

# Table of Contents

- DevOps Practice from Alibaba Cloud*** ..... 3
- Install Jenkins ..... 3
- Install Maven ..... 3
- Install Docker ..... 3
- Configuration ..... 3
- Install Terraform ..... 7



# DevOps Practice from Alibaba Cloud

## Install Jenkins

[steps here](#)

## Install Maven

[steps here](#)

## Install Docker

[steps here](#)

## Configuration

### sudo for jenkins user

```
usermod -aG sudo jenkins
```

Modify sudoers with visudo as follows:

```
%sudo    ALL=(ALL:ALL) NOPASSWD: ALL
```

### create build path

```
mkdir /var/lib/jenkins/build  
chown jenkins.jenkins /var/lib/jenkins/build
```

### Jenkins configuration

Enter to web page and follow steps as show on images:

The screenshot shows the Jenkins 'Manage Jenkins' interface. The main heading is 'Manage Jenkins System Configuration'. There are several configuration options:

- Configure System**: Configure global settings and paths.
- Global Tool Configuration**: Configure tools, their locations and automatic installers.
- Manage Plugins**: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Manage Nodes and Clouds**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Security**:
  - Configure Global Security**: Secure Jenkins; define who is allowed to access/use the system.
  - Manage Credentials**: Configure credentials.
  - Configure Credential Providers**: Configure the credential providers and types.

On the left sidebar, there are links for 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Lockable Resources', and 'New View'. Below these are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle).

The screenshot shows the 'Global Tool Configuration' page in Jenkins. It is divided into sections for 'Maven Configuration' and 'JDK'.

**Maven Configuration**

- Default settings provider: Use default maven settings
- Default global settings provider: Use default maven global settings

**JDK**

JDK installations

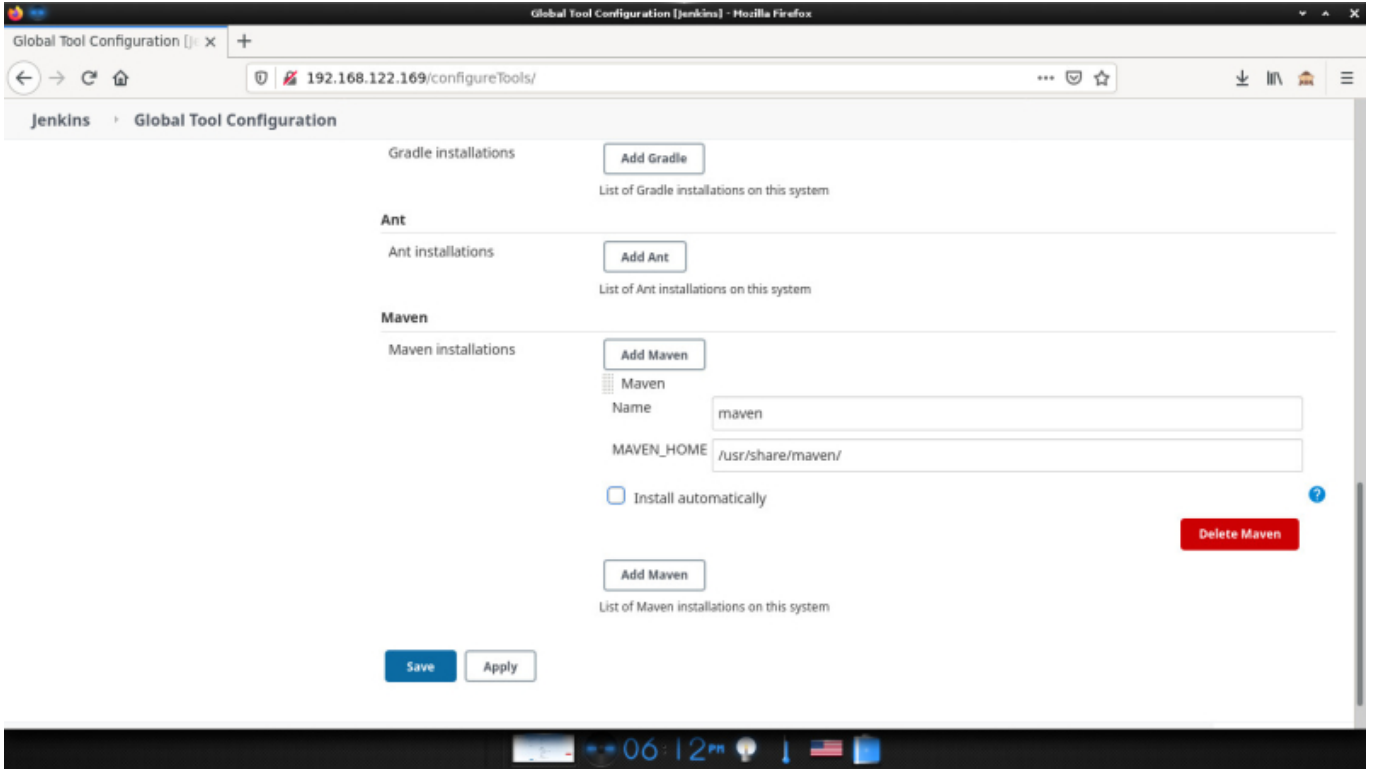
- Add JDK** button
- Existing installation:
  - Name: openjdk
  - JAVA\_HOME: /usr/lib/jvm/java-11-openjdk-amd64/
  - Install automatically
  - Delete JDK** button
- Add JDK** button

List of JDK installations on this system

**Git**

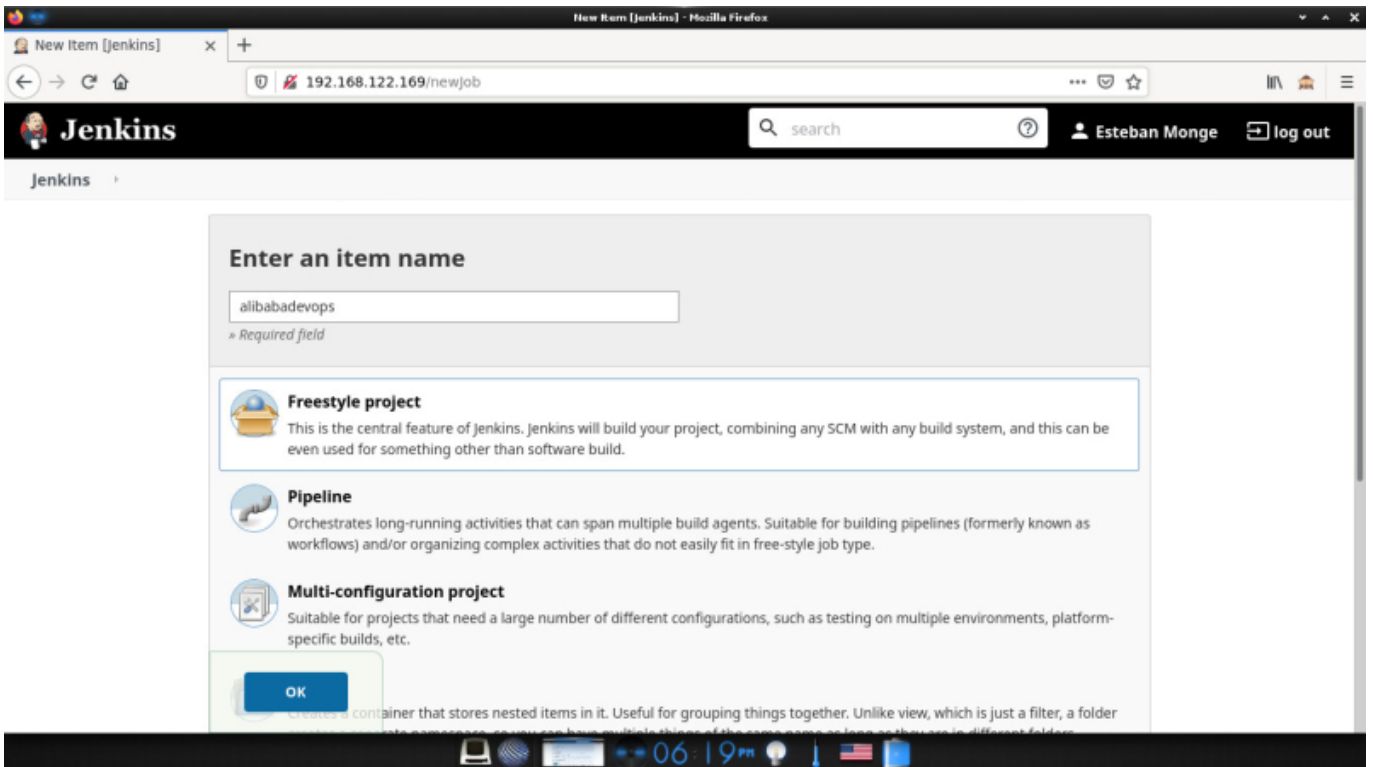
Buttons: **Save**, **Apply**

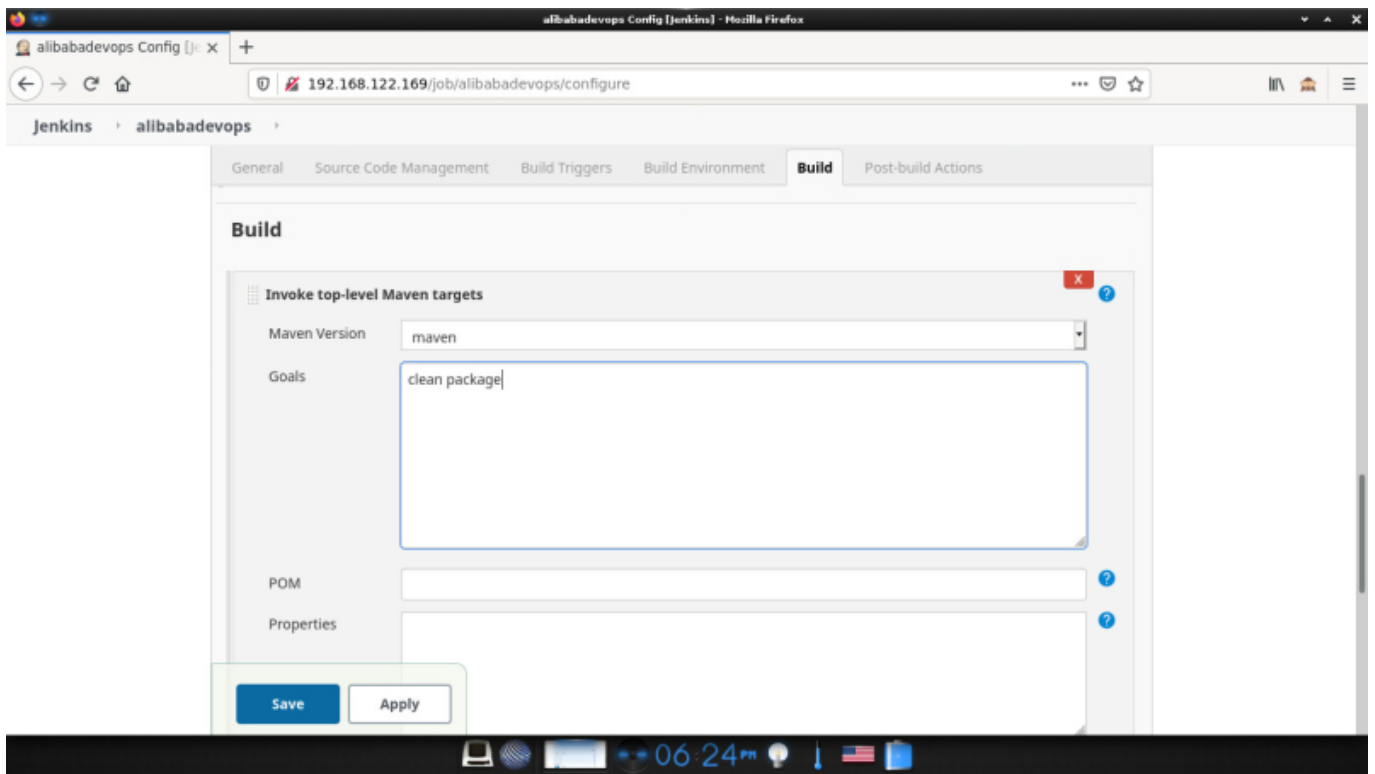
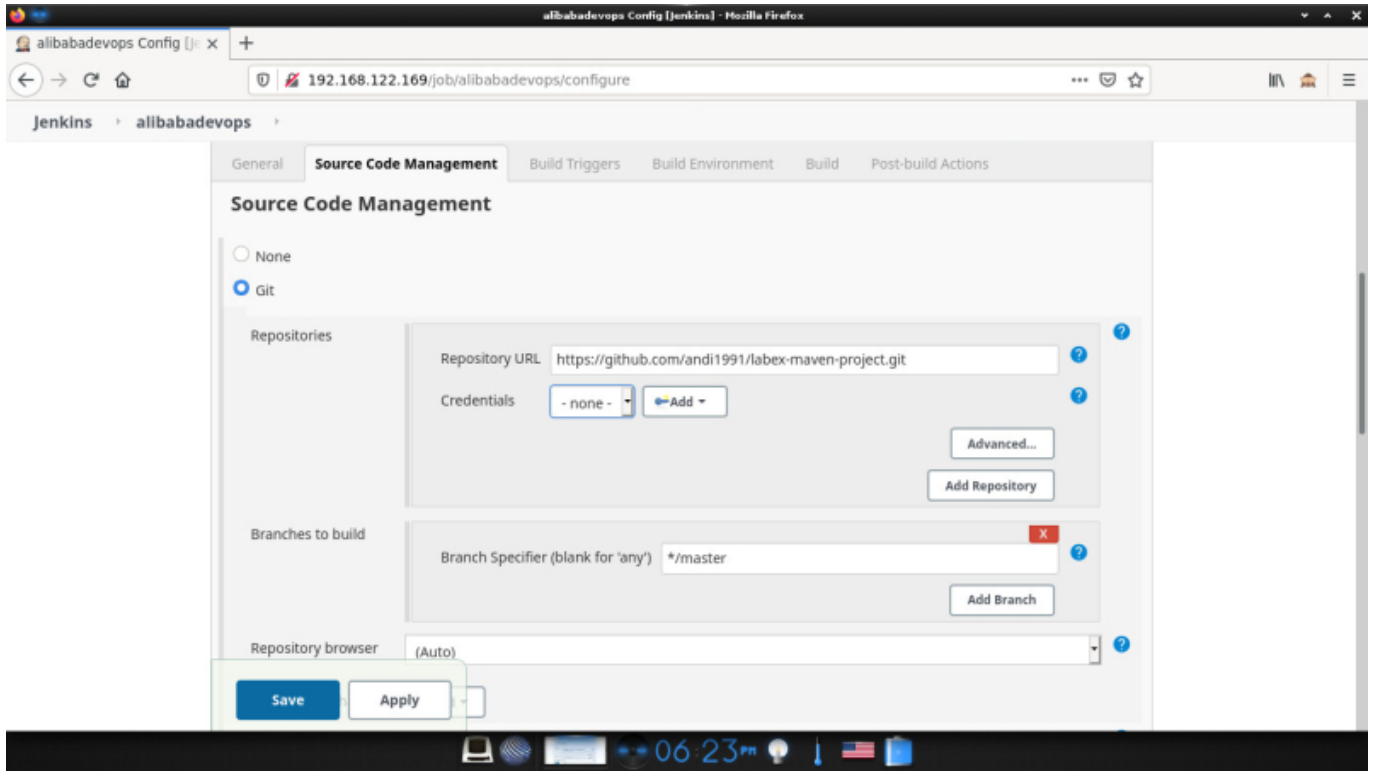
/usr/lib/jvm/java-11-openjdk-amd64/

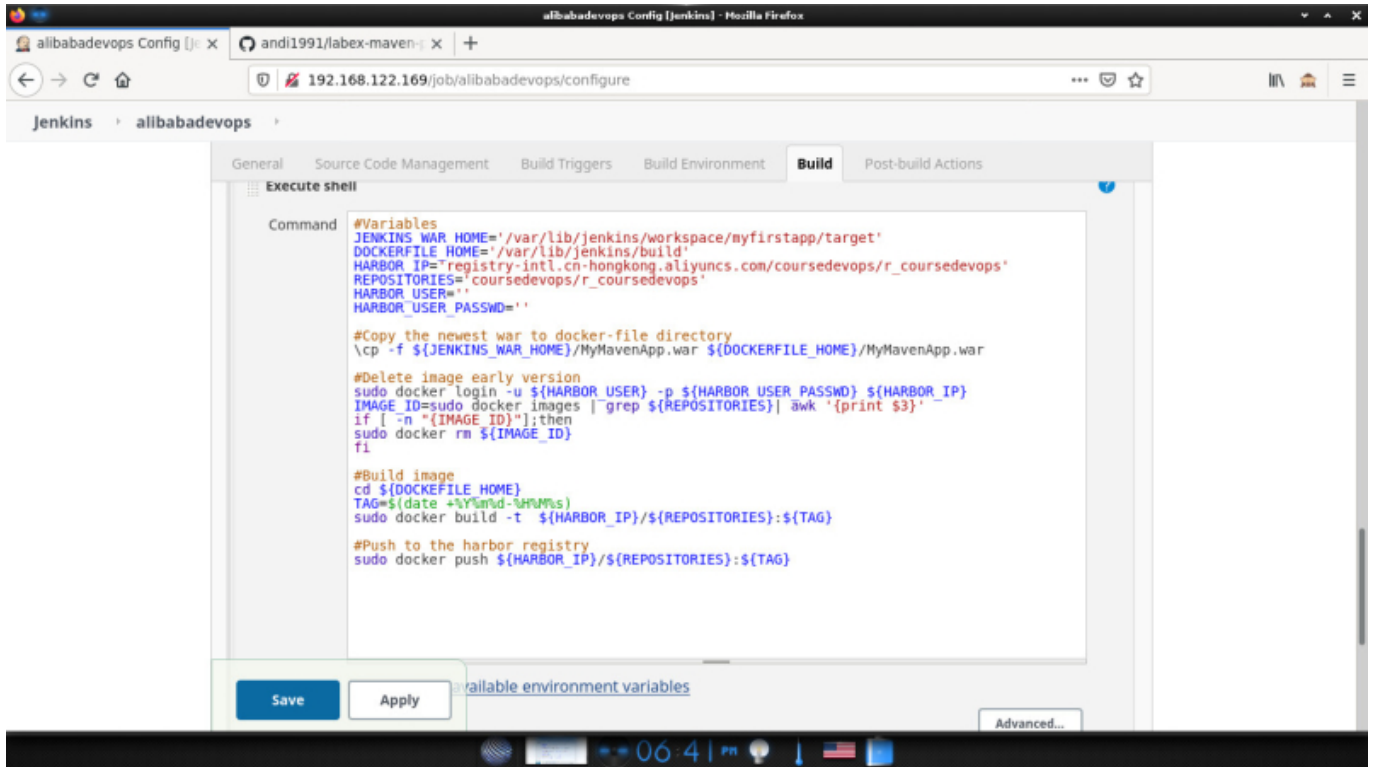


/usr/share/maven/

Create a new Freestyle project job:







```
#Variables
JENKINS_WAR_HOME='/var/lib/jenkins/workspace/alibabadevops/target'
DOCKERFILE_HOME='/var/lib/jenkins/build'
HARBOR_IP='registry-intl.cn-hongkong.aliyuncs.com/coursedevelops/r_coursedevelops'
REPOSITORIES='coursedevelops/r_coursedevelops'
HARBOR_USER=''
HARBOR_USER_PASSWD=''

#Copy the newest war to docker-file directory
\cp -f ${JENKINS_WAR_HOME}/MyMavenApp.war ${DOCKERFILE_HOME}/MyMavenApp.war

#Delete image early version
sudo docker login -u ${HARBOR_USER} -p ${HARBOR_USER_PASSWD} ${HARBOR_IP}
IMAGE_ID=$(sudo docker images | grep ${REPOSITORIES} | awk '{print $3}')
if [ -n "${IMAGE_ID}" ];then
sudo docker rm ${IMAGE_ID}
fi

#Build image
cd ${DOCKERFILE_HOME}
TAG=$(date +%Y%m%d-%H%M%S)
sudo docker build -t ${HARBOR_IP}/${REPOSITORIES}:${TAG}

#Push to the harbor registry
sudo docker push ${HARBOR_IP}/${REPOSITORIES}:${TAG}
```

## Install Terraform

[steps here](#)

From:

<https://estebanmonge.site/> - **Esteban Monge**

Permanent link:

[https://estebanmonge.site/doku.php?id=alibaba\\_devops](https://estebanmonge.site/doku.php?id=alibaba_devops)

Last update: **2020/09/05 10:47**

