Deploying maven build “war” file to tomcat server using SSH-AGENT

Once maven build is successful we need to copy the war file from jenkins server to tomcat server into tomcat’s webapps directory over ssh.

STEP-1: install SSH-AGENT plugin in jenkins

Manage jenkins 🡺 Manage Plugins 🡺 Available plugins 🡺 Search for “ssh”.

Graphical user interface, text, application

Description automatically generated

Click on “Install without restart”

Graphical user interface, text, application

Description automatically generated

Once installation success restarts the jenkins server.

A picture containing graphical user interface

Description automatically generated

STEP-2: Once restart done go to jenkins and add the ssh credentials to connect with tomcat server instance from jenkins server instance.

Manage jenkins 🡺 Manage Credentials

Graphical user interface, application

Description automatically generated

Click on Jenkins

Graphical user interface, text, application, Teams

Description automatically generated

Click on “Global credentials”

Graphical user interface, text, application, email

Description automatically generated

Click on Add Credentials

Enter kind

Enter scope

Enter ID, Description

Graphical user interface, text, application, email, Teams

Description automatically generated

Enter the EC2 instance default username

1. ubuntu # for ubuntu os
2. ec2-user # for redhat and amazon linux os

click on “Enter directly” on private key

Now, open you aws pem key file and copy the key inside it

Graphical user interface, text

Description automatically generated

Copy this key and paste it in the “enter directly” section

Graphical user interface, application

Description automatically generated

Click on “Create”.

Graphical user interface, text, application

Description automatically generated

Now, our credentials for connecting jenkins server with tomcat instance are added successfully.

STEP-3: Now, go to jenkins pipeline job and add the new stage to copy the build file from jenkins to tomcat server over ssh agent.

Graphical user interface, application, Teams

Description automatically generated

Go to Pipeline section

Add a new stage to copy the build file from jenkins to tomcat server.

Graphical user interface, text, application

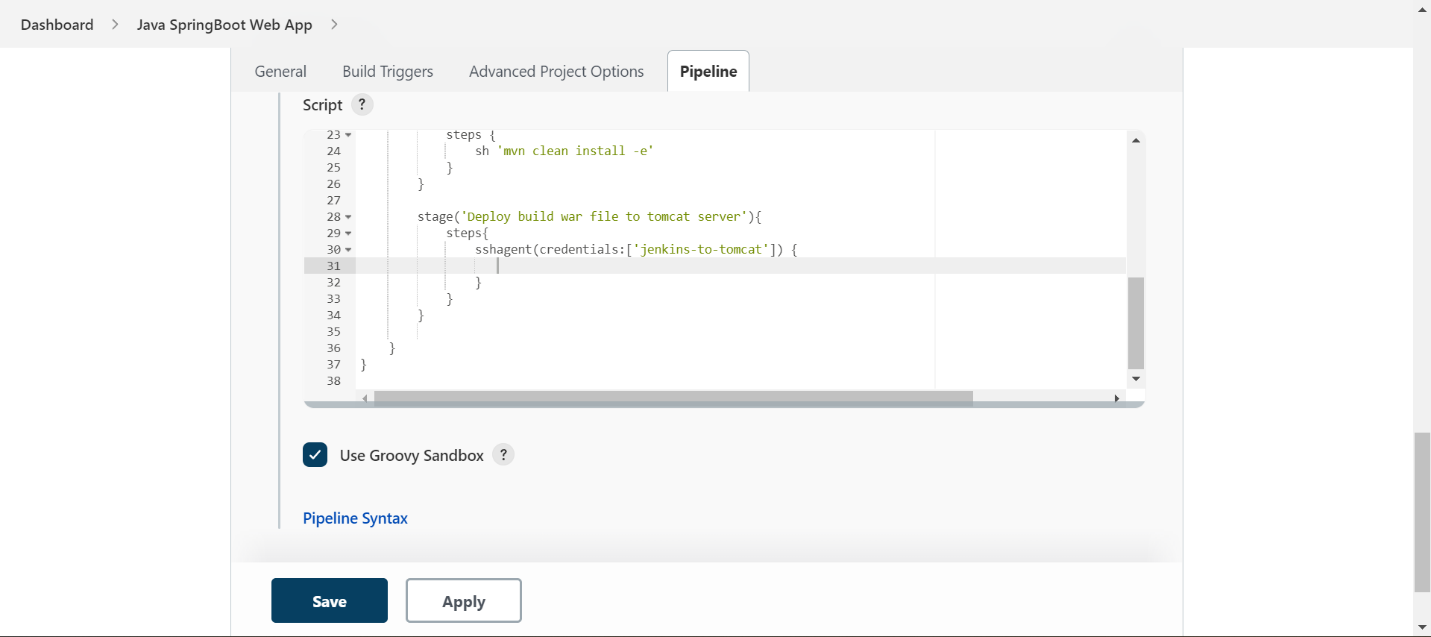
Description automatically generated

Now, generate the step code using “Pipeline Syntax”.

Graphical user interface, text, application, email

Description automatically generated

This is the code generated, copy this code and paste it in the step.



Now, write the scp (secure copy) command in the “sshagent” step.

**scp syntax:**

scp <source path> remote-user@ip-address:<destination path>

* sh 'scp -o StrictHostKeyChecking=no /home/ec2-user/jenkin\_workspace/workspace/Java-SpringBoot-WebApp/target/ROOT.war ubuntu@54.90.96.16:/opt/apache-tomcat-9.0.65/webapps/myapp.war'

If we execute this command, it will ask the prompt like “would you like to allow login or not”, so our stage will halted of failed until we provide yes/no.

We have to bypass this check.

Add this parameter in the scp command.

-o StrictHostKeyChecking=no

here, -o 🡺 for authentication

* sh 'scp -o StrictHostKeyChecking=no /home/ec2-user/jenkin\_workspace/workspace/Java-SpringBoot-WebApp/target/ROOT.war ubuntu@54.90.96.16:/opt/apache-tomcat-9.0.65/webapps/myapp.war'

copy the scp command and paste in the step.

Graphical user interface, application

Description automatically generated

Apply and save the changes.

Now, lets build the pipeline

Graphical user interface, application, Word

Description automatically generated

Graphical user interface, application, table

Description automatically generated

Build 35 started

Graphical user interface, table

Description automatically generated

Our build got failed in deploy stage, let’s check the build logs.

Click on build number 35.

Graphical user interface, text, application, email

Description automatically generated

Go to “Console Output”

A picture containing diagram

Description automatically generated

It’s unable to find the “Destination” path.

**Cause:**

We are logging to the tomcat server as “ubuntu” user, But our tomcat installation directory path files are owned by “root” user.

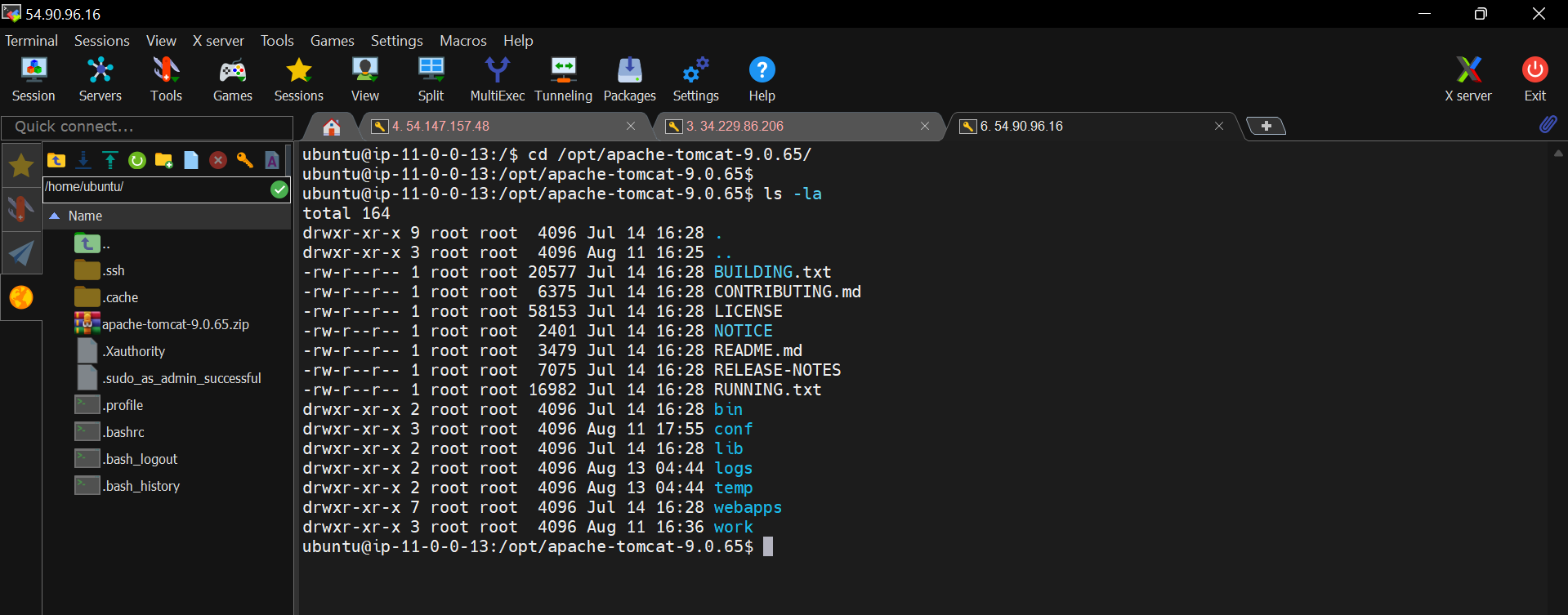
**Resolution:**

Change the ownership of tomcat installation directory path from root user to ubuntu user.

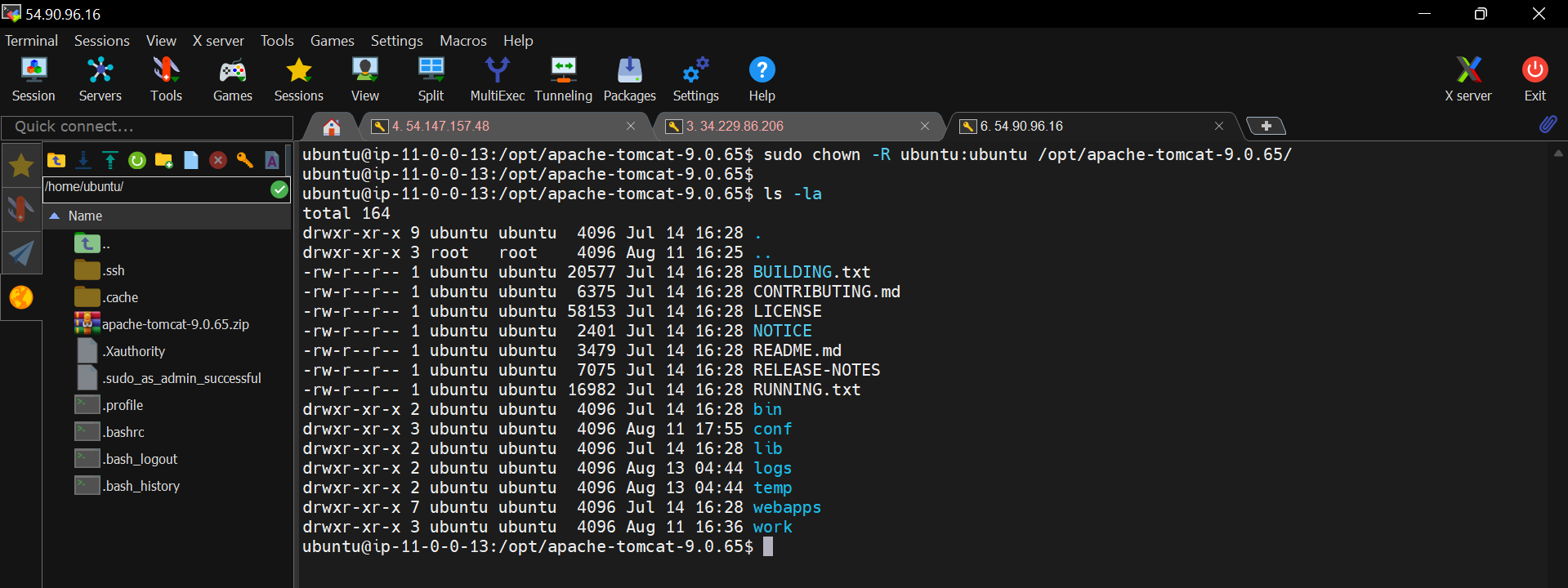
* sudo chown -R ubuntu:ubuntu /opt/apache-tomcat-9.0.65

this will give the access to all the files in tomcat cat installation directory to ubuntu user.

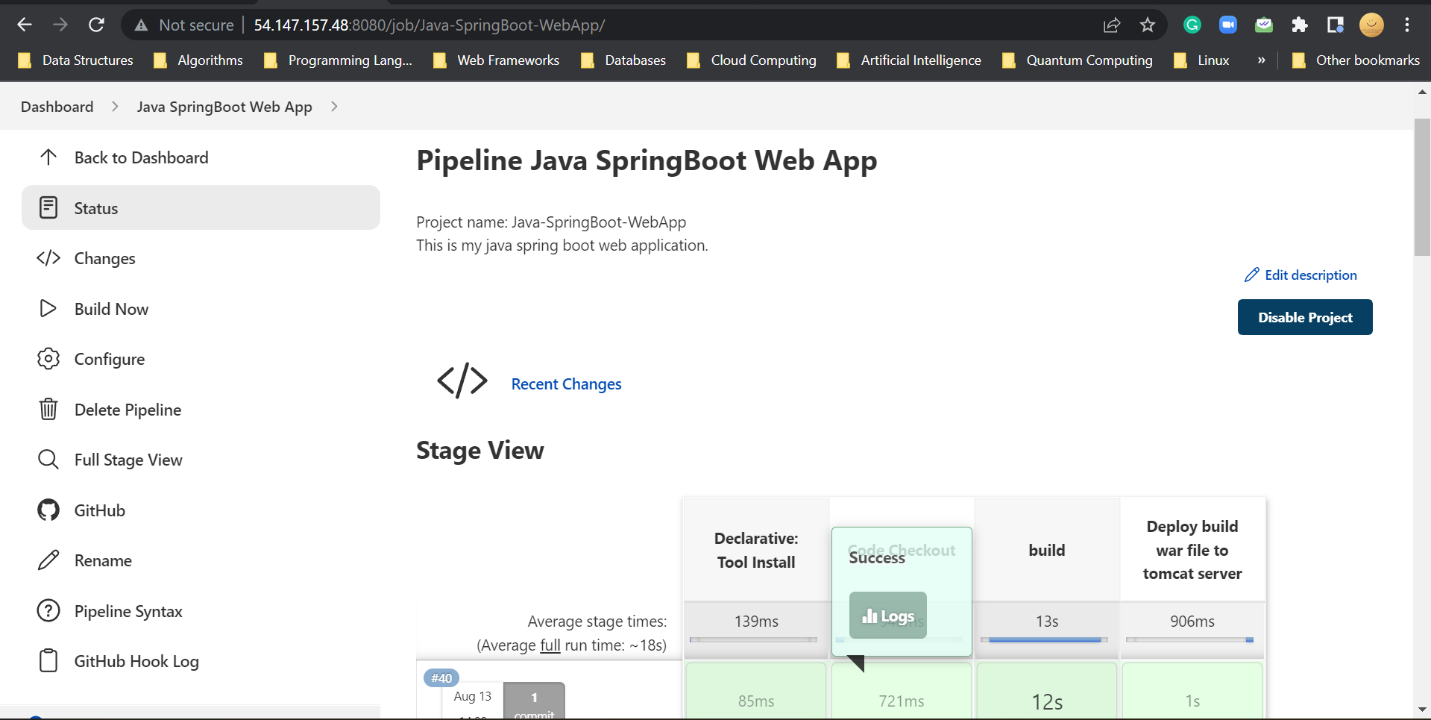
Before “chown” command

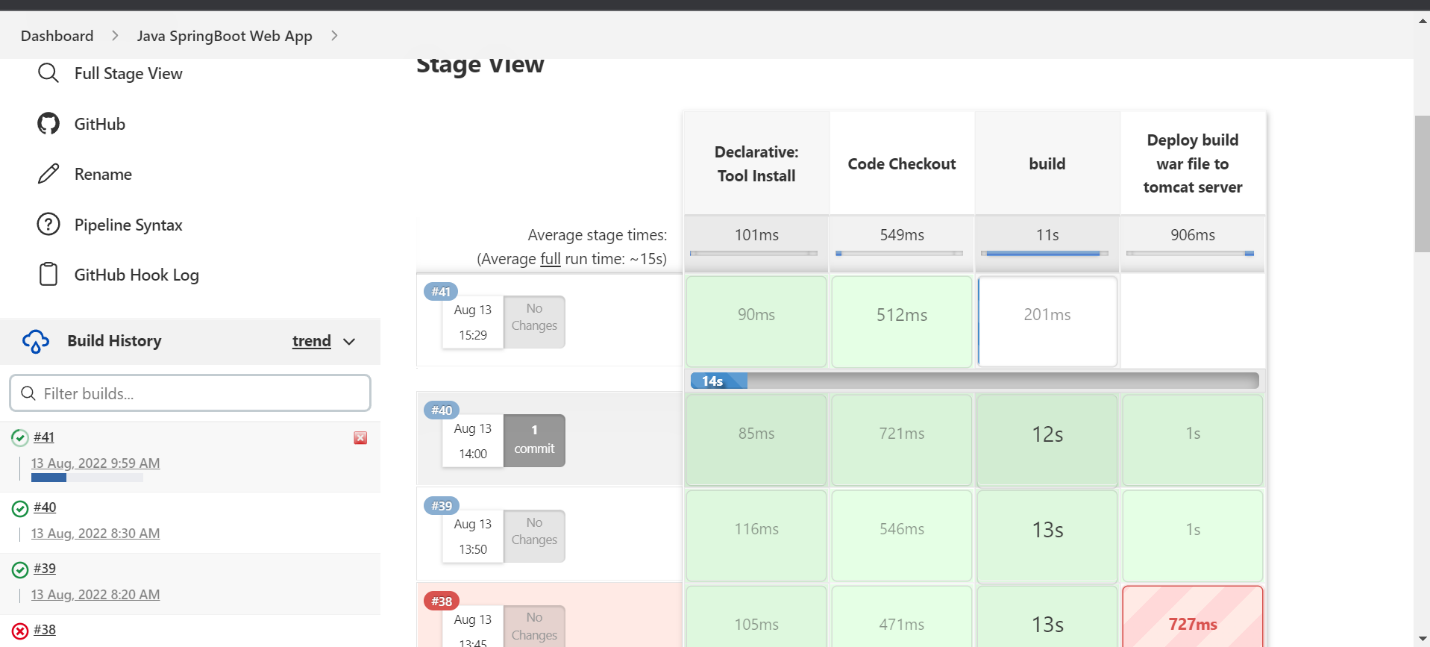


After “chown” command



Now, run the build



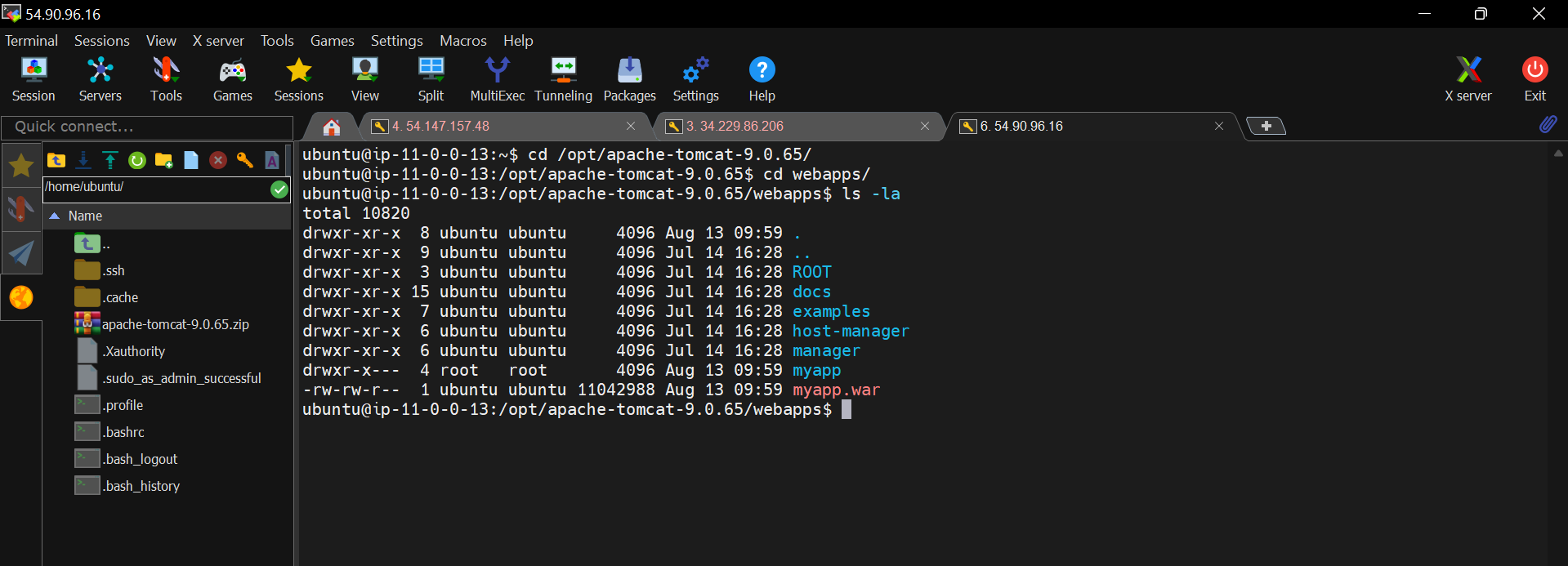


New build 41 is started.

Table

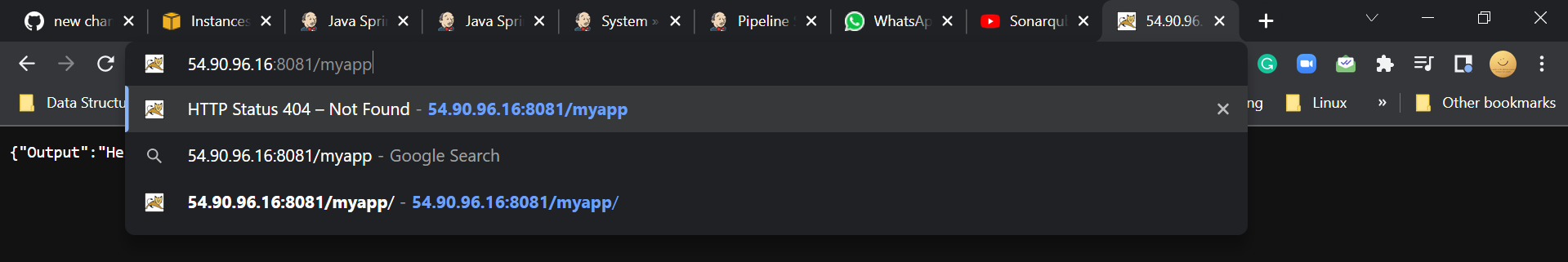
Description automatically generated

Build got success and deployed to tomcat server successfully.



We can see that our war file successfully copied to tomcat server.

Now, go to tomcat server and enter the app name after the port number.



Graphical user interface, text

Description automatically generated

We can see our app output here.