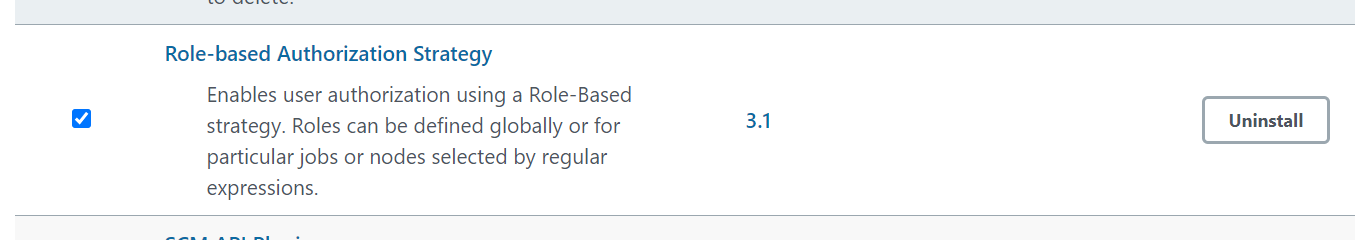
**Try to implement an end-to-end  cicd pipeline with the below requirement**

The build approval should be done by a user called build approver and not by user admin. For this you have to implement role based authorization in Jenkins. Once you install this plugin you will get manage and assign roles option in manage Jenkins. In that you will have global and project roles there. In pipeline script,

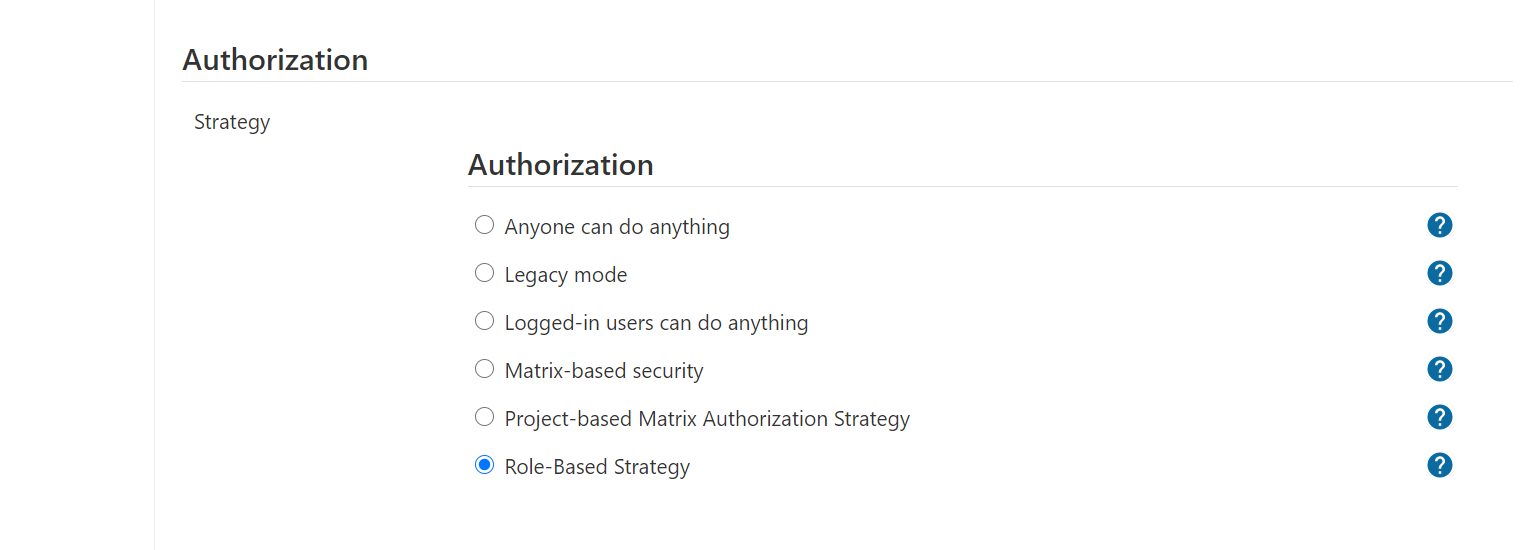
**slackSend baseUrl: 'https://hooks.slack.com/services/', channel: 'cicd1', color: 'bad', message: "${​​​​​​​​env.BUILD\_URL}​​​​​​​​", tokenCredentialId: 'slackid1', username: 'Buildapprover'**

**For that we need to install one plugin as a name of Role Base Authentication plugin as shown below.**



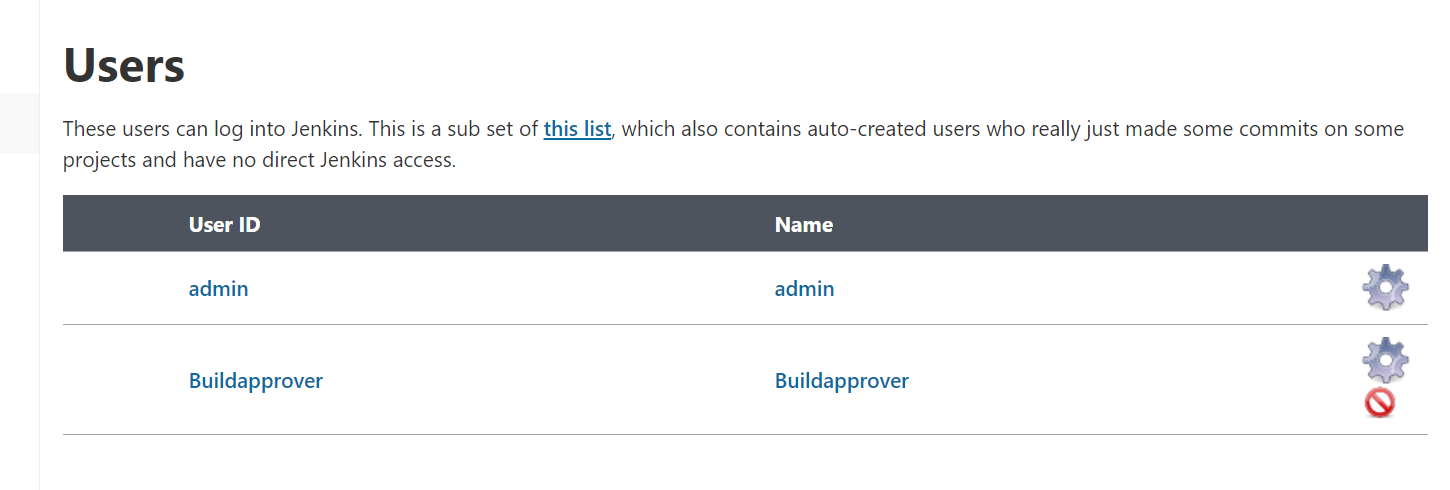
Then we need to change some settings in configure global security in manage Jenkins.

In that Authorization we need to choose the Role-Base Strategy as shown below.



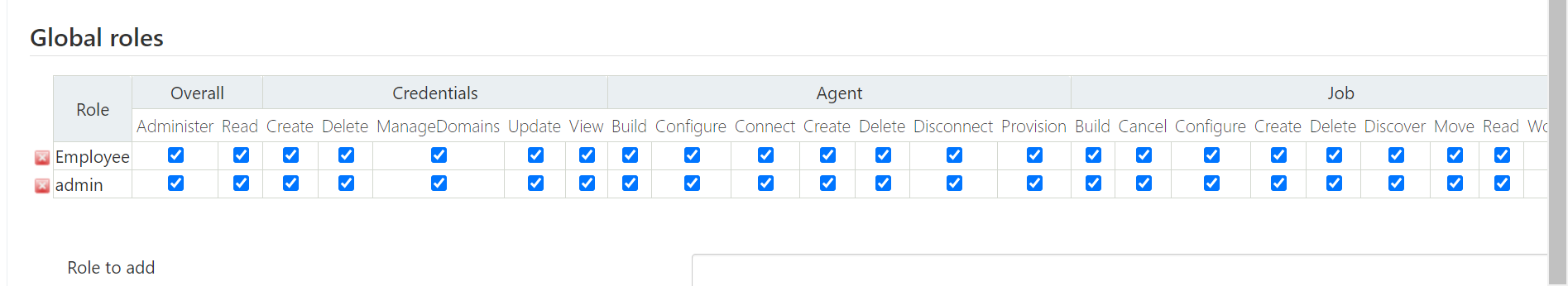
Then save the changes.

Now we need to add a user as a name of Buildapprover in Manage Users . now click on create user as shown below.



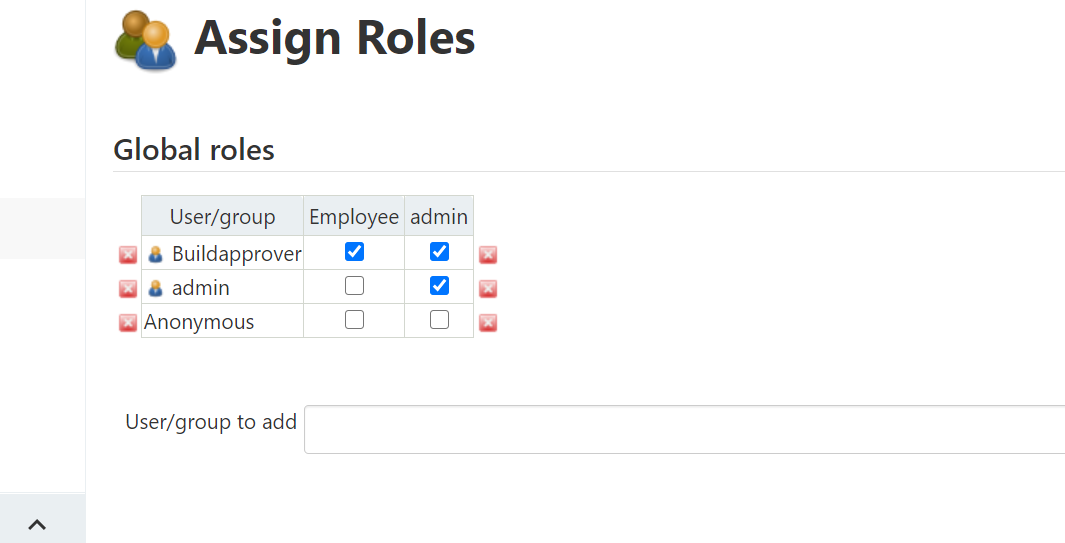
Now our goal is assign our rolles into that project in the path sshown below.

Manage Jenkins🡪Manage and Assign Roles here you can click on Manage Roles .

Here I am adding a user name of Employee and I am allowing all settings as shown below. 

Click on save amd apply

Now you can assign role by selecting Assiugn Role in that page.



Now we can run all the required containers and start the Mysql\_Server.

Here I am creating one pipeline job as a name of CICD\_PIPEINE\_BUILDAPPROVER for parking\_backend

And write pipeline script as s hown below.

pipeline {

agent any

stages {

stage('SCM Checkout') {

steps {

git 'https://github.com/Gkasiraju/parking\_backend.git'

}

}

stage('Build Approval'){

steps{

slackSend baseUrl: 'https://hooks.slack.com/services/', channel: 'jenkins', color: 'bad', message: "${env.BUILD\_URL}", tokenCredentialId: 'slackid1', username: 'Buildapprover'

script{

def userInput

try {

userInput = input(

id: 'Proceed1', message: 'Approval',submitterParameter: 'submitter', submitter:'admin', parameters: [

[$class: 'BooleanParameterDefinition', defaultValue: true, description: '', name: 'Please Provide Build Approval']])

}

catch(err) {

def user = err.getCauses()[0].getUser()

userInput = false

echo "Aborted by: [${user}]"

}}

}

}

stage ('Publish Unit Test Report')

{ steps {

sh '/opt/maven/bin/mvn verify -Dmaven.test.skip=true'

junit allowEmptyResults: true, testResults: '\*\*/target/surefire-reports/\*.xml'

}

}

stage ('Build && SonarQube analysis') {

steps {

withSonarQubeEnv('sonar'){

sh '/opt/maven/bin/mvn clean package sonar:sonar -Dsonar.password=admin123 -Dsonar.login=admin -Dmaven.test.skip=true'

}

}

}

stage('Quality Gate'){

steps {

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

waitForQualityGate abortPipeline: true

}

}

}

stage ('Artifact-Deployment') {

steps {

sh '/opt/maven/bin/mvn clean deploy -Dmaven.test.skip=true'

}

}

stage ('Release') {

steps {

sh 'export JENKINS\_NODE\_COOKIE=dontkillme ;nohup java -jar $WORKSPACE/target/\*.jar &'

}

}

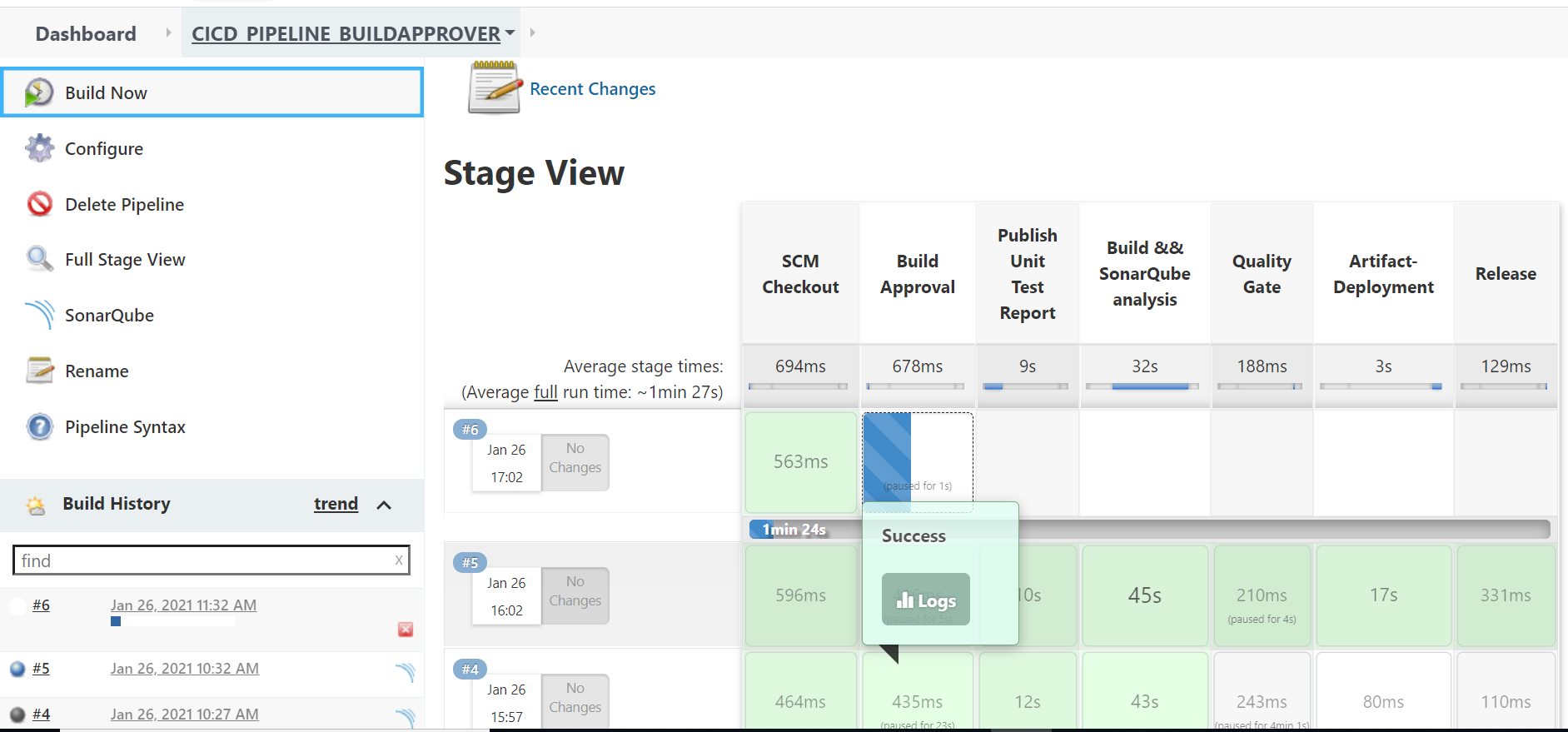
}

}

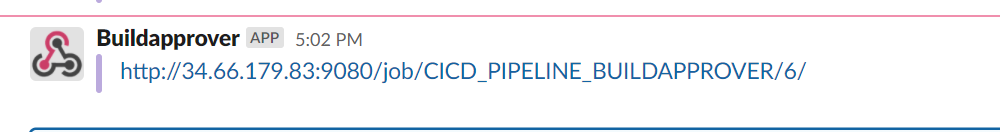
Click on Save and apply.

Here In the build approval stage we are allowing one Buildapprover to approve the build rather than the admin so that’s why we are creating the new user in that Jenkins.

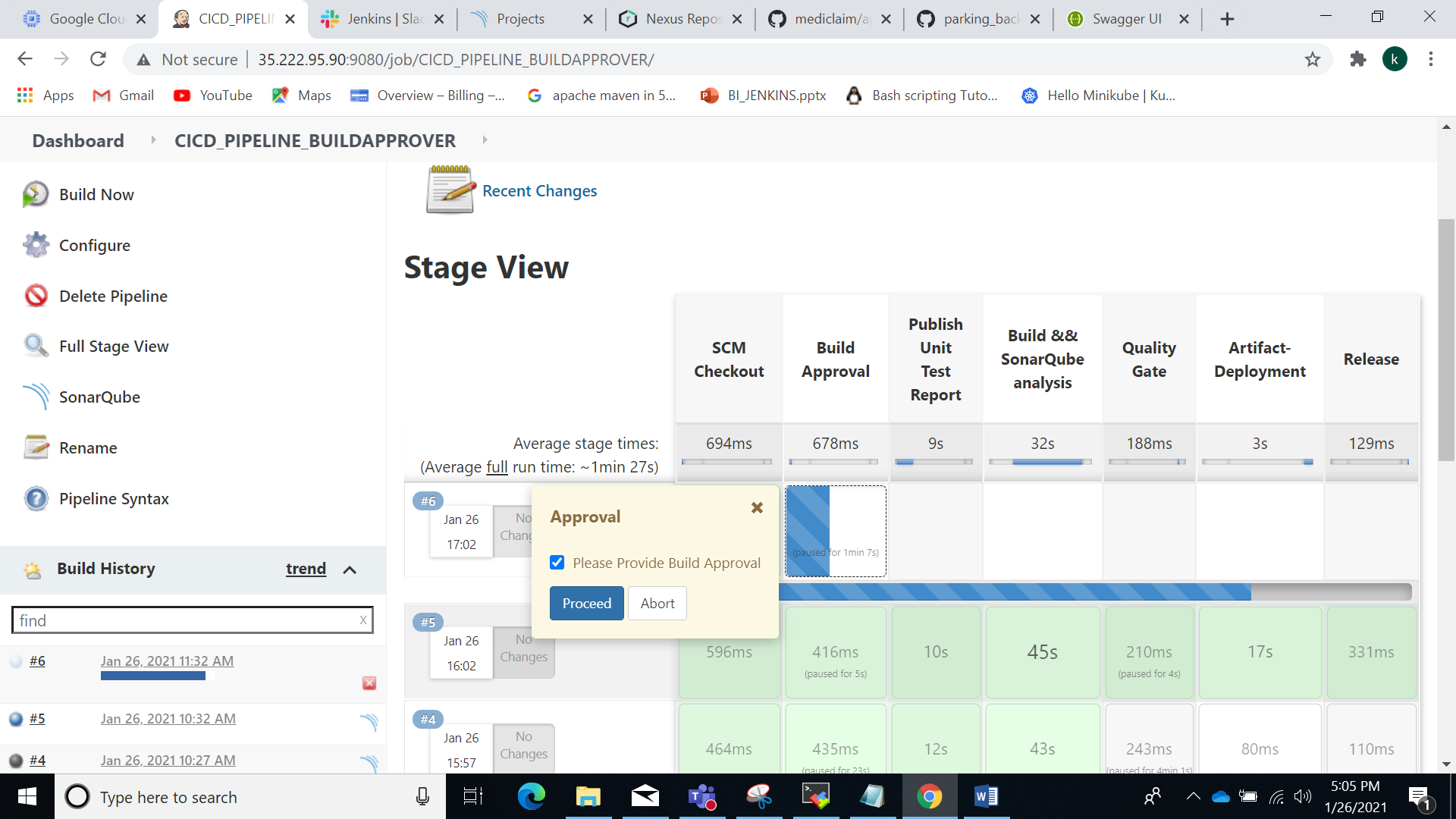
Now currently we build that jon in admin we can build approver by Buildapprover login.

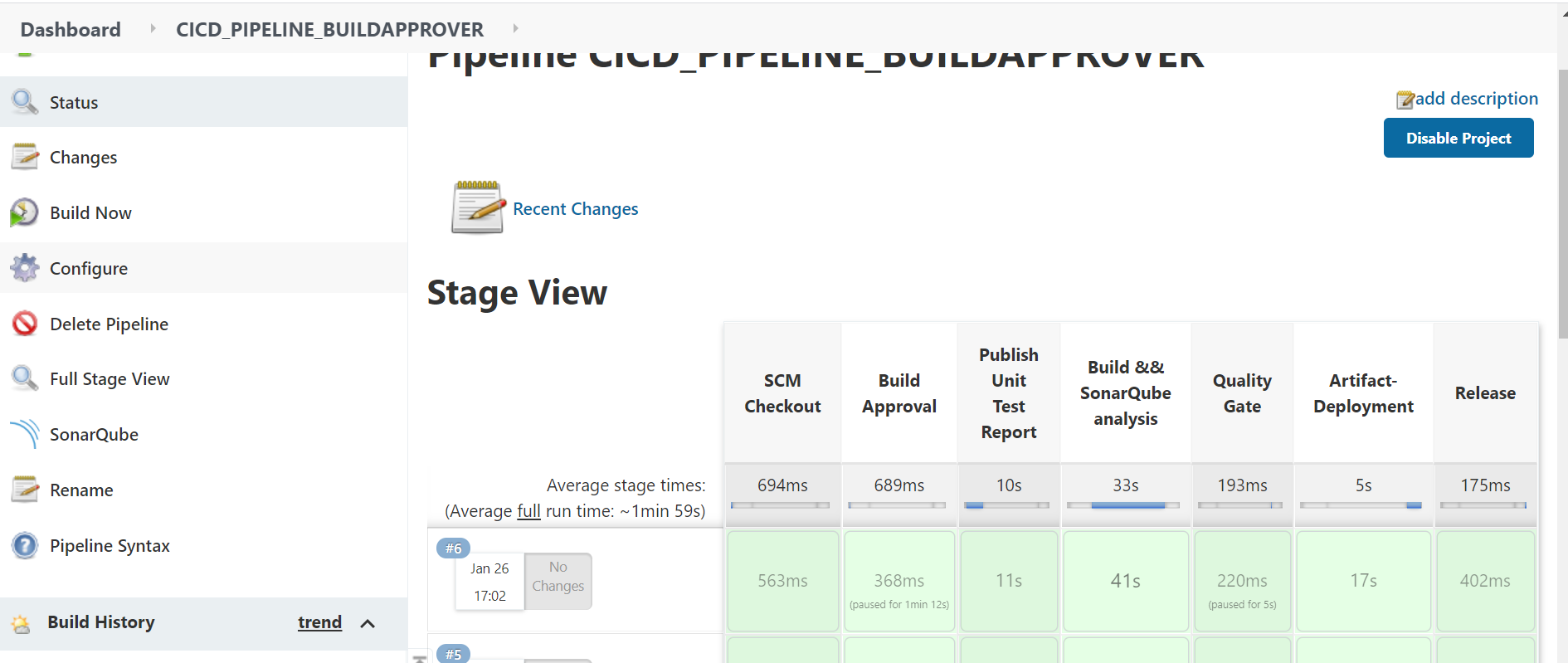


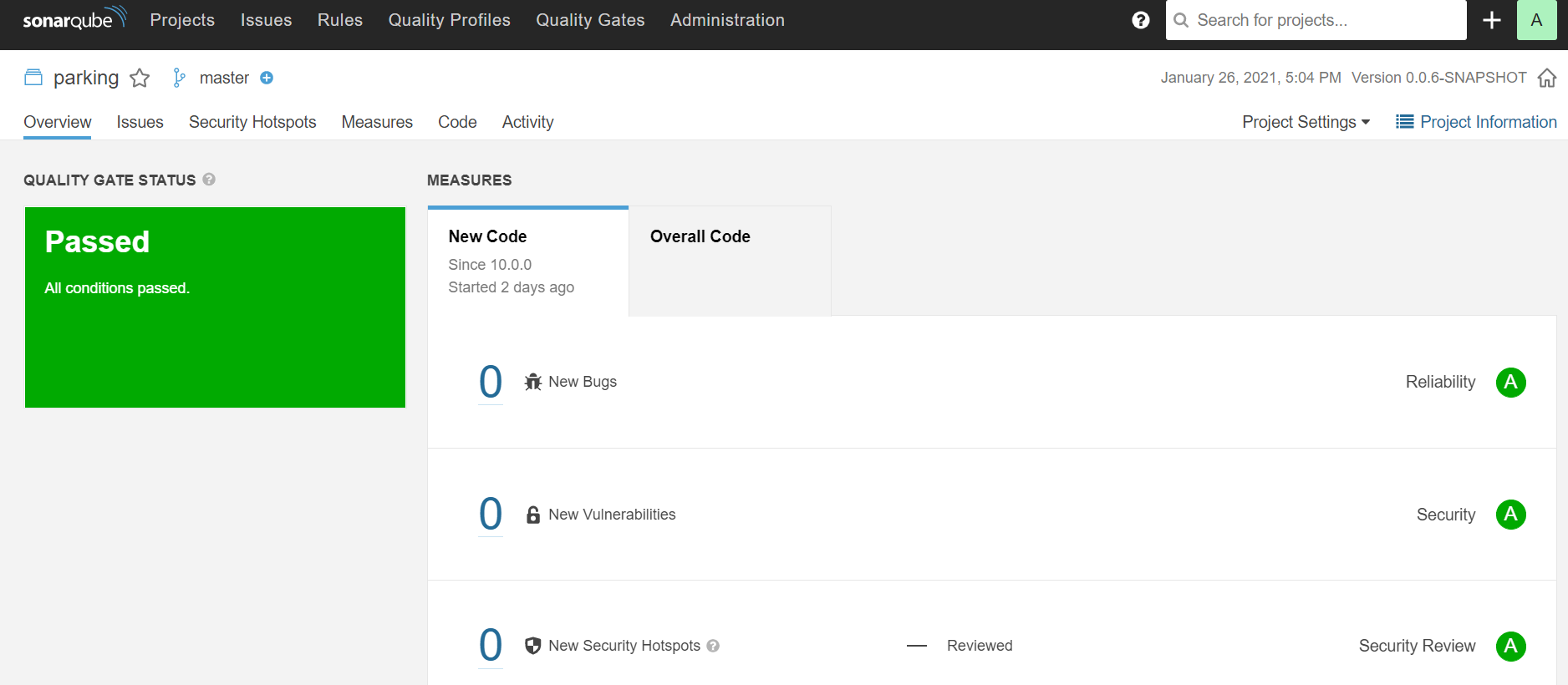
Notification on slack.

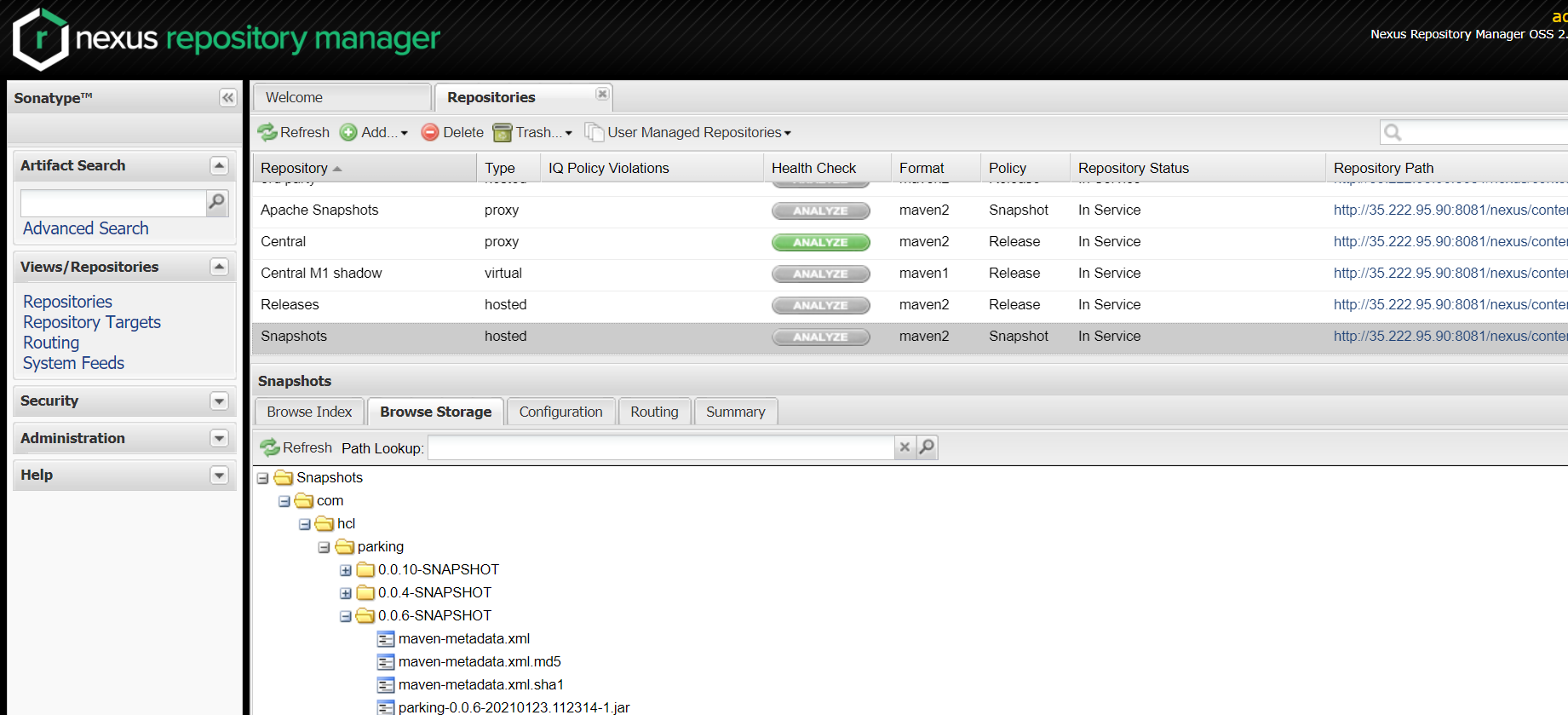


Here I am login as Buildapprover for approve that build as shown below









Finally we can see the swagger page: http://35.222.95.90:9096/parking/swagger-ui.html

