第2章 Jenkins Server的安装部署方式

Jenkins Server的安装部署方式

1.1 安装依赖

Jenkins是基于Java开发的一种持续集成工具，可以在安装Java Runtime Environment的任何机器独立运行。部署前需要安装Java开发环境。要求是JDK8+，目前已经支持JDK11。关于JDK的安装可以通过yum或者源码包方式。

rpm方式

-
-

```
rpm -ivh jdk-xxxxx.rpm
yum -y install java-1.8.0-openjdk java-1.8.0-openjdk-devel
```

源码包方式

```
tar zxf jdk-xxxx.tar.gz -C /usr/local
vim /etc/profile
export JAVA_HOME=/usr/local/jdkxxx
export PATH=$PATH:$JAVA_HOME/bin
source /etc/profile
```

1.2 安装部署

1.2.1 使用Tomcat部署

如果使用war包部署建议采用tomcat，当然Jenkins的war包可以直接通过`java -jar jenkins.war`启动。我们可以部署一个tomcat服务然后将jenkins.war包放到tomcat的webapps目录下。最后通过浏览器`http://tomcatserver/jenkins`访问。

```
wget http://mirror.bit.edu.cn/apache/tomcat/tomcat-9/v9.0.24/bin/apache-tomcat-9.0.24.tar.gz
tar xf apache-tomcat-9.0.24.tar.gz -C /usr/local/
cp jenkins.war apache-tomcat-9.0.24/webapps/
# 启动/usr/local/tomcat/bin/catalina.sh start
# 停止/usr/local/tomcat/bin/shutdown.sh
```

1.2.2 CentOS环境部署

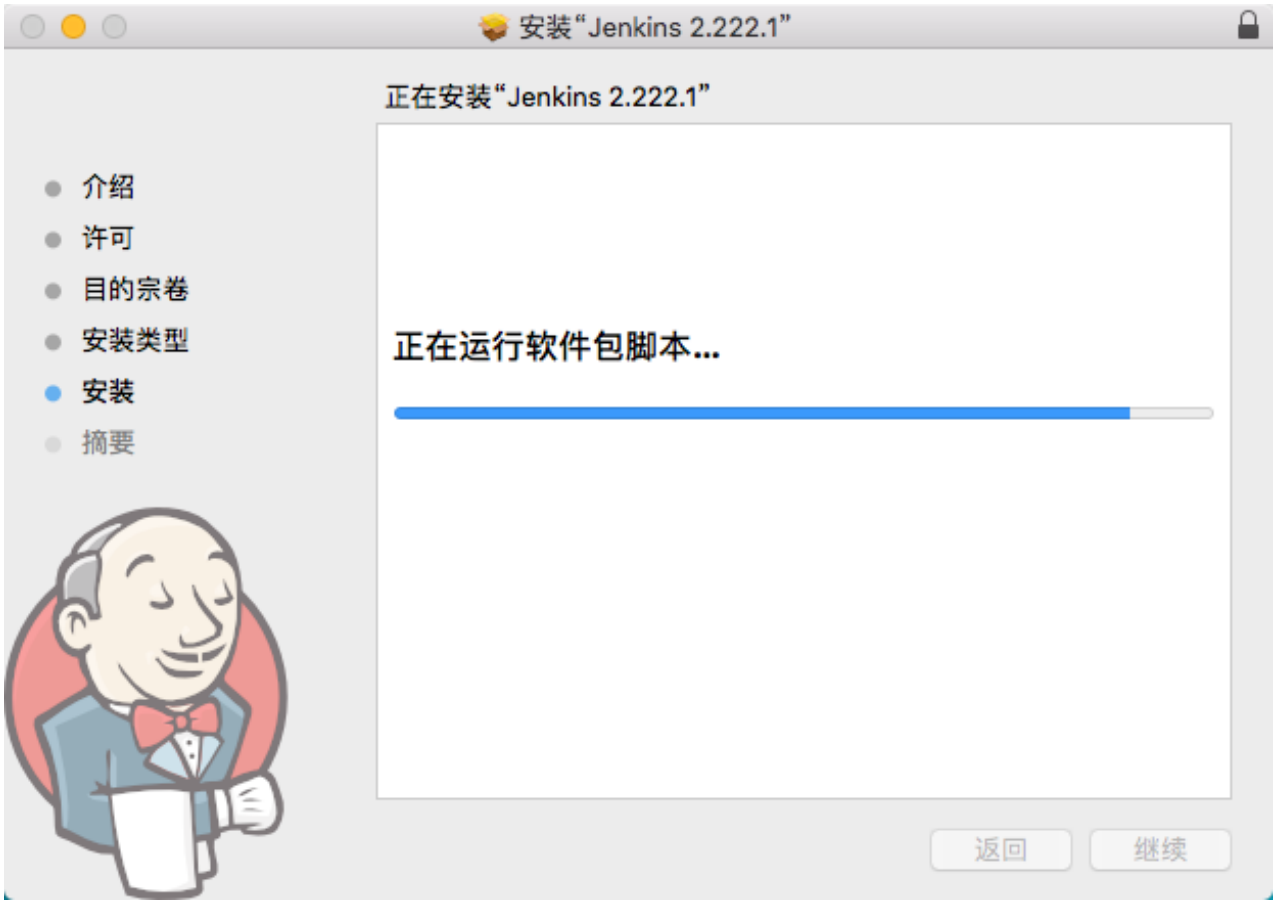
普遍在centos系统部署Jenkins，可以下载jenkins的最新rpm包部署。

```
wget https://mirrors.tuna.tsinghua.edu.cn/jenkins/redhat-stable/jenkins-2.235.2-1.1.noarch.rpm
rpm -ivh jenkins-2.235.2-1.1.noarch.rpm
systemctl start jenkinssystemctl enable jenkins
```

1.2.3 MacOS环境部署

在Mac OS X机器上安装Jenkins CI服务器，可以手动下载pkg包安装或者通过brew自动安装。

手动下载pkg包安装，根据向导进一步安装。



使用brew部署

```
brew install jenkins-lts

##############################################################################
100.0%==> Downloading http://mirrors.jenkins.io/war-
stable/2.235.2/jenkins.war==> Downloading from
http://mirror.serverion.com/jenkins/war-
stable/2.235.2/jenki##############################################################
############# 100.0%==> Installing dependencies for jenkins-lts: openjdk@11==>
Installing jenkins-lts dependency: openjdk@11==> Pouring openjdk@11-
11.0.7+10.high_sierra.bottle.tar.gz==> CaveatsFor the system Java wrappers to
find this JDK, symlink it with  sudo ln -sfn
/usr/local/opt/openjdk@11/libexec/openjdk.jdk
/Library/Java/JavaVirtualMachines/openjdk-11.jdk
openjdk@11 is keg-only, which means it was not symlinked into
/usr/local,because this is an alternate version of another formula.
If you need to have openjdk@11 first in your PATH run:  echo 'export
PATH="/usr/local/opt/openjdk@11/bin:$PATH"' >> /Users/zeyang/.bash_profile
For compilers to find openjdk@11 you may need to set:  export CPPFLAGS="-
I/usr/local/opt/openjdk@11/include"
==> Summary🍺  /usr/local/Cellar/openjdk@11/11.0.7+10: 650 files, 295.7MB==>
Installing jenkins-lts==> /usr/local/opt/openjdk@11/bin/jar xvf jenkins.war==>
CaveatsNote: When using launchctl the port will be 8080.
```

```
To have launchd start jenkins-lts now and restart at login:  brew services
start jenkins-ltsOr, if you don't want/need a background service you can just
run:  jenkins-lts==> Summary🍺  /usr/local/Cellar/jenkins-lts/2.235.2: 7 files,
66.4MB, built in 11 seconds==> Caveats==> openjdk@11For the system Java
wrappers to find this JDK, symlink it with  sudo ln -sfn
/usr/local/opt/openjdk@11/libexec/openjdk.jdk
/Library/Java/JavaVirtualMachines/openjdk-11.jdk
openjdk@11 is keg-only, which means it was not symlinked into
/usr/local,because this is an alternate version of another formula.
If you need to have openjdk@11 first in your PATH run:  echo 'export
PATH="/usr/local/opt/openjdk@11/bin:$PATH"' >> /Users/zeyang/.bash_profile
For compilers to find openjdk@11 you may need to set:  export CPPFLAGS="-
I/usr/local/opt/openjdk@11/include"
==> jenkins-ltsNote: When using launchctl the port will be 8080.
To have launchd start jenkins-lts now and restart at login:  brew services
start jenkins-ltsOr, if you don't want/need a background service you can just
run:  jenkins-lts       # Start.brew services start jenkins-lts
# Stop.brew services stop jenkins-lts
```

1.2.4 基于Docker部署

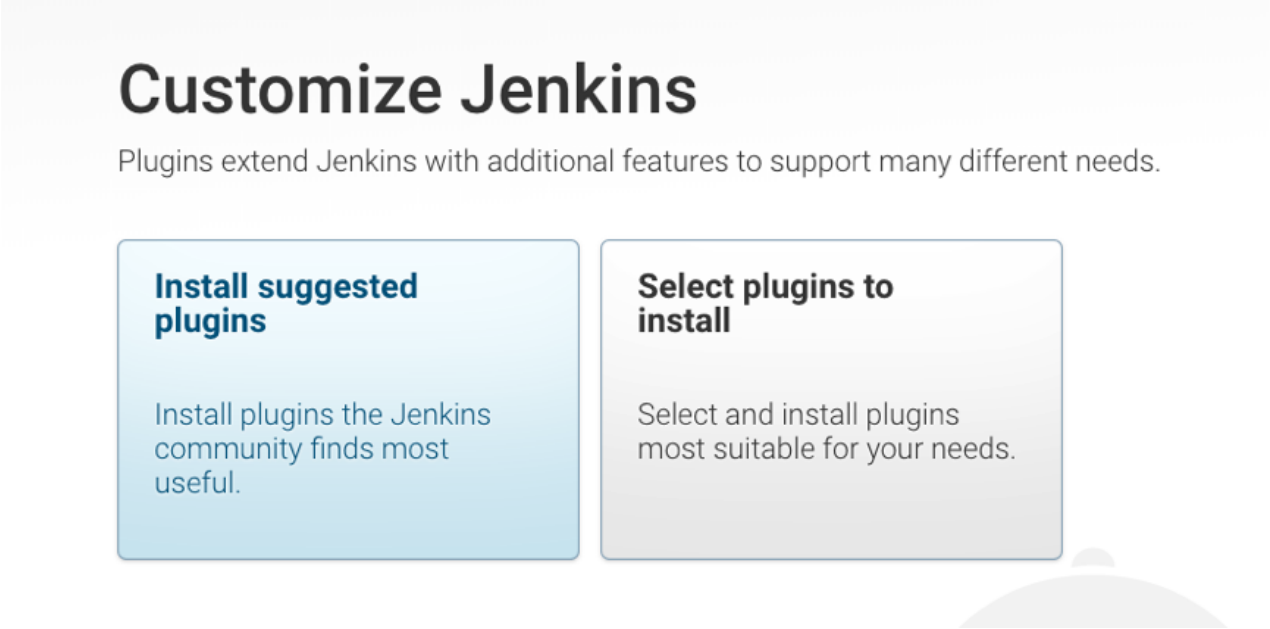
```
docker pull jenkins/jenkins:lts
docker run -d -v jenkins_home:/var/jenkins_home -p 8080:8080 -p 50000:50000
jenkins/jenkins:lts

## 参数解释-d 后台运行-v 持久化jenkins数据目录-p 端口绑定  server:8080  agent:50000
```

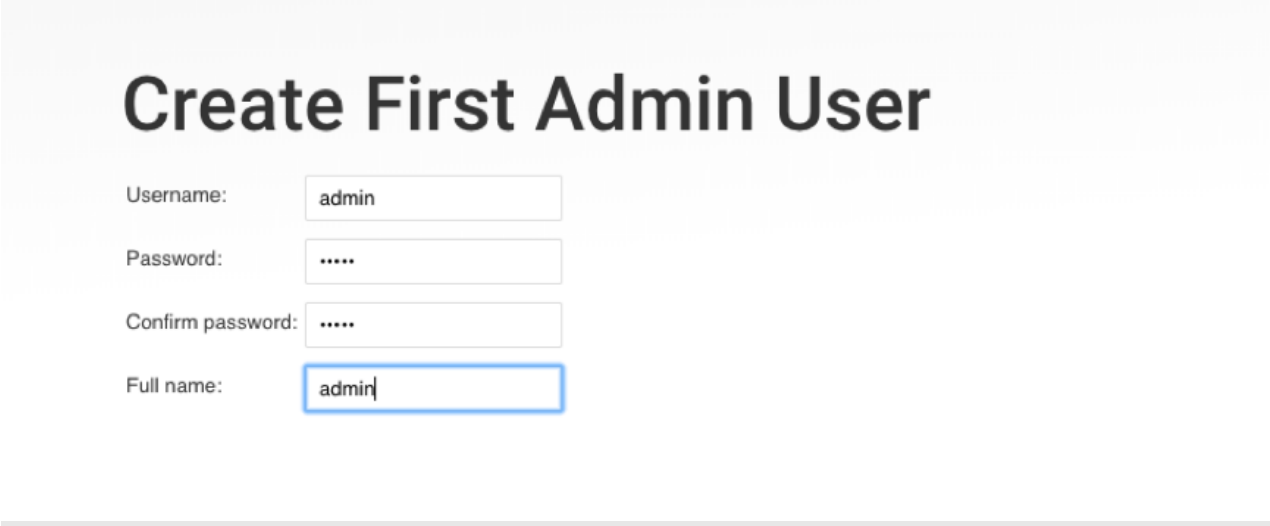
1.3 配置

解锁：当您第一次访问Jenkins的时候，系统会要求您使用自动生成的密码对其进行解锁。解锁秘钥可以通过$JENKINS_HOME/secrets/initialAdminPassword文件获取。还可以通过在启动日志中获取。

安装插件：分别是安装默认的插件和自定义选择要安装的插件，推荐还是使用默认的插件安装。



创建用户：在这里创建一个管理员账号，到此jenkins的初始化配置已完成。



配置更新站点：由于使用官方的站点速度相对很慢，这里采用清华大学的jenkins更新站点。

这块经常出现问题：当改完后其实还是很慢，因为清华源代理的json文件的内容中还是指向的外网地址。推荐几种方法：

方法1：使用jenkins中文社区的插件源。安装中文社区插件（插件搜索关键字chinese），然后点击页面最下方超链接。最后会有提示使用插件源。

方法2: 配置代理域名转发

-
-
-

```
location /download {    proxy_pass
http://mirrors.tuna.tsinghua.edu.cn/jenkins/;}
```

方法3:进入 jenkins 工作目录，目录下面有一个 updates 的目录，下面有一个 default.json 文件，我们执行下面的命令替换插件地址：替换完成后，需要重启 Jenkins。

-

```
$ sed -i 's/http:\/\/updates.jenkins-
ci.org\/download/https:\/\/mirrors.tuna.tsinghua.edu.cn\/jenkins/g'
default.json && sed -i 's/http:\/\/www.google.com/https:\/\/www.baidu.com/g'
default.json
```

我们一般应用最多的应该是在linux或者docker中使用。在linux环境中jenkins默认的配置文件在/etc/sysconfig/jenkins中，默认的启动用户为jenkins，默认端口为8080。后期都是可以基于配置文件进行修改的。