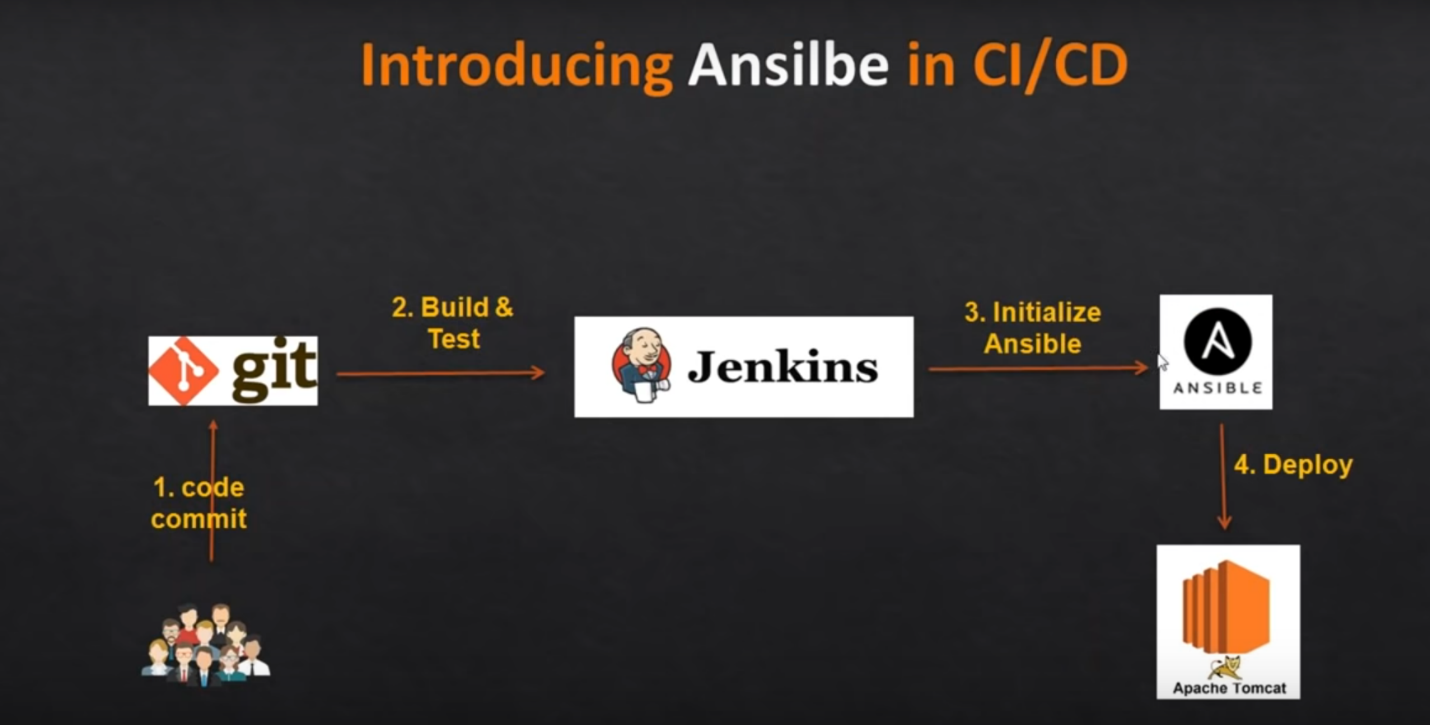
**Project AWS and DevOps- Simple CI-CD Project through Jenkins, Ansible and Tomcat in AWS**



Follow The Following Steps to achieve above project.



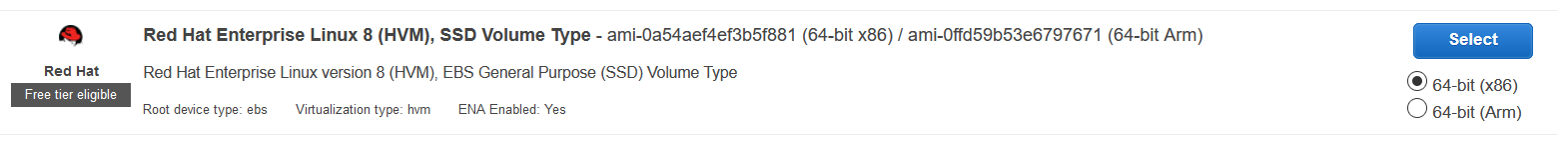
**Step-1:**

Launch 3 EC2 Machine (**Red Hat Enterprise Linux 8 (HVM),)**

Jenkins Server

Tomcat Webserver

Ansible Server



**Step-2:**

**Tomcat installation on EC2 instance**

1. Login to Tomcat Server EC2 Machine
2. Install Java1.8 if not installed.
3. We will be using open java for our demo, Get the latest version from <http://openjdk.java.net/install/>
4. yum install java-1.8\*

#yum -y install java-1.8.0-openjdk-devel

1. Confirm Java Version and set the java home
2. java -version
3. find /usr/lib/jvm/java-1.8\* | head -n 3
4. JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-<Java version which seen in the above output>
5. export JAVA\_HOME
6. PATH=$PATH:$JAVA\_HOME
7. # To set it permanently update your .bash\_profile

vi ~/.bash\_profile

1. Install Tomcat

sudo dnf install wget

wget <https://downloads.apache.org/tomcat/tomcat-8/v8.5.55/bin/apache-tomcat-8.5.55.tar.gz>

tar -xvzf apache-tomcat-8.5.55.tar.gz

cd apache-tomcat-8.5.55

find / -name context.xml

vi /opt/apache-tomcat-8.5.55/webapps/host-manager/META-INF/context.xml

vi /opt/apache-tomcat-8.5.55/webapps/manager/META-INF/context.xml

cd conf

vi tomcat-users.xml

./startup.sh

1. **Install Python**

 Install Python latest version (on Control node and Managed host)

yum install python3 -y

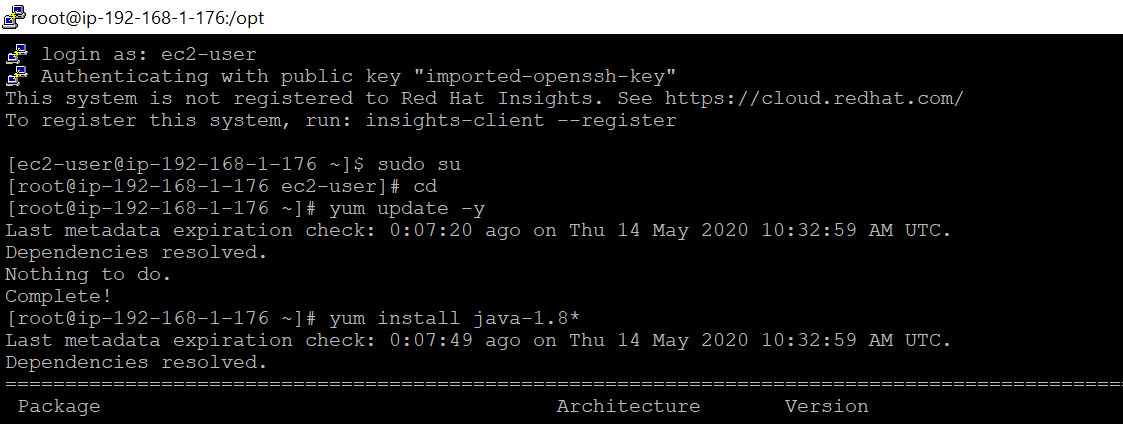
 By default, python3 is the command to run python commands. to use just python, use "alternatives" command. (on Control node and Managed host)

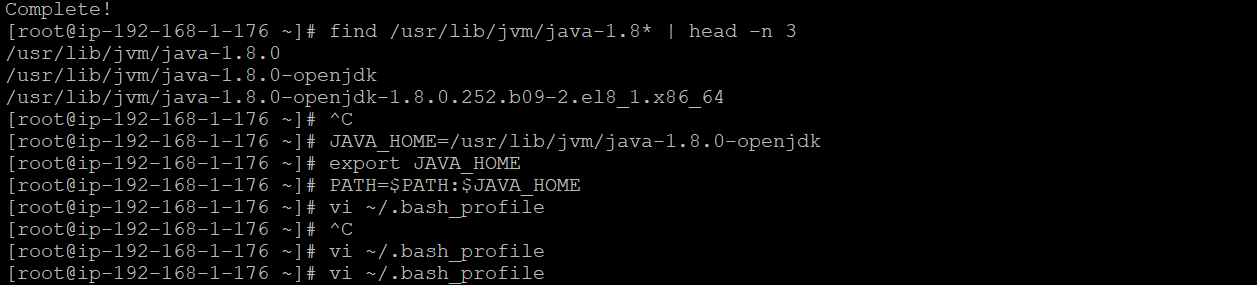
alternatives --set python /usr/bin/python3

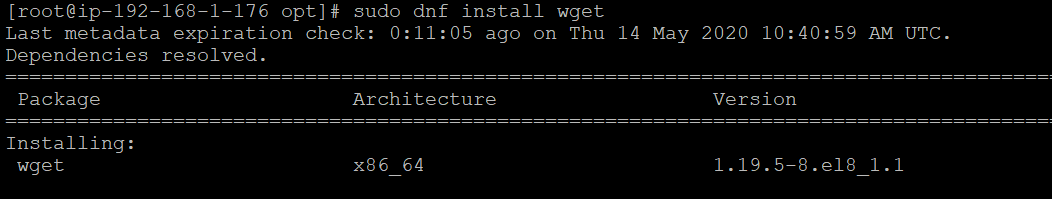
 Check for Python version

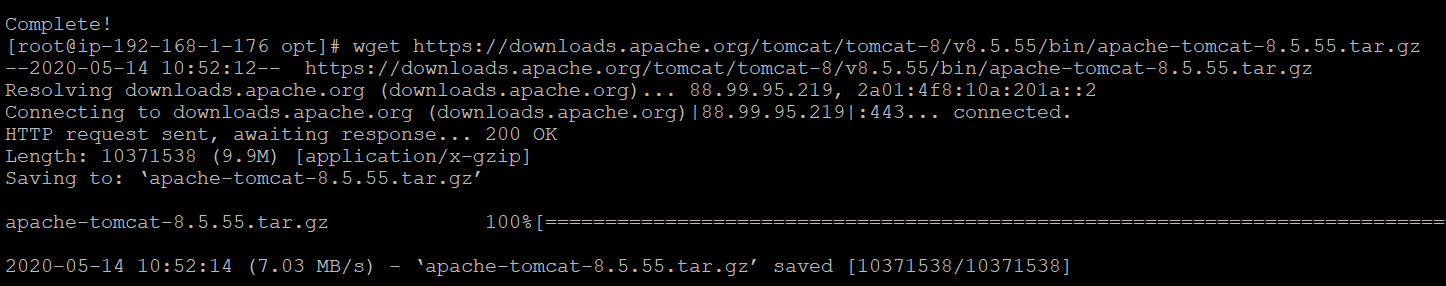
python --version

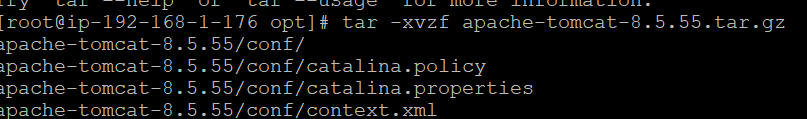
**Screenshots for Ref:**

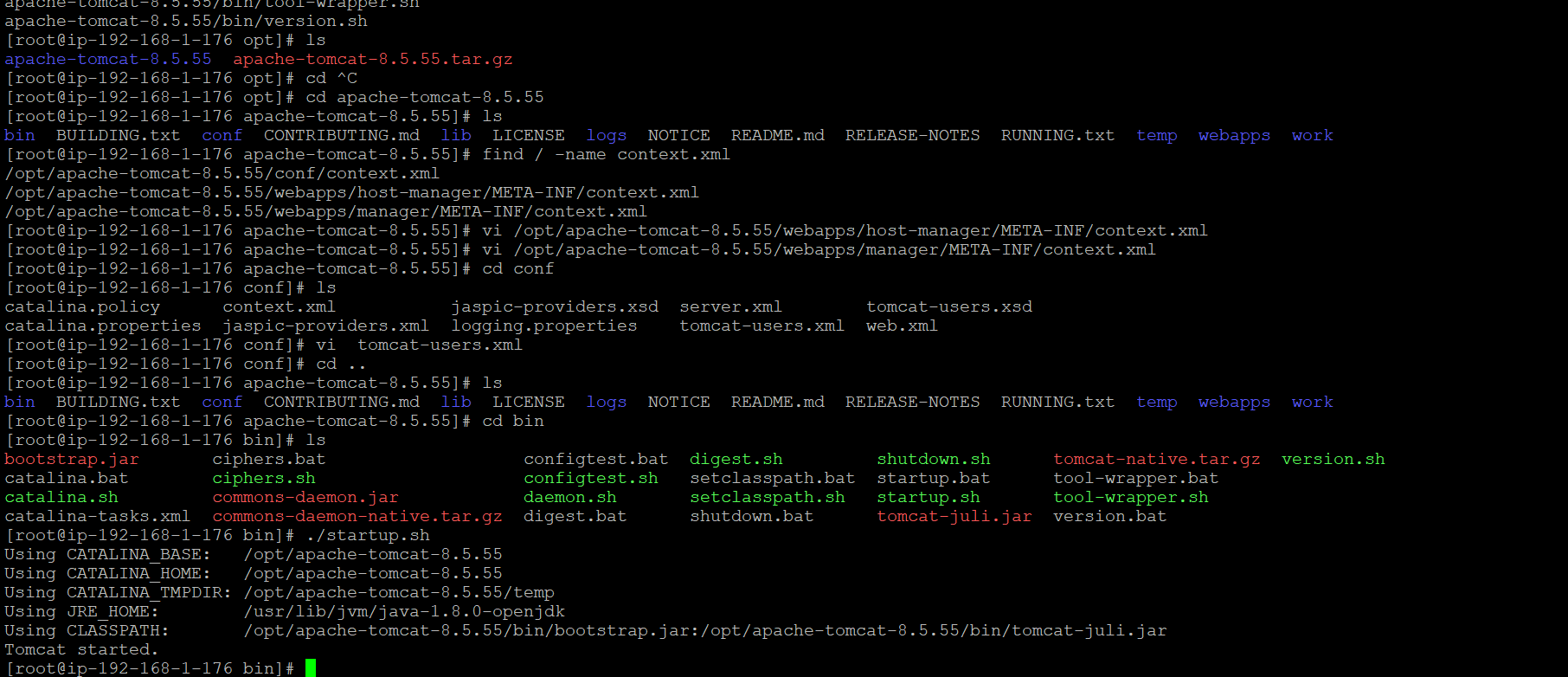


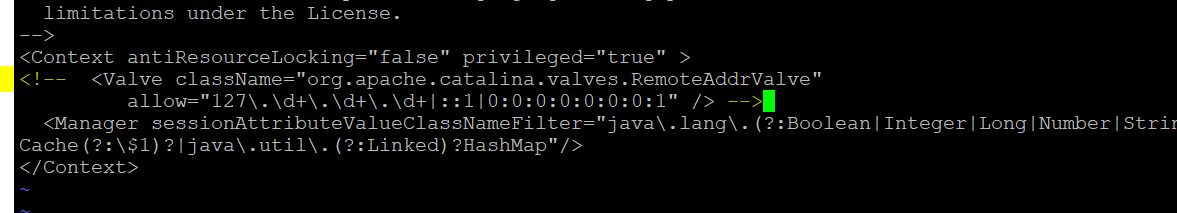


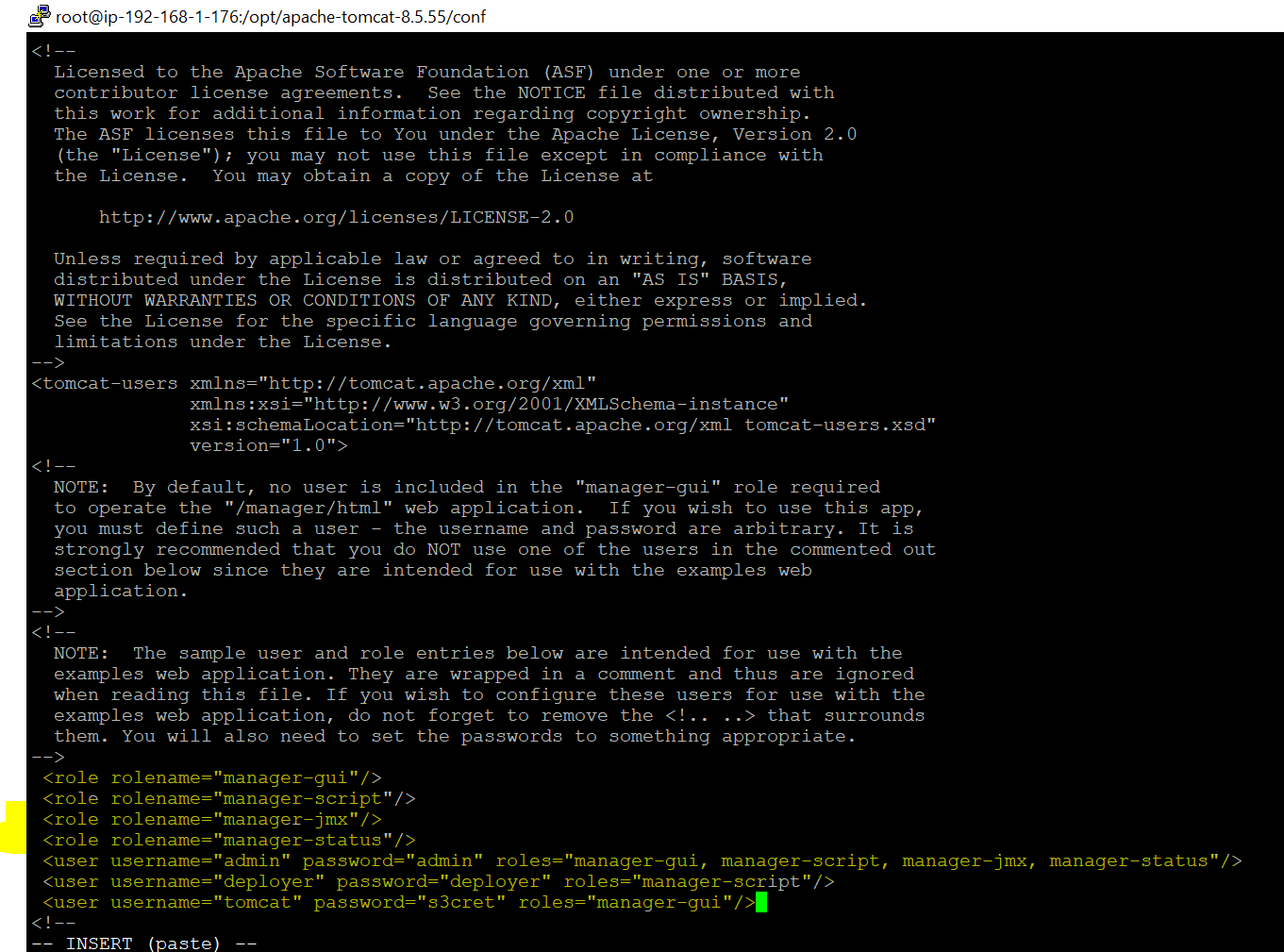


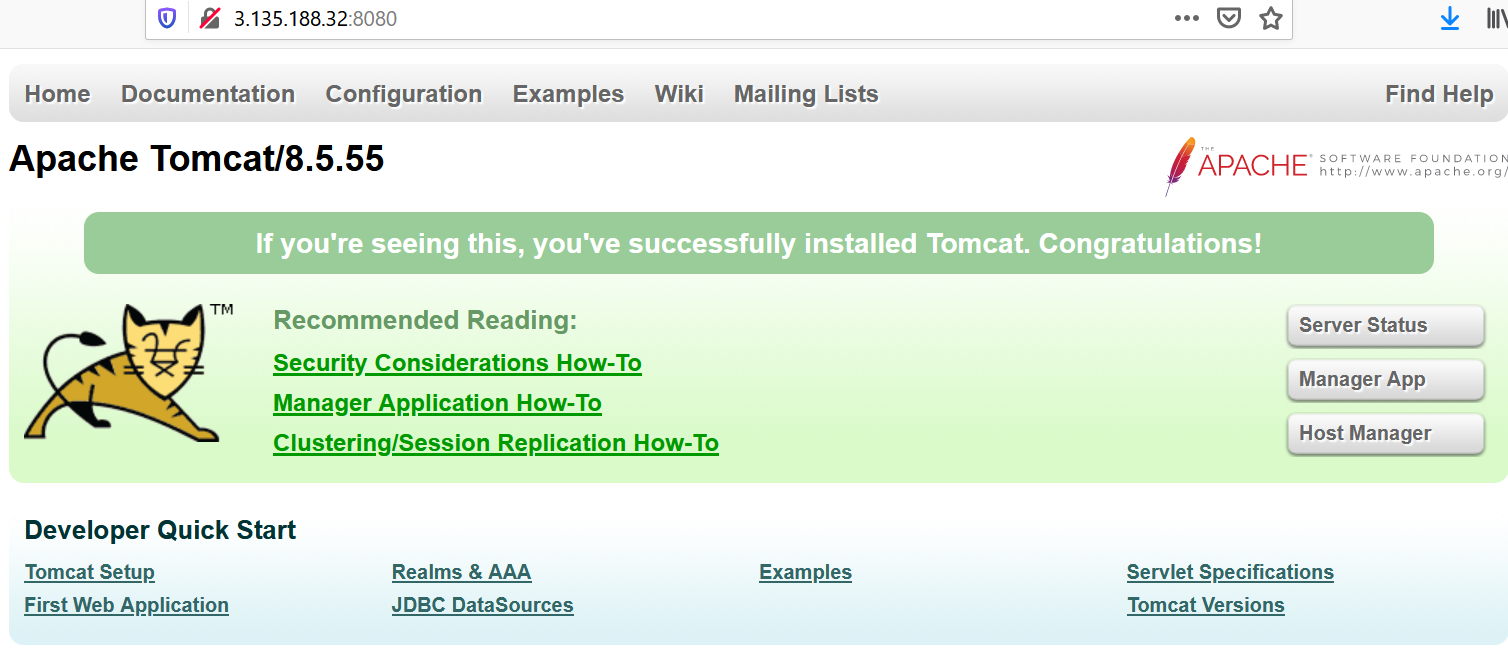


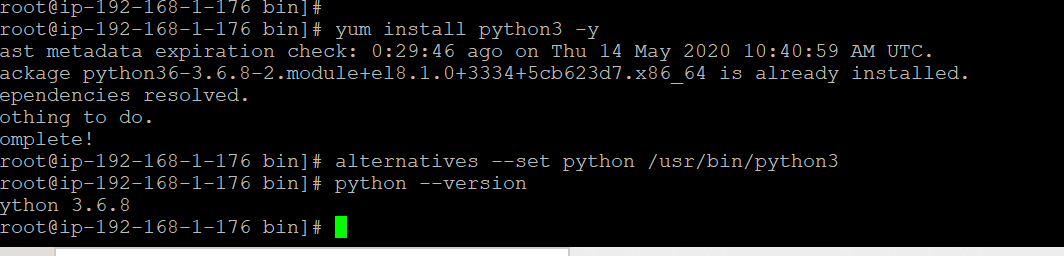












**Step-3:**

# Install Jenkins on AWS EC2

Jenkins is a self-contained Java-based program, ready to run out-of-the-box, with packages for Windows, Mac OS X and other Unix-like operating systems. As an extensible automation server, Jenkins can be used as a simple CI server or turned into the continuous delivery hub for any project.

### Prerequisites

1. EC2 Instance
   * With Internet Access
   * Security Group with Port 8080 open for internet
2. Java v1.8.x

## Install Java

1. We will be using open java for our demo, Get the latest version from <http://openjdk.java.net/install/>
2. yum install java-1.8\*

#yum -y install java-1.8.0-openjdk-devel

1. Confirm Java Version and set the java home
2. java -version
3. find /usr/lib/jvm/java-1.8\* | head -n 3
4. JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-<Java version which seen in the above output>
5. export JAVA\_HOME
6. PATH=$PATH:$JAVA\_HOME
7. # To set it permanently update your .bash\_profile

vi ~/.bash\_profile

The output should be something like this,

[root@~]# java -version

openjdk version "1.8.0\_151"

OpenJDK Runtime Environment (build 1.8.0\_151-b12)

OpenJDK 64-Bit Server VM (build 25.151-b12, mixed mode)

## Install Jenkins

You can install jenkins using the rpm or by setting up the repo. We will set up the repo so that we can update it easily in the future.

1. Get the latest version of jenkins from <https://pkg.jenkins.io/redhat-stable/> and install
2. yum -y install wget
3. sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
4. sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

yum -y install jenkins

### Start Jenkins

# Start jenkins service

service jenkins start

# Setup Jenkins to start at boot,

chkconfig jenkins on

### Accessing Jenkins

By default jenkins runs at port 8080, You can access jenkins at

http://YOUR-SERVER-PUBLIC-IP:8080

#### Configure Jenkins

* The default Username is admin
* Grab the default password
* Password Location:/var/lib/jenkins/secrets/initialAdminPassword
* Skip Plugin Installation; We can do it later
* Change admin password
  + Admin > Configure > Password
* Configure java path
  + Manage Jenkins > Global Tool Configuration > JDK
* Create another admin user id

**Install Python(For Ansible)**

 Install Python latest version (on Control node and Managed host)

yum install python3 -y

 By default, python3 is the command to run python commands. to use just python, use "alternatives" command. (on Control node and Managed host)

alternatives --set python /usr/bin/python3

 Check for Python version

python --version

**Install Git on Jenkins server**

1. Install git packages on jenkins server

yum install git -y

**Setup Git on jenkins console**

* Install git plugin without restart
  + Manage Jenkins > Jenkins Plugins > available > github
* Configure git path
  + Manage Jenkins > Global Tool Configuration > git

**Install Maven on Jenkins Server**

mkdir /opt/maven

cd /opt/maven

# downloading maven version 3.6.0

wget http://mirrors.estointernet.in/apache/maven/maven-3/3.6.1/binaries/apache-maven-3.6.1-bin.tar.gz

tar -xvzf apache-maven-3.6.1-bin.tar.gz

Setup M2\_HOME and M2 paths in .bash\_profile of the user and add these to the path variable

vi ~/.bash\_profile

M2\_HOME=/opt/maven/apache-maven-3.6.1

M2=$M2\_HOME/bin

PATH=<Existing\_PATH>:$M2\_HOME:$M2

**Setup maven on Jenkins console**

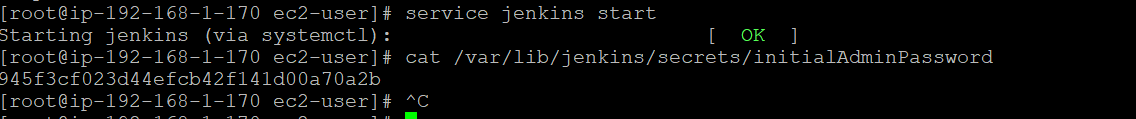
1. Install maven plugin without restart

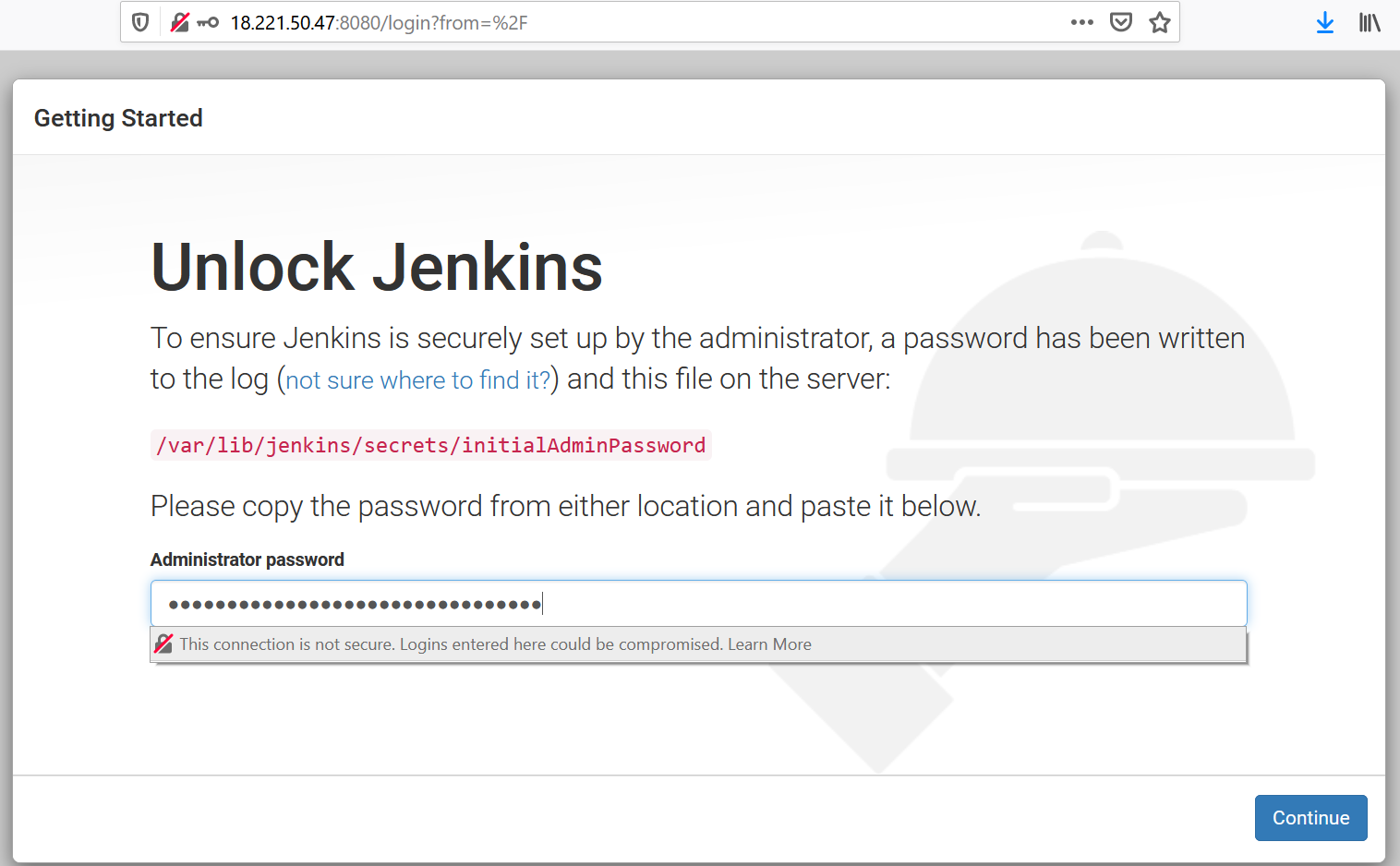
* Manage Jenkins > Jenkins Plugins > available > Maven Invoker
* Manage Jenkins > Jenkins Plugins > available > Maven Integration

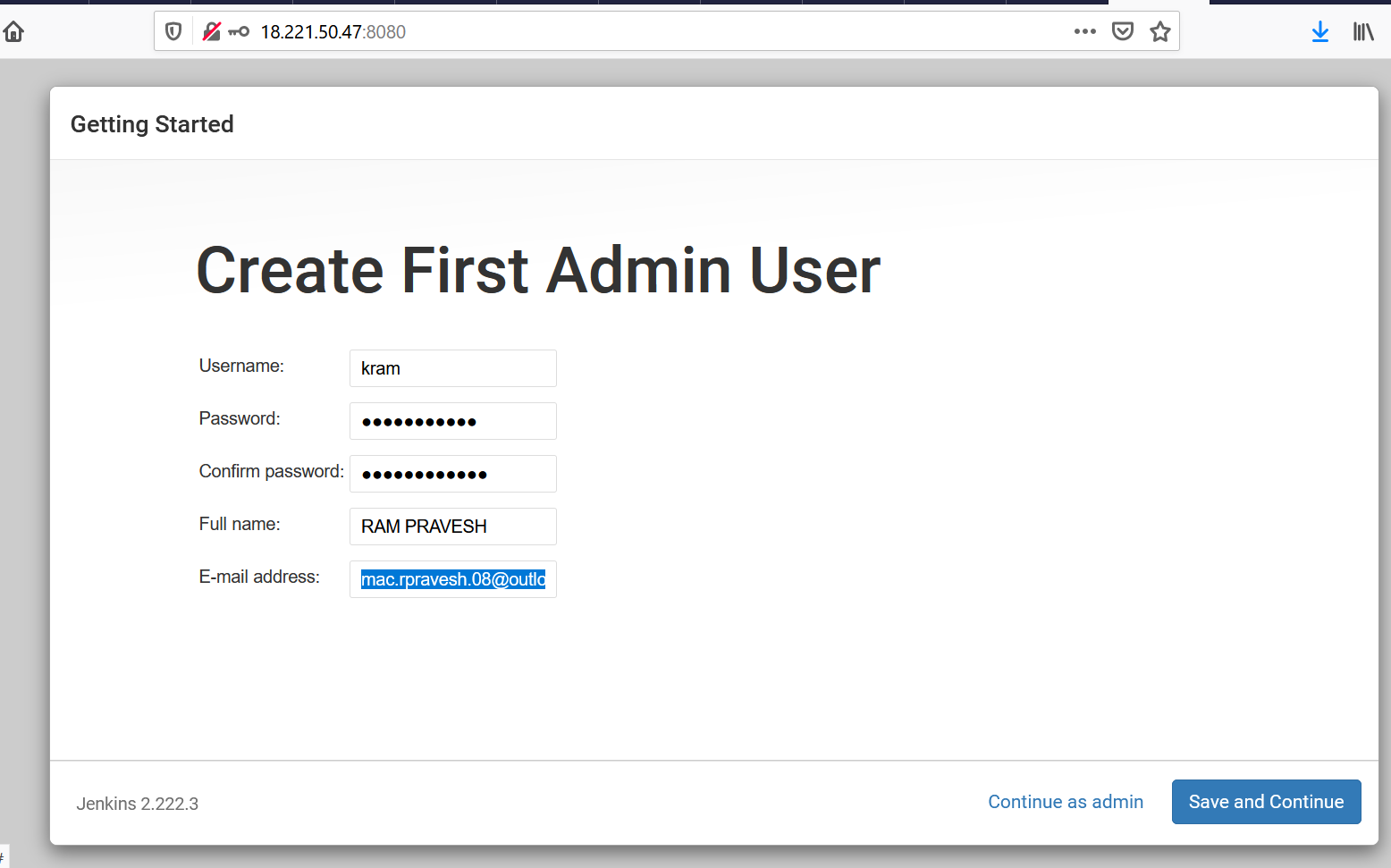
1. Configure maven path

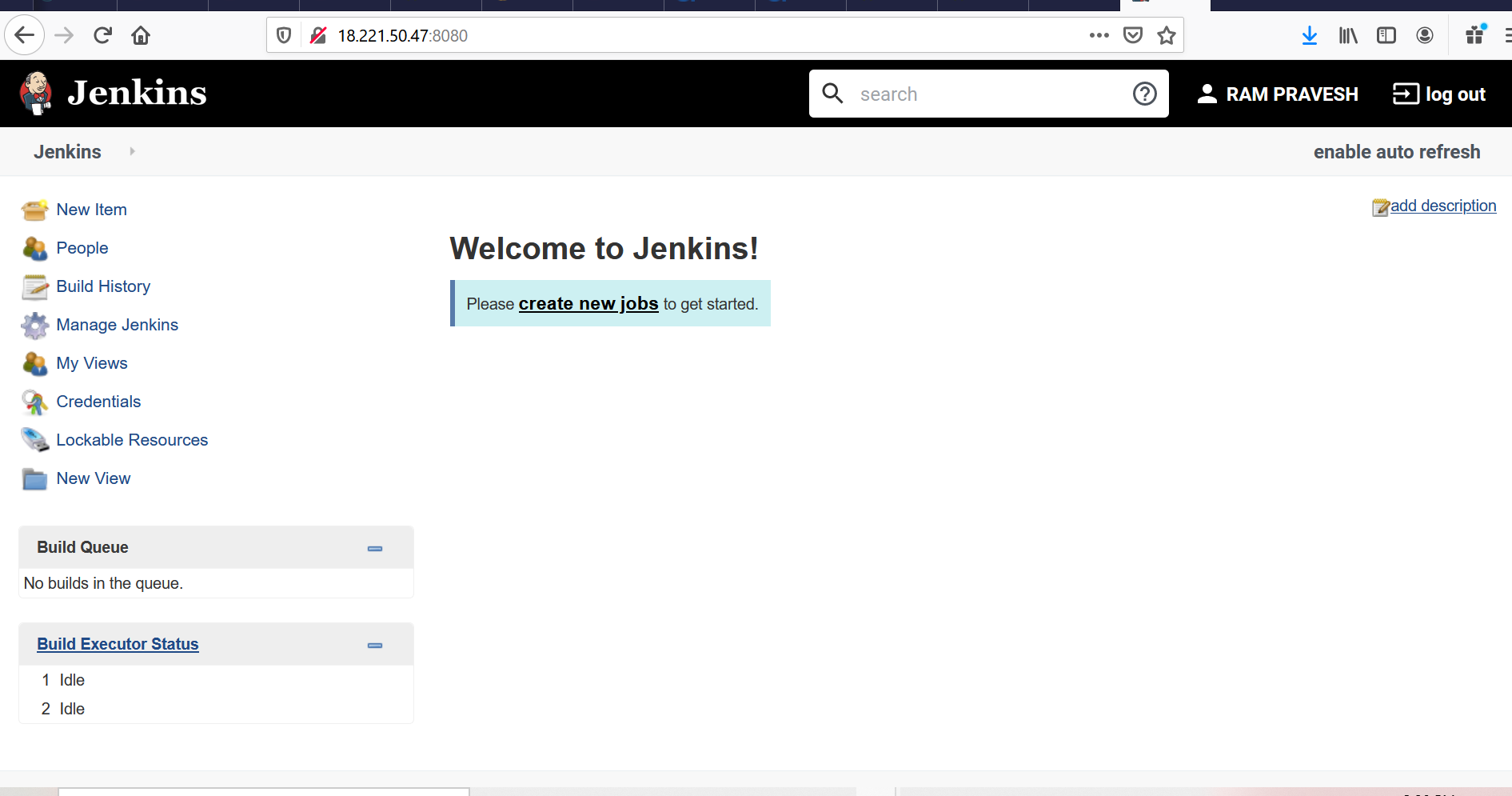
* Manage Jenkins > Global Tool Configuration > Maven

**Screenshot in below -**









**Step-4:**

**Ansible Server Configurations**

#### On RHEL 8.x server

#### Install Python:

1. Install Python latest version (on Control node and Managed host)

yum install python3 -y

1. By default, python3 is the command to run python commands. to use just python, use "alternatives" command. (on Control node and Managed host)

alternatives --set python /usr/bin/python3

1. Check for Python version

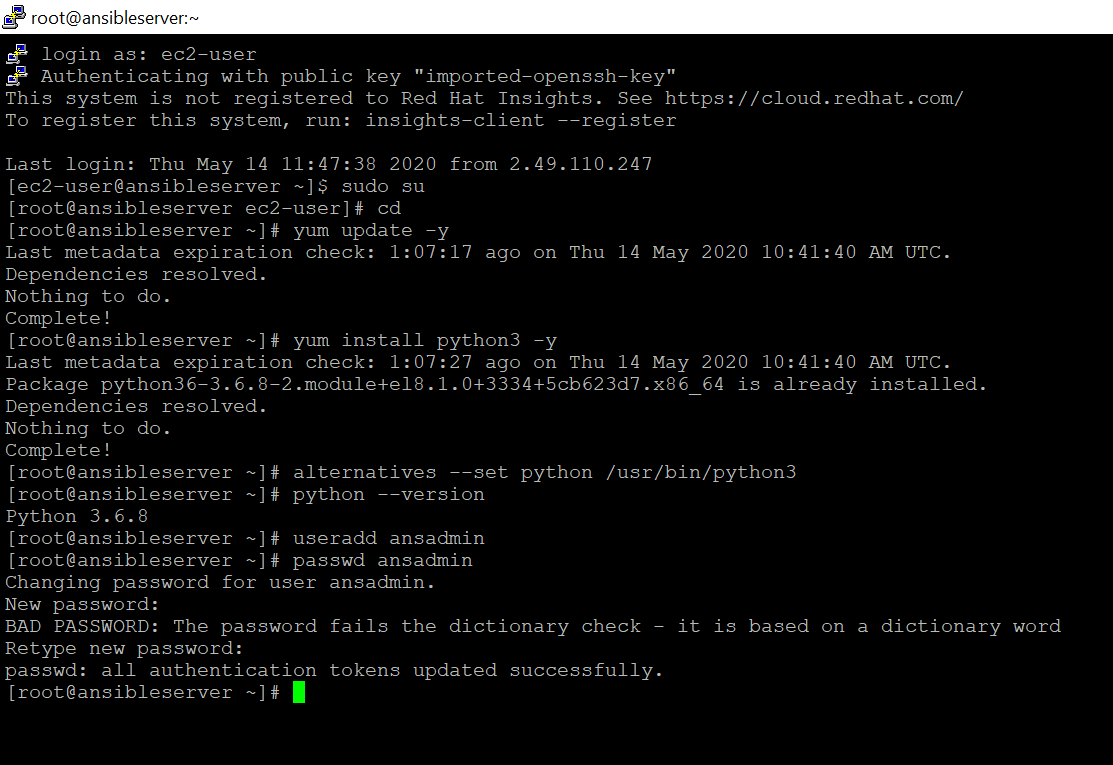
python –version

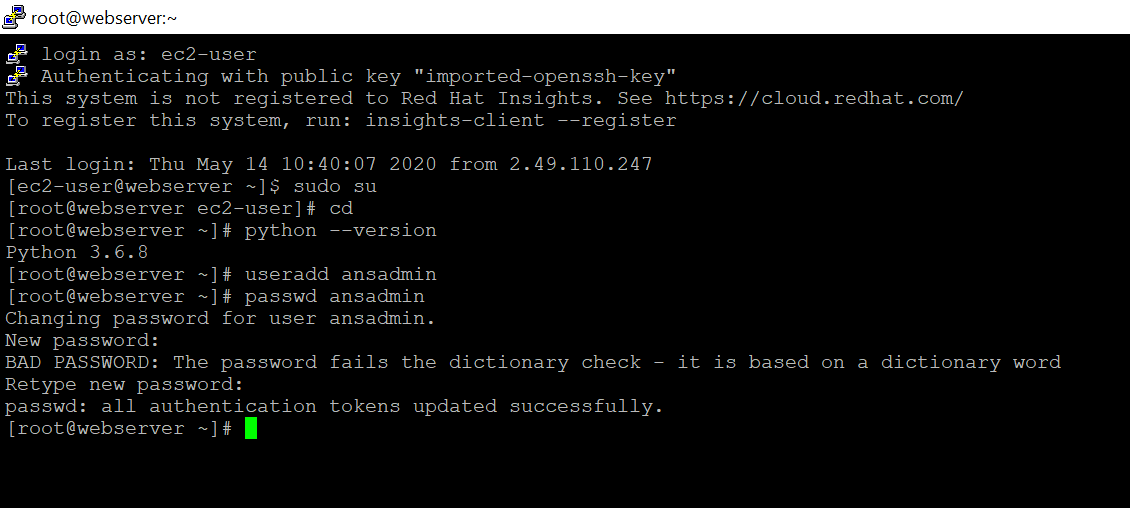
**Create a new user** for ansible administration & grant admin access to the user (on Ansible Machine and Tomcat Server Machine)

useradd ansadmin

passwd ansadmin

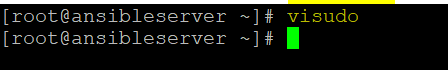
**Screenshot for Ref:**

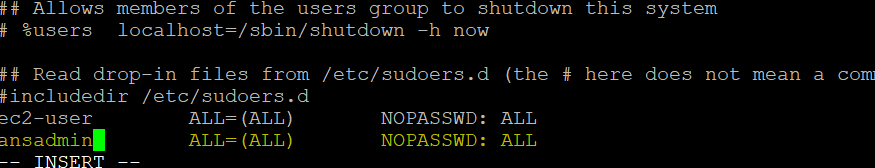




 Below command adds ansadmin to sudoers file. But we strongly recommended using "visudo" command if you are aware vi or nano editor. (on Ansible Machine and Tomcat Server Machine)

ansadmin ALL=(ALL) NOPASSWD: ALL /etc/sudoers

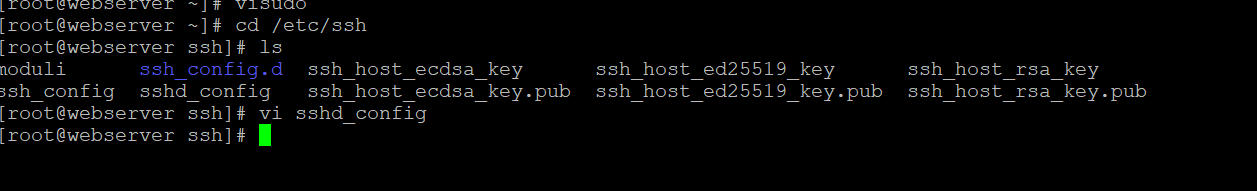


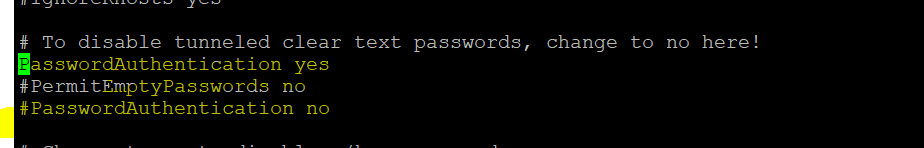


**Using key-based authentication is advised**. If you are still at the learning stage use password-based authentication (on Ansible Machine and Tomcat Server Machine)

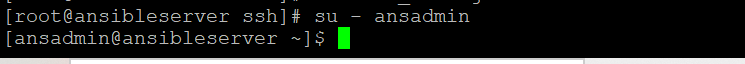
cd /etc/ssh

vi sshd\_config

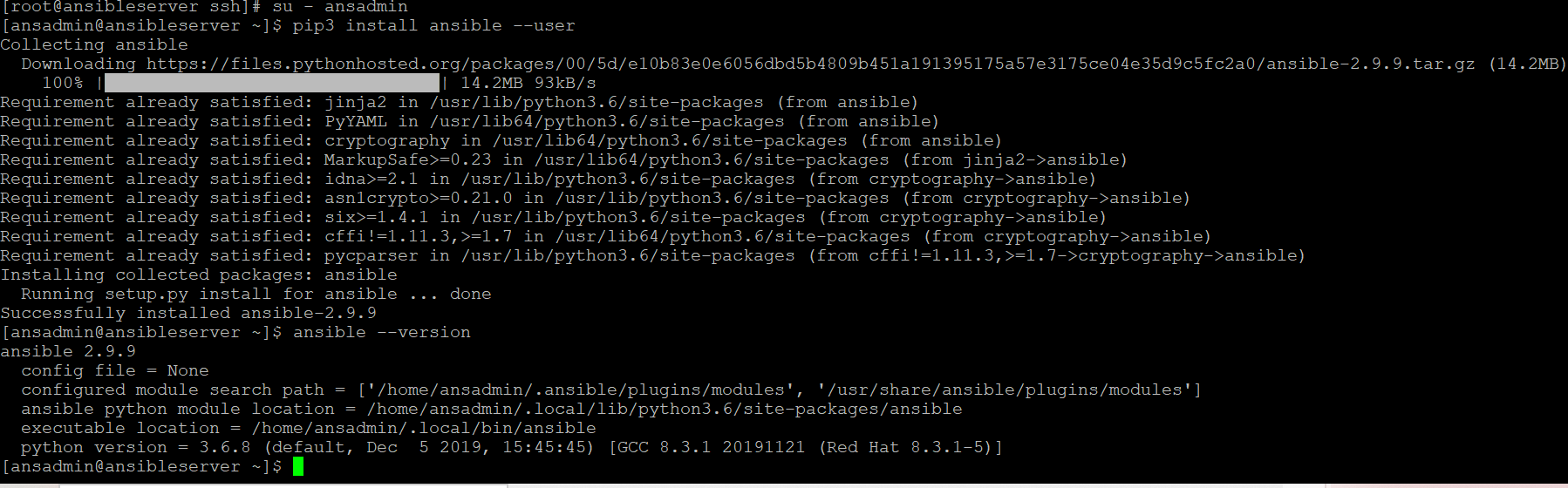




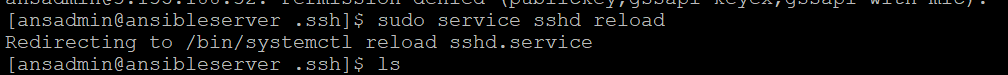
#### Install Ansible as a ansadmin user (on Ansible Machine)



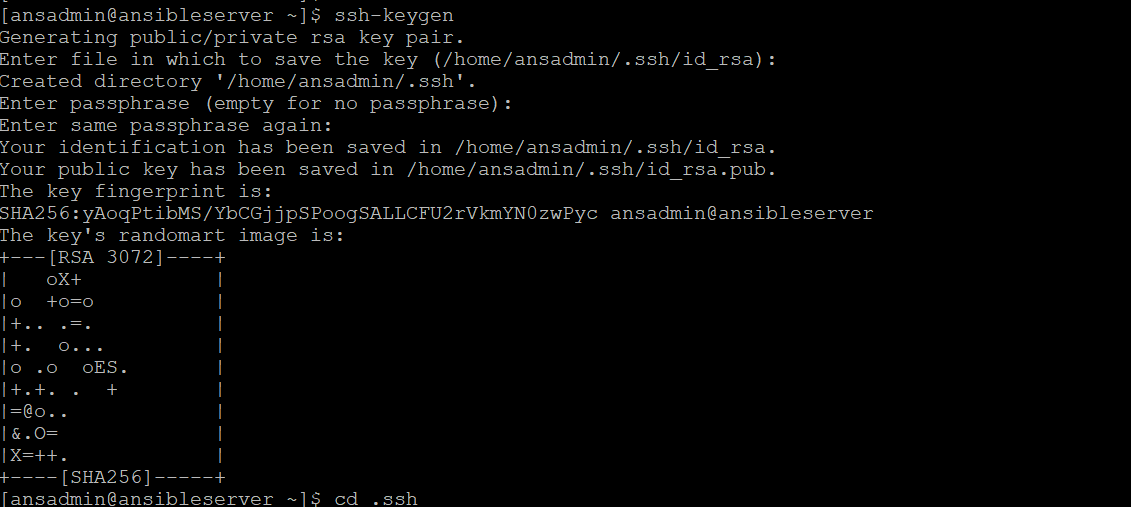
pip3 install ansible --user



sudo service sshd reload

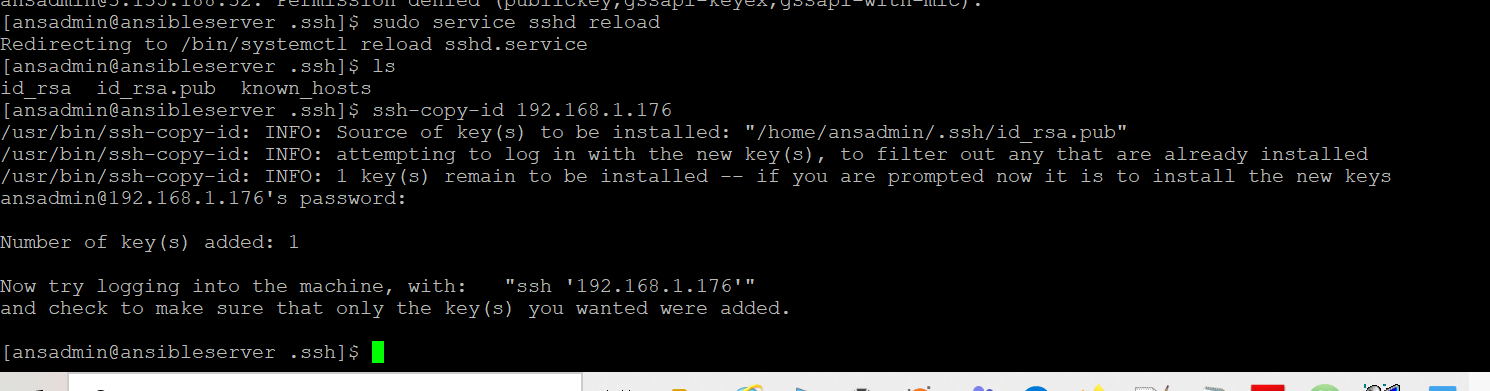


Log in as a ansadmin user on master and generate ssh key (on Control node)



 Copy keys onto all ansible managed hosts (on Control node)

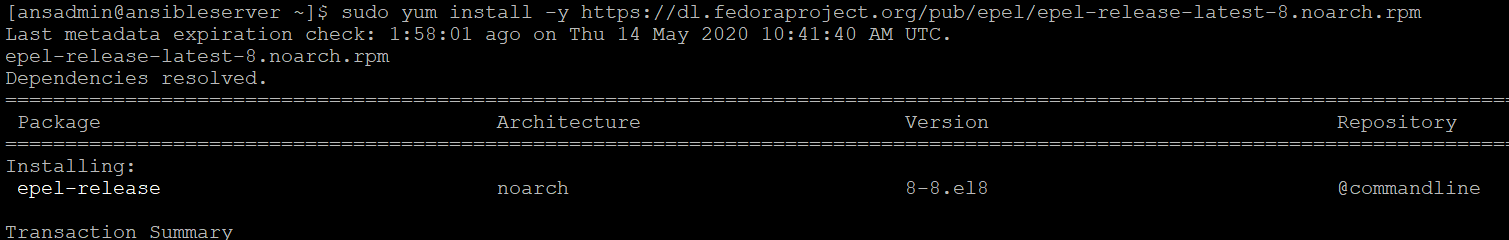
ssh-copy-id ansadmin@

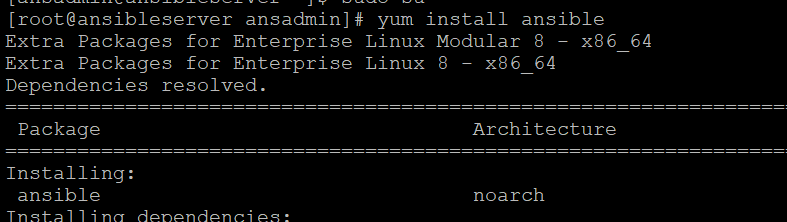


**Install ansible package**

sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm

yum install ansible

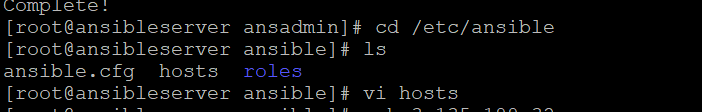


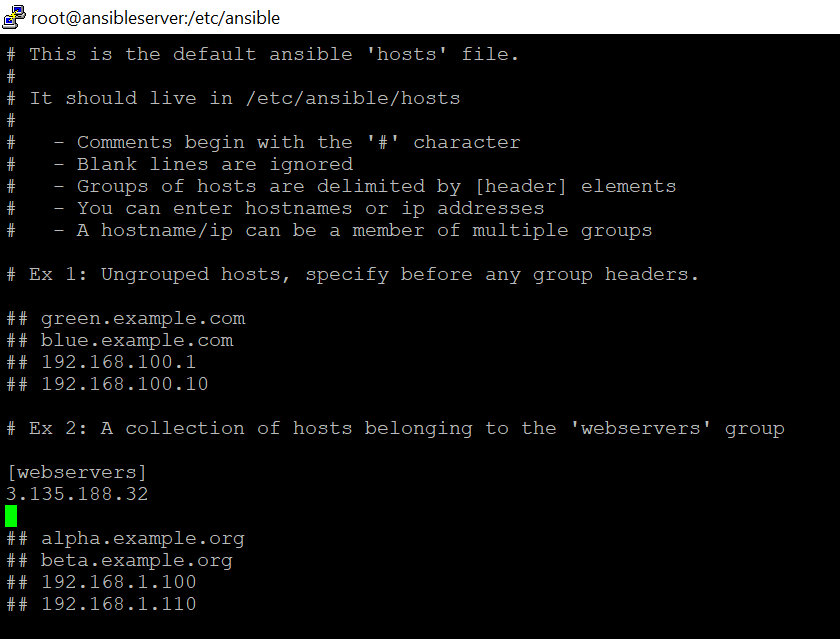


### Validation test

1. Create a directory /etc/ansible and create an inventory file called "hosts" add control node IP address in it.
2. Run ansible command as ansadmin user it should be successful (Master)

ansible all -m ping







Now, we can observer password less access via SSH is enabled b/w ansible and tomcat webserver machines. Hence, Let’s configure the Jenkins Jobs, Access of folder location in ansible server from Jenkins.

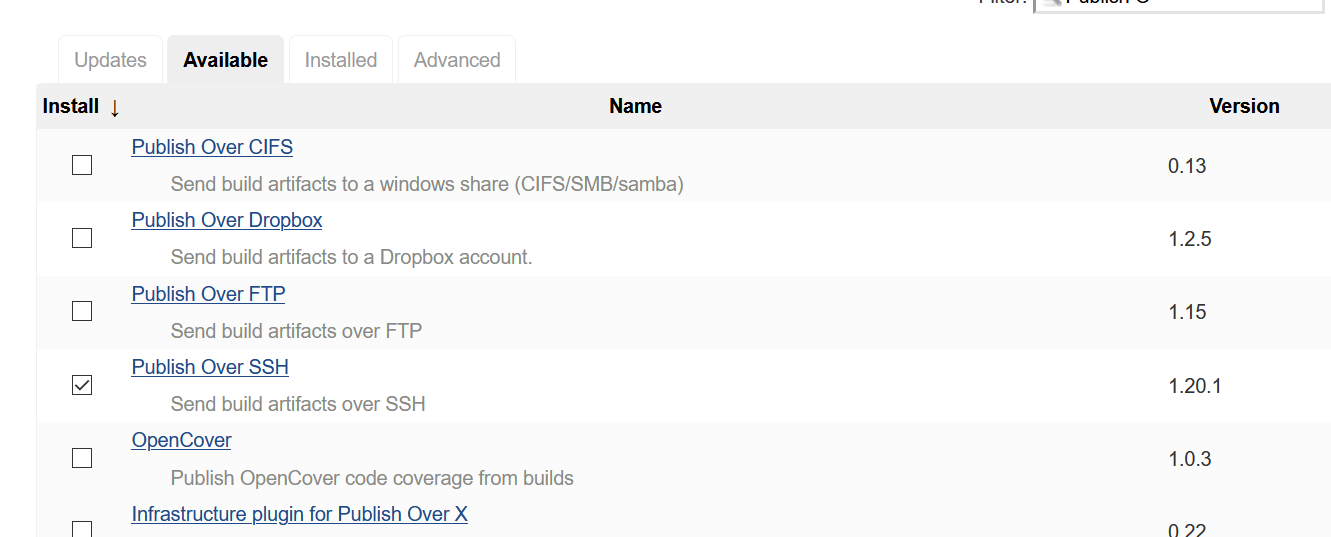
**Step-5:**

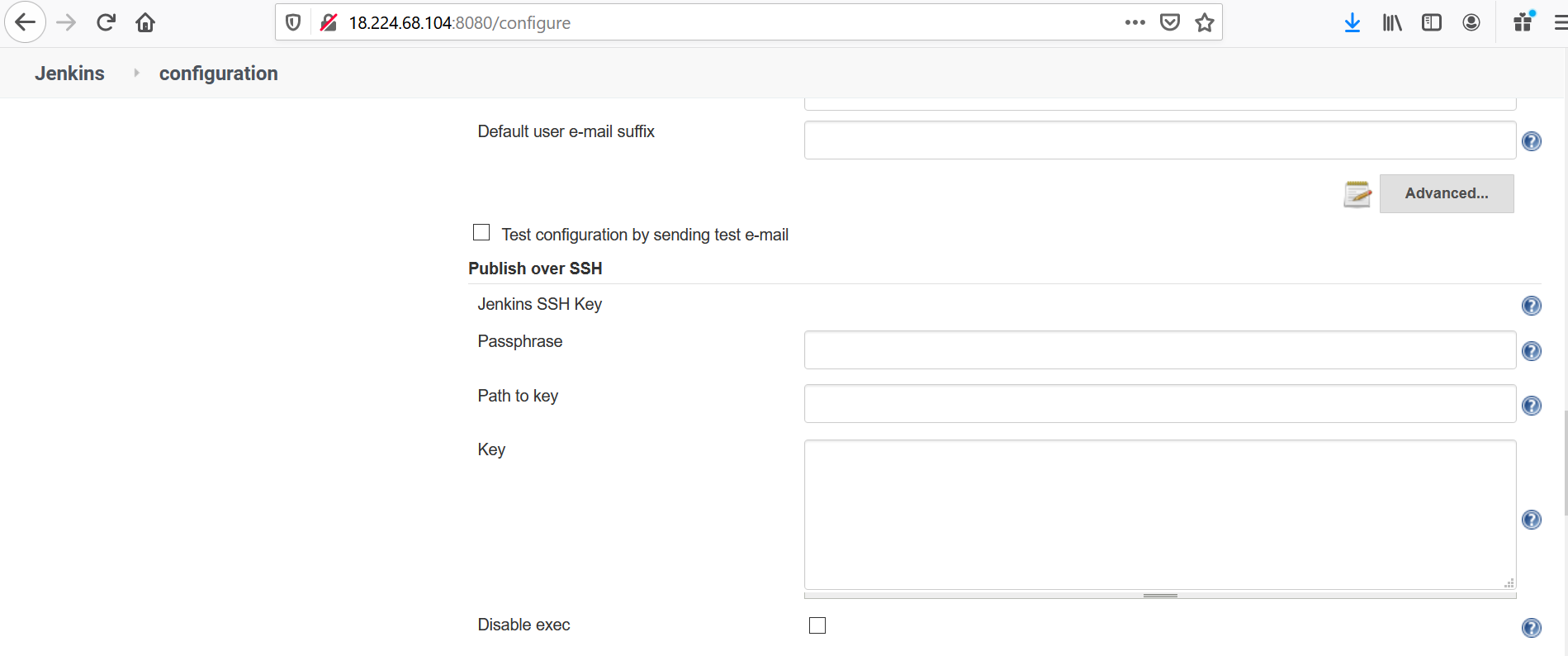
Let’s configure the Jenkins Jobs, Access of folder location in ansible server from Jenkins and Deploy .war in Tomcat Webserver.

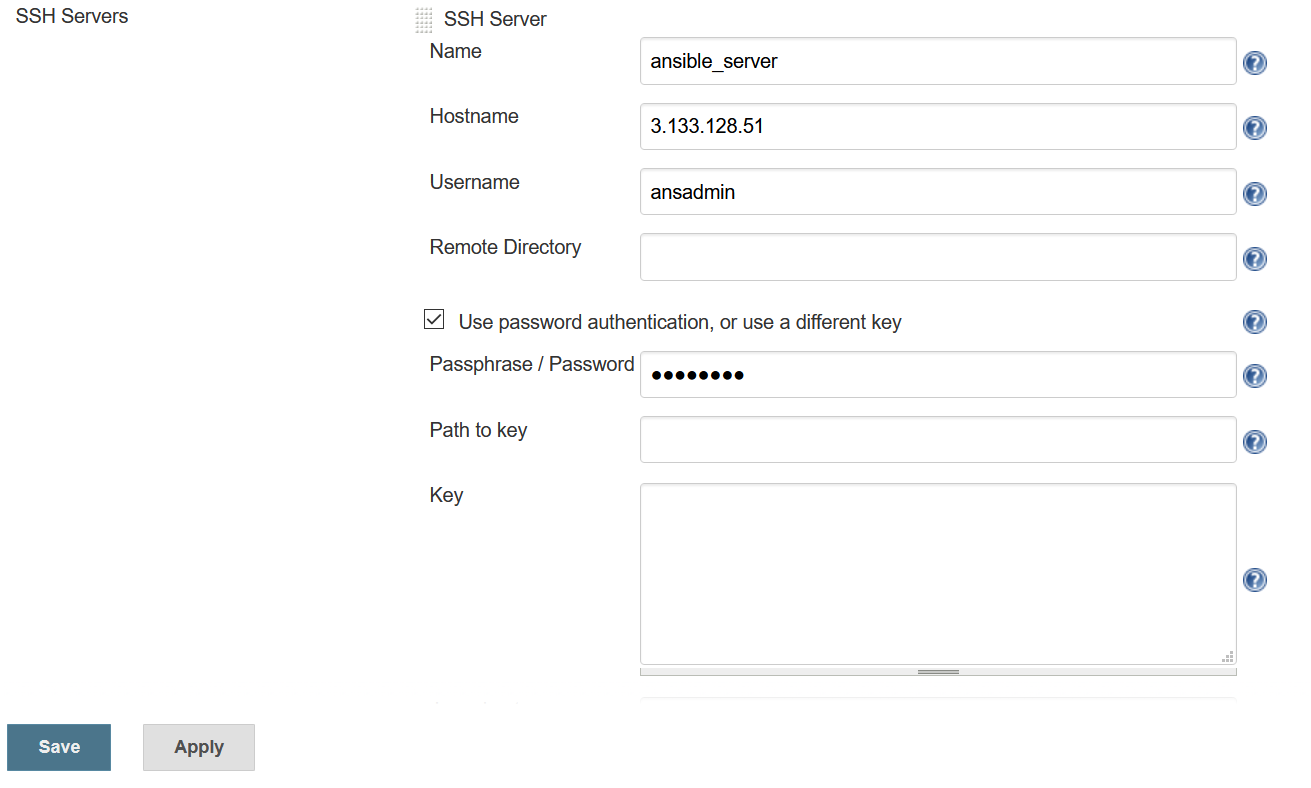
Go to Jenkins Console

**Setup Publish Over ssh on jenkins console**

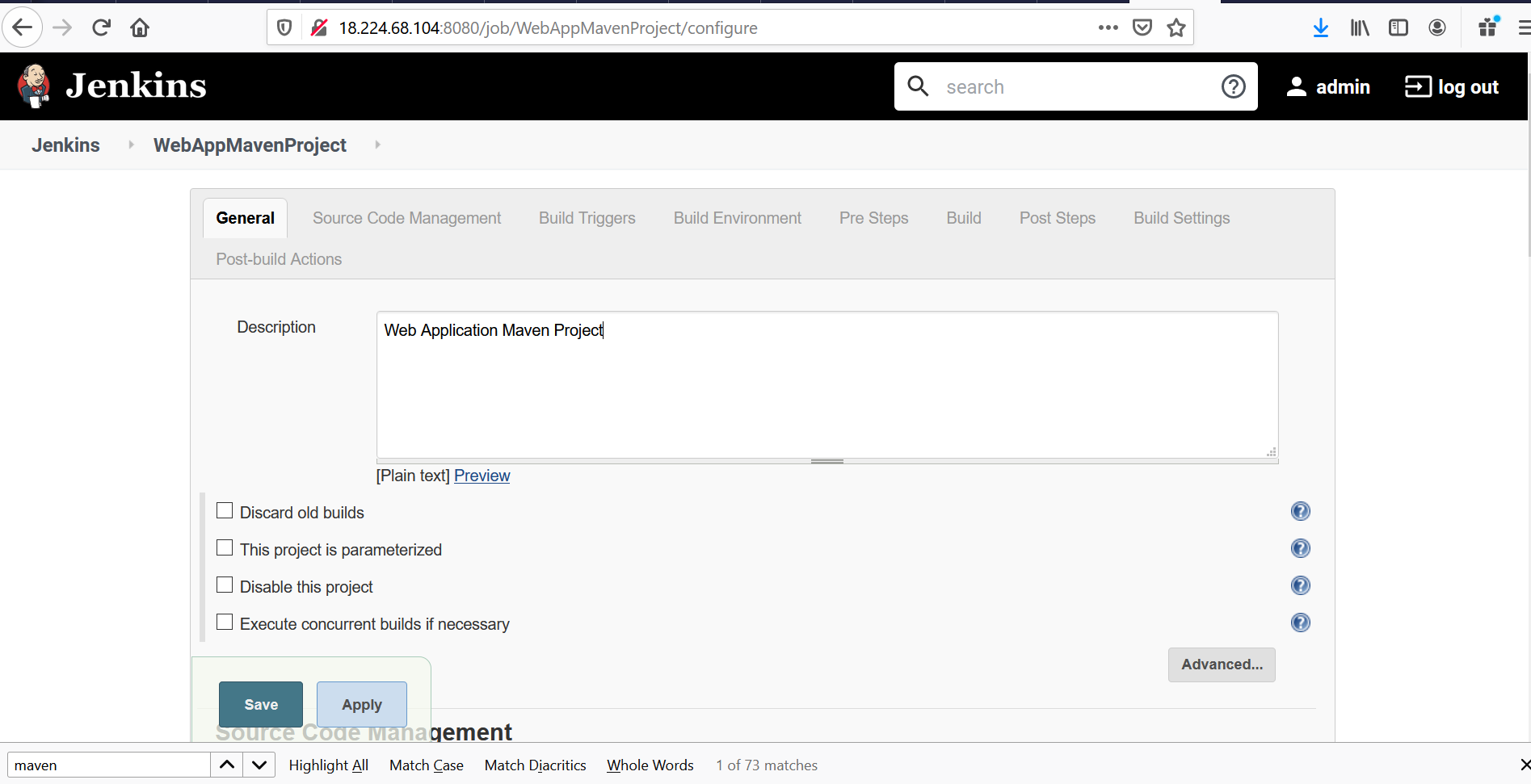
Access of folder location in ansible server from Jenkins

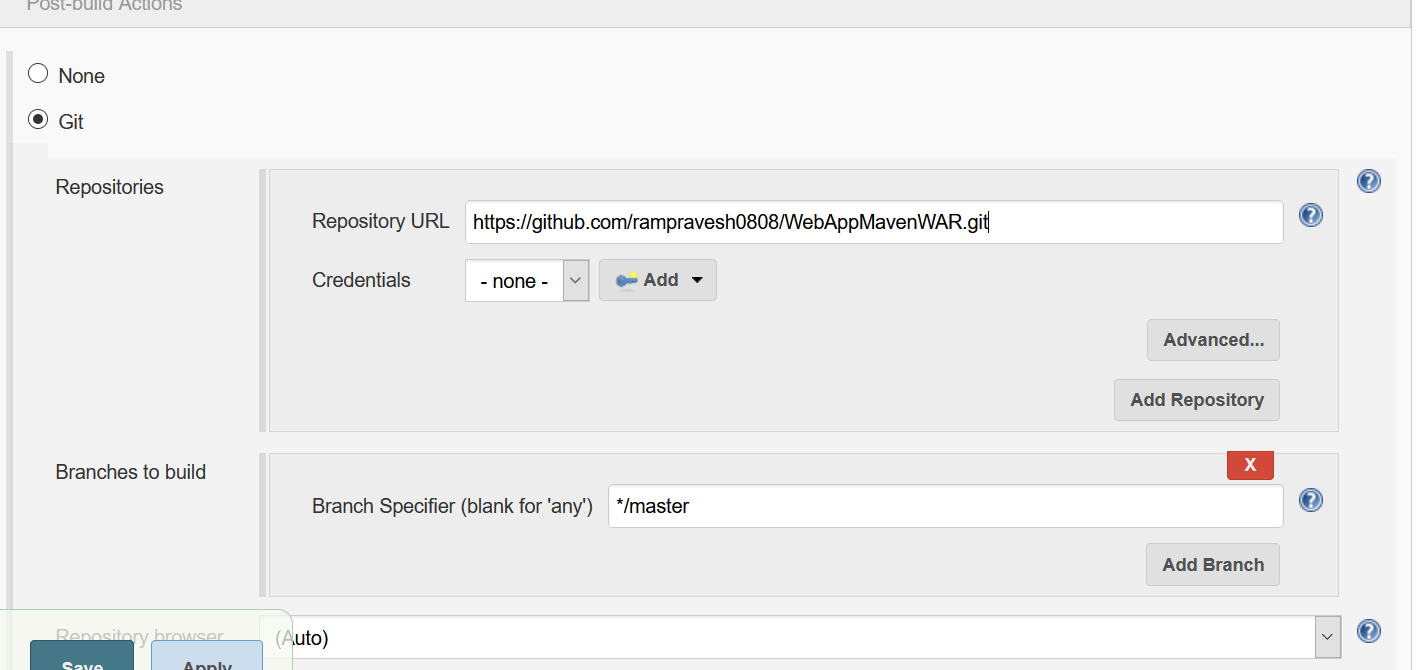






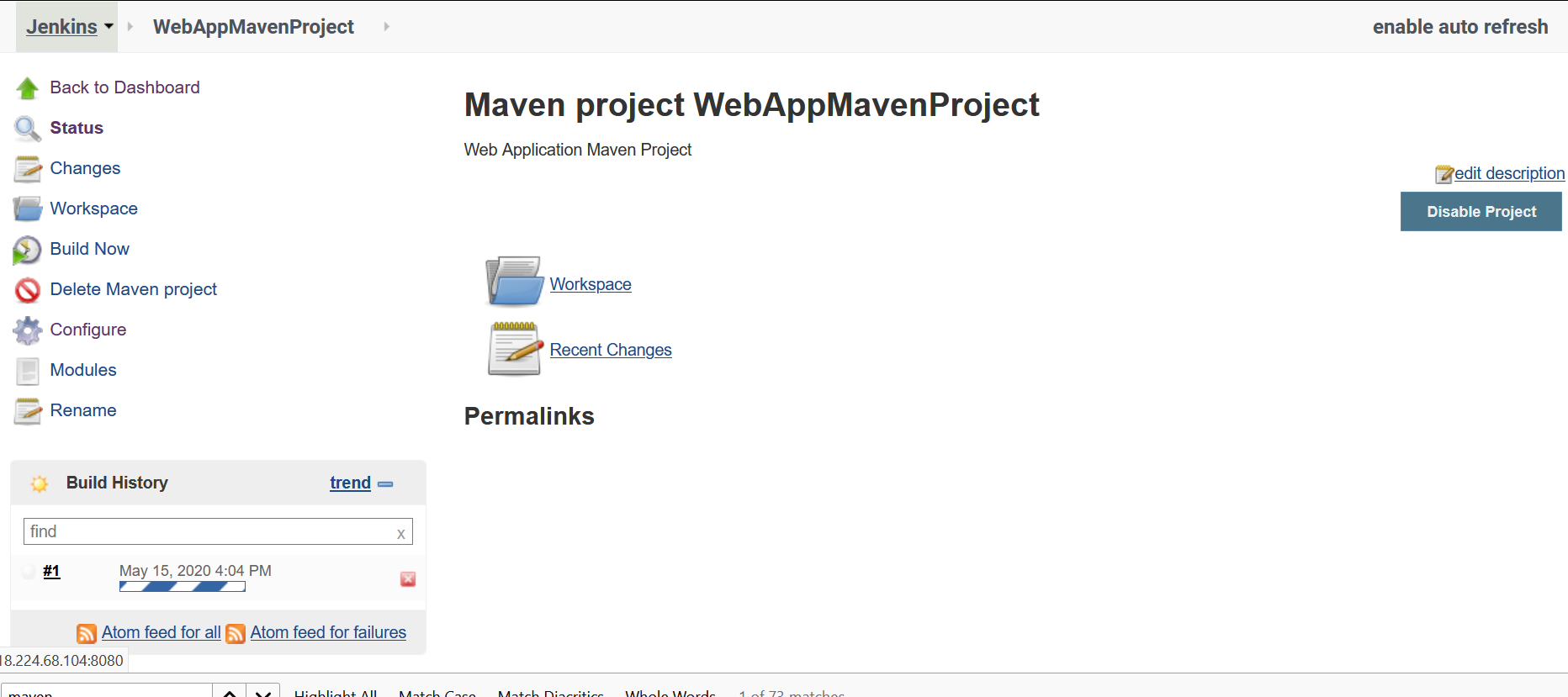
**Now, Create Jenkins Job**



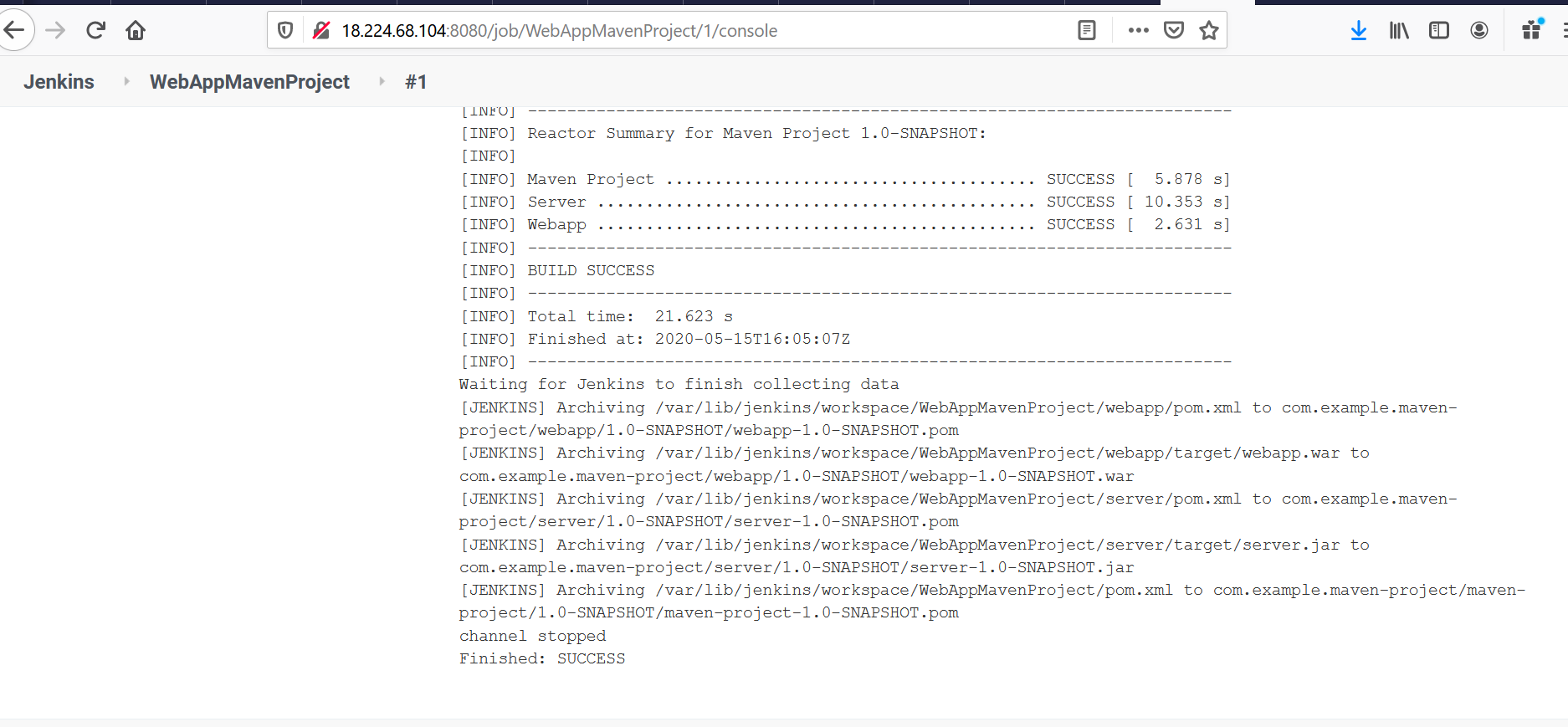


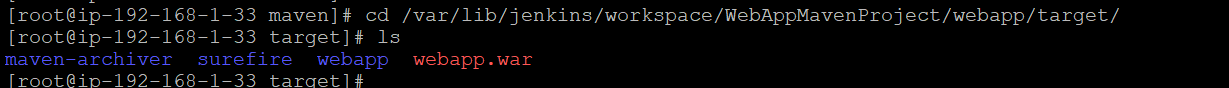


Now, Run the Job and confirm build is happening properly

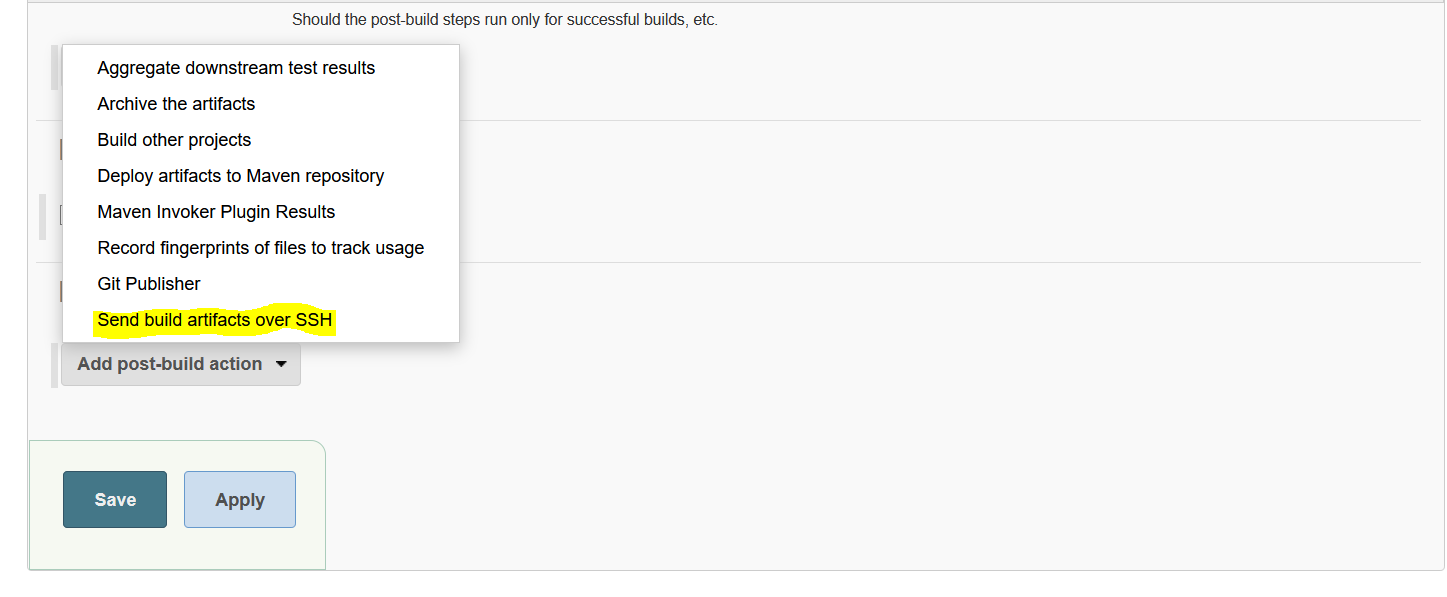


**We can observe that war file generated successfully in Jenkins server, Below screenshot for ref:**



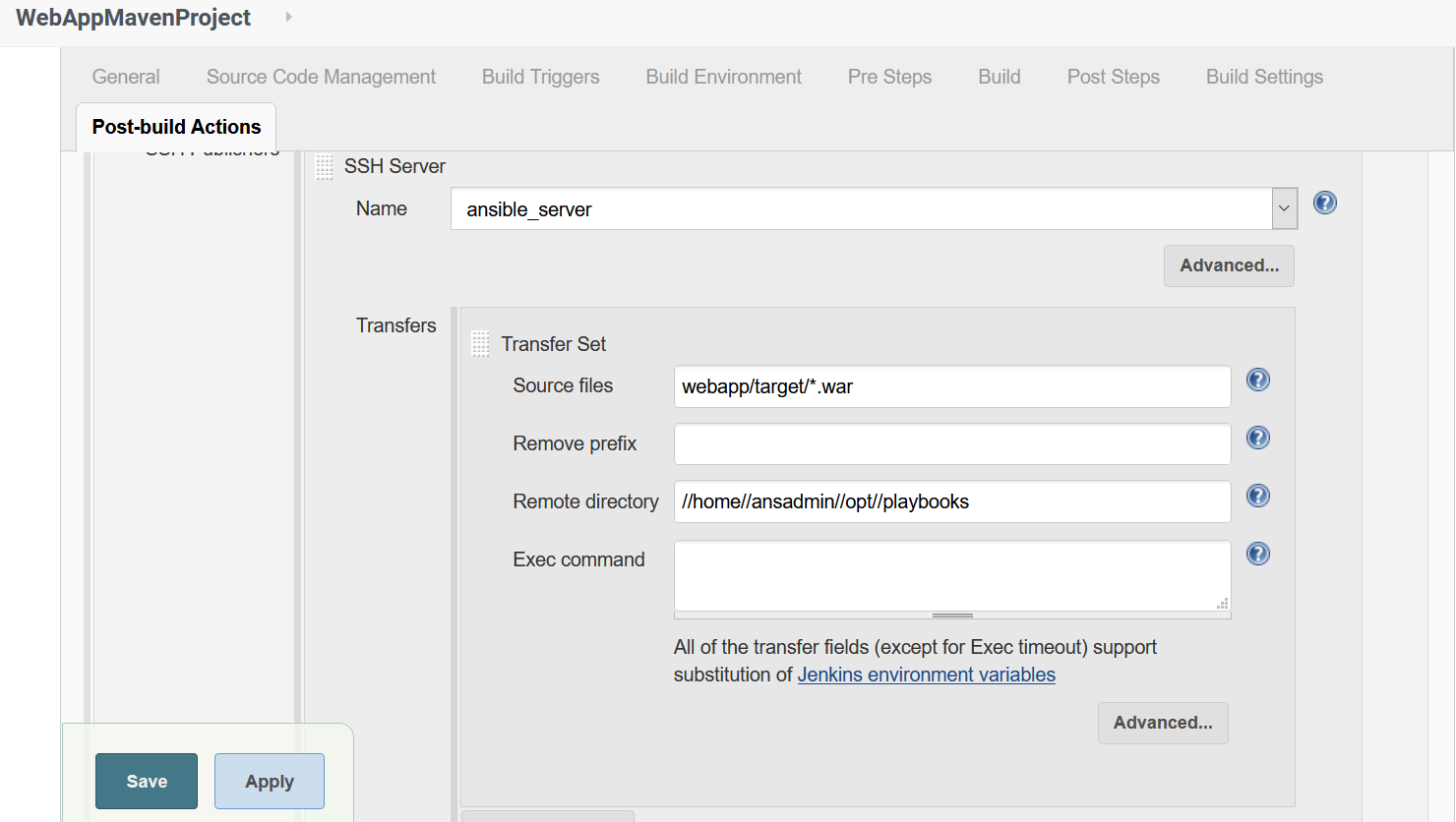


Now, Configure Publish Over SSH path so that .war(artifacts) copied to ansible server.



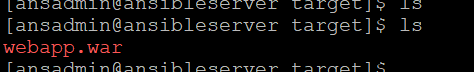
Check folder location in ansible server.





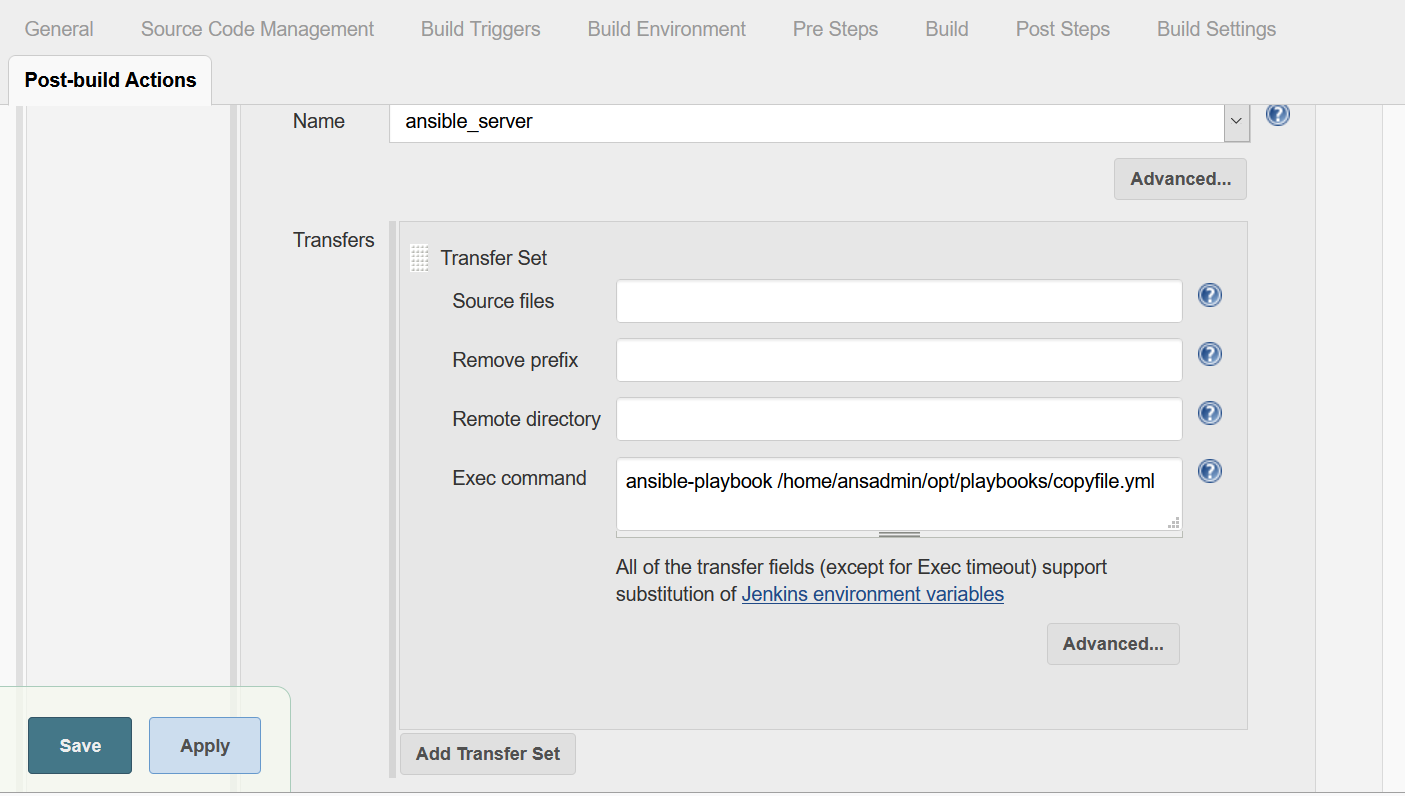
Try and Run the build…..

Expected: war should be copied at location(/home/ansadmin/opt/playbooks/webapp/target) on ansible server.



Configure ansible command to run **playbooks** on **ansible server**.

**Copyfile.yml** is playbooks which will deploy .war file into Tomcat server.



**Save.**

Create Copyfile.yml and save for below content.

---

- hosts: webservers

become: true

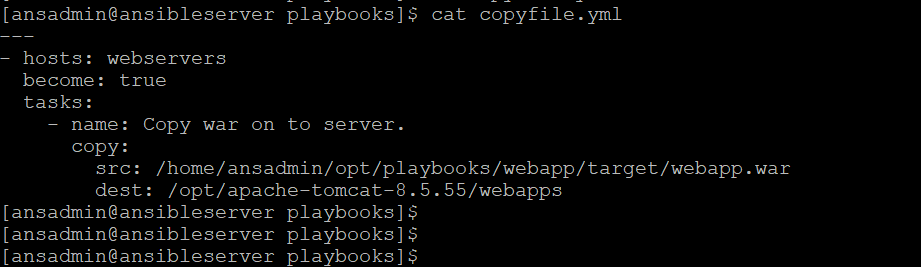
tasks:

- name: Copy war on to server.

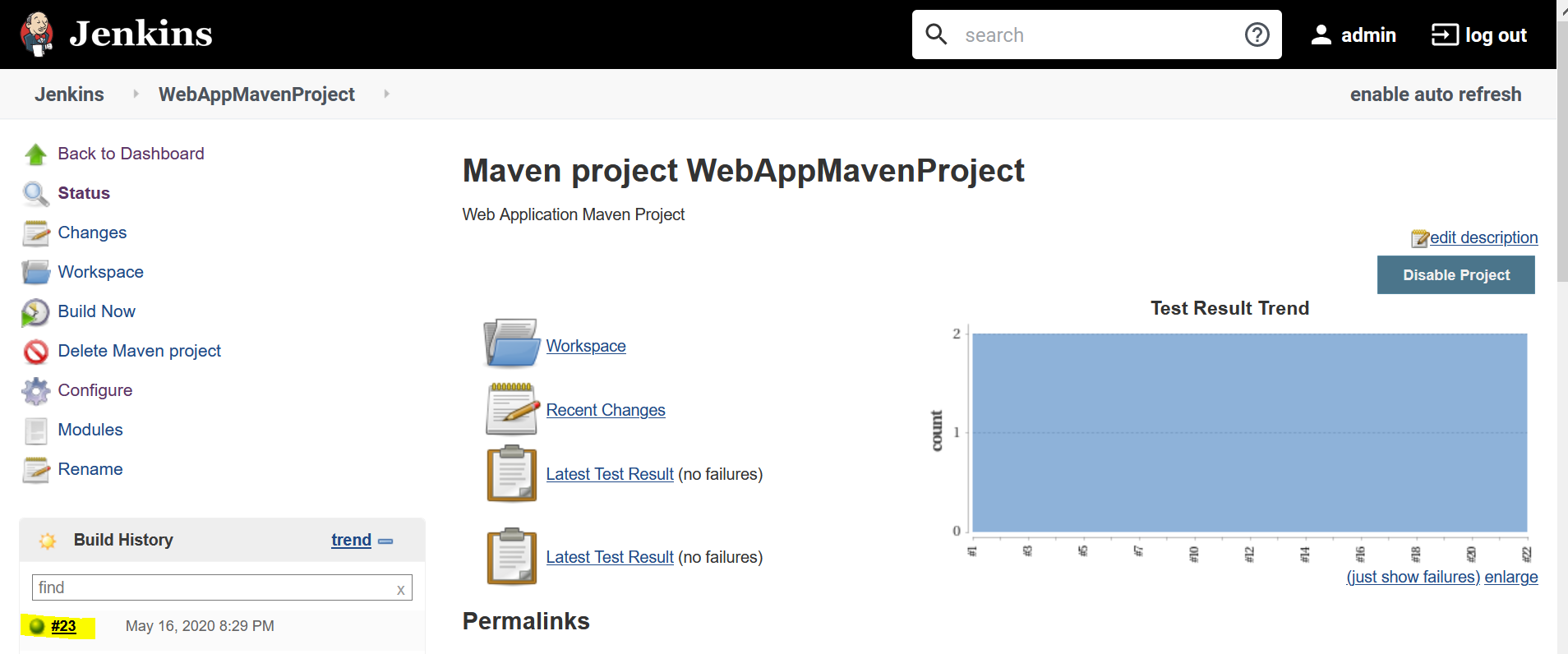
copy:

src: /opt/playbooks/webapp/target/webapp.war

dest: /opt/apache-tomcat-8.5.55/webapps



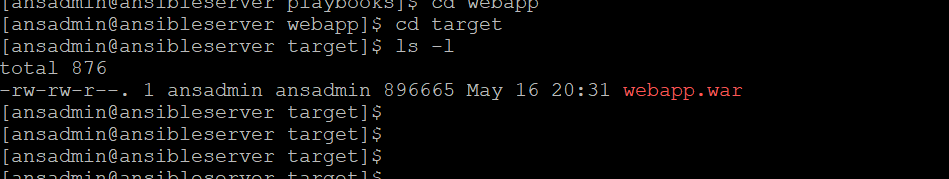
**Now, Run the build.**



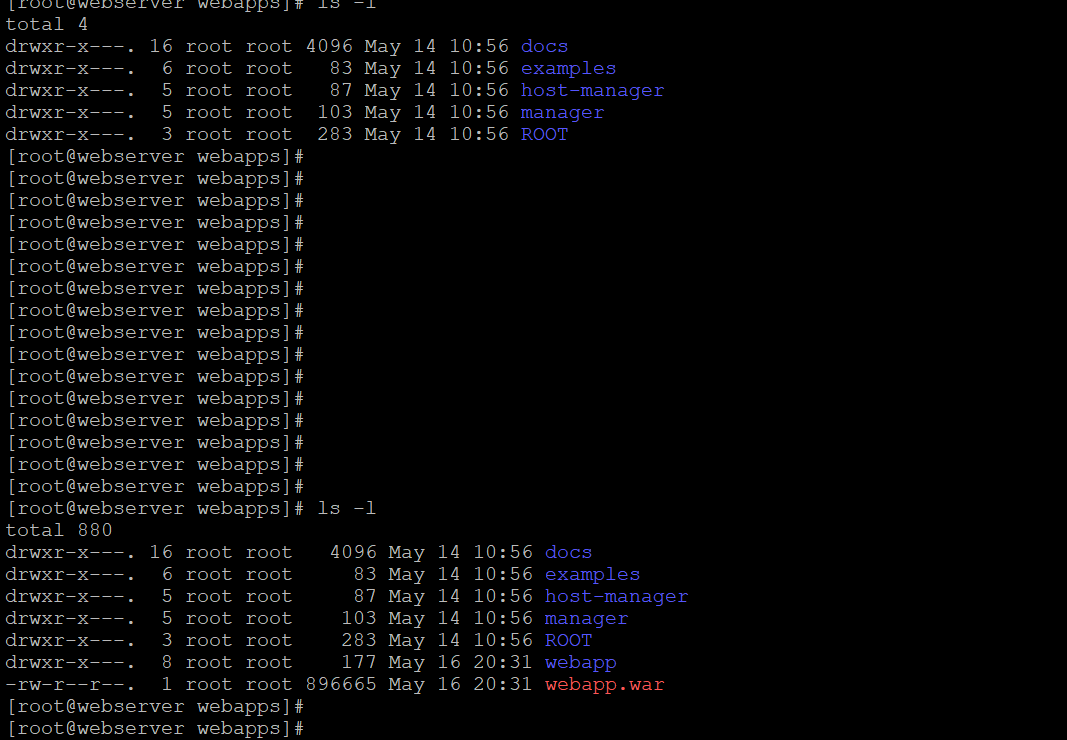


We can observe that build is successfully done!

Webapp.war file is copied in ansible server at below location-



Webapp.war file is deployed in Tomcat server-



Now, Web Application is up and running on tomcat server –

