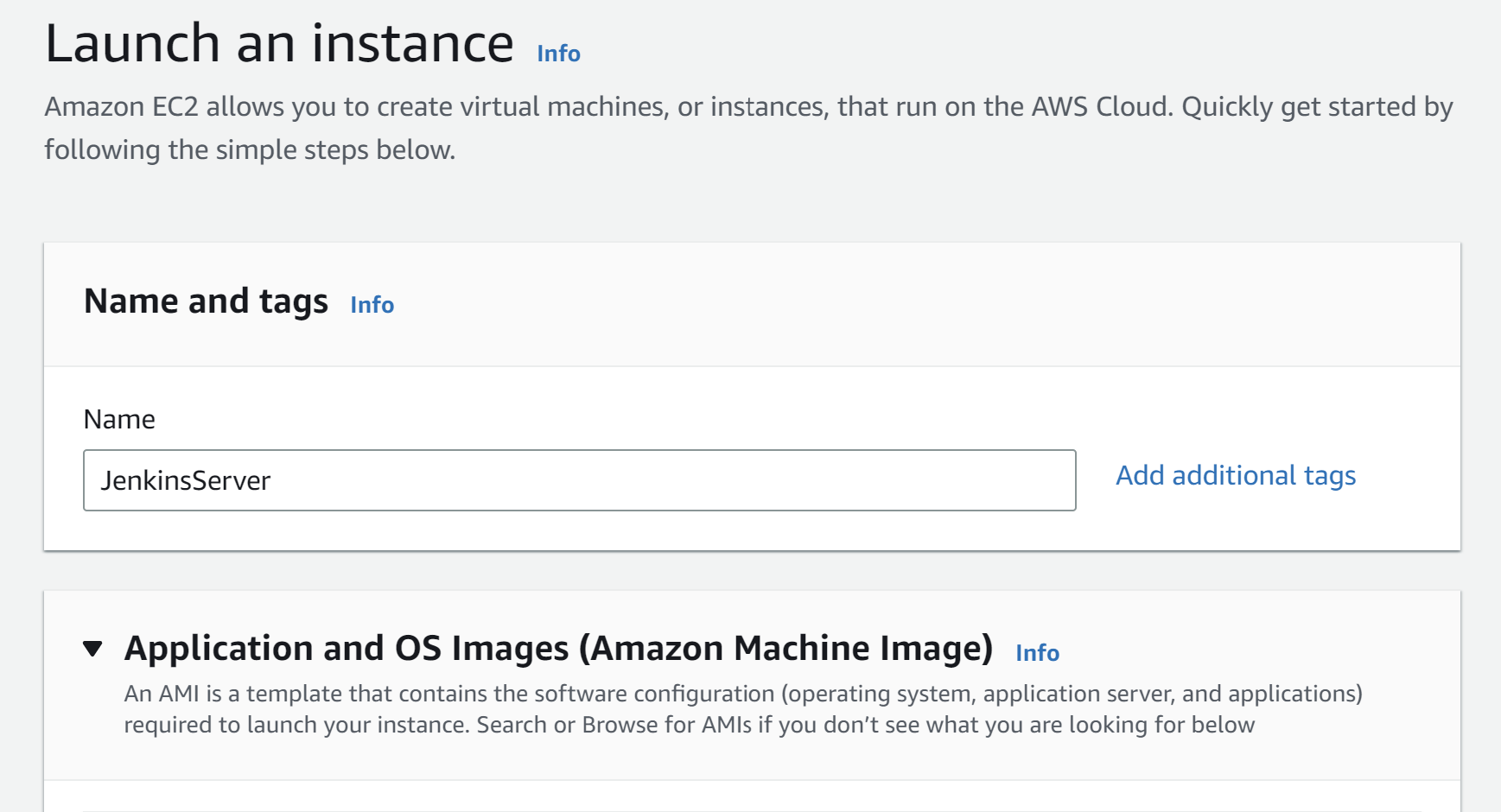
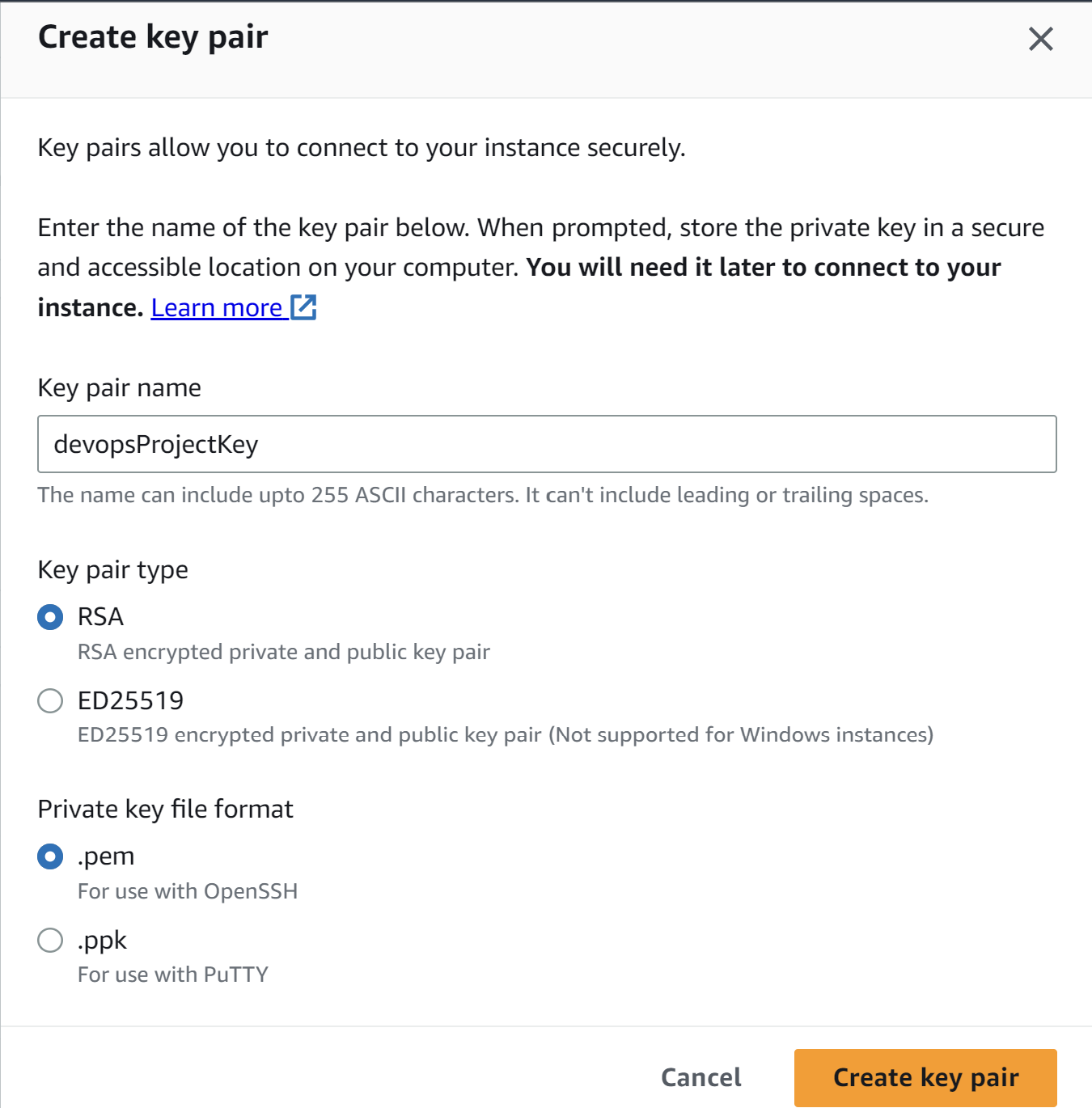
CI/CD pipeline using git, Jenkins and maven:

Setup Jenkins Server:

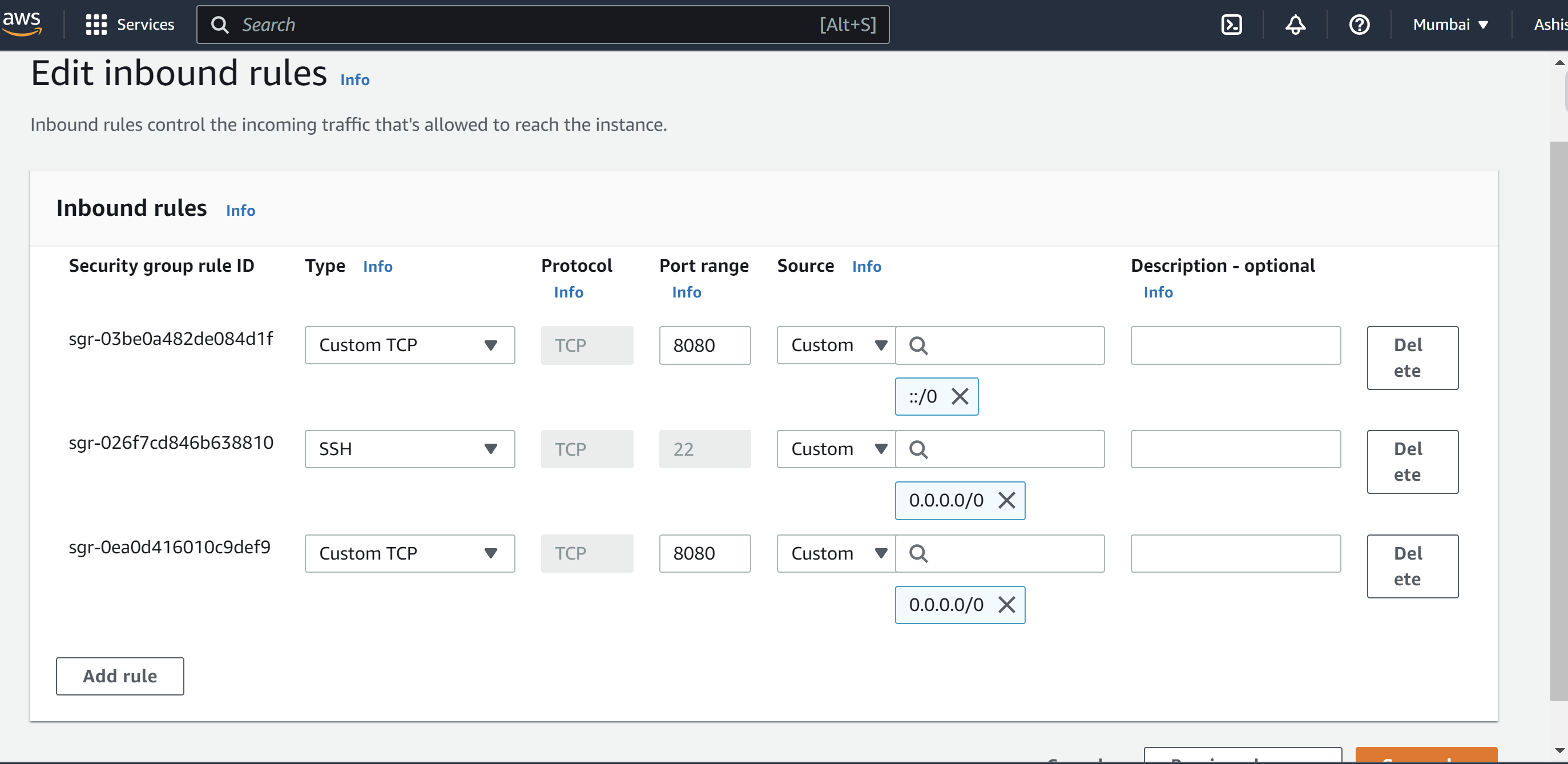
1. Setup a Linux ec2 instance



Creating key in ec2 instance:

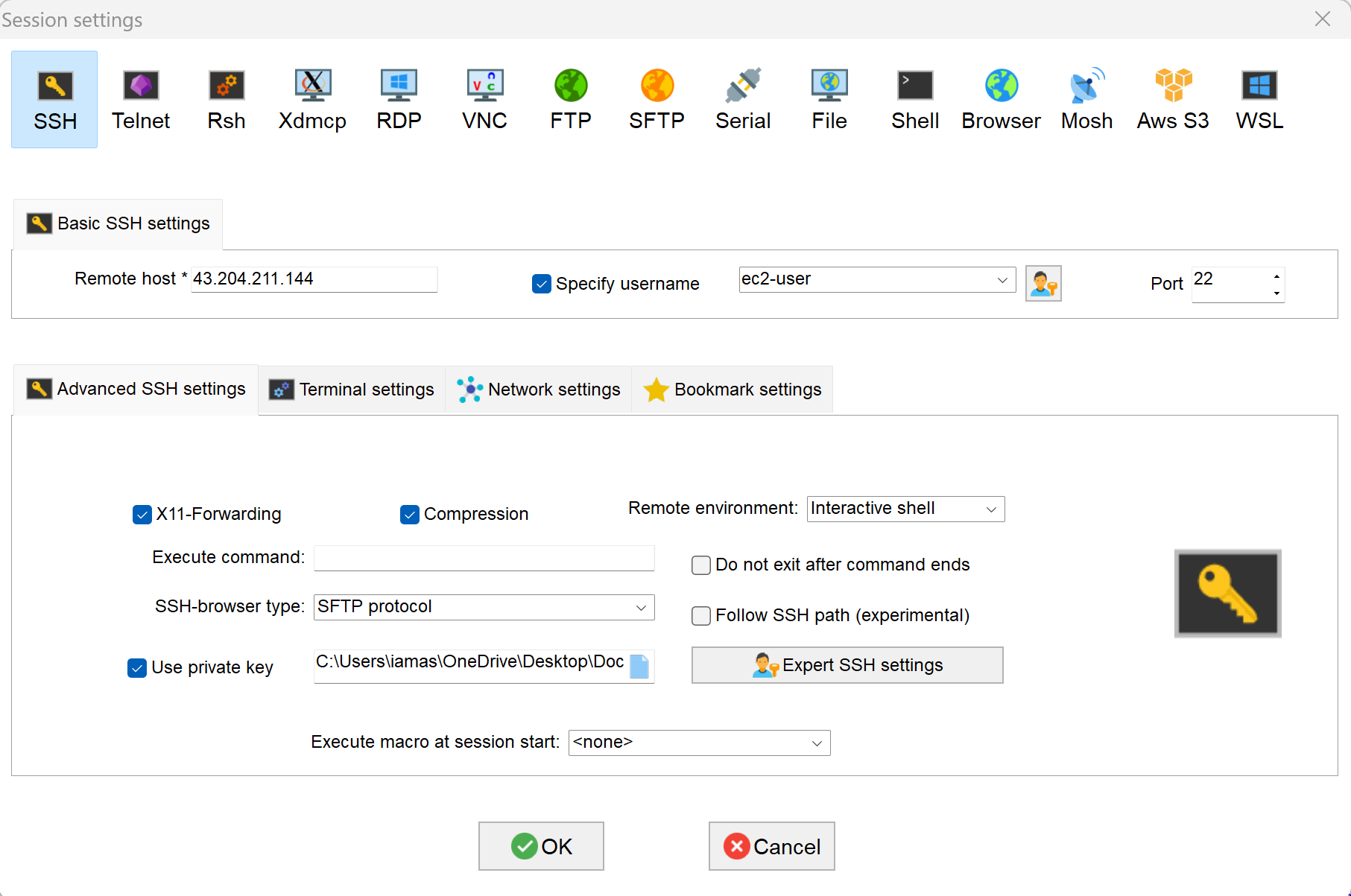
****

Edit inbound rules for port 8080:

****

We can use mobaxterm software to connect to ec2 instance, can use both linux and GUI for the server:

**Go to 🡪** session>ssh> enter IP address and username(default-ec2-user) and add pem file(downloaded during creation of ec2 instance).

****

To become a super user or root user **🡪 sudo su –**

Go to official Jenkins site for the instruction for stable release for [Red Hat/Fedora/Alma/Rocky/CentOS](https://pkg.jenkins.io/redhat-stable/)

1. **Install java and** Jenkins

sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-

stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

amazon-linux-extras install epel **(Will work on amazon linux2 instances only)**

**amazon-linux-extras install java-openjdk11**

**java -version (check java version)**

**yum install Jenkins**

**service Jenkins start 🡪 to start Jenkins**

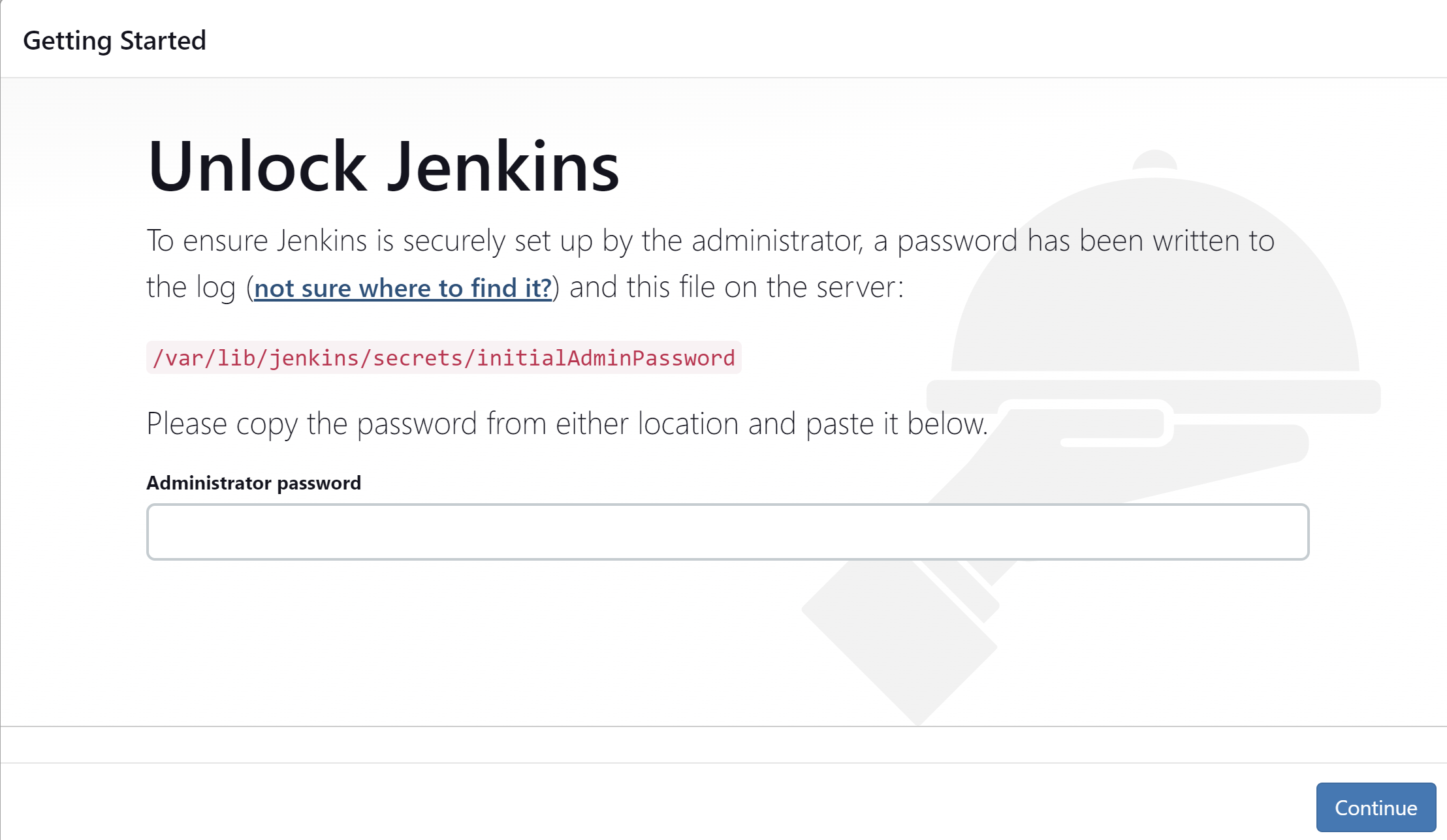
**service Jenkins status 🡪 to check status**

**open the Jenkins server on IP:8080**

**if it asks for administrative password then open the file using cat command mentioned in the ui and use that password.**

**Default username : admin**

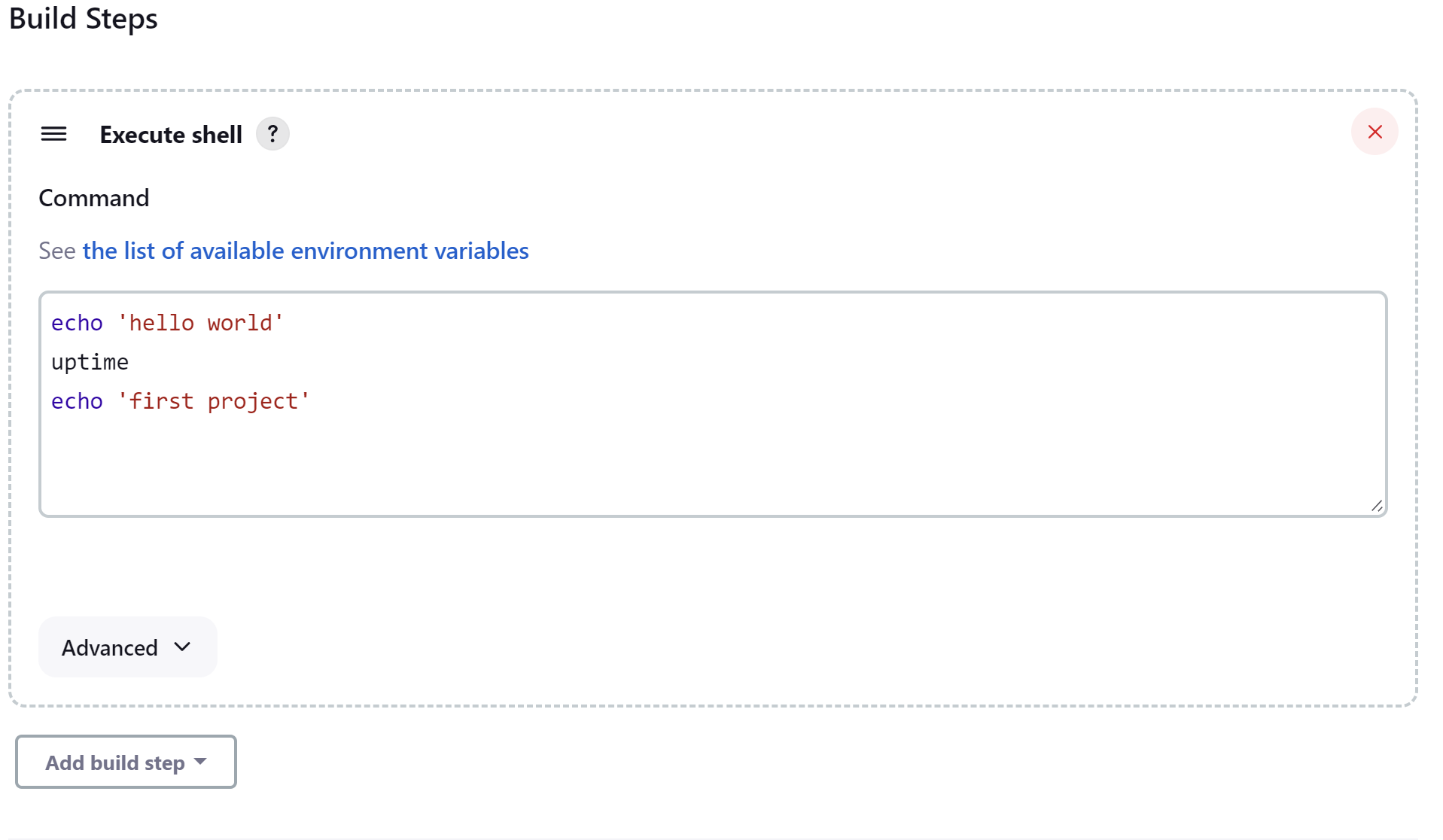
**Password: cat /var/lib/jenkins/secrets/initialAdminPassword**

****

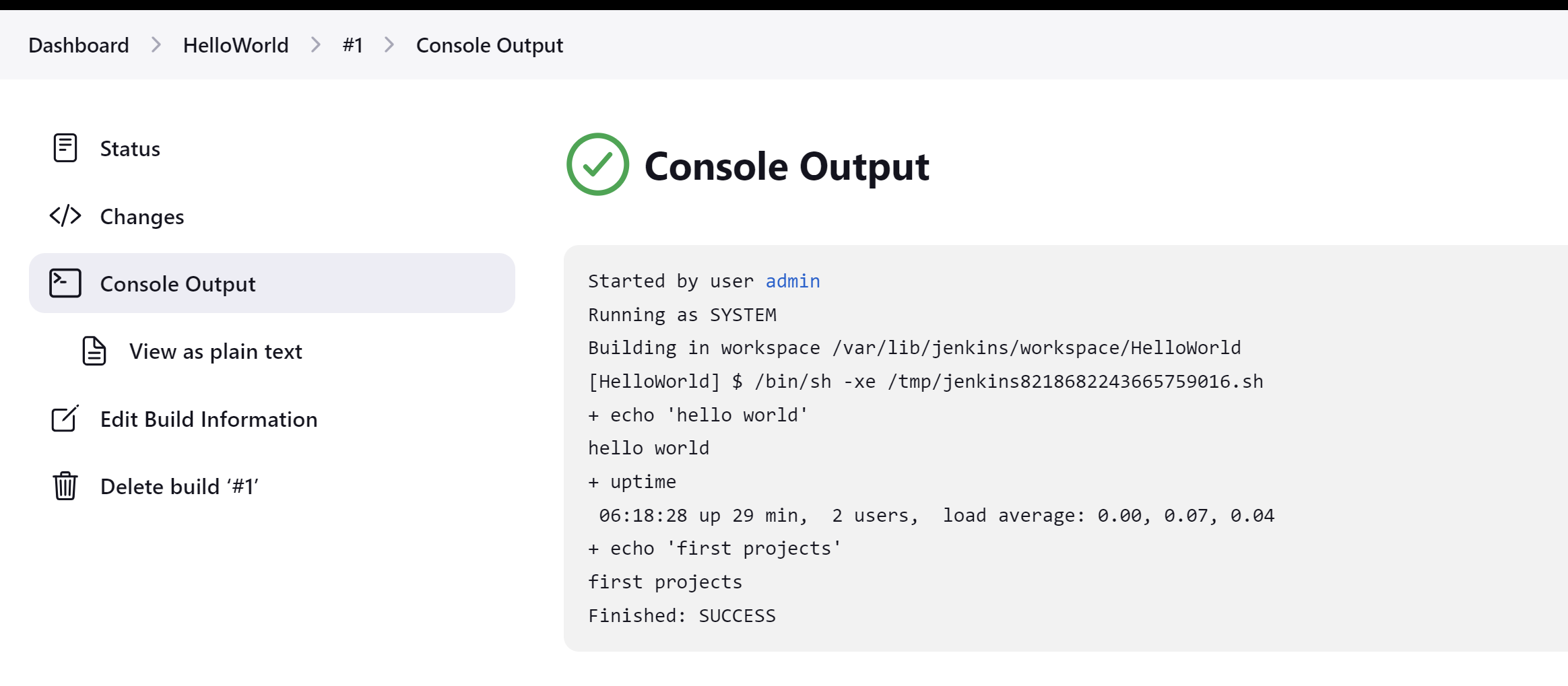
**Running our first Jenkins Job:**

**New item>freestyle project**

**On build steps choose execute shell:**

****

**Output when we do build now.**

****

**CI/CD using git, Jenkins and Maven:**

**Integrating git with jenkins**

**Installing git on linux as :**

yum install git

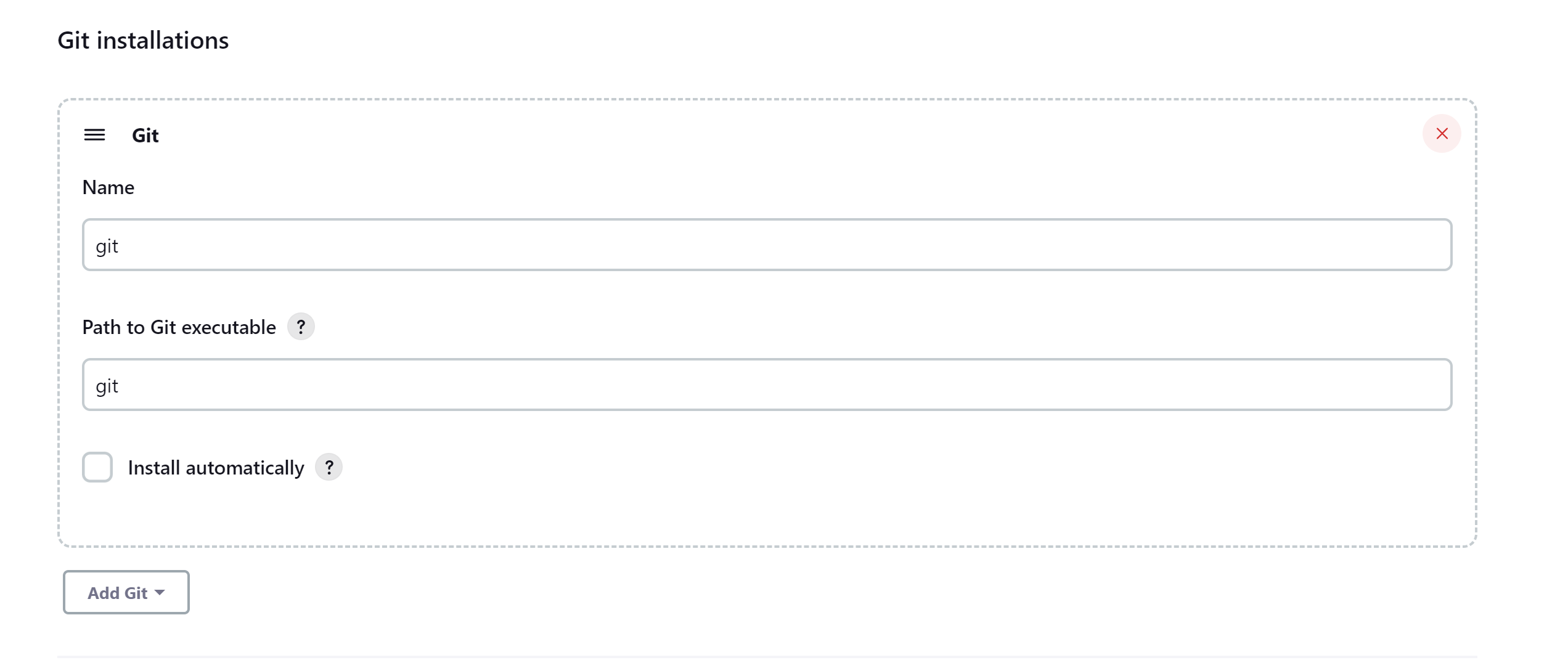
**Adding GitHub plugin:**

**Go to 🡪** manage Jenkins>manage plugins>search for “**github**” and install it.

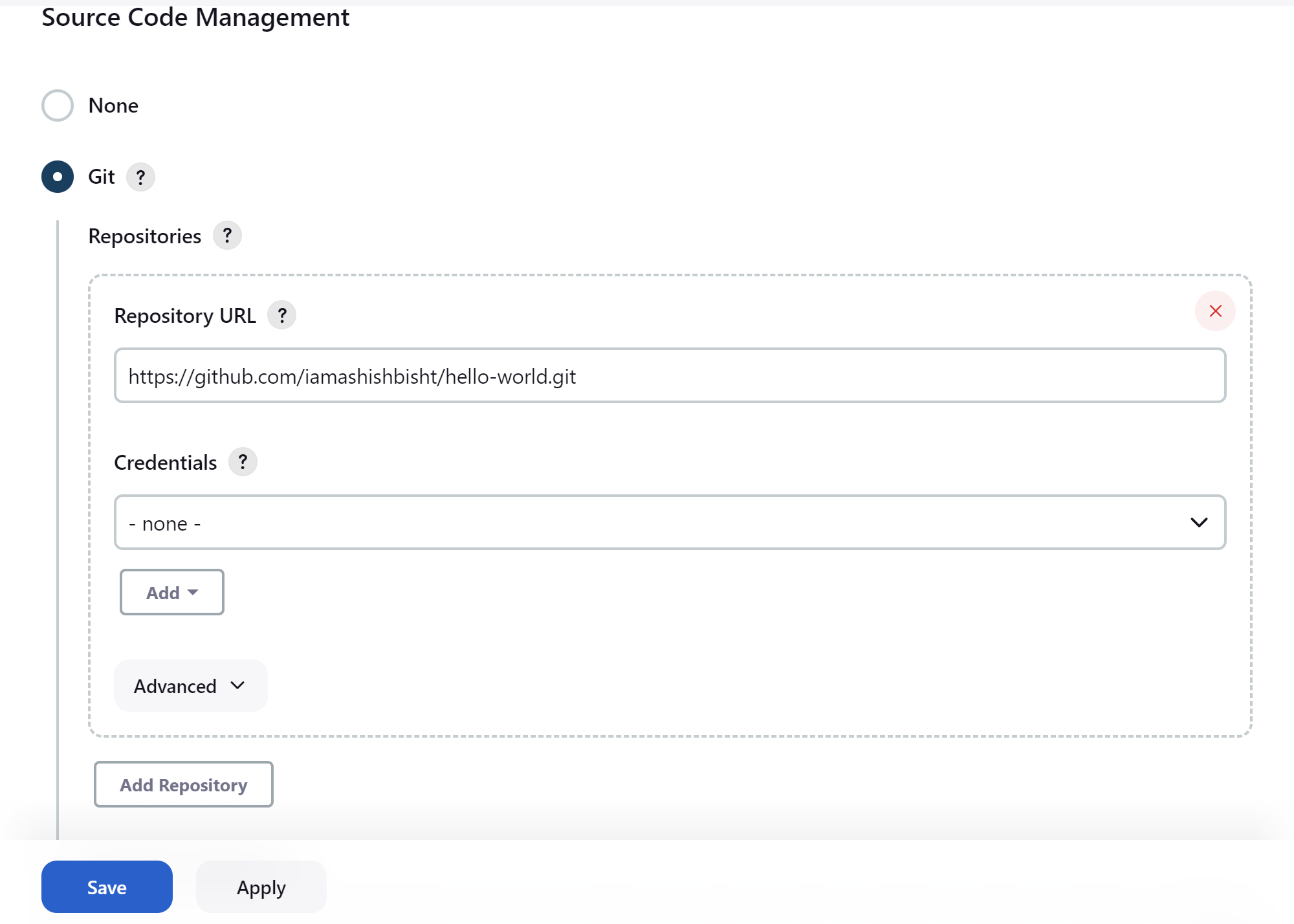
**Configuring git in Jenkins so that it can use commands to pick the code and other operations.**

**Go to 🡪** manage Jenkins>Global tool configurations

**On Git installation section configure git.**

****

After configuring git and github we will have git present in source code management section.



Building the project after doing git configuration:



All the Jenkins jobs stores the job in the folder mentioned in console output as /var/lib/jenkins/workspace/PullCodeFromGithub

**Integrating Maven with Jenkins**

Can use a different server for maven build. But will use same server.

Go to /opt folder (this folder is reserved for add on installation software) and download tar file for maven as:

**wget** [**https://dlcdn.apache.org/maven/maven-3/3.9.1/binaries/apache-maven-3.9.1-bin.tar.gz**](https://dlcdn.apache.org/maven/maven-3/3.9.1/binaries/apache-maven-3.9.1-bin.tar.gz)

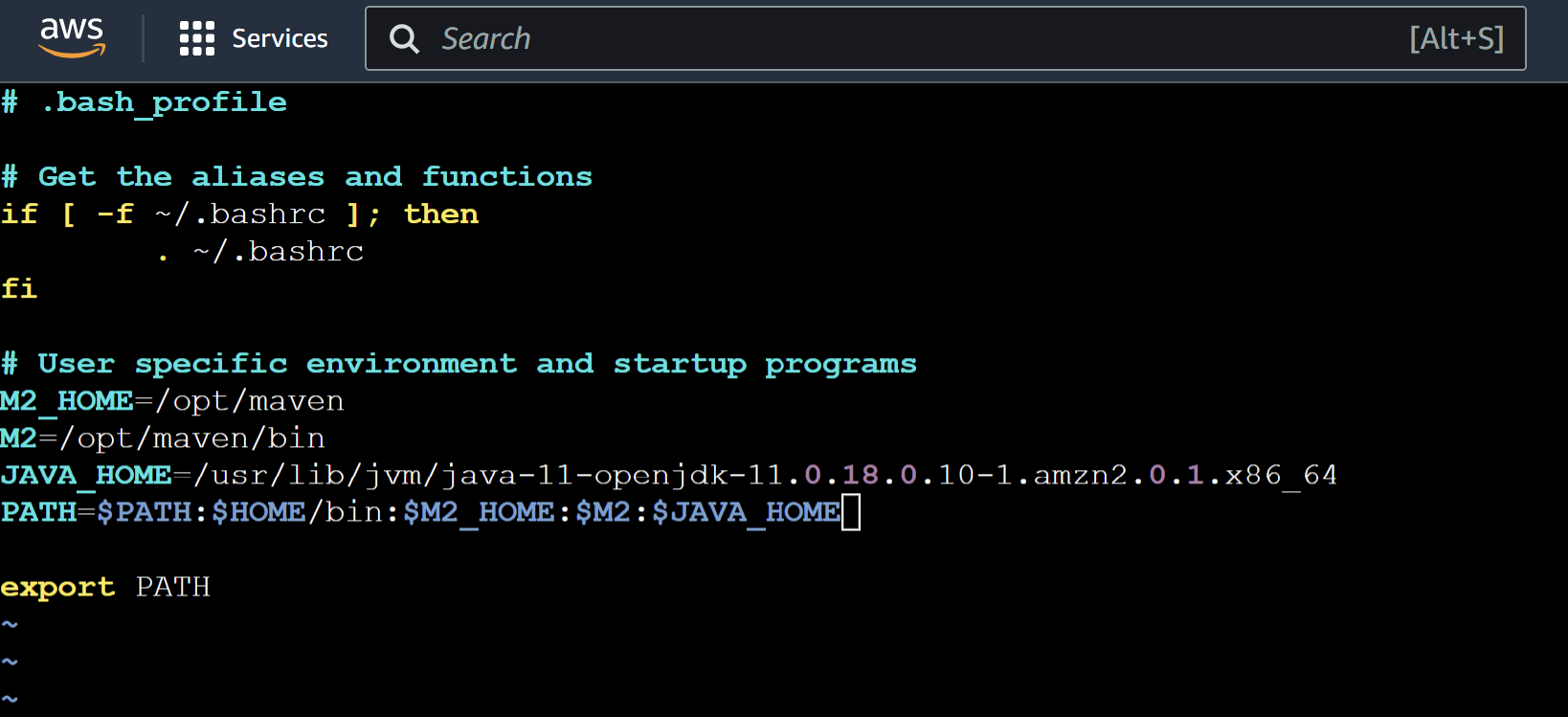
**extract the downloaded package as:** tar -xvzf apache-maven-3.9.1-bin.tar.gz

moving this file in maven folder: mv apache-maven-3.9.1 maven

to get java 11 path command is : find / -name java-11\*

setting environment variable in .bash\_profile in /root directory as :

1. vi .bash\_profile 🡪 edit the file with:

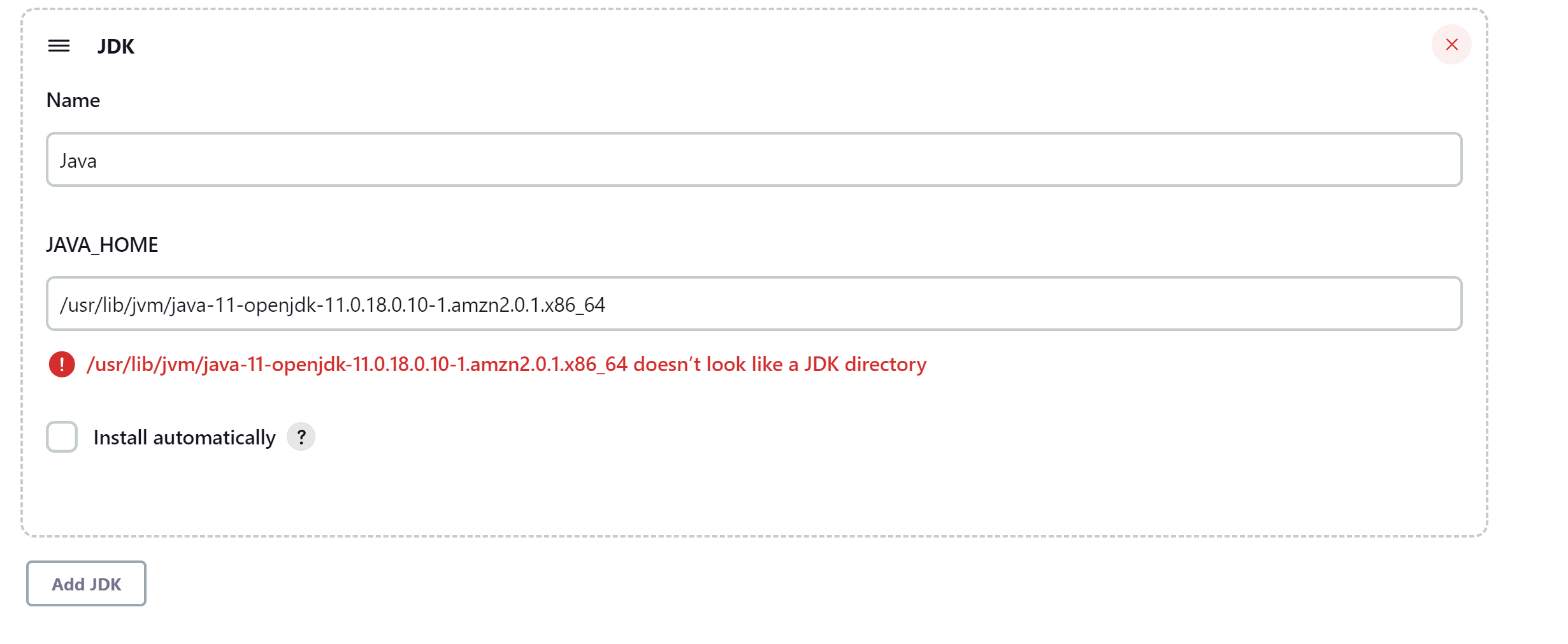


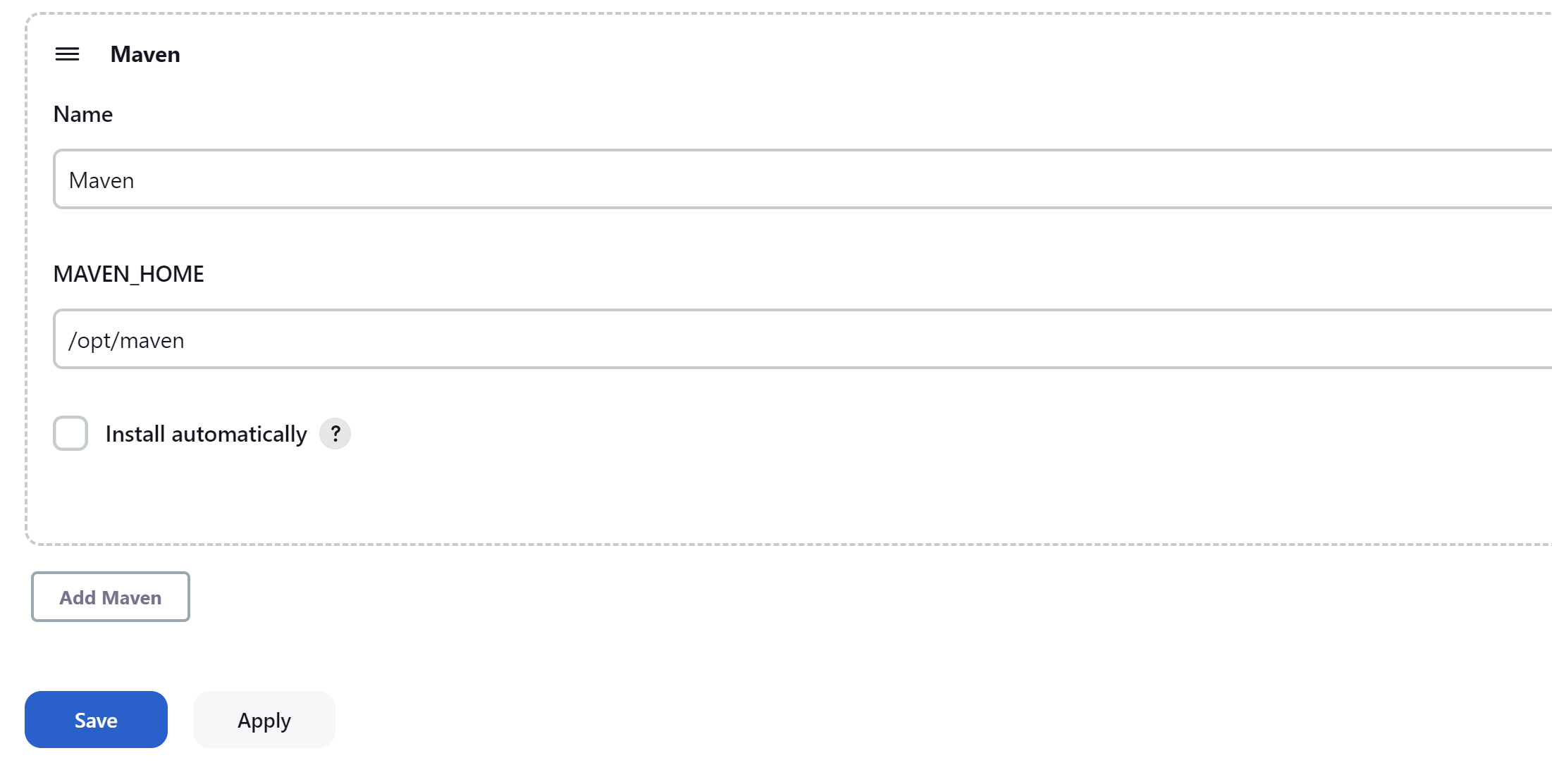
Changes will be reflected for the path if we restart the server or refresh the file to read as : **source .bash\_profile**

**Maven Integration in Jenkins.**

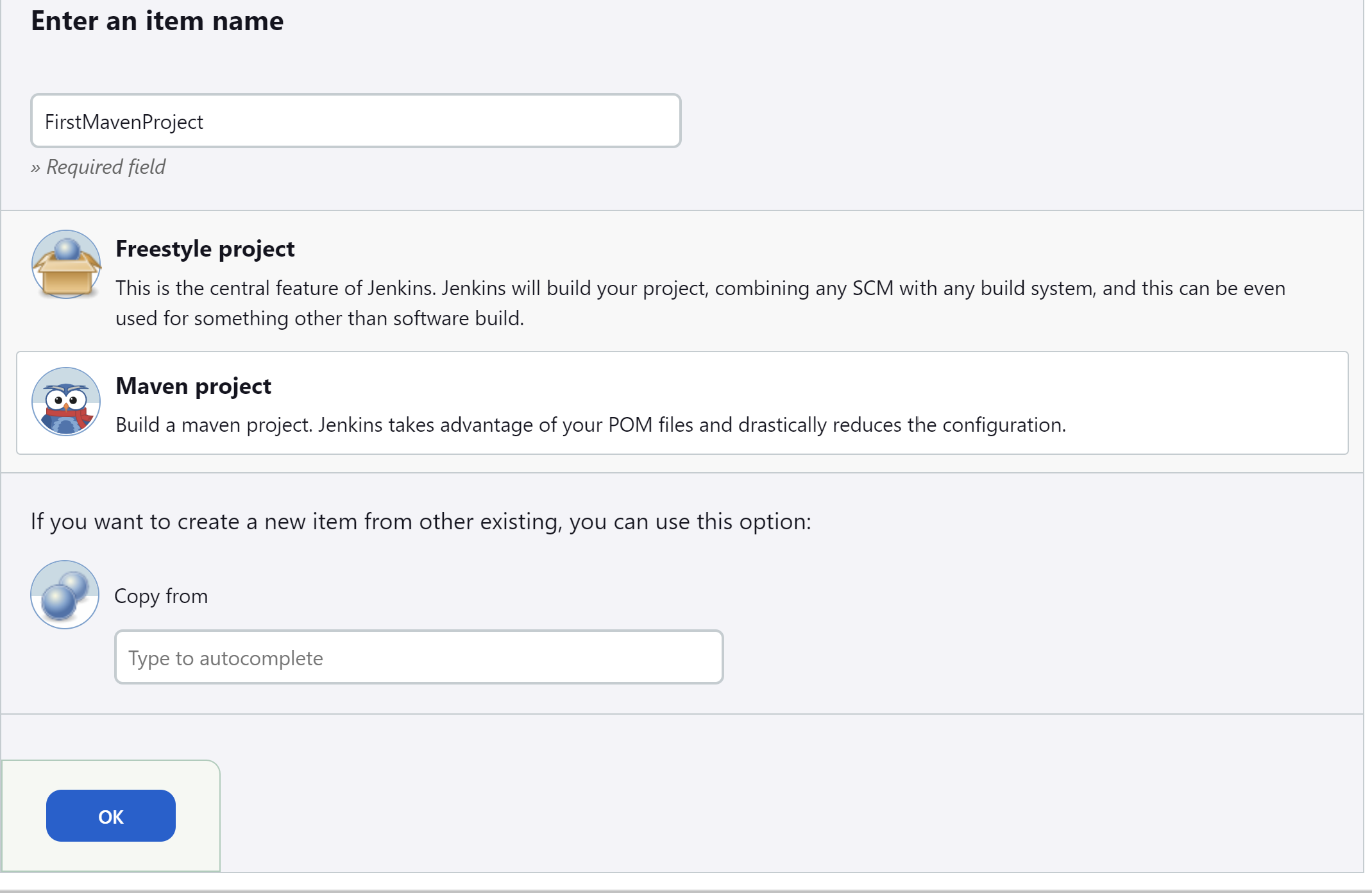
**Add maven integration plugin from manage Jenkins>manage plugins**

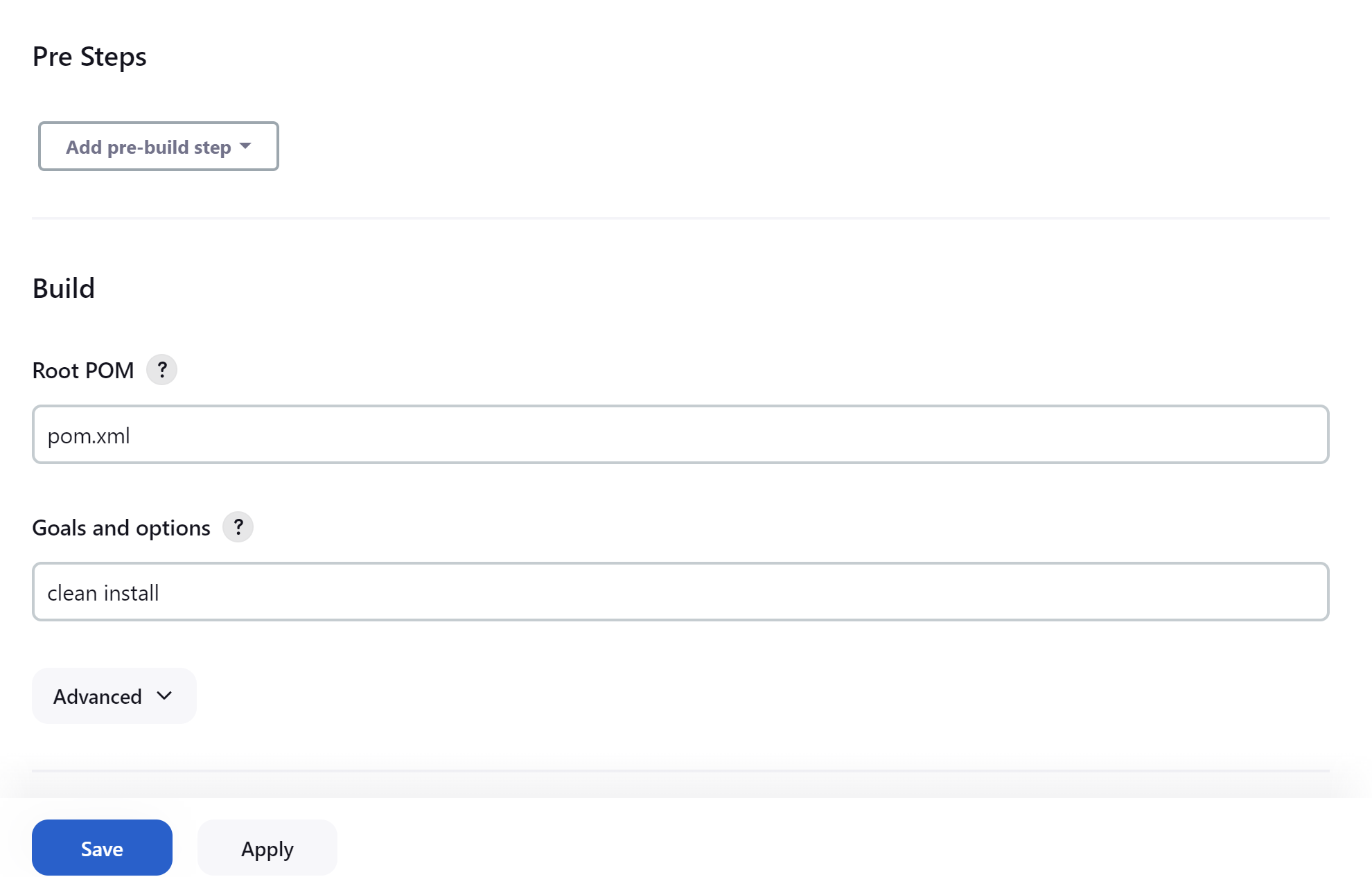
**Now configuring it in Global tool configuration as:**



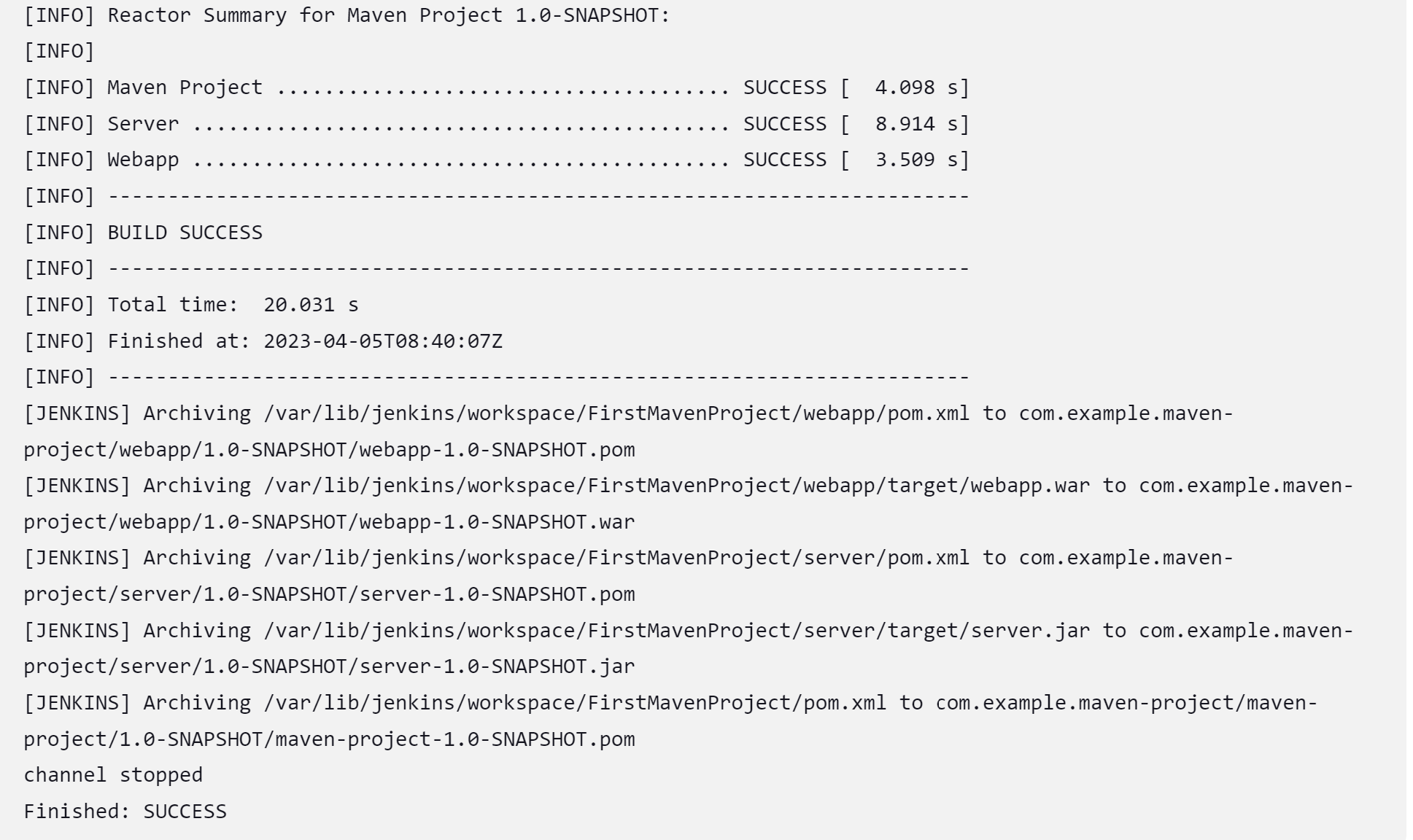


Now creating a maven project in Jenkins as:





Console logs for project build



Integration Tomcat Server in CI/CD pipeline:

Launch a instance for tomcat server in AWS.

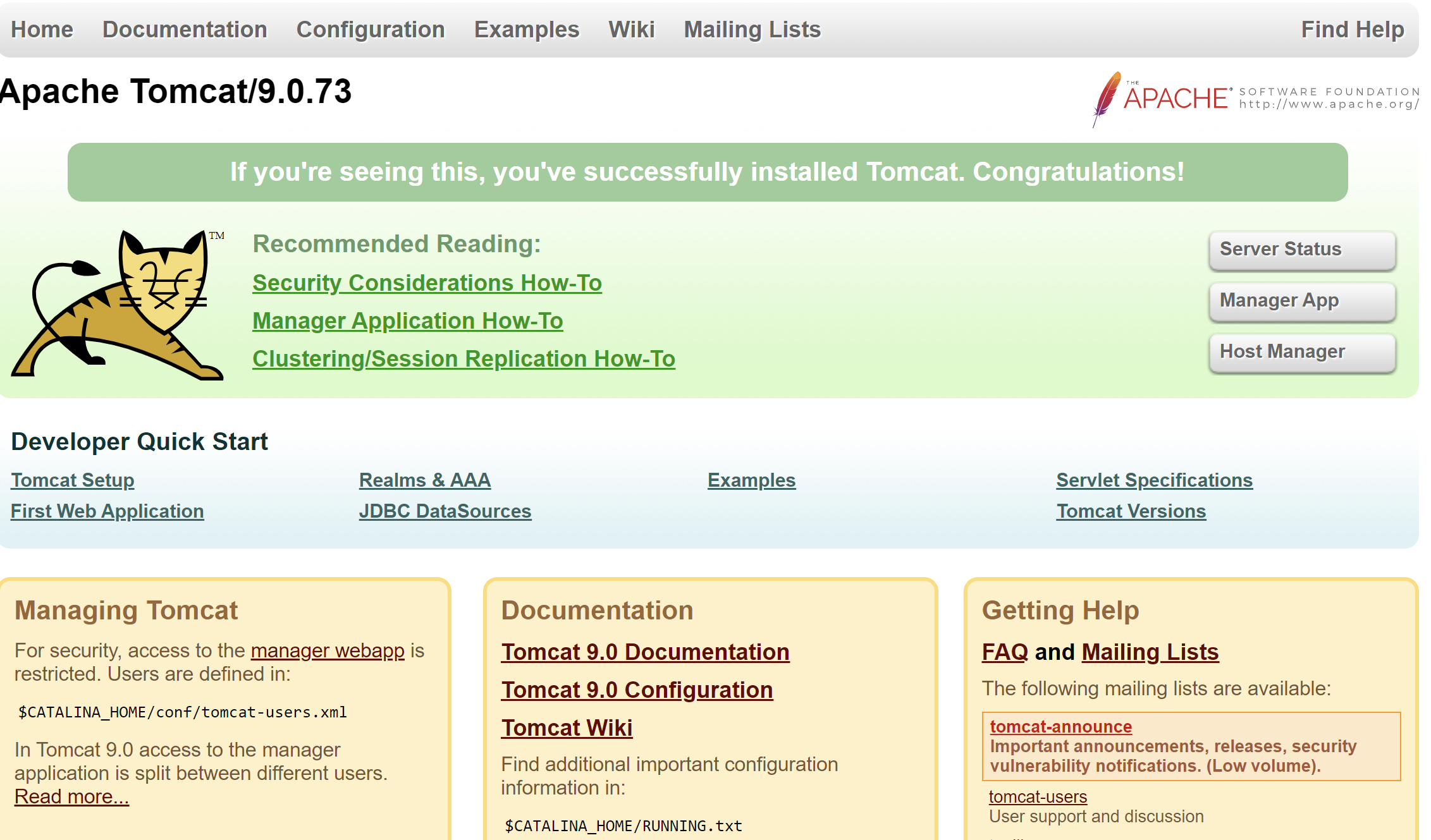
Download tomcat for linux in **/opt** folder using: wget <https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.73/bin/apache-tomcat-9.0.73.tar.gz>

Unzip the file and start the tomcat using **./startup.sh**

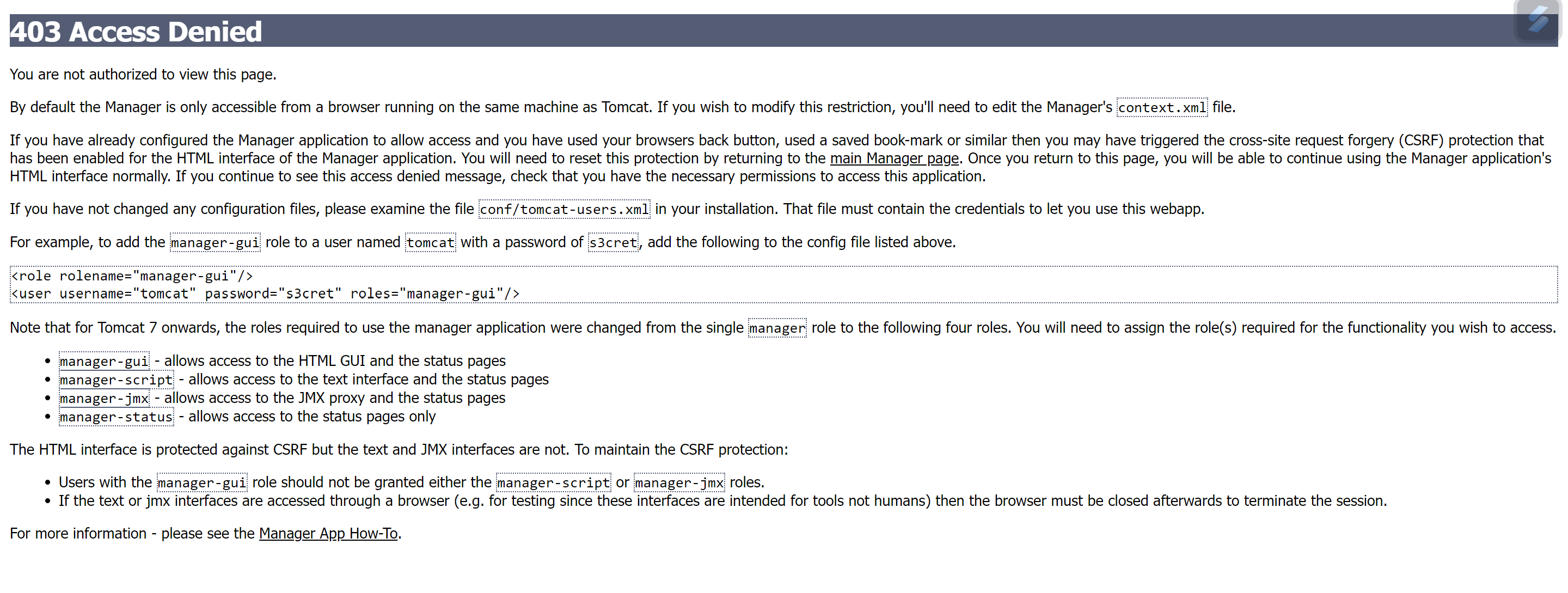
To run Tomcat we need jdk to be installed. So on the tomcat server install java.

**amazon-linux-extras install java-openjdk11**

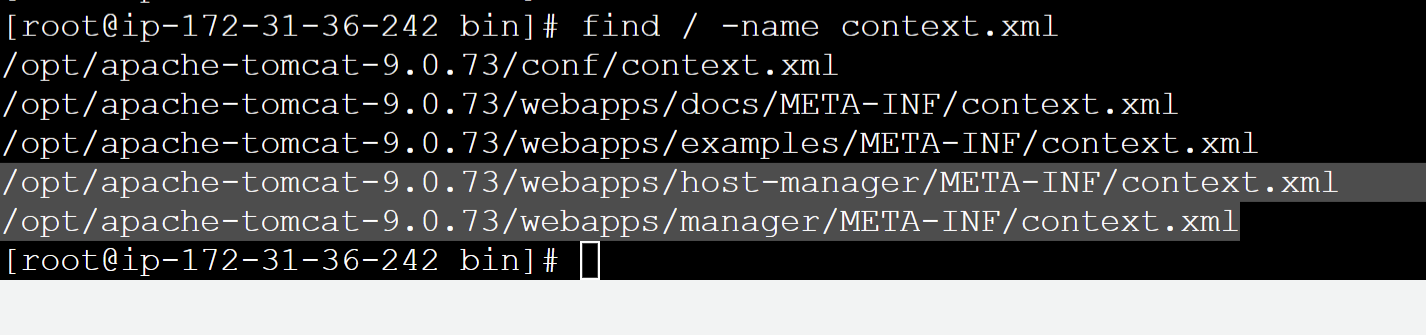
Hit tomcat at port 8080



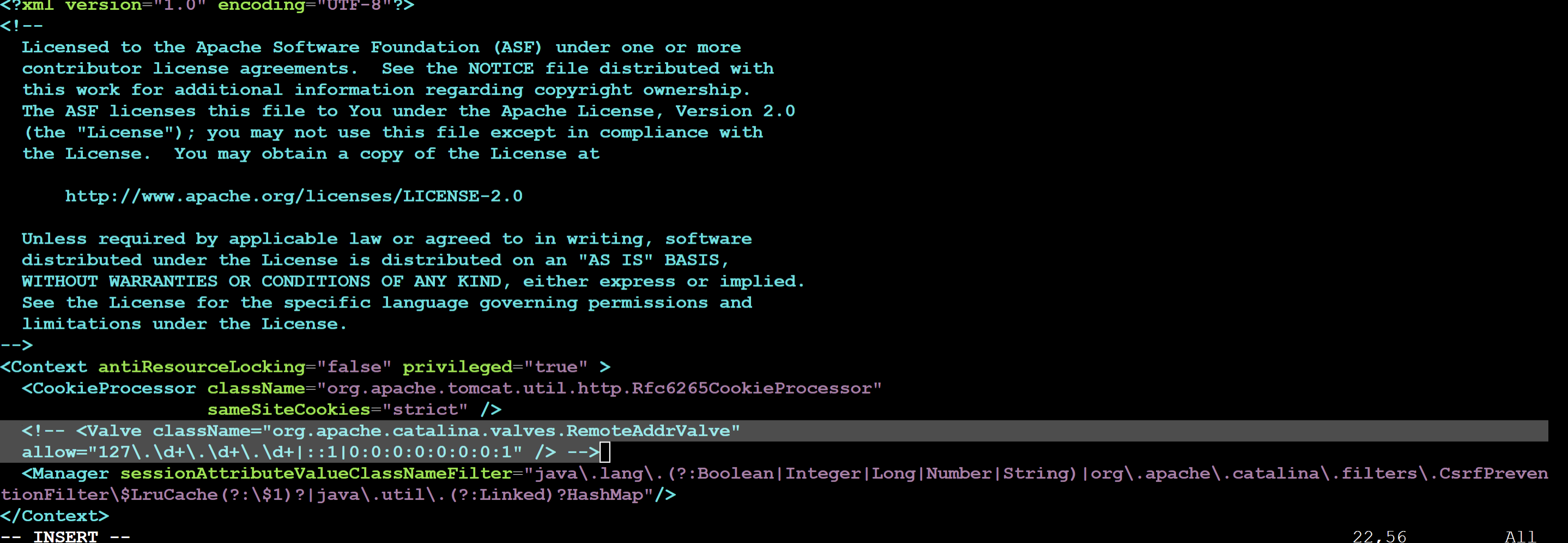
Go to manager app we get the unauthorized page because we can access tomcat web UI only from the machine’s browser where it is installed.



To make the web ui accessible from other browser we need to change context.xml, we need to make changes in these 2 context.xml:

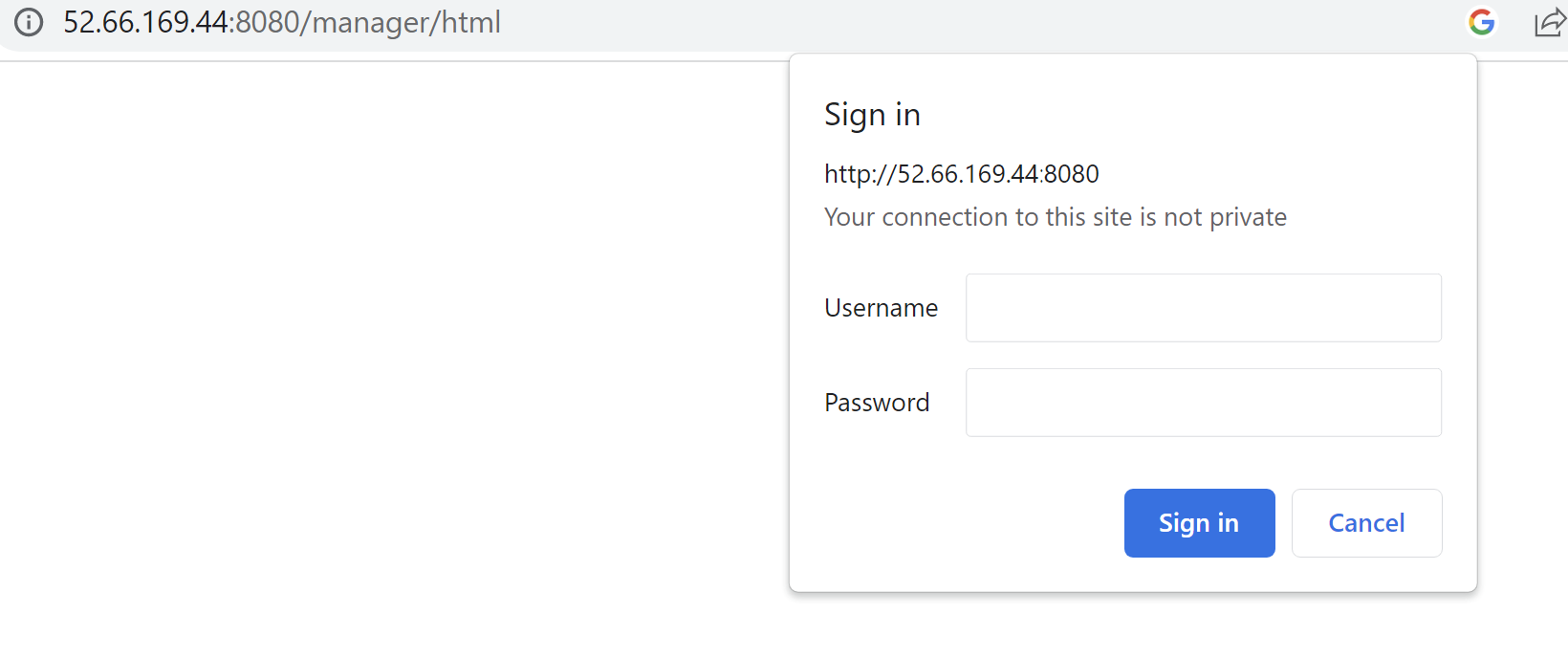


The changes should be commenting the line which is defined to coonect from localhost or from the same server.

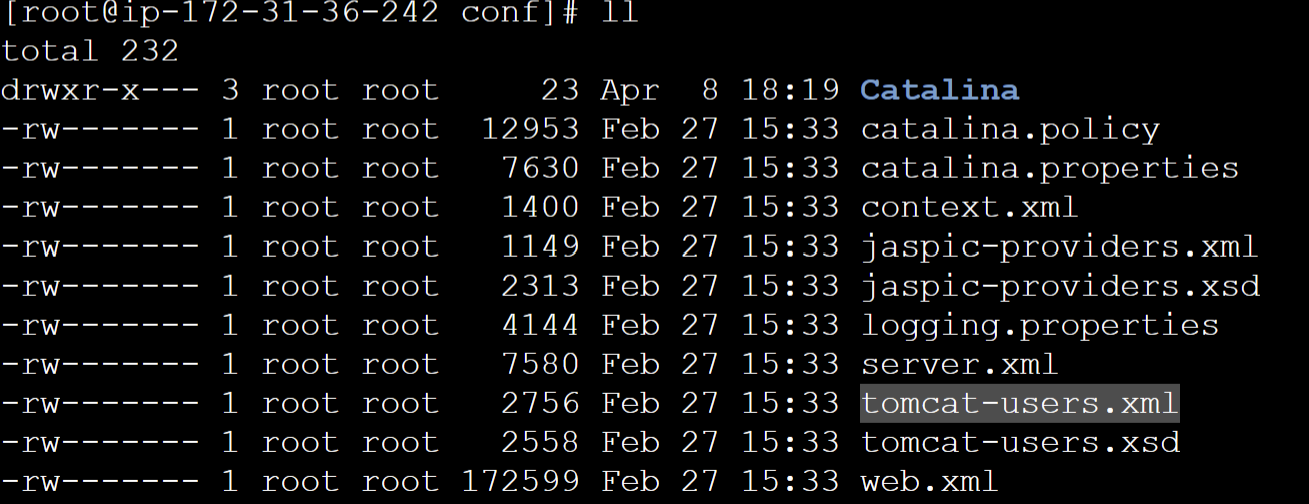


Restart tomcat server after making changes.

After going to manager apps it will ask for username.password.



Edit tomcat-users.xml for adding username/password



Edit the file as:

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

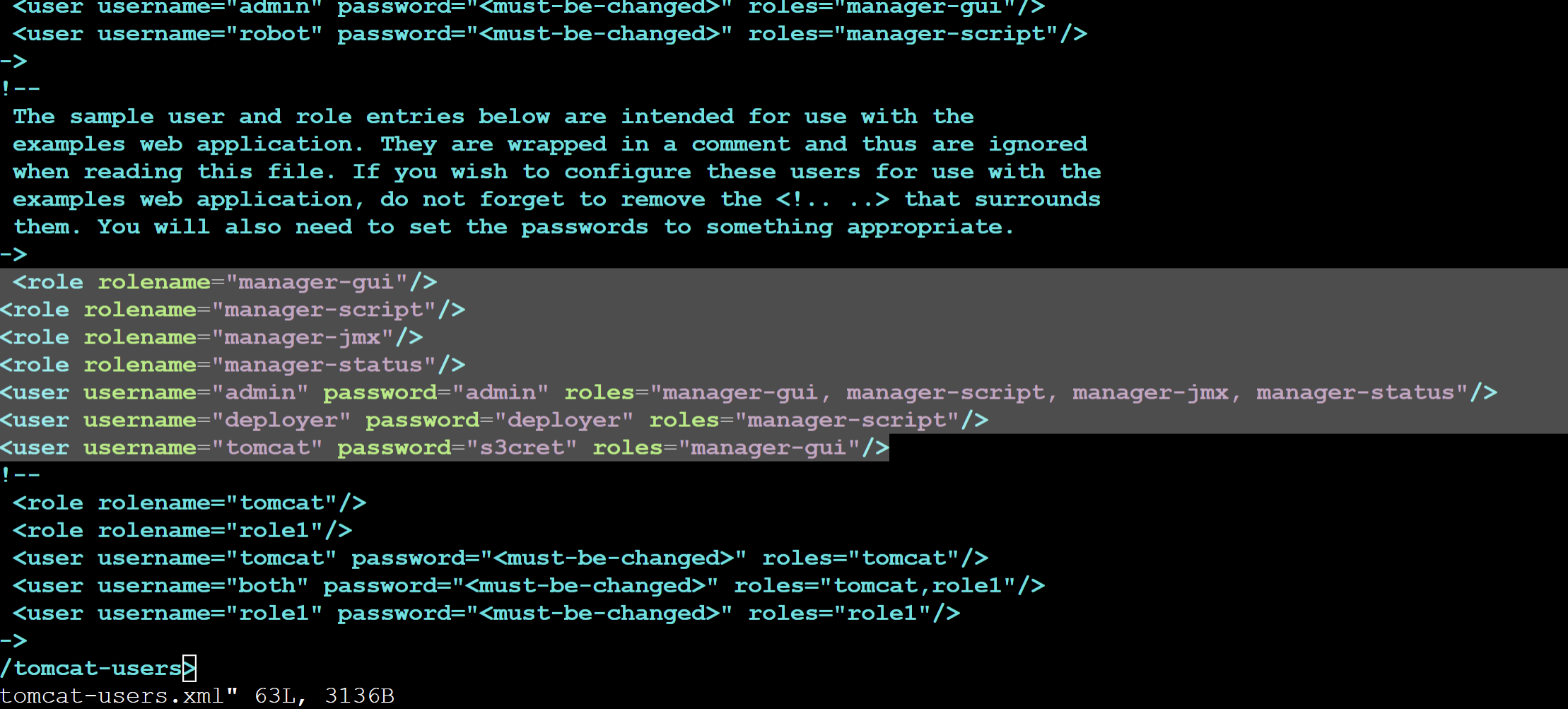
<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>

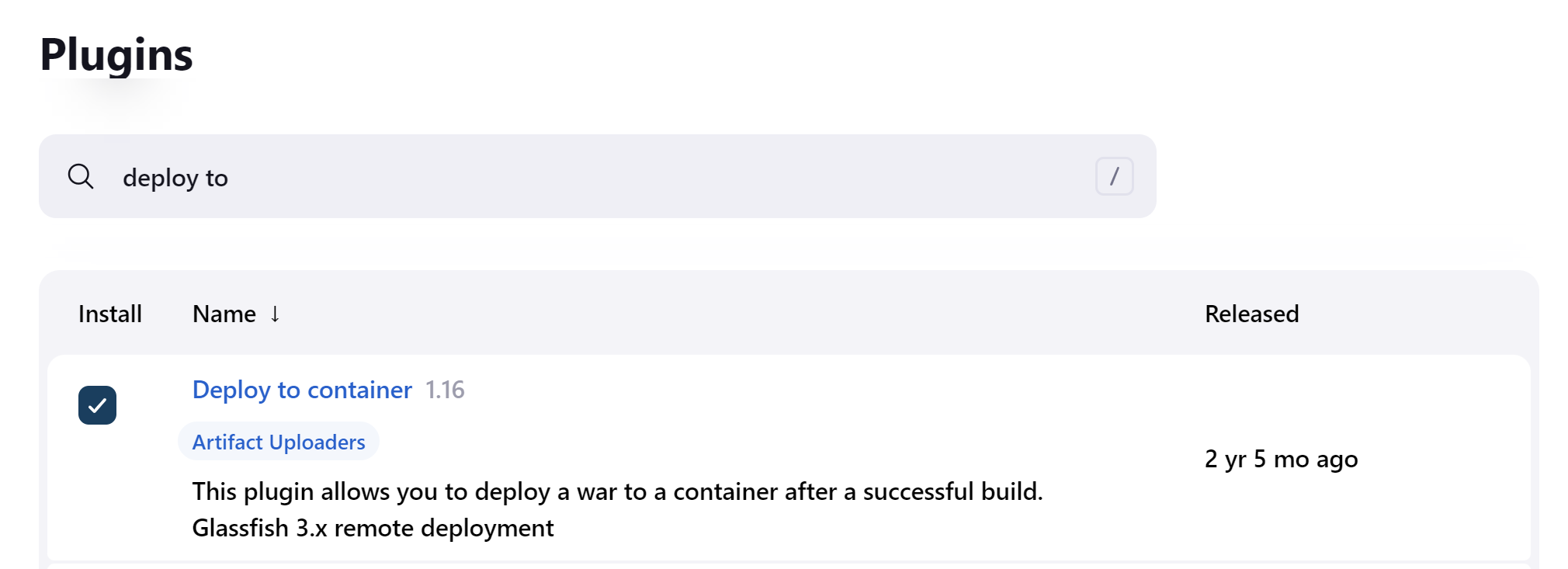
<user username="deployer" password="deployer" roles="manager-script"/>

<user username="tomcat" password="s3cret" roles="manager-gui"/>

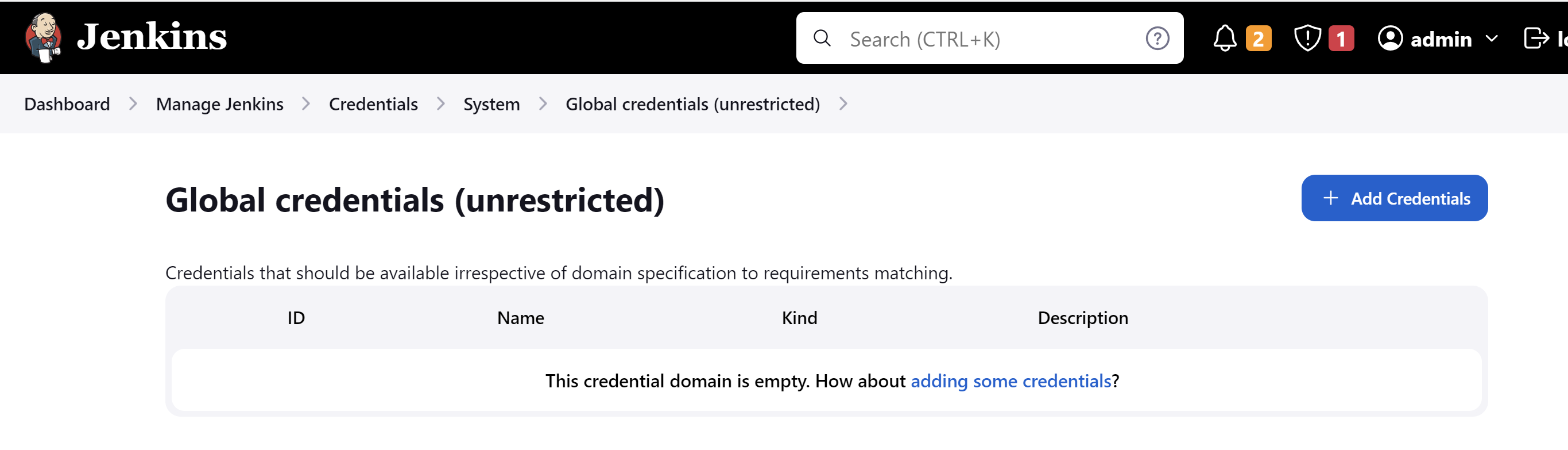


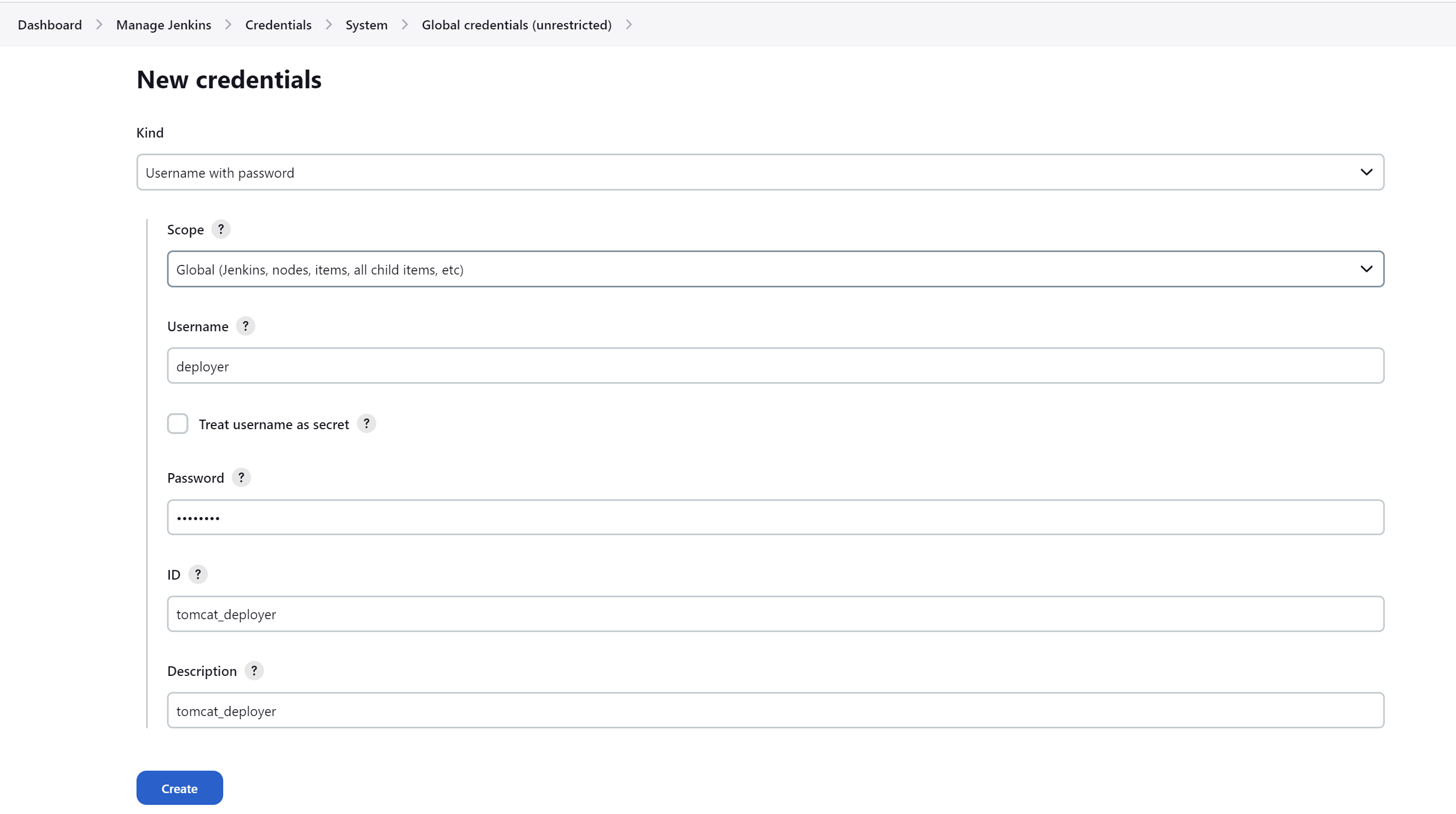
**Integrating Tomcat with Jenkins**

**Deploy to container** plugin to be installed to integrate tomcat server with Jenkins.

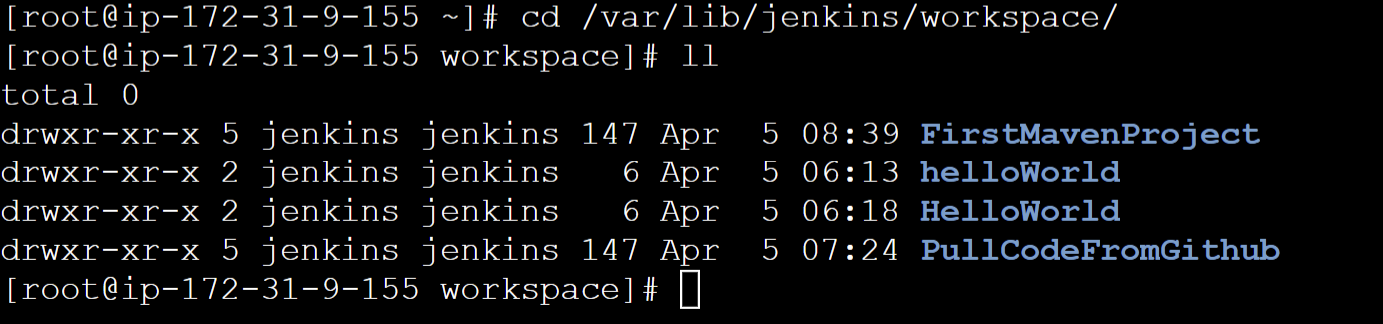


To manage Tomcat credentials we have to configure it in Jenkins as:

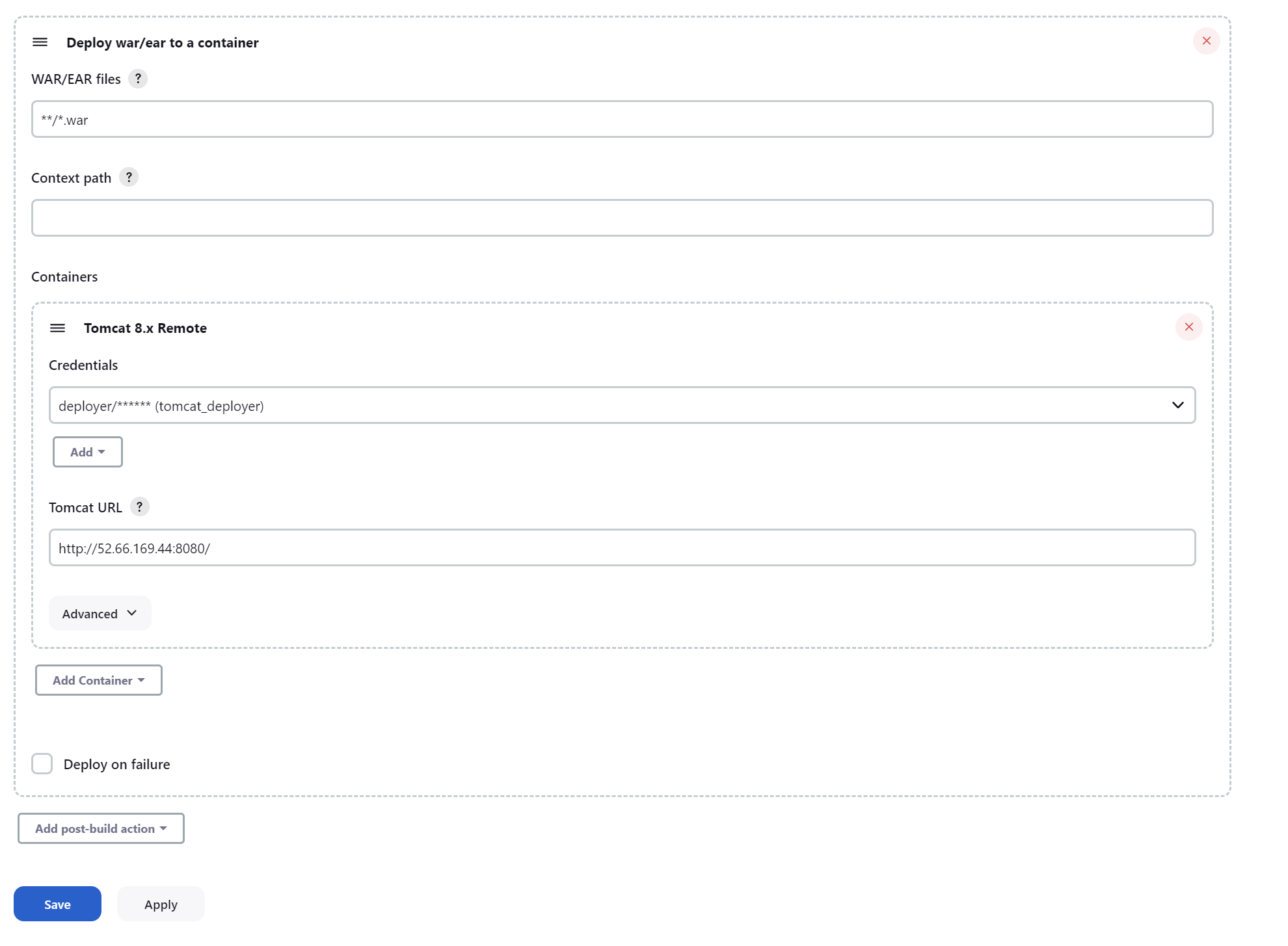




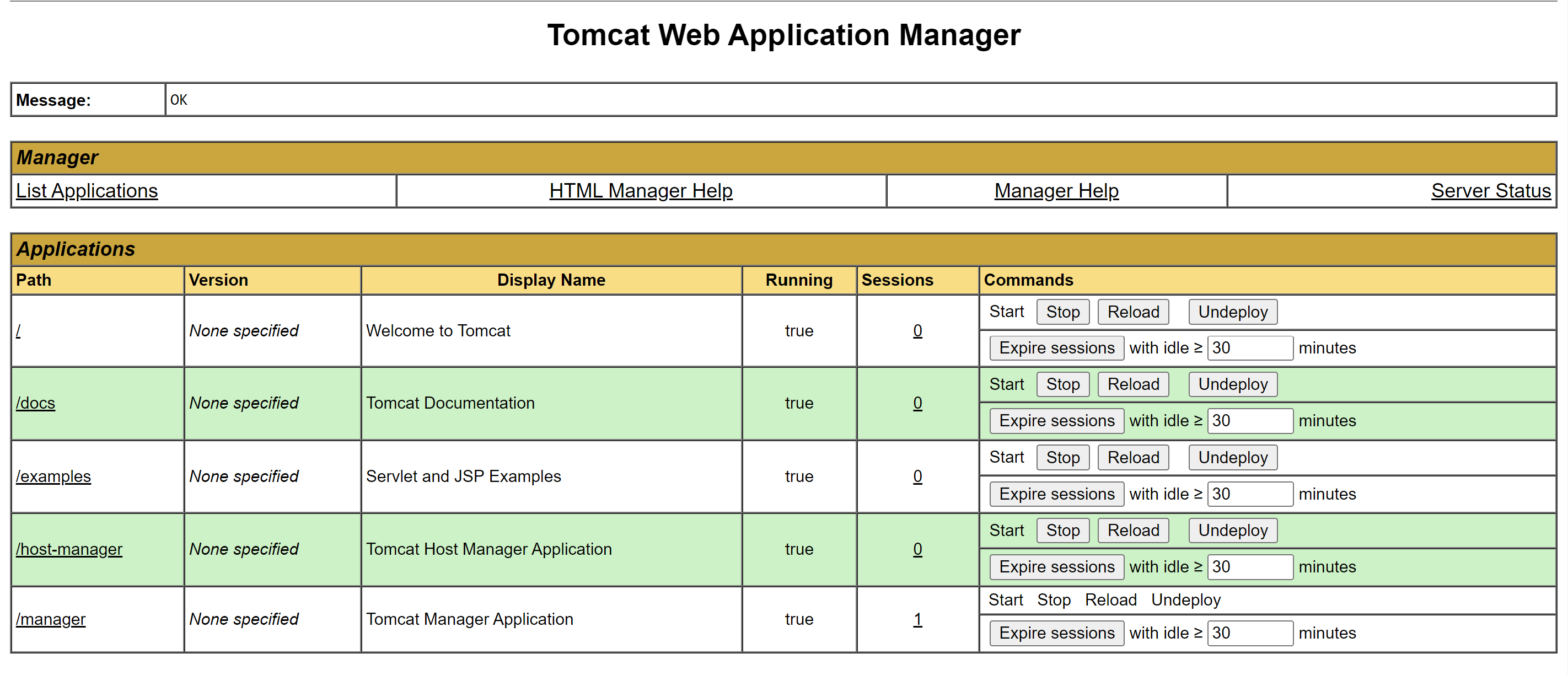
We will create another Jenkins project for “build and deploy” and will specify where the war is stored. In Jenkins war is stored at /var/lib/Jenkins/workspace/ as:



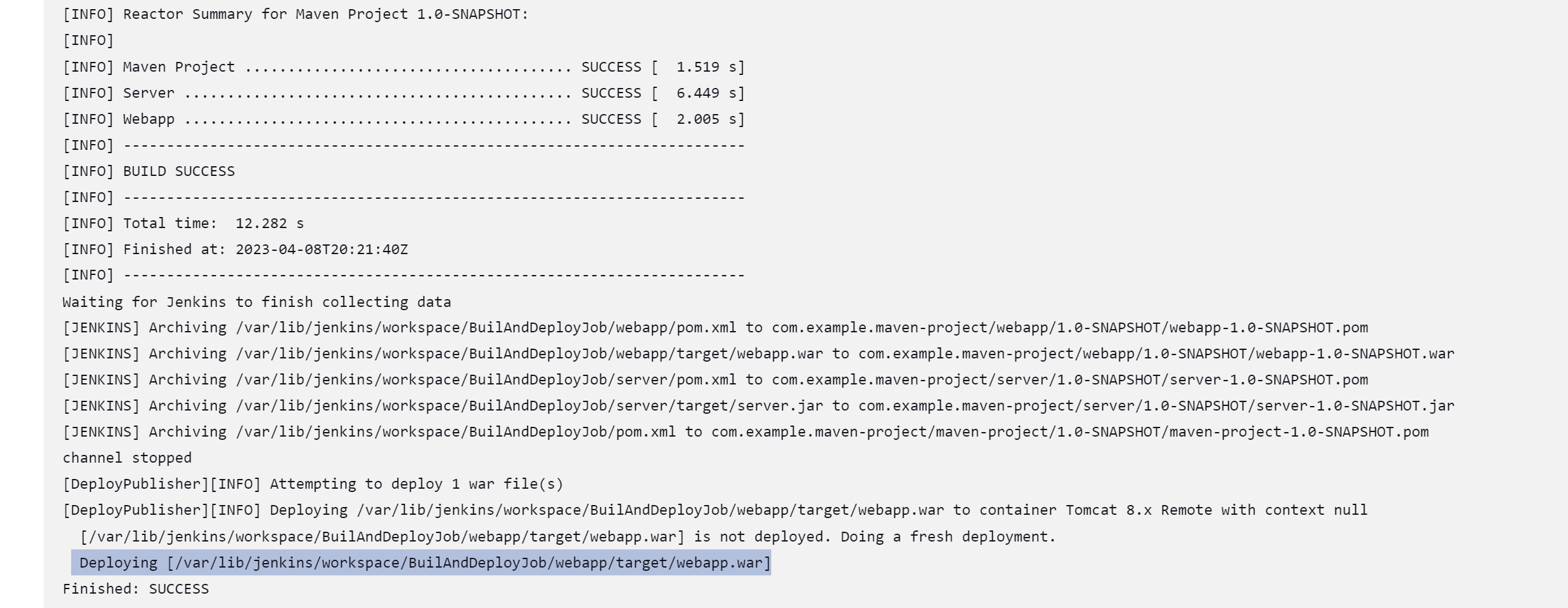
**\*\*/\*.war** means any war file within our project directory in /var/lib/Jenkins/workspace/ourProjectName



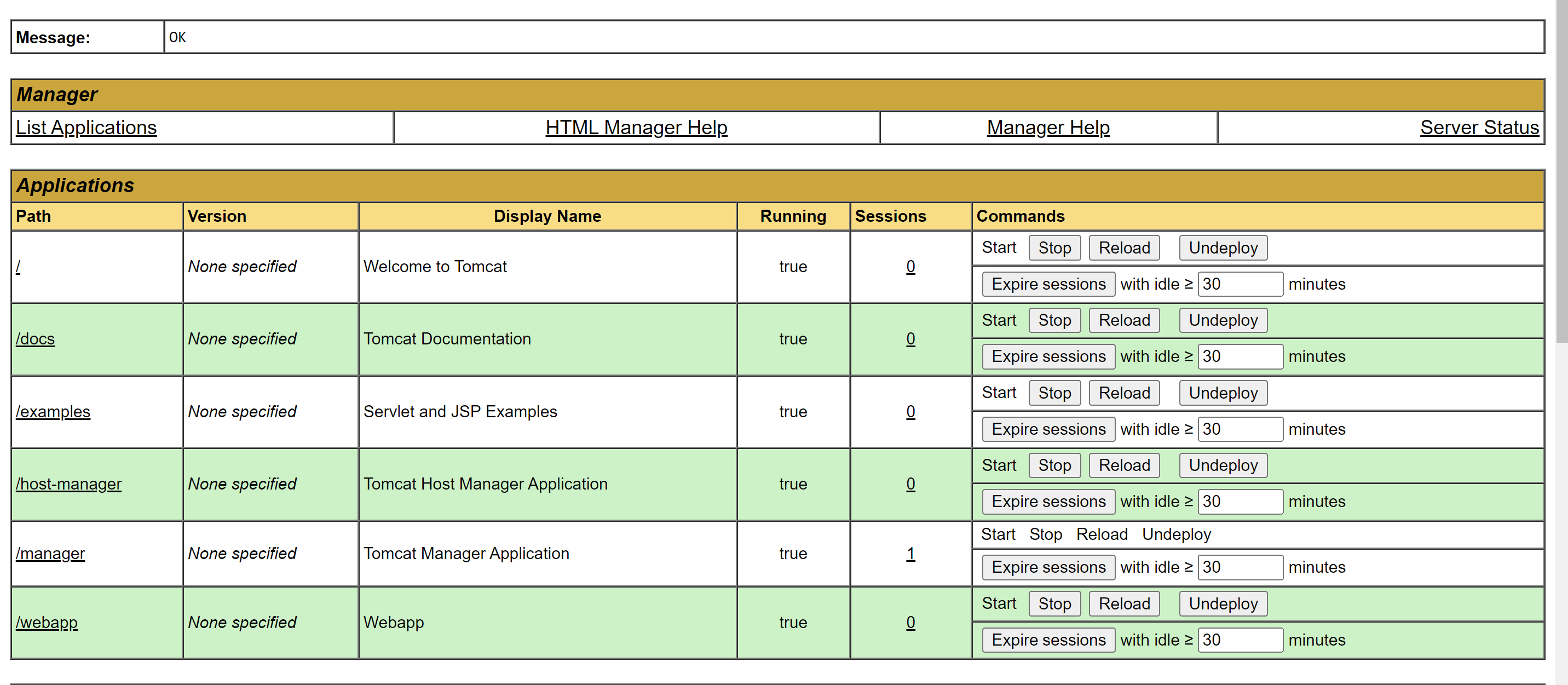
Before running our Jenkins job.



After running our Jenkins Job.



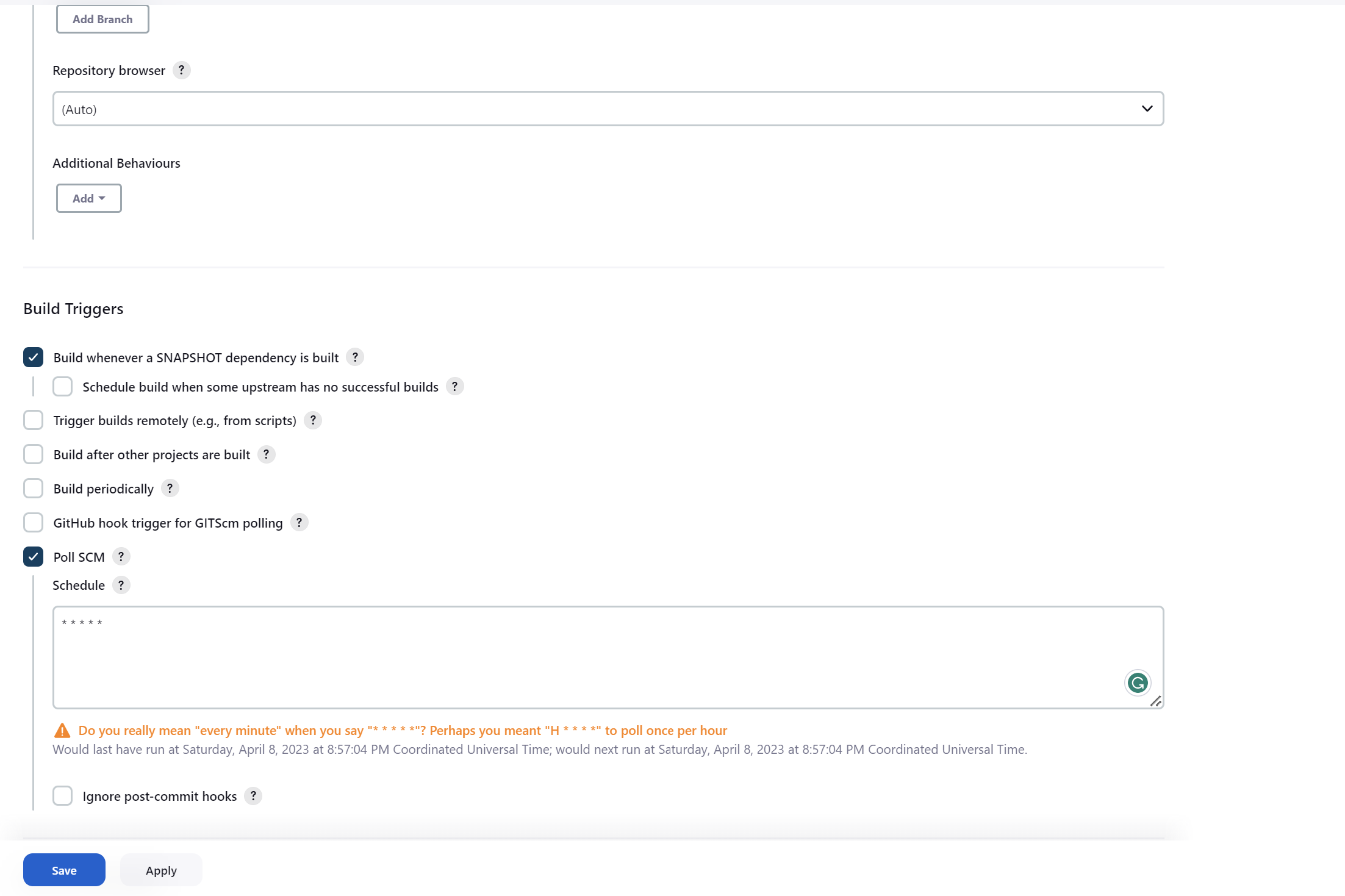
Webapp folder appears, clicking on /webapp will open our application.



Automate deploy code when there is change in code in git.

In build Trigger option in Jenkins build for git we have to enable poll scm. Which will check for any change in git repository.

In git section:



Whenever a new change is commited in git, it will be automatically picked and get deployed.