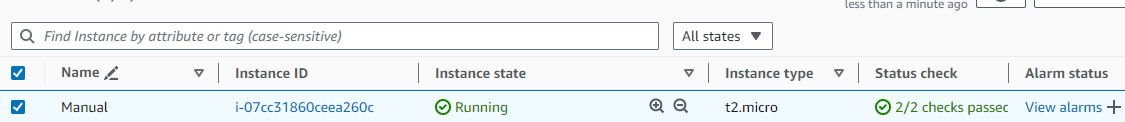
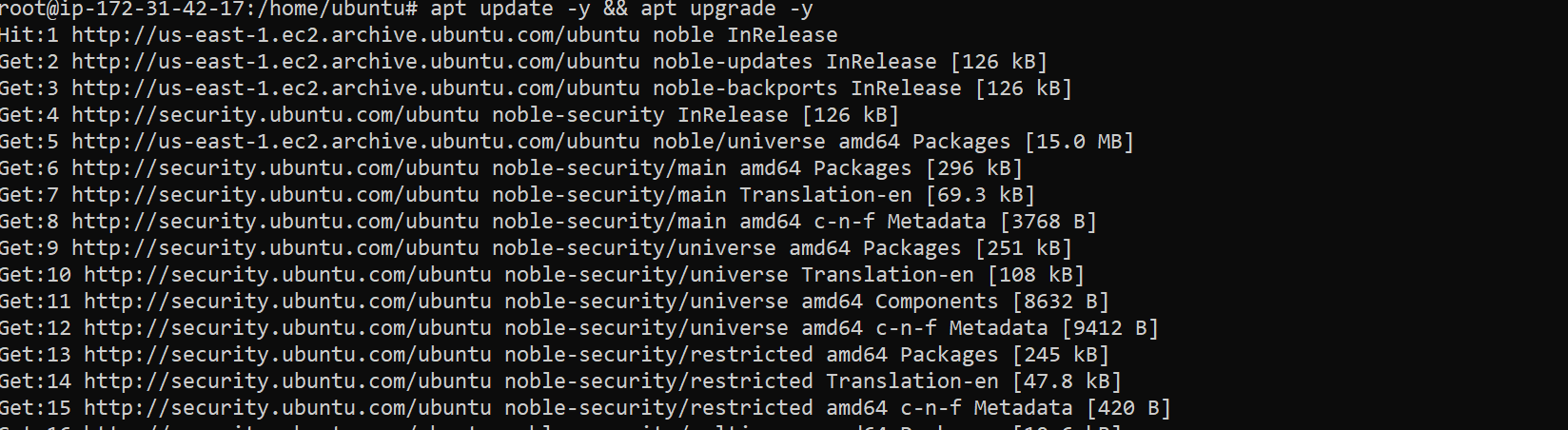
**Manual Project Build with Tomcat 9**

1. **Launch an EC2 Instance**

****

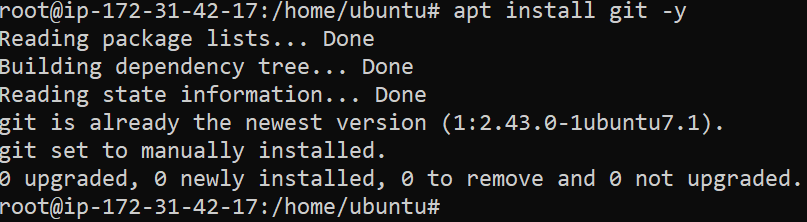
1. **Now login and run the below commands to update and upgrade the packages of the OS.**

apt update -y && apt upgrade -y



1. **Now install git and git version using the below command.**

apt install git -y

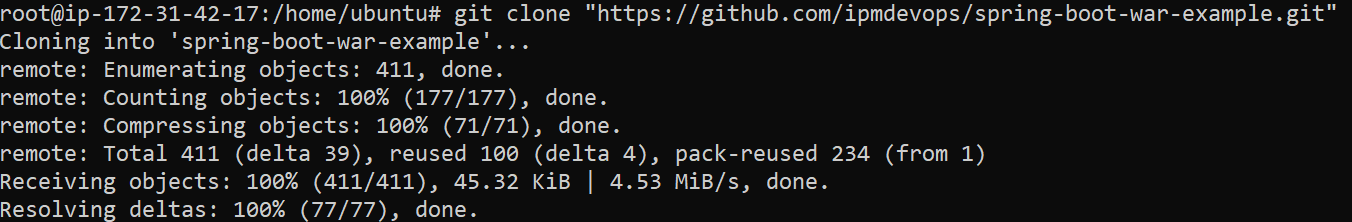


git –version

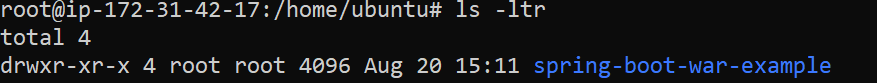


1. **Now clone the git repository which you want to build the project and check if the repo is cloned properly.**

git clone "https://github.com/ipmdevops/spring-boot-war-example.git"



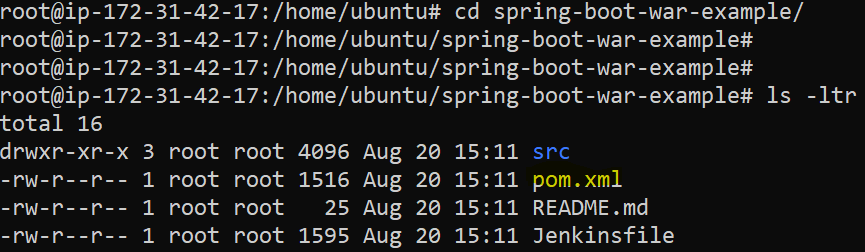
ls -ltr



1. **Go inside the cloned project directory and verify the source code and pom.xml file is present so that we can build the project.**

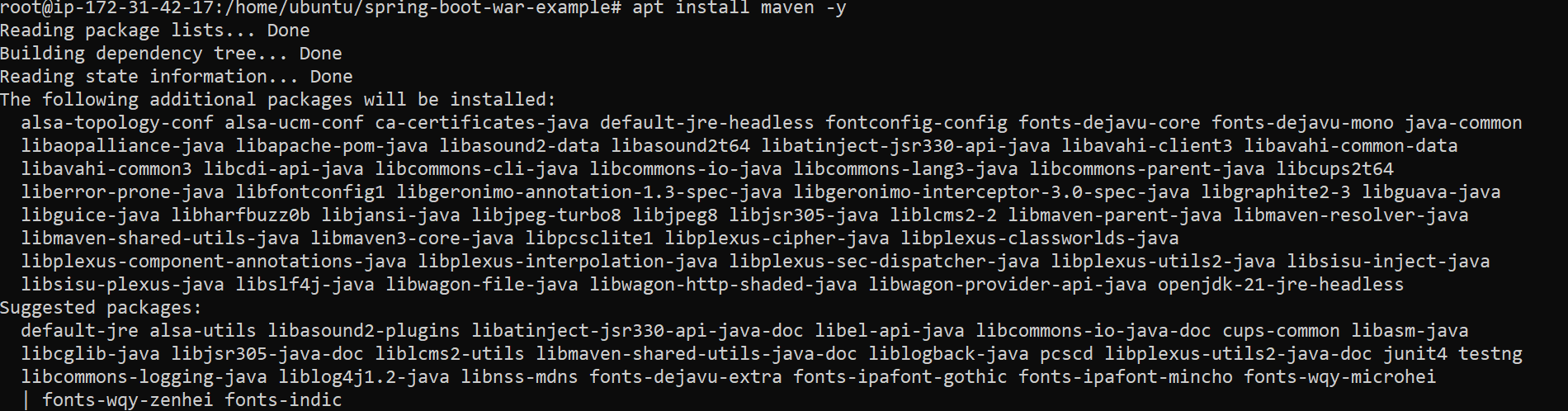
cd spring-boot-war-example

ls -ltr

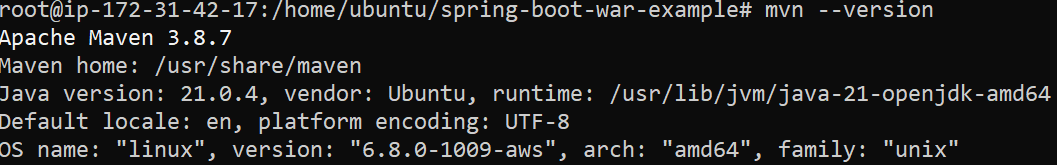
****

1. **To Build the project, we need to install Apache Maven Tool and verify version which helps in testing the source code and builds it as a war or jar file in case of java.**

apt install maven -y



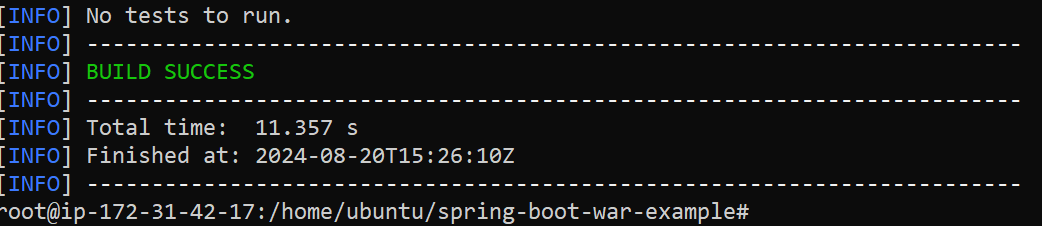
mvn –version



1. **Now we will test the code using below Maven command which uses information from pom.xml.**

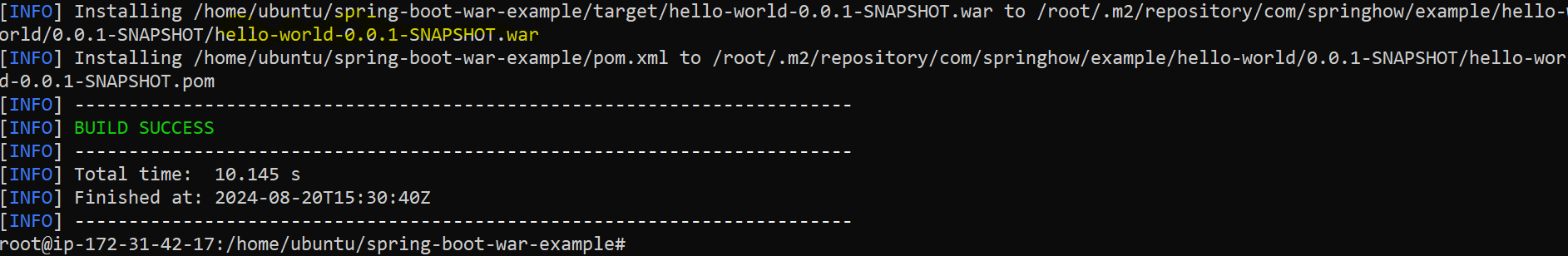
**NOTE: We need to run the below command from the same location where pom.xml is located**

mvn test [Below output comes if the source code is without any errors]



1. **Now once we get the above output, we run the below command to build the project.**

mvn install [Below output shows project build is successful and war file is created]

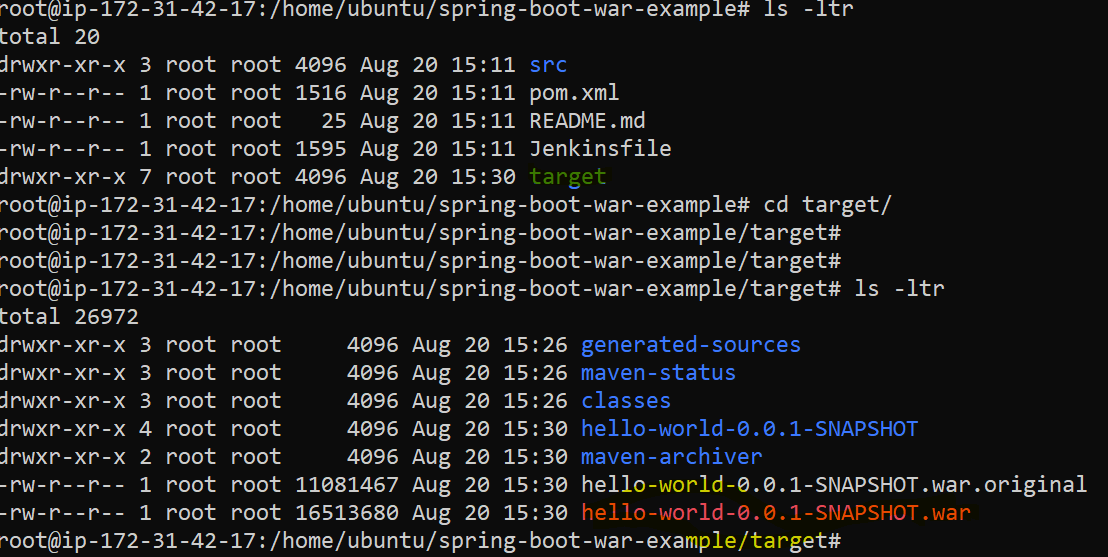


1. **Once the build is successful, a folder named “target” is created which has the war file in it.**

ls -ltr

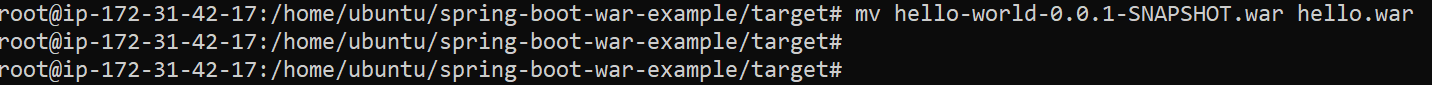
cd target/

war file created by the name hello-world-0.0.1-SNAPSHOT.war

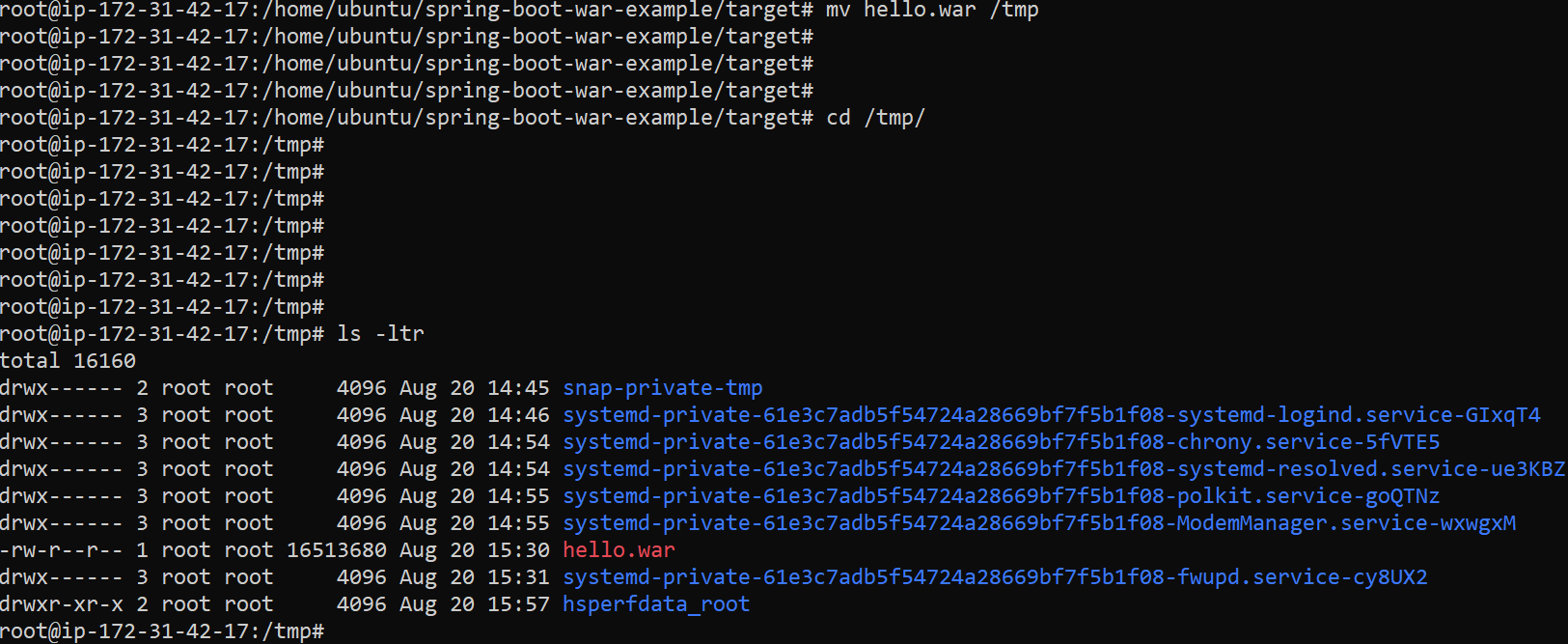
****

1. **Now we can rename the war file to example hello.war and move it to an artifact (in our case the artifact is /tmp)**

mv hello-world-0.0.1-SNAPSHOT.war hello.war (rename war file)



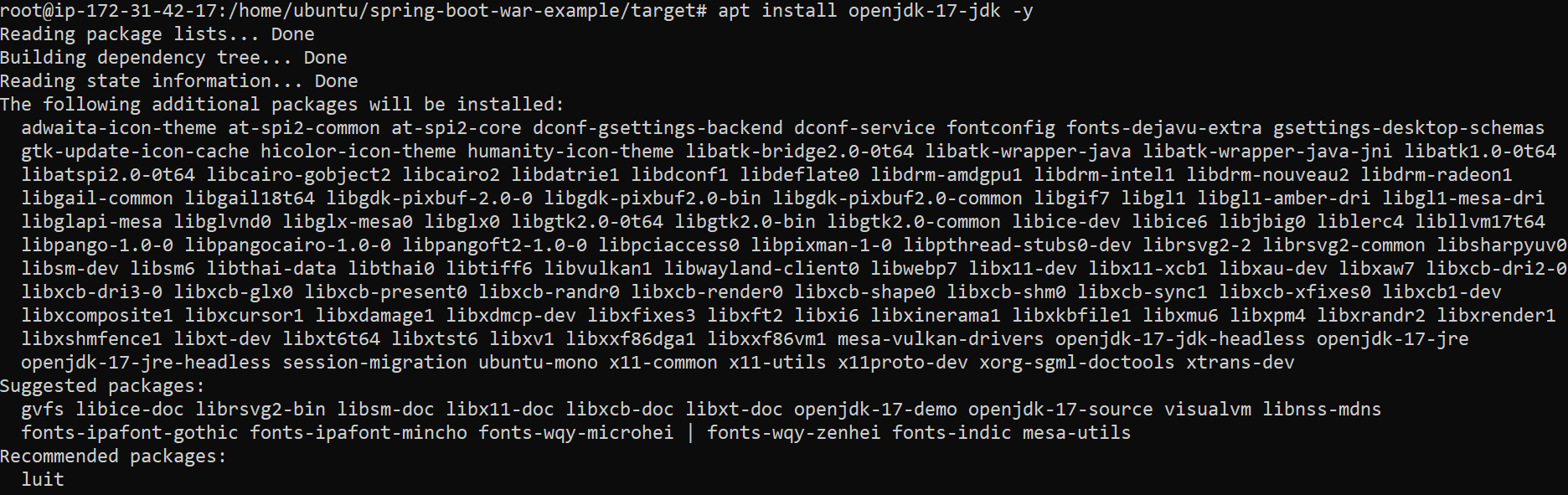
mv hello.war /tmp (war file moved to artifact location)



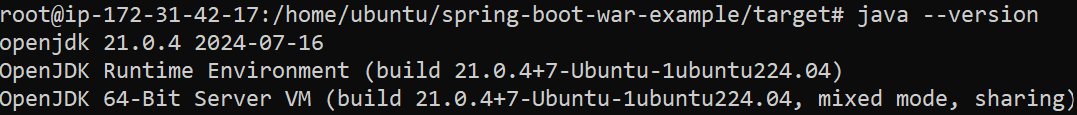
1. **Now we install Apache tomcat version 9 so that we can check code is working in WEBGUI.**

Pre-requisite for installing Tomcat is to install open-jdk and check java version

apt install openjdk-17-jdk -y



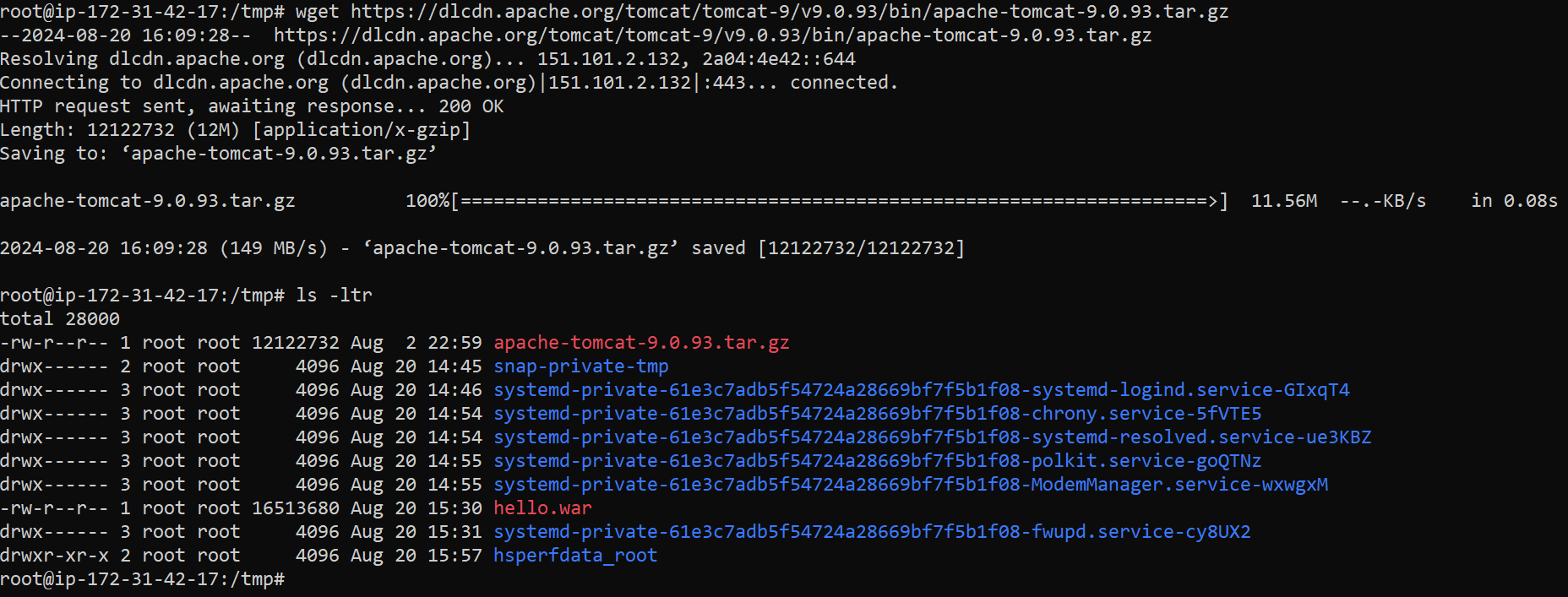
java –version



1. **Since Tomcat version 9 is not available as default package in ubuntu so we have to install Tomcat 9 manually. Below are the steps to download and install tomcat 9.**

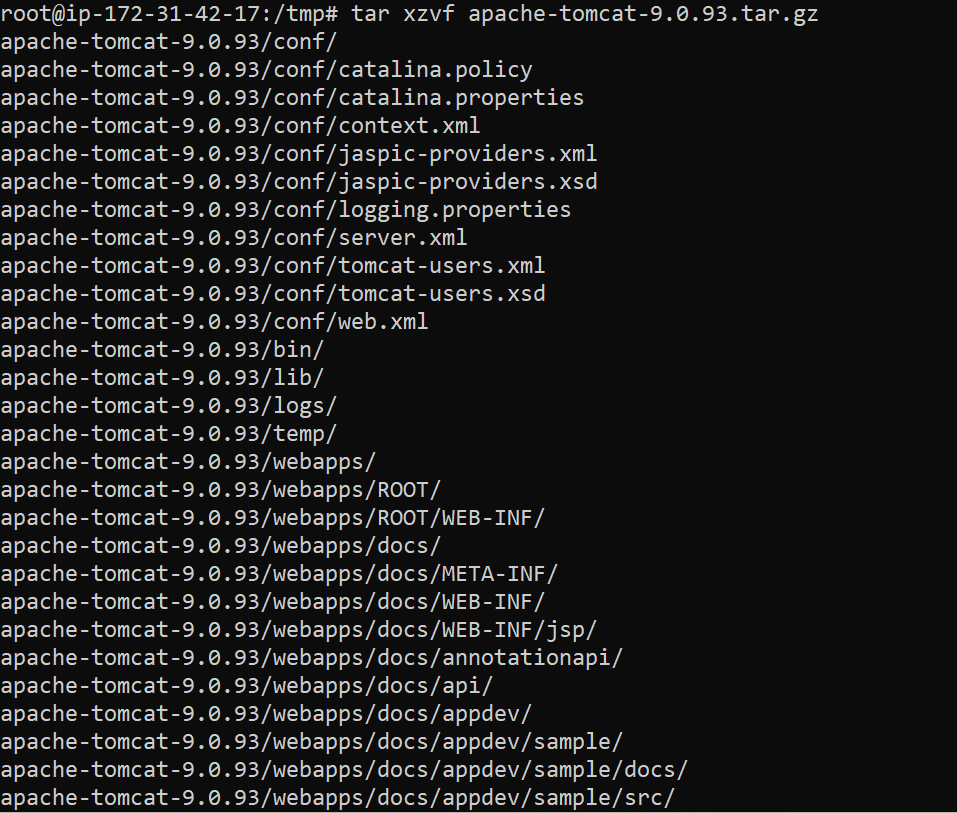
wget <https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.tar.gz>

ls -ltr to check if the file is downloaded



1. **Untar the tar file using below command and check tomcat folder is created or not.**

tar xzvf apache-tomcat-9.0.93.tar.gz



ls -ltr



1. **Now rename the extracted folder tomcat so that its easy to remember and check.**

mv apache-tomcat-9.0.93 tomcat

ls -ltr

****

1. **Now we have to create the tomcat service manually in the system location mentioned below and add the contents in the service file.**

vi /etc/systemd/system/tomcat.service

Contents of tomcat.service file

------------------------------------------

[Unit]

Description=Tomcat 9 servlet container

After=network.target

[Service]

Type=forking

Environment="JAVA\_HOME=/usr/lib/jvm/java-17-openjdk-amd64"

Environment="JAVA\_OPTS=-Djava.security.egd=file:///dev/urandom -Djava.awt.headless=true"

Environment="CATALINA\_BASE=/tmp/tomcat/"

Environment="CATALINA\_HOME=/tmp/tomcat/"

Environment="CATALINA\_PID=/tmp/tomcat/temp/tomcat.pid"

Environment="CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

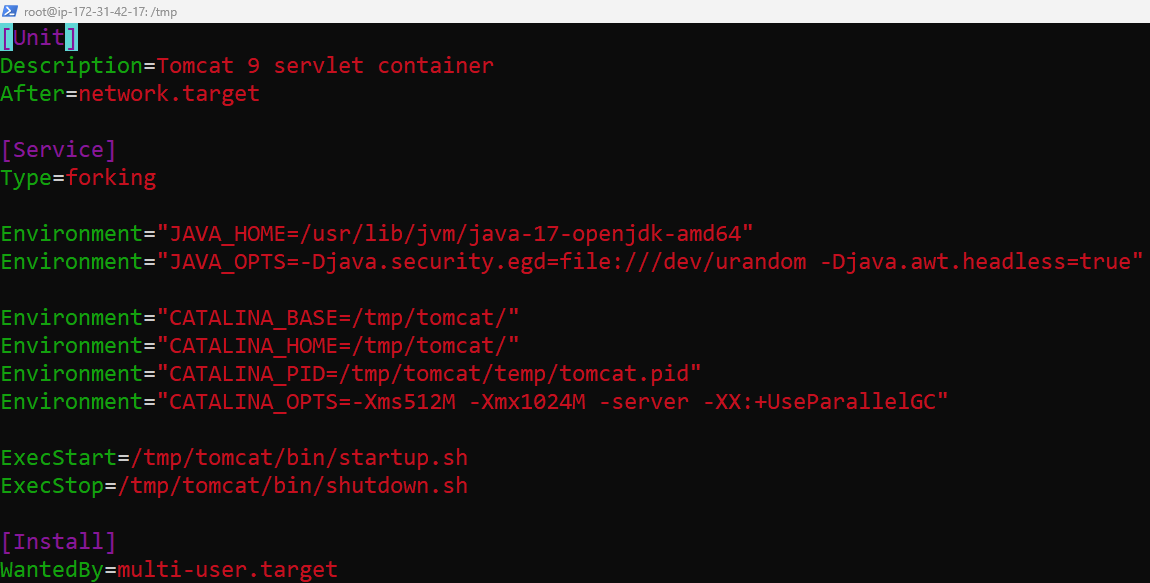
ExecStart=/tmp/tomcat/bin/startup.sh

ExecStop=/tmp/tomcat/bin/shutdown.sh

[Install]

WantedBy=multi-user.target



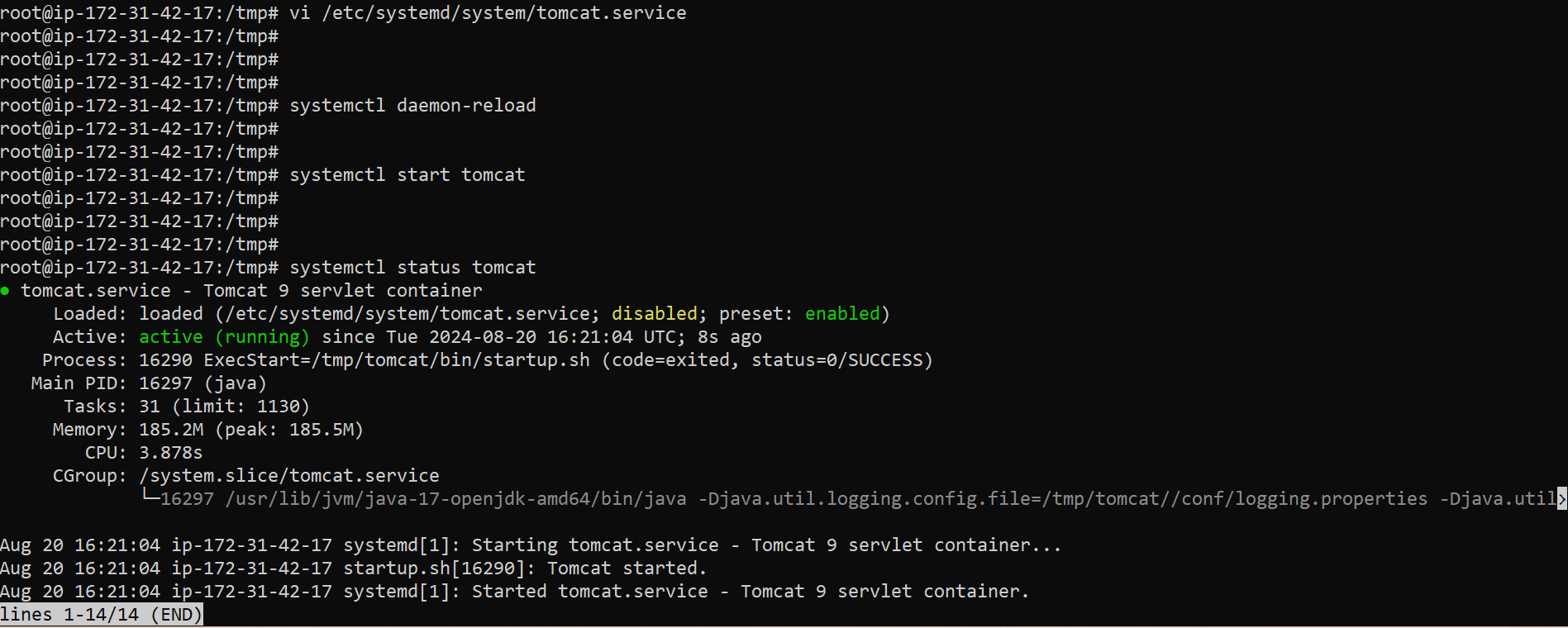


1. **Now we will reload the daemon and start tomcat service and also verify that the service is running.**

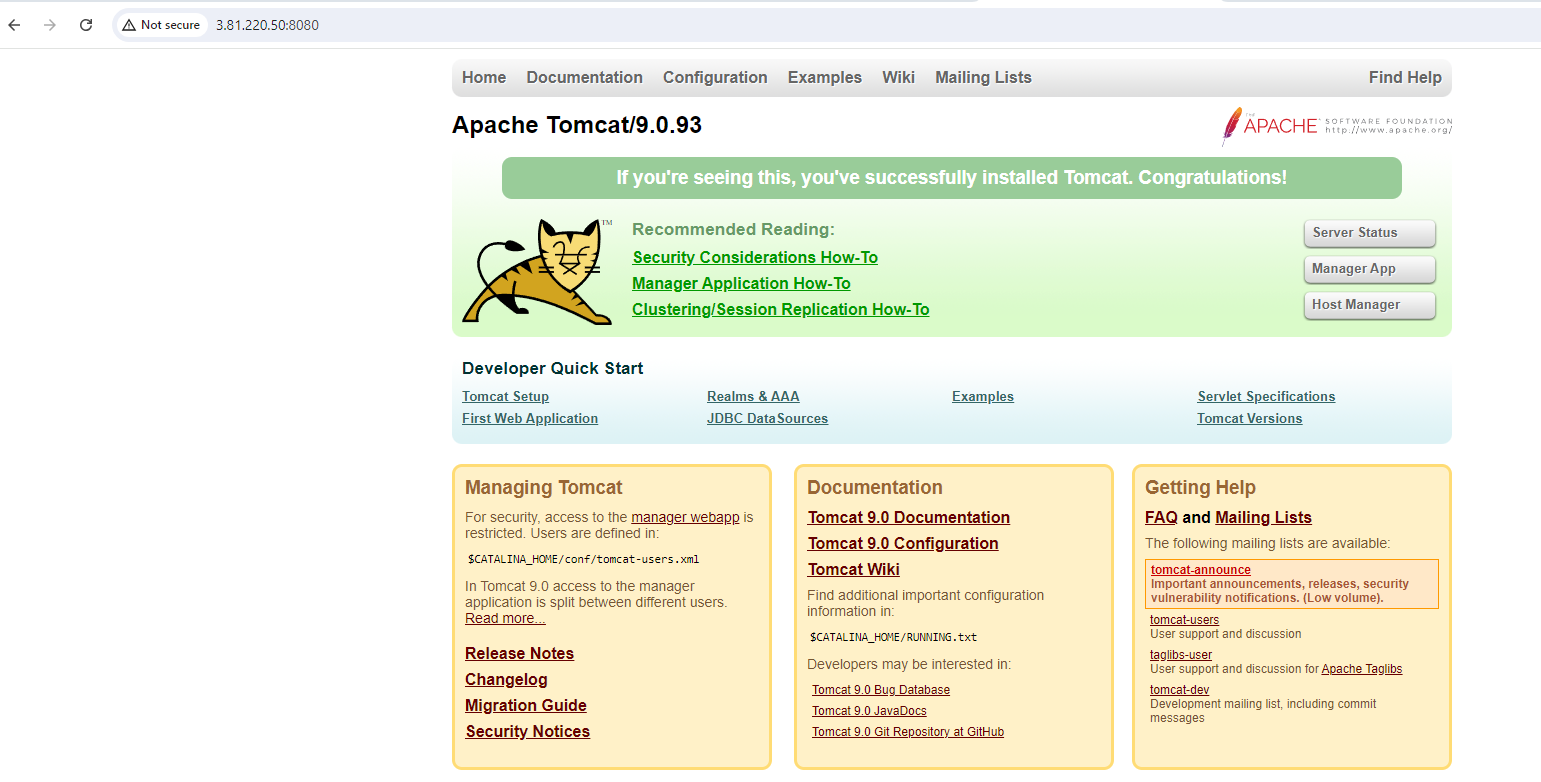
systemctl daemon-reload

systemctl start tomcat

systemctl status tomcat

****

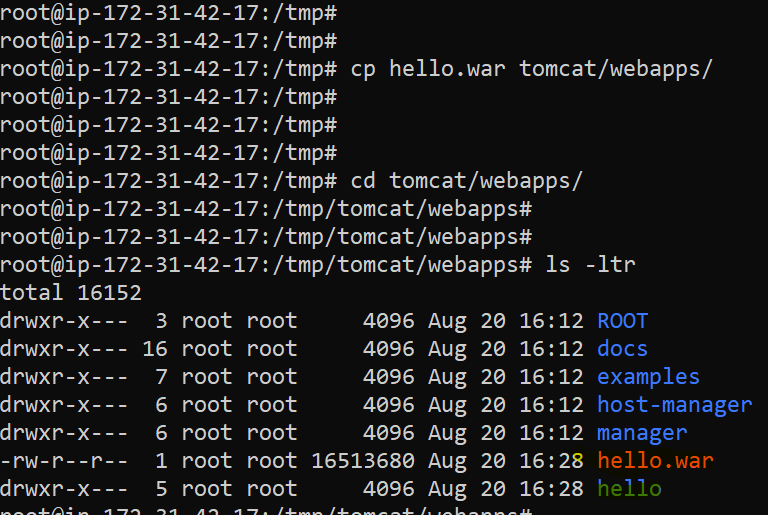
1. **We will verify that tomcat is running or not on the WEG UI by accessing tomcat URL – PublicIP of EC2 instance:8080 (Tomcat works on port 8080)**

****

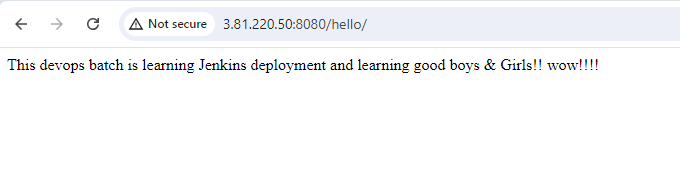
1. **Now we will copy the hello.war file from /tmp to tomcat/webapps. Once the war file is copied in webapps folder we see that hello folder is also created in it.**

cp hello.war tomcat/webapps

ls -ltr

****

1. **Now we will verify that code is accessible on tomcat GUI by accessing tomcat** [**URL:8080/hello**](URL:8080/hello)

****

1. **Once we get the above output, we are sure that our manual build is successful visible and tested.**