2022年微信大数据挑战赛-容器镜像提交说明

在复赛阶段，我们采用容器的镜像提交方式。请按如下要求，将本地代码打包成镜像提交，运行结束即可在提交页面查询运行日志及评测结果。

### 资源配置：

每次提交都会分配一个18核80GV100\*2的容器资源。

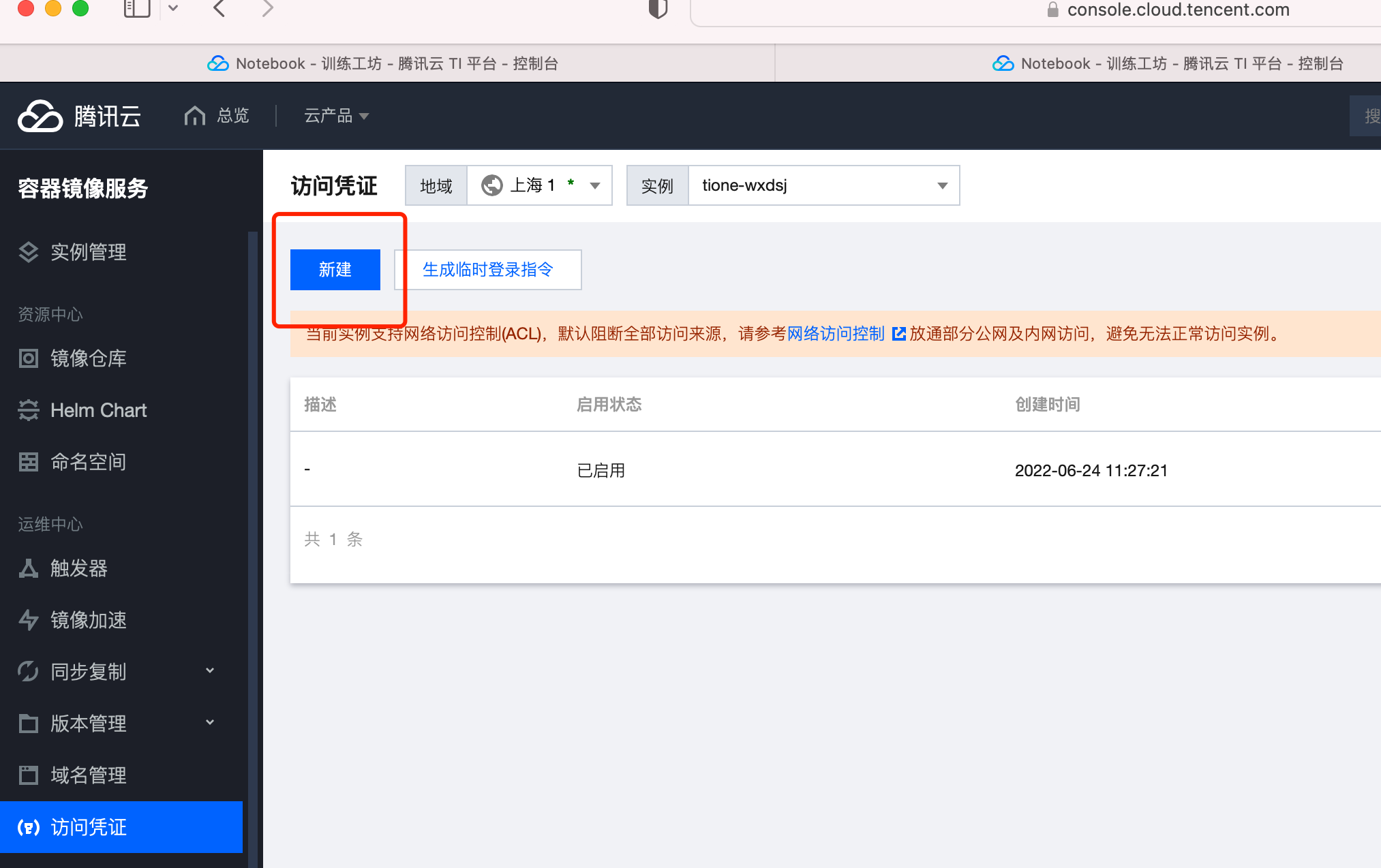
注：每只队伍在同一场大赛中同时只能存在一个运行中的任务。

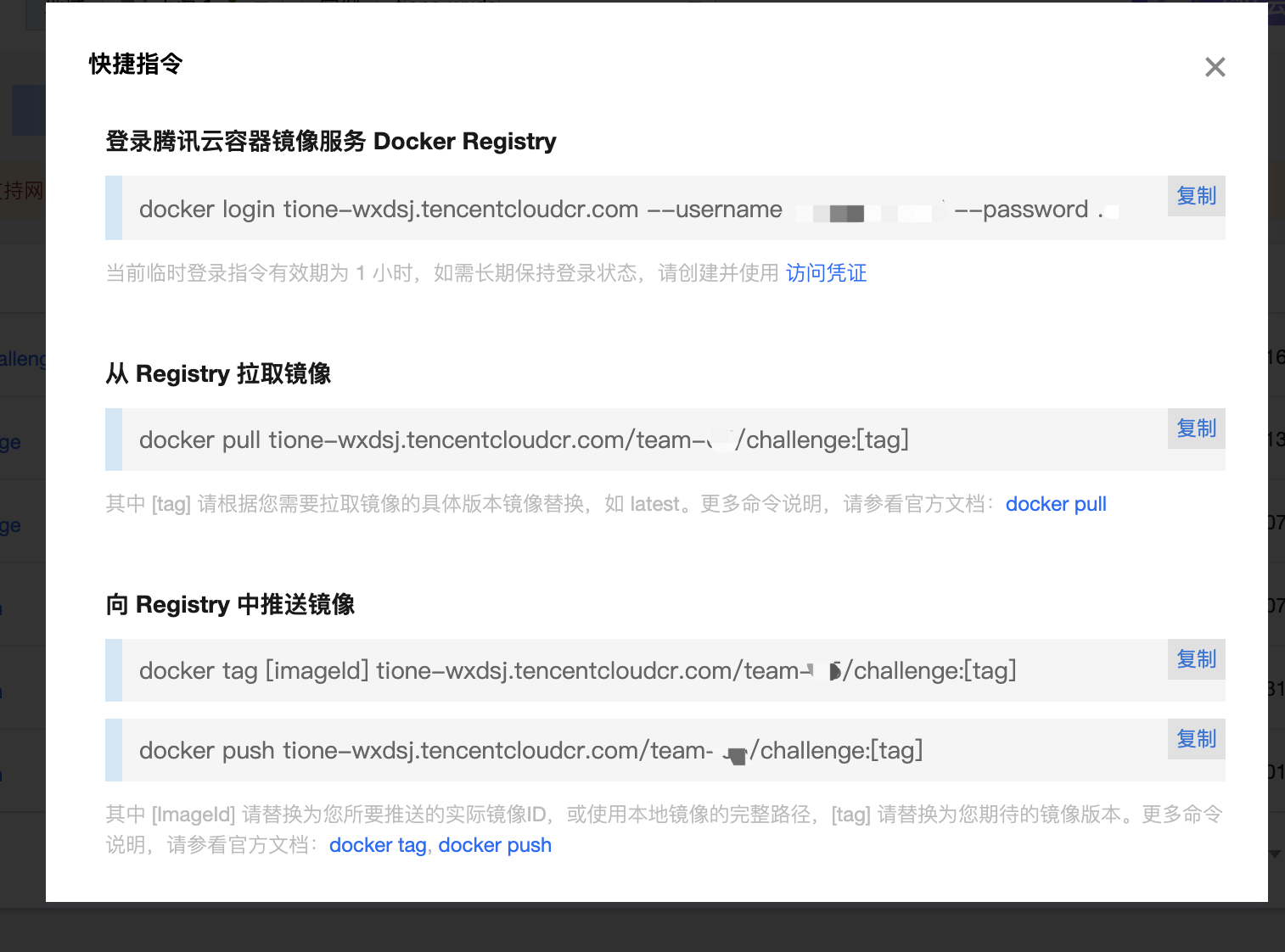
### notebook的docker本地环境：

notebook环境已经安装docker环境，选手可以直接在notebook中使用docker 命令，包括docker的pull build push run等命令，受安全限制run命令的-v、--privileged等参数已被禁止，其它可正常使用。

### 腾讯云容器镜像仓库

比赛提交镜像的仓库为腾讯云容器镜像仓库，大赛已经为每个队伍分配了一个命名空间。

<https://console.cloud.tencent.com/tcr/token> 选手在此处创建一个访问凭证（**地域请选择上海**）：



选手在notebook中登陆本队伍账号下的镜像仓库：

sudo docker login tione-wxdsj.tencentcloudcr.com --username （账户id） --password (访问凭证)

### Base镜像：

官方已经提供的base\_image:

选手可选择以下镜像作为基础镜像构建自己的模型镜像。

tione-wxdsj.tencentcloudcr.com/base/pytorch:py38-torch1.9.0-cu111-1.0.0

tione-wxdsj.tencentcloudcr.com/base/tensorflow:py37-tf1.15-cu10.0-1.0.0

tione-wxdsj.tencentcloudcr.com/base/tensorflow:py38-tf2.4-cu11.0-1.0.0

ccr.ccs.tencentyun.com/ti\_containers/pytorch:1.9.1-gpu-cu111-py38

### 镜像要求：

* 1. 入口脚本为sh /opt/ml/wxcode/start.sh；
  2. 镜像工作目录请设置为/opt/ml/wxcode；
  3. 数据输入目录为/opt/ml/input/data（官方将会把测试数据挂载至该目录下）；
  4. 输出结果至：/opt/ml/output/result.csv 格式和初赛一致；

### 测试数据集：

数据集挂载在/opt/ml/input/data/下，名称及内容格式如下

注： /opt/ml/input/data路径的权限为只读

/opt/ml/input/data

├── annotations

│ └── test.json

└── zip\_frames

├── test

│ ├── 000

│ │ ├── 13510838688783997000.zip

│ │ ├── ···

│ ├── 001

│ │ ├── 13510313293397887001.zip

│ │ ├── ···

│ ├── ···

### 运行效率限制：

在组委会指定的机器上（具体配置与复赛训练环境相同），预测给定视频的分类结果，记录预测时长。最低性能要求：QPS >= 70，即平均每秒钟至少预测 70 个视频。

### 运行日志：

为了方便选手进行问题定位，我们开放了运行日志，请到大赛官网提交页面查看日志。运行日志做了日志大小和行数限制，详见日志查看页面。

同时，请选手不要尝试打印复赛测试集内容、模型预测结果等行为，赛方会对选手日志内容进行检查，若发现选手有打印复赛测试集内容、模型预测结果等作弊行为，会终止其复赛资格。

### 示例Dockerfile:

FROM tione-wxdsj.tencentcloudcr.com/base/pytorch:py38-torch1.9.0-cu111-1.0.0

WORKDIR /opt/ml/wxcode

COPY ./opensource\_models ./opensource\_models

COPY ./save ./save

COPY ./requirements.txt ./

RUN pip install -r requirements.txt -i https://mirrors.cloud.tencent.com/pypi/simple

COPY ./\*.py ./

COPY ./start.sh ./

CMD sh -c "sh start.sh"

### docker提交流程示例：

* 在notebook登陆镜像仓库：

sudo docker login tione-wxdsj.tencentcloudcr.com --username （账户id） --password (访问凭证)

* 拉取基础镜像：

sudo docker pull tione-wxdsj.tencentcloudcr.com/base/pytorch:py38-torch1.9.0-cu111-1.0.0

* 比赛代码准备（这里使用官方提供的复赛baseline，代码位置已经准备存到了cfs的baseline/ 文件夹中）
* 构建镜像,进入到代码目录（目录下包含Dockerfile文件），然后构建模型镜像：

cd baseline

sudo docker build -t tione-wxdsj.tencentcloudcr.com/team-xxxxx/challenge:v1.0 .

Dockerfile内容：

------------------------------------------------------------------

FROM tione-wxdsj.tencentcloudcr.com/base/pytorch:py38-torch1.9.0-cu111-1.0.0

WORKDIR /opt/ml/wxcode

COPY ./opensource\_models ./opensource\_models

COPY ./save ./save

COPY ./requirements.txt ./

RUN pip install -r requirements.txt -i https://mirrors.cloud.tencent.com/pypi/simple

COPY ./\*.py ./

COPY ./start.sh ./

CMD sh -c "sh start.sh"

-----------------------------------------------------

编写Dockerfile时尽量把不易变动文件(如环境)放当前面，易变动的文件(如代码模型等)容易变动的放到后面，来加快构建和提交速度。

requirements.txt中必须加入coscmd以便上传测试输出结果。

* 代码中的路径规范：

镜像的启命令为：sh /opt/ml/wxcode/start.sh  官方将使用该命令运行选手的模型镜像。

测试数据输入：

test\_zip\_frames=/opt/ml/input/data/zip\_frames/test

test\_annotation=/opt/ml/input/data/annotations/test.json

测试结果输出：

test\_output\_csv=/opt/ml/output/result.csv

docker内将会挂载相应输入输出目录。

* 构建好的镜像push队伍的对应命名空间内（[登陆账号后在此处查看本队命名空间](https://console.cloud.tencent.com/tcr/namespace)）：

sudo docker push tione-wxdsj.tencentcloudcr.com/team-xxxxx/challenge:v1.0

tione-wxdsj.tencentcloudcr.com/[team-xxxxx](https://console.cloud.tencent.com/tcr/namespace)/challenge:v1.0 ,xxxxx替换为本队伍的账号id。

* 选手在大赛官网提交镜像地址，官方将部署选手的模型镜像，运行出结果后直接打分显示结果。



### 常见问题：

* 本次使用的腾讯云资源均在上海，notebook和容器镜像服务请选择上海资源。

根据选手的提问，定期更新中...