上午课程核心内容

1. Shell编程-常见命令

awk：文档编辑工作，可以用来提取文本的某些行的某些列的信息，并可以进行一些简单的统计。

awk 中有一个数组功能，类似于 python 中的字典。

2. Shell编程-控制语句

if 语句：

```
#! /bin/bash

# 单分支if
if [ "$1" == "man" ]
then
    echo "你的性别是：男"
fi

# 双分支if
if [ "$1" == "man" ]
then
    echo "您的性别是：男"
else
    echo "您的性别是：女"
fi

# 多分支if
if [ "$1" == "man" ]
then
    echo "您的性别是：男"
elif [ "$1" == "woman" ]
then
    echo "您的性别是：女"
else
    echo "您的性别，我不知道"
fi
```

case 语句：

```
#! /bin/bash

# 多if语句的使用场景
if [ "$1" == "start" ]
then
    echo "服务启动中..."
elif [ "$1" == "stop" ]
then
    echo "服务关闭中..."
elif [ "$1" == "restart" ]
then
    echo "服务重启中..."
else
    echo "$0 脚本的使用方式：$0 [ start | stop | restart ]"
fi
```

循环语句：

```
#! /bin/bash

# for循环
for i in $(ls /root)
do
    echo $i
done
```

```
#! /bin/bash
# while 的示例
a=1

while [ "${a}" -lt 5 ]
do
    echo "${a}"
    a=$((a+1))
done
```

```
#! /bin/bash
# until的示例
a=1

until [ "${a}" -eq 5 ]
do
    echo "${a}"
    a=$((a+1))
done
```

3. Shell编程-函数

```
#! /bin/bash


# 定义 shell 函数
func1() {
    echo 'my name is smart'
}

# 调用函数，不传参
# func1


func2() {
    echo "my name is $1"
}

# 调用函数，传参
# func2 'xiaohui'

# 区分函数体之外的 $n 和函数体之内的 $n 的区别
# 函数体之外的 $n: 表示获取执行脚本时，给脚本传递的第 n 个参数
# 函数体之内的 $n: 表示获取调用函数时，给函数传递的第 n 个参数

func3() {
    echo "my name is $1"
}

func3 "$2"
~
```

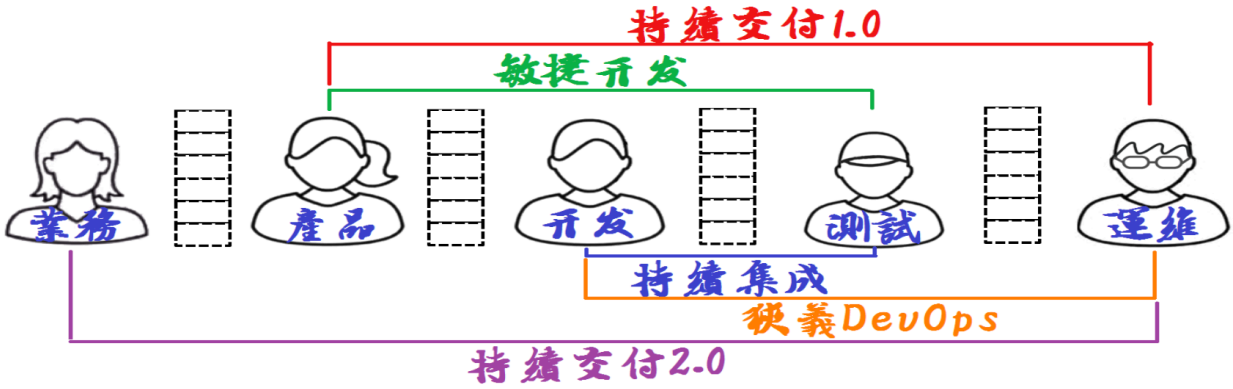
3. Jenkins-基础知识

项目开发生命周期:

一个项目的生命周期一般会分为如下几个阶段:

阶段	说明
调研阶段	产品经理或业务人员和客户进行对接，就相关业务问题进行沟通，收集一些必要的数据
设计阶段	根据调研的结果，业务、产品和开发共同确定产品的具体需求，最终产生需求分析说明文档，并由产品经理设计产品原型图
开发阶段	根据需求分析的结果，由研发部门(前端vs后端)的研发人员进行产品功能的具体实现
测试阶段	对于研发产生的结果，由测试部门的测试人员来验证产品功能的正确性，保障产品质量
运营阶段	对于经过测试，功能稳定的产品由运维人员负责上线部署，并进入持续的线上运营阶段

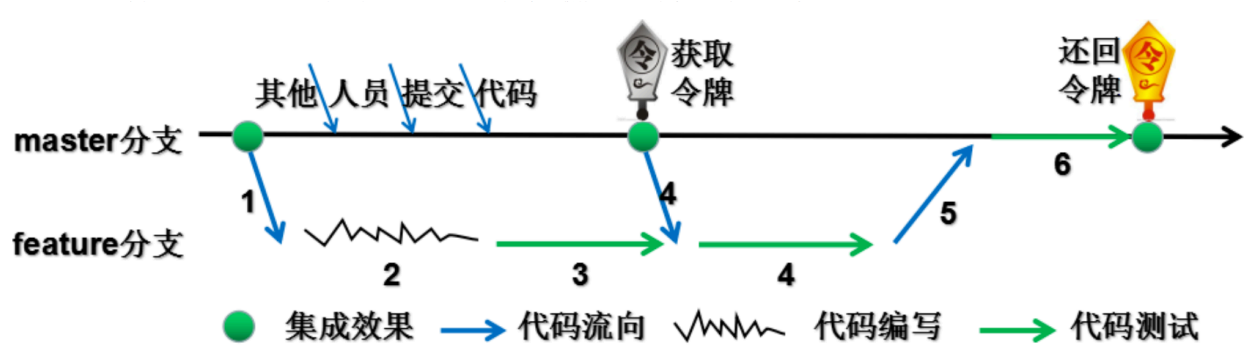
开发过程中的术语:



工作中涉及到的各种开发环境：

1）个人开发环境：你个人的电脑环境
2）公司开发环境：公司一台内部服务器，用来集成每个开发人员开发代码
3）测试环境：公司一台内部服务器，给测试团队进行使用的，开发团队开发项目代码需要运行到测试环境中，测试团队在测试环境进行测试工作
4）预发布环境(Staging环境)：公司线上服务器的其中一台，用来在项目实际上线运营之前，先把经过测试项目运行到预发布环境，进行最后一道的功能验证：支付、安全测试、压力测试...
5）线上环境(生产环境)：公司线上服务器(一台或多台)，实现项目上线运行的服务器

代码持续集成 6 步提交法：



开发集成体系：

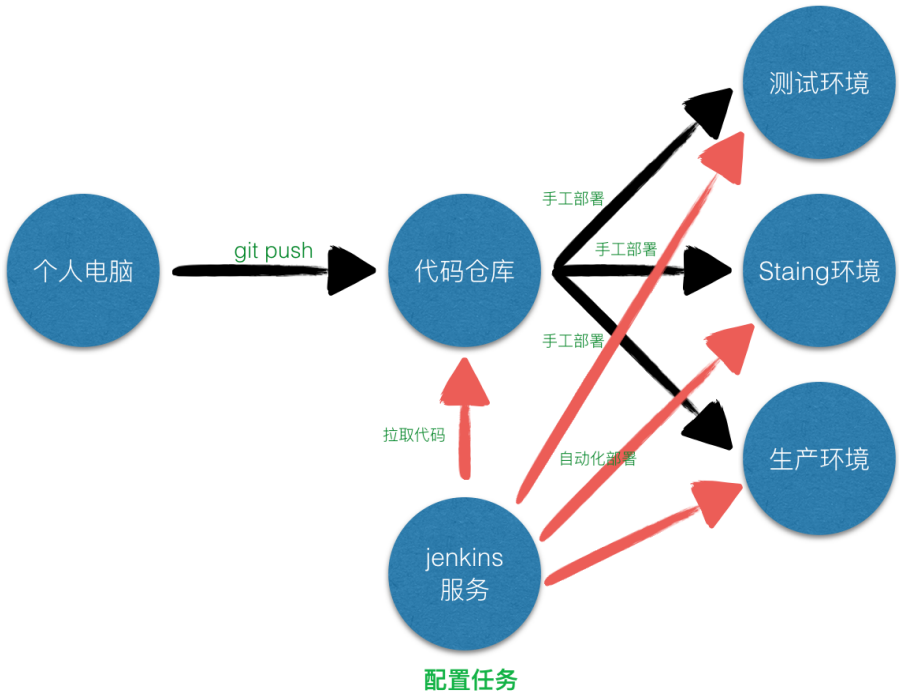
CI：持续集成

CD：持续交付

4. Jenkins-软件简介和安装

软件简介：

Jenkins：一个开源的软件平台，用来在项目开发的过程中，实现持续集成、自动化测试、持续交付等自动化的过程，目标是解放劳动力，让人把精力放在机器不擅长的场景上面。



Jenkins安装：

1）安装 java 环境

2）安装 jenkins 软件

注意：安装之前先把网断了，让其走离线安装过程。

Jenkins配置：

1）全局配置

2）插件配置：在线安装和离线安装

下午课程核心内容

1. Jenkins-认证配置

1）ssh 认证配置

注：

jenkins 服务上配置的是 jenkins 主机的私钥.

对应代码仓库上配置的是 jenkins 主机的公钥.

```
# 生成对应主机的公钥和私钥
ssh-keygen -t rsa -C "备注信息"
```

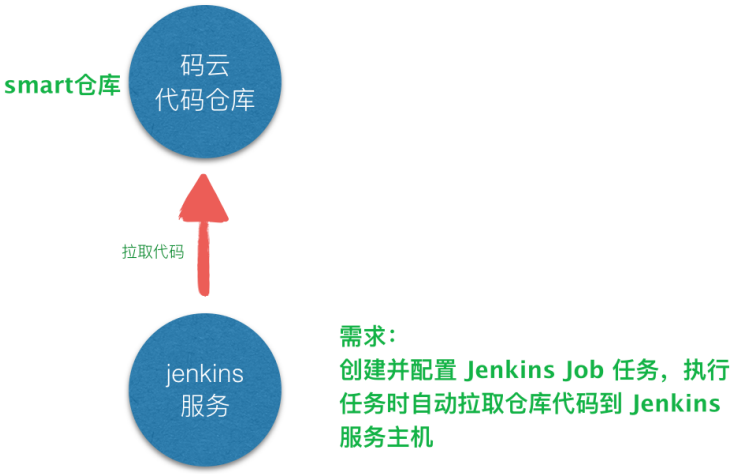
2）用户名和密码认证配置

jenkins 服务上配置的对应代码仓库的账户名和密码.

2. Jenkins-job 任务创建和执行

job 任务需求：

从码云代码仓库自动拉取代码到 jenkins 服务主机。



job 任务创建：

1）选择 "New Item"

2）设置 job 名称，选择 "free style" 任务类型

3）进行任务设置

job 任务执行：点击 build now 执行对应 job 任务

job 相关目录：workspace 和 jobs

- workspace：下方有一个任务同名目录，是 job 任务的工作目录，拉取的代码在下方
- jobs：下方有一个任务同名目录，存放对应 job 的任务配置和每一次构建执行记录

Jenkins freestyle 任务 6 部分配置：

1）全局配置(General)：项目基本配置，项目名字描述、参数、禁用项目、并发构建、限制构建默认node等

2）源代码管理(Source code managemet)：源代码仓库配置信息，支持Git、Subversion等

3）触发构建(Build Triggers)：构建触发方式，包含周期性构建、poll scm、远程脚本触发构建、其他项目构建结束后触发等

4）构建环境(Build Environment)：构建环境相关设置，构建前删除工作空间、输出信息添加时间戳、设置构建名称、插入环境变量等

5）执行构建(Build)：设置项目构建任务的执行方式、种类等，具体的执行信息

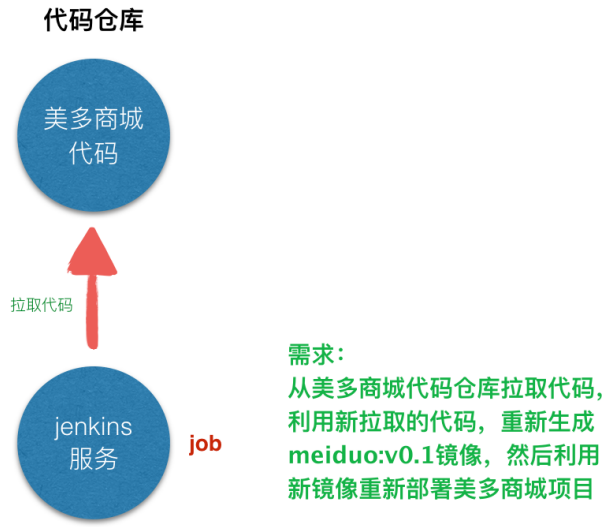
6）构建后行为(Post-build Actions)：Artifact归档、邮件通知、发布单元测试报告、触发下游项目等

3. Jenkins - 实践案例

3.1 案例1：jenkins job 拉取美多商城代码并实现自动化部署

jobs 任务需求：

jenkins job 拉取美多商城代码并实现自动化部署。



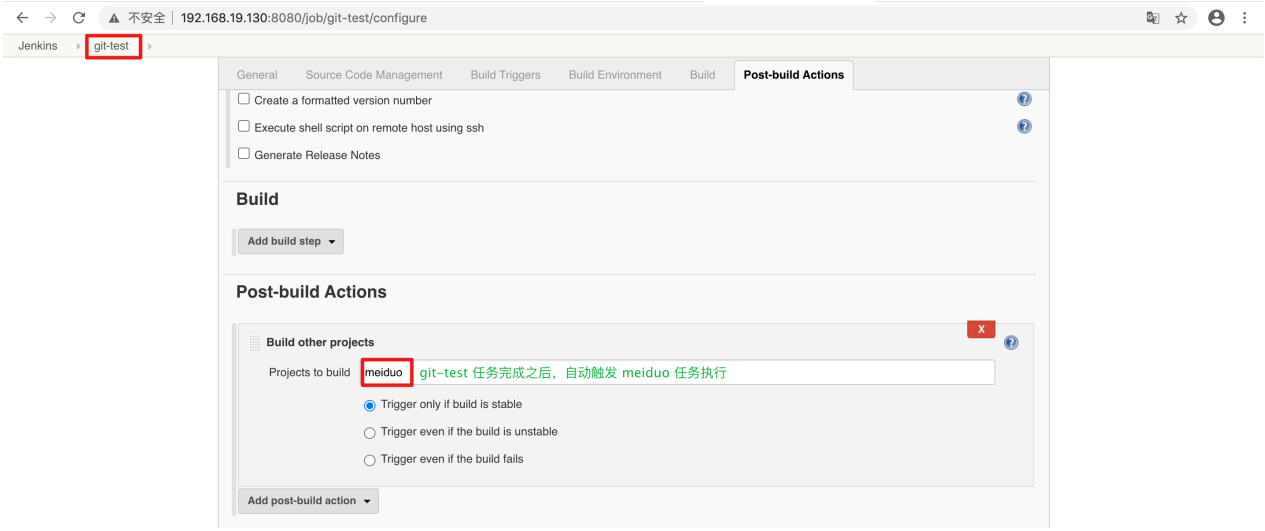
3.2 jenkins job 的触发方式介绍

触发构建的含义主要有两个：触发当前任务执行，触发其他任务执行.

方式	说明
触发当前任务	① Trigger builds remotely (e.g., from scripts)：远程触发构建 ② Build periodically：定期构建 ③ Poll SCM：轮训检查源代码变动后构建 ④ 在任务的"Build Triggers"里面有一项"Build after other projects are built"：其他任务执行完毕后，再执行当前任务 ⑤ Hook触发：git|Gitlab|GitHub hooks
触发其他任务执行	① 在任务的"Post-build Actions"里有一项"Build other projects"：当前任务执行完毕后，再执行其他任务

3.3 案例2：git-test 任务完成之后触发 meiduo 任务的执行

参考讲义



3.4 案例3：参数化构建

实际代码仓库中的项目代码，会有多个不同的分支，每个分支上的代码会有差异，通过 jenkins job 的参数化构建配置，可以实现选择哪个分支，拉取哪个分支代码并执行操作的任务。