Sonar Setup

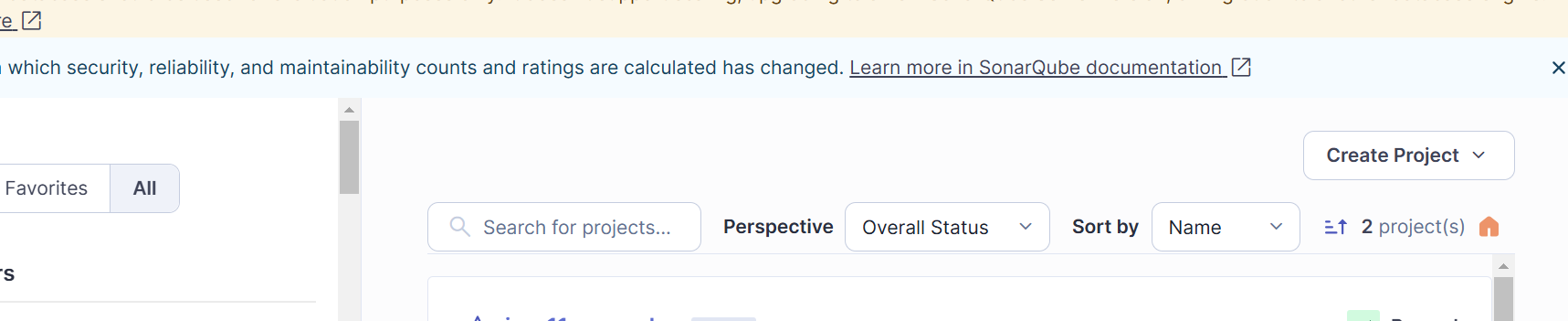
By default, the server running within the container will listen on port 9000. You can expose the container port 9000 to the host port 9000 with the -p 9000:9000 argument to docker run, like the command below:

docker run --name sonarqube-custom -p 9000:9000 sonarqube:community

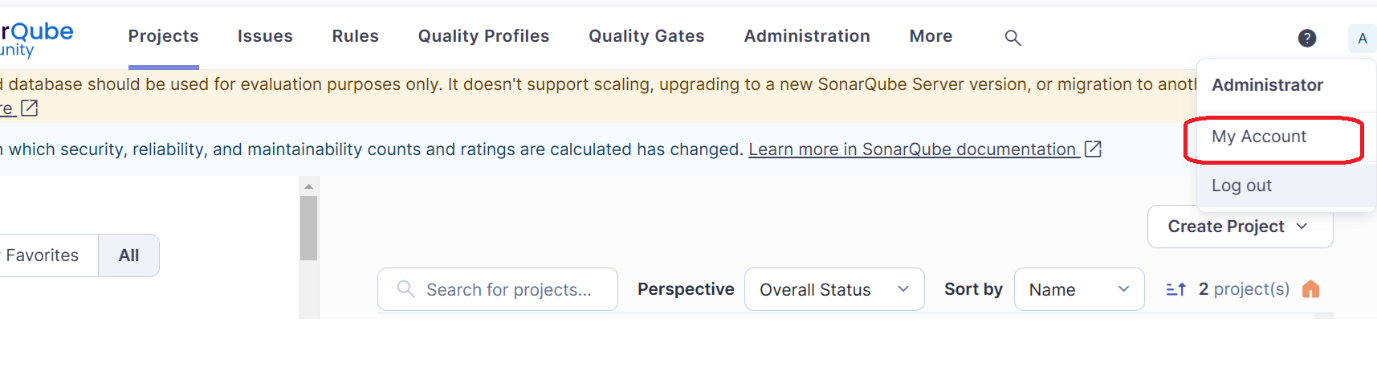
You can then browse to http://localhost:9000 or http://host-ip:9000 in your web browser to access the web interface.

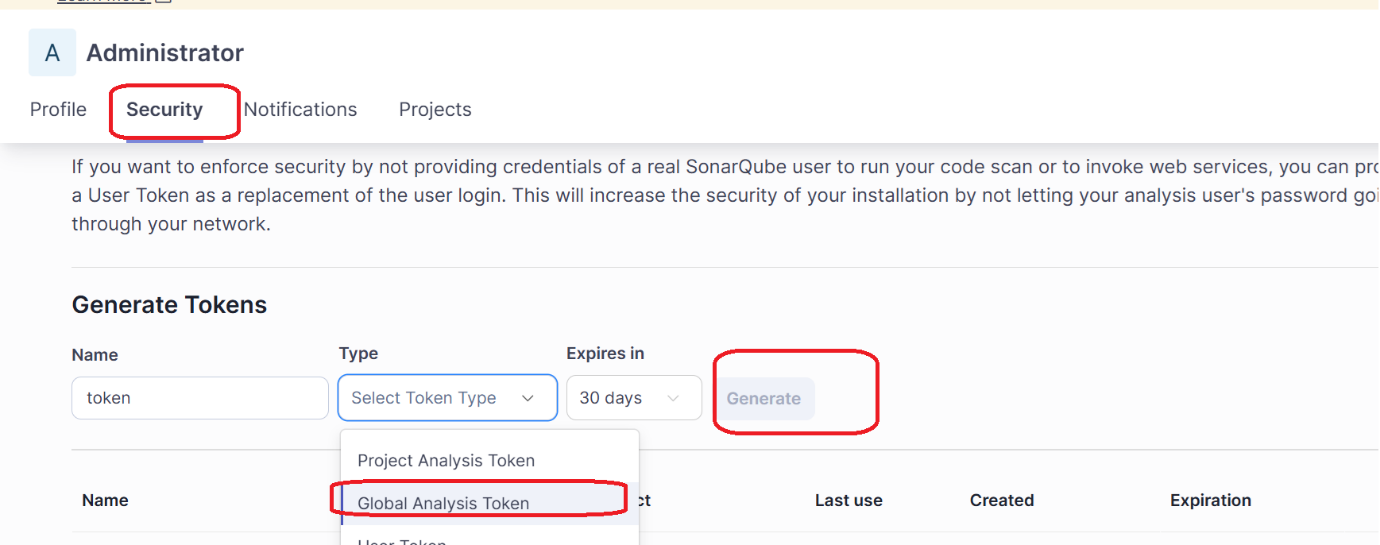
Login in the sonar portal

Create project

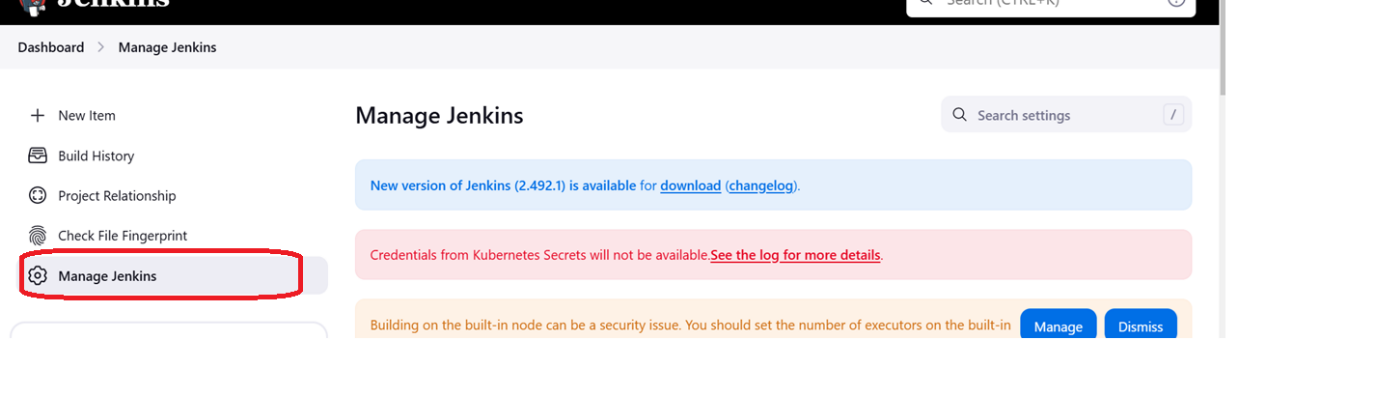


Create Token

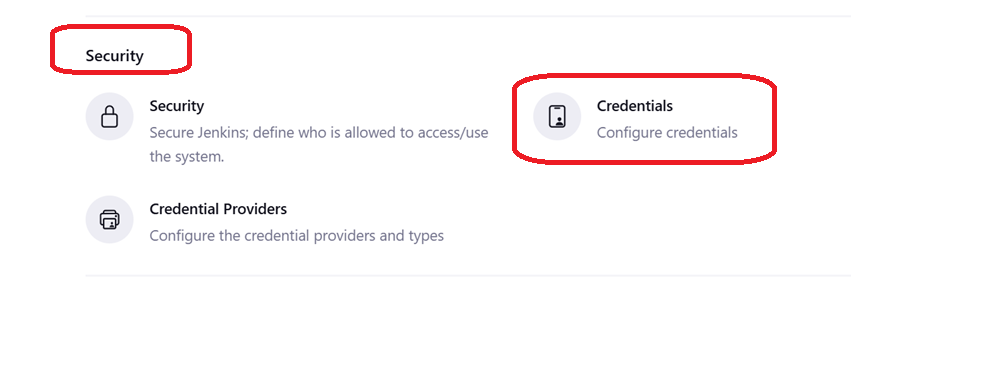




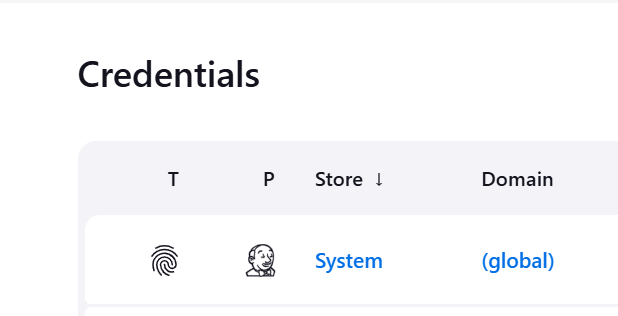
Go to Jenkins dashboard🡪 Manage Jenkins

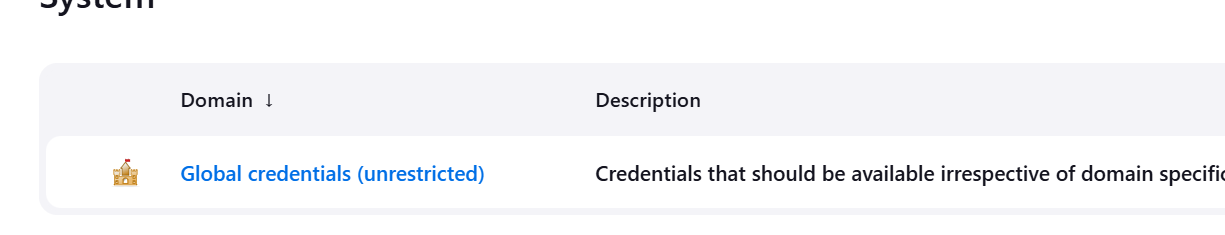


Security -> Credentials



Click on

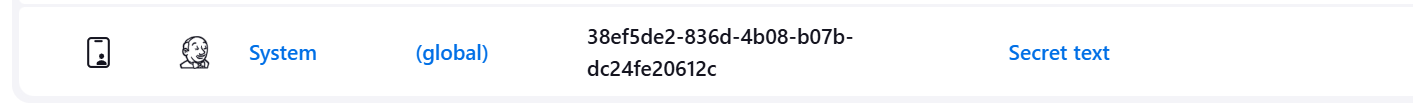


]

Secret text



Credentials id



Create pipeline job: Modify the field highlighted in red and run pipeline

pipeline {

agent any

environment {

SONARQUBE\_URL = 'http://192.168.111.130:9000' /\* Adjust SonarQube server URL if needed \*/

SONARQUBE\_PROJECT\_KEY = 'myproject'

}

stages {

stage('Checkout SCM') {

steps {

script {

checkoutSCM()

}

}

}

stage('Compiling and Running Test Cases') {

steps {

script {

compileAndRunTests()

}

}

}

stage('Generating Cucumber Reports') {

steps {

script {

generateCucumberReports()

}

}

}

stage('Creating Package') {

steps {

script {

createPackage()

}

}

}

stage('Adding Generate Report') {

steps {

script {

addGenerateReport()

}

}

}

stage('Analyzing Code Quality') {

steps {

script {

analyzeCodeQuality()

}

}

}

}

}

def checkoutSCM() {

checkout scm: [

$class: 'GitSCM',

branches: [[name: '\*/master']],

userRemoteConfigs: [[url: 'https://github.com/hellokaton/java11-examples.git']]

]

}

def compileAndRunTests() {

sh 'mvn clean'

sh 'mvn compile'

sh 'mvn test'

}

def generateCucumberReports() {

sh 'mvn verify'

}

def createPackage() {

sh 'mvn package'

}

def addGenerateReport() {

sh 'mvn verify'

}

def analyzeCodeQuality() {

withCredentials([string(credentialsId: '38ef5de2-836d-4b08-b07b-dc24fe20612c', variable: 'SONAR\_TOKEN')]) {

sh '''

mvn sonar:sonar \

-Dsonar.projectKey="$SONARQUBE\_PROJECT\_KEY" \

-Dsonar.host.url="$SONARQUBE\_URL" \

-Dsonar.login="$SONAR\_TOKEN"

'''

}

}