

# JENKINS ANSWERS

## 1. Write a Multi Stage Jenkins Declarative Pipeline.

Ans:

```
pipeline {  
    agent any  
  
    stages {  
        stage ("MUSTAFA") {  
            steps {  
                echo "This is stage-1"  
            }  
        }  
        stage ("devops") {  
            steps {  
                echo "we are learning devops"  
            }  
        }  
        stage ("aws") {  
            steps {  
                echo "we are learning AWS also"  
            }  
        }  
    }  
}
```

A large, light gray watermark of the Jenkins logo is centered in the background. It features a circular emblem with a stylized 'J' made of yellow and white 3D blocks.

## 2. Write a Jenkins Scripted Pipeline to get the code from GitHub and build the code using maven

Ans:

```
node {  
    stage ("Code") {  
        git "https://github.com/devops0014/one.git"  
    }  
    stage ("Build") {  
        sh 'mvn clean package'  
    }  
}
```

## 3. Explain the following post build actions in Jenkins Pipeline.

Ans:

**FIXED** : This block runs when the current status is success and the previous one was failed or unstable

**REGRESSION**: This block runs when the current status is anything except success but the previous one was successful

**ABORTED**: This block runs when the build process is aborted.

**UNSUCCESSFUL**: This block runs when the current status is anything except success.

**CHANGED**: This block runs when the current status is different than the previous one

## 4. Implement Master-Slave concept using Jenkins and deploy the application in slave server.

### STEPS TO IMPLEMENT JENKINS MASTER-SLAVE:

1. LAUNCH **3 INSTANCES** WITH **KEYPAIR** (PEM FILE)
  - a. MASTER
  - b. SLAVE-1
  - c. SLAVE-3
2. SETUP **JENKINS IN MASTER**
3. install **JAVA11** on slaves (amazon-linux-extras install java-openjdk11 -y)

4. go to manage jenkins >> manage nodes and clouds >> new node --> give node name & select permanent agent

Dashboard > Manage Jenkins > Nodes >

+ New Node

Configure Clouds

Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Name

slave-1

Description

this is for slave-1

Number of executors

3

Remote root directory

/home/ec2-user/jenkins/

Labels

dev

Usage

Only build jobs with label expressions matching this node

Launch method

Launch agents via SSH

Host

172.31.44.108

Credentials

slavekeypair.pem

Show all

ADD CREDENTIALS:

Dashboard > Manage Jenkins > Nodes >

Only build jobs with label expressions matching this node

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

SSH Username with private key

Scope

Global (Jenkins, nodes, items, all child items, etc)

ID

Description

this is for slave-2

Username

ec2-user

Dashboard > Manage Jenkins > Nodes >

Only build jobs with label expressions matching this node

Description ?  
this is for slave-2

Username  
ec2-user

☐ Treat username as secret ?

Private Key  
☒ Enter directly

Key  
Enter New Secret Below  
-----BEGIN RSA PRIVATE KEY-----  
MIIEpAIBAAKCAQEA...  
-----END RSA PRIVATE KEY-----

Passphrase

☐ Disable deferred wipeout on this node ?  
☐ Environment variables

Dashboard > Manage Jenkins > Nodes > slave-1

Host ?  
172.31.44.108

Credentials ?  
ec2-user (this is for slave-1)

Host Key Verification Strategy ?  
Non verifying Verification Strategy

Availability ?  
Keep this agent online as much as possible

Node Properties  
☐ Disable deferred wipeout on this node ?  
☐ Environment variables  
☐ Tool Locations

REST API Jenkins 2.375.2

slavekeypair.pem

Show all x

CLICK ON SAVE

## DEPLOYMENT REQUIREMENTS:

- LAUNCH 2 SERVER (jenkins, prod)
- GET THE CODE FROM THE DEVELOPERS
- SETUP JENKINS IN JENKINS SERVER
- SETUP TOMCAT IN PROD SERVER

## PROCEDURE:

**STEP-1:** LAUNCH 2 INSTANCES WITH 8080 PORT

**STEP-2:** SETUP JENKINS IN SERVER

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

```
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
```

```
amazon-linux-extras install java-openjdk11 -y
```

```
yum install jenkins -y
```

```
systemctl restart jenkins
```

**STEP-3:** FORK THE GITHUB (<https://github.com/devops0014/one.git>)

**STEP-4:** INSTALL GIT IN OUR SERVER

```
yum install git -y
```

**STEP-5:**

CREATE JOB AND INTEGRATE GIT TO JENKINS AND BUILD IT.

ONCE WE BUILD THE JOB, FILES PRESENT IN MASTER BRANCH WILL COMES INTO CI SERVER

**STEP-6:** NEXT STEP IS BUILD THE SOURCE CODE WHICH ARE PRESENT IN CI SERVER. TO BUILD THE WE NEED TO USE MAVEN

INSTALL JAVA-1.8.0 & MAVEN IN OUR SERVER

```
yum install java-1.8.0-openjdk -y
```

```
yum install maven -y
```

**STEP-7:**

**CONFIGURE** THE SAME JOB AND CLICK ON **BUILD STEP** AND SELECT **ADD BUILD STEP**

SELECT **invoke top level maven targer.**

in the goal : **clean package**

SAVE THE JOB AND BUILD.

SO WE WILL GET A WAR FILE IN TARGET FOLDER.

**STEP-8:** SETUP THE TOMCAT SERVER IN PROD SERVER.

1. download tomcat file from dlcdn: `wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.70/bin/apache-tomcat-9.0.70.tar.gz`
2. untar the file: `tar -zxvf apache-tomcat-9.0.70.tar.gz`
3. go to the folder: `cd apache-tomcat-9.0.70/webapps/manager/META-INF`
4. open the context.xml in vim editor and make some change (delete 2 lines (21 and 22 lines))
5. go to three steps back: `cd ../../..`
6. and go to conf folder and open tomcat-user.xml file in vim editor

```
-->
<role rolename="manager-gui" />
<role rolename="manager-script" />
<user username="tomcat" password="123456" roles="manager-gui, manager-script" />
</tomcat-users>
```

7. go to one step back: `cd ..`
8. go to bin folder and execute startup.sh file
9. `./startup.sh`

**STEP-9:** Go to manager apps and it will ask the user name and password enter it

**STEP-10:** go to jenkins dashboard

- install plugin (manage jenkins --> manage plugin --> available plugin --> **deploy to container**)
- after installing the plugin go to our job and select post build actions --> add post build actions.
- select deploy war/ear to container

## Post-build Actions

Deploy war/ear to a container

WAR/EAR files ?  
target/\*.war

Context path ?  
swiggy

Containers  
Add Container ▾

☐ Deploy on failure

Add post-build action ▾

Save Apply

- click on add container(9th version). and add credentials (username & password of tomcat)

Add Credentials

Domain  
Global credentials (unrestricted) ▾

Kind  
Username with password ▾

Scope ?  
Global (Jenkins, nodes, items, all child items, etc)

Username ?  
tomcat

☐ Treat username as secret ?

Password ?  
\*\*\*\*\*

ID ?

Save Cancel

- add tomcat url

Tomcat URL ?  
http://54.241.136.194:8081

Advanced...

Add Container ▾

☐ Deploy on failure

Add post-build action ▾

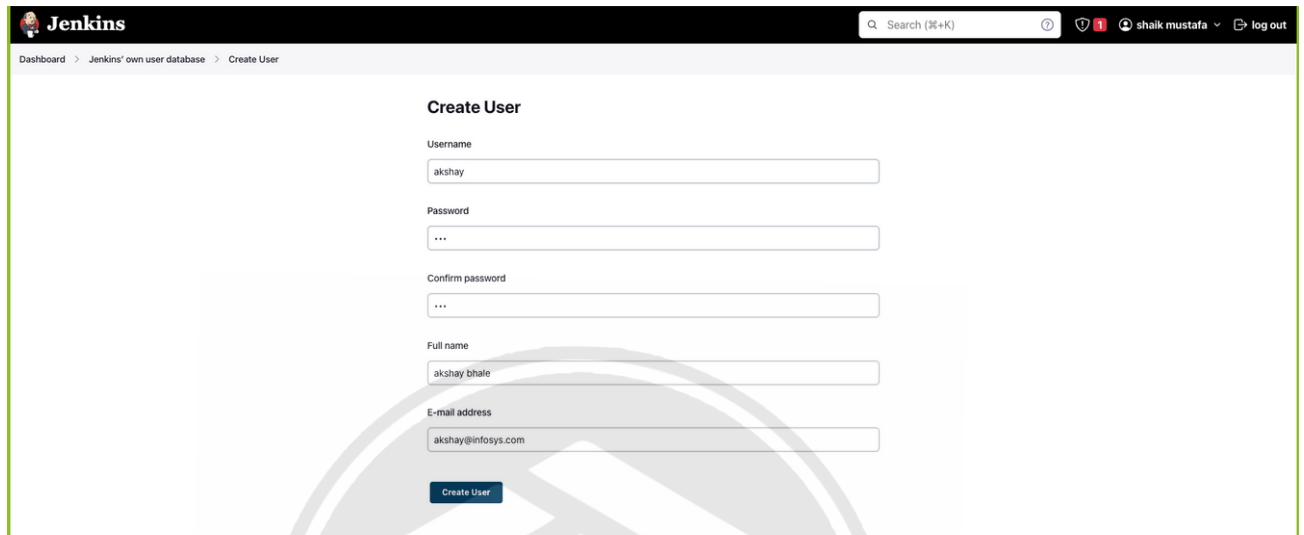
Save Apply

- save and build the job and go to tomcat

you will see swiggy folder. click on the folder you can access the client application.

## 5. Create a user in Jenkins and give build permissions to all the jobs

- INSTALL PLUGIN (role-based authorization strategy). INSTALL WITHOUT RESTART & RESTART JENKINS
- go to manage jenkins and select manage users.
- create user

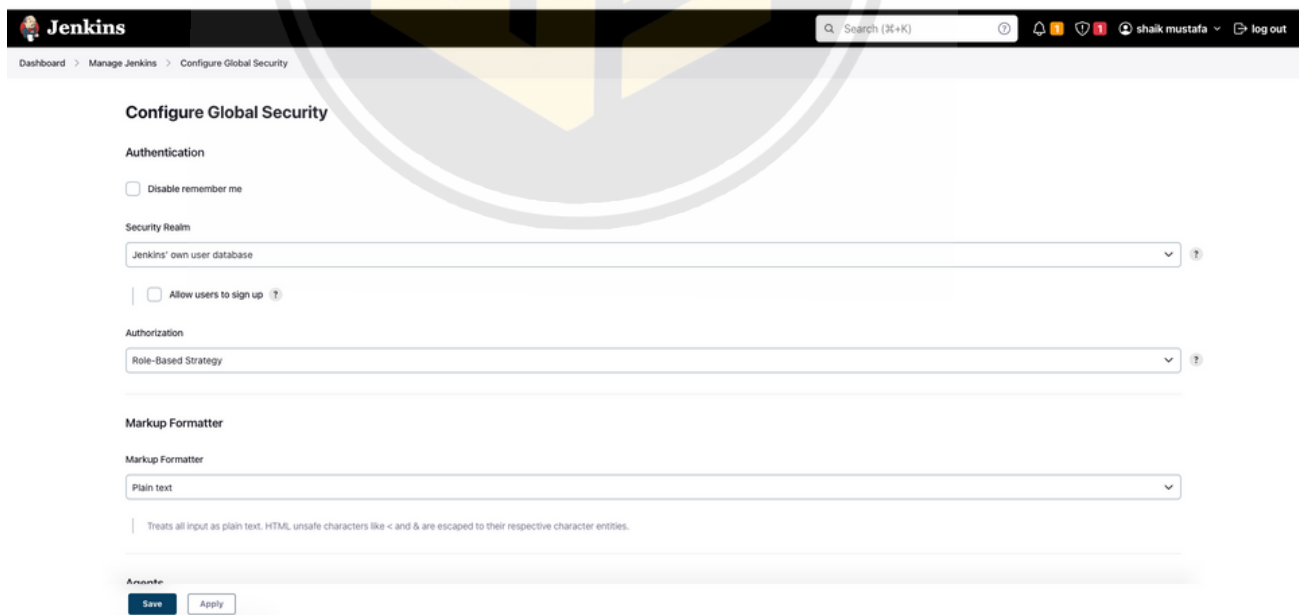


The screenshot shows the Jenkins 'Create User' form. The form is titled 'Create User' and is located under the breadcrumb 'Dashboard > Jenkins' own user database > Create User'. The form contains the following fields:

- Username: akshay
- Password: ...
- Confirm password: ...
- Full name: akshay bhale
- E-mail address: akshay@infosys.com

At the bottom of the form is a 'Create User' button.

- 4. Go to manage jenkins >> config global security and change the authorization to role based strategy and save it



The screenshot shows the Jenkins 'Configure Global Security' page. The page is titled 'Configure Global Security' and is located under the breadcrumb 'Dashboard > Manage Jenkins > Configure Global Security'. The page contains the following sections:

- Authentication**
  - ☐ Disable remember me
  - Security Realm: Jenkins' own user database
  - ☐ Allow users to sign up
- Authorization**
  - Role-Based Strategy
- Markup Formatter**
  - Markup Formatter: Plain text

At the bottom of the page are 'Save' and 'Apply' buttons.

- Go to manage jenkins >> manage and assign roles and select manage roles and select build roles.