# GIT 入门操作手册—for BUAA

## 目标:

* 通过百度效率云iCode建立自己的代码库，完成一次代码的提交

## 操作步骤:

1. 从统一的远程仓库下载代码到本地
2. 在本地创建分支;
3. 在分支上添加一个文件;
4. 完成本地代码提交;
5. 将本地的变更提交到远程代码仓库;

## 具体实验手册:

### STEP1: 登录百度效率云，建立一个新的开发项目

#### STEP1-1 登录百度效率云:

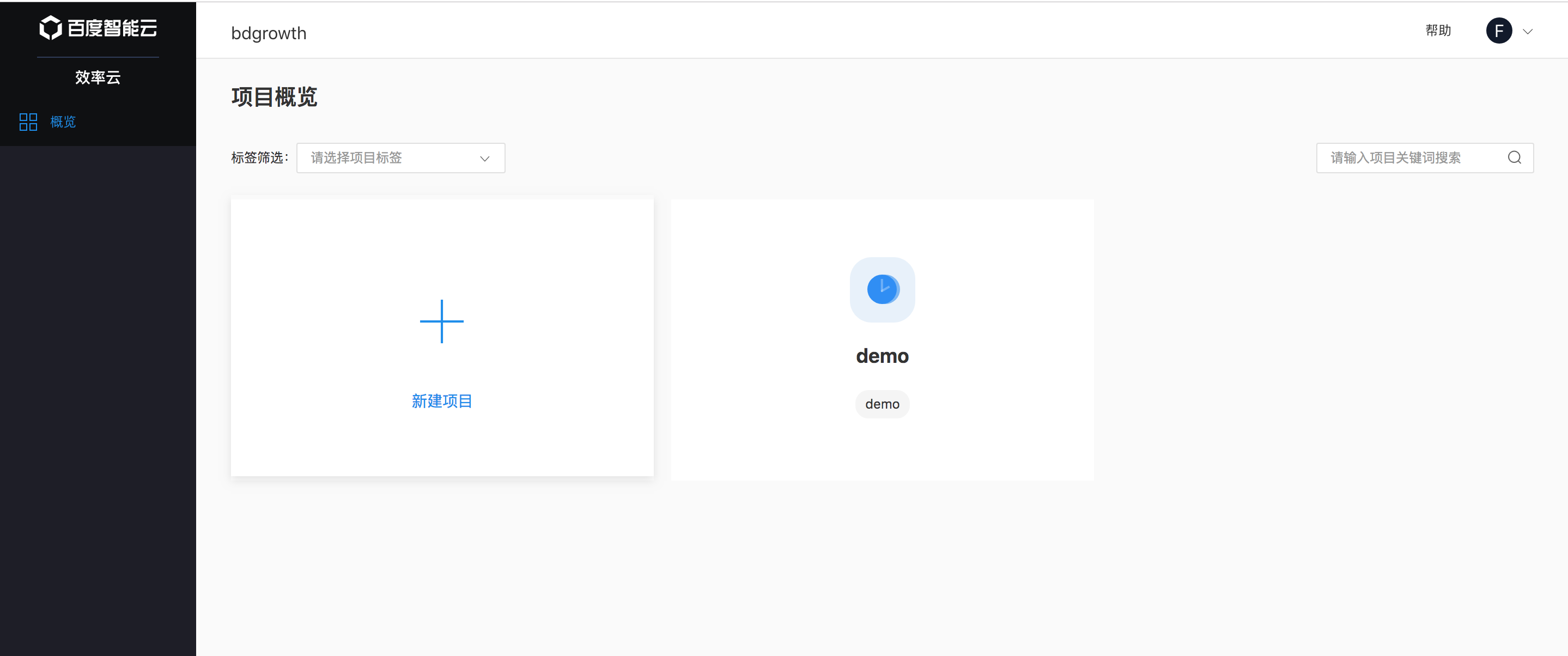
1-1-1登录百度效率云

登录地址: XXX.login.bce.baidu.com

用户名: 使用你的课程报名的邮箱

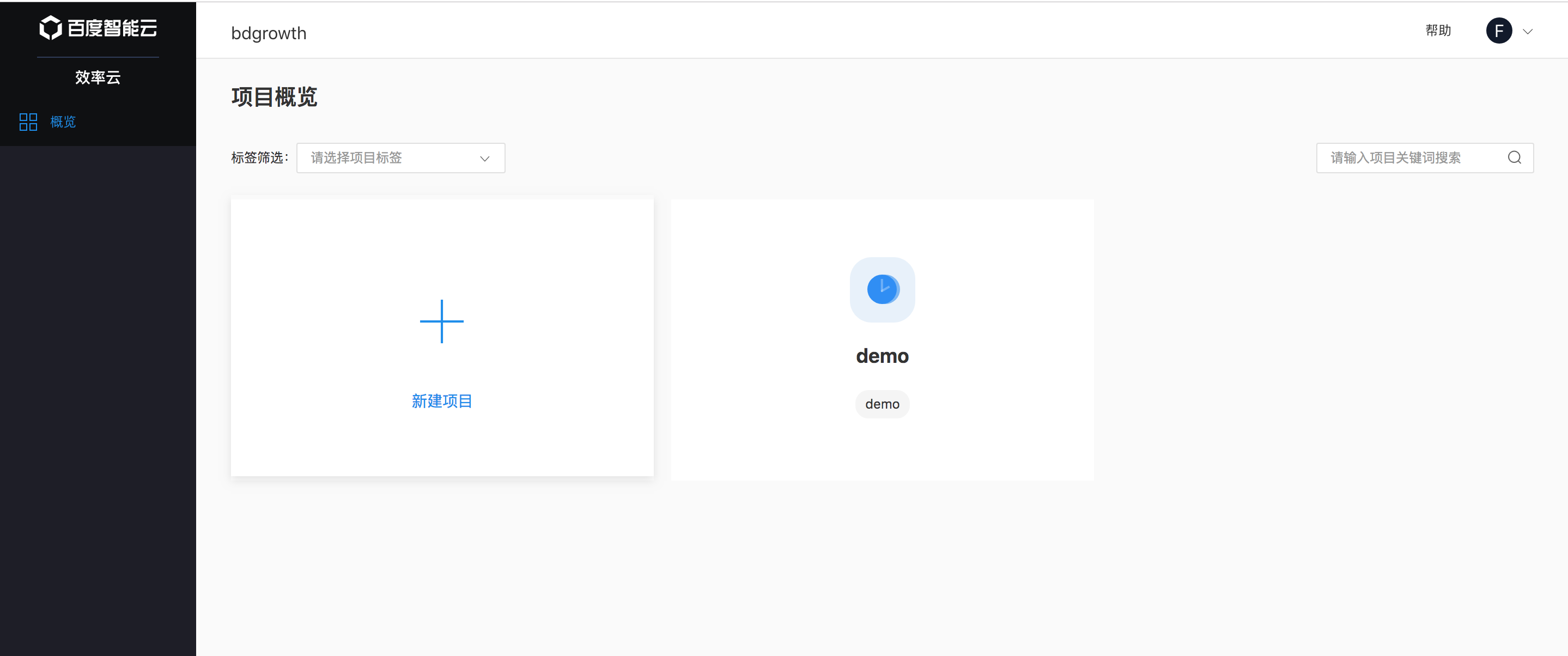
密码: BUAA2019!, 登录后请自行修改

登录后的界面如下:



#### STEP1-2 新建一个开发项目:

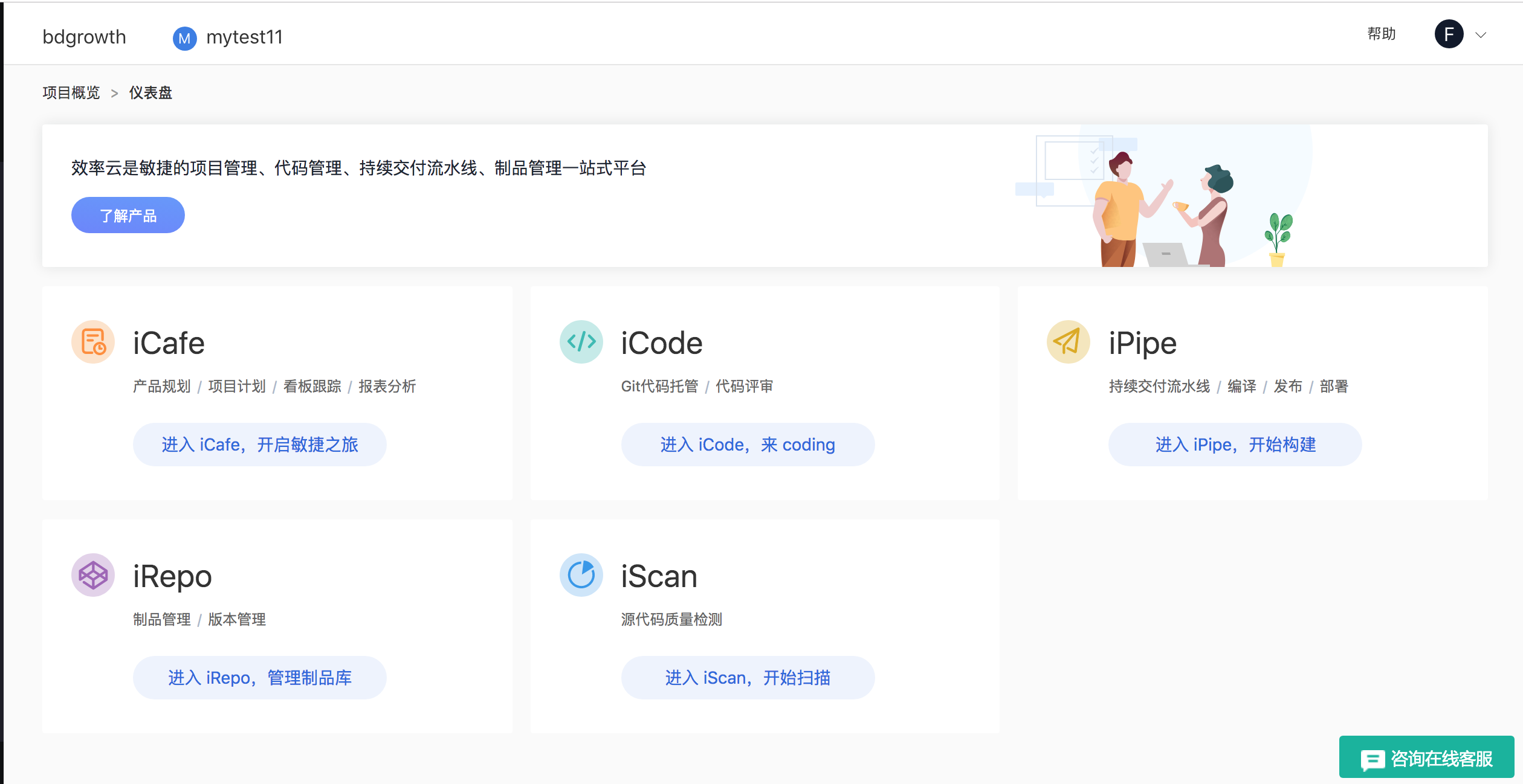
1-2-1： 点击新建项目:



1-2-2: 在弹出窗口里配置项目名称，标识[[1]](#footnote-1)



项目建好后，点击项目卡片，进入项目主页面，显示如下:



恭喜各位同学，你们已经成功的建立了一个开发项目

### STEP2: 修改自己的GIT仓库密码，新建一个代码库

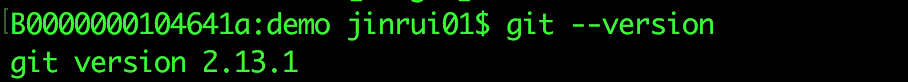
#### STEP2-1: 新建一个GIT仓库，将空白仓库克隆到本地电脑

**2-1-0:** 在本地安装GIT, 同学们可以登录GIT主站: <https://git-scm.com/downloads>

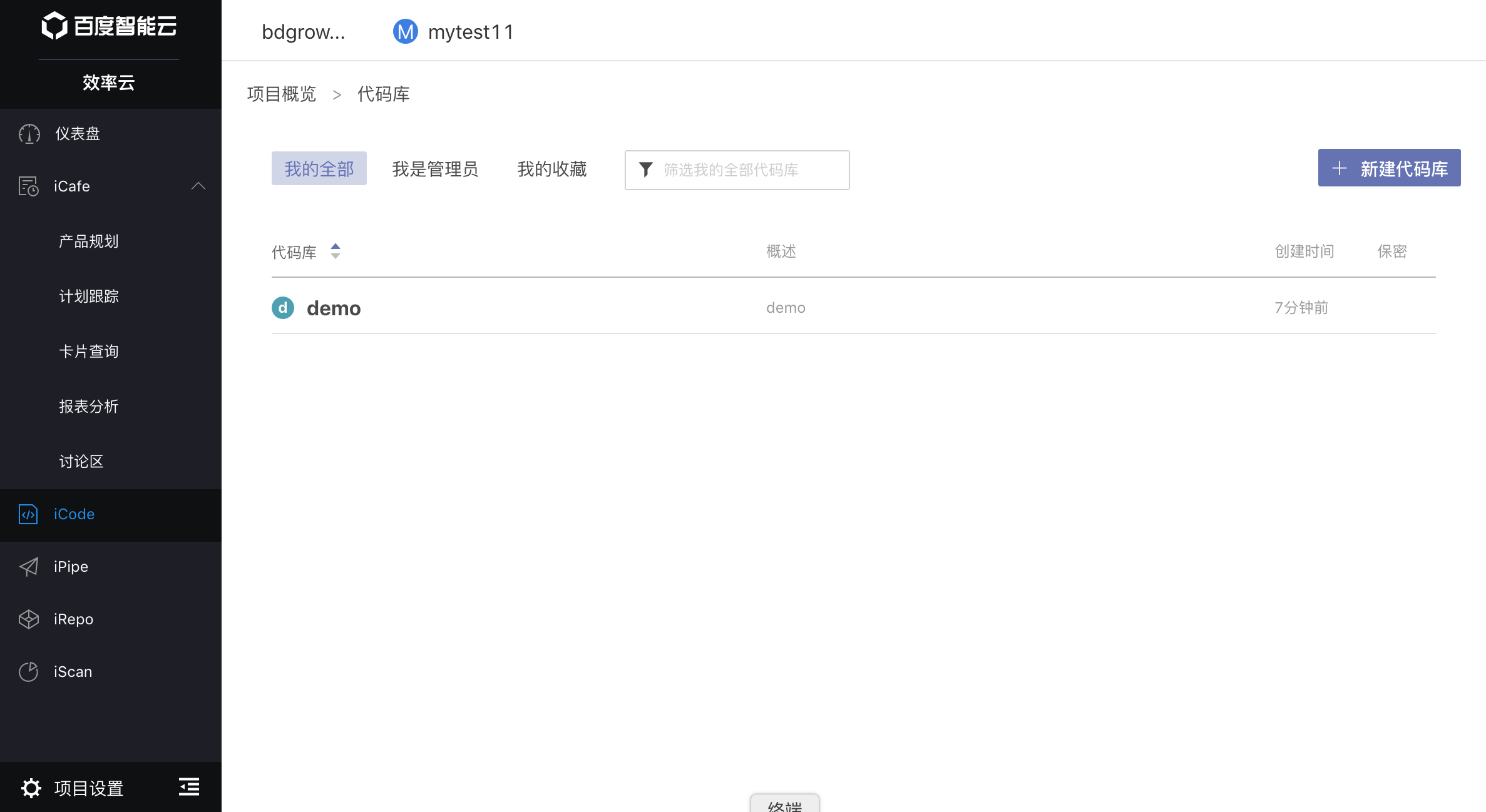
结合自己本地的操作系统[[2]](#footnote-2)进行安装, 完成安装后，在操作系统的命令行输入:

Git --version

如果能够正确显示git 版本号，则安装完成, 如下图：

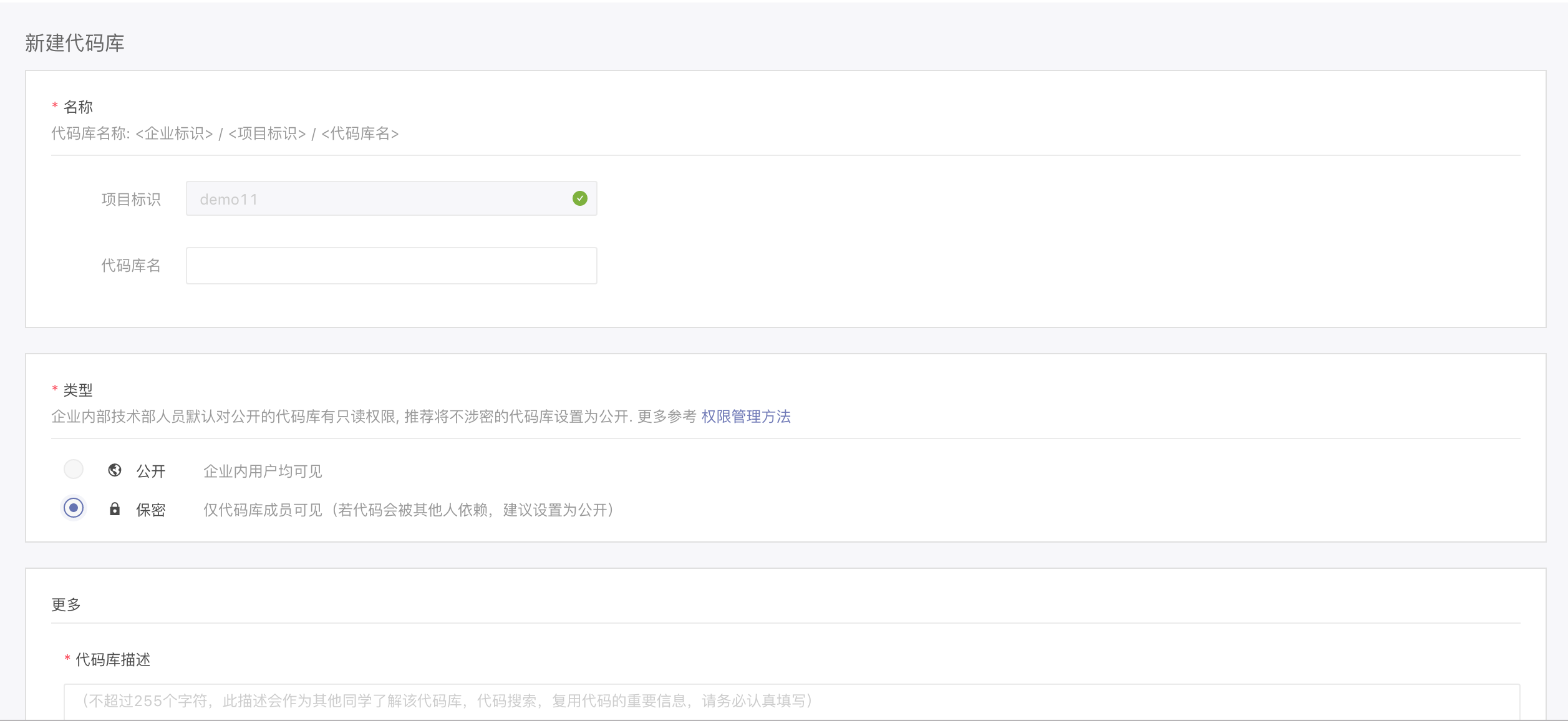


**2-1-1:** 新建GIT仓库，点击项目首页上的”iCode”卡片 ( iCode是百度效率云上对标gitlab的产品，支持团队级别的开发协作场景，后续关于代码的相关操作，都在iCode中进行), 进入项目代码管理:



上述页面是iCode的首页，也就是代码库列表页面，如图所示，在这里我们提供了一个默认的java代码库: demo；熟悉java开发的同学，可以直接使用该代码库；或者，同学们可以自己新建一个代码库:

**2-1-2:** 新建一个代码库，点击上述页面右上角的”新建代码库”，进入下面界面，输入代码库名称，将代码库设置为”私有”以及代码库说明之后，点击最下方”确定”完成新建工作



**2-1-3:** 修改自己的git密码，第一次执行代码clone,或代码提交的时候，系统会提示你输入密码。在代码库列表，或者代码库主页面，点击右上角的图标，进入GIT密码设置页面



修改密码的入口

GIT密码设置页面如下, 密码要求长度8-20位，同时包括大小写字母，数字以及下列特殊符号: %+\_.- 。手机验证后完成密码的设置



**2-1-4：** 在代码库列表，选择你刚才新建的代码库, 进入代码库主页面,如下:



场景2的脚本

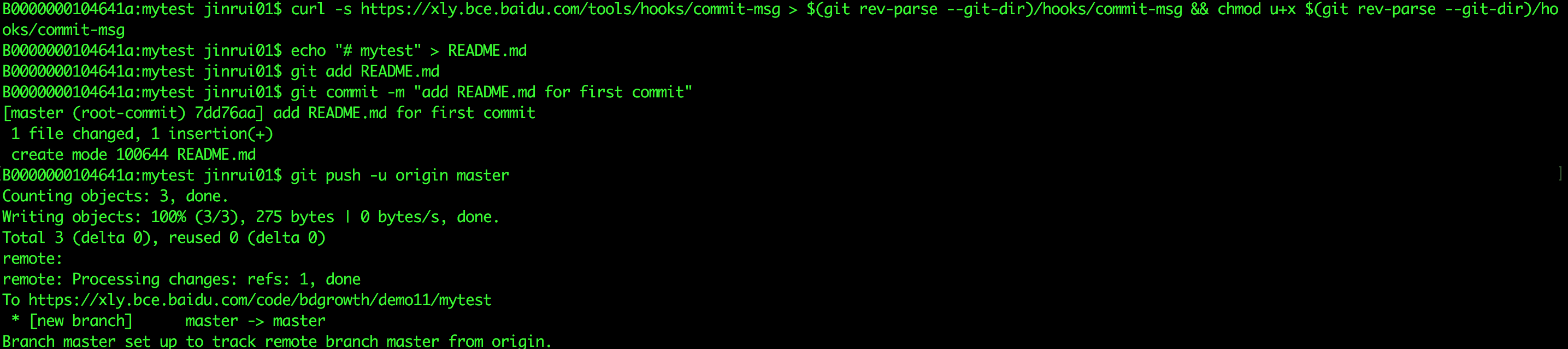
场景1的脚本

查看右侧的文档，里面介绍了两种情况, 我们分别介绍:

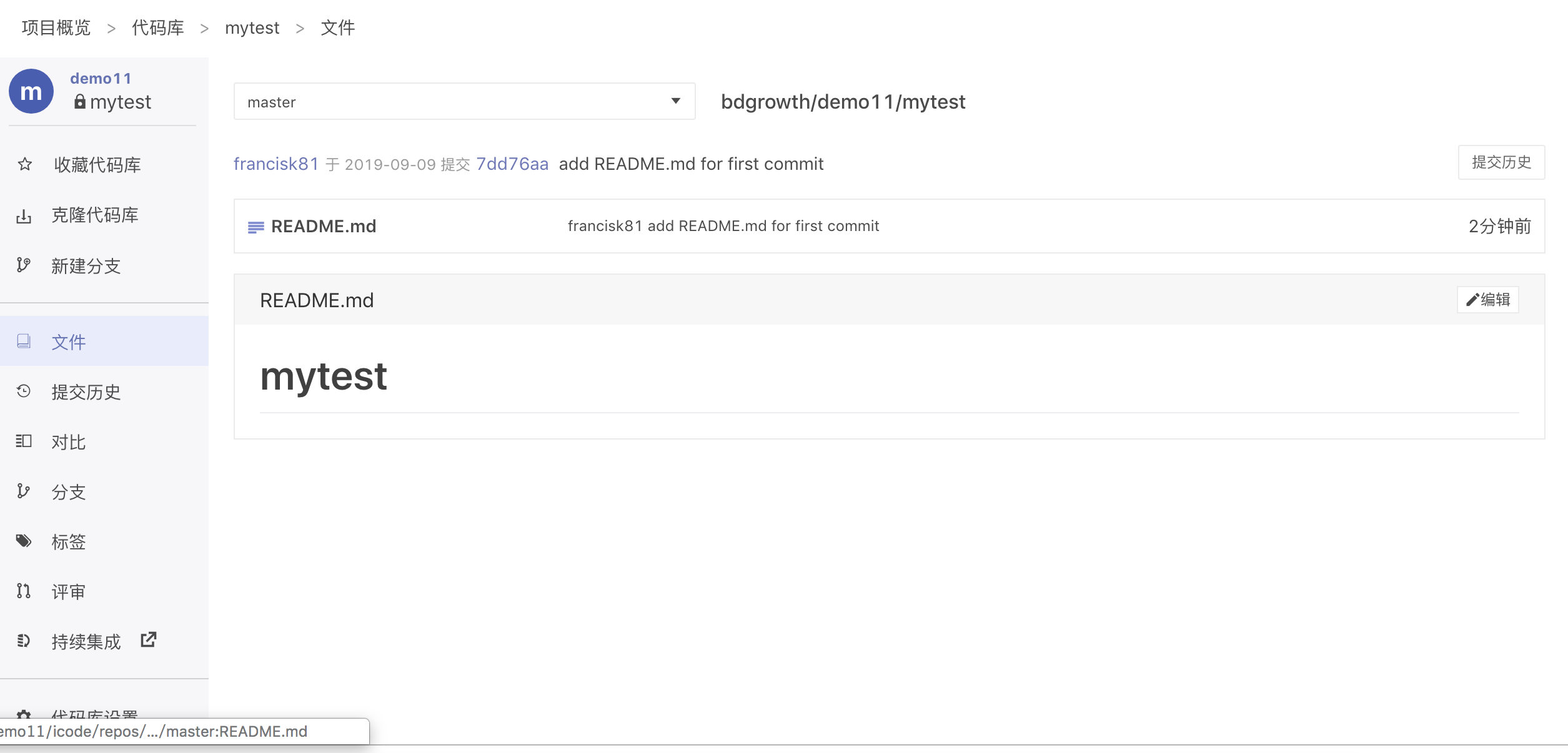
1. 你之前没有现成的git代码库，只有一些本地代码，或者干脆没有代码，参考场景1，具体执行过程如下:
   1. 在本地新建一个文件夹，做为本地代码库的地址
   2. 在文件夹的根目录输入上述脚本(拷贝粘贴即可)
   3. 根据提示，输入刚才你设置好的密码



* 1. 命令执行到 git push -u origin master的时候，直接按回车完成第一次代码提交



* 1. 提交成功后，再次刷新当前代码库的页面，显示应当如下:



1. 如果你之前在github，或者gitlab上已经托管的代码，参考场景2,具体执行过程如下:
   1. 拷贝场景2的脚本，直接本地执行，同样，执行成功后页面会显示你之前的代码

### STEP3: 在本地建立一个开发分支，修改一部分代码

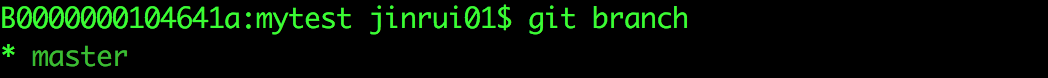
我们之前完成了代码库的设置，克隆，接下来我们将专注在本地的GIT工具操作：

#### STEP3-1: 在本地新建一个开发分支

分支是GIT非常重要的概念, 在企业实际的开发场景中，我们将不同的任务在不同的分支上开发。用以隔离任务之间的影响；

**3-1-1 查看本地的分支, 执行命令:**

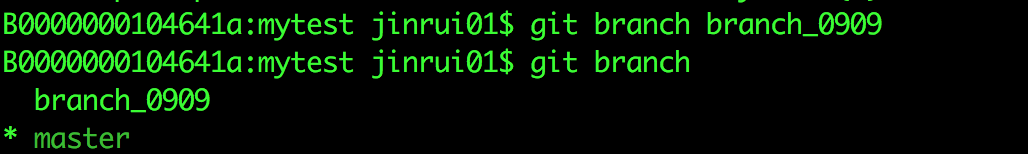
git branch



这时我们看到，本地只有一个叫”master”的分支

**3-1-2 新建一个本地分支, 执行命令:**

git branch <分支名称>

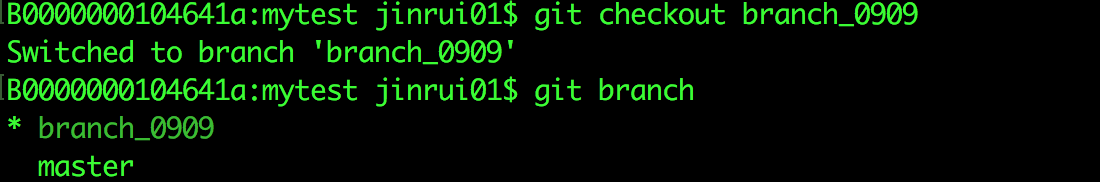
****

如上图所示，我新建了一条名为”branch\_0909”的分支；当我们再次执行git branch的时候，会发现本地已经新建了一个叫branch\_0909的开发分支

**3-1-3切换到新建的分支**

注意到上面截图中master前面的\*了么？这代表我们当前在哪条分支进行开发。当我们想切换到新建的分支时，我们使用下面命令:

git checkout <分支名称>



执行命令后，系统会提示我们已经swtich to branch ‘branch\_0909’

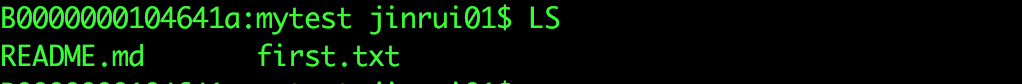
再次执行git branch，\*已经转移到我们刚才建立的分支了

**3-1-4 在本地添加一个新的文件，执行命令:**

echo “This is my first commit”> first.txt



上面这条命令代表我们将”This is my first commit”这条文本信息输入到名为first.txt文件中，如果本地没有这个文件，则先新建一个first.txt文件。执行之后再查看本地文件,windows下使用”dir”命令，linux和MAC下使用”ls”命令，我们会发现多了一个first.txt文件:



至此，我们在当前的开发分支上完成了第一次的代码变更。下一步我们将本地的代码变更同步到远程代码库，以方便其它同学同步和下载

### STEP4: 完成第一次代码提交

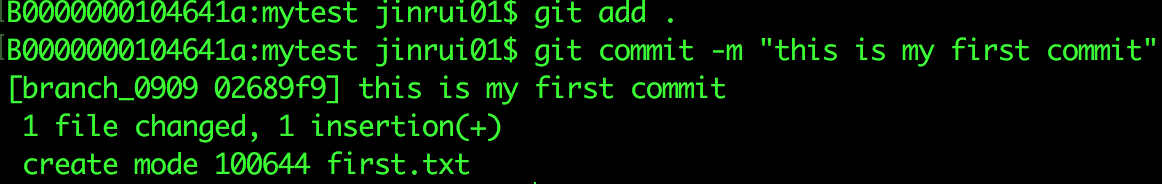
#### STEP4-1: 完成本地的代码提交:

**4-1-1: 提交代码到本地仓库,执行两条命令:**

Git add .

Git commit -m “my first commit”

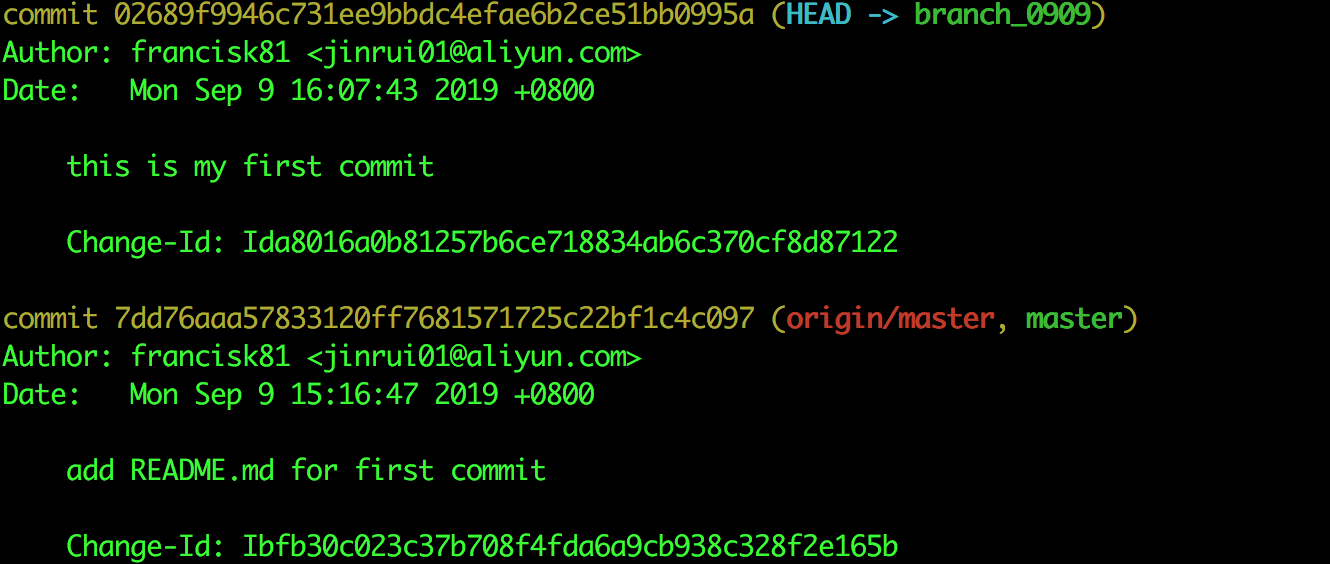
第一条命令git add. 代表将本地所有的改动添加到本地仓库的变更索引『index』中；第二条命令git commit 代表将本地所有的改动添加到本地仓库，-m是git commit的执行参数，后面””中的文本代表本次提交的附带消息，这样别人看到我们这次提交后，根据-m里的信息，就知道我们本次代码提交的目的了，执行过程如下:



提交成功后，我们看到系统提示『1 file changed, 1 insertion(+)』，代表着本次代码变更的简单记录。

**4-1-2: 查看本地提交记录，执行命令：**

Git log

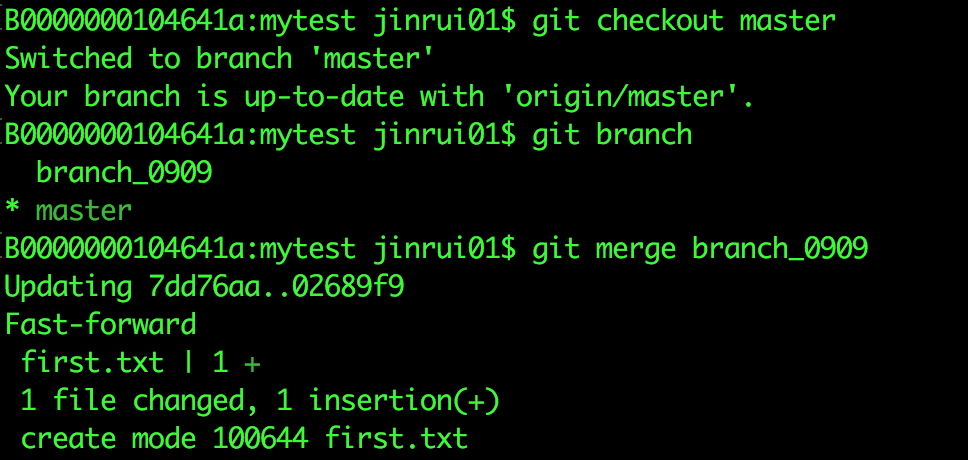


如上图所示，执行git log后，在命令行里会显示两条commit记录，自上而下按照提交时间排序，最上面的一条是最新提交的一条记录。在这里我们会看到两条记录，最上面一条就是我们刚才完成的代码提交；下面一条是我们第一次拷贝icode的脚本执行时的提交记录。

**4-1-3 切换到本地的master分支，将刚才在分支上的改动同步到master分支，执行下列几条命令**

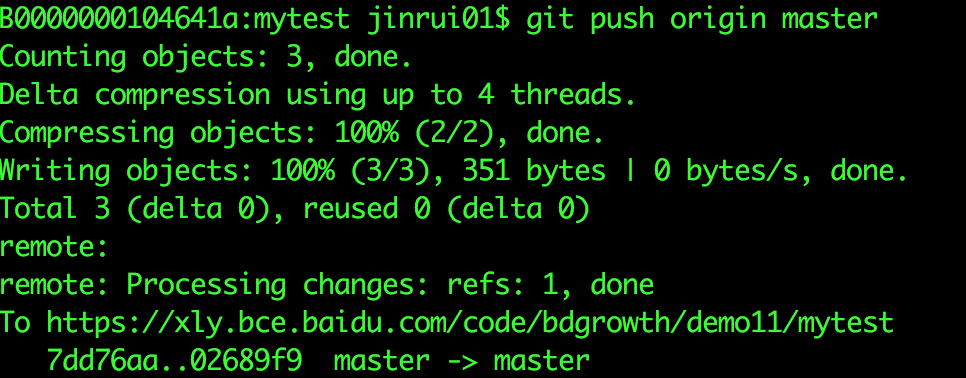
Git checkout master **(切换回master分支)**

Git merge branch\_0909 **(将branch\_0909上的改动合并到master分支)**

****

**4-1-4 将代码提交到iCode所在的远程代码库, 执行命令:**

Git push origin master



此时我们刷新icode页面，发现刚才在本地添加的first.txt已经出现在icode的系统中，说明我们本地的代码变更已经推送到icode所在的远程代码库了。

### 总结:

本地实验使用的git 命令包括:

Git branch

Git checkout

Git add

Git commit

Git push

同学们可以下载git cheat sheet获得更多关于git命令的操作，git cheat sheet在网上有很多种下载方式。同学们可以关注”百度效率云官方公众号”,输入”git-cheat”，在弹出的网盘小程序中获取PDF版本

1. 注: 项目标识在同一组织下唯一，各位同学在起名时如果出现上述提示: 说明项目标识或名称已经被其它人使用，换一个名称或标识即可 [↑](#footnote-ref-1)
2. 对于使用windows的同学，需要在本地安装git-bash工具，bash是Linux系统中的shell命令，具体指令可以参考这篇文章: <https://blog.csdn.net/francisk84/article/details/100582270> [↑](#footnote-ref-2)