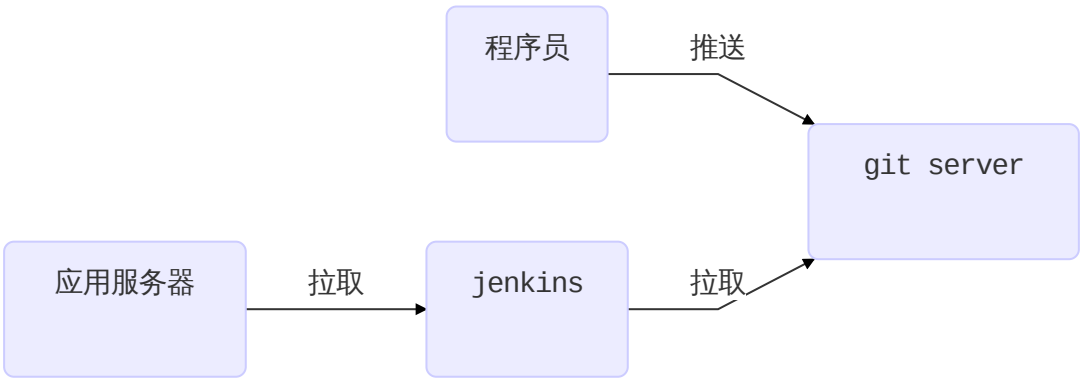
nsd1903_devops_day05



Jenkins：实现CI

安装

```
[root@node7 ~]# rpm -ihv jenkins-2.177-1.1.noarch.rpm
[root@node7 ~]# systemctl start jenkins
[root@node7 ~]# systemctl enable jenkins
# 注意，jenkins是java编写的，运行需要java环境
```

配置

需要联网。否则不能安装插件。联网，也可以不安装插件

访问http://x.x.x.x:8080。在安装插件页面选择自定义，不安装任何插件（因为默认安装插件时访问的是国外站点，速度慢）。创建管理员帐号时，不创建，直接使用admin登陆。

安装插件

1. 更新插件源

首页 -> Manage Jenkins -> Manage Plugins -> Advanced -> Update site : https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json -> submit

2. Available选项卡 -> Localization: Chinese (Simplified) / Git Parameter -> 点击Install Without Restart -> 勾选 Restart Jenkins when installation is complete and no jobs are running

建立工程

首页 -> 新建Item -> 名称：mytest / 风格：FreeStyle -> This project is parameterized: git parameter / name: mytag / parameter type: Branch or Tag / Default Value: origin/master -> 源码管理: git / http://192.168.4.6/devops/myweb.git / Branches to build: $mytag -> 保存 ->Build with Parameters -> 选择一个tag后点开始构建。

构建出来的内容： /var/lib/jenkins/workspace/下

完整过程：

1. 程序员编写代码

```
[root@node5 ~]# git init myblog
[root@node5 ~]# cd myblog/
[root@node5 myblog]# echo '<h1>My Blog</h1>' > index.html
[root@node5 myblog]# git add .
[root@node5 myblog]# git commit -m "blog 1.0"
[root@node5 myblog]# git tag 1.0
[root@node5 myblog]# echo 'blog 2.0' >> index.html
[root@node5 myblog]# git add .
[root@node5 myblog]# git commit -m "blog 2.0"
[root@node5 myblog]# git tag 2.0
```

2. 在gitlab上创建项目，名为myblog，允许用户可以上传
3. 上传代码到gitlab

```
[root@node5 myblog]# git remote rename origin old-origin
[root@node5 myblog]# git remote add origin git@192.168.4.6:devops/myblog.git
[root@node5 myblog]# git push -u origin --all
[root@node5 myblog]# git push -u origin --tags
```

4. jenkins下载代码
5. 修改下载代码到子目录。配置->Additional Behaviours: 新增 checkout to a sub-directory: myblog-${blogver} -> 保存 ->Build with Parameters
6. 修改jenkins工程，实现以下目标
    - 将下地的版本打包，放到apache目录下

```
/var/www/html/deploy/live_ver: 最新版本
/var/www/html/deploy/last_ver: 前一版本
/var/www/html/deploy/pkgs/:保存打包的软件和相应的md5值
```

```
[root@node7 ~]# yum install -y httpd
[root@node7 ~]# systemctl start httpd
[root@node7 ~]# mkdir -p /var/www/html/deploy/pkgs
[root@node7 ~]# chown -R jenkins.jenkins /var/www/html/deploy/
```

配置 -> 构建 -> 增加构建步骤 -> execute shell:

```
pkg_dir=/var/www/html/deploy/pkgs
cp -r myblog-${blogver} $pkg_dir
cd $pkg_dir
rm -rf myblog-${blogver}/.git
tar czf myblog-${blogver}.tar.gz myblog-${blogver}
rm -rf myblog-${blogver}
md5sum myblog-${blogver}.tar.gz |  awk '{print $1}' > myblog-${blogver}.tar.gz.md5
cd ..
[ -f live_ver ] && cat live_ver > last_ver
echo ${blogver} > live_ver
```

自动部署

- 目录规划

  ```
  /var/www/download：保存下载的压缩包
  /var/www/deploy: 保存解压后的目录和live_ver
  /var/www/html/nsd1903:指向当前生效的网站的快捷方式
  ```

- 编写程序，实现自动部署
  - 检查是否有新版本
  - 有新版本，则下载
  - 校验下载的软件包是否损坏
  - 如果没有损坏则部署
  - 本地生成当前应用的版本