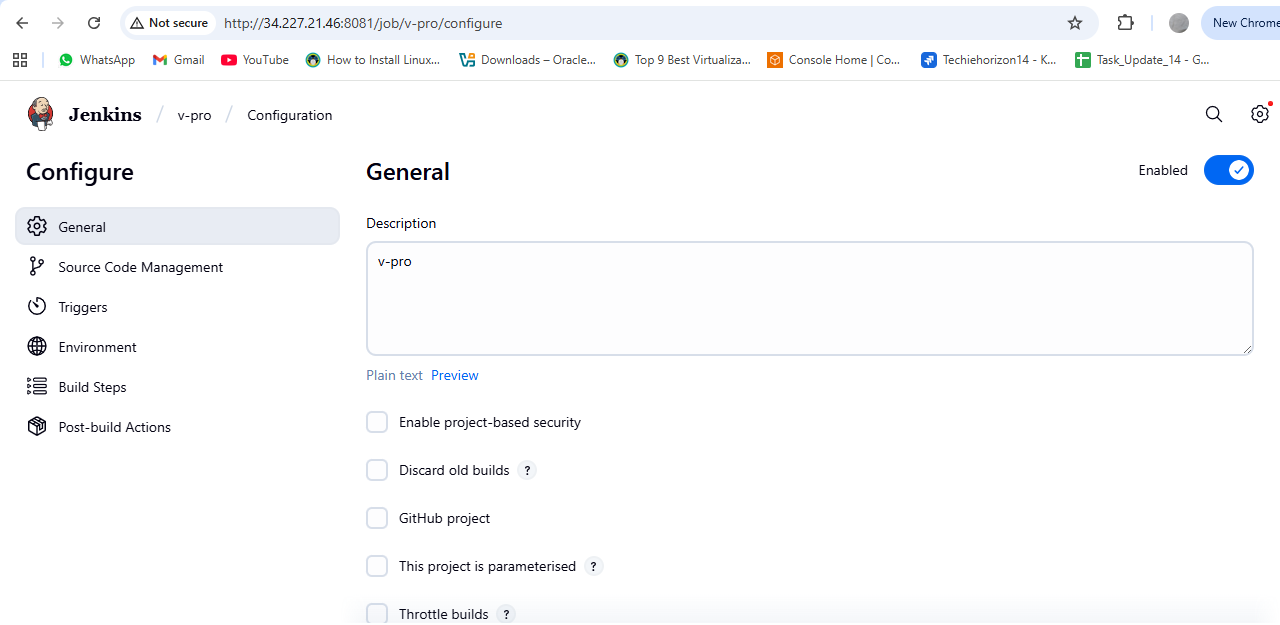
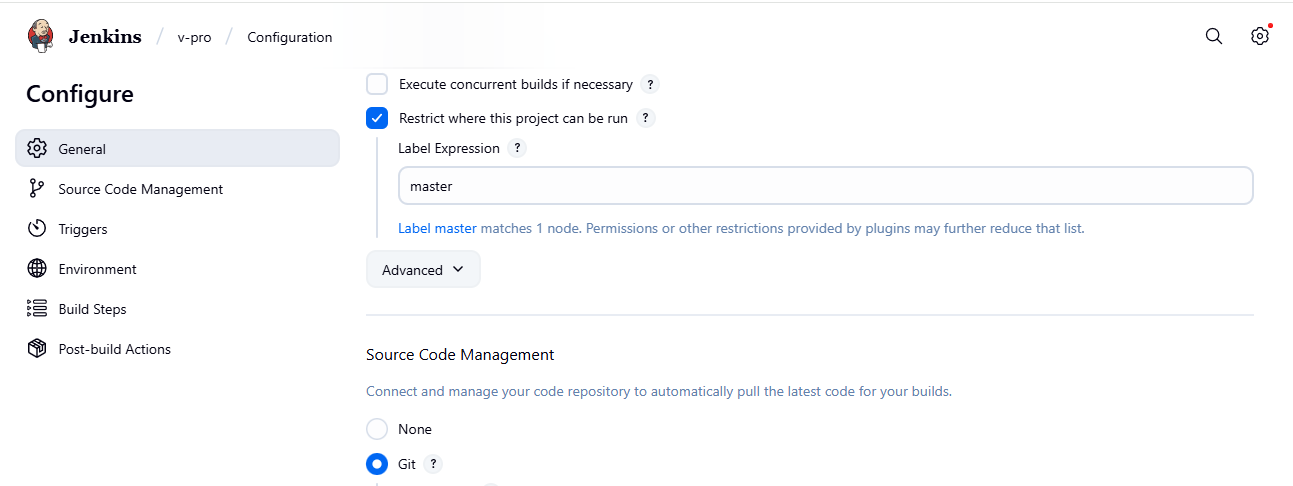
**1) Create one jenkins job using the below code and create three stages.**

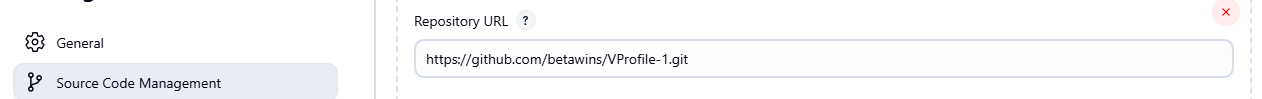
**stage1: Git clone to download the source code.**

**Copying the code from repo and configure the job.**

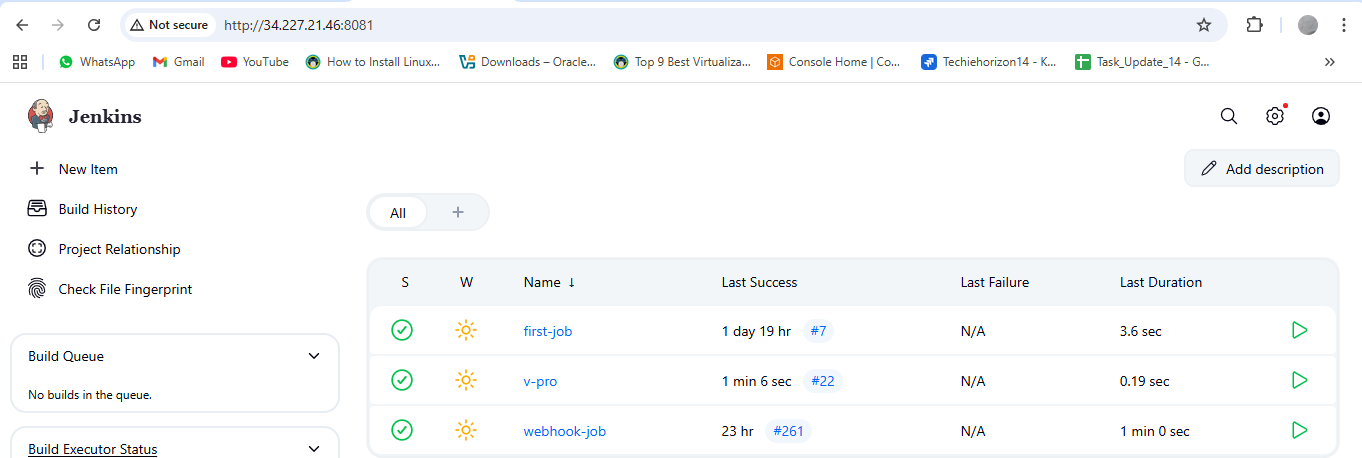
[**https://github.com/betawins/VProfile-1.git**](https://github.com/betawins/VProfile-1.git)

****

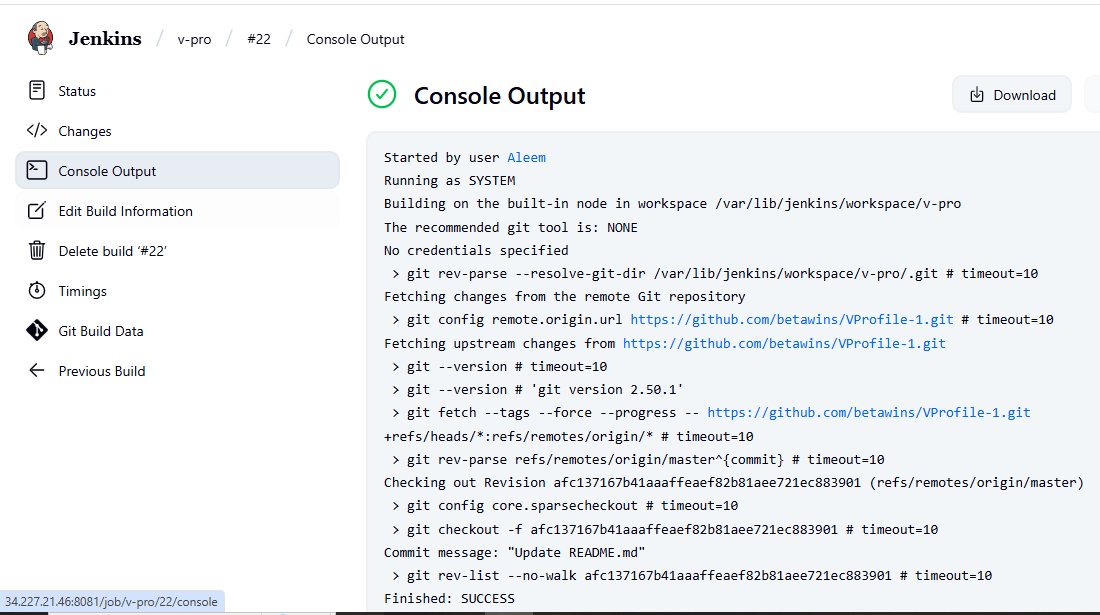
****

****

**save and build the job.**

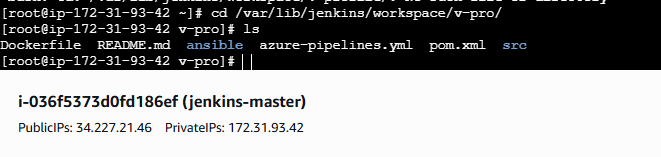
****

**Logs**

**Successfully clone the source code.**

**Then go to local machine and check V-profile available or not.**

**cd /var/lib/jenkins/workspace/v-pro/**

****

**stage2: Sonarqube Integration to check the quality of code**

**SonarQube Installation**

**SonarQube provides the capability to not only show health of an application but also to highlight issues newly introduced. With a Quality Gate in place, you can fix the leak and therefore improve code quality systematically.**

**### Prerequisites**

**1. EC2 instance with Java installed**

**2. Use t2.large with atleast 20gb memory to run sonarqube.**

**1. MySQL Database Server or MyQL RDS instance.**

**### Installation**

**Install java 1.8 version**

**```sh**

**yum install java-1.8\***

**```**

**Add mysql rpm Repository**

**```sh**

**yum update**

**sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm**

**sudo yum localinstall mysql57-community-release-el7-11.noarch.rpm**

**rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022**

**sudo yum install mysql-community-server**

**sudo systemctl start mysqld.service**

**```**

**Start MySQL and Enable Start at Boot Time**

**```sh**

**systemctl start mysqld**

**systemctl enable mysqld**

**```**

**Check if mysql is running or not**

**```sh**

**netstat -na | grep 3306**

**```**

**Configure the MySQL Root Password**

**You will see default MySQL root password**

**```sh**

**grep 'temporary' /var/log/mysqld.log**

**```**

**Login to mysql using the default password**

**```sh**

**mysql -u root -p**

**```**

**Now replace the default password with a new and strong password**

**```sh**

**ALTER USER 'root'@'localhost' IDENTIFIED BY 'Admin@123';**

**flush privileges;**

**```**

**Test Using new password**

**```sh**

**mysql -u root -p**

**```**

**Download stable SonarQube version from below website.**

**- Website: https://www.sonarqube.org/downloads/**

**- Note: This Article written for SonarQube6.0**

**Download & unzip SonarQube 6.0**

**```sh**

**# cd /opt**

**# wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.6.zip**

**# unzip sonarqube-6.6.zip**

**# mv /opt/sonarqube-6.6 /opt/sonar**

**```**

**Login to mysql**

**```sh**

**mysql -u root -p**

**```**

**Create a new sonar database**

**```sh**

**CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8\_general\_ci;**

**```**

**Create a local and a remote user**

**```sh**

**CREATE USER sonar@localhost IDENTIFIED BY 'Sonar@123';**

**CREATE USER sonar@'%' IDENTIFIED BY 'Sonar@123';**

**```**

**Grant database access permissions to users**

**```sh**

**GRANT ALL ON sonar.\* TO sonar@localhost;**

**GRANT ALL ON sonar.\* TO sonar@'%';**

**```**

**check users and databases**

**```sh**

**show databases;**

**SELECT User FROM mysql.user;**

**FLUSH PRIVILEGES;**

**QUIT**

**```**

**So for you have configured required database information on mysql. Let’s Jump back to your EC2 instance and enable SonarQube properties file to connect his Database.**

**### ON EC2 Instance**

**Edit sonar properties file to uncomment and provide required information for below properties.**

**- File Name: /opt/sonar/conf/sonar.properties**

**- sonar.jdbc.username=`sonar`**

**- sonar.jdbc.password=`Sonar@123`**

**- sonar.jdbc.url=jdbc:mysql://`localhost:3306`/sonar?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance&useSSL=false**

**- sonar.web.host=`0.0.0.0`**

**- sonar.web.context=`/sonar`**

**# Sonar version 7 and higher cannot be run using root user,please switch to any other user and change the permissions to sonar directory and start sonar.**

**Create new user**

**```sh**

**# useradd sabear**

**# passwd sabear**

**```**

**Add user sabear to sudoers file**

**```sh**

**vi /etc/sudoers**

**## Same thing without a password**

**# %wheel        ALL=(ALL)       NOPASSWD: ALL**

**sabear          ALL=(ALL)       NOPASSWD: ALL**

**```**

**Restart the sshd service**

**```sh**

**# systemctl restart sshd**

**```**

**Switch to newly created user**

**```sh**

**# sudo su sabear**

**```**

**Start SonarQube service**

**```sh**

**# cd /opt/sonar/bin/linux-x86-64/**

**# ./sonar.sh start**

**```**

**##### Run SonarQube as a default service**

**Implement SonarQube server as a service**

**```sh**

**Copy sonar.sh to etc/init.d/sonar and modify it according to your platform.**

**# sudo cp /opt/sonar/bin/linux-x86-64/sonar.sh /etc/init.d/sonar**

**# sudo vi /etc/init.d/sonar**

**```**

**Add below values to your /etc/init.d/sonar file**

**```sh**

**Insert/modify below values**

**SONAR\_HOME=/opt/sonar**

**PLATFORM=linux-x86-64**

**WRAPPER\_CMD="${SONAR\_HOME}/bin/${PLATFORM}/wrapper"**

**WRAPPER\_CONF="${SONAR\_HOME}/conf/wrapper.conf"**

**PIDDIR="/var/run"**

**```**

**Start SonarQube server**

**```sh**

**# service sonar start**

**```**

**SonarQube application uses port 9000. access SonarQube from browser**

**```sh**

**http://<EC2\_PUBLIC\_IP>:9000/sonar**

**```**

**Default credentials for sonarqube.**

**Username : admin**

**Password: admin**

**###  NOTES**

**1) Check whether you enabled port 9000 in EC2 instance security group**

**### Important Points:**

**1) mysql port number : 3306**

**2) sonarqube port: 9000**

**3) sonarqube logs : /opt/sonar/logs**

**4) We can find four different logs**

**a)access.log**

**b)es.log aka elastic search**

**c)sonar.log**

**d)web.log**

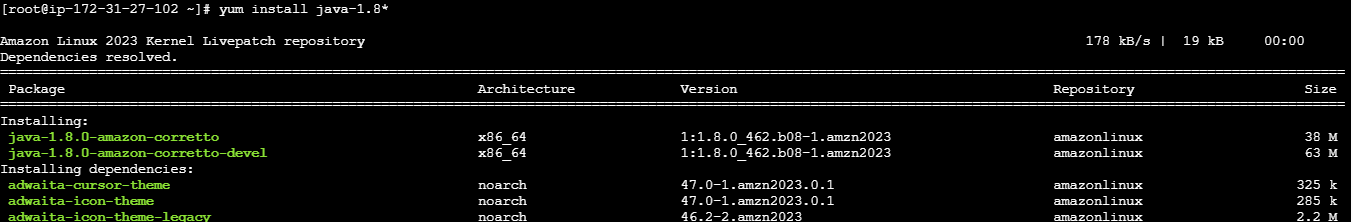
**### Videos for reference:**

**### sonarqube installation:**

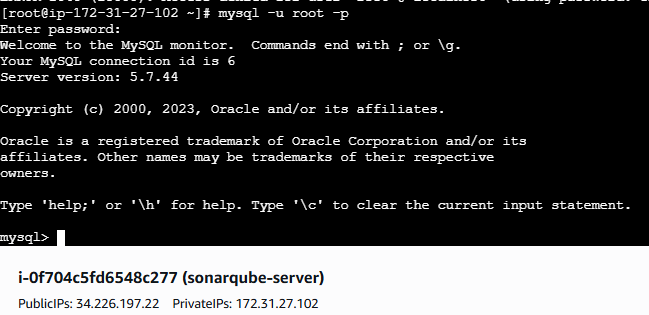
**https://www.youtube.com/watch?v=zRQrcAi9UdU**

**#### mysql rds instance aws:**

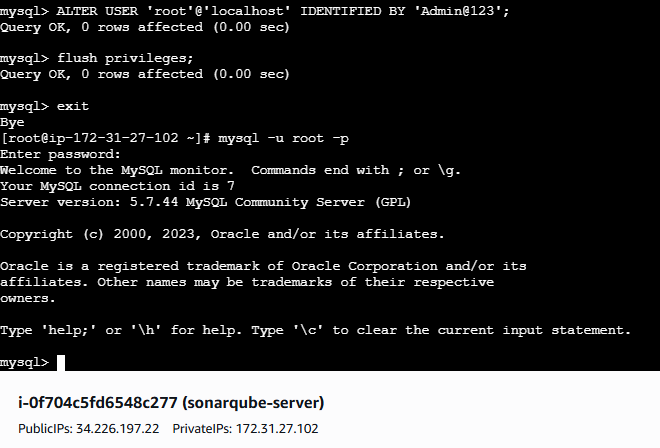
[**https://www.youtube.com/watch?v=vLaW6b441x0**](https://www.youtube.com/watch?v=vLaW6b441x0)

****

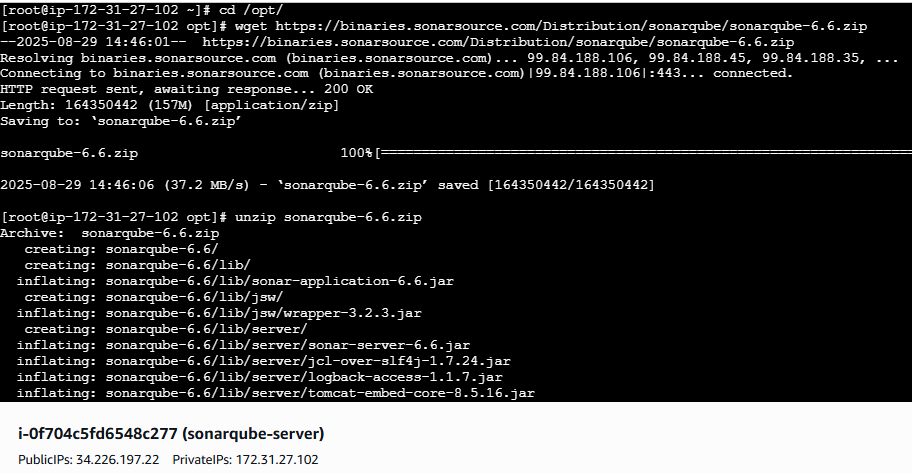
**Enter a auto generated password to enter in the Mysql**

****

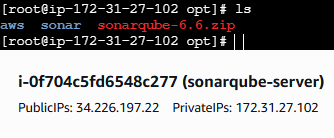
**Enter a new password to test and Exit from Mysql**

****

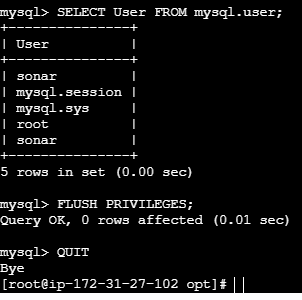
**Now install sonarqube and extract directory**

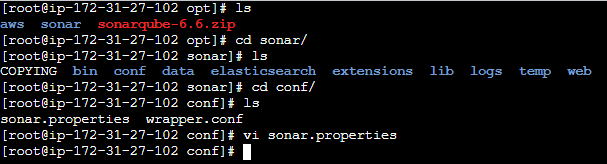
****

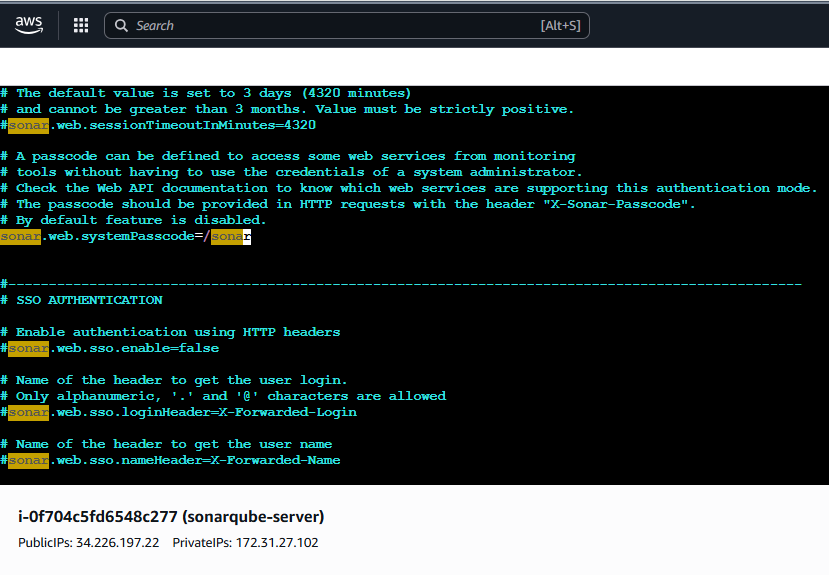
**Renaming (optional)**

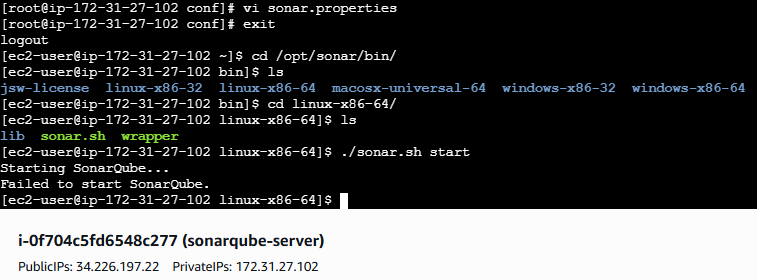
****

**Follow the list of commands**

****

****

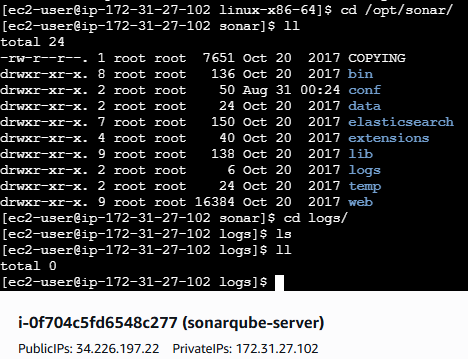
****

****

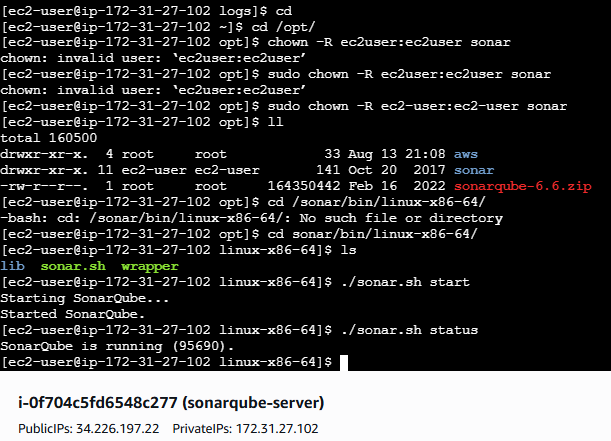
**Fail to start sonarQube.**

**Need to troubleshoot**

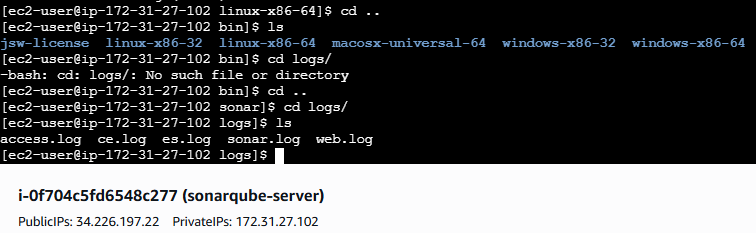
**Check logs weather any error available or not. If not then need to change because the owner is “root user”**

****

**In this step we are changing ownership from root to ec2- user.**

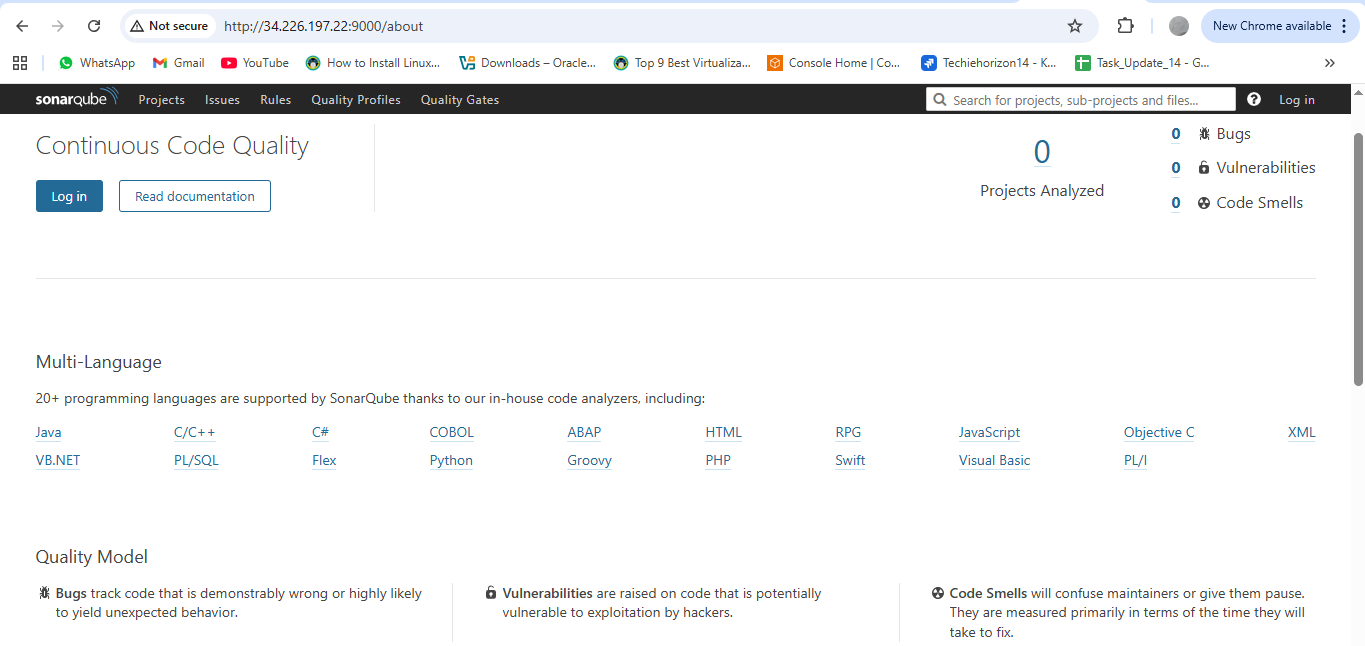
****

**If some thin wrong always need to check sonar logs other having some issue.**

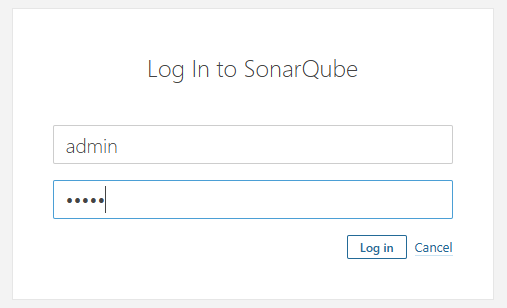
****

**If logs are available then go to web browser and using public IP with port no 9000**

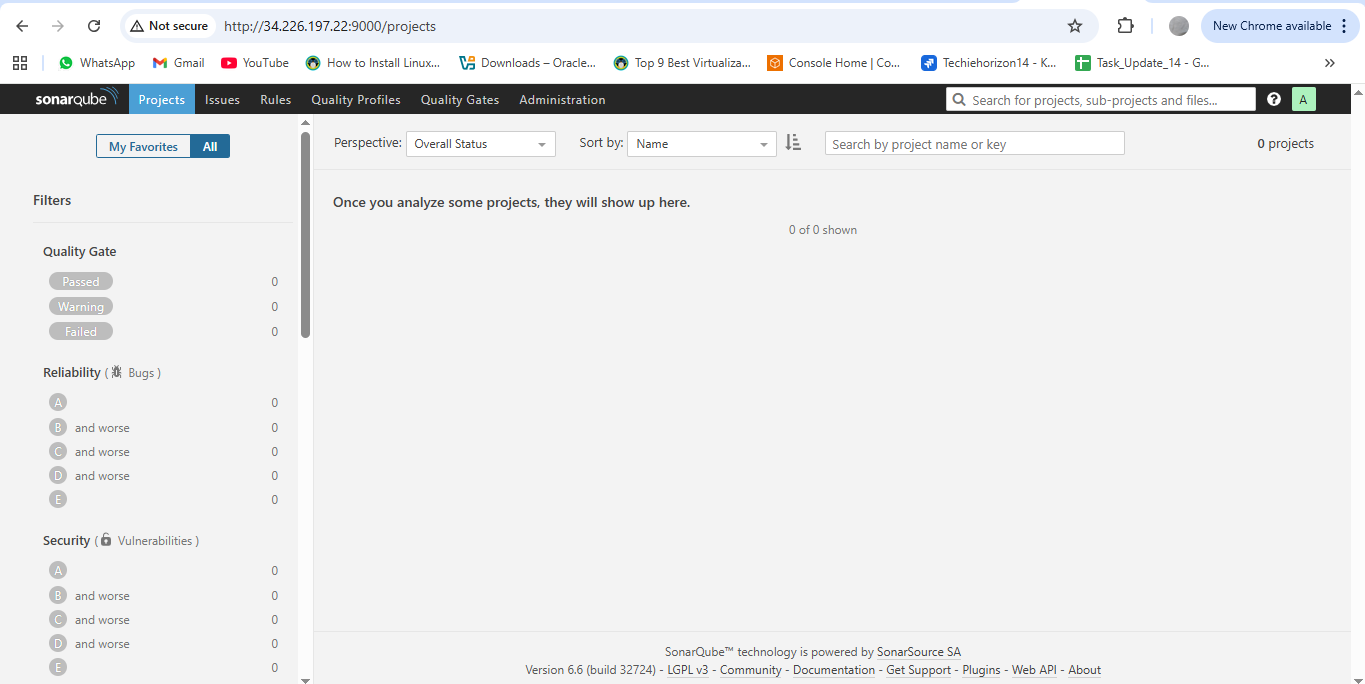
**Check for the conformation sonarqube is available on same port.**

****

**After successfully started sonarqube then login using user: admin and password: admin**

****

**We have successfully created sonarqube and can see the Dashbosrd.**

****

**SonarQube Integration with Jenkins**

**Integration SonarQube server with Jenkins is necessary to store your reports. Follow below steps to enable that.**

**### Follow this in \*\*[YouTube](https://www.youtube.com/watch?v=k-3krTRuAFA)\*\***

**### Prerequisites**

**1. SonarQube Server \*\*[Get Help here](https://www.youtube.com/watch?v=zRQrcAi9UdU)\*\***

**1. Jenkins Server  \*\*[Get Help here](https://www.youtube.com/watch?v=M32O4Yv0ANc)\*\***

**### Implementation**

**Login to Jenkins server and install sonarqube scanner.**

**- SonarQube scanner URL : https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner**

**- Package : https://sonarsource.bintray.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-3.2.0.1227-linux.zip**

**-**

**```sh**

**# wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.6.2.2472-linux.zip**

**# unzip sonar-scanner-cli-4.6.2.2472-linux.zip**

**# mv sonar-scanner-4.6.2.2472-linux /opt/sonar\_scanner**

**```**

**Set SonarQube server details in sonar-scanner property file**

**- Sonar properties file: /opt/sonar\_scanner/conf/sonar-scanner.properties**

**- sonar.host.url=http://`<SONAR\_SERVER\_IP>`:9000**

**Login to Jenkins GUI console and install " SonarQube scanner" plugin**

**- `Manage Jenkins` > `Manage Plugins` > `Avalable` > `SonarQube scanner`**

**Configure SonarQube scanner home path**

**- `Manage Jenkins` > `Global Tool Configuration` > `SonarQube Scanner`**

**- Name  : `sonar\_scanner`**

**- SONAR\_RUNNER\_HOME : `/opt/sonar\_scanner`**

**Configure SonarQube server name and authentication token**

**- `Manage Jenkins` > `Configure Systems` > `SonarQube Servers`**

**- Name : `SonarQube`**

**- ServerURL : `http://<Sonarqube\_server>:9000/sonar`**

**- Server `authentication token`**

**To Get Authentication code follow below steps.**

**Login to SonarQube server as a admin  `My Account` > `Security` > `Generate Token`**

**Create a job to test SonarQube. Provide below sonar properties details in the job under build**

**-SCM:**

**```**

**- https://github.com/betawins/VProfile-1.git**

**```**

**- Build:**

**- `Invoke top-level Maven targets`**

**`Goals : clean install`**

**- `Execute SonarQube Scanner` > `Analysis properties`  (it is mandatary).**

**```sh**

**sonar.projectKey=Sabear**

**sonar.projectName=Sabear**

**sonar.projectVersion=1.0**

**sonar.sources=/var/lib/jenkins/workspace/$JOB\_NAME/src/**

**sonar.binaries=target/classes/com/visualpathit/account/controller/**

**sonar.junit.reportsPath=target/surefire-reports**

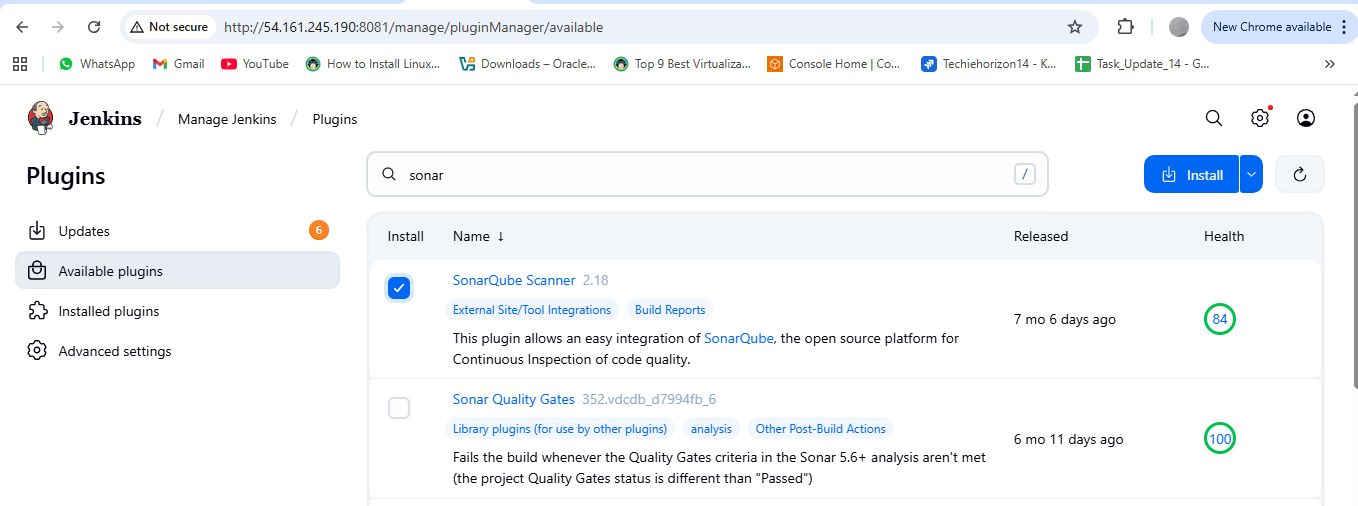
**sonar.jacoco.reportPath=target/jacoco.exec**

**sonar.java.binaries=src/main/java/com/visualpathit/account/**

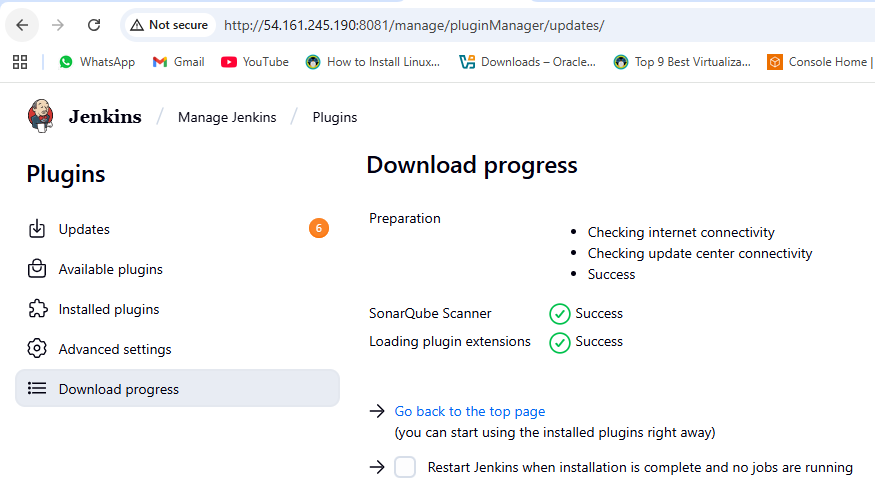
**```**

**Execute job to get analysis report.**

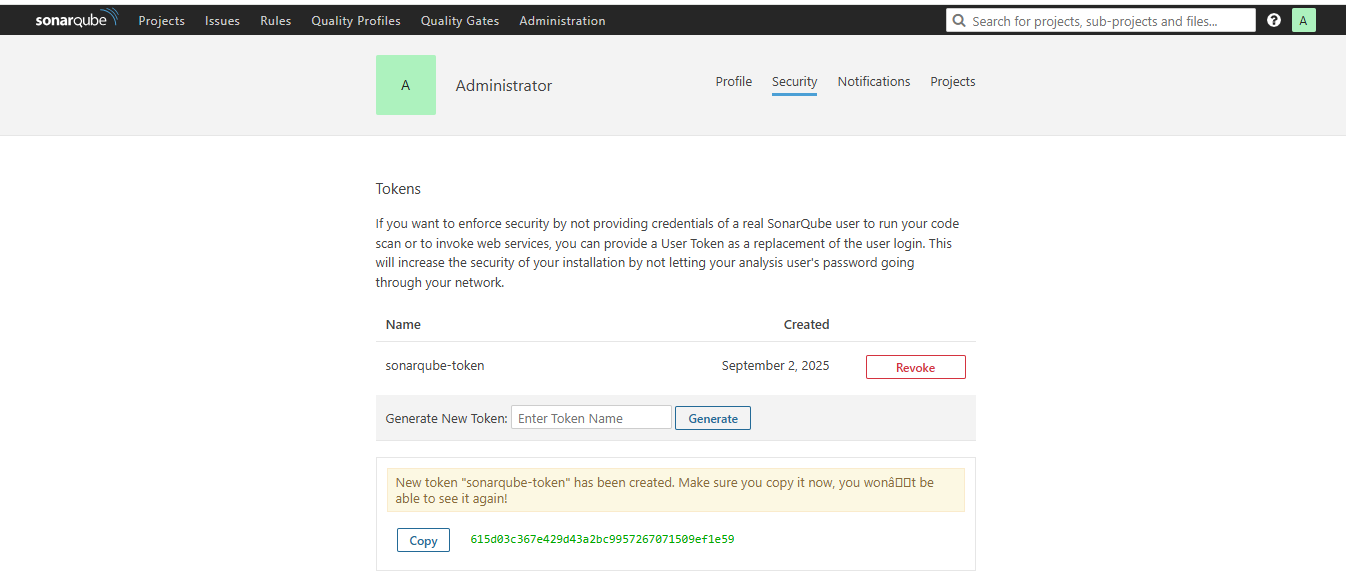
**Manage Jenkins > Plugins > Available Plugins > sonarqube > Install**

****

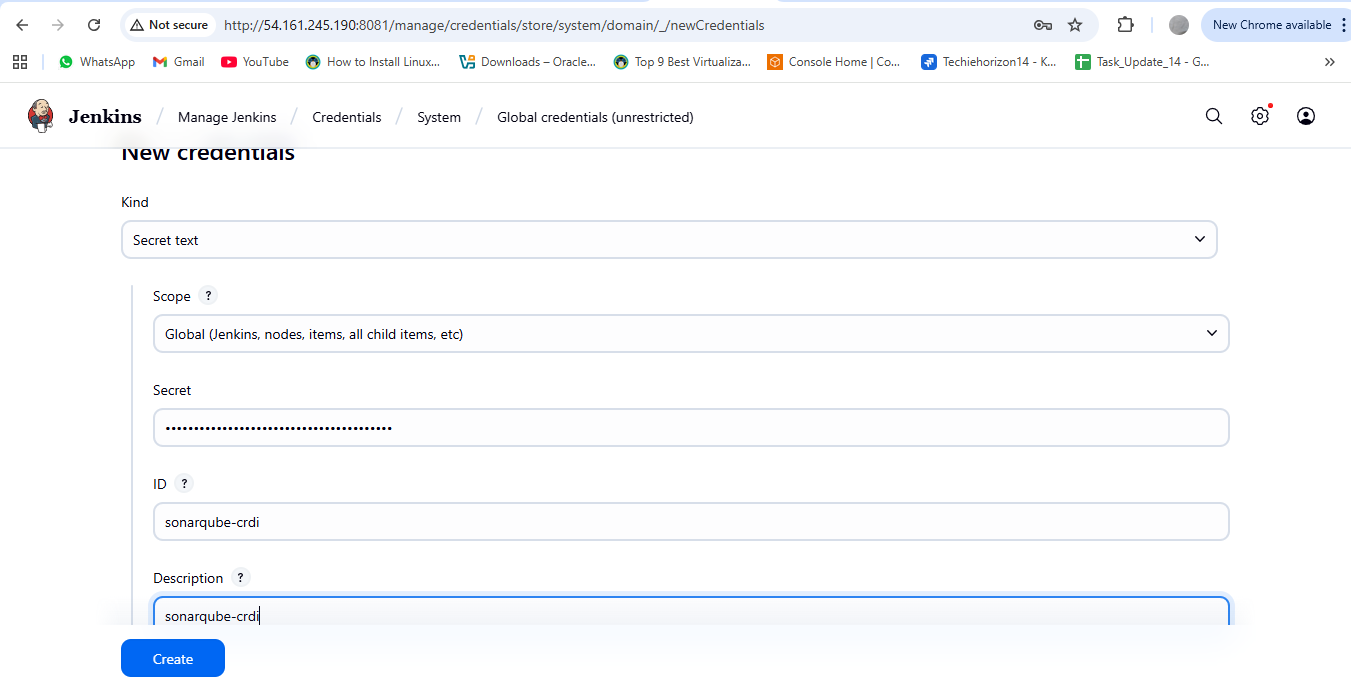
**Plugins installation has been done.**

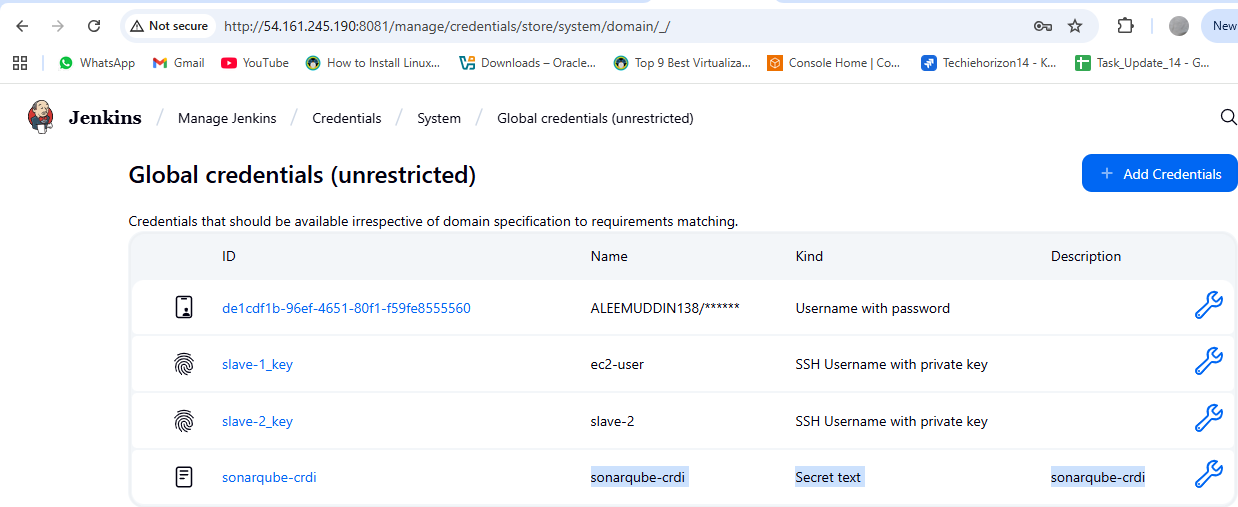
****

**Go to sonarqube and create token for Credentials**

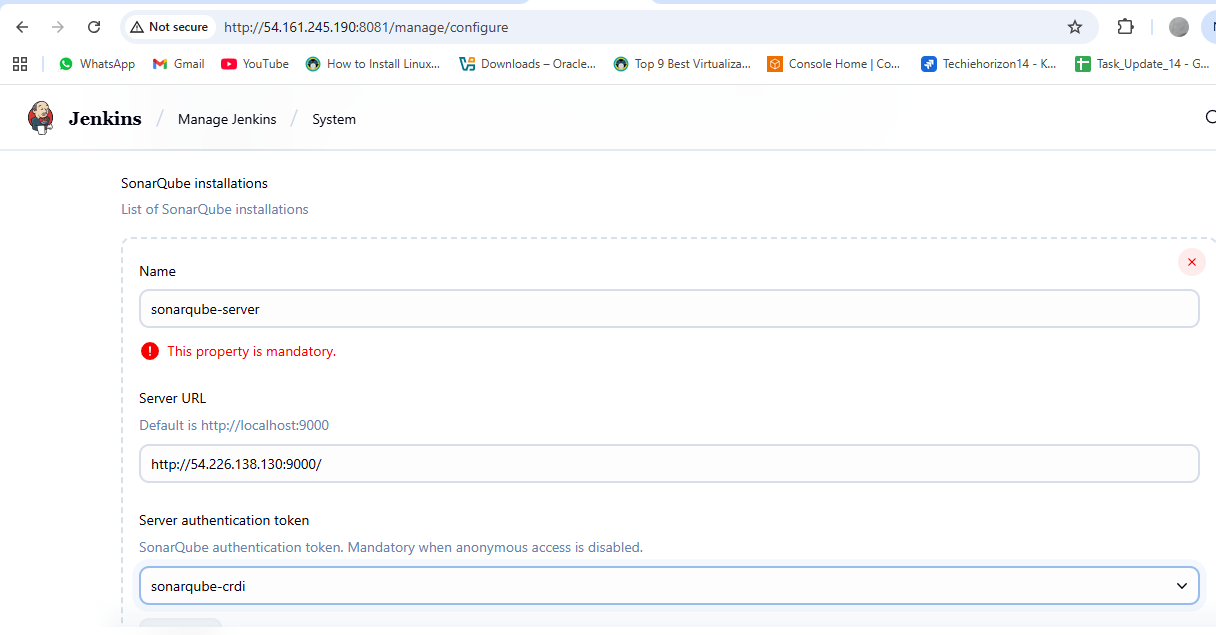
****

**Create Credentials**

****

****

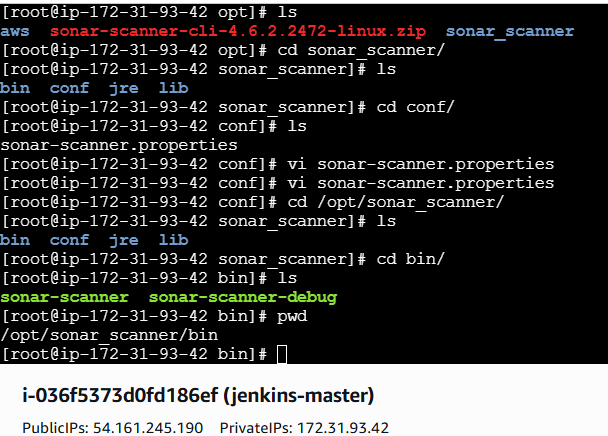
**Then add credentials to sonarqube installation at system configuration**

****

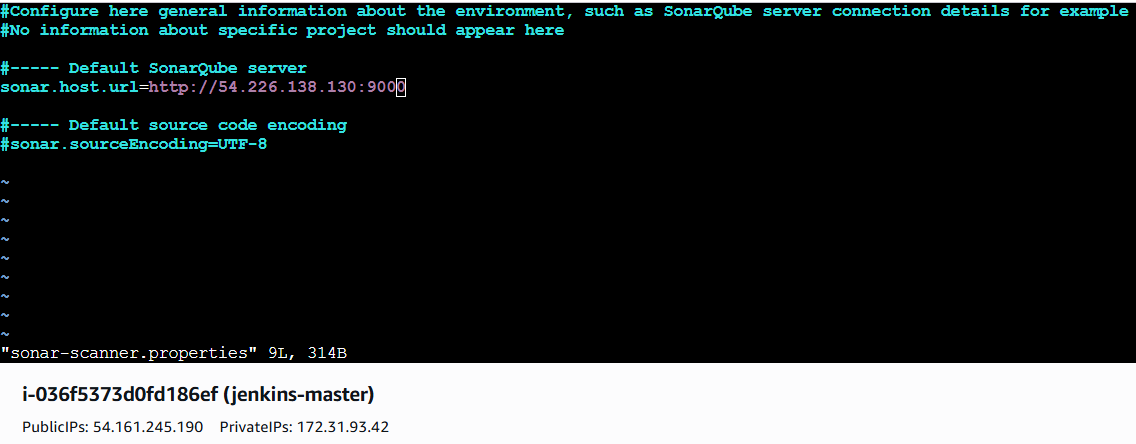
**Install sonar scanner in master server**

****

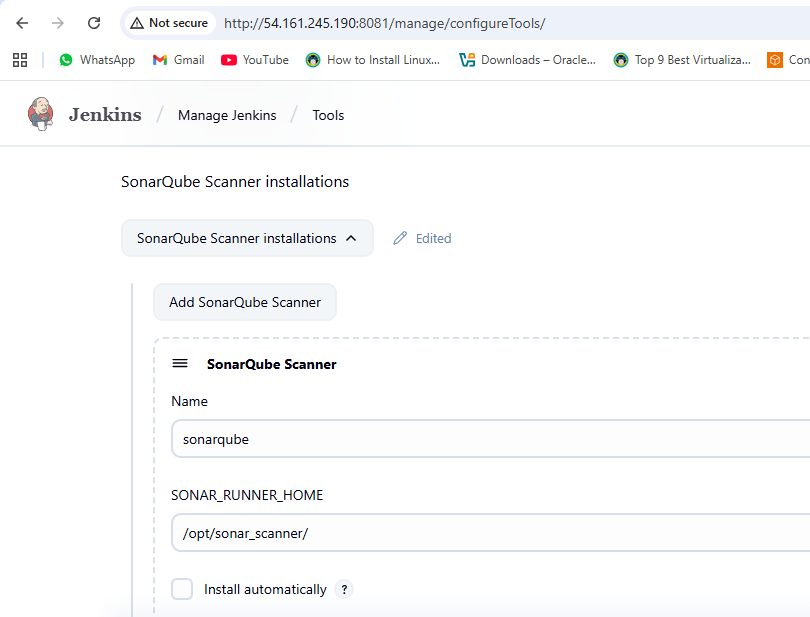
**Unzip the sonar scanner zip file**

****

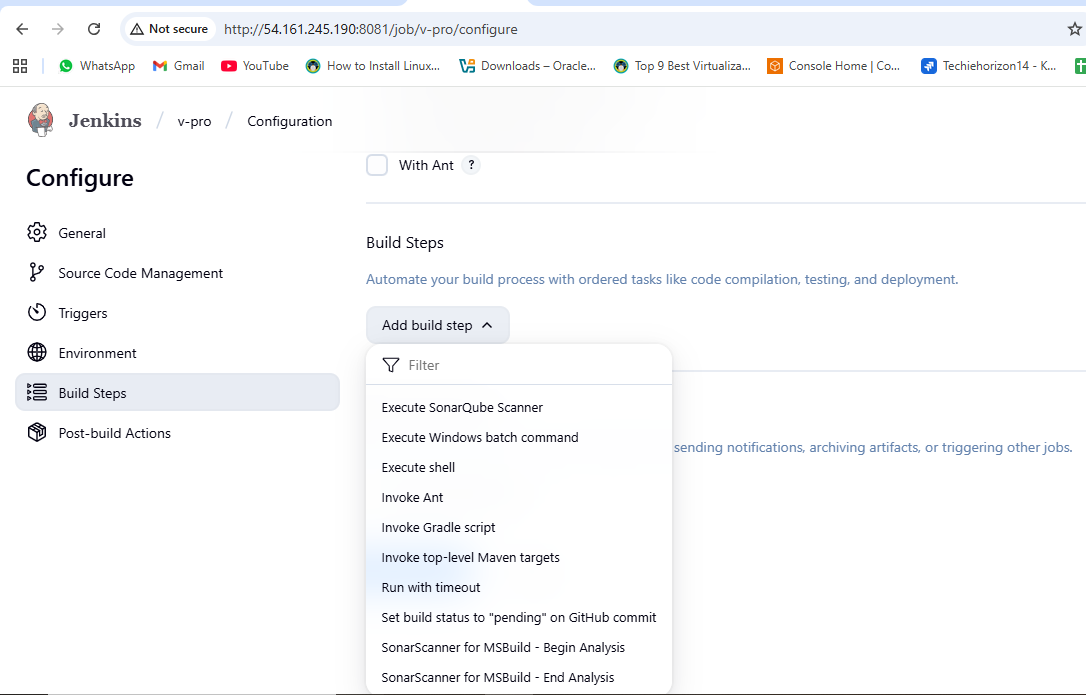
**Add IP in the soner scanner properties using vi sonar-scanner.properties**

****

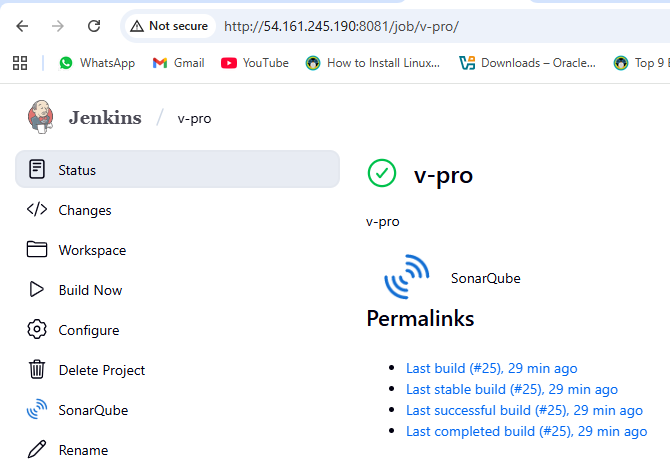
**Add sonar scanner runner path in tools**

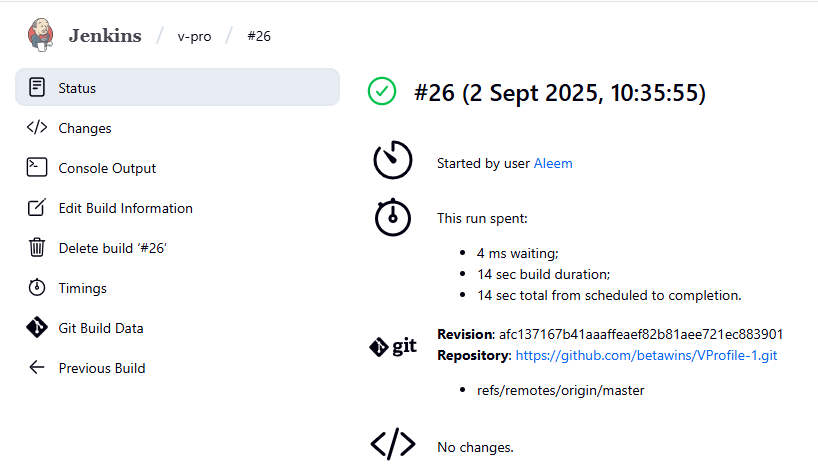
****

**Go to v-job configuration and add sonar scanner**

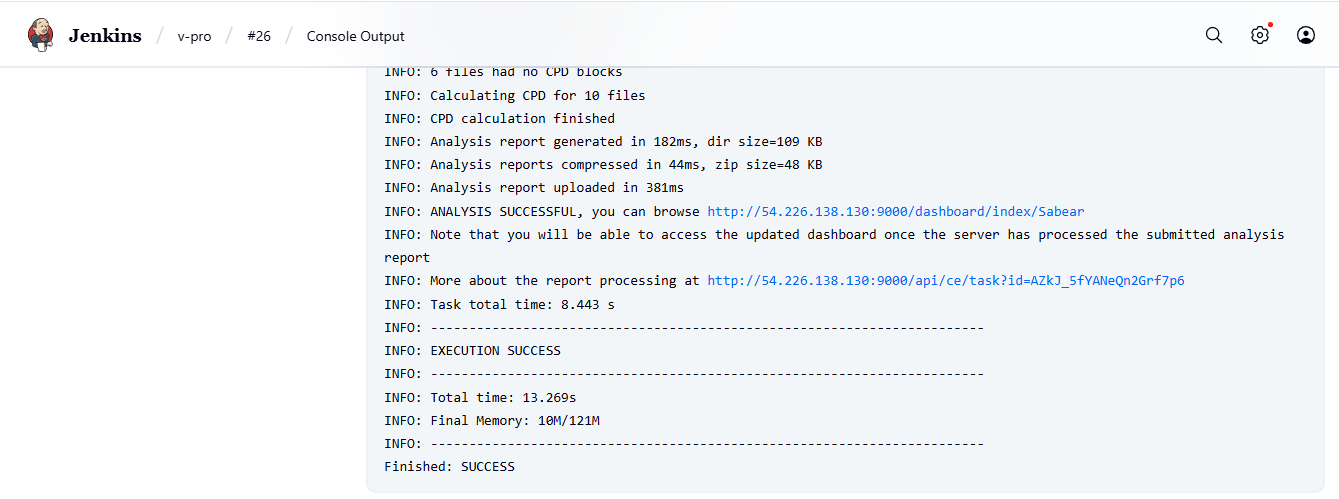
****

**Then build the job successfully**

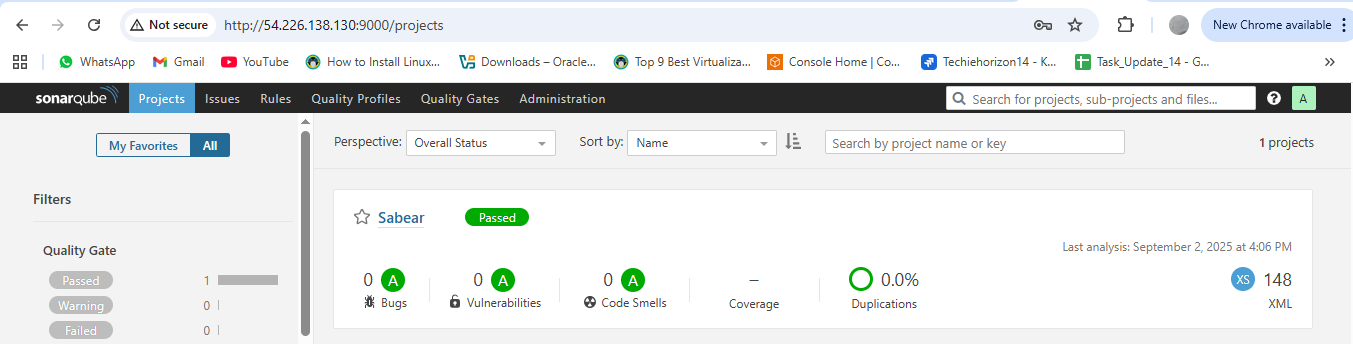
****

****

**We can see the analysis reports has been generated in the console output.**

****

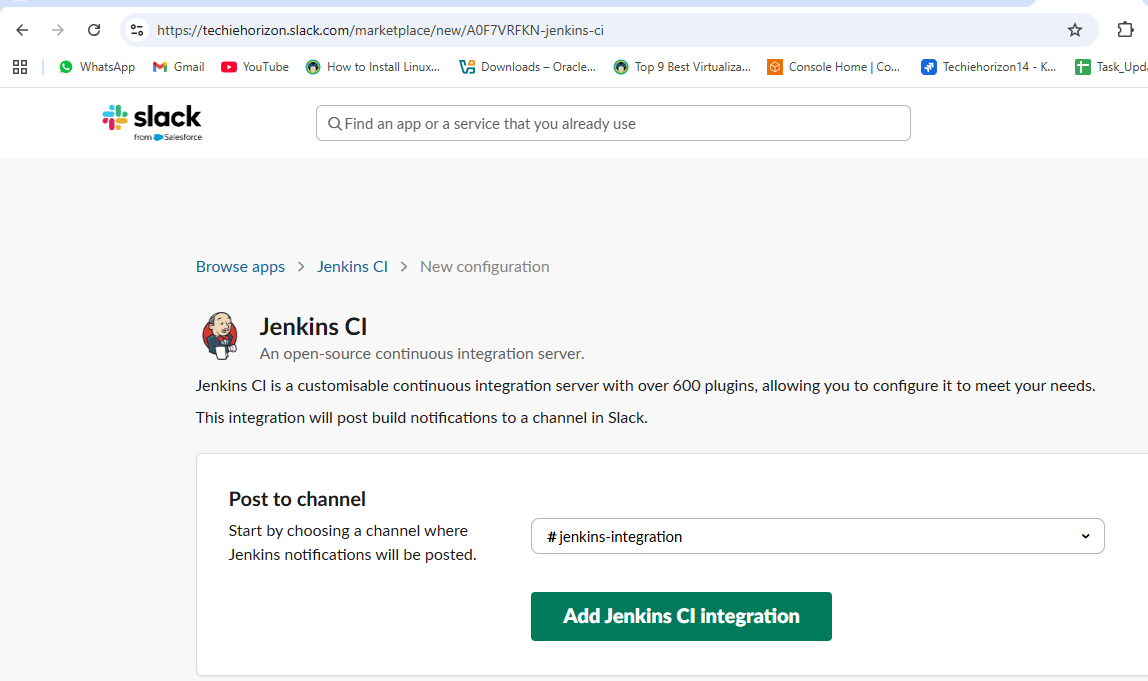
**Go to sonarqube server and can see one project has been created**

****

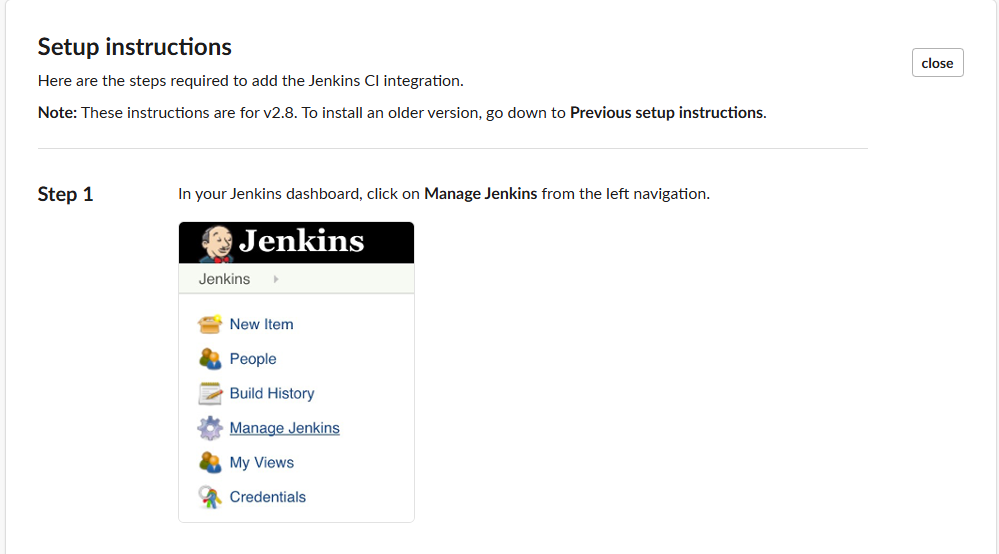
**stage3: Slack Integration to send the alerts to slack.**

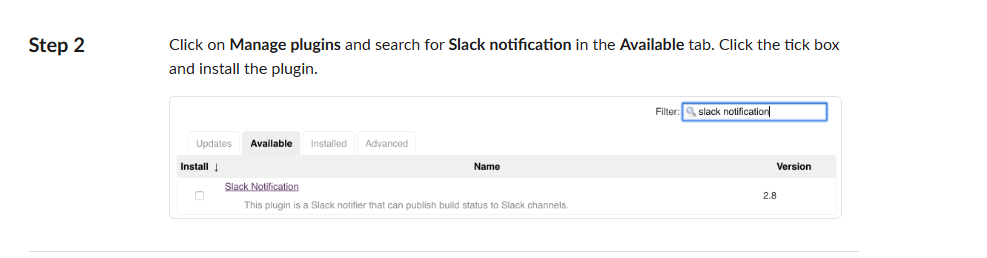
**URL:** [**https://github.com/betawins/VProfile-1.git**](https://github.com/betawins/VProfile-1.git)

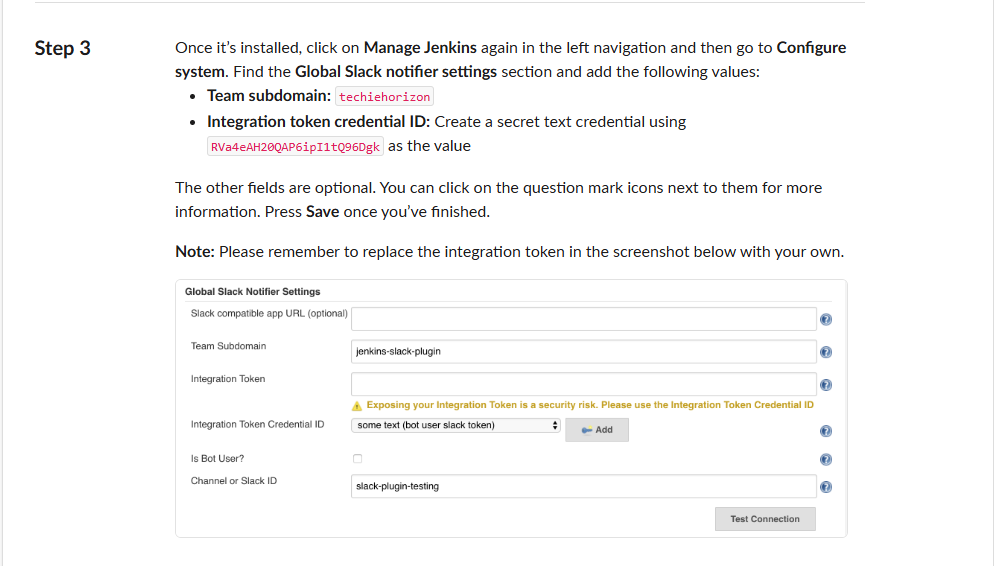
**Go to slack > Add aps > Jenkins CI > New configuration**

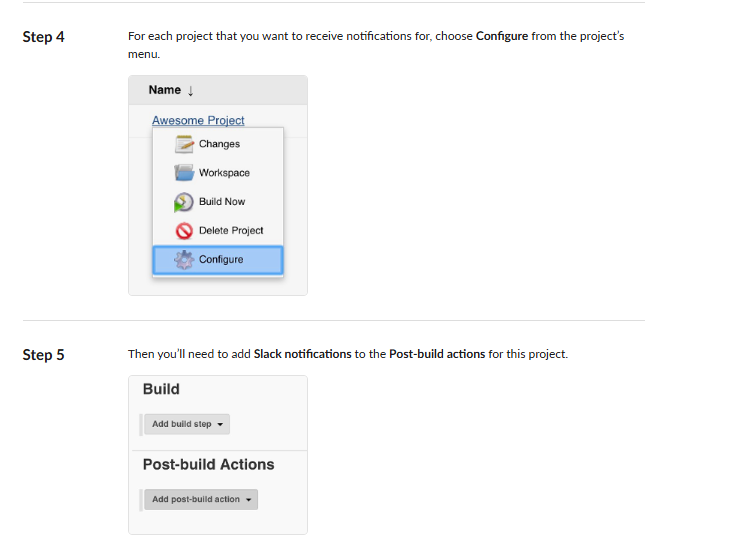
****

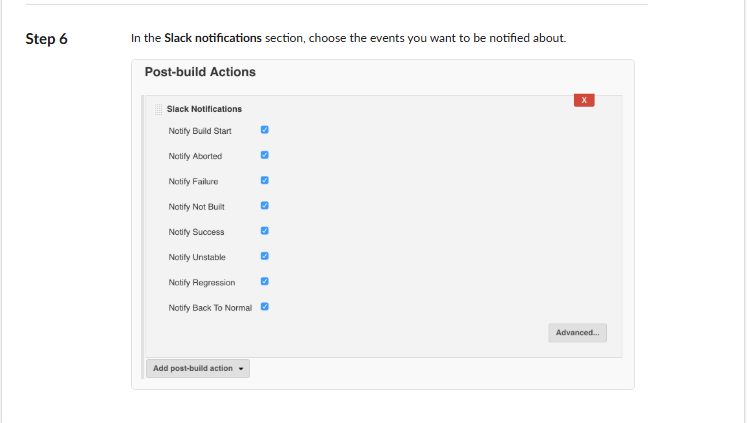
**Follow the below steps**

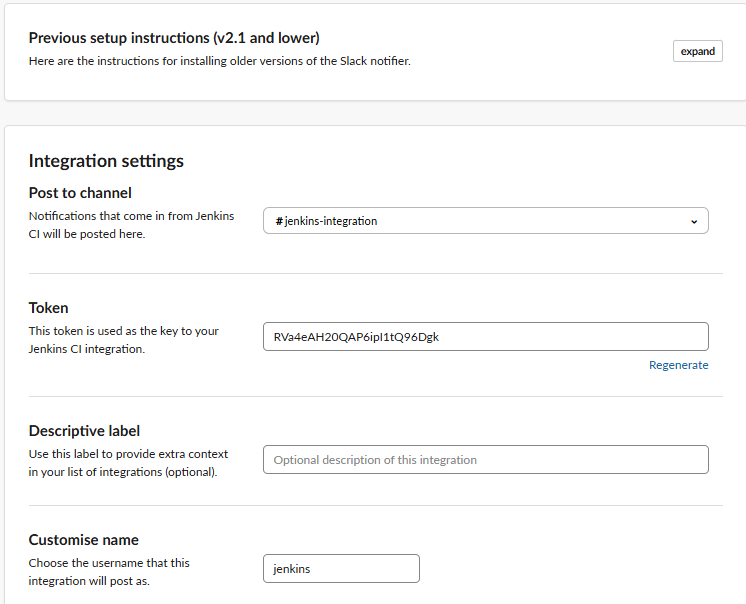
****

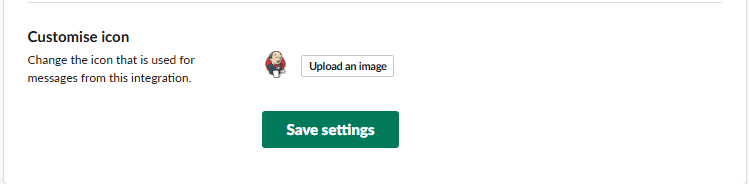
****

****

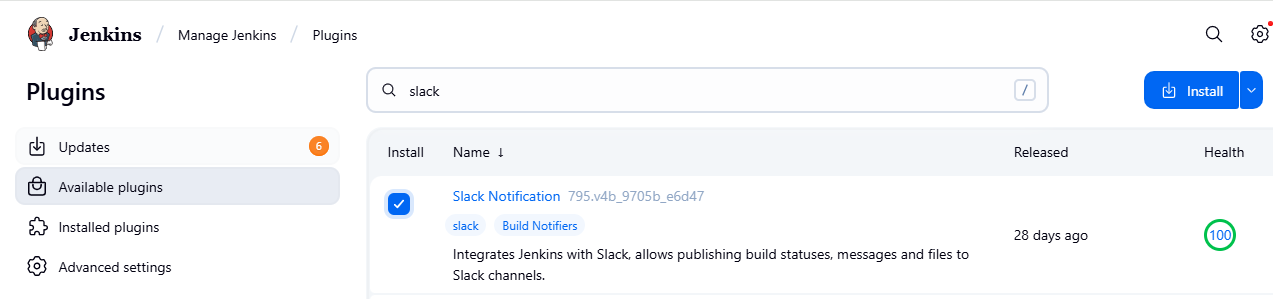
****

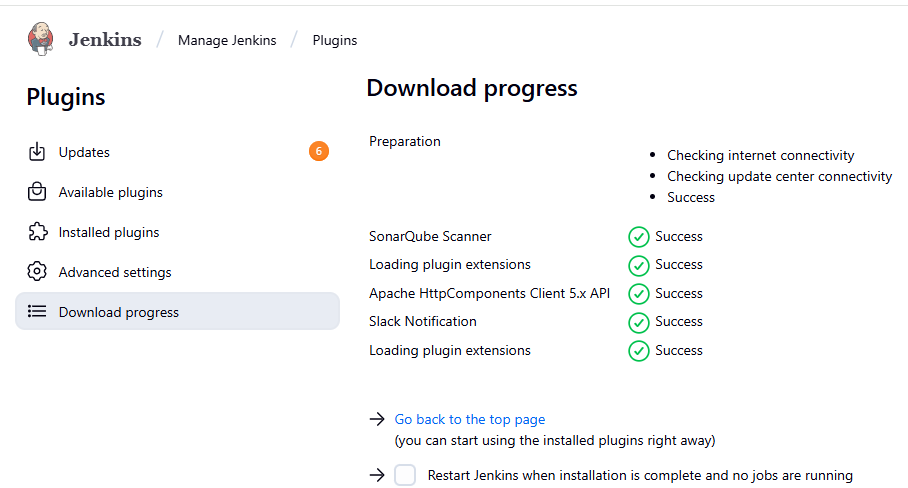
****

****

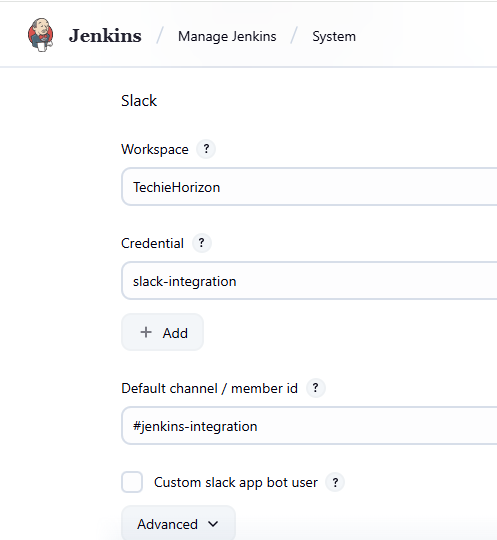
****

**Manage Jenkins > plugins > available plugins > slack notifications > Install**

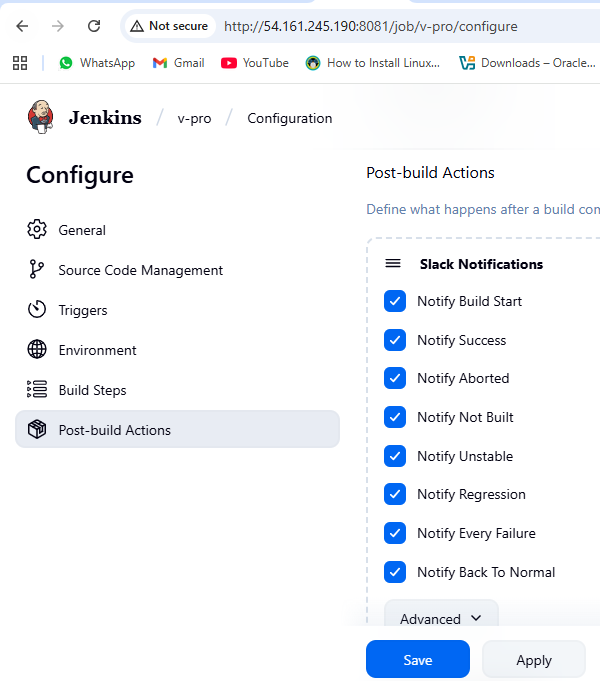
****

****

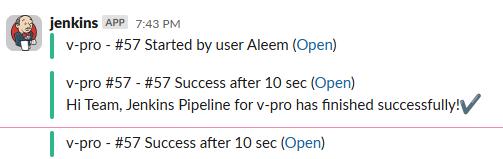
**Manage Jenkins > system > Fill all the details + create credentials as per the given token in the slack.**

****

**Go to job configuration and select all post build actions**

****

**Then in slack channel notification will reflect**

****

**2) 1) Create one jenkins job using the below code and create three stages.**

**stage1: Git clone to download the source code.**

**stage2: Sonarqube Integration to check the quality of code**

**stage3: Slack Integration to send the alerts to slack.**

**URL:** [**https://github.com/betawins/hiring-app.git**](https://github.com/betawins/hiring-app.git)

**In sonarqube EC2**

**yum -y install docker**

**systemctl start docker**

**docker pull sonarqube:9.9.4-community**

**Docker images**

**docker run -itd -p 9003:9000 sonarqube:9.9.4-community**

**Docker ps**

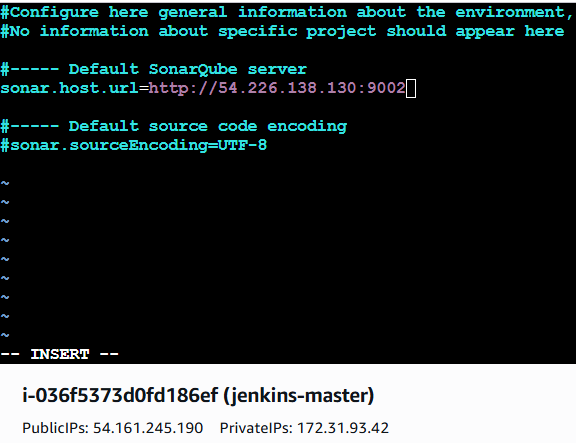
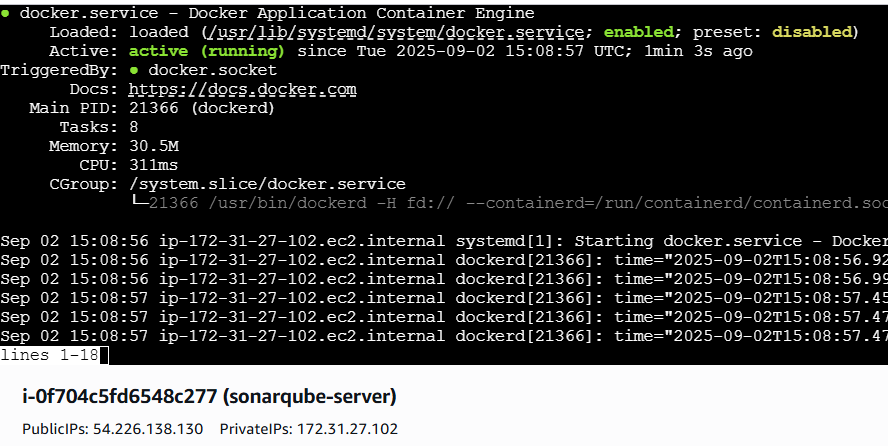
**Now go to jenkins server and change port info**

**cd /opt/sonar\_scanner/conf**

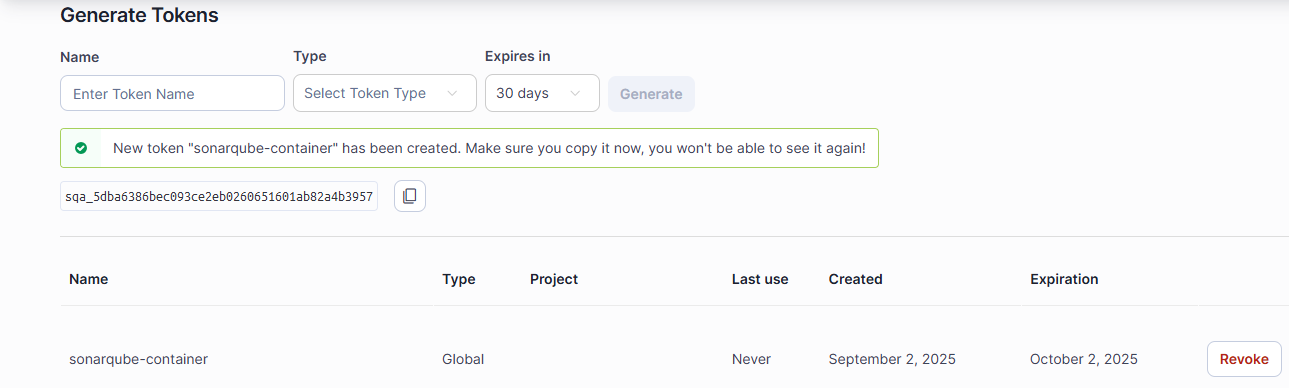
**Vi sonar-scanner.properties**

**sonar.host.url=http:// 54.226.138.130:9003**

**Manage jenkins → system change port info of sonarqube and add credentials**

**Go to sonarqube and create token**

**Create and copy the token from sonar server**

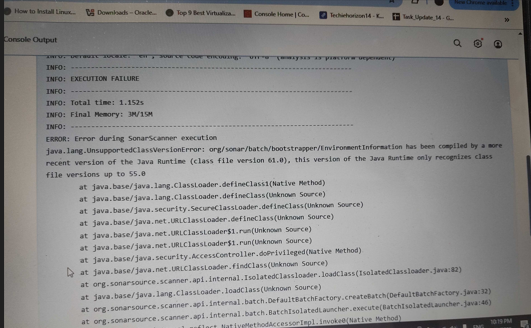
****

**Update url & create credential by pasting token**

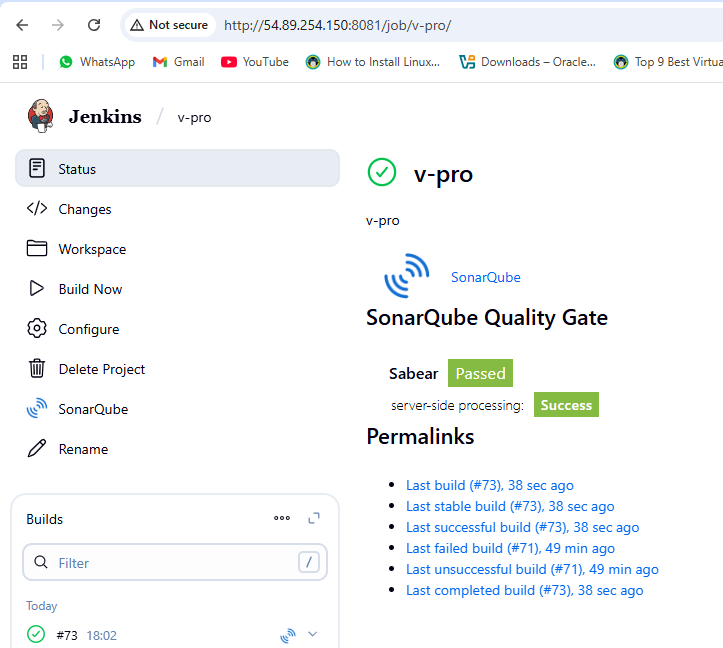
****

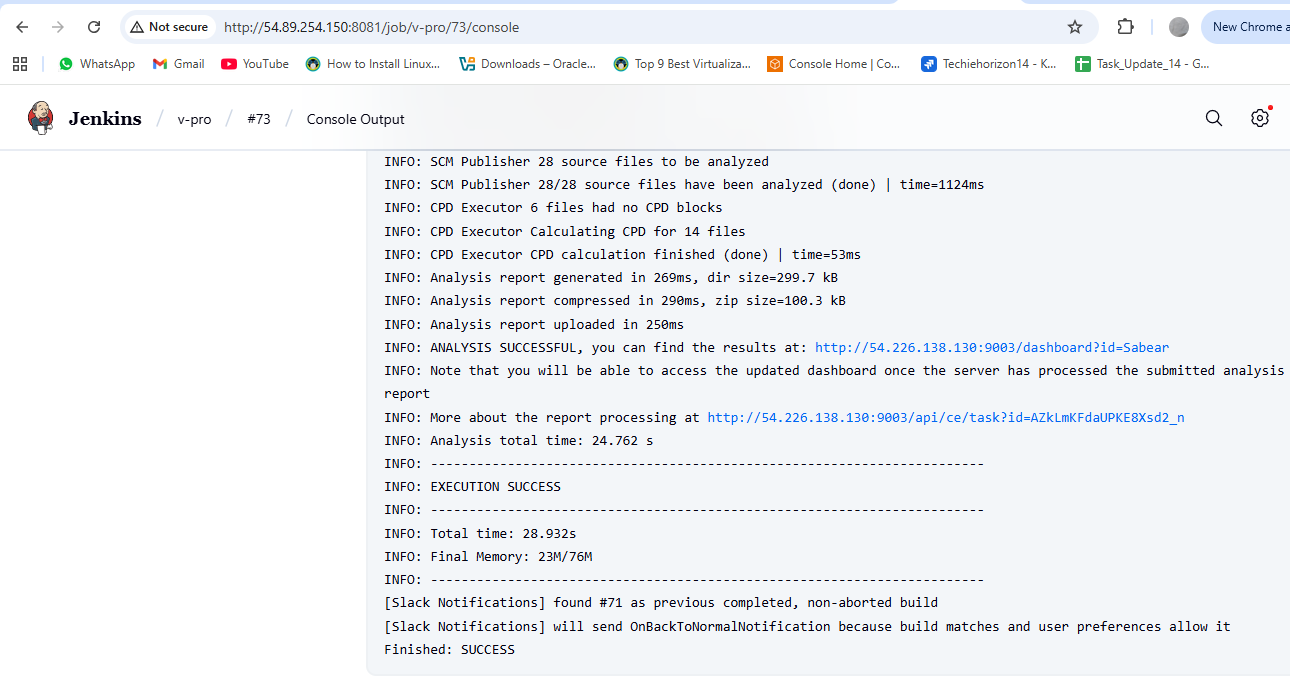
**During build the job facing with error related to java-17 not supported to sonar image**

**Then updated with latest version (sonarqube:9.9.4-community)**

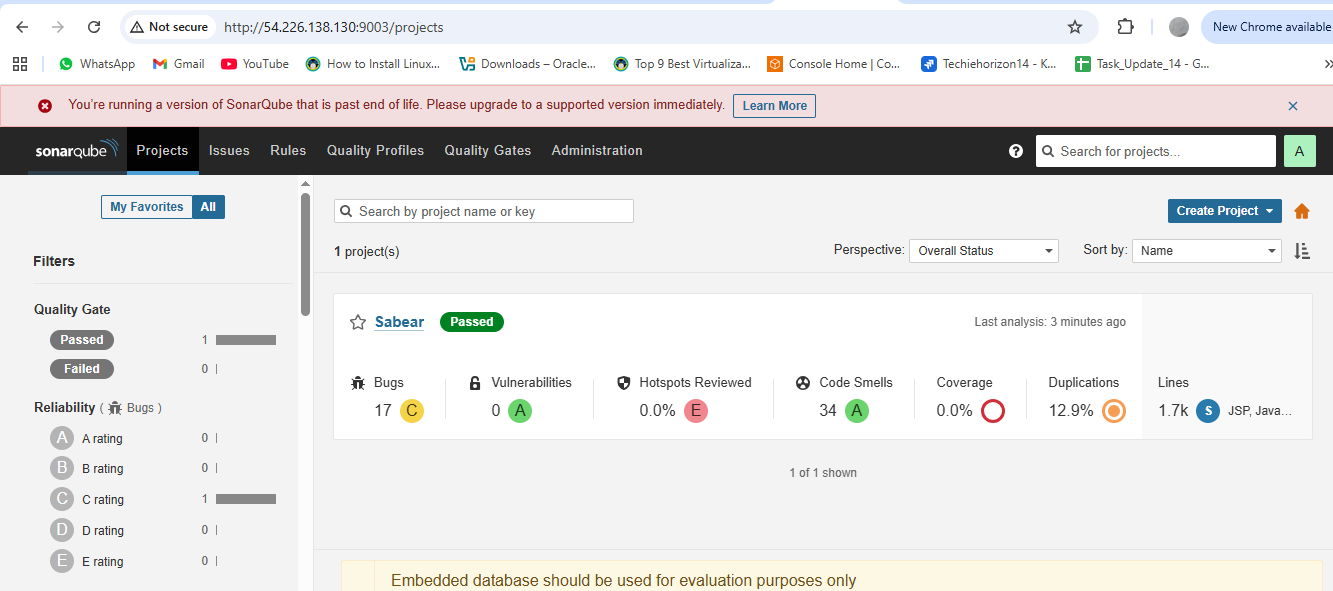
****

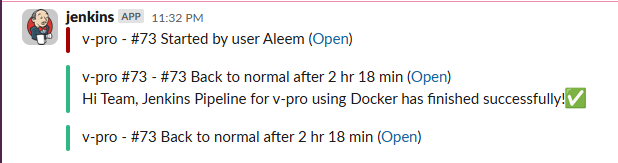
**Then build the job succssfully**

****

****

**We can see one project reflect in the server and passed the quality.**

****

****