SonarQube Installation and configuration

SonarQube is an open-source platform used for continuous inspection of code quality. It helps developers identify bugs, code smells, and security vulnerabilities through static code analysis. SonarQube integrates easily with CI/CD tools like Jenkins to maintain high code quality standards.

Let's start the installation of the latest version of SonarQube on Ubuntu 22.04 LTS.

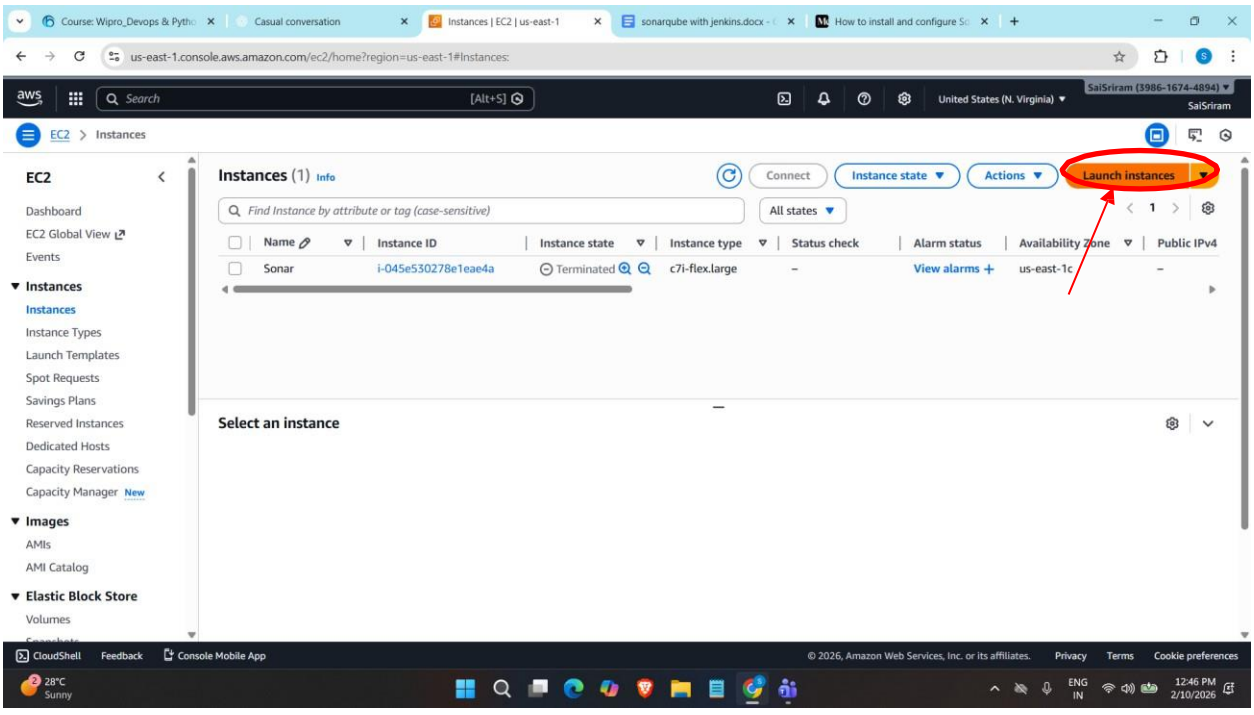
Note: The same steps also work on the AWS EC2 Ubuntu 20.04

Server Specification (Please choose as per your requirement):

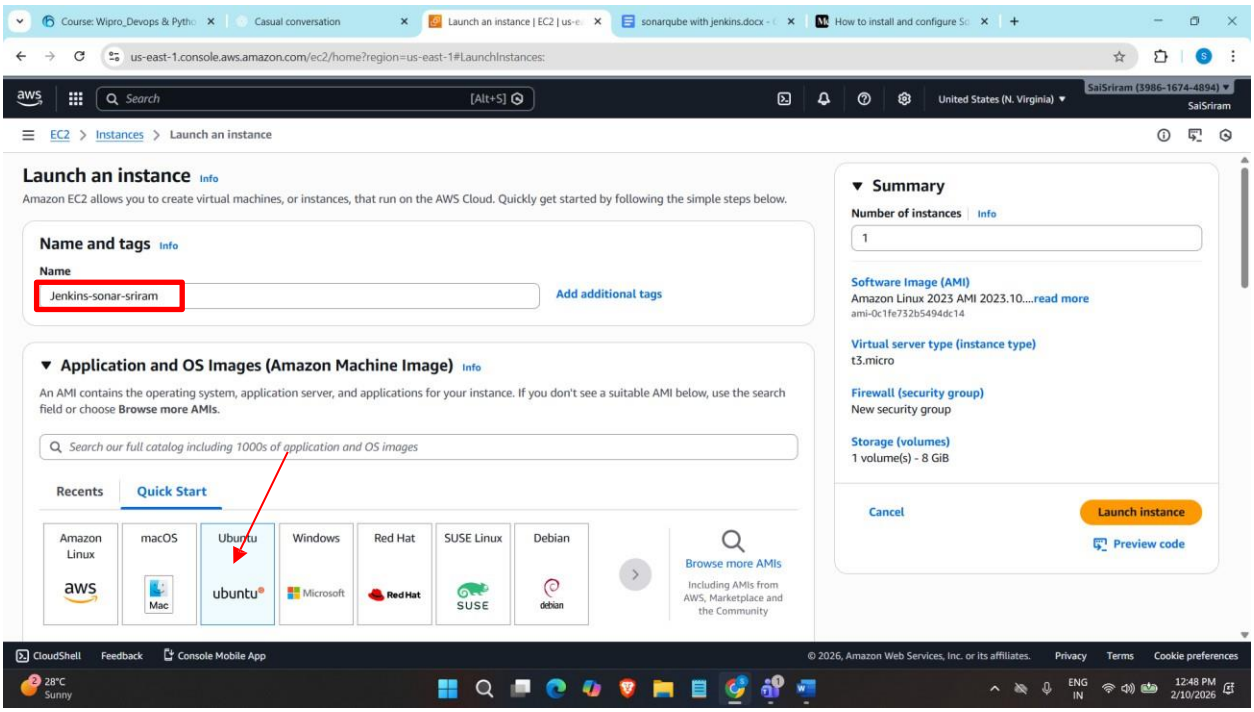
1. **OS** = Ubuntu 22.04 LTS
2. **CPU**: 2 vCPU
3. **RAM**: 4 GB
4. **Storage**: 20 GB

To create an Instance in AWS:

Login AWS ➔ Launch Instance.

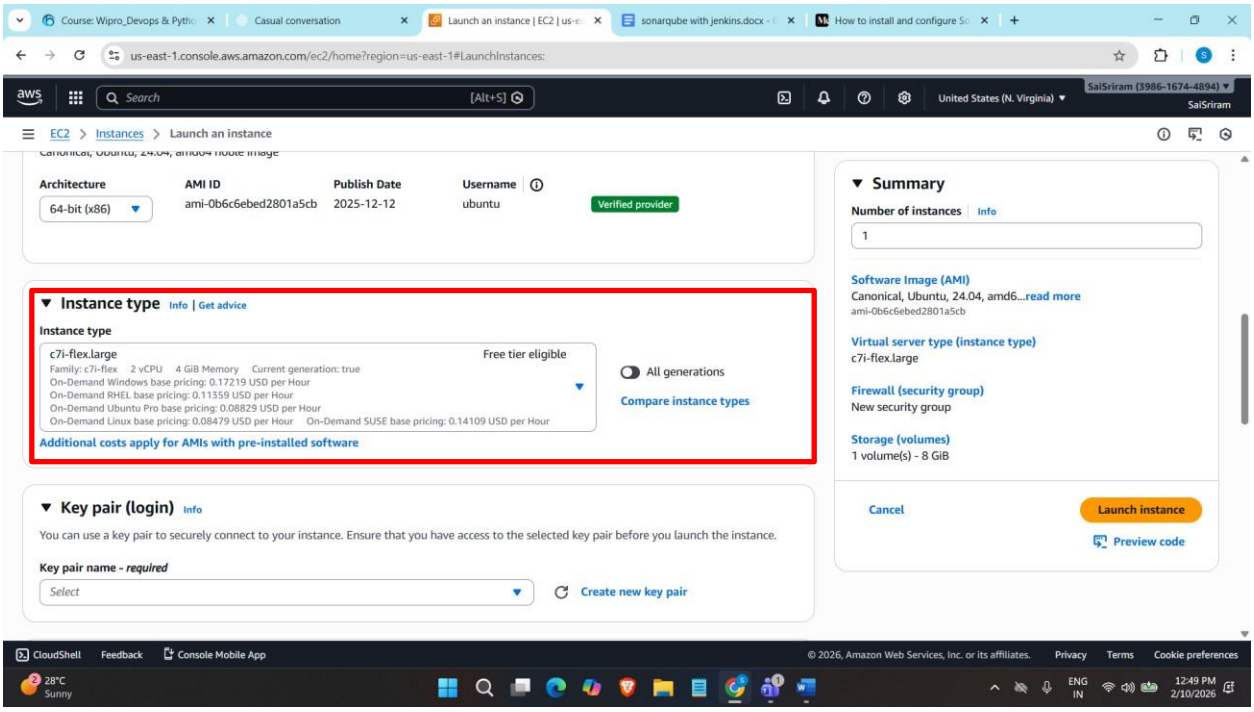


Create name and select machine as ubuntu:

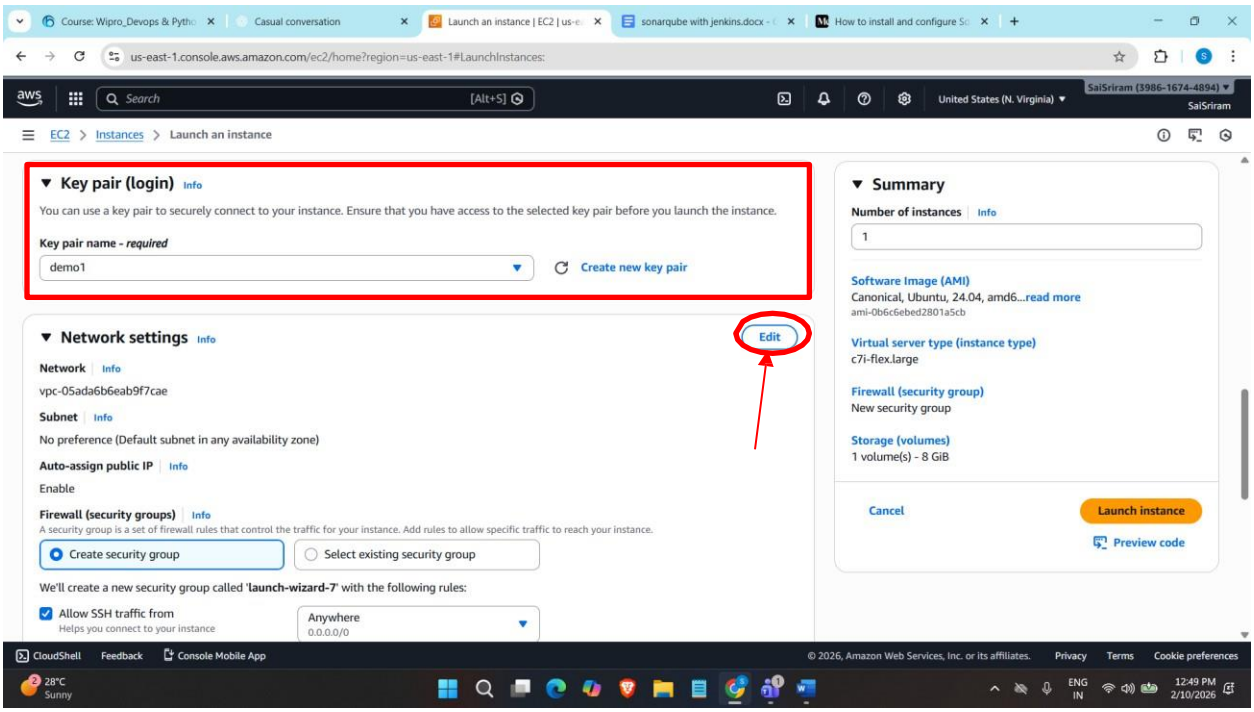


Instance type:

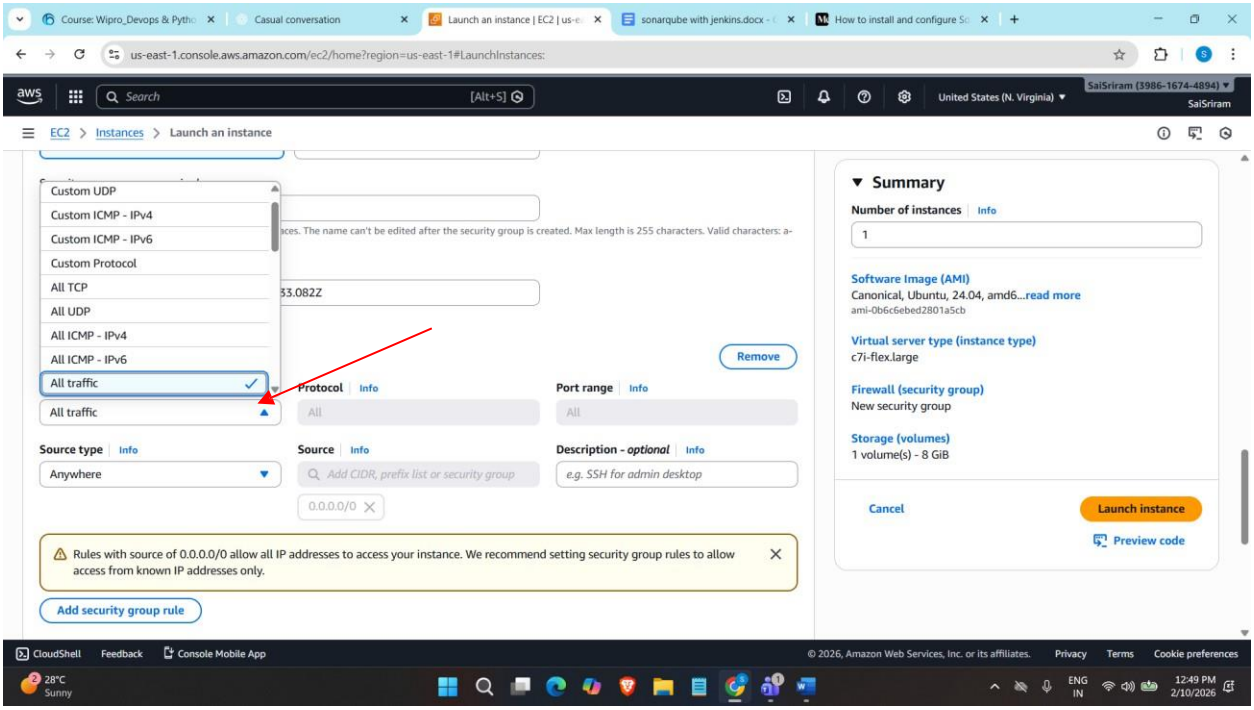
5. Select some large instance type so that you can work easily.



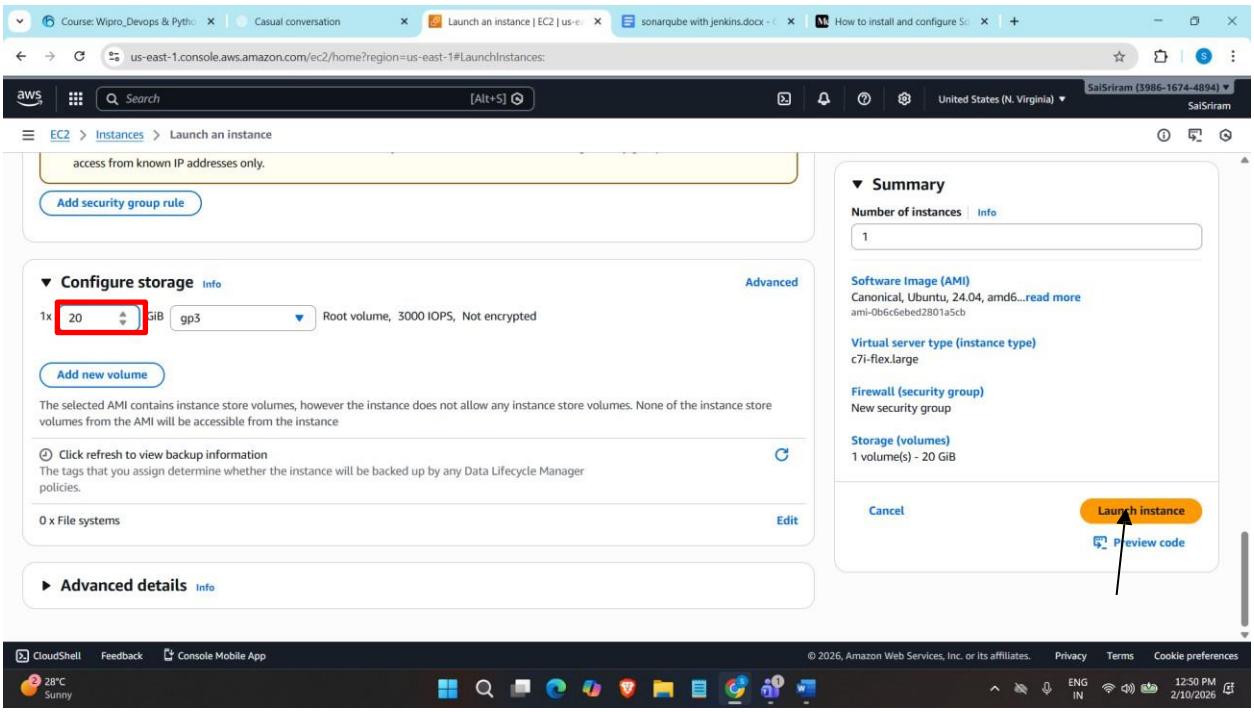
6. **Create a new key pair or you can select the existing one and edit the security settings.**



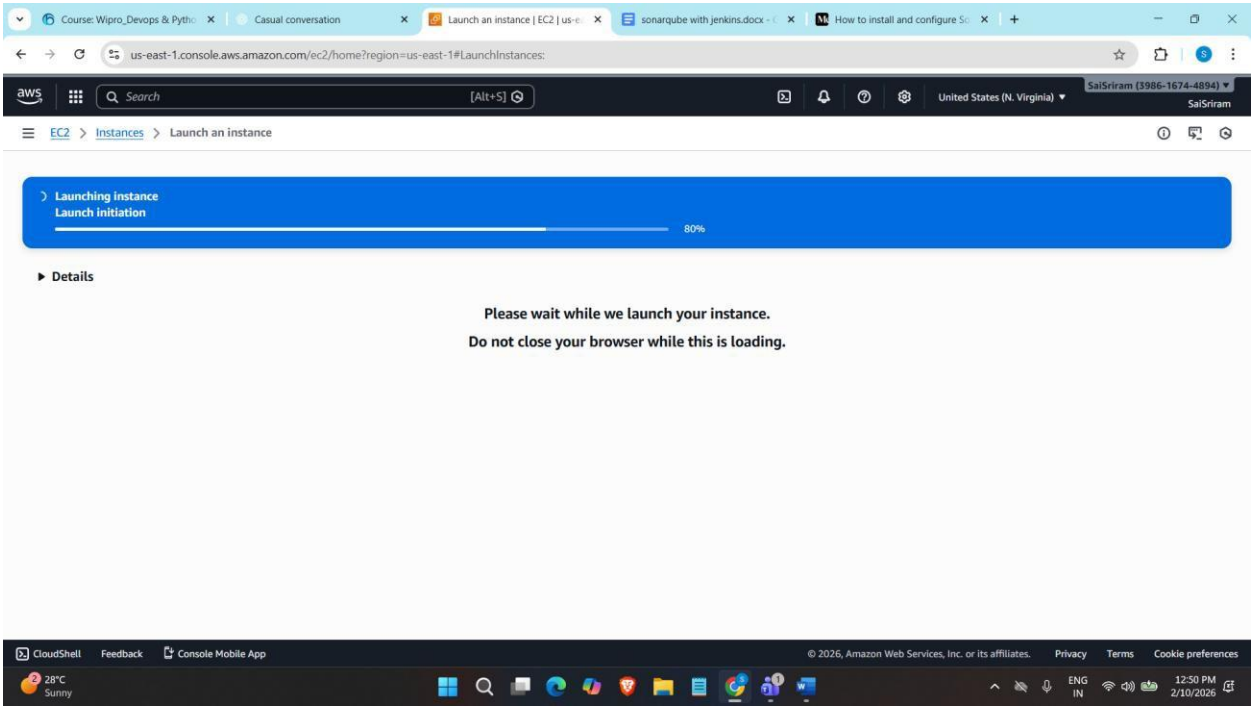
7. **In place of SSH Port select as All traffic so you can launch the page easily with all ports.**



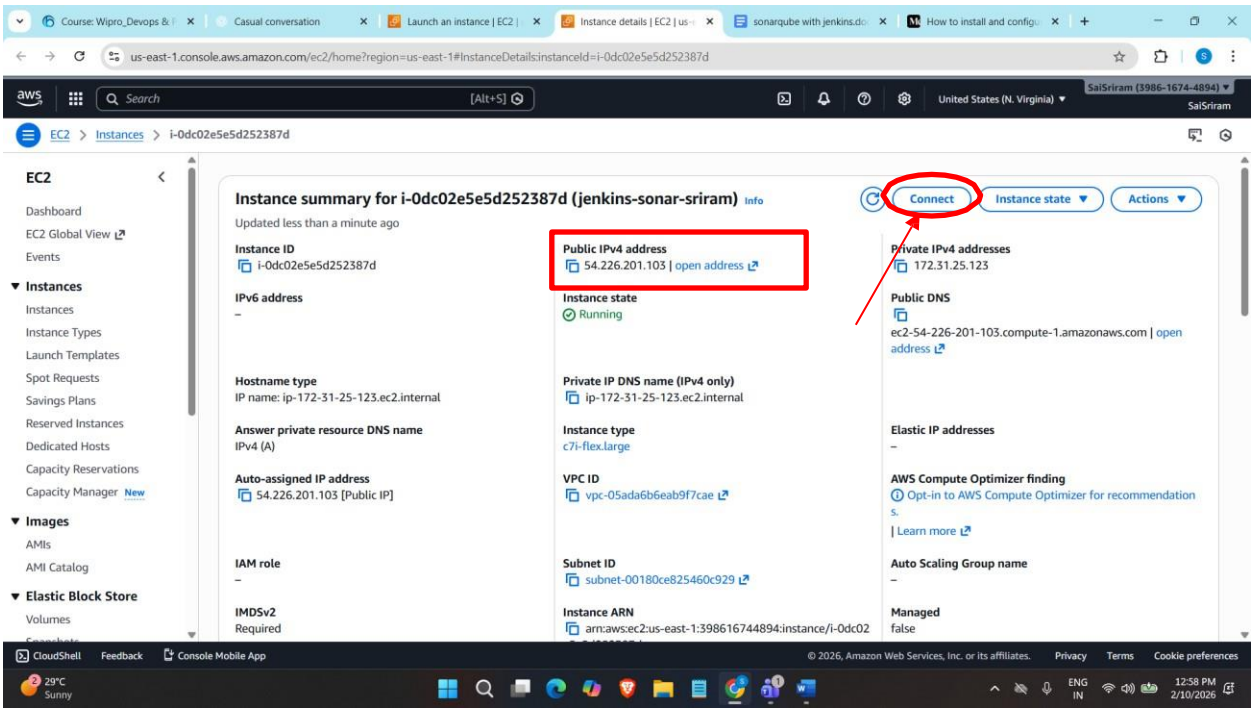
8. Configure the Storage as 20gb and launch instance



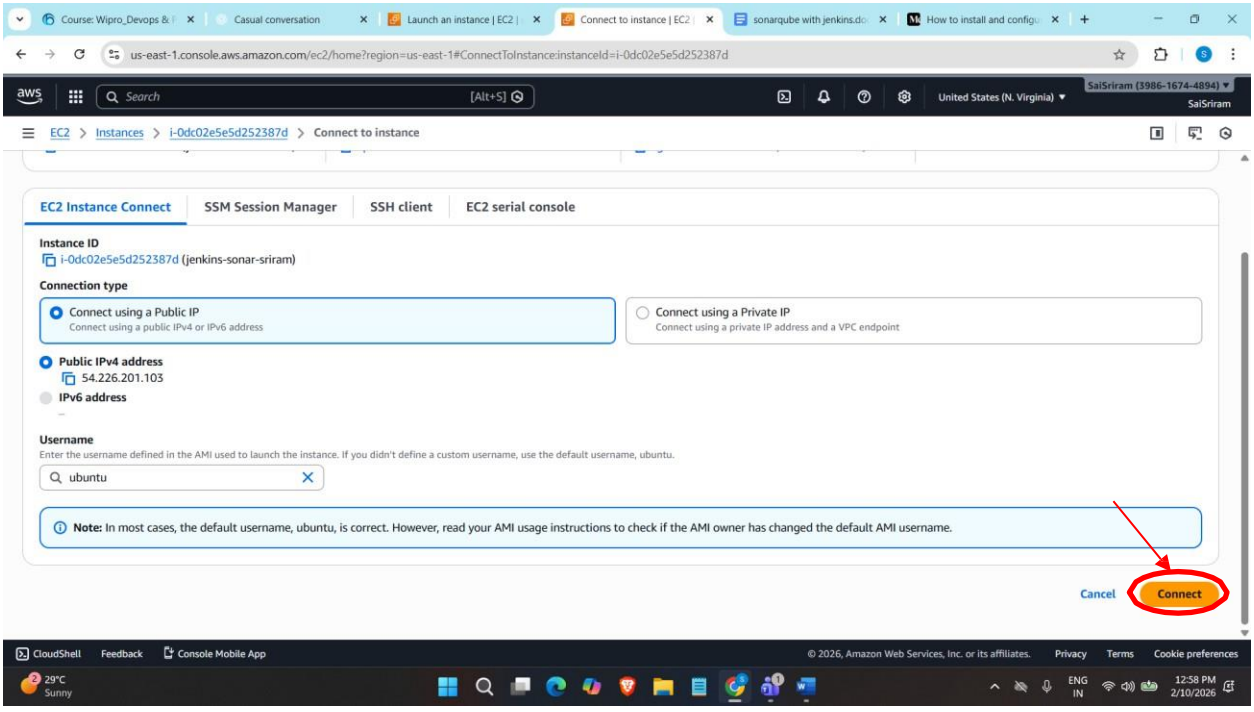
9. Launching Instance



10. Instance created and connect it.

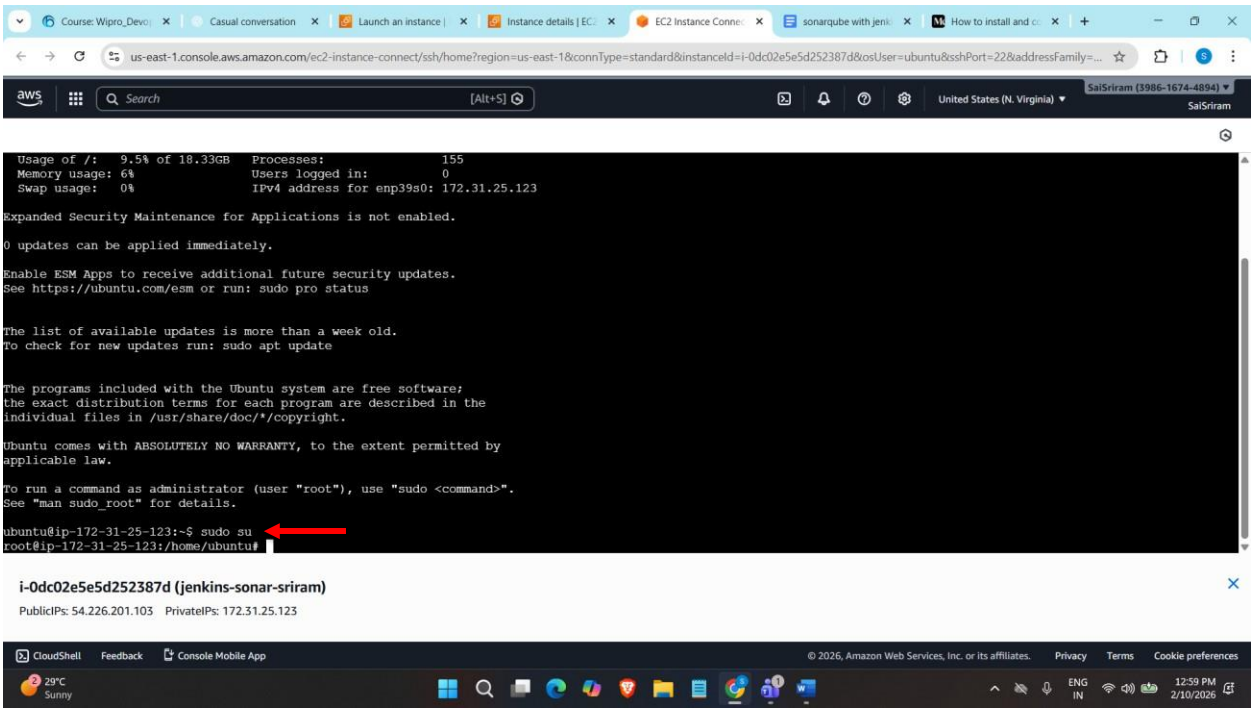


11. Connect using a Public IP and start connecting

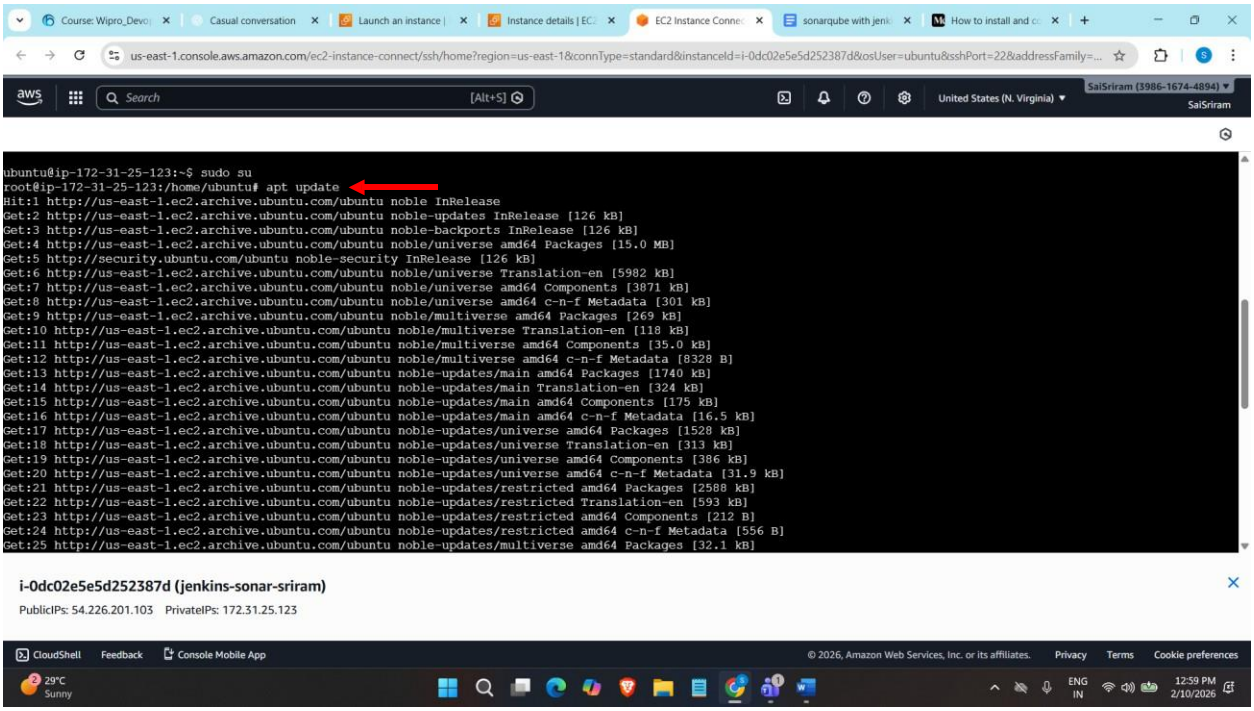


Let's start installing SonarQube and configure things.

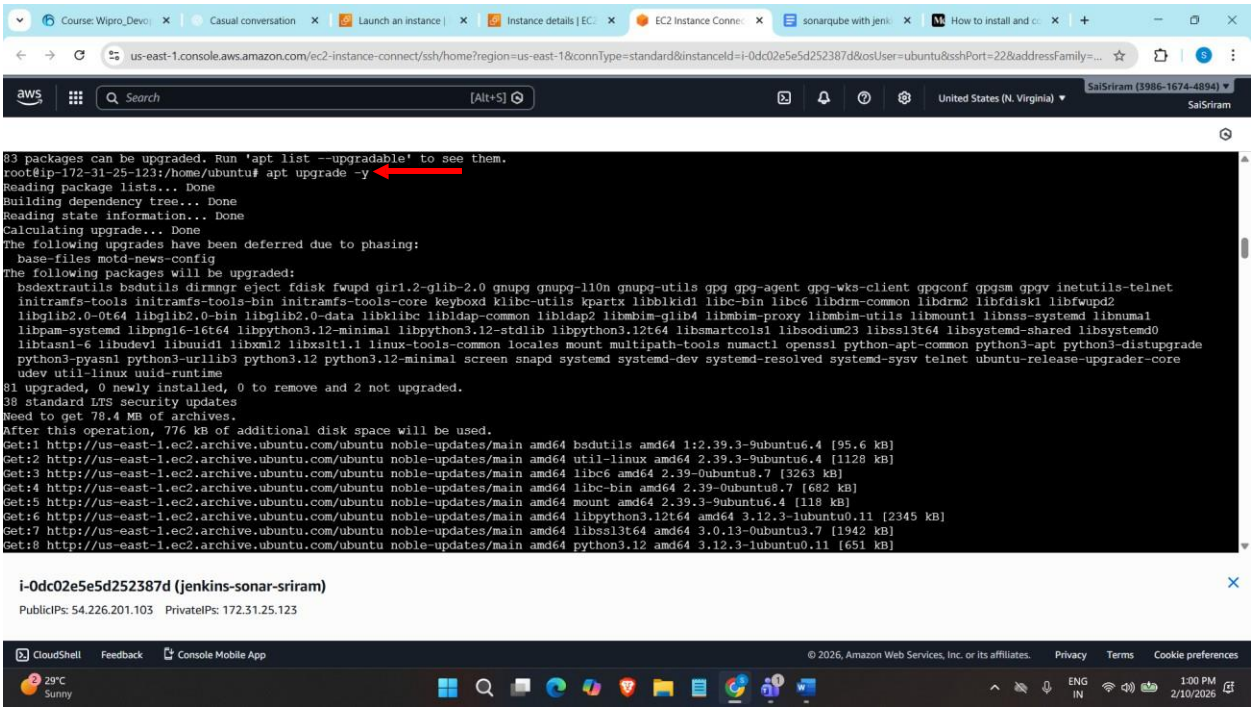
12. **Let's rejuvenate our Ubuntu server Use Command:** `sudo su`



Use command: `apt update or sudo apt update`

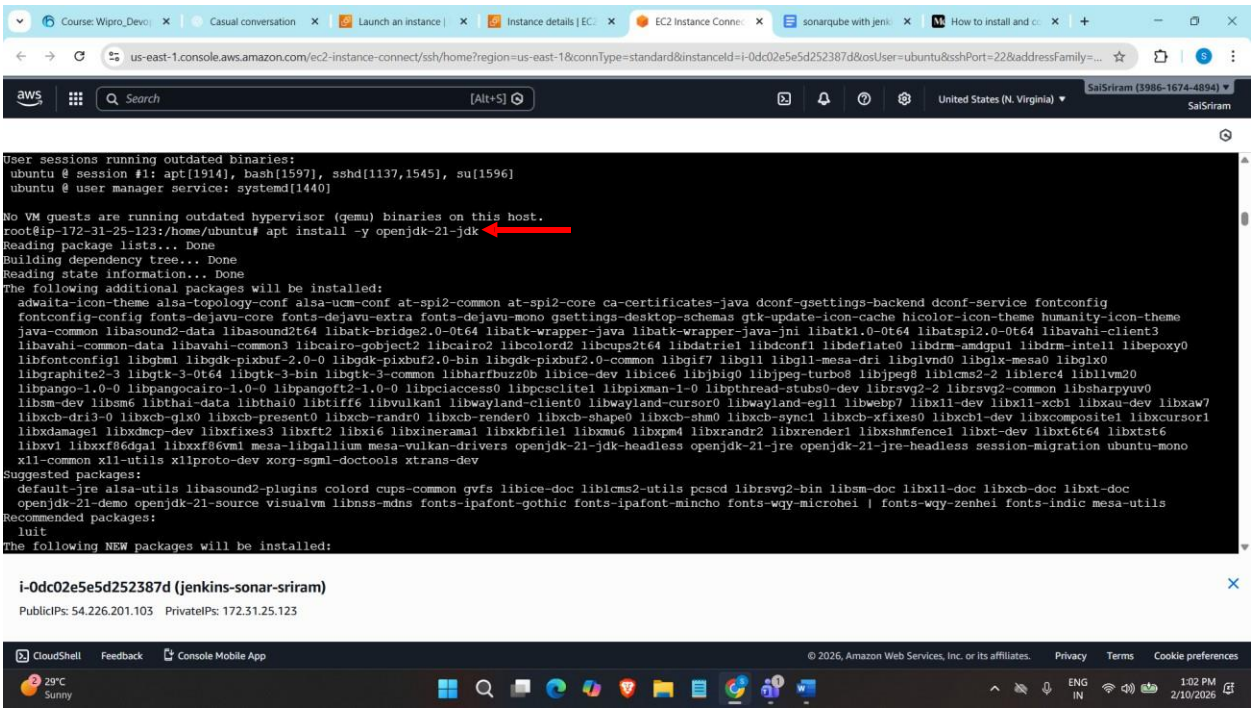


Use Command: `apt upgrade -y`



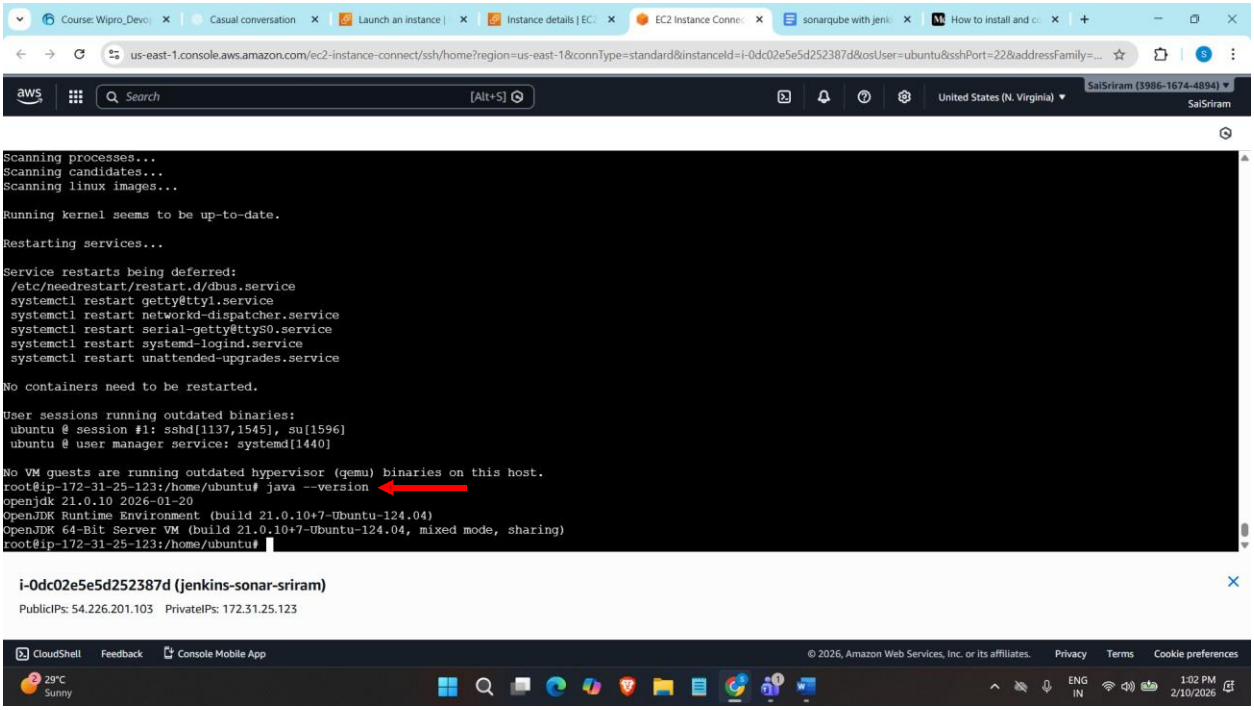
1. Install Java development kit
1. Install the latest OpenJDK 21

Use Command: `apt install -y openjdk-21-jdk`



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

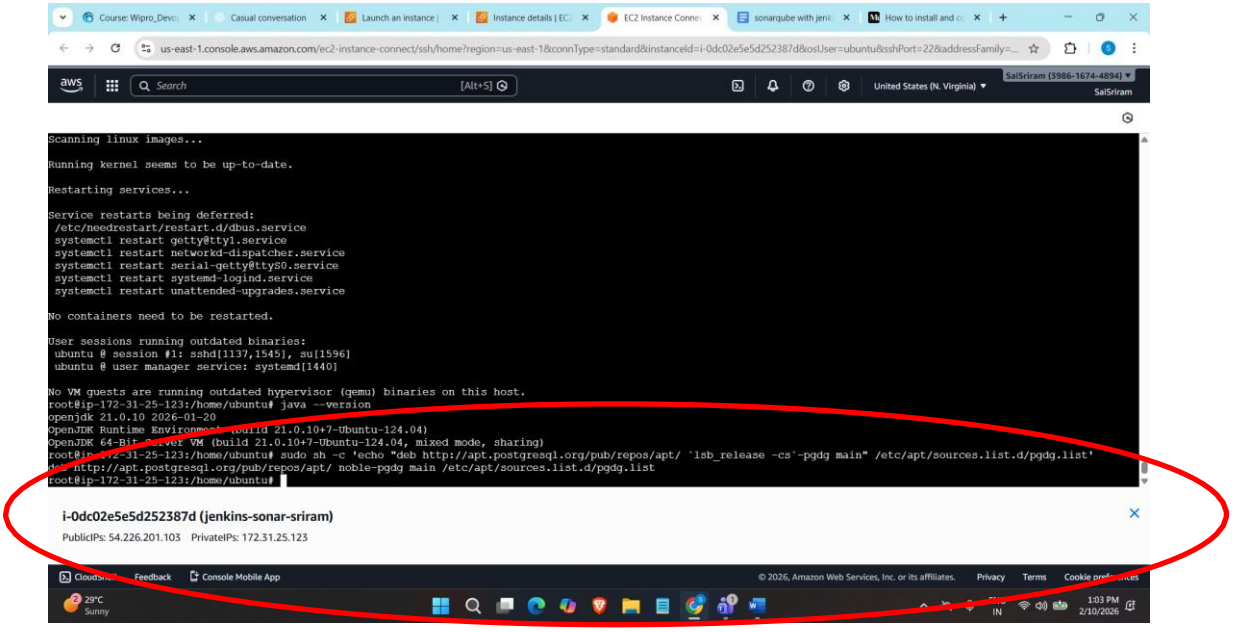
Command: `Java --version`



1. Install and Configure PostgreSQL

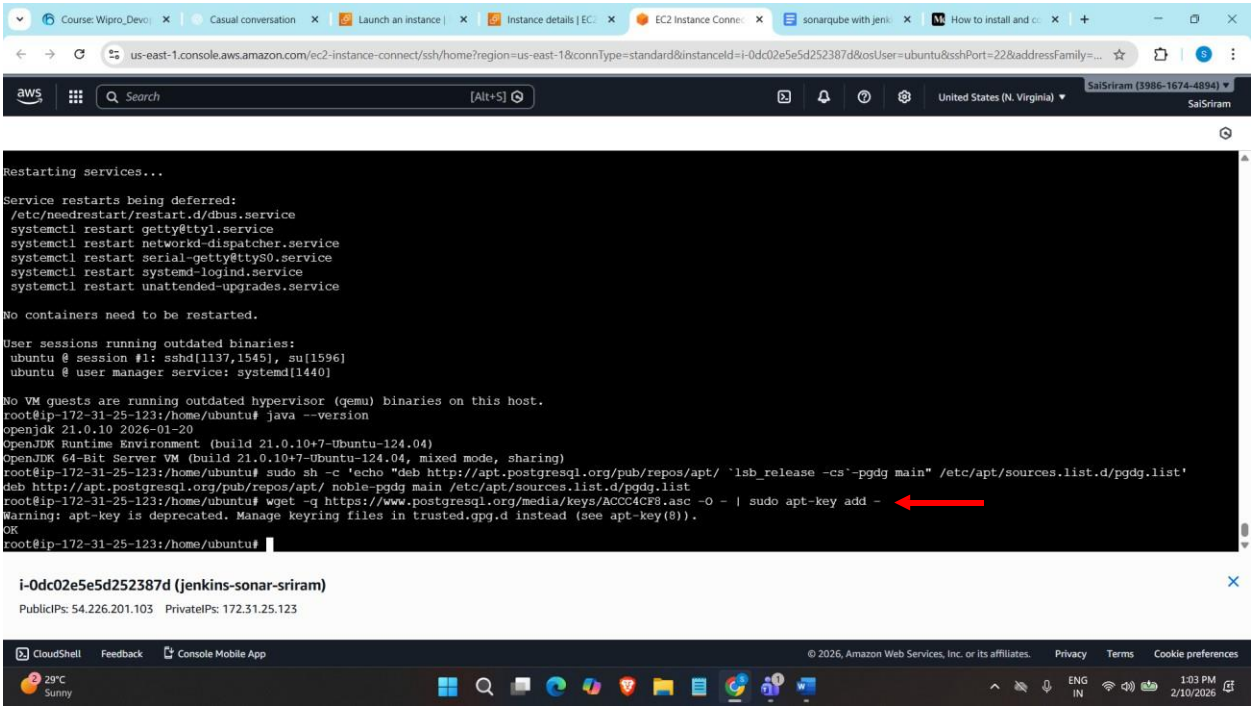
1. Add the PostgreSQL repository.

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



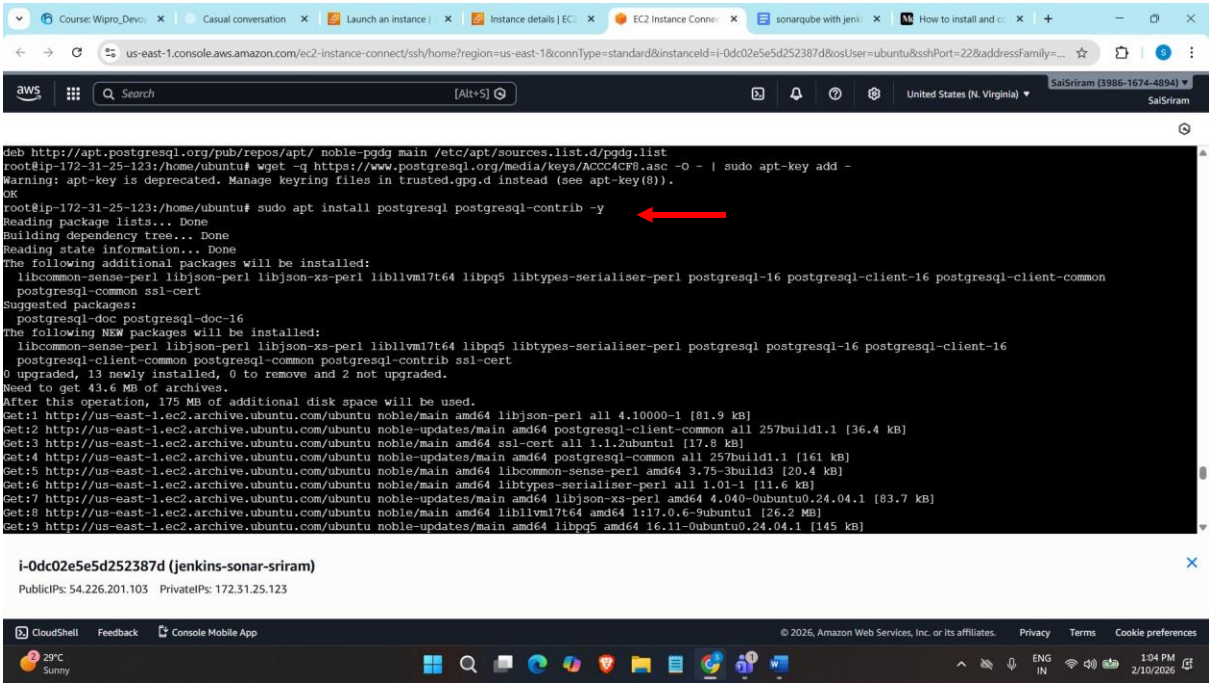
2. Add the PostgreSQL signing key.

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



3. Install PostgreSQL.

Use Command: `sudo apt install postgresql postgresql-contrib -y`

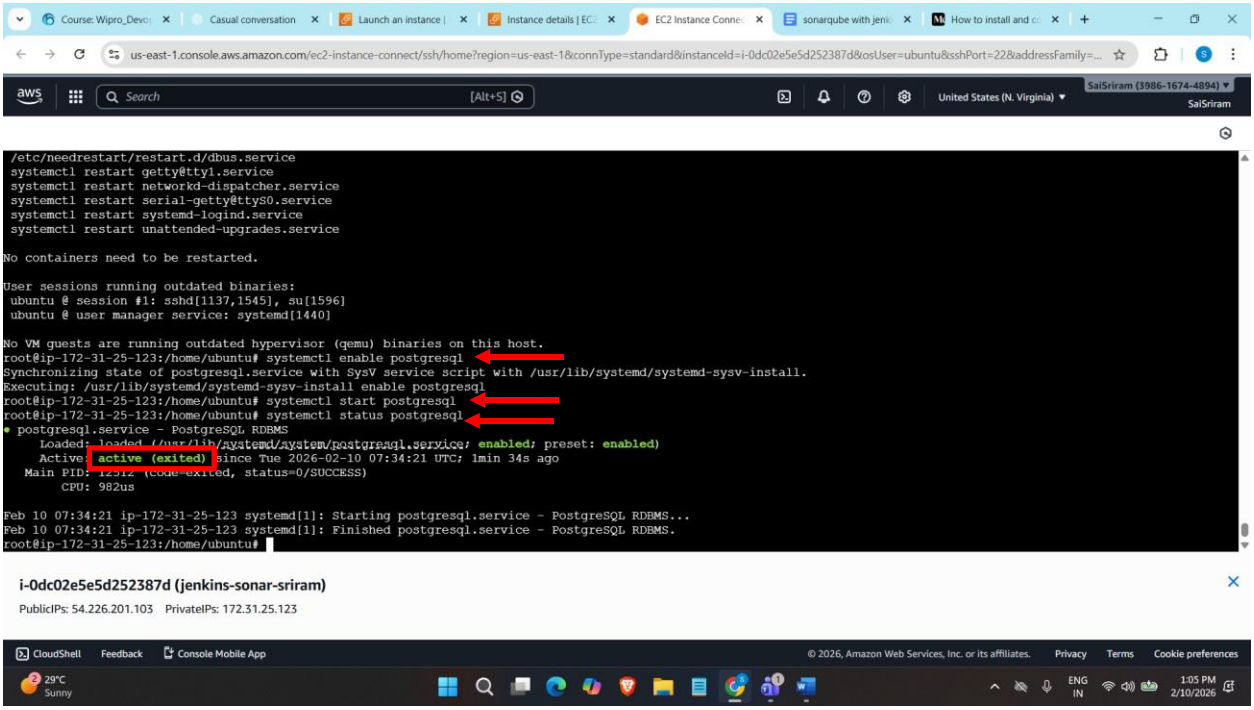


4. Enable the database server to start automatically on reboot, start the database, check status.

To enable use Command: `sudo systemctl enable postgresql`

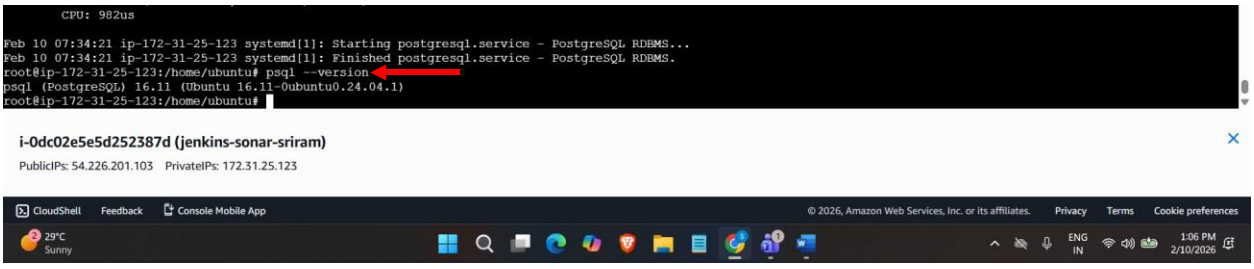
To start use Command: `sudo systemctl start postgresql`

To check status: `sudo systemctl status postgresql`



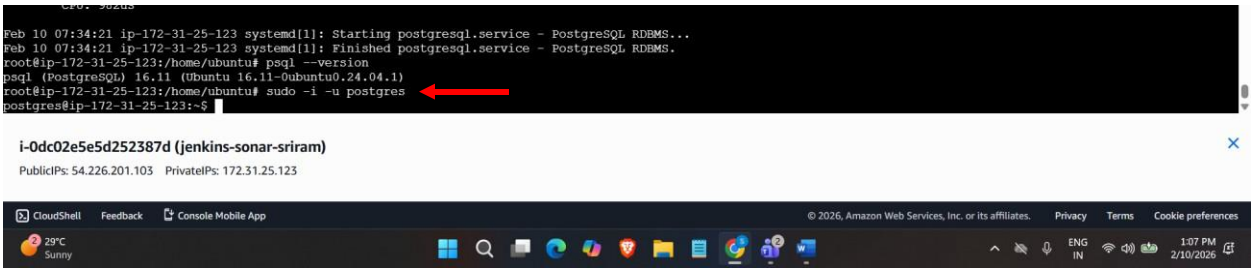
5. Let's check the installed version of the install Postgres DB. VALIDATION IS IMPORTANT. Use

Command: `psql --version`



6. Switch to the Postgres user.

To Switch use Command: `sudo -i -u postgres`

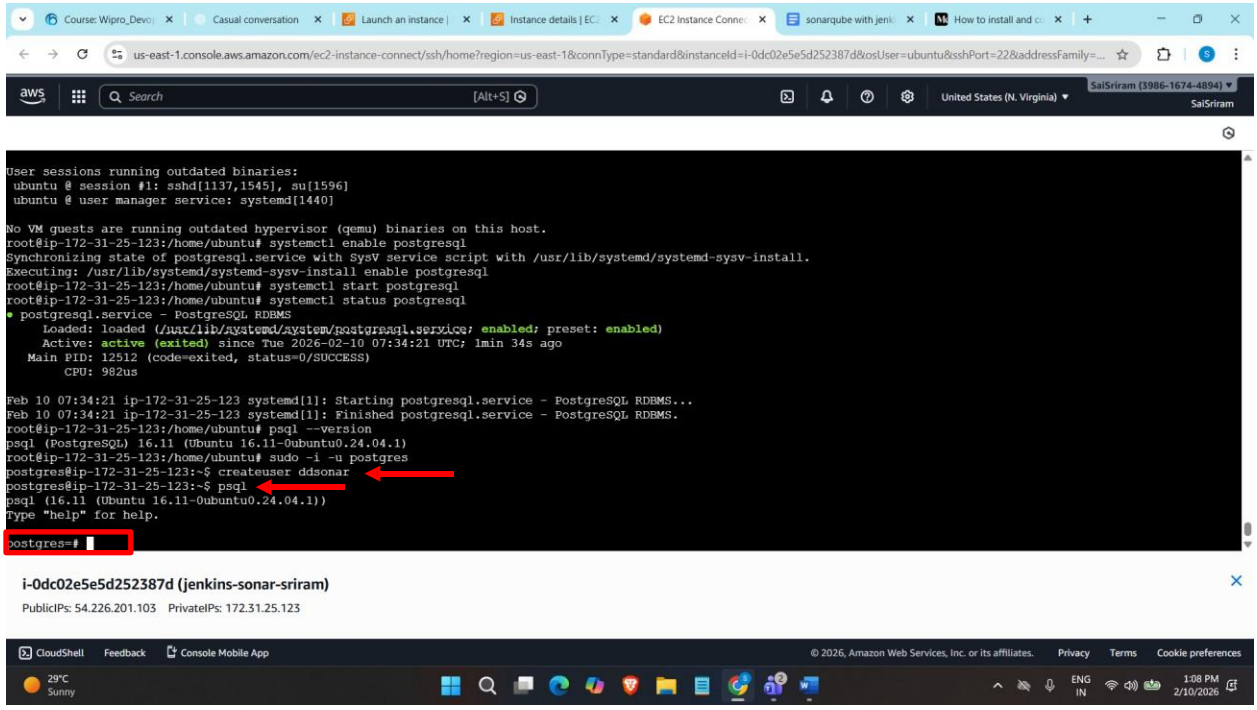


7. Create a database user named ddsonar and login to postgreSQL

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.

To Create database user use Command: `createuser ddsonar`

To login to PostgreSQL use Command: `psql`



8. Set a password for the ddsonar user, Create a SonarQube database and set the owner to ddsonar, Grant all the privileges on the ddsonarqube database to the ddsonar user.

1. **To set password:**

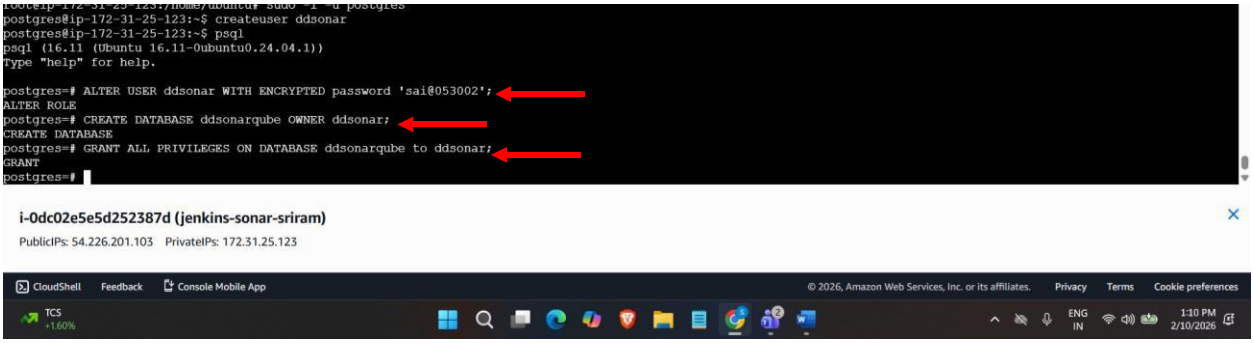
```
ALTER USER ddsonar WITH ENCRYPTED password 'sai@123456';
```

2. **To create database:**

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

3. **To grant all privileges:**

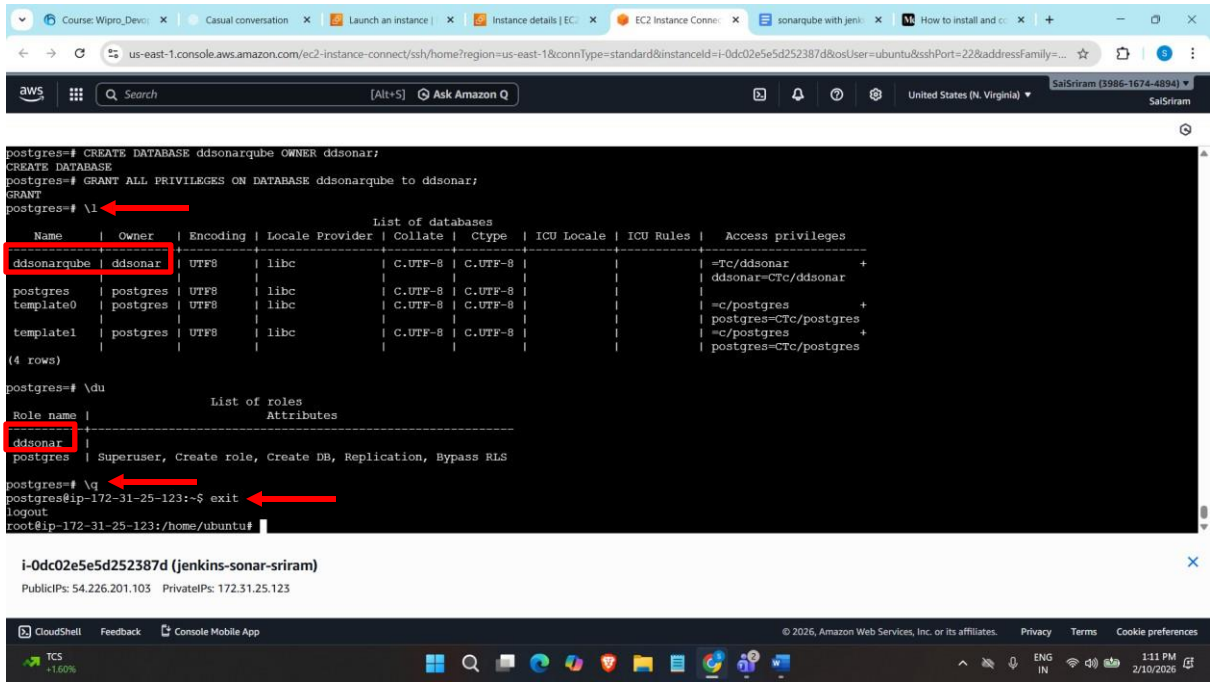
```
GRANT ALL PRIVILEGES ON DATABASE ddsonarqube to ddsonar;
```



9. Let's check the created user and the database.

1. **To check created database use command:** `\l`, **to check the created database user:** `\du`,

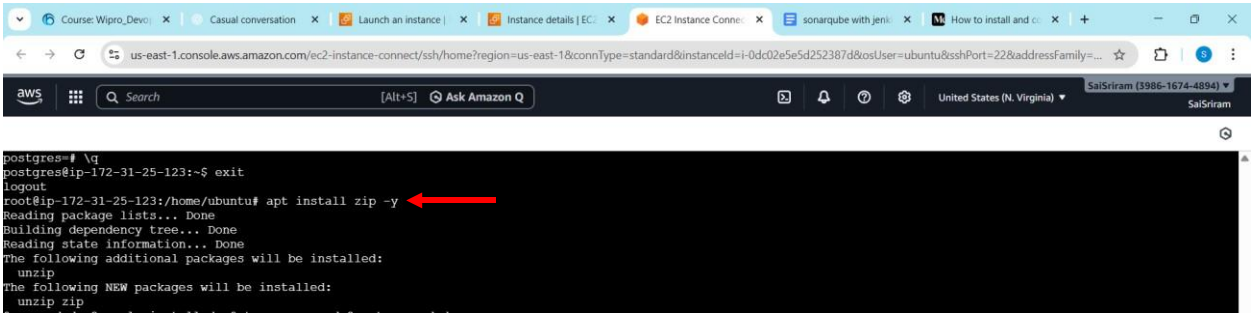
to exit postgresql: `q`, **to return non root sudo user:** `exit`



2. Download and Install SonarQube

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

Use command: `apt install zip -y`



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

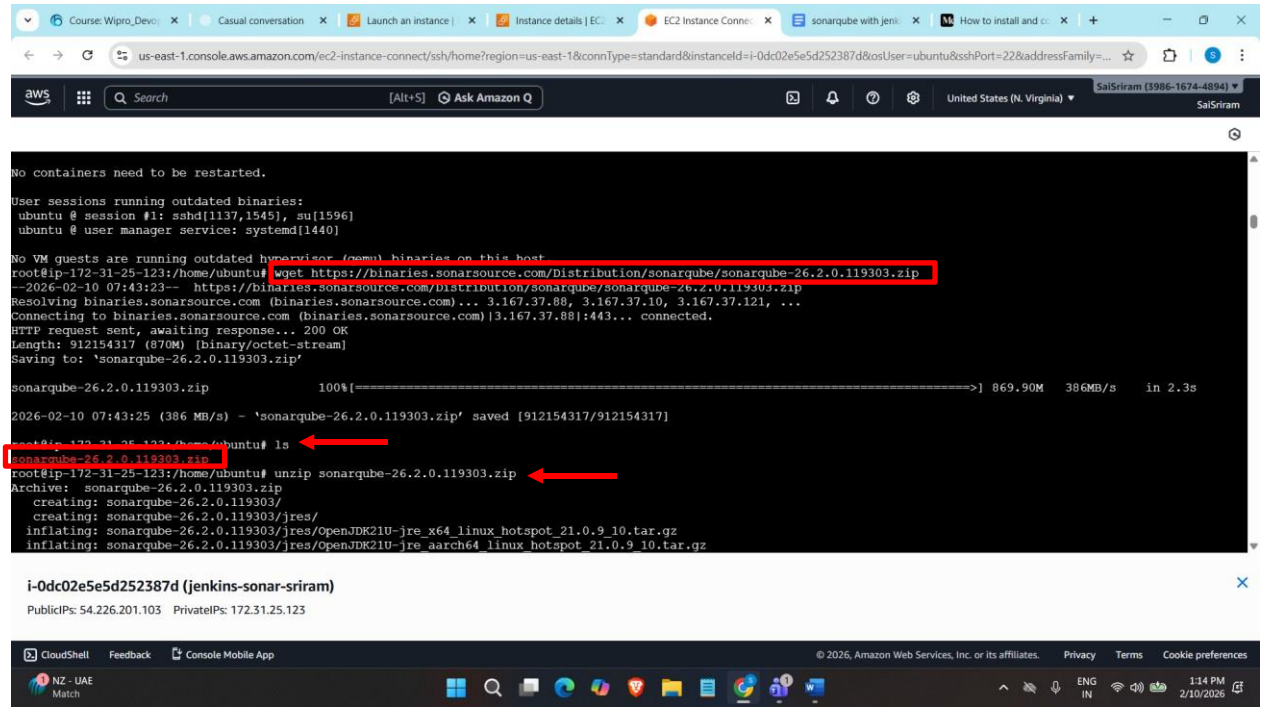
URL: https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-26.2.0.119303.zip

Use command:

`wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-26.2.0.119303.zip`

to List use command: `ls`

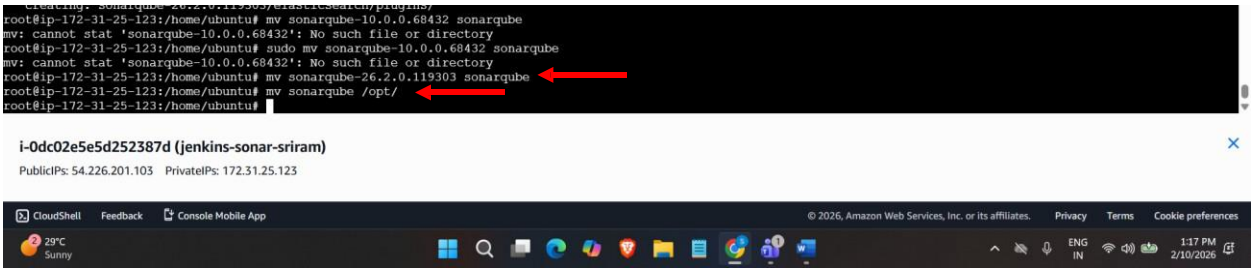
to unzip use command: `unzip sonarqube-26.2.0.11903`



3. Move the unzipped files to /opt/sonarqube directory

Use Command: `sudo mv sonarqube-26.2.0.11903 sonarqube`

Use command: `sudo mv sonarqube /opt/`



3. Add SonarQube Group and User

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

1. Create a sonar group

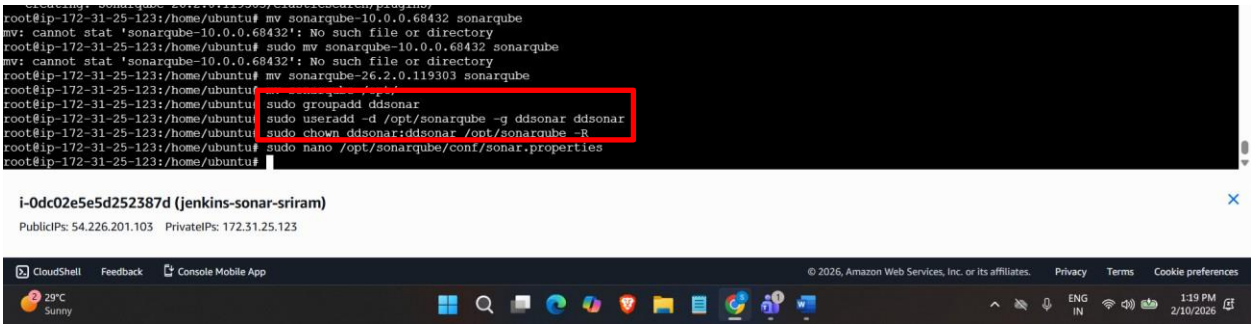
`sudo groupadd ddsonar`

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

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

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

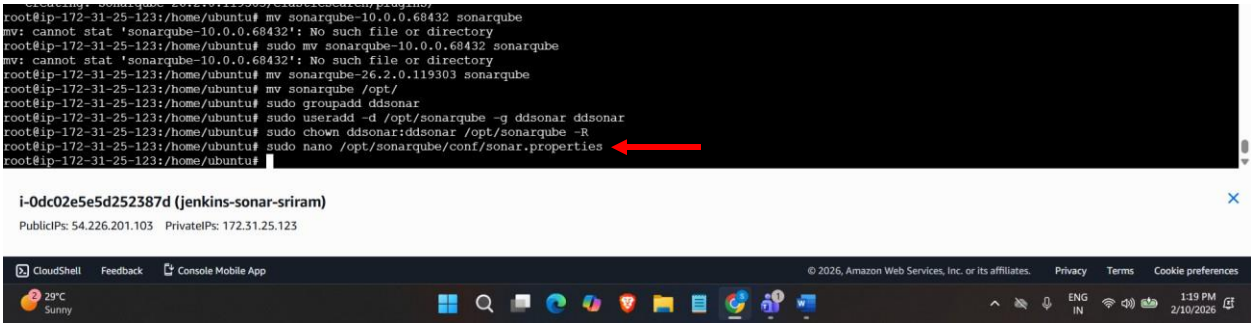
`sudo chown ddsonar:ddsonar /opt/sonarqube -R`



4. Configure SonarQube:

1. Edit the SonarQube configuration file use command:

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



1. Find the following lines:

#sonar.jdbc.username=

#sonar.jdbc.password=

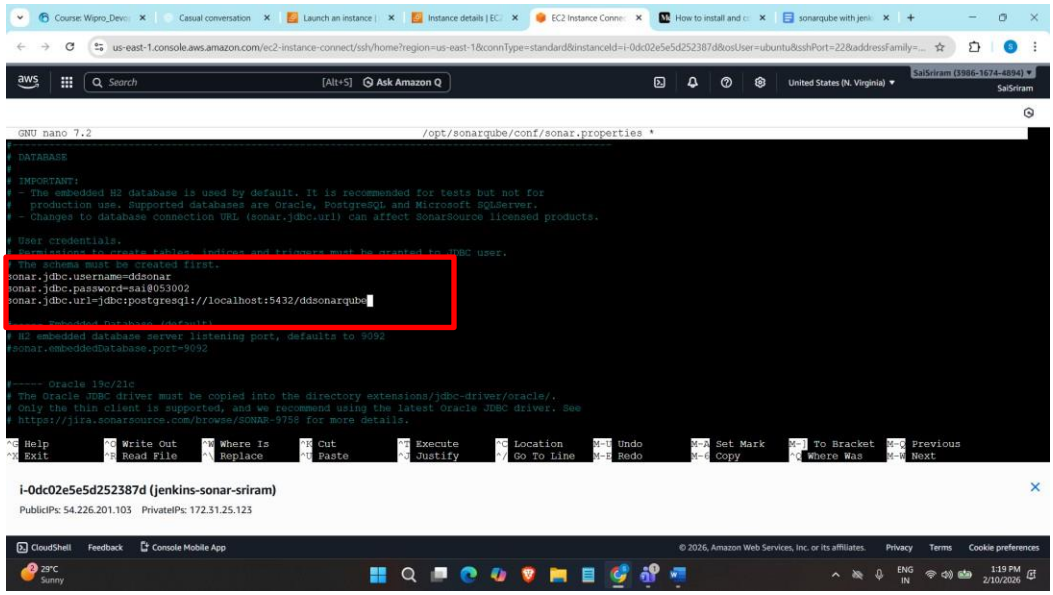
2. Uncomment the lines, and add the database user and Database password you created in Step 4 (xi and xii). For me, it's:

sonar.jdbc.username=ddsonar

sonar.jdbc.password=sai@123456

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

sonar.jdbc.url=jdbc:postgresql://localhost:5432/ddsonarqube **Save and exit the file.**



1. Edit the sonar script file.

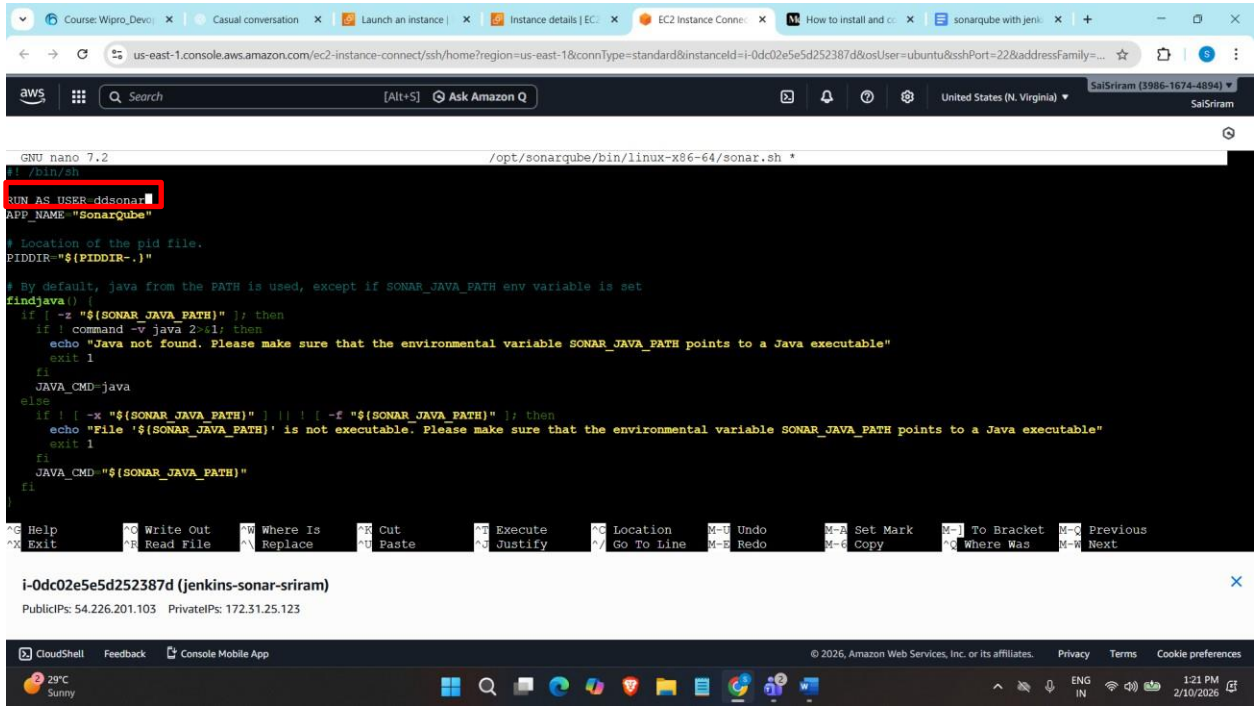
```
sudo nano /opt/sonarqube/bin/linux-x86-64/sonar.sh
```

1. 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).

2. Save and exit the file



3. Setup Systemd service

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

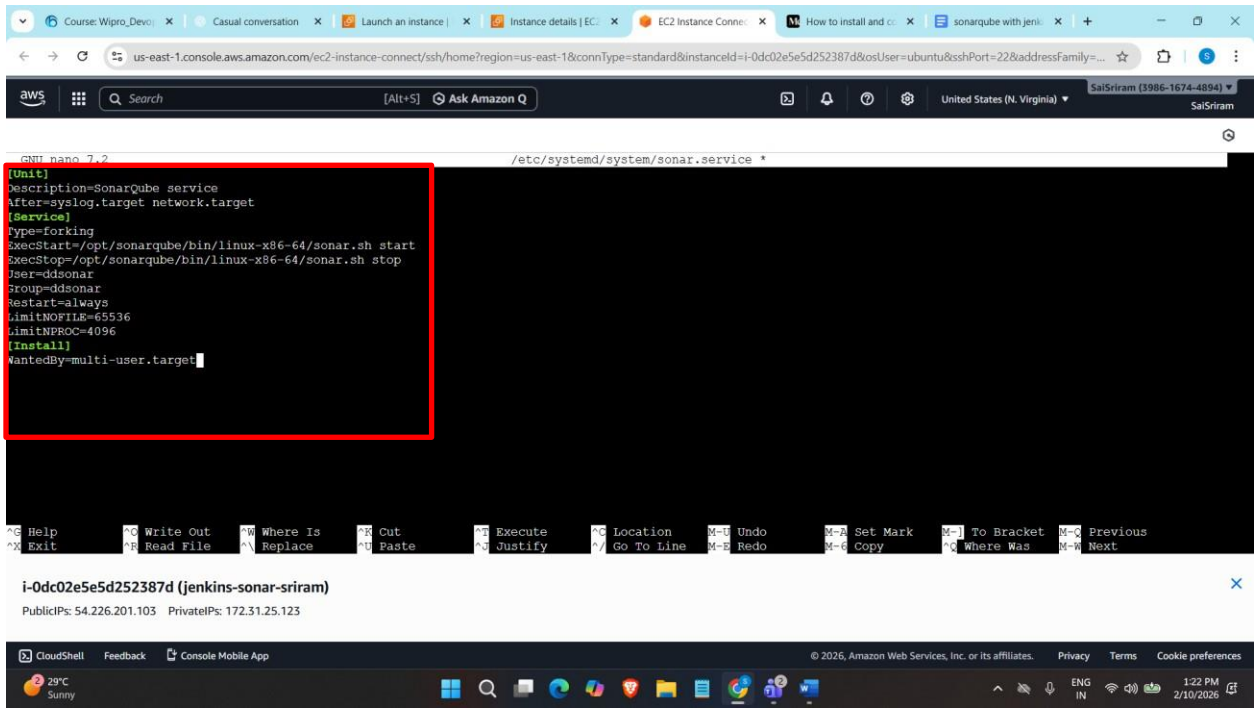
```
sudo nano /etc/systemd/system/sonar.service
```

2. 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
```

3. Save and exit the file



4. Enable the SonarQube service to run at system startup.

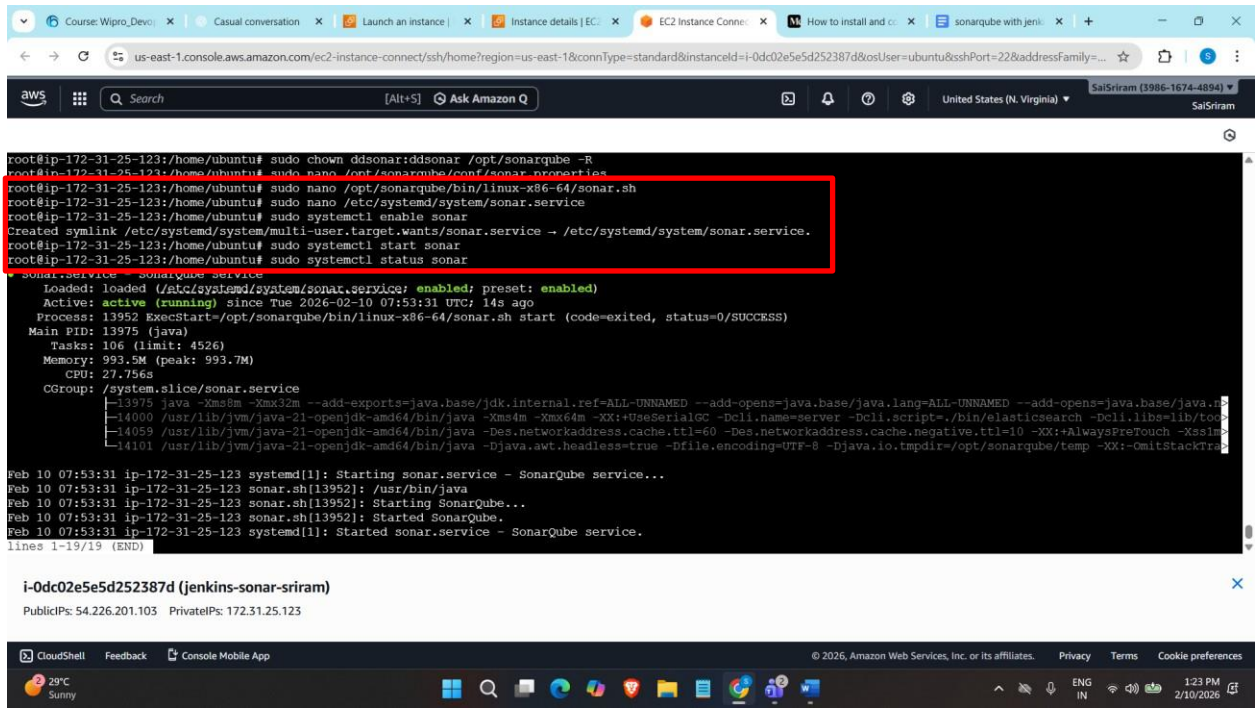
```
sudo systemctl enable sonar
```

5. Start the SonarQube service.

```
sudo systemctl start sonar
```

6. Check the service status.

```
sudo systemctl status sonar
```

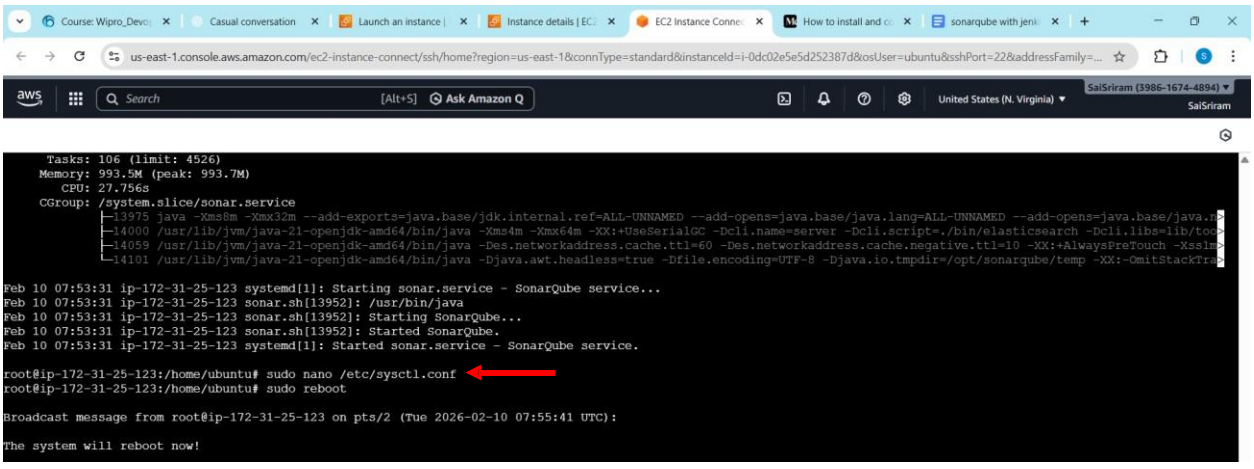


4. Modify Kernel System Limits

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

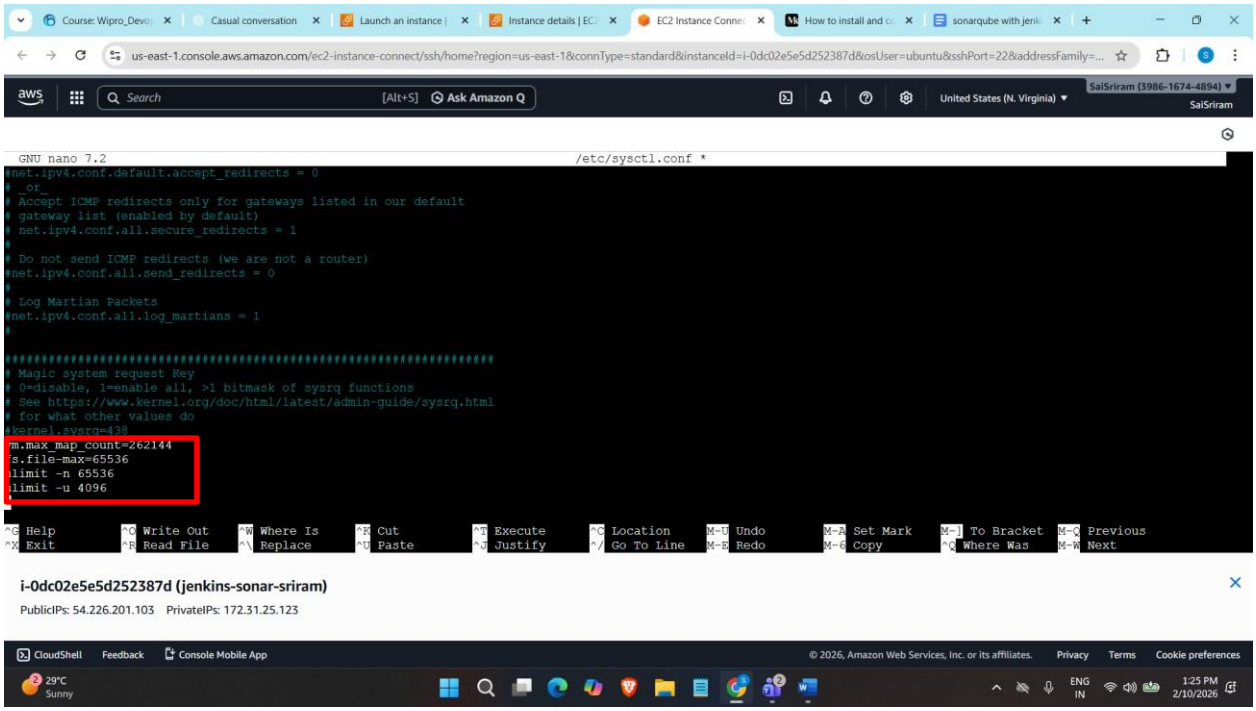
1. Edit the sysctl configuration file.

```
sudo nano /etc/sysctl.conf
```



2. Add the following lines.

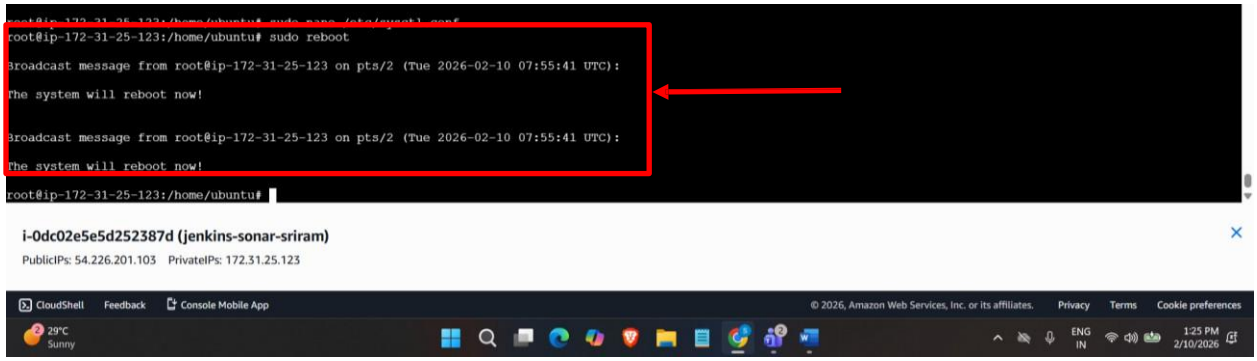
```
vm.max_map_count=262144

fs.file-max=65536

ulimit -n 65536

ulimit -u 4096
```



3. Save and exit the file

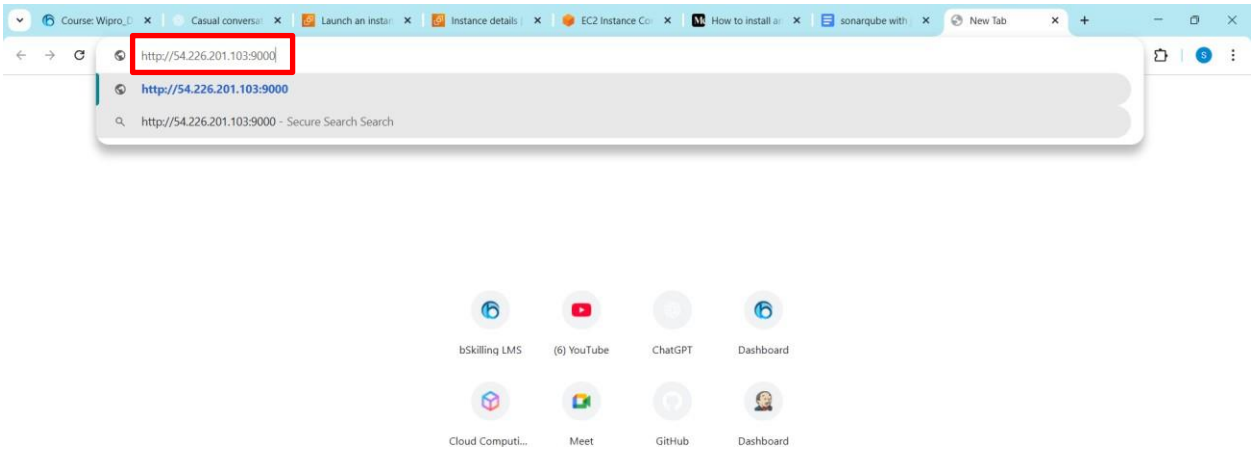
4. Reboot the system to apply the changes.

```
sudo reboot
```

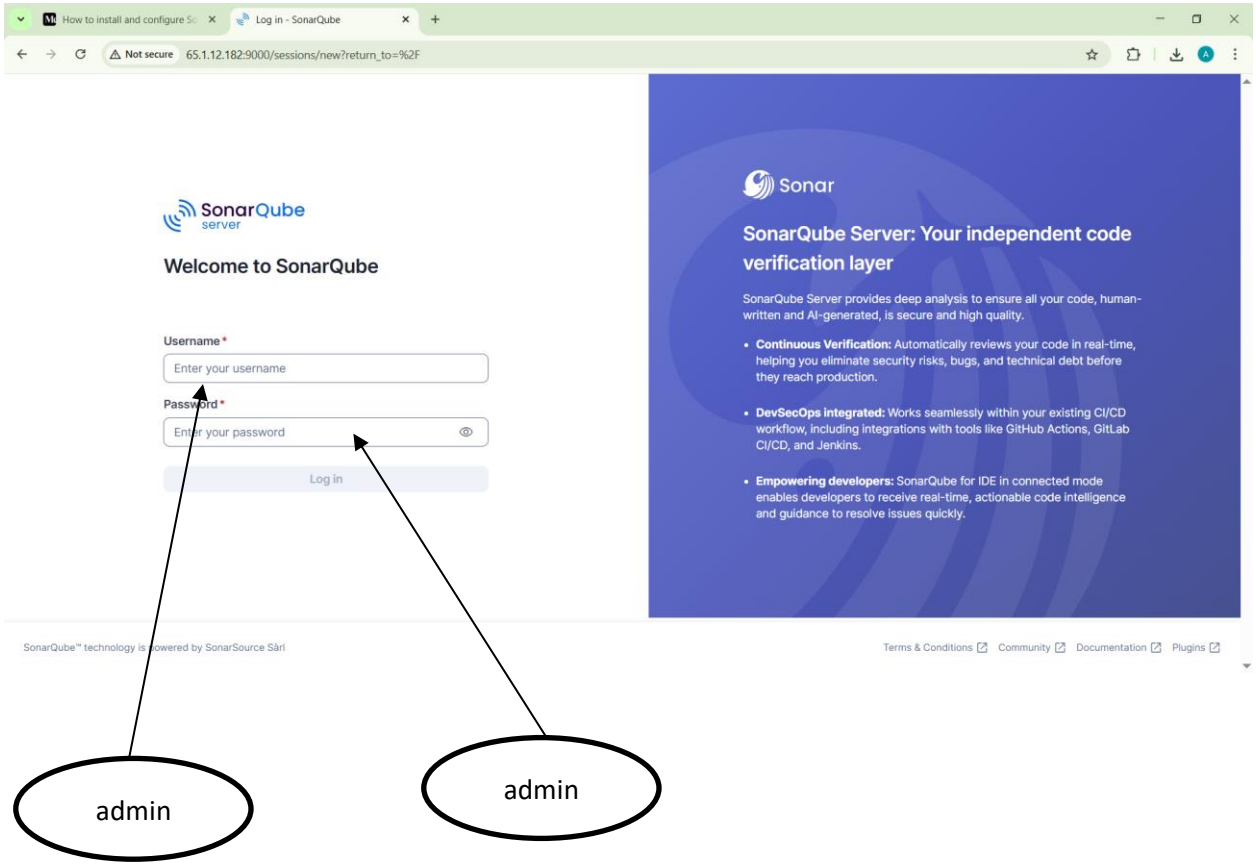


SonarQube Tool and configuration With Jenkins

1. Open the browser and navigate to **http://public:9000**
2. Log in to sonarqube server with System Administrator credentials (admin/admin)

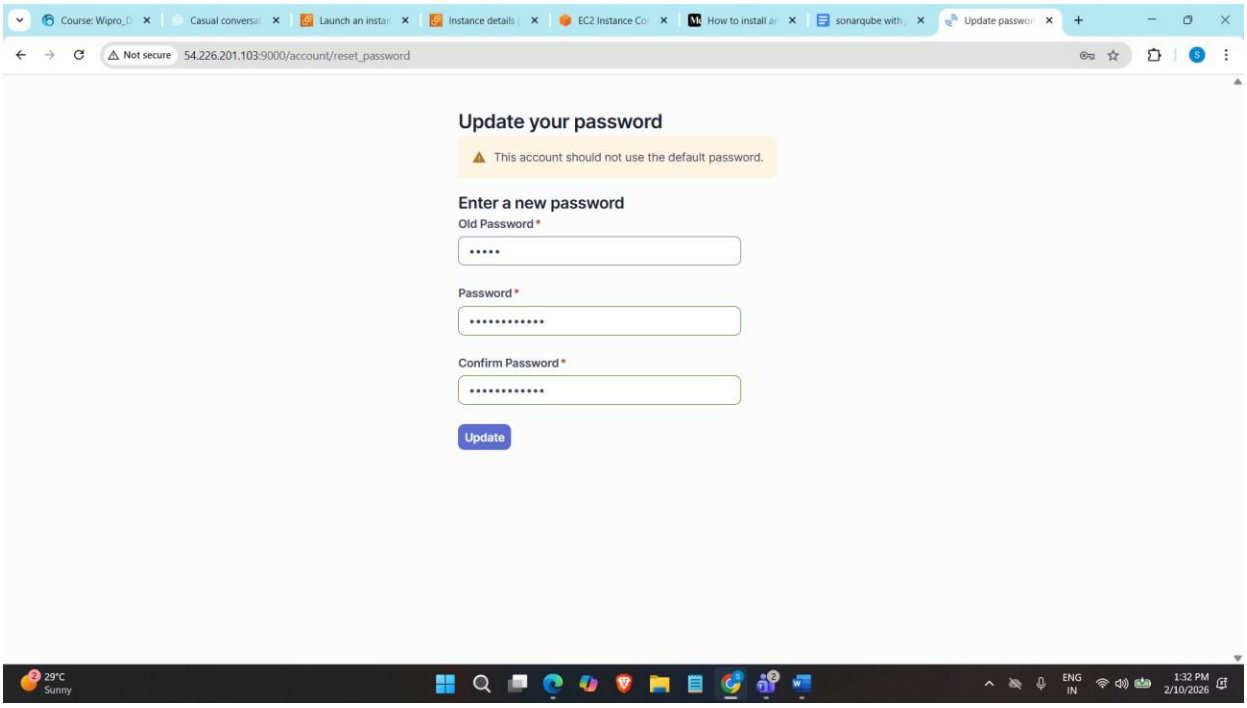


- **Note: the default username and password are admin and admin respectively.**

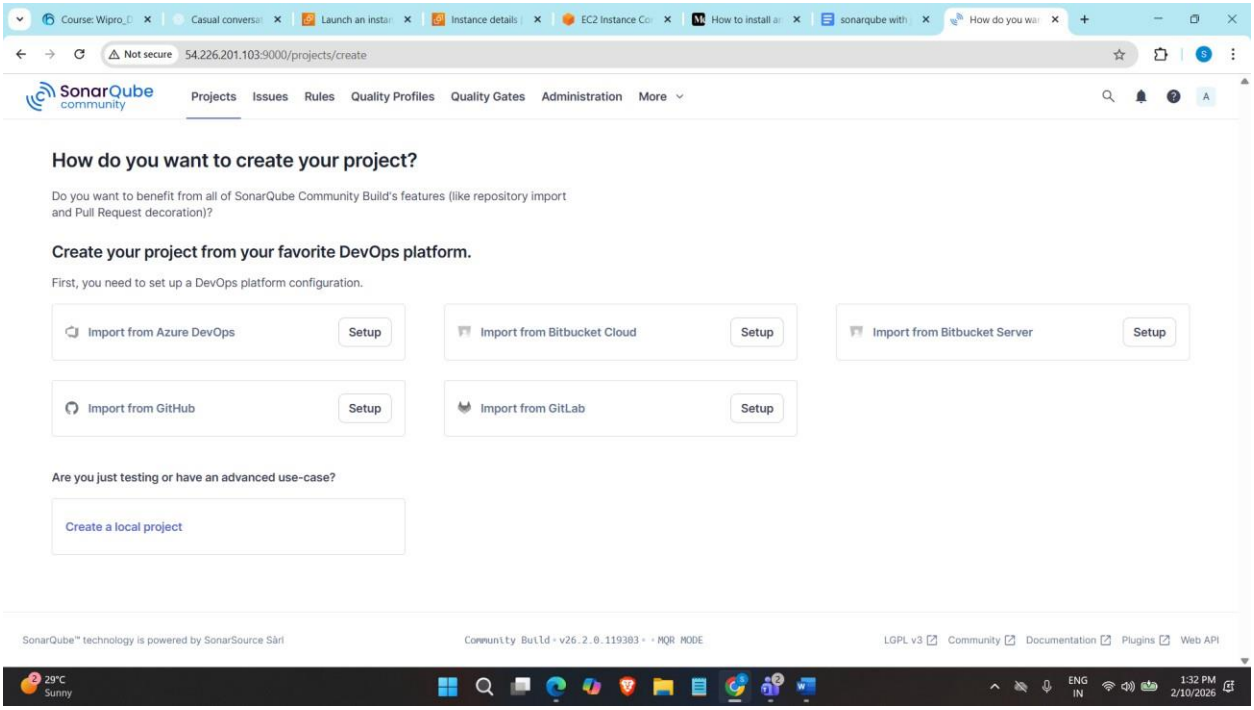


1. Change the Old password with a New one.

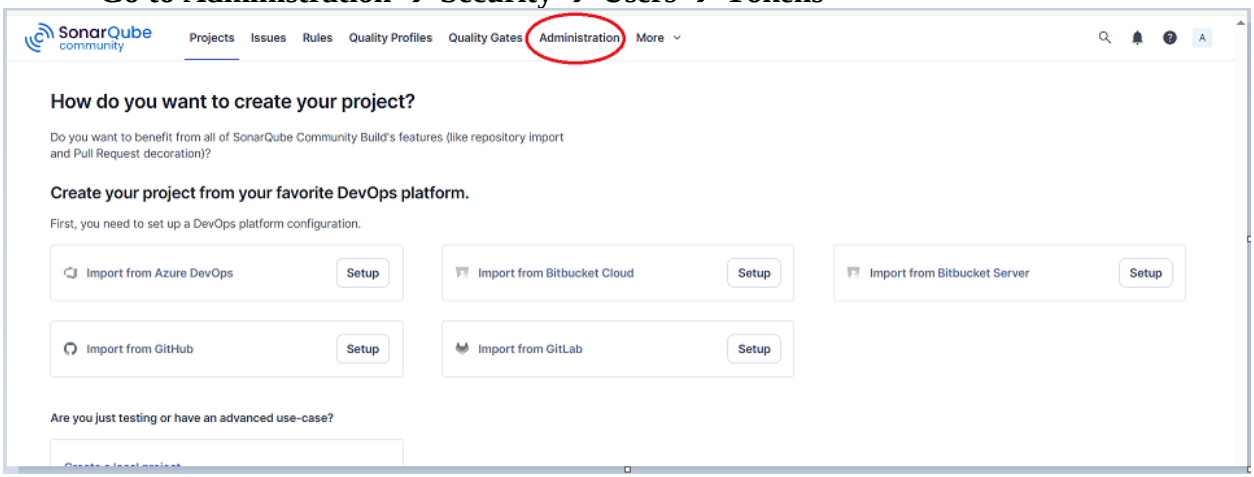
Log in with username admin and password admin. In the next step, SonarQube will prompt you to change your password. CHANGE THE PASSWORD.

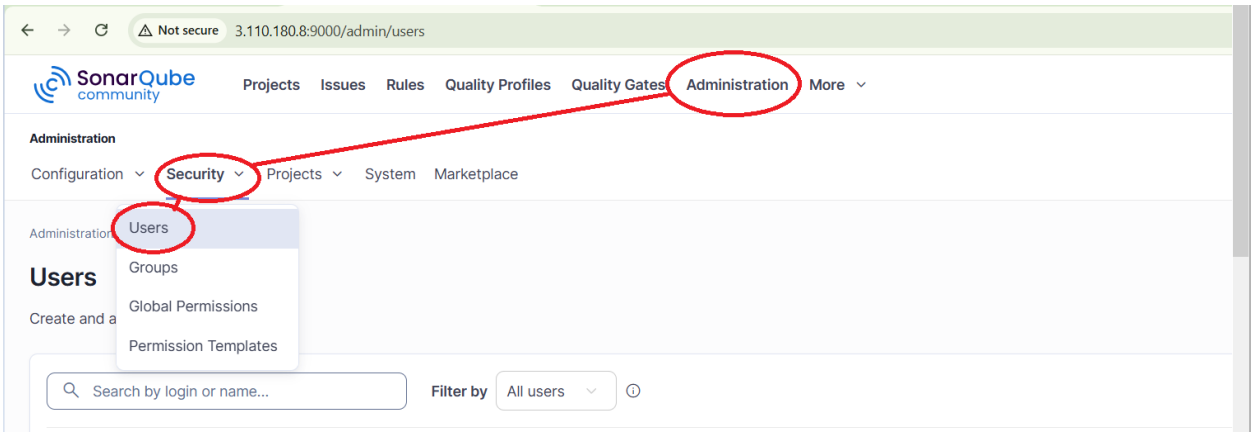


- **Finally landing page of SonarQube:**

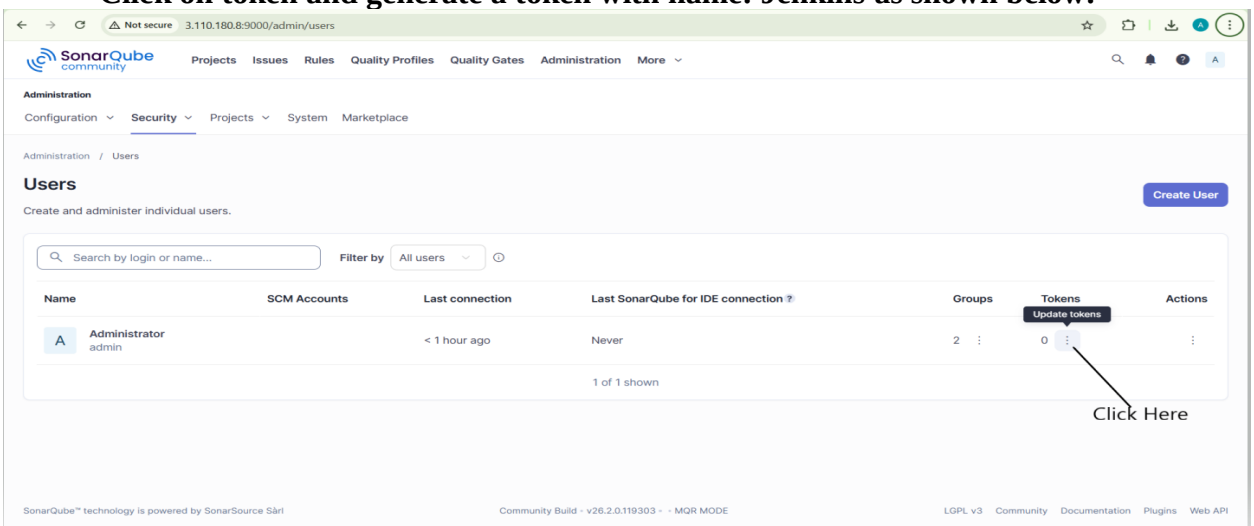


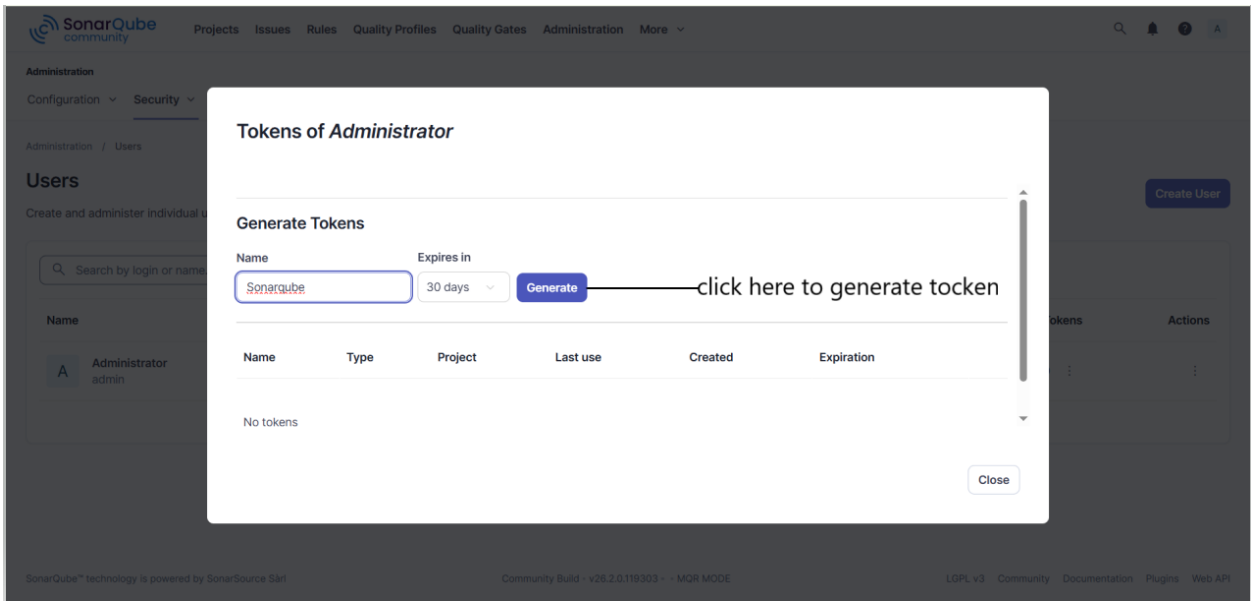
- **Go to Administration → Security → Users → Tokens**



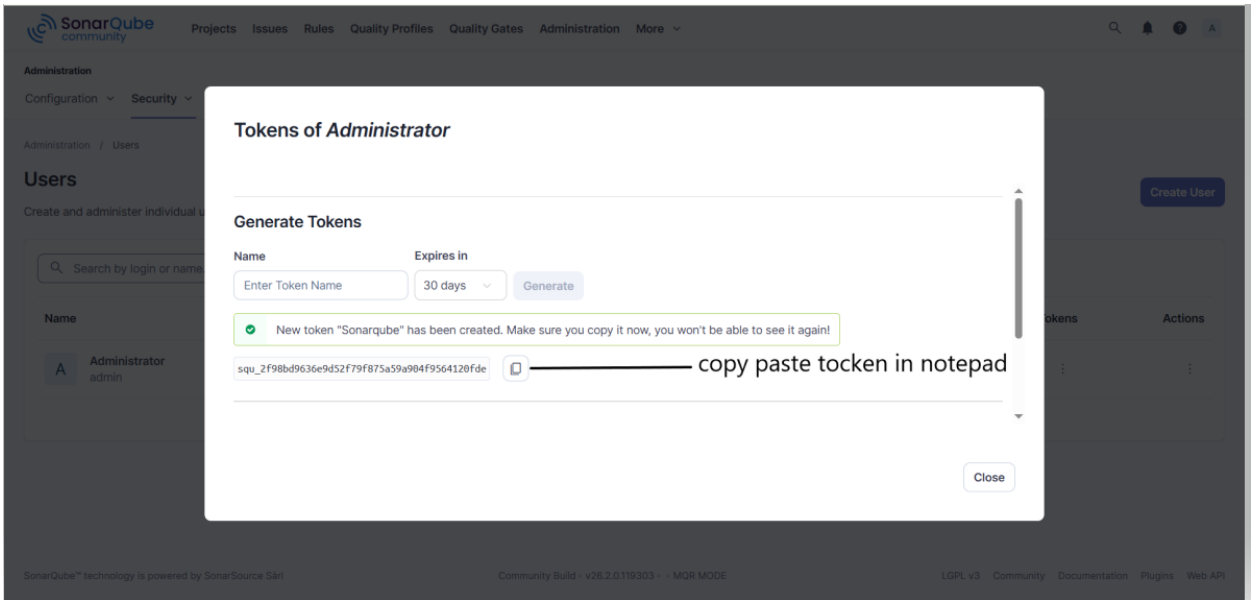


- **Click on token and generate a token with name: Jenkins as shown below:**



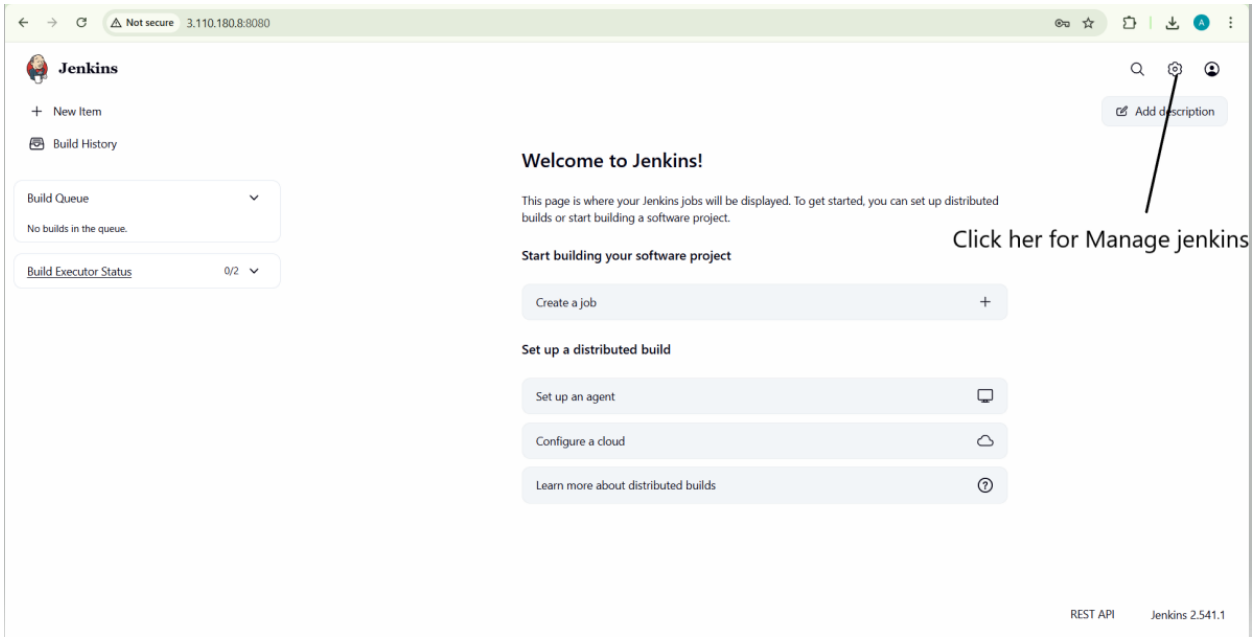


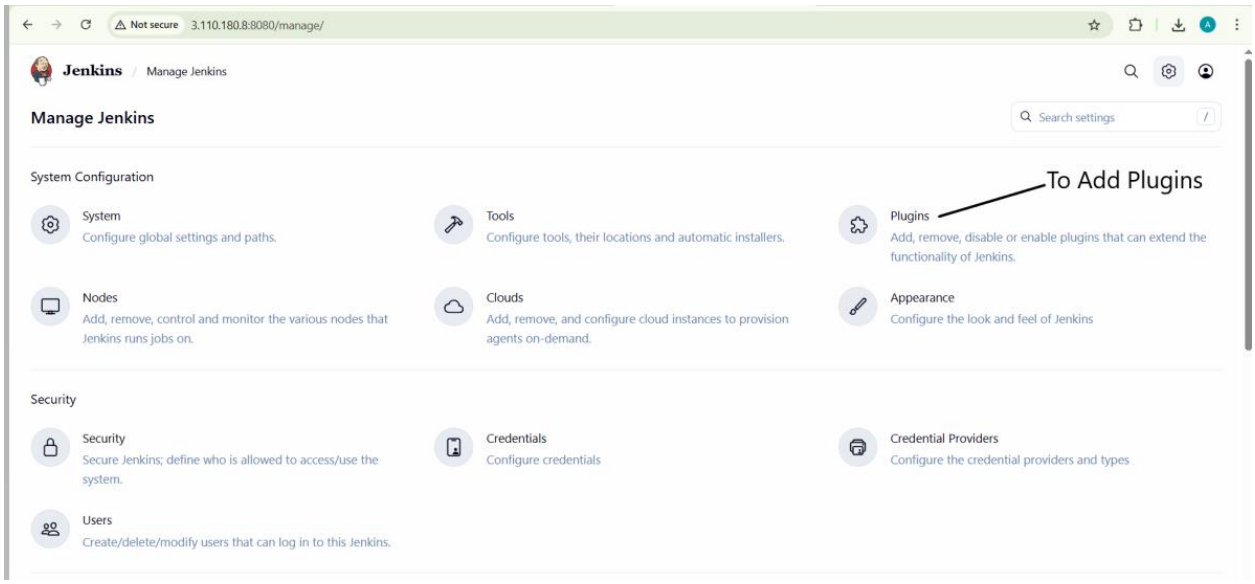
- **Copy the generated token and note it down. It will be used in Jenkins for Sonar authentication**

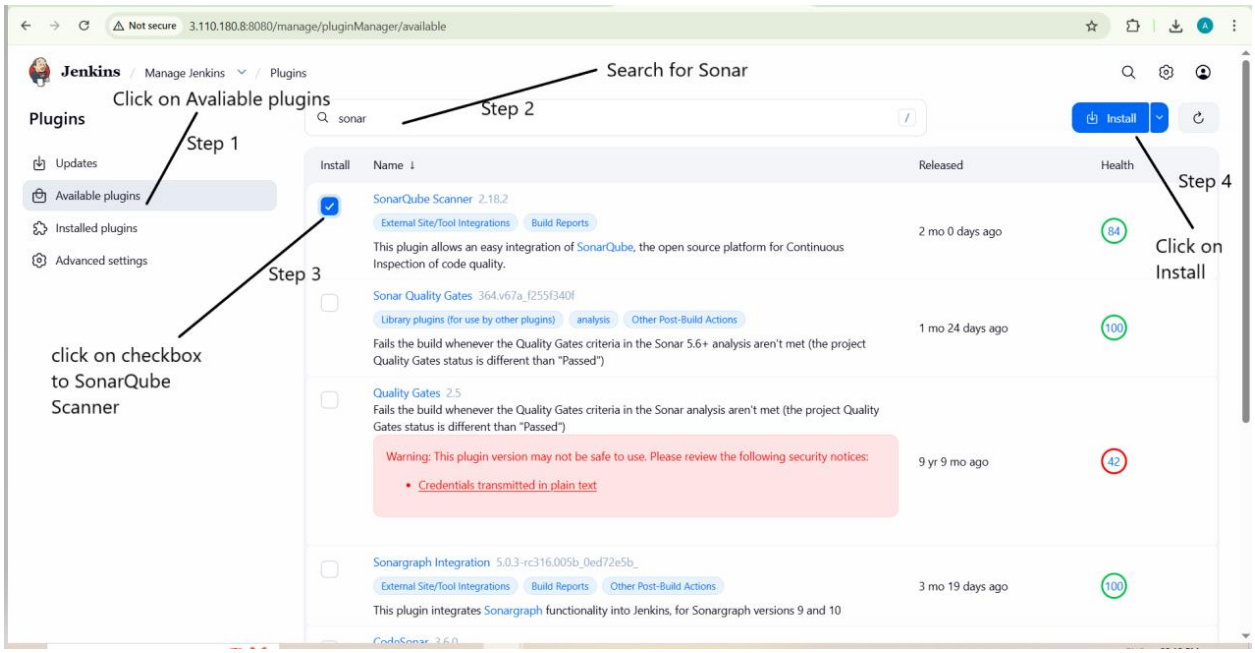


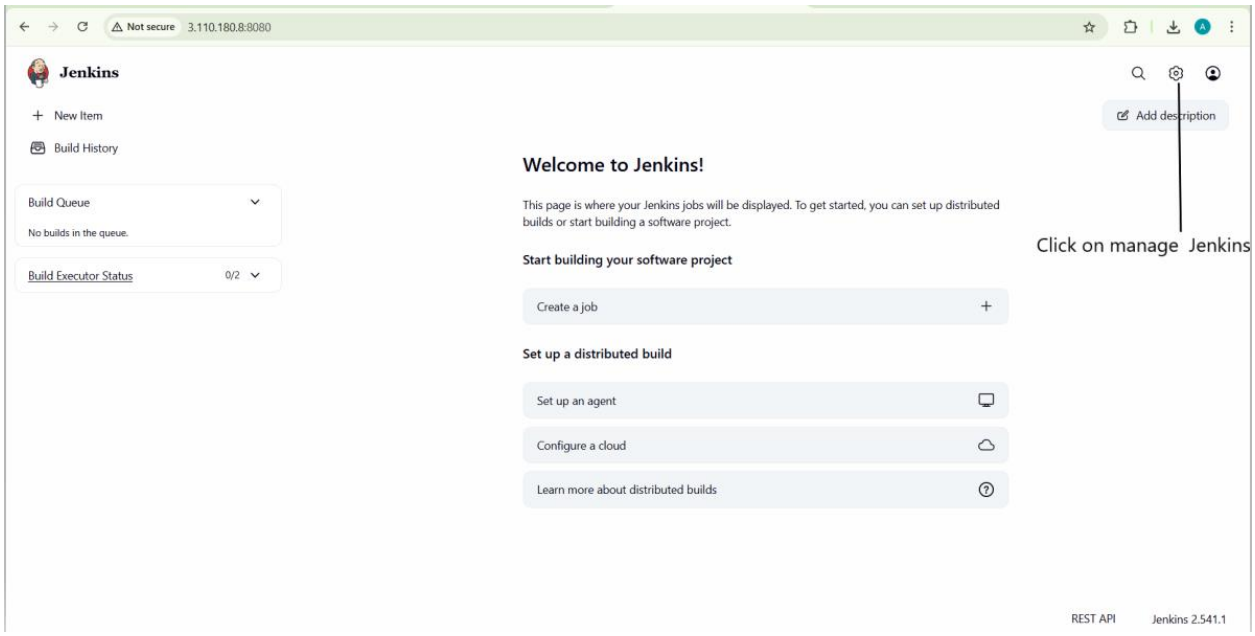
Step 2: Installing and configuring SonarQube plugin in Jenk

- **Go to Manage Jenkins → Manage Plugins → Available → search for SonarQube Scanner→ Click on install without restart**

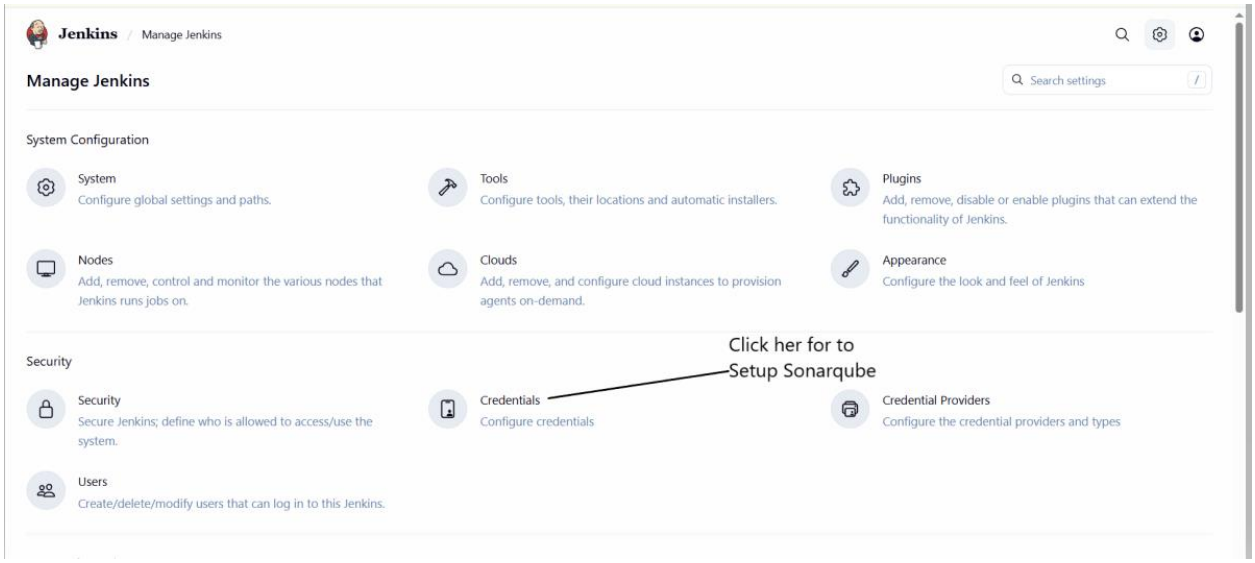




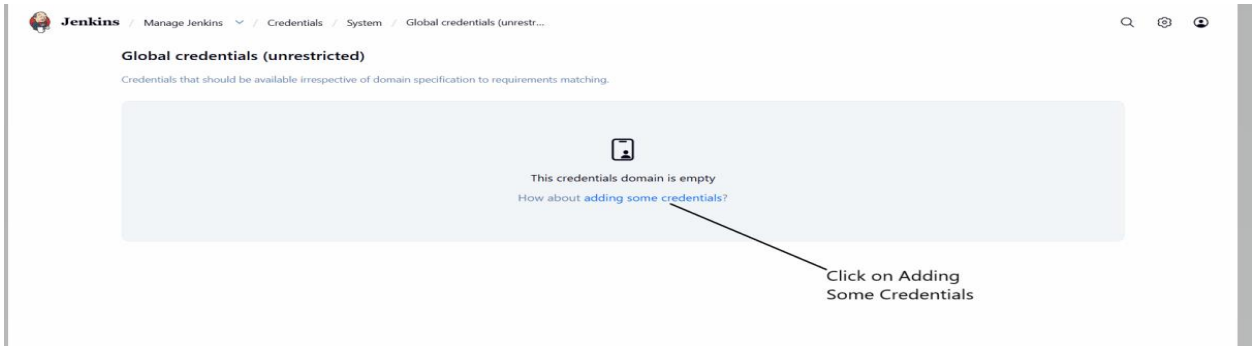

Not secure 3.110.180.8:8080/manage/pluginManager/available
Jenkins / Manage Jenkins / Plugins
Click on Avaliable plugins
Search for Sonar
Plugins
Step 1
Step 2
sonar
Install
Step 4
Updates
Available plugins
Installed plugins
Advanced settings
Install
Name
Released
Health
Step 3
Click on Install
SonarQube Scanner 2.18.2
External Site/Tool Integrations
Build Reports
This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.
2 mo 0 days ago
84
Sonar Quality Gates 364.v67a_f255f340f
Library plugins (for use by other plugins)
analysis
Other Post-Build Actions
Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")
1 mo 24 days ago
100
click on checkbox to SonarQube Scanner
Quality Gates 2.5
Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed")
Warning: This plugin version may not be safe to use. Please review the following security notices:
Credentials transmitted in plain text
9 yr 9 mo ago
42
Sonargraph Integration 5.0.3-rc316.005b_0ed72e5b_
External Site/Tool Integrations
Build Reports
Other Post-Build Actions
This plugin integrates Sonargraph functionality into Jenkins, for Sonargraph versions 9 and 10
3 mo 19 days ago
100


Not secure 3.110.180.8:8080
Jenkins
New Item
Build History
Add description
Build Queue
No builds in the queue.
Welcome to Jenkins!
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.
Click on manage Jenkins
Build Executor Status 0/2
Start building your software project
Create a job
Set up a distributed build
Set up an agent
Configure a cloud
Learn more about distributed builds
REST API
Jenkins 2.541.1

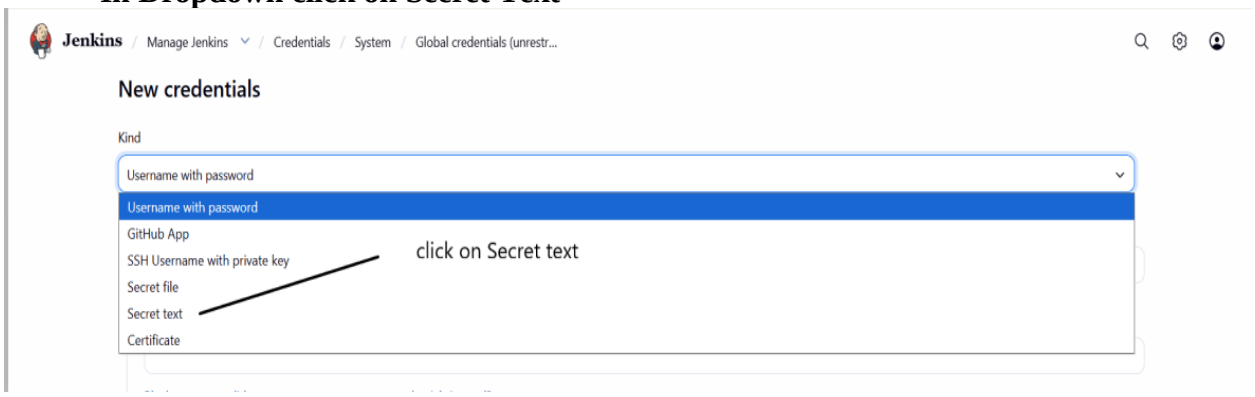
- **Go to Jenkins dashboard → Manage Jenkins → Manage Credentials**



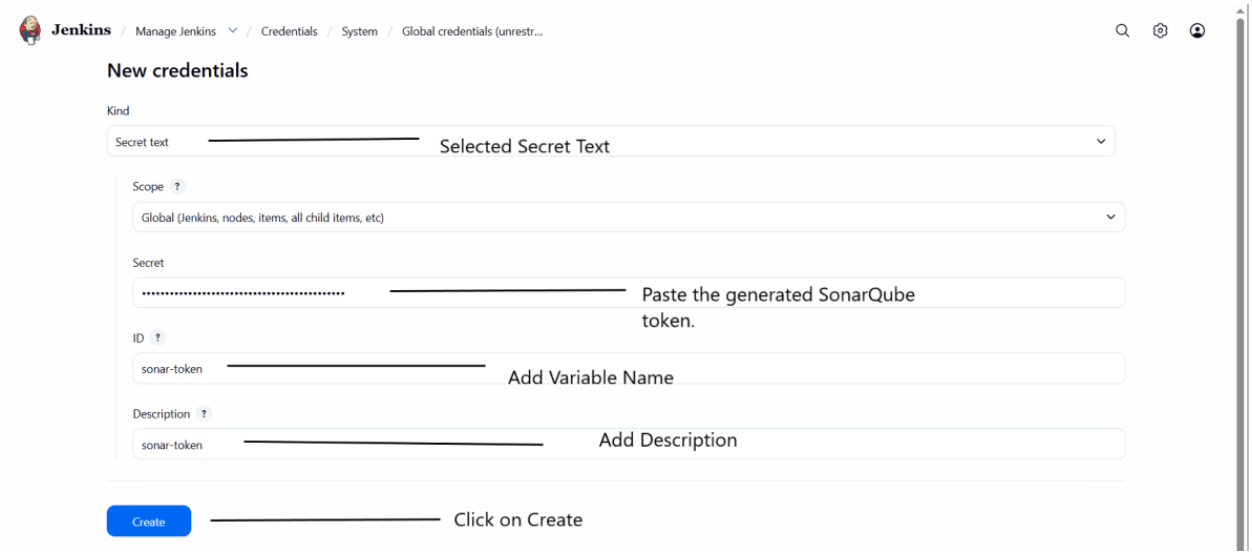
- **Click on Adding Some Credentials**



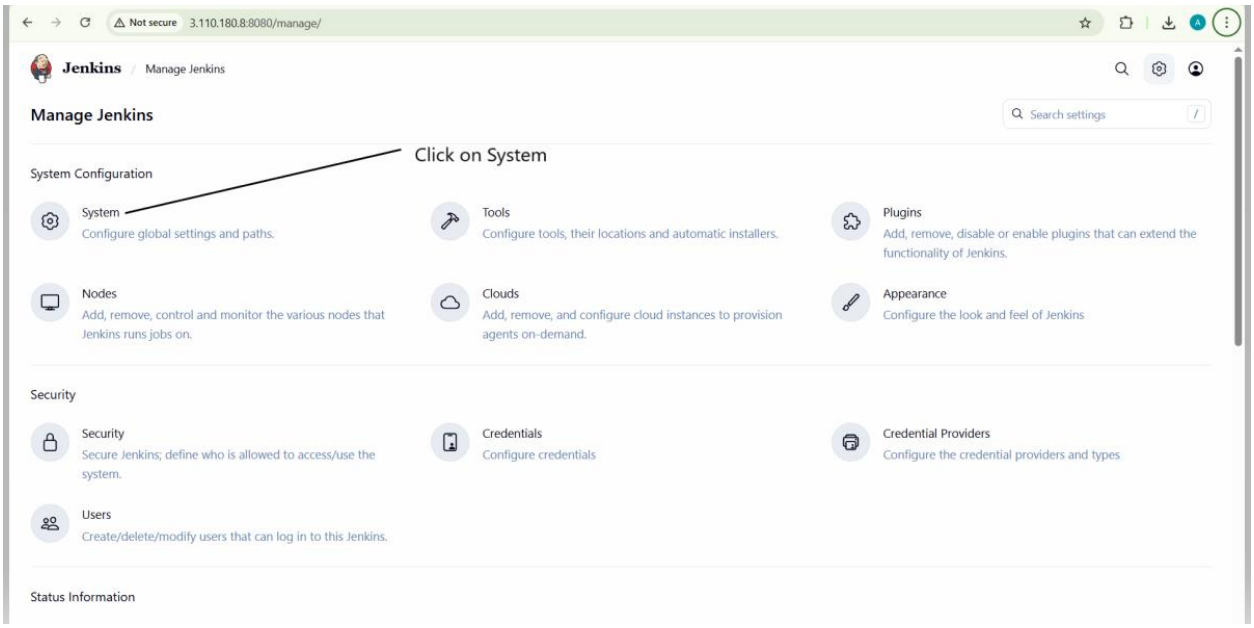
- **In Dropdown click on Secret Text**



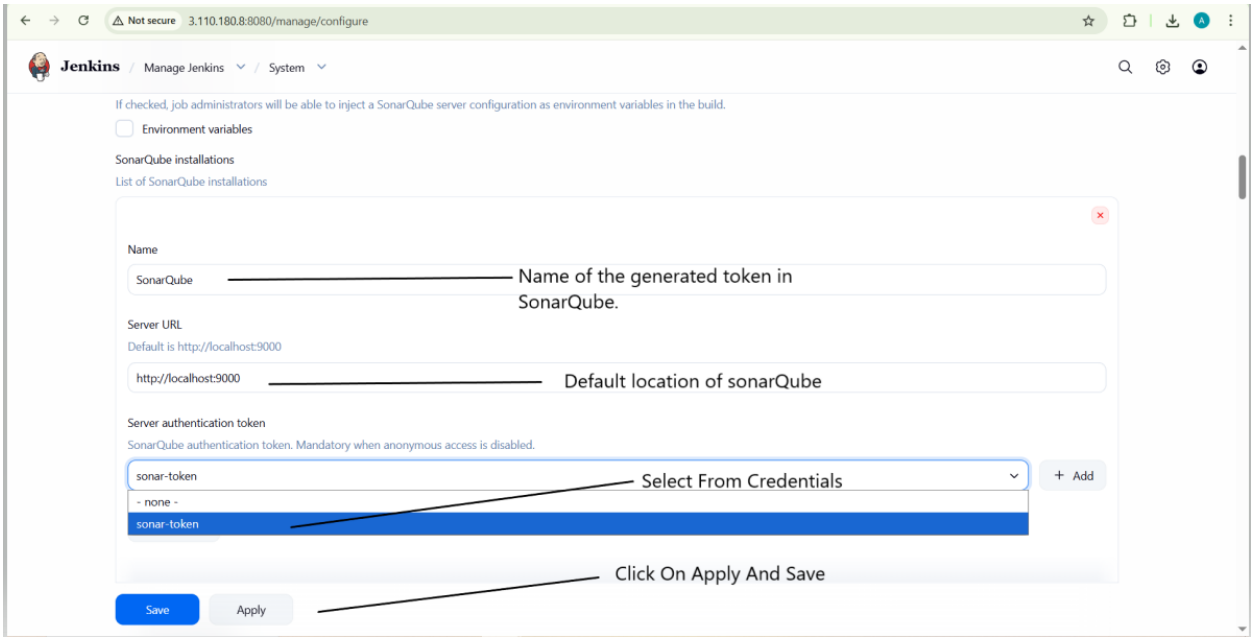
- **Add Paste the generated token → name the variable→ Add Some decription.**



- **Click on System on Manage Jenkins**



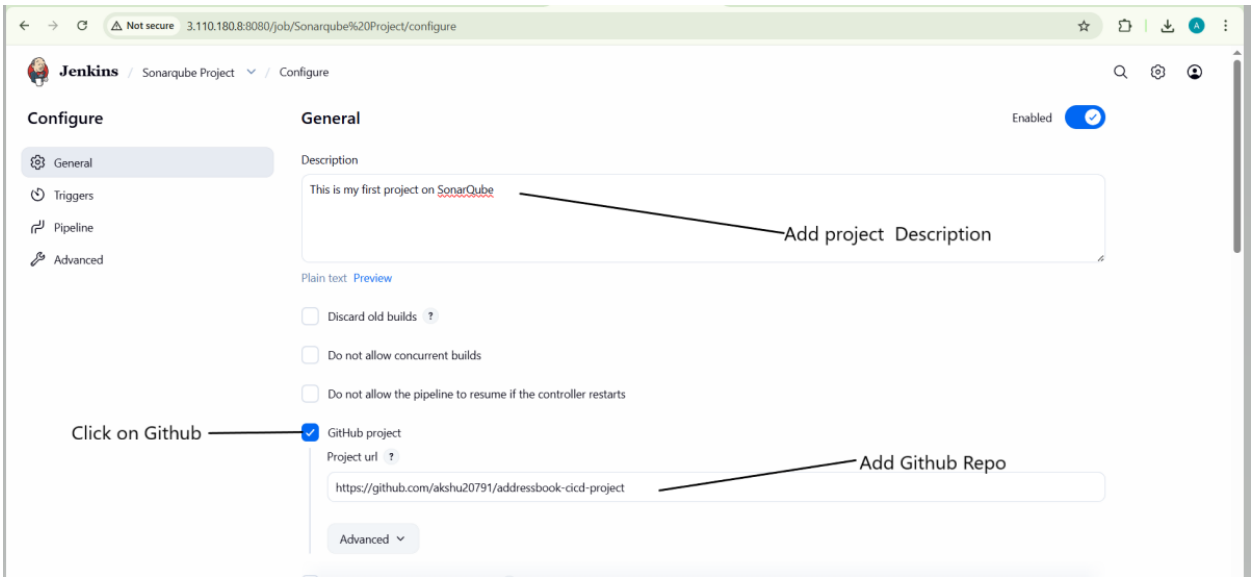
- **Add Generated Token Name → give default location of SonarQube → select the SonarQube Authentication from dropdown→ Apply→ Save**



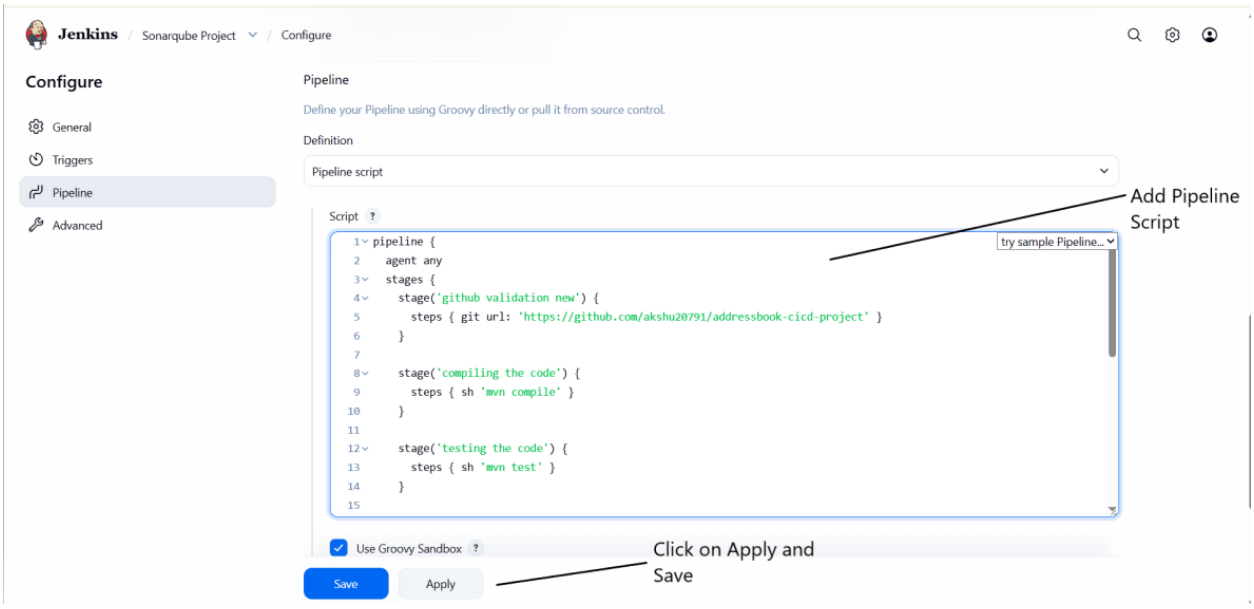
- **Create a New Pipleine for project**
-

- **Name the project → Click on pipleine to be get added→click on ok**

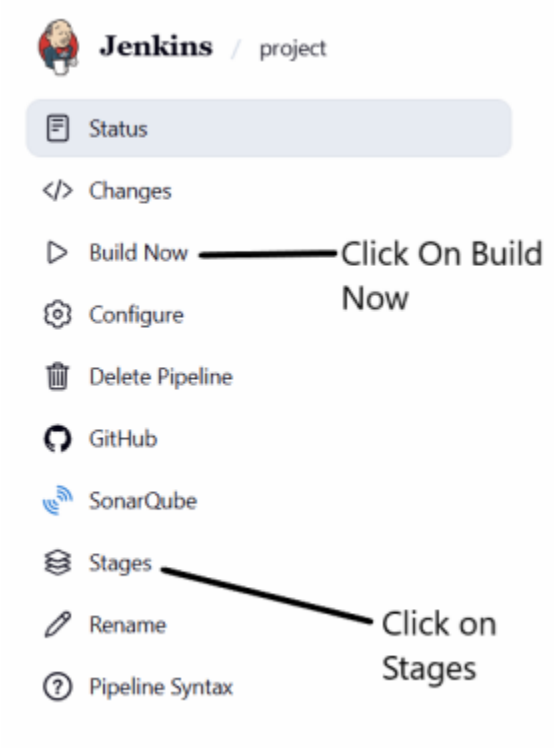
- **Add Project Description → click on checkbox on GitHub → Add GitHub Repository**

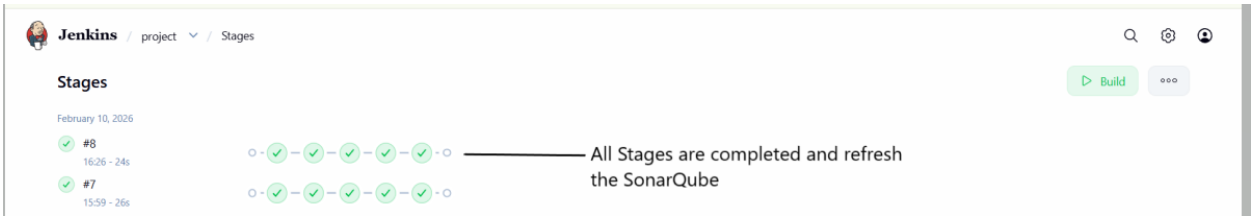


- **Write a pipline script for the project → click on Apply and save**



- **Click on Apply and Save**
- **Build the job**
- **On successful completion of the build from the console output you can see the project in the sonarqube server by clicking the link as shown in the output**





- **Refresh SonarQube → click on Project**