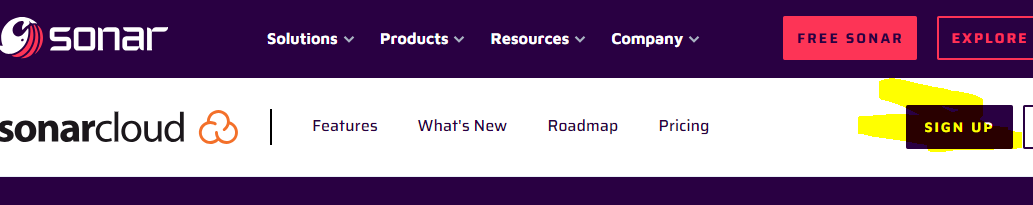
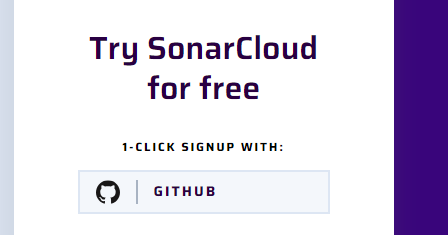
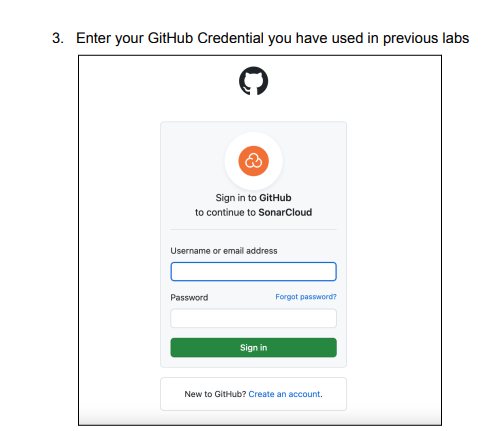
Signup and SetUp - SonarQube Cloud account using GitHub Objective: In this lab, you will signup SonarQube Cloud account using GitHub

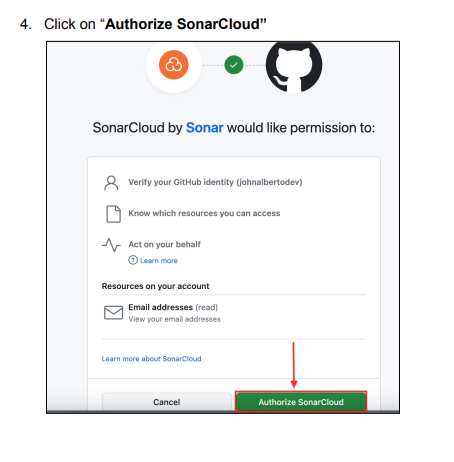
Creating a SonarQube Cloud account using GitHub 1. Go to SonarQube cloud website <https://www.sonarsource.com/products/sonarcloud/>

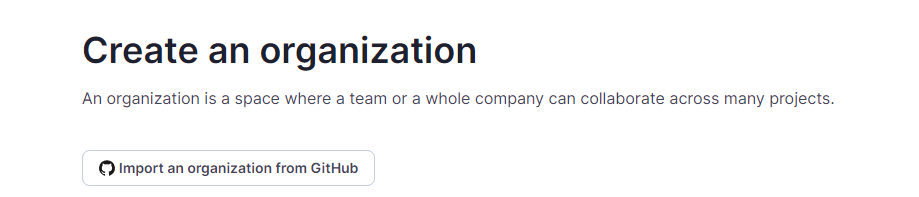
Singup using github

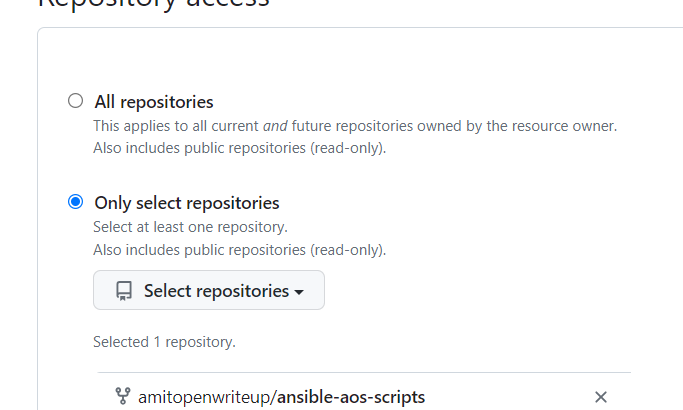




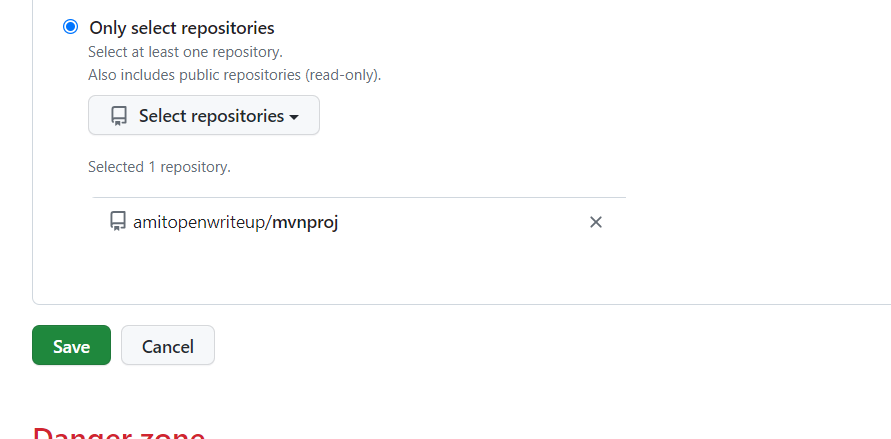




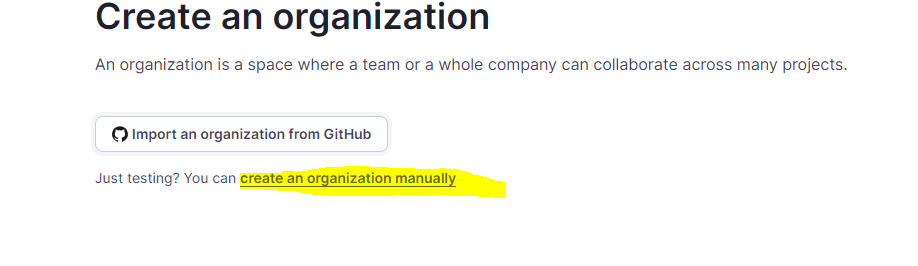
5. 

6. 

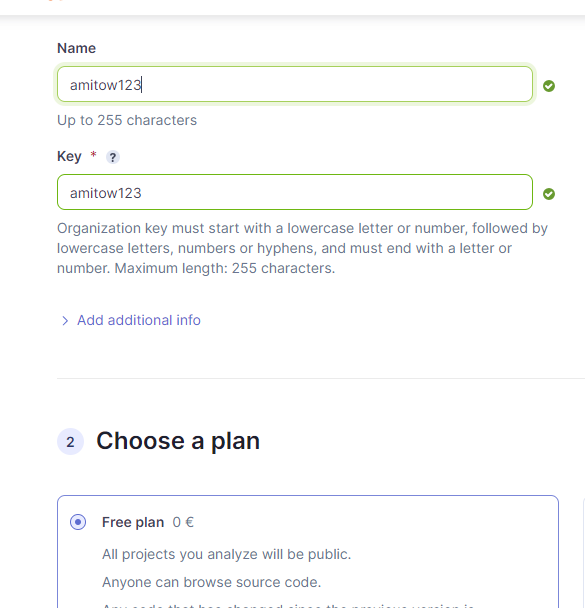
7. Select the ‘Only selected repositories’ and select your repository with name and save

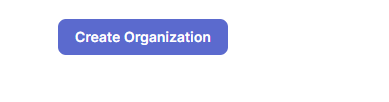


8. Create an organization manually

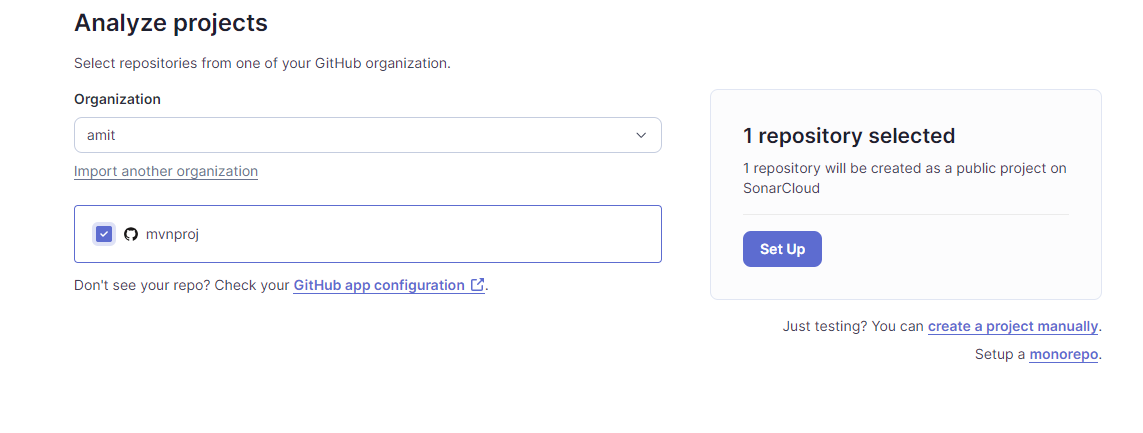


9. Provide unique name and key; choose free plan

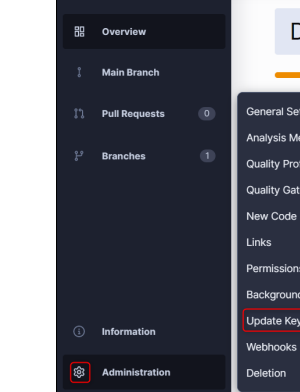
9. 



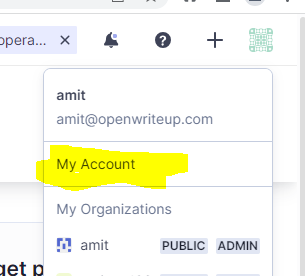
10. Select your repositories and then click on the Set Up button.

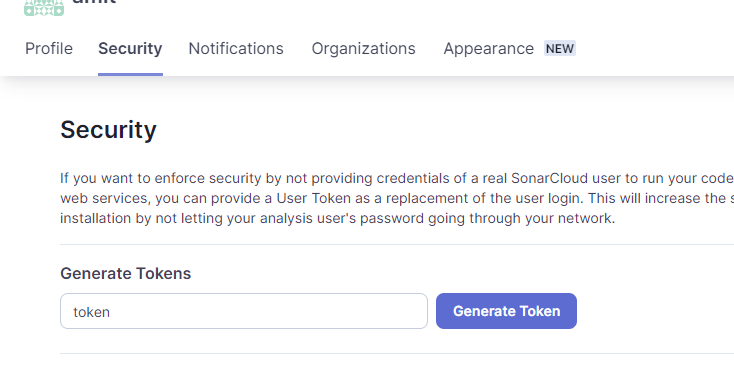


11. Update key



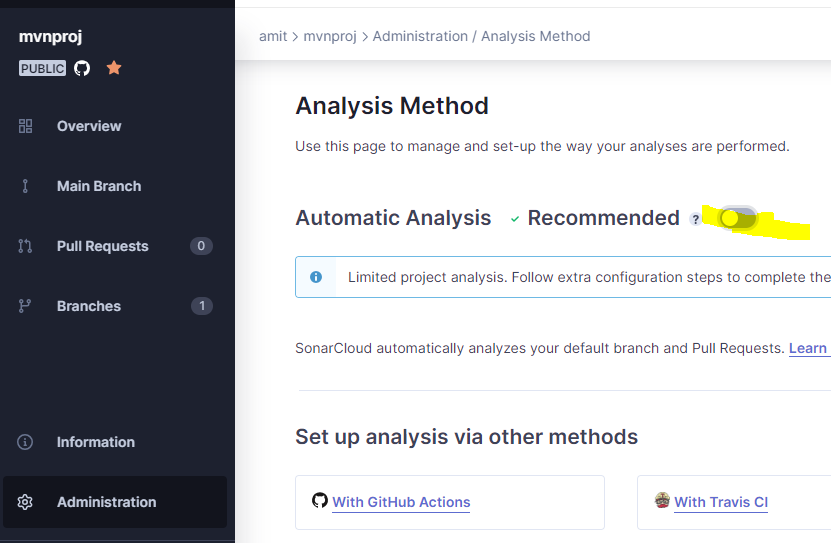
12. Go to the Account section. Click on ‘My Account”



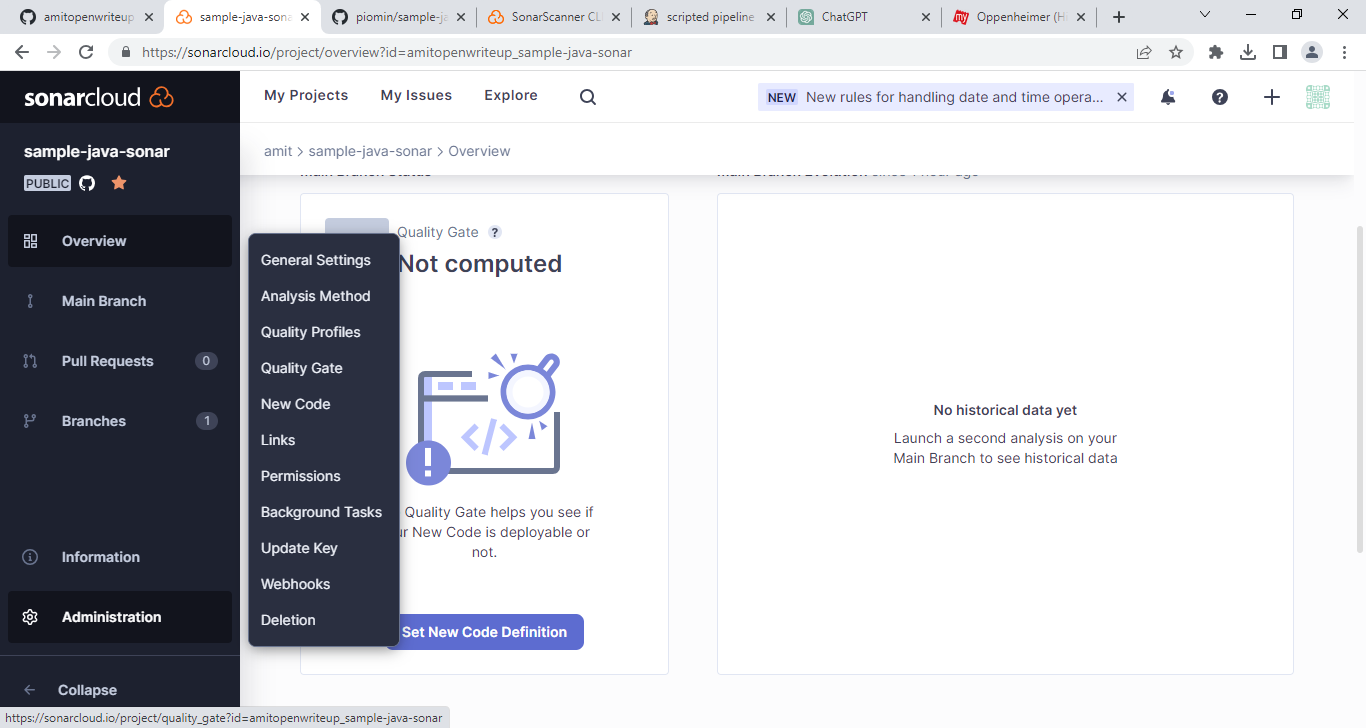


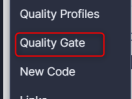
1. Use existing Pipeline ( java-pipeline ) to analyze code quality through SonarQube.

2. Prior to performing a Sonar scan through Jenkins, it is necessary to deactivate the automatic analysis option in the SonarQube web graphical user interface (GUI). a. Go to SonarQube Web GUI and click on “Administration” → “Analysis Method

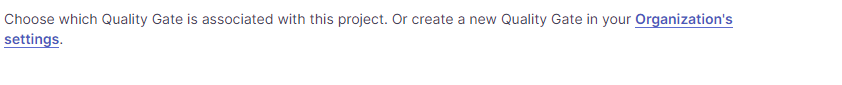


Create Quality gate

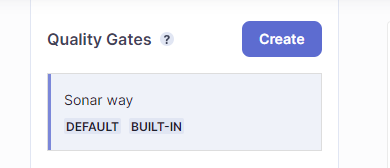


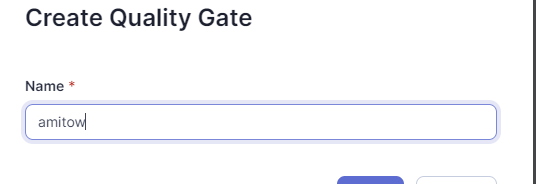


Click on organization setting

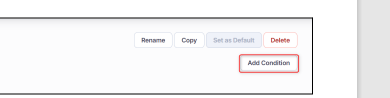


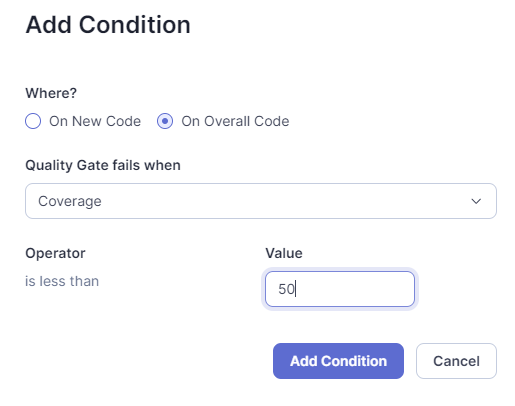
Click on Create



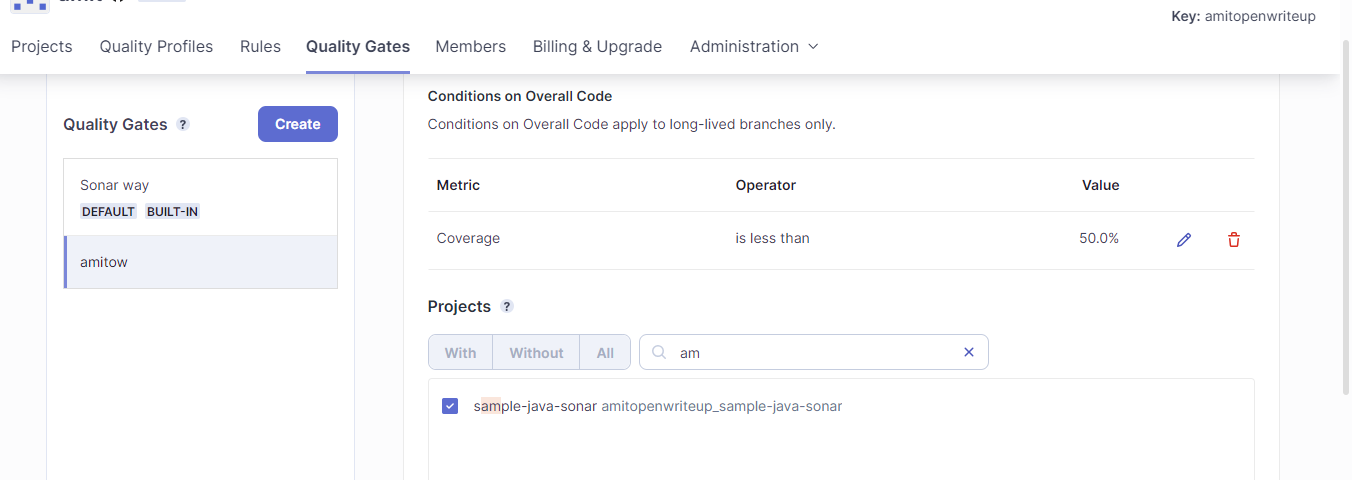


Add condition

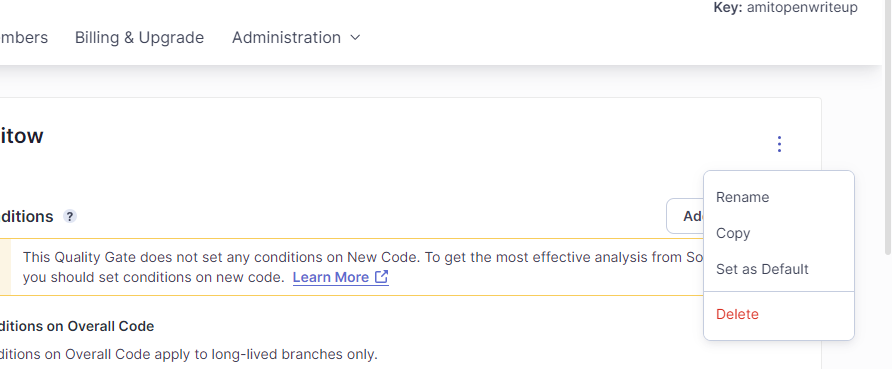




Map the project



Set as default



create pipeline project [ **Please update code with token,projectkey and organization**]

pipeline {

agent any

stages {

stage('chckout scm') {

steps {

checkout scmGit(branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/amitopenwriteup/sample-java-sonar.git']])

}

}

stage('Compiling and Running Test Cases') {

steps {

sh 'mvn clean'

sh 'mvn compile'

sh 'mvn test'

}

}

stage('Generating a Cucumber Reports') {

steps {

script {

// Run Cucumber tests and generate reports

sh 'mvn verify'

}

}

}

stage('Creating Package') {

steps {

sh 'mvn package'

}

}

stage('adding genrerate report'){

steps {

sh 'mvn verify'

}

}

stage('Install sonarqube cli') {

steps {

// Step to install SonarQube CLI

sh 'wget -O sonar-scanner.zip https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.7.0.2747-linux.zip'

sh 'unzip -o -q sonar-scanner.zip'

sh 'sudo rm -rf /opt/sonar-scanner'

sh 'sudo mv --force sonar-scanner-4.7.0.2747-linux /opt/sonar-scanner'

sh 'sudo sh -c \'echo "#/bin/bash \nexport PATH=\\\"$PATH:/opt/sonar-scanner/bin\\\"" > /etc/profile.d/sonar-scanner.sh\''

sh 'chmod +x /opt/sonar-scanner/bin/sonar-scanner'

sh '. /etc/profile.d/sonar-scanner.sh'

}

}

stage('Analyzing Code Quality') {

steps {

// Step to analyze code quality with SonarQube

sh '/opt/sonar-scanner/bin/sonar-scanner -Dsonar.projectKey=amitopenwriteup\_sample-java-sonar -Dsonar.organization=amitopenwriteup -Dsonar.qualitygate.wait=true -Dsonar.qualitygate.timeout=300 -Dsonar.sources=src/main/java/ -Dsonar.java.binaries=target/classes -Dsonar.host.url=https://sonarcloud.io -Dsonar.login=5e608b307d89c17066fe612b7ad418857b10c5f8'

}

}

}

}

stage ('Analyzing Code Quality') {

steps {

sh '/opt/sonar-scanner/bin/sonar-scanner

-Dsonar.projectKey=amitopenwriteup\_mvnproj

-Dsonar.organization=amit -Dsonar.qualitygate.wait=true

-Dsonar.qualitygate.timeout=300 -Dsonar.sources=src/main/java/

-Dsonar.java.binaries=target/classes -Dsonar.host.url=https://sonarcloud.io

-Dsonar.login=5e608b307d89c17066fe612b7ad418857b10c5f8'

}

}

https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.7.0.2747-linux.zip?\_gl=1\*13zpo4s\*\_ga\*MTY5MTM2Nzk3NC4xNjgzODYzMDQw\*\_ga\_9JZ0GZ5TC6\*MTY4OTY5NTc2NC40LjEuMTY4OTY5ODA1Mi40My4wLjA.