

AI 科学计算服务

命令行工具 CLI

文档版本 01
发布日期 2025-07-09



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目 录

1 什么是 AI 科学计算 ai4s-toolkit..... 1

2 获取并使用命令行工具 ai4s-toolkit..... 3

2.1 步骤 1：获取认证信息..... 3

2.2 步骤 2：下载并安装命令行工具 ai4s-toolkit.....4

2.3 步骤 3：初始化配置.....7

2.4 步骤 4：查看与执行操作命令..... 10

3 系统设置命令..... 12

3.1 获取系统标签..... 12

3.2 添加系统标签..... 12

3.3 删除系统标签..... 13

3.4 获取系统资源..... 13

3.5 修改性能加速资源..... 14

3.6 获取商标和标题设置..... 14

3.7 修改商标和标题配置..... 15

4 空间管理命令..... 16

4.1 获取空间..... 16

4.2 切换空间..... 18

4.3 创建空间..... 19

4.4 更新空间..... 19

4.5 删除空间..... 20

4.6 转移空间..... 20

4.7 添加空间成员角色..... 21

4.8 修改空间成员角色..... 21

4.9 移除空间成员的角色..... 22

5 数据管理命令..... 23

5.1 获取数据作业..... 23

5.2 删除数据作业..... 25

5.3 重试数据作业..... 25

5.4 取消数据作业..... 26

6 镜像管理命令..... 27

6.1 标记镜像..... 27

6.2 上传镜像.....	28
6.3 下载镜像.....	29
6.4 查询镜像.....	29
6.5 导入镜像.....	30
6.6 更新镜像.....	31
6.7 删除镜像标签.....	31
7 应用管理命令.....	33
7.1 应用配置文件说明.....	33
7.2 创建应用.....	34
7.3 修改应用.....	35
7.4 查询应用.....	36
7.5 删除应用.....	39
7.6 导入应用.....	39
8 AI 科学计算服务流程管理命令.....	41
8.1 流程配置文件说明.....	41
8.2 创建流程.....	42
8.3 修改流程.....	43
8.4 查询流程.....	44
8.5 删除流程.....	47
8.6 导入流程.....	47
9 AI 科学计算服务作业管理命令.....	49
9.1 作业配置文件说明.....	49
9.2 启动作业.....	50
9.3 查询作业详情.....	53
9.4 删除指定作业.....	58
9.5 重试作业.....	58
9.6 取消作业.....	59
9.7 导出作业.....	59
9.8 获取任务实例.....	60
10 notebook 命令.....	62
10.1 创建 notebook.....	62
10.2 获取 notebook 信息.....	63
10.3 删除 notebook.....	65
10.4 编辑 notebook.....	65
10.5 启动 notebook.....	66
10.6 停止 notebook.....	66

1 什么是 AI 科学计算 ai4s-toolkit

AI科学计算服务命令行工具（ai4s-toolkit）为AI科学计算服务各功能组件提供命令行管理工具。借助此工具，可以辅助您对AI科学计算平台空间中数据、应用、流程和作业资源进行管理和使用。

ai4s-toolkit具备灵活性高、扩展性强的特点，具体体现在：

1. 单一可执行文件，方便复制与安装。
2. 支持多操作系统，包括Windows、Linux（支持CentOS 7及以上版本和openEuler 20.03 LTS - openEuler 20.09 LTS版本，本文中提到的所有Linux操作系统均需要以此处描述的版本为准。）和macOS，满足不同开发者需求。

产品功能

ai4s-toolkit提供了以下功能：

- 系统设置
您可以使用该工具获取系统标签和系统资源，也可以获取和修改供应商logo、名称设置。
- 空间管理
您可以使用该工具获取、切换、创建、更新、删除和转移空间，也可以添加、修改、移除空间成员角色。
- 数据管理
您可以使用该工具对数据作业执行获取、删除、重试、取消操作。
- 应用管理
应用是生物信息学软件和运行该软件所依赖的运行环境的镜像封装。
您可以使用该工具创建应用，并进行修改、删除、查询、导入操作。
- 流程管理
流程包含分析过程中所需应用的执行信息和数据的输入、输出等参数定义，流程至少由一个应用组成。
您可以使用该工具创建流程，并进行修改、删除、查询、导入操作。对于由多个应用构成的流程，可通过设置不同应用的输入、输出关系搭建为流程。
- 作业管理
您可以使用该工具启动分析作业，并进行查询、删除、重试、取消操作。

- notebook

您可以使用该工具创建notebook，并进行查询、编辑、删除和启停等操作。

2 获取并使用命令行工具 ai4s-toolkit

2.1 步骤 1：获取认证信息

获取 AK/SK

AK/SK（Access Key ID/Secret Access Key）即访问密钥，包含访问密钥ID（AK）和秘密访问密钥（SK）两部分，华为云通过AK识别用户的身份，通过SK对请求数据进行签名验证，用于确保请求的机密性、完整性和请求者身份的正确性。

1. 登录[华为云管理控制台](#)，鼠标指向页面右上角的用户名，在下拉列表中单击“我的凭证”。

图 2-1 我的凭证入口



2. 在“我的凭证”页面中选择“访问密钥”页签。单击“新增访问密钥”，按操作指引获取认证账号的AK/SK，请妥善保管AK/SK信息。

图 2-2 访问密钥



📖 说明

- 每个用户仅允许新增两个访问密钥。
- 为保证访问密钥的安全，访问密钥仅在初次生成时自动下载，后续不可再次通过管理控制台页面获取。请在生成后妥善保管。

获取区域名称

区域是一个地理区域的概念，由于带宽原因，会在多个地区建立数据中心，提供服务。

AI科学计算服务部署在华北-北京四，区域名称为cn-north-4；部署在华东-上海一，区域名称为cn-east-3。

获取项目 ID

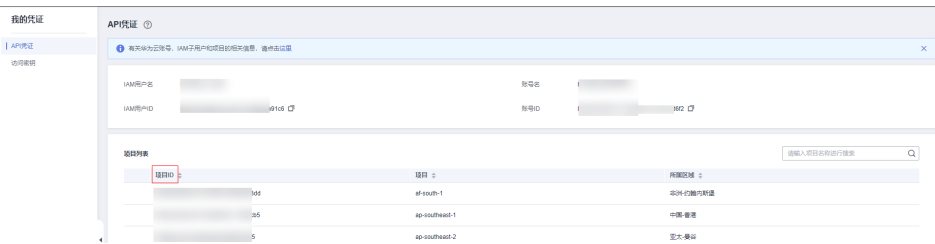
1. 登录[管理控制台](#)。
2. 鼠标移动到右上角的用户名上，在下拉列表中选择“我的凭证”。

图 2-3 我的凭证



3. 在“我的凭证 > API凭证”页面，可以查看用户名、账号名，在项目列表中查看项目。

图 2-4 查看项目 ID



多项目时，展开“所属区域”，从“项目ID”列获取子项目ID。

2.2 步骤 2：下载并安装命令行工具 ai4s-toolkit

下载命令行工具 ai4s-toolkit

针对不同操作系统，ai4s-toolkit下载地址如下所示。

表 2-1 下载列表

支持平台	下载地址
Windows 64位	ai4s-windows-x86_64.zip 、 ai4s-windows-x86_64.zip.sha256 ，老版本的软件（ health-windows-x86_64.zip 、 health-windows-x86_64.zip.sha256 ）
Linux ARM 64位	ai4s-linux-aarch64.tar 、 ai4s-linux-aarch64.tar.sha256 ，老版本的软件（ health-linux-aarch64.tar 、 health-linux-aarch64.tar.sha256 ）
Linux AMD 64位	ai4s-linux-x86_64.tar 、 ai4s-linux-x86_64.tar.sha256 ，老版本的软件（ health-linux-x86_64.tar 、 health-linux-x86_64.tar.sha256 ）
macOS	ai4s-macOS-x86_64.tar 、 ai4s-macOS-x86_64.tar.sha256 ，老版本的软件（ health-macOS-x86_64.tar 、 health-macOS-x86_64.tar.sha256 ）

文件操作工具详见[OBS工具](#)，可完成查看、上传、下载、复制、导入、引入、创建和删除桶中的数据。

镜像操作详见[image-syncer工具](#)，可完成对应容器镜像的迁移。

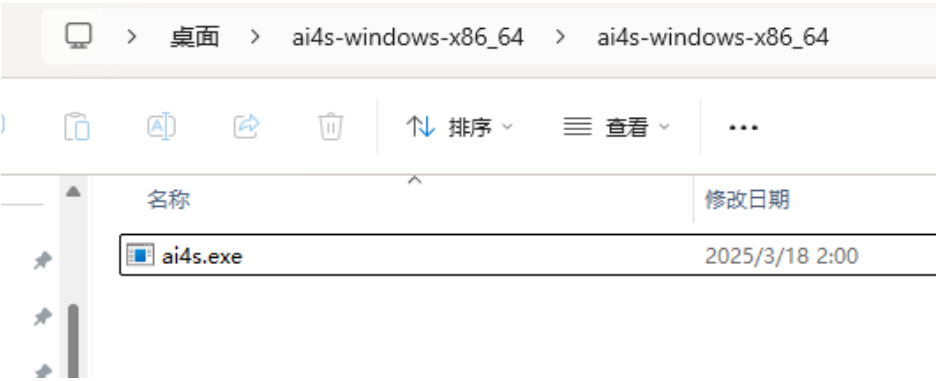
说明

- 本页面命令行工具下载后，在使用时，需用到您注册华为账号并开通华为云时提供的用户名等信息，用于登录并操作AI科学计算平台的空间、应用、流程等资产。这些信息的处理将遵循您已接收的《[华为云用户协议](#)》及《[隐私政策声明](#)》约束。
- 下载地址中带有sha256后缀的链接，指的是对应软件包的校验文件。例如：Windows x64版本的下载链接是ai4s-windows-x86_64，它的校验文件下载链接则是ai4s-windows-x86_64.zip.sha256。

安装 ai4s-toolkit

- 本示例中以Windows系统为例，介绍安装命令行工具的方法。
 - a. 获取Windows版本的命令行工具，得到ai4s.exe文件，ai4s文件无需安装，放置在任一文件夹中即可。

图 2-5 下载命令行工具



- b. 使用win键+R，输入**cmd**打开windows的cmd窗口。进入工具所在的目录，输入**ai4s**命令，即可使用。

如果cmd窗口显示目录不是ai4s文件所在目录，请使用**cd**命令切换路径。例如，切换至D盘：

```
cd /d d:
```

图 2-6 客户端

```
C:\Users\ \Desktop\ai4s-windows-x86_64\ai4s-windows-x86_64>ai4s
A simple command client for ai4s.

Usage:
  ai4s [flags]
  ai4s [command]

Available Commands:
  add          add a member...
  cancel       cancel or forcibly terminate the job
  clear        delete the checkpoint dir
  completion   Generate the autocompletion script for the specified shell
  config       configure your config file ...
  create       create a app,workflow,job...
  delete       delete a app, workflow, job
  edit         edit a app,workflow...
  export       export job...
  get          get project, app, workflow
  help        Help about any command
  import       import app, workflow...
  retry        run the job or notebook again
  start        start notebook ...
  stop         stop notebook ...
  switch       switch project ...
  transfer     transfer project...
  version      show cli version

Flags:
  -h, --help  help for ai4s

Use "ai4s [command] --help" for more information about a command.
```

- 本示例中以Linux系统为例，介绍安装命令行工具的方法。
 - a. 获取Linux版本的命令行工具，得到ai4s文件，ai4s文件无需安装，放置在任一文件夹中即可。
 - b. 假设ai4s存放在/home/user-name/test/client目录，请使用**cd**命令进入ai4s所在目录。

```
cd /home/user-name/test/client/
```

图 2-7 客户端

```
[root@... client]# ./ai4s
A simple command client for ai4s.

Usage:
  ai4s [flags]
  ai4s [command]

Available Commands:
  add          add a member...
  cancel       cancel or forcibly terminate the job
  clear        delete the checkpoint dir
  completion   Generate the autocompletion script for the specified shell
  config       configure your config file ...
  create       create a app,workflow,job...
  delete       delete a app, workflow, job
  edit         edit a app,workflow...
  export       export job...
  get          get project, app, workflow
  help         Help about any command
  import       import app, workflow...
  retry        run the job or notebook again
  start        start notebook ...
  stop         stop notebook ...
  switch       switch project ...
  transfer     transfer project...
  version      show cli version

Flags:
  -h, --help  help for ai4s

Use "ai4s [command] --help" for more information about a command.
```

说明

- 使用Linux版本命令行工具时，您需要在本地搭建Linux环境，并将下载的ai4s文件放至所需的目录下。macOS执行命令和linux一致，
- 如果当前目录为ai4s所在目录，可以使用./ai4s命令使用命令行工具。
 - 如果当前目录不是ai4s所在目录，需要使用绝对路径。如当前目录为/opt，假设ai4s存放在/root/ai4s-toolkit/下，需要指定/root/ai4s-toolkit/ai4s路径进行使用。
 - 如果无法运行，提示Permission denied，使用**chmod 755 ai4s**命令设置执行权限。

2.3 步骤 3：初始化配置

在使用命令行工具前，需要初始化配置信息，通过**config**命令对ai4s-toolkit进行初始化配置。本节以Windows为例介绍配置过程，Linux和macOS环境配置过程相同。

命令结构

执行**ai4s config add**命令进行初始化配置。

```
ai4s config add [flags]
```

表 2-2 参数说明

参数	简写	是否必选	说明
--domain-name	-d	是	与管理员（购买平台的账户）的账号名一致。

参数	简写	是否必选	说明
--user-name	-u	是	子用户的用户名。 管理员（购买平台的账户）登录时，user-name和domain-name一致。
--password	-w	是	密码。
--ak	-a	是	AK(Access Key ID)：访问密钥ID。
--sk	-s	是	SK(Secret Access Key)：与访问密钥ID结合使用的密钥。
--region	-r	是	服务区域名称。依据购买服务所在的区域进行选择，可选cn-north-4、cn-east-3、cn-south-1。 
--platform-id	-i	是	项目ID，获取方法请参见 获取认证信息 。
--iam-endpoint	-m	否	IAM终端节点名称，请在 地区与终端节点 中获取。
--ai4s-endpoint	-e	否	AI4S终端节点名称，请在 地区与终端节点 中获取。
--log-path	-l	否	日志路径，不填写时默认为命令行工具当前路径下ai4s.log文件。 路径设置格式： - Windows系统为“路径\文件名”。 - Linux系统格式为“路径/文件名”。
--http-proxy	-p	否	HTTP代理配置，格式为“http://username:password@your-proxy:your-port”。

命令示例

- 初始化配置
ai4s config add -d xxx -u xxx -w xxx -i xxx -r cn-north-4 -a xxx -s xxx -D https://xxx -q xxx -f -m xxx -e xxx -t xxx
执行成功返回结果如下
add ak successfully!
add sk successfully!
add region successfully!
add platform-id successfully!
add user-name successfully!
add password successfully!
add domain-name successfully!
add iam-endpoint successfully!
add ai4s-endpoint successfully!

- 初始化配置并设置日志存储路径

```
ai4s config add -d xxx -u xxx -w xxx -i xxx -r cn-north-4 -a xxx -s xxx -D https://xxx -q xxx --log-path D:\log\ai4s.log -f -m xxx -e xxx -t xxx
# 执行成功返回结果如下
add ak successfully!
add sk successfully!
add region successfully!
add platform-id successfully!
add user-name successfully!
add password successfully!
add domain-name successfully!
add log-path successfully!
add iam-endpoint successfully!
add ai4s-endpoint successfully!
add swr-endpoint successfully!
```

- 初始化配置并设置日志存储路径

```
ai4s config add -d xxx -u xxx -w xxx -i xxx -r cn-north-4 -a xxx -s xxx -D https://xxx -q xxx --log-path D:\log\ai4s.log
# 执行成功返回结果如下
add ak successfully!
add sk successfully!
add region successfully!
add platform-id successfully!
add user-name successfully!
add password successfully!
add domain-name successfully!
add log-path successfully!
```

- 清空配置请执行 **ai4s config clear** 命令

- 在 Notebook 中使用命令行工具

在 AI 科学计算平台开发环境 Notebook 中使用命令行工具时，请依据以下步骤配置代理。

- a. 打开 Notebook，并选择 Terminal，打开 Notebook 的命令行界面。
- b. 执行以下命令下载命令行工具，并获取配置 Notebook 代理所需的域名和端口信息。

示例中下载的版本为 Linux ARM 64 位。

```
wget https://eihealth-cli.obs.cn-north-4.myhuaweicloud.com/current/ai4s-linux-aarch64.tar
```

```
sh-4.3$ wget https://eihealth-cli.obs.cn-north-4.myhuaweicloud.com/current/health-linux-arm64.tar
--2021-06-17 14:16:03-- https://eihealth-cli.obs.cn-north-4.myhuaweicloud.com/current/health-linux-arm64.tar
Resolving proxy-notebook.modelarts-dev-proxy.com (proxy-notebook.modelarts-dev-proxy.com)... 192.168.1.1
Connecting to proxy-notebook.modelarts-dev-proxy.com (proxy-notebook.modelarts-dev-proxy.com)|192.168.1.1:80... connected.
Proxy request sent, awaiting response... 200 OK
Length: 12001280 (11M) [application/x-tar]
Saving to: 'health-linux-arm64.tar'

health-linux-arm64.tar  100%[=====] 11.45M  --.-KB/s  in 0.1s

2021-06-17 14:16:03 (109 MB/s) - 'health-linux-arm64.tar' saved [12001280/12001280]
```

- c. 初始化配置时，添加 **--http-proxy** 命令，用于配置代理。

```
ai4s config add --http-proxy http://域名:端口号
```

📖 说明

- 执行以上命令，会在系统所在的用户目录下自动生成“.ai4s”文件夹，文件夹中包含 config.ini 配置文件，用于存储任务执行所涉及到的配置，如密钥、区域、当前空间等信息。
- 生成的配置文件不建议直接修改，如需改动请使用命令行工具修改。
- 配置文件中保存有用户的 AK、SK 信息，为了避免密钥泄露，会对文件中的 SK 进行加密以保护密钥安全。
- 初始化配置时，如果命令同时填写了 AK/SK 和密码，默认 AK/SK 方式登录。
- 初始化配置命令会在 history 中暴露 ak、sk，建议使用 set +o history 命令关闭 history 再执行。执行后可使用 set -o history 命令恢复。

清理命令记录

为防止配置文件中的敏感信息泄露，建议使用**ai4s config clear**命令定时清除本地配置文件。

对于执行的历史命令，可通过以下方法清除。

- Linux：执行**history -c**命令清除历史记录，重新登录命令行工具后，记录可恢复。执行**rm -f \$HOME/.bash_history**命令，可删除记录文件，清空历史。
- macOS系统：执行**ps -p \$\$**命令检查shell类型。
 - 类型为bash：操作与Linux系统相同。
 - 类型为zsh：执行**history -p**命令清除历史记录，重新登录命令行工具后，记录可恢复。执行**rm -rf ~/.zsh_history**命令，退出终端后再次执行**history -p**命令可删除记录文件，清空历史。
- windows系统：对于当前执行的命令，可通过关闭cmd窗口实现历史命令的清理。

2.4 步骤 4：查看与执行操作命令

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。配置信息导入后，即可查询命令行工具支持的操作，并执行相关命令，使用AI科学计算平台。

详细的操作命令请参见[系统设置命令](#)、[空间管理命令](#)、[数据管理命令](#)、[镜像管理命令](#)、[应用管理命令](#)、[AI科学计算服务流程管理命令](#)、[AI科学计算服务作业管理命令](#)或[notebook命令](#)章节。

- 查询操作命令列表。
执行**ai4s --help**查询支持的操作命令。Linux系统下，需添加./指定当前路径。

```
Use "ai4s [command] --help" for more information about a command.

C:\>ai4s -h
A simple command client for ai4s.

Usage:
  ai4s [flags]
  ai4s [command]

Available Commands:
  add          add a member...
  cancel       cancel or forcibly terminate the job
  clear        delete the checkpoint dir
  completion   Generate the autocompletion script for the specified shell
  config       configure your config file ...
  create       create a app,workflow,job...
  delete       delete a app, workflow, job
  edit         edit a app,workflow...
  export       export job...
  get          get project, app, workflow
  help         Help about any command
  import       import app, workflow...
  retry        run the job or notebook again
  start        start notebook ...
  stop         stop notebook ...
  switch       switch project ...
  transfer     transfer project...
  version      show cli version

Flags:
  -h, --help  help for ai4s

Use "ai4s [command] --help" for more information about a command.
```

- 执行操作命令，获取空间信息。
执行ai4s get project命令查询当前账号下所拥有的空间和空间信息。Linux系统下，需添加./指定当前路径。

Project Name	Project Id	Owner	Role	Status	Size	Created	Modified
test222	2	t	Developer	ACTIVE	119.47MB	2024-11-20 16:05:33	2024-11-25 16:05:22
uyiulul	5	e	Owner	ACTIVE	797.89GB	2024-09-30 10:45:36	2024-09-30 10:45:37
test	0	t	Administrator	ACTIVE	18.63MB	2024-06-11 19:11:10	2024-06-11 19:11:12
test-0000	f	t	Uploader	ACTIVE	117.74GB	2024-05-25 11:21:32	2024-11-25 16:18:03
yyf-test	2	e	Owner	ACTIVE	63.94GB	2023-12-22 15:57:00	2023-12-22 15:57:02
test-project-new-permissions-700	3	w	Administrator	ACTIVE	291.51GB	2023-11-09 21:07:38	2024-07-05 16:23:05
test-bjl	1	e	Owner	ACTIVE	713.91GB	2023-06-03 15:30:32	2024-05-14 17:23:25
performance-test-get	0	e	Owner	ACTIVE	65.20GB	2023-04-13 00:22:20	2024-01-04 09:42:24
performance-test-create	6	e	Owner	ACTIVE	253.33GB	2023-04-13 00:22:14	2023-11-16 15:16:37
performe-test	a	t	Administrator	ACTIVE	769.97GB	2023-02-10 10:05:09	2023-02-15 10:42:55
test-new-project	d	w	Uploader	ACTIVE	256.18GB	2023-01-09 19:18:03	2024-11-14 16:08:00
eihealth_api_project_cpts	b	e	Administrator	ACTIVE	241.93GB	2022-11-08 17:11:59	2022-11-08 17:12:06
eihealth_api_explorer_project_002	0	e	Owner	ACTIVE	120.80MB	2022-09-13 11:04:51	2023-09-06 16:38:11
eihealth_api_explorer_project_001	b	e	Owner	ACTIVE	62.10GB	2022-09-13 11:04:37	2022-09-13 11:04:39
guancha	5	t	Administrator	ACTIVE	823.81GB	2022-07-29 15:06:12	2023-02-27 11:25:34
test-lugw	d	t	Administrator	ACTIVE	14.52GB	2022-07-27 17:20:37	2024-02-26 16:27:23
new-test-console	c	w	Administrator	ACTIVE	0.00B	2022-07-20 09:10:12	2022-10-27 14:13:33
cli-001	4	e	Viewer	ACTIVE	19.34GB	2022-07-19 09:51:29	2023-08-09 11:39:00
gem-assets	b	w	Viewer	ACTIVE	769.25GB	2022-07-19 09:34:53	2024-11-26 14:58:14
eihealth_api_project_001	7	e	Administrator	ACTIVE	7.01TB	2022-07-19 09:11:18	2025-01-22 07:32:25

说明

- 使用数据、应用、流程、作业命令时，需先使用switch命令进入待操作的空间中。使用逻辑与AI科学计算服务一致，进入空间，再对空间内的数据等内容进行操作。
- 命令行的参数缩写支持合并使用，例如，-r -s 可以写成-rs。

3 系统设置命令

3.1 获取系统标签

使用get命令获取系统标签库列表。

命令结构

```
ai4s get label [flags]
```

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux和macOS环境使用方法基本相同，可参考。

```
ai4s get label
# 执行成功返回结果如下
Id          Label          Creator  Created      Updated      Description
d242057-46af-11ed  drug-screening  lmx      2021-02-01 11:11:27  2021-02-01 11:11:27  药物筛选
8942046-46af-11ed  eiflow         hpw      2021-02-01 11:11:27  2021-02-01 11:11:27  流程编排
2642052-46af-11ed  project        lmx      2021-02-01 11:11:27  2021-02-01 11:11:27  project
manage
```

3.2 添加系统标签

添加系统标签库。

命令结构

```
ai4s create label [name] [flags]
```

表 3-1 参数说明

参数	简写	是否必选	说明
name	不涉及	是	标签名。
description	-d	否	标签描述。

命令示例

```
ai4s create label name -d xxx
{
  "id": "b84ff2ce-0e51-11ed-9bd3-fa163e504fdd"
}
create label name is successful
```

3.3 删除系统标签

删除或批量删除系统标签库。

命令结构

```
ai4s delete label [flags]
```

表 3-2 参数说明

参数	简写	是否必选	描述
list	-l	否	删除的标签id列表，json数组格式。

命令示例

```
ai4s delete label -l "[\"1001\", \"1002\"]"
delete label is successful
```

3.4 获取系统资源

使用get命令获取绑定的性能加速资源。

命令结构

```
ai4s get resource [flags]
```

表 3-3 参数说明

参数	简写	是否必选	说明
--type	-t	是	名称，支持performance，代表性能加速资源。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

- 获取性能加速资源列表
ai4s get resource -t performance
执行成功返回结果如下

ID	Name		Billing Mode	Used Capacity	Used Capacity	Percent
Specifications	Job Quota	Status	Purchased	Running Jobs		
01c24e2f-2038-4718-acb0-a0c388a2f93c			sfs-turbo-218766ac-5855-4ad4-			
a5fd-67130951afdd		XXX.storage.ioacc.spec.ioacc	xxx	0.00GB	0.00%	
100GB	1	ACTIVE / Schedulable	2023-01-19 21:57:51	0		

3.5 修改性能加速资源

修改性能加速资源节点。

命令结构

```
ai4s edit performance-resource <ID> [flags]
```

表 3-4 参数说明

参数	简写	是否必选	描述
--job-quota	-j	否	修改性能加速资源作业配额。
--schedulable	-s	否	设置性能加速资源是否可调度，取值：true、false。

命令示例

设置作业配额和可调度

```
ai4s edit performance-resource bea1ab31-7dfb-481e-bfe1-b149e0b4642e -s true -j 12
```

执行成功返回结果如下

```
edit performance-resource succeed!
```

3.6 获取商标和标题设置

使用get命令获取商标和标题配置，默认将供应商logo图片下载到本地路径。

命令结构

```
ai4s get vendor-config [flags]
# 其中vendor-config可简写为vc
```

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s get vendor-config
# 执行成功返回结果如下
the vendor name is xxxx, and download the logo picture to /root/demo/ successfully!
```

3.7 修改商标和标题配置

使用update或edit命令修改商标和标题配置。

命令结构

```
ai4s edit vendor-config [flags] 或者 ai4s update vendor-config [flags]
# 其中vendor-config可简写为vc
```

表 3-5 参数说明

参数	简写	是否必选	说明
--name	-n	否	供应商名称。
--logo	-l	是	供应商图标本地路径。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

- 设置供应商名称

```
ai4s edit vendor-config --name xxx1 --log /root/demo/logo.png
# 执行成功返回结果如下
set vendor name to xxx1 successfully!
```
- 设置供应商图标

```
ai4s edit vendor-config --log /root/demo/logo.png
# 执行成功返回结果如下
set vendor logo successfully!
```

4 空间管理命令

4.1 获取空间

使用`get`命令获取当前用户有权限访问的所有空间列表，以及空间信息，包含空间名称、ID、所有者、空间角色、存储大小、状态、创建时间、更新时间。

在使用该命令前，您需要通过AI科学计算平台创建空间。

命令结构

```
ai4s get project <project-name> [flags]
```

表 4-1 参数说明

参数	简写	是否必选	说明
project-name	无	否	空间名称，在AI科学计算平台创建空间时，由用户填写。 - 指定具体空间名称时，返回此空间的详情信息。 - 为空时，获取当前用户有权限访问的空间列表。
--current	-c	否	同时使用project-name和--current，查询当前所在的空间详情信息。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

- 不指定参数。
执行`ai4s get project`命令获取当前用户有权限访问的空间列表。

```
ai4s get project
# 执行成功返回结果如下
Project Name      Project Id      Owner      Role
```

Status	Size	Created	Modified	
maomaochong		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		ei_eihealth_lxxx_01
Administrator	ACTIVE	23.76GB	2025-04-24 10:35:29	2025-04-30 10:20:03
test-project-0421		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		ei_eihealth_lxxx_01
Administrator	ACTIVE	21.95MB	2025-04-21 11:24:06	2025-04-22 14:41:39
liuyuwei		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		test-liuyw
Administrator	ACTIVE	704.16MB	2025-04-07 04:20:58	2025-04-18 06:17:32
eihealth_api_project_002		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		
eihealth_api_001	Administrator	ACTIVE	37.52GB	2025-04-03 09:44:50
2025-04-03 10:05:15				
test-whh		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		test-www-0331
Owner	ACTIVE	308.52GB	2025-04-01 05:08:06	2025-04-01 05:08:07
test-hxy		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		ei_eihealth_lxxx_01
Uploader	ACTIVE	163.22GB	2025-03-31 12:01:17	2025-03-31 12:01:17
eihealth_api_project_006		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		
eihealth_api_002	Administrator	ACTIVE	11.10GB	2025-03-31 05:38:22
2025-04-03 09:39:37				
eihealth_api_project_004		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		
eihealth_api_002	Administrator	ACTIVE	37.52GB	2025-03-31 05:37:55
2025-04-03 09:39:26				
eihealth_api_project_001		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		
eihealth_api_001	Administrator	ACTIVE	62.13GB	2025-03-31 05:34:21
2025-04-03 09:39:44				
test-bjl		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		ei_eihealth_lxxx_01
Administrator	ACTIVE	352.66GB	2025-03-31 05:26:06	2025-03-31 05:26:07

- 指定“project-name”参数。
例如查看名为“test-bjl”空间信息，执行**ai4s get project test-bjl**命令。

```
ai4s get project test-bjl
# 执行成功返回结果如下
{
  "id": "XXXXX",
  "name": "test-bjl",
  "creator": "xxx",
  "role": "Administrator",
  "roles": [
    {
      "role_type": "Owner",
      "users": [
        {
          "id": "XXXXX",
          "name": "XXXX"
        }
      ]
    },
    {
      "role_type": "XXXX",
      "users": [
        {
          "id": "XXXXX",
          "name": "user007"
        },
        {
          "id": "XXXXX",
          "name": "cpp-test"
        },
        {
          "id": "XXXXX",
          "name": "test-bjl13"
        },
        {
          "id": "XXXXX",
          "name": "yjn-test"
        }
      ]
    }
  ],
  "size": XXXX,
  "status": "ACTIVE",
}
```

```
"description": "",
"create_time": "2025-03-31T02:26:06Z",
"update_time": "2025-03-31T02:26:07Z",
"is_core": false,
"storage_quota": -1,
"storage_tenant": "TERMINATE"
}
```

- 指定 “--current”。

先执行 **ai4s switch project test-bjl**，切换到 “test-bjl” 空间，再执行 **ai4s get project --current** 命令，查询当前所在空间的详细信息。

```
ai4s get project --current
# 执行成功返回结果如下
{
  "id": "XXXXX",
  "name": "test-bjl",
  "creator": "xxx",
  "role": "Administrator",
  "roles": [
    {
      "role_type": "Owner",
      "users": [
        {
          "id": "XXXXX",
          "name": "xxx"
        }
      ]
    },
    {
      "role_type": "Administrator",
      "users": [
        {
          "id": "XXXXX",
          "name": "user007"
        },
        {
          "id": "XXXXX",
          "name": "cpp-test"
        },
        {
          "id": "XXXXX",
          "name": "test-bjl13"
        }
      ]
    }
  ],
  "size": XXXXX,
  "status": "ACTIVE",
  "description": "",
  "create_time": "2025-03-31T02:26:06Z",
  "update_time": "2025-03-31T02:26:07Z",
  "is_core": false,
  "storage_quota": -1,
  "storage_tenant": "TERMINATE"
}
```

4.2 切换空间

使用 **switch** 命令切换到指定的空间中。

使用该命令前，您需要先创建空间，再使用 **switch** 命令进入该空间。

命令结构

```
ai4s switch project <project-name>
```

表 4-2 参数说明

参数	简写	是否必选	说明
project-name	无	是	需要进入的空间名称，名称在创建空间时由用户填写。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

执行如下命令，切换到名为“lmx-test-01”的空间。

```
ai4s switch project lmx-test-01
# 执行成功返回结果如下
switch to project lmx-test-01 successfully!
```

4.3 创建空间

使用create或submit命令创建空间。

命令结构

```
ai4s create project <project-name> [flags] 或 ai4s submit project <project-name> [flags]
```

表 4-3 参数说明

参数	简写	是否必选	说明
project-name	无	是	空间名称。
bucketname	-n	是	obs桶名。
isnewbucket	-c	否	是否新建新的obs桶。
--description	-d	否	空间描述。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

执行如下命令，创建空间。

```
ai4s create project demo-project -d description -c -n demo
# 执行成功返回结果如下
create project successfully! project id is xxx.
```

4.4 更新空间

使用update或edit命令更新空间描述信息、标签或状态。

命令结构

```
ai4s update project <project-name> [flags] 或 ai4s edit project <project-name> [flags]
```

表 4-4 参数说明

参数	简写	是否必选	说明
project-name	无	否	空间名称，不填时默认为当前所在空间。
--description	-d	否	空间描述。 只有空间所有者和管理员能修改。

命令示例

- 更新空间描述和标签信息。
ai4s update project demo-project --description description
执行成功返回结果如下
edit project successfully!

4.5 删除空间

使用**delete**命令删除空间。空间的所有者有权限执行该操作。

命令结构

```
ai4s delete project <project-name> [flags]
```

表 4-5 参数说明

参数	简写	是否必选	说明
project-name	无	否	空间名称，不填时默认为当前所在空间。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

执行以下命令，删除空间demo-project。

```
ai4s delete project demo-project
```

4.6 转移空间

使用**transfer**命令转移空间给空间admin成员。空间的所有者有权限执行该操作。

命令结构

```
ai4s transfer project <project-name> [flags]
```

表 4-6 参数说明

参数	简写	是否必选	说明
project-name	无	否	空间名称，不填时默认为当前所在空间。
--user	-u	是	转移用户名称。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s transfer project demo-project --user demo-user
# 执行成功返回结果如下
transfer project demo-project to demo-user successfully!
```

4.7 添加空间成员角色

使用add命令将用户添加至空间，并赋予成员角色。空间的所有者和管理员有权限执行该操作。

命令结构

```
ai4s add member <member-name> [flags]
```

表 4-7 参数说明

参数	简写	是否必选	说明
member-name	无	是	用户名称。
--project	-p	否	空间名称，不填时默认为当前所在空间。
--role	-r	是	用户空间角色，只支持Administrator，Developer，Uploader或者Viewer。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s add member demo-user --project demo-project --role Developer
# 执行成功返回结果如下add member demo-user to project demo-project successfully!
```

4.8 修改空间成员角色

使用update或edit命令修改空间成员角色。空间的所有者和管理员有权限执行该操作。

命令结构

```
ai4s update member <member-name> [flags] 或 ai4s edit member <member-name> [flags]
```

表 4-8 参数说明

参数	简写	是否必选	说明
member-name	无	是	用户名称。
--project	-p	否	空间名称，不填时默认为当前所在空间。
--role	-r	是	用户空间角色，只支持Administrator，Developer，Uploader或者Viewer。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s update member demo-user --project demo-project --role Viewer
# 执行成功返回结果如下
modify the role of member demo-user to Viewer successfully!
```

4.9 移除空间成员的角色

使用delete命令删除空间成员的角色，并将成员移除空间。空间的所有者和管理员有权限执行该操作。

命令结构

```
ai4s delete member <member-name> [flags]
```

表 4-9 参数说明

参数	简写	是否必选	说明
member-name	无	是	用户名称。
--project	-p	否	空间名称，不填时默认为当前所在空间。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s delete member demo-user --project demo-project
# 执行成功返回结果如下delete member demo-user of project demo-project successfully!
```

5 数据管理命令

5.1 获取数据作业

使用`get`命令获取数据作业列表、数据作业详情、数据作业执行日志或者下载数据作业执行日志。

命令结构

```
ai4s get data-job <data-job-id> [flags]
```

表 5-1 参数说明

参数	简写	是否必选	说明
data-job-id	无	否	数据作业id，不填写时未获取数据作业列表，填写时为获取数据作业详情。
--name	-n	否	数据作业名称，根据名称搜索作业，获取数据作业列表时生效。
--type	-t	否	数据作业类型，根据类型搜索作业，获取数据作业列表时生效。支持获取以下类型数据： - CLONE_DATA - DELETE_DATA - IMPORT_DATA - IMPORT_NETWORK_DATA
--end-time	-e	否	查询该时间之前创建的数据作业，获取数据作业列表时生效。 时间类型为时间戳。
--status	-s	否	数据作业状态，根据作业状态筛选作业，获取数据作业列表时生效。

参数	简写	是否必选	说明
--begin-time	-b	否	查询该时间之后创建的数据作业，获取数据作业列表时生效。 时间类型为时间戳。
--creator	-c	否	创建者名称，根据创建者筛选作业，获取数据作业列表时生效。
--limit	-l	否	限制量，单次查询总量，必须由数字组成，默认为10，取值范围[0,1000]，获取数据作业列表时生效。
--offset	-o	否	偏移量，查询起始偏移，必须由数字组成，默认为0，取值范围[0,100000000]，获取数据作业列表时生效。
--finish-from-time	-x	否	查询提供时间之后的所有任务。例如，2006-01-02 15:04:05。
--finish-to-time	-y	否	查询提供时间以前的所有任务。例如，2006-01-02 15:04:05。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

- 获取作业列表

```
ai4s get data-job
# 返回结果如下
Name   ID    Type           Status   Creator  Created      Duration
job-01 xxxxx BACKUP_DATA     Failed   LMX-01  2021-02-01 11:11:27 2s
job-02 xxxxx BACKUP_DATA_DELETE Succeeded LMX-01  2021-02-01 11:11:27 48m44s
```

- 获取作业详情

```
ai4s get data-job 12345678
# 返回结果如下
{
  "complete_data": [
    "0000000000/"
  ],
  "running_data": [],
  "creator": "ei_eihealth_02",
  "source_project_name": "drug-screening-demo-01",
  "id": "xxxx",
  "name": "abc/abc",
  "sources": [
    "0000000000/"
  ],
  "create_time": "2021-06-17T07:57:35Z",
  "end_time": "2021-06-17T11:46:37Z",
  "status": "RUNNING",
  "destinations": [
    "xxxx/"
  ],
}
```

```
"type": "CLONE_DATA"  
}
```

5.2 删除数据作业

使用`delete`命令删除数据作业。

命令结构

```
ai4s delete data-job <data-job-id> [flags]
```

表 5-2 参数说明

参数	简写	是否必选	说明
data-job-id	无	是	数据作业id。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s delete data-job 12344567  
# 返回结果如下  
delete data job 1234356 successfully!
```

5.3 重试数据作业

使用`retry`命令重试数据作业。

命令结构

```
ai4s retry data-job <data-job-id> [flags]
```

表 5-3 参数说明

参数	简写	是否必选	说明
data-job-id	无	是	数据作业id。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s retry data-job 122344  
# 返回结果如下  
retry data job 1234356 successfully!
```

5.4 取消数据作业

使用cancel命令取消数据作业。

命令结构

```
ai4s cancel data-job <data-job-id> [flags]
```

表 5-4 参数说明

参数	简写	是否必选	说明
data-job-id	无	是	数据作业id。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s cancel data-job 1234356
# 返回结果如下
cancel data job 1234356 successfully!
```

6 镜像管理命令

6.1 标记镜像

使用**ai4s docker tag**命令给待上传的镜像打标签。

执行镜像相关命令，请在本地搭建Docker环境，要求安装的容器引擎版本必须为1.11.2及以上。

默认标记到当前所在空间，需先使用切换空间命令进入想要上传镜像的空间。

命令结构

```
ai4s docker tag <source-image-name:source-tag-name> <target-image-name:target-tag-name>
```

表 6-1 参数说明

参数	简写	是否必选	说明
source-image-name	无	是	本地源镜像名称。
source-tag-name	无	是	本地源镜像tag名称。
target-image-name	无	是	新创建的镜像名称。
target-tag-name	无	是	新创建的镜像tag名称。

命令示例

```
ai4s docker tag demo-image:v1.0 project-demo-image:v1.0
```

- 若当前所在空间为lmx-test-01，执行如上命令后，本地镜像列表会新增lmx-test-01/project-demo-image:v1.0。

- 若当前不在某个空间内，会提示：
Error: you're currently not in any project. run the switch project command to enter a project first
Usage:
ai4s switch project <project-name> [flags]

Flags:
-h, --help help for switch

6.2 上传镜像

通过ai4s docker push命令将镜像上传至AI科学计算平台空间中。

命令结构

```
ai4s docker push <image-name:tag-name> [flags]
```

表 6-2 参数说明

参数	简写	是否必选	说明
image-name	无	是	镜像名称。
tag-name	无	是	镜像tag名称。
--description	-d	否	描述。
--type	-t	否	镜像类型，可选APP、NOTEBOOK或OTHER，不填写默认为OTHER。
--chip	-c	否	芯片类型，只支持X86和ARM。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

- Windows
ai4s docker push demo-image:v1.0 -d "this is a desc" -t APP
- Linux
./ai4s docker push demo-image:v1.0 -d 'this is a desc' -t APP
- 上传成功返回值
fe4c16cbf7a4: Pushed
v1.0: digest: sha256:eb7e3bbd8e3040efa71d9c2cacfa12a8e39c6b2ccd15eac12bdc49e0b66cee63 size: 948
- 上传失败可能原因
 - 镜像命名不规范
 - 镜像版本不存在
 - 上传镜像数量超出配额
 - 待上传空间不存在

 说明

对于--description 参数，当描述中带有空格时，需要添加引号来获取完整的描述信息。

6.3 下载镜像

通过ai4s docker pull命令下载空间中的镜像。

命令结构

```
ai4s docker pull <project-name>/<image-name>:<tag-name> [flags]
```

表 6-3 参数说明

参数	简写	是否必选	说明
project-name	无	否	源空间名称，要下载的镜像所在的源空间名称，不填时默认为当前空间。
image-name	无	是	镜像名称。
tag-name	无	是	镜像tag名称。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

```
ai4s docker pull demo-project/demo-image:v1.0
# 执行成功返回结果如下
6be353e4af51: Pull complete
Digest: sha256:8c00710a899746ec54677db64fd3284d2f880bd6265eb0347be1e0cb1cce6530
Status: Downloaded newer image for demo-project/demo-image:v1.0
```

6.4 查询镜像

使用ai4s docker images命令查询空间中的镜像。

命令结构

```
ai4s docker images [flags]
```

表 6-4 参数说明

参数	简写	是否必选	说明
--name	-n	否	查询指定镜像名下的所有tag信息，支持模糊查找。
--type	-t	否	查询某个类型的镜像，可选APP、NOTEBOOK、OTHER。

参数	简写	是否必选	说明
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

- 查询当前所在空间demo-project下面的类型为APP的镜像。
ai4s docker images -t APP
执行成功返回结果如下
Name Source Project Type Chip Type Created Updated
autodruglikeness demo-project APP -- 2021-05-12 10:18:32 2021-05-12 10:18:32

Total images of project demo-project is: 1
- 查询当前空间下面的镜像名为demo-image的所有tag信息。
ai4s docker images -n demo-image
执行成功返回结果如下
Name Version Size Created Updated
demo-image-1 1.0 3.44M 2021-02-01 11:11:27 2021-02-01 11:11:27
demo-image-2 1.1 9.12M 2021-02-01 11:11:27 2021-02-01 11:11:27
demo-image-3 1.1 100M 2021-02-01 11:11:27 2021-02-01 11:11:27

6.5 导入镜像

使用import命令从源空间导入镜像到当前空间。

只支持导入私有镜像，导入镜像和订阅的镜像不支持再次导入到别的空间。

命令结构

```
ai4s docker import <project-name/image-name:tag-name> [flags]
```

表 6-5 参数说明

参数	简写	是否必选	说明
project-name	无	是	要导入的镜像当前所在的空间名称。
image-name	无	是	要导入的镜像名称。
tag-name	无	是	要导入的镜像tag名称。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

从源空间lmx-project-02导入镜像到当前空间。

```
ai4s docker import lmx-project-02/demo-image-1:1.0  
# 执行成功返回结果如下  
import image successfully!
```

6.6 更新镜像

使用**update**命令更新当前空间中的某个镜像仓库的描述信息或者类型。

命令结构

```
ai4s docker update <project-name/image-name> [flags]
```

表 6-6 参数说明

参数	简写	是否必选	说明
project-name	无	否	要更新的镜像的源空间名称，不填写时默认为当前所在的空间。
image-name	无	是	要更新的镜像名称。
--type	-t	否	镜像类型，只支持APP或者NOTEBOOK。
--description	-d	否	镜像描述。当描述中带有空格时，需要添加引号来获取完整的描述信息。
--chip	-c	否	镜像芯片类型，只支持ARM或者X86。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

- Linux
./ai4s docker update lmx-project-01/demo-image --type APP --description 'this is a demo'
- Windows
ai4s docker update lmx-project-01/demo-image --type APP --description "this is a demo"
- 不填写空间名时，若当前所在空间是lmx-test-01，则执行如下命令后会更新当前空间中源空间为lmx-test-01的镜像demo-image的类型。
ai4s docker update demo-image --type APP --chip ARM
执行成功返回结果如下
update image successfully!

6.7 删除镜像标签

使用**rmi**命令删除当前空间中指定镜像标签。

- 对于本空间的私有镜像tag会做彻底删除，即删除数据库记录及远程仓库中的镜像tag。
- 对于其他空间的导入镜像tag或者资产市场订阅的镜像tag仅删除导入或者订阅关系，即只删除数据库记录。

命令结构

```
ai4s docker rmi <project-name/image-name:tag-name> [flags]
```

表 6-7 参数说明

参数	简写	是否必选	说明
project-name	无	否	要删除的镜像的源空间名称，不填时默认为当前所在空间。
image-name	无	是	要删除的镜像tag所在的仓库名称。
tag-name	无	是	要删除的镜像tag名称。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

假设当前所在空间为lmx-project-01，命令及返回结果示例：

- 删除其他空间lmx-project-02的导入镜像tag。
ai4s docker rmi lmx-project-02/demo-image:v1.1
执行成功返回结果如下
untagged:lmx-project-02/demo-image:v1.1
- 删除资产市场订阅的镜像tag。
ai4s docker rmi [AssetMarket]/demo-image:v1.1
执行成功返回结果如下
untagged:[AssetMarket]/demo-image:v1.1
- 删除本空间的私有镜像tag，可不填源空间名。
ai4s docker rmi demo-image:v1.1 或 ai4s docker rmi lmx-project-01/demo-image:v1.1
执行成功返回结果如下
untagged:lmx-project-01/demo-image:v1.1

7

应用管理命令

7.1 应用配置文件说明

AI科学计算平台中的每一个分析作业都依托于应用运行。应用是生物信息学软件和运行该软件所依赖的运行环境的镜像封装。

在AI科学计算平台，创建应用的过程通过图形化的界面操作完成。在命令行工具中，该过程以配置文件的形式给出。您可以基于获取到的模板使用命令行工具创建应用，创建好的应用将同步显示到AI科学计算平台。

获取应用模板

使用`ai4s get app -s`命令获取创建应用的模板，复制模板并保存到本地，您可以保存成.yaml或.txt文件，保存为txt文件时，其内容需为yaml格式。

在本地编写模板，模板中的参数与在AI科学计算平台新建应用过程一致。模板修改好后，再使用命令行工具上传模板，创建应用。

镜像制作和上传方法请参见[镜像管理](#)和[创建FastQC应用样例](#)。

```
# 详细说明可参考API文档中应用管理-创建应用
app:
  name: 'demo-app'          # 应用名称 取值范围：[1,56]，以大小写字母开头，允许出现中划线(-)、大小写字母和数字，且必须以大小写字母或数字结尾。更新应用时，应用名称不支持修改。
  version: '1.0.0'          # 应用版本 取值范围：[1,24]，以小写字母或数字或大写字母开头，允许出现中划线，必须以小写字母或数字或大写字母结尾。更新应用时，应用版本不支持修改。
  summary: 'summary'        # 应用简述 取值范围[0,128]
  description: 'description' # 应用描述 取值范围[0,65535]，后续支持markdown文本
  labels:                   # 应用标签 取值范围[0,5]，单个标签最大长度32字符，支持中文、字母、数字、空格、下划线和中划线，且不能以空格开头或者结尾。
    - labelA
    - labelB
  image: 'gwj-test-01/busybox:latest' # docker镜像地址，取值范围[5-255]，不能包含中文字符
  commands:                   # docker启动时执行命令 单行命令长度取值范围[0-256]，不能包含中文字符，支持多行输入，最多支持300行。
    - 'echo eihealth;'
  node_labels:                # 计算节点标签 标签个数取值范围[0-1],单个节点标签最大长度56个字符，以字母或数字开头，可以包含下划线(_)、中划线(-)、点(.)、字母和数字，且必须以字母或数字结尾
    - ai4s.label
  resources:                  # 应用的资源配额
    cpu_type: 'X86'           # cpu架构类型，不填默认X86
    cpu: '0.1C'               # cpu申请使用量，取值范围[0.1-128]，单位C，支持一位小数。对于应用，不填默认1C；对于流程和作业，不填默认使用前一级的配置，填值会覆盖更新。覆盖关系：作业->流程->应用
```

```
memory: '0.1G' # 内存申请使用量，取值范围[0.1-3072]，单位G，支持一位小数。对于应用，不填默认1G；对于流程和作业，不填默认使用前一级的配置，填值会覆盖更新。覆盖关系：作业->流程->应用
gpu_type: " # gpu架构类型，取值范围 '|GPU|Snt9'。对于流程和作业，不填默认使用前一级的配置，填值会覆盖更新。覆盖关系：作业->流程->应用
gpu: '0' # gpu申请使用量，取值范围[0-16]，仅支持整数，Snt9有特殊约束，申请数量需要是0,1,2,4,8。
# 对于应用，不填默认0；对于流程和作业，不填默认使用前一级的配置，填值会覆盖更新。覆盖关系：作业->流程->应用
inputs:
  - name: 'input-dir' # 参数名称，单个应用内唯一。取值范围：长度为[1,32]，以小写字母开头，允许出现中划线(-)、小写字母和数字，且必须以小写字母或数字结尾。
    type: DIRECTORY # 参数类型。取值：[STRING, FILE, DIRECTORY, ENUM]
    pattern: '*.fastq' # 提示用户参数填写的格式，取值范围：[0-64]。对于STRING类型，匹配字符串内容，比如后缀约束.fastq；
    # 对于ENUM类型，可以提示一定要在param_enum列表范围内取值；对于FILE类型，约束文件后缀类型；对于DIRECTORY类型，提示xxx；
    required: true # 参数是否必须，取值[true,false]
    concurrent: vars_iter # 是否并发，取值 vars_iter 时为开启并发状态，不设置则不开启
    description: " # 参数描述。取值范围：[0-255]
    values: # 参数取值 如填写，只支持填一项，根据参数类型进行不同的校验
      - 'gwj-test-01:/test'
  - name: 'input-enum' # 参数名称，单个应用内唯一。取值范围：长度为[1,32]，以小写字母开头，允许出现中划线(-)、小写字母和数字，且必须以小写字母或数字结尾。
    type: ENUM # 参数类型。取值：[STRING, FILE, DIRECTORY, ENUM]
    pattern: " # 提示用户参数填写的格式，取值范围：[0-64]。对于STRING类型，匹配字符串内容，比如后缀约束.fastq；
    # 对于ENUM类型，可以提示一定要在param_enum列表范围内取值；对于FILE类型，约束文件后缀类型；对于DIRECTORY类型，提示xxx；
    required: true # 参数是否必须，取值[true,false]
    description: " # 参数描述。取值范围：[0-255]
    enum: # type 为 ENUM 类型时需要添加enum选项
      - test
      - test2
    values: # 参数取值 如填写，只支持填一项，只支持enum内的选项
      - test
outputs:
  - name: 'output-dir'
    type: DIRECTORY
    pattern: '*'
    required: true
    description: "
    values:
      - '空间名称:/output'
```

7.2 创建应用

使用create app命令引用本地的配置文件，创建应用。

命令结构

```
ai4s create app [flags]
```

表 7-1 参数说明

参数	简写	是否必选	说明
--yaml	-y	否	本地的应用模板路径。获取应用模板方法请参见 应用配置文件说明 。
--description	-d	否	应用的详细描述信息。

参数	简写	是否必选	说明
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

使用本地已填写好的应用模板app.yaml创建应用。

```
ai4s create app D:\app.yaml 或 ai4s create app -y D:\app.yaml
# 返回结果如下
create app succeed! app id : 84ffe6f7-3f4a-11eb-868a-fa163e3ddba1
```

7.3 修改应用

使用edit或update命令修改应用的内容，应用的名称和版本不支持修改。

命令结构

```
ai4s edit app ID [flags]
# edit和update作用相同
ai4s update app ID [flags]
```

表 7-2 参数说明

参数	简写	是否必选	说明
ID	无	是	可选： - 应用的ID（ app-id ），使用命令行工具创建应用时生成。示例请参见创建应用命令示例。 - 应用的名称、版本、所在空间名称（ app-name:version:srcproject ），srcproject为源空间名称，可选。不指定srcproject时，默认为当前空间。例如 GATK:4.0:projectname。
--yaml	-y	否	本地的应用模板路径。
--summary	-s	否	应用的简要描述。
--description	-d	否	应用的详细描述。
--commands	-c	否	镜像启动命令。设置方法请参见 创建FastQC应用样例 。
--labels	-l	否	标签，多个标签使用;符号分隔开。

参数	简写	是否必选	说明
--nodeLabels	-n	否	用于让应用调度到设置了该标签的节点上。标签数量取值范围[0,1]。单个标签最大长度为56字符。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

说明

- 如果同时使用yaml模板和参数命令对描述、标签等参数进行了修改，最终修改结果以参数命令为准。
- 对于可选参数，如果命令中包含了特殊字符，需使用""括起来。
- 如果特殊字符中包含了双引号，需依据不同操作系统对特殊字符的规范来设置参数命令，例如：

```
# 原镜像启动命令
bash -c "source activate my-rdkit-env && python adme.py --file_in ${file-in} --file_out ${dir-out}/adme_info.txt > ${dir-out}/adme.log 2>&1";cat ${dir-out}/adme.log

# windows系统下镜像启动命令
"bash -c ""source activate my-rdkit-env ^&& python adme.py --file_in ${file-in} --file_out ${dir-out}/adme_info.txt ^> ${dir-out}/adme.log 2^>^&1""";cat ${dir-out}/adme.log"

# Linux系统下镜像启动命令
'bash -c "source activate my-rdkit-env && python adme.py --file_in ${file-in} --file_out ${dir-out}/adme_info.txt > ${dir-out}/adme.log 2>&1";cat ${dir-out}/adme.log'
```

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

- 通过指定“app-id”修改应用。
ai4s edit app 550e8400-e29b-41d4-a716-446655440000 -y D:\template.yaml
返回结果如下
edit app succeed!
- 通过指定“app-name:version:srcproject”修改应用，并修改描述信息、标签。
ai4s edit app app-name:version:srcproject -d "详细描述" -s "简要描述" -c "fastqc" -l "labelA;labelB"
返回结果如下
edit app succeed!

7.4 查询应用

使用get命令查询应用的详细信息，该命令同时可用于获取应用模板。

命令结构

```
ai4s get app ID [flags]
```

表 7-3 参数说明

参数	简写	是否必选	说明
ID	无	否	- 不选此参数时，列出当前所在空间的所有应用信息。 - 指定app-id或app-name:version:srcproject，列出具体的应用信息，srcproject为源空间名称，可选。不指定srcproject时，默认为当前空间。例如 GATK:4.0:projectname。
--sample	-s	否	获取应用模板，模板为yaml格式。
--downloadPath	-d	否	获取应用详情时，将内容下载到的指定文件夹路径（文件夹需要存在）。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。
--label	-l	否	根据label标签搜索应用。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

- 使用**ai4s get app -s**命令获取模板，详细的模板介绍和使用请参见[获取应用模板](#)。
- 使用**ai4s get app**命令查询指定应用信息。

```
ai4s get app 550e8400-e29b-41d4-a716-446655440000

ai4s get app app-name:version

get app successfully.
指定app-id时返回结果:
app:
  id: f165b7a6-5bc9-11eb-9fef-fa163ef9b34d
  name: lk-app-0121
  version: 1.0.0
  summary: dsfsdf
  description: fdsfsdf
  labels:
    - fdsfsdfsdf
  image: ei_eihealth_xxxx_02/modelarts-base-cpu-py3:custom-2.0.2
  commands:
    - echo lk-app-0121
  resources:
    cpu_type: X86
    cpu: 0.1C
    memory: 0.1G
    gpu_type: ""
    gpu: "0"
  inputs:
    - name: input1
      pattern: asfas
      type: DIRECTORY
      required: false
      description: dasdad
      enum: []
```

```
values: []
- name: input2
  pattern: ""
  type: FILE
  required: false
  description: ""
  enum: []
  values: []
outputs:
- name: output1
  pattern: dsfsdf
  type: DIRECTORY
  required: false
  description: sdfsdf
  values:
    - 空间名称:/sdsds
  license: ""

ai4s get app -s
获取app.yaml模板，以yaml格式打印到控制台
app:
# 系统唯一标识，由系统随机分配，说明如下：
# 1. 通过文件创建app时(UI\CMD)，id必须为空，不允许指定
# 2. 通过文件修改app时(CMD)，id为空或与命令行中指定的保持一致
# 3. 资产市场中发布的app，id为空
id: "                # 可选

# 业务唯一标识由name、version组成，空间内唯一
# 1. 通过命令行修改app时，如命令行中指定app:version形式，必须和文件中的app:version保持一致
name: 'test-app'      # 必选 # app名称
version: '1.0.0'      # 必选 # app版本，app名称+app版本在一个空间内需要保持唯一且不允许修改
summary: 'summary'    # 可选 # 短描述
description: 'description' # 可选 # 完整描述文本

labels:                # 可选 # 标签
- labelA
- labelB
image: 'xxxx'          # 必选 # 镜像地址
commands:              # 必选 # 命令列表
- xxx
resources:             # 必选 # 运行环境要求
  cpu_type: X86         # 必选 # 'CPU架构，如X86、ARM'
  cpu: 2C               # 可选 # 'CPU数量'
  memory: 4G            # 可选 # '内存大小'
  gpu_type:             # 可选 # 'GPU类型'
  gpu:                 # 可选 # 'GPU数量'
inputs:                # 可选 # 输入参数列表
- name: 'input_name1'   # 参数名称
  type: FILE            # 参数类型 # STRING|FILE|DIRECTORY|ENUM
  pattern: '*.fastq'     # 参数模式，界面展示，不做强校验
  required: true         # 是否必填
  description: "         # 参数描述
  enum:                 # 枚举类型时，枚举值列表
    - xxx
  values:               # 参数值，非数组类型仅允许填写单个值
    - xxx
outputs:               # 可选 # 输出参数列表，app创建、修改时指定
- name: 'output_name1'
  type: FILE            # FILE和DIRECTORY两种类型，指定输出的相对路径
  pattern: '*'          # 参数模式，界面展示，不做强校验
  required: true        # 是否必填
  description: "        # 描述
  values:
    - 空间名称:/xxx      #当type为FILE or DIRECTORY时，values中指定的路径为子路径，
                          实际路径为${workflow.output_dir}/xxx
  license: xxx           #app的许可证
```

```
ai4s get app -s > app.yaml
将app yaml模板文件存储到当前所在路径的app.yaml文件中

ai4s get app 91a3bdf6-c445-11ec-a9f3-fa163e507c84 -d ./a
download app detail successfully!

# 根据label标签搜索应用
ai4s get app -l labelA
id                name                version
label

                                source_eihealth_project_name user_name
create_time      update_time
913b4116-aaa2-11ee-a057-fa163ef319da cli-demo-app    2.0.0
labelA,labelB

                                test-new-project      wxw-test-admin      2024-01-04
09:42:49      2024-01-04 09:42:49
```

7.5 删除应用

使用delete命令删除指定应用。

命令结构

```
ai4s delete app ID [flags]
```

表 7-4 参数说明

参数	简写	是否必选	说明
ID	无	是	应用的ID（ app-id ）或应用的名称、版本、所在空间名称（ app-name:version:srcproject ）。srcproject为源空间名称，可选。不指定srcproject时，默认为当前空间。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

通过指定应用的ID或名称和版本删除应用。

```
ai4s delete app 550e8400-e29b-41d4-a716-446655440000 或 ai4s delete app app-name:version:srcproject
# 返回结果如下
delete app success,app id : 550e8400-e29b-41d4-a716-446655440000
```

7.6 导入应用

使用import命令导入应用，当前只支持导入单个应用。

订阅和已导入的应用不支持再被导入到别的空间。

命令结构

```
ai4s import app <app-name:version:source_project_name> [flags] 或 ai4s import app <app-id:source_project_name> [flags]
```

表 7-5 参数说明

参数	简写	是否必选	说明
source_project_name	无	是	要导入的应用当前所在的源空间名称。
app-id	无	否	源应用ID。
app-name	无	否	源应用名称。
version	无	否	源应用版本。
--rename	-r	否	重命名目标应用名称，不填时与源应用名称保持一致。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

从源空间lmx-project-01导入应用，并重命名应用。

```
ai4s import app ae81debc-c209-11eb-8a0d-fa163e3ddba1:lmx-project-01 --rename lmx-test-import-app 或
ai4s import app testapp:testversion:lmx-project-01 --rename lmx-test-import-app
# 返回结果如下
import app finished! result is:
[
  {
    "source_app_id": "93a4f757-f5b9-11eb-9fef-fa163ef9b34d",
    "destination_app_id": "0c13b198-f5c9-11eb-9fef-fa163ef9b34d",
    "destination_app_name": "lmx-test-import-app",
    "version": "1.0.0",
    "status": "IMPORT_SUCCESS"
  }
]
```

8 AI 科学计算服务流程管理命令

8.1 流程配置文件说明

AI科学计算服务中的分析流程包含分析过程中所需应用的执行信息和数据的输入、输出等参数定义。分析流程至少由一个应用组成，在多个应用构成的流程中，一个应用的输出作为另一个应用的输入，流程中的各个应用由其前后顺序形成完整的计算 workflow。

在AI科学计算平台，创建流程通过拖拽应用的方式完成。在命令行工具中，该过程以配置文件的形式给出，对于由多个应用搭建出来的流程，命令行工具中通过指定不同应用间的输入输出关系，完成应用的连接。您可以基于获取到的模板使用命令行工具创建流程，创建好的流程将同步显示到AI科学计算平台。

获取流程模板

使用 **ai4s get workflow -s** 命令获取创建流程的模板，复制模板并保存到本地，您可以保存成 .yaml 或 .txt 文件，保存为 .txt 文件时，其内容需为 .yaml 格式。

在本地编写模板，模板填写好后，再使用命令行工具上传模板，创建流程。示例流程由 app1 和 app2 两个应用构成，通过指定输入、输出关系链接两个应用。

详细说明可参考API文档中流程管理- 创建流程

```
workflow:
  name: 'demo-workflow'          # 流程名称，取值范围[1,56]，以大小写字母开头，允许出现中划线(-)、大小写字母和数字，且必须以大小写字母或数字结尾。更新流程时，流程名称不支持修改。
  version: '1.0.0'               # 流程版本，取值范围[1,24]，以小写字母或数字或大写字母开头，允许出现中划线，必须以大小写字母或数字结尾。更新流程时，流程版本不支持修改。
  summary: 'summary'             # 流程简述 取值范围[0,128]
  description: 'description'      # 流程描述 取值范围[0,65535]，后续支持markdown文本
  labels:                        # 流程标签，取值范围[0,5]，单个标签最大长度32字符，支持中文、字母、数字、空格、下划线和中划线，且不能以空格开头或者结尾。
    - labelA
    - labelB
  timeout: 1440                  # 流程超时时间，取值范围[1,144000]，单位分钟，默认1440
  output_dir:                   # 流程的当前工作目录，默认为根目录，用户可显示指定，输出路径必须以斜杠(/)开头且不能以斜杠(/)结尾，不能包含两个以上相邻的斜杠(/)，不能包含以下特殊字符：\ ; * ? < " > |。其中单个文件夹名称不能以中划线(-)开头，不能以英文句号(.)或斜杠(/)或空格开头或结尾
  tasks:                        # 流程中子任务的描述信息，子任务数量取值范围:[1,128]
    - task_name: app1-1          # 子任务实际名称，取值范围[1,32]，只能以大小写字母开头，由大小写字母、数字、中划线(-)、下划线(_)组成，以大小写字母或数字结尾。
      app_id: 'demo-app::1.0.0:gwj-test-01' # 应用id，取值范围[1,135]，正则先不能有中文，两种格式。特殊id，采用{app_name}::{app_version}::{src_project_name}格式，用于手动创建场景；
```

```
display_name: '步骤1' # 其他场景, app_id为系统分配的唯一标识
resources: # 流程的子任务展示名称, 最大长度64
  cpu: '0.1C' # 子任务的资源配额
  # cpu申请使用量, 取值范围[0.1-128], 单位C, 支持一位小数。对于应用, 不填默认1C; 对于流程和作业, 不填默认使用前一级的配置, 填值会覆盖更新。覆盖关系: 作业->流程->应用
  memory: '0.1G' # 内存申请使用量, 取值范围[0.1-3072], 单位G, 支持一位小数。对于应用, 不填默认1G; 对于流程和作业, 不填默认使用前一级的配置, 填值会覆盖更新。覆盖关系: 作业->流程->应用
  gpu_type: '' # gpu架构类型, 取值范围 '|GPU|Snt9'。对于流程和作业, 不填默认使用前一级的配置, 填值会覆盖更新。覆盖关系: 作业->流程->应用
  gpu: '0' # gpu申请使用量, 取值范围[0-16], 仅支持整数, Snt9有特殊约束, 申请数量需要是0,1,2,4,8。对于应用, 不填默认0; 对于流程和作业, 不填默认使用前一级的配置, 填值会覆盖更新。
  # 覆盖关系: 作业->流程->应用
output_dir: # 子任务的输出存放路径, 用户可显示指定, 输出路径必须以斜杠 (/) 开头且不能以斜杠 (/) 结尾, 不能包含两个以上相邻的斜杠 (/), 不能包含以下特殊字符: \ : ; * ? < " > |。其中单个文件夹名称不能以中划线 (-) 开头, 不能以英文句号 (.) 或斜杠 (/) 或空格开头或结尾
inputs:
  - name: 'input-dir' # 子任务的参数名称, 长度为[1,32], 以小写字母开头, 允许出现中划线 (-)、小写字母和数字, 且必须以小写字母或数字结尾。需要和已有应用的参数名称一致。
  values: # 子任务的参数数值, 根据参数类型进行合法性校验
    - 'gwj-test-01:/test'
- task_name: app1-2
  app_id: 'demo-app::1.0.0'
  display_name: '步骤2'
  resources:
    cpu: '0.1C'
    memory: '0.1G'
    gpu_type: ''
    gpu: '0'
  output_dir:
  inputs:
    - name: 'input-dir'
    values: # 子任务的参数数值, 格式为${task_name.output_name}, 表示连线关系, 即app1-2需要app1-1成功执行后才会运行
      - '${app1-1.output-dir}'
```

8.2 创建流程

使用create workflow命令引用本地的配置文件, 创建流程。

命令结构

```
ai4s create workflow [flags]
```

表 8-1 参数说明

参数	简写	是否必选	说明
--yaml	-y	否	本地的流程模板路径。获取流程模板方法请参见 流程配置文件说明 。
--description	-d	否	流程的详细描述信息。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

使用本地已填写好的流程模板workflow.yaml创建流程。

```
ai4s create workflow D:\workflow.yaml 或 ai4s create workflow -y D:\workflow.yaml
# 返回结果如下
create workflow succeed! workflow id : 65b7a404-3f4f-11eb-868a-fa163e3ddb81
```

8.3 修改流程

使用edit或update命令修改流程的内容，流程的名称和版本不支持修改。

命令结构

```
ai4s edit workflow ID [flags]
# edit和update作用相同
ai4s update workflow ID [flags]
```

表 8-2 参数说明

参数	简写	是否必选	说明
ID	无	是	可选： - 流程的ID（ workflow-id ），使用命令行工具创建流程时生成。示例请参见创建流程命令示例。 - 流程的名称、版本、所在空间名称（ workflow-name:version:srcproject ），srcproject为源空间名称，可选。不指定srcproject时，默认为当前空间。例如 ngs:1.0:srcproject。
--yaml	-y	否	本地的流程模板路径。
--summary	-s	否	流程的简要描述。
--description	-d	否	流程的详细描述。
--output_dir	-o	否	输出路径。不指定时，以当前空间的根目录为工作路径。
--timeout	-t	否	超时时间。运行时间超过设置时间时，认为超时，默认1440分钟，最大可设置为144000分钟。
--labels	-l	否	标签。多个标签使用;符号分隔开。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

 说明

对于可选参数，如果命令中包含了特殊字符，Windows系统下需使用""括起来，Linux系统下需使用"括起来。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

- 通过指定“workflow-id”修改流程。
ai4s edit workflow 65b7a404-3f4f-11eb-868a-fa163e3ddb1 -y D:\template.yaml
返回结果如下
edit workflow succeed!
- 通过指定“workflow-name:version:srcproject”修改流程，并修改描述信息、输出路径、超时时间、标签。
ai4s edit workflow workflow-name:version:srcproject -d "详细描述" -s "简要描述" -o "/data/output" -t 2000 -l "labelA;labelB"
返回结果如下
edit workflow succeed!

8.4 查询流程

使用get命令查询流程的详细信息，该命令同时可以用于获取流程模板。

命令结构

```
ai4s get workflow ID [flags]
```

表 8-3 参数说明

参数	简写	是否必选	说明
ID	无	否	- 不选此参数时，列出当前所在空间的所有流程信息。 - 指定workflow-id或workflow-name:version:srcproject，列出具体应用的信息，srcproject为源空间名称，可选。不指定srcproject时，默认为当前空间。
--sample	-s	否	获取流程模板，模板为yaml格式。
--downloadPath	-d	否	获取workflow详情时，将内容下载到的指定文件夹路径（文件夹需要存在）。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。
--label	-l	否	根据label标签搜索workflow。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

- 使用**ai4s get workflow -s**命令获取模板，详细的模板介绍和使用请参见[获取流程模板](#)。
- 使用**ai4s get workflow**命令查询指定的流程信息。

```
ai4s get workflow 550e8400-e29b-41d4-a716-446655440000
ai4s get workflow workflow-name:version
```

```
workflow:
  id: 3a7fa782-5bcb-11eb-9fef-fa163ef9b34d
  name: lk-workflow-0121
  version: 1.0.0
  summary: SDFDSF
  description: SDFDSFDSF
  labels:
  - SDFDSF
  timeout: 1440
  output_dir: /lk-workflow-0121
  tasks:
  - task_name: task-1-lk-app-0121
    app_id: f165b7a6-5bc9-11eb-9fef-fa163ef9b34d
    name: ""
    resources:
      cpu: 0.1C
      memory: 0.1G
      gpu_type: ""
      gpu: "0"
    inputs:
    - name: input1
      values:
      - ${task-2-lk-app-0121.output1}
    output_dir: /sdsd
  - task_name: task-2-lk-app-0121
    app_id: f165b7a6-5bc9-11eb-9fef-fa163ef9b34d
    name: ""
    resources:
      cpu: 0.1C
      memory: 0.1G
      gpu_type: ""
      gpu: "0"
    inputs:
    - name: input1
      values:
      - ei_eihealth_xxxx_02:/yexihao/
    output_dir: /asdsd
  - task_name: task-4-lk-app-0121
    app_id: f165b7a6-5bc9-11eb-9fef-fa163ef9b34d
    name: ""
    resources:
      cpu: 0.1C
      memory: 0.1G
      gpu_type: ""
      gpu: "0"
    inputs:
    - name: input1
      values:
      - ei_eihealth_xxxx_02:/yexihao/
    - name: input2
      values: []
    output_dir: /sd
  - task_name: task-5-zx-2id-2od
    app_id: "5"
    name: ""
    resources:
      cpu: 0.1C
```

```
memory: 0.1G
gpu_type: ""
gpu: "0"
inputs:
- name: indir-01
  values:
  - ${task-1-lk-app-0121.output1}
- name: indir-02
  values:
  - ${task-4-lk-app-0121.output1}
output_dir: /sadsd
license: BSD

ai4s get workflow -s
获取workflow yaml模板文件，以yaml格式打印到控制台

workflow:
# 处理规则同app
id: xxxx                                # 可选 # 流程唯一标识，由系统随机分配

# 处理规则同app
name: xxxx                             # 必选 # workflow名称
version: xxxx                           # 必选 # workflow版本，workflow名称+workflow版本在一个空间内保持唯一
summary: xxx                            # 可选 # 短描述
description: 'description'               # 可选 # 完整描述文本

labels:                                 # 可选 # 标签
- labelA
- labelB
timeout: 1440                           # 可选 # 流程超时时间，取值范围[1,144000]，单位分钟，默认1440
output_dir:                             # 可选 # workflow的当前工作目录，默认为根目录，用户可显示指定
tasks:                                  # 必选 # 任务列表
- task_name: app1-1                     # 任务名称
  # 唯一标识处理规则如下：
  # 1. 特殊id，采用{app_name}::{app_version}::{src_project_name}格式，用于手动创建场景
  # src_project_name在使用导入ori订阅的资源是需要填写，为空表示本空间
  # 2. 其他场景，app_id为系统分配的唯一标识
  app_id: xxx                           # 必选 app唯一标识
  name:                                 # task展示名称
  resources:
    cpu: 2C                             # 'CPU数量'
    memory: 4G                           # '内存大小'
    gpu_type:                             # 'GPU类型'
    gpu:                                 # 'GPU数量'
    output_dir:                           # task输出子目录，默认为空时，自动生成task-name子目录，允许在
workflow中配置
  inputs:                               # 输入参数配置，默认覆盖workflow、app中同名配置
  - name: 'input_name1'                  # 参数名
    values:                              # 参数值，覆盖基础配置
    - xxx
  - task_name: app2-1                    # 任务名称
    app_id: xxx                           # app唯一标识
    name:                                 # task展示名称
    resources:
      cpu: 2C                             # 'CPU数量'
      memory: 4G                           # '内存大小'
      gpu_type:                             # 'GPU类型'
      gpu:                                 # 'GPU数量'
      output_dir:                           # task输出子目录，默认为空时，自动生成task-name子目录，允许在
workflow中配置
    inputs:                              # 输入参数配置，默认覆盖workflow、app中同名配置
    - name: 'input_name2'                # 参数名
      values:                            # 参数值，覆盖基础配置
      - xxx
    license: xxx                          # workflow的许可证

# 下载workflow详情到a文件夹
```

```
ai4s get workflow 3df8c00e-0291-11ed-973f-fa163e507c84 -d ./a
download workflow detail successfully!

# 根据label标签搜索workflow
ai4s get workflow -l labelA
id                name                version
label            source_eihealth_project_name user_name
create_time      update_time
501c20c9-aaf8-11ee-a057-fa163ef319da cli-demo-workflow 1.0.0
labelA,labelB
admin 2024-01-04 19:56:36 2024-01-04 19:56:36 test-project-new-permissions-700wwx-test-
4dd32f12-aaf7-11ee-a057-fa163ef319da cli-demo-workflow 2.0.0
labelA,labelB
admin 2024-01-04 19:49:23 2024-01-04 19:49:23 test-project-new-permissions-700wwx-test-
```

8.5 删除流程

使用delete命令删除指定流程。

命令结构

```
ai4s delete workflow ID [flags]
```

表 8-4 参数说明

参数	简写	是否必选	说明
ID	无	是	流程的ID（ workflow-id ）或流程的名称、版本、所在空间名称（ workflow-name:version:srcproject ）， srcproject为源空间名称，可选。不指定srcproject时，默认为当前空间。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

通过指定流程的ID或名称和版本删除流程。

```
ai4s delete workflow 4f712af9-3f51-11eb-868a-fa163e3ddba1 或 ai4s delete workflow workflow-
name:version:srcproject
# 返回结果如下
delete workflow success,workflow id : 4f712af9-3f51-11eb-868a-fa163e3ddba1
```

8.6 导入流程

使用import命令从源空间导入流程到当前空间。

订阅和已导入的流程不支持再被导入到别的空间。

命令结构

```
ai4s import workflow <workflow-name:version:source_project_name> [flags] 或 ai4s import workflow <workflow-id:source_project_name> [flags]
```

表 8-5 参数说明

参数	简写	是否必选	说明
source_project_name	无	是	要导入的流程当前所在的源空间名称。
workflow-id	无	否	源流程ID。
workflow-name	无	否	源流程名称。
version	无	否	源流程版本。
--rename	-r	否	重命名目标流程名称，不填时与源流程名称保持一致。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

从源空间lmx-project导入流程，并重命名流程。

```
ai4s import workflow lmx-wf:v1:lmx-project --rename XXX 或 ai4s import workflow 1467490-2-hfddff:lmx-project --rename XXX
# 返回结果如下
import workflow successfully!
```

9 AI 科学计算服务作业管理命令

9.1 作业配置文件说明

流程创建完成后，可基于已创建的流程运行分析作业。

在AI科学计算平台，运行分析作业的过程通过图形化的界面操作完成。在命令行工具中，该过程以配置文件的形式给出。您可以基于已获取到的模板使用命令行工具启动分析作业，运行的分析作业将同步显示到AI科学计算平台。

获取作业模板

使用`ai4s get job -s`命令获取启动分析作业的模板，复制并保存模板至本地，您可以保存成.yaml或.txt文件，保存为txt文件时，其内容需为yaml格式。

在本地编写模板，模板中的参数与在AI科学计算平台[启动作业](#)过程一致。模板修改好后，再使用命令行工具上传模板，启动分析作业。

```
# 详细说明可参考API文档中作业管理-创建作业
job:
  name: demo-job          # 作业名称，取值范围：[1,63]，允许大小写字母、数字、以及特殊字符中划线(-)
  labels:                 # 作业标签 取值范围[0,5]，单个标签最大长度32字符，支持中文、字母、数字、空格、下划线和中划线，且不能以空格开头或者结尾。
    - labelA
    - labelB
  description: description # 作业描述,取值范围：输入字符最大长度为255
  tool_id:               # 作业依赖的组件id，组件当前仅支持流程，取值范围[1,135]，支持大小写字母和数字。目前支持两种格式，特殊id：{流程名称}::{流程版本}::{源空间名称}；正常id：流程id
  tool_type:             # 作业依赖的组件类型，支持填写app和workflow，如果为app，则需要填写tool_id，如果为workflow，则需要填写workflow_id
  priority: 0            # 作业的优先级,取值范围[0,9]，0最低，默认数值0
  timeout: 1440          # 作业执行超时时长，取值范围：[1, 144000]，单位：分钟，默认数值1440
  output_dir:            # 作业存储目录，不指定则在workflow的工作目录下生产job同名子目录，指定则已指定路径为准，输出路径必须以斜杠(/)开头且不能以斜杠(/)结尾，不能包含两个以上相邻的斜杠(/)，不能包含以下特殊字符：\ ; * ? < " > |。其中单个文件夹名称不能以中划线(-)开头，不能以英文句号(.)或斜杠(/)或空格开头或结尾
  workflow_id: demo-workflow::1.0.0::gwj-test-01 # 作业依赖的组件id，组件当前仅支持流程，取值范围[1,135]，支持大小写字母和数字。目前支持两种格式，特殊id：{流程名称}::{流程版本}::{源空间名称}；正常id：流程id
  io_acc_id:             # IO加速实例id，为空则不开启IO加速，设置相应加速包id将会开启加速，当id设置为 auto_schedule，则会自动调度加速包，当id设置为 local_disk，则开启本地盘加速。
  node_labels:           # 计算节点标签 标签个数取值范围[0-1],单个节点标签最大长度56个字符，以字母或数字开头，可以包含下划线(_)、中划线(-)、点(.)、字母和数字，且必须以字母或数字结尾
```

```
- ai4s.label
tasks:
- task_name: app1-1                                # 子任务实际名称，取值范围[1,32]，只能以大小写字母开头，由字母、数字、中划线(-)、下划线(_)组成，以大小写字母或数字结尾。需要和已有子任务的名称一致。
  inputs:
  - name: input-dir                                # 子任务的输入参数信息
    values:                                         # 子任务的参数数值，根据参数类型进行合法性校验
    - 'gwj-test-01:/test'
  resources:                                       # 子任务的资源配额
    cpu: 0.1C                                     # cpu申请使用量，取值范围[0.1-128]，单位C，支持一位小数。对于应用，不填默认1C；对于流程和作业，不填默认使用前一级的配置，填值会覆盖更新。覆盖关系：作业->流程->应用
    memory: 0.1G                                  # 内存申请使用量，取值范围[0.1-3072]，单位G，支持一位小数。对于应用，不填默认1G；对于流程和作业，不填默认使用前一级的配置，填值会覆盖更新。覆盖关系：作业->流程->应用
    gpu_type: ''                                   # gpu架构类型，取值范围'|GPU|Snt9'。对于流程和作业，不填默认使用前一级的配置，填值会覆盖更新。覆盖关系：作业->流程->应用
    gpu: '0'                                       # gpu申请使用量，取值范围[0-16]，仅支持整数，Snt9有特殊约束，申请数量需要是0,1,2,4,8。对于应用，不填默认0；对于流程和作业，不填默认使用前一级的配置，填值会覆盖更新。
                                                    # 覆盖关系：作业->流程->应用
    io_acc_type: ''                               # 子任务使用的IO加速类型，取值范围SFS、EVS。为空表示不使用加速。SFS表示使用io加速，EVS表示使用本地盘加速
- task_name: app1-2
  inputs:
  - name: input-dir
    values:
    - '${app1-1.output-dir}'
  resources:
    cpu: 0.1C
    memory: 0.1G
    gpu_type: ''
    gpu: '0'
```

9.2 启动作业

通过使用create或 submit命令引用本地的配置文件，启动分析作业。

命令结构

```
ai4s create job [flags]
# create和submit作用相同
ai4s submit job [flags]
```

表 9-1 参数说明

参数	简写	是否必选	说明
--yaml	-y	是	本地的作业模板路径。获取作业模板方法请参见 作业配置文件说明 。
--workflow	-w	否	基于流程ID创建分析作业，可使用ai4s get workflow workflow-name:version:srcproject命令查询流程ID，srcproject为源空间名称，可选。不指定srcproject时，默认为当前空间。 --yaml和--workflow命令不能同时存在。
--description	-d	否	作业的详细描述信息。

参数	简写	是否必选	说明
--input	-i	否	输入参数名称，该参数与流程中的输入参数对应，通过修改--input，修改输入数据。 选择时，input和input-file二选一。 说明 - 作业运行时，每个应用称之为一个task，应用可以设置多个输入参数，一个输入参数可以设置多个输入值。 - task间使用;;分隔，input的参数路径间使用;;分隔。 例如 task1.input1=datapath1;;datapath2;;task1.input2=xxx;xxx;;task2.input1=xxx;xxx
--input-file	-f	否	输入参数对应的文件路径（本地路径），该参数与流程中的输入文件路径对应，通过修改input-file，修改输入数据。 选择时，input和input-file二选一。 说明 - 格式为：{"task0": {"input-1": "datapath"}, "task1": {"inputs": "datapath"}}。 - 同时，输入参数可以设置多个参数值。例如： --input-file D:\data\data.txt # 本地文件中的内容需为json格式，示例如下。 # 同一个input参数，有多个参数值时，需使用;分隔。 <pre>{ "task1": { "data-01": "projectname:/test.txt", "data-02": "projectname:/test2.txt;projectname:/test3.txt" } }</pre>
--output_dir	-o	否	输出路径（AI科学计算平台数据路径）。可自定义。不指定时，按照“作业名称+UUID”格式自动生成存放输出结果的目录。输出路径只能以/开头，不能以/结尾。
--timeout	-t	否	超时时间。运行时间超过设置时间时，认为超时，默认1440分钟，最大可设置为144000分钟。
--priority	-p	否	优先级，[0,9]，9代表最高，0代表最低，默认为0，优先级高的作业会被优先执行。
--name	-n	否	作业的名称。 通过流程ID（workflow id）启动作业时，必选。

参数	简写	是否必选	说明
--io-acc-id	-c	否	IO加速实例id，为空则不开启IO加速，设置相应加速包id将会开启加速。实例id可参考“系统设置命令>获取系统资源”章节获取。当id设置为：auto_schedule，则会自动调度加速包，当id设置为local_disk，则开启本地盘加速。 • 获取性能加速资源列表 <pre>health get resource -t performance # 执行成功返回结果如下 Id Name Specifications xxx sfs-turbo-0l eihealth.storage.ioacc.spec.ioacc</pre>
--nodeLabels	-l	否	用于让作业调度到设置了该标签的节点上。如果所有节点都不满足该标签，则调度失败。 标签数量取值范围[0,1]。单个标签最大长度为56字符。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。
--io-acc-tasks	-s	否	设定任务的I/O加速类型 例如：`taskname1:EVS`;`taskname3:SFS` 不写的task默认不带加速类型

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

说明

启动作业时，如果包含不存在的计算节点标签，回显类似下图，请先添加计算节点标签。

```

D:\Health\health-windows-v1.6.0>ai4s create job -w id/e9714-b96e-11ec-8c0a -n sss
submit job succeed! job id : 453cc72-ba5b-11ec-
The compute node labels of the following app in the job are not set on any compute node. Before you set the labels, the app will be stuck in the waiting for scheduling state.
name      node labels
demo-app-cl health.labels1
You can go to Compute Resources to add the labels.
```

计算节点标签添加方法：在平台右上角单击用户名，选择“系统资源 > 计算资源”，在计算节点的操作列单击“更多 > 标签管理”，添加标签。上图示例中需要添加的标签名称为labelsss1。

- 使用本地已填写好的作业模板job.yaml运行分析作业。
ai4s create job -y D:\job.yaml -d "详细描述"
返回结果如下
submit job succeed! job id : a30f3afd-3f6d-11eb-868a-fa163e3ddb1
- 基于流程（ workflow ）创建分析作业。
ai4s create job -w 550e8400-e29b-41d4-a716-446655440000 -t "120" -d "详细描述" -n "job-name"
返回结果如下
submit job succeed! job id : a30f3afd-3f6d-11eb-868a-fa163e3ddb1
- 使用本地已填写好的作业模板job.yaml运行分析作业，并修改输入数据路径、描述。
ai4s create job -y D:\job.yaml -i "D:\data\hg19.fa" -d "descript"
返回结果如下
submit job succeed! job id : a30f3afd-3f6d-11eb-868a-fa163e3ddb1

- 使用本地已填写好的作业模板job.yaml运行分析作业，并修改输出路径、超时时间、优先级。
ai4s create job -y D:\job.yaml -o "/bucket" -t 5 -p 9
返回结果如下
submit job succeed! job id : a30f3afd-3f6d-11eb-868a-fa163e3ddba1
- 使用本地已填写好的作业模板job.yaml运行分析作业，并增加子task加速类型。
ai4s create job -y D:\job.yaml -s `task-3-two-cp:SFS`;task-2-cp-dir:EVS`
返回结果如下
submit job succeed! job id : a30f3afd-3f6d-11eb-868a-fa163e3ddba1

9.3 查询作业详情

使用get命令查询作业的详细信息，该命令同时可以用于获取作业模板。

命令结构

```
ai4s get job ID [flags]
```

表 9-2 参数说明

参数	简写	是否必选	说明
ID	无	否	- 不选此参数时，列出当前所在空间的所有作业信息。 - 指定job-id时，列出具体的作业信息。可以同时指定多个job-id。 - 不带--detail参数，以yaml格式展示作业基本信息。 - 带--detail参数，以json格式展示作业基本信息。
--detail	-d	否	配合ID使用，返回作业的详细信息。
--sample	-s	否	获取作业模板，模板为yaml格式。
--limit	-l	否	代表当次请求获取的最大查询条数（默认为10）。
--offset	-o	否	偏移量，从第几条数据开始查询，默认为0。
--event	-e	否	获取作业事件或者作业某一task事件，单独使用表示获取作业事件；与--task一起使用表示获取某一个task事件，并同时输出task实例列表。
--log	-g	否	本地存放task日志的路径，必须与--task一起使用以获取作业某一task的日志。
--task	-a	否	task名称。如果是并发的task，那么默认获取索引号为0的task实例，如果要查看别的实例，格式：--task task名称;实例索引，如--task task-1;1。
--finish-from-time	-x	否	查询任务完成起始时间。例如，2006-01-02 15:04:05。

参数	简写	是否必选	说明
--finish-to-time	-y	否	查询任务完成结束时间。例如，2006-01-02 15:04:05。
--create-from-time	-c	否	询任务创建起始时间，例子：--create-from-time="2006-01-02 15:04:05"。
--create-to-time	-m	否	询任务创建结束时间，例子：--create-to-time="2006-01-02 15:04:05"。
--labels	-k	否	作业标签列表。以","分隔，如："a,b"。
--status	-q	否	作业状态(用于获取作业列表)，取值：Succeeded、Running、Pending、Failed、Cancelling、Cancelled、Unknown。
--workflow-name	-t	否	流程名。
--user-name	-u	否	用户名。
--job-name	-j	否	任务名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

- 使用**ai4s get job -s**命令获取模板，详细的模板介绍和使用请参见[获取作业模板](#)。

- 获取作业详情，以模板方式展示。

```
ai4s get job 000c6057-cc6c-11ed-bbec-fa163ef30f89
job:
  id: 000c6057-cc6c-11ed-bbec-fa163ef30f89
  name: job-7402
  description: ""
  priority: 0
  timeout: 1440
  output_dir: /job-7402-de91a3e0-076c-4327-a41c-8e88c7aec6ae
  workflow_id: f1af14bb-cc69-11ed-bbec-fa163ef30f89
  io_acc_id: ""
  node_labels: []
  tasks:
    - task_name: task-1-test-echo
      inputs: []
      resources:
        cpu: 0.1C
        memory: 0.1G
        gpu: "0"
      tool_type: workflow
      tool_id: f1af14bb-cc69-11ed-bbec-fa163ef30f89
      labels: []
```

- 获取作业详情，以json方式展示。

```
ai4s get job f17a3542-3f7c-11eb-868a-fa163e3ddba1 --detail

{
```

```
"jobs": [{
  "id": "2",
  "name": "zx-1030-mkdir",
  "description": "测试文件创建",
  "priority": 0,
  "timeout": 1440,
  "output_dir": "",
  "status": "SUCCEEDED",
  "create_time": "2021-01-20T03:38:14Z",
  "finish_time": "2021-01-20T03:43:23Z",
  "tool_info": {
    "tool_id": "",
    "tool_name": "",
    "tool_version": "",
    "tool_type": ""
  },
  "tasks": [{
    "task_name": "task0",
    "display_name": "",
    "output_dir": "",
    "whole_output_dir": "",
    "resources": {
      "cpu": "0.1C",
      "memory": "0.1G",
      "gpu_type": "",
      "gpu": "0"
    },
    "inputs": [{
      "name": "in-dir",
      "values": [
        "ei_eihealth_xxxx_02/zx-1030/"
      ]
    },
    {
      "name": "in-str",
      "values": [
        "mkdir1030"
      ]
    }
  ]
}],
  "app_info": {
    "app_id": "2",
    "app_name": "zx-1030-mkdir",
    "app_version": "1.0.0",
    "app_src_project_name": "",
    "app_labels": [],
    "app_summary": "",
    "app_description": "",
    "app_image": "ei_eihealth_xxxx_02/modelarts-base-cpu-py3:custom-2.0.2",
    "app_commands": [
      "mkdir ${in-dir}${in-str}"
    ],
    "app_input_parameters": [{
      "name": "in-dir",
      "pattern": "",
      "type": "DIRECTORY",
      "required": true,
      "description": ""
    },
    {
      "name": "in-str",
      "pattern": "",
      "type": "STRING",
      "required": true,
      "description": ""
    }
  ],
  "app_output_parameters": []
}]
```

```
    },
    "task_runtime_info": [{
      "task_name": "task0",
      "status": "SUCCEEDED",
      "create_time": "2021-01-20 11:38:22",
      "finish_time": "2021-01-20 11:43:22",
      "run_time": "5m0s"
    }],
    "dag": {
      "task0": {}
    },
    "io_acc_expected_usage": 10,
    "io_acc_info": {
      "id": "35673038-d57b-4dab-942a-72cf3e11e7df",
      "type": "IO_PERFORMANCE_BANDWIDTH",
      "space": 500,
      "free_space": 500.0
    }
  }
},
"count": 1
}
```

● 获取作业列表。

ai4s get job #不带任何参数默认获取100条

job_id	job_name	tool_name	tool_version	tool_type
status	user_name	create_time	finish_time	labels
4b682e15-ab92-11ee-a057-fa163ef319da	cli-demo-job	cp-test	2.0.0	
workflow	PENDING	wwx-test-admin	2024-01-05 14:18:51	--
e7e55c6e-aaf6-11ee-a057-fa163ef319da	cli-demo-job-import	cli-demo-workflow	4.0.0	
workflow	FAILED	wwx-test-admin	2024-01-04 19:46:32	2024-01-04 19:47:50
aee9e91a-aaf6-11ee-a057-fa163ef319da	job-6685	cli-demo-workflow	4.0.0	
workflow	FAILED	wwx-test-admin	2024-01-04 19:44:56	2024-01-04 19:45:50
58a8f13b-aaf3-11ee-a057-fa163ef319da	job	cp-test	2.0.0	
workflow	FAILED	wwx-test-admin	2024-01-04 19:21:03	2024-01-04 19:23:54
35ff73b3-aaf3-11ee-a057-fa163ef319da	job	cp-test	2.0.0	
workflow	SUCCEEDED	wwx-test-admin	2024-01-04 19:20:05	2024-01-04 19:24:52
24b72eee-aaf3-11ee-a057-fa163ef319da	job	cp-test	2.0.0	
workflow	SUCCEEDED	wwx-test-admin	2024-01-04 19:19:36	2024-01-04 19:25:10
4ccef1fb-aaf2-11ee-a057-fa163ef319da	job	cp-test	2.0.0	
workflow	SUCCEEDED	wwx-test-admin	2024-01-04 19:13:34	2024-01-04 19:17:34

ai4s get job -j cli-demo-job

job_id	job_name	tool_name	tool_version	tool_type
status	user_name	create_time	finish_time	labels
70f1baa8-ab96-11ee-a057-fa163ef319da	cli-demo-job	cp-test	2.0.0	
workflow	SUCCEEDED	wwx-test-admin	2024-01-05 14:48:32	2024-01-05 14:55:13
6c6098f0-ab96-11ee-a057-fa163ef319da	cli-demo-job	cp-test	2.0.0	
workflow	SUCCEEDED	wwx-test-admin	2024-01-05 14:48:24	2024-01-05 14:54:25

ai4s get job -l 3 同 ai4s get job -l 3 -o 0

#列出当前project的job的基本信息

#表示取3条数据，也就是取1-3 条数据

ai4s get job -o 10 同 ai4s get job -l 100 -o 10

#列出当前project的job的基本信息

#表示取100条数据，也就是取11-110 100 条数据

ai4s get job -l 10 -o 3

#列出当前project的job的基本信息

#表示跳过3条数据,从第4条数据开始取，取10条数据，也就是取4-13 10条数据

● 获取作业事件。

ai4s get job 550e8400-e29b-41d4-a716-446655440000 --event

成功关联执行器

2024-01-05 14:18:51

执行 create，共计 1 个子任务

2024-01-05 14:18:51

```
-
执行 create， 共计 1 个子任务
2024-01-05 14:18:51
-----
-
创建k8s Job对象 task-3-two-cp-0-bd5e1f7dac10005f 成功.
2024-01-05 14:18:51
-----
-
等待任务 task-3-two-cp-0-bd5e1f7dac10005f 执行完成
2024-01-05 14:18:56
-----
-
元素(task-3-two-cp-0)第1次重试执行(create)， 当前异常:Failed to wait the Job(task-3-two-cp-0-
bd5e1f7dac10005f) has desiredReplicas: the pod list of job:task-3-two-cp-0-bd5e1f7dac10005f is
empty .
2024-01-05 14:18:51
-----
-
创建k8s Job对象 task-2-cp-dir-0-bd5e1f7dac10005f 成功.
2024-01-05 14:18:56
-----
-
```

● 获取作业某一task事件。

```
ai4s get job 550e8400-e29b-41d4-a716-446655440000 --event --task task-lmx-job-1

Task event list:
Status      Times Type      Details                                     First Report Time      Last
Report Time
SuccessfulCreate 1      Normal    Created pod: task-1-rename-0-1b840133ac100049-hkppv
2022-05-24 18:04:55 2022-05-24 18:04:55
JobsComplete 1      Normal    Pod exits with success, the job is complete      2022-05-24
18:07:09 2022-05-24 18:07:09

Task instances list:
Name              Status      PodIP      Node              RestartCount
Request/Limit(CPU) Request/Limit(Memory) CreateTime
task-1-rename-0-1b840133ac100049-hkppv Succeeded  172.16.1.20 192.168.125.40
0 / / 2022-05-24T10:04:55Z
```

● 获取并发task的实例事件。

```
ai4s get job c5b3d272-f398-11ec-845a-fa163ef3fac0 --task task-1-test-bingfasmial;1 --event

Task event list:
Status      Times Type      Details                                     First Report Time
Last Report Time
SuccessfulCreate 1      Normal    Created pod: task-1-test-bingfasmial-1-59620029ac100038-
jkdpt 2022-06-24 16:37:20 2022-06-24 16:37:20
JobsComplete 1      Normal    Pod exits with success, the job is complete
2022-06-24 16:37:23 2022-06-24 16:37:23

Task instances list:
Name              PodIP      Node              RestartCount Request/
Limit(CPU) Request/Limit(Memory) CreateTime
task-1-test-bingfasmial-1-59620029ac100038-jkdpt 172.16.3.37 192.168.54.255
0 1/1 1G/1G 2022-06-24 16:37:20
```

● 获取作业某一task日志。

```
ai4s get job 550e8400-e29b-41d4-a716-446655440000 --log ./test/demo.log --task task-xxx-job-1
download the log of task task-lmx-job-1 successfully!
```

● 获取作业列表。

```
ai4s get job --status Failed --user-name ei_eihealth --create-from-time "2022-12-15 00:40:11" --
create-to-time "2022-12-17 00:40:11" --finish-from-time "2022-12-14 17:05:09" --finish-to-time
"2022-12-19 23:04:07" --labels "label1,lab_el-A"
--job-name h-err-1 --workflow-name herr --limit 1 --offset 1

job_id      job_name      tool_name tool_version tool_type status
user_name      create_time      finish_time      labels
```

```
8a6078d9-c307-11ed-a824-fa163e504fdd job-4127-01 new-01 wewe workflow
FAILED ei_eihealth_h00541446_01 2023-03-15 16:01:07 2023-03-15 16:02:51 label1,lab_el-A
```

9.4 删除指定作业

使用`delete`命令删除指定作业。

命令结构

```
ai4s delete job ID [flags]
```

表 9-3 参数说明

参数	简写	是否必选	描述
ID	无	是	作业ID（job-id）。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s delete job f17a3542-3f7c-11eb-868a-fa163e3ddba1
# 返回结果如下
delete job success! job id : f17a3542-3f7c-11eb-868a-fa163e3ddba1
```

9.5 重试作业

使用`retry`命令重试作业。

命令结构

```
ai4s retry job <job-id> [flags]
```

表 9-4 参数说明

参数	简写	是否必选	描述
job-id	无	是	作业ID。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s retry job f17a3542-3f7c-11eb-868a-fa163e3ddba1
# 返回结果如下
retry job successfully!
```

9.6 取消作业

使用`cancel`命令取消或强制终止作业。

命令结构

```
ai4s cancel job <job-id> [flags]
```

表 9-5 参数说明

参数	简写	是否必选	描述
job-id	无	是	作业ID。
--force	-f	否	强制操作，不进行询问提示。默认为false。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s cancel job f17a3542-3f7c-11eb-868a-fa163e3ddba1 -f
# 返回结果如下
cancel job successfully!
```

9.7 导出作业

使用`export`命令导出作业信息。

导出的作业信息会以zip文件格式保存到当前目录，zip文件中包含以下两个文件：

- jobDetails.zip：保存job的yaml文件
- jobInfo.csv：保存job信息

命令结构

```
# 使用job-id导出作业
ai4s export job --ids <job-id> [flags]
# 使用job-id文件导出作业
ai4s export job --file <job-id-file> [flags]
```

表 9-6 参数说明

参数	简写	是否必选	描述
--ids	-i	否	要导出的id列表，使用job-id导出作业时必选。

参数	简写	是否必选	描述
job-id	无	否	作业id。 导出多个job-id时，用分号（;） 隔开。 格式：id1;id2;id3。
--file	-f	否	要导出的id列表文件，使用job-id文件导出作业时必选。
job-id-file	无	否	要导出的id列表的文件位置。 文件内容格式：{"ids":"id1;id2;id3"}。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

- 使用job-id导出作业

```
ai4s export job -i f17a3542-3f7c...
# 返回结果如下
export job successfully!
```
- 使用job-id文件导出作业

```
ai4s export job -f /user/path/ids.txt
# 返回结果如下
export job successfully!
```

9.8 获取任务实例

使用get命令获取任务实例相关信息，如任务实例事件、任务实例详情。

命令结构

```
ai4s get task <task-name> --job-id <job-id> --instance-name <instance-name> [flags]
```

表 9-7 参数说明

参数	简写	是否必选	描述
task-name	无	是	任务名称。
--job-id	-j	是	作业id。
--instance-name	-n	是	实例名称。
--event	-e	否	该参数用于获取任务实例事件列表。不带该参数，默认输出任务实例详情（默认json格式）。

参数	简写	是否必选	描述
--file-path	-p	否	获取任务实例详情时，设置任务实例详情下载路径。
--yaml	-y	否	获取任务实例详情时，以yaml格式输出任务实例详情（默认json格式）。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

- 获取json格式实例详情

```
ai4s get task your-task -j your-job -n your-instance
# 返回结果如下
{
  "kind": "Pod",
  "apiVersion": "v1",
  "metadata": {
    ...
  }
}
```

- 获取事件

```
ai4s get task your-task -j your-job -n your-instance -e
# 返回结果如下
Index  Type  Count  Reason          Message          FirstTimestamp
LastTimestamp
1      Normal  1      Scheduled       Successfully assigned to      2022-05-24T10:04:55Z
2022-05-24T10:04:55Z
2      Normal  2      SuccessfulMountVolume  Successfully mounted volumes
2022-05-24T10:04:56Z  2022-05-24T10:07:07Z
3      Normal  1      Pulling         Pulling image      2022-05-24T10:04:56Z
2022-05-24T10:04:56Z
4      Normal  1      Pulled          Successfully pulled image    2022-05-24T10:07:02Z
2022-05-24T10:07:02Z
5      Normal  1      SuccessfulCreate      Created container      2022-05-24T10:07:06Z
2022-05-24T10:07:06Z
6      Normal  1      Started          Started container     2022-05-24T10:07:06Z
2022-05-24T10:07:06Z
```

- 获取yaml格式详情文件

```
ai4s get task your-task -j your-job -n your-instance -y -p ./a.yaml
# 返回结果如下
>ls ./a.yaml
a.yaml

> head a.yaml
apiVersion: v1
kind: Pod
metadata:
  annotations:
    ...
```

10 notebook 命令

10.1 创建 notebook

使用create命令创建notebook。

命令结构

```
ai4s create notebook <notebook-name> [flags]
```

表 10-1 参数说明

参数	简写	是否必选	说明
notebook-name	无	是	notebook名称。 取值范围[1,63]，仅支持小写字母、数字、中划线（-），开始只能是小写字母，结束只能是小写字母或数字。
--description	-d	否	描述。长度范围为[0,1024]。
--image	-i	是	notebook镜像，支持PY3或者指定空间中镜像（格式为SourceProjectName/ ImageName:TagName）。 说明 - 若选择PY3，则代表使用系统镜像创建 notebook； - 若选择SourceProjectName/ ImageName:TagName，则代表使用自定义镜像创建notebook。

参数	简写	是否必选	说明
--flavor	-f	否	flavor规格，格式为cpu:gpu:memory[:cpu_type:gpu_type]，不填默认为1:0:2，即cpu规格默认1C，范围[1,128]。gpu规格默认0，范围[0,16]。memory规格默认2G，范围[2,512]。cpu_type（选填）默认不填，默认值为X86，支持X86和ARMgpu_type（选填）默认不填，支持GPU和Snt9。
--storage-path	-s	是	notebook存储路径。包含本空间桶最多挂载6个，不包含本空间桶最多挂载5个。格式为ProjectName:/path，多个路径用分号(;)隔开。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

- 创建SYSTEM镜像类型的notebook

```
ai4s create notebook lmx-notebook-system -i PY3 -f 1:0:2 -s lmx-test-01:/path1;/lmx-test-02:/path2/
# 返回结果如下
create notebook successfully! notebook id is xxx
```
- 创建CUSTOMER镜像类型的notebook

```
ai4s create notebook lmx-notebook-customer -i lmx-test-01/image1:tag1 -f 1:0:2 -s lmx-test-02:/path1/
# 返回结果如下
create notebook successfully! notebook id is xxx
```

10.2 获取 notebook 信息

使用get命令获取notebook列表、详情。

命令结构

```
ai4s get notebook <notebook-id> [flags]
```

表 10-2 参数说明

参数	简写	是否必选	说明
notebook-id	无	否	notebook id，不填写时表示获取notebook列表，填写时表示获取notebook详情。
--url	-u	否	获取打开notebook的地址。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

● 获取notebook列表

```
ai4s get notebook
# 返回结果如下
Name      ID  Status  Container Image   Created      Updated
notebook-01 xxx Running mul-kernel-cpu-cp27 2021-02-01 11:11:27 2021-02-01 11:11:27
notebook-02 xxx Failed  mul-kernel-cpu-cp27 2021-02-01 11:11:27 2021-02-01 11:11:27
notebook-03 xxx Succeed mul-kernel-cpu-cp27 2021-02-01 11:11:27 2021-02-01 11:11:27
```

● 获取notebook详情

```
ai4s get notebook xxxx
# 返回结果如下
{
  "id": "439274bd-be09-4300...",
  "name": "a",
  "creator": "ei_eihealth_02",
  "url": "https://x.x.x.x/v1/0xxx/notebooks/xxx/services/notebook",
  "flavor": {
    "cpu_type": "ARM",
    "cpu": 1.0,
    "gpu": 0.0,
    "gpu_type": "Snt9",
    "memory": 2.0
  },
  "status": "Creating",
  "image": {
    "image_type": "SYSTEM",
    "image_info": {
      "source_project_name": "",
      "image_name": "eihealth-notebook",
      "image_tag": "cuda11.0-custom-v1.0.6",
      "profile": "PY3"
    }
  },
  "storages": [
    {
      "path": "/home/ai4s-user/work/wyq-test"
    }
  ],
  "create_time": "2022-06-30T11:25:43Z",
  "update_time": "2022-06-30T11:25:53Z",
  "failed_message": "pod failed for:Unschedulable, message:0/2 nodes are available: 2 Insufficient cpu.",
  "events": [
    {
      "type": "Warning",
      "count": 3,
      "reason": "FailedScheduling",

```

```
"message": "0/2 nodes are available: 2 Insufficient cpu.",  
"first_timestamp": "2022-06-30T11:25:43Z",  
"last_timestamp": "2022-06-30T11:25:46Z"  
}  
]  
}
```

- 获取打开notebook的地址信息

```
ai4s get notebook xxxx --url  
# 返回结果如下https://x.x.x.x/v1/0xxx/notebooks/xxx/services/notebook?token=xxxx
```

10.3 删除 notebook

使用**delete**命令删除notebook。

命令结构

```
ai4s delete notebook <notebook-id> [flags]
```

表 10-3 参数说明

参数	简写	是否必选	说明
notebook-id	无	是	notebook id。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s delete notebook xxxx  
# 返回结果如下  
delete notebook xxxx successfully!
```

10.4 编辑 notebook

使用**update**或**edit**命令更新notebook描述信息。

命令结构

```
ai4s update notebook <notebook-id> [flags] 或者 ai4s edit notebook <notebook-id> [flags]
```

表 10-4 参数说明

参数	简写	是否必选	说明
notebook-id	无	是	notebook id。
--description	-d	是	描述。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s update notebook xxxx --description update
# 返回结果如下
update the description of notebook xxxx successfully!
```

10.5 启动 notebook

使用start命令启动notebook。

命令结构

```
ai4s start notebook <notebook-id> [flags]
```

表 10-5 参数说明

参数	简写	是否必选	说明
notebook-id	无	是	notebook id。
--project	无	否	指定空间名。未填写则使用配置文件中的空间名。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s start notebook xxxx
# 返回结果如下
start notebook xxxx successfully!
```

10.6 停止 notebook

使用stop命令停止notebook。

命令结构

```
ai4s stop notebook <notebook-id> [flags]
```

表 10-6 参数说明

参数	简写	是否必选	说明
notebook-id	无	是	notebook id。

命令示例

本节以Windows为例介绍ai4s-toolkit的使用过程，Linux、macOS和Windows环境使用方法基本相同，可参考此章节。

```
ai4s stop notebook xxxx  
# 返回结果如下  
stop notebook xxxx successfully!
```