Sonar Jenkins Integration

Open WSL
Requirements
Step 1 :: Machine must have jdk-17 in wsl

```
sudo apt update
sudo apt install openjdk-17-jdk -y
java -version
```

Step 2:: install PostgreSQL

```
sudo apt install postgresql postgresql-contrib -y
#Connect with Posgres for creating user
sudo -u postgres psql
#Let's write a query for creating user
CREATE USER sonar with PASSWORD 'sonar';
CREATE DATABASE sonarqube OWNER sonar;
\q
#exit from postgres
```

Step 3 :: Install Required Dependencies
sudo apt install unzip wget gnupg2 -y
https://github.com/SonarSource/sonarqube/releases (from here you can choose the release version for sonarqube)
#move to the root folder

```
#Create one directory

sudo mkdir opt1

cd opt1

sudo wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-
10.5.1.90531.zip
# (link has been taken from the sonar official website only)
sudo unzip sonarqube-10.5.1.90531.zip
# move to the folder
sudo mv sonarqube-10.5.1.90531 sonarqube
```

```
sudo chown -R $USER:$USER sonarqube
#Edit Configuration file
nano /opt1/sonarqube/conf/sonar.properties
```

edit configuration file for adding credentials and DB URL

```
# User credentials.
# Permissions to create tables, ind
# The schema must be created first.
sonar.jdbc.username=sonar
sonar.jdbc.password=sonar
```

```
#----- PostgreSQL 13 or greater
# By default the schema named "public" is used. It can be ov
sonar.jdbc.url=jdbc:postgresql://localhost/sonarqube
```

```
sonar.projectKey=Sonar_Python
sonar.sources=.
sonar.python.version=3.10
#----- Embedded Database (default)
# H2 embedded database server listening
#sonar.embeddedDatabase.port=9092
```

set Above mentioned Project properies.

ctrl + O , enter , ctrl + X

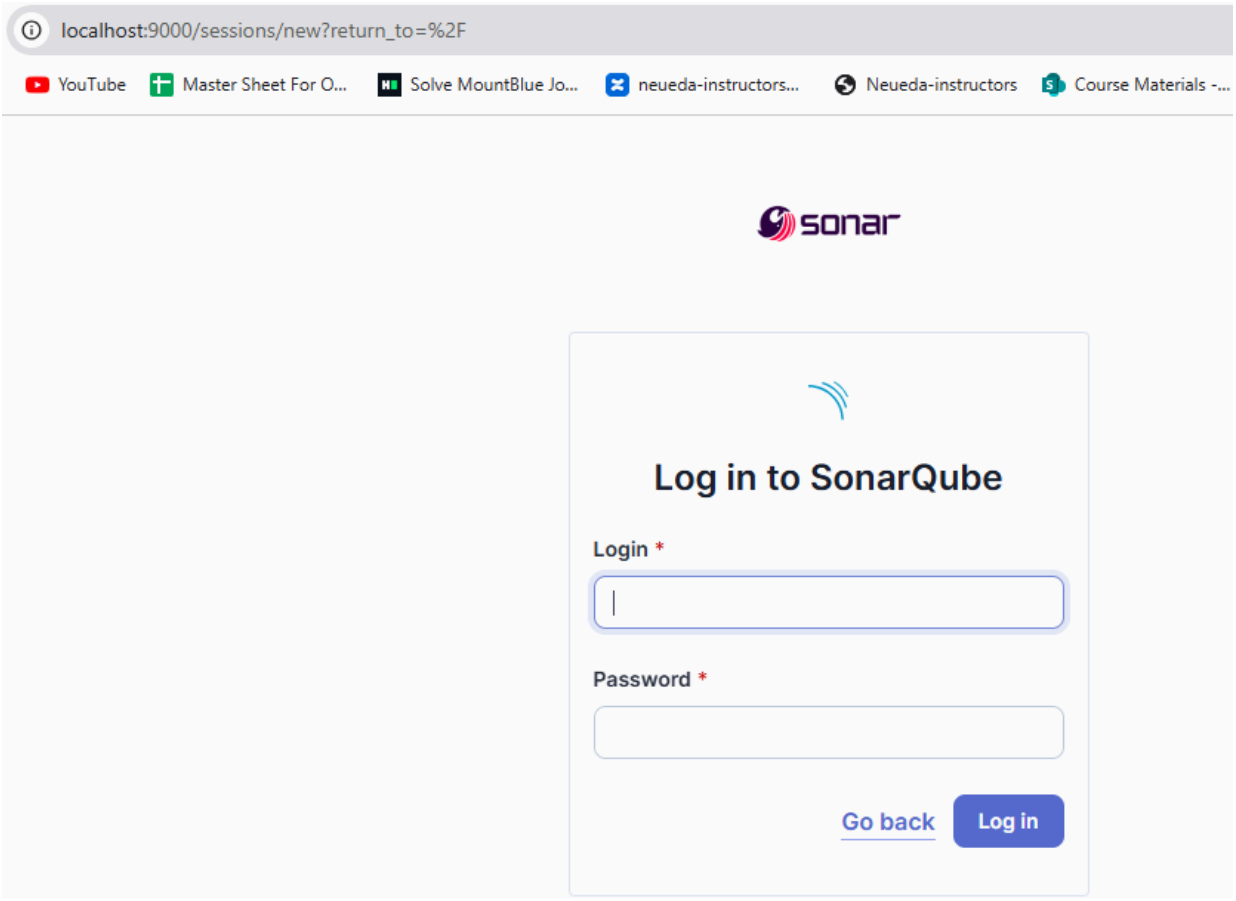
Let's Start Sonar Qube

```
sonam@DESKTOP-4F8ELLU:/opt1$ cd sonarqube/bin/linux-x86-64/
sonam@DESKTOP-4F8ELLU:/opt1/sonarqube/bin/linux-x86-64$ ./sonar.sh start
/usr/bin/java
Starting SonarQube...
Started SonarQube.
sonam@DESKTOP-4F8ELLU:/opt1/sonarqube/bin/linux-x86-64$ ./sonar.sh status
/usr/bin/java
SonarQube is running (1942).
```

to see the Logs go back to the logs folder.

```
sonam@DESKTOP-4F8ELLU:/opt1/sonarqube/bin/linux-x86-64$ cd ../../
sonam@DESKTOP-4F8ELLU:/opt1/sonarqube$ cd logs/
sonam@DESKTOP-4F8ELLU:/opt1/sonarqube/logs$ cat sonar.log
2025.06.14 04:15:41 INFO  app[][o.s.a.AppFileSystem] Cleaning or creating t
be/temp
2025.06.14 04:15:42 INFO  app[][o.s.a.es.EsSettings] Elasticsearch listenin
 TCP: 127.0.0.1:{}]
2025.06.14 04:15:42 INFO  app[][o.s.a.ProcessLauncherImpl] Launch process[E
onarqube/elasticsearch]: /usr/lib/jvm/java-17-openjdk-amd64/bin/java -Xms4m
Dcli.name=server -Dcli.script=./bin/elasticsearch -Dcli.libs=lib/tools/serv
```

try to access http://localhost:9000 you will get below login screen



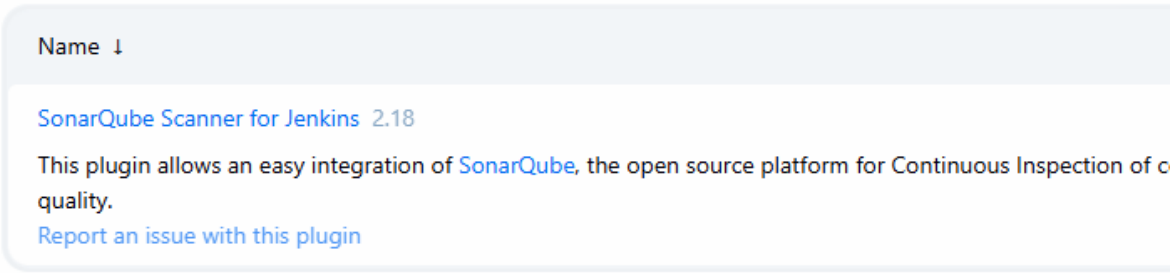
Enter admin admin for both username and password.
Change the Password and then you will be redirected to your Dashboard Screen.

Start Jenkins and Integrate Jenkins With Sonar.

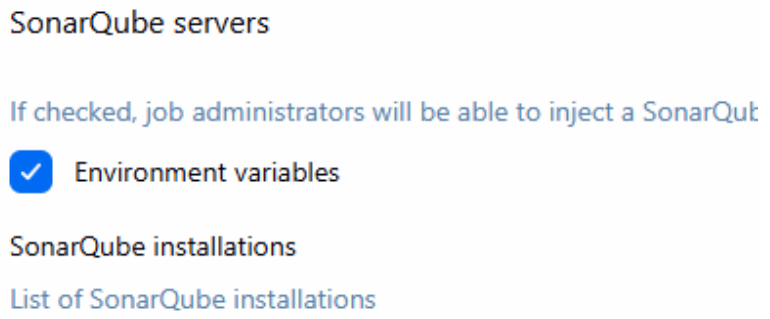
```
sudo systemctl start jenkins
```

```
sudo systemctl status jenkins (after checking status close using ctrl+c)
```

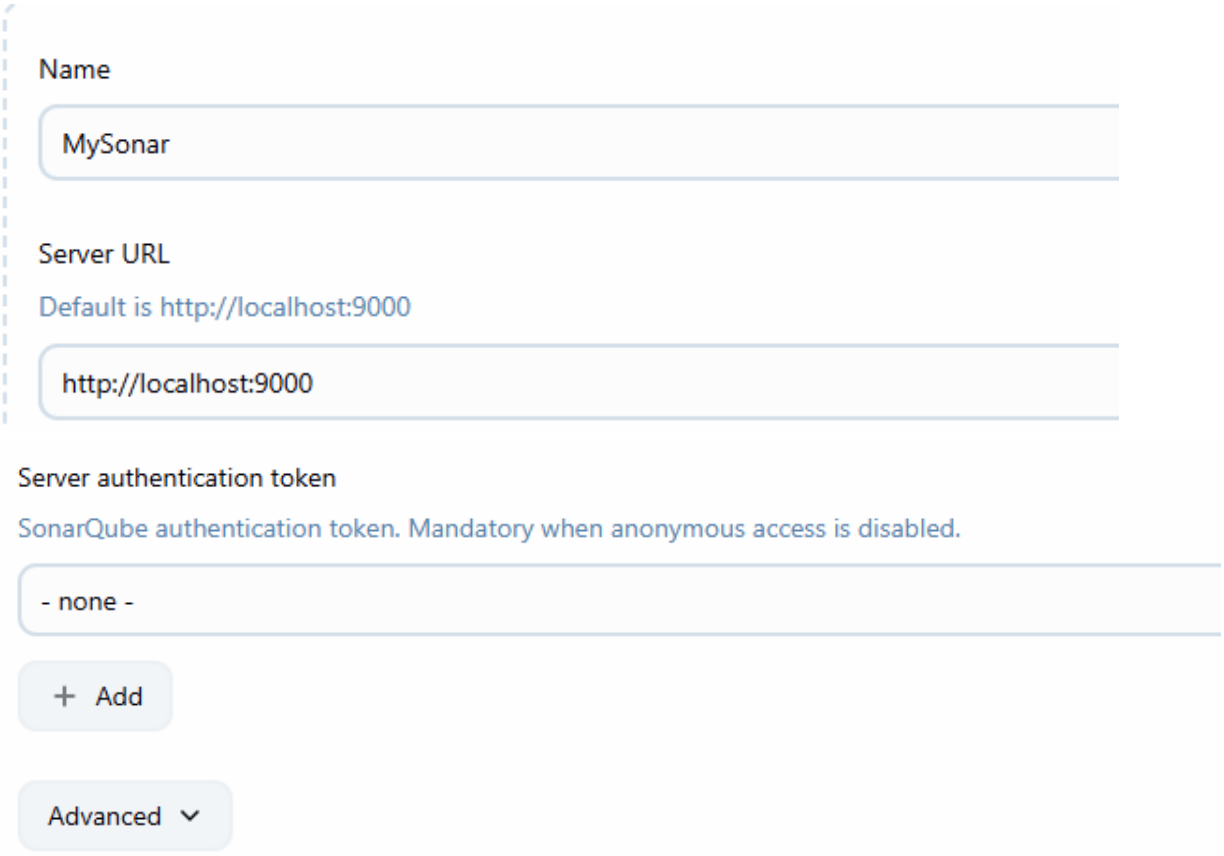
open browser and access http://localhost:8080/login?from=%2F
enter login credentials and continue.
Manage Jenkins --> plugins --> Available Plugins --> SonarQube Scanner for Jenkins --> Install



Now configurations for SonarQube
Manage Jenkins --> System (System Configurations) --> click

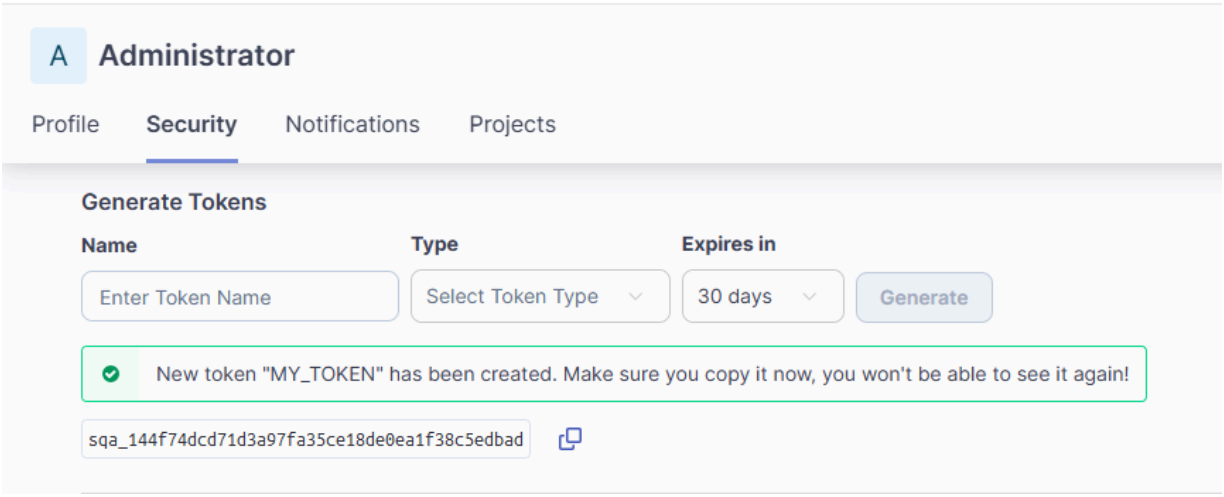


Click on Add SonarQube

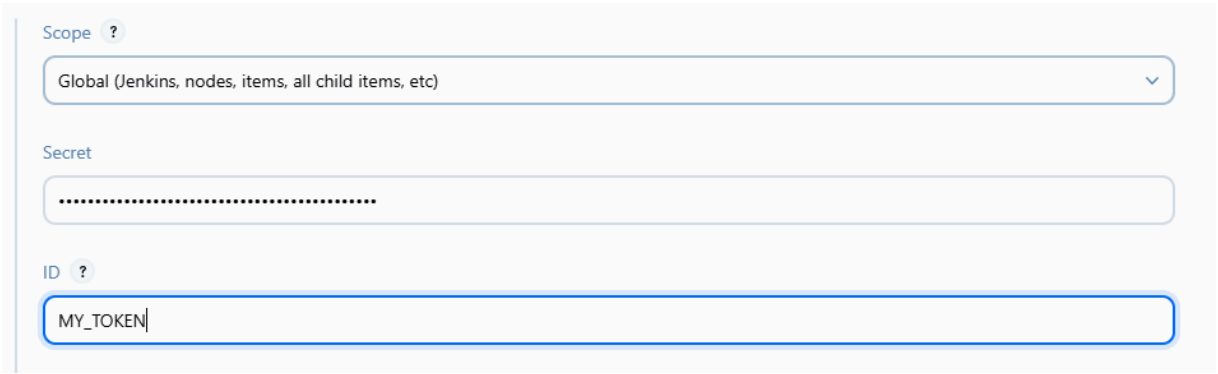


click on Add --> select jenkins for server Authentication token.

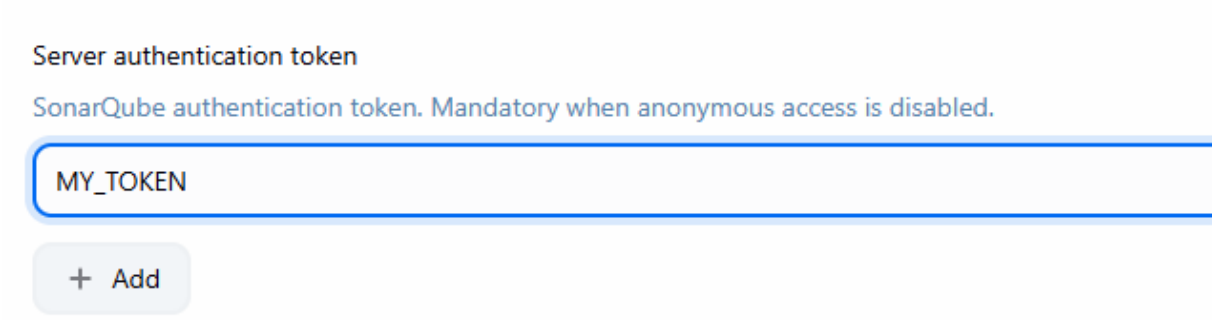
We need to use the Toke n which is generated from SonarQube.
If you don't have token then you can use below steps to generate token
Go to Sonar --> MyAccount -->Security
Give Token Name MY_TOKEN -- type global analysis token click on generate.



This token we need to use in Jenkins.



Choose the created Token from the dropdown.



Let's Download Sonar Scanner also in opt1 folder.

- Sudo wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-5.0.1.3006-linux.zip
- unzip sonar-scanner-cli-5.0.1.3006-linux.zip
- sudo mv sonar-scanner-cli-5.0.1.3006-linux/ sonar-scanner

Now the Configuration is Done and we need to setup a Project and Jenkins Pipeline for the same.

Pipeline Script

```
pipeline {
agent any
environment{
SONARQUBE_ENV='MySonar'
}
stages {
stage('Checkout') {
steps {
git url: 'https://github.com/sonam-niit/python-project-sonar.git',branch:
'main'
}
}
stage('SonarQube Analysis'){
steps{
withSonarQubeEnv("${SONARQUBE_ENV}"){
sh '''
/opt1/sonar-scanner/bin/sonar-scanner \
-Dsonar.projectKey=Sonar_Python \
-Dsonar.sources=app \
-Dsonar.tests=tests \
-Dsonar.python.version=3.10 \
-Dsonar.login=${MY_TOKEN}
'''
}
}
}
stage('Quality Gate'){
steps{
timeout(time: 2, unit: 'MINUTES'){
waitForQualityGate abortPipeline: true
}
}
}
}
}
```

save and try to build
Once the build done you can access your project in sonarqube server
http://localhost:9000/project/issues?id=Sonar_Python