Practice for Lesson 5: Understanding of Parallel Jenkins Jobs and Jenkins Slave on AWS

Practices for Lesson 5

Overview

In these practices, you will learn how to Build and Deploy an Application to Webserver using Jenkins Pipeline. Further create a parallel Agent Pipeline Job on Jenkins.

.

Practice 5-1: Build and Deploy an Application to Webserver using Jenkins Pipeline

Overview

In this practice, you will learn how to Build and Deploy an Application to Webserver using Jenkins Pipeline.

Assumptions

You should have completed the Practice of Lesson 4.

Tasks

1. Sign in to **Jenkins Instance Dashboard**.
   a. In a browser on your local machine, enter the **Public IP** address of the EC2 instance followed by the IP address to sign in to the Jenkins Dashboard (for example, **<Public-IP>:8080**).

   b. Enter the user name and password provided.
   c. You will have access to the Jenkins Dashboard.
2. Install Maven on the **Linux instance** (Production server).
   a. Connect to the **Linux instance** from the **Putty**, run the code to install the **Maven** and **Git** in the server as shown below.

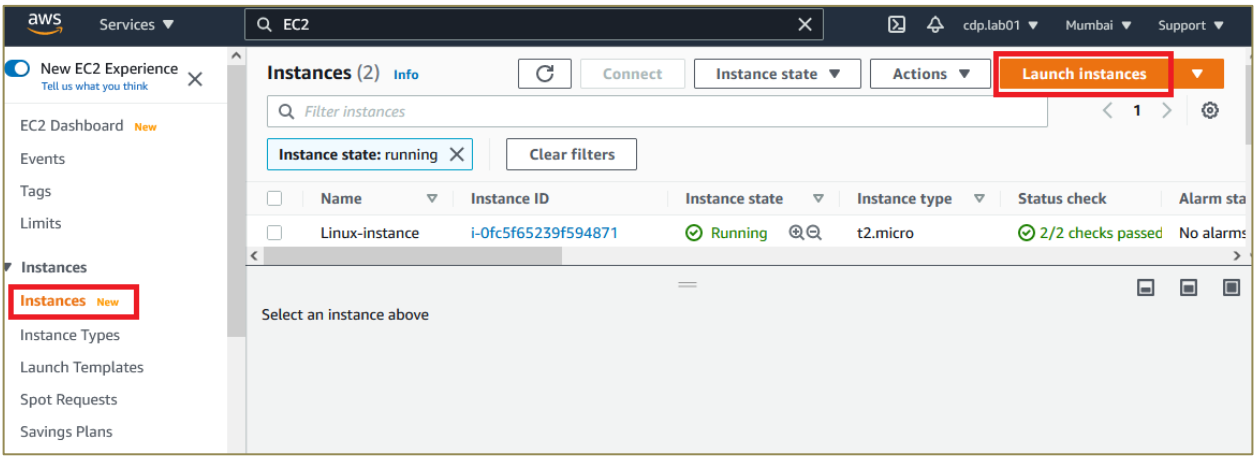
```
[ec2-user@ip-172-31-35-174 ~]$ sudo yum install -y git maven

Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                | 3.7 kB     00:00
amzn2extra-docker                                         | 3.0 kB     00:00
jenkins                                                   | 2.9 kB     00:00
Package git-2.23.4-1.amzn2.0.1.x86_64 already installed and latest version
Resolving Dependencies
--> Running transaction check
---> Package maven.noarch 0:3.0.5-17.amzn2 will be installed
--> Processing Dependency: sisu-inject-plexus for package: maven-3.0.5-17.amzn2.
noarch
--> Processing Dependency: sisu-inject-bean for package: maven-3.0.5-17.amzn2.no
arch
```

b. Check the version of **Git** & **Maven** as shown below.

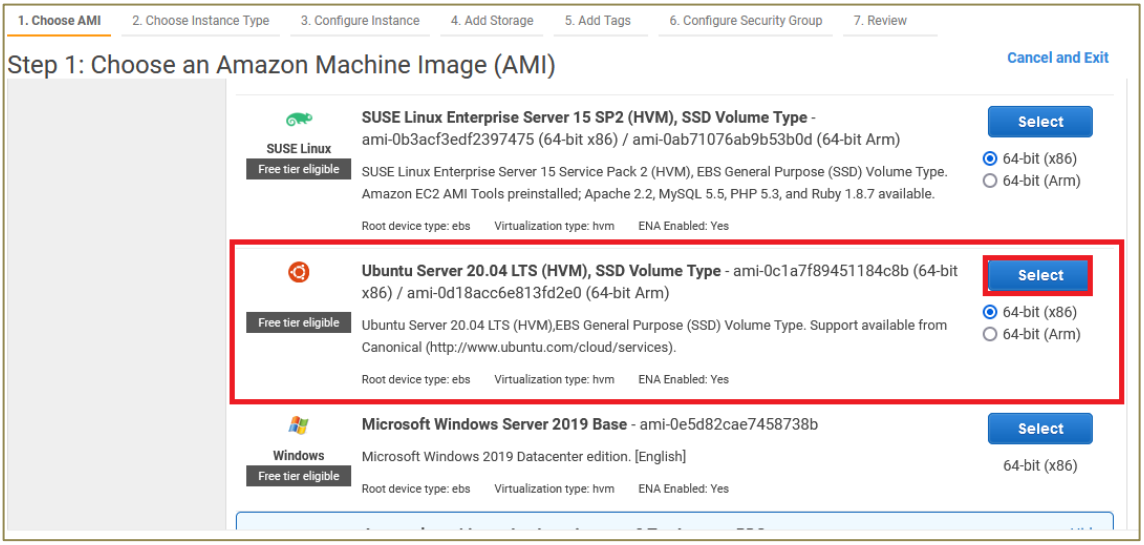
```
[ec2-user@ip-172-31-35-174 ~]$ git --version
git version 2.23.4
[ec2-user@ip-172-31-35-174 ~]$ mvn --version
Apache Maven 3.0.5 (Red Hat 3.0.5-17)
Maven home: /usr/share/maven
Java version: 1.8.0_282, vendor: Red Hat, Inc.
Java home: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.14.232-176.381.amzn2.x86_64", arch: "amd64", famil
y: "unix"
[ec2-user@ip-172-31-35-174 ~]$
```

3. Create Webserver Instance in the AWS Console.

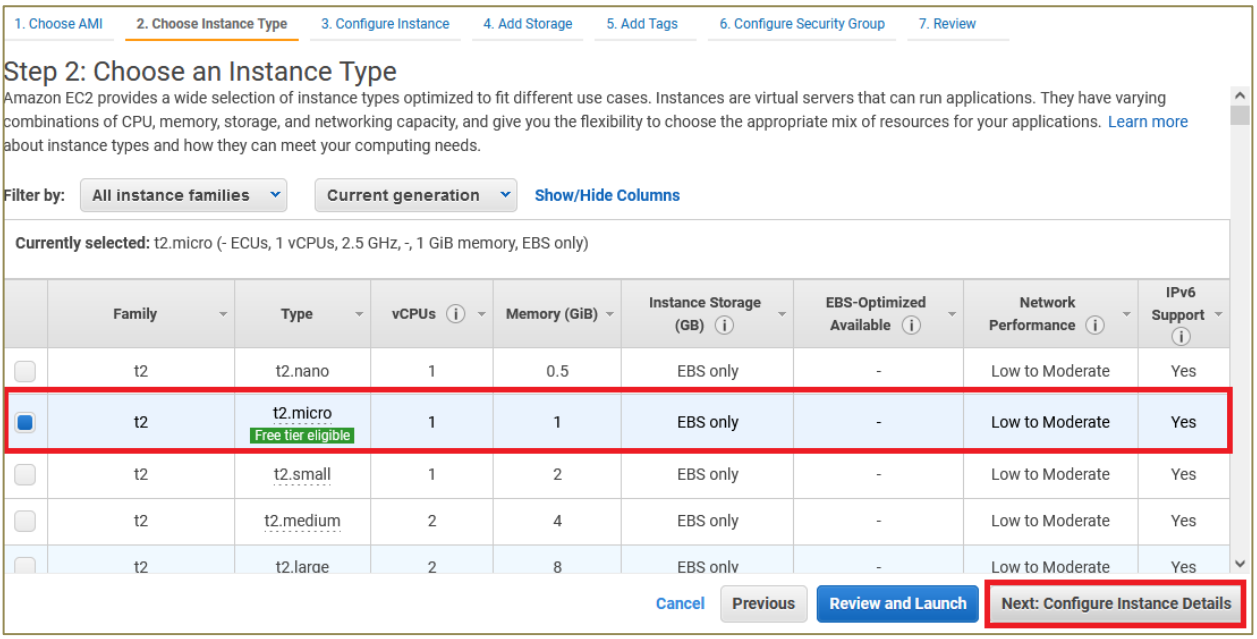
a. Open AWS console and navigate to **EC2** as shown below, select **Launch Instance** to create an instance to the Webserver.



b. In “Step 1: Choose an Amazon Machine Image (AMI)”, select **Ubuntu Server** 64-bit (x86) as shown below.



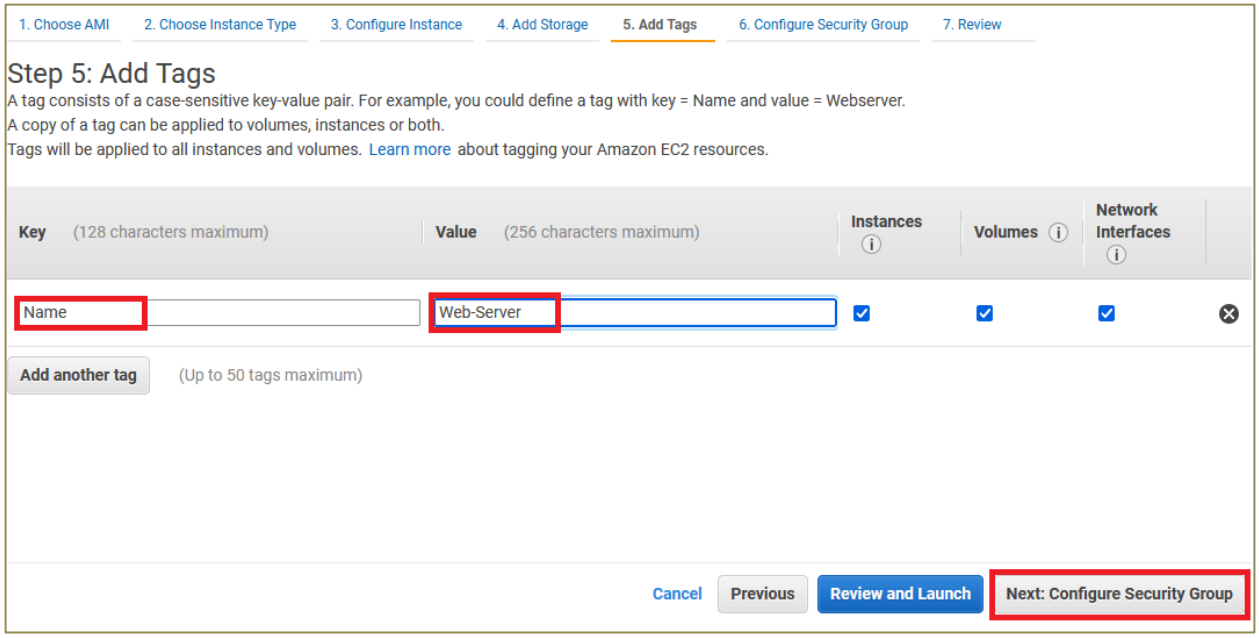
c. In “Step 2: Choose an Instance Type”, select instance size **t2.micro** and click **Next: Configure Instance Details** as shown below.



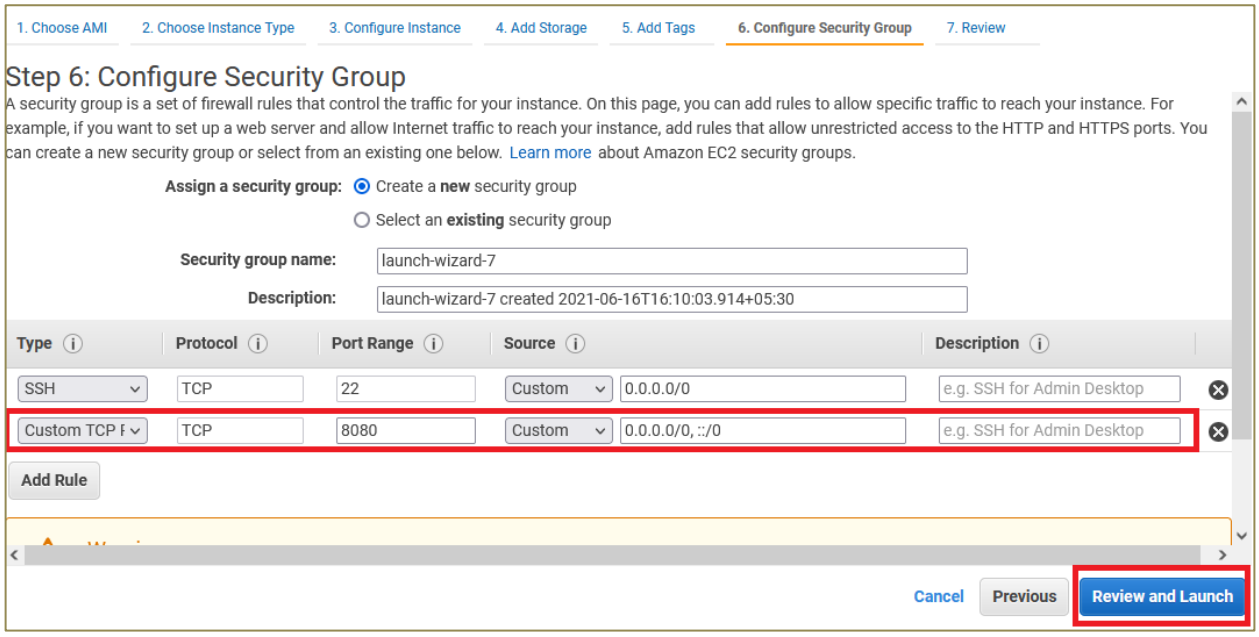
d. In “Step 3: Configure instance” accept the default values and click **Next**.

e. In “Step 4: Add Storage”, accept the default values and click **Next: Add Tags**.

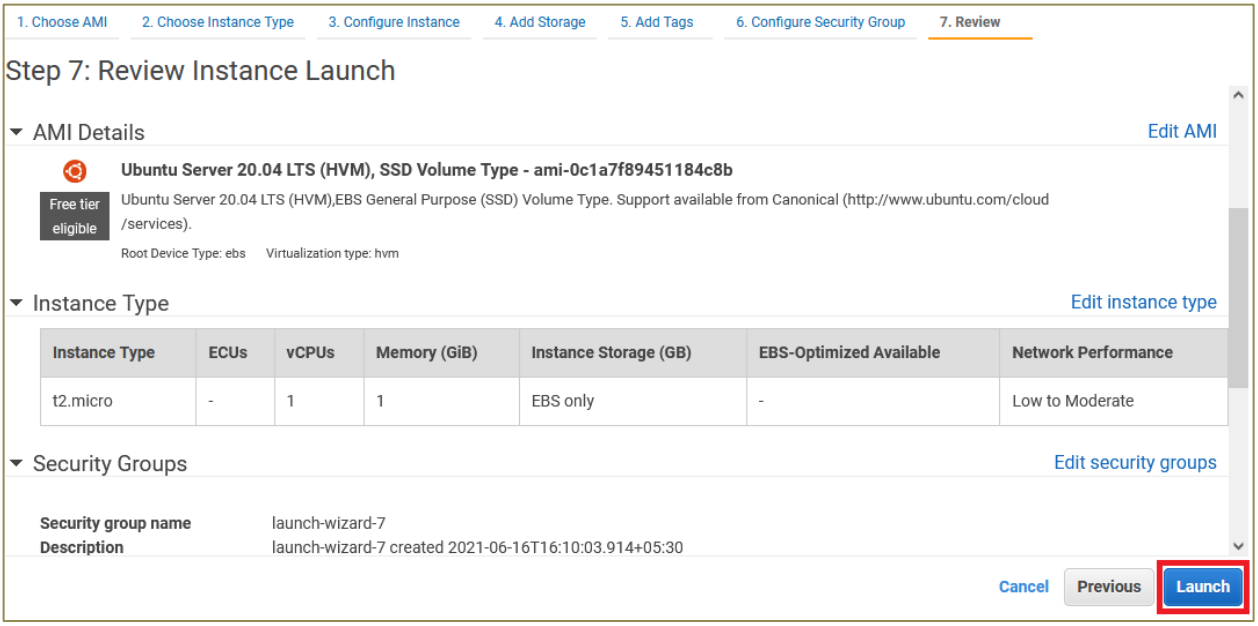
f. In “Step 5: Add Tags”, click **Add Tag** to provide the **Key** as **Name** and **Value** as **Web-Server** and click **Next: Configure Security Group** as shown below.



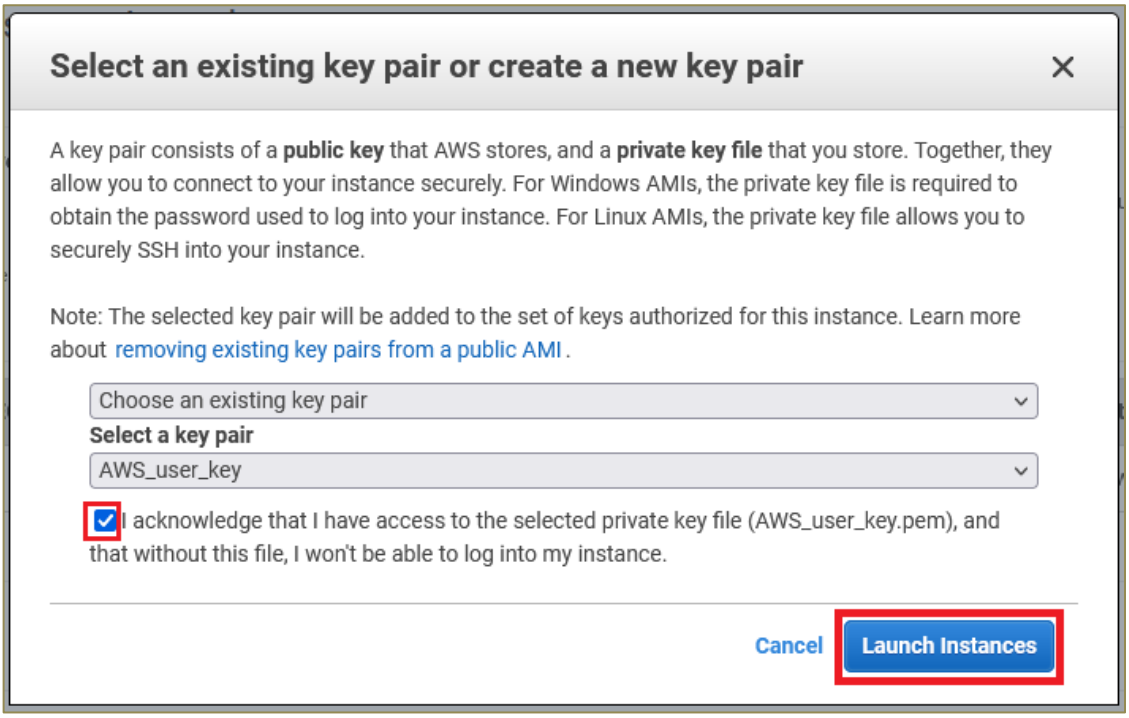
g. In “Step 6: Configure Security Group” click **Add Rule** to open the ports as shown below. Click **Review and Launch.**



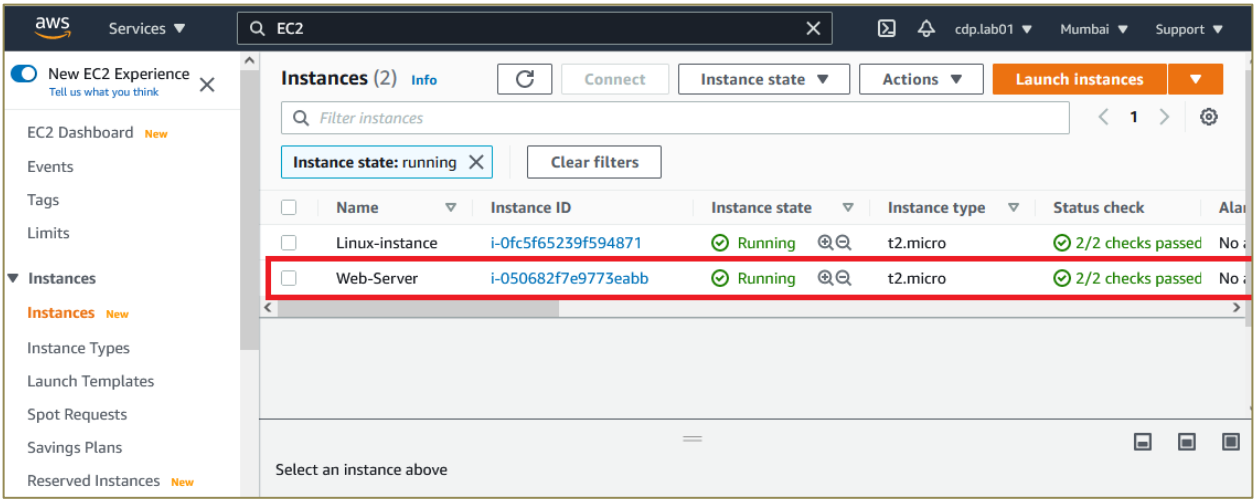
h. In “Step 7: Review Instance Launch”, review the details of the instance and click **Launch** as shown below.



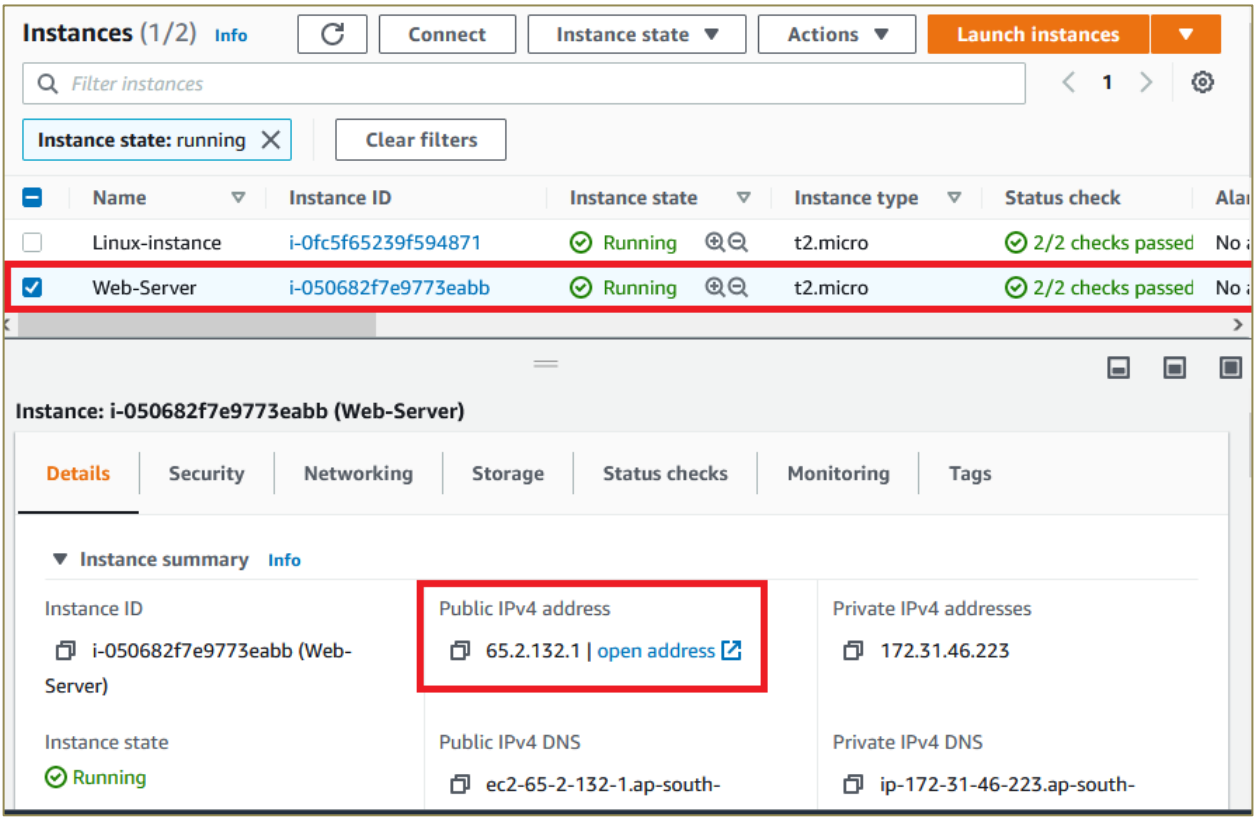
i. Select from the drop down **Choose an existing key pair**, select the **Key Pair**. Click **Launch Instances** as shown below.



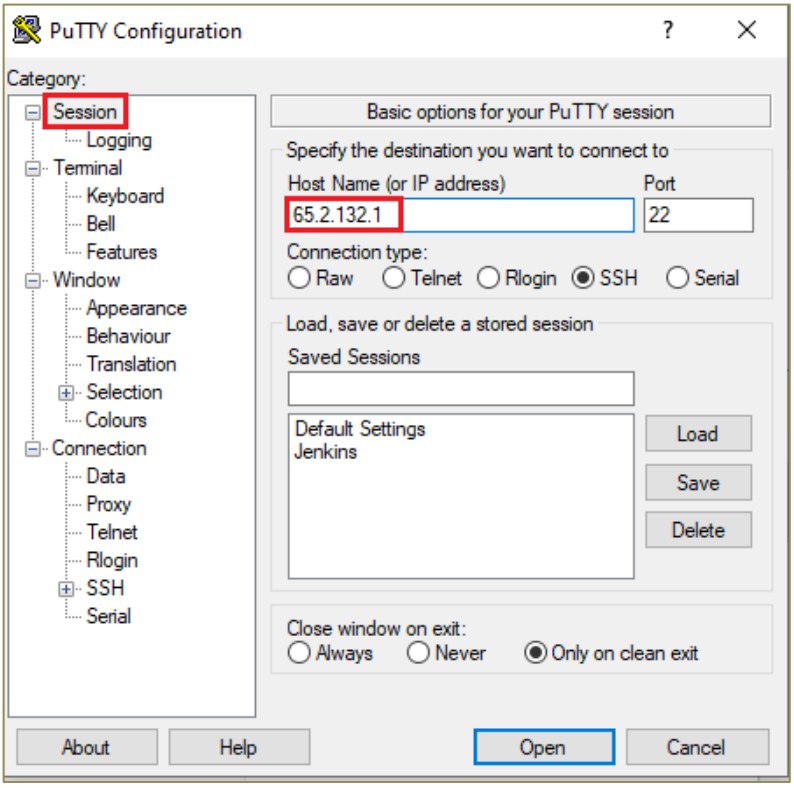
j. The AWS EC2 **Web-Server** is created successfully and its status is **running** as shown below.



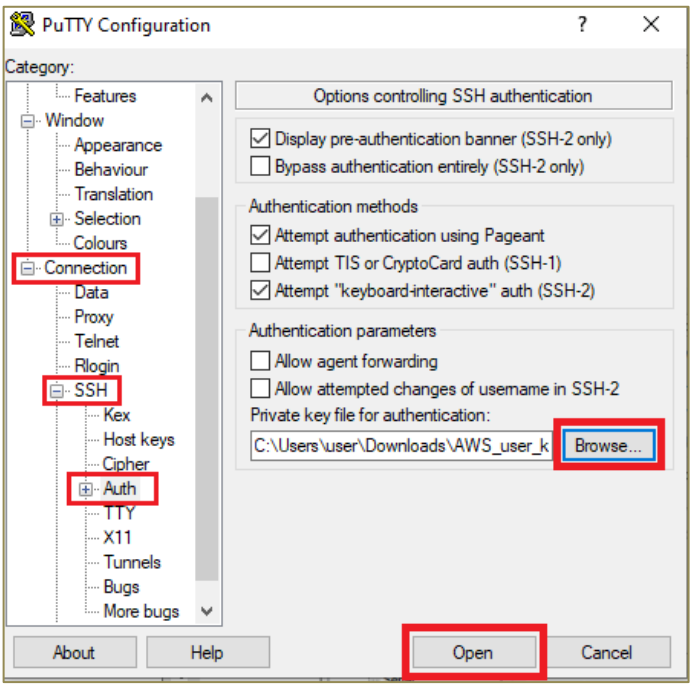
k. To connect to the Web-Server instance copy the public IP address as shown below.



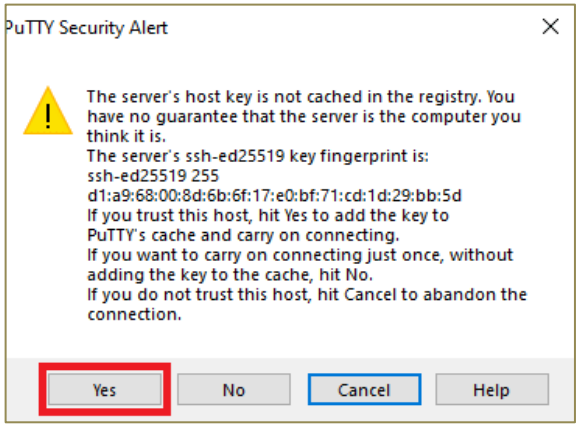
l. Open Putty Configuration, navigate to **Session** and paste the instance public IP address in the **Host Name** as shown below.



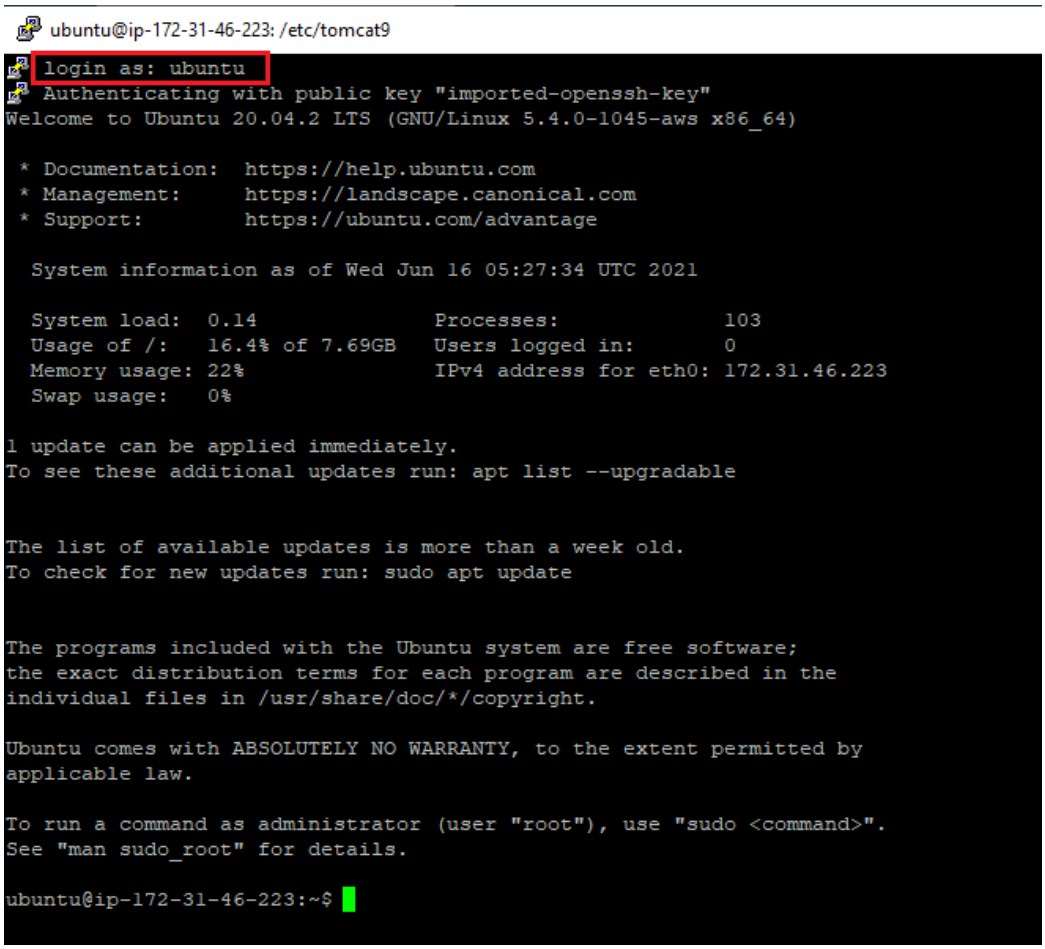
m. Navigate to **Connection**, go to **SSH** and select **Auth**. Click **Browse** to add .ppk file and click **Open** as shown below.



n. Putty connects to the AWS **Web_Server** instance and a Putty Security Alert pop-up displays as shown below. Click **Yes**.



o. In the terminal window, for **login as:** type **ubuntu** and the login is successful as shown below.



p. Run the command to Update the apt repository as shown below.

```
ubuntu@ip-172-31-46-223:~$ sudo apt-get update
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease [265 kB
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
[114 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelea
[101 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [7
kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Pa
ges [8628 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [1
kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metad
 [7780 B]
Get:9 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packa
 [251 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/restricted Translatio
n [36.5 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 c-n-
etadata [456 B]
Get:12 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packag
[588 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe Transla
```

q. Run the command to Install **Tomcat** in the **Web-Server** as shown below.

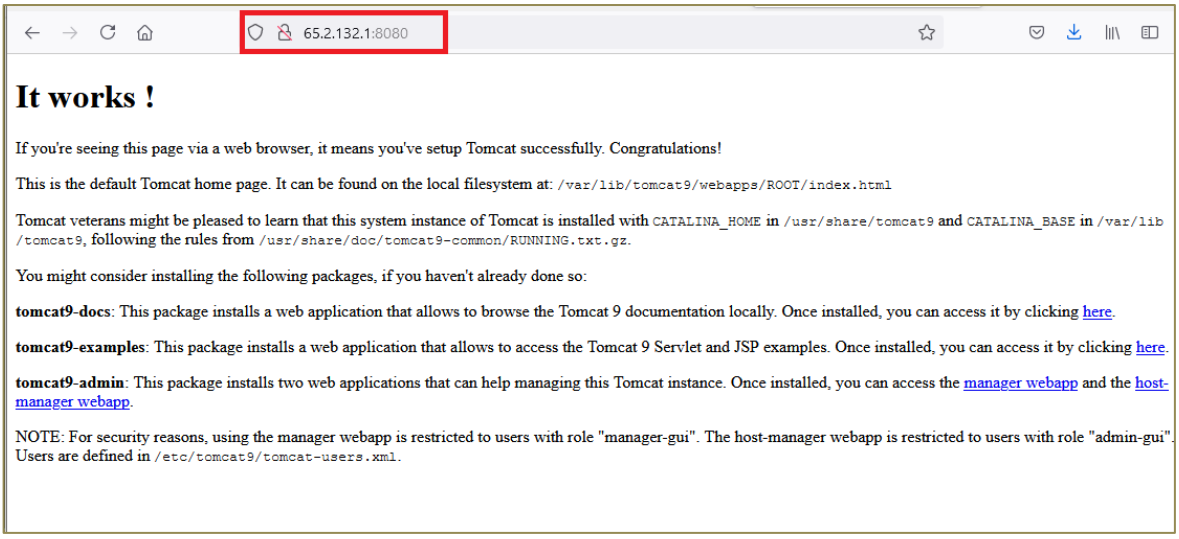
```
ubuntu@ip-172-31-46-223:~$ sudo apt-get install -y tomcat9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java default-jre-headless fontconfig-config fonts-dejavu-core java-co
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2 libeclipse-jdt-core-j
  libfontconfig1 libgraphite2-3 libharfbuzz0b libjpeg-turbo8 libjpeg8 liblcms2-2 libnsp
  libpcsclite1 libtcnative-1 libtomcat9-java openjdk-11-jre-headless tomcat9-common
Suggested packages:
  default-jre cups-common liblcms2-utils pcscd libnss-mdns fonts-dejavu-extra fonts-ipa
  fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei fonts-indic tomcat9-admin
  tomcat9-examples tomcat9-user
The following NEW packages will be installed:
  ca-certificates-java default-jre-headless fontconfig-config fonts-dejavu-core java-co
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2 libeclipse-jdt-core-j
  libfontconfig1 libgraphite2-3 libharfbuzz0b libjpeg-turbo8 libjpeg8 liblcms2-2 libnsp
  libpcsclite1 libtcnative-1 libtomcat9-java openjdk-11-jre-headless tomcat9 tomcat9-co
0 upgraded, 25 newly installed, 0 to remove and 60 not upgraded.
Need to get 53.2 MB of archives.
After this operation, 197 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 java-common all
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libavahi-common-
```

r. Install the one more package for the **Tomcat** Admin as shown below.

```
ubuntu@ip-172-31-46-223:~$ sudo apt-get install -y tomcat9-admin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tomcat9-admin
0 upgraded, 1 newly installed, 0 to remove and 60 not upgraded.
Need to get 24.6 kB of archives.
After this operation, 189 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe
.31-1ubuntu0.1 [24.6 kB]
Fetched 24.6 kB in 1s (25.6 kB/s)
Selecting previously unselected package tomcat9-admin.
(Reading database ... 61123 files and directories currently installed.)
Preparing to unpack .../tomcat9-admin_9.0.31-1ubuntu0.1_all.deb ...
Unpacking tomcat9-admin (9.0.31-1ubuntu0.1) ...
Setting up tomcat9-admin (9.0.31-1ubuntu0.1) ...
ubuntu@ip-172-31-46-223:~$
```

s. To verify the Tomcat installation, copy the Public IP of the Web-Server instance and paste in the browser with the port number 8080 as shown below.

Syntax: **<Public-IP Address>:8080**



4. Setting the path of Tomcat in Jenkins.

a. Connect to the **Web-Server** and execute the command to navigate to the Tomcat directory to list the files as shown below.

```
ubuntu@ip-172-31-46-223:~$ cd /etc/tomcat9/
ubuntu@ip-172-31-46-223:/etc/tomcat9$ ls
Catalina             context.xml          logging.properties  server.xml        web.xml
catalina.properties  jaspic-providers.xml policy.d            tomcat-users.xml
ubuntu@ip-172-31-46-223:/etc/tomcat9$
```

b. Open the file **tomcat-users.xml** using the vi editor as shown below.

```
ubuntu@ip-172-31-46-223:/etc/tomcat9$ sudo vi tomcat-users.xml
```

c. Navigate to the end of the file and add the statement given below add the username, password and roles as shown below.

```
<user username="training" password="freefree" roles="manager-script,manager-status,manager-gui"/>
```

```
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
<!--
  NOTE:  By default, no user is included in the "manager-gui" role required
  to operate the "/manager/html" web application.  If you wish to use this app,
  you must define such a user - the username and password are arbitrary. It is
  strongly recommended that you do NOT use one of the users in the commented out
  section below since they are intended for use with the examples web
  application.
-->
<!--
  NOTE:  The sample user and role entries below are intended for use with the
  examples web application. They are wrapped in a comment and thus are ignored
  when reading this file. If you wish to configure these users for use with the
  examples web application, do not forget to remove the <!.. ..> that surrounds
  them. You will also need to set the passwords to something appropriate.
-->
<!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <user username="role1" password="<must-be-changed>" roles="role1"/>
-->
<user username="training" password="freefree" roles="manager-script,manager-status,manager-gui"/>
</tomcat-users>
-- INSERT --                                                          44,97          B
```

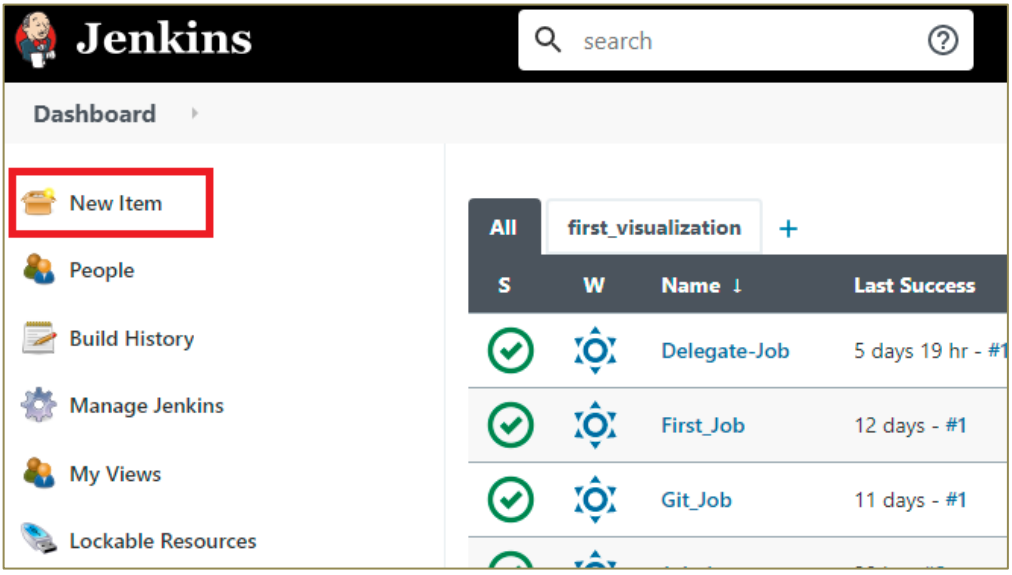
d. Save and Quit the file by pressing ecs key and type **:wq** and press enter.

e. Run the code to restart the service as shown below.

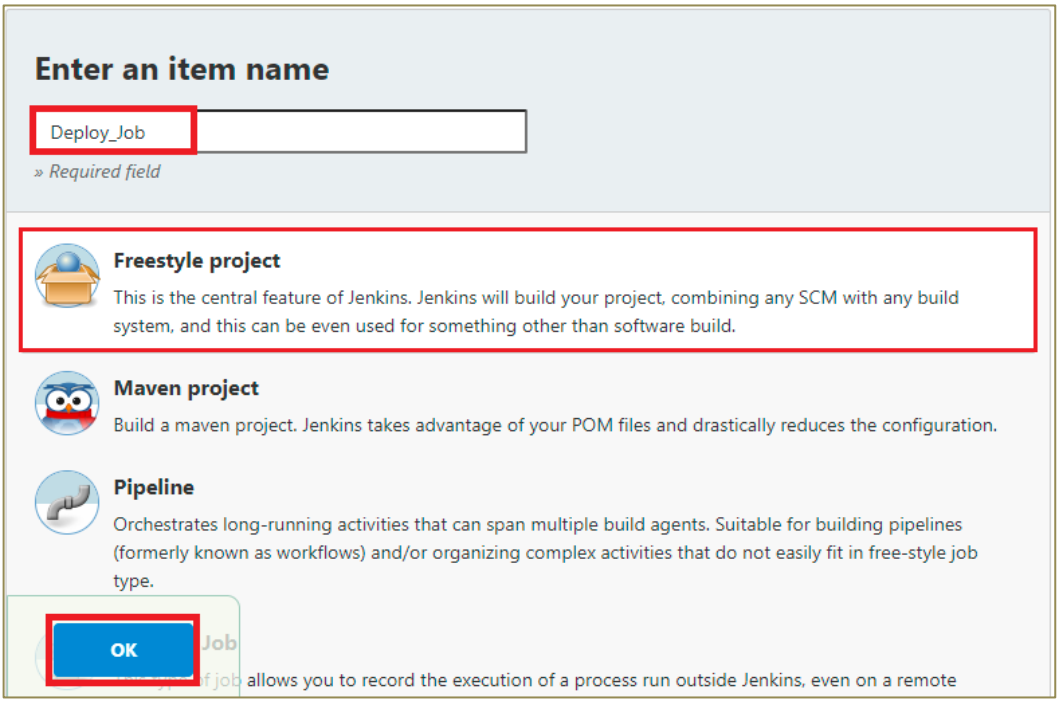
```
ubuntu@ip-172-31-46-223:/etc/tomcat9$ sudo service tomcat9 restart
ubuntu@ip-172-31-46-223:/etc/tomcat9$
```

5. Continuous Download START CI-CD.

a. In the Jenkins Dashboard, navigate to main menu and select **New Item** to create a Deployment Job as shown below.

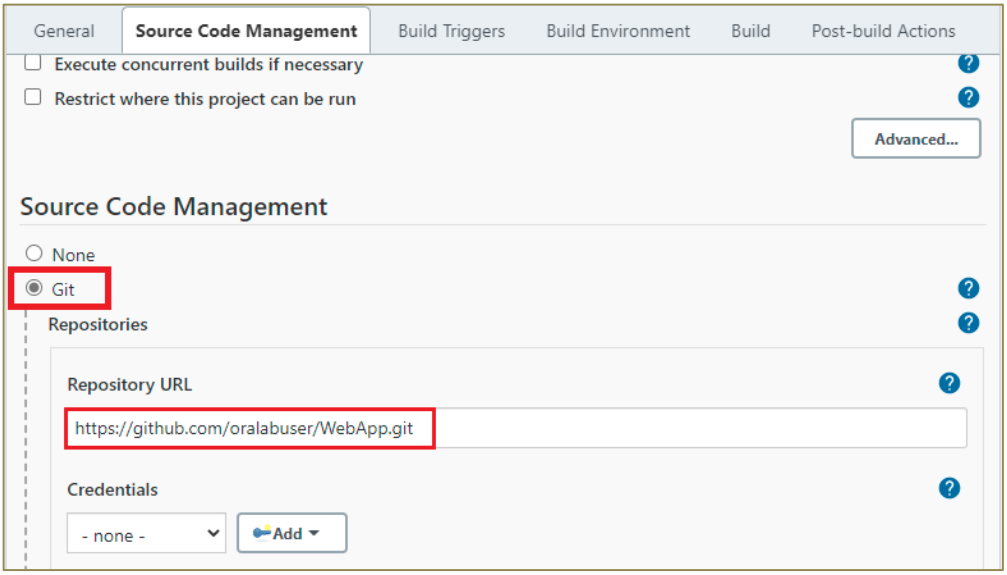


b. Provide the name for Job, select **Freestyle project** and click **OK**.

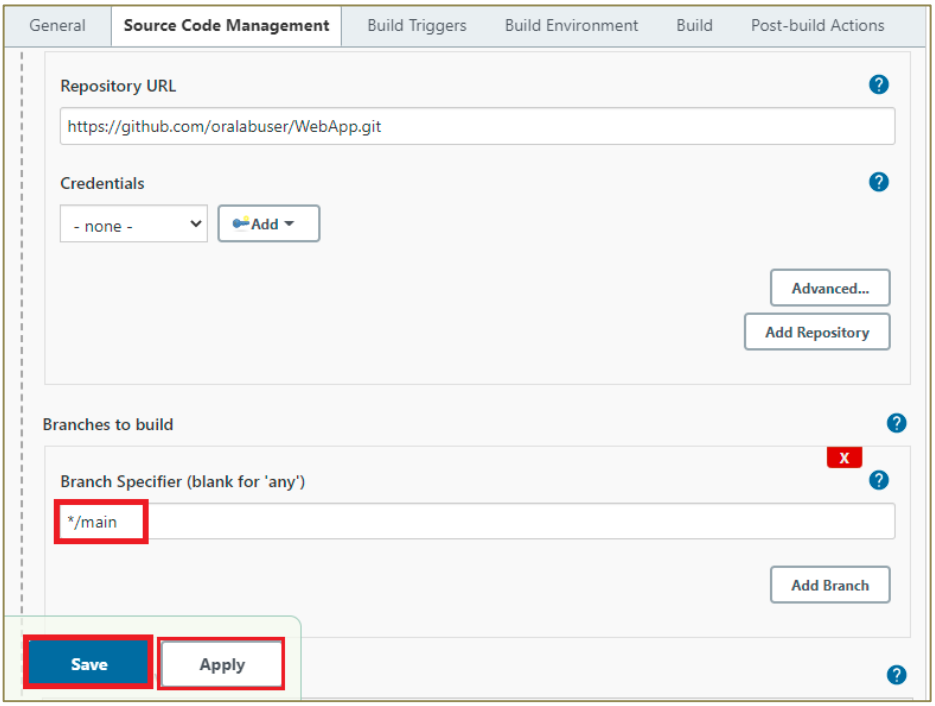


c. Navigate to **Source Code Management** and select **Git**. Enter the URL of the **GitHub** repository given below.

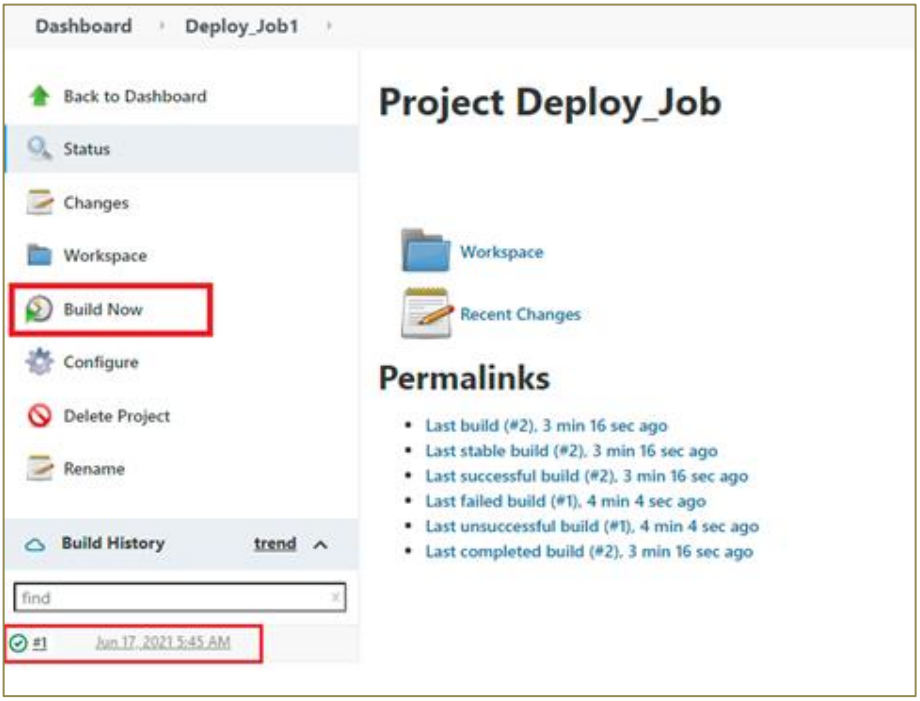
```
https://github.com/oralabuser/WebApp.git
```



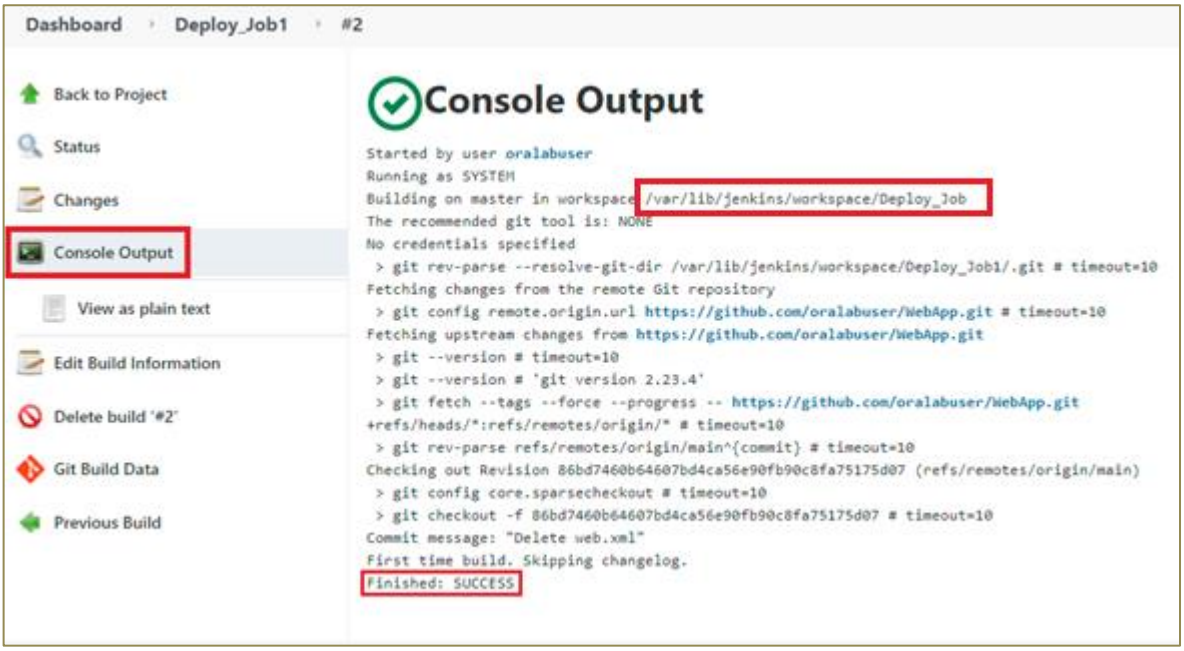
d. Change the **Branch Specifier** has **main** as shown below. Click **Apply** and **Save**.



e. Job is created successfully, click **Build Now** to execute the pipeline. Click on build created under **Build History** as shown below.



f. In the **Build** page, click **Console Output** to view the output of the job. The path is provided where the code is downloaded.

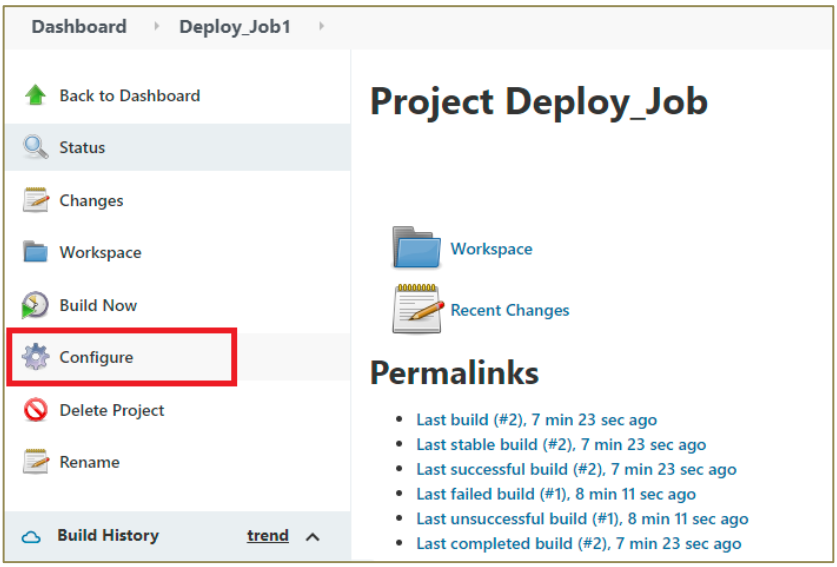


g. In the terminal navigate to the respective path provided and list the files to verify as shown below.

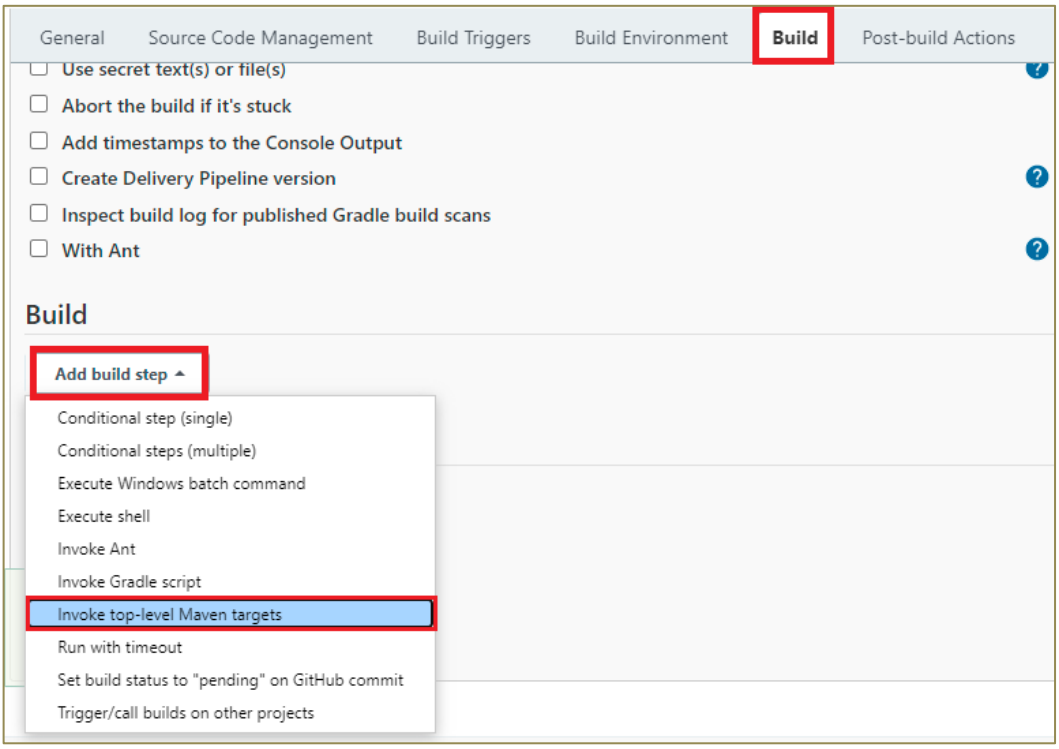
```
[ec2-user@ip-172-31-35-174 ~]$ cd /var/lib/jenkins/workspace/Deploy_Job
[ec2-user@ip-172-31-35-174 Deploy_Job]$ ls
DeclarativePipeline_Jenkinsfile  pom.xml                          server
git2                             README.md                        webapp
Jenkinsfile                      ScriptedPipeline_Jenkinsfile.txt
[ec2-user@ip-172-31-35-174 Deploy_Job]$
```

6. Continuous Build process using the Jenkins Job.

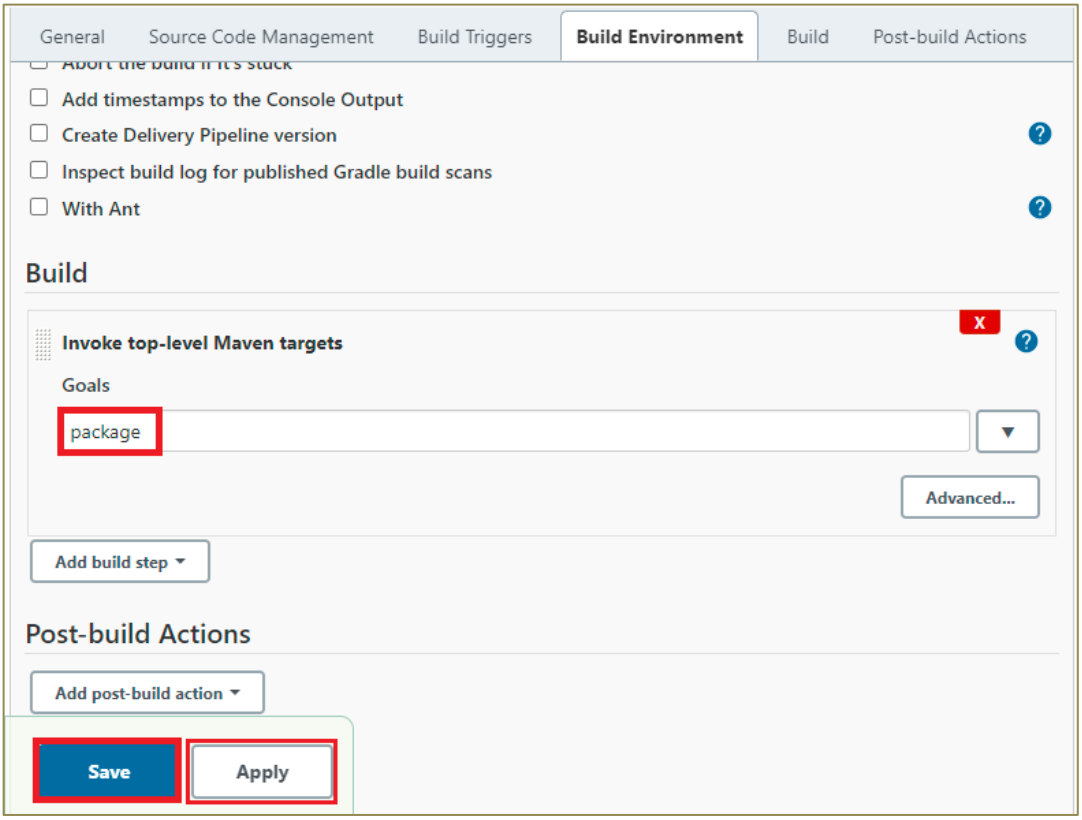
a. Click on **Configure** of the same job **Deploy_Job** as shown below.



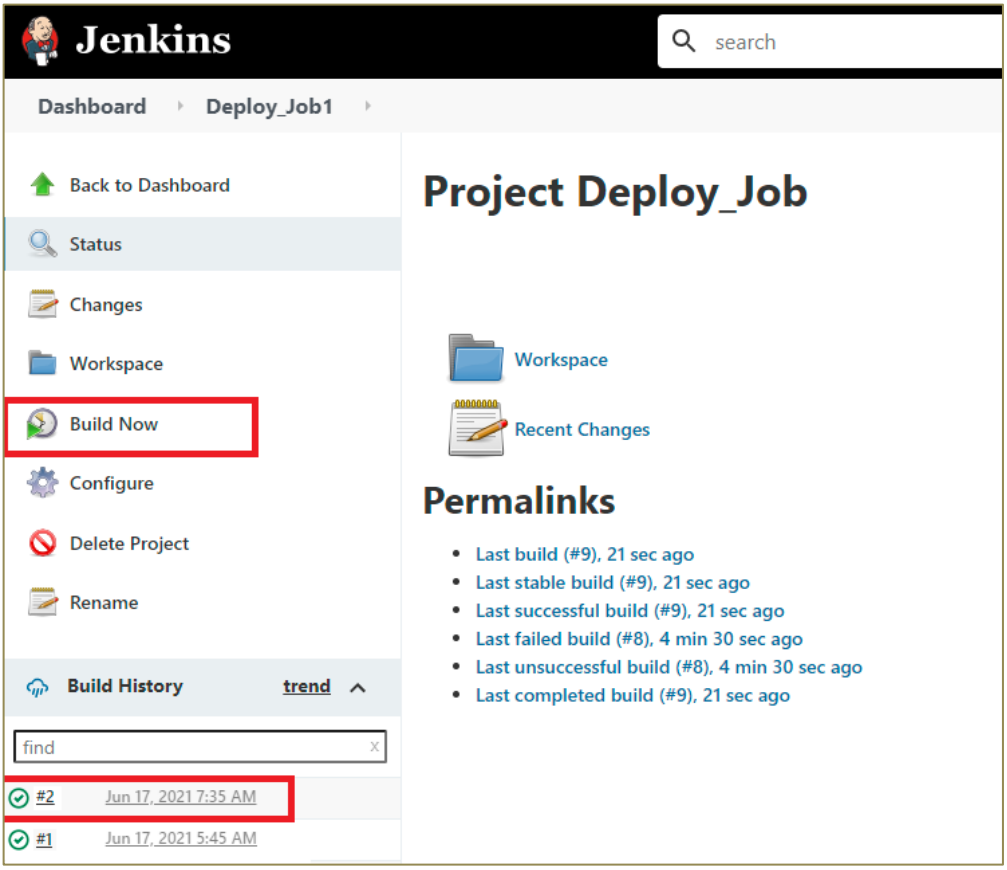
b. Navigate to **Build** section, click **Add build step** and select **Invoke Top-level Maven targets** as shown below.



c. Enter the **Goals** as **package**, click **Apply** and **Save** to update the configuration of the job as shown below.



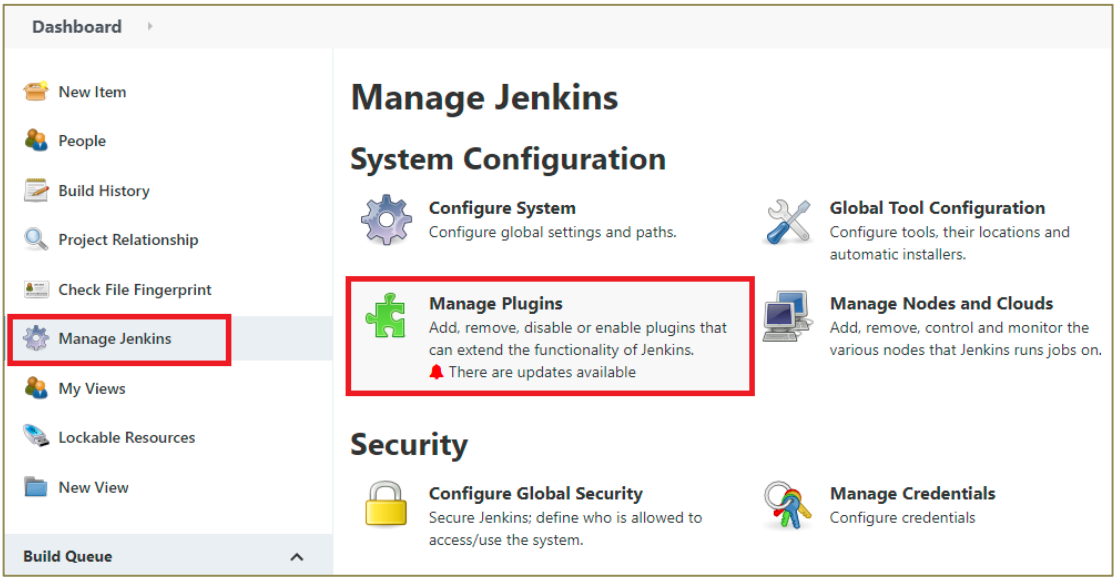
d. Select **Build Now** to build the job again in Jenkins, click on the new build created under **Build History** as shown below



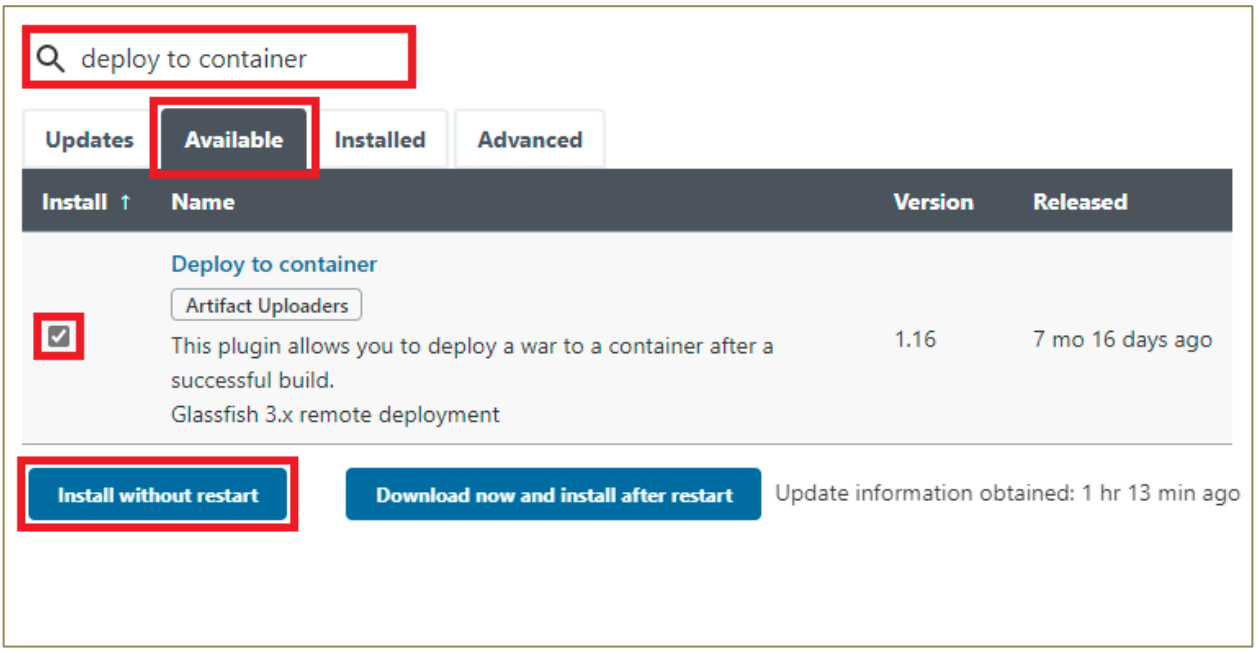
e. Click **Console Output** to view the output of the job. The Console output is displayed as shown below.

```
Results :

Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-war-plugin:2.1.1:war (default-war) @ webapp ---
[INFO] Packaging webapp
[INFO] Assembling webapp [webapp] in [/var/lib/jenkins/workspace/Deploy_Job1/webapp/target/webapp]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/Deploy_Job1/webapp/src/main/webapp]
[INFO] Webapp assembled in [34 msecs]
[INFO] Building war: /var/lib/jenkins/workspace/Deploy_Job1/webapp/target/webapp.war
[INFO] WEB-INF/web.xml already added, skipping
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] Maven Project ..................................... SUCCESS [0.002s]
[INFO] Server ............................................ SUCCESS [3.585s]
[INFO] Webapp ............................................ SUCCESS [0.933s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 4.707s
[INFO] Finished at: Thu Jun 17 07:35:56 UTC 2021
[INFO] Final Memory: 15M/38M
[INFO] ------------------------------------------------------------------------
Finished: SUCCESS
```

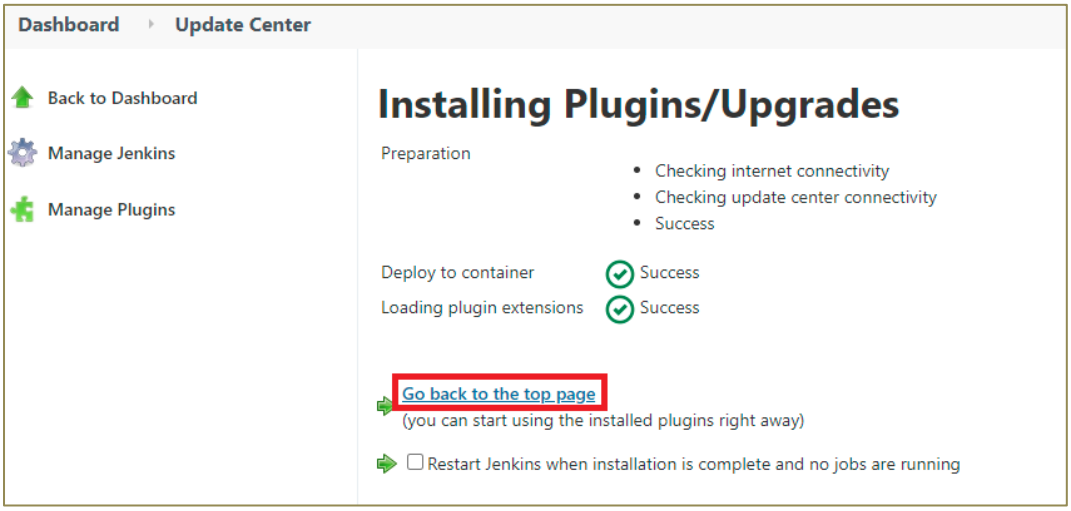
7. Continuous Deployment process using the Jenkins Job.
   a. To deploy the war file into the Web-Server first install the "**deploy to container**" plugin. Navigate to Dashboard, select **Manage Jenkins** and click **Manage Plugins** as shown below.



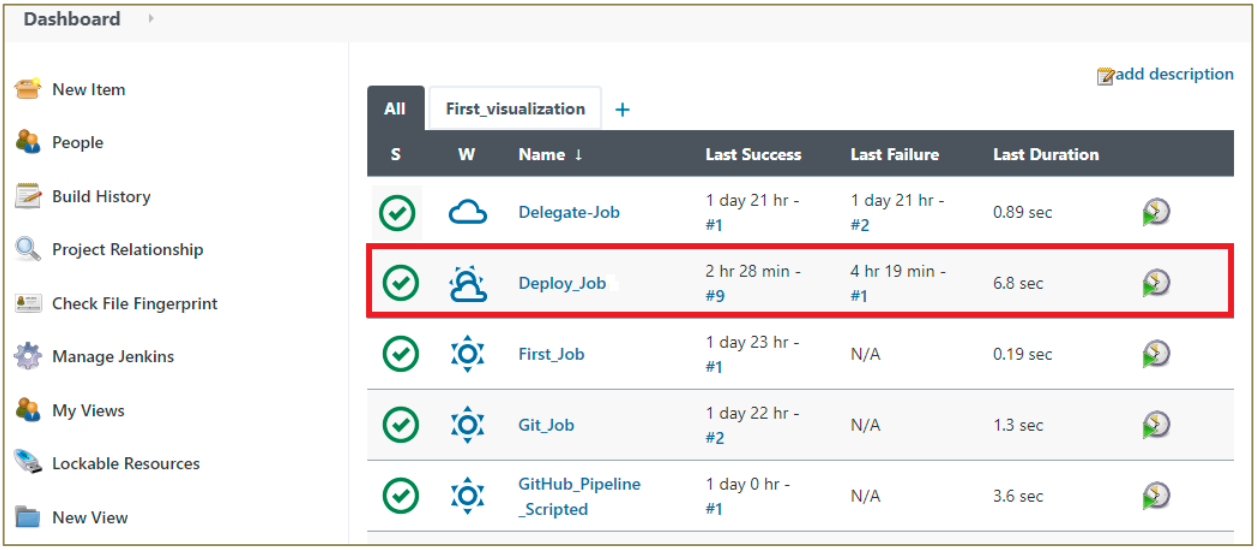
b. Select **Available,** search for “Deploy to container” and select the check box of the plugin. Click **Install without restart.**



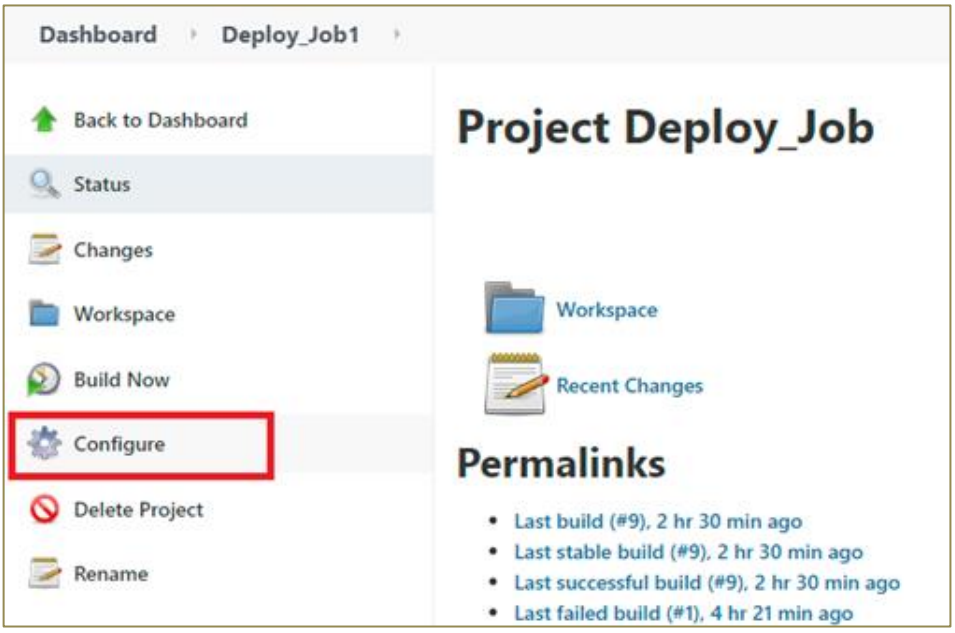
c. The installation process will proceed and Success message will be displayed as shown below.



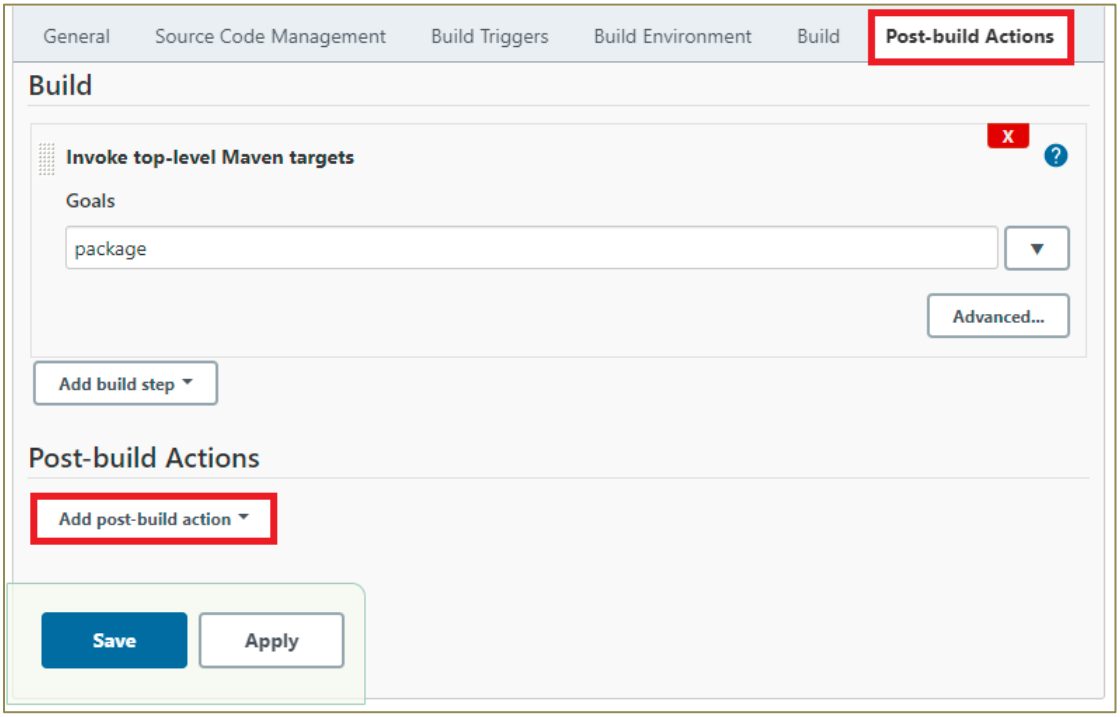
d. Navigate to **Dashboard** and select **Deploy_Job** to configure the continues deployment in the job as shown below.



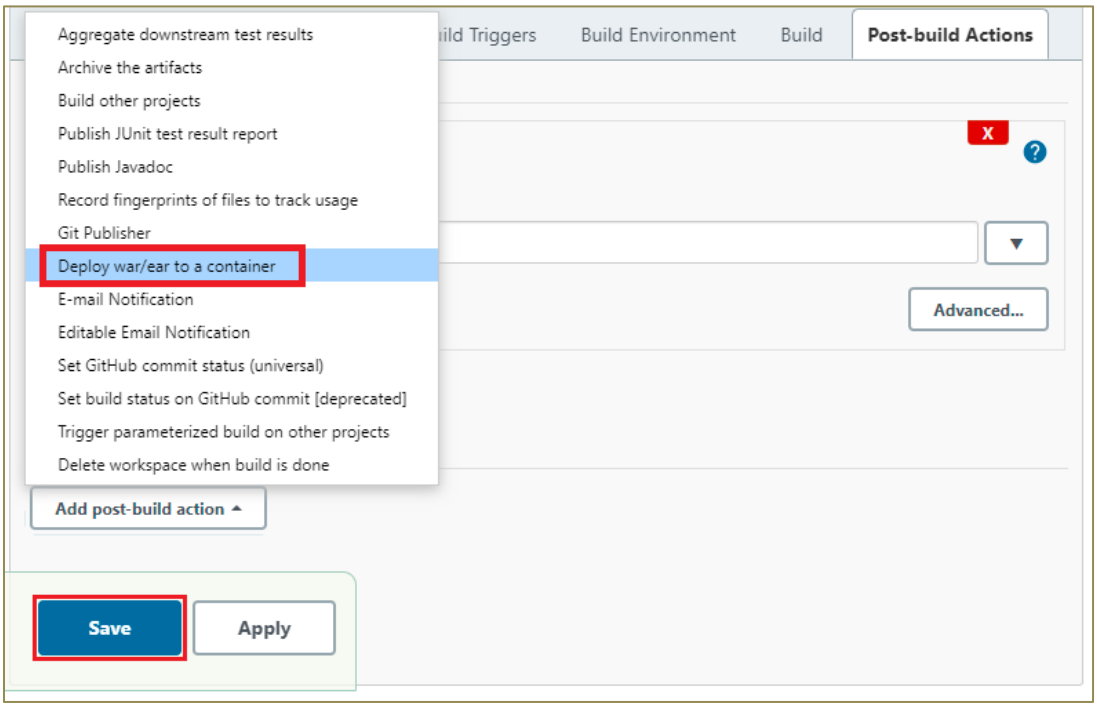
e. Select **Configure** to configure the Job.



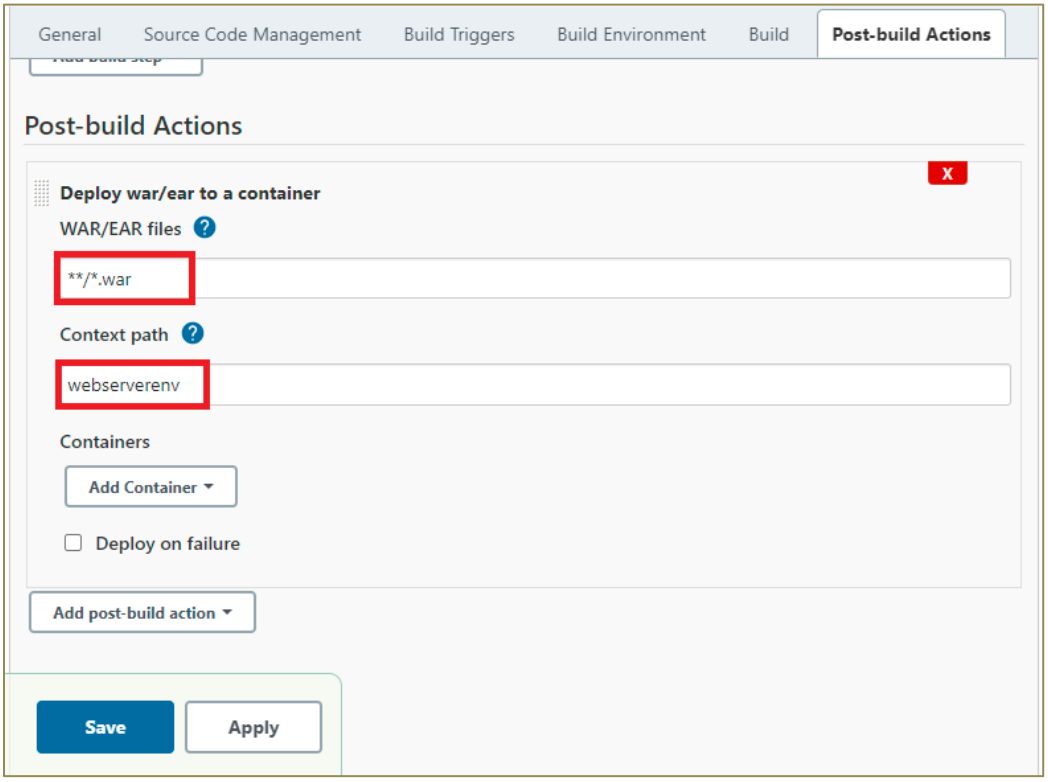
f. Navigate to **Post-build Actions** and select **Add post-build actions** as shown below



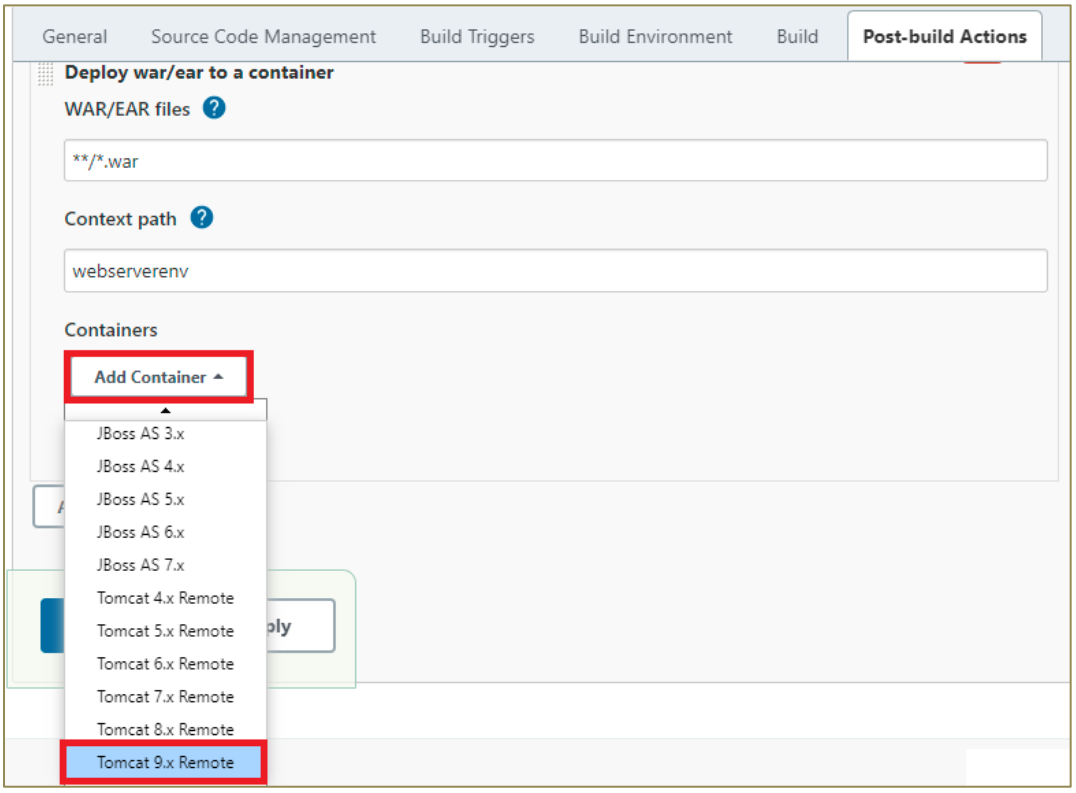
g. Select **Deploy war/ear to container** as shown below.



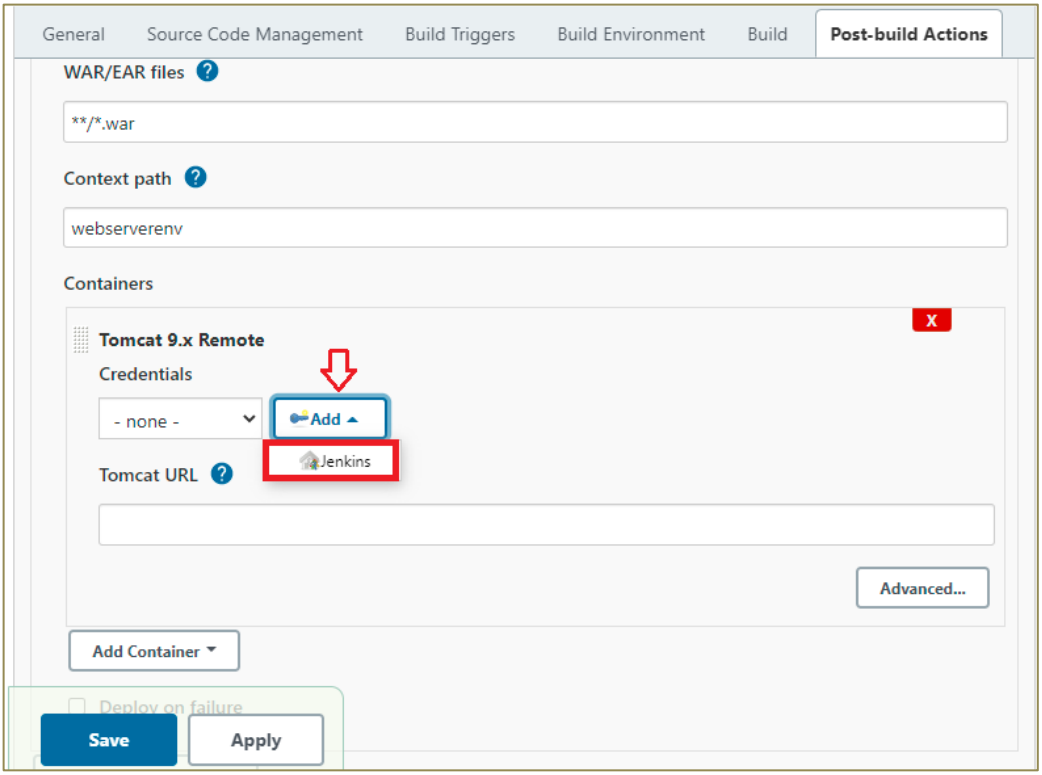
h. Enter the path of the war file or provide the ****/*.war** in **war/ear files** and provide the **Context path** as shown below.



i. Select **Containers** as **Tomcat 9** (installed Tomcat version in server).

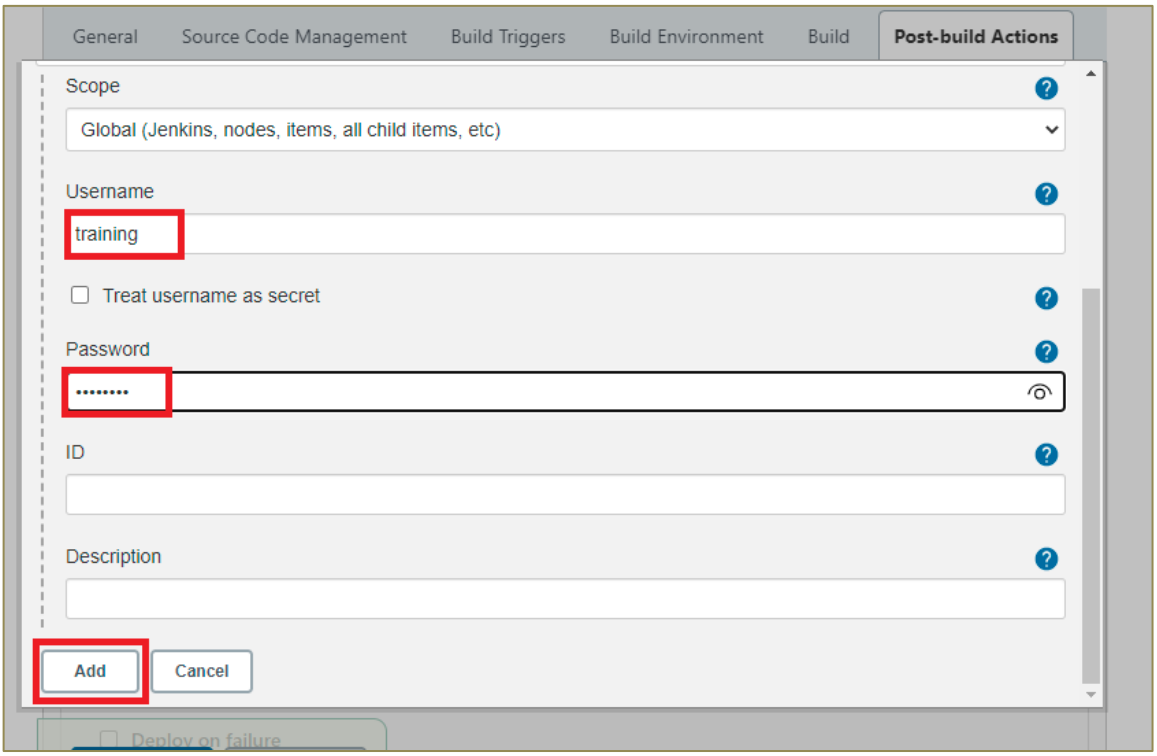


j. To provide **credentials** click **Add** and select **Jenkins** as shown below.

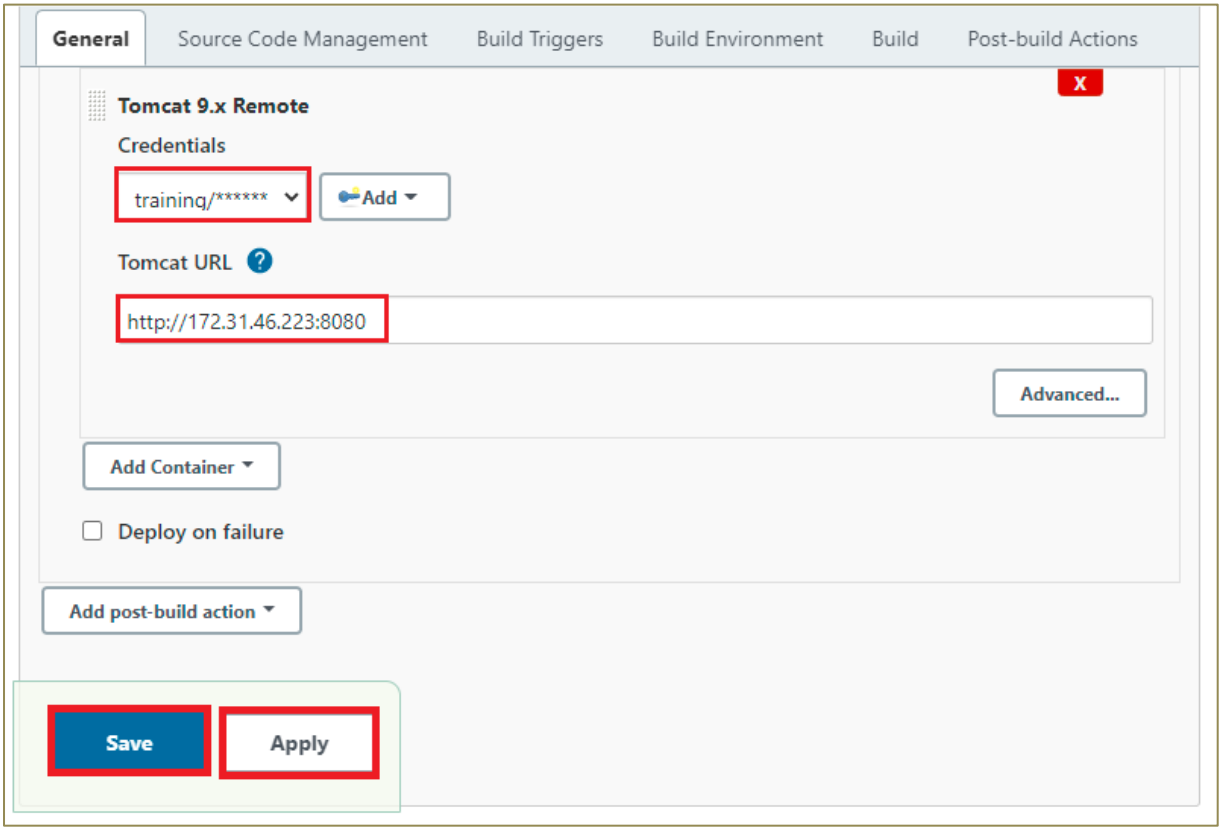


k. Provide the Tomcat **Username** and **Password** as shown below. Click **Add**.

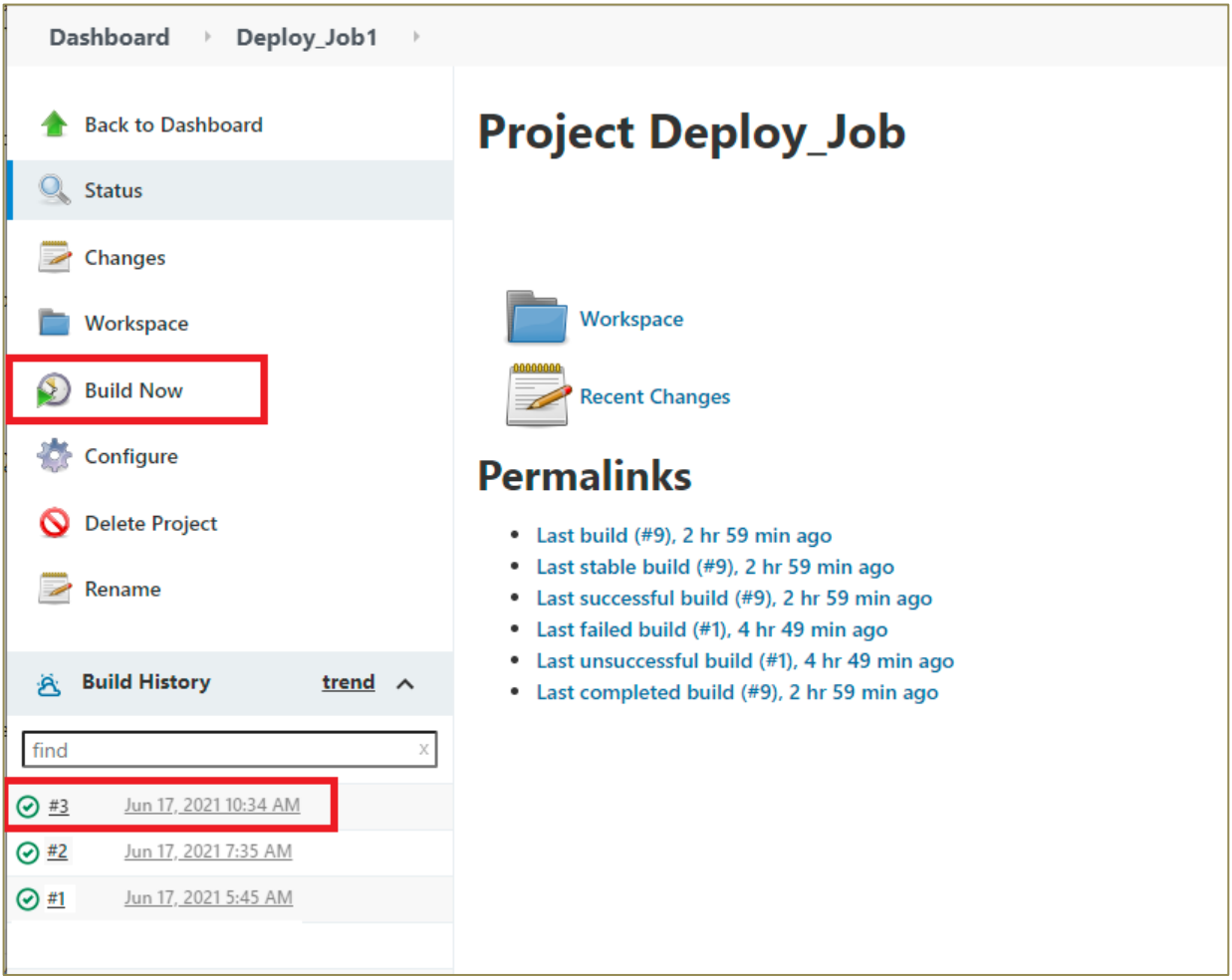
```
Username: training
Password: freefree
```



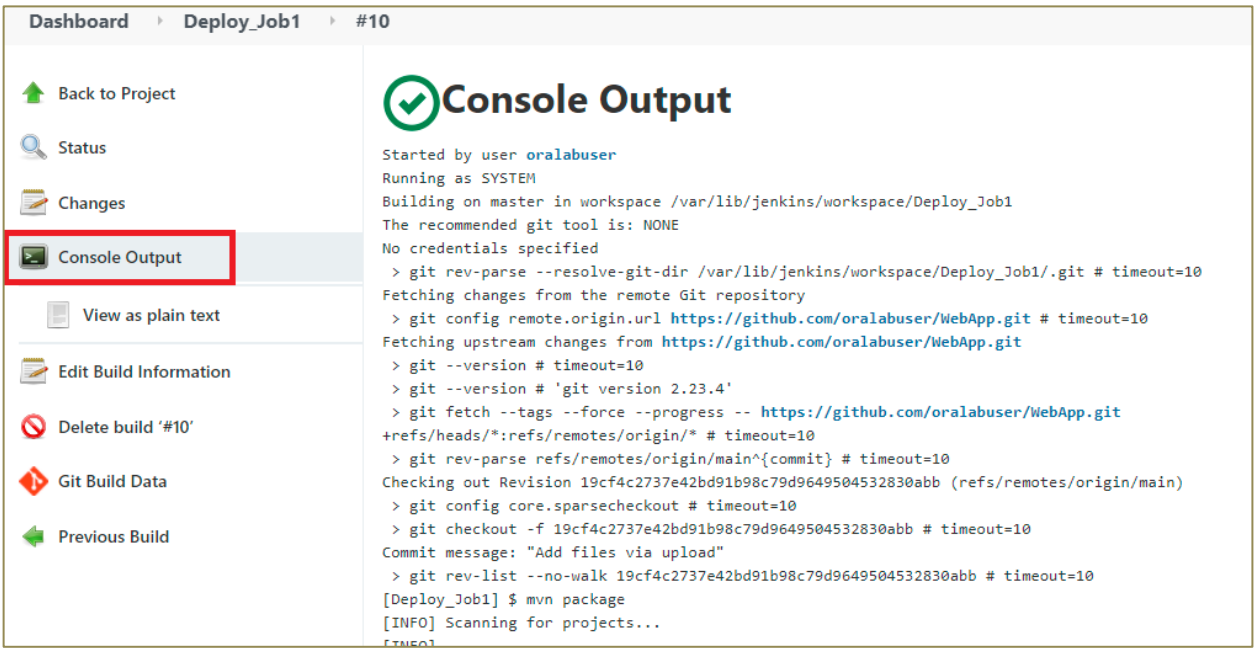
l. Select **Credentials** provided, give the private IP address of the Web-Server instance has **http://private_ip:8080** as shown below. Click **Apply** and **Save**.



m. Click **Build Now** to build the job in Jenkins and click on the latest build job link provided under **Build History** to verify the execution as shown below.



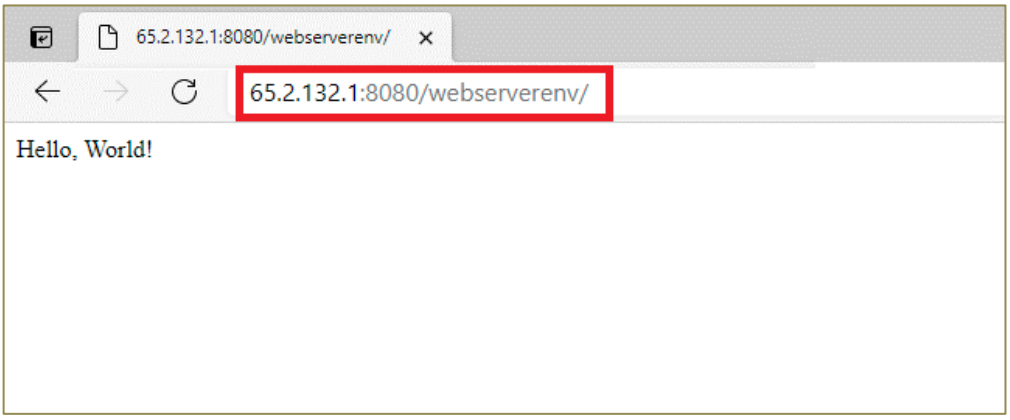
n. Select **Console Output** to view the execution as shown below.



```
[INFO] Maven Project ..................................... SUCCESS [0.001s]
[INFO] Server ............................................ SUCCESS [2.141s]
[INFO] Webapp ............................................ SUCCESS [1.014s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 3.336s
[INFO] Finished at: Thu Jun 17 10:35:04 UTC 2021
[INFO] Final Memory: 7M/19M
[INFO] ------------------------------------------------------------------------
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/Deploy_Job1/webapp/target/webapp.war
to container Tomcat 9.x Remote with context webserverenv
  Redeploying [/var/lib/jenkins/workspace/Deploy_Job1/webapp/target/webapp.war]
  Undeploying [/var/lib/jenkins/workspace/Deploy_Job1/webapp/target/webapp.war]
  Deploying [/var/lib/jenkins/workspace/Deploy_Job1/webapp/target/webapp.war]
Finished: SUCCESS
```

o. To access the home page verify it in the browser by providing the **Public_IP** of **Web-Server** with the port **8080** followed by the **Context path**.

```
<Public_IP of Web-Server>:8080/webserverenv/
```



8. Keep the Jenkins Dashboard, Linux-instance terminal and the AWS Management Console open for the next practice.