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/VProfile-1.git>

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>

GitHub**GitHub**

[**GitHub - betawins/VProfile-1: The purpose of the research project is for the students to learn how to make CI,CD pipeline by using jenkins and scripting techniques taught in class.**](https://github.com/betawins/VProfile-1.git)

The purpose of the research project is for the students to learn how to make CI,CD pipeline by using jenkins and scripting techniques taught in class. - GitHub - betawins/VProfile-1: The purpose o... (49 kB)

<https://github.com/betawins/VProfile-1.git>

GitHub**GitHub**

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

Contribute to betawins/hiring-app development by creating an account on GitHub. (33 kB)

…………………………………………………………………………………………………………………………………………………………….

Previous Points Recall:

Whenever there was sample code or source code related to our **Application** that was been stored in Github, From Git hub it has been pulled to Jenkins and in Jenkins we are integrating Maven, Sonarq which will help us to compile the source code and do the check on the quality and then it will be pull to the nexus.

Nexus is something call Artifactory which will help us to store our packages.

So once this package is ready, we are going to deploy this in the tomcat or apache , nginx or web logic or web spear or JBOSS.

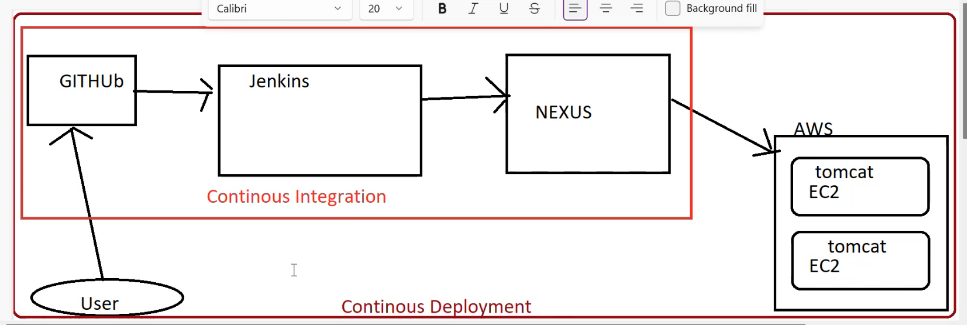
So in our case we can assume as tomcat.

Where is out tomcat installed, Inside EC2 and our EC2 is available in AWS. Again in AWS we have the highly available architrecture what we are seen in the previous classes.

CI is – GIT Hub – Jenkins – Nexus

CD is – Once our package is ready in the nexus, it’s automatically deploy on to the End Environment

The complete below diagram comes under the CD



CI : It’s mean that we are storing our artifacts in any of the centre location. we are not deploying the application, we are just integrating maven, sonarq, xyz..tools for the quality and storing the artiactory.

CD depolyment : It actually means that, the application is automatically getting deployed without any manual intervention.

CD Contineous Delivery : Means it will only make sure that, the end environment is up and ready, Means the environment where we want to deploy is ready. But it will not deploy the application.

……………………………………………………………………………………………………………………………………………….

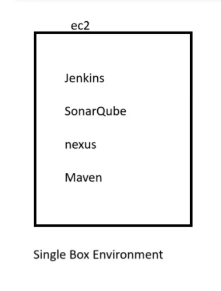
Video Starts:

In this session will discuss about different stages in our CI/CD pipeline, I you have a sample code then what needs to be done with that code, How can we create pipeline and how can we create packages out of that code.

Interview: What are the different stages available in your pipeline?

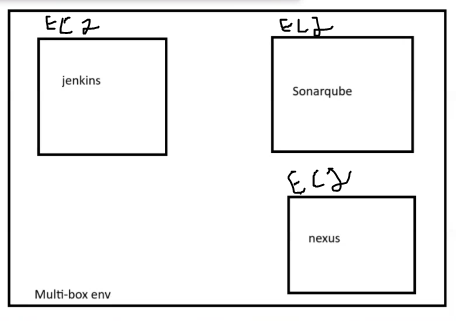
**We can have 2 different types of configurations:**

**Single Box Environment**: In one server ec2 we are installing all the software’s which are required for our pipeline and we are trying to execute our pipeline. This is not recommended for production environment. This is only for learning purpose. Only for learnig it’s ok.



2. **Multi-box environment**:

Where every ec2 serve is responsible for specific tool or store any specific code or any packages

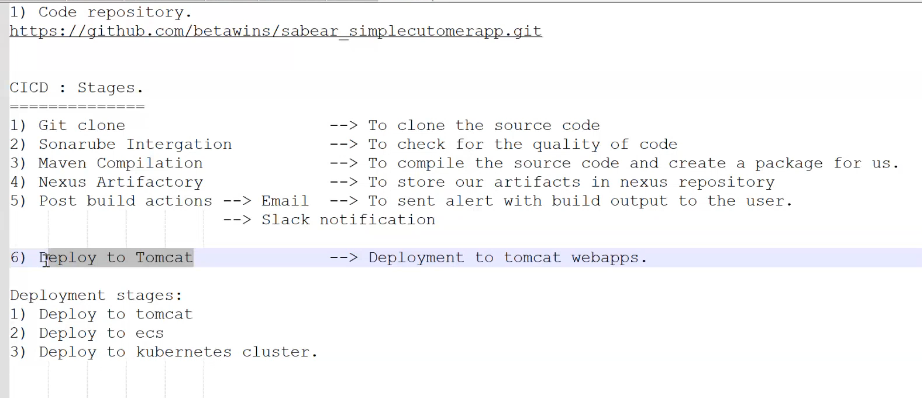


Scenario: You got one code in repository which is developed by developer and advised to create CI/CD pipeline from scratch.

1. Code repository

<https://github.com/noorabdul03/sabear_simplecutomerapp.git>

1. We need to create different different stages in our pipeline by using the particular code.



Deploy to tomcat , ECS has out dated , previously we were using deploy to tomcat and ecs.

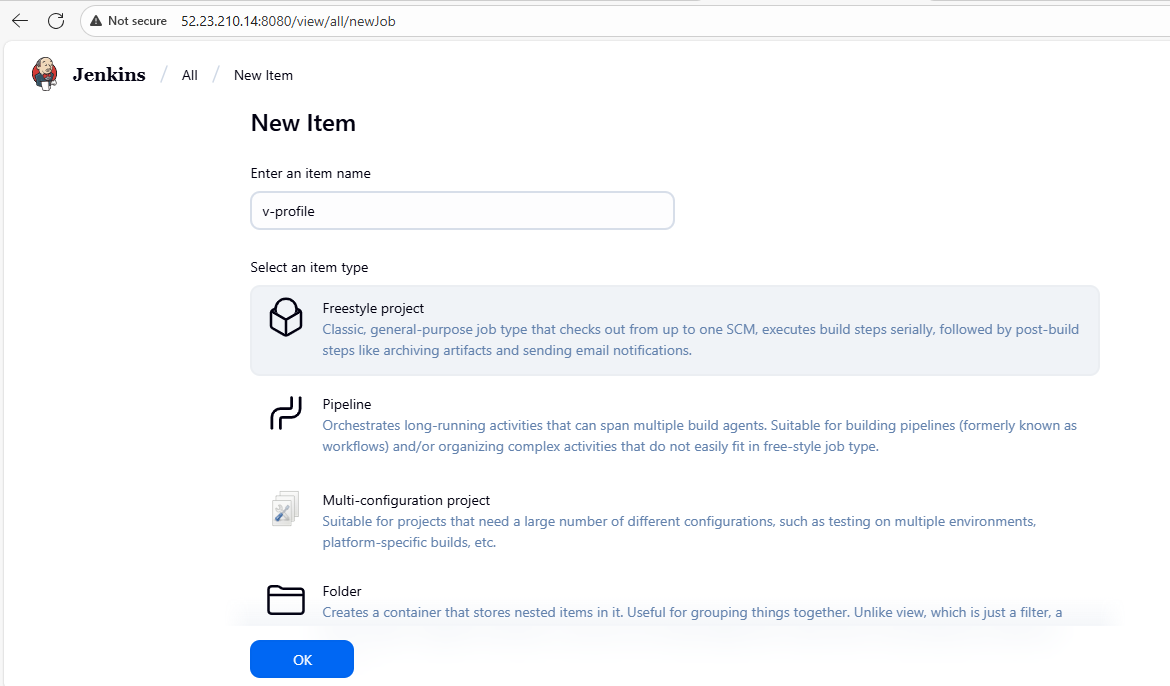
Currently using deploy to kubernetes cluster.

1. Start Jenkins server GUI
2. Create first stage among six stages
3. Git clone : First stage

Click on New item

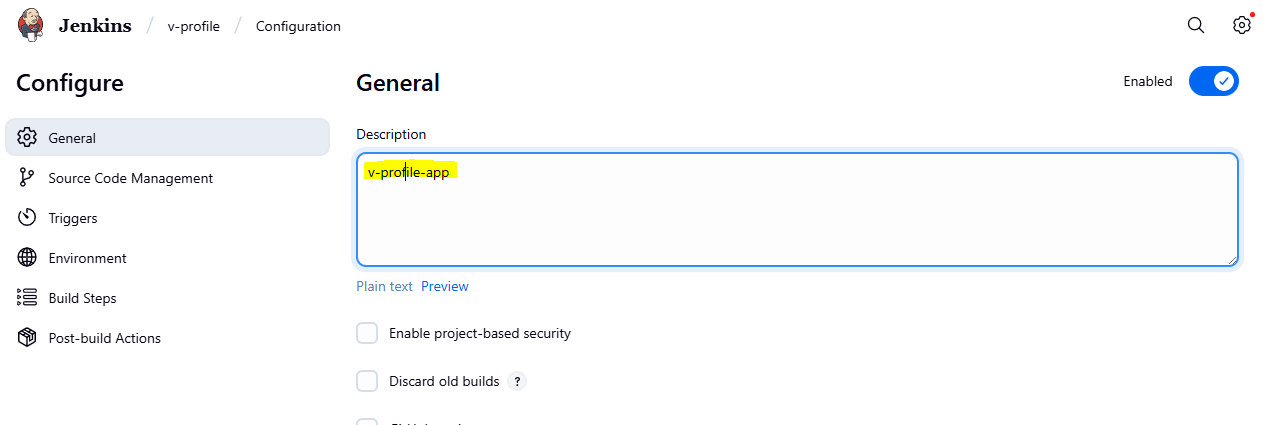
Give name

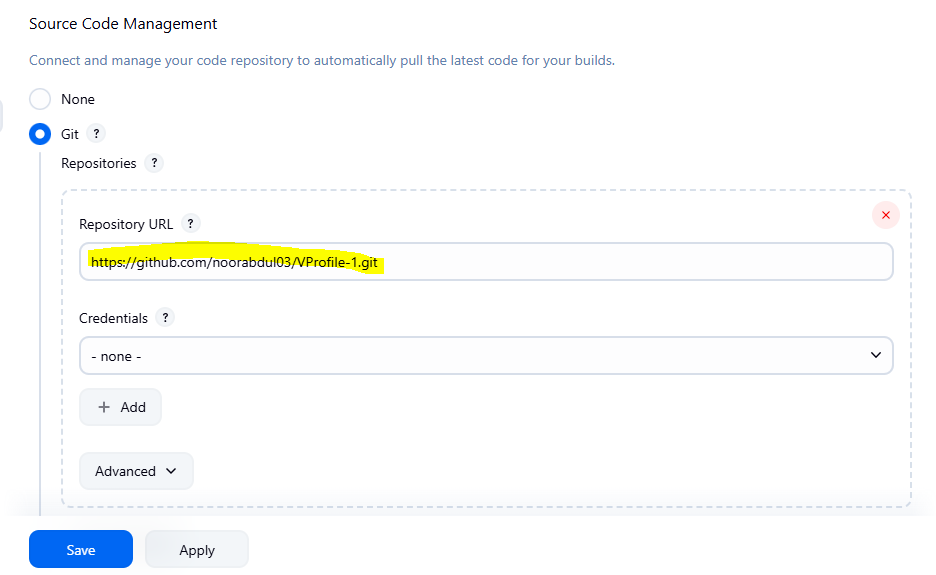
Click on Freestyle project



Click on ok

Note: Why we are using freestyle bcz we can customize as per requirement or we can use as per our requirement.



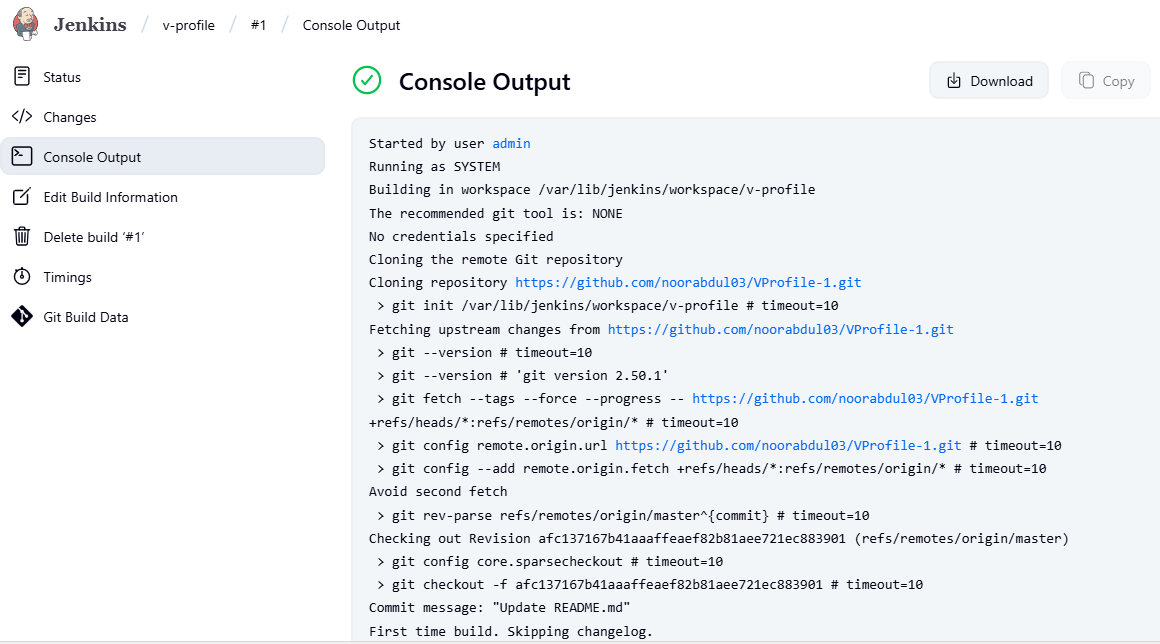


Click on Save

1. Click on Build Now :

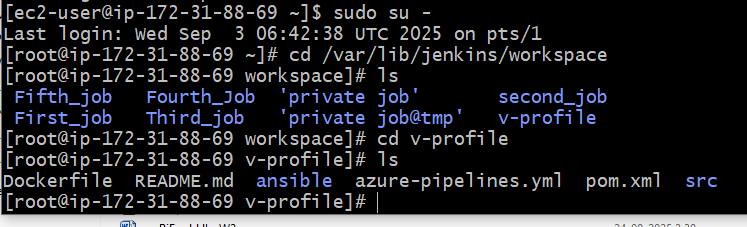
If I click on build now 🡪 It will clone this particular project – it will successfully done.

Go to v-profile job – click on build now



You can this job in your server

Cd/var/lib/Jenkins/workspace



The clone, first stage has successfully completed.

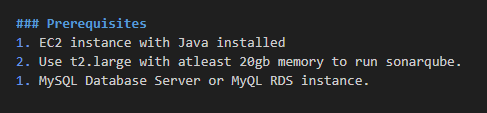
1. Now integrate Sonar – Second stage

It will check for the quality of code.

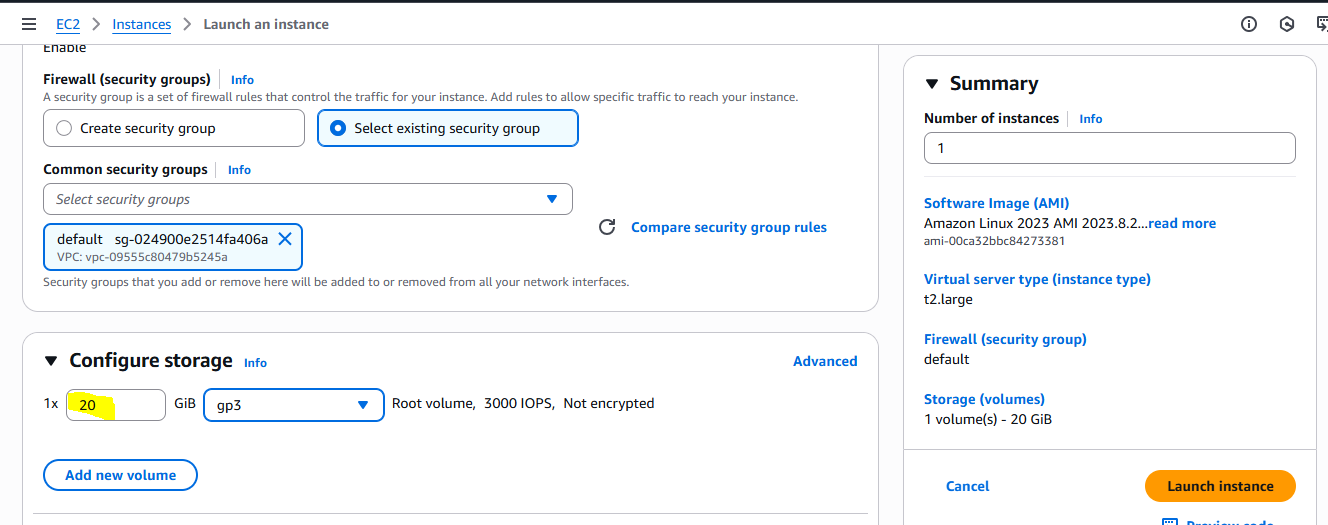
Note: If the quality of the code is not good then our code might leak the data or someone might easily hack our code.

At what base on metrics it will check the quality of code: Metrics duplicates in code, bugs, in the code, code smells, hot spots, vulnerabilities.

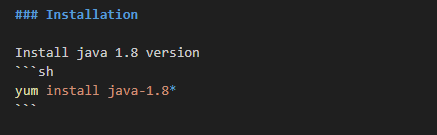
**Before creating check for Prerequisites**

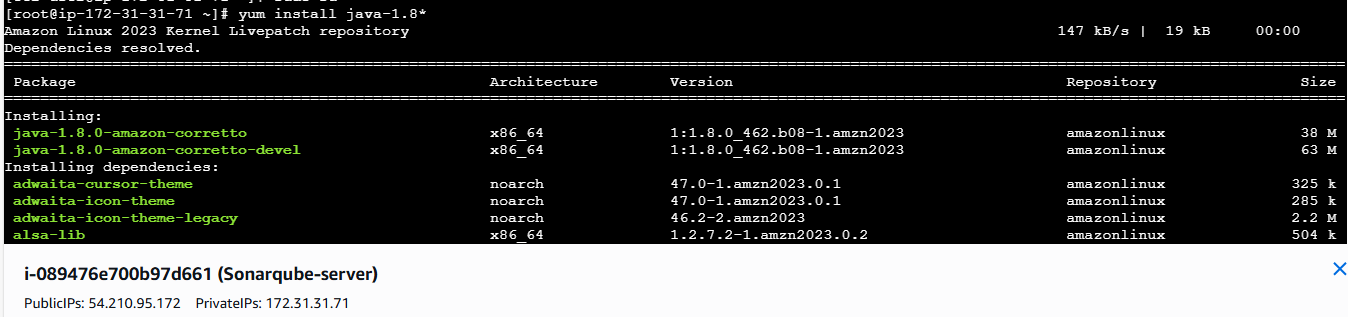


1. **Create EC2 Instance by keeping t2.large and 20gb memory.**



1. **Connect to Sonarqube server. And Install Java**

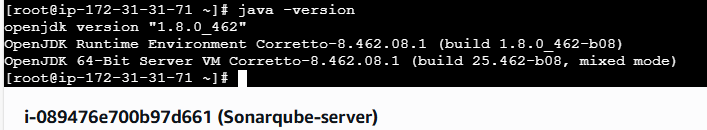




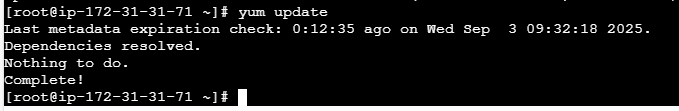
**Note: The reason why 1.8 version is using because the code in repository supports java version 1.8 and this code will not support java version 17 or other.**

**Sonarube also Java based web application. Sonarqube also developed in java language.**

**Java –version**



**Yum update**

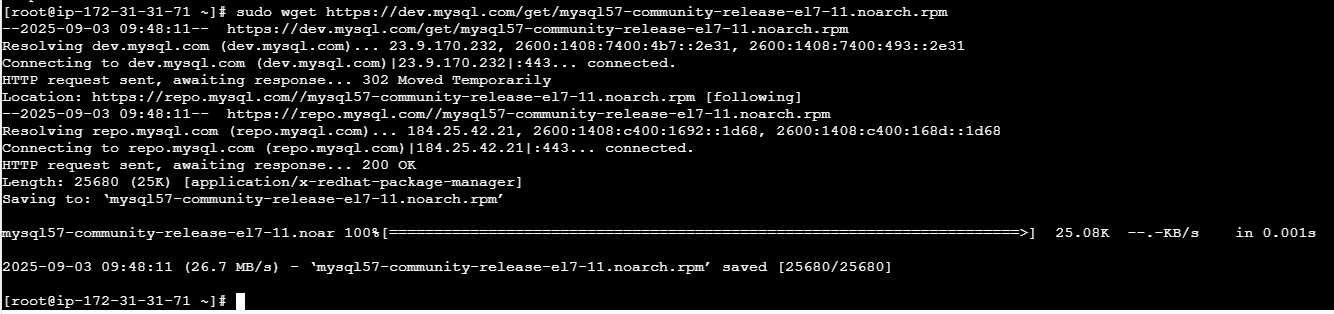


1. **Install Mysql**

**Download the rpm packages for mysql 5.7 version**

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

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



Then execute the second command

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

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



Complete!

In the above first we took the pacckage from the website and by using the second command we installed mysql. (Refers by Naveen P)

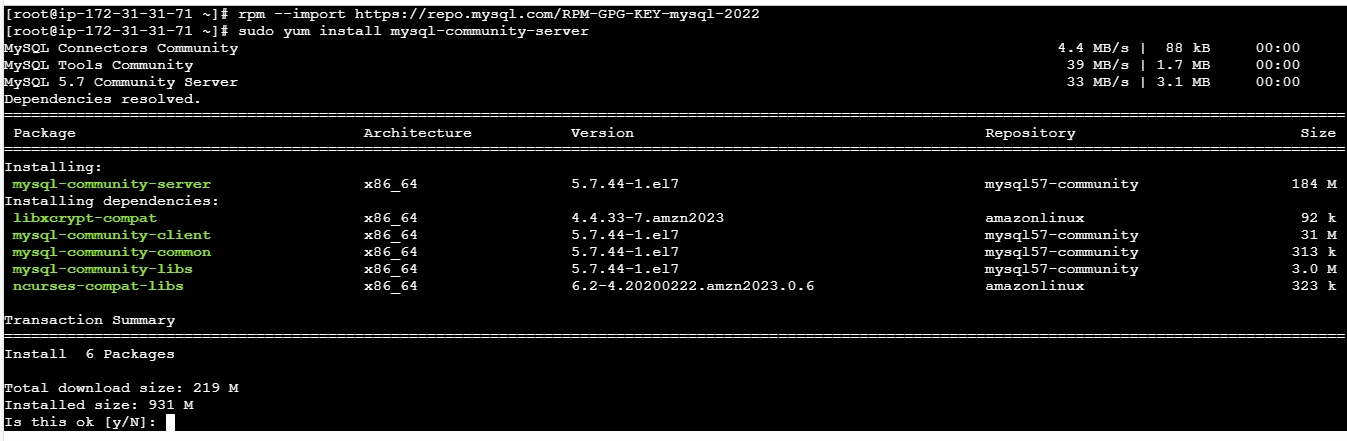
**Now let import the keygen**

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

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

sudo yum install mysql-community-server

sudo yum install mysql-community-server



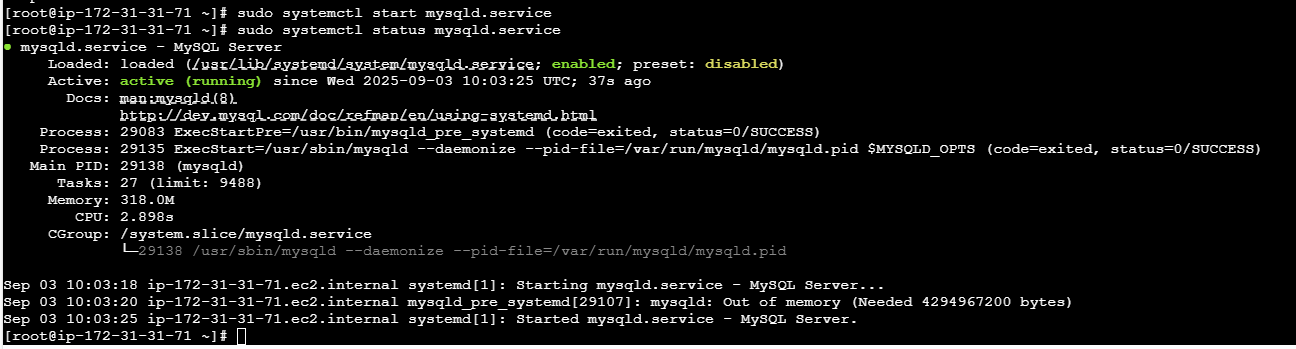
Complete!

**Now start the mysql service.**

sudo systemctl start mysqld.service

sudo systemctl start mysqld.service

sudo systemctl status mysqld.service



Note: Mysql runs on which port number

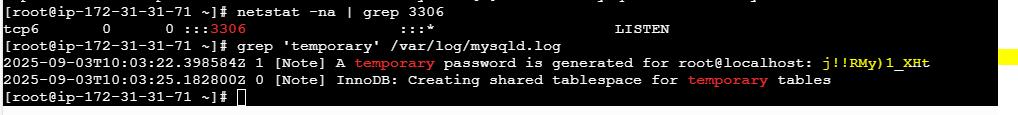
Command : netstat –na | grep 3306

Listen

**Now create a temporary root password to mysql db**

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

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



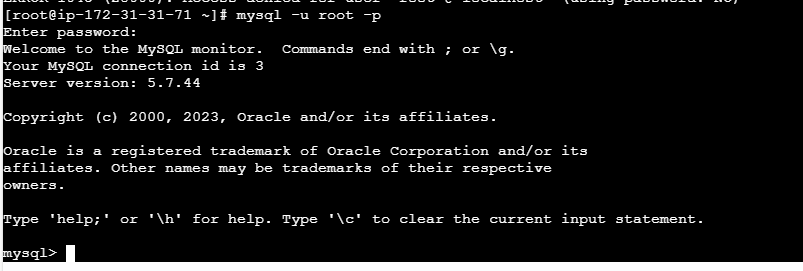
j!!RMy)1\_XHt

**Now Login to the server and change the password**

**Command to login into mysql server**

mysql -u root -p

**mysql -u root -p**



Now successully we were able login into mysql database.

**Now change the password as I got earlier the temporary password.**

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

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



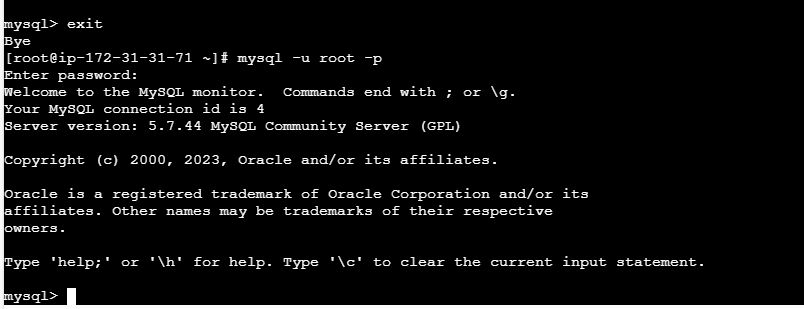
flush privileges;

flush privileges;

This will try to update the priviledges.

Then exit and let me check the new password working or not

New password is as above Admin@123



Once exit from the mysql database



Install Sonarqube 6.0 version

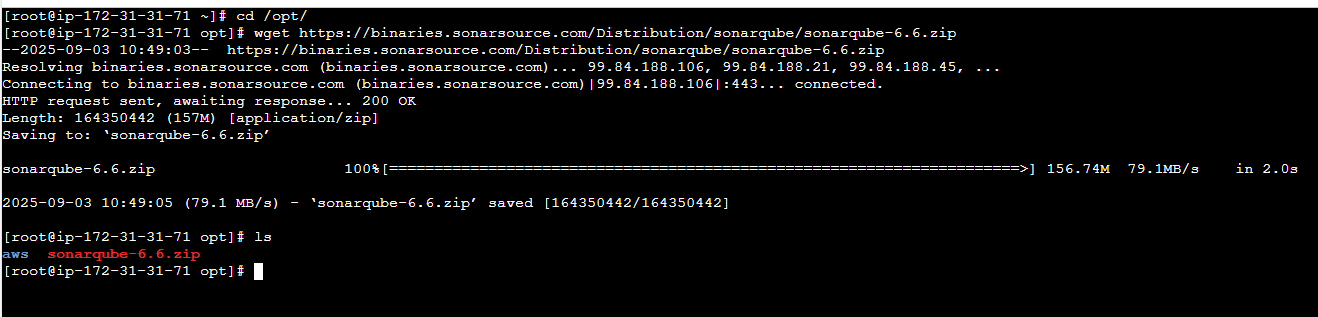
Note : Always use cd /opt/ directory to download anything.

And download package from this perticular location

Cd /opt/

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

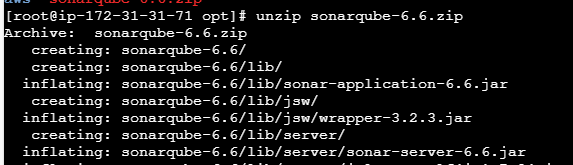
wget <https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.6.zip>

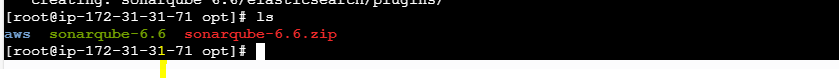


Then unzip this perticular directory

unzip sonarqube-6.6.zip

unzip sonarqube-6.6.zip





Successfully unzipped

**Now rename this unzip file to sonar**

**If you don’t want**

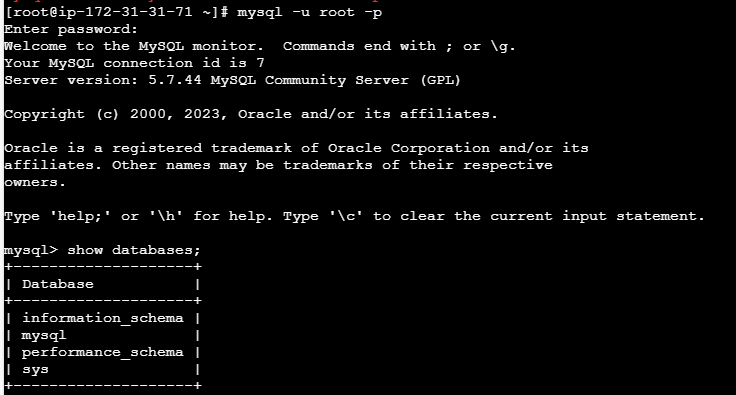
mv /opt/sonarqube-6.6 /opt/sonar

mv /opt/sonarqube-6.6 /opt/sonar



Beore creating a database check the existing database command is

show databases;

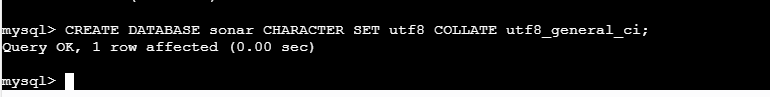


The above 4 are deault databases has been created

Now create a database with the name o sonar or any

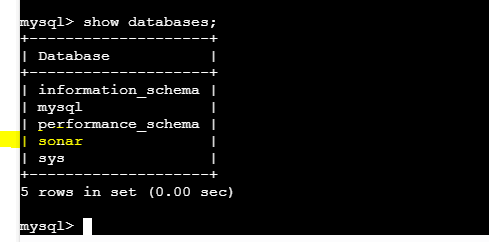
CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8\_general\_ci;

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



I you want to check the new databases has created or not, you can check

Show databases;



**Now create 2 different users**

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

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

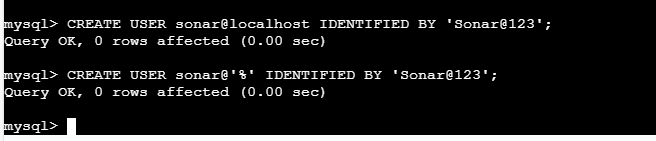
One is local user

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

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

One is root user

The reason o creating users is to connect with database.



Now provide permission to the database sonar by this user

Or else this user will not be able to store anything in this database

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

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

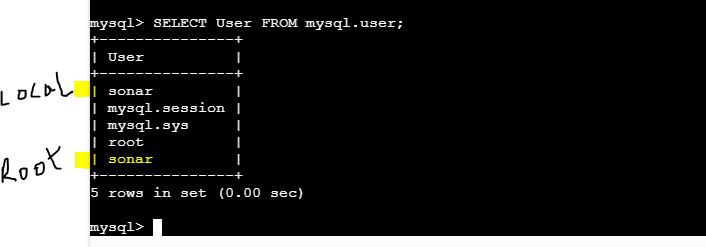
GRANT ALL ON sonar.\* TO sonar@'%';

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

If you want check your user are create or not

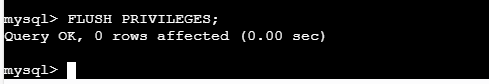
SELECT User FROM mysql.user;

SELECT User FROM mysql.user;



FLUSH PRIVILEGES;

FLUSH PRIVILEGES;



Exit

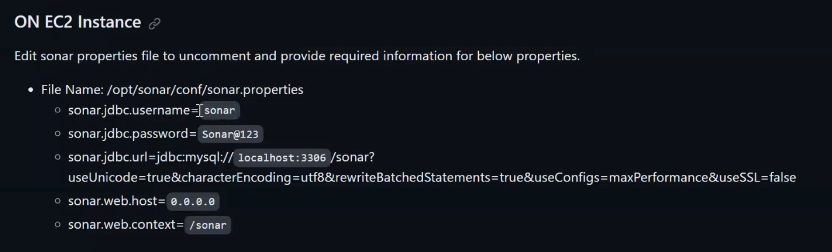


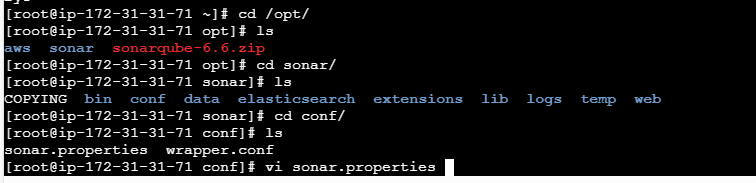
Now our database tasks are completed

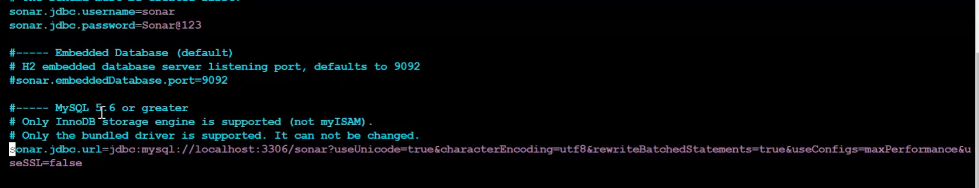
……………………………………………………………………………………………………………………

In the downloaed sonarqube files there is a file call sonar.properties

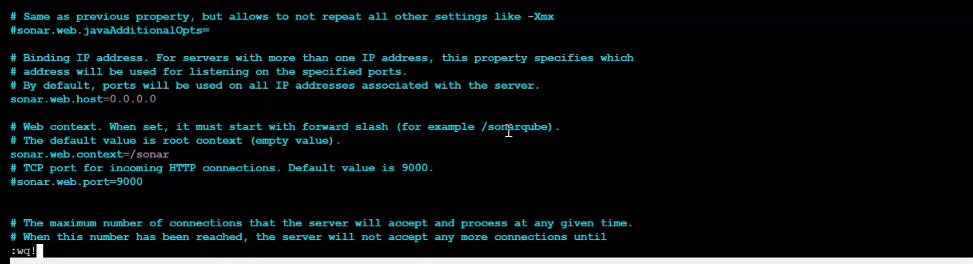
There we need to add the jdbc user and password. Means whatever user created in database.







Change as it is the grey line and beside.



Change as it is the grey line and beside.

And save this file.

And create user but I am not creating because we have alreay ec2-user

The reason it is asking to create user is we can’t run sonarqube as a root user. It will get fail.

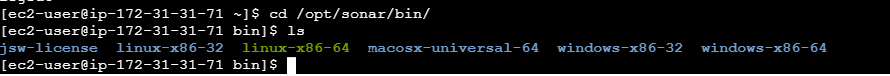
So we need to execute sonarube from non root user only.

In my case I have user ec2-user

**Start SonarQube service**

Be in the ec2-user

cd /opt/sonar/bin/

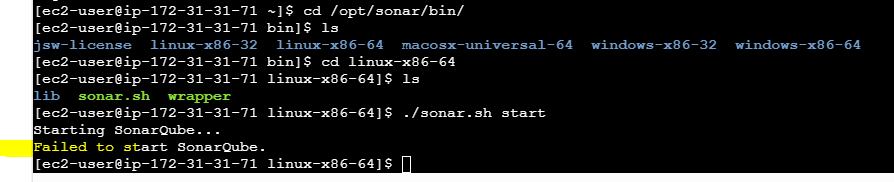


Out of these files linux-x86-64 basically it is the hardware reuirement, means the image what we use while creating ec2.

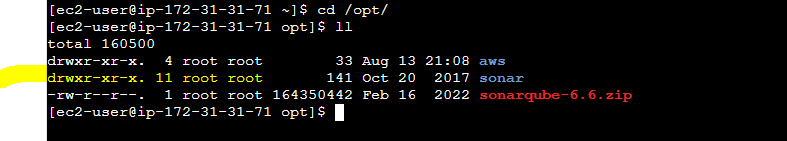
Inside this file we have one file called sonar.sh

And I need to execute this perticular file the command is

./sonar.sh start



To resolve the error

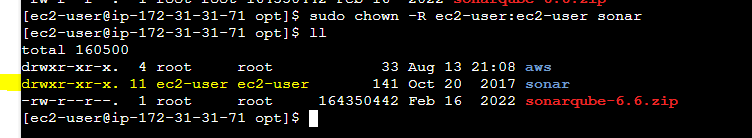


The sonar directory is having ownership is root and we are trying start sonar from being in ec-user

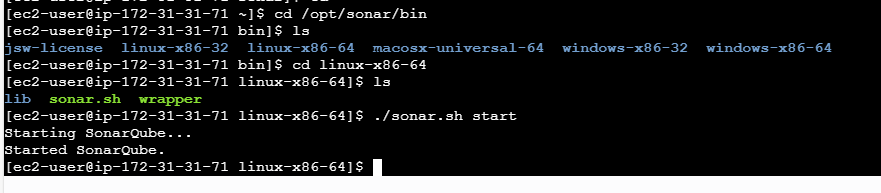
So first change the ownership

sudo chown -R ec2-user:ec2-user sonar

now it has changed

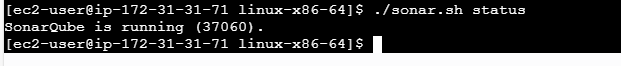


Now try to start sonar

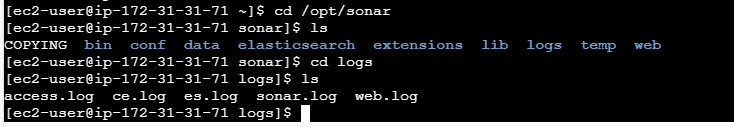


Now check the status

./sonar.sh status

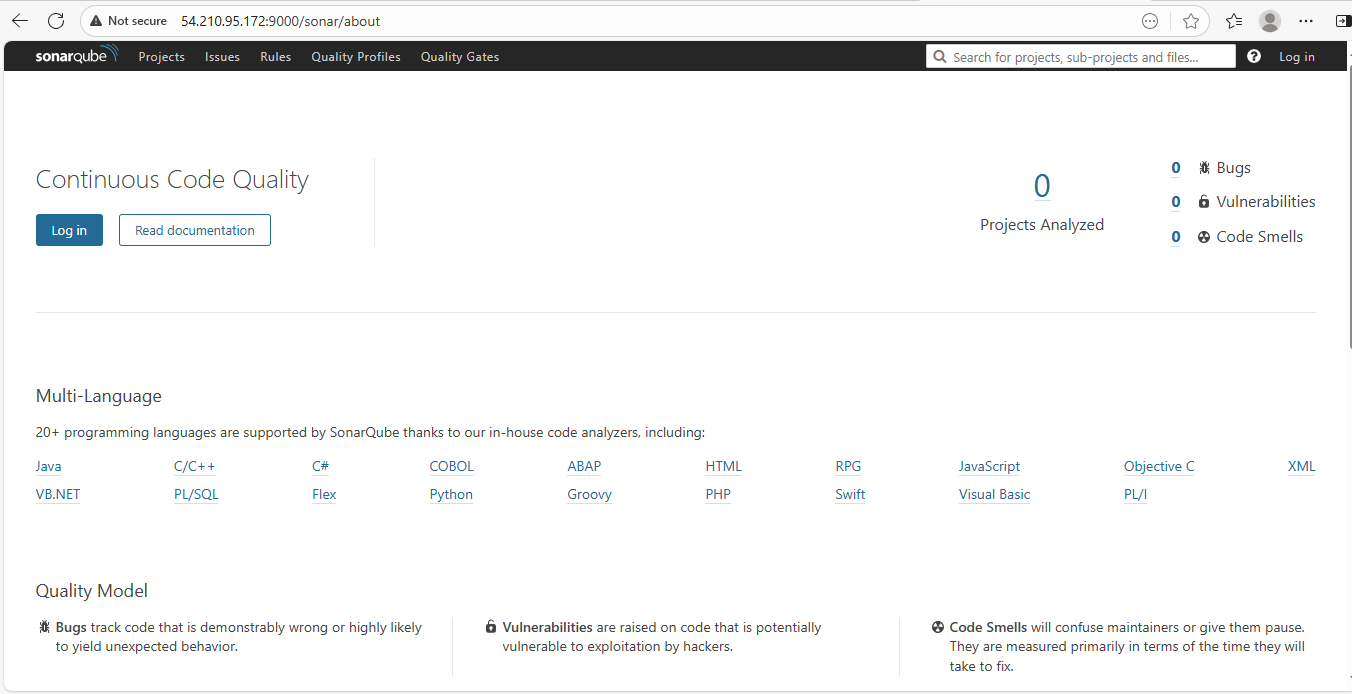


To check the logs



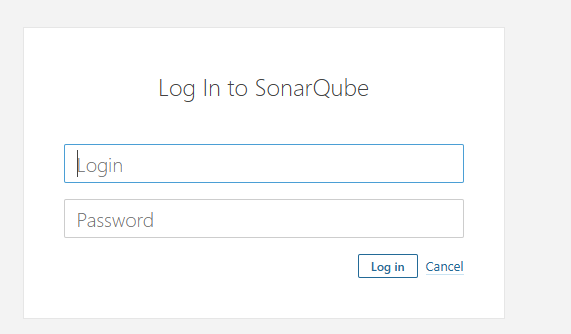
**Now Access Sonarqube**

**Copy the piblic IP of sonar ec2 server with 9000 port number with path prefix (/sonar)bcz we gave in**

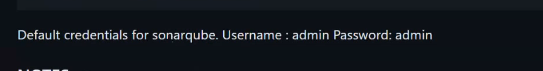


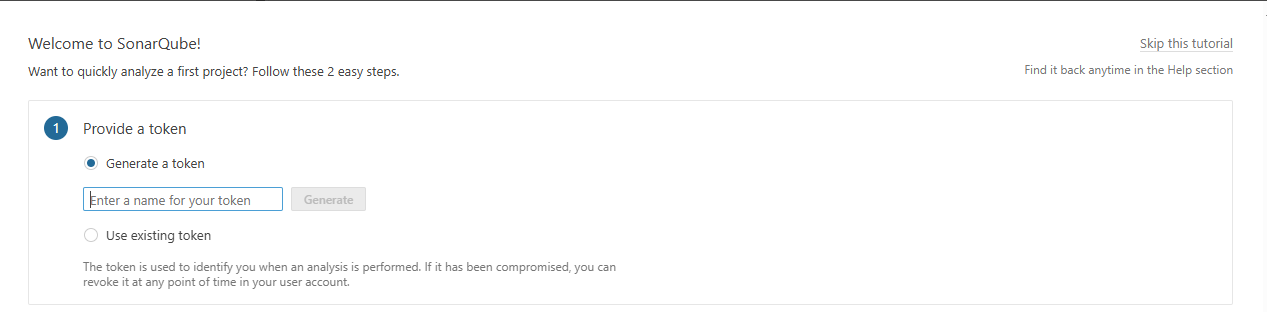
**Now we are able to see our sonar dashboard successfully**

**Now LOGIN**

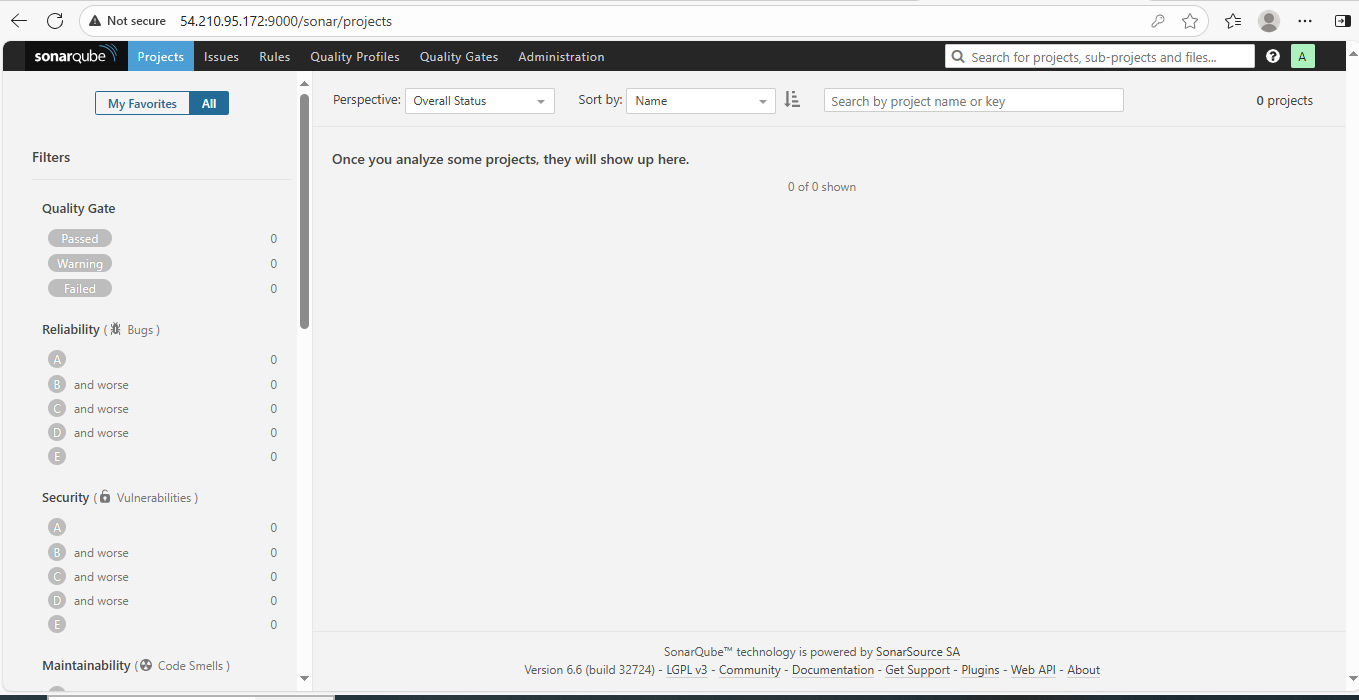


**Default username and password**





**It is asking genearte token, but now clcik on skip this tutorial**

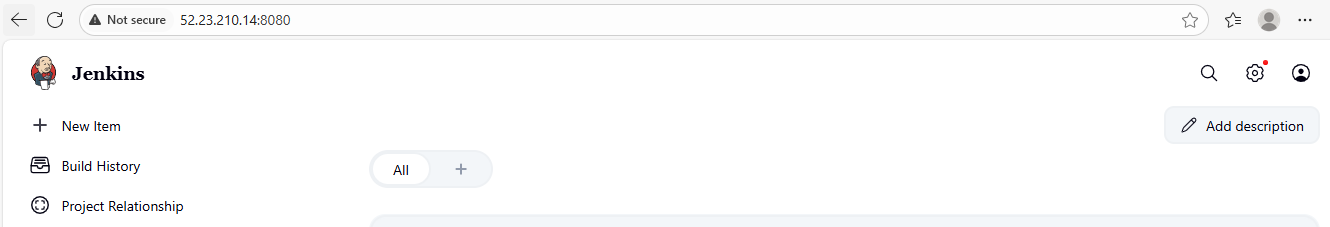


**So we have successully created sonarube server and we have successull see the dashboard.**

**Now I need to integrate soanrqube means I should tell to jenkins that this is my sonarqube server. This is where you need to push the reports .**

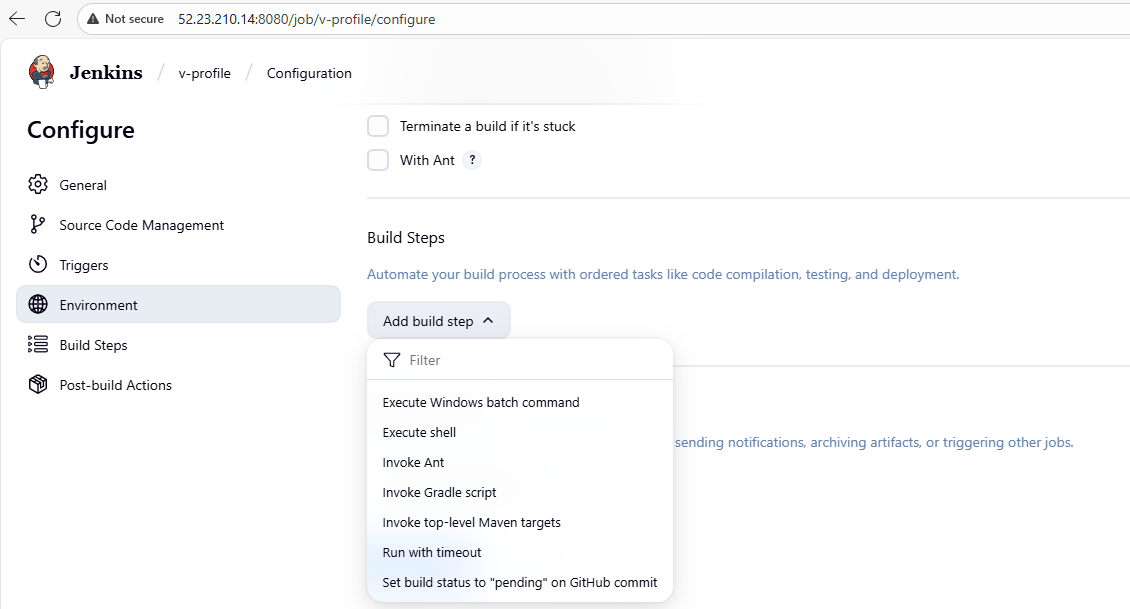
1. **For any integration first we need to install the plugins.**

**Go to Jenkins server GUI**



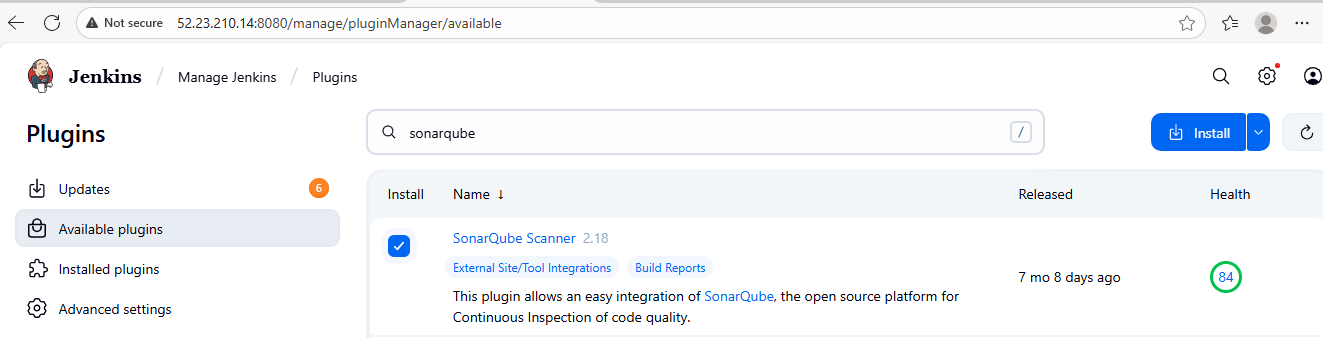
**Got to mnage jenkins**

**Before this just duplicate this page and see the effects.**

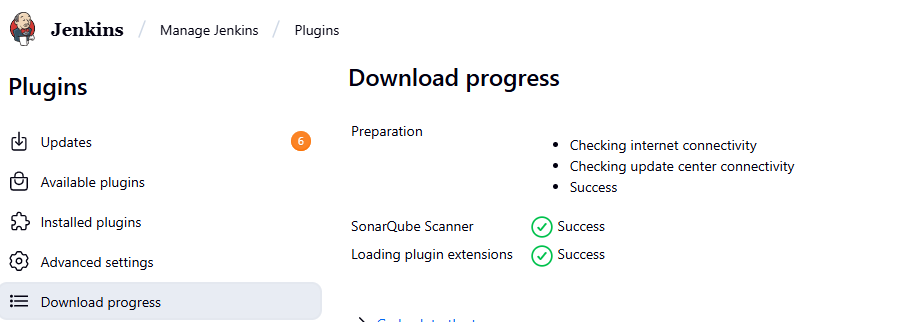


**Here it is not showing plugins as sonar scanner. But once we install plugins it will show here.**

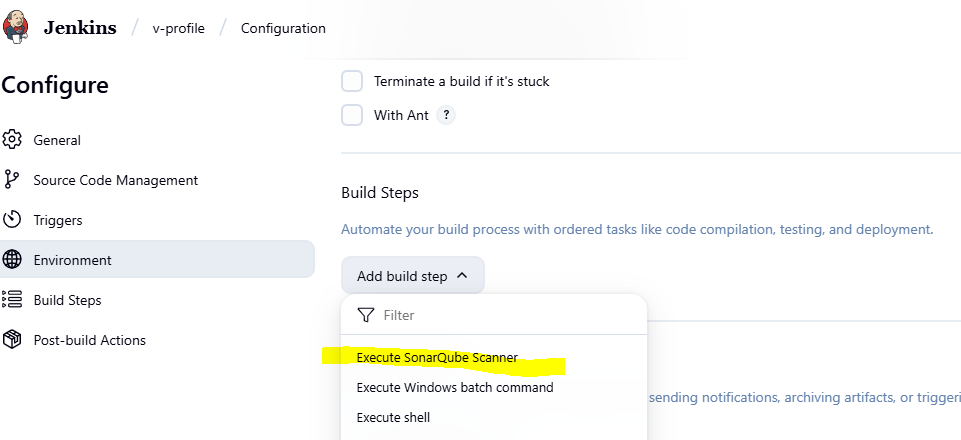
**So go to jenkins – manage jenkins – plugins**



**Clcik on install**



**If you see in the job coniguration the plugin has added**



**Apart from this If you go to**

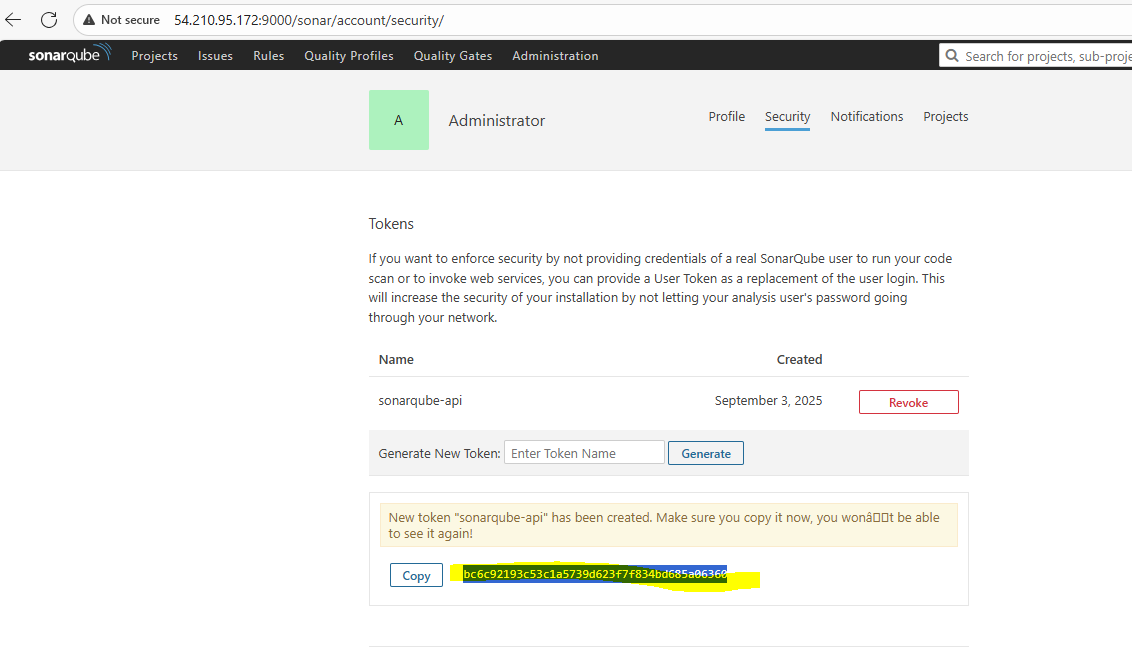
**Here you can see to which server you want to connect. The server url you can map**

**And it is asking sonarqube authentication token**

**Means, how do you want to connect to your server what should be the authentication. So we should configure credential. But sonarqube will not help us create any credentials.**

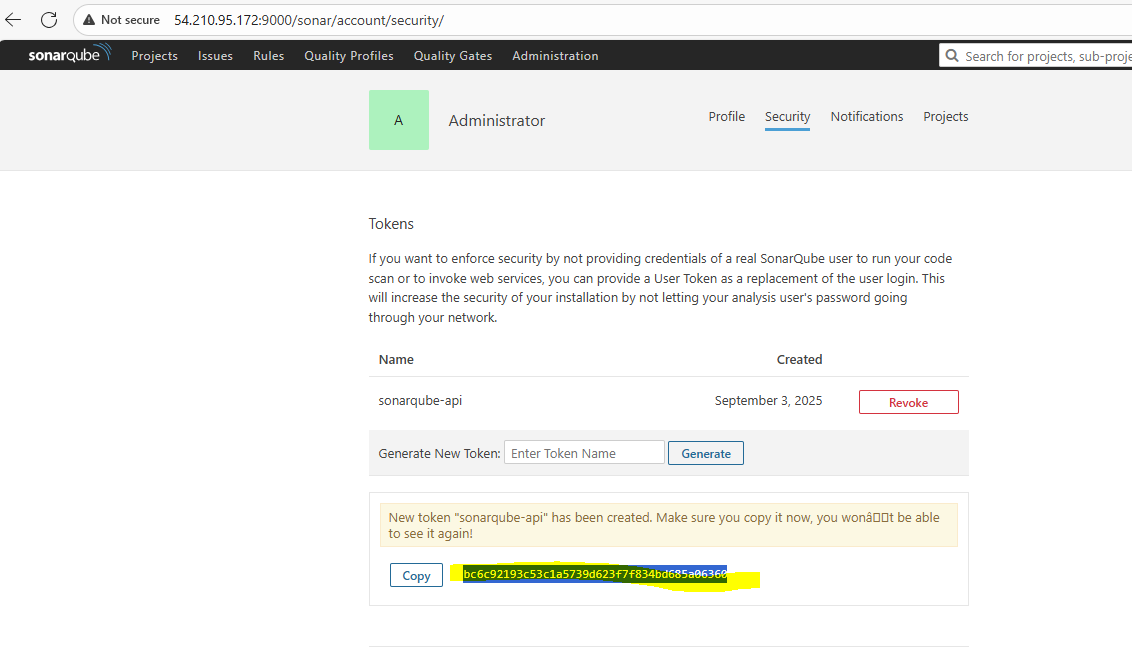
**So we need to create one authentication token.**

**GOTO sonarqube server gui – click on admin- clcik my account – Goto security**

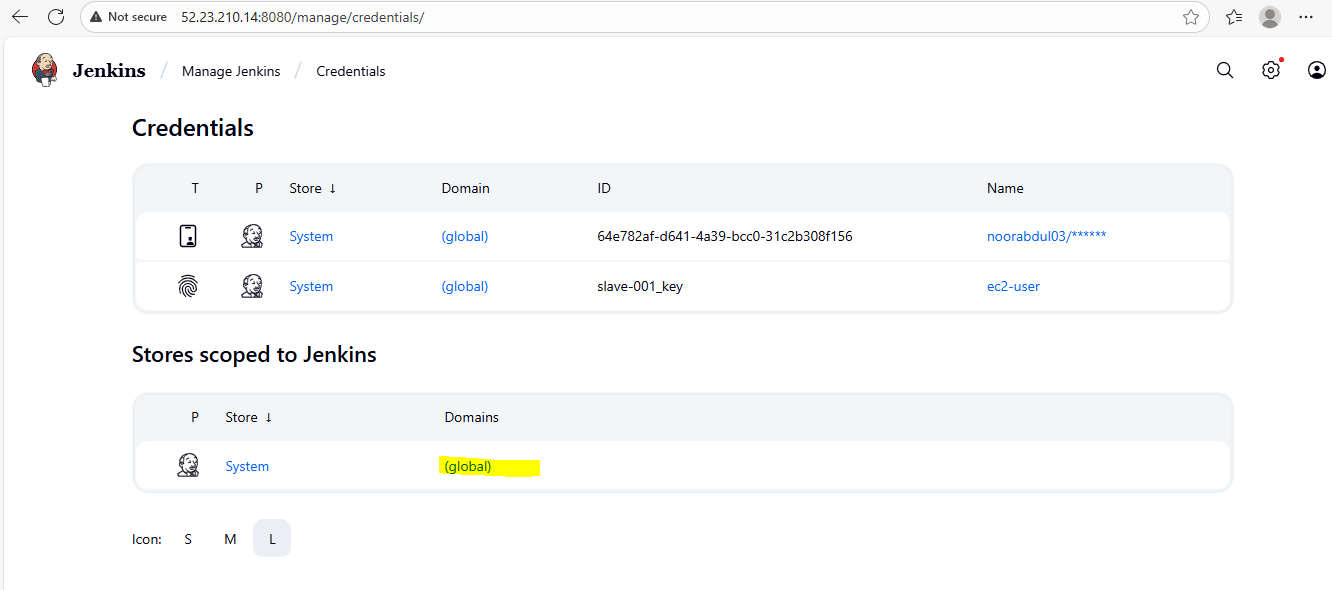


**You can give any name**

**This token will help me to connect to my sonarqube server from jenkins**

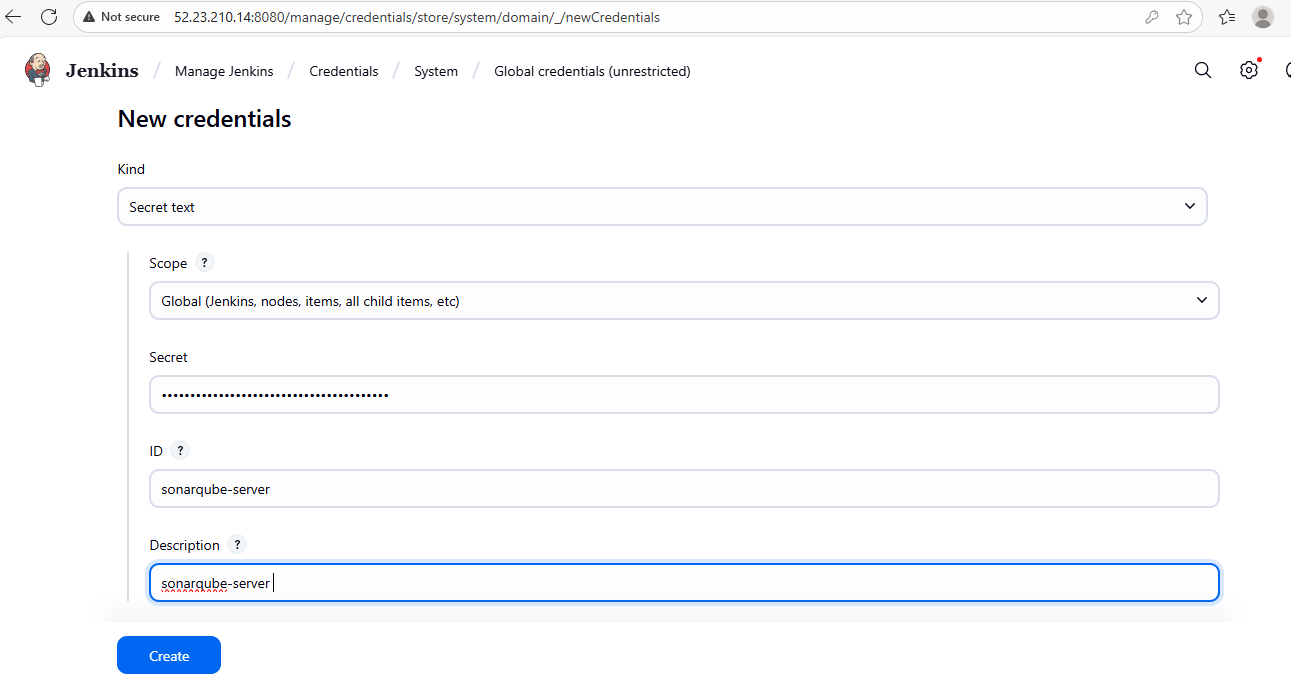


**Create one credentials first in Jenkins serverGUI**



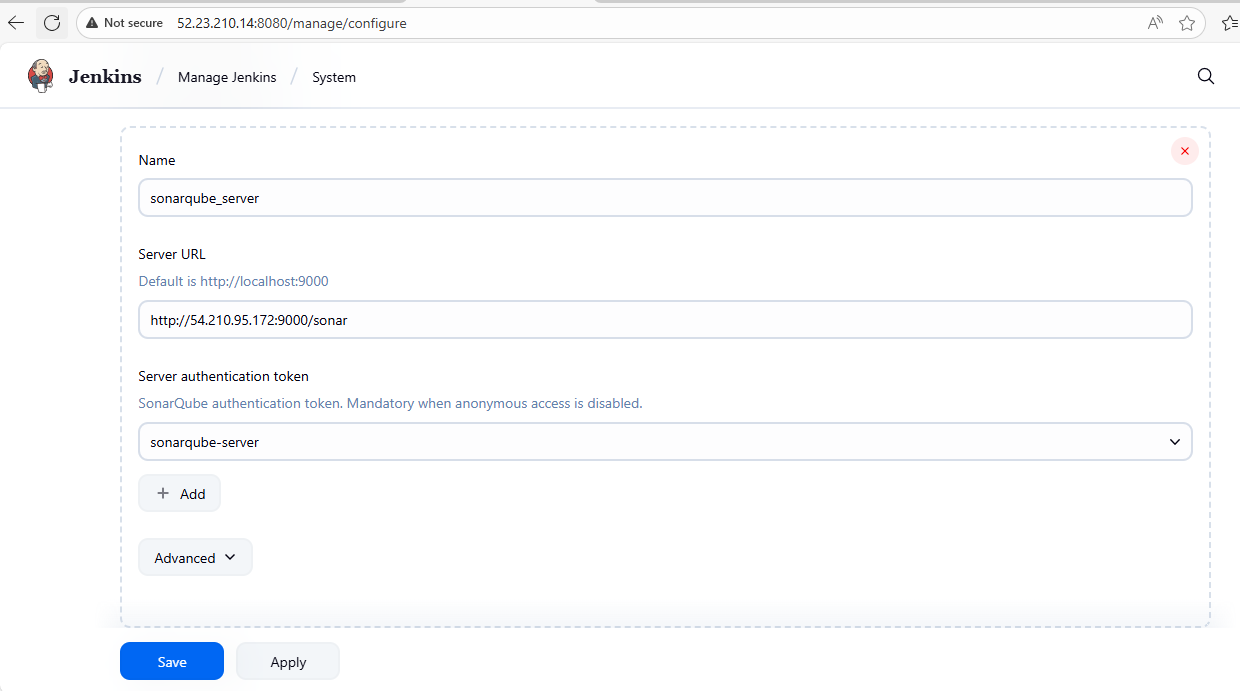
**Clcik on global**

**Clcik on add credentials**



**ID is for understanding**

**Click on create**



**Click on save**

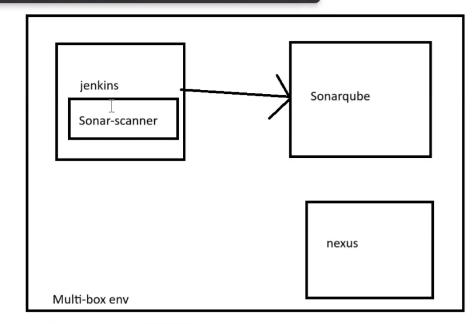
**Now our jenkins is successfully with sonarqube server.**

**………………….**

**Once the above done we need to have some mechanism in our jenkins server which will help us to push the reports. Menas whatever generated on this jenkins should be sent sonarqube.**

**So for that purpose we have some AGENT call SONAR SCANNER**

**Sonar scanner is responsibe to push all the reports to the sonarqube server.**



**Now install SONAR SCANNER in JENKINS SERVER**

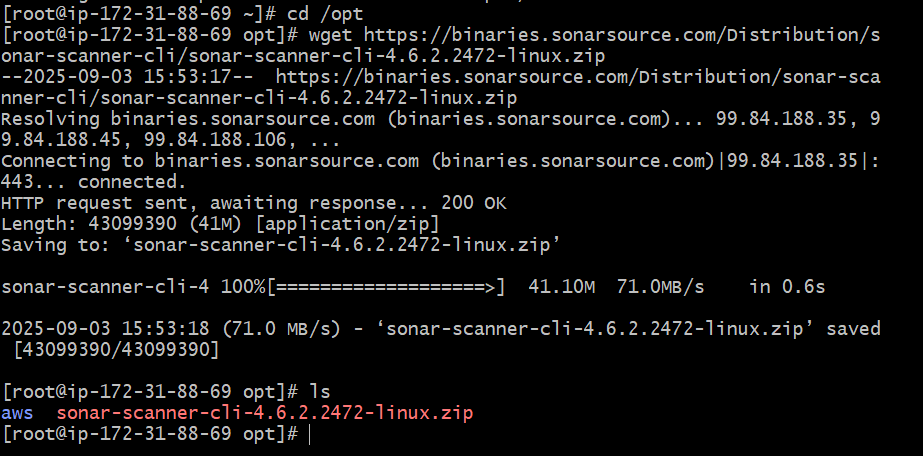
**(open the document SonarQube integration with Jenkins)**

**Goto cd /opt/**

**Always do any downloads in opt**

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

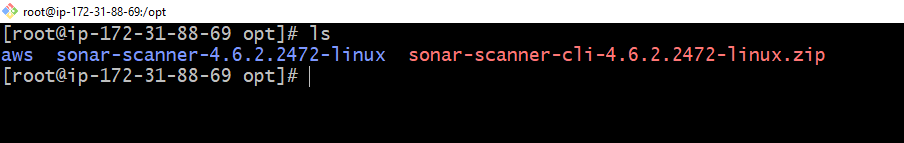
**wget** [**https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.6.2.2472-linux.zip**](https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.6.2.2472-linux.zip)



**Now unzip the sonar scanner file**

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



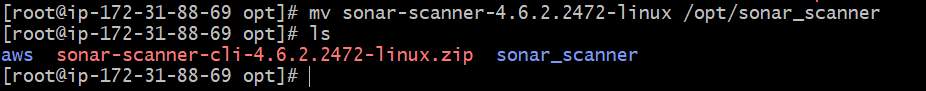


**See the above is unzipped file and the version is 4.6.2**

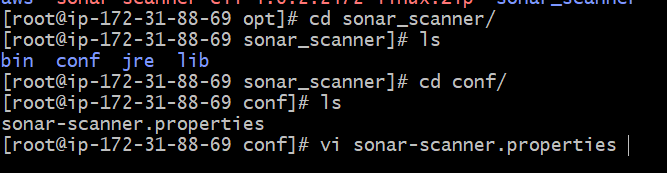
**Now rename the unzipped file**

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

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

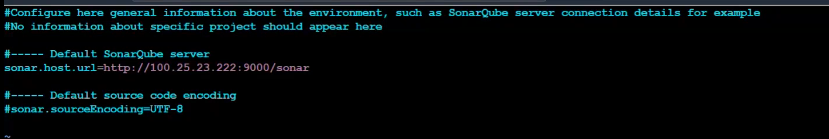


**Now go to the prpoerties file**



**So what we need to specify is**

**Means,inside this file pass the details of sonarqube server. Means this is my sonarqube server go and push to this perticular sonarqube**

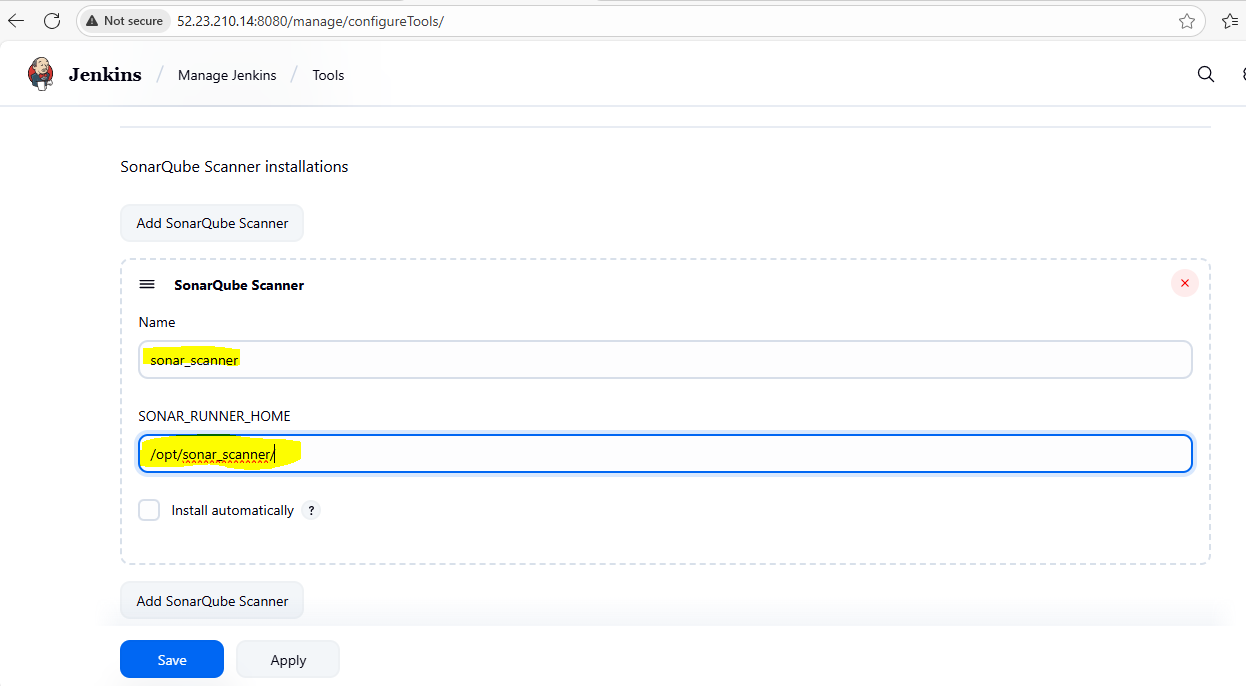


**Save this.**

**What we have did we were able to steup sonarqube scanner as well as integrate this sonar scanner to sonarqube**

**Now**

**Go to Jenkins – manage jenkins – Tools**

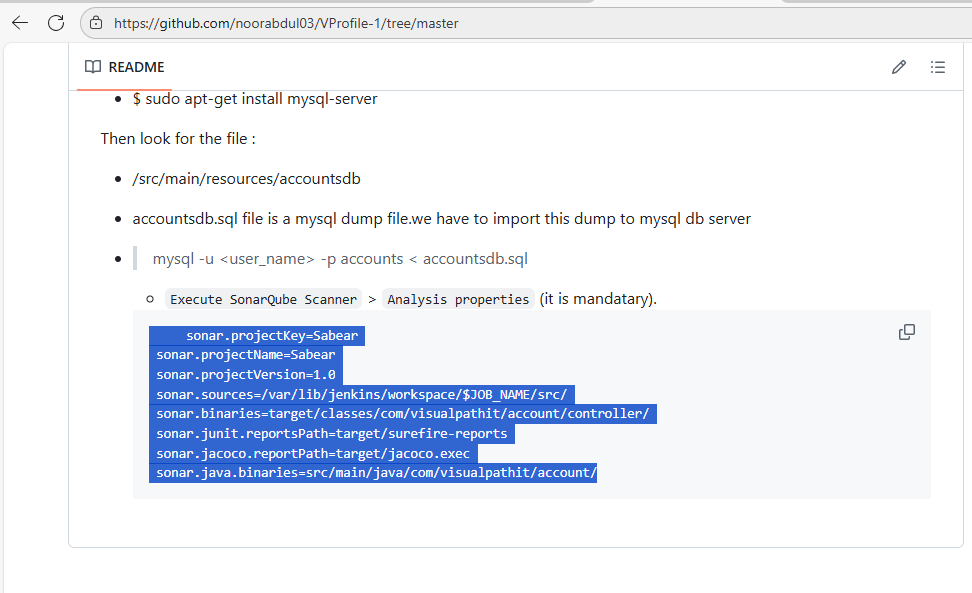


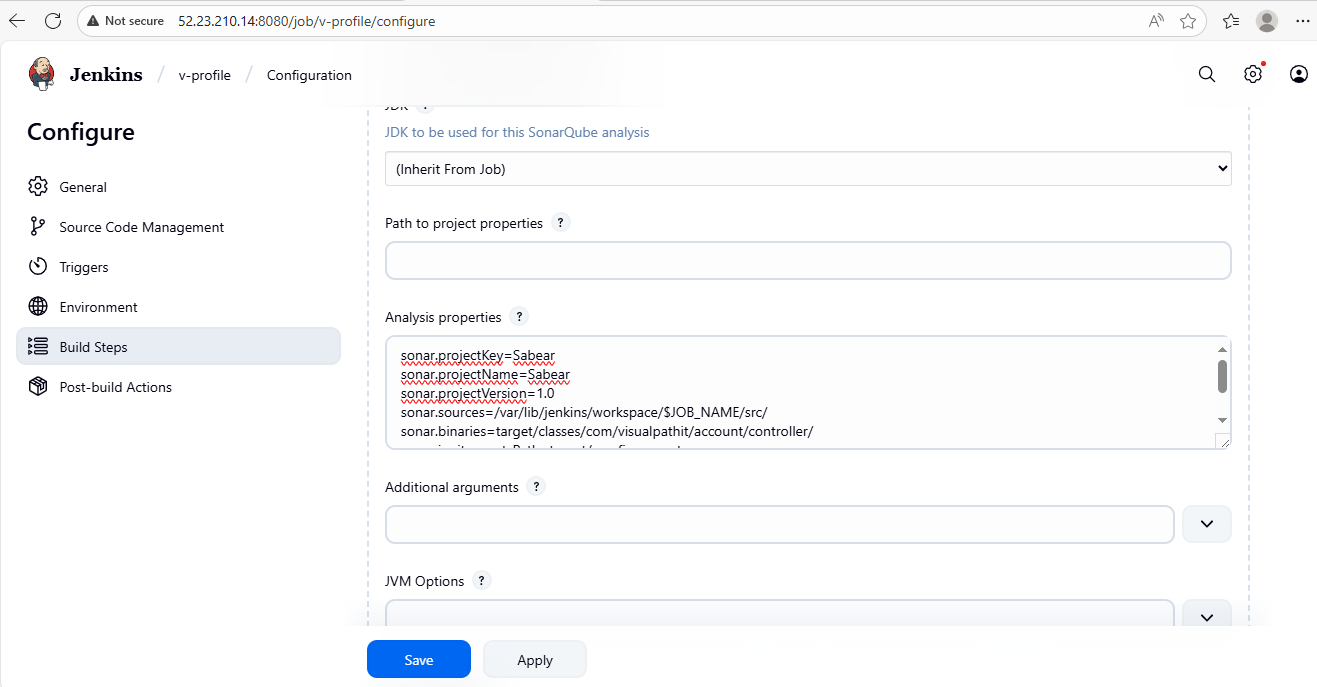
**Click on save**

**Now we need to add the details means, we need to tell to our jenkins (v-proile) job that go to this perticular scanner and check for the report**

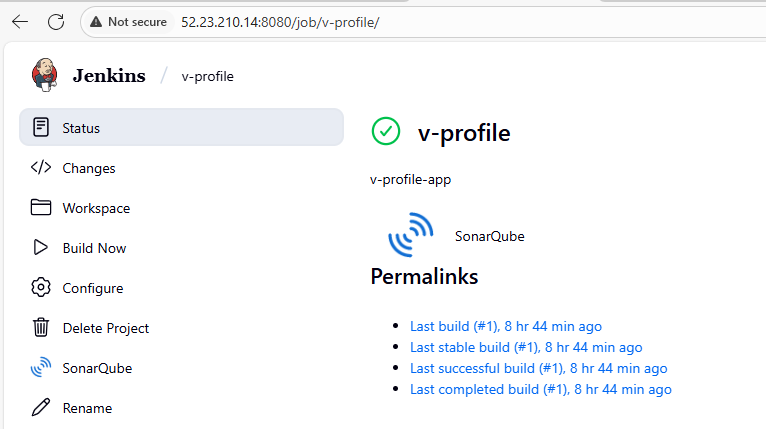
**Analysis properties : this details generally we will get from the developers only because we are not aware path and other details. So they will share**

**So whenever we get any code always check for README.md**



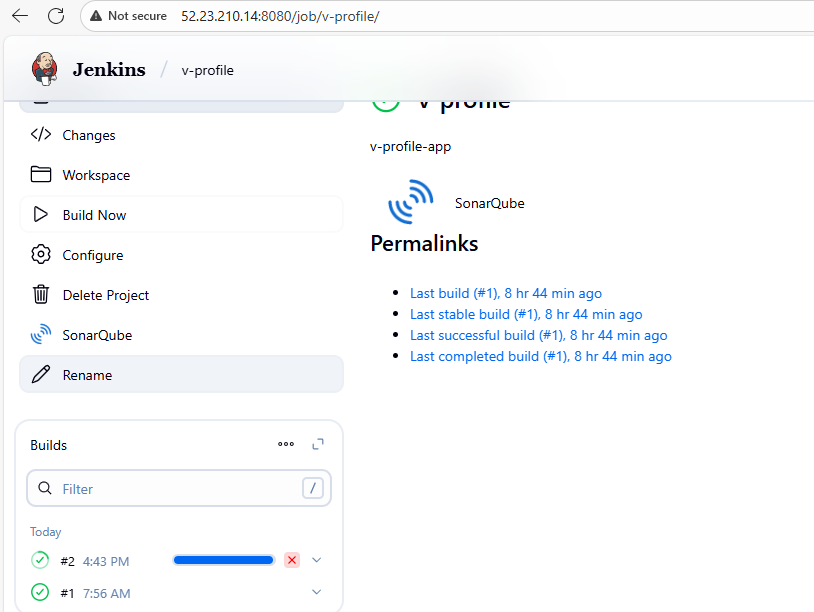


**Click on save**

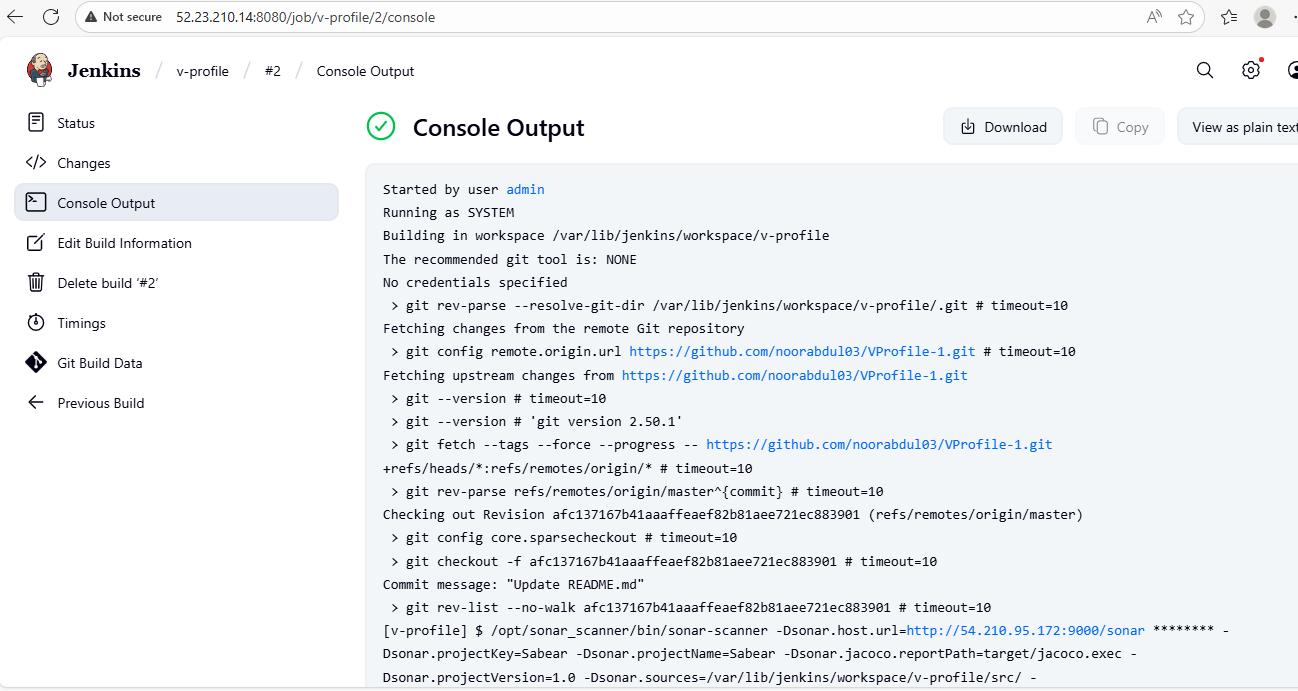


**If you see above, our job has changed to sonarqube**

**Now click on build now**



**Click on console output**

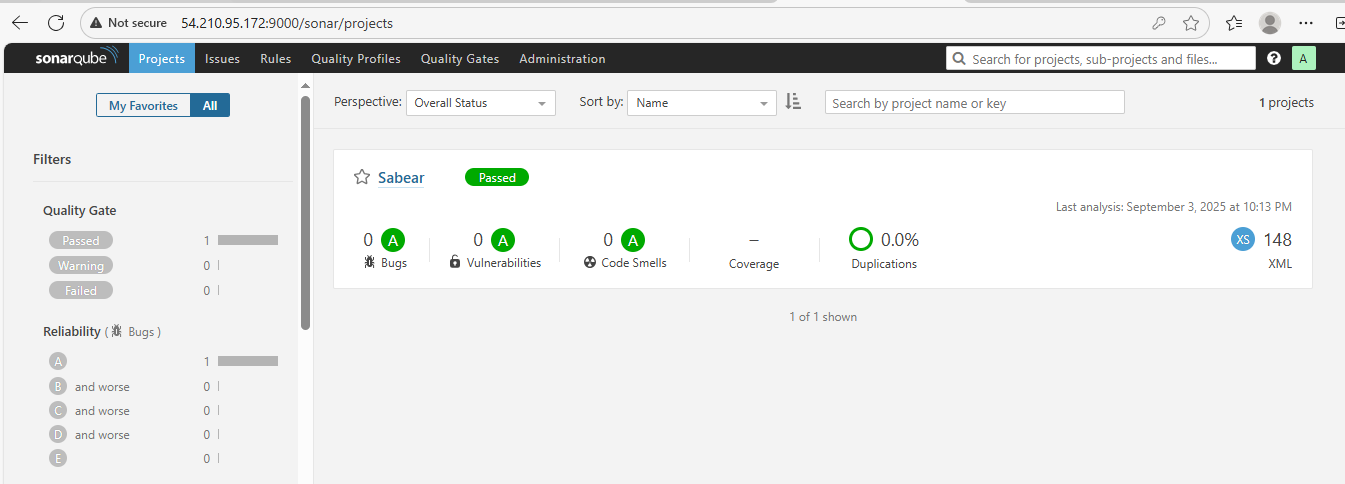


**Now If I go to sonarube server**

**Click on home button SonarQube at top left**

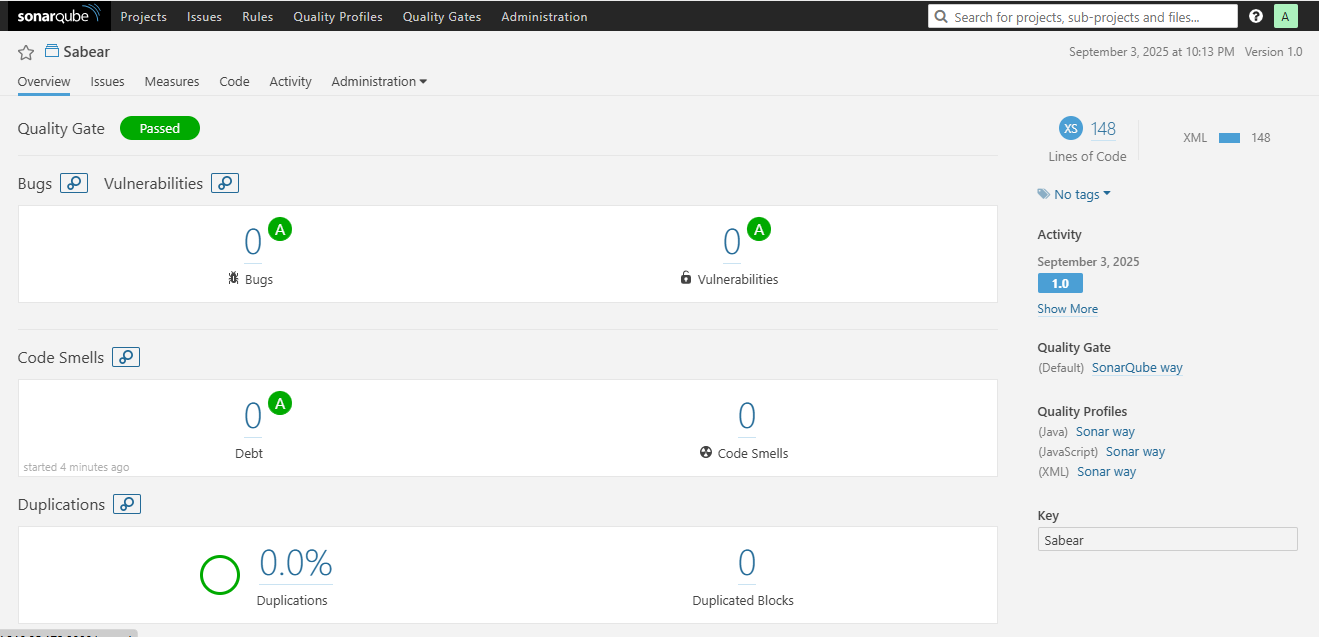
**Click on skip**

**Now see here our project sabear successfully created**



**Click on sabear**

**Here you can see quality gate has been passed**



**Here we can see everything is zero 0 , we are not able to see any reports as of now . why ? don’t we have any bugs etc., ?**

**We do have some bugs in this code**