|| Jenkins ||

Installation of Jenkins (/root)

• sudo apt-get update -y

• sudo apt-get upgrade -y

• sudo apt-cache search openjdk

• sudo apt-get install openjdk-8-jdk

• java -version

• wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | apt-key add -

• sudo vi /etc/apt/sources.list --> deb https://pkg.jenkins.io/debian-stable binary/

• sudo apt-get update -y

• sudo apt-cache search jenkins

• apt-cache madison jenkins

• sudo apt-get install jenkins -y

• sudo service jenkins status

• cat /var/lib/jenkins/secrets/initialAdminPassword

Setting Webhook(Github)

• Go to particular repository

• Go to repository setting

• Go to webhook

• Add wedhook ---------- http//:3.0.12.145.8080/github-webhook/

• save

Connect Jenkins with Backend Server (/home/ubuntu)

• ls -a

• cd .ssh/

• ssh-keygen

• ls

• cat id\_rsa.pub -------------- copy public key and paste in on backend server’s authorized key

• go to backend server

• ls -a

• cd .ssh/

• ssh-keygen

• ls

• sudo nano authorized\_keys

• sudo chown -R ubuntu:ubuntu /home/ubuntu/new\_chatapp

Creating pipeline in Jenkins

Install ssh agent plugins

Create credential for SSH Agent [

• go to Jenkins server

• cd .ssh/

• ls

• cat id\_rsa --------- copy the content in credential add section ]

Pipeline Syntax :

pipeline{

agent any

stages{

stage("git"){

steps{

git 'https://github.com/Priya412-coder/new\_chatapp-1.git'

}

}

stage('build') {

steps {

sshagent(['jenkins-deploy']) {

sh "scp -r -o StrictHostKeyChecking=no fundoo ubuntu@10.0.1.127:/home/ubuntu/new\_chatapp "

}

}

}

stage('deploy') {

steps {

sh 'ssh -i /var/lib/jenkins/jenkins.pem ubuntu@10.0.1.127 "bash /home/ubuntu/new\_chatapp/scripts/Start\_Server.sh"'

}

}

Jenkins using Sonarqube :

• Create a sonarqube server using AWS EC2

• Install sonarqube on same server

• https://www.vultr.com/docs/install-sonarqube-on-ubuntu-20-04-lts/

https://www.how2shout.com/linux/install-sonarqube-on-ubuntu-20-04-18-04-server/

• login to sonarqube server

• create project inside the sonar server

• copy the generated token in notepad

• go to manage Jenkins -- go to manage credential -- create a new credentials using secret text and paste the generated token from sonarqube project

• go to manage jenkins --> configure system --> go to sonarqube add name along with the IP Server address of Sonarqube | Ex: http://localhost:9000

install sonarqube scanner plugins

• go to manage Jenkins -- Global tool configuration --- add sonarqube scanner with sonarqube credential

Pipeline Syntax : (using sonarqube)

pipeline{

agent any

stages{

stage("git"){

steps{

git 'https://github.com/Priya412-coder/new\_chatapp-1.git'

}

}

stage ("SonarQube Analysis") {

steps {

script {

def scannerHome = tool 'sonar-scanner'; --- scanner name that we have given in global tool config

withSonarQubeEnv('sonarqube') { ---------------- sonarqube name that we have give in config system

sh "${tool("sonar-scanner")}/bin/sonar-scanner -Dsonar.projectKey=sonar\_qube -Dsonar.projectName=new-chat-app" (Note: Projectkey is credential key for sonarqube and project name is the name you have given to project in sonarqube)

}

}

}

}

stage('build') {

steps {

sshagent(['jenkins-deploy']) {

sh "scp -r -o StrictHostKeyChecking=no fundoo ubuntu@10.0.1.127:/home/ubuntu/new\_chatapp "

}

}

}

stage('deploy') {

steps {

sh 'ssh -i /var/lib/jenkins/jenkins.pem ubuntu@10.0.1.127 "bash /home/ubuntu/new\_chatapp/scripts/Start\_Server.sh"'

}

}

Jenkins using Sonarqube with quality gate :

Go to sonarqube server --- project -- add webhook -- give name to webhook --- http//:21.0.3.4:8080/sonarqube-webhook

Go to sonarqube server --- quality gates --- create new quality gate with condition --- select all option in project

Pipeline Syntax : (using sonarqube and quality gate)

pipeline{

agent any

stages{

stage("git"){

steps{

git 'https://github.com/Priya412-coder/new\_chatapp-1.git'

}

}

stage ("SonarQube Analysis") {

steps {

script {

def scannerHome = tool 'sonar-scanner'; --- scanner name that we have given in global tool config

withSonarQubeEnv('sonarqube') { ---------------- sonarqube name that we have give in config system

sh "${tool("sonar-scanner")}/bin/sonar-scanner -Dsonar.projectKey=sonar\_qube -Dsonar.projectName=new-chat-app" (Note: Projectkey is credential key for sonarqube and project name is the name you have given to project in sonarqube)

}

}

}

stage("Quality Gate") {

steps {

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

waitForQualityGate abortPipeline: true

}

}

}

}

stage('build') {

steps {

sshagent(['jenkins-deploy']) {

sh "scp -r -o StrictHostKeyChecking=no fundoo ubuntu@10.0.1.127:/home/ubuntu/new\_chatapp "

}

}

}

stage('deploy') {

steps {

sh 'ssh -i /var/lib/jenkins/jenkins.pem ubuntu@10.0.1.127 "bash /home/ubuntu/new\_chatapp/scripts/Start\_Server.sh"'

}

}