

Course: DevOps

Name: ch.Lakshmi Priyanka

Module: AWS and VPC

Batch no: 115

Topic: VPC, Subnets, and Internet Gateway

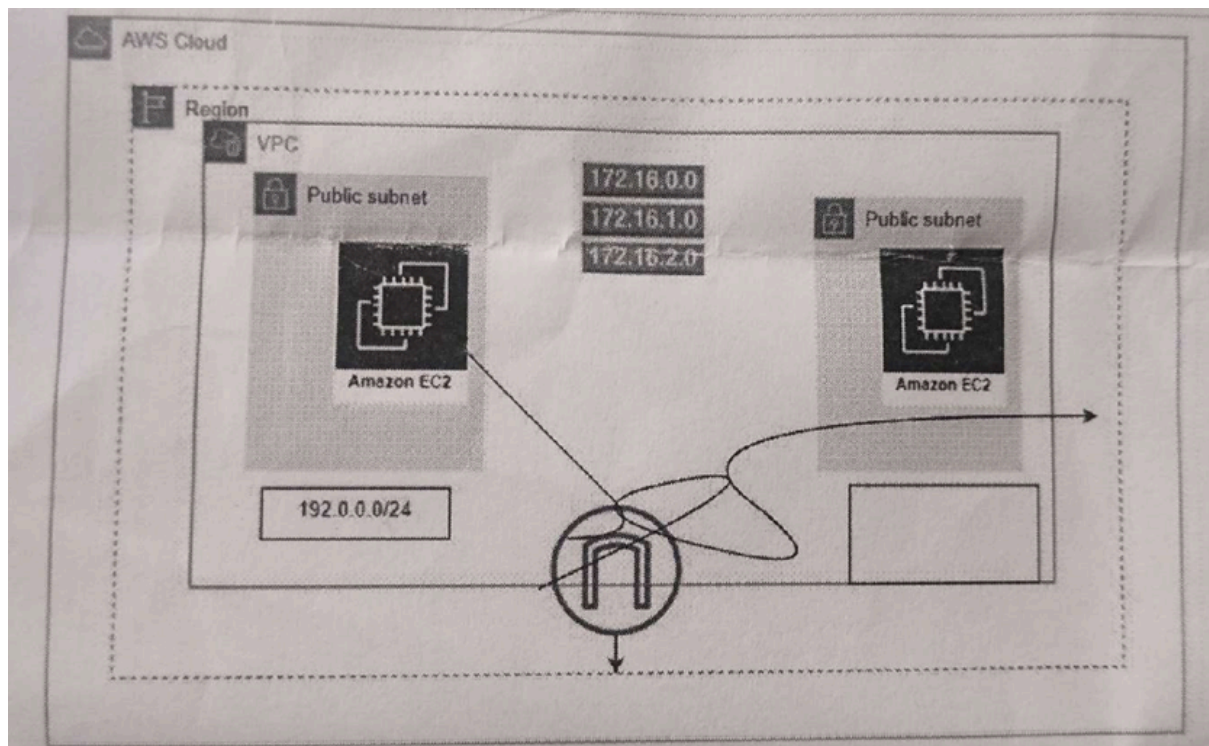
Assignment no: 07

Trainer Name: Mr. Madhukar sir

Date of submission: 4 – Feb – 2024

Mail-ID: chlakshmipriyanka9@gmail.com

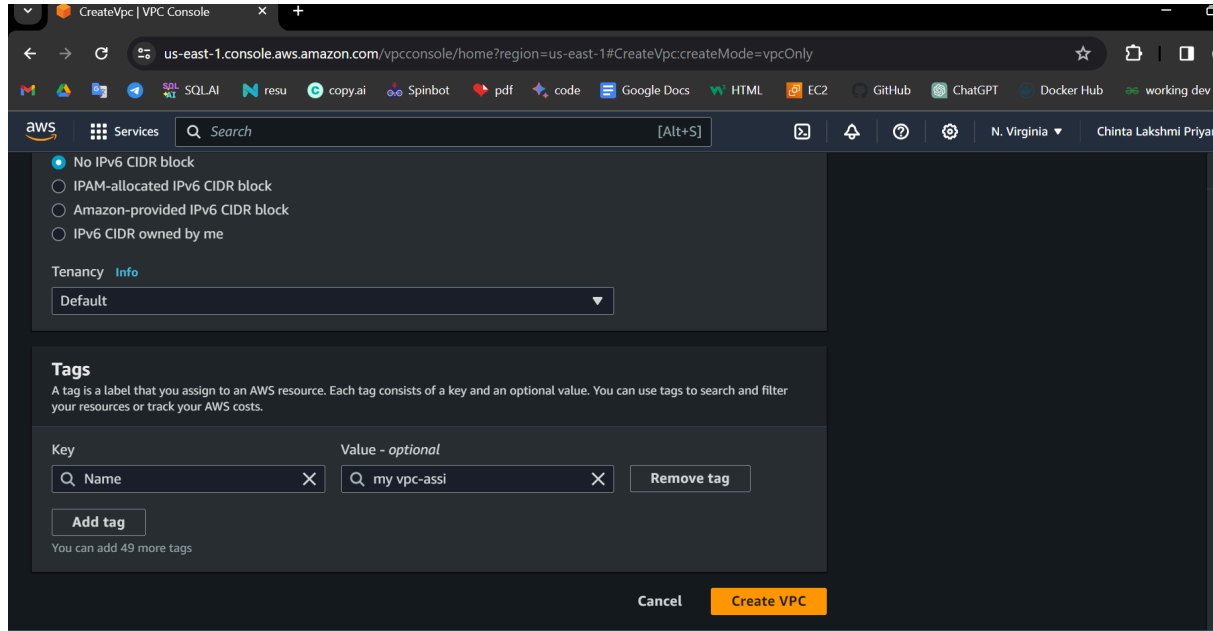
Assignment: Implement the VPC using the given details.



- Creating a VPC
- Creating a Subnets (172.16.0.0)
- ^ Public subnet (172.16.1.0/24)
- ^ Private subnet (172.16.2.0/24)
- Creating a Route table (public, private)
- Creating an Internet gateway
- Creating an Ec2 instance using VPC and Subnets.
- I am creating a network for private subnets using a NAT gateway with Elastic IP.

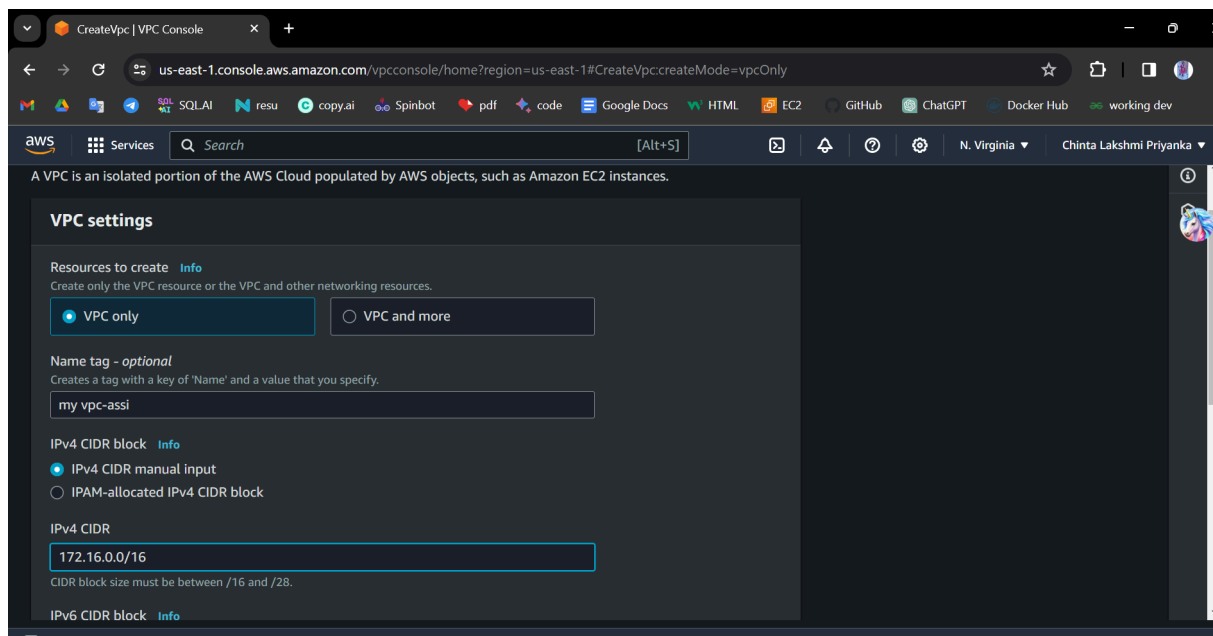
1. Creating a VPC (Virtual Private Cloud):

1. Create a VPC, go to services, and type VPC.

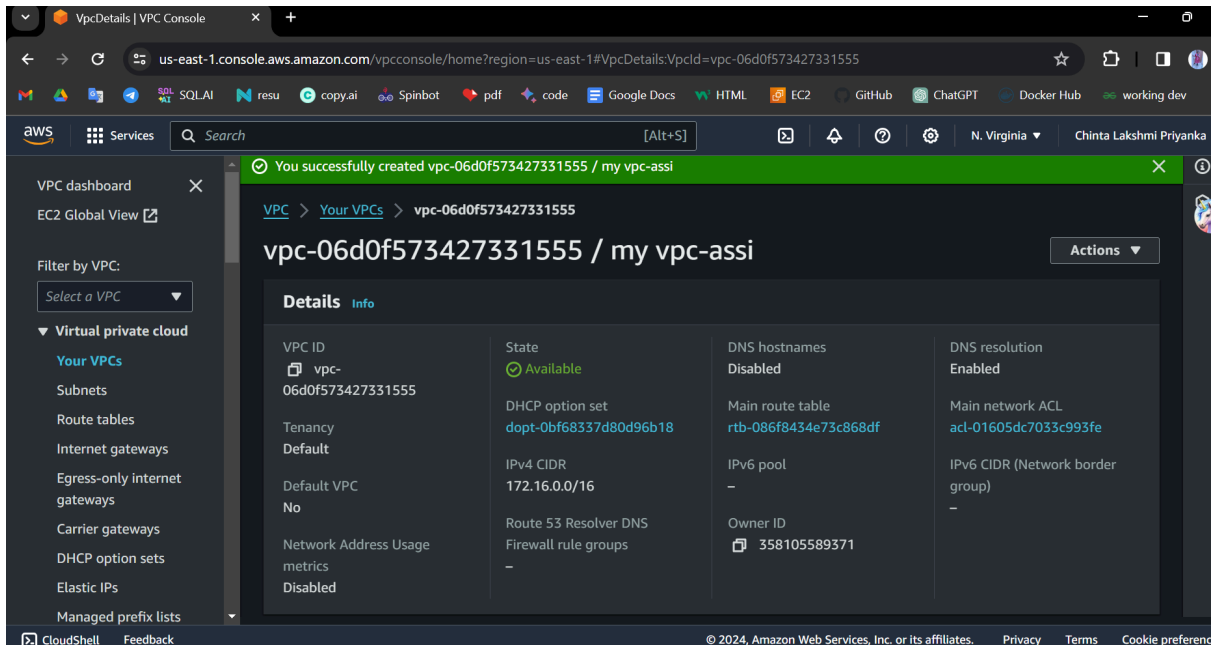


The screenshot shows the 'Create VPC' page in the AWS Management Console. The browser address bar indicates the URL: `us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateVpc:createMode=vpcOnly`. The page has a dark theme. On the left, there are two main sections: 'No IPv6 CIDR block' and 'Tags'. The 'No IPv6 CIDR block' section has four radio button options: 'No IPv6 CIDR block' (selected), 'IPAM-allocated IPv6 CIDR block', 'Amazon-provided IPv6 CIDR block', and 'IPv6 CIDR owned by me'. Below this is a 'Tenancy' dropdown menu set to 'Default'. The 'Tags' section includes a description: '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.' It features a table with columns 'Key' and 'Value - optional'. One tag is already added: 'Name' with value 'my vpc-assi'. There is an 'Add tag' button and a note 'You can add 49 more tags'. At the bottom right, there are 'Cancel' and 'Create VPC' buttons.

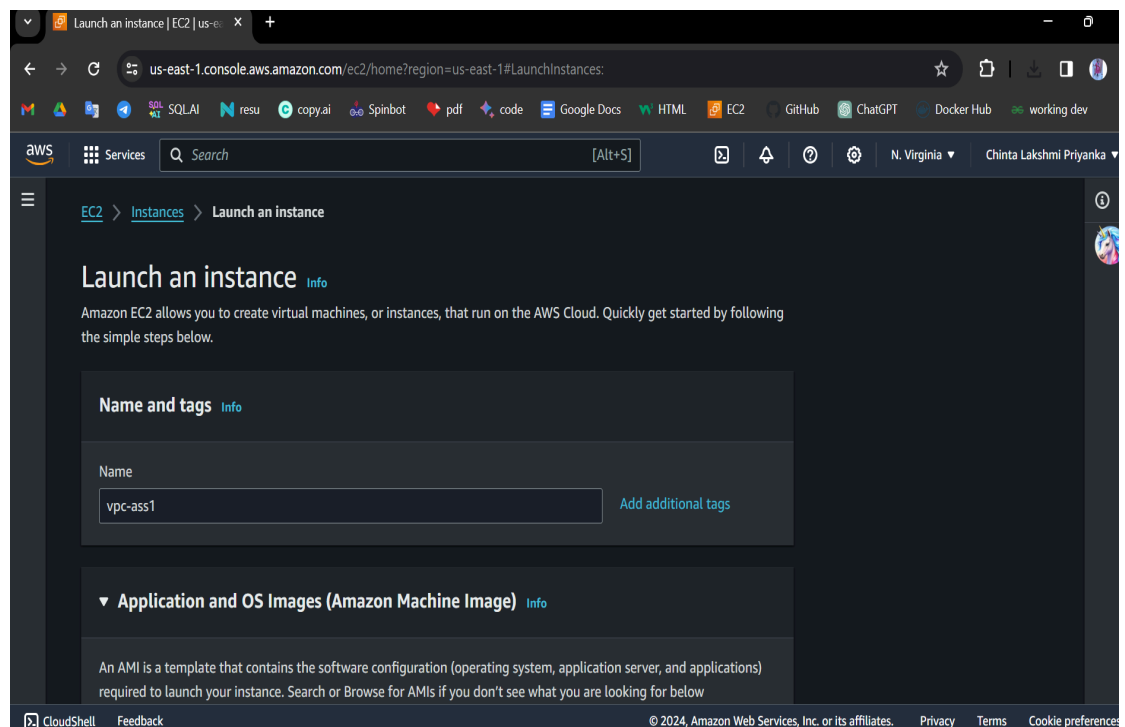
2. Click on Create VPC



The screenshot shows the 'VPC settings' page in the AWS Management Console. The browser address bar indicates the URL: `us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateVpc:createMode=vpcOnly`. The page has a dark theme. At the top, there is a description: 'A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.' Below this is the 'VPC settings' section. It includes a 'Resources to create' section with two radio button options: 'VPC only' (selected) and 'VPC and more'. There is a 'Name tag - optional' section with a description: 'Creates a tag with a key of 'Name' and a value that you specify.' and a text input field containing 'my vpc-assi'. The 'IPv4 CIDR block' section has two radio button options: 'IPv4 CIDR manual input' (selected) and 'IPAM-allocated IPv4 CIDR block'. Below this is a text input field for the 'IPv4 CIDR' block, containing '172.16.0.0/16'. A note below the input field states: 'CIDR block size must be between /16 and /28.' There is also an 'IPv6 CIDR block' section with an 'Info' link. At the bottom, there is a footer with '© 2021 Amazon Web Services, Inc. or its affiliates' and other links.



1. Resources to create: VPC only
2. Name tag: My Project VPC
5. IPv4 CIDR block: IPv4 CIDR manual Input
6. IPv4 CIDR: 172.16.0.0/16
7. VPC is Created then create the subnets
- a. Public and Private Subnets



5. IPv4 CIDR block: IPv4 CIDR manual Input
6. IPv4 CIDR: 172.16.0.0/16

7. VPC is Created then create the subnets

a. Public and Private Subnets.

8. Public Subnet

a. Subnet name: my-public-sub

b. Availability Zone: **Europe (London)/ eu-west-2a**

c. IPv4 VPC CIDR block: 172.16.0.0/16 (VPC IP address)

d. IPv4 subnet CIDR block: 172.16.1.0/24

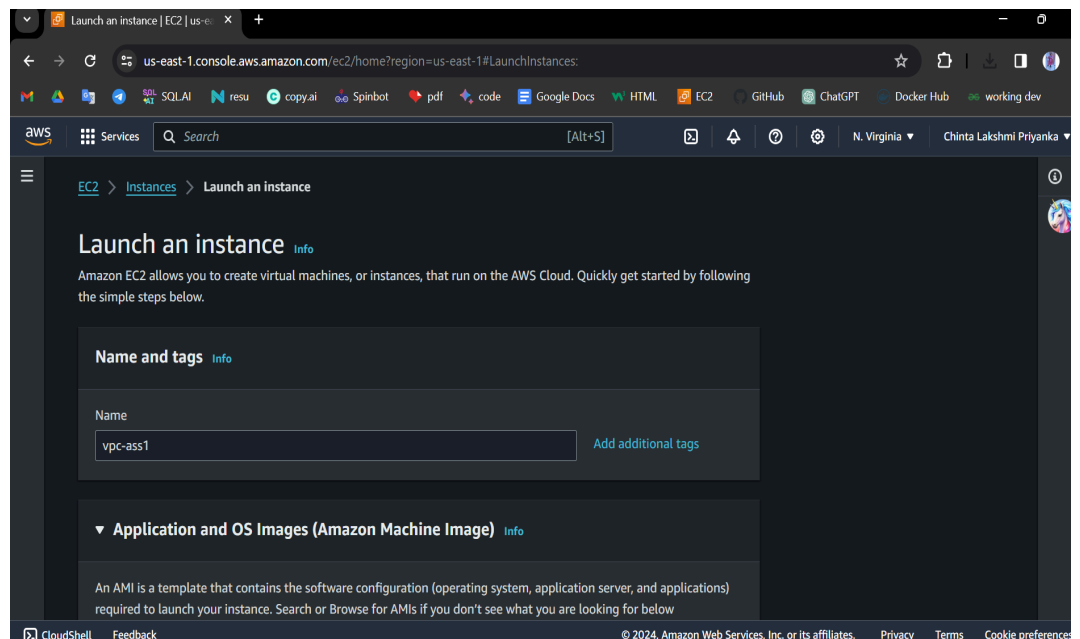
8. private Subnet

a. Subnet name: my-private-sub

b. Availability Zone: **Europe (London)/ eu-west-2c**

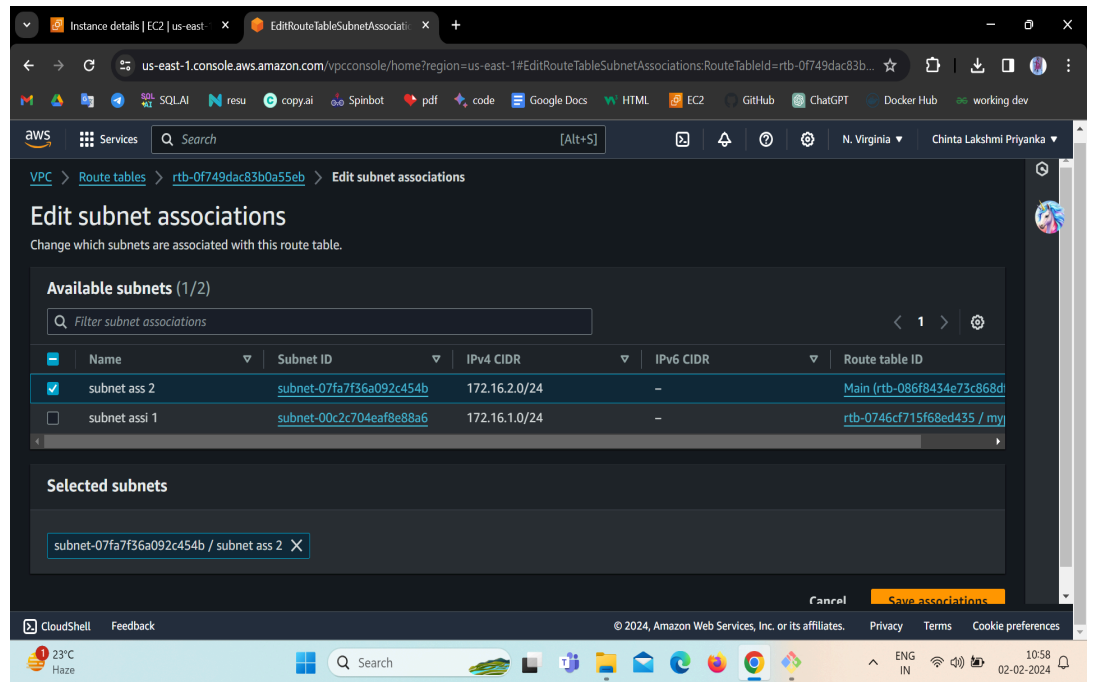
c. IPv4 VPC CIDR block: 172.16.0.0/16 (VPC IP address)

d. IPv4 subnet CIDR block: 172.16.2.0/24

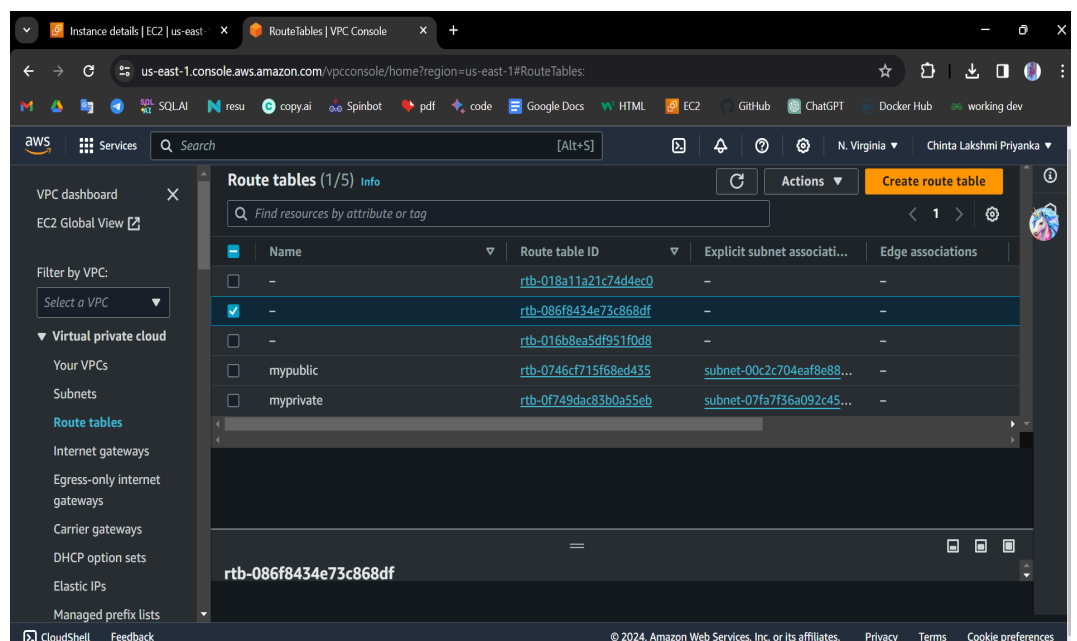


*** This is a public subnet**

*** This is a private subnet**



10. Create a Route table, to create the Route table click on the Route tables on the left of the VPC dashboard.



10. Create a Route table, to create the Route table click on the Route tables on the left of the VPC dashboard.

Click on the Create Route table

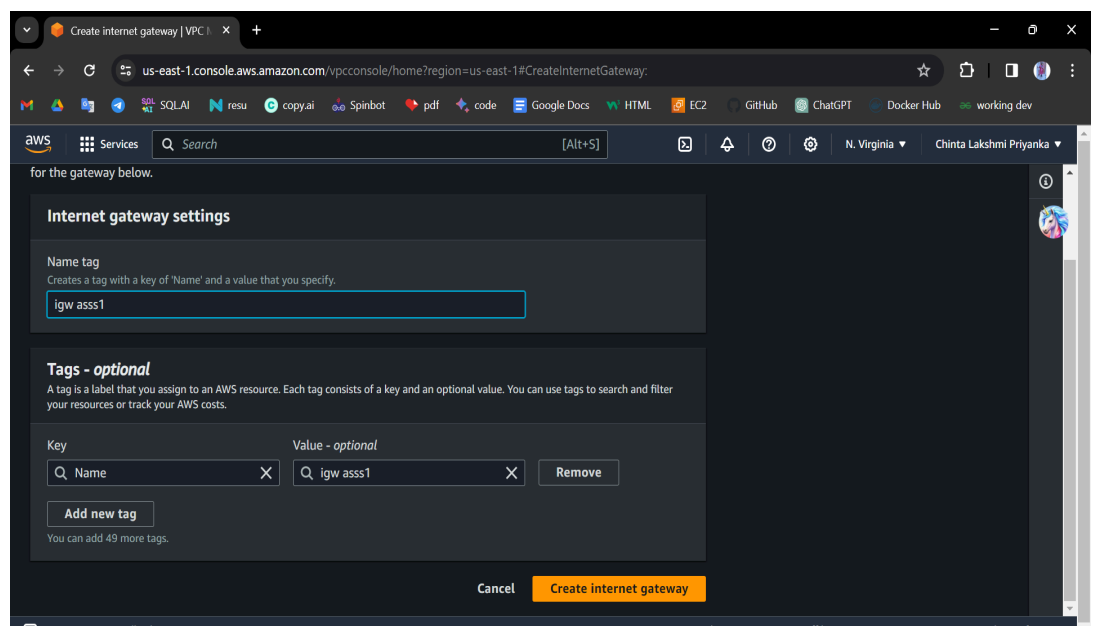
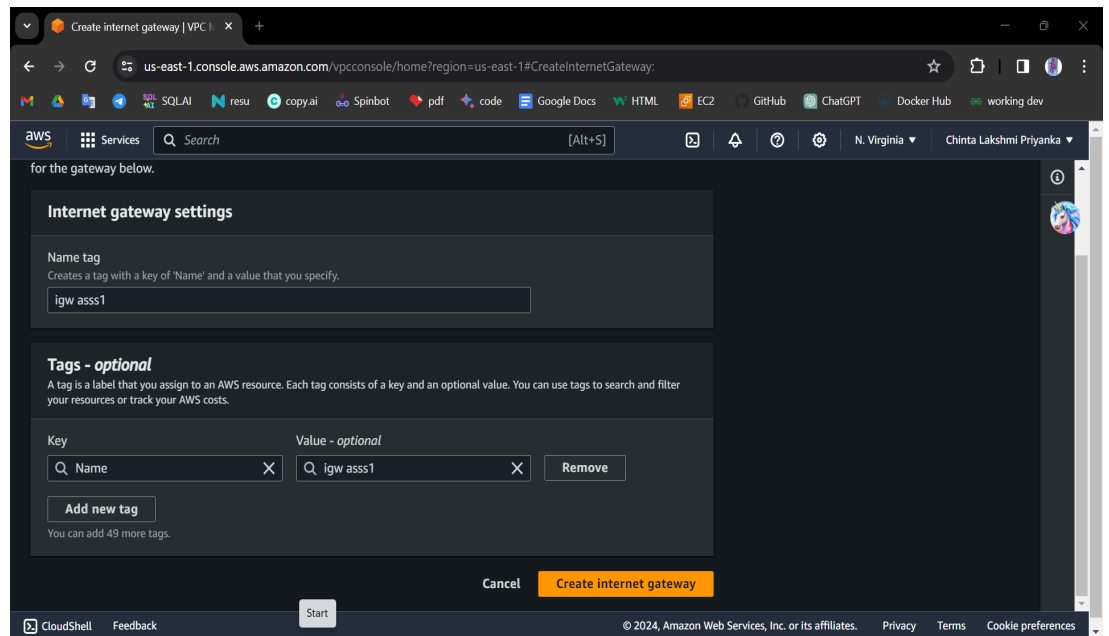
11. Create two Route tables

a. Public Route tables (Associate Internet gateway)

b. Private Route tables (Not Associate Internet gateway)

12. Route table name: my public (Associate with Public Subnet)

13. Route table name: my private (Associate with Private Subnet)



Click on Create Internet Gateway

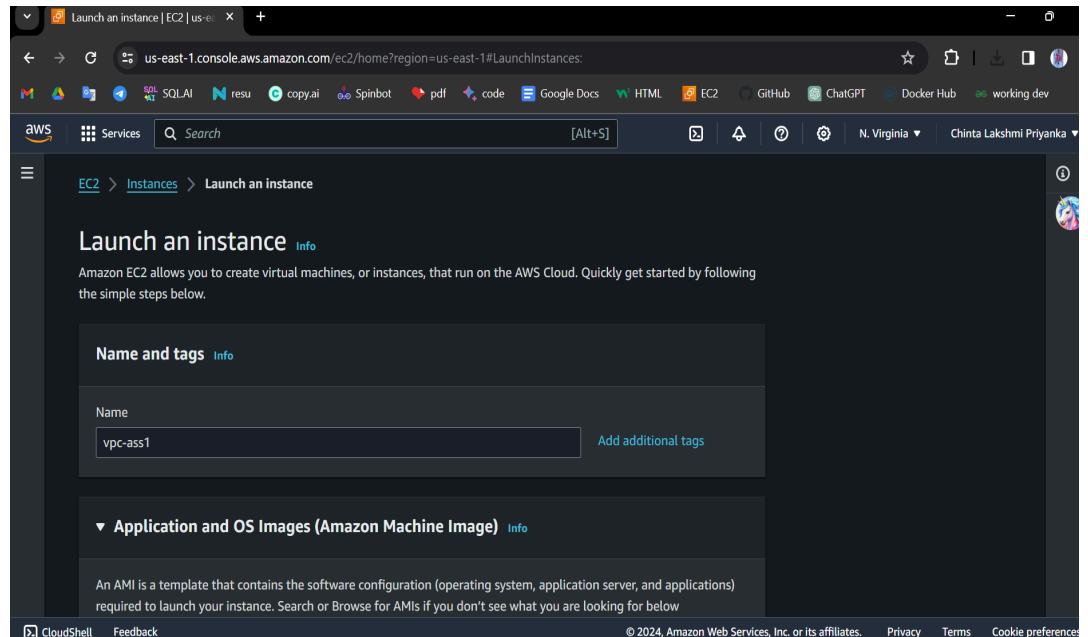
14. After Creating the Internet gateway Attach with VPC (Which we create)

Click on the Attach Internet gateway

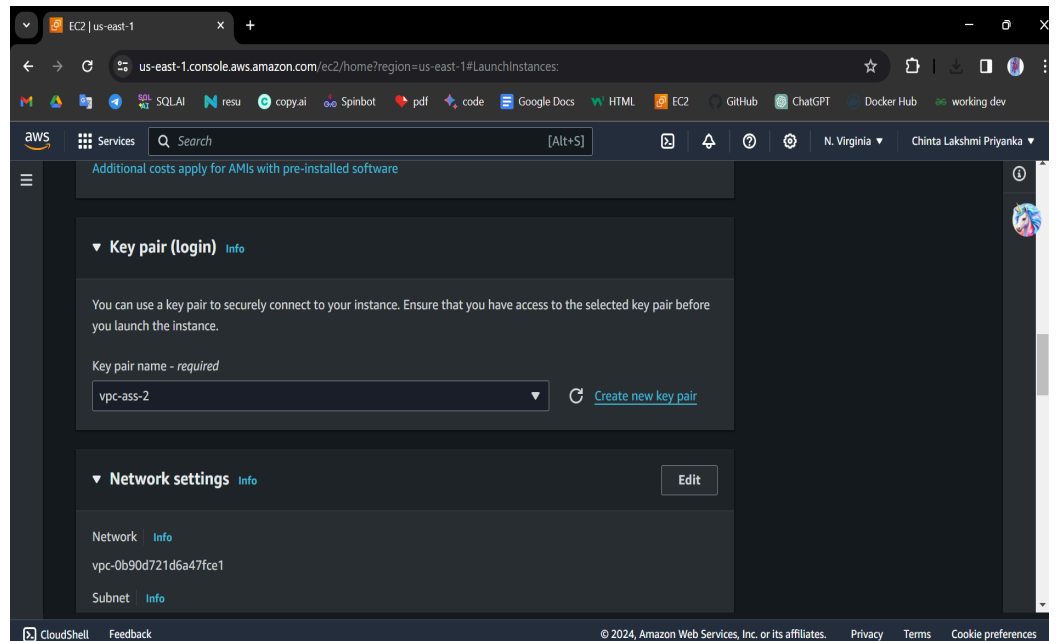
15. To connect the Internet gateway for the Route table to make the subnet public.

a. Go to edit routes

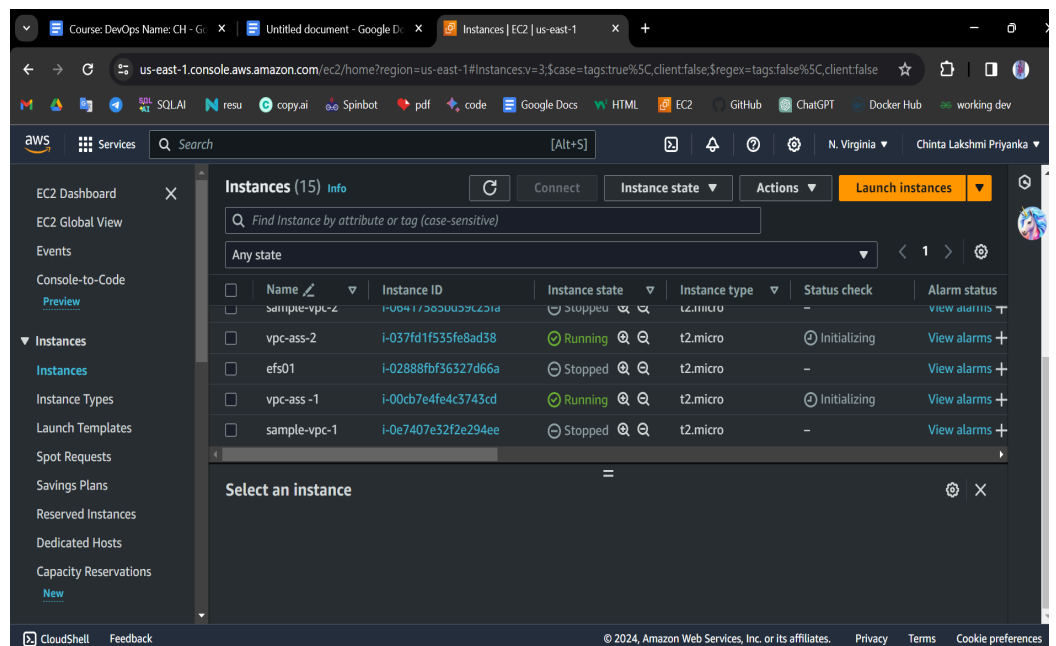
- b. Connect the Internet Gateway with (0.0.0.0./0)
- 15. To connect the Internet gateway for the Route table to make the subnet public.
 - a. Go to edit routes
 - b. Connect the Internet Gateway with (0.0.0.0./0)
- Click on Save Changes
- 16. Create an EC2 instance using our own VPC
- 17. click on the launch instance.



- . Click on edit VPC
 - a. VPC Select our VPC (vpc -ass1)
 - b. Subnet: Attach the my-public-sub
 - c. Auto-assign Public IP: Enable
 - d. Click on Launch Instance
- This instance is used for the public subnet.
- 19. Create another instance for the Private Subnet.
 - a. VPC Select our VPC (vpc-ass2)
 - b. Subnet: Attach the my-private-subnet
 - c. Auto-assign Public IP: Enable
 - d. Click on Launch Instance.
- 20. Note: Use different KeyPairs.



Launch Instance



Open Instances and connect the EC2 sever to git bash.

23. We can open the Public Instances in the git bash, we can't open Private instances in the git bash.

24. We should Connect the Private using the Public instances and use the Private IPv4 to connect the server to the git bash.

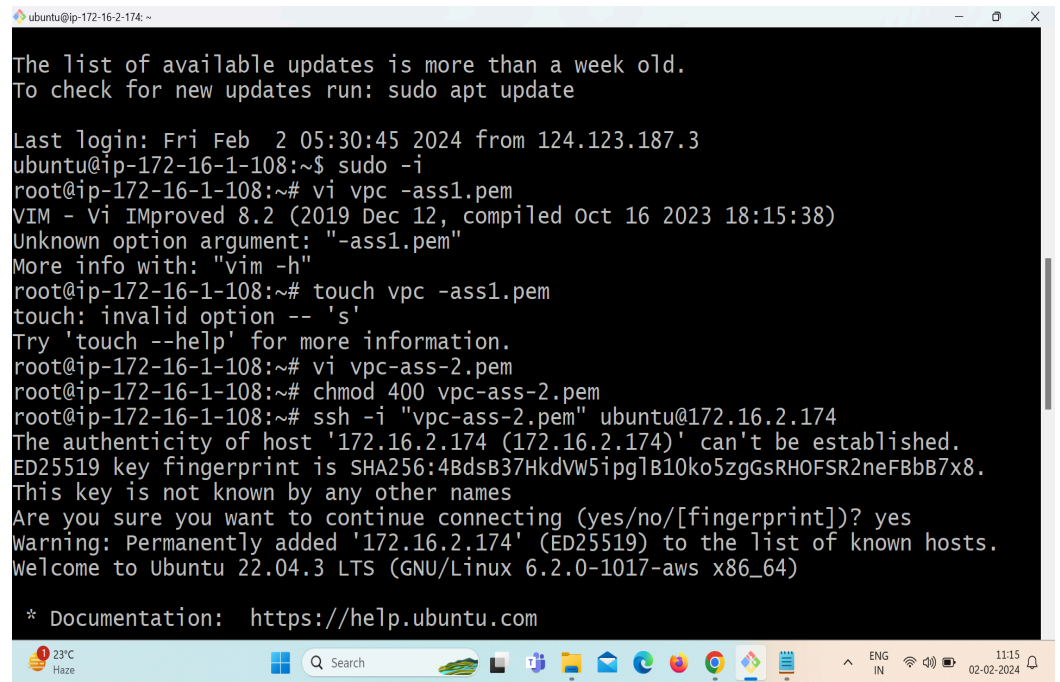

```
root@ip-172-16-1-108: ~  
ch1ak@ZENBOOK MINGW64 /c/Desktop  
$ chmod 400 "vpc-ass1.pem"  
ch1ak@ZENBOOK MINGW64 /c/Desktop  
$ ssh -i "vpc-ass1.pem" ubuntu@54.161.98.31  
The authenticity of host '54.161.98.31 (54.161.98.31)' can't be established.  
ED25519 key fingerprint is SHA256:EmlduHNUDDX5u4YjHwVCLciuxns1HDe+2NMI2HCaaY.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '54.161.98.31' (ED25519) to the list of known hosts.  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)  
  
 * Documentation:  https://help.ubuntu.com  
 * Management:    https://landscape.canonical.com  
 * Support:       https://ubuntu.com/advantage  
  
System information as of Fri Feb  2 05:30:44 UTC 2024  
  
System load:  0.369140625      Processes:            103  
Usage of /:   20.6% of 7.57Gb   Users logged in:      0  
Memory usage: 20%             IPv4 address for eth0: 172.16.1.108  
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.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-16-1-108:~$ sudo -i  
root@ip-172-16-1-108:~# |
```

See in the Picture, we get the Public we can't get the Private.

- a. So we should connect the first Public instances.
- b. Then we should copy the Private (Pem) to the Public instances.
- c. After copying the Pem file private, then use the command
- i. Chmod 400 (Private pem file name.pem)
- d. Connect the Private instances to the Public instances, using the Private IPv4 to connect.

```
MINGW64/c/Desktop  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-16-1-108:~$ sudo -i  
root@ip-172-16-1-108:~# exit  
logout  
ubuntu@ip-172-16-1-108:~$ client_loop: send disconnect: Connection reset by peer  
  
ch1ak@ZENBOOK MINGW64 /c/Desktop  
$ chmod 400 "vpc-ass-2.pem"  
  
ch1ak@ZENBOOK MINGW64 /c/Desktop  
$ ssh -i "vpc-ass-2.pem" ubuntu@44.192.128.120  
ssh: connect to host 44.192.128.120 port 22: Connection timed out  
  
ch1ak@ZENBOOK MINGW64 /c/Desktop  
$
```

```
ubuntu@ip-172-16-2-174: ~  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
Last login: Fri Feb  2 05:30:45 2024 from 124.123.187.3  
ubuntu@ip-172-16-1-108:~$ sudo -i  
root@ip-172-16-1-108:~# vi vpc -ass1.pem  
VIM - Vi IMproved 8.2 (2019 Dec 12, compiled Oct 16 2023 18:15:38)  
Unknown option argument: "-ass1.pem"  
More info with: "vim -h"  
root@ip-172-16-1-108:~# touch vpc -ass1.pem  
touch: invalid option -- 's'  
Try 'touch --help' for more information.  
root@ip-172-16-1-108:~# vi vpc-ass-2.pem  
root@ip-172-16-1-108:~# chmod 400 vpc-ass-2.pem  
root@ip-172-16-1-108:~# ssh -i "vpc-ass-2.pem" ubuntu@172.16.2.174  
The authenticity of host '172.16.2.174 (172.16.2.174)' can't be established.  
ED25519 key fingerprint is SHA256:4BdsB37HkdVW5ipglB10ko5zgGsrHOFsr2neFBbB7x8.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '172.16.2.174' (ED25519) to the list of known hosts.  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com
```

The image shows a terminal window on an Ubuntu desktop. The terminal displays a series of commands and their outputs, including system updates, login history, and an SSH session. The desktop environment includes a search bar, application icons, and a system status bar at the bottom showing temperature, language, and time.

THE-END