**如何使用airscheduler**

url: http://192.168.9.64:30887/api/v1/airscheduler/task

# 快速启动任务或服务

**简单示例：**

# coding: utf-8

import json

import requests

input\_data = {

'name': ...,

...

'start\_now': True

----------------------------------------------------------------------------- # **任务基本信息**

'running\_config': json.dumps(running\_config),

------------------- # **任务配置信息**

‘resource\_info‘: json.dumps(resource\_info),

----------------------- # **任务资源需求信息**

}

return\_data = requests.post(' http://192.168.9.64:30887/api/v1/airscheduler/task ', json=input\_data, headers={'token': token})

**完整示例：**

{

'name': 'test\_task',

'usage': '用于进行任务调度流程测试',

'source\_service\_id': 0,

'working\_type': 0,

'namespace': 'airtest',

'image\_repo\_tag': 'www.registry.cyber.ai/airevaluation/onnx/cc849cf470:v1.9',

'start\_now': True,

'running\_config': json.dumps({

# 应用后台生成

'volumes': [

{

'path': '/mnt/mfs/data', # 宿主机路径

'mount\_path': '/mnt/mfs/data', # 容器内挂载路径

'mount\_name': '', # 不传随机生成

'read\_only': False # 不传默认False

},

{

'is\_nfs': True, # 不传默认False

'server': '192.168.9.64', # 有NFS必须有Server

'path': '/mnt/mfs/data', # NFS路径

'mount\_path': '/mnt/mfs/data', # 容器内挂载路径

'mount\_name': '', # 不传随机生成

'read\_only': False # 不传默认False

},

],

# pvc依据需要追加

'pvc': [{

'pvc\_requests': 100 << 20, # 单位字节

'pvc\_limits': 1 << 30, # 单位字节

'mount\_path': '/data' # 可以不给，不给的话挂载路径与pvc路径一致

}],

# 端口映射依据需要添加

'port': [

{

'protocol': 'TCP', # 不给的话默认tcp

'node\_port': 25486, # 不给的话，随机分配

'container\_port': 5000 # 必须给出容器内需要映射的端口号

}

],

# 环境变量依据需要添加

'env': {

'locale': 'utf-8'

},

'timeout': 6 \* 3600, # 单位s，不传表示不限制运行时间

'command': ‘python3 /workspace/inference.py ["/mnt/mfs/data/Airplane/2.tiff"] ["/mnt/mfs/data/Airplane/2.test.xml"]', # 用户输入以及应用后台补全

}),

'resource\_info': json.dumps({

'schedule\_types': ["Kubernetes"],

'cpu\_count': 8,

'mem\_size': int(4 \* 1024 \*\* 3),

‘shm\_size’: 2 << 30,

'gpu\_dict': {'GeForce RTX 2080 Ti': 1}

})

}

# 输入数据含义

## 任务基本信息 — input\_data (dict)

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 描述 | 默认值 |
| name | string | 该任务或服务的名称（最好可以用于辨别是哪个功能以及哪个任务） | 无 |
| usage | string | 说明该任务申请资源的用途 (建议中文，用户看到能明确哪个功能哪个任务用掉了资源) | 无 |
| source\_service\_id | integer | 源服务端存储id | 无 |
| working\_type | integer | 0: 任务 1: 服务 | 无 |
| namespace | string | 命名空间 | 无 |
| image\_repo\_tag | string | 镜像REPO:TAG | 无 |
| distributed | boolean | 是否为分布式 | False |
| start\_now | boolean | 是否创建后立即启动 | False |

## 任务配置信息 — running\_config (dict)

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 描述 | 默认值 |
| volumes | list | 容器目录挂载映射 | 无 |
| pvc | list | 指定容器内挂载pvc，按需提供，默认不提供 | 无 |
| port | list | 当前服务或者任务希望映射的端口号信息 | 无 |
| env | dict | 环境变量 | 无 |
| command | string | 任务运行命令，直接给字符串命令 | 无 |
| work\_dir | string | 容器内代码工作路径，依据实际情况给定，默认不使用镜像默认工作路径 | 无 |
| yaml\_data | Json.dumps(dict) | **特殊字段**，支持直接从yaml文件启动任务。普通任务忽略该字段 |  |

### volumes挂载卷字段说明

volumes字段存在于running\_config当中，属于一个列表，每一个元素是一个字典，说明了当前容器的一个挂载卷信息。具体的挂载卷信息如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 描述 | 默认值 |
| is\_nfs | bool | 当前挂载卷是否为NFS类型挂载 | False |
| server | str | NFS挂载卷所在的服务器IP | 无 |
| path | str | 宿主机路径 | 无 |
| mount\_path | str | 容器内挂载路径 | 无 |
| mount\_name | str | 挂载卷名称，不提供系统可以自动生成 | 无，推荐不提供 |
| read\_only | bool | 是否以只读的形式挂载 | False |

### port挂载卷字段说明

port字段存在于running\_config当中，属于一个列表，每一个元素是一个字典，说明了当前容器的一个端口映射信息。具体的端口映射信息如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 描述 | 默认值 |
| protocol | str | tcp/TCP/udp/UDP，端口映射协议 | tcp |
| node\_port | int | 需要映射到宿主机的端口号，如果不提供则随机分配 | 无 |
| container\_port | int | 容器内需要映射的端口号 | 无 |

### pvc挂载卷字段说明

pvc字段存在于running\_config当中，属于一个列表，每一个元素是一个字典，说明了当前容器的一个PVC挂载信息。具体的PVC信息如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 描述 | 默认值 |
| pvc\_requests | int | 申请的PVC大小，单位字节 | 无 |
| pvc\_limits | int | 申请的PVC最大大小，单位字节 | 无 |
| mount\_path | str | 当前pv挂载到容器内的路径 | 无 |

## 任务资源需求信息 — resource\_info (dict)

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 描述 | 默认值 |
| cpu\_count | integer | 任务或服务需要的cpu个数（单位：个） | 无 |
| mem\_size | integer | 任务或服务需要的内存大小（单位：字节） | 无 |
| gpu\_dict | Json.dumps(dict) | 任务或服务需要的GPU类型与数量，不提供则不会分配GPU | 无 |
| shm\_size | integer | 任务或服务需要的共享内存大小（单位：字节） | 64 << 20；  默认大小约为64MB |
| scheduler\_types | list | 任务或服务的调度类型 | Any（任意） |

* 附：目前可调用的GPU型号

GeForce GTX 1080; GeForce 1080 Ti; GeForce RTX 2080 Ti 等

# 功能函数

**\* 库位置：lib.service\_util**

## 创建新任务或新服务

### 请求示例

# coding: utf-8

from lib.service\_util import create\_object\_in\_airscheduler

return\_data = create\_object\_in\_scheduler ( base\_url, input\_data, token )

### 功能描述

* 创建新任务，如果start\_now为True：快速启动任务。反之，挂起任务，等待功能（d）启动该任务。

### 输入参数

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 描述 |
| base\_url | str | 调度服务IP，比如“192.168.9.64:30887” |
| input\_data | dict | 任务创建信息，具体参见第二章节 |
| token | str | 用户Token |

### 返回状态

0：成功，返回已创建任务的详细信息，添加专属字段id

990003：输入数据错误

111003：任务启动失败

## 编辑任务或服务

请求：

# coding: utf-8

from lib.service\_util import modify\_object\_in\_airscheduler

return\_data = modify\_object\_in\_scheduler ( base\_url, input\_data, token )

* input\_data 中必须额外包含已创建任务的id

功能描述：

* 修改某个已创建的任务的详细信息，如果该任务已运行，会强制停止并删除已运行容器，并将修改后的任务挂起，等待功能（d）启动该任务。
* 如果该任务未运行，修改该任务的详细信息并挂起，等待功能（d）启动该任务。
* token 来源于用户登录，默认采用 账号：qiyan1 密码：123456 获取token

返回状态：

0：成功

990003：输入数据错误

111001：该任务不存在

111002：该任务正在运行

## 彻底删除任务或服务

请求：

# coding: utf-8

from lib.service\_util import delete\_object\_in\_airscheduler

return\_data = delete\_object\_in\_scheduler ( base\_url, task\_id, token )

功能描述:

* 彻底删除某个任务以及其对应的数据库
* token 来源于用户登录

返回状态：

0：成功

990003：输入数据错误

## 启动任务或服务

请求：

# coding: utf-8

from lib.service\_util import start\_object\_in\_airscheduler

return\_data = start\_object\_in\_scheduler ( base\_url, task\_id, token )

功能描述：

* 启动某一个挂起的任务。
* token 来源于用户登录

返回状态：

0：成功

990003：输入数据错误

111001：该任务不存在

111003：任务启动出错

111005：重复启动该任务

## 获取任务或服务的详细信息

请求：

# coding: utf-8

from lib.service\_util import observe\_object\_in\_airscheduler

return\_data = observe\_object\_in\_scheduler ( base\_url, task\_id, token )

功能描述：

* 查询某一个任务的详细信息。
* token 来源于用户登录

返回状态：

0：成功

990003：输入数据错误

111001：该任务不存在

## 筛选任务或服务

请求：

# coding: utf-8

from lib.service\_util import filter\_object\_in\_airscheduler

return\_data = filter\_object\_in\_scheduler ( base\_url, col\_name, col\_values, token )

功能描述：

* 根据特定条件批量筛选任务，得到多个任务的详细信息。
* token 来源于用户登录

返回状态：

0：成功

990003：输入数据错误

## 停止任务或服务

请求：

# coding: utf-8

from lib.service\_util import stop\_object\_in\_airscheduler

return\_data = stop\_object\_in\_scheduler ( base\_url, task\_id, token )

功能描述：

* 删除某一个已启动的任务。
* token 来源于用户登录，默认采用 账号：qiyan1 密码：123456 获取token

返回状态：

0：成功

111001：该任务不存在

111006：停止任务错误

## 任务静态日志查询

请求：

# coding: utf-8

from lib.service\_util import object\_logs\_in\_airscheduler

return\_data = object\_logs\_in\_scheduler ( base\_url, task\_id, start\_line, end\_line, token)

获取行数的规则:

start\_line: None(等效0), end\_line: None(等效-1), 获取所有日志

start\_line: 0, end\_line: None(等效-1), 获取所有日志

start\_line: None(等效0), end\_line: 23, 获取前23行日志

start\_line: 2, end\_line: 23, 获取2-23行的日志

start\_line: 2, end\_line: -2, 获取第2行到倒数第二行之间的日志

start\_line: -200, end\_line: 67, 获取倒数第200行到第67行之间的日志

start\_line: -200, end\_line: -1, 获取倒数200行的日志

start\_line: -200, end\_line: -1000, 不合法范围

start\_line: 1000, end\_line: 200, 不合法范围

如果范围明确不合法，则返回错误

如果范围合法性与日志长度有关，且范围实际不合法，存在交叠，返回空内容

返回结果包含start\_line和end\_line

功能描述：

* 读取并返回某任务所创建容器的日志信息。
* token 来源于用户登录

返回状态：

0：成功，包含当前日志的总行数，以及日志内容

111001：输入数据错误

111007：日志索引出错

返回数据：

{ code : 0, message: success, logs : ‘当前任务截取的日志’, ‘total\_lines’: 当前任务日志的总行数}