Table of Contents

[Gotoro: Guideline to connect to Jenkins 1](#_Toc118790854)

[Gotoro: Guideline to Add CICD pipeline to Deploy Backend -Dev 1](#_Toc118790855)

[Set up trigger when create Pull Request on Oracle Repo 1](#_Toc118790856)

# Gotoro: Guideline to connect to Jenkins

Step 1: Create file private key name: **dev-user.pem**.

Details key:

-----BEGIN RSA PRIVATE KEY-----

MIIEpAIBAAKCAQEA5cavfLzUqu+eUg5j/nl+hDu9fiGFvI//TX6qpuPe+Njb6N2Z

HxYKN2UYXlpCHay2E5T58laZ/n8JZAZMMXnXD7NpktDNMf9OrDP1dMLeL0y2ByIQ

8DpJOGyPH2nmiGANXhusEPeFSfDMkjZn995jqJnyGrzOnLeunUnkWYpkDdV22nHT

bSb+xiH8kuFNTwQtiKSBDrN+lcThO6sqQz3+DPvK3ZYrb4wD9c1U404MZmgtJxRG

bDjwiuaeBBZg0zoKFgqvWlBrd+jqq3O8z4IqA6UAu8aN+fPNYOWwrtdgVPzHKF2e

M/D8WxZpeQUCv9JKs3i3TwUaXL9vooD/KzjJEQIDAQABAoIBAH50znez5Mb+gP9f

9CBE9SAJUMxt3pBcH4GiyIB4wGnrz3tlGbRGCPFbrgGfYuu59arpPnXku3OYrq5h

2OURzQ5TCzNFvuhZPg7/uYK9DFKzwSpD0VakfU+UT2BWhA1h73lveU8NKkn7QFrn

nmiH+Aj+UIPujTcqz+q4k/fDdjU1zvH/fVIxa3yxP56fm3SvA0RGU6fsqZKBqre4

5t+gYbv6iGW8T2P0Ap9AYr10b39w+5ahW/L/AnokSRAc80rcgBfAlVMBFnGKMq/b

hVCbpp9MpLtnIQNFRk6f8BsnAG8gB5AURaa0yEzRDSZ+5HWPxEgyPng5AnmPBYKG

cYFCHQECgYEA8qaC9iOraJEyWDuVmLBI/dC2dQKcBFrZZxi+dpJJcRtHAjls4hAU

8vng6l+p53jUMvacjFIF/Vlu/uetFdapb7gYINm7dAe+a6BUIE6yzQNqv2QGWUwv

/EdZ2tG22cYKQCI/peNSppJS2E9+RHyVkzDqhMFgzABrj8k5p/UmJ/kCgYEA8mra

HZSOmL84+SUxAOAu0pLBRnWlgY+WLrExloe1bZkf/Bylmh1Rmb91RikxoyeMdetU

TihuImYkGfSJSyiD63MJfSFJAsacimKbukjCerSDs7qjEuexv8DzcUgU42jfOyJs

g9zv4W1kh21okjmq+UNsu/KEoZWUz5HgSCNH39kCgYBpbqhgwsbrvIsfg2RQ3+gX

QOTGmsenrMHh5gJ8CrwWhT+pW8JPx6USnjh4HfD5PqsVKcqDvfAUURPFdmzinE8q

19bED1c/5015/kQPdaKDKgTr/E415sDC4U3NomnM3hv+GeOF9OworFOK4Z9/qpJj

TlHYNR4WBei2pTyEGiTmIQKBgQCFpmuCi3jiteqPUfXmabLm3Yo5cntBJwzByWdx

eniIaDe+I30Zrrhib7AUcnmvXgqOqKPeQ2QVCwM0XCGlkjzI/q9okYVTRnTfxuXc

mmi+he9mk6t8rMjO025kMj1x80u/qobxXEnUmx8lmyhgFfMxYf780ARXocRdfCuP

GgRyYQKBgQDY0Ko/F/4KPZWYVSFWvnFSzZ4CJAMgNHANqf49ixuCPaFrv7QfRlMJ

lBo5VIFredy551+2UA4eFJMPC1sbdF2pFN9kgHqIzEBXWPpZcZUfK8Y8l/Diz5n0

jflWWVReIhs3MDuK1gt74F1OXbY5aBLRsr4IIcv6jr214jrL+y5qNQ==

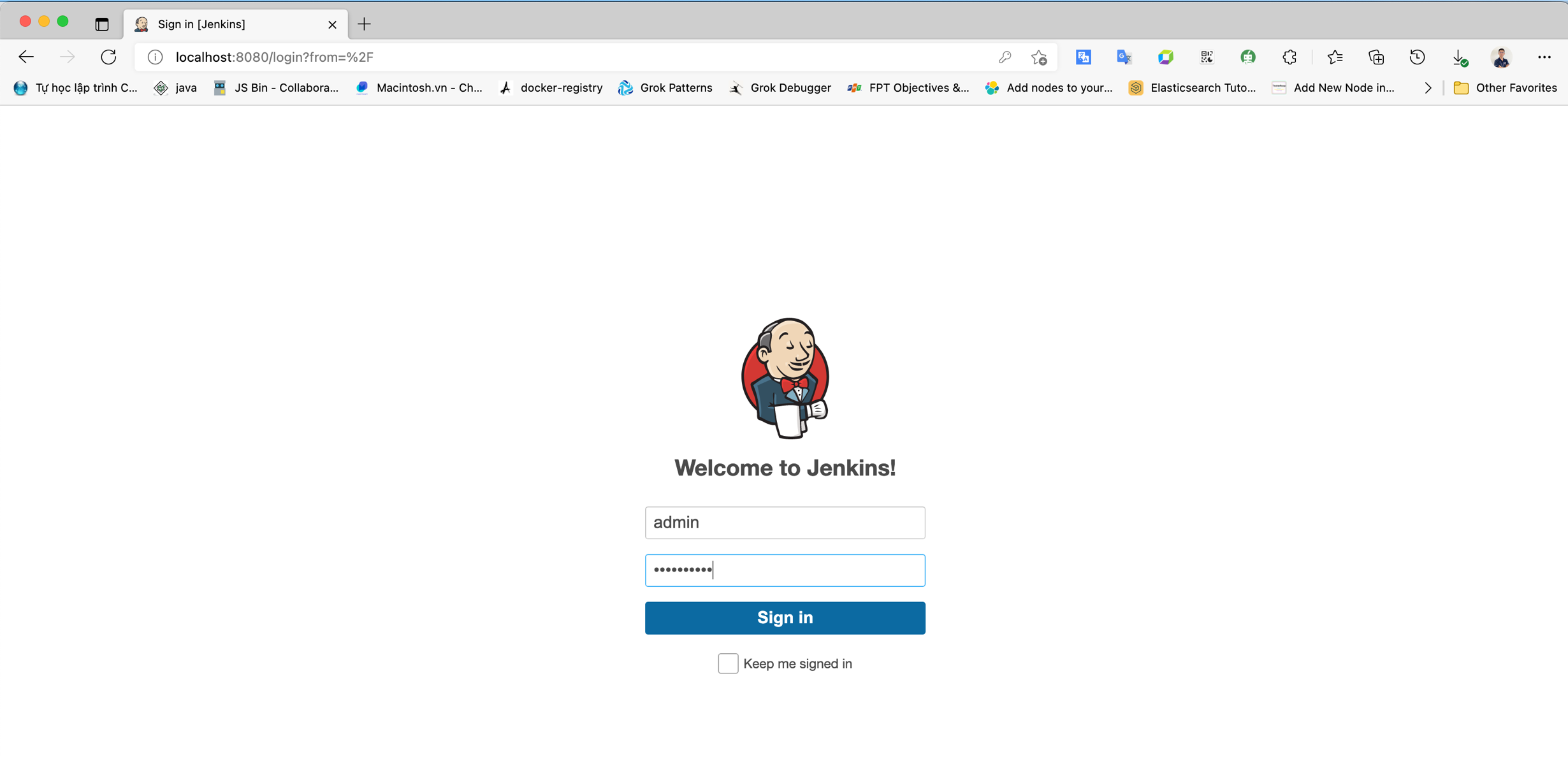
-----END RSA PRIVATE KEY-----

Step 2: Create tunnel connect to Database in private subnet by **dev-user.pem**.

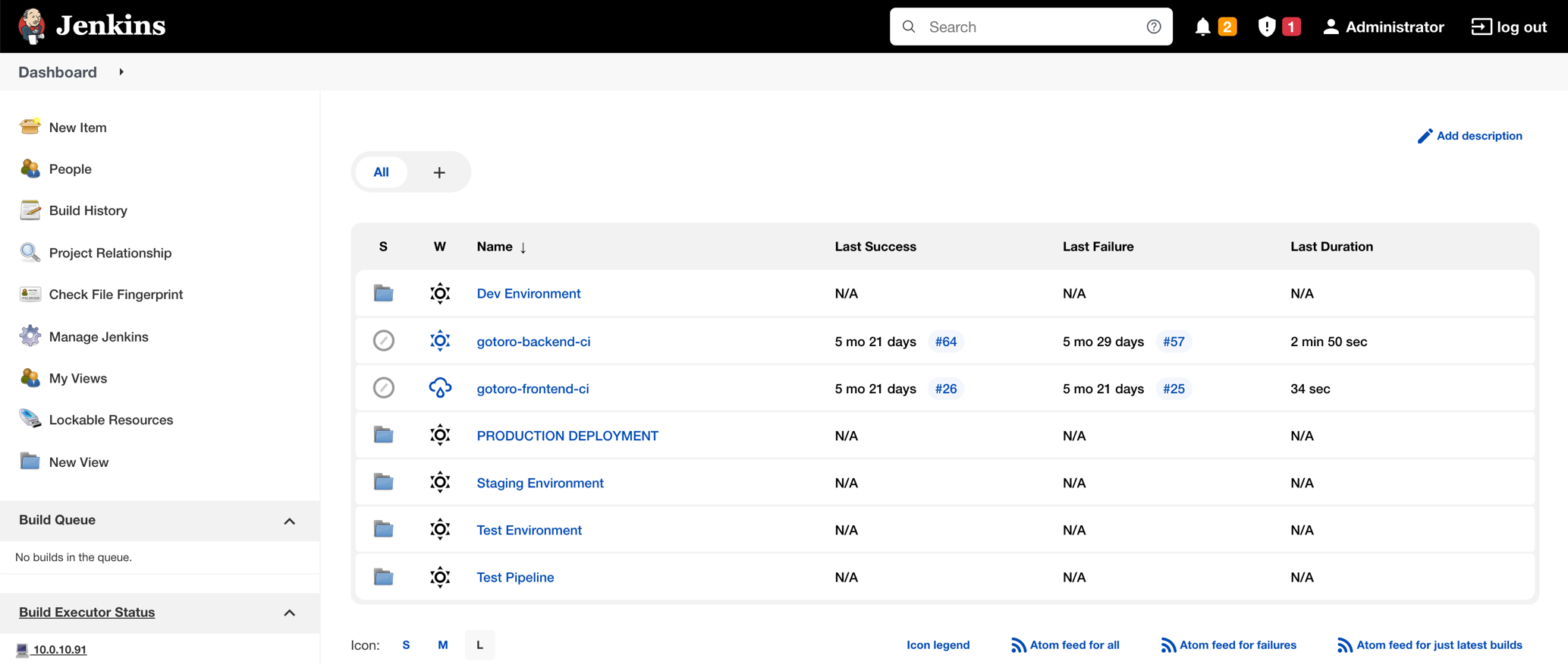
ssh -i dev-user.pem -vNL 8080:10.0.13.63:8080 [opc@150.136.245.180](mailto:opc@150.136.245.180) -p 443

Step 3: Open Edge Browser and access to localhost:8080, Username and password is **admin/DevOps@321.**

Example:

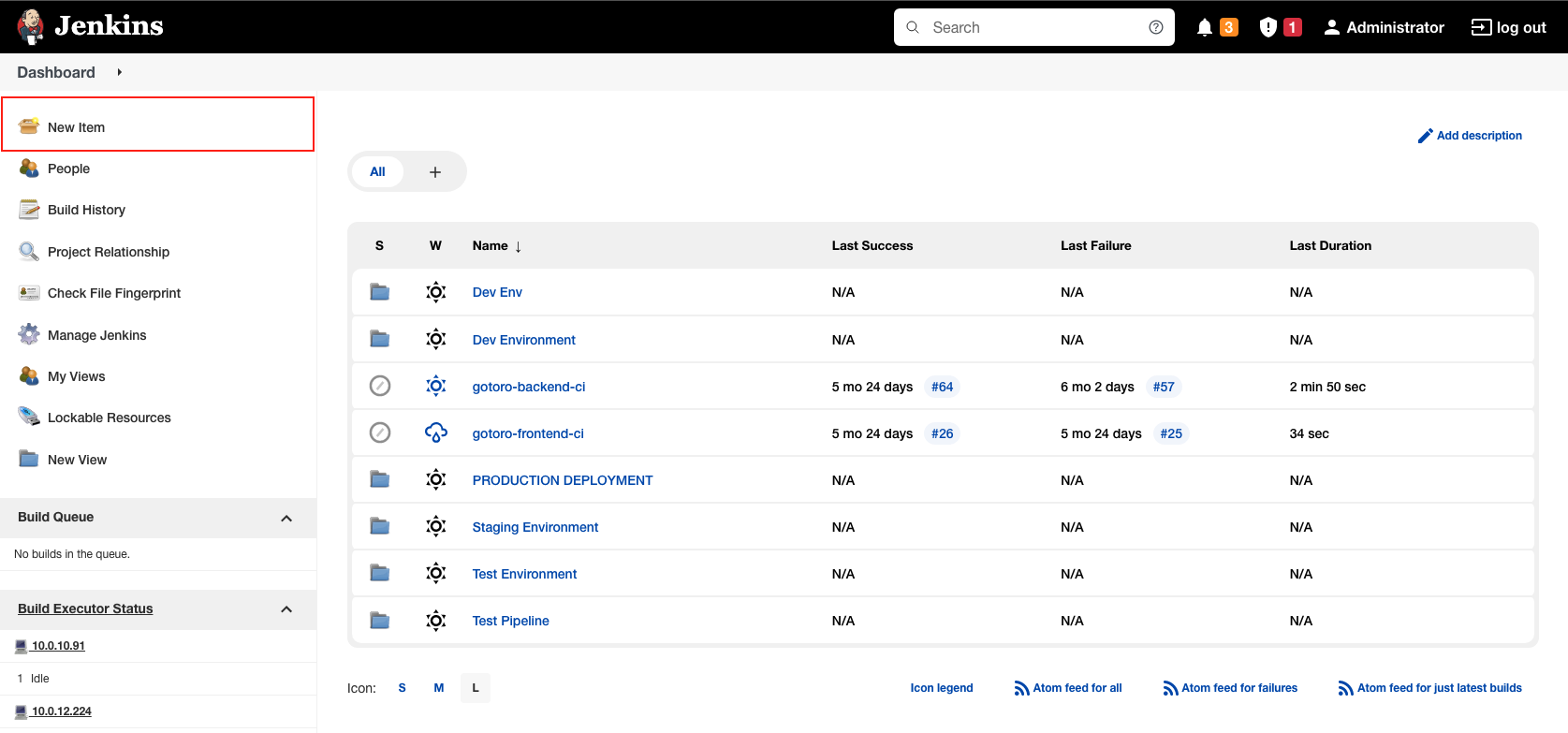


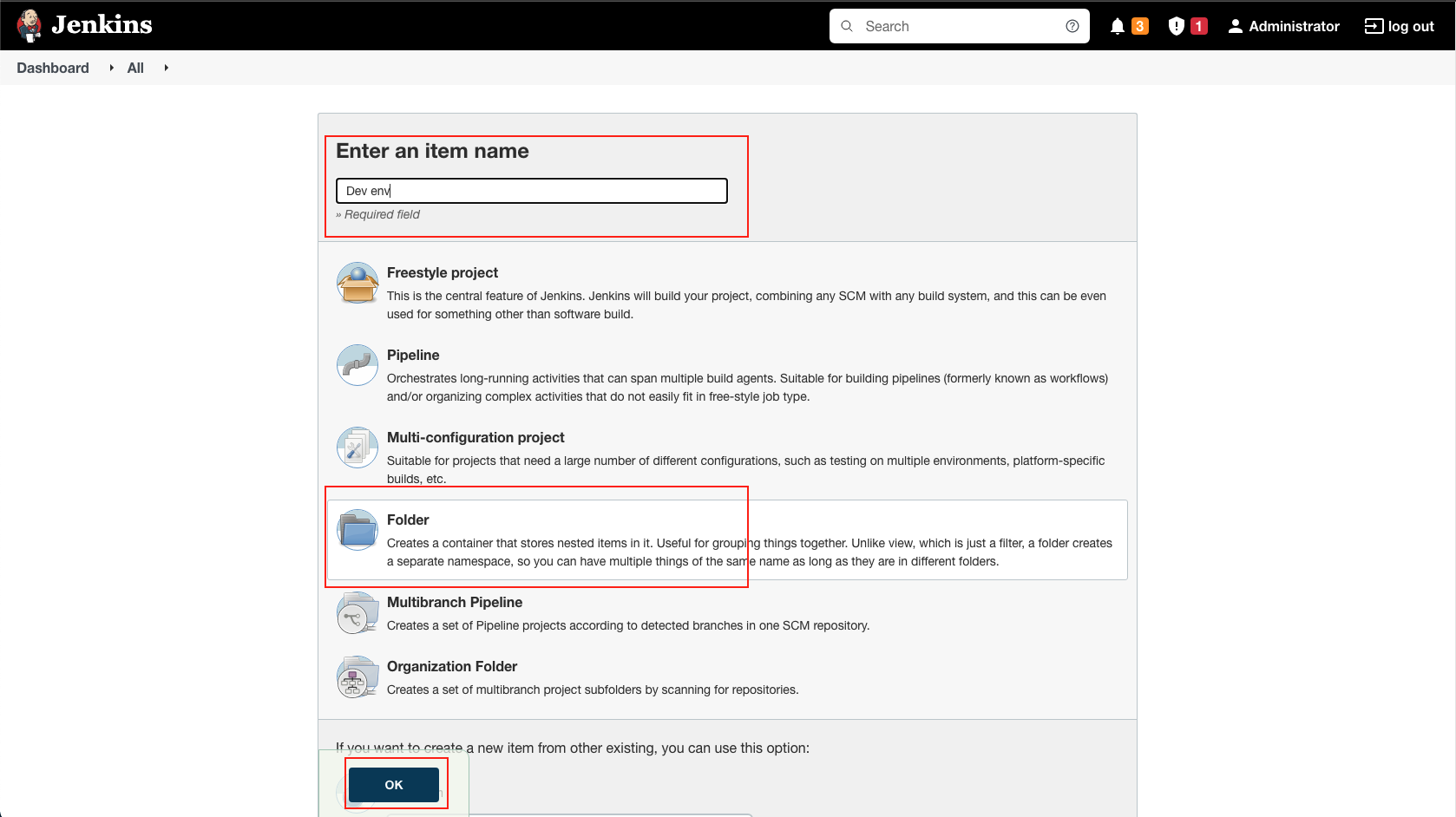
Result:



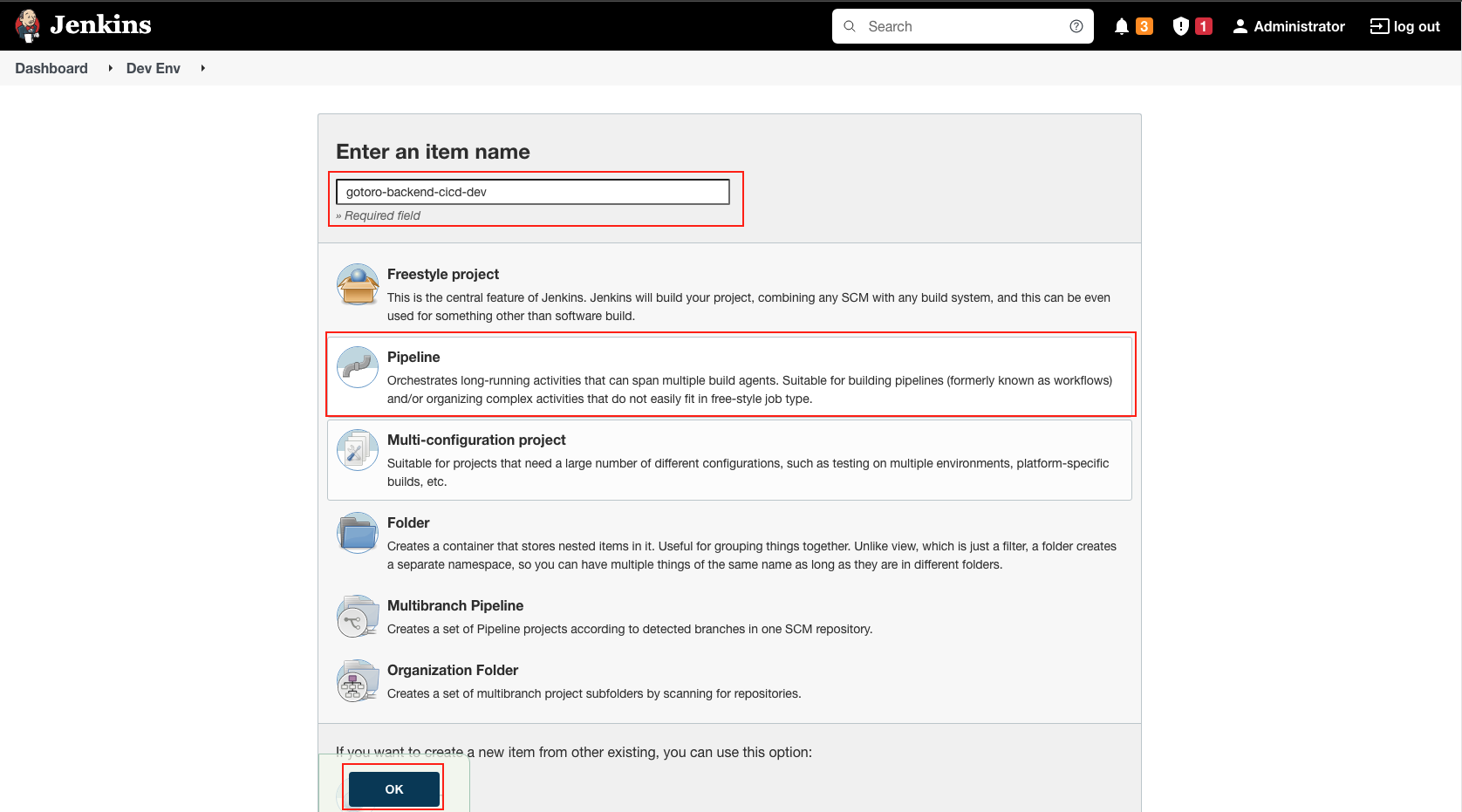
# Gotoro: Guideline to Add CICD pipeline to Deploy Backend -Dev

Step 1: Open Jenkins, Create folder for Dev ENV.

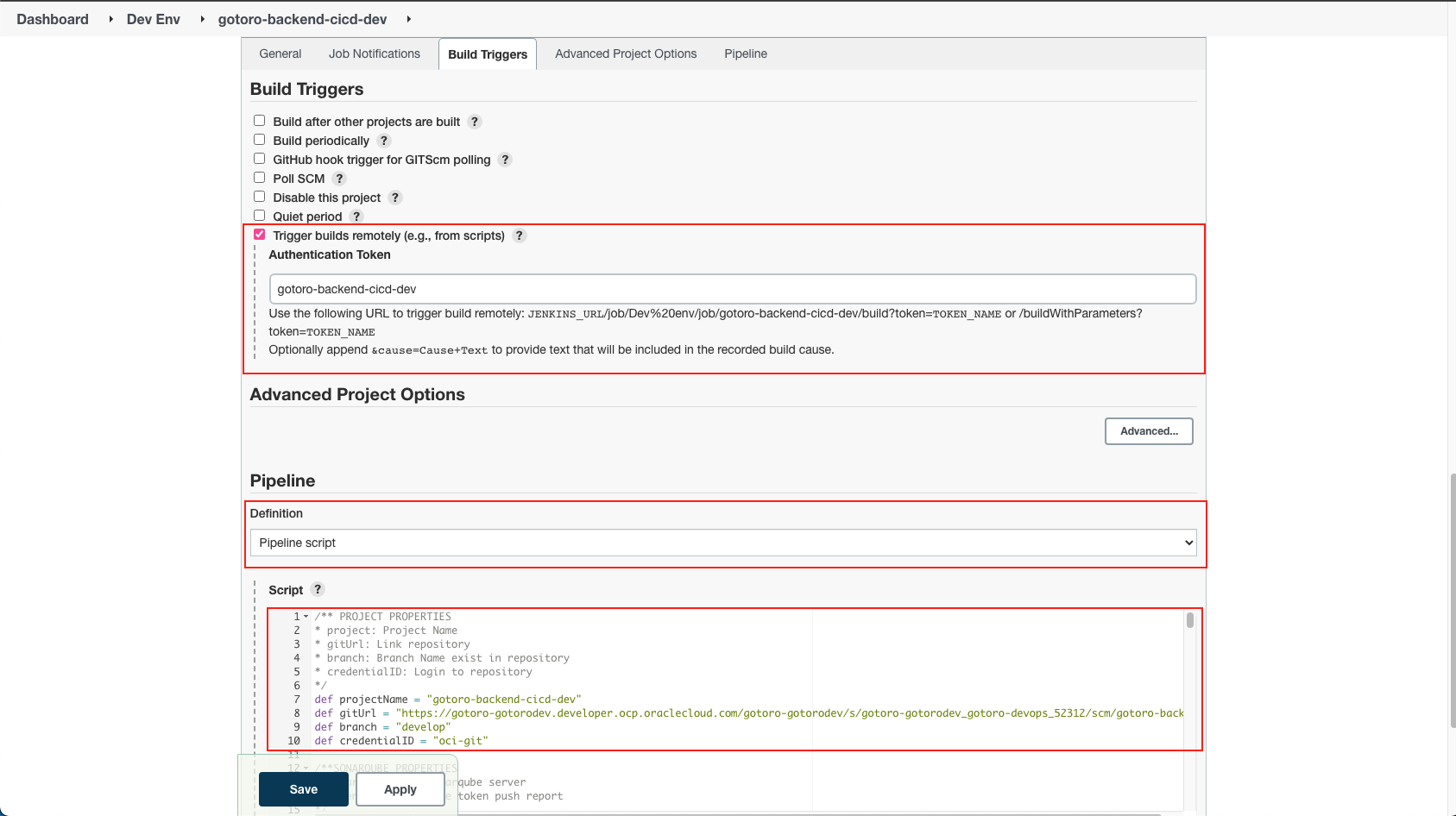




Step 2: Create pipeline



Step 3: Fill trigger and pipeline script.



Pipeline scipt:

/\*\* PROJECT PROPERTIES

\* project: Project Name

\* gitUrl: Link repository

\* branch: Branch Name exist in repository

\* credentialID: Login to repository

\*/

def projectName = "gotoro-backend-cicd-dev"

def gitUrl = "https://gotoro-gotorodev.developer.ocp.oraclecloud.com/gotoro-gotorodev/s/gotoro-gotorodev\_gotoro-devops\_52312/scm/gotoro-backend.git"

def branch = "develop"

def credentialID = "oci-git"

/\*\*SONARQUBE PROPERTIES

\* sonar\_host\_url: Sonarqube server

\* token\_sonarqube: Use token push report

\*/

def sonar\_host\_url = "http://10.0.13.243:9000"

def sonar\_access\_token = "97587d024005e1c8510708a6748d099ff081dfbc"

pipeline {

agent {

label 'agent'

}

environment {

SONAR\_HOME = "${tool 'sonar-scanner'}"

NODEJS\_HOME = "${tool 'nodejs-16'}"

BROWSERSLIST\_IGNORE\_OLD\_DATA = true

PATH = "${env.NODEJS\_HOME}/bin:${env.SONAR\_HOME}/bin:${env.PATH}"

ENV\_FOR\_DEPLOY = "dev"

DOCKER\_TAG = "${ENV\_FOR\_DEPLOY}-${currentBuild.number}"

PROJECT = "gotoro-backend"

REGISTRY\_URL = "us-ashburn-1.ocir.io"

BRANCH = "develop"

TOPIC\_ID = "ocid1.onstopic.oc1.iad.aaaaaaaaggp2sxsprepqgldrgfjktqnagvkkdutr7oecahn6fwpw3c3oyhqq"

SWAGGER\_OLD = "0.0.1"

SWAGGER\_NEW = "0.0.${currentBuild.number}"

}

/\*\* Checkout

\* Get source code from Git...

\*/

stages {

stage('Checkout') {

steps {

cleanWs()

checkout([$class: 'GitSCM',

branches: [[name: "${branch}"]],

doGenerateSubmoduleConfigurations: false,

extensions: [], gitTool: 'Default',

submoduleCfg: [],

userRemoteConfigs: [[credentialsId: "${credentialID}",

url: "${gitUrl}"]]

])

}

}

stage("Build Docker Images") {

steps {

sh "mv src/application.properties.${ENV\_FOR\_DEPLOY}.yaml src/application.properties.yaml"

// sh "echo SWAGGER\_OLD=${SWAGGER\_OLD}"

// sh "echo SWAGGER\_NEW=${SWAGGER\_NEW}"

// sh ''' sed -i "s/version: ${SWAGGER\_OLD}/version: ${SWAGGER\_NEW}/" src/application.properties.yaml '''

sh "echo Build Docker Images... "

sh "docker build -t ${PROJECT}:${DOCKER\_TAG} . "

}

}

stage("Push Image To Private Registry") {

steps {

sh "docker tag ${PROJECT}:${DOCKER\_TAG} ${REGISTRY\_URL}/idapel7w8ikn/${PROJECT}:${DOCKER\_TAG}"

sh "echo Push image to Oracle Private Registry... "

sh "docker push ${REGISTRY\_URL}/idapel7w8ikn/${PROJECT}:${DOCKER\_TAG}"

sh "docker image prune -af"

}

}

stage('Backup Container Log') {

agent {

node {

label 'application'

}

}

steps {

sh "/home/ubuntu/backup-log.sh"

}

}

stage('Deploy') {

agent {

node {

label 'dev-application'

customWorkspace '/home/ubuntu/gotoro-dev'

}

}

steps {

sh ''' export version=$(cat /home/ubuntu/gotoro-${ENV\_FOR\_DEPLOY}/docker-compose.yml | grep -m1 "${REGISTRY\_URL}/idapel7w8ikn/${PROJECT}:" | awk '{print $2}' | cut -d ":" -f2) && sed -i "s/${PROJECT}:$version/${PROJECT}:${DOCKER\_TAG}/" /home/ubuntu/gotoro-${ENV\_FOR\_DEPLOY}/docker-compose.yml '''

sh "cat /home/ubuntu/gotoro-${ENV\_FOR\_DEPLOY}/docker-compose.yml"

sh 'docker-compose up -d '

sh "docker image prune -af"

}

}

// stage('Send Notification') {

// agent {

// node {

// label 'application'

// }

// }

// steps {

// sh '''

// /root/bin/oci ons message publish --topic-id ${TOPIC\_ID} --title "Deployment status for Gotoro project" --body "A new version has been deployed on the PROJECT: ${PROJECT}, BRANCH: ${BRANCH}, ENVIRONMENT: ${ENV\_FOR\_DEPLOY}, VERSION: ${DOCKER\_TAG}."

// '''

// }

// }

stage('Backup package-lock.json') {

agent {

node {

label 'application'

}

}

steps {

cleanWs()

sh 'git config --global credential.helper cache'

sh 'git config --global push.default simple'

checkout([

$class: 'GitSCM',

branches: [[name: "develop\_package-lock\_backup"]],

extensions: [

[$class: 'CloneOption', noTags: true, reference: '', shallow: true]],

submoduleCfg: [],

userRemoteConfigs: [

[ credentialsId: 'oci-git', url: 'https://gotoro-gotorodev.developer.ocp.oraclecloud.com/gotoro-gotorodev/s/gotoro-gotorodev\_gotoro-devops\_52312/scm/gotoro-backend.git']]])

sh 'git checkout develop\_package-lock\_backup'

sh 'docker cp gotoro-backend-dev:/app/package-lock.json .'

sh 'git add .'

sh """

git diff-index --quiet HEAD || git commit -m "update package-lock.json on buildNumber: ${env.BUILD\_NUMBER}"

"""

sh 'git push'

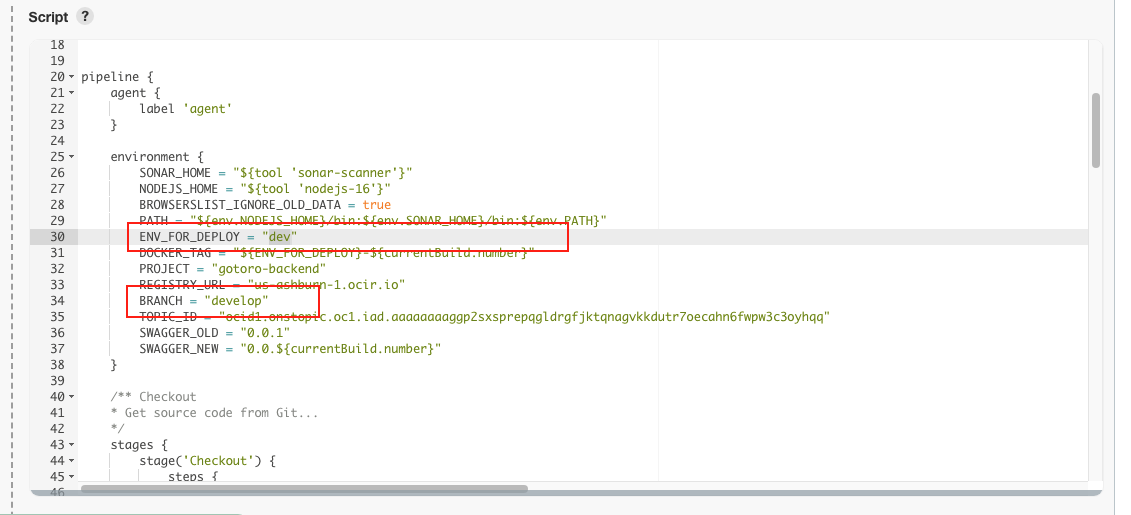
}

}

}

}

NOTE: Remember change ENV\_FOR\_DEPLOY, BRANCH checkout source code and Deploy stage Agent label:





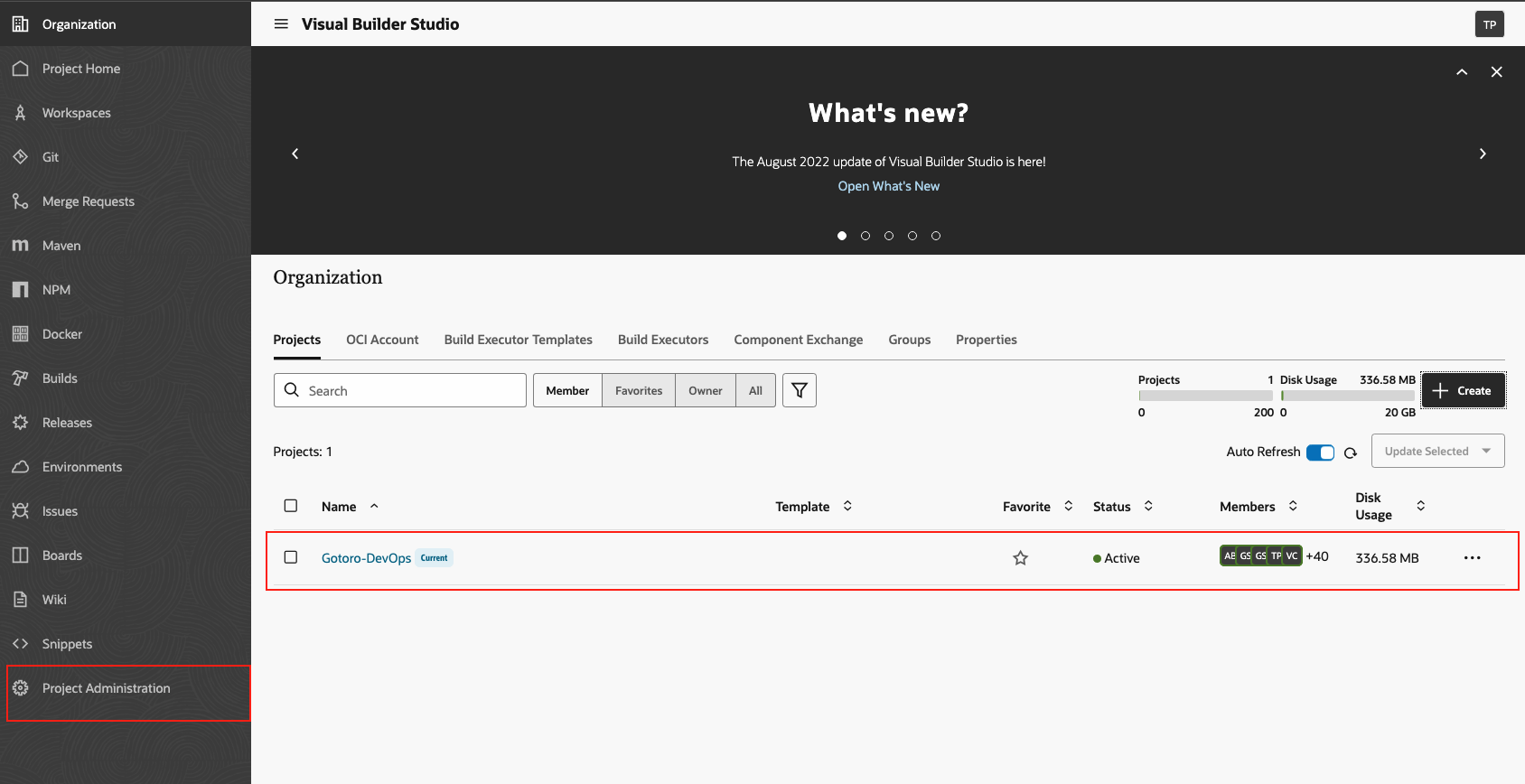
Step 4: Save.

# Set up trigger when create Pull Request on Oracle Repo

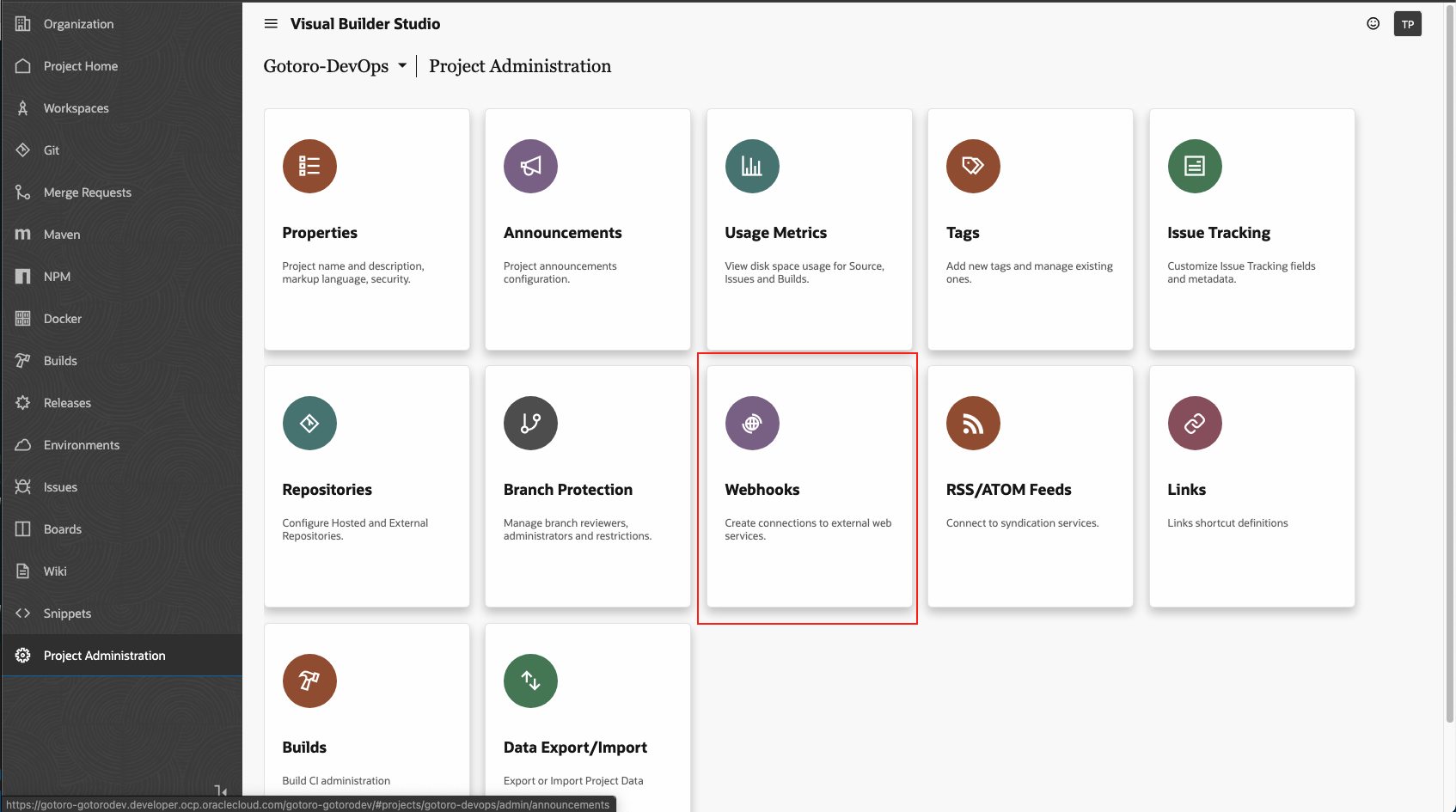
Step 1: Access to Gotoro Visual Builder Studio: [Welcome to Visual Builder Studio! (oraclecloud.com)](https://gotoro-gotorodev.developer.ocp.oraclecloud.com/gotoro-gotorodev/#?orgTab=projects)

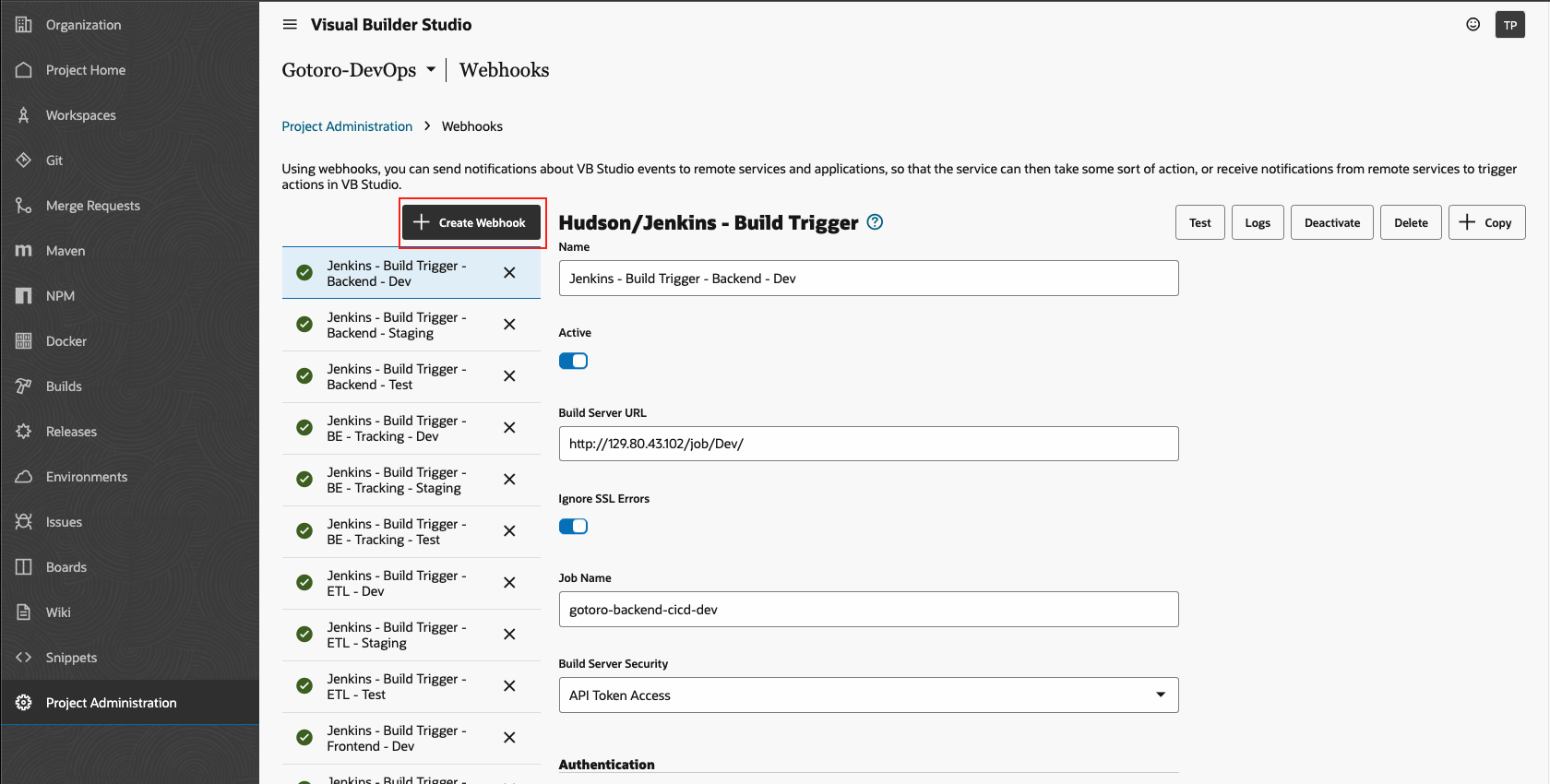
Login with your oracle account.

After login Click to project and Project Administrastion.

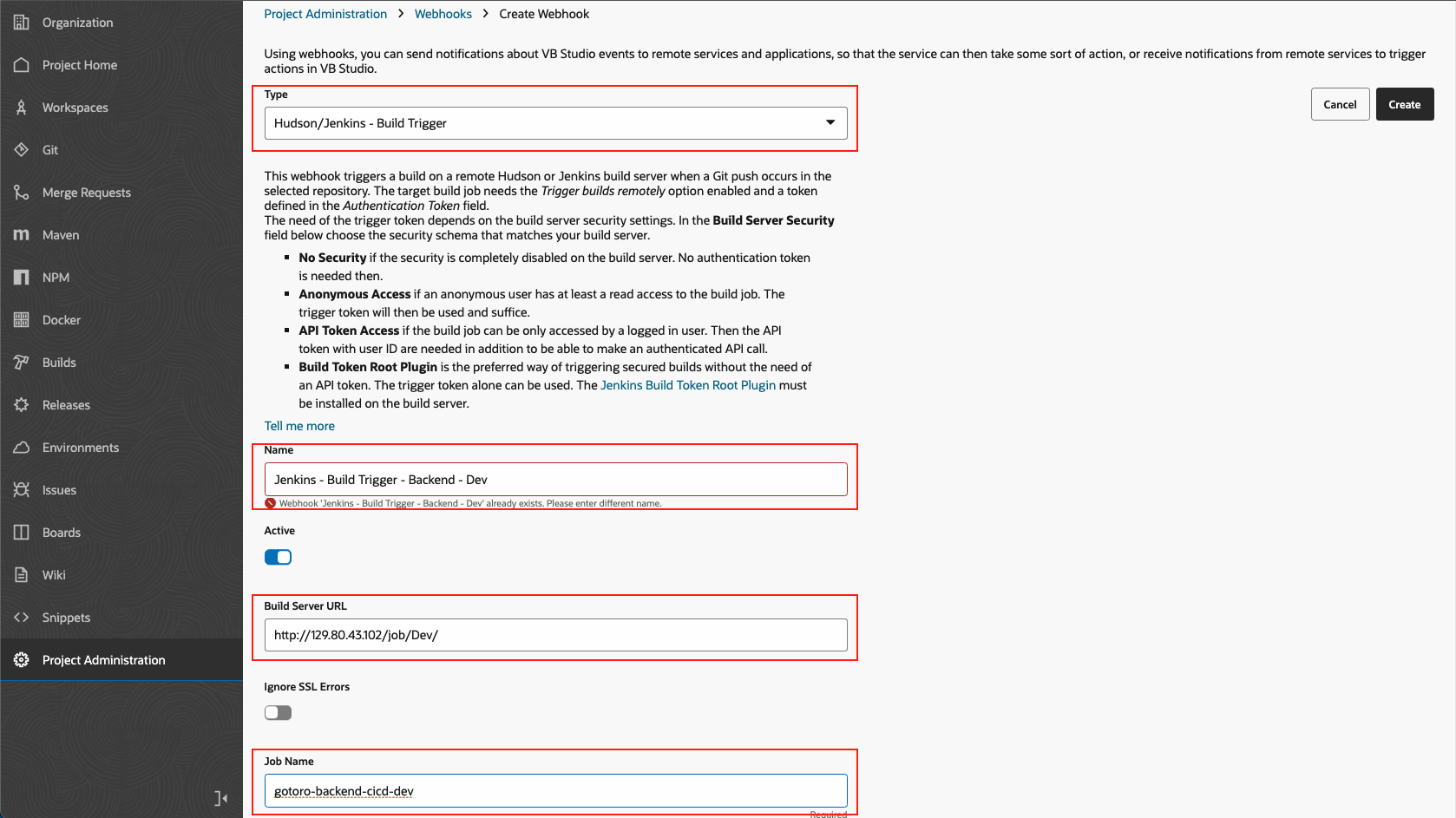


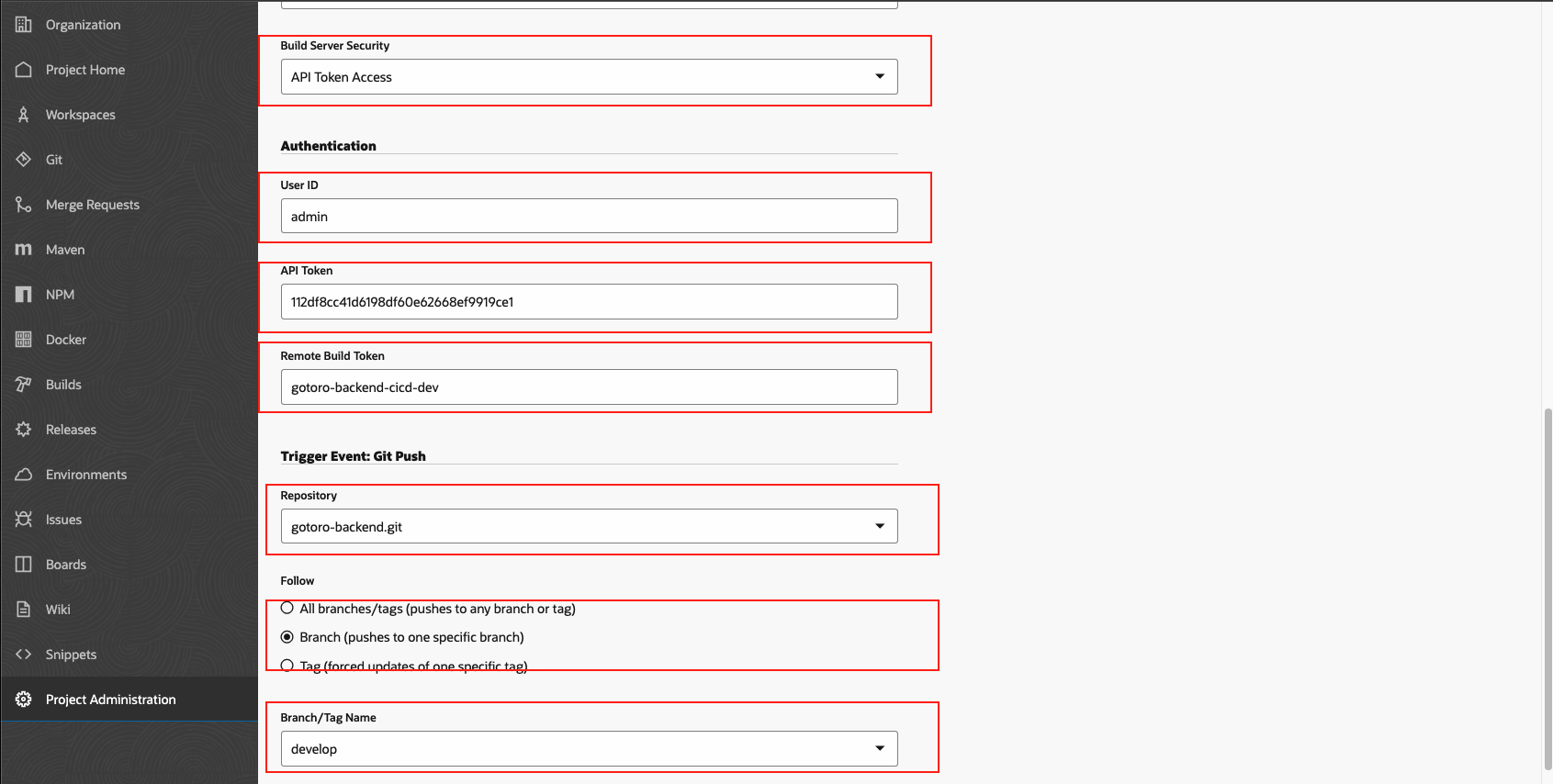
Step 2: Open Webhooks config and Create Webhooks.



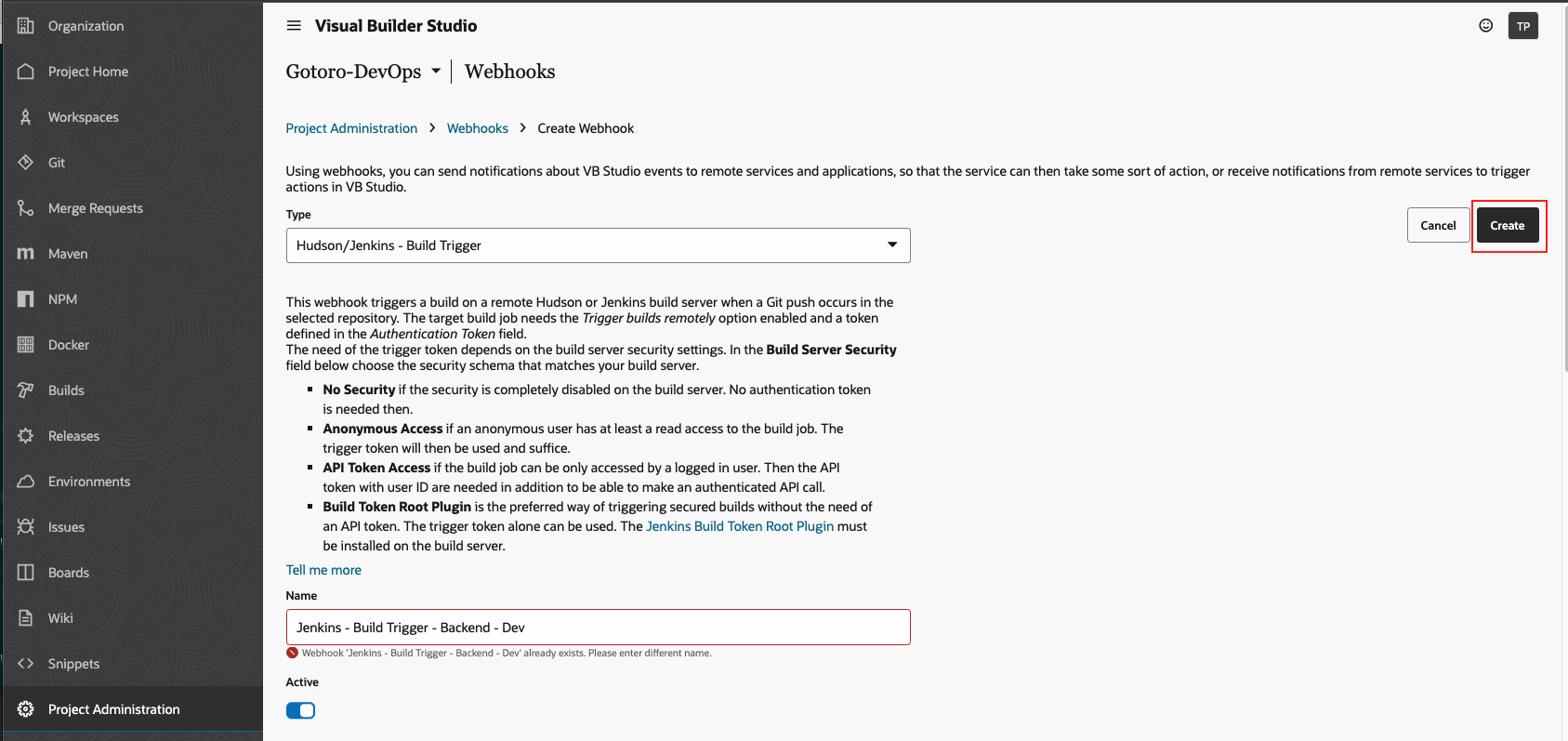


Step 3: Chose type and fill information.





Step 4: Create



**End Guideline.**