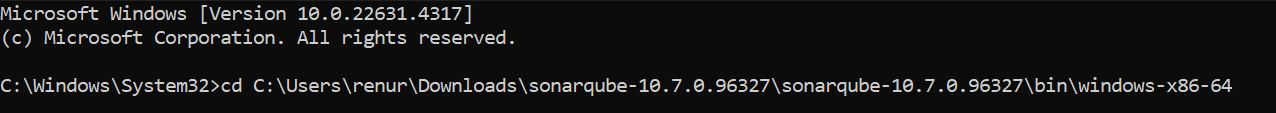
**Installing SonarQube**

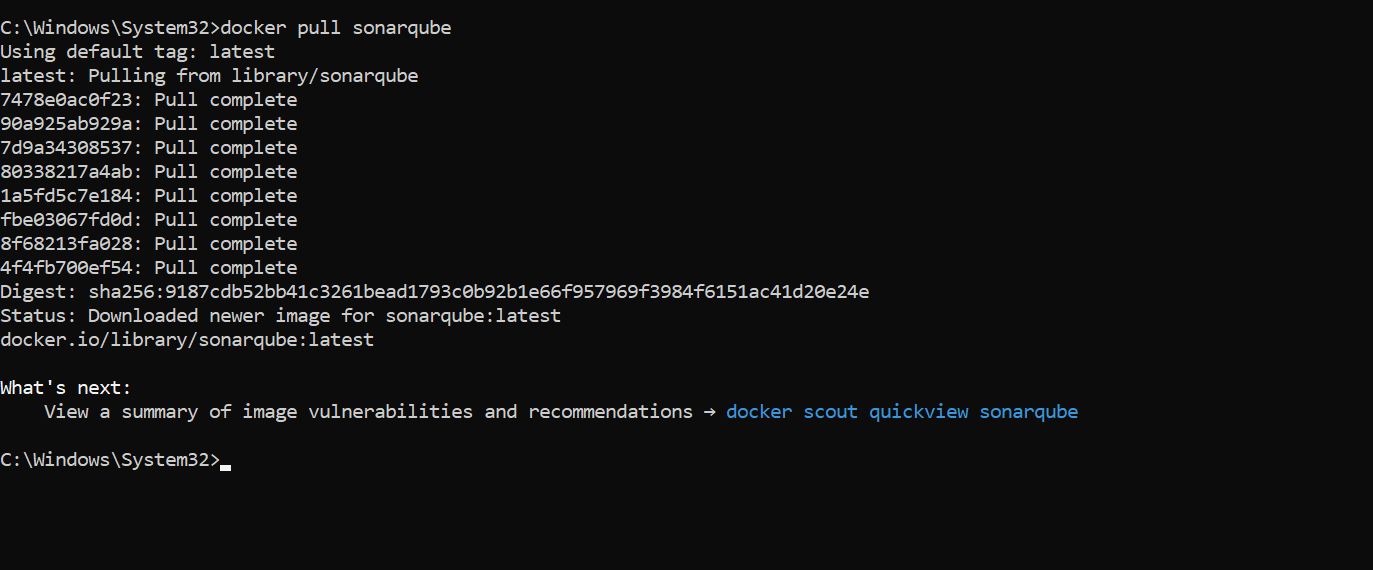
1. Go to official website of sonarqube and download the latest community edition
2. Extract the zip file
3. Navigate inside the sonarqube/sonarqube/bin/windows-x86-64. Check if a StartSonar.bat file is present.
4. If the StartSonar.bat file is present then copy the path of that folder
5. Open CMD as Run As Administrator
6. Type cd C:\Users\renur\Downloads\sonarqube-10.7.0.96327\sonarqube-10.7.0.96327\bin\windows-x86-64



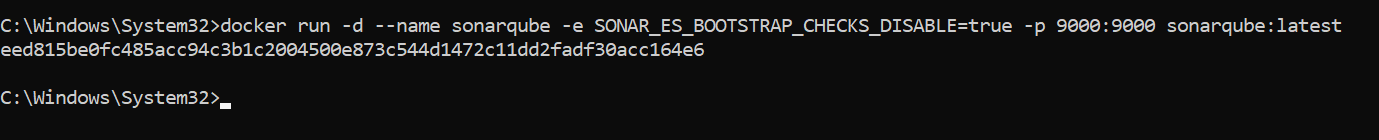
1. Type StartSonar.bat Gives errors:
2. **New approach: Using Docker image**
3. Open DockerDesktop. And pull the sonarqube image



1. Pull the image

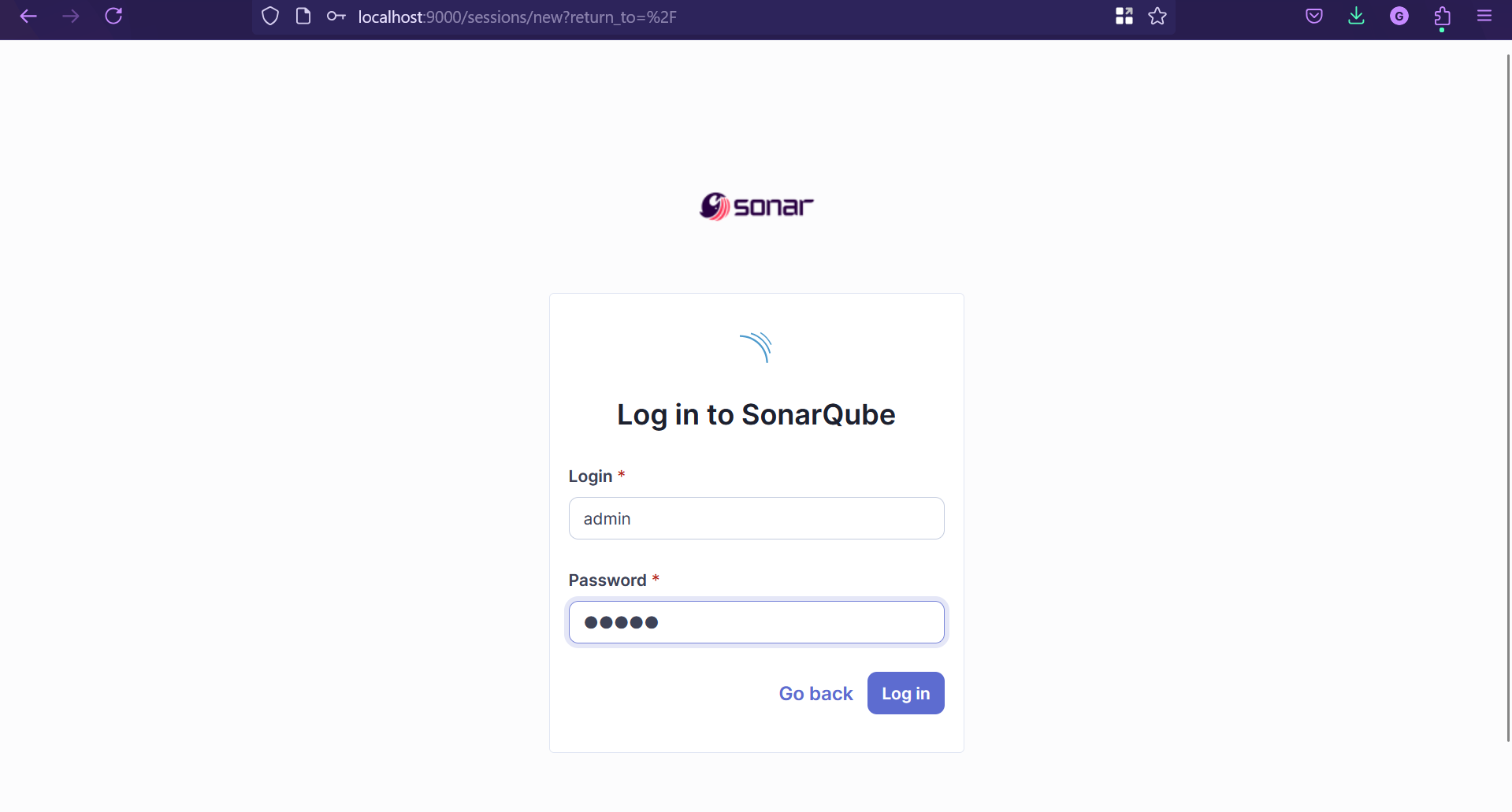


1. Run the image: docker run -d --name sonarqube -e SONAR\_ES\_BOOTSTRAP\_CHECKS\_DISABLE=true -p 9000:9000 sonarqube:latest

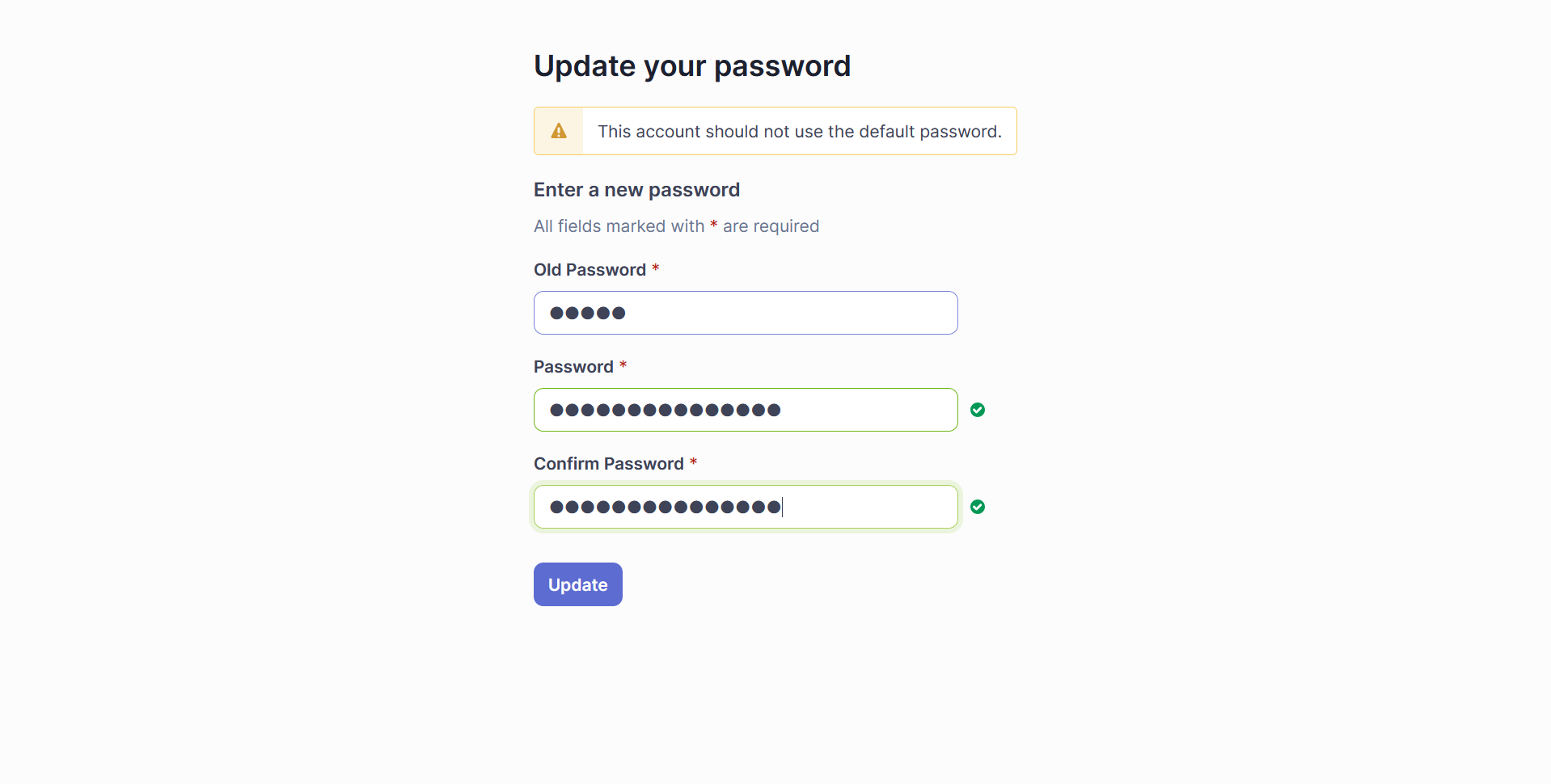




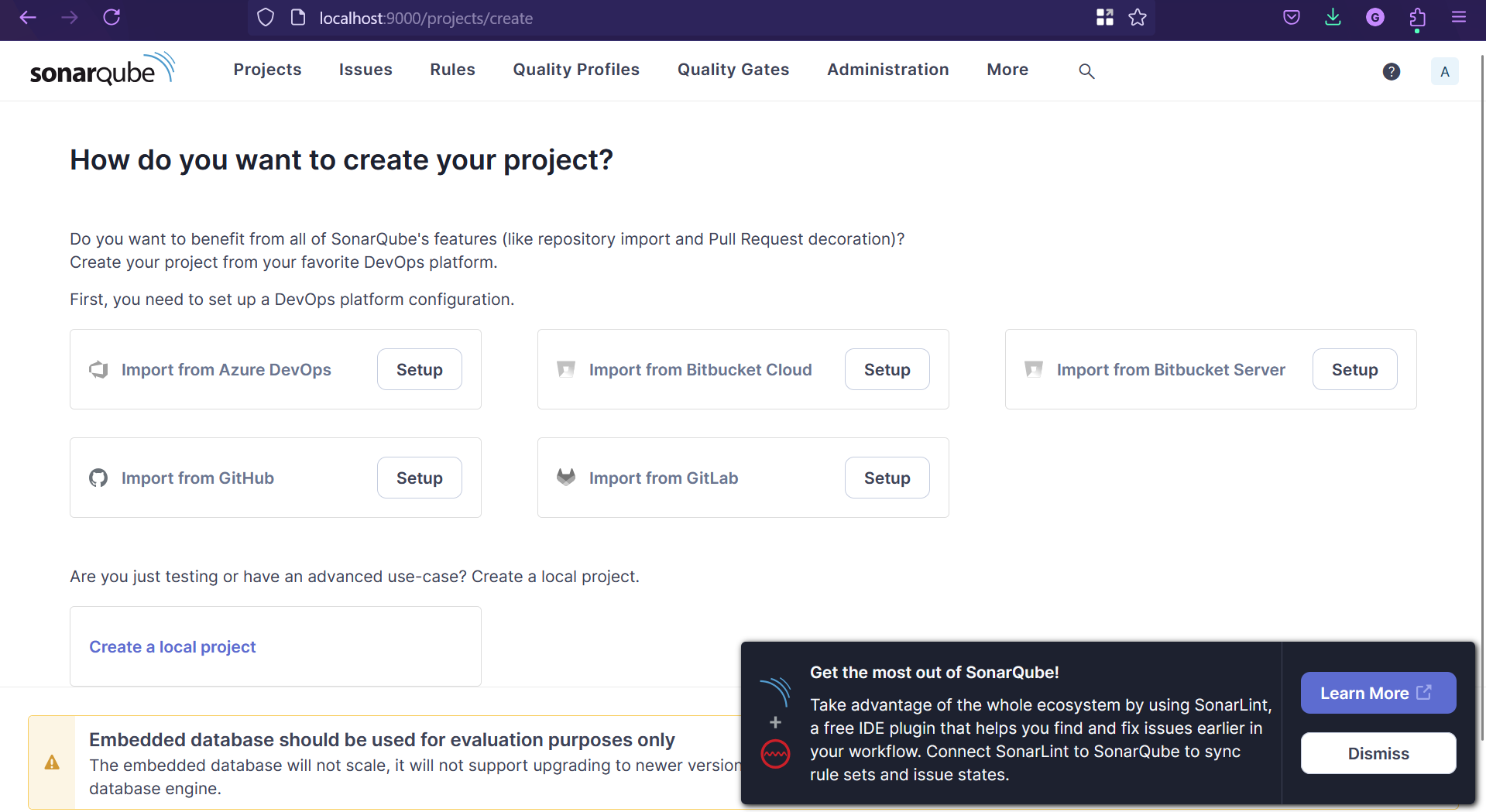
1. SonarQube will now run at localhost:9000
2. Default login = admin and password = admin



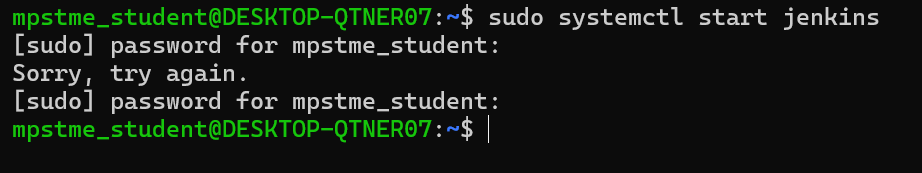
1. Update your password



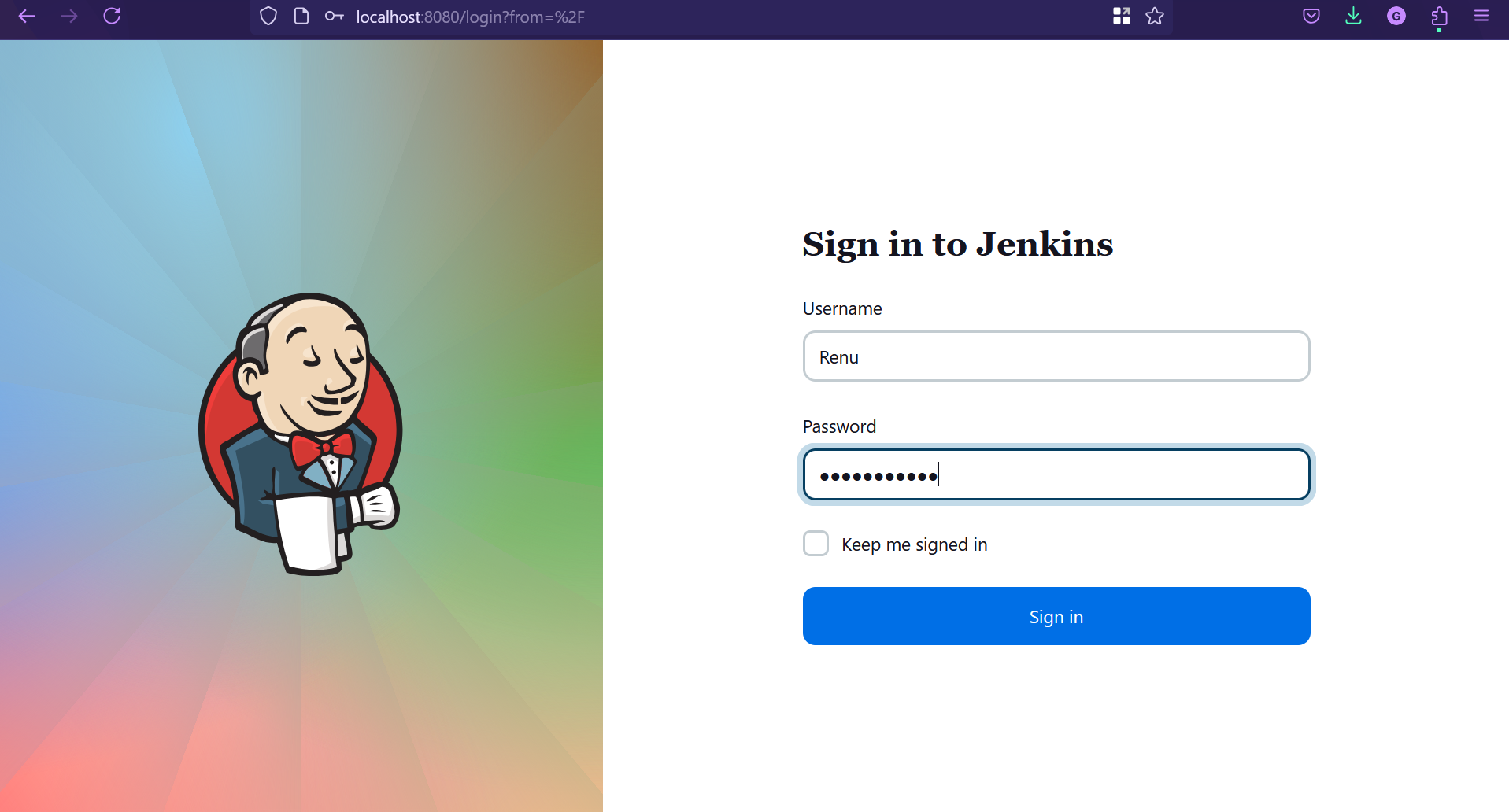
1. After logging in



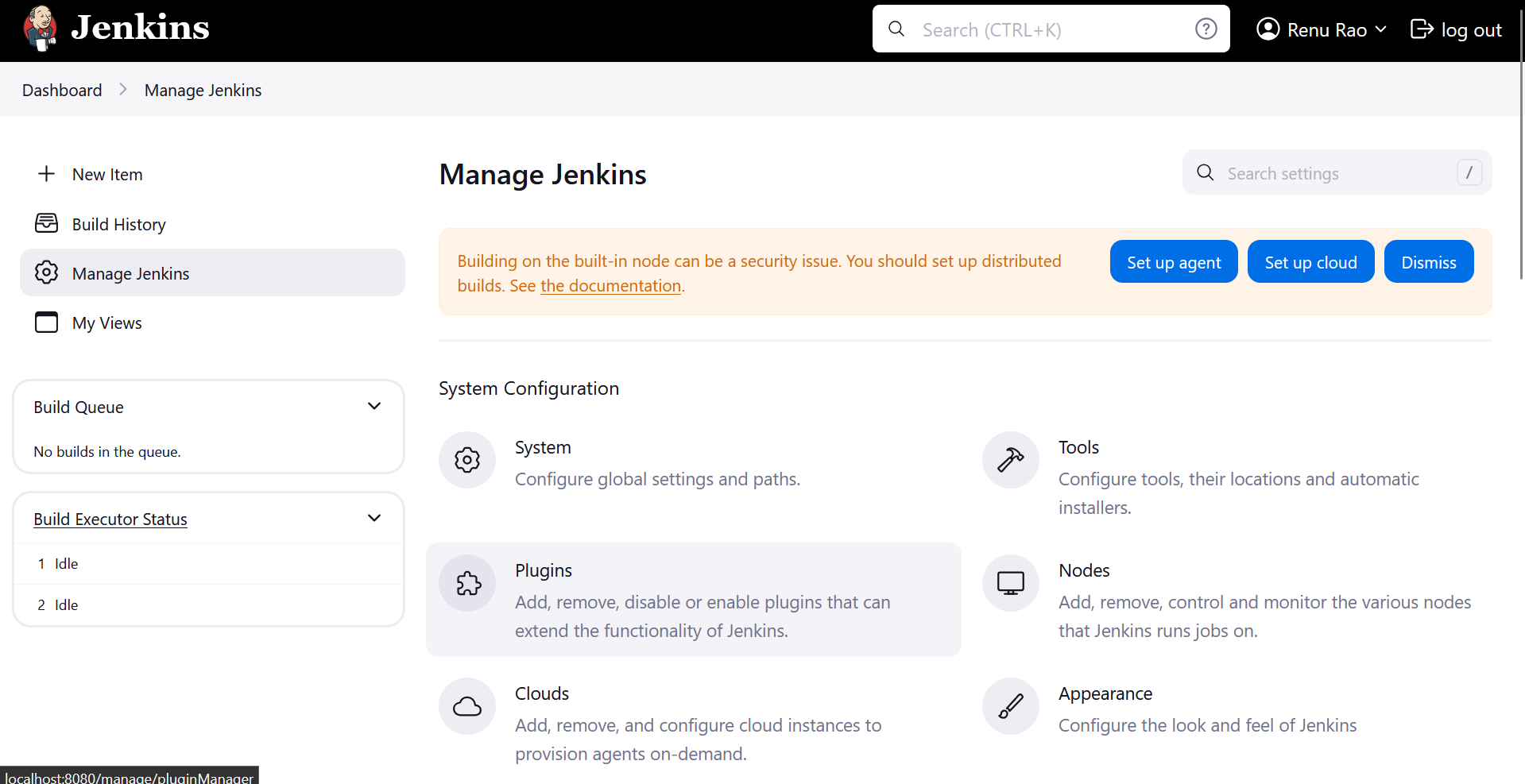
1. **Steps to integrate SonarQube with Jenkins.**
2. Start Jenkins on WSL Ubuntu: sudo systemctl start Jenkins



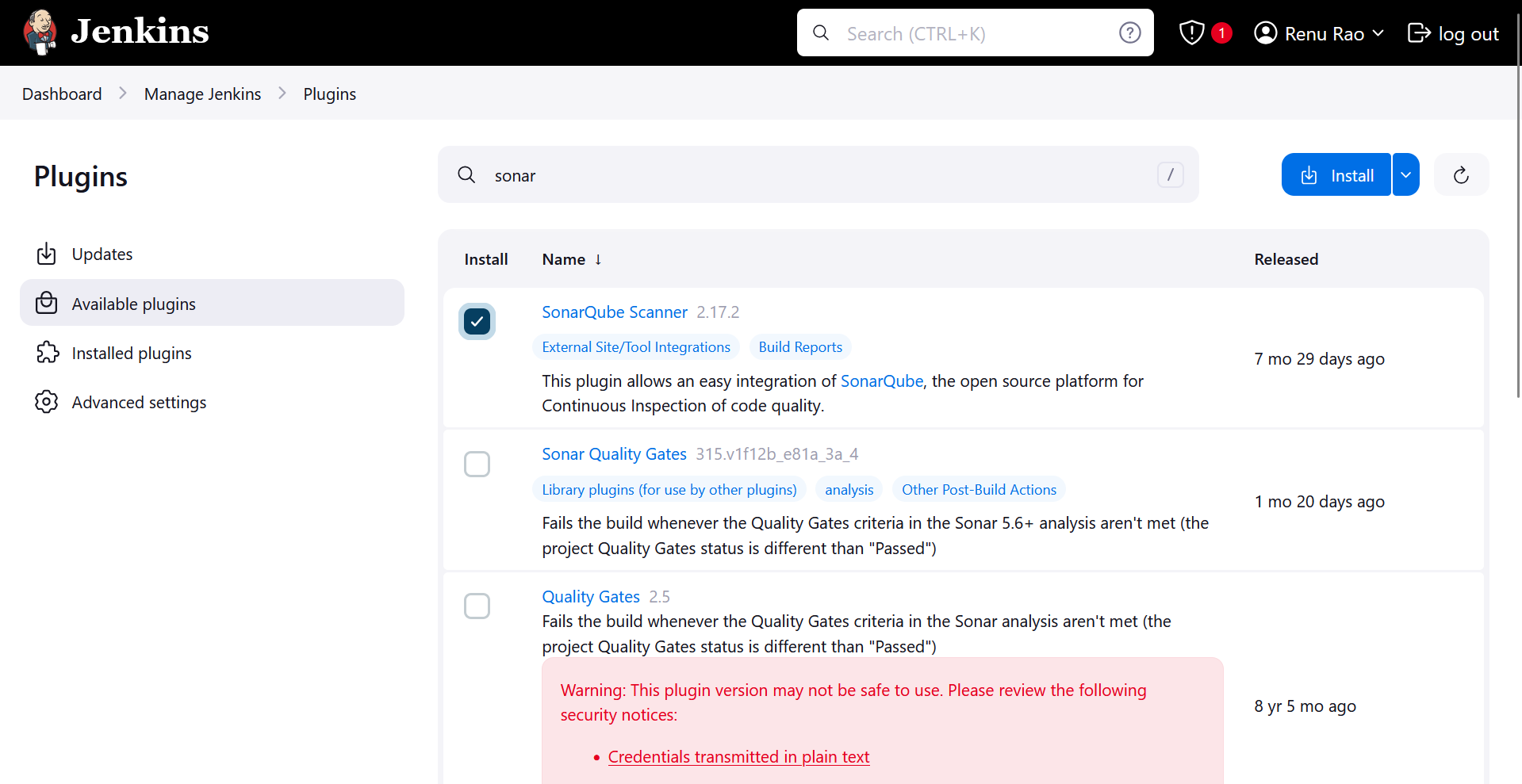
1. Go to localhost:8080 and give your Username and Password



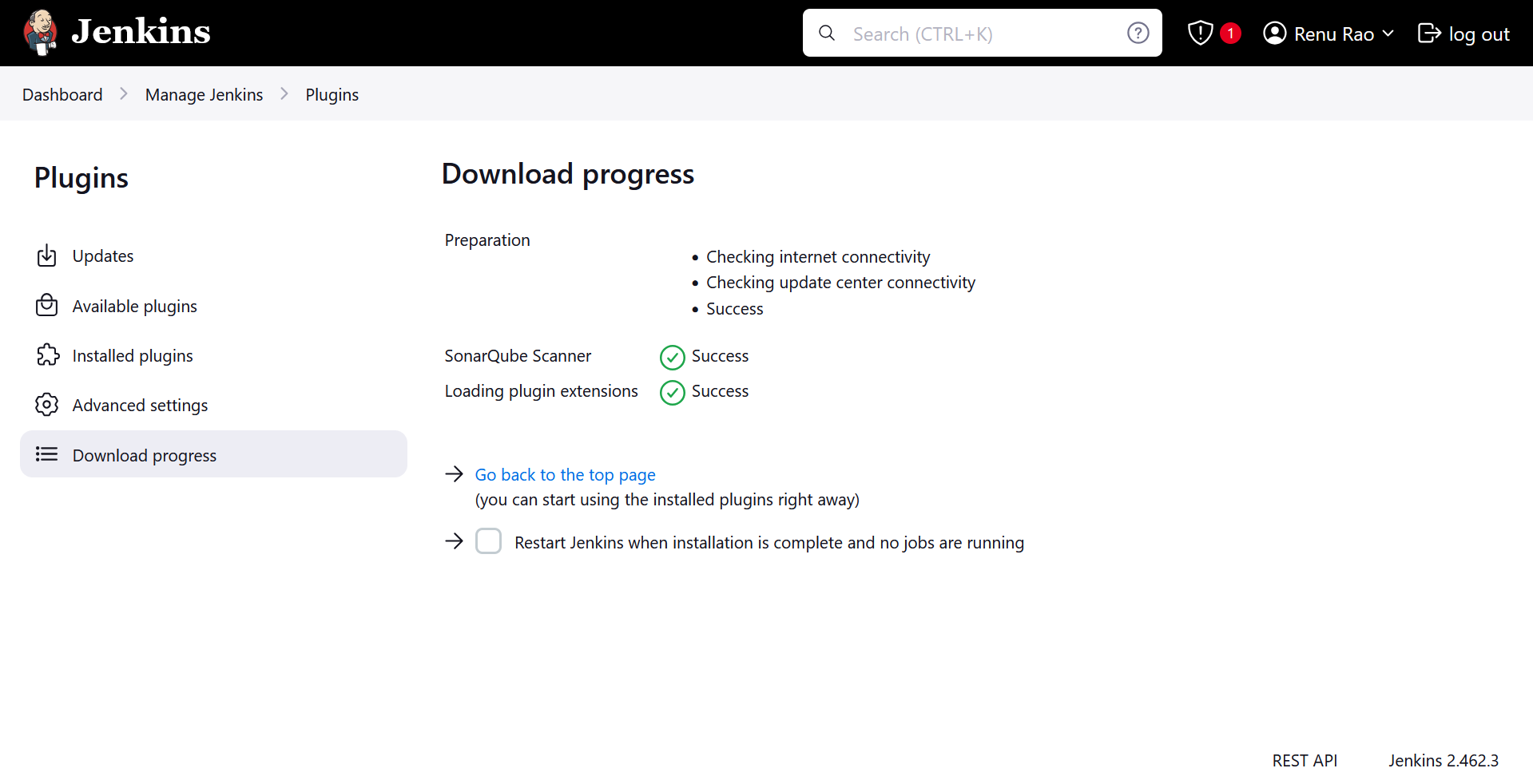
1. Manage Jenkins -> Plugins



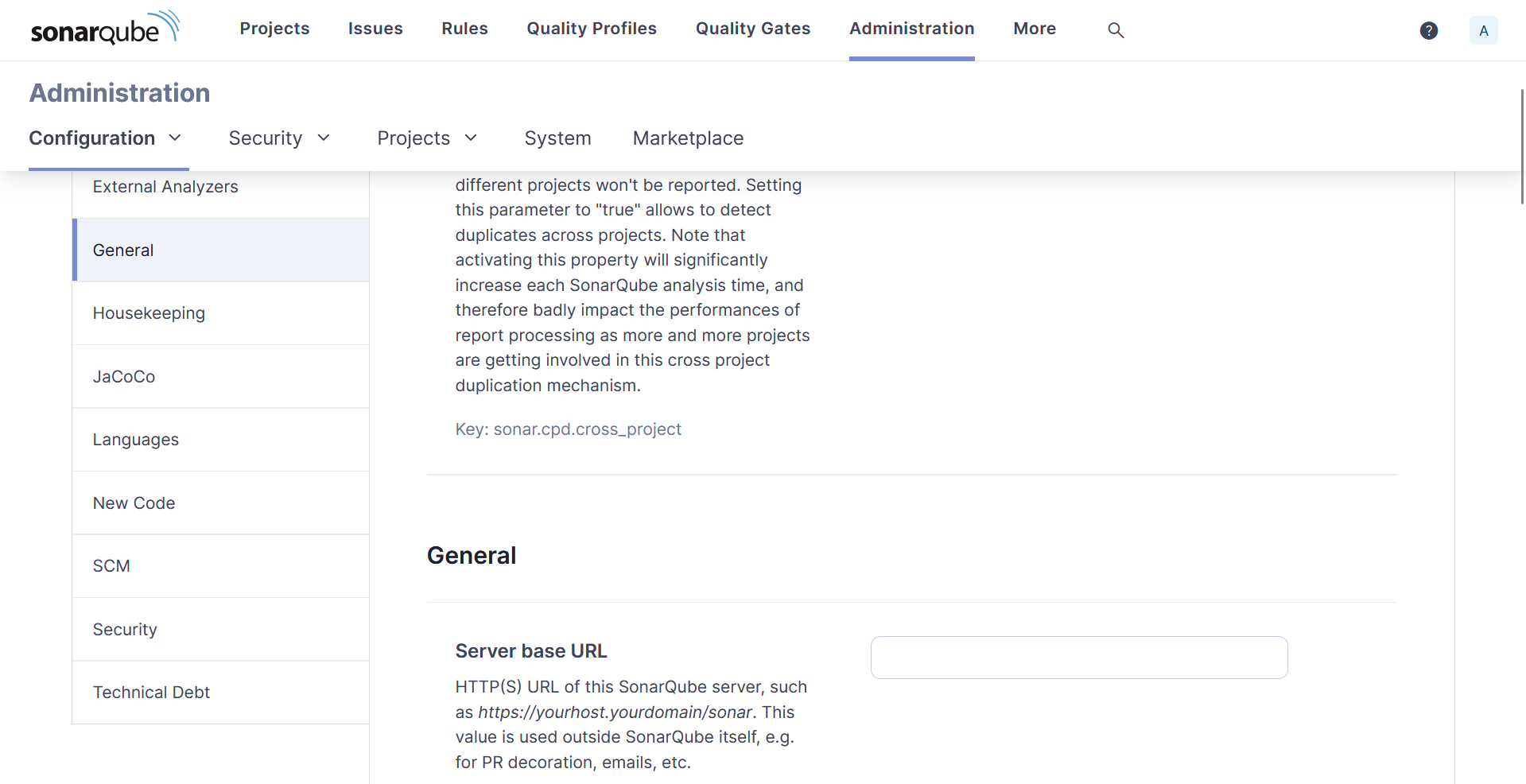
1. Avaible Plugins -> Search sonar -> Click on SonarQubeScanner -> Install



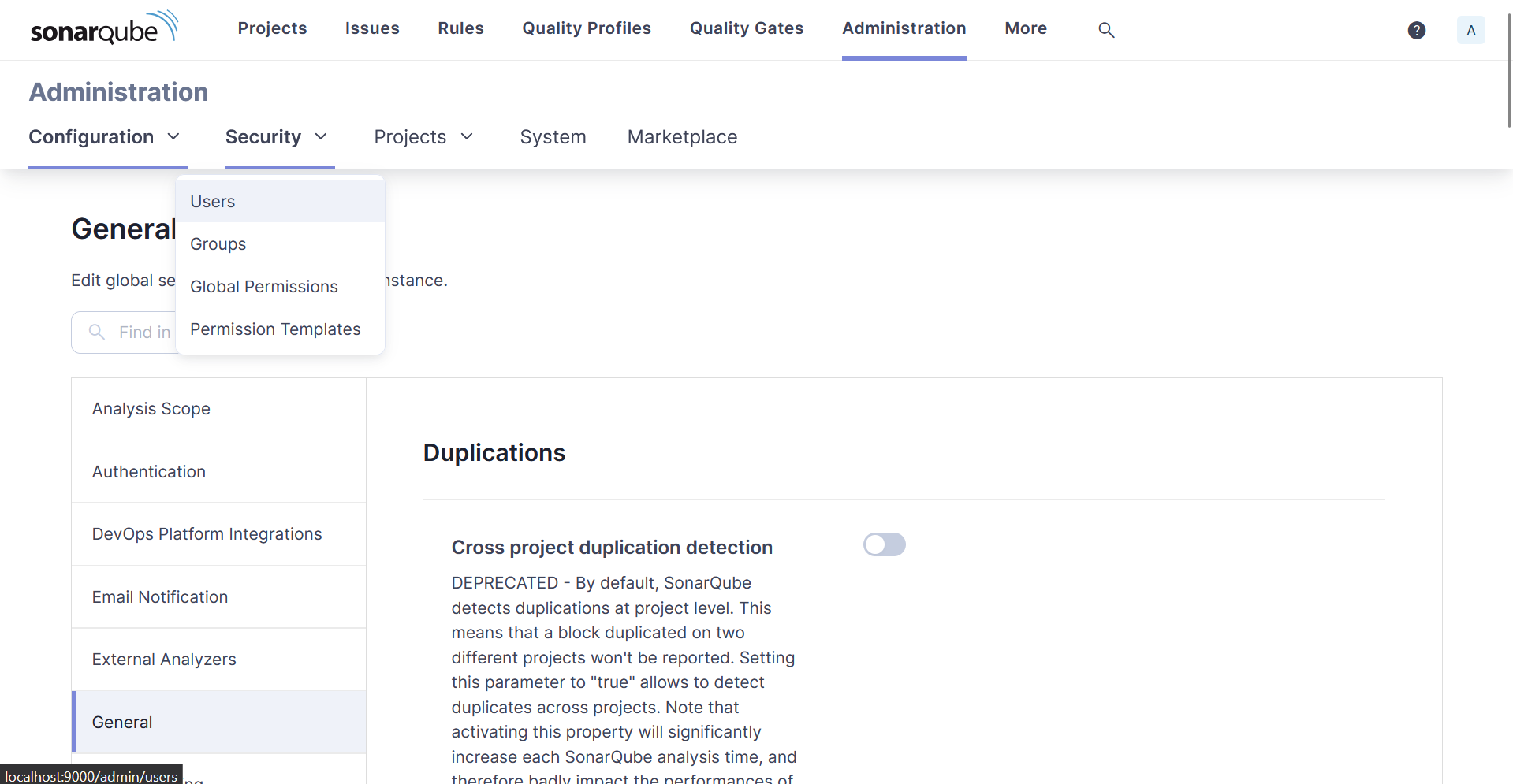
1. After installing



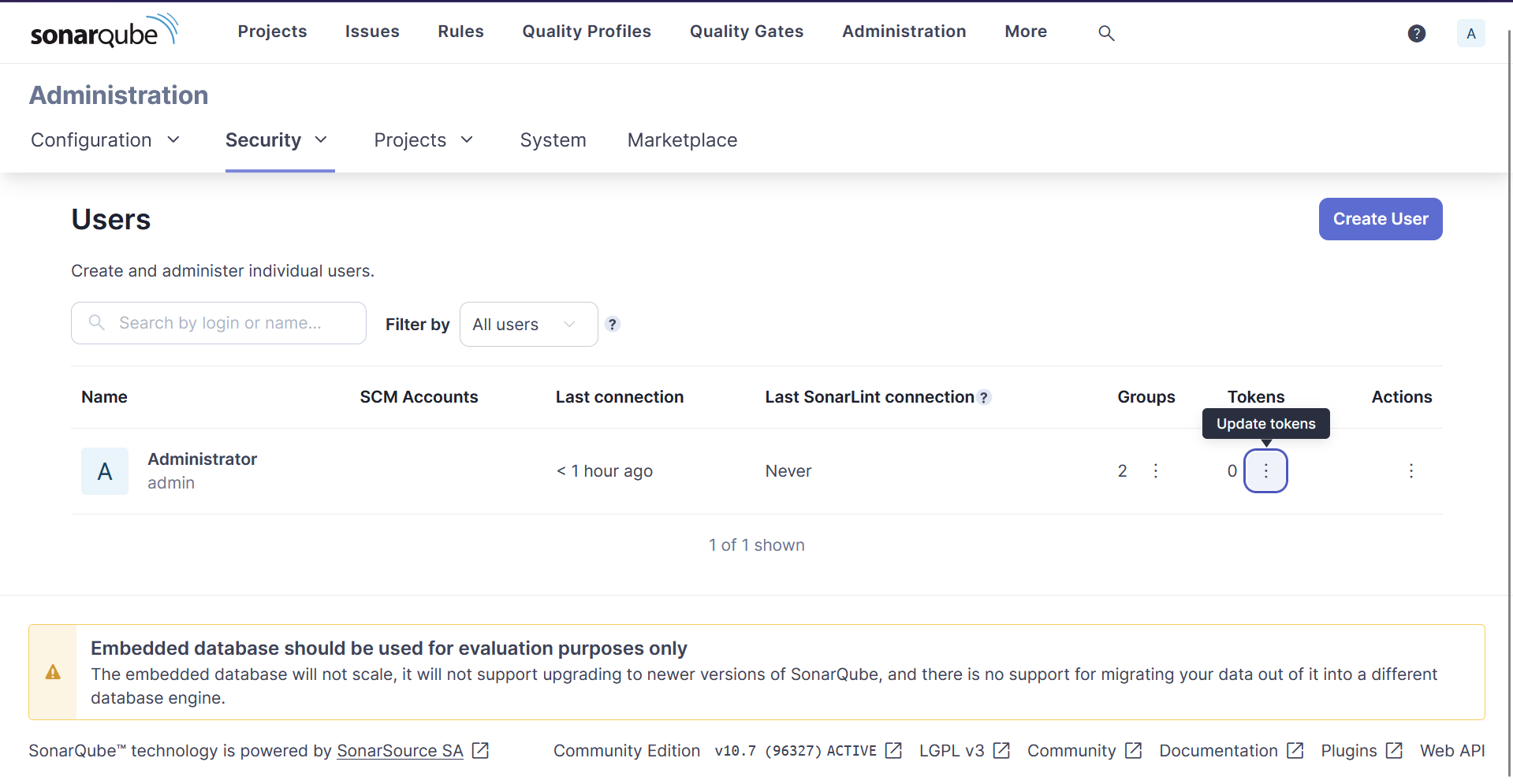
1. Go to localhost:9000 -> Administration -> Security

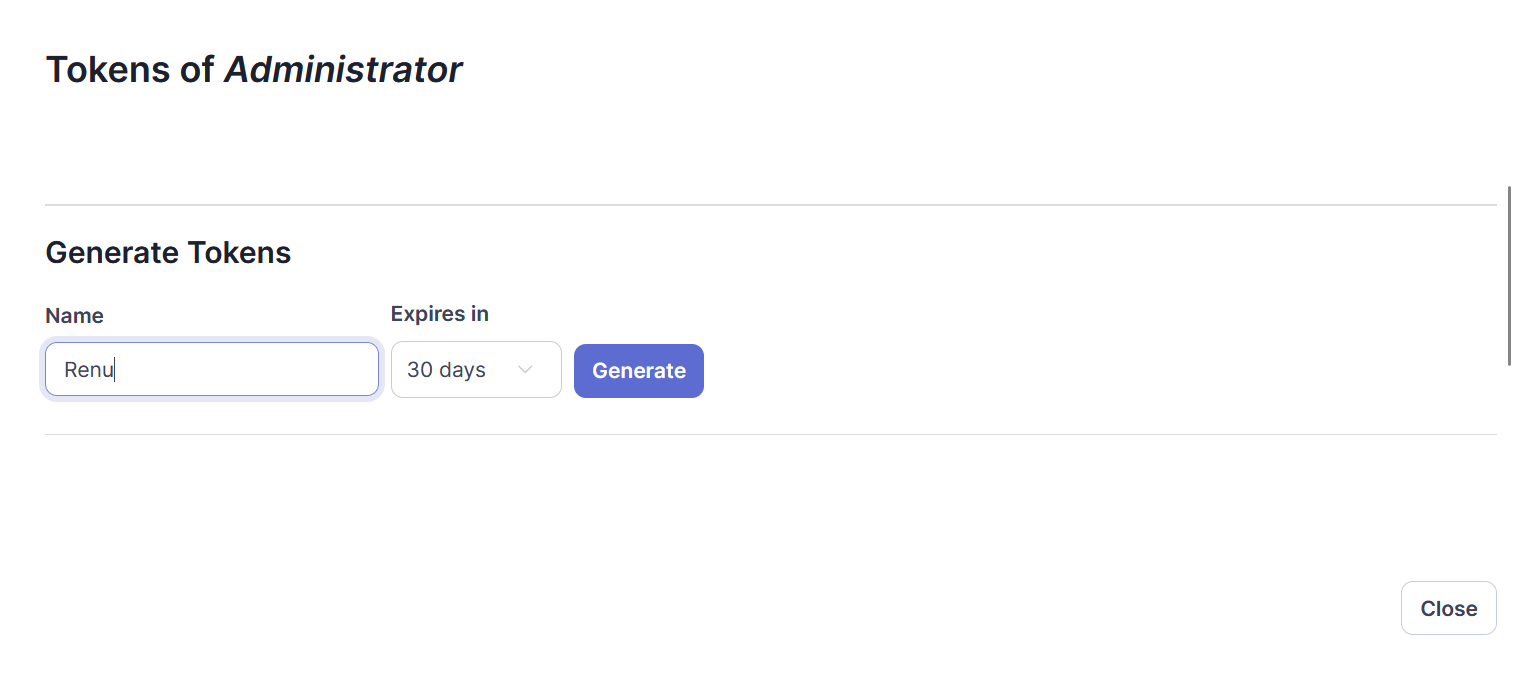


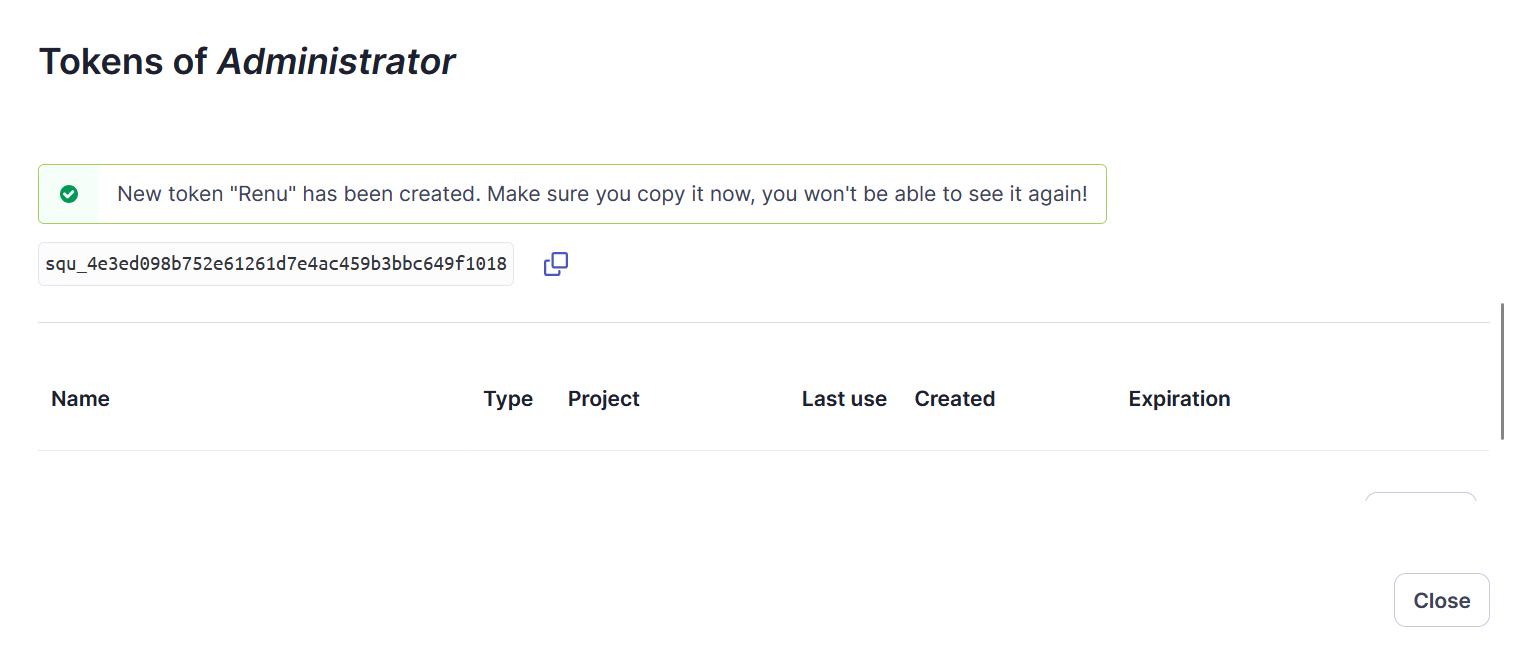
1. In Secutity dropdown -> select Users

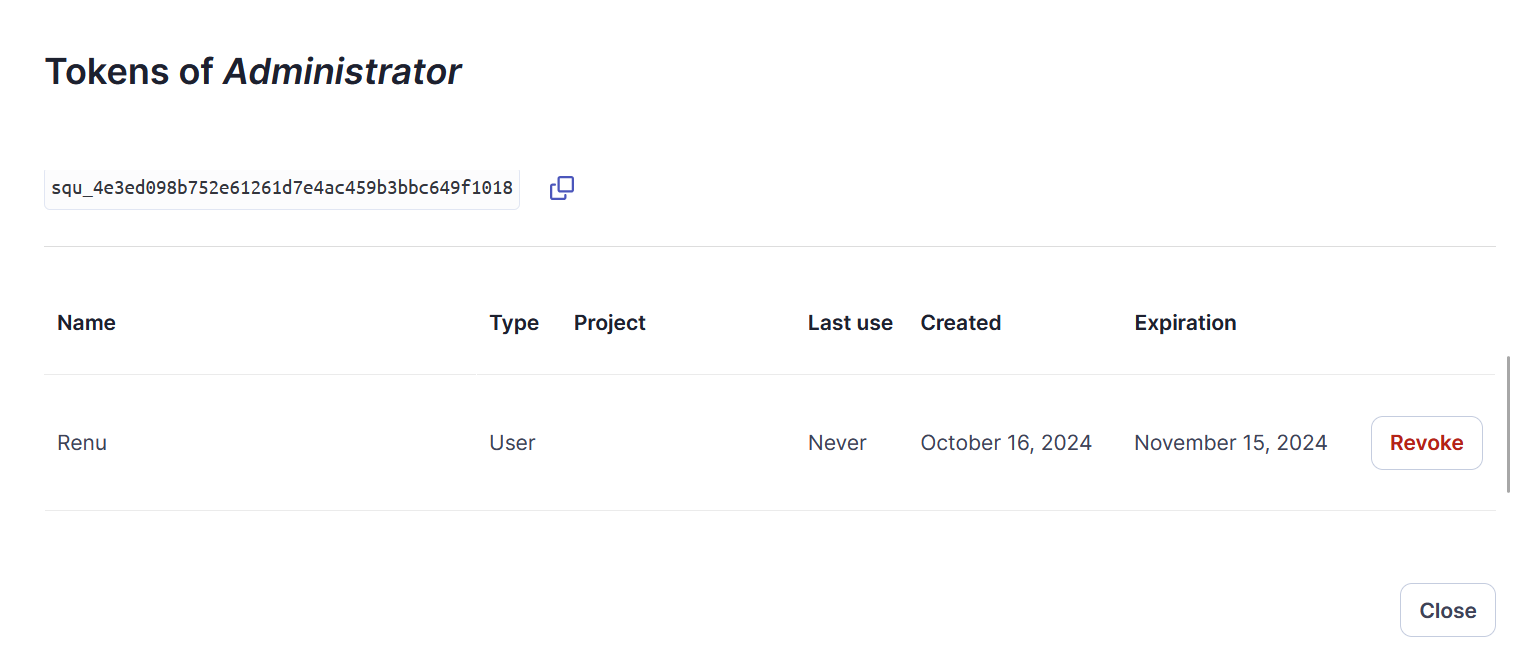


1. Click on 3 dots -> Generate token with some name



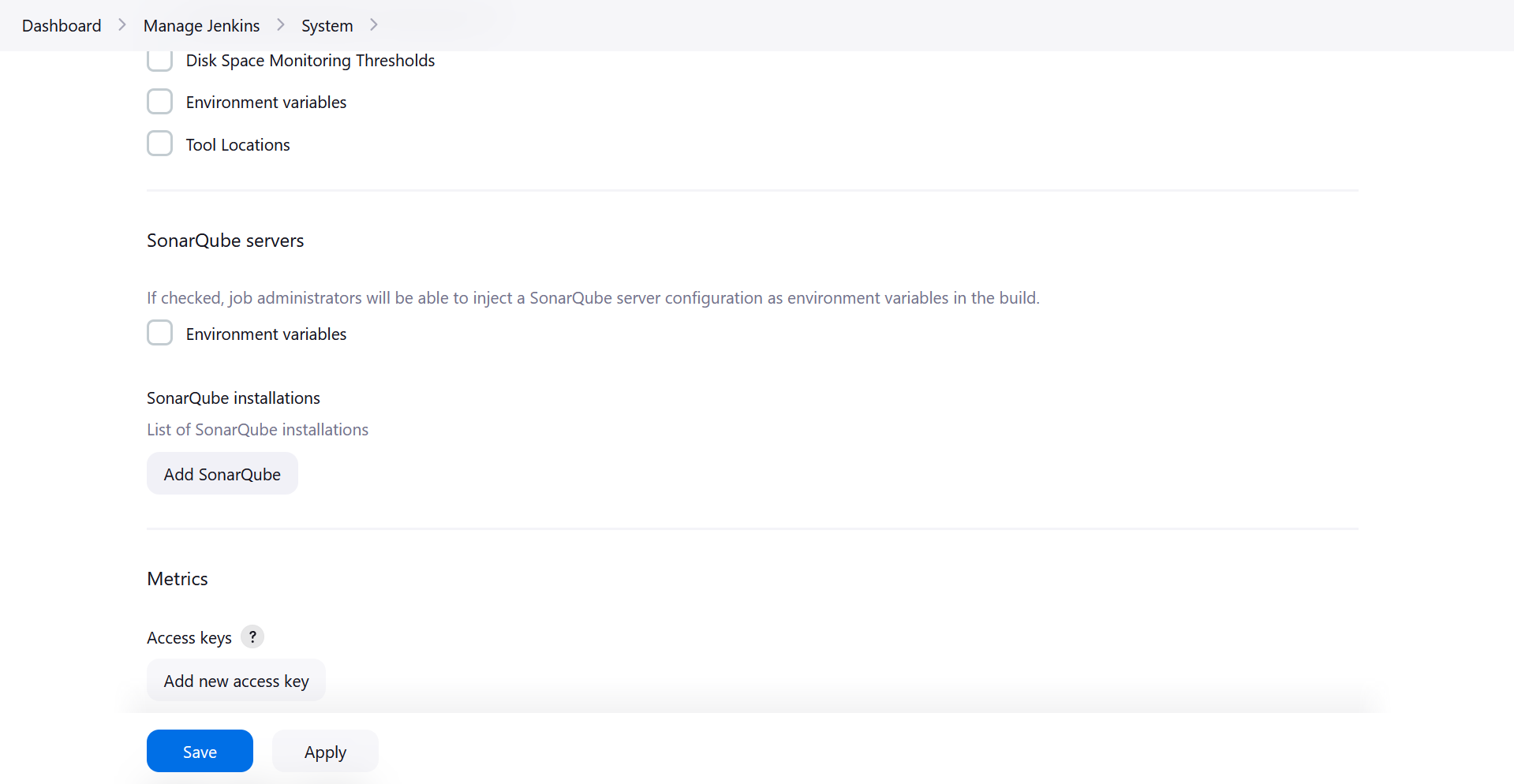




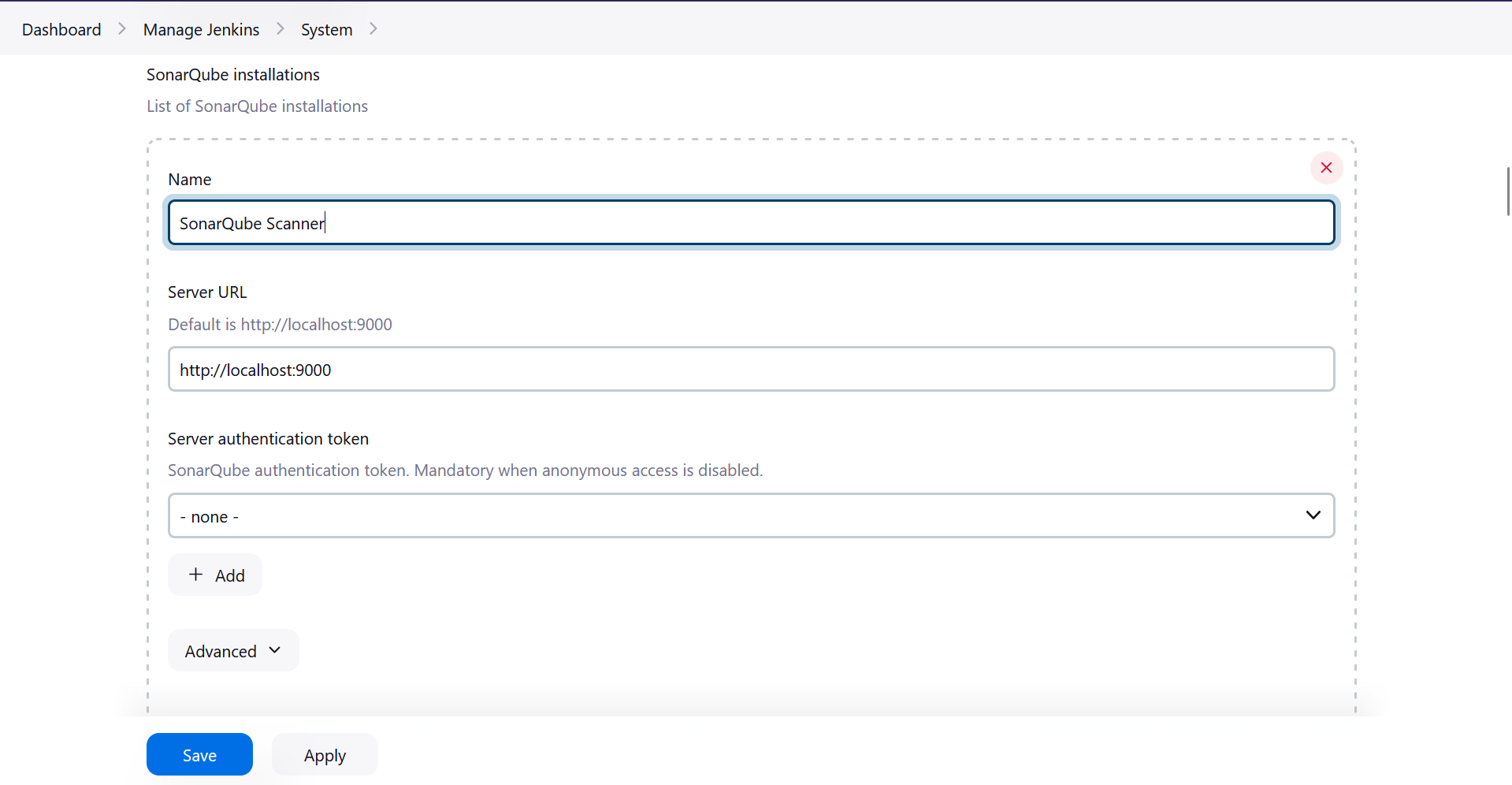


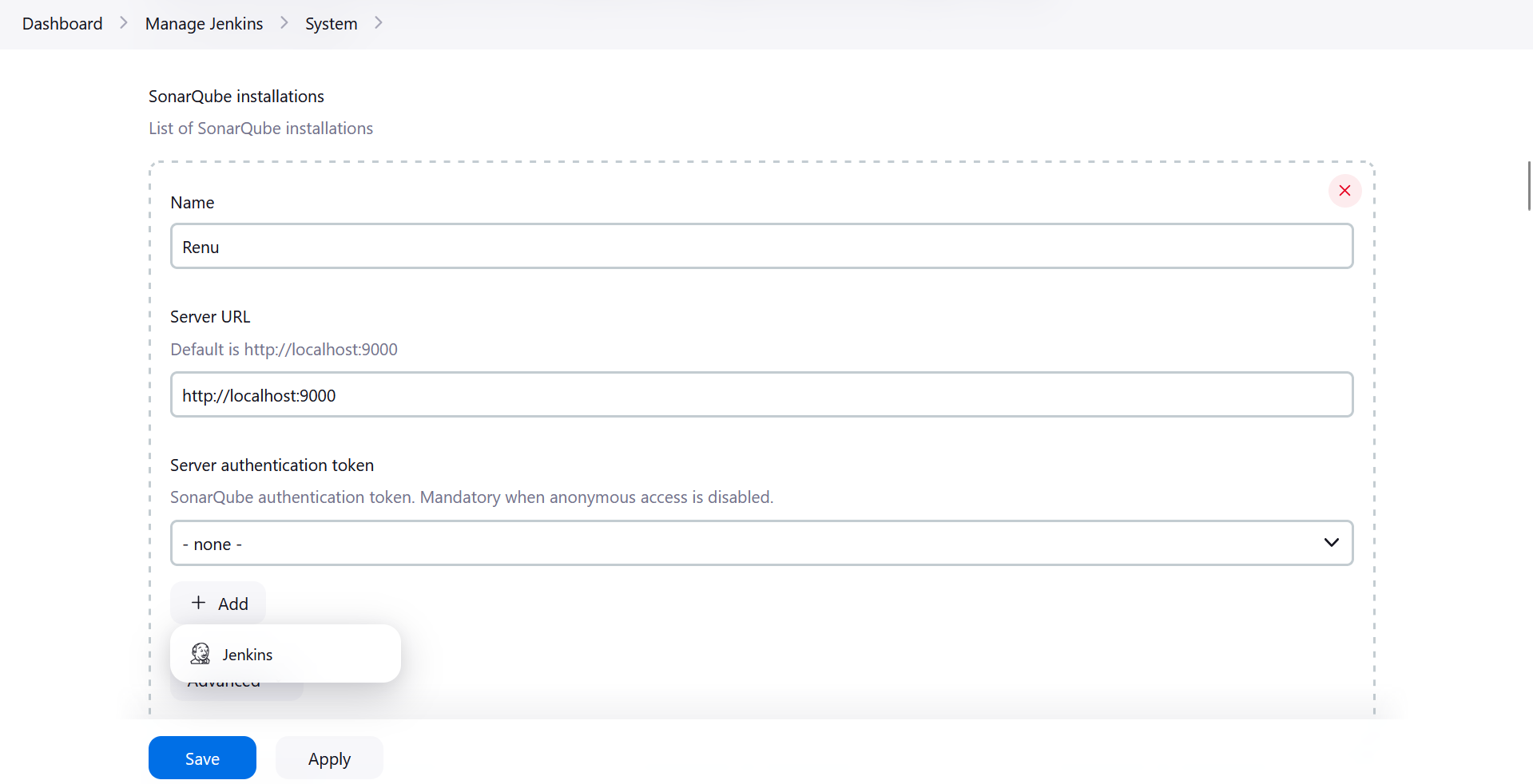
squ\_4e3ed098b752e61261d7e4ac459b3bbc649f1018

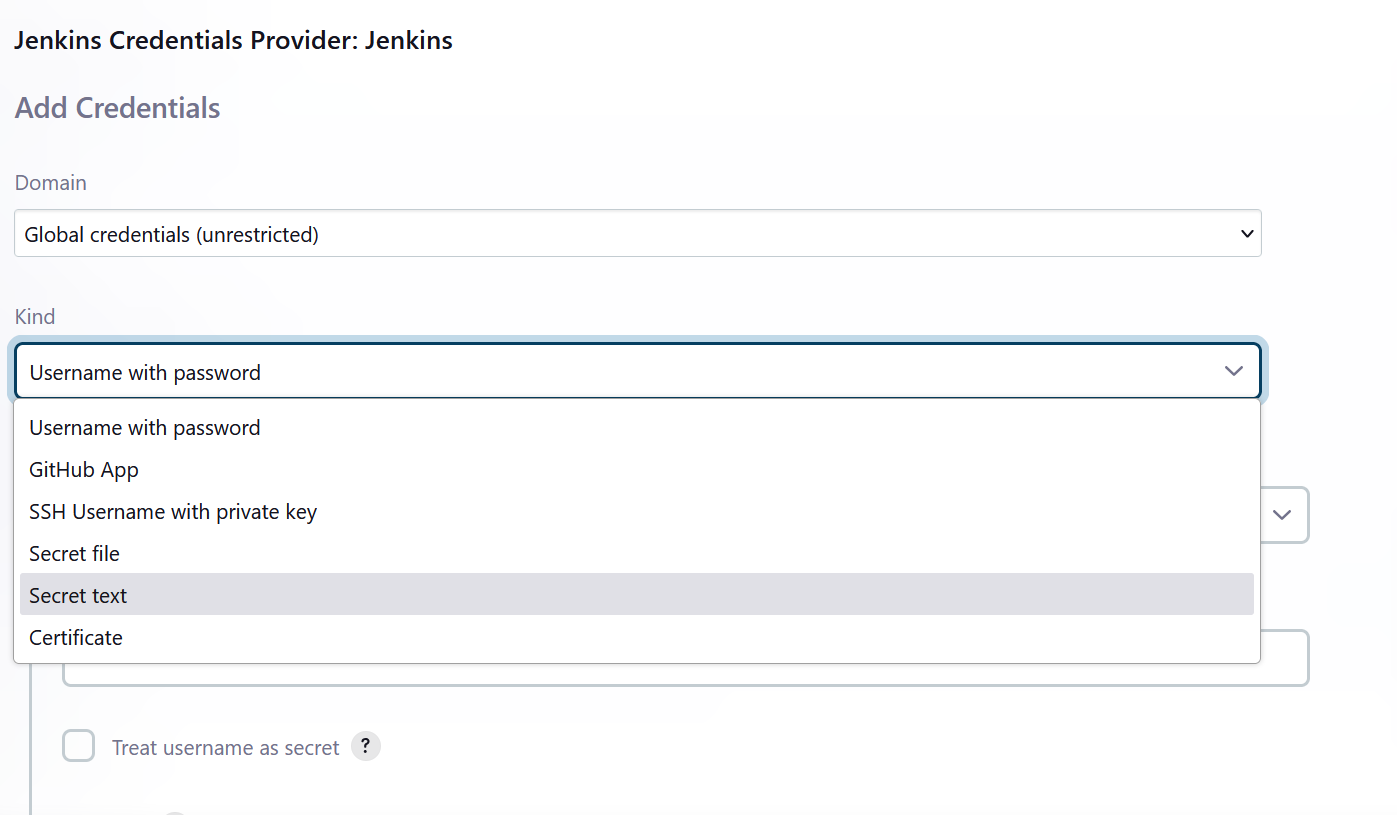
1. Go to localhost:8080 -> Manage Jenkins -> System -> Scroll down and click Add SonarQube



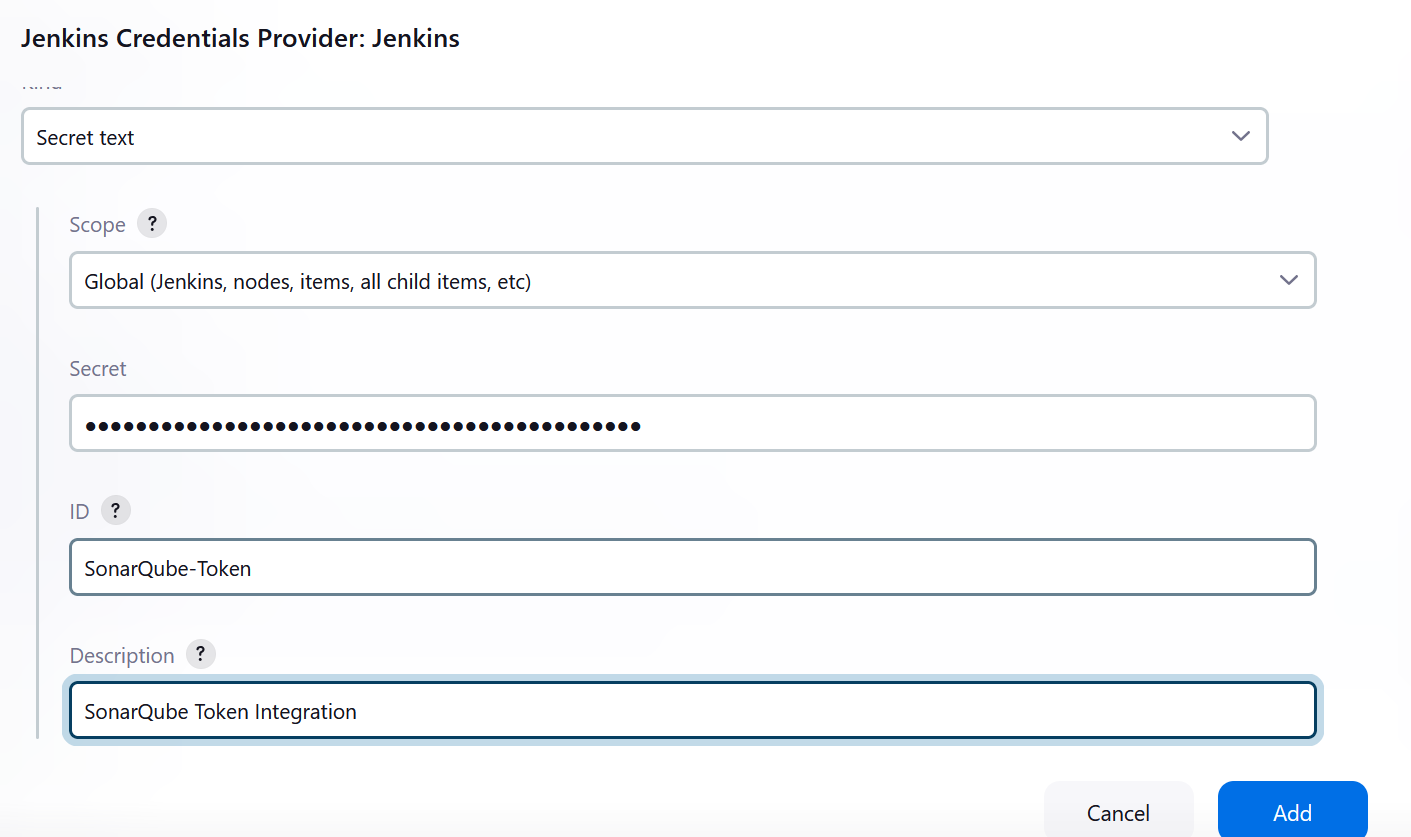
1. After click Add SonarQube



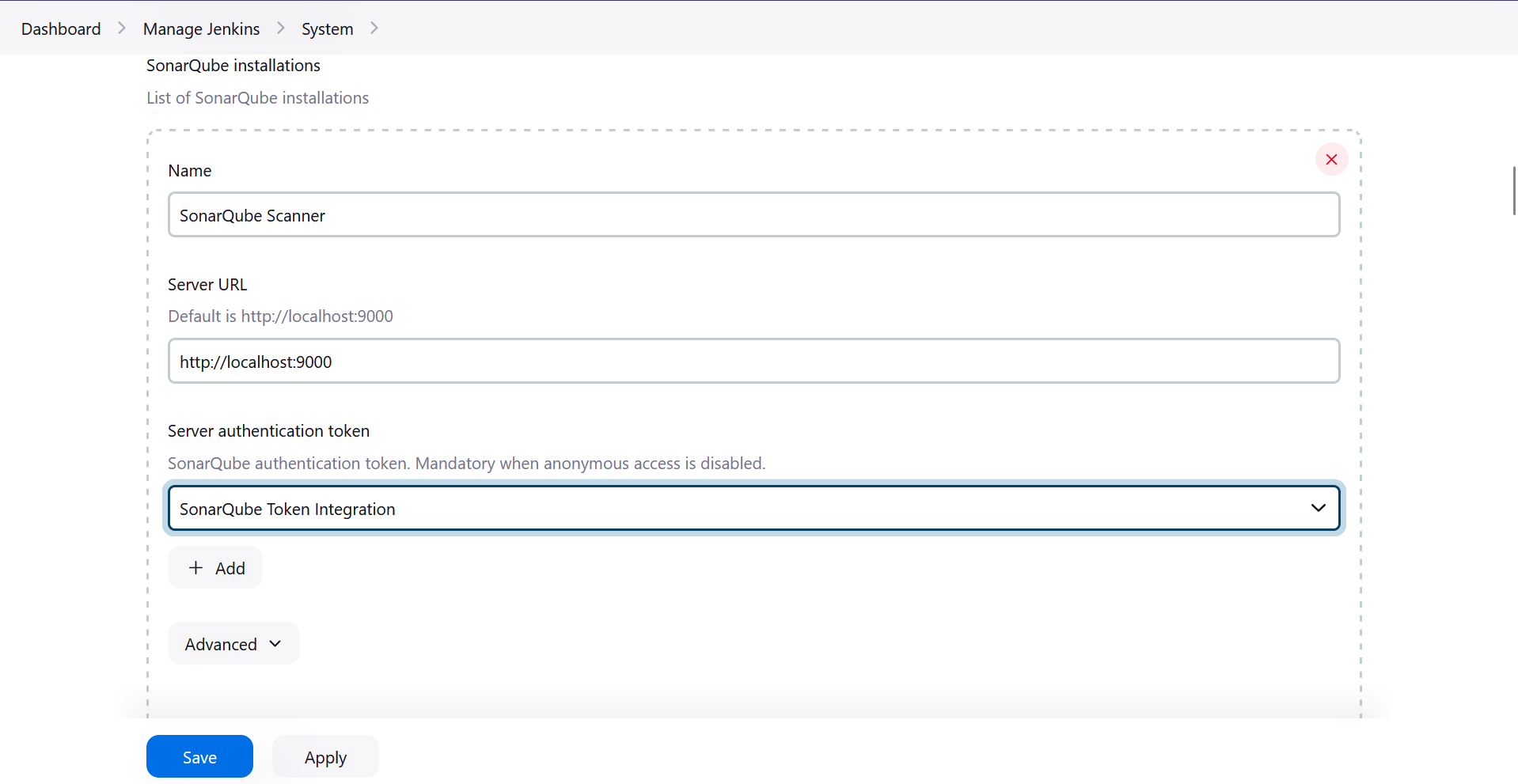




1. Paste the generate access token form Jenkins in Secret field, Put SonarQube-Token in ID and Description SonarQube Token Integration



1. Click Add -> You will get a Credentials created msg
2. Select SonarQube Token Integration



1. Hence, understood Static Analysis SAST process and successfully integrated Jenkins with SonarQube.
2. **Running Jenkins with SonarQube on a simple python project**
3. Create a folder in Downloads -> Name it as PythonProject
4. Create a file named app.py and copy paste the following:

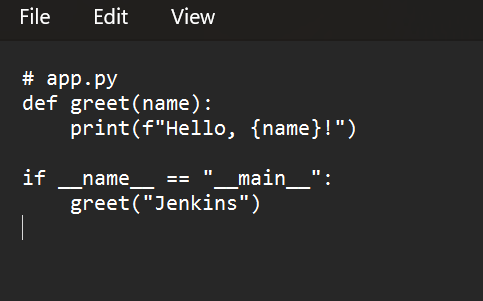
# app.py

def greet(name):

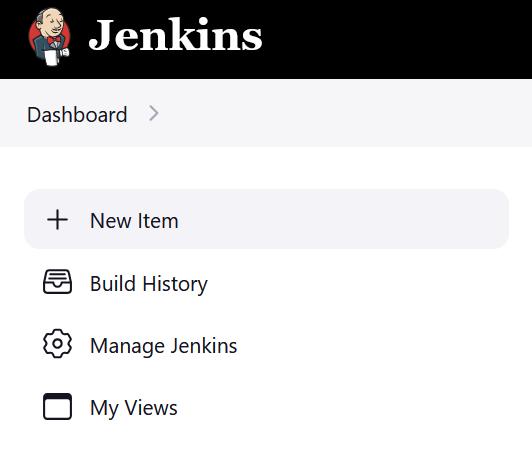
print(f"Hello, {name}!")

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

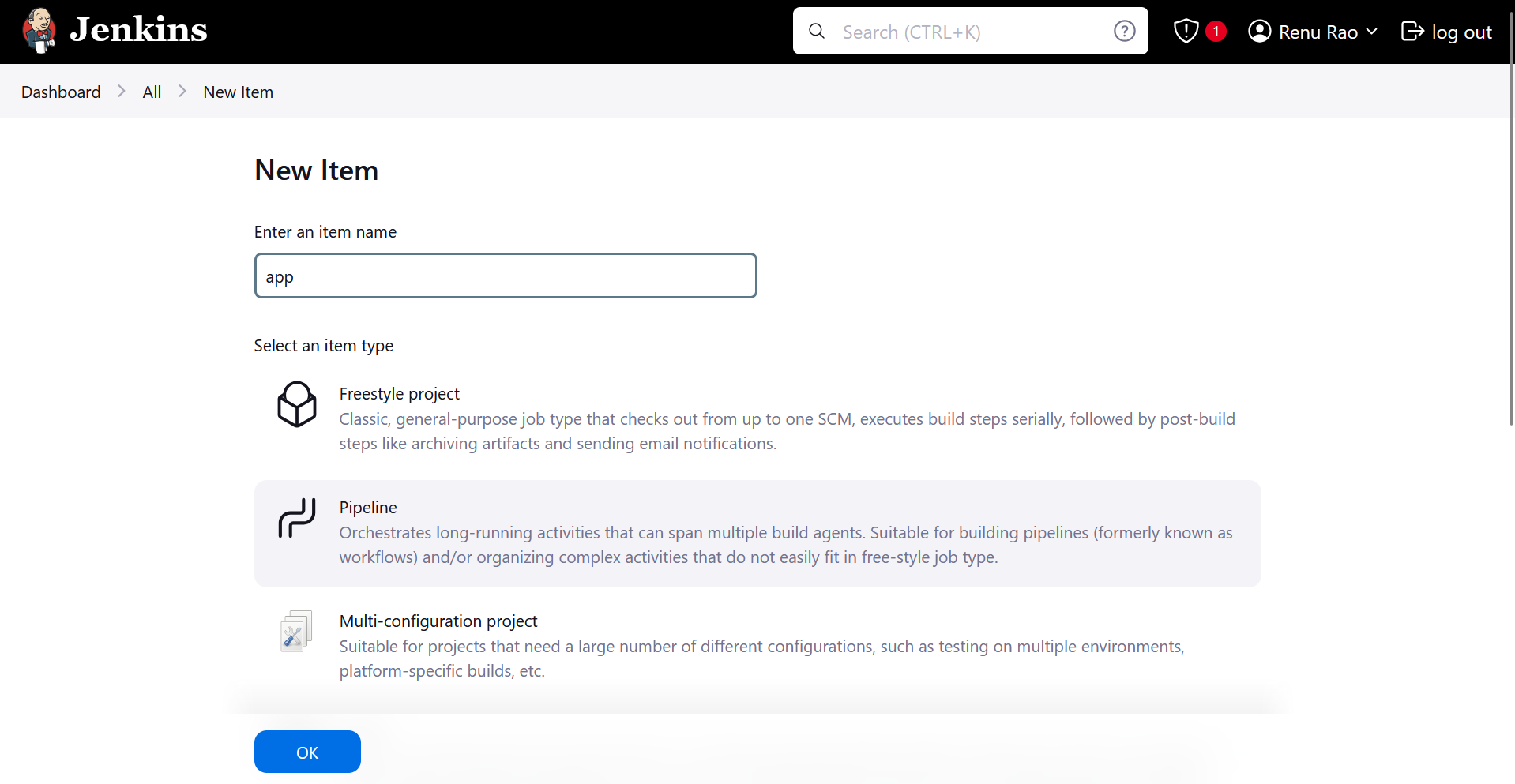
greet("Jenkins")



1. Make sure SonarQube is properly configured in your Jenkins instance as previously mentioned.
2. On Jenkins Dashboard -> Click New Item



1. Give a name -> Select Pipeline



1. In left pane, Click on Pipeline and add the following to the Pipeline Script:

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:\\Users\\renur\\Downloads\\PythonProject\\app.py' // Use bat for Windows commands

}

}

stage('SonarQube Analysis') {

steps {

withSonarQubeEnv('SonarQube') { // '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 {

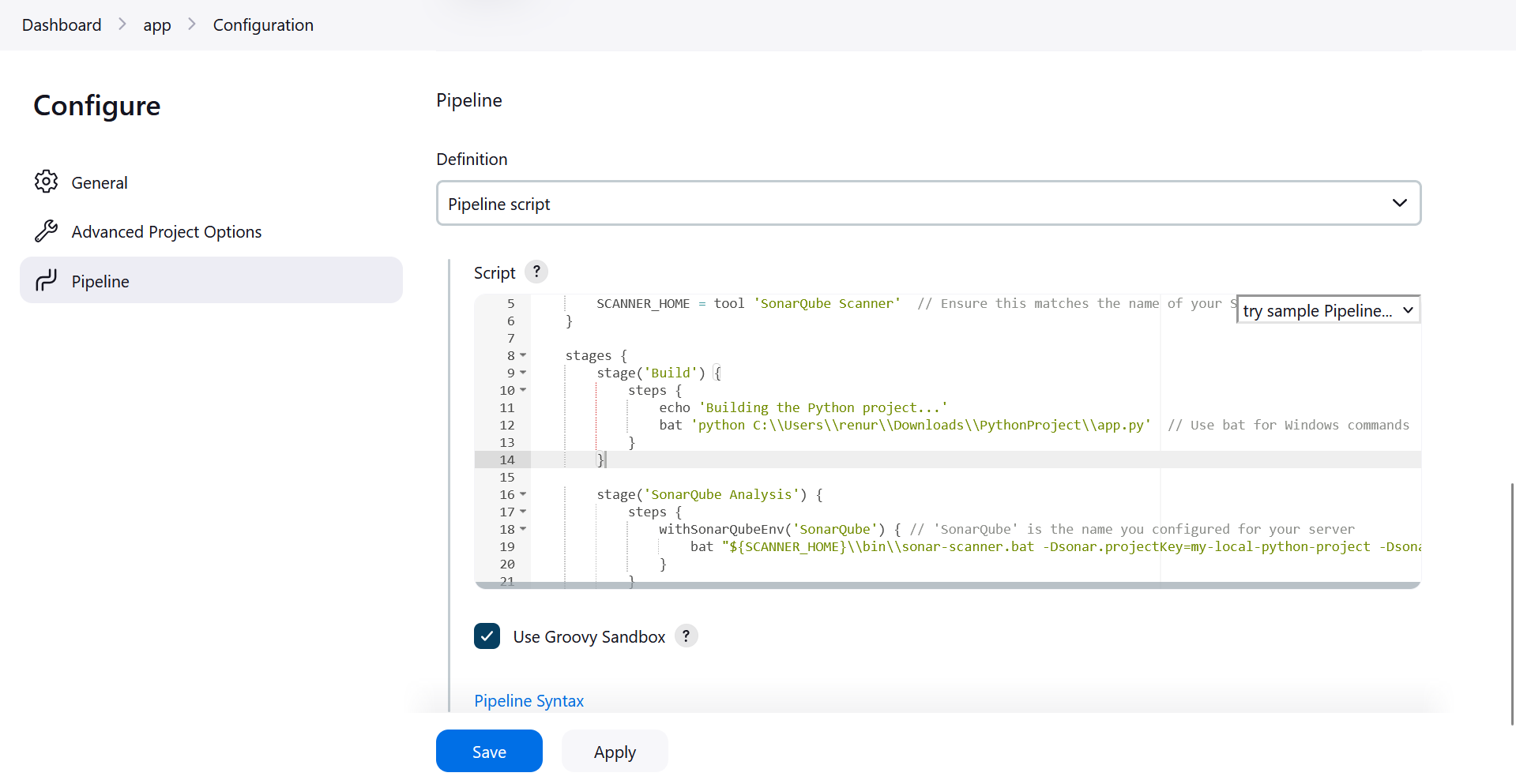
always {

echo 'Pipeline completed'

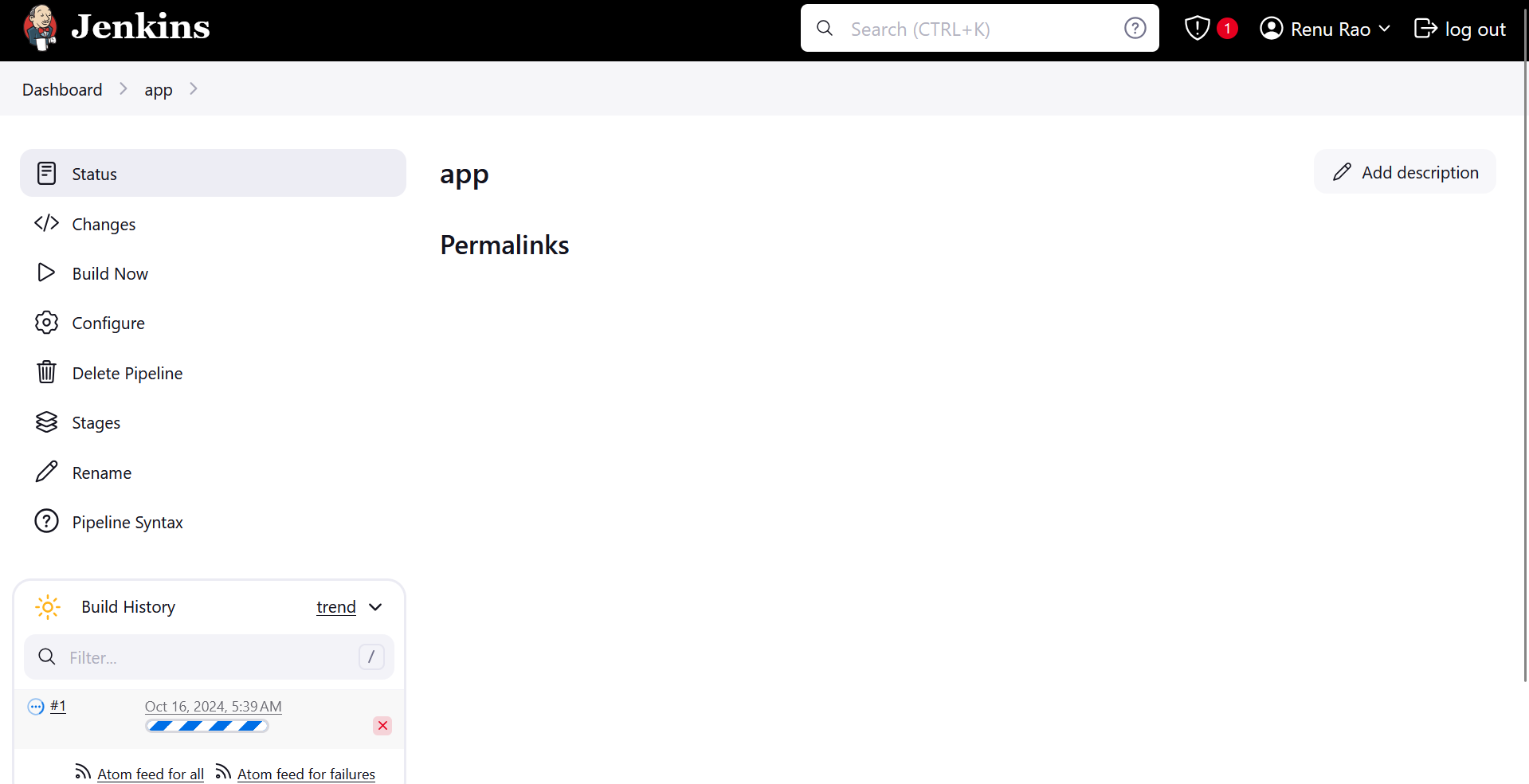
}

}

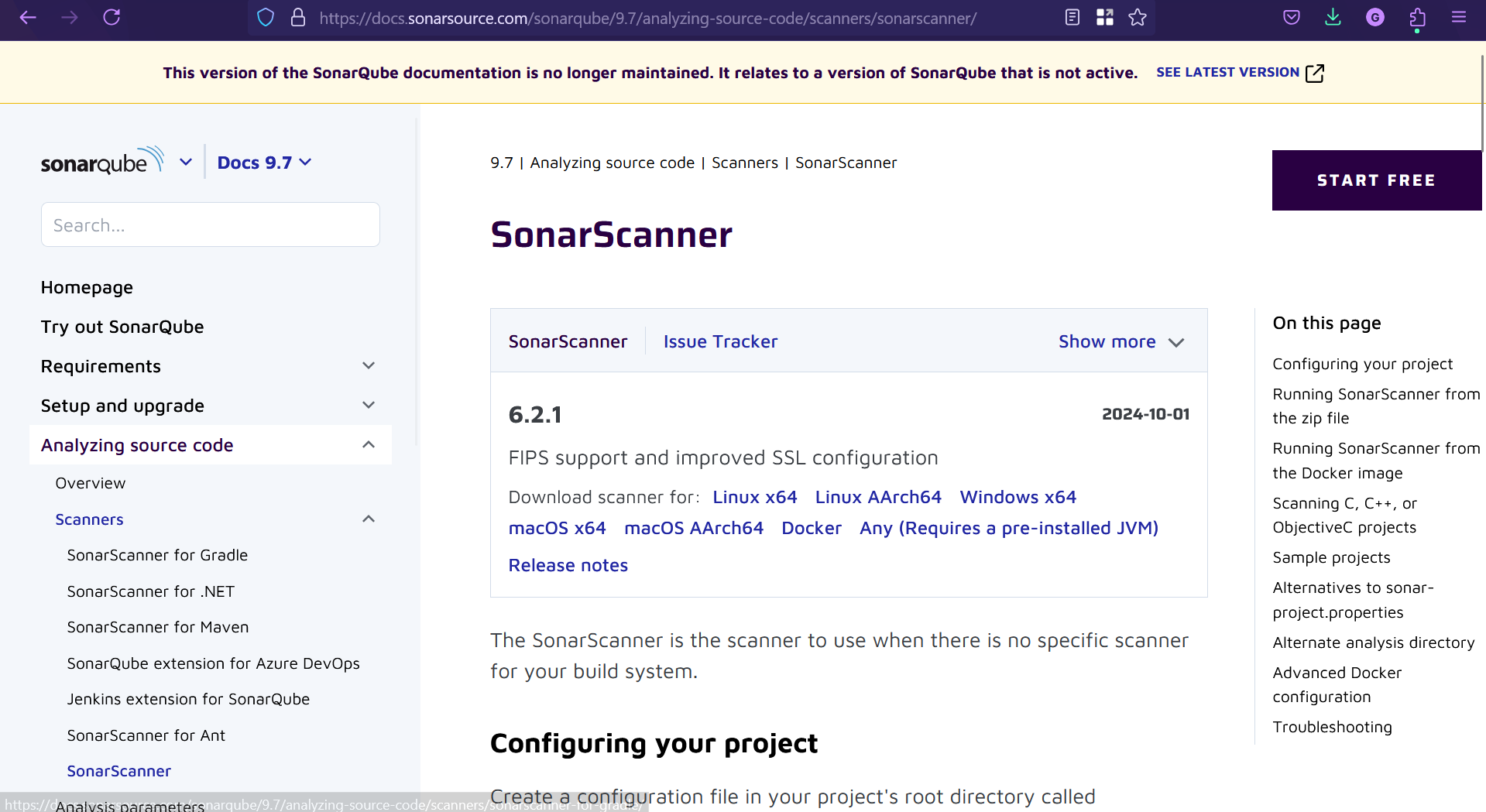
}



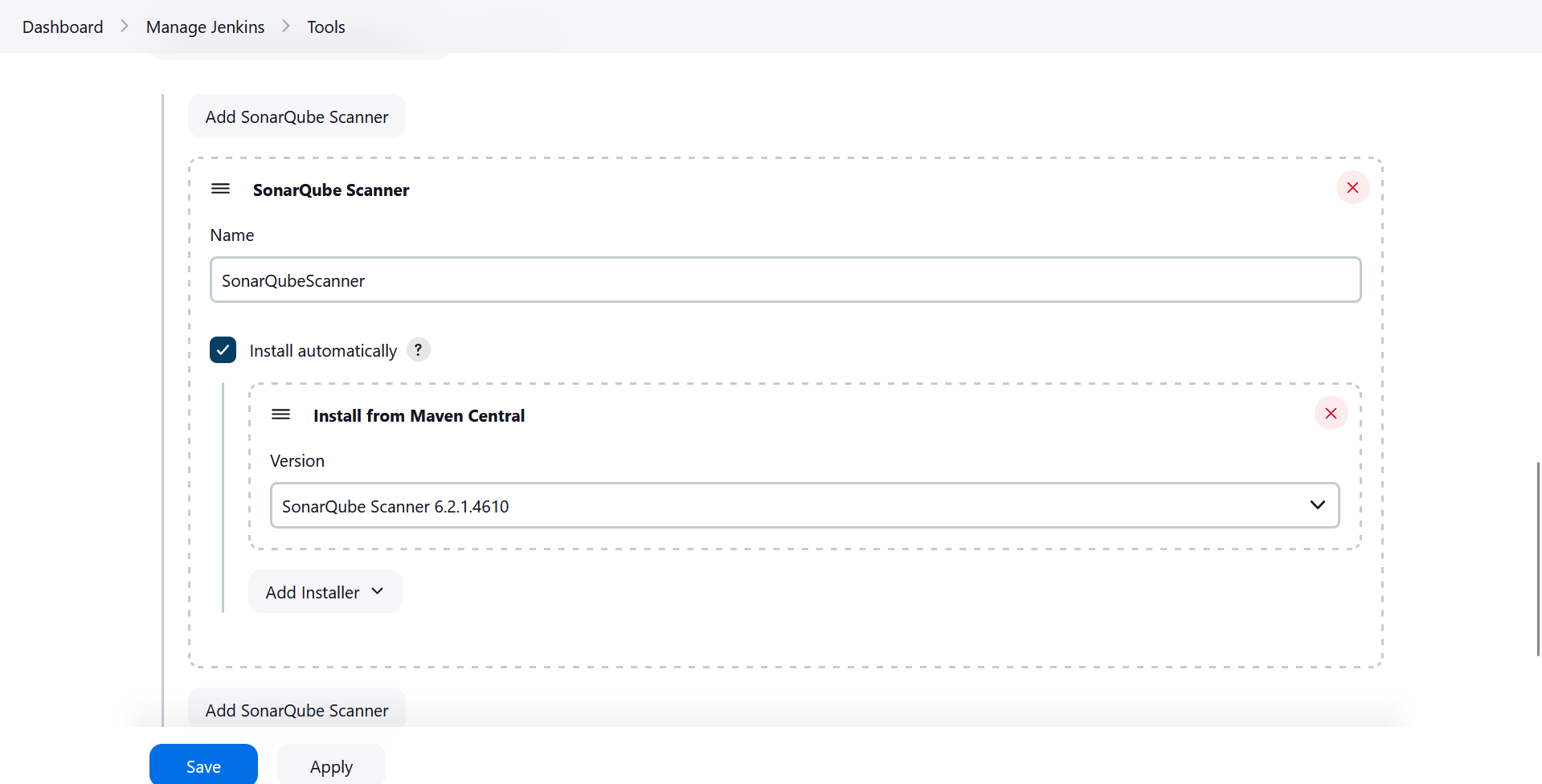
1. Click Apply and Save -> Build Now

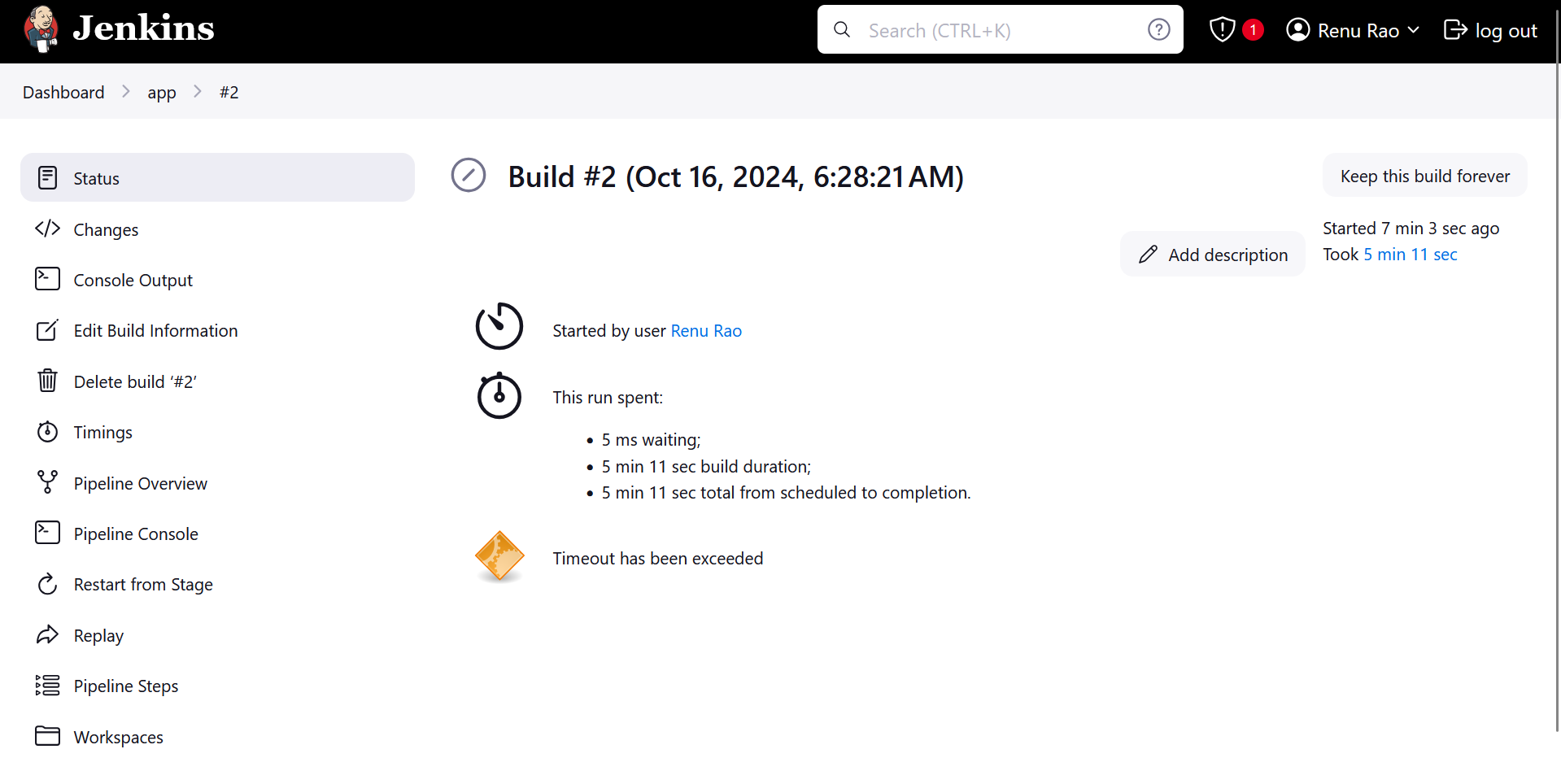


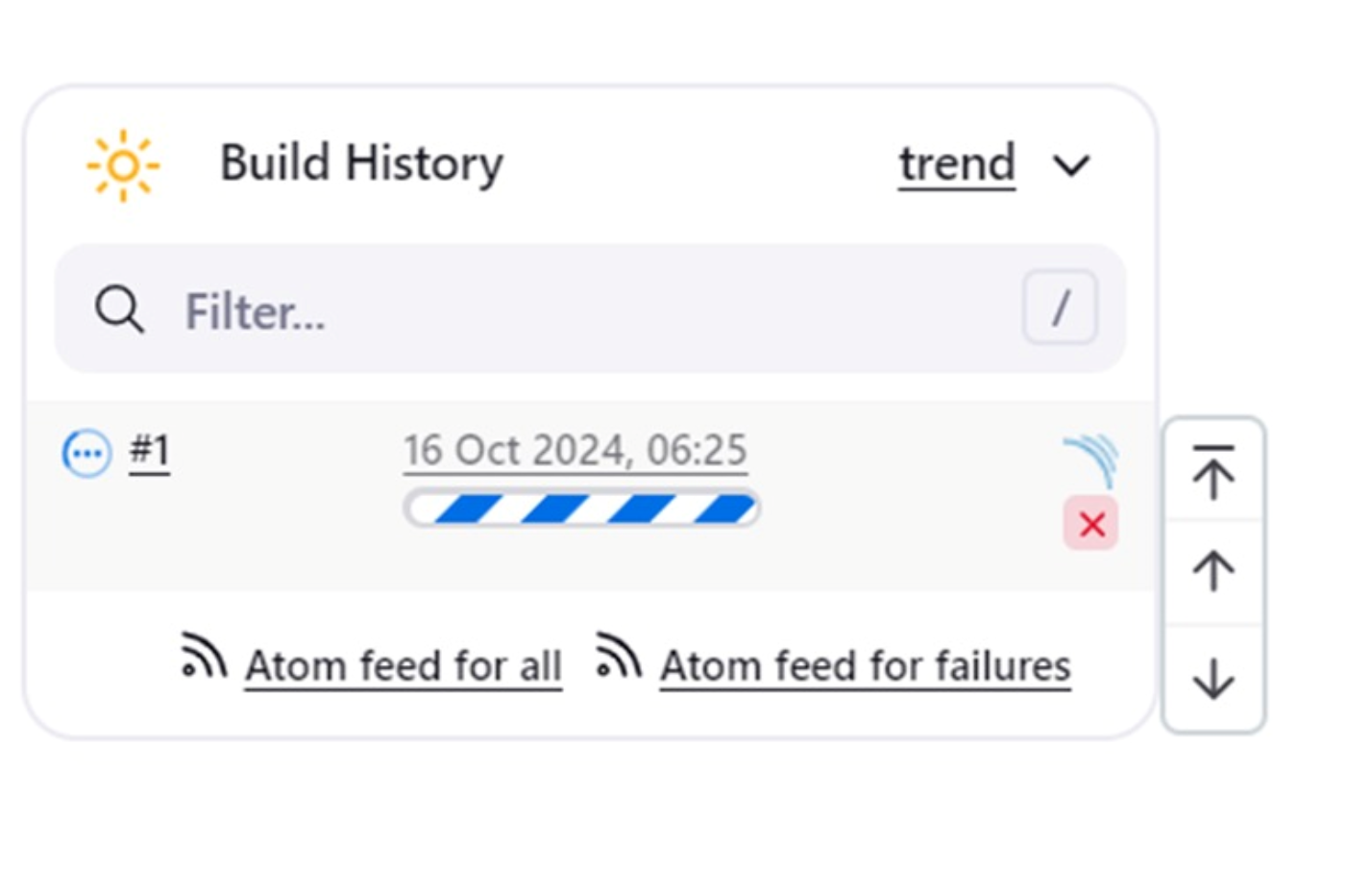
1. SonarQube Scanner

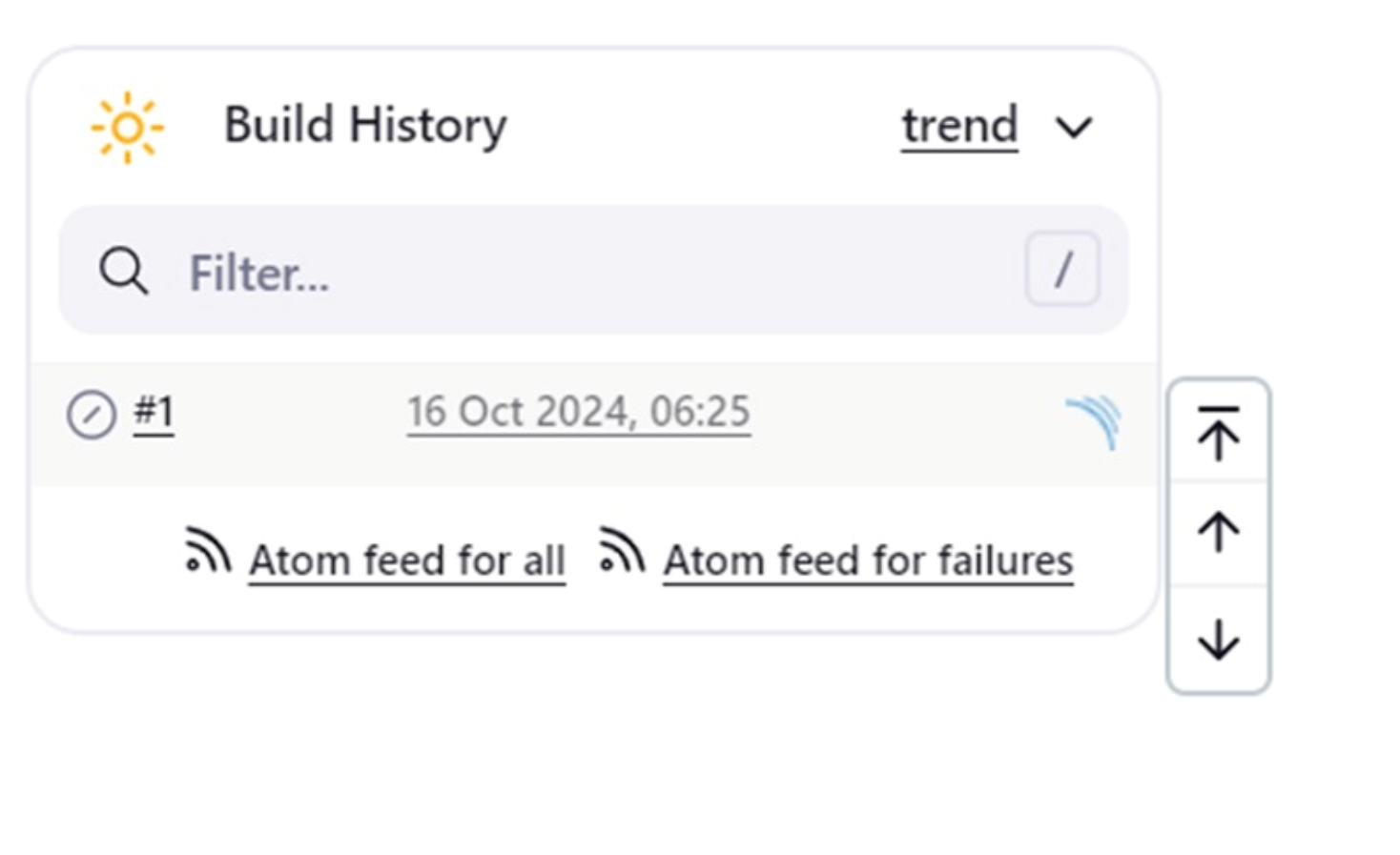


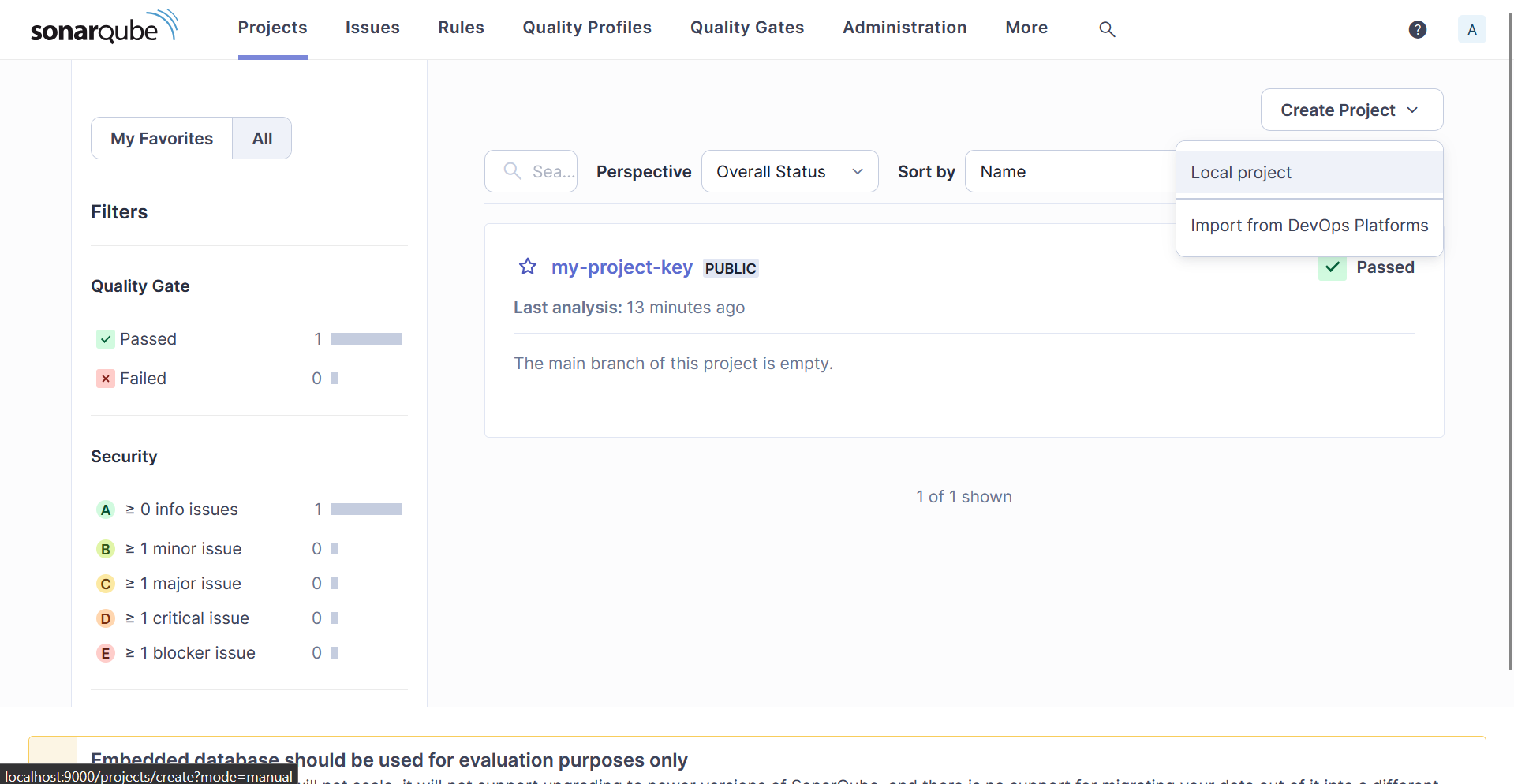
1. Go to Manage Jenkins -> Tools -> Add SonarQube Scanner -> Give name as SonarQube Scanner -> Check Install Auto -> Add Installer -> Select Extract .zip/.tar.gz -> Subdirectory of extracted archive ->

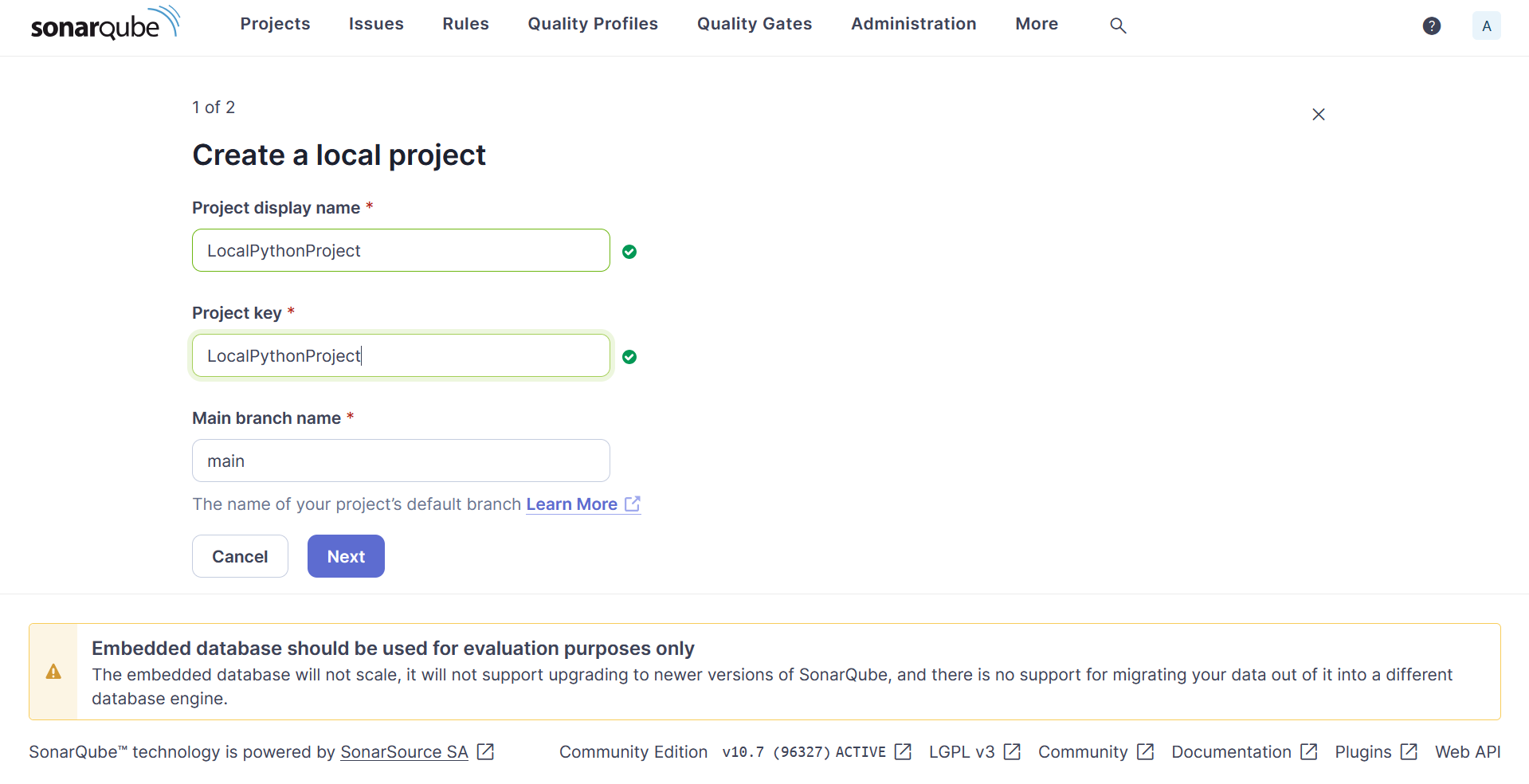


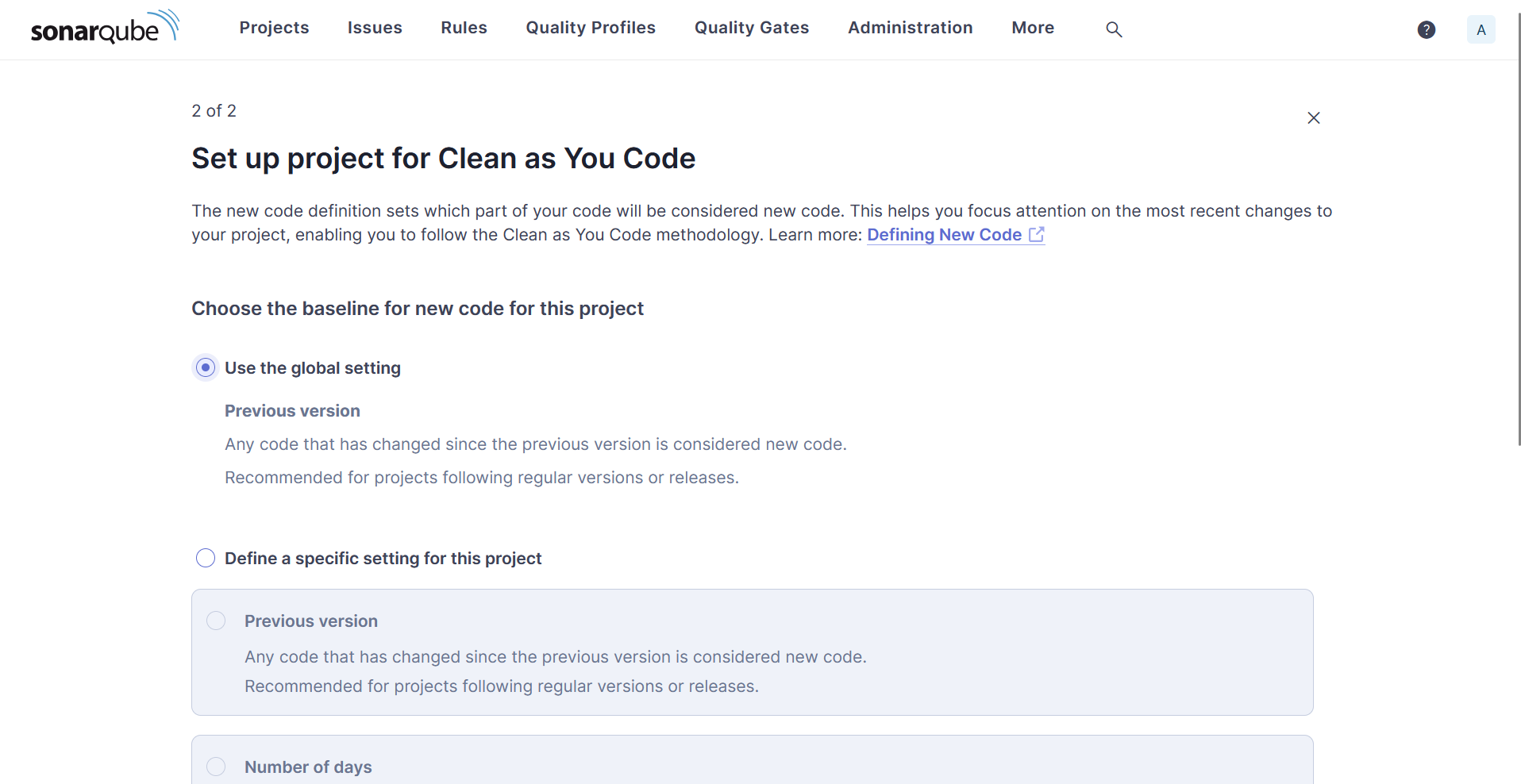


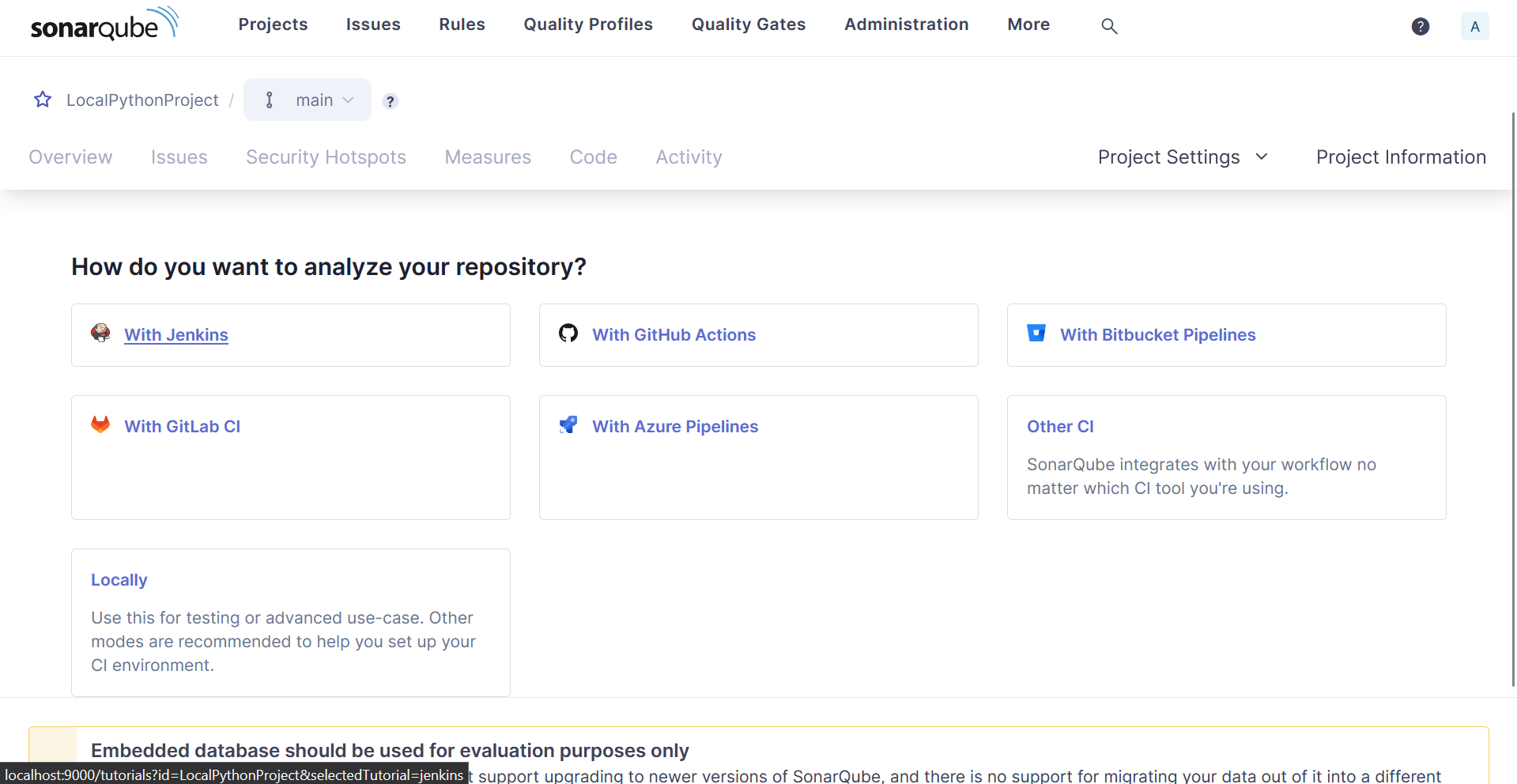


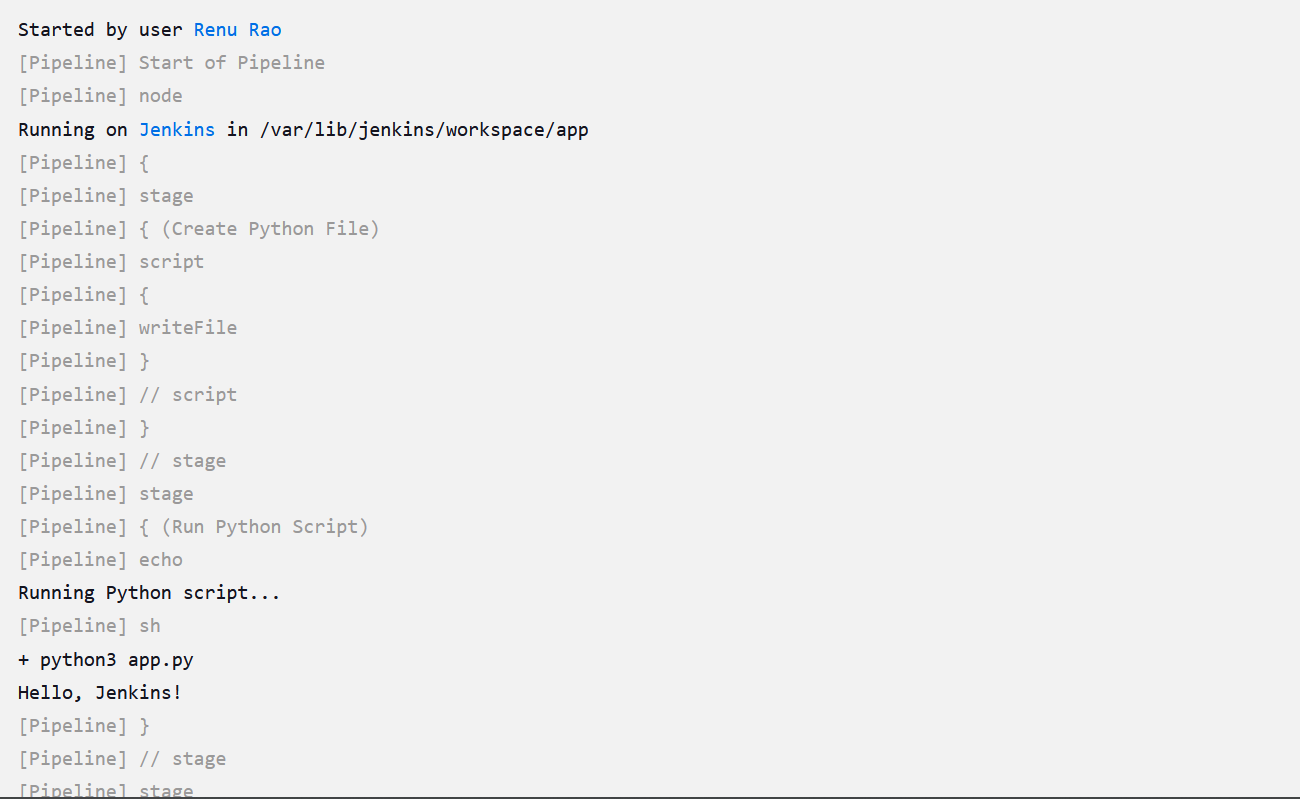




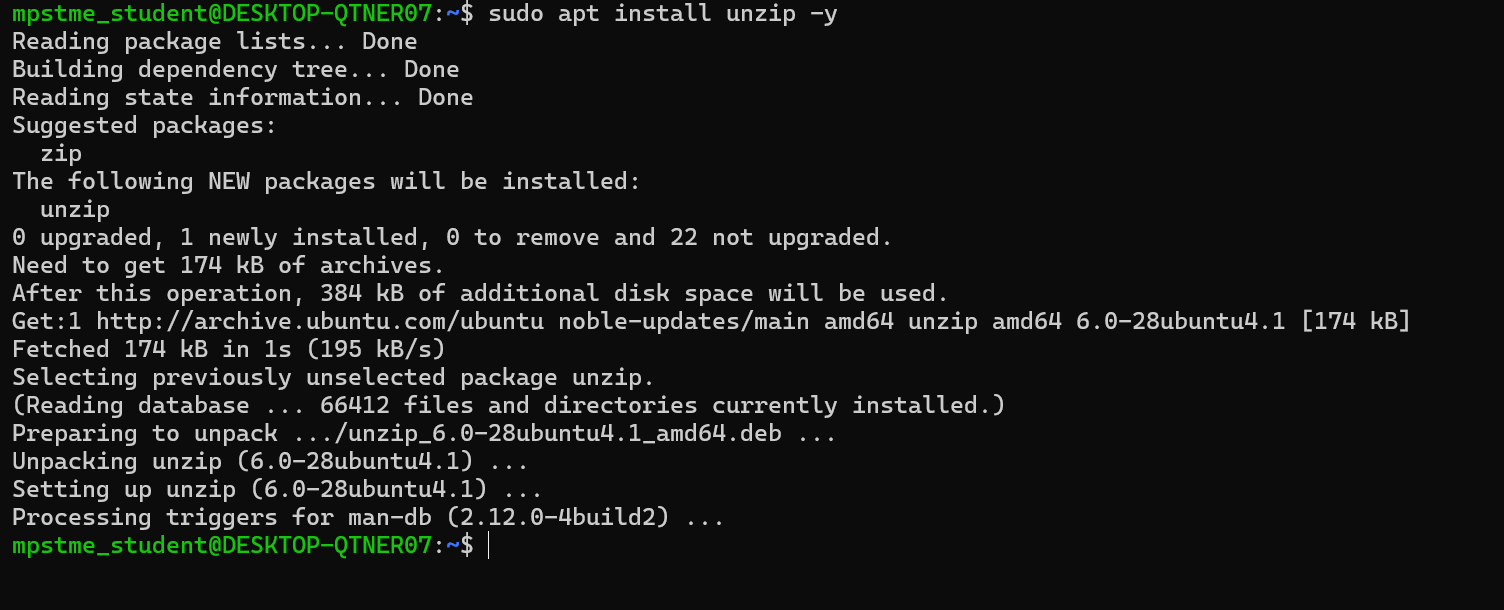


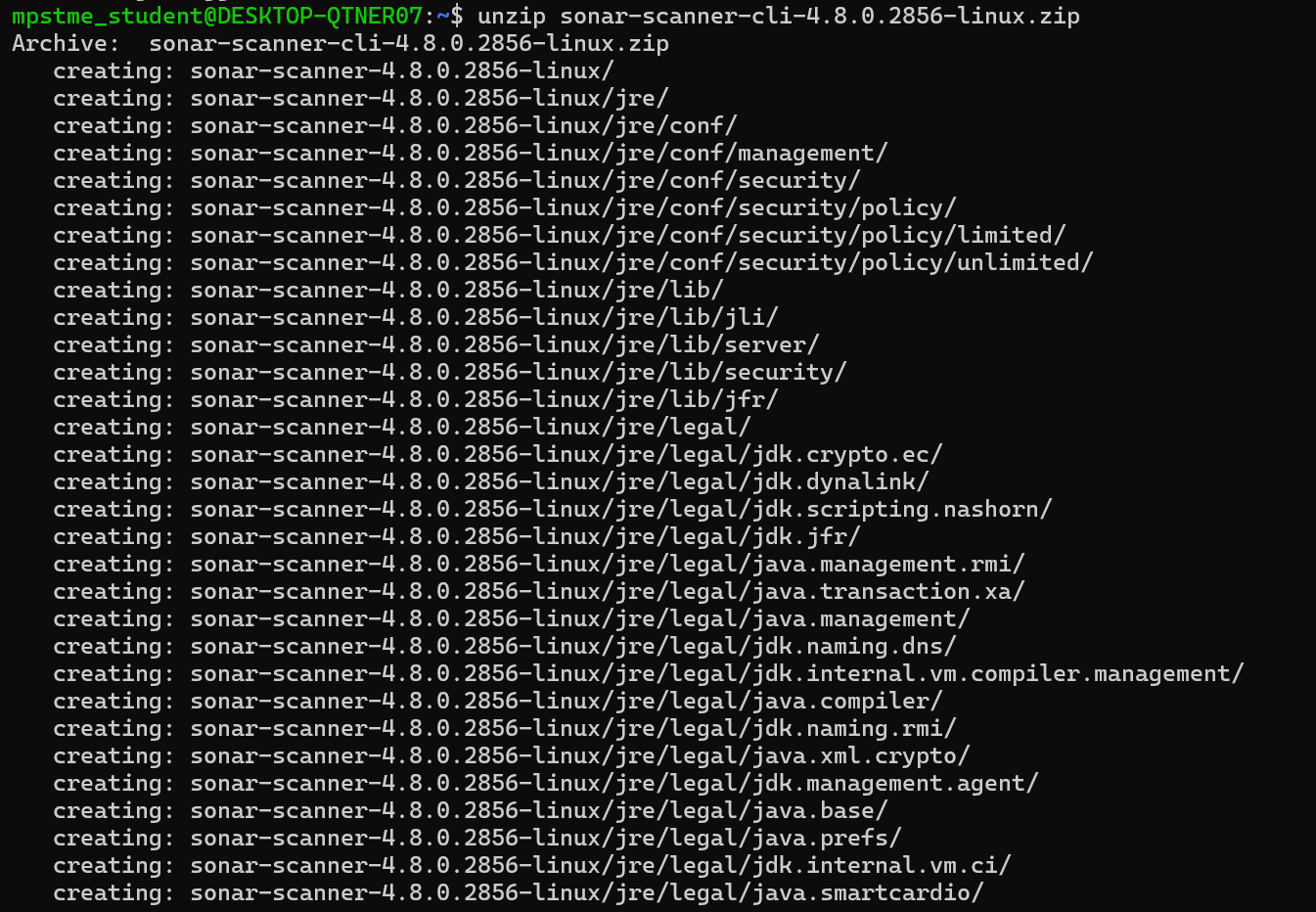


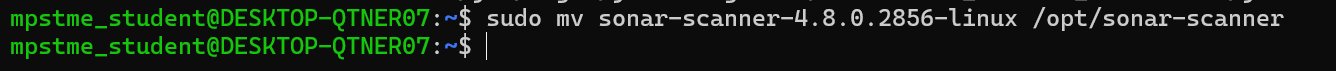


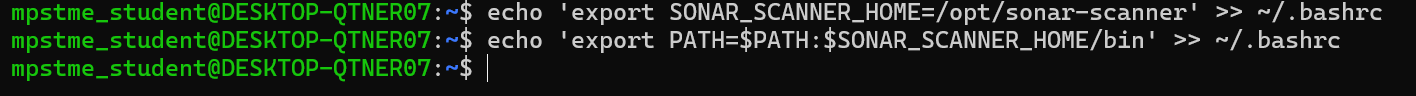


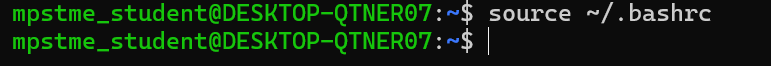


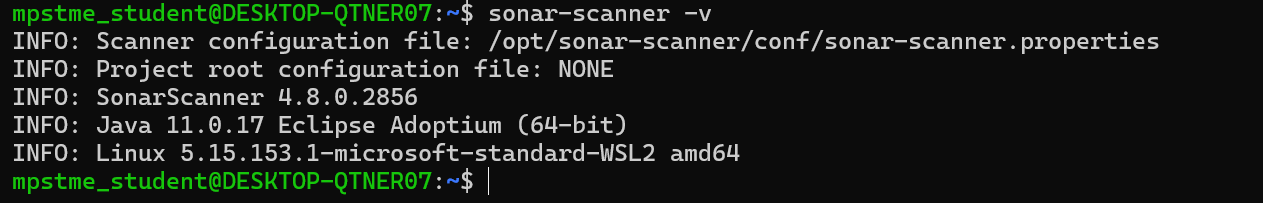


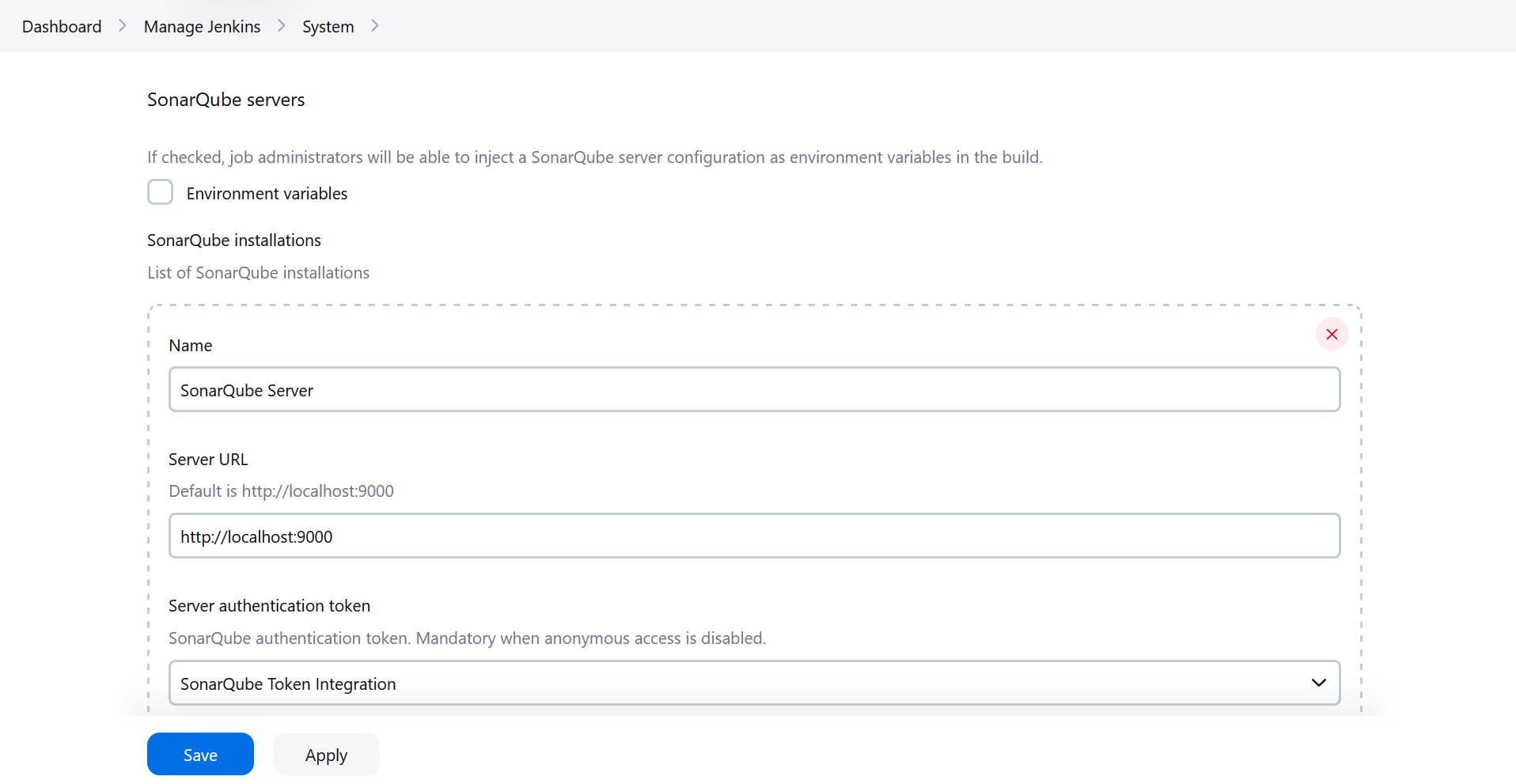




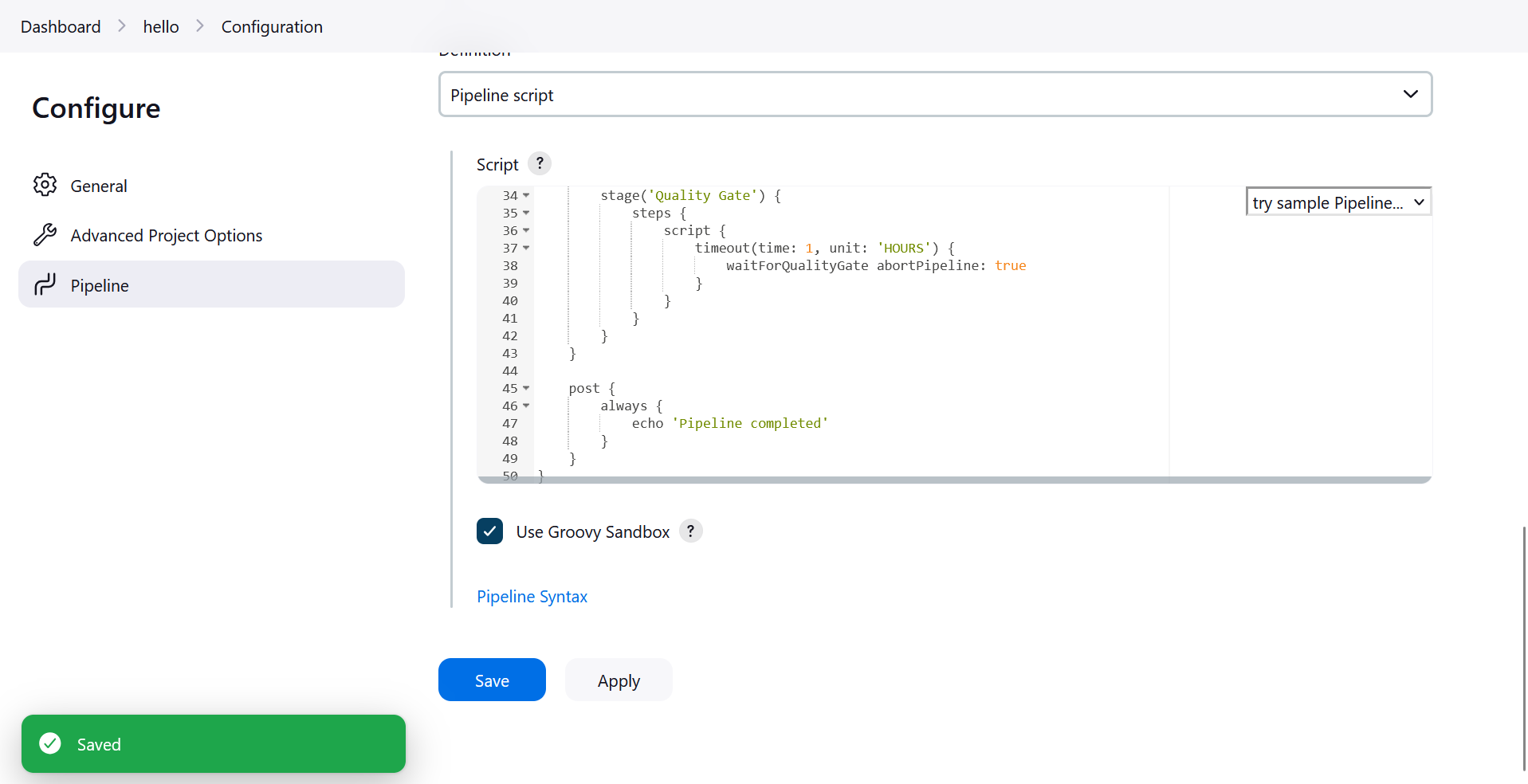












pipeline {

agent any

stages {

stage('Create Python File') {

steps {

script {

writeFile file: 'hello.py', text:

'''

def greet(name):

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

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

greet("Jenkins")

'''

}

}

}

stage('Run Python Script') {

steps {

echo 'Running Python script...'

sh 'python3 hello.py'

}

}

stage('SonarQube Analysis') {

steps {

withSonarQubeEnv('SonarQube Server') { // Use the name you provided in Jenkins

// Run the sonar-scanner using the absolute path

sh '/opt/sonar-scanner/bin/sonar-scanner -Dsonar.projectKey=LocalPythonProject -Dsonar.sources=.'

}

}

}

stage('Quality Gate') {

steps {

script {

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

waitForQualityGate abortPipeline: true

}

}

}

}

}

post {

always {

echo 'Pipeline completed'

}

}

}

Correct python indentation code