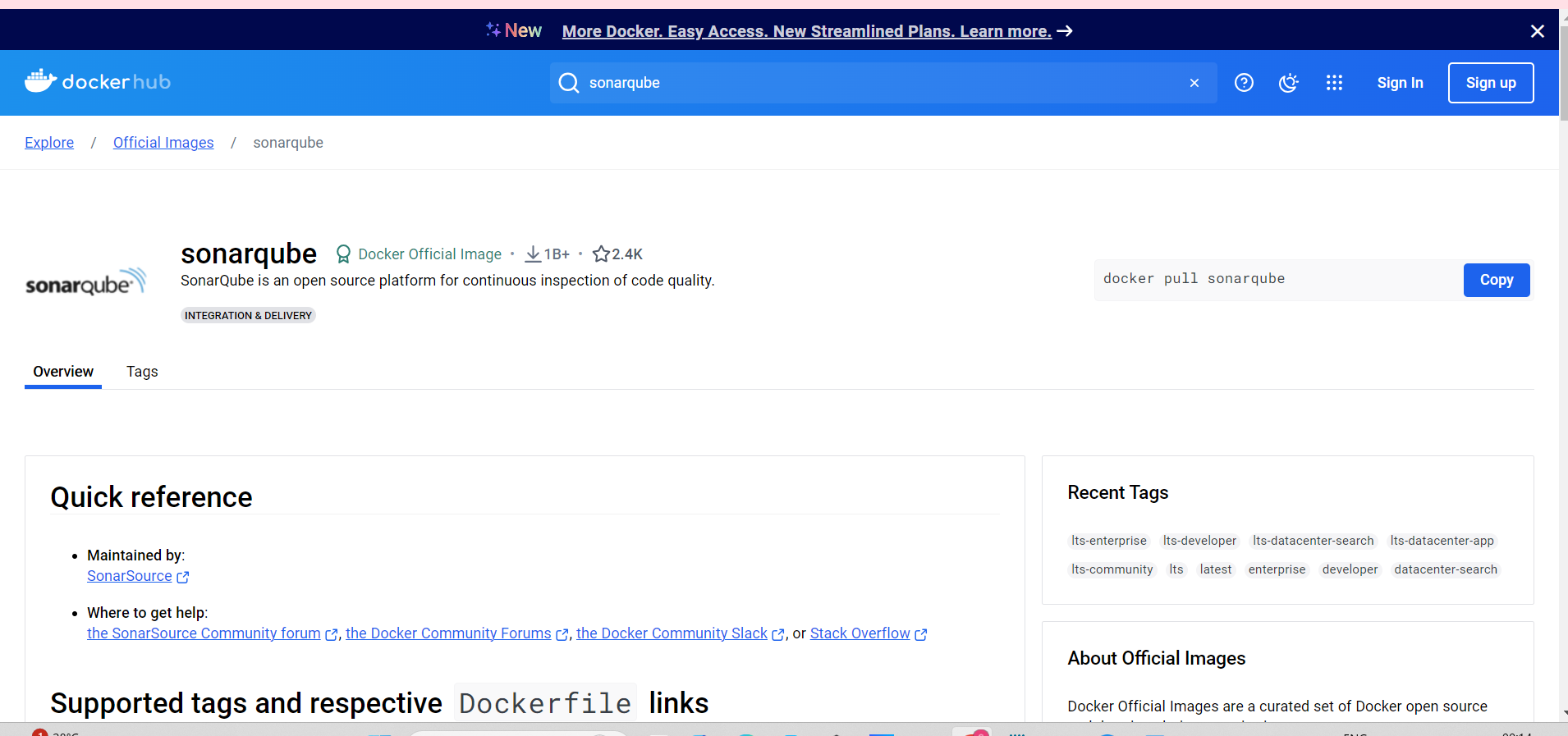
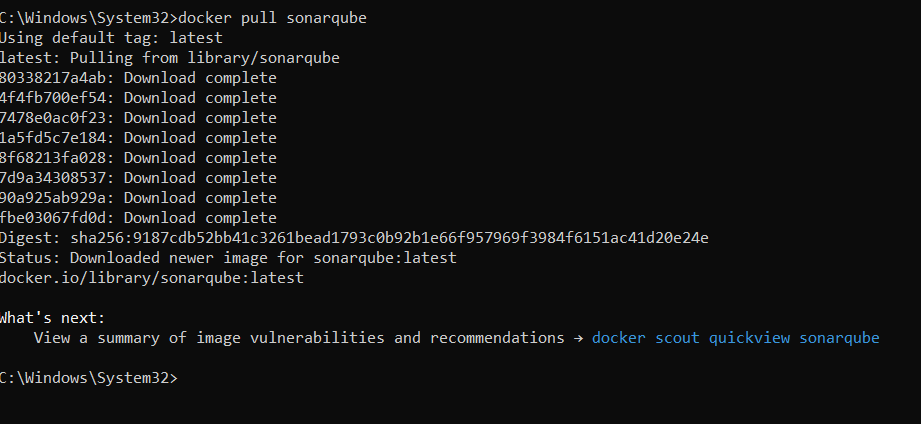


Using docker hub pull sonarcube

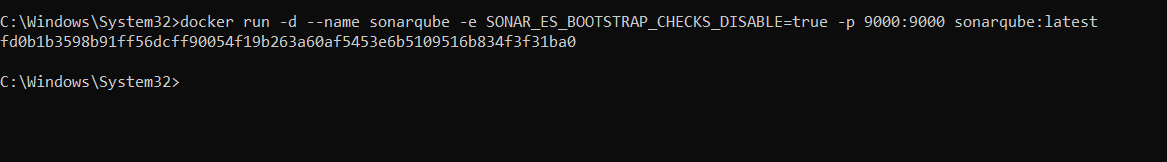


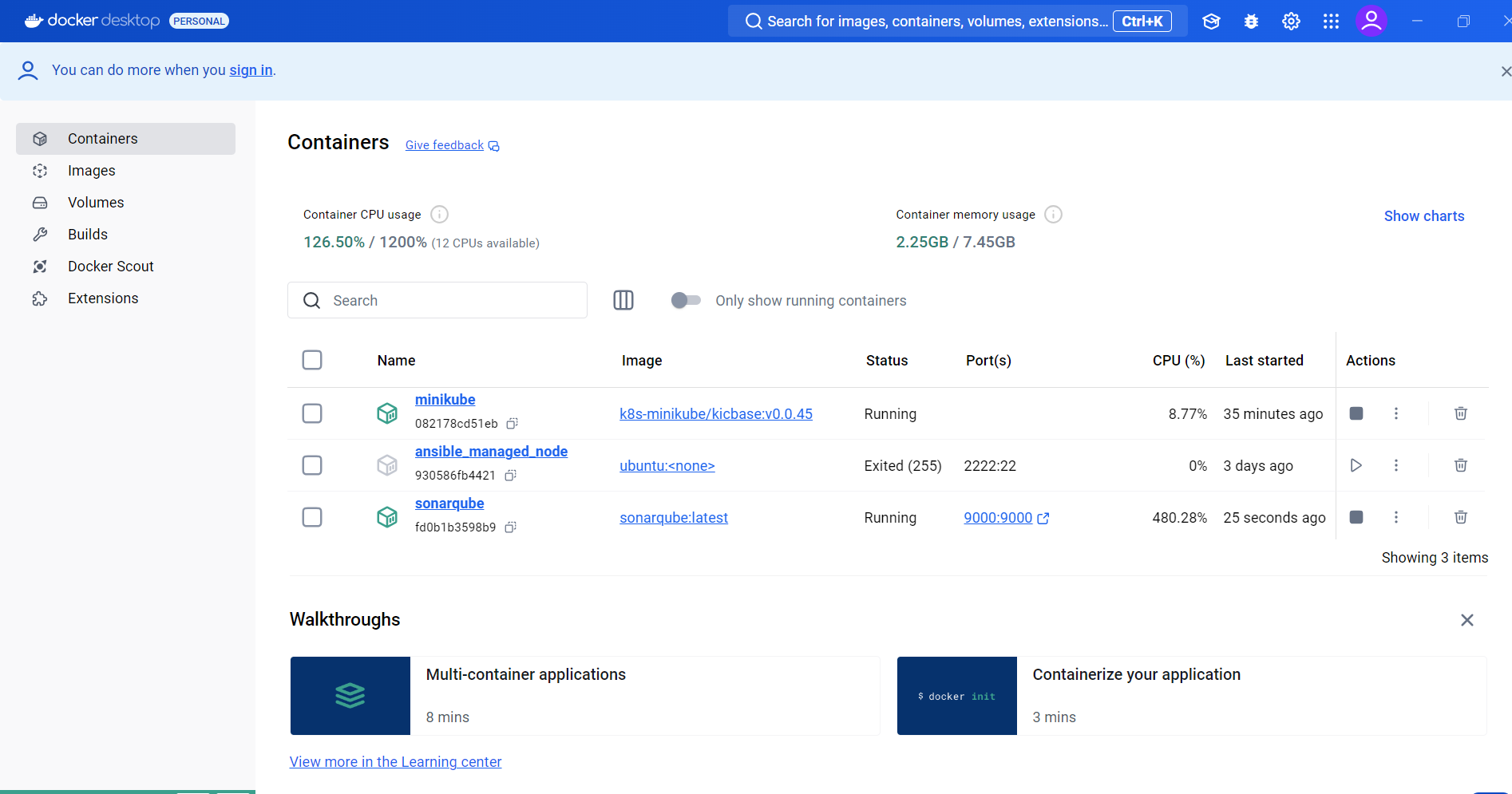
After successful pull :



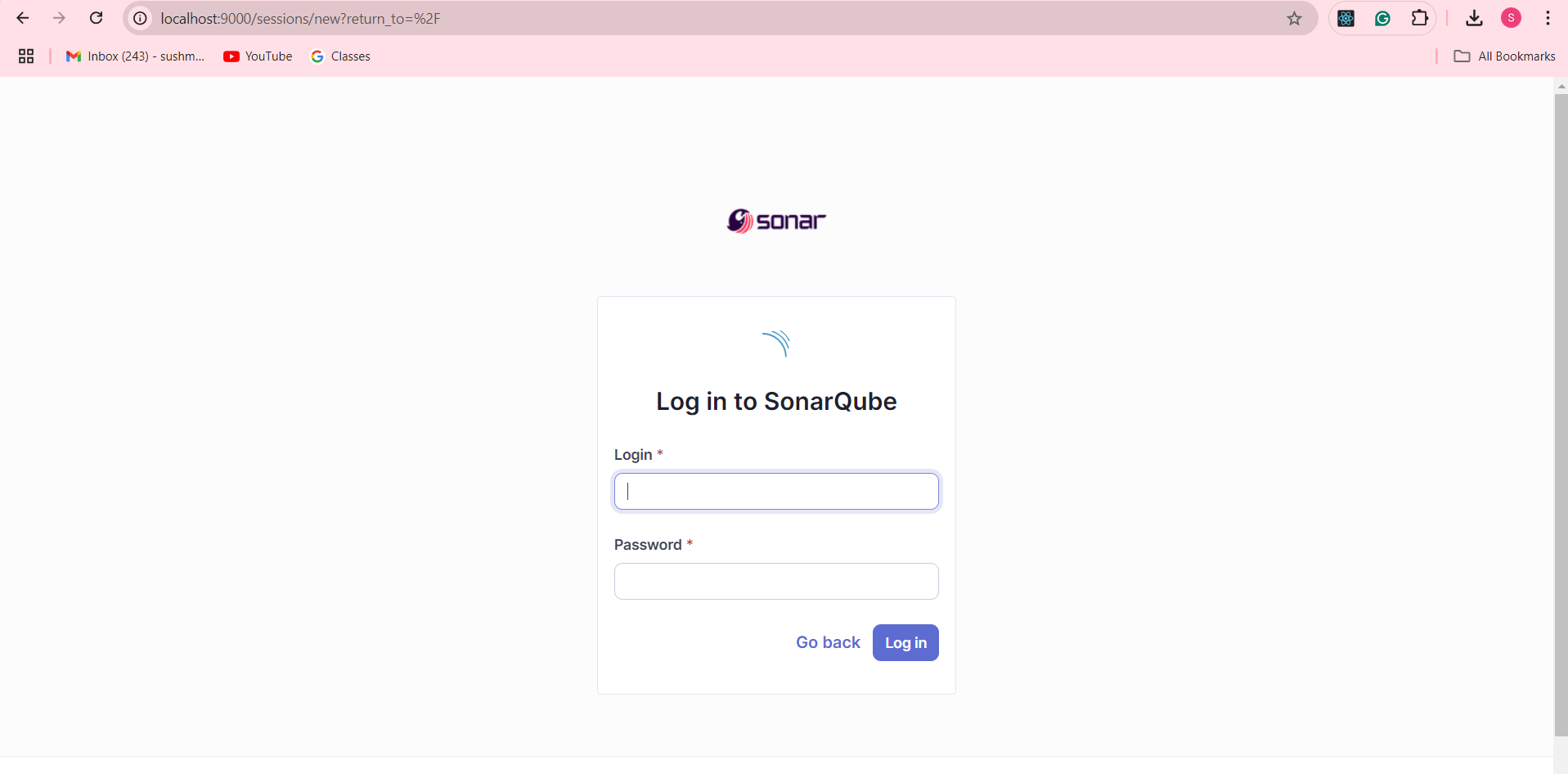
Run the following command :

docker run -d --name sonarqube -e SONAR\_ES\_BOOTSTRAP\_CHECKS\_DISABLE=true -p 9000:9000 sonarqube:latest

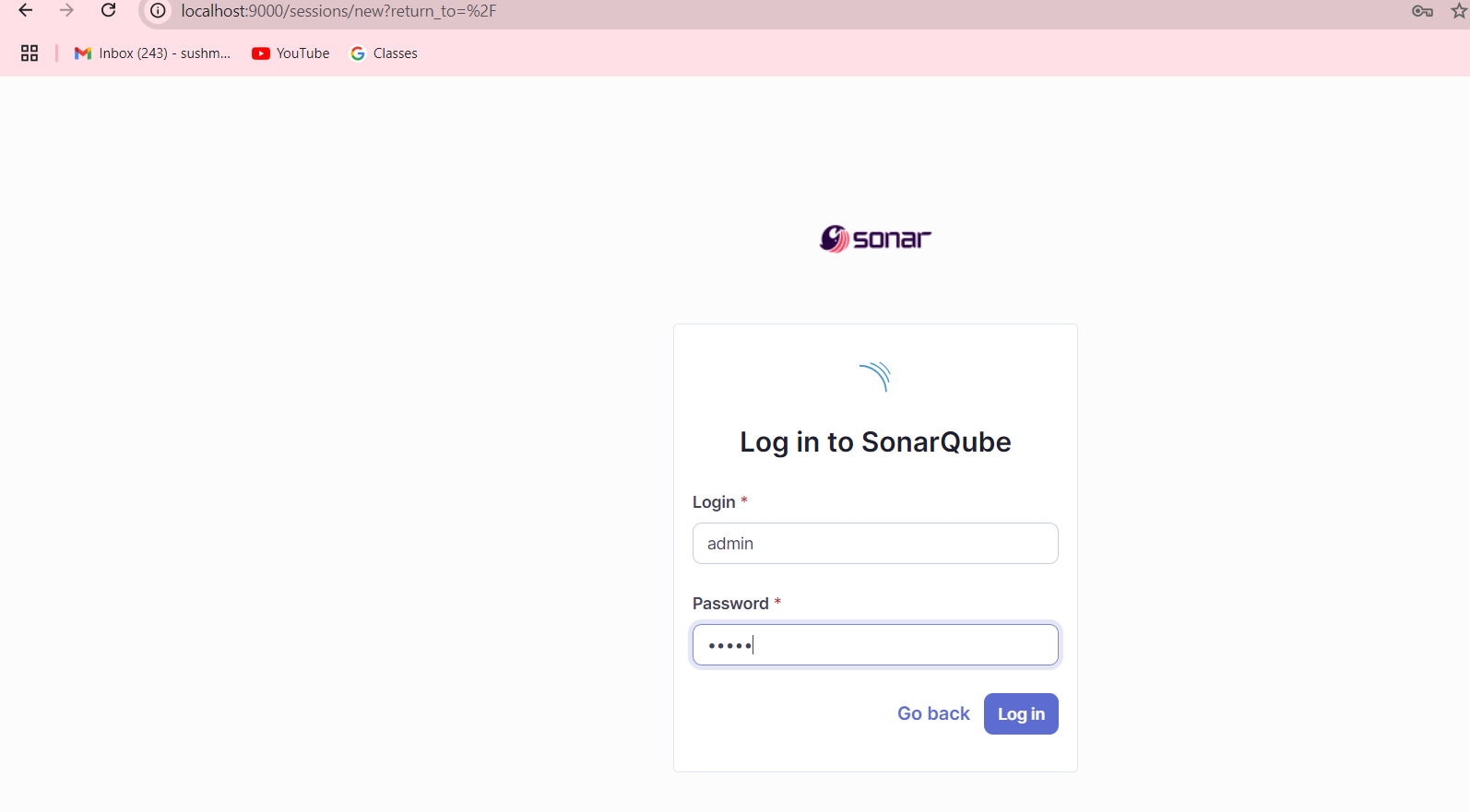




then open localhost 9000



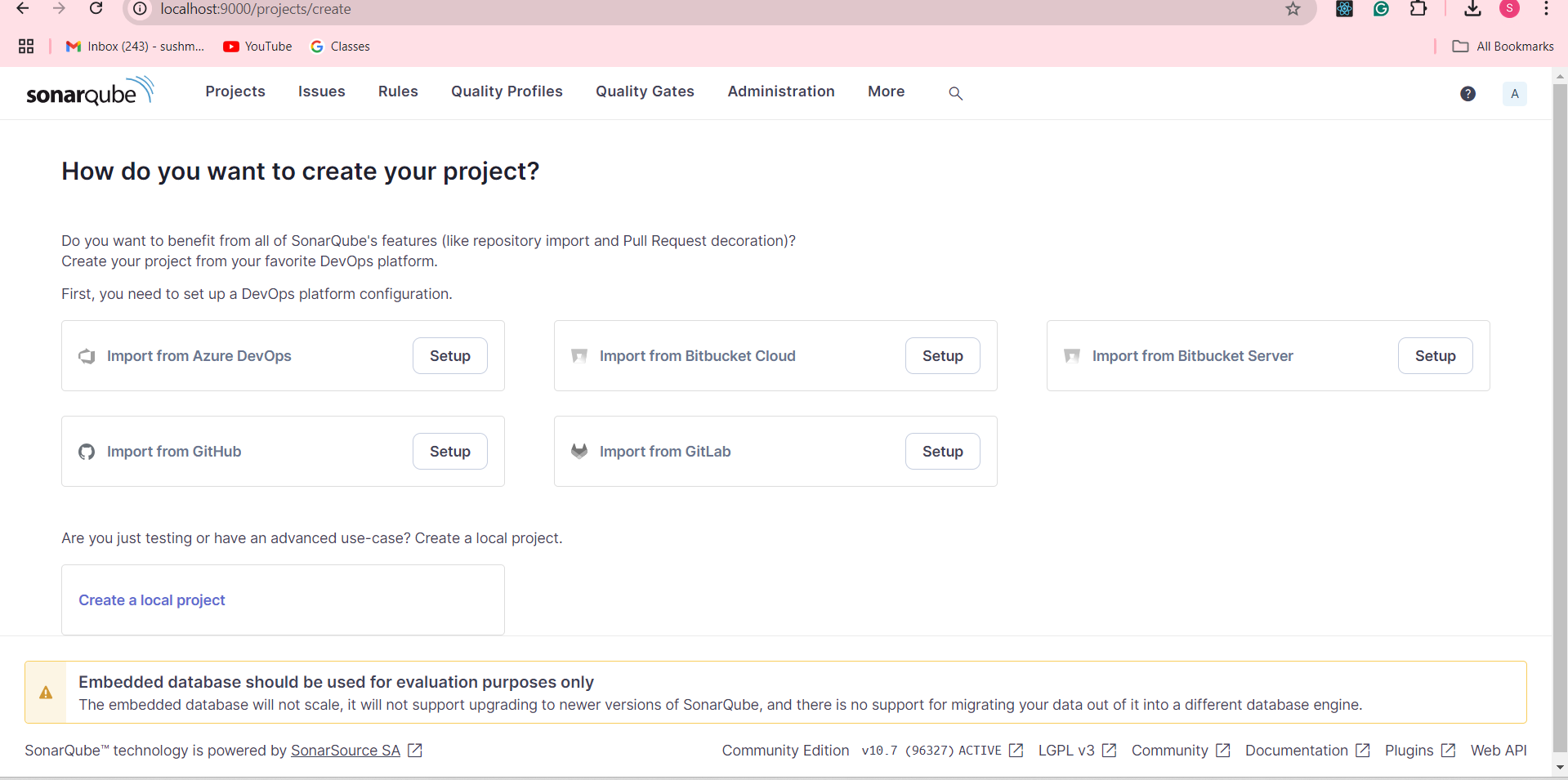
default admin password is admin admin



then it will prompt you to change the password :

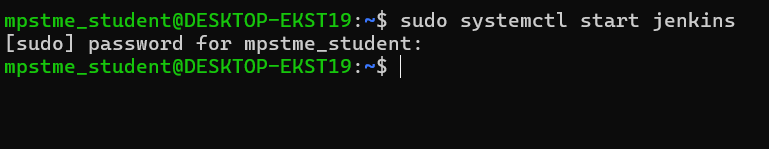
Password@12345

Created:

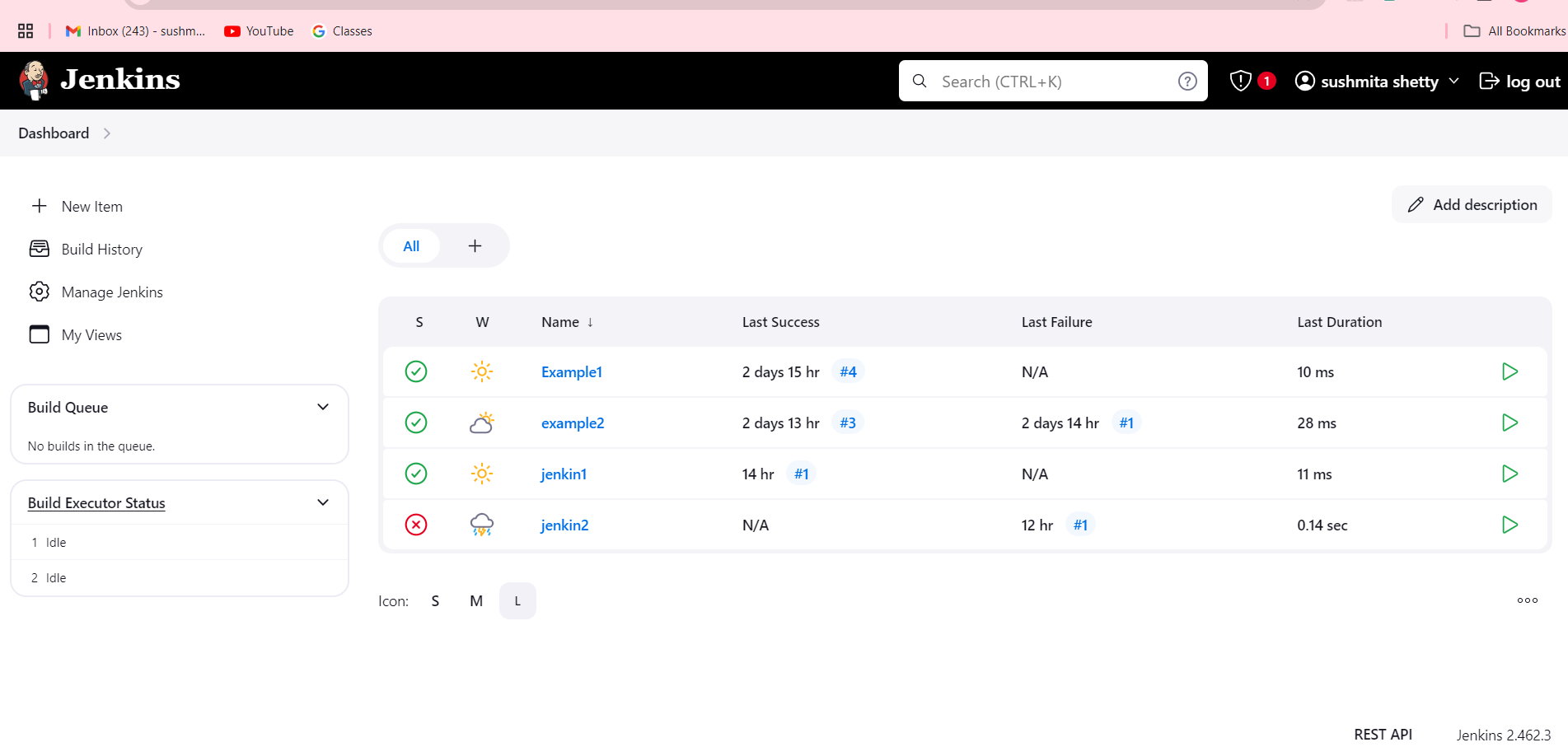


Now steps to integrate sonarcloud with Jenkins

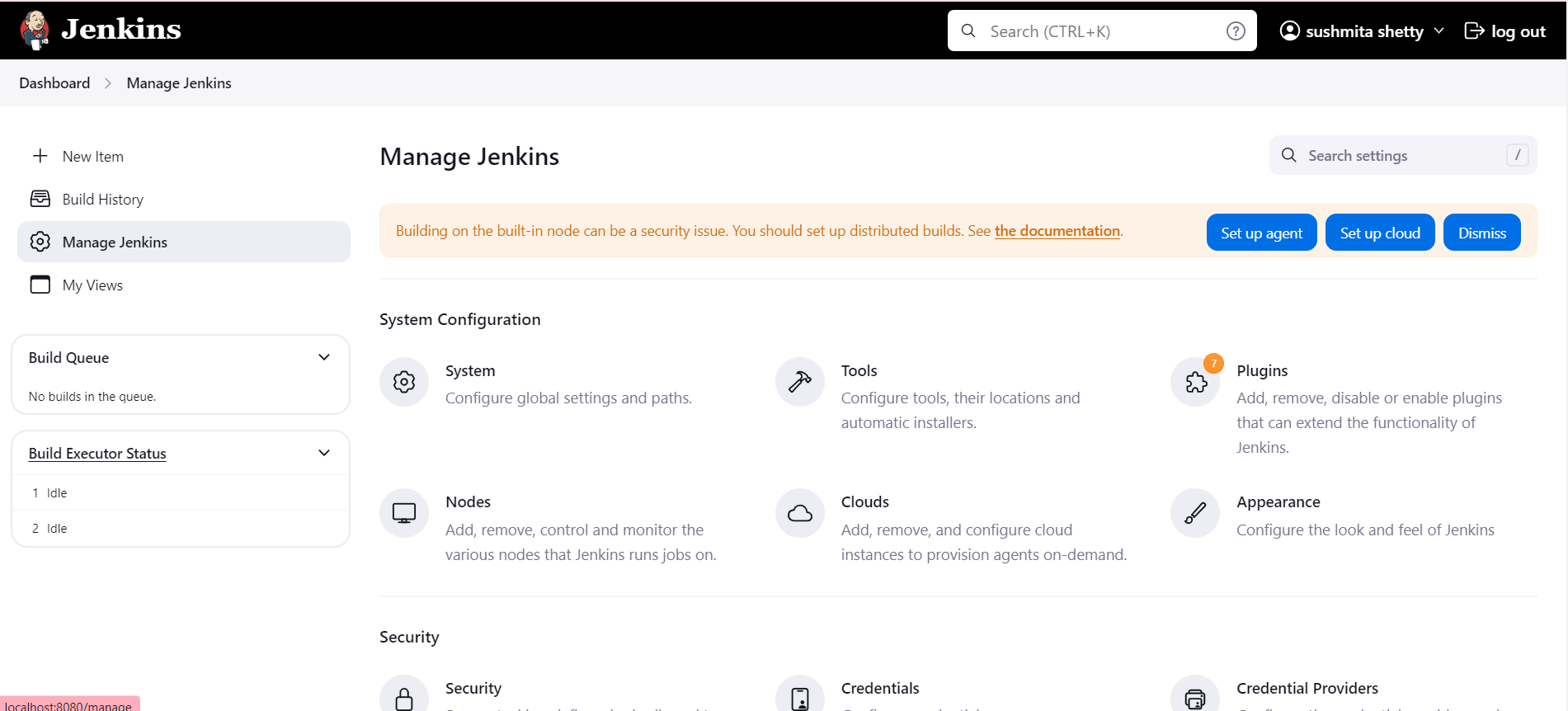
Start Jenkins on wsl : sudo systemctl start jenkins



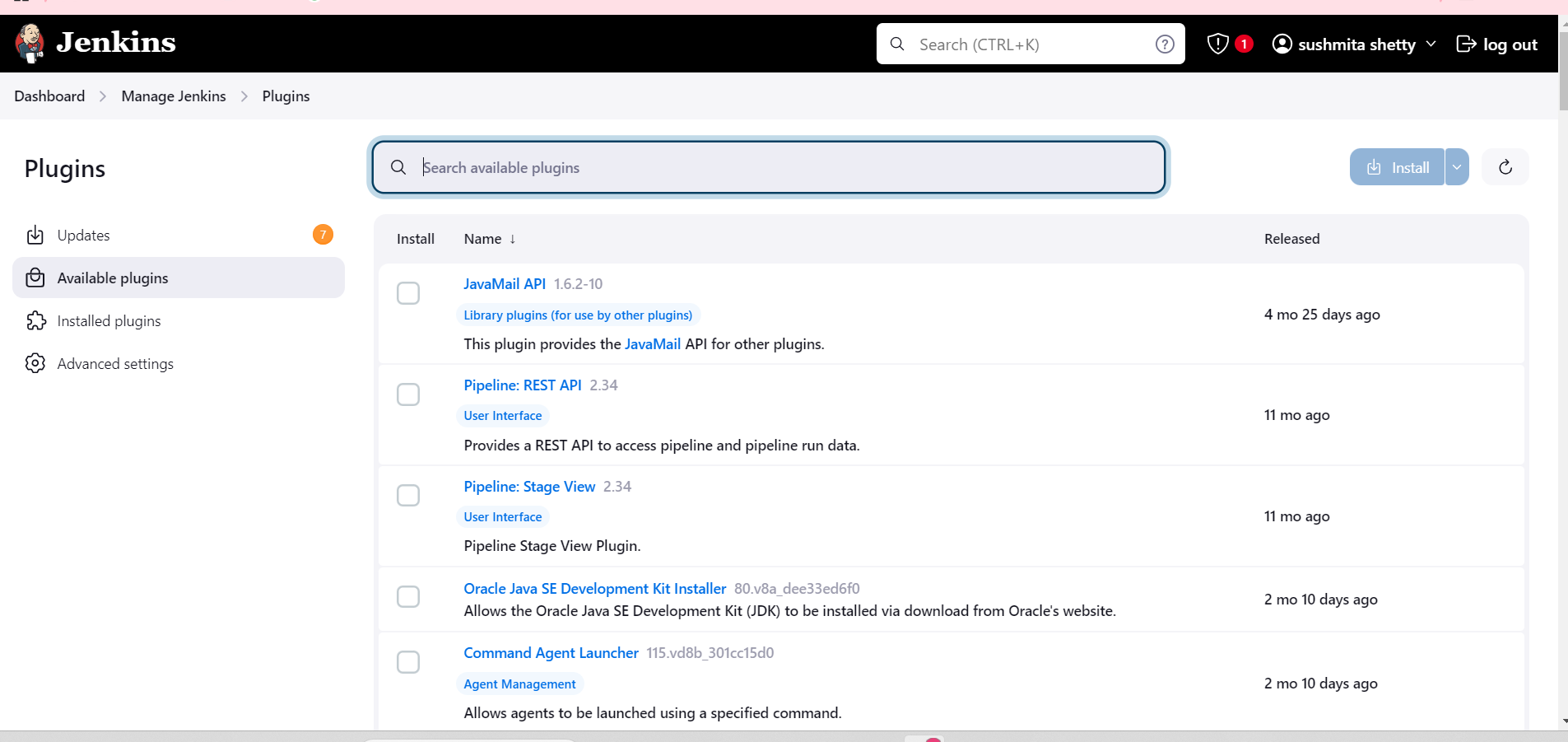
Go to local host 8080

Sign in to your Jenkins  


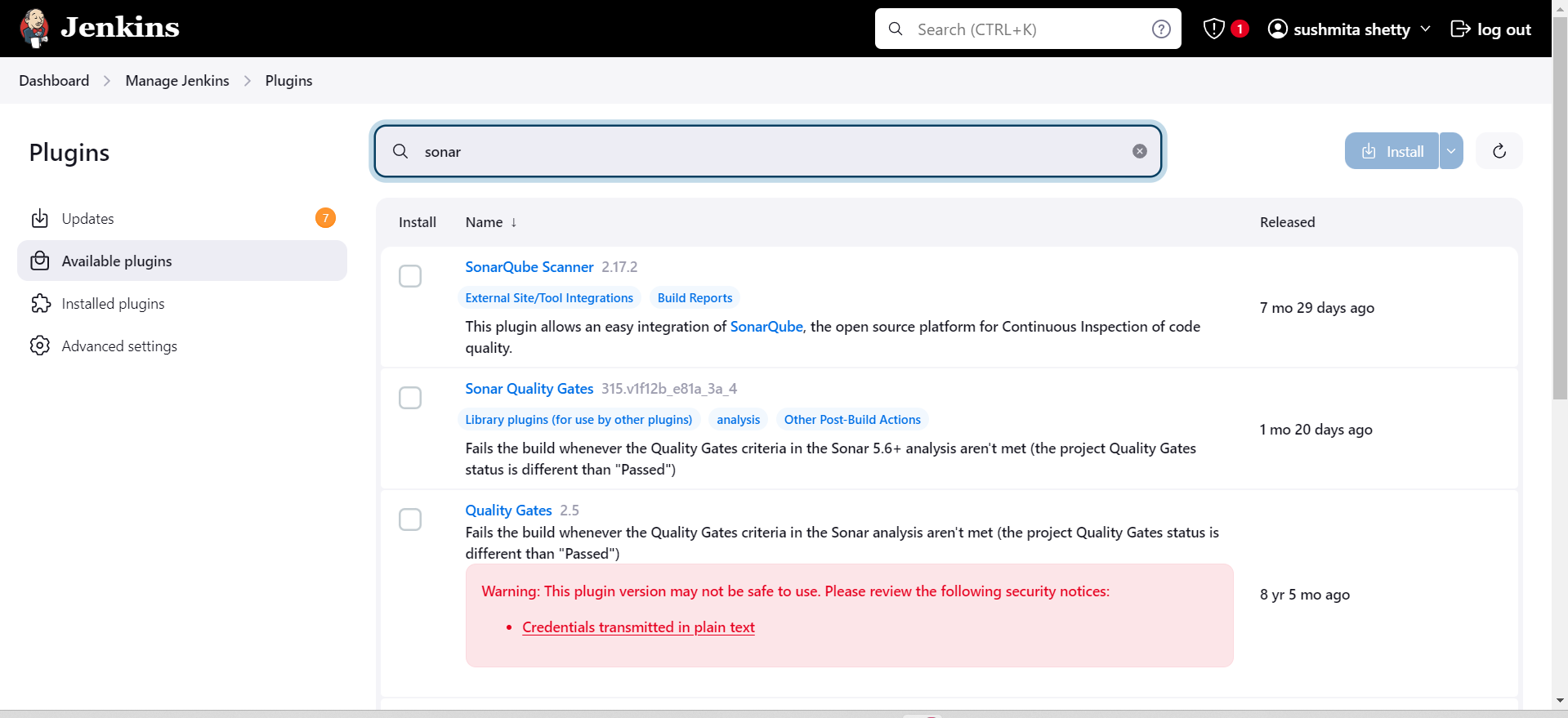
Go to manage Jenkins

 🡪 then plugin

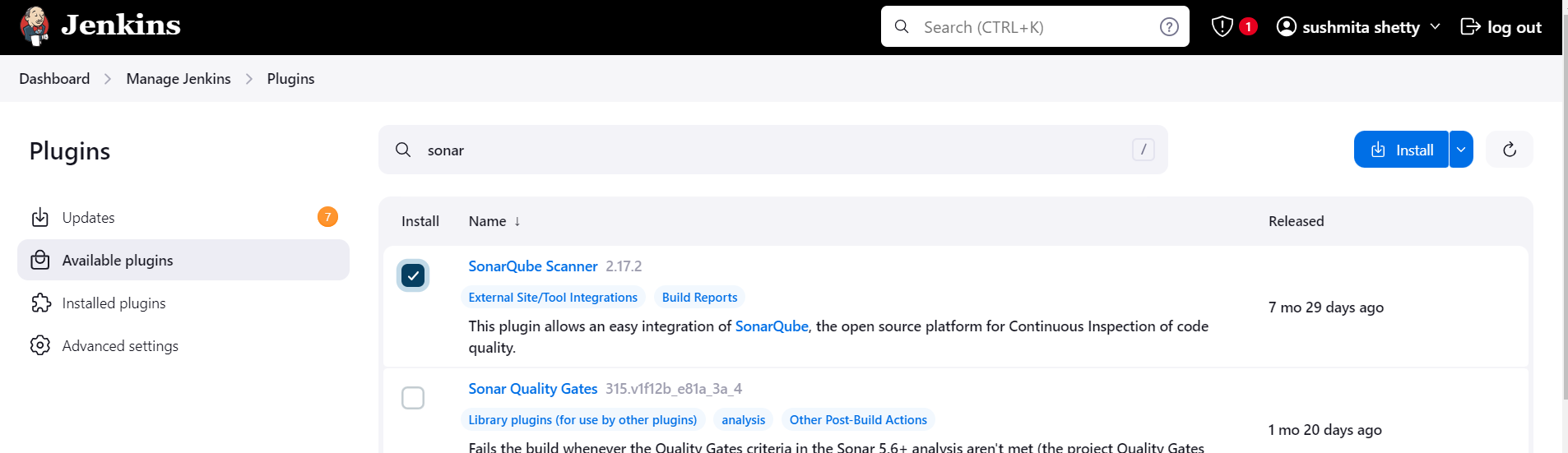
🡪then available plugin :

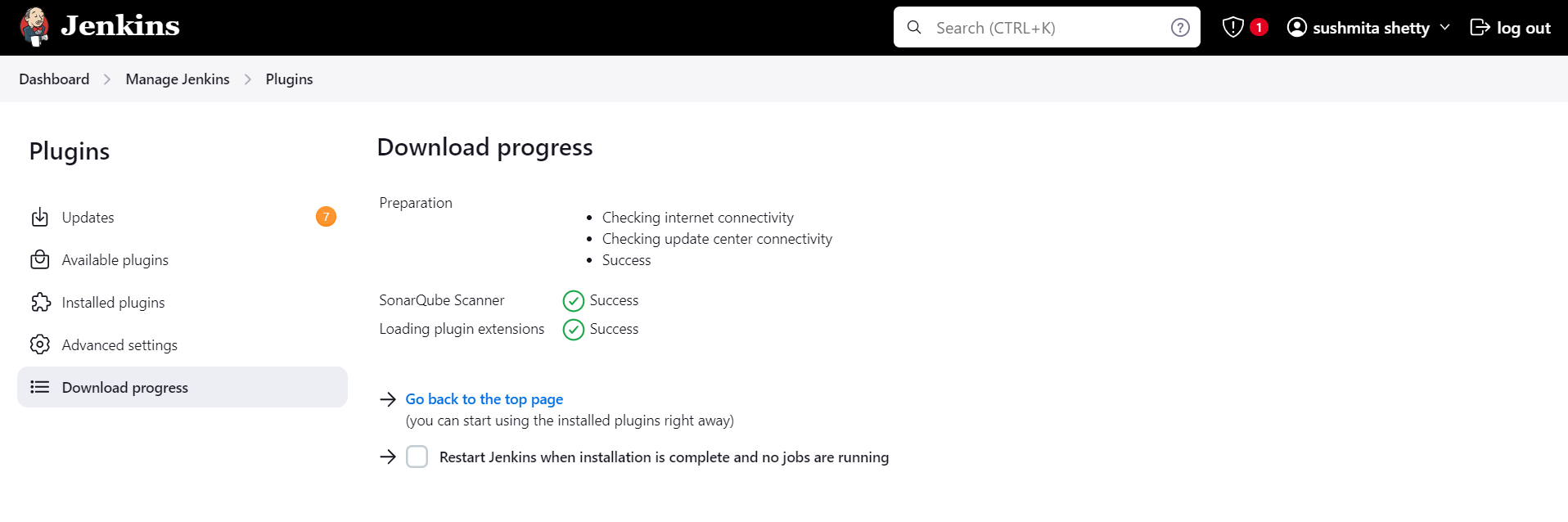


Search sonar click on sonarcube scanner:

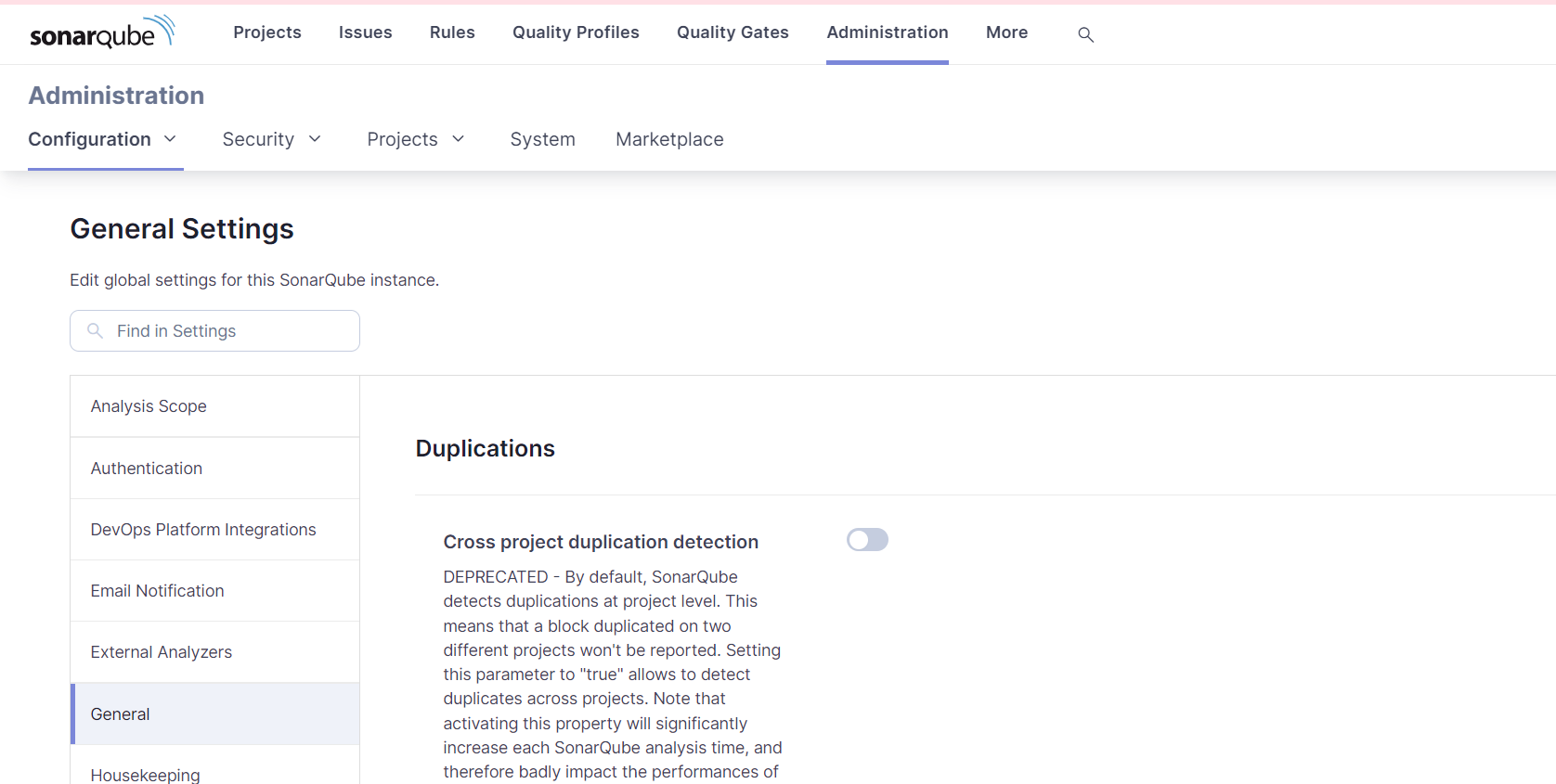


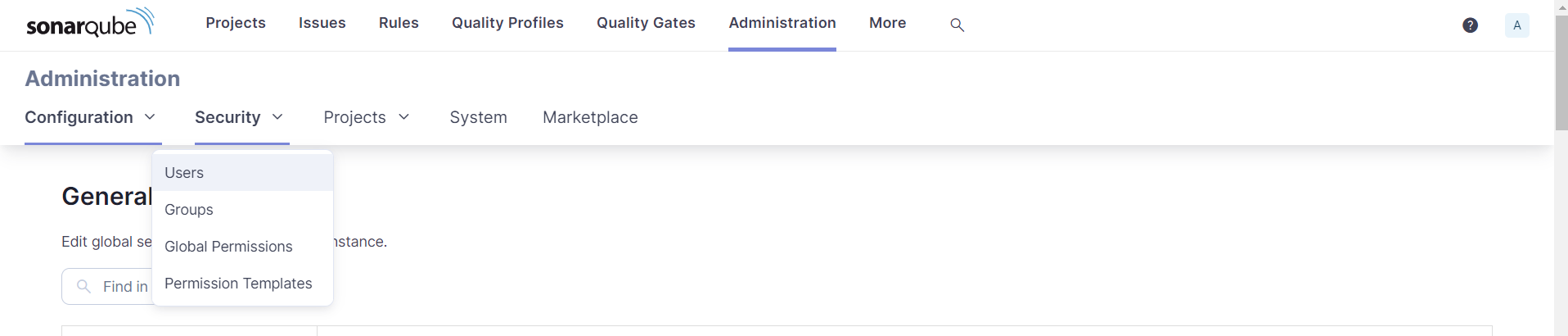
Now install the sonarcube scanner:



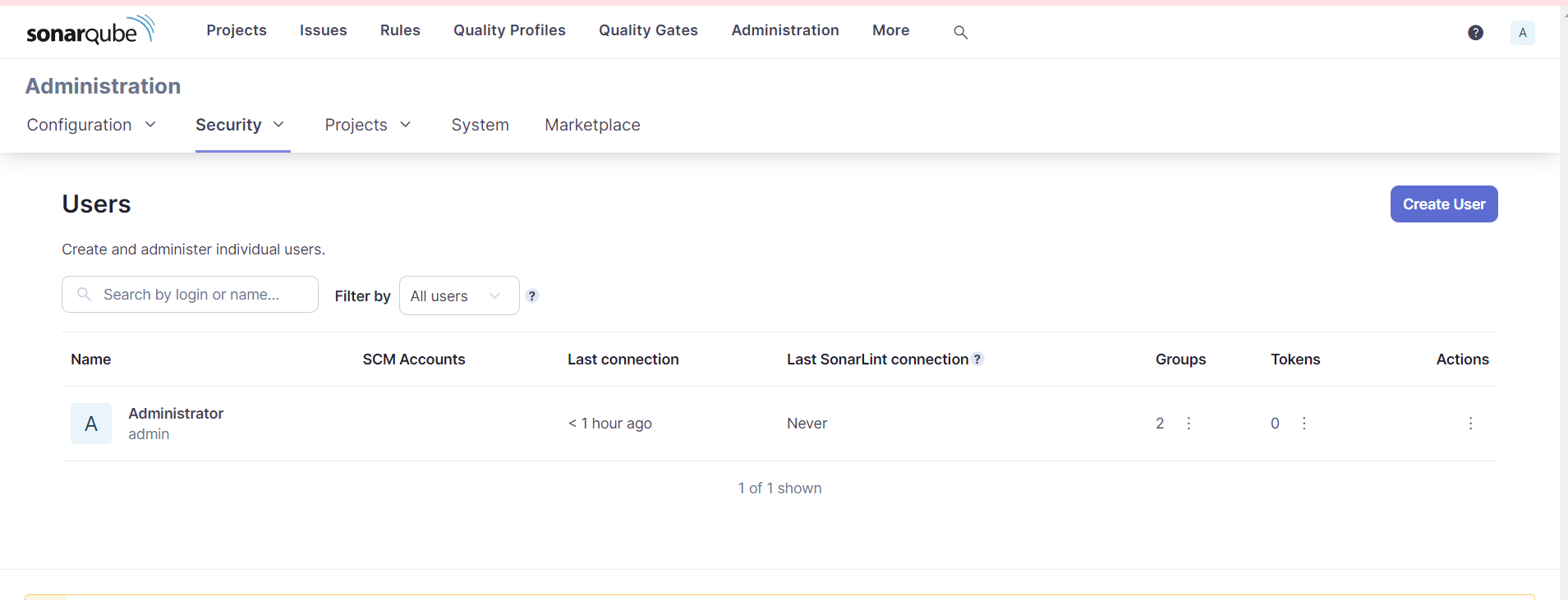


Now go to sonar 🡪click on adminstartion🡪

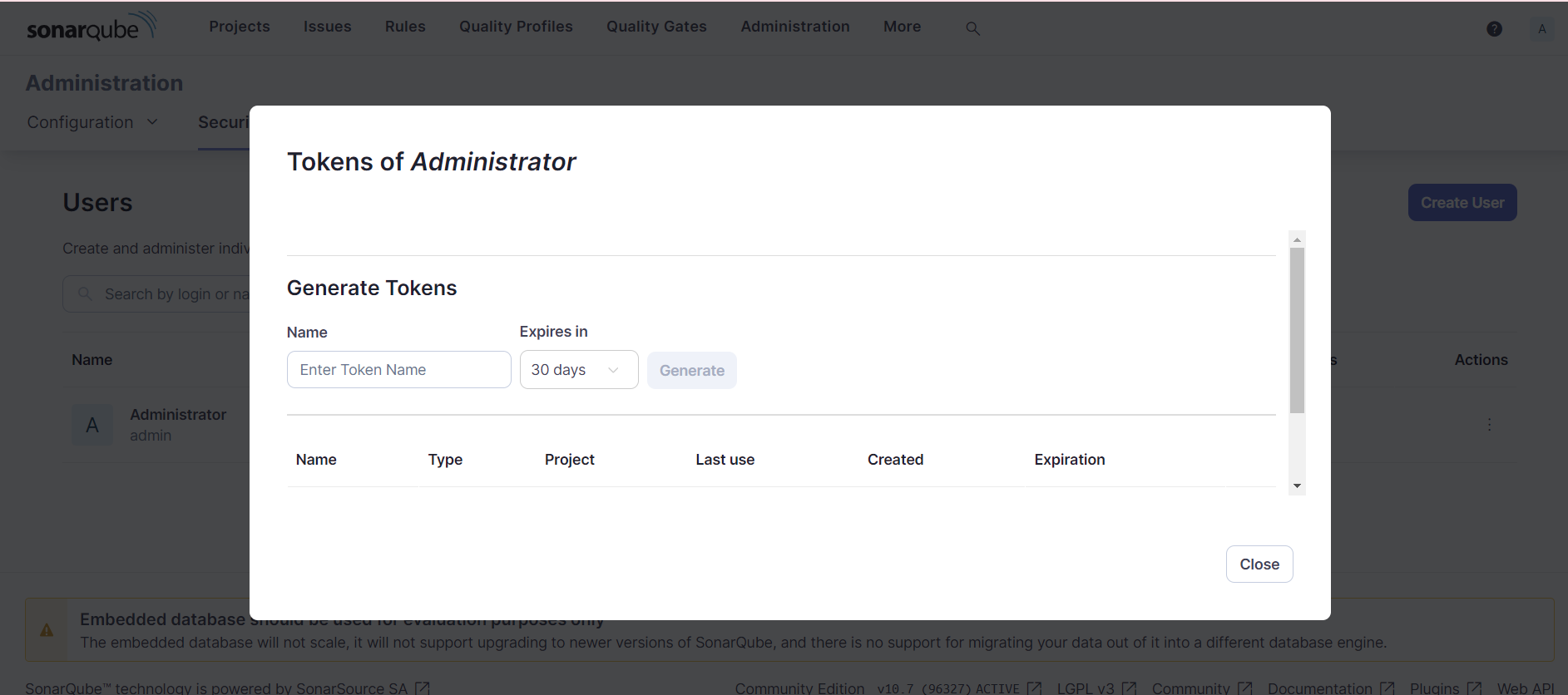
go to security🡪

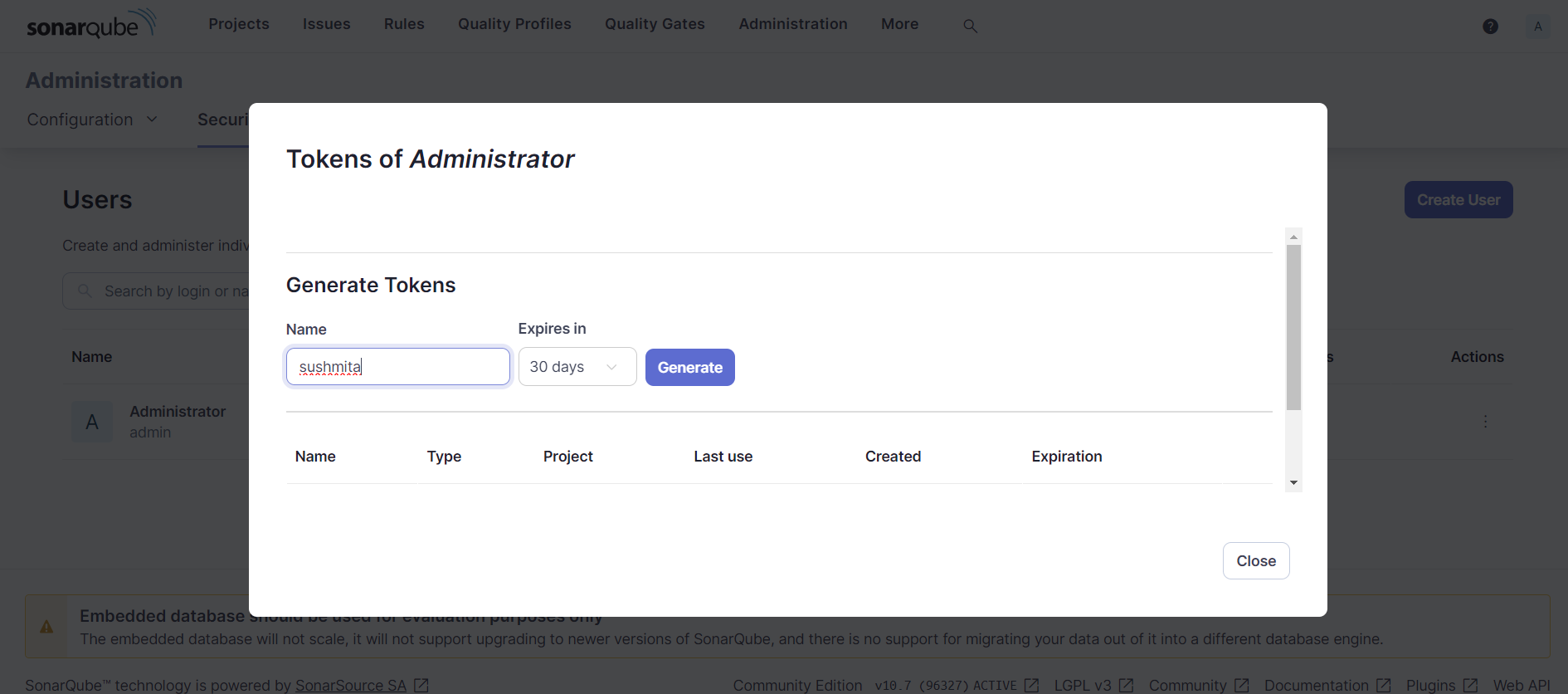


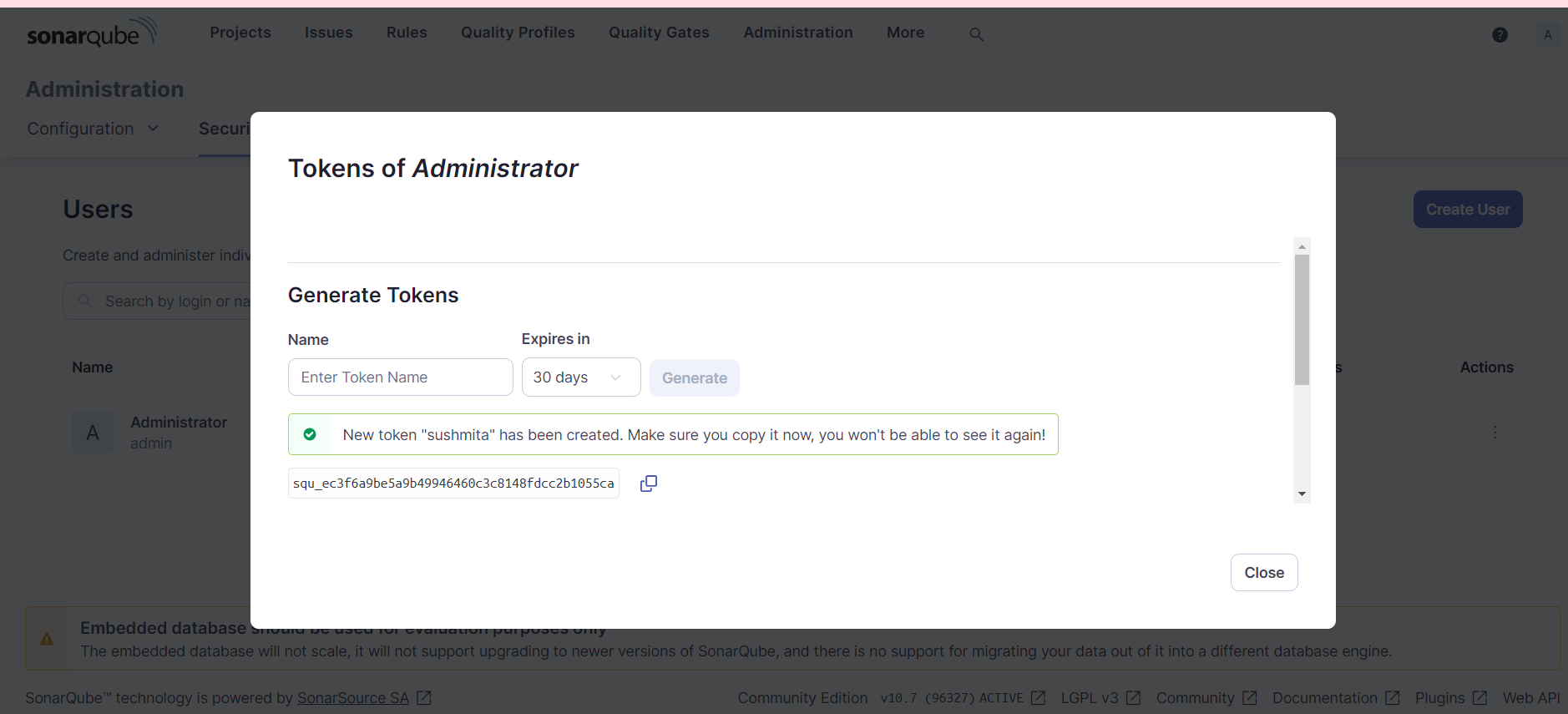
user🡪



Now generate token with some name



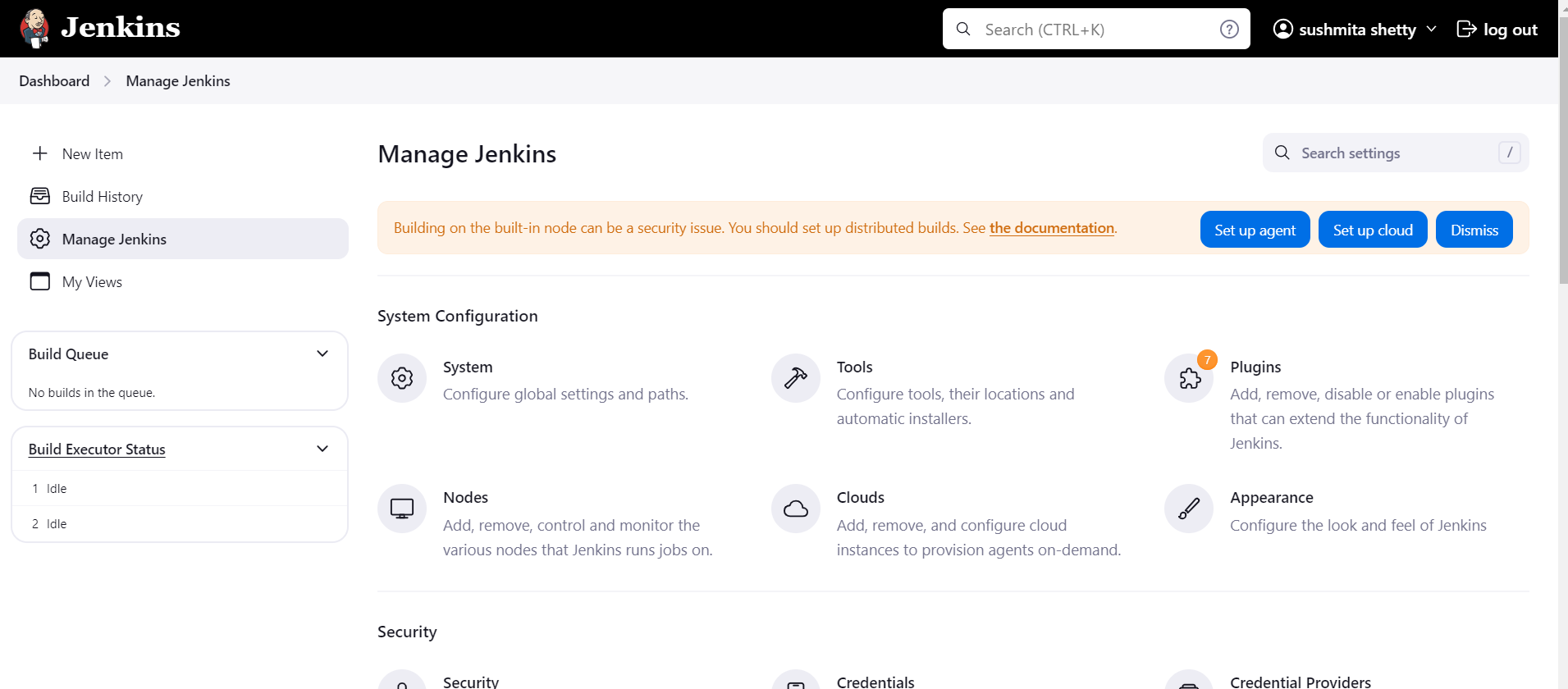




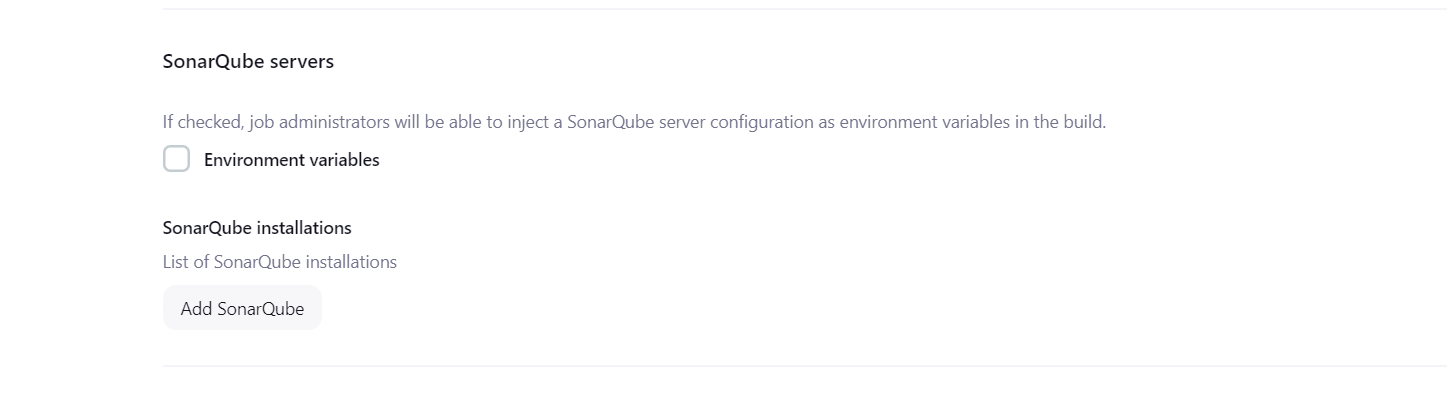
Token:

squ\_ec3f6a9be5a9b49946460c3c8148fdcc2b1055ca

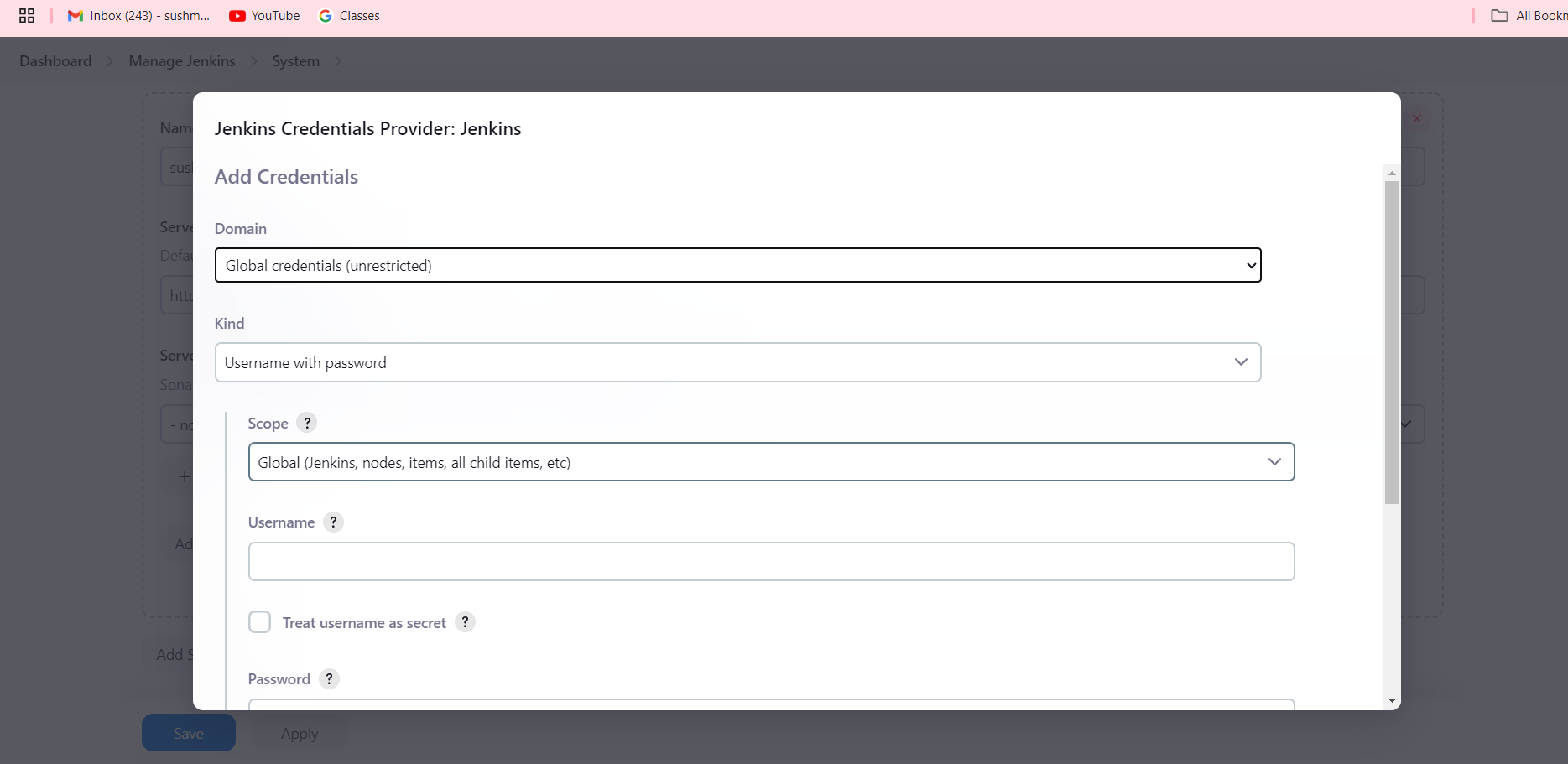
now go to manage jenkin:



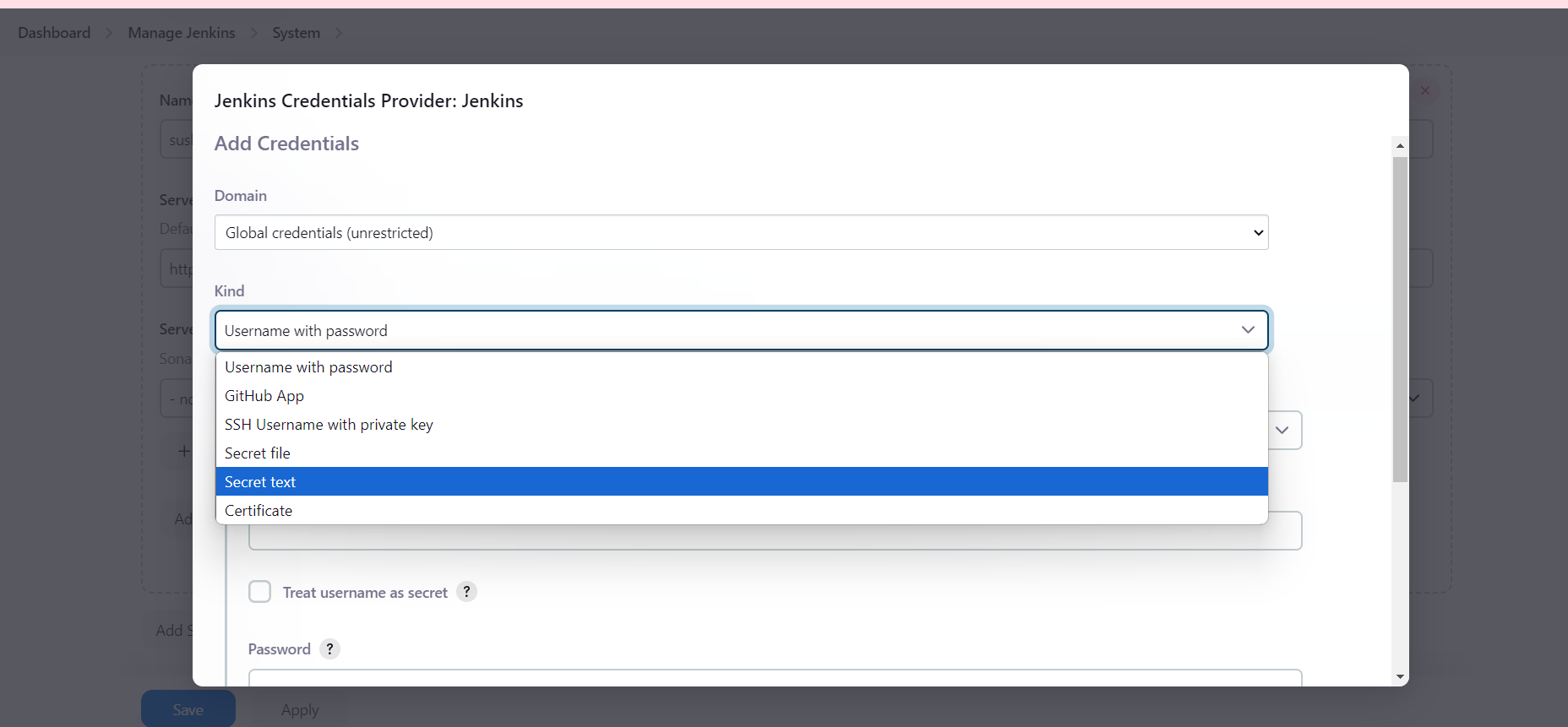
Manage jenkin🡪system🡪Go to sonarcube server section:



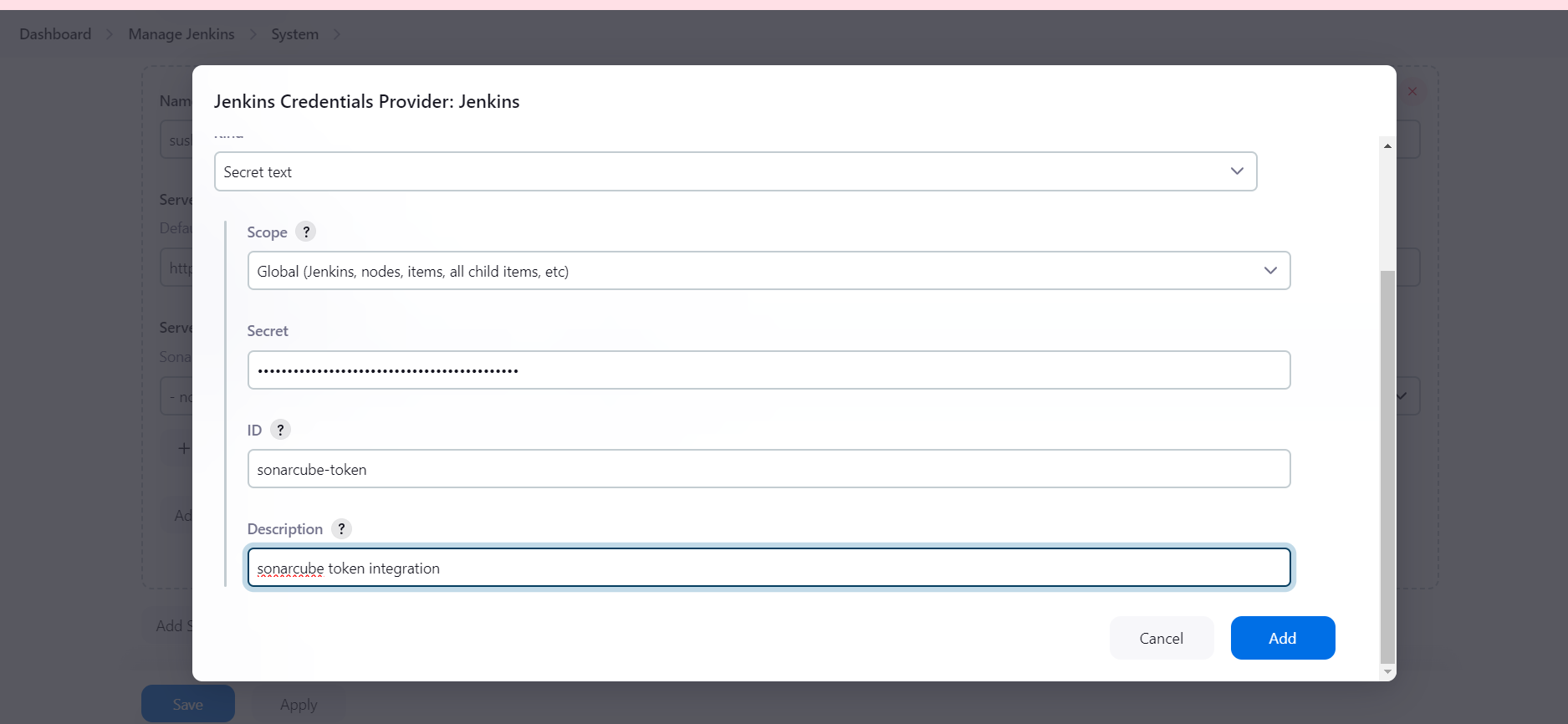
Click on add 🡪then jenkin credential 🡪



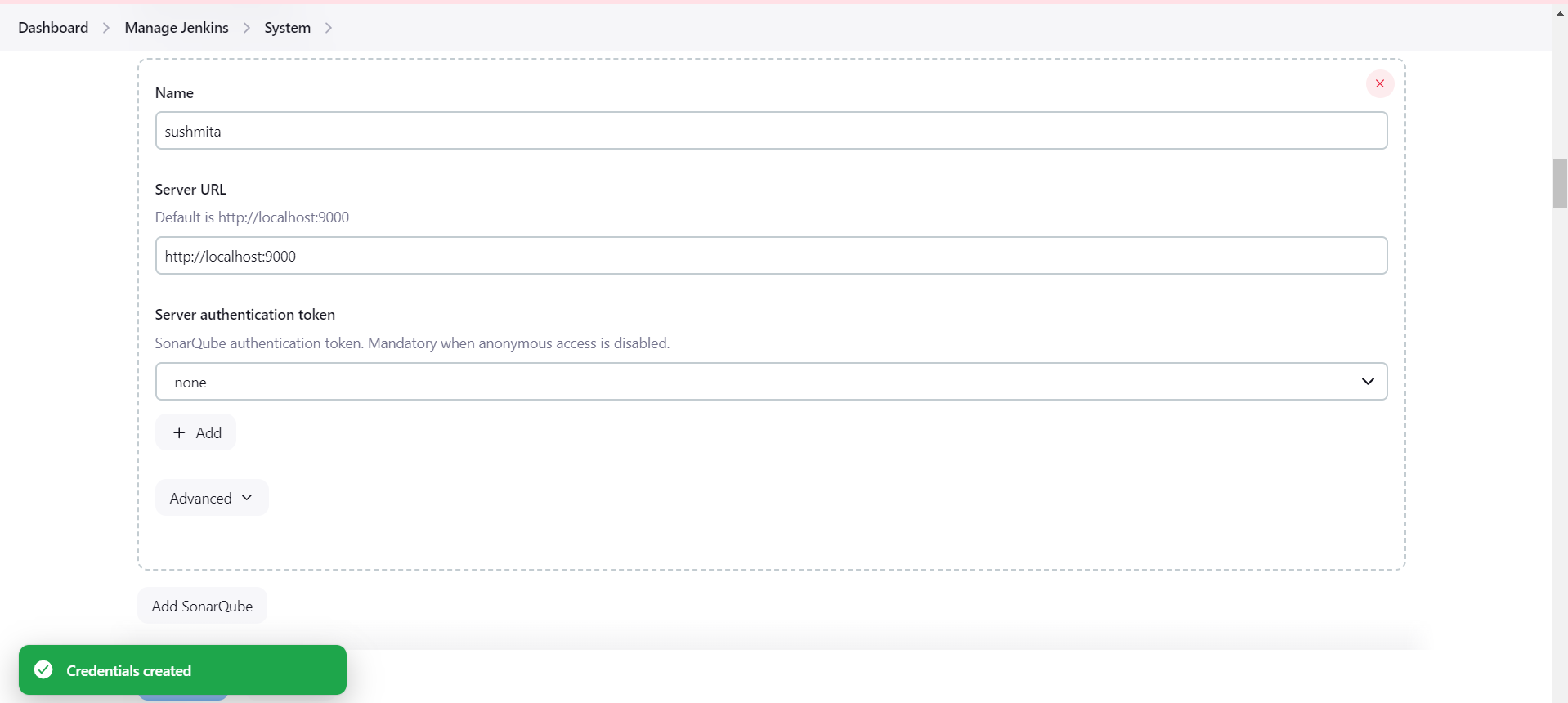
Select secret text 🡪addjenkin🡪token



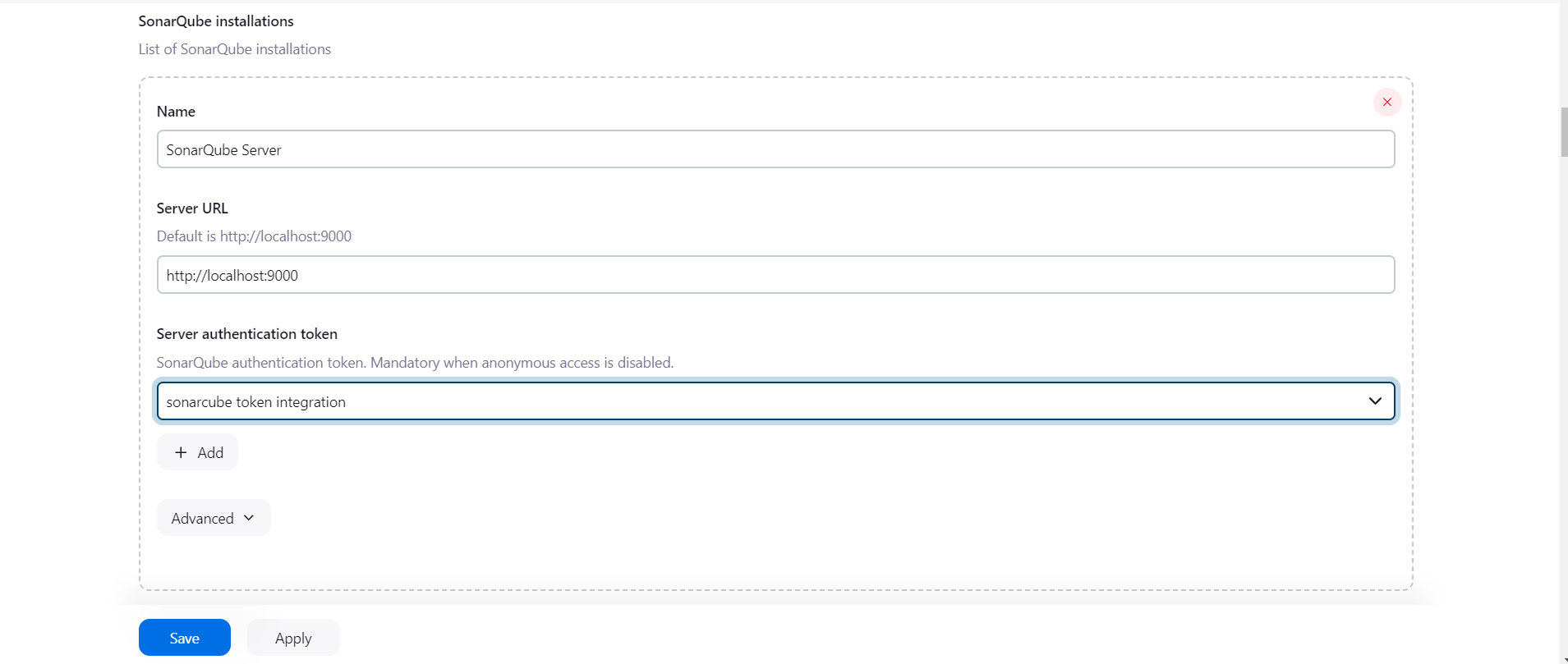
Add token , description , id :



You will get a credential created:

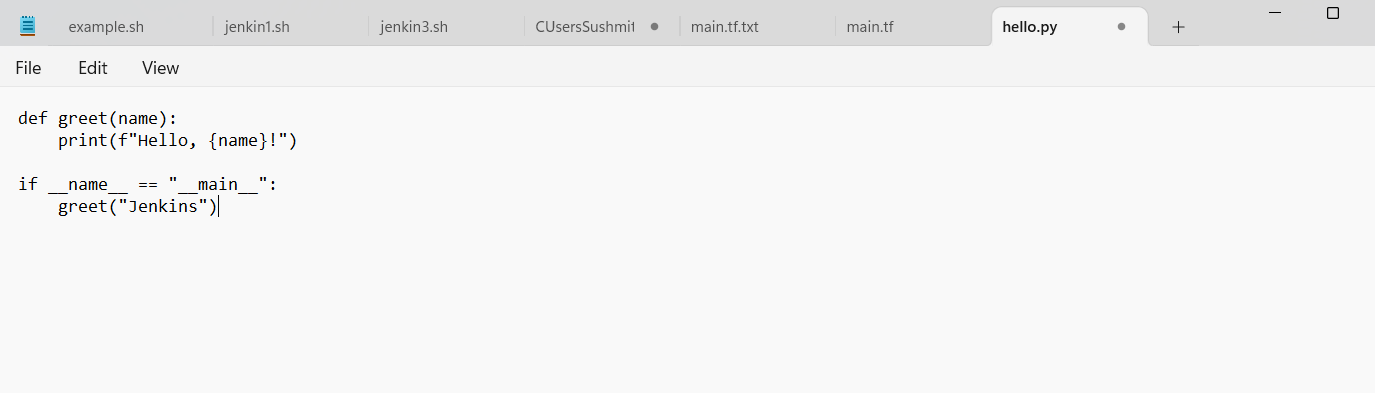


Now name it and select sonarcloud token integration



Now create a python hello world project :

Hello.py



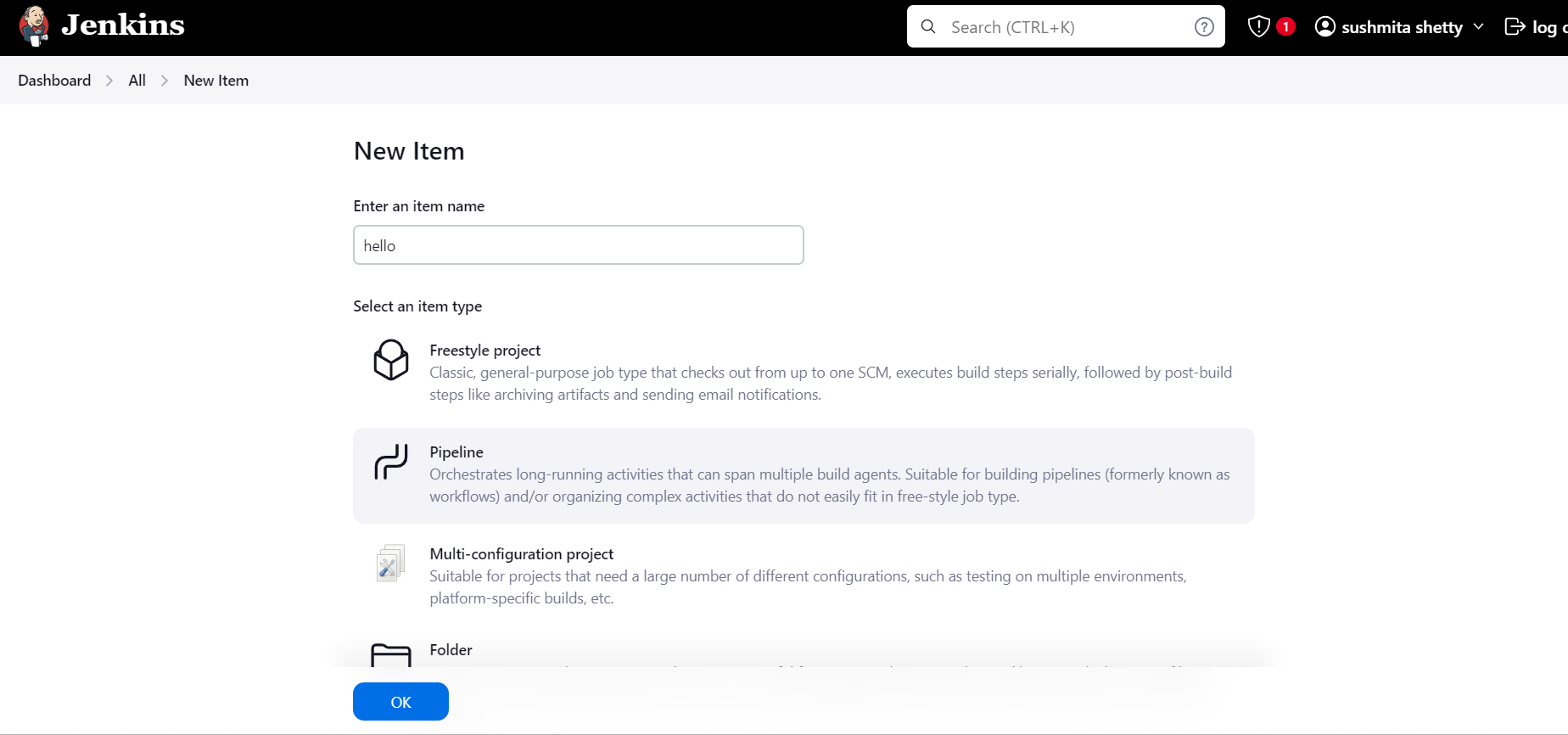
def greet(name):

print(f"Hello, {name}!")

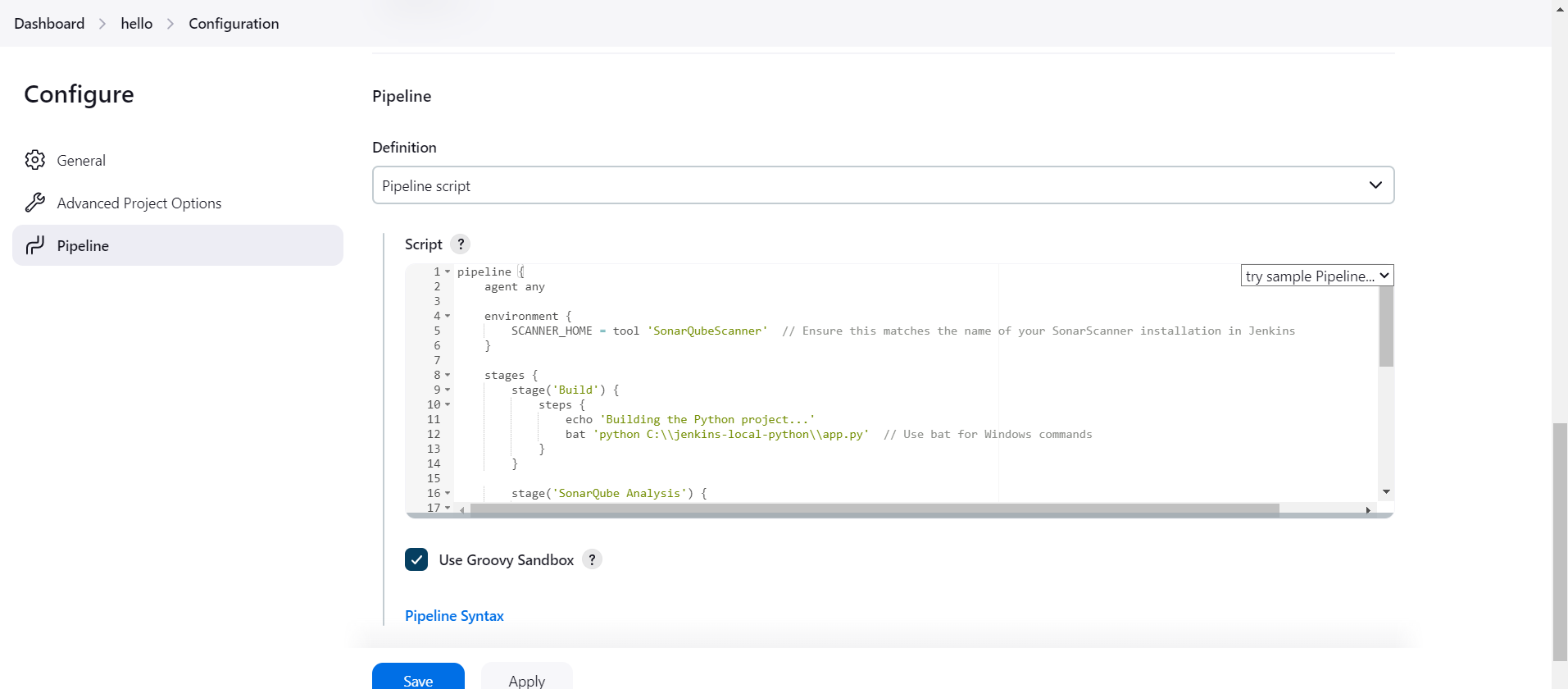
if \_\_name\_\_ == "\_\_main\_\_":

greet("Jenkins")

go to jenkin dashboard🡪new item



Go to pipeline : script



Paste the code:

pipeline {

agent any

environment {

SCANNER\_HOME = tool 'SonarQube Scanner' // Ensure this matches the name of your SonarScanner installation in Jenkins

}

stages {

stage('Build') {

steps {

echo 'Building the Python project...'

bat 'python C:\\jenkins-local-python\\app.py' // Use bat for Windows commands

}

}

stage('SonarQube Analysis') {

steps {

withSonarQubeEnv('SonarQube Server') { // 'SonarQube' is the name you configured for your server

bat "${SCANNER\_HOME}\\bin\\sonar-scanner.bat -Dsonar.projectKey=my-local-python-project -Dsonar.sources=C:\\jenkins-local-python"

}

}

}

stage('Quality Gate') {

steps {

timeout(time: 1, unit: 'MINUTES') {

waitForQualityGate abortPipeline: true

}

}

}

}

post {

always {

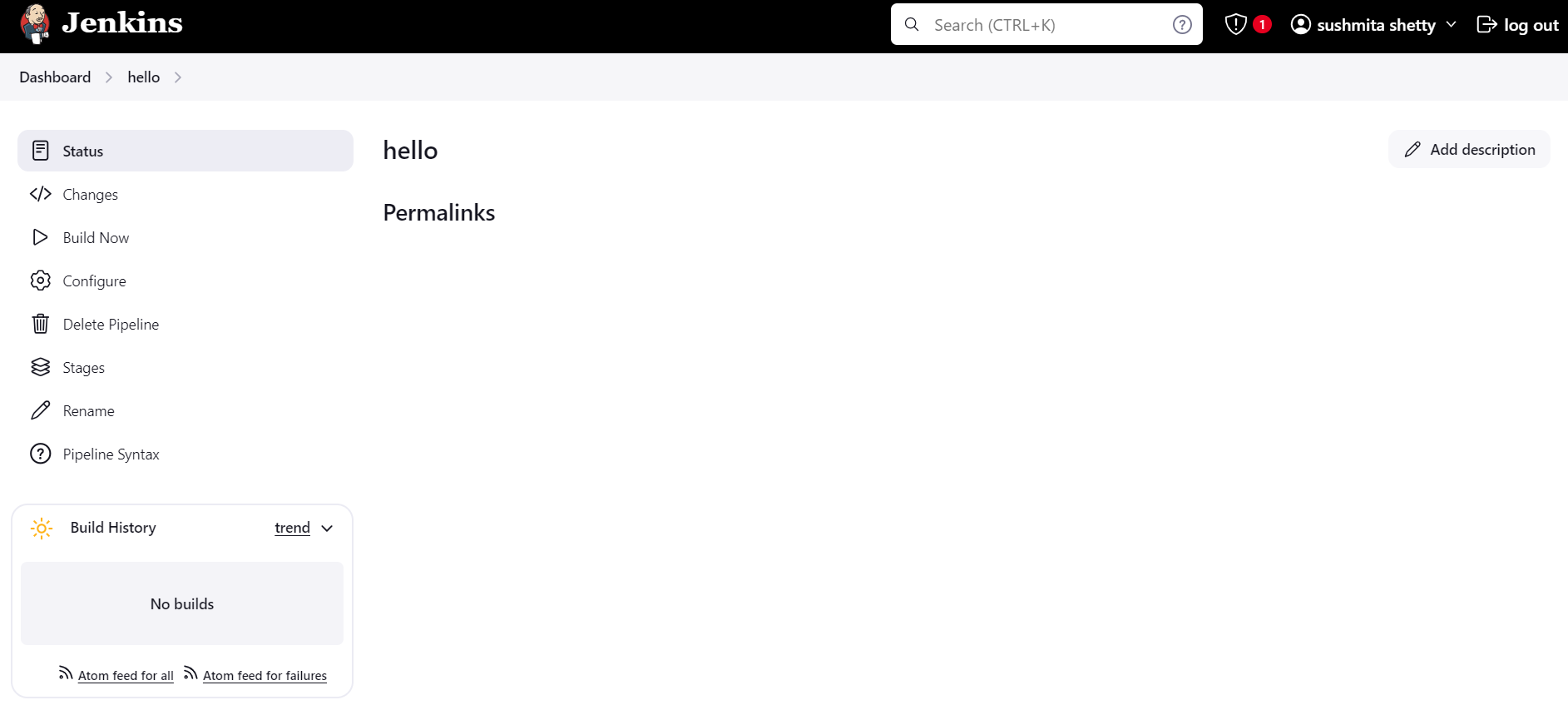
echo 'Pipeline completed'

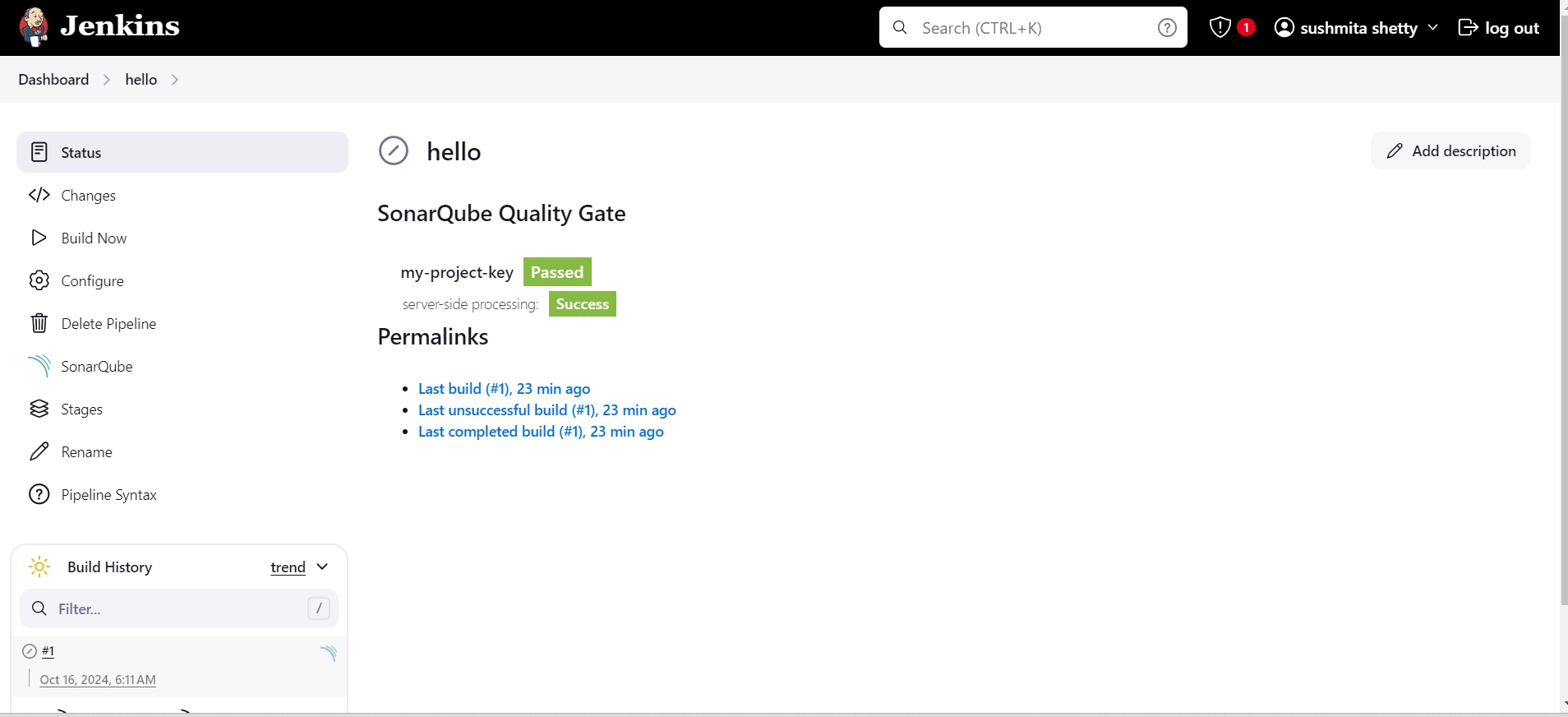
}

}

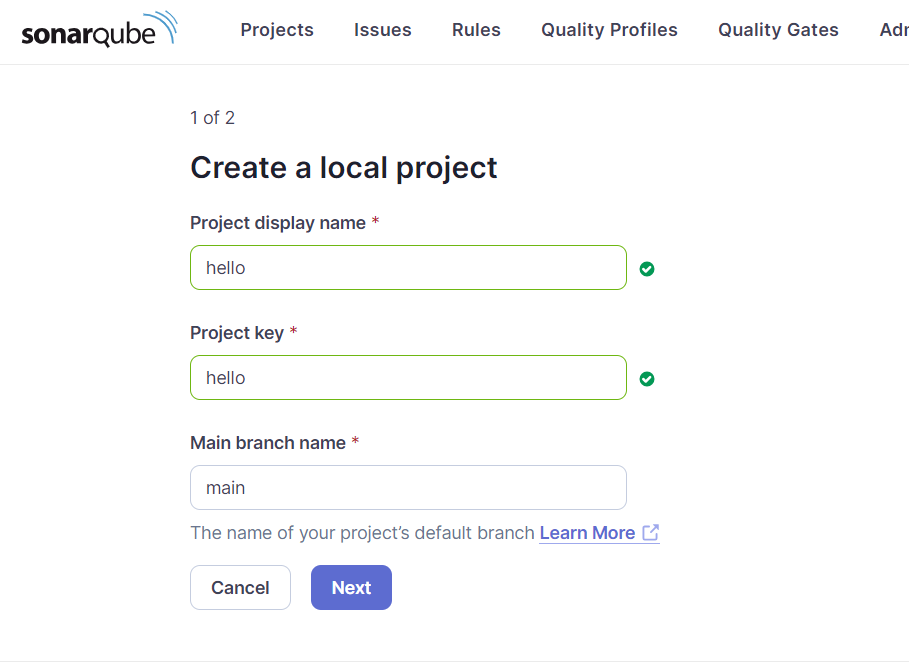
}

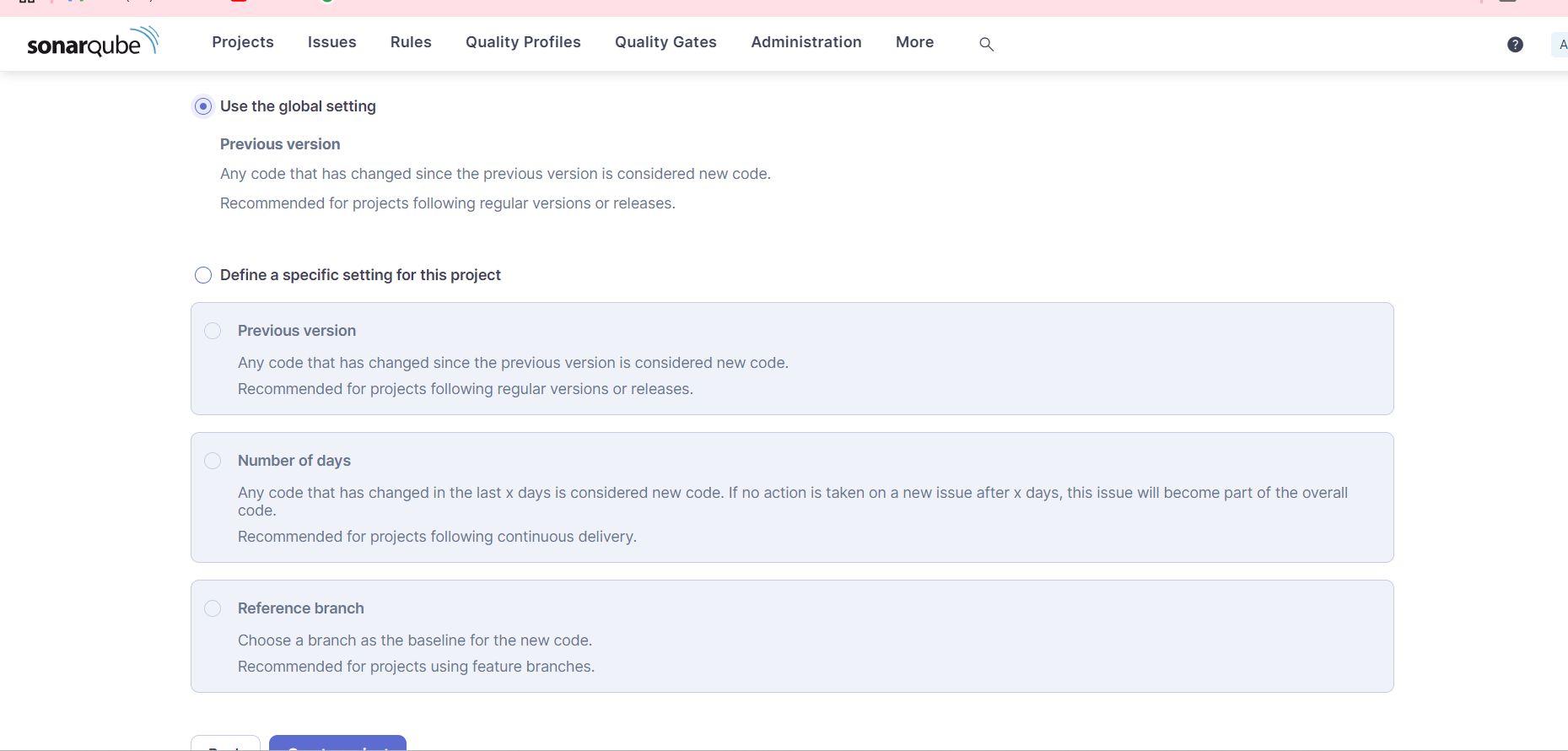
Apply🡪save 🡪build now

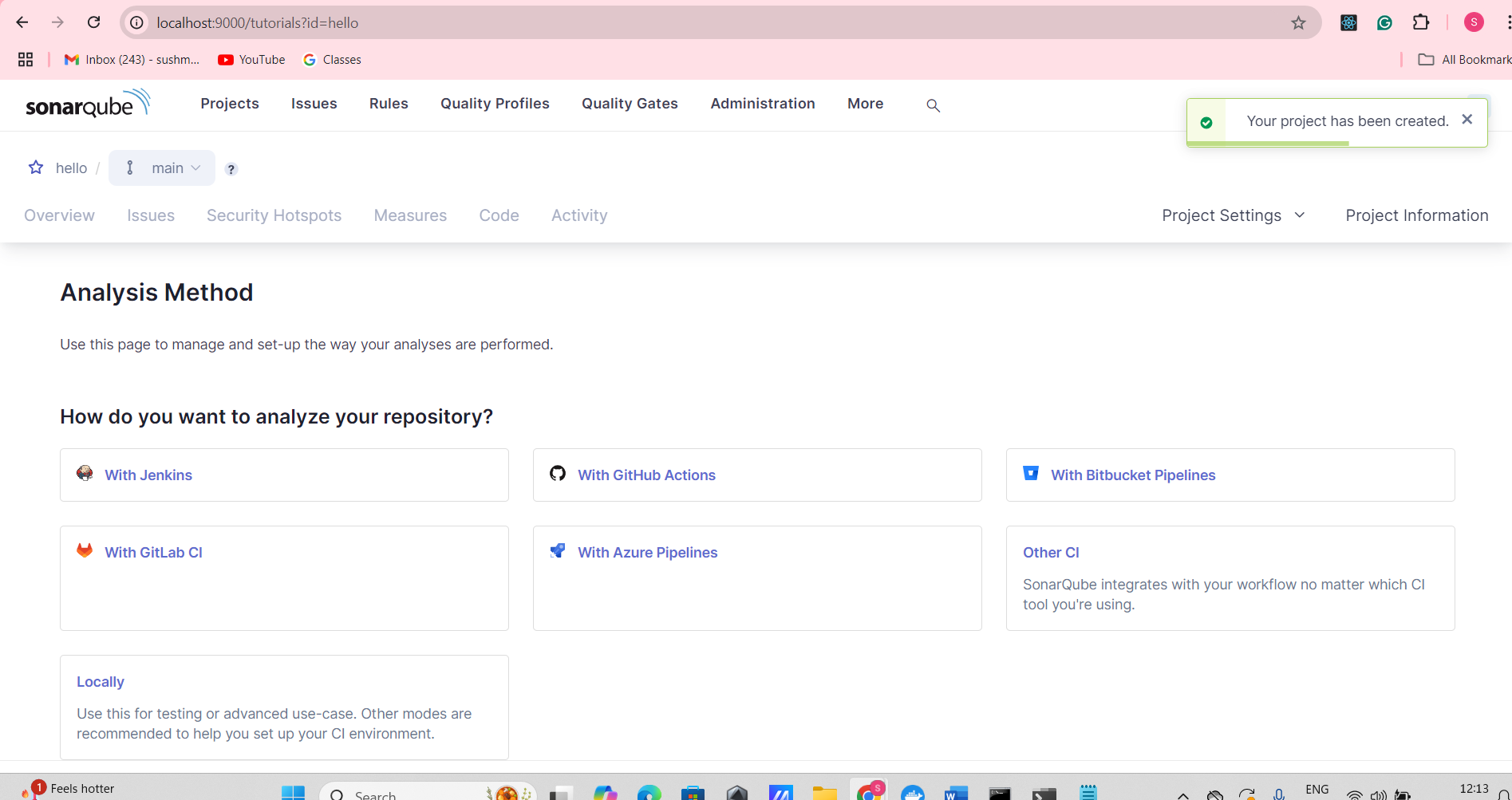


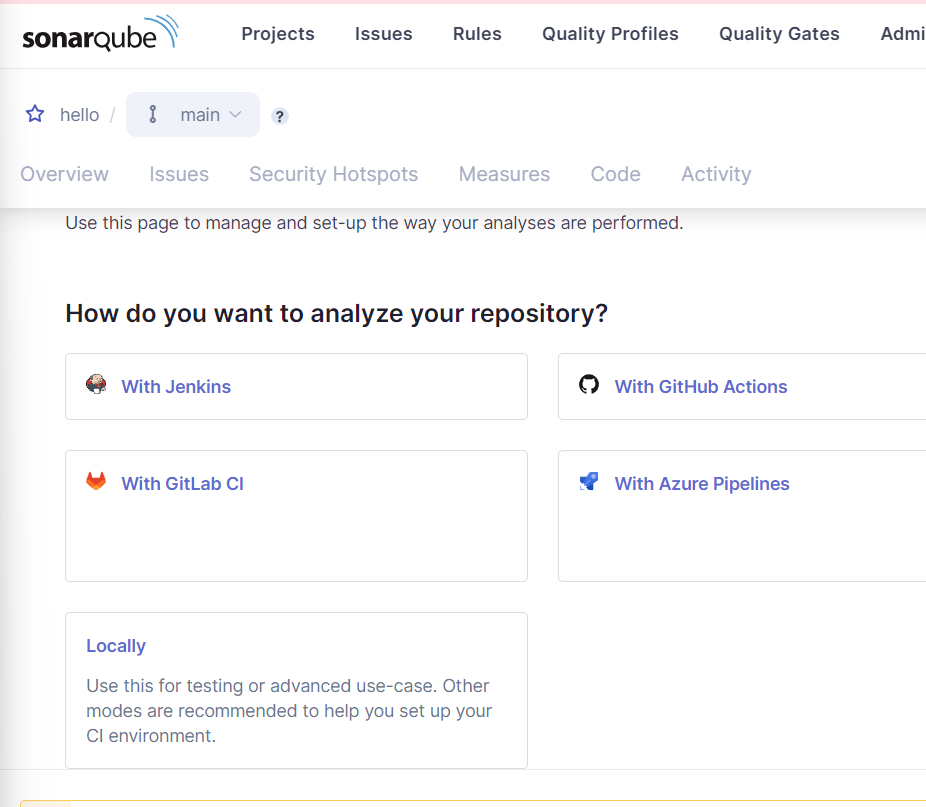


Create project 🡪local project in sonar









Created a hell.py:

print("hello world")

then create a same name project in solarqube

then create a new item in jenkin with same name and add the following script:

pipeline {

agent any // Use any available agent

stages {

stage('Create Python File') {

steps {

script {

// Write the Python code to hellllo.py

writeFile file: 'hellllo.py', text: '''\

print("hello world")

'''

}

}

}

stage('Run Python Script') {

steps {

echo 'Running Python script...'

// Execute the Python script

sh 'python3 hellllo.py'

}

}

stage('SonarQube Analysis') {

steps {

script {

// Define the SonarQube properties

def scannerHome = tool 'SonarQube Scanner' // Use the SonarQube Scanner installed in Jenkins

withSonarQubeEnv('SonarQube Server') { // Use the configuration name for your SonarQube server

sh "${scannerHome}/bin/sonar-scanner -Dsonar.projectKey=hello -Dsonar.sources=."

}

}

}

}

stage('Quality Gate') {

steps {

script {

// Wait for the quality gate to pass

waitForQualityGate abortPipeline: true

}

}

}

}

post {

always {

// This block runs after all stages, regardless of success or failure

echo 'Pipeline completed'

}

failure {

// Additional actions on failure

echo 'Pipeline failed, check the logs for more information.'

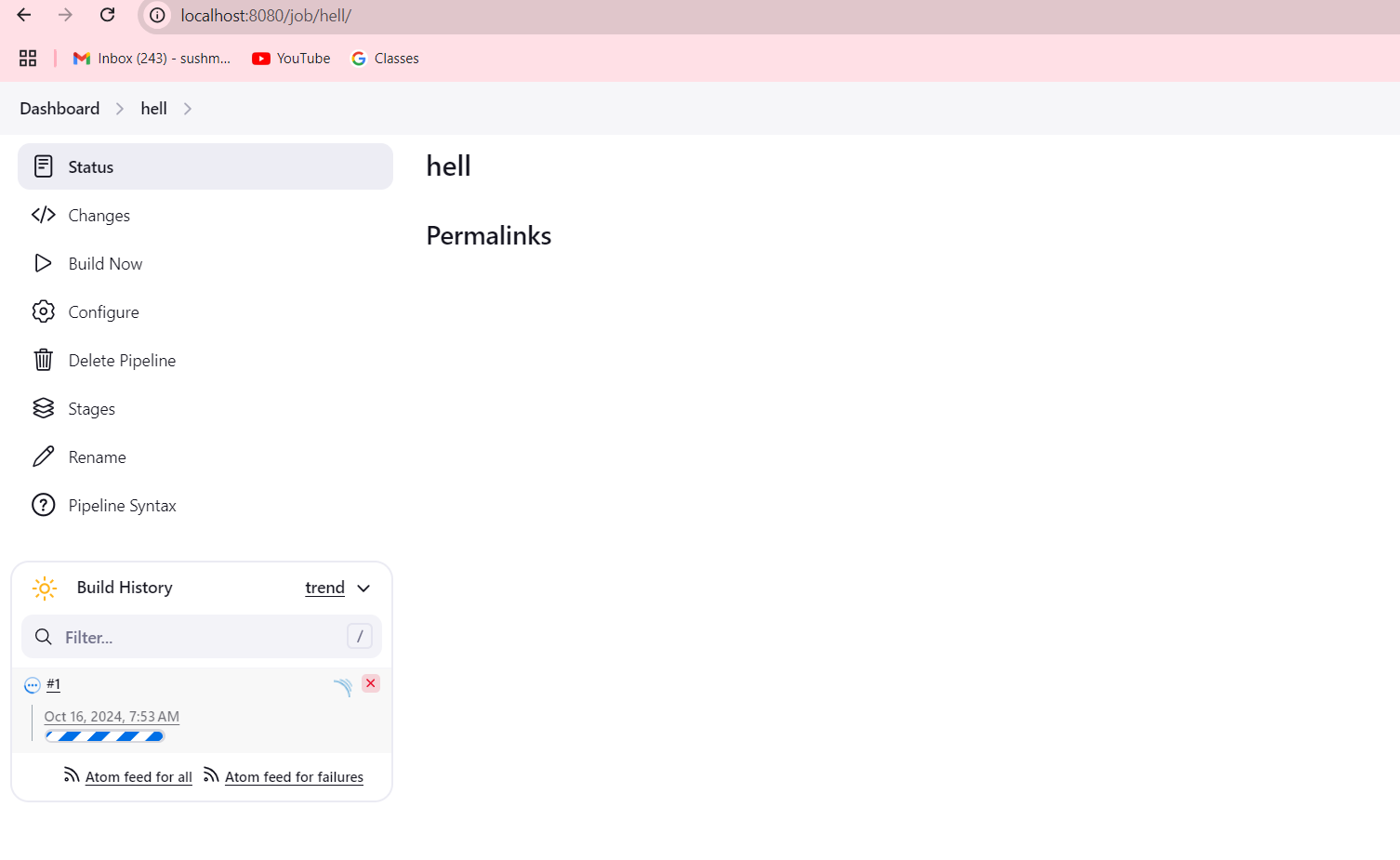
}

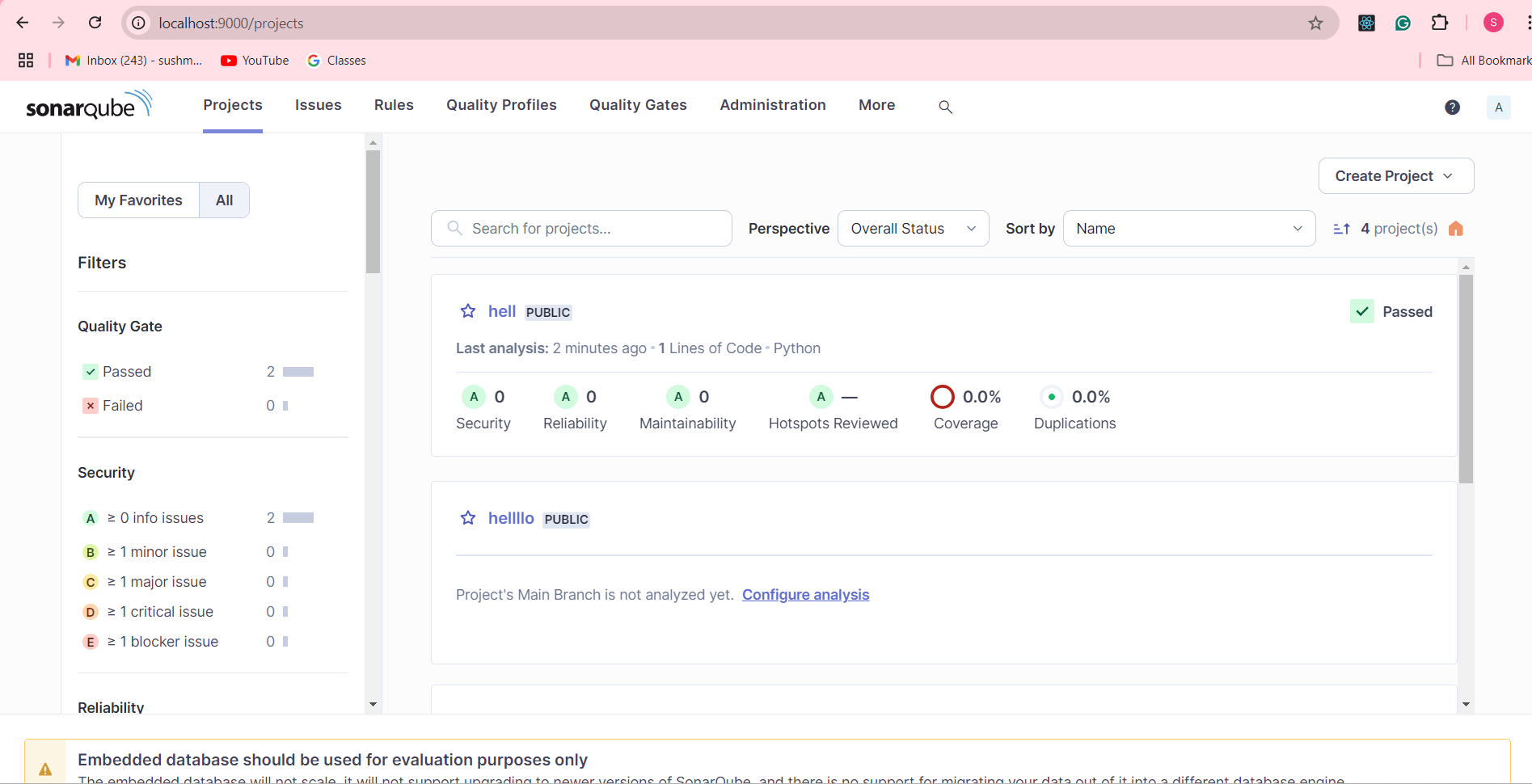
}

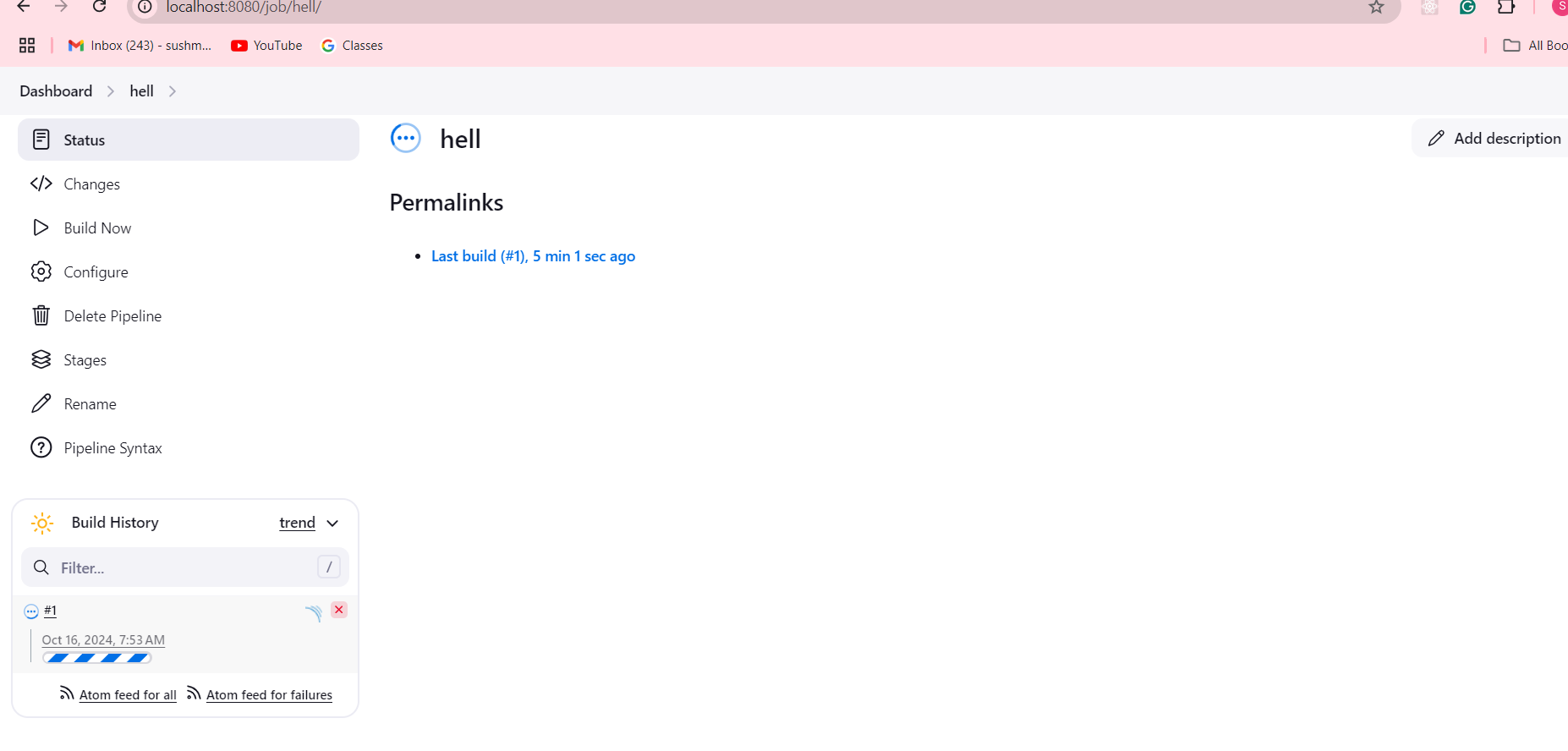
}

Do the proper changes of fle name and secret key

Then build the project in jenkin







**sqp\_8c42aa280518c8a10f9b27c4d5798a100c6849c0**

"helloworld": **sqp\_8c42aa280518c8a10f9b27c4d5798a100c6849c0**