SIMPLE DEVOPS PROJECT-1

CI/CD THROUGH JENKINS

Creating Tomcat-Server

Step 1: creating an AWS instance with RedHat Linux and establishing the connection with putty using <filename>.ppk file

Step 2: using sudo su – command to become root user then, installing java using yum install java-1.8\*.

Step 3: Now installing Apache tomcat server using wget command in the new directory named opt.

Step 4: Extracting Apache tomcat zip file using tar -zvxf < **tar -zvxf apache-tomcat-8.5.79.tar.gz** >

Step 5: check the extracted file using ls command.

Step 6: Now we’re moving into the directory of Apache tomcat.

Step 7: From this directory we’re moving into the bin directory.

Step 8: There we can see startup.sh and shutdown.sh.

Step 9: We are making those two script files to executable using chmod +x <filename> command.

Step 10: Now we are creating a softlink to star and stop the tomcat server, the command ln -s /opt/apache-tomcat-8.5.79/bin/startup.sh usr/local/bin/tomcatup as well as for shutdown use the command ln -s /opt/apache-tomcat-8.5.79/bin/shutdown.sh usr/local/bin/tomcatdown **(**After executing these commands we can just give tomcatup and tomcatdown to start and stop the server**).**

Step 11: Next, we’re going to start the server using tomcatup. Then check if the server is running or not using the public ipv4 address followed by 8080. Before that add the inbound rule for the range 8080 and check it with <http://43.204.35.228:8080/>

Step 12: Now again change the inbound rule because Jenkins also runs in the same port 8080. So, we must change that range from 8080 to 8090. Then check it with <http://43.204.35.228:8090/>

Step 13: To change the port from 8080 to 8090 go to the conf file and inside that edit the server.xml file using the command **vi server.xml. While editing change the port inside the server.xml file from 8080 to 8090 and save it using :wq!**

Step 14: To access the sub modules inside the tomcat page, we need to edit the context.xml file as (<!-- and -->) for values that are available in webapps like **/opt/apache-tomcat-8.5.79/webapps/host-manager/META-INF/context.xml**and**/opt/apache-tomcat-8.5.79/webapps/manager/META-INF/context.xml.**

Step 14: Then you need to add the username and password to access the **manager app page.**

Step 14: To add username inside the conf folder edit the **tomcat-users.xml** file with the code like

**<role rolename="manager-gui"/>**

**<role rolename="manager-script"/>**

**<role rolename="manager-jmx"/>**

**<role rolename="manager-status"/>**

**<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>**

**<user username="deployer" password="deployer" roles="manager-script"/>**

**<user username="tomcat" password="s3cret" roles="manager-gui"/>**

Then save the **tomcat-users.xml**using **:wq!** command.

Creating Jenkins-Server

Step 1: Creating a new instance in AWS as Jenkins-Server.

Step 2: using sudo su – command to become root user then, installing java using yum install java-1.8.0. Then check java version using java -version.

Step 3: Install Jenkins but, before that we must install wget using yum -y install wget.

Step 4: Then using wget we must add the repository of Jenkins like

wget -O /etc/yum.repos.d/jenkins.repo <https://pkg.jenkins.io/redhat-stable/jenkins.repo>

Step 5: Then rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io.key>

Step 6: Then Install Jenkins using yum -y install jenkins

Step 7: Start the jenkins service using systemctl start jenkins

Step 8: Setup Jenkins to start at boot using systemctl enable jenkins

Step 9: Access the Jenkins using the public ipv4 address <http://65.0.29.202:8080/>

Step 10: To configure Jenkins we must add Admin Password Grab the default password from this Password Location:/var/lib/jenkins/secrets/initialAdminPassword using

* cat /var/lib/jenkins/secrets/initialAdminPassword

Step 11: Skip the plugins installation or install the recommended plugins

Step 12: Create a new user with username, password, full name, email address then, click save and continue.

### Step 13: Test Jenkins Jobs

### Create “new item”

### Enter an item name as “Test\_job”

### Chose Freestyle project

### Under Build section Execute shell: echo "Hello Guys…! Welcome to Jenkins"

### Save your job

### Build job

### Check "console output"

CI/CD THROUGH JENKINS

Step 1: First login to the AWS account and restart the jenkins server and Tomcat server (Web Server and Jenkins\_Server).

Step 2: Using the IPV4 address of the instances connect them using putty.

Step 3: Using the localhost addresses 52.66.18.47(Jenkins\_Server) and 52.66.206.40(Web\_Server) and start the jenkins and tomcat server inside the server like tomcatup and systemctl start jenkins. Then, check if the server is running or not by using

ps -ef | grep tomcat and systemctl status jenkins.

Step 4: Now install the git inside the Jenkins server using the commands **sudo su and yum install git** and go to path: /usr/bin/git and copy that path using pwd command.

Step 5: Now go to the /opt folder and create a directory called maven using the command mkdir maven.

Step 6: Go to the maven folder by using the command cd maven

Step 7: Now copy the link to download the maven from the webpage and download it in Jenkin server using the command

wget <https://dlcdn.apache.org/maven/maven-3/3.8.5/binaries/apache-maven-3.8.5-bin.tar.gz>

Step 8: Unzip the tomcat file using the command tar -zvxf apache-tomcat-8.5.79.tar.gz

Step 9: Then get into that unzipped folder using cd apache-tomcat-8.5.79

Step 10: Copy the path of the maven using pwd

Step 11: Login to the Jenkins webpage » Give the respective username and password » Get into the Jenkins dashboard.

Step 12: Install some plugins for git and maven like maven integration, deploy to container and unleash maven etc.,

Step 13: Now go to Manage Jenkins » Global tool Configuration and then give the name and add the path of the maven and git. For git /usr/bin/git » For maven /opt/maven/apache-maven-3.8.5

Step 14: Now apply and save the Global tool configuration

Step 15: Create **Jenkins job**, Fill the following details,

* + **Source Code Management:**
    - Repository: https://github.com/AthulyaG/hello-world.git
    - Branches to build: \*/master
  + **Build:**
    - Root POM: pom.xml
    - Goals and options: clean install package

Step 16: To deploy our build artifacts on tomcat server our Jenkins server need access. For this we should setup credentials. This option is available in Jenkins’s home page

* setup credentials
  + credentials > Jenkins > Global credentials > add credentials
    - Username: deployer
    - Password: XXXXXXXX
    - id: Tomcat\_user
    - Description: Tomcat user to deploy on tomcat server

Step 17: Post Steps

* + Deploy war/ear to container
    - WAR/EAR files: \*\*/\*.war
    - Containers: Tomcat 8.x
      * Credentials: deployer/\*\*\*\*\*\* (tomcat credentials)
      * Tomcat URL: http://<PUBLIC\_IP>:8090

Step 18: Save and run the job now.

Step 19: Continuous Integration and Continuous Deployment (CI/CD)

Now job is running fine but to make this as Continuous Integration and Continuous Deployment go back and modify job as below.

* Build Triggers » Poll SCM
  + - schedule \*/2 \* \* \* \*

Save the job and modify the code in GitHub. Then you could see your job get trigger a build without any manual intervention.

Step 20: To check the output of the job that is built above use the tomcat URL as http://<PUBLIC\_IP>:8090/webapp