Sonaqube

>> Always verify webhook, ip address of jenkins and ip address in servers and ip address everywhere between labs

<https://stackoverflow.blog/2021/11/01/why-solid-principles-are-still-the-foundation-for-modern-software-architecture/>

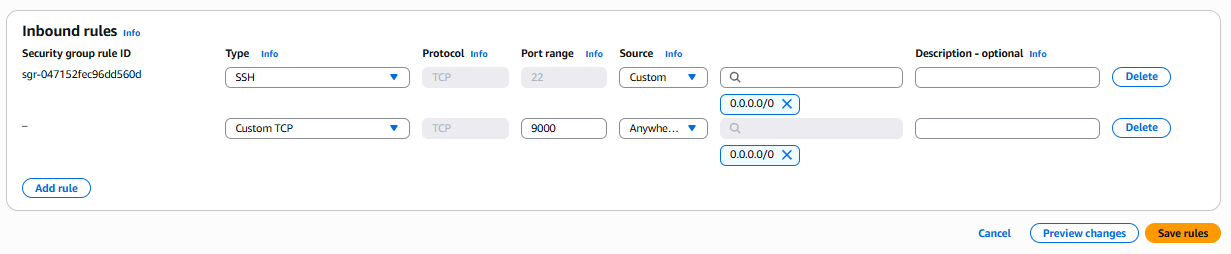
<https://developer.ibm.com/articles/creating-a-12-factor-application-with-open-liberty/>

<https://www.blackduck.com/glossary/what-is-owasp-top-10.html>

Steps to work with Sonarqube code analyzer in Jenkins

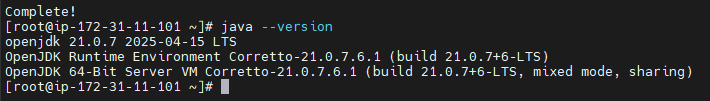
Goto EC2 >> Launch instance >> sonarqube-serverJune2025-amazonlinux-t2.medium-ping098(key) >> Create

Goto Nexus-serverJune2025 >> Security >> Edit inbound rules >> Enable sonarqube-SG with port number open, 22, 9000



*>> Install java version as jenkinsmaster to avoid java inconcurrency*

* sudo yum install java-21\*



OR

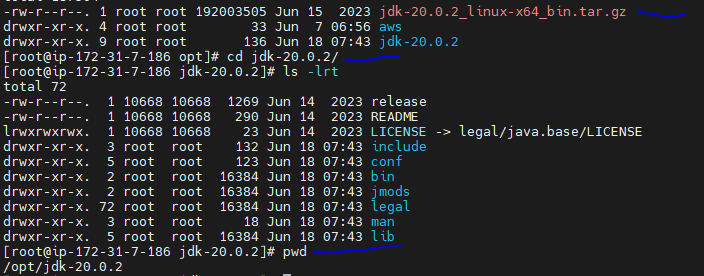
Always use below option

*>> Check java variables in devopsmaster for verification and create java variables in Nexus too*

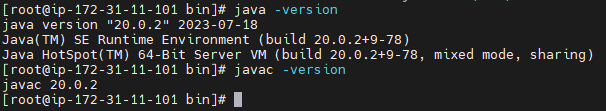
* cd /opt
* Wget -c --header "Cookie: oraclelicense=accept-securebackup-cookie" <https://download.oracle.com/java/20/archive/jdk-20.0.2_linux-x64_bin.tar.gz>

>> Downloads -rw-r--r--. 1 root root 192003505 Jun 15 2023 jdk-20.0.2\_linux-x64\_bin.tar.gz

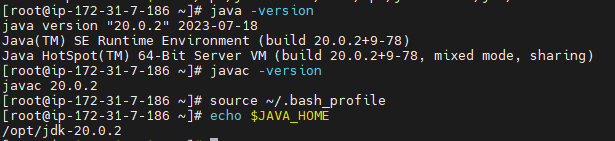
* tar -zxvf jdk-20.0.2\_linux-x64\_bin.tar.gz
* cd /opt/jdk-20.0.2
* ls –lrt



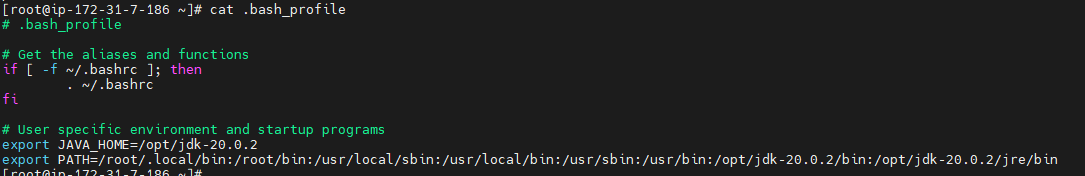
* alternatives --install /usr/bin/java java /opt/jdk-20.0.2/bin/java 2
* alternatives --config java
* alternatives --config javac
* alternatives --install /usr/bin/jar jar /opt/jdk-20.0.2/bin/jar 2
* alternatives --install /usr/bin/javac javac /opt/jdk-20.0.2/bin/javac 2
* alternatives --set jar /opt/jdk-20.0.2/bin/jar
* alternatives --set javac /opt/jdk-20.0.2/bin/javac



* cd
* echo "export JAVA\_HOME=/opt/jdk-20.0.2" >> .bash\_profile
* echo "export PATH=$PATH:/opt/jdk-20.0.2/bin" >> .bash\_profile
* source ~/.bash\_profile
* echo $JAVA\_HOME
* java –version
* javac –version



* cat .bash\_profile ……verify path being set in bash profile

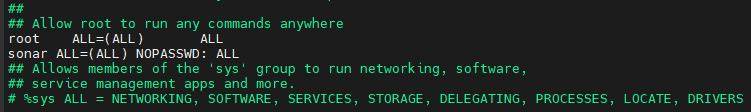


*Add the user sonar and Give sudo accecss to sonar*

* useradd sonar
* visudo

sonar ALL=(ALL) NOPASSWD: ALL



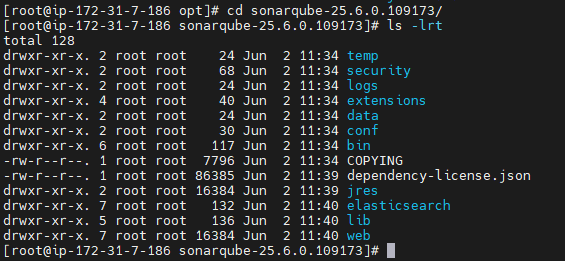


*Download sonarqube*

* wget <https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-25.6.0.109173.zip>

>>Downloads -rw-r--r--. 1 root root 857201296 Jun 2 13:59 sonarqube-25.6.0.109173.zip

* unzip sonarqube-25.6.0.109173.zip

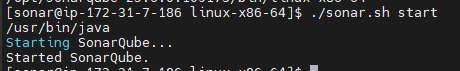


>> Change the owner and group

* chown -R sonar:sonar /opt/sonarqube-9.8.0.63668
* sudo su – sonar

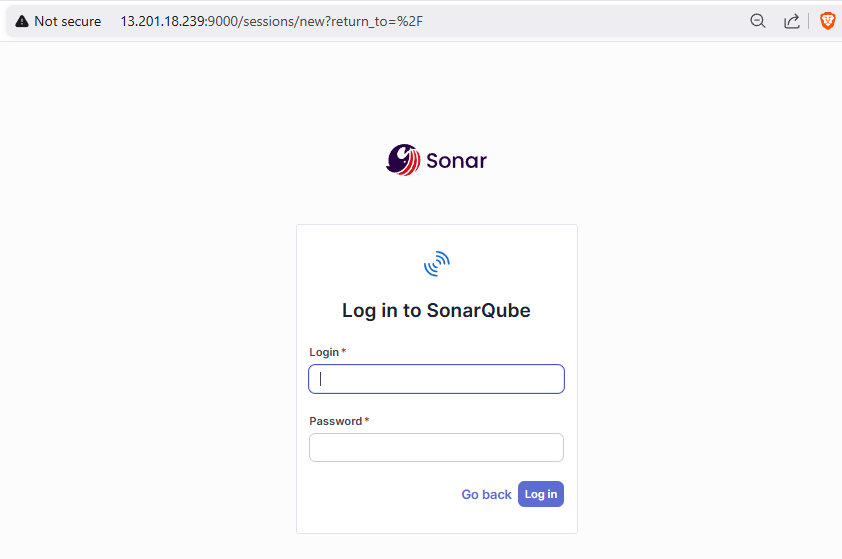
>> Start the service sonar

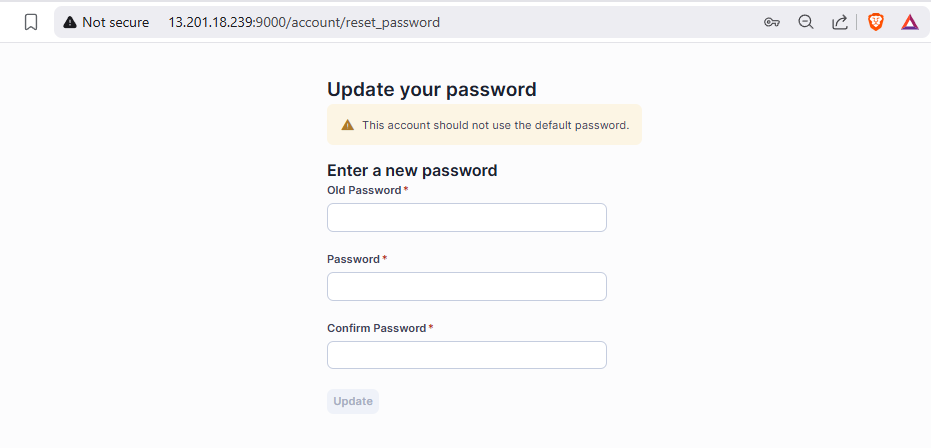
* cd /opt/sonarqube-25.6.0.109173/bin/linux-x86-64
* ./sonar.sh start



>> login to sonarqube in browser

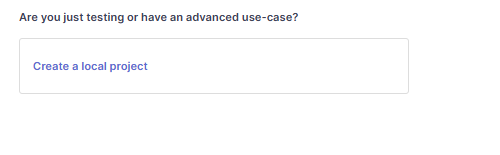
* publicip:9000
* u:admin
* p:admin

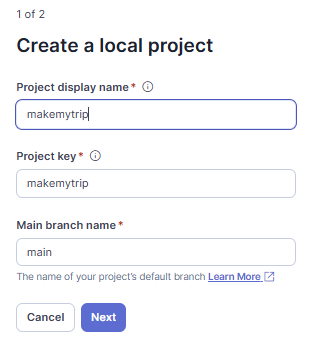


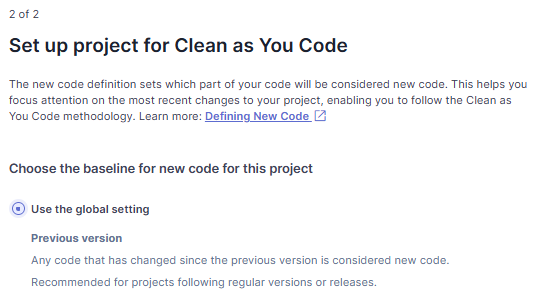


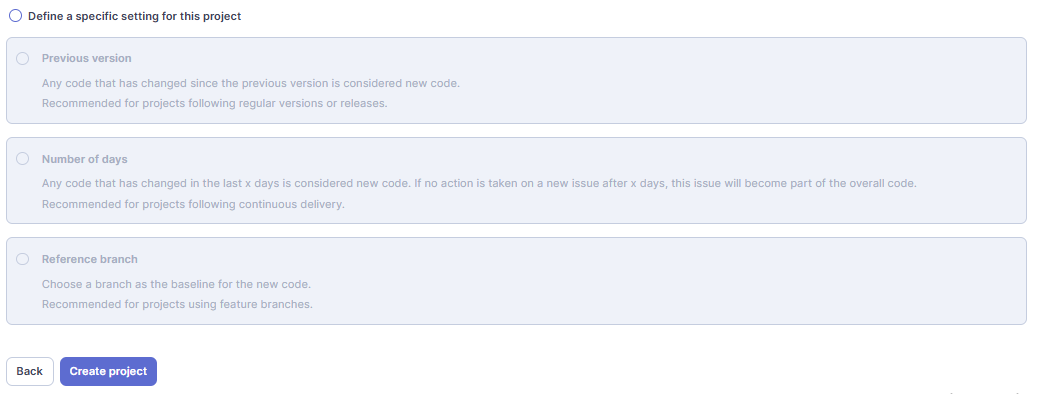
>> New password=sonarQube@123

>> Click on projects and click on manually



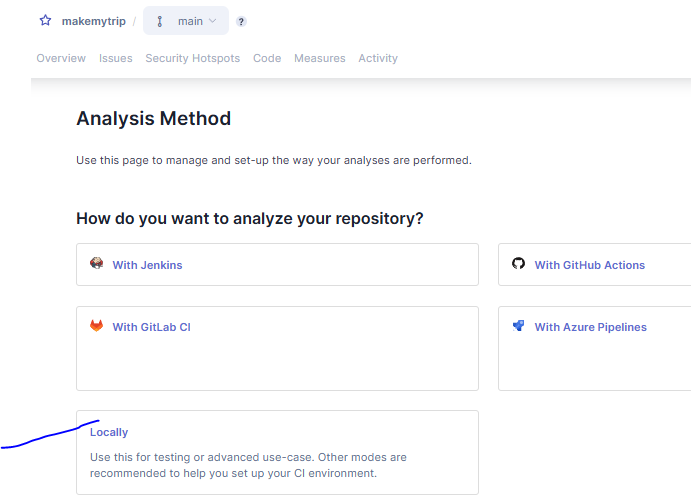




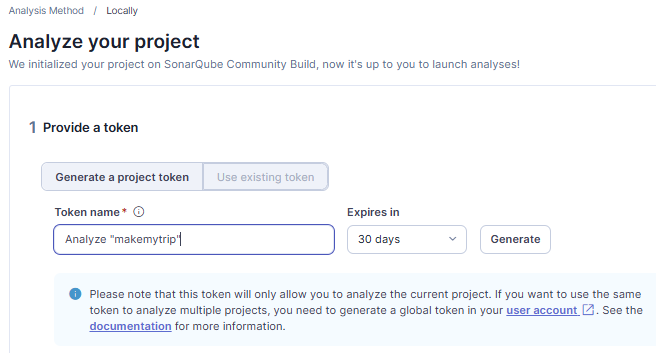


>> Click on create project

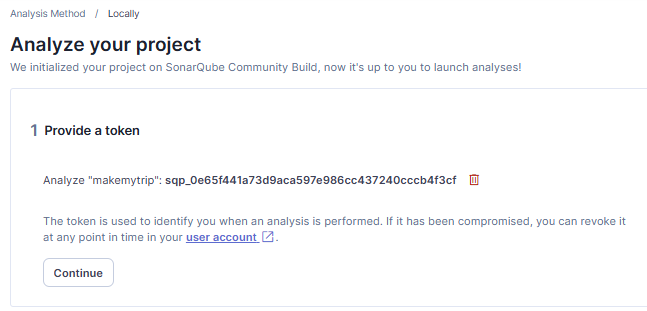
>> Click on Locally



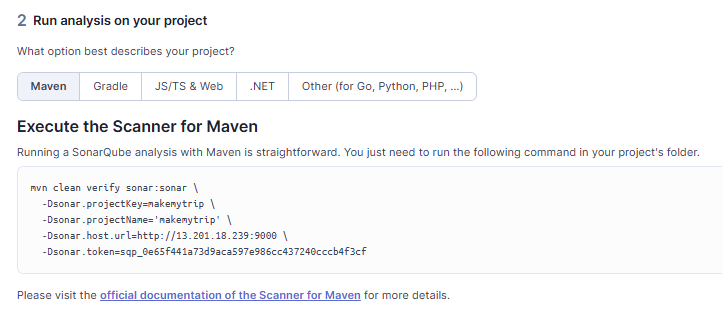
>> Click on Generate



>> Click on continue



>> Select Maven

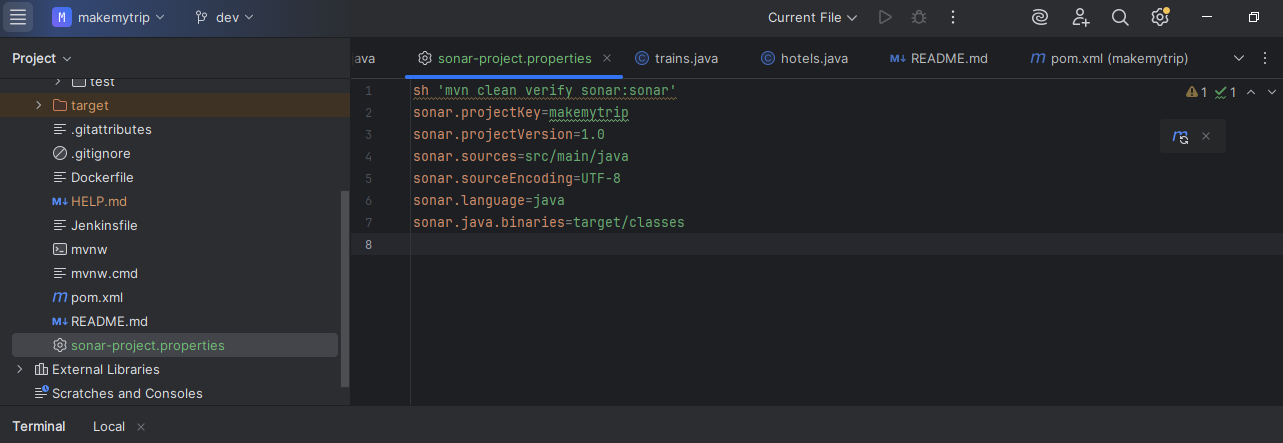


>> Goto intelij Idea and create a file named sonar-project.properties

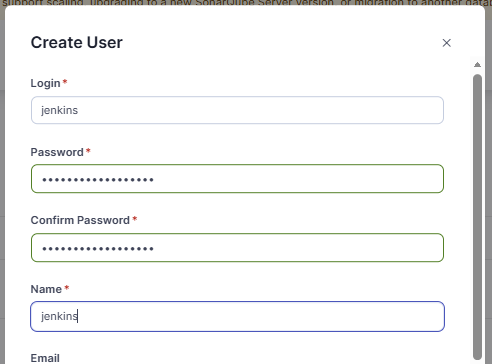
>> Copy the first 2 lines from 2nd step of above screenshot. Remove the \ and -D

And then the below lines need to be pasted under the sonar.login property like the next screenshot.

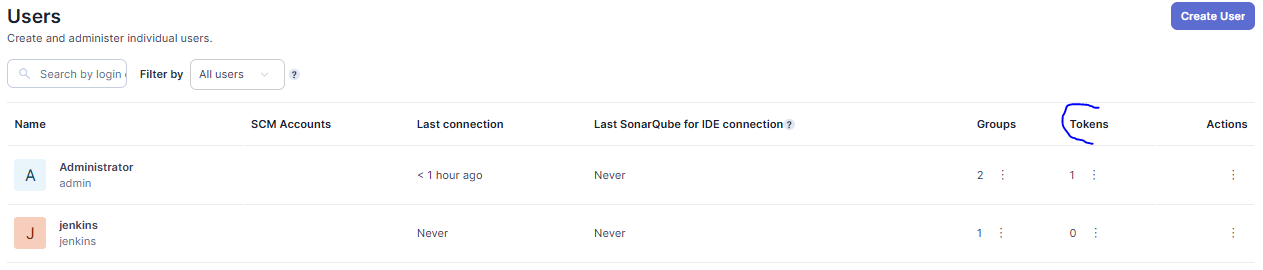
**sonar.projectVersion**=**1.0  
sonar.sources**=**src/main/java  
sonar.sourceEncoding**=**UTF-8  
sonar.language**=**java  
sonar.java.binaries**=**target/classes**



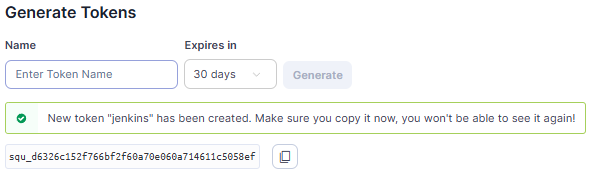
Goto Sonarqube >> Administration >> security >> users >> Create user



Login=Name=jenkins, Password=Jenkinsjenkins@123 >> Create

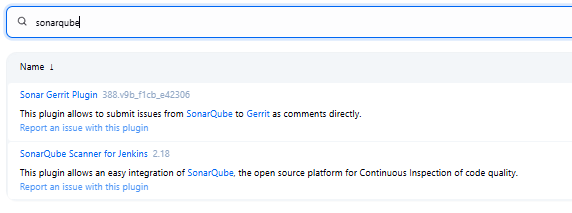


Click on create then click on the token (blue rounded on the below screenshot) to generate token for Jenkins user

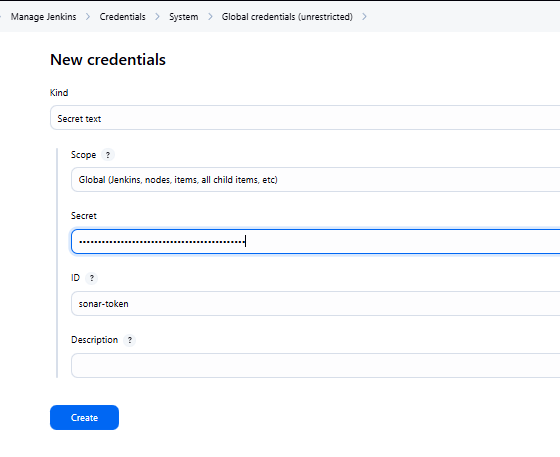


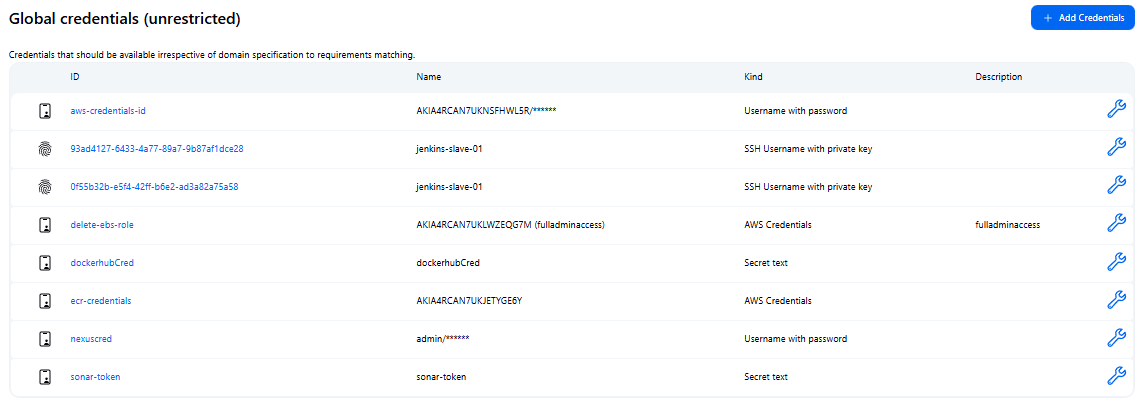
You can click on generate token then you can see on token has been generated. Copy the token to a note pad for future purpose

>> Goto Jenkins and download plugin sonarqube scanner, restart jenkins



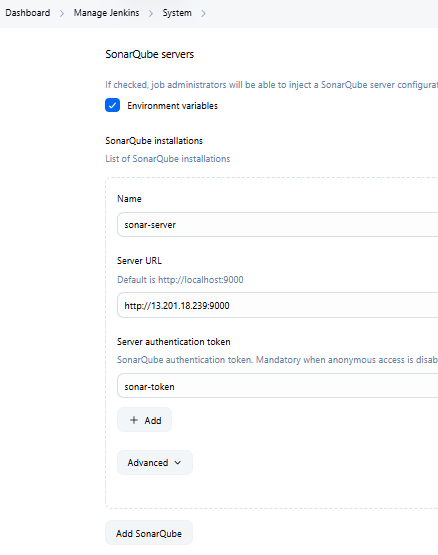
>> Add the global credentials with the token which is earlier generated while creating the user jenkins in SonarQube.





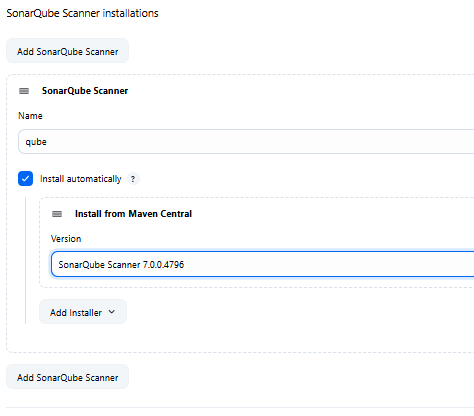
Go to dashboard-> Manage Jenkins ->Configure system

Go to SonarQube servers. Select Environment Variables. Type the name, Server URL of SonarQube. Select the token that we added. And click on save.



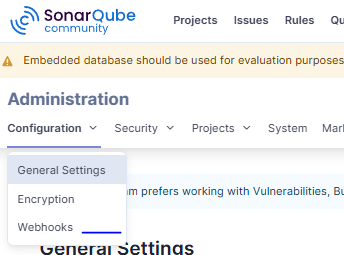
Goto Manage Jenkins->Global Tool Configuration

Goto Sonarqube scanner part. Click on Add SonarQube Scanner then type the name and click on save.

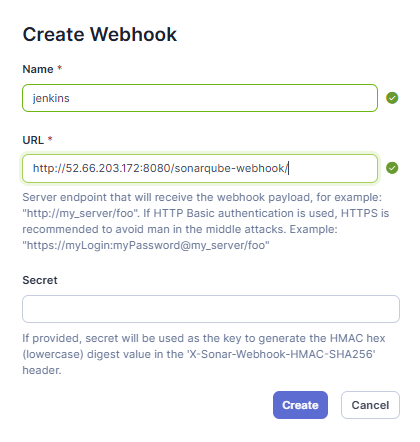


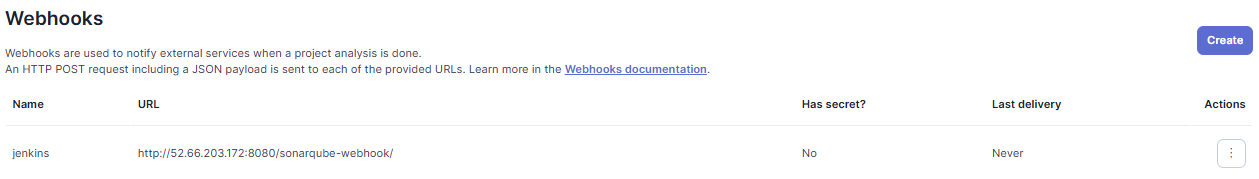
Create web hook for quality gate

Go to SonarQube Server -> Administration -> Configuration -> Webhooks



>> Click on create. Type the Jenkins URL like below screenshot and click on create.

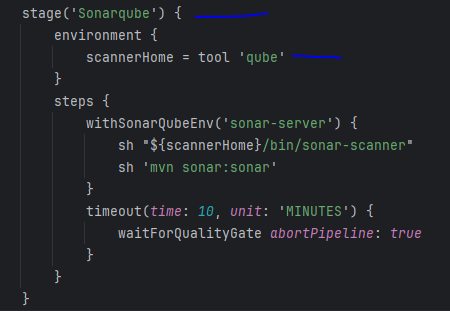




>> Go to Intellij Idea and open Jenkins file.

>> Type the script for sonarqube analysis. (Here qube is the name of sonarqube scanner which is mentioned in Global tool configuration, and sonar-server is the name of sonarqube installation from Configure system.)

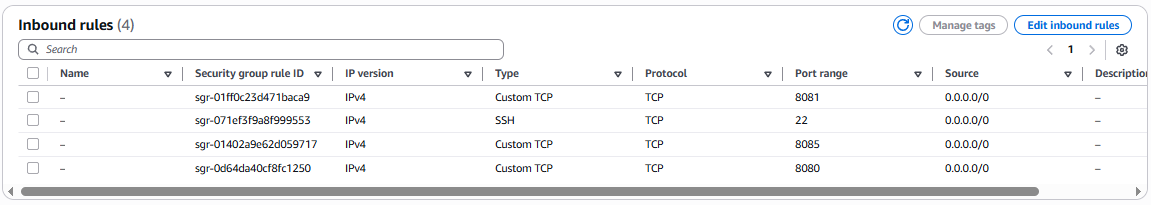
stage(**'Sonarqube'**) {  
 environment {  
 scannerHome **=** tool **'qube'** }  
 steps {  
 withSonarQubeEnv(**'sonar-server'**) {  
 sh **"${scannerHome}/bin/sonar-scanner"** sh **'mvn sonar:sonar'** }  
 timeout(***time***: ***10***, ***unit***: **'MINUTES'**) {  
 waitForQualityGate ***abortPipeline***: ***true*** }  
 }  
}

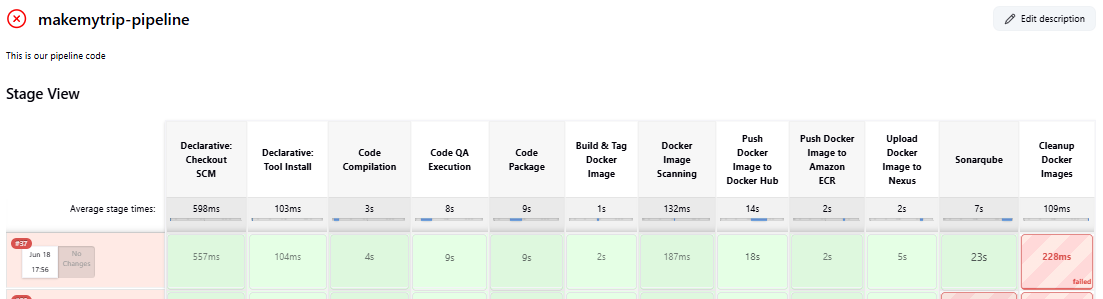


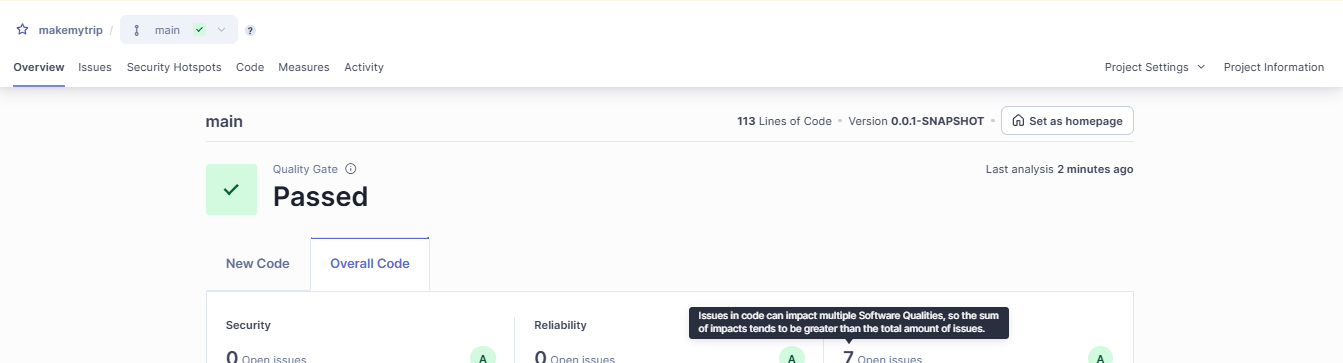
Push the changes to Git hub. And run the project in Jenkins.

There you can see your project has do the sonarqube analysis success fully. You can see an option sonarqube on the left side. You can click and check the overview and code scanning results.

>>Open nexus ports on jenkinsmasterdevops



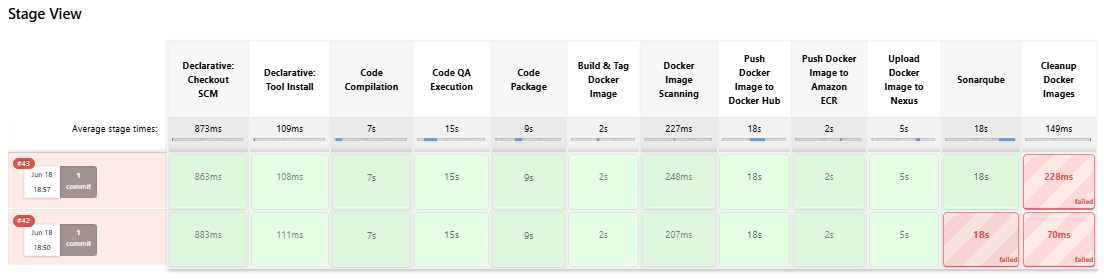




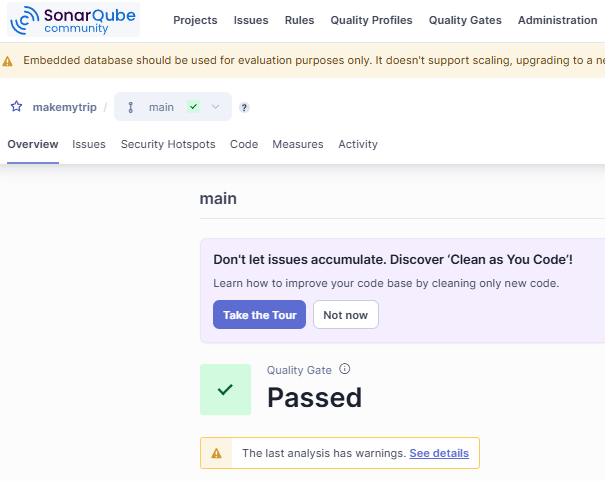
Blueocean stages:



Stage view



>> Sonarqube report



>> Sonarqube repository

