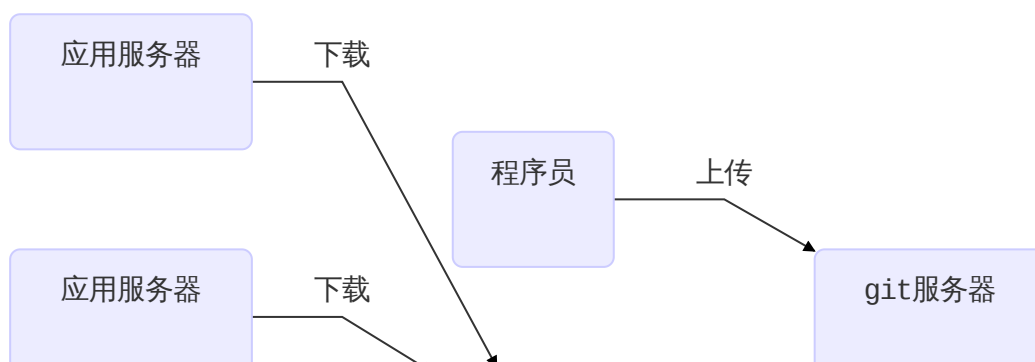


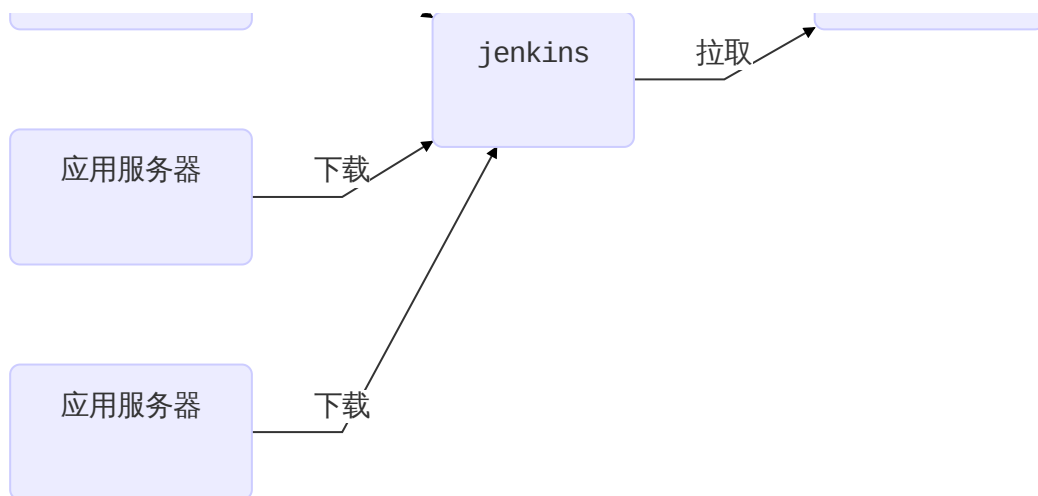
nsd_1908_devops_day04

CI/CD：持续集成/持续交付

编程语言：

- 解释执行：shell / python / php / javascript
- 编译执行：c / c++ / go / java





git：开源的软件版本控制管理工具

准备git

将node4作为程序员编写代码的主机

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

# 命令补全功能需要安装的包
[root@node4 ~]# rpm -q bash-completion
bash-completion-2.1-6.el7.noarch

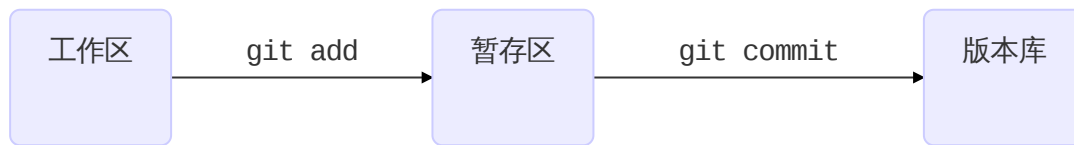
# 启用git的tab补全功能，可以重启系统，也可以采用以下方法
[root@node4 ~]# source /etc/bash_completion.d/git

# 配置用户信息
[root@node4 ~]# git config --global user.name zzg
[root@node4 ~]# git config --global user.email zzg@tedu.cn
[root@node4 ~]# git config --global core.editor vim
[root@node4 ~]# git config --list
user.name=zzg
user.email=zzg@tedu.cn
core.editor=vim
[root@node4 ~]# cat ~/.gitconfig
[user]
    name = zzg
    email = zzg@tedu.cn
[core]
    editor = vim
```

git的工作区域

- 工作区：程序项目目录

- 暂存区：工作区与版本库之间的缓冲地带，.git/index
- 版本库：程序项目目录下的.git目录



git应用

初始化git方法一

```
[root@node4 ~]# git init mytest
初始化空的 Git 版本库于 /root/mytest/.git/
[root@node4 ~]# ls -A mytest/
.git
```

初始化git方法二

```
[root@node4 ~]# mkdir myweb
[root@node4 ~]# cd myweb/
[root@node4 myweb]# echo '<h1>my web site</h1>' > index.html
[root@node4 myweb]# git init
初始化空的 Git 版本库于 /root/myweb/.git/
[root@node4 myweb]# ls -A
.git index.html
```

查看状态

```
[root@node4 myweb]# git status
[root@node4 myweb]# git status -s
?? index.html  # ??表示状态未知
```

将项目目录下所有内容加入到暂存区

```
[root@node4 myweb]# git add .
[root@node4 myweb]# git status
[root@node4 myweb]# git status -s
A index.html  # A表示新增加的文件
```

提交到版本库

```
[root@node4 myweb]# git commit # 跳出vim写日志，如果什么也不写，则不提交
[root@node4 myweb]# git commit -m "init project"
[root@node4 myweb]# git status
```

位于分支 master

```
无文件要提交，干净的工作区
[root@node4 myweb]# git status -s
```

将指定的文件加入跟踪

```
[root@node4 myweb]# cp /etc/hosts .
[root@node4 myweb]# cp /etc/passwd .
```

```

[root@node4 myweb]# ls
hosts index.html passwd
[root@node4 myweb]# git add hosts
[root@node4 myweb]# git commit -m "add hosts"

# 设置不需要通过git管理的文件
[root@node4 myweb]# vim .gitignore
*.swp
passwd
.gitignore
[root@node4 myweb]# git status
# 位于分支 master
无文件要提交，干净的工作区

# 将文件从暂存区中撤出
[root@node4 myweb]# cp /etc/issue .
[root@node4 myweb]# git add .
[root@node4 myweb]# git status
[root@node4 myweb]# git reset HEAD issue

# 恢复误删除的文件
[root@room8pc16 tmp]# du -sh nsd2019/
215M    nsd2019/
[root@room8pc16 tmp]# cd nsd2019/
[root@room8pc16 nsd2019]# rm -rf *
[root@room8pc16 nsd2019]# du -sh .
75M    .
[root@room8pc16 nsd2019]# ls -A
.git .gitignore
[root@room8pc16 nsd2019]# git status | more
[root@room8pc16 nsd2019]# git checkout -- *
[root@room8pc16 nsd2019]# ls
ansible_project  nsd1902  nsd1905  nsd1908  review
ebooks           nsd1903  nsd1906  ppts     software
nsd1812          nsd1904  nsd1907  README.md

# 删除文件
[root@node4 myweb]# git add .
[root@node4 myweb]# git commit 'add issue'
[root@node4 myweb]# ls
hosts index.html issue passwd
[root@node4 myweb]# git rm issue
[root@node4 myweb]# git status
[root@node4 myweb]# git commit -m "delete issue"
[root@node4 myweb]# git status

# 改名
[root@node4 myweb]# git mv hosts hosts.txt
[root@node4 myweb]# git status -s
R  hosts -> hosts.txt
[root@node4 myweb]# git commit -m "mv hosts hosts.txt"
[root@node4 myweb]# git status -s

```

```

# 切换到某一个commit点
[root@node4 myweb]# git log      # 找到add issue的确认点，复制其id号
[root@node4 myweb]# git status  # 工作区需要是干净的
# 位于分支 master
无文件要提交，干净的工作区
[root@node4 myweb]# git checkout \
2b60da12002b0fad328f4071a3fb7f49299ea5ad
[root@node4 myweb]# ls
hosts  index.html  issue  passwd

# 切换回最新状态
[root@node4 myweb]# git checkout master
[root@node4 myweb]# ls
hosts.txt  index.html  passwd

```

分支管理

- 默认情况下，git有一个名为master的分支
- 用户也可以创建自己的分支

```

# 查看分支
[root@node4 myweb]# git branch
* master

# 新建分支b1
[root@node4 myweb]# git branch b1
[root@node4 myweb]# git branch
b1
* master    # 当前分支

# 在master分支上提交代码
[root@node4 myweb]# cp /etc/motd .
[root@node4 myweb]# git add .
[root@node4 myweb]# git commit -m "add motd"
[root@node4 myweb]# ls
hosts.txt  index.html  motd  passwd

# 切换分支
[root@node4 myweb]# git checkout b1
切换到分支 'b1'
[root@node4 myweb]# ls  # b1分支的工作区没有motd文件
hosts.txt  index.html  passwd

# 在b1上提交代码
[root@node4 myweb]# cp /etc/security/access.conf .
[root@node4 myweb]# git add .
[root@node4 myweb]# git commit -m "add access.conf"
[root@node4 myweb]# ls
access.conf  hosts.txt  index.html  passwd

# 将b1汇入主干

```

```
[root@node4 myweb]# git checkout master
[root@node4 myweb]# ls
hosts.txt  index.html  motd  passwd
[root@node4 myweb]# git merge b1 -m "merge b1"
[root@node4 myweb]# ls
access.conf  hosts.txt  index.html  motd  passwd

# 删除分支
[root@node4 myweb]# git branch -d b1
[root@node4 myweb]# git branch
* master
```

tag标记

```
# 查看标记
[root@node4 myweb]# git tag

# 为当前提交打标记
[root@node4 myweb]# git tag 1.0
[root@node4 myweb]# git tag
1.0

# 继续编代码，提交。为新提交打标记
[root@node4 myweb]# echo '<h2>2nd version</h2>' >> index.html
[root@node4 myweb]# git add .
[root@node4 myweb]# git commit -m "modify index.html"
[root@node4 myweb]# git tag 1.1
[root@node4 myweb]# git tag
1.0
1.1
```

gitlab服务器

- 准备一台内存至少4GB以上的虚拟机
- 虚拟机上安装docker，将镜像导入

```
# ls /linux-soft/05/gitlab_zh.tar
[root@node5 images]# docker load -i gitlab_zh.tar

# 将宿主机ssh更改端口
[root@node5 ~]# vim /etc/ssh/sshd_config
Port 2022
[root@node5 ~]# systemctl restart sshd
[root@node5 ~]# exit
[root@room8pc16 phase5]# ssh -p2022 node5

# 启动容器
[root@node5 ~]# docker run -d -h gitlab --name gitlab -p 443:443 -p 80:80 -p 22:22 --
restart always -v /srv/gitlab/config:/etc/gitlab -v /srv/gitlab/logs:/var/log/gitlab -v
/srv/gitlab/data:/var/opt/gitlab gitlab_zh:latest
```

```
# 查看状态，当显示healthy时，容器才可用
[root@node5 ~]# docker ps
```

gitlab初始化：<http://192.168.4.5>。首次访问时，需要为root用户设置密码，密码要求长一些、复杂一些。

gitlab中重要的三个概念

- 项目：project，对应程序员编写的软件项目
- 群组：group，对应的是开发团队
- 用户：对应使用gitlab的用户

创建用户：创建用户时不能设置用户的密码，但是编辑用户时可以。

创建组：再把创建的用户回入到组中，成为组的主程序员

创建项目：创建名为myweb的项目，为devops组创建，类型是公开的

程序员将代码推送到gitlab服务器

```
[root@node4 ~]# cd myweb/
[root@node4 myweb]# git remote add origin \
http://192.168.4.5/devops/myweb.git
[root@node4 myweb]# git push -u origin --all
Username for 'http://192.168.4.5': zzg
Password for 'http://zzg@192.168.4.5':
[root@node4 myweb]# git push -u origin --tags
Username for 'http://192.168.4.5': zzg
Password for 'http://zzg@192.168.4.5':

# 创建README
[root@node4 myweb]# vim README.md
# my web site
## 使用方法
- 搭建web服务器
- 启动服务
- 将my web site拷贝到web服务器
...

[root@node5 ~]# yum install -y httpd
[root@node5 ~]# systemctl start httpd
[root@node5 ~]# systemctl enable httpd
...

[root@node4 myweb]# vim README.md
[root@node4 myweb]# git add .
[root@node4 myweb]# git commit -m "add README.md"
[root@node4 myweb]# git push
```

实现ssh免密推送代码

```
# 生成密钥
[root@node4 myweb]# ssh-keygen -C "zzg@tedu.cn" -b 4096
[root@node4 myweb]# cat ~/.ssh/id_rsa.pub
```

将查看到的公钥内容拷贝到gitlab用户ssh密钥窗格

修改本地git配置

```
[root@node4 myweb]# git remote remove origin # 删除远程仓库的关联
```

```
[root@node4 myweb]# git remote add origin \  
git@192.168.4.5:devops/myweb.git # 重新关联到ssh
```

上传测试

```
[root@node4 myweb]# echo 'new line' >> index.html
```

```
[root@node4 myweb]# git add .
```

```
[root@node4 myweb]# git commit -m "add line to index.html"
```

```
[root@node4 myweb]# git push
```