链滴

瞎扯淡：Jenkins 搭建要点

作者：flhuoshan

原文链接：https://ld246.com/article/1500018641100

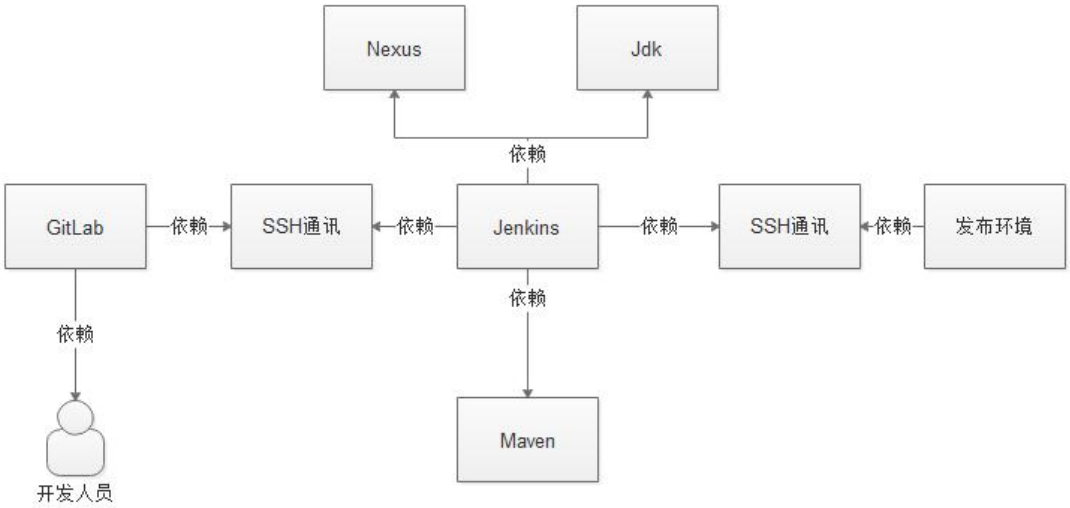
来源网站：链滴

前言

● Jenkins作为持续集成工具，大大的解放了ITER的双手，将大量重复性的，繁琐的更新、构建、部操作融于一体。之前用了一些时间为项目组搭建了一套Jenkins环境，将搭建的步骤做一次梳理，分出来。

概述

● 总的来说，Jenkins是一个协调者的身份，管理和协调了代码库，代码仓库，代码运行环境等。Jenkns和其他部分的依赖关系如图：



步骤

● 有了上述的关系图，我们就可以开始着手搭建CI环境了。

JDK，NEXUS，MAVEN

此三者是Jenkins直接依赖的环境，因此需要先行安装配置。

● 下载安装JDK，配置环境变量。java -version命令查看是否安装成功。

● 下载安装MAVEN，配置环境变量。

● 安装私有的NEXUS， 见此处。

● 关联MAVEN与NEXUS，将MAVEN的远程仓库修改为私有NEXUS(在MAVEN的settings.xml文件设置）。不建议使用MAVEN中央仓库，因为部分jar包被墙且下载速度慢，同时无法发布自己的私有Ja包（审核时间长且不适合放在公网），造成构建发布失败。

Jenkins

● 安装过程略

- 启停命令： service jenkins start|stop
- 默认地址：http://ip:8080   admin/admin

至此，Jenkins + Maven + Nexus + Jdk的环境已经搭好，然而这仅仅是一个开始而已。

GitLab

- 俗话说，巧妇难为无米之炊，目前的Jenkins还没有米下锅。而GitLab就是Jenkins的米仓，Jenkin拿到米之后，将它们做成了香喷喷的饭并发送到部署环境。GitLab安装过程略，说一说怎么完成这个生米煮成熟饭”的过程。
- Jenkins工作的前半部分，是将GitLab上的代码拉到Jenkins本机的环境中，然后Maven打包。拉的程，我们用的是SSH将Jenkins的公钥分发到GitLab库中完成远程传递文件。
- 可以在Jenkins主机上，通过两个命令完成

```
> ssh-keygen -t rsa -P ''
> ssh-copy-id -i ~/.ssh/id_rsa.pub <IP>
```

- 命令一完成了SSH公钥的生成，命令二完成了SSH公钥的分发，公钥分发给GitLab远程主机后，就以免密登陆进而拉取GitLab上的最新代码。

发布环境

- 如果说GitLab是巧妇的米仓，而发布环境就是递到客人手中的饭碗，饭碗中盛的是一个个运行的应程序。

同GitLab一样，需要完成Jenkins到发布环境的公钥分发，方法与上一步同样。

Jenkins其他配置

- 这里主要指，在Jenkins中GitLab和发布环境的配置，具体方法请自行搜索。

其他的东东

- 分享两个服务启停脚本，可以放在真实环境使用：

```
启动：
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-2.b11.el7_3.x86_64
nohup java -server -Xms512M -Xmx512M -Xss256k \
      -XX:+UseStringDeduplication \
      -XX:+HeapDumpOnOutOfMemoryError \
      -jar member.jar \
      --server.port=9092 \
      --spring.profiles.active=pre \
      > /dev/null 2>&1 &

停止：
PID=$(ps -ef | grep member.jar | grep -v grep | awk '{ print $2 }')
if [ -z "$PID" ]
then
    echo Application is already stopped
```

```
else
    echo kill $PID
    kill -9 $PID
fi
```

总结

- 本文叫“Jenkins搭建要点”，是一个对之前做工作的概要总结。因此注定不是一个面面俱到的资。希望看到后的ITER，看完本文之后能对Jenkins有一个比较清晰的认识和判断。