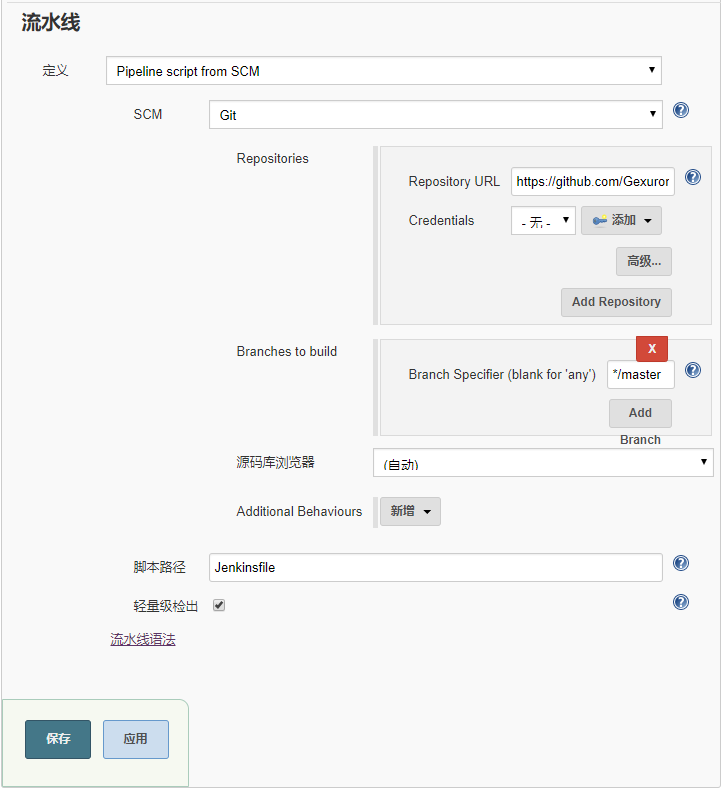
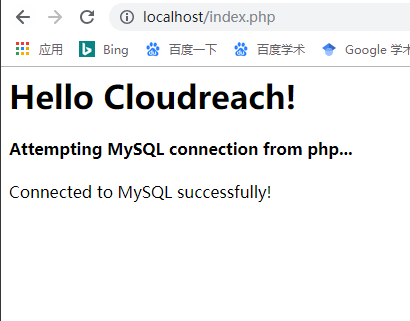
CICD文件夹下是一个完整的CI、CD的demo。

**使用方式：**

1. 在Jenkins中创建一个pipeline，（新建item-流水线），名字随意。在流水线区域将定义改为：“pipeline script from scm”。这一步的目的是让流水线根据远程仓库上Jenkinsfile中的内容执行操作。之后填入git仓库地址。脚本路径是指Jenkinsfile所在的路径，默认的“Jenkinsfile”是指Jenkinsfile在根目录下且名字就是“Jenkinsfile”。



1. Build now，在build history中会出现新的构建历史，点击数字旁边的小三角可以看console output。
2. 在不出意外的情况下，Jenkins会自动从git仓库中拉取代码，之后打印一个hello world，之后自动创建一个php-apache-mysql的镜像。（Windows环境下，MAC环境需要将相应的bat文件改成sh文件。本机需要先装docker。在流水线开始之前保证端口80和3306没有占用。）
3. 在浏览器中输入“localhost:80/index.php”，应该可以看到页面。表示部署成功。



注：这个Jenkins流水线只完成了部署的任务，照理来说应该还有build和test。因为php不需要build，所以就没那一步。之后Test我还没有写。所以目前这个流水线执行完之后不会自动关闭，而是一直运行这一个docker 容器。如果要关闭就手动从cmd里关。

docker stop $(docker ps -a -q)--🡪停止所有运行的容器。

或者

docker stop apache

docker stop php

docker stop mysql

-🡪停止本流水线运行的三个容器。

关于Docker：

采用了<https://github.com/mzazon/php-apache-mysql-containerized.git>这个仓库上的例子。使用了yml来创建多个容器并将其结合在一起。

Php文件放在public\_html文件夹下，就能跑。

关于Jenkinsfile：

这个Jenkinsfile的逻辑非常简单，就是单纯的调用一个之前已经写好的bat。所以如果是其他系统，换成相同功能的.sh，并且修改Jenkins的调用就可以了。（还是Linux比较好用讲道理但我不会用……）

关于 setup.bat:

是全部部署信息。首先会输出本机docker版本到a.txt，诚然这步没有什么用。

然后会删除冲突的container，之后部署自己的container。结束。