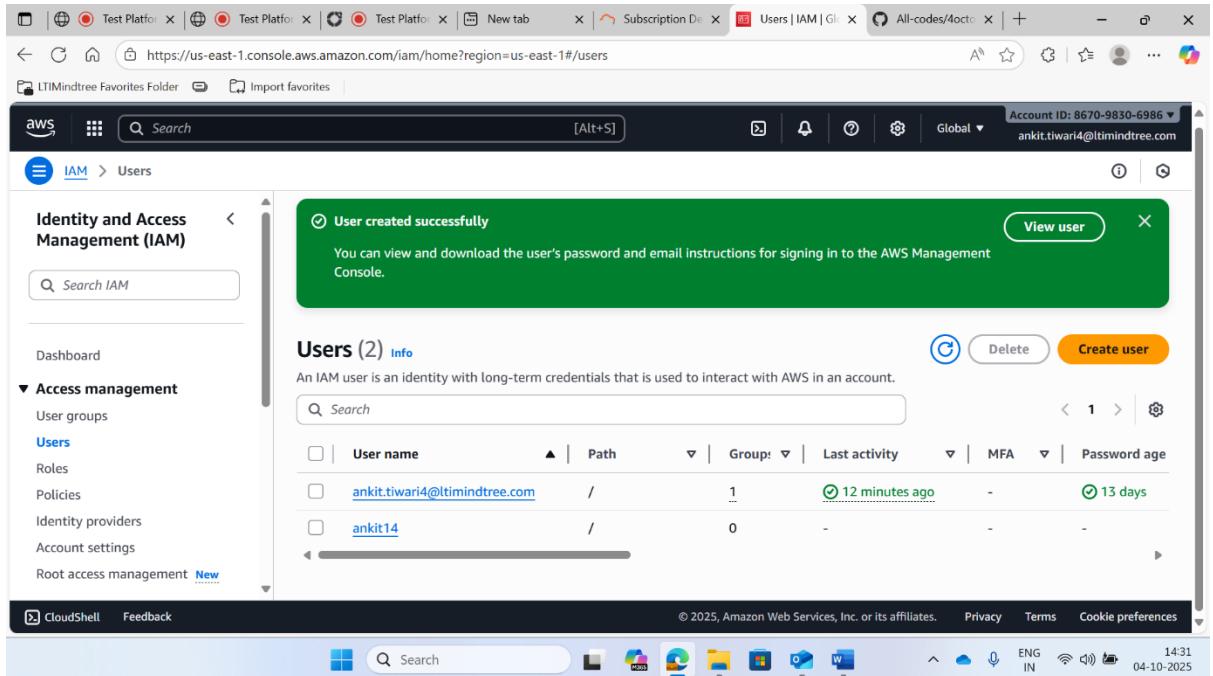
Q s3fu X All types 1 match

Policy name Type Attached entities

AmazonS3FullAccess AWS managed 0

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 14:31 IN 04-10-2025

→ User successfully generated

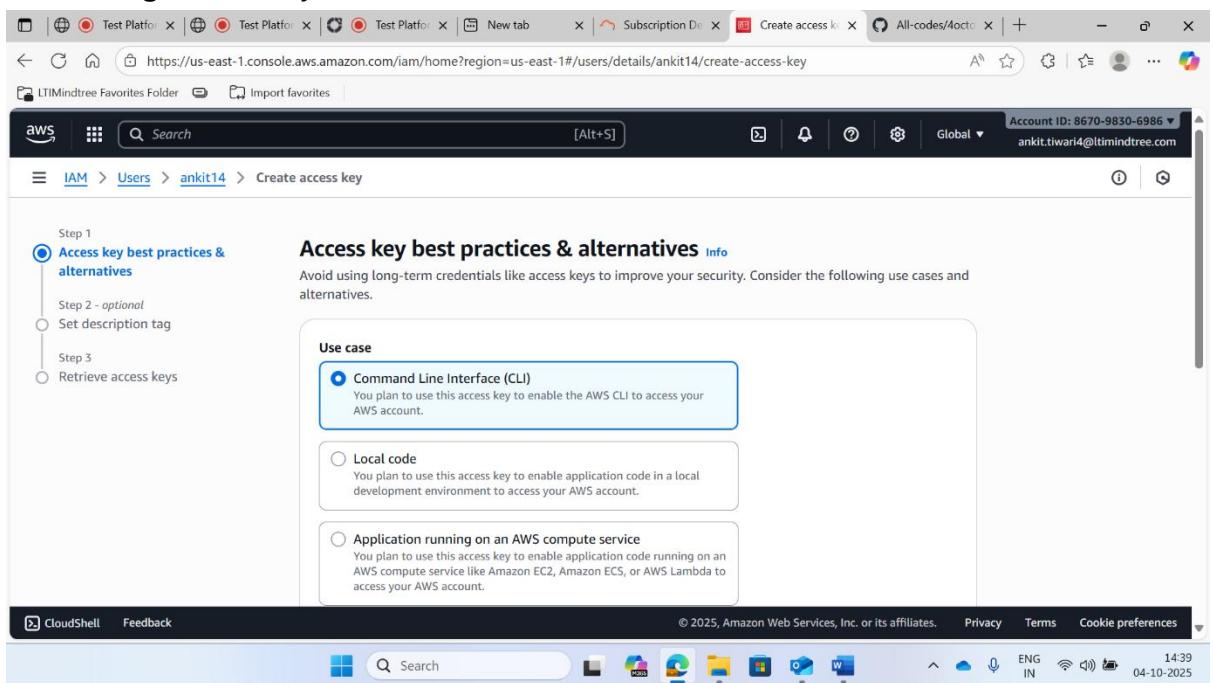


The screenshot shows the AWS IAM Users page. A green success message box at the top right says "User created successfully" and "You can view and download the user's password and email instructions for signing in to the AWS Management Console." Below this, the "Users (2) Info" section shows a table with two entries:

User name	Path	Group	Last activity	MFA	Password age
ankit.tiwari4@ltimindtree.com	/	1	12 minutes ago	-	13 days
ankit14	/	0	-	-	-

The left sidebar shows the IAM navigation menu with "Users" selected. The bottom of the screen includes standard browser navigation and status bars.

→ Generating access key for the user

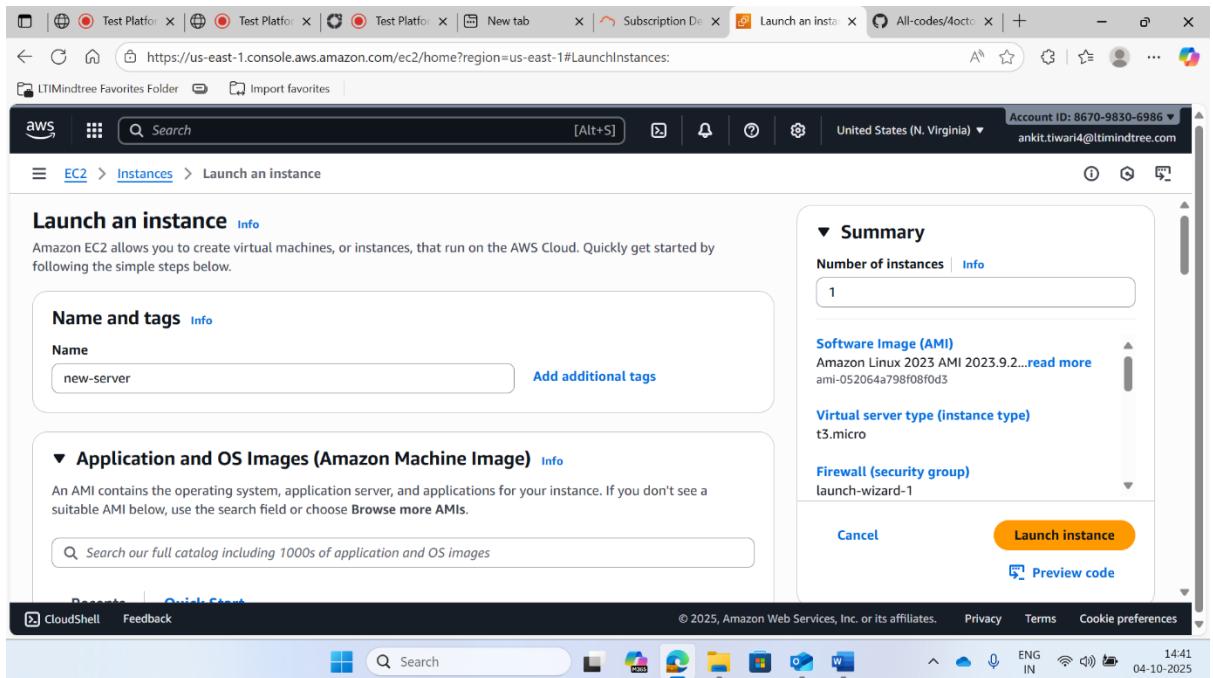


The screenshot shows the "Create access key" step 1 page. The left sidebar has three options: "Access key best practices & alternatives" (selected), "Set description tag" (unchecked), and "Retrieve access keys" (unchecked). The main content area is titled "Access key best practices & alternatives" and includes a note: "Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives." Below this is a "Use case" section with three options:

- Command Line Interface (CLI)**
You plan to use this access key to enable the AWS CLI to access your AWS account.
- Local code**
You plan to use this access key to enable application code in a local development environment to access your AWS account.
- Application running on an AWS compute service**
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

The bottom of the screen includes standard browser navigation and status bars.

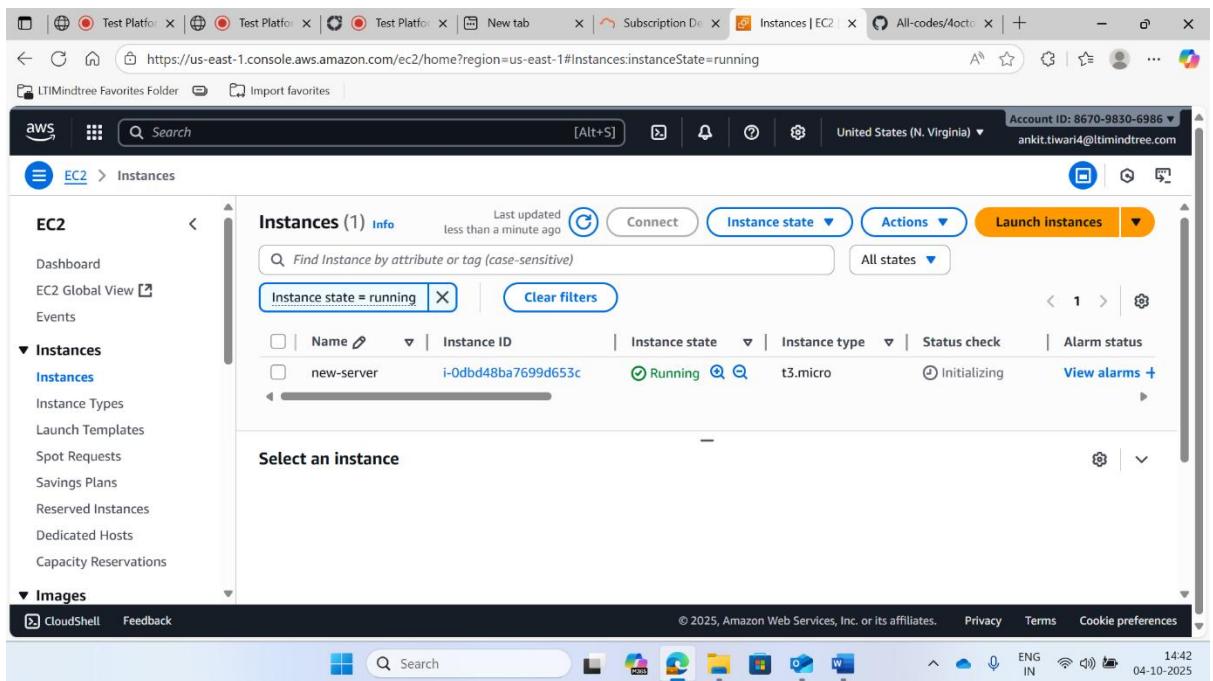
→ Creating an instance in EC2



The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. The steps are as follows:

- Name and tags**: A text input field contains 'new-server'. A link 'Add additional tags' is visible.
- Application and OS Images (Amazon Machine Image)**: A search bar with placeholder 'Search our full catalog including 1000s of application and OS images'.
- Summary**: Shows 1 instance, AMI 'Amazon Linux 2023 AMI 2023.9.2...', instance type 't3.micro', and security group 'launch-wizard-1'. Buttons for 'Cancel', 'Launch instance', and 'Preview code' are present.

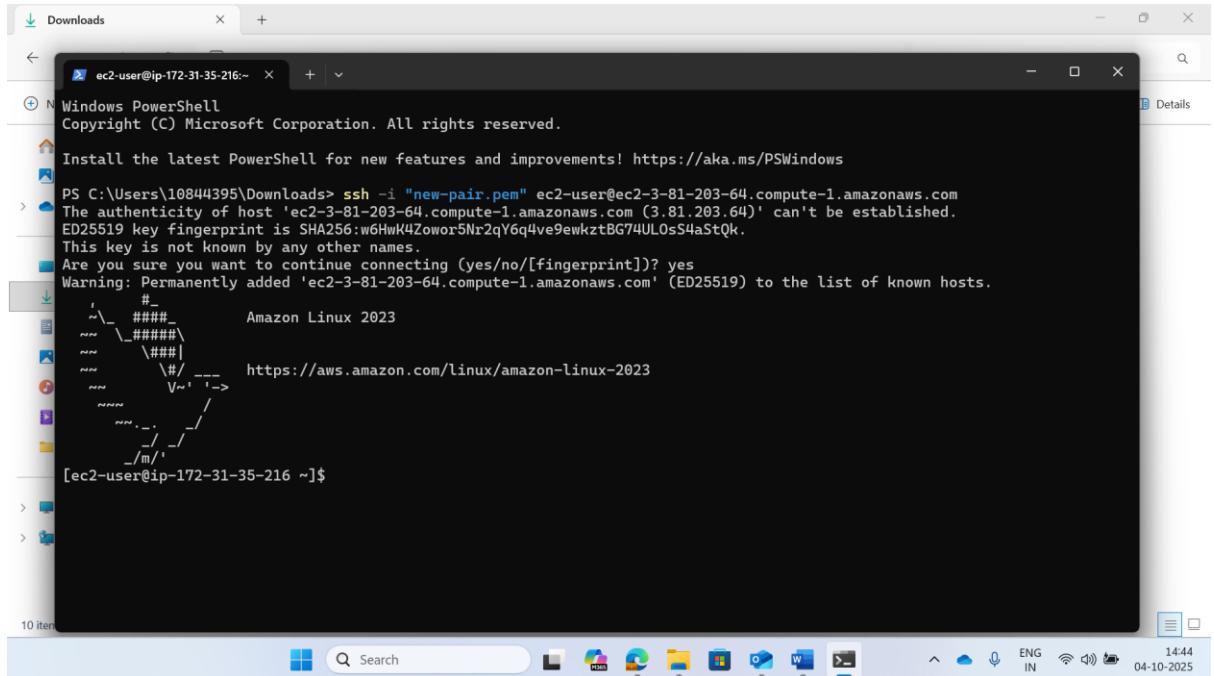
→ Instance created



The screenshot shows the 'Instances' page in the AWS EC2 console, displaying the newly created instance 'new-server'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
new-server	i-0dbd48ba7699d653c	Running	t3.micro	Initializing	View alarms

→ Connecting instance to terminal via ssh



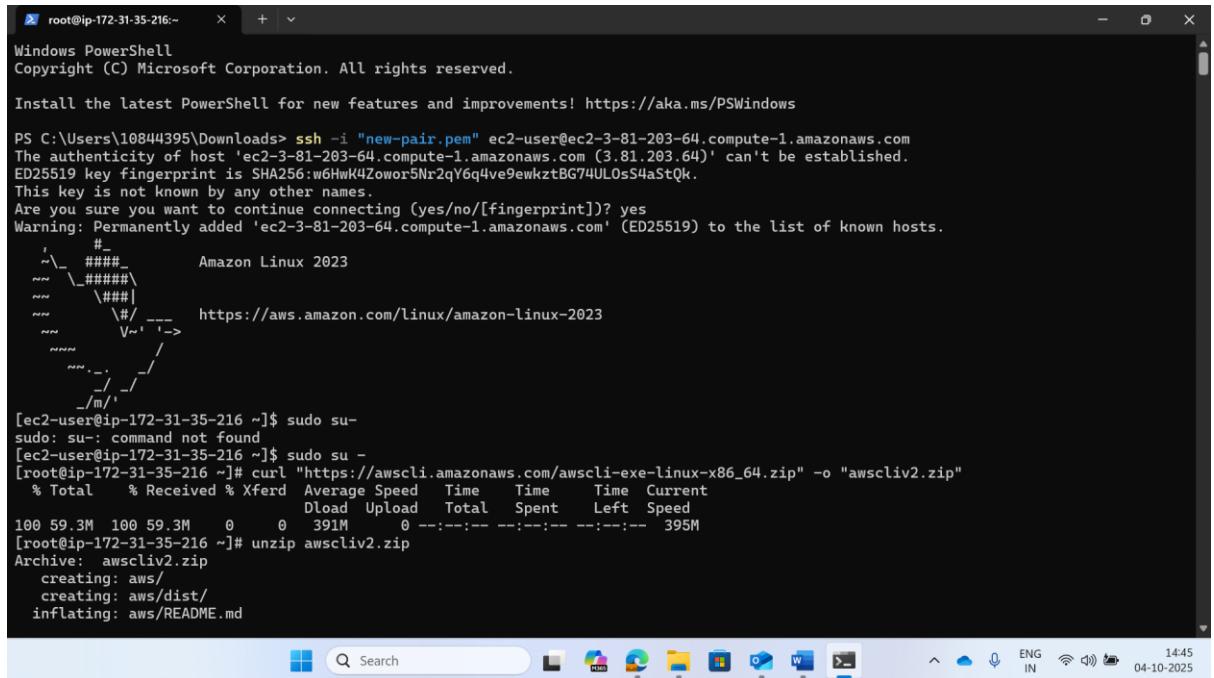
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10844395\Downloads> ssh -i "new-pair.pem" ec2-user@ec2-3-81-203-64.compute-1.amazonaws.com
The authenticity of host 'ec2-3-81-203-64.compute-1.amazonaws.com (3.81.203.64)' can't be established.
ED25519 key fingerprint is SHA256:w6HwK4Zowor5Nr2qY6q4ve9ewkztBG74UL0s54aStQk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-81-203-64.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_
`_\_ #####_      Amazon Linux 2023
~~ \_\####\_
~~ \###|
~~ \#/  ___>  https://aws.amazon.com/linux/amazon-linux-2023
~~ \~'`-->
~~ /`_
~~ /`_/
~~ /`_/
[ec2-user@ip-172-31-35-216 ~]$
```

→ Installing awcli package on the terminal



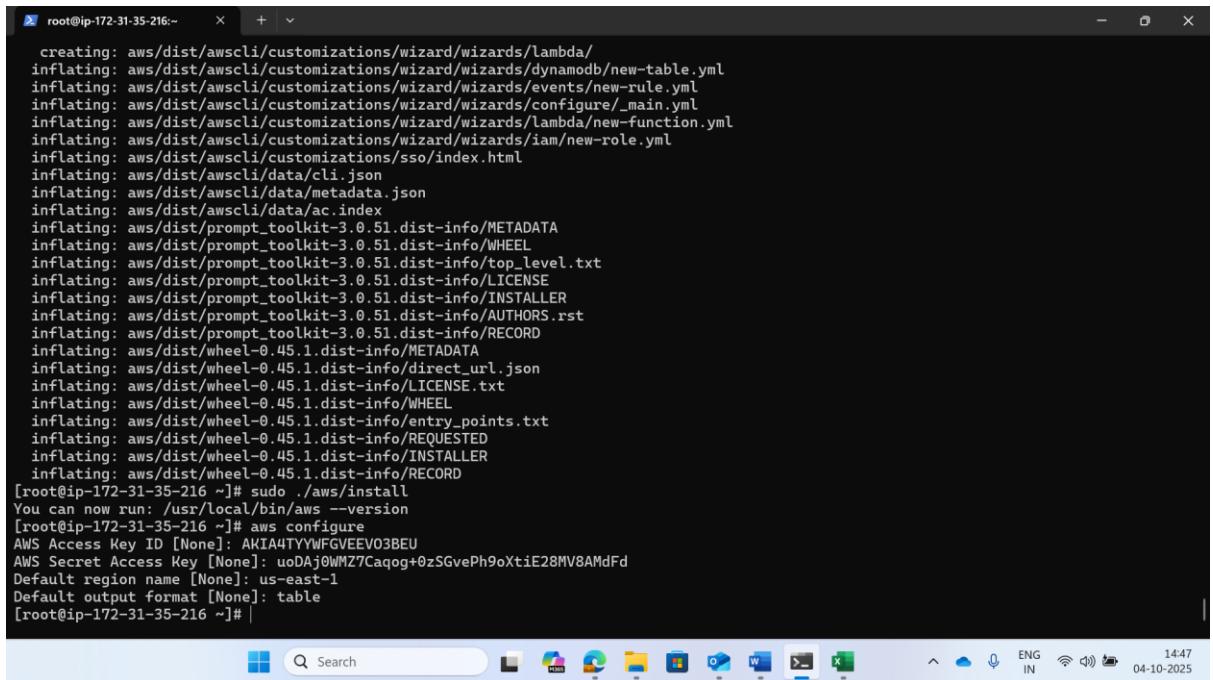
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10844395\Downloads> ssh -i "new-pair.pem" ec2-user@ec2-3-81-203-64.compute-1.amazonaws.com
The authenticity of host 'ec2-3-81-203-64.compute-1.amazonaws.com (3.81.203.64)' can't be established.
ED25519 key fingerprint is SHA256:w6HwK4Zowor5Nr2qY6q4ve9ewkztBG74UL0s54aStQk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-81-203-64.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

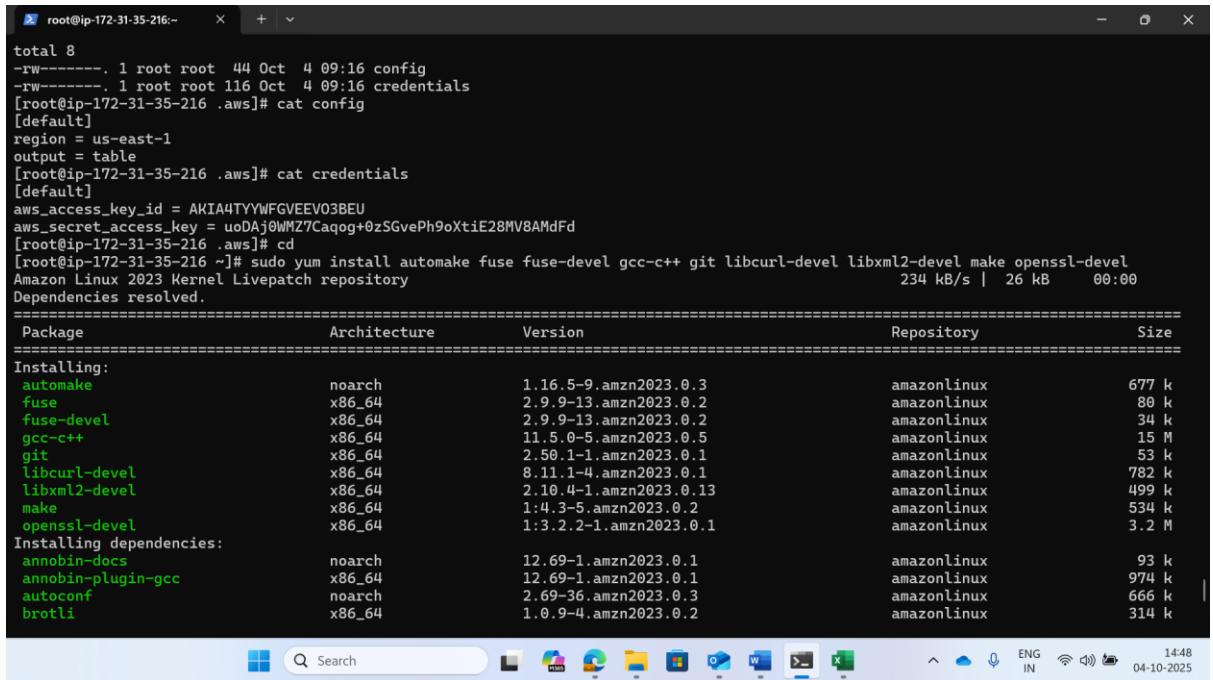
#_
`_\_ #####_      Amazon Linux 2023
~~ \_\####\_
~~ \###|
~~ \#/  ___>  https://aws.amazon.com/linux/amazon-linux-2023
~~ \~'`-->
~~ /`_
~~ /`_/
~~ /`_/
[ec2-user@ip-172-31-35-216 ~]$ sudo su-
sudo: su: command not found
[ec2-user@ip-172-31-35-216 ~]$ sudo su -
[root@ip-172-31-35-216 ~]# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload Total   Spent    Left Speed
100 59.3M 100 59.3M  0      0  391M  0  --:--:--  --:--:-- 395M
[root@ip-172-31-35-216 ~]# unzip awscliv2.zip
Archive: awscliv2.zip
  creating: aws/
  creating: aws/dist/
  inflating: aws/README.md
```

→ providing access key , region ,format etc



```
creating: aws/dist/awscli/customizations/wizard/wizards/lambda/
inflating: aws/dist/awscli/customizations/wizard/wizards/dynamodb/new-table.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/events/new-rule.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/configure/_main.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/lambda/new-function.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/iam/new-role.yml
inflating: aws/dist/awscli/customizations/sso/index.html
inflating: aws/dist/awscli/data/cli.json
inflating: aws/dist/awscli/data/metadata.json
inflating: aws/dist/awscli/data/ac.index
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/METADATA
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/WHEEL
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/top_level.txt
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/LICENSE
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/INSTALLER
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/AUTHORS.rst
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/RECORD
inflating: aws/dist/wheel-0.45.1.dist-info/METADATA
inflating: aws/dist/wheel-0.45.1.dist-info/direct_url.json
inflating: aws/dist/wheel-0.45.1.dist-info/LICENSE.txt
inflating: aws/dist/wheel-0.45.1.dist-info/WHEEL
inflating: aws/dist/wheel-0.45.1.dist-info/entry_points.txt
inflating: aws/dist/wheel-0.45.1.dist-info/REQUESTED
inflating: aws/dist/wheel-0.45.1.dist-info/INSTALLER
inflating: aws/dist/wheel-0.45.1.dist-info/RECORD
[root@ip-172-31-35-216 ~]# sudo ./aws/install
You can now run: /usr/local/bin/aws --version
[root@ip-172-31-35-216 ~]# aws configure
AWS Access Key ID [None]: AKIA4TYWFGVEEV03BEU
AWS Secret Access Key [None]: uoDAj0WMZ7Caqog+0zSGvePh9oXtiE28MV8AMdFd
Default region name [None]: us-east-1
Default output format [None]: table
[root@ip-172-31-35-216 ~]# |
```

->installing automake fuse package



```
total 8
-rw----- 1 root root 44 Oct  4 09:16 config
-rw----- 1 root root 116 Oct  4 09:16 credentials
[root@ip-172-31-35-216 .aws]# cat config
[default]
region = us-east-1
output = table
[root@ip-172-31-35-216 .aws]# cat credentials
[default]
aws_access_key_id = AKIA4TYWFGVEEV03BEU
aws_secret_access_key = uoDAj0WMZ7Caqog+0zSGvePh9oXtiE28MV8AMdFd
[root@ip-172-31-35-216 .aws]# cd
[root@ip-172-31-35-216 ~]# sudo yum install automake fuse fuse-devel gcc-c++ git libcurl-devel libxml2-devel make openssl-devel
Amazon Linux 2023 Kernel Livepatch repository                                         234 kB/s |  26 kB   00:00
Dependencies resolved.
=====
Package           Architecture      Version       Repository      Size
=====
Installing:
automake          noarch          1.16.5-9.amzn2023.0.3      amazonlinux   677 k
fuse              x86_64          2.9.9-13.amzn2023.0.2      amazonlinux   80 k
fuse-devel        x86_64          2.9.9-13.amzn2023.0.2      amazonlinux   34 k
gcc-c++           x86_64          11.5.0-5.amzn2023.0.5      amazonlinux   15 M
git               x86_64          2.50.1-1.amzn2023.0.1      amazonlinux   53 k
libcurl-devel     x86_64          8.11.1-4.amzn2023.0.1      amazonlinux   782 k
libxml2-devel     x86_64          2.10.4-1.amzn2023.0.13     amazonlinux   499 k
make              x86_64          1:4.3-5.amzn2023.0.2      amazonlinux   534 k
openssl-devel     x86_64          1:3.2.2-1.amzn2023.0.1     amazonlinux   3.2 M
Installing dependencies:
annobin-docs       noarch          12.69-1.amzn2023.0.1      amazonlinux   93 k
annobin-plugin-gcc x86_64          12.69-1.amzn2023.0.1      amazonlinux   974 k
autoconf          noarch          2.69-36.amzn2023.0.3      amazonlinux   666 k
brotli            x86_64          1.0.9-4.amzn2023.0.2      amazonlinux   314 k
```

->Adding access key id and secret access key in vim editor file for accessing the user.

->Performing make operation

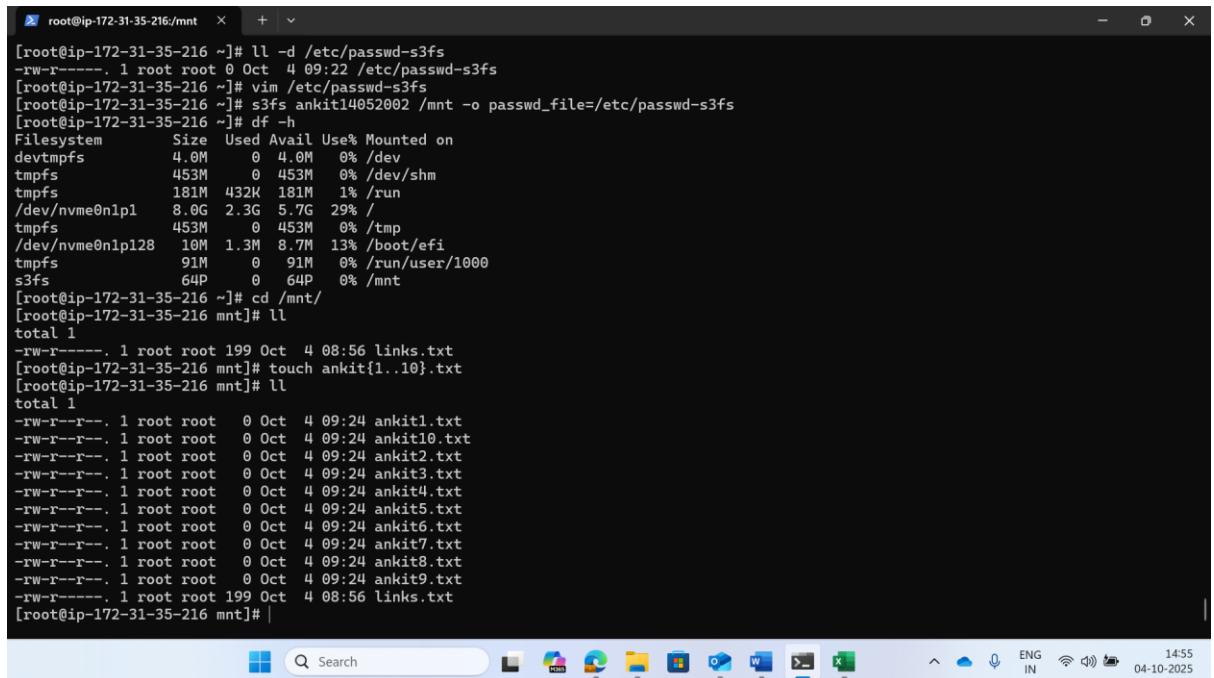
```
root@ip-172-31-35-216:/mnt ~ + - 14:55 04-10-2023 ENG IN WiFi 04-10-2023

mv -f .deps/write_multiblock.Tpo .deps/write_multiblock.Po
g++ -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_FORTIFY_SOURCE=3 -std=c++14 -g -O2 -o write_multiblock write_multiblock.o -ldl
g++ -DHAVE_CONFIG_H -I. -I. -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_FORTIFY_SOURCE=3 -std=c++14 -g -O2 -MT mknod_test.o
-MD -MP -MF .deps/mknod_test.Tpo -c -o mknod_test.o mknod_test.cc
In file included from /usr/include/c++/11/x86_64-amazon-linux/bits/os_defines.h:39,
                 from /usr/include/c++/11/x86_64-amazon-linux/bits/c++config.h:2747,
                 from /usr/include/c++/11/cerrno:41,
                 from mknod_test.cc:21:
/usr/include/features.h:424:5: warning: #warning _FORTIFY_SOURCE > 2 is treated like 2 on this platform [-Wcpp]
 424 | # warning _FORTIFY_SOURCE > 2 is treated like 2 on this platform
| ~~~~~

mv -f .deps/mknod_test.Tpo .deps/mknod_test.Po
g++ -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_FORTIFY_SOURCE=3 -std=c++14 -g -O2 -o mknod_test mknod_test.o -ldl
g++ -DHAVE_CONFIG_H -I. -I. -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_FORTIFY_SOURCE=3 -std=c++14 -g -O2 -MT truncate_read_file.o
-MD -MP -MF .deps/truncate_read_file.Tpo -c -o truncate_read_file.o truncate_read_file.cc
In file included from /usr/include/c++/11/x86_64-amazon-linux/bits/os_defines.h:39,
                 from /usr/include/c++/11/x86_64-amazon-linux/bits/c++config.h:2747,
                 from /usr/include/c++/11/cstdio:41,
                 from truncate_read_file.cc:21:
/usr/include/features.h:424:5: warning: #warning _FORTIFY_SOURCE > 2 is treated like 2 on this platform [-Wcpp]
 424 | # warning _FORTIFY_SOURCE > 2 is treated like 2 on this platform
| ~~~~~

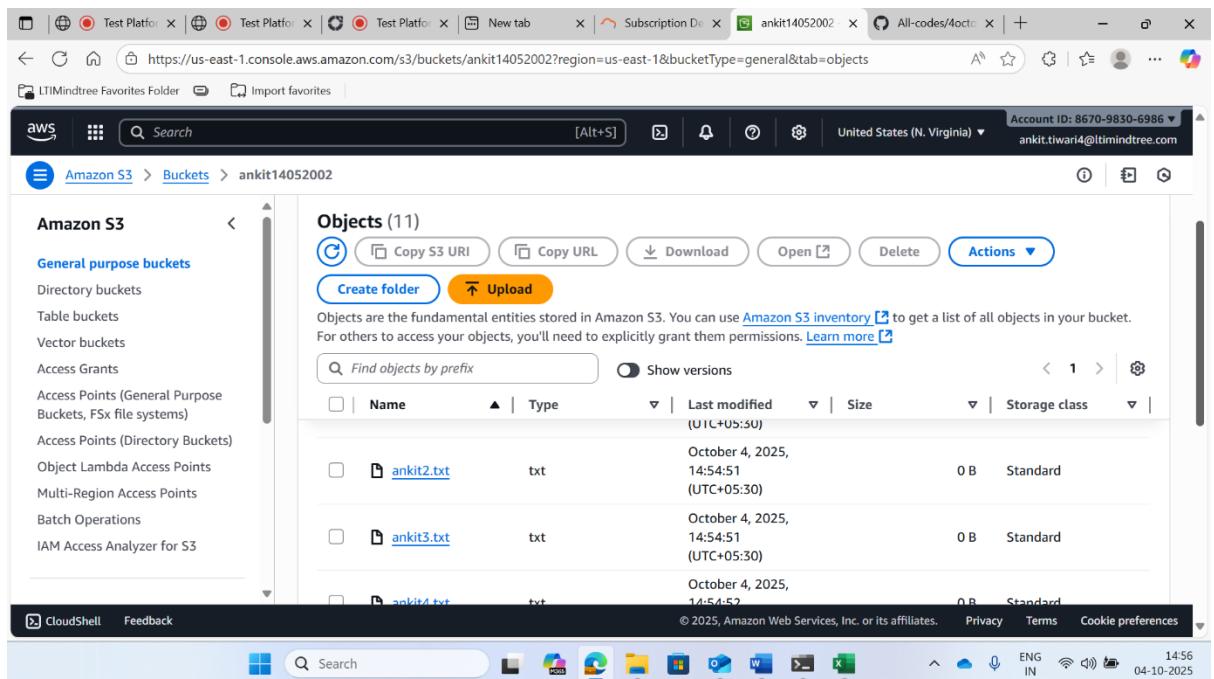
mv -f .deps/truncate_read_file.Tpo .deps/truncate_read_file.Po
g++ -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_FORTIFY_SOURCE=3 -std=c++14 -g -O2 -o truncate_read_file truncate_read_file.o
-lldl
g++ -DHAVE_CONFIG_H -I. -I. -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_FORTIFY_SOURCE=3 -std=c++14 -g -O2 -MT cr_filename.o
-MD -MP -MF .deps/cr_filename.Tpo -c -o cr_filename.o cr_filename.cc
In file included from /usr/include/c++/11/x86_64-amazon-linux/bits/os_defines.h:39,
                 from /usr/include/c++/11/x86_64-amazon-linux/bits/c++config.h:2747,
                 from /usr/include/c++/11/cstdio:41,
                 from cr_filename.cc:21:
```

->Giving access to my bucket via terminal



```
[root@ip-172-31-35-216 ~]# ll -d /etc/passwd-s3fs
-rw-r-----. 1 root root 0 Oct  4 09:22 /etc/passwd-s3fs
[root@ip-172-31-35-216 ~]# vim /etc/passwd-s3fs
[root@ip-172-31-35-216 ~]# s3fs ankit14052002 /mnt -o passwd_file=/etc/passwd-s3fs
[root@ip-172-31-35-216 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0  4.0M  0% /dev
tmpfs          453M   0  453M  0% /dev/shm
tmpfs          181M  432K 181M  1% /run
/dev/nvme0n1p1  8.0G  2.3G 5.7G 29% /
tmpfs          453M   0  453M  0% /tmp
/dev/nvme0n1p128 10M  1.3M  8.7M 13% /boot/efi
tmpfs          91M   0  91M  0% /run/user/1000
s3fs           64P   0  64P  0% /mnt
[root@ip-172-31-35-216 ~]# cd /mnt/
[root@ip-172-31-35-216 mnt]# ll
total 1
-rw-r--r--. 1 root root 199 Oct  4 08:56 links.txt
[root@ip-172-31-35-216 mnt]# touch ankit{1..10}.txt
[root@ip-172-31-35-216 mnt]# ll
total 1
-rw-r--r--. 1 root root 0 Oct  4 09:24 ankit1.txt
-rw-r--r--. 1 root root 0 Oct  4 09:24 ankit10.txt
-rw-r--r--. 1 root root 0 Oct  4 09:24 ankit2.txt
-rw-r--r--. 1 root root 0 Oct  4 09:24 ankit3.txt
-rw-r--r--. 1 root root 0 Oct  4 09:24 ankit4.txt
-rw-r--r--. 1 root root 0 Oct  4 09:24 ankit5.txt
-rw-r--r--. 1 root root 0 Oct  4 09:24 ankit6.txt
-rw-r--r--. 1 root root 0 Oct  4 09:24 ankit7.txt
-rw-r--r--. 1 root root 0 Oct  4 09:24 ankit8.txt
-rw-r--r--. 1 root root 0 Oct  4 09:24 ankit9.txt
-rw-r-----. 1 root root 199 Oct  4 08:56 links.txt
[root@ip-172-31-35-216 mnt]# |
```

→ files ankit.txt received on my bucket

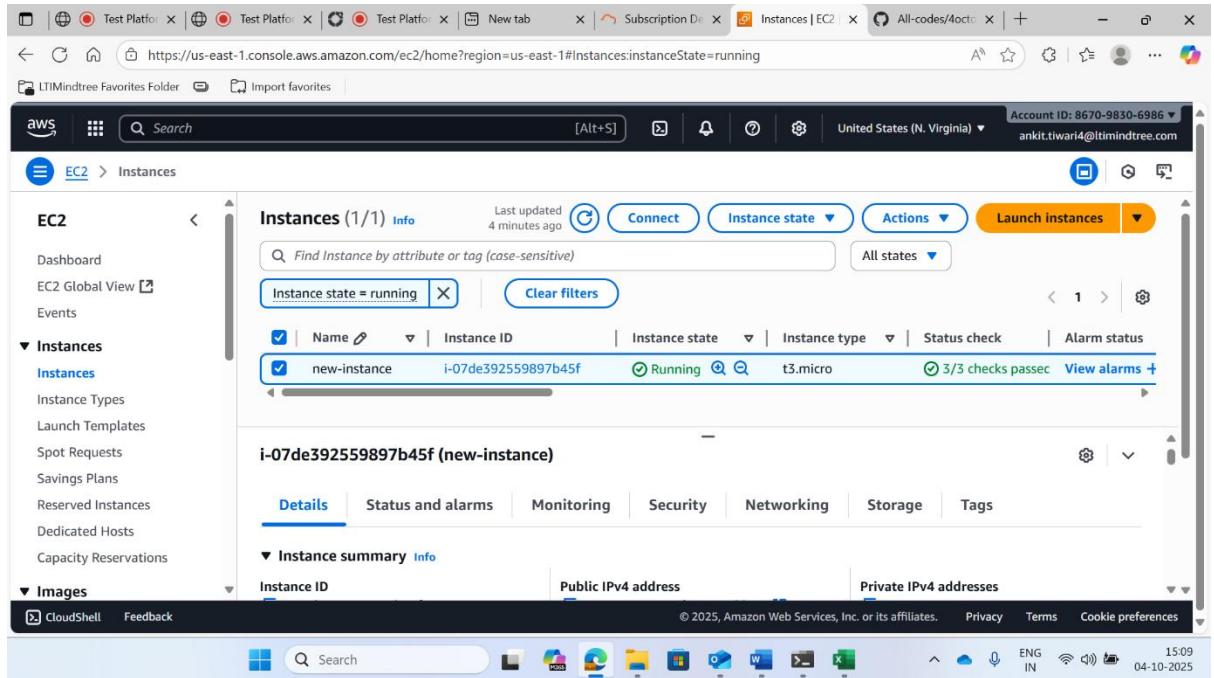


The screenshot shows the AWS S3 console interface. On the left, the navigation pane lists 'Amazon S3' and 'General purpose buckets'. The main area is titled 'Objects (11)' and shows a list of 11 text files named 'ankit1.txt' through 'ankit10.txt'. Each file is 0 B in size and has a 'Standard' storage class. The objects were last modified on October 4, 2025, at 14:54:51 (UTC+05:30). The interface includes standard S3 actions like 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', and 'Actions' dropdown.

Q2->I have one web server in North Virginia Region where my website is running. I need same server in ohio region. Migrate this web server from North virginia to ohio.

Ans :-

->Instance is created from ec2



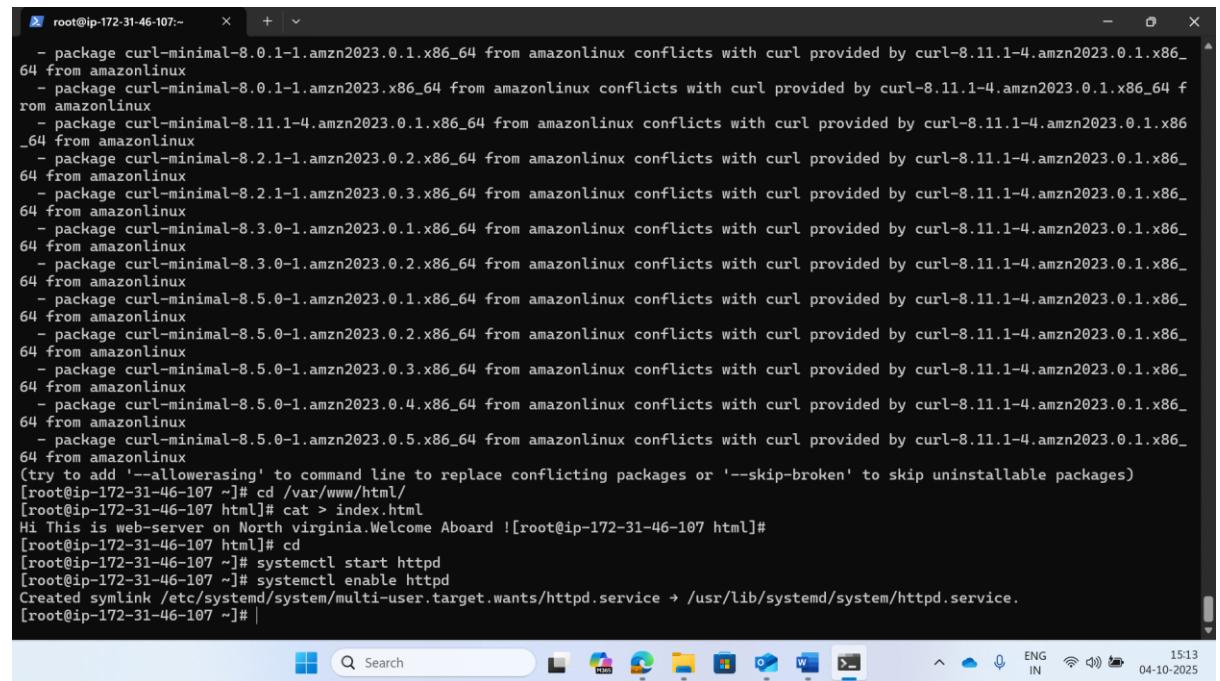
→ connecting to server in terminal via ssh

```
PS C:\Users\10844395> cd downloads
PS C:\Users\10844395\downloads> ssh -i "new-pair.pem" ec2-user@ec2-35-153-176-130.compute-1.amazonaws.com
The authenticity of host 'ec2-35-153-176-130.compute-1.amazonaws.com (35.153.176.130)' can't be established.
ED25519 key fingerprint is SHA256:PO+Ar0PtHfLDIJ9xT9NHhiF+IzdbtZ8k+ZdX0NDubk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-35-153-176-130.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_
~\_ ##### Amazon Linux 2023
~~\_ #####\
~~ \###|
~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~ \~' '=>
~~ \~' /_
~~ \~' /_
~~ \~' /_
[ec2-user@ip-172-31-46-107 ~]$ sudo su -
[ec2-user@ip-172-31-46-107 ~]$ yum install httpd -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
=====
Package           Architecture      Version       Repository      Size
=====
Installing:
httpd            x86_64          2.4.65-1.amzn2023.0.1    amazonlinux    47 k
Installing dependencies:
apr              x86_64          1.7.5-1.amzn2023.0.4    amazonlinux    129 k
apr-util         x86_64          1.6.3-1.amzn2023.0.1    amazonlinux    98 k
generic-logos-httpd    noarch        18.0.0-12.amzn2023.0.3  amazonlinux    19 k
httpd-core       x86_64          2.4.65-1.amzn2023.0.1    amazonlinux    1.4 M
httpd-filesystem    noarch        2.4.65-1.amzn2023.0.1    amazonlinux    13 k
httpd-tools       x86_64          2.4.65-1.amzn2023.0.1    amazonlinux    81 k

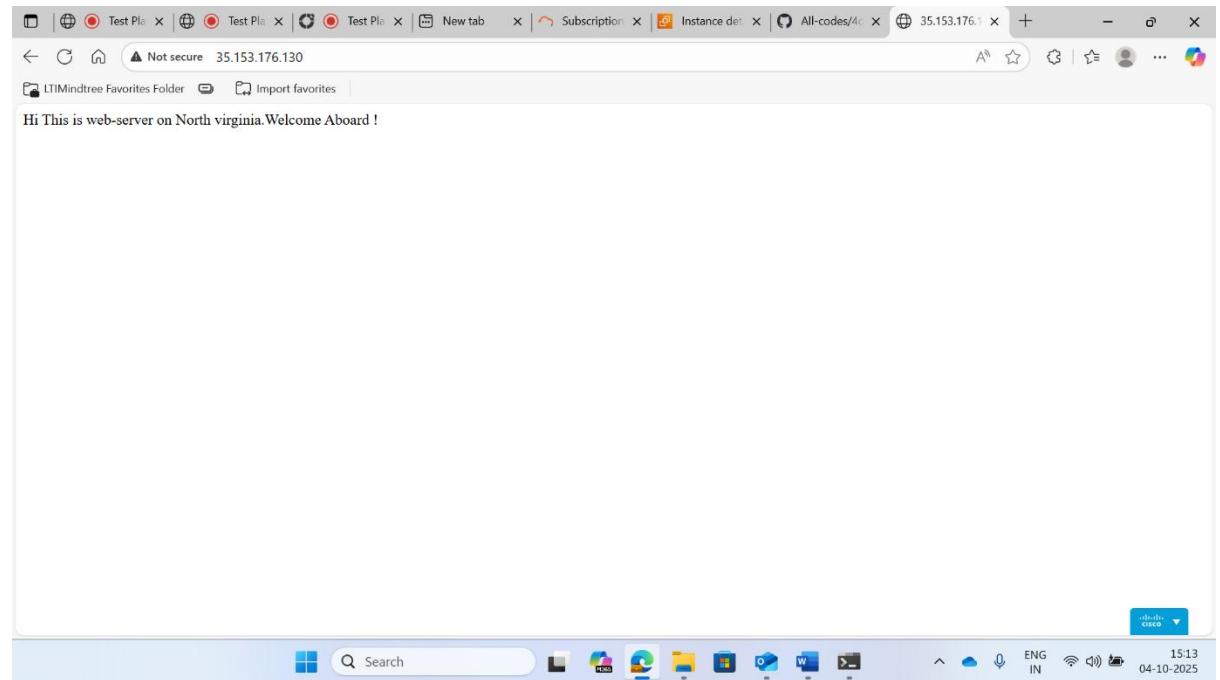
```

-> creating the web-server in North virginia

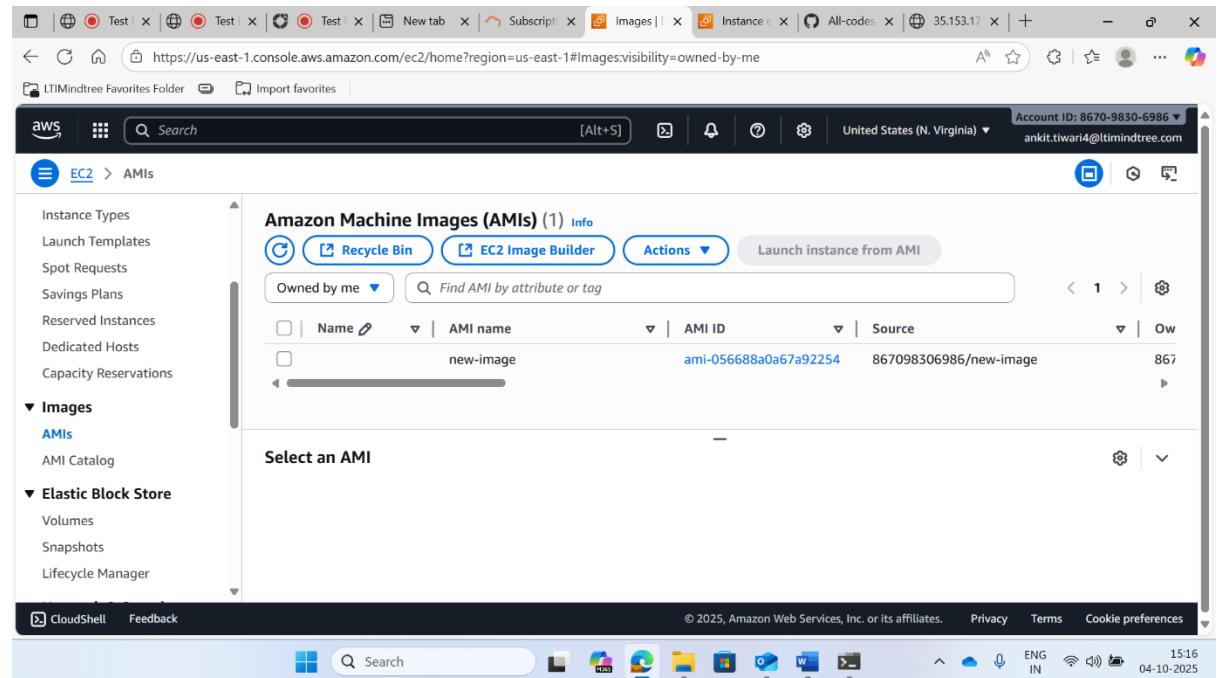


```
root@ip-172-31-46-107:~ x + v
- package curl-minimal-8.0.1-1.amzn2023.0.1.x86_64 from amazonlinux conflicts with curl provided by curl-8.11.1-4.amzn2023.0.1.x86_64 from amazonlinux
- package curl-minimal-8.0.1-1.amzn2023.x86_64 from amazonlinux conflicts with curl provided by curl-8.11.1-4.amzn2023.0.1.x86_64 from amazonlinux
- package curl-minimal-8.11.1-4.amzn2023.0.1.x86_64 from amazonlinux conflicts with curl provided by curl-8.11.1-4.amzn2023.0.1.x86_64 from amazonlinux
- package curl-minimal-8.2.1-1.amzn2023.0.2.x86_64 from amazonlinux conflicts with curl provided by curl-8.11.1-4.amzn2023.0.1.x86_64 from amazonlinux
- package curl-minimal-8.2.1-1.amzn2023.0.3.x86_64 from amazonlinux conflicts with curl provided by curl-8.11.1-4.amzn2023.0.1.x86_64 from amazonlinux
- package curl-minimal-8.3.0-1.amzn2023.0.1.x86_64 from amazonlinux conflicts with curl provided by curl-8.11.1-4.amzn2023.0.1.x86_64 from amazonlinux
- package curl-minimal-8.3.0-1.amzn2023.0.2.x86_64 from amazonlinux conflicts with curl provided by curl-8.11.1-4.amzn2023.0.1.x86_64 from amazonlinux
- package curl-minimal-8.5.0-1.amzn2023.0.1.x86_64 from amazonlinux conflicts with curl provided by curl-8.11.1-4.amzn2023.0.1.x86_64 from amazonlinux
- package curl-minimal-8.5.0-1.amzn2023.0.2.x86_64 from amazonlinux conflicts with curl provided by curl-8.11.1-4.amzn2023.0.1.x86_64 from amazonlinux
- package curl-minimal-8.5.0-1.amzn2023.0.3.x86_64 from amazonlinux conflicts with curl provided by curl-8.11.1-4.amzn2023.0.1.x86_64 from amazonlinux
- package curl-minimal-8.5.0-1.amzn2023.0.4.x86_64 from amazonlinux conflicts with curl provided by curl-8.11.1-4.amzn2023.0.1.x86_64 from amazonlinux
- package curl-minimal-8.5.0-1.amzn2023.0.5.x86_64 from amazonlinux conflicts with curl provided by curl-8.11.1-4.amzn2023.0.1.x86_64 from amazonlinux
[try to add '--allow-erasing' to command line to replace conflicting packages or '--skip-broken' to skip uninstallable packages]
[root@ip-172-31-46-107 ~]# cd /var/www/html/
[root@ip-172-31-46-107 html]# cat > index.html
Hi This is web-server on North virginia.Welcome Aboard !
[root@ip-172-31-46-107 html]# cd
[root@ip-172-31-46-107 ~]# systemctl start httpd
[root@ip-172-31-46-107 ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-46-107 ~]#
```

-> server reachable from Public ip of instance at North virginia

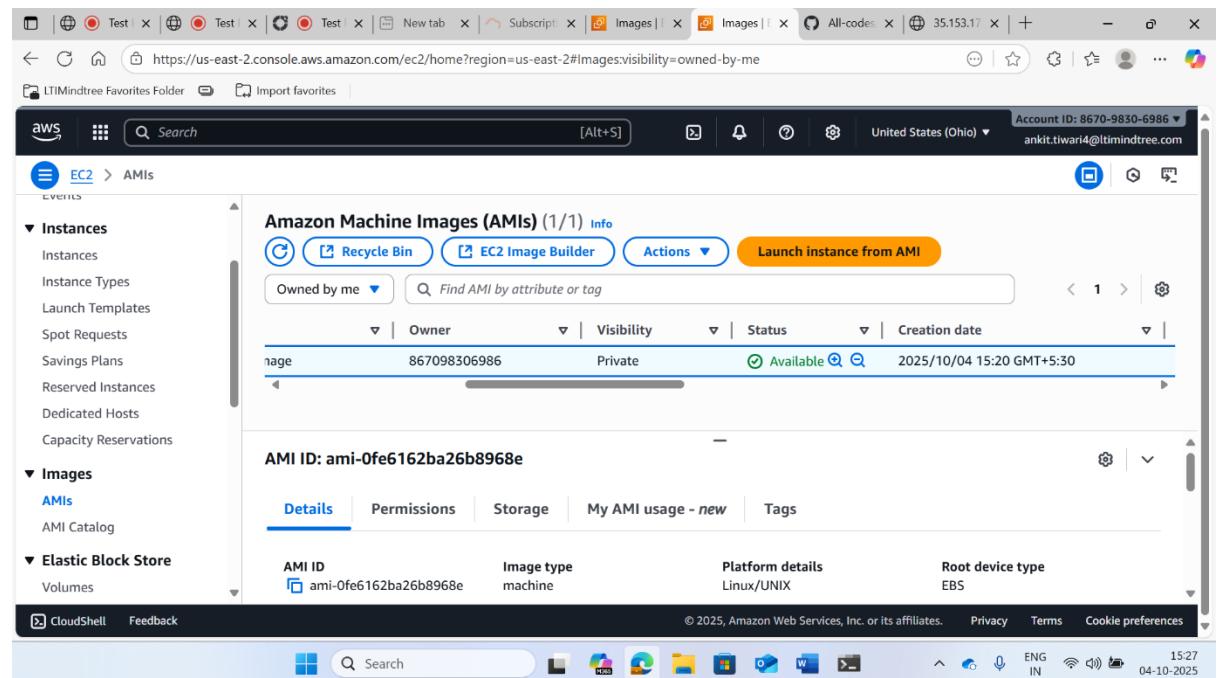


-> Trying to access the same server via AMI (created AMI in N.virginia)



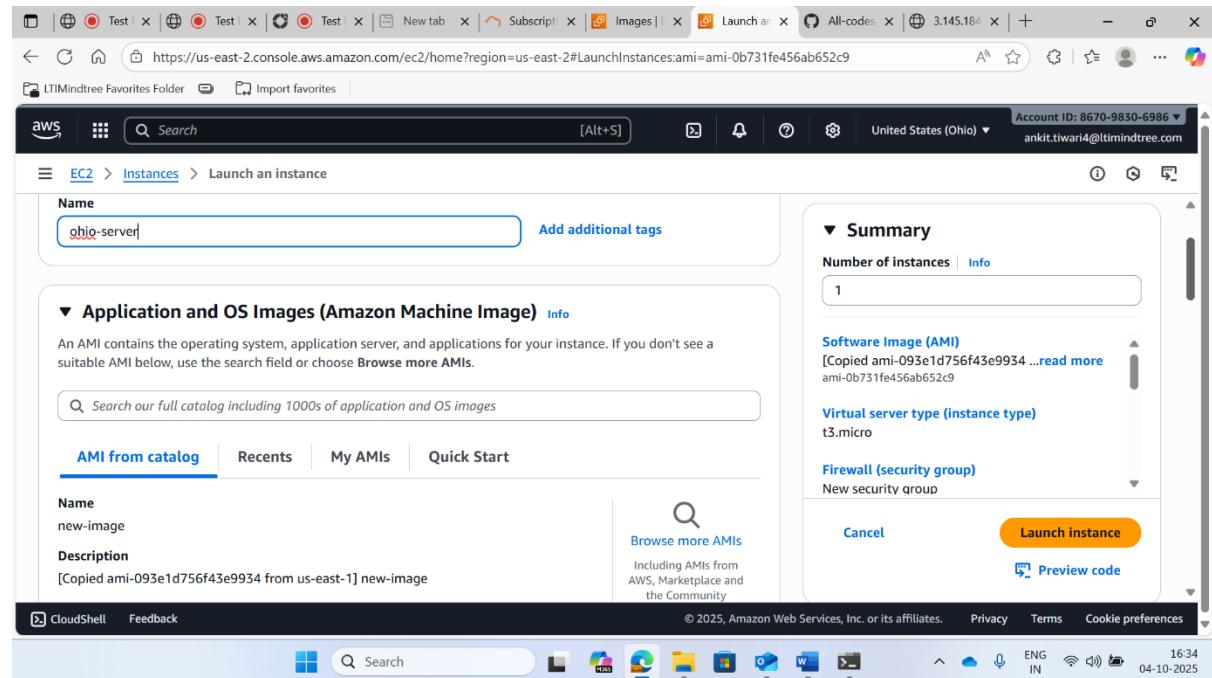
The screenshot shows the AWS EC2 AMIs page in the N.Virginia region. The left sidebar shows navigation options like Instance Types, Launch Templates, and Images. Under Images, the 'AMIs' tab is selected. The main content area displays a table titled 'Amazon Machine Images (AMIs) (1)'. The table has columns for Name, AMI name, AMI ID, and Source. One entry is listed: 'new-image' with AMI ID 'ami-056688a0a67a92254' and Source '867098306986/new-image'. Below the table is a section titled 'Select an AMI'.

-> copied the AMI into ohio region



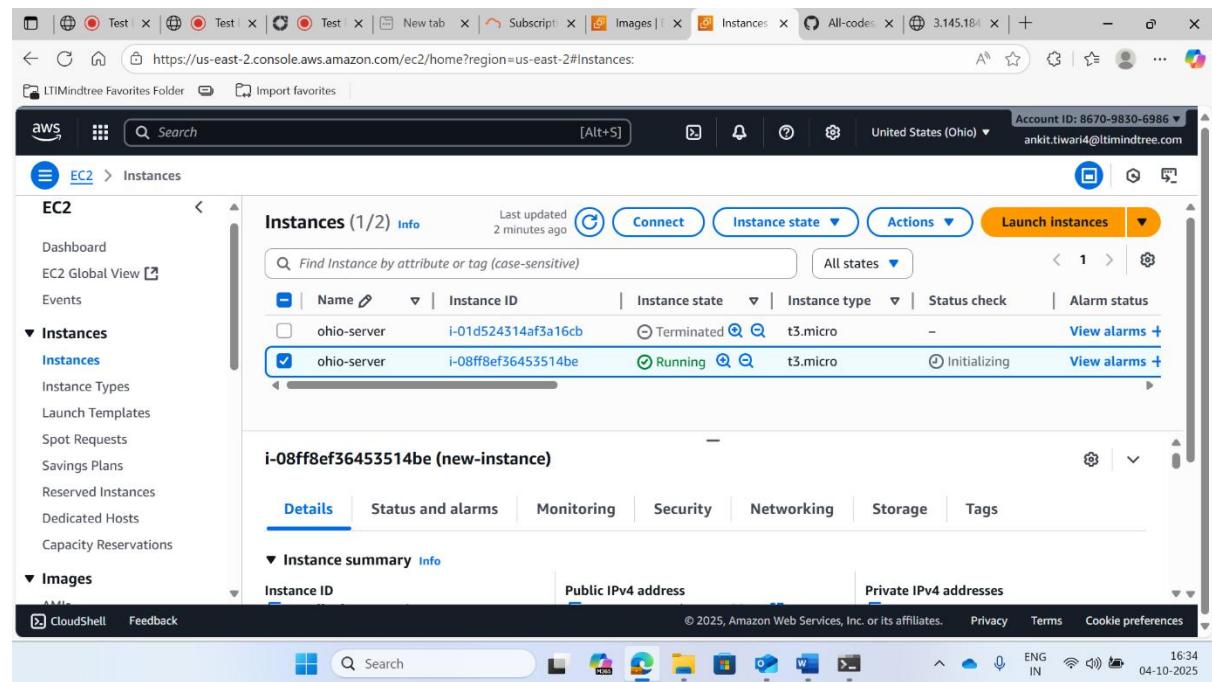
The screenshot shows the AWS EC2 AMIs page in the Ohio region. The left sidebar shows navigation options like Instances, AMIs, and Elastic Block Store. The main content area displays a table titled 'Amazon Machine Images (AMIs) (1/1)'. The table has columns for Owner, Visibility, Status, and Creation date. One entry is listed: 'new-image' with Owner '867098306986', Visibility 'Private', Status 'Available', and Creation date '2025/10/04 15:20 GMT+5:30'. Below the table is a section titled 'AMI ID: ami-0fe6162ba26b8968e' with tabs for Details, Permissions, Storage, My AMI usage - new, and Tags. The Details tab is selected, showing the AMI ID 'ami-0fe6162ba26b8968e', Image type 'machine', Platform details 'Linux/UNIX', and Root device type 'EBS'.

->Now launching instance from AMI



The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The 'Name' field is set to 'ohio-server'. The 'Software Image (AMI)' section shows a copied AMI: 'ami-093e1d756f43e9934'. The 'Virtual server type (instance type)' is set to 't3.micro'. The 'Firewall (security group)' is a new security group. The 'Launch instance' button is highlighted in orange. The browser status bar at the bottom shows the URL: https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstances:ami=ami-0b731fe456ab652c9.

-> instance created

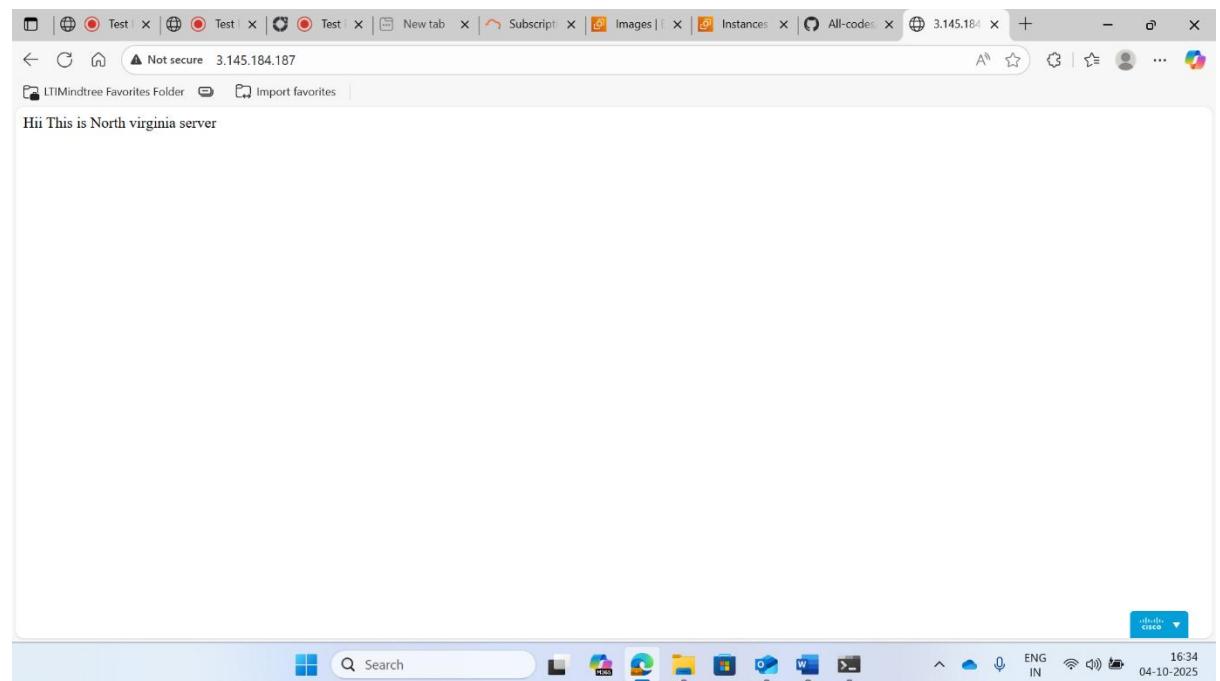


The screenshot shows the 'Instances' page in the AWS Management Console. The left sidebar shows 'Instances' selected. The main table lists two instances: 'ohio-server' (terminated) and 'ohio-server' (running). The 'running' instance is selected. The 'Details' tab is active for the selected instance. The instance summary table shows the instance ID 'i-08ff8ef36453514be' and its state 'Running'. The browser status bar at the bottom shows the URL: https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Instances:.

-> Trying to access North virginia server via ohio instance server

Public ip: (North Virginia) 54.175.217.10

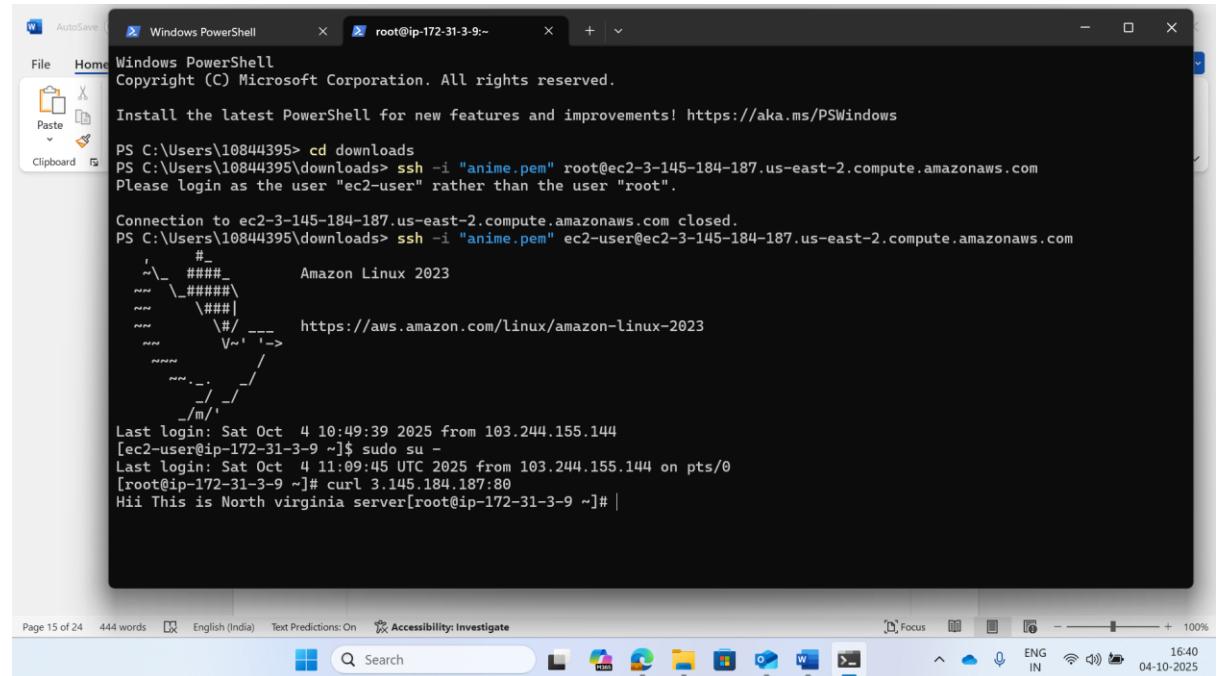
Public ip: (Ohio) 3.145.184.187



->Server created on North virginia instance

A screenshot of a Windows PowerShell window. The command run is 'docker run -d -p 80:80 nginx'. The output shows the container ID '4a3a2a2a2a2a2a2a2a2a2a2a2a2a2a2a2'. The system tray at the bottom right shows the date as 04-10-2025 and the time as 16:40.

->Accessible on ohio server instance



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

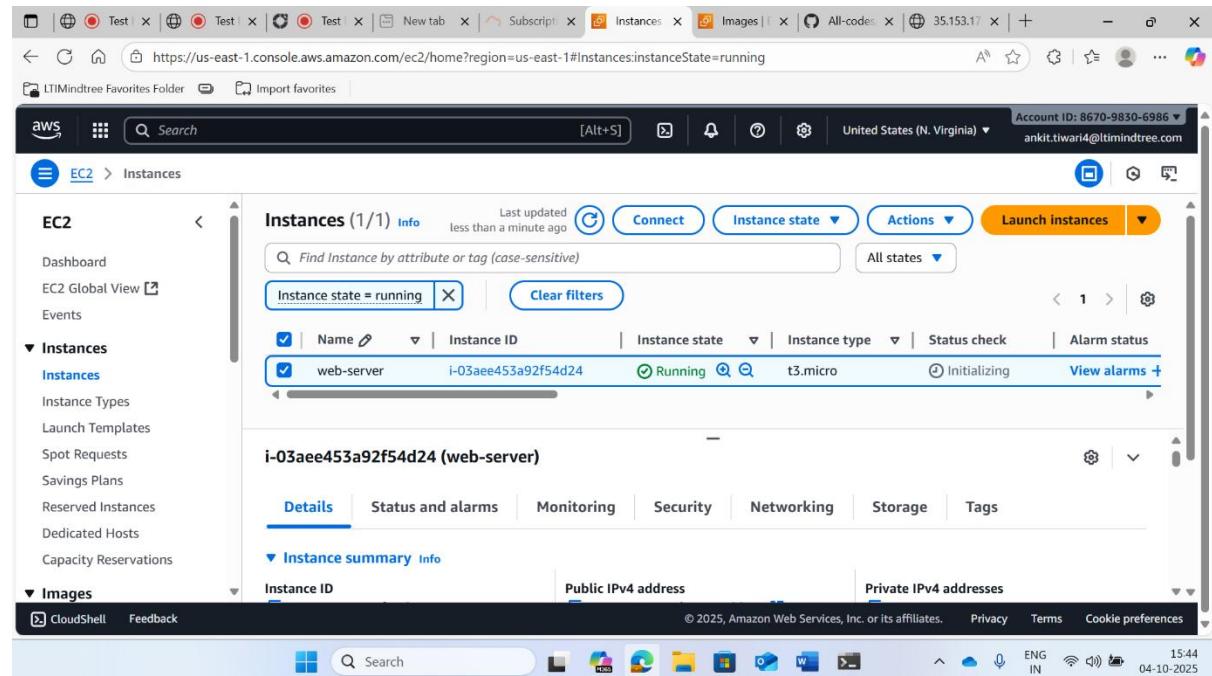
PS C:\Users\10844395> cd downloads
PS C:\Users\10844395\downloads> ssh -i "anime.pem" root@ec2-3-145-184-187.us-east-2.compute.amazonaws.com
Please login as the user "ec2-user" rather than the user "root".

Connection to ec2-3-145-184-187.us-east-2.compute.amazonaws.com closed.
PS C:\Users\10844395\downloads> ssh -i "anime.pem" ec2-user@ec2-3-145-184-187.us-east-2.compute.amazonaws.com
      _#
     ~\_\###_      Amazon Linux 2023
    ~~ \###\_
    ~~ \###|
    ~~  \#/   https://aws.amazon.com/linux/amazon-linux-2023
    ~~   \~'-->
    ~~   /
    ~~-.-/-
    _/`-/
    _/m/`|Last login: Sat Oct  4 10:49:39 2025 from 103.244.155.144
[ec2-user@ip-172-31-3-9 ~]$ sudo su -
Last login: Sat Oct  4 11:09:45 UTC 2025 from 103.244.155.144 on pts/0
[root@ip-172-31-3-9 ~]# curl 3.145.184.187:80
Hii This is North virginia server[root@ip-172-31-3-9 ~]# |
```

Q3->Configure web server on it and make it live. and attach a 5gb ebs volume. where create 10 files. like training.txt

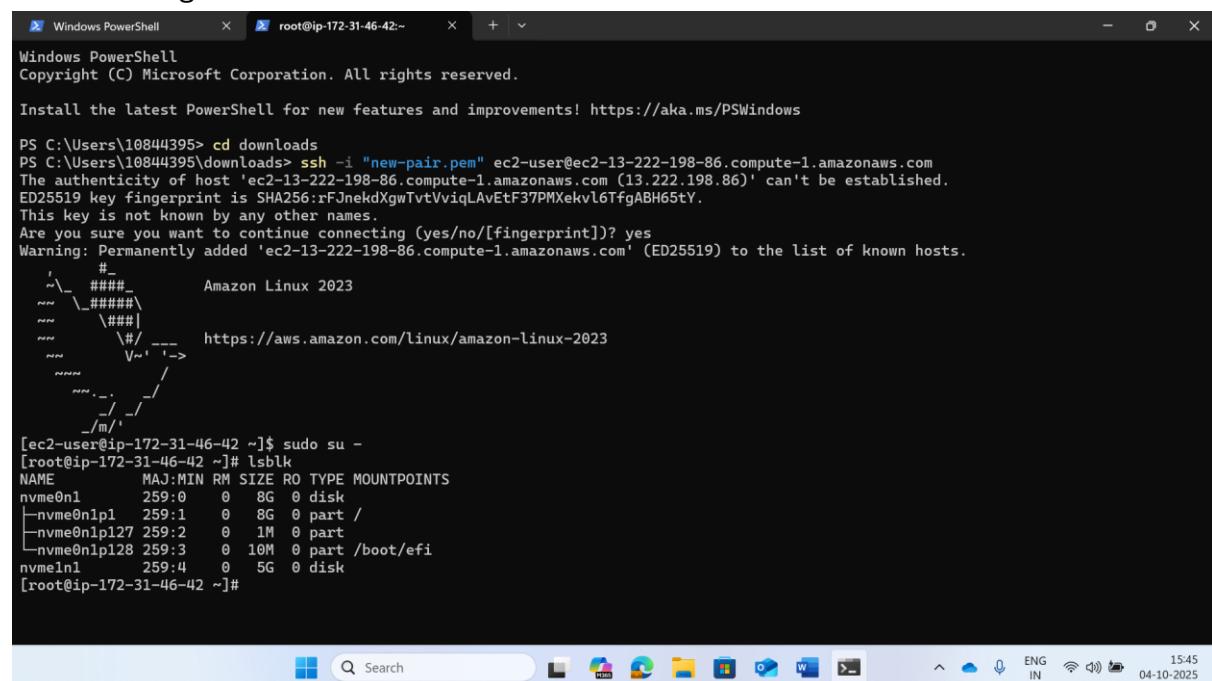
Ans->

-> creating an instance in EC2



The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with links for Dashboard, EC2 Global View, Events, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), and Images. The main content area shows a table for 'Instances (1/1)'. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. One row is selected, showing 'web-server' with Instance ID 'i-03aee453a92f54d24', State 'Running', Type 't3.micro', Status 'Initializing', and an 'View alarms' link. Below the table, there's a section for 'i-03aee453a92f54d24 (web-server)' with tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Details' tab is selected. At the bottom of the page is a navigation bar with CloudShell and Feedback.

->connecting to terminal via ssh



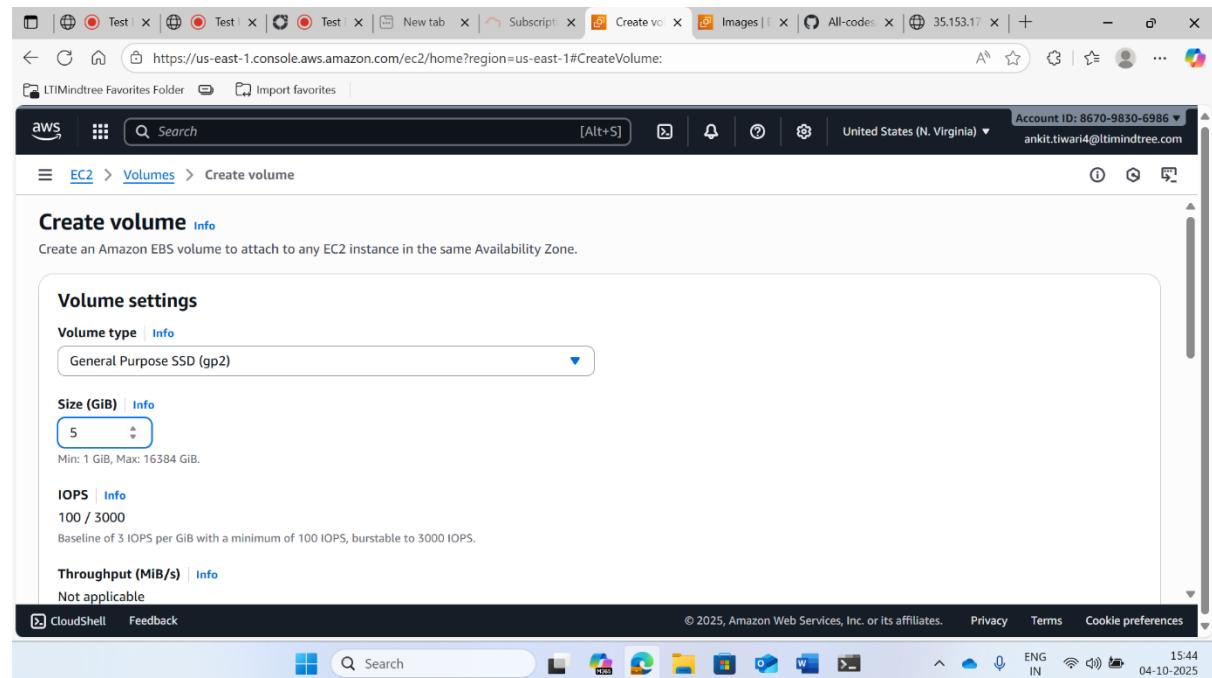
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10844395> cd downloads
PS C:\Users\10844395\downloads> ssh -i "new-pair.pem" ec2-user@ec2-13-222-198-86.compute-1.amazonaws.com
The authenticity of host 'ec2-13-222-198-86.compute-1.amazonaws.com (13.222.198.86)' can't be established.
ED25519 key fingerprint is SHA256:rFJnekdXgwTvtViqLavEtF37PMXekvl6TfgABH65tY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-222-198-86.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

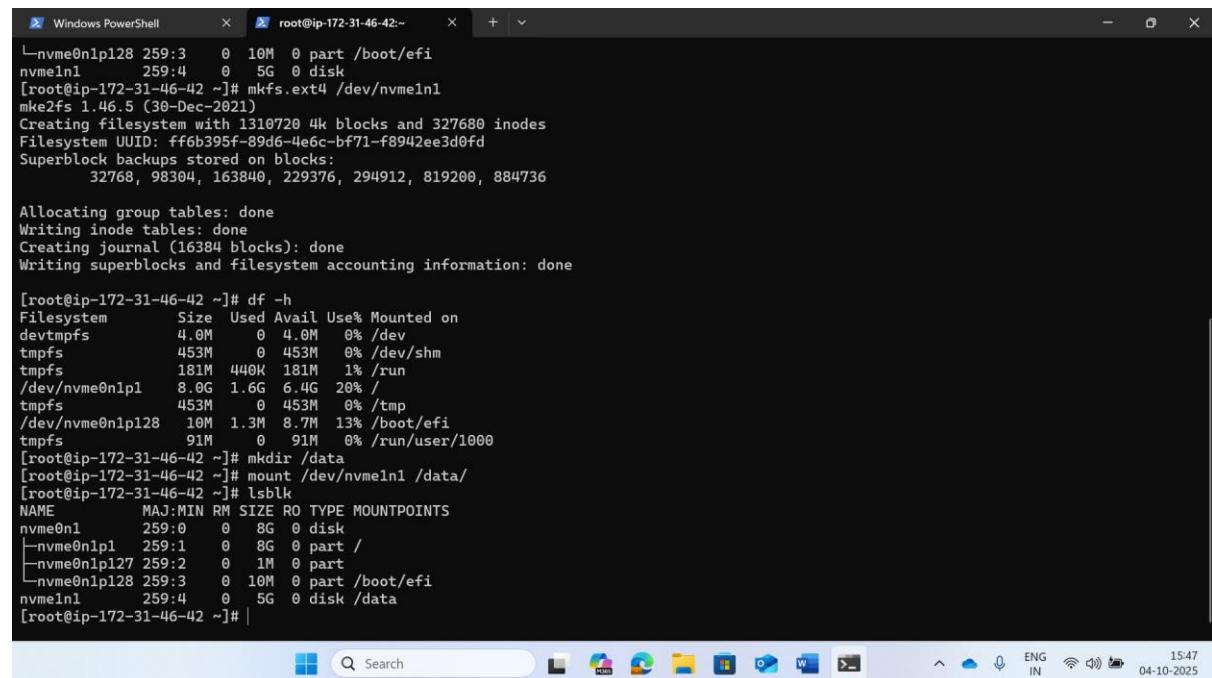
  _#
 /_###_      Amazon Linux 2023
  \_#####
   \##|
   \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
    \~' '---/
     \_/
      /_/
       /_/
        /_/
         /_/
          /_/
          /_m/ [ec2-user@ip-172-31-46-42 ~]$ sudo su -
[root@ip-172-31-46-42 ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
nvme0n1   259:0   0   8G  0 disk
└─nvme0n1p1 259:1   0   8G  0 part /
nvme0n1p127 259:2   0   1M  0 part
└─nvme0n1p128 259:3   0 10M  0 part /boot/efi
nvme1n1   259:4   0   5G  0 disk
[root@ip-172-31-46-42 ~]#
```

-> assigning ebs volume to our instance



The screenshot shows the AWS CloudShell interface with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateVolume>. The page is titled 'Create volume' and shows 'Volume settings' for a 'General Purpose SSD (gp2)' volume. The size is set to 5 GiB, IOPS to 100 / 3000, and throughput to 'Not applicable'. The CloudShell tab is active at the bottom.

-> creating file system and mounting

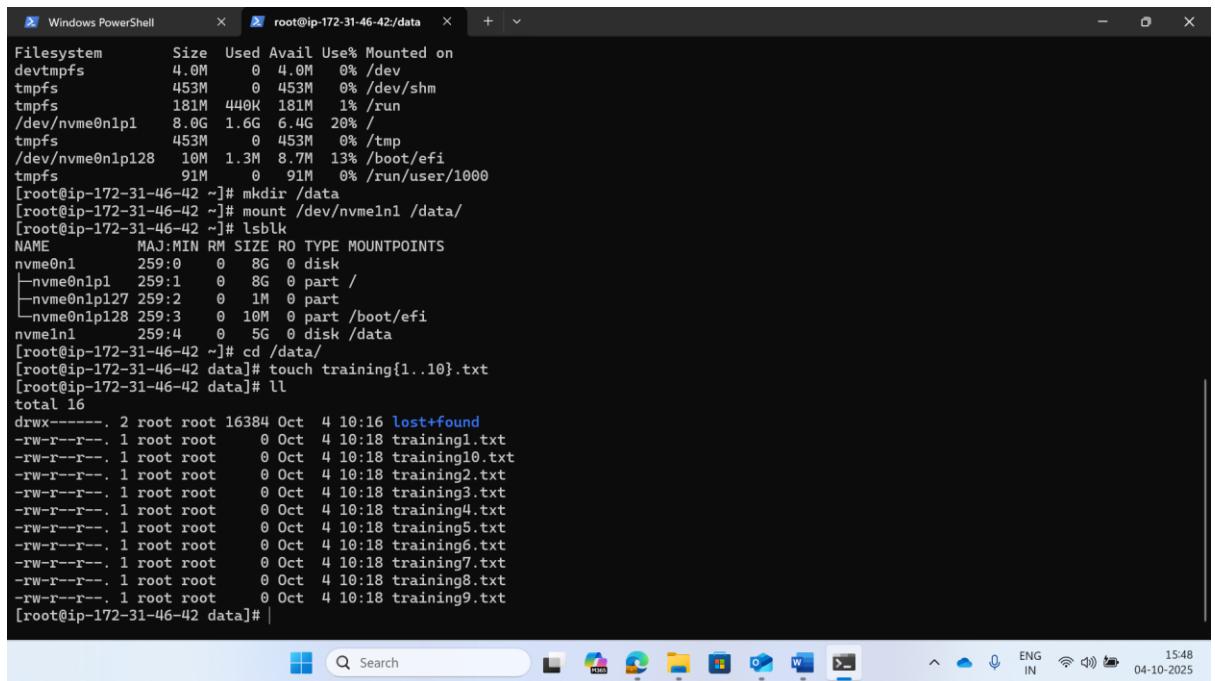


```
Windows PowerShell
root@ip-172-31-46-42:~# mkfs.ext4 /dev/nvme1n1
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 1310720 4k blocks and 327680 inodes
Filesystem UUID: ff6b395f-89d6-4e6c-bf71-f8942ee3d0fd
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

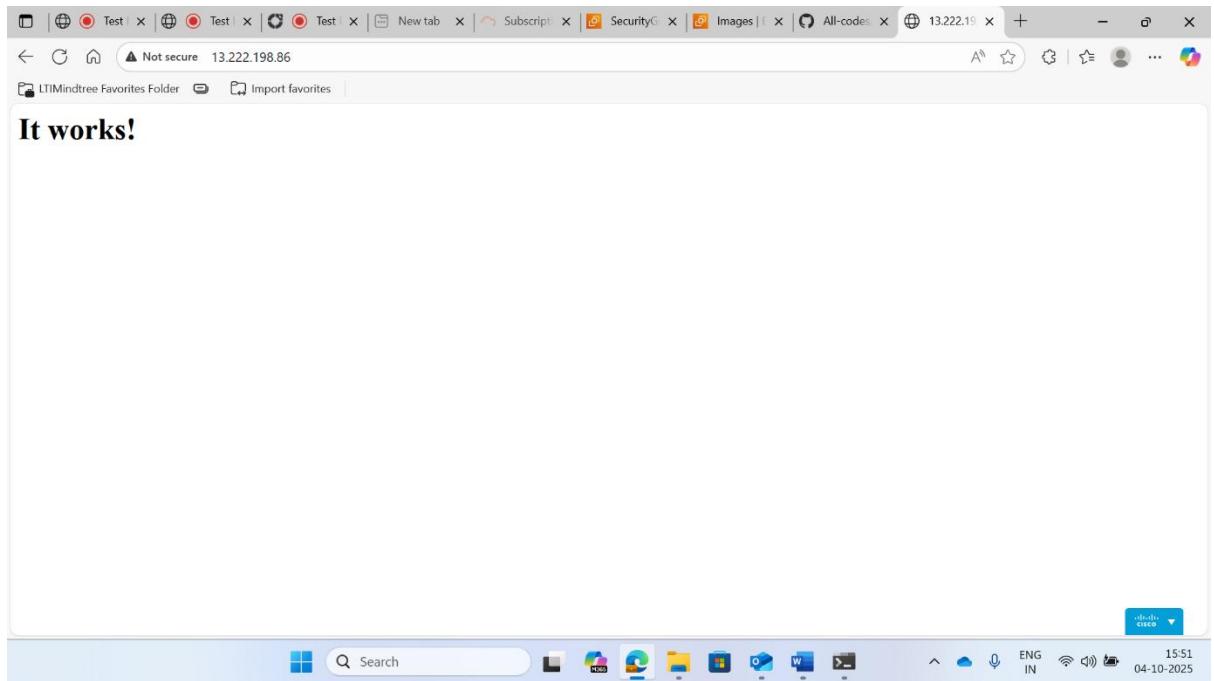
[root@ip-172-31-46-42:~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0M  4.0M  0% /dev
tmpfs          453M   0M  453M  0% /dev/shm
tmpfs          181M  440K 181M  1% /run
/dev/nvme0n1p1   8.0G  1.6G  6.4G  20% /
tmpfs          453M   0M  453M  0% /tmp
/dev/nvme0n1p128 10M  1.3M  8.7M  13% /boot/efi
tmpfs          91M   0M  91M  0% /run/user/1000
[root@ip-172-31-46-42:~]# mkdir /data
[root@ip-172-31-46-42:~]# mount /dev/nvme1n1 /data
[root@ip-172-31-46-42:~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
nvme0n1    259:0    0   8G  0 disk
└─nvme0n1p1 259:1    0   8G  0 part /
nvme0n1p127 259:2    0   1M  0 part
└─nvme0n1p128 259:3    0 10M  0 part /boot/efi
nvme1n1    259:4    0   5G  0 disk /data
[root@ip-172-31-46-42:~]# |
```

➔ files training.txt have been created



```
Windows PowerShell      root@ip-172-31-46-42:/data      + | 
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0  4.0M  0% /dev
tmpfs          453M   0  453M  0% /dev/shm
tmpfs          181M 440K 181M  1% /run
/dev/nvme0n1p1   8.0G  1.6G  6.4G  20% /
tmpfs          453M   0  453M  0% /tmp
/dev/nvme0n1p128 10M  1.3M  8.7M  13% /boot/efi
tmpfs          91M   0  91M  0% /run/user/1000
[root@ip-172-31-46-42 ~]# mkdir /data
[root@ip-172-31-46-42 ~]# mount /dev/nvme0n1 /data/
[root@ip-172-31-46-42 ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
nvme0n1    259:0   0   8G  0 disk
└─nvme0n1p1 259:1   0   8G  0 part /
nvme0n1p127 259:2   0   1M  0 part
└─nvme0n1p128 259:3   0  10M  0 part /boot/efi
nvme0n1    259:4   0   5G  0 disk /data
[root@ip-172-31-46-42 ~]# cd /data/
[root@ip-172-31-46-42 data]# touch training{1..10}.txt
[root@ip-172-31-46-42 data]# ll
total 16
drwxr-x---. 2 root root 16384 Oct  4 10:16 lost+found
-rw-r--r--. 1 root root    0 Oct  4 10:18 training1.txt
-rw-r--r--. 1 root root    0 Oct  4 10:18 training10.txt
-rw-r--r--. 1 root root    0 Oct  4 10:18 training2.txt
-rw-r--r--. 1 root root    0 Oct  4 10:18 training3.txt
-rw-r--r--. 1 root root    0 Oct  4 10:18 training4.txt
-rw-r--r--. 1 root root    0 Oct  4 10:18 training5.txt
-rw-r--r--. 1 root root    0 Oct  4 10:18 training6.txt
-rw-r--r--. 1 root root    0 Oct  4 10:18 training7.txt
-rw-r--r--. 1 root root    0 Oct  4 10:18 training8.txt
-rw-r--r--. 1 root root    0 Oct  4 10:18 training9.txt
[root@ip-172-31-46-42 data]# |
```

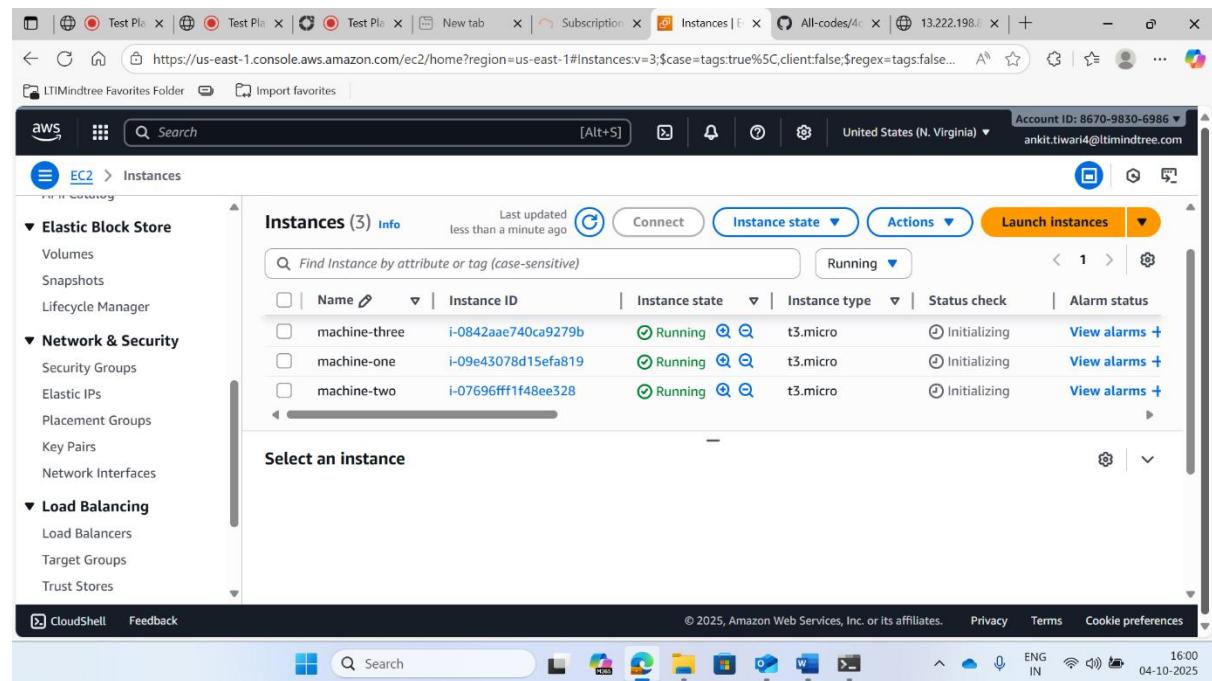
➔ the server is live as you can see



Q4-> We people are working on a common project in a same region. But my servers are in different zones. So, I want to share project information with everyone simultaneously. Configure efs storage it should be mount on every server.

Ans->

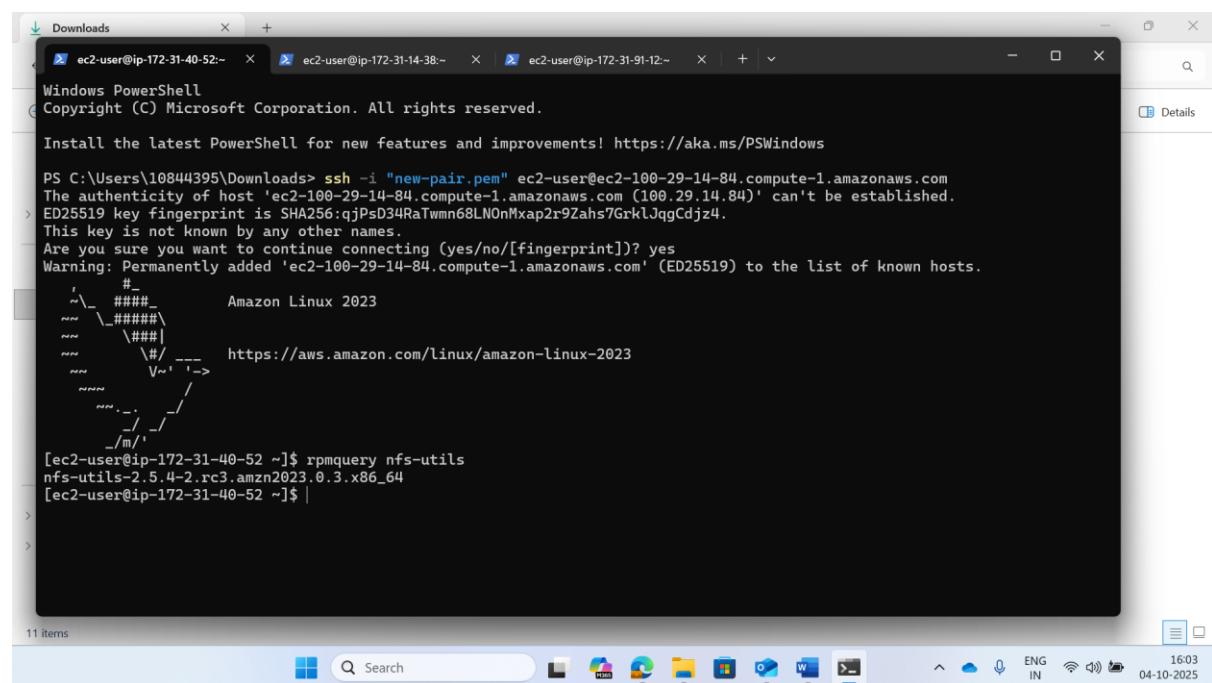
-> created three different instances on different zones but same region



The screenshot shows the AWS EC2 Instances page. The left sidebar has sections for Elastic Block Store, Network & Security, and Load Balancing. The main area displays three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
machine-three	i-0842aae740ca9279b	Running	t3.micro	Initializing	View alarms
machine-one	i-09e43078d15efa819	Running	t3.micro	Initializing	View alarms
machine-two	i-07696fff1f48ee328	Running	t3.micro	Initializing	View alarms

-> connected all the terminals



The screenshot shows a Windows terminal window with three tabs, each representing a different EC2 instance:

- Tab 1: ec2-user@ip-172-31-40-52:~
- Tab 2: ec2-user@ip-172-31-14-38:~
- Tab 3: ec2-user@ip-172-31-91-12:~

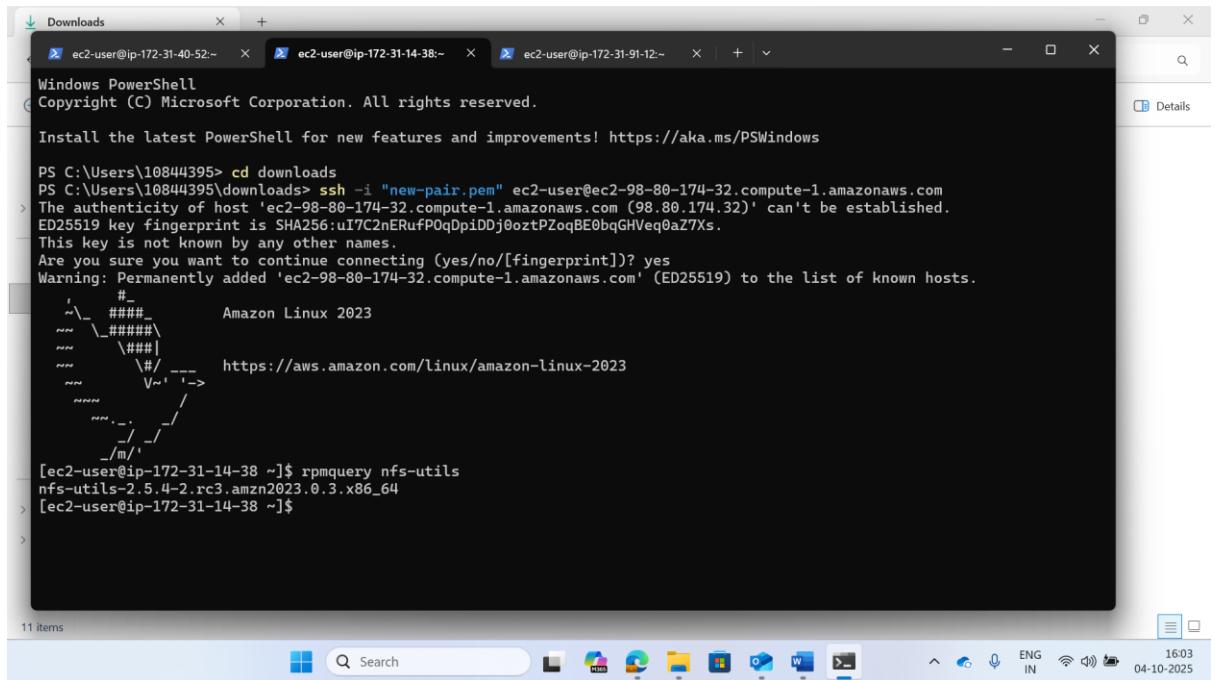
The terminal output for the first tab shows the user connecting via SSH:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10844395\Downloads> ssh -i "new-pair.pem" ec2-user@ec2-100-29-14-84.compute-1.amazonaws.com
The authenticity of host 'ec2-100-29-14-84.compute-1.amazonaws.com (100.29.14.84)' can't be established.
> ED25519 key fingerprint is SHA256:qjPsD34RaTwmn68LNOnMxap2r9Zahs7GrkLJqgCdjz4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-100-29-14-84.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
```

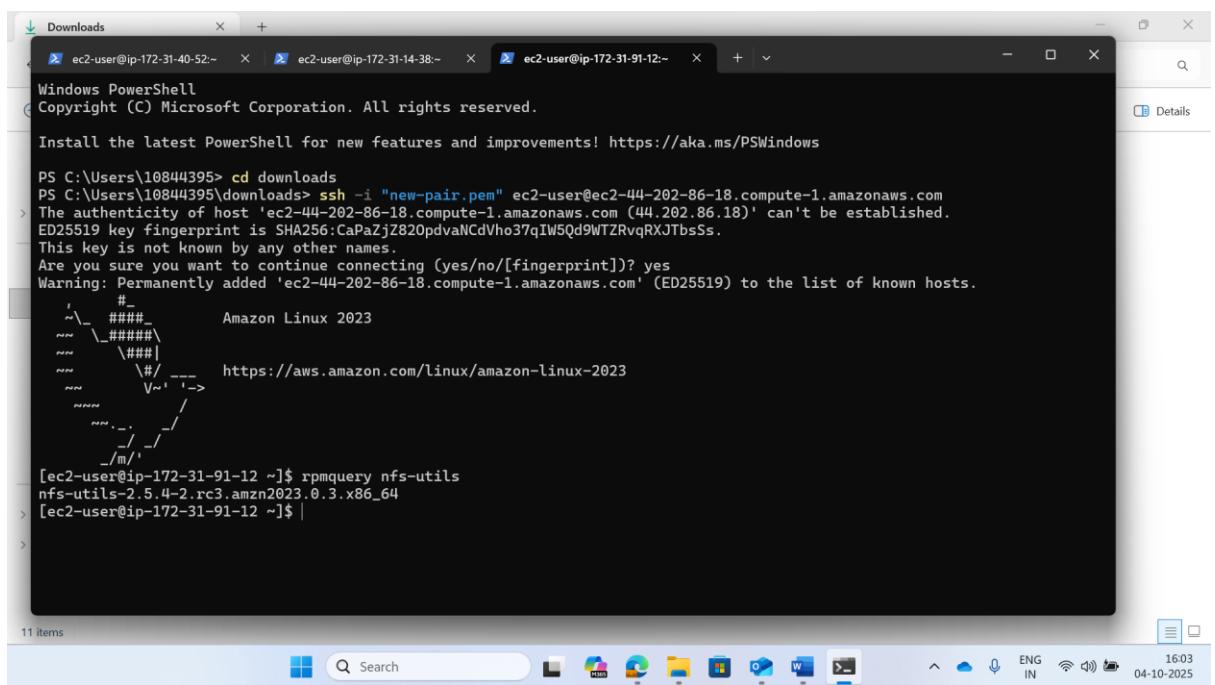
Below the terminal, the taskbar shows the Windows Start button, a search bar, and various pinned icons.



```
PS C:\Users\10844395\Downloads> ssh -i "new-pair.pem" ec2-user@ec2-98-80-174-32.compute-1.amazonaws.com
The authenticity of host 'ec2-98-80-174-32.compute-1.amazonaws.com (98.80.174.32)' can't be established.
ED25519 key fingerprint is SHA256:uI7C2nErufP0qDpiDDj0oZtPZoqBE0bqGHVeq0aZ7Xs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-98-80-174-32.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

' _#
~\_\_ #####_ Amazon Linux 2023
~~ \_\#\#\#\_
~~ \#\#\#
~~ \#/ --- https://aws.amazon.com/linux/amazon-linux-2023
~~ V~' '-->
~~~ /
~~~ .-.
~~~ /_/
~/m/'

[ec2-user@ip-172-31-14-38 ~]$ rpmquery nfs-utils
nfs-utils-2.5.4-2.rc3.amzn2023.0.3.x86_64
[ec2-user@ip-172-31-14-38 ~]$
```

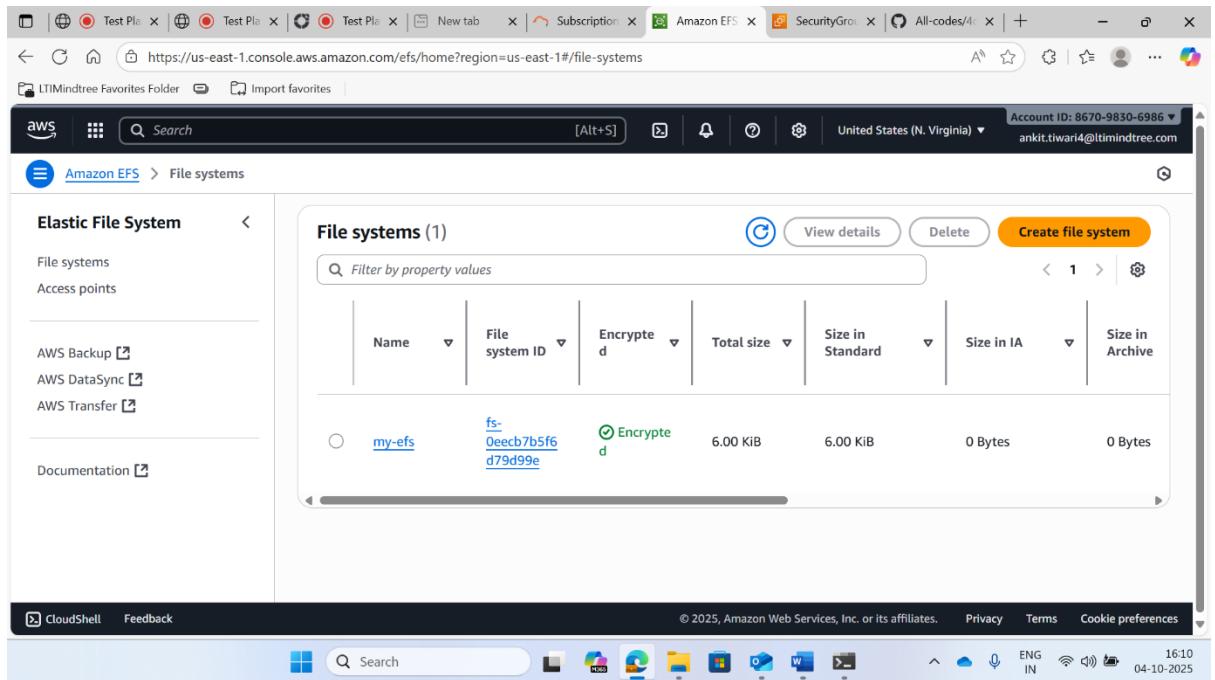


```
PS C:\Users\10844395\Downloads> ssh -i "new-pair.pem" ec2-user@ec2-44-202-86-18.compute-1.amazonaws.com
The authenticity of host 'ec2-44-202-86-18.compute-1.amazonaws.com (44.202.86.18)' can't be established.
ED25519 key fingerprint is SHA256:CaPaZjz820pdvalCdVh037qIW5Qd9WTZRvqRXJTsSs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-202-86-18.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

' _#
~\_\_ #####_ Amazon Linux 2023
~~ \_\#\#\#\_
~~ \#\#\#
~~ \#/ --- https://aws.amazon.com/linux/amazon-linux-2023
~~ V~' '-->
~~~ /
~~~ .-.
~~~ /_/
~/m/'

[ec2-user@ip-172-31-91-12 ~]$ rpmquery nfs-utils
nfs-utils-2.5.4-2.rc3.amzn2023.0.3.x86_64
[ec2-user@ip-172-31-91-12 ~]$
```

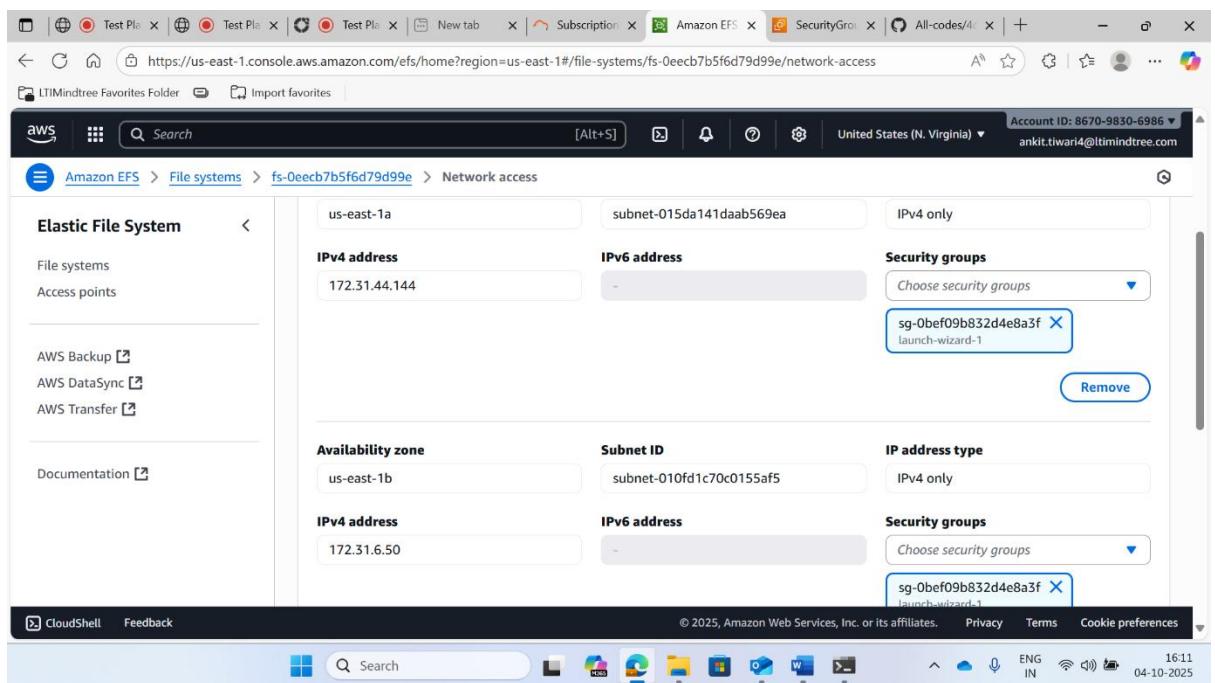
->creating EFS file system



File systems (1)

Name	File system ID	Encrypted	Total size	Size in Standard	Size in IA	Size in Archive
my-efs	fs-Oeeccb7b5f6d79d99e	Encrypted	6.00 KiB	6.00 KiB	0 Bytes	0 Bytes

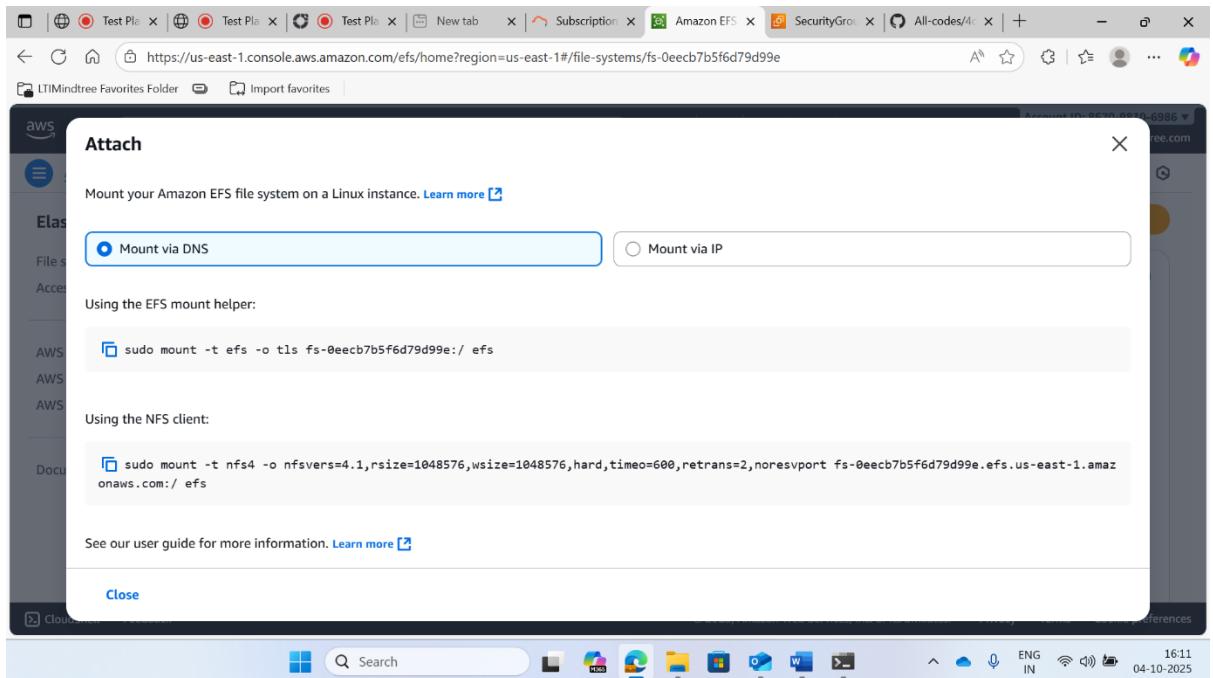
-> managing network in efs adding zones 1a,1b,1c



File systems (1)

Name	File system ID	Encrypted	Total size	Size in Standard	Size in IA	Size in Archive
my-efs	fs-Oeeccb7b5f6d79d99e	Encrypted	6.00 KiB	6.00 KiB	0 Bytes	0 Bytes

→ Attaching Dns mount



→ Mounting on terminal and creating files on 1st server

```
root@ip-172-31-40-52:/data1 ~ % rpmquery nfs-utils
nfs-utils-2.5.4-2.rc3.amzn2023.0.3.x86_64
root@ip-172-31-40-52 ~ % sudo su -
[root@ip-172-31-40-52 ~]# mkdir /data1
[root@ip-172-31-40-52 ~]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-0eeccb7b5f6d79d99e.efs.us-east-1.amazonaws.com:/ /data1
[root@ip-172-31-40-52 ~]# cd /data1
[root@ip-172-31-40-52 data1]# touch ankit{1..10}.txt
[root@ip-172-31-40-52 data1]# ll
total 40
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit1.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit10.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit2.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit3.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit4.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit5.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit6.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit7.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit8.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit9.txt
```

→ Adding NFS security group port 2049

Inbound rules [Info](#)

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

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-0f4c9c5fd4e021b00	HTTP	TCP	80	Cu... Info	<input type="text" value="0.0.0.0/0"/> X	Delete
sgr-0e8220ce021f10253	SSH	TCP	22	Cu... Info	<input type="text" value="0.0.0.0/0"/> X	Delete
sgr-0bfe48e3fe76f9121	NFS	TCP	2049	Cu... Info	<input type="text" value="0.0.0.0/0"/> X	Delete

[Add rule](#)

CloudShell [Feedback](#) © 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

16:13 04-10-2025

-> Files accessible on other servers

→ Machine2

```
root@ip-172-31-40-52:/data1  x  root@ip-172-31-14-38:/data2  x  root@ip-172-31-91-12:/data3  x  +  -
root@ip-172-31-14-38 ~]# rpmquery nfs-utils
nfs-utils-2.5.4-2.rc3.amzn2023.0.3.x86_64
[ec2-user@ip-172-31-14-38 ~]# sudo su -
[root@ip-172-31-14-38 ~]# mkdir /data2
[root@ip-172-31-14-38 ~]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wszie=1048576,hard,timeout=600,retrans=2,noresvport fs-0eeccb7
b5fd679d99e.efs.us-east-1.amazonaws.com:/ /data2
[root@ip-172-31-14-38 ~]# cd /data2
[root@ip-172-31-14-38 data2]# ll
total 40
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit1.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit10.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit2.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit3.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit4.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit5.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit6.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit7.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit8.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit9.txt
[root@ip-172-31-14-38 data2]# |
```

→ Machine 3

```
root@ip-172-31-40-52:/data1  X  root@ip-172-31-14-38:/data2  X  root@ip-172-31-91-12:/data3  X  +  -
#_
#_ \_ #####_      Amazon Linux 2023
#_ \_ #####\_
#_ \###_
#_ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
#_ V~` ' ->
#_ /_
#_ /_ /_
#_ /_ /_
[ec2-user@ip-172-31-91-12 ~]$ rpmquery nfs-utils
nfs-utils-2.5.4-2.rock3.amzn2023.0.3.x86_64
[ec2-user@ip-172-31-91-12 ~]$ sudo su -
[root@ip-172-31-91-12 ~]# mkdir /data3
[root@ip-172-31-91-12 ~]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wszie=1048576,hard,timeo=600,retrans=2,noresvport fs-0eeccb7
b5f6d79d99e.ebs.us-east-1.amazonaws.com:/ /data3
[root@ip-172-31-91-12 ~]# cd /data3
[root@ip-172-31-91-12 data3]# ll
total 40
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit1.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit10.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit2.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit3.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit4.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit5.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit6.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit7.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit8.txt
-rw-r--r--. 1 root root 0 Oct  4 10:39 ankit9.txt
[root@ip-172-31-91-12 data3]#
```