

Table of Contents

<i>DevOps Practice from Alibaba Cloud</i>	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. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Manage Jenkins' (which is selected), 'My Views', 'Lockable Resources', and 'New View'. Below this are sections for 'Build Queue' (empty) and 'Build Executor Status' (2 Idle). The main content area is titled 'Manage Jenkins' and 'System Configuration'. It includes several configuration links: 'Configure System' (gear icon), 'Global Tool Configuration' (wrench and screwdriver icon), 'Manage Plugins' (puzzle piece icon), 'Manage Nodes and Clouds' (server icon), 'Configure Global Security' (padlock icon), 'Manage Credentials' (key icon), and 'Configure Credential Providers' (key icon). A search bar at the top right contains the placeholder 'search'.

The screenshot shows the Jenkins Global Tool Configuration interface. At the top, it says 'Global Tool Configuration [Jenkins] - Mozilla Firefox'. The main section is titled 'Maven Configuration' and shows fields for 'Default settings provider' (set to 'Use default maven settings') and 'Default global settings provider' (set to 'Use default maven global settings'). Below this is the 'JDK' section, which lists an 'openjdk' entry with 'Name' set to 'openjdk' and 'JAVA_HOME' set to '/usr/lib/jvm/java-11-openjdk-amd64/'. There are checkboxes for 'Install automatically' and 'Delete JDK'. At the bottom, there's a 'Git' section and two buttons: 'Save' and 'Apply'. The status bar at the bottom of the browser window shows the time as 06:07 pm.

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

The screenshot shows the Jenkins Global Tool Configuration interface. Under the Maven section, a new Maven installation is being configured. The 'Name' field is set to 'maven' and the 'MAVEN_HOME' field is set to '/usr/share/maven/'. A checkbox for 'Install automatically' is checked. At the bottom right of the configuration form is a red 'Delete Maven' button.

/usr/share/maven/

Create a new Freestyle project job:

The screenshot shows the Jenkins 'New Item' creation dialog. In the 'Enter an item name' field, the text 'alibabadevops' is entered. Below the input field, there are three project type options: 'Freestyle project', 'Pipeline', and 'Multi-configuration project'. The 'Freestyle project' option is highlighted with a blue border. A tooltip for 'Freestyle project' states: 'This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.' An 'OK' button is visible at the bottom left of the dialog.

The screenshot shows the Jenkins job configuration interface for the 'alibabadevops' job. The 'Source Code Management' tab is selected, displaying the following configuration:

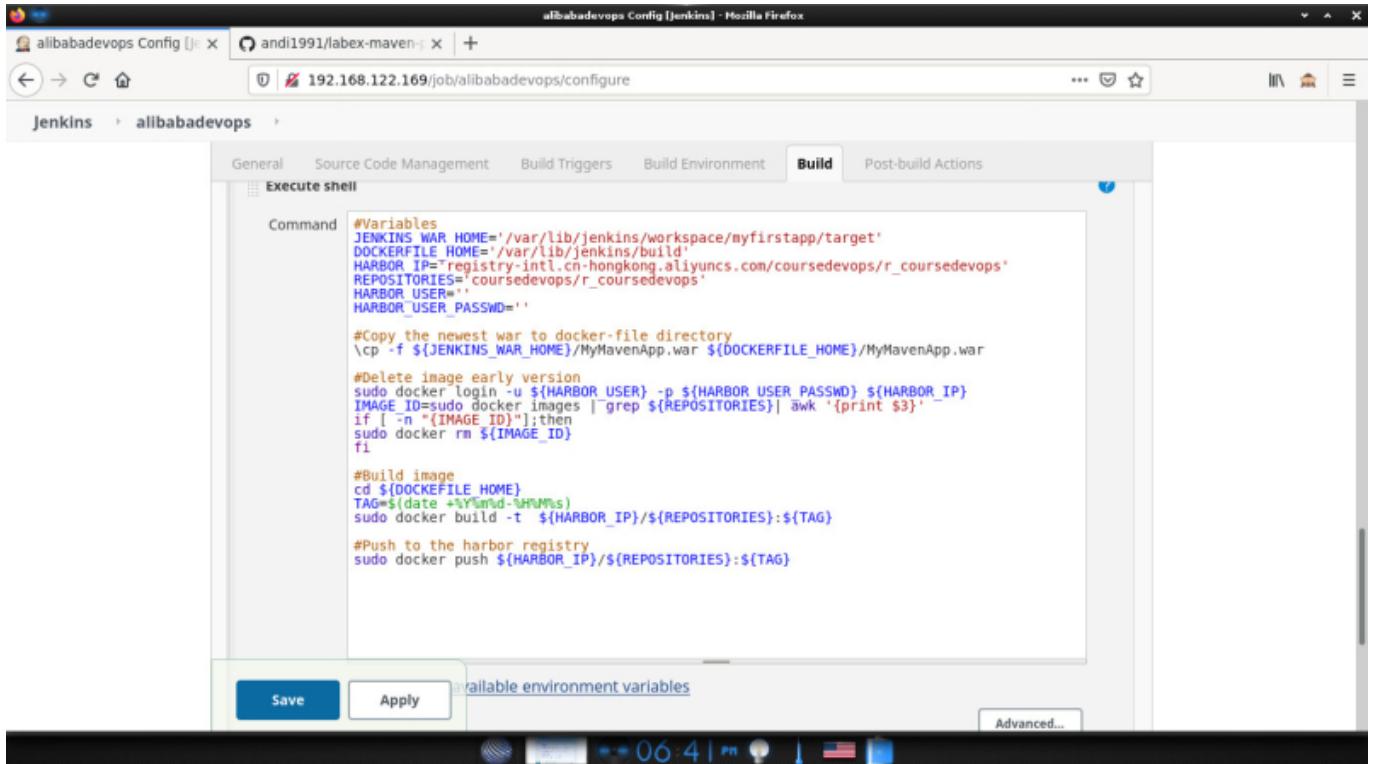
- Repository type: Git (selected)
- Repository URL: <https://github.com/andi1991/labex-maven-project.git>
- Credentials: - none - (dropdown menu)
- Branch Specifier (blank for 'any'): */master
- Repository browser: (Auto)

At the bottom are 'Save' and 'Apply' buttons.

The screenshot shows the Jenkins job configuration interface for the 'alibabadevops' job. The 'Build' tab is selected, displaying the following configuration:

- Invoke top-level Maven targets:
 - Maven Version: maven
 - Goals: clean package
- POM: (empty field)
- Properties: (empty field)

At the bottom are 'Save' and 'Apply' buttons.



```

#Variables
JENKINS_WAR_HOME='/var/lib/jenkins/workspace/alibabadevops/target'
DOCKERFILE_HOME='/var/lib/jenkins/build'
HARBOR_IP='registry-intl.cn-
hongkong.aliyuncs.com/coursedevops/r_coursedevops'
REPOSITORIES='coursedevops/r_coursedevops'
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

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