# 安装maven

192.168.10.236测试机上安装maven、git ,拉取Shop-Cloud项目代码，进行手动构建，研究maven、jenkins之间的运行是如何进行的。

一、安装jdk环境

使用yum源，这里需要注意yum源的配置

1、查看yum库中都有哪些jdk版本

[root@lyh:] # yum search java|grep jdk

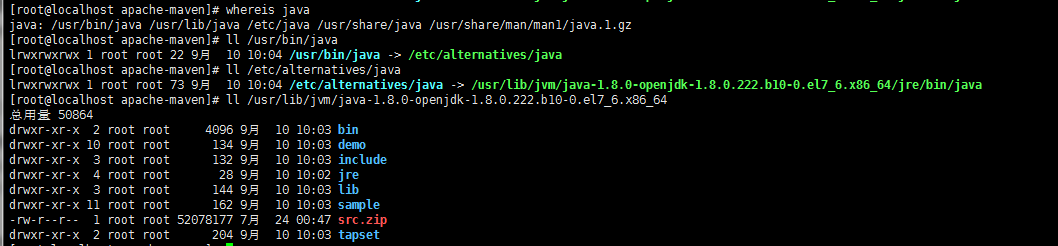
2、选择指定的版本安装，注意最后的 \* 以及yum源安装的是openjdk，注意openjdk的区别。

[root@lyh:] # yum install java-1.8.0-openjdk\*

3、安装完成后查看版本信息

[root@lyh:] # java -version

4、查找JAVA\_HOME



5、添加环境变量

[root@localhost apache-maven]# vim /etc/profile

末尾添加：export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.222.b10-0.el7\_6.x86\_64

[root@localhost apache-maven]# source /etc/profile

[root@localhost apache-maven]# echo $JAVA\_HOME

/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.222.b10-0.el7\_6.x86\_64

1. 安装maven

[root@localhost Shop-Cloud]# wget <http://mirrors.hust.edu.cn/apache/maven/maven-3/3.6.0/binaries/apache-maven-3.6.0-bin.tar.gz>

#下载的版本视情况而定

[root@localhost Shop-Cloud]# tar -xf apache-maven-3.6.0-bin.tar.gz

[root@localhost Shop-Cloud]# mv apache-maven-3.6.0 /usr/local/apache-maven

[root@localhost Shop-Cloud]# mvn -version

[root@localhost Shop-Cloud]# vim /etc/profile

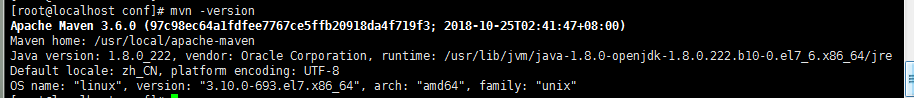
添加如下信息：

export MAVEN\_HOME=/usr/local/apache-maven

export PATH=$JAVA\_HOME/bin:$MAVEN\_HOME/bin:$PATH

[root@localhost Shop-Cloud]# source /etc/profile

[root@localhost Shop-Cloud]# mvn -version



[root@localhost conf]# ls /root/.m2/repository/ #本地仓库位置

1. 安装git

[root@localhost Shop-Cloud]# yum -y install git

[root@localhost Shop-Cloud]# git --version

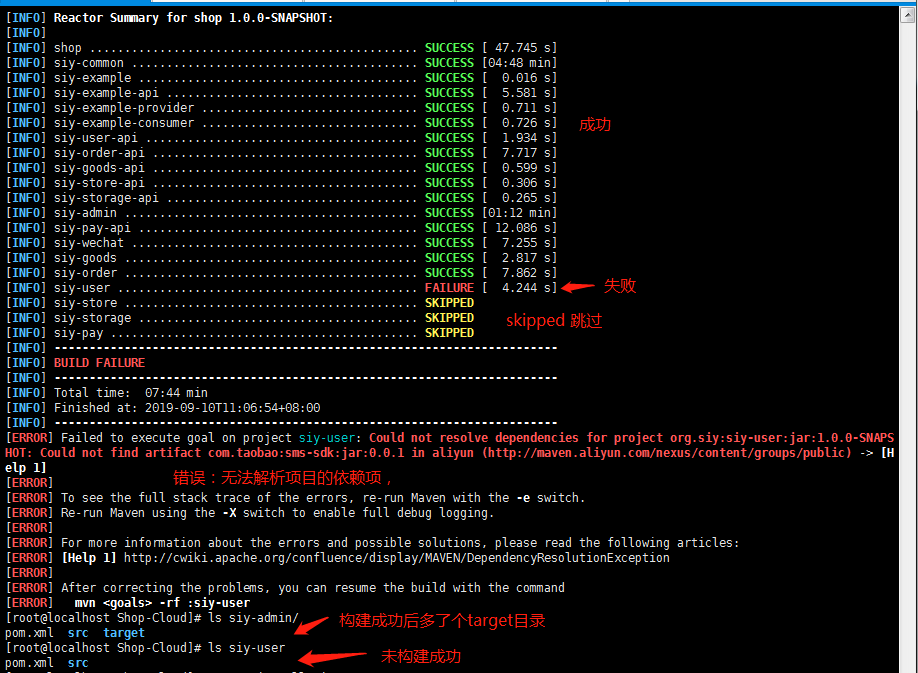
[root@localhost Shop-Cloud]# cd /home/

[root@localhost Shop-Cloud]# git clone http://192.168.10.245/SiY/Shop/Shop-Cloud.git

[root@localhost Shop-Cloud]# cd Shop-Cloud/

1. 构建

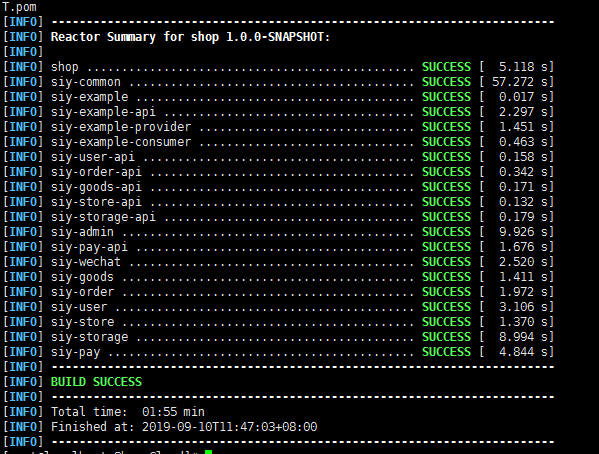
[root@localhost Shop-Cloud]# mvn install -Dmaven.test.skip=true #-Dmaven.test.skip=true跳过测试



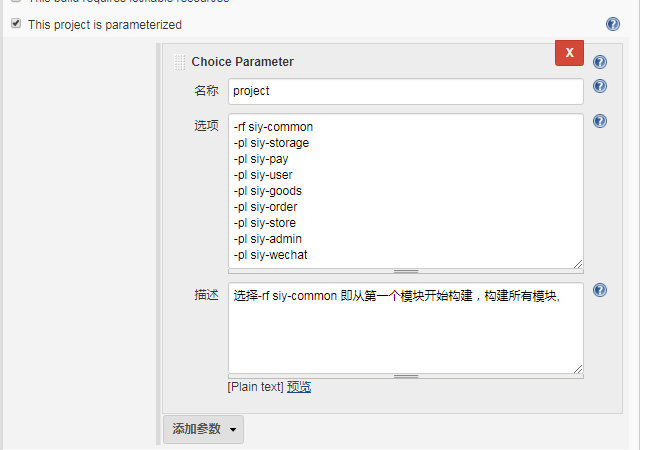
解决办法：

将在用的192.168.10.242:/etc/maven/setting.xml 文件放到本测试机192.168.10.236:/usr/local/apache-maven/conf/

后再进入项目目录中重新执行mvn install -Dmaven.test.skip=true 进行构建即可成功，



# jenkins测试环境中选择性构建案例：



-rf siy-common

-pl siy-storage

-pl siy-pay

-pl siy-user

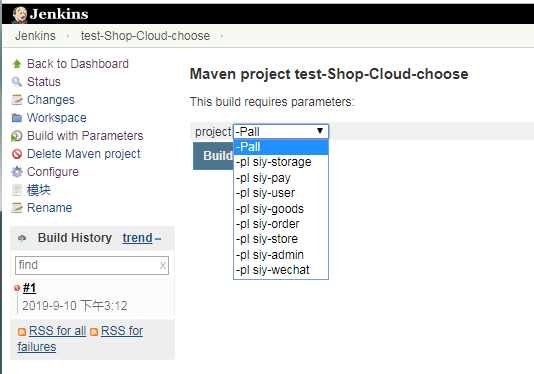
-pl siy-goods

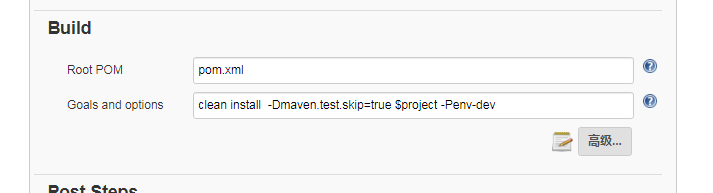
-pl siy-order

-pl siy-store

-pl siy-admin

-pl siy-wechat





clean install -Dmaven.test.skip=true $project -Penv-dev # $project传的上面传的参数（-Pall -pl siy-storage 等等）

实际执行的是：mvn clean install -Dmaven.test.skip=true $project

如果maven.test.skip不存在pom.xml，它将被设置。

如果maven.test.skip已经存在pom.xml，其值将被作为参数传递的值覆盖-D。

要发送多个变量，请使用多个空格分隔符加-D：

mvn -DpropA=valueA -DpropB=valueB -DpropC=valueC clean package

构建完之后执行shell

if [ "$project" == "-Pall" ];then

echo 1111

cp /home/jenkins/workspace/test-Shop-Cloud/siy-storage/target/siy-storage-1.0.0-SNAPSHOT.jar /home/siy/shop-cloud/siy-storage/

cp /home/jenkins/workspace/test-Shop-Cloud/siy-pay/target/siy-pay-1.0.0-SNAPSHOT.jar /home/siy/shop-cloud/siy-pay/

cp /home/jenkins/workspace/test-Shop-Cloud/siy-user/target/siy-user-1.0.0-SNAPSHOT.jar /home/siy/shop-cloud/siy-user/

cp /home/jenkins/workspace/test-Shop-Cloud/siy-goods/target/siy-goods-1.0.0-SNAPSHOT.jar /home/siy/shop-cloud/siy-goods/

cp /home/jenkins/workspace/test-Shop-Cloud/siy-order/target/siy-order-1.0.0-SNAPSHOT.jar /home/siy/shop-cloud/siy-order/

cp /home/jenkins/workspace/test-Shop-Cloud/siy-store/target/siy-store-1.0.0-SNAPSHOT.jar /home/siy/shop-cloud/siy-store/

cp /home/jenkins/workspace/test-Shop-Cloud/siy-admin/target/siy-admin-1.0.0-SNAPSHOT.jar /home/siy/shop-cloud/siy-admin/

cp /home/jenkins/workspace/test-Shop-Cloud/siy-wechat/target/siy-wechat-1.0.0-SNAPSHOT.jar /home/siy/shop-cloud/siy-wechat/

BUILD\_ID=DONTKILLME

sh /home/siy/shop-cloud-restart-all.sh

else

echo 2222

#echo $project

newname=${project:4}

echo $newname

cp /home/jenkins/workspace/test-Shop-Cloud/$newname/target/$newname-1.0.0-SNAPSHOT.jar /home/siy/shop-cloud/$newname/

cd /home/siy/shop-cloud/$newname/

BUILD\_ID=DONTKILLME

./start.sh restart

./start.sh status

fi

pscp -i F:\Users\Administrator\240privatekey.ppk E:\jenkins\workspace\test-Shop-Cloud\siy-storage\target\siy-storage-1.0.0-SNAPSHOT.jar root@192.168.10.242:/home/siy/shop-cloud/backup

pscp -i F:\Users\Administrator\240privatekey.ppk E:\jenkins\workspace\test-Shop-Cloud\siy-pay\target\siy-pay-1.0.0-SNAPSHOT.jar root@192.168.10.242:/home/siy/shop-cloud/backup

pscp -i F:\Users\Administrator\240privatekey.ppk E:\jenkins\workspace\test-Shop-Cloud\siy-user\target\siy-user-1.0.0-SNAPSHOT.jar root@192.168.10.242:/home/siy/shop-cloud/backup

pscp -i F:\Users\Administrator\240privatekey.ppk E:\jenkins\workspace\test-Shop-Cloud\siy-goods\target\siy-goods-1.0.0-SNAPSHOT.jar root@192.168.10.242:/home/siy/shop-cloud/backup

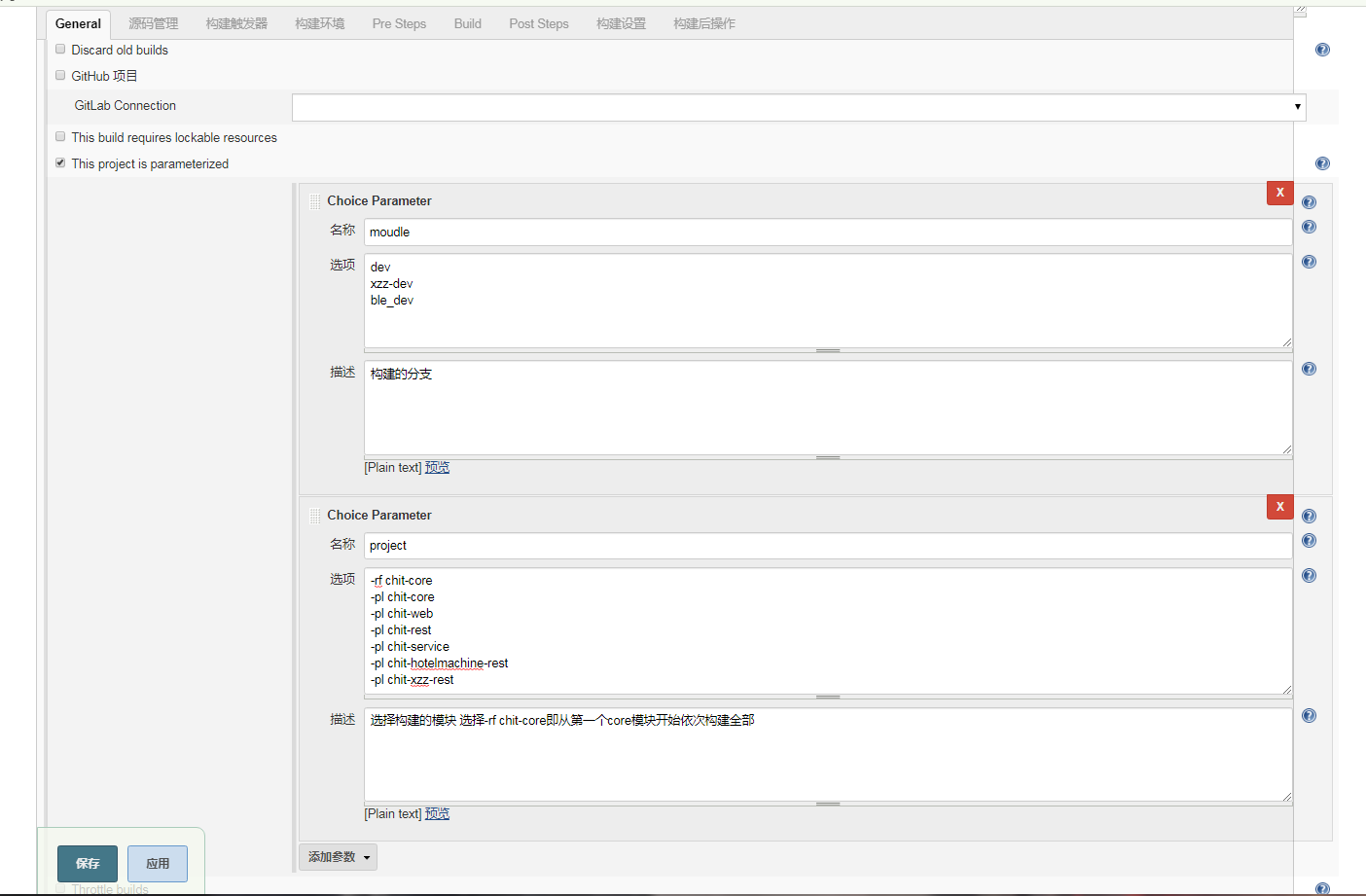
pscp -i F:\Users\Administrator\240privatekey.ppk E:\jenkins\workspace\test-Shop-Cloud\siy-order\target\siy-order-1.0.0-SNAPSHOT.jar root@192.168.10.242:/home/siy/shop-cloud/backup

pscp -i F:\Users\Administrator\240privatekey.ppk E:\jenkins\workspace\test-Shop-Cloud\siy-store\target\siy-store-1.0.0-SNAPSHOT.jar root@192.168.10.242:/home/siy/shop-cloud/backup

pscp -i F:\Users\Administrator\240privatekey.ppk E:\jenkins\workspace\test-Shop-Cloud\siy-admin\target\siy-admin-1.0.0-SNAPSHOT.jar root@192.168.10.242:/home/siy/shop-cloud/backup

pscp -i F:\Users\Administrator\240privatekey.ppk E:\jenkins\workspace\test-Shop-Cloud\siy-wechat\target\siy-wechat-1.0.0-SNAPSHOT.jar root@192.168.10.242:/home/siy/shop-cloud/backup

test-ylgx



project

-rf chit-core

-pl chit-core

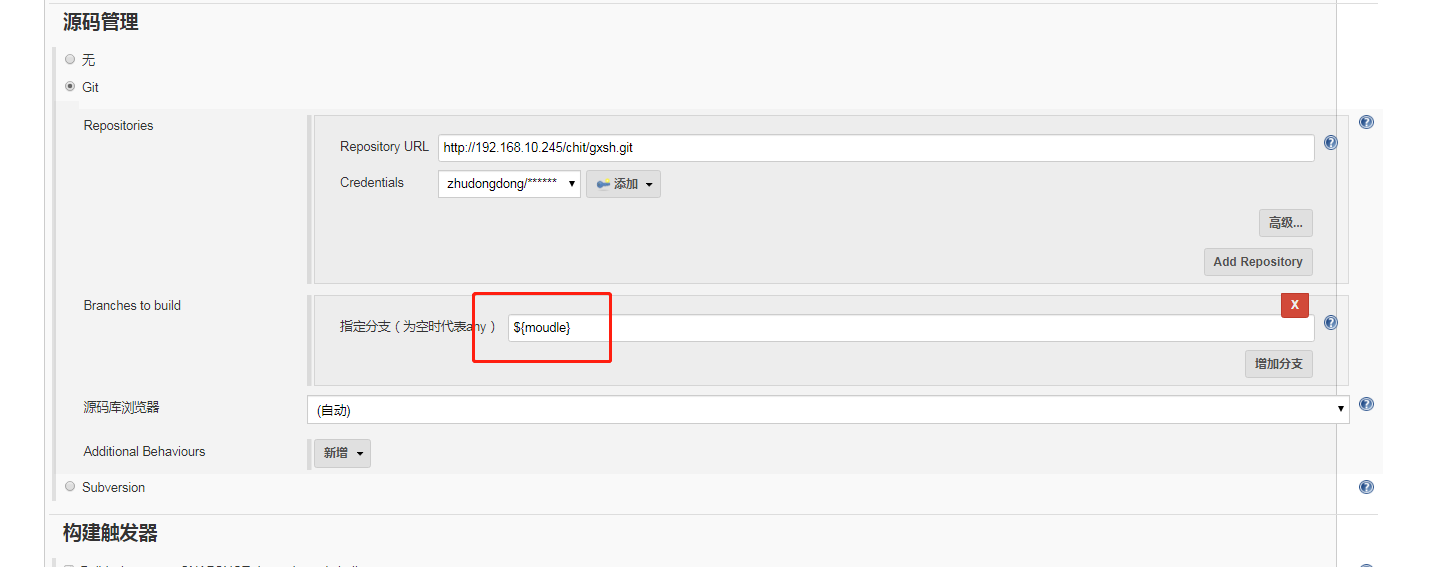
-pl chit-web

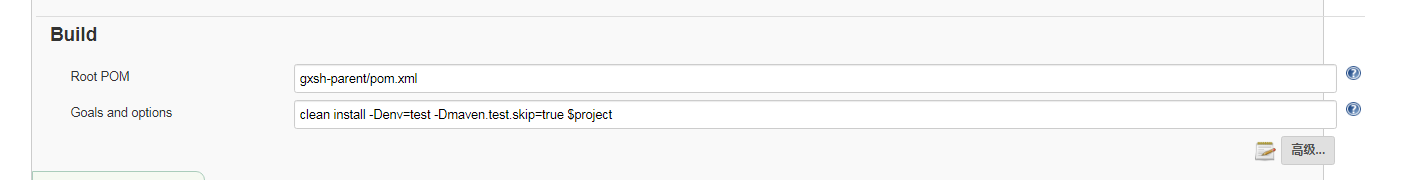
-pl chit-rest

-pl chit-service

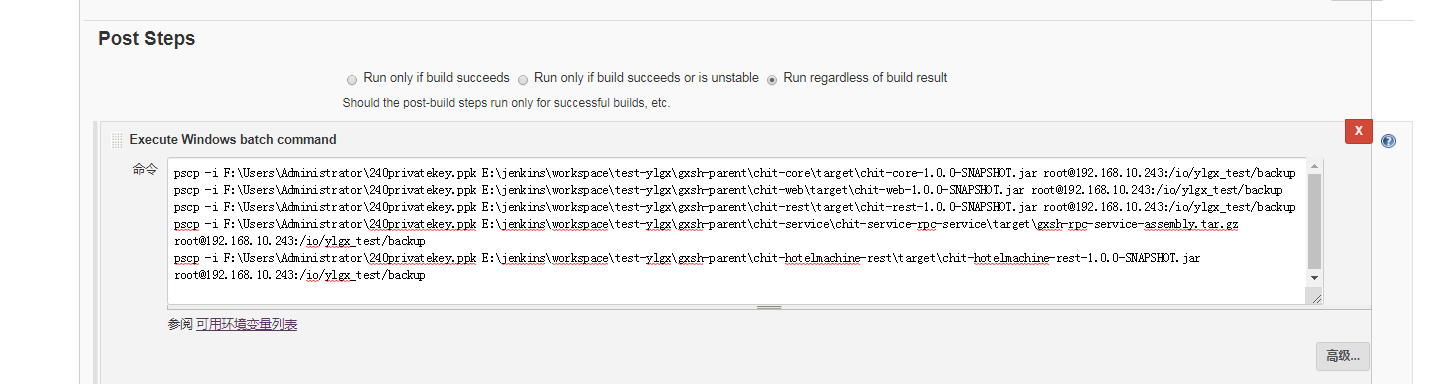
-pl chit-hotelmachine-rest

-pl chit-xzz-rest





clean install -Denv=test -Dmaven.test.skip=true $project



pscp -i F:\Users\Administrator\240privatekey.ppk E:\jenkins\workspace\test-ylgx\gxsh-parent\chit-core\target\chit-core-1.0.0-SNAPSHOT.jar root@192.168.10.243:/io/ylgx\_test/backup

pscp -i F:\Users\Administrator\240privatekey.ppk E:\jenkins\workspace\test-ylgx\gxsh-parent\chit-web\target\chit-web-1.0.0-SNAPSHOT.jar root@192.168.10.243:/io/ylgx\_test/backup

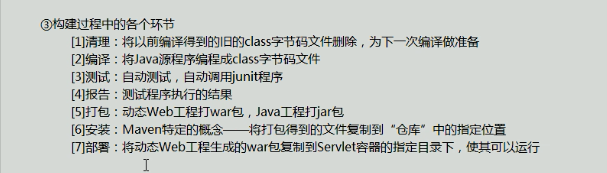
pscp -i F:\Users\Administrator\240privatekey.ppk E:\jenkins\workspace\test-ylgx\gxsh-parent\chit-rest\target\chit-rest-1.0.0-SNAPSHOT.jar root@192.168.10.243:/io/ylgx\_test/backup

pscp -i F:\Users\Administrator\240privatekey.ppk E:\jenkins\workspace\test-ylgx\gxsh-parent\chit-service\chit-service-rpc-service\target\gxsh-rpc-service-assembly.tar.gz root@192.168.10.243:/io/ylgx\_test/backup

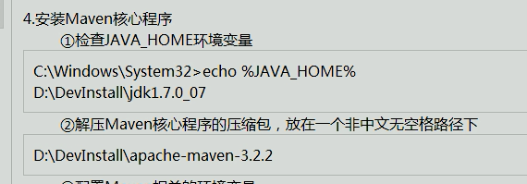
pscp -i F:\Users\Administrator\240privatekey.ppk E:\jenkins\workspace\test-ylgx\gxsh-parent\chit-hotelmachine-rest\target\chit-hotelmachine-rest-1.0.0-SNAPSHOT.jar root@192.168.10.243:/io/ylgx\_test/backup

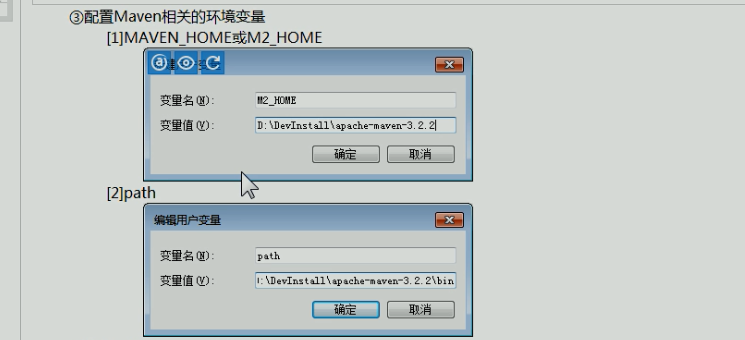
# Maven学习资料

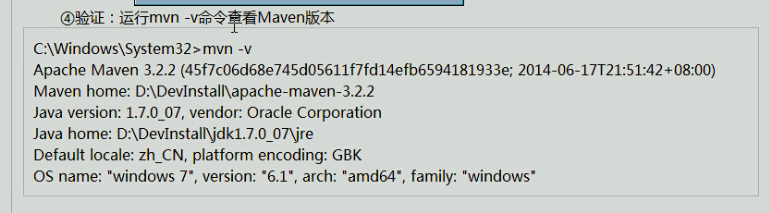
## 什么是构建

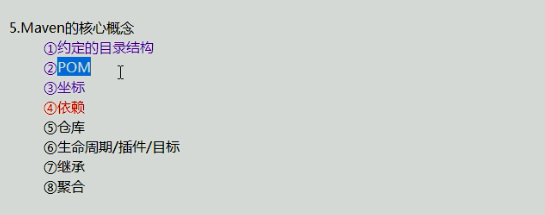


## Maven核心程序安装

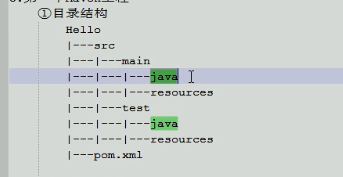


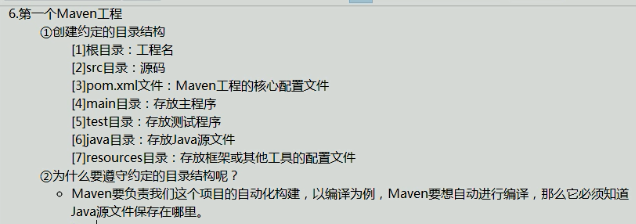




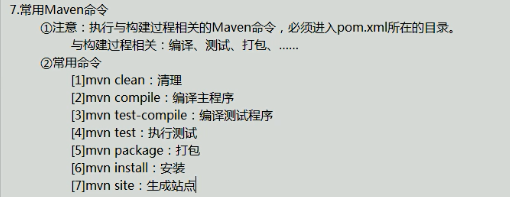


## Maven项目约定的目录结构





## Maven常用命令



-am --also-make 同时构建所列模块的依赖模块；

-amd -also-make-dependents 同时构建依赖于所列模块的模块；

-pl --projects <arg> 构建制定的模块，模块间用逗号分隔；

-rf -resume-from <arg> 从指定的模块恢复反应堆。

实际中用到的命令：

clean install -Dmaven.test.skip=true $project

#clean能够保证上一次构建的输出不会影响到本次构建。

#-Dmaven.test.skip=true，不执行测试用例，也不编译测试用例类。

#jenkins中mvn可省略



Mvn clean install -Dmaven.test.skip=true -Pall

# -Pall，-P 命令maven 会激活项目下的pom.xml配置的<profiles>标签下id为all,没有则默认第一个

mvn clean install -Dmaven.test.skip=true -e -pl siy-pay

#-pl siy-pay 选择构建项目中的siy-pay模块

mvn clean install -Dmaven.test.skip=true -e -pl siy-pay -e

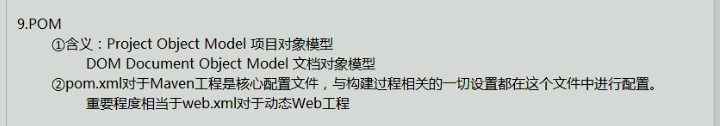
#如果构建出现异常，该-e 参数能让Maven打印完整的stack trace(堆栈跟踪)，以方便分析错误原因。

mvn clean install -Dmaven.test.skip=true -e -pl siy-pay -X #输出的详细日志，显示maven允许的debug信息

Mvn compile #下载很多的东西，第一次使用会下载很多依赖包

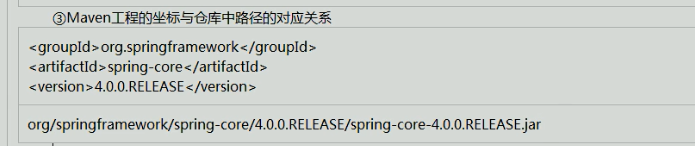


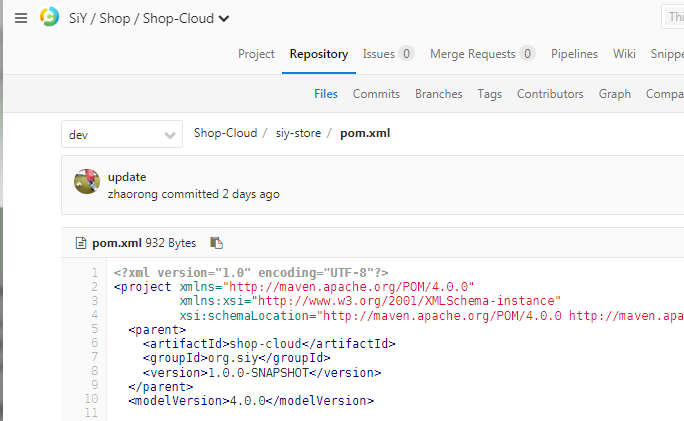
## Pom



## 坐标







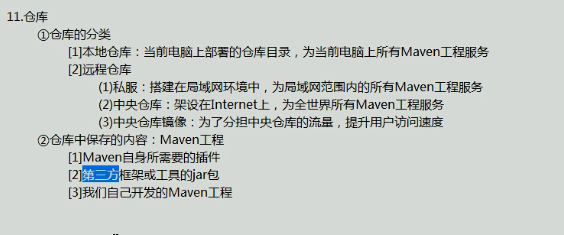
[root@dev repository]# ls /root/.m2/repository/org/siy/shop-cloud/

1.0.0-SNAPSHOT maven-metadata-local.xml

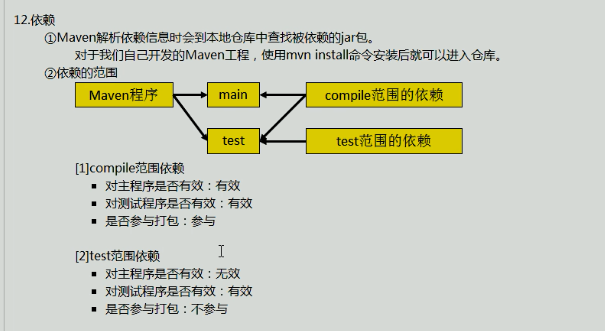
[root@dev repository]# ls /root/.m2/repository/org/siy/shop-cloud/1.0.0-SNAPSHOT/

maven-metadata-local.xml \_remote.repositories shop-cloud-1.0.0-SNAPSHOT.pom

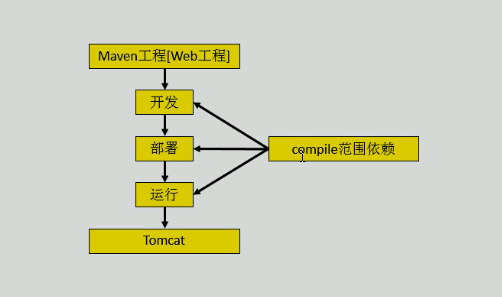
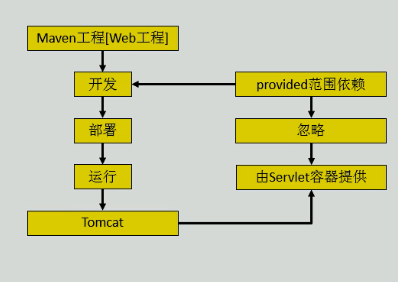
仓库：



## 依赖：





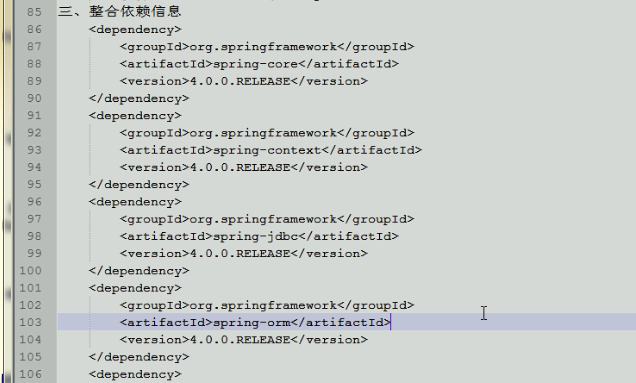
 

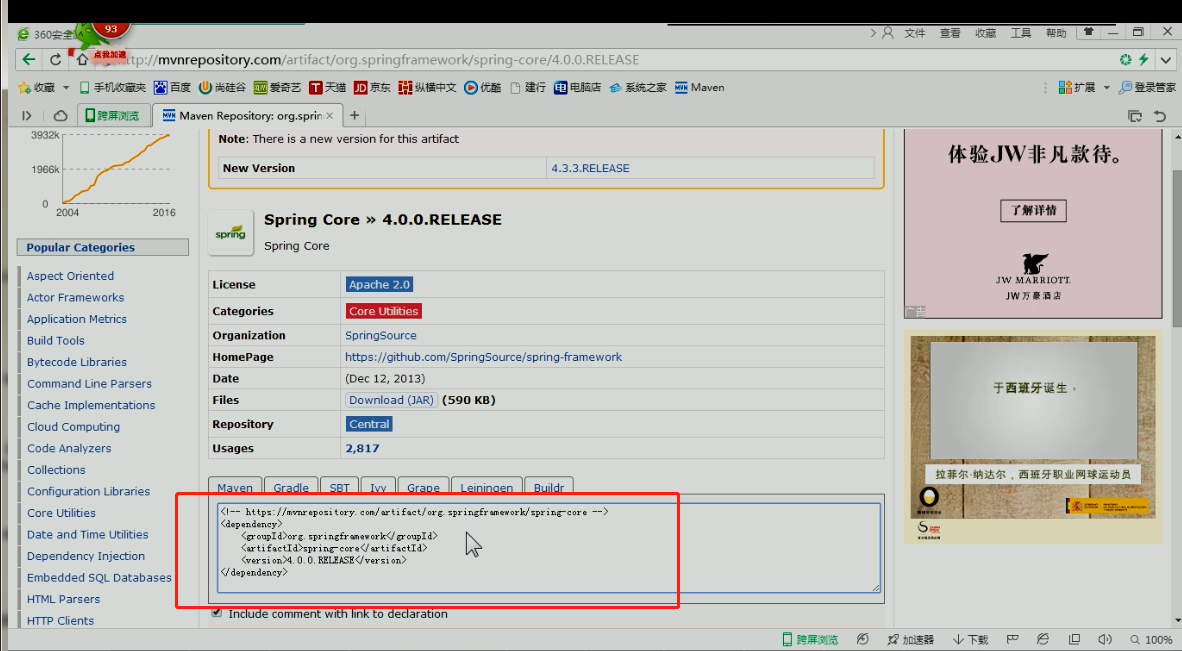
依赖的传递性

依赖的排除

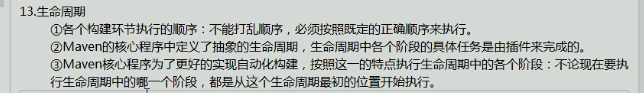
统一管理所依赖jar包的版本

依赖的原则：解决jar包冲突

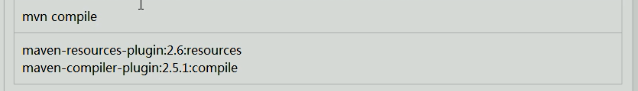




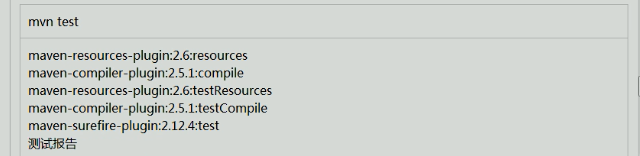
## 生命周期：



例子1



例子2



例子3



Pom.xml部分配置意思

