

天池布匹缺陷检测比赛 Docker 提交教程

教程参考：天池官方资料 <https://tianchi.aliyun.com/competition/entrance/231759/tab/226>

一、安装 docker

sudo apt install docker.io

```
root@iZ2ze2m4m2ejsfo4g70nuiZ:~# apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (18.09.7-0ubuntu1~18.04.4).
The following packages were automatically installed and are no longer required:
  libopts25 snmp
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 78 not upgraded.
```

验证安装是否成功：

docker info

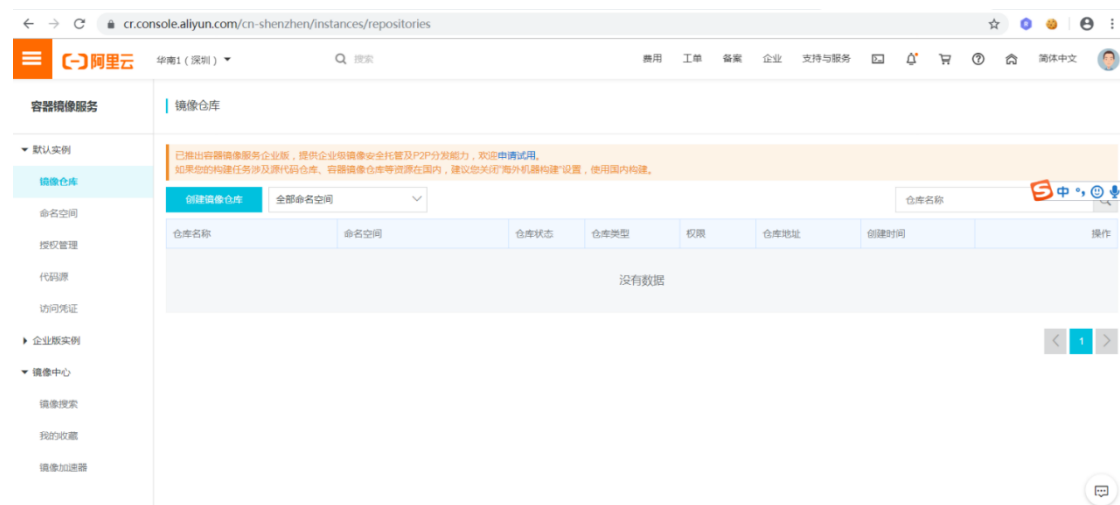
```
(base) root@iZuf6iqguqrattddczvocf2Z:~# docker info
Containers: 96
  Running: 3
  Paused: 0
  Stopped: 93
Images: 100
Server Version: 18.09.7
```

二、创建镜像仓库

这里以申请阿里云容器镜像服务（免费），并创建仓库为例，其他仓库如 dockerhub、谷歌、亚马逊、腾讯等详见对应产品说明书。

阿里云容器服务地址为 (<https://cr.console.aliyun.com>)

注册开通后产品页面如下



第一步切换标签页到命名空间, 创建地址唯一的命名空间



根据大赛要求选择对应的地域, 其他的按照自己需求选择或填写



下一步, 选择本地仓库, 不建议其他选项, 完成创建。

创建镜像仓库



1

仓库信息

2

代码源

代码源

您可以通过命令行推送镜像到镜像仓库。

上一步

创建镜像仓库

取消

点击管理，可查看详情。

仓库名称	命名空间	仓库状态	仓库类型	权限	仓库地址	创建时间	操作
test_for_tianchi_submit	test_for_tianchi	● 正常	私有	管理	📄	2019-09-29 15:01:23	管理 删除

<

test_for_tianchi_submit

华南1 (深圳) | 私有 | 本地仓库 | ● 正常

基本信息

仓库授权

触发器

镜像版本

镜像同步

基本信息

仓库名称 test_for_tianchi_submit

仓库地域 华南1 (深圳)

仓库类型 私有

代码仓库 无

公网地址 ● registry-cn-shenzhen.aliyuncs.com/test_for_tianchites [复制](#)

专有网络 ● registry-vpc-cn-shenzhen.aliyuncs.com/test_for_tianchi [复制](#)

经典网络 ● registry-internal-cn-shenzhen.aliyuncs.com/test_for_tianchi [复制](#)

摘要 ● 天池大赛提交演示

[操作指南](#) [镜像描述](#)

按照页面的指令在本地完成登陆:

操作指南

镜像描述

1. 登录阿里云Docker Registry

```
$ sudo docker login --username=***** registry.cn-shanghai.aliyuncs.com
```

用于登录的用户名为阿里云账号全名，密码为开通服务时设置的密码。

您可以在访问凭证页面修改凭证密码。

```
sudo docker login --username=***** your_registry_url
```

三、构建镜像并推送

在安装好 Docker 环境的本机/服务器构建并推送容器镜像。过程中可能会使用 docker 命令，如拉取 docker pull，推送 docker push，构建 docker build 等等。

为简化构建镜像的难度，天池已准备了常用的 Python 基础镜像，可直接拉取使用，更多基础镜像说明

<https://tianchi.aliyun.com/forum/postDetail?spm=5176.12586973.0.0.176a22322ijx xm&postId=67720>

pytorch:

registry.cn-shanghai.aliyuncs.com/tcc-public/pytorch:latest-py3

registry.cn-shanghai.aliyuncs.com/tcc-public/pytorch:latest-cuda9.0-py3

registry.cn-shanghai.aliyuncs.com/tcc-public/pytorch:1.1.0-cuda10.0-py3

registry.cn-shanghai.aliyuncs.com/tcc-public/pytorch:1.4-cuda10.1-py3

这里我们选择 pytorch1.4 的镜像

registry.cn-shanghai.aliyuncs.com/tcc-public/pytorch:1.4-cuda10.1-py3

1. 准备所需文件

(1) 新建一个文件夹（例如 tianchi_submit_demo）用于存放这次任务镜像所需的文件，文件夹中内容示例，其中在 defect_detection 文件夹中所有内容复制到

tianchi_submit_demo 文件（除了数据），也可以将 defect_detection 改名为

tianchi_submit_demo

```
jiangbx@ubuntu:~/work/dock/tianchi_submit_demo$ ls
convertTrainLabel.py  data_process  Dockerfile  inference  README.md  results.txt  run.sh  test_data  train.py  weights
data                  detect.py     hubconf.py  models    requirements.txt  runs        tcdata  test.py   utils
```

(2) 新建一个 Dockerfile 文件（这步很重要，注意是新建一个文件 Dockerfile 在

tianchi_submit_demo 中）文件内容如下：

```
# Base Images
## 从天池基础镜像构建
FROM registry.cn-shanghai.aliyuncs.com/tcc-public/pytorch:1.4-cuda10.1-py3

## 把当前文件夹里的文件构建到镜像的根目录下
ADD . /

## 指定默认工作目录为根目录（需要把 run.sh 和生成的结果文件都放在该文件夹下，提交后才能运行）
WORKDIR /

## 镜像启动后统一执行 sh run.sh
CMD ["sh", "run.sh"]
```

2. 构建镜像

执行

```
docker build -t registry.cn-shenzhen.aliyuncs.com/test_for_tianchi/test_for_tianchi_submit:1.0 .
docker build -t registry.cn
shenzhen.aliyuncs.com/test_for_tianchi/test_for_tianchi_submit:1.0 .
```

注意：registry.~~~是上面创建仓库的公网地址，用**自己仓库地址**替换。地址后面的：

1.0 为自己指定的版本号，用于区分每次 build 的镜像。最后的.是构建镜像的路径，不可以省掉。

```

root@iZ2ze2m4m2ejsfo4g70nuiZ:~/tianchi_submit_demo# docker build -t registry.cn-shenzhen.aliyuncs.com/test_for_tianchi/test_for_tianchi_submit:1.0 .
Sending build context to Docker daemon 4.096kB
Step 1/4 : FROM registry.cn-shanghai.aliyuncs.com/tcc-public/python:3
3: Pulling from tcc-public/python
c5e155d5a1d1: Pull complete
221d88d08ae9: Pull complete
4250b3117dca: Pull complete
3b7ca19181b2: Pull complete
425d7b2a5bcc: Pull complete
dc3049ff3f44: Pull complete
472a6afc6332: Pull complete
5f79c90f8d7c: Pull complete
1051ee813012: Pull complete
Digest: sha256:6268ecdc5f04d54bd411c8a64e49c714589e53ae402a49c6c12eaf91a5d0425
Status: Downloaded newer image for registry.cn-shanghai.aliyuncs.com/tcc-public/python:3
----> a4cc999cf2aa
Step 2/4 : ADD ./ /
----> e8a2742605bb
Step 3/4 : WORKDIR /
----> Running in 08a939d36879
Removing intermediate container 08a939d36879
----> 21328bd8aa39
Step 4/4 : CMD ["sh", "run.sh"]
----> Running in 60f7d4b07d53
Removing intermediate container 60f7d4b07d53
----> 1a48a53ae4a3
Successfully built 1a48a53ae4a3
Successfully tagged registry.cn-shenzhen.aliyuncs.com/test_for_tianchi/test_for_tianchi_submit:1.0
root@iZ2ze2m4m2ejsfo4g70nuiZ:~/tianchi_submit_demo#

```

3. 进入容器中配置环境

(1) 查看镜像的 ID

`docker images`

```

jlangbx@ubuntu:~/work/dock/tianchi_submit_demo$ docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
registry.cn-shanghai.aliyuncs.com/jlangbao/test_for_submit	1.2	600b3f758f14	10 hours ago	9.75GB

(2) 进入容器

`docker run -it 600b(image id 自己替换) /bin/bash`

```

jlangbx@ubuntu:~/work/dock/tianchi_submit_demo$ docker run -it 600b /bin/bash
root@ffabade154d6:/# ls
Dockerfile  bin  convertTrainLabel.py  data_process  dev  home  inference  lib64  mnt  opt  requirements.txt  root  run.sh /sbin  sys  test.py  usr  var
README.md  boot  data  detect.py  etc  hubconf.py  lib  media  models  proc  result.json  run  runs  srv  tcdat  test_data  train.py  utils  weights
root@ffabade154d6:/#

```

(3) 配置环境：镜像已经安装好了 torch1.4

只需要在安装 `cv2`、`matplotlib`、`scipy`

`pip install opencv-python`

`pip install matplotlib`

`pip install scipy`

有可能会出先 `import cv2 error` 错误

`ImportError: libGL.so.1: cannot open shared object file: No such file or directory`

`ImportError: libgthread-2.0.so.0: cannot open shared object file: No such file or directory`

解决方法：

`apt update`

`apt install libgl1-mesa-glx`

`apt-get install -y libglib2.0-0`

(4) 保存镜像:

1 退出但不关闭容器: ctrl + P + Q

2 保存容器为镜像

根据容器的 id (可以在上面的 root 后面信息看到) 比如这里就是 ffab

```
root@ffaba4e154d6:/# jiangbx@ubuntu:~/work/dock/tianchi_submit_demo$  
jiangbx@ubuntu:~/work/dock/tianchi_submit_demo$ docker run -it 2a49f /bin/bash
```

或者使用 docker ps

```
jiangbx@ubuntu:~/work/dock/tianchi_submit_demo$ docker ps  
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES  
ffaba4e154d6   600b      "/bin/bash"             11 minutes ago Up 11 minutes        loving_raman  
76959accad85   9ee78     "/bin/bash"             10 hours ago  Up 10 hours        strange_fernat  
64af295c6135   learn/ping "/bin/bash"             11 hours ago  Up 11 hours        quirky_kilby  
b4f388522311   045aa6    "/bin/bash"             11 hours ago  Up 11 hours        sad_thompson
```

docker commit ffab(容器 id 自己替换) registry.cn.....(自己的仓库地址).....:1.0

```
jiangbx@ubuntu:~/work/dock/tianchi_submit_demo$ docker commit 04f30 registry.cn.....bao/test_for_sunbit:1.0  
sha256:03cd77489a1d41cdcd602bda4a19b5043079b5a2116aeffaaf3670efde3b9ed8  
jiangbx@ubuntu:~/work/dock/tianchi_submit_demo$
```

其实就是上面 build 生成的镜像 repository: tag 号码
覆盖掉原来的镜像

4. 测试镜像

查看镜像 id docker images

```
jiangbx@ubuntu:~/work/dock/tianchi_submit_demo$ docker images  
REPOSITORY      TAG      IMAGE ID      CREATED        SIZE  
registry.cn-s... gbao/test_for_sunbit 1.0          03cd77489a1d 2 minutes ago  9.75GB
```

1 docker run your_image (自己替换) sh run.sh

```
jiangbx@ubuntu:~/work/dock/tianchi_submit_demo$ docker run 600b sh run.sh  
amespace(agnostic_ms=False, augment=False, classes=None, conf_thres=0.04, device='', img_size=1024, iou_thres=0.05, output='inference/output', save_dir='./', save_txt=False, source='/tcddata/guangdong1_r...  
und2_testB_20191024', update=False, view_img=False, weights='weights/best.pt')  
v4.0  
Using CPU  
/tcddata/guangdong1_round2_testB_20191024  
files: ['/tcddata/guangdong1_round2_testB_20191024/0818A1_0be69af1e895667202019081818405450K', '/tcddata/guangdong1_round2_testB_20191024/0818A1_0d9cfe9f022807102019081814140900K', '/tcddata/guangdong1_rou...  
nd2_testB_20191024/0818A1_0e9cc3986138a63602019081814223340K', '/tcddata/guangdong1_round2_testB_20191024/0818A1_0f323680c20f807002019081814103090K', '/tcddata/guangdong1_round2_testB_20191024/0818A1_0f5ac8...  
b1b9e646e02019081818383920K', '/tcddata/guangdong1_round2_testB_20191024/0818A1_0fe22dfb25be92d02019081818403060K', '/tcddata/guangdong1_round2_testB_20191024/0818A1_2a93baf172e747902019081814052520K', '  
/tcddata/guangdong1_round2_testB_20191024/0818A1_3de0eaa2a7f8372c0201908181340430K', '/tcddata/guangdong1_round2_testB_20191024/0818A1_445a7ed63eb61ad02019081814255150K', '/tcddata/guangdong1_round2_testB_...  
20191024/0818A1_5d01dbfa27e472d0201908181757320K']  
image: /tcddata/guangdong1_round2_testB_20191024/0818A1_0be69af1e895667202019081818405450K/0818A1_0be69af1e895667202019081818405450K.jpg: 0818A1_0be69af1e895667202019081818405450K  
result: {'name': '0818A1_0be69af1e895667202019081818405450K.jpg', 'category': 1, 'bbox': [2944.0, 1505.0, 3002.0, 1592.0], 'score': 0.16642743349075317}  
0818A1_0be69af1e895667202019081818405450K  
result: {'name': '0818A1_0be69af1e895667202019081818405450K.jpg', 'category': 1, 'bbox': [2874.0, 226.0, 2960.0, 324.0], 'score': 0.08032198995351791}
```

开始运行就是没问题了

4. 推送镜像

(1) 查看镜像 id docker images

```
jiangbx@ubuntu:~/work/dock/tianchi_submit_demo$ docker images  
REPOSITORY      TAG      IMAGE ID      CREATED        SIZE  
registry.cn-s... gbao/test_for_sunbit 1.0          03cd77489a1d 2 minutes ago  9.75GB
```

(2) 推送镜像

就是这个景象的 repository: tag 号码

docker push registry.cn-

shenzhen.aliyuncs.com/test_for_tianchi/test_for_tianchi_submit:1.0 (自己替换仓库地址)

```

jiangbx@ubuntu:~/work/dock/tlancht_submitt_demo$ docker push registry.cn-shanghai.aliyuncs.com/mgbao/test_for_sunbtt:1.2
The push refers to repository [registry.cn-shanghai.aliyuncs.com/mgbao/test_for_sunbtt]
e11d8e81c155: Layer already exists
9743ff33e8ea: Layer already exists
c22e4288fc2d: Layer already exists
ea78d5832506: Layer already exists
8ef82b3da275: Layer already exists
2cc3547a3251: Layer already exists
b0643e258249: Layer already exists
4d64a8bbbc06: Layer already exists
35ff166c35a9: Layer already exists
c0cb45b106e8: Layer already exists
486a27d7fdcd: Layer already exists
1db09913a256: Layer already exists
2e282f599fd6: Layer already exists
e6f174f76be4: Layer already exists
808f0332a58a: Layer already exists
b16caf11cb7d9: Layer already exists
37b9a4b22186: Layer already exists
e0b3afb09dc3: Layer already exists
6c01b5a53aac: Layer already exists
2c6ac8e5063e: Layer already exists
cc967c529ced: Layer already exists
1.2: digest: sha256:66eec88bad5916226c614b91ed7dc6d0ee9c6a85d78fc35df38d63751a4f8062 size: 4741

```

这样就上传成功了

四、提交镜像地址

在左侧【提交结果】中填写推送的镜像路径、用户名和密码，即可提交。根据【我的成绩】中的分数和日志可以查看运行情况。

首页>天池大赛>【入门】Docker练习场

状态	举办方	赛季1	奖金
2020-06-01	¥0		

【入门】Docker练习场

赛制

赛题与数据

排行榜

论坛

手把手教程

提交结果

我的成绩

我的团队

第一期

镜像路径：

暂无数据

请填写镜像相关信息

推荐使用阿里云私有镜像服务，公开镜像地址可不必填写用户名及密码。

镜像路径

用户名

密码

关闭 确定

配置路径

提交