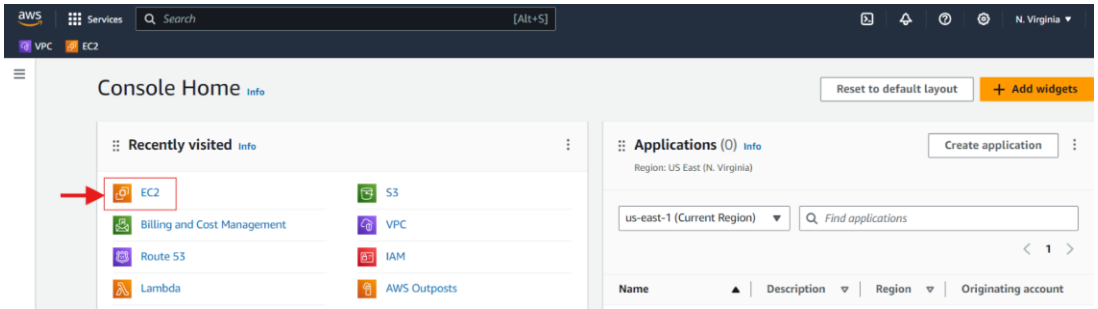
INSTALLING PRE-REQUISITES

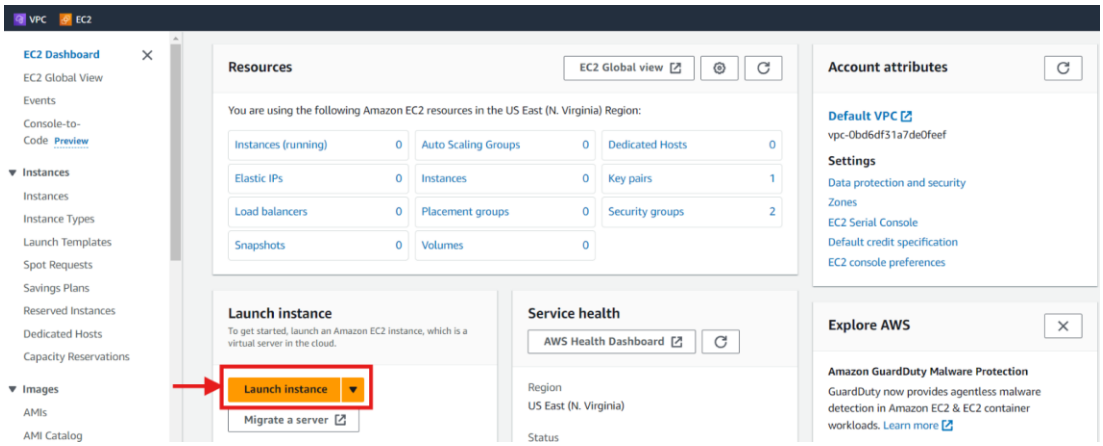
INSTALL AND SETUP JENKINS

✓ Spin Up EC2 instance in AWS

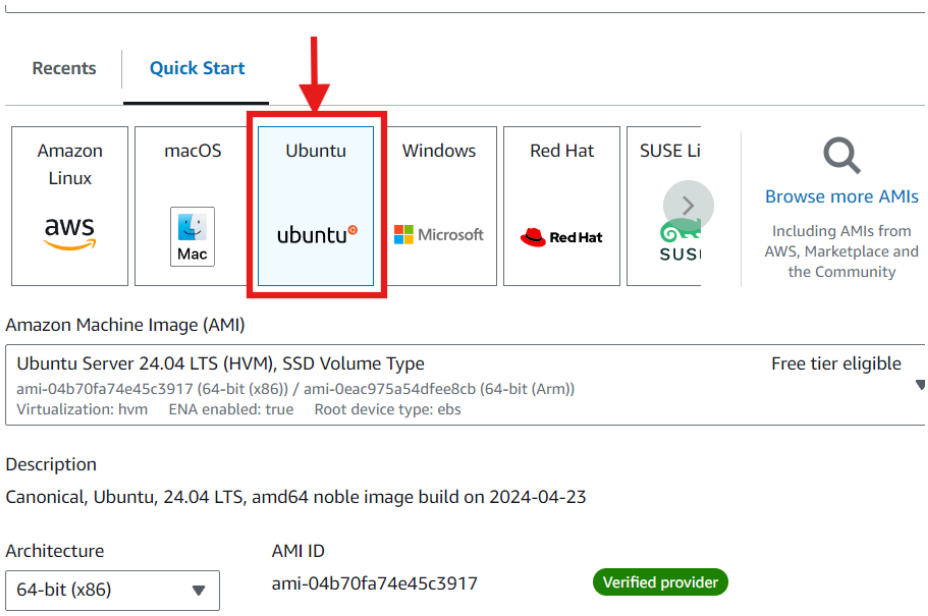
- Login to **AWS console and select EC2 service** (If you're using a noon root account make sure you have the necessary permissions/roles)



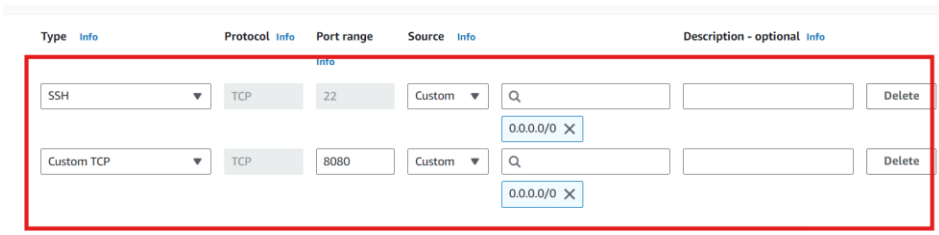
- Select **Launch Instance,** in EC2 service dashboard



- In the launch instance windows, configure the following:-
  - Give the instance a name
  - Select the **Ubuntu Server 20.04 LTS** Quick start Image



- Select any of the Instance types (min **T2.small**)
- **Select a key pair** from available once (* need for SSH)
- In the network setting leave everything as defaults and **select a Security group with the following Inbound traffic ports opened 22 (SSH) 8080 (Jenkins UI)**



- In a configure Storage section set minimum 10GB.(* not mandatory)
- Finish and launch the instance.

✓ Install Java 17

➢ Login to server, and run the following commands (*Ensure user has sudo user privileges)

```
sudo apt update
```

```
sudo apt install openjdk-17-jre
```

✓ Setup Jenkins repository in APT

- Run the following commands

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
```

✓ Install and start Jenkins

- Run the following commands

```
sudo apt-get update && \
sudo apt-get install jenkins -y && \
sudo systemctl enable jenkins  &&\
sudo systemctl start jenkins && \
sudo systemctl status jenkins
```

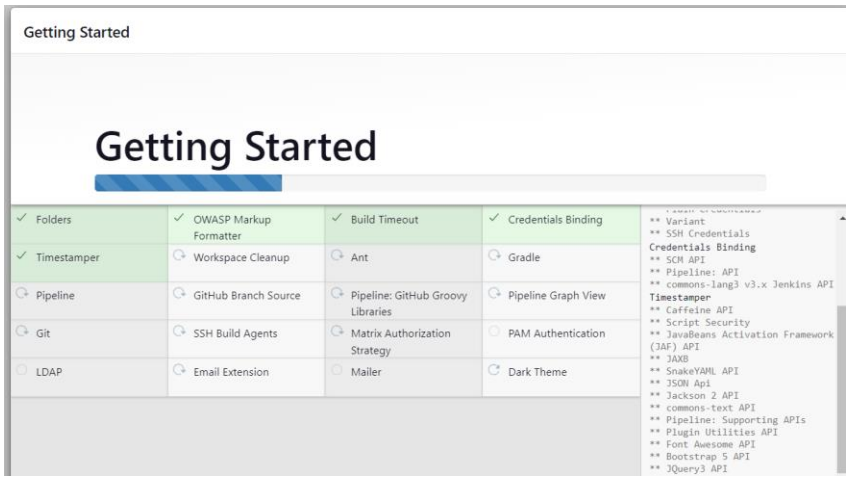
- Make sure that Jenkins service status is running.

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-07-02 08:09:21 UTC; 4s ago
   Main PID: 4659 (java)
      Tasks: 40 (limit: 2338)
     Memory: 343.2M (peak: 393.8M)
        CPU: 48.472s
     CGroup: /system.slice/jenkins.service
             └─4659 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080
```

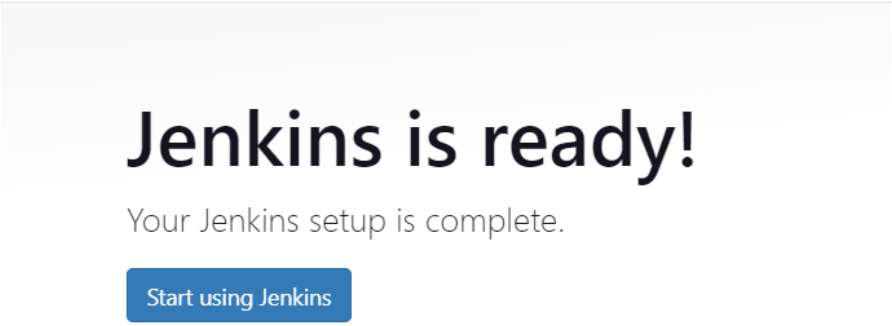
- Access the https URL of Jenkins (**http://<server-public-ip>:8080**) and setup the server.
- Extract the Default Admin password, run the following command

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

- Start the setup and install suggested plugins



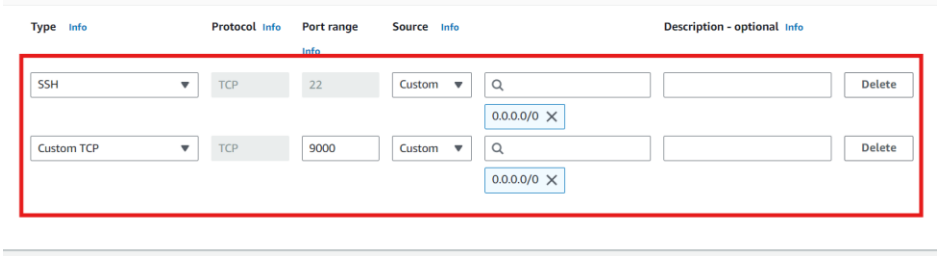
- Finish the setup



INSTALL AND SETUP SONARQUBE SERVER

✓ Spin Up EC2 instance in AWS

- Login to **AWS console and select EC2 service** (If you're using a noon root account make sure you have the necessary permissions/roles.
- Select **Launch Instance,** in EC2 service dashboard
- In the launch instance windows, configure the following:-
  - Give the instance a name
  - Select the **Ubuntu Server 20.04 LTS** Quick start Image
  - Select any of the Instance types (min **T2.medium**)
  - **Select a key pair** from available once (* need for SSH)
  - In the network setting leave everything as defaults and **select a Security group with the following Inbound traffic ports opened 22 (SSH) 9000 (Sonarqube UI)**



- In a configure Storage section set minimum 10GB.(* not mandatory)
- Finish and launch the instance.

✓ Install Docker

- Login to server, and run the following commands (*Ensure user has sudo user privileges)
- Add Docker's official GPG key:

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

- Add the repository to Apt sources:

```
echo \
  "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

- Install docker and its components

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-
buildx-plugin docker-compose-plugin
```

✓ Configure non-root executable permission to Docker

- Run the following commands

```
sudo usermod -aG docker $USER
newgrp docker
```

✓ Start SonarQube Container

- Run the following commands (*here we're using the lts community version since it doesn't require any license)

```
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
```

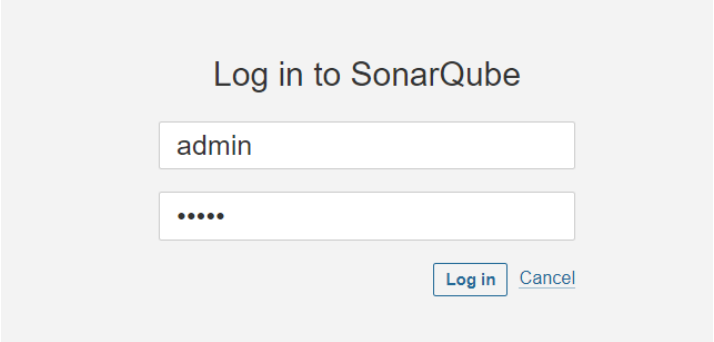
```
ubuntu@ip-172-31-90-0:~$ docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
2ec76a50fe7c: Pull complete
fab7f202453a: Pull complete
ee59ca42def8: Pull complete
2ce2282f972f: Pull complete
d2a9e456ba82: Pull complete
3c4f423f73fe: Pull complete
0110391fdd36: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:0d36ee97e5f5458351913e69aa5ae874d832d886704573bbdce43779c0294733
Status: Downloaded newer image for sonarqube:lts-community
41a99420487275eb30d3051338658f66268308a095116cd125ea3e6b3fec7549
```

- Check if the container is up and running by running following commands

```
docker ps
```

```
ubuntu@ip-172-31-90-0:~$ docker ps
CONTAINER ID   IMAGE                   COMMAND                  CREATED              STATUS              PORTS                                       NAMES
41a994204872   sonarqube:lts-community   "/opt/sonarqube/dock…"   About a minute ago   Up About a minute   0.0.0.0:9000->9000/tcp, :::9000->9000/tcp   sonar
```

- Login to Sonarqube UI (http://<server-public-ip>:9000) using the default password **(Username: admin, Password: admin)**.

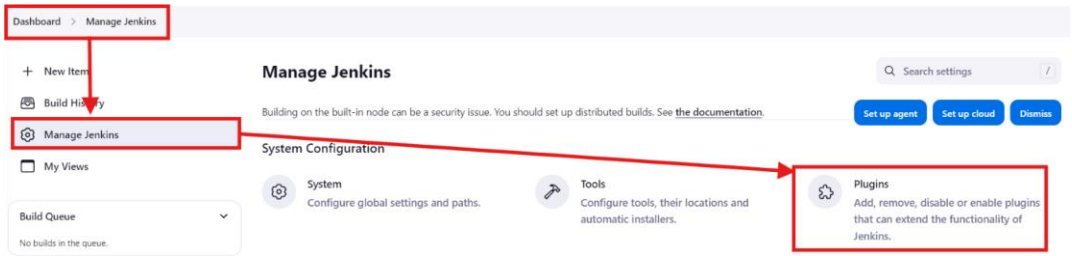


- Update the password and finish the setup.

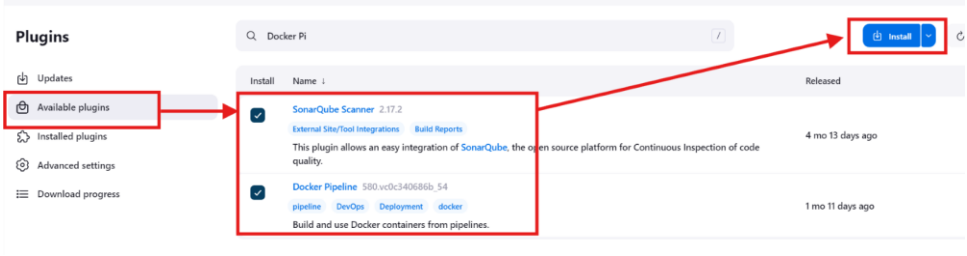
POST INSTALLATION CONFIGURATION

✓ Install the necessary Plugins (SonarScanner, Docker Pipeline)

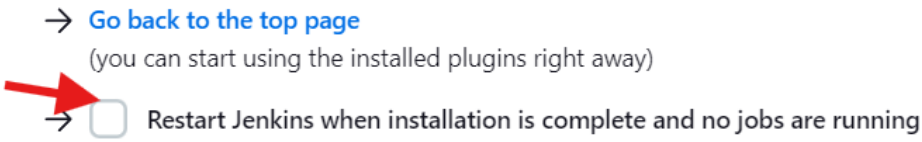
- In the **Jenkins Dashboard select Manage Jenkins**.



- Select Plugins > Available Pulgins > Search for
  - **SonarQube Scanner (Configure Sonar server and Scanner)**
  - **Docker Pipeline (for using docker node to run pipelines)**

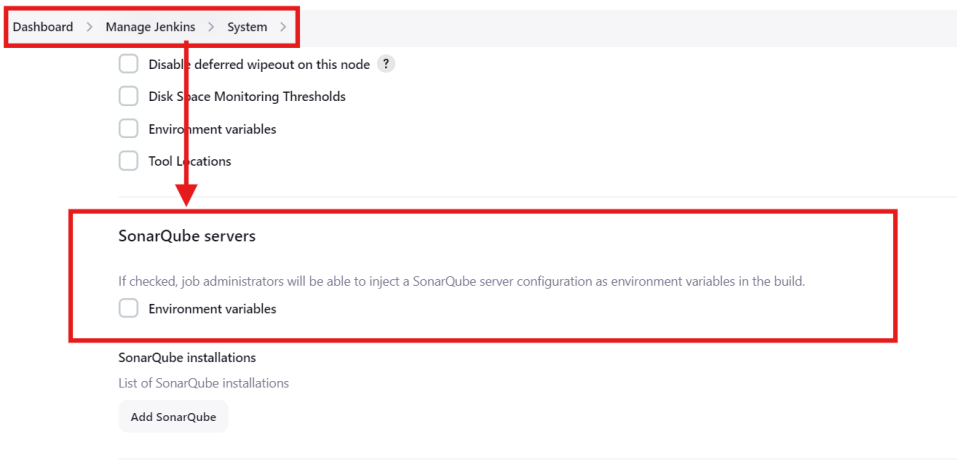


  - Install the Plugin's

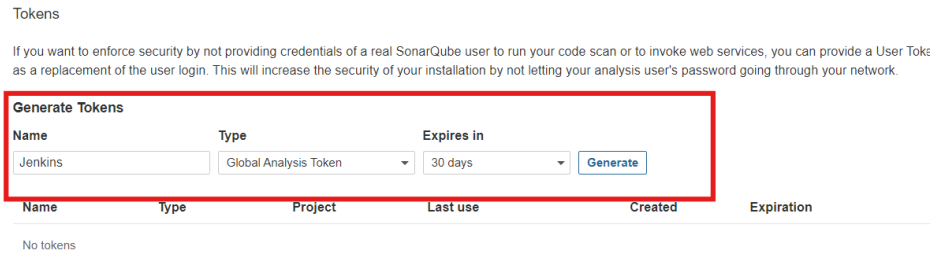


✓ Configure Sonar Server and Scanner in Jenkins

- In the **Jenkins Dashboard select Manage Jenkins**.
- Select the **System option and scroll down to Sonar-Server Section**

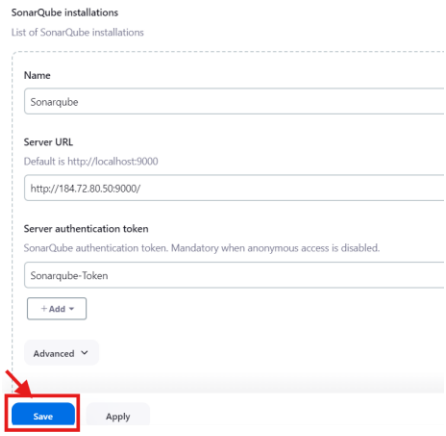


- Setup the Sonar-sever instance.
- Give a preferred name to the instance (*to be used in pipeline stage import Sonar sever configuration as environment variables)
- Input the server URL
- Create a Token in Sonar server by heading over to **User > Security > Global Token Create Token**. Make sure to copy the token.

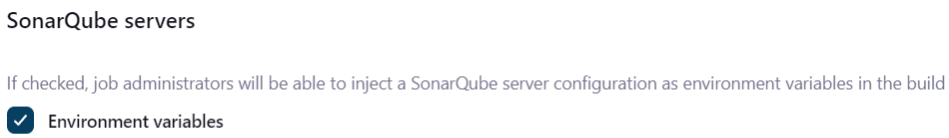


- Select add token, and create a new Secret file from the drop-down menu

- Save the Configuration

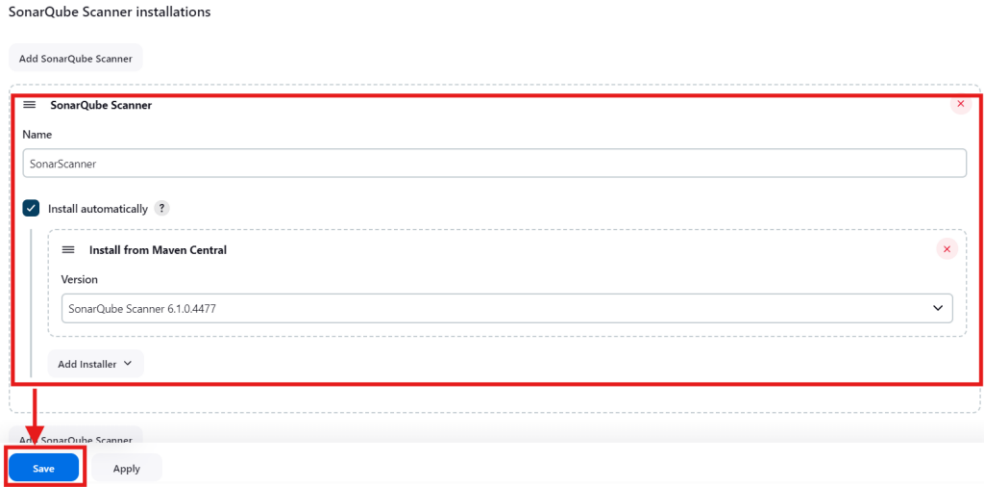


- Ensure to tick the option Environment variable (*to ingest configuration as environment variable to builds)



✓ Configure SonarScanner

- In the **tools section headover to SonarScanner**
- Configure the scanner **Version(* not mandatory if latest version is suitable), and click Install Automatically**
- Save the changes



✓ Configure Docker

- In the **tools section headover to Docker**
- Give it a name
- Input the installation root directory if already installed docker on host machine, else tick install automatically
- Save the changes