

Lab1 GIT & DevCloud入门

2019年 软件工程课程系列实验

1. 实验概述

1.1. 实验目标

通过本次Lab，安装并学习Git的基本操作，在华为云平台上进行包括代码托管、代码修改、代码提交、编译、构建、测试在内的覆盖软件开发全生命周期的尝试。

1.2. 实验准备（前提）

- 熟悉掌握并安装版本控制系统Git
- 熟悉Java编程，了解Java Web
- 熟悉计算机操作系统，Linux或Windows

什么是版本控制？版本控制是一种记录一个或若干文件内容变化，以便将来查阅特定版本修订情况的系统。基于版本控制系统（VCS），你可以将某个文件（甚至整个项目）回退到过去某个时间的状态，比较每个修订版本之间的变化细节，查出是谁对哪个地方进行了修改，寻找由于某次修改引发的问题等等。

工具：

1. Maven：是一个项目管理工具，可以对Java项目进行构建、依赖管理。
2. Git：Git是一种分布式版本控制系统，可以用于软件项目的版本管理。
3. SpringBoot：SpringBoot 是一个轻量级框架，可用于快速获得一个正常运行的Spring 应用程序，例如Web应用。

参考资料：

1. Pro Git book: <https://git-scm.com/book/en/v2>
2. Git 教程：
<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000>

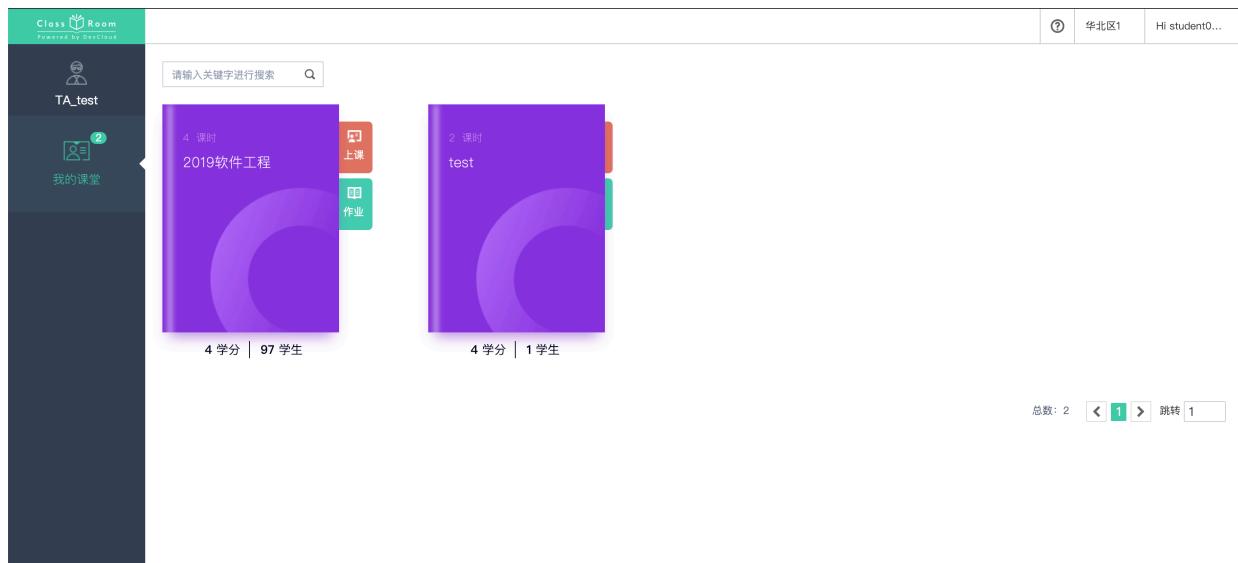
1.3. 实验要求

- 自学Git：安装Git，并熟悉Git常用操作；
- 根据实验内容，华为云平台操作，能够使用华为云独立开发和部署应用；
- 软件工程生命周期；
- 撰写实验报告，报告自己的实验过程，实验体会和认识。

2. 实验内容

2.1. 登录

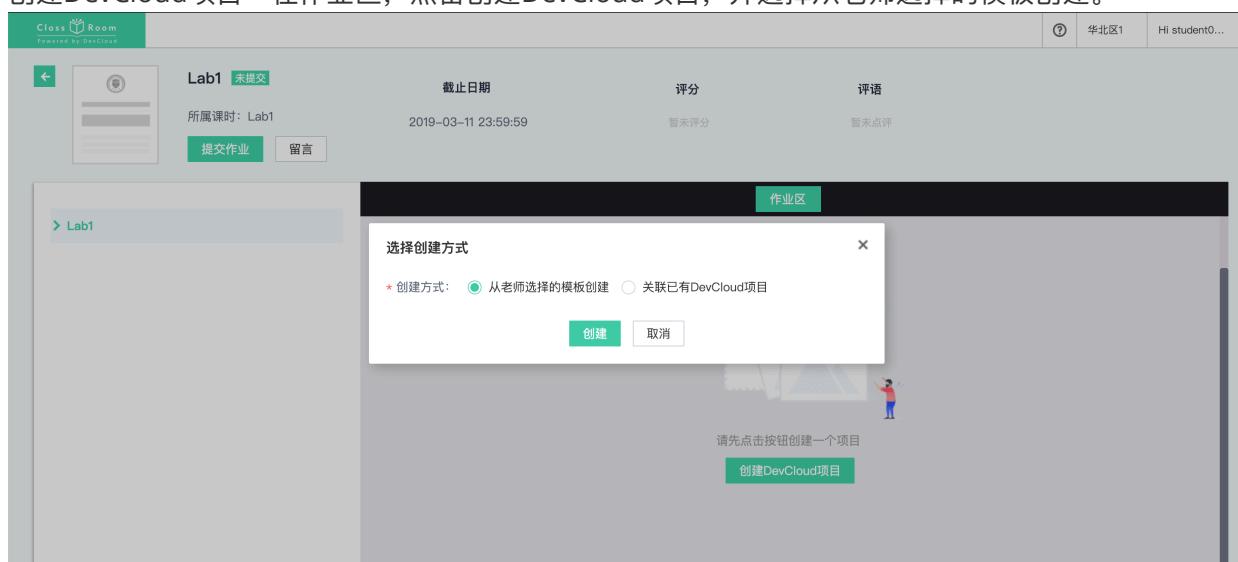
首先登录ClassRoom平台，选择IAM验证登录，根据分配的账户密码进行登录。进入课程-作业，点击软件工程课程-作业。



选择相应Lab（本次实验为Lab1），并点击进入作业。



创建DevCloud项目：在作业区，点击创建DevCloud项目，并选择从老师选择的模板创建。



(这里等待系统项目创建完成) 等待系统项目创建完成后，点击进入项目。

Class Room
Powered by DevCloud

Lab1 未提交

所属课时: Lab1 截止日期: 2019-03-11 23:59:59 评分: 暂未评分 评语: 暂未点评

提交作业 留言

作业区

进入项目 更新项目数据

姓名	工作项 (已完成/全部)	增加的代码行数	删除的代码行数	提交次数
TA_test	暂无工作项	0	0	0

2.2. DevCloud工作台

进入DevCloud工作台。

DEVCLOUD test_Lab1_小组student... 仪表盘 工作 代码 构建&发布 测试 Wiki 文档 设置 华北-北京一 Hi student000

请输入关键字, 按... 搜索 所有工作项 高级过滤

编号 标题 迭代 处理人 优先级 状态 结束日期 操作

抱歉, 没有找到相匹配的工作项

2.3. 代码托管

2.3.1. 代码仓库

首先, 在上方的菜单中点击代码-代码托管。查看当前代码仓库, 点击对应仓库 (本次实验为Lab1) , 进入本次实验的代码仓库。

DEVCLOUD test_Lab1_小组student... 仪表盘 工作 代码 构建&发布 测试 Wiki 文档 设置 华北-北京一 Hi student000

首页 > test_Lab1_小组student000_0 > 代码托管 代码检查

新建仓库 设置我的SSH密钥 设置我的HTTPS密码

状态	仓库名称	类型	仓库URL	合并请求	仓库容量 (GIT L...)	创建者	最近更新时间	操作
●	Lab1	私有	SSH HTTPS	0	0.54M 0.00M	FudanSoft	2019-03-01 19:...	

(总条数: 1)

进入仓库后查看仓库详情。

仓库描述: Lab1 on Maven and SpringBoot.
5 提交 1 分支 0 标签 2 仓库成员 仓库大小: 0.54MB (0.54MB/1G Git库, 0MB LFS)
Shell 44.21% Batchfile 28.18% Java 27.61%

文件名称	更新时间	创建者	备注
.mvn	1天前	wangxin	init commit.
src	20分钟前	kylinxiang	update lab1
.gitignore	1天前	wangxin	modify .gitignore
README.md	2天前	FudanSoft	Initial commit
mvnw	1天前	wangxin	init commit.
mvnw.cmd	1天前	wangxin	init commit.

在仓库详情界面中将访问方式改为HTTPS，并点击右侧复制按钮复制仓库地址。

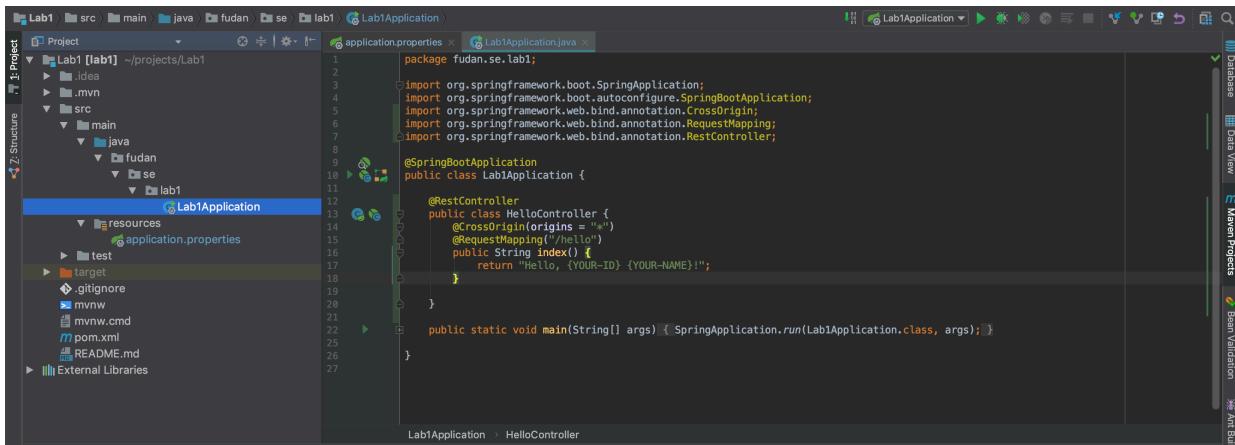
2.3.2. 克隆仓库

在本地命令行中进行如下操作: `git clone address_of_your_repository`, 并根据提示依次输入DevCloud的账号、密码。这里的账号为FudanSoft/你的账号。

```
1. fish /Users/xiangqilin/ta (fish)
xiangqilin@xiangqilin MacBook-Pro ~/ta>
git clone https://codehub.devcloud.huawei.com/test_Lab1_xzstudent000_000001/Lab1.git
Cloning into 'Lab1'...
Username for 'https://codehub.devcloud.huawei.com': FudanSoft/student000
Password for 'https://FudanSoft/student000@codehub.devcloud.huawei.com':
remote: Counting objects: 53, done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 53 (delta 5), reused 0 (delta 0)
Unpacking objects: 100% (53/53), done.
xiangqilin@xiangqilin MacBook-Pro ~/ta>
```

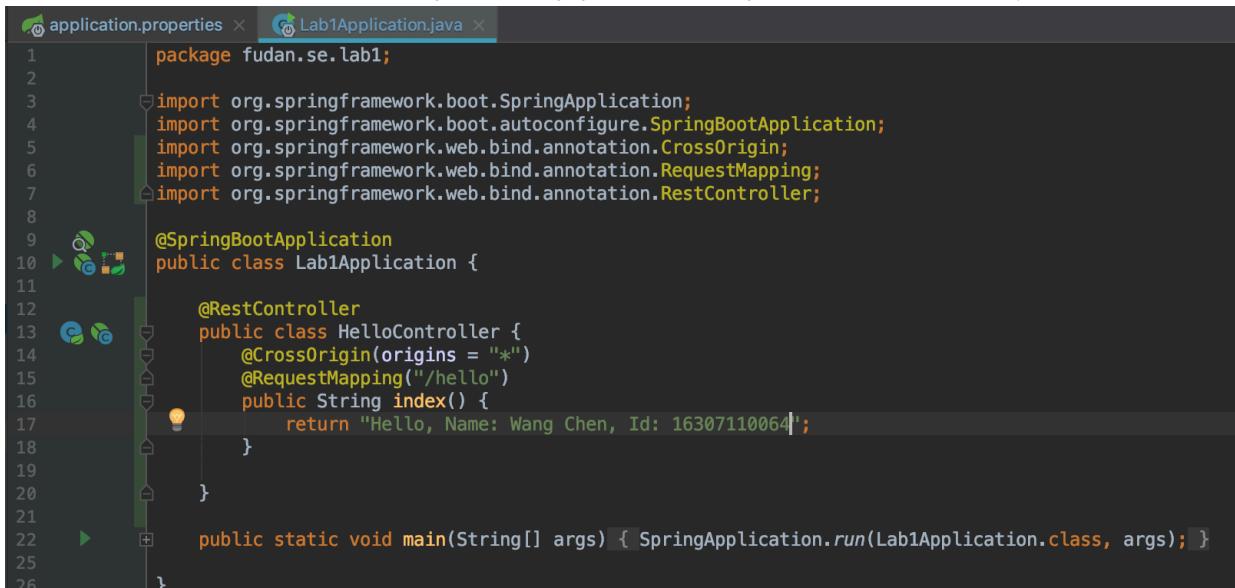
2.3.3. 修改源代码

在本地IDE或编辑器中查看源代码。



The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The code editor displays the `Lab1Application.java` file. The code contains a main class `Lab1Application` and a controller class `HelloController`. The `HelloController` has a method `index()` that returns a string with placeholder variables. The code editor highlights the placeholder `{YOUR-ID} {YOUR-NAME}`.

对源代码进行修改：将源代码中的 `{ YOUR-ID } {YOUR-NAME}` 改为你的姓名和学号，并保存。

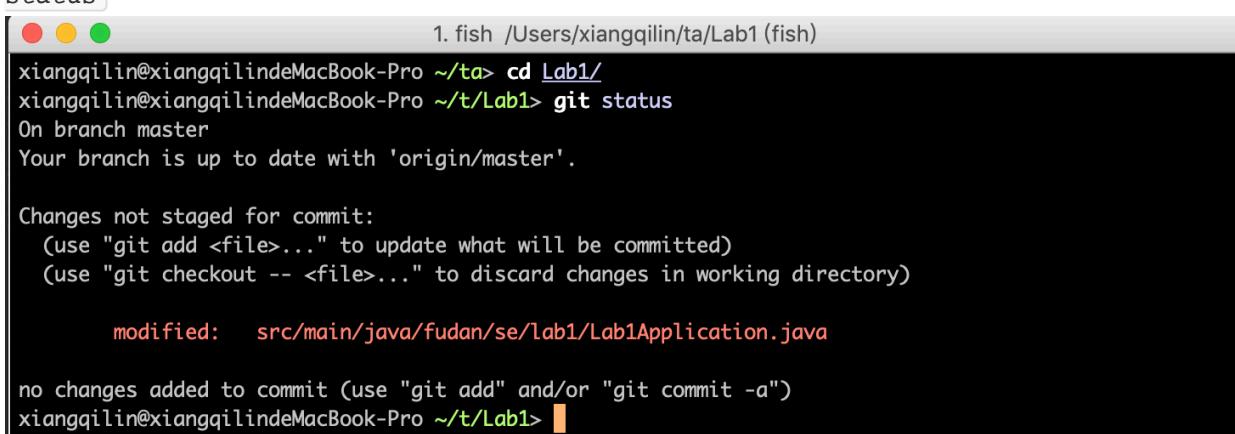


The screenshot shows the IntelliJ IDEA interface with the code editor displaying the modified `Lab1Application.java`. The placeholder `{YOUR-ID} {YOUR-NAME}` has been replaced with the user's name and ID: "Wang Chen, Id: 16307110064". The code editor shows syntax highlighting and some annotations.

2.3.4. 查看仓库状态

回到命令行，进入你的项目的根目录中：`cd xxxx/lab1/`，查看当前仓库状态使用命令：`git`

`status`



The screenshot shows a terminal window with the following command and output:

```
xiangqilin@xiangqilindeMacBook-Pro ~/ta> cd Lab1/
xiangqilin@xiangqilindeMacBook-Pro ~/t/Lab1> git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   src/main/java/fudan/se/lab1/Lab1Application.java

no changes added to commit (use "git add" and/or "git commit -a")
xiangqilin@xiangqilindeMacBook-Pro ~/t/Lab1>
```

2.3.5. 提交操作

使用`git add .`将你的修改添加到Git的本地暂存区中，并使用以下进行提交，并加入commit信息。`git commit . -m "update your name and id"`，成功提交后会显示你的仓库代码修改情况。

```
1. fish /Users/xiangqilin/ta/Lab1 (fish)
xiangqilin@xiangqilindeMacBook-Pro ~/t/Lab1> git add .
xiangqilin@xiangqilindeMacBook-Pro ~/t/Lab1> git commit -m "update name and id"
[master c421c74] update name and id
 1 file changed, 1 insertion(+), 1 deletion(-)
xiangqilin@xiangqilindeMacBook-Pro ~/t/Lab1>
```

类似以上操作，请修改项目的application.properties文件中配置的端口号为你分配端口。

分配的端口详见第3节 注意事项

eg. 假如你分配的端口为8081，则修改为server.port=8081

修改之后，执行依次执行`git add`, `git commit`命令提交修改

2.3.6. 推送修改

使用`git push`命令将你的这次修改提交（推送）到远程仓库。

```
1. fish /Users/xiangqilin/ta/Lab1 (fish)
xiangqilin@xiangqilindeMacBook-Pro ~/t/Lab1> git push
Counting objects: 9, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (9/9), 632 bytes | 632.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0)
To https://codehub.devcloud.huaweicloud.com/test_Lab1_xzstudent000_000001/Lab1.git
 52317d4..c421c74 master -> master
xiangqilin@xiangqilindeMacBook-Pro ~/t/Lab1>
```

2.4. 构建

回到DevCloud工作台，点击构建发布-编译构建，点击新建任务。并将任务名称设置为“Lab1_your_id”（Lab1下划线 你的学号），并点击下一步。

The screenshot shows the DevCloud interface with the 'Build & Release' tab selected. A modal dialog titled 'New Build Task' is open, showing the first step: 'Basic Information'. It has three tabs at the top: 'Basic Information' (selected), 'Select Source Code', and 'Select Build Template'. The 'Task Name' field contains 'lab1_16307110064'. The 'Project' dropdown is set to 'test_Lab1_小组student000_000001'. At the bottom of the dialog are 'Next Step' and 'Cancel' buttons.

源码源选择DevCloud，源码仓库选择Lab1，分支选择master，如图所示，并点击下一步。

新建编译构建任务

基本信息 选择代码源 选择构建模板

* 源码源: DevCloud

* 源码仓库: Lab1

* 分支: master

[上一步](#) [下一步](#) [取消](#)

选择“Maven模板”，并点击确定。

新建编译构建任务

基本信息 选择代码源 选择构建模板

推荐模板

- Maven** 使用Apache Maven构建Java项目
- 系统模板 (21)
- 自定义模板 (0)

系统模板 [不使用模板，直接创建](#)

Maven 使用Apache Maven构建Java项目

Gradle 使用Gradle项目自动化建构工具建构Java, Groovy和Scala项目

[上一步](#) [确定](#) [取消](#)

在高级选项中，在mvn后加入clean，这个命令会帮助你在每次构建时清空上一次的构建。点击新建，你将获得一个构建任务。

上传软件包到软件发布库

上传软件包到软件发布库

试试将您的步骤组合保存为自定义模版，其他构建任务就可以直接使用该模版了。[保存为自定义模版](#)

工具版本: maven3.5.3-jdk1.8

高级选项

```

1 #只打包不执行单元测试
2 # mvn package -Dmaven.test.skip=true
3
4 # 打包:跳过单元测试;每次构建检查更新
5 mvn clean package -Dmaven.test.skip=true -U

```

[新建](#) [取消](#)

点击开始构建，平台将会根据配置步骤进行构建，并将你的软件包发布到发布仓库。这里需要很长一段时间构建，请稍作休息。构建成功后，将会出现 **Finished Success** 提示。你可以在构建历史中查看你的每次构建。

```

20190301.1
student000 构建于 2019-03-01 19:41
构建时长 00:00:23
下裁构建包
代码检出 (1.835s)
Maven构建 (18.298s)
上传软件包到软件发... (3.146s)

102 [2019-03-01 19:41:30.129] [INFO] [上传软件包到软件发布库:上传软件包到软件发布库] : Begin to upload file : target/lab1-0.0.1-SNAPSHOT.jar
103 [2019-03-01 19:41:30.129] [INFO] [上传软件包到软件发布库:上传软件包到软件发布库] : Remote path is: /lab1_16307110064/20190301.1/
104 [2019-03-01 19:41:32.769] [INFO] [上传软件包到软件发布库:上传软件包到软件发布库] : Upload filename is : lab1-0.0.1-SNAPSHOT.jar
105 [2019-03-01 19:41:32.770] [INFO] [上传软件包到软件发布库:上传软件包到软件发布库] : Uploading artifact lab1-0.0.1-SNAPSHOT complete
106 [2019-03-01 19:41:32.770] [INFO] [上传软件包到软件发布库:上传软件包到软件发布库] : Versionset status code:1, please config the service
107 [2019-03-01 19:41:32.813] [INFO] [上传软件包到软件发布库:上传软件包到软件发布库] : step finished
108 [2019-03-01 19:41:32.838] [INFO] [上传软件包到软件发布库] : StagePostExecution started
109 [2019-03-01 19:41:32.838] [INFO] [上传软件包到软件发布库] : StagePostExecution finished
110 [2019-03-01 19:41:34.264] [INFO] : PipelinePostExecution started
111 [2019-03-01 19:41:34.264] [INFO] : PipelinePostExecution finished
112 [2019-03-01 19:41:34.296] [job_4f6fede2-11e0-449e-824a-7027d7fb29e8_1551440459377] Running shell script
113 [2019-03-01 19:41:34.555] 3.5.3-jdk-1.8: Pulling from codeci/maven
114 [2019-03-01 19:41:34.555] Digest: sha256:aefae89ced51943501e73a600c83d94ef0d065cb7e8070fe2fb49ad8c6af8273
115 [2019-03-01 19:41:34.555] Status: Image is up to date for ****:****:****:****:2020/codicei/maven:3.5.3-jdk-1.8
116 [2019-03-01 19:41:34.575] [job_4f6fede2-11e0-449e-824a-7027d7fb29e8_1551440459377] Running shell script
117 [2019-03-01 19:41:34.871] ****:****:****:****-slave does not seem to be running inside a container
118 [2019-03-01 19:41:35.784] [job_4f6fede2-11e0-449e-824a-7027d7fb29e8_1551440459377] Running shell script
119 [2019-03-01 19:41:37.324] [job_4f6fede2-11e0-449e-824a-7027d7fb29e8_1551440459377] Running shell script
120 onCompleted: job_4f6fede2-11e0-449e-824a-7027d7fb29e8_1551440459377 #1
121 Finished: SUCCESS
122

```

2.5. 部署

2.5.1. 新建任务

点击新建任务，选择新建任务（模板），任务名称为deploy_your_id（deploy 你的学号）。

新建部署任务 [如何创建任务](#)

基本信息 **选择部署模板**

* 任务名称:

* 所属项目:

描述 [?](#)

下一步 **取消**

点击下一步，选择SpringBoot应用部署，点击确定。

新建部署任务 [如何创建任务](#)

基本信息 **选择部署模板**

请输入内容

推荐使用

- 推荐使用 (0)
- 系统模板 (4)**
- 自定义模板 (0)

系统模板 [不使用模板, 直接创建](#)

- SpringBoot应用部署** 在主机上部署springboot的jar包，并启动该服务 [示例: 新建SpringBoot应用部署](#)
- Tomcat应用部署** 在主机上部署Tomcat的war包，并启动该服务 [示例: 新建Tomcat应用部署](#)
- PhonixDemo**

上一步 **确定** **取消** [源，并部署到容器集群](#)

2.5.2. 调整部署步骤

将左侧部署步骤一栏中的“停止SpringBoot服务”拖动到最后，并删除“URL健康测试”，并在“启动SpringBoot服务”和“停止SpringBoot服务”两个步骤中添加步骤，选择执行等待服务。步骤如图所示。

The screenshot shows the DEVCloud deployment interface with the following steps visible:

- 安装JDK**: Java icon, "在主机上安装JDK".
- 选择部署来源**: Folder icon, "选择软件发布仓的文件，或者选择某次构建...".
- 启动SpringBoot服务**: Power icon, "根据服务的指定路径来启动、停止服务".
- 执行等待**: Timer icon, "根据用户输入的时间进行等待". This step is highlighted with a blue border and has a checked checkbox and a trash bin icon.
- 停止SpringBoot服务**: Power icon, "根据服务的指定路径来启动、停止服务".

At the bottom, a note says: "试试将您的步骤组合保存为自定义模板，其他部署任务就可以直接使用该模板了。"

2.5.3. 配置安装JDK

选择左侧部署步骤一栏中的“安装JDK”，点击创建主机组。

点击新建，如下图所示，按照下面的表格填写信息。

上图中信息，按照下表填写。

组号	姓名	姓名	姓名	姓名	ip	主机名	用户名	密码
1	林家南	吴妍琪	杨妍	孙梦	114.115.150.198	se01		
2	李翀	周钰承	王尚	李周清	114.116.40.188	se02		
3	孟超	李玎善	于志昊	杜东方				
4	罗蓉	金毅铭	张廷	周君怡				
5	林晨	俞继涛	邓朋	张岑溪	114.115.213.211	se03		
6	王宸	刘佳兴	黄佳妮	唐昕悦				
7	储开东	夏应锋	谢梦康	郑志强	114.115.149.192	se04		
8	林国鹏	张思源	胡彦雯	刘书宁				
9	王昱	童麦可	胡梦涵	邱咏婷	114.115.217.166	se05		
10	沈家辉	张宏年	宋普琪	朱向宇				
11	石睿欣	宋怡景	胡宵宵	张逸涵	114.116.136.180	se06		
12	姜尔玲	张浩	杨溪溪	黎沈少杰				
13	李江渝	王永立	杨辉	李璠	114.115.164.240	se07		
14	刘佳楠	翟登展	刘海强	王献迪				
15	赵潇乾	周振昊	符根	黄鼎骏				
16	夏维	申浩凡	陈昂伸	陈裕	114.115.164.58	se08		
17	周子裕	林晓青	王和旺	梁伟业				
18	卢永强	秦智超	赵骞云	方冠军	114.116.112.8	se09		
19	黄睿佳	陈国华	崔欣宇	高祥兴				
20	黄蕙茹	李舒阳	兰萍	罗天越	114.115.220.136	se10		
21	陆逸凡	黄元敏	吴琛宁	吴新铭				
22	王嵩立	颜谋福	陈靖元	姜向阳	114.115.154.36	se11		
23	吴楚盈	刘铭涵	贺曦	夏禹天				
24	丁凡宇	田紫薇	刘建	顾淳宇	114.115.164.135	se12		

配置完成后，然后选择你创建的主机组。

安装路径为“/usr/local/jdk”。

The screenshot shows the DEVCloud interface under the 'Deployment' tab. On the left, there's a list of deployment steps: 'Stop SpringBoot service', 'Install JDK' (selected), 'Choose deployment source', 'Start SpringBoot service', and 'URL health check'. The 'Install JDK' step is highlighted with a blue border. On the right, the 'Install JDK' configuration panel is open, showing fields for 'Display name' (set to 'Install JDK'), 'Host group' (set to 'Create host group'), 'JDK version' (set to 'JDK-8.1.8.0'), 'JDK installation path' (set to '/usr/local/jdk'), and a note about permissions. At the bottom are 'Save', 'Save as draft', and 'Cancel' buttons.

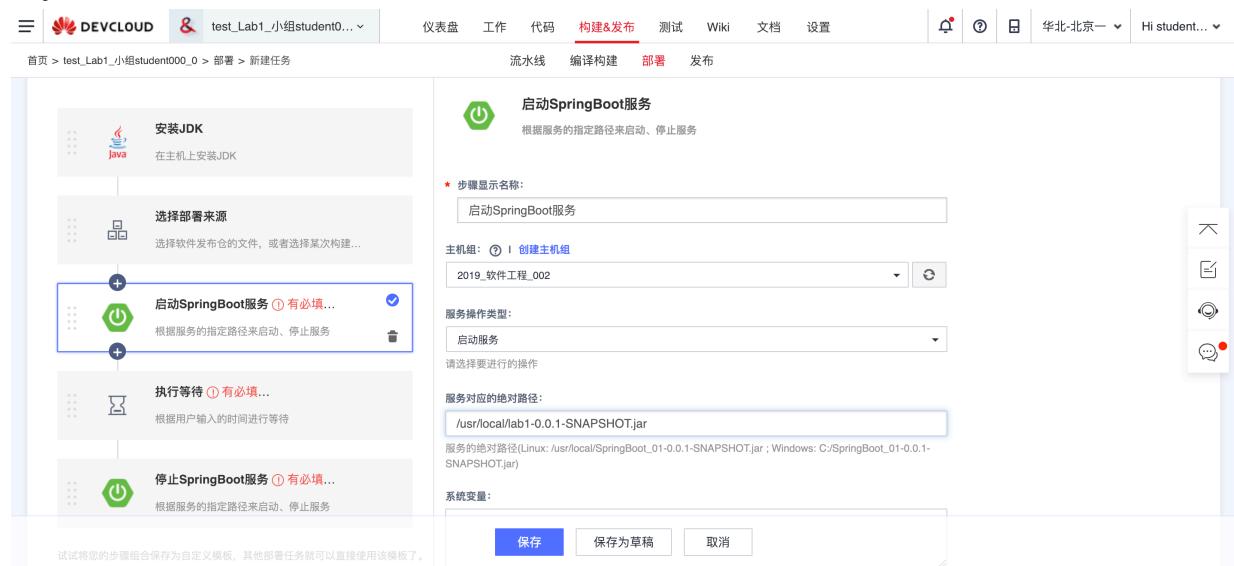
2.5.4. 配置部署来源

选择左侧部署步骤一栏中的“选择部署来源”，选择你之前创建的构建任务，序号选择Latest，下载文件到主机路径填/usr/local。配置如图所示，主机组选择你创建的主机组。

The screenshot shows the DEVCloud interface under the 'Deployment' tab. On the left, there's a list of deployment steps: 'Select deployment source (必填...)', 'Stop SpringBoot service (必填...)', 'Execute wait', and 'Start SpringBoot service (必填...)'. The 'Select deployment source' step is highlighted with a blue border. On the right, the 'Select deployment source' configuration panel is open, showing fields for 'Display name' (set to '选择部署来源'), 'Source type' (set to 'Build task'), 'Host group' (set to 'Create host group'), 'Build task selection' (set to 'lab1_16307110064'), 'Build sequence' (set to 'Latest'), and 'Download file to host path' (set to '/usr/local'). A note at the bottom says 'The path must have write permission for the current user (Linux: /usr/local; Windows: C:/download)'. At the bottom are 'Save', 'Save as draft', and 'Cancel' buttons.

2.5.5. 配置启动服务

选择左侧部署步骤一栏中的“启动SpringBoot服务”，服务操作类型为启动服务，服务对应的绝对路径填/usr/local/lab1-0.0.1-SNAPSHOT.jar。配置如图所示，配置如图所示，主机组选择你创建的主机组。

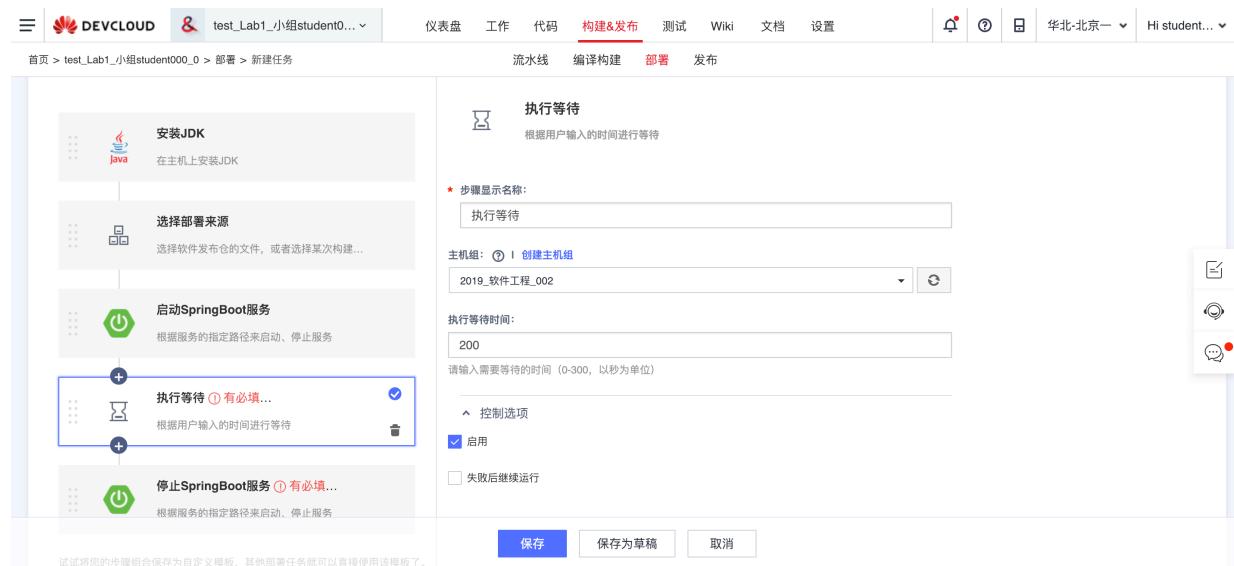


The screenshot shows the DevCloud interface with the 'Deployment' tab selected. On the left, there's a list of deployment steps: 'Install JDK', 'Select Deployment Source', 'Start SpringBoot Service' (which is highlighted with a red border), 'Wait Execution', and 'Stop SpringBoot Service'. The 'Start SpringBoot Service' step has a validation error indicated by a red circle with a question mark. The configuration panel on the right is titled 'Start SpringBoot Service' and includes fields for 'Step Display Name' (set to '启动SpringBoot服务'), 'Host Group' (set to 'I 创建主机组'), 'Service Operation Type' (set to 'Start Service'), 'Service Absolute Path' (set to '/usr/local/lab1-0.0.1-SNAPSHOT.jar'), and a 'System Variable' section. At the bottom are 'Save', 'Save as Draft', and 'Cancel' buttons.

2.5.6. 配置执行等待

选择左侧部署步骤一栏中的“执行等待”，创建时间为200s，你将具有200s的时间访问你启动的服务。

配置如图所示，配置如图所示，主机组选择你创建的主机组。



The screenshot shows the DevCloud interface with the 'Deployment' tab selected. The 'Wait Execution' step is selected and highlighted with a red border. Its configuration panel on the right includes fields for 'Step Display Name' (set to '执行等待'), 'Host Group' (set to 'I 创建主机组'), 'Execution Wait Time' (set to '200'), and a 'Control Options' section with a checked 'Enable' checkbox. At the bottom are 'Save', 'Save as Draft', and 'Cancel' buttons.

2.5.7. 配置停止服务

选择左侧部署步骤一栏中的“停止SpringBoot服务”，服务操作类型为停止服务，服务对应的绝对路径填/usr/local/lab1-0.0.1-SNAPSHOT.jar。

配置如图所示，主机组选择你所在小组对应的主机组！！！

2.5.8. 进行部署

点击保存，保存部署配置。点击执行即可进行部署。当你在执行详情中发现你的部署已经进入执行等待环节（如图所示），请珍惜你的测试时间（200s），在浏览器中测试你的服务接口。

```

4 [2019-03-01 20:24:44.580]
5 [2019-03-01 20:24:44.580] TASK [Gathering Facts] *****
6 [2019-03-01 20:24:44.580] ok: [***,***,***,***]
7 [2019-03-01 20:24:44.627]
8 [2019-03-01 20:24:44.627] TASK [Startup information] *****
9 [2019-03-01 20:24:44.627] ok: [***,***,***,***] ={
10 [2019-03-01 20:24:44.627]     "msg": "Start execution wait"
11 [2019-03-01 20:24:44.627] }
12 [2019-03-01 20:24:44.676]
13 [2019-03-01 20:24:44.677] TASK [Execution parameter] *****
14 [2019-03-01 20:24:44.677] ok: [***,***,***,***] ={
15 [2019-03-01 20:24:44.677]     "msg": "wait_time=200"
16 [2019-03-01 20:24:44.677] }
17

```

2.5.9. 测试部署的应用

测试方法为在导航栏中输入“`http://主机组的ip:端口/hello`”，此时若服务部署成功，你将看到你的接口返回“Hello, your_name, your_id !”字样。

测试时间过后，将停止服务。

The screenshot shows the DEVCloud interface for a deployment named 'test_Lab1_小组student0...'. It displays a timeline of events with execution times:

- 初始化: 7.224s
- 安装JDK: 6.641s
- 选择部署来源: 11.543s
- 启动SpringBoot服务: 6.057s
- 执行等待: 206.466s
- 停止SpringBoot服务: 7.11s

The log output shows the following tasks:

```

4 [2019-03-01 20:28:12.284]
5 [2019-03-01 20:28:12.284] TASK [Gathering Facts] ****
6 [2019-03-01 20:28:12.284] ok: [***.***.***.***]
7 [2019-03-01 20:28:12.284]
8 [2019-03-01 20:28:12.284] TASK [Startup information] ****
9 [2019-03-01 20:28:12.285] ok: [***.***.***.***] ={ 
10 [2019-03-01 20:28:12.285] "msg": "Start or stop SpringBoot service"
11 [2019-03-01 20:28:12.285] }
12 [2019-03-01 20:28:12.285]
13 [2019-03-01 20:28:12.285] TASK [Execution parameter] ****
14 [2019-03-01 20:28:12.285] ok: [***.***.***.***] ={ 
15 [2019-03-01 20:28:12.285] "msg": "/usr/local/lab1-0.0.1-SNAPSHOT.jar"
16 [2019-03-01 20:28:12.285] }
17 [2019-03-01 20:28:12.285]
18 [2019-03-01 20:28:12.285] TASK [Get p_id] ****
19 [2019-03-01 20:28:12.285] changed: [***.***.***.***]
20 [2019-03-01 20:28:12.285]
21 [2019-03-01 20:28:12.285] TASK [Set pids] ****
22 [2019-03-01 20:28:12.285] ok: [***.***.***.***]
23 [2019-03-01 20:28:12.285]
24 [2019-03-01 20:28:12.285] TASK [Debugmess pids] ****
25 [2019-03-01 20:28:12.285] ok: [***.***.***.***] ={ 
26 [2019-03-01 20:28:12.285] "msg": "568"
27 [2019-03-01 20:28:12.285] }
28 [2019-03-01 20:28:12.286]
29 [2019-03-01 20:28:12.286] TASK [Stop] ****
30 [2019-03-01 20:28:12.286] changed: [***.***.***.***] =(item=568)
31 [2019-03-01 20:28:12.286]
32 [2019-03-01 20:28:12.286] PLAY RECAP ****

```

停止服务后，无法进行测试，再访问url将会失败。

The screenshot shows a browser window with the URL 114.116.6.241:8080/hello. The error message is:

无法访问此网站
114.116.6.241 拒绝了我们的连接请求。
请在 Google 中搜索“114 116 241 8080 hello”
ERR_CONNECTION_REFUSED

3. 注意事项

由于服务器数量限制，2个小组将使用一台服务器（绑定了一个公网IP）。每个同学在每台服务器上部署应用都指定了端口号（姓名后面对应的端口号）。

组号	姓名	端口	姓名	端口	姓名	端口	姓名	端口	ip
1	林家南	8081	吴妍琪	8082	杨妍	8083	孙梦	8084	114.115.150.198
2	李翀	8085	周钰承	8086	王尚	8087	李周清	8088	
3	孟超	8081	李玎善	8082	于志昊	8083	杜东方	8084	114.116.40.188
4	罗蓉	8085	金毅铭	8086	张廷	8087	周君怡	8088	
5	林晨	8081	俞继涛	8082	邓朋	8083	张岑漫	8084	114.115.213.211
6	王宸	8085	刘佳兴	8086	黄佳妮	8087	唐昕悦	8088	
7	储开东	8081	夏应锋	8082	谢梦康	8083	郑志强	8084	114.115.149.192
8	林国鹏	8085	张思源	8086	胡彦雯	8087	刘书宁	8088	
9	王昱	8081	童麦可	8082	胡梦涵	8083	邱咏婷	8084	114.115.217.166
10	沈家辉	8085	张宏年	8086	宋普琪	8087	朱向宇	8088	
11	石睿欣	8081	宋怡景	8082	胡宵宵	8083	张逸涵	8084	114.116.136.180
12	姜尔玲	8085	张浩	8086	杨溪溪	8087	黎沈少杰	8088	
13	李江渝	8081	王永立	8082	杨辉	8083	李璠	8084	114.115.164.240
14	刘佳楠	8085	瞿登展	8086	刘海强	8087	王戴迪	8088	
15	赵潇乾	8081	周振昊	8082	符根	8083	黄鼎骏	8084	114.115.164.58
16	夏维	8085	申浩凡	8086	陈昂伸	8087	陈裕	8088	
17	周子裕	8081	林晓青	8082	王和旺	8083	梁伟业	8084	114.116.112.8
18	卢永强	8085	秦智超	8086	赵骞云	8087	方冠军	8088	
19	黄睿佳	8081	陈国华	8082	崔欣宇	8083	高祥兴	8084	114.115.220.136
20	黄蕙茹	8085	李舒阳	8086	兰萍	8087	罗天越	8088	
21	陆逸凡	8081	黄元敏	8082	吴琛宁	8083	吴新铭	8084	114.115.154.36
22	王嵩立	8085	颜谋福	8086	陈靖元	8087	姜向阳	8088	
23	吴楚盈	8081	刘铭涵	8082	贺曦	8083	夏禹天	8084	114.115.164.135
24	丁凡宇	8085	田紫薇	8086	刘建	8087	顾淳宇	8088	

4. 作业提交

最终提交两个作业：项目、实验报告。Classroom中会分别对应两个作业。

4.1. 提交项目

- 进入Classroom点击提交作业。

4.2. 提交实验报告

- 实验报告无固定模板，要求将实验的主要步骤截图，并阐述该步骤的意义。
- 最后阐述你对实验的认识和理解。
- 实验报告以作业附件方式提交至DevCloud平台。
- 截止时间2019年3月10日（本周日）24: 00点。