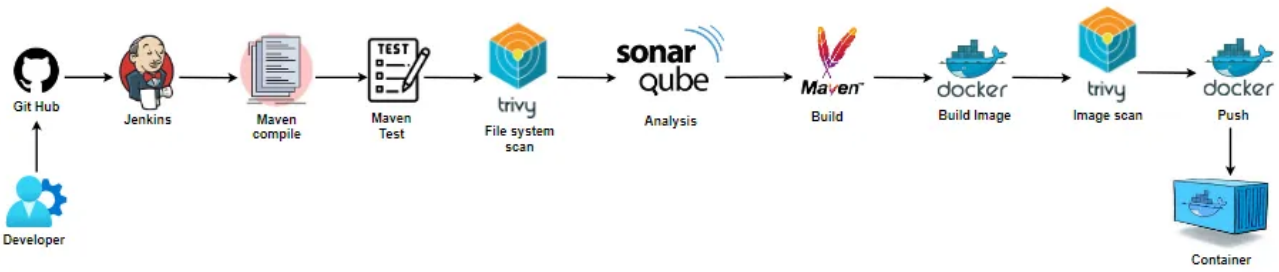
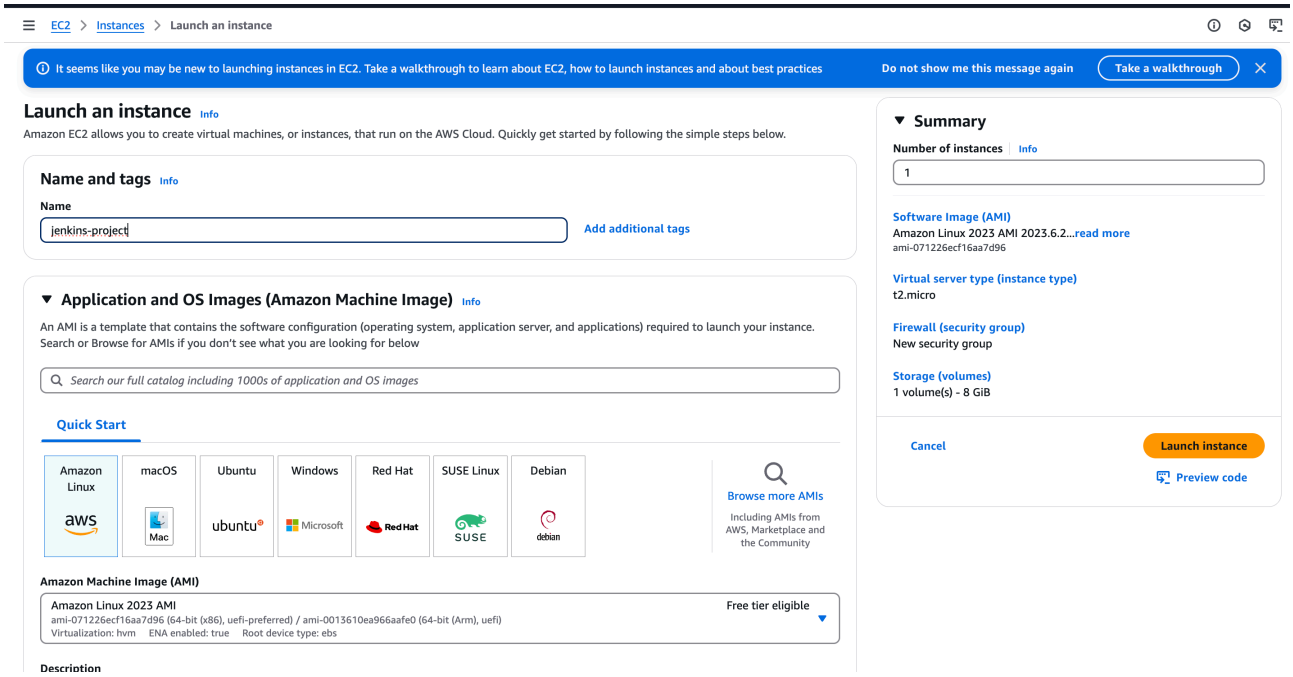
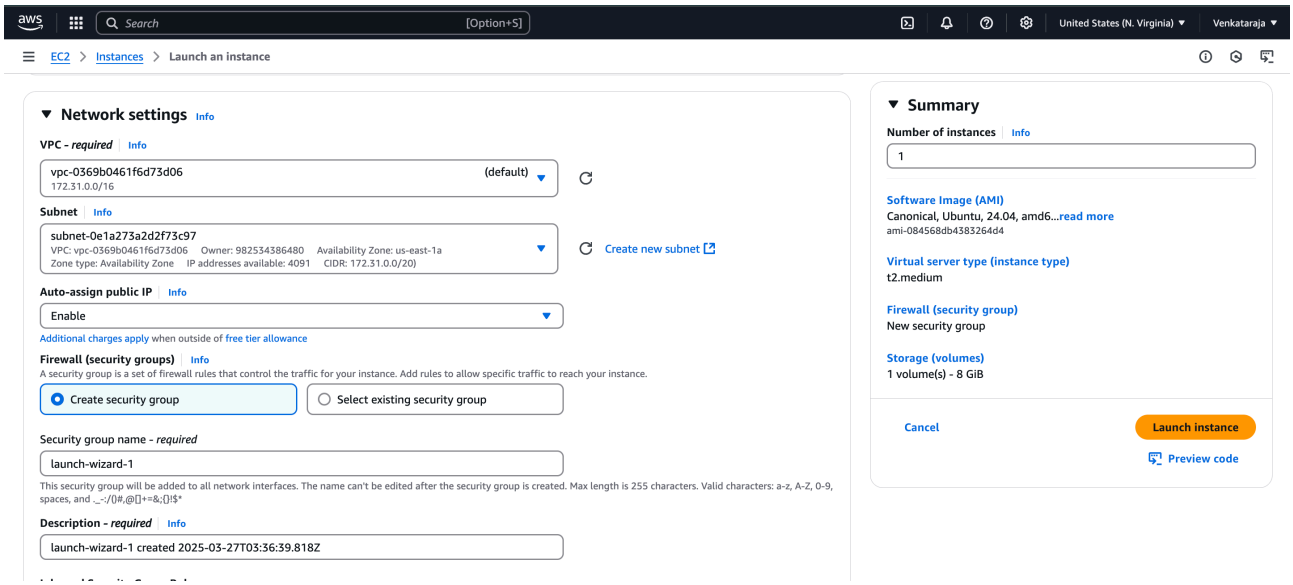
CI PIPELINE SETUP WITH VARIOUS TOOLS

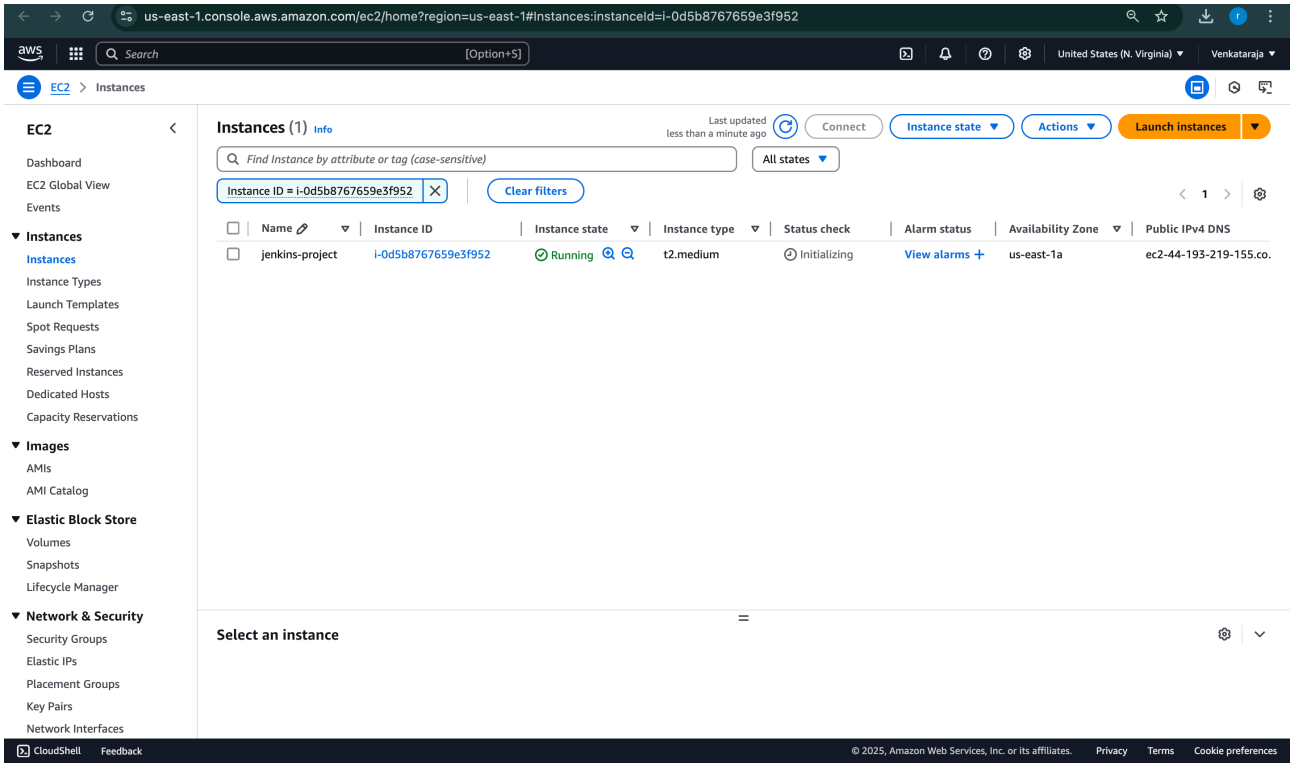


Step 1: Create an EC2 Instance

1. Log in to your AWS account.
2. Navigate to **EC2 Dashboard** -> **Launch Instance**.
3. Choose **Ubuntu t2.medium**.
4. Allocate **30GB EBS volume**.
5. Select **US-EAST-1** as the region.
6. Configure security group rules to allow necessary ports (8080 for Jenkins, 9000 for SonarQube, etc.).







Step 2: Connect to EC2 and Install Required Tools

1. Connect to the instance using SSH:

   ssh -i <your-key.pem> ubuntu@<EC2_PUBLIC_IP>

2. Switch to root user:

   sudo -i

Step 3: Install Jenkins

Run the following script to install Jenkins:

sudo apt update -y

sudo apt upgrade -y

sudo apt install openjdk-17-jre -y

*curl -fsSL https://pkg.jenkins.io/debian-stable/. jenkins.io-2023.key | sudo tee *

/usr/share/keyrings/jenkins-keyring.asc > /dev/null

echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc

*https://pkg.jenkins.io/debian-stable binary/ | sudo tee *

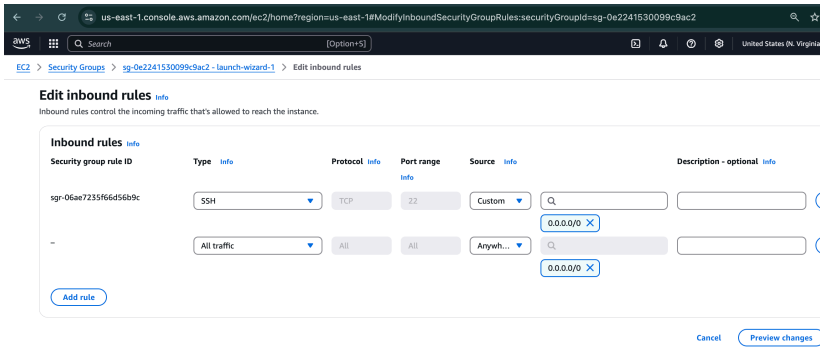
/etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt-get update -y

sudo apt-get install jenkins -y

Step 4: Update EC2 Security Group

- Add an **inbound rule** to allow **port 8080** (for Jenkins access).
- Save the changes.



Step 5: Access Jenkins Dashboard

1. Open a browser and enter:

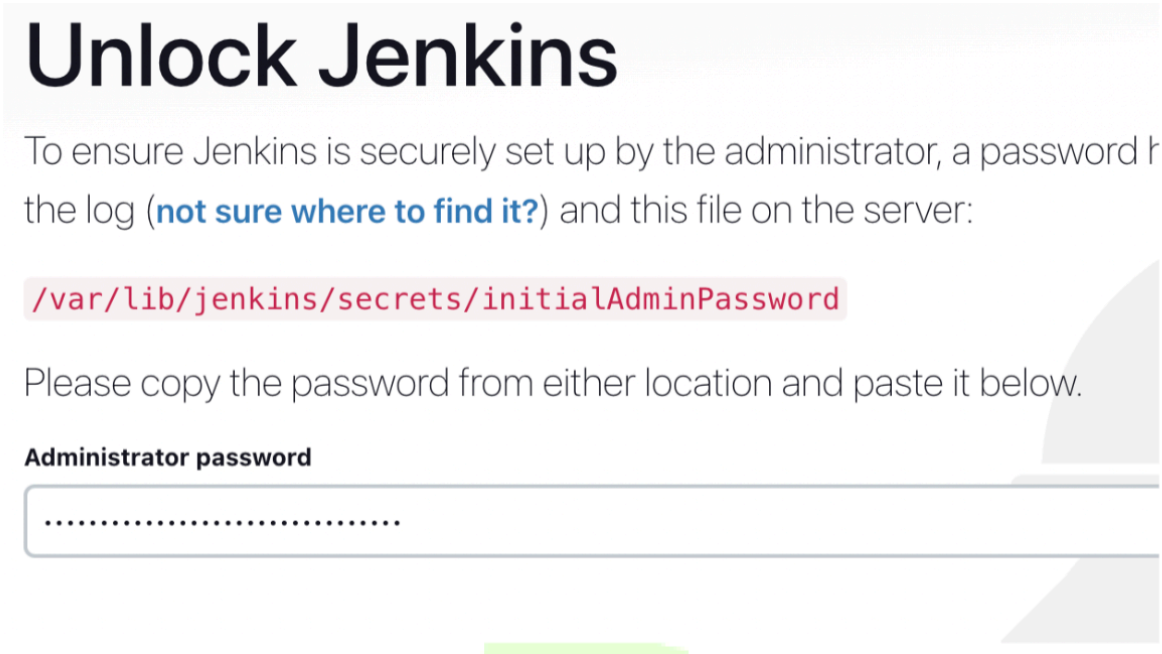
   http://<EC2_PUBLIC_IP>:8080/

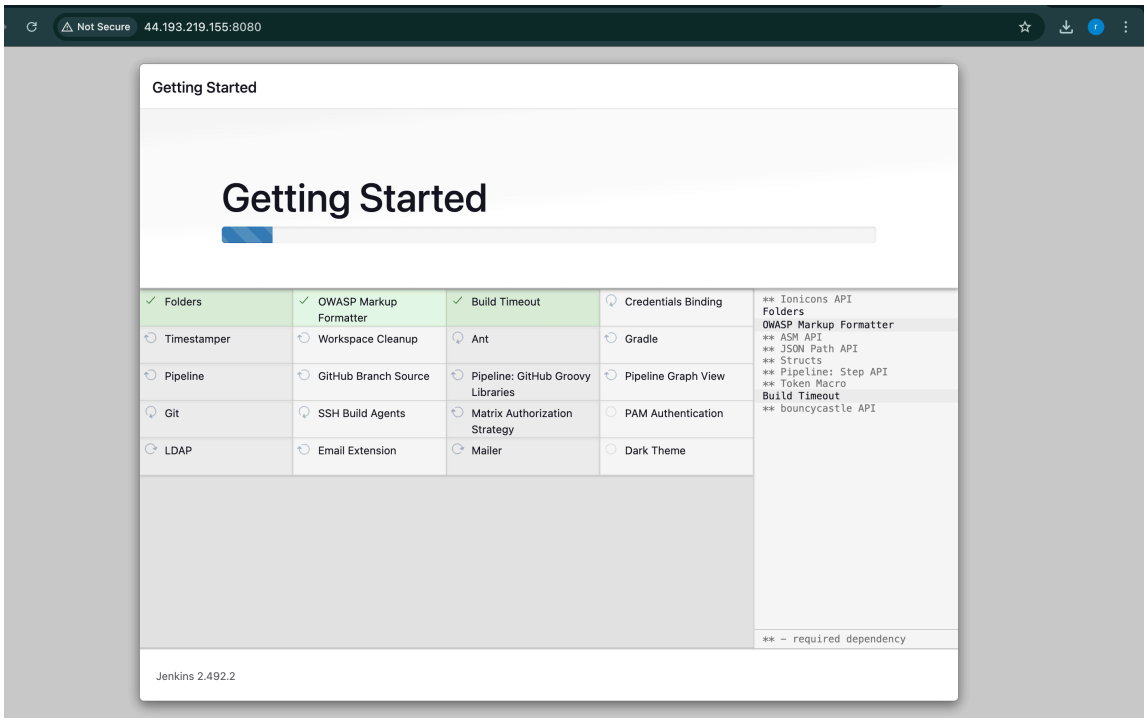
2. Retrieve the administrator password:

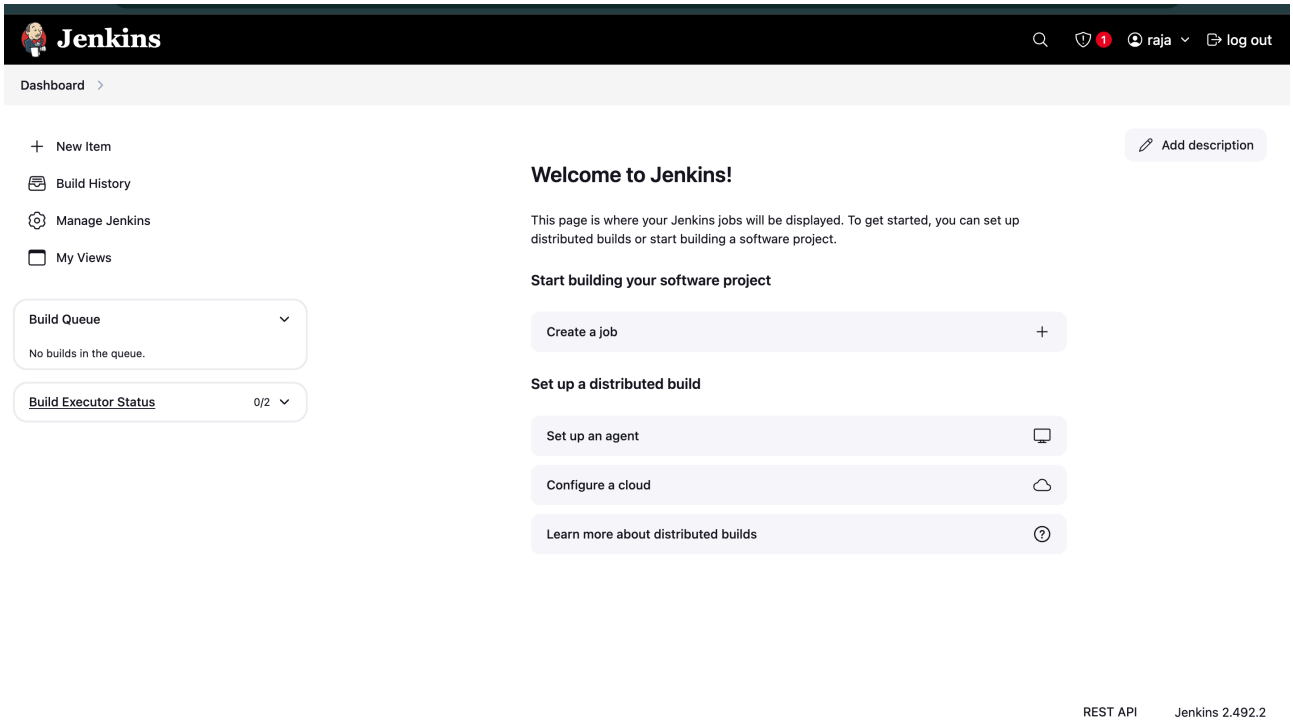
   cat /var/lib/jenkins/secrets/initialAdminPassword

3. Install all suggested plugins.

4. Create the first admin user.



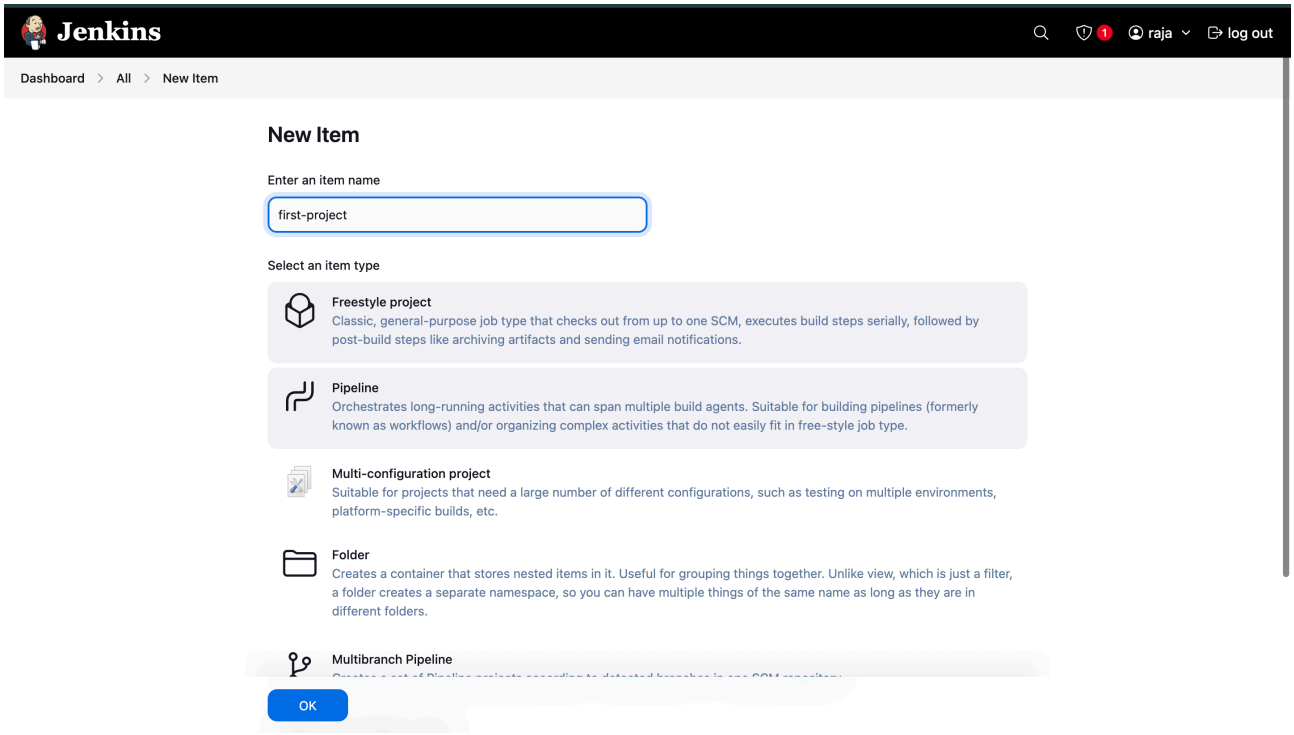


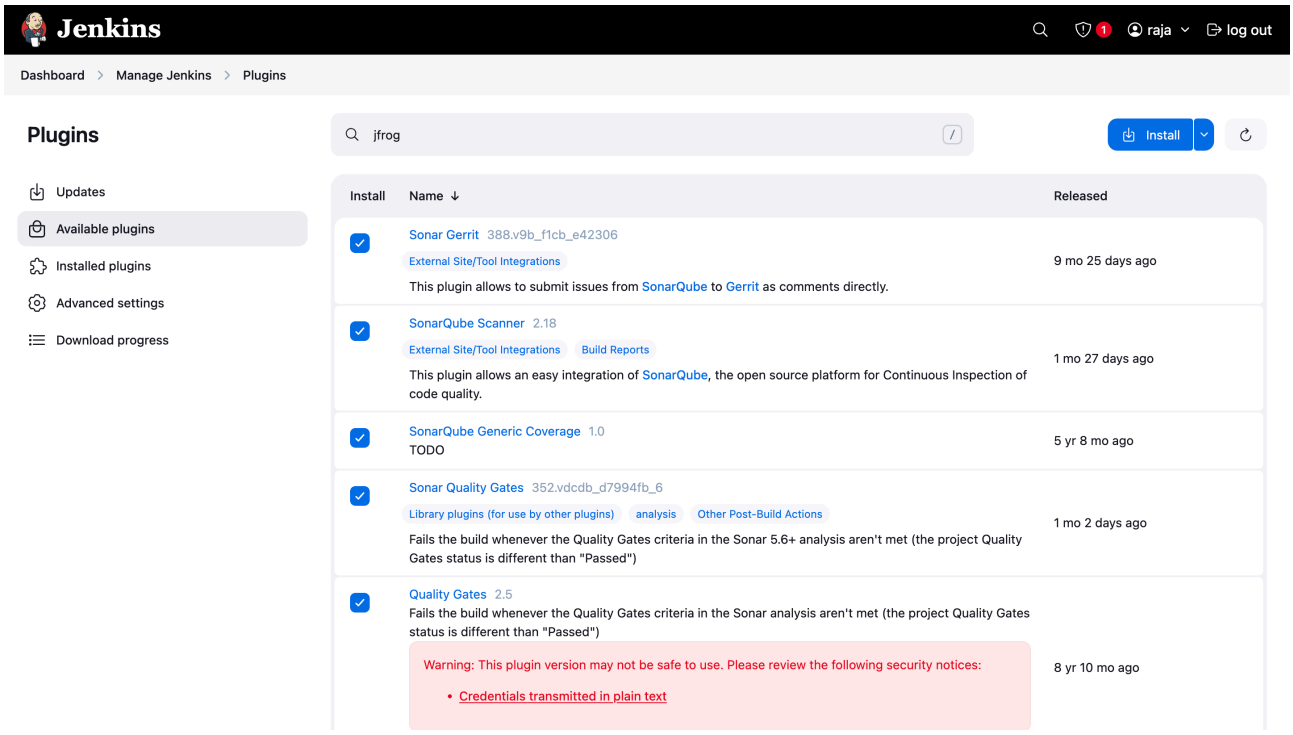


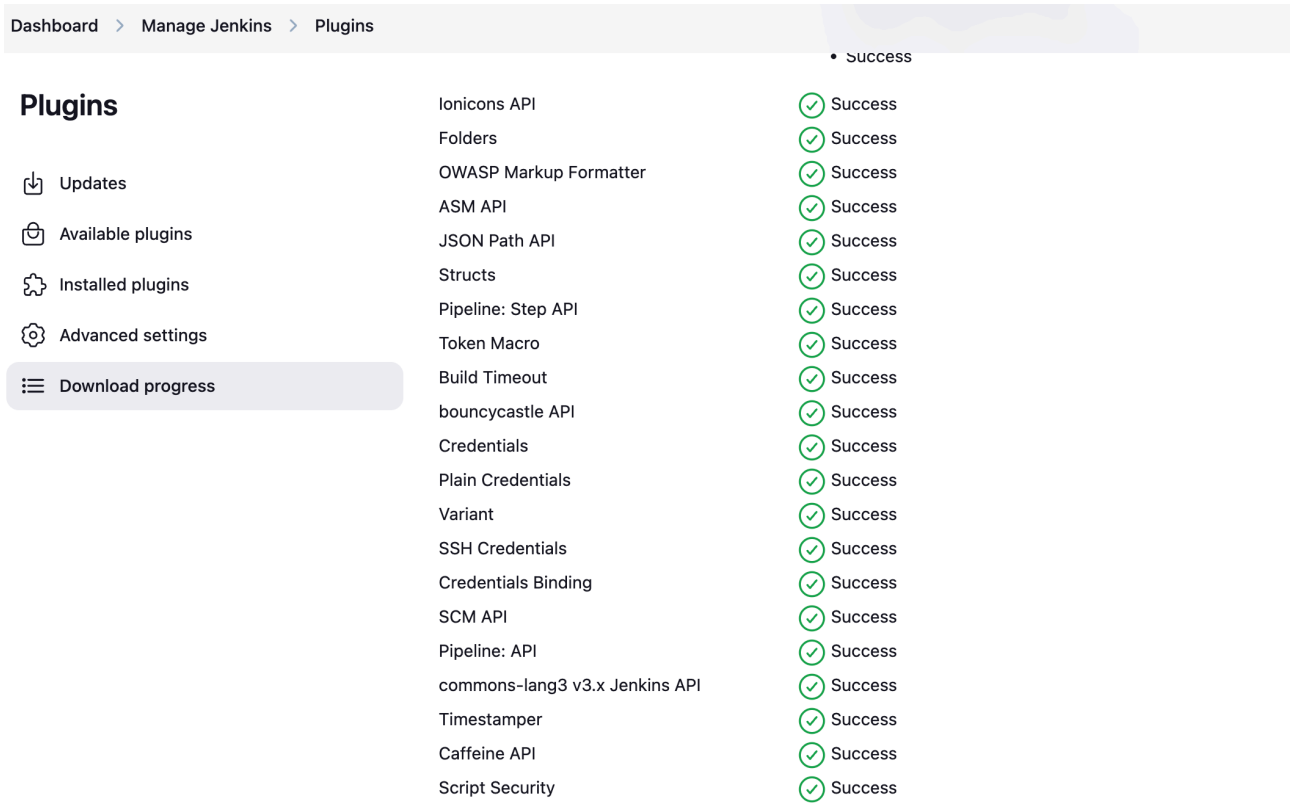
Step 6: Setup CI/CD Pipeline in Jenkins

1. **Create a New Pipeline Job**
2. **Use SCM for Pipeline Script:**

   `https://github.com/venkatraja1234/Java_app_3.0.git`
3. **Install Required Plugins:**
   - SonarQube Scanner
   - Quality Gates
   - Artifactory
   - JFrog
   - Docker


Jenkins
raja
log out
Dashboard
All
New Item
New Item
Enter an item name
first-project
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
Multibranch Pipeline
OK


Jenkins
raja
log out
Dashboard
Manage Jenkins
Plugins
Plugins
jfrog
Install
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Install
Name
Released
Sonar Gerrit 388.v9b_f1cb_e42306
External Site/Tool Integrations
This plugin allows to submit issues from SonarQube to Gerrit as comments directly.
9 mo 25 days ago
SonarQube Scanner 2.18
External Site/Tool Integrations
Build Reports
This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.
1 mo 27 days ago
SonarQube Generic Coverage 1.0
TODO
5 yr 8 mo ago
Sonar Quality Gates 352.vdcdb_d7994fb_6
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 2 days ago
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
8 yr 10 mo ago



Step 7: Install Docker

Run the following script to install Docker:

sudo apt update -y

sudo apt install apt-transport-https ca-certificates curl software-properties-common -y

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable" -y

sudo apt update -y

apt-cache policy docker-ce -y

sudo apt install docker-ce -y

sudo chmod 777 /var/run/docker.sock

Verify installation:

docker -v

```
Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.

Restarting services...

Service restarts being deferred:
 /etc/needrestart/restart.d/dbus.service
 systemctl restart networkd-dispatcher.service
 systemctl restart systemd-logind.service
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #2: sshd[1115]
 ubuntu @ user manager service: systemd[1120]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-14-134:~# chmod 777 /var/run/docker.sock
root@ip-172-31-14-134:~# docker -v
Docker version 24.0.2, build cb74dfc
root@ip-172-31-14-134:~# ps
    PID TTY          TIME CMD
  18089 pts/3    00:00:00 sudo
  18090 pts/3    00:00:00 bash
  20382 pts/3    00:00:00 ps
root@ip-172-31-14-134:~#  docker run -d --name sonarqube -p 9000:9000 -p 9092:9092 sonarqube
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
5a7813e071bf: Pull complete
dbe46403441a: Pull complete
f9f4ee04af87: Pull complete
e3da94a33fa1: Pull complete
f03e4717322c: Pull complete
c9439e8e4945: Download complete
62f1017e9142: Download complete
4f4fb700ef54: Download complete
$
```

```
root@ip-172-31-14-134:~#  docker run -d --name sonarqube -p 9000:9000 -p 9092:9092 sonarqube
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
5a7813e071bf: Pull complete
dbe46403441a: Pull complete
f9f4ee04af87: Pull complete
e3da94a33fa1: Pull complete
f03e4717322c: Pull complete
c9439e8e4945: Pull complete
62f1017e9142: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:c0a734bd7e62c1a35794e3a070b4945f5a081b0053201eb926bcb936b0e5d2e6
Status: Downloaded newer image for sonarqube:latest
76fc760871bf4e5749608d9bf8f5b499553fbc48233ea227cc289026c895d40a
root@ip-172-31-14-134:~# $docker ps-a
ps-a: command not found
root@ip-172-31-14-134:~# $docker ps -a
    PID TTY          TIME CMD
   1275 pts/0    00:00:00 sudo
   1277 pts/1    00:00:00 bash
  18088 pts/2    00:00:00 sudo
  18090 pts/3    00:00:00 bash
  20854 pts/3    00:00:00 ps
root@ip-172-31-14-134:~#  docker ps -a
CONTAINER ID   IMAGE       COMMAND                  CREATED         STATUS         PORTS
                                    NAMES
76fc760871bf   sonarqube   "/opt/sonarqube/dock…"   2 minutes ago   Up 2 minutes   0.0.0.0:9000->9000/tcp, :::9000->9000/tcp, 0.0.
0.0:9092->9092/tcp, :::9092->9092/tcp   sonarqube
root@ip-172-31-14-134:~# docker start
"docker start" requires at least 1 argument.
See 'docker start --help'.

Usage:  docker start [OPTIONS] CONTAINER [CONTAINER...]

Start one or more stopped containers
root@ip-172-31-14-134:~# docker start 76fc760871bf
76fc760871bf
root@ip-172-31-14-134:~#
```

Step 8: Install and Configure SonarQube

Run the installation script:

docker run -d --name sonarqube -p 9000:9000 -p 9092:9092 sonarqube

Start SonarQube if it's not running:

docker ps -a # Get container ID

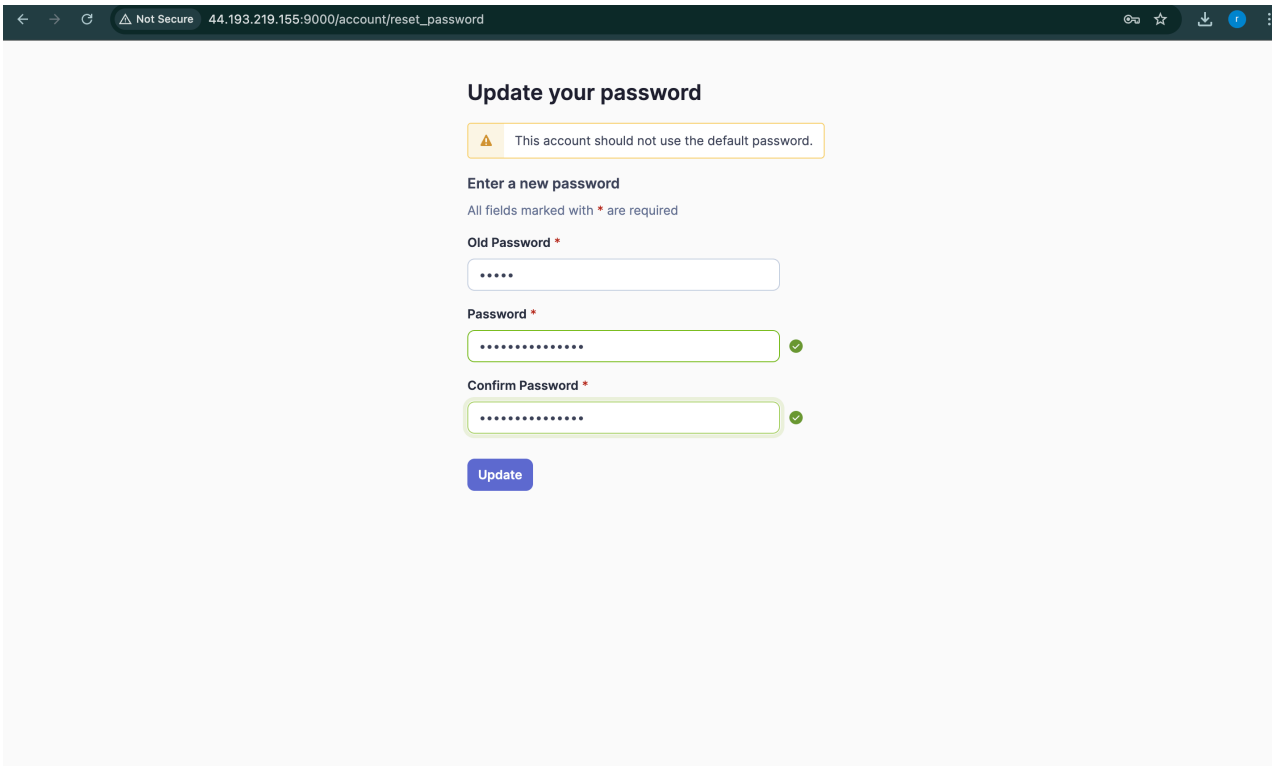
docker start <containerID>

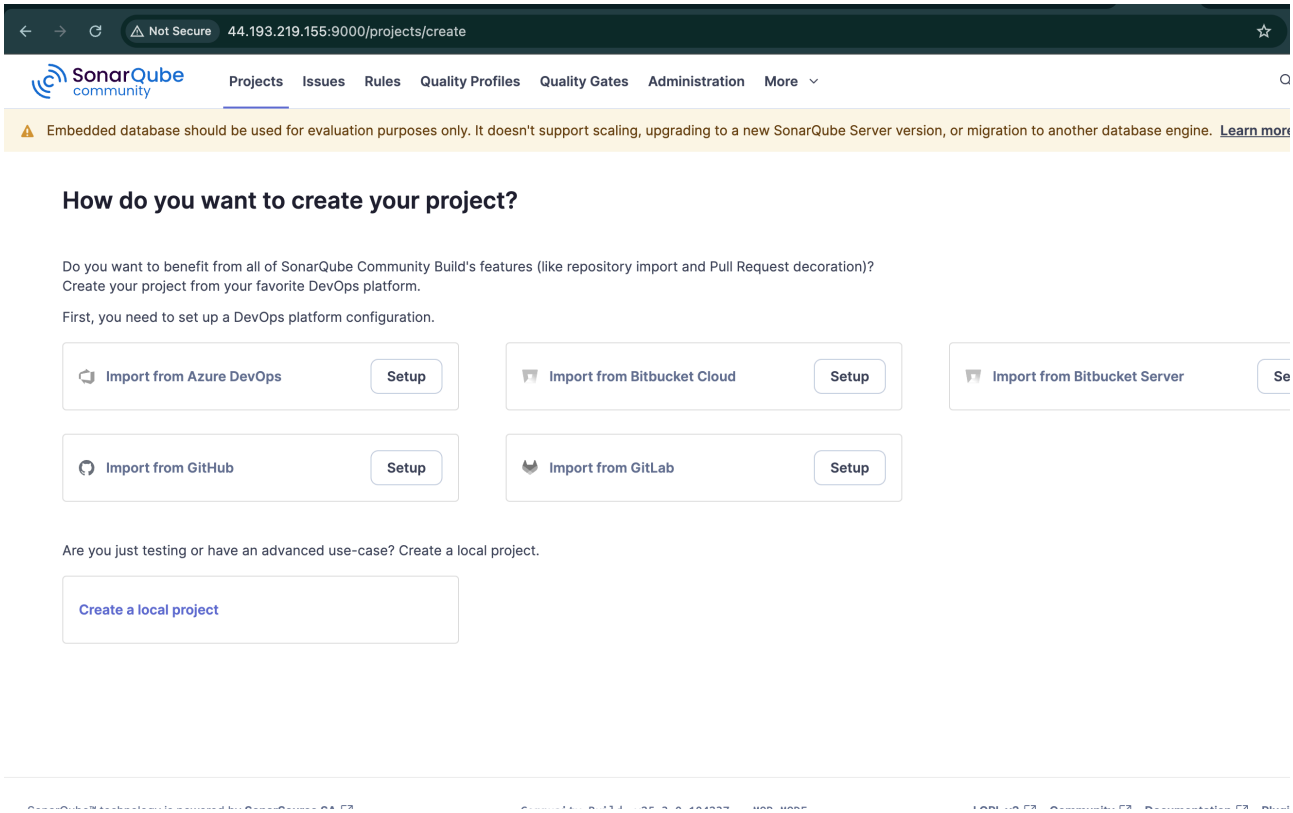
Access SonarQube:

http://<EC2_PUBLIC_IP>:9000/

Login Credentials:

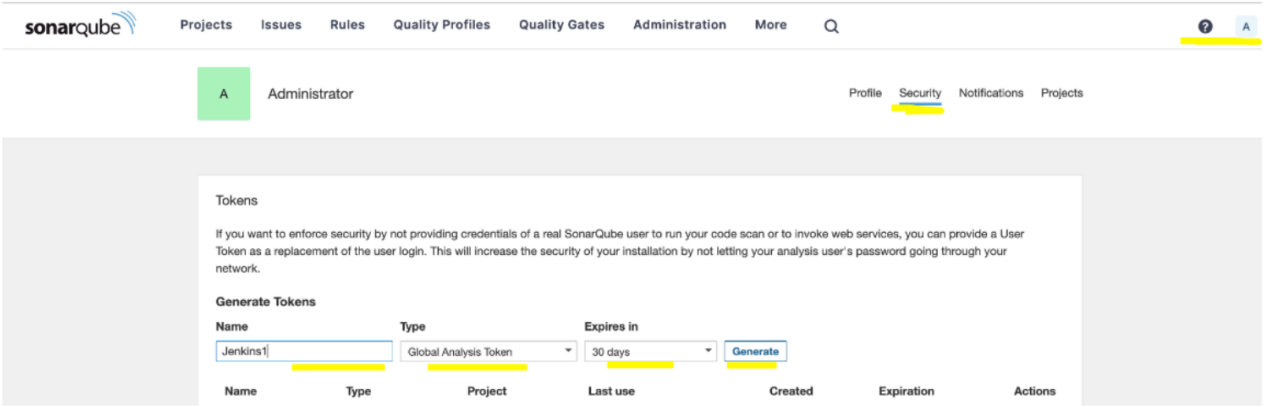
- Username: `admin`
- Password: `admin`





Generate a token:

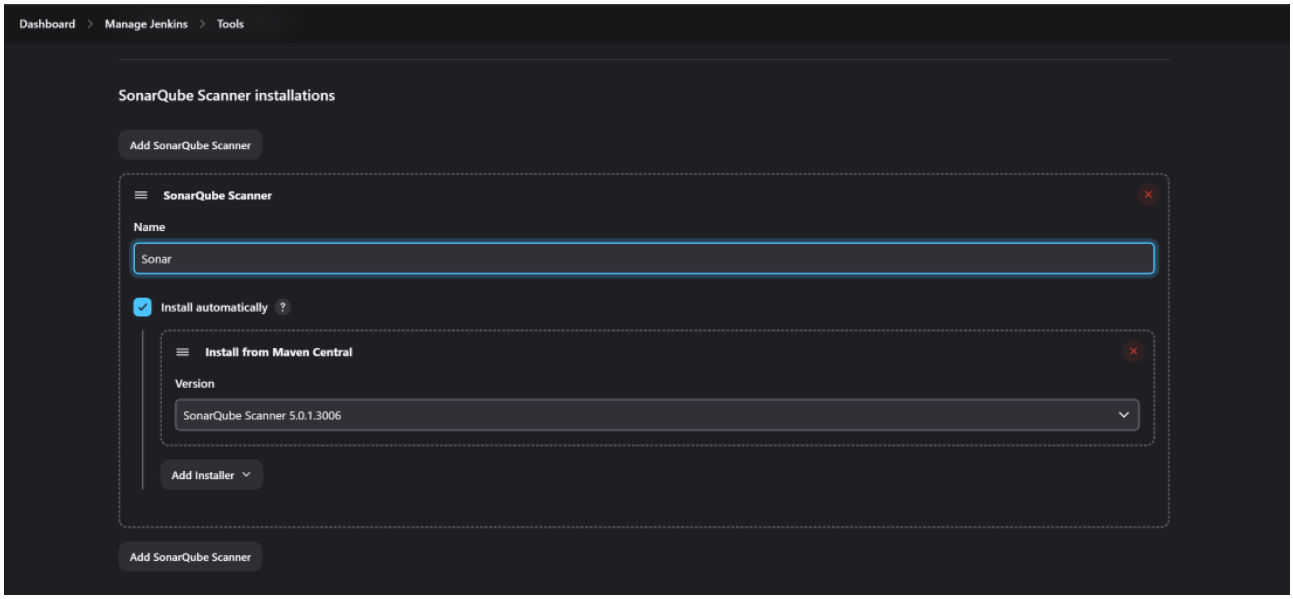
1. Go to **Administration -> My Account -> Security**.
2. Click **Create Token**.
3. Save the token for Jenkins integration.

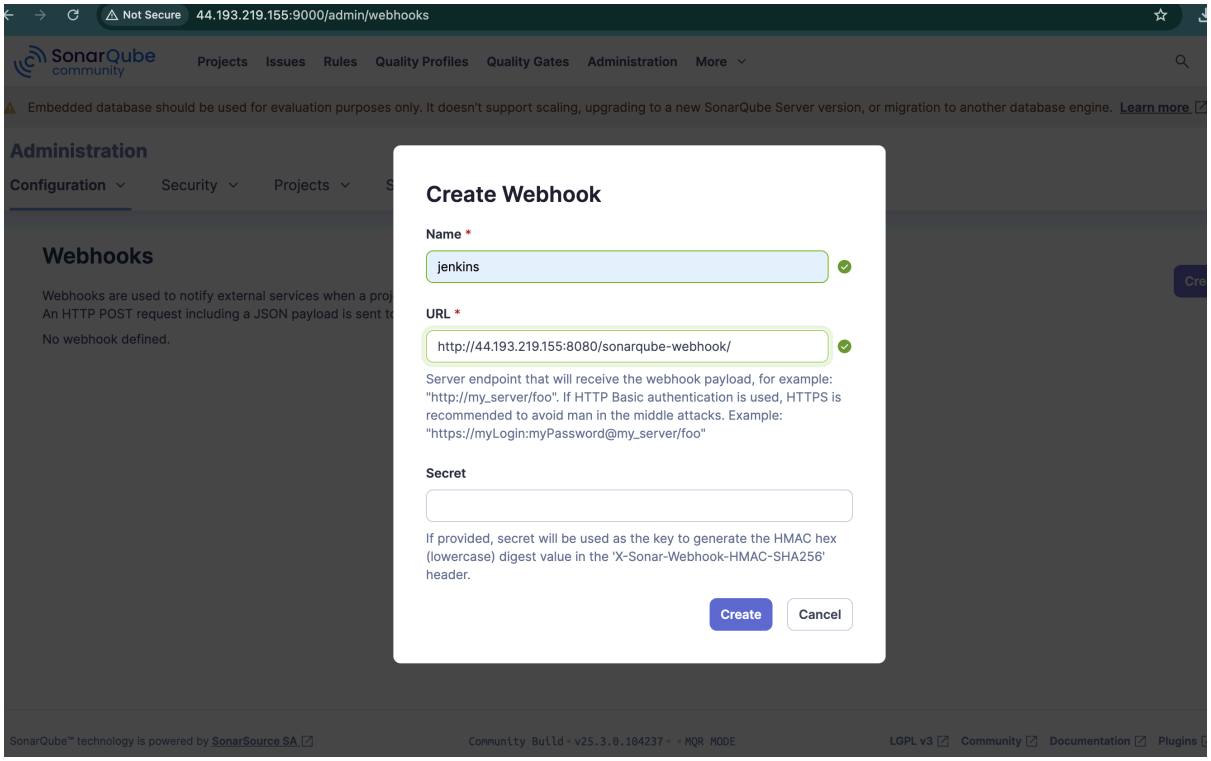


Step 9: Integrate SonarQube with Jenkins

1. Go to **Jenkins Dashboard -> Manage Jenkins -> Configure System**.
2. Locate **SonarQube servers**.
3. Add SonarQube URL and Token (generated in Step 8).
4. Configure **webhook**:

   http://<EC2_IP>:8080/sonarqube-webhook/





Step 10: Install Maven

Run the following script:

sudo apt update -y

sudo apt install maven -y

mvn -version

Step 11: Install Trivy for Security Scanning

Run the script:

sudo apt-get install wget apt-transport-https gnupg lsb-release

wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add -

echo deb https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main | sudo tee -a /etc/. apt/sources.list.d/trivy.list

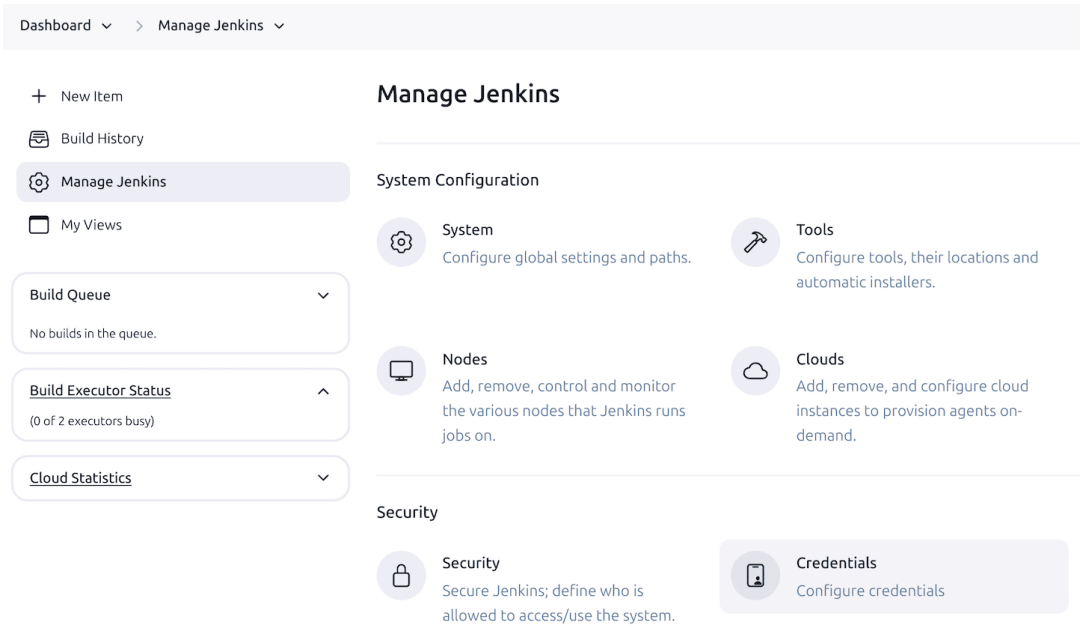
sudo apt-get update

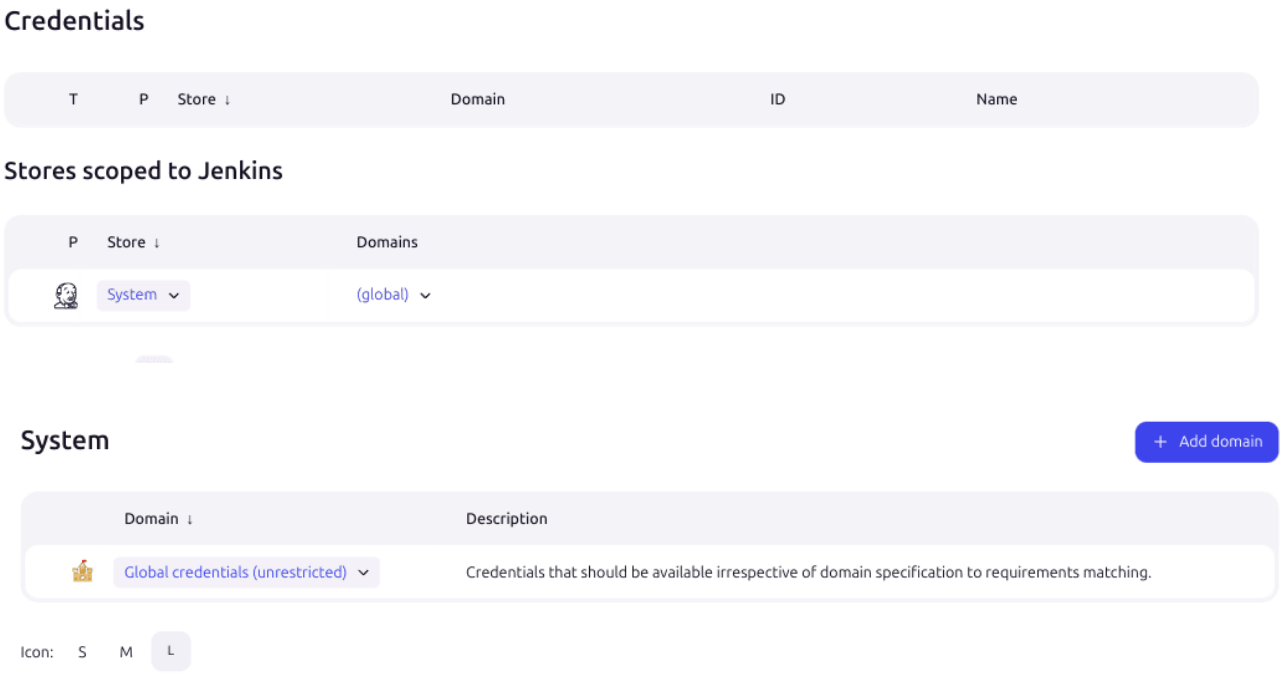
sudo apt-get install trivy

Step 12: Configure Jenkins Credentials

12.1 Add Docker Hub Credentials

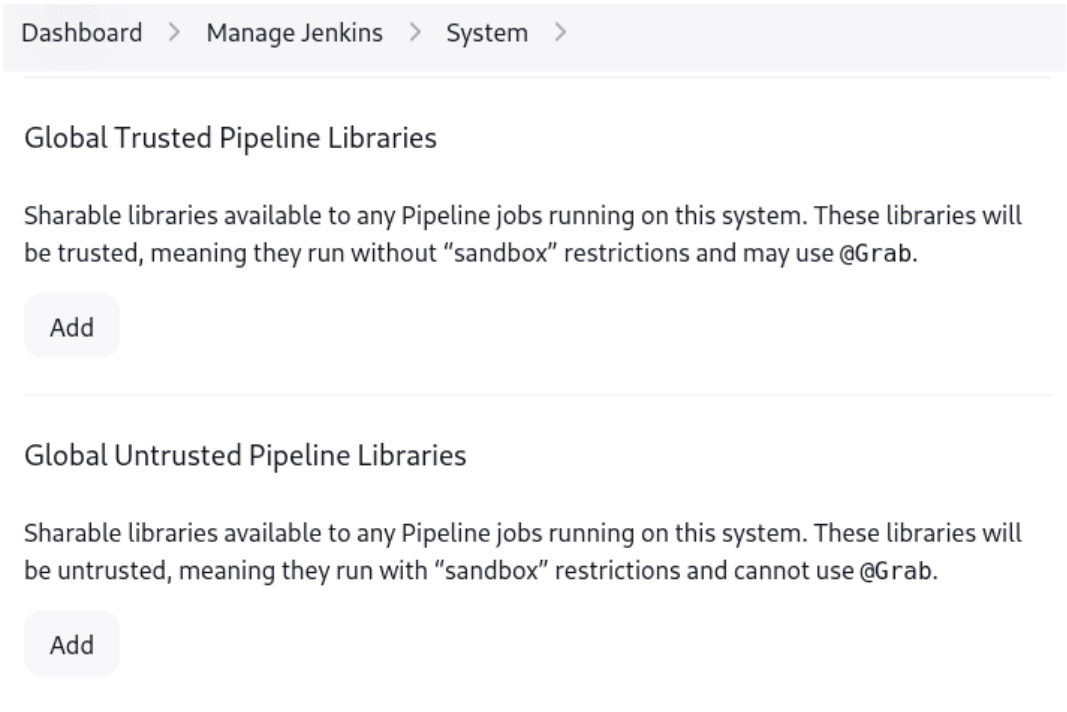
1. Go to **Jenkins Dashboard -> Manage Jenkins -> Credentials -> System -> Global Credentials**.
2. Add **Docker Hub credentials** with **ID: docker**.

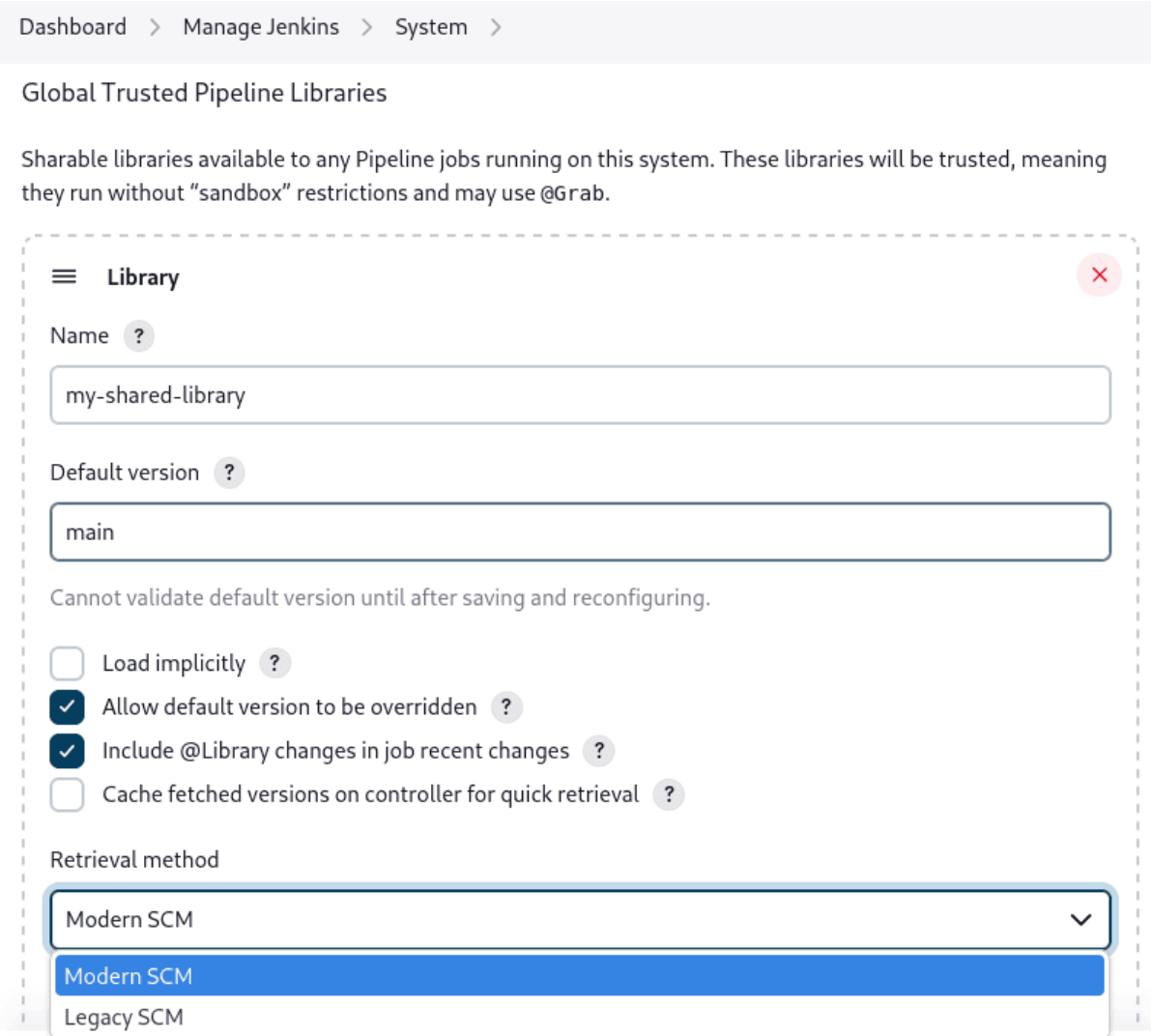




12.2 Add Jenkins Shared Library

1. Navigate to **Manage Jenkins -> Configure System -> Global Pipeline Library**.
2. Add:
    - **Name:** `my-shared-library`
    - **Default Version:** `main`
    - **Git URL: https://github.com/venkatraja1234/jenkins_shared_lib.git**

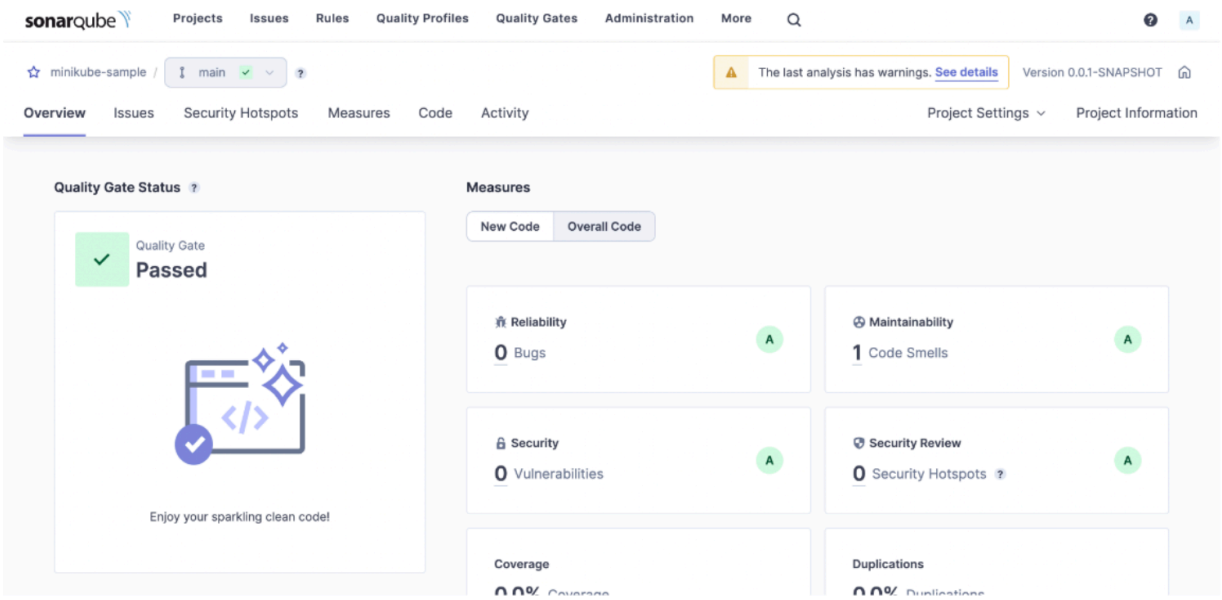




Step 13: Verify the CI/CD Pipeline

After running the pipeline, verify the following:

- Jenkins logs for build errors.
- **Trivy Scan** results for vulnerabilities.
- **SonarQube Dashboard** for code quality reports.


sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
A
minikube-sample
main
The last analysis has warnings. See details
Version 0.0.1-SNAPSHOT
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
Quality Gate Status
Quality Gate
Passed
Enjoy your sparkling clean code!
Measures
New Code
Overall Code
Reliability
0 Bugs
A
Maintainability
1 Code Smells
A
Security
0 Vulnerabilities
A
Security Review
0 Security Hotspots
A
Coverage
Duplications