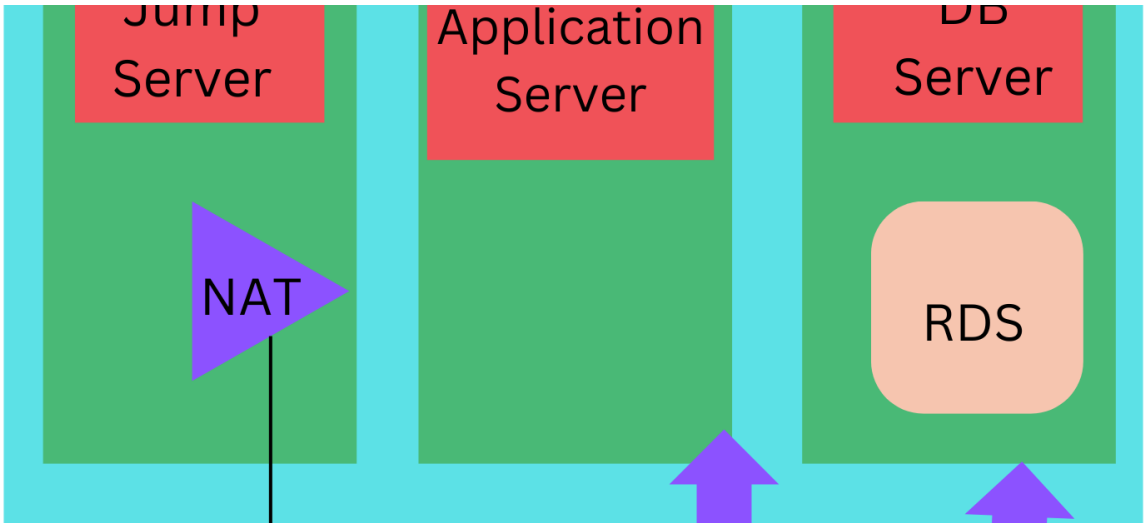


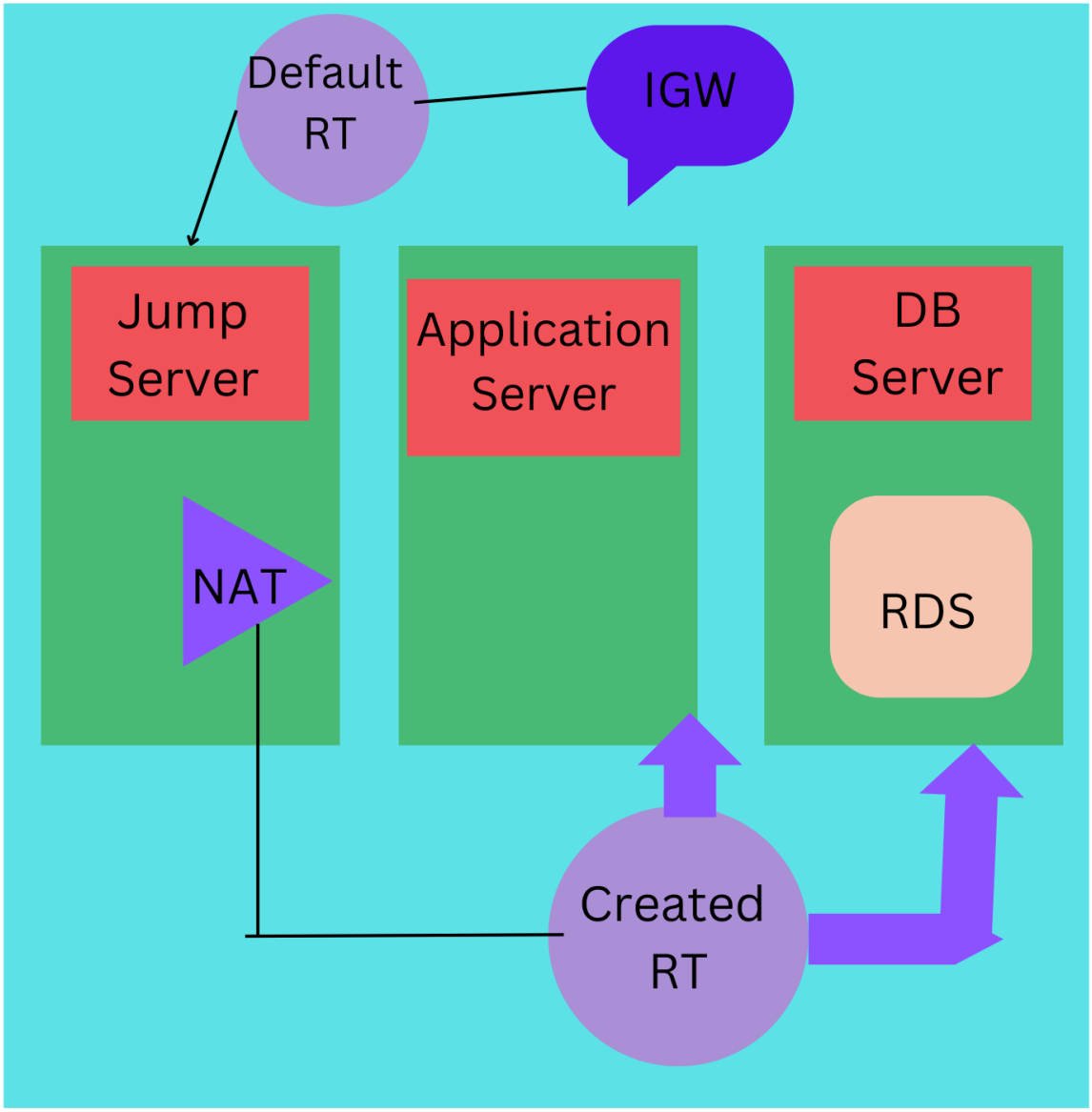
3 Tier Application

What is Three-Tier Architecture?

Three-Tier Architecture is an is an well established software application design pattern which will organizes the application in the three logical and physical computing tiers as following:

- **Presentation Tier**
- **Application Tier**
- **Data Tier**

The major benefit of the three tiers in client-server architecture is that these tiers are developed and maintained independently and this would not impact the other tiers in case of any modification. It allows for better performance and even more scalability in architecture can be made as with the increasing demand, more servers can be added.



The Three Tiers In Detail

Presentation Tier/Jump Server

It is the user interface and topmost tier in the architecture. Its purpose is to take request from the client and displays information to the client. It communicates with other tier by doing ssh. because we have created other two tier in private subnet.

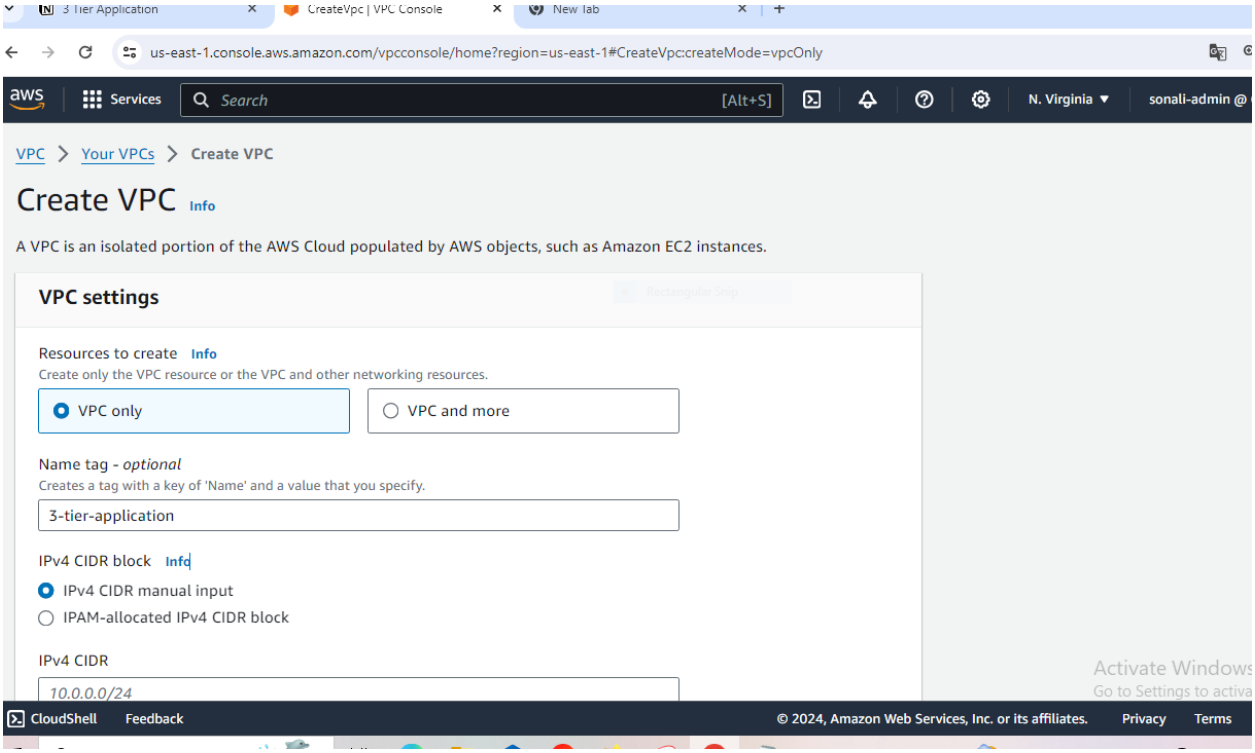
Step1: Login into your AWS Console

Step2: Create Infrastructure

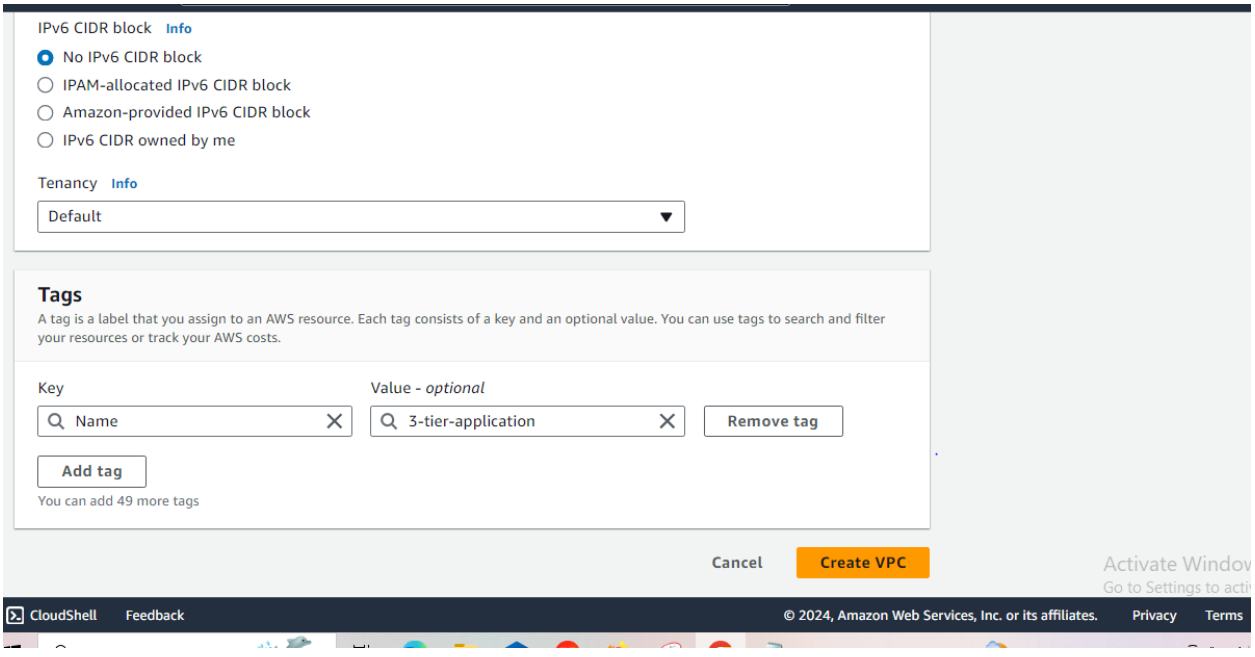
- Create VPC
- Create 3 Subnet in that one is public and other two is Private in different availability Zone.
- Make One Public by Enabling in Setting
- Create one IGW for Public Subnet
- Attach that subnet to VPC
- Create one NAT Gateway to make Subnet Private
- Created RDS in private Subnet
- We have connected Default RT to Public Subnet and added Route of Internet Gateway
- We have Created another RT for other two subnet and associated with them and added Route to NAT Gateway.
- Created 3 Server in 3 different subnet

Step 3:

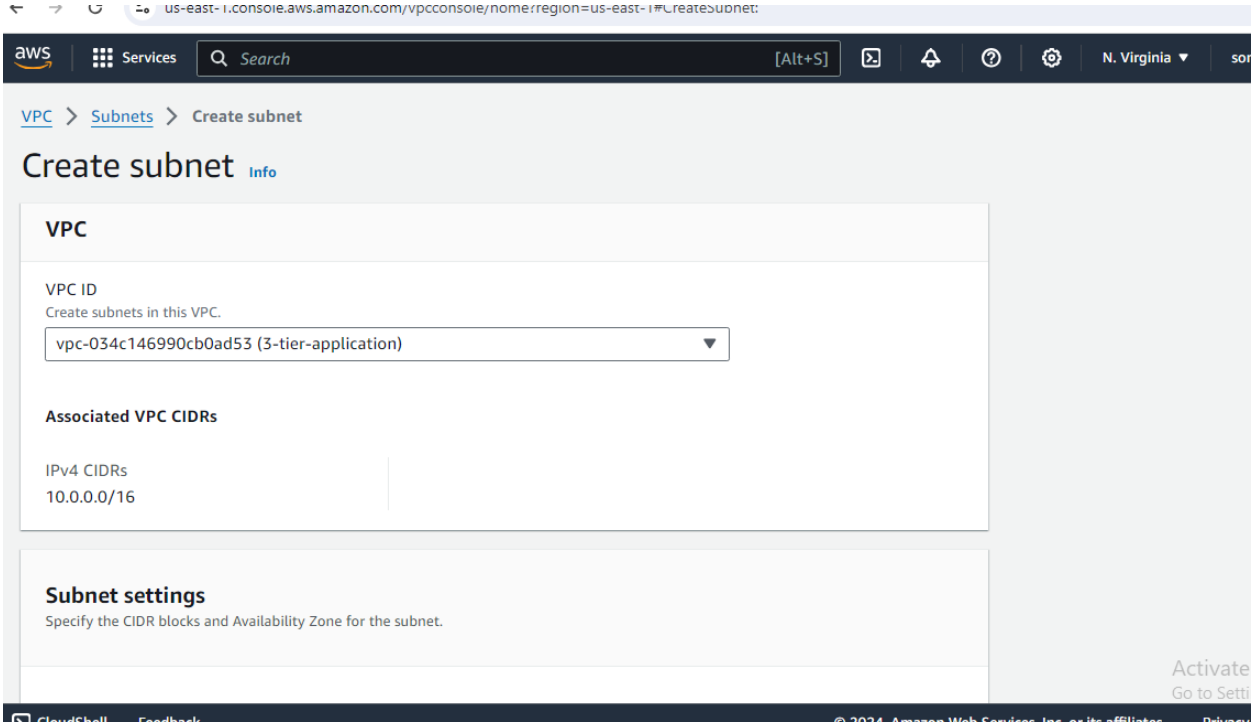
- In the AWS Management Console search bar, enter VPC, and click the VPC result under Services:
- To start creating VPC, in the left down side, Click on **Your VPC** to Create VPC:



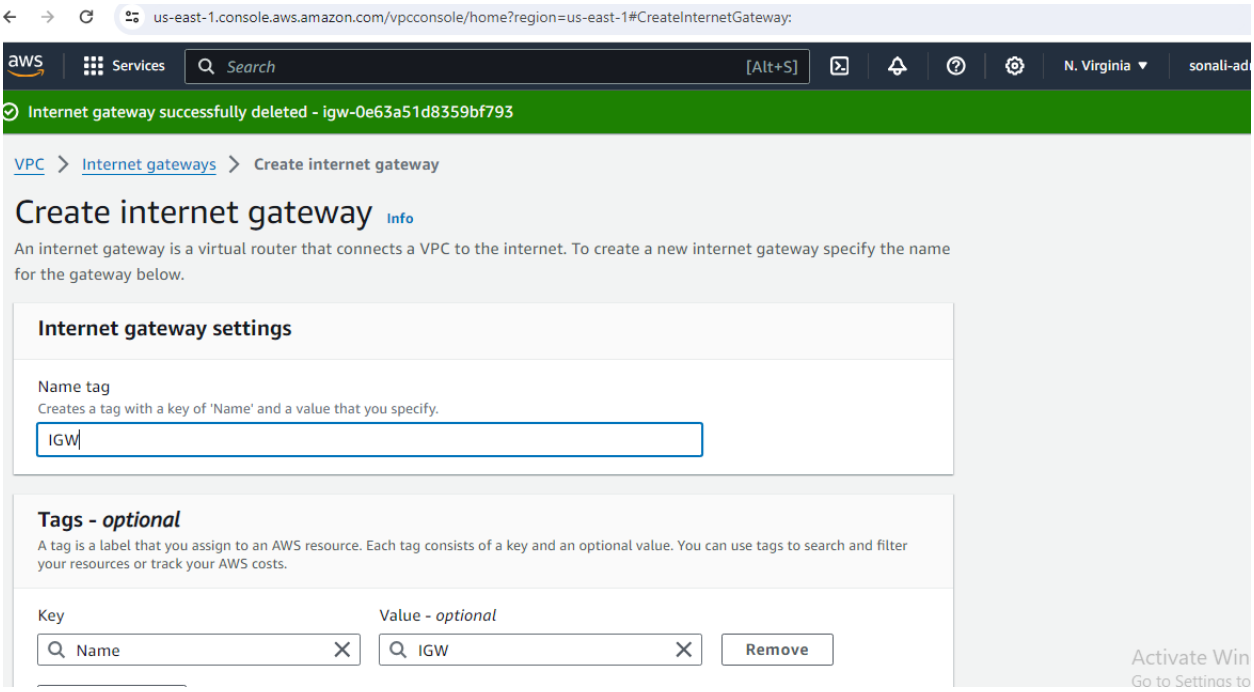
- Enter VPC name and IPV4 CIDR Range Click on create VPC
- Click on below button create VPC



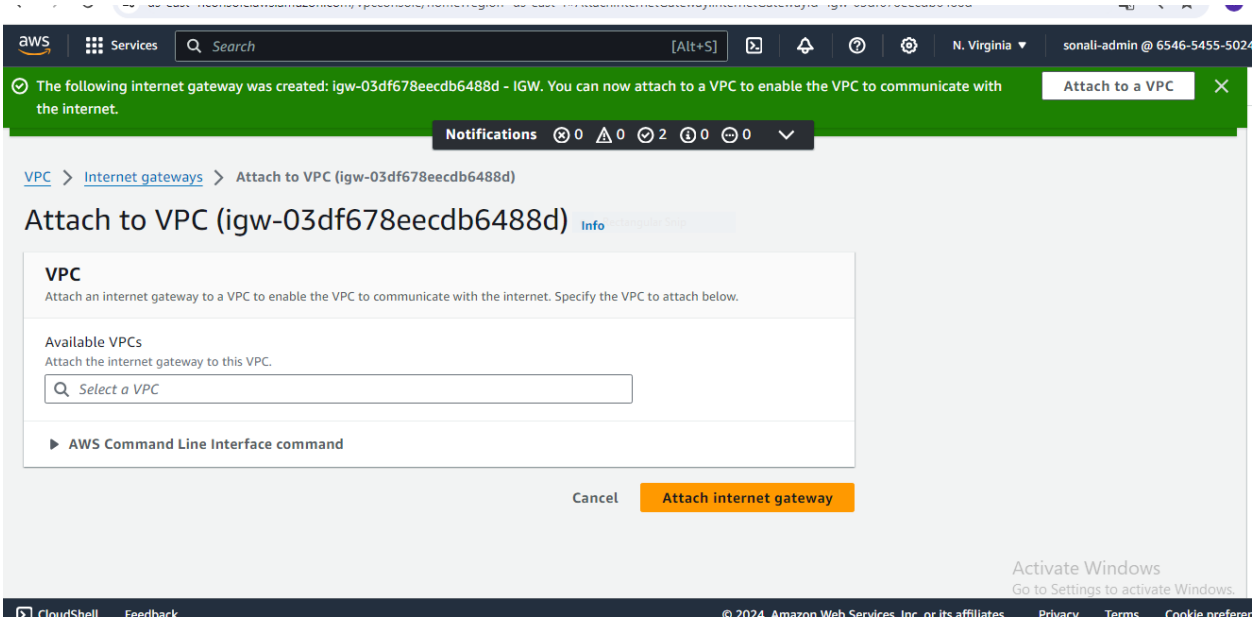
- To start creating subnet , in the left down side, click on Create **subnets**:
- Create 3 Subnet in that one is public and other two is Private in different availability Zone.



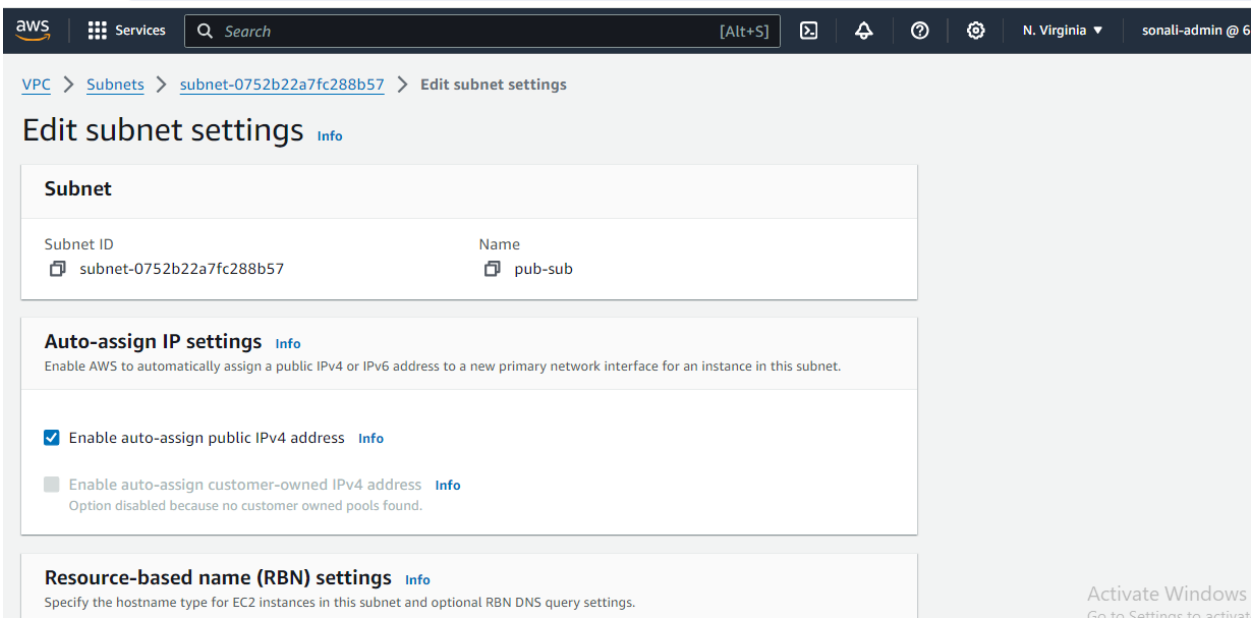
- To start creating IGW , in the left down side, click Create Internet Gateway:
- Give name to internet Gateway and click on create internet gateway



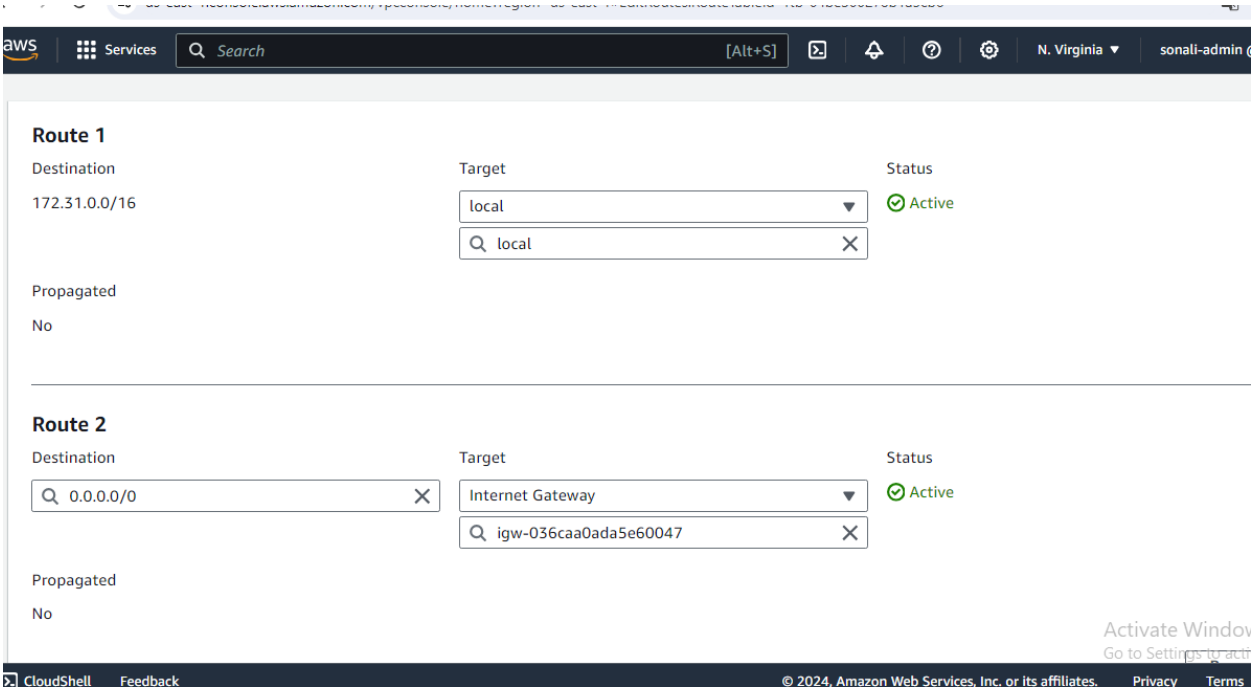
- Select IGW Which you have created recently and Click on **Action** and select option **Attach to VPC**



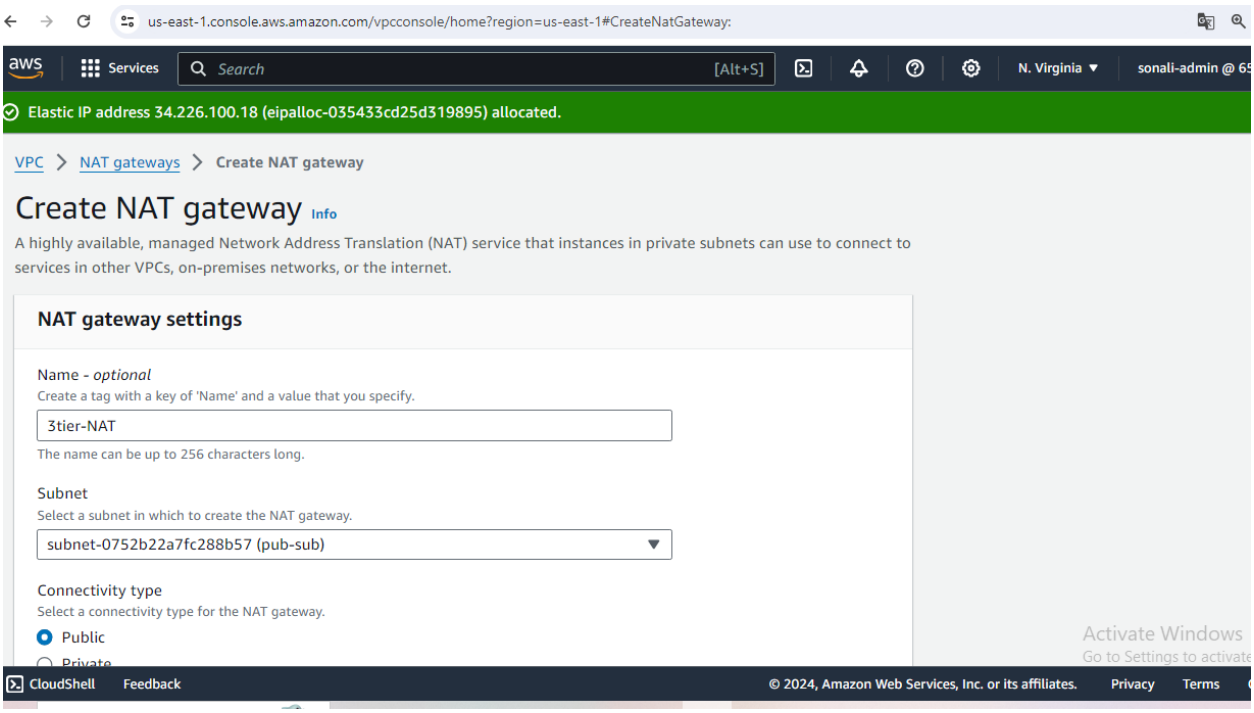
- Enable Public Setting of Auto -Assign-IP to make our subnet is public



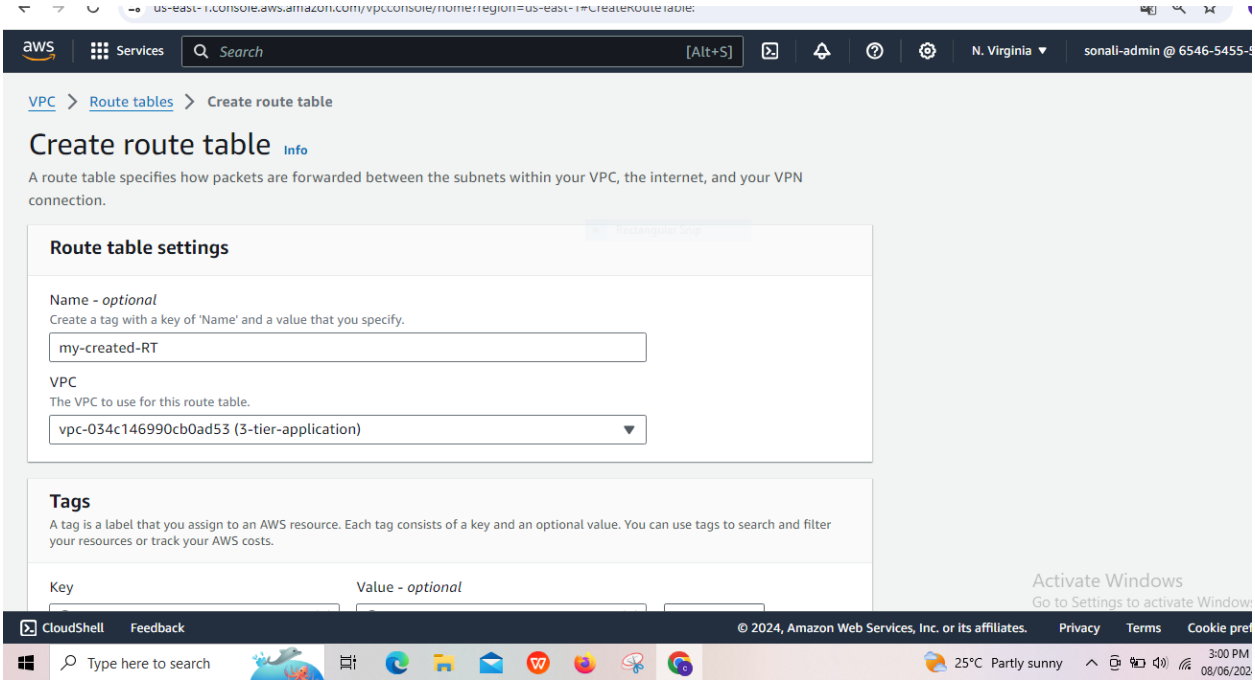
- Add Route in Default RT to Internet Gateway



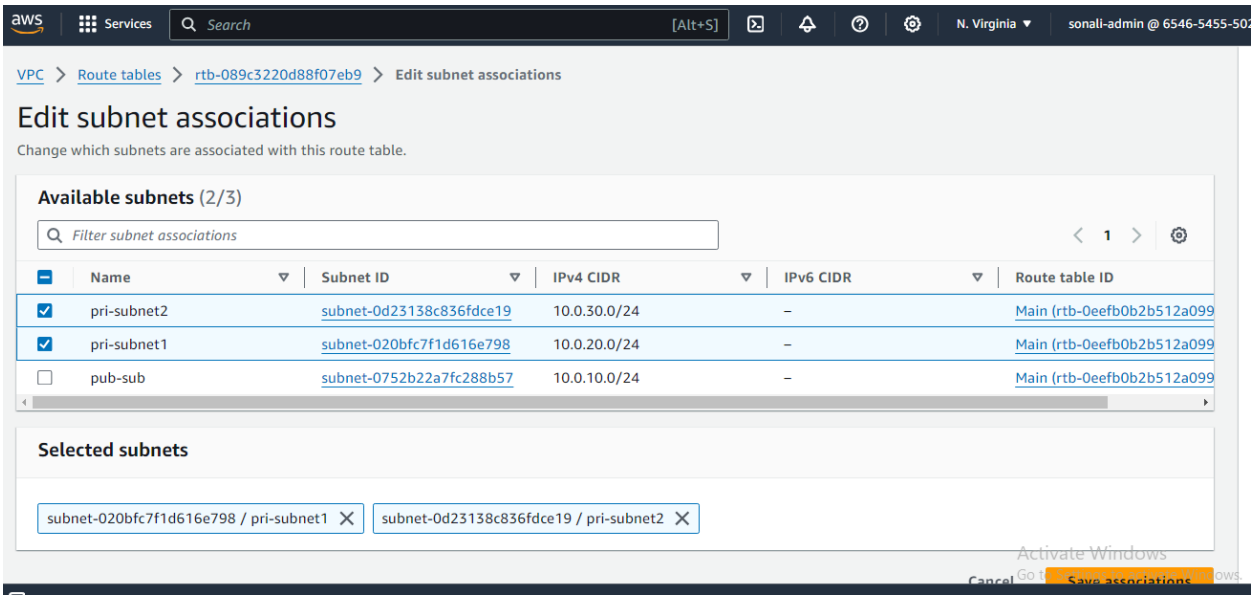
- Create NAT Gateway for Private Subnet



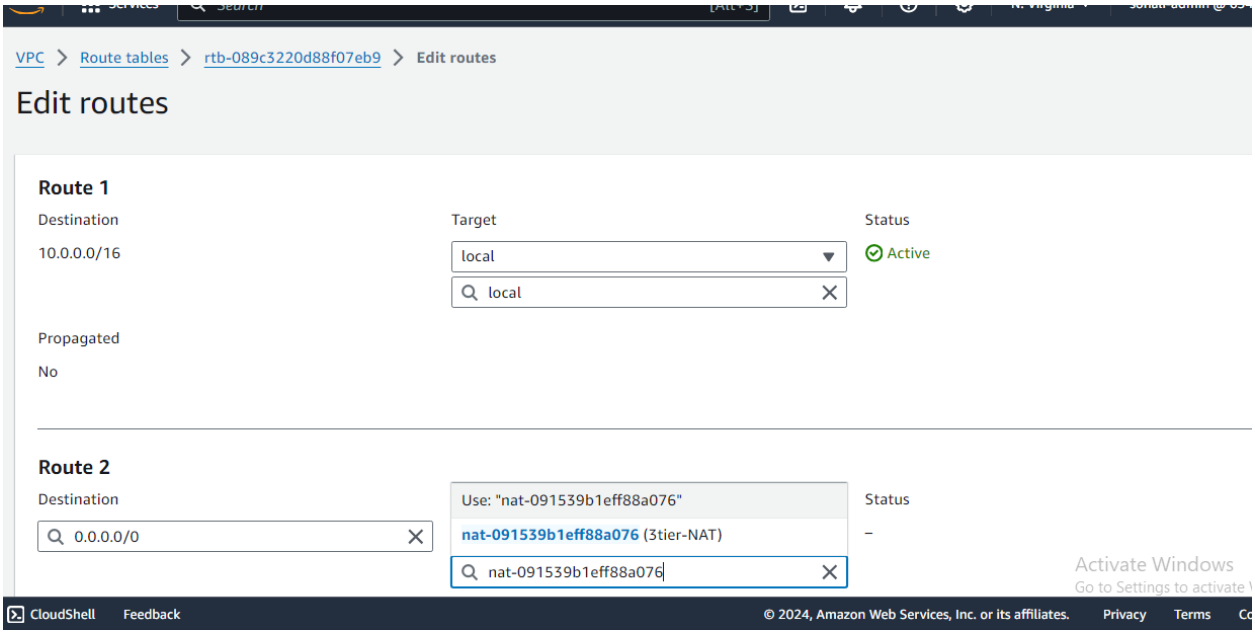
Create one Route table that is my created RT



- Associate that Route table with two private Subnet

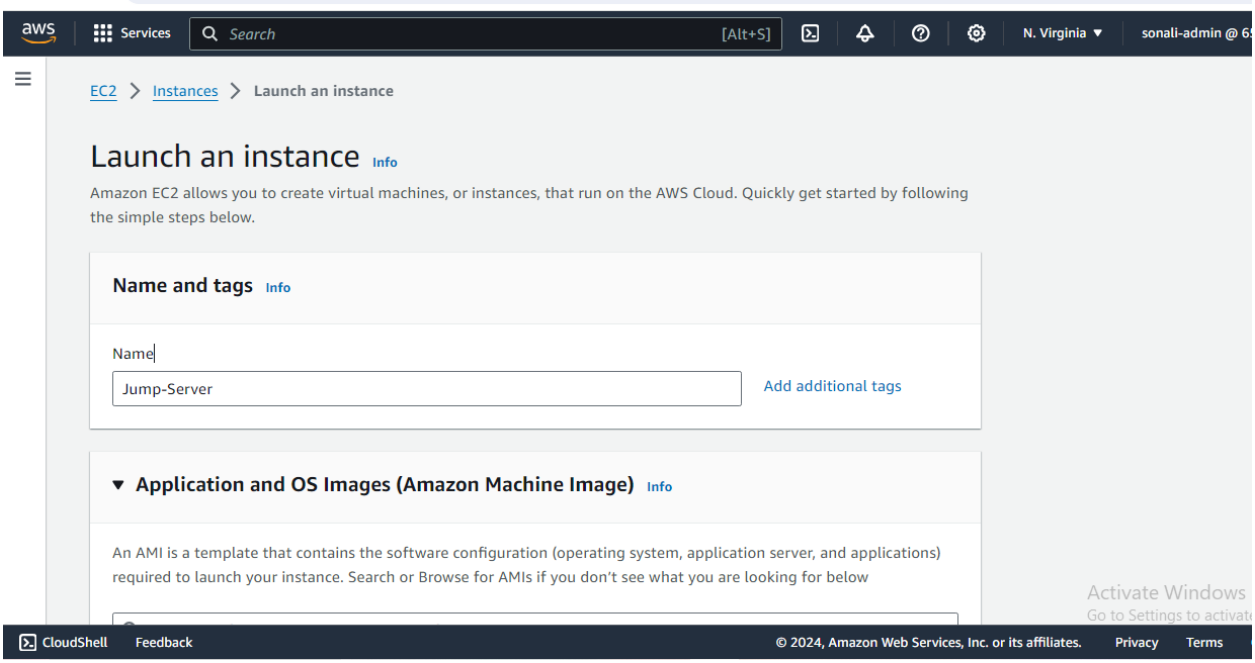


- Add Route to NAT Gateway into Recently created Route Table

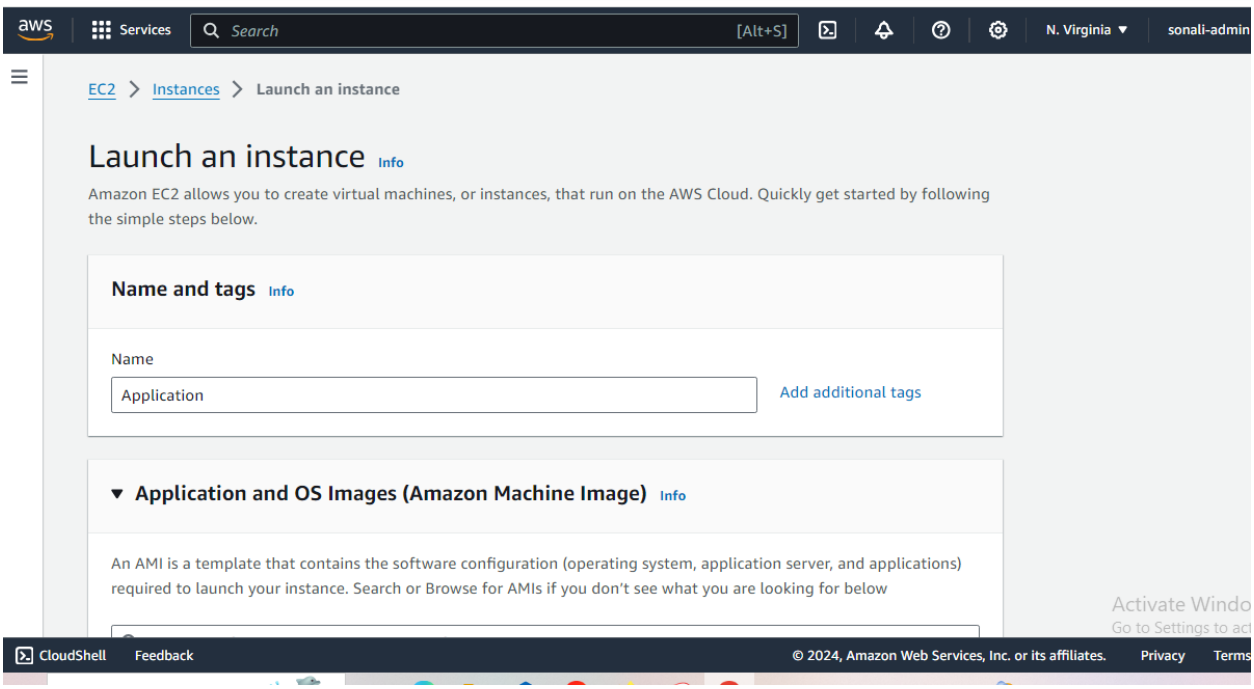


Launch the Instance for public server and give name to that as Jump Server

- Add SSH Inbound Rule Security Group
- Add **8080 Port in Security Group**

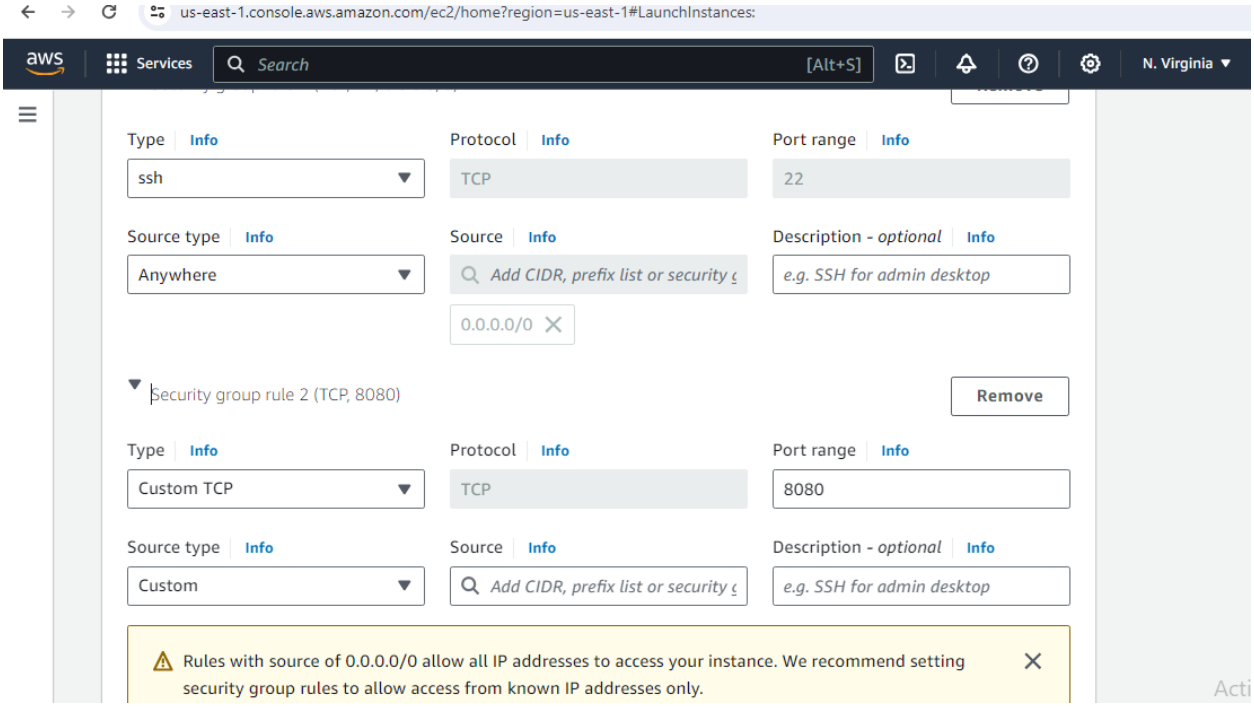


- Create Second Instance for another server in private subnet in different Availability Zone
- Give Name to that as Application

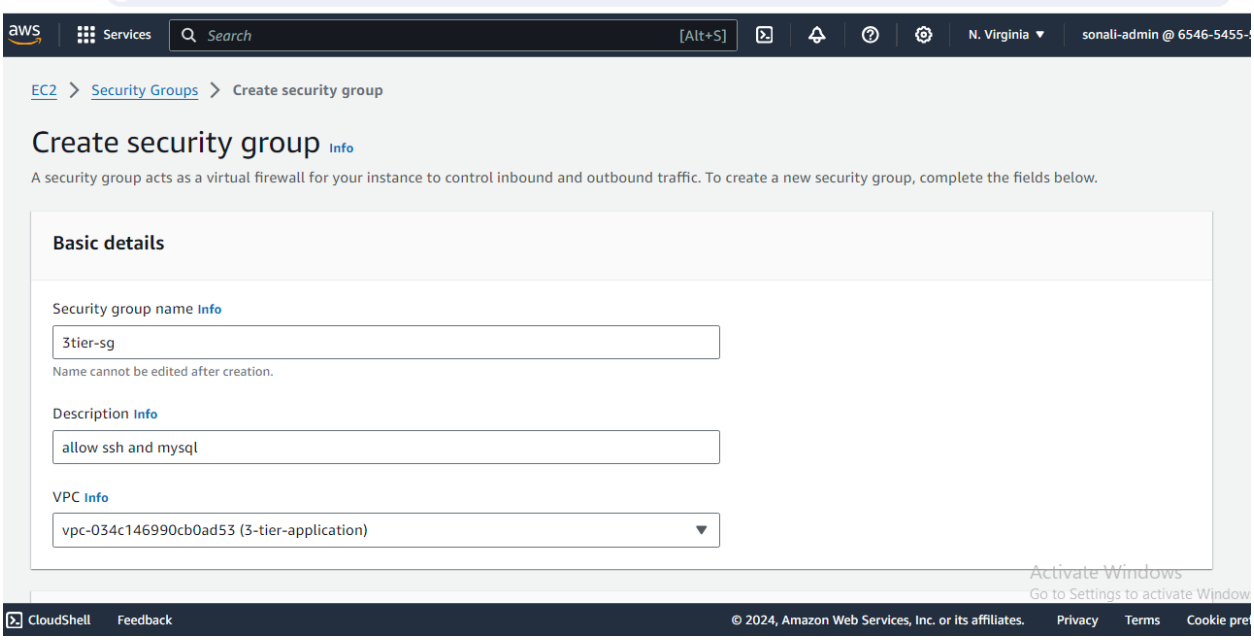


- Add SSH Inbound Rule Security Group

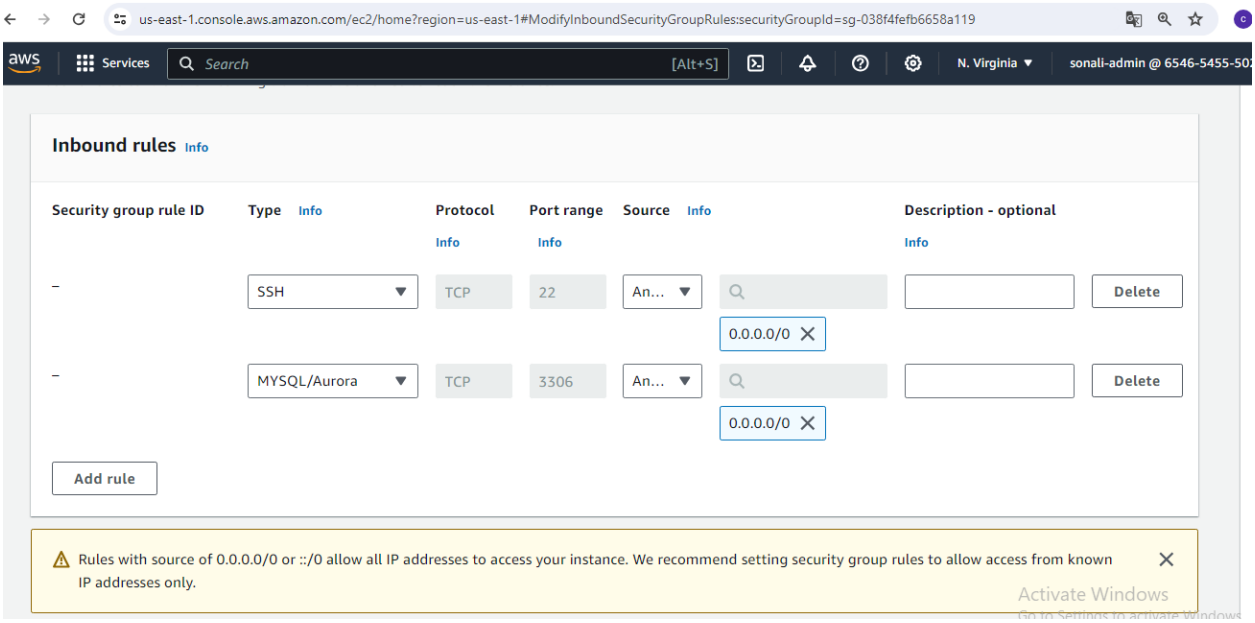
- Add **8080 Port in Security Group**



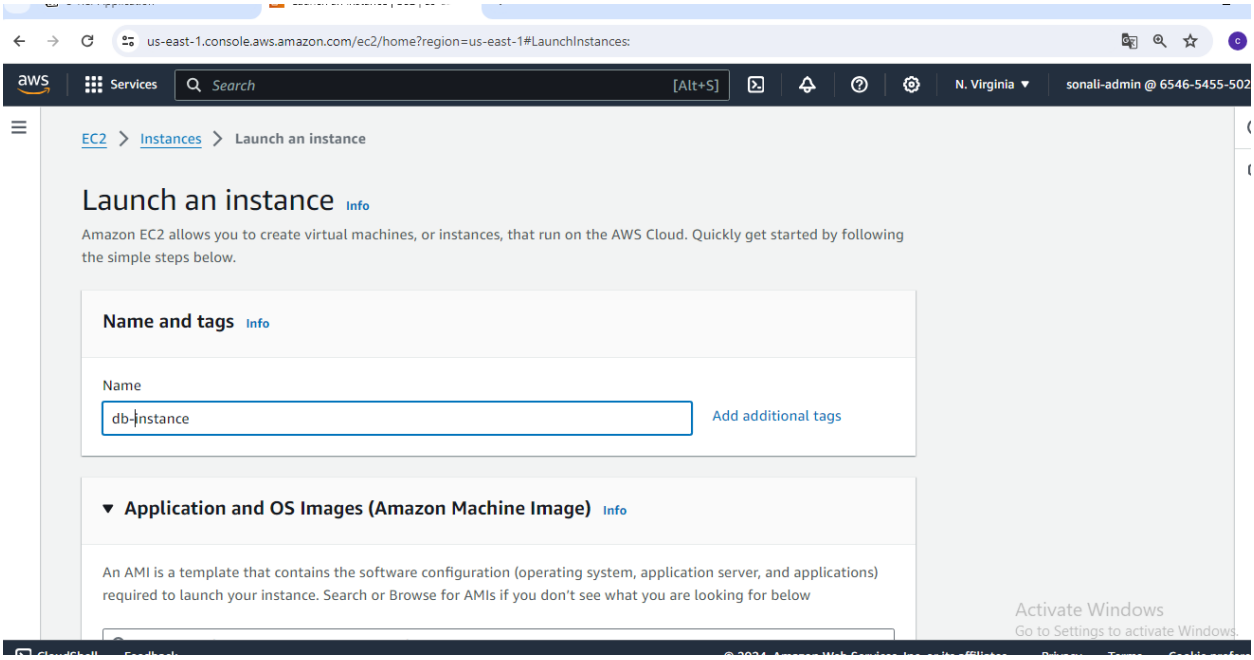
- Create One Security Group For database and Database Instance that is 3tier-sg



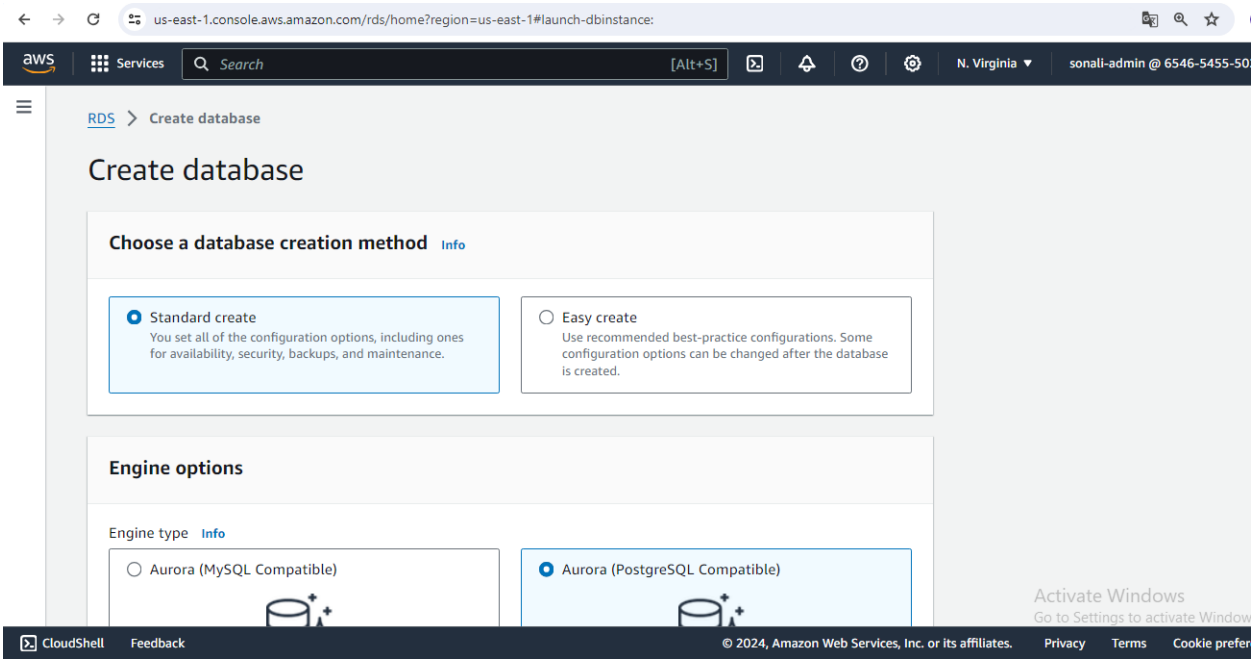
- Add SSH Inbound Rule Security Group
- Add **MYSQL/Aurora that is 3360 Port in Security Group**



- Launch DB-Instance and attach security group that we recently created that is 3tier-sg

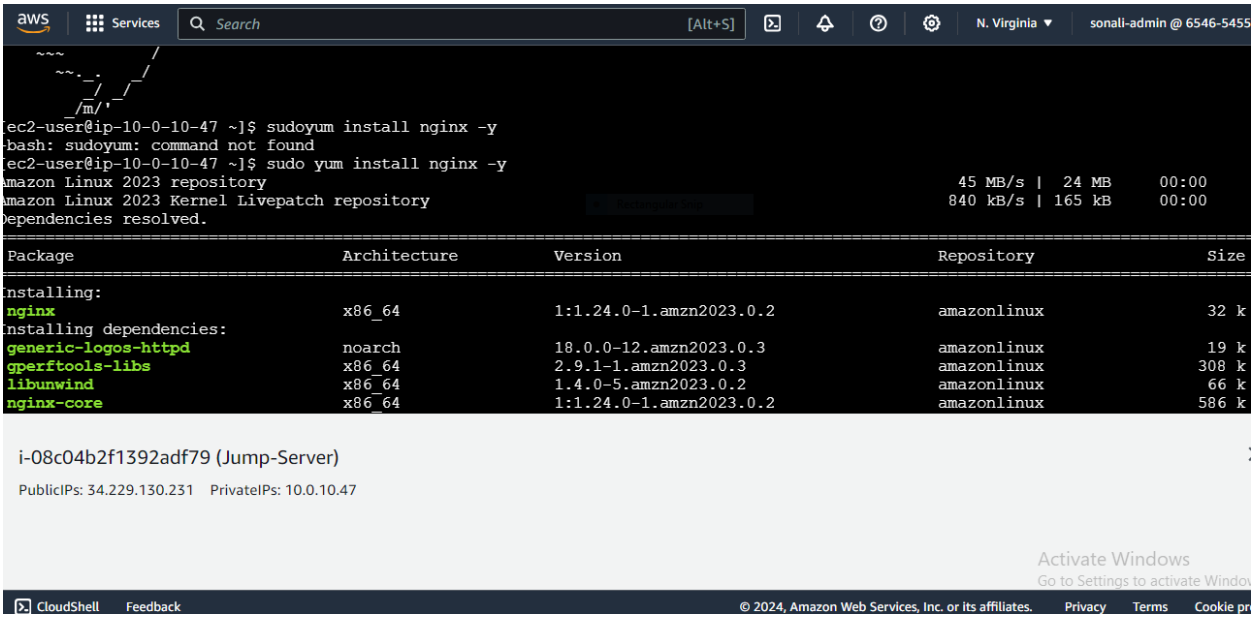


Create RDS and Security Group that is 3tier-sg

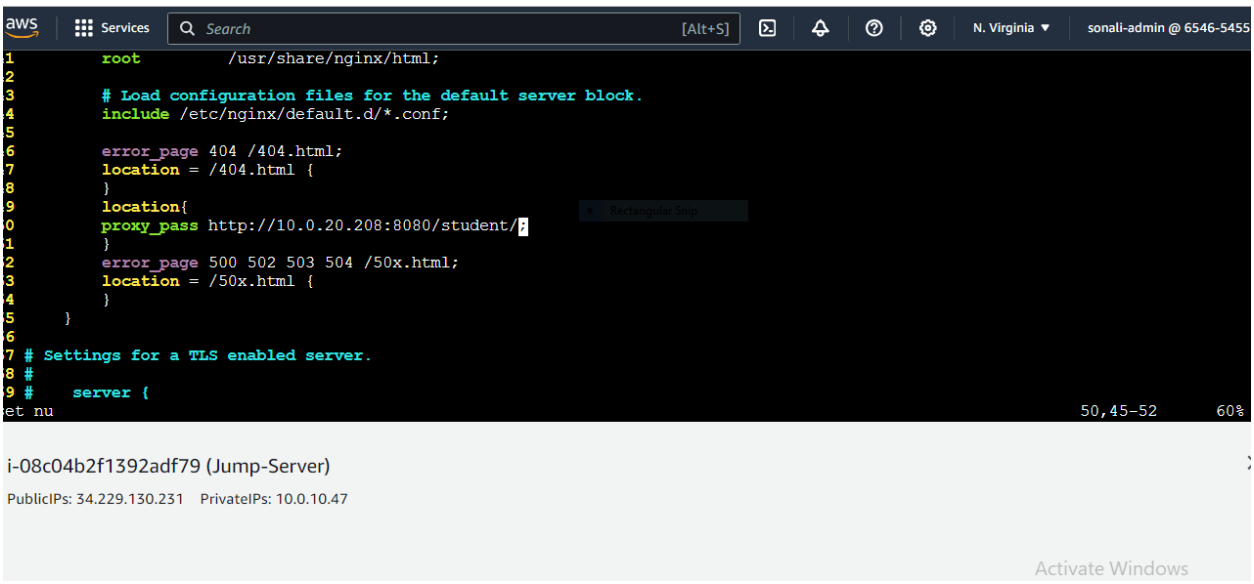


Step 4: Configuration

- Install Nginx in Jump-Server

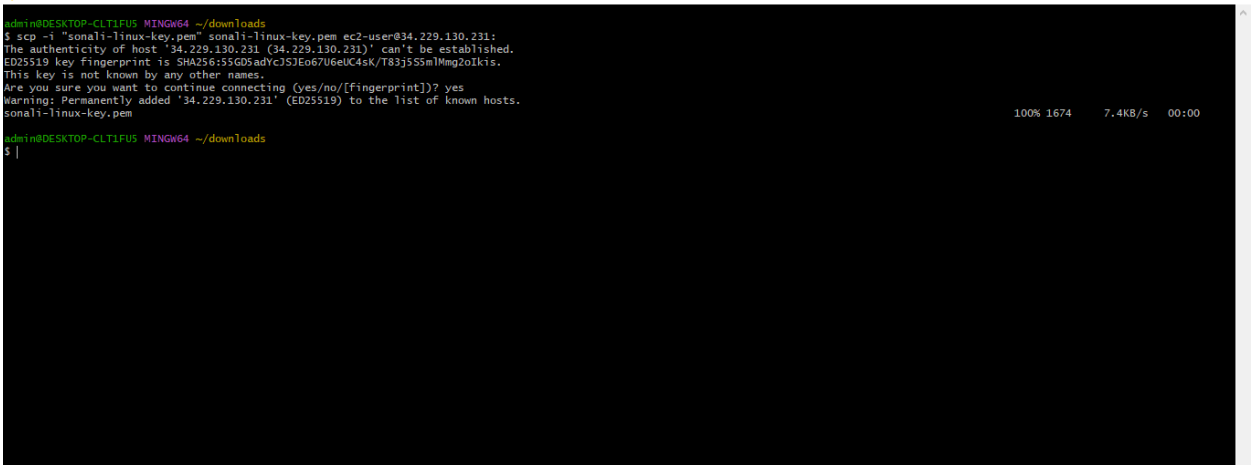


- Change In Configuration File and add the IP of Application Server and other details
- Restart the Nginx

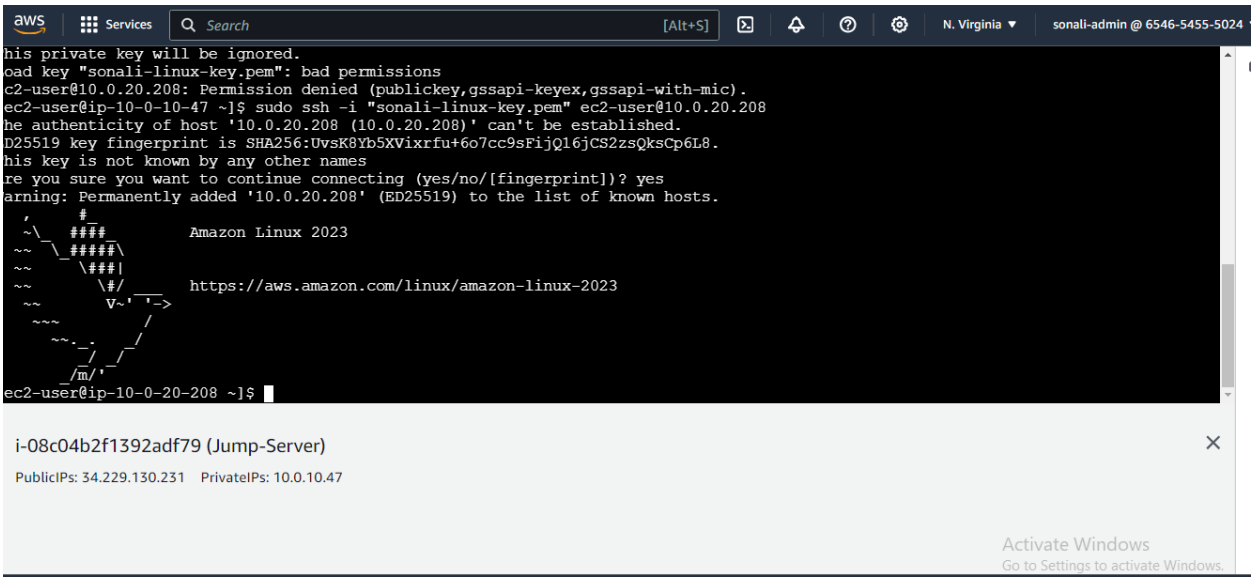


- Do ssh To Application Server For that
- Copy the Key in Jump-Server from your local machine by doing scp

- Through the Git we have copied key

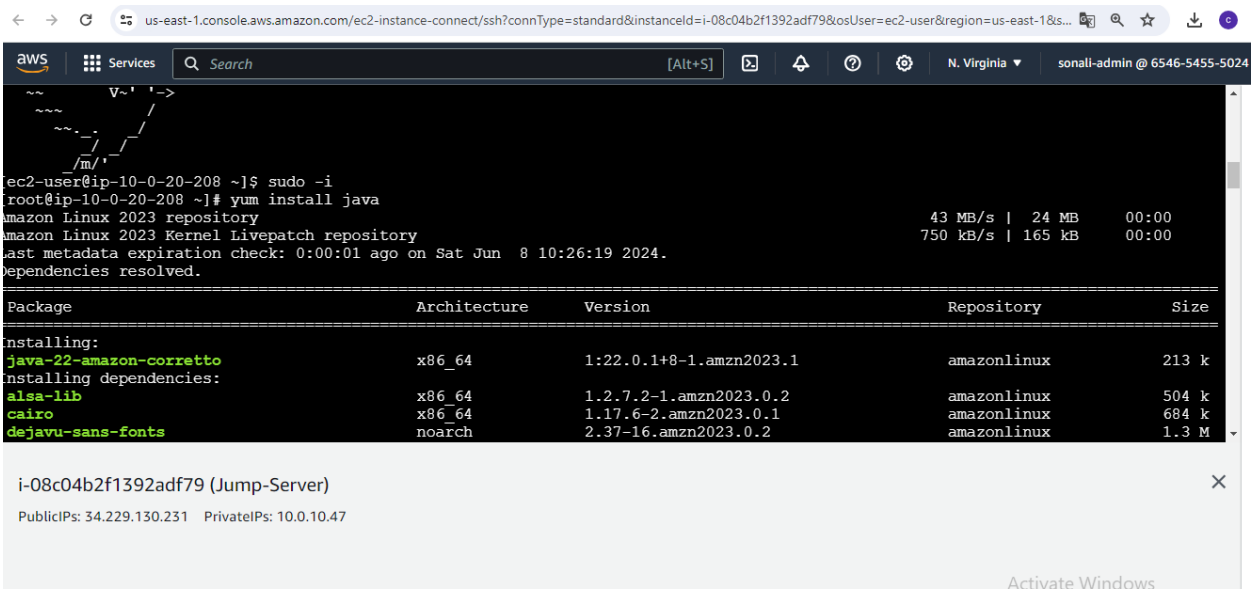


- SSH done From Jump-Server to Application Server



In Application Server

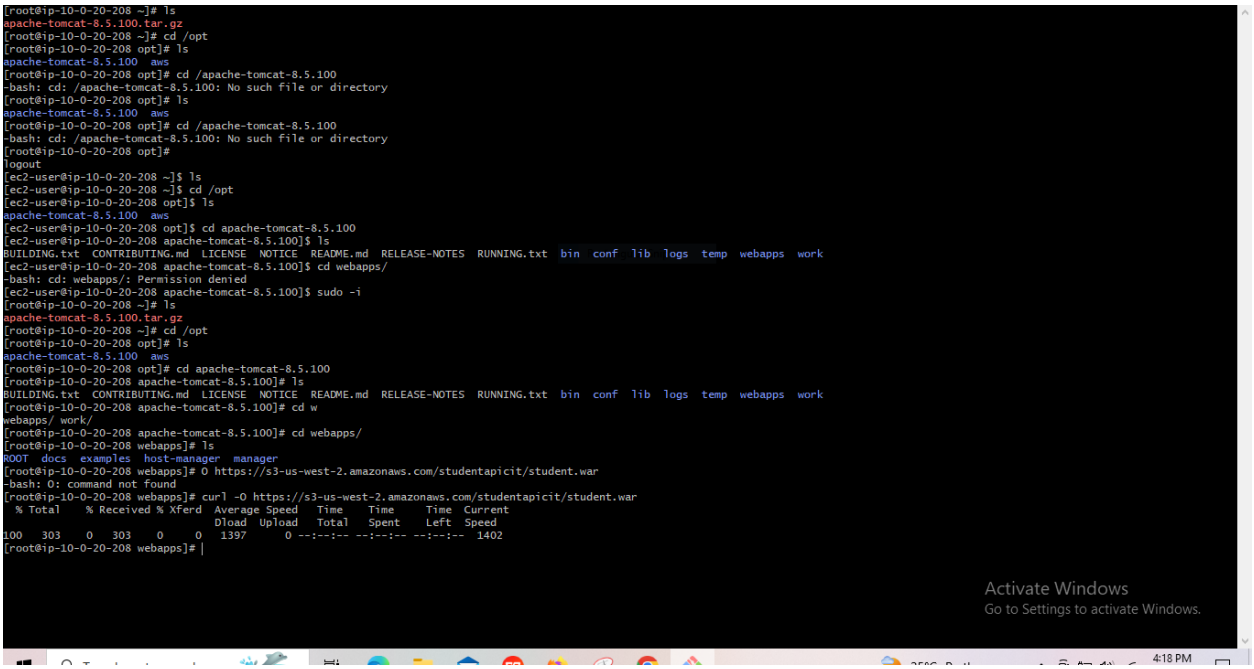
- Install **Java**



- Install Tomcat
- Download Tomcat First by Curl -O and (url of tomcat) option

curl -O https://dlcdn.apache.org/tomcat/tomcat8/v8.5.100/bin/apache-tomcat- 8.5.100.tar.gz

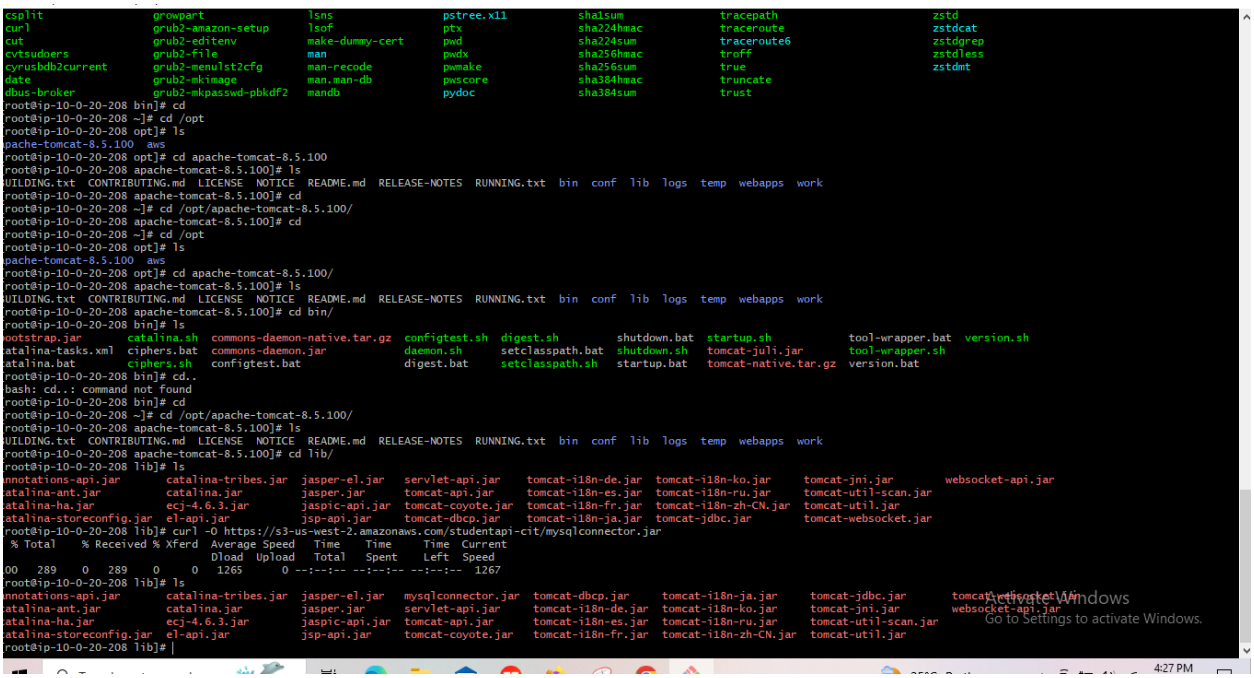
- Unzip that Tar –xvf apache-tomcat-8.5.100.tar.gz -C /opt



Cd apache-tomcat-8.5.100/

- Cd webapps
- curl -O

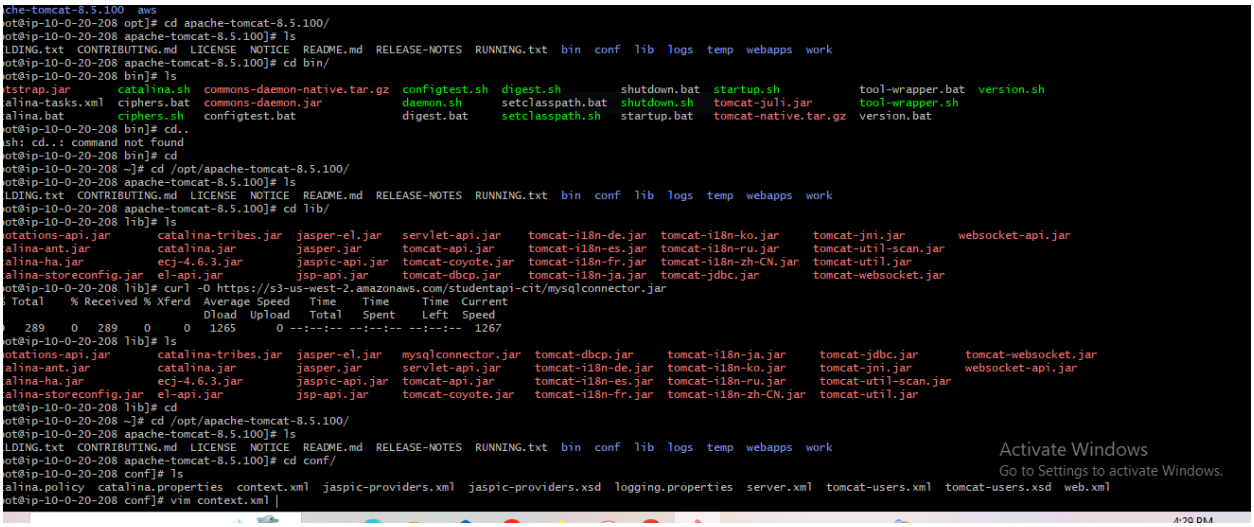
https://s3-us-west-2.amazonaws.com/studentapicit/student.war



Cd ../lib

- curl -O

https://s3-us-west-2.amazonaws.com/studentapi-cit/mysqlconnector.jar



- Cd ../conf/
- Vim context.xml
- (in context tab [last of page]) <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource" maxTotal="500" maxIdle="30" maxWaitMillis="1000" username="admin" password="12345678" driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://endpoint:3306/studentapp?useUnicode=yes&characterEncoding=utf8"/>
- Cd ../bin
  - ./catalina.sh start
  - Exit

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<!-- The contents of this file will be loaded for each web application -->
<Context>

    <!-- Default set of monitored resources. If one of these changes, the    -->
    <!-- web application will be reloaded.                                   -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->
    <Resource name="jdbc/TestDB"
auth="Container" type="javax.sql.DataSource" maxTotal="500"
maxIdle="30" maxWaitMillis="1000"
 username="admin" password="12345678"
driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://database-1.c9e8oiomeuei.us-east-1.rds.amazonaws.com:3306/studentapp?
useUnicode=yes&amp;characterEncoding=utf8"/>
</Context>
~
~
~
~
~
~
~
~
~
~
~
~
"context.xml" 37L, 1675B                                              37,10
```


- Do SSH From Jump-Server to DB-Instance
- Ssh –i key-name ec2-user@dbinstanceip (connected to db. instance)
- Yum install mariadb105 –y

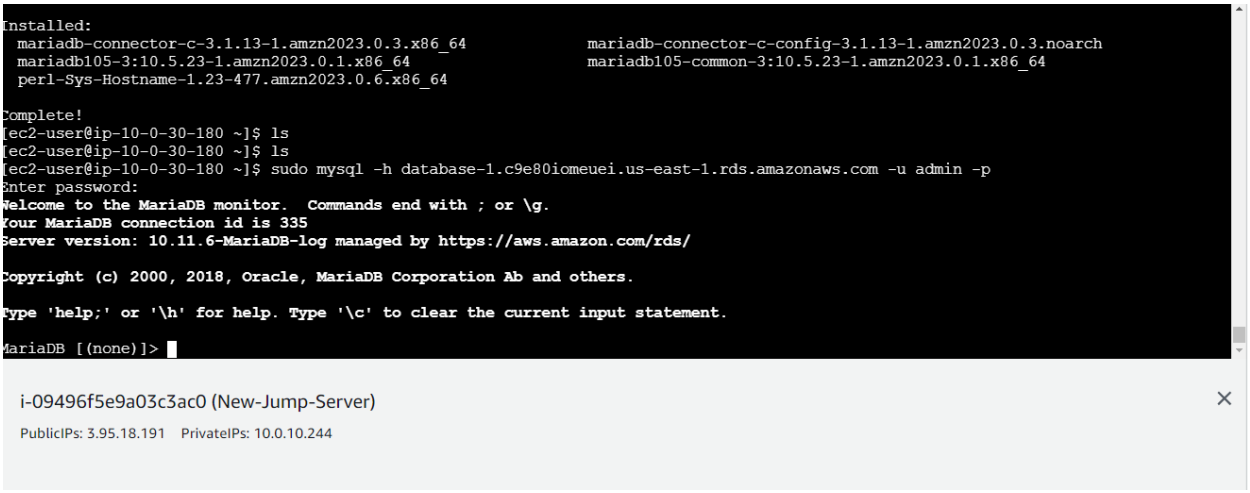
```
Load key "sonali-linux-key.pem": bad permissions
ec2-user@10.0.30.180: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-0-10-244 ~]$ sudo ssh -i "sonali-linux-key.pem" ec2-user@10.0.30.180
The authenticity of host '10.0.30.180 (10.0.30.180)' can't be established.
ED25519 key fingerprint is SHA256:muAU1qTFw36We72WXFhYq0hBi32vWk2cU0dSbJReoj8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.30.180' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-30-180 ~]$ ls
[ec2-user@ip-10-0-30-180 ~]$ sudo yum install mariadb105 -y
```

i-09496f5e9a03c3ac0 (New-Jump-Server)

PublicIPs: 3.95.18.191 PrivateIPs: 10.0.10.244

• Mysql –h rdsendpoint –u admin –p(connected to RDS)
• Create database;

Use database;
• CREATE TABLE if not exists students(student_id INT NOT NULL
AUTO_INCREMENT,
student_name VARCHAR(100) NOT NULL,
student_addr VARCHAR(100) NOT NULL,
student_age VARCHAR(3) NOT NULL,
student_qual VARCHAR(20) NOT NULL,
student_percent VARCHAR(10) NOT NULL,
student_year_passed VARCHAR(10) NOT NULL,
PRIMARY KEY (student_id)
);
• Show table



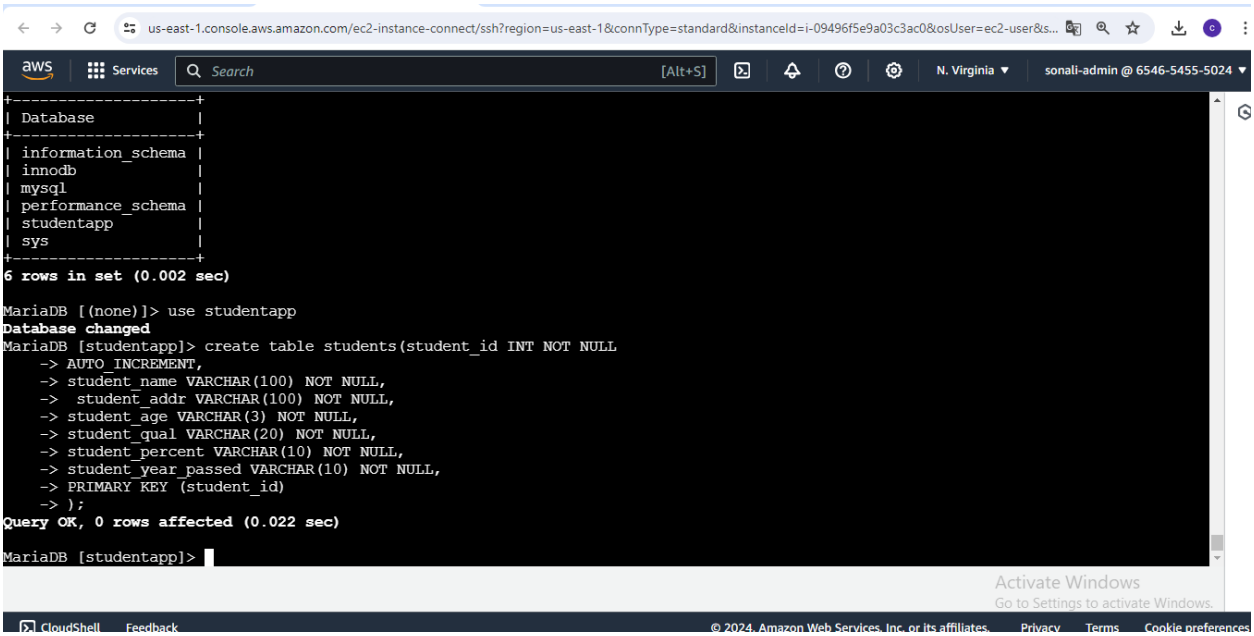
```
    -> show databases;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the
right syntax to use near 'show databases
show databases' at line 2
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.005 sec)

MariaDB [(none)]> create database studentapp;
Query OK, 1 row affected (0.005 sec)

MariaDB [(none)]>
```

i-09496f5e9a03c3ac0 (New-Jump-Server)

PublicIPs: 3.95.18.191 PrivateIPs: 10.0.10.244



Go to google and hit our jump server public IP

Student Registration Form

Student Name	
Student Address	
Student Age	
Student Qualification	
Student Percentage	
Year Passed	

register