Sonarqube and jenkins integration

A : In sonarqube

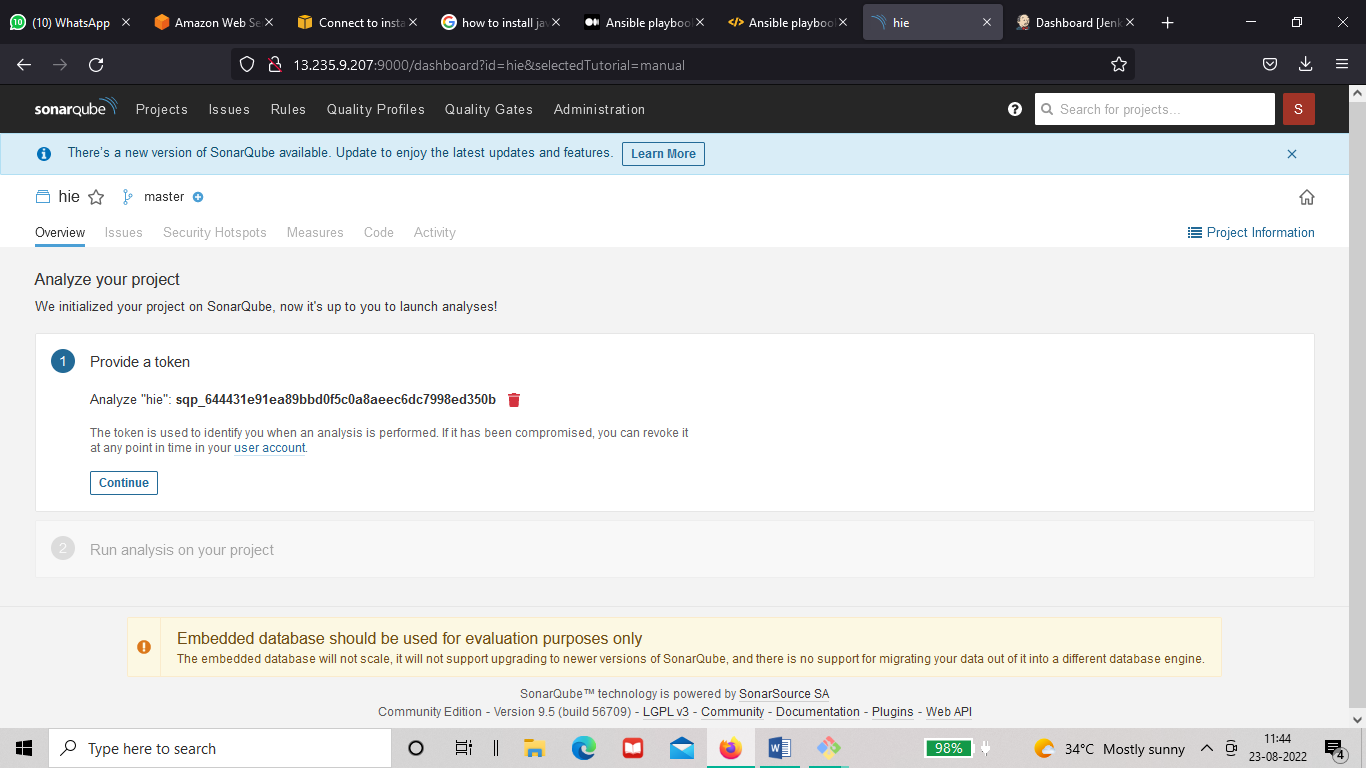
1 : click on CREATE PROJECT -> MANUALLY

2 : give “project display name” and “project key name” and click SETUP

3 : In the next following page click on LOCALLY

4 : now click on GENERATE and copy the token and keep it in a text document, so that u can refer later.

5 : click CONTINUE.



B : In jenkins

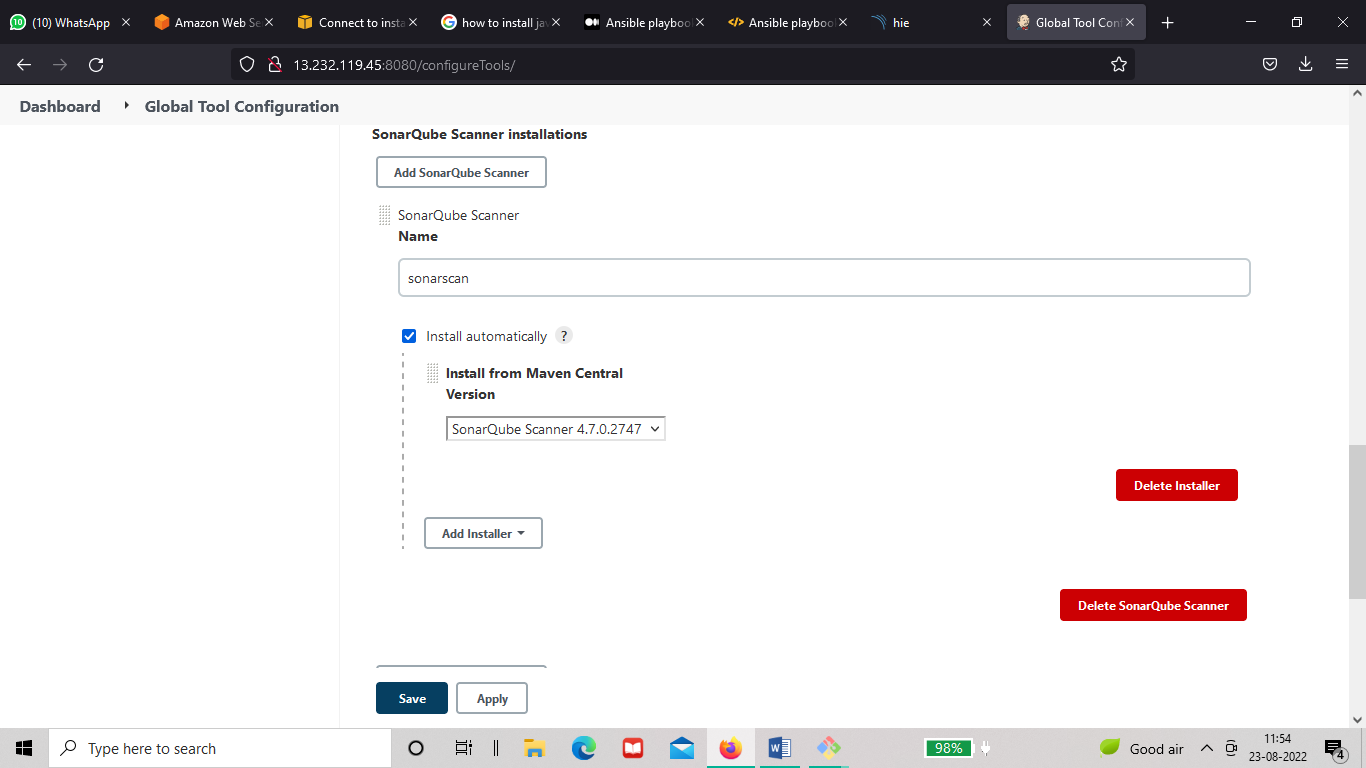
1 : go to manage jenkins -> manage plugins -> available -> search for “sonarqube scanner “

2 : install the above plugin without restart

3 : go back to jenkins dashboard -> manage jenkins -> global tool configuration

4 : In global tool configuration search for sonarqube scanner -> sonarqube scanner installations -> add

Sonarqube -> give it a name .Apply and save



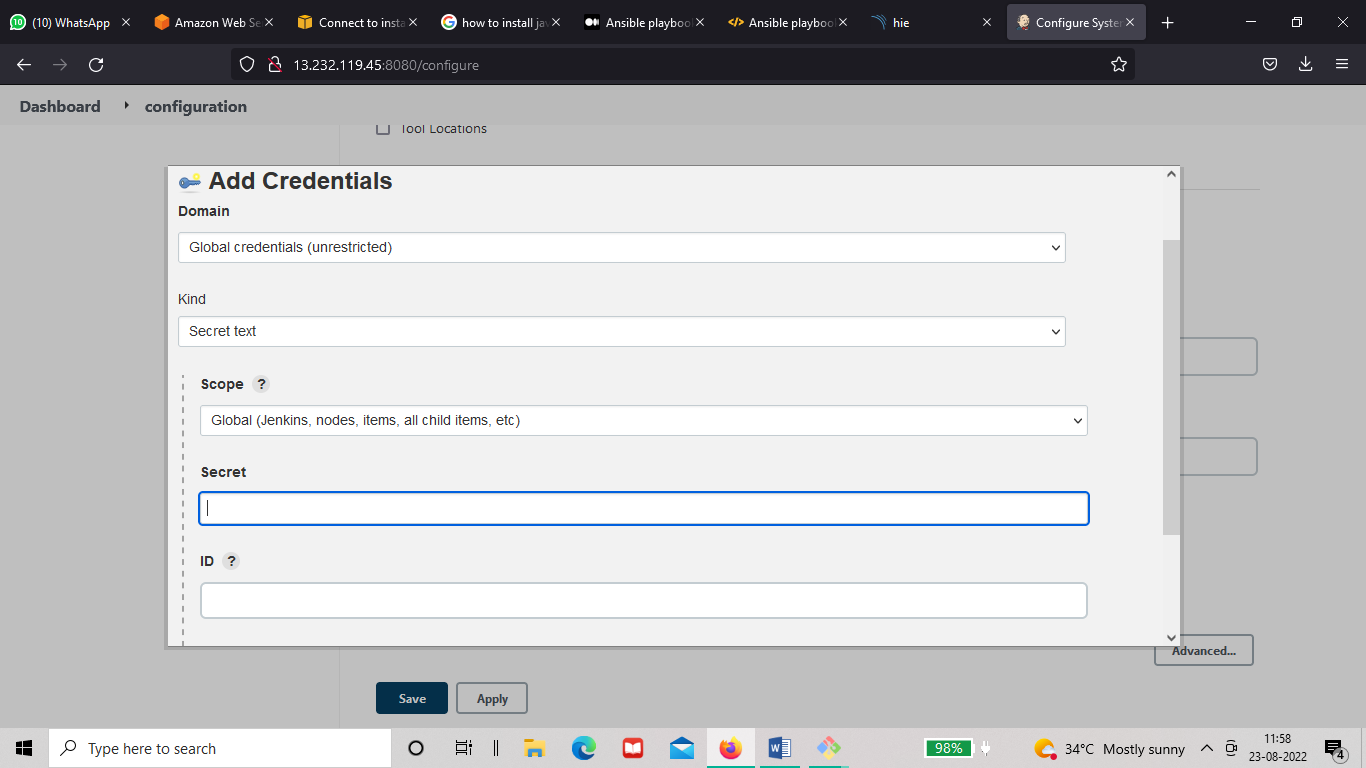
5 : Now go to manage jenkins ->configure system -> sonarqube servers -> click check in the environment variables box -> give a name -> in server url give the url of sonarqube server

6 : In server authentication token -> hover to add and click on the drop down -> click on Jenkins

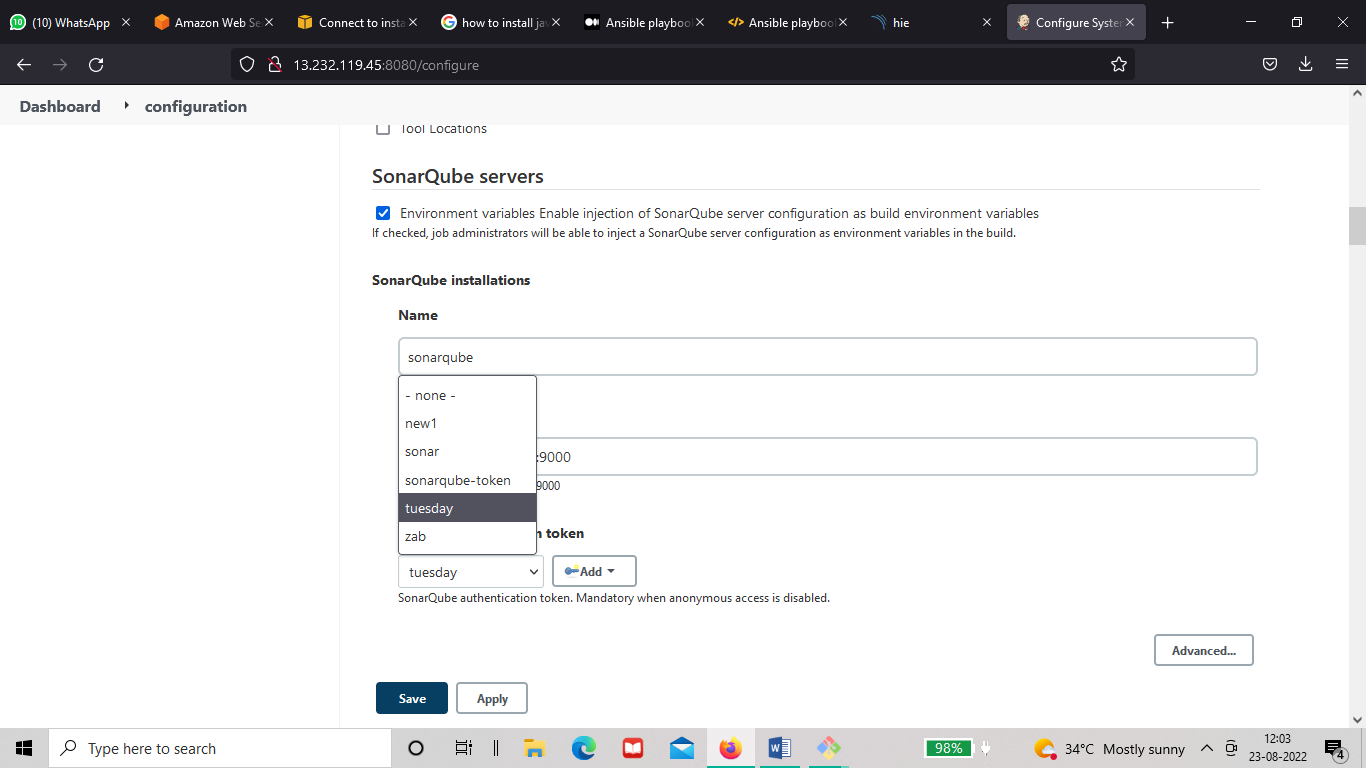
7 : A page as shown below will pop up, now click on kind and click secret text and then in secret

Paste the token that we have copied while creating a project in sonarqube

Give id and description and save



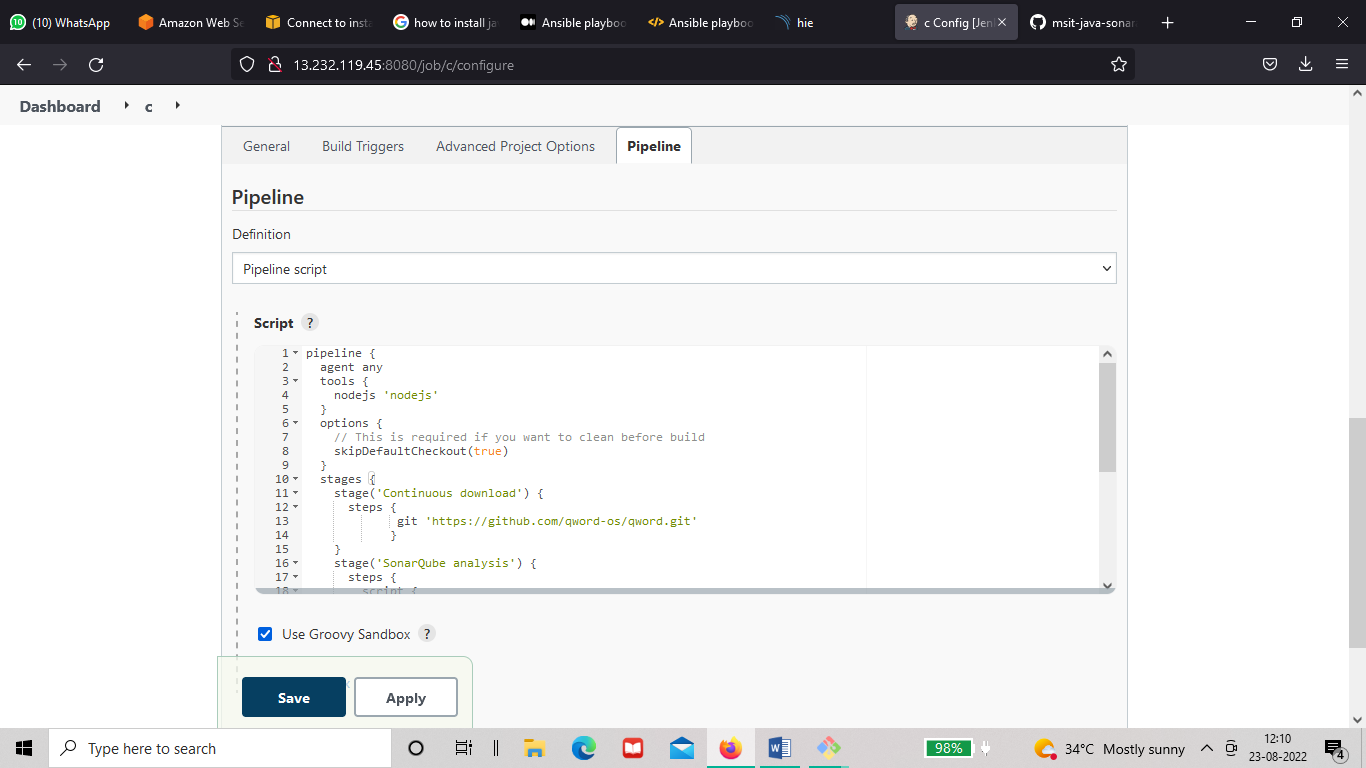
8: now click on the drop down on the left of add key under server authentication token u will see the id name u have given click on it and then apply and save.



C : Creating a pipeline job for sonarqube analysis

1 : go to new item and select pipeline

2 : in the configuration of the pipeline -> under pipeline script paste the below given pipeline



The following pipeline is a java-based pipeline

properties([disableConcurrentBuilds()])

pipeline {

agent any

tools {

nodejs 'nodejs'

}

options {

// This is required if you want to clean before build

skipDefaultCheckout(true)

}

stages {

stage('Continuous download') {

steps {

git ' https://github.com/msit-solutions/msit-java-sonaranalysis.git '

}

}

stage('SonarQube analysis') {

steps {

script {

def scannerHome = tool 'sonarscan'; (replace sonarscan with the name in globaltool for sonarqube)

withSonarQubeEnv('sonarqube') { (replace sonarqube with the name in configuresystem for sonarqube)

sh "${tool("sonarscan")}/bin/sonar-scanner \ (replace sonarscan with globaltool name like above)

-D sonar.login=admin \ (give the username of sonarqube in place of admin)

-D sonar.password=sana \ (and password of sonarqube in place of sana )

-D sonar.projectKey=new \ (give the projectkeyname u have created in place of new)

-D sonar.exclusions=vendor/\*\*,resources/\*\*,\*\*/\*.java \

-D sonar.host.url=http://65.2.4.95:9000/" (give the sonarqube server here)

}

}

}

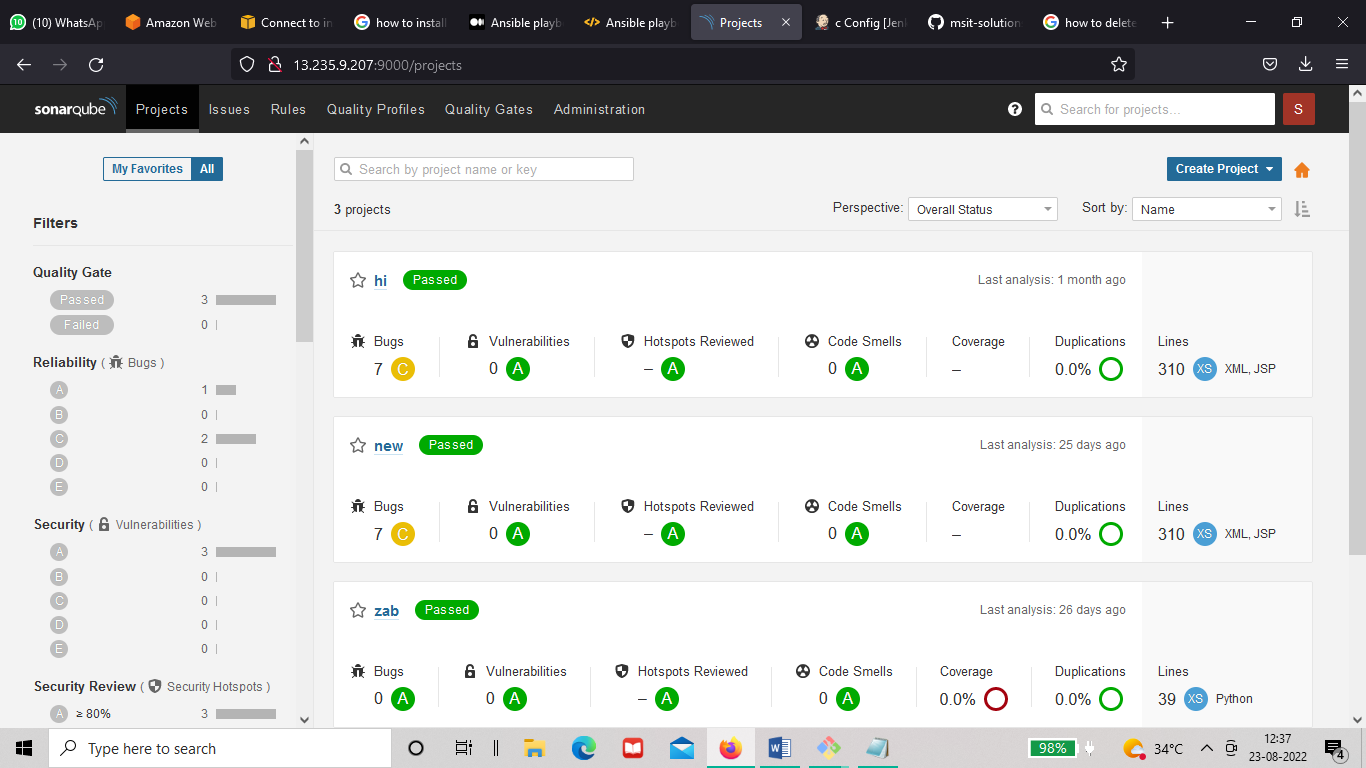
}

}

}

\*In the above pipeline the coloured sentences under the brackets should not be copied with the pipeline \*

Now run the job, the output in the sonarqube should look something like this.



Documented by

Rafiya Sana