

Jenkins+docker+git+maven 构建项目

环境：2 台云服务器

A 服务器环境：1.8jdk, docker,ssh 外网访问地址 222.24.63.59 端口 8080:9064
8081:9066

B 服务器环境：1.8jdk, docker,ssh 外网访问地址 222.24.63.59, 端口 8080:9065
8081:9067

思路：在 A 服务器上从 git 上获得源码，通过 mvn 打包成.jar，再上传到 B 服务器，在 B 服务器上生成镜像，然后运行。

1.从 A 服务器上 docker run 一个 jenkins 镜像 (chmod 777

/usr/jenkins/jenkins-home)

给文件权限否则不可读入

```
docker run -p 8081:8080 -d -v /usr/jenkins/jenkins-home:/var/jenkins_home  
jenkins/jenkins:lts
```

```
help'.  
rmycl9p4j2z ~]# docker run -p 8082:8080 -d -v /usr/jenkins/jenkins-home:/var/jenkins_home  
ge 'jenkins/jenkins:lts' locally  
jenkins/jenkins  
complete  
complete  
complete  
complete  
complete
```

在此之前需要(chmod 777 /usr/jenkins/jenkins-home)

```
ns-home]# cd ../  
ns]# chmod 777 jenkins-home  
el# docker run -p 8082:8080 -d
```

给文件权限否则不可读入

2. 获取初始密码并访问,输入密码后等待时间有点长

```
[root@izwz97dq477rmycl9p4j2z /]# cat /usr/jenkins/jenkins-home/secrets/initialAdmin  
7bc7365clf834ebf82905d8d5463af7a  
[root@izwz97dq477rmycl9p4j2z /]#
```

访问

<http://222.24.63.59:9066/>

解锁 Jenkins

为了确保管理员安全地安装 Jenkins，密码已写入到日志中（[不知](#)在服务器上：

```
/var/jenkins_home/secrets/initialAdminPassword
```

请从本地复制密码并粘贴到下面。

管理员密码

.....

 此连接不安全。在此页面输入的登录信息可以被窃取。详细了解

4. 进入后选择安装插件 先安装推荐一波

自定义 Jenkins

插件通过附加特性来扩展 Jenkins 以满足不同的需求。

安装推荐的插件

安装 Jenkins 社区推荐的插件。

选择插件来安装

选择并安装最适合的插件。

注意：可能存在下面的问题



解决方法：<https://www.cnblogs.com/ninefish/p/9818080.html>

结束后创建一个账号，手动安装插件。



管理JENKINS



系统设置

全局设置和路径



全局安全配置

Jenkins安全，定义谁可以访问或使用



凭据配置

配置凭据的提供者和类型



全局工具配置

工具配置，包括它们的位置和自动安



读取设置

放弃当前内存中所有的设置信息并从



插件管理

添加、禁用或启用Jenkins功能



系统信息

通过手动安装 Push Over SSH 插件和 Maven Integration plugin



5.配置全局工具



管理Jenkins



系统设置

全局设置和路径



全局安全配置

Jenkins安全，定义谁可以访问或使用系统



凭据配置

配置凭据的提供者和类型



全局工具配置

工具配置，包括它们的位置和自动安装器
全局工具配置



读取设置

放弃当前内存中所有的设置信息并从配置文件中读取

git 和 jdk 在镜像中已经存在，jdk 目录如下（感兴趣的可以去找找）

JDK

JDK 安装

新增 JDK

JDK

别名

jdk1.8

JAVA_HOME

/usr/lib/jvm/java-1.8.0-openjdk-amd64

☐ 自动安装

新增 JDK

系统下JDK 安装列表

Git

Git installations

Git

Name

Default

Path to Git executable

git

☐ 自动安装

没有 maven 环境所以选择自动安装。构建时会自动下载

系统下Ant 安装列表

Maven

Maven 安装

新增 Maven

Maven

Name maven

☒ 自动安装

从 Apache 安装

版本 3.6.0

新增安装

新增 Maven

系统下Maven 安装列表

Docker

保存

应用

https://blog.csdn.net/qq_434

6.系统设置

这里可以配置通知邮件（需要去获取发送邮件权限）再进行配置（这里就不配了）

POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV服务

开启服务：	POP3/SMTP服务（如何使用 Foxmail 等软件收发邮件？）	已开启 关闭
	IMAP/SMTP服务（什么是 IMAP，它又是如何设置？）	已关闭 开启
	Exchange服务（什么是Exchange，它又是如何设置？）	已关闭 开启
	CardDAV/CalDAV服务（什么是CardDAV/CalDAV，它又是如何设置？）	已关闭 开启
	(POP3/IMAP/SMTP/CardDAV/CalDAV服务均支持SSL连接。如何设置？)	

温馨提示：在第三方登录QQ邮箱，可能存在邮件泄露风险，甚至危害Apple ID安全，建议使用QQ邮箱官方客户端。

<https://blog.csdn.net/>

配置其他服务器，这里去配置连上 B 服务器

Disable exec

SSH Servers

☐

SSH Server

Name

docker-1

Hostname



Username

root

Remote Directory

/

☒ Use password authentication, or use a different

Passphrase / Password

.....

Path to key

Key

7. 创建 maven 任务

输入一个任务名称

jenkins-test

» 必填项



构建一个自由风格的软件项目

这是Jenkins的主要功能.Jenkins将会结合任何SCM和任何构建系统来构建你的项目,甚至



构建一个maven项目

构建一个maven项目.Jenkins利用你的POM文件,这样可以大大减轻构建配置.



流水线

精心地组织一个可以长期运行在多个节点上的任务。适用于构建流水线（更加正式地应当型。

https://blog.csdn.net/qq_34374609/article/details/80000000

配置 git 地址

☐ 无
☒ Git

Repositories

Repository URL

Credentials

Branches to build

Branch Specifier (blank for 'any')

配置 mvn shell 打包成.jar

```
cd /usr/jenkins/jenkins-home/workspace/jenkins-test/demo0-0
```

```
mvn clean package -Dmaven.test.skip=true
```



jenkins-test 是这个任务的名称，运行时会生成一个文件夹，这个文件夹下是从 git 上下载下来的 demo0-0

注意：我这上传到 git 上的是 demo0-0 文件夹目录结构如下。

master ▼ **jenkintest / demo0-0**

 于 3天前 Signed-off-by: mengzuoxin <1203224763@qq.com>

← ...

 .mvn/ wrapper  Signed-off-by: men

 src  Signed-off-by: men

 .gitignore  Signed-off-by: men

 mvnw  Signed-off-by: men

 mvnw.cmd  Signed-off-by: men

 pom.xml  Signed-off-by: men

https://blog.csdn.net/qq_43482425

Build

Root POM

Goals and options

最后一步通过前面配置的服务器 B。上传.jar 到服务器 B 并生成镜像，运行

Send build artifacts over SSH

SSH Publishers

SSH Server

Name:

Transfers

Transfer Set

Source files:

Remove prefix:

Remote directory:

Exec command:

All of the transfer fields (except for Exec timeout, substitution of [Jenkins environment variables](#)

Source files :.jar 的相对路径，相对与这个任务 jenkins-test 文件夹

Remove prefix: 去除前面不需要上传的部分。

Remote directory: 需要上传到服务器 B 那个目录下。

Exec command: 上传后再 B 上运行的脚本。

最后点击保存。