**Deployment with tomcat using Ansible**

**Pre-Rqusites:**

* Jenkins-server
* Ansible-server
* Tomcat-server

**Ansible-server:**

Install Ansible into this server

Step1:

wget <http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

rpm -ivh epel-release-latest-7.noarch.rpm

yum repolist

yum install ansible -y

ansible --version

step2:

Create user with the name of **ansible**

useradd ansible

Change password for **ansible**

passwd ansible

step3:

Allow **sudo** permissions

visudo

D:\Naresh\snipping\Capture9.JPG

Insert data as:

ansible ALL=(ALL) NOPASSWD: ALL



step 4:

Please enable **yes** for **PasswordAuthentication**

vi /etc/ssh/sshd\_config



**Tomcat-server:**

Install tomcat in this server

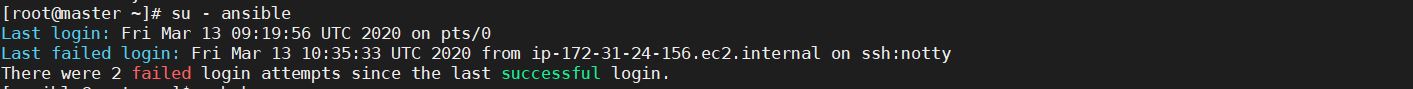
This server treat it as node of Ansible Master. So we need to create **user**, allow **sudo** **permissions** and **PasswordAuthentication** need to be enable to yes

For this we need to repeat step2, step3 and step4 of Ansible-server

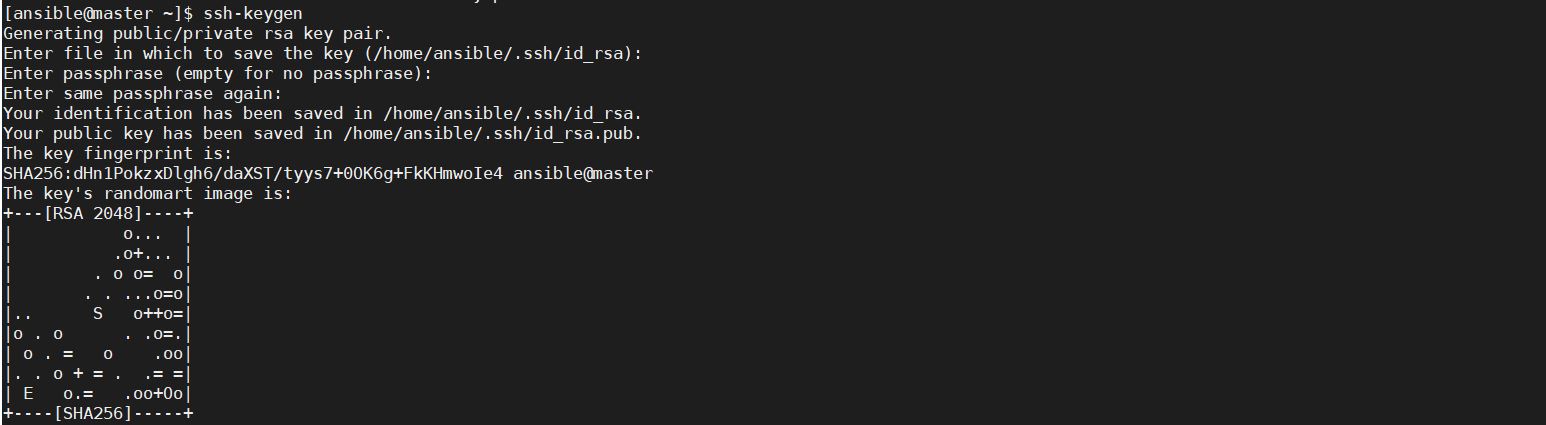
Again come to Ansible server:

ssh-keygen on ansible control server by login on ansible ( ssh is user specific)

su - ansible



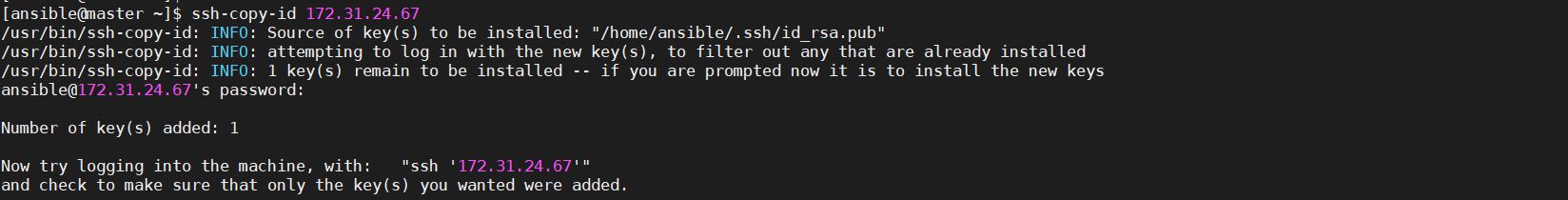
ssh-keygen



Copy private IP address of node or tomcat-server

ssh-copy-id 172.31.24.67

note: provide password for node



Try to login node from master:

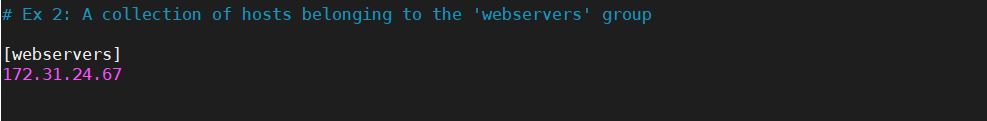
ssh ansible@172.31.24.67



For logout from node give **exit** command.

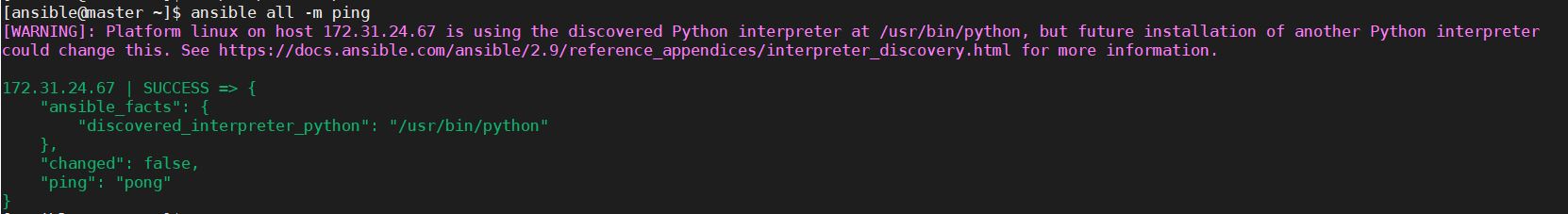
To run playbooks, For this we need keep private IP’s of nodes at Inventory file (/etc/ansible/hosts)

sudo vi /etc/ansible/hosts



Just ping node once whether it pinging or not:

ansible all -m ping



Create a Directory to copy **artifacts**:

mkdir artifacts

Change ownership to **ansible** user for **artifacts** directory.

chown ansible:ansible artifacts

Write a playbook to copy file from Remote src to tomcat webapps folder:

vi copyfile.yml

---

- hosts: all

become: true

tasks:

- name: copy artifacts to webapps folder

copy:

src: /home/ansible/artifacts/mavewebappdemo/target/mavewebappdemo-2.0.0-SNAPSHOT.war

dest: /root/apache-tomcat-8.5.51/webapps

To check syntax for playbook:

ansible-playbook copyfile.yml --syntax-check



**Jenkins-server:**

* Install git
* Install maven
* Install Jenkins

Need to add below plugins:

* Maven Integration
* Publish Over SSH

1. Configure Jenkins with GIT:

Goto Manage Jenkins 🡪 Global Tool Configuration



Give details as above where you installed GIT in your local Jenkins server

1. Configure Jenkins with Maven:

Goto Manage Jenkins 🡪 Global Tool Configuration



Give details as above where you installed MAVEN in your local Jenkins server

1. Configure Ansible server with Jenkins:

Goto Manage Jenkins 🡪 Configure System

And give details as below:

SSH Server Name : anything you can give

Hostname : Private IP of Ansible Server

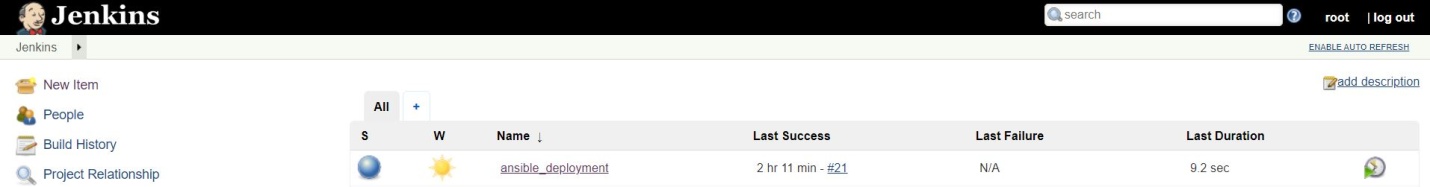
Username : Username (which you created)

Password : Password of the User

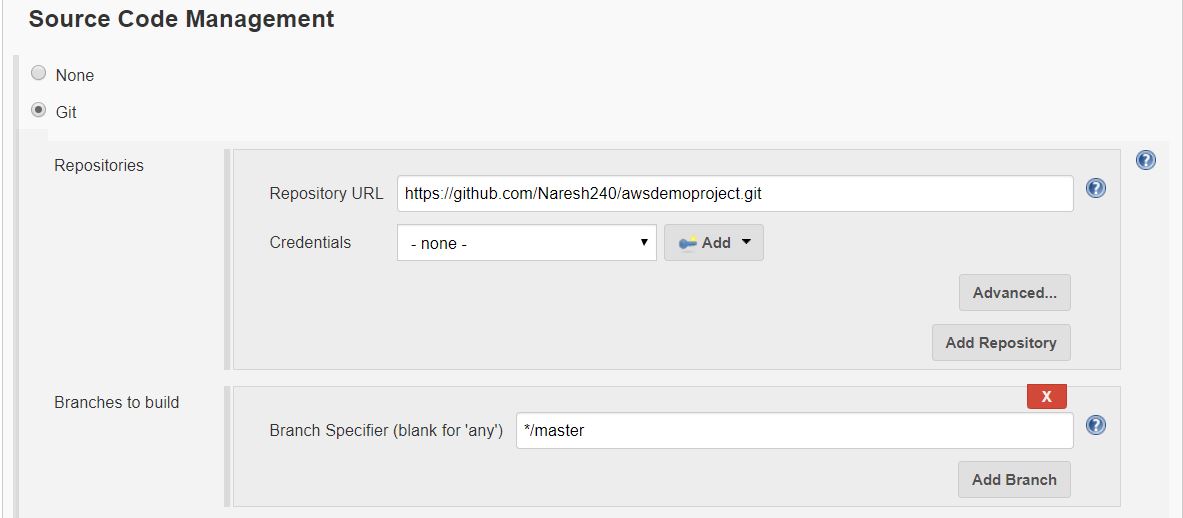


**Deployment:**

Create a Jenkins job with the name of “**ansible\_deployment**”



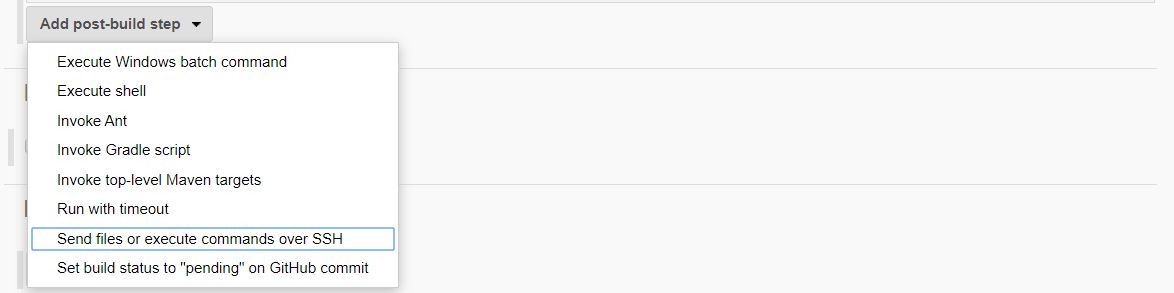
Goto **Source Code Management** Section:



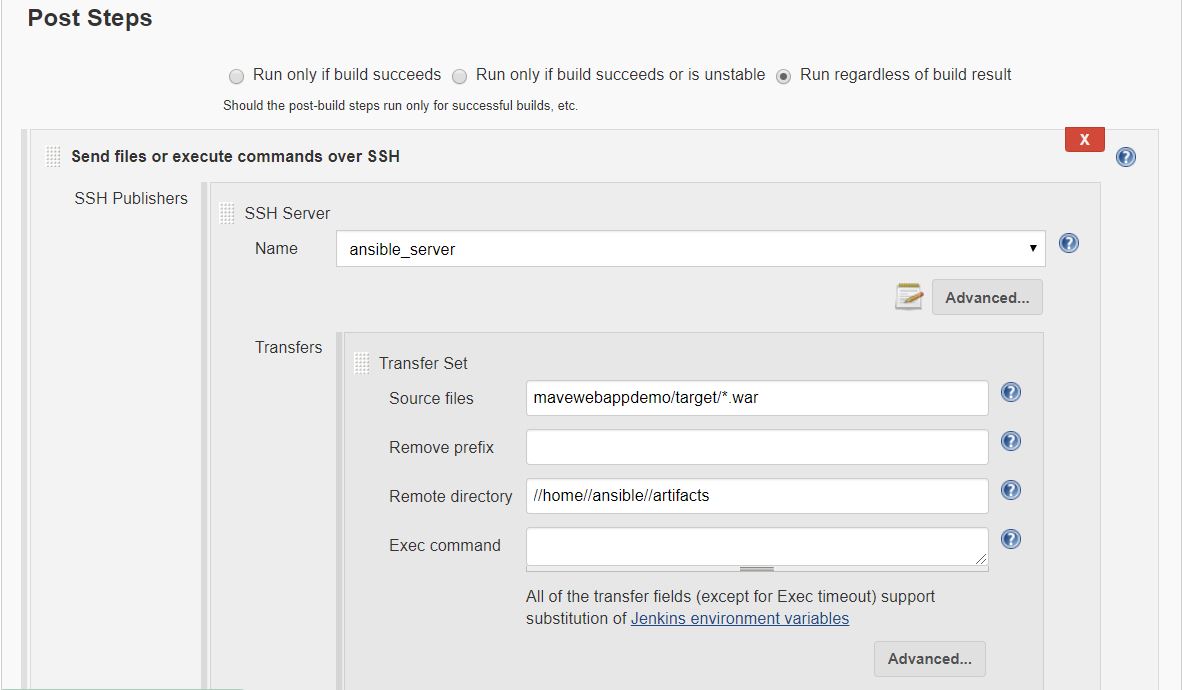
Goto Build Section and provide details as below:



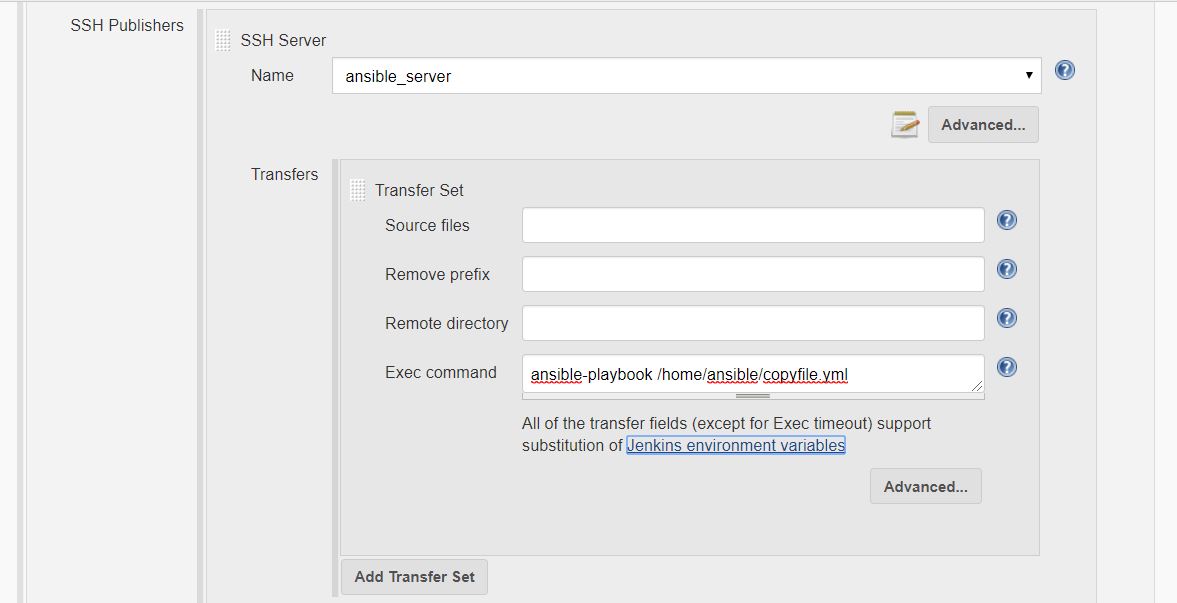
Goto Post Steps click on send files or execute commands over SSH:



Here we get one SSH Server page and give details as shown below image:



Click on Add Server and give details as below:



Click apply and save.

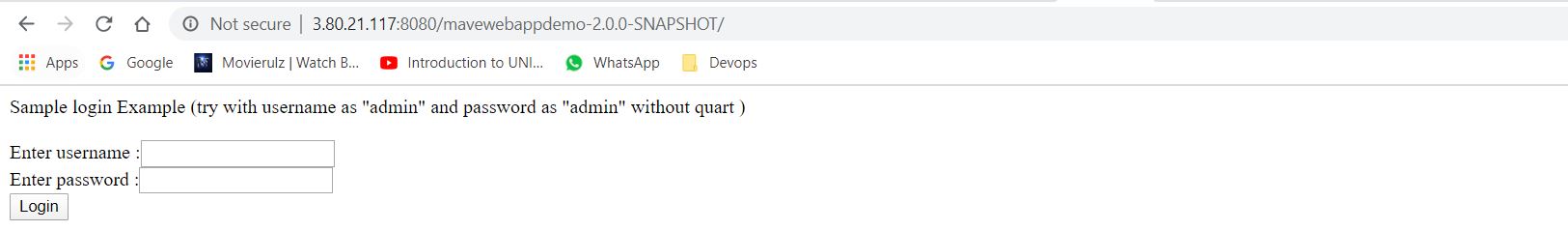


Click on Build



Now we can check output of our application:

<http://3.80.21.117:8080/mavewebappdemo-2.0.0-SNAPSHOT/>

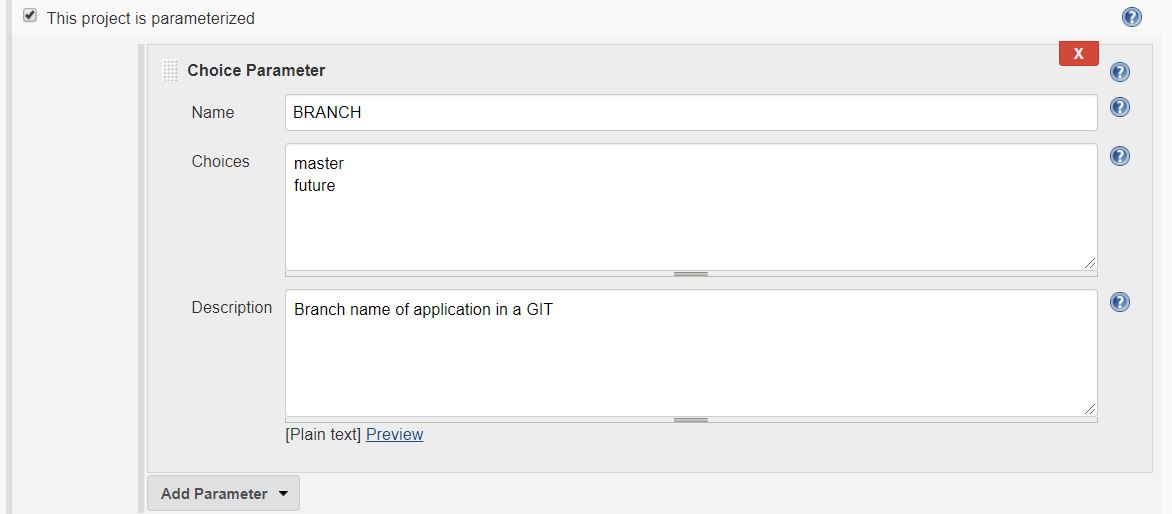


Deployement with Pipeline:

Create a Pipeline job with Deployment\_Pipeline”

Here I taken Parameterized Build for git Branch:

Goto General section and enable checkbox: **This project is parameterized**



Goto pipeline section write below code:



node{

stage("Source Code Management"){

git credentialsId: 'github', url: 'https://github.com/Naresh240/awsdemoproject.git'

}

stage("Build Artifacts"){

dir('mavewebappdemo') {

def MAVEN\_HOME = tool name: 'maven3', type: 'maven'

def MAVEN\_CMD = "${MAVEN\_HOME}/bin/mvn"

sh "${MAVEN\_CMD} clean package"

}

}

stage("Copy file"){

sshPublisher(publishers: [sshPublisherDesc(configName: 'ansible\_server',

transfers: [sshTransfer(cleanRemote: false,

remoteDirectory: '//home//ansible//artifacts',

remoteDirectorySDF: false,

removePrefix: '',

sourceFiles: 'mavewebappdemo/target/\*.war')],

usePromotionTimestamp: false,

useWorkspaceInPromotion: false,

verbose: false)])

}

stage("Deploy to tomcat"){

sshPublisher(publishers: [sshPublisherDesc(configName: 'ansible\_server',

transfers: [sshTransfer(cleanRemote: false,

excludes: '',

execCommand: 'ansible-playbook /home/ansible/copyfile.yml',

execTimeout: 120000,

flatten: false,

makeEmptyDirs: false,

noDefaultExcludes: false,

patternSeparator: '[, ]+',

remoteDirectory: '',

remoteDirectorySDF: false,

removePrefix: '', sourceFiles: '')], usePromotionTimestamp: false, useWorkspaceInPromotion: false, verbose: false)])

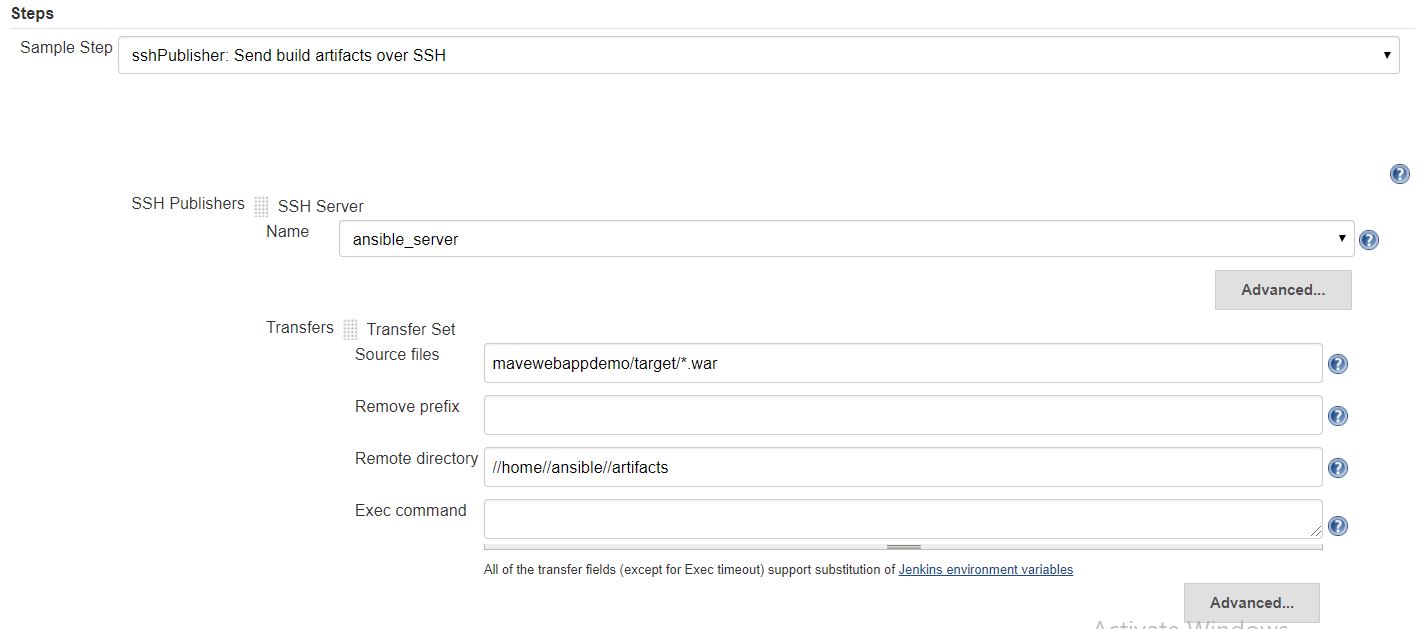
}

}

Reference:

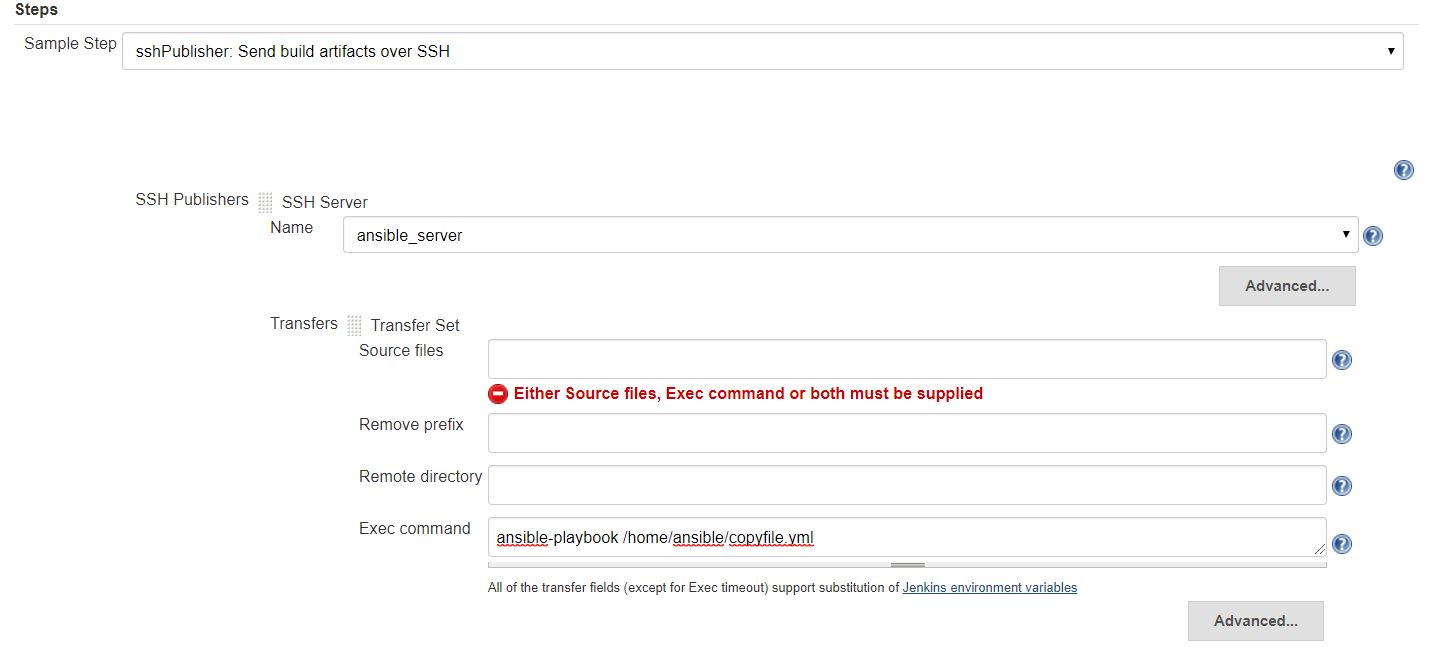
If code is too big just Click on Pipeline Syntax:

For stage("Copy file"):



Click on Generate Pipeline Script

For Stage(("Deploy to tomcat"):



Click on Generate Pipeline Script

Click on Build And Check output in UI:

<http://3.80.21.117:8080/mavewebappdemo-2.0.0-SNAPSHOT/>

