

产品介绍

什么是 AI 实训平台

人工智能行业目前人才稀缺，一方面，相关人才培养难度大、周期长，但需求旺盛，另一方面，很多开发者希望向 AI 开发转型但无法找到合适的培训渠道。**OF 智能云瞄准这一业界痛点，推出 AI 实训平台，平台内置大量精品算法模型并进行代码级拆解，同时附带线上开发环境，让开发者边学边练，更易掌握 AI 模型研发的相关技术。**该产品也可作为高校及培训机构授课工具，提升学校进行 AI 人才培养的教学效率。

同时为开发者提供在线编程环境、GPU 算力、开源算法解读，帮助开发者快速创建和部署模型。AI 实训平台致力于成为大模型领域的业界权威社区和分享交流平台。

产品优势

1精品实训教程	2极简操作体验	3 适配主流 AI 框架	4丰富的开发模式
编写的高质量教程,教程和样例代码完整衔接,匹配高质量数据集。	支持 Fork 公开项目, 在线预览与编辑, 一键命令行部署模型。	集成深度学习的主流框架, 例 TensorFlow、PyTorch、OneFlow 等用户自定义环境。	支持Jupyter/WebIDE VScode在线开发、脚本任务开发、命令行 CLI 开发、Pycharm\VScode 等 IDE 开发。

快速上手

一键 Fork 项目

[例]迁移学习：以垃圾分类为例

1. 在 AI 实训平台的“项目”页，选中公开项目进入公开项目详情页。



2. 在项目详情页，点击【Fork】后，弹出 Fork 项目基本信息，用户可编辑，点击【提交】 Fork 项目。

* 项目名称 迁移学习：以垃圾分类为例1640001872850

* 项目描述 基于 MobileNet 的迁移学习，用于40类常见垃圾的垃圾分类。包含训练与推理脚本。

配图



取消

提交

3. Fork 成为我的项目后，用户可以修改或运行调试算法。

智能云 项目 数据集 镜像管理 帮助中心

我的项目 / 迁移学习：以垃圾分类为例1652239795395

迁移学习：以垃圾分类为例1652239795395

项目学习 脚本任务 oneflow-master+torch-1.9... 资源配置 自动命令 2022-05-11

基于 MobileNet 的迁移学习，用于40类常见垃圾的垃圾分类。包含训练与推理脚本。

修改 删除

概览 文件 数据详情 监控分析 在线推理 运行 部署

迁移学习：以垃圾分类为例

OneFlow 对 PyTorch 用户很友好

对于单机持卡训练，OneFlow 的体验已经与 PyTorch 非常接近。熟悉 PyTorch 的用户对 OneFlow 上手一定没有陌生感。还可以在此基础上体验最通用的分布式训练。以本项目为例，以下方式导入包就是 PyTorch 训练。

```
import torch
from torchvision import transforms, datasets
import torchvision
```

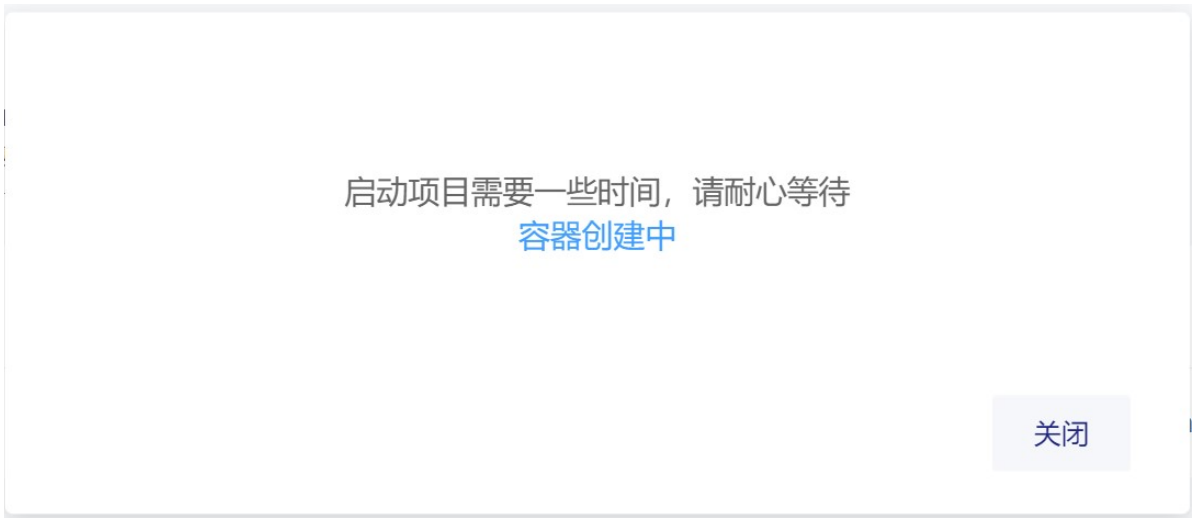
以下方式导入包就是 OneFlow 训练。

```
import oneflow as torch
from flowvision import transforms, datasets
import flowvision as torchvision
```

项目文件说明

4. 用户选择【运行】启动，前提需要保证用户账户余额大于 0，以及用户最大启动任务是 2 个。

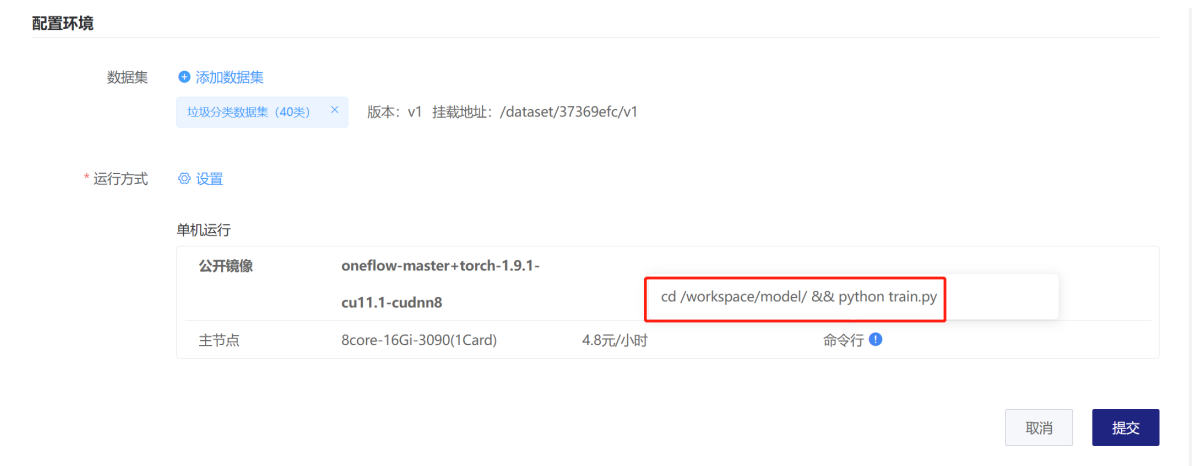
任务启动过程中需要完成资源的申请、拉取镜像和创建实例，请耐心等待。启动成功后，开始训练模型。



5. 自定义脚本任务启动成功后，训练任务状态为 **运行中**。



启动命令行方式调试程序。自定义脚本任务命令行执行完成后，会停止任务。



自定义脚本任务训练结果输出会打印在 `/workspace/logs` 日志里。

```
资源监控  日志  监控指标

training-task-475df0v1-train-master-dkcb4

ws server connected
* Starting OpenBSD Secure Shell server sshd
..done.
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Collecting onefrain
Downloading
https://pypi.tuna.tsinghua.edu.cn/packages/7c/ed/bb53df0f8e1759a3b6d92d4213148c493807dc94c9c3e678e9b7b9c80319/onebrain-0.0.9-py3-none-any.whl (17 kB)
Installing collected packages: onefrain
Successfully installed onefrain-0.0.9
loaded library: /usr/lib/x86_64-linux-gnu/libibverbs.so.1
/usr/local/miniconda3/lib/python3.7/site-packages/oneflow/nn/modules/activation.py:163: UserWarning: ReLU6 module do not support inplace now
warnings.warn("ReLU6 module do not support inplace now")
DEVICE USED: cuda
Epoch 1/3
-----
Training Loss: 4.5737 Acc: 0.2540
Test Loss: 2.0259 Acc: 0.5086
Epoch 2/3
-----
Training Loss: 1.7576 Acc: 0.5540
Test Loss: 0.9962 Acc: 0.7121
Epoch 3/3
-----
Training Loss: 1.2107 Acc: 0.6532
Test Loss: 0.9613 Acc: 0.7224
Training complete in 13 min 55 seconds.
Best acc:0.7224
```

项目

在线开发

WebIDE开发

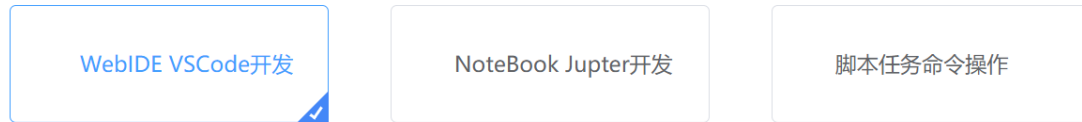
WebIDE是AI实训平台提供的在线VScode集成开发环境。

1. 在 AI 实训平台的“项目”页，点击【创建项目】按钮，进入新建项目页。



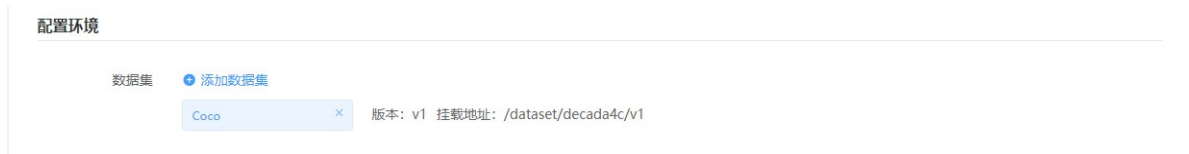
2. 新建项目页需填写信息，选择 webIDE vscode 在线开发

项目类型



1) 数据集

用户可以选择挂载公开数据集版本或自定义的数据集版本，支持最多挂载 5 条数据集。用户可通过在项目中挂载数据源的方式来访问和分析数据。



2) 工作环境

预置公开镜像，支持主流国内外主流的 AI 框架例 TensorFlow、Pytorch，以及我司自主研发的 OneFlow 框架。同时支持用户自定义的镜像。



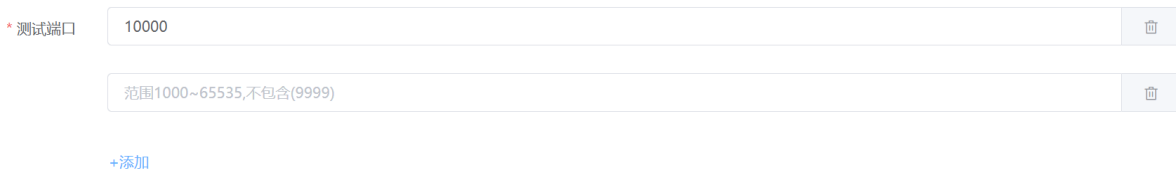
3) 资源配置

运行在 Docker 容器中，所需运行资源，用户根据需要选择。支持资源配置单机多卡训练。如果该选项置灰，表示该资源配置暂不足，不能选择。

4) 测试端口

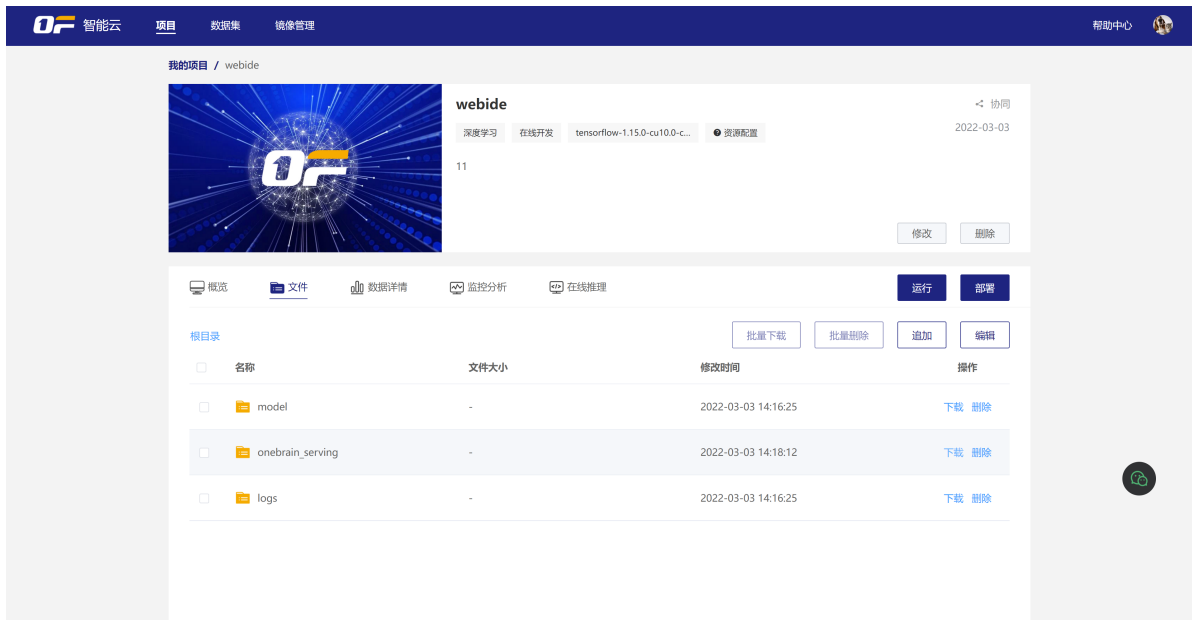
泛域名部署升级后。

项目启动后，通过在浏览器访问 `http(s)://{randomStr}_webide-proxy-{port}.domain` 访问他在容器中占用此端口的服务。



具体说明查看[WebIDE测试端口](#)。

5) 完成配置后，单击【提交】，创建任务。



3. 文件存储目录

- AI 实训平台为每个用户云端持久化存储空间，访问路径为项目下 `/workspace`，存储到该目录下的文件将被持久化存储。用户在重启任务时文件不会初始化重置丢失。

- TensorBoard**可视化文件目录

1) AI 实训平台支持在开发环境中开启**TensorBoard**可视化工具，提供机器学习所需的可视化功能和工具。具体使用参见[TensorFlow 官方网站说明](#)。

2) 建议将可视化数据存放在 `/workspace/logs` 文件系统中，在开发环境中启动 TensorBoard 时，开发环境会自动挂载文件系统目录读取可视化数据。

- 文件管理维护

- 支持在线Markdown编辑,上传/下载/存储。
- 【追加】，单文件追加或选择文件夹方式追加文件。
- 文件在线【编辑】，选择【编辑】后，进入文件编辑模式。

4. 数据集挂载目录

挂载的数据集将会放在 `/dataset` 目录下，例 `/decada4c/v1`，项目下 `/dataset/数据集ID/版本号`下。详细功能详见数据集功能。

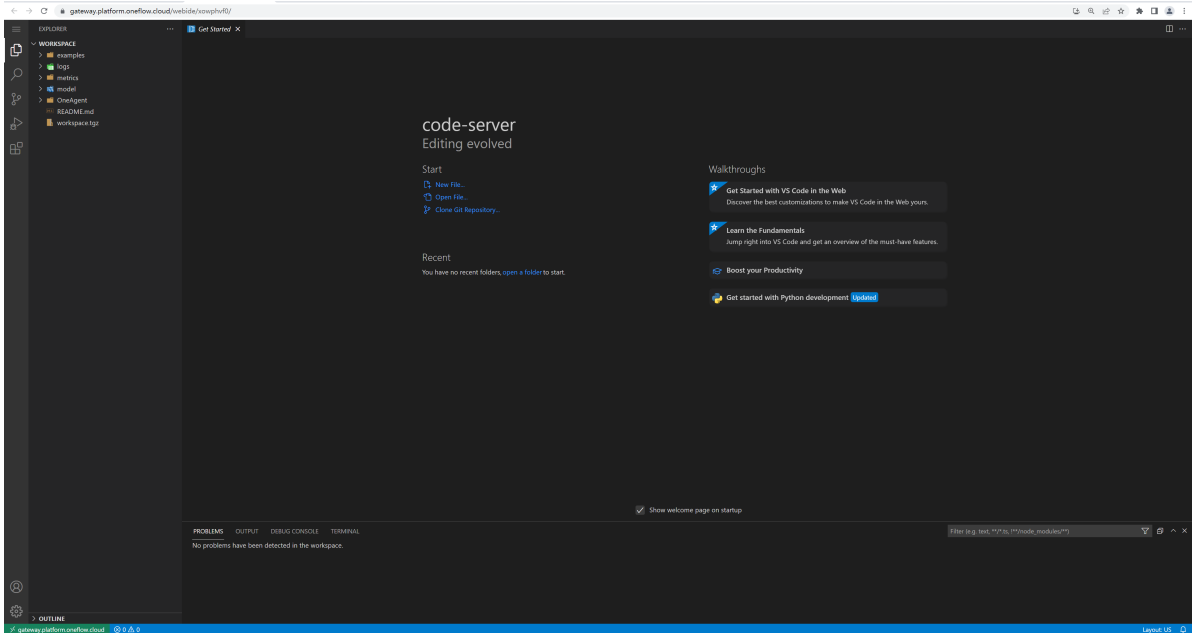
- 用户可以选择挂载公开数据集版本或自定义的数据集版本，支持最多挂载 5 条数据集。
- 用户可通过在项目中挂载数据源的方式来访问和分析数据。



5. 完成配置后，单击【运行】。

启动过程中需要完成资源的申请、拉取镜像和创建实例，请耐心等待。

6. WebIDE启动成功后，点击【进入】，页面新打开 WebIDE VScode在线工具。

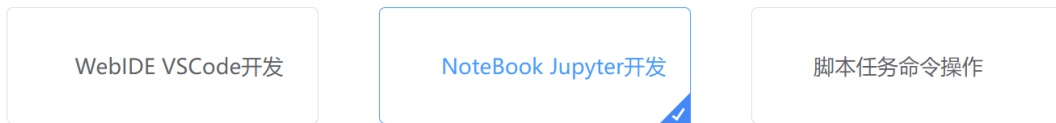


Jupyter开发

Jupyter Notebook 是基于网页的用于交互计算的应用程序。其可被应用于全过程计算：开发、文档编写、运行代码和展示结果。

1. 在 AI 实训平台的“项目”页，点击【创建项目】按钮，进入新建项目页。
2. 新建页需填写信息，如下

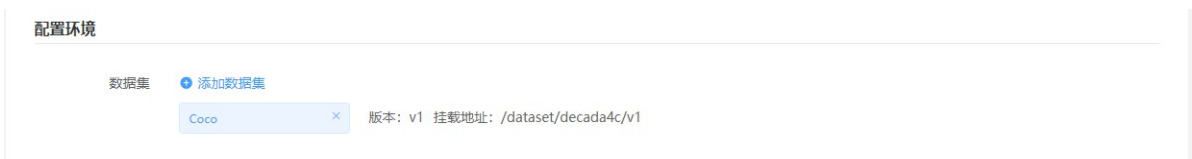
项目类型



1) 数据集

用户可以选择挂载公开数据集版本或自定义的数据集版本，支持最多挂载 5 条数据集。

用户可通过在项目中挂载数据源的方式来访问和分析数据。



2) 工作环境

预置公开镜像，支持主流国内外主流的 AI 框架例 TensorFlow、Pytorch，以及我司自主研发的 OneFlow 框架。同时支持用户自定义的镜像。



3) 资源配置

运行在 Docker 容器中，所需运行资源，用户根据需要选择。支持资源配置单机多卡训练。

如果该选项置灰，表示该资源配置暂不足，不能选择。

4) 完成配置后，单击【提交】，创建任务。

3. 文件存储目录

- AI 实训平台为每个用户云端持久化存储空间，访问路径为项目下 `/workspace`，存储到该目录下的文件将被持久化存储。用户在重启任务时文件不会初始化重置丢失。

- **TensorBoard**可视化文件目录

1) AI 实训平台支持在开发环境中开启**TensorBoard**可视化工具，提供机器学习所需的可视化功能和工具。具体使用参见[TensorFlow 官方网站说明](#)。

2) 建议将可视化数据存放在 `/workspace/logs` 文件系统中，在开发环境中启动 TensorBoard 时，开发环境会自动挂载文件系统目录读取可视化数据。

- 文件管理维护

1) 支持在线Markdown编辑,上传/下载/存储。

2) 【追加】，单文件追加或选择文件夹方式追加文件。

3) 文件在线【编辑】，选择【编辑】后，进入文件编辑模式。

4. 数据集挂载目录

挂载的数据集将会放在 `/dataset` 目录下，例 `/decada4c/v1`，项目下 `/dataset/数据集ID/版本号`。详细功能详见数据集功能。

1) 用户可以选择挂载公开数据集版本或自定义的数据集版本，支持最多挂载 5 条数据集。

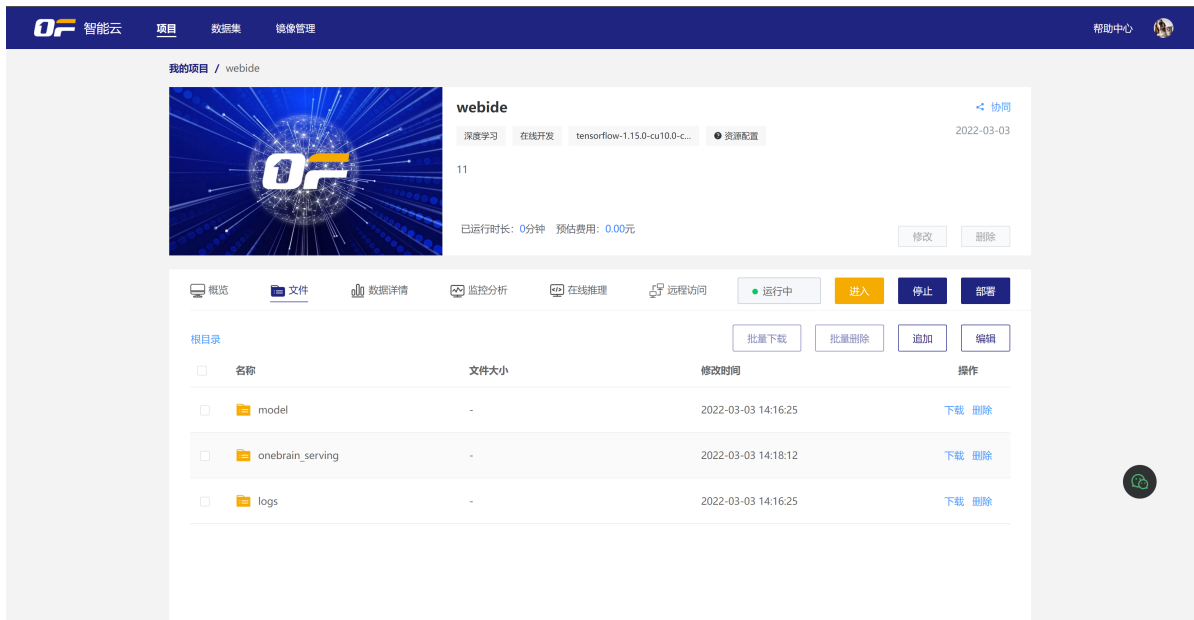
2) 用户可通过在项目中挂载数据源的方式来访问和分析数据。



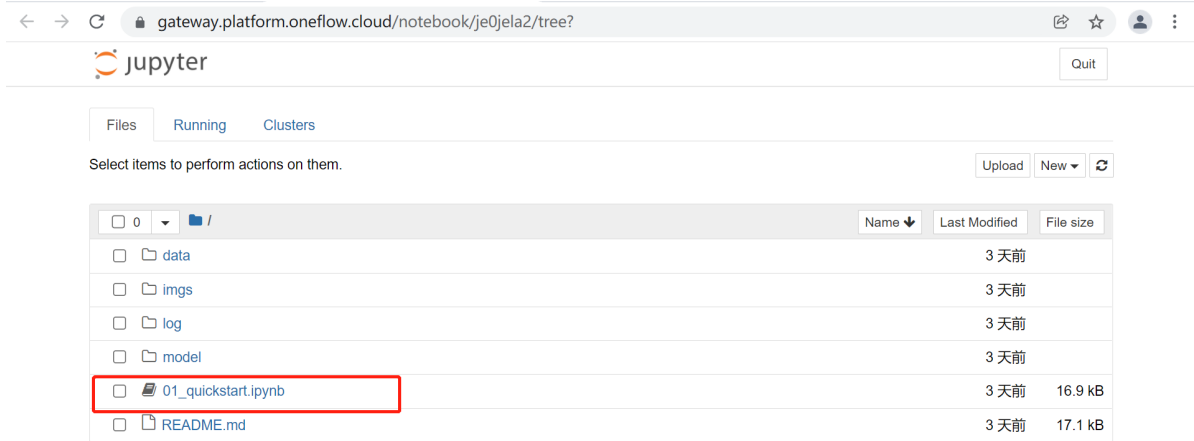
5. 完成配置后，单击【运行】。

启动过程中需要完成资源的申请、拉取镜像和创建实例，请耐心等待。

6. 启动成功后，单击【进入】，页面新打开 Jupyter Notebook 在线交互式编辑工具页。



7. 用户进入 jupyter 程序调试

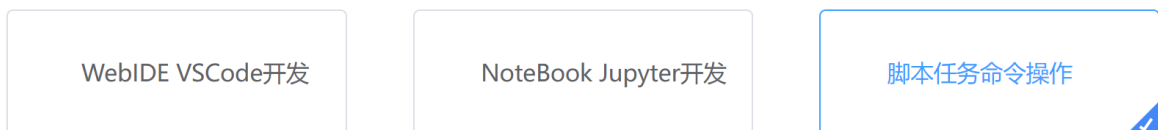


自定义脚本建模

自定义脚本建模提供高性能的计算资源配置，用户可进行大规模分布式的模型训练及脚本调优。自定义脚本任务命令行执行完成后，会停止任务。

1. 首先用户在项目页，点击【创建项目】后，进入创建项目页。
2. 新建页需填写信息，如下

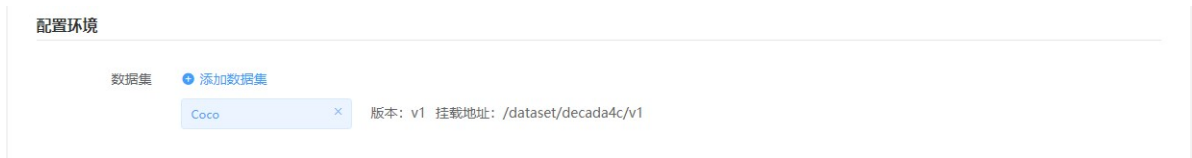
项目类型



1) 数据集

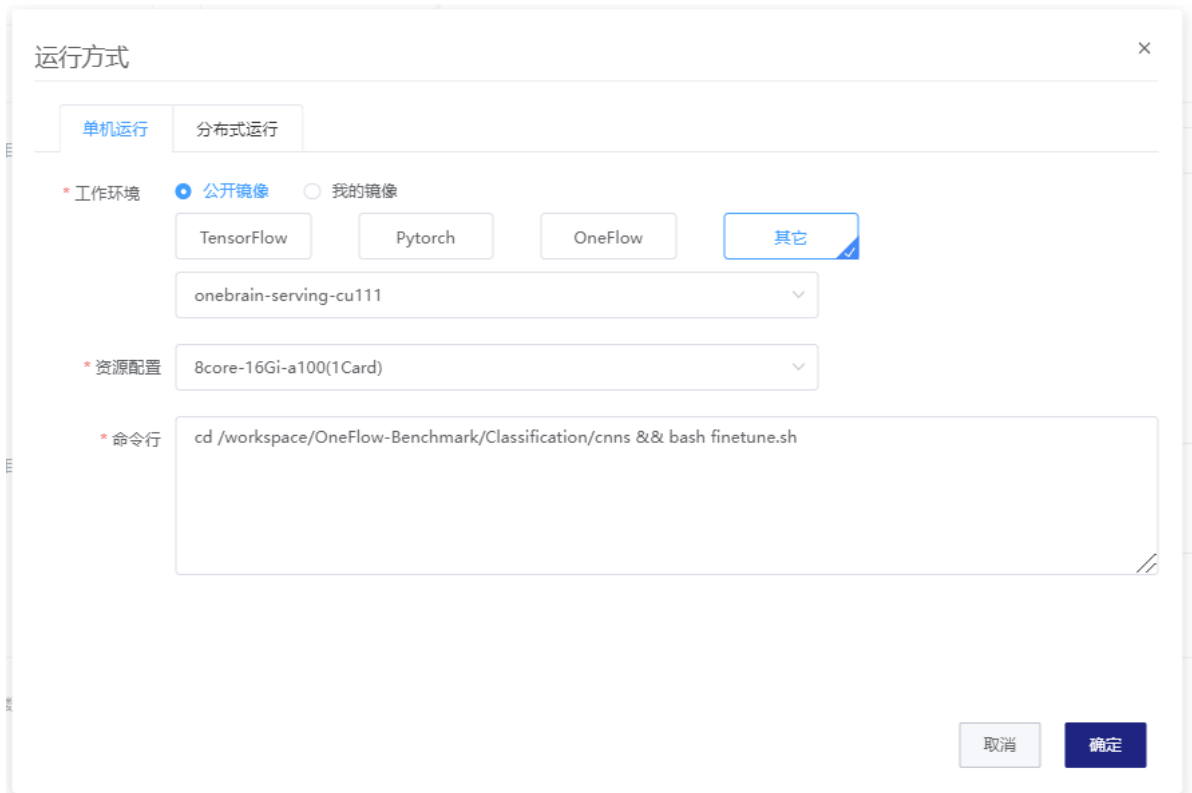
用户可以选择挂载公开数据集版本或自定义的数据集版本，支持最多挂载 5 条数据集。

用户可通过在项目中挂载数据源的方式来访问和分析数据。



2) 运行方式

支持单机运行或分布式运行。



分布式训练适用于大规模模型训练，通过云原生技术，根据用户的需求动态启用多机多卡的资源，多容器之间开启免密认证等，方便用户进行分布式的模型开发。

分布式主-从节点配置，子节点数最大可设置为 3。

3) 工作环境

预置公开镜像，支持主流国内外主流的 AI 框架例 TensorFlow、Pytorch，以及我司自主研发的 OneFlow 框架，也支持强化学习环境。同时支持用户自定义的镜像。

4) 填写命令行

输入脚本执行的命令行，例如 `python /workspace/train.py`，调用 `/workspace` 下的程序文件 `train.py`。如不调用程序，通过 SSH 登录容器 Pod 调试，请写 `sleep 1d`，d 代表 1 天，其他单位 s 秒、m 分钟、h 分钟。

5)完成配置后，单击【提交】，创建任务后进入项目文件页签，可以选择上传脚本代码文件。

3. 文件存储目录

- AI 实训平台为每个用户云端持久化存储空间，访问路径为项目下 `/workspace`，存储到该目录下的文件将被持久化存储。用户在重启任务时文件不会初始化重置丢失。
- **TensorBoard**可视化文件目录

1) AI 实训平台支持在开发环境中开启**TensorBoard**可视化工具，提供机器学习所需的可视化功能和工具。具体使用参见[TensorFlow 官方网站说明](#)。

2) 建议将可视化数据存放在 `/workspace/logs` 文件系统中，在开发环境中启动 TensorBoard 时，开发环境会自动挂载文件系统目录读取可视化数据。

- 文件管理维护

1) 支持在线Markdown编辑,上传/下载/存储。

2) 【追加】，单文件追加或选择文件夹方式追加文件。



3) 文件在线【编辑】，选择【编辑】后，进入文件编辑模式。

4. 数据集挂载目录

挂载的数据集将会放在 /dataset 目录下，例 /decada4c/v1，项目下 /dataset/数据集ID/版本号下。详细功能详见数据集功能。

- 1) 用户可以选择挂载公开数据集版本或自定义的数据集版本，支持最多挂载 5 条数据集。
- 2) 用户可通过在项目中挂载数据源的方式来访问和分析数据。

5. 完成配置后，单击【运行】。



任务启动过程中需要完成资源的申请、拉取镜像和创建实例，请耐心等待。

6. 自定义脚本任务启动成功后，训练任务状态为 **运行中**。

智能云 项目 数据集 镜像管理

我的项目 / 迁移学习: 以垃圾分类为例1637916327380

迁移学习: 以垃圾分类为例1637916327380

2021-11-26

深度学习 脚本任务 oneflow-master+torch-1.9... 资源配置

基于 MobileNet 的迁移学习, 用于40类常见垃圾的垃圾分类, 包含训练与推理脚本。

已运行时长: 0分钟 预估费用: 0.00元 [自SSH登录说明](#) [公开](#)

概览 文件 数据详情 监控分析 在线推理 ssh信息 运行中 停止 部署

迁移学习: 以垃圾分类为例

OneFlow 对 PyTorch 用户很友好

对于单机单卡训练, OneFlow 的体验已经与 PyTorch 非常接近, 熟悉 PyTorch 的用户对 OneFlow 上手一定没有陌生感, 还可以在此基础上体验最易用的分布式训练。

以本项目为例, 以下方式导入包就是 PyTorch 训练。

```
import torch
from torchvision import transforms, datasets
```

启动命令行方式调试程序。自定义脚本任务命令行执行完成后, 会停止任务。

自定义脚本任务训练结果输出会打印在Pod容器日志里。

资源监控 日志 监控指标

training-task-475df0v1-train-master-dkcb4

```
ws server connected
* Starting OpenBSD Secure Shell server sshd
...done.
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Collecting oneflow
Downloading
https://pypi.tuna.tsinghua.edu.cn/packages/7c/ed/bb53df0f8e1759a3b6d92d4213148c493807dc94c9c3e678e9b7b9c80319/oneflow-0.9-py3-none-any.whl (17 kB)
Installing collected packages: oneflow
Successfully installed oneflow-0.9
loaded library: /usr/lib/x86_64-linux-gnu/libibverbs.so.1
/usr/local/miniconda3/lib/python3.7/site-packages/oneflow/nn/modules/activation.py:163: UserWarning: ReLU6 module do not support inplace now
warnings.warn("ReLU6 module do not support inplace now")
DEVICE USED: cuda
Epoch 1/3
-----
Training Loss: 4.5737 Acc: 0.2540
Test Loss: 2.0259 Acc: 0.5086
Epoch 2/3
-----
Training Loss: 1.7576 Acc: 0.5540
Test Loss: 0.9962 Acc: 0.7121
Epoch 3/3
-----
Training Loss: 1.2107 Acc: 0.6532
Test Loss: 0.9613 Acc: 0.7224
Training complete in 13 min 55 seconds.
Best acc:0.7224
```

监控分析

在线开发务运行成功后, 可查看输出数据如下:

资源监控

监控训练任务计算节点资源被使用情况, 指标包含 CPU/GPU/MEM 内存。让用户更细致、全面地了解任务的运行状态, 进而对自己的运行代码进行更准确的调整。



监控日志

自定义脚本任务训练结果输出会打印在Pod容器日志里。

资源监控
日志
监控指标

training-task-475df0v1-train-master-dkcb4

```

ws server connected
* Starting OpenBSD Secure Shell server sshd
...done.
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Collecting onebrain
Downloading
https://pypi.tuna.tsinghua.edu.cn/packages/7c/ed/bb53df0f8e1759a3b6d92d4213148c493807dc94c9c3e678e9b7b9c80319/onebrain-0.0.9-py3-none-any.whl (17 kB)
Installing collected packages: onebrain
Successfully installed onebrain-0.0.9
loaded library: /usr/lib/x86_64-linux-gnu/libibverbs.so.1
/usr/local/miniconda3/lib/python3.7/site-packages/oneflow/nn/modules/activation.py:163: UserWarning: ReLU6 module do not support inplace now
warnings.warn("ReLU6 module do not support inplace now")
DEVICE USED: cuda
Epoch 1/3
-----
Training Loss: 4.5737 Acc: 0.2540
Test Loss: 2.0259 Acc: 0.5086
Epoch 2/3
-----
Training Loss: 1.7576 Acc: 0.5540
Test Loss: 0.9962 Acc: 0.7121
Epoch 3/3
-----
Training Loss: 1.2107 Acc: 0.6532
Test Loss: 0.9613 Acc: 0.7224
Training complete in 13 min 55 seconds.
Best acc:0.7224

```

帮助用户通过查看日志来分析运行过程中的异常信息。

概览
文件
数据详情
监控分析
在线推理
ssh信息
运行中
进入
停止
部署

资源监控
日志
监控指标
Tensorboard

training-notebook-267eaf-267eaf-jupyter-master-0

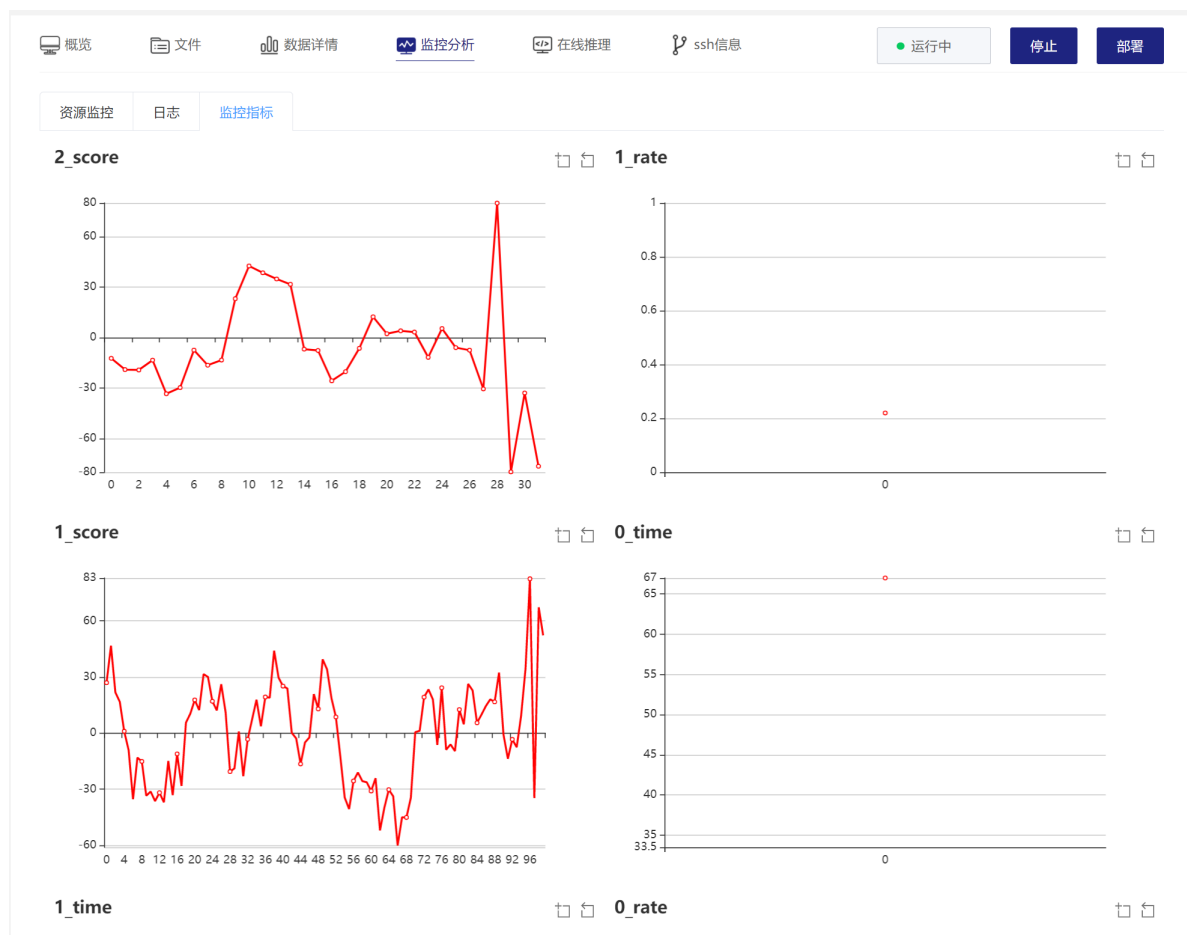
```

ws server connected
mesg: ttyname failed: Inappropriate ioctl for device
* Starting OpenBSD Secure Shell server sshd
...done.
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ConnectTimeoutError(<Connection to pypi.tuna.tsinghua.edu.cn timed out. (connect timeout=15)>): /simple/onebrain/
Collecting onebrain
Downloading
https://pypi.tuna.tsinghua.edu.cn/packages/7c/ed/bb53df0f8e1759a3b6d92d4213148c493807dc94c9c3e678e9b7b9c80319/one-0.0.9-py3-none-any.whl (17 kB)
Installing collected packages: onebrain
Successfully installed onebrain-0.0.9
WARNING: You are using pip version 20.0.2; however, version 21.3.1 is available.
You should consider upgrading via the '/usr/bin/python3 -m pip install --upgrade pip' command.
[I 08:56:57.859 NotebookApp] Writing notebook server cookie secret to
/root/.local/share/jupyter/runtime/notebook_cookie_secret
[W 08:56:58.170 NotebookApp] All authentication is disabled. Anyone who can connect to this server will be able to run
code.
[I 08:56:58.175 NotebookApp] Serving notebooks from local directory: /workspace
[I 08:56:58.175 NotebookApp] The Jupyter Notebook is running at:
[I 08:56:58.175 NotebookApp] http://training-notebook-267eaf-267eaf-jupyter-master-0:8888/notebook/ceg4m5y5/
[I 08:56:58.175 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

```

监控指标

平台提供 API 接口支持用户自定义监控指标数据输出。可以定义监控训练和部署中的各种指标包括 Loss, 精确度等。[具体查看 SDK 参考](#)。



Tensorboard

TensorBoard 是一个可视化工具，能够有效地展示 TensorFlow 在运行过程中的计算图、各种指标随着时间的变化趋势以及训练中使用到的数据信息。TensorBoard 当前只支持基于 TensorFlow 和 MXNet 引擎的训练作业。TensorBoard 相关概念请参考 TensorBoard 官网。

- TensorBoard 可视化文件目录

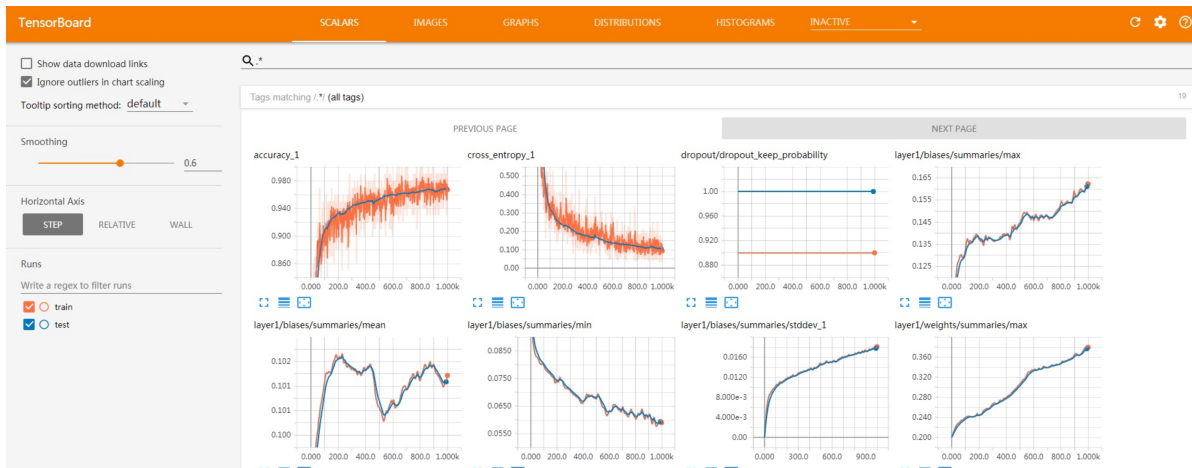
1) AI 实训平台支持在开发环境中开启**TensorBoard**可视化工具，提供机器学习所需的可视化功能和工具。具体使用参见[TensorFlow 官方网站说明](#)。

2) 训练模型时建议将可视化数据存储于 /workspace/logs 文件系统中，在开发环境中启动 TensorBoard 时，开发环境会自动挂载文件系统目录读取可视化数据。

- 平台提供 API 接口支持用户训练中指标的可视化。



打开【Tensorboard】新打开 TensorBoard 是一个可视化界面。



SSH 登录调试

训练任务启动成功后，支持 SSH 登录远程开发，多容器之间开启免密认证等，方便用户进行分布式的模型开发。支持本地 IDE 远程访问训练任务，通过开启 SSH 连接，用户本地 IDE 可以远程连接到平台的训练任务环境中，可 pip 安装软件或调试。

支持 Linux、Windows、Mac 操作平台，具体使用可查看[SSH 登录说明](#)。

支持 CLI 开发工具，不限制操作系统，具体可查看[CLI 开发工具说明](#)。

SSH Key

创建生成 SSH Key

使用 SSH 公钥可以让你在你的电脑和 OF 智能云通讯的时候使用安全连接，用户使用 SSH 登录容器调试，不需要输入密码。

windows 生成公钥方法

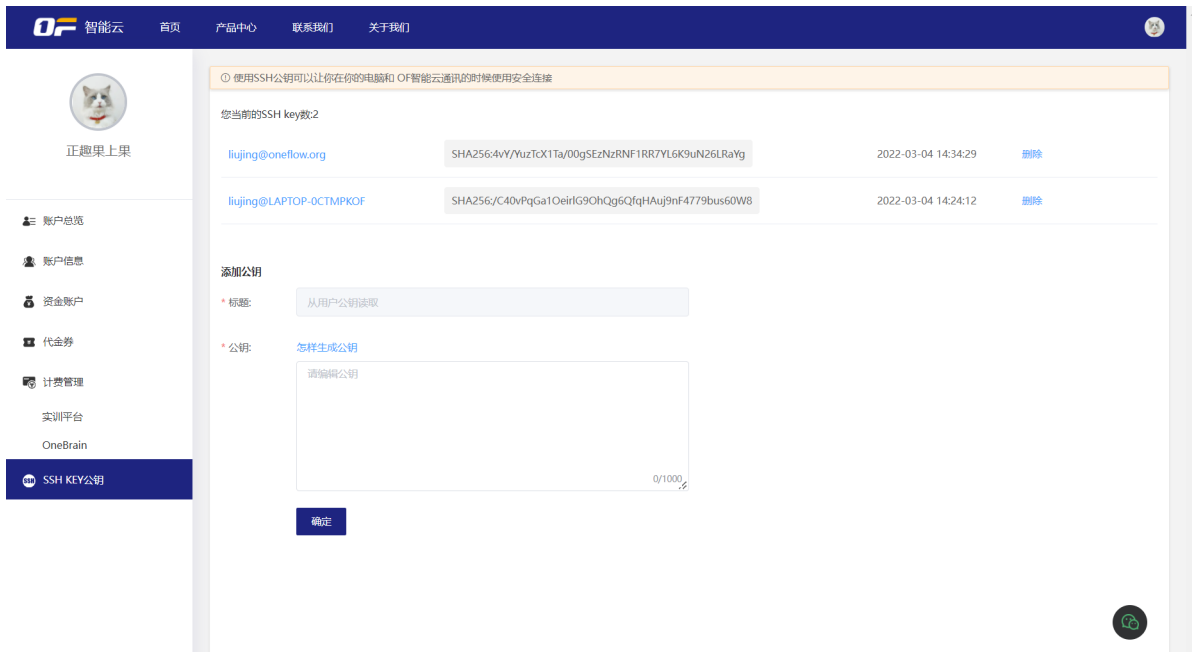
1. win + R 打开运行窗口，输入并运行 cmd.exe
2. 在终端窗口中输入 ssh-keygen.exe，根据提示生成 id_rsa(私钥)和 id_rsa.pub(公钥)。
3. 公钥和私钥的默认保存位置在 <系统盘>:\Users<username>\.ssh 文件夹下

linux/macOS 生成公钥方法

1. 打开终端，如未安装 openssh，请提前安装
 - centos: yum install openssh-server openssh-client -y
 - ubuntu: apt install openssh-server openssh-client -y
2. 在终端窗口中输入 ssh-keygen.exe，根据提示生成 id_rsa(私钥)和 id_rsa.pub(公钥)。
3. 公钥和私钥默认保存位置在 ~/.ssh 路径下

生成安全链接

1. 用户在 OF 智能云的用户中心下，选择【SSH KEY】进入公钥管理页。



2. 添加公钥

用户复制自己系统的公钥信息并确认后，OF智能云验证用户登录密码通过后，生产安全链接。用户在使用ssh 登录容器远程调试时，无需再次输入密码。

WebIDE协同开发

Webide项目支持项目运行成功后，复制地址链接分享授权给注册AI实训平台用户手机号，协同开发调试。



- 1) 授权有效期为WebIDE启动成功后有权访问开发链接，任务停止后，移除授权不能访问。
- 2) 未授权用户不能访问开发链接，报错“无权访问链接地址”。
- 3) 每次启动项目后，项目地址都会更新。

WebIDE测试端口

泛域名部署升级后。

项目启动后，通过在浏览器访问 `http(s)://{randomStr}_webide-proxy-{port}.domain/hello` 访问他在容器中占用此端口的服务。

* 测试端口

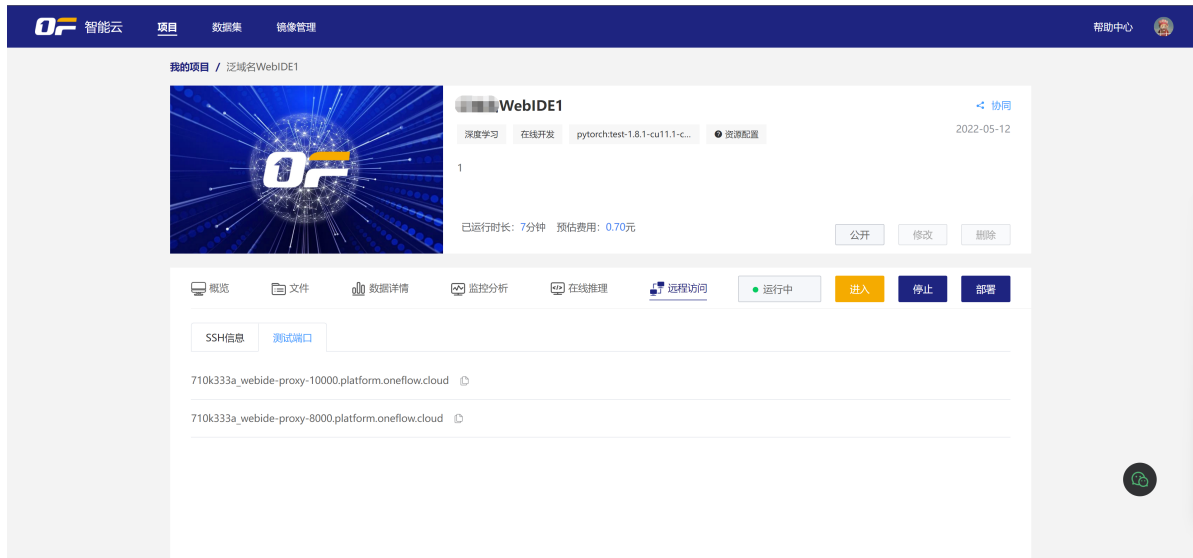
🗑

范围1000~65535,不包含(9999)

+添加

用户在创建webide时，填写了两个测试端口，为10000和10001。

用户启动webide后，在webide中启动了两个flask的Web服务，分别占用10000和10001端口。



代码范例

```
from flask import Flask

app = Flask(__name__)

import os
port0 = int(os.environ.get('TESTING_PORT_0'))

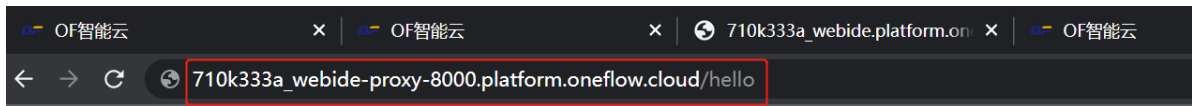
@app.route('/hello')
def hello_world():
    return 'Hello, world!'

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=port0)
```

访问10000的服务，浏览器路径 `https://710k333a_webide-proxy-10000.platform.oneflow.cloud/hello`。

访问10001的服务，浏览器路径 `https://710k333a_webide-proxy-10001.platform.oneflow.cloud/hello`。

访问后输出如图



Hello, World!

文件存储目录

持久化存储空间

AI 实训平台为每个用户云端持久化存储空间，访问路径为项目下 `/workspace`，存储到该目录下的文件将被持久化存储。用户在重启任务时文件不会初始化重置丢失。

TensorBoard 可视化文件目录

- 1) AI 实训平台支持在开发环境中开启TensorBoard可视化工具，提供机器学习所需的可视化功能和工具。具体使用参见[TensorFlow 官方网站说明](#)。
- 2) 建议将可视化数据存放在 `/workspace/logs` 文件系统中，在开发环境中启动 TensorBoard 时，开发环境会自动挂载文件系统目录读取可视化数据。

文件管理维护

- 1) 支持在线Markdown编辑,上传/下载/存储。
- 2) 【追加】，单文件追加或选择文件夹方式追加文件。

[大文件上传建议使用CLI开发工具](#)

- 3) 文件在线【编辑】，选择【编辑】后，进入文件编辑模式。

数据集挂载目录

挂载的数据集将会放在 `/dataset` 目录下，例 `/decada4c/v1`，项目下 `/dataset/数据集ID/版本号`下。详细功能详见数据集功能。

- 1) 用户可以选择挂载公开数据集版本或自定义的数据集版本，支持最多挂载 5 条数据集。
- 2) 用户可通过在项目中挂载数据源的方式来访问和分析数据。



分布式训练

说明

- 当在平台上启动分布式任务时，平台会按照用户的配置参数在集群中启动多组容器，每组容器都对应一个角色。
- `master` 作为必需的组而存在，副本数为 1。`slave` 按照配置可存在 0 到多个，每个 `slave` 组中又可以有多个副本。
- 分布式任务会按照组的配置顺序依次启动，前一组启动成功后，再启动下一组，直至全部完成。如果启动过程中出现问题，会在页面上提示错误信息。
- 每一组都可以指定使用不同的对象，被启动的容器会使用用户指定的镜像。
- 每一组都可以执行不同的命令，被启动的容器会运行用户输入的命令。当命令执行完成时，容器会被终止。
- 当 `master` 组的容器终止时，平台会认为分布式任务结束。根据命令执行的返回结果，判断任务最终成功还是失败。
- 分布式任务所拉起的每个节点都会得到一个唯一的内网 `ip`。当容器重启时，该 `ip` 不会发生改变。
- 分布式任务所拉起的每个节点都会得到一组默认环境变量（详见默认环境变量说明）。
- 分布式任务所拉起的每个节点在内网中互通，彼此之间都可以通过 `ssh` 免密登录。
- 分布式任务所拉起的每个节点都拥有一份全节点的描述文件，该文件放在 `/home/hostfile.json`。

默认环境变量说明

分布式训练会按照一主多组从的形式启动资源，每个被拉起的容器内都包含以下环境变量：

- `NODE_NUM`：分布式节点数量。
- `OWN_IP_ADDRESS`：节点本身的 `ip` 地址。
- `RUN_ID`：平台产生的 `resourceId`，由平台服务端在启动时指定。
- `ONEBRAIN_TRACKING_NODE`：pod 本身被调度到的节点名称。
- 分布式节点的 `ip` 与名称：环境变量名为大写的组+下标，环境变量值是 `ip`。比如 1 主 2 组从，每组从有 2 个副本，那么依次就是 `MASTER`，`SLAVE_0_0`，`SLAVE_0_1`，`SLAVE_1_0`，`SLAVE_1_1`，因为主只会会有一个所以必然第一个是 `MASTER`，`SLAVE_{_}_{_}`，第一个是组下标，第二个是副本下标，都从 0 开始。

示例

例如在平台上创建一个分布式任务。配置为一组 `master`，一组 `slave`，`slave` 副本数为 2 的 3 节点任务。

* 运行方式 [设置](#)

分布式运行

公开镜像	torch-1.6.0-cu10.1-cudnn7		
主节点	8core-16Gi-3090(1Card)	4.8元/小时	命令行 !
子节点	8core-16Gi-3090(1Card)	4.8元/小时	命令行 ! 数量 2

当任务创建完成，将它启动起来，等待任务状态变为 `运行中`。

点击 `ssh` 信息页，复制 `ssh` 连接命令和密码。通过终端登录到 `master` 节点。

操作平台 Linux

ssh地址 `ssh -o ProxyCommand="ncat --proxy-type http --proxy sshproxy.platform.oneflow.cloud:32321 %h %p" root@10.7.221.40` [!](#)

ssh密码 ***** [!](#)

查看全节点描述文件。

```
(base) root@training-task-69f7b9v1-train-slave-0-1:~# cat /home/hostfile.json |python -m json.tool
[
  {
    "role": "master",
    "ip": "10.7.221.40",
    "index": 0,
    "group": "MASTER"
  },
  {
    "role": "slave",
    "ip": "10.7.74.38",
    "index": 0,
    "group": "SLAVE_0"
  },
  {
    "role": "slave",
    "ip": "10.7.173.237",
    "index": 1,
    "group": "SLAVE_0"
  }
]
```

其中

- `role` 是节点的角色, `master` 或者 `slave`。
- `ip` 是节点的内网 ip 地址。
- `index` 是节点在组内的下标。
- `group` 表示节点所在组的名称

查看节点内的环境变量。

```
$ env
```

当前节点中存在以下环境变量

```
HOSTNAME=training-task-69f7b9v1-train-master-ch2sk
ONEBRAIN_TRACKING_NODE=onebrain-h1455-rtx3090-02
NODE_NUM=3
RUN_ID=31lead808c0659250dbfa319c3432e973
MASTER=10.7.221.40
SLAVE_0_1=10.7.173.237
SLAVE_0_0=10.7.74.38
OWN_IP_ADDRESS=10.7.221.40
.....
```

通过 ssh 免密登录到另外两个节点任意之一。

```
(base) root@training-task-69f7b9v1-train-master-ch2sk:~# ssh ${SLAVE_0_1}
Warning: Permanently added '10.7.173.237' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

(base) root@training-task-69f7b9v1-train-slave-0-1:~#
```

查看 `slave` 节点中的环境变量。

```
$ env
```

当前节点中存在以下环境变量

```
HOSTNAME=training-task-69f7b9v1-train-slave-0-1
ONEBRAIN_TRACKING_NODE=onebrain-h1455-rtx3090-02
NODE_NUM=3
RUN_ID=31lead808c0659250dbfa319c3432e973
MASTER=10.7.221.40
SLAVE_0_1=10.7.173.237
SLAVE_0_0=10.7.74.38
OWN_IP_ADDRESS=10.7.173.237
.....
```

`slave` 节点也存在 `hostfile.json` 文件。

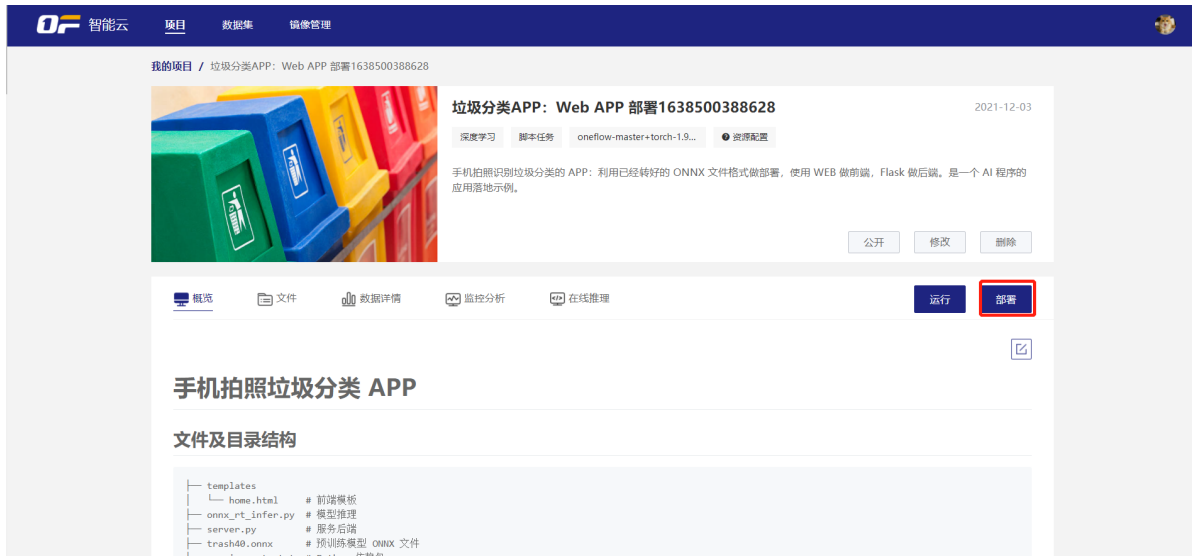
```
(base) root@training-task-69f7b9v1-train-master-ch2sk:~# cat /home/hostfile.json |python -m json.tool
[
  {
    "role": "master",
    "ip": "10.7.221.40",
    "index": 0,
    "group": "MASTER"
  },
  {
    "role": "slave",
    "ip": "10.7.74.38",
    "index": 0,
    "group": "SLAVE_0"
  },
  {
    "role": "slave",
    "ip": "10.7.173.237",
    "index": 1,
    "group": "SLAVE_0"
  }
]
```

用户可以解析 `hostfile.json`，比较 `OWN_IP_ADDRESS` 与 `ip` 的值判断当前节点的角色和所属组。

若 `slave` 节点因故障而重启，重启后的 `ip` 不会发生改变。

部署服务

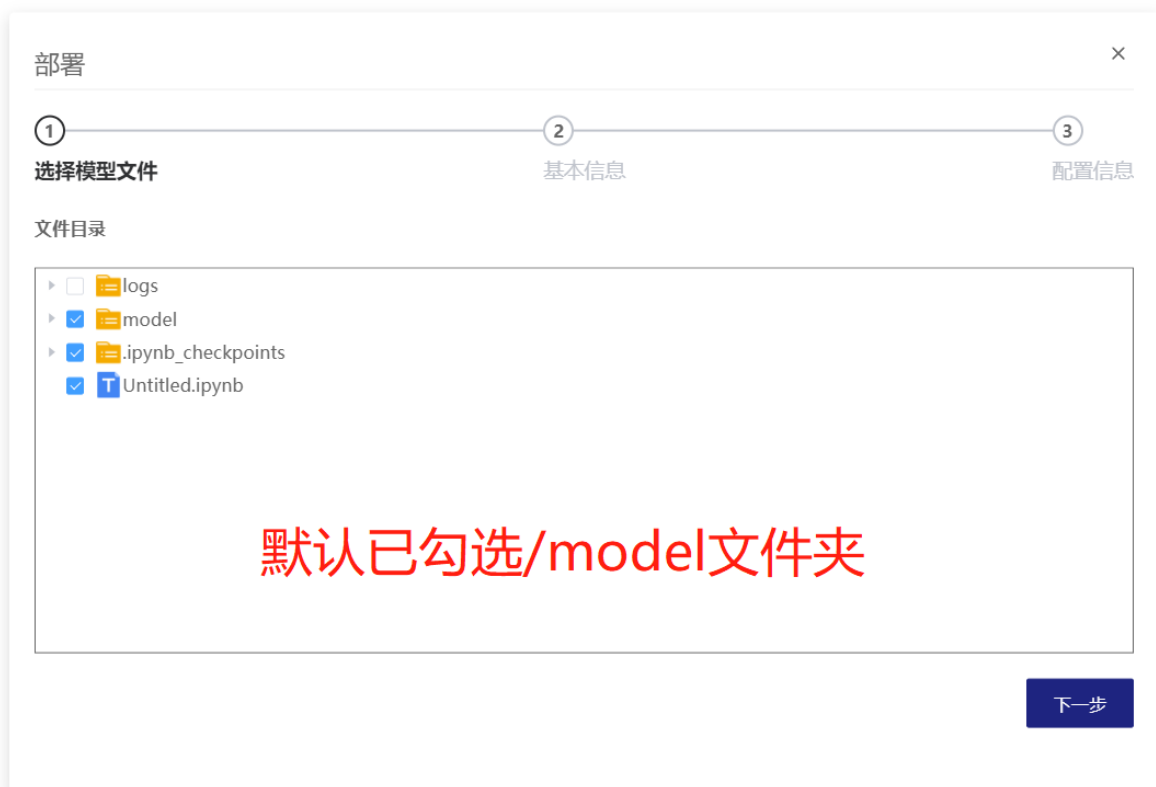
训练完成的模型存储在公有云上，可以在部署服务里建立实例，用户通过调用 API 的方式实现在云端推理预测，以验证模型。



操作说明

1. 通过在线开发建模或自定义脚本建模训练出模型。
2. 创建部署服务。在我的项目详情页，点击【部署】后，弹框设置部署信息。

1) 选择模型，默认勾选 model 文件夹目录。



2) 基本信息设置，部署名称、分类、版本和描述信息。说明设置名称加版本号需唯一。

部署

1 选择模型文件 2 基本信息 3 配置信息

* 名称

* 分类

* 版本号

* 描述

3) 配置信息设置，部署需要工作环境、部署类型、启动命令行、端口信息。

部署

1 选择模型文件 2 基本信息 3 配置信息

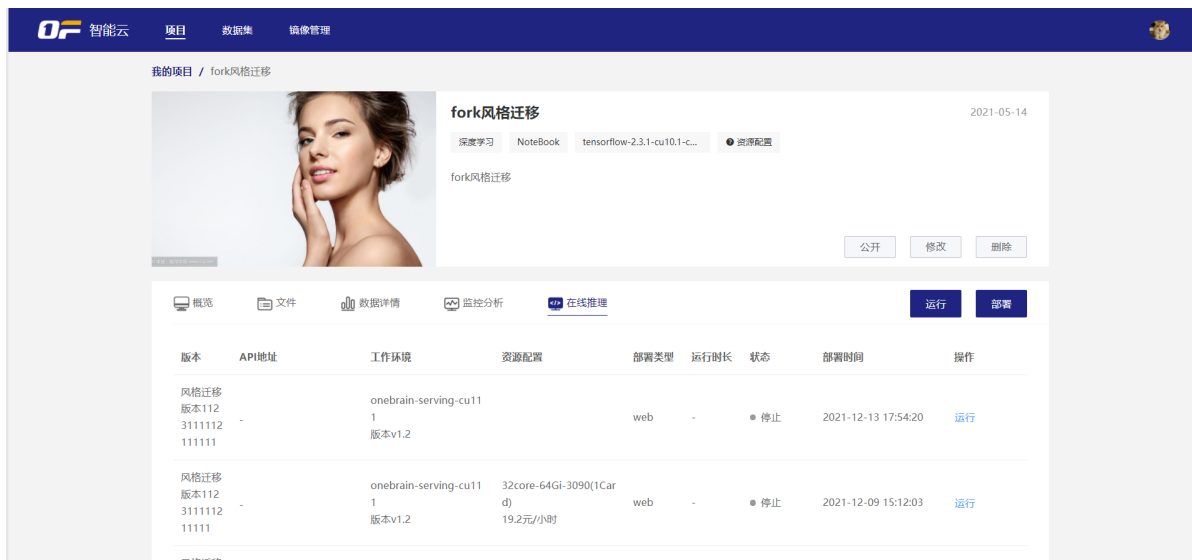
* 工作环境 公开环境 我的环境

* 部署类型 Web类型 数据接口类型

启动命令行

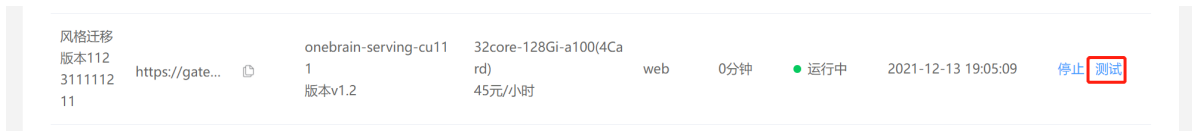
端口

部署确认信息后，系统生成部署实例。

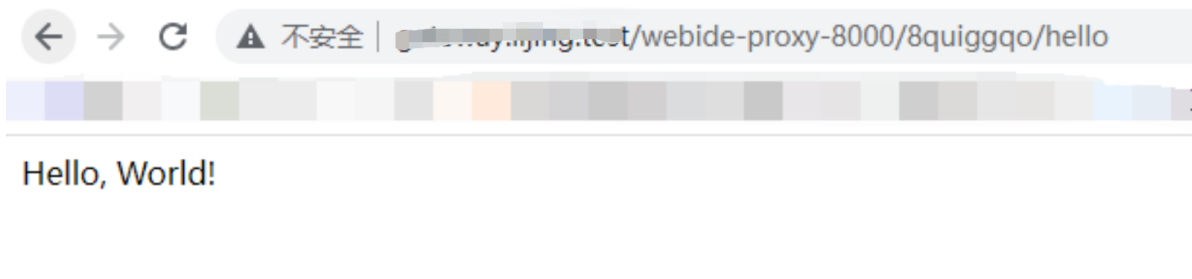


3. 启动运行部署服务。实例需要进行资源配置，运行过程中将按照您选择的资源按需计费。

启动过程中需要一定时间请耐心等待。运行成功后，点击【测试】后，进入公有云 web 推理 UI 界面，可以推理预测模型效果。



测试UI如图



数据集

AI 实训平台提供丰富且高质量的公开数据集，用户也可以自定义数据集。用户可通过在项目中挂载数据源的方式来访问和分析数据。

数据的接入与数据集版本管理功能，让用户可以简单快速的进行数据分析工作。

- 支持统一数据集管理，通过面向对象存储实现非结构化数据（如语音、图片、视频等）的存储和使用，实现数据在平台各服务之间的共享。
- 支持数据分类和数据目录结构建立，数据内容包含待标注数据、已标注数据、预处理后数据、算法数据、模型数据等分类；
- 支持原始数据与标注数据的版本管理，轻松的追溯数据的版本迭代过程。



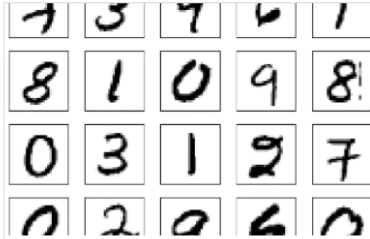
summer2winter_yosemite

本数据集可用于非配对的图像到图像翻译任务，其包含两个域的数据，分别是夏天和冬天的图片。

其它

dangkai

0



MNIST

手写字符图像分类数据集

图像分类

dangkai

0



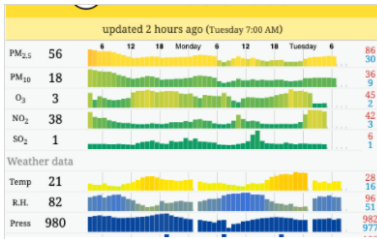
midi

数据集下载自<https://github.com/olofmogren/c-rnn-gan>，包含3451个midi文件。

语音识别

Xuan

1



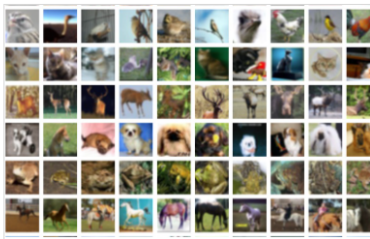
Krakow Air quality dataset

Air quality data of PM1, PM2.5, PM10, temp, pres and hum data for 2017 year from Krakow, Pol...

语音识别

Xuan

1



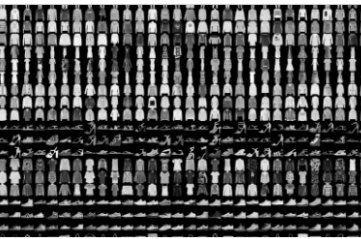
CIFAR-10

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes.

图像分类

Xuan

1



FashionMNIST

Zalando's article images—a training set of 60,000 examples and a test set of 10,000 examples.

图像分类

Xuan

0

数据集版本管理

版本管理是 AI 实训平台的主要功能，它可以帮助团队、个人进行数据开发的版本管理，解决难以比较数据修改内容、合并数据、取消修改、对一份数据进行并行开发、并且难以追溯数据信息等痛点问题。

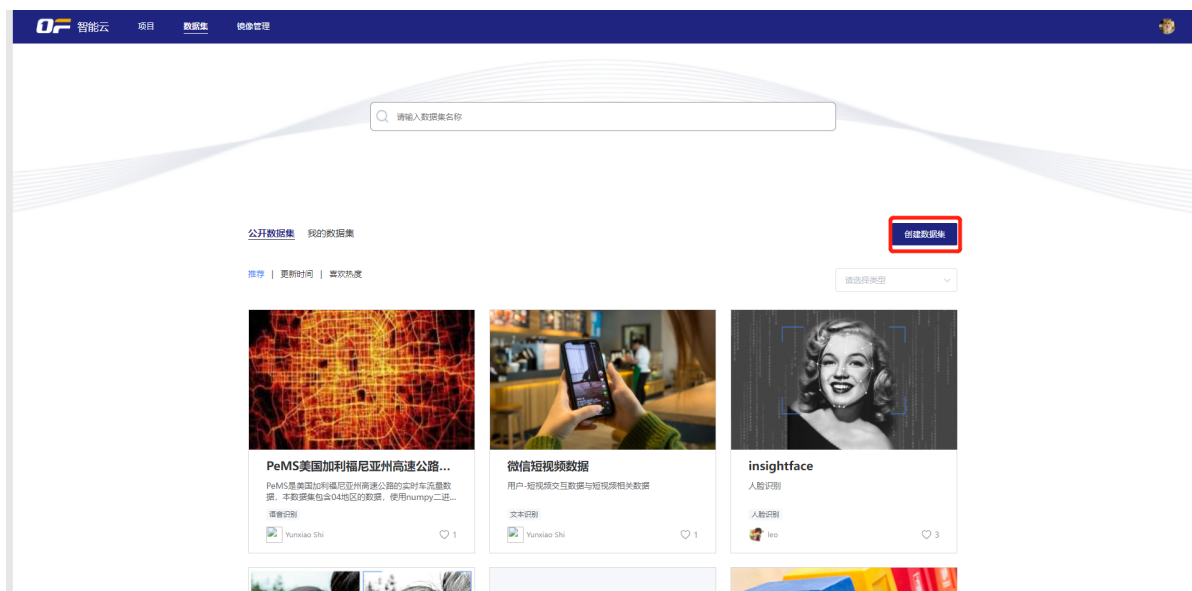
Dataset 数据集版本管理工具

- Commit 记录了 Dataset 在某一时间点的快照，Commit 可以对其所有迭代内容进行管理和追溯。
- 分支管理，记录版本信息。
- 记录所有版本记录，但是不增加存储开销。
- 草稿页编辑数据集的工作区。可以对数据文件进行添加、修改、删除等操作。编辑完成后可以选择发布版本，一经发布成功后的版本不可在编辑操作。用户也可以在版本记录里选择签出为草稿后，在当前版本下进行数据文件维护。

操作说明

1. 创建数据集组

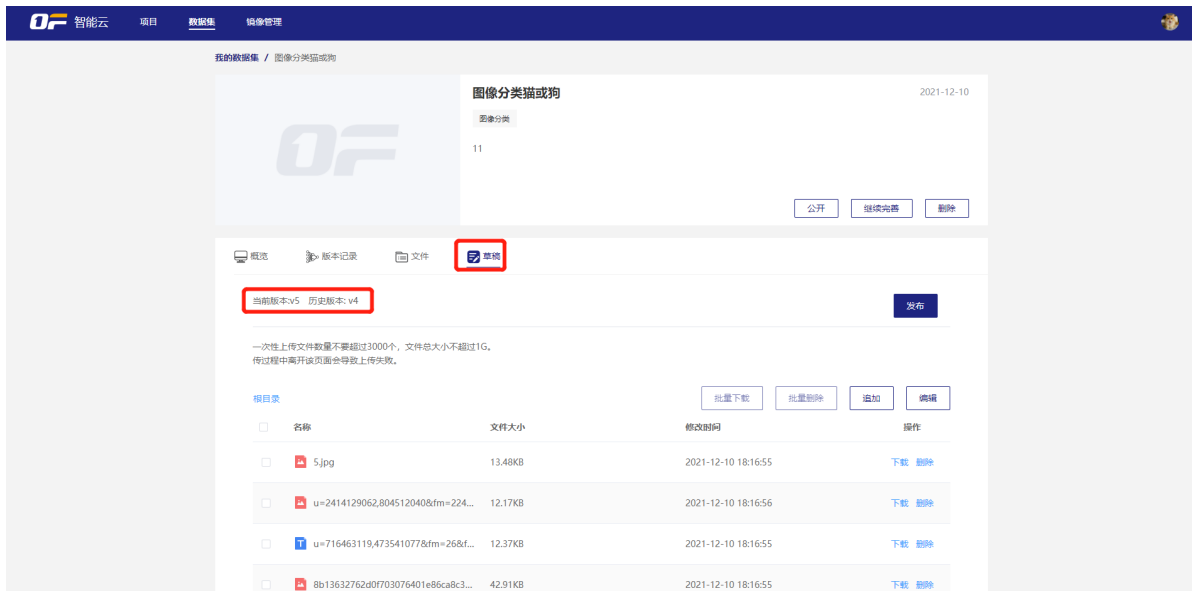
1) 支持本地数据文件上传，在 AI 实训平台的“数据集”页，点击【创建数据集】，弹框快速设置数据集信息。



2) 设置数据集基本信息如下，点击【确定】，创建数据集组，进入数据集组草稿页。

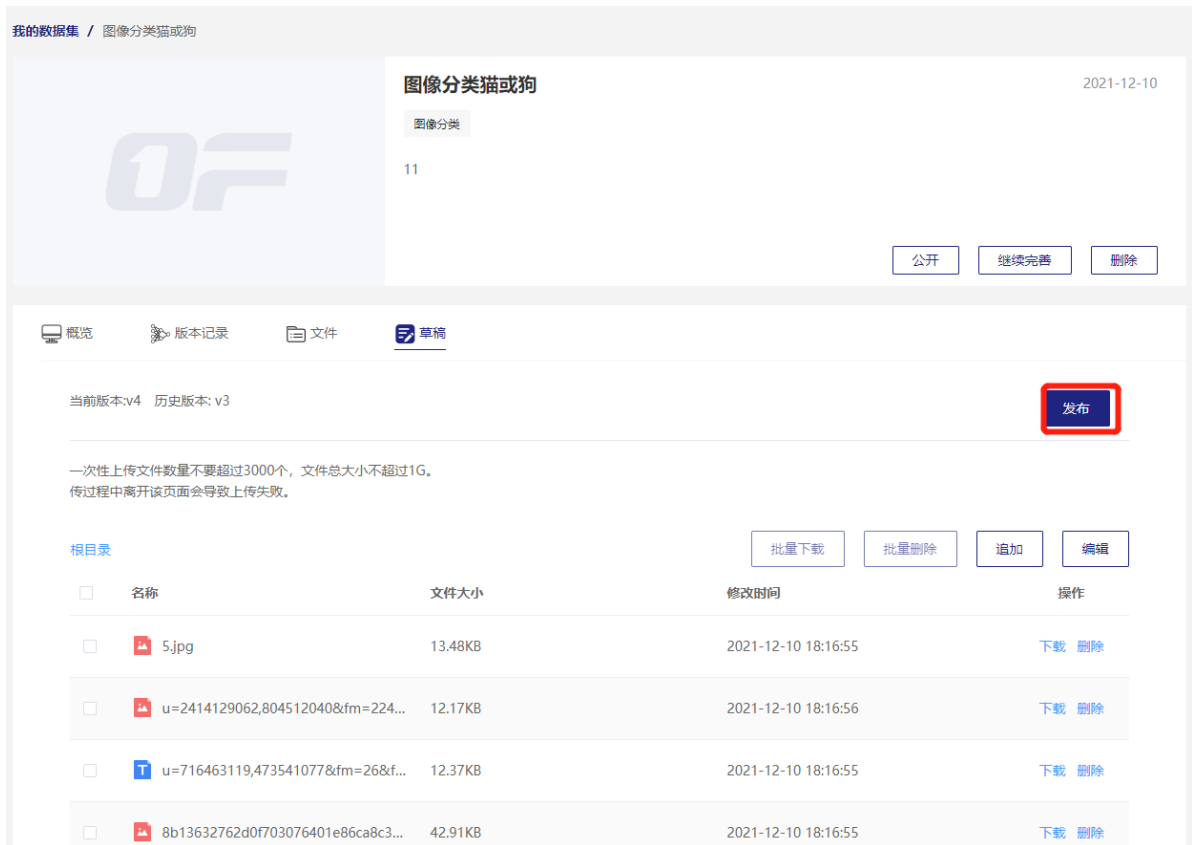
2. 草稿页编辑数据集的工作区

可以对数据文件进行添加、修改、删除等操作。记录历史版本信息，当前版本（草稿发版后的记录）信息。



3. 发布数据集版本记录。

1) 数据编辑完成后可以选择【发布】版本，一经发布成功后的版本不可在编辑操作。草稿页记录最新发布成功版本记录。



弹框设置新版本信息，确认发布。发布成功后，进入版本记录页。

发布新版本 ×

版本号:

历史版本: v3

描述:

确认

2) 用户也可以在版本记录里选择【签出为草稿】后，在当前版本下进行数据文件维护。

版本	版本ID	历史版本	文件数量	文件大小	创建时间	操作
v3	d0995174661a9c3dc2fd1fe09505c4db	v2	21	2.40MB	2021-12-10	下载 删除
v2	781148195c9382db0eda4965d0dbefd4	v1	2	1.39MB	2021-12-10	下载 删除 签出为草稿
v1	780a79a3d92c4a2e1e91457fa69f0036	-	2	1.39MB	2021-12-10	下载 删除 签出为草稿

用户点击【签出为草稿】后，弹出二次确认框如下。用户确认后签出当前数据集版本文件记录进入草稿页。

签出为草稿本 ×

当前草稿有正在编辑文件，如果签出当前版本记录，将删除正在编辑文件记录

继续编辑
确认

如何挂载数据集，请查看 [在线开发建模或自定义脚本建模说明](#)。

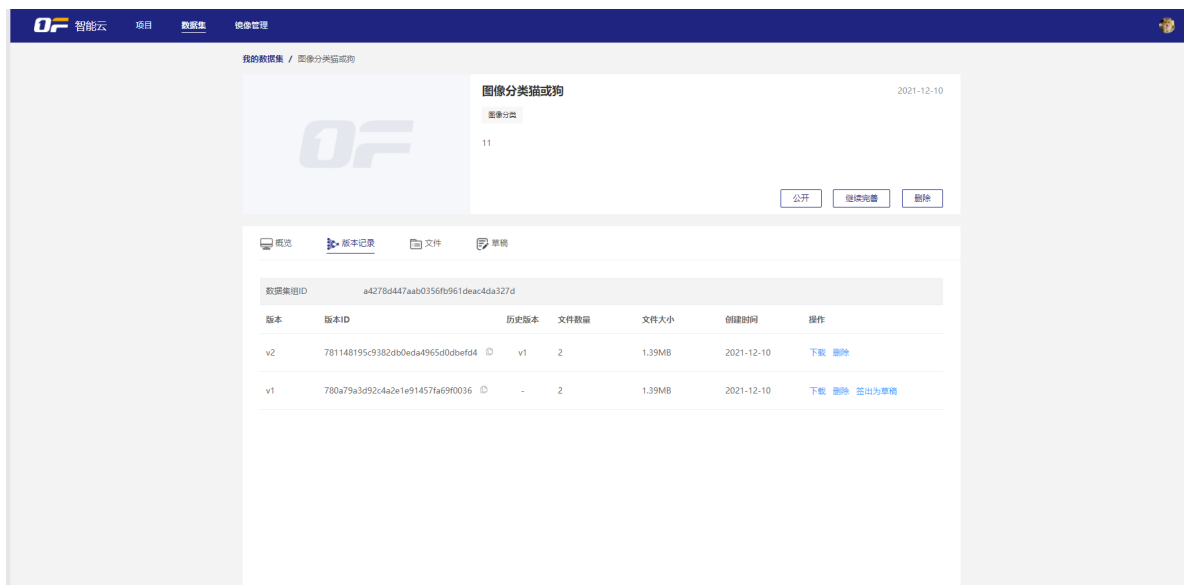
4. 其他

- [数据集组概览页](#)

介绍数据集说明、数据集信息、数据集最新版本说明、版权方归属方和官网等说明。

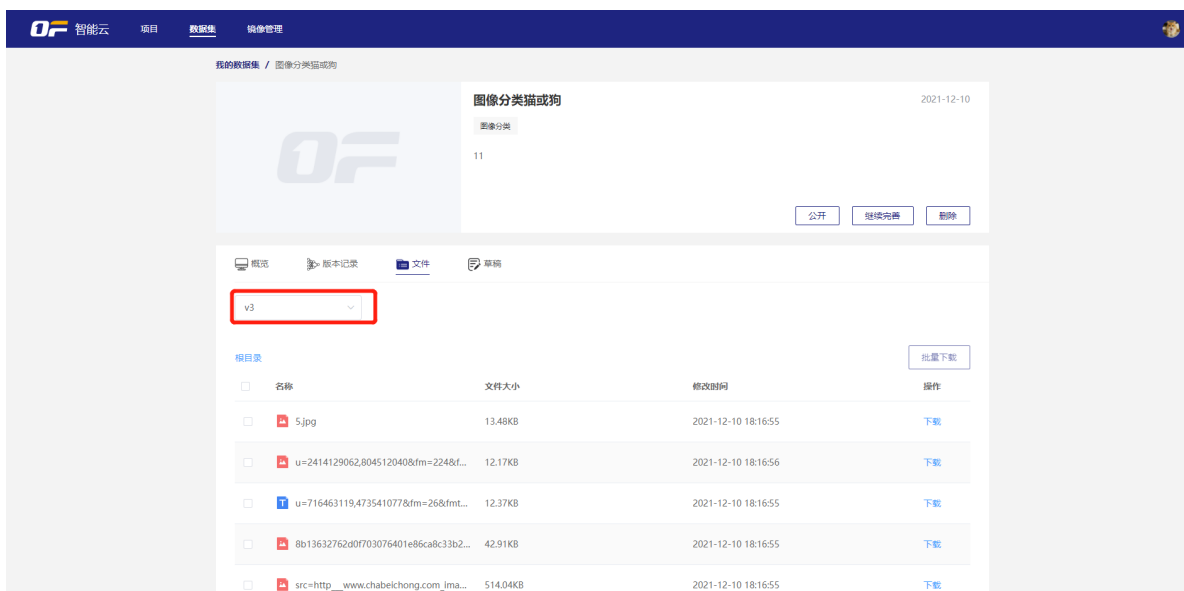
- [版本记录页](#)

记录数据集版本信息，数据版本 id、历史版本、当前版本文件数量等信息。用户可以下载数据集版本，也可以删除数据集版本，删除数据集版本不影响已经挂载的训练任务。



- 文件页签

用户可以查看数据集版本文件信息。



我的镜像

AI 实训平台支持用户自定义镜像，以满足用户训练任务或部署推理使用环境。

步骤

- 构建镜像只能使用平台端基础镜像作为FROM。
- 构建基础镜像后，新建我的项目，用于真实镜像构建过程的测试。
- 回到我的镜像构建基础镜像，选择【复制】后，编辑新dockerfile，并重新构建镜像。

构建基础镜像

1. 创建镜像任务

用户在镜像管理页，点击【创建镜像】后，进入新建镜像页。

2. 新建镜像页信息。

需用户填写镜像名称、版本号、应用范围，镜像描述信息等。

平台内置支持了多框架/版本/cuda/cudnn 的基础镜像供选择, 优先选择最贴近需求的基础镜像, 减少多余构建。

用户填写完成信息后, 点击【确定】后, 进入镜像详情页。

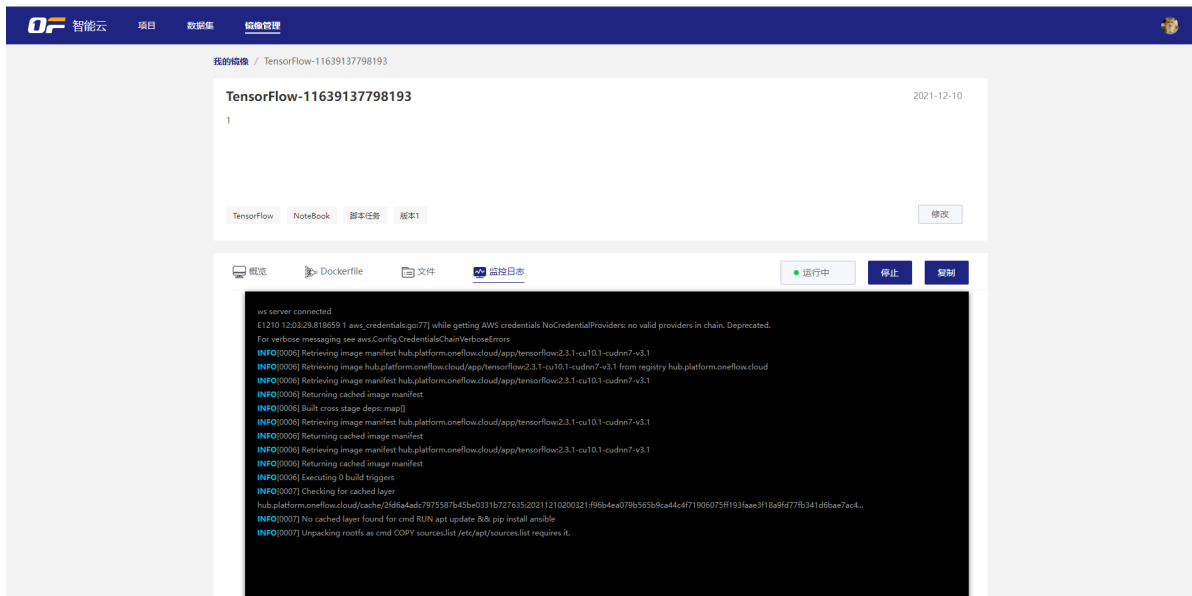
3. 编辑dockerfile并点击【运行】提交构建镜像。

1) 基础镜像已经替换对应系统源为mirror.aliyun.com, dockerfile编写时无需再次替换。

2) 容器内cuda|cudnn不建议通过dockerfile修改, 可能会构建镜像不成功。

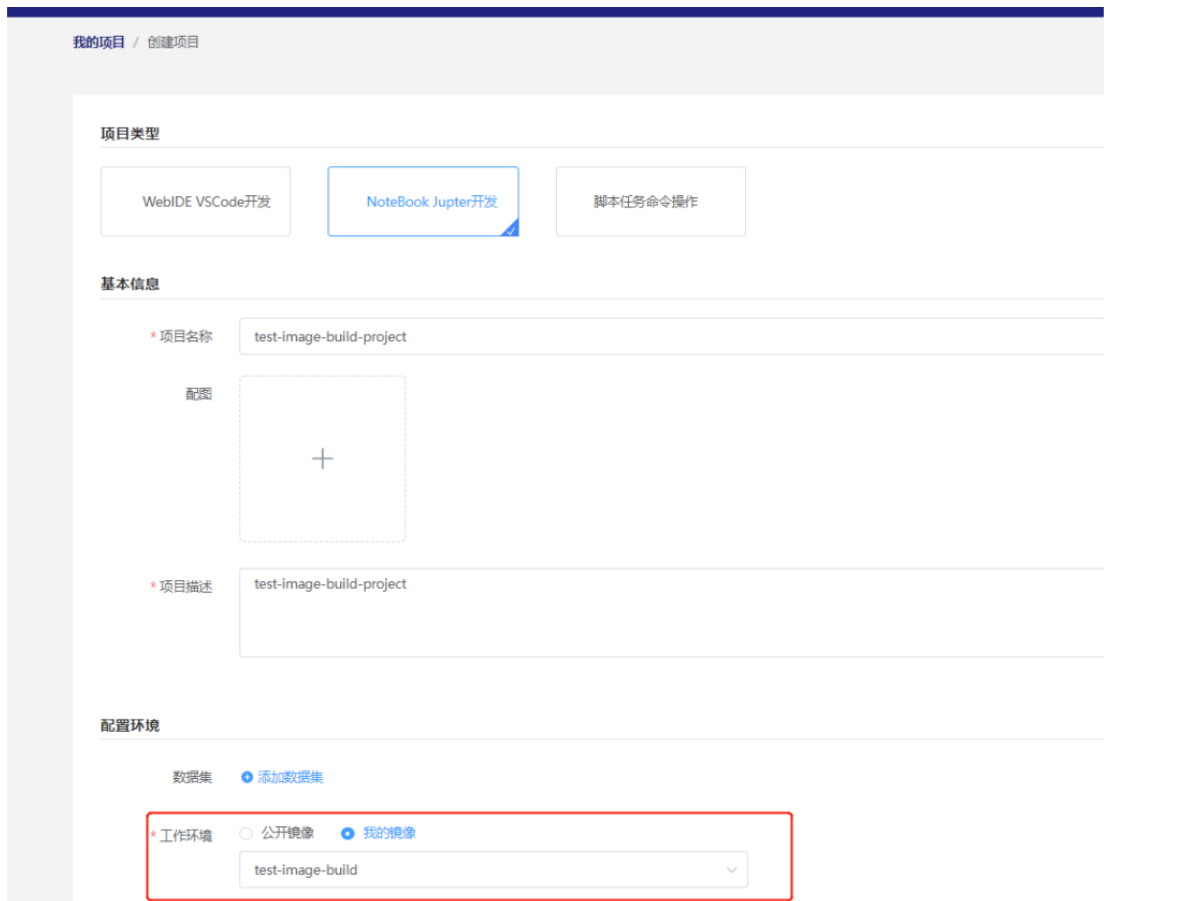
4. 创建成功之后直接运行构建任务, dockerfile不做修改, 等待确认构建成功, 就完成了—一个基础镜像的私有镜像。

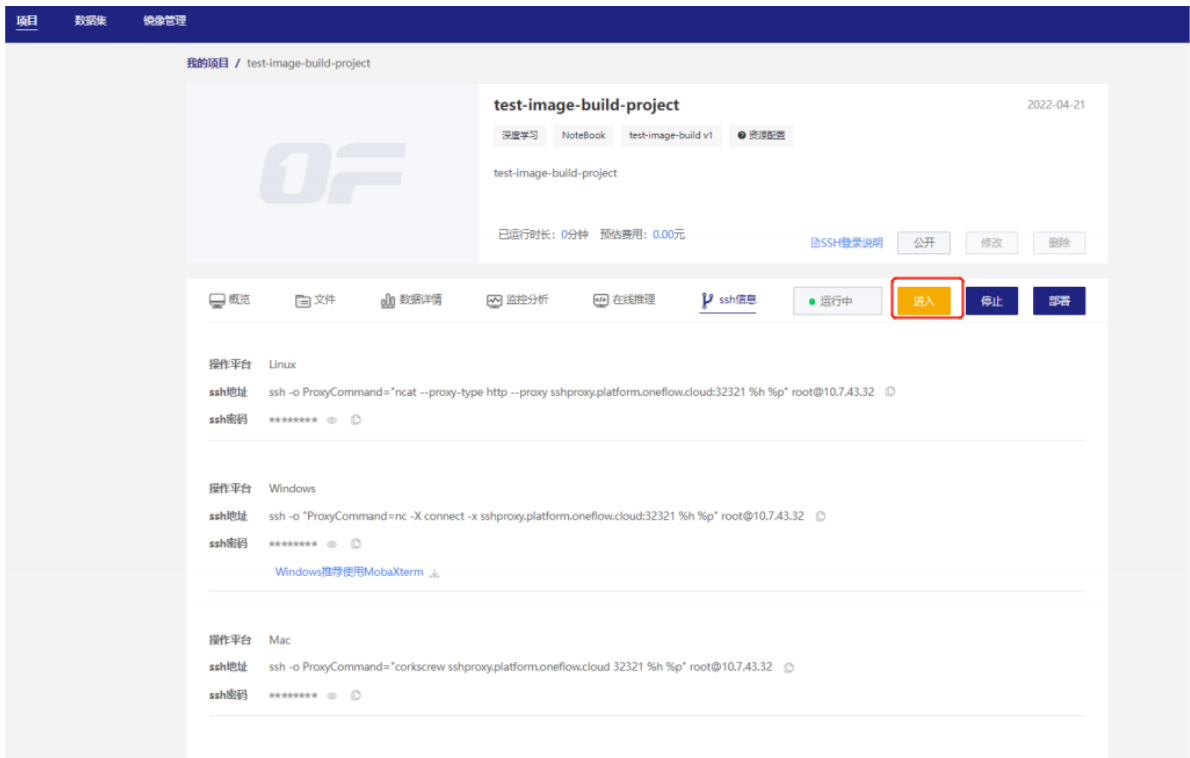
日志信息通过采集器拉取, 非实时日志, 如果提示暂无日志输出, 可稍后查看。



测试基础镜像并安装调试

1. 创建项目，选择我的镜像，上面构建的基础镜像，并运行。





2. SSH 登录远程构建/安装流程

基于成功安装的测试, 可以有效减少dockerfile的构建调试工作。

比如这里测试并安装了libibverbs-dev。

```
(base) ~ - ofcloud project ls
auth failed
code: 401, msg: refreshToken缓存失效
username: 17710636396
password: *****
auth OK:
tokens: *****
fetchTime: 2022-04-21 19:54:34 +0800 CST
expireIn: 86398 seconds
```

NAME	PRODUCTNAME	IMAGENAME	FORKFROM	DISTRIBUTED	RUNNINGTYPE	COMMAND	STATUS	COST
test-image-build-project	16core-32Gi-nlu770(4Card)	test-image-build	-	false	notebook	sleep 365d	running	0.22
tttt	6core-42Gi-3090(1Card)	oneflow-0.7.0+torch-1.8.1-cu11.2-cudnn8	-	false	task	sleep 365d	running	0.85

```
(base) ~ - ofcloud project ssh -n test-image-build-project

Welcome to OFCloud!

Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 4.15.0-20-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

(base) root@training-notebook-4e540b-4e540b-jupyter-master-0:/workspace#
```

```
libverbs-providers/focal,now 28.0-lubuntu1 amd64 [installed]
User space provider drivers for libibverbs

libverbs-utils/focal 28.0-lubuntu1 amd64
Examples for the libibverbs library

libibverbs-dev/focal 28.0-lubuntu1 amd64
Development files for the libibverbs library

libverbs1/focal,now 28.0-lubuntu1 amd64 [installed]
Library for direct userspace use of RDMA (InfiniBand/iWARP)

(base) root@training-notebook-4e540b-4e540b-jupyter-master-0:/workspace# apt update && apt install -y libibverbs-dev
Hit:1 http://mirrors.aliyun.com/ubuntu focal InRelease
Hit:2 http://mirrors.aliyun.com/ubuntu focal-security InRelease
Hit:3 http://mirrors.aliyun.com/ubuntu focal-updates InRelease
Hit:4 http://mirrors.aliyun.com/ubuntu focal-proposed InRelease
Hit:5 http://mirrors.aliyun.com/ubuntu focal-backports InRelease
Ign:6 https://developer.download.nvidia.cn/compute/cuda/repos/ubuntu2004/x86_64 InRelease
Hit:7 https://developer.download.nvidia.cn/compute/cuda/repos/ubuntu2004/x86_64 Release
Ign:9 https://developer.download.nvidia.cn/compute/machine-learning/repos/ubuntu2004/x86_64 InRelease
Hit:10 https://developer.download.nvidia.cn/compute/machine-learning/repos/ubuntu2004/x86_64 Release
Reading package lists ... Done
Building dependency tree
Reading state information ... Done
130 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists ... Done
Building dependency tree
Reading state information ... Done
The following additional packages will be installed:
  libnl-3-dev libnl-route-3-dev
The following NEW packages will be installed:
  libibverbs-dev libnl-3-dev libnl-route-3-dev
0 upgraded, 3 newly installed, 0 to remove and 130 not upgraded.
Need to get 702 kB of archives.
After this operation, 3417 kB of additional disk space will be used.
Get:1 http://mirrors.aliyun.com/ubuntu focal/main amd64 libnl-3-dev amd64 3.4.0-1 [92.2 kB]
Get:2 http://mirrors.aliyun.com/ubuntu focal/main amd64 libnl-route-3-dev amd64 3.4.0-1 [166 kB]
Get:3 http://mirrors.aliyun.com/ubuntu focal/main amd64 libibverbs-dev amd64 28.0-lubuntu1 [444 kB]
Fetched 702 kB in 0s (5002 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libnl-3-dev:amd64.
(Reading database ... 48785 files and directories currently installed.)
Preparing to unpack .../libnl-3-dev_3.4.0-1_amd64.deb ...
Unpacking libnl-3-dev:amd64 (3.4.0-1) ...
Selecting previously unselected package libnl-route-3-dev:amd64.
Preparing to unpack .../libnl-route-3-dev_3.4.0-1_amd64.deb ...
Unpacking libnl-route-3-dev:amd64 (3.4.0-1) ...
Selecting previously unselected package libibverbs-dev:amd64.
Preparing to unpack .../libibverbs-dev_28.0-lubuntu1_amd64.deb ...
Unpacking libibverbs-dev:amd64 (28.0-lubuntu1) ...
Setting up libnl-3-dev:amd64 (3.4.0-1) ...
Setting up libnl-route-3-dev:amd64 (3.4.0-1) ...
Setting up libibverbs-dev:amd64 (28.0-lubuntu1) ...
```

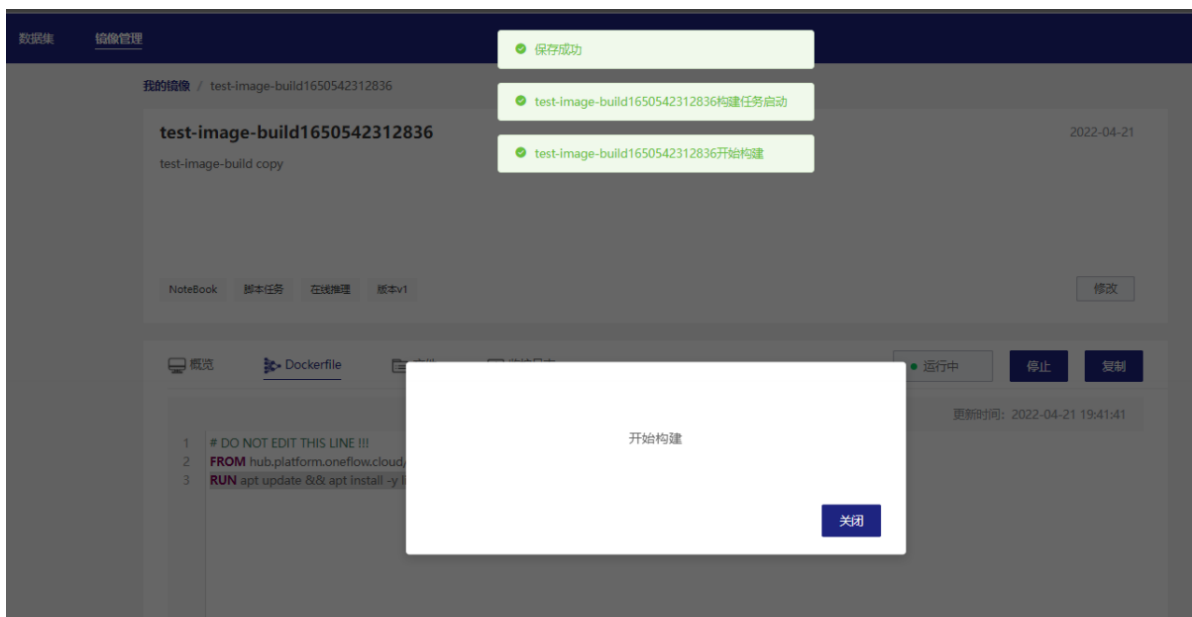
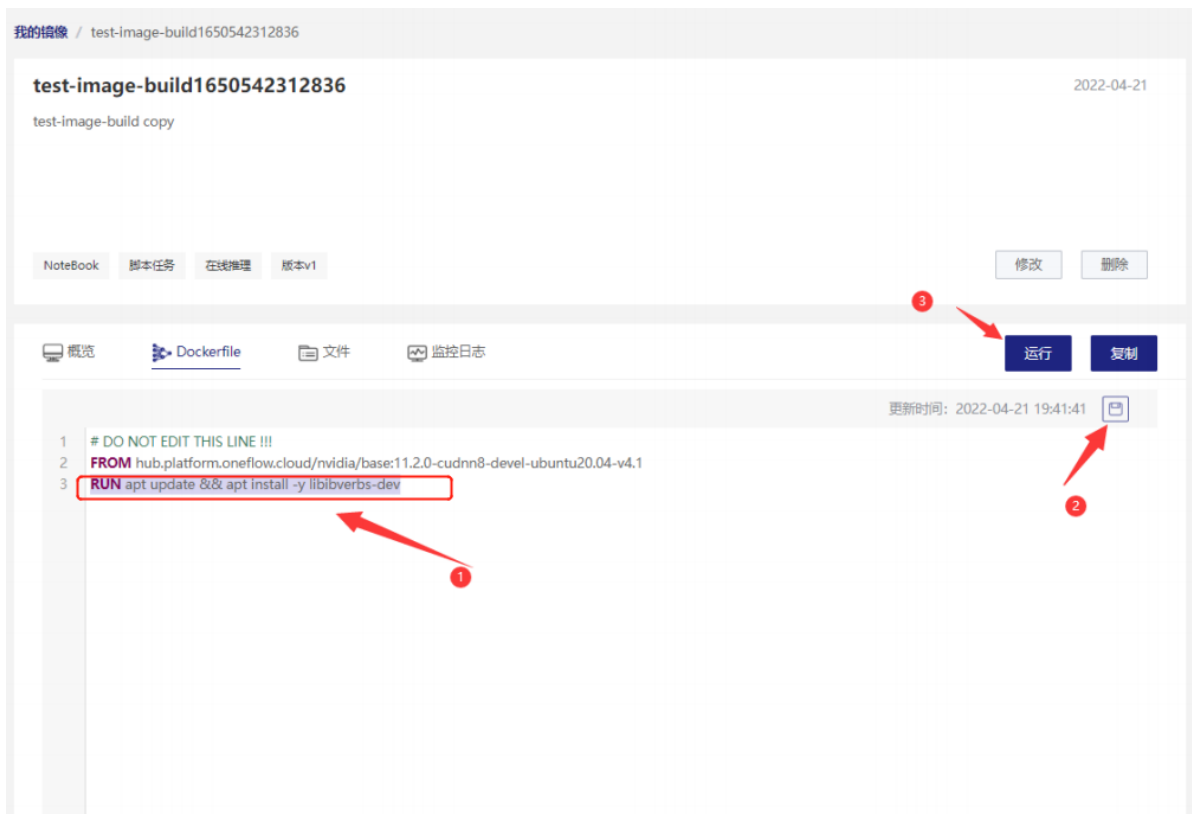
重新构建镜像

返回我的镜像中选择构建的基础记性，复制后编辑新的dockerfile，并重新构建。

The screenshot shows a web interface for managing Docker images. At the top, it displays 'test-image-build' with a timestamp of '2022-04-21'. Below this, there are tabs for 'NoteBook', '脚本任务', '在线推理', and '版本v1'. On the right side, there are buttons for '修改' (Modify) and '删除' (Delete). A progress bar at the bottom indicates the build is '成功' (Success), with a '复制' (Copy) button highlighted in a red box. The build log shows the following content:

```
1 # DO NOT EDIT THIS LINE !!!
2 FROM hub.platform.oneflow.cloud/nvidia/base:11.2.0-cudnn8-devel-ubuntu20.04-v4.1
```

The update time is shown as '更新时间: 2022-04-21 19:41:41'.



构建成功镜像，创建项目时即可选用新的构建好的镜像应用。

常见问题

Q1: 资源如何计费?

AI 实训平台模型训练和部署服务，均需申请资源，先使用后付费原则。

具体规则如下：

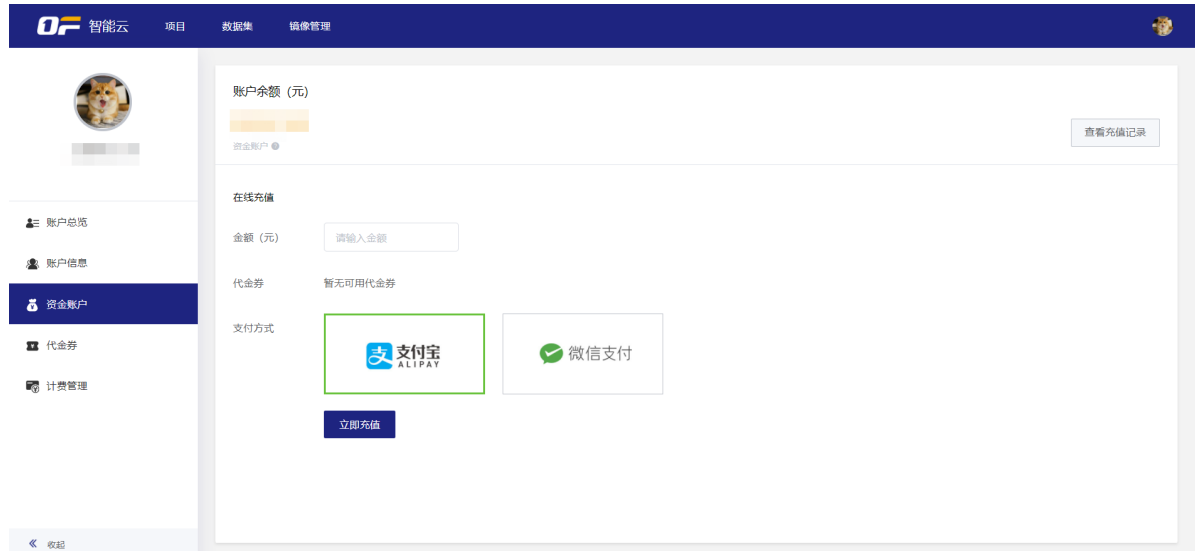
- 在训练任务和部署推理任务启动成功后，系统定时每一小时扣费，如果扣费后账户可用余额<0时，将会停止任务，并下发短信通知用户补足欠费。
- 在训练任务和部署推理任务启动成功后实时计费，请保证账户余额充足，如不用请及时停止释放任务。
- 按分钟计费，不足 1 分钟按 1 分钟计。
- 训练任务或部署推理任务启动时，校验用户账户是否欠费，当账户余额<=0 时，不能启动任务。

- 计费公式=资源单价 × 节点数 × 使用时长
- 运行时长=任务状态为运行中的计费时长

如果因为 AI 实训平台系统异常导致训练任务失败，则不会参与计费。

Q2: 如何充值?

OF 智能云提供支付渠道有微信和支付宝，用户使用支付渠道扫码即可支付。



Q3: 文件或安装工具包如何持久化存储?

容器内自行安装的工具包默认是临时的，下次再运行项目的时候会重置初始化，用户自定义构建镜像解决。

Q4: 数据集挂载目录?

- 数据集被存放在磁盘上的/dataset 文件夹中，/dataset 就是数据集的挂载目录。
- 挂载数据集信息，例/dataset/decada4c/v1，项目下/dataset/数据集 ID/版本号下。

Q5: 训练任务启动失败原因?

可能是以下原因等

- 资源不足
- 拉取镜像失败
- 命令行错误

具体失败原因请查看日志分析。

SDK 参考

onebrain-sdk 上传训练监控指标

安装

首先镜像里需安装 onebrain-sdk。可以采用 whl 包本地安装的形式。

一般来说，平台提供的镜像里应该已经集成 onebrain-sdk 了。

如未集成，可以手动安装：

```
# 打最新的whl包
git clone https://gitee.com/onebrain/onebrain-sdk.git
python setup.py bdist_wheel -d onebrain
```

将 whl 包上传至镜像

```
# 安装whl包
pip install onebrain-0.0.9-py3-none-any.whl
```

因为 onebrain-sdk 会迭代更新，并发布到<https://test.pypi.org>，也可以通过网络直接安装最新版：

```
pip install -i https://test.pypi.org/simple/ onebrain
```

SDK 说明

使用示例

```
from onebrain.api import OnebrainClient
#创建与平台通信的client对象
client = OnebrainClient()
```

然后就可以通过 client 对象来上传指标到平台了。

以 pytorch 训练任务为例

```
# ...
def train(data_loader:DataLoader, model:nn.Module, loss_fn,
optimizer:torch.optim.Optimizer, step):
    image_size = len(data_loader.dataset)
    print(f"数据集的图片总数: {image_size}")

    for batch, (x_train, y_train) in enumerate(data_loader):
        x_train, y_train = x_train.to(device), y_train.to(device)
        y_hat = model(x_train)
        loss = loss_fn(y_hat, y_train)
        loss.backward()
        optimizer.step()
        optimizer.zero_grad()

        if batch % 100 == 0:
            loss, current = loss.item(), batch * len(x)
            print(f'batch:{batch}, loss: {loss}, , image_num:
{current}/{image_size}')
            # 上传指标
            client.log_metric(key="loss", value=loss, step=step)

    for epoch in range(epochs):
        print(f"Epoch {epoch}\n -----")
        train(train_data_loader, model, loss_fn, optimizer, epoch)
```

log_metric 方法参数说明：

run_id: 训练任务的唯一标识，应从环境变量获取，一般为平台生成的 resource_id，sdk 自动填充。

key: 指标名称。

value: 指标值。

timestamp: 时间戳, 不填默认为当前时间。

step: 训练的 step。

CLI开发工具

ssh代理 proxy

提供基于 netcat 的代理穿透功能, 在不依赖第三方工具的情况下代理至 [OF智能云](#) 实训平台项目

```
# ssh 调用 链路
```

```
本地服务器(安装ofcloud 工具)
```

```
|
```

```
----↓
```

```
OF智能云SSH跳转服务器
```

```
|
```

```
----↓
```

```
OF智能云实训平台项目
```

host: 必选, 指定目标主机

port: 必选, 指定目标端口

```
$ ofcloud proxy --help
use ofcloud cli toolset as ssh proxy

Usage:
  ofcloud proxy [flags]

Flags:
  -h, --help           help for proxy
  --host string        target host
  --port string        target port (default "22")

Global Flags:
  -c, --config string  config file (default is
$HOME/.config/onebrain/ofcloud.toml)
```

使用示例

```
# 获取实训平台项目ssh信息
```

```
$ ofcloud project ssh --info -n torch
```

```
Project Master Connection Info:
```

```
+ Platform      + Address      + Password
```

```
+ Cmd
```

```
+-----+-----+-----
```

```
+-----+-----+-----
```

```
-
```

```

+ Linux + root@10.7.174.85 + aJzDMuL5
+ ssh -o ProxyCommand="ncat --proxy-type http --proxy sshproxy.pre-
platform.oneflow.cloud:32321 %h %p" root@10.7.174.85
+ Windows + root@10.7.174.85 + aJzDMuL5
+ ssh -o "ProxyCommand=nc -X connect -x sshproxy.pre-
platform.oneflow.cloud:32321 %h %p" root@10.7.174.85
+ Mac + root@10.7.174.85 + aJzDMuL5
+ ssh -o ProxyCommand="corkscrew sshproxy.pre-platform.oneflow.cloud
32321 %h %p" root@10.7.174.85

```

Project Pod Info:

```

+-----+
|  POD IP  | STATUS |          POD NAME          |
+-----+
| 10.7.174.85 |      | training-task-21fa21v13-train-master-vrtbg |
+-----+

```

```

# 使用ssh客户端配合ofcloud跳转(只需关注上一步中的`address`和`password`)
$ ssh -o ProxyCommand="ofcloud proxy --host %h --port %p" root@10.7.174.85

```

output

```

Warning: Permanently added '10.7.174.85' (ED25519) to the list of known hosts.
root@10.7.174.85's password:
Warning: No xauth data; using fake authentication data for X11 forwarding.
X11 forwarding request failed on channel 0
welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-90-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Thu Mar  3 10:27:55 2022 from 10.5.0.46
(base) root@training-task-21fa21v13-train-master-vrtbg:/workspace#

```

~/.ssh/config 配置参考

```

$ cat ~/.ssh/config
Host ofcloud-*
    ProxyCommand ofcloud proxy --host %h --port %p

Host ofcloud-torch
    hostname 10.7.174.85
    user root
    port 22

$ ssh ofcloud-torch
Warning: Permanently added '10.7.174.85' (ED25519) to the list of known hosts.
root@10.7.174.85's password:
Warning: No xauth data; using fake authentication data for X11 forwarding.

```



```
X11 forwarding request failed on channel 0  
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-90-generic x86_64)
```

```
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage
```

This system has been minimized by removing packages and content that are not required on a system that users **do** not log into.

To restore this content, you can run the **'unminimize'** command.

```
Last login: Thu Mar  3 11:55:36 2022 from 10.5.0.46
```

```
(base) root@training-task-21fa21v13-train-master-vrtbg:/workspace#
```

[下载 PDF](#)
