JENKINS - DAY 02

Project 2: JENKINS + GITHUB + MAVEN + TOMCAT - Integration
~ Code is usually available in GITHUB Repo. We have to create a Jenkins job that job should take code from GITHUB repo. It should compile and package that code using Maven, and then the war file gets generated. This war file should be deployed into the Tomcat Webserver. So, here we are going to do Build & Deployment Process.
~ Whenever we do any changes in the Source Code available in the GITHUB Repo, automatically the jenkins job should build and package the updated code and deploy into the webserver.
\sim By default the Tomcat will run on Port No. 8080
~ By default the Jenkins will run on Port No. 8080
\sim Here we have to change the Port No. of Tomcat to 9090.
Tasks:
Part 1
1. Create an EC2 Instance - Connect to the instance using MobaXTerm Tool.
Part 2

JENKINS INSTALLATION ON LINUX EC2 INSTANCE

```
~ Add Jenkins repo. to our YUM repo:
       sudo wget -O /etc/yum.repos.d/jenkins.repo \
  https://pkg.jenkins.io/redhat-stable/jenkins.repo
~ Import a key-file from Jenkins-CI to enable the installation from package:
       sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
       sudo yum upgrade
~ Installing Java
       sudo amazon-linux-extras install java-openjdk11 -y
       java -version
~ Install Jenkins:
       sudo yum install jenkins -y
~ Lets start, enable, and check the status of Jenkins
       sudo systemetl enable jenkins
       sudo systemctl start jenkins
       sudo systemetl status jenkins
               You should see Active & Running in Green Colour, which means the Jenkins
service is successfully installed
~ Check the jenkins accessibility. By default the Jenkins will run on Port No. 8080.
So, open Port No. 8080 for EC2 instance to access Jenkins
       "Check" the instance ----> Security ----> Click on the link under SGs ----> Add Rules ----
> Custom TCP, 8080, Anwhere ----> Save Rules.
```

Open new tab: <Paste the Public IP of Instance>:8080 ----> You will see the Jenkins Home Page ----> Now we have to Unlock the Jenkins ----> You will see a code in the red colour ----> Copy the code in red colour ----> Go to MobaXTerm ----> sudo cat <paste the red color code> ----> You will see a password. Copy that password and paste in jenkins home page ----> Click on suggested plugins ----> You will see "Create First Admin User" ----> Enter Username, Password, Name, Email Id ----> Save and Continue ----> Jenkins URL:http://52.30.109.196:8080/ ----> Save and Finish ----> You will see "Jenkins is ready!" ----> Click on "Start using jenkins" ----> Now you can see the jenkins home page and here you can create Jenkins Jobs.

Part 3

TOMCAT WEBSERVER INSTALLATION ON LINUX EC2 INSTANCE

https://tomcat.apache.org/download-90.cgi ----> Downloads ----> Tomcat 9 ----> Copy the link of tar.gz file ----> https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.73/bin/apache-tomcat-9.0.73.tar.gz ----> sudo wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.73/bin/apache-tomcat-9.0.73.tar.gz ----> ls -l ----> You will the tomcat file in red colour. This is a tar.gz file and we have to extract the tar.gz file ----> tar -xvf <enter the name of tomcat file in red colour> ----> All the files will get extracted ----> ls -l ----> You will see the list of extracted files ---> cd <enter the name of tomcat file in blue colour>

Lets start the tomcat server:

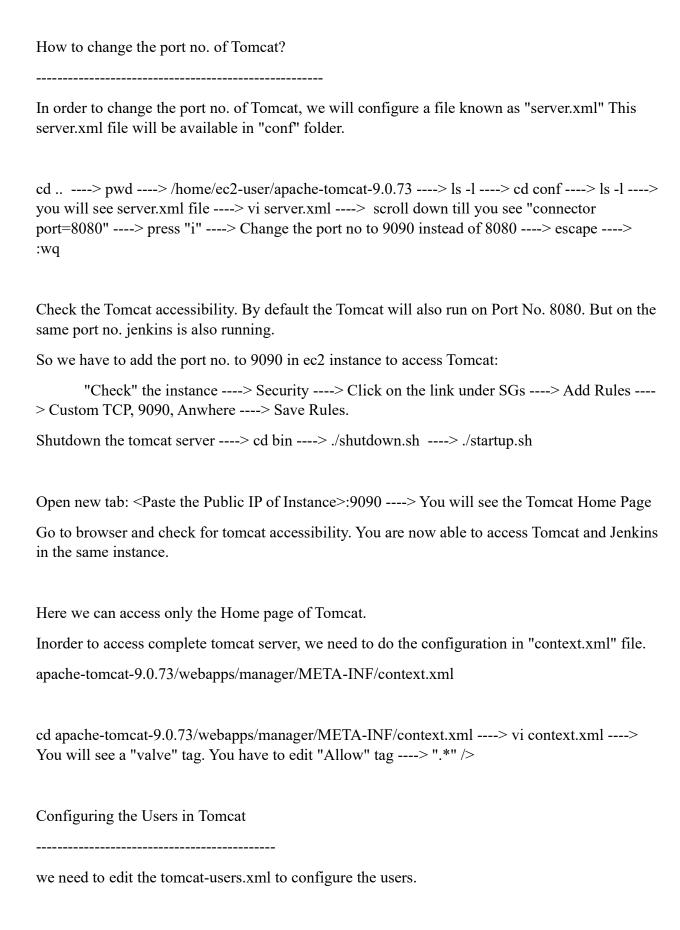
Inorder to start the tomcat server, we need a file called "startup.sh"

This "startup.sh" file will be there in "bin" folder ----> cd bin ----> ls -l ----> you will see startup.sh file ----> ./startup.sh ----> You will see "Tomcat Started"

Lets access the tomcat server:

Since, the tomcat also runs on Port No. 8080, we cannot access tomcat as on the same port no. jenkins is also running.

Now, we have to change the Port No. of Tomcat. (Tomcat Port No. 9090)



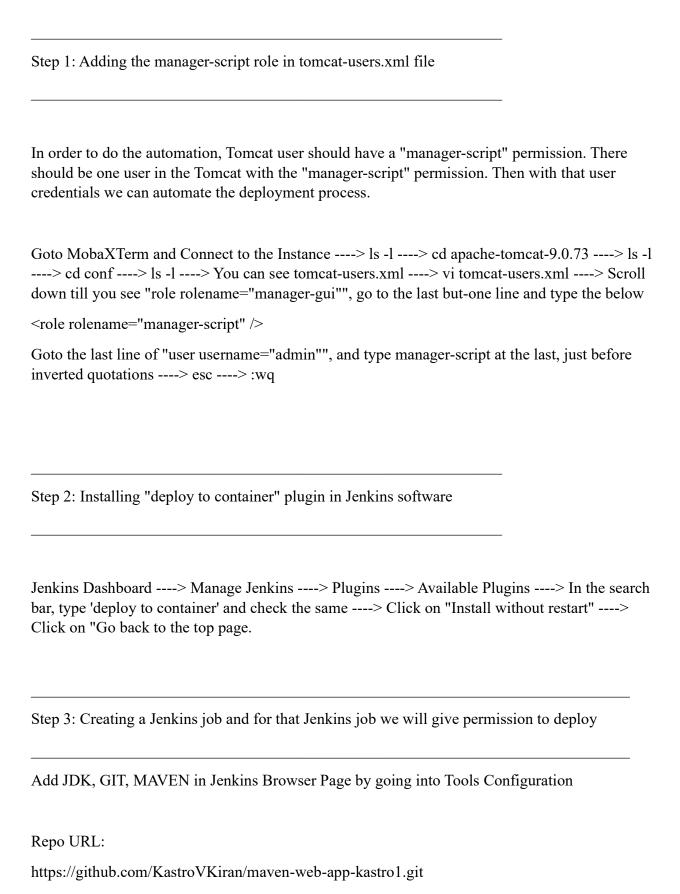
<role rolename="manager-gui"></role>
<user password="tomcat" roles="manager-gui" username="tomcat"></user>
<role rolename="admin-gui"></role>
<user password="admin" roles="manager-gui,admin-gui" username="admin"></user>
cd bin
./shutdown.sh
./startup.sh
Access Tomcat on Browser> Click on Server Status> Enter the username (admin) and password (admin)
Install git on VM
Install GIT in EC2 Instance where Jenkins is running
sudo yum install git -y
Jenkins Port No.: 8080
Tomcat Port No.: 9090

Part 4

Inorder to do the automation, Jenkins is going to execute some script to deploy the war file into

the Tomcat Webserver.

The "tomcat-users.xml" file is available in "conf" folder



Part 5

1) Create Jenkins Job

- -> New Item.
- -> Enter Item Name (Job Name) Name: git-maven-tomcat-integration
- -> Select Free Style Project & Click OK
- -> Enter some description
- -> Go to "Source Code Management" Tab and Select "Git"
- -> Enter Project "Git Repo URL" i.e https://github.com/KastroVKiran/maven-web-app-kastro1.git

Branch name: */main

- -> Go to "Build Triggers" tab. Here i will trigger the job manually.
- -> Go to "Build Steps" tab
 - Click on Add Build Step and Select 'Inovke Top Level Maven Targets'
- -> Select Maven and enter goals
 - Maven Version: Maven-3.9.6
 - Goals: clean package
- -> Click on 'Post Build Action' and Select 'Deploy war/ear to container' option
- -> Give path of war file (You can give like this also: **/*.war)
- -> Enter Context Path (give project name Ex: java web app)
- -> Click on 'Add Container' and select Tomcat version 9.x
- -> Credentials ---> Click on "Add" ---> A dia ---> Username: admin, Password: admin, ID: Tomcat-Credentials, Description: tomcat

credentials ---> Add

-> Credentials ---> Select "Tomcat-Credentials"

- -> Enter Tomact Server URL (http://ec2-vm-public-ip:tomcat-server-port)
- -> Click on Apply and Save

Note: To do any edits to the job, click on "Configure" and edit accordingly

- 2) Run the job now using 'Build Now' option and see see 'Console Output' of job
- 3) Once Job Executed successfully, go to tomcat server dashboard and see application should be displayed.
- 4) Goto Tomcat browser and click on "maven-web-app"

Click on the application name (it should display application)

For ex, assume that i have modified the code in github repo; does the changes gets reflected in our app? Lets see

Goto repo https://github.com/ashokitschool/maven-web-app.git ---> src/main/webapp/index.jsp ---> Change the code accordingly ---> Commit changes ---> Goto Tomcat browser and reload ---> You dont see any changes.

So what we need to do is; we need to rebuild the job ---> Goto Jenkins UI ---> Build Now ---> Goto Tomcat Browser ---> Refresh the page ---> You will see the changes.

But here there is a problem, whenever there is a code change, how DevOps engineers knows it? So, here we need to do automation in such that, whenever there is a change in the code in github repo, automatically those changes should reflect in the app. Lets do that.

Jenkins UI ---> Configure ---> Build triggers ---> Poll SCM ---> Schedule: * * * * * ---> Apply ---> Save

Goto repo ---> src/main/webapp/index.jsp ---> Change the code accordingly ---> Commit changes ---> Goto Tomcat browser and reload ---> You see changes now (wait for a minute)

Goto Jenkins UI ---> You will see the next build triggered automatically.

This is called Continous Integration and Continous Deployment

Do again somechanges in repo. and look for changes in app browser tab

By Kastro Kiran V

LinkedIn: https://www.linkedin.com/in/kastro-kiran/

YouTube: https://www.youtube.com/playlist?list=PLs-PsDpuAuTdOcZa-

DDgG8KRbtMI XRrC