

DEPLOYING A WEB APPLICATION USING TOMCAT SERVER

STEP-1: LAUNCH 2 INSTANCES (MASTER & SLAVE) WITH 8080 PORTS WITH KEYPAIR

STEP-2: JENKINS SETUP IN MASTER (install only epel, java11 \$ jenkins)

STEP-3: INSTALL THE PLUGIN (DEPLOY TO WAR/EAR TO CONTAINER)

STEP-4: INSTALL JAVA-OPENJDK11 IN SLAVE

STEP-5: SETUP AN AGENT IN MASTER

STEP-6: INSTALL GIT, JAVA-1.8.0-OPENJDK, MAVEN IN MASTER & SLAVE

(update-alternatives --config java)

STEP-7: GET THE SOURCE CODE FROM THE GITHUB(<https://github.com/devops0014/myweb-01.git>)

STEP-10: GO TO JENKINS>>BUILD STEP>> INVOKE TOP LEVEL MAVEN TARGETS >> CLEAN PACKAGE AND BUILD NOW

STEP-11: GO TO SLAVE AND INSTALL TOMCAT (<https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.70/bin/apache-tomcat-9.0.70.tar.gz>)

- tar -zxvf file_name
- go to apache folder
- vim webapps/manager/META-INF/context.xml

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
      allow="127\.\d+\.\d+\.\d+|:::1|0:0:0:0:0:0:0:1" />
```

Delete these 2 lines (21 & 22)

- vim conf/tomcat-users.xml

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

Add these 3 lines at the end of the file

- ./bin/startup.sh
Tomcat will get starts
Public_ip:8080
Go to manager apps

STEP-12: go to Jenkins >> add build step >> deploy war ear to container

STEP-13: ADD CONTAINER-9, GIVE TOMCAT URL AND TOMCAT CREDENTIALS (USERNAME: tomcat, PASSWORD: tomcat) [target/*.war & /swiggy]

STEP-14 : POST BUILD ACTIONS -- > DEPLOY TO EAR/WAR CONTAINER -- > DETAILS

TERRAFOR4M PROJECT:

STEP-1: SETUP JENKINS SERVE

STEP-2: WRITE THE CODE AND PUSH TO GITHUB

STEP-3: INSTALLING THE TERRAFORM PLUGINS

STEP-4: CREATE PIPELINE

```
pipeline {  
  
    agent any  
  
    stages {  
  
        stage('checkout') {  
  
            steps {  
  
                git 'https://github.com/RAHAMSHAIK007/terrarepo.git'  
  
            }  
  
        }  
  
        stage('terraform init') {  
  
            steps {  
  
                sh ("terraform init");  
  
            }  
  
        }  
  
        stage('terraform Action') {  
  
            steps {  
  
                echo "terraform action from the parameter is -- > ${action}"  
  
            }  
  
        }  
  
    }  
}
```

```
sh ("terraform ${action} --auto-approve");
```

```
}
```

```
}
```

```
}
```

```
}
```