

上海公共数据开放平台-安全沙箱 操作手册

1. 前言.....	3
2. 沙箱申请.....	4
3. 上传镜像.....	6
3.1 快速开始.....	6
3.2 镜像仓库.....	8
3.2.1 公有镜像仓库.....	8
3.2.2 私有镜像仓库.....	9
3.3 镜像仓库使用.....	9
3.3.1 本地电脑打包和上传镜像.....	9
3.3.2 本地电脑下载镜像.....	9
4. 拉取平台数据.....	9
5. 平台内运行 Docker 容器.....	10
5.1 沙箱环境.....	10
5.2 登录沙箱环境.....	10
5.3 沙箱中运行镜像.....	10
5.3.1 使用 session 管理工具.....	10
5.3.2 运行镜像.....	11
6. 提交并下载结果.....	11
6.1 结果提交方式.....	11

版本变更记录

版本号	状态	变更时间	变更内容
V0.1	创建	2020/01/13	初稿

1. 前言

欢迎使用上海公共数据开放平台，为了使用户能够尽快熟悉本平台的安全沙箱功能，我们特意编写了本操作手册。本手册的目的是为用户提供系统操作指导，面向的对象是初次使用本平台并具有一定 Docker 使用基础的用户。对于初次接触 Docker 的用户建议先查阅 Docker 官网 (<https://www.docker.com>) 相关文档，或自行搜索 Docker 相关基础教程，对 Docker 有一定理解之后，再参阅本手册。本手册按照运营平台的常规操作顺序，分为沙箱申请，上传镜像，运行镜像，提交并申请下载结果等几个部分。手册中重点事项以**红色文字标注**，请用户特别注意。

2. 沙箱申请

用户使用浏览器打开上海公共数据开放平台登录页面，输入在门户网站申请的账号和密码，并在“个人中心”通过认证，然后登录平台的“安全沙箱”模块中申请使用。

The screenshot displays the Shanghai Public Data Open Platform interface. At the top, the navigation bar includes '首页', '数据资源', '数据概览', '地图服务', '典型应用', '开放生态', '互动社区', and '安全沙箱'. The '安全沙箱' link is highlighted with a red box. Below the navigation bar, there is a search bar and a user profile 'wangzl66'. The main content area features an illustration of a '安全计算沙箱' (Security Calculation Sandbox) with various data visualization elements like charts and tables. To the left, there is a text block describing the sandbox's purpose and a '产品文档' link. At the bottom, a four-step process is shown: '注册账号' (Register Account), '实名认证' (Real-name Authentication), '申请沙箱' (Apply for Sandbox), and '线下提供账号' (Provide account offline). A red arrow points from the '安全沙箱' link to the '申请使用' (Apply for Use) button in the final step.

上海市公共数据开放平台

请输入数据资源的名称和关键词...

wangzl66

安全沙箱

安全计算沙箱

安全计算沙箱是上海市公共数据开放平台解决数据流通安全问题的可行方式之一，其核心价值在于解决数据开放主体和数据利用主体之间的数据互信使用、安全合规流通、数据价值如何赋能等问题。安全计算沙箱以“逻辑封闭，进出审计，内部自由”为功能特点进一步降低数据分析挖掘门槛。

产品文档

注册账号

实名认证

申请沙箱

线下提供账号

简单四步——即可使用沙箱环境

申请使用

点击上图申请使用后跳转至沙箱申请页面，选择所需要使用的沙箱时间段，点击提交，即可申请。

请选择有效期：*

选择	有效时间-起始	有效时间-结束	剩余可申请资...
<input type="radio"/>	2020-01-14	2020-01-17	0
<input type="radio"/>	2020-01-18	2020-01-18	3
<input type="radio"/>	2020-01-22	2020-01-25	3
<input type="radio"/>	2020-01-26	2020-01-29	4
<input checked="" type="radio"/>	2020-01-30	2020-02-02	5
<input type="radio"/>	2020-02-03	2020-02-06	5
<input type="radio"/>	2020-02-07	2020-02-10	5
<input type="radio"/>	2020-02-11	2020-02-14	5
<input type="radio"/>	2020-02-15	2020-02-18	4
<input type="radio"/>	2020-02-19	2020-02-22	4

沙箱套件 

CPU-标准版（默认）

CPU： 4核
内存： 8GB
系统盘： SSD本地盘20GB
数据盘： 普通云盘300GB（数据盘大小可调整）
功能： 包括网络隔离、权限控制、监控管理、运维管理等安全防护功能

[> CPU-升级版](#)

[> GPU-标准版（敬请期待）](#)

请阅读并同意 [安全沙箱用户协议](#)

提交

取消

注意：

1. 使用时间从有效起始日期的 0: 00 至结束日期的 23: 59， 到期后沙箱内的所有资料都会清理， 请妥善保存您的资料。
2. 同一个用户每个周期（3 天）只能申请一次，不可重复申请。
3. 申请成功后，同一个用户处于‘可使用’和‘排队中’状态的申请最多为 3 条。
4. 每个周期（3 天）最大申请人数为 5。

申请沙箱成功后，请按期到“个人中心 - 我的沙箱 - 沙箱预约”中 进入沙箱使用。

- 基本信息
- 数据使用
- 收藏订阅
- 应用管理
- 互动管理
- 我的沙箱
- 沙箱预约

我的沙箱 / 沙箱预约

沙箱预约

欢迎前往[首页](#)[意见反馈](#) 提交您关于沙箱使用的任何建议,帮助我们更好的提升服务质量

序号	提交时间	有效期-起始	有效期-截止	状态	操作
1	2020/1/13 14:26:20	2020/1/18	2020/1/21	可使用	查看 进入沙箱
2	2020/1/13 14:44:55	2020/1/22	2020/1/25	可使用	查看 进入沙箱
3	2020/1/13 14:45:0	2020/1/26	2020/1/29	可使用	查看 进入沙箱

进入后，上半部分显示沙箱配置以及 SSH 登录沙箱的操作，下半部分显示用户上传成功的镜像列表。



The screenshot shows a web interface for resource management. At the top, there's a button '下载用户操作手册'. Below it, the '资源管理' section shows 'SSH 配额使用情况 (已使用/总配额)' as '0/10 SSH'. A table lists a resource with ID '51' and specifications: 'CPU: 8核 / 内存: 48GB / GPU: P40 / 系统盘: 40GB SSD / 数据盘: 500GB SSD + 2TB HDD'. An 'SSH登录' button is next to it. The '基础信息' section has a '显示登录信息' button and a refresh icon. Below is a table of Docker images:

Docker镜像名称	下载次数	Pull 命令	更新时间	操作
flag_009/zal-docker	2	docker pull 120.132.13.81/flag_009/zal-docker:0.0.2	2020-01-19 14:49:15	Tag列表
flag_009/new-docker001	0	docker pull 120.132.13.81/flag_009/new-docker001:0.0.1	2020-01-19 11:39:54	Tag列表

3. 上传镜像

用户程序代码都会在 Docker 容器中运行，因此用户首先需要在本地构建自己的镜像，然后将镜像上传至平台镜像仓库，最后提交下载结果时运行脚本中也需包含对应的镜像地址。为了保障镜像安全，本平台使用专用的 Docker Hub（Docker 镜像仓库）用于存放镜像。

3.1 快速开始

用户进入平台之后，获取自己镜像仓库的帐号和密码(点击“显示登录信息”按钮查看)。之后即可通过自己本地电脑打包并推送（Push）镜像至私有镜像仓库。接下来通过一个简单的示例来演示如何在本地打包和上传一个镜像：

- a) 首先要在本地构建镜像，需要在本地准备目录下准备 Dockerfile 和 hello.sh 两个文件，目录结构为：

```
→ hub_demo git:(master) X tree
.
├── Dockerfile
└── hello.sh

0 directories, 2 files
```

- b) 其中 Dockerfile 内容为：

```
FROM centos:7.2.1511

COPY hello.sh /

CMD bash /hello.sh
```

hello.sh 内容为：

```
echo 'Hello world !'
```

c) 运行 `docker build -t hello_world .` 命令构建新的镜像

```
→ hub_demo git:(master) X docker build -t hello_world .
Sending build context to Docker daemon 3.072kB
Step 1/3 : FROM centos:7.2.1511
----> 4cbf48630b46
Step 2/3 : COPY hello.sh /
----> 35ff1e126803
Step 3/3 : CMD bash /hello.sh
----> Running in fad700539812
Removing intermediate container fad700539812
----> 21da98cd5b2a
Successfully built 21da98cd5b2a
Successfully tagged hello_world:latest
```

d) 登录平台，点击“显示登录信息”获取私有仓库帐号密码：

资源管理

SSH 配额使用情况 (已使用/总配额)

0/10 SSH

沙箱ID	资源	操作
13	CPU: 2核 / GPU: - / 内存: 4G / 磁盘总容量: 200G	SSH登录 交卷

基础信息 显示登录信息 刷新

Docker镜像名称	下载次数	Pull 命令	更新时间	操作
No data				

e) 登录私有镜像仓库，注意仓库地址为：cainiaohub.sf.ucloud.cn

```
→ hub_demo git:(master) X docker login hub.sf.ucloud.cn
Username: ga3overwatch_002
Password:
Login Succeeded
```

f) 推送构建好的本地镜像，注意推送前必须按指定格式为镜像打上 tag：

```
<!--为构建的镜像打上 Tag, 结构格式为 [hub 地址]/[大赛平台用户名]/[镜像名称]:[标签名称]-->
docker tag hello_world hub.sf.ucloud.cn/ga3overwatch_002/hello_world:latest

<!--推送镜像到 Hub-->
docker push hub.sf.ucloud.cn/ga3overwatch_002/hello_world
```

g) 推送完成之后可以在平台中查看已推送的镜像：

Docker镜像名称	下载次数	Pull 命令	更新时间	操作
ga3overwatch_002/hello_world	10	docker pull hub.sf.ucloud.cn/ga3overwatch_002/hello_world:latest	2019-04-10 10:22:35	Top
ga3overwatch_002/ubuntu	7	docker pull hub.sf.ucloud.cn/ga3overwatch_002/ubuntu:latest	2019-04-10 10:17:25	Top

- h) 点击“SSH 登录”按钮，打开沙箱主机终端页面，首先同样需要先登录私有镜像仓库（可使用左上角“粘贴”按钮复制私有镜像仓库的帐号密码）：

```

Last login: Wed Apr 10 12:17:15 2019 from 192.168.0.22
[chinaai@10-23-0-52 ~]$ docker login hub.sf.ucloud.cn
Username: ga3overwatch_005
Password:
WARNING! Your password will be stored unencrypted in /home/chinaai/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[chinaai@10-23-0-52 ~]$

```

- i) 直接复制页面中的命令 COPY 到沙箱中执行

Docker镜像名称	下载次数	Pull 命令
ga3overwatch_002/hello_world	0	<u>docker pull hub.sf.ucloud.cn/ga3overwatch_002/hello_world:latest</u>
ga3overwatch_002/ubuntu	0	docker pull hub.sf.ucloud.cn/ga3overwatch_002/ubuntu:latest

- j) 在沙箱中执行上传的镜像：

```

[root@192-168-0-192 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS              NAMES
[root@192-168-0-192 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
hub.sf.ucloud.cn/ga3overwatch_002/hello_world latest            21da98cd5b2a       About an hour ago
195MB
[root@192-168-0-192 ~]# docker run --rm hub.sf.ucloud.cn/ga3overwatch_002/hello_world
Hello world!
[root@192-168-0-192 ~]#

```

- k) 当出现“Hello world!”时表示镜像成功执行。

3.2 镜像仓库

本平台专用镜像仓库域名为 `cainiaohub.sf.ucloud.cn`，此镜像仓库分为公有镜像仓库和私有镜像仓库两部分。

3.2.1 公有镜像仓库

公有镜像仓库中已经预存了常用的标准镜像，如有需要用户可以下载这些标准化镜像，定制开发打包成自己的专用镜像之后再上传至自己的私有镜像仓库。目前仓库中已有的公共镜像列表可参考[附录 1](#)。

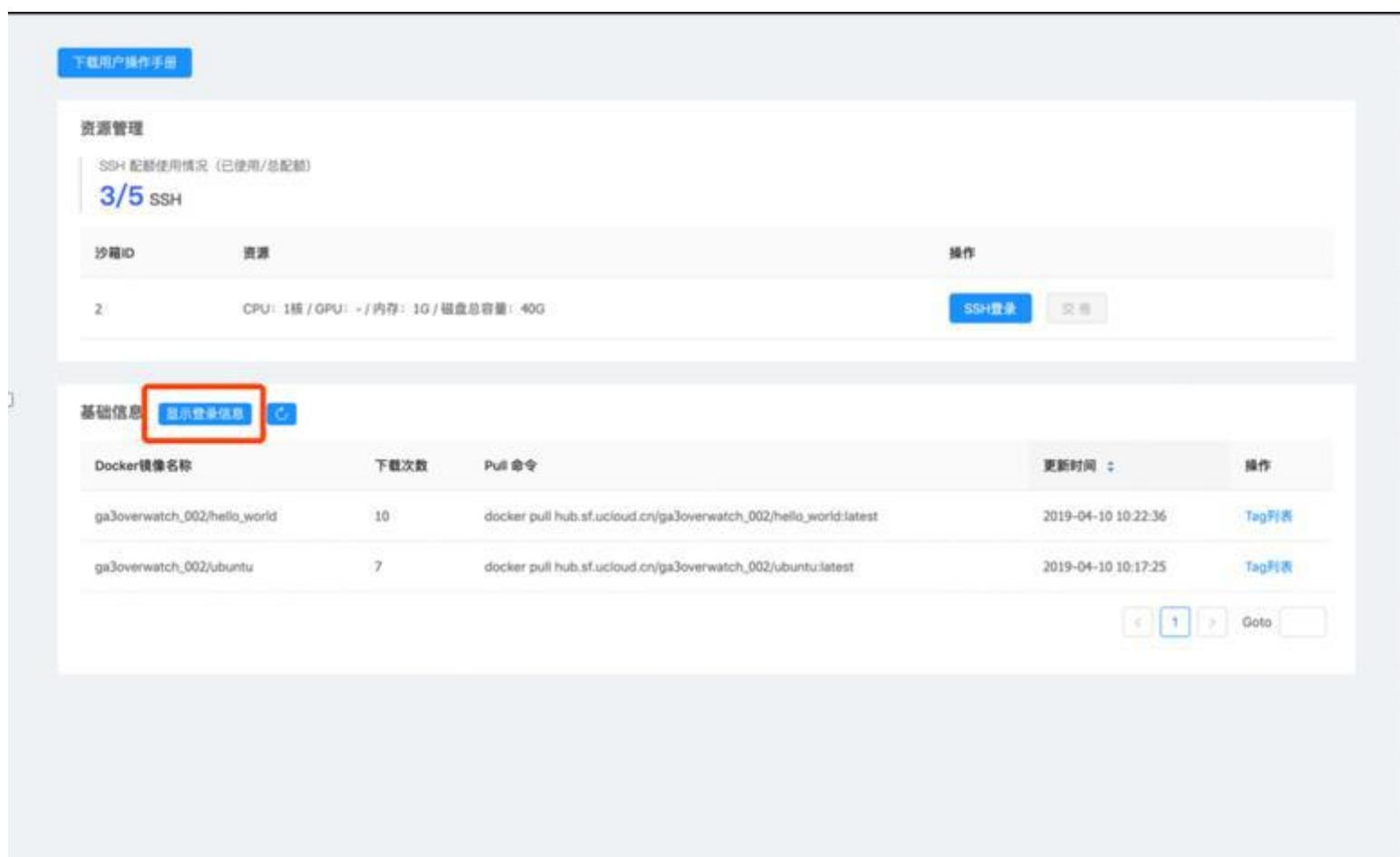
这些镜像存放在 `library` 项目下，以拉取镜像 `hello_world` 为例，可直接在本地电脑输入上述命令拉取镜像（需要首先[安装 docker 环境](#)）：

```
liu@SimondeMacBook-Pro.local ~
└─> docker pull hub.sf.ucloud.cn/library/hello_world
Using default tag: latest
latest: Pulling from library/hello_world
f2d1d709a1da: Pull complete
69cb9de3faed: Pull complete
Digest: sha256:0160d97fb2c61a58300ae7ff1248b137e1c7a62a4787f9f06dbfc96d19b6ae48
Status: Downloaded newer image for hub.sf.ucloud.cn/library/hello_world:latest
liu@SimondeMacBook-Pro.local ~
```

3.2.2 私有镜像仓库

平台为每个用户分配了私有镜像仓库，仓库名与用户登录平台的用户名相同。例如用户登录平台的**用户名**为 $\${username}$ ，则对应的私有镜像仓库地址为：`cainiaohub.sf.ucloud.cn/ ${username}`，私有镜像仓库推送镜像和拉取镜像需要先登录后再进行，详细步骤见 3.1 章节。

登录私有镜像仓库的账号和密码可在本平台中点击“**显示登录信息**”查看。



同时平台中还展示了私有镜像仓库内的镜像信息，包含镜像名称，下载次数，pull 命令地址，更新时间等信息，可供用户查看。

3.3 镜像仓库使用

3.3.1 本地电脑打包和上传镜像

可参考 3.1 章节快速开始。

3.3.2 本地电脑下载镜像

用户上传至私有镜像仓库的镜像可根据需要随时下载至本地。操作流程为：

1. 输入登录命令：`docker login cainiaohub.sf.ucloud.cn`
2. 输入用户名： $\${username}$
3. 输入对应私有仓库密码
4. 使用命令拉取自己的镜像：`docker pull cainiaohub.sf.ucloud.cn /${username}/hello_world` (需要已按 3.1 章节上传过 `hello_world` 镜像)

4. 拉取平台数据

用户使用用户名/密码 (`cainiao_data / cainiao_data`) 登录平台数据 SFTP 服务器 (`cainiaodownload.sf.ucloud.cn`) 拉取需要使用的数据。命令参考如下：

(1) 进入平台环境的 `/data1` 目录，**所有平台数据建议都放在该目录下** `$cd /data1`

(2) 使用用户名 `cainiao_data` 登录服务器 `$lftp sftp://cainiao_data@cainiaodownload.sf.ucloud.cn 口令: cainiao_data`

(3) 进入 contest_data 目录，该目录存放所有平台数据

```
>cd contest_data
```

(4) 使用 mirror 或 get 命令下载需要的数据

其中 mirror 用来下载文件目录，get 用来下载文件

```
>mirror abc
```

```
>get def.xlsx
```

中文字符不能直接在命令行中输入，可在左上侧“粘贴”对话框中输入中文进行粘贴操作

(5) 解压文件

对于数据中的压缩文件，可通过以下命令执行：

- 将 .7z 压缩包解压到当前目录

```
7za x file_name.7z
```

- 将 .rar 压缩包解压到当前目录

```
unrar e file_name.rar
```

- 将 .zip 压缩包解压到当前目录

```
zip file_name.zip
```

(6) 退出 LFTP

```
>exit
```

(7) 预览数据

对于 excel 格式数据文件，平台提供工具可预览文件内前 20 行数据。请参考以下使用方式：

第一步：从平台公共镜像仓库拉取工具到沙箱内部

```
docker pull cainiaohub.sf.ucloud.cn/library/read_excel
```

第二步：进入需要预览的文件所在目录，执行如下命令：

```
docker run -e file_name="2018 年 10 月网格数据.xlsx" -
```

```
v "$PWD":/app/read cainiaohub.sf.ucloud.cn/library/read_excel
```

注：进到要预览的 excel 文件所在的目录，修改 file_name，运行该容器即可。

5. 平台内运行 Docker 容器

5.1 沙箱环境

为了保护平台数据的安全性，本平台通过 Web SSH 的方式连接至一个独立的沙箱环境并提供了运行 Docker 的权限。

沙箱环境特点：

- 管控数据流入：不可从沙箱内主机直接访问外网服务，所有需要进入沙箱内部的软件和数据只能通过本地打包成 Docker 镜像->推送至平台镜像仓库->从沙箱内主机下载镜像的方式进行。
- 镜像仓库限制：为了保证数据安全，在沙箱内部只能使用 pull 命令拉取专用镜像仓库（包括公有和私有仓库）的镜像，而不能使用 push 命令推送镜像。
- 操作记录与审计：所有在沙箱中的操作行为都会被记录并供平台方进行审计，本平台保留追究法律责任的权力。

5.2 登录沙箱环境

本平台为用户设置了登录的配额，每个用户允许同时有 10 个终端登录。用户在资源管理区点击登录即可进入沙箱环境，并可在资源管理下方看见目前已使用的登录配额情况。



5.3 沙箱中运行镜像

5.3.1 使用 session 管理工具

进入到沙箱环境后，用户可使用终端命令行界面操作。



为了保证用户在网络断开之后程序仍能正常运行，**建议用户使用 screen 或 tmux 工具（沙箱内已安装）**。详细使用方式可参考[附录 2](#)或自行搜索相关教程。

5.3.2 运行镜像

- 下载提交的镜像：通过命令 `docker pull` 命令从私有仓库中拉取已推送的镜像，例如：
`docker pull cainiaohub.sf.ucloud.cn /${username}/hello_world`（需要已按 3.1 章节上传过 `hello_world` 镜像）
- 用户在调试环境时可直接使用 `docker run` 命令在沙箱中运行：
`docker run cainiaohub.sf.ucloud.cn /${username}/hello_world`
- 同时可使用 **-v 参数**将沙箱主机中目录挂载至 `docker` 中，例如：
`docker run -v /data1/sample:/data/sample cainiaohub.sf.ucloud.cn /${username}/hello_world`
其中“`-v /data1/sample:/data/sample`”参数就是将主机中“`/data1/sample`”目录挂载到 `hello_world` 中对应“`/data/sample`”目录下，这样用户就可以在 `docker` 内部对“`/data/sample`”中的数据进行处理。

6. 提交并下载结果

6.1 结果提交方式

运行结果在沙箱环境中通过 SFTP 命令行界面的方式提交。提交步骤为：

1. 通过 `sftp` 命令登录 ftp 服务器：`$ sftp ${username}@cainiaosftp.sf.ucloud.cn`，输入密码登录。（**sftp 服务器的帐号和密码与登录私有镜像仓库的帐号密码相同**）
2. 上传结果文件至 SFTP 服务器上：`sftp>put ./result1.csv ./`，`sftp>put ./run1.sh ./` 当文件不在当前目录时可以使用绝对路径：`sftp>put /path/runN.sh /result`

注意结果服务器的地址是 /result，所有结果文件都提交到该目录下

```
sftp> put 1.txt /result
Uploading 1.txt to /result/1.txt
1.txt                                     100% 14    23.8KB/s   00:00
sftp> ls
result
sftp> cd result
sftp> ls
1.txt
```

3. 全部结果上传完毕，需要最终确认上传结果：`sftp>ls`
4. 退出 SFTP 服务：`sftp>exit`

```
sftp> ls
result1.csv  result2.csv  result3.csv  result4.csv  result5.csv  run1.sh      run2.sh
run3.sh      run4.sh      run5.sh
sftp> exit
[chinaai@10-23-15-229 results]$
```

用户可登录 cainiaosftp.sf.ucloud.cn 下载结果，用户名和密码同[登录私有镜像仓库的帐号密码](#)。

附录 1：公共仓库镜像列表

[Pytorch]

```
docker pull cainiaohub.sf.ucloud.cn/library/pytorch/pytorch
docker pull cainiaohub.sf.ucloud.cn/library/pytorch/pytorch:1.2-cuda10.0-cudnn7-runtime
```

[TensorFlow]

```
docker pull cainiaohub.sf.ucloud.cn/library/tensorflow/tensorflow
docker pull cainiaohub.sf.ucloud.cn/library/tensorflow/tensorflow:latest-py3-jupyter
docker pull cainiaohub.sf.ucloud.cn/library/tensorflow/tensorflow:latest-py3
```

[Caffe2/caffe2]

```
docker pull cainiaohub.sf.ucloud.cn/library/caffe2/caffe2:snapshot-py2-gcc5-ubuntu16.04
docker pull cainiaohub.sf.ucloud.cn/library/caffe2/caffe2:snapshot-py2-cuda8.0-cudnn7-ubuntu16.04
```

[caffe2ai/caffe2]

```
docker pull cainiaohub.sf.ucloud.cn/library/caffe2ai/caffe2
docker pull cainiaohub.sf.ucloud.cn/library/caffe2ai/caffe2:c2v0.8.1.cpu.full.ubuntu14.04
```

[keras-jupyter]

```
docker pull cainiaohub.sf.ucloud.cn/library/ermaker/keras-jupyter
docker pull cainiaohub.sf.ucloud.cn/library/ermaker/keras-jupyter:py3
```

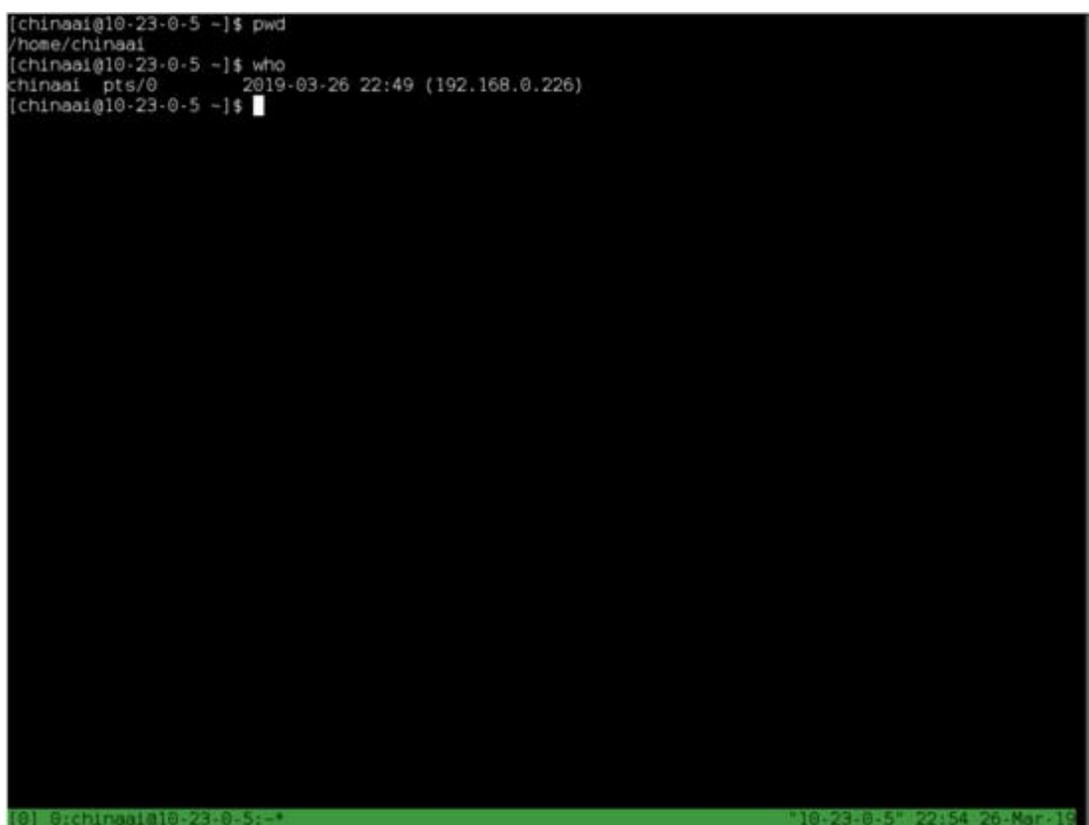
[theano]

```
docker pull cainiaohub.sf.ucloud.cn/library/spellrun/theano
docker pull cainiaohub.sf.ucloud.cn/library/spellrun/theano:8b4be4ac7
```

附录 2：如何使用 tmux

以 tmux 为例，用户在通过 SSH 登录按钮登录沙箱环境之后：

- a) 输入：tmux 回车，建立一个新的窗口并在新的窗口内执行各种终端命令。



```
[chinaai@10-23-0-5 ~]$ pwd
/home/chinaai
[chinaai@10-23-0-5 ~]$ who
chinaai pts/0      2019-03-26 22:49 (192.168.0.226)
[chinaai@10-23-0-5 ~]$
```

The screenshot shows a terminal window with a black background and white text. The user has executed the 'pwd' command, which returns '/home/chinaai'. Then, the user has executed the 'who' command, which returns 'chinaai pts/0 2019-03-26 22:49 (192.168.0.226)'. The terminal prompt is '[chinaai@10-23-0-5 ~]\$'. At the bottom of the terminal, there is a green status bar with the text '0] 8:chinaai@10-23-0-5:~*' and '10-23-0-5" 22:54 26-Mar-19'.

- b) 断开连接之后重新登录 SSH ，可输入：tmux ls 回车，显示上次建立的 tmux 窗口。

```
Last login: Tue Mar 26 22:49:34 2019 from 192.168.0.226
[chinaai@10-23-0-5 ~]$ tmux ls
0: 1 windows (created Tue Mar 26 22:49:43 2019) [100x39]
[chinaai@10-23-0-5 ~]$
```

c) 输入: `tmux a` 回车, 重新进入上次建立的 `tmux` 窗口即可恢复以前的工作。

```
[chinaai@10-23-0-5 ~]$ pwd
/home/chinaai
[chinaai@10-23-0-5 ~]$ who
chinaai pts/0      2019-03-26 22:49 (192.168.0.226)
[chinaai@10-23-0-5 ~]$
```

附录 3: 常见问题 Q&A

1. 如何提交结果?

结果需要提交至 SFTP 服务器: `cainiaosftp.sf.ucloud.cn`, 帐号密码与用户的 `docker hub` 的帐号密码相同, 具体可参见本手册 [6.1 节结果提交方式](#)。

2. Docker 容器内部如何访问主机上的文件?

可以在运行 `docker` 时使用参数 `-v /a:/b` 的方式挂载, 其中 `/a` 为主机上的文件夹绝对路径; `/b` 为 `docker` 容器内部文件夹的绝对路径。

3. SFTP 的帐号密码是多少?

SFTP 的帐号密码与 Docker 私有镜像仓库的帐号密码相同。

4. Docker 镜像可以重复上传吗?

在使用结束前 Docker 镜像暂不限制上传次数。

5. 如何将本地模型导入沙箱主机中?

用户只能将模型打包在 Docker 镜像中上传, 再在沙箱主机中下载运行此镜像。

6. 什么是 Docker 镜像? 什么是 Docker 容器? 什么是 Docker 仓库?

Docker 的镜像概念类似虚拟机的镜像, 包括运行容器所需的数据, 可以用来创建新的容器, 可使用 `docker images` 命令查看。Docker 容器类似虚拟机, 是由 Docker 镜像创建的运行实例, 可用 `docker ps -a` 命令查看。Docker 仓库用于在线存储 Docker 镜像, 类似 `git` 的代码仓库。

7. 为什么“SSH 登录”按钮是蓝色可点击状态, 我点击却没反应? 建议用户使用 **chrome 浏览器**, 并检查确认网络环境正常。

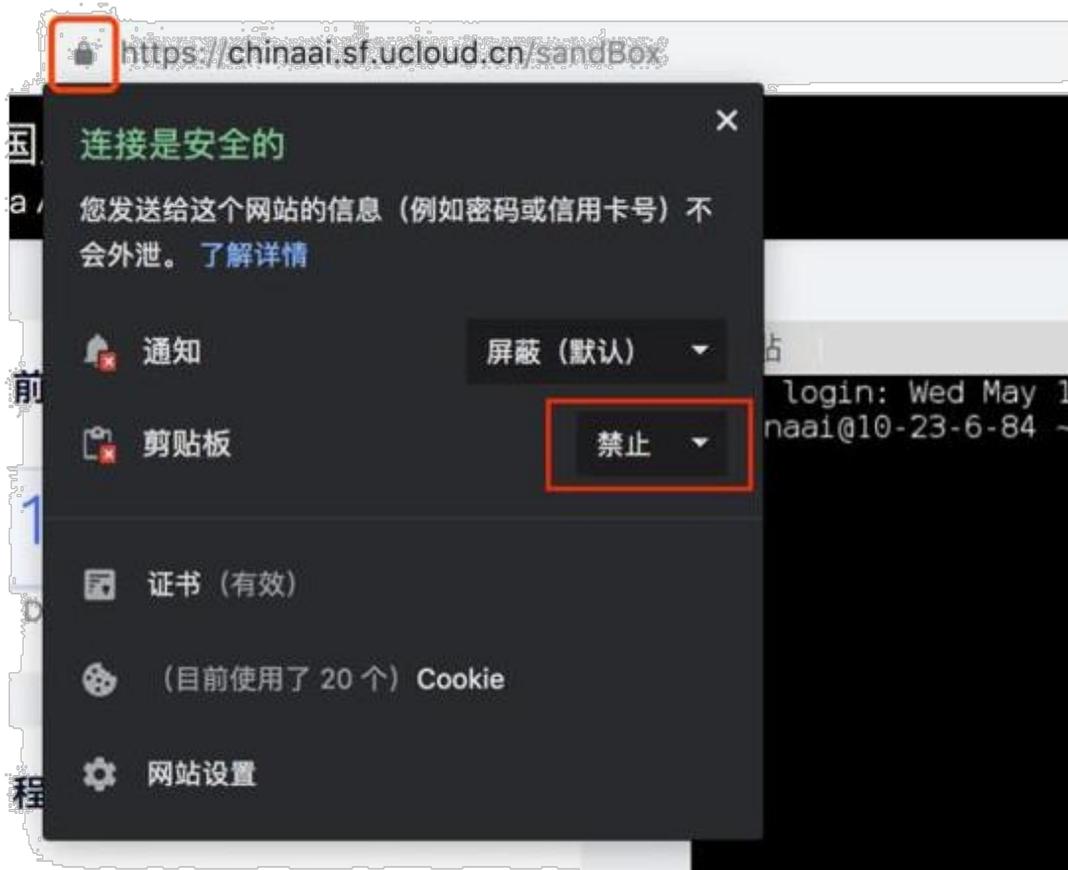
8. 为什么镜像信息显示 “No data” ?

因为镜像信息中显示的是用户私有镜像仓库中的镜像情况, 初始未上传镜像时其中没有任何数据。

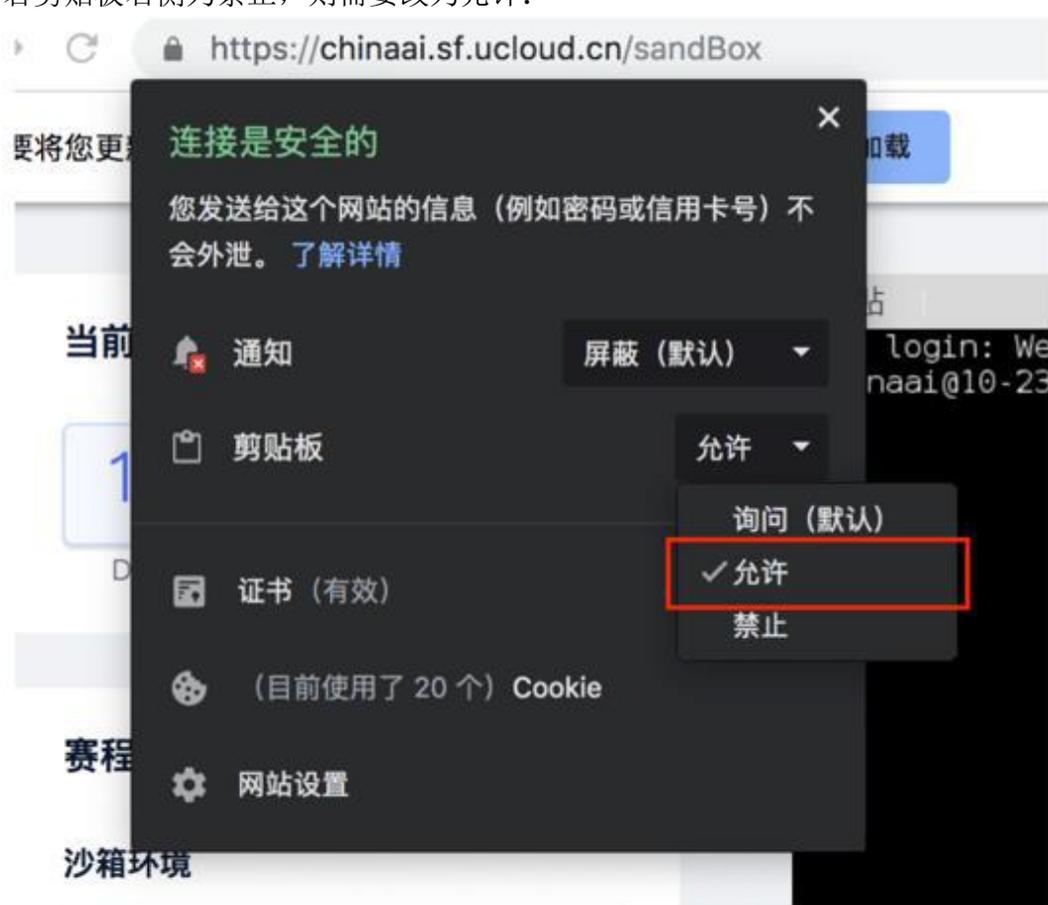
9. 沙箱操作页面点击粘贴按钮没有反应?

推荐使用谷歌 `chrome` 浏览器, 避免其他浏览器特性不一致导致的差异性问题的。如果使用谷歌 `chrome` 浏览器依然发生该问题, 请确认是否在配置中禁用了谷歌浏览器剪贴板的授权。

确认剪贴板是否被禁用, 点击浏览器地址栏左边 “ ” 图标:



若剪贴板右侧为禁止，则需要改为允许：



再重新点击粘贴按钮尝试。