# Build and Deployment Process Using Two Servers

This document describes the step-by-step process for setting up a Build Server and a Deployment Server for Java web application deployment using Maven and Tomcat. The Build Server handles code compilation and WAR creation, while the Deployment Server hosts the application using Apache Tomcat.

## Server Setup Overview

1. Build Server: Used to build and create the artifact (.war file).

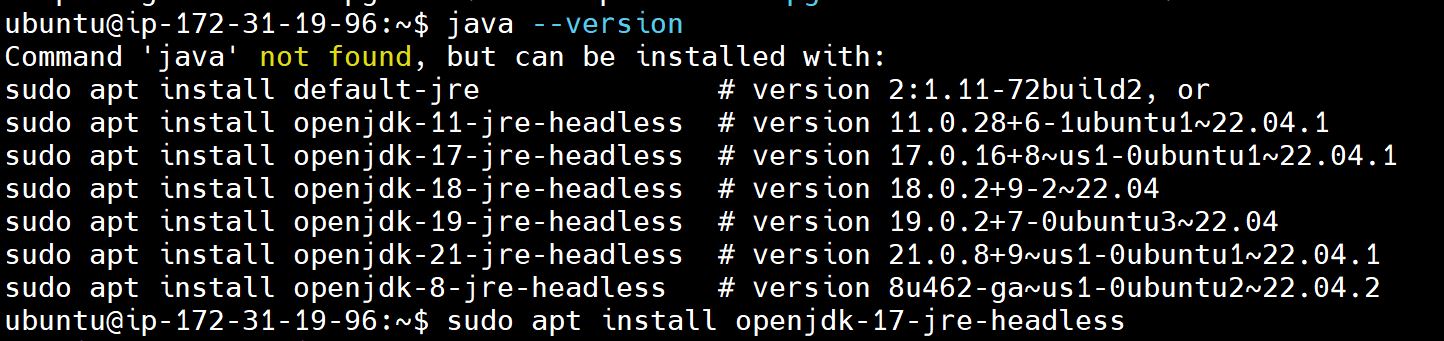
- Install Java and Maven.

2. Deploy Server: Used to deploy and host the application using Tomcat.

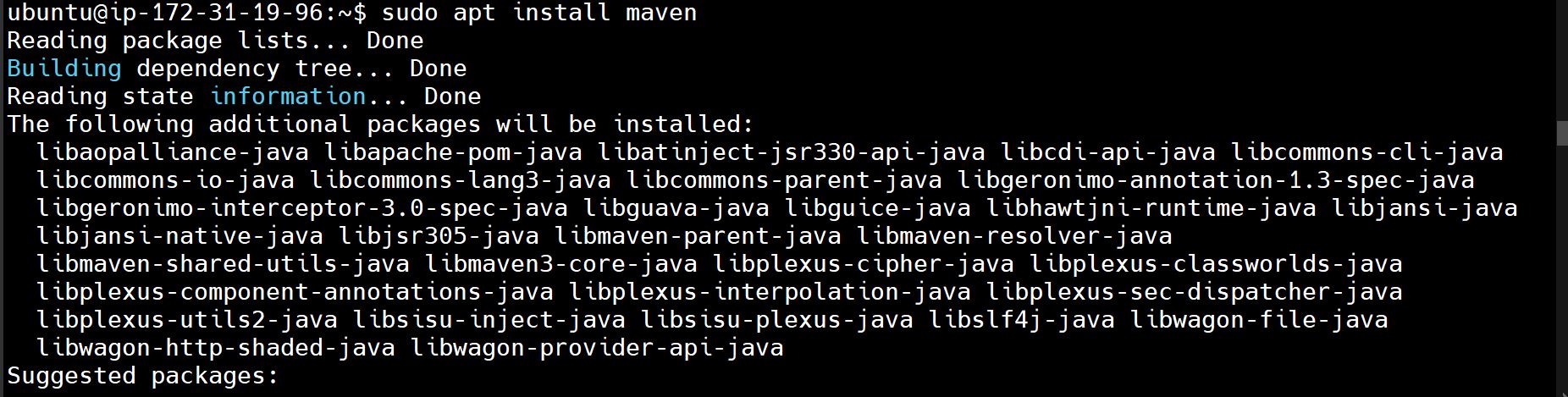
- Install Java and Tomcat.

## Step-by-Step Process

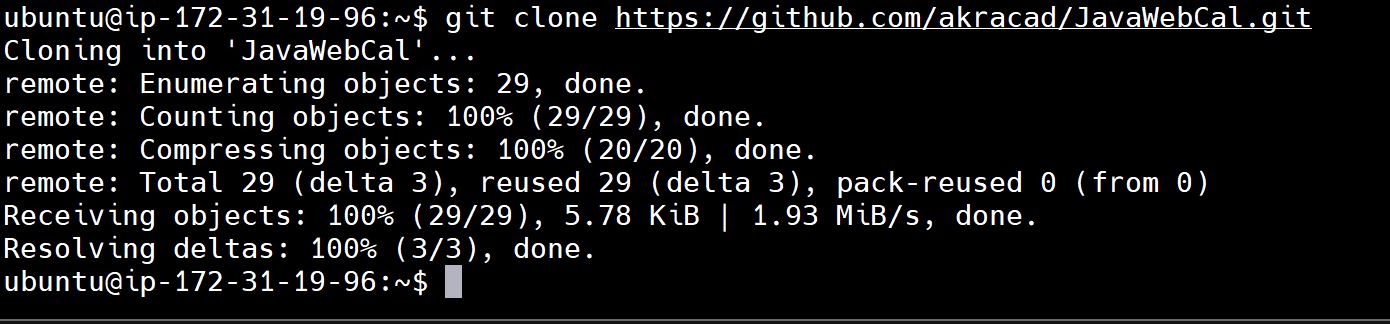
1. Install Java on Build Server



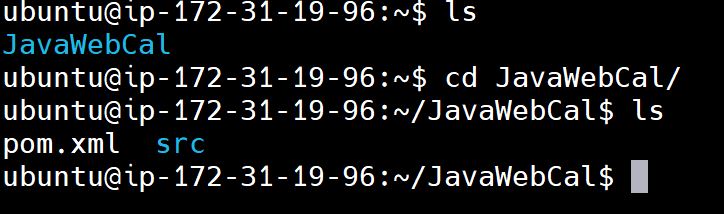
1. Install Maven on Build Server



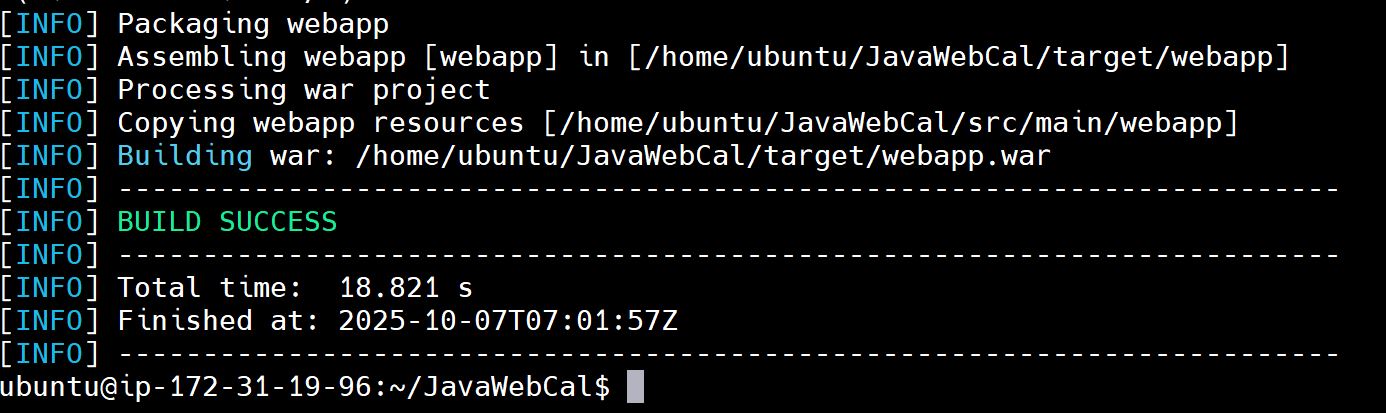
1. Clone the code repository from GitHub on Build server



Now check, whether you have cloned the code and pom.xml file exists.

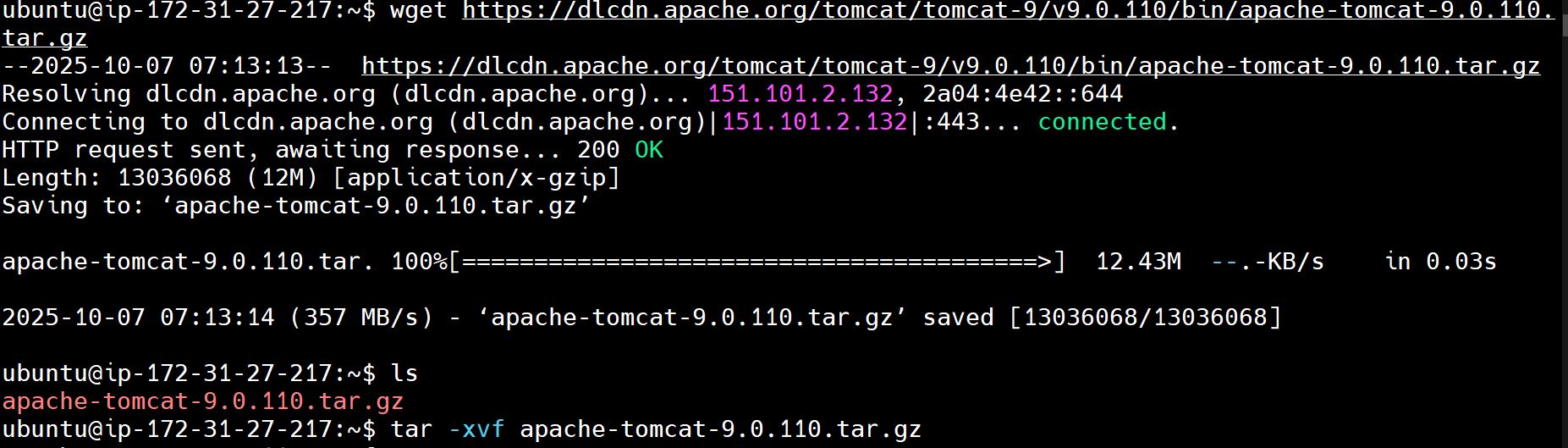


1. Build the artifact using “mvn package” command, then you can see the output as below



1. Install Java on Deploy server using command “sudo apt install openjdk-17-jre-headless” (Same way we have installed java on Build server)
2. Install Tomcat on the Deploy Server

* Go to Apache tomcat official website
* Copy the tar extension file url link
* Download it using wget command followed by the url link
* After downloading tar file, extract the tar file using “tar -xvf apache-tomcat-9.0.110.tar.gz” command

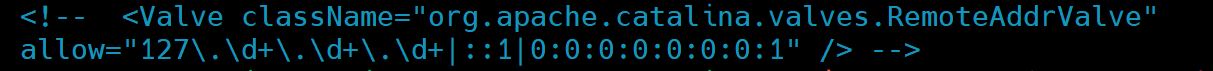


1. Configure “webapps/host-manager/META-INF/context.xml” on the Deploy Server

* After opening the context.xml file, locate the following lines

<Valve className="org.apache.catalina.valves.RemoteAddrValve"

allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" and comment it out by enclosing it within <!-- and --> tags.

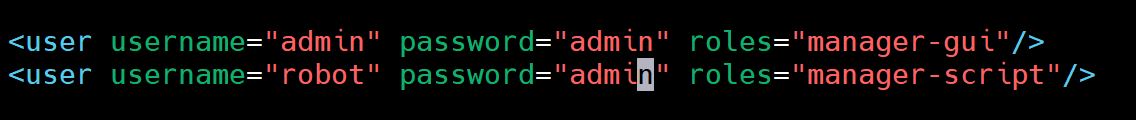


1. Configure webapps/manager/META-INF/context.xml on the Deploy Server.

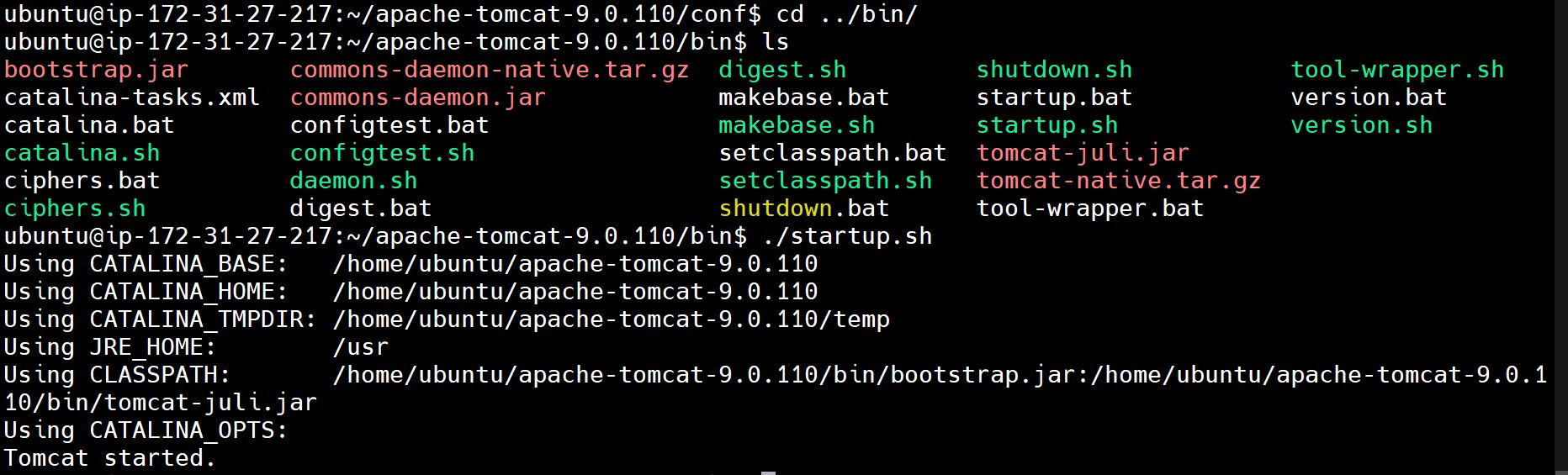
This step is the same as the previous one, but here you need to modify the context.xml file located in webapps/manager/META-INF/.

1. Configure conf/tomcat-users.xml on the Deploy Server with appropriate roles and users

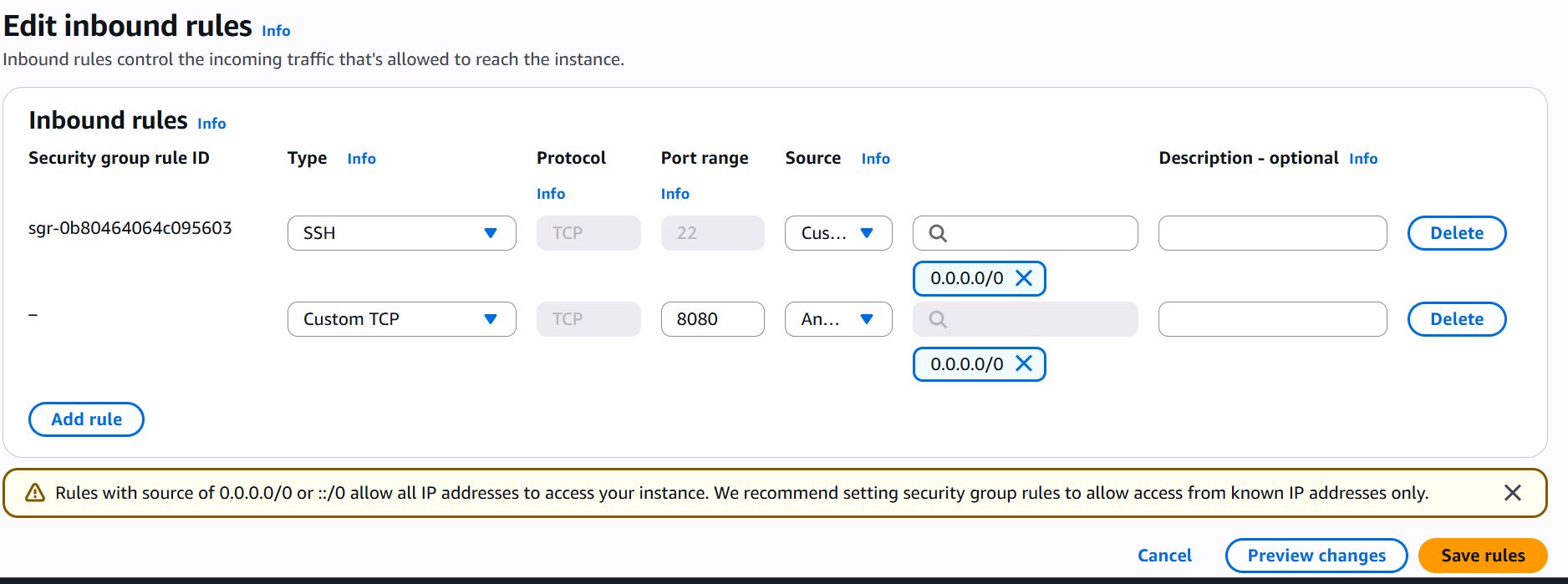
* After opening file, locate for roles lines which are pre-defined and commented, those lines should be uncommented and passwords should be changed.



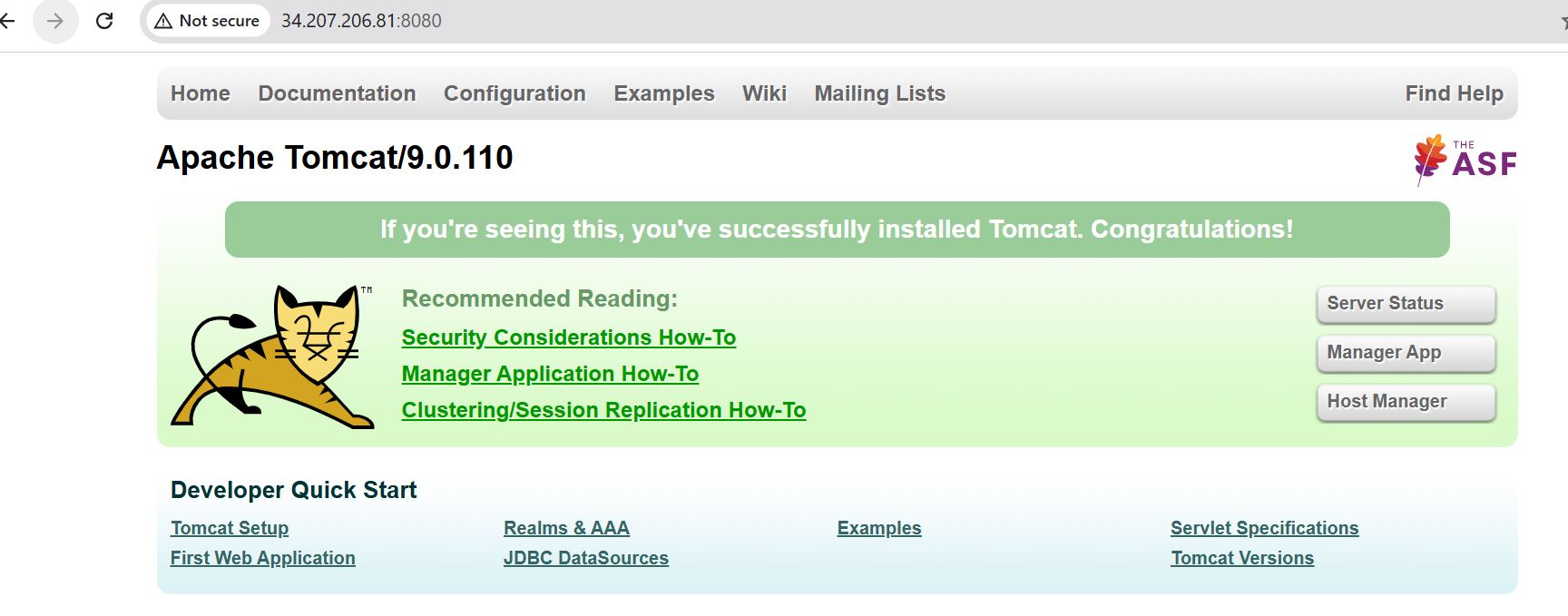
1. Start the Tomcat service on the Deploy Server by changing to “bin” directory



1. Allow inbound rules for port 8080 on the Deploy Server in AWS security groups



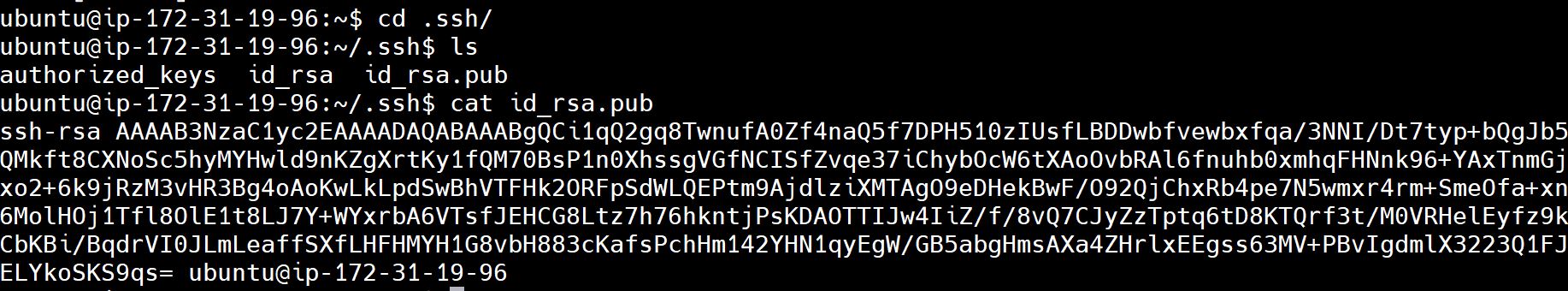
1. Access the Tomcat web interface on the browser using the public IP of the Deploy Server.



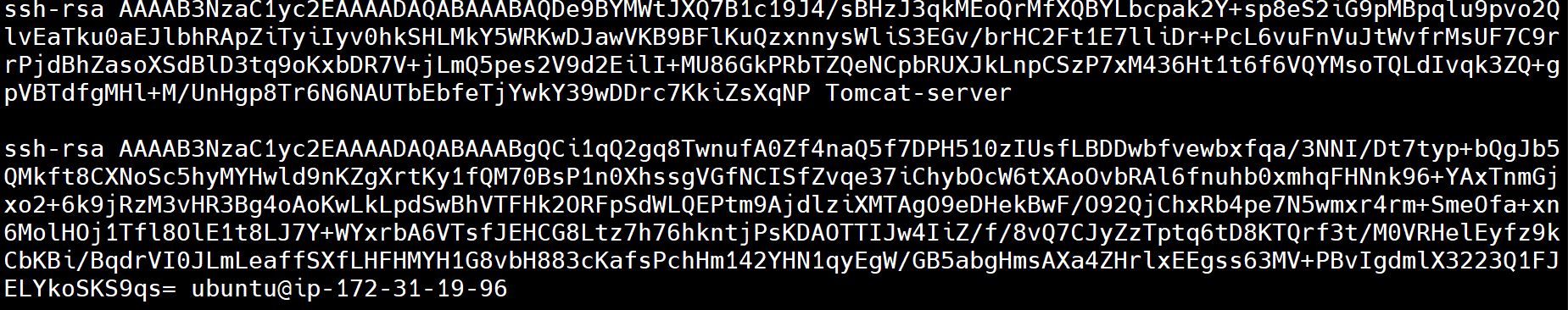
1. Now, to copy the .war artifact from Build server to Deploy server you need to connect both the servers through ssh keys.

Create ssh keys using “ssh-keygen” command on Build server

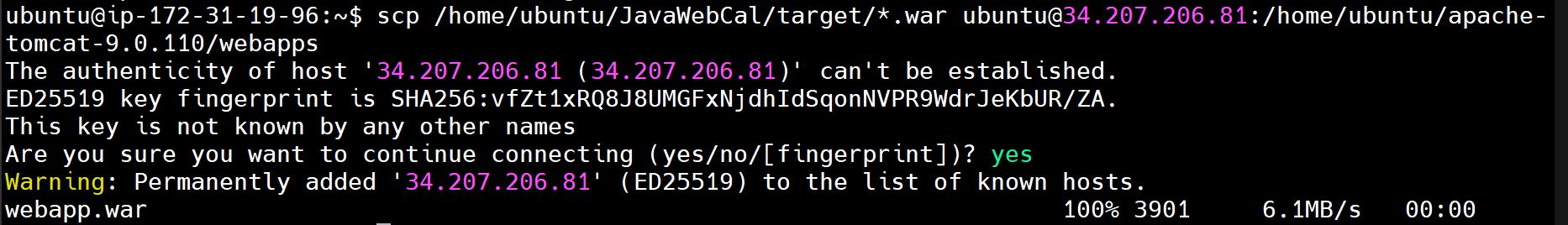
Copy the public keys “id\_rsa.pub” of Build server



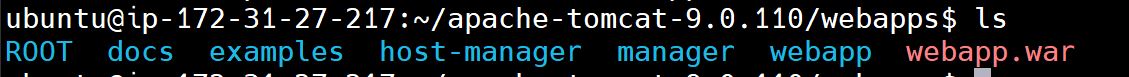
Paste the copied public keys of build server to authorized keys (by changing to .ssh directory) of Deploy server



1. Copy the .war file from the Build Server to the Deploy Server using SCP command



1. Verify that the .war file has been copied to /opt/tomcat/webapps/ on the Deploy Server



1. Access the deployed application through the browser using the public IP of the Deploy Server.

