

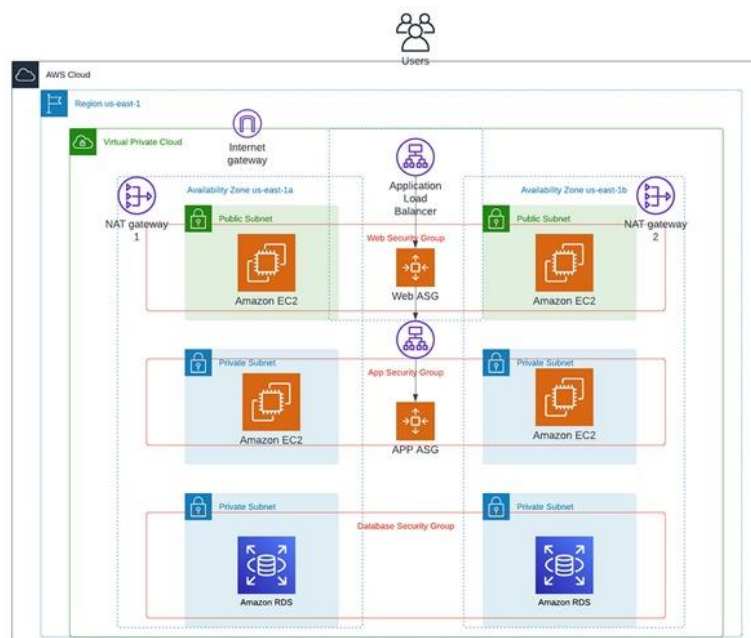


## Project-1

Course: DEVOPS

MODULE: AWS 3- Tier Architecture.

TRAINER: Mr. MADHUKAR



Name: Geethasri Thirunahari

Mobile: 9346847694

Email: [geethasrithirunahari@gmail.com](mailto:geethasrithirunahari@gmail.com)

Log in to the AWS management console.

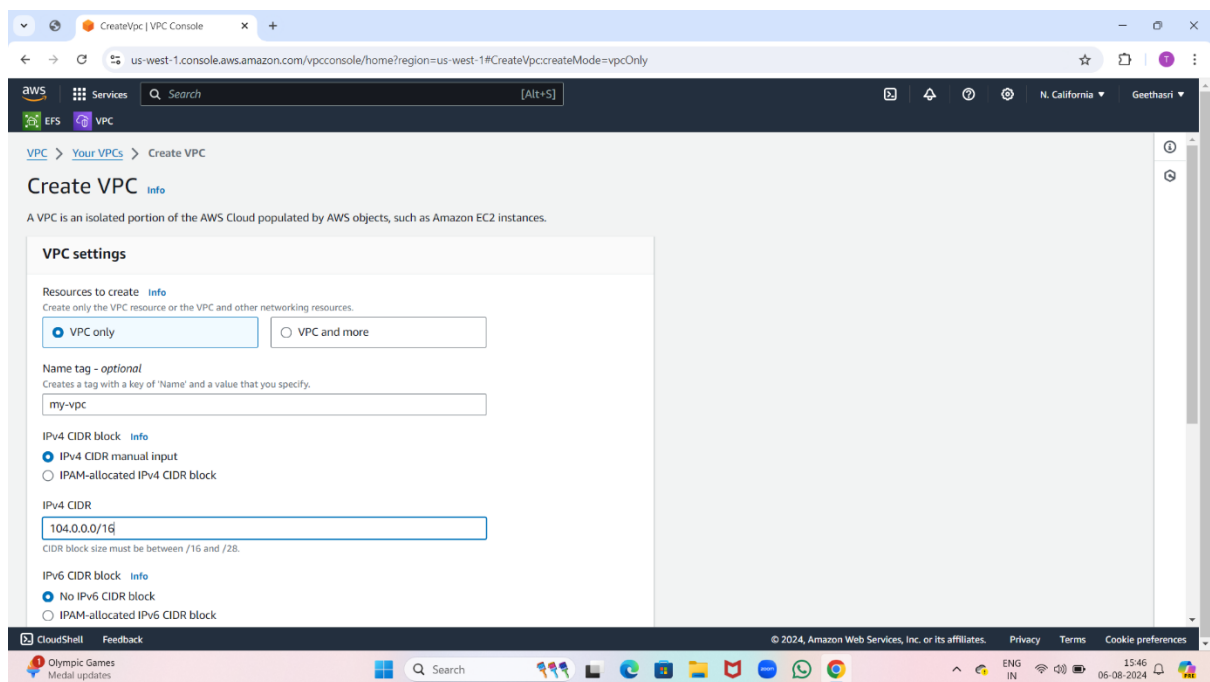
Navigate to the VPC dashboard.

Click on “create VPC”.

Give name to your VPC.

Enter IPV4 CIDR (104.0.0.0/16) as your own Ip address.

Click on “create VPC”.



Now your own VPC created.

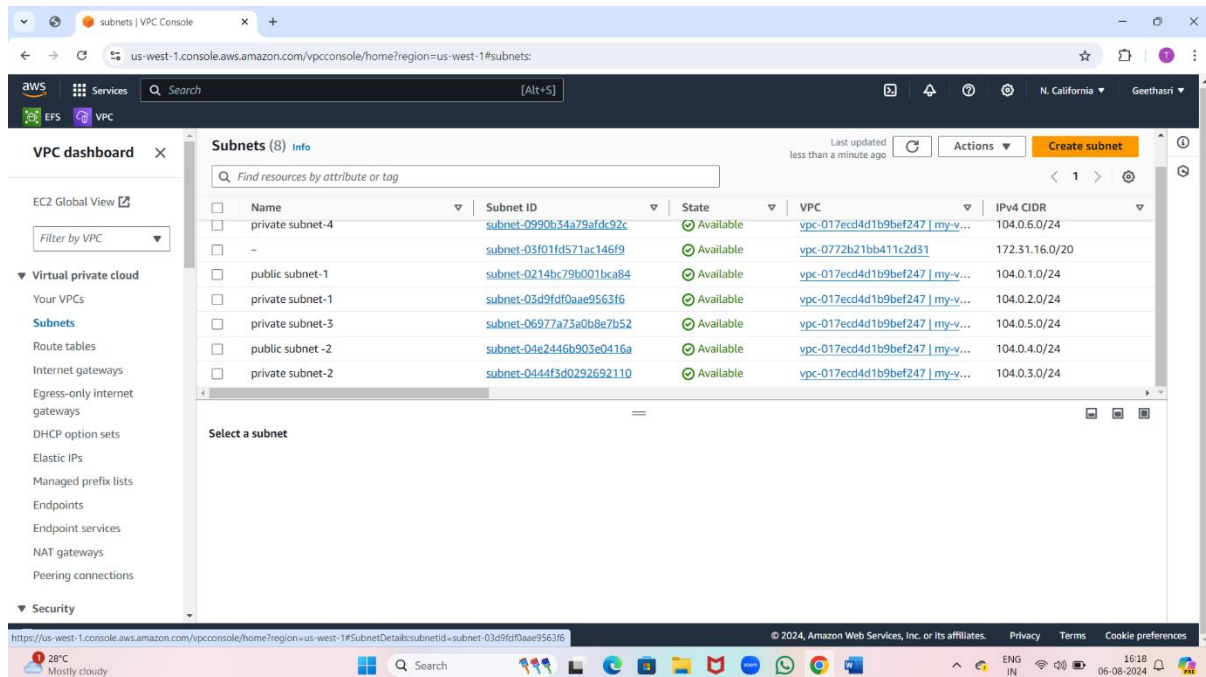
Now go to subnets and click on “create subnets”.

Select your own VPC and give subnet name public one.

Enter IPV4 CIDR block and select availability zone.

Then click on “create subnet”.

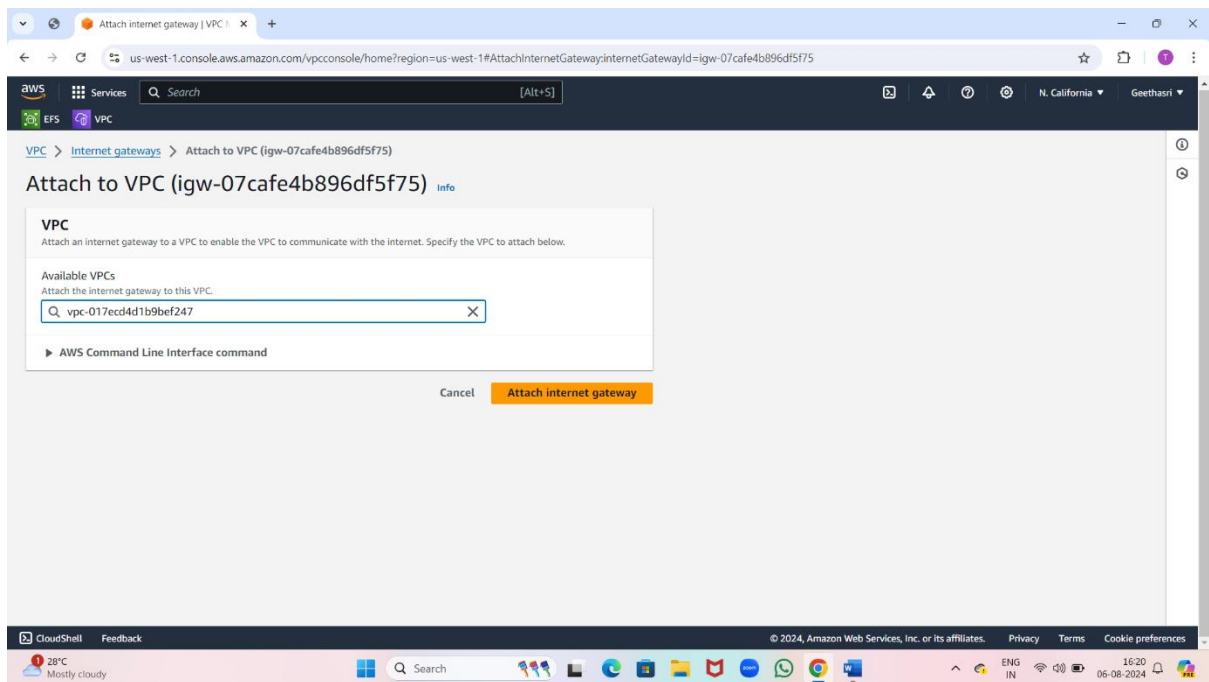
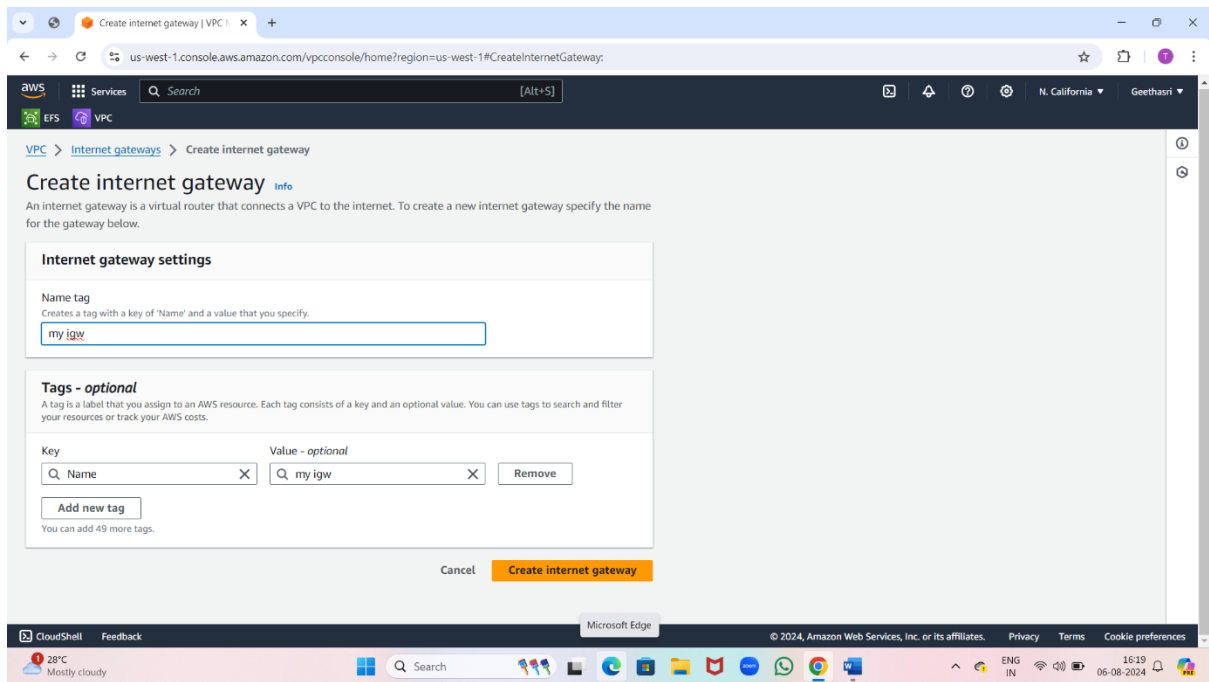
Now create two another two public subnets with different availability zone and create three private subnets with different availability zone.



Now create an INTERNET GATEWAY (IGW).

Click on “create internet gateway”.

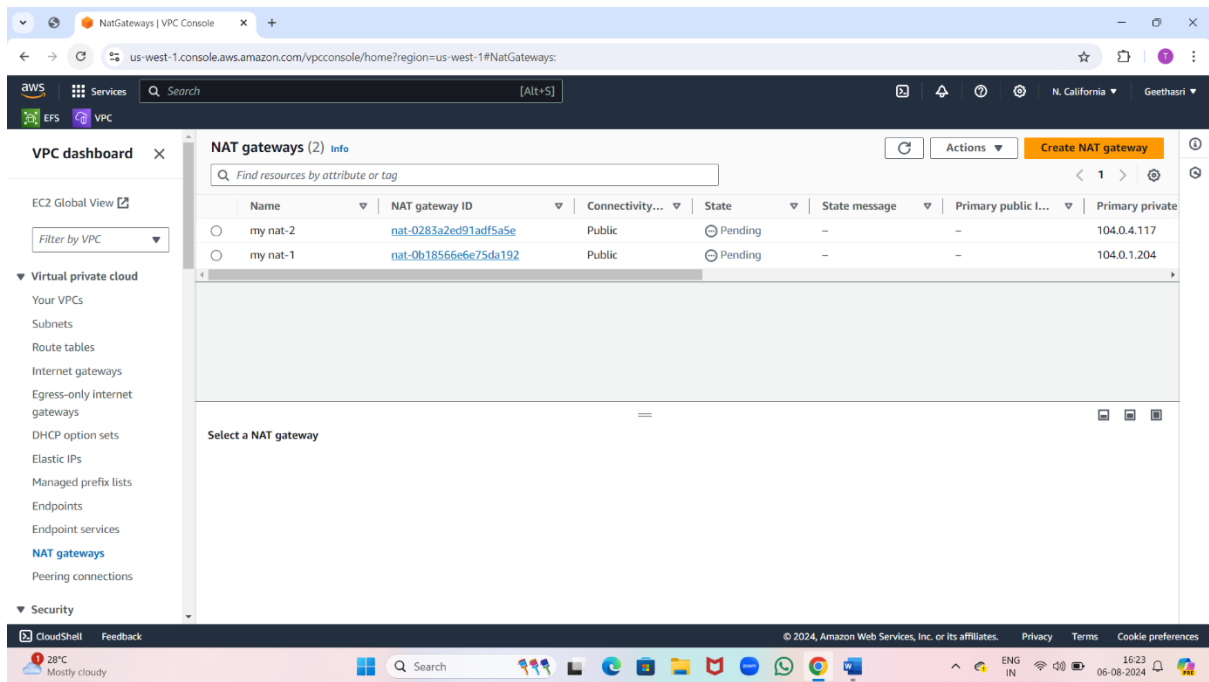
Attach the internet gateway to you VPC.



Now create a NAT gateway .

Click on “create NAT gateway”.

Attach VPC to NAT gateway.



Now create a route table.

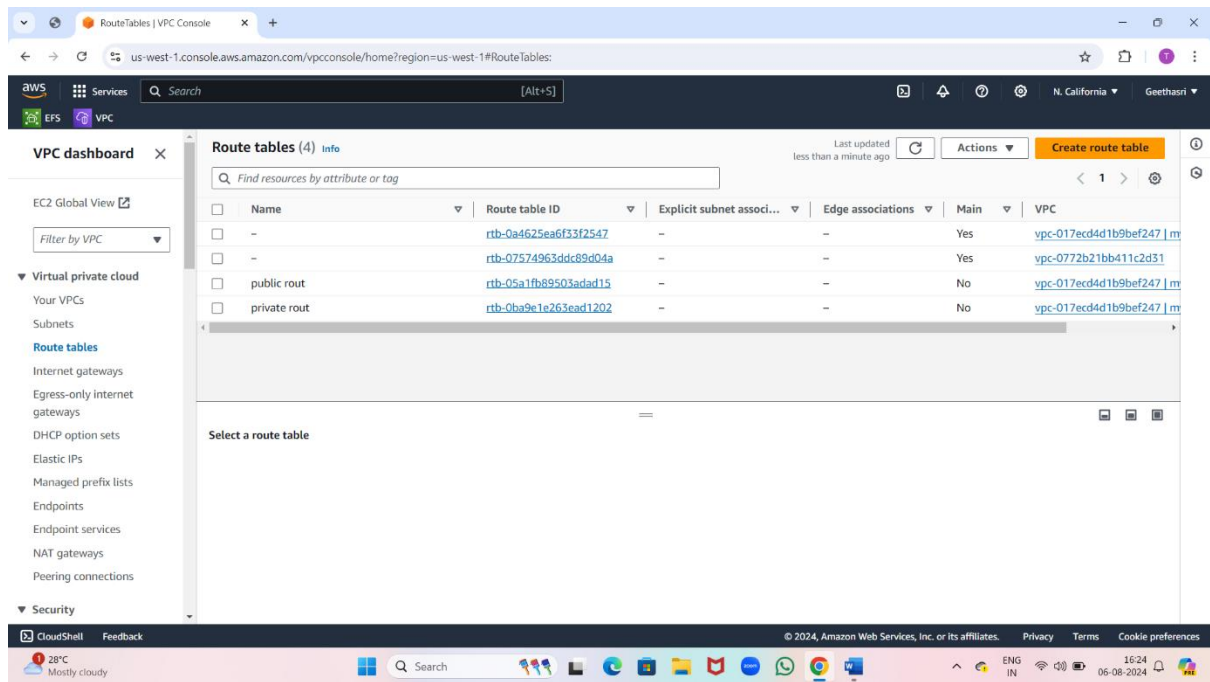
Click on “create route table”.

Associate subnets with route tables.

Click on “edit subnet”.

Select the route table.

Click on “ save changes”.

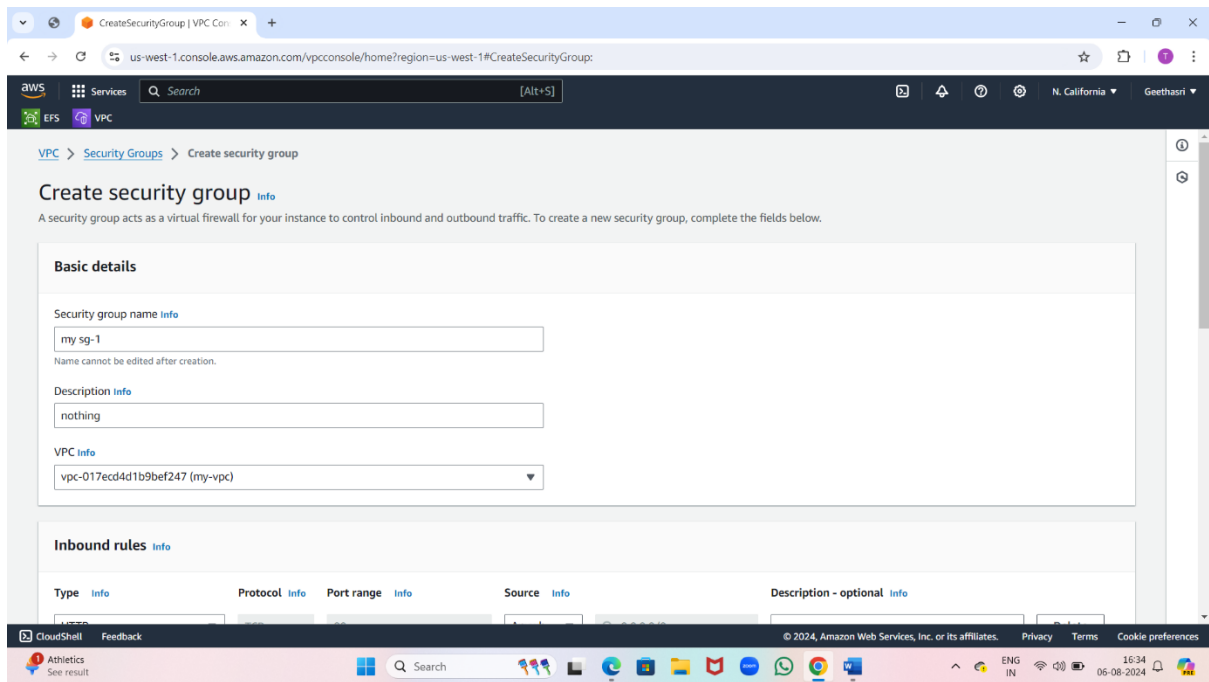


Now go to security group.

Click on “create security group”.

Enter the security group name and select the VPC where you want to create the security group.

Now click on “create security group”.



Now launch two templates.

Go to the EC2 dashboard.

Click on “instances” In the left- hand menu.

Select “ launch template” as the instance launch method.

Choose the launch template you want to use.

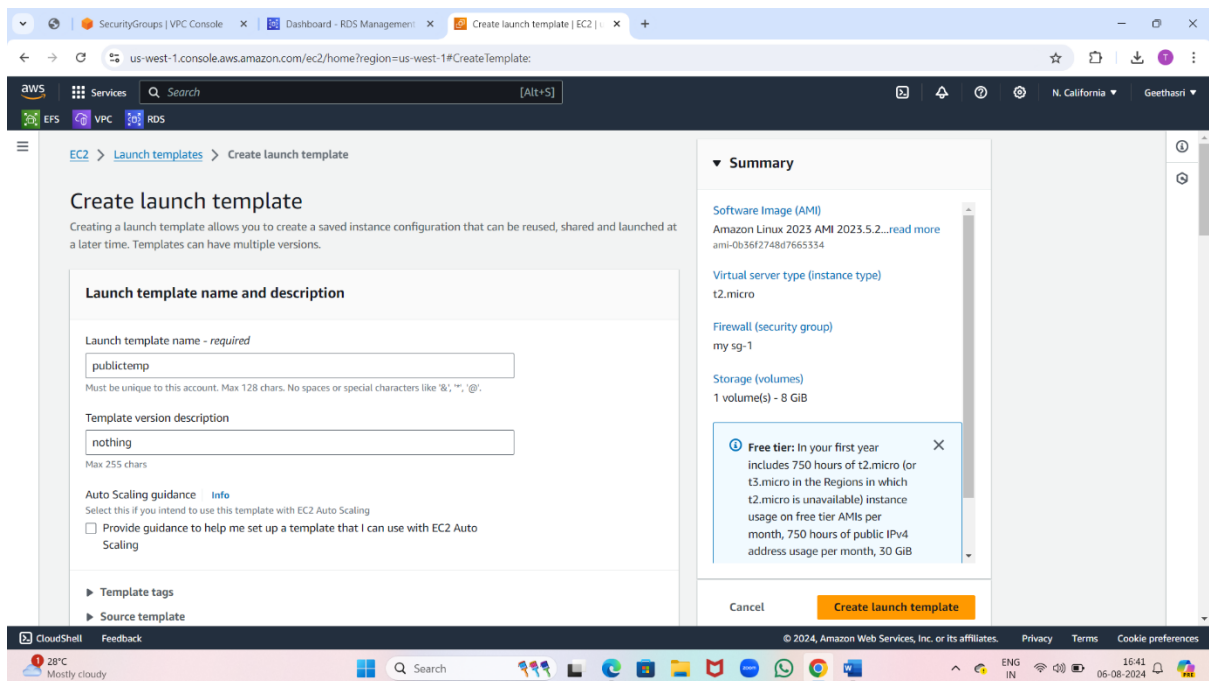
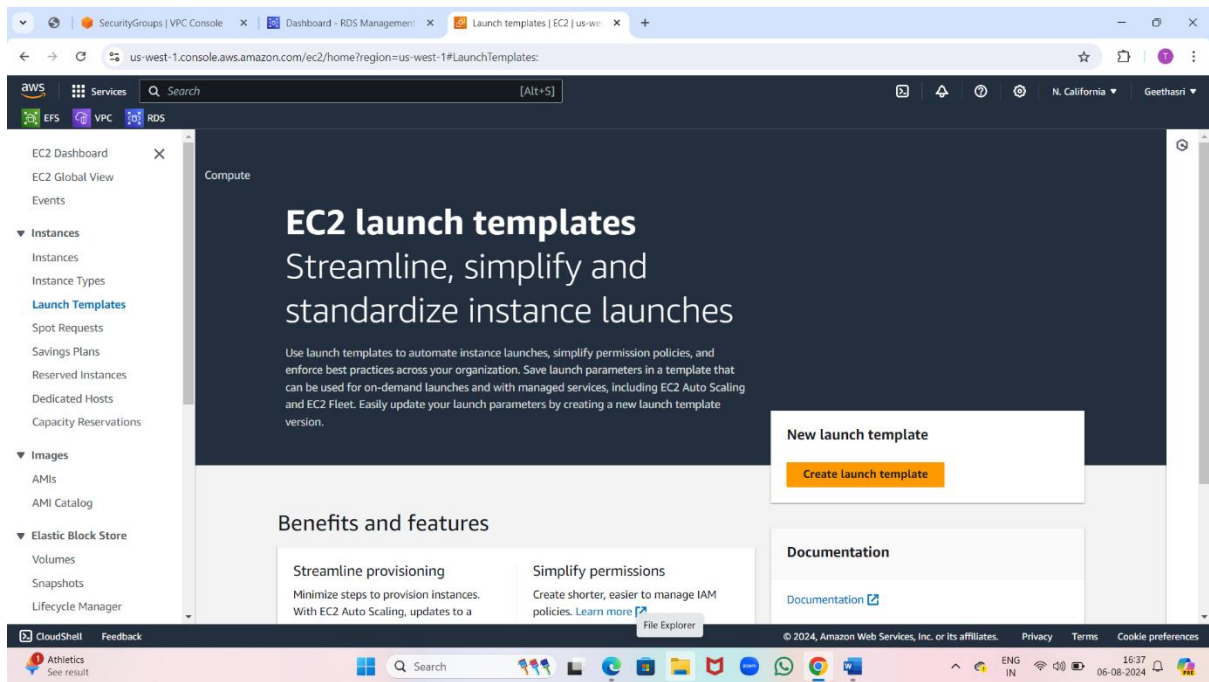
Select instance type.

Create key pair.

Choose security groups.

Select the subnet.

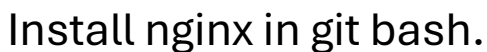
Click on “launch instances”.



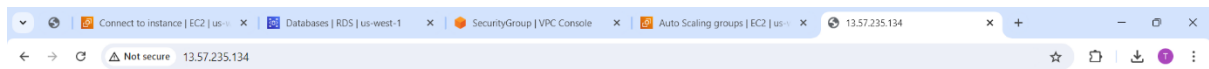
Now automatically EC2 instances will create.

Give names to created instances.





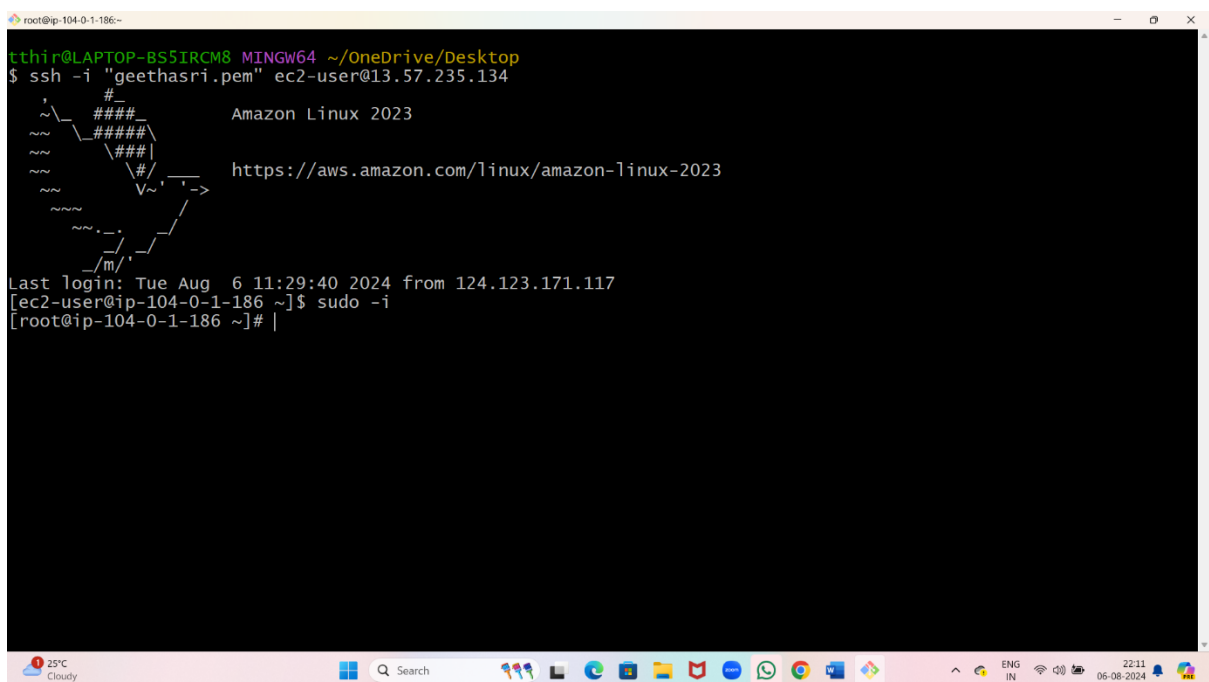




web-1



Now connect public 2 instance to web and install nginx.





```
root@ip-104-0-1-186:/usr/share/nginx/html
Nothing to do.
Complete!
[root@ip-104-0-1-186 ~]# yum install nginx -y
Last metadata expiration check: 5:23:44 ago on Tue Aug 6 11:22:46 2024.
Package nginx-1:1.24.0-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-104-0-1-186 ~]# cd /usr/share/nginx/html
[root@ip-104-0-1-186 html]# rm index.html
rm: remove regular file 'index.html'? yes
[root@ip-104-0-1-186 html]# vi index.html
[root@ip-104-0-1-186 html]# systemctl restart nginx
Unknown command verb restart.
[root@ip-104-0-1-186 html]# systemctl restart nginx
[root@ip-104-0-1-186 html]# vi geethasri.pem
[root@ip-104-0-1-186 html]# chmod 777 geethasri.pem
[root@ip-104-0-1-186 html]# ssh -i "geethasri.pem" ec2-user@104.0.1.186
The authenticity of host '104.0.1.186 (104.0.1.186)' can't be established.
ED25519 key fingerprint is SHA256:1EkwnfkrNKR7wyZ9Yt7b7JvVEcSwlyPk4wKNWmWEhEY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '104.0.1.186' (ED25519) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                WARNING: UNPROTECTED PRIVATE KEY FILE!                @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0777 for 'geethasri.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "geethasri.pem": bad permissions
ec2-user@104.0.1.186: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[root@ip-104-0-1-186 html]#
```

Now attach public 2 instance to private 2 instance.

```
root@ip-104-0-4-231:/usr/share/nginx/html
[root@ip-104-0-4-231 ~]# yum install nginx -y
Last metadata expiration check: 5:33:56 ago on Tue Aug 6 11:22:42 2024.
Package nginx-1:1.24.0-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-104-0-4-231 ~]# cd /usr/share/nginx/html
[root@ip-104-0-4-231 html]# rm index.html
rm: remove regular file 'index.html'? yes
[root@ip-104-0-4-231 html]# vi index.html
[root@ip-104-0-4-231 html]# systemctl restart nginx
[root@ip-104-0-4-231 html]# vi geethasri.pem
[root@ip-104-0-4-231 html]# chmod 777 geethasri.pem
[root@ip-104-0-4-231 html]# ssh -i "geethasri.pem" ec2-user@104.0.4.231
The authenticity of host '104.0.4.231 (104.0.4.231)' can't be established.
ED25519 key fingerprint is SHA256:fKrkou2vvHpnTLQKAJl314SaMD1RVHcDe4LlYiddRuE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '104.0.4.231' (ED25519) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                WARNING: UNPROTECTED PRIVATE KEY FILE!                @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0777 for 'geethasri.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "geethasri.pem": bad permissions
ec2-user@104.0.4.231: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[root@ip-104-0-4-231 html]#
```

Now create database.

Go to the AWS management console and type RDS.

Click on RDS.

Click on “create database”.

Enter database name and master username and master password.

Click on “create database”.

The screenshot shows the AWS Management Console for the 'us-west-1' region, specifically the 'Launch db instance' wizard. The 'Additional configuration' section is expanded, displaying the following information:

**Additional configuration**  
Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned on.

**Estimated Monthly costs**

Category	Cost (USD)
DB instance	253.31
Storage	110.40
Provisioned IOPS	660.00
<b>Total</b>	<b>1023.71</b>

This billing estimate is based on on-demand usage as described in [Amazon RDS Pricing](#). Estimate does not include costs for backup storage, IOs (if applicable), or data transfer.

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#).

**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.

At the bottom of the wizard, there are 'Cancel' and 'Create database' buttons. A disclaimer states: 'You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.'