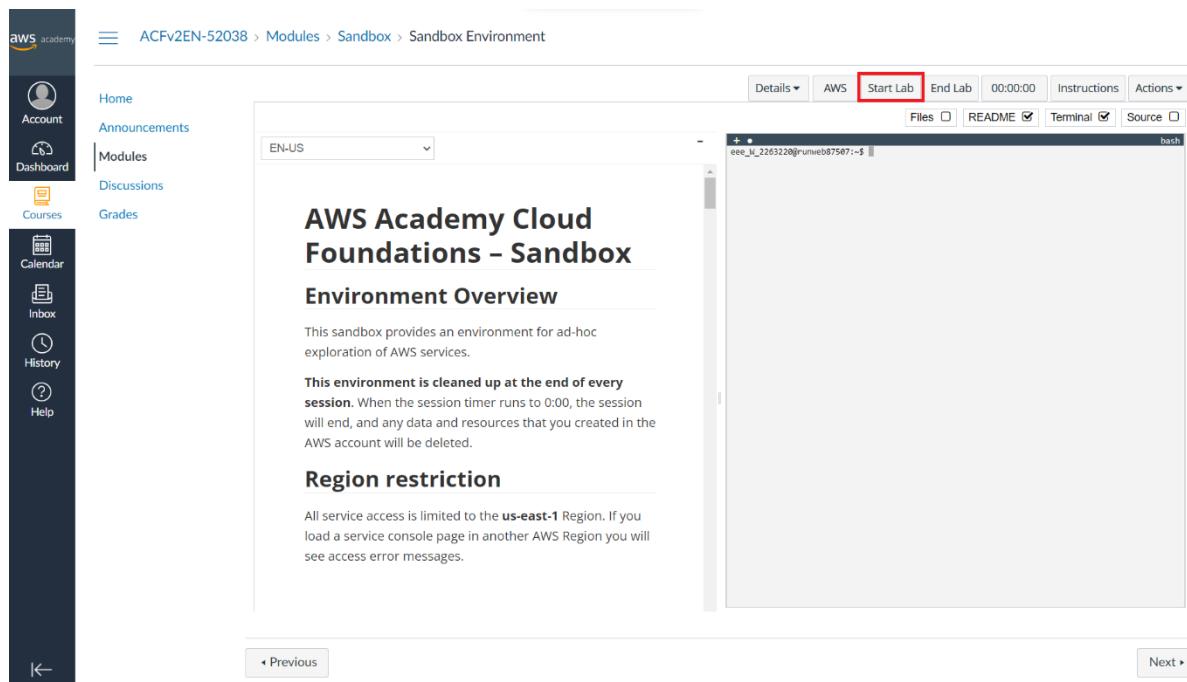


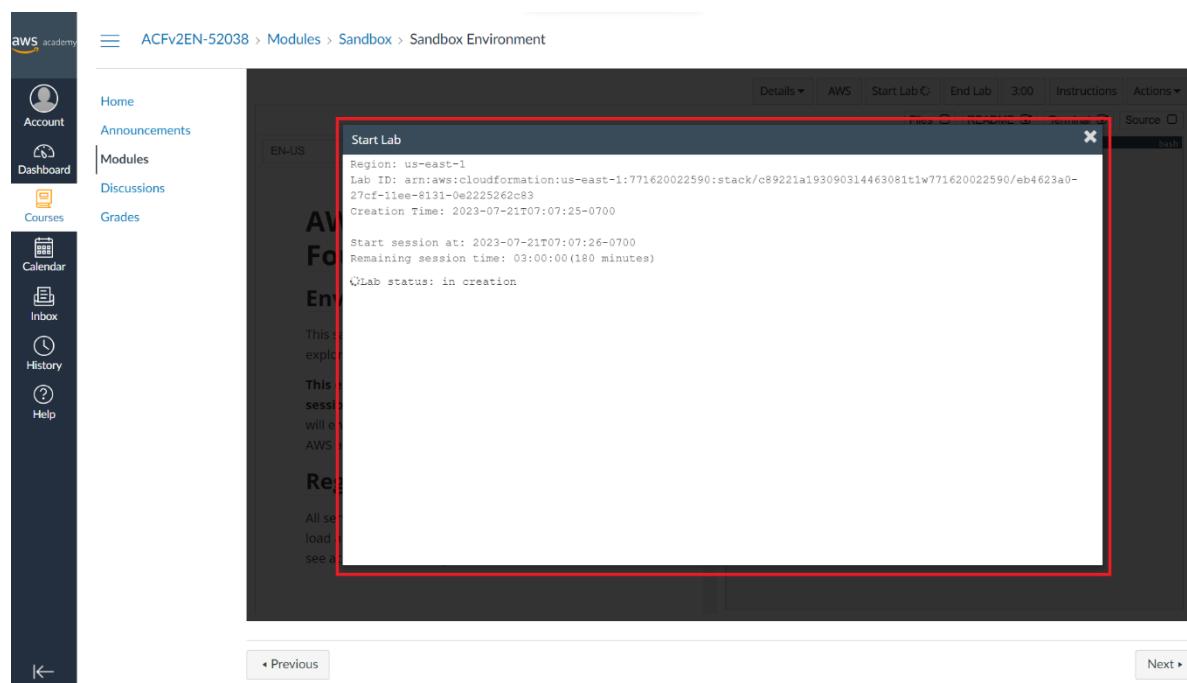
Practical 01 : Setting Up A Cloud Environment

Step 01 : Go To AWS Canvas > Modules Section > Find Sandbox And Click On The "Start Lab" Button. Then, Wait For The "Start Lab" Box To Appear.

Snapshot :



The screenshot shows the AWS Canvas interface for a course named 'ACFv2EN-52038'. The user is in the 'Modules' section, specifically viewing the 'Sandbox' environment. The top navigation bar includes 'Details', 'AWS', 'Start Lab' (which is highlighted with a red box), 'End Lab', '00:00:00', 'Instructions', and 'Actions'. Below the navigation bar, there's a dropdown for 'EN-US'. The main content area displays the 'AWS Academy Cloud Foundations – Sandbox' environment overview, which states: 'This sandbox provides an environment for ad-hoc exploration of AWS services.' It also mentions that the environment is cleaned up at the end of every session. A 'Region restriction' section notes that service access is limited to the 'us-east-1' Region. At the bottom of the page, there are 'Previous' and 'Next' navigation buttons.



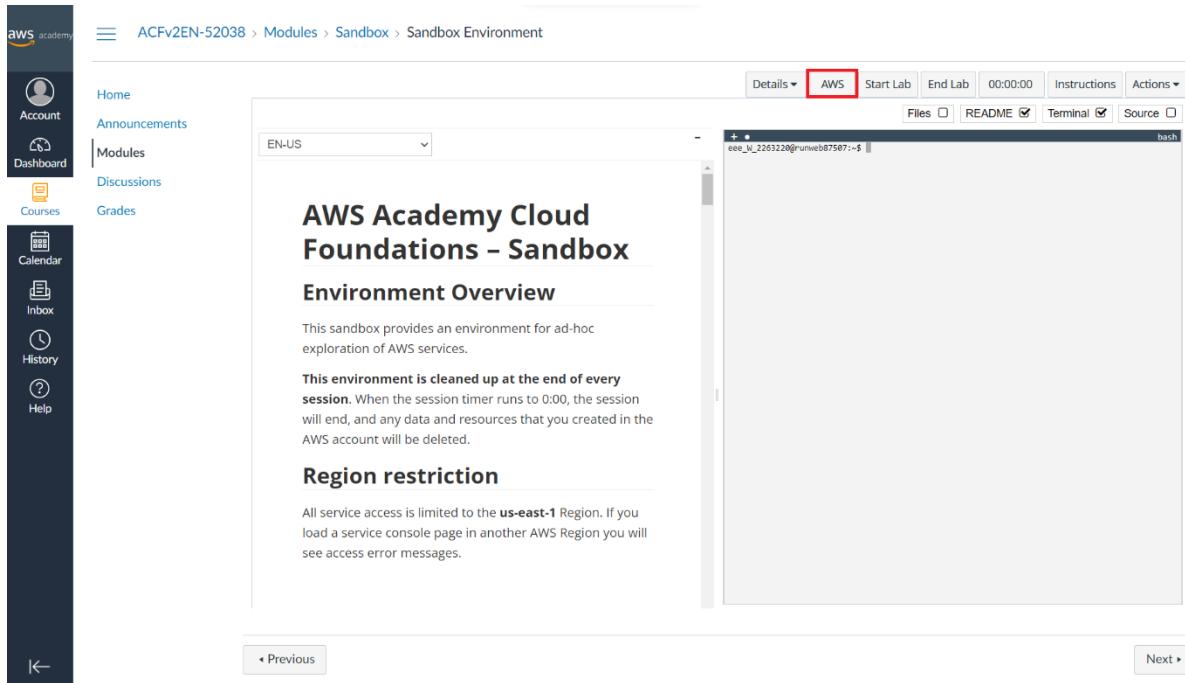
This screenshot shows the same AWS Canvas interface as the previous one, but with a 'Start Lab' dialog box overlaid on the page. The dialog box contains the following information:

- Region: us-east-1
- Lab ID: arn:aws:cloudformation:us-east-1:771620022590:stack/c89221a193090314463081t1w771620022590/eb4623a0-27cf-11ee-8131-0e2225262c83
- Creation Time: 2023-07-21T07:07:25-0700
- Start session at: 2023-07-21T07:07:26-0700
- Remaining session time: 03:00:00(180 minutes)
- Lab status: in creation

The rest of the page is dimmed, and the 'Start Lab' button in the top navigation bar is also dimmed. Navigation buttons for 'Previous' and 'Next' are visible at the bottom.

Step 02 : After The Lab Status Is Ready, Close The 'Start Lab' Box And Click On The AWS Button

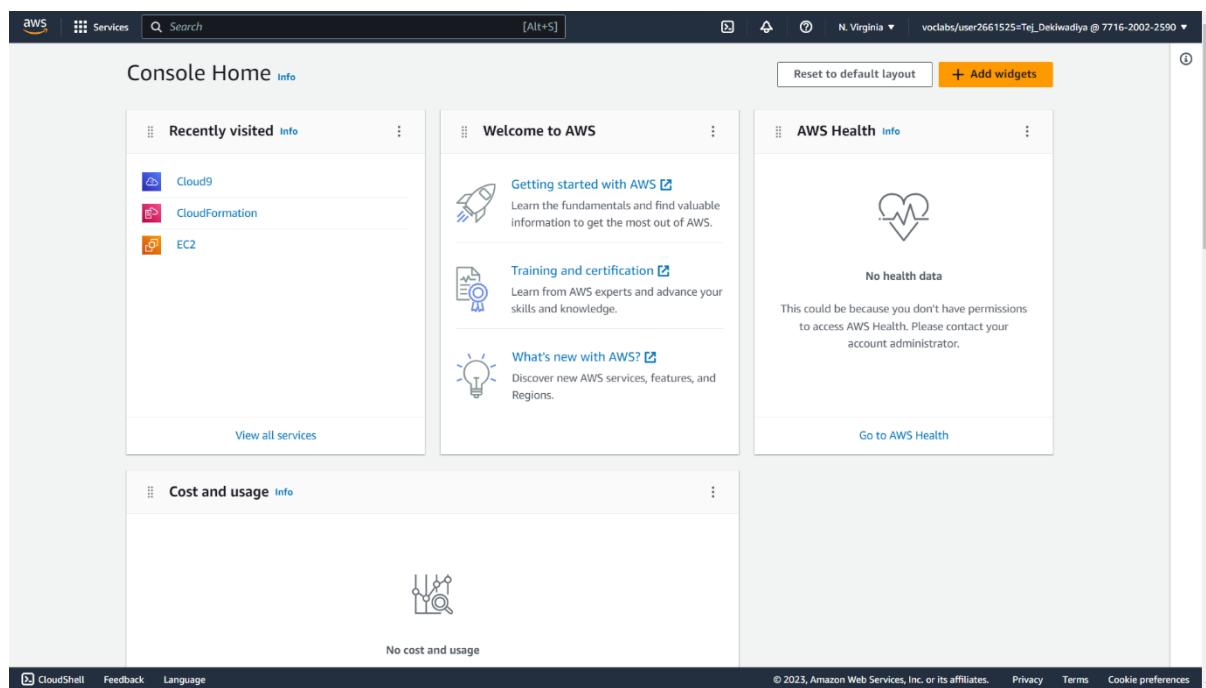
Snapshot :



The screenshot shows the AWS Academy Cloud Foundations – Sandbox environment. The 'AWS' tab is highlighted in red. The interface includes a sidebar with navigation links like Home, Announcements, Modules, Discussions, Grades, Calendar, Inbox, History, and Help. The main content area displays course details, environment overview, region restriction, and a terminal window.

Step 03 : AWS Website In New Tab Is Look Like Snapshot

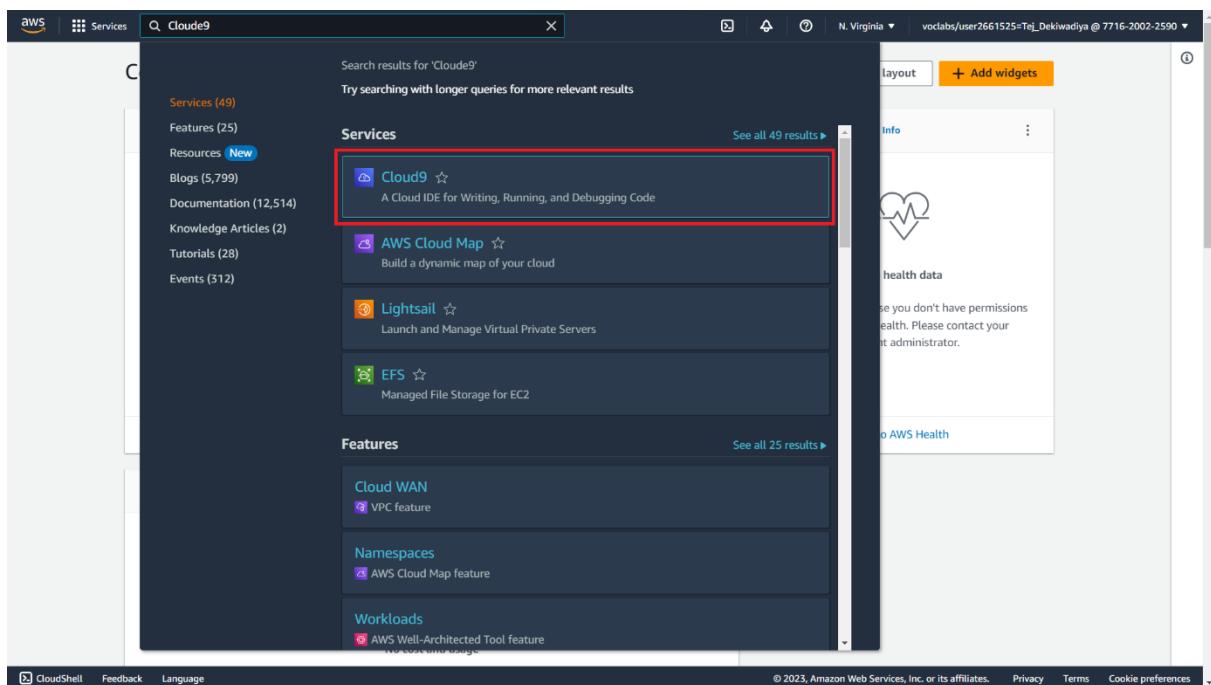
Snapshot :



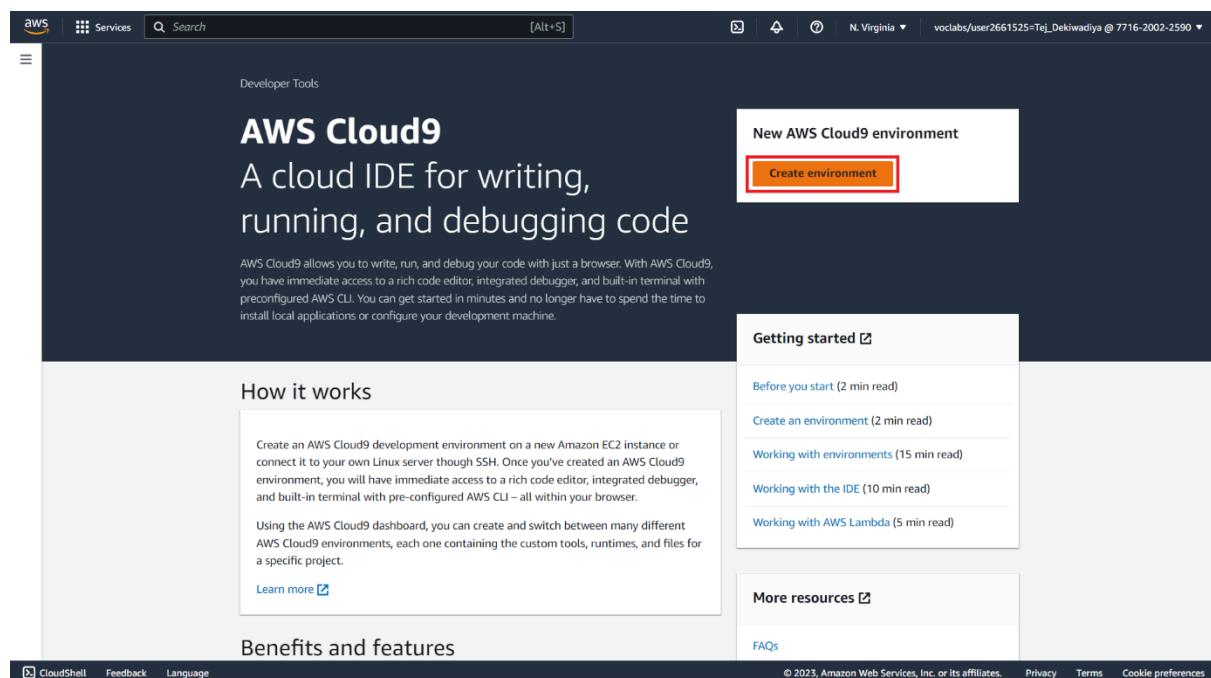
The screenshot shows the AWS Console Home page. It features sections for Recently visited services (Cloud9, CloudFormation, EC2), Welcome to AWS (Getting started with AWS, Training and certification, What's new with AWS?), AWS Health (No health data), and Cost and usage (No cost and usage). The bottom of the page includes links for CloudShell, Feedback, Language, and various legal and preference links.

Step 03 : Main Part Of This Practical Started From Here. Now, Type "Cloud9" In The Search Bar And Open The First Site. Then, Click On The "Create Environment" Button To Create An Environment In AWS.

Snapshot :



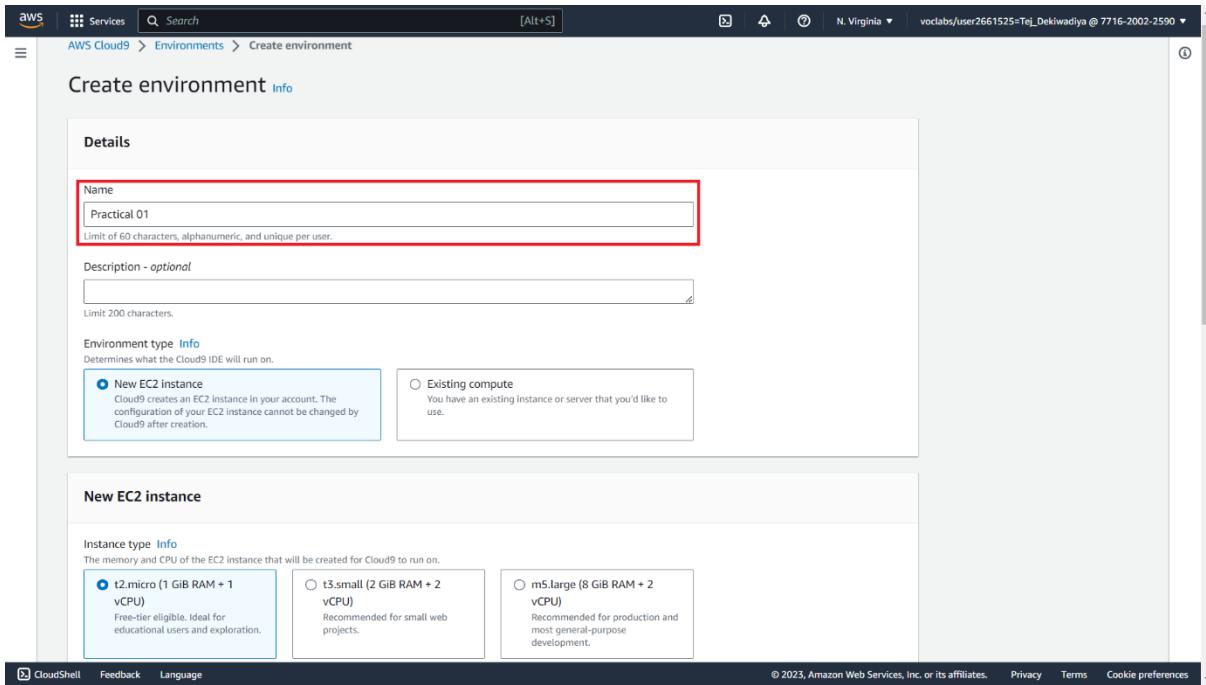
The screenshot shows the AWS search interface with the query "Cloud9" entered in the search bar. The search results are displayed under the "Services" category. The first result, "Cloud9", is highlighted with a red box. Other results listed include "AWS Cloud Map", "Lightsail", and "EFS". To the right of the search results, there is a sidebar with options like "layout", "+ Add widgets", "Info", "health data", and "AWS Health". At the bottom of the page, there are links for "CloudShell", "Feedback", "Language", and copyright information.



The screenshot shows the AWS Cloud9 landing page. The main heading is "AWS Cloud9" with the subtext "A cloud IDE for writing, running, and debugging code". A callout box highlights the "Create environment" button. Below the main heading, there is a section titled "How it works" and another titled "Benefits and features". On the right side, there are sections for "Getting started" and "More resources", each with a list of links. At the bottom of the page, there are links for "CloudShell", "Feedback", "Language", and copyright information.

Step 04 : Next Step Is To Configure The Environment. First, Set The Name Of The Environment

Snapshot :



Create environment [Info](#)

Details

Name Limit of 60 characters, alphanumeric, and unique per user.

Description - *optional*

Limit 200 characters.

Environment type [Info](#)
Determines what the Cloud9 IDE will run on.

New EC2 instance
Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.

Existing compute
You have an existing instance or server that you'd like to use.

New EC2 instance

Instance type [Info](#)
The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

t2.micro (1 GiB RAM + 1 vCPU)
Free-tier eligible. Ideal for educational users and exploration.

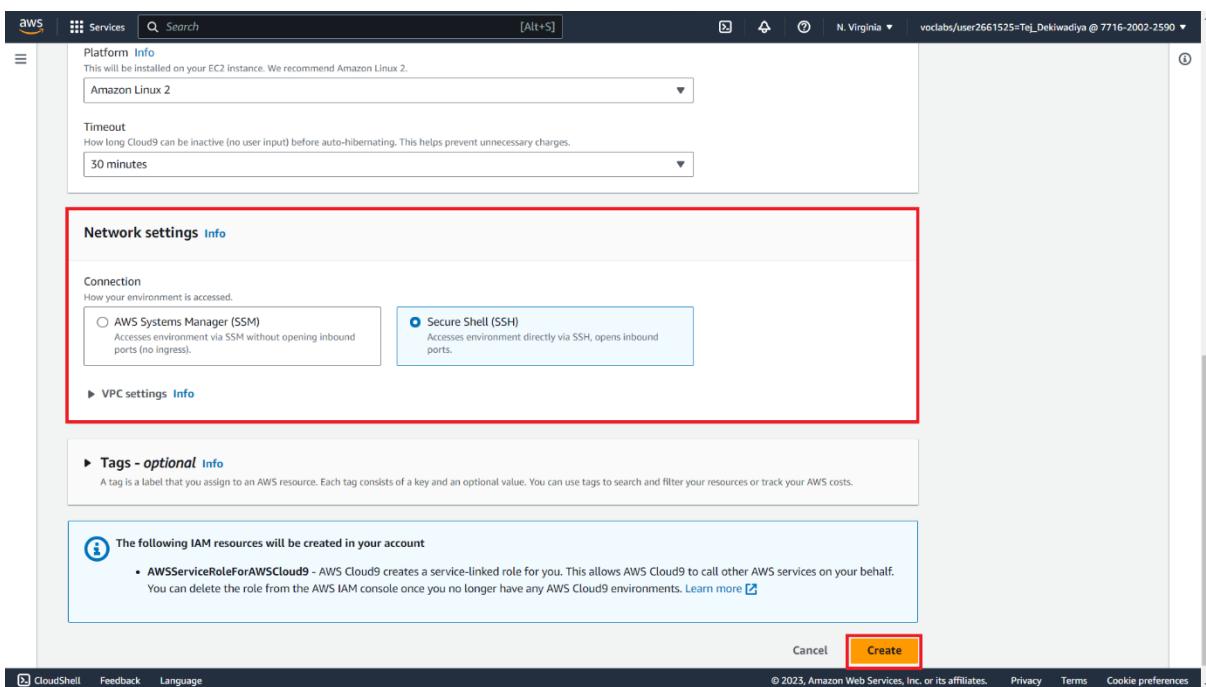
t3.small (2 GiB RAM + 2 vCPU)
Recommended for small web projects.

m5.large (8 GiB RAM + 2 vCPU)
Recommended for production and most general-purpose development.

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Step 05 : Choose The "Connection To Secure Shell (Ssh)" Option. Don't Change Any Other Settings, Then Click On The "Create" Button To Generate The Environment.

Snapshot :



Platform [Info](#)
This will be installed on your EC2 instance. We recommend Amazon Linux 2.

Timeout
How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.

Network settings [Info](#)

Connection
How your environment is accessed.

AWS Systems Manager (SSM)
Accesses environment via SSM without opening inbound ports (no ingress).

Secure Shell (SSH)
Accesses environment directly via SSH, opens inbound ports.

► VPC settings [Info](#)

Tags - *optional* [Info](#)
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

The following IAM resources will be created in your account

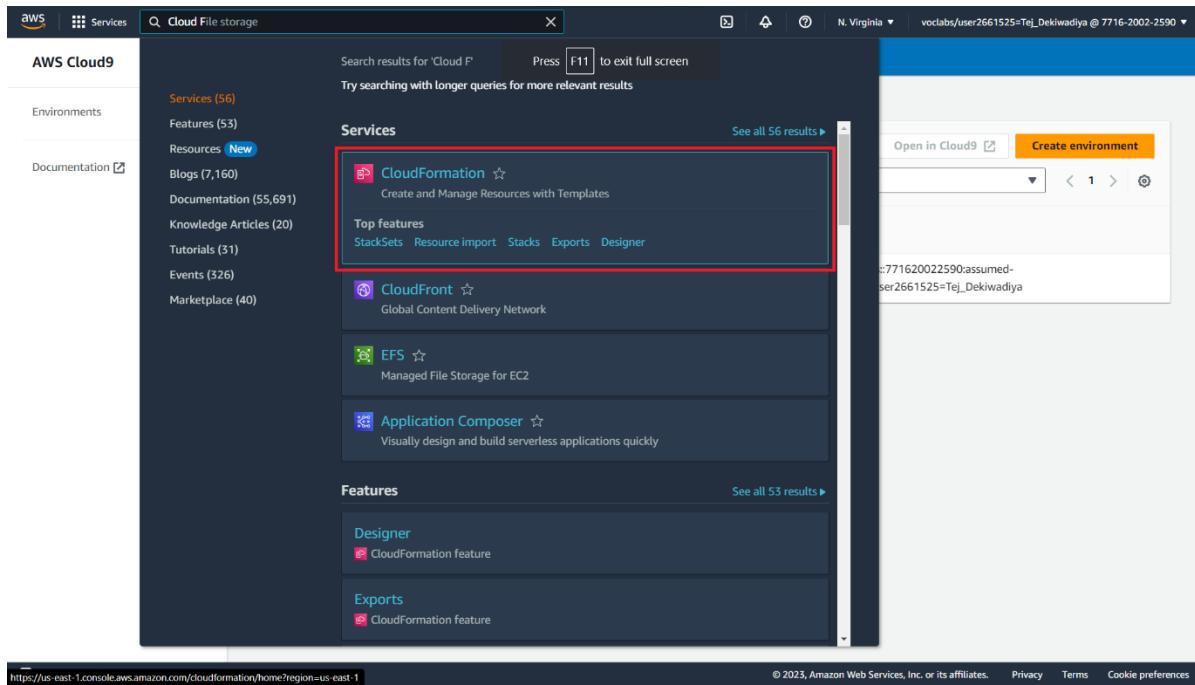
- **AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

[Cancel](#) [Create](#)

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Step 06 : Now, Type "Cloudformation" In The Search Bar, Open The First Site In The Search Results In A New Tab

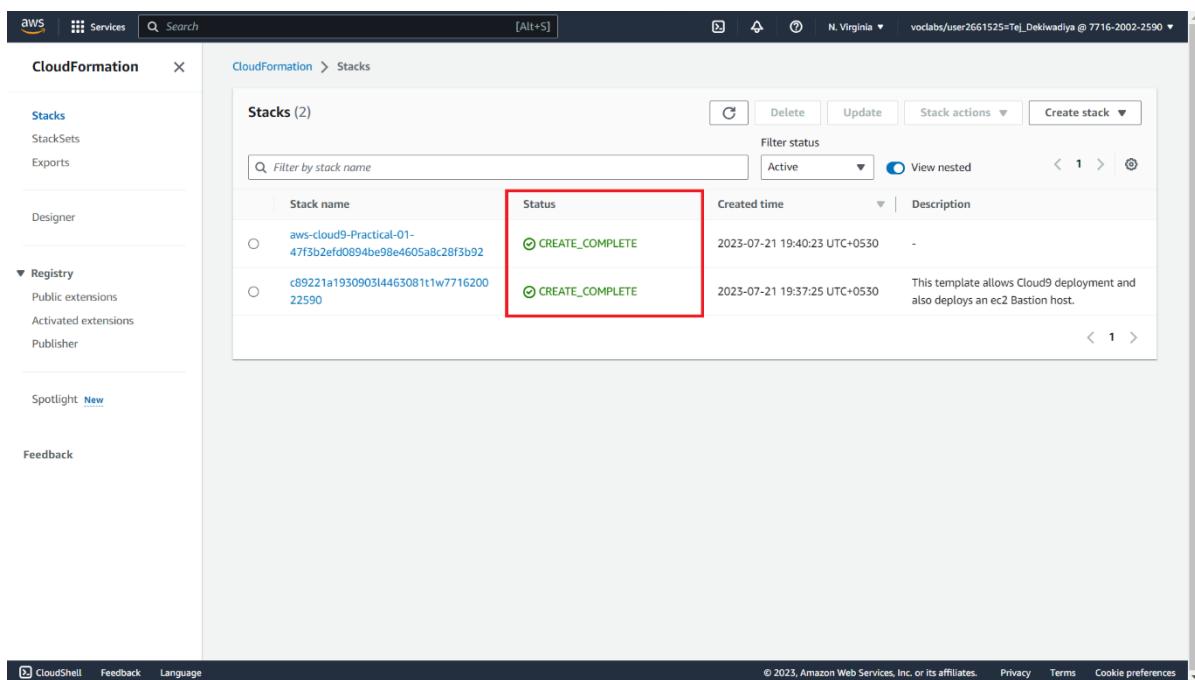
Snapshot :



The screenshot shows the AWS Cloud9 search interface. The search bar at the top contains the query "Cloud File storage". Below the search bar, there is a message: "Search results for 'Cloud F'" followed by "Press F11 to exit full screen" and "Try searching with longer queries for more relevant results". On the left, there is a sidebar with navigation links: AWS Cloud9, Environments, Documentation (56), Features (53), Resources (New), Blogs (7,160), Documentation (55,691), Knowledge Articles (20), Tutorials (31), Events (326), and Marketplace (40). The main content area displays search results under "Services" and "Features". The "Services" section includes cards for CloudFormation, CloudFront, EFS, and Application Composer. The "Features" section includes cards for Designer and Exports. The "CloudFormation" card is highlighted with a red box. It has a star icon, the name "CloudFormation", and the description "Create and Manage Resources with Templates". Below the card are links for "Top features", "StackSets", "Resource import", "Stacks", "Exports", and "Designer". The URL in the address bar is <https://us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1>.

Step 07 : Then Check The Status Of The Stacks To See If They Are Successfully Created Or Not.

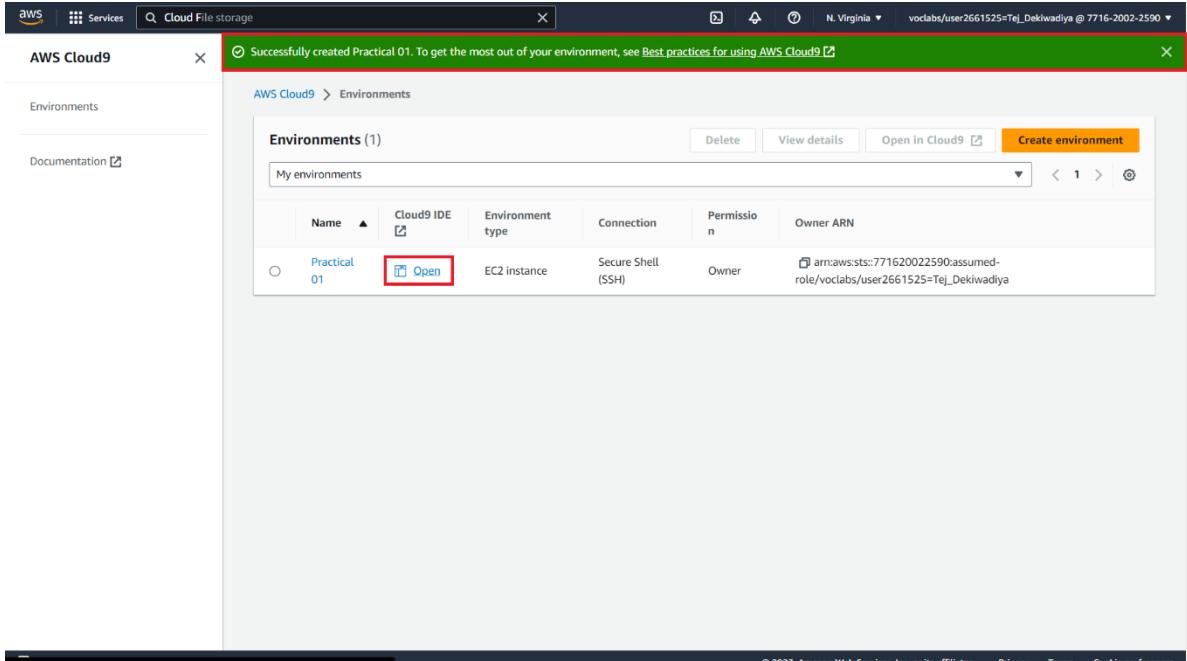
Snapshot :



The screenshot shows the CloudFormation service interface. The left sidebar has links for CloudFormation, Stacks, StackSets, Exports, Designer, Registry (Public extensions, Activated extensions, Publisher), Spotlight (New), and Feedback. The main content area is titled "CloudFormation > Stacks" and shows a table of stacks. The table has columns for "Stack name", "Status", "Created time", and "Description". There are two entries: "aws-cloud9-Practical-01-47f3b2efd0894be98e4605a8c28f3b92" with status "CREATE_COMPLETE" and created time "2023-07-21 19:40:23 UTC+0530", and "c89221a1930903l4463081t1w771620022590" with status "CREATE_COMPLETE" and created time "2023-07-21 19:37:25 UTC+0530". The "Status" column for both entries is highlighted with a red box. The URL in the address bar is <https://us-east-1.console.aws.amazon.com/cloudformation/stacks?region=us-east-1>.

Step 08 : After Successfully Creating The Environment, Click On "Open" To Entering In The AWS Cloud9 Environment.

Snapshot :



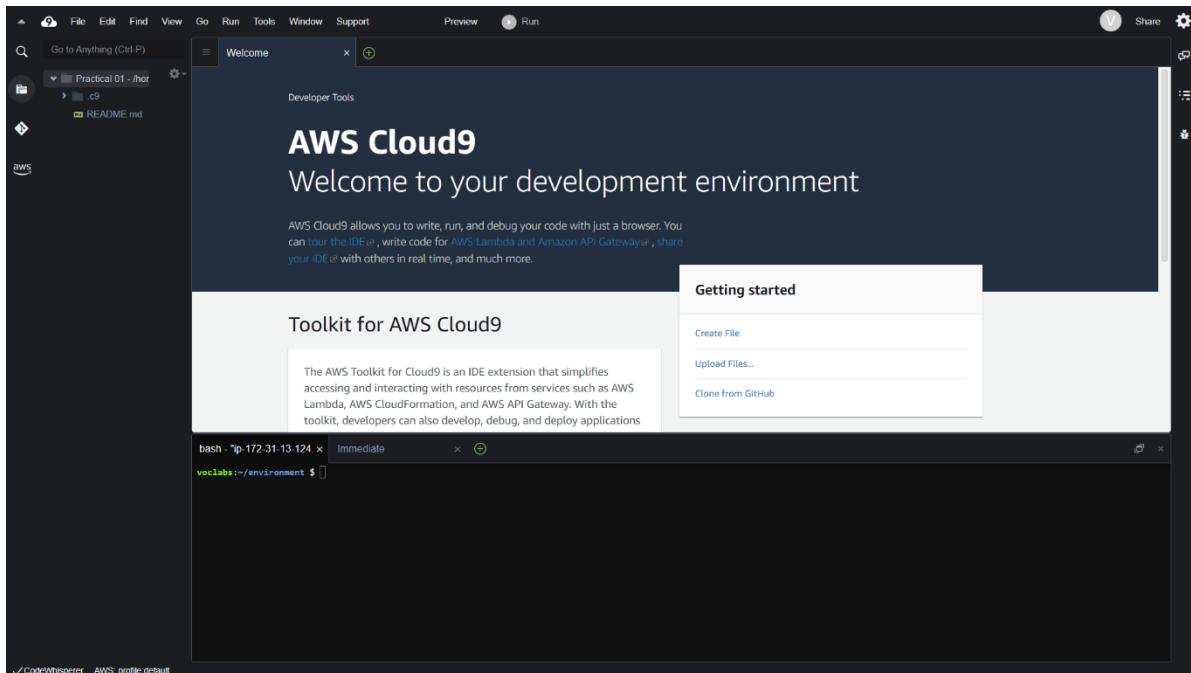
The screenshot shows the AWS Cloud9 interface. At the top, there's a success message: "Successfully created Practical 01. To get the most out of your environment, see Best practices for using AWS Cloud9". Below this, the "Environments" section displays a table with one row:

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
Practical 01	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::771620022590:assumed-role/voclabs/user2661525=Tej_Dekiwadiya

The "Open" link for the "Practical 01" environment is highlighted with a red box. Below the table, there's a large blue cloud icon containing the number "9". At the bottom of the screen, there's a status bar with the URL "https://us-east-1.console.aws.amazon.com/cloud9/ide/473b2ef0894be98e405a8c28f3b92", copyright information "© 2023, Amazon Web Services, Inc. or its affiliates.", and links for "Privacy", "Terms", and "Cookie preferences".

Step 09 : Now, You Can See The Welcome Screen Of AWS Cloud9.

Snapshot :



Step 10 : In The AWS Cloud9 Environment, Execute Various Linux Commands And Observe Their Outputs.

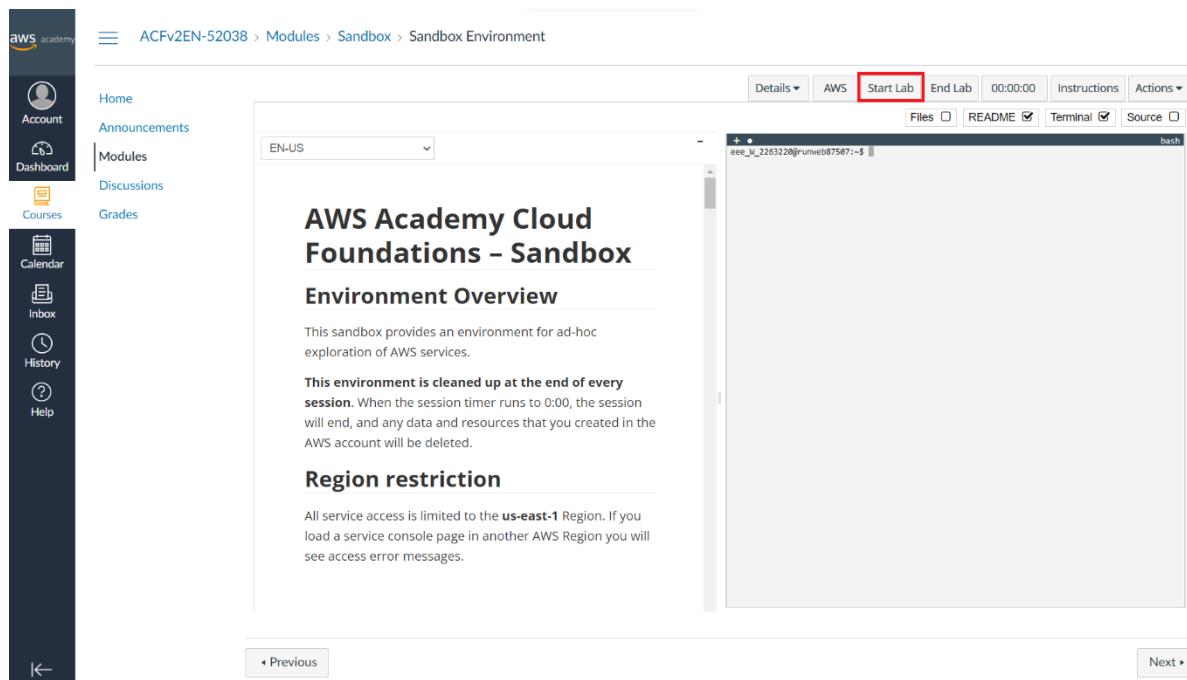
Snapshot :

```
bash - "ip-172-31-13-124.x" Immediate x + 
vclabs:~/environment $ ls
README.md
vclabs:~/environment $ mkdir Practical01
vclabs:~/environment $ ls
Practical01 README.md
vclabs:~/environment $ cd Practical01
vclabs:~/environment/Practical01 $ nano
vclabs:~/environment/Practical01 $ ls
AWSCloud
vclabs:~/environment/Practical01 $ cat AWSCloud
Welcome to your development environment
vclabs:~/environment/Practical01 $ rm AWSCloud
```

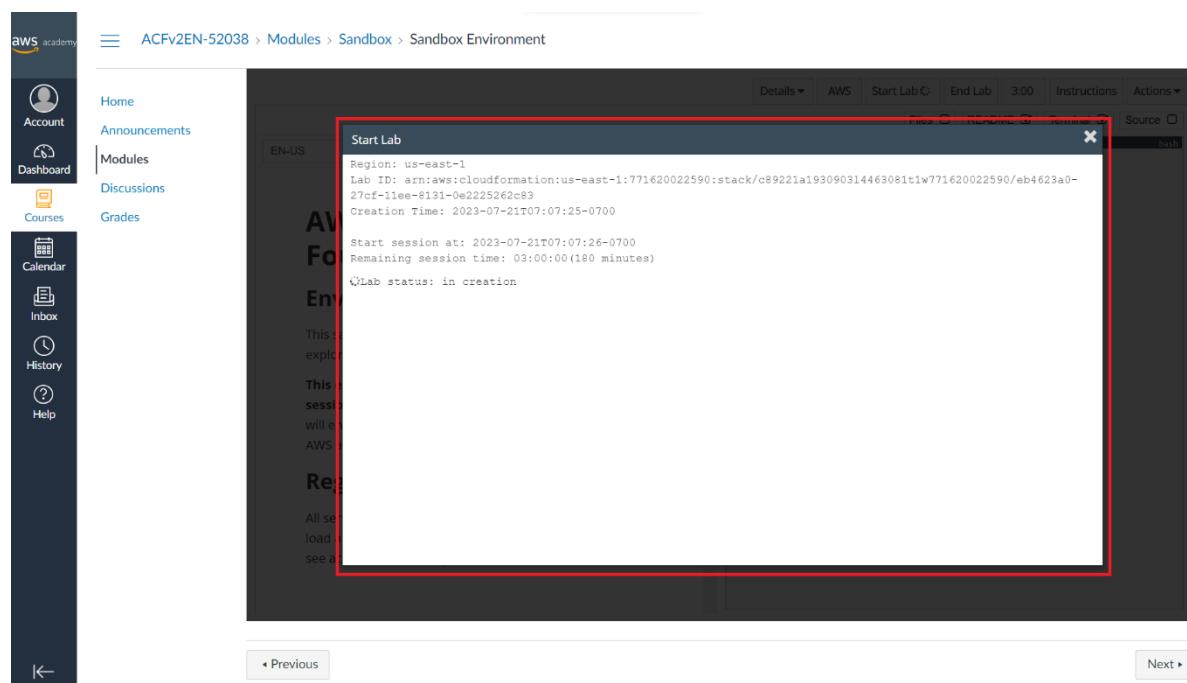
Practical 02 : Deploy Virtual Machine

Step 01 : Go To AWS Canvas > Modules Section > Find Sandbox And Click On The "Start Lab" Button. Then, Wait For The "Start Lab" Box To Appear.

Snapshot :



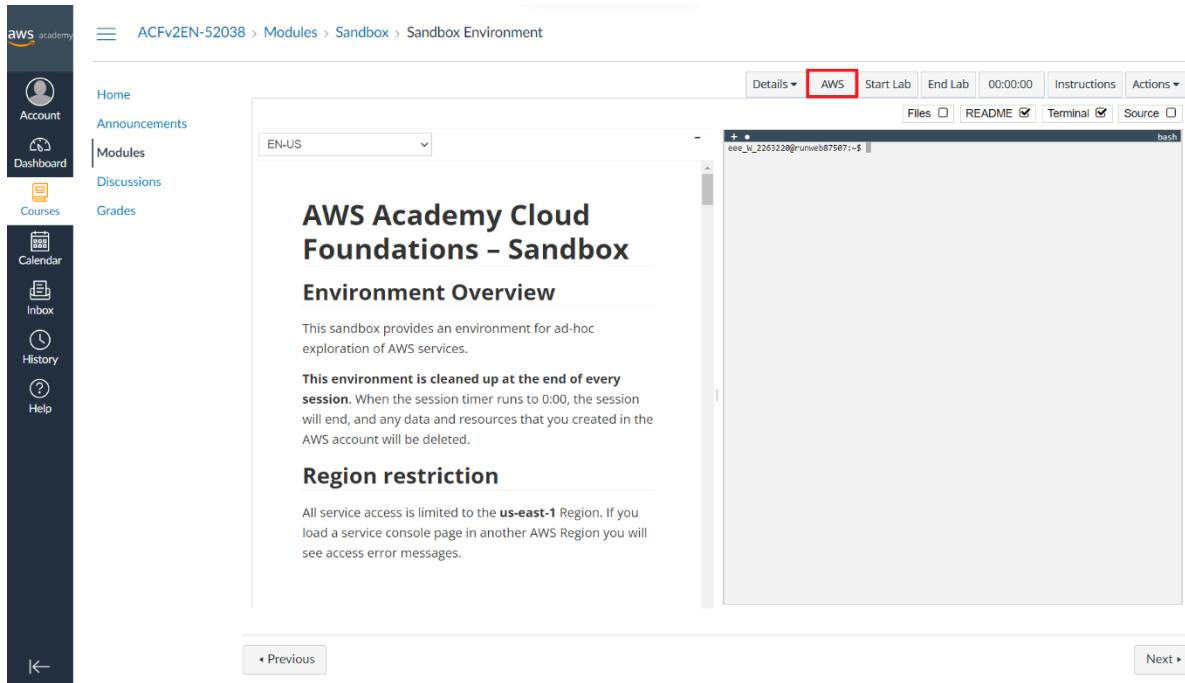
The screenshot shows the AWS Canvas interface for a course module. The left sidebar includes links for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area displays the 'Sandbox Environment' page for the 'ACFv2EN-52038' course. The top navigation bar has tabs for Details, AWS, Start Lab (which is highlighted with a red box), End Lab, 00:00:00, Instructions, and Actions. Below the navigation is a file browser with options for Files, README, Terminal, and Source. The main content area shows the 'AWS Academy Cloud Foundations – Sandbox' environment overview, which includes a note about region restrictions and session cleanup. At the bottom of the page are 'Previous' and 'Next' navigation buttons.



This screenshot shows the same AWS Canvas interface as the previous one, but with a modal dialog box overlaid. The dialog is titled 'Start Lab' and contains detailed information about the lab session: Region: us-east-1, Lab ID: arn:aws:cloudformation:us-east-1:771620022590:stack/c89221a193090314463081t1w771620022590/eb4623a0-27cf-11ee-8131-0e2225262c83, Creation Time: 2023-07-21T07:07:25-0700, Start session at: 2023-07-21T07:07:26-0700, Remaining session time: 03:00:00(180 minutes), and QLab status: in creation. The background of the canvas page is dimmed, and there are 'Previous' and 'Next' navigation buttons at the bottom.

Step 02 : After The Lab Status Is Ready, Close The 'Start Lab' Box And Click On The AWS Button

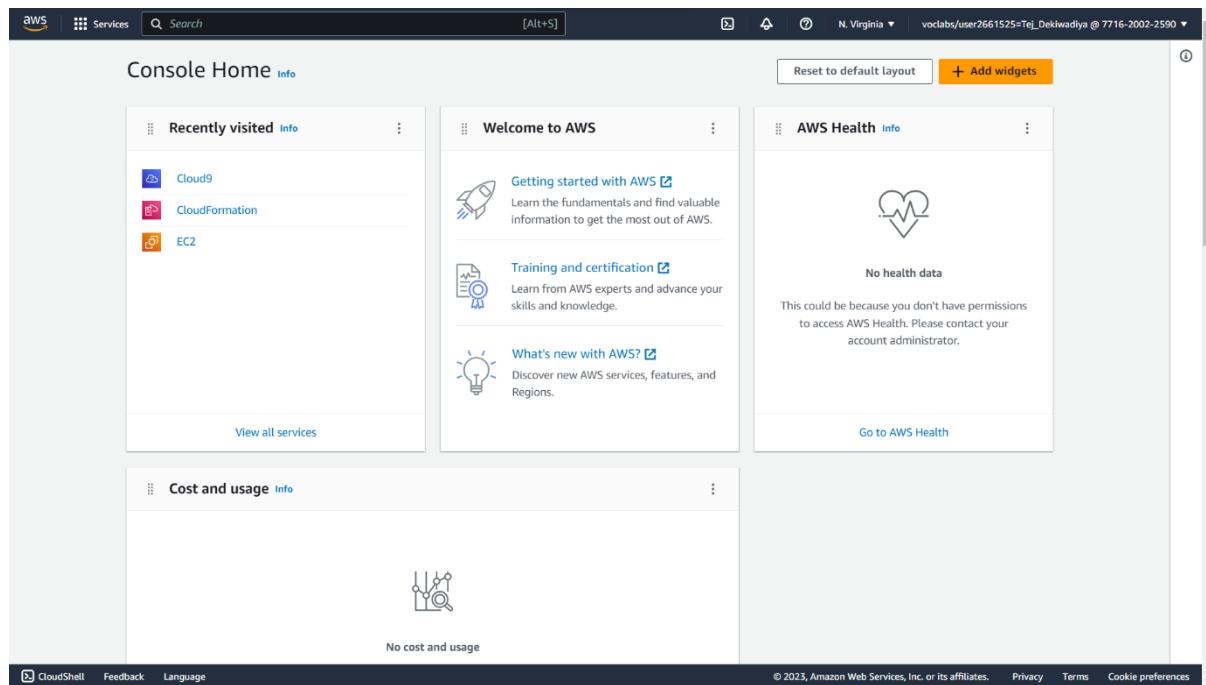
Snapshot :



The screenshot shows the AWS Academy Cloud Foundations – Sandbox environment. The 'AWS' tab is highlighted in red. The interface includes a sidebar with navigation links like Home, Announcements, Modules, Discussions, Grades, Calendar, Inbox, History, and Help. The main content area displays the 'AWS Academy Cloud Foundations – Sandbox' environment overview, which states: 'This sandbox provides an environment for ad-hoc exploration of AWS services.' It also mentions that the environment is cleaned up at the end of every session. A 'Region restriction' section notes that service access is limited to the us-east-1 Region. A terminal window at the bottom shows a command prompt.

Step 03 : AWS Website In New Tab Is Look Like Snapshot

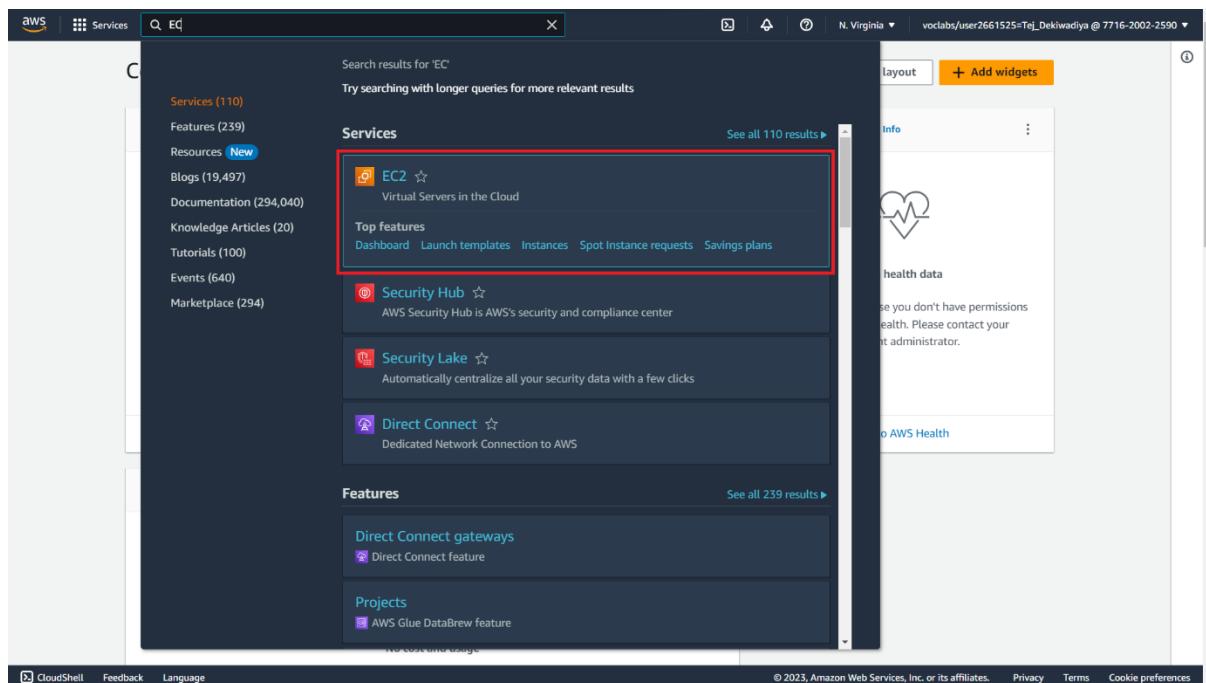
Snapshot :



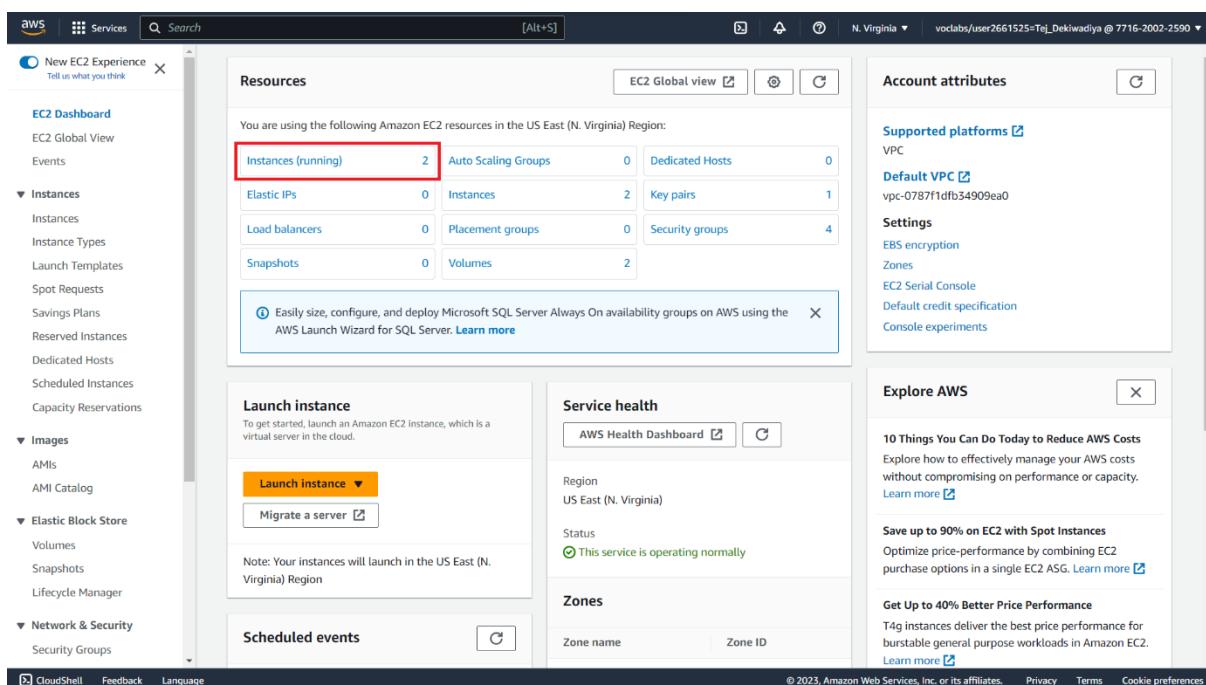
The screenshot shows the AWS Console Home page. It features a 'Recently visited' section with links to Cloud9, CloudFormation, and EC2. The 'Welcome to AWS' section includes links to 'Getting started with AWS', 'Training and certification', and 'What's new with AWS?'. The 'AWS Health' section indicates 'No health data' and suggests contacting account administrators. The 'Cost and usage' section shows 'No cost and usage'. The footer includes links for CloudShell, Feedback, Language, and various legal and preference options.

Step 03 : Main Part Of This Practical Started From Here. Now, Type "Ec2" In The Search Bar And Open The First Site. Then, Click On The "**Instances (Running)**" In Resources Section.

Snapshot :



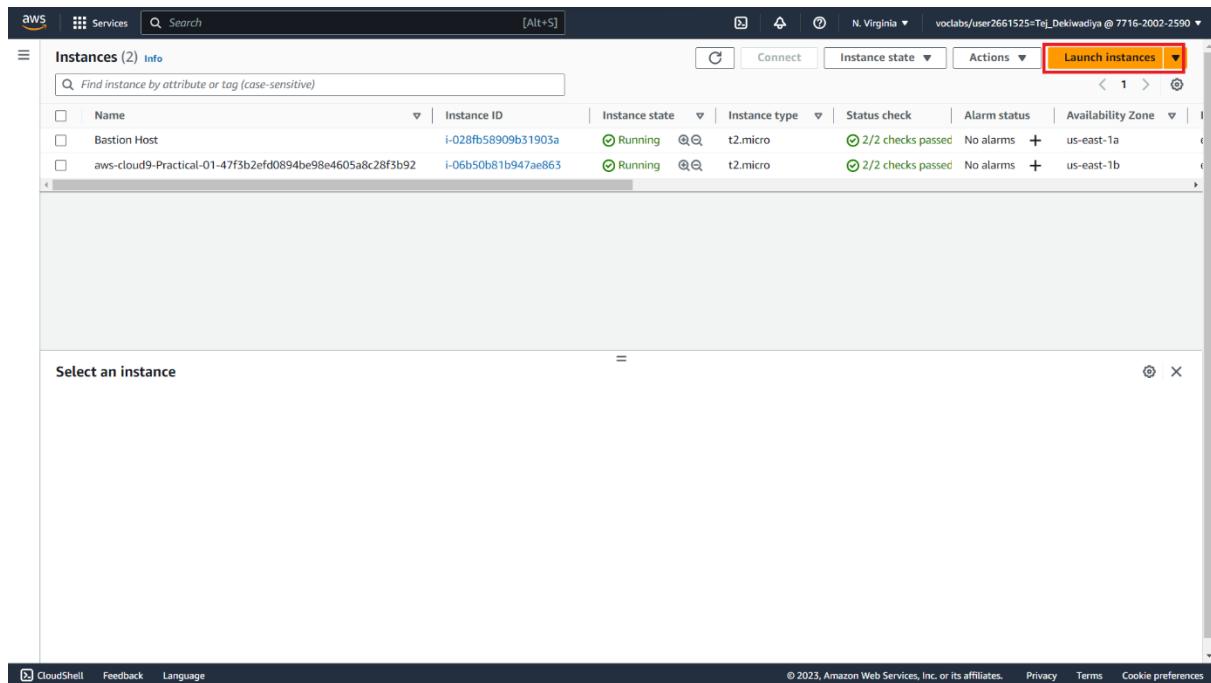
The screenshot shows the AWS search interface with the search term 'EC' entered. The results list includes the EC2 service, which is highlighted with a red box. Other results listed include Security Hub, Security Lake, and Direct Connect. The EC2 entry has a sub-section titled 'Top features' with links to Dashboard, Launch templates, Instances, Spot Instance requests, and Savings plans.



The screenshot shows the AWS EC2 Global View dashboard. The left sidebar shows navigation options like EC2 Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main area displays a summary of resources: Instances (running) 2, Auto Scaling Groups 0, Dedicated Hosts 0, Elastic IPs 0, Instances 2, Key pairs 1, Load balancers 0, Placement groups 0, Security groups 4, Snapshots 0, and Volumes 2. A callout box highlights the 'Instances (running)' section. The right sidebar contains sections for Account attributes, Supported platforms (VPC), Default VPC (vpc-0787f1fdb34909ea0), Settings (EBS encryption, Zones, EC2 Serial Console, Default credit specification, Console experiments), and Explore AWS (10 Things You Can Do Today to Reduce AWS Costs, AWS Health Dashboard, Service health, Region US East (N. Virginia), Status (This service is operating normally), Zones, Scheduled events).

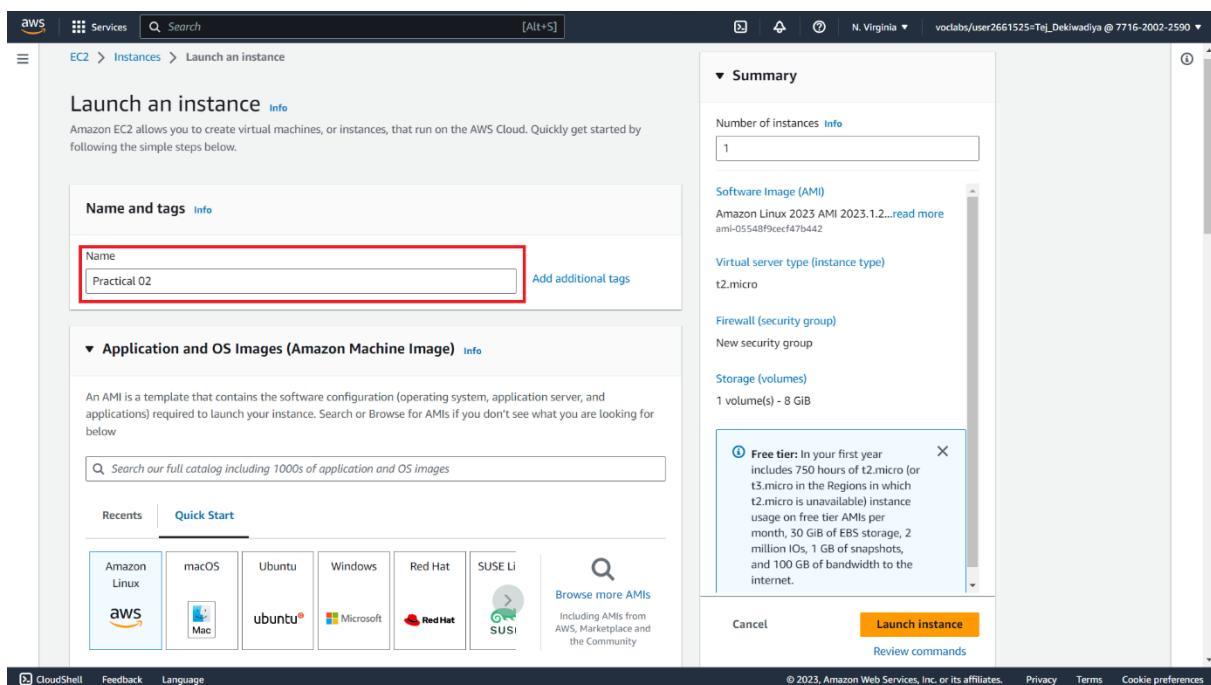
Step 04 : Create A New Instance By Clicking On "Launch Instance"

Snapshot :

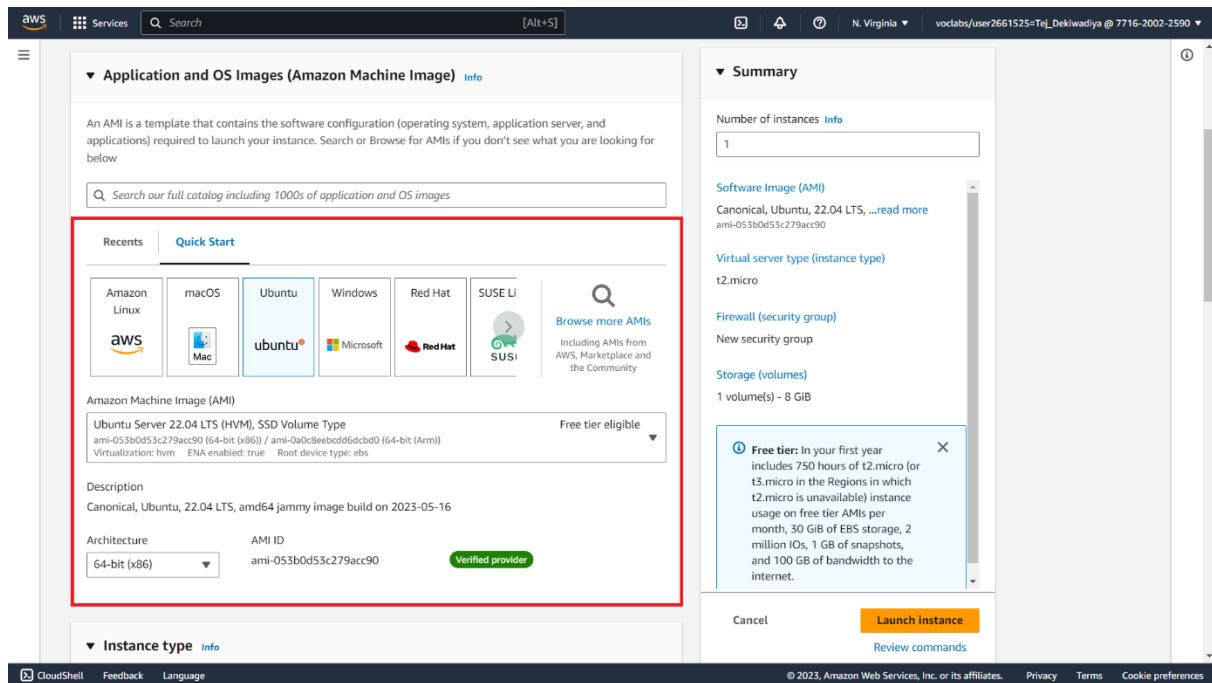


Step 05 : Next Step Is To Configure The Instance. First, Set The Name Of The Instance

Snapshot :

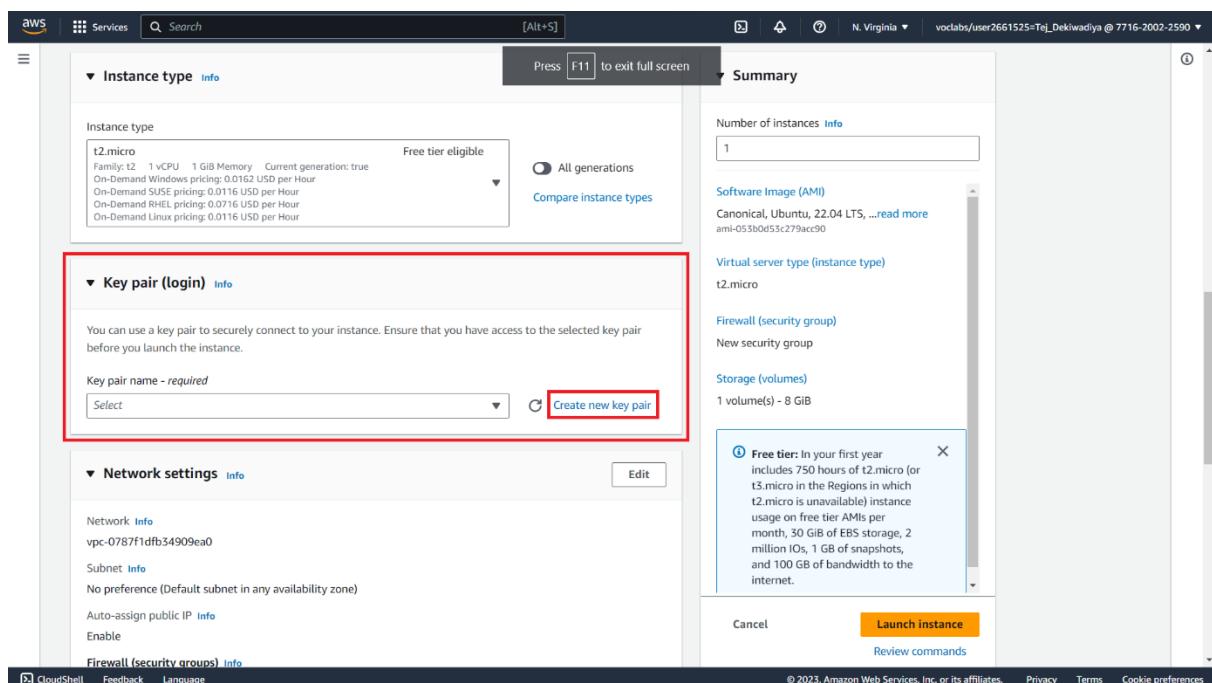


Step 06 : Next Step Is To Choose The Application And OS Images Ubuntu > Ubuntu Server 22.04 LTS Snapshot :



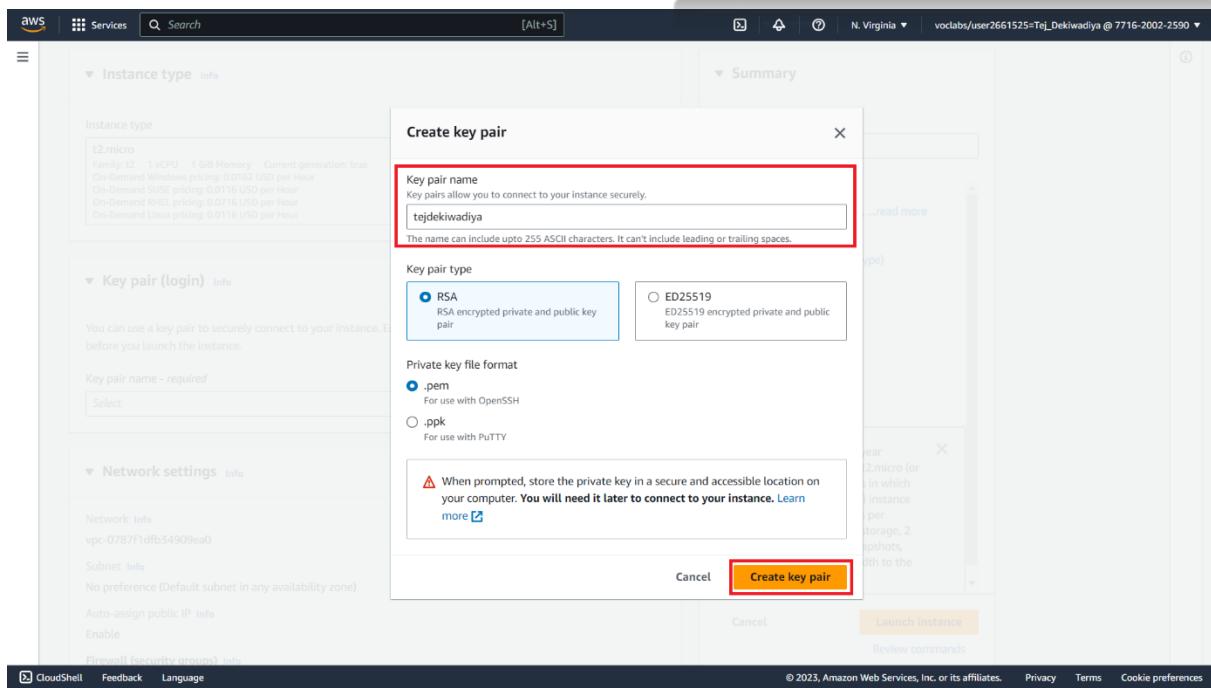
Step 07 : Next Step Is To Create New Key Pair In Key Pair (Login) Section

Screenshot :



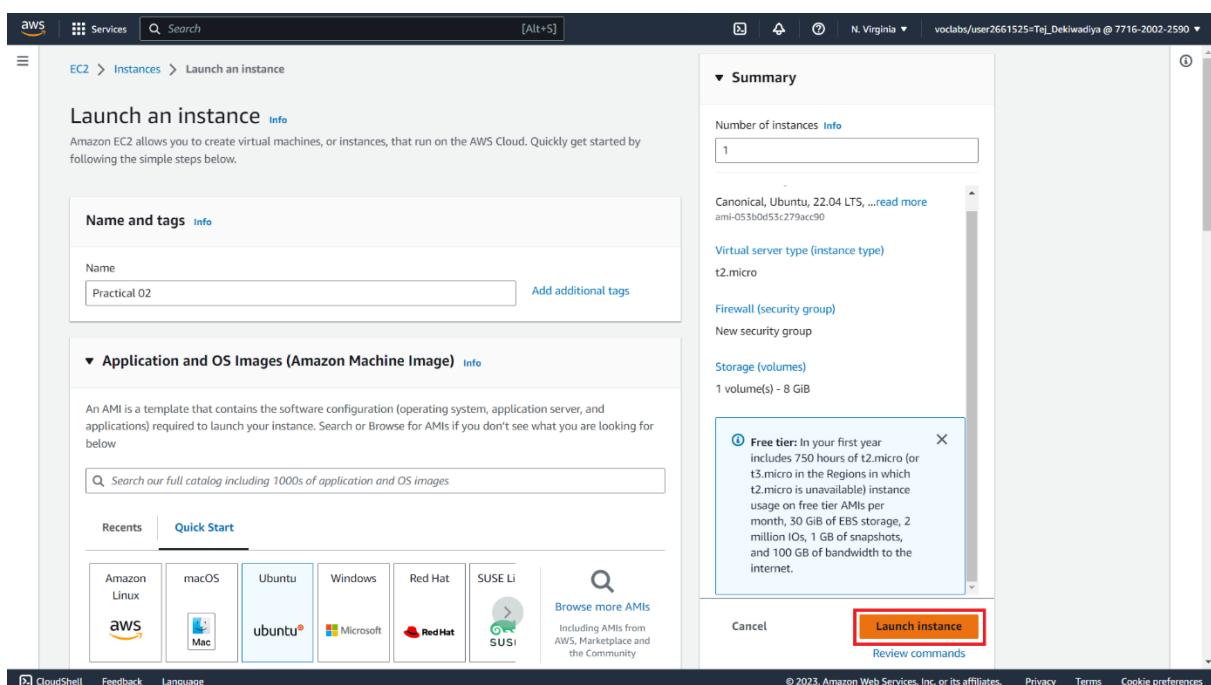
Step 08 : Provide A Key Pair Name And Then Click On "Create Key Pair" To Generate The Key Pair.
(Downloading One File While Creating)

Snapshot :



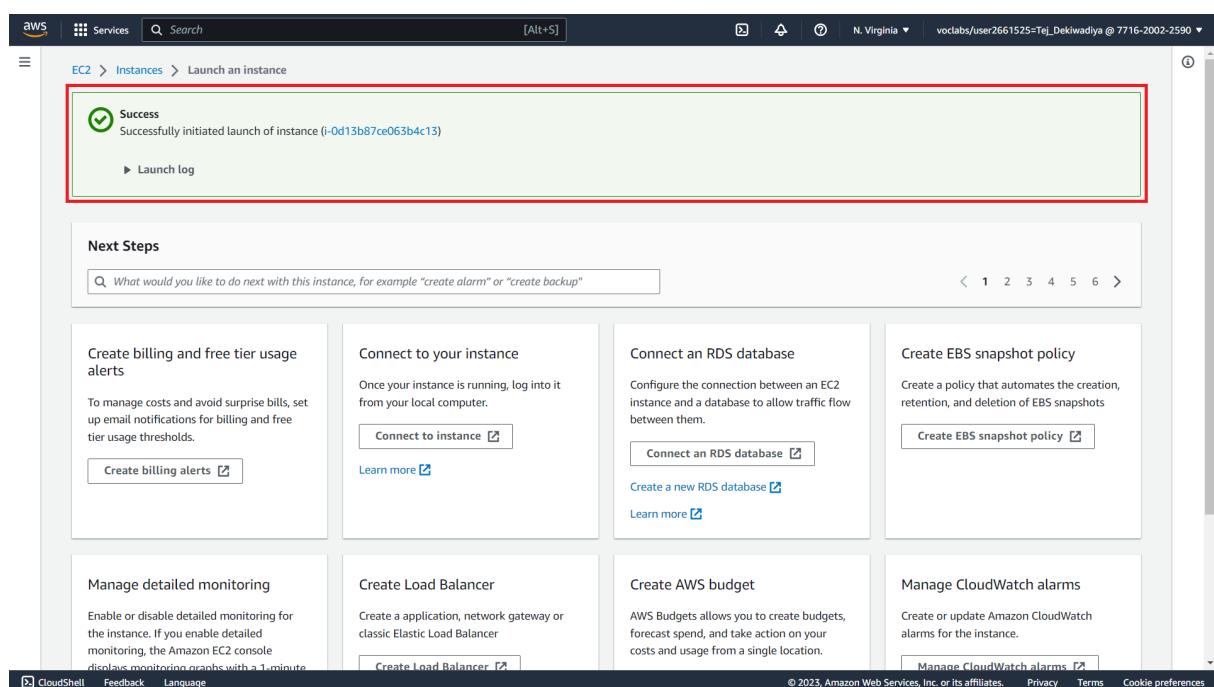
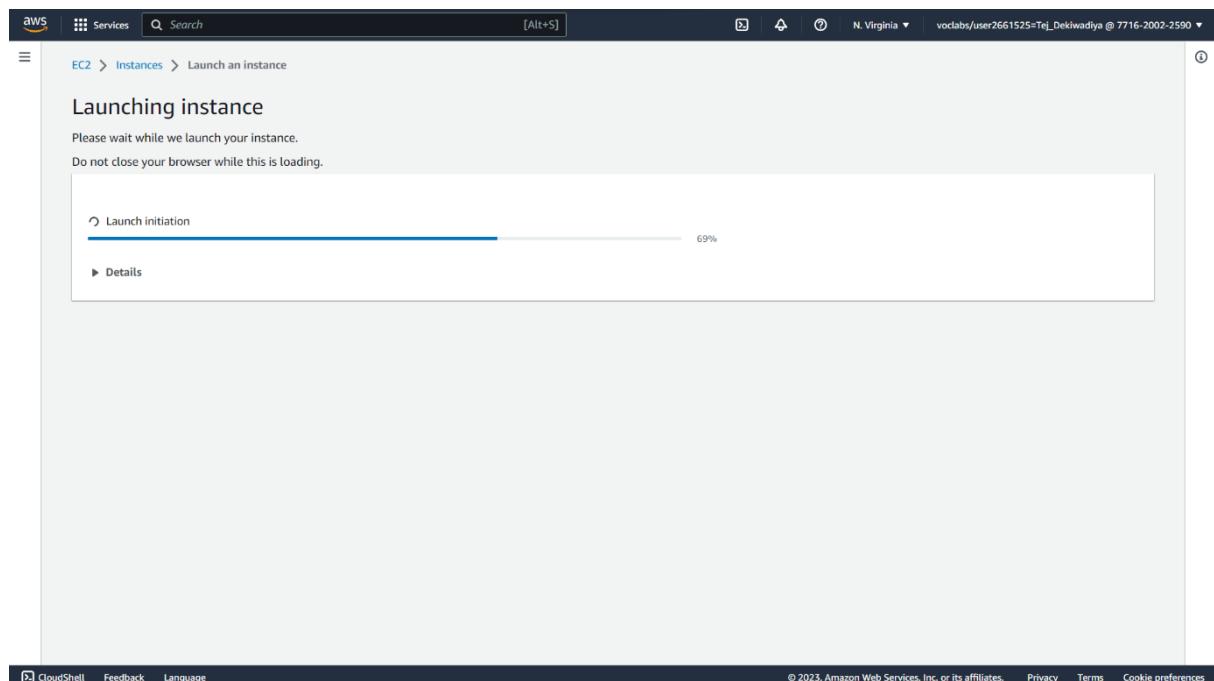
Step 09 : Do Not Modify Any Other Instance Settings. Click On "Launch Instance" To Proceed.

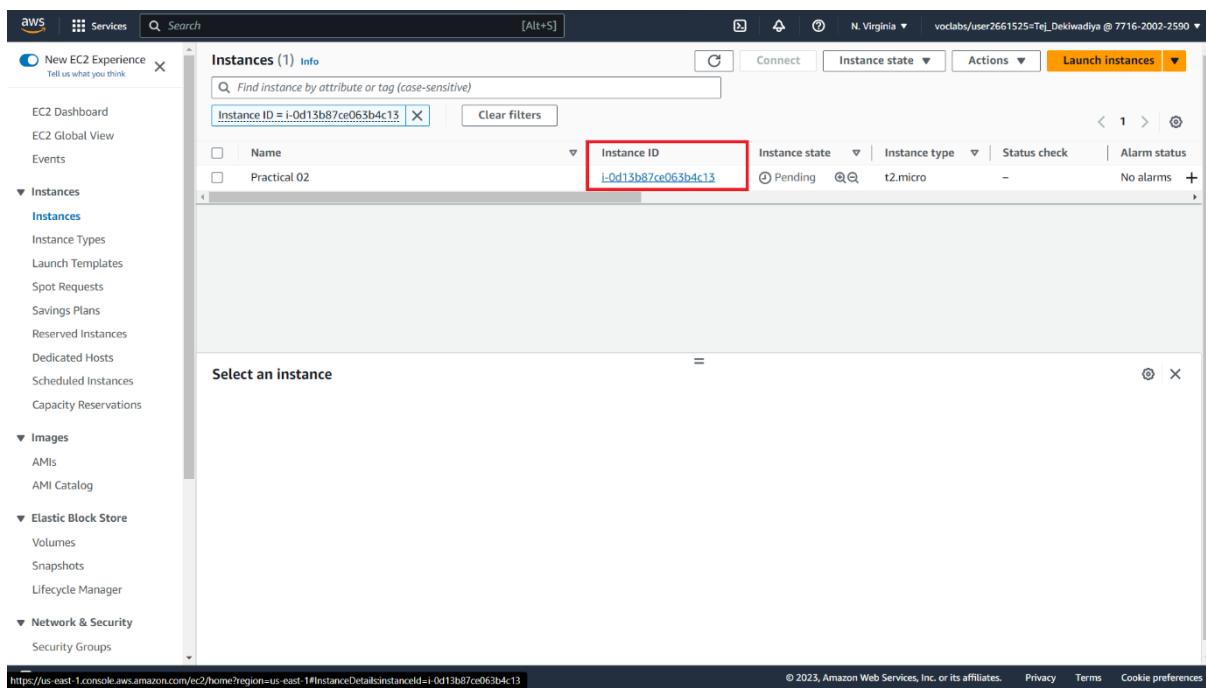
Snapshot :



Step 10 : The Process Of Launching The Instance Will Begin, And It May Take A Few Seconds To Complete Successfully. Once It's Ready, Click On The Instance Id To Launch The Created Instance

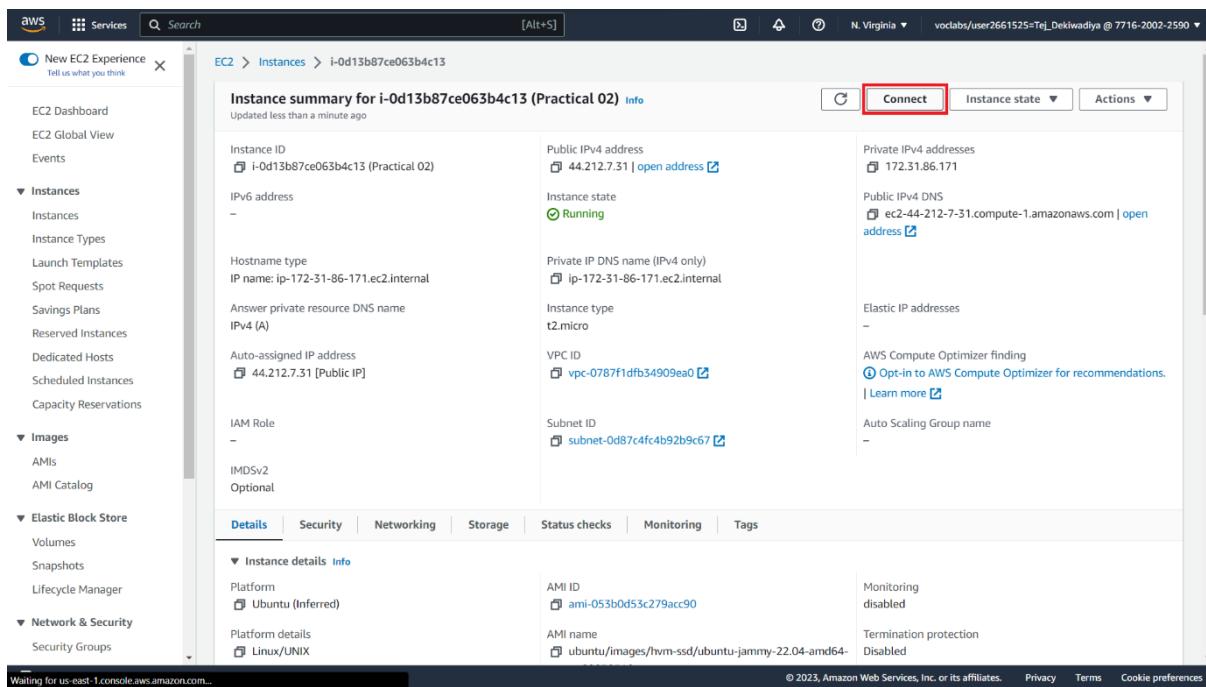
Snapshot :





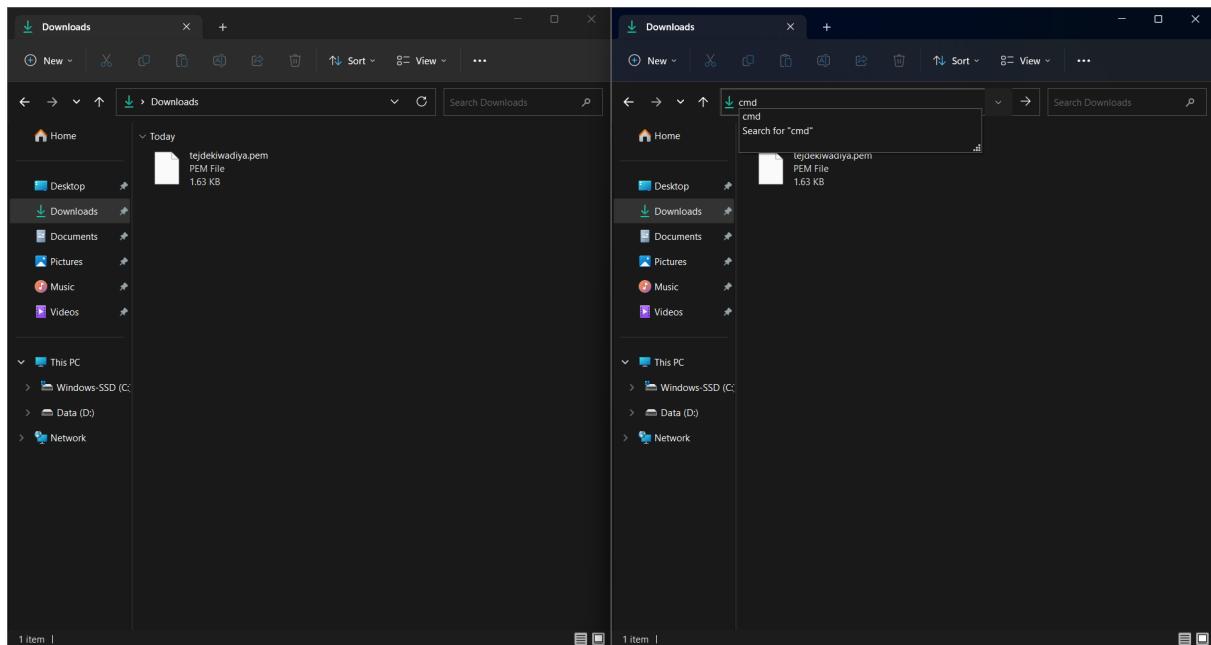
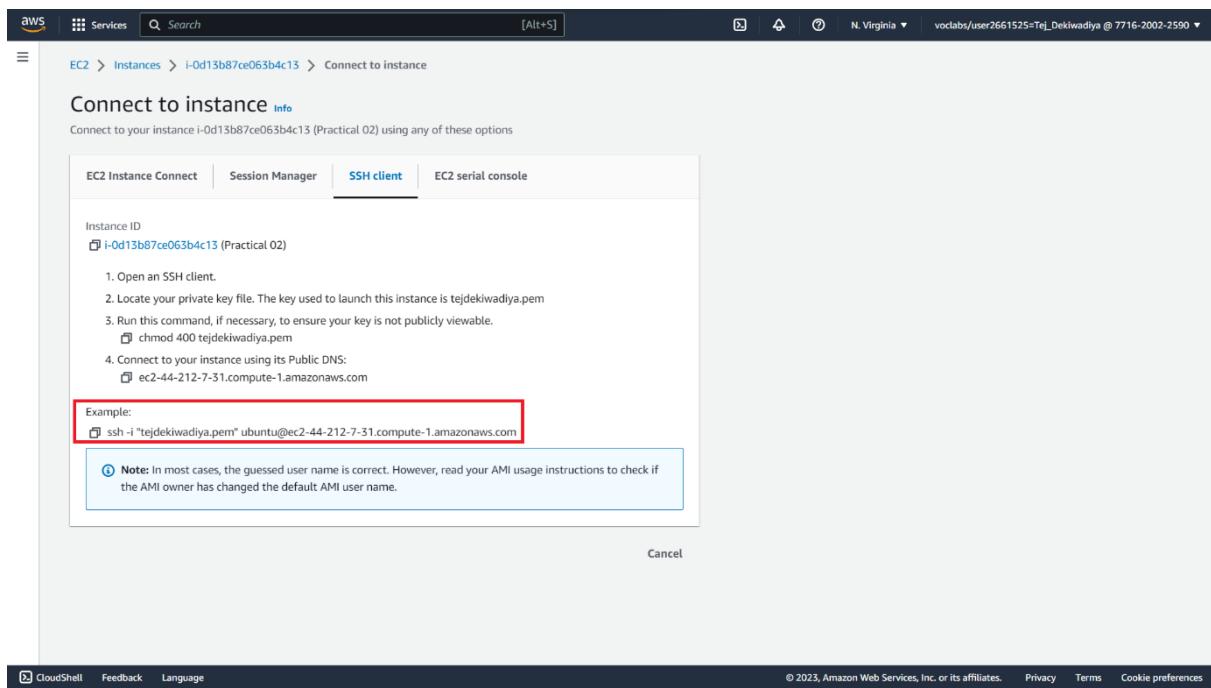
Step 11 : After Successfully Launching The Instance, Click On The "**Connect**" Button To Connect To The Instance.

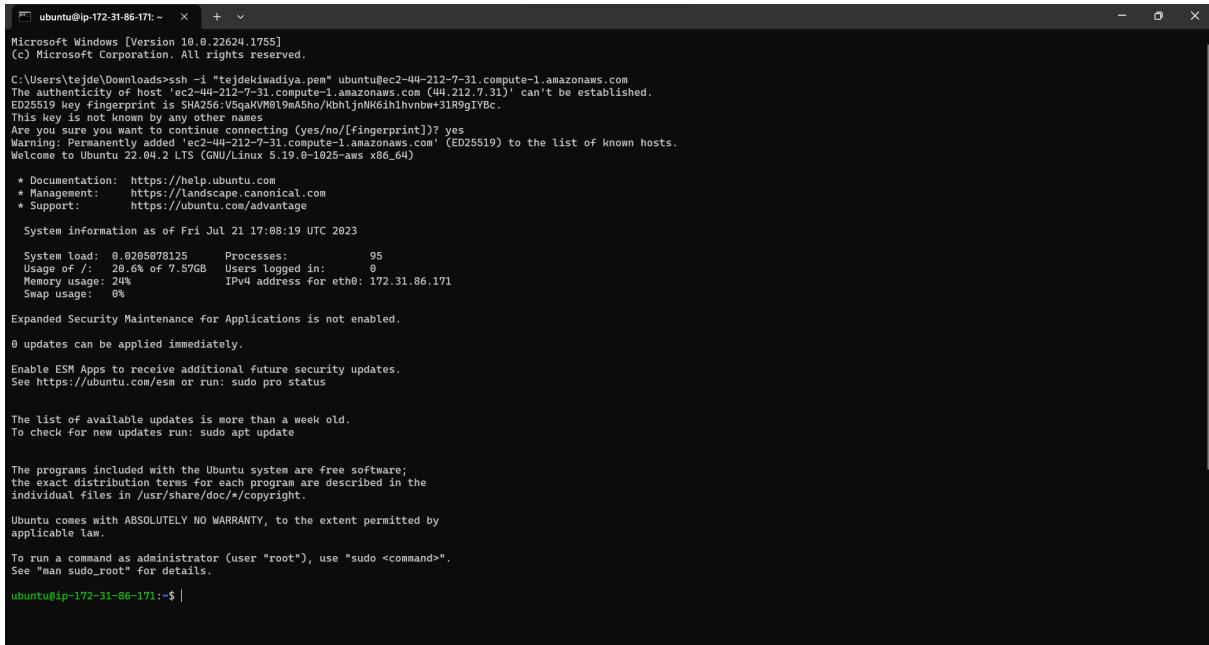
Snapshot :



Step 12 : After Downloading The Key Pair File During The Creation, Navigate To The Folder In The **CMD** (Or Terminal) And Execute The Example Code From The **SSH Client Panel**

Snapshot :





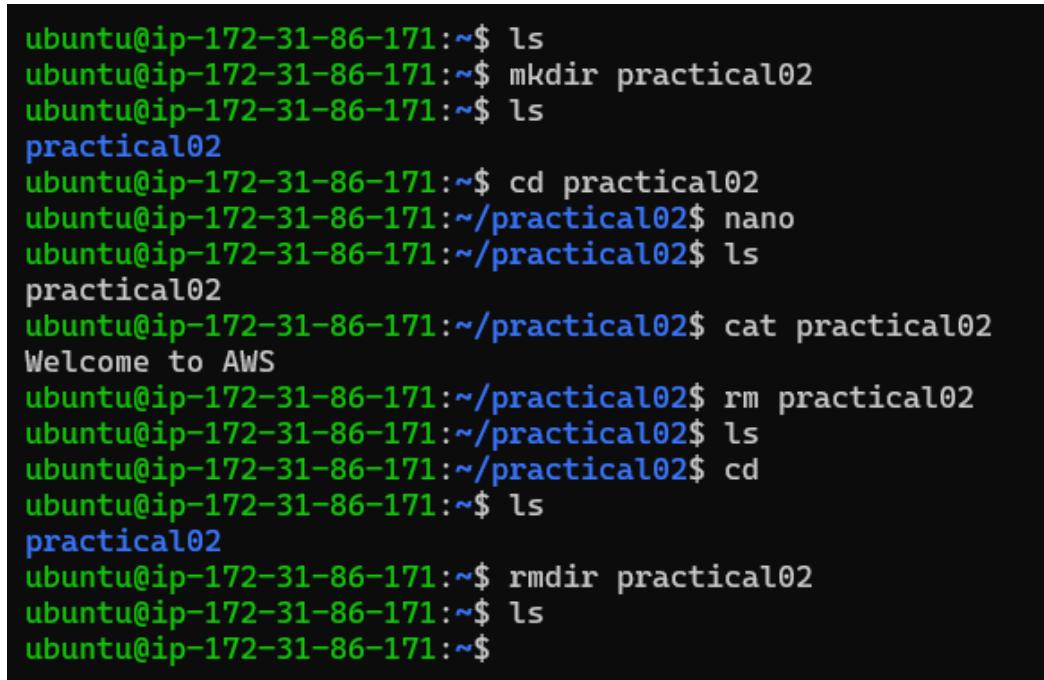
```

ubuntu@ip-172-31-86-171:~$ ls
ubuntu@ip-172-31-86-171:~$ mkdir practical02
ubuntu@ip-172-31-86-171:~$ ls
practical02
ubuntu@ip-172-31-86-171:~$ cd practical02
ubuntu@ip-172-31-86-171:~/practical02$ nano
ubuntu@ip-172-31-86-171:~/practical02$ ls
practical02
ubuntu@ip-172-31-86-171:~/practical02$ cat practical02
Welcome to AWS
ubuntu@ip-172-31-86-171:~/practical02$ rm practical02
ubuntu@ip-172-31-86-171:~/practical02$ ls
ubuntu@ip-172-31-86-171:~/practical02$ cd
ubuntu@ip-172-31-86-171:~$ ls
practical02
ubuntu@ip-172-31-86-171:~$ rmdir practical02
ubuntu@ip-172-31-86-171:~$ ls
ubuntu@ip-172-31-86-171:~$ 

```

Step 13 : After Executing The Example Code, The Instance Will Run In The Terminal Gracefully. Enjoy The **Virtual Machine Instance And Run Linux Commands** To Observe Their Outputs.

Snapshot :



```

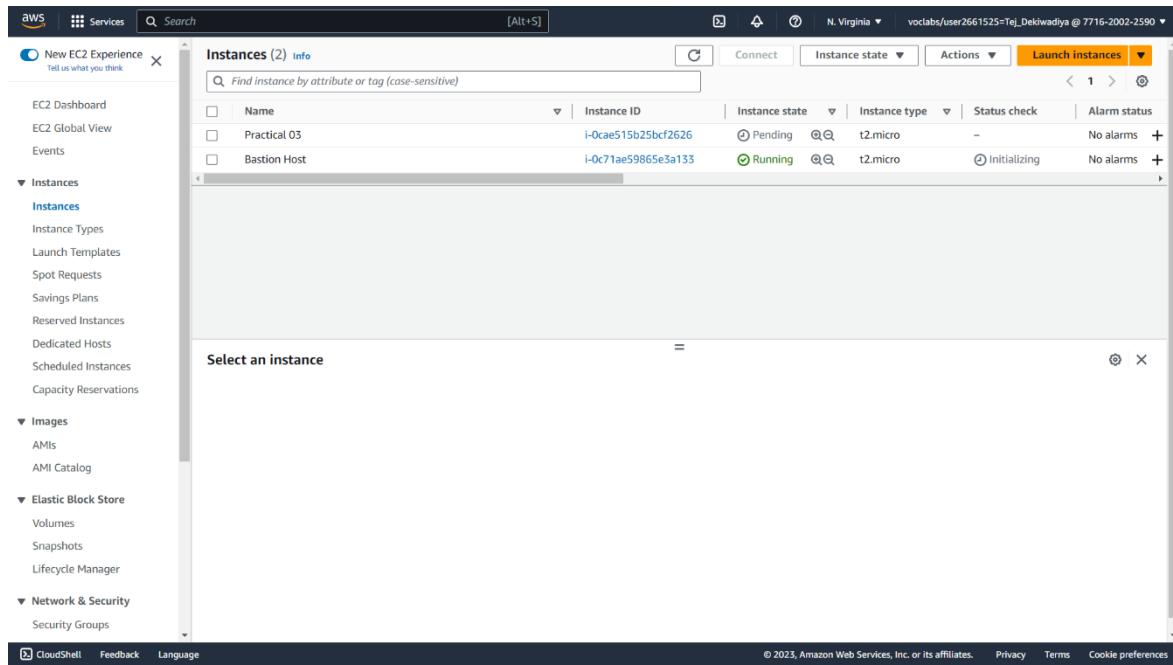
ubuntu@ip-172-31-86-171:~$ ls
ubuntu@ip-172-31-86-171:~$ mkdir practical02
ubuntu@ip-172-31-86-171:~$ ls
practical02
ubuntu@ip-172-31-86-171:~$ cd practical02
ubuntu@ip-172-31-86-171:~/practical02$ nano
ubuntu@ip-172-31-86-171:~/practical02$ ls
practical02
ubuntu@ip-172-31-86-171:~/practical02$ cat practical02
Welcome to AWS
ubuntu@ip-172-31-86-171:~/practical02$ rm practical02
ubuntu@ip-172-31-86-171:~/practical02$ ls
ubuntu@ip-172-31-86-171:~/practical02$ cd
ubuntu@ip-172-31-86-171:~$ ls
practical02
ubuntu@ip-172-31-86-171:~$ rmdir practical02
ubuntu@ip-172-31-86-171:~$ ls
ubuntu@ip-172-31-86-171:~$ 

```

Practical 03 : Virtual Machine Management

Step 01 : In this practical, I have changed Public IPV4 to an elastic IP address, so for that, right now, I have one existing instance.

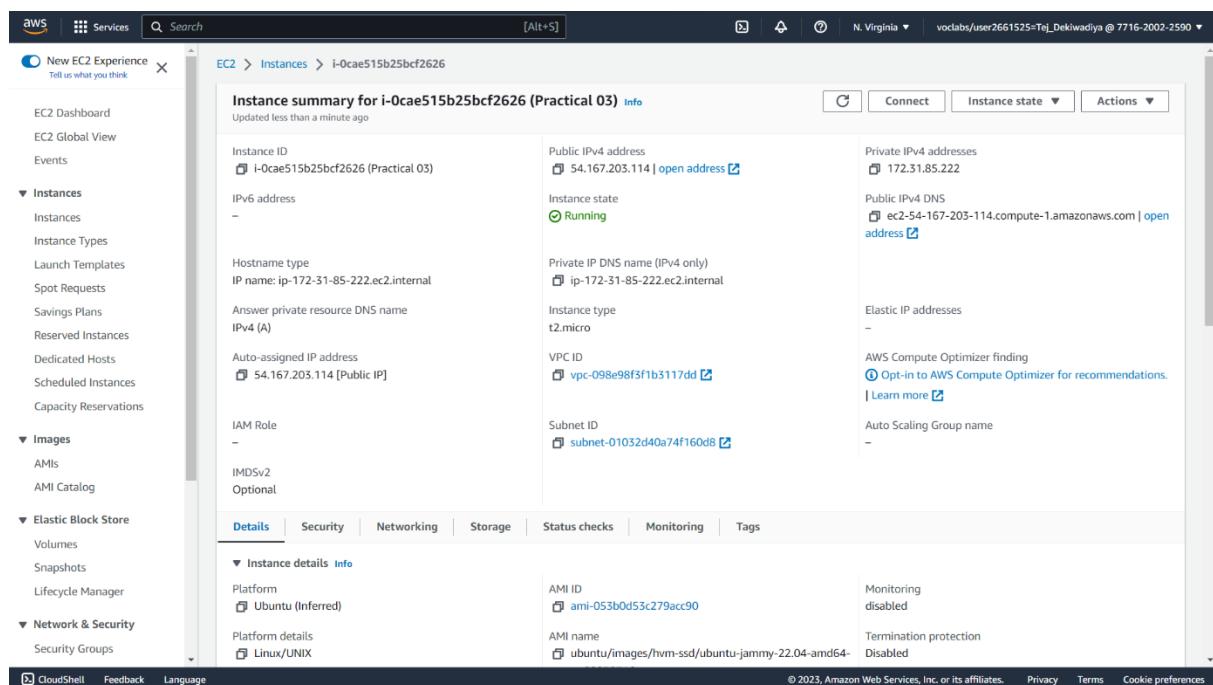
Snapshot :



Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Practical 03	i-0cae515b25bcf2626	Pending	t2.micro	-	No alarms +
Bastion Host	i-0c71ae59865e3a133	Running	t2.micro	Initializing	No alarms +

Step 02 : Here Instance name is Practical 03 and the address of Public IPV4 is 54.167.203.114

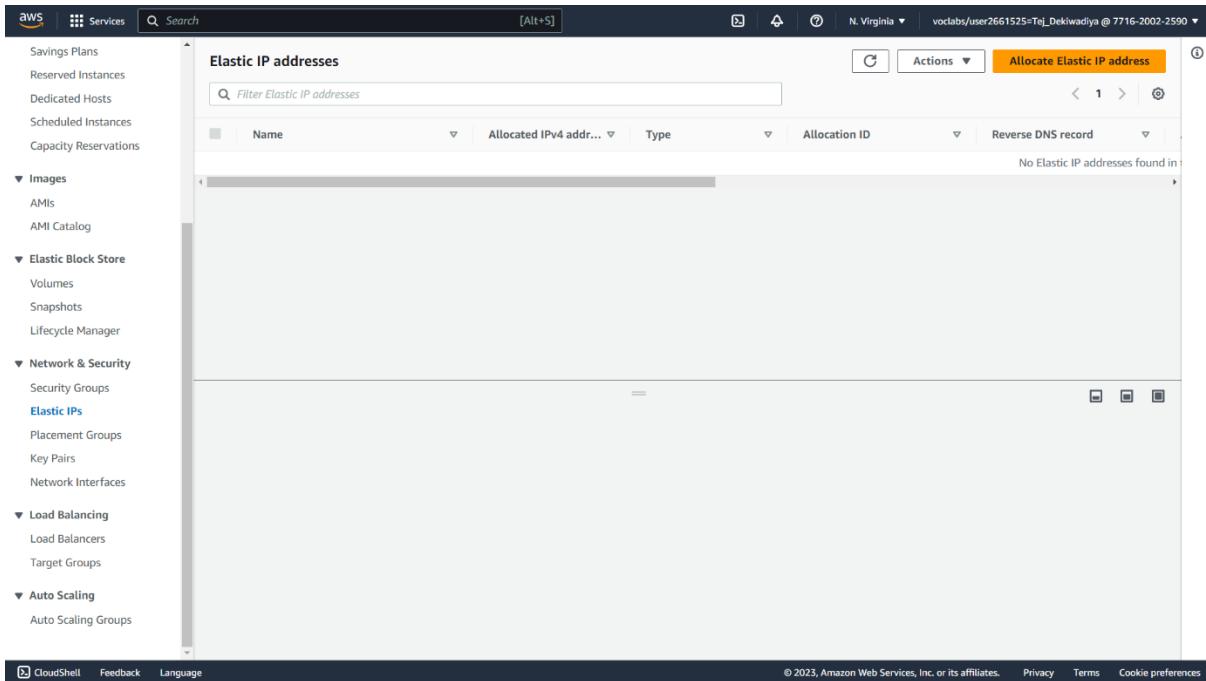
Snapshot :



Instance ID	Public IPv4 address	Private IPv4 addresses
i-0cae515b25bcf2626 (Practical 03)	54.167.203.114 [open address]	172.31.85.222
IPv6 address	Instance state	Public IP4 DNS
-	Running	ec2-54-167-203-114.compute-1.amazonaws.com [open address]
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-85-222.ec2.internal	ip-172-31-85-222.ec2.internal	-
Answer private resource DNS name	Instance type	AWS Compute Optimizer finding
IPV4 (A)	t2.micro	Opt-in to AWS Compute Optimizer for recommendations. [Learn more]
Auto-assigned IP address	VPC ID	Auto Scaling Group name
54.167.203.114 [Public IP]	vpc-098e98f3f1b3117dd	-
IAM Role	Subnet ID	
-	subnet-01032d40a74f160d8	
IMDSv2		
Optional		

Step 03 : Now Open Elastic IP Address Window.

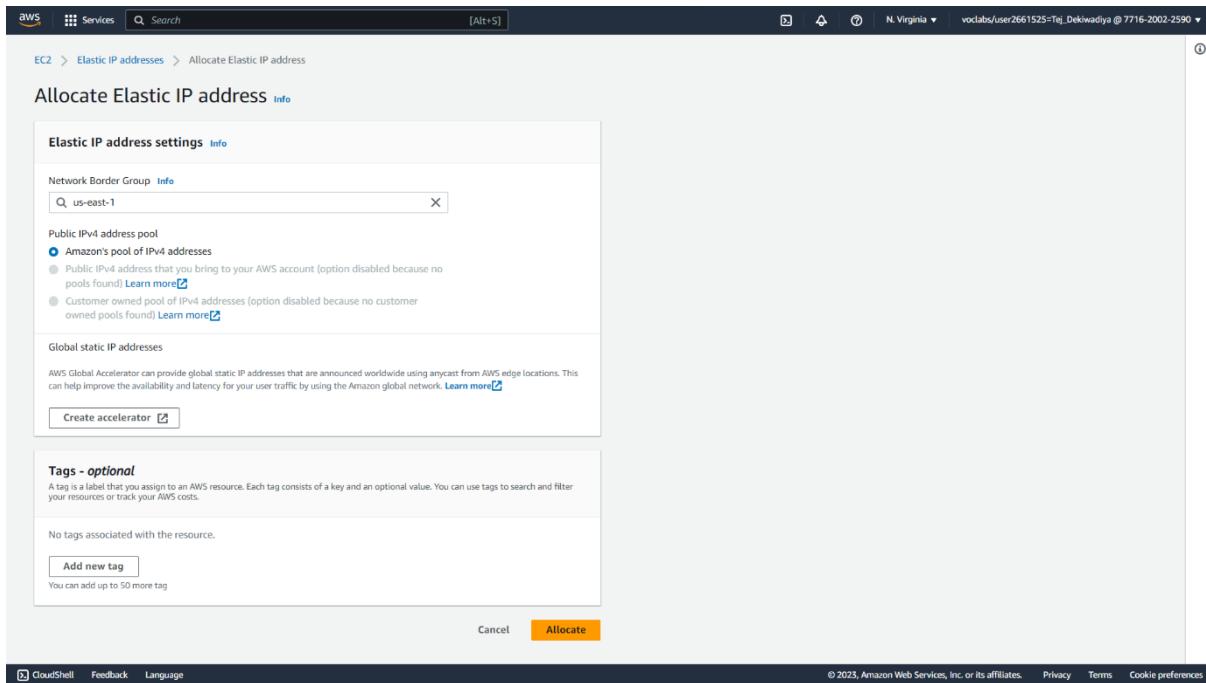
Snapshot :



The screenshot shows the AWS Management Console interface. The left sidebar is expanded, showing various services like Savings Plans, Reserved Instances, Dedicated Hosts, etc., under the 'Services' category. Under 'Network & Security', the 'Elastic IPs' option is selected. The main content area is titled 'Elastic IP addresses' and contains a table with columns: Name, Allocated IPv4 addr..., Type, Allocation ID, and Reverse DNS record. A search bar at the top of the table allows filtering. An orange button labeled 'Allocate Elastic IP address' is located in the top right corner of the table area. The bottom of the screen shows the AWS footer with links for CloudShell, Feedback, Language, and copyright information.

Step 04 : After that click on Allocate Elastic IP Address and then Choose Network Border group it means our Instance ID.

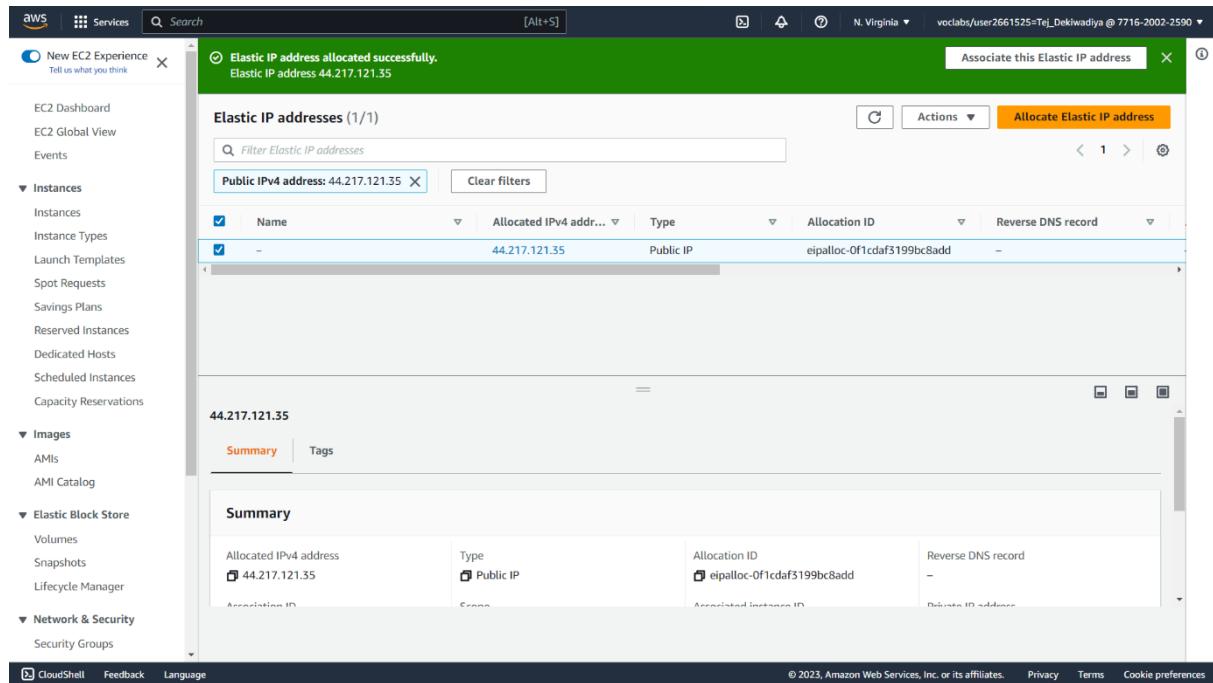
Snapshot :



The screenshot shows the 'Allocate Elastic IP address' configuration page. At the top, there's a breadcrumb trail: EC2 > Elastic IP addresses > Allocate Elastic IP address. The main section is titled 'Allocate Elastic IP address' with a 'Info' link. It contains several configuration fields: 'Elastic IP address settings' (with a 'Info' link), 'Network Border Group' (set to 'us-east-1'), 'Public IPv4 address pool' (selected 'Amazon's pool of IPv4 addresses'), 'Global static IP addresses' (with a note about AWS Global Accelerator), and a 'Create accelerator' button. Below these are sections for 'Tags - optional' (with a note about tags) and 'Add new tag' (with a note about tag limits). At the bottom of the page are 'Cancel' and 'Allocate' buttons, along with the standard AWS footer.

Step 05 : Here allocated New Elastic IP Address.

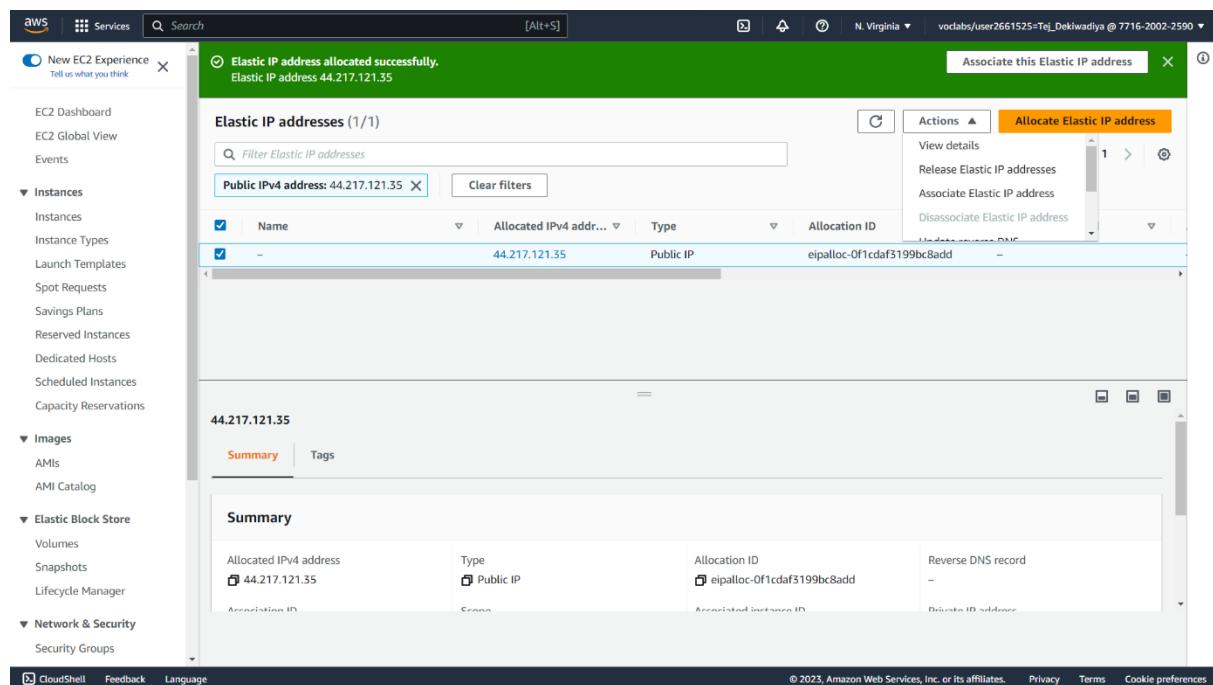
Snapshot :



The screenshot shows the AWS EC2 service dashboard. In the center, the 'Elastic IP addresses' section displays one entry: 'Elastic IP address allocated successfully. Elastic IP address 44.217.121.35'. Below this, a table lists the allocated IP address. The table has columns: Name, Allocated IPv4 address, Type, Allocation ID, and Reverse DNS record. The single row shows: Name '-' (checkbox checked), Allocated IPv4 address '44.217.121.35', Type 'Public IP', Allocation ID 'eipalloc-0f1cdaf3199bc8add', and Reverse DNS record '-'. At the bottom of the table, there are tabs for 'Summary' and 'Tags'. On the right side of the table, there are 'Actions' and 'Associate Elastic IP address' buttons. The left sidebar shows navigation links for EC2 Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, and Network & Security.

Step 06 : In Elastic IP Address tab click on action and choose Associate Elastic IP Address.

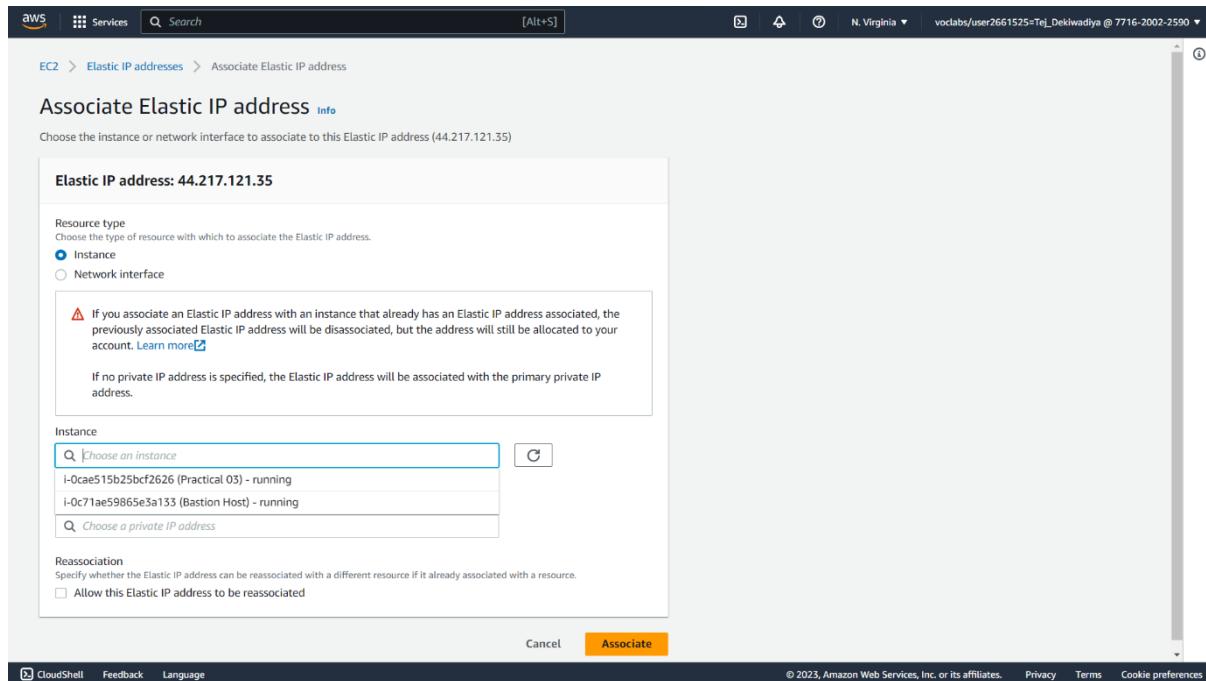
Snapshot :



This screenshot is identical to the previous one, showing the successful allocation of the Elastic IP address. However, the 'Actions' dropdown menu is open on the right side of the table. The dropdown options include 'View details', 'Release Elastic IP addresses', 'Associate Elastic IP address' (which is highlighted in blue), and 'Disassociate Elastic IP address'. The rest of the interface, including the table data and sidebar, remains the same.

Step 07 : Now here in a instance choose our Instance and Submit.

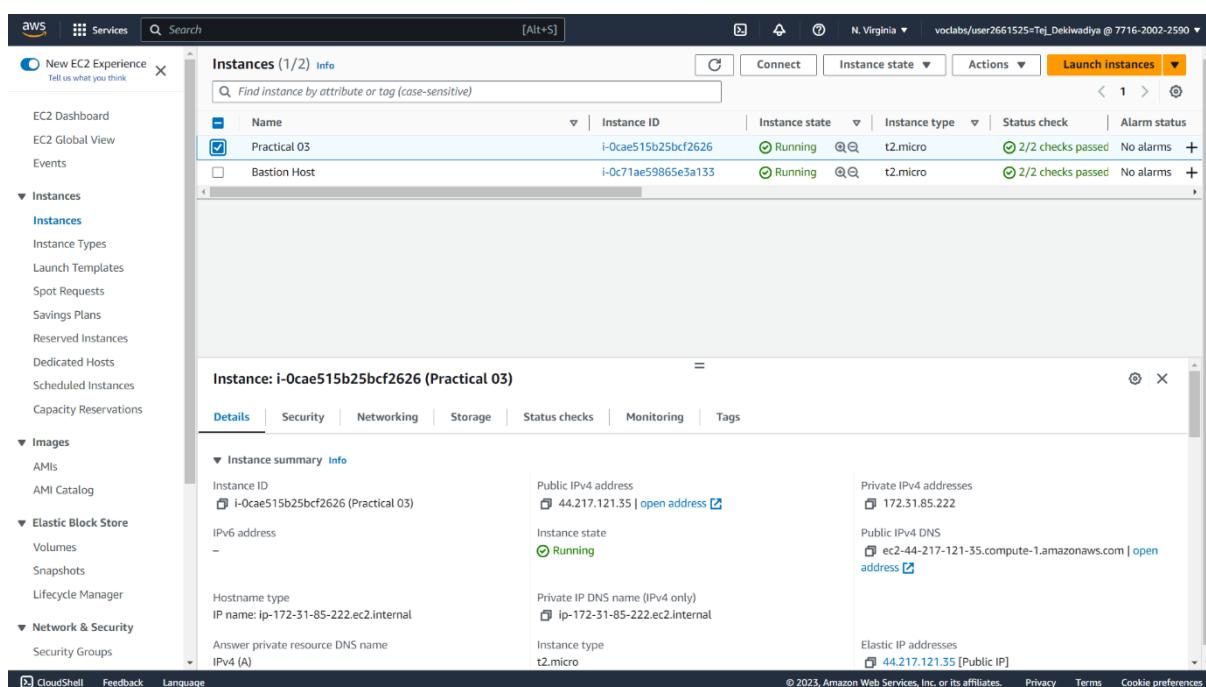
Snapshot :



The screenshot shows the 'Associate Elastic IP address' page in the AWS EC2 service. The 'Elastic IP address' dropdown is set to '44.217.121.35'. Under 'Resource type', 'Instance' is selected. A warning message states: 'If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account.' Below this, it says 'If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.' In the 'Instance' section, 'Choose an instance' is selected, showing two options: 'i-0cae515b25bcf2626 (Practical 03) - running' and 'i-0c71ae59865e3a133 (Bastion Host) - running'. At the bottom, there is a checkbox 'Allow this Elastic IP address to be reassociated' and a yellow 'Associate' button.

Step 08 : At last Public IPV4 Address changed with Elastic IP Address -> 44.217.121.35

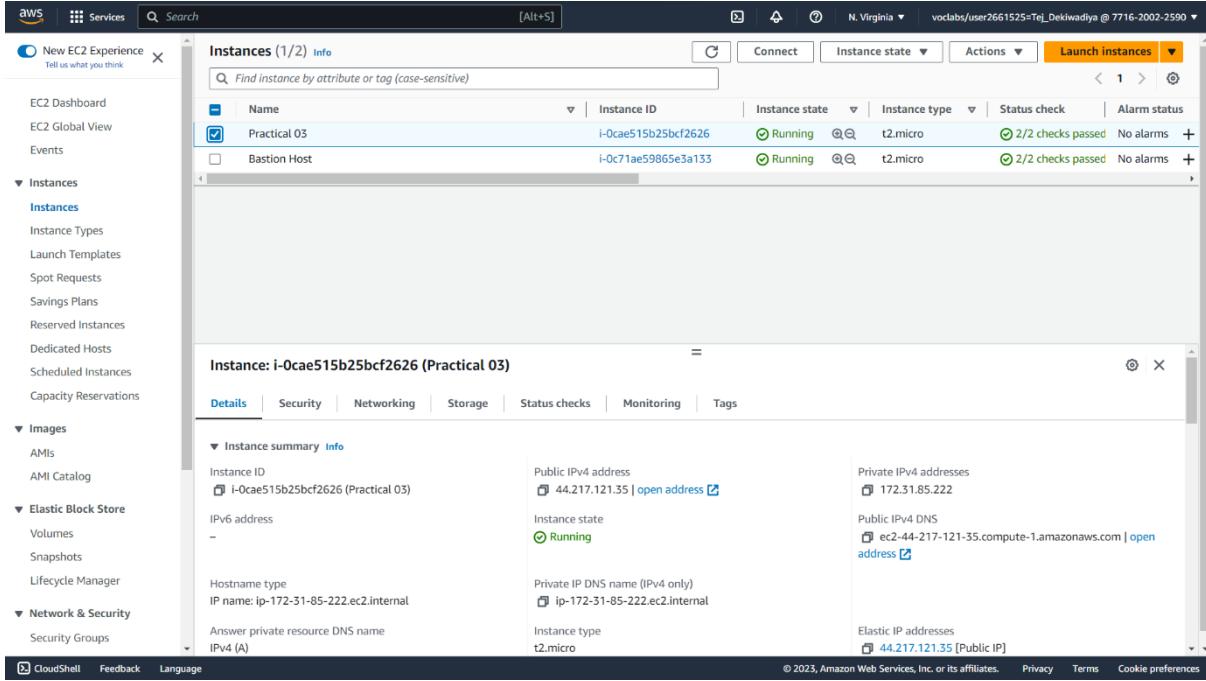
Snapshot :



The screenshot shows the 'Instances (1/2)' page in the AWS EC2 service. The table lists two instances: 'Practical 03' (selected) and 'Bastion Host'. The 'Practical 03' row shows 'Public IPv4 address' as '44.217.121.35'. Below the table, the 'Instance: i-0cae515b25bcf2626 (Practical 03)' details pane is open. It shows the 'Details' tab with various configuration settings, including 'Public IPv4 address' set to '44.217.121.35' and 'Private IP4 address' set to '172.31.85.222'.

Step 09 : At last Public IPV4 Address changed with Elastic IP Address → 44.217.121.35

Snapshot :



Instances (1/2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Practical 03	i-0cae515b25bcf2626	Running	t2.micro	2/2 checks passed	No alarms
Bastion Host	i-0c71ae59865e3a133	Running	t2.micro	2/2 checks passed	No alarms

Instance: i-0cae515b25bcf2626 (Practical 03)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

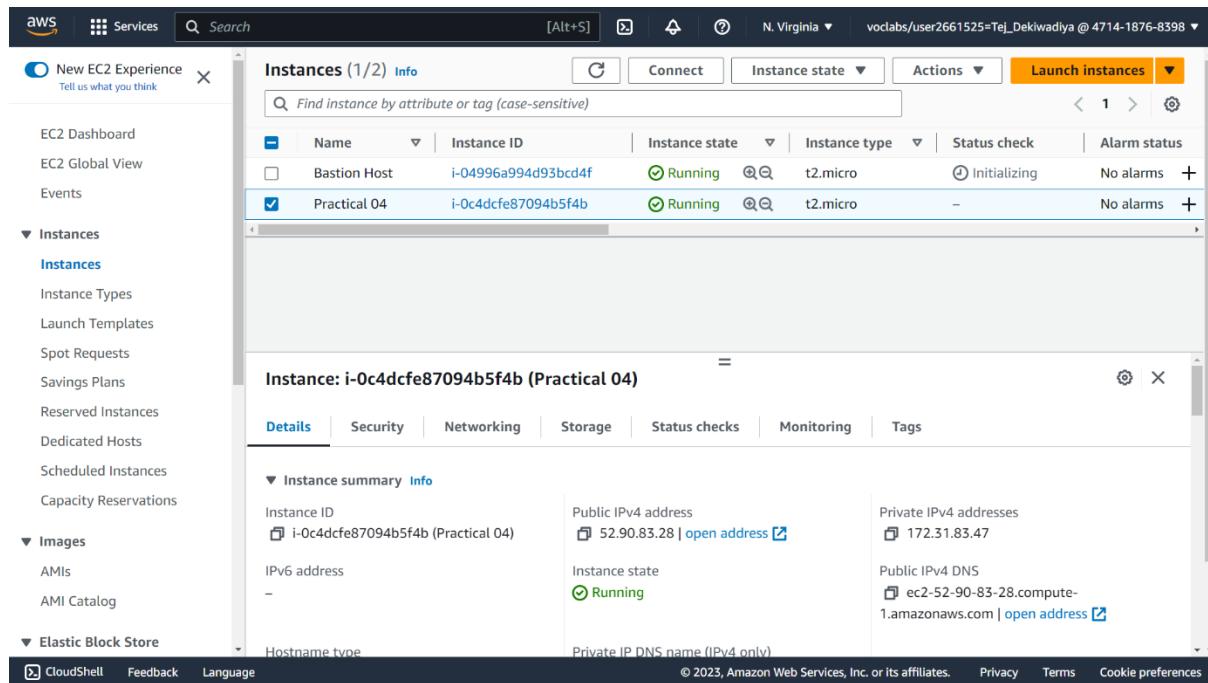
Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0cae515b25bcf2626 (Practical 03)	44.217.121.35 open address	172.31.85.222
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-44-217-121-35.compute-1.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-85-222.ec2.internal	ip-172-31-85-222.ec2.internal	44.217.121.35 [Public IP]
Answer private resource DNS name	Instance type	
IPv4 (A)	t2.micro	

Practical 04 : Deploy a Web Application on AWS

Step 01 : Create one new Instance and whenever create Instance choose all the settings by default.

Snapshot :



Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Bastion Host	i-04996a994d93bcd4f	Running	t2.micro	Initializing	No alarms
Practical 04	i-0c4dcfe87094b5f4b	Running	t2.micro	-	No alarms

Instance: i-0c4dcfe87094b5f4b (Practical 04)

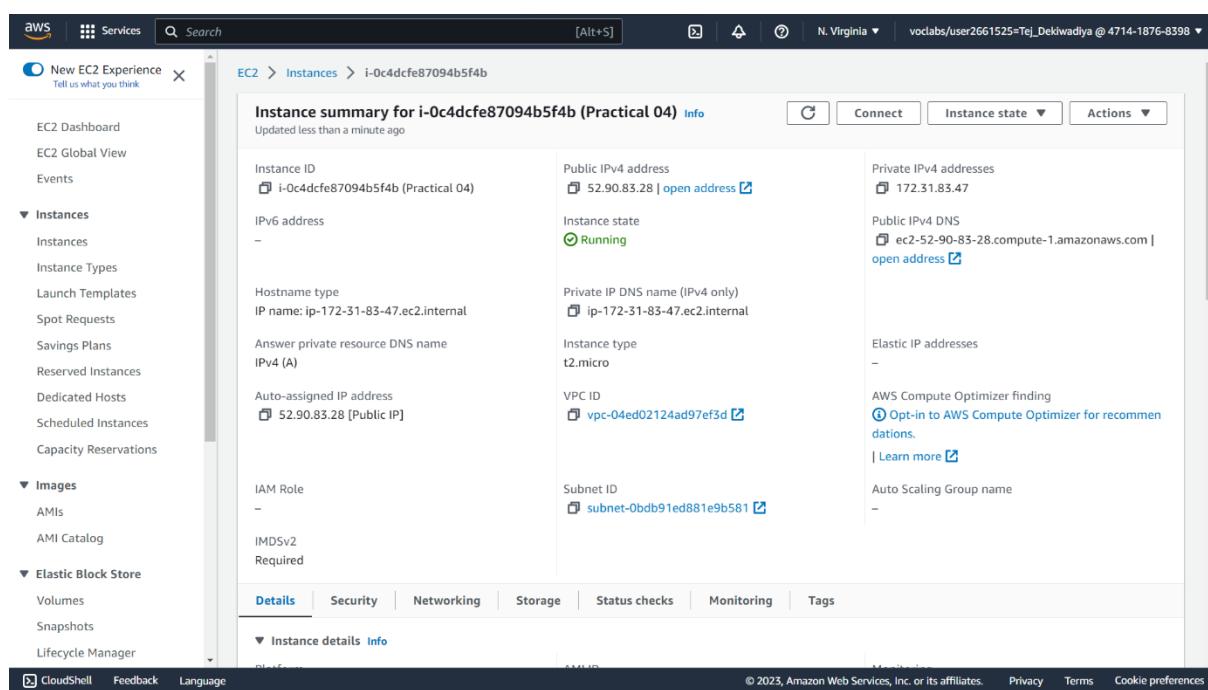
Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary

Instance ID i-0c4dcfe87094b5f4b (Practical 04)	Public IPv4 address 52.90.83.28 open address	Private IPv4 addresses 172.31.83.47
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-52-90-83-28.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-83-47.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-83-47.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 52.90.83.28 [Public IP]	VPC ID vpc-04ed02124ad97ef3d	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-0bdb91ed881e9b581	
IMDSv2 Required		

Step 02 : Check all the configurations of your Instance .

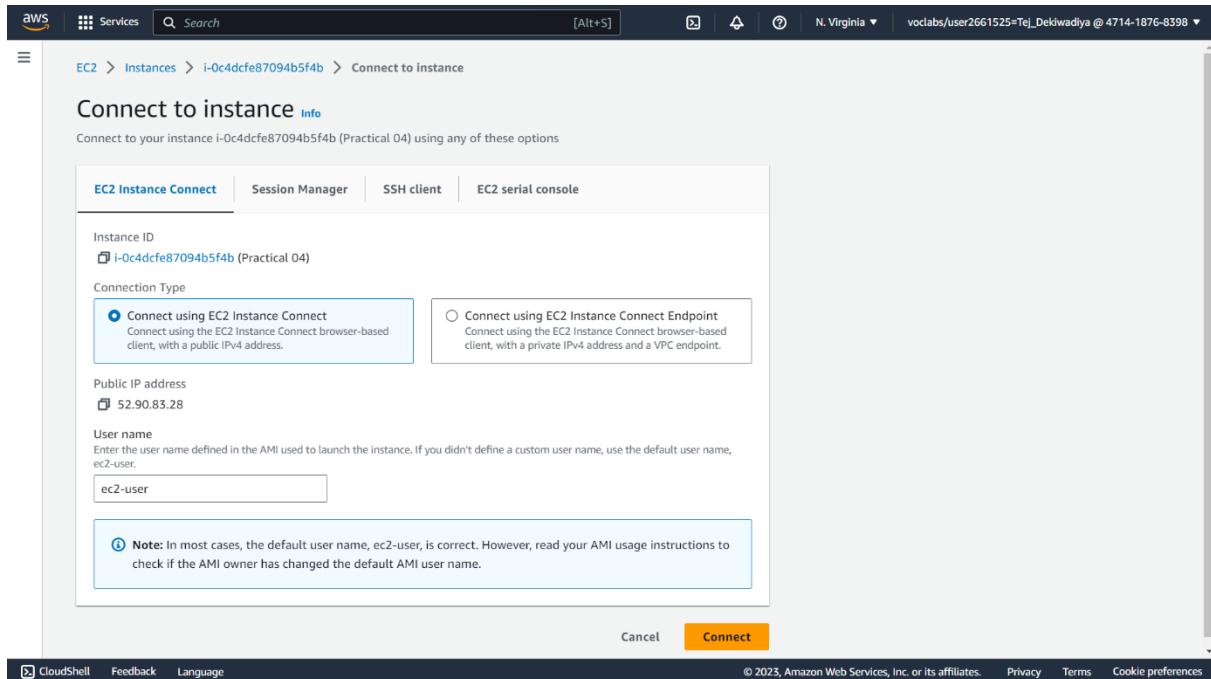
Snapshot :



Instance ID	Public IPv4 address	Private IPv4 addresses
i-0c4dcfe87094b5f4b (Practical 04)	52.90.83.28 open address	172.31.83.47
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-52-90-83-28.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-83-47.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-83-47.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 52.90.83.28 [Public IP]	VPC ID vpc-04ed02124ad97ef3d	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-0bdb91ed881e9b581	
IMDSv2 Required		

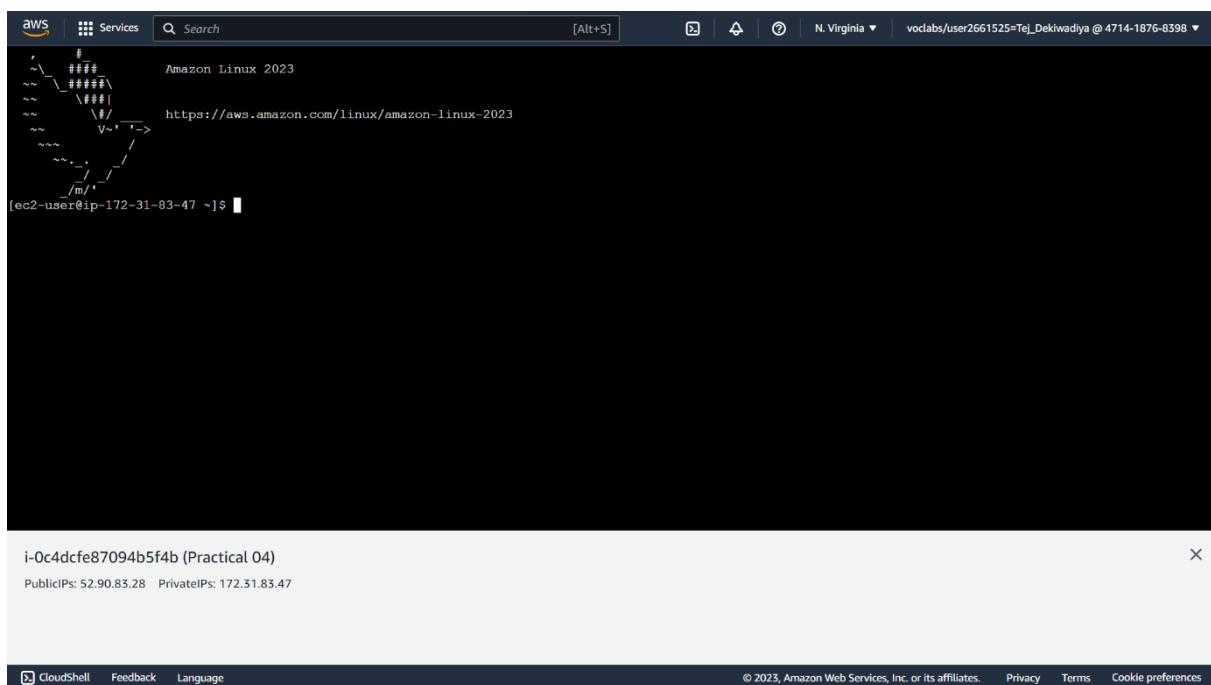
Step 03 : Now connect Instance with Terminal.

Snapshot :



Step 04 : Kali default Terminal Opened.

Snapshot :

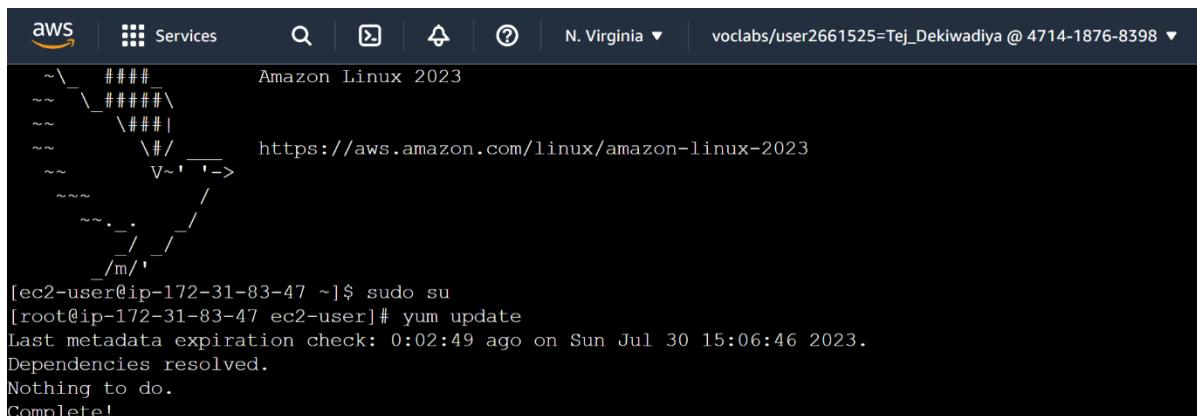


Step 05 : Now here first of all take root permission using **sudo su**

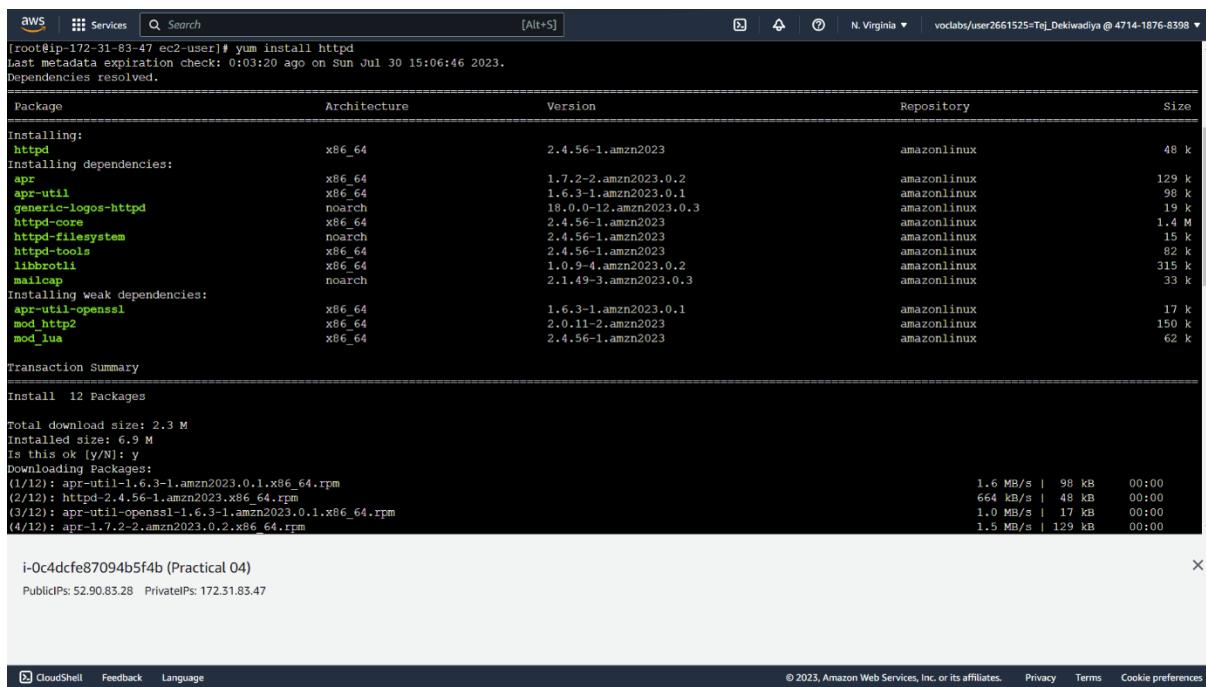
After that update the terminal through command **yum update**

Then install httpd using command **yum install httpd**

Snapshot :



```
[ec2-user@ip-172-31-83-47 ~]$ sudo su
[root@ip-172-31-83-47 ec2-user]# yum update
Last metadata expiration check: 0:02:49 ago on Sun Jul 30 15:06:46 2023.
Dependencies resolved.
Nothing to do.
Complete!
```

```
root@ip-172-31-83-47 ec2-user]# yum install httpd
Last metadata expiration check: 0:03:20 ago on Sun Jul 30 15:06:46 2023.
Dependencies resolved.

Transaction Summary
Install 12 Packages

Total download size: 2.3 M
Installed size: 6.9 M
Is this ok [y/N]: y
Downloading Packages:
(1/12): apr-util-1.6.3-1.amzn2023.0.1.x86_64.rpm 1.6 MB/s | 98 KB 00:00
(2/12): httpd-2.4.56-1.amzn2023.x86_64.rpm 664 kB/s | 49 kB 00:00
(3/12): apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64.rpm 1.0 MB/s | 17 kB 00:00
(4/12): apr-1.7.2-2.amzn2023.0.2.x86_64.rpm 1.5 MB/s | 129 kB 00:00

i-0c4dcfe87094b5f4b (Practical 04)
PublicIPs: 52.90.83.28 PrivateIPs: 172.31.83.47
```

CloudShell Feedback Language Privacy Terms Cookie preferences

© 2023, Amazon Web Services, Inc. or its affiliates.

```

aws Services Search [Alt+S] N. Virginia v vocabs/user2661525=Tej_Dekiwadiya @ 4714-1876-8398 ▾
(5/12): libbrotli-1.0.9-4.amzn2023.0.2.x86_64.rpm          13 MB/s | 315 KB  00:00
(6/12): httpd-core-2.4.56-1.amzn2023.x86_64.rpm           32 MB/s | 1.4 MB  00:00
(7/12): mod_lua-2.4.56-1.amzn2023.x86_64.rpm             1.9 MB/s | 62 kB   00:00
(8/12): httpd-tools-2.4.56-1.amzn2023.x86_64.rpm          4.2 MB/s | 82 kB   00:00
(9/12): generic-logos-httd-18.0.0-12.amzn2023.0.3.noarch.rpm 1.2 MB/s | 19 kB   00:00
(10/12): mod_http2-2.0.11-2.amzn2023.x86_64.rpm          6.6 MB/s | 150 kB  00:00
(11/12): mailcap-2.1.49-3.amzn2023.0.3.noarch.rpm        1.3 MB/s | 33 kB   00:00
(12/12): httpd-filesystem-2.4.56-1.amzn2023.noarch.rpm    1.0 MB/s | 15 kB   00:00
Total                                         11 MB/s | 2.3 MB  00:00

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Installing : apr-1.7.2-2.amzn2023.0.2.x86_64          1/12
Installing : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 2/12
Installing : apr-util-1.6.3-1.amzn2023.0.1.x86_64       3/12
Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch     4/12
Installing : httpd-tools-2.4.56-1.amzn2023.x86_64      5/12
Running scriptlet: httpd-filesystem-2.4.56-1.amzn2023.noarch 6/12
Installing : httpd-filesystem-2.4.56-1.amzn2023.noarch 6/12
Installing : httpd-core-2.4.56-1.amzn2023.x86_64       7/12
Installing : mod_lua-2.4.56-1.amzn2023.x86_64         8/12
Installing : mod_http2-2.0.11-2.amzn2023.x86_64       9/12
Installing : generic-logos-httd-18.0.0-12.amzn2023.0.3.noarch 10/12
Installing : libbrotli-1.0.9-4.amzn2023.0.2.x86_64     11/12
Installing : httpd-2.4.56-1.amzn2023.x86_64          12/12
Running scriptlet: httpd-2.4.56-1.amzn2023.x86_64      12/12
Verifying  : apr-1.7.2-2.amzn2023.0.2.x86_64          1/12
Verifying  : httpd-2.4.56-1.amzn2023.x86_64          2/12
Verifying  : apr-util-1.6.3-1.amzn2023.0.1.x86_64      3/12
Verifying  : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 4/12
i-0c4dcfe87094b5f4b (Practical 04)
PublicIPs: 52.90.83.28 PrivatelPs: 172.31.83.47

```

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```

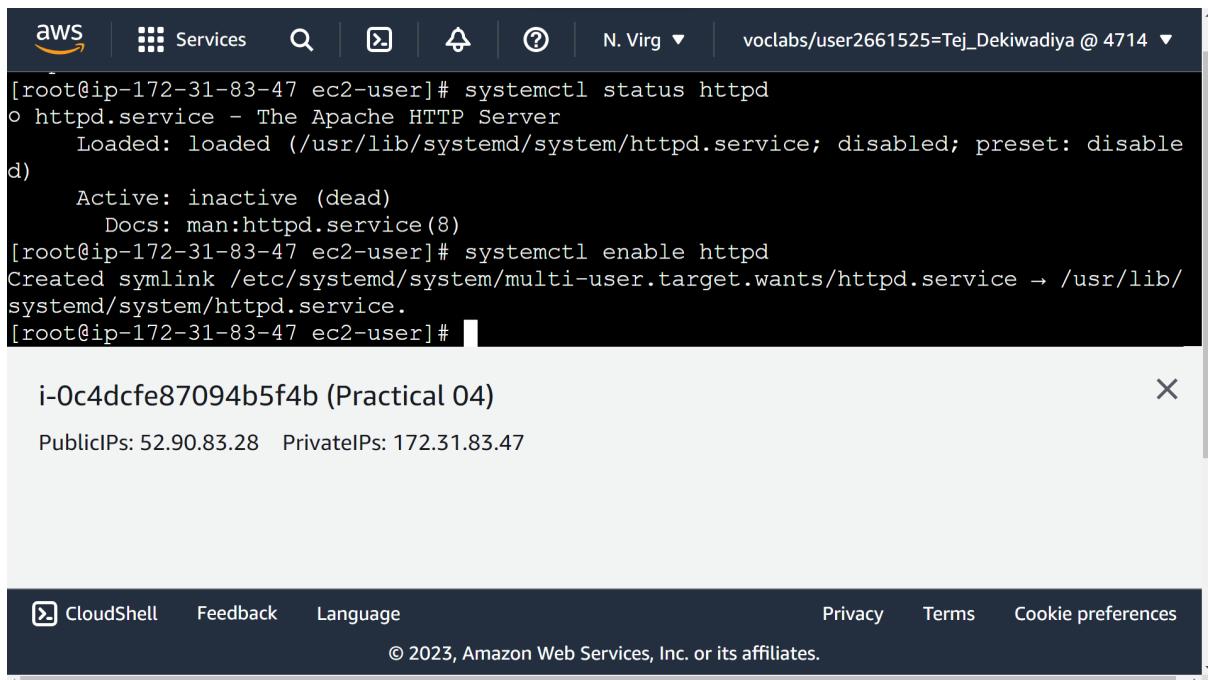
aws Services Search [Alt+S] N. Virginia v vocabs/user2661525=Tej_Dekiwadiya @ 4714-1876-8398 ▾
Installing : apr-1.7.2-2.amzn2023.0.2.x86_64          1/12
Installing : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 2/12
Installing : apr-util-1.6.3-1.amzn2023.0.1.x86_64       3/12
Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch     4/12
Installing : httpd-tools-2.4.56-1.amzn2023.x86_64      5/12
Running scriptlet: httpd-filesystem-2.4.56-1.amzn2023.noarch 6/12
Installing : httpd-filesystem-2.4.56-1.amzn2023.noarch 6/12
Installing : httpd-core-2.4.56-1.amzn2023.x86_64       7/12
Installing : mod_lua-2.4.56-1.amzn2023.x86_64         8/12
Installing : mod_http2-2.0.11-2.amzn2023.x86_64       9/12
Installing : generic-logos-httd-18.0.0-12.amzn2023.0.3.noarch 10/12
Installing : libbrotli-1.0.9-4.amzn2023.0.2.x86_64     11/12
Installing : httpd-2.4.56-1.amzn2023.x86_64          12/12
Running scriptlet: httpd-2.4.56-1.amzn2023.x86_64      12/12
Verifying  : apr-1.7.2-2.amzn2023.0.2.x86_64          1/12
Verifying  : httpd-2.4.56-1.amzn2023.x86_64          2/12
Verifying  : apr-util-1.6.3-1.amzn2023.0.1.x86_64      3/12
Verifying  : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 4/12
Verifying  : httpd-core-2.4.56-1.amzn2023.x86_64       5/12
Verifying  : libbrotli-1.0.9-4.amzn2023.0.2.x86_64     6/12
Verifying  : mod_lua-2.4.56-1.amzn2023.x86_64         7/12
Verifying  : httpd-tools-2.4.56-1.amzn2023.x86_64       8/12
Verifying  : mod_http2-2.0.11-2.amzn2023.x86_64       9/12
Verifying  : mailcap-2.1.49-3.amzn2023.0.3.noarch     10/12
Verifying  : generic-logos-httd-18.0.0-12.amzn2023.0.3.noarch 11/12
Verifying  : httpd-filesystem-2.4.56-1.amzn2023.noarch 12/12
Complete!
[root@ip-172-31-83-47 ec2-user] # i-0c4dcfe87094b5f4b (Practical 04)
PublicIPs: 52.90.83.28 PrivatelPs: 172.31.83.47

```

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 06 : Now check the status of **httpd** → **systemctl status httpd**
 Then active the status of **httpd** → **systemctl enable httpd**

Snapshot :



```
[root@ip-172-31-83-47 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
[root@ip-172-31-83-47 ec2-user]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-83-47 ec2-user]# 
```

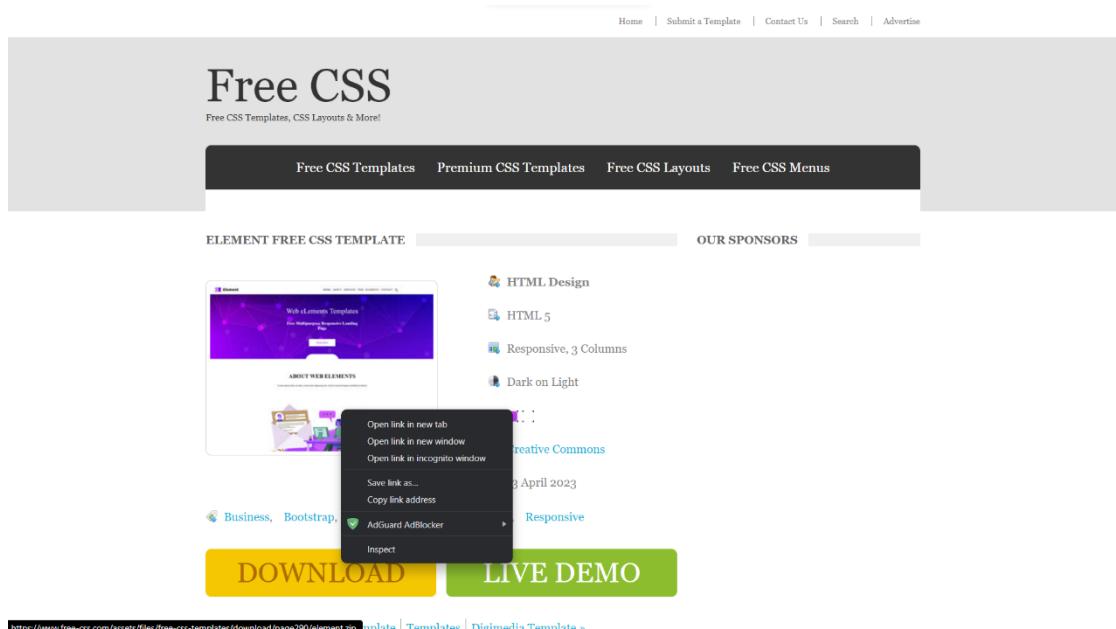
i-0c4dcfe87094b5f4b (Practical 04) X

PublicIPs: 52.90.83.28 PrivateIPs: 172.31.83.47

CloudShell Feedback Language Privacy Terms Cookie preferences
 © 2023, Amazon Web Services, Inc. or its affiliates.

Step 07 : Choose any template from Internet and on download button through right click select copy link address.

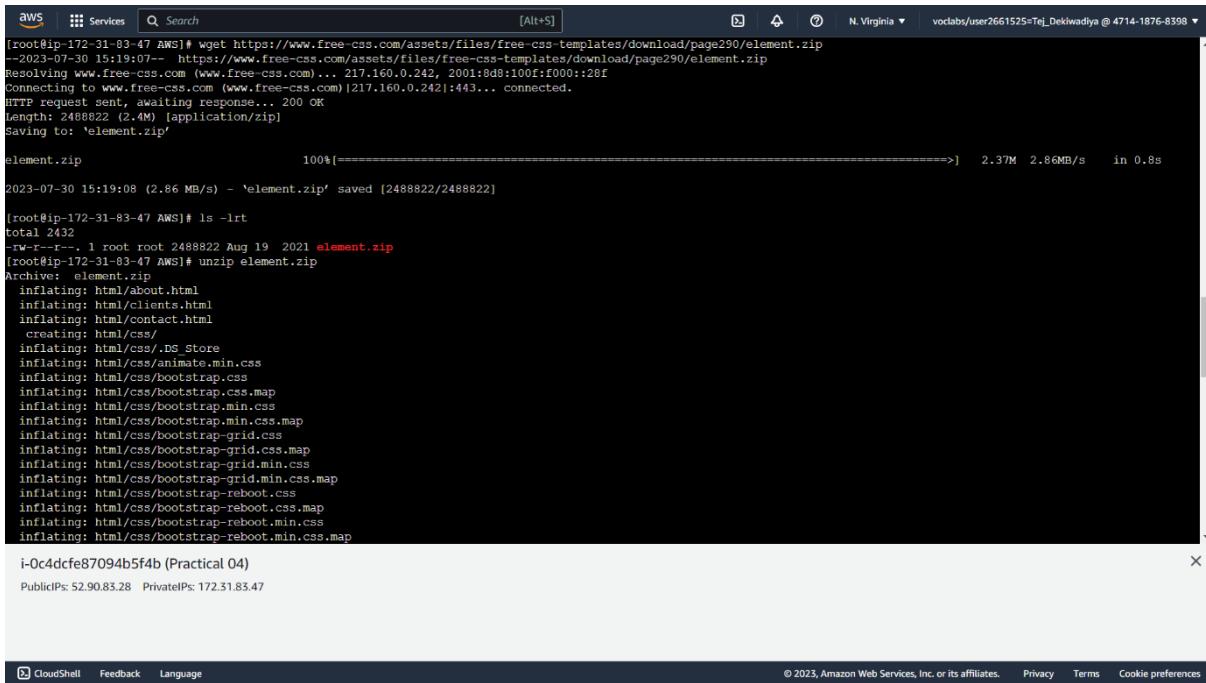
Snapshot :



Step 08 : Paste url link in terminal → wget url_link

In that link attached one zip file after that unzip that file → unzip filename

Snapshot :



```

aws Services Search [Alt+S] N. Virginia vocabs/user2661525=Tej_Dekiwadiya @ 4714-1876-8398
[root@ip-172-31-83-47 AWS]# wget https://www.free-css.com/assets/files/free-css-templates/download/page290/element.zip
--2023-07-30 15:19:07-- https://www.free-css.com/assets/files/free-css-templates/download/page290/element.zip
Resolving www.free-css.com (www.free-css.com)... 217.160.0.242, 2001:8d8:100f:f000::28f
Connecting to www.free-css.com (www.free-css.com)|217.160.0.242|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2408822 (2.4M) [application/zip]
Saving to: 'element.zip'

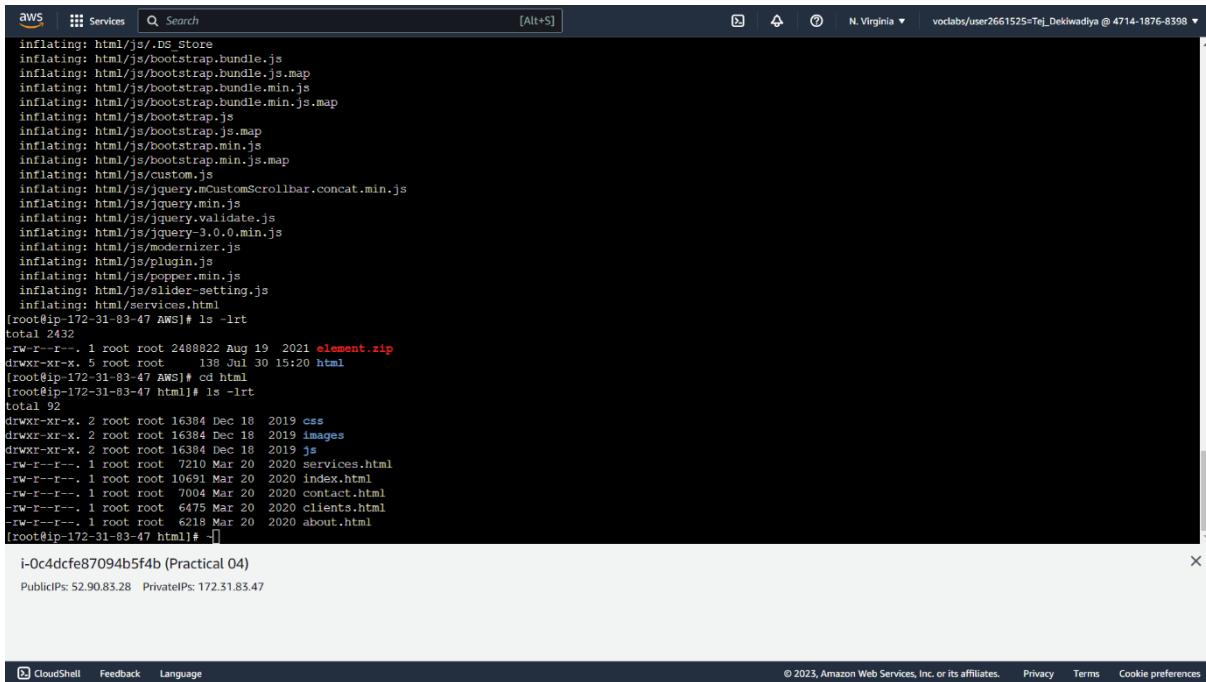
element.zip          100%[=====] 2.37M 2.86MB/s   in 0.8s

2023-07-30 15:19:08 (2.86 MB/s) - 'element.zip' saved [2408822/2408822]

[root@ip-172-31-83-47 AWS]# ls -lrt
total 2432
-rw-r--r--, 1 root root 2408822 Aug 19 2021 element.zip
[root@ip-172-31-83-47 AWS]# unzip element.zip
Archive: element.zip
  inflating: html/about.html
  inflating: html/clients.html
  inflating: html/contact.html
  creating: html/css/
  inflating: html/css/.DS_Store
  inflating: html/css/animate.min.css
  inflating: html/css/bootstrap.css
  inflating: html/css/bootstrap.css.map
  inflating: html/css/bootstrap.min.css
  inflating: html/css/bootstrap.min.css.map
  inflating: html/css/bootstrap-grid.css
  inflating: html/css/bootstrap-grid.css.map
  inflating: html/css/bootstrap-grid.min.css
  inflating: html/css/bootstrap-grid.min.css.map
  inflating: html/css/bootstrap-reboot.css
  inflating: html/css/bootstrap-reboot.css.map
  inflating: html/css/bootstrap-reboot.min.css
  inflating: html/css/bootstrap-reboot.min.css.map
i-0c4dcfe87094b5f4b (Practical 04)
Public IPs: 52.90.83.28 Private IPs: 172.31.83.47

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```



```

aws Services Search [Alt+S] N. Virginia vocabs/user2661525=Tej_Dekiwadiya @ 4714-1876-8398
inflating: html/js/.DS_Store
inflating: html/js/bootstrap.bundle.js
inflating: html/js/bootstrap.bundle.js.map
inflating: html/js/bootstrap.bundle.min.js
inflating: html/js/bootstrap.bundle.min.js.map
inflating: html/js/bootstrap.js
inflating: html/js/bootstrap.js.map
inflating: html/js/bootstrap.min.js
inflating: html/js/bootstrap.min.js.map
inflating: html/js/custom.js
inflating: html/js/jquery.mCustomScrollbar.concat.min.js
inflating: html/js/jquery.min.js
inflating: html/js/jquery.validate.js
inflating: html/js/jquery-3.0.0.min.js
inflating: html/js/modernizer.js
inflating: html/js/plugin.js
inflating: html/js/popper.min.js
inflating: html/js/slider-setting.js
inflating: html/services.html
[root@ip-172-31-83-47 AWS]# ls -lrt
total 2432
-rw-r--r--, 1 root root 2408822 Aug 19 2021 element.zip
drwxr-xr-x, 2 root root 16384 Dec 18 2019 css
drwxr-xr-x, 2 root root 16384 Dec 18 2019 images
drwxr-xr-x, 2 root root 16384 Dec 18 2019 js
-rw-r--r--, 1 root root 7210 Mar 20 2020 services.html
-rw-r--r--, 1 root root 10691 Mar 20 2020 index.html
-rw-r--r--, 1 root root 7004 Mar 20 2020 contact.html
-rw-r--r--, 1 root root 6475 Mar 20 2020 clients.html
-rw-r--r--, 1 root root 6218 Mar 20 2020 about.html
[root@ip-172-31-83-47 html]#
i-0c4dcfe87094b5f4b (Practical 04)
Public IPs: 52.90.83.28 Private IPs: 172.31.83.47

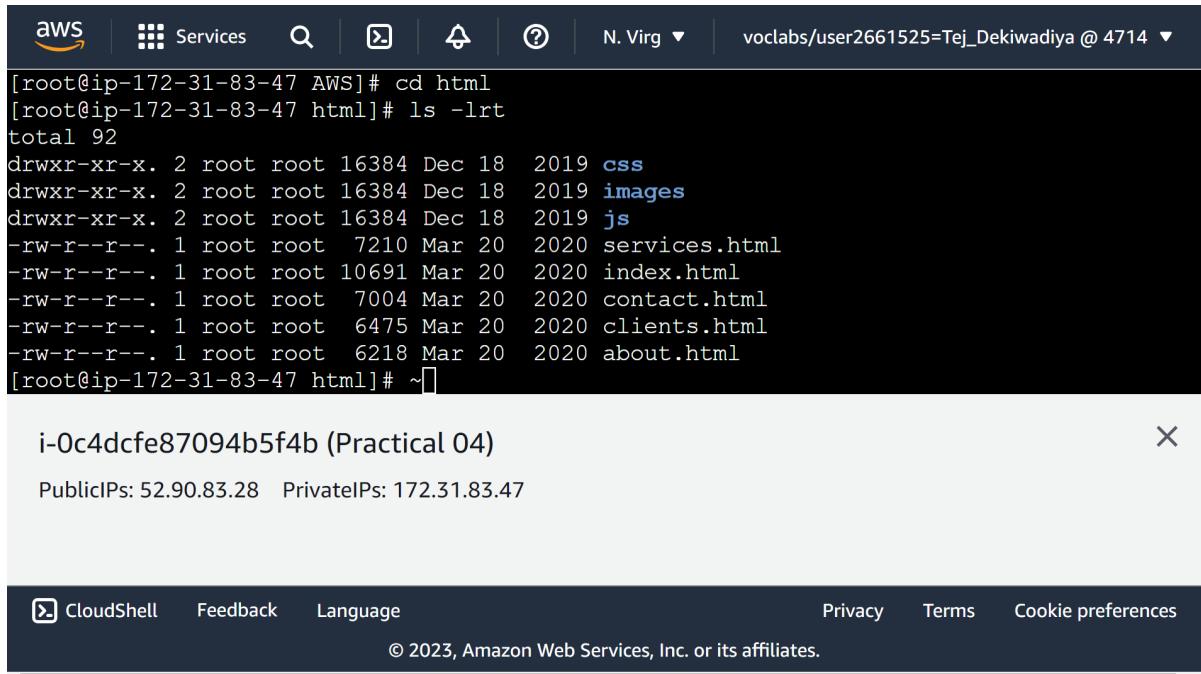
CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```

Step 09 : Now check all the files → **ls -lrt**

After that move all the file in particular directory **/var/www/html** → **mv * /var/www/html/**
 And go to that directory → **cd /var/www/html/**

Snapshot :

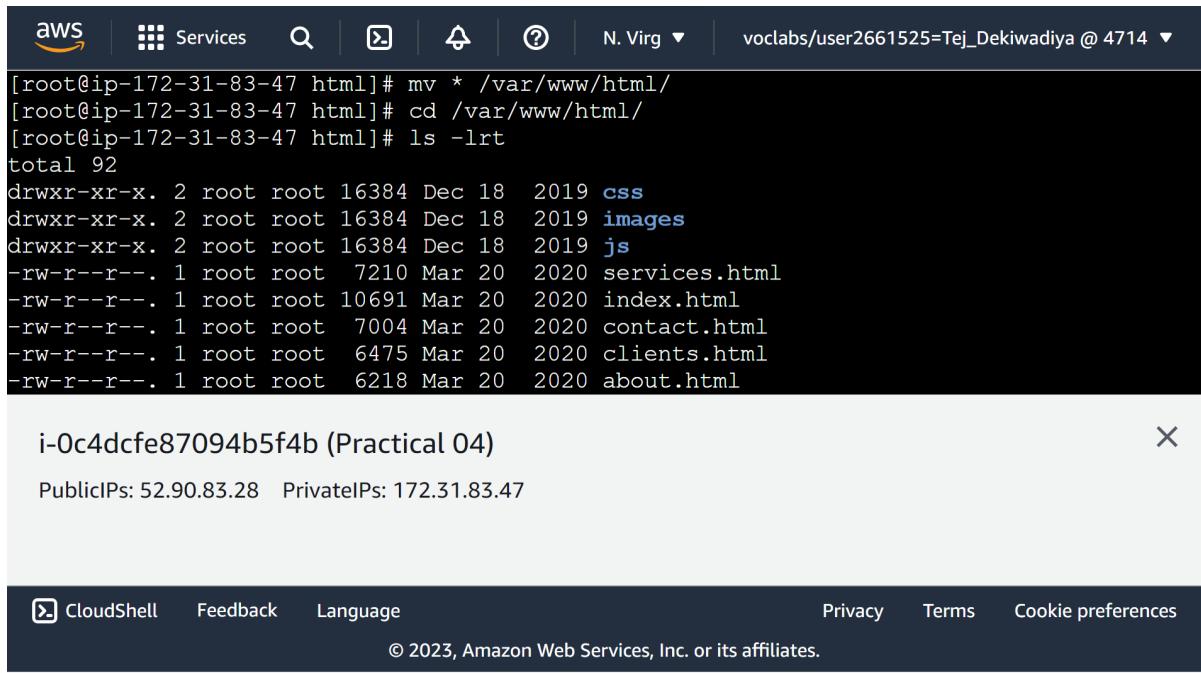


```
[root@ip-172-31-83-47 AWS]# cd html
[root@ip-172-31-83-47 html]# ls -lrt
total 92
drwxr-xr-x.. 2 root root 16384 Dec 18 2019 css
drwxr-xr-x.. 2 root root 16384 Dec 18 2019 images
drwxr-xr-x.. 2 root root 16384 Dec 18 2019 js
-rw-r--r--.. 1 root root 7210 Mar 20 2020 services.html
-rw-r--r--.. 1 root root 10691 Mar 20 2020 index.html
-rw-r--r--.. 1 root root 7004 Mar 20 2020 contact.html
-rw-r--r--.. 1 root root 6475 Mar 20 2020 clients.html
-rw-r--r--.. 1 root root 6218 Mar 20 2020 about.html
[root@ip-172-31-83-47 html]# ~[]
```

i-0c4dcfe87094b5f4b (Practical 04)

PublicIPs: 52.90.83.28 PrivateIPs: 172.31.83.47

CloudShell Feedback Language Privacy Terms Cookie preferences © 2023, Amazon Web Services, Inc. or its affiliates.



```
[root@ip-172-31-83-47 html]# mv * /var/www/html/
[root@ip-172-31-83-47 html]# cd /var/www/html/
[root@ip-172-31-83-47 html]# ls -lrt
total 92
drwxr-xr-x.. 2 root root 16384 Dec 18 2019 css
drwxr-xr-x.. 2 root root 16384 Dec 18 2019 images
drwxr-xr-x.. 2 root root 16384 Dec 18 2019 js
-rw-r--r--.. 1 root root 7210 Mar 20 2020 services.html
-rw-r--r--.. 1 root root 10691 Mar 20 2020 index.html
-rw-r--r--.. 1 root root 7004 Mar 20 2020 contact.html
-rw-r--r--.. 1 root root 6475 Mar 20 2020 clients.html
-rw-r--r--.. 1 root root 6218 Mar 20 2020 about.html
```

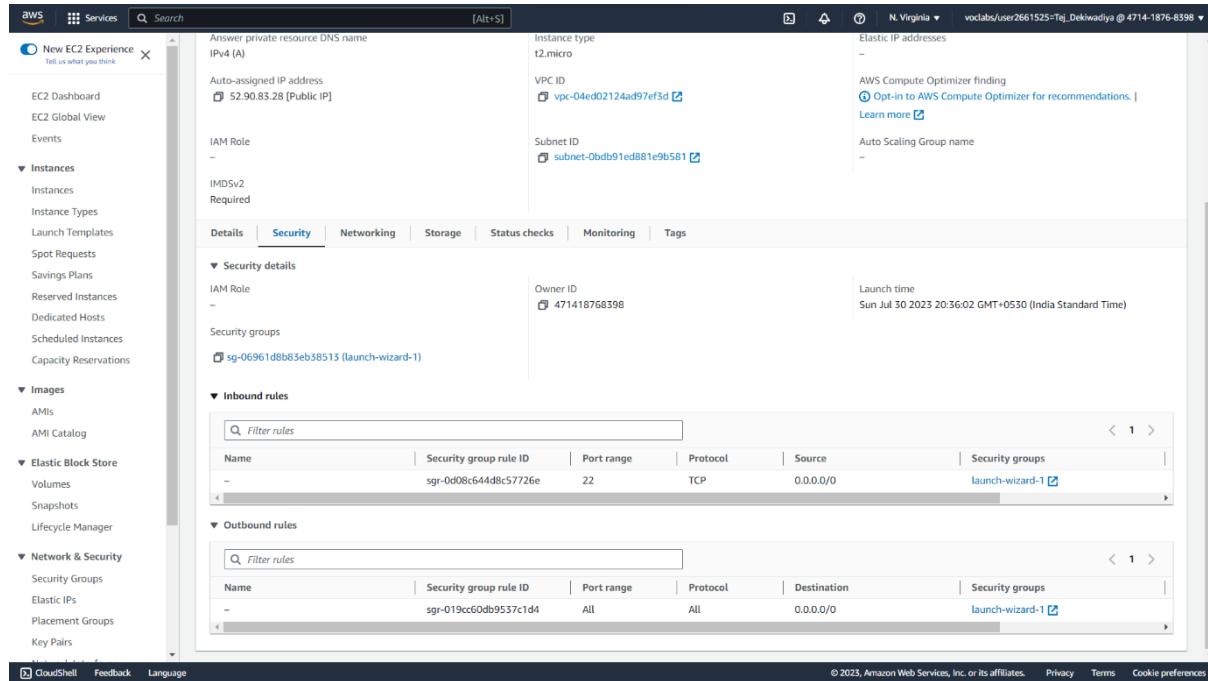
i-0c4dcfe87094b5f4b (Practical 04)

PublicIPs: 52.90.83.28 PrivateIPs: 172.31.83.47

CloudShell Feedback Language Privacy Terms Cookie preferences © 2023, Amazon Web Services, Inc. or its affiliates.

Step 10 : Now open Instance Configuration and click on Security then choose Security Group.

Snapshot :

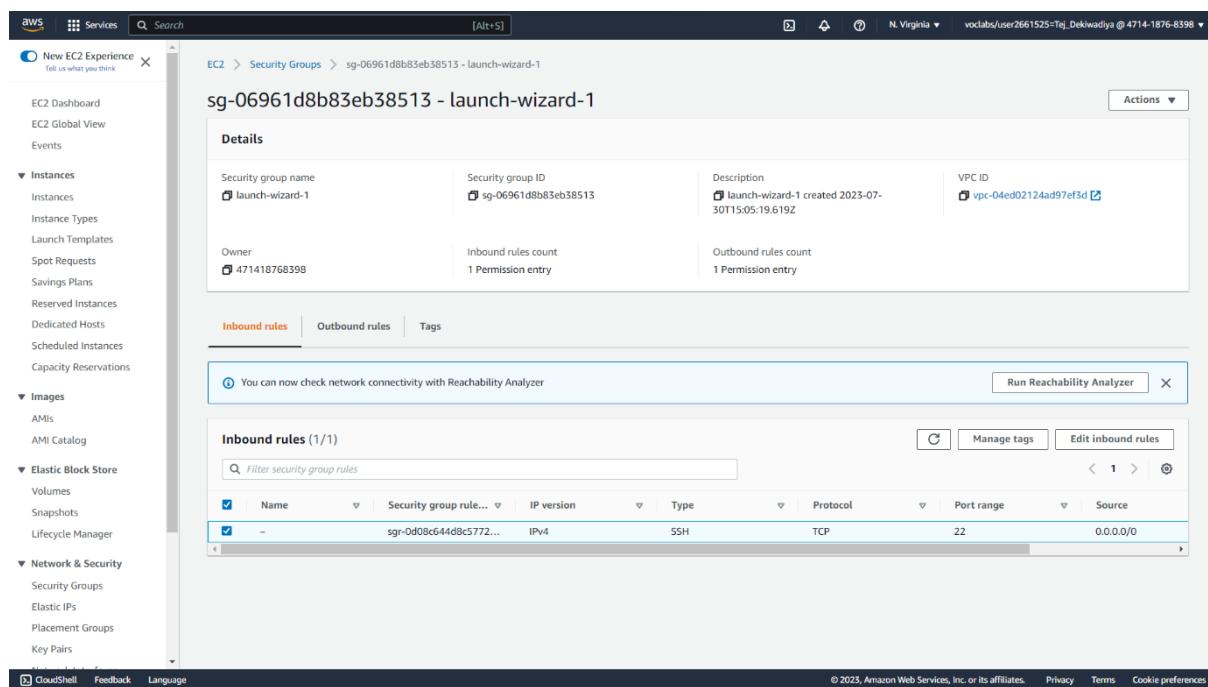


Name	Security group rule ID	Port range	Protocol	Source	Security groups
-	sgr-0d08c644d8c57726e	22	TCP	0.0.0.0/0	launch-wizard-1

Name	Security group rule ID	Port range	Protocol	Destination	Security groups
-	sgr-019cc60db9537c1d4	All	All	0.0.0.0/0	launch-wizard-1

Step 11 : Now Here in Inbound Rules only SSH allow right now.

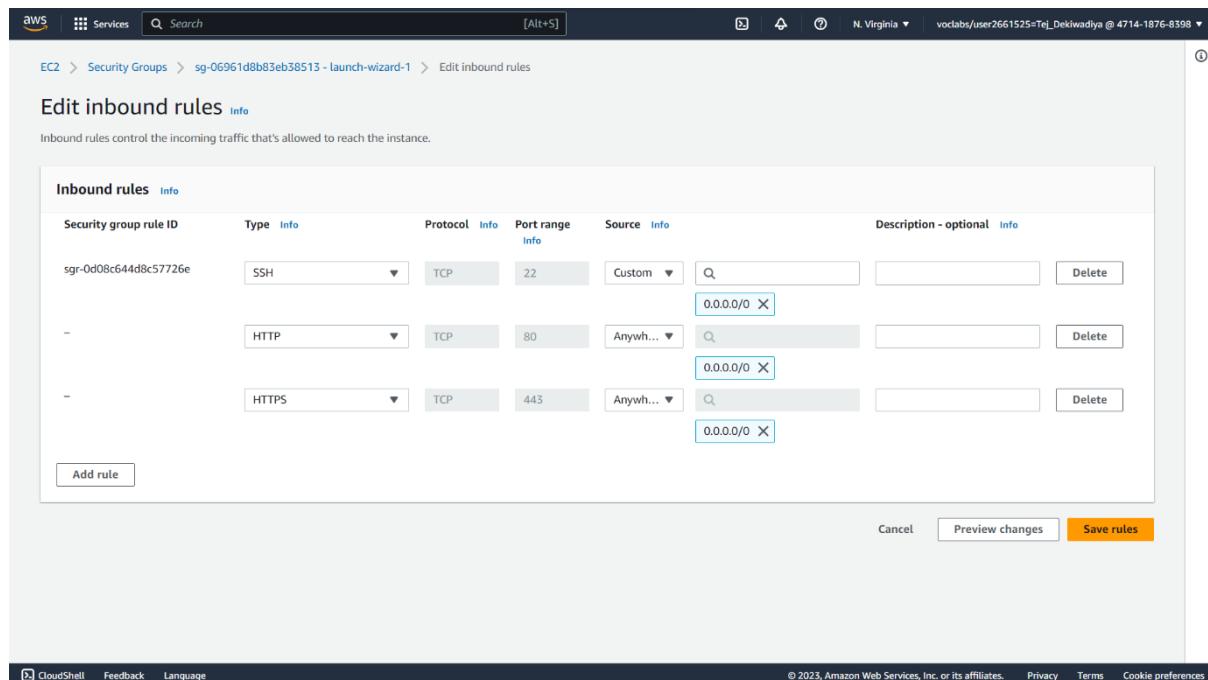
Snapshot :



Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0d08c644d8c57726e	IPv4	SSH	TCP	22	0.0.0.0/0

Step 12 : Click on Edit Inbound Rules and add HTTP & HTTPS protocol and allow everywhere so for that choose 0.0.0.0/0 then save data.

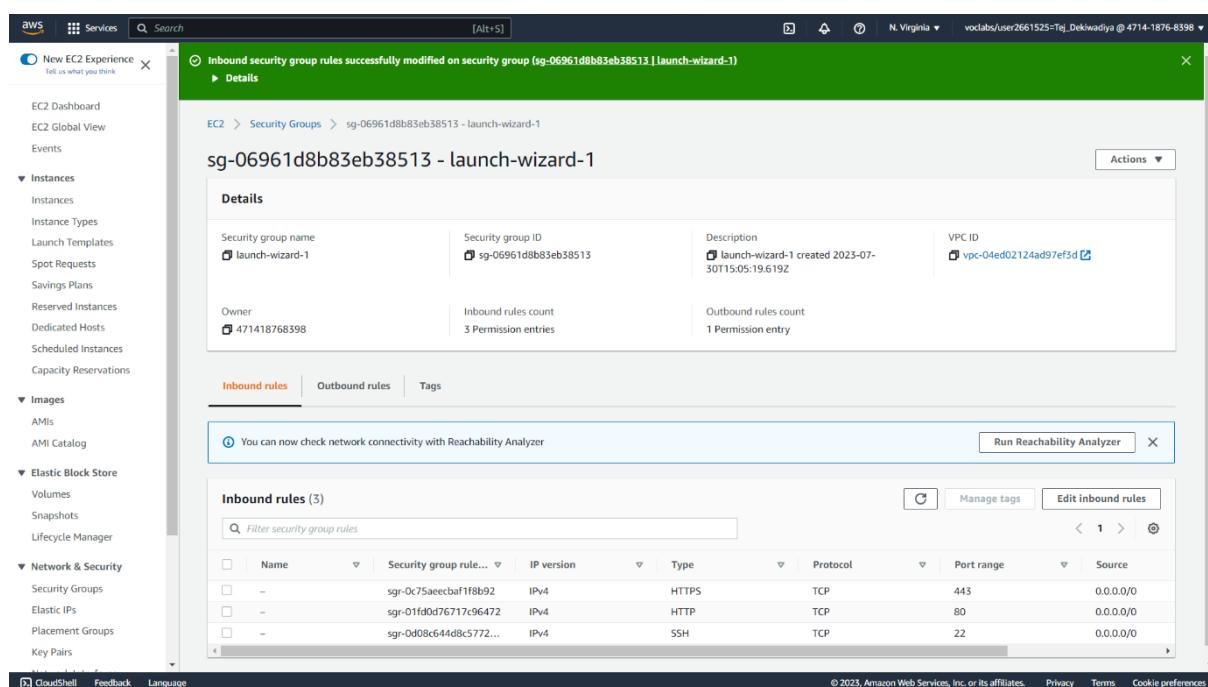
Snapshot :



Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0d08c644d8c57726e	SSH	TCP	22	Custom	0.0.0.0/0
-	HTTP	TCP	80	Anywhere	0.0.0.0/0
-	HTTPS	TCP	443	Anywhere	0.0.0.0/0

Step 13 : Now here All the Added rules are visible.

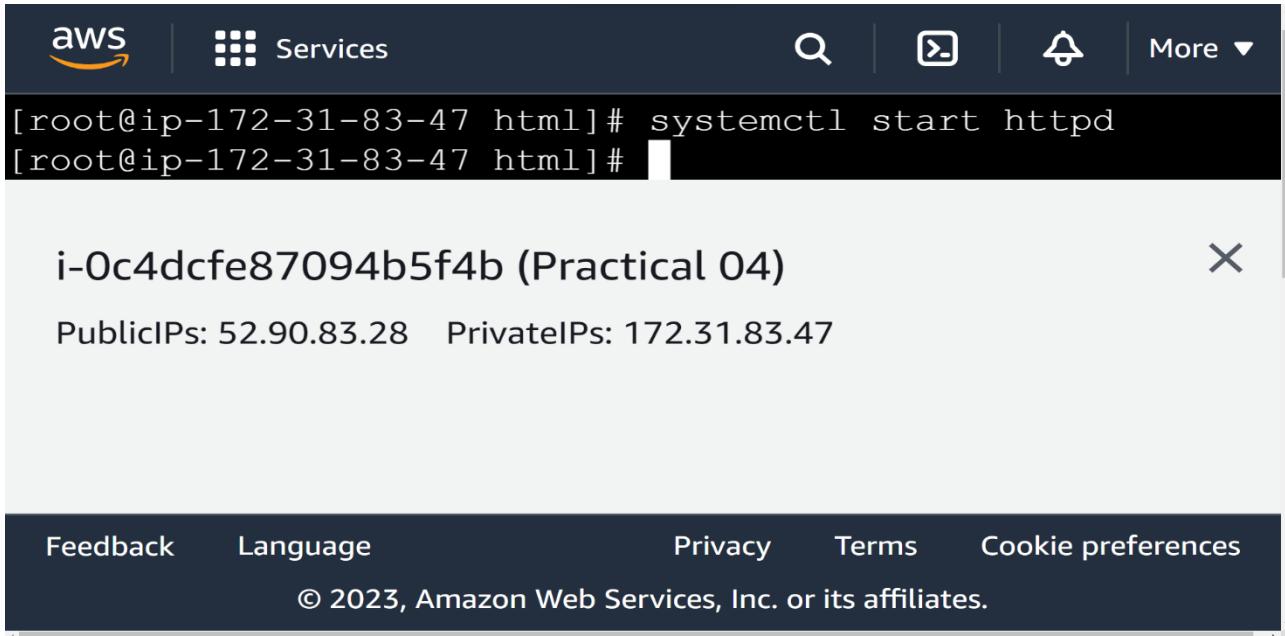
Snapshot :



Name	IP version	Type	Protocol	Port range	Source
sgr-0c75aeeecbf1fb8b92	IPv4	HTTPS	TCP	443	0.0.0.0/0
sgr-01fd0d76717c96472	IPv4	HTTP	TCP	80	0.0.0.0/0
sgr-0d08c644d8c5772...	IPv4	SSH	TCP	22	0.0.0.0/0

Step 14 : At last start httpd service → `systemctl start httpd`

Snapshot :



```
[root@ip-172-31-83-47 html]# systemctl start httpd
[root@ip-172-31-83-47 html]#
```

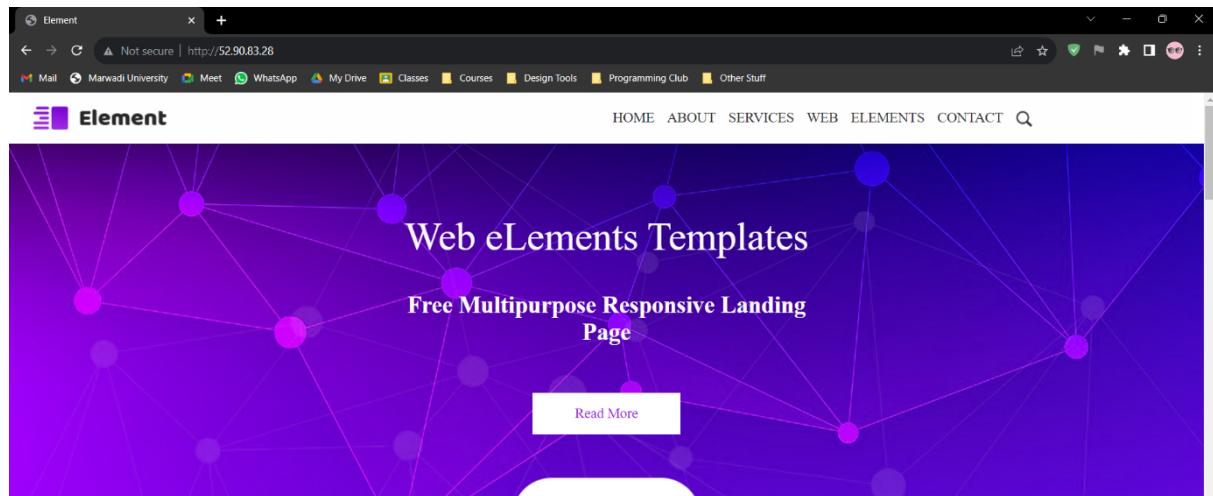
i-0c4dcfe87094b5f4b (Practical 04) X

Public IPs: 52.90.83.28 Private IPs: 172.31.83.47

Feedback Language Privacy Terms Cookie preferences
© 2023, Amazon Web Services, Inc. or its affiliates.

Step 15 : Finally copy Public IPV4 address new a new tab and run so now our web Site deploy on Cloud Server.

Snapshot :

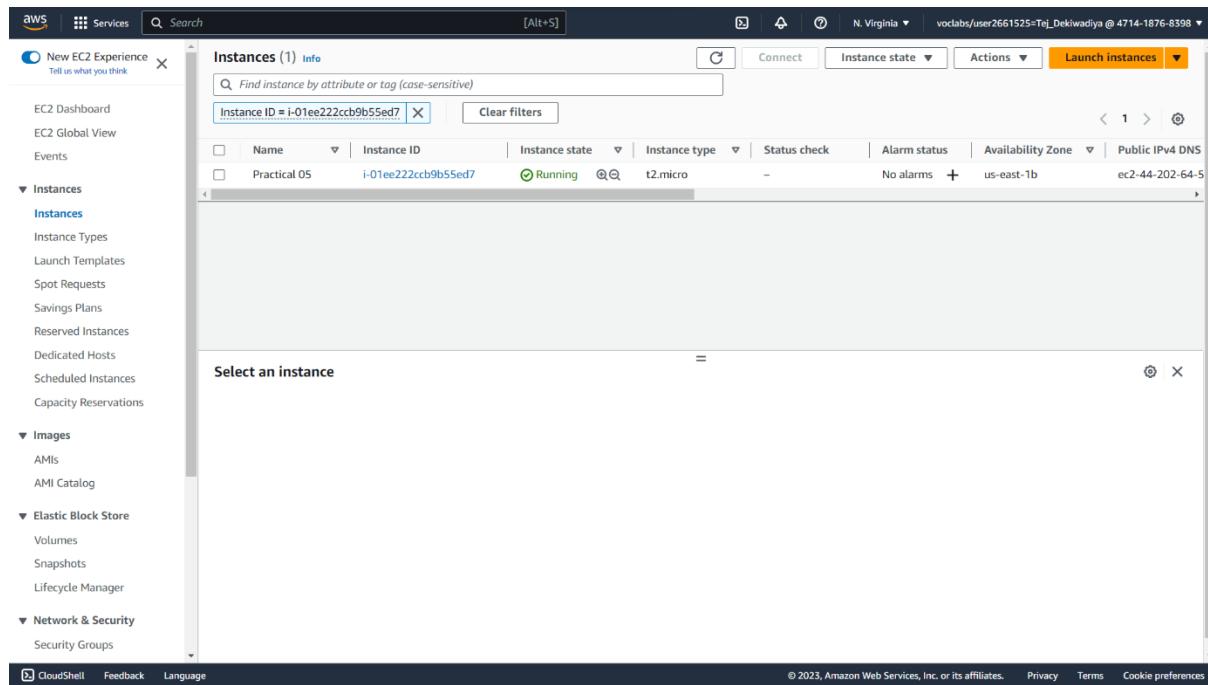


ABOUT WEB ELEMENTS

Practical 05 : Amazon EC2 Instances with Microsoft Windows

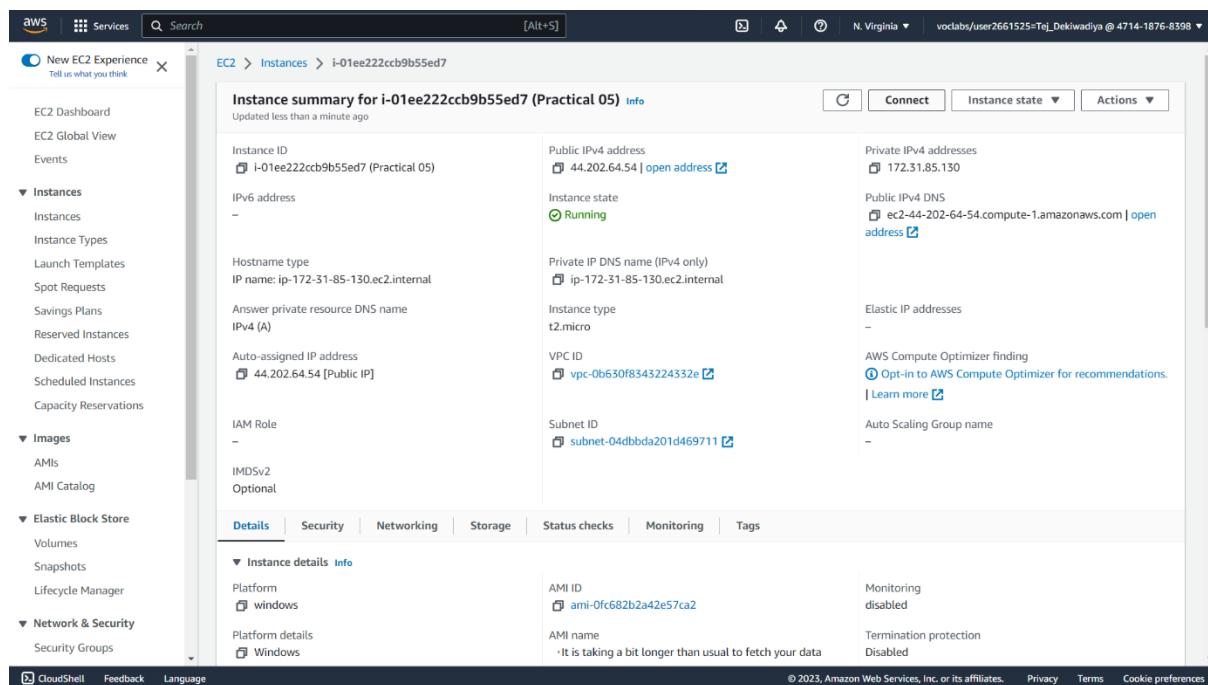
Step 01 : Create one new Instance and whenever create Instance choose all the settings by default.

Snapshot :



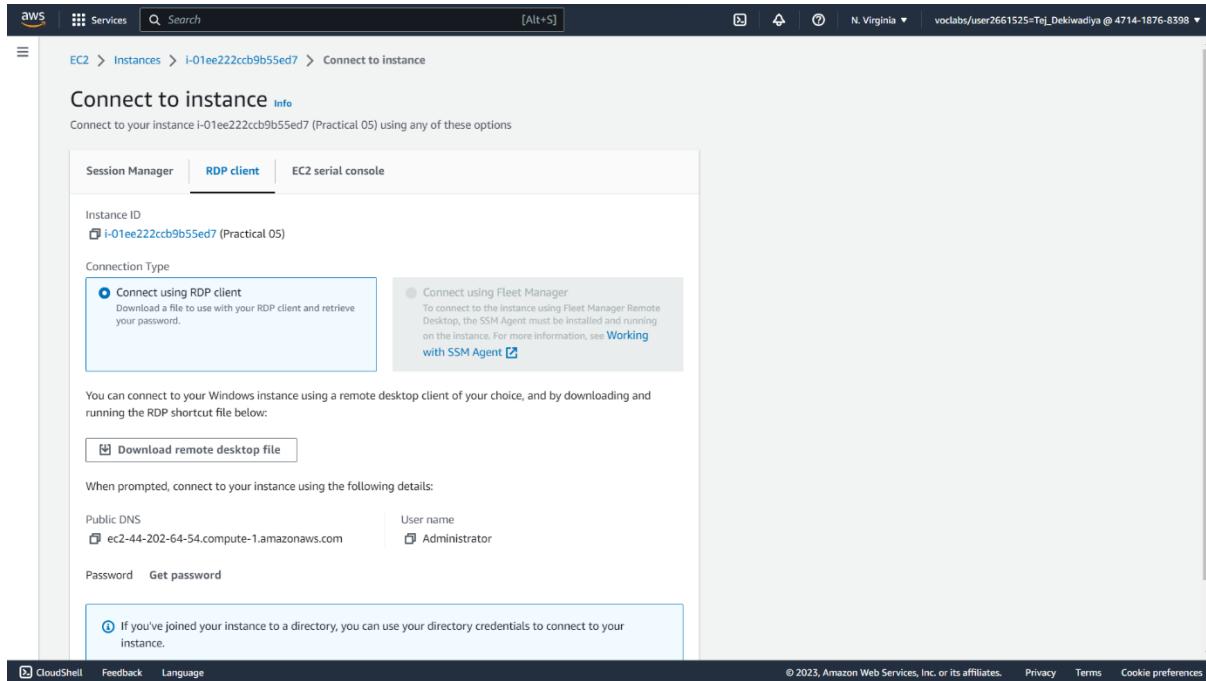
Step 02 : After the instance is created go to your instance id and click on ‘connect’

Snapshot :



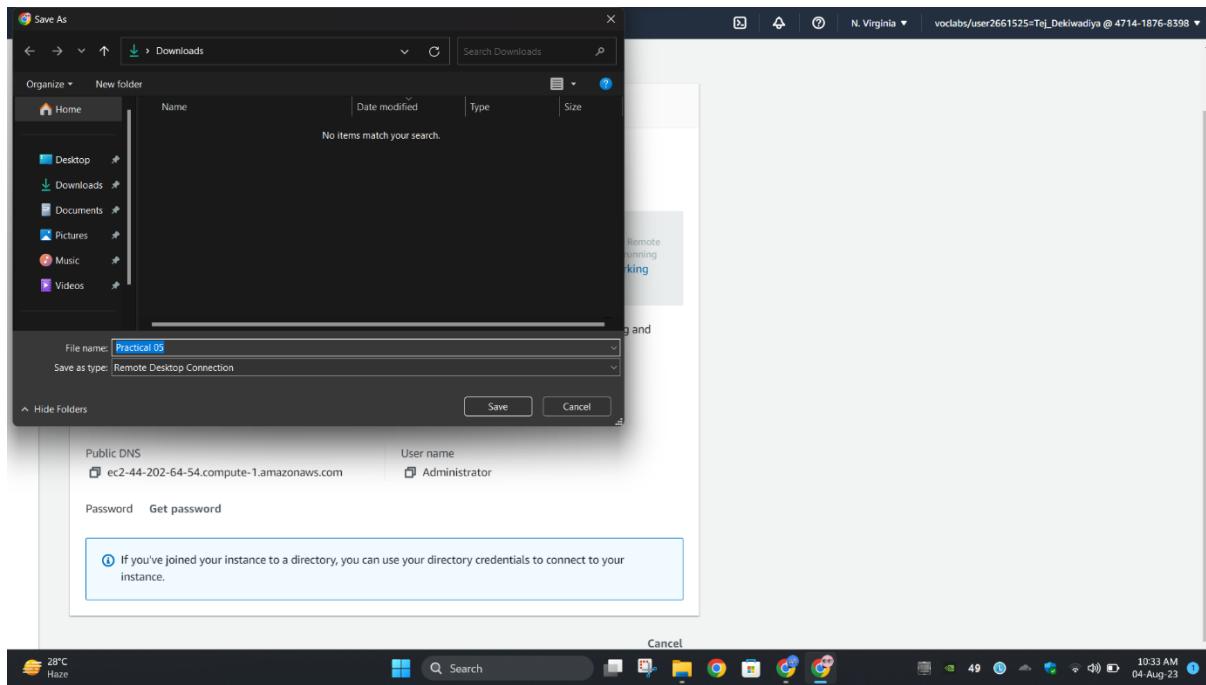
Step 03 : Now go to ‘RDP Client’.

Snapshot :



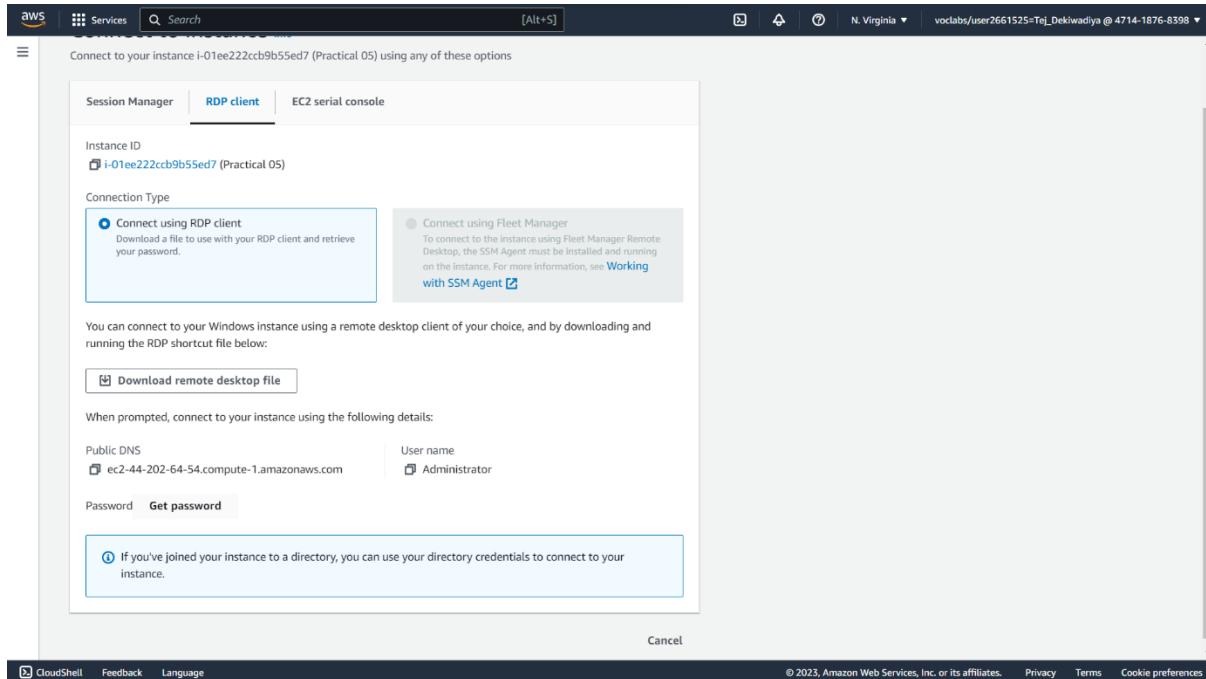
Step 04 : Now download the remote desktop file.

Snapshot :



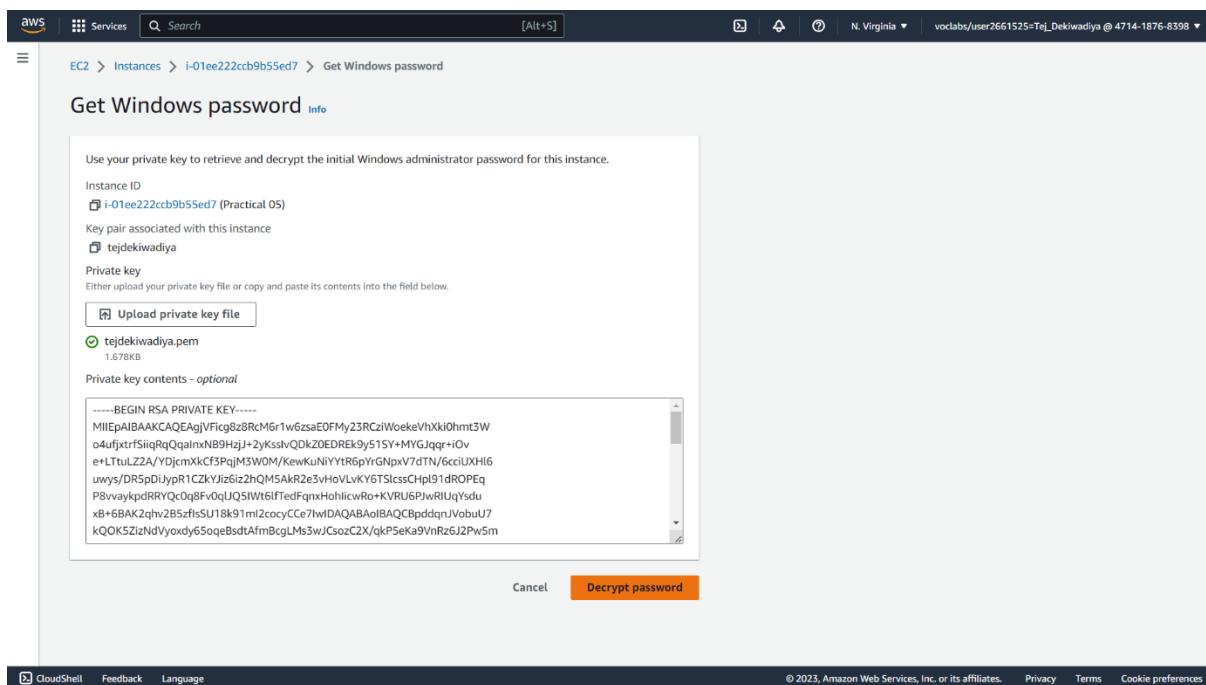
Step 05 : After downloading the remote desktop file, go to ‘Get password’.

Snapshot :



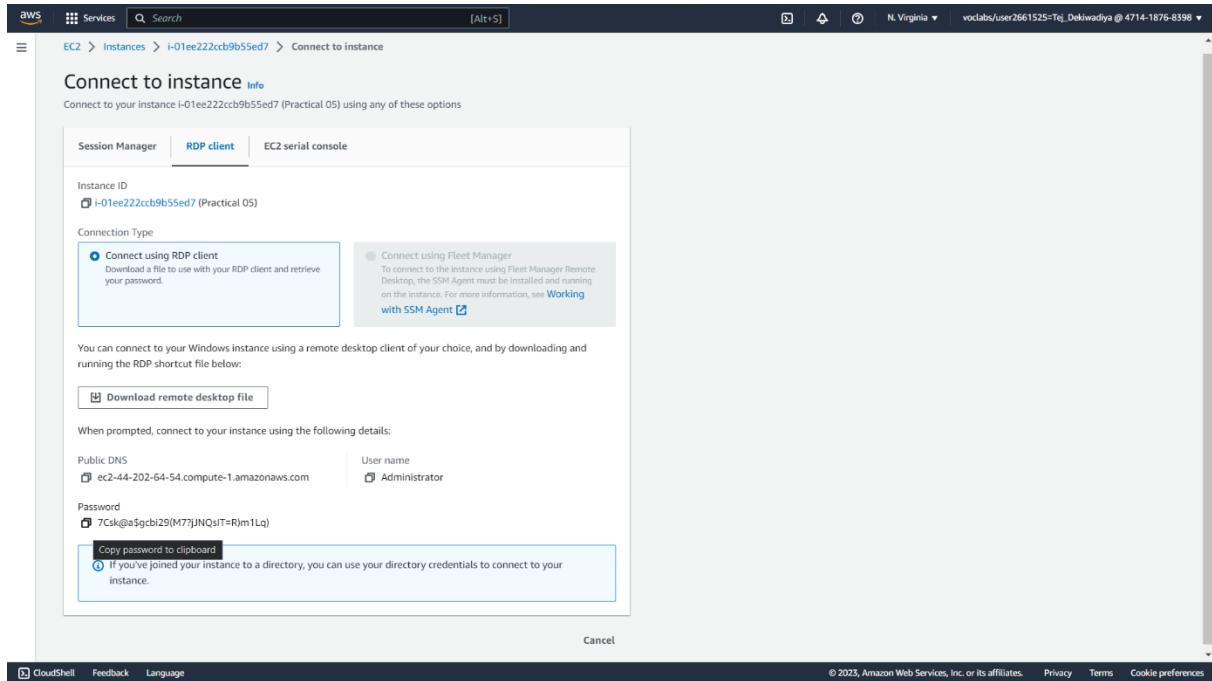
Step 06 : Upload private key file and click on Decrypt password.

Snapshot :



Step 07 : Now the password is visible to us. Copy the password.

Snapshot :

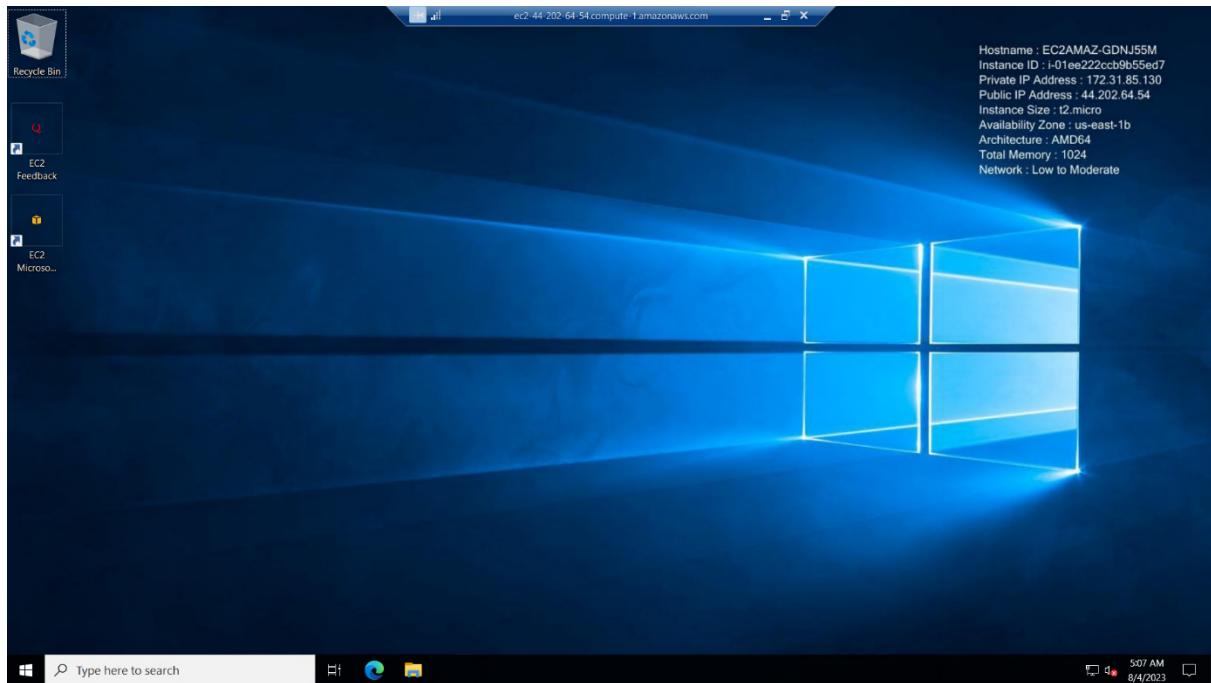


Step 08 : Open the remote desktop file and paste the copied password.

Snapshot :

Step 09 : Successfully created EC2 instance with Microsoft Windows.

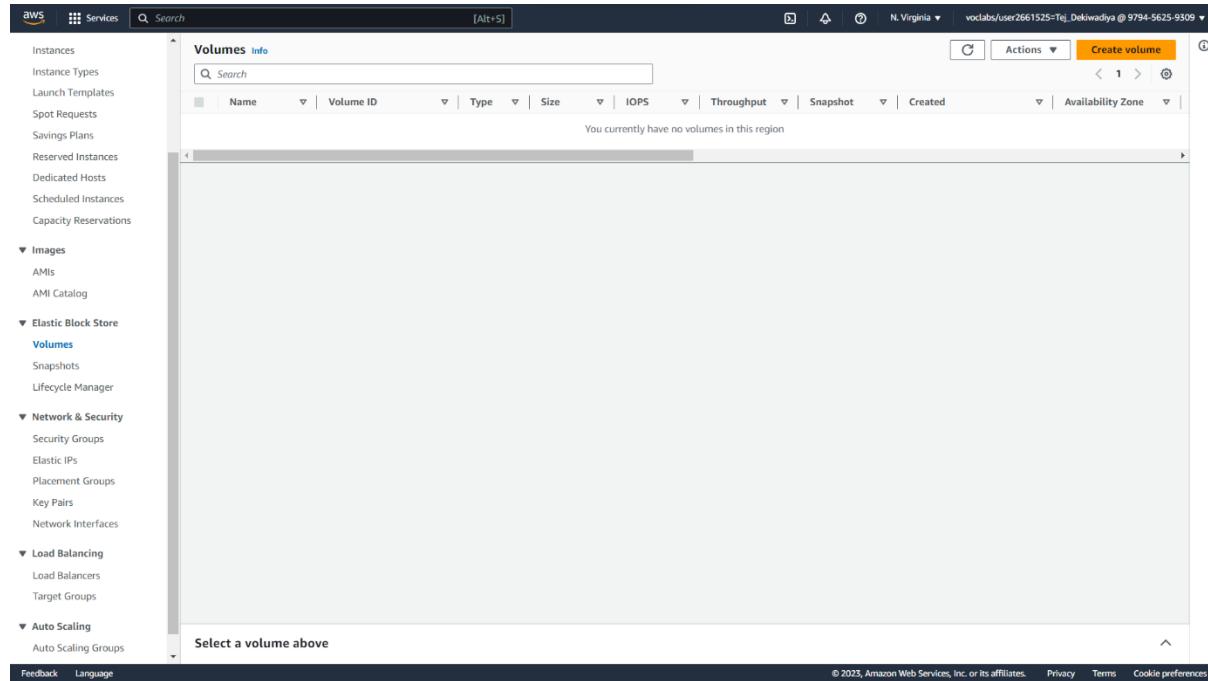
Snapshot :



Practical 06 : Amazon Elastic Block Store

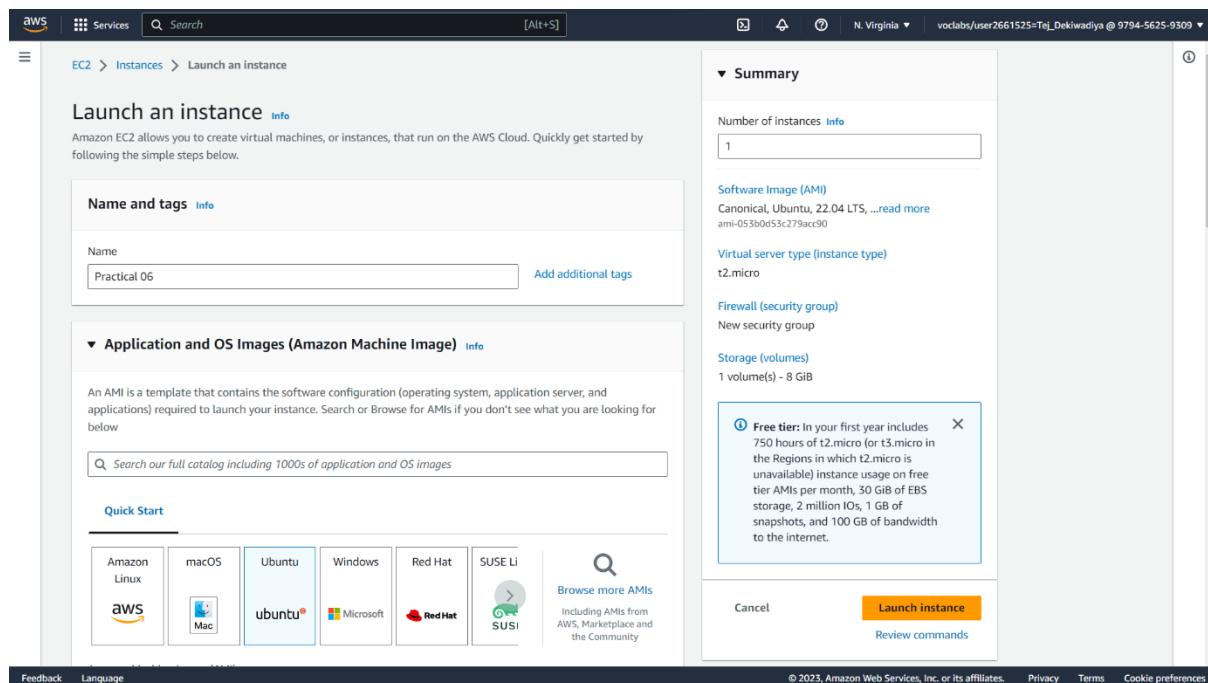
Step 01 : Go to manager panel → Elastic Block Store → Volumes. And Check how many exists.

Snapshot :



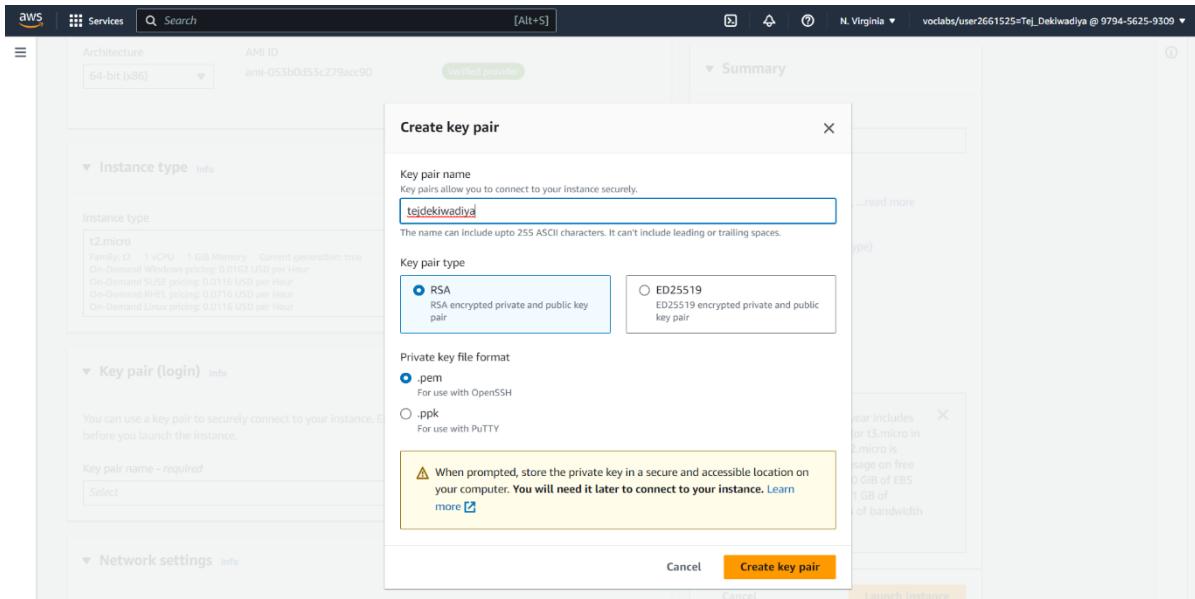
Step 02 : To Configure The Instance. First, Set The Name Of The Instance and Next Step Is To Choose The Application And OS Images Ubuntu > Ubuntu Server 22.04 LTS

Snapshot :



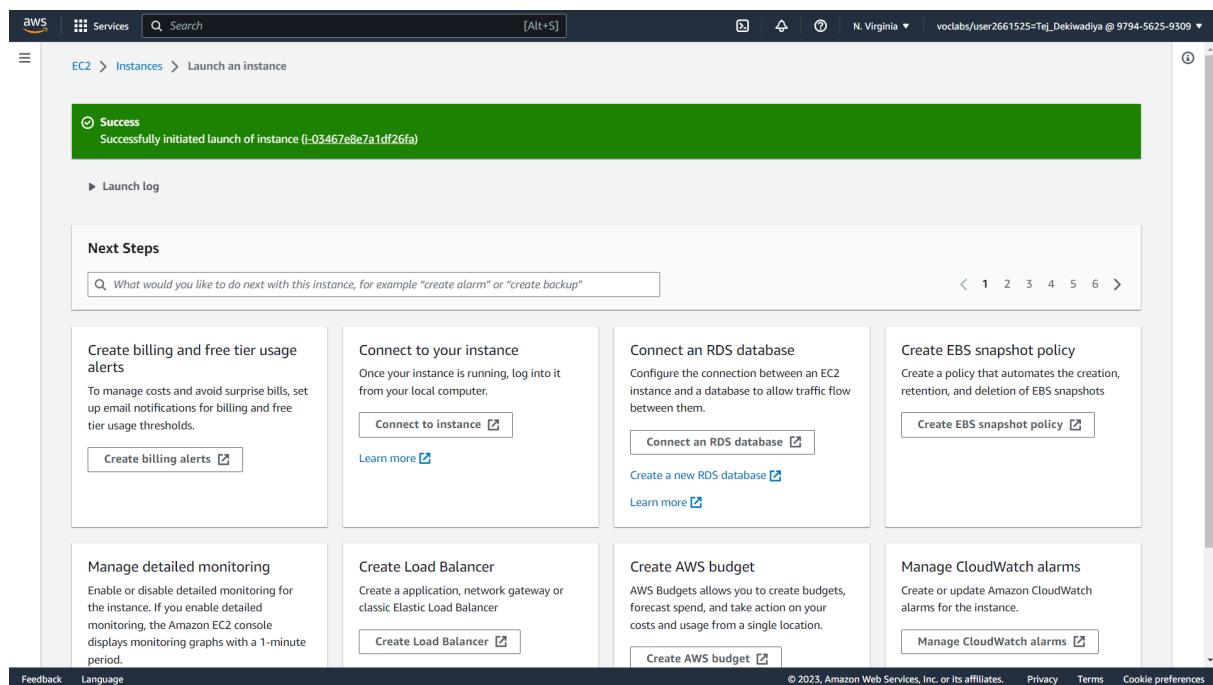
Step 03 : Next Step Is To Create New **Key Pair** In Key Pair (Login) Section and Provide A **Key Pair Name** And Then Click On "Create Key Pair" To Generate The Key Pair. (Downloading One File While Creating)

Snapshot :



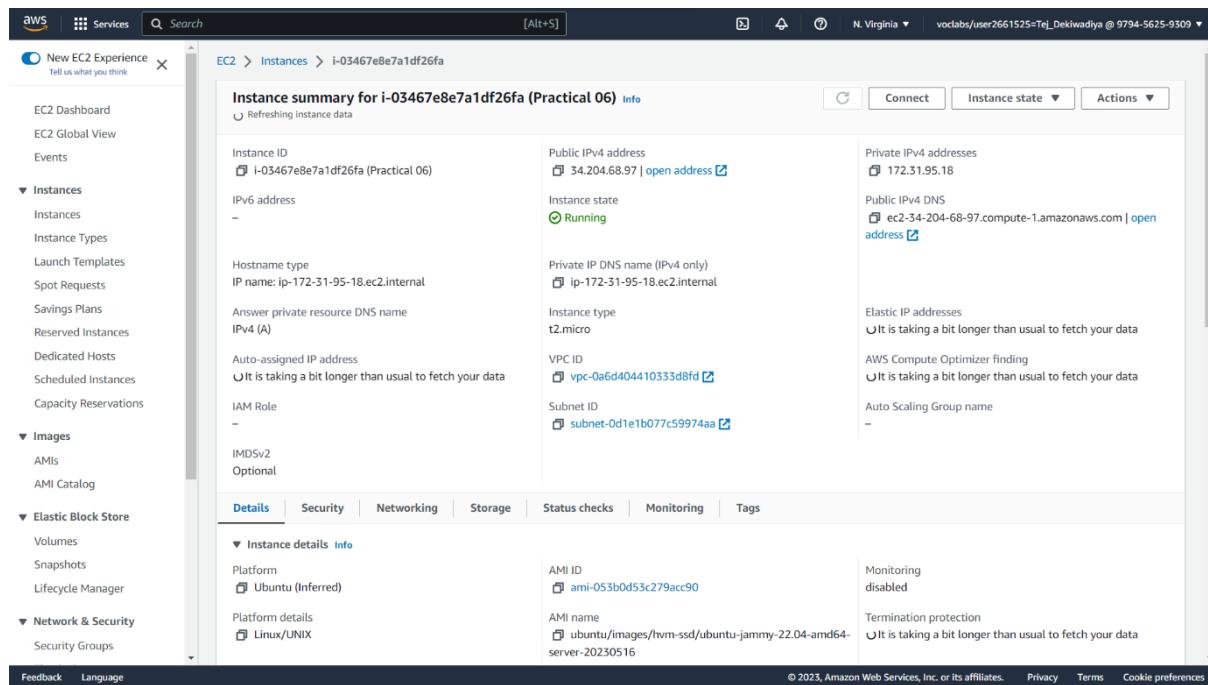
Step 04 : The Process Of Launching The Instance Will Begin, And It May Take A Few Seconds To Complete Successfully. Once It's Ready, Click On The Instance Id To Launch The Created Instance

Snapshot :



Step 05 : After Successfully Launching The Instance, Click On The "Connect" Button To Connect To The Instance.

Snapshot :

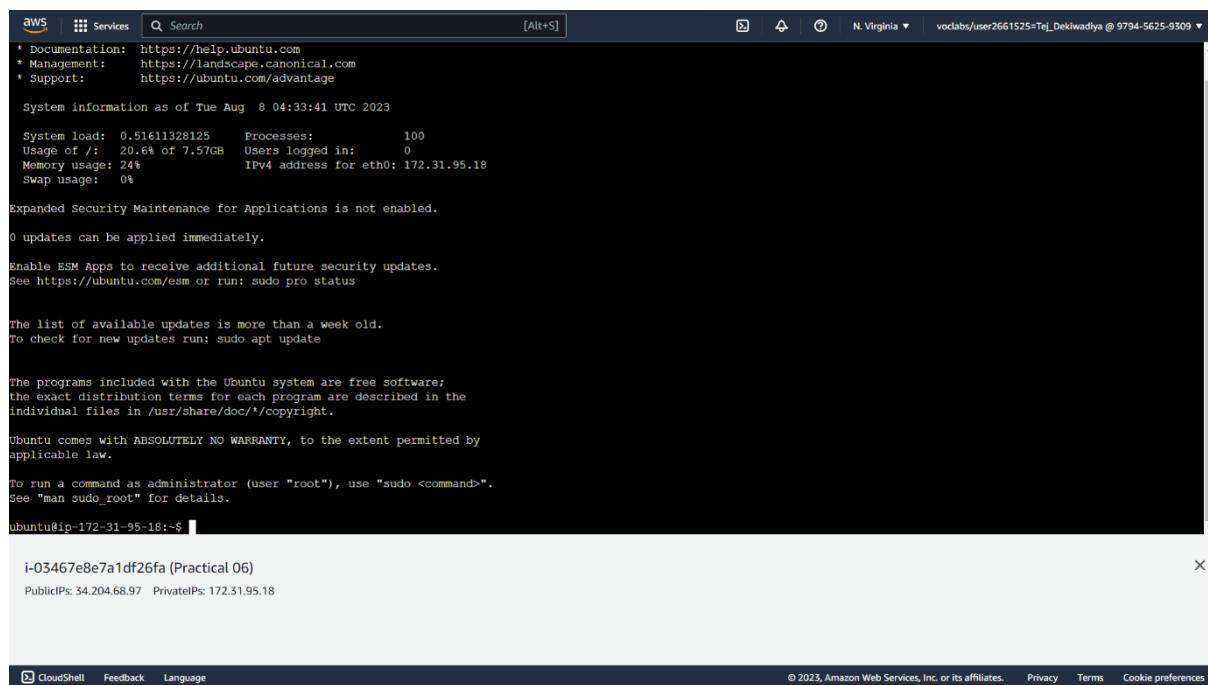


The screenshot shows the AWS EC2 Instances page. The instance summary for i-03467e8e7a1df26fa (Practical 06) is displayed. Key details include:

- Instance ID: i-03467e8e7a1df26fa (Practical 06)
- Public IPv4 address: 34.204.68.97 | [open address](#)
- Instance state: Running
- Private IP DNS name (IPv4 only): ip-172-31-95-18.ec2.internal
- Instance type: t2.micro
- VPC ID: vpc-0a6d404410333d8fd
- Subnet ID: subnet-0d1e1b077c59974aa
- AMI ID: ami-053b0d53c279acc90
- AMI name: ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230516

Step 06 : Default Terminal Opened.

Snapshot :



```

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

System information as of Tue Aug  8 04:33:41 UTC 2023

System load: 0.51611328125 Processes: 100
Usage of /: 20.6% of 7.57GB Users logged in: 0
Memory usage: 24% IPv4 address for eth0: 172.31.95.18
Swap usage: 0%

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.

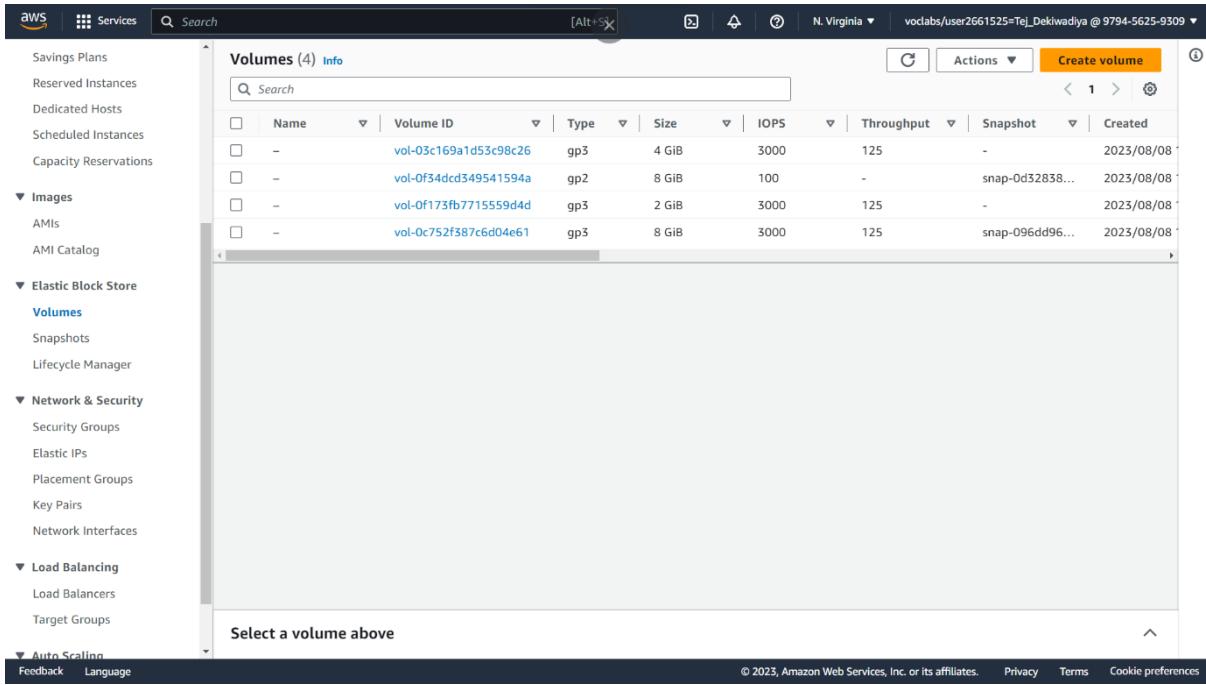
no run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-95-18:~$ 
```

i-03467e8e7a1df26fa (Practical 06)
 PublicIPs: 34.204.68.97 PrivateIPs: 172.31.95.18

Step 07 : Go to manager panel → Elastic Block Store → Volumes. And Check the volumes successfully create by us.

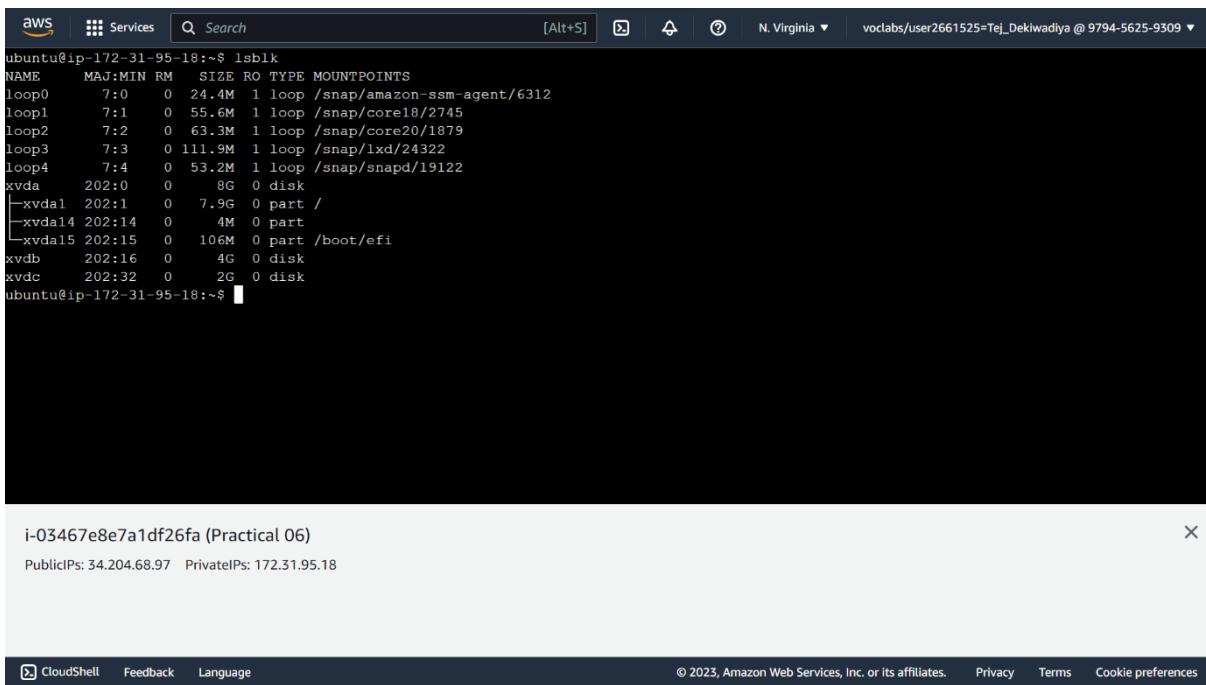
Snapshot :



Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created
-	vol-03c169a1d53c98c26	gp3	4 GiB	3000	125	-	2023/08/08
-	vol-0f34dc349541594a	gp2	8 GiB	100	-	snap-0d32838...	2023/08/08
-	vol-0f173fb7715559d4d	gp3	2 GiB	3000	125	-	2023/08/08
-	vol-0c752f587c6d04e61	gp3	8 GiB	3000	125	snap-096dd96...	2023/08/08

Step 08 : Write lsblk to check list block .

Snapshot :



```
ubuntu@ip-172-31-95-18:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0    7:0    0 24.4M  1 loop /snap/amazon-ssm-agent/6312
loop1    7:1    0 55.6M  1 loop /snap/core18/2745
loop2    7:2    0 63.3M  1 loop /snap/core20/1879
loop3    7:3    0 111.9M 1 loop /snap/lxd/24322
loop4    7:4    0 53.2M  1 loop /snap/snapd/19122
xvda   202:0    0   8G  0 disk 
└─xvda1 202:1    0   7.9G 0 part /
  ├─xvda14 202:14   0   4M  0 part 
  ├─xvda15 202:15   0 106M 0 part /boot/efi
  xvdb   202:16   0   4G  0 disk 
  xvdc   202:32   0   2G  0 disk 
ubuntu@ip-172-31-95-18:~$
```

i-03467e8e7a1df26fa (Practical 06)
 PublicIPs: 34.204.68.97 PrivateIPs: 172.31.95.18

Practical 07 : Google App Engine

1) What is the Google App Engine?

Ans: Google App Engine is a Platform-as-a-Service (PaaS) offering by Google Cloud that allows developers to build, deploy, and manage applications on Google's infrastructure without having to worry about the underlying hardware, networking, and server management. It is designed to simplify the process of developing and hosting web applications, APIs, and other services.

- Key features of Google App Engine include:
 - Scalability: App Engine automatically manages the scaling of your application based on the incoming traffic. It can handle sudden spikes in traffic without manual intervention.
 - Managed Infrastructure: Developers don't need to worry about provisioning servers, managing networking, or configuring load balancers. Google takes care of these aspects.
 - Languages and Runtimes: App Engine supports multiple programming languages such as Python, Java, Node.js, Ruby, Go, and PHP. It also offers various runtime environments for different languages.
 - Data Storage: Google Cloud services like Cloud Datastore and Cloud SQL can be integrated for data storage. These services are designed to work seamlessly with App Engine.
 - Automatic Updates: Google handles operating system updates and security patches for the underlying infrastructure, freeing developers from these tasks.
 - Security: App Engine provides security features like sandboxing, access control, and automatic SSL certificates for secure communication.
 - Task Queues: It offers task queues to manage asynchronous tasks, allowing you to offload time-consuming processes from the main application.
 - Deployment: Deploying your application is as simple as uploading your code. Google takes care of the deployment process, ensuring that it's distributed across multiple servers for redundancy.
 - App Versions: You can deploy multiple versions of your application, which can be useful for testing and rolling out updates gradually.
 - Monitoring and Logging: App Engine provides tools for monitoring the performance of your application and analyzing logs to troubleshoot issues.
 - Billing: App Engine operates on a pay-as-you-go model, where you're billed based on the resources your application consumes.
 - App Engine is suitable for various types of applications, from small prototypes and startups to large-scale, production-grade applications. It abstracts much of the complexity involved in managing infrastructure, allowing developers to focus on writing code and building features.

2) What is the use of the Google App Engine?

Ans: Google App Engine serves as a Platform-as-a-Service (PaaS) offering that provides a range of use cases for developers and businesses. Here are some common uses of Google App Engine:

- **Web Application Hosting:** Developers can use App Engine to host and deploy web applications, whether they are simple websites, content management systems, or more complex web services. App Engine's automatic scaling ensures that applications can handle varying levels of traffic without manual intervention.
- **API Services:** App Engine can be used to create and host APIs (Application Programming Interfaces) that allow different applications to communicate with each other. This is particularly useful for building backend services for mobile apps, web apps, and other client applications.
- **Microservices:** Developers can break down their applications into smaller, independently deployable services known as microservices. App Engine's managed infrastructure and automatic scaling make it easier to develop, deploy, and manage these microservices.
- **Background Processing:** App Engine offers task queues that enable developers to offload background processing tasks from the main application. This is useful for handling tasks that are time-consuming and don't need to be executed immediately.
- **Data Processing:** Developers can utilize App Engine to process and analyze large volumes of data using tools like MapReduce, BigQuery, or other data processing frameworks. This is particularly helpful for applications that require data-intensive operations.
- **Prototyping and Testing:** App Engine allows developers to quickly prototype and test their ideas without the need to set up and manage complex infrastructure. This can help accelerate the development cycle.
- **Scalability:** For applications that experience fluctuating traffic patterns, App Engine's automatic scaling ensures that resources are allocated as needed to accommodate varying levels of user activity.
- **Content Management Systems (CMS):** App Engine can host content management systems that allow users to create, manage, and publish digital content without the need for complex server management.
- **Real-time Applications:** Applications requiring real-time communication, such as chat applications or collaborative tools, can benefit from App Engine's scalability and support for WebSockets.
- **IoT (Internet of Things) Backend:** App Engine can serve as a backend for IoT devices, collecting and processing data from various connected devices and sensors.
- **Game Backend:** Game developers can use App Engine to build and deploy the backend services required for multiplayer games, leaderboards, and user authentication.
- **E-commerce:** App Engine can be used to host e-commerce platforms, handling the online storefront, product catalog, and checkout processes. In essence, Google App Engine simplifies the process of developing, deploying, and managing web applications and services, allowing developers to focus on writing code and creating innovative features without getting bogged down in the complexities of server management.

3) What types of services are provided by Google App Engine?

Ans: Google App Engine provides various services and features to help developers build, deploy, and manage applications. These services can be broadly categorized into the following types:

- **Compute Services:**
 - **Standard Environment:**
 - This is a runtime environment that supports specific programming languages, such as Python, Java, Node.js, Ruby, and Go. It offers automatic scaling, load balancing, and managed infrastructure. However, it has some limitations in terms of customization and third-party libraries.
 - Flexible Environment (App Engine Flex): This environment allows more customization and supports a wider range of programming languages and runtimes. It's based on containers and provides more control over the underlying infrastructure.
- Data Storage and Management:
 - Cloud Datastore:
 - A NoSQL database service that provides scalable storage for structured data. It is designed to handle large amounts of data with high availability.
 - Cloud SQL: A managed relational database service that supports MySQL, PostgreSQL, and SQL Server. It offers automated backups, scaling, and security features.
 - Cloud Storage: A scalable object storage service for storing and retrieving large amounts of unstructured data, such as files and media.
- Networking and Integration:
 - Network and Load Balancing:
 - App Engine manages network traffic and load balancing to distribute incoming requests across multiple instances of your application.
 - Cloud Pub/Sub: A messaging service that allows applications to send and receive messages between independent components.
 - Cloud Endpoints: A service for building, deploying, and managing APIs, providing tools for generating client libraries and managing API access.
- Security and Identity:
 - Identity-Aware Proxy: A service that provides identity-based access control and security for applications running on App Engine.
 - App Engine Identity and Access Management (IAM): Allows you to control access to resources within your App Engine applications using IAM roles and permissions. Monitoring and Logging:

- Stackdriver Monitoring: Provides tools to monitor the performance of your application, including resource utilization, response times, and error rates.
- Stackdriver Logging: Offers centralized logging and analysis of application logs, helping you troubleshoot and identify issues. Task Queues and Background Processing:
- Task Queues: App Engine provides task queues for managing asynchronous tasks, allowing you to offload timeconsuming processes from your main application.
- Development and Deployment:
 - App Deployment: Deploy your application code to App Engine using the gcloud command-line tool or through continuous integration pipelines.
 - App Versions: Deploy multiple versions of your application for different purposes, such as development, testing, and production.
- Runtime Services:
 - App Engine Services: In addition to the main application, you can create separate services within the same project to serve different purposes. Each service can have its own configuration and scaling settings. Task Scheduling:
- Cron Jobs: Schedule and automate tasks to run at specified intervals using App Engine's cron service. These services collectively provide a comprehensive platform for building and running web applications, APIs, microservices, and other types of software solutions on Google Cloud's infrastructure, with a focus on minimizing the operational overhead for developers.

4) Comparison between Google App Engine and AWS

Ans: Google App Engine and Amazon Web Services (AWS) are both cloud computing platforms that offer a range of services for building, deploying, and managing applications. However, they have some differences in terms of features, architecture, and focus. Here's a comparison between Google App Engine and AWS:

- Platform Focus:
 - Google App Engine: Google App Engine focuses on providing a Platform-as-a-Service (PaaS) environment, abstracting much of the underlying infrastructure management from developers. It emphasizes ease of use, automatic scaling, and simplified deployment.
 - AWS: AWS offers a broader set of services, including Infrastructure-as-a-Service (IaaS), PaaS, and more specialized services. While AWS provides a wide range of infrastructure options, it also offers higher flexibility but often requires more configuration and management.
- Service Models:
 - Google App Engine: Primarily offers PaaS services with managed runtimes, automatic scaling, and abstracted infrastructure components.
 - AWS: Offers a mix of IaaS, PaaS, and specialized services. AWS Elastic Beanstalk is the closest PaaS equivalent to Google App Engine.
- Compute Services:
 - Google App Engine: Provides both Standard and Flexible environments, offering automatic scaling and abstraction of infrastructure. It's particularly suitable for web applications and APIs.
 - AWS: Offers Amazon EC2 for virtual machines and AWS Elastic Beanstalk for PaaS-like deployment. AWS Lambda provides serverless compute for event-driven applications.
- Data Storage:
 - Google App Engine: Integrates with Google Cloud Datastore (NoSQL), Cloud SQL (relational), and Cloud Storage (object storage).
 - AWS: Offers Amazon RDS (relational), Amazon DynamoDB (NoSQL), Amazon S3 (object storage), and more.
- Networking:
 - Google App Engine: Provides networking and load balancing for applications.
 - AWS: Offers Amazon VPC for custom networking setups, Elastic Load Balancing, and other networking services.

- Serverless:
 - Google App Engine: Offers automatic scaling, but serverless computing is more emphasized in Google Cloud Functions.
 - AWS: Emphasizes serverless computing with AWS Lambda, allowing you to run code without provisioning or managing servers.
 - Containers: Google App Engine: Offers containerization through the Flexible environment, allowing more control over the runtime environment.
 - AWS: Provides Amazon ECS (Elastic Container Service) and Amazon EKS (Elastic Kubernetes Service) for container orchestration.
- Managed Services:
 - Google App Engine: Provides many managed services, such as Datastore, Pub/Sub, and Stackdriver for monitoring.
 - AWS: Offers a wide range of managed services like AWS RDS, DynamoDB, S3, and more.
- Market Share and Ecosystem:
 - Google App Engine: Has a smaller market share compared to AWS, but is favored by developers who prefer Google's technologies and services.
 - AWS: Dominates the cloud computing market with a wide array of services and a large customer base. In summary, Google App Engine is a PaaS offering focused on simplicity, while AWS offers a broader range of services including both IaaS and PaaS. The choice between them depends on your application's requirements, your familiarity with the platform, and your preference for specific services and ecosystems.

5) What are the Advantages of using the Google App Engine?

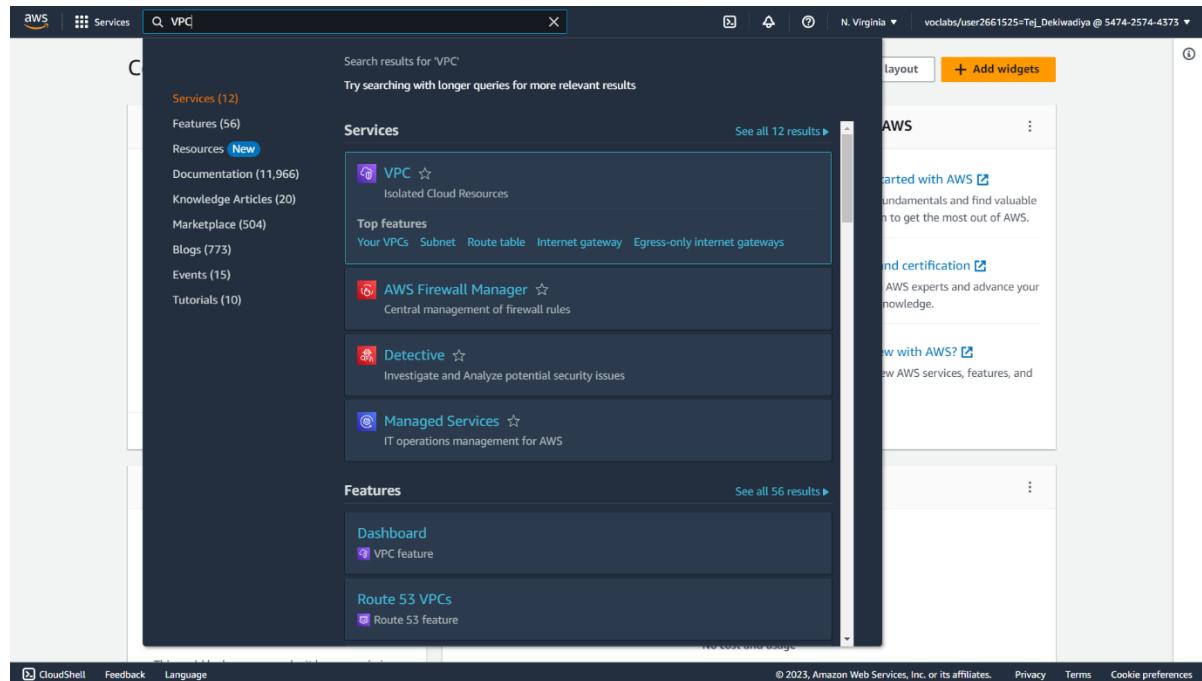
Ans: Using Google App Engine offers several advantages for developers and businesses looking to build and deploy applications. Some of the key advantages include:

- Simplified Development: App Engine abstracts much of the underlying infrastructure, allowing developers to focus on writing code rather than managing servers, networking, and other infrastructure components. This speeds up development and reduces operational overhead.
- Automatic Scaling: App Engine automatically scales your application based on incoming traffic. This means your application can handle fluctuations in user activity without manual intervention. It's particularly beneficial for applications with unpredictable or variable workloads.
- Managed Infrastructure: Google handles hardware provisioning, load balancing, server maintenance, and security patching. This reduces the burden on developers and ensures a high level of availability and reliability.
- Multiple Runtimes: App Engine supports multiple programming languages and runtimes, including Python, Java, Node.js, Ruby, Go, and more. This allows developers to work in their preferred language without constraints.
- High Availability: Google's infrastructure is designed for high availability and fault tolerance. App Engine applications are distributed across multiple data centers, reducing the risk of downtime due to hardware failures.
- Data Storage Options: App Engine integrates with Google Cloud Datastore for NoSQL data storage and Cloud SQL for relational databases. This simplifies data management and allows you to choose the best fit for your application's data needs.
- Security Features: App Engine provides security features such as sandboxing, automatic SSL certificates, and Identity-Aware Proxy for controlling access to resources. Google's security expertise helps protect your application and data.
- Integrated Services: App Engine seamlessly integrates with other Google Cloud services, including Pub/Sub for messaging, Cloud Storage for file storage, and Stackdriver for monitoring and logging.
- Task Queues: App Engine offers task queues that enable you to offload time-consuming tasks to be executed asynchronously. This ensures responsiveness and optimal resource utilization.
- Deployment and Versioning: You can deploy multiple versions of your application and easily switch between them. This facilitates testing, gradual rollouts, and quick fixes.
- Cost Efficiency: With pay-as-you-go pricing, you only pay for the resources you use. App Engine's automatic scaling and efficient resource allocation contribute to cost savings.
- Elasticity: App Engine's ability to scale up or down based on demand means you don't need to over-provision resources, which can lead to cost savings during periods of lower usage.
- Vendor Lock-In Mitigation: While vendor lock-in is a consideration for any cloud provider, using App Engine within Google Cloud provides a level of familiarity and consistency if you're already using other Google Cloud services.
- Global Reach: Google Cloud has data centers located around the world, allowing you to deploy your application closer to your target audience for reduced latency and improved user experience.
- In summary, Google App Engine offers simplicity, automatic scaling, managed infrastructure, and a range of services that make it a powerful platform for developing and deploying web applications, APIs, and services. It's particularly suitable for developers who prioritize speed, ease of use, and efficient resource management.

Practical 08 : Creating and configuring database instance

Step 01 : In the AWS Management Console, on the Services menu, choose VPC

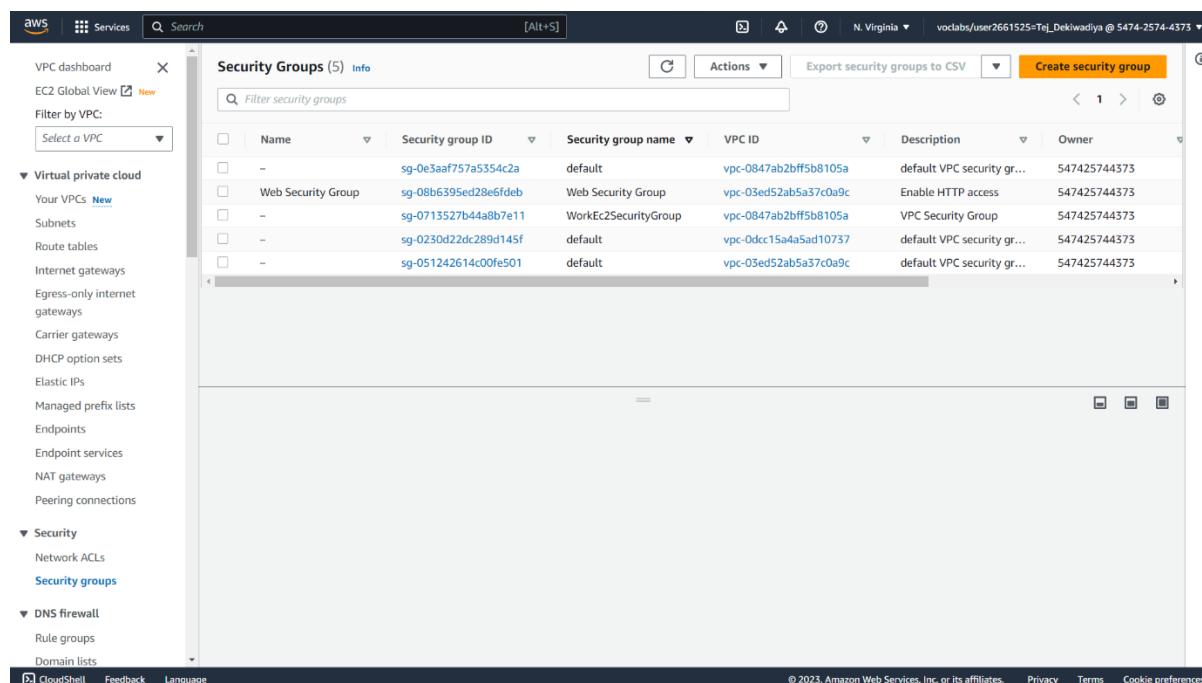
Snapshot :



The screenshot shows the AWS Management Console search results for 'VPC'. The search bar at the top contains 'VPC'. Below it, a search result for 'VPC' is shown under the 'Services' category. The 'VPC' service is described as 'Isolated Cloud Resources'. It has several 'Top features': Your VPCs, Subnet, Route table, Internet gateway, and Egress-only internet gateways. Other services listed include AWS Firewall Manager, Detective, and Managed Services. Below the services, there are sections for 'Features' like Dashboard and Route 53 VPCs.

Step 02 : In the left navigation pane, choose Security Groups.

Snapshot :



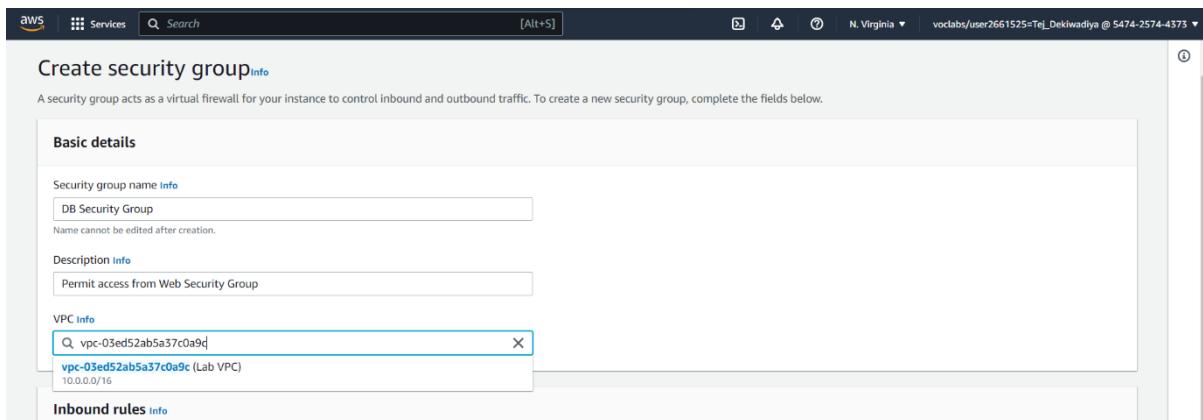
The screenshot shows the AWS Management Console Security Groups page. The left navigation pane is expanded, showing categories like VPC dashboard, Virtual private cloud, Security, and DNS firewall. Under the Security group, 'Security groups' is selected. The main area displays a table of security groups with columns: Name, Security group ID, Security group name, VPC ID, Description, and Owner. Five security groups are listed:

Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-0e3aaef757a5354c2a	default	vpc-0847ab2bf5fb8105a	default VPC security gr...	547425744373
Web Security Group	sg-08b6395ed28e6fdeb	Web Security Group	vpc-03ed52ab5a37c0a9c	Enable HTTP access	547425744373
-	sg-0713527b44ab7e11	WorkEc2SecurityGroup	vpc-0847ab2bf5fb8105a	VPC Security Group	547425744373
-	sg-0230d22dc289d145f	default	vpc-0dc215a4a5ad10737	default VPC security gr...	547425744373
-	sg-051242614c00fe501	default	vpc-03ed52ab5a37c0a9c	default VPC security gr...	547425744373

Step 03 : Choose Create security group and then configure:

- **Security group name:** DB Security Group
- **Description:** Permit access from Web Security Group
- **VPC:** Lab VPC

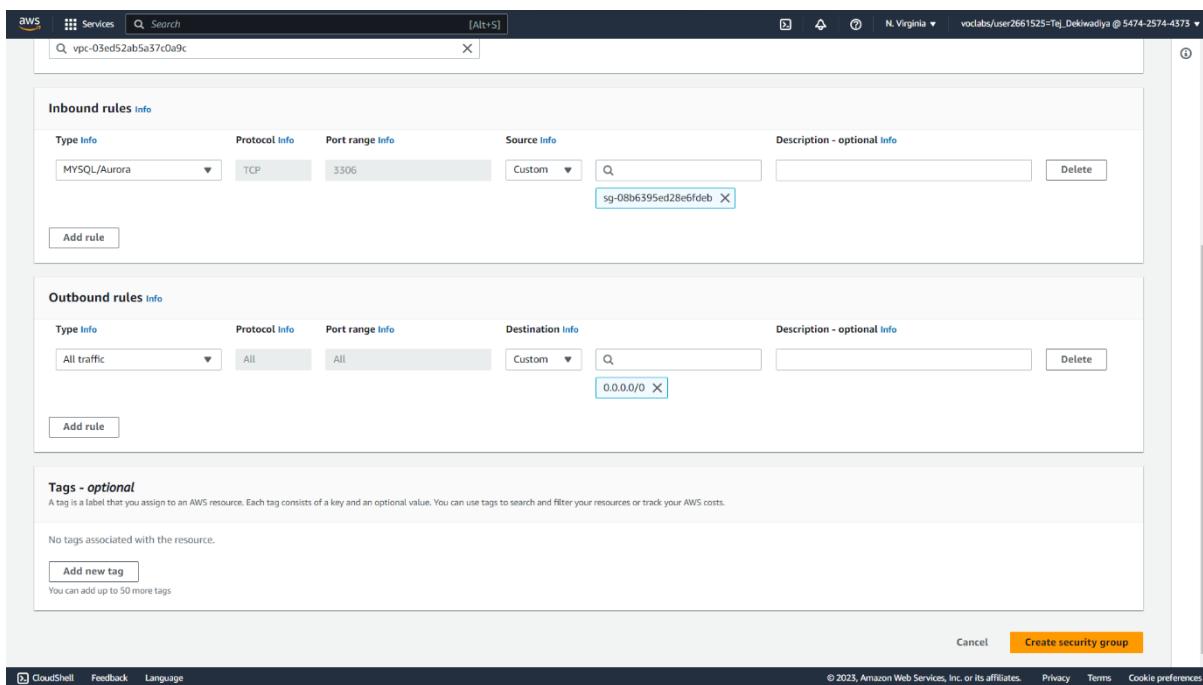
Snapshot :

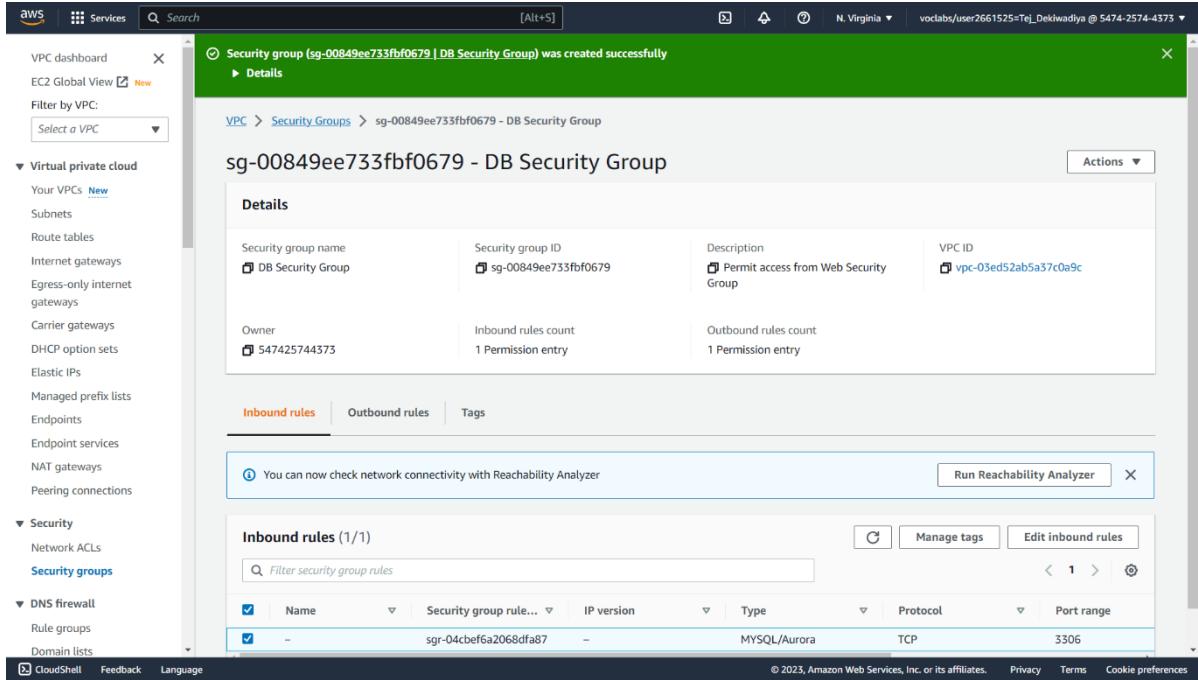


Step 04 : In the Inbound rules pane, choose Add rule.

- **Type:** MySQL/Aurora (3306)
- **CIDR, IP, Security Group or Prefix List:** Type **sg** and then select Web Security Group.
- Choose Create security group

Snapshot :

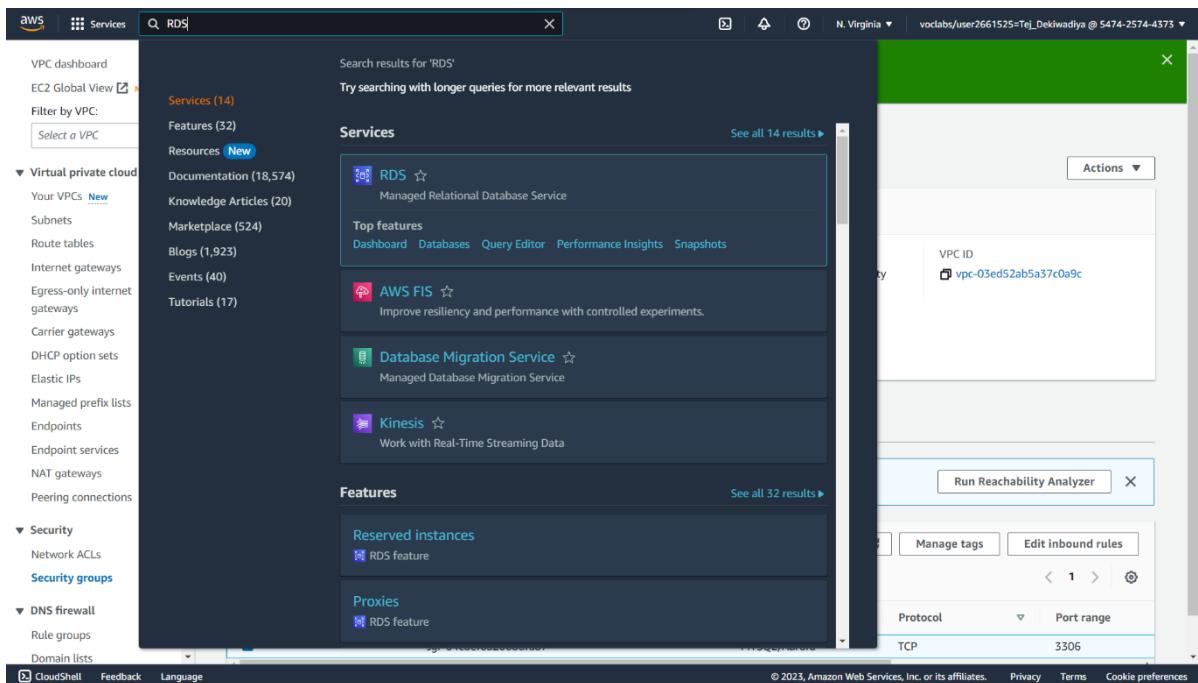




The screenshot shows the AWS VPC dashboard. A modal window at the top right indicates that a security group was created successfully. The main view displays the details of the newly created security group, 'sg-00849ee733fbf0679 - DB Security Group'. The 'Details' tab is selected, showing the security group name ('DB Security Group'), security group ID ('sg-00849ee733fbf0679'), description ('Permit access from Web Security Group'), VPC ID ('vpc-03ed52ab5a37c0a9c'), owner ('547425744373'), inbound rules count ('1 Permission entry'), and outbound rules count ('1 Permission entry'). Below the details, tabs for 'Inbound rules' (selected), 'Outbound rules', and 'Tags' are visible. A message at the bottom left says 'You can now check network connectivity with Reachability Analyzer' with a 'Run Reachability Analyzer' button.

Step 05 : On the Services menu, choose RDS.

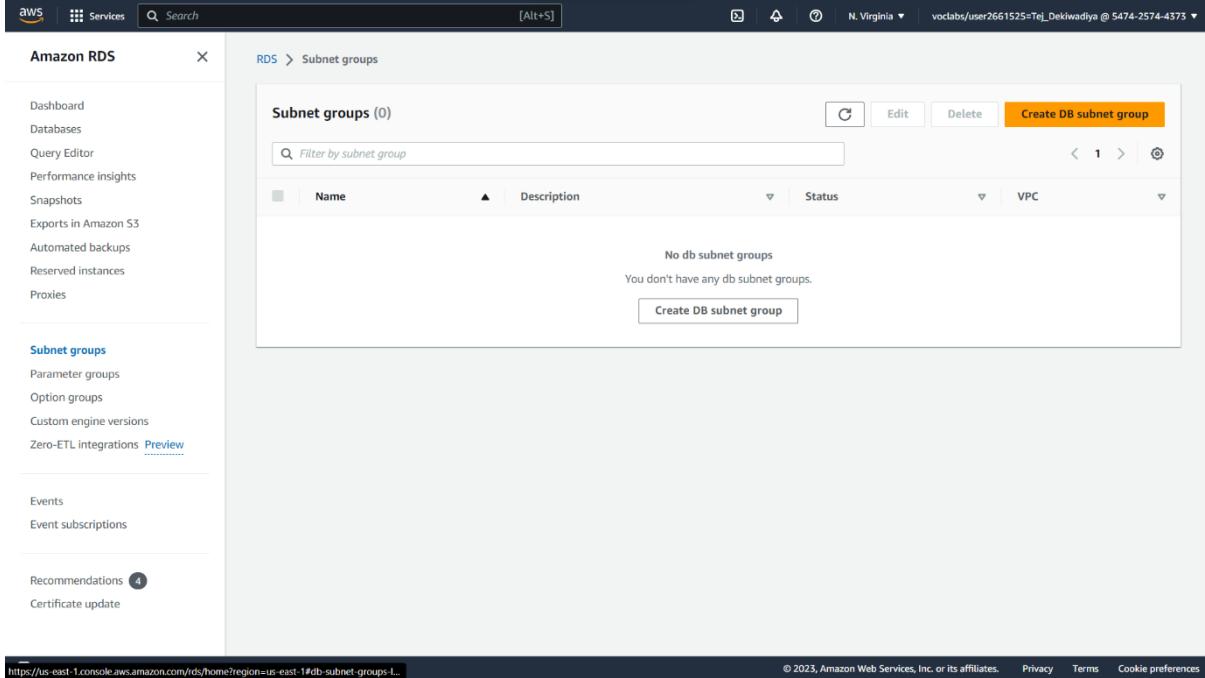
Snapshot :



The screenshot shows the AWS Services menu with the search bar set to 'RDS'. The results list includes the RDS service card, which is highlighted, showing it's a 'Managed Relational Database Service' with 'Top features' like Dashboard, Databases, Query Editor, Performance Insights, and Snapshots. Other services listed include AWS FIS, Database Migration Service, and Kinesis. Below the services, there are sections for 'Features' (Reserved instances, Proxies) and 'AWS Lambda functions' (AWS Lambda function, AWS Lambda feature).

Step 06 : In the left navigation pane, choose Subnet groups.

Snapshot :

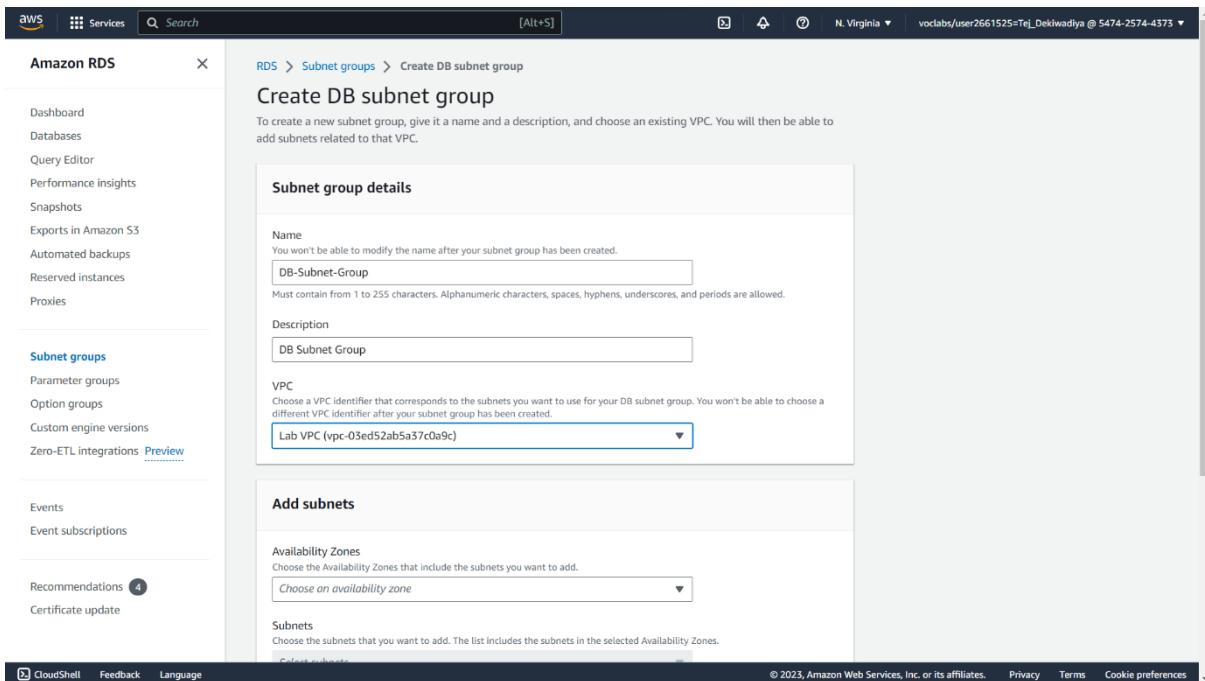


The screenshot shows the AWS RDS console. The left sidebar has 'Subnet groups' selected under the 'Amazon RDS' section. The main content area is titled 'Subnet groups (0)' and contains a message: 'No db subnet groups' and 'You don't have any db subnet groups.' A 'Create DB subnet group' button is visible at the bottom. The top bar shows the region as 'N. Virginia' and the user as 'voclabs/user2661525=Tej_Dekiwadiya @ 5474-2574-4373'. The URL in the address bar is 'https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#db-subnet-groups...'. The footer includes links for 'Privacy', 'Terms', and 'Cookie preferences'.

Step 07 : Choose Create DB Subnet Group then configure:

- **Name:** DB-Subnet-Group
- **Description:** DB Subnet Group
- **VPC:** Lab VPC

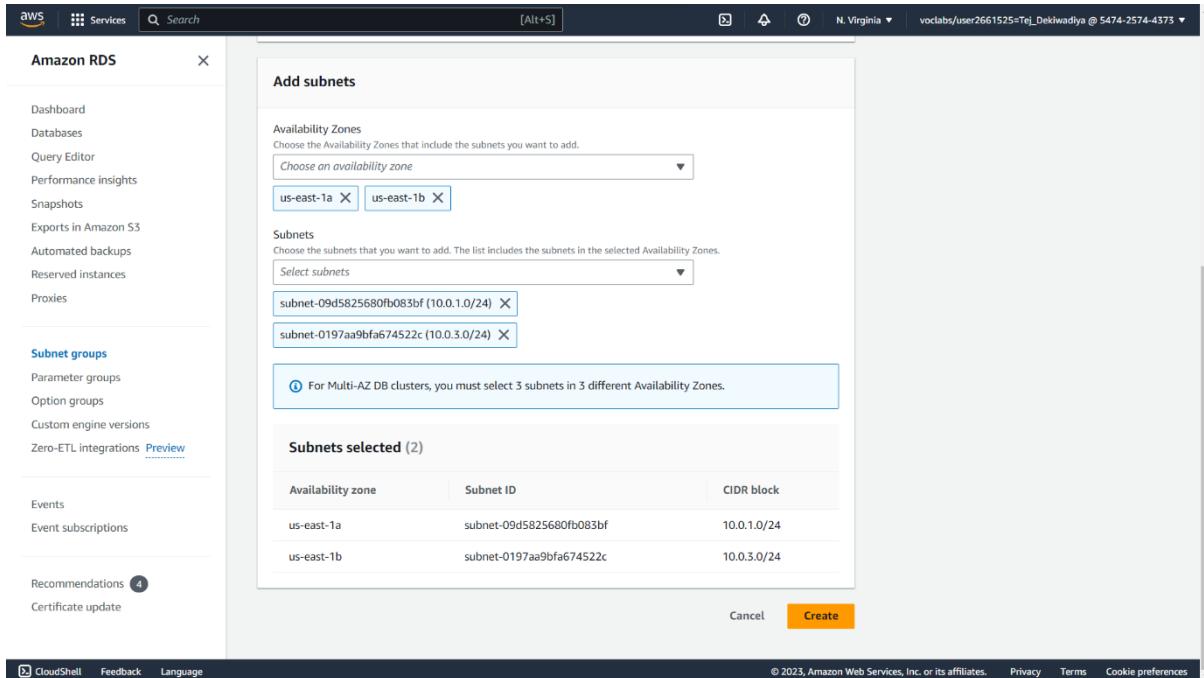
Snapshot :



The screenshot shows the 'Create DB subnet group' configuration page. The left sidebar has 'Subnet groups' selected under the 'Amazon RDS' section. The main content area is titled 'Create DB subnet group' and contains instructions: 'To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.' It has two main sections: 'Subnet group details' and 'Add subnets'. In 'Subnet group details', the 'Name' field is 'DB-Subnet-Group' and the 'Description' field is 'DB Subnet Group'. In 'Add subnets', the 'Availability Zones' dropdown is set to 'Choose on availability zone' and the 'Subnets' dropdown is set to 'Select subnets'. The top bar shows the region as 'N. Virginia' and the user as 'voclabs/user2661525=Tej_Dekiwadiya @ 5474-2574-4373'. The URL in the address bar is 'https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#db-subnet-groups>Create DB subnet group'. The footer includes links for 'CloudShell', 'Feedback', 'Language', 'Privacy', 'Terms', and 'Cookie preferences'.

Step 08 : Scroll down to the Add Subnets section. Expand the list of values under Availability Zones and select the first two zones: us-east-1a and us-east-1b. Expand the list of values under Subnets and select the subnets associated with the CIDR ranges 10.0.1.0/24 and 10.0.3.0/24 Than Choose Create.

Snapshot :



Add subnets

Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
us-east-1a X us-east-1b X

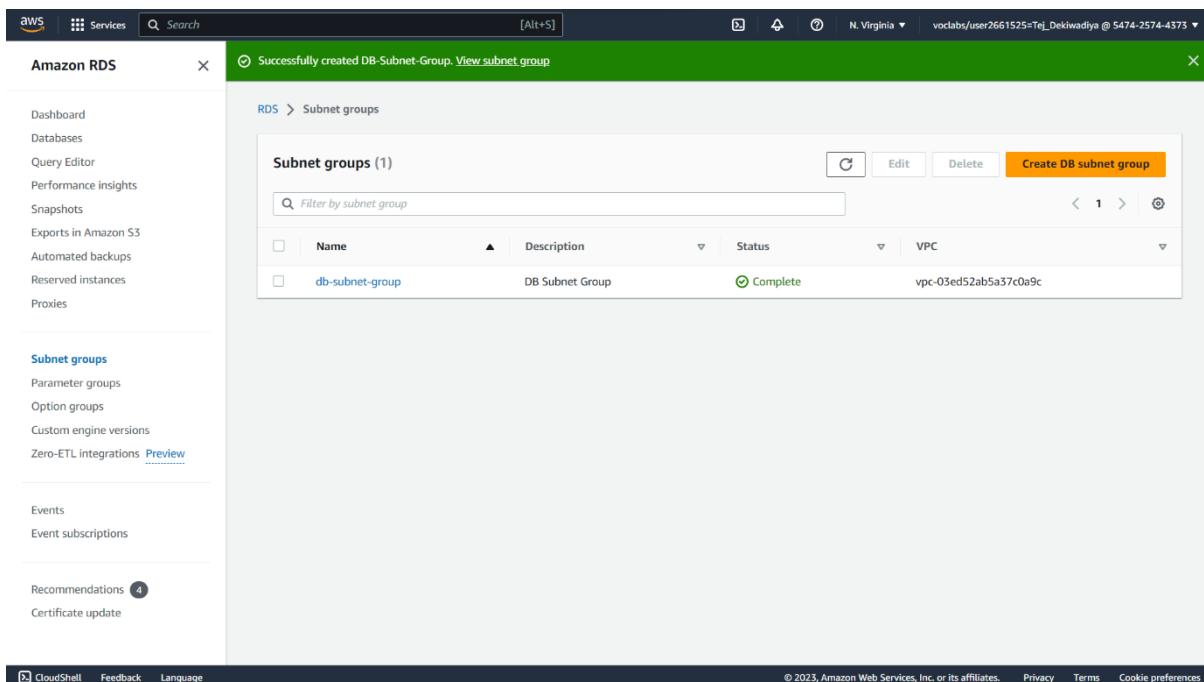
Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets
subnet-09d5825680fb083bf (10.0.1.0/24) X
subnet-0197aa9bfa674522c (10.0.3.0/24) X

ⓘ For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-east-1a	subnet-09d5825680fb083bf	10.0.1.0/24
us-east-1b	subnet-0197aa9bfa674522c	10.0.3.0/24

Create



Subnet groups (1)

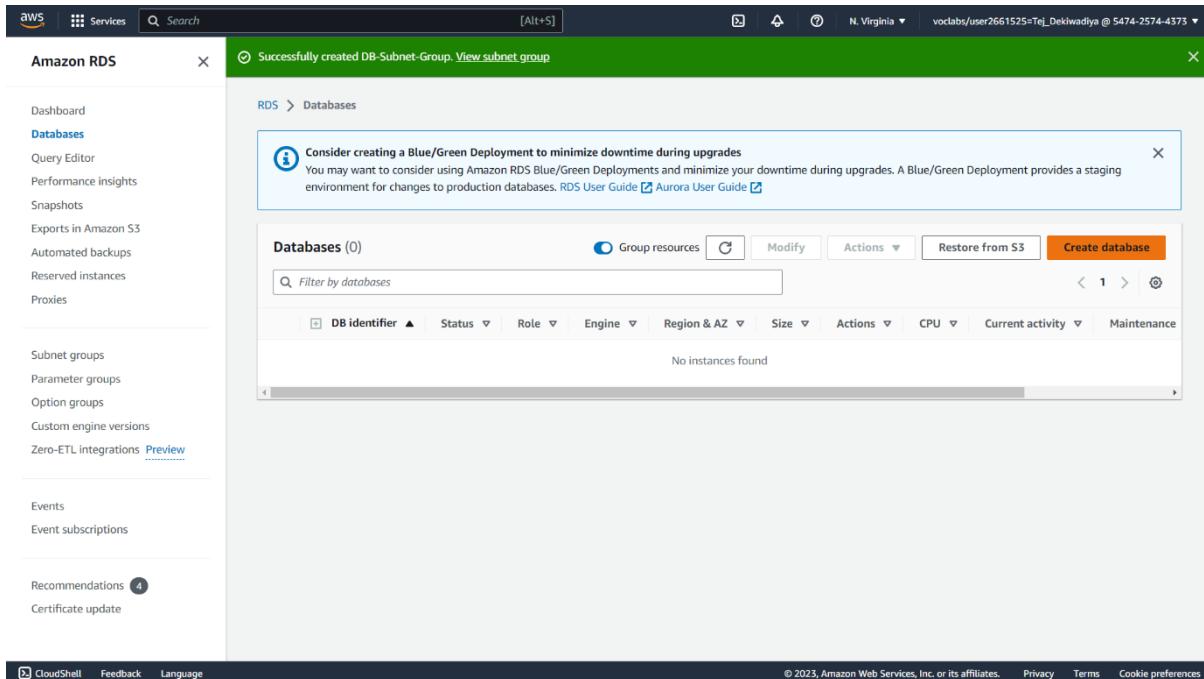
Filter by subnet group

Name	Description	Status	VPC
db-subnet-group	DB Subnet Group	Complete	vpc-03ed52ab5a37c0a9c

Create DB subnet group

Step 09 : In the left navigation pane, choose Databases. Choose Create database.

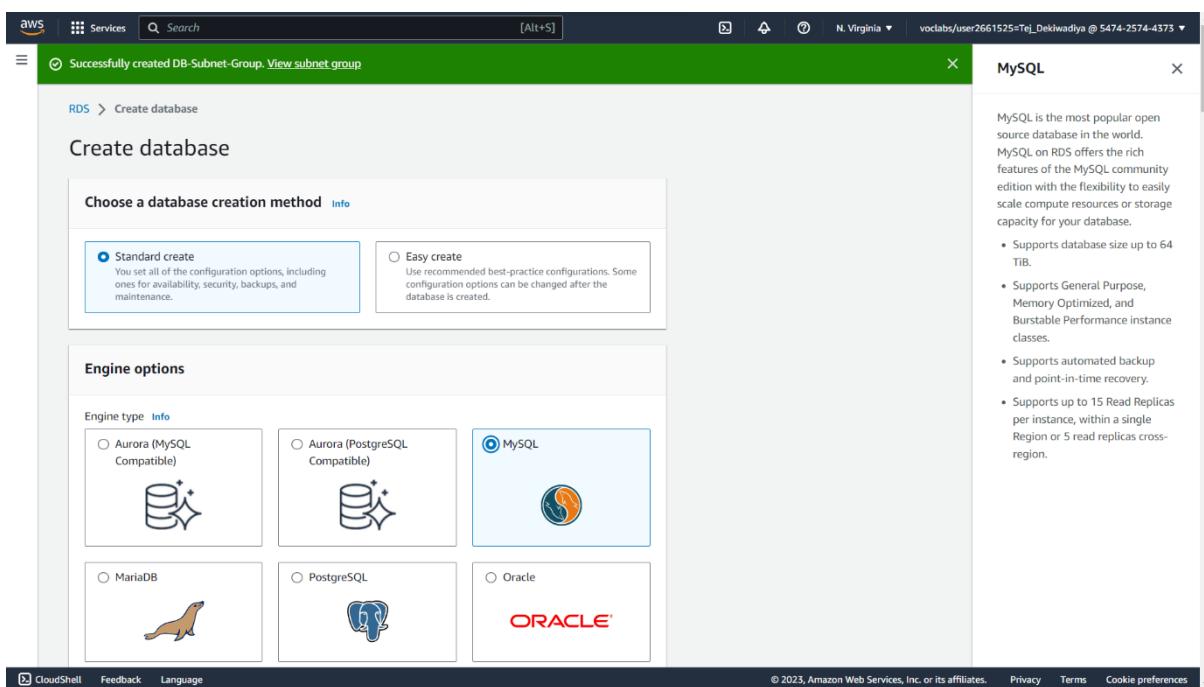
Snapshot :



The screenshot shows the AWS RDS Databases page. A green header bar at the top indicates "Successfully created DB-Subnet-Group. View subnet group". Below this, a modal window titled "Consider creating a Blue/Green Deployment to minimize downtime during upgrades" provides information about using Amazon RDS Blue/Green Deployments. The main "Databases (0)" table has a single row labeled "No instances found". At the top right of the table, there is a prominent orange "Create database" button. The left sidebar contains links for Dashboard, Databases (selected), Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations (with a "Preview" link), Events, Event subscriptions, Recommendations (with a notification icon), and Certificate update.

Step 10 : Select MySQL under Engine Options.

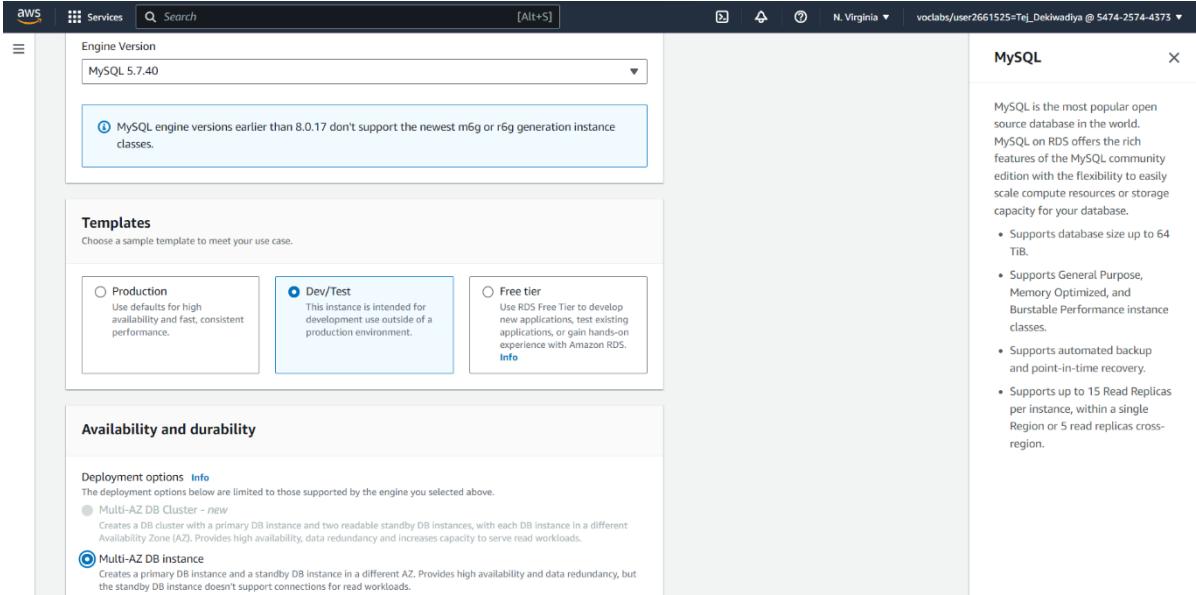
Snapshot :



The screenshot shows the "Create database" wizard. The first step, "Choose a database creation method", shows two options: "Standard create" (selected) and "Easy create". The second step, "Engine options", shows several database engines with their icons and names: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MySQL (selected), MariaDB, PostgreSQL, and Oracle. To the right of the MySQL option, a detailed description states: "MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database." It also lists several bullet points: "Supports database size up to 64 TiB.", "Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.", "Supports automated backup and point-in-time recovery.", and "Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region."

Step 11 : Under Templates choose Dev/Test Under Availability and durability choose Multi-AZ DB instance.

Snapshot :



MySQL

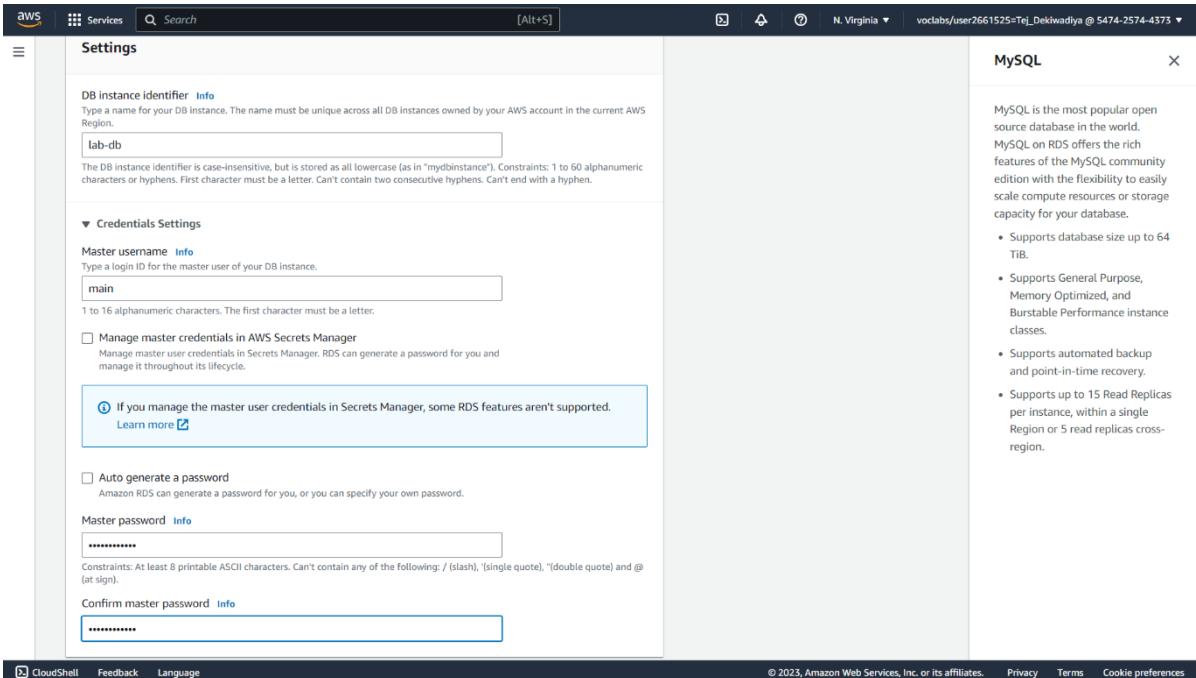
MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Step 12 : Under Settings, configure:

- DB instance identifier:** lab-db
- Master username:** main
- Master password:** lab-password
- Confirm password:** lab-password

Snapshot :



MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Step 13 : Under DB instance class, configure:

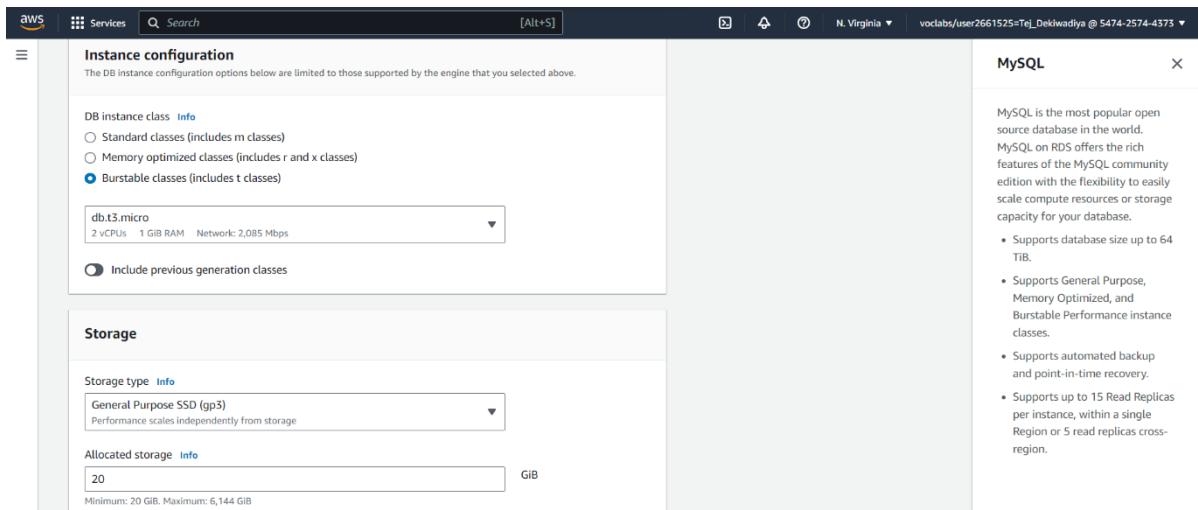
- Select Burstable classes (includes t classes).

- Select db.t3.micro

➤ Under Storage, configure:

- Storage type: General Purpose (SSD)
- Allocated storage: 20

Snapshot :



Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)

Allocated storage [Info](#)

20 GiB

MySQL

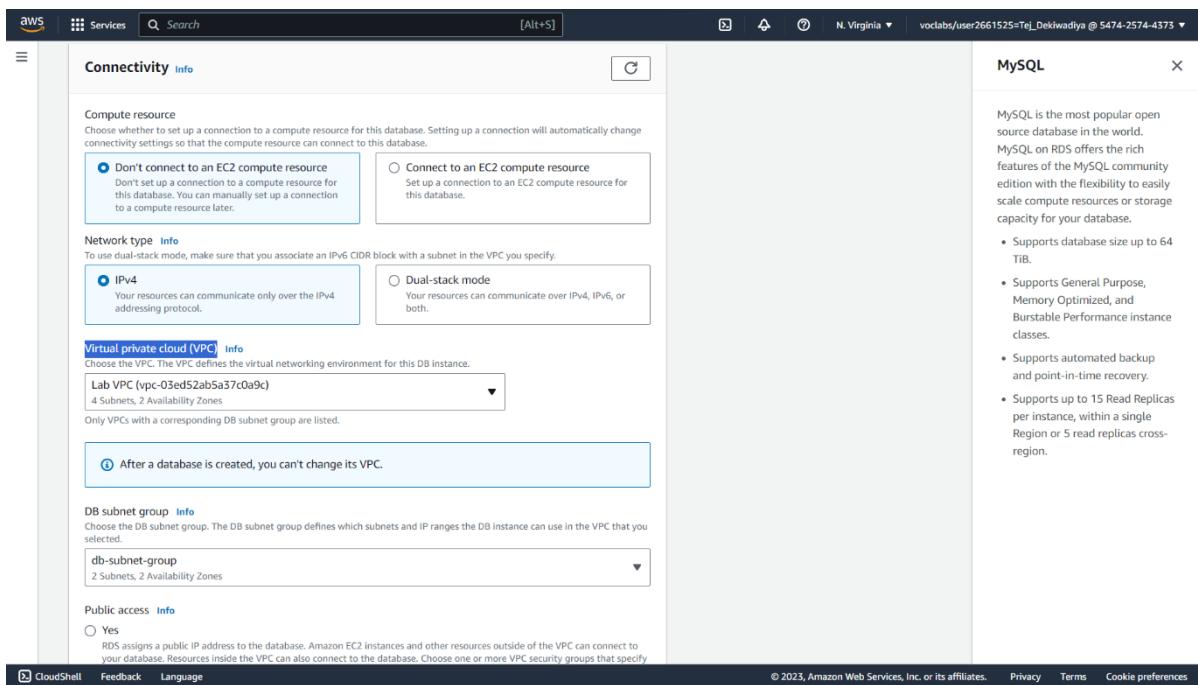
MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Step 14 : Under Connectivity, configure:

- **Virtual Private Cloud (VPC):** Lab VPC.

Snapshot :



Connectivity [Info](#)

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource

Set up a connection to an EC2 compute resource for this database.

Network type [Info](#)

To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4

Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode

Your resources can communicate over IPv4, IPv6, or both.

Virtual private cloud (VPC) [Info](#)

Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Lab VPC (vpc-03ed52ab5a37c0a9c)
4 Subnets, 2 Availability Zones

MySQL

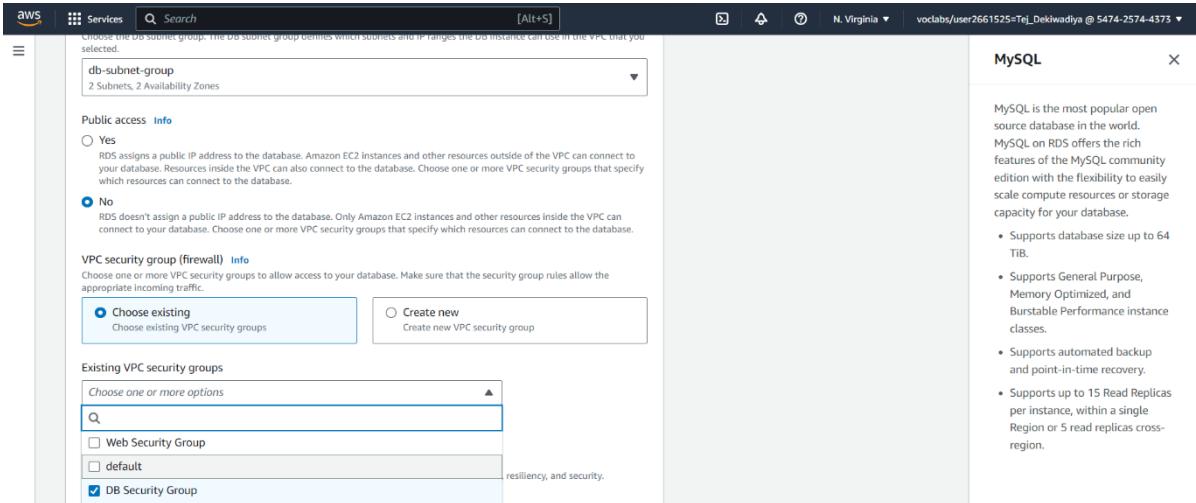
MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Step 15 : Under Existing VPC security groups, from the dropdown list:

- Choose DB Security Group.
- Deselect default.

Snapshot :

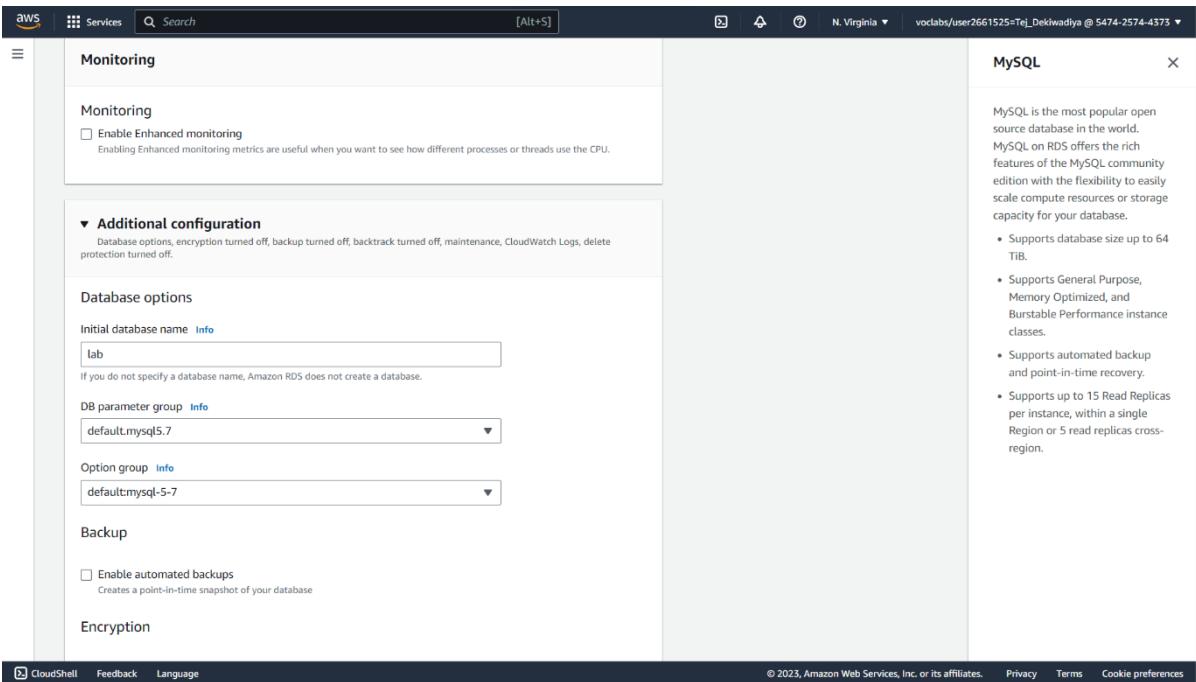


The screenshot shows the AWS RDS MySQL configuration interface. On the left, under 'Existing VPC security groups', the 'DB Security Group' checkbox is selected. On the right, a modal window titled 'MySQL' provides information about the MySQL database, including its popularity and various instance classes. The configuration section on the left also includes options for 'Public access' (set to 'No') and 'VPC security group (firewall)' (set to 'Choose existing').

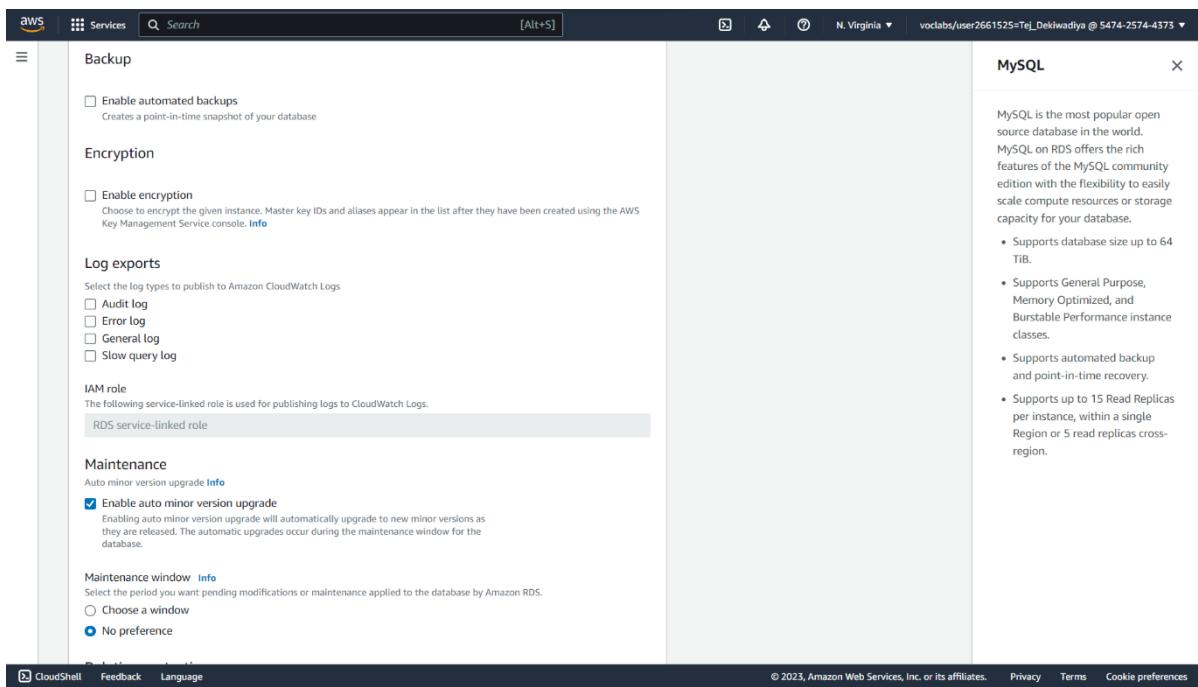
Step 16 : Expand Additional configuration, then configure:

- Initial database name: lab
- Uncheck Enable automatic backups.
- Uncheck Enable encryption
- Uncheck Enable Enhanced monitoring. Then Choose Create database

Snapshot :



The screenshot shows the AWS RDS MySQL configuration interface with the 'Additional configuration' section expanded. It includes fields for 'Initial database name' (set to 'lab'), 'DB parameter group' (set to 'default.mysql5.7'), and 'Option group' (set to 'default:mysql-5.7'). Other sections like 'Monitoring', 'Database options', 'Backup', and 'Encryption' are also visible. A modal window on the right provides general information about MySQL and its features.



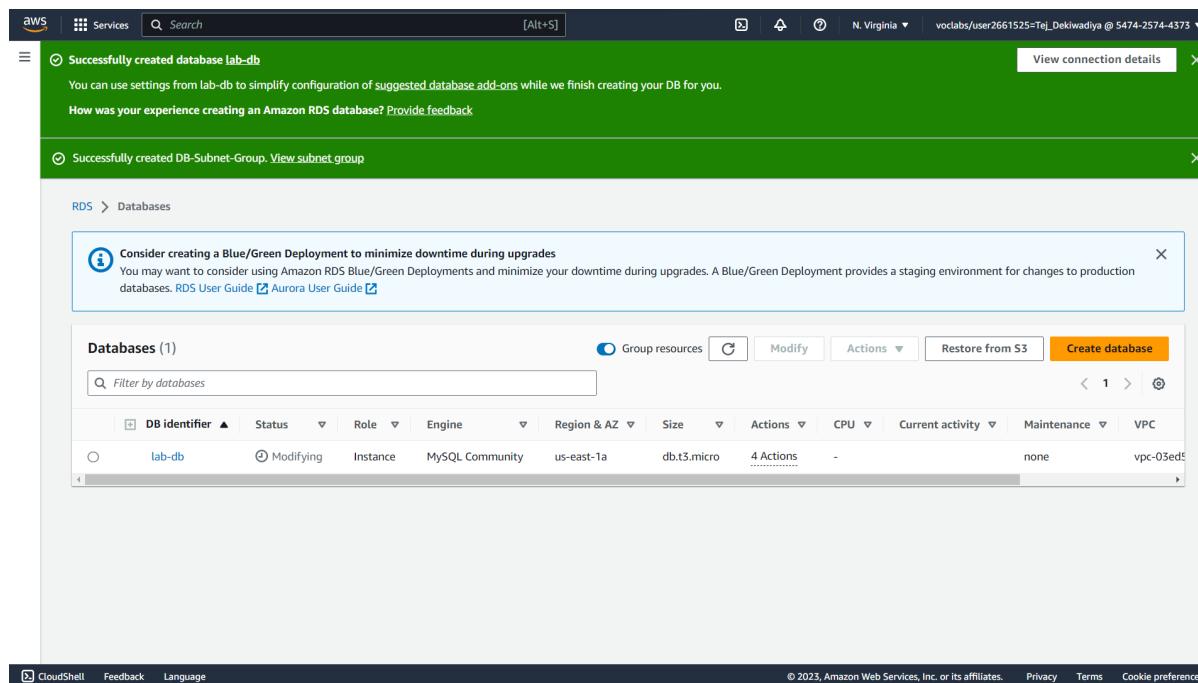
MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Step 17 : Choose lab-db (choose the link itself).

Snapshot :



Successfully created database lab-db

You can use settings from lab-db to simplify configuration of suggested database add-ons while we finish creating your DB for you.

How was your experience creating an Amazon RDS database? [Provide feedback](#)

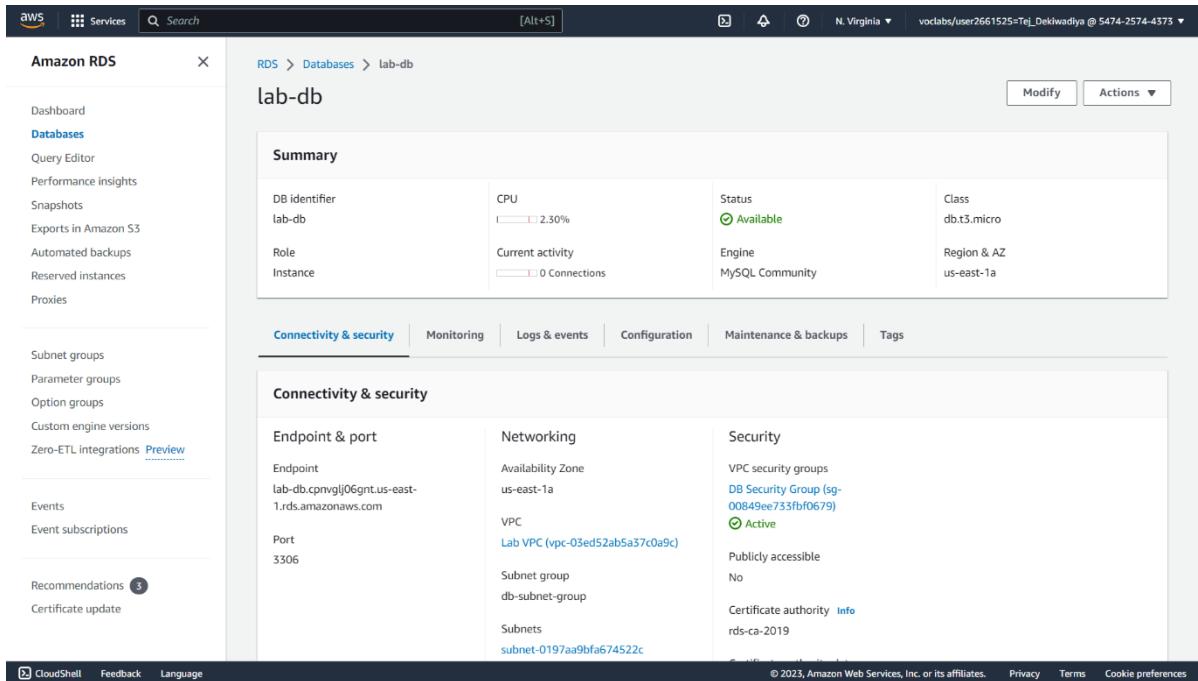
Successfully created DB-Subnet-Group. [View subnet group](#)

Databases (1)

DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU	Current activity	Maintenance	VPC
lab-db	Modifying	Instance	MySQL Community	us-east-1a	db.t3.micro	4 Actions	-	-	none	vpc-03ed5

Step 18 : Wait until Info changes to Modifying or Available.

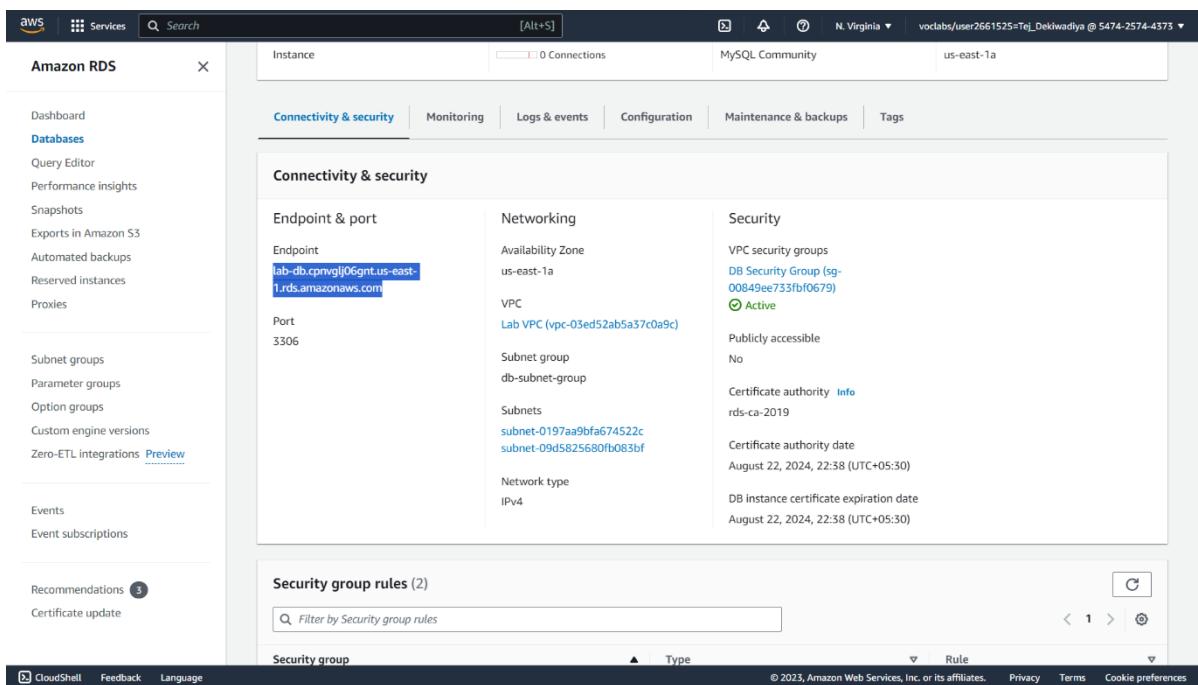
Snapshot :



The screenshot shows the AWS RDS console for the 'lab-db' database. In the 'Summary' section, the DB identifier is 'lab-db', CPU usage is 2.30%, Status is 'Available', and Class is 'db.t3.micro'. Under the 'Connectivity & security' tab, the Endpoint is 'lab-db.cpnvglj06gnt.us-east-1.rds.amazonaws.com' and the Port is 3306. The Networking section shows the Availability Zone as 'us-east-1a' and the VPC as 'Lab VPC (vpc-03ed52ab5a37c0a9c)'. The Security section indicates that the DB Security Group (sg-00849ee733fb0f0679) is active. The database is publicly accessible and has a certificate authority of 'rds-ca-2019'.

Step 19 : Scroll down to the Connectivity & security section and copy the Endpoint field.

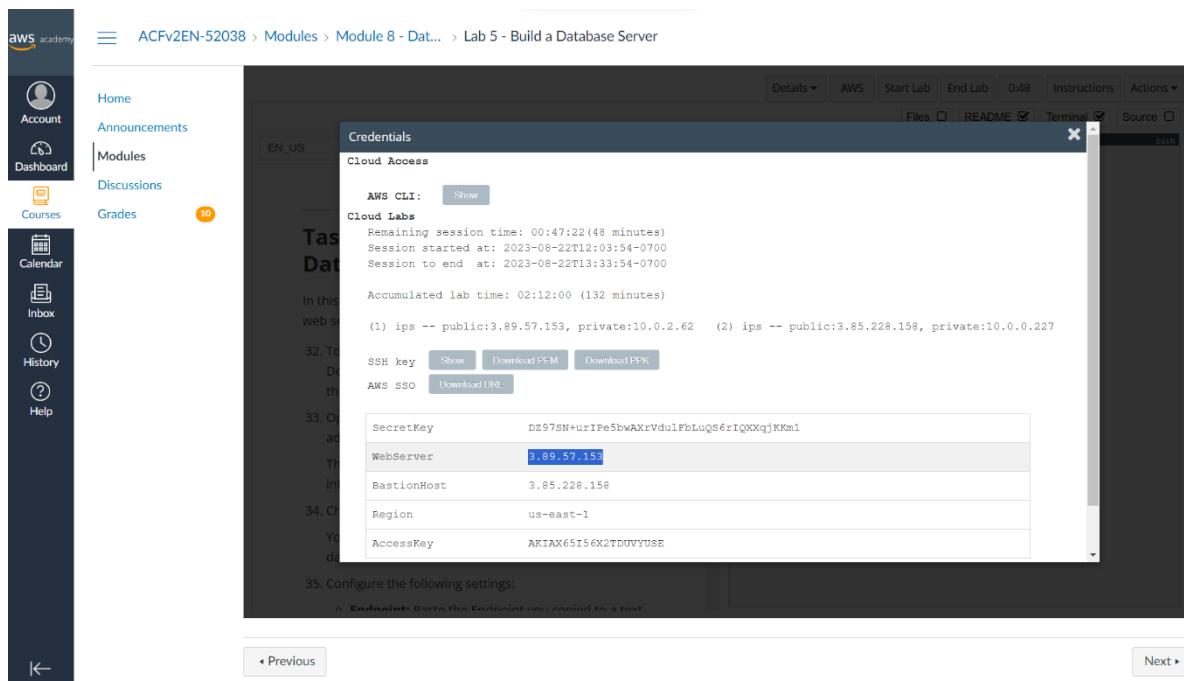
Snapshot :



The screenshot shows the AWS RDS console for the 'lab-db' database. In the 'Summary' section, the DB identifier is 'lab-db.cpnvglj06gnt.us-east-1.rds.amazonaws.com', CPU usage is 0%, Status is 'MySQL Community', and Class is 'us-east-1a'. Under the 'Connectivity & security' tab, the Endpoint is 'lab-db.cpnvglj06gnt.us-east-1.rds.amazonaws.com' and the Port is 3306. The Networking section shows the Availability Zone as 'us-east-1a' and the VPC as 'Lab VPC (vpc-03ed52ab5a37c0a9c)'. The Security section indicates that the DB Security Group (sg-00849ee733fb0f0679) is active. The database is publicly accessible and has a certificate authority of 'rds-ca-2019'. The 'Network type' is IPv4. In the 'Security group rules' section, there are two rules listed.

Step 20 : To copy the WebServer IP address, choose on the Details drop down menu above these instructions, and then choose Show. and Open a new web browser tab, paste the WebServer IP address and press Enter.

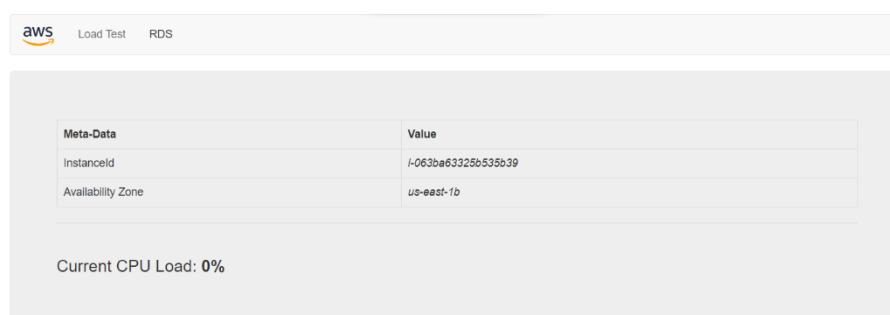
Snapshot :



The screenshot shows the AWS Academy interface for a lab session titled "ACFv2EN-52038 > Modules > Module 8 - Dat... > Lab 5 - Build a Database Server". A modal window titled "Credentials" is open, specifically the "Cloud Access" section. In this section, there is a table with the following data:

SecretKey	DZ978N+urIPe5bwAXrVdulFbLuQS6rIQXxqjKKm1
WebServer	3.89.57.153
BastionHost	3.85.228.158
Region	us-east-1
AccessKey	AKIAK65I56X2TDUVVUSE

Below the table, it says "35. Configure the following settings:" followed by a note "Estimated: 5 min, the Estimated time is based on the last test". At the bottom of the modal are "Previous" and "Next" buttons.

The screenshot shows the AWS RDS console for an instance. At the top, there are tabs for "aws", "Load Test", and "RDS". Below the tabs, there is a table titled "Meta-Data" with the following data:

Meta-Data	Value
InstanceId	i-063ba63325b535539
Availability Zone	us-east-1b

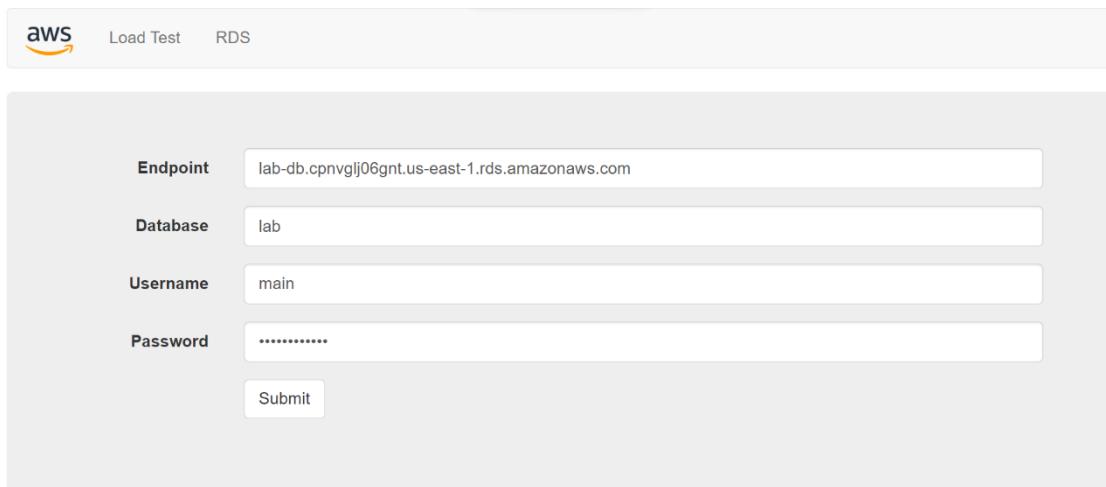
Below the table, it says "Current CPU Load: 0%".

3.89.57.153/rds.php

Step 21 : Choose the RDS link at the top of the page then Configure the following settings:

- Endpoint: Paste the Endpoint you copied to a text editor earlier
- Database: lab
- Username: main
- Password: lab-password
- Choose Submit

Snapshot :



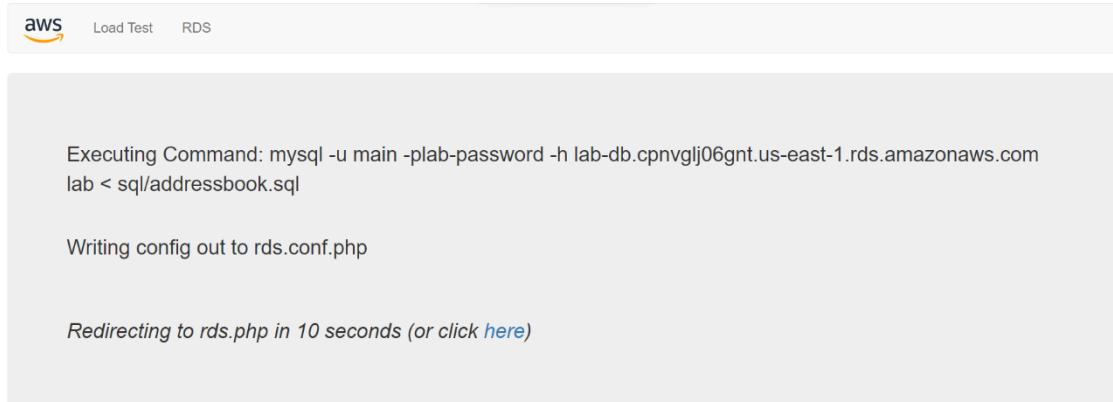
aws Load Test RDS

Endpoint: lab-db.cpnvglij06gnt.us-east-1.rds.amazonaws.com

Database: lab

Username: main

Password:
Submit



aws Load Test RDS

Executing Command: mysql -u main -plab-password -h lab-db.cpnvglij06gnt.us-east-1.rds.amazonaws.com lab < sql/addressbook.sql

Writing config out to rds.conf.php

Redirecting to rds.php in 10 seconds (or click [here](#))

Step 22 : Test the web application by adding, editing and removing contacts.

Snapshot :

aws Load Test RDS

Address Book

Last name	First name	Phone	Email	Admin
Doe	Jane	010-110-1101	janed@someotheraddress.org	Add Contact Edit Remove
Johnson	Roberto	123-456-7890	robertoj@someaddress.com	Edit Remove

aws Load Test RDS

Address Book

Add Contact

Last Name: Demo
 First Name: 1
 Phone: 012355
 Email: mskk@vbn.com

Last name	First name	Phone	Email	Admin
Johnson	Roberto	123-456-7890	robertoj@someaddress.com	Add Contact Edit Remove

aws Load Test RDS

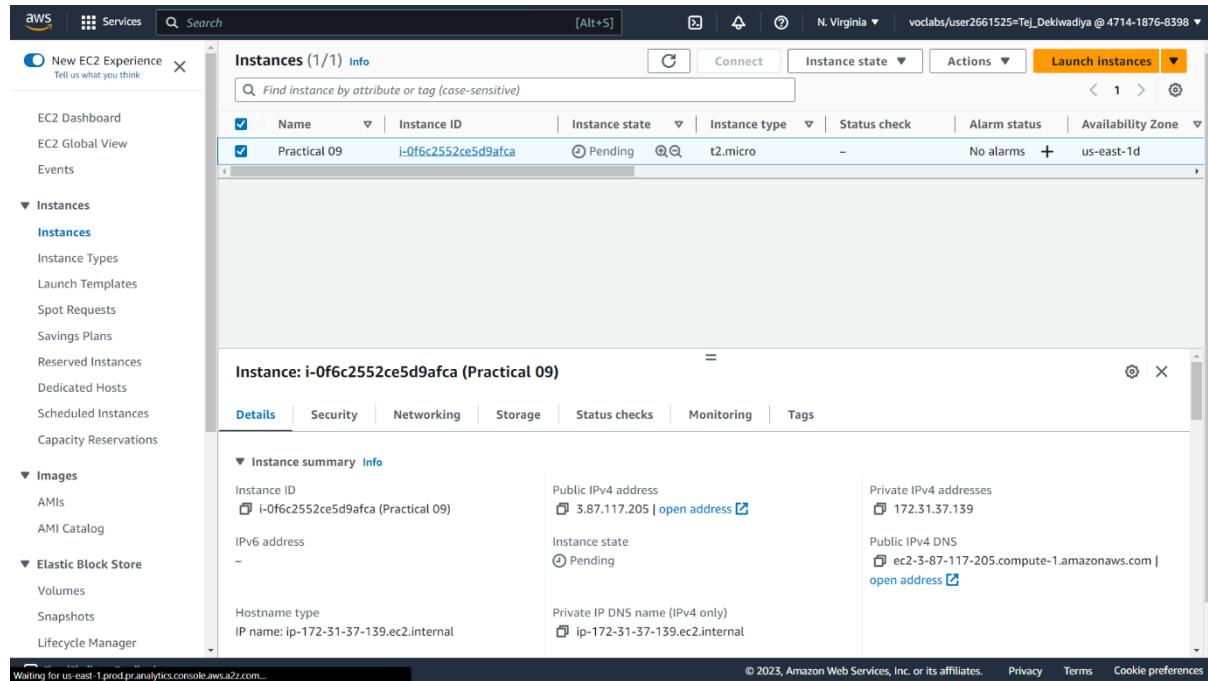
Address Book

Last name	First name	Phone	Email	Admin
Demo	1	012355	mskk@vbn.com	Add Contact Edit Remove
Johnson	Roberto	123-456-7890	robertoj@someaddress.com	Edit Remove

Practical 09 : Containerization with Docker

Step 01 : Create Instances in EC2

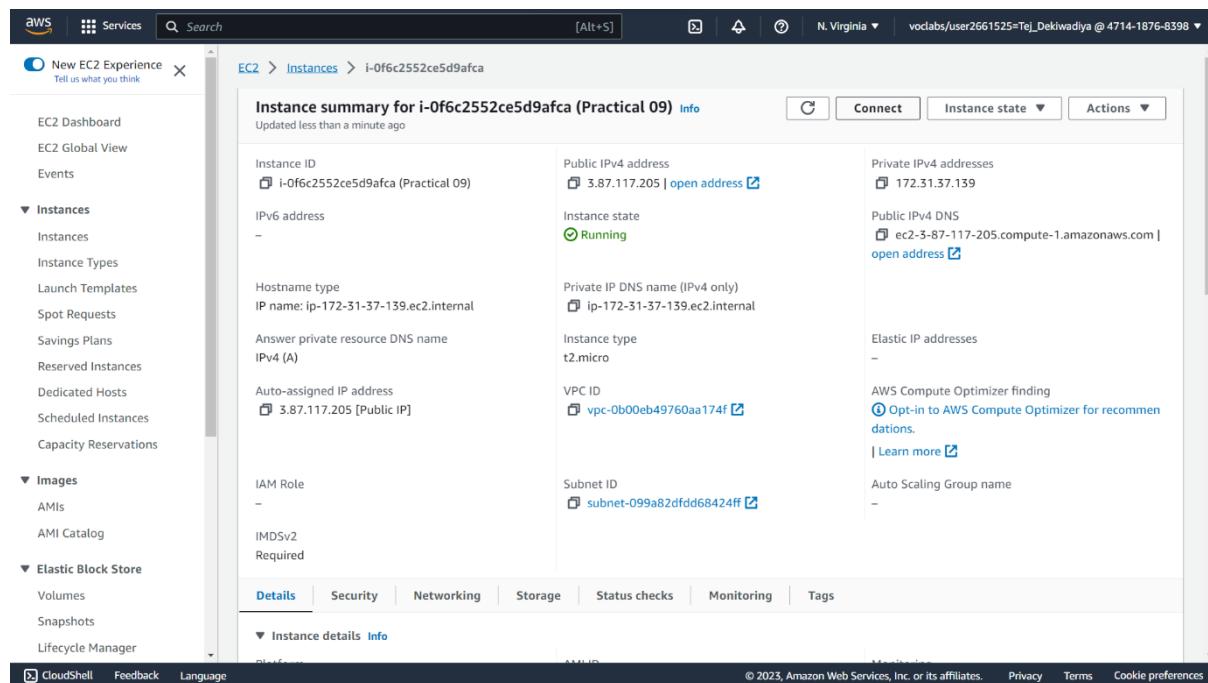
Snapshot :



The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with links like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area displays a table titled 'Instances (1/1) Info'. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. One row is selected, showing 'Practical 09' with the instance ID 'i-0f6c2552ce5d9afca', state 'Pending', type 't2.micro', and zone 'us-east-1d'. Below the table, a detailed view for 'Instance: i-0f6c2552ce5d9afca (Practical 09)' is shown. It includes tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. Under the Details tab, there's an 'Instance summary' section with fields for Instance ID (i-0f6c2552ce5d9afca), Public IPv4 address (3.87.117.205), Private IP4 addresses (172.31.37.139), IPv6 address (empty), Instance state (Pending), Public IPv4 DNS (ec2-3-87-117-205.compute-1.amazonaws.com), Hostname type (IP name: ip-172-31-37-139.ec2.internal), Private IP DNS name (IPv4 only) (ip-172-31-37-139.ec2.internal), and Elastic IP addresses (empty). At the bottom, there are links for CloudShell, Feedback, and Language.

Step 02 : Click On Connect Button

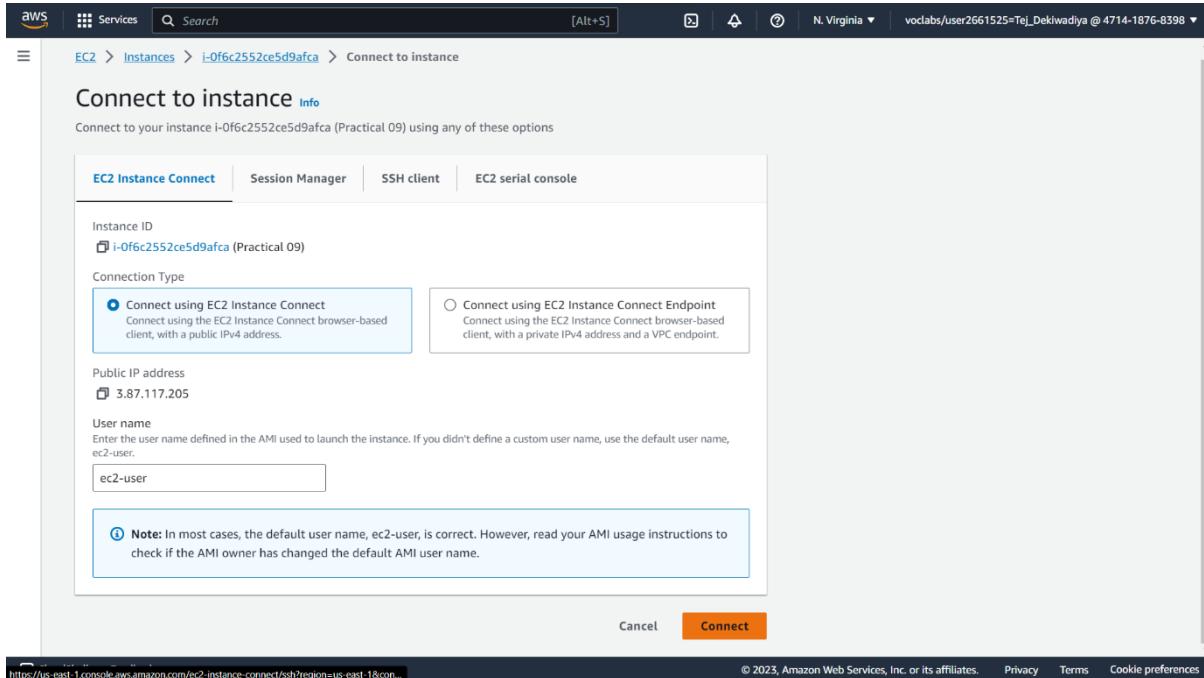
Snapshot :



This screenshot shows the same EC2 Instances page as the previous one, but the instance 'Practical 09' is now listed as 'Running'. The 'Connect' button is highlighted in orange at the top right of the instance card. The rest of the interface is identical to the first screenshot, including the sidebar and the detailed instance view below.

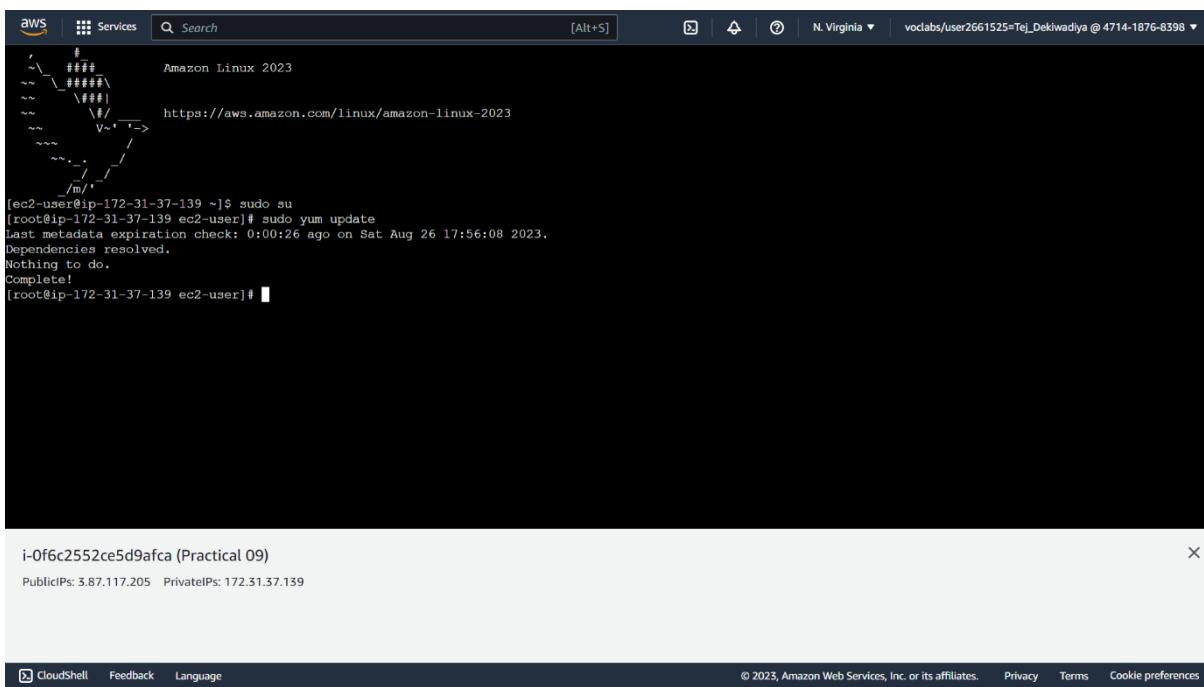
Step 03 : After Click On Connect button page rendering on connect to instance page than Connect to terminal

Snapshot :



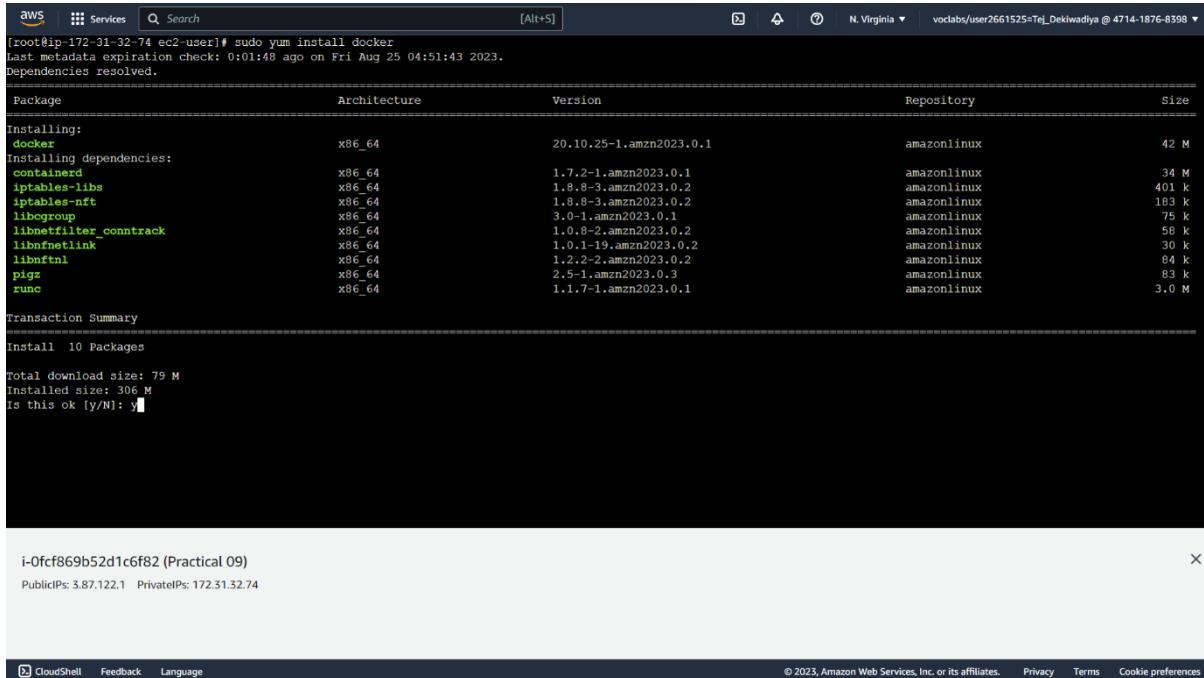
Step 04 : run sudo su and sudo yum update command

Snapshot :



Step 05 : after run previous command then write **sudo yum install docker**

Snapshot :



```
[root@ip-172-31-32-74 ec2-user]# sudo yum install docker
Last metadata expiration check: 0:01:48 ago on Fri Aug 25 04:51:43 2023.
Dependencies resolved.

Installing:
  docker           x86_64        20.10.25-1.amzn2023.0.1      amazonlinux          42 M
Installing dependencies:
  containerd       x86_64        1.7.2-1.amzn2023.0.1      amazonlinux          34 M
  iptables-libc   x86_64        1.8.8-3.amzn2023.0.2      amazonlinux          401 k
  iptables-nft    x86_64        1.8.8-3.amzn2023.0.2      amazonlinux          183 k
  libcgroup        x86_64        3.0-1.amzn2023.0.1      amazonlinux          75 k
  libnetfilter_conntrack x86_64  1.0.8-2.amzn2023.0.2      amazonlinux          58 k
  libnftnl         x86_64        1.0.1-19.amzn2023.0.2     amazonlinux          30 k
  libnftnl        x86_64        1.2.2-2.amzn2023.0.2      amazonlinux          84 k
  pigz             x86_64        2.5-1.amzn2023.0.3      amazonlinux          83 k
  runc             x86_64        1.1.7-1.amzn2023.0.1      amazonlinux          3.0 M

Transaction Summary

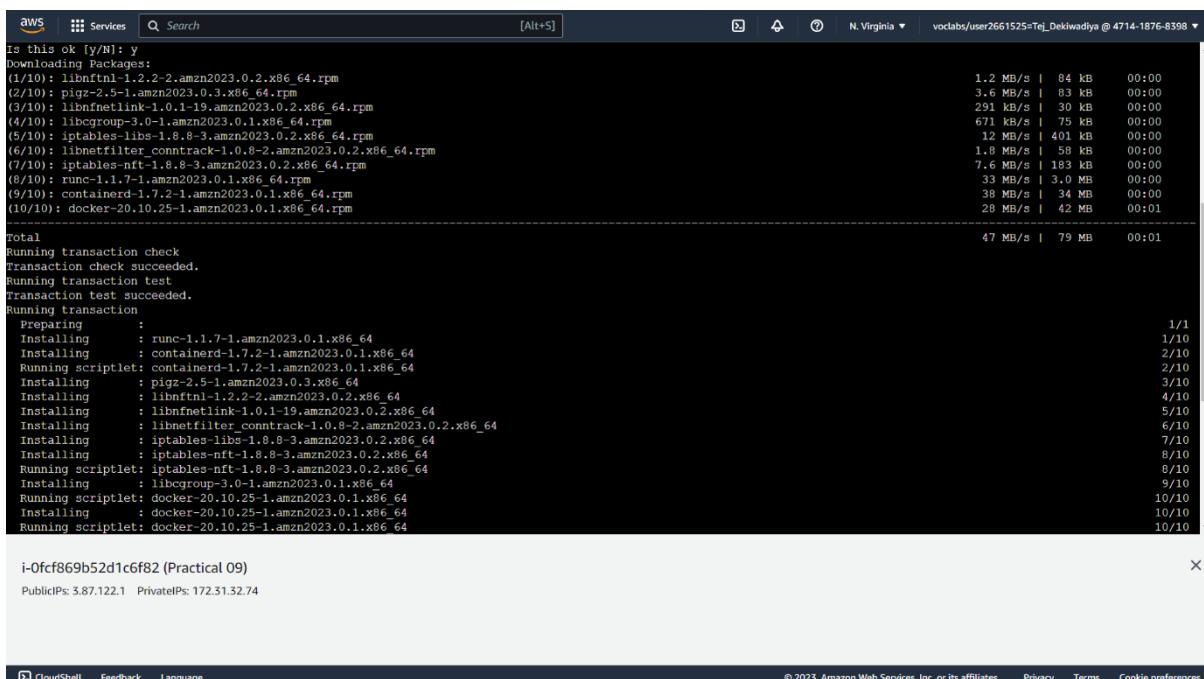
Install 10 Packages

Total download size: 79 M
Installed size: 306 M
Is this ok [y/N]: y■
```

i-Ofcf869b52d1c6f82 (Practical 09)

PublicIPs: 3.87.122.1 PrivateIPs: 172.31.32.74

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

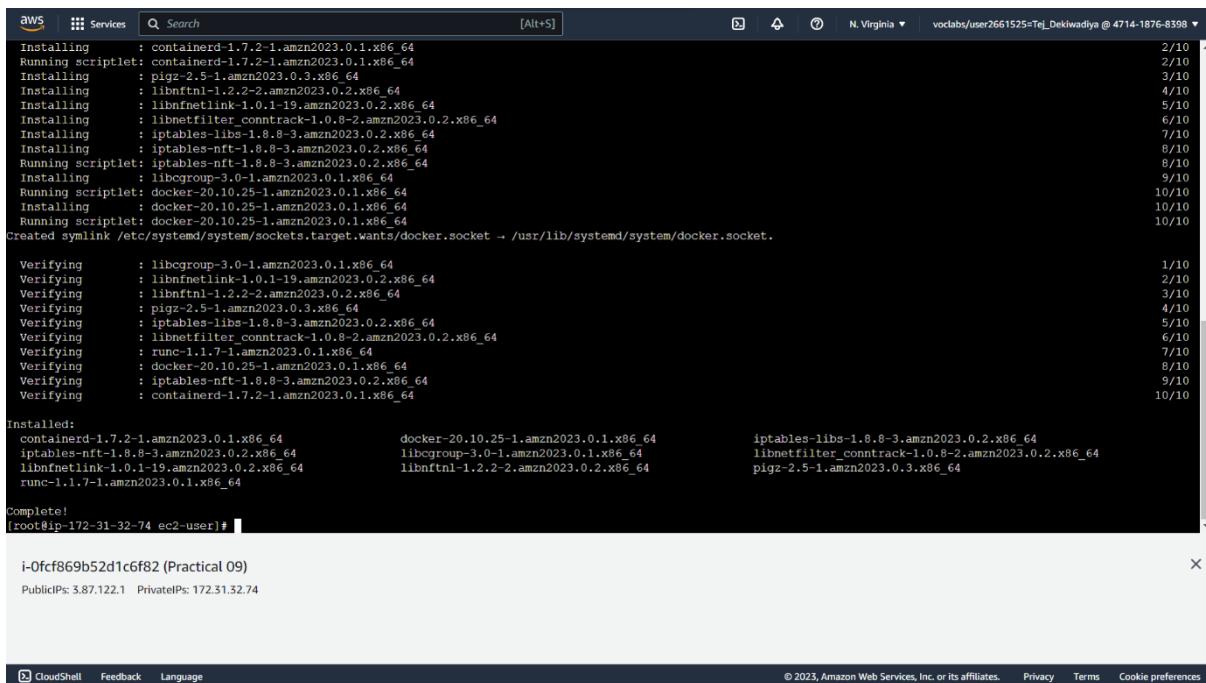


```
Is this ok [y/N]: y
Downloading Packages:
(1/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm 1.2 MB/s | 84 KB 00:00
(2/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm 3.6 MB/s | 83 KB 00:00
(3/10): libnftnl-link-1.0.1-19.amzn2023.0.2.x86_64.rpm 291 kB/s | 30 kB 00:00
(4/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm 671 kB/s | 75 kB 00:00
(5/10): iptables-libc-1.8.8-3.amzn2023.0.2.x86_64.rpm 12 MB/s | 401 kB 00:00
(6/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 1.8 MB/s | 58 kB 00:00
(7/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm 7.6 MB/s | 183 kB 00:00
(8/10): runc-1.1.7-1.amzn2023.0.1.x86_64.rpm 33 MB/s | 3.0 MB 00:00
(9/10): containerd-1.7.2-1.amzn2023.0.1.x86_64.rpm 38 MB/s | 34 MB 00:00
(10/10): docker-20.10.25-1.amzn2023.0.1.x86_64.rpm 28 MB/s | 42 MB 00:01

Total 47 MB/s | 79 MB 00:01
Running transaction check
transaction check succeeded.
Running transaction test
transaction test succeeded.
Running transaction
  Preparing: 1/1
  Installing : runc-1.1.7-1.amzn2023.0.1.x86_64 1/10
  Installing : containerd-1.7.2-1.amzn2023.0.1.x86_64 1/10
  Running scriptlet: containerd-1.7.2-1.amzn2023.0.1.x86_64 2/10
  Installing : pigz-2.5-1.amzn2023.0.3.x86_64 2/10
  Installing : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 3/10
  Installing : libnftnl-link-1.0.1-19.amzn2023.0.2.x86_64 4/10
  Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 5/10
  Installing : iptables-libc-1.8.8-3.amzn2023.0.2.x86_64 6/10
  Installing : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 7/10
  Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 8/10
  Installing : libcgroup-3.0-1.amzn2023.0.1.x86_64 9/10
  Running scriptlet: docker-20.10.25-1.amzn2023.0.1.x86_64 10/10
  Installing : docker-20.10.25-1.amzn2023.0.1.x86_64 10/10
  Running scriptlet: docker-20.10.25-1.amzn2023.0.1.x86_64 10/10

i-Ofcf869b52d1c6f82 (Practical 09)
PublicIPs: 3.87.122.1 PrivateIPs: 172.31.32.74
```

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



```

aws Services Search [Alt+S] N. Virginia v voclabs/user2661525=Tej_Dekiwadiya @ 4714-1876-8398 ▾
Installing : containerd-1.7.2-1.amzn2023.0.1.x86_64 2/10
Running scriptlet: containerd-1.7.2-1.amzn2023.0.1.x86_64 2/10
Installing : pigz-2.5-1.amzn2023.0.3.x86_64 3/10
Installing : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 4/10
Installing : libnftnl-link-1.0.1-19.amzn2023.0.2.x86_64 5/10
Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Installing : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 7/10
Installing : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 8/10
Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 8/10
Installing : libcgroup-3.0-1.amzn2023.0.1.x86_64 9/10
Running scriptlet: docker-20.10.25-1.amzn2023.0.1.x86_64 10/10
Installing : docker-20.10.25-1.amzn2023.0.1.x86_64 10/10
Running scriptlet: docker-20.10.25-1.amzn2023.0.1.x86_64 10/10
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

Verifying : libcgroup-3.0-1.amzn2023.0.1.x86_64 1/10
Verifying : libnftnl-link-1.0.1-19.amzn2023.0.2.x86_64 2/10
Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 3/10
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64 4/10
Verifying : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 5/10
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Verifying : runc-1.1.7-1.amzn2023.0.1.x86_64 7/10
Verifying : docker-20.10.25-1.amzn2023.0.1.x86_64 8/10
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 9/10
Verifying : containerd-1.7.2-1.amzn2023.0.1.x86_64 10/10

Installed:
containerd-1.7.2-1.amzn2023.0.1.x86_64 docker-20.10.25-1.amzn2023.0.1.x86_64
iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 libcgroup-3.0-1.amzn2023.0.1.x86_64
libnftnl-link-1.0.1-19.amzn2023.0.2.x86_64 libnftnl-1.2.2-2.amzn2023.0.2.x86_64
runc-1.1.7-1.amzn2023.0.1.x86_64 iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
pigz-2.5-1.amzn2023.0.3.x86_64 libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64

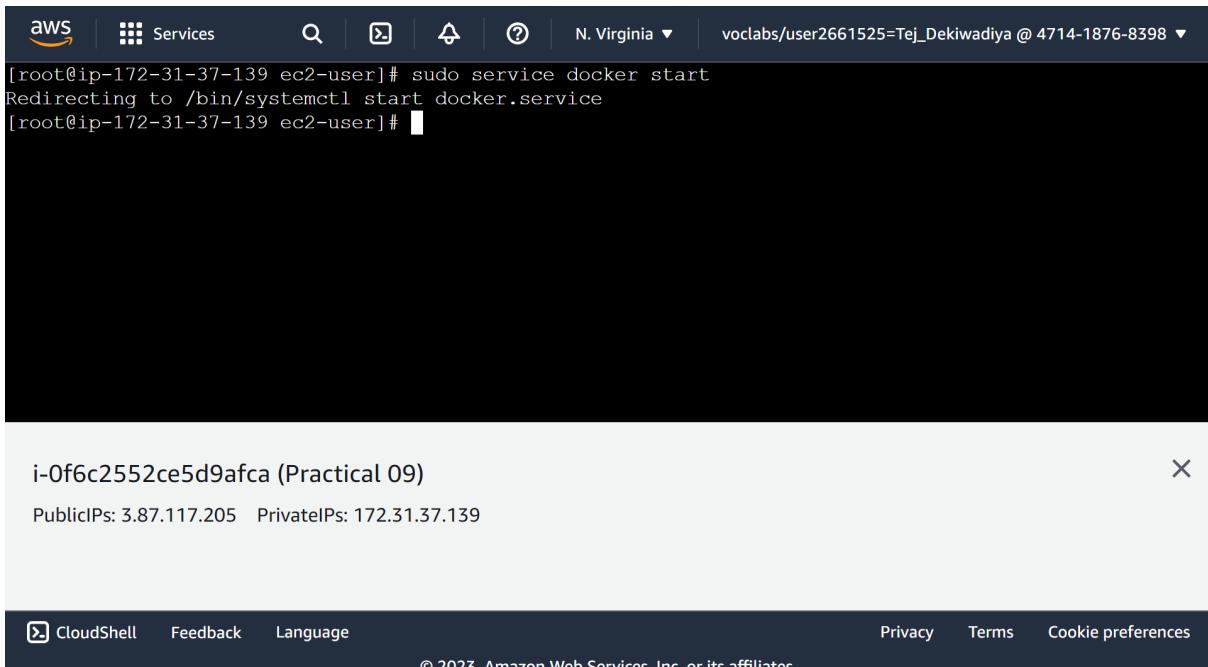
Complete!
[root@ip-172-31-32-74 ec2-user]# i-0fcf869b52d1c6f82 (Practical 09)
Public IPs: 3.87.122.1 Private IPs: 172.31.32.74

```

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 06 : run sudo service docker start command

Snapshot :



```

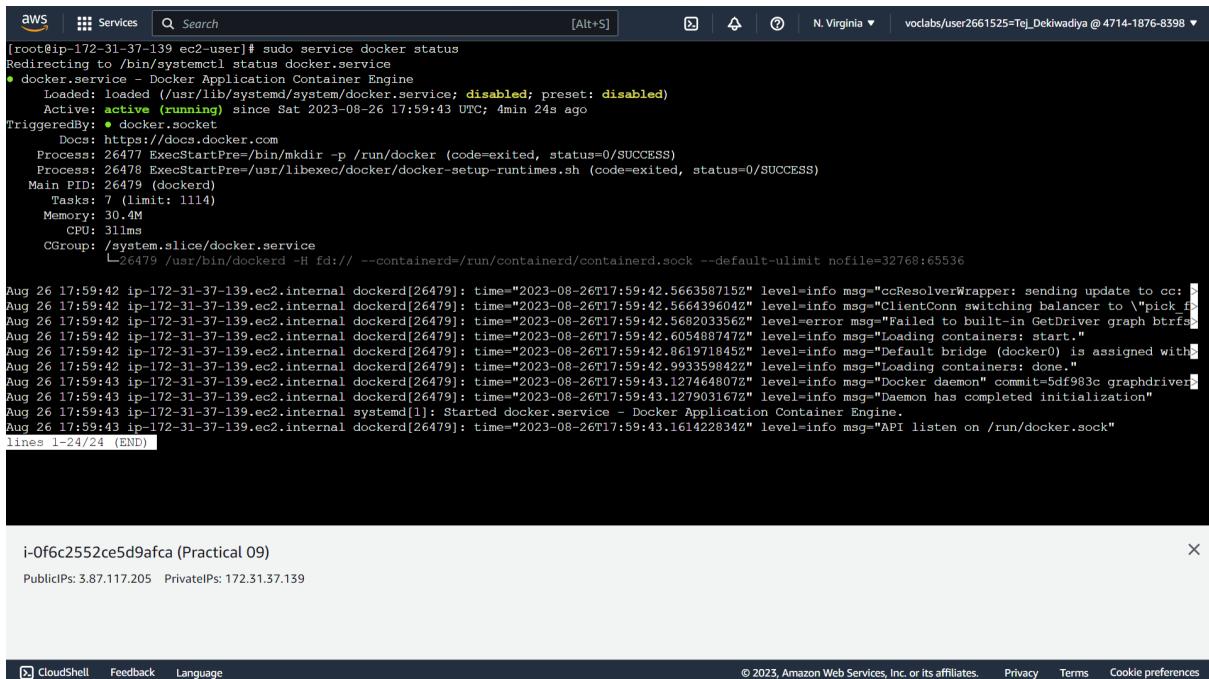
aws Services Search [Alt+S] N. Virginia v voclabs/user2661525=Tej_Dekiwadiya @ 4714-1876-8398 ▾
[root@ip-172-31-37-139 ec2-user]# sudo service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-31-37-139 ec2-user]# i-0f6c2552ce5d9afca (Practical 09)
Public IPs: 3.87.117.205 Private IPs: 172.31.37.139

```

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 07 : run sudo service docker status

Snapshot :



```
aws Services Search [Alt+S] N. Virginia v vodlabs/user2661525=Tej_Dekiwadiya @ 4714-1876-8398
[root@ip-172-31-37-139 ec2-user]# sudo service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; preset: disabled)
    Active: active (running) since Sat 2023-08-26 17:59:43 UTC; 4min 24s ago
TriggeredBy: ● docker.socket
  Docs: https://docs.docker.com
 Process: 26477 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
 Process: 26478 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
 Main PID: 26479 (dockerd)
   Tasks: 7 (limit: 1114)
  Memory: 30.4M
     CPU: 311ms
    CGroup: /system.slice/docker.service
            └─26479 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

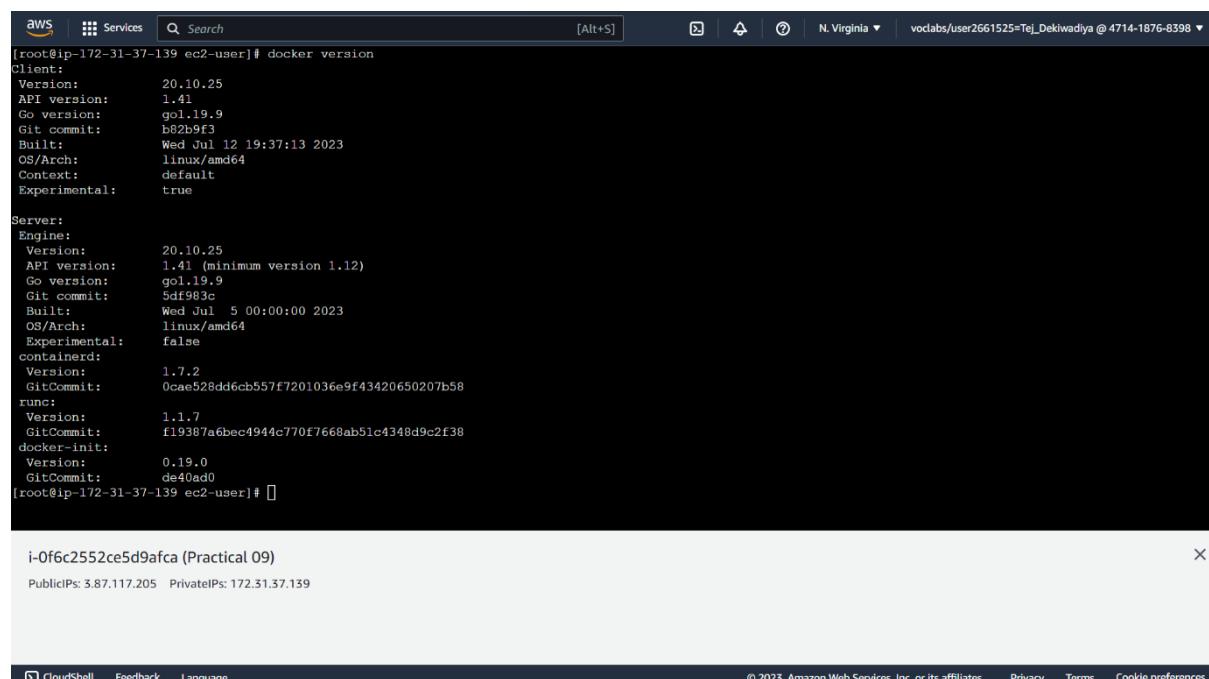
Aug 26 17:59:42 ip-172-31-37-139.ec2.internal dockerd[26479]: time="2023-08-26T17:59:42.566358715Z" level=info msg="ccResolverWrapper: sending update to cc: <nil>" 
Aug 26 17:59:42 ip-172-31-37-139.ec2.internal dockerd[26479]: time="2023-08-26T17:59:42.566439604Z" level=info msg="ClientConn switching balancer to 'pick_first'" 
Aug 26 17:59:42 ip-172-31-37-139.ec2.internal dockerd[26479]: time="2023-08-26T17:59:42.568203356Z" level=error msg="Failed to built-in GetDriver graph btrfs" 
Aug 26 17:59:42 ip-172-31-37-139.ec2.internal dockerd[26479]: time="2023-08-26T17:59:42.605488747Z" level=info msg="Loading containers: start." 
Aug 26 17:59:42 ip-172-31-37-139.ec2.internal dockerd[26479]: time="2023-08-26T17:59:42.861971845Z" level=info msg="Default bridge (docker0) is assigned with IP address 172.31.37.1 (brd smc) brd smc" 
Aug 26 17:59:42 ip-172-31-37-139.ec2.internal dockerd[26479]: time="2023-08-26T17:59:42.993359842Z" level=info msg="Loading containers: done." 
Aug 26 17:59:43 ip-172-31-37-139.ec2.internal dockerd[26479]: time="2023-08-26T17:59:43.127903167Z" level=info msg="Docker daemon" commit=5df983c graphdriver=overlay2 
Aug 26 17:59:43 ip-172-31-37-139.ec2.internal dockerd[26479]: time="2023-08-26T17:59:43.127903167Z" level=info msg="Daemon has completed initialization" 
Aug 26 17:59:43 ip-172-31-37-139.ec2.internal systemd[1]: Started docker.service - Docker Application Container Engine.
Aug 26 17:59:43 ip-172-31-37-139.ec2.internal dockerd[26479]: time="2023-08-26T17:59:43.161422834Z" level=info msg="API listen on /run/docker.sock"
lines 1-24/24 (END)
```

i-0f6c2552ce5d9afca (Practical 09)
PublicIPs: 3.87.117.205 PrivateIPs: 172.31.37.139

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 08 : after install and start docker service check version using docker version command

Snapshot :



```
aws Services Search [Alt+S] N. Virginia v vodlabs/user2661525=Tej_Dekiwadiya @ 4714-1876-8398
[root@ip-172-31-37-139 ec2-user]# docker version
Client:
 Version: 20.10.25
 API version: 1.41
 Go version: go1.19.9
 Git commit: b82b9f3
 Built: Wed Jul 12 19:37:13 2023
 OS/Arch: linux/amd64
 Context: default
 Experimental: true

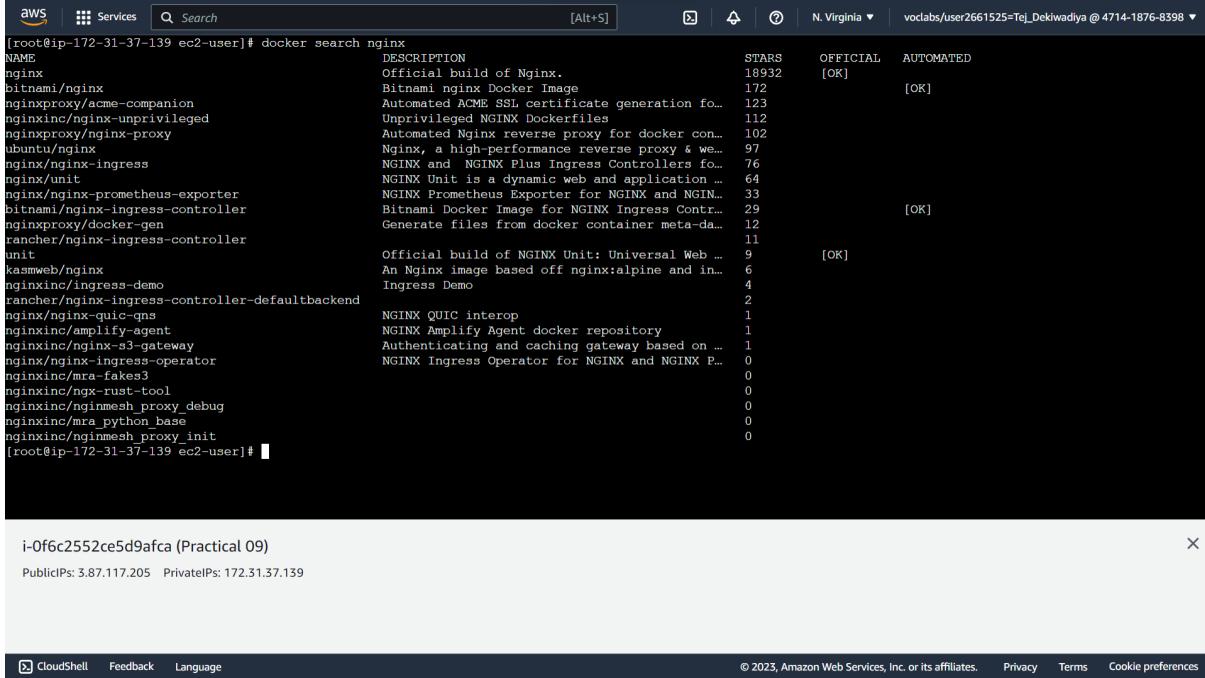
Server:
 Engine:
 Version: 20.10.25
 API version: 1.41 (minimum version 1.12)
 Go version: go1.19.9
 Git commit: 5df983c
 Built: Wed Jul 5 00:00:00 2023
 OS/Arch: linux/amd64
 Experimental: false
 containerd:
 Version: 1.7.2
 GitCommit: 0cae528dd6cb557f7201036e9f43420650207b58
 runc:
 Version: 1.1.7
 GitCommit: f19387a6bec4944c770f7668ab51c4348d9c2f38
 docker-init:
 Version: 0.19.0
 GitCommit: de40ad0
[root@ip-172-31-37-139 ec2-user]# 
```

i-0f6c2552ce5d9afca (Practical 09)
PublicIPs: 3.87.117.205 PrivateIPs: 172.31.37.139

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 09 : run docker search nginx

Snapshot :



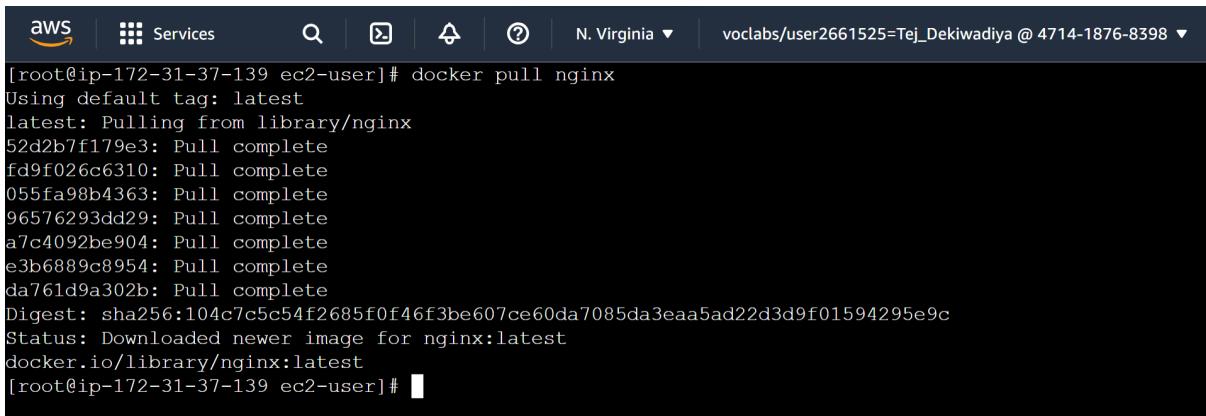
```
[root@ip-172-31-37-139 ec2-user]# docker search nginx
NAME                                            DESCRIPTION                                     STARS      OFFICIAL   AUTOMATED
nginx                                           Official build of Nginx.                           18932     [OK]
bitnami/nginx                                         Bitnami Nginx Docker Image                         172       [OK]
nginxproxy/acme-companion                         Automated ACME SSL certificate generation fo...    123
nginxinc/nginx-unprivileged                      Unprivileged NGINX Dockerfiles                   112
nginxproxy/nginx-proxy                            Automated Nginx reverse proxy for docker con...    102
ubuntu/nginx                                         Nginx, a high-performance reverse proxy & we...    97
nginx/nginx-ingress                               NGINX and NGINX Plus Ingress Controllers fo...    76
nginx/unit                                         NGINX Unit is a dynamic web and application ...    64
nginx/nginx-prometheus-exporter                  NGINX Prometheus Exporter for NGINX and NGIN...    33
bitnami/nginx-ingress-controller                 Bitnami Docker Image for NGINX Ingress Contr...    29
nginxproxy/docker-gen                            Generate files from docker container meta-da...    12
rancher/nginx-ingress-controller                 11
unit                                              Official build of NGINX Unit: Universal Web ...    9       [OK]
kasmweb/nginx                                       An Nginx image based off nginx:alpine and in...    6
rancher/nginx-ingress-demo                       Ingress Demo                                      4
rancher/nginx-ingress-controller-defaultbackend  2
nginx/nginx-quic-qns                            NGINX QUIC interop                                1
nginxinc/amplify-agent                          NGINX Amplify Agent docker repository           1
nginxinc/nginx-s3-gateway                      Authenticating and caching gateway based on ...    1
nginx/nginx-ingress-operator                     NGINX Ingress Operator for NGINX and NGINX P...    0
nginxinc/mra-fakes3                            0
nginxinc/nginx-rust-tool                        0
nginxinc/nginmash_proxy_debug                  0
nginxinc/mra_python_base                       0
nginxinc/nginmash_proxy_init                   0
[root@ip-172-31-37-139 ec2-user]#
```

i-0f6c2552ce5d9afca (Practical 09) X
 Public IPs: 3.87.117.205 Private IPs: 172.31.37.139

[CloudShell](#) [Feedback](#) [Language](#) © 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Step 10 : run docker pull nginx

Snapshot :



```
[root@ip-172-31-37-139 ec2-user]# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
52d2b7f179e3: Pull complete
fd9f026c6310: Pull complete
055fa98b4363: Pull complete
96576293dd29: Pull complete
a7c4092be904: Pull complete
e3b6889c8954: Pull complete
da761d9a302b: Pull complete
Digest: sha256:104c7c5c54f2685f0f46f3be607ce60da7085da3eaa5ad22d3d9f01594295e9c
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
[root@ip-172-31-37-139 ec2-user]#
```

i-0f6c2552ce5d9afca (Practical 09)

Public IPs: 3.87.117.205 Private IPs: 172.31.37.139

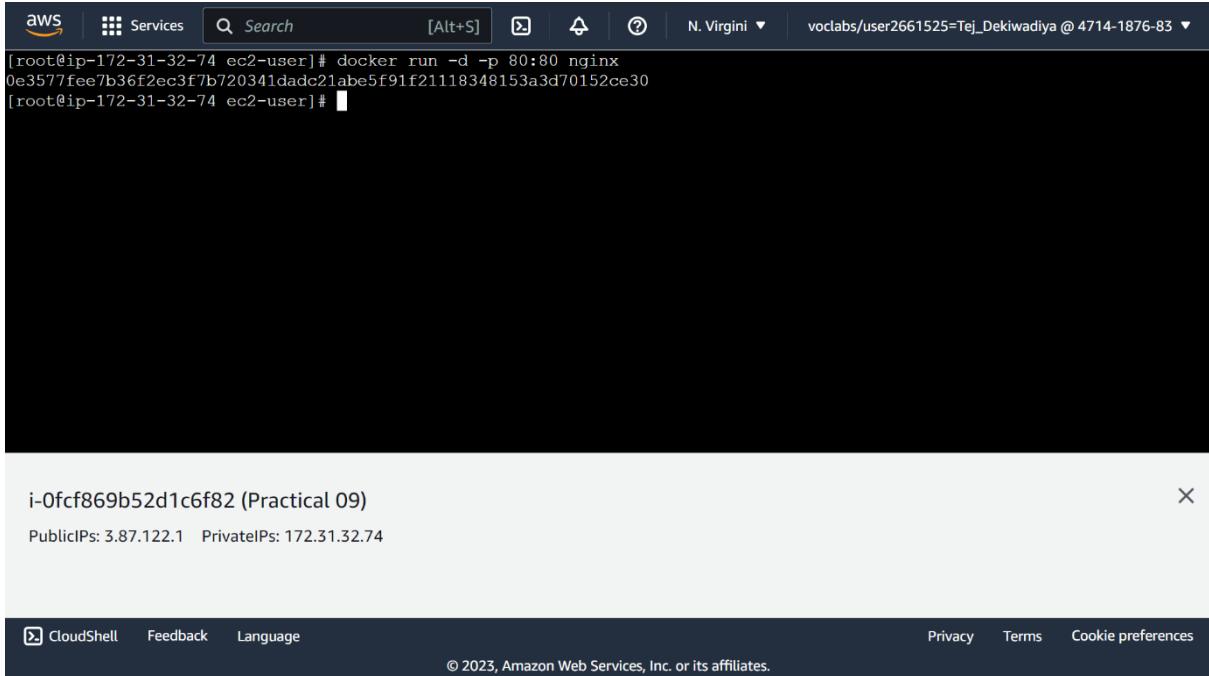
[CloudShell](#) [Feedback](#) [Language](#)

[Privacy](#) [Terms](#) [Cookie preferences](#)

© 2023, Amazon Web Services, Inc. or its affiliates.

Step 11: Run docker run -d -p 80:80 nginx

Snapshot :



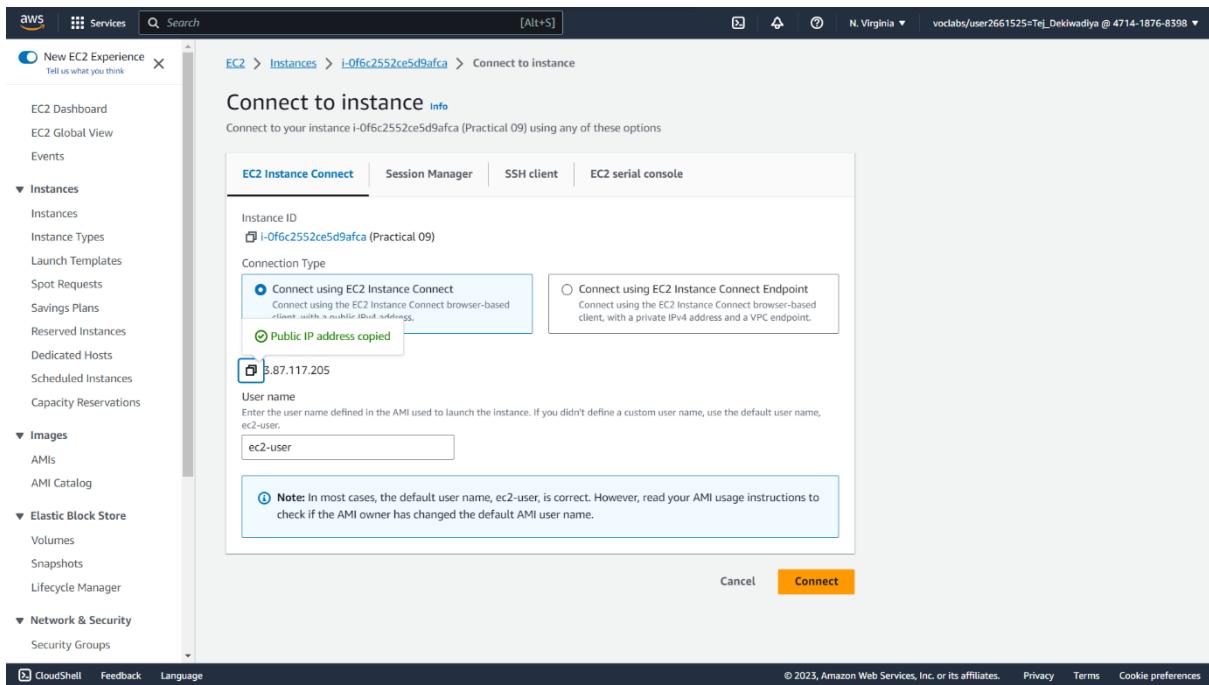
```
(root@ip-172-31-32-74 ec2-user]# docker run -d -p 80:80 nginx
0e3577fee7b36f2ec3f7b720341dadc21abe5f91f21118348153a3d70152ce30
[root@ip-172-31-32-74 ec2-user]#
```

i-0fcf869b52d1c6f82 (Practical 09) X
 Public IPs: 3.87.122.1 Private IPs: 172.31.32.74

CloudShell Feedback Language Privacy Terms Cookie preferences
 © 2023, Amazon Web Services, Inc. or its affiliates.

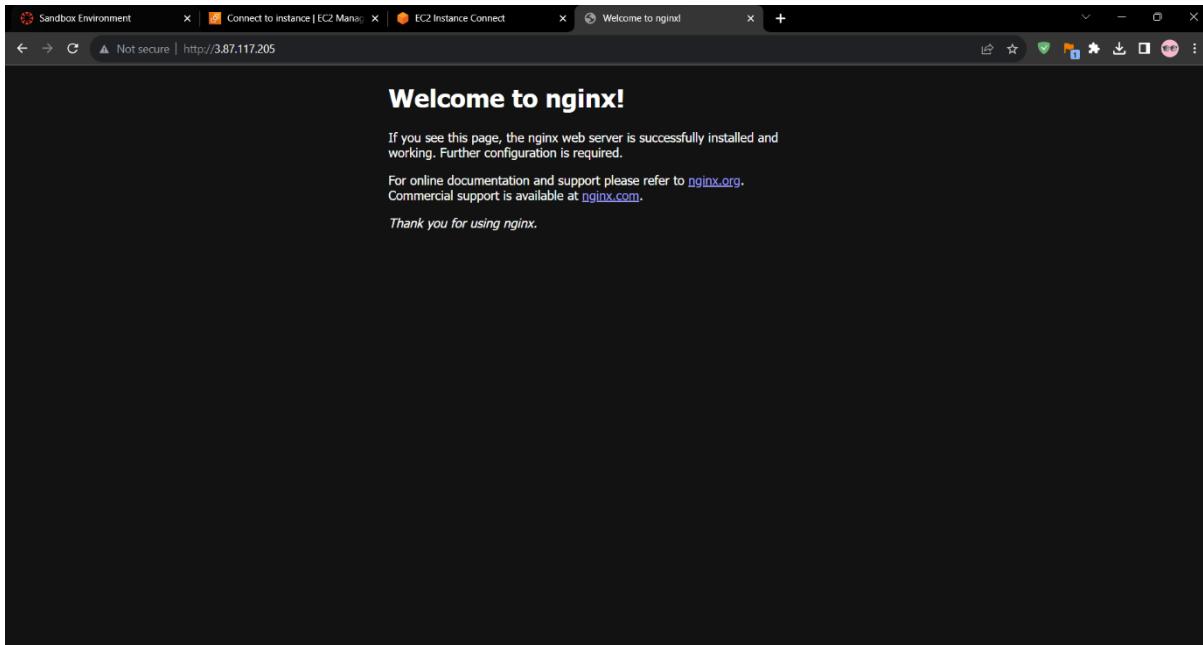
Step 12 : Copy Public IP Address In Connect to instance section

Snapshot :



Step 13 : You can see server has been deployed

Snapshot :



Step 14 : then run docker ps

Snapshot :

```
[root@ip-172-31-37-139 ec2-user]# docker ps
CONTAINER ID        IMAGE       COMMAND             CREATED            STATUS              PORTS
 NAMES
7815cdef7fff      nginx      "/docker-entrypoint...."   5 minutes ago    Up 5 minutes     0.0.0.0:80->80/tcp
, :::80->80/tcp   youthful_curran
[root@ip-172-31-37-139 ec2-user]#
```

i-0f6c2552ce5d9afca (Practical 09) X

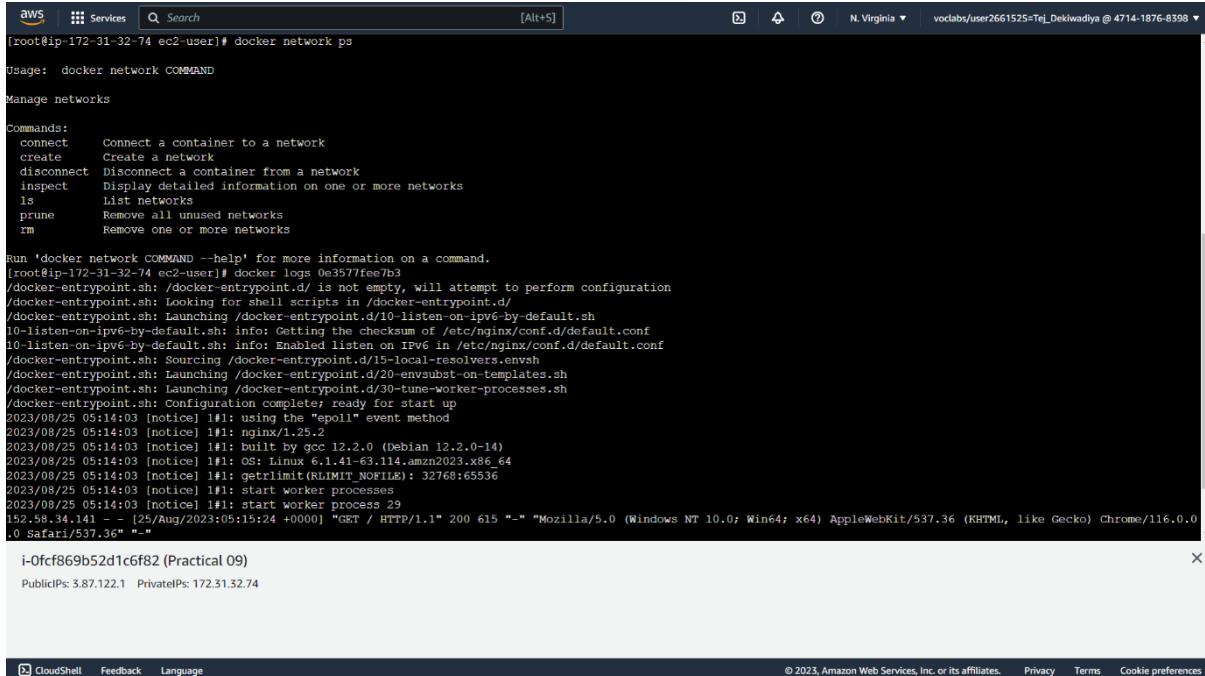
Public IPs: 3.87.117.205 Private IPs: 172.31.37.139

CloudShell Feedback Language Privacy Terms Cookie preferences

© 2023, Amazon Web Services, Inc. or its affiliates.

Step 15 : then run docker network ps

Snapshot :



```
[root@ip-172-31-32-74 ec2-user]# docker network ps
Usage: docker network COMMAND

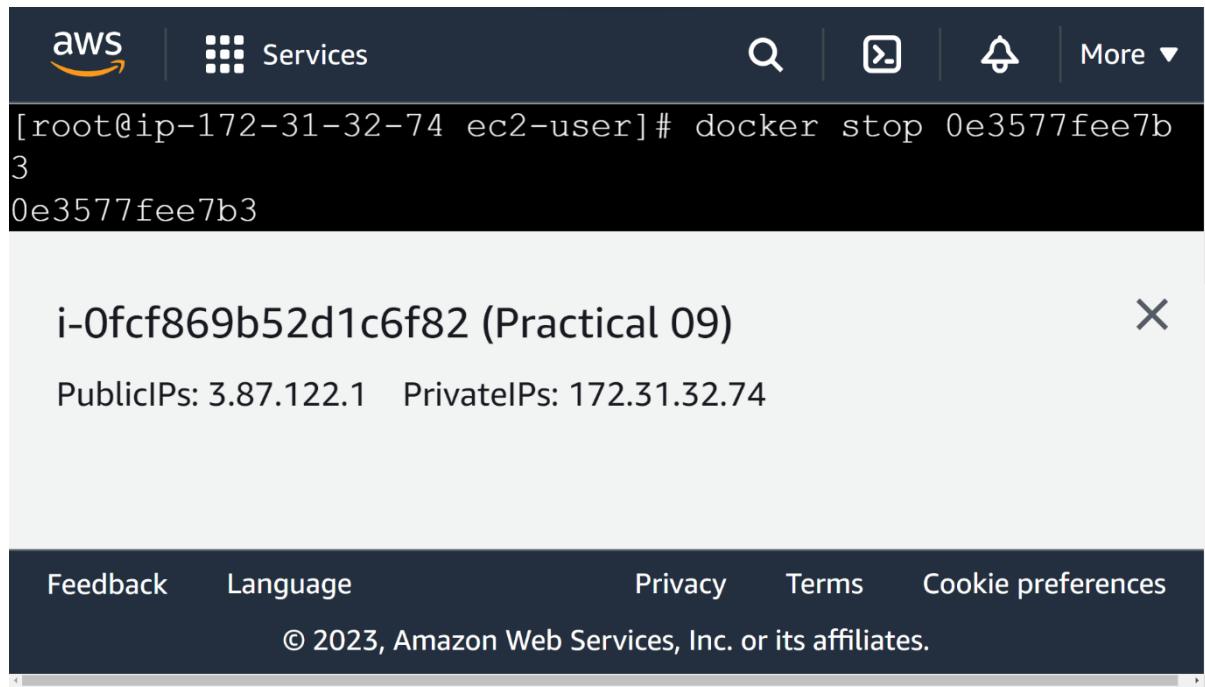
Manage networks

Commands:
connect      Connect a container to a network
create       Create a network
disconnect   Disconnect a container from a network
inspect     Display detailed information on one or more networks
ls          List networks
prune       Remove all unused networks
rm          Remove one or more networks

Run 'docker network COMMAND --help' for more information on a command.
[root@ip-172-31-32-74 ec2-user]# docker logs 0e3577fee7b3
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/08/25 05:14:03 [notice] 1#1: using the "epoll" event method
2023/08/25 05:14:03 [notice] 1#1: nginx/1.25.2
2023/08/25 05:14:03 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2023/08/25 05:14:03 [notice] 1#1: OS: Linux 6.1.41-63.114.amzn2023.x86_64
2023/08/25 05:14:03 [notice] 1#1: getrlimit (RLIMIT_NOFILE): 32768:65536
2023/08/25 05:14:03 [notice] 1#1: start worker processes
2023/08/25 05:14:03 [notice] 1#1: start worker process 29
152.58.34.141 - [25/Aug/2023:05:15:24 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36" "-"
i-0fcf869b52d1c6f82 (Practical 09)
PublicIPs: 3.87.122.1 PrivateIPs: 172.31.32.74
```

Step 16 : run docker stop <Container ID> to stop docker service

Snapshot :



```
[root@ip-172-31-32-74 ec2-user]# docker stop 0e3577fee7b3
0e3577fee7b3
```

i-0fcf869b52d1c6f82 (Practical 09)

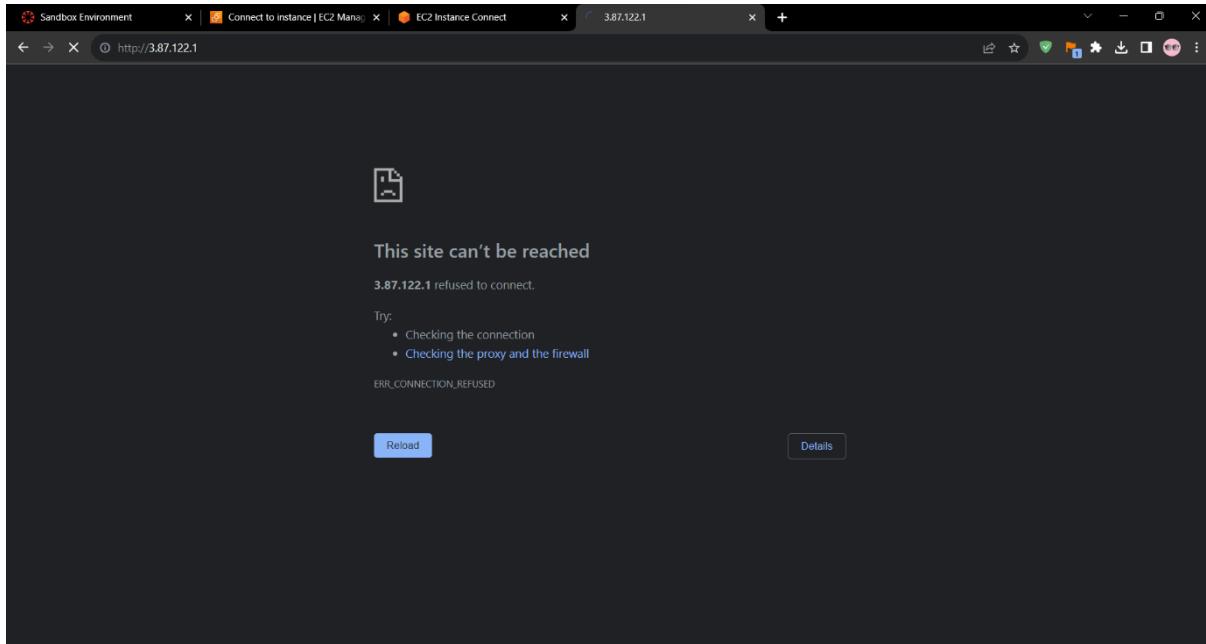
PublicIPs: 3.87.122.1 PrivateIPs: 172.31.32.74

Feedback Language Privacy Terms Cookie preferences

© 2023, Amazon Web Services, Inc. or its affiliates.

Step 17 : Refresh Public IP Tab and you can see no response from server

Snapshot :



Step 18 : than run docker push nginx, docker ps

Snapshot :

```
//3.87.122.1/
[root@ip-172-31-32-74 ec2-user]# docker stop 0e3577fee7b3
0e3577fee7b3
[root@ip-172-31-32-74 ec2-user]# docker push nginx
Using default tag: latest
The push refers to repository [docker.io/library/nginx]
563c64030925: Layer already exists
6fb960878295: Layer already exists
e161c3f476b5: Layer already exists
8a7e12012e6f: Layer already exists
d0a62f56ef41: Layer already exists
4713cb24eef9: Layer already exists
511780f88f80: Layer already exists
errors:
denied: requested access to the resource is denied
unauthorized: authentication required

[root@ip-172-31-32-74 ec2-user]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[root@ip-172-31-32-74 ec2-user]# []

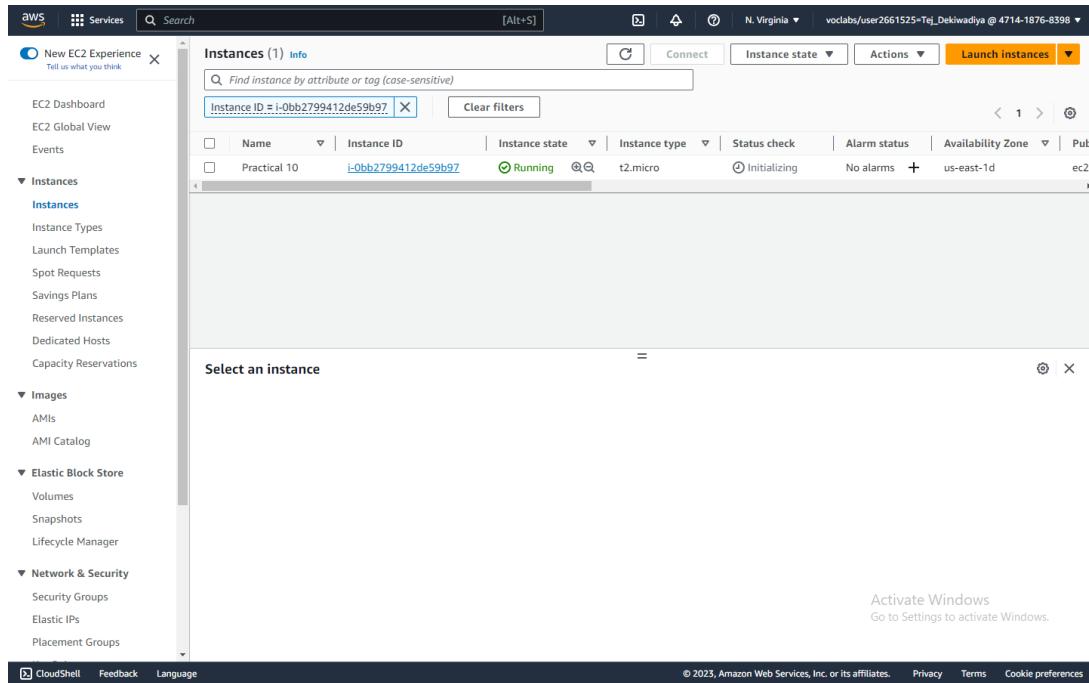
i-0fcf869b52d1c6f82 (Practical 09) X
PublicIPs: 3.87.122.1 PrivateIPs: 172.31.32.74

CloudShell Feedback Language Privacy Terms Cookie preferences
© 2023, Amazon Web Services, Inc. or its affiliates.
```

Practical 10 : Deploy Docker Technology

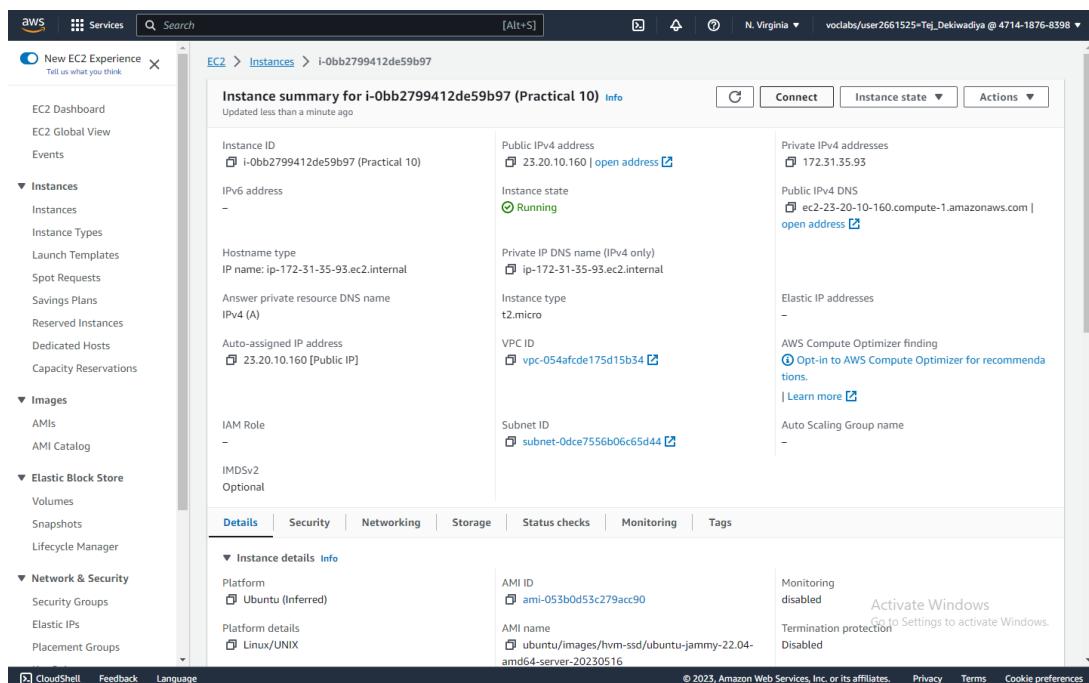
Step 01 : Create an Instance in EC2

Snapshot :



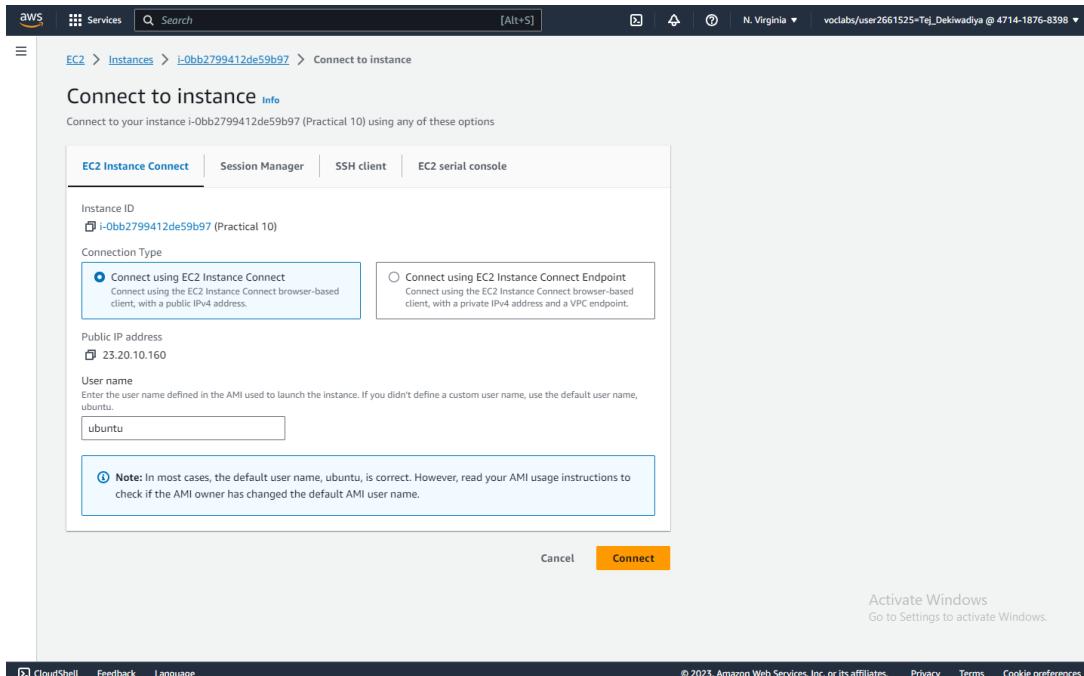
Step 02 : Click on instance ID and click Connect

Snapshot :



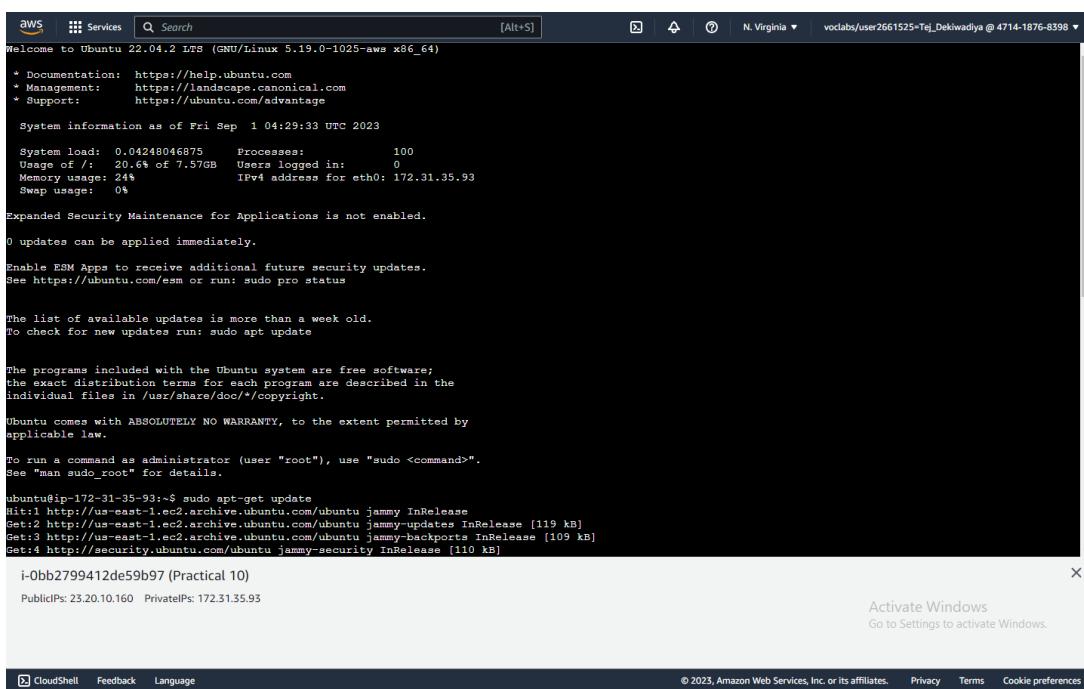
Step 03 : Connect the Instance

Snapshot :



Step 04 : Run command - sudo apt-get update

Snapshot :





```

AWS Services Search [Alt+S] N. Virginia v vodlabs/user2661525+Tej_Dekiwadiya @ 4714-1876-8398 ▾
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [724 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [931 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [103 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 C-n-f Metadata [15.6 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [787 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [126 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [536 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [977 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [213 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [21.7 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [41.6 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [9768 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [476 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [40.8 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [10.2 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [22.3 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [15.4 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [576 B]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [160 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.2 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [771 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [106 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [536 B]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [778 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [142 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.6 kB]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7060 B]
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 26.9 MB in 5s (5420 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-35-93:~$
```

i-Obb2799412de59b97 (Practical 10)
PublicIPs: 23.20.10.160 PrivateIPs: 172.31.35.95

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 05: Run - sudo apt-get install docker.io

Snapshot :

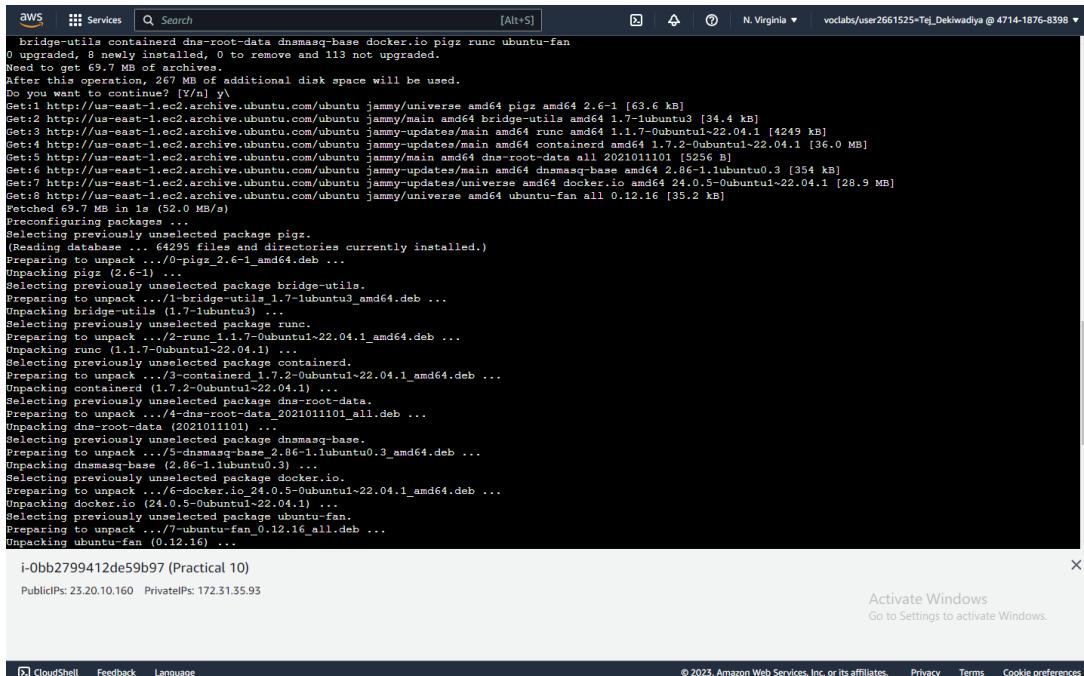
```

AWS Services Search [Alt+S] N. Virginia v vodlabs/user2661525+Tej_Dekiwadiya @ 4714-1876-8398 ▾
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [213 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [21.7 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [41.6 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [9768 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [476 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [40.8 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [10.2 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [22.3 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [15.4 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [576 B]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [160 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.2 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [771 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [106 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [536 B]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [778 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [142 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.6 kB]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7060 B]
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 26.9 MB in 5s (5420 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-35-93:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
bridge-utils contained dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
bridge-util contained dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 113 not upgraded.
Need to get 69.7 MB of archives.
After this operation, 267 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

i-Obb2799412de59b97 (Practical 10)
PublicIPs: 23.20.10.160 PrivateIPs: 172.31.35.95

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



```

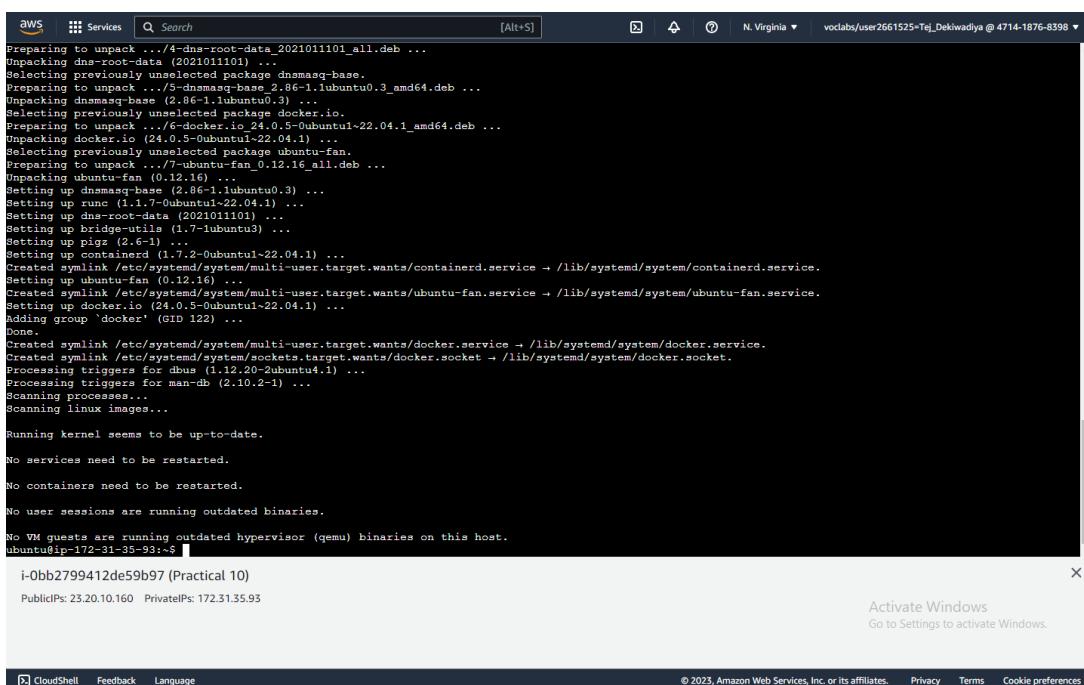
AWS Services Search [Alt+S] N. Virginia v vclabs/user2661525-Tej_Dekiwadiya @ 4714-1876-8398
bridge-utils contained in dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 113 not upgraded.
Need to get 69.7 MB of archives.
After this operation, 267 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu0.3 [34.4 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 runc amd64 1.1.7-0ubuntu1-22.04.1 [4249 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.2-0ubuntu1-22.04.1 [36.0 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 dns-root-data all 2021011101 [5256 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dnsmasq-base amd64 2.86-1.1ubuntu0.3 [354 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io amd64 24.0.5-0ubuntu1-22.04.1 [28.9 MB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 69.7 MB in 1s (52.0 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 64295 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7-1ubuntu0.3_amd64.deb ...
Unpacking bridge-utils (1.7-1ubuntu0.3) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.1.7-0ubuntu1-22.04.1_amd64.deb ...
Unpacking runc (1.1.7-0ubuntu1-22.04.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.7.2-0ubuntu1-22.04.1_amd64.deb ...
Unpacking containerd (1.7.2-0ubuntu1-22.04.1) ...
Selecting previously unselected package dns-root-data.
Preparing to unpack .../4-dns-root-data_2021011101_all.deb ...
Unpacking dns-root-data (2021011101) ...
Selecting previously unselected package dnsmasq-base.
Preparing to unpack .../5-dnsmasq-base_2.86-1.1ubuntu0.3_amd64.deb ...
Unpacking dnsmasq-base (2.86-1.1ubuntu0.3) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../6-docker.io_24.0.5-0ubuntu1-22.04.1_amd64.deb ...
Unpacking docker.io (24.0.5-0ubuntu1-22.04.1) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../7-ubuntu-fan_0.12.16_all.deb ...
Unpacking ubuntu-fan (0.12.16) ...
i-0bb2799412de59b97 (Practical 10)

PublicIPs: 23.20.10.160 PrivateIPs: 172.31.35.95

```

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



```

AWS Services Search [Alt+S] N. Virginia v vclabs/user2661525-Tej_Dekiwadiya @ 4714-1876-8398
Preparing to unpack .../4-dns-root-data_2021011101_all.deb ...
Unpacking dns-root-data (2021011101) ...
Selecting previously unselected package dnsmasq-base.
Preparing to unpack .../5-dnsmasq-base_2.86-1.1ubuntu0.3_amd64.deb ...
Unpacking dnsmasq-base (2.86-1.1ubuntu0.3) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../6-docker.io_24.0.5-0ubuntu1-22.04.1_amd64.deb ...
Unpacking docker.io (24.0.5-0ubuntu1-22.04.1) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../7-ubuntu-fan_0.12.16_all.deb ...
Unpacking ubuntu-fan (0.12.16) ...
Setting up dnsmasq-base (2.86-1.1ubuntu0.3) ...
Setting up runc (1.1.7-0ubuntu1-22.04.1) ...
Setting up dns-root-data (2021011101) ...
Setting up bridge-utils (1.7-1ubuntu0.3) ...
Setting up pigz (2.6-1) ...
Setting up containerd (1.7.2-0ubuntu1-22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (24.0.5-0ubuntu1-22.04.1) ...
Adding group 'docker' (GID 122) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Processing triggers for man-db (2.10.2-1) ...
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-172-31-35-93:~$ i-0bb2799412de59b97 (Practical 10)

PublicIPs: 23.20.10.160 PrivateIPs: 172.31.35.95

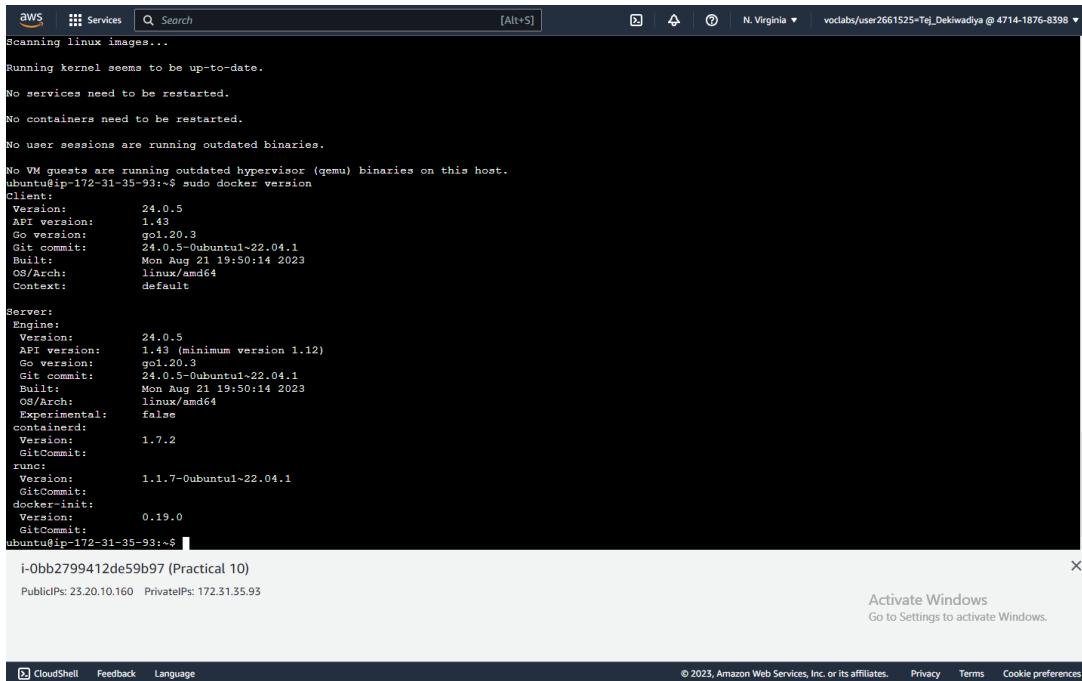
```

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 06 : Run command - sudo docker version.

Snapshot :



```
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-172-31-35-93:~$ sudo docker version
Client:
  Version:          24.0.5
  API version:     1.43
  Go version:      go1.20.3
  Git commit:      24.0.5-0ubuntu1-22.04.1
  Built:           Mon Aug 21 19:50:14 2023
  OS/Arch:         linux/amd64
  Context:         default

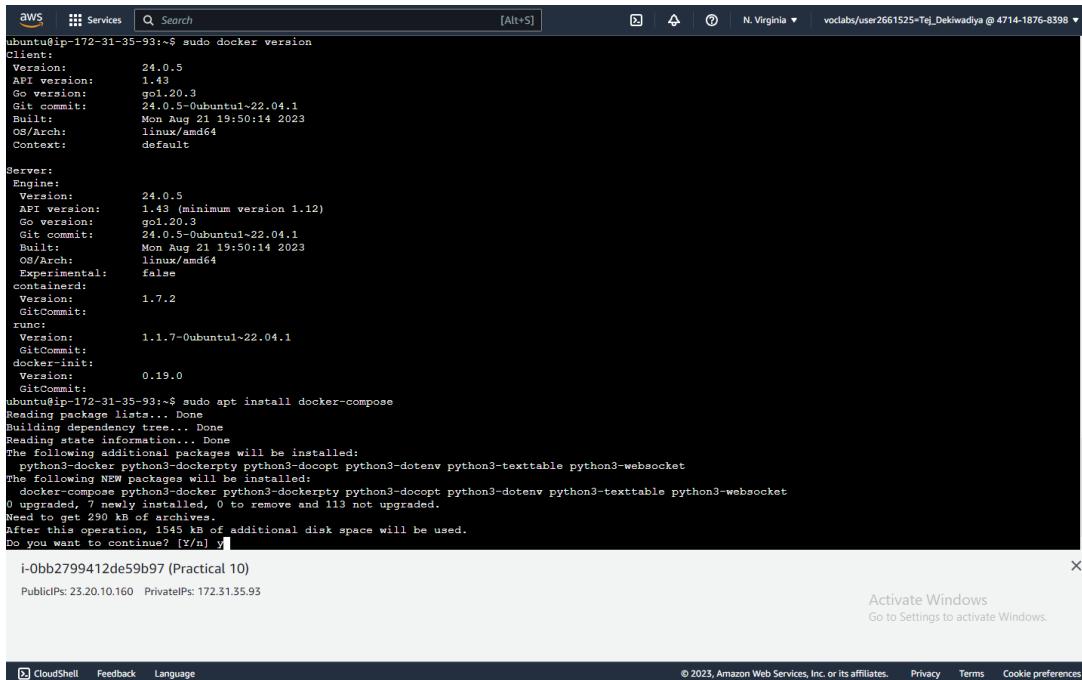
Server:
  Engine:
    Version:          24.0.5
    API version:     1.43 (minimum version 1.12)
    Go version:      go1.20.3
    Git commit:      24.0.5-0ubuntu1-22.04.1
    Built:           Mon Aug 21 19:50:14 2023
    OS/Arch:         linux/amd64
    Experimental:   false
  containerd:
    Version:          1.7.2
    GitCommit:        1.1.7-0ubuntu1-22.04.1
    Version:          0.19.0
    GitCommit:        0.19.0
ubuntu@ip-172-31-35-93:~$ i-0bb2799412de59b97 (Practical 10)
PublicIPs: 23.20.10.160 PrivateIPs: 172.31.35.93
```

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 07 : Run command - sudo apt install docker-compose

Snapshot :

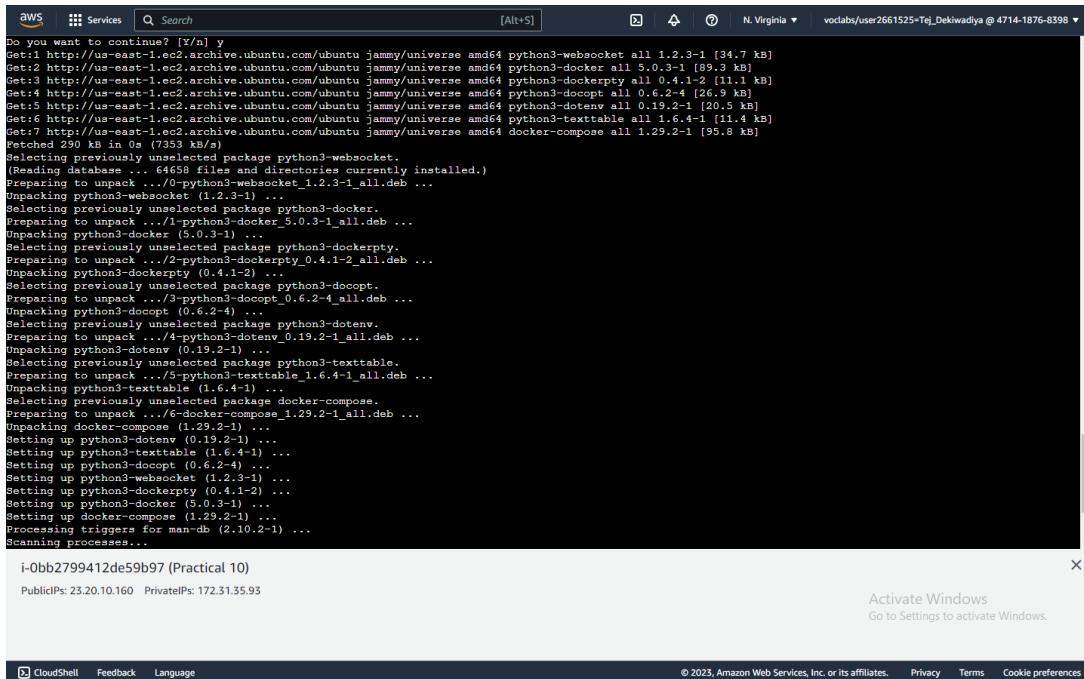


```
ubuntu@ip-172-31-35-93:~$ sudo docker version
Client:
  Version:          24.0.5
  API version:     1.43
  Go version:      go1.20.3
  Git commit:      24.0.5-0ubuntu1-22.04.1
  Built:           Mon Aug 21 19:50:14 2023
  OS/Arch:         linux/amd64
  Context:         default

Server:
  Engine:
    Version:          24.0.5
    API version:     1.43 (minimum version 1.12)
    Go version:      go1.20.3
    Git commit:      24.0.5-0ubuntu1-22.04.1
    Built:           Mon Aug 21 19:50:14 2023
    OS/Arch:         linux/amd64
    Experimental:   false
  containerd:
    Version:          1.7.2
    GitCommit:        1.1.7-0ubuntu1-22.04.1
    Version:          0.19.0
    GitCommit:        0.19.0
ubuntu@ip-172-31-35-93:~$ sudo apt install docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libpython3.8-dev libpython3.8-stdlib python3-distro python3-distro-locale
  python3-docker python3-dockerpty python3-dockerpy python3-dockerpy-pty
  python3-dockerpy-distro python3-dockerpy-distro-locale python3-dockerpy-pty
  python3-distro python3-distro-locale python3-distro-pty
The following NEW packages will be installed:
  docker-compose python3-docker python3-dockerpty python3-docopt python3-dotenv
  python3-texttable python3-websocket
0 upgraded, 7 newly installed, 0 to remove and 113 not upgraded.
Need to get 290 kB of additional archives.
After this operation, 1545 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
i-0bb2799412de59b97 (Practical 10)
PublicIPs: 23.20.10.160 PrivateIPs: 172.31.35.93
```

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

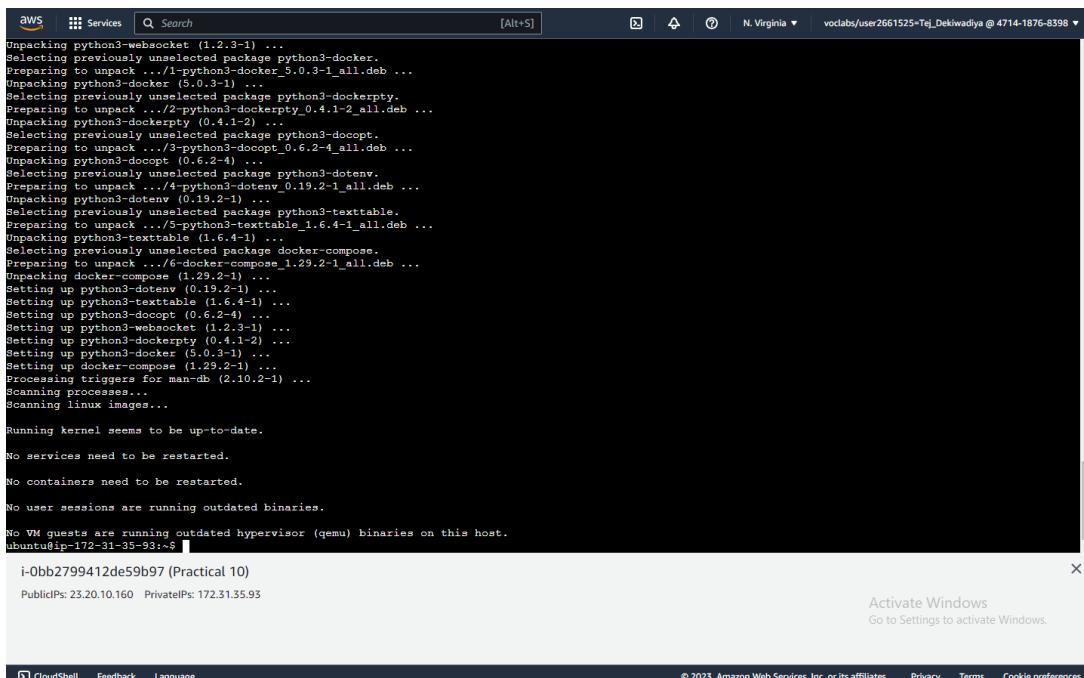


```

AWS Services Search [Alt+S] N. Virginia v vocabs/user2661525-Tej_Dekiwadiya @ 4714-1876-8398 ▾
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-websocket all 1.2.3-1 [34.7 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-docker all 5.0.3-1 [89.3 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-dockerpty all 0.4.1-2 [11.1 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-dcopt all 0.6.2-4 [26.9 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-dotenv all 0.19.2-1 [20.5 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-texttable all 1.6.4-1 [11.4 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 docker-compose all 1.29.2-1 [95.8 kB]
Patched 290 kB in 0s (7353 kB/s)
Selecting previously unselected package python3-websocket.
(Reading database ... 64658 files and directories currently installed.)
Preparing to unpack .../0-python3-websocket_1.2.3-1_all.deb ...
Unpacking python3-websocket (1.2.3-1) ...
Selecting previously unselected package python3-docker.
Preparing to unpack .../1-python3-docker_5.0.3-1_all.deb ...
Unpacking python3-docker (5.0.3-1) ...
Selecting previously unselected package python3-dockerpty.
Preparing to unpack .../2-python3-dockerpty_0.4.1-2_all.deb ...
Unpacking python3-dockerpty (0.4.1-2) ...
Selecting previously unselected package python3-dcopt.
Preparing to unpack .../3-python3-dcopt_0.6.2-4_all.deb ...
Unpacking python3-dcopt (0.6.2-4) ...
Selecting previously unselected package python3-dotenv.
Preparing to unpack .../4-python3-dotenv_0.19.2-1_all.deb ...
Unpacking python3-dotenv (0.19.2-1) ...
Selecting previously unselected package python3-texttable.
Preparing to unpack .../5-python3-texttable_1.6.4-1_all.deb ...
Unpacking python3-texttable (1.6.4-1) ...
Selecting previously unselected package docker-compose.
Preparing to unpack .../6-docker-compose_1.29.2-1_all.deb ...
Unpacking docker-compose (1.29.2-1) ...
Setting up python3-dotenv (0.19.2-1) ...
Setting up python3-texttable (1.6.4-1) ...
Setting up python3-dcopt (0.6.2-4) ...
Setting up python3-websocket (1.2.3-1) ...
Setting up python3-dockerpty (0.4.1-2) ...
Setting up python3-docker (5.0.3-1) ...
Setting up docker-compose (1.29.2-1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
i-0bb2799412de59b97 (Practical 10)
PublicIPs: 23.20.10.160 PrivateIPs: 172.31.35.95

```

Activate Windows
Go to Settings to activate Windows.



```

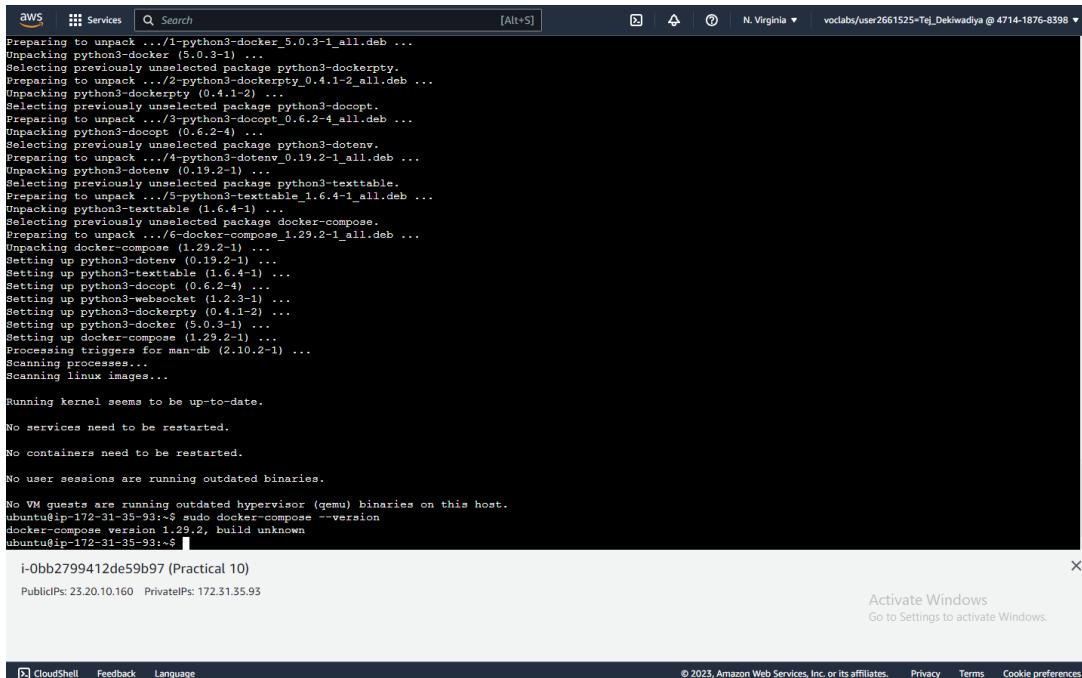
AWS Services Search [Alt+S] N. Virginia v vocabs/user2661525-Tej_Dekiwadiya @ 4714-1876-8398 ▾
Unpacking python3-websocket (1.2.3-1) ...
Selecting previously unselected package python3-docker.
Preparing to unpack .../1-python3-docker_5.0.3-1_all.deb ...
Unpacking python3-docker (5.0.3-1) ...
Selecting previously unselected package python3-dockerpty.
Preparing to unpack .../2-python3-dockerpty_0.4.1-2_all.deb ...
Unpacking python3-dockerpty (0.4.1-2) ...
Selecting previously unselected package python3-dcopt.
Preparing to unpack .../3-python3-dcopt_0.6.2-4_all.deb ...
Unpacking python3-dcopt (0.6.2-4) ...
Selecting previously unselected package python3-dotenv.
Preparing to unpack .../4-python3-dotenv_0.19.2-1_all.deb ...
Unpacking python3-dotenv (0.19.2-1) ...
Selecting previously unselected package python3-texttable.
Preparing to unpack .../5-python3-texttable_1.6.4-1_all.deb ...
Unpacking python3-texttable (1.6.4-1) ...
Selecting previously unselected package docker-compose.
Preparing to unpack .../6-docker-compose_1.29.2-1_all.deb ...
Unpacking docker-compose (1.29.2-1) ...
Setting up python3-dotenv (0.19.2-1) ...
Setting up python3-texttable (1.6.4-1) ...
Setting up python3-dcopt (0.6.2-4) ...
Setting up python3-websocket (1.2.3-1) ...
Setting up python3-dockerpty (0.4.1-2) ...
Setting up python3-docker (5.0.3-1) ...
Setting up docker-compose (1.29.2-1) ...
Processing triggers for man-db (2.10.2-1) ...
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-172-31-35-93:~$ 
i-0bb2799412de59b97 (Practical 10)
PublicIPs: 23.20.10.160 PrivateIPs: 172.31.35.95

```

Activate Windows
Go to Settings to activate Windows.

Step 08 : Run command – sudo docker-compose –version.

Snapshot :



```

AWS Services Search [Alt+S] N. Virginia v vocabs/user2661525+Tej_Dekiwadiya @ 4714-1876-8398
Preparing to unpack .../1/python3-docker_5.0.3-1_all.deb ...
Unpacking python3-docker (5.0.3-1) ...
Selecting previously unselected package python3-dockerpty.
Preparing to unpack .../2/python3-dockerpty_0.4.1-2_all.deb ...
Unpacking python3-dockerpty (0.4.1-2) ...
Selecting previously unselected package python3-docopt.
Preparing to unpack .../3/python3-docopt_0.6.2-4_all.deb ...
Unpacking python3-docopt (0.6.2-4) ...
Selecting previously unselected package python3-dotenv.
Preparing to unpack .../4/python3-dotenv_0.19.2-1_all.deb ...
Unpacking python3-dotenv (0.19.2-1) ...
Selecting previously unselected package python3-texttable.
Preparing to unpack .../5/python3-texttable_1.6.4-1_all.deb ...
Unpacking python3-texttable (1.6.4-1) ...
Selecting previously unselected package docker-compose.
Preparing to unpack .../6-docker-compose_1.29.2-1_all.deb ...
Unpacking docker-compose (1.29.2-1) ...
Setting up python3-dotenv (0.19.2-1) ...
Setting up python3-texttable (1.6.4-1) ...
Setting up python3-docopt (0.6.2-4) ...
Setting up python3-websocket (1.2.3-1) ...
Setting up python3-dockerpty (0.4.1-2) ...
Setting up python3-docker (5.0.3-1) ...
Setting up docker-compose (1.29.2-1) ...
Processing triggers for man-db (2.10.2-1) ...
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-172-31-35-93:~$ sudo docker-compose --version
docker-compose version 1.29.2, build unknown
ubuntu@ip-172-31-35-93:~$ i-0bb2799412de59b97 (Practical 10)
PublicIPs: 23.20.10.160 PrivateIPs: 172.31.35.93

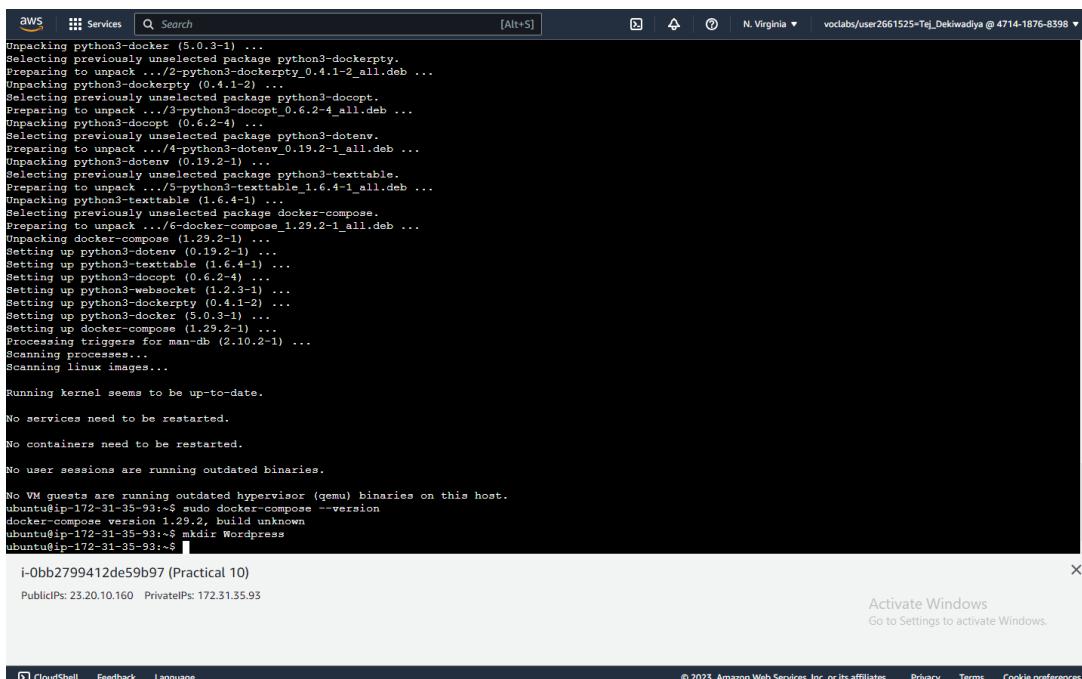
```

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 09 : Run command – mkdir WordPress

Snapshot :



```

AWS Services Search [Alt+S] N. Virginia v vocabs/user2661525+Tej_Dekiwadiya @ 4714-1876-8398
Preparing python3-docker (5.0.3-1) ...
Selecting previously unselected package python3-dockerpty.
Preparing to unpack .../2/python3-dockerpty_0.4.1-2_all.deb ...
Unpacking python3-dockerpty (0.4.1-2) ...
Selecting previously unselected package python3-docopt.
Preparing to unpack .../3/python3-docopt_0.6.2-4_all.deb ...
Unpacking python3-docopt (0.6.2-4) ...
Selecting previously unselected package python3-dotenv.
Preparing to unpack .../4/python3-dotenv_0.19.2-1_all.deb ...
Unpacking python3-dotenv (0.19.2-1) ...
Selecting previously unselected package python3-texttable.
Preparing to unpack .../5/python3-texttable_1.6.4-1_all.deb ...
Unpacking python3-texttable (1.6.4-1) ...
Selecting previously unselected package docker-compose.
Preparing to unpack .../6-docker-compose_1.29.2-1_all.deb ...
Unpacking docker-compose (1.29.2-1) ...
Setting up python3-dotenv (0.19.2-1) ...
Setting up python3-texttable (1.6.4-1) ...
Setting up python3-docopt (0.6.2-4) ...
Setting up python3-websocket (1.2.3-1) ...
Setting up python3-dockerpty (0.4.1-2) ...
Setting up python3-docker (5.0.3-1) ...
Setting up docker-compose (1.29.2-1) ...
Processing triggers for man-db (2.10.2-1) ...
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-172-31-35-93:~$ sudo docker-compose --version
docker-compose version 1.29.2, build unknown
ubuntu@ip-172-31-35-93:~$ mkdir Wordpress
ubuntu@ip-172-31-35-93:~$ i-0bb2799412de59b97 (Practical 10)
PublicIPs: 23.20.10.160 PrivateIPs: 172.31.35.93

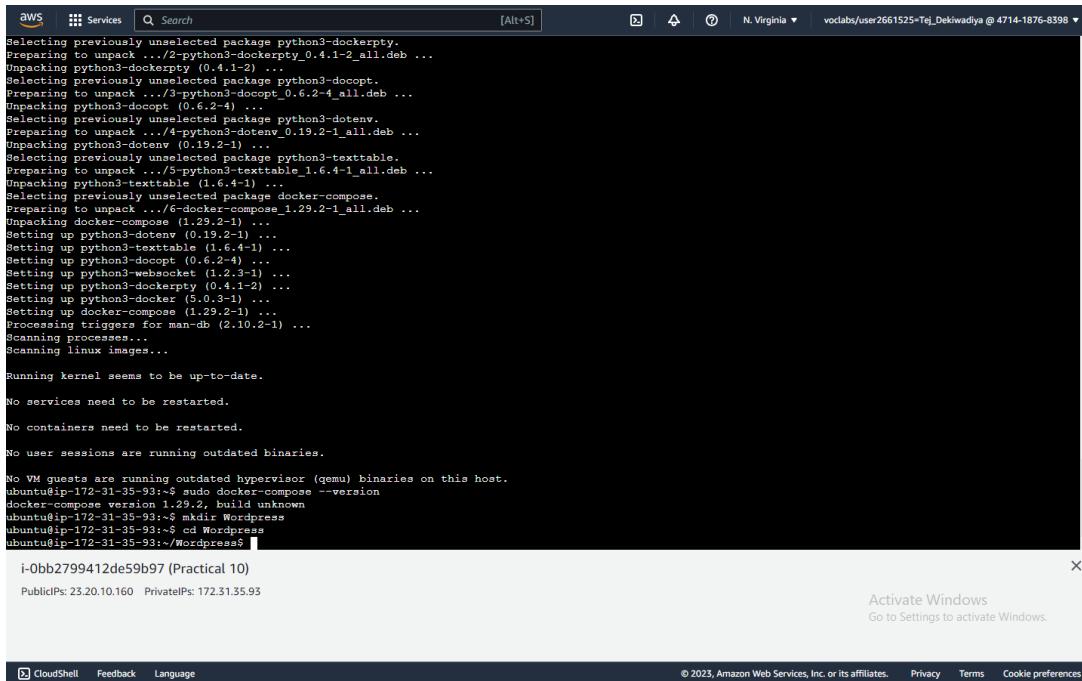
```

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 10 : Run command – cd WordPress

Snapshot :



```

AWS Services Search [Alt+S] N. Virginia v vclabs/user2661525-Tej_Dekiwadiya @ 4714-1876-8398
Selecting previously unselected package python3-dockerpty.
Preparing to unpack .../2-python3-dockerpty_0.4.1-2_all.deb ...
Unpacking python3-dockerpty (0.4.1-2) ...
Selecting previously unselected package python3-docopt.
Preparing to unpack .../3-python3-docopt_0.6.2-4_all.deb ...
Unpacking python3-docopt (0.6.2-4) ...
Selecting previously unselected package python3-dotenv.
Preparing to unpack .../4-python3-dotenv_0.19.2-1_all.deb ...
Unpacking python3-dotenv (0.19.2-1) ...
Selecting previously unselected package python3-texttable.
Preparing to unpack .../5-python3-texttable_1.6.4-1_all.deb ...
Unpacking python3-texttable (1.6.4-1) ...
Selecting previously unselected package docker-compose.
Preparing to unpack .../6-docker-compose_1.29.2-1_all.deb ...
Unpacking docker-compose (1.29.2-1) ...
Setting up python3-dotenv (0.19.2-1) ...
Setting up python3-texttable (1.6.4-1) ...
Setting up python3-docopt (0.6.2-4) ...
Setting up python3-websocket (1.2.3-1) ...
Setting up python3-dockerpty (0.4.1-2) ...
Setting up python3-docker (5.0.3-1) ...
Setting up docker-compose (1.29.2-1) ...
Processing triggers for man-db (2.10.2-1) ...
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-172-31-35-93:~$ sudo docker-compose --version
docker-compose version 1.29.2, build unknown
ubuntu@ip-172-31-35-93:~$ mkdir Wordpress
ubuntu@ip-172-31-35-93:~/Wordpress$ i-0bb2799412de59b97 (Practical 10)
PublicIPs: 23.20.10.160 PrivateIPs: 172.31.35.93

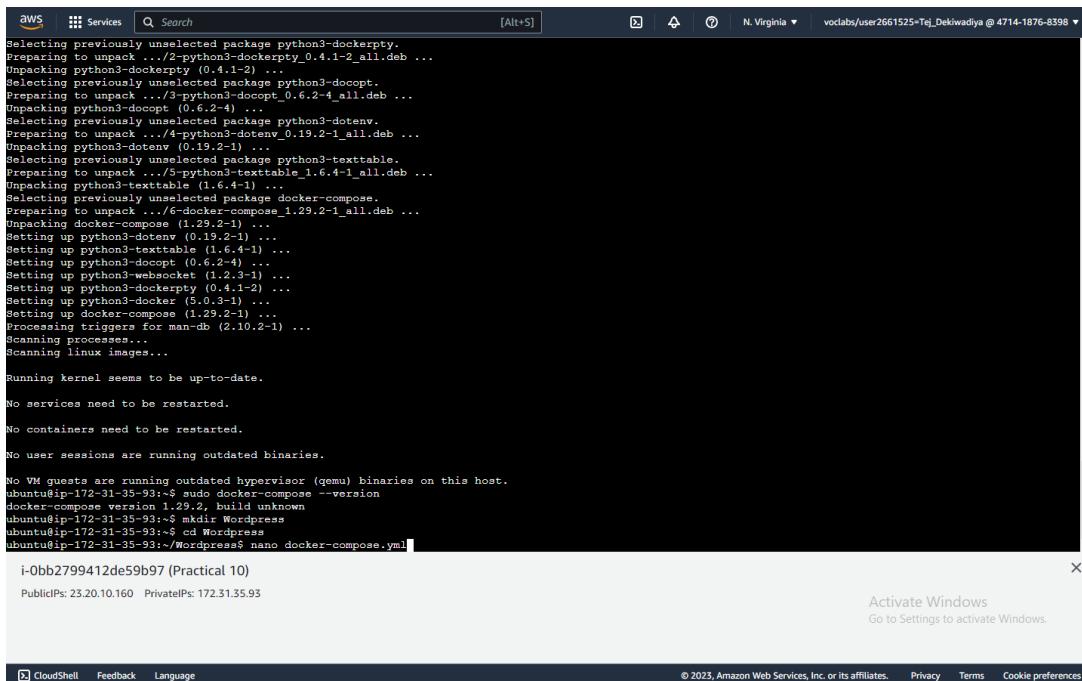
```

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 11 : Run command – nano docker-compose.yml to create a new yml file

Snapshot :



```

AWS Services Search [Alt+S] N. Virginia v vclabs/user2661525-Tej_Dekiwadiya @ 4714-1876-8398
Selecting previously unselected package python3-dockerpty.
Preparing to unpack .../2-python3-dockerpty_0.4.1-2_all.deb ...
Unpacking python3-dockerpty (0.4.1-2) ...
Selecting previously unselected package python3-docopt.
Preparing to unpack .../3-python3-docopt_0.6.2-4_all.deb ...
Unpacking python3-docopt (0.6.2-4) ...
Selecting previously unselected package python3-dotenv.
Preparing to unpack .../4-python3-dotenv_0.19.2-1_all.deb ...
Unpacking python3-dotenv (0.19.2-1) ...
Selecting previously unselected package python3-texttable.
Preparing to unpack .../5-python3-texttable_1.6.4-1_all.deb ...
Unpacking python3-texttable (1.6.4-1) ...
Selecting previously unselected package docker-compose.
Preparing to unpack .../6-docker-compose_1.29.2-1_all.deb ...
Unpacking docker-compose (1.29.2-1) ...
Setting up python3-dotenv (0.19.2-1) ...
Setting up python3-texttable (1.6.4-1) ...
Setting up python3-docopt (0.6.2-4) ...
Setting up python3-websocket (1.2.3-1) ...
Setting up python3-dockerpty (0.4.1-2) ...
Setting up python3-docker (5.0.3-1) ...
Setting up docker-compose (1.29.2-1) ...
Processing triggers for man-db (2.10.2-1) ...
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-172-31-35-93:~$ sudo docker-compose --version
docker-compose version 1.29.2, build unknown
ubuntu@ip-172-31-35-93:~$ mkdir Wordpress
ubuntu@ip-172-31-35-93:~/Wordpress$ nano docker-compose.yml
ubuntu@ip-172-31-35-93:~/Wordpress$ i-0bb2799412de59b97 (Practical 10)
PublicIPs: 23.20.10.160 PrivateIPs: 172.31.35.93

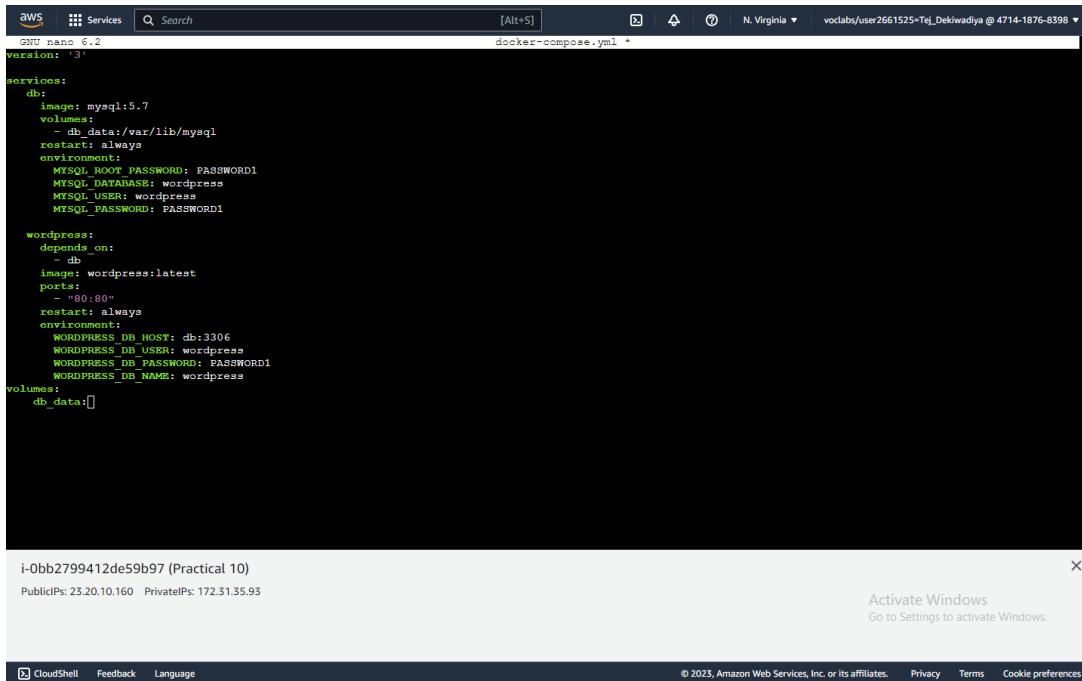
```

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 12 : Inside file write down code that redirects to Wordpress Website

Snapshot :



```

AWS Services Search [Alt+S] N. Virginia v vocabs/user2661525-Tej_Dekiwadiya @ 4714-1876-8398
DNS: nano 6.2
version: '3'
services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: PASSWORD1
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: PASSWORD1
  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "80:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: PASSWORD1
      WORDPRESS_DB_NAME: wordpress
volumes:
  db_data:[]

```

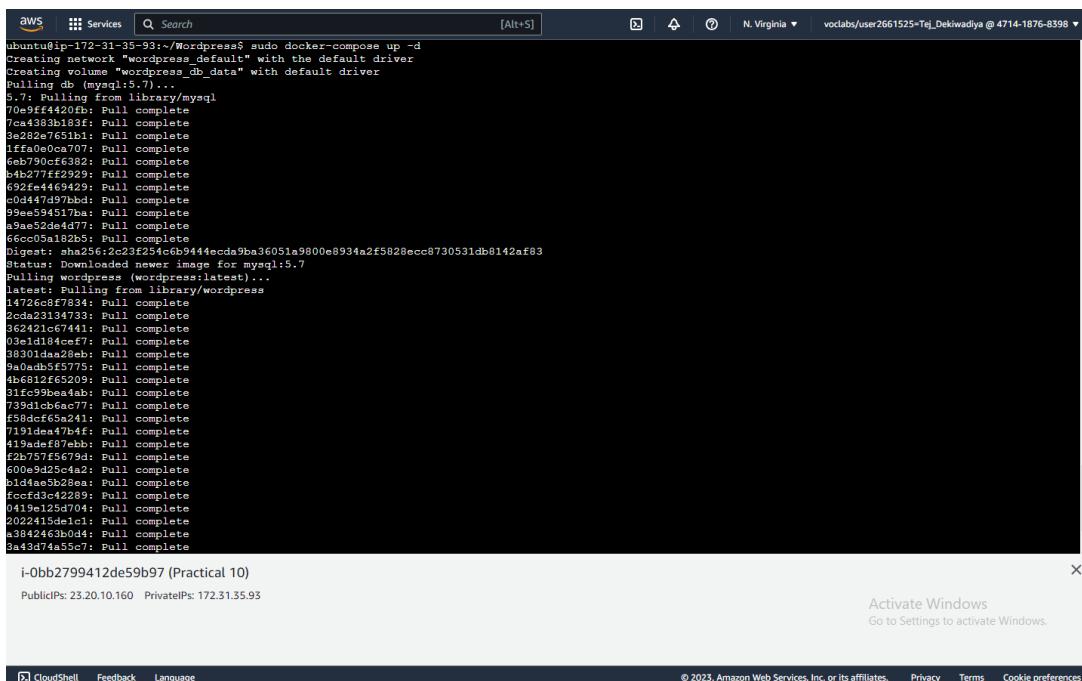
i-0bb2799412de59b97 (Practical 10)
Public IPs: 23.20.10.160 Private IPs: 172.31.35.93

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 13 : Run command – sudo docker-compose up -d.

Snapshot :



```

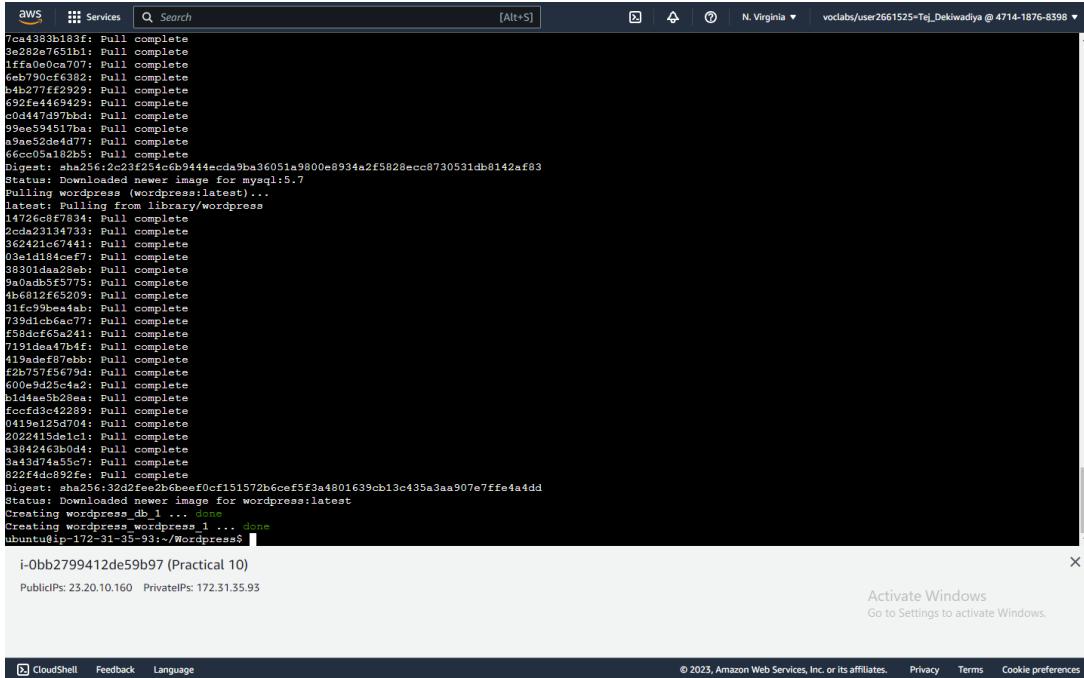
ubuntu@ip-172-31-35-93:~/Wordpress$ sudo docker-compose up -d
Creating network "wordpress_default" with the default driver
Creating volume "wordpress_db_data" with default driver
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
70e9ff4f420fb: Pull complete
7ca4383b183f: Pull complete
3e985a05430c: Pull complete
1fffa0a6cc707: Pull complete
6eb790cf6382: Pull complete
b4b277fe2f929: Pull complete
692fe4469429: Pull complete
c0d447d97hb: Pull complete
99ea594517ba: Pull complete
a9ea52daed77: Pull complete
66cc05a182b5: Pull complete
Digest: sha256:2c23f254c6b944ecda9ba36051a9800e8934a2f5828ecc8730531db8142af83
Status: Download newer image for mysql:5.7
Pulling wordpress (wordpress:latest)...
library/wordpress: Pulling from library/wordpress
1473c8e7f834: Pull complete
2cd523134733: Pull complete
362421c67441: Pull complete
03e1d184cef7: Pull complete
38301daa29eb: Pull complete
9aa0ab5f5775: Pull complete
4b6812f65209: Pull complete
31fc99bea4ab: Pull complete
739d1cb6ac7c7: Pull complete
f58dcf65a241: Pull complete
7191d6e47b4f: Pull complete
4159a233333: Pull complete
62975745c794: Pull complete
600e9d25c54a2: Pull complete
b1d44ae5b2b8ea: Pull complete
ffccfd3c42289: Pull complete
0419e125d704: Pull complete
2022415delcl1: Pull complete
a3842463b0d4: Pull complete
3a43d74a55c7: Pull complete

```

i-0bb2799412de59b97 (Practical 10)
Public IPs: 23.20.10.160 Private IPs: 172.31.35.93

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



```

aws Services Search [Alt+S] N. Virginia v vocabs/user2661525-Tej_Dekiwadiya @ 4714-1876-8398 ▾

7ca4383b183f: Pull complete
3e282e7651b1: Pull complete
1ffa0e0ca707: Pull complete
6eb790c6f6382: Pull complete
b4b277fe2929: Pull complete
692fe4469429: Pull complete
c0d0bb4bbd: Pull complete
98a594515a11: Pull complete
a9ac522de4d77: Pull complete
66cc55a182b5: Pull complete
Digest: sha256:2c23f254:c69444ecda9ba36051a9800e8934a2E5828ecc8730531db8142af83
Status: Downloaded newer image for mysql:5.7
Pulling wordpress (wordpress:latest)...
latest: Pulling from library/wordpress
14726c8f7834: Pull complete
2ca2a3134733: Pull complete
362441c67441: Pull complete
User: Pull complete
9301daa29ab: Pull complete
930adab5f5775: Pull complete
4b6812f65209: Pull complete
311c939bea4ab: Pull complete
739d1cb6a77: Pull complete
f58dcf65a241: Pull complete
7191dea47b4f: Pull complete
419adef87eb: Pull complete
f2b757f45679d: Pull complete
600e9d25c4a2: Pull complete
b1d1a38a2b28ea: Pull complete
f0c1a38a2b28ea: Pull complete
0419c125d704: Pull complete
2022415delc1: Pull complete
e30424463b0d4: Pull complete
3a43d74a55c7: Pull complete
822f4dc892f: Pull complete
Digest: sha256:32d2fee2b6beef0cf151572b6cef5f3a4801639cb13c435a3aa907e7ffe4a4dd
Status: Downloaded newer image for wordpress:latest
Creating wordpress_1 ... done
Creating wordpress_db_1 ... done
Creating wordpress_wordpress_1 ... done
ubuntu@ip-172-31-35-93:~/Wordpress$ [REDACTED]

```

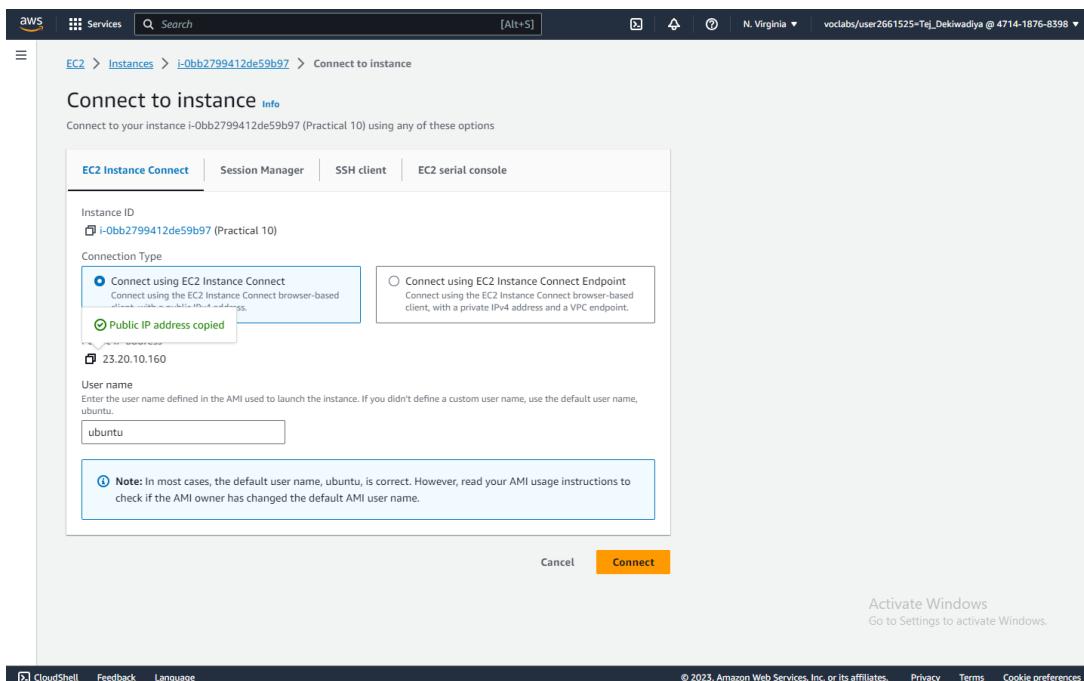
i-0bb2799412de59b97 (Practical 10)
PublicIPs: 23.20.10.160 PrivateIPs: 172.31.35.95

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

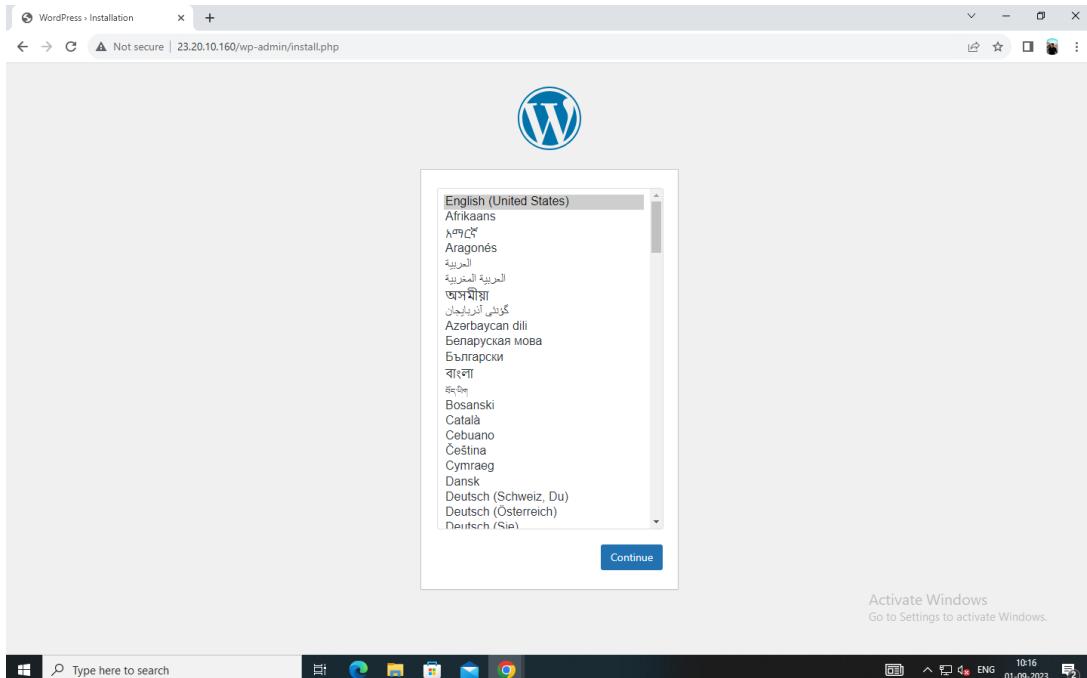
Step 14 : Copy the public IP address.

Snapshot :



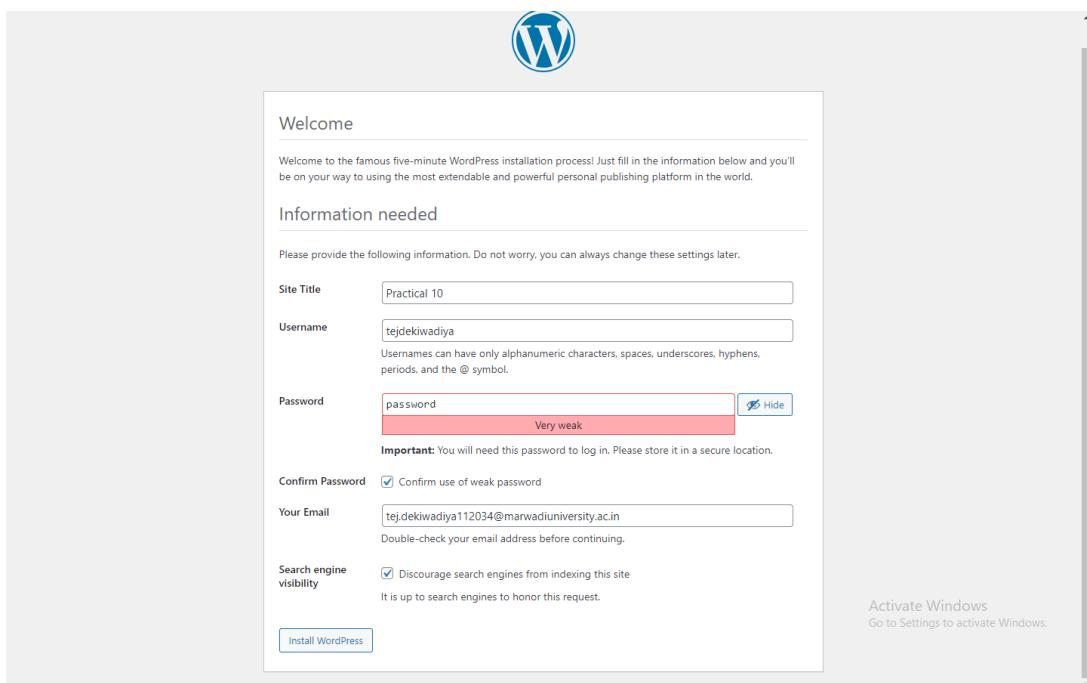
Step 15 : Paste the public IP address in a new tab. And click on continue.

Snapshot :



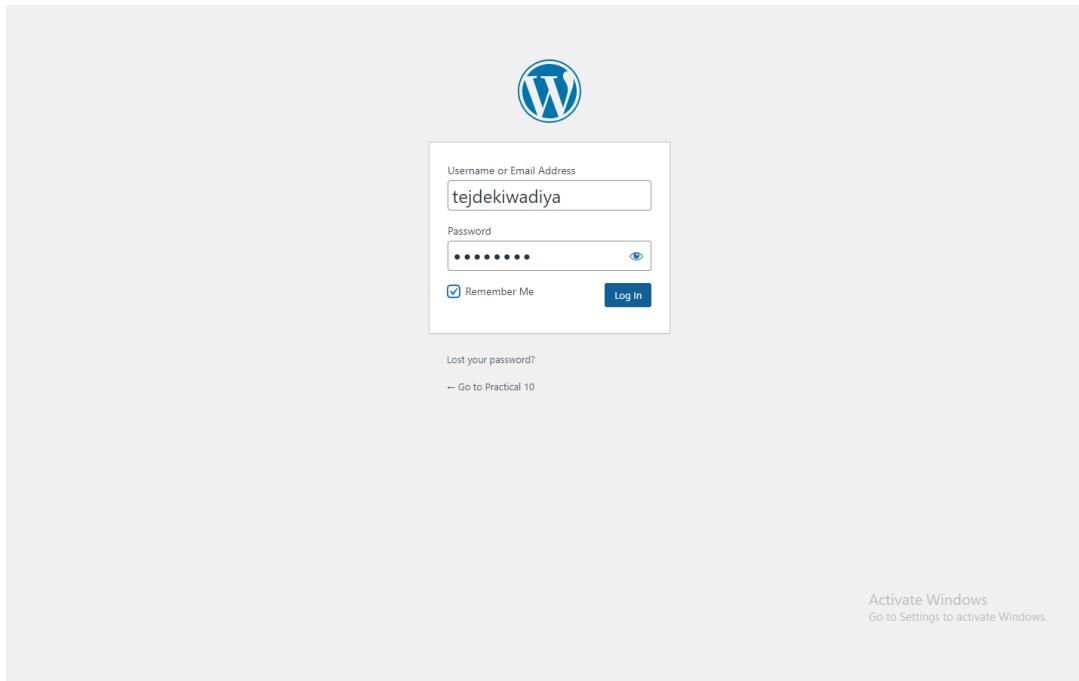
Step 16 : Fill the details to create an account in WordPress. And click on ‘Install WordPress’.

Snapshot :



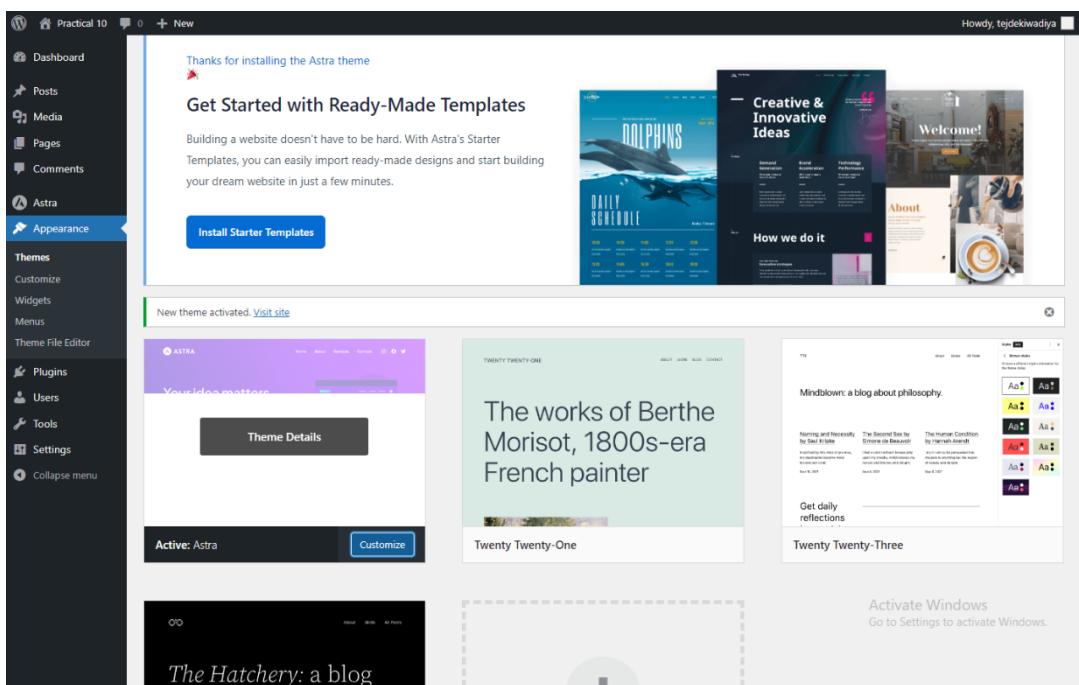
Step 17 : Log In to WordPress.

Snapshot :



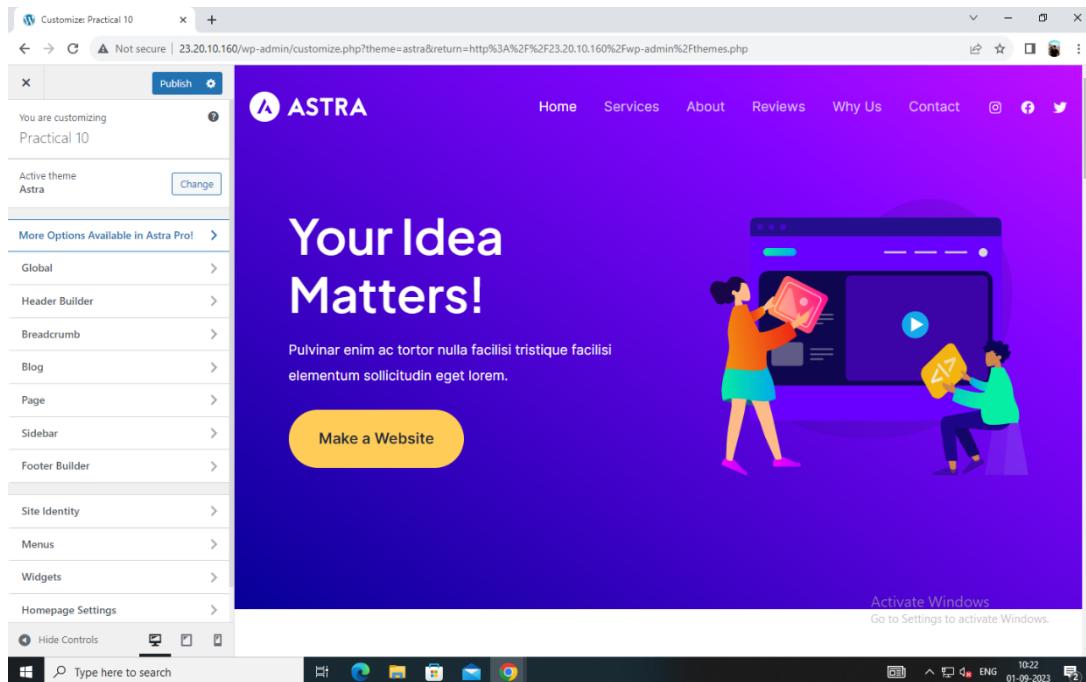
Step 18 : Now go to Appearance >> Themes.

Snapshot :



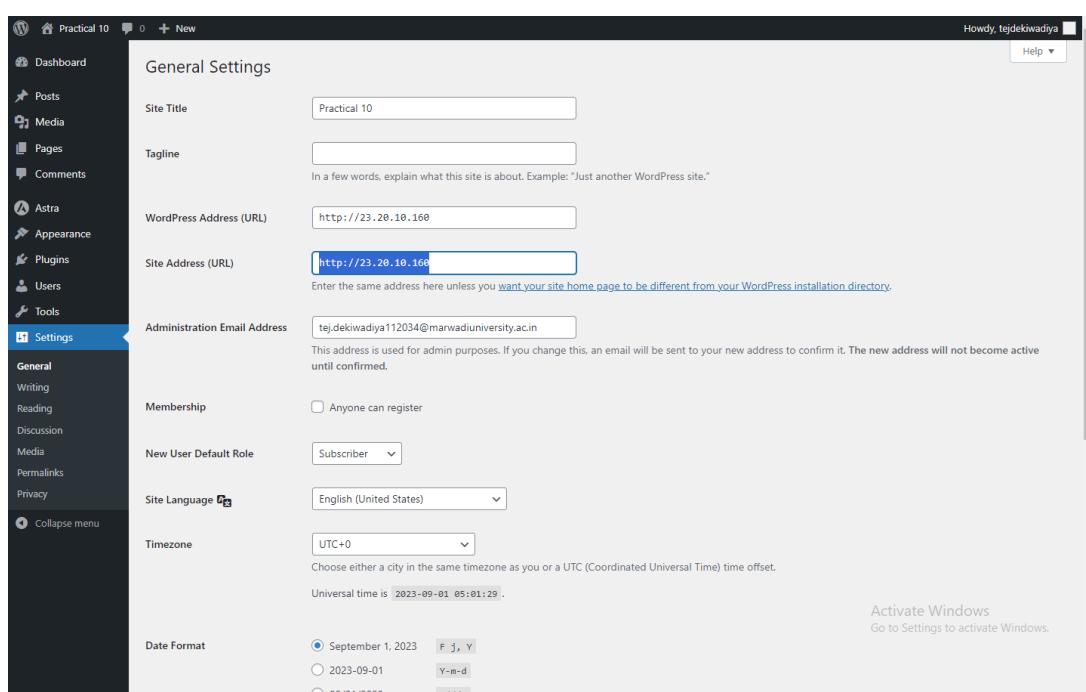
Step 19 : Edit Template and Publish it.

Snapshot :



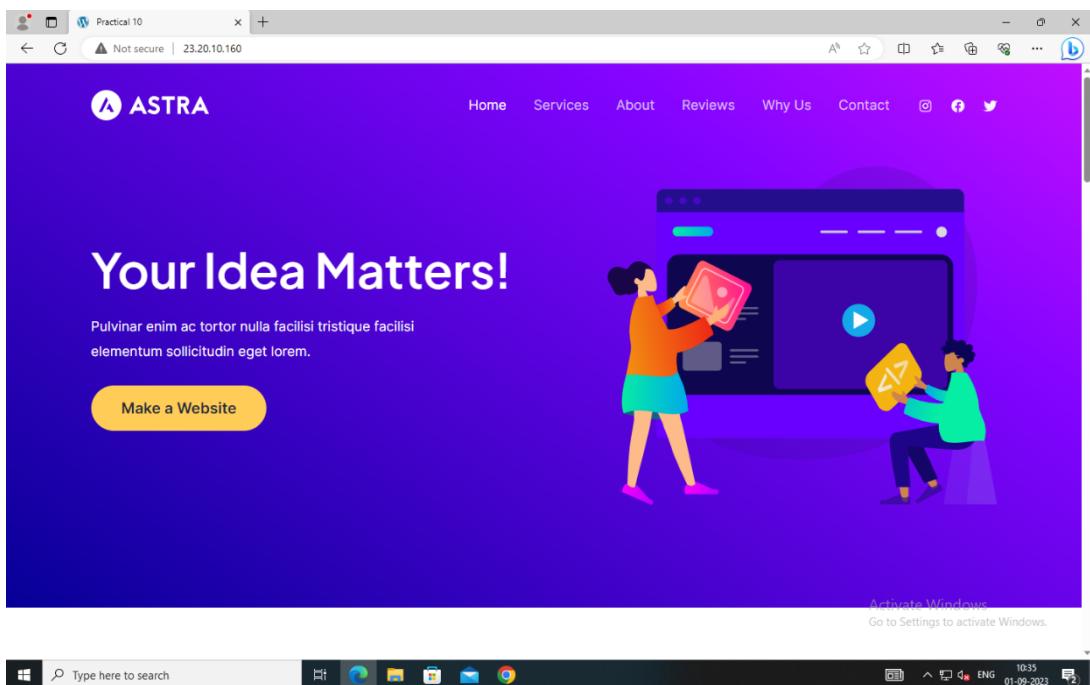
Step 20 : Go to settings and copy Site Address(URL)

Snapshot :



Step 21 : Paste the Site Address (URL) in a new tab.

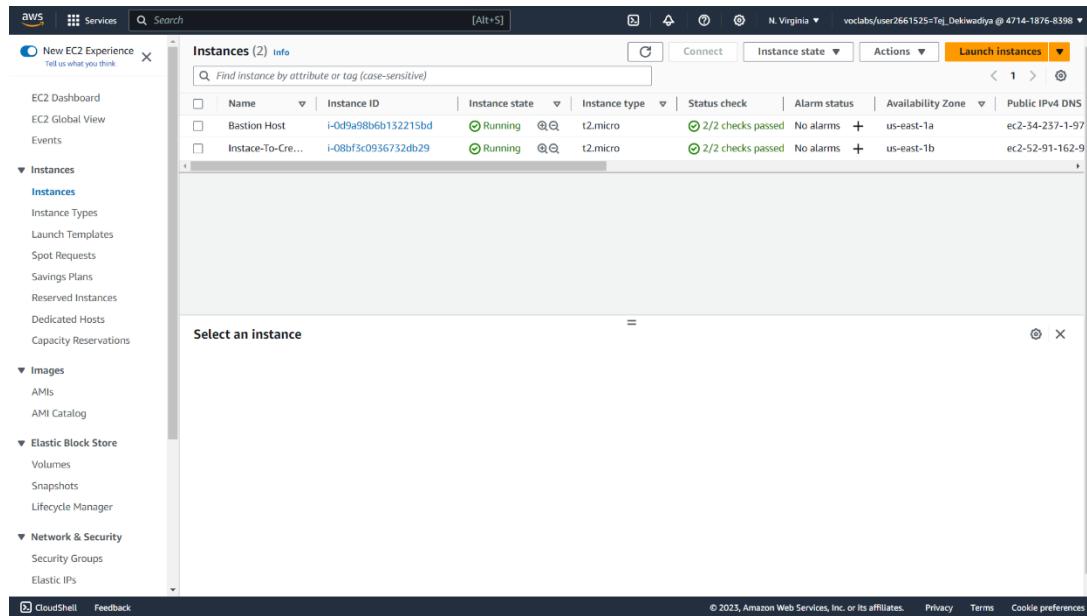
Snapshot :



Practical 11 : Study on Hadoop Framework

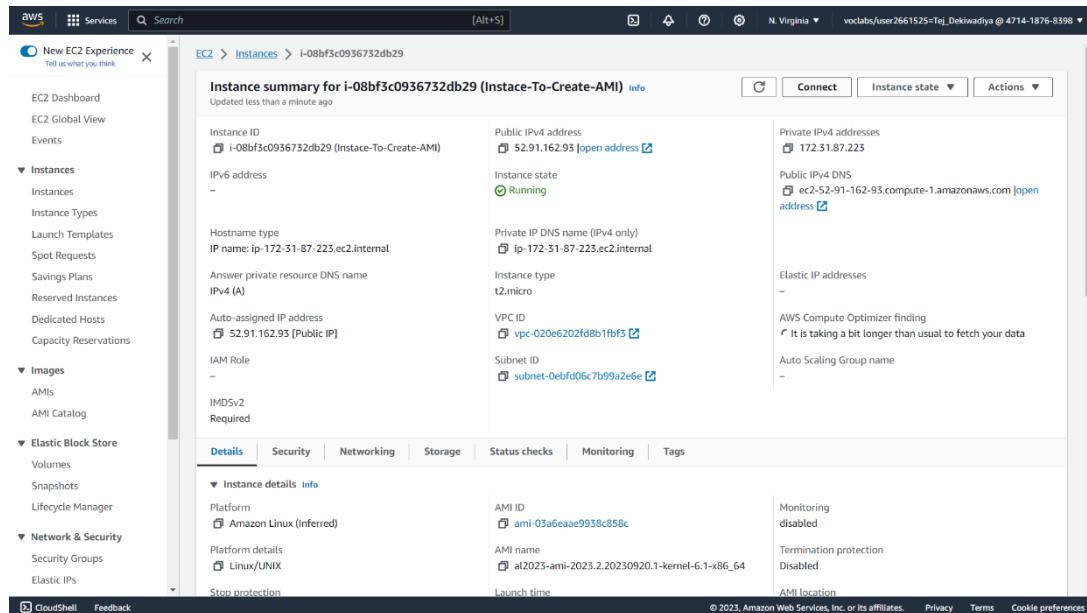
Step 01 : Create an Instance in EC2. Go to Instance ID.

Snapshot :



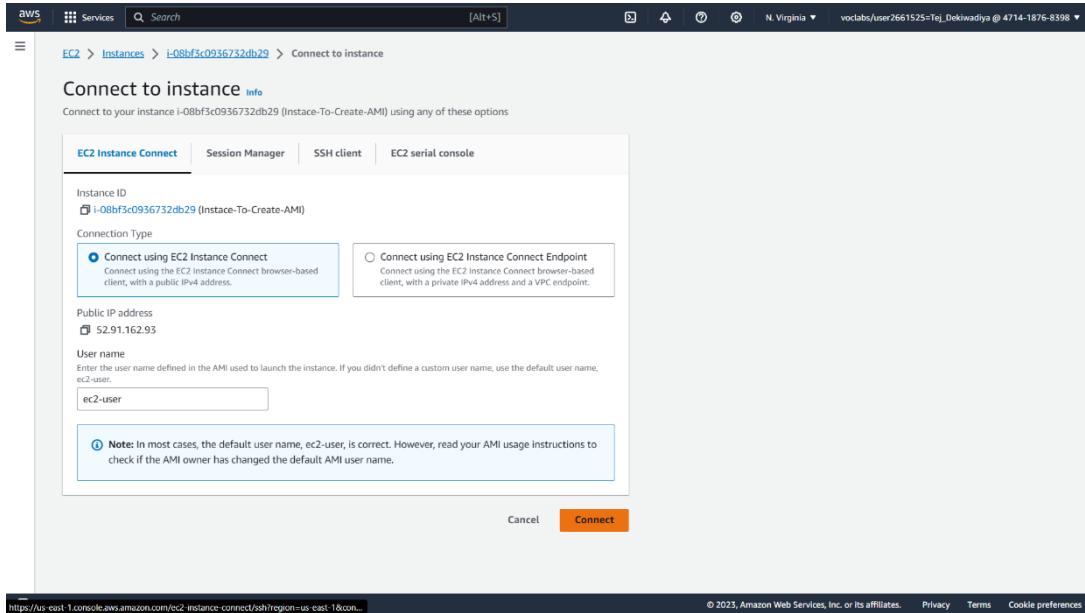
Step 02 : Click on Connect.

Snapshot :



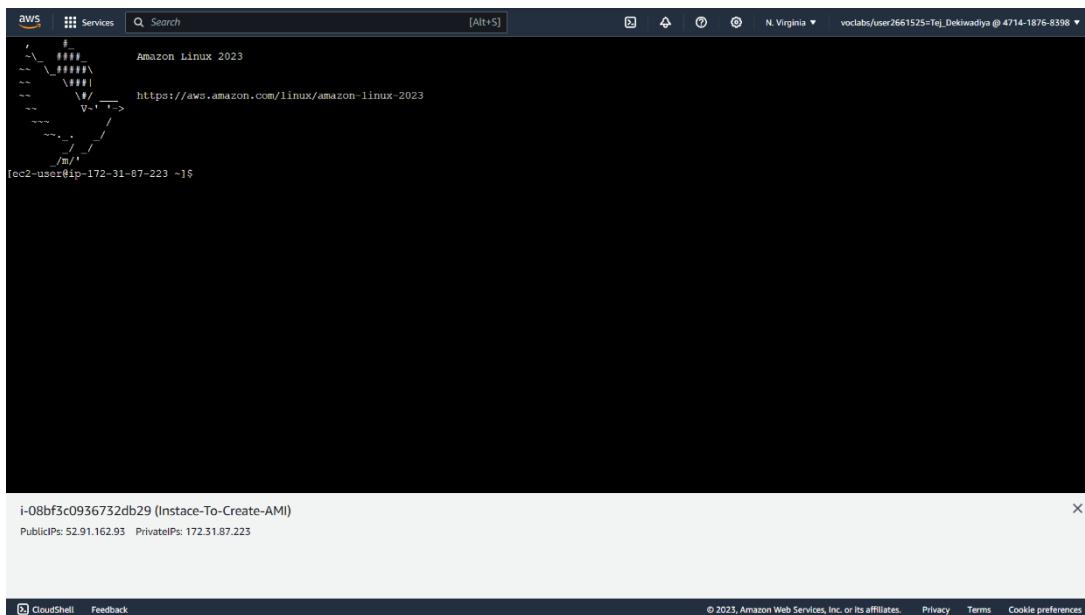
Step 03 : Click on Connect.

Snapshot :



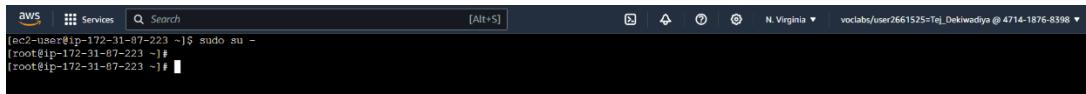
Step 04 : An Instance in EC2 is created.

Snapshot :



Step 05 : Run command ‘sudo su -’.

Snapshot :



```
[ec2-user@ip-172-31-87-223 ~]$ sudo su -
[root@ip-172-31-87-223 ~]#
```

Step 06 : Run command ‘yum update -y’.

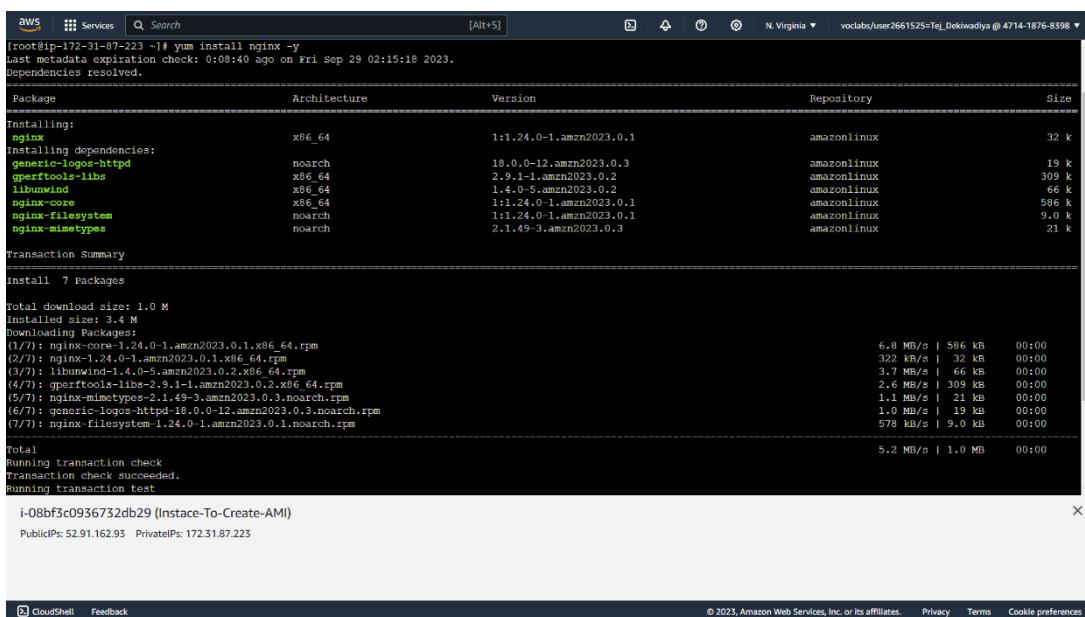
Snapshot :



```
[root@ip-172-31-87-223 ~]# yum update -y
Last metadata expiration check: 0:06:36 ago on Fri Sep 29 02:15:18 2023.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-87-223 ~]#
```

Step 07 : Run command ‘yum install nginx -y’.

Snapshot :



```
[root@ip-172-31-87-223 ~]# yum install nginx -y
Last metadata expiration check: 0:08:40 ago on Fri Sep 29 02:15:18 2023.
Dependencies resolved.

=====
Installing:
nginx.x86_64 1:1.24.0-1.amzn2023.0.1
=====
Installing dependencies:
generic-logs-nginx.x86_64 18.0.0-12.amzn2023.0.3
gperftools-libs.x86_64 2.9.1-1.amzn2023.0.2
libunwind.x86_64 1.4.0-5.amzn2023.0.2
nginx-core.x86_64 1:1.24.0-1.amzn2023.0.1
nginx-filesystem.noarch 1:1.24.0-1.amzn2023.0.1
nginx-mimetypes.noarch 2.1.49-3.amzn2023.0.3

=====
Transaction Summary
=====
Install 7 Packages

Total download size: 1.0 M
Installed size: 3.4 M
Downloaded Packages:
(1/7): nginx-1.24.0-1.amzn2023.0.1.x86_64.rpm 6.8 MB/s | 586 kB 00:00
(2/7): generic-logs-nginx-18.0.0-12.amzn2023.0.1.x86_64.rpm 322 kB/s | 32 kB 00:00
(3/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm 3.7 MB/s | 66 kB 00:00
(4/7): gperftools-libs-2.9.1-1.amzn2023.0.2.x86_64.rpm 2.6 MB/s | 309 kB 00:00
(5/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm 1.1 MB/s | 21 kB 00:00
(6/7): generic-logs-nginx-18.0.0-12.amzn2023.0.3.noarch.rpm 1.0 MB/s | 19 kB 00:00
(7/7): nginx-filesystem-1.24.0-1.amzn2023.0.1.noarch.rpm 578 kB/s | 9.0 kB 00:00

Total
Running transaction check
Transaction check succeeded.
Running transaction test

i-08bf3c0936732db29 (Instate-To-Create-AMI)
PublicIPs: 52.91.162.93 PrivateIPs: 172.31.87.223
```

```
aws Services Search [Alt+S] N. Virginia vobcls/user661S2sTej.Dekiwadiya @ 4714-1876-8398
(5/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm          1.1 MB/s | 21 kB   00:00
(6/7): generic-logos-httpsd-18.0.0-12.amzn2023.0.3.noarch.rpm    1.0 MB/s | 19 kB   00:00
(7/7): nginx-filesystem-1.24.0-1.amzn2023.0.1.noarch.rpm        578 kB/s | 9.0 kB   00:00

Total                                         5.2 MB/s | 1.0 MB   00:00

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing:
    Running scriptlet: nginx-filesystem-1:1.24.0-1.amzn2023.0.1.noarch
  Installing : nginx-filesystem-1:1.24.0-1.amzn2023.0.1.noarch
  Installing : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
  Installing : generic-logos-httpsd-18.0.0-12.amzn2023.0.3.noarch
  Installing : libunwind-1.4.0-5.amzn2023.0.2.x86_64
  Installing : gperftools-libs-2.9.1-1.amzn2023.0.2.x86_64
  Installing : nginx-core-1:1.24.0-1.amzn2023.0.1.x86_64
  Installing : nginx-1:1.24.0-1.amzn2023.0.1.x86_64
  Running scriptlet: nginx-1:1.24.0-1.amzn2023.0.1.x86_64
  Verifying   : gperftools-libs-2.9.1-1.amzn2023.0.2.x86_64
  Verifying   : nginx-core-1:1.24.0-1.amzn2023.0.1.x86_64
  Verifying   : nginx-1:1.24.0-1.amzn2023.0.1.x86_64
  Verifying   : libunwind-1.4.0-5.amzn2023.0.2.x86_64
  Verifying   : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
  Verifying   : generic-logos-httpsd-18.0.0-12.amzn2023.0.3.noarch
  Verifying   : nginx-filesystem-1:1.24.0-1.amzn2023.0.1.noarch

Installed:
  generic-logos-httpsd-18.0.0-12.amzn2023.0.3.noarch           gperftools-libs-2.9.1-1.amzn2023.0.2.x86_64
  nginx-1:1.24.0-1.amzn2023.0.1.x86_64                         nginx-core-1:1.24.0-1.amzn2023.0.1.x86_64
  nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-172-31-87-223 ~]#
```

Step 08 : Run command ‘`sudo service nginx start`’.

Snapshot :

```
[root@81c-172-31-87-223 ~]# sudo service nginx start
[root@81c-172-31-87-223 ~]# sudo systemctl start nginx.service
[root@81c-172-31-87-223 ~]#
```

Step 09 : Run command ‘`systemctl start nginx.service`’.

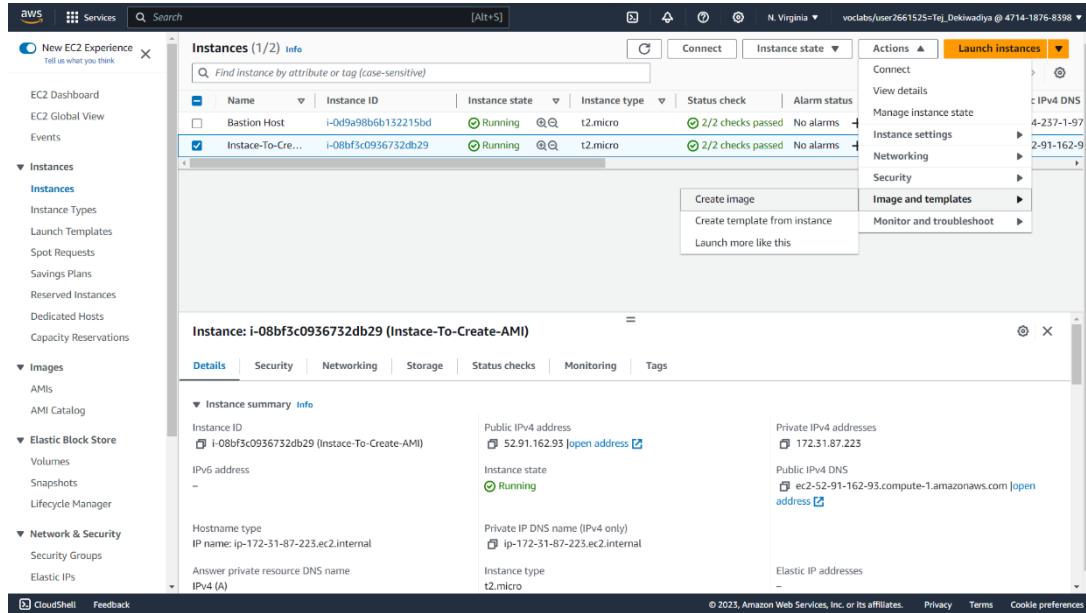
Snapshot :

```
aws Services Search [Alt+S] N. Virginia vocabs/user2661525=Tej_Dekwadiya @ 4714-1876-8398
(root@ip-172-31-87-223 ~) | sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[root@ip-172-31-87-223 ~] systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
   Active: active (running) since Fri 2023-09-29 02:25:17 UTC; 41s ago
     Process: 25976 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 25977 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 25978 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
 Main PID: 25979 (nginx)
   Tasks: 2 (limit: 1114)
  Memory: 2.2M
    CPU: 55ms
   CGroup: /system.slice/nginx.service
           ├─25979 "nginx: master process /usr/sbin/nginx"
           └─25980 "nginx: worker process"

Sep 29 02:25:16 ip-172-31-87-223.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Sep 29 02:25:16 ip-172-31-87-223.ec2.internal nginx[25977]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Sep 29 02:25:16 ip-172-31-87-223.ec2.internal nginx[25977]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Sep 29 02:25:17 ip-172-31-87-223.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[root@ip-172-31-87-223 ~] #
```

Step 10 : Go to Instances >> Actions >> Image and Templates >> Create image.

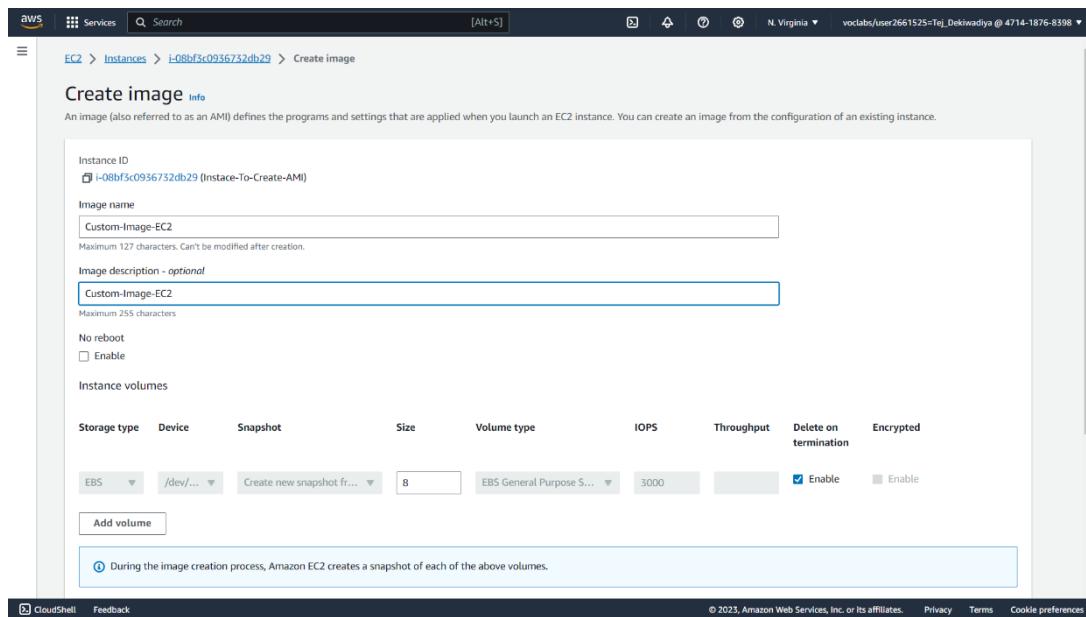
Snapshot :



The screenshot shows the AWS EC2 Instances page. There are two instances listed: 'Bastion Host' and 'Instace-To-Cre...'. Both instances are running and of type t2.micro. A context menu is open over the 'Instace-To-Cre...' instance, with the 'Create image' option highlighted under the 'Actions' dropdown.

Step 11 : Create image.

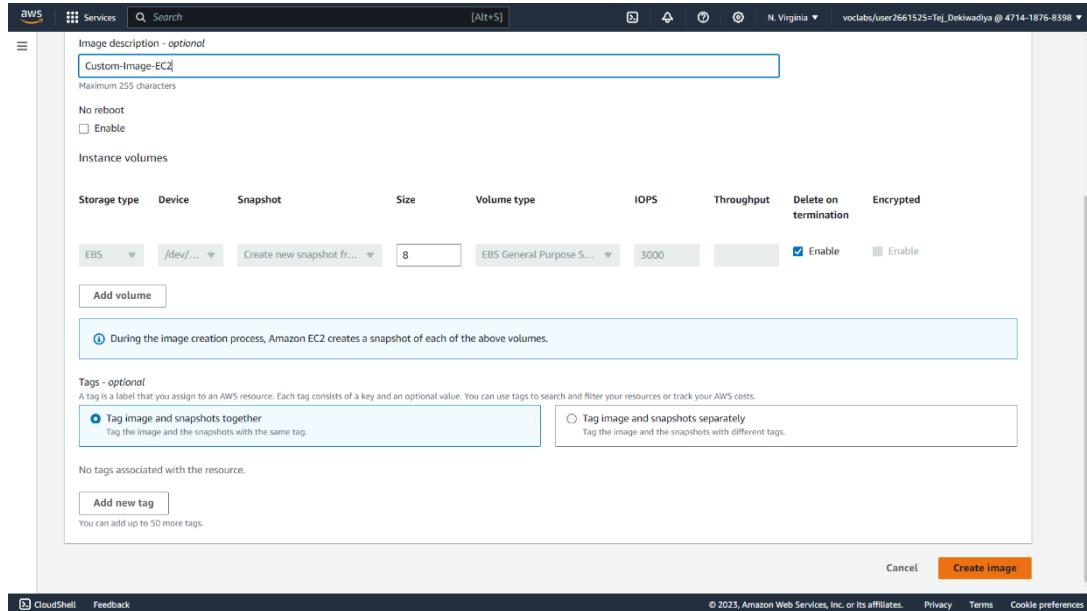
Snapshot :



The screenshot shows the 'Create image' wizard for the selected EC2 instance. The 'Image name' field is set to 'Custom-Image-EC2'. The 'Image description' field is also set to 'Custom-Image-EC2'. Under 'Instance volumes', there is one EBS volume selected with a size of 8 GiB, volume type EBS General Purpose S..., IOPS 3000, Throughput 1000, Delete on termination checked, and Encrypted unchecked. A note at the bottom states: 'During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.'

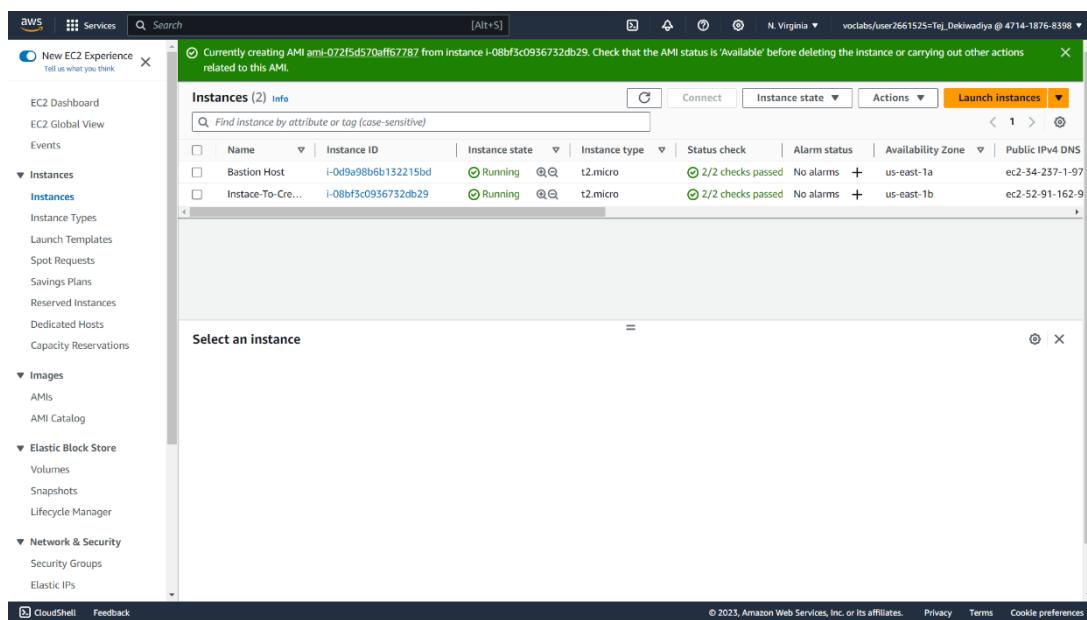
Step 12 : Select ‘Tag image and snapshots together.’ And click on Create image.

Snapshot :



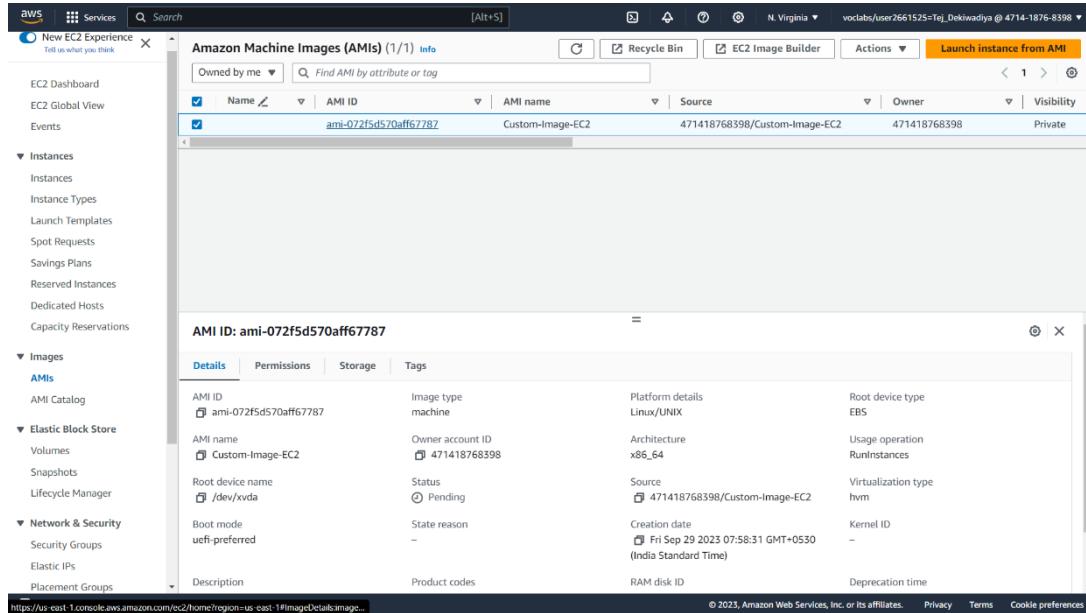
Step 13 : AMI created.

Snapshot :



Step 14 : Go to Images >> AMIs >> AMI ID

Snapshot :



The screenshot shows the AWS EC2 console with the 'Images' section selected. Under 'AMIs', one item is listed:

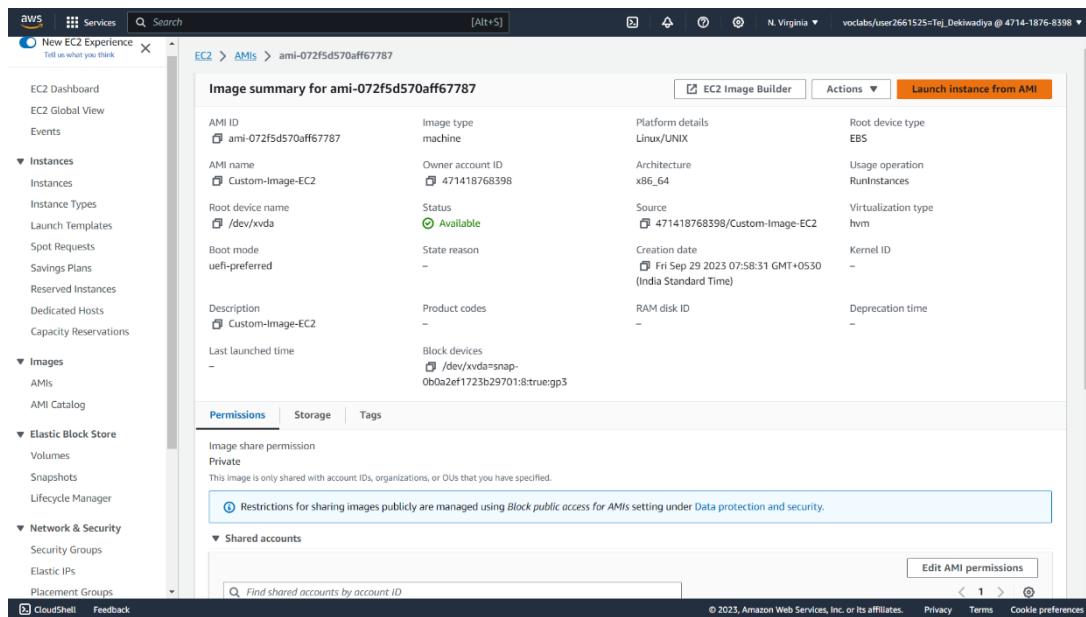
Name	AMI ID	AMI name	Source	Owner	Visibility
Custom-Image-EC2	ami-072f5d570aff67787	Custom-Image-EC2	471418768398/Custom-Image-EC2	471418768398	Private

Details for the selected AMI (ami-072f5d570aff67787) are shown in a modal window:

AMI ID	Image type	Platform details	Root device type
ami-072f5d570aff67787	machine	Linux/UNIX	EBS
AMI name	Owner account ID	Architecture	Usage operation
Custom-Image-EC2	471418768398	x86_64	RunInstances
Root device name	Status	Source	Virtualization type
/dev/xvda	Pending	471418768398/Custom-Image-EC2	hvm
Boot mode	State reason	Creation date	Kernel ID
uefi-preferred	-	Fri Sep 29 2023 07:58:31 GMT+0530 (India Standard Time)	-
Description	Product codes	RAM disk ID	Deprecation time
Custom-Image-EC2	-	-	-

Step 15 : Launch instance from AMI. Create 3 Instances.

Snapshot :



The screenshot shows the AWS EC2 console with the 'Images' section selected. Under 'AMIs', the previously selected AMI is shown in more detail:

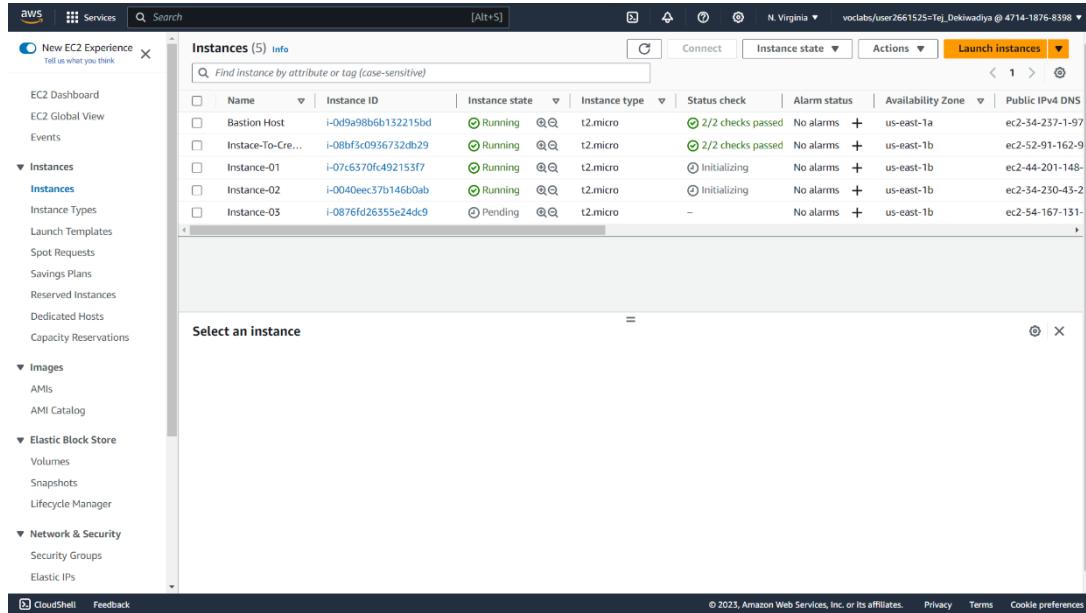
AMI ID	Image type	Platform details	Root device type
ami-072f5d570aff67787	machine	Linux/UNIX	EBS
AMI name	Owner account ID	Architecture	Usage operation
Custom-Image-EC2	471418768398	x86_64	RunInstances
Root device name	Status	Source	Virtualization type
/dev/xvda	Available	471418768398/Custom-Image-EC2	hvm
Boot mode	State reason	Creation date	Kernel ID
uefi-preferred	-	Fri Sep 29 2023 07:58:31 GMT+0530 (India Standard Time)	-
Description	Product codes	RAM disk ID	Deprecation time
Custom-Image-EC2	-	-	-
Last launched time	Block devices		
-	/dev/xvda=snap-0b0a2ef1723b29701:8:true:gp3		

Permissions for the AMI are also displayed:

- Image share permission: Private
- This image is only shared with account IDs, organizations, or OUs that you have specified.
- Restrictions for sharing images publicly are managed using Block public access for AMIs setting under Data protection and security.

Step 16 : Instances created.

Snapshot :

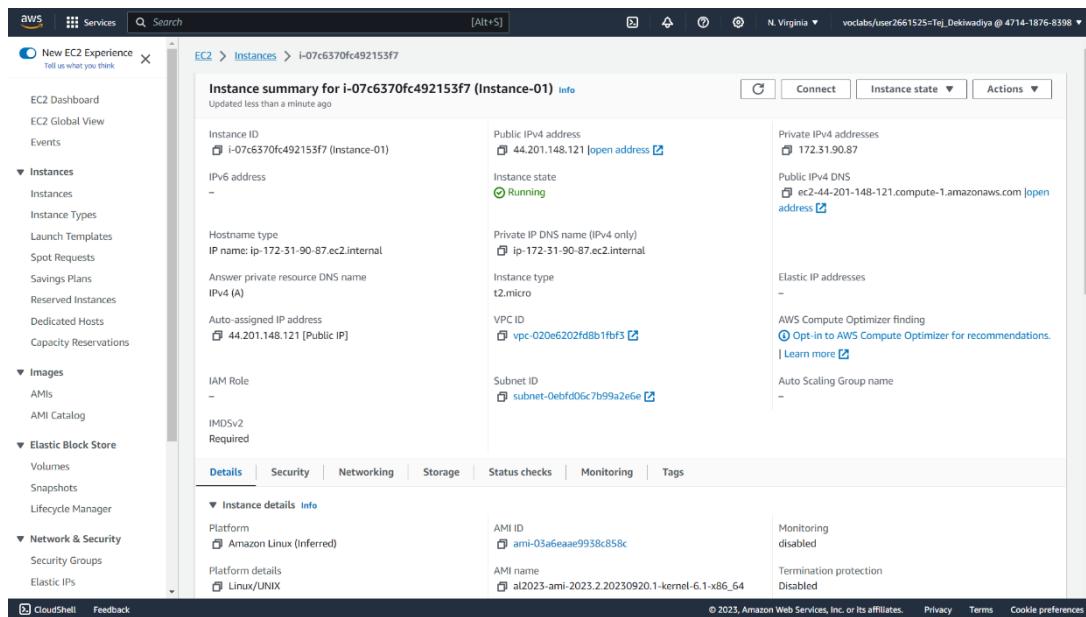


The screenshot shows the AWS EC2 Instances page with the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Bastion Host	i-0d9a98b6b132215bd	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-34-237-1-97
Instace-To-Cre...	i-08bf3c0936732db29	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-52-91-162-9
Instance-01	i-07c6370fc492153f7	Running	t2.micro	Initializing	No alarms	+ us-east-1b	ec2-44-201-148-
Instance-02	i-0040eec37b146b0ab	Running	t2.micro	Initializing	No alarms	+ us-east-1b	ec2-34-230-43-2
Instance-03	i-0876fd26355c24dc9	Pending	t2.micro	-	No alarms	+ us-east-1b	ec2-54-167-131-

Step 17 : Connect Instance 1.

Snapshot :



The screenshot shows the Instance summary for instance i-07c6370fc492153f7 (Instance-01) with the following details:

Instance ID	Private IPv4 address	Private IPv4 addresses
i-07c6370fc492153f7 (Instance-01)	44.201.148.121 [open address]	172.31.90.87
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-44-201-148-121.compute-1.amazonaws.com [open address]
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-90-87.ec2.internal	ip-172-31-90-87.ec2.internal	-
Answer private resource DNS name	Instance type	AWS Compute Optimizer finding
IPv4 (A)	t2.micro	Opt-in to AWS Compute Optimizer for recommendations.
Auto-assigned IP address	VPC ID	Learn more
44.201.148.121 [Public IP]	vpc-020e6202fd8b1fb3	
IAM Role	Subnet ID	Auto Scaling Group name
-	subnet-0ebfd06c7b99a2e6e	-
IMDSv2 Required		

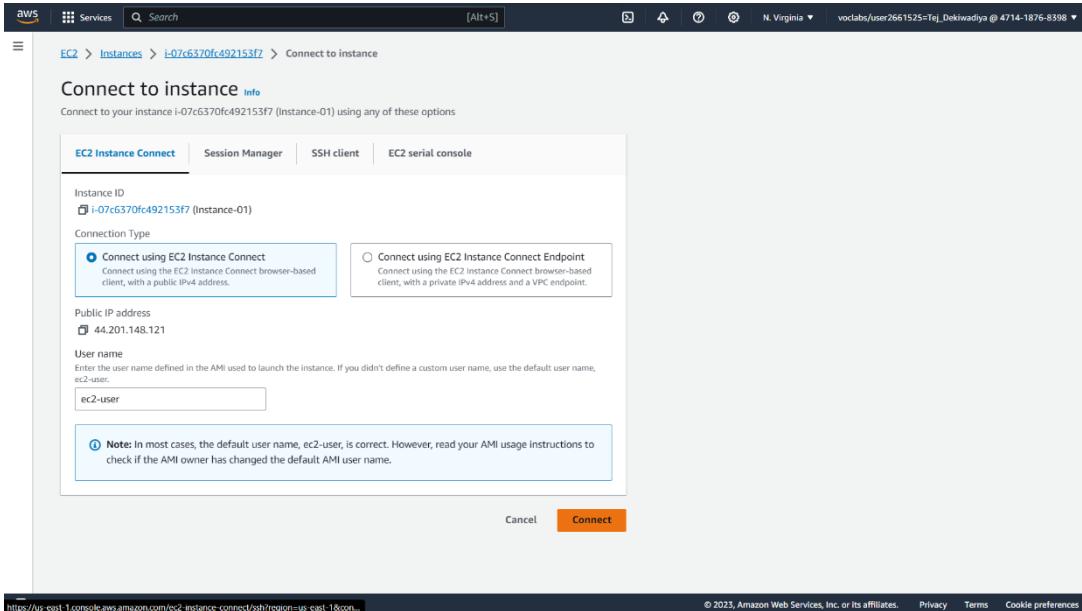
Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance details Info

Platform	AMI ID	Monitoring
Amazon Linux (Inferred)	ami-03a6eaae9938c858c	disabled
Platform details	AMI name	Termination protection
Linux/UNIX	al2023-ami-2023.2.20230920.1-kernel-6.1-x86_64	Disabled

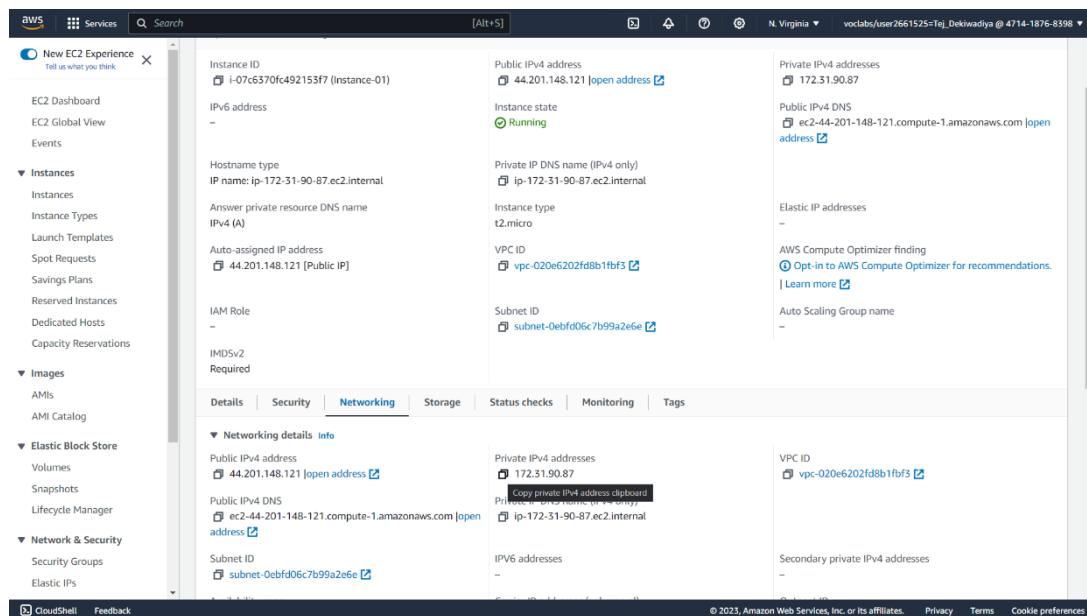
Step 18 : Connect.

Snapshot :



Step 19 : Copy IPv4 address.

Snapshot :



Step 20 : Run ping command Instance 1

Snapshot :

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

```
[ec2-user@ip-172-31-90-87 ~]$ sudo su
[root@ip-172-31-90-87 ec2-user]# ping 44.201.148.121
PING 44.201.148.121 (44.201.148.121) 56(84) bytes of data.
^C

44.201.148.121 ping statistics ---
18 packets transmitted, 0 received, 100% packet loss, time 17682ms

[root@ip-172-31-90-87 ec2-user]# ping 172.31.90.87
PING 172.31.90.87 (172.31.90.87) 56(84) bytes of data.
64 bytes from 172.31.90.87: icmp_seq=1 ttl=127 time=0.020 ms
64 bytes from 172.31.90.87: icmp_seq=2 ttl=127 time=0.036 ms
64 bytes from 172.31.90.87: icmp_seq=3 ttl=127 time=0.036 ms
64 bytes from 172.31.90.87: icmp_seq=4 ttl=127 time=0.034 ms
64 bytes from 172.31.90.87: icmp_seq=5 ttl=127 time=0.035 ms
64 bytes from 172.31.90.87: icmp_seq=6 ttl=127 time=0.039 ms
64 bytes from 172.31.90.87: icmp_seq=7 ttl=127 time=0.033 ms

^C
172.31.90.87 ping statistics ---
7 packets transmitted, 0 received, 0% packet loss, time 6215ms
rtt min/avg/max/mdev = 0.020/0.033/0.033/0.005 ms
[root@ip-172-31-90-87 ec2-user]# ^C
[root@ip-172-31-90-87 ec2-user]# ^C
[root@ip-172-31-90-87 ec2-user]# ]
```

Step 21 : Run ping command in Instance 2

Snapshot :

aws Services Search [Alt+S] N. Virginia vocabs/user2661525=Tej_Dekiwadiya @ 4714-1876-8398



Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

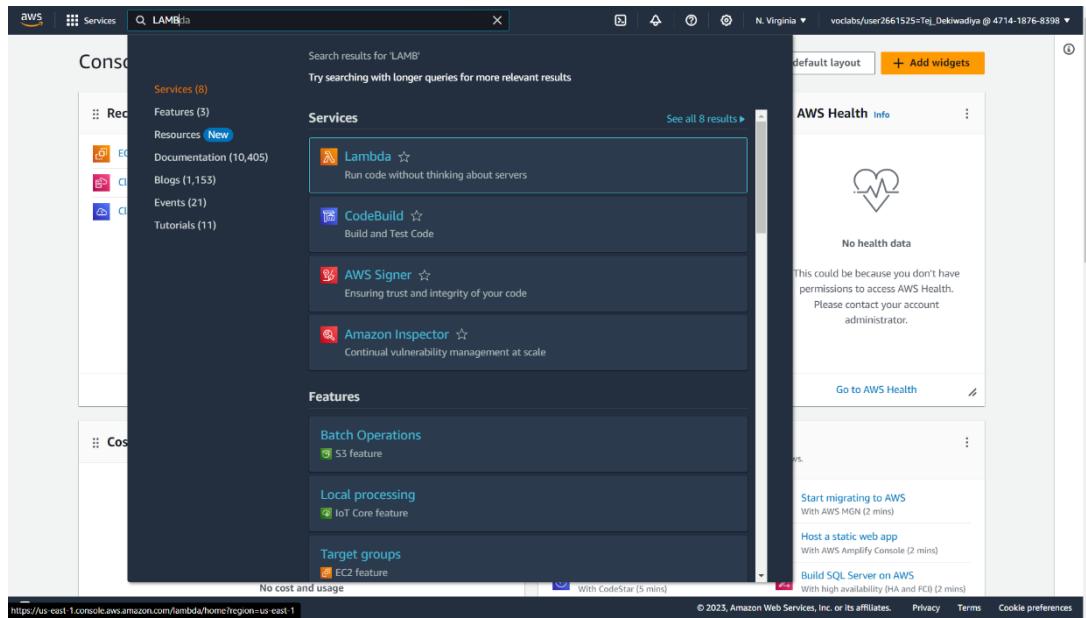
```
fec2-user@ip-172-31-86-219 ~]$ sudo su
-bash: ~sudo: command not found
[ec2-user@ip-172-31-86-219 ~]$ ping 172.31.90.87
PING 172.31.90.87 (172.31.90.87) 56(84) bytes of data.
```

i-0040eec37b146b0ab (Instance-02)
PublicIPs: 34.230.43.209 PrivateIPs: 172.31.86.219

Practical 12 : Using Auto Scaling with AWS Lambda and Lifecycle Hooks

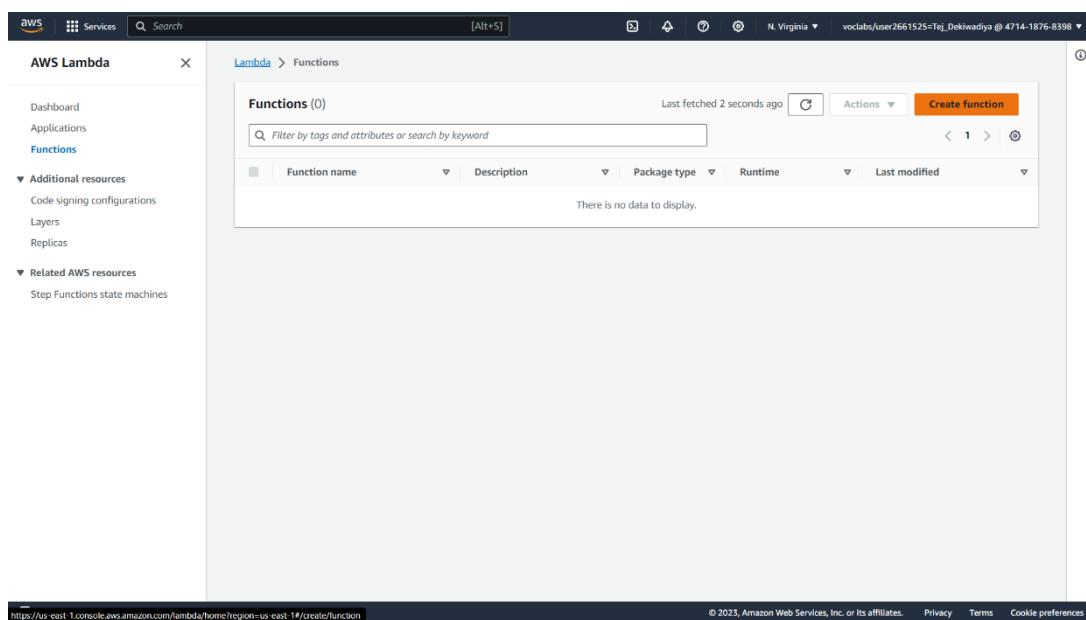
Step 01 : Search Lambda in service section

Snapshot :



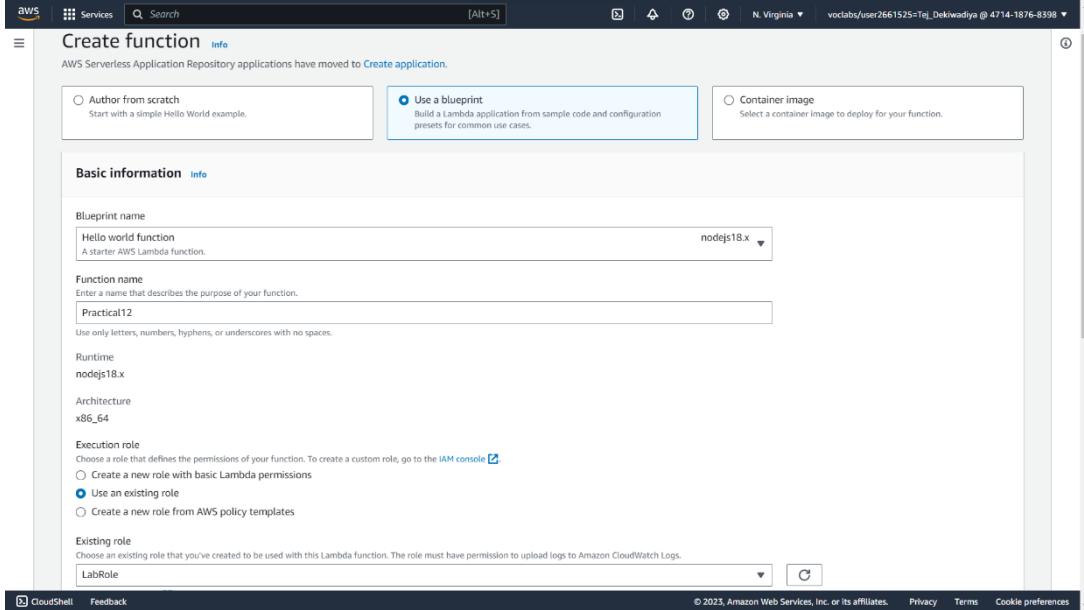
Step 02 : Click on create function

Snapshot :



Step 03 : Create a function using Lambda service

Snapshot :



Basic information

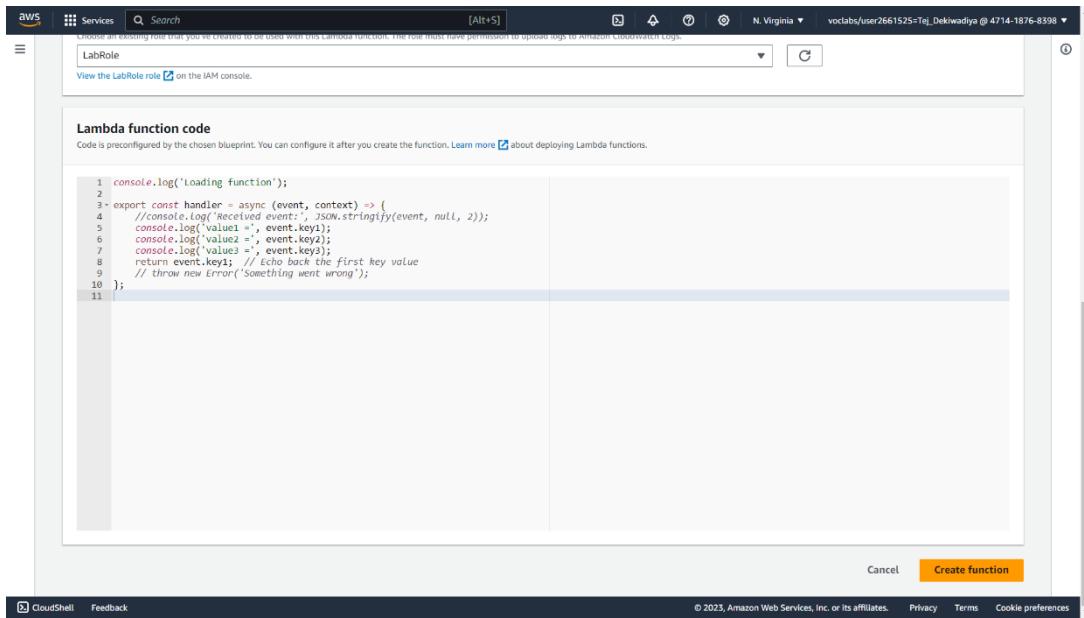
Blueprint name: Hello world function

Runtime: nodejs18.x

Function name: Practical12

Execution role: LabRole

Create function



```

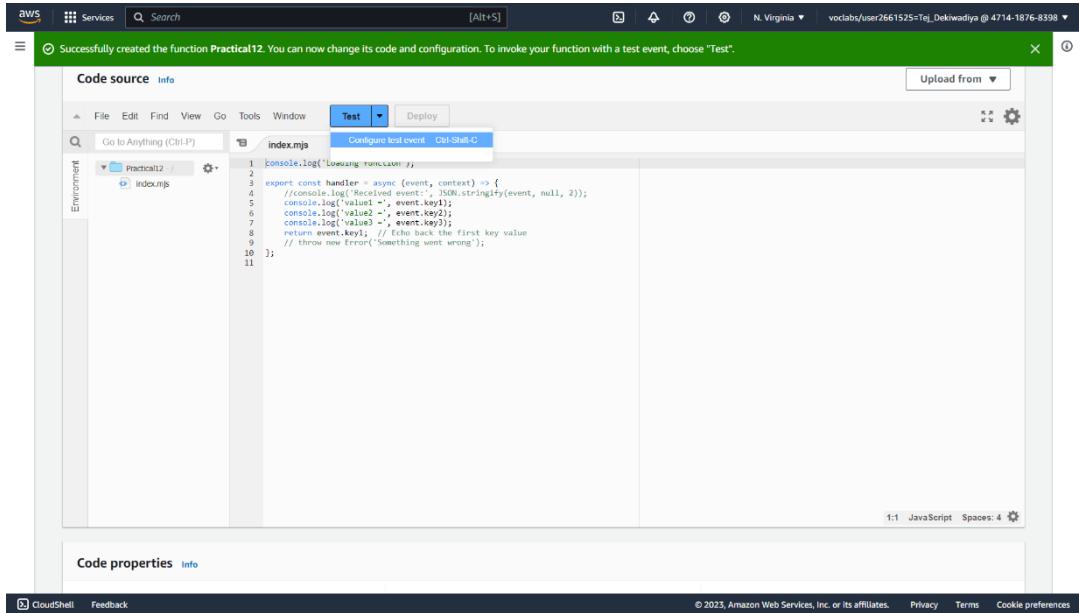
1 console.log('Loading function');
2
3 export const handler = async (event, context) => {
4   //console.log('Received event:', JSON.stringify(event, null, 2));
5   console.log(`value1 = ${event.key1}`);
6   console.log(`value2 = ${event.key2}`);
7   console.log(`value3 = ${event.key3}`);
8   return event.key1; // Echo back the first key value
9   // throw new Error('Something went wrong');
10 };
11

```

Cancel Create function

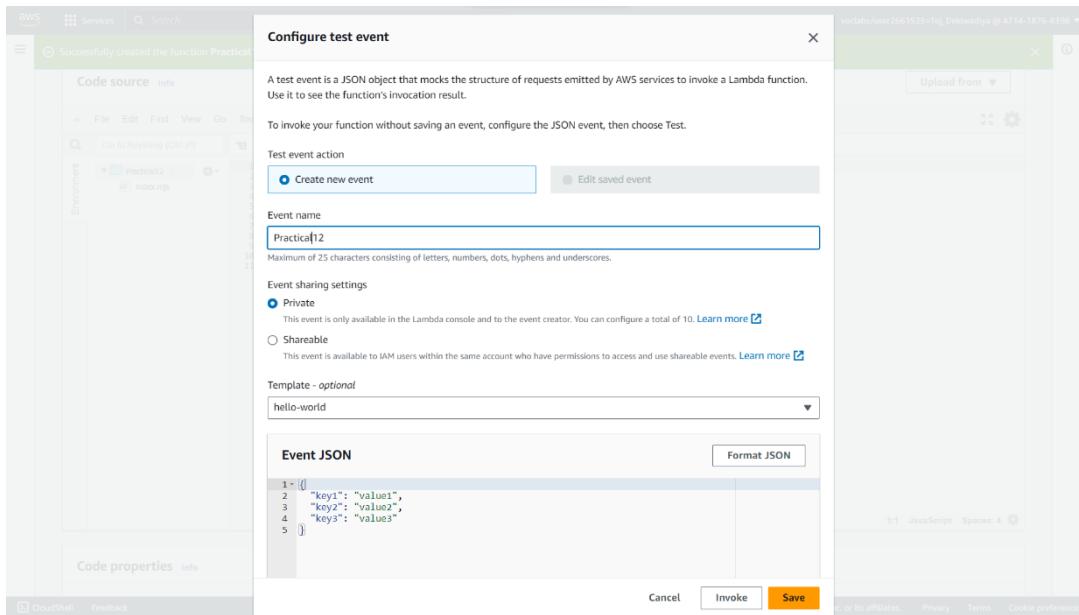
Step 04 : Test >> Configure Test Event

Snapshot :



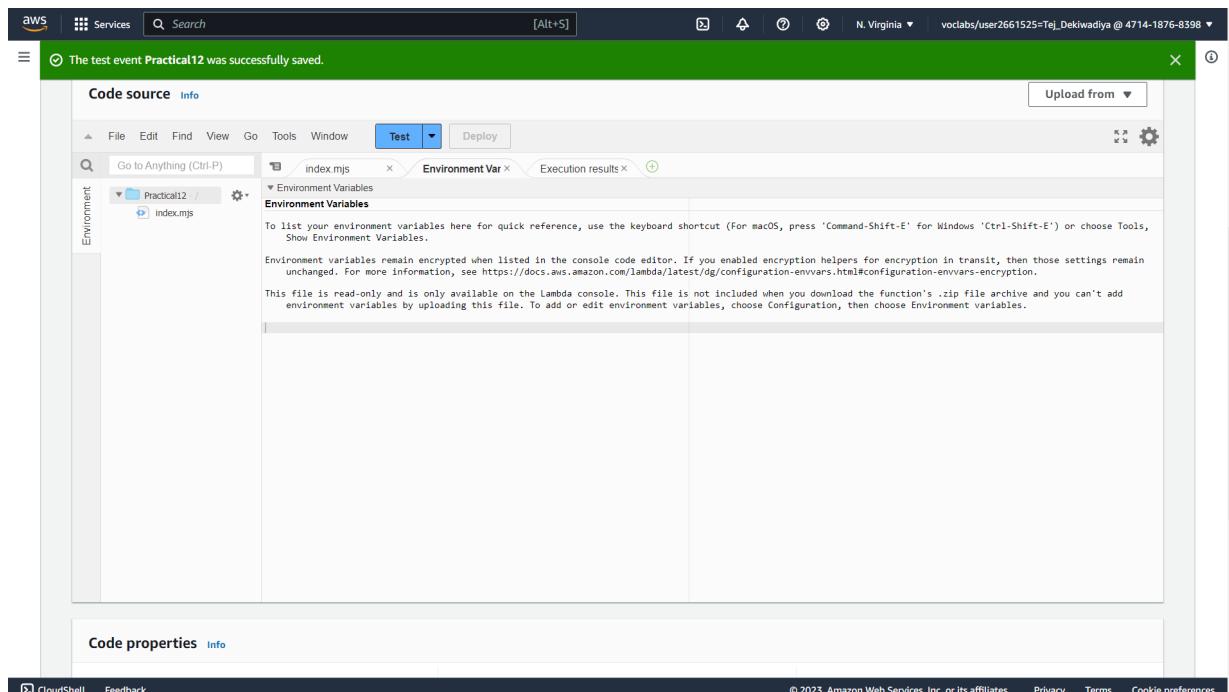
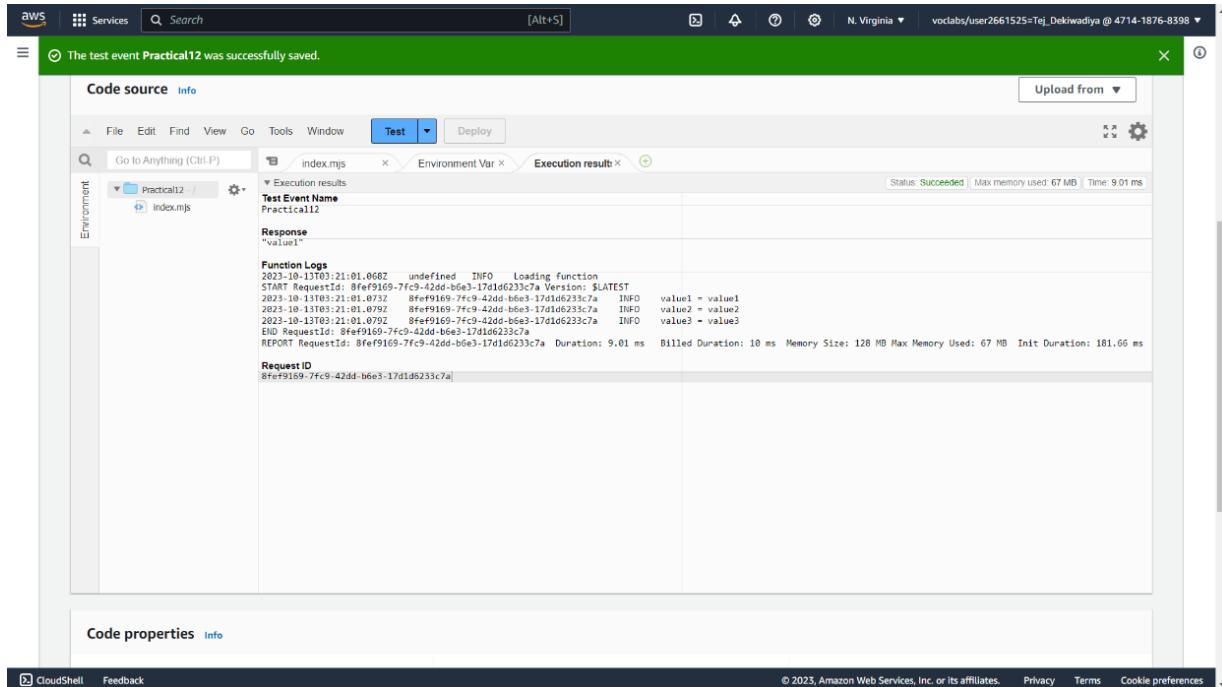
Step 05 : Create a Configure Test Event

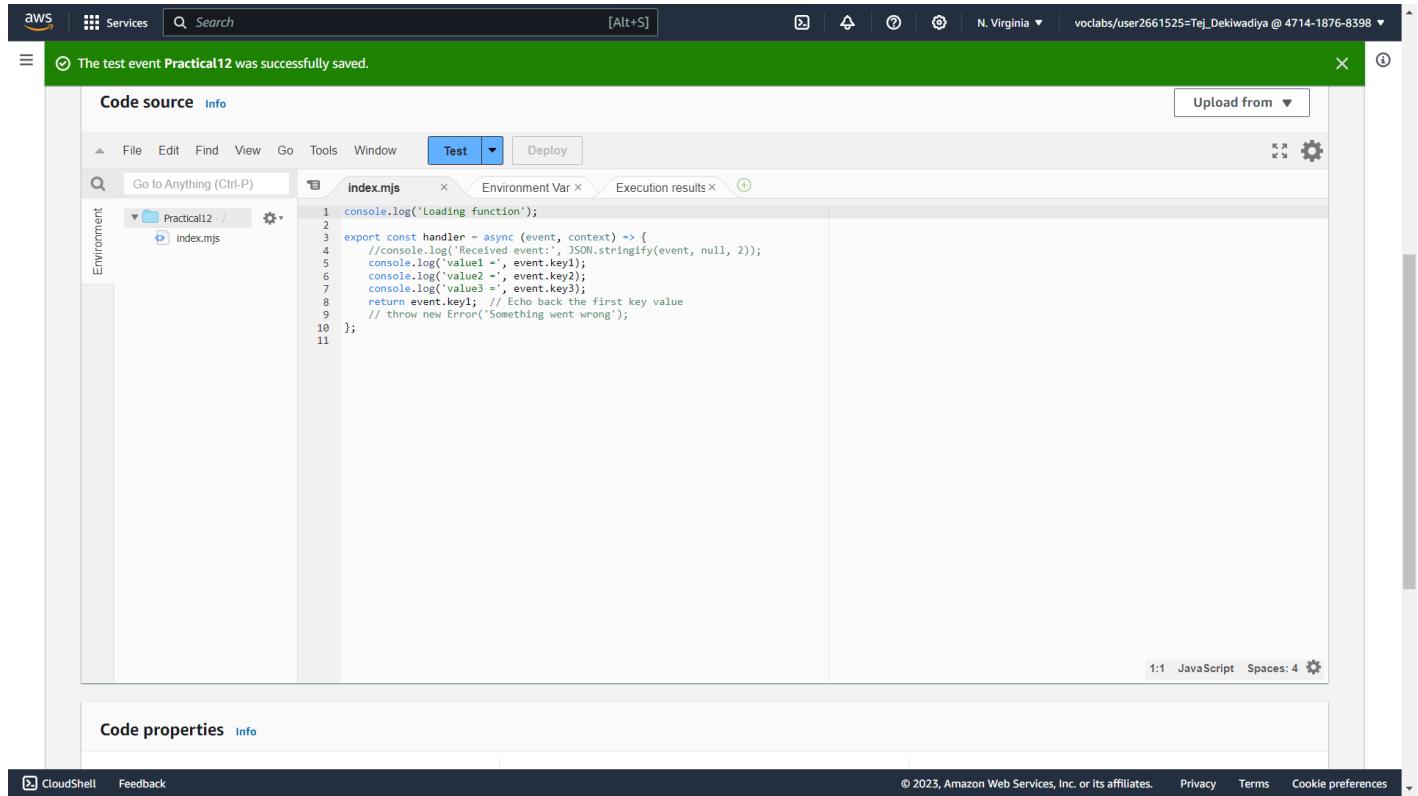
Snapshot :



Step 06 : Click on Test to check config.

Snapshot :





The test event Practical12 was successfully saved.

Code source [Info](#)

File Edit Find View Go Tools Window **Test** Deploy

Go to Anything (Ctrl+P)

index.mjs Environment Var Execution results

```
1 console.log('Loading function');
2
3 export const handler = async (event, context) => {
4     //console.log('Received event:', JSON.stringify(event, null, 2));
5     console.log('value1 -', event.key1);
6     console.log('value2 -', event.key2);
7     console.log('value3 -', event.key3);
8     return event.key1; // Echo back the first key value
9     // throw new Error('Something went wrong');
10 };
11
```

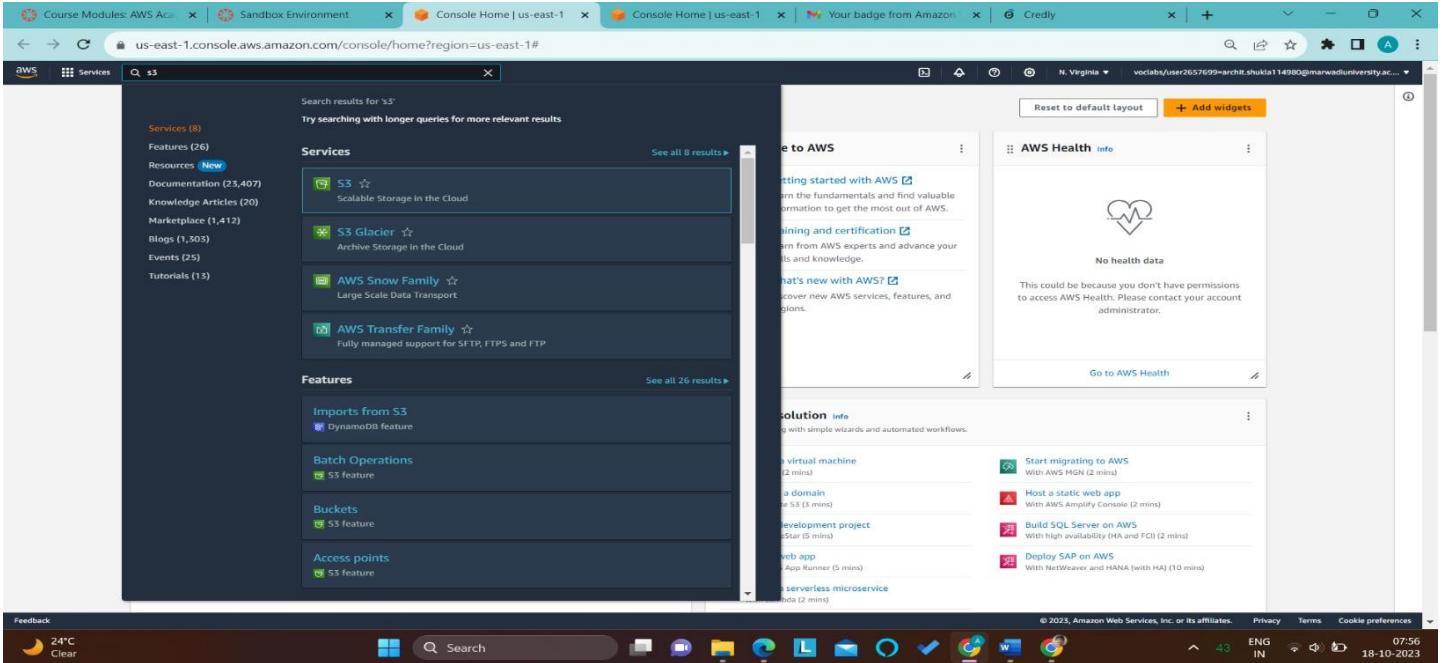
Code properties [Info](#)

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

1:1 JavaScript Spaces: 4

Practical 13: Implementing a Serverless Architecture with AWS Managed Step

1: Search for S3 in services.

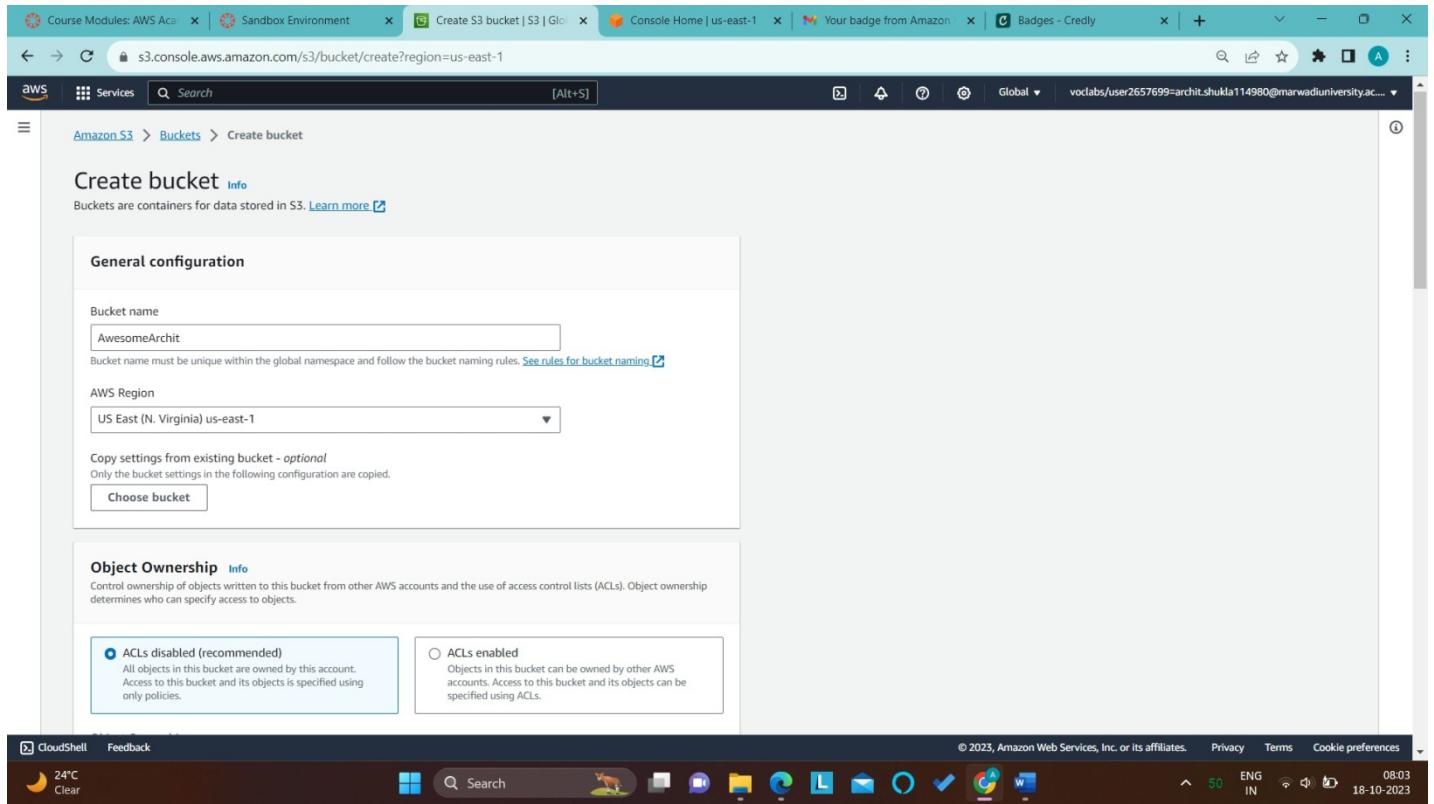


The screenshot shows the AWS Management Console search interface. The search bar at the top contains the query 's3'. Below the search bar, there are two main sections: 'Services' and 'Features'.

- Services:**
 - S3** (Scalable Storage in the Cloud)
 - S3 Glacier** (Archive Storage in the Cloud)
 - AWS Snow Family** (Large Scale Data Transport)
 - AWS Transfer Family** (Fully managed support for SFTP, FTPS and FTP)
- Features:**
 - Imports from S3** (DynamoDB feature)
 - Batch Operations** (S3 feature)
 - Buckets** (S3 feature)
 - Access points** (S3 feature)

On the right side of the search results, there is a sidebar titled 'Welcome to AWS' which includes links like 'Getting started with AWS', 'Training and certification', and 'What's new with AWS?'. Below the sidebar is a section titled 'AWS Health' with a message stating 'No health data'.

Step 2: Then write down bucket name and uncheck block public access and at last click on create bucket so bucket created.



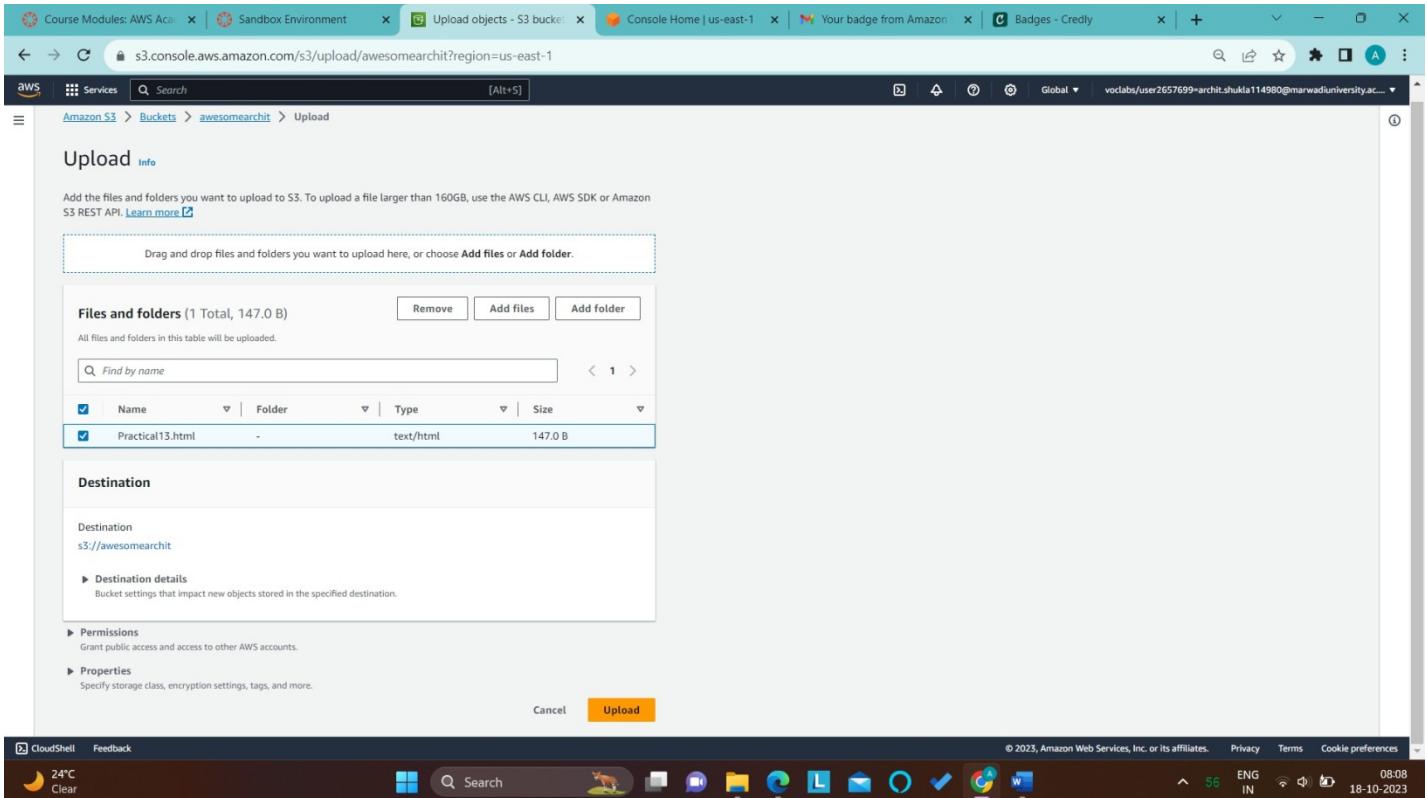
The screenshot shows the 'Create bucket' wizard in the AWS S3 service. The first step, 'General configuration', is displayed. The user has entered 'AwesomeArchit' as the bucket name and selected 'US East (N. Virginia) us-east-1' as the region. The 'ACLs disabled (recommended)' option is selected under 'Object Ownership'.

General configuration

- Bucket name:** AwesomeArchit
- AWS Region:** US East (N. Virginia) us-east-1
- Object Ownership:**
 - ACLs disabled (recommended)**: All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
 - ACLs enabled**: Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

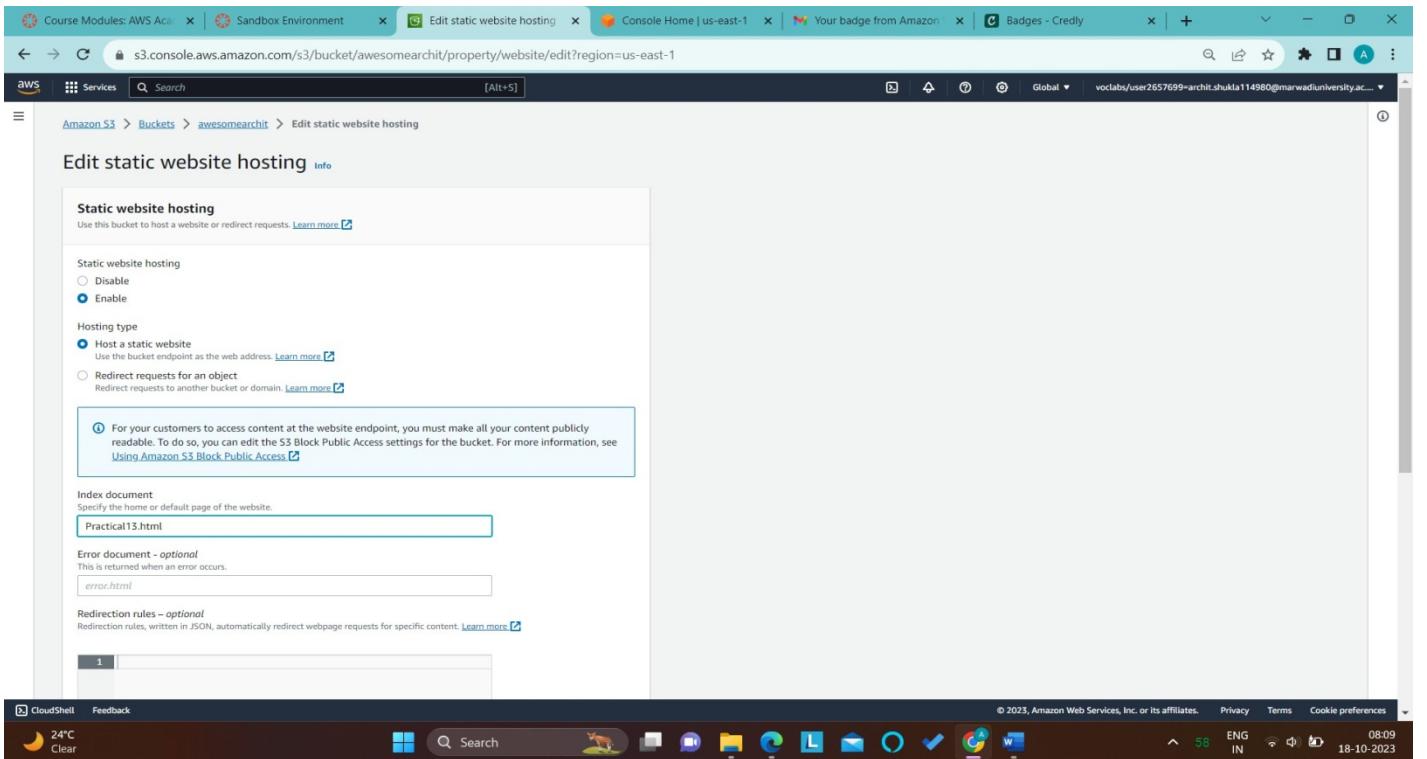
At the bottom of the wizard, there is a 'Create bucket' button.

Step3: Now make a one html static page. After the click on bucket id open object homepage here click on upload and upload your html file that you made.



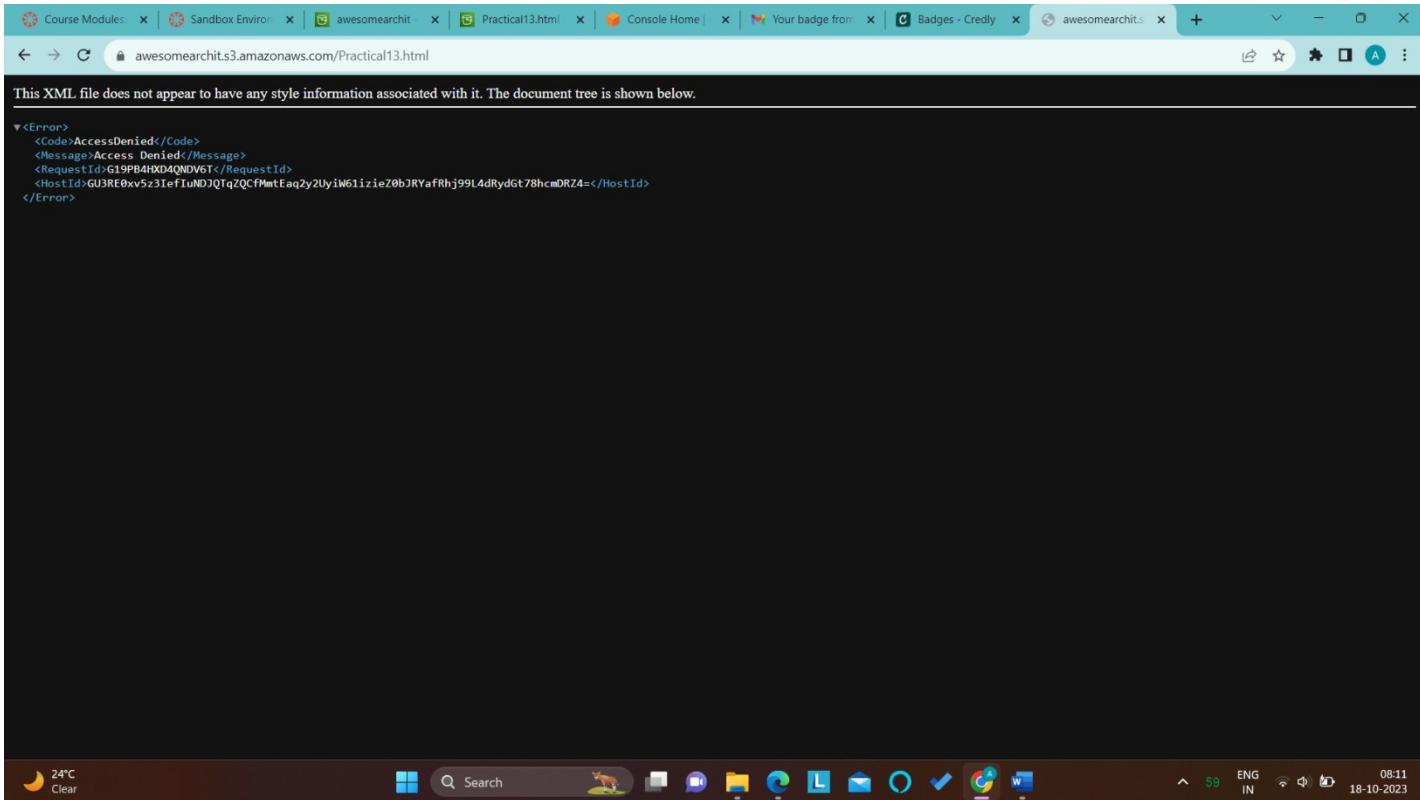
The screenshot shows the AWS S3 'Upload' interface. A file named 'Practical13.html' (147.0 B, text/html) is selected for upload. The destination is set to 's3://awesomearchit'. The 'Upload' button is highlighted in orange at the bottom right.

Step4: Next in properties go to Static Website hosting and here rewrite down your html file name in Object Index and then save that settings.

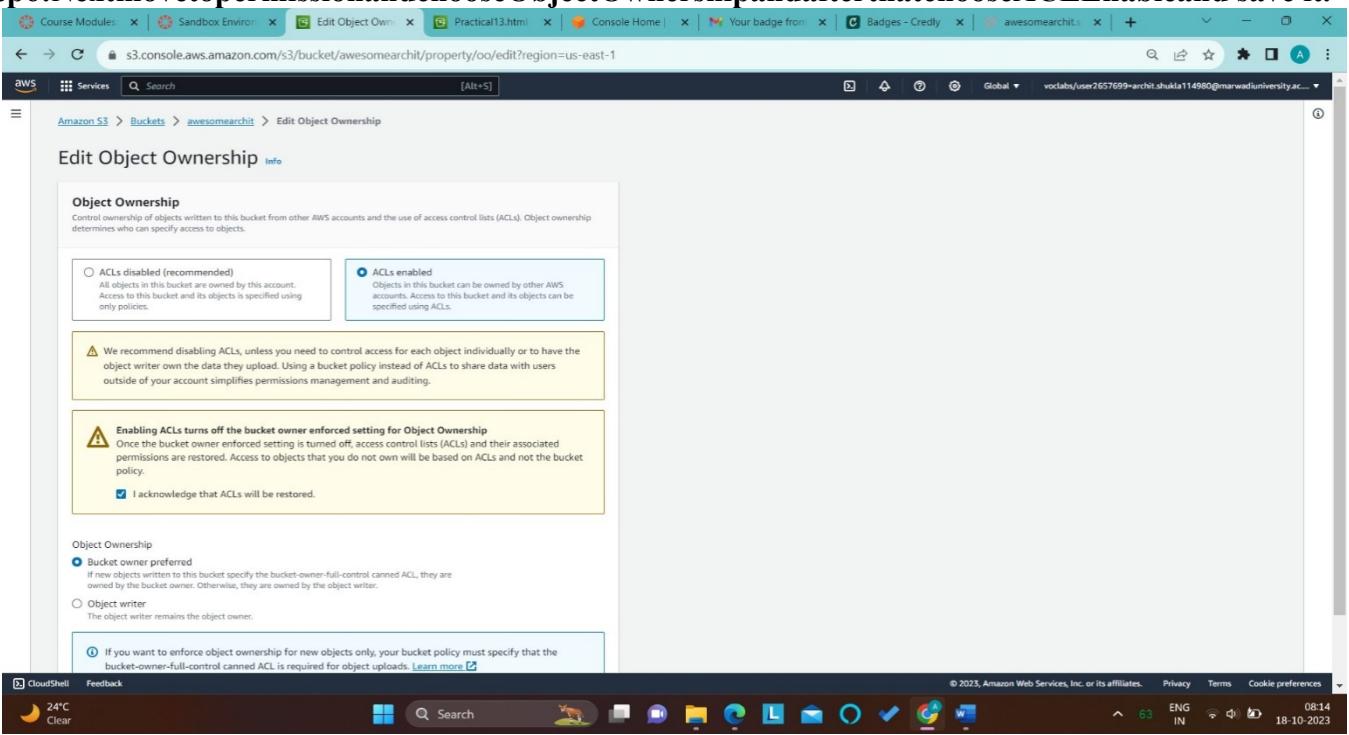


The screenshot shows the 'Edit static website hosting' configuration for the 'awesomearchit' bucket. Under 'Static website hosting', 'Enable' is selected. Under 'Hosting type', 'Host a static website' is selected. In the 'Index document' section, 'Practical13.html' is specified as the index document. The 'Error document - optional' field contains 'error.html'. The 'Redirection rules - optional' section is present but empty. The 'Save' button is visible at the bottom left.

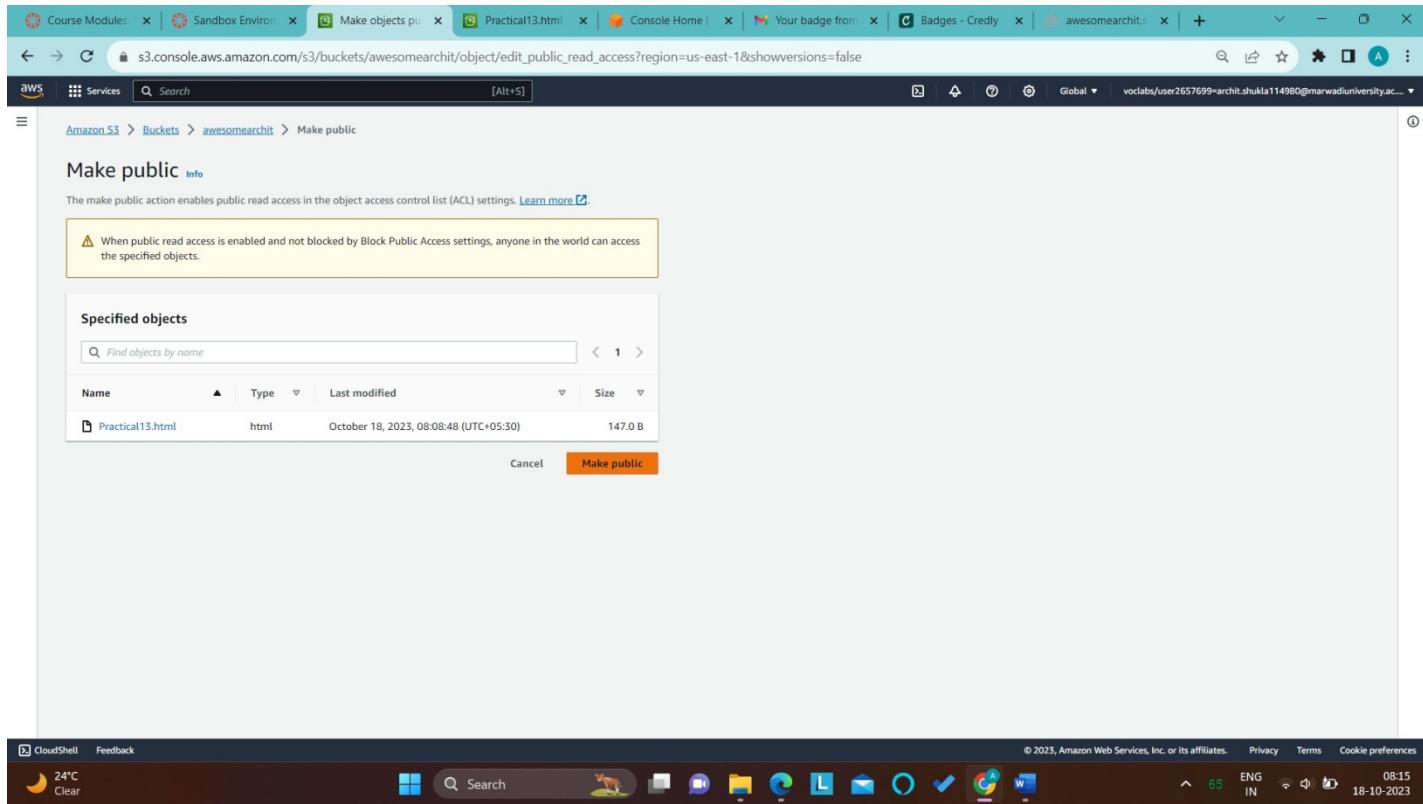
Step 5: Now in a object choose your html file and open that so you are able to see that object url and open it in a new web page. So currently that file is not working.



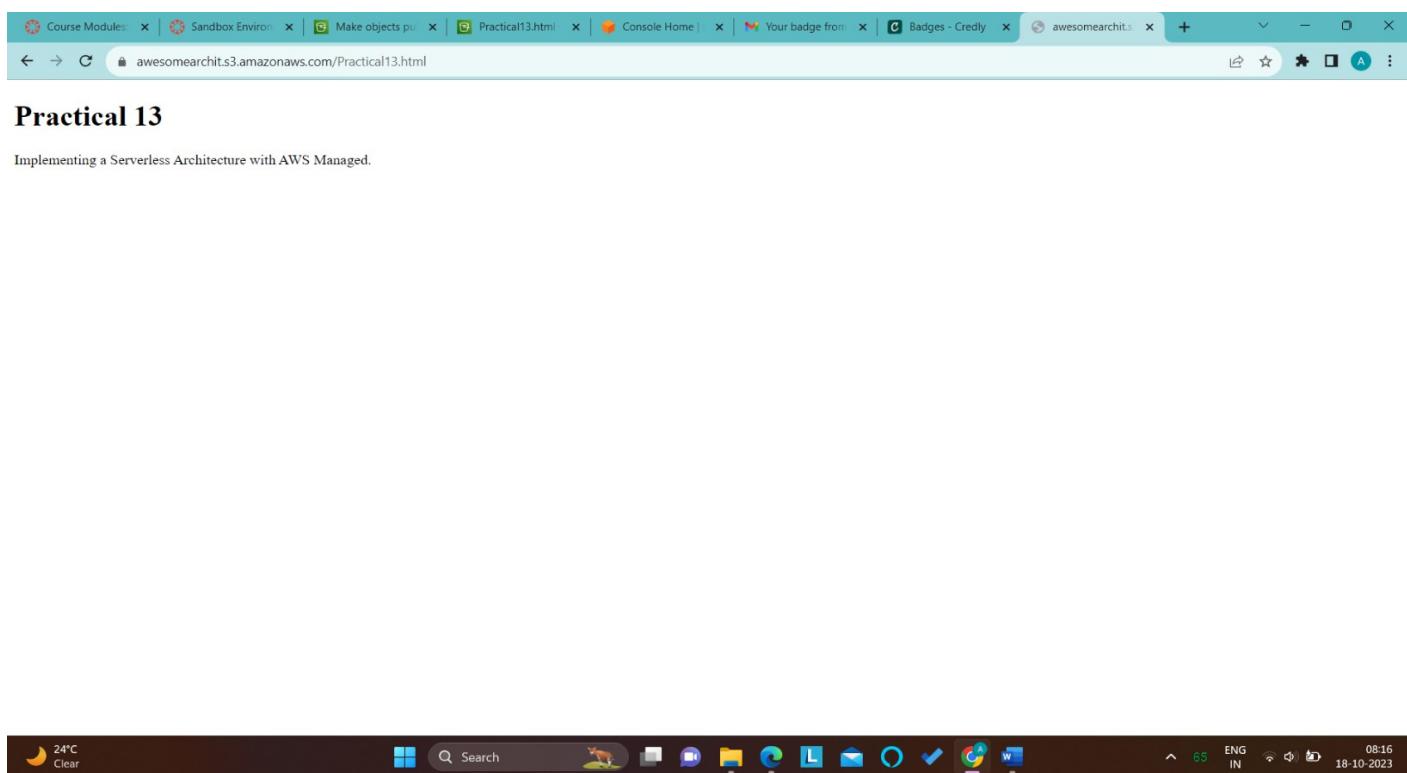
Step 6: Next move to permission and choose Object Ownership and after that choose ACL Enable and save it.



Step 7: Now select your object and from Action choose Make public using ACL and save default setting.

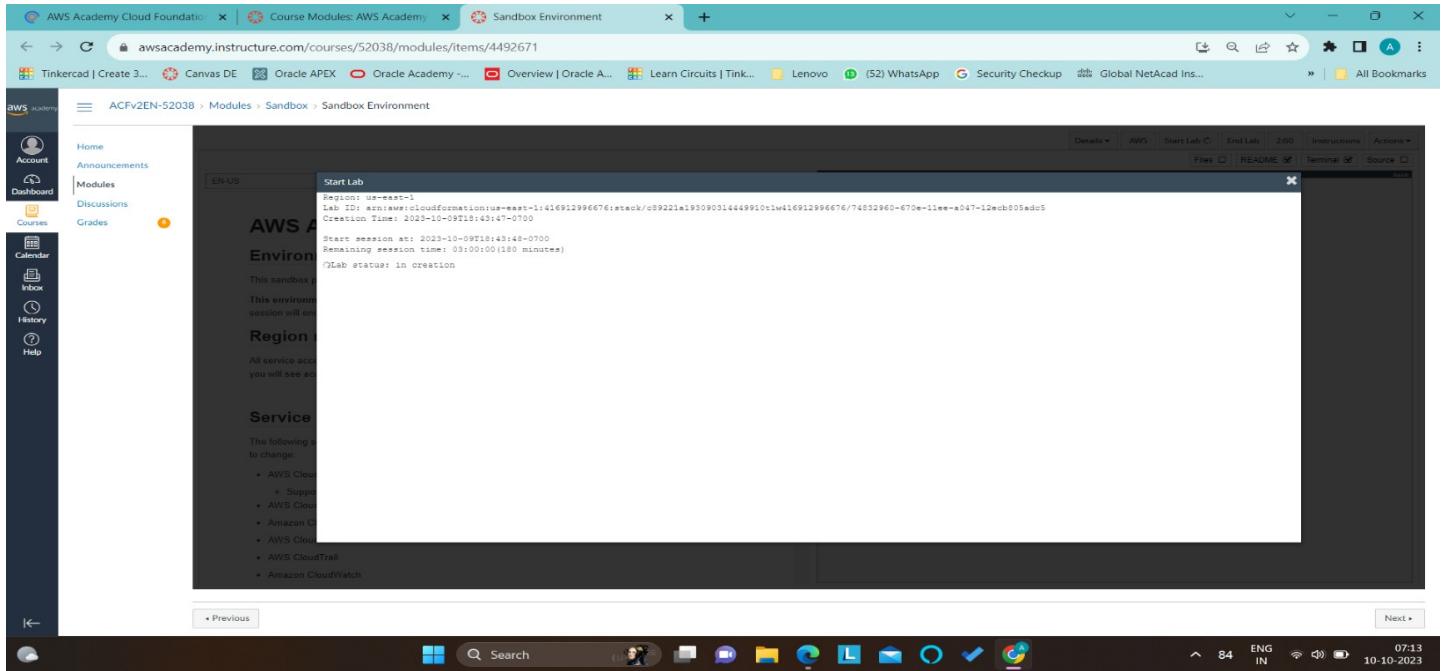


Step 8: Now again refresh your website URL so now it's working.

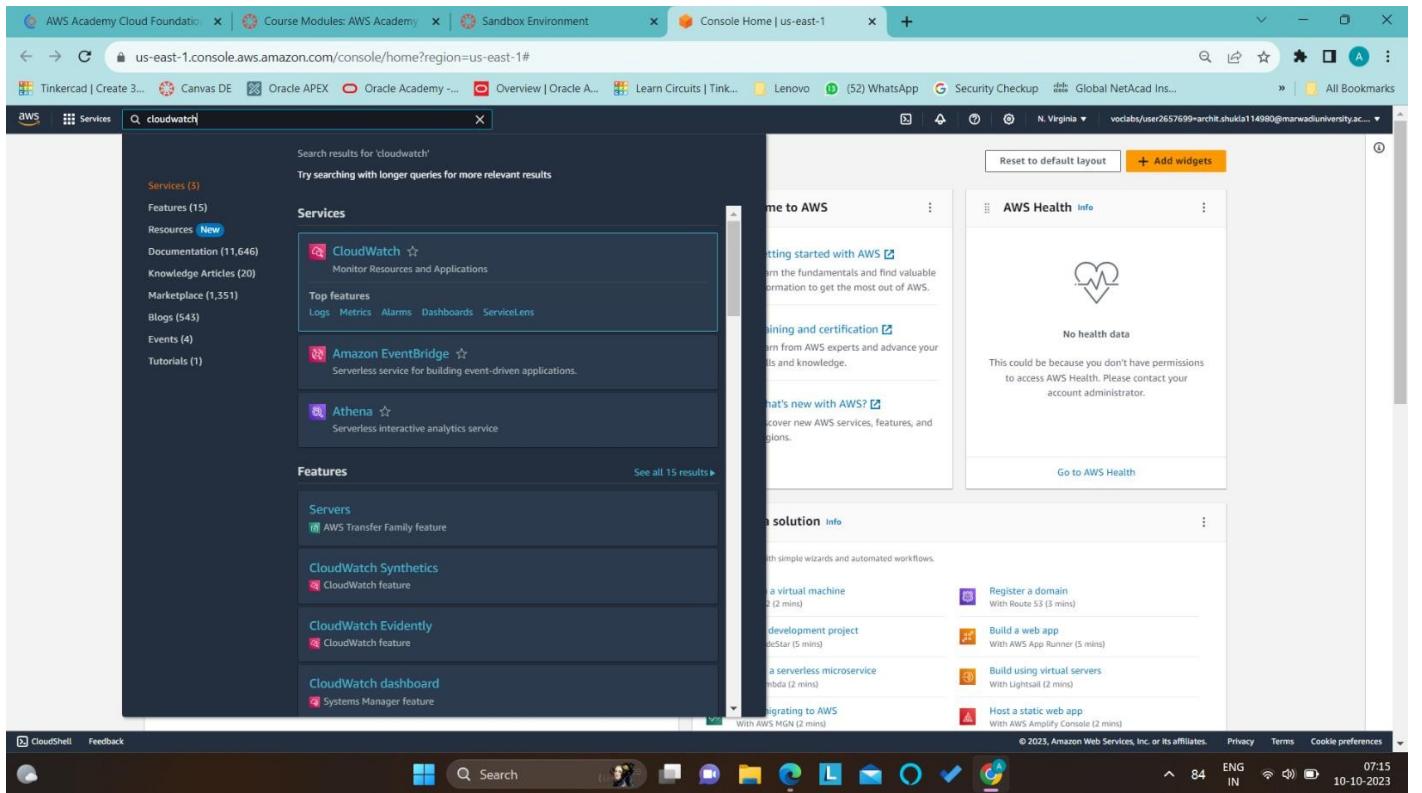


Practical 14: Cloud monitoring and management tools -> AWS CloudWatch

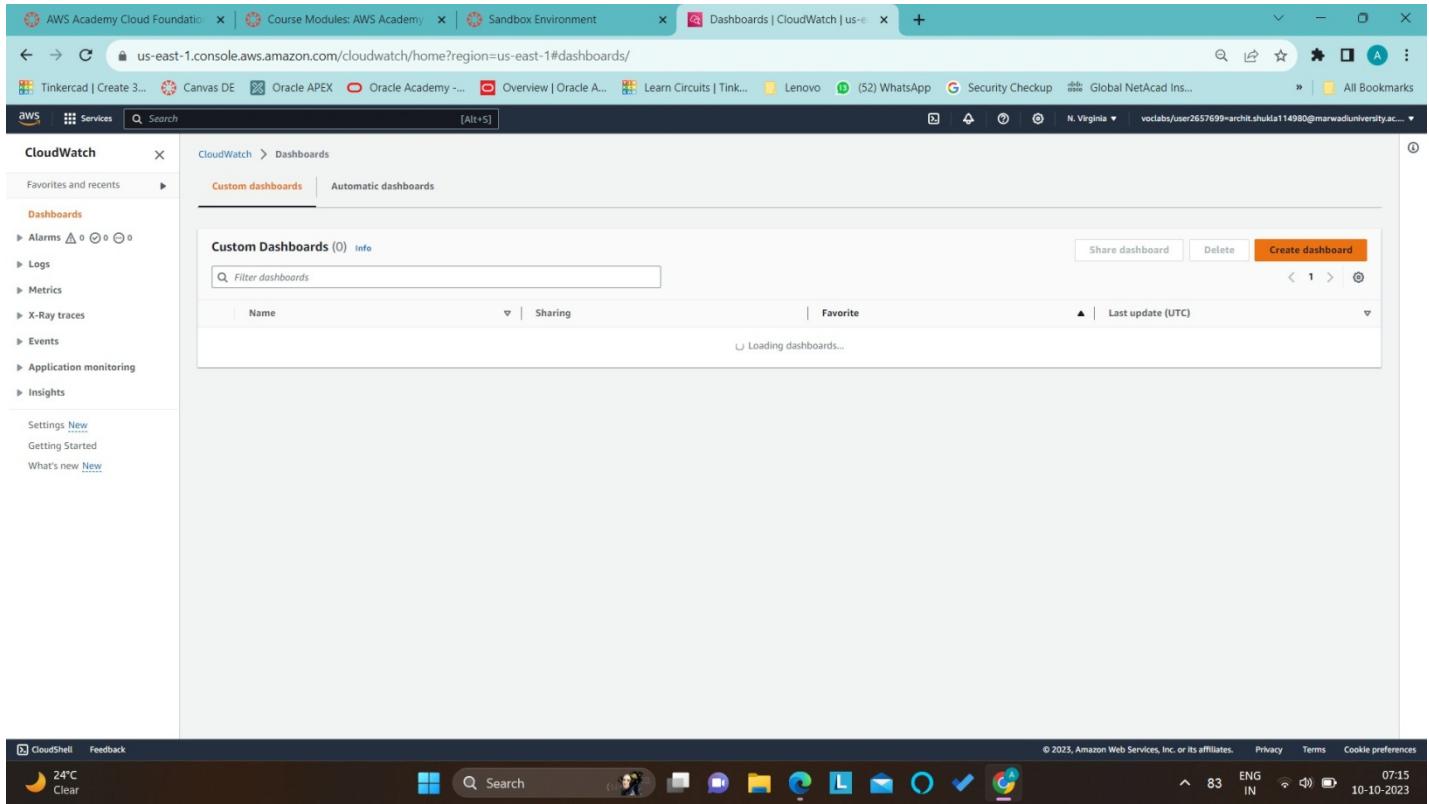
Step 1: Click on start lab.



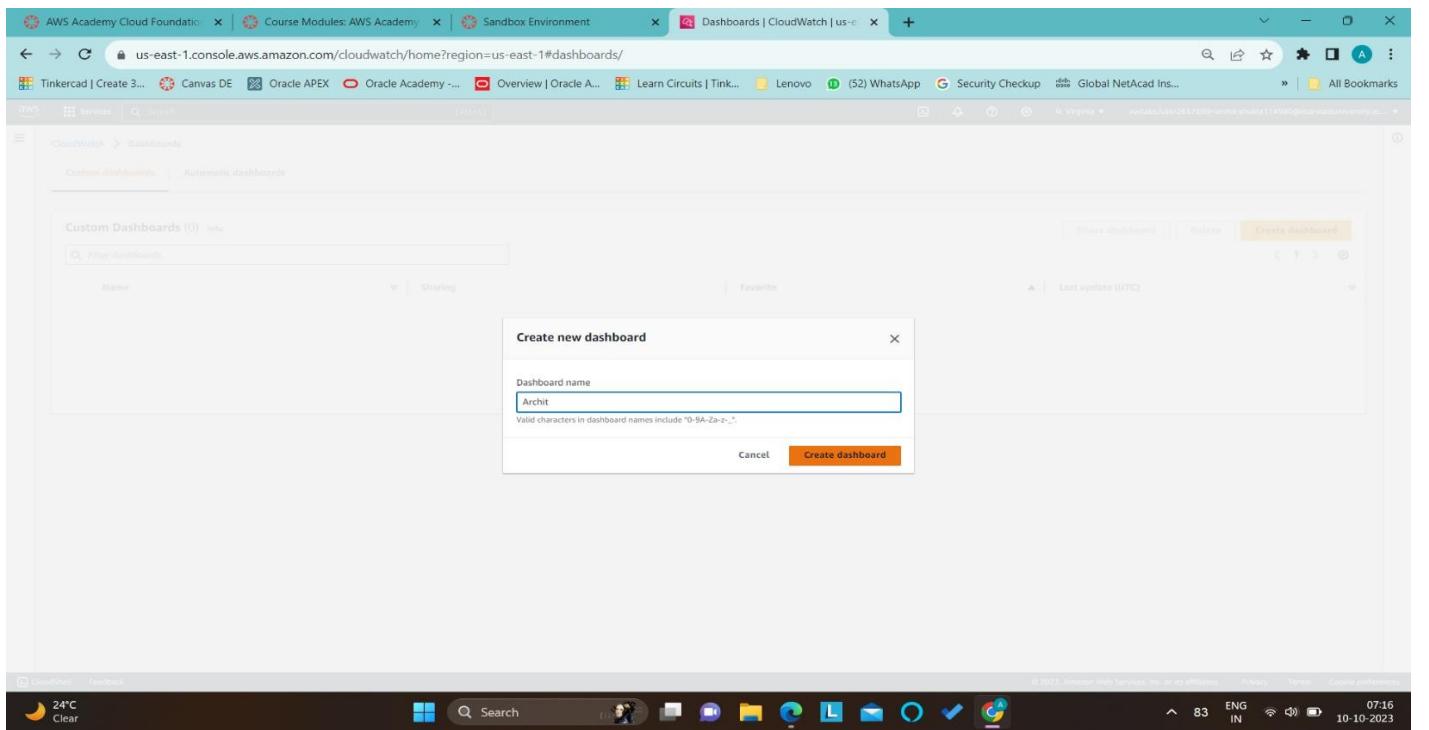
Step 2: Click on services and in that click on CloudWatch.



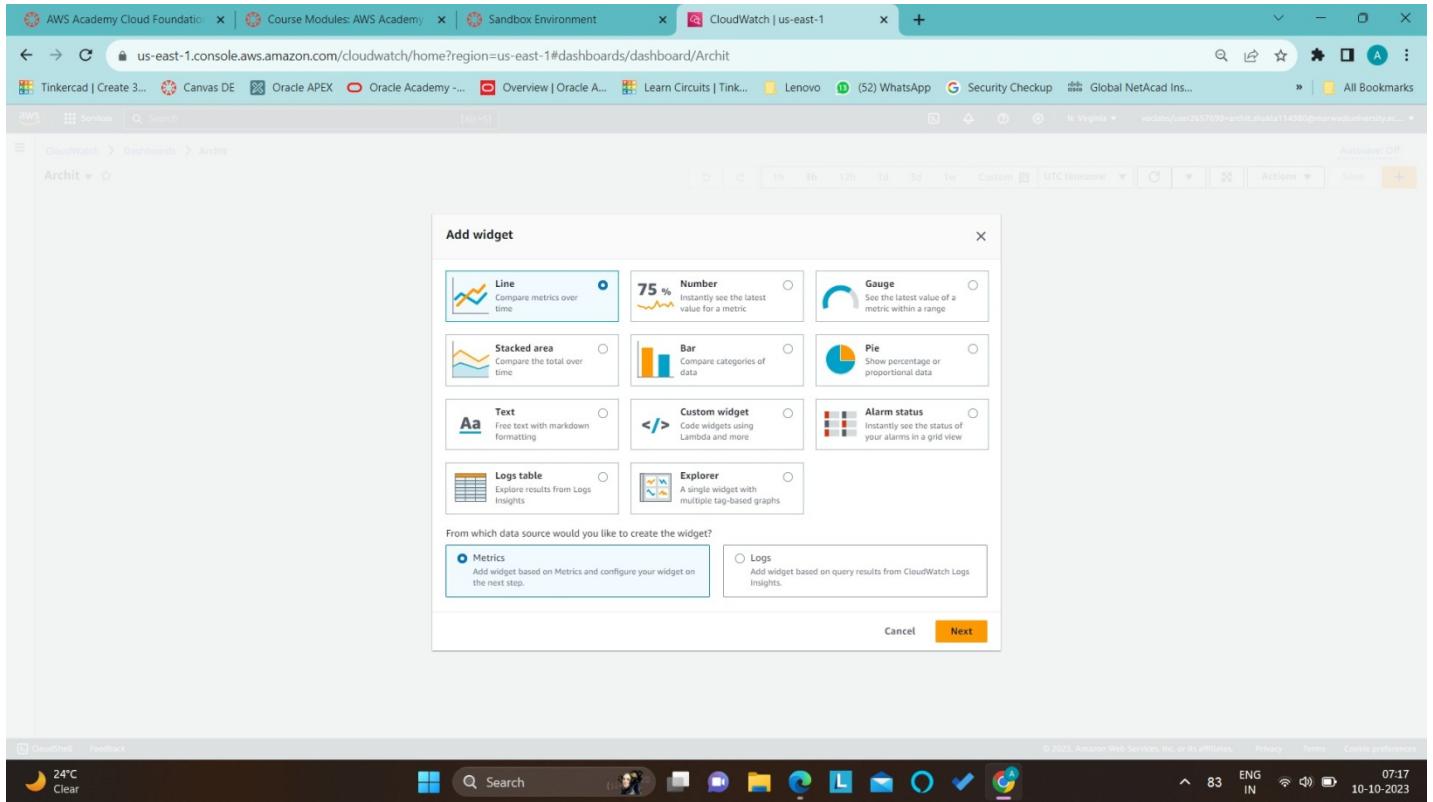
Step 3: Click on Cloudwatch dashboards and click on create dashboard.



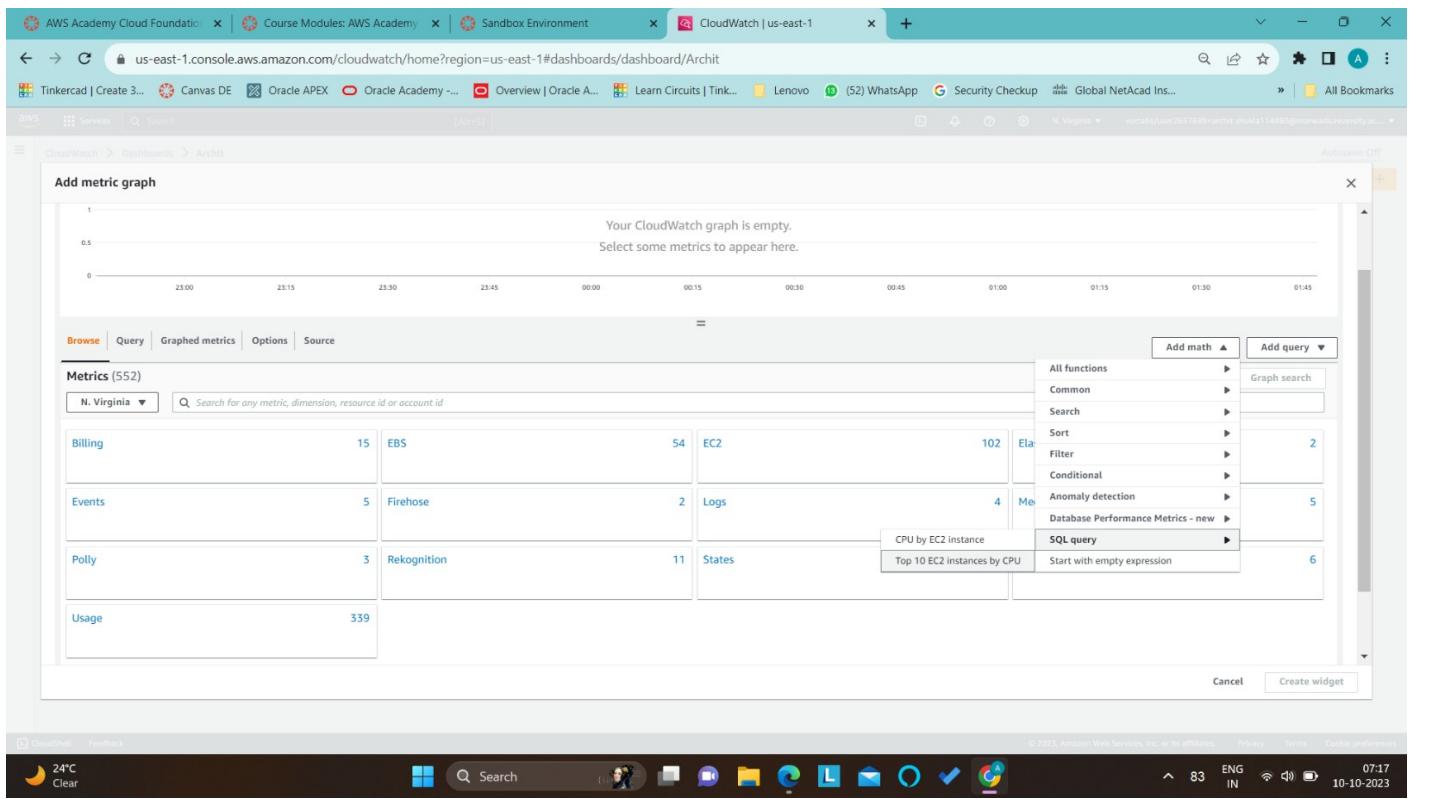
Step 4: Now, Create new dashboard and give name to Archit, and click on create dashboard.



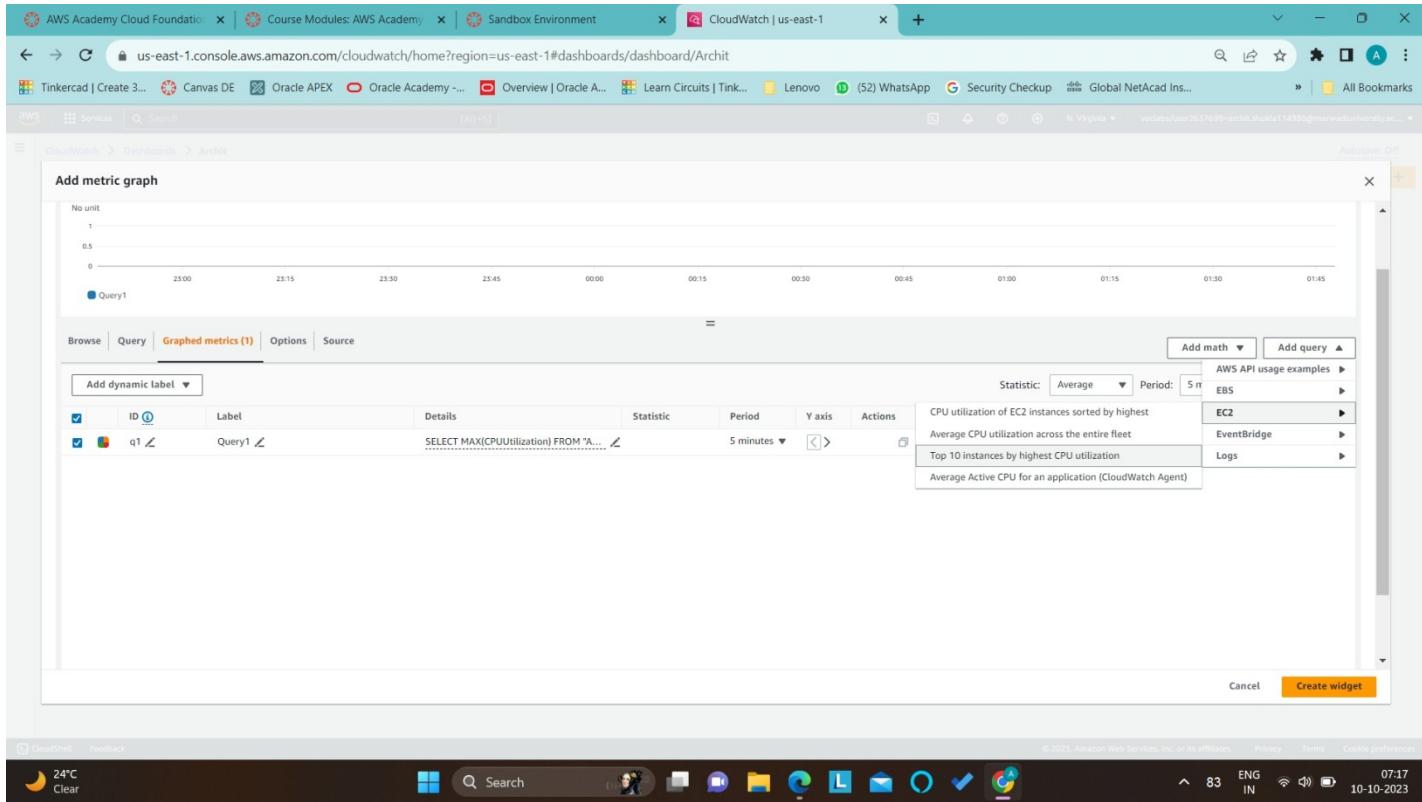
Step 5: Now, click on Line and add widget and click next.



Step 6: Now click on add math and select SQL query and select Top 10 EC2 instances by CPU.

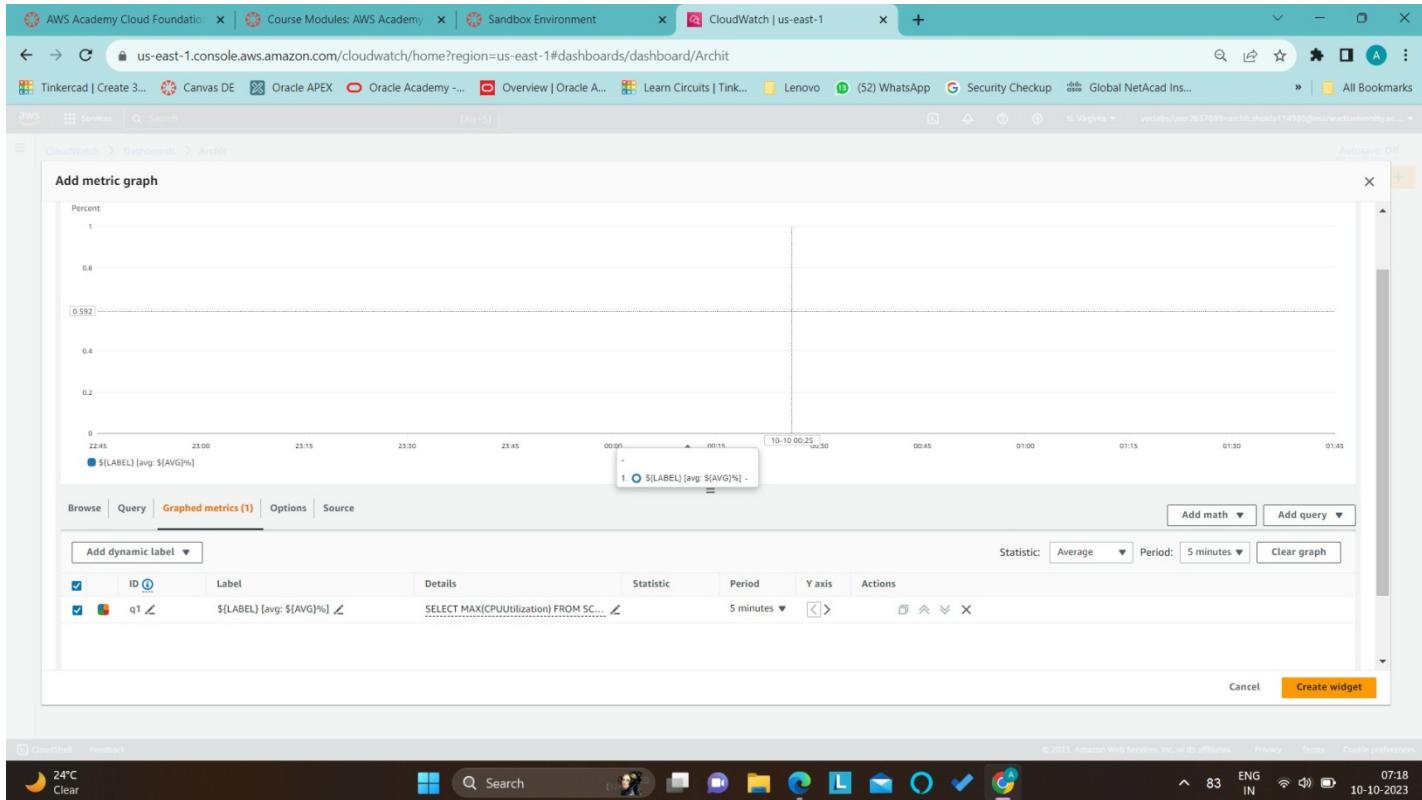


Step 7: Now click on Add query and select EC2 and select Top 10 instances by highest CPU utilization.



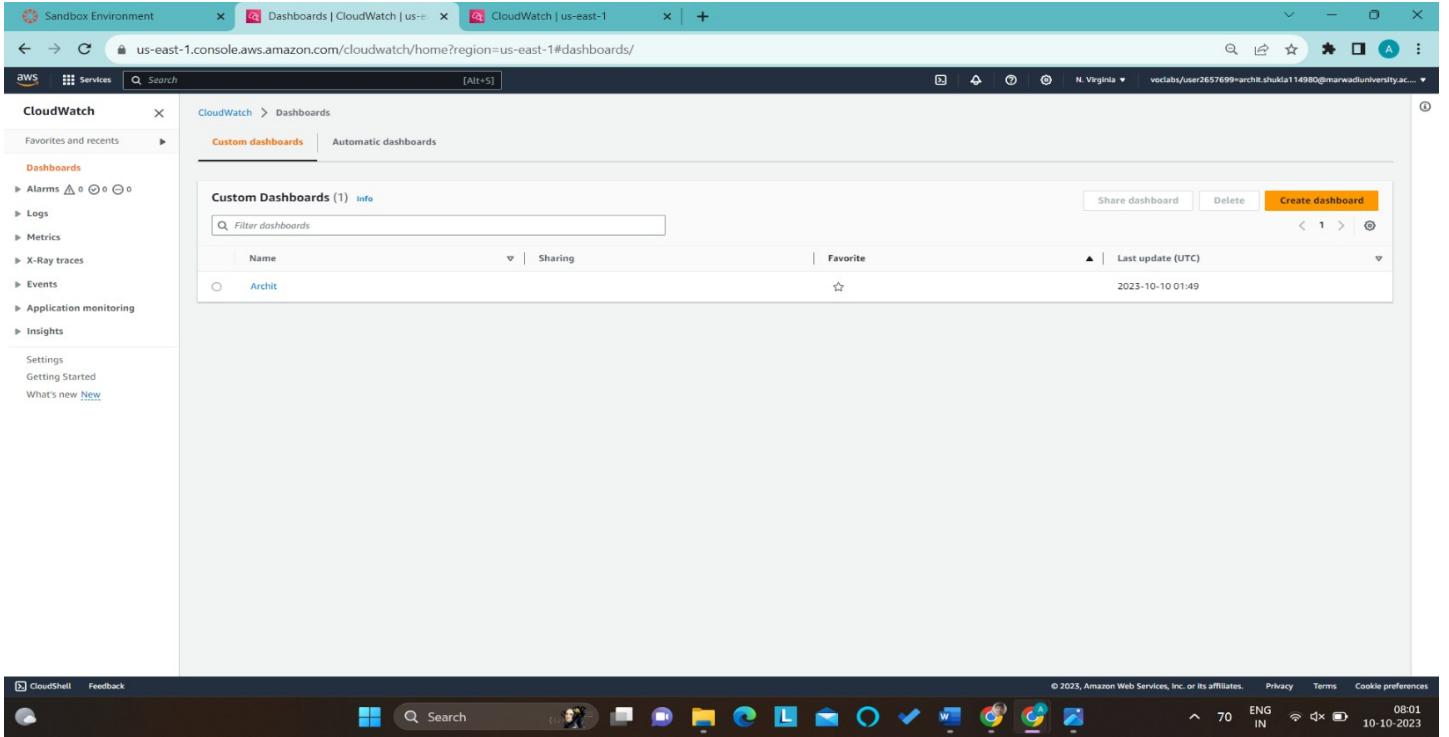
The screenshot shows the AWS CloudWatch Metrics console with the 'Add metric graph' interface. A dropdown menu on the right is open, showing 'EC2' selected under 'AWS API usage examples'. Other options like 'Logs' and 'EventBridge' are also visible.

Step 8: Now in Graphed metrics click on create widget.



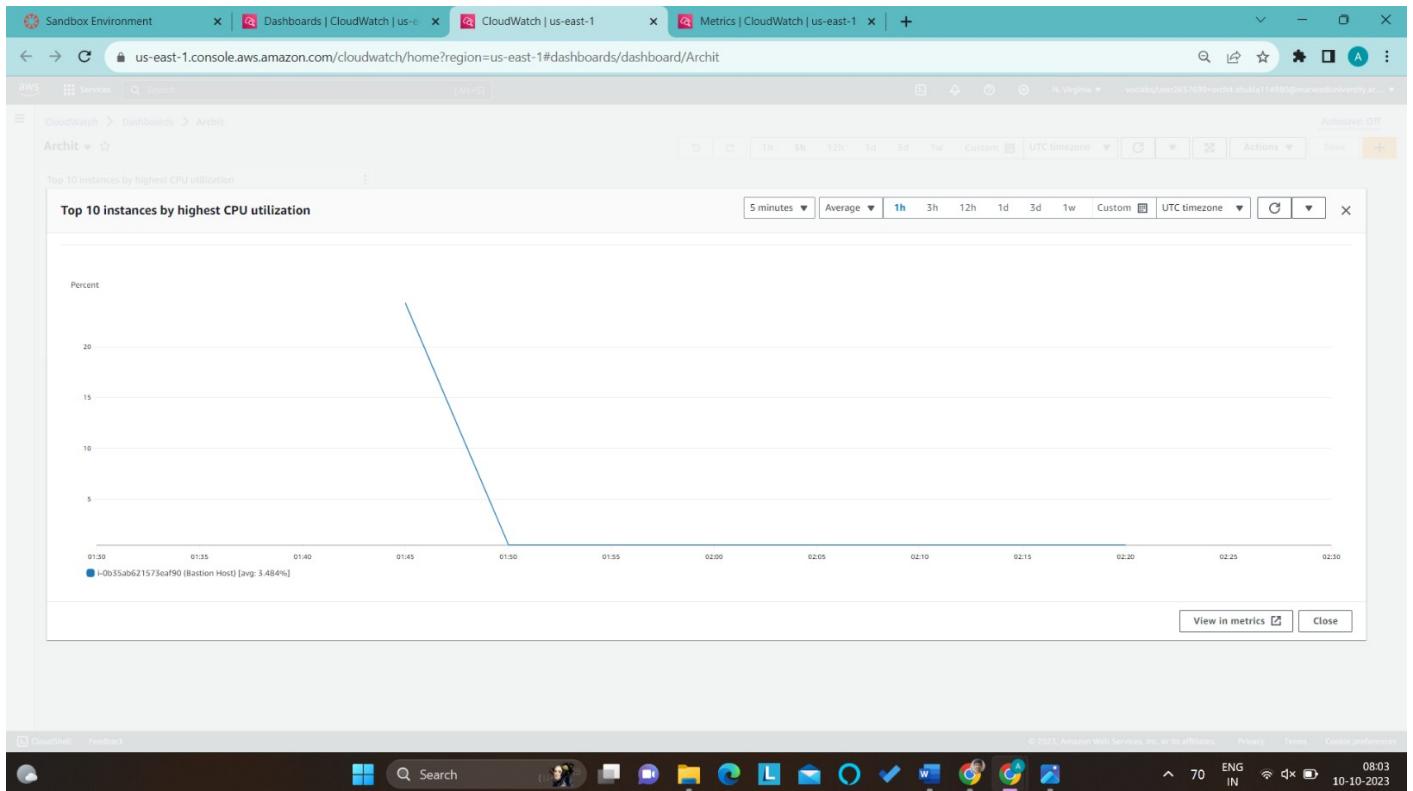
The screenshot shows the AWS CloudWatch Metrics console with the 'Graphed metrics (1)' tab selected. A 'Create widget' button is highlighted at the bottom right of the interface.

Step 9: Now click on Archit, the dashboard that we have created.

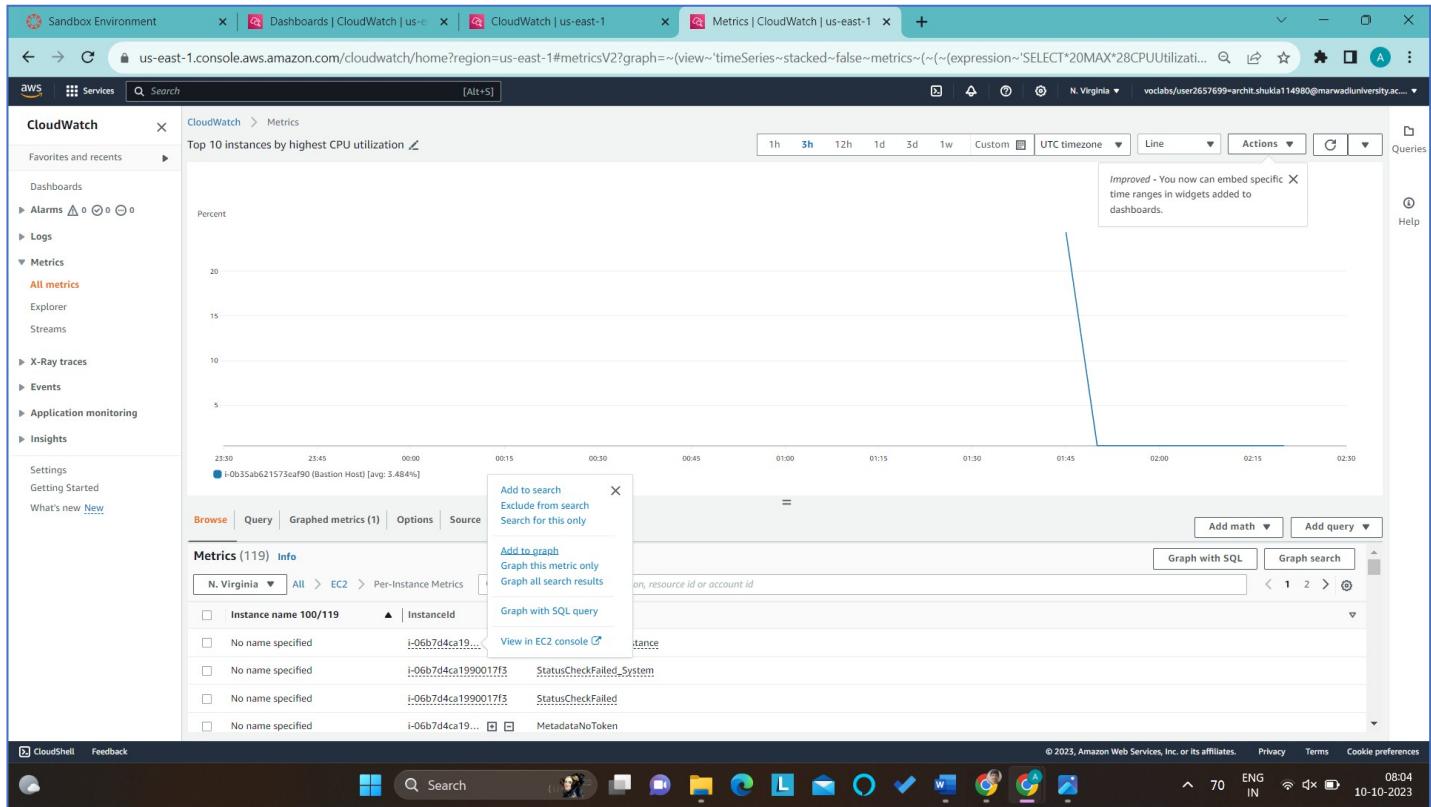


The screenshot shows the AWS CloudWatch Metrics interface. On the left, there's a sidebar with navigation links like 'CloudWatch', 'Dashboards', 'Logs', 'Metrics', etc. The main area displays a table titled 'Custom Dashboards (1)'. It lists one dashboard named 'Archit', which was last updated on 2023-10-10 01:49. There are buttons for 'Share dashboard', 'Delete', and 'Create dashboard' at the top right of the table. Below the table, there's a large empty space where the dashboard would be displayed.

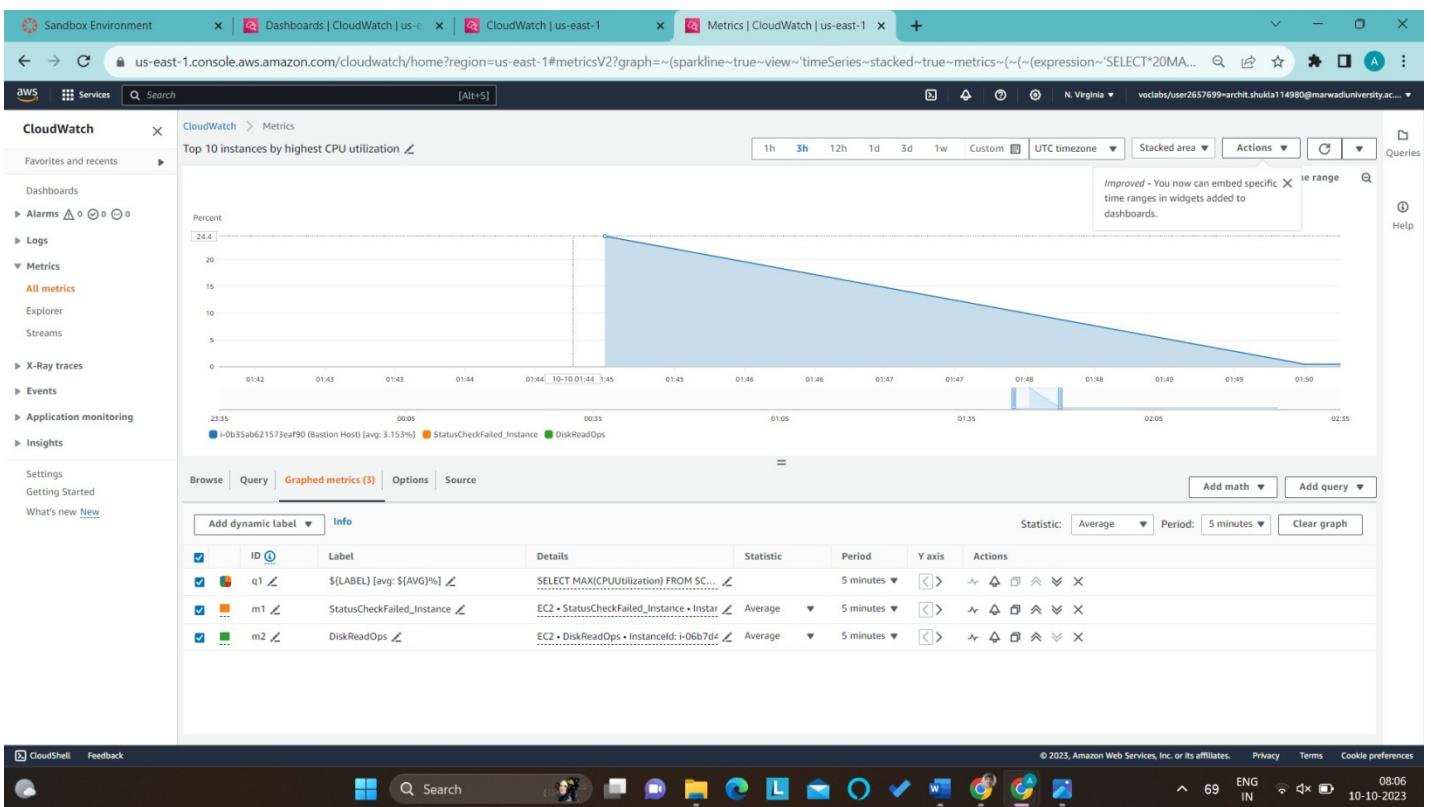
Step 10: Now we can see this utilization of the CPU, then click on view in metrics



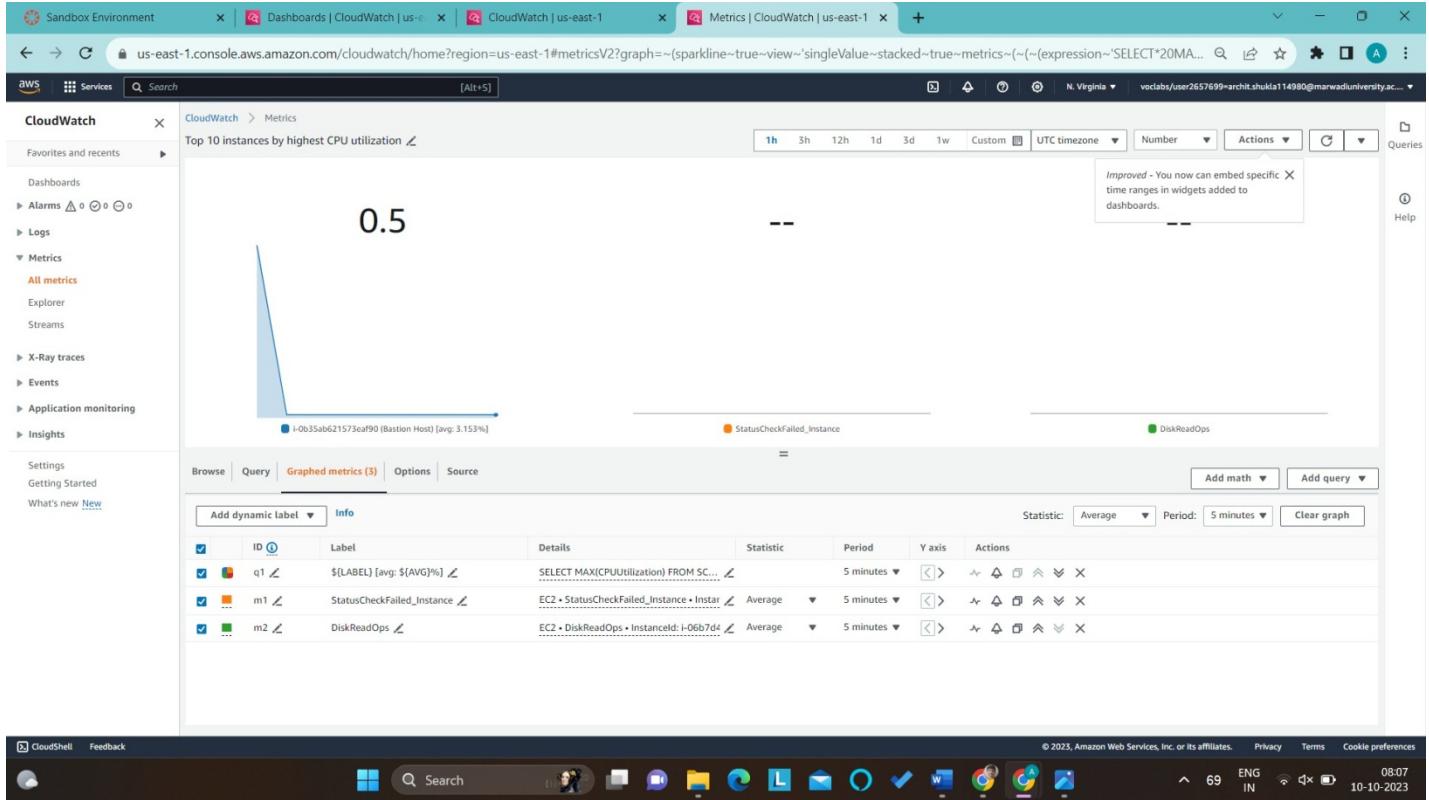
Step 11: Now click on the instance what we need to monitor.



Step 12: Now we can see the CPU utilization of that instance.



Step 13: Click on Number and we can see the CPU utilization of that instance in number format and click on Graphed metrics.



Step 14: Click on create widget & Save the instance. Now we can monitor our resources for the selected instance.

