**云龙系统操作指导**

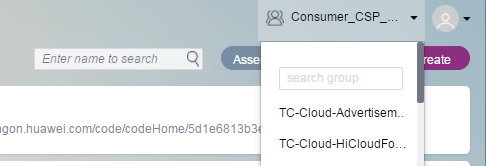
修订记录

| 日期 | 修订版本 | 描述 | 作者 |
| --- | --- | --- | --- |
| 2017/05/31 | 2.0 | 初版 | 江元宝 |
| 2017/07/07 | 2.0 | 1. 微服务命名限制要求 2. VMP中的路径限制变更 3. Codedex/uadp maven项目的私服指定 4. 增加自动部署不依赖tomcat的package.json样例 | 江元宝 |
| 2017/07/28 | 2.0 | （1）增加组装服务的指导 | 江元宝 |
| 2017/07/31 | 2.0 | （1）资源标签配置修改，使用云服务自己申请的服务器，解决资源等待  （2） 更新compile的任务配置，为了解决不同组微服务同名问题，增加serviceID信息的自动获取；  （3）compile任务中app\_define.json/versionset.xml文件的自动生成和解析json文件配置加入模板中，后面不需用户手动添加此任务的配置 | 江元宝 |
| 2017/08/09 | 2.0 | （1）VMP更新，在构建之前可以输入VMP信息 | 江元宝 |

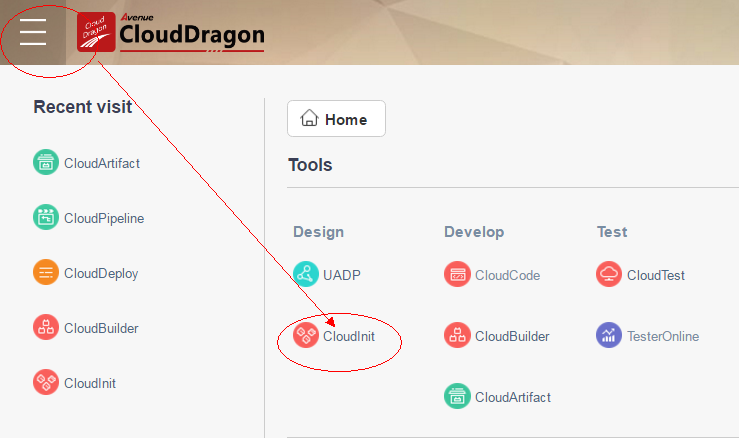
**云龙Dragon 访问地址**：http://szg1.clouddragon.huawei.com/home

1. **微服务创建**

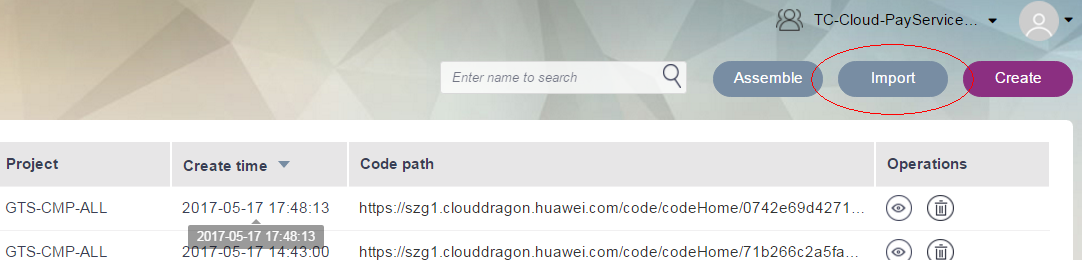
登录系统后，请先右上角对应的群组是否正确



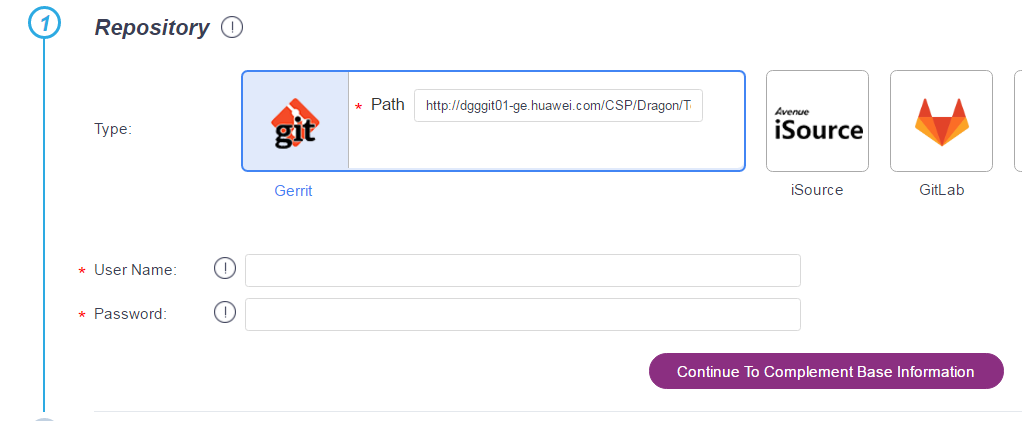
Toolchain 🡪 CloudInit 进入为服务器创建页面



由于当前咱们都是提前建好了git仓，所以**选择“import”方式**来创建微服务



**1.1 填写*Repository*信息**

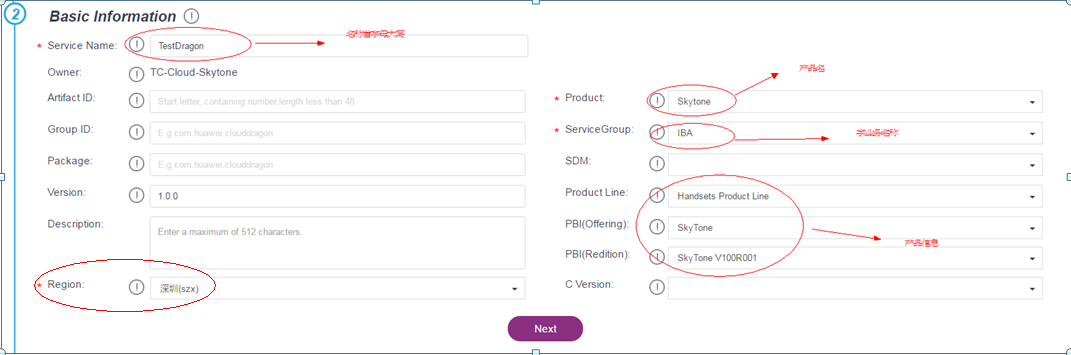


输入git仓路径和域账号信息

**说明：Git path start with 'http://' or 'https://' or 'git@' , and end with '.git'**

1. **在路径中添加.git后缀**
2. **如果是ssh路径的，注意转化为http路径**

**1.2 设置*Basic Information*信息**



**说明**

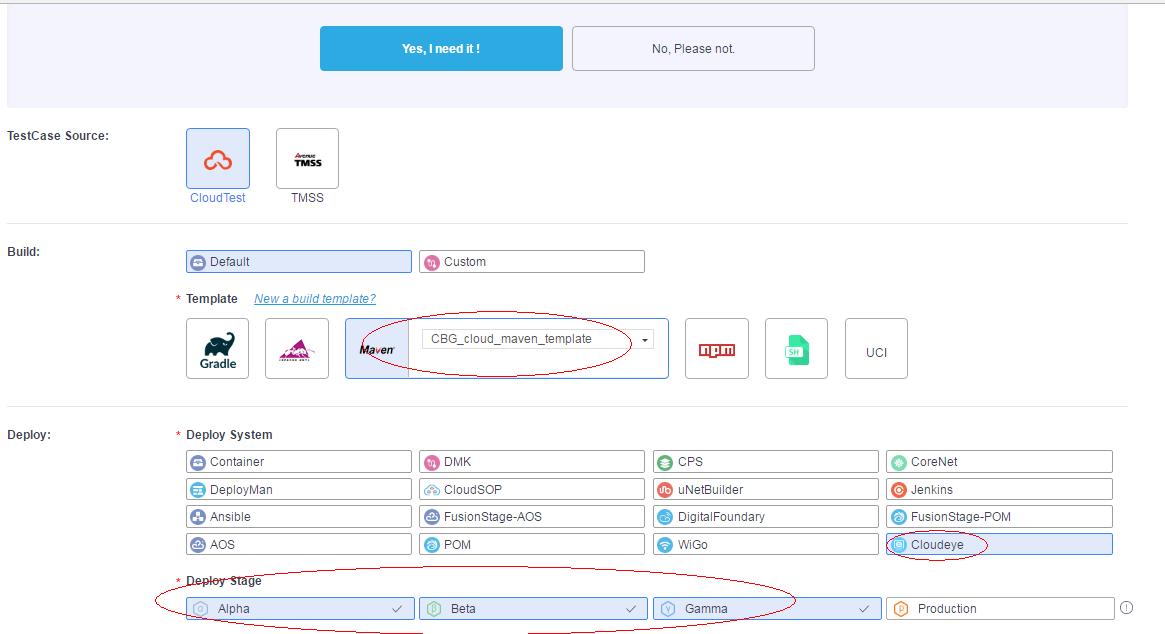
（1）输入微服务名，即工程名，**首字母大写，不要有中划线**

（2）选择产品和子业务信息，对应关系可参阅下面的说明

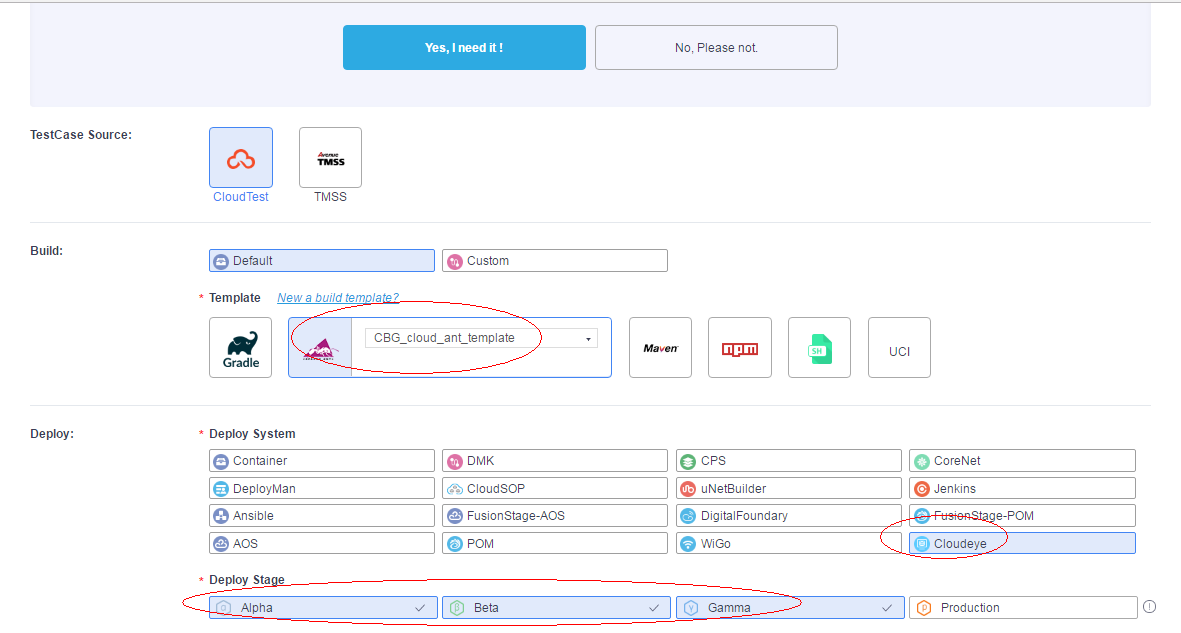
<http://3ms.huawei.com/hi/group/1005981/thread_6185319.html?mapId=7783045&for_statistic_from=all_group_forum>

**1.3 设置构建信息**

**Maven构建，模板选择CBG\_cloud\_maven\_template**



**Ant构建，模板选择CBG\_cloud\_ant\_template**



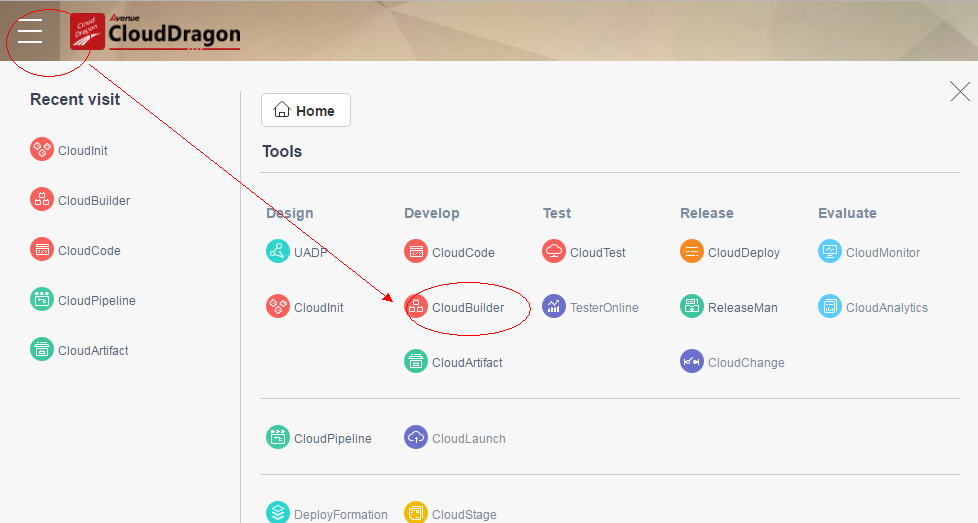
构建任务： 代码下载 + 编译 + 静态检查 + 自动部署 + 自动化测试

**Deploy System 选择**Cloudeye

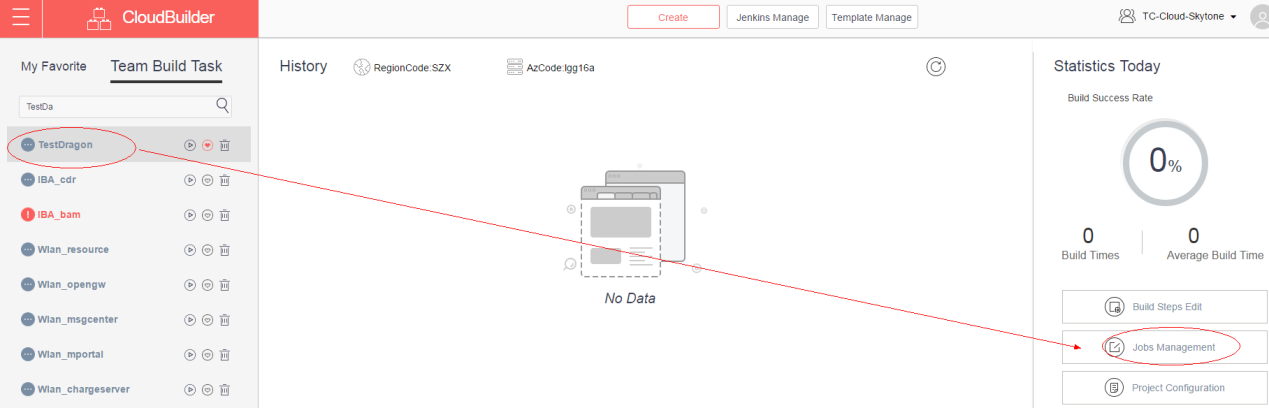
**2 任务检查配置修改**

**2.1 进入任务管理界面**

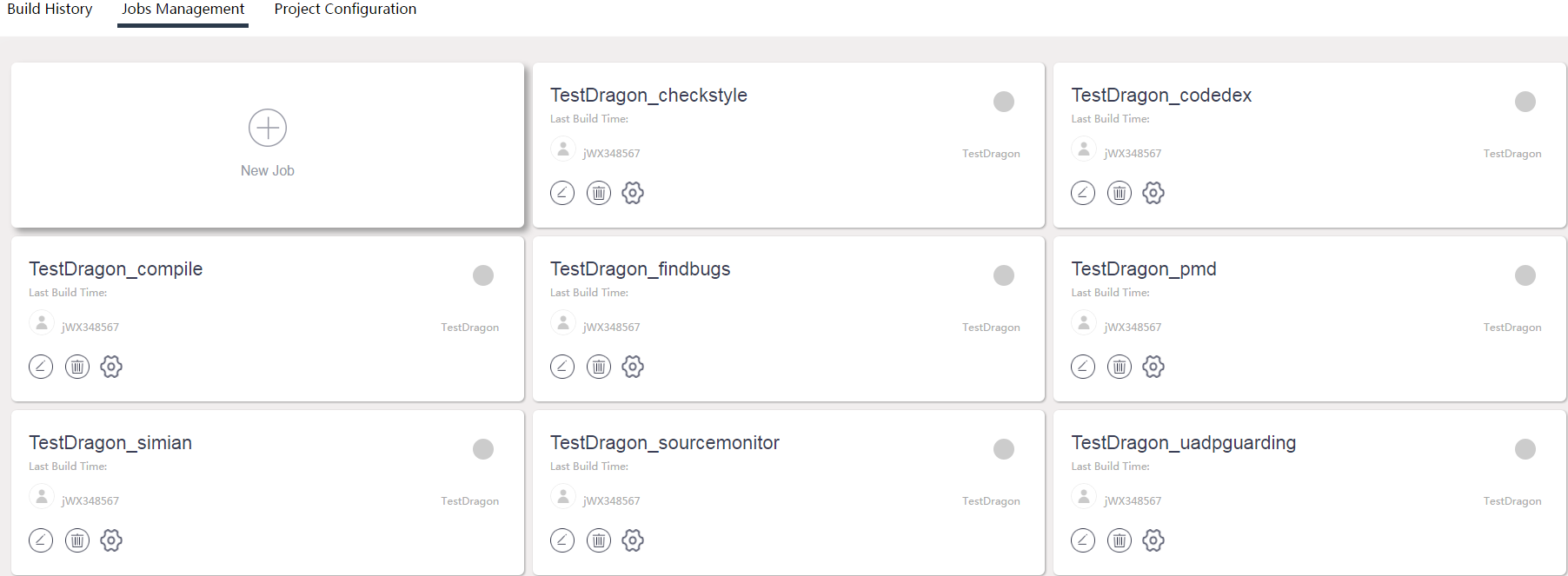
在菜单栏中选择CloudBuilder下



找到自己的微服务，点击“[Jobs Management](https://szg1.clouddragon.huawei.com/build/detail/jobs?projectName=TestDragon&group=g0d578ced2ef24f3ea01aa650035aff5b&projectId=dad51d2e488443e885e1679174b333ca)”



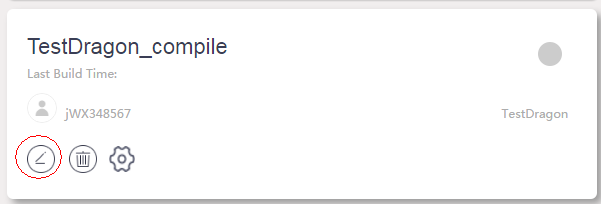
单击对应的服务进入详细页面

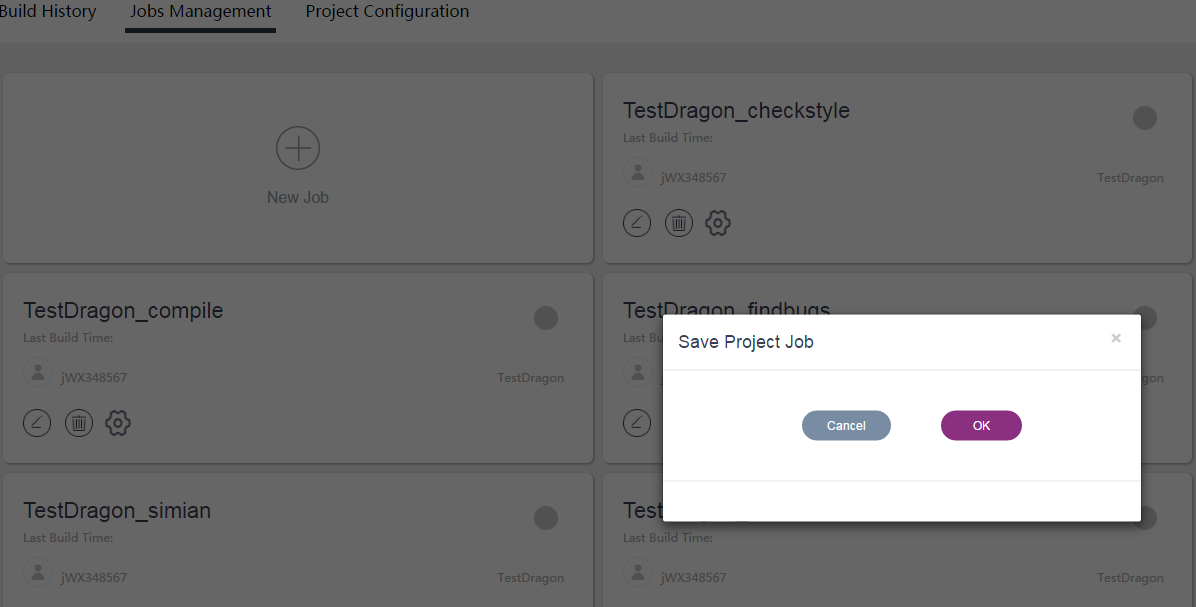


**注意：jenkins配置保存后必须回到云龙原先的云龙界面点击OK才会保存**

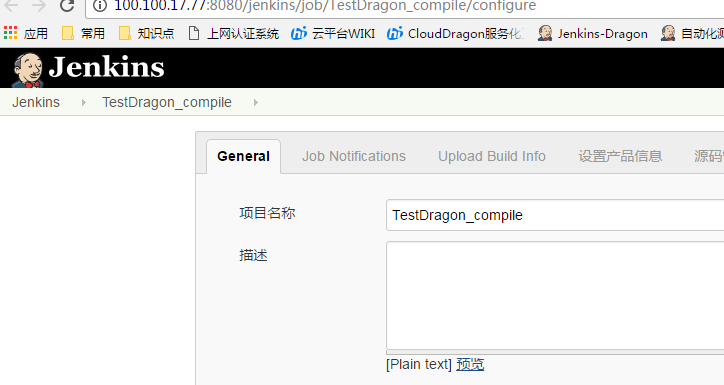
**2.2 Compile 配置修改**

（1）点击“编译”图标进入jenkins 配置界面



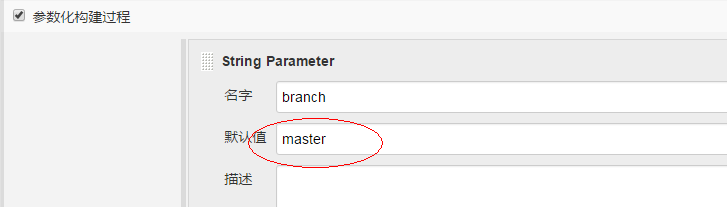


**点击编辑时，浏览器可能会阻止弹出下面的jenkins窗口，注意选择允许**



**（2）修改jenkins配置**

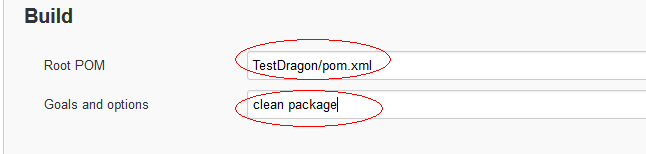
修改Git仓的分支



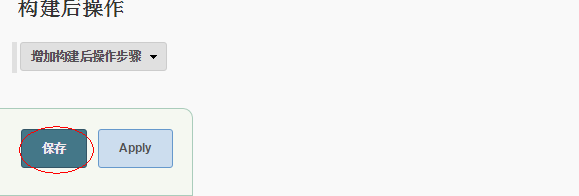
Ant 场景：确认构建脚本名和路径， 可以指定target参数



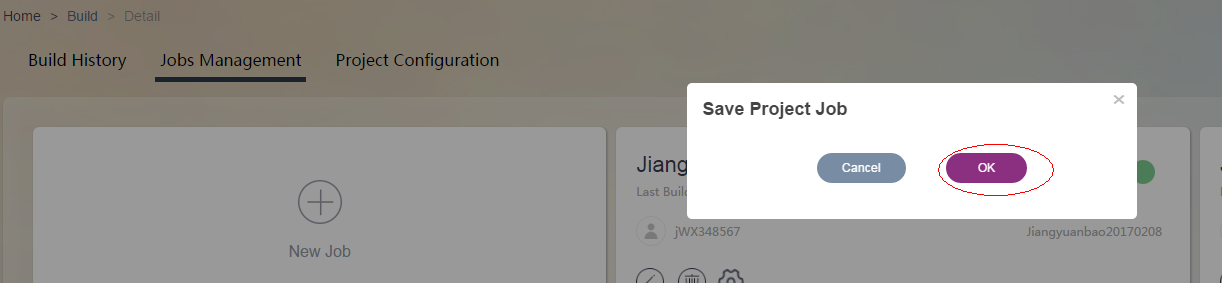
Maven场景：确认构建脚本名和路径，还有goals



（3）保存配置



**注意：jenkins配置保存后必须回到云龙原先的云龙界面点击OK才会保存**



（4）保存之后，可在jenkins 中执行“[Build with Parameters](http://100.100.17.77:8080/jenkins/job/TestDragon_compile/build?delay=0sec)”来检测下配置是否OK，如果有问题重新修改配置



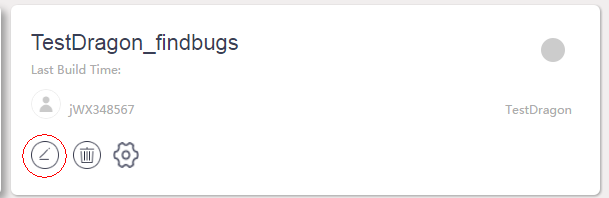


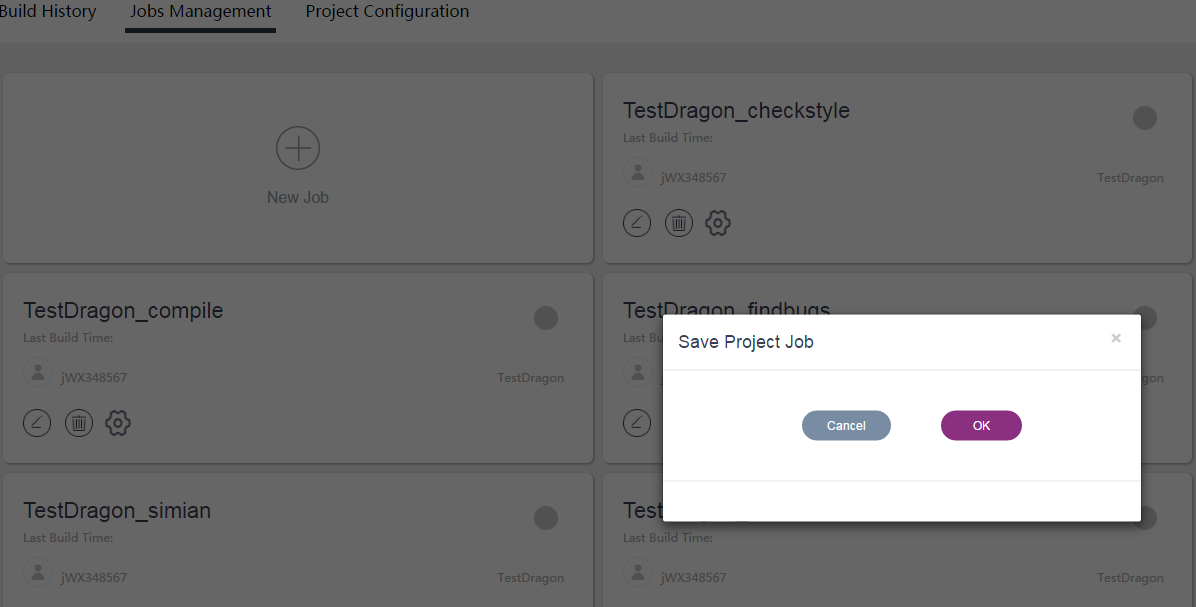




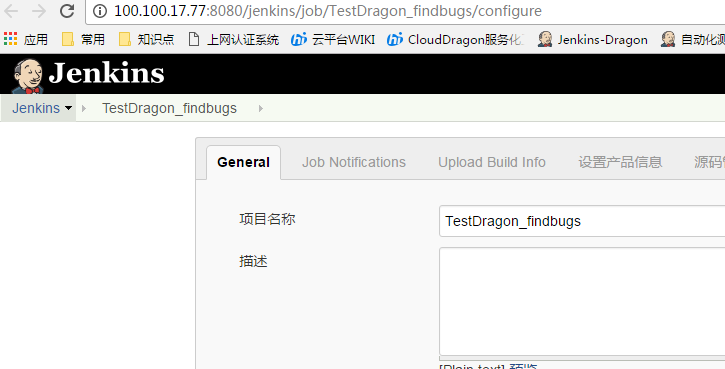
**2.3 Findbugs配置修改**

（1）点击“编译”图标进入jenkins 配置界面



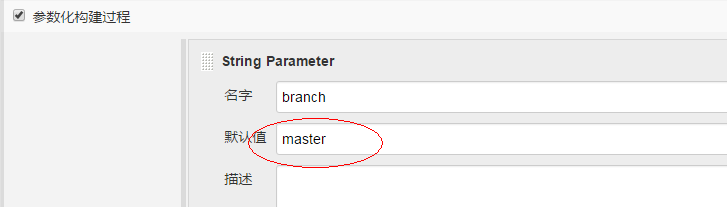


**点击编辑时，浏览器可能会阻止弹出下面的jenkins窗口，注意选择允许**

****

**（2）修改编译脚本名和class文件保存路径**

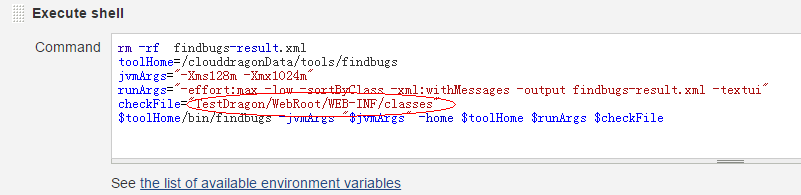
修改Git仓的分支



**Findbugs插件场景：**修改构建脚本名，对于ant， 可以指定target参数



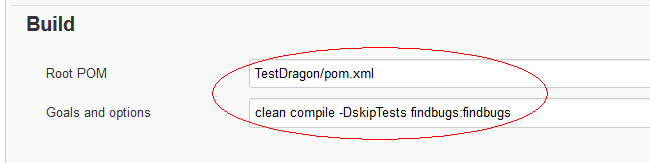
修改class文件保存路径



如果需要添加告警过滤文件的话，添加exclude参数，命令格式如下

$toolHome/bin/findbugs -jvmArgs "$jvmArgs" -home $toolHome $runArgs **-exclude TestDragon/CI/rule/findbugs/excludeFilter.xml** $checkFile

**Maven插件场景：**确认构建脚本名和路径



注意需在pom.xml脚本里指定findbugs的插件版本，云龙中服务器jdk 是1.8，findbugs需使用3.0以上的版本，这里统一指定为3.0.1版本。

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>findbugs-maven-plugin</artifactId>

<version>3.0.1</version>

<configuration>

<findbugsXmlOutput>true</findbugsXmlOutput>

<effort>Max</effort>

<threshold>Low</threshold>

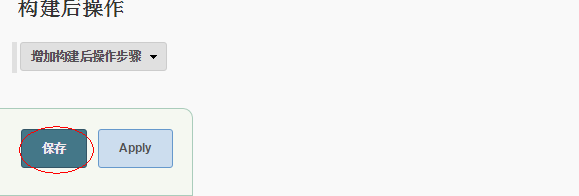
<timeout>9000000</timeout>

<maxHeap>1024</maxHeap>

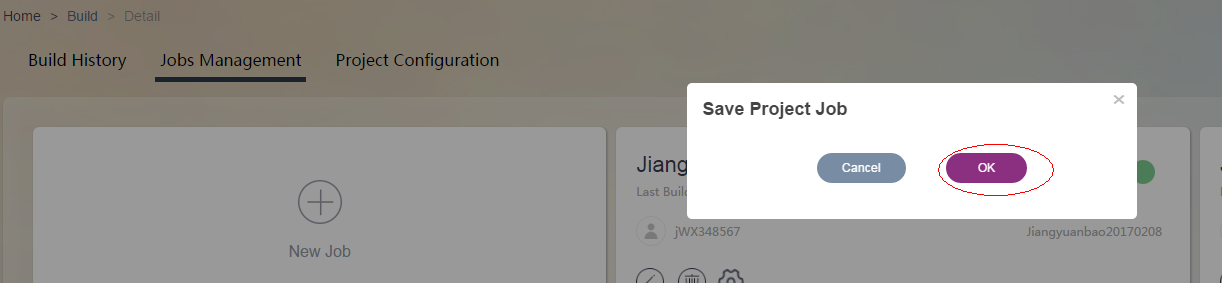
</configuration>

</plugin>

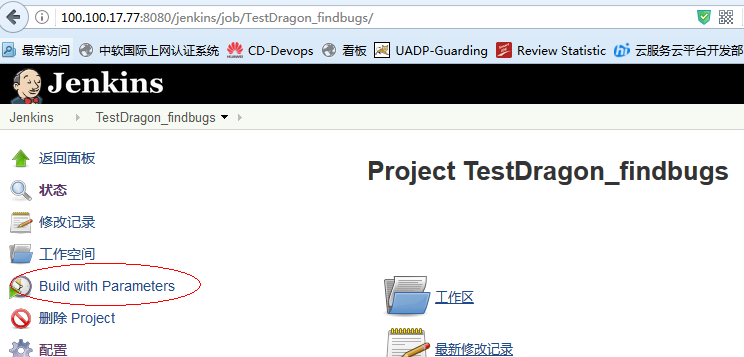
（3）保存配置



**注意：jenkins配置保存后必须回到云龙原先的云龙界面点击OK才会保存**



（4）保存之后，可在jenkins 中执行“[Build with Parameters](http://100.100.17.77:8080/jenkins/job/TestDragon_compile/build?delay=0sec)”来检测下配置是否OK，如果有问题重新修改配置



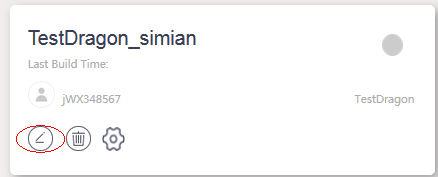


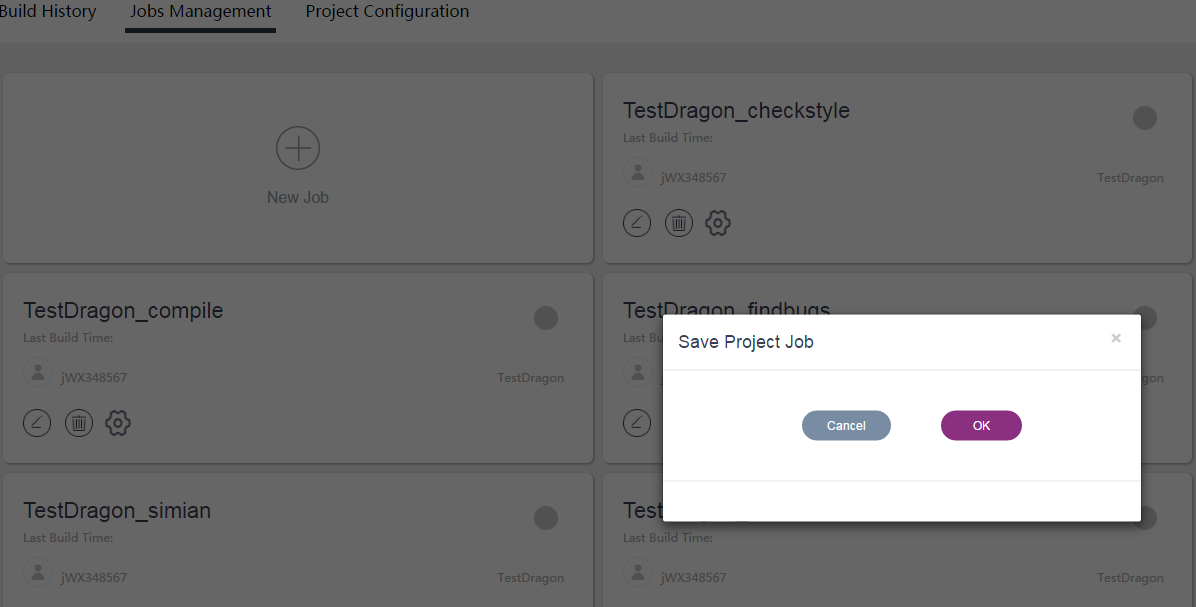




**2.4 Simian(重复率)配置修改**

（1）点击“编译”图标进入jenkins 配置界面



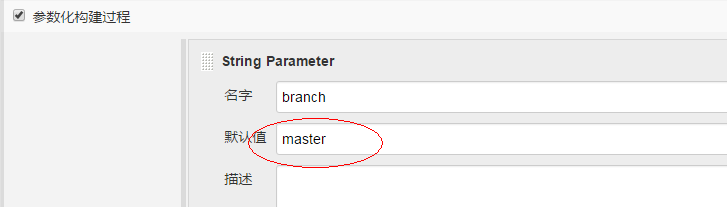


**点击编辑时，浏览器可能会阻止弹出下面的jenkins窗口，注意选择允许**

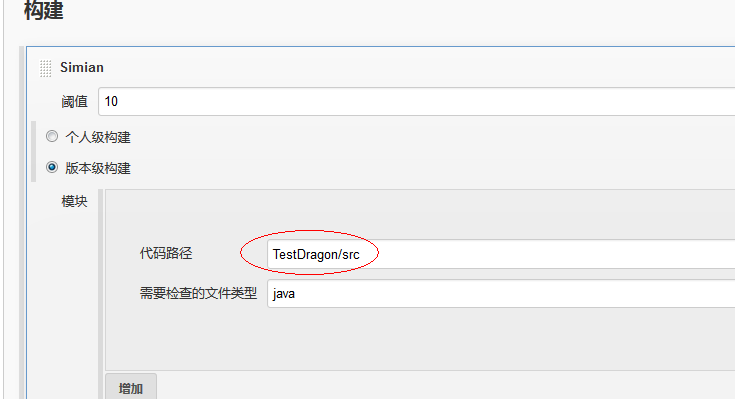


**（2）修改代码扫描路径，相对于代码跟路径**

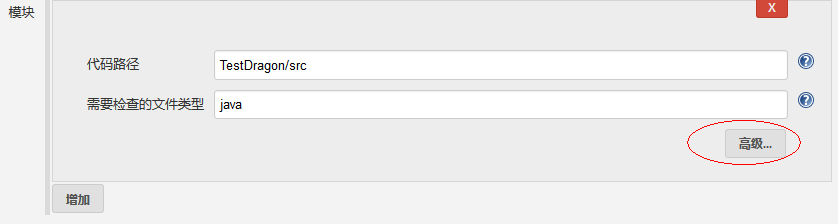
修改Git仓的分支



修改检查目录

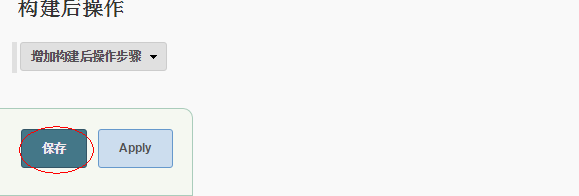


如果扫描不止一个目录的话， 点击高级，通过配置多个“需要包含的文件”来配置多个目录，配置的路径是相对于配置的代码路径

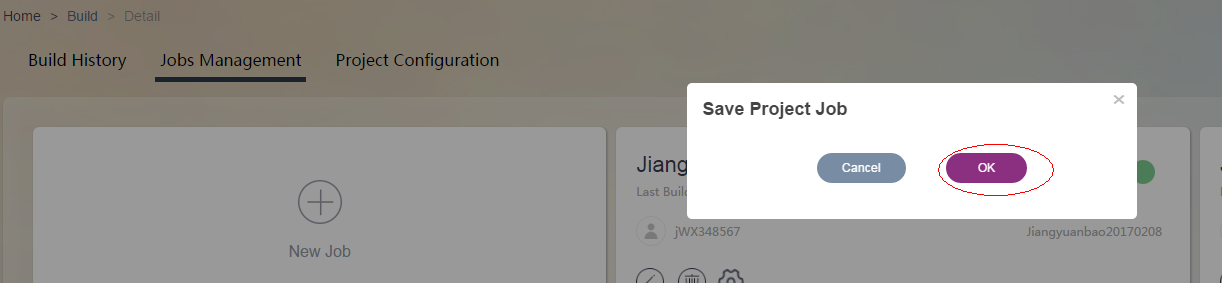




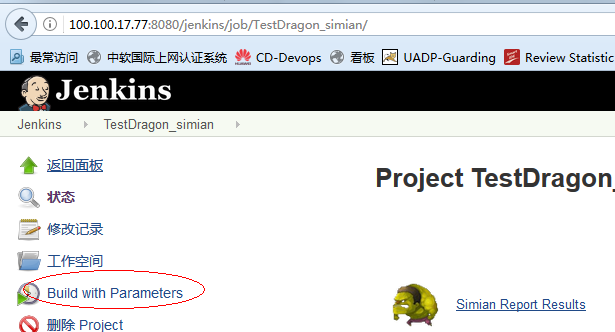
（3）保存配置



**注意：jenkins配置保存后必须回到云龙原先的云龙界面点击OK才会保存**



（4）保存之后，可在jenkins 中执行“[Build with Parameters](http://100.100.17.77:8080/jenkins/job/TestDragon_compile/build?delay=0sec)”来检测下配置是否OK，如果有问题重新修改配置



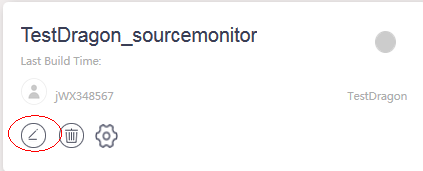


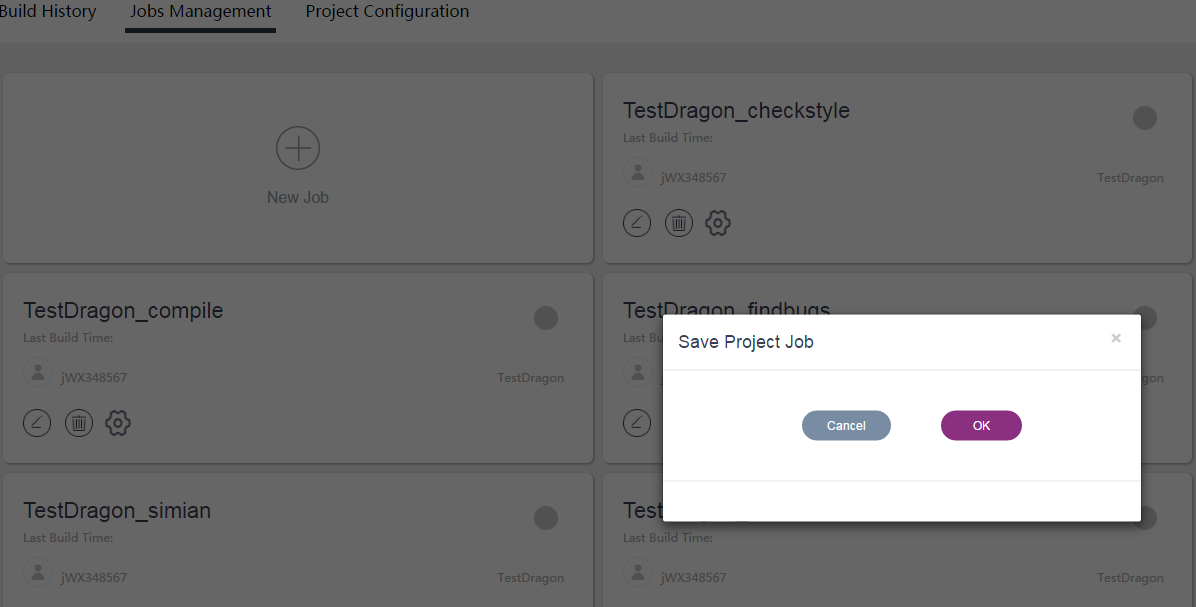




**2.5 SourceMonitor（复杂度）配置修改**

（1）点击“编译”图标进入jenkins 配置界面



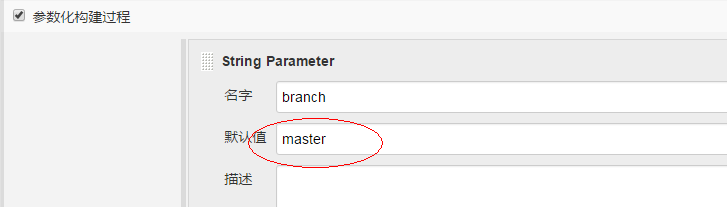


**点击编辑时，浏览器可能会阻止弹出下面的jenkins窗口，注意选择允许**

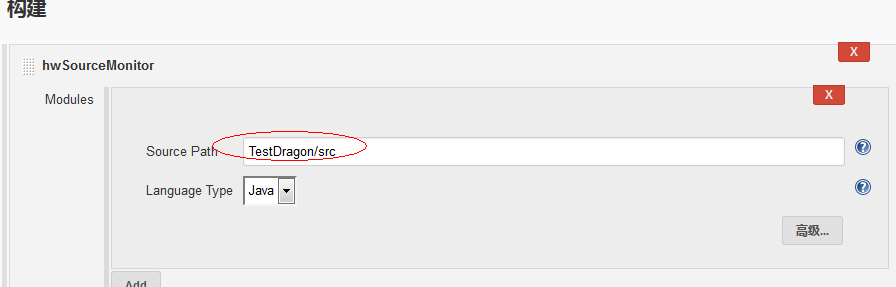


**（2）修改代码扫描路径，相对于代码跟路径**

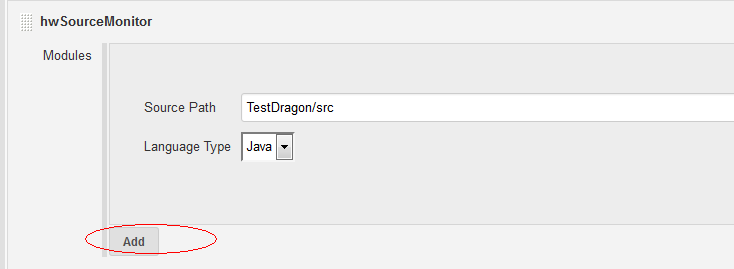
修改Git仓的分支



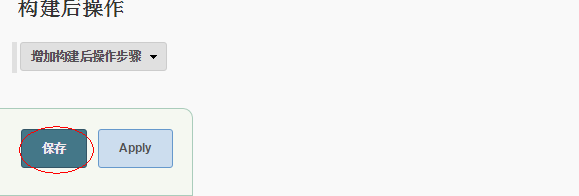
确认检查目录



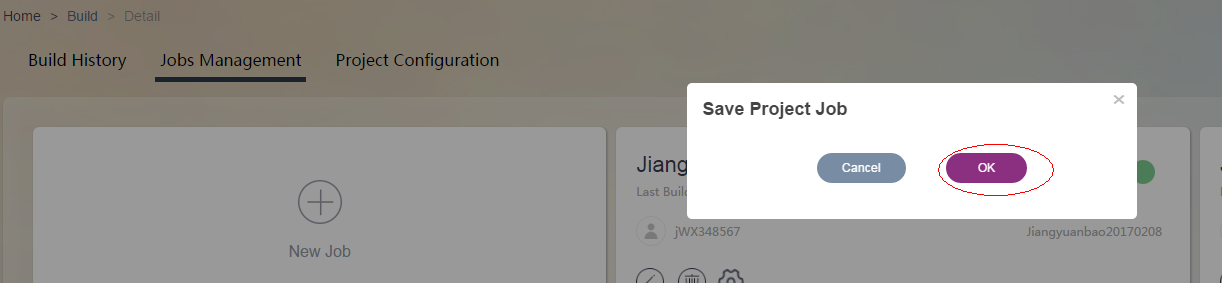
如果不止一个目录的话，点击“Add”可以新增代码扫描目录



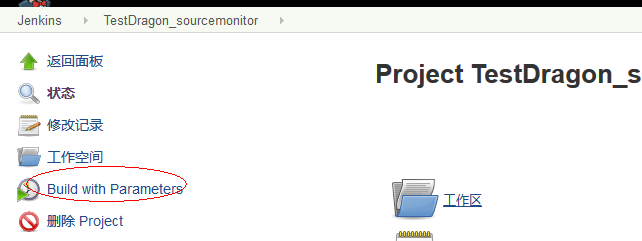
（3）保存配置



**注意：jenkins配置保存后必须回到云龙原先的云龙界面点击OK才会保存**



（4）保存之后，可在jenkins 中执行“[Build with Parameters](http://100.100.17.77:8080/jenkins/job/TestDragon_compile/build?delay=0sec)”来检测下配置是否OK，如果有问题重新修改配置





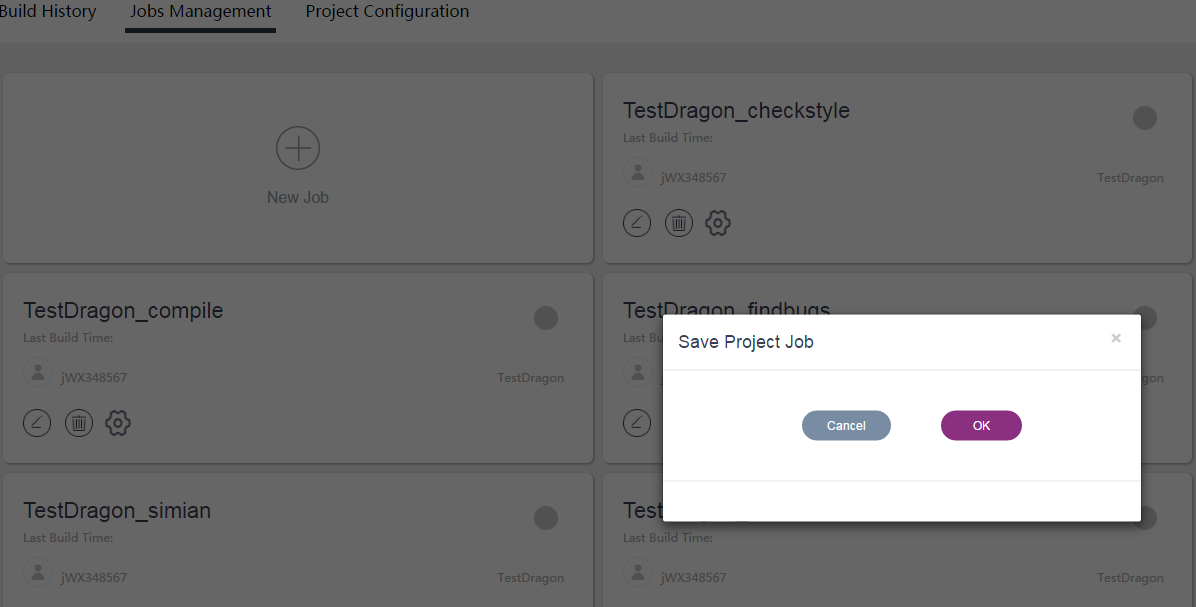




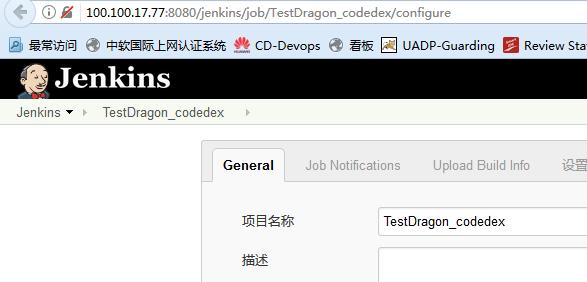
**2.6 CodeDex任务配置**

1. 点击“编译”图标进入jenkins 配置界面





**点击编辑时，浏览器可能会阻止弹出下面的jenkins窗口，注意选择允许**

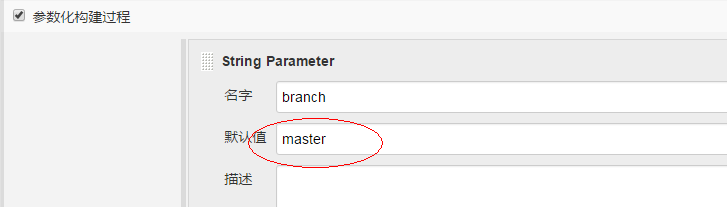


**（2）修改Jenkins配置修改配置**

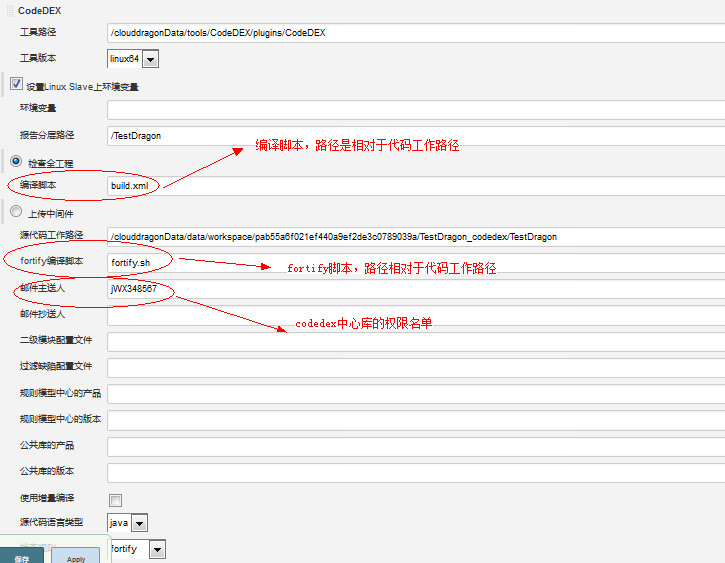
更新产品信息



修改Git仓的分支



更新编译脚本名和forify脚本名及主送人员名单，由于服务器是linux，只支持ant脚本或sh脚本



**说明**：

1. 编译脚本由于只支撑ant脚本和shell脚本，因此对于maven项目，需要新增一个shell的编译脚本，如build\_codedex.sh ，云龙目前使用咱云服务的私服，所有指定的-s参数



#!/bin/sh

mvn clean compile -DskipTests **-s /clouddragonData/tools/apache-maven-3.3.3/conf/settings\_ConsumerCloud.xml**

1. fortify 编译脚本，注意红色部分的修改，其中SCAN\_DIR的根路径与codedex配置中的“源代码工作路径”保持一致

#!/bin/sh

fortify\_HOME=/clouddragonData/tools/CodeDEX/plugins/CodeDEX/tool/fortify/bin

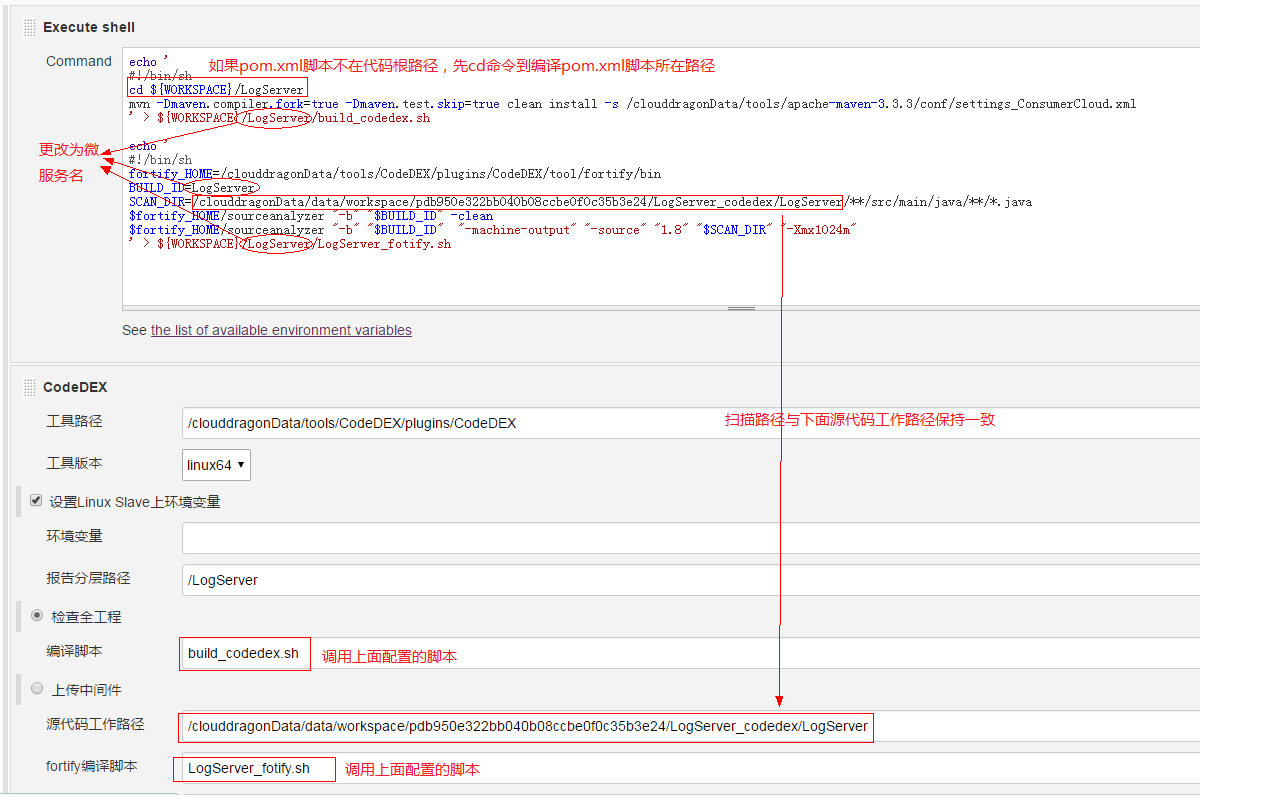
BUILD\_ID=InApp

SCAN\_DIR=/clouddragonData/data/workspace/InApp\_codedex/inapp/\*\*/src/main/java/\*\*/\*.java

$fortify\_HOME/sourceanalyzer "-b" "$BUILD\_ID" -clean

$fortify\_HOME/sourceanalyzer "-b" "$BUILD\_ID" "-machine-output" "-source" "1.8" "$SCAN\_DIR" "-Xmx1024m"

(C) maven编译脚本和fortify 编译脚本也可在jenkins上进行配置，这样可以避免脚本变更需频繁提交GIT库的操作，如下图所示：



echo '

#!/bin/sh

cd ${WORKSPACE}/LogServer

mvn -Dmaven.compiler.fork=true -Dmaven.test.skip=true clean install -s /clouddragonData/tools/apache-maven-3.3.3/conf/settings\_ConsumerCloud.xml

' > ${WORKSPACE}/LogServer/build\_codedex.sh

echo '

#!/bin/sh

fortify\_HOME=/clouddragonData/tools/CodeDEX/plugins/CodeDEX/tool/fortify/bin

BUILD\_ID=LogServer

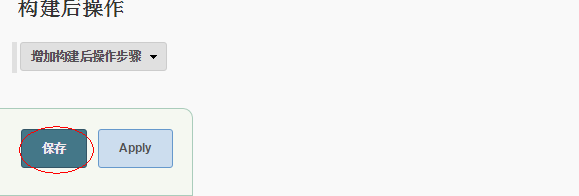
SCAN\_DIR=/clouddragonData/data/workspace/pdb950e322bb040b08ccbe0f0c35b3e24/LogServer\_codedex/LogServer/\*\*/src/main/java/\*\*/\*.java

$fortify\_HOME/sourceanalyzer "-b" "$BUILD\_ID" -clean

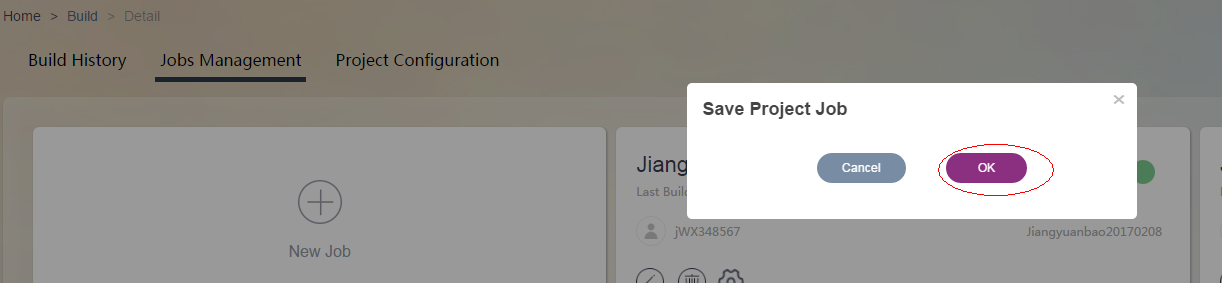
$fortify\_HOME/sourceanalyzer "-b" "$BUILD\_ID" "-machine-output" "-source" "1.8" "$SCAN\_DIR" "-Xmx1024m"

' > ${WORKSPACE}/**LogServer/LogServer**\_fotify.sh

（3）保存配置

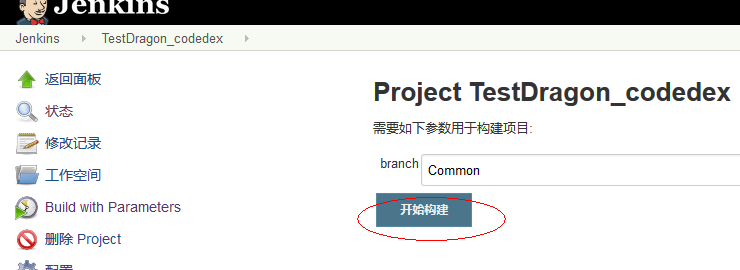


**注意：jenkins配置保存后必须回到云龙原先的云龙界面点击OK才会保存**



（4）保存之后，可在jenkins 中执行“[Build with Parameters](http://100.100.17.77:8080/jenkins/job/TestDragon_compile/build?delay=0sec)”来检测下配置是否OK，如果有问题重新修改配置







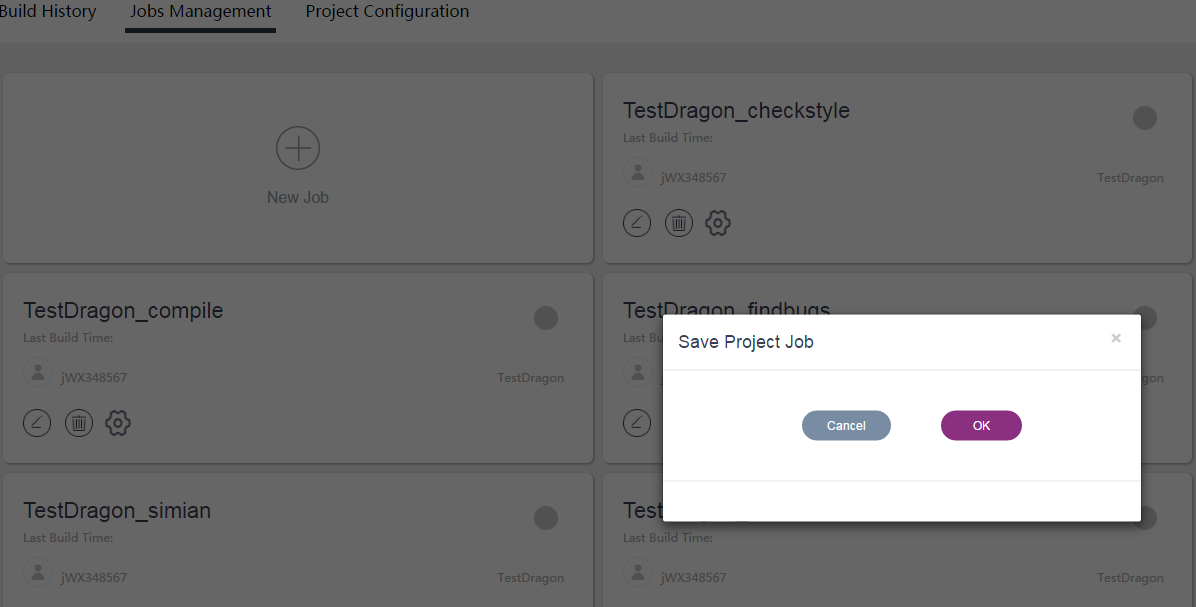


**2.7 UADP任务配置**

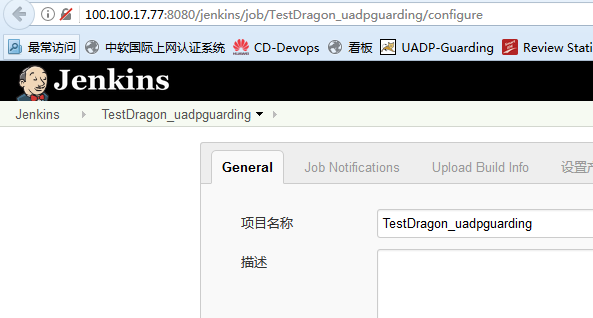
**2.7.1 云龙配置**

（1）点击“编译”图标进入jenkins 配置界面





**点击编辑时，浏览器可能会阻止弹出下面的jenkins窗口，注意选择允许**

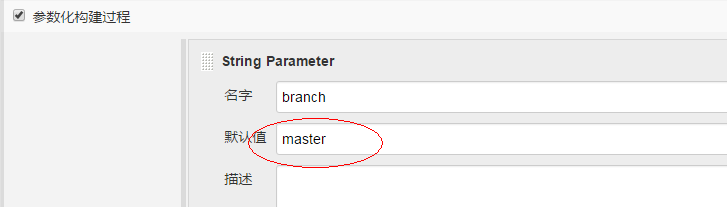


（2）修改Jenkins配置

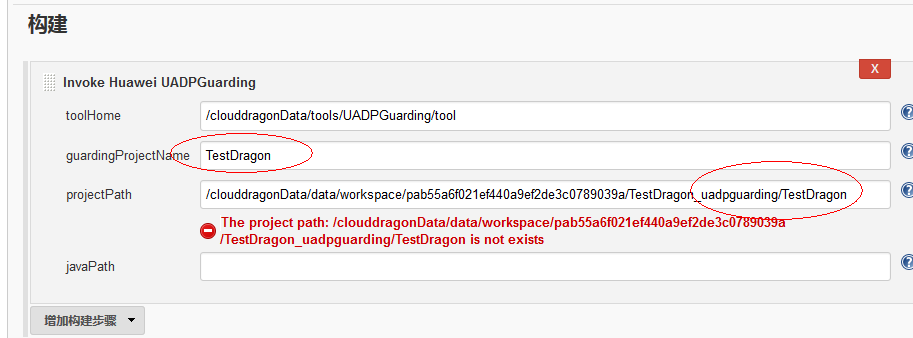
更新产品信息



修改Git仓的分支

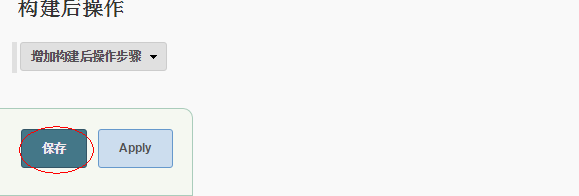


定义工程名，默认是以微服务名作为工程名



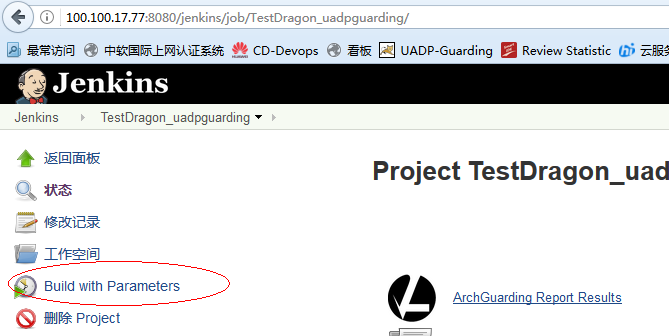
**注意：配置的产品信息和工程名必须与在UADP系统中配置的保持一致**

（3）保存配置



**注意：jenkins配置保存后必须回到云龙原先的云龙界面点击OK才会保存**

（4）保存之后，等在UADP系统也配置好了，可在jenkins 中执行“[Build with Parameters](http://100.100.17.77:8080/jenkins/job/TestDragon_compile/build?delay=0sec)”来检测下配置是否OK，如果有问题重新修改配置



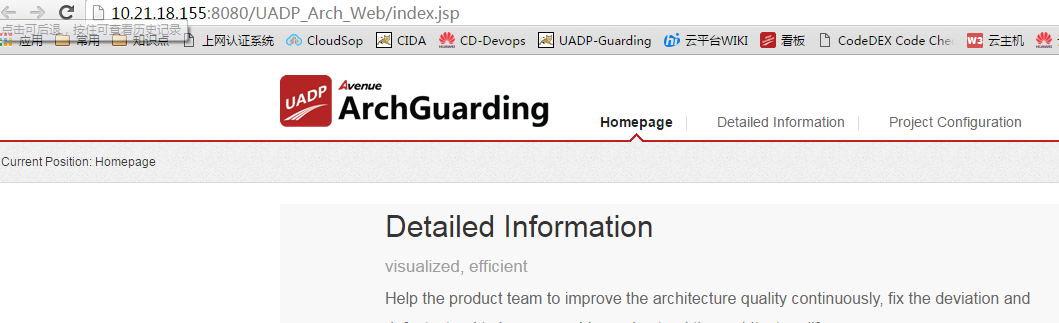






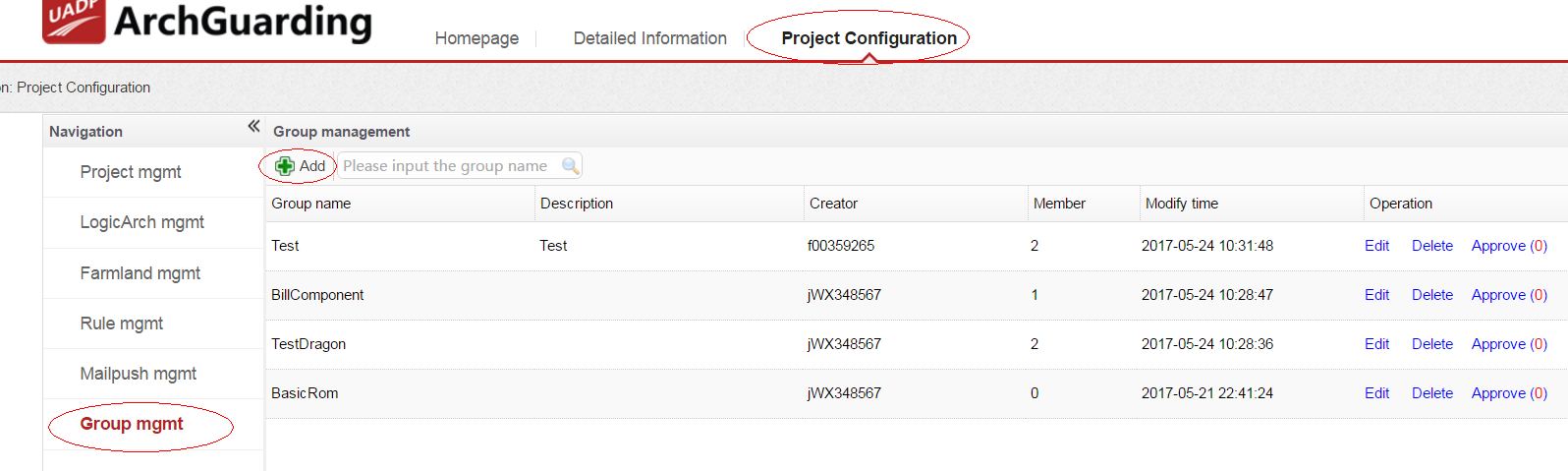
**2.7.2 UADP系统配置**

登录UADP系统：<http://10.21.18.155:8080/UADP_Arch_Web/index.jsp>

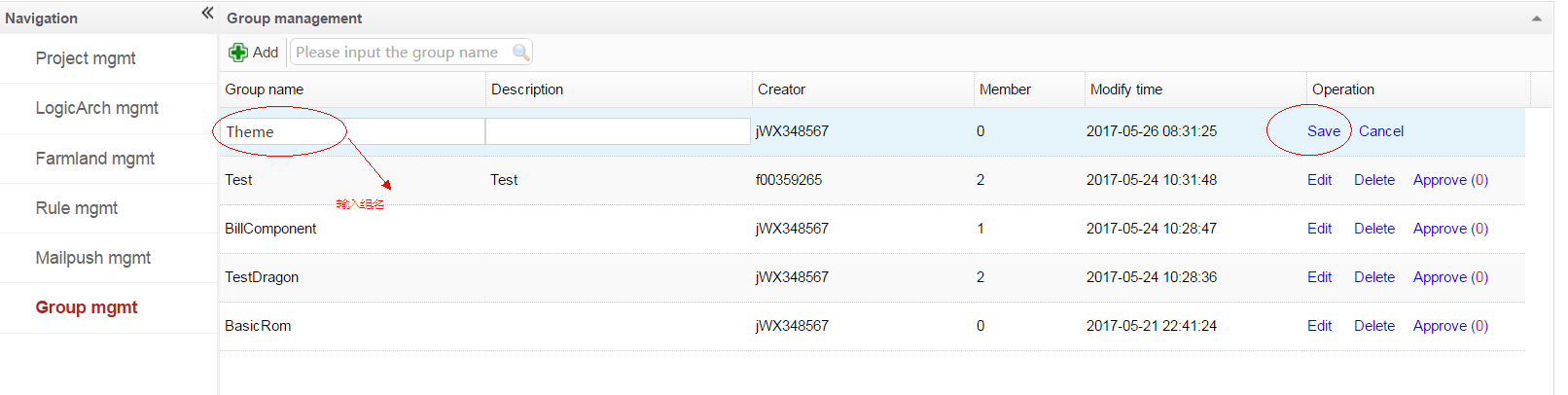


1. 添加群组

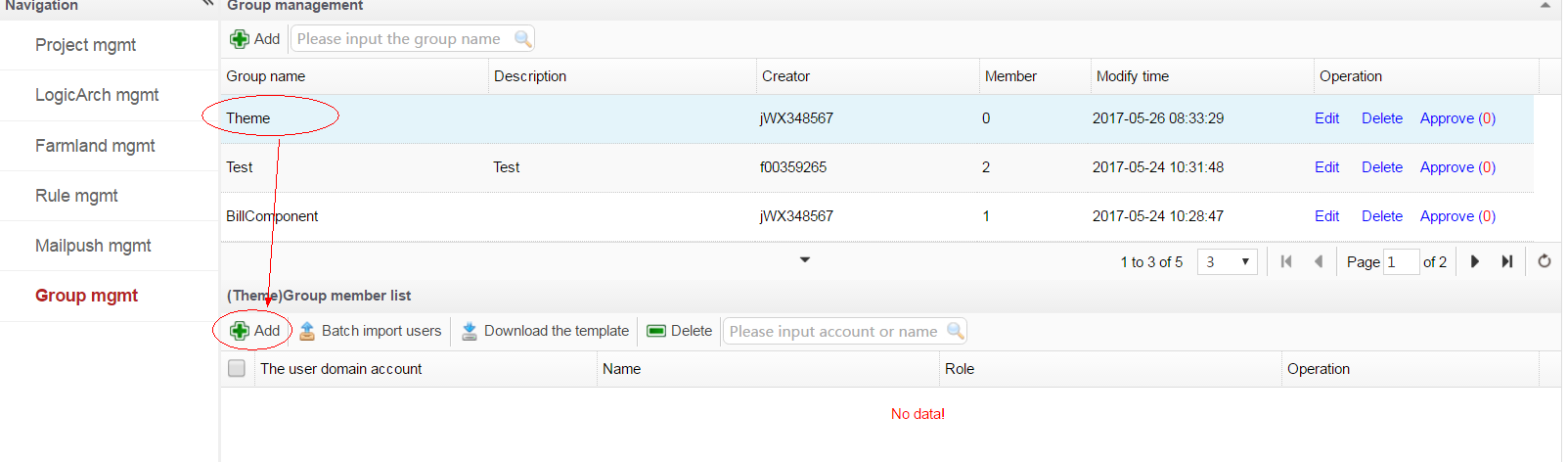
添加一个组，控制访问权限，一个项目或一个产品创建一个即可，可将同项目的人都加入此群组，后面每个模块配置时都选择此group



输入组名，Save保存

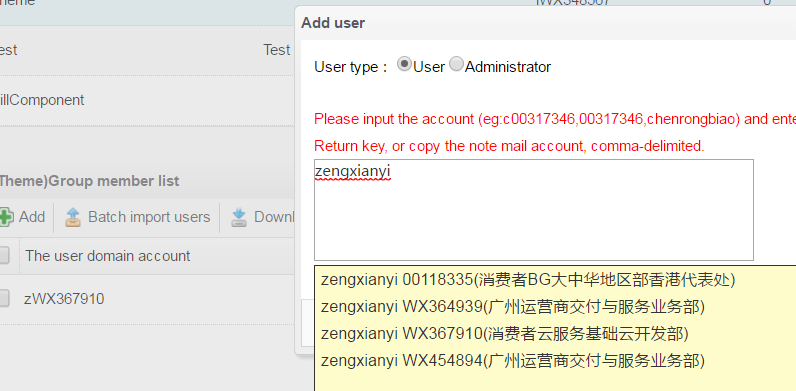


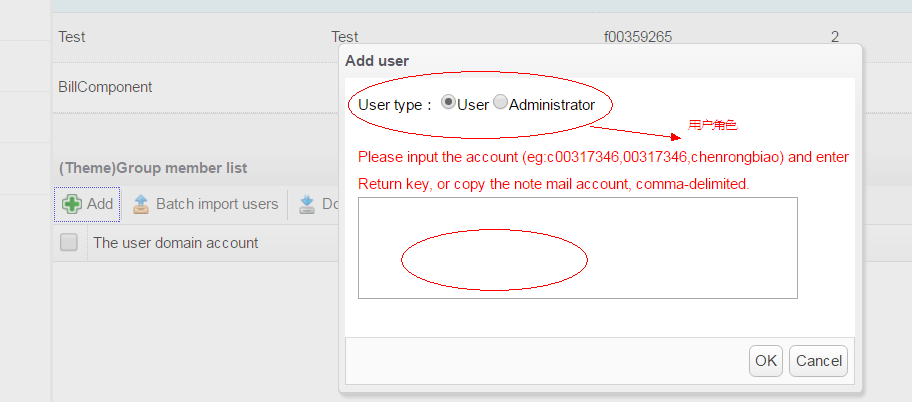
焦点放在新建的组上，点击”Add”添加成员



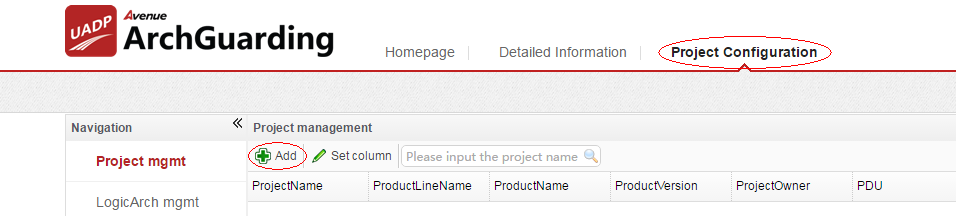
选择用户角色，输入域账号，多个之间逗号分隔，然后OK保存

**注：默认组的创建者已是管理员**

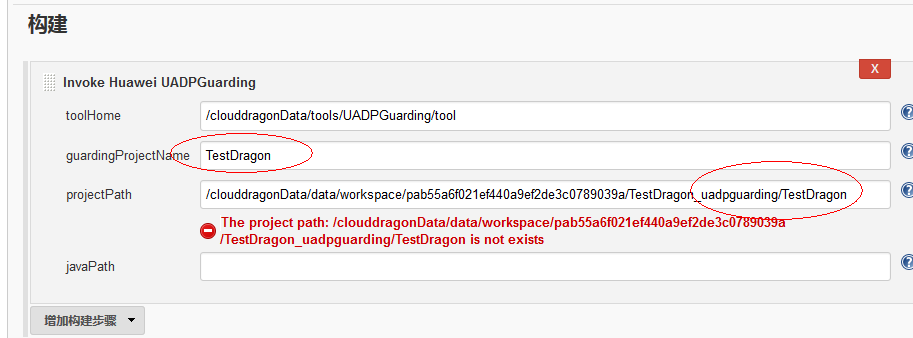


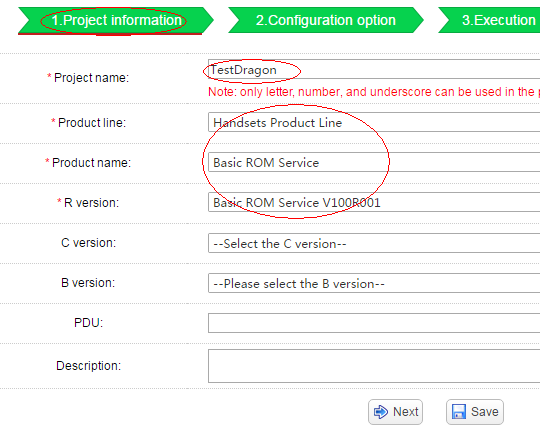


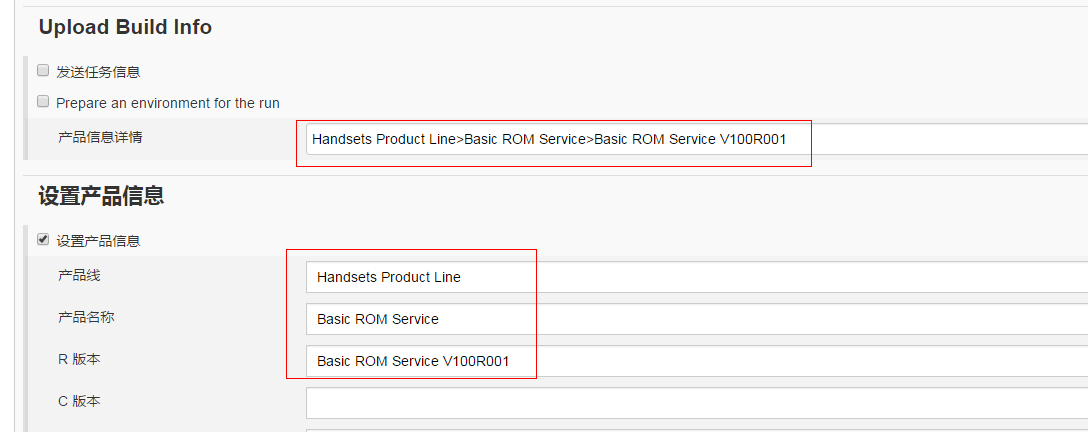
（2）添加工程



**输入工程名和产品信息，注意与在jenkins中配置的guardingProjectName保持一致**

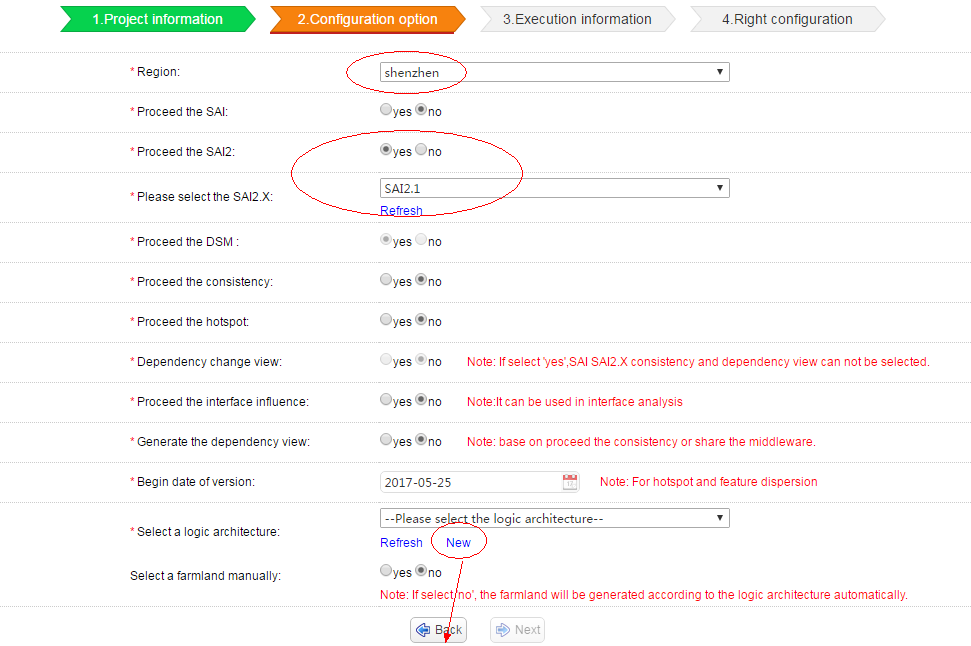






注意：产品线信息需与jenkins上配置的保持一致，否则构建会失败。

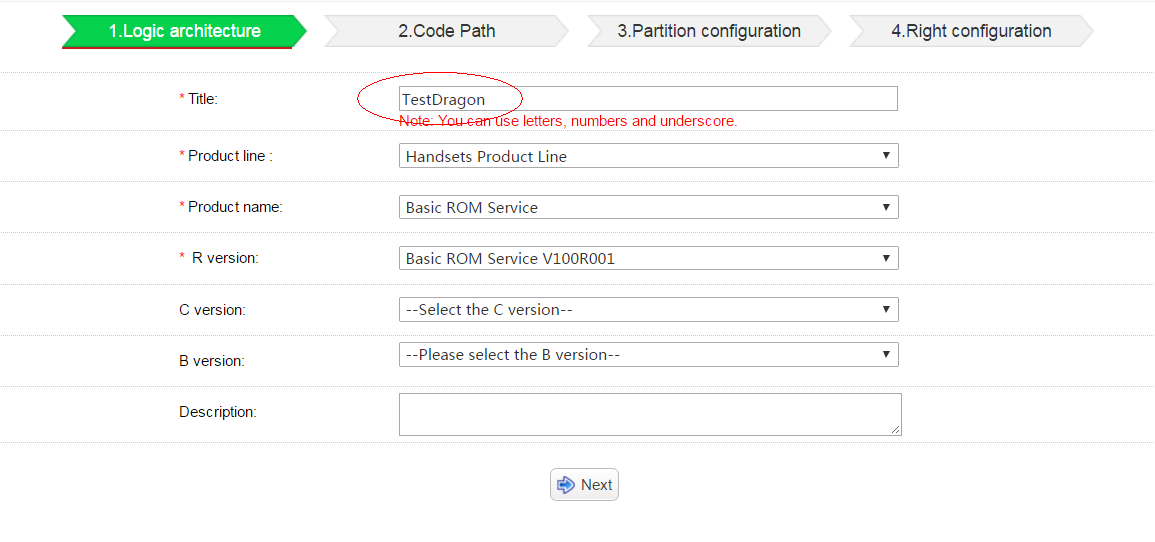
度量选择SAI2中的2.1版本



**点击New会弹出一个新窗口**，配置逻辑结果

1. 配置逻辑结构

输入Title信息，记住此名字，后面会用到

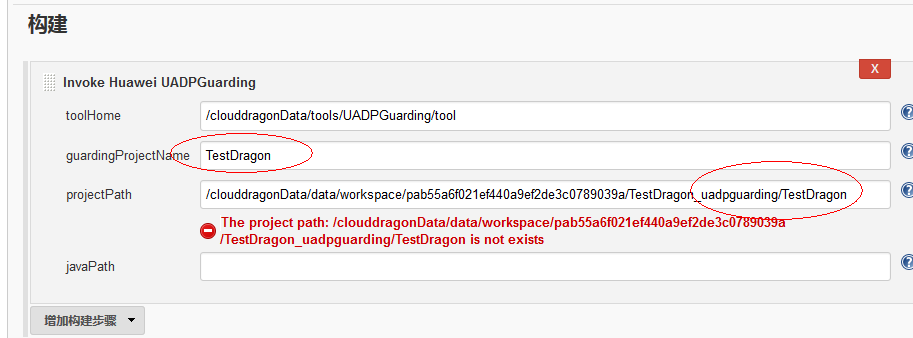


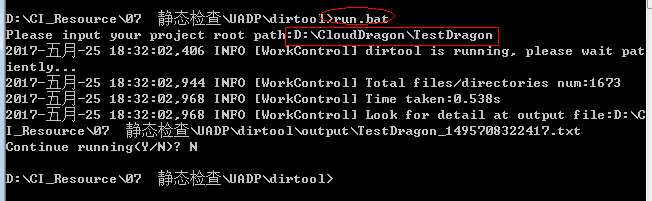
选择语言



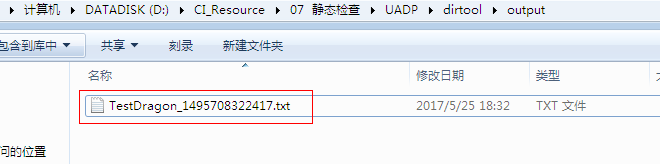
由于使用的是git库，Code repository type选择“Local directory”

此种方式需要上传代码路径的配置文件，下载提供的工具dirtool.zip，解压后执行在本地执行run.bat，执行过程中输入代码在本地的根路径，**注意输入的目标路径与上面jenkins中配置的一致， 否则构建时会失败报路径错误；例如：下面例子所示jenkins中projectPath路径配置到TestDragon这级目录，那输入的本地路径也到TestDragon这级目录。**





将在工具的**output**目录下生成的txt文件上传到UADP系统

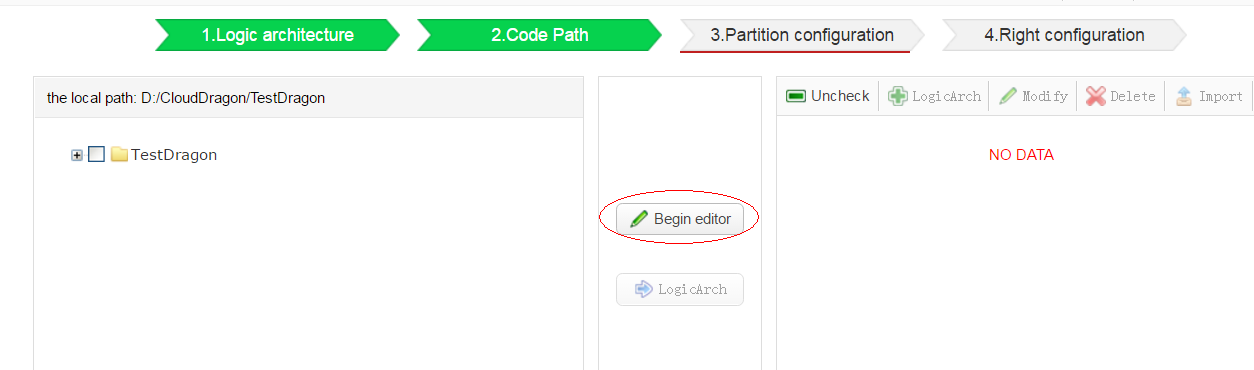


上传生成的txt文件



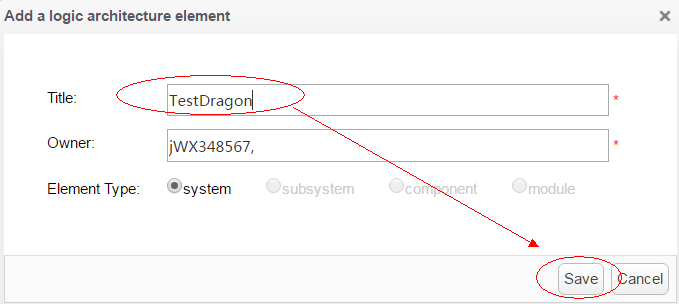


代码扫描范围设置

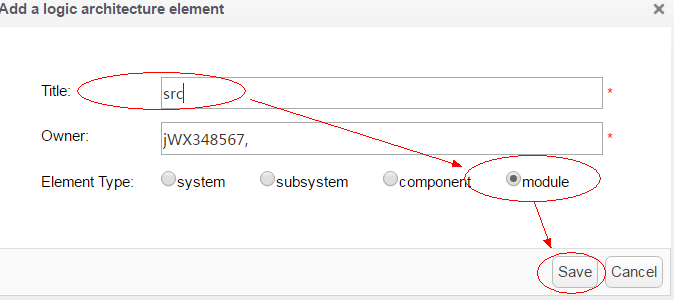


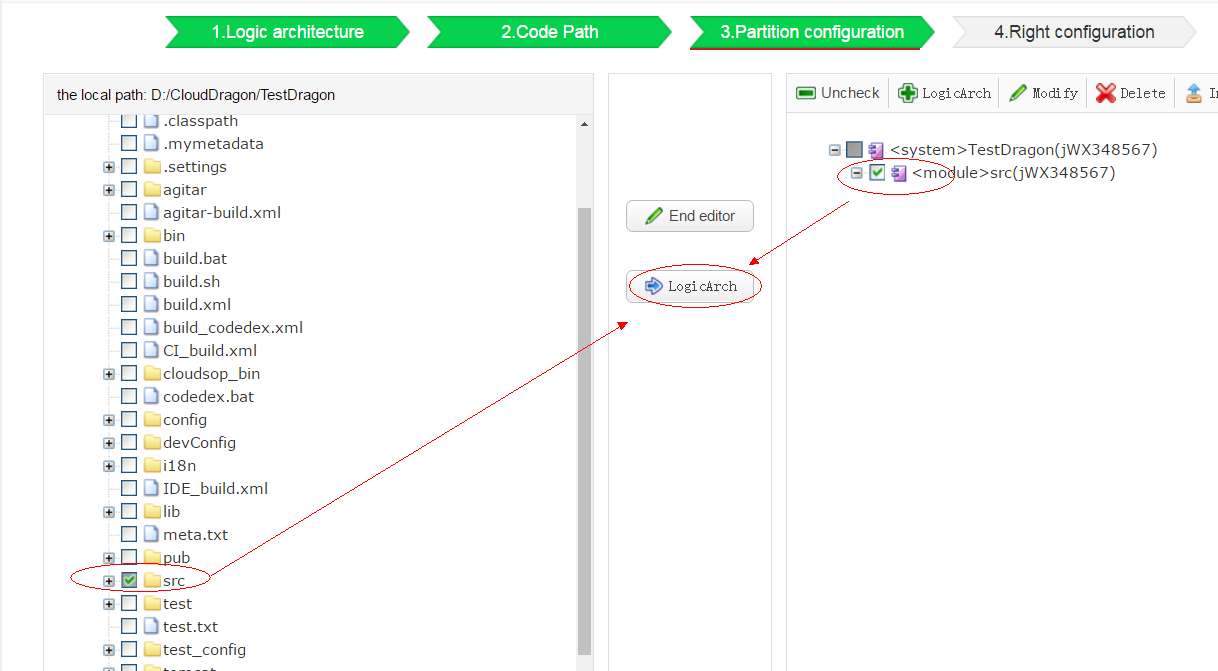
**先增加system节点，然后在system节点下增加module节点，注意：只有moudule节点下才可添加扫描范围**

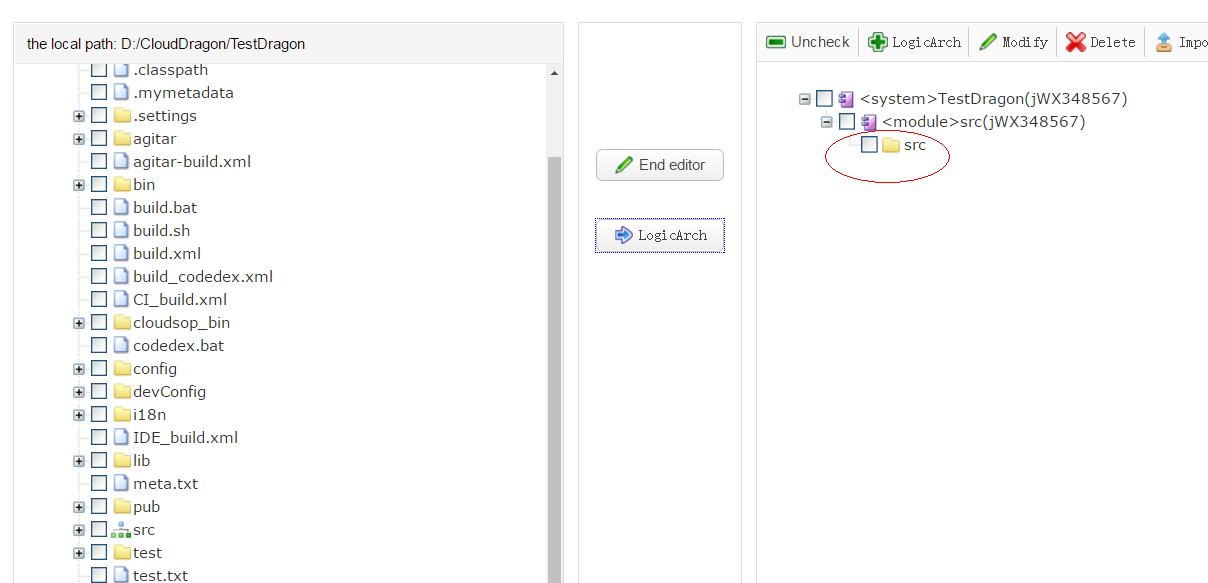






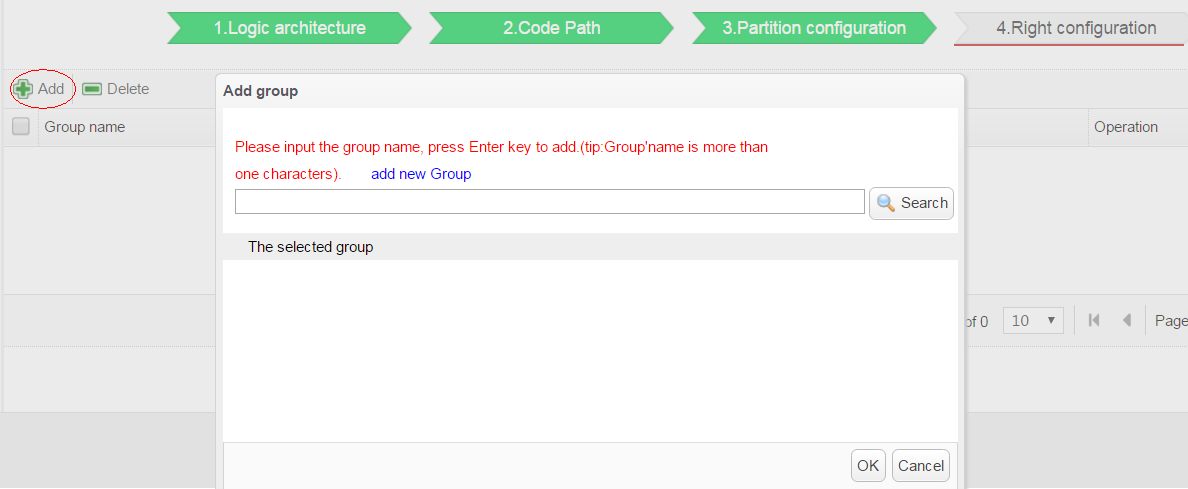


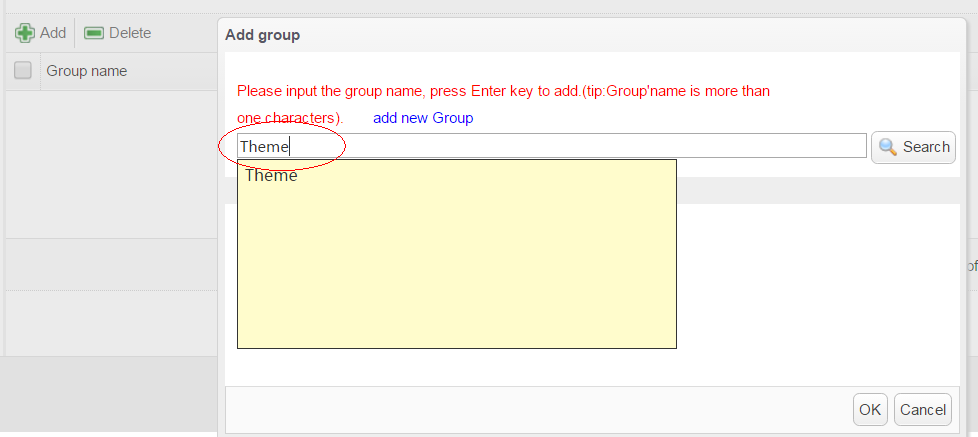


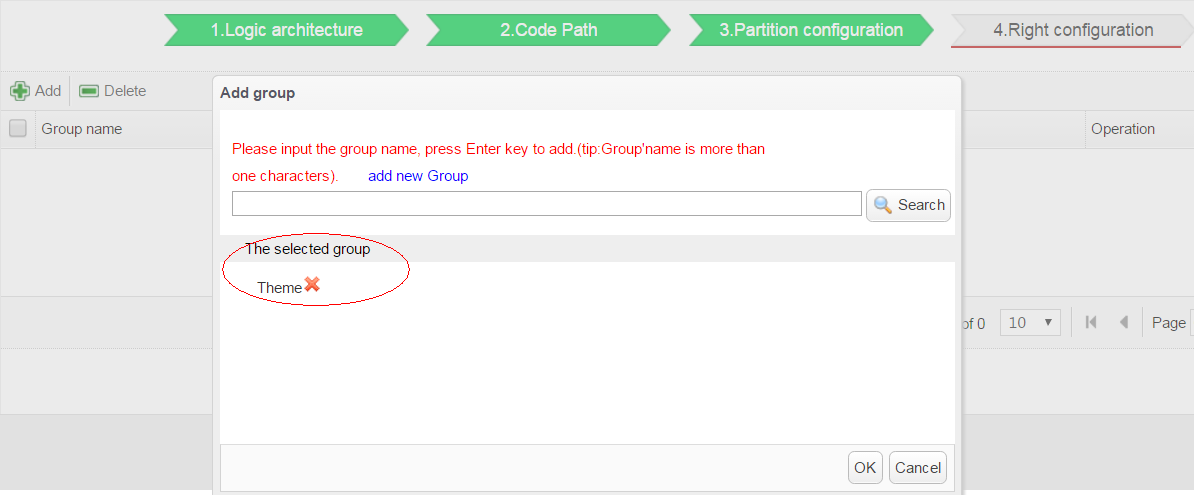


Next下一步权限配置,Add选择新建的组或已存在的组

Add



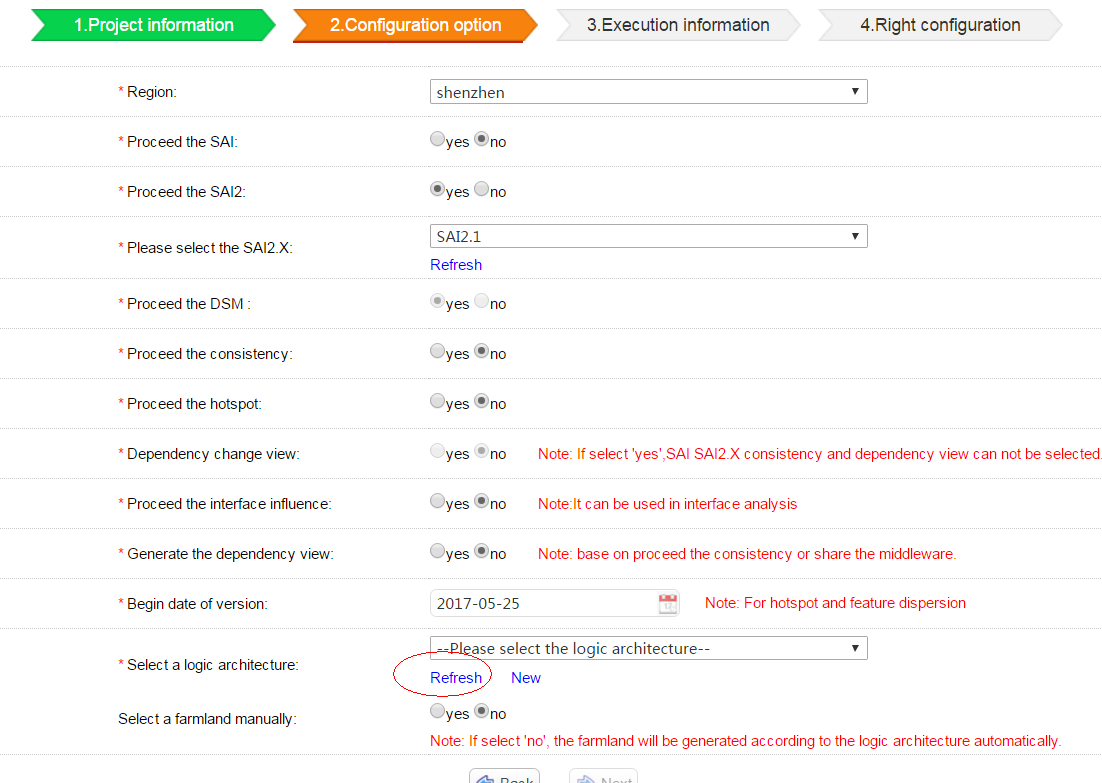




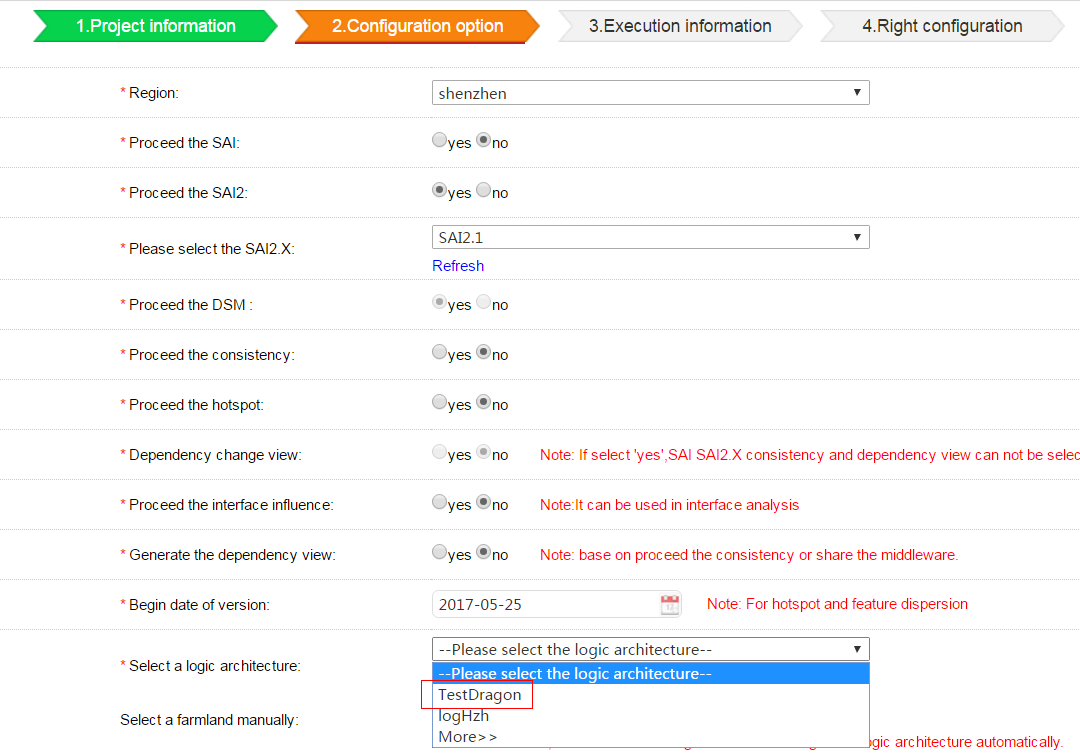
保存配置



（4）回到工程配置窗口，选择“Refresh”



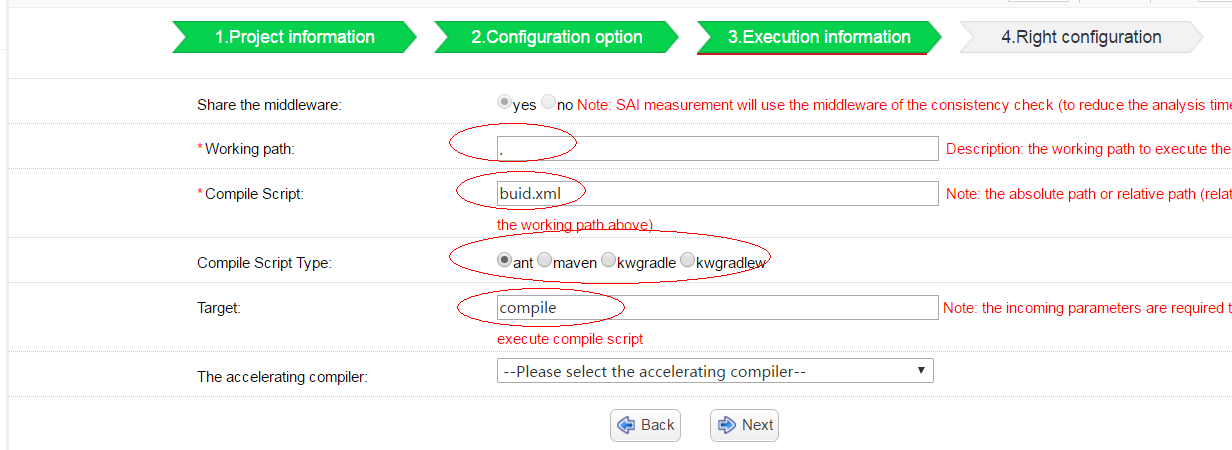
选择（3）中创建的逻辑结构名



下一步配置工作路径和指定编译脚本

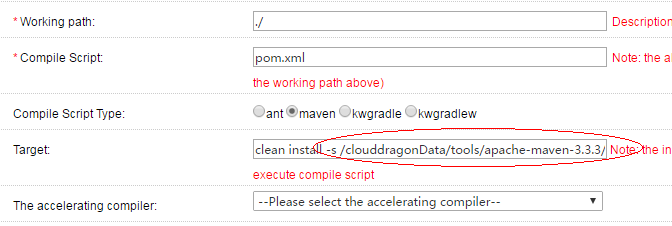
工作路径是指编译脚本所在路径相对于jenkins中配置的projectPath路径，./ 为当前路径， ../为上一级路径，编译脚本配置工程的编译脚本名称。注意：UADP新环境依赖编译，路径及编译脚本配置错误会导致构建失败，并需先确认编译能通过。

选择脚本类型

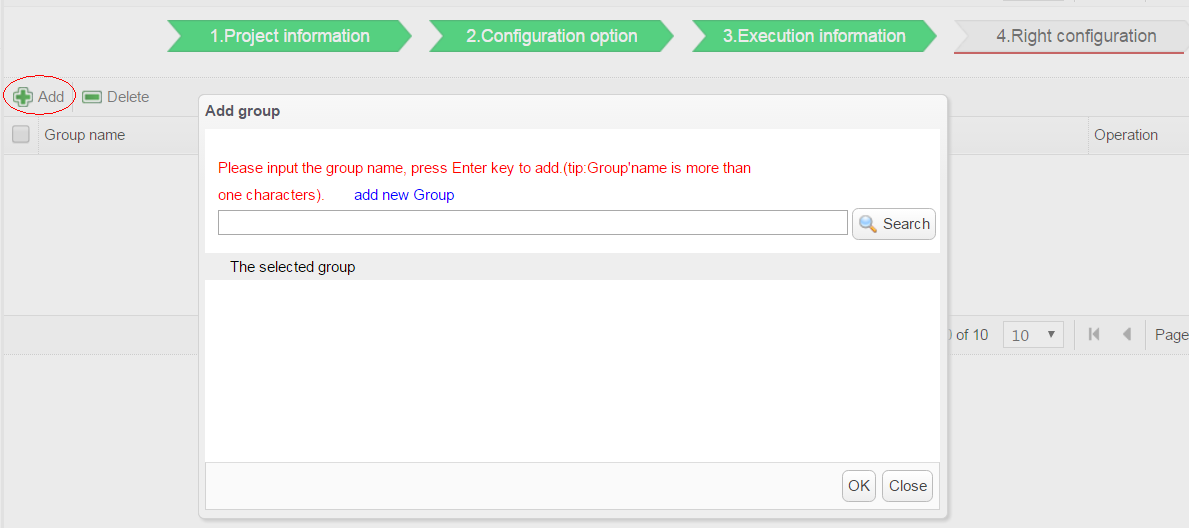


**说明：**

1. 针对maven项目，由于目前仍是使用云服务的私服，因此在Target参数中需添加-s /clouddragonData/tools/apache-maven-3.3.3/conf/settings\_ConsumerCloud.xml 参数信息，如



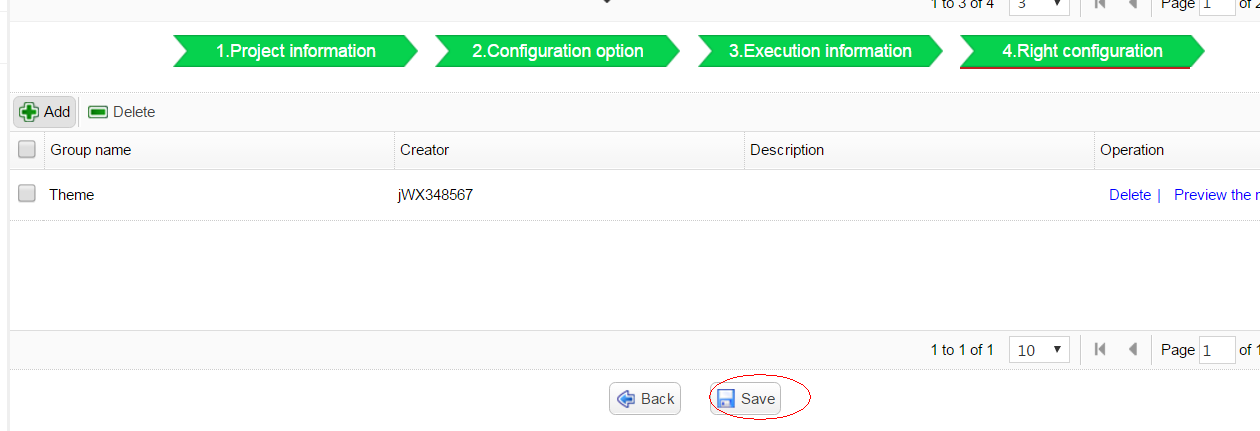
Next ，点击Add ，选择对应的组







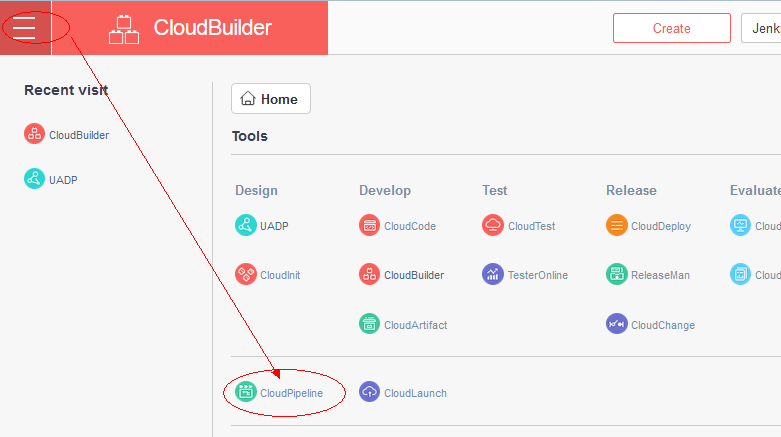
保存配置



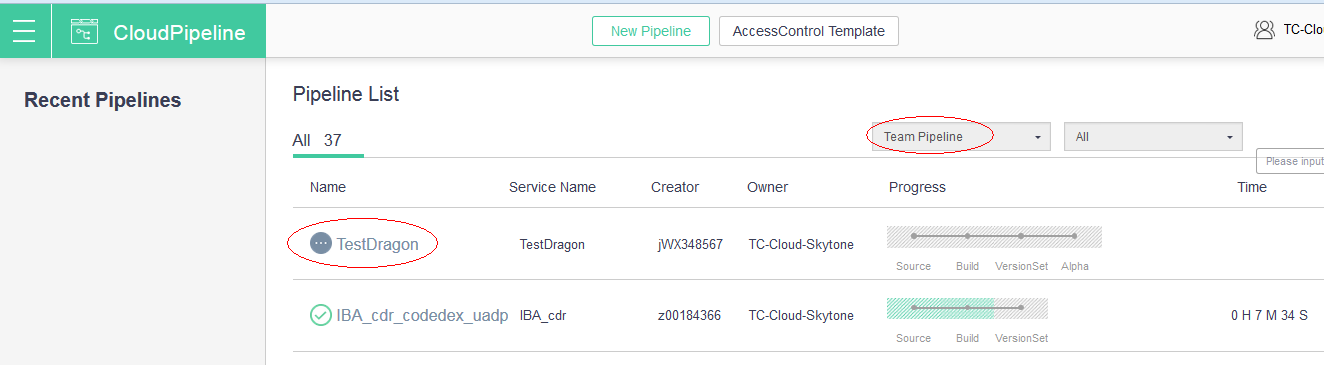
**3 流水线（Pipeline）配置**

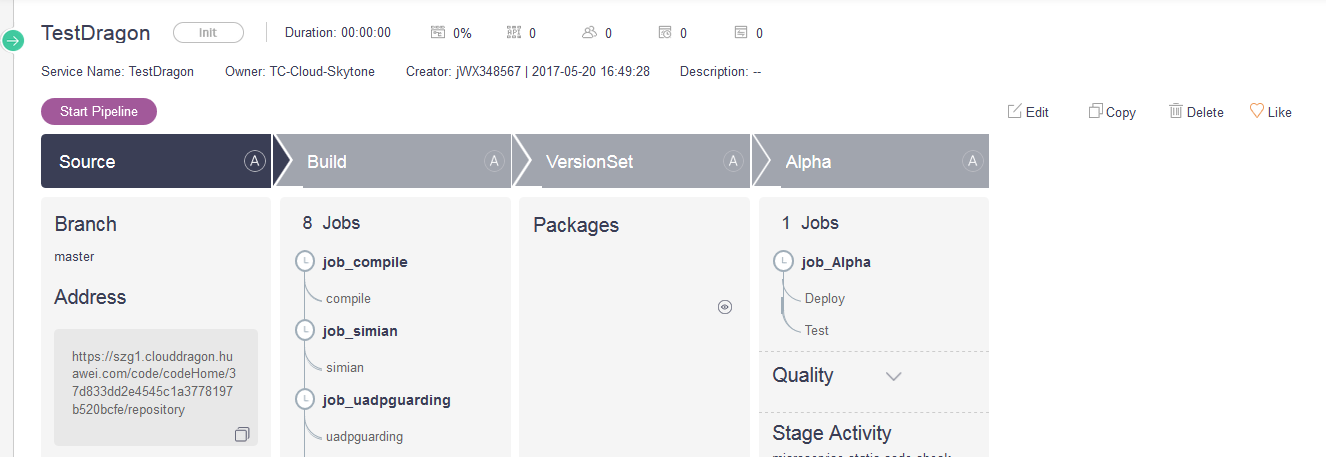
**3.1 进入流水线编辑界面**

主菜单中选择CloudPipeline 进入pileline主界面，点击对应的流水线

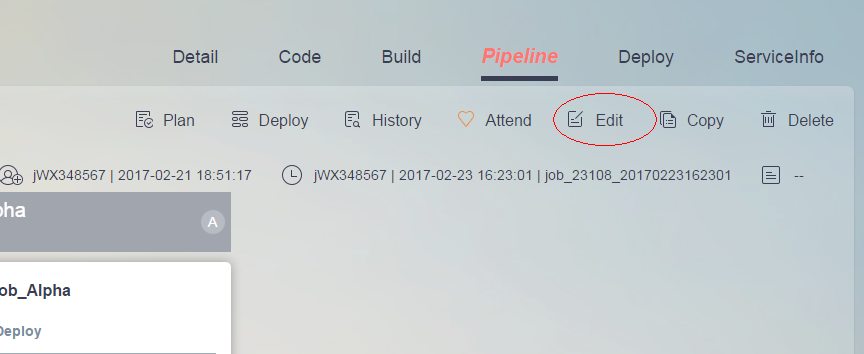


找到对应的微服务

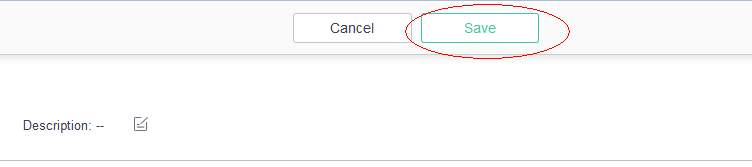




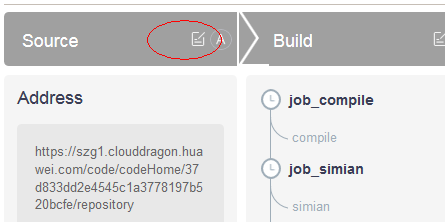
点击“Edit”来编译流水线



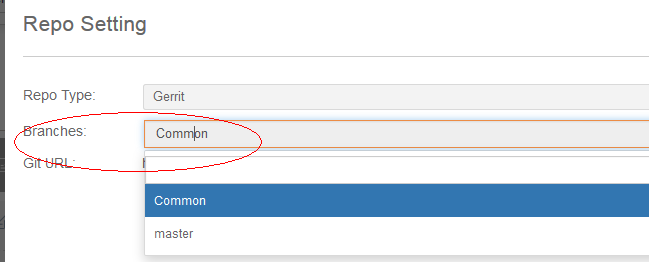
**注意：**在pipleline中各任务信息都配置完后，一定要点击右上角的Save 来保存所有的修改配置，否则所有的修改都不会生效



**3.2 修改分支信息**

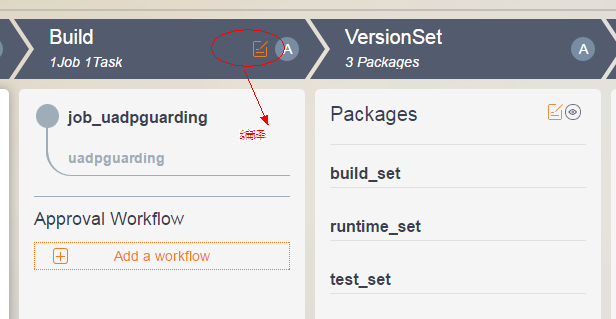


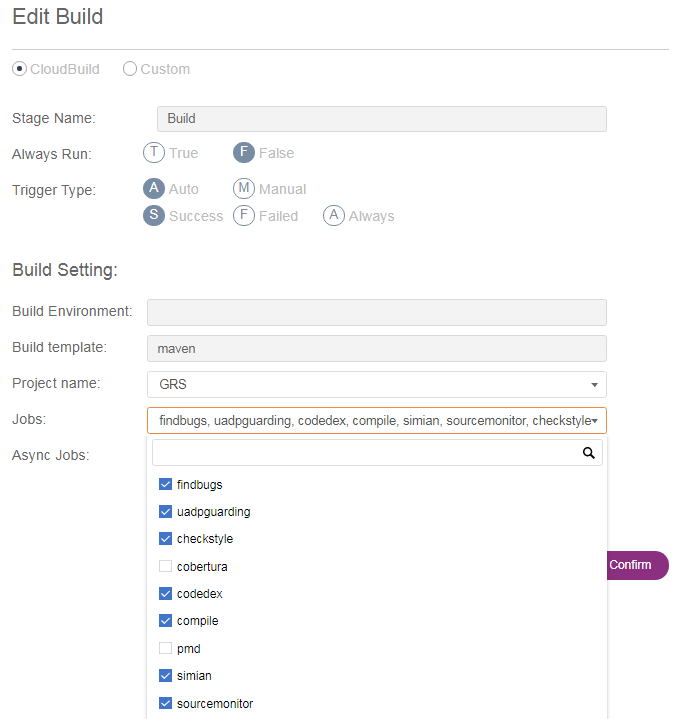
选择分支



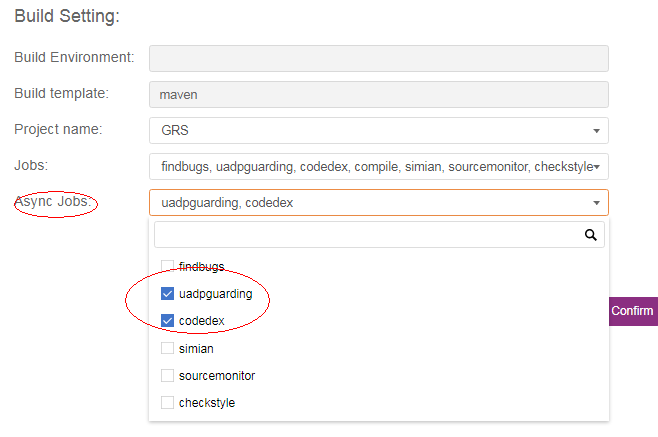
**3.3 修改Build中的任务列表**

**在pipeline中Build中完成代码的编译和静态检查**





由于CODEDEX和UADP执行时间较长，为了在后面发布时能够获取到两个质量指标数据，需将这两个检查与findbugs等检查放在一条流水线中，但为了不影响构建，将这两个任务设置为异步任务，即在开始执行后，触发这两个任务让在后台执行，但其执行不影响整个的构建和结果



**3.4 自动部署和测试配置**

按照模板创建的为pipeline中，都会有部署和测试任务，如果当前的模块没有测试任务，可以删除pipeline中的部署和测试任务



* + 1. **自动部署配置**

**3.4.1.1 打包和云眼系统配置**

绿区环境：

自动化部署系统：<http://100.100.147.139:8082>

配置中心：<http://10.21.153.136:18080>

廊坊环境：

自动化部署系统：<http://10.21.0.207:8082>

配置中心：<http://10.23.0.210:18080/>

打包和云眼系统的配置，请详阅“云服务中云眼自动部署配置指导”



**注：**对于**廊坊的部署**，在执行此操作之前，需**先执行VMP传包操作**，否则云眼不能获取到对应的软件包

Alpha --对应冒烟测试

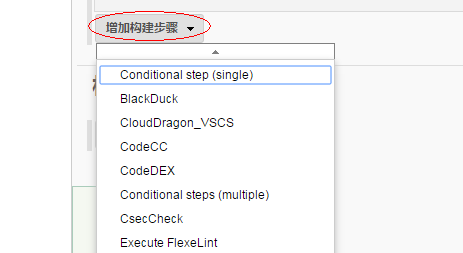
Beta --对应回归测试

Gamma –对应SDV测试

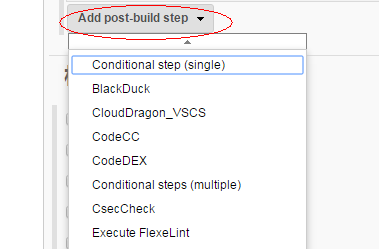
**3.4.1.2 云龙配置 -编译配置修改**

在原先的编译任务基础上添加如下配置，看考2.3 的指导进入编译界面

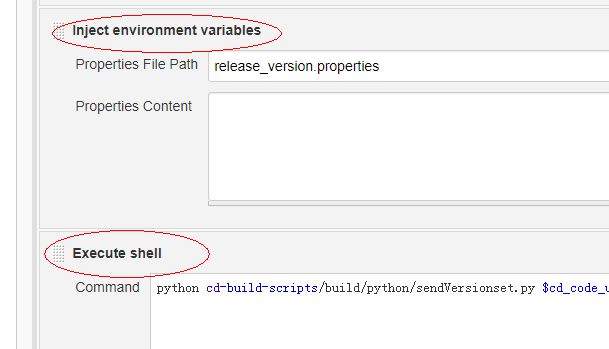
添加任务入库（Ant）



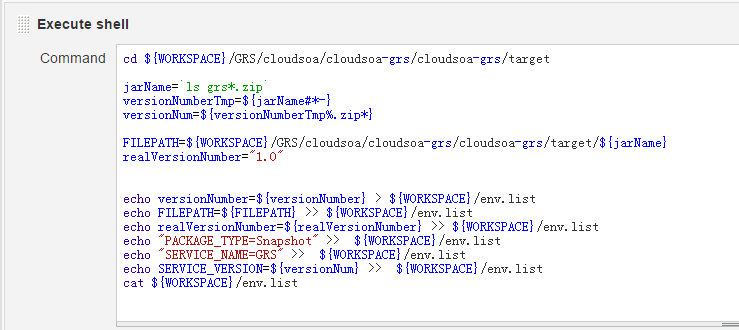
添加任务入库（Maven）



在编译任务的这两个任务中间添加如下3个任务



**任务1：**



cd ${WORKSPACE}/**GRS/cloudsoa/cloudsoa-grs/cloudsoa-grs/target**

jarName=`ls **grs\*.zip**`

versionNumberTmp=${jarName#\*\_}

versionNum=${versionNumberTmp%.zip\*}

FILEPATH=${WORKSPACE}/**GRS/cloudsoa/cloudsoa-grs/cloudsoa-grs/target**/${jarName}

realVersionNumber="1.0"

echo versionNumber=${versionNumber} > ${WORKSPACE}/env.list

echo FILEPATH=${FILEPATH} >> ${WORKSPACE}/env.list

echo realVersionNumber=${realVersionNumber} >> ${WORKSPACE}/env.list

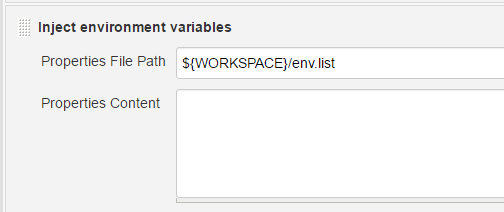
echo "PACKAGE\_TYPE=Snapshot" >> ${WORKSPACE}/env.list

echo "SERVICE\_NAME=**GRS**" >> ${WORKSPACE}/env.list

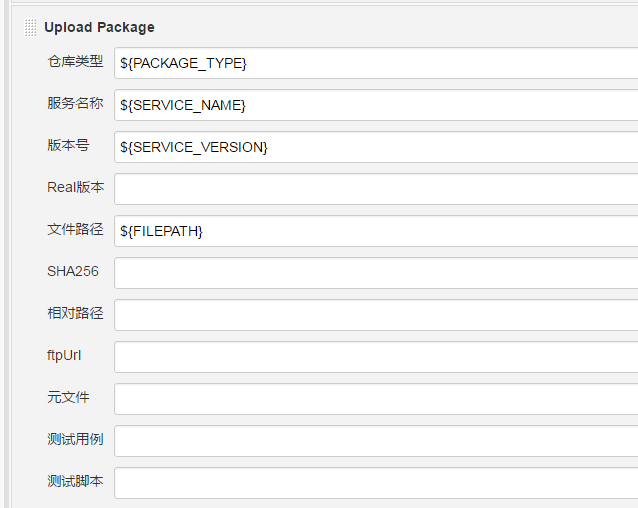
echo SERVICE\_VERSION=${versionNum} >> ${WORKSPACE}/env.list

**注：修改软件包路径和关键字信息**

**任务2：**



**任务3：**



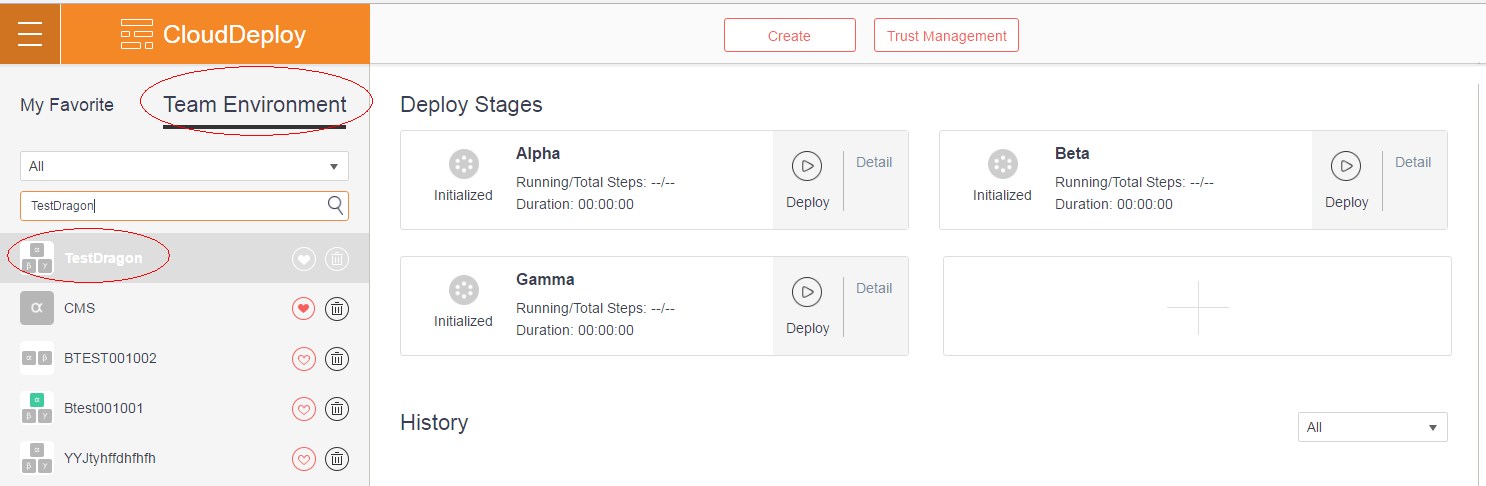
**编译任务配置好后，构建一下看是否OK,OK之后进行后面的配置**

**3.4.1.3 云龙配置–部署配置修改**

在菜单栏中选择CloudDeploy

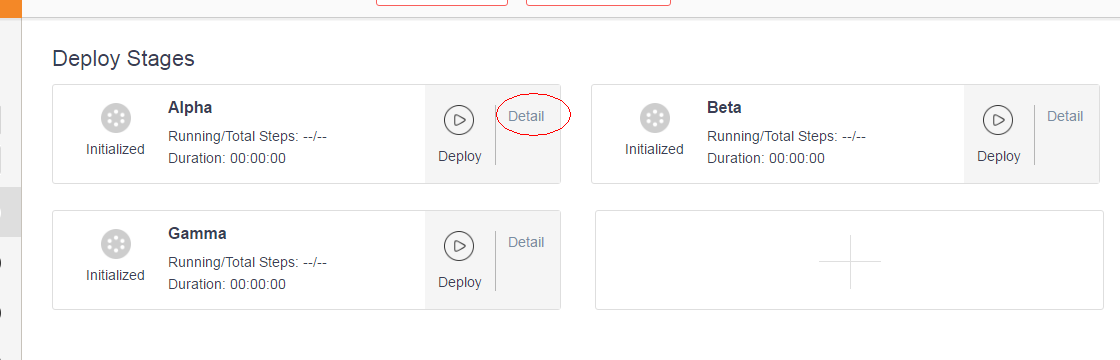


找到对应的微服务



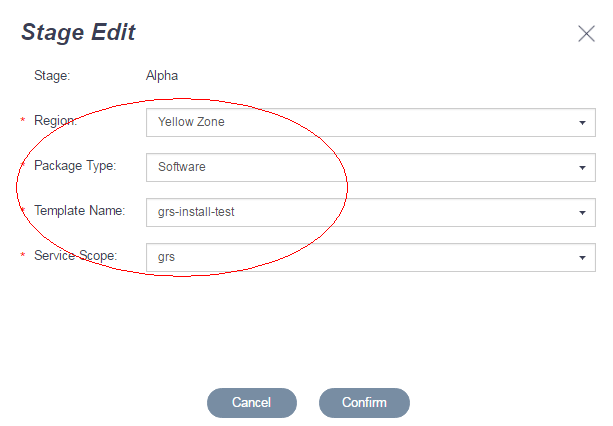
**Alpha 对应冒烟测试，Beta对应回归测试，Gamma对应SDV测试**

选择对应的节点，点击”Detail”进行编译



配置云眼系统信息



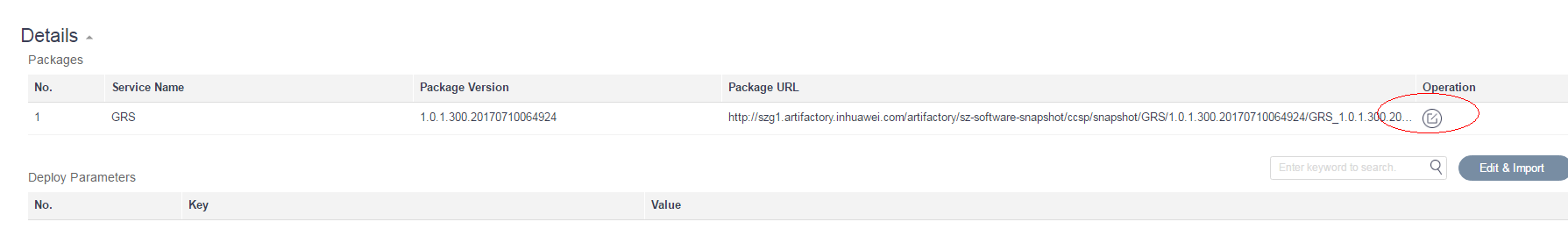


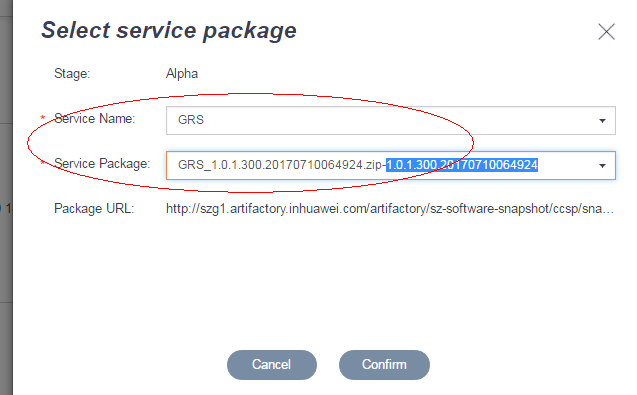
注：（1）**绿区环境中，Region 选择Yellow Zone, Cloudeye IP Port 配置100.100.147.139:8082**

**廊坊环境，Region 选择Internet Zone, Cloudeye IP Port 配置10.21.0.207:8082**

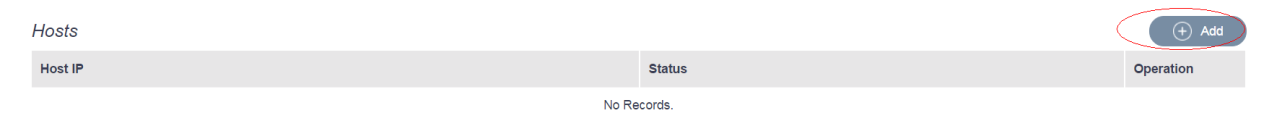
1. Template Name选择在云眼环境中创建的模板
2. Service Scope选择在云眼中对应的领域

选择微服务名和软件包





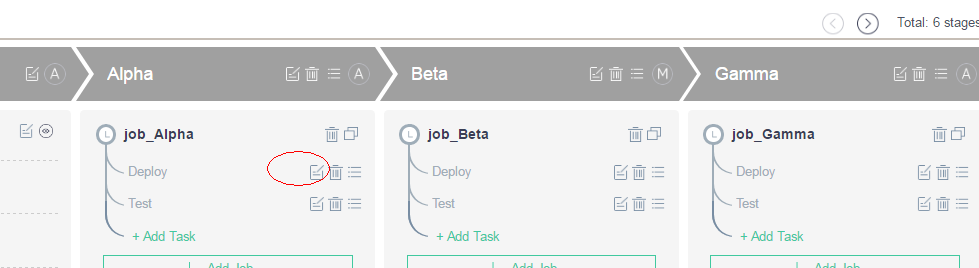
选择在云眼系统中注册的主机IP



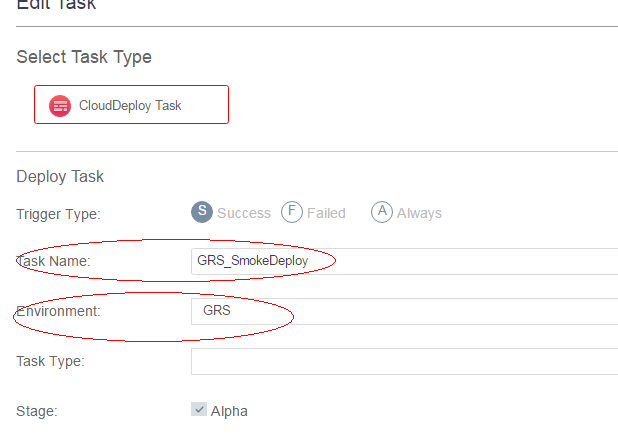


**3.4.1.4 云龙配置–流水线修改**

进入流水线编译页面，编译相应阶段的部署任务



修改任务名和选择Environment，Environment 值为微服务

****

修改好后保存配置

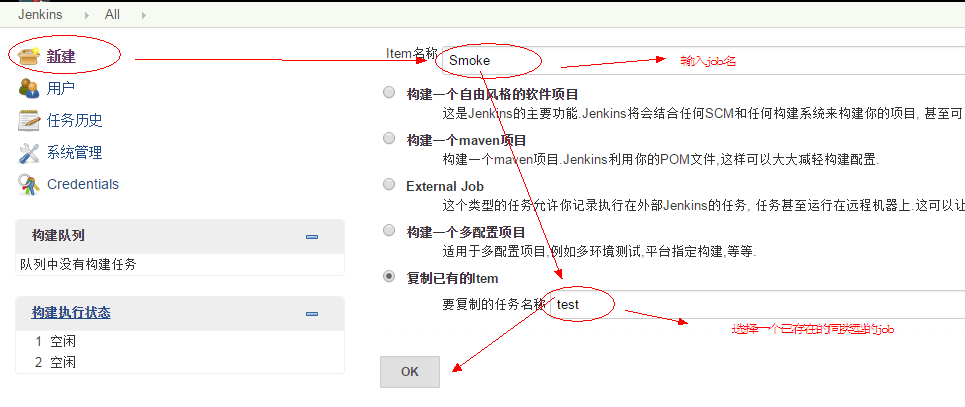
**3.4.2 自动化测试配置**

**3.4.2.1 在Jenkins中添加测试**

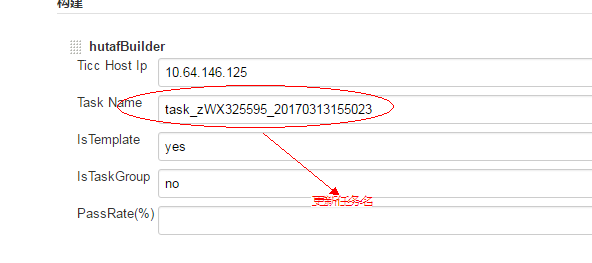
需提前在下面的jenkins中添加一个自动化测试job

http://100.101.31.88:8080/jenkins

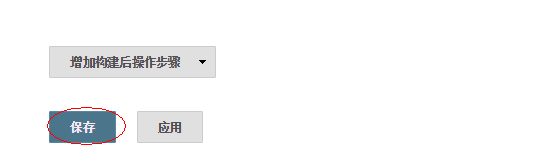
登录之后，新建一个测试job



更新hutaf 任务的task name信息



保存配置



**说明：**

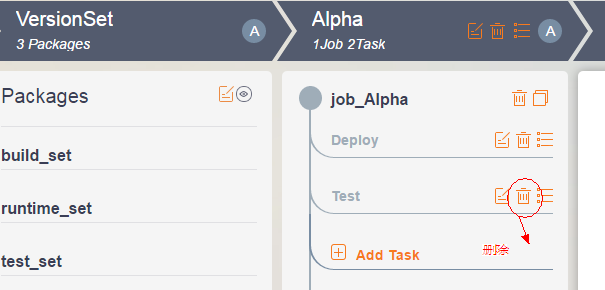
TICC绿区：10.64.146.125

TICC廊坊：100.101.11.152

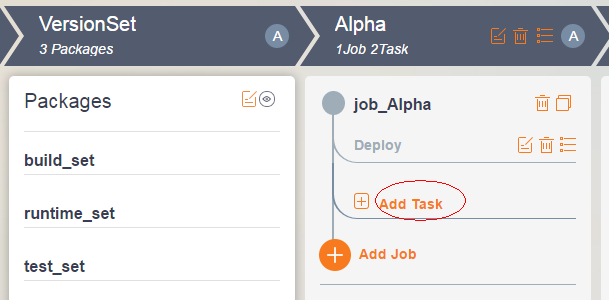
**3.4.2.2 流水线修改**

创建微服务中，由于测试的task type默认是CloudTask,而咱需要使用的Jenkins task，因此需要删除已存在的test 任务，然后新建

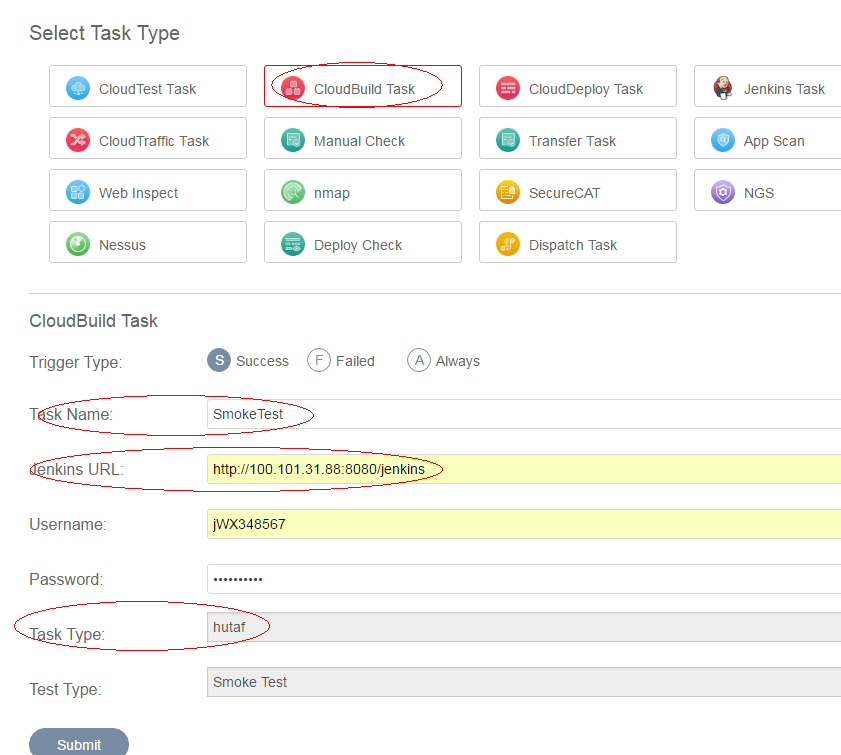
在编译pipeline页面



选择Add Task



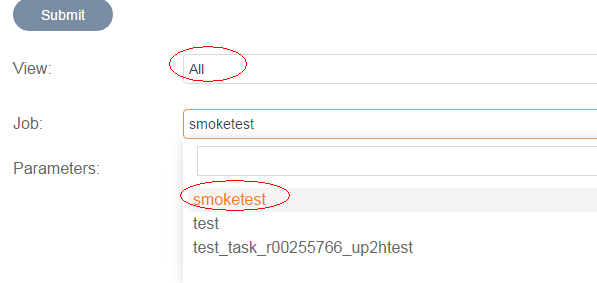
**Task Type选择CloudBuild Task**，输入测试任务名、jenkins地址 和账号信息之后，点击“Submit”



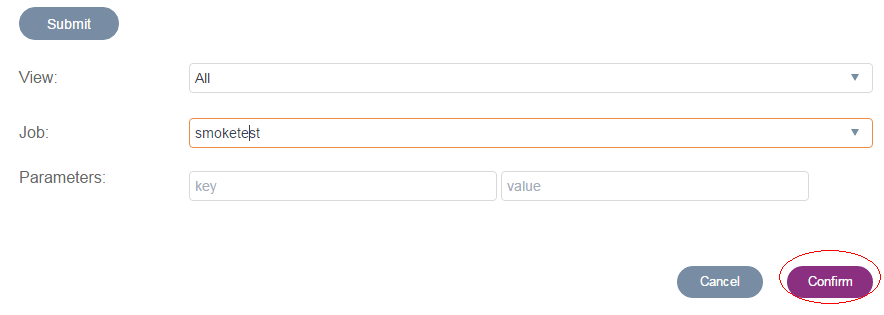
**说明：**

1. **Jenkins地址是**[**http://100.101.31.88:8080/jenkins/**](http://100.101.31.88:8080/jenkins/)
2. **Task Type选择hutaf**
3. **Test Type根据实际情况选择**

在输入的信息提交后，需要在jenkins中添加的测试job，其中View选择All



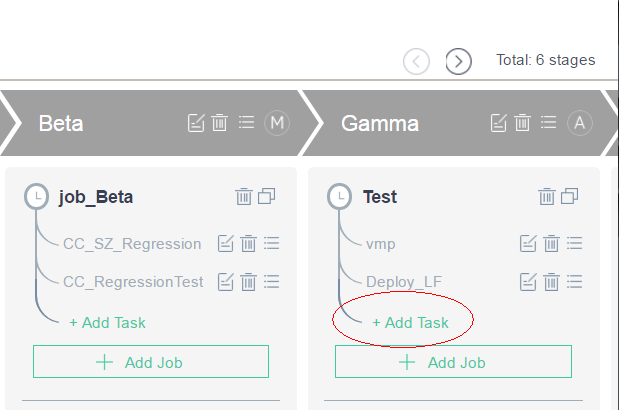
保存配置



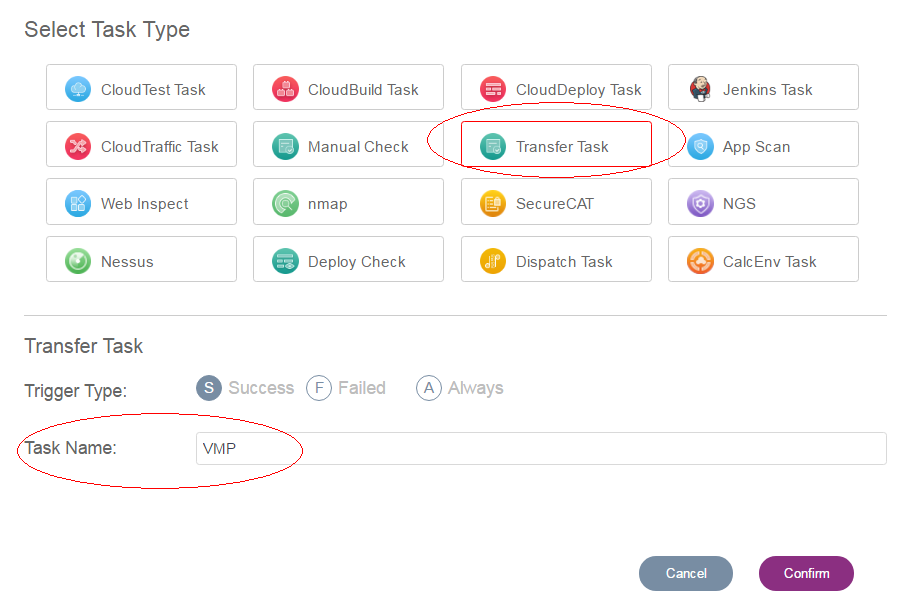
**注： 如果不需要参数，将Parameters 删除**

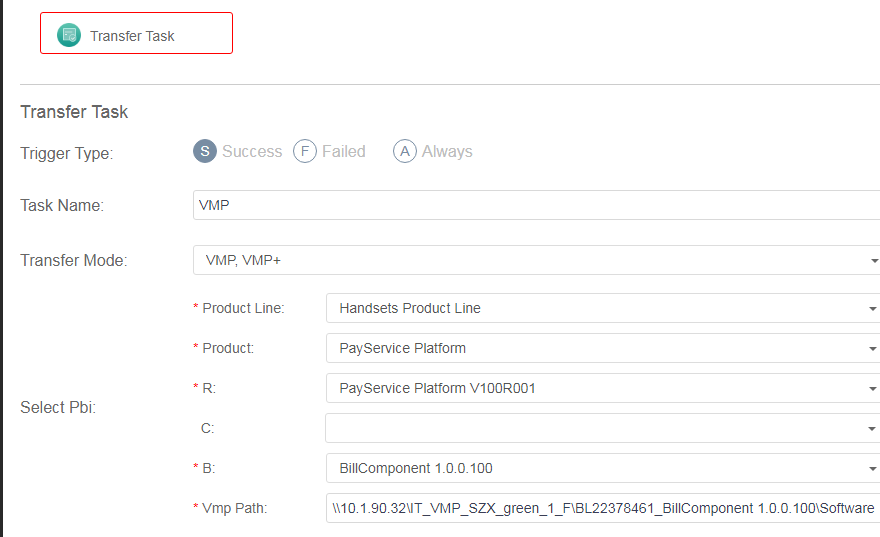
**3.5 VMP传包配置**

编译流水线，在对应的测试节点选择添加任务



选择Transfer Task ，输入任务名，保存





**说明：**

**（1） VMP路径的服务器需要IP,暂时不支持域名的格式，如**

**\\10.1.90.32\IT\_VMP\_SZX\_green\_2\_F\BL22438026\_GRS 1.0.1.103\Software**

1. 软件包上传到VMP的Software目录中
2. **当前对传包有限制，对同一个版本的软件包，如果已经上传过，再上传就会报错，所以需保证每次是新包**
3. **Transfer Mode不能为空，VMP和VMP+都选上**
4. **在正式发布的时候，在构建之前重新配置VMP路径信息**

**3.6 特性流水线（CODEDEX/UADP）**

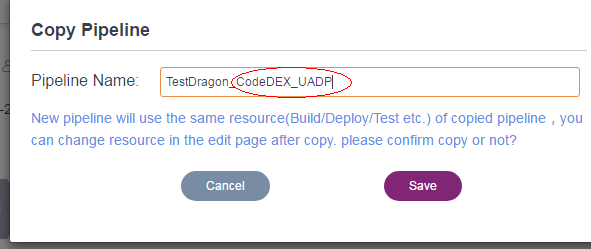
**由于将CODEDEX、UADP放在主流水线（compile、findbugs等）中异步执行了，此特性流水线可以不用创建**。如果想单独看这两个数据的话，可以按照下面的流水线来执行

操作流程如下

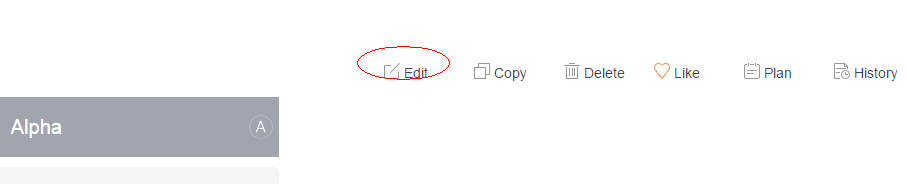
复制原先的流水线

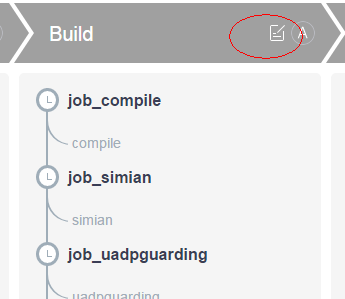


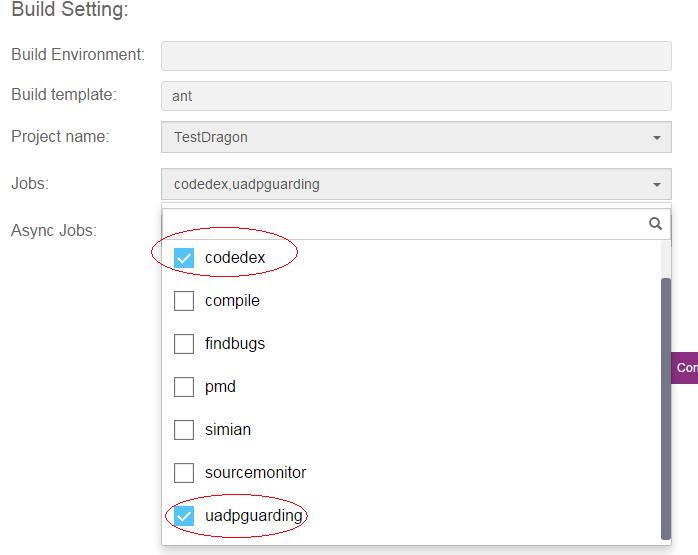
输入流水线名



编译配置

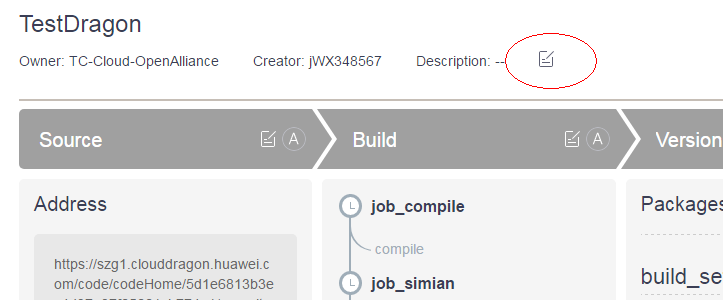




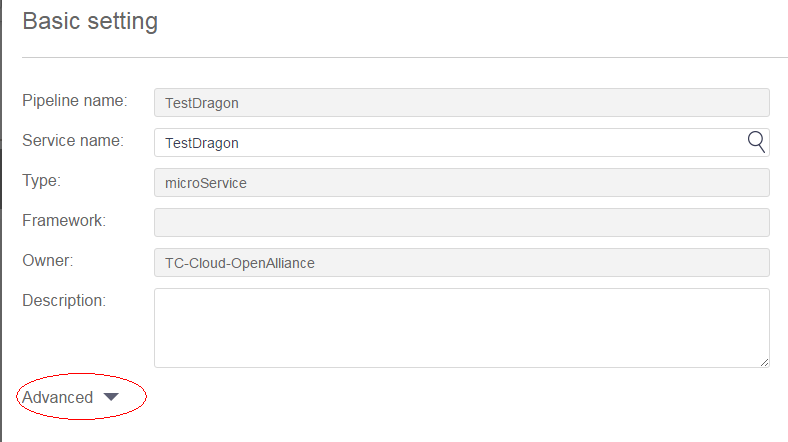


**3.6 发送邮件设置**

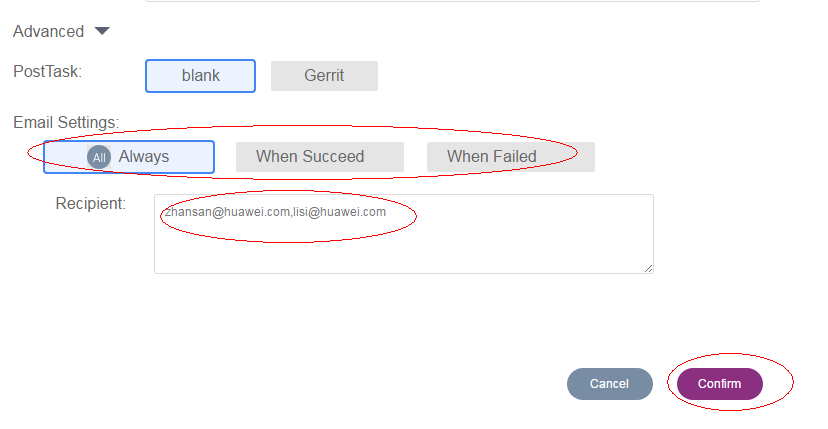
进入流水线编辑页面，然后点击下图的图标



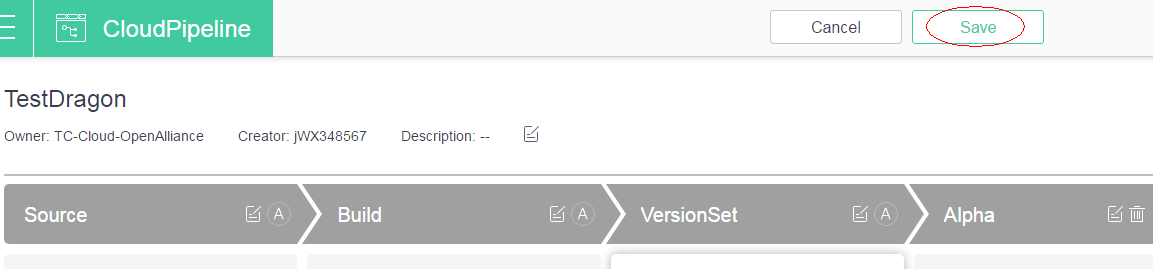
打开高级设置



选择发送的条件和添加人员名单列表，多个之间用逗号分隔



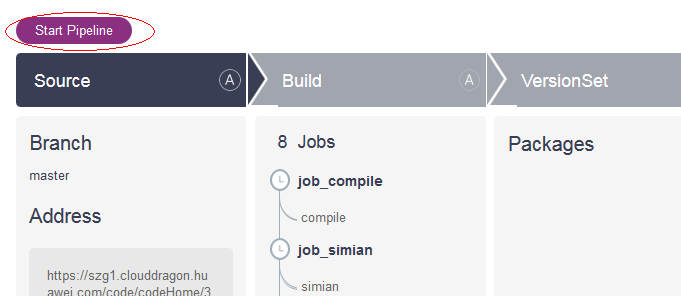
确认后保存配置

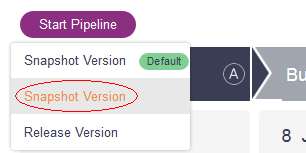


**3.8 Pipeline构建**

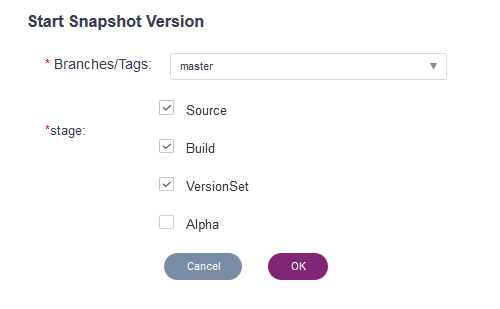
**3.8.1 手动构建**

点击Start启动流水线构建



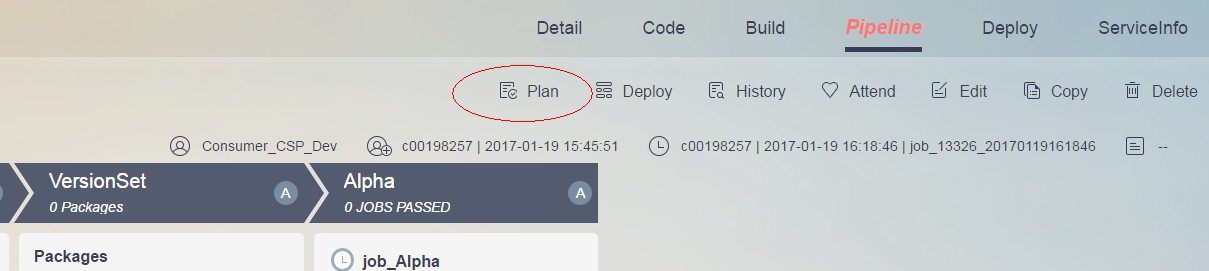


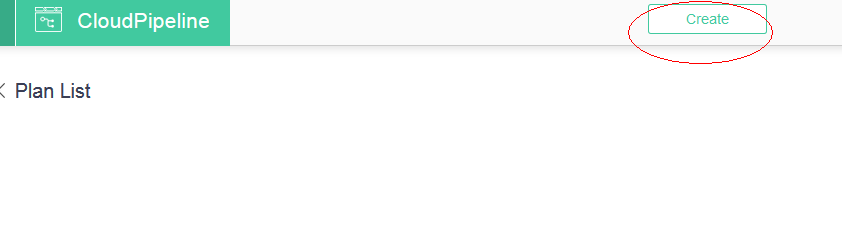
选择代码分支和本次构建需要执行的任务，如果哪个任务不执行单击变成灰色即可



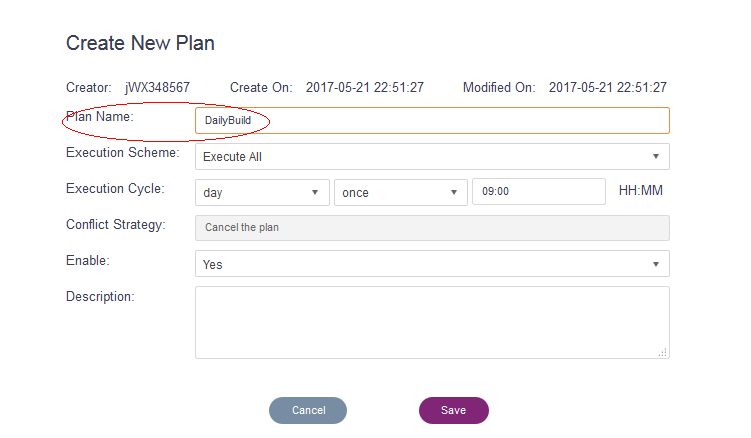
**3.8.2 定时构建**

制定定时计划

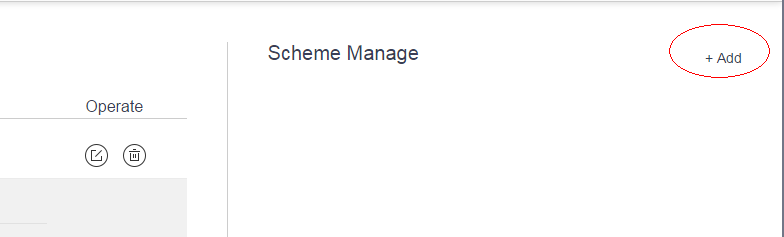


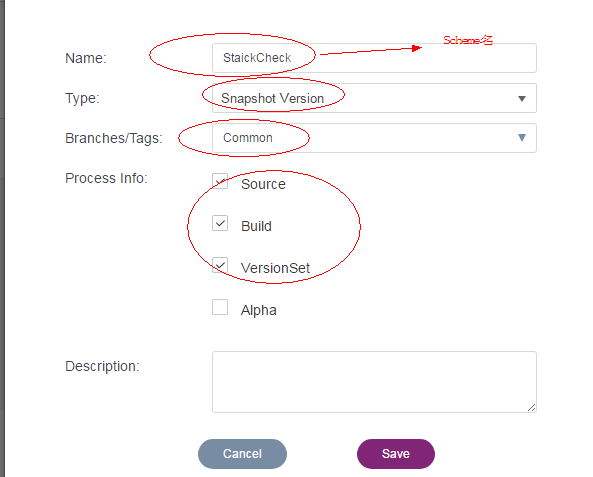


输入定时配置信息

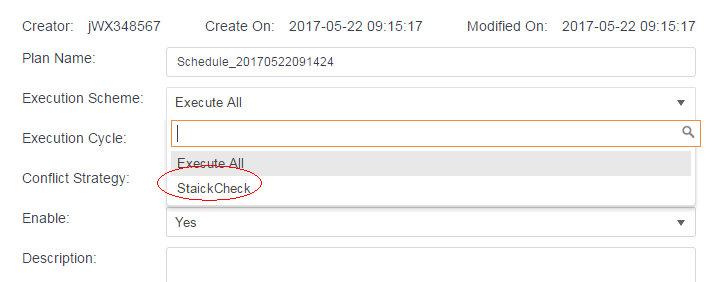


默认是执行流水线的所有任务，如果不是话，可以定义自动以任务列表





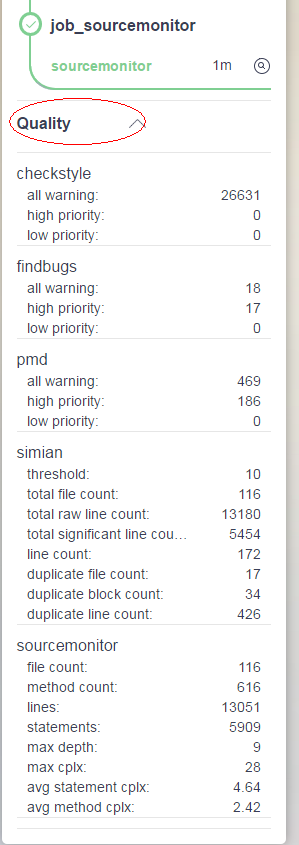
保存后在计划中就可以选择新建的scheme

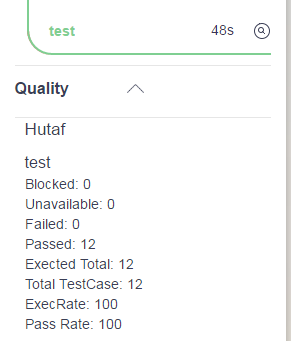


**3.9 报告和日志查看**

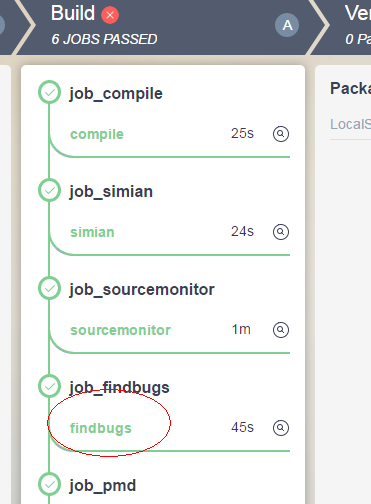
**3.9.1 报告查看**

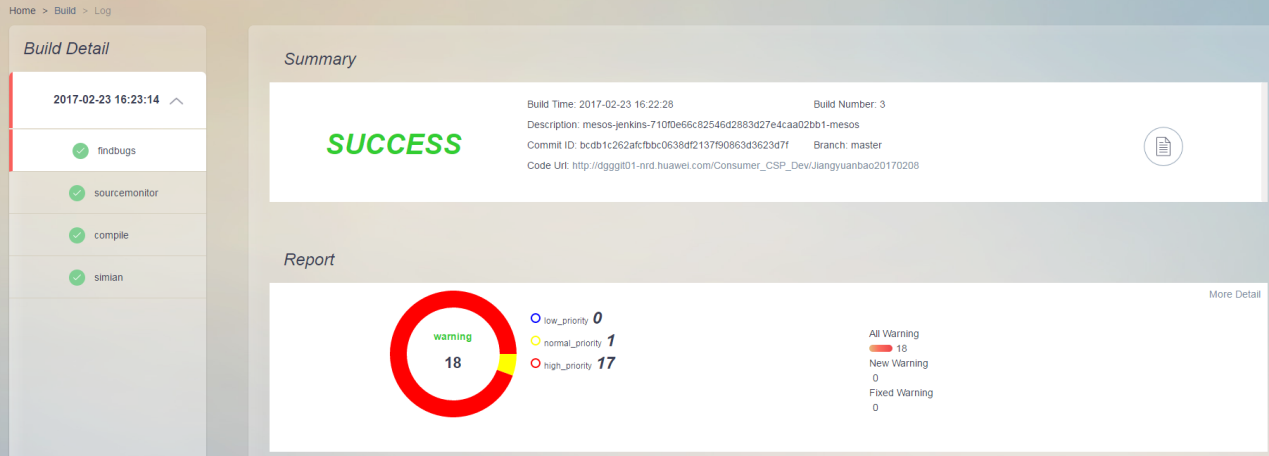
（1）点击Build 或Alpha 下的Quality 可以查看对应的任务报告





（2） 点击任务链接



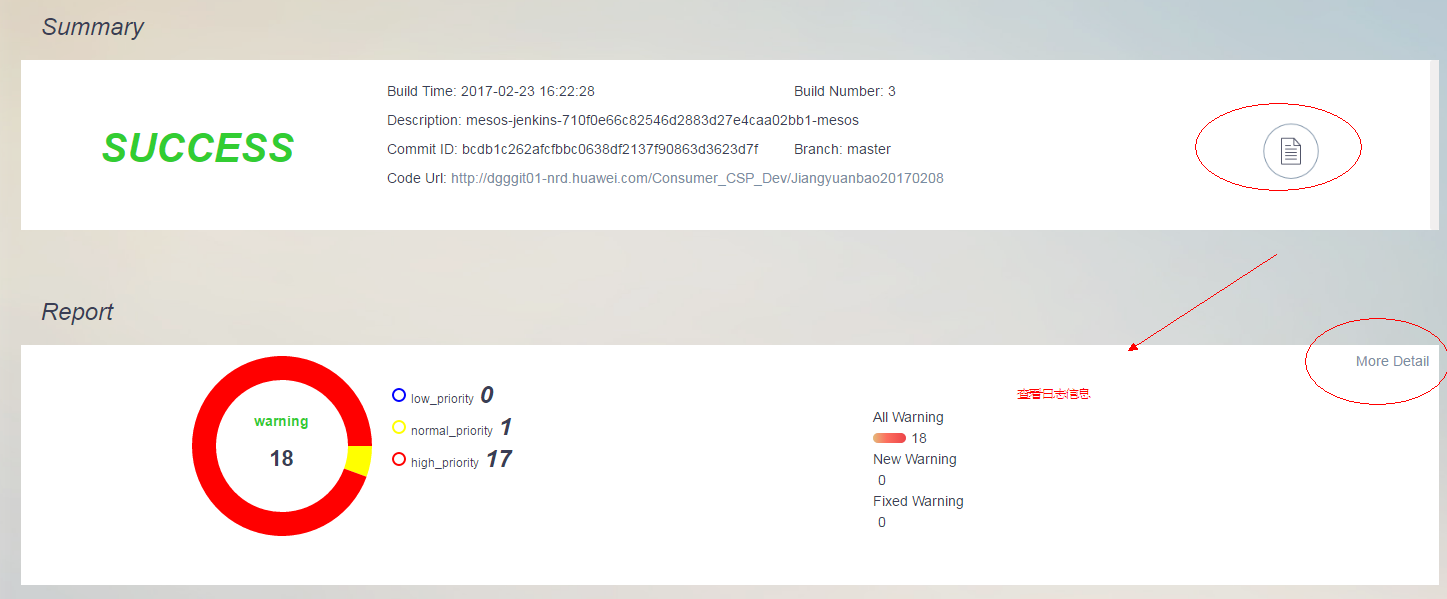


**3.9.2 日志查看**

点击任务的此图标查看日志

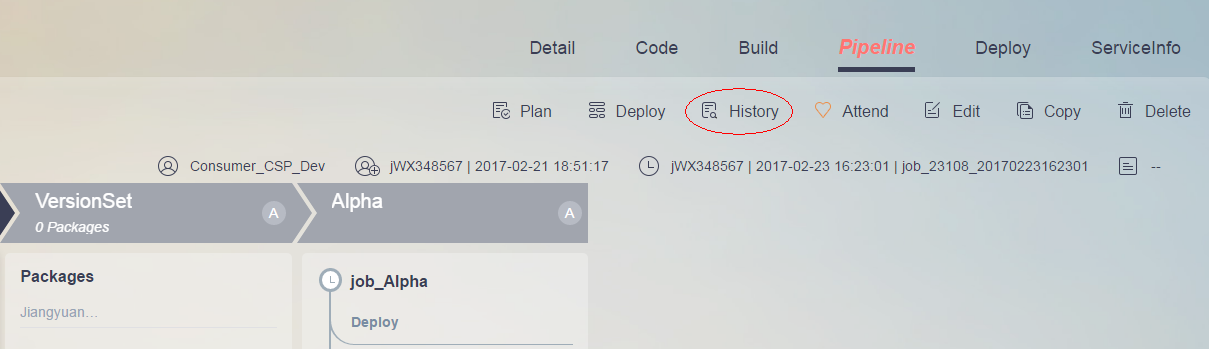


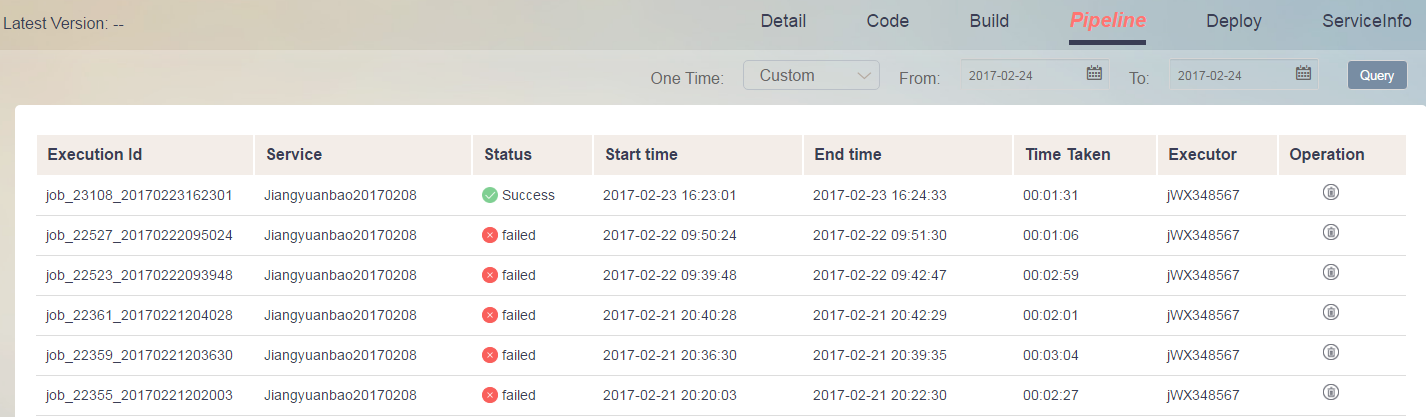
在报告页面可以查看日志信息，点击More Detail 可以到jenkins中查看详细日志



**3.10 历史构建查询**

在流水线界面，点击“History” 可以查看构建记录



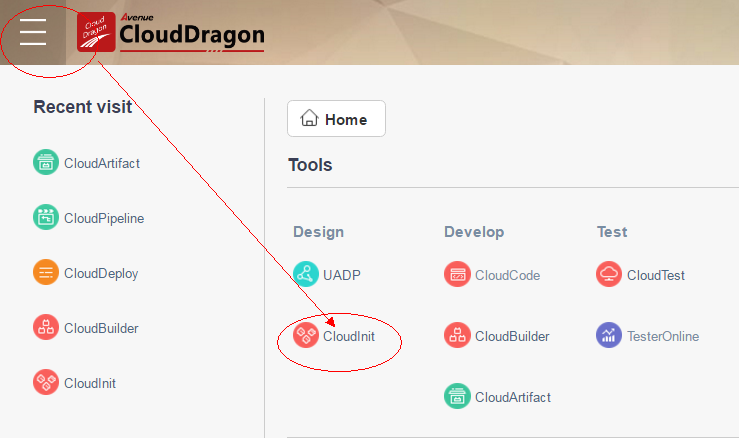


**4 组装服务**

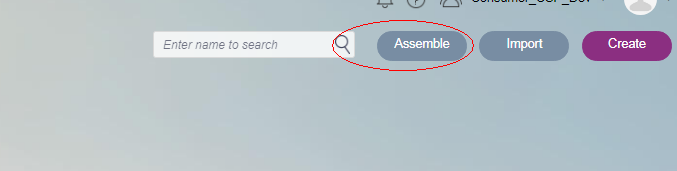
组装服务主要是针对自动化测试需要多个微服务都部署OK才进行自动化测试的场景

**4.1 创建组装服务**

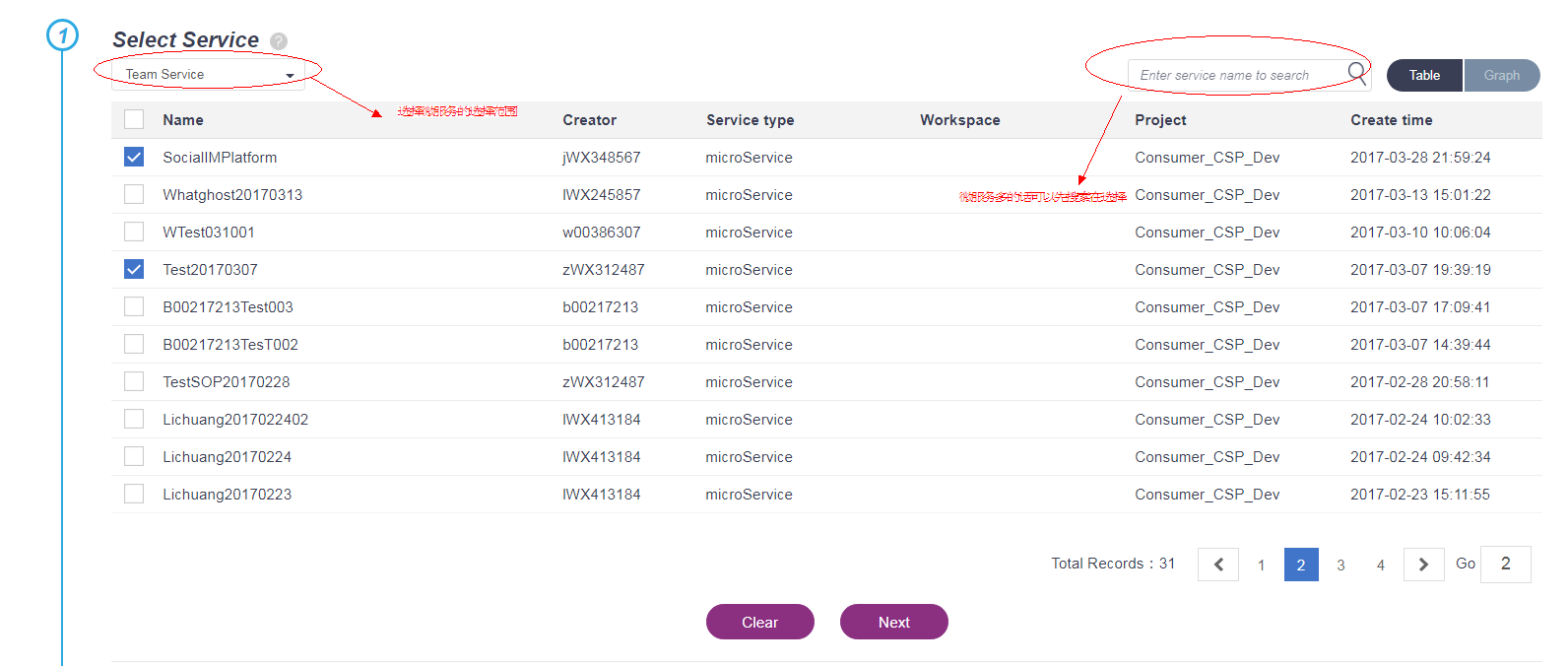
Toolchain 🡪 CloudInit 进入为服务器创建页面



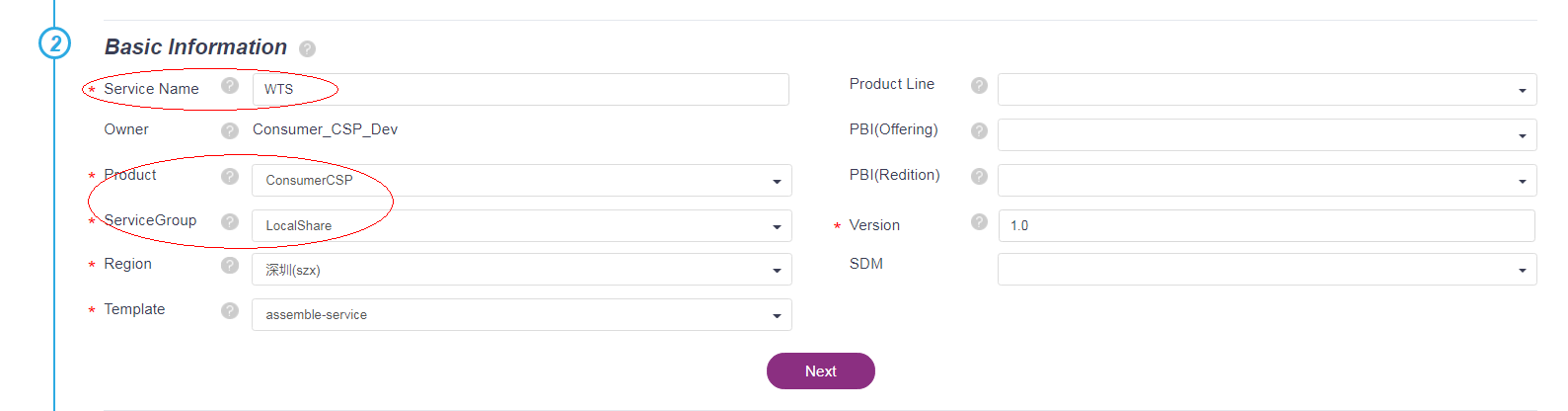
**选择“Assembel”方式**来创组装服务



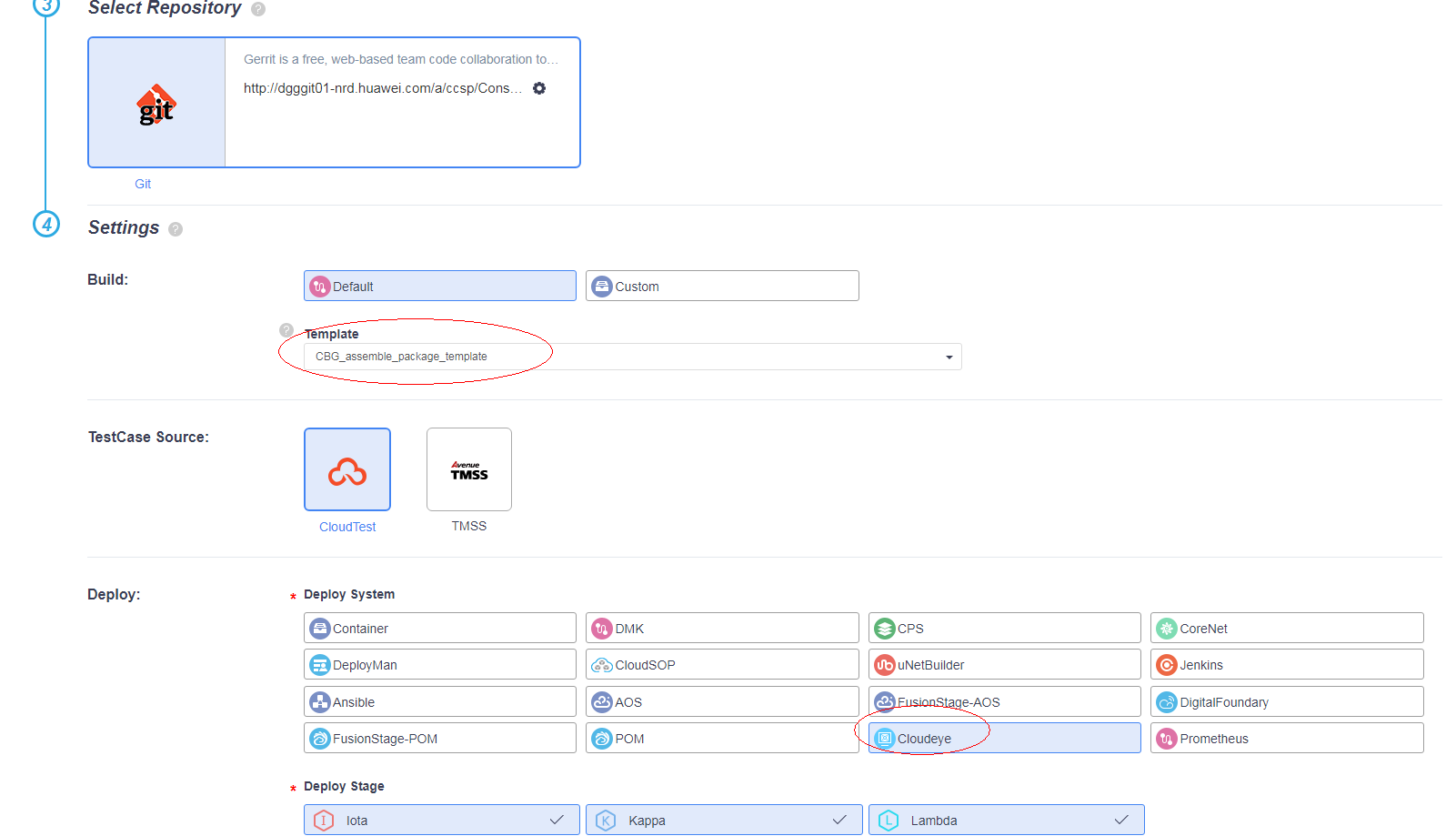
选择需要组装的微服务



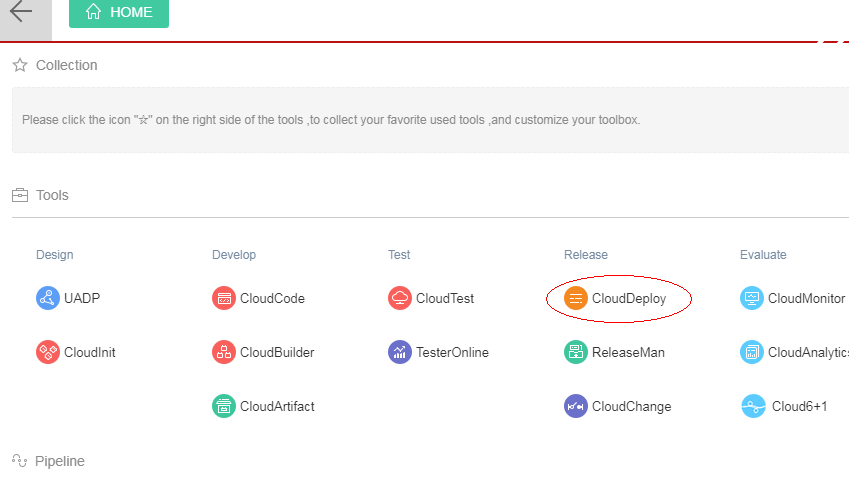
输入组装服务名、选择产品和ServiceGroup



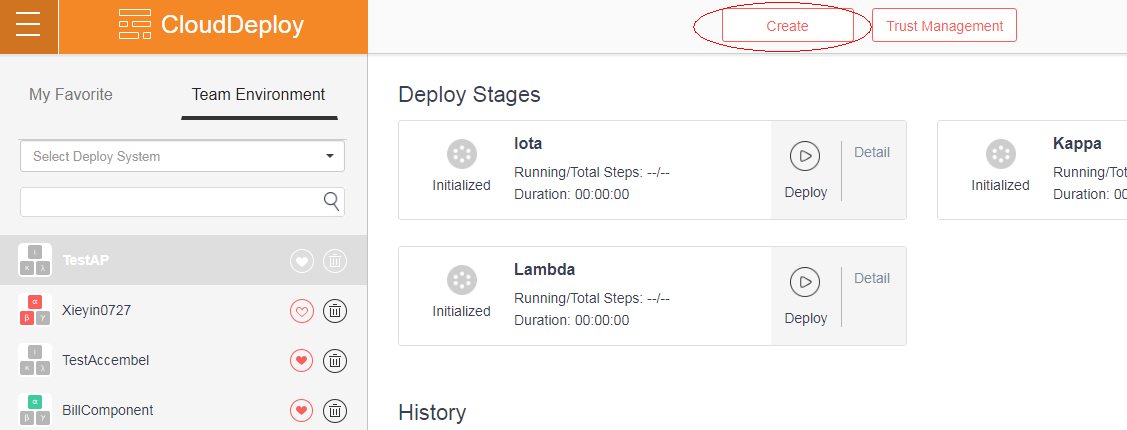
Build模板选择CBG\_assemble\_package\_template



**4.2 创建部署环境**



**组装服务下的每个微服务创建一套部署环境**

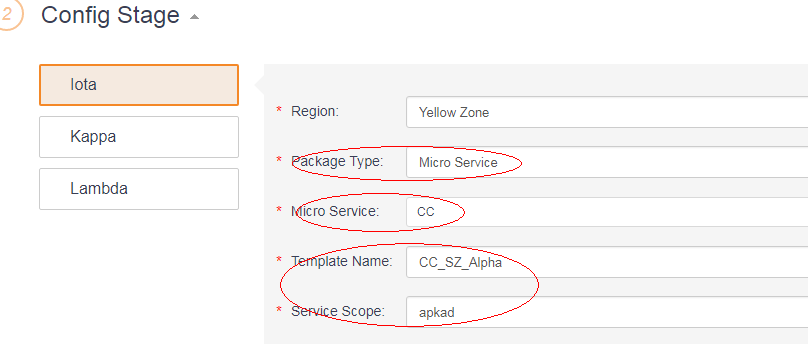




注： （1）**Service Modal选择Assemble Service**

（2）Name就输入“组装服务\_微服务”的信息，作为后面部署的Environment信息；

Service 选择组装服务



**注：**

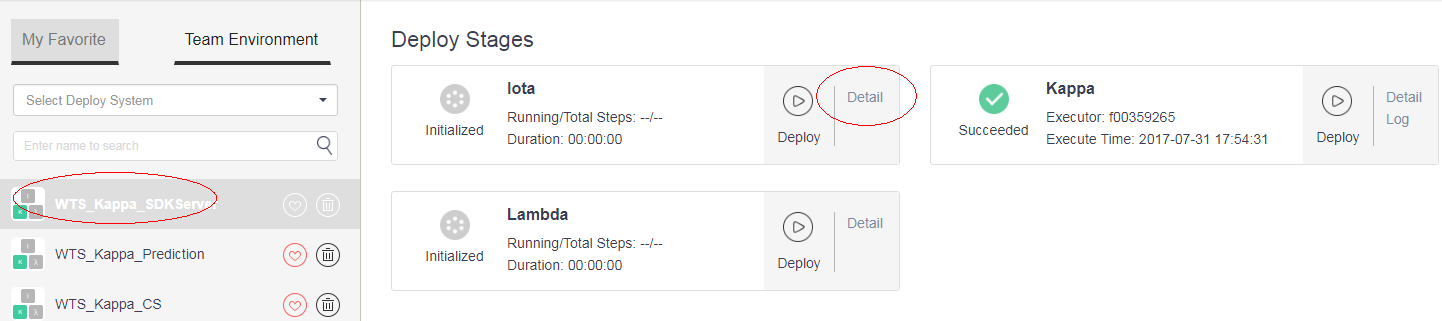
（1）Region，绿区环境选择Yellow Zone, 廊坊环境选择Internet Zone

（2）Package Type 选择Micro Service

（3）Micro Service 就输入组装的微服务

（4）Template Name 和Service Scope就选择微服务在云眼中对应的信息

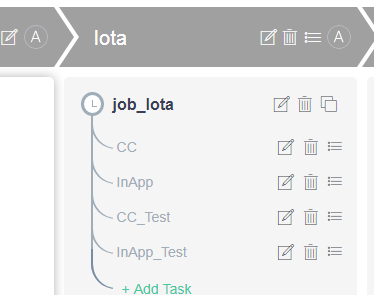
保存后，选择此微服务在各应阶段使用的测试服务器



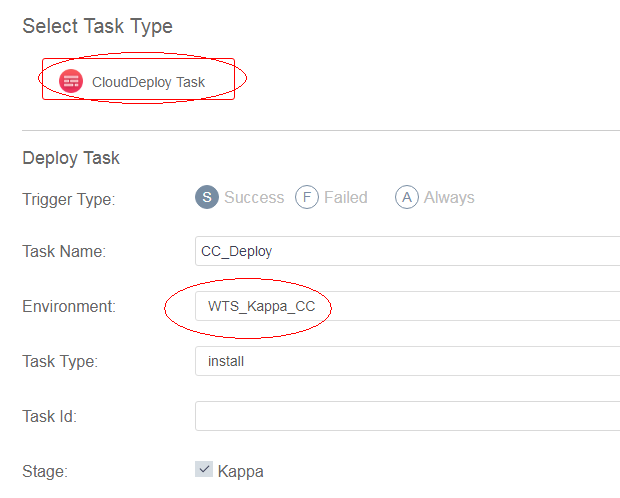


**4.3 组装流水线中配置**

在组装流水线中，在相应的阶段增加部署任务，每个微服务对应一个部署任务，可以根据依赖关系控制各微服务的执行顺序



其中Environment 就选择在前面创建的部署环境



测试任务添加请参考3.4.2

**5学习资料**

https://szxsvn04-ex:3690/svn/TC\_EMUI\_DOC\_SVN/07.安全与测试/06 项目管理/01 EMUI 6+1/02 参考资料/CloudDrogen

云龙视频资料

<http://3ms.huawei.com/hi/group/3015093/wiki_4458779.html?for_statistic_from=all_group_wiki>

云龙WIKI:

<http://3ms.huawei.com/hi/group/1005981/threads.html#category=1452895>