

MLFlow使用手册(MLFlowGuide)

[toc]

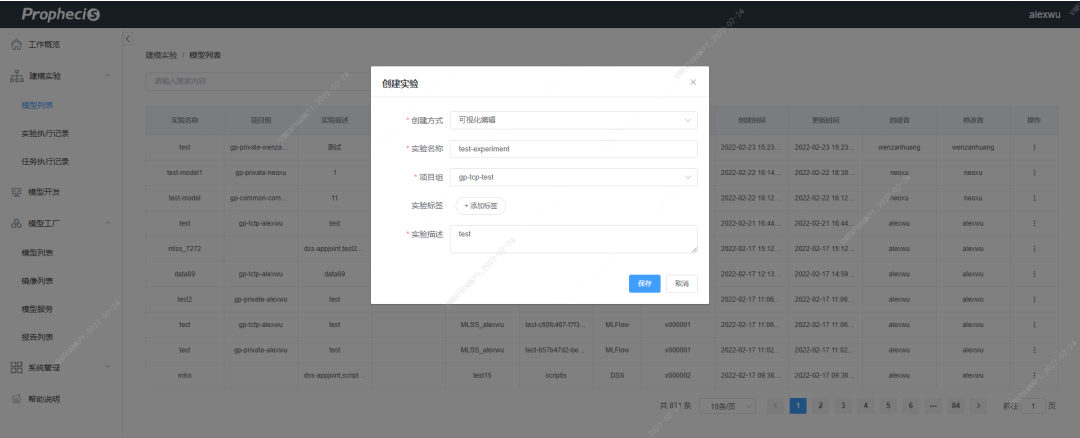
1. MLFlow实验管理简介

MLFlow实验管理是Prophecis系统核心模块，用于验证模型、节点。有实验管理、实验执行记录、任务执行记录三大块组成。

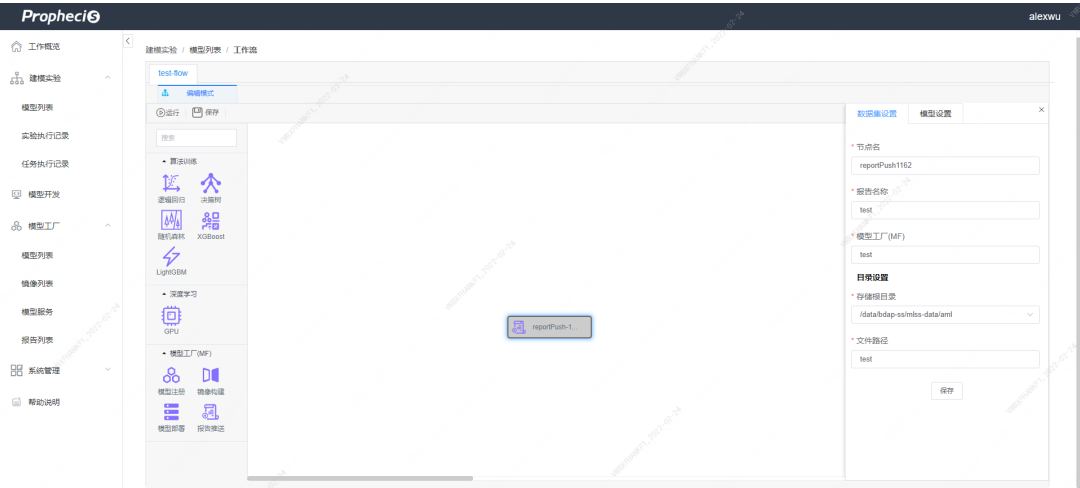
2. 实验管理

2.1 创建实验

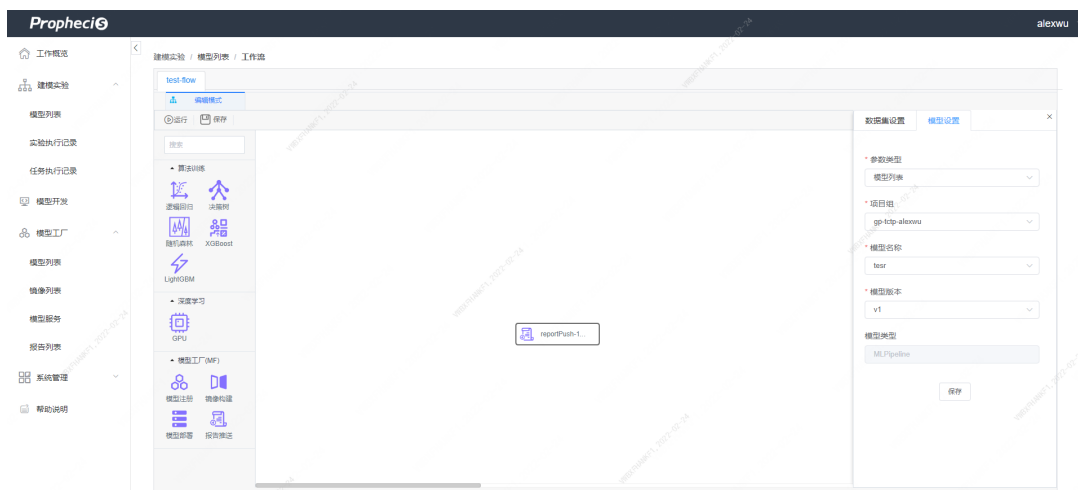
- 点击实验列表，创建实验



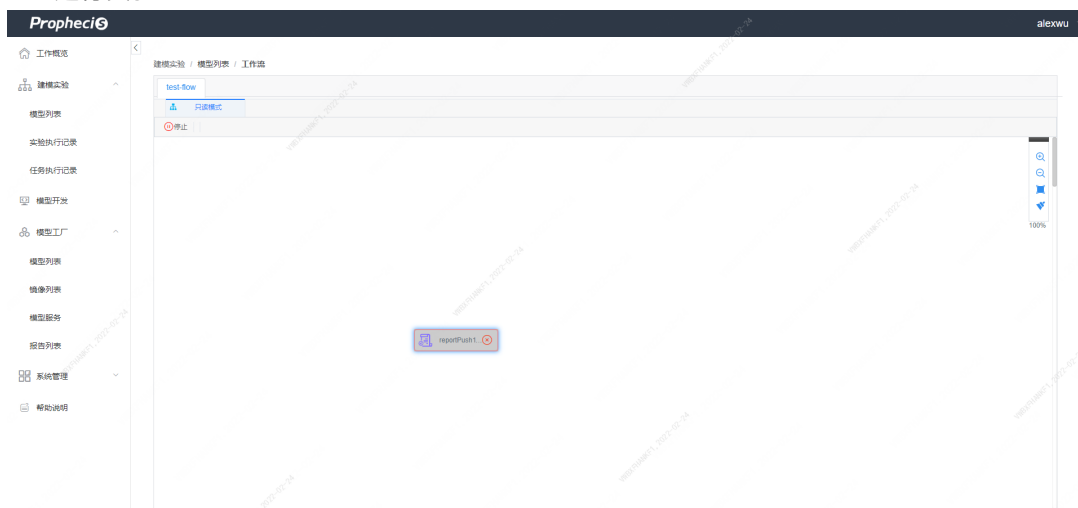
- 编辑实验，创建实验任务，数据集设置



- 模型设置

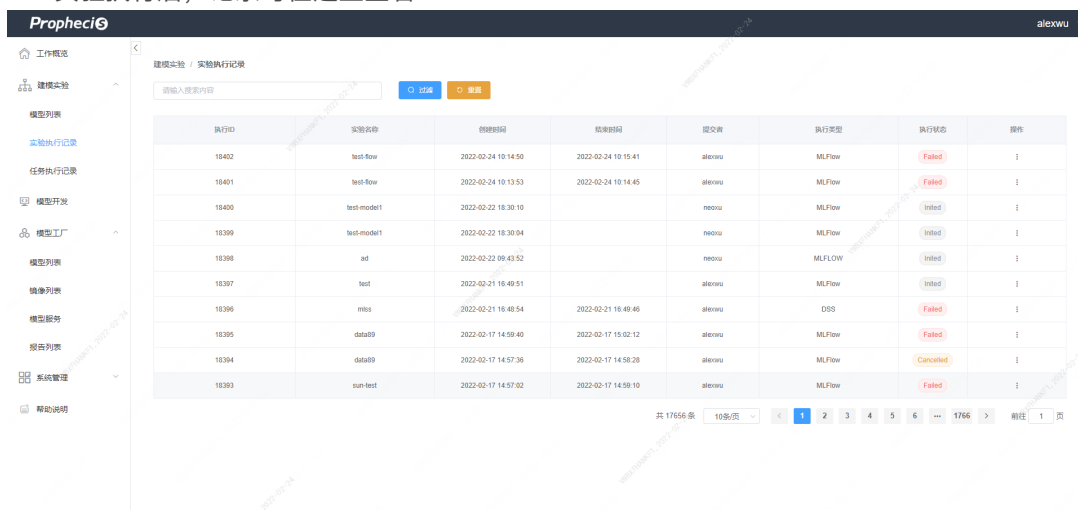


运行实验



2.2. 实验执行

- 实验执行后，记录可在这里查看



- 在操作中点击任务列表，可查看该实验的任务执行记录

| | | | | | | | | | |
|----------|-----------------------------------------------------------------------------------|--|--|--|--|--|--|--|--|
| Prophecy | | | | | | | | | |
| 工作概览 | 构建实验 / 任务执行记录 | | | | | | | | |
| 构建实验 | 命名空间 <input type="text"/> 用户 <input type="text"/> 创建 删除 | | | | | | | | |
| 模型列表 | | | | | | | | | |
| 实验执行记录 | | | | | | | | | |
| 任务执行记录 | | | | | | | | | |
| 模型开发 | | | | | | | | | |
| 模型工厂 | | | | | | | | | |
| 模型列表 | | | | | | | | | |
| 镜像列表 | | | | | | | | | |
| 模型服务 | | | | | | | | | |
| 服务列表 | | | | | | | | | |
| 系统管理 | | | | | | | | | |
| 帮助说明 | | | | | | | | | |

| 任务执行ID | 任务名称 | 实验执行ID | 实验名称 | 提交时间 | 结束时间 | 提交者 | 执行状态 | 操作 |
|-----------------------------|----------|--------|----------|---------------------|---------------------|--------|-----------|-----|
| training-zwt13bt1-e6Stukmy9 | GPU-5016 | 16377 | mlh_6411 | 2022-02-16 15:33:43 | 2022-02-16 15:33:59 | alexwu | COMPLETED | [-] |

共 1 条 10条/页 < 1 > 前往 1 页

实验执行状态：Succeed Canceled Initiated Failed UNKNOWN Running

3. 任务创建(单机及分布式任务)

3.1 创建GPU集群任务

点击创建GPU集群任务后，根据创建任务的引导填写任务相关信息，创建任务需填写以下信息：基础信息设置、镜像设置、计算资源设置、训练任务目录设置、任务告警设置、任务执行设置。

3.1.1 单机任务填写范例

1. 基础信息配置

- 任务名称及备注：配置任务名称、任务描述等基本信息

| | | | | | | | | | |
|----------|-----------------------------------------------------------------------------------|--|--|--|--|--|--|--|--|
| Prophecy | | | | | | | | | |
| 工作概览 | 分布式训练 / 分布式训练 | | | | | | | | |
| 分布式训练 | 命名空间 <input type="text"/> 用户 <input type="text"/> 创建 删除 | | | | | | | | |
| 模型开发 | | | | | | | | | |
| 系统管理 | | | | | | | | | |
| 基础平台管理 | | | | | | | | | |
| 用户信息管理 | | | | | | | | | |
| 项目组信息管理 | | | | | | | | | |
| 命名空间管理 | | | | | | | | | |
| 计算节点管理 | | | | | | | | | |
| 资源管理 | | | | | | | | | |
| 帮助说明 | | | | | | | | | |

创建训练任务

基础信息设置

镜像设置

计算资源设置

训练任务目录设置

任务执行设置

任务告警设置

基础信息设置

* 训练任务名称

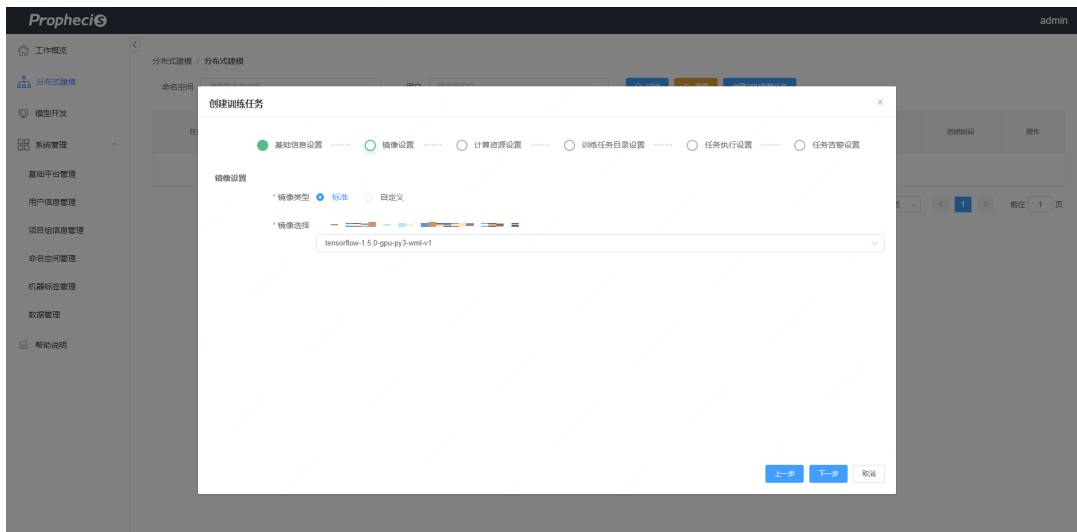
* 描述

下一步

取消

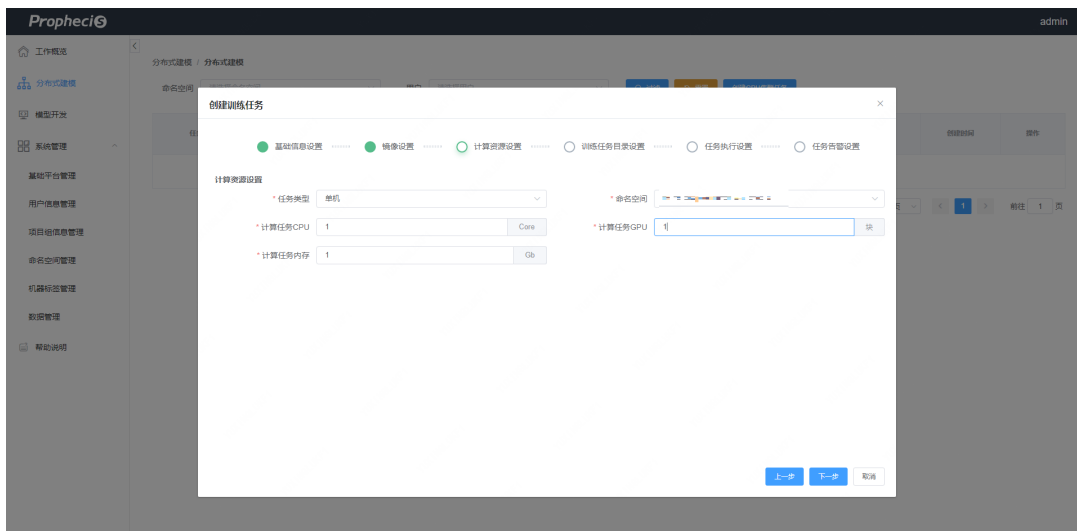
2. 镜像配置

- 镜像类型：标准为列表中提供的镜像，自定义可设置任意Tag
- 镜像名称：Docker Image Tag



3. 计算资源配置

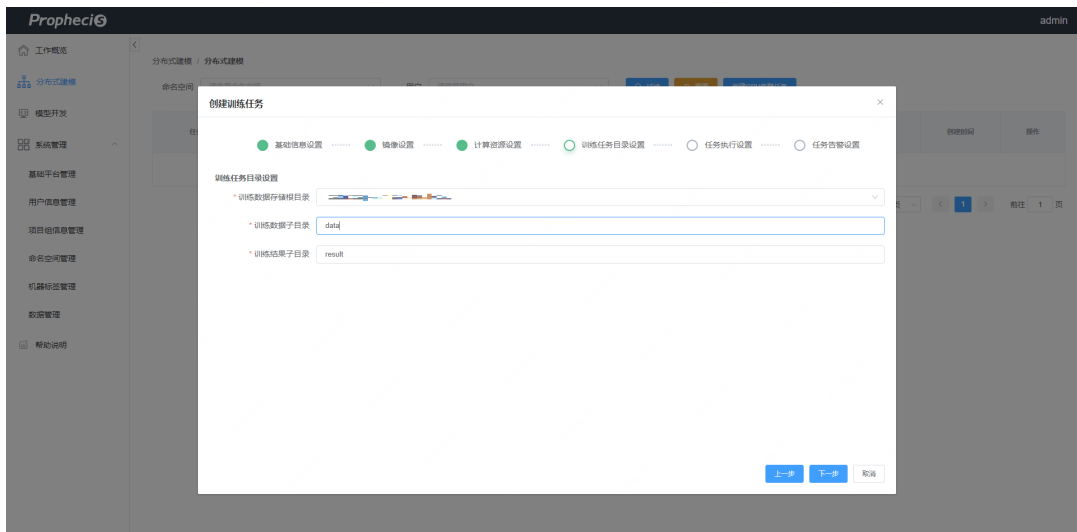
- 命名空间：命名空间会与硬件机器做绑定，对应命名空间下的任务会启动到对应的服务器上面。
- 任务类型：在任务资源页面任务类型可选择单机或分布式，分布式任务即tensorflow的分布式模式，下图为单机模式的范例。
- CPU/GPU/Mmemory：填写执行任务所需的CPU/GPU/Memory即可。



4. 训练任务目录配置

训练任务目录配置进行存储以及相关任务的目录。

- 训练数据存储根目录：代表宿主机目录（即执行代码机器的目录）
- 训练数据子目录：通常作为存放数据的目录，和DATA_DIR相对应。
- 训练结果子目录：通常作为存放日志的目录，和RESULT_DIR相对应。

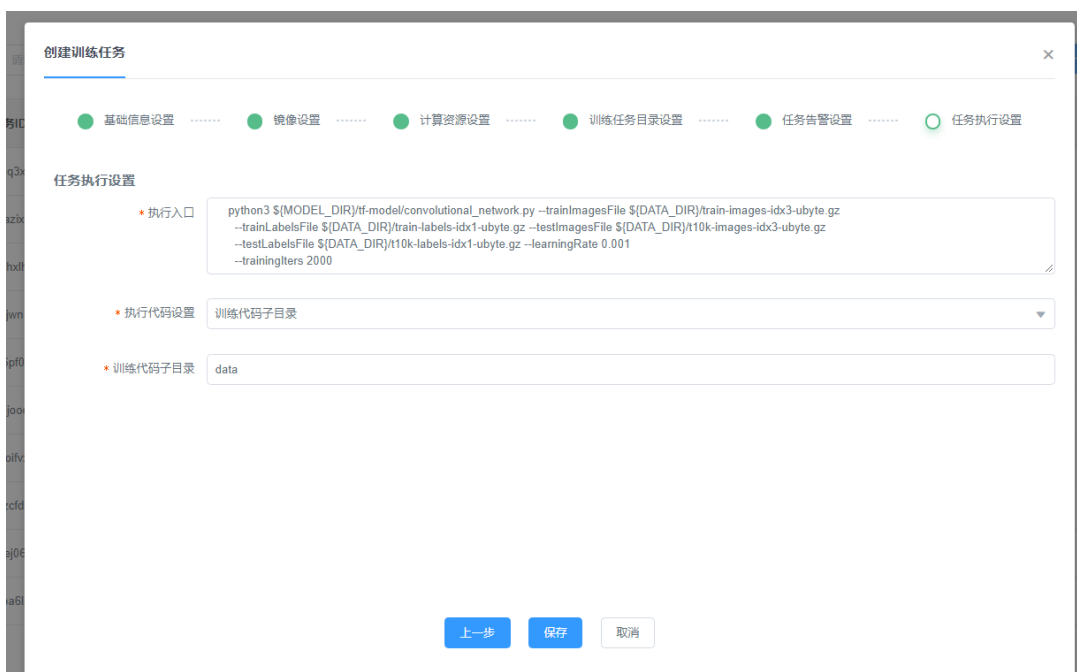


5. 任务执行设置

- 执行入口设置：

配置执行任务的代码入口，提交方式可以为上传代码和训练代码子目录。

- 执行代码设置：
 - 上传代码：手动上传代码压缩包
 - 训练代码子目录：使用宿主机共享目录中的代码
 - 环境变量：设置完代码目录，会在容器生成一下环境变量：
 - `${MODEL_DIR}`:
 - 选择训练代码子目录时，宿主机路径为训练数据存储根目录+训练代码子目录
 - 选择上传代码时，依据解压代码zip后的代码文件所在位置填写即可
 - `${DATA_DIR}`: 宿主机路径为 训练数据存储根目录+训练数据子目录
 - `${RESULT_DIR}`: 宿主机路径为 训练数据存储根目录+训练结果子目录



7. 任务告警配置

告警配置，配置任务执行的相关告警，可以添加多个告警，也可以不配置。

3.1.2 分布式任务填写范例

分布式任务和单机任务填写的唯一差别页面在资源配置页面。当选择为分布式时，分布式任务需要额外填写参数服务器数量，参数服务器CPU，参数服务器内存，参数服务器镜像。

资源配置页填写范例：

并且这部分信息会通过环境变量映射到代码中，可以通过环境变量设置

`Tensorflow.train.ServerDef`：

```
1 tf_config_json = os.environ.get("TF_CONFIG", "{}")
2 tf_config = json.loads(tf_config_json)
3 task = tf_config.get("task", {})
4 cluster_spec = tf_config.get("cluster", {})
5 cluster_spec_object = tf.train.ClusterSpec(cluster_spec)
```

```
job_name = task["type"]
task_id = task["index"]
server_def = tf.train.ServerDef(
    cluster=cluster_spec_object.as_cluster_def(),
    protocol="grpc",
    job_name=job_name,
    task_index=task_id)
server = tf.train.Server(server_def)
```

3.2 查看任务日志

任务列表可以看到当前任务的执行状态，更详细的任务执行情况可以通过查看日志，日志会包含任务的一些信息，同时py执行时的控制台输出也会显示在上面。当任务处于COMPLETED时，任务处于完成状态，之后可到相应result目录查看结果。

Prophecy

工作概述

实验记录

实验列表

实验执行记录

任务执行记录

模型开发

模型工厂

模型列表

模型部署

模型服务

系统管理

帮助说明

任务执行记录

命名空间: 选择命名空间

用户: 选择用户

Q. 检索

重置

| 任务执行ID | 任务名称 | 实验执行ID | 实验名称 | 提交时间 | 结束时间 | 提交者 | 执行状态 | 操作 |
|-----------------------------|------------|--------|------|---------------------|---------------------|-------|-----------|----|
| training-01amcd9lpcscmbly6 | test-cl-ga | | | 2021-06-30 18:35:20 | 2021-07-05 15:14:47 | ●●●●● | COMPLETED | ↓ |
| training-01qlydy6lpcscmbly6 | test-cl-ga | | | 2021-06-30 17:35:20 | 2021-07-05 15:14:37 | ●●●●● | COMPLETED | ↓ |
| training-r0d7d9qpl0dscmbly6 | test-cl-ga | | | 2021-06-29 10:35:58 | 2021-07-05 15:14:26 | ●●●●● | COMPLETED | ↓ |
| training-dyld9d0dscmbly6 | test-cl-ga | | | 2021-06-24 02:35:21 | 2021-06-24 02:35:29 | ●●●●● | COMPLETED | ↓ |
| training-7xld9d0dscmbly6 | test-cl-ga | | | 2021-06-24 01:35:20 | 2021-06-24 01:35:37 | ●●●●● | COMPLETED | ↓ |
| training-qpld9d0dscmbly6 | test-cl-ga | | | 2021-06-24 00:35:21 | 2021-06-24 00:35:34 | ●●●●● | COMPLETED | ↓ |
| training-dfpld9d0dscmbly6 | test-cl-ga | | | 2021-06-23 23:35:21 | 2021-06-23 23:35:32 | ●●●●● | COMPLETED | ↓ |
| training-0d5ld9d0dscmbly6 | test-cl-ga | | | 2021-06-23 22:35:21 | 2021-06-23 22:35:30 | ●●●●● | COMPLETED | ↓ |
| training-0d2ld9d0dscmbly6 | test-cl-ga | | | 2021-06-23 21:35:21 | 2021-06-23 21:35:28 | ●●●●● | COMPLETED | ↓ |
| training-buagld9d0dscmbly6 | test-cl-ga | | | 2021-06-23 20:35:21 | 2021-06-23 20:35:36 | ●●●●● | COMPLETED | ↓ |

共 3270 条 100页 < 1 2 3 4 5 6 ... 327 > 前往 1 页

Prophecy
Windows

- 工作台
- 新建实验
- 实验列表
- 实验执行记录
- 任务执行记录
- 模型开发
- 模型工厂
- 模型列表
- 模型部署
- 模型保存
- 系统管理
- 帮助说明

新建实验 | 任务执行记录 | 日志

任务ID: training-dtkmciqplacxnddy6

RESULT_DIR: /mnt/results/result

total 12

```
MODEL_DIR /job/model-code
Contents of MODEL_DIR
DATA_DIR /mnt/data/data
TRAINING_COMMAND echo 1
Storing trained model as:
Training with training/test data as:
TRAINING_DIR
drecvcrve 3 root root 4096 Jul 5 15:14 .
total 110360
Contents of DATA_DIR
drecvcrve 3 root root 4096 Jul 5 15:14 .
drecvcrve 4 root root 4096 Jul 5 15:14 .
drecvcrve 2 root root 4096 Jul 5 15:14 tf-model
--drecvcrve 1 s *** * **H7129405 Aug 28 2020 tlla.ov
drecvcrve 2 s *** * ** 4096 Jan 16 2020 logs
drecvcrve 3 s ** s1 **** 4096 Dec 23 2019 _RACXX
drecvcrve 3 s ** s1 ** 4096 Mar 11 10:07 .
drecvcrve 2 s **** s1 4096 Aug 28 2020 igymb_checkpoint
--drecvcrve 1 s ** s1 **** 28 Nov 13 2019 alar-test_4e30cc8c_B8D_6bb_M77_F05DeDFChA
--drecvcrve 1 s **** s1 30857 Mar 11 10:07 teststata.igymb
--drecvcrve 1 s *** s1 s1 s1 23512425 Dec 30 2019 tensorflow-sample-code.tar.gz
--drecvcrve 1 s *** s1 s1 s1 4542 Dec 19 2019 tllq-labelridr3-shyte.gz
drecvcrve 3 s ** s1 **** 4096 Dec 23 2019 tensorflow-sample-code
drecvcrve 3 s ** s1 **** 4096 Dec 23 2019 tensorflow-sample-code-hak
--drecvcrve 1 s *** s1 s1 s1 20210 Feb 15 2020 test.ov
--drecvcrve 1 164877 Dec 19 2019 tllq-lapaperid3-shyte.gz
```

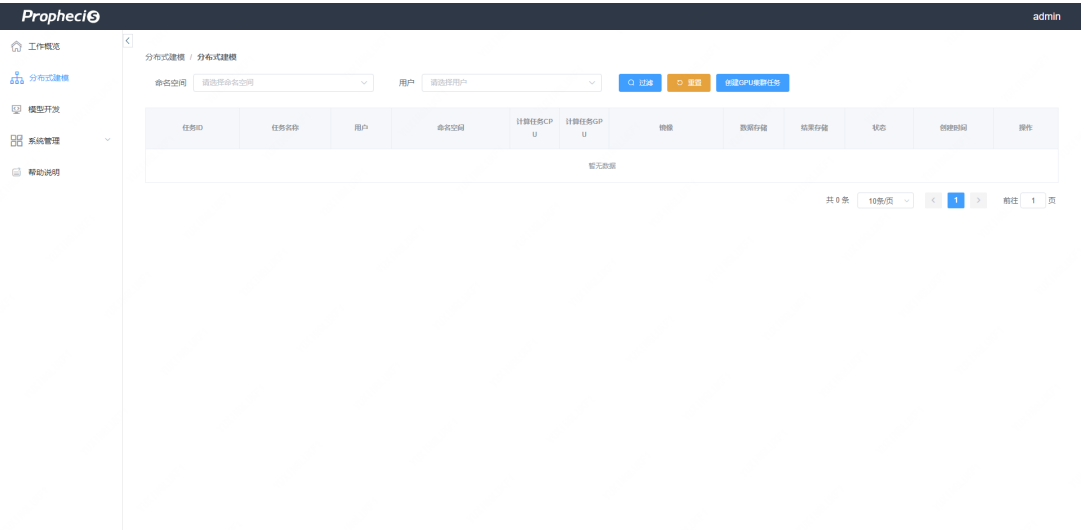
- 查看已提交的任务配置详情
- 查看任务训练日志
- 删除现有任务
 - 如果任务正在运行，会马上进行容器清除工作并删除

- 如果任务已经已完成的任务则会删除历史信息

3.3 任务列表页面

该页面中包含已创建任务得主要信息，包括任务ID、任务名称、命名空间等，其中：

- 命名空间（namespace）：管理员用户通过CC控制台创建，与所用硬件资源相关。
- 状态：任务的运行状态，分为QUEUED、PENDING、RUNNING、FAILED、COMPLETED。



4. CLI使用

4.1 CLI介绍

CLI是Prophecis-DI提供的命令行工具，可以对Prophecis-DI进行基本的命令操作，包括提交任务、查看任务列表、查看任务日志等。使用CLI之前必须先进行基本的部署操作（获取CLI文件以及环境配置配置）。

4.2 CLI基本命令

- 执行\$CLI_CMD help可以查看命令的相关帮助

```
1 $ $CLI_CMD help
2 NAME:
3     bluemix dl - A new cli application
4 USAGE:
5     ffdl-linux [global options] command [command options] [arguments...]
6 COMMANDS:
7     init      bx dl init NAME
8     train     bx dl train MANIFEST_FILE
9              (MODEL_DEFINITION_ZIP|MODEL_DEFINITION_DIR)
10    show      bx dl show MODEL_ID
11    delete   bx dl delete MODEL_ID
12    list      bx dl list
```

```

12  download  bx dl download MODEL_ID (--definition|--trainedmodel) [--
    filename FILENAME]
13  logs      bx dl logs MODEL_ID [--follow] [--metrics]
14  loglines  bx dl loglines MODEL_ID [--follow] [--metrics]
15  emetrics  bx dl emetrics MODEL_ID [--follow] [--metrics]
16  halt      bx dl halt MODEL_ID
17  version   bx dl version
18  help, h   Shows a list of commands or help for one command
19  GLOBAL OPTIONS:
20  --help, -h  show help

```

\$CLI_CMD train

- 执行 `$CLI_CMD train <用户代码文件夹>/<任务描述文件> <用户代码文件夹>` 命令 可以提交任务到Prophecis-DI，任务描述文件代表提交任务的相关参数。

4.3 任务定义manifest文件

- 通过上传代码提交任务方式，manifest文件模板：

```

1  name: tf-convolutional-network-tutorial
2  description: Convolutional network model using tensorflow
3  version: "1.0"
4  gpus: 1
5  cpus: 1
6  memory: 1Gb
7  # 资源所在的命名空间
8  namespace: ns-common-common-prophecis--test
9  code_selector: codeFile
10 data_stores:
11   - id: hostmount
12     type: mount_volume
13     training_data:
14       container: data
15     training_results:
16       container: result
17     connection:
18       type: host_mount
19       name: "host-mount"
20       # 用户目录
21       path: "/data/bdp/prophecis-data/admin"
22 framework:
23   # 镜像仓库
24   name: "wedatasphere/di"
25   # 镜像名字
26   version: "tensorflow-1.14.0-gpu-py3"
27   command: >
28     python3 ${MODEL_DIR}/tf-model/convolutional_network.py --
    trainImagesFile ${DATA_DIR}/train-images-idx3-ubyte.gz

```

```

29     --trainLabelsFile ${DATA_DIR}/train-labels-idx1-ubyte.gz --
testImagesFile ${DATA_DIR}/t10k-images-idx3-ubyte.gz
30     --testLabelsFile ${DATA_DIR}/t10k-labels-idx1-ubyte.gz --
learningRate 0.001
31     --trainingIters 2000

```

- 通过通过训练代码子目录提交任务，manifest文件模板：

```

1  name: tf-convolutional-network
2  description: Convolutional network model using tensorflow
3  version: "1.0"
4  gpus: 1
5  cpus: 1
6  memory: 1Gb
7  # 资源所在的命名空间
8  namespace: ns-common-common-test
9  code_selector: storagePath
10 data_stores:
11   - id: hostmount
12     type: mount_volume
13     training_data:
14       container: data
15     training_results:
16       container: result
17     training_workspace:
18       container: workspace
19     connection:
20       type: host_mount
21       name: "host-mount"
22       # 用户目录
23       path: "/data/bdp/prophecis-data/admin"
24 framework:
25   # 镜像仓库
26   name: "wedatasphere/di"
27   # 镜像名字
28   version: "tensorflow-1.14.0-gpu-py3"
29   command: >
30     python3 ${MODEL_DIR}/tf-model/convolutional_network.py --
trainImagesFile ${DATA_DIR}/train-images-idx3-ubyte.gz
31     --trainLabelsFile ${DATA_DIR}/train-labels-idx1-ubyte.gz --
testImagesFile ${DATA_DIR}/t10k-images-idx3-ubyte.gz
32     --testLabelsFile ${DATA_DIR}/t10k-labels-idx1-ubyte.gz --
learningRate 0.001
33     --trainingIters 2000

```

4.4 CLI文件安装

1. 准备训练的CLI二进制文件

将CLI文件夹拷贝至Prophecis的文件目录下。

2. 配置环境变量

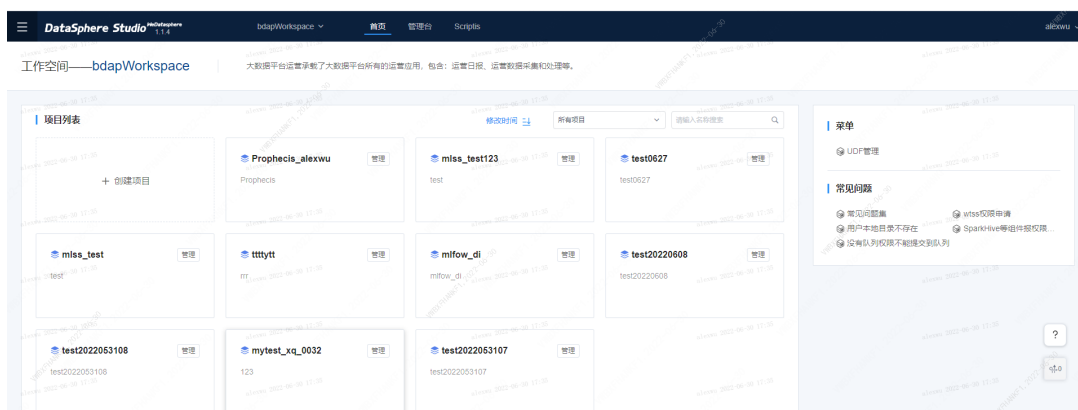
```
1 export DLAAS_URL=http://127.0.0.1:30959
2 export CLI_CMD=/data/Install/ProphecisInstall
3 export MLSS_AUTH_USER=<用户名>
4 export MLSS_AUTH_PASSWD=<base64编码后的用户密码>
5 export MLSS_AUTH_TYPE=UM
```

3. 验证安装成功

执行\$CLI_CMD list，若成功获取任务列表，代表安装成功。

5. DataSphere Studio使用Prophecis Node

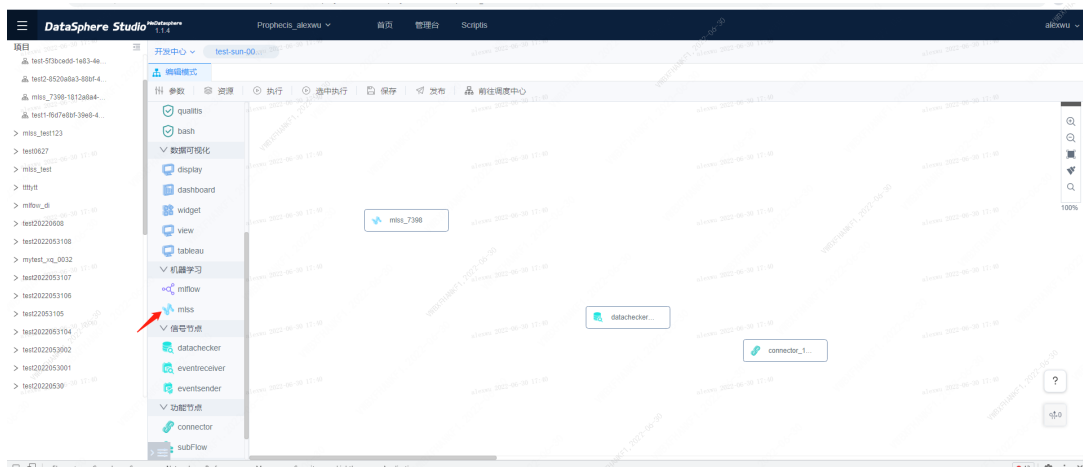
一、登录DSS-新建项目



二、添加工作流



三、使用mlss工作流节点



- (1) 双击可进入Prophecis node节点内部执行界面
- (2) 点击运行按钮，也可在dss执行mlss工作流