How to install and configure SonarQube with Database on AWS EC2 Ubuntu

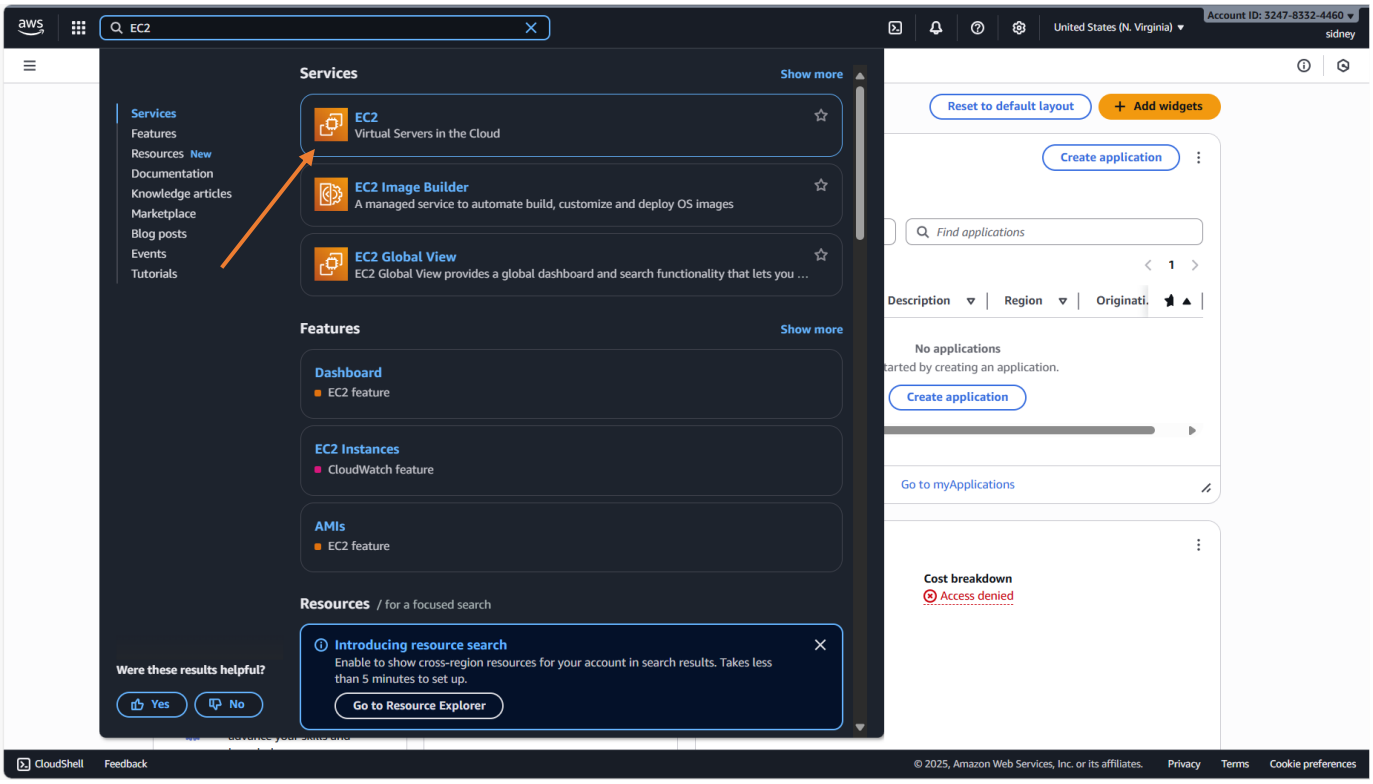
In this document, I will show how to install SonarQube with database on Ubuntu EC2 instance, This could be a step while building a CI/CD pipeline with Jenkins, Gitlab or CircleCI.

STEP 1: Launch an Ubuntu EC2 instance and SSH Connect to it

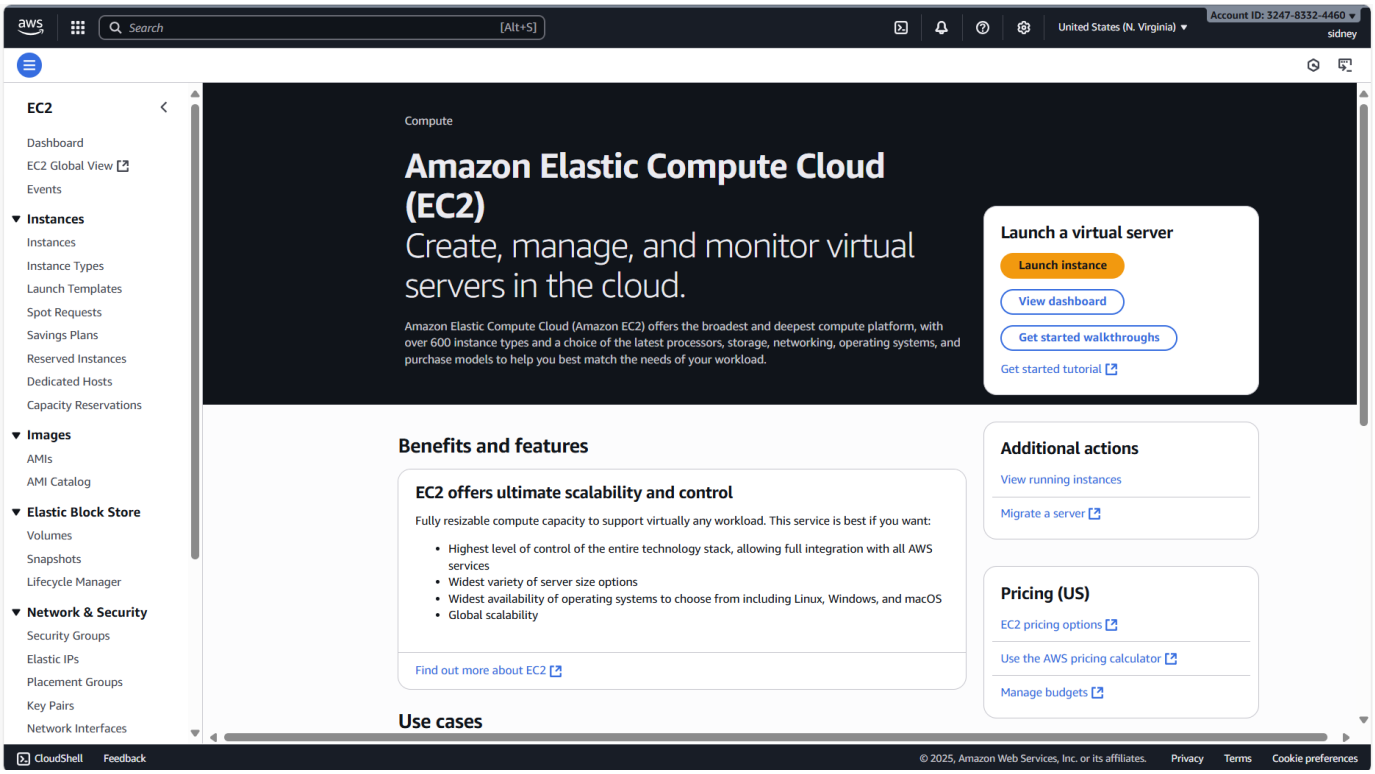
We will launch an ubuntu EC2 instance and later on SSH connect to it to install our dependencies.

PART 1: Launch an Ubuntu EC2 instance

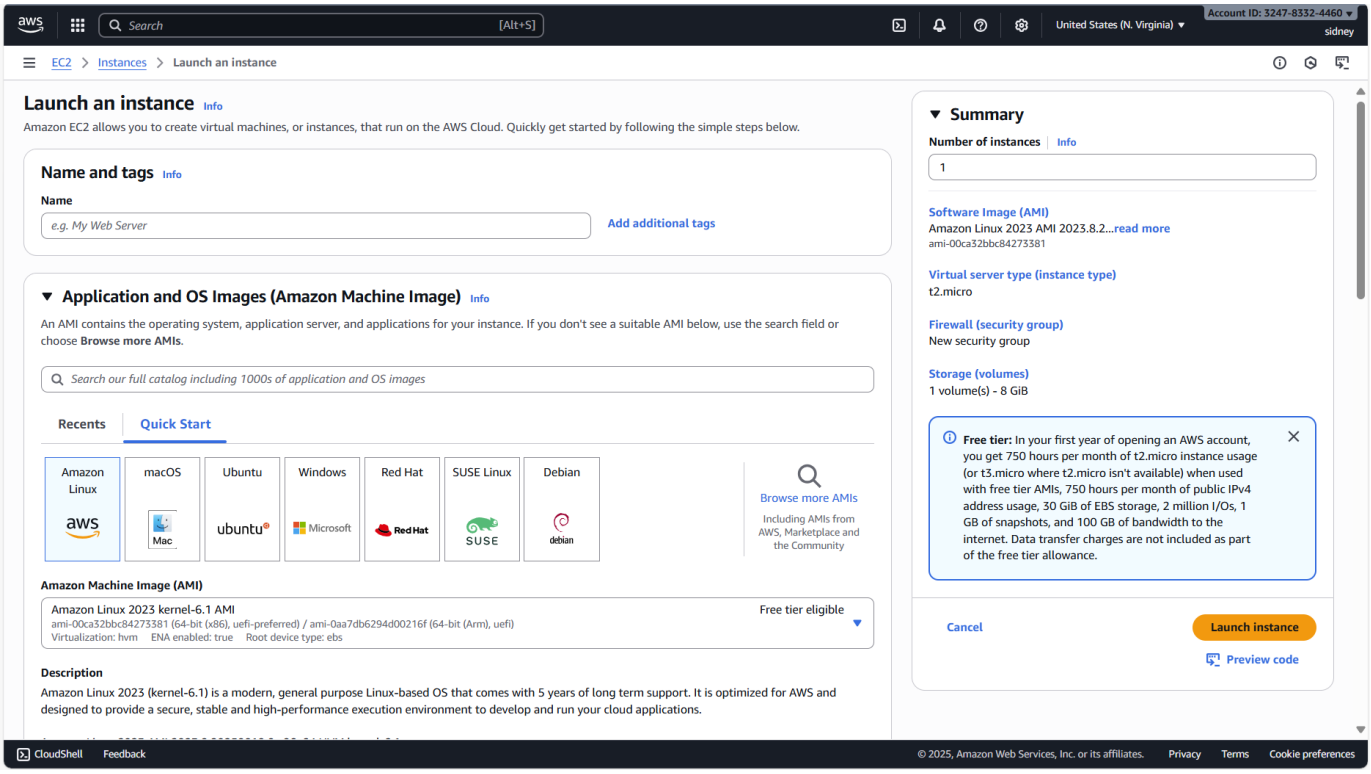
Go to AWS Management console and search for "**EC2**" Instance.



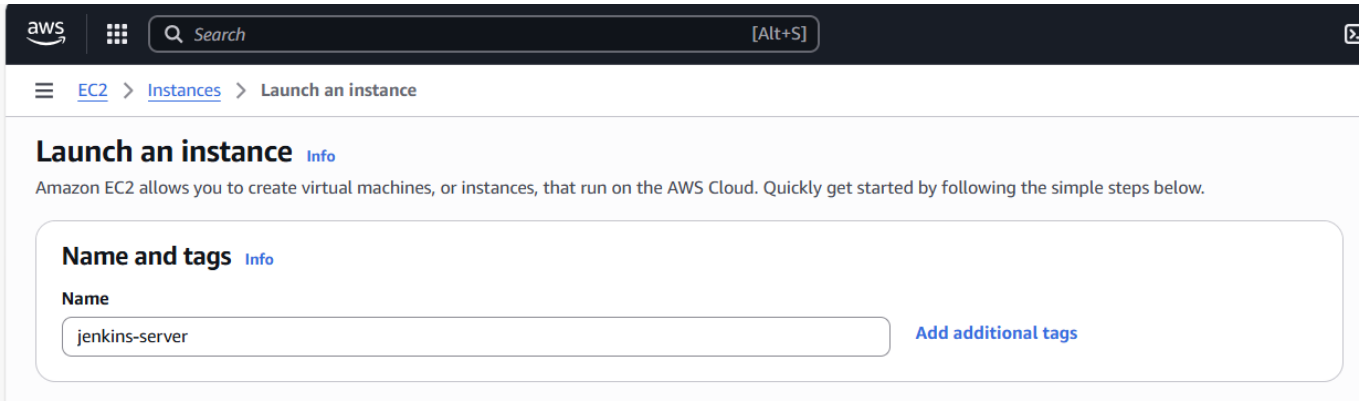
Click on "**EC2**"



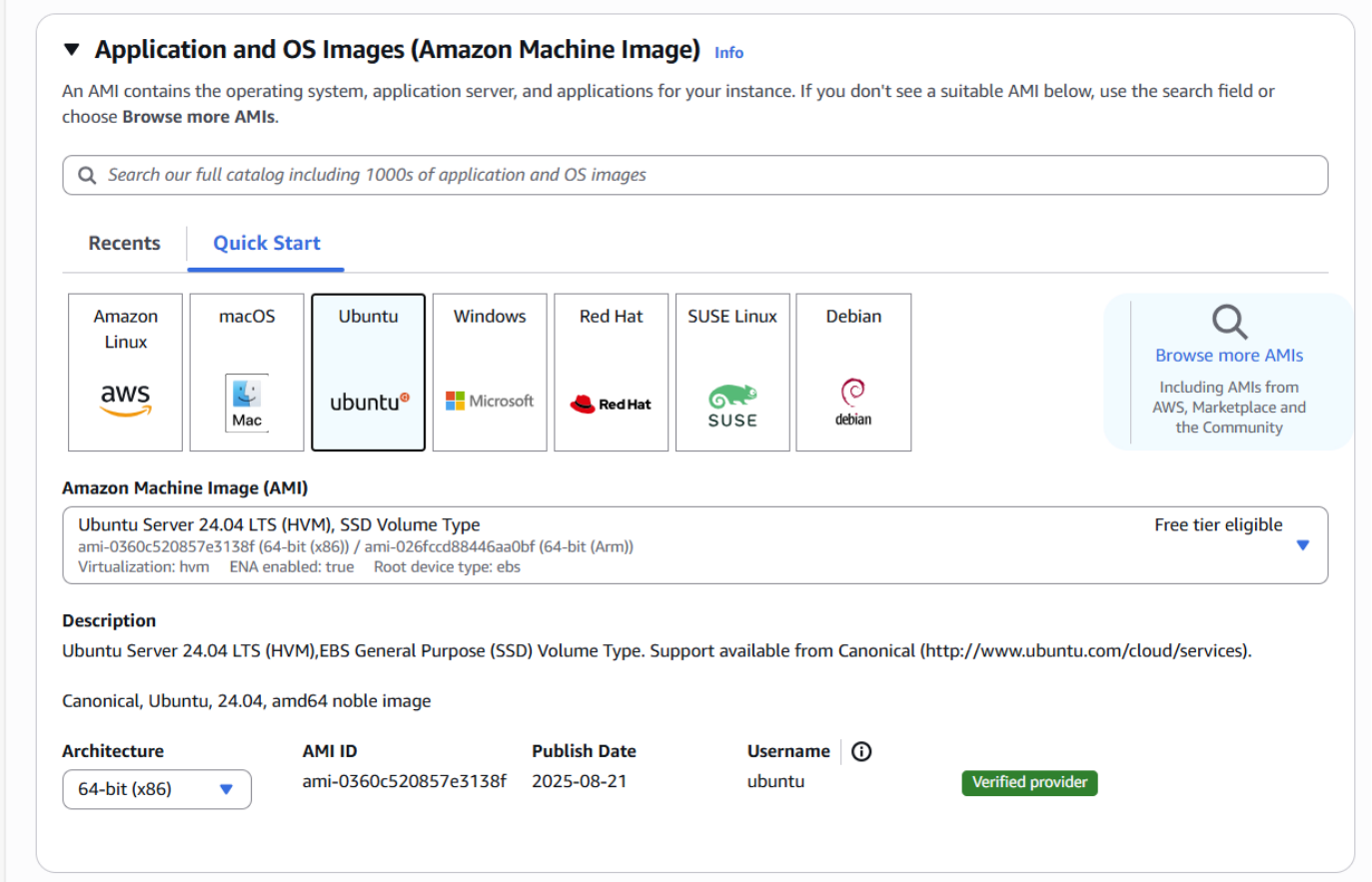
Click on "**Launch Instance**"



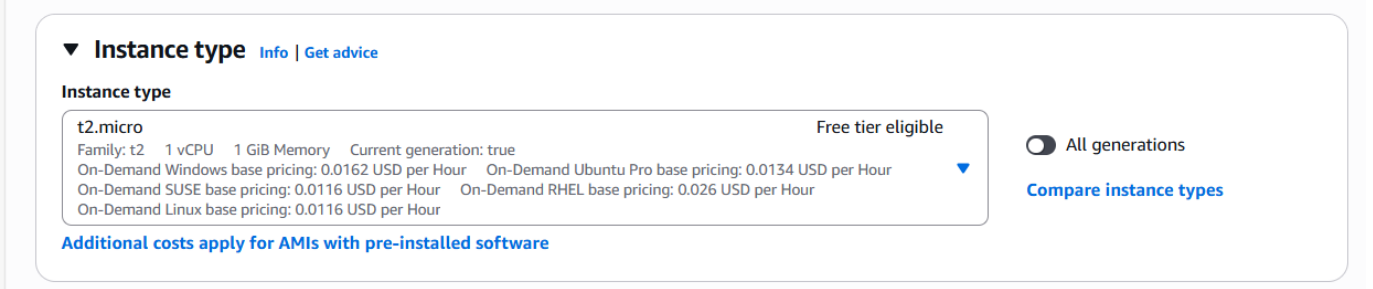
We will call the instance "**sonar-server**"



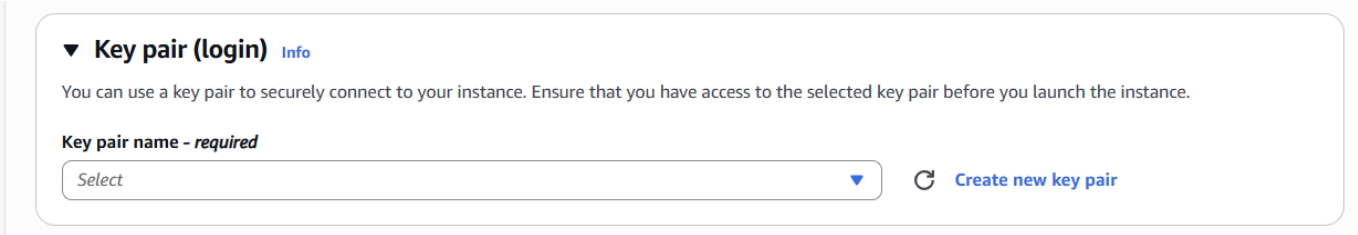
Scroll down to "**AMI**", select "**ubuntu**"



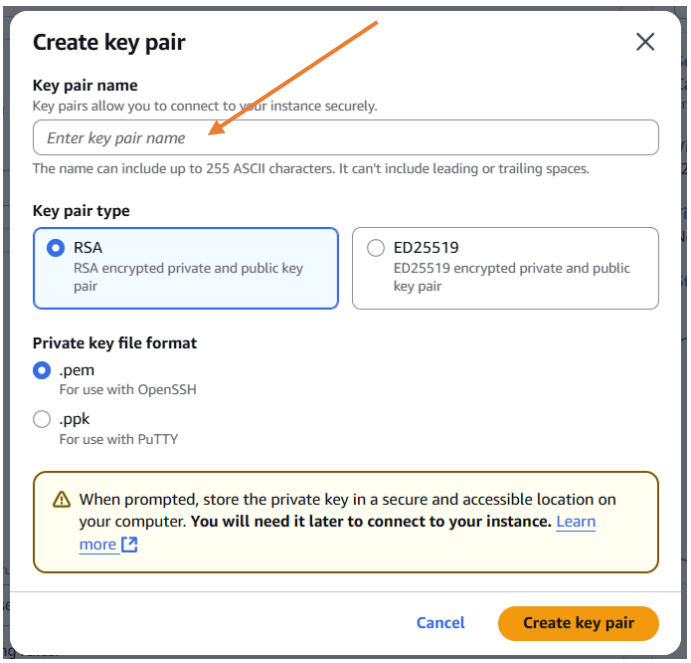
Scroll down to "**Instance Type**" and select "**t2.micro**"



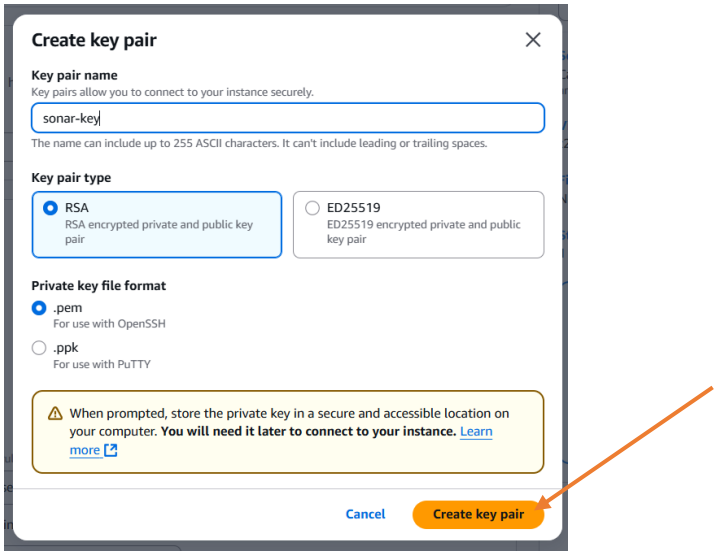
Scroll down to “**Key Pair**”



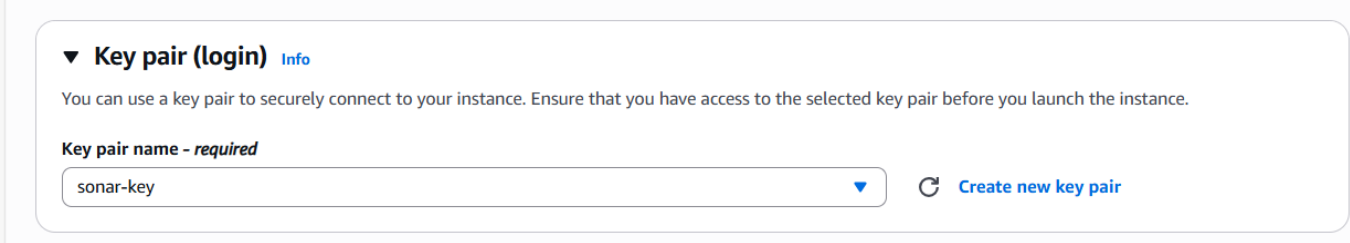
Click on “**Create new key pair**”



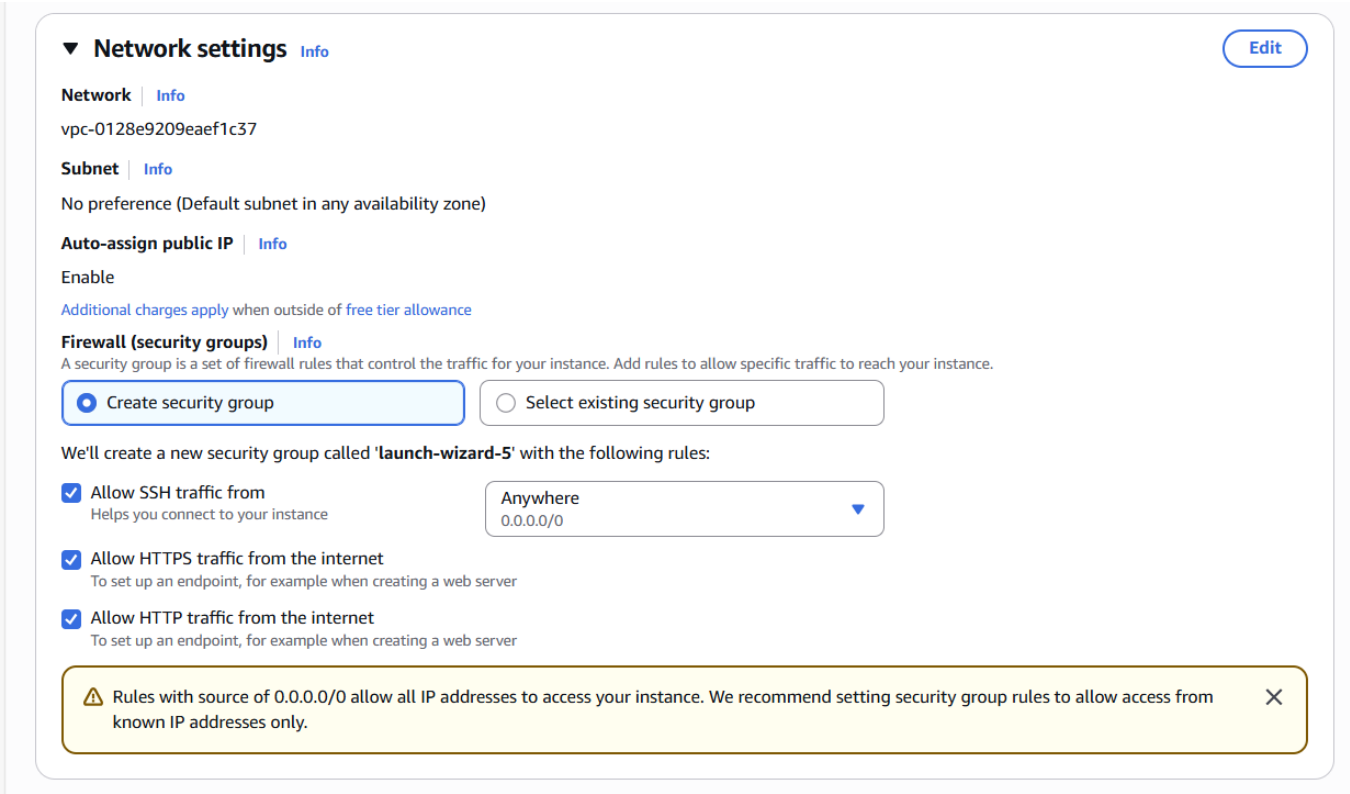
Give the key pair a name, I will call it “**sonar-key**”



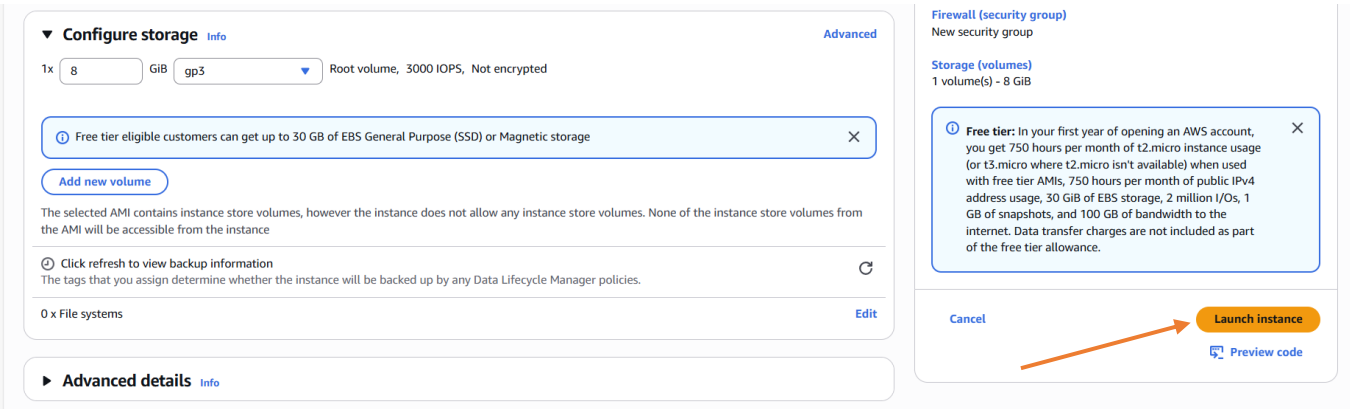
Click on “**Create key pair**”



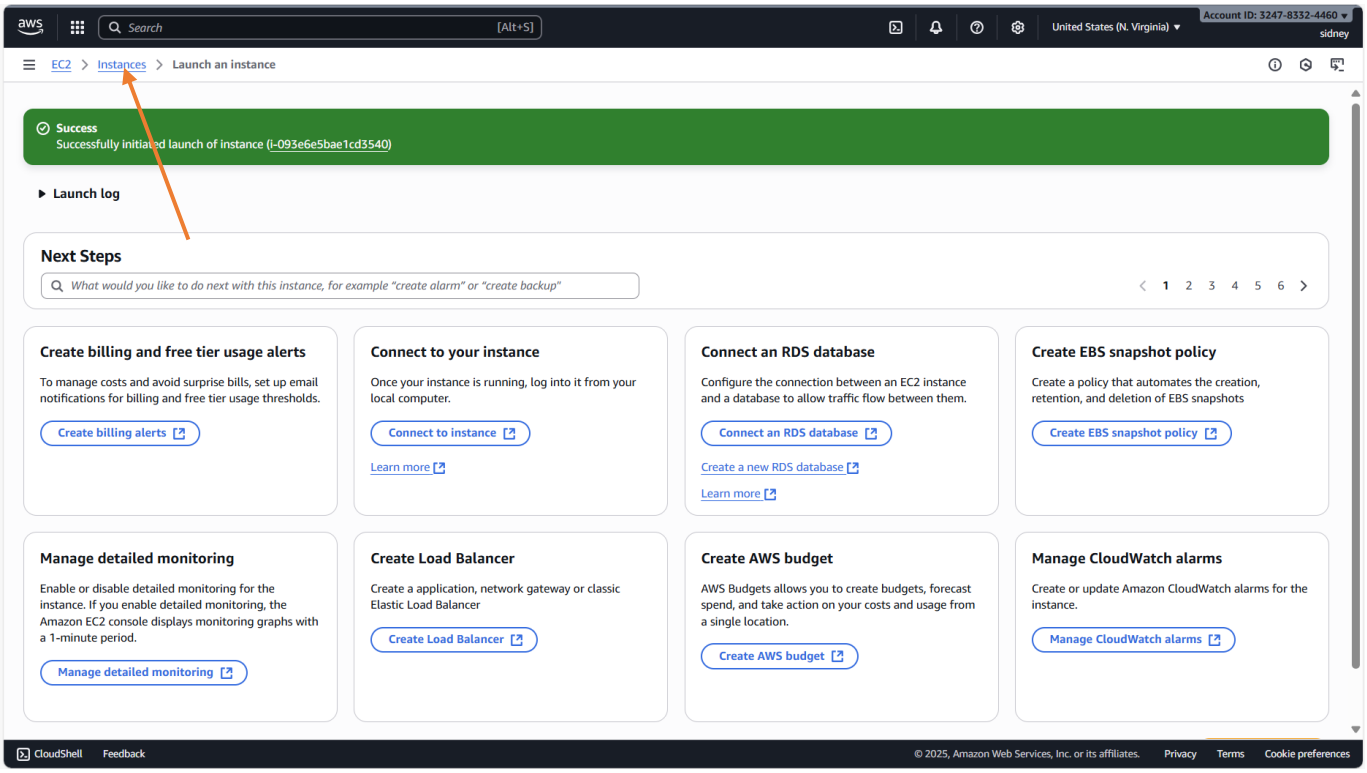
Scroll down to "**Network Settings**". Select "**Allow HTTP traffic from the internet**", "**Allow HTTP traffic from the internet** and "**Allow SSH traffic from**".



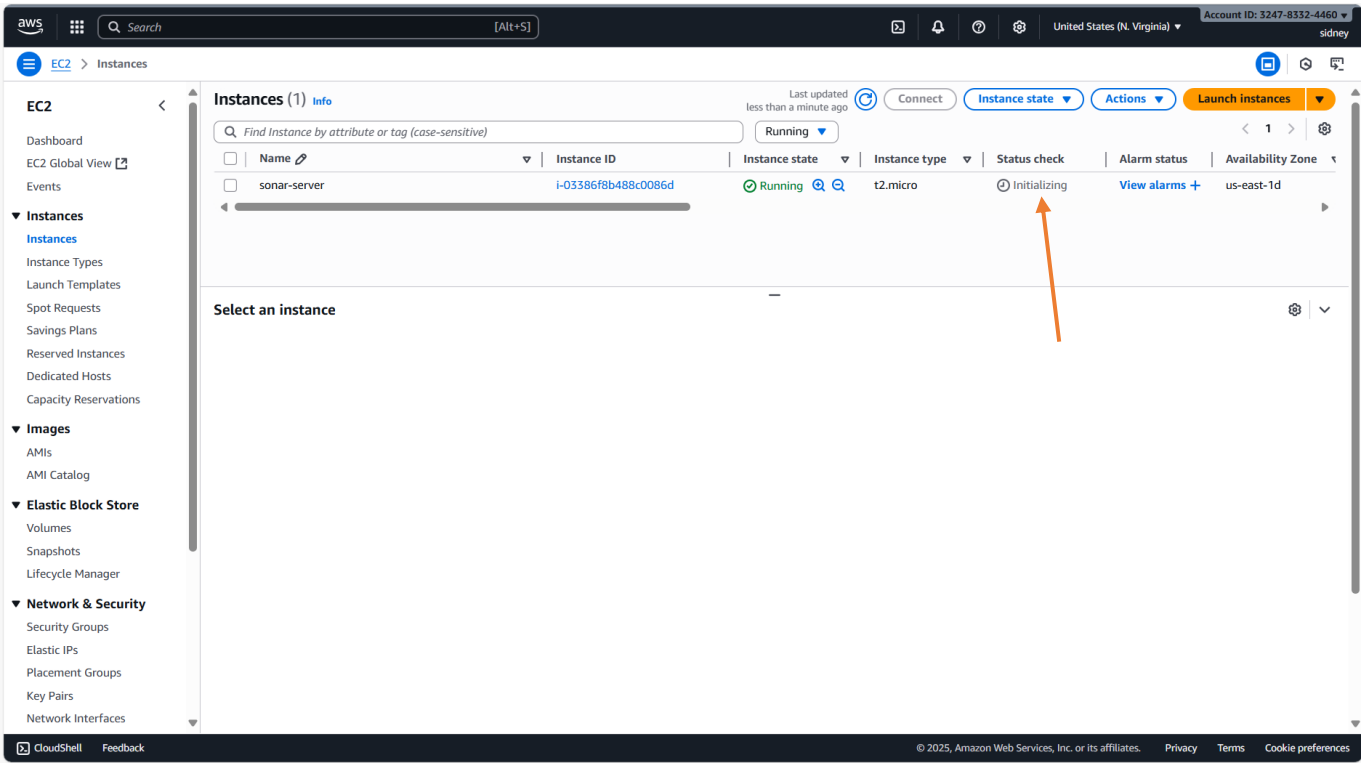
Scroll down to the end



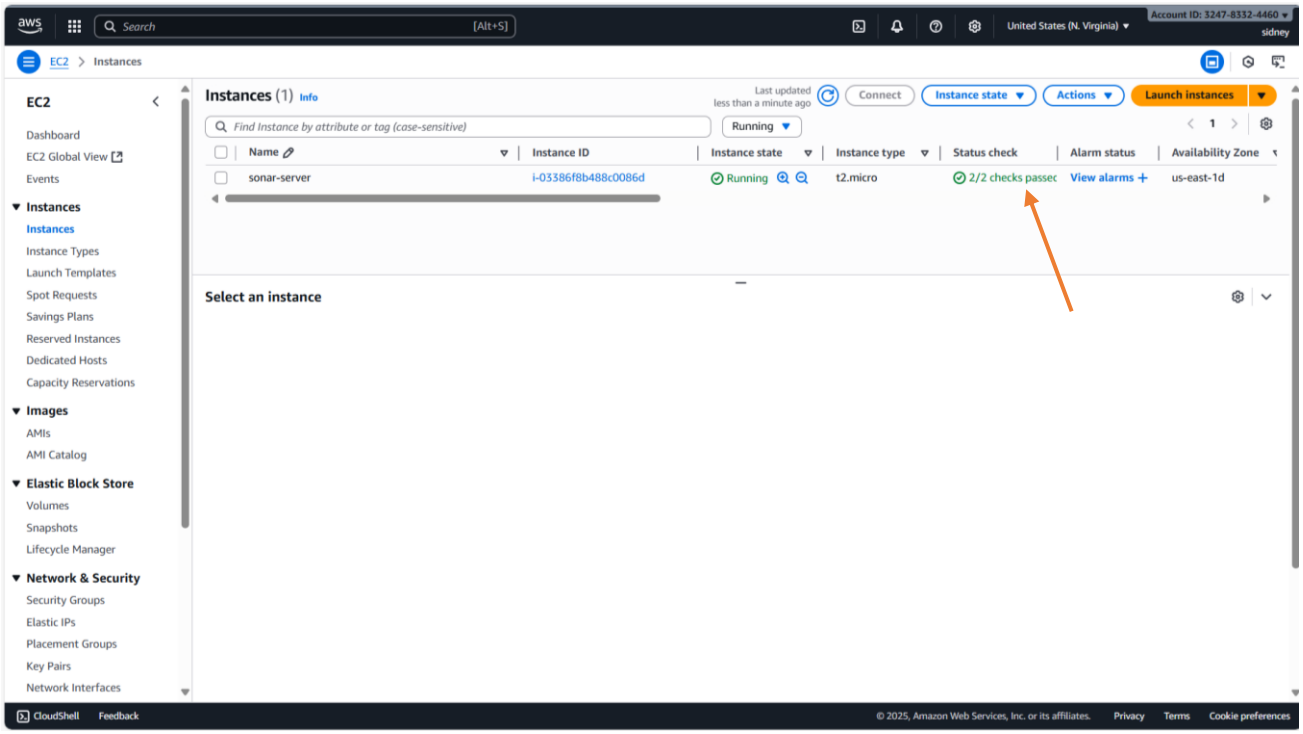
Click on "**Launch Instance**"



Click on "**Instances**"



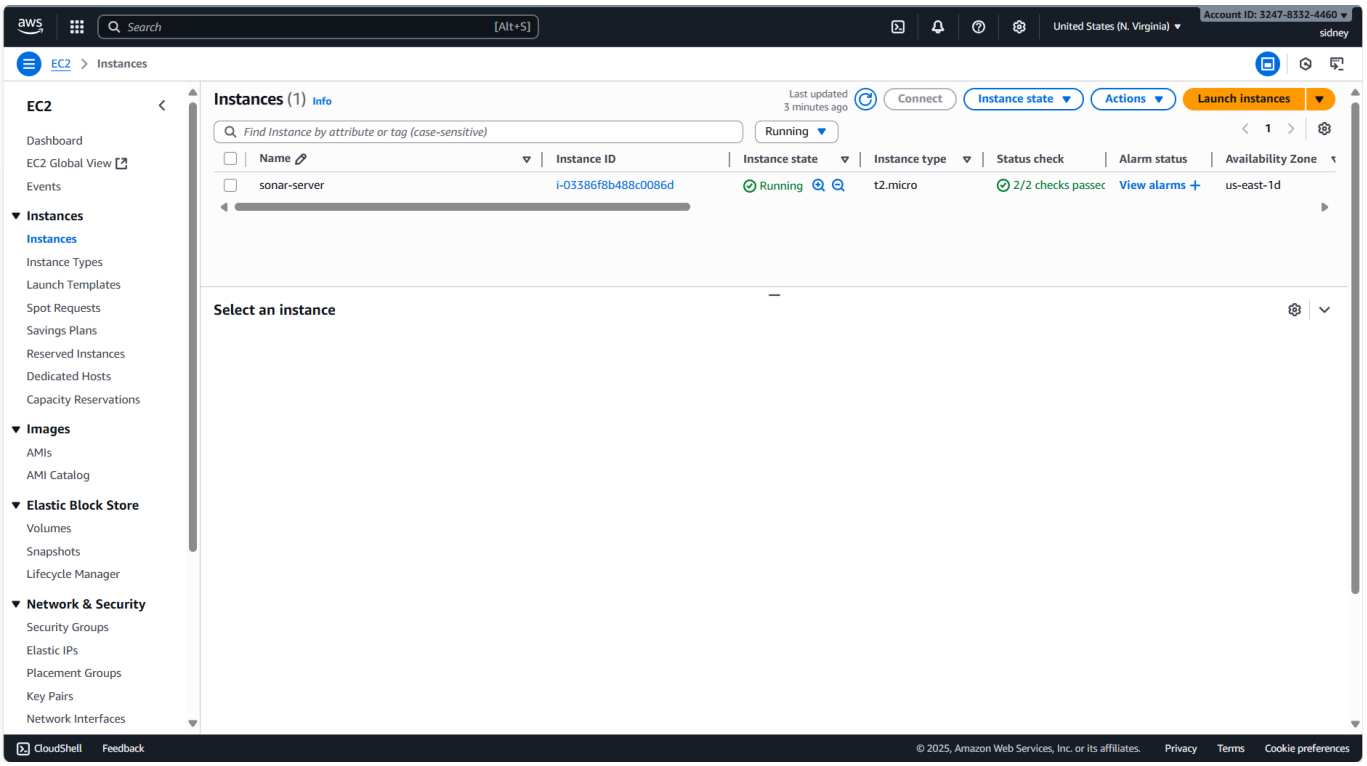
You can see that our just created instance is initializing. Wait for it to pass the "**2/2 check**"



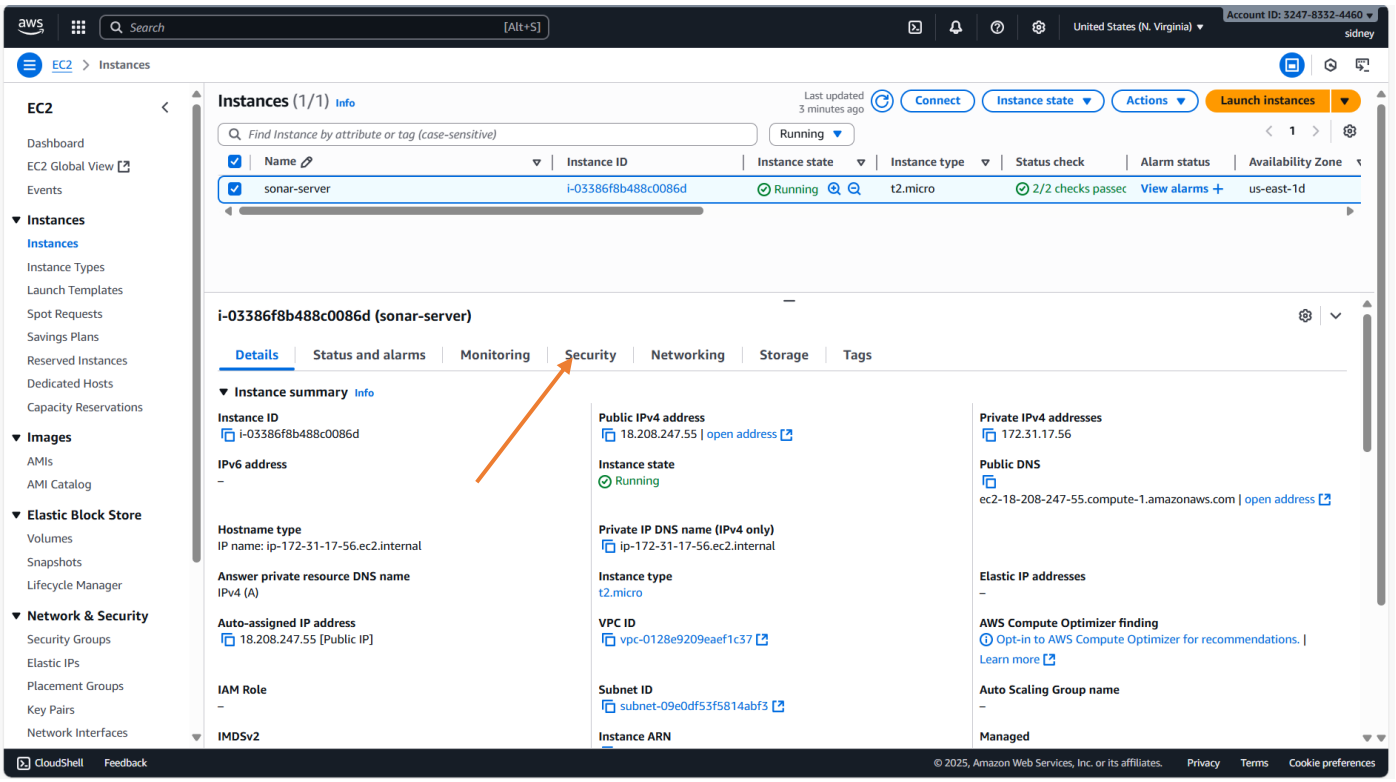
The instance has passed the "**2/2 check**"

PART 2: Add Port 9000 for SonarQube

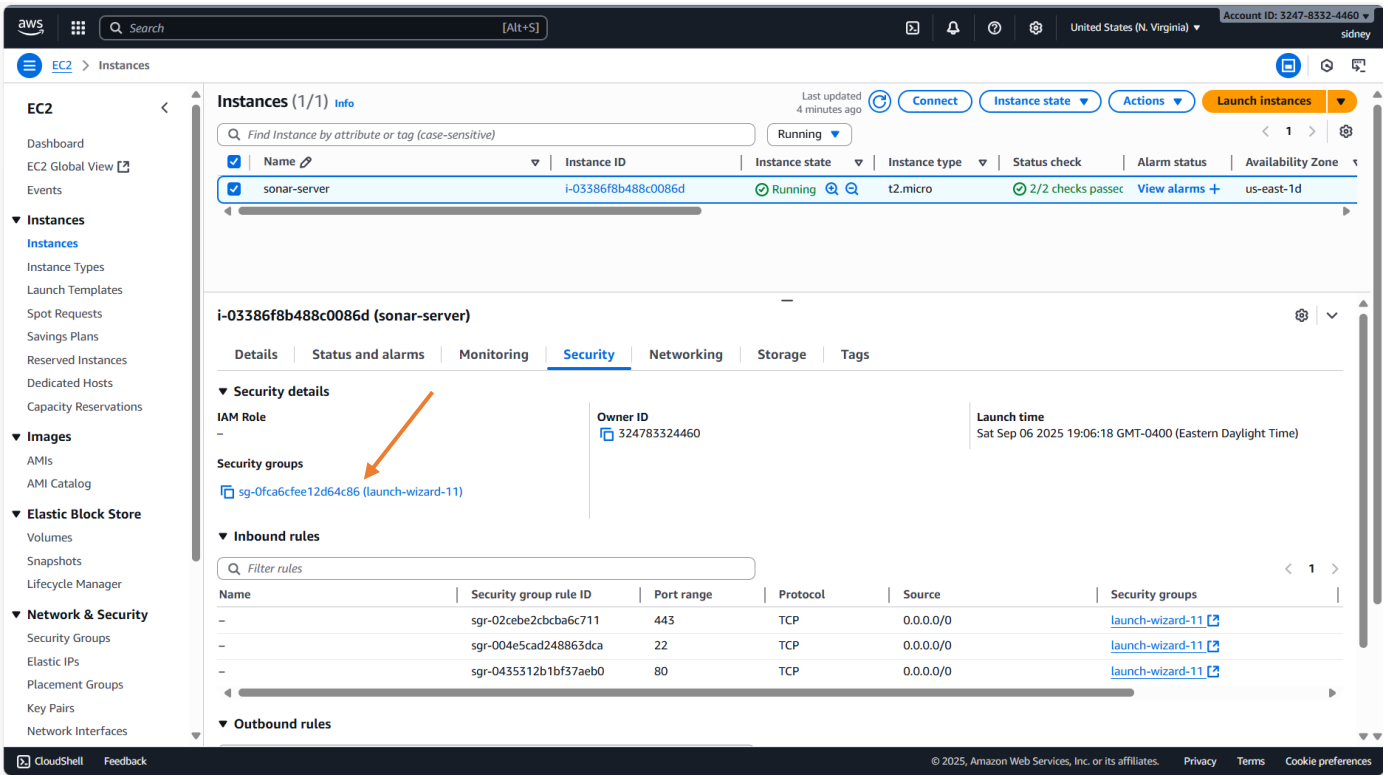
We will add port 9000 that will be used to access SonarQube through the browser. Now, we have to add port 9000 to enable access SonarQube from our browser. Go to our EC2 instance



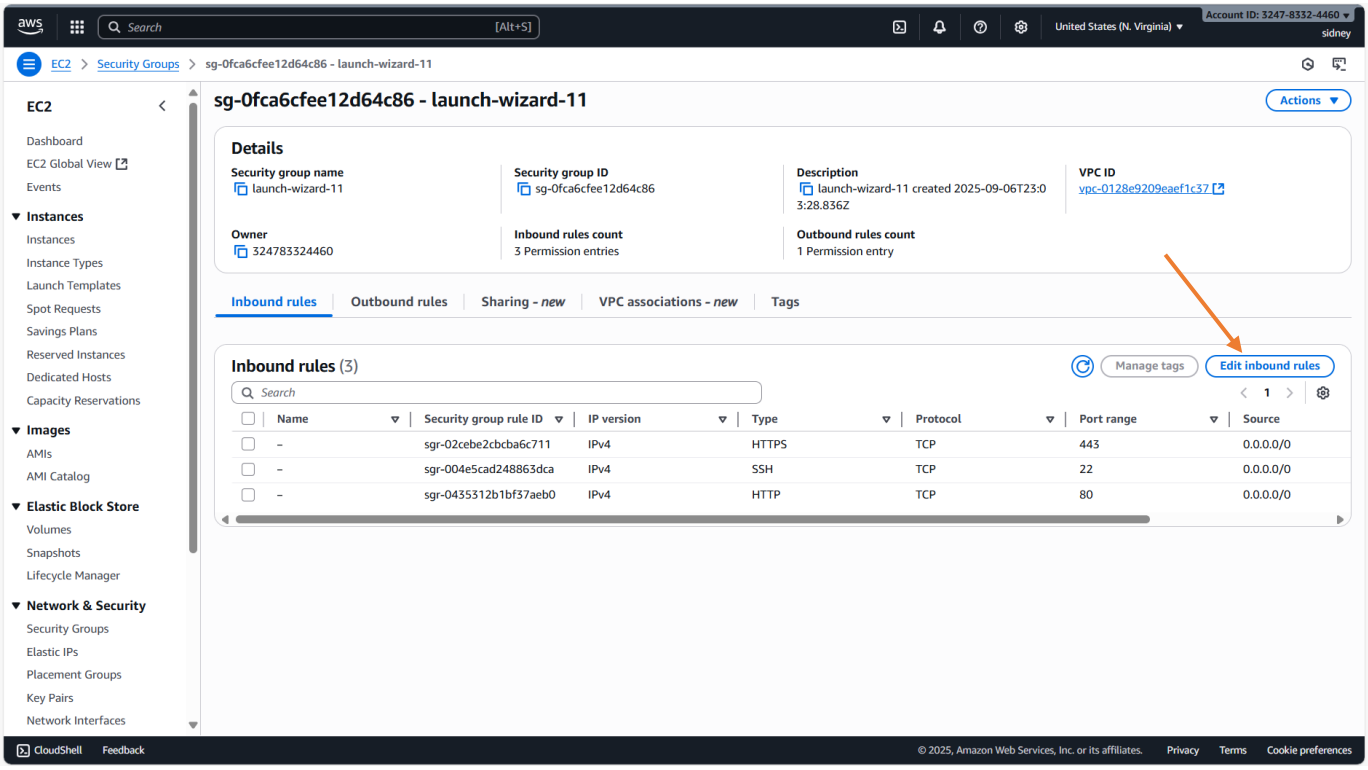
Select the EC2 instance



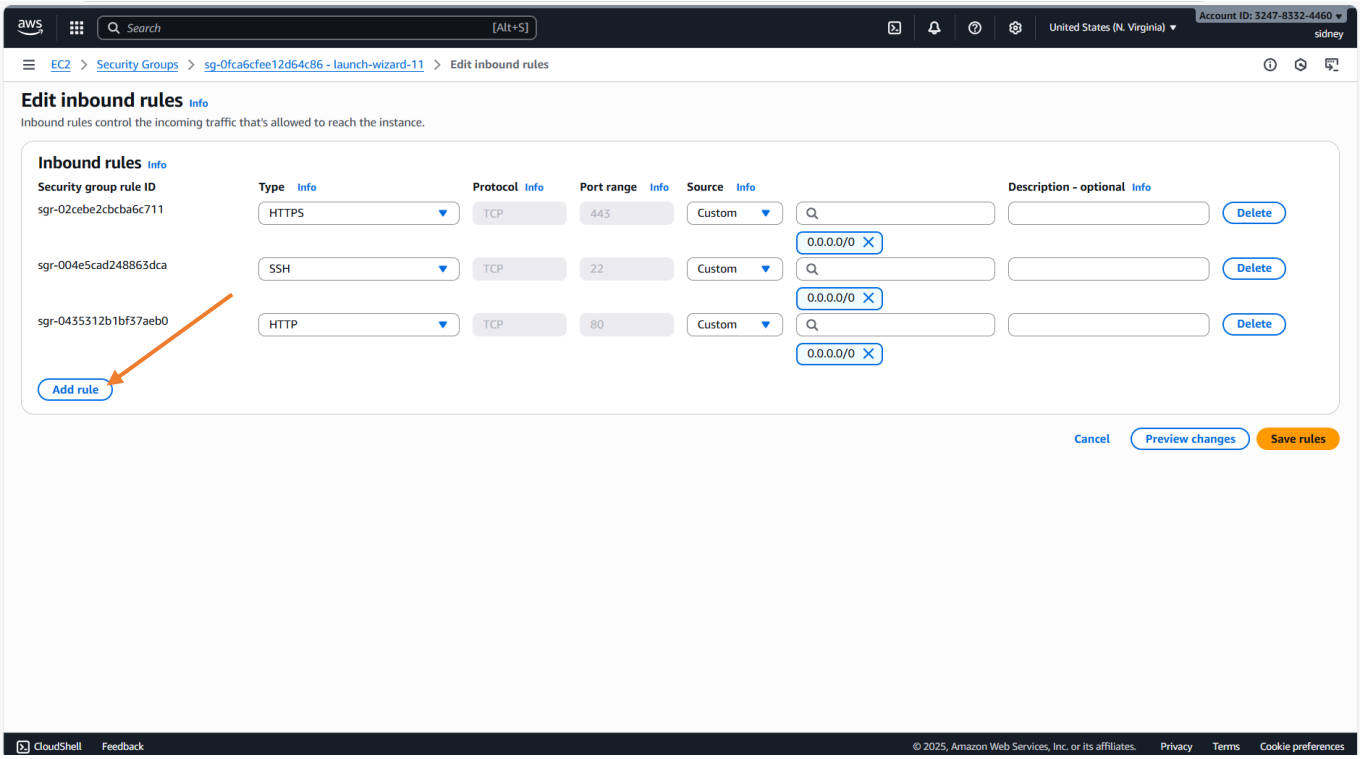
Click on the "**Security**" tab



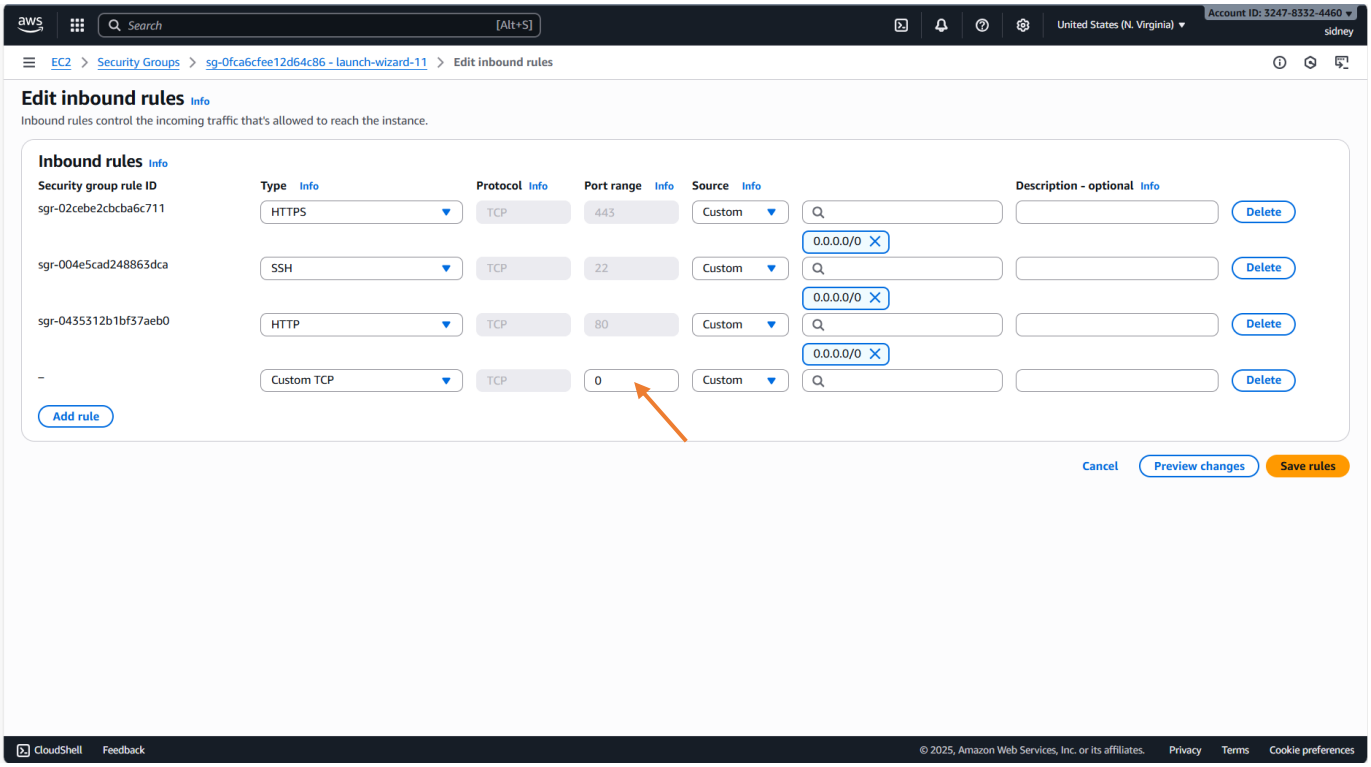
Click on the "**Security Group**" url



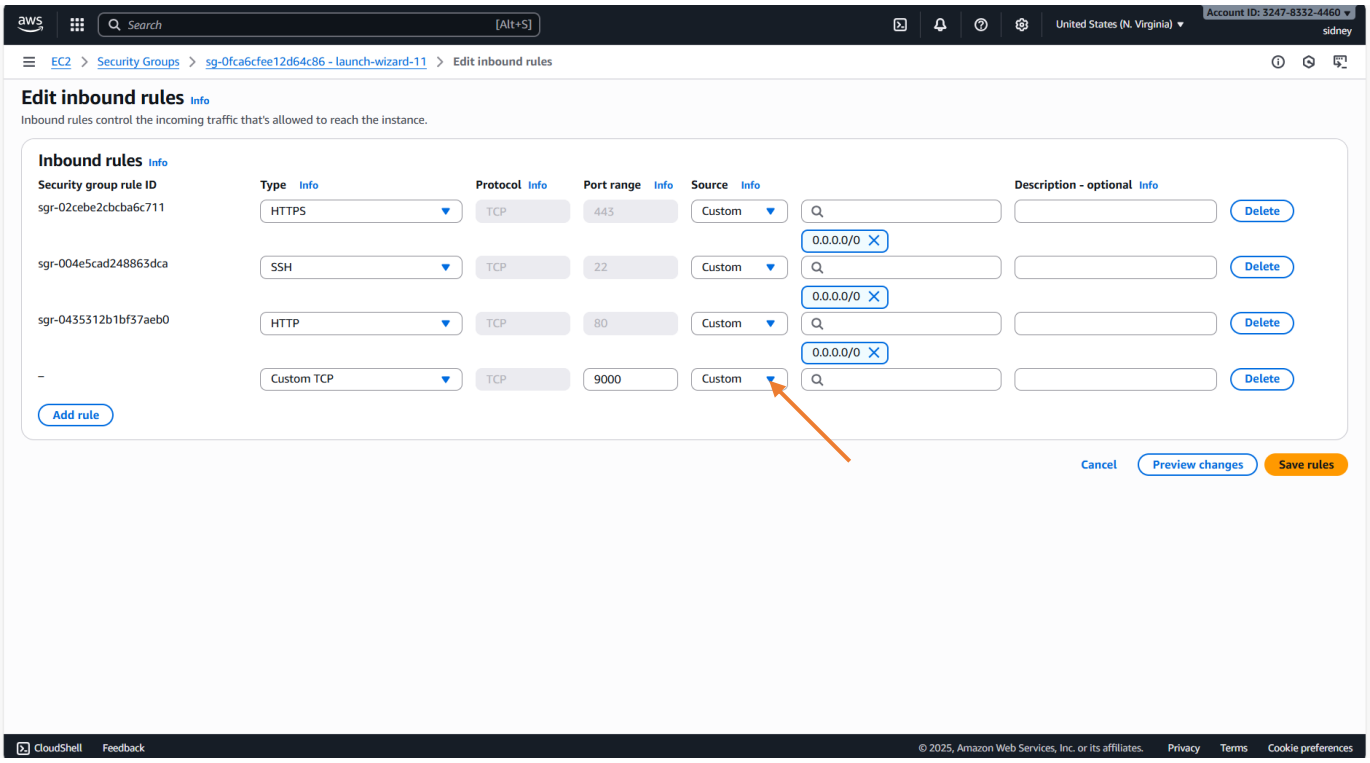
Click on "**Edit Inbound rules**"



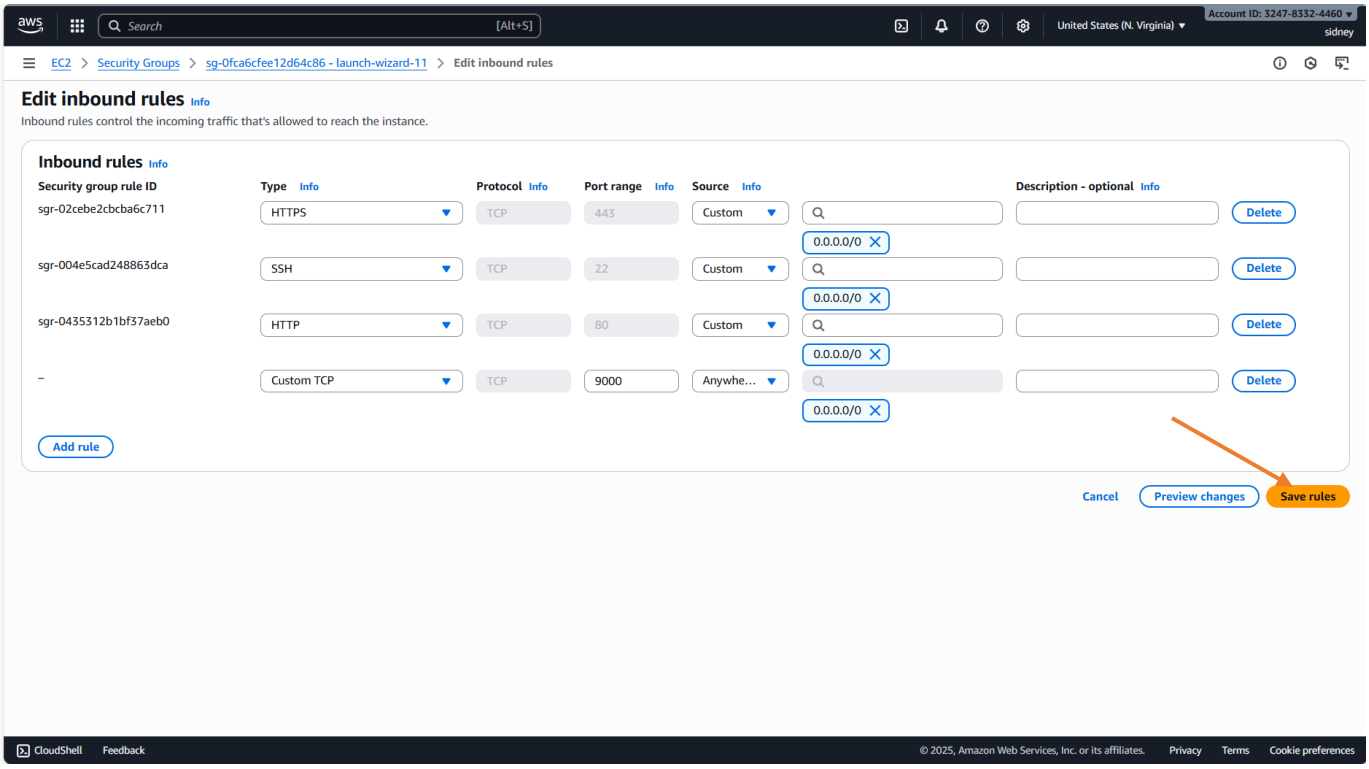
Click on "**Add Rule**"



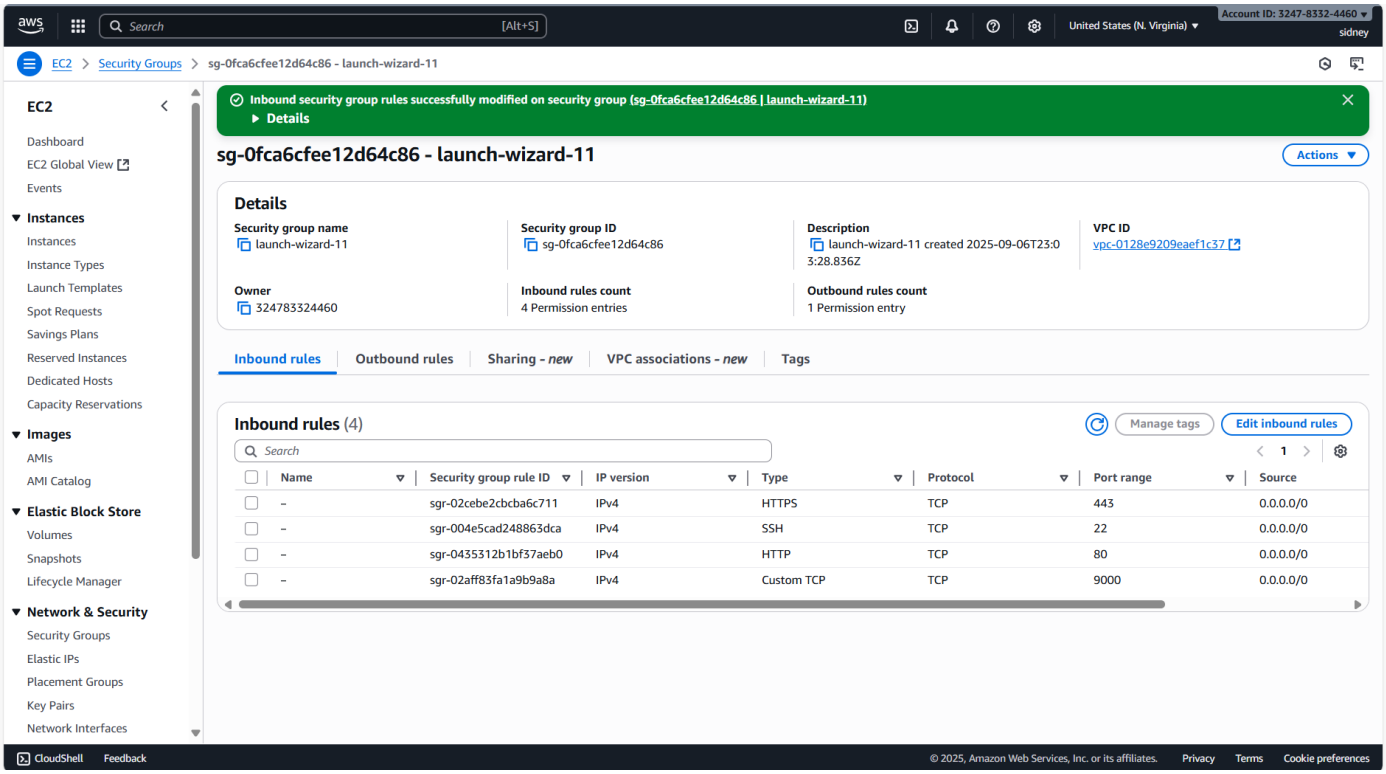
Click on the drop down on "**Port Range**" and enter the value "**9000**"



Click on the drop down on "**source**" and select "**Anywhere-IPv4**"



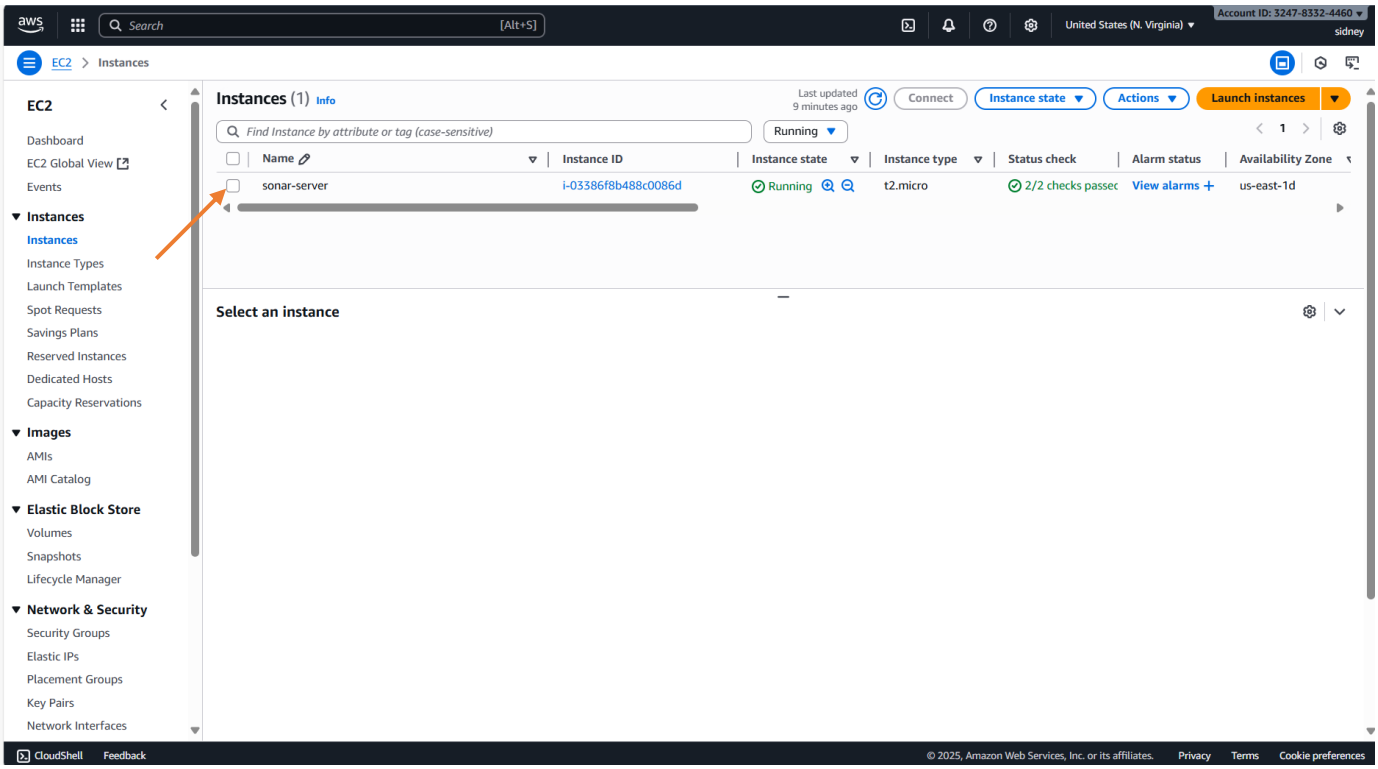
Click on "**Save rules**"



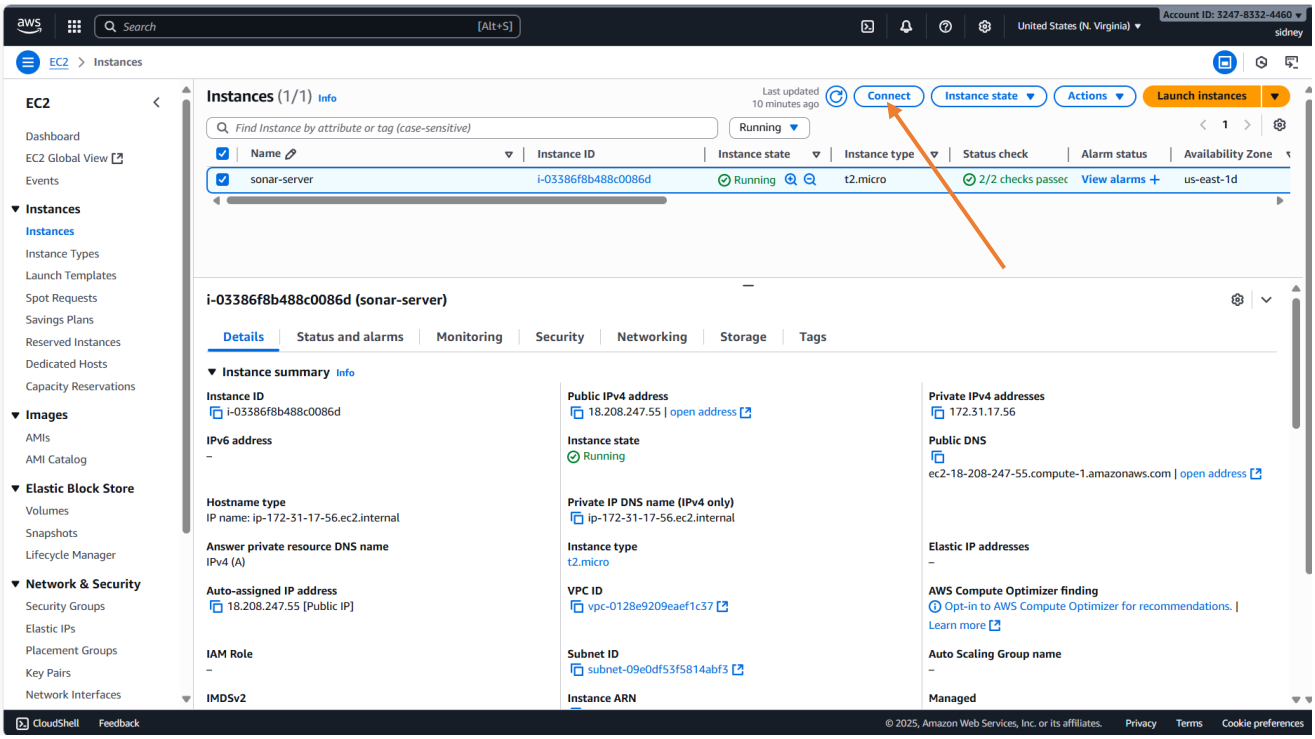
Port 9000 has been added.

PART 4: SSH Connect to the EC2 instance

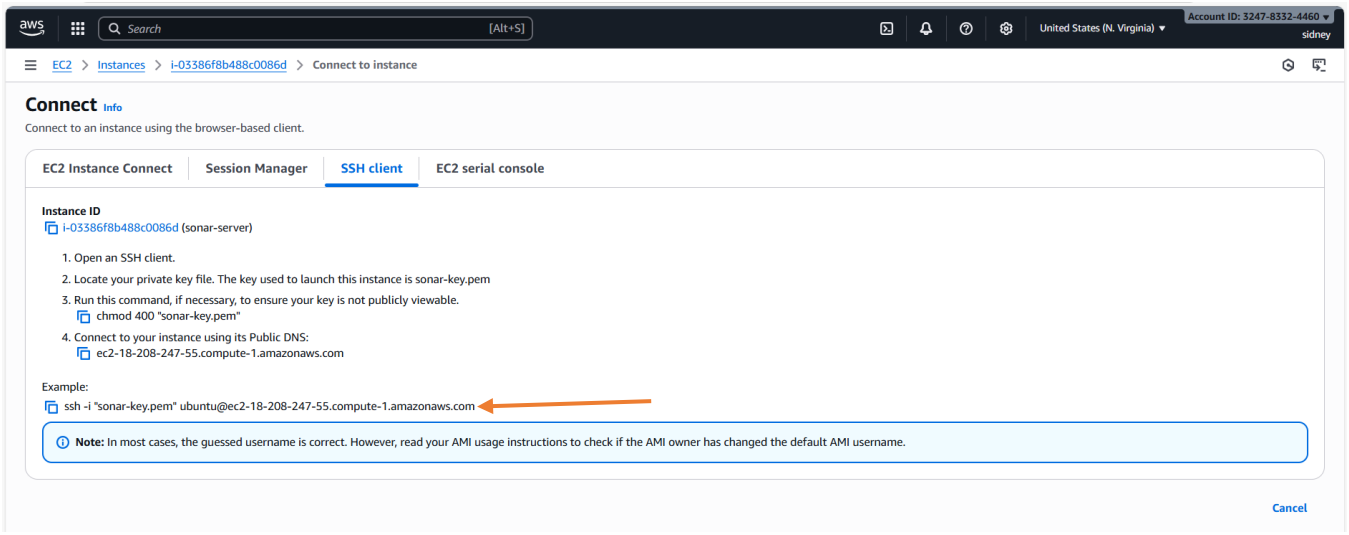
Now, let us connect to our EC2 instance through SSH



Select the instance

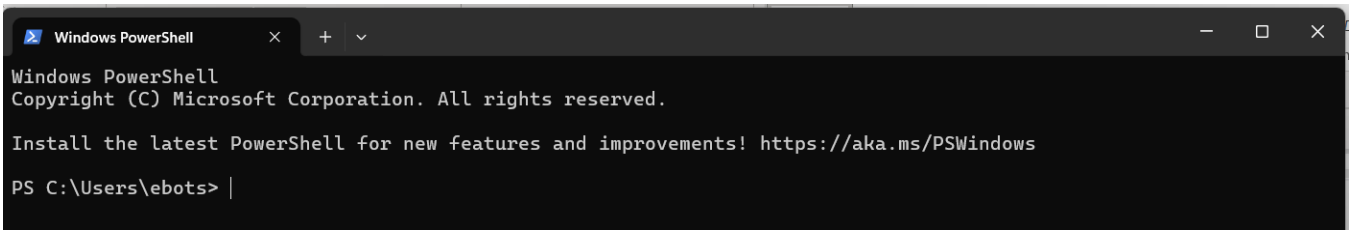


Click on "**Connect**"



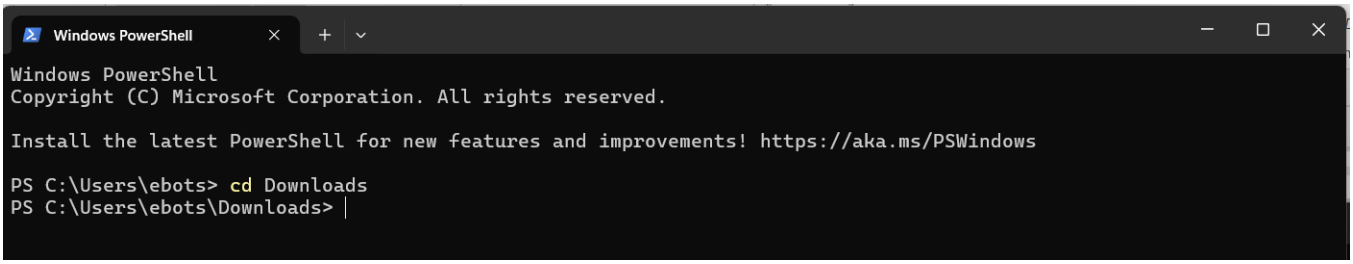
Copy this command and open PowerShell

ssh -i "sonar-key.pem" ubuntu@ec2-18-208-247-55.compute-1.amazonaws.com



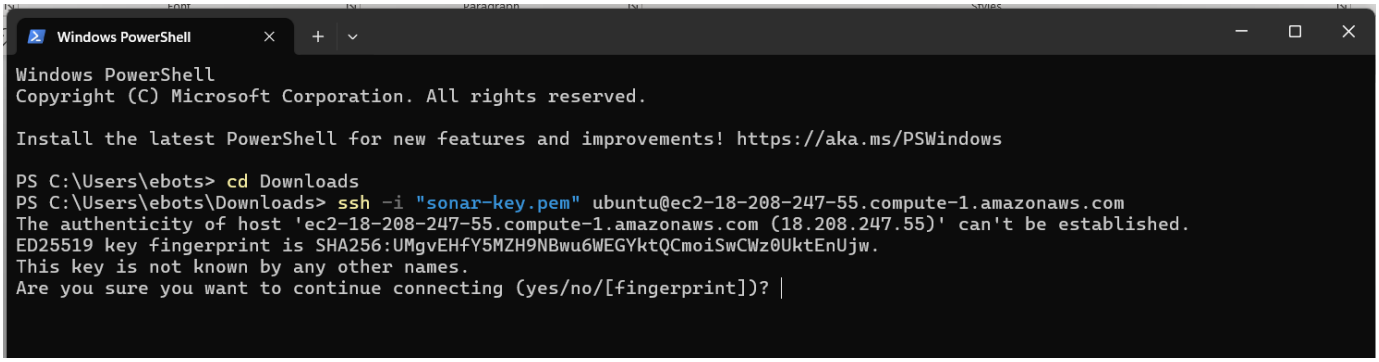
Navigate to the "**Downloads**" folder where our "**.pem**" file is stored by using the command:

cd Downloads

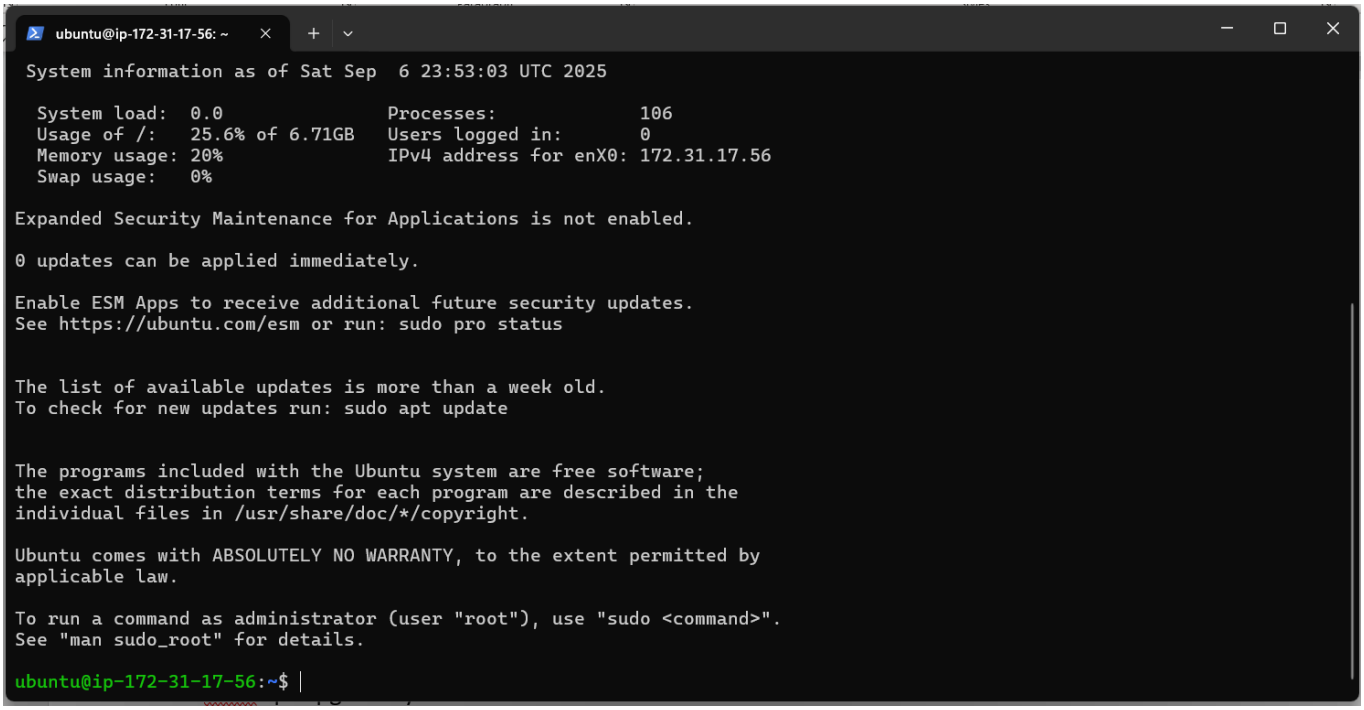


Then run the copied command:

ssh -i "sonar-key.pem" ubuntu@ec2-18-208-247-55.compute-1.amazonaws.com



Type "**yes**" and press ENTER



We have successfully SSH connected to our EC2 instance

Clear the terminal using the command:

clear

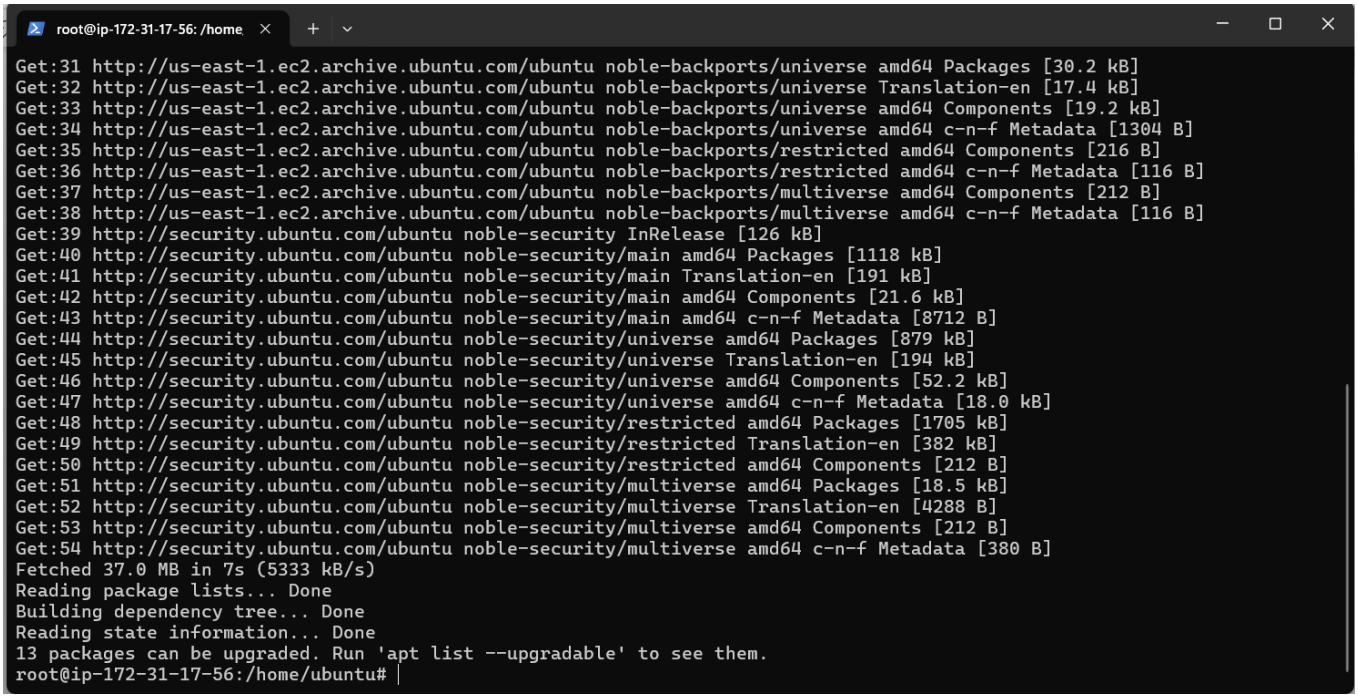
Grand Root user access using the command:

sudo su

STEP 2: Install Java JDK 17

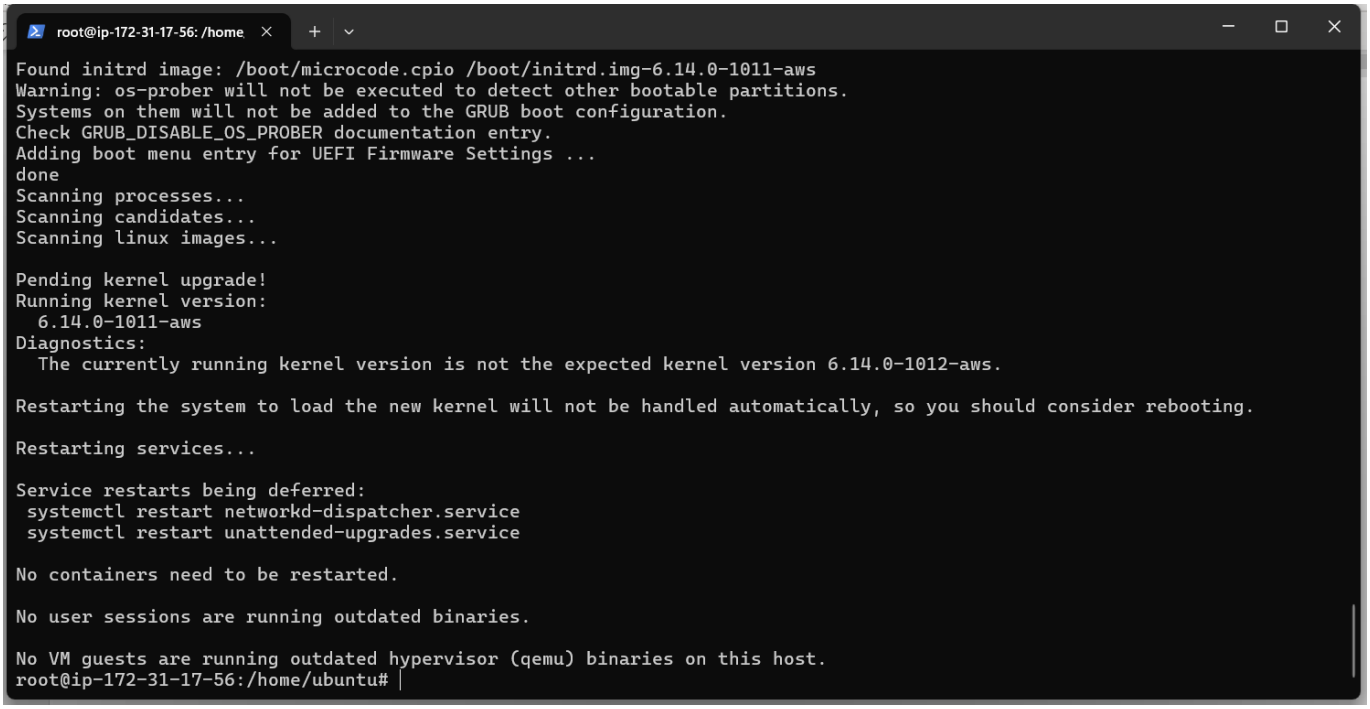
Let's rejuvenate our Ubuntu server

sudo apt update



Update the server

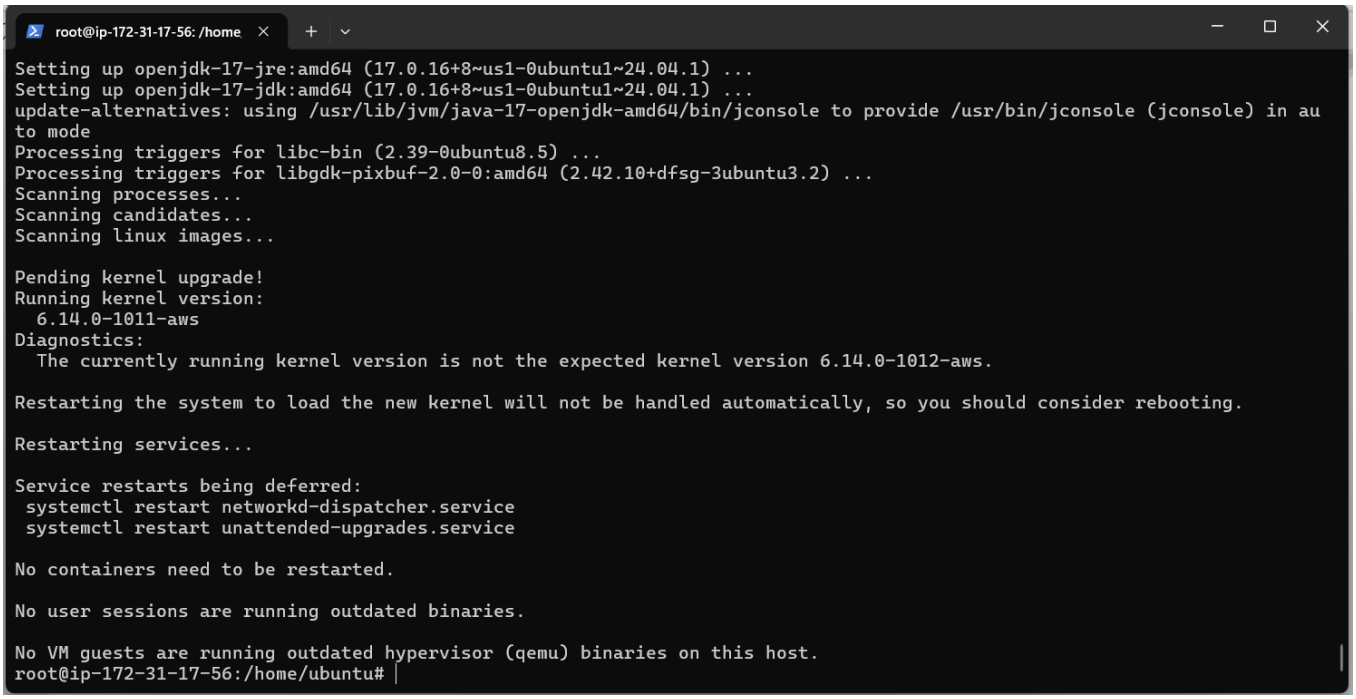
sudo apt upgrade -y



Install OpenJDK 17

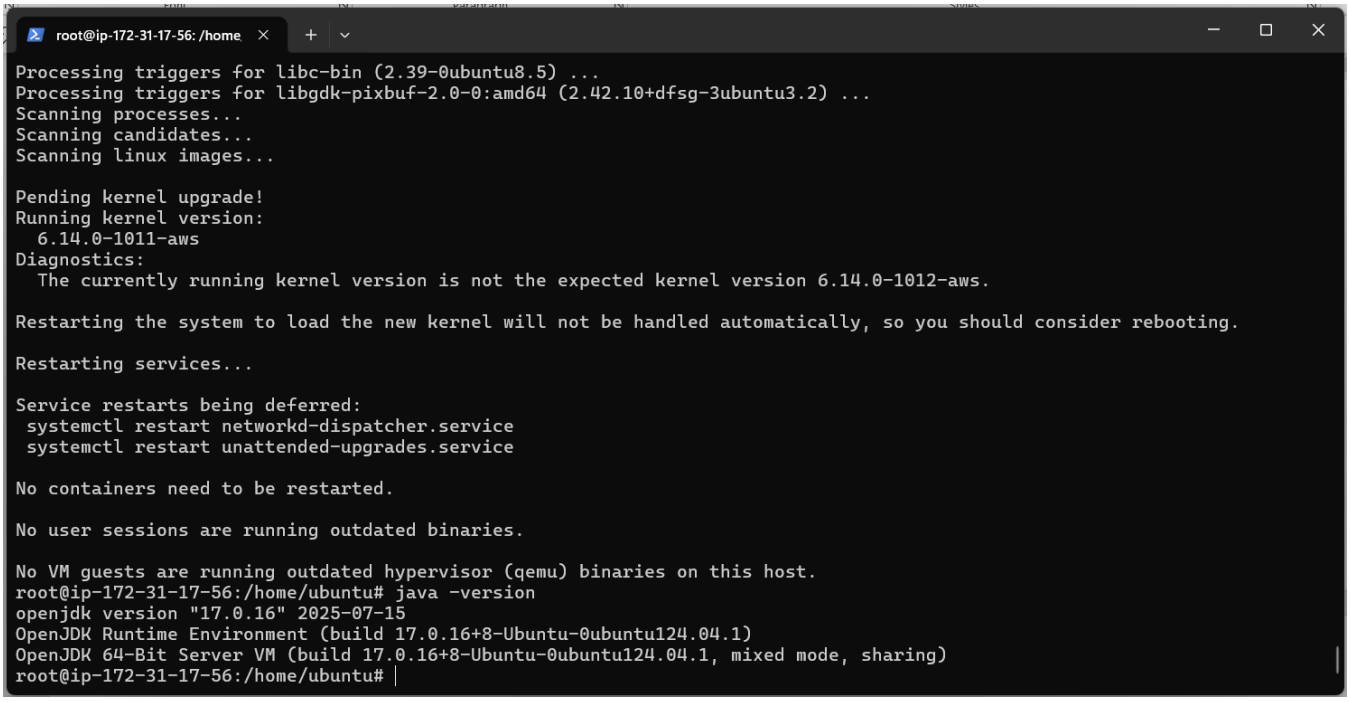
Install OpenJDK 17 (needed for the latest version of SonarQube (version 10.0).

```
sudo apt install -y openjdk-17-jdk
```



Let's check the installed version of Java. VALIDATION IS IMPORTANT.

java -version



STEP 3: Install and Configure PostgreSQL

Add the PostgreSQL repository

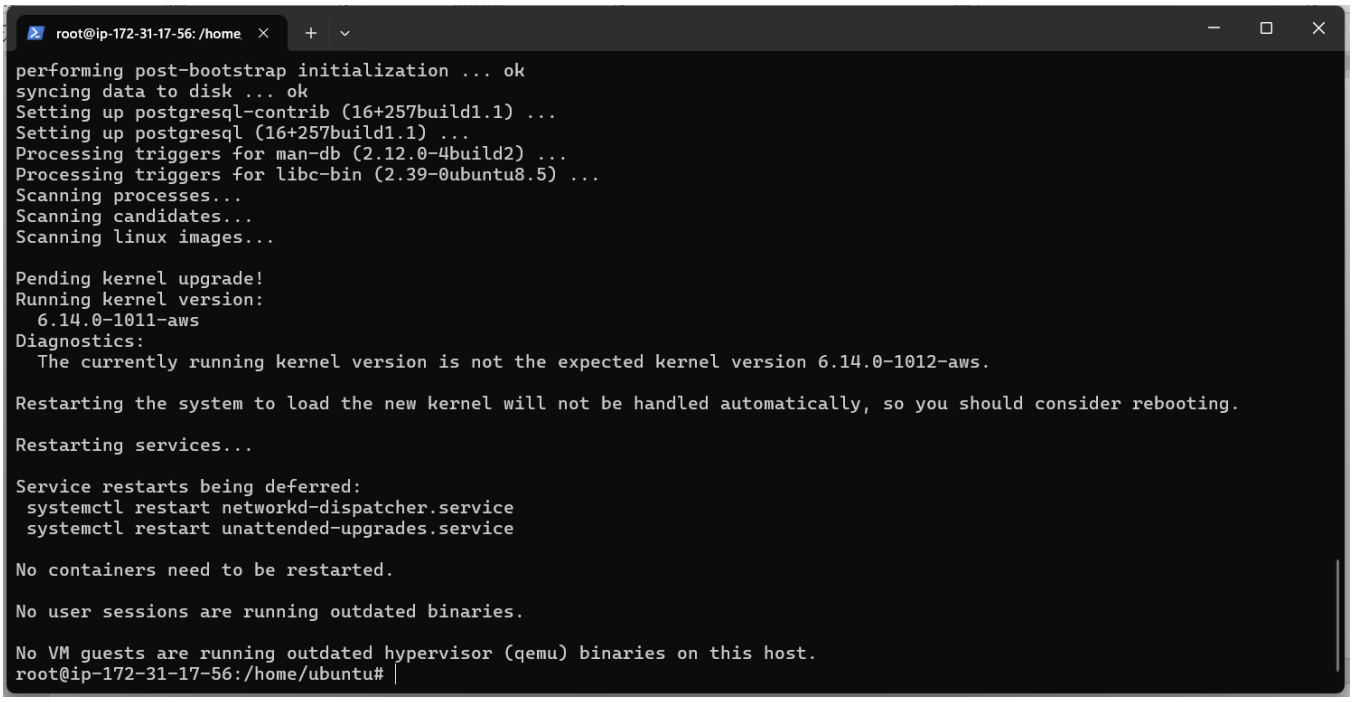
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-pgdg main" /etc/apt/sources.list.d/pgdg.list'

Add the PostgreSQL signing key

wget -q https://www.postgresql.org/media/keys/ACCC4CF8.asc -O - | sudo apt-key add -

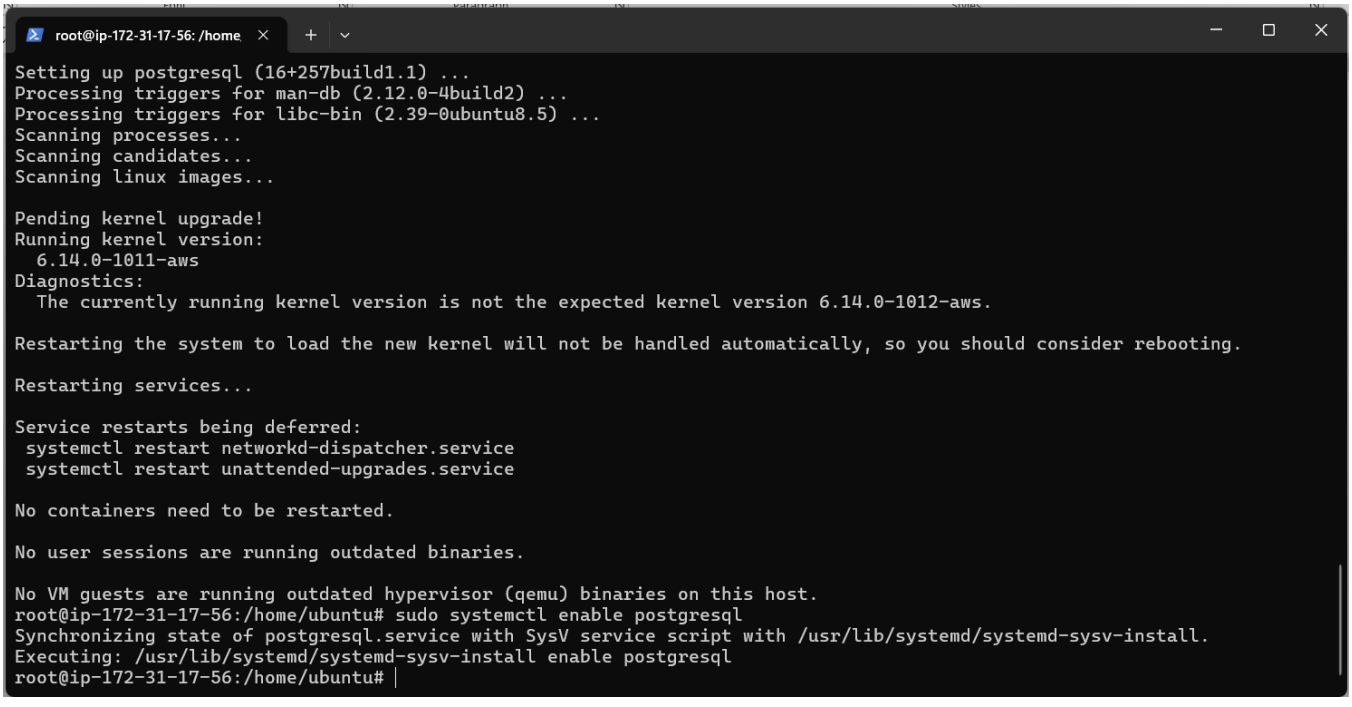
Install PostgreSQL

sudo apt install postgresql postgresql-contrib -y



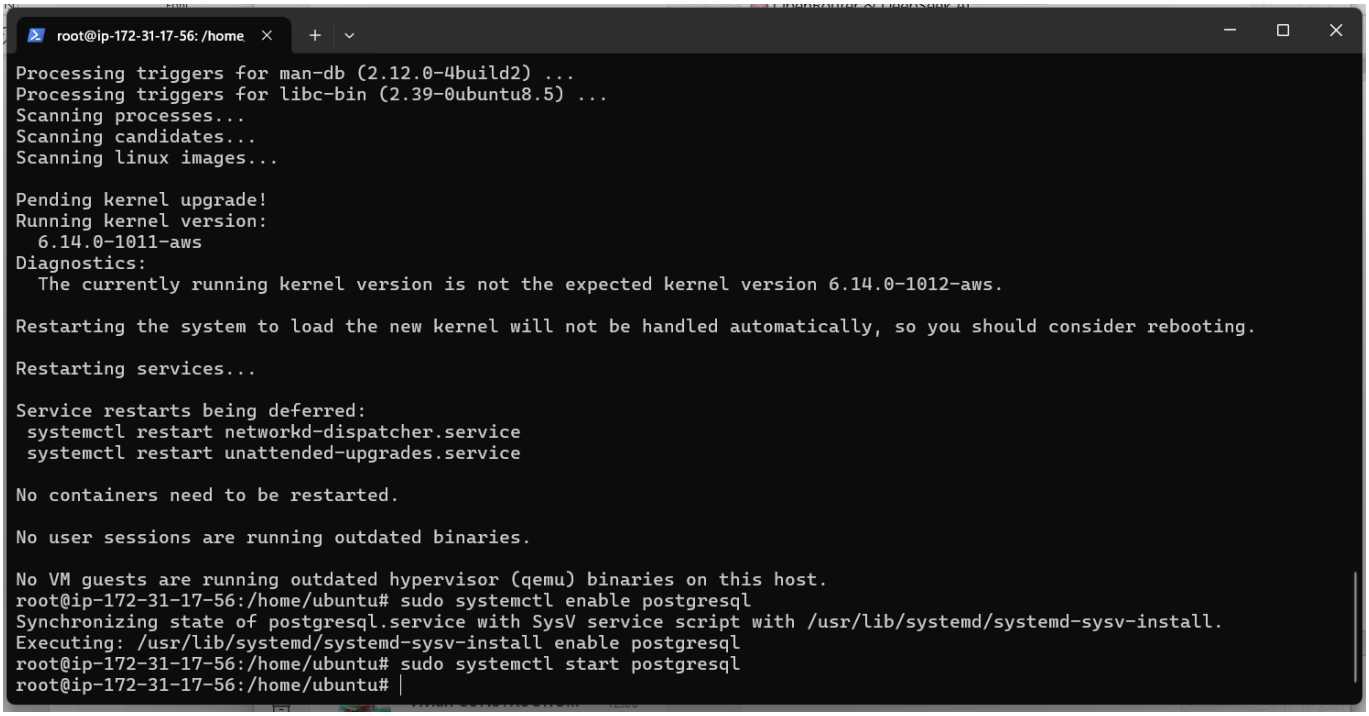
Enable the database server to start automatically on reboot

sudo systemctl enable postgresql



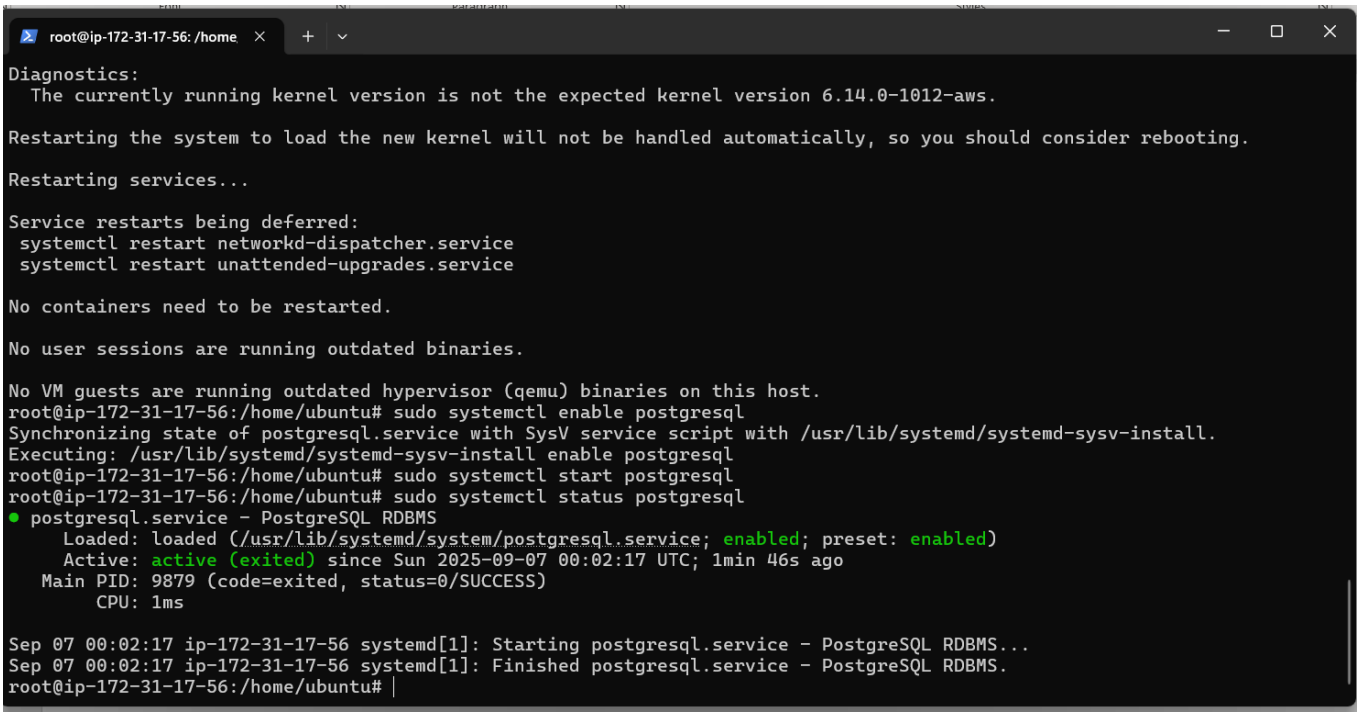
Start the database server

sudo systemctl start postgresql



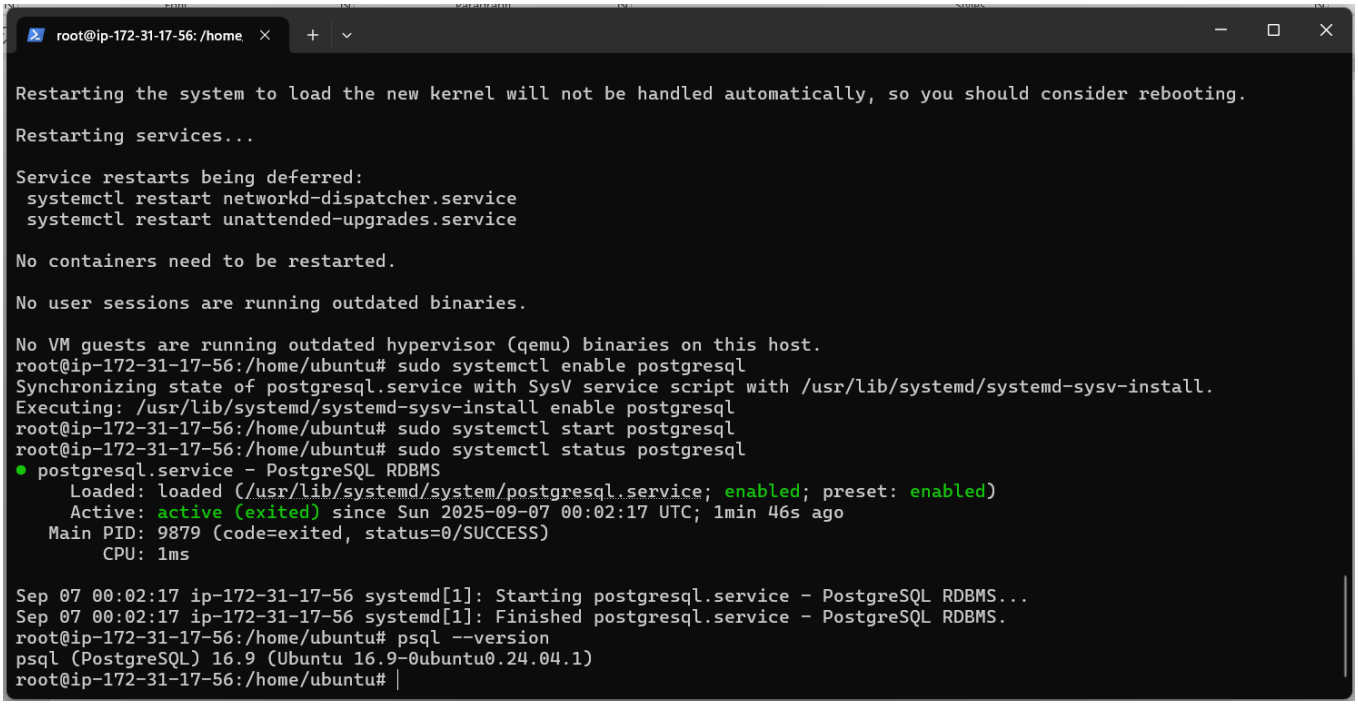
Check the status of the database server

sudo systemctl status postgresql



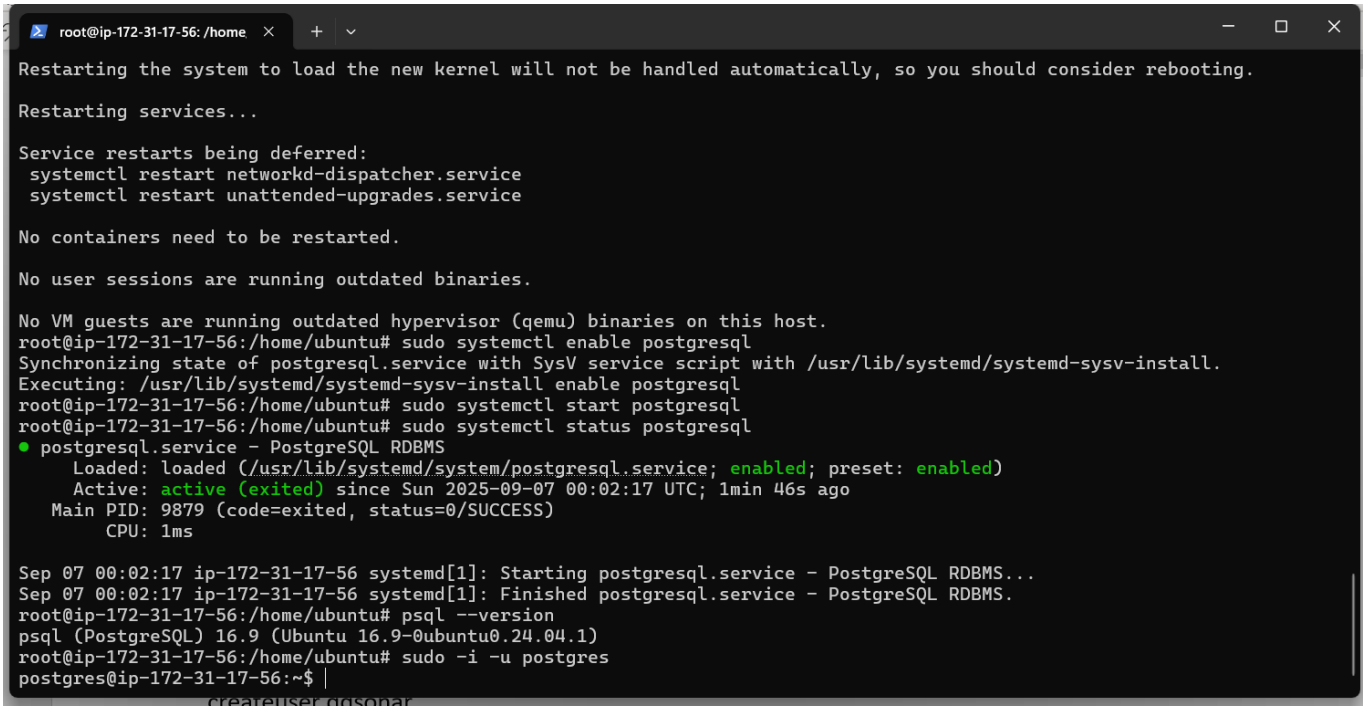
Let's check the installed version of the install Postgres DB. VALIDATION IS IMPORTANT

psql --version



Switch to the Postgres user

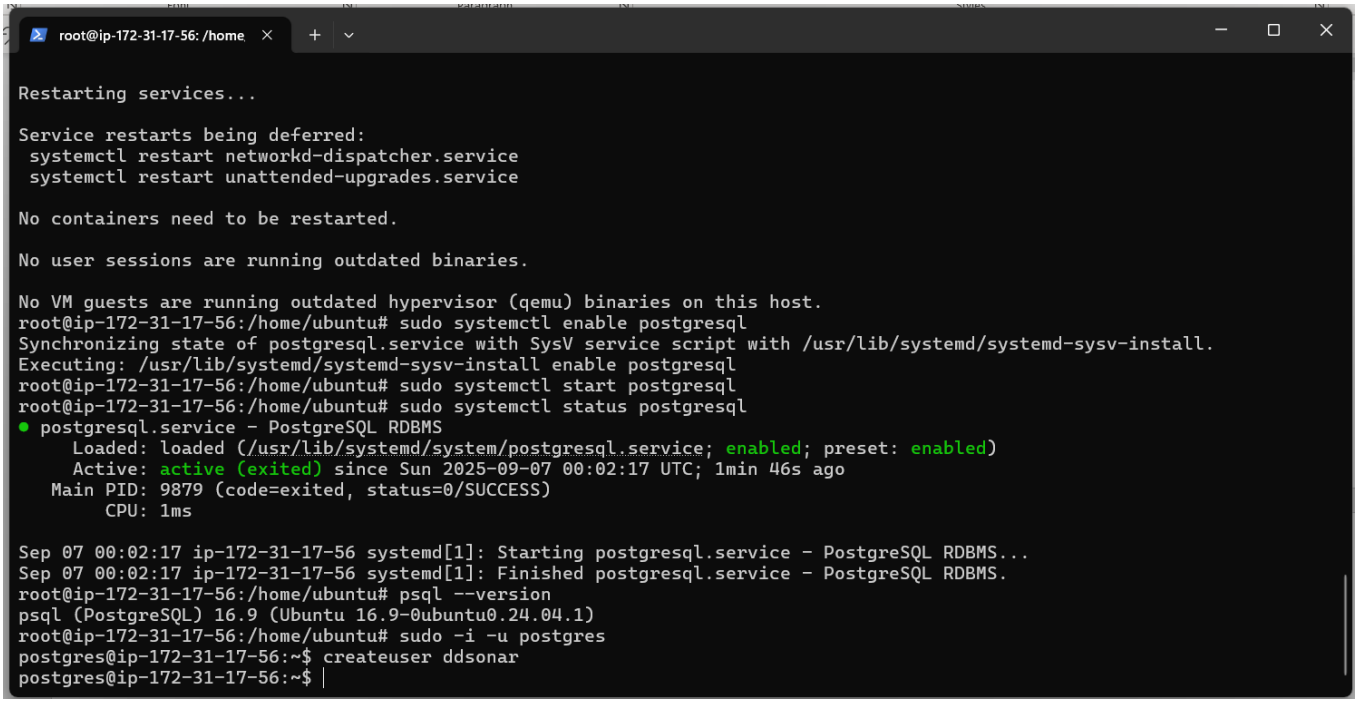
sudo -i -u postgres



Create a database user named ddsonar

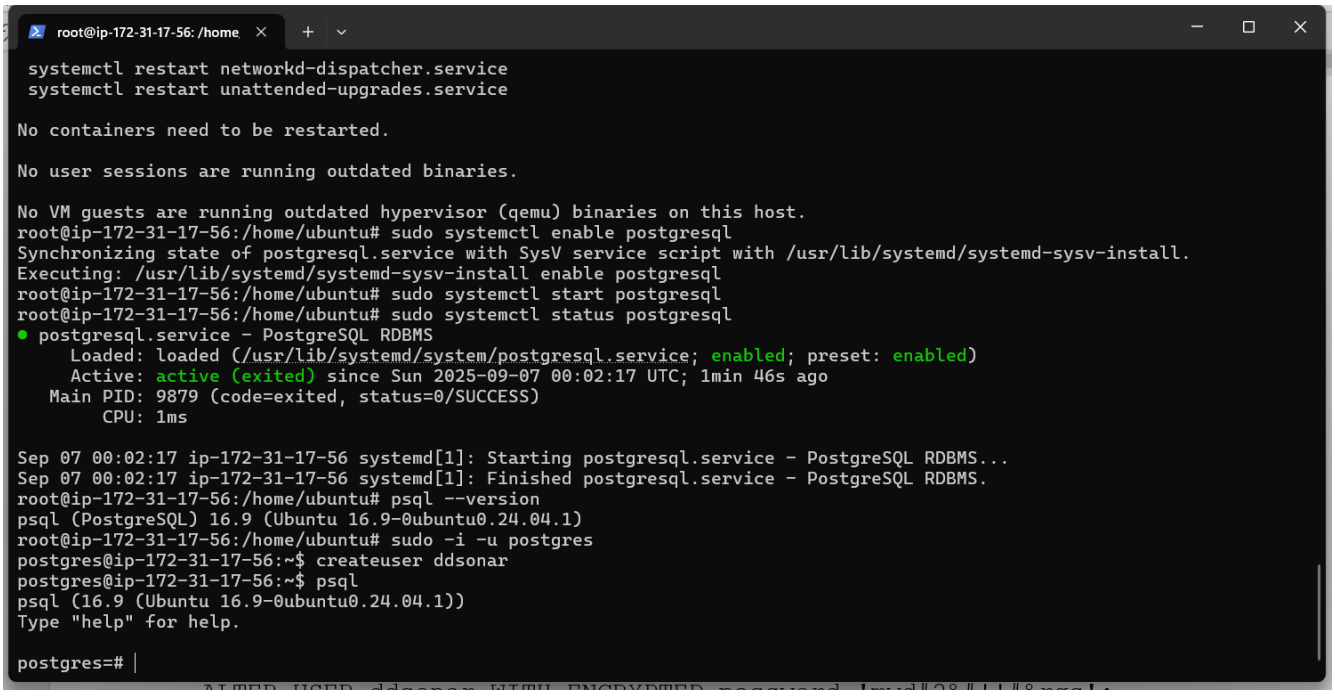
Note: You can provide the name of your own but make a note of it, as we will be needing this name in further steps.

createuser ddsonar



Log in to PostgreSQL

psql

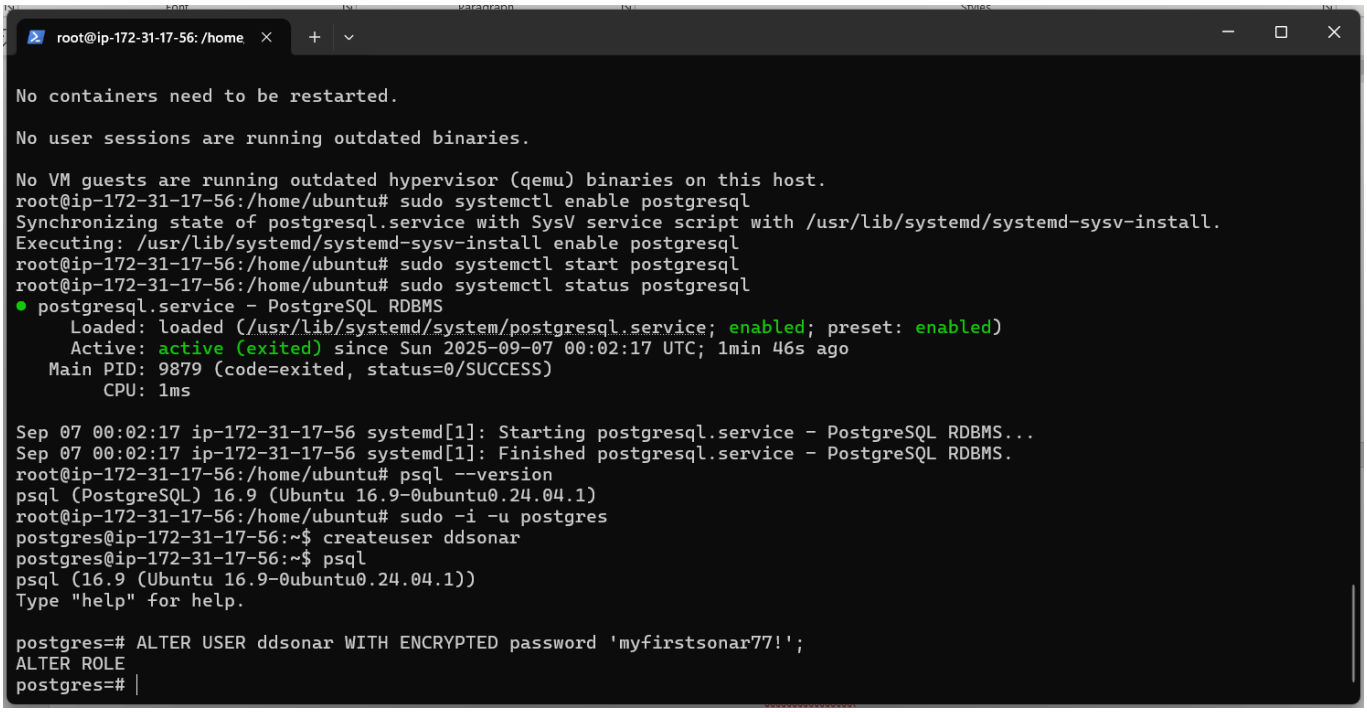


Set a password for the ddsonar user. Use a strong password in place of my_strong_password.

```
ALTER USER [Created_user_name] WITH ENCRYPTED password 'my_strong_password';
```

For example (In my case it will be):

```
ALTER USER ddsonar WITH ENCRYPTED password 'myfirstsonar77!';
```

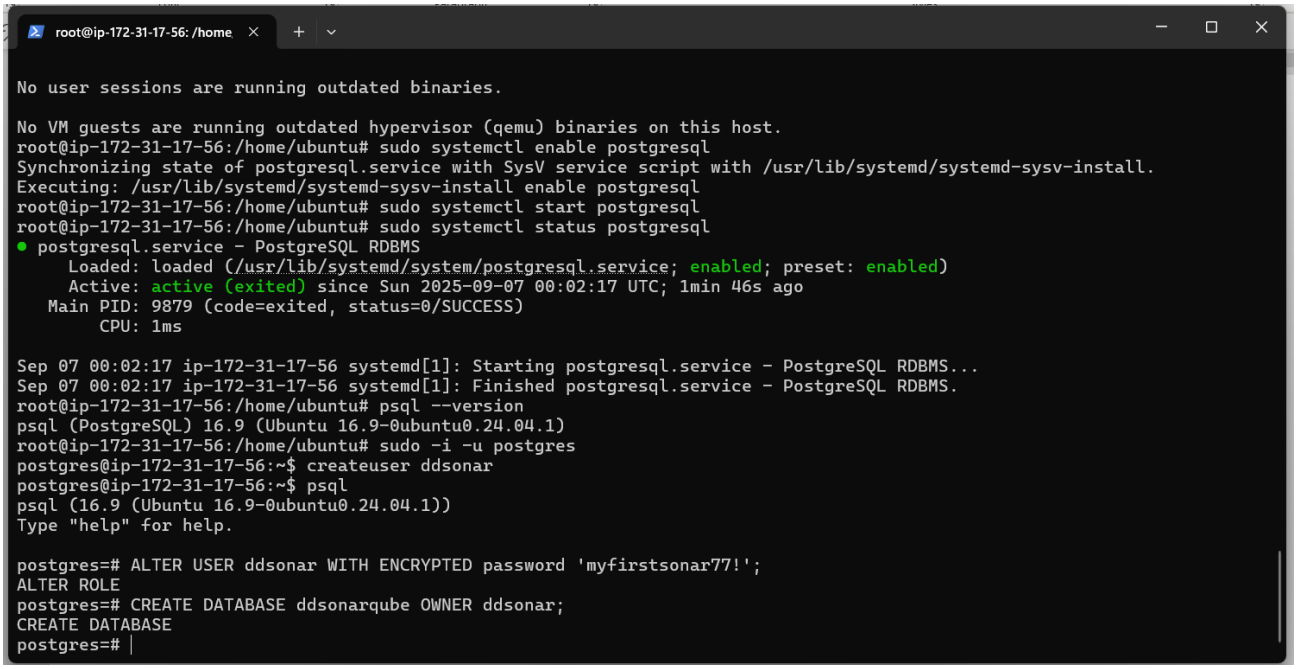


Create a SonarQube database and set the owner to ddsonar.

```
CREATE DATABASE [database_name] OWNER [Created_user_name];
```

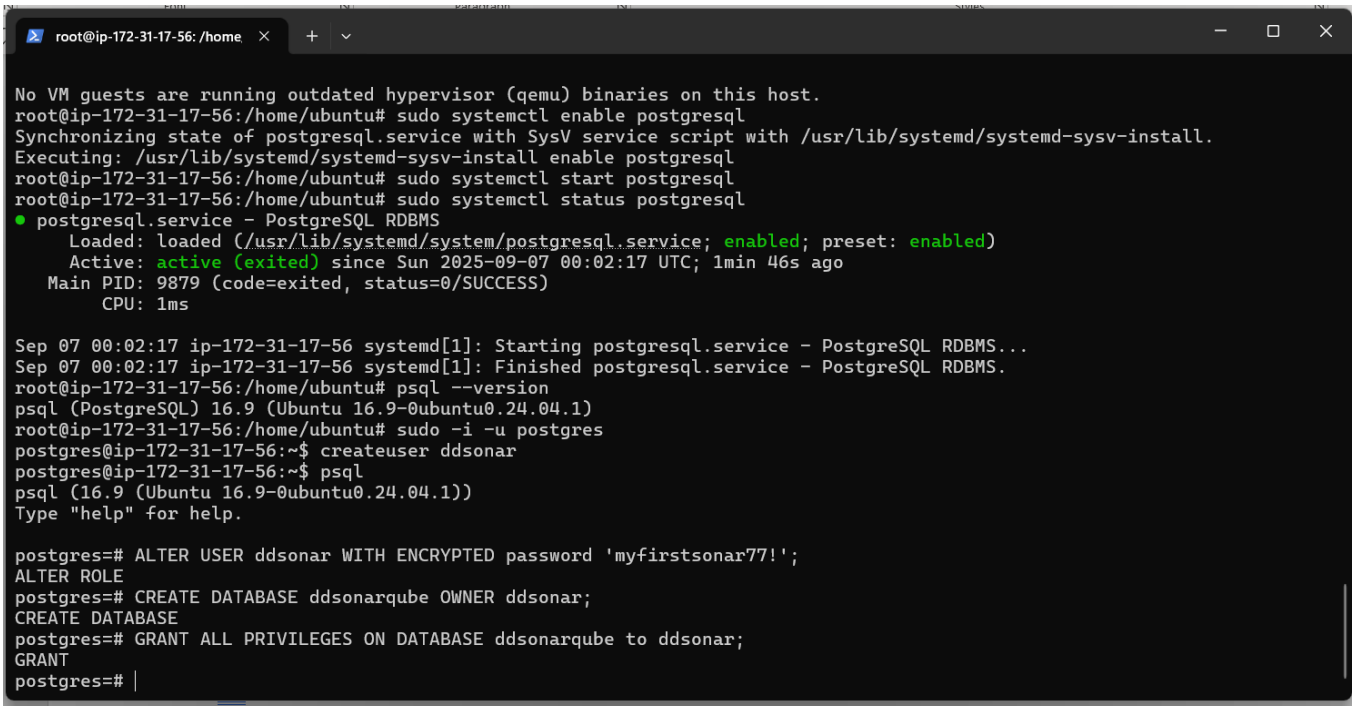
For example (In our case it will be) — feel free to give an awesome database name as per your requirement:

```
CREATE DATABASE ddsonarqube OWNER ddsonar;
```



Grant all the privileges on the ddsonarqube database to the ddsonar user.

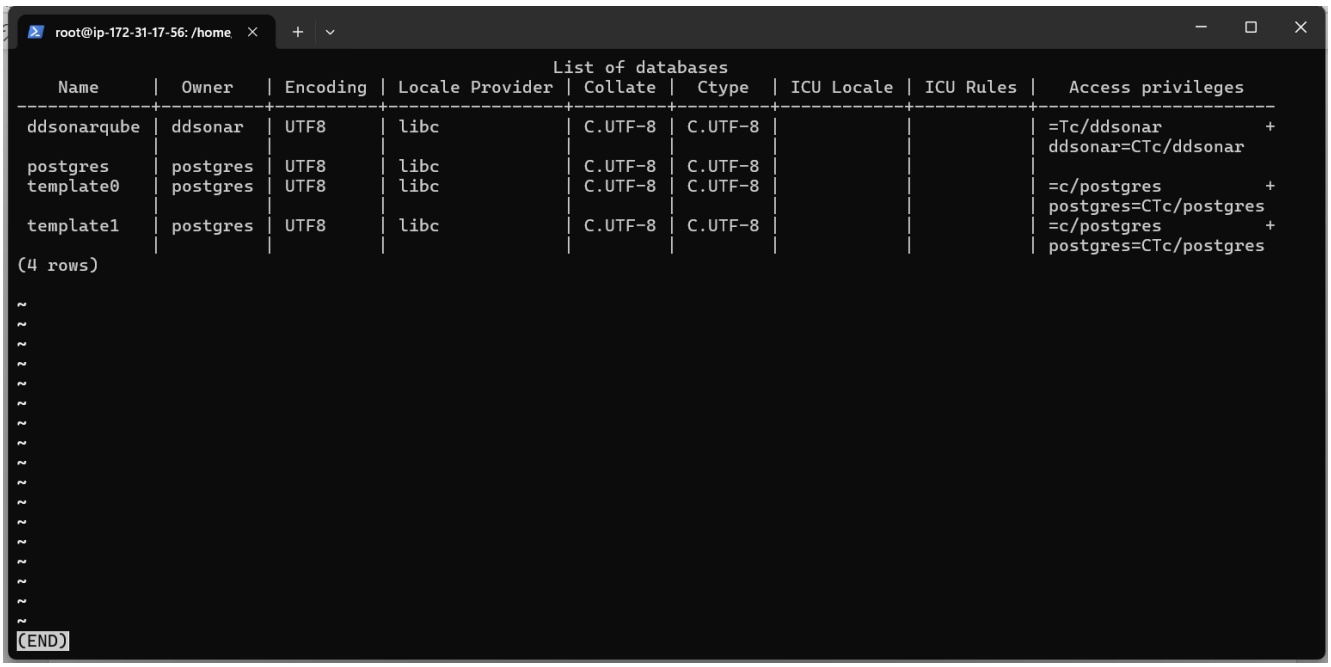
```
GRANT ALL PRIVILEGES ON DATABASE ddsonarqube to ddsonar;
```



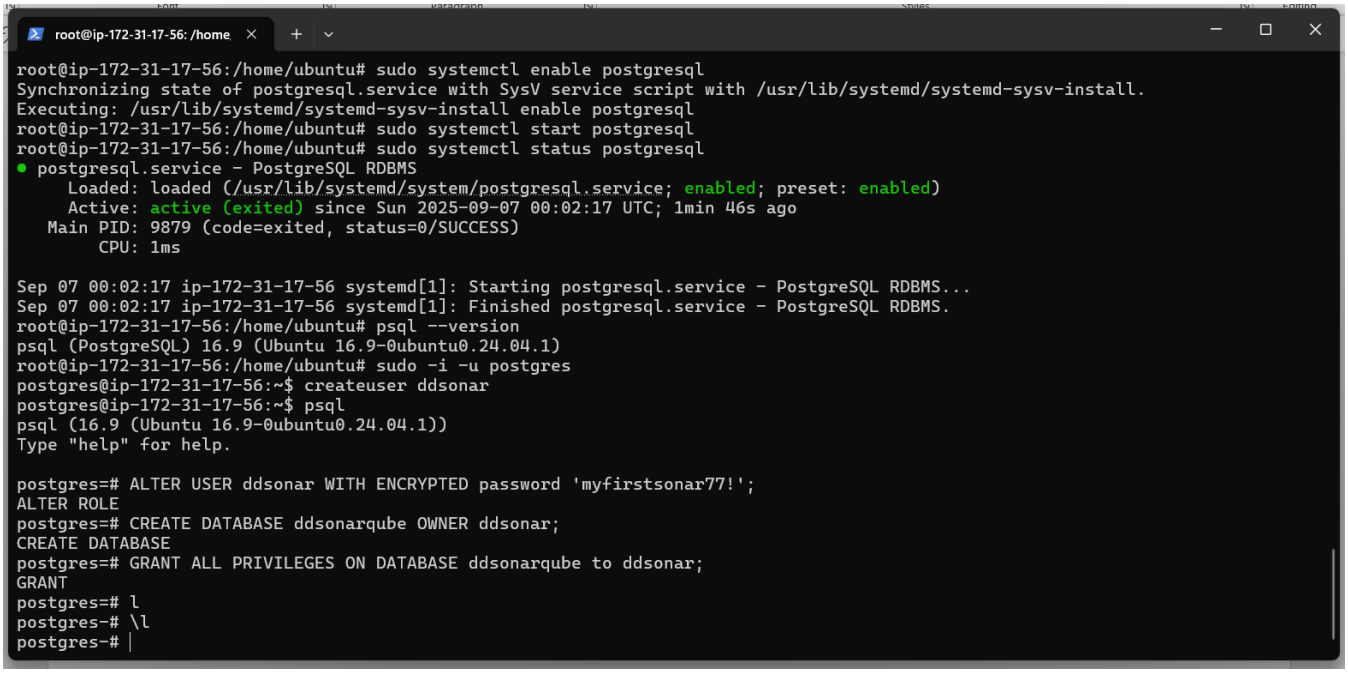
Let's check the created user and the database.

a) To check the created database using the command:

```
\l
```

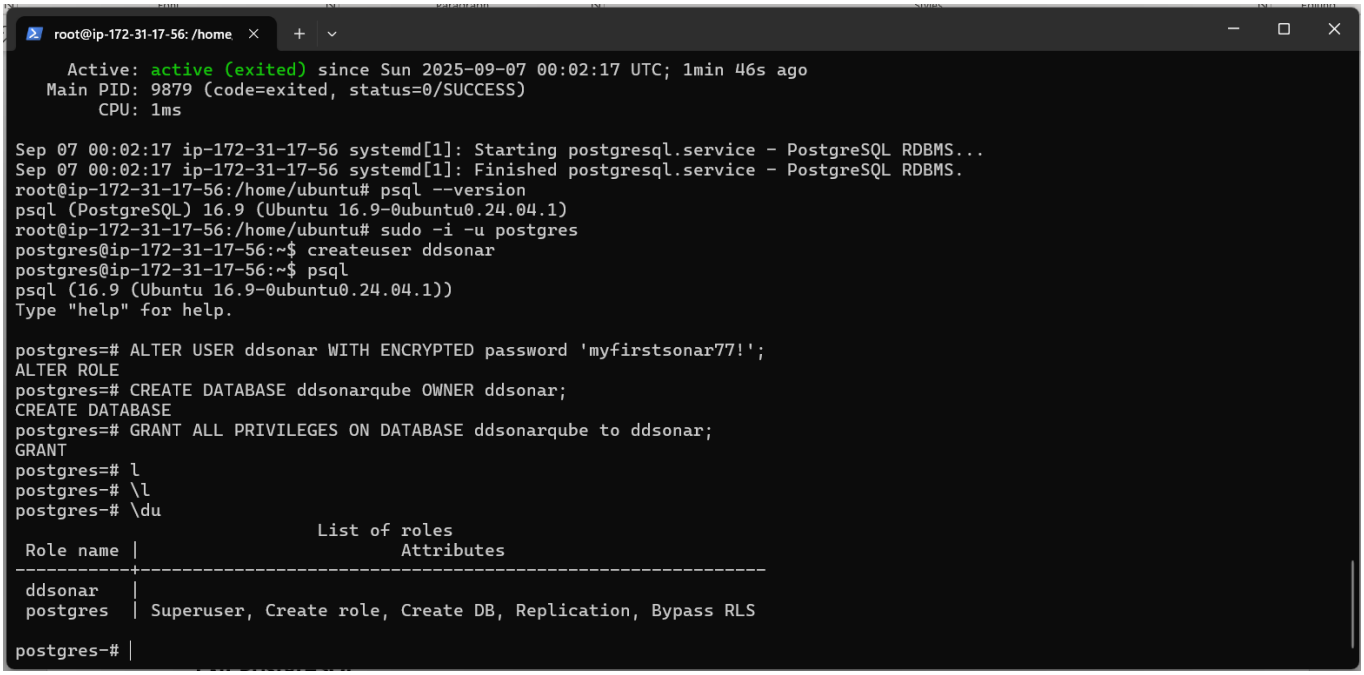


Press q



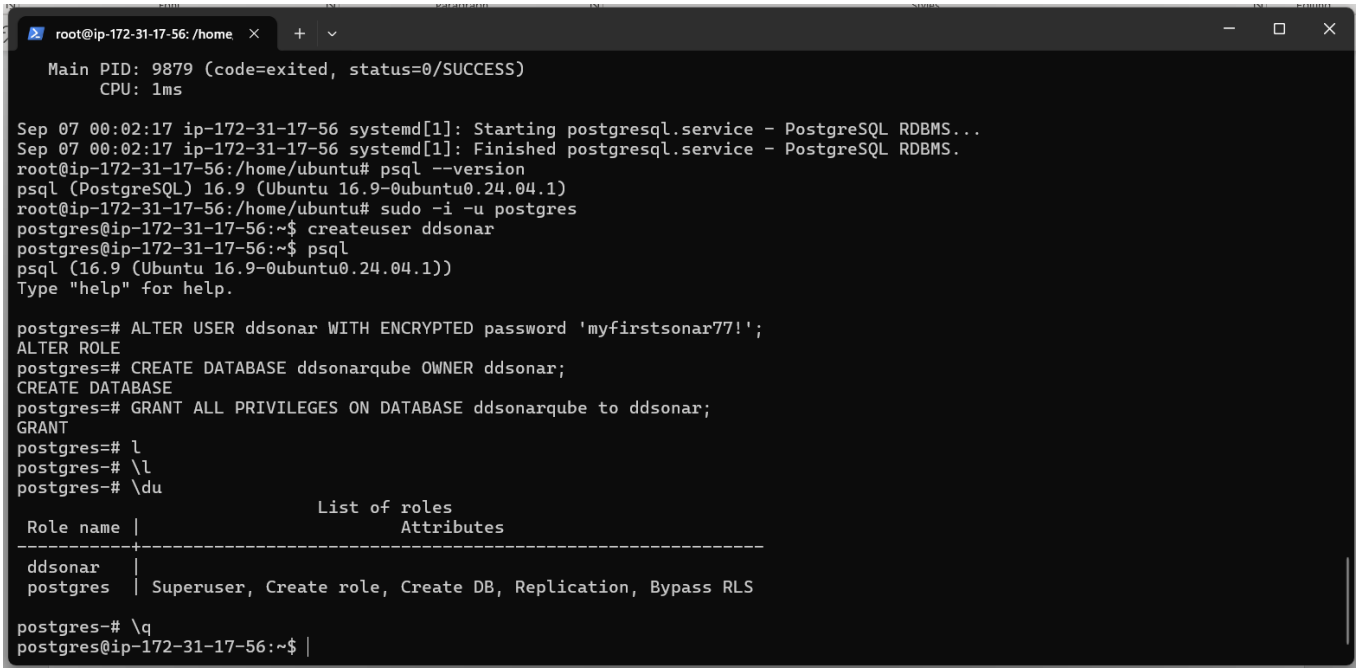
b) To check the created database user

```
\du
```



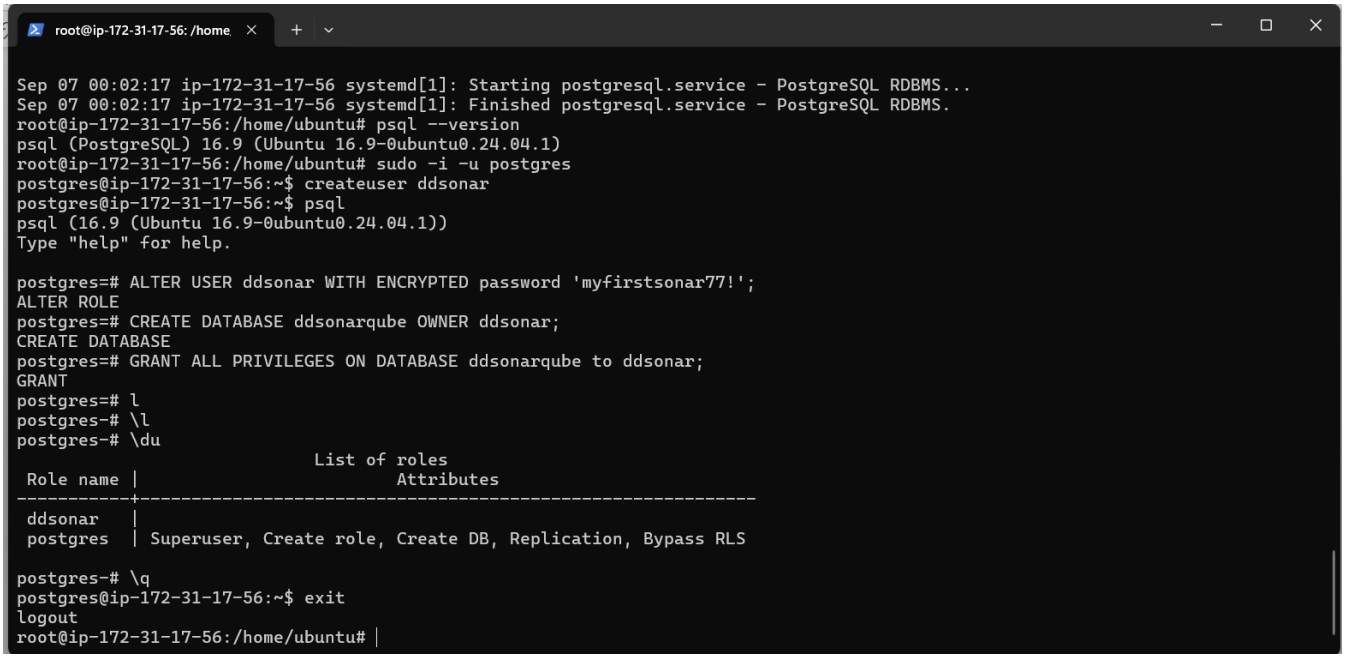
Exit PostgreSQL.

```
\q
```



xvi) Return to your non-root sudo user account.

`exit`

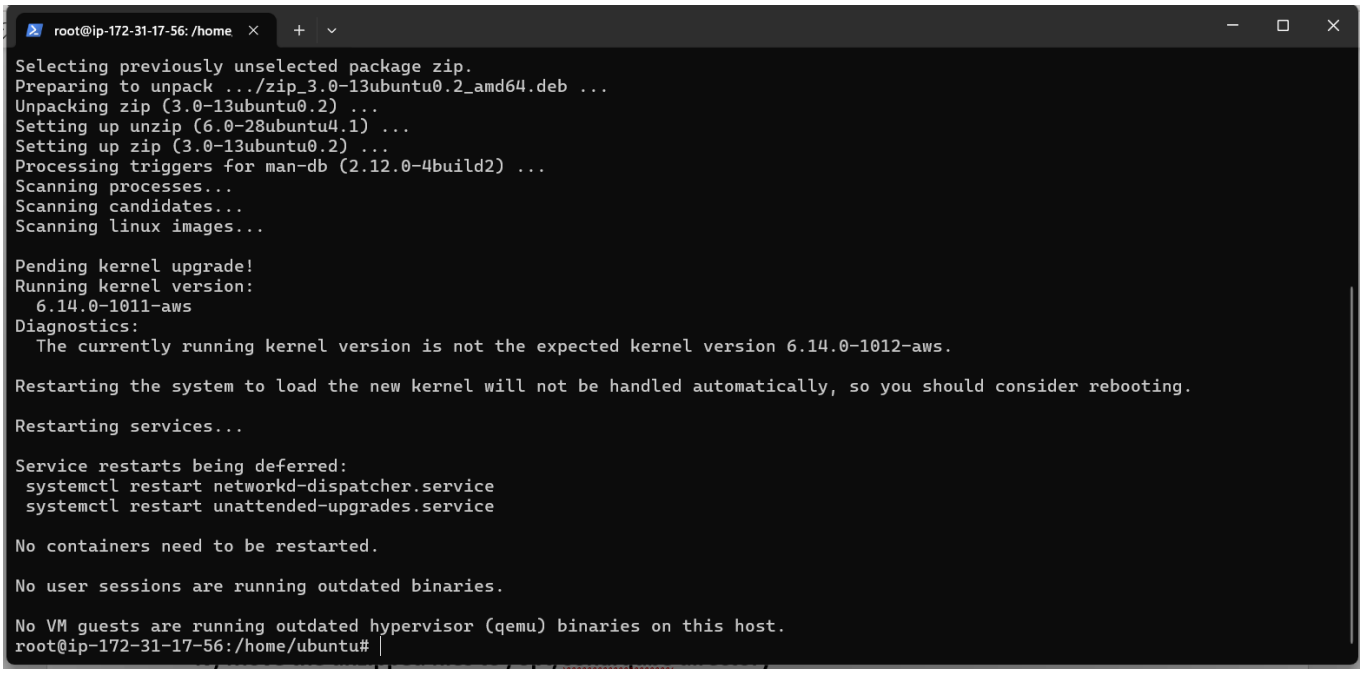


STEP 4: Install SonarQube

Part 1: Download and Install SonarQube

Install the zip utility, which is needed to unzip the SonarQube files.

```
sudo apt install zip -y
```



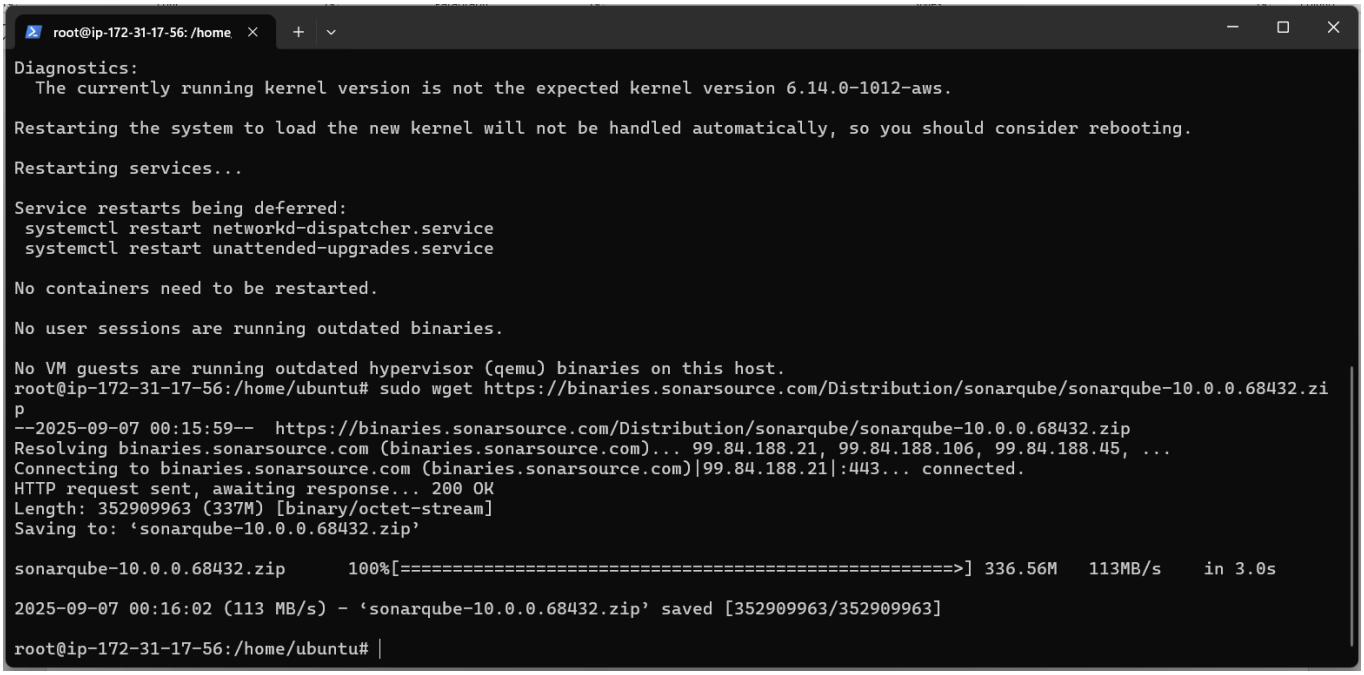
Locate the latest download URL from the SonarQube official download page.

Download the SonarQube distribution files. (you can download the latest SonarQube distribution using the following link)

https://www.sonarsource.com/products/sonarqube/downloads/

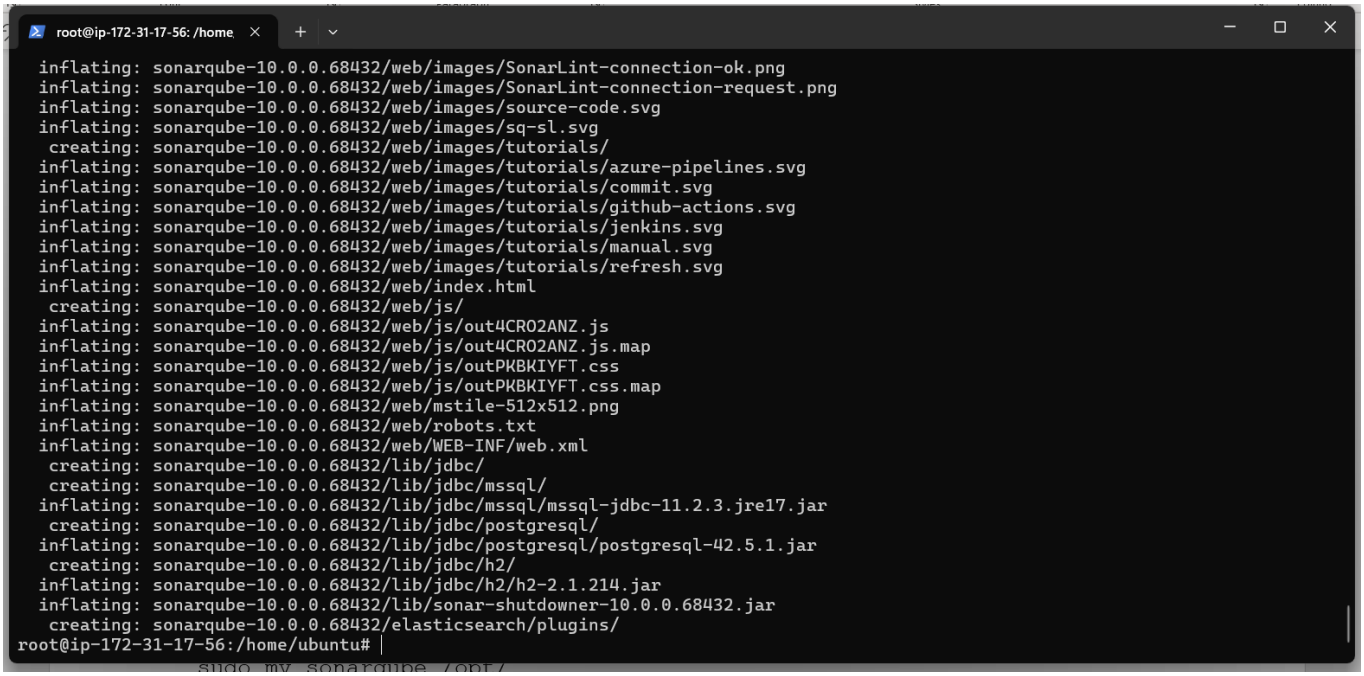
Here we are installing the latest version of SonarQube 10.0 community edition (free one)

```
sudo wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-10.0.0.68432.zip
```



Unzip the downloaded file.

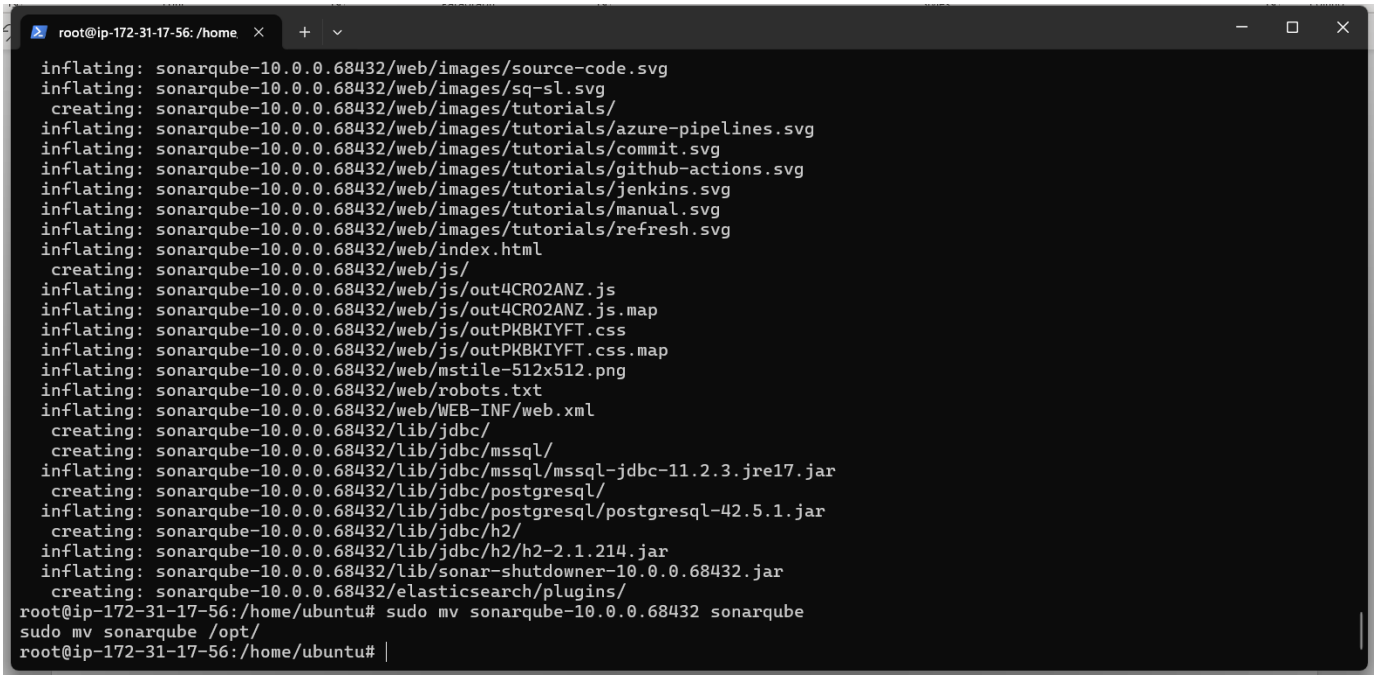
```
sudo unzip sonarqube-10.0.0.68432.zip
```



Move the unzipped files to /opt/sonarqube directory

```
sudo mv sonarqube-10.0.0.68432 sonarqube
```

```
sudo mv sonarqube /opt/
```



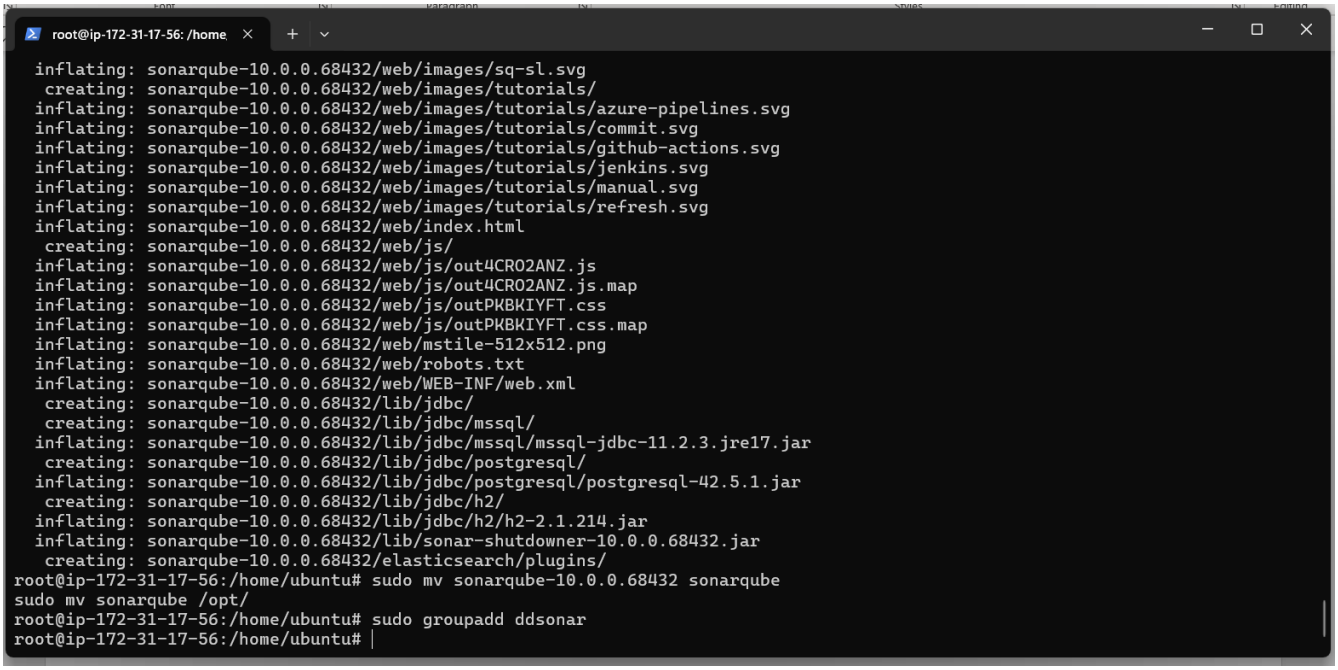
Part 2: Add SonarQube Group and User

Create a dedicated user and group for SonarQube, which can not run as the root user.

Note: You can give any name for the sonar user and group. I have here given the user and group name to be the same i.e **ddsonar**.

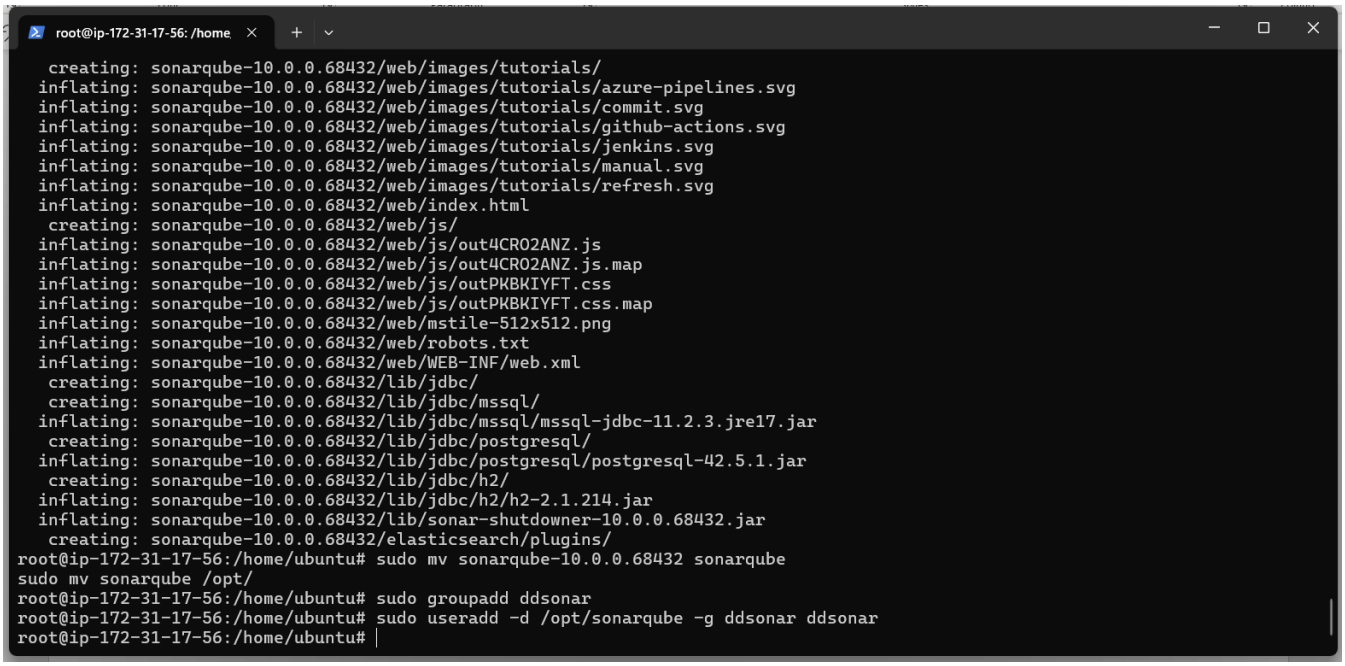
Create a sonar group.

```
sudo groupadd ddsonar
```



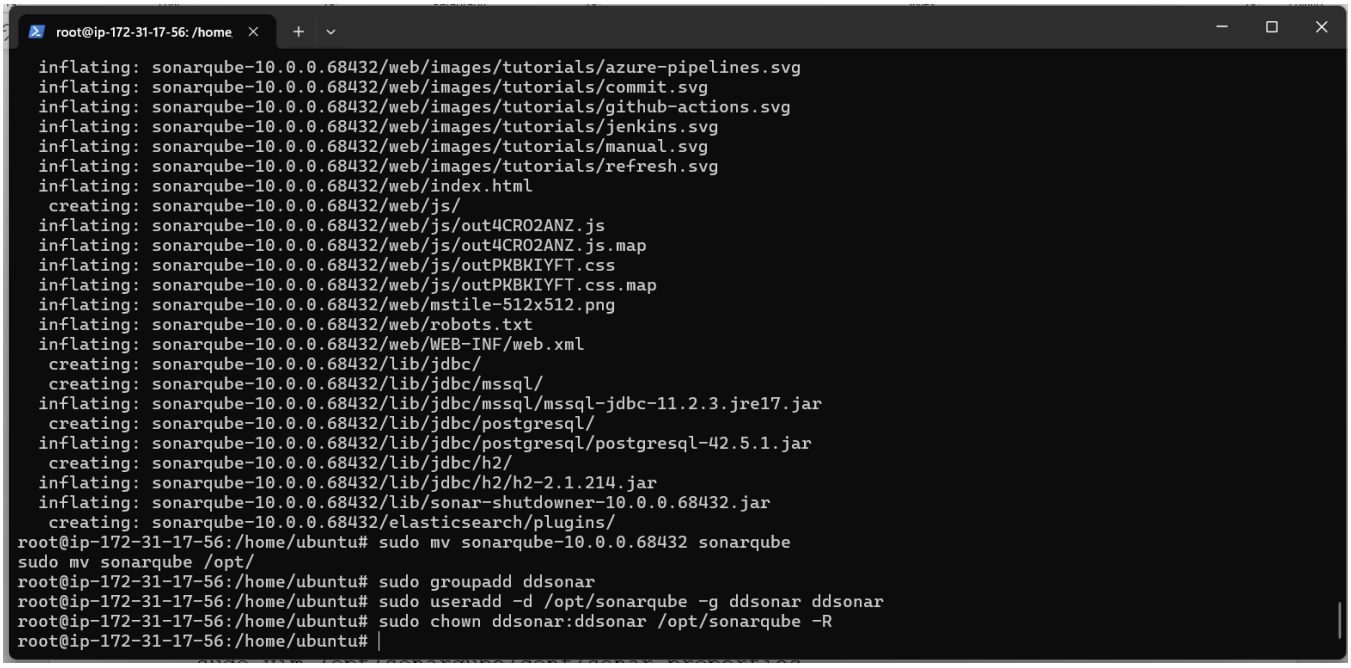
Create a sonar user and set /opt/sonarqube as the home directory.

```
sudo useradd -d /opt/sonarqube -g ddsonar ddsonar
```



Grant the sonar user access to the /opt/sonarqube directory.

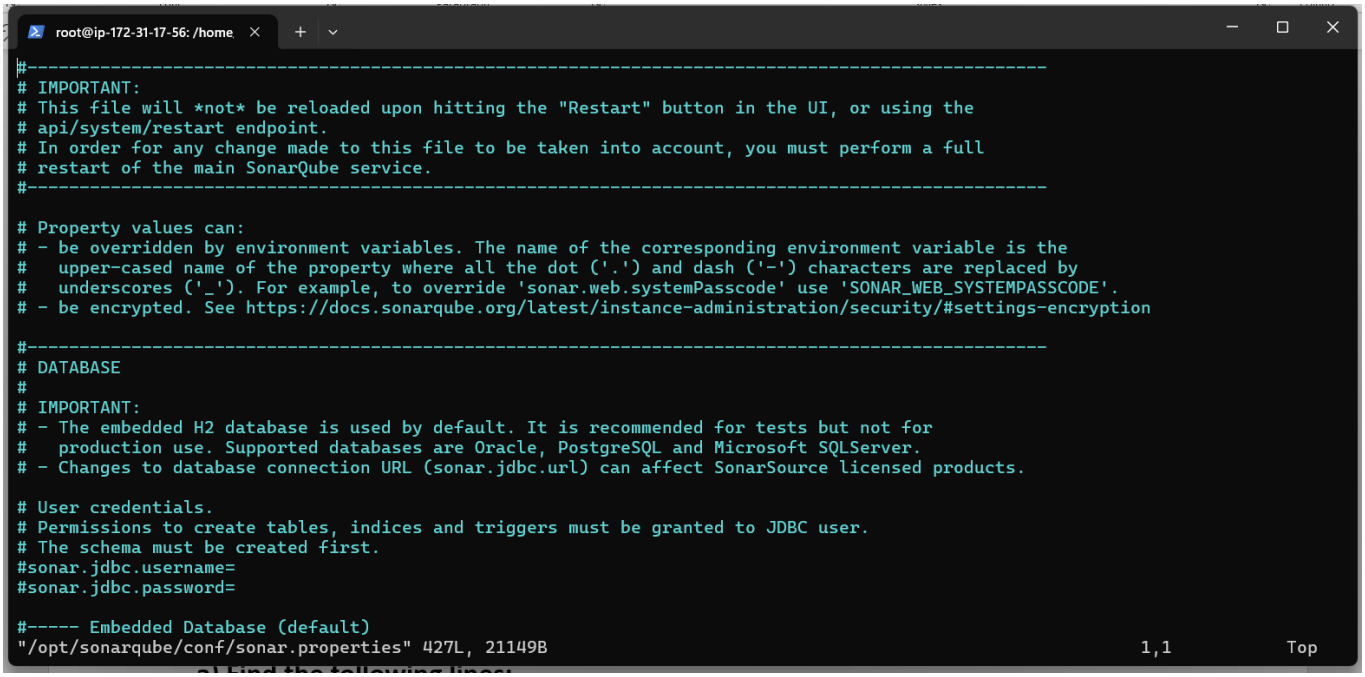
```
sudo chown ddsonar:ddsonar /opt/sonarqube -R
```



Part 3: Configure SonarQube

Edit the SonarQube configuration file.

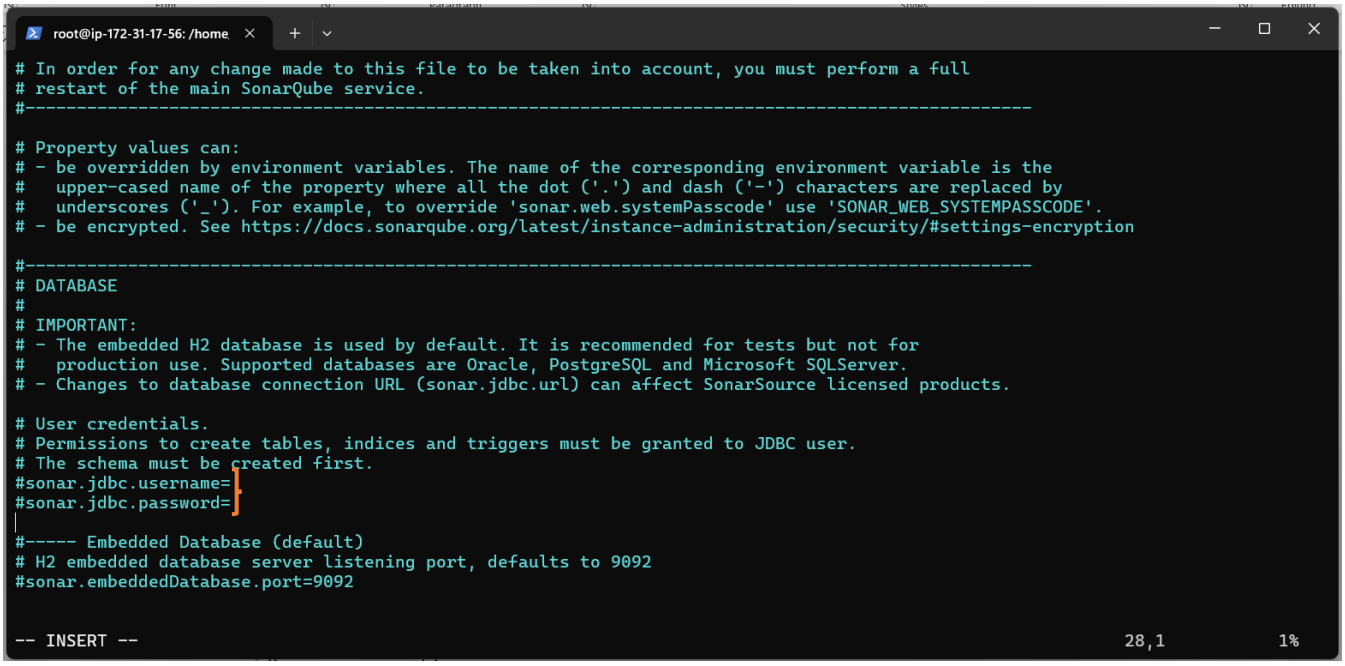
```
sudo vim /opt/sonarqube/conf/sonar.properties
```



Find the following lines:

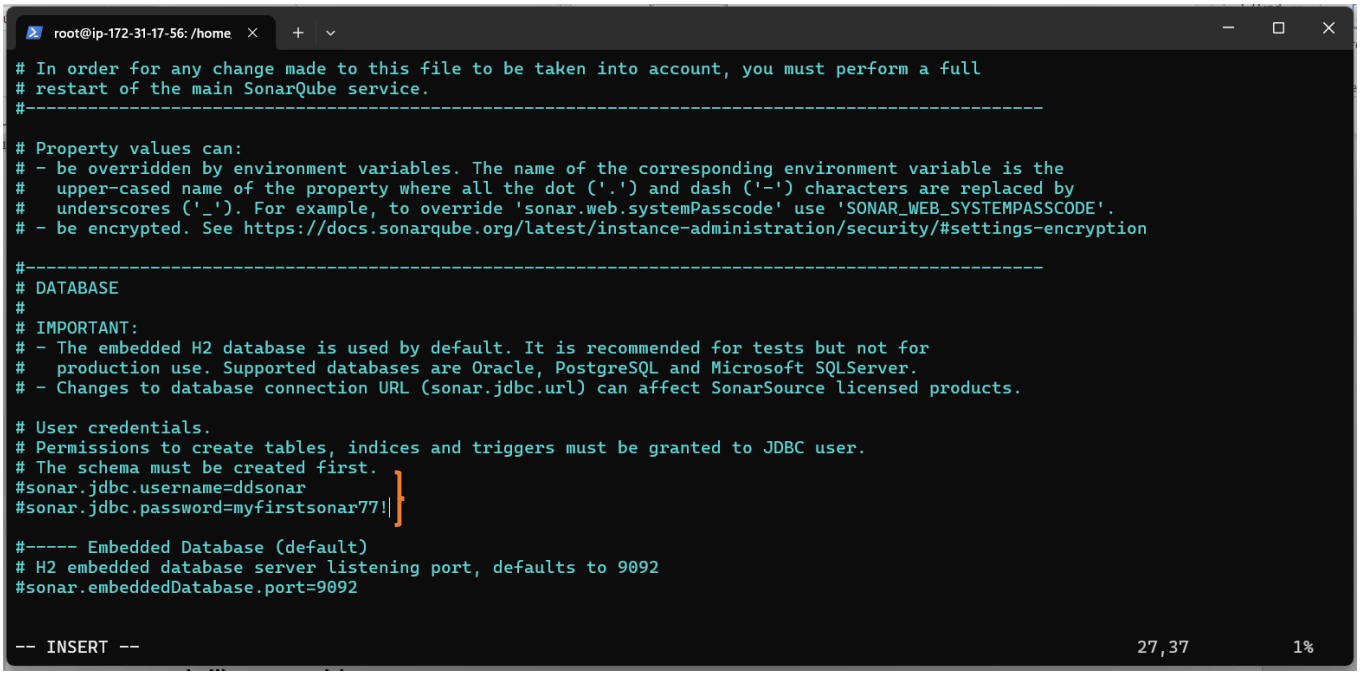
```
#sonar.jdbc.username=
```

```
#sonar.jdbc.password=
```



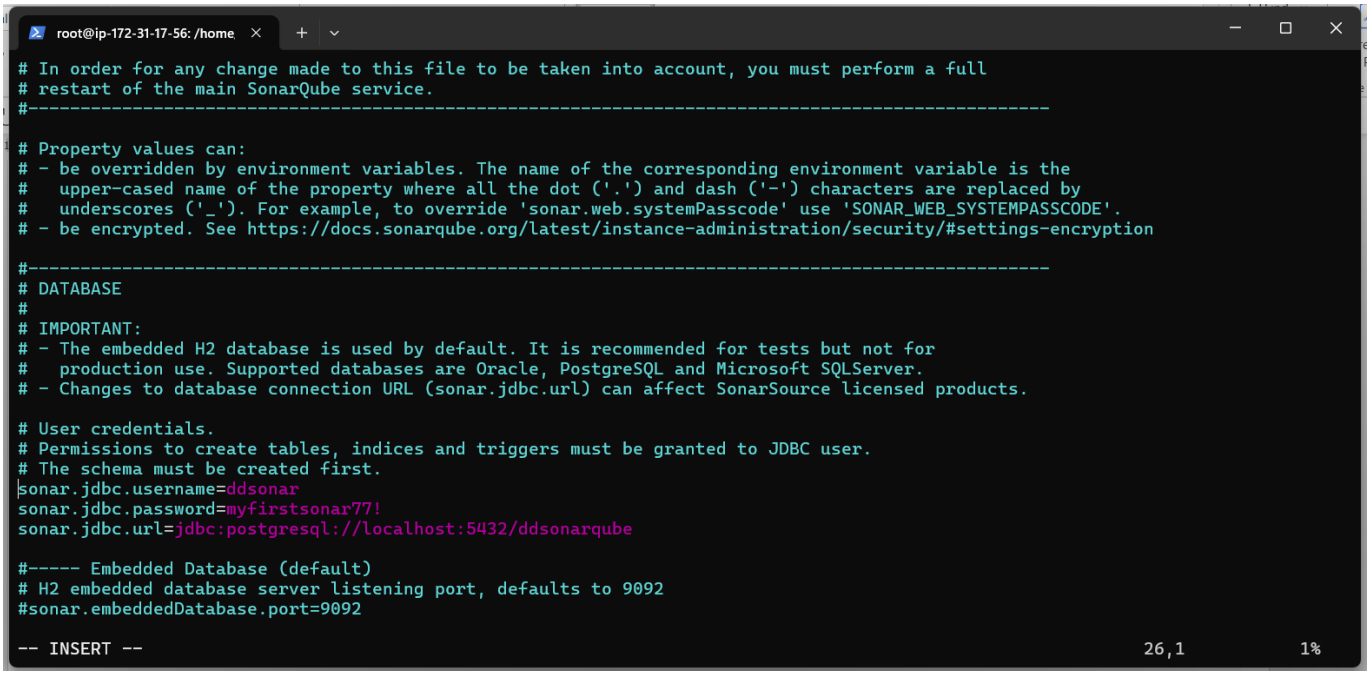
Uncomment the lines, and add the database user and Database password you created in Step 3. For me, it's:

```
sonar.jdbc.username=ddsonar
sonar.jdbc.password=myfirstsonar77!
```



Below these two lines, add the following line of code.

```
sonar.jdbc.url=jdbc:postgresql://localhost:5432/ddsonarqube
```



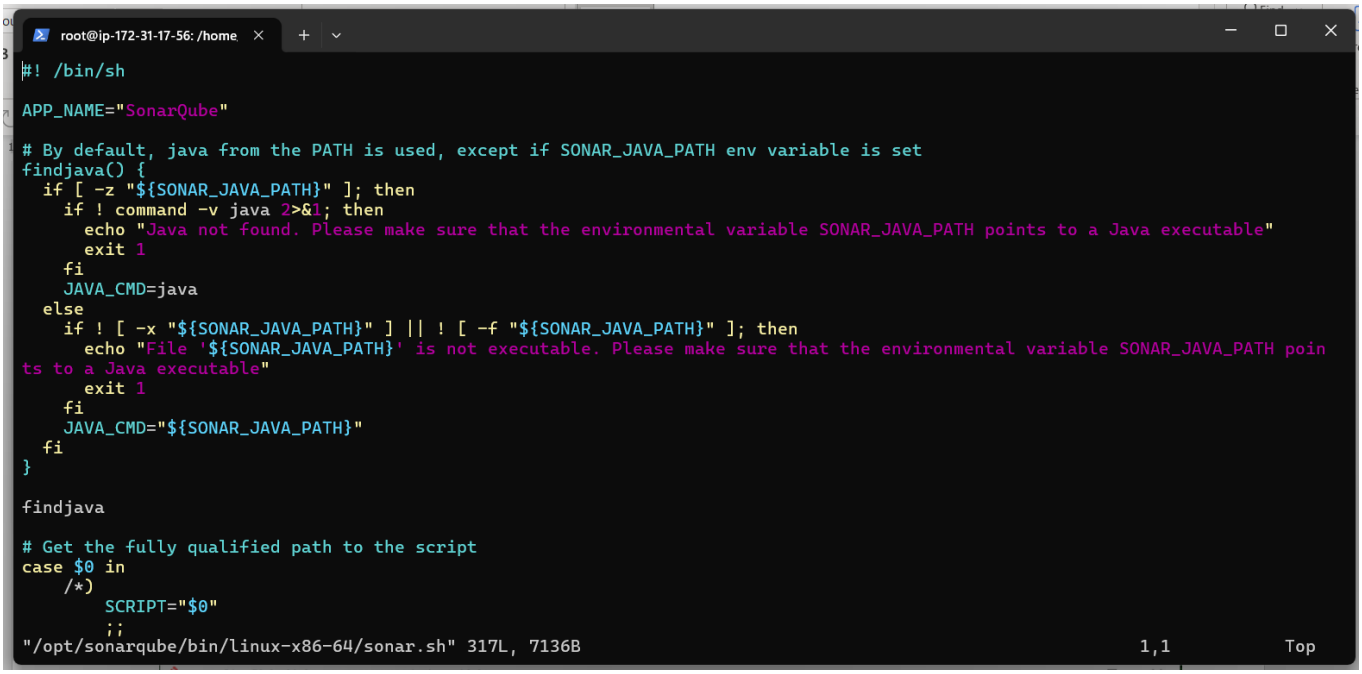
Here, ddsonarqube is the database name created.

Save and exit the file.

Press ESC, followed by :wq and press ENTER

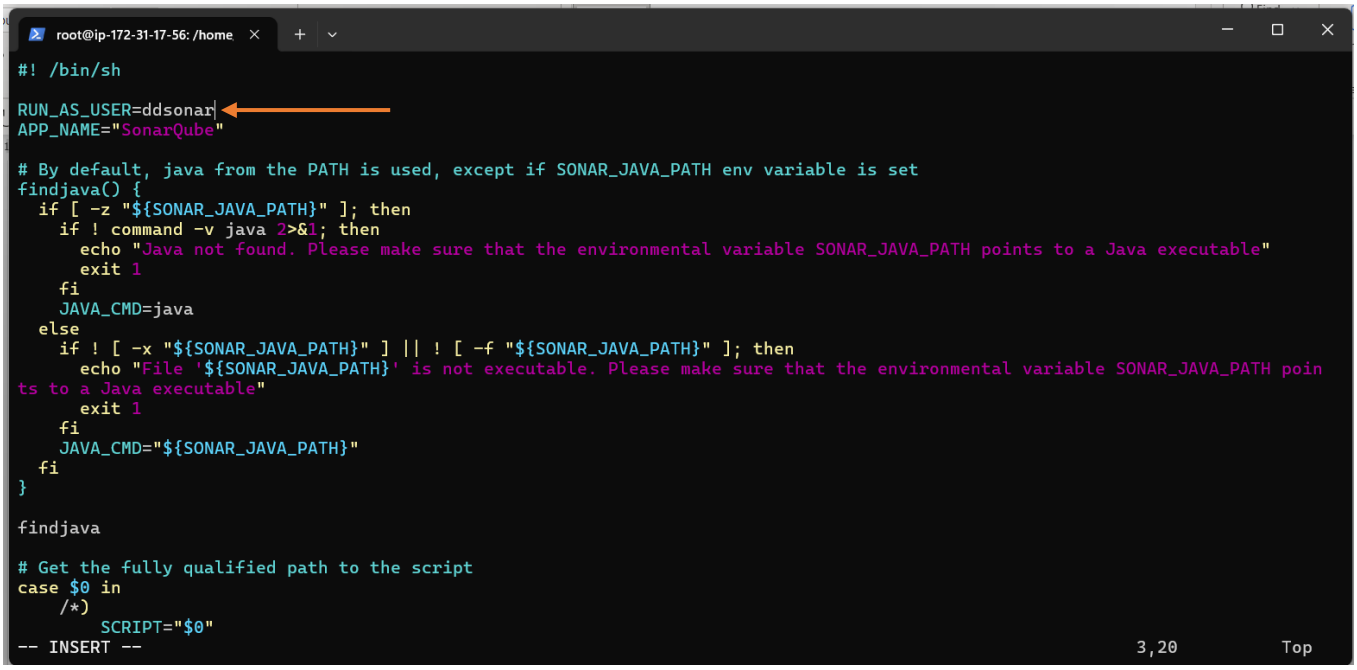
Edit the sonar script file.

`sudo vim /opt/sonarqube/bin/linux-x86-64/sonar.sh`



Add the following line

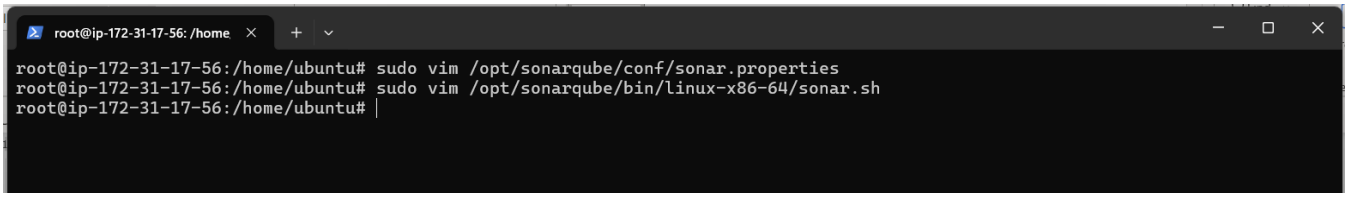
`RUN_AS_USER=ddsonar`



Here, ddsonar is the name of the user that we have created in step number 6 (ii).

Save and exit the file.

Press ESC, followed by :wq and press ENTER



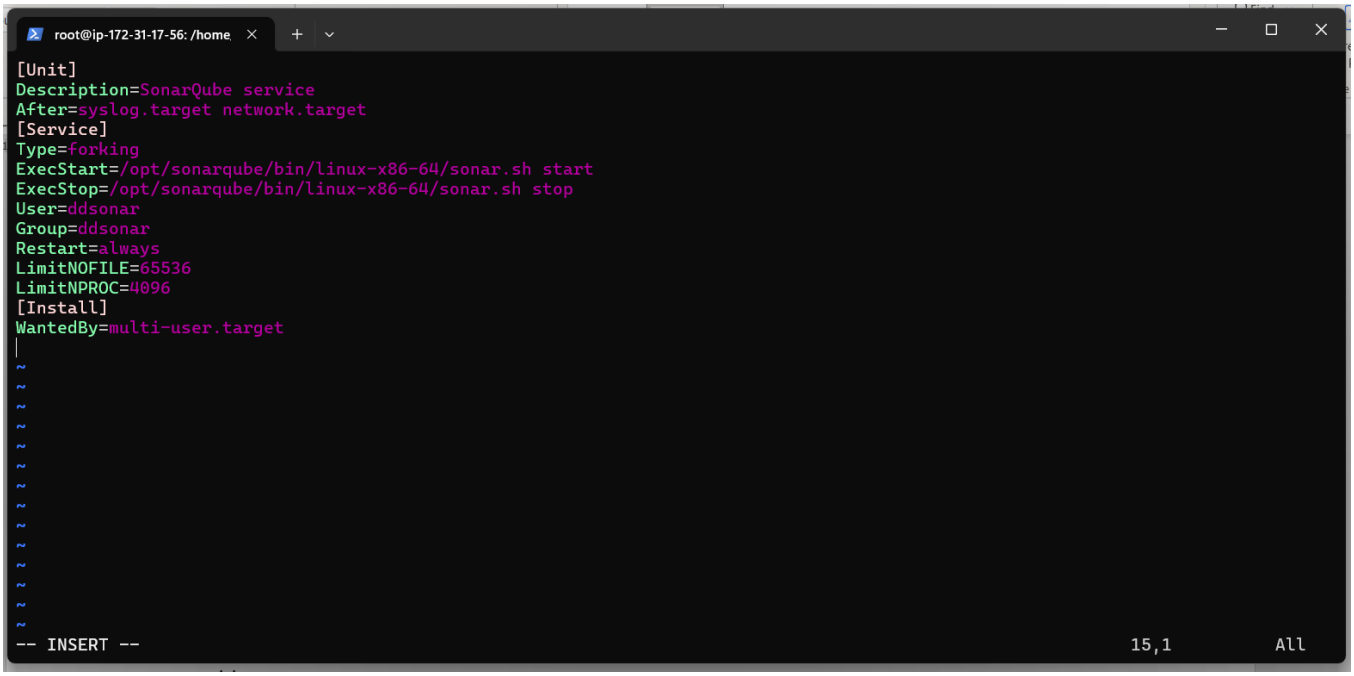
Part 4: Setup Systemd service

Create a systemd service file to start SonarQube at system boot.

```
sudo vim /etc/systemd/system/sonar.service
```

Paste the following lines to the file.

```
[Unit]
Description=SonarQube service
After=syslog.target network.target
[Service]
Type=forking
ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start
ExecStop=/opt/sonarqube/bin/linux-x86-64/sonar.sh stop
User=ddsonar
Group=ddsonar
Restart=always
LimitNOFILE=65536
LimitNPROC=4096
[Install]
WantedBy=multi-user.target
```



Note: Here in the above script, make sure to change the User and Group section with the value that you have created. For me its:

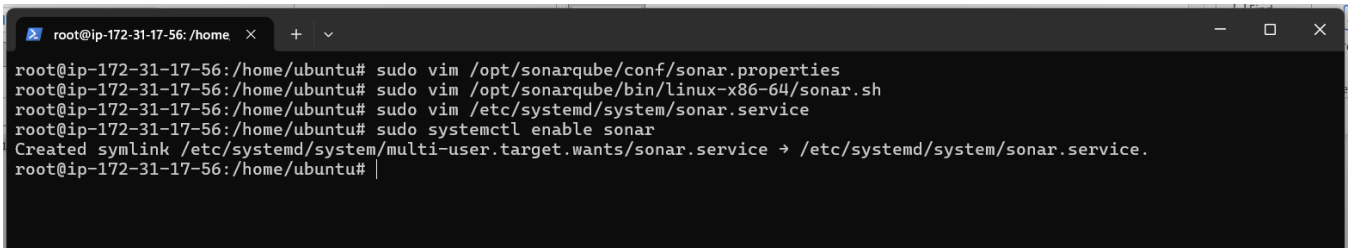
```
User=ddsonar
Group=ddsonar
```

Save and exit the file.

Press ESC, followed by :wq then press ENTER

Enable the SonarQube service to run at system startup.

```
sudo systemctl enable sonar
```

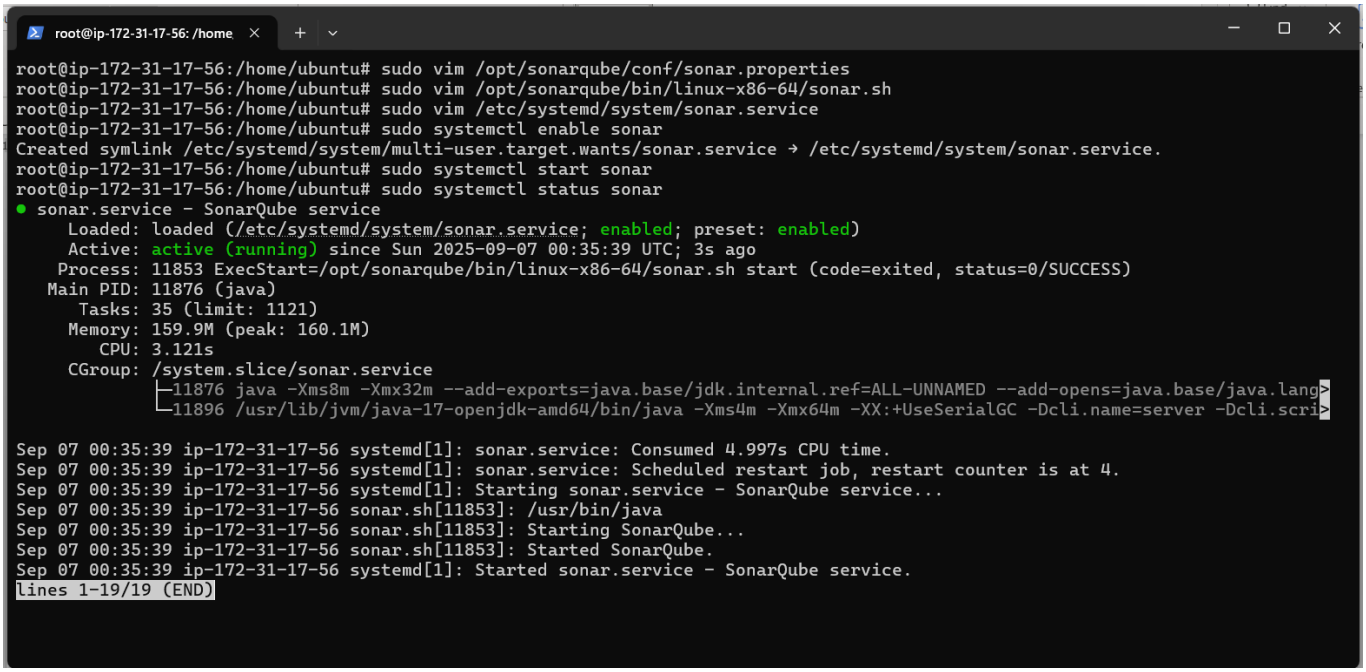


Start the SonarQube service.

```
sudo systemctl start sonar
```

Check the service status.

```
sudo systemctl status sonar
```



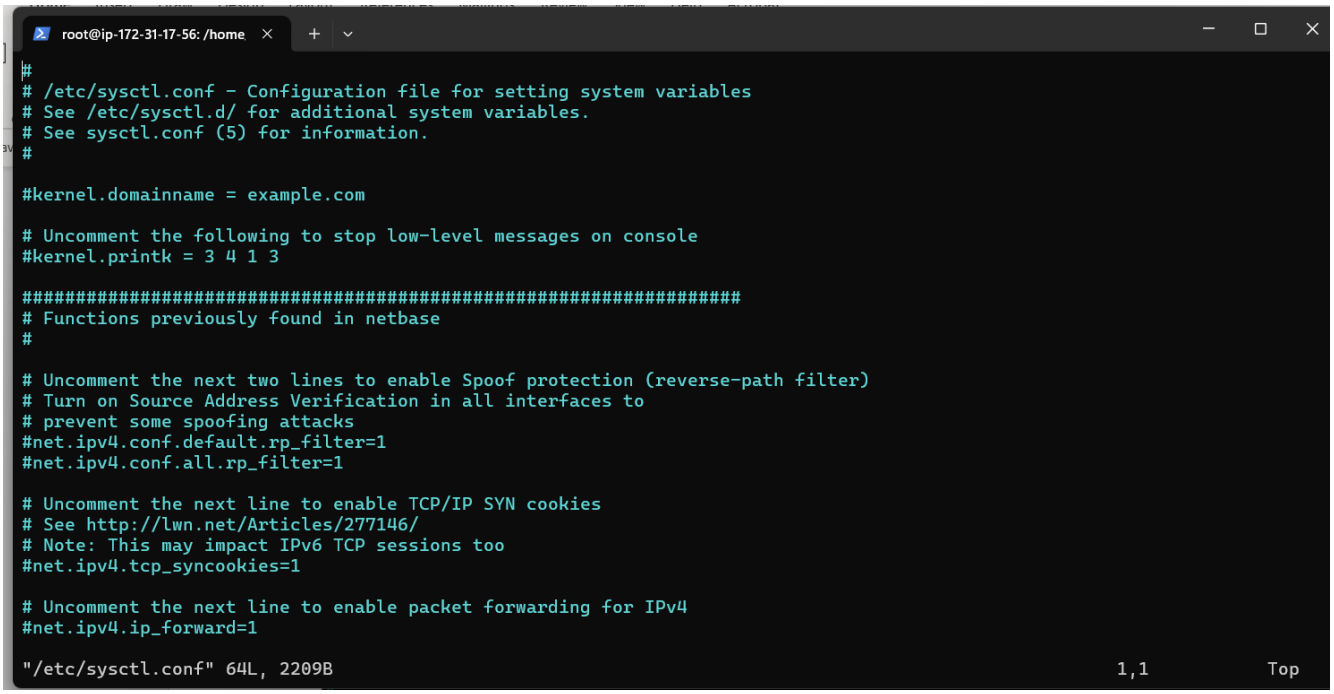
It is up and running. Press “**q**” to exit this mode

Part 5: Modify Kernel System Limits

SonarQube uses Elasticsearch to store its indices in an MMap FS directory. It requires some changes to the system defaults.

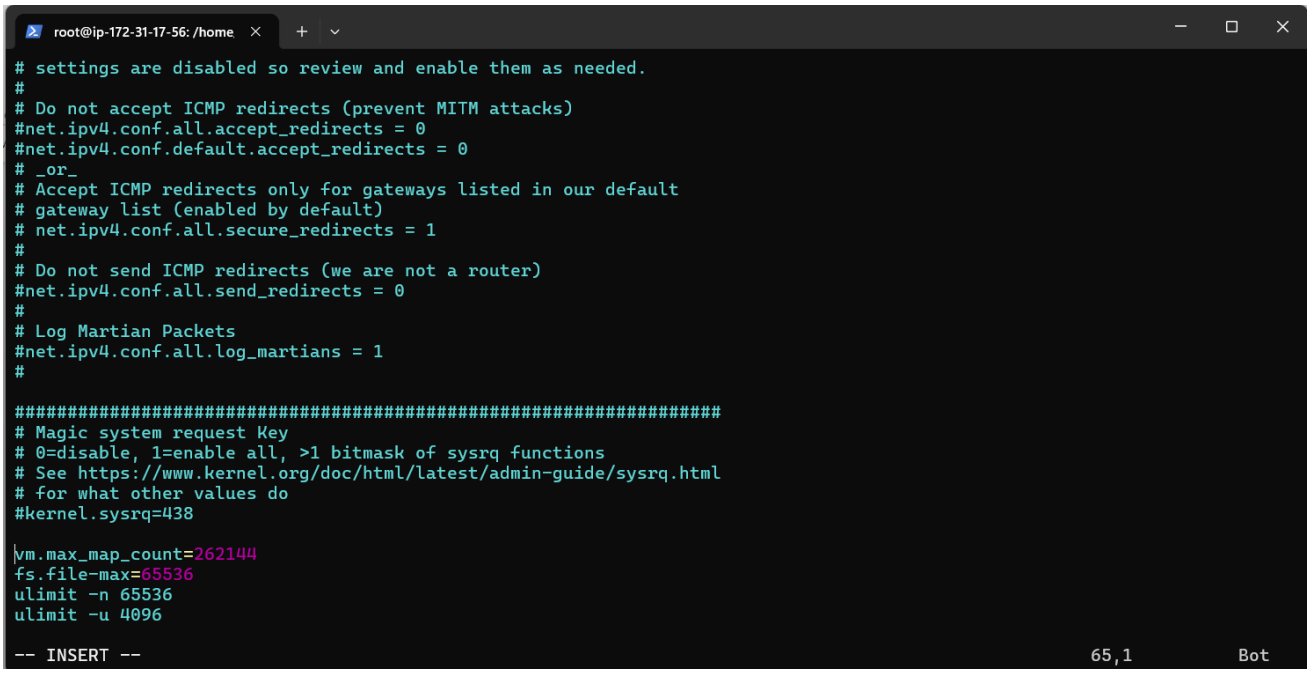
Edit the sysctl configuration file.

```
sudo vim /etc/sysctl.conf
```



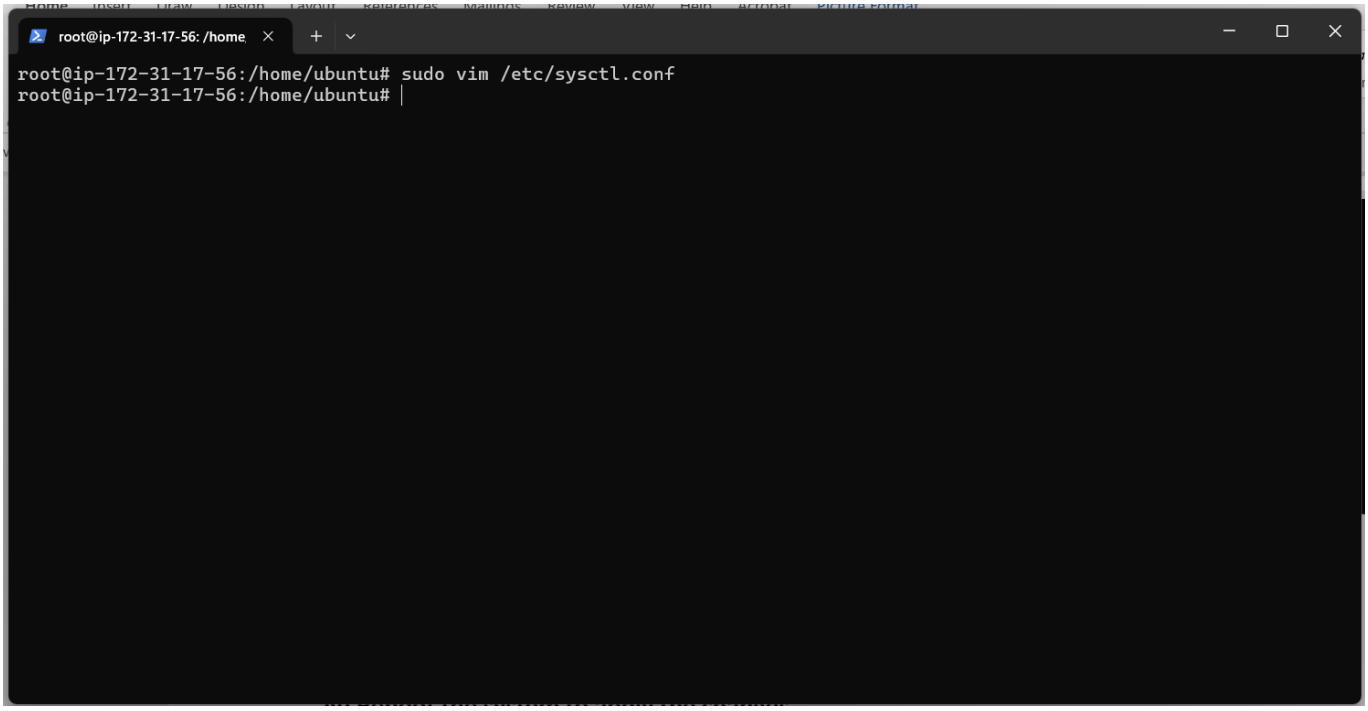
Add the following lines.

```
vm.max_map_count=262144
fs.file-max=65536
ulimit -n 65536
ulimit -u 4096
```



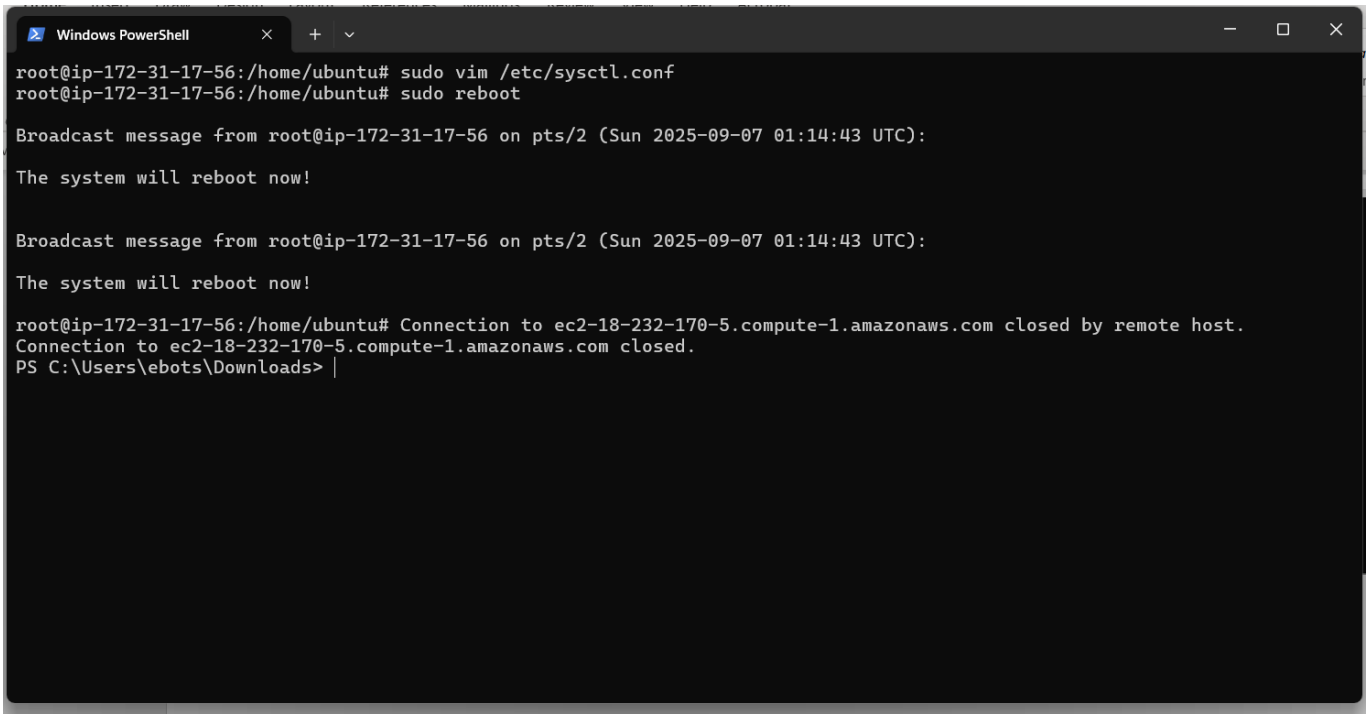
Save and exit the file

Press ESC, followed by :wq and press ENTER



Reboot the system to apply the changes.

```
sudo reboot
```

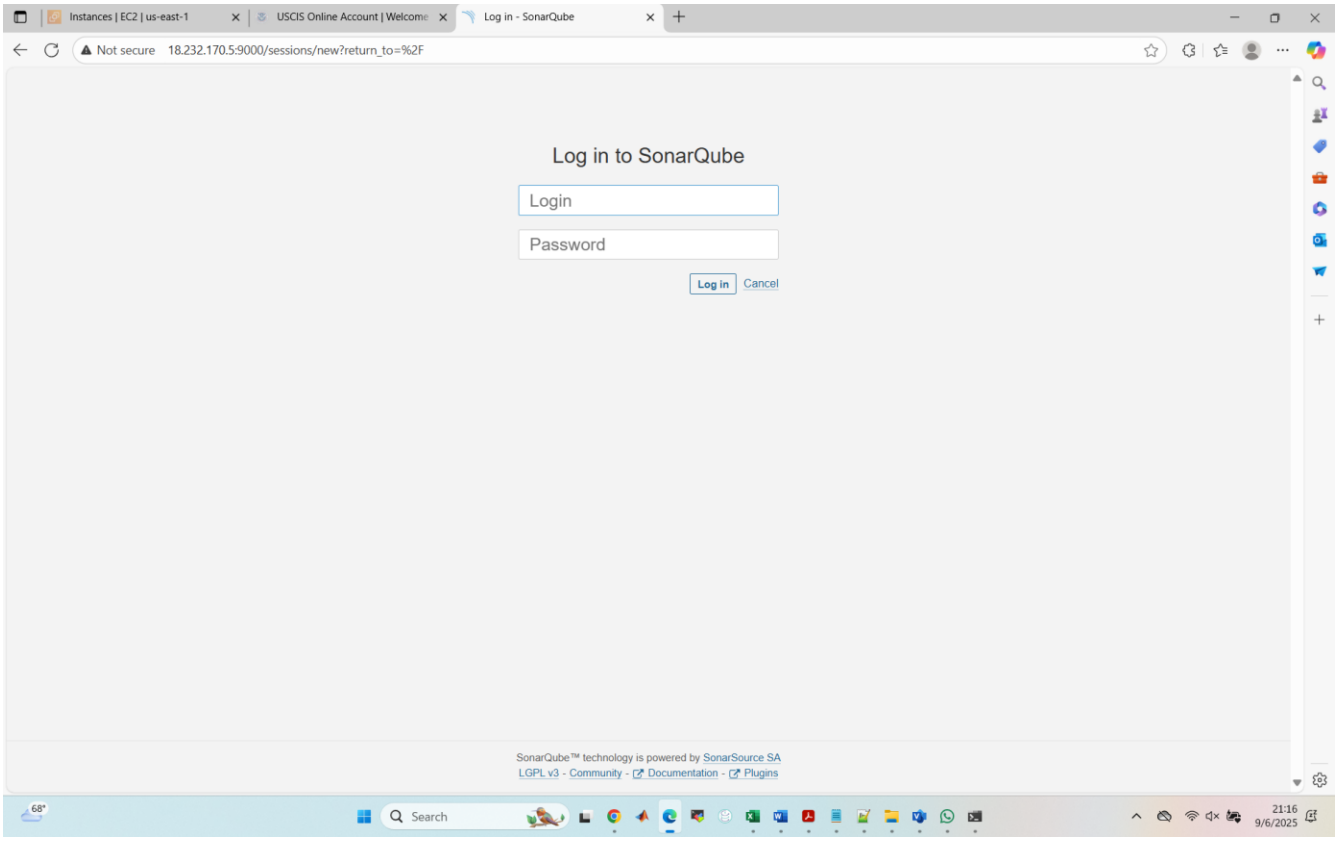

Windows PowerShell
root@ip-172-31-17-56:/home/ubuntu# sudo vim /etc/sysctl.conf
root@ip-172-31-17-56:/home/ubuntu# sudo reboot
Broadcast message from root@ip-172-31-17-56 on pts/2 (Sun 2025-09-07 01:14:43 UTC):
The system will reboot now!
Broadcast message from root@ip-172-31-17-56 on pts/2 (Sun 2025-09-07 01:14:43 UTC):
The system will reboot now!
root@ip-172-31-17-56:/home/ubuntu# Connection to ec2-18-232-170-5.compute-1.amazonaws.com closed by remote host.
Connection to ec2-18-232-170-5.compute-1.amazonaws.com closed.
PS C:\Users\ebots\Downloads>

STEP 5: Access SonarQube Web Interface

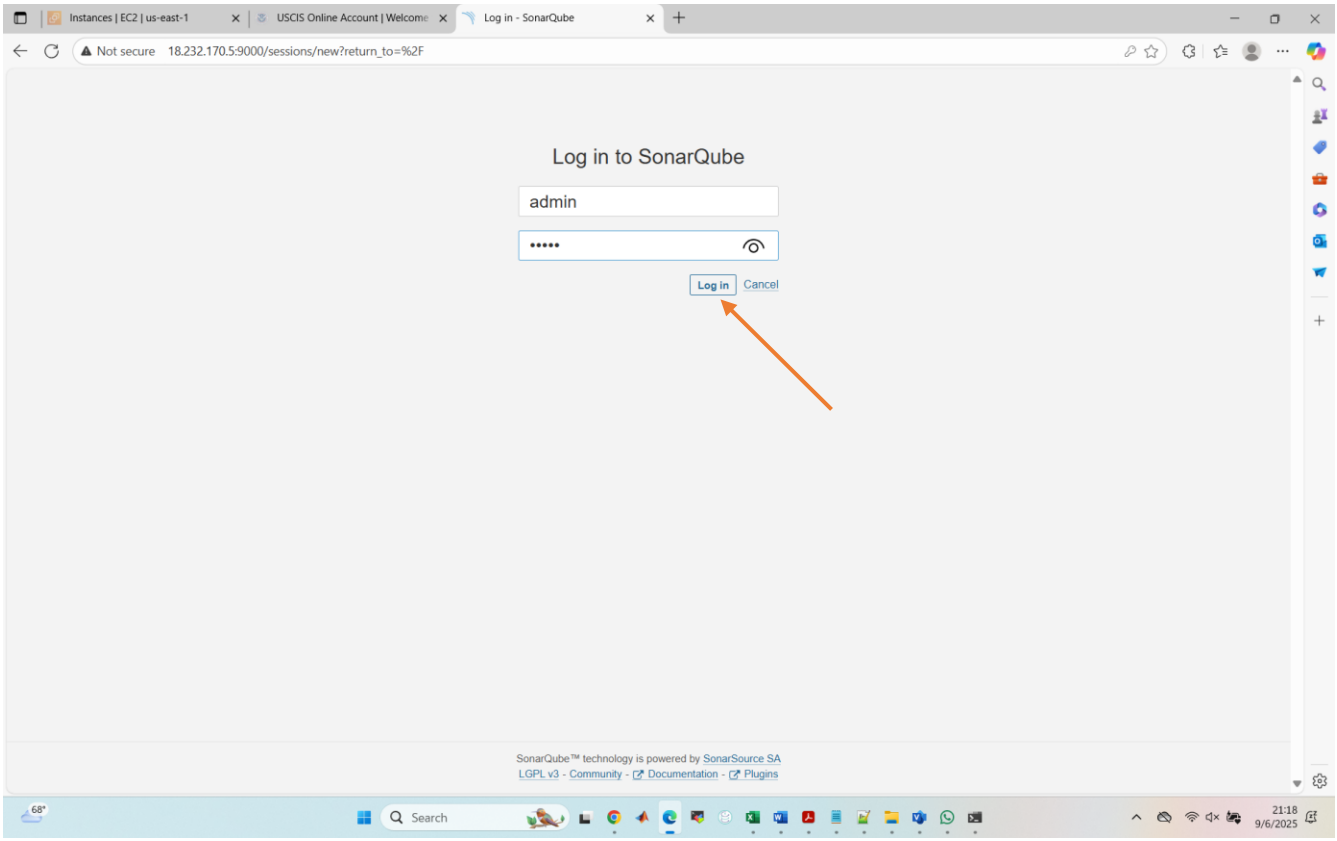
Access SonarQube in a web browser at your server's IP address on port 9000.

For example, http://Public IPv4 Address:9000

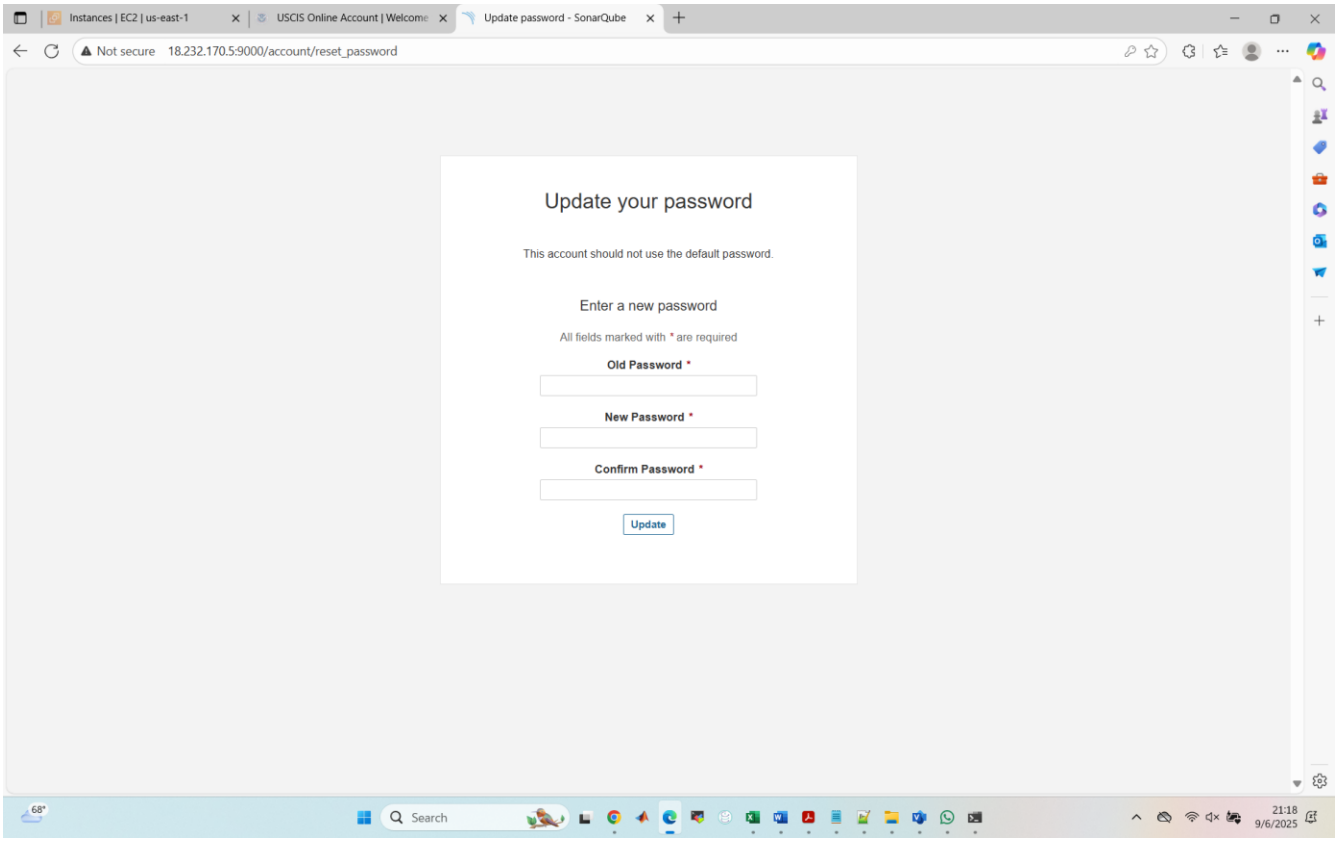
That is **http://18.232.170.5:9000**



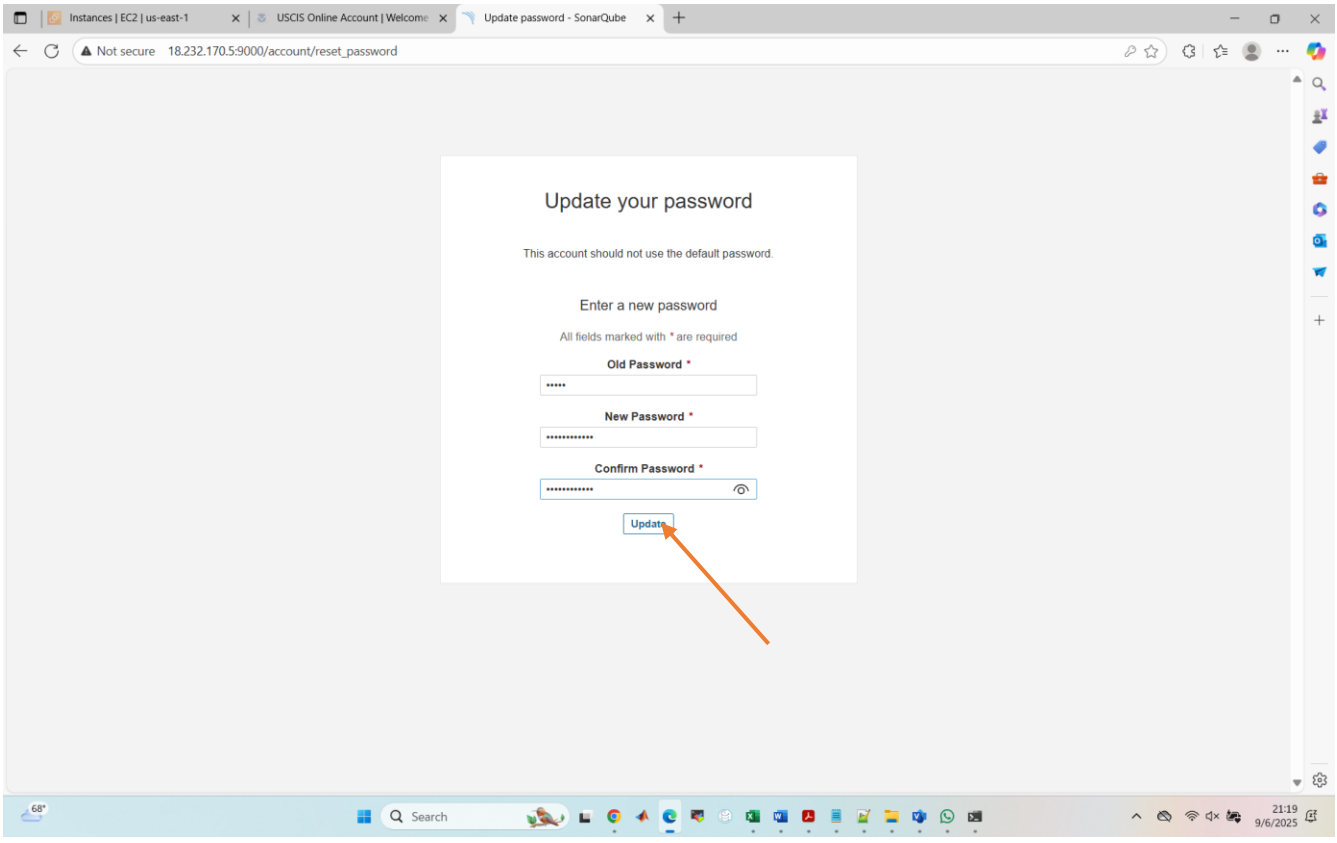
Note: the default username and password are **admin** and **admin** respectively.



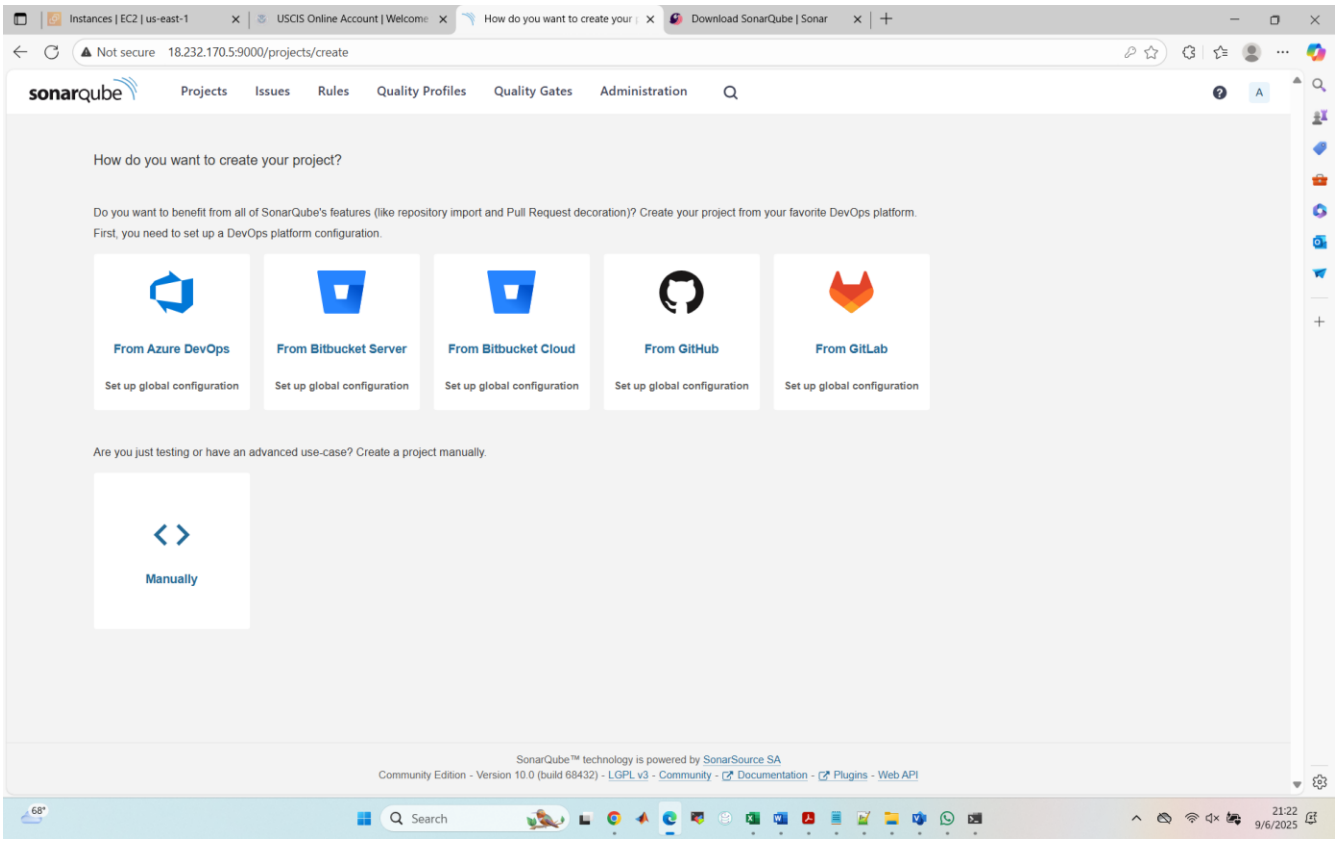
Click on “Login”. In the next step, SonarQube will prompt you to change your password. CHANGE THE PASSWORD.



Change the Old password with a New one.



Click on “**Update**”



We have successfully installed the SonarQube Community version 10.0 on AWS EC2 Ubuntu.