**OnlineJudge正式部署说明**

**正式部署说明**

以下均是在 Ubuntu 14.04 64位系统上进行的测试。由于内核版本和Python版本等问题，建议不要使用更低的系统版本，如12.04。

首先选择一下安装目录，我选择的是 /home/ 下，下面的操作不加特殊说明的话，都是相对这个目录的。如果需要更改，请自行替换所有的路径。

如果不是 root 用户，下面的命令需要的自行添加 sudo。

以下的安装需要git、curl、pip、vim，

请运行apt-get update && apt-get install -y git curl python-pip vim

**下载源代码**

git clone https://github.com/QingdaoU/OnlineJudge.git /home/OnlineJudge

**安装 Docker 和 docker-compose**

因为国内特殊网络环境，Docker 的安装和使用并不方便，我们使用 [DaoCloud](https://www.daocloud.io/) 的安装镜像。但是有时候也会出现添加 key 失败的问题，这时候可以使用[官方的安装方案](https://docs.docker.com/engine/installation/linux/ubuntulinux/)。

curl -sSL https://get.daocloud.io/docker | sh

安装 docker-compose

pip install docker-compose

**配置目录映射**

Docker 在运行的时候，如果没有配置目录映射，数据都是保存在 Docker 容器里面的，如果 Docker 容器被删除，数据就会丢失，所以需要将容器内的数据映射到服务器上存储。

我们使用下面的文件夹进行映射

* /home/data/mysql MySQL 的数据文件
* /home/data/redis Redis 的持久化文件
* /home/test\_case 上传的测试用例
* /home/log 各种日志
* /home/upload 上传的图片等

请自行创建相关的文件夹。

命令参考

mkdir -p /home/data/mysql /home/data/redis /home/test\_case /home/log /home/upload

**build 需要的镜像**

首先需要 MySQL 和 Redis 的镜像，可以从阿里云镜像 pull 下来（**推荐国内服务器使用，非常快**）或者从docker hub pull（**推荐国外服务器使用，国内比较慢**）。**选择一种就可以**

* 从阿里云镜像pull

docker pull registry.aliyuncs.com/v-image/redis

docker tag registry.aliyuncs.com/v-image/redis redis

docker pull registry.aliyuncs.com/v-image/mysql

docker tag registry.aliyuncs.com/v-image/mysql mysql

* 从docker hub pull

docker pull mysql

docker pull redis

注意:

* 在服务器上运行 MySQL 至少需要1G内存，否则很容易出现异常退出的问题。

OJ 需要的镜像，可以从阿里云镜像 pull 下来（**推荐国内服务器使用，非常快**）或者自己 build（**推荐国外服务器使用，国内比较慢**）。**选择一种就可以**

* 从阿里云镜像pull

docker pull registry.aliyuncs.com/v-image/oj\_web\_server

docker pull registry.aliyuncs.com/v-image/judger

docker tag registry.aliyuncs.com/v-image/oj\_web\_server qduoj/oj\_web\_server

docker tag registry.aliyuncs.com/v-image/judger qduoj/judger

* 自己build

docker build -t qduoj/oj\_web\_server /home/OnlineJudge/dockerfiles/oj\_web\_server/

docker build -t qduoj/judger /home/OnlineJudge/dockerfiles/judger/

运行 docker images 就能看到所有的镜像了。

**配置 docker-compose**

Docker Compose 是在使用 Docker 容器部署分布式应用时的工具，可以定义哪个容器运行哪个应用。使用Compose，你只需定义一个多容器应用的 yml 文件，然后使用一条命令即可部署运行所有容器。

在 dockerfiles/oj\_web\_server和dockerfiles/judger分别有一个 docker-compose.example.yml。请各复制一份为docker-compose.yml。不要删除docker-compose.example.yml文件。（可删可不删）

先解释下部分参数

* image 使用的镜像
* volumes 目录映射，将冒号前面的服务器上的映射到冒号后面容器内的路径
* env 容器内环境变量
* link 两个容器的连接
* ports 端口映射

如果你配置的映射目录或者端口有变化，修改对应的行就行。里面值为{YOUR\_PASSWORD}（**注意从这开始，下文所有的{XXX}都是代表一个变量，使用时请去除大括号**）是需要自己修改或者配置的密码，其中MySQL的可以随机设置一个。（不用改任何属性）

**启动所有的容器**

运行 docker-compose up –d

先在judger文件夹，然后在oj\_web\_server文件夹，都进行运行

docker ps -a可以看到所有的容器的运行状态。如果都是up的状态，说明成功启动了。如果有exited的状态，请先查看 CONTAINER ID，然后运行docker logs {CONTAINER\_ID}查看错误提示。还可以去log文件夹看日志，修复了问题之后，docker-compose up -d重启容器。

**创建数据表**

需要进入 oj\_web\_server 容器，docker ps -a （一共有四个id）可以看到这个容器是 running，然后复制 CONTAINER ID

运行（写命令时，写入对应的id即可，不要加花括号）

docker exec -i -t {CONTAINER ID} /bin/bash

然后容器中运行

python tools/create\_db.py

python manage.py migrate

python manage.py migrate --database=submission

python manage.py initadmin

最后一个命令是创建了超级管理员用户，密码是随机生成的，请妥善保管。如果忘记超级管理员root密码，也可以使用 initadmin 命令，重新生成超级管理员账号密码，之前的账号密码作废。

运行 python tools/release\_static.py，会生成压缩版的js，可能会比较慢，耐心等候。

然后 exit 退出 docker 容器，docker restart {CONTAINER ID}重启。

**配置nginx**

服务器上安装nginx，然后创建下面的配置文件，使用vim在此文件夹 /etc/nginx/conf.d/oj.conf ，创建以下内容

listen为端口，80与其他冲突，不能用，建议换其他端口，其他不动

server {

listen 86;

# 如果没有域名,就写你当前的ip或者使用 Host 绑定

server\_name xxxxxxx.com;

location /static/upload {

expires max;

alias /home/upload;

}

location /static {

etag off;

expires max;

alias /home/OnlineJudge/static/release;

}

location / {

client\_max\_body\_size 100m;

proxy\_pass http://127.0.0.1:8080;

proxy\_set\_header Host $host;

}

}

nginx -t 可以测试配置文件是否正确,

然后 service nginx restart 即可。

**server安装成功**

访问http://127.0.0.1:86如果可以访问了,请在 /login/ 处登录 root 用户,然后访问 /admin/ 添加一道题目试试吧。但是现在还不能判题。

**配置判题**

同上，在 dockerfiles/judger 文件夹也有一个 docker-compose.examlple.yml 文件，请先复制一份为docker-compose.yml。其中 {YOUR\_PASSWORD} (为修改判断的口令)自行修改，后面会用到。

然后运行

docker-compose up -d

* 如果判题和web服务器在同一台服务器上，请运行 docker inspect {JUDGER\_CONTAINER\_ID}（container\_id的获取方法见上文），查看它的ip地址，端口为8080。（在网页中配置一下判断机的属性，需要用到ip，端口，口令）
* 如果判题和web服务器不在同一台服务器上，ip为judger服务器的公网ip，端口为8085。

在 admin 界面 判题服务器 tab 里面，填写ip、端口和密码，创建一个判题服务器就可以了。

**网站名称和smtp密码**

请自行修改oj/custom\_settings.py文件

注意：

每次打开虚拟机需要开启容器，先在judger文件夹，然后在oj\_web\_server文件夹，都进行运行docker-compose up –d，然后运行docker ps –a，即可看到所有的容器的运行状态（一共是四个容器）