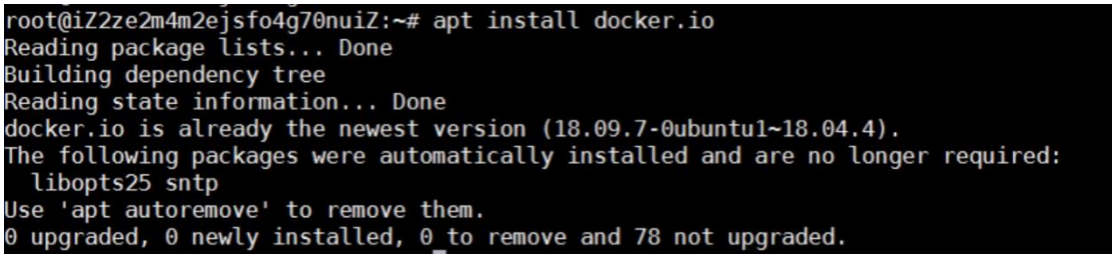
天池布匹缺陷检测比赛 Docker 提交教程

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



验证安装是否成功：

docker info

二、创建镜像仓库

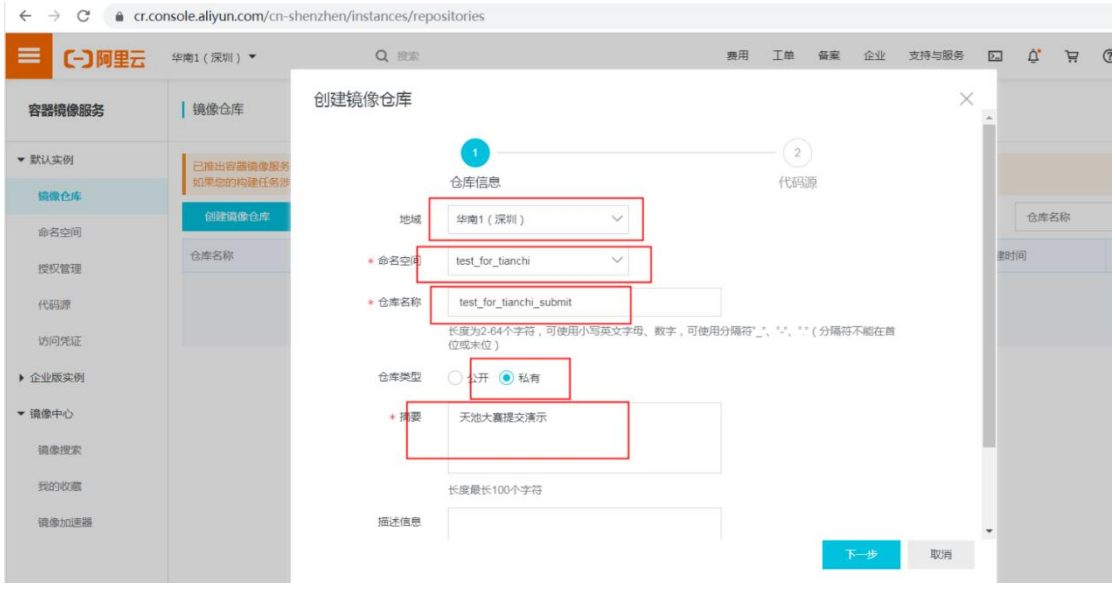
这里以申请阿里云容器镜像服务（免费），并创建仓库为例，其他仓库如 dockerhub、谷歌、亚马逊、腾讯等详见对应产品说明书。

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。

点击管理，可查看详情。

按照页面的指令在本地完成登陆:


```
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=5176.12586973.0.0.176a22322ijxxm&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 pytoch1.4 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 改名成 tianchi_submit_demo

（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
221d80d00ae9: Pull complete
4250b3117dca: Pull complete
3b7ca19181b2: Pull complete
425d7b2a5bcc: Pull complete
dc3049ff3f44: Pull complete
472a6afc6332: Pull complete
5f79c90f8d7c: Pull complete
1051ee813012: Pull complete
Digest: sha256:6268ecdce5f04d54bd411cba64e49c714589e53ae482a49c6c12eaf91a5d0425
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

```
jiangbx@ubuntu:~/work/dock/tianchi_submit_demo$ docker images
REPOSITORY                                                  TAG                 IMAGE ID            CREATED             SIZE
registry.cn-shanghai.aliyuncs.com/jiangbao/test_for_sumbit  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

```
jiangbx@ubuntu:~/work/dock/tianchi_submit_demo$ docker run -it  600b  /bin/bash
root@ffaba4e154d6:/# ls
Dockerfile  bin   convertTrainLabel.py  data_process  dev  home        inference  lib64  mnt    opt   requirements.txt  root  run.sh  sbin  sys     test.py    tmp   usr   var
README.md   boot  data                 detect.py     etc  hubconf.py  lib        media  models  proc  result.json       run   runs    srv   tcdata  test_data  train.py  utils  weights
root@ffaba4e154d6:/#
```

（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

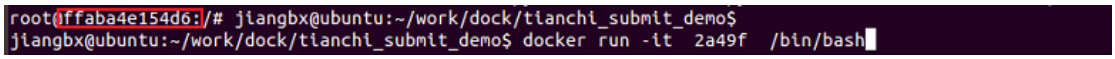
apt-get install -y libglib2.0-0

（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

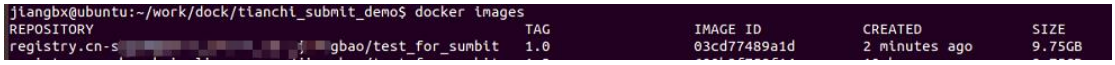


或者使用 docker ps

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

4．测试镜像

查看镜像 id　　docker images

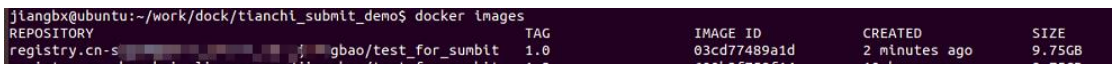


1 docker run your_image（自己替换） sh run.sh

开始运行就是没问题了

4．推送镜像

（1）查看镜像 id　　docker images



（2）推送镜像

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/tianchi_submit_demo$ docker push registry.cn-shanghai.ali[redacted]mgbao/test_for_sumbit:1.2
The push refers to repository [registry.cn-shanghai.a[redacted]bao/test_for_sumbit]
e11d8e81c155: Layer already exists
9743ff33e8ca: 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
808fd332a58a: Layer already exists
b16af11cbf29: 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: sha256:66eec88bad5916226c614b91ed7dcd60ee9c0a85d78fc35df38d03751a4f8062 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。