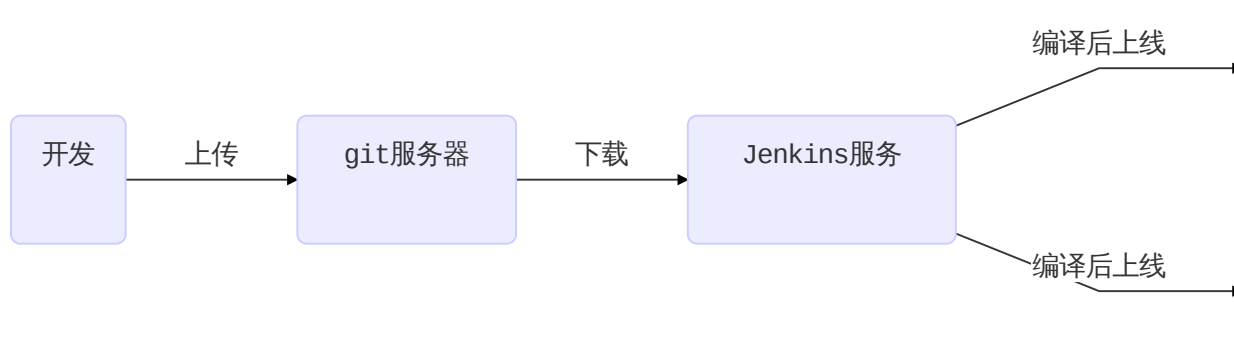


tedu_nsd1812_devops_day05

CI/CD：持续集成/持续交付



Jenkins：实现CI的开源工具

准备Jenkins服务器

配置IP地址、主机名、yum、安装java

安装jenkins

```
[root@node4 ~]# yum install -y jenkins-2.138.2-1.1.noarch.rpm
[root@node4 ~]# systemctl start jenkins
[root@node4 ~]# systemctl enable jenkins
```

初始化jenkins

打开<http://192.168.122.73:8080>。安装插件选择自定义=>无。不用创建管理员帐号，使用admin登陆即可。登陆后，将管理员的密码改掉。

在Jenkins上安装插件

配置方法详见：<https://blog.csdn.net/you227/article/details/81076032>

清华大学插件地址：<https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json>

在可选插件中安装git parameter。

因为现在使用的是jenkins低版本，无法安装插件，插件需要更高的版本。所以下载新版本：<https://pkg.jenkins.io/redhat-stable/jenkins-2.164.3-1.1.noarch.rpm>

<http://mirrors.jenkins-ci.org/redhat/jenkins-2.177-1.1.noarch.rpm>

升级jenkins：

```
[root@node4 ~]# systemctl stop jenkins
[root@node4 ~]# yum update -y jenkins-2.177-1.1.noarch.rpm
[root@node4 ~]# systemctl start jenkins
```

jenkins2.177默认已集成插件git parameter。

如果是英文界面，可以在插件管理中，安装：Localization: Chinese (Simplified)

在jenkins上安装git

```
[root@node4 ~]# yum install -y git
```

项目流程

1. 程序员在自己的电脑上编写程序

```
[root@node3 ~]# git init myweb
[root@node3 myweb]# echo '<h1>My Web Site</h1>' > index.html
[root@node3 myweb]# git add .
[root@node3 myweb]# git commit -m "web 1.0"
[root@node3 myweb]# git tag 1.0 # 将当前状态标记为1.0版本
[root@node3 myweb]# echo '<h2>2nd version</h2>' >> index.html
[root@node3 myweb]# git add .
[root@node3 myweb]# git commit -m "web 2.0"
[root@node3 myweb]# git tag 2.0
```

2. 在gitlab上，在devops组中创建名为myweb的公开项目，新建的用户是该项目的主程序员

3. 程序员上传代码到gitlab

```
[root@node3 myweb]# git remote rename origin old-origin
[root@node3 myweb]# git remote add origin git@192.168.122.137:devops/myweb.git
[root@node3 myweb]# git push -u origin --all
[root@node3 myweb]# git push -u origin --tags
```

4. 在jenkins上创建工程，自动下载代码。创建一个自由风格的项目=>勾选This project is parameterized=>添加参数 git parameter=> name: webver, Default Value: origin/master=>源码管理：git=>Repository URL: <http://192.168.122.137/devops/myweb.git> => Branch Specifier (blank for 'any'): \$webver => 保存

5. 执行任务，jenkins将会下载myweb项目到/var/lib/jenkins/workspace目录

```
[root@node4 ~]# ls /var/lib/jenkins/workspace/  
myweb  
[root@node4 ~]# cat /var/lib/jenkins/workspace/myweb/index.html  
<h1>My Web Site</h1>
```

完善jenkins

- 1.jenkins下载web项目后，将其打包
2. 为了应用服务器可以下载项目软件包，在jenkins上安装httpd服务，以便应用服务器下载

```
[root@node4 ~]# yum install -y httpd  
[root@node4 ~]# systemctl start httpd  
[root@node4 ~]# systemctl enable httpd  
[root@node4 ~]# mkdir -p /var/www/html/deploy/pkgs  
# /var/www/html/deploy/: 保存livever、lastver，即当前版本和前一个版本的版本号  
# /var/www/html/deploy/pkgs/: 保存软件压缩包和它的md5值  
[root@node4 ~]# chown -R jenkins.jenkins /var/www/html/deploy/
```

- 3.jenkins服务器上有很多版本，需要标明最新（当前）版本和前一版本
- 4.jenkins服务器需要公布软件包的md5值，供文件完整性检查

编辑项目，选择下面的Additional Behaviour => 新增 Checkout to a sub-directory: myweb-\$webver

增加构建步骤=> Execute shell:

```
deploy_dir=/var/www/html/deploy  
pkgs_dir=/var/www/html/deploy/pkgs  
cp -r myweb-$webver $pkgs_dir # 将下载的软件目录拷贝到web服务器目录  
cd $pkgs_dir  
rm -rf myweb-$webver/.git # 删除版本库文件  
tar czf myweb-$webver.tar.gz myweb-$webver # 打包压缩  
md5sum myweb-$webver.tar.gz | awk '{print $1}' > myweb-$webver.tar.gz.md5 # 计算并保存md5值  
rm -rf myweb-$webver # 删除程序目录  
cd $deploy_dir  
[ -f livever ] && cat livever > lastver # 将当前版本内容写到前一版本文件  
echo $webver > livever # 更新当前版本
```

编写python工具，实现代码自动上线

1. 检查是否有新版本
2. 如果有新版本，下载
3. 检查下载的压缩包是否损坏
4. 如果没有损坏，部署

将物理主机作为应用服务器，对外提供服务：

/var/www/download/：用于存储下载的软件包

/var/www/deploy/：用于存储当前版本文件和解压后的软件包

/var/www/html/nsd1812：指向软件目录的链接

