

医疗智能体

命令行工具

文档版本 04
发布日期 2023-04-27



版权所有 © 华为云计算技术有限公司 2024。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 什么是医疗智能体 eihealth-toolkit.....	1
2 获取并使用命令行工具 eihealth-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 eihealth-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.....	11
3 系统设置命令.....	13
3.1 获取系统标签.....	13
3.2 添加系统标签.....	13
3.3 删除系统标签.....	14
3.4 获取系统资源.....	14
3.5 购买计算资源.....	16
3.6 删除计算资源节点.....	17
3.7 修改计算资源.....	17
3.8 修改性能加速资源.....	18
3.9 查询消息及已完成作业保留设置.....	19
3.10 修改消息及已完成作业保留配置.....	19
3.11 获取供应商 logo 和名称设置.....	20
3.12 修改供应商 logo 和名称配置.....	20
3.13 获取系统配额信息.....	20
3.14 获取消息中心消息列表.....	21
4 项目管理命令.....	23
4.1 获取项目.....	23
4.2 切换项目.....	25
4.3 创建项目.....	26
4.4 更新项目.....	26
4.5 删除项目.....	27
4.6 转移项目.....	28
4.7 添加项目成员角色.....	28
4.8 修改项目成员角色.....	29
4.9 移除项目成员的角色.....	30
5 数据管理命令.....	31

5.1 列举对象.....	31
5.2 显示当前目录.....	34
5.3 切换路径.....	34
5.4 文件拷贝.....	35
5.5 创建子目录.....	40
5.6 上传数据.....	40
5.7 清理本地记录的上传文件.....	46
5.8 下载数据.....	46
5.9 删除数据.....	51
5.10 数据导入.....	53
5.11 创建归档.....	54
5.12 获取归档数据.....	55
5.13 删除归档.....	56
5.14 恢复归档.....	56
5.15 获取数据归档列表.....	57
5.16 修改归档区域.....	57
5.17 获取数据作业.....	58
5.18 删除数据作业.....	60
5.19 重试数据作业.....	61
5.20 取消数据作业.....	61
5.21 移动对象.....	62
5.22 增量同步.....	66
5.23 删除分段上传任务.....	72
5.24 修改权限.....	73
5.25 查看对象属性.....	74
6 数据库管理命令.....	76
6.1 创建数据库模板.....	76
6.2 获取数据库模板.....	76
6.3 删除数据库模板.....	80
6.4 导入数据库模板.....	80
6.5 创建数据库实例.....	81
6.6 获取数据库实例.....	82
6.7 删除数据库实例.....	84
6.8 引用数据库或者导入数据到指定数据库.....	85
6.9 修改数据库实例.....	86
7 镜像管理命令.....	88
7.1 标记镜像.....	88
7.2 上传镜像.....	89
7.3 下载镜像.....	90
7.4 查询镜像.....	90
7.5 导入镜像.....	91
7.6 更新镜像.....	92

7.7 删除镜像标签.....	92
8 应用管理命令.....	94
8.1 应用配置文件说明.....	94
8.2 创建应用.....	96
8.3 修改应用.....	96
8.4 查询应用.....	98
8.5 删除应用.....	100
8.6 导入应用.....	101
9 EIHealth 流程管理命令.....	103
9.1 流程配置文件说明.....	103
9.2 创建流程.....	104
9.3 修改流程.....	105
9.4 查询流程.....	106
9.5 删除流程.....	109
9.6 导入流程.....	109
10 EIHealth 作业管理命令.....	111
10.1 作业配置文件说明.....	111
10.2 启动作业.....	112
10.3 查询作业详情.....	115
10.4 删除指定作业.....	120
10.5 重试作业.....	120
10.6 取消作业.....	121
10.7 导出作业.....	121
10.8 获取任务实例.....	122
11 Nexflow 流程管理命令.....	124
11.1 创建流程.....	124
11.2 修改流程.....	125
11.3 查询流程详情.....	125
11.4 删除流程.....	127
12 Nextflow 作业管理命令.....	128
12.1 启动作业.....	128
12.2 查询作业.....	129
12.3 删除指定作业.....	132
12.4 重试作业.....	133
12.5 停止作业.....	133
12.6 获取任务信息.....	134
13 notebook 命令.....	136
13.1 创建 notebook.....	136
13.2 获取 notebook 信息.....	137
13.3 删除 notebook.....	138

13.4 编辑 notebook.....	139
13.5 启动 notebook.....	139
13.6 停止 notebook.....	140

1 什么是医疗智能体 eihealth-toolkit

医疗智能体命令行工具（eihealth-toolkit）是配套[EIHealth平台](#)，为EIHealth平台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EIHealth平台项目中数据、应用、流程和作业资源进行管理和使用。

eihealth-toolkit具备灵活性高、扩展性强：

1. 单一可执行文件，方便拷贝与安装。
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

eihealth-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、系统资源、系统配额信息、消息列表，查询和修改消息及完成作业保留配置，也可以获取和修改供应商logo、名称设置。
- **项目管理**
您可以使用该工具获取、切换、创建、更新、删除和转移项目，也可以添加、修改、移除项目成员角色。
- **数据管理**
您可以使用该工具查看、上传、下载、复制、导入、引入、创建和删除项目中的数据，创建、获取、删除、恢复归档数据，也可以对数据作业执行获取、删除、重试、取消操作。
- **数据库管理**
您可以使用该工具创建、获取、删除和导入数据库模板，创建、获取和删除数据库实例，也可以引用数据库。
- **镜像管理**
您可以使用该工具对镜像执行标记、上传、下载、查询、导入、更新和删除标签等操作。
- **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。

下载命令行工具 eihealth-toolkit

针对不同操作系统，eihealth-toolkit下载地址如下所示。

表 1-1 下载列表

支持平台	下载地址
Windows 64位	health-windows-x86_64.zip 、 health-windows-x86_64.zip.sha256
Linux ARM 64位	health-linux-aarch64.tar 、 health-linux-aarch64.tar.sha256
Linux AMD 64位	health-linux-x86_64.tar 、 health-linux-x86_64.tar.sha256
macOS	health-macOS-x86_64.tar 、 health-macOS-x86_64.tar.sha256

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EIHealth平台的项目、数据等资产。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health-windows-x86_64，它的校验文件下载链接则是health-windows-x86_64.zip.sha256。

详细的使用方法介绍请参见[获取并使用命令行工具eihealth-toolkit](#)。

2 获取并使用命令行工具 eihealth-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。

医疗智能体服务部署在华北-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。


获取平台 ID

平台ID与“命令行工具 > 步骤3 初始化配置”中的platform-id对应。

1. 登录医疗智能体平台。
2. 在右上角用户名中选择“个人设置”，获取平台ID。

图 2-3 平台 ID

系统信息

平台 ID e53fb 

2.2 步骤 2：下载并安装命令行工具 eihealth-toolkit

下载命令行工具 eihealth-toolkit

针对不同操作系统，eihealth-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	health-windows-x86_64.zip 、 health-windows-x86_64.zip.sha256
Linux ARM 64位	health-linux-aarch64.tar 、 health-linux-aarch64.tar.sha256
Linux AMD 64位	health-linux-x86_64.tar 、 health-linux-x86_64.tar.sha256
macOS	health-macOS-x86_64.tar 、 health-macOS-x86_64.tar.sha256

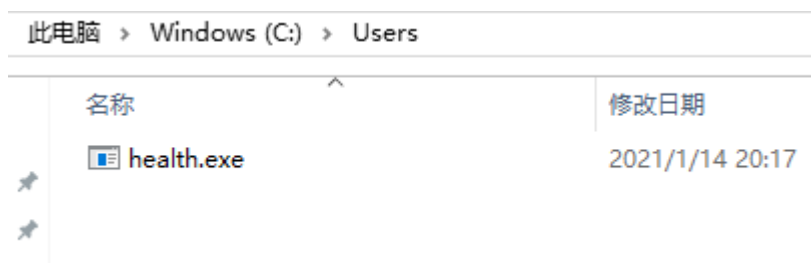
📖 说明

- 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EIHealth平台的项目、数据等资产。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health-windows-x86_64，它的校验文件下载链接则是health-windows-x86_64.zip.sha256。

安装 eihealth-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health.exe文件，health文件无需安装，放置在任一文件夹中即可。

图 2-4 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health命令，即可使用。

如果cmd窗口显示目录不是health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 2-5 客户端

```
C:\Users\>health
A simple command client for ei health.

Usage:
  health [flags]
  health [command]

Available Commands:
  add      add a member...
  cancel   cancel or forcibly terminate the job
  cd        change project directory
  config   configure your config file ...
  cp        copy project objects
  create   create a app,workflow,job,template,database...
  delete   delete a app, workflow, job, template, database...
  docker   push, pull or list images...
  download download objects from project directory to local directory
  edit     edit a app,workflow...
  get      get project, app, workflow, template, database ...
  help     Help about any command
  import   import template, data of database, app, workflow...
  ls       list project objects
  mkdir    create new project directory
  pwd      print current work directory
  retry    run the job or notebook again
  rm       delete project objects
  switch   switch project ...
  transfer transfer project...
  upload   upload objects from local directory to project directory
  version  show cli version

Flags:
  -h, --help  help for health

Use "health [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health文件，health文件无需安装，放置在任一文件夹中即可。
 - b. 假设health存放在/home/user-name/test/client目录，请使用cd命令进入health所在目录。

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

图 2-6 客户端

```
(base) root@ :/home/: # cd /home/ /test/client
(base) root@ :/home/: \test/client# ./health
A simple command client for ei health.

Usage:
health [flags]
health [command]

Available Commands:
abort      abort multipart uploads
add        add a member...
cancel     cancel or forcibly terminate the job
cd         change project directory
chattri    set object properties
clear      delete the checkpoint dir
completion Generate the autocompletion script for the specified shell
config     configure your config file ...
cp         copy project objects
create     create a app,workflow,job,template,database...
delete     delete a app, workflow, job, template, database...
docker     push, pull or list images...
download   download objects from project directory to local directory
edit       edit a app,workflow...
export     export job...
get        get project, app, workflow, template, database ...
help       Help about any command
import     import template, data of database, app, workflow...
ls         list project objects
mkdir      create new project directory
mv         move project objects
nextflow   create, edit or get nextflow...
pwd        print current work directory
restore    restore the archive...
retry      run the job or notebook again
rm         delete project objects
start      start notebook ...
stat       show the properties of a bucket or an object
stop       stop notebook ...
switch     switch project ...
sync       synchronize objects from the source to the destination
transfer   transfer project...
upload     upload objects from local directory to project directory
version    show cli version

Flags:
-h, --help help for health

Use "health [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health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health所在目录，可以使用./health命令使用命令行工具。
- 如果当前目录不是health所在目录，需要使用绝对路径。如当前目录为/opt，假设health存放在/root/health-toolkit/下，需要指定/root/health-toolkit/health路径进行使用。
- 如果无法运行，提示Permission denied，使用chmod 755 health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eihealth-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health config add命令进行初始化配置。

```
health 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	平台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。
--health-endpoint	-e	否	EiHealth终端节点名称，请在 地区与终端节点 中获取。
--swr-endpoint	-t	否	SWR镜像仓库地址。 获取方式： 1. 登录容器镜像服务管理控制台。 2. 单击界面右侧“登录指令”，获取内网登录指令末尾的SWR镜像仓库地址。例如 100.78.15.50:20202。
--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healthcli.log文件。 路径设置格式： <ul style="list-style-type: none"> 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”。
--obs-endpoint	-o	否	OBS终端节点名称，请在 地区与终端节点 中获取。

参数	简写	是否必选	说明
--obs-install-path	-q	否	设置obsutil安装路径，默认安装在当前运行目录。 设置时，该路径必须为obsutil运行文件名，如/home/path/obsutil、/home/path/obsutil-1.1.1
--obs_down_load_url	-D	否	obsutil下载链接，obsutil将下载到obs-install-path上。 参数有改动时才会触发下载。 下载链接的内容可以是zip、tar.gz文件、二进制文件，如果是压缩文件，文件夹内的obsutil必须命名为obsutil（和obsutil官方链接保持一致）。
--force	-f	否	强制操作。如果下载obsutil时，指定的obs-install-path上已经有同名文件，不带-f时会提示用户，带上-f会直接覆盖原文件。

命令示例

- 初始化配置**

```
health config add -d xxx -u xxx -w xxx -i xxx -r cn-north-4 -o obs.cn-north-4.myhuaweicloud.com -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

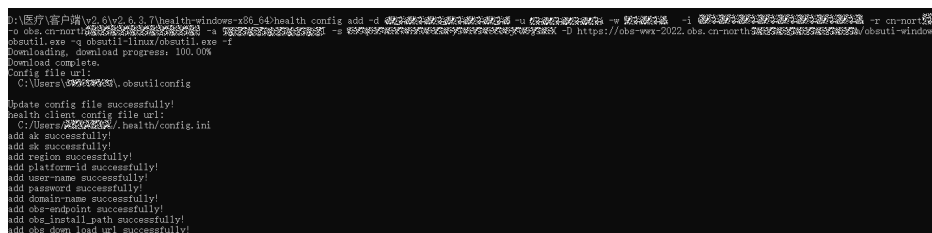
```
add ak successfully!
add sk successfully!
add region successfully!
add platform-id successfully!
add user-name successfully!
add password successfully!
add domain-name successfully!
add obs-endpoint successfully!
add obs_install_path successfully!
add obs_down_load_url successfully!
add iam-endpoint successfully!
add health-endpoint successfully!
add swr-endpoint successfully!
```
- 初始化配置**

```
health config add -d xxx -u xxx -w xxx -i xxx -r cn-north-4 -a xxx -s xxx -D https://xxx -q 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 obs_install_path successfully!
add obs_down_load_url successfully!
```

图 2-7 命令示例



```

D:\医疗客户端> .\obs-6.2.1\health-windows> health config add -d xxx -u xxx -w xxx -i xxx -r cn-north-4 -o obs.cn-north-4.myhuaweicloud.com -a
xxx -s xxx -D https://xxx -q xxx --log-path D:\log\eihealth.log -f -m xxx -e xxx -t xxx
Downloading. Download progress: 100.00%
Download complete.
Config file url:
C:\Users\ADMINI~1\AppData\Local\Temp\obsutilconfig
Update config file successfully!
health client config file url:
C:\Users\ADMINI~1\AppData\Local\Temp\obsutilconfig\health/config.ini
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 obs-endpoint successfully!
add obs_install_path successfully!
add obs_down_load_url successfully!
  
```

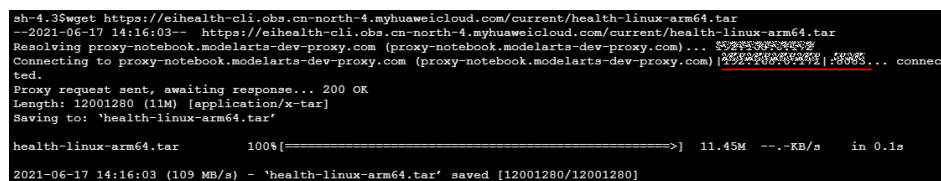
- 初始化配置并设置日志存储路径
health config add -d xxx -u xxx -w xxx -i xxx -r cn-north-4 -o obs.cn-north-4.myhuaweicloud.com -a xxx -s xxx -D https://xxx -q xxx --log-path D:\log\eihealth.log -f -m xxx -e xxx -t xxx
执行成功返回结果如下
add ak successfully!
add sk successfully!
add region successfully!
add platform-id successfully!
add user-name successfully!
add password successfully!
add domain-name successfully!
add obs-endpoint successfully!
add obs_install_path successfully!
add obs_down_load_url successfully!
add log-path successfully!
add iam-endpoint successfully!
add health-endpoint successfully!
add swr-endpoint successfully!
- 初始化配置并设置日志存储路径
health 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\eihealth.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 obs_install_path successfully!
add obs_down_load_url successfully!
add log-path successfully!
- 清空配置请执行health config clear命令
- 在Notebook中使用命令行工具

在EIHealth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。

- 打开Notebook，并选择Terminal，打开Notebook的命令行界面。
- 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/health-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)...
Connecting to proxy-notebook.modelarts-dev-proxy.com (proxy-notebook.modelarts-dev-proxy.com)|192.168.1.100|:8080... connec
ted.
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]
  
```

- 初始化配置时，添加--http-proxy命令，用于配置代理。
health 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“.health”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、当前项目等信息。
- 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health 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eihealth-toolkit的使用过程，Linux和macOS环境使用方法基本相同，可参考。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EIHealth平台。

详细的操作命令请参见其他章节。

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

```
C:\Users\ >health --help
A simple command client for ei health.

Usage:
  health [flags]
  health [command]

Available Commands:
  add      add a member...
  cancel   cancel or forcibly terminate the job
  cd       change project directory
  config   configure your config file ...
  cp       copy project objects
  create   create a app,workflow,job,template,database...
  delete   delete a app, workflow, job, template, database...
  docker   push, pull or list images...
  download download objects from project directory to local directory
  edit     edit a app,workflow...
  get      get project, app, workflow, template, database ...
  help     Help about any command
  import   import template, data of database, app, workflow...
  ls       list project objects
  mkdir    create new project directory
  pwd      print current work directory
  retry    run the job or notebook again
  rm       delete project objects
  switch   switch project ...
  transfer transfer project...
  upload   upload objects from local directory to project directory
  version  show cli version

Flags:
  -h, --help  help for health

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

- 执行操作命令，获取项目信息。
执行**health get project**命令查询当前账号下所拥有的项目和项目信息。Linux系统下，需添加./指定当前路径。

```
C:\Users\ >health get project
```

Project Name	Project Id	Owner	Role	Status	Size	Created	Modified
ixia	3a19c	ei_eihea	Owner	ACTIVE	13.49GB	2021-10-14 17:52:14	2021-10-14 17:52:15
ei-lab-test	4e91b	ei_eihea	Owner	ACTIVE	25.59GB	2021-07-05 15:53:18	2021-10-22 15:59:29
test-deletion	761fd	ei_eihea	Owner	ACTIVE	29.38GB	2021-07-02 18:14:17	2021-08-27 10:41:57
wx-project-viewer	4306f	ei_eihea	Owner	ACTIVE	0.00B	2021-05-10 11:00:38	2021-10-25 16:50:18
wx-project-uploader	ca57f	ei_eihea	Owner	ACTIVE	0.00B	2021-05-10 10:59:53	2021-10-08 15:46:50
wx-project-developer	ca92d	ei_eihea	Owner	ACTIVE	938.00B	2021-05-10 10:59:22	2021-10-08 15:46:34
wx-project-admin	8258e	ei_eihea	Owner	ACTIVE	362.72GB	2021-05-10 10:58:30	2021-10-14 14:33:11
wx-project-owner	737e3	wow-test	Administrator	ACTIVE	0.00B	2021-05-08 11:30:42	2021-11-18 10:32:43
test-wq-0507	31079	ei_eihea	Owner	ACTIVE	15.44GB	2021-05-07 16:16:41	2021-09-16 09:34:27
test-zh-0507	f8811	ei_eihea	Owner	ACTIVE	146.06GB	2021-05-07 15:54:48	2021-09-16 11:25:22

📖 说明

- 使用数据、应用、流程、作业命令时，需先使用**switch**命令进入待操作的项目中。使用逻辑与EIHealth平台一致，进入项目，再对项目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

```
health get label [flags]
```

命令示例

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

```
health 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

```
health 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

```
health create label name -d xxx
{
  "id": "b84ff2ce-0e51-11ed-9bd3-fa163e504fdd"
}
create label hl1 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

```
health 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

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

3.4 获取系统资源

使用get命令获取购买的计算资源、性能加速资源、数据库资源、存储资源列表。

命令结构

```
health 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computing、performance、database、storage，分别代表计算资源、性能加速资源、数据库资源和存储资源。

参数	简写	是否必选	说明
--action	-a	否	查询计算节点方式。 取值范围： <ul style="list-style-type: none"> get-labels: 获取计算资源标签列表。get-labels要和--node一起组合使用。 list: 获取计算资源列表。 flavors: 获取可用区下的计算节点规格列表。 默认值: list。
--node	-n	否	计算节点id。设置了--label后使用，获取某个计算资源节点下的标签列表。
--zone	-z	否	可用区id。如: cn-north-7c。

命令示例

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

- 获取计算资源列表

```
health get resource -t computing
# 执行成功返回结果如下
```

```
Id                               Name                               Data Disk           Status
Specifications                   Billing Mode
Purchased                         Schedulable
8a178a22-38ca-4707-bab3-e12324cdc96d health-ef0ec0fb-4022-46c9-898e-
fbd440b44c09                       eihealth.cpu.spec.cpu4.xlarge.2   Pay-per-use        SAS|200GB
ACTIVE                             2024-01-02 17:30:28             true
```

- 获取性能加速资源列表

```
health get resource -t performance
# 执行成功返回结果如下
```

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

- 获取数据库资源列表

```
health get resource -t database
# 执行成功返回结果如下
```

```
Name                               Specifications                   Billing Mode Usage
Data Disk Status Purchased
医疗智能体数据库标准版 200 GB                       eihealth.dbm.spec.standard.200g(4vCPUS|8GB)
Pay-per-use 2.36GB 40GB NORMAL 2023-12-27 16:07:12
```

- 获取存储资源列表

```
health get resource -t storage
# 执行成功返回结果如下
```

```
Name                               Specifications                   Billing Mode Usage Purchased
EIHealthSingleAZStoragePackage eihealth.storage.spec.singleaz xxx 4.06TB 2021-09-10
20:58:03
```

- 获取计算资源节点标签

```
health get resource -t computing -a get-labels -n 9b1727a3-2102-4d92-95b9-7eaa86b398ac

1
111
123
copy-in
```

- 获取计算资源规格列表

```
health get resource -t computing -a flavors -z cn-north-7c

Flavor Name          vCPUs | Memory      Assured/Maximum Bandwidth  Packets per
Second(PPS)         Sold Out
eihealth.cpu.spec.cpu1.8xlarge.2  32vCPUS/64GB      8/15 Gbit/s                2600000
pps                  true
eihealth.cpu.spec.cpu1.8xlarge.4  32vCPUS/128GB    8/15 Gbit/s                2600000
pps                  true
eihealth.cpu.spec.cpu1.15xlarge.2  60vCPUS/128GB    16/16 Gbit/s               5000000
pps                  true
eihealth.cpu.spec.cpu1.15xlarge.4  60vCPUS/256GB    16/16 Gbit/s               5000000
pps                  true
eihealth.cpu.spec.cpu3.xlarge.2    4vCPUS/8GB       2.4/8 Gbit/s                800000
pps                  false
eihealth.cpu.spec.cpu3.2xlarge.2   8vCPUS/16GB      4.5/15 Gbit/s              1500000
pps                  false
eihealth.cpu.spec.cpu3.4xlarge.2   16vCPUS/32GB     9/20 Gbit/s                2800000
pps                  true
eihealth.cpu.spec.cpu3.8xlarge.2   32vCPUS/64GB     18/30 Gbit/s               5500000
pps                  true
eihealth.cpu.spec.cpu3.8xlarge.4   32vCPUS/128GB    18/30 Gbit/s               5500000
pps                  true
```

3.5 购买计算资源

按需方式购买计算资源。

命令结构

```
health create compute-resources [flags]
```

表 3-4 参数说明

参数	简写	是否必选	描述
--code	-s	是	计算节点规格。
--data-disk-spec-code	-d	否	额外数据盘规格。
--extra-data-disk-size	-e	否	额外数据盘大小。 单位：G。 默认值：100。
--zone	-z	是	可用区id。如：cn-north-7c。
--purchased-quantity	c	否	购买节点数。 默认值：1。

命令示例

- 购买计算节点

```
health create compute-resources -s eihealth.cpu.spec.cpu3.xlarge.2 -z cn-north-7c
purchase compute-resource successful
```

3.6 删除计算资源节点

删除计算资源节点。

命令结构

```
health delete compute-resources <node id>
```

表 3-5 参数说明

参数	简写	是否必选	描述
node id	不涉及	是	计算节点id。

命令示例

- 删除计算节点

```
health delete compute-resources 1f6d8d7a-0709-44ec-b464-7c5c00eb9740
delete compute node:1f6d8d7a-0709-44ec-b464-7c5c00eb9740 success
```

3.7 修改计算资源

修改计算资源节点。

命令结构

```
health edit compute-resources [flags]
```

表 3-6 参数说明

参数	简写	是否必选	描述
--force	-f	否	是否强制操作。
--labels	-l	否	需要添加或删除的标签列表。json数组格式。 如: -l "[\"h1\", \"h2\"]"
--node	-n	是	计算资源节点id

参数	简写	是否必选	描述
-- operation- type	-t	是	对计算资源的操作: [add-labels \ delete-labels \ stop \ reboot \ start \ schedulable \ non-schedule]。add-labels: 添加计算节点标签。delete-labels: 删除计算节点标签。stop: 停止节点。reboot: 重启节点。start: 启动节点。schedulable: 设置节点可被调度。non-schedule: 设置节点不可被调度

命令示例

- 删除计算节点标签**

```
health edit compute-resources -t delete-labels -l "[h1\","h2\]" -n 9b1727a3-2102-4d92-95b9-7eaa86b398ac
```

edit compute resources using the delete-labels mode is successful
- 添加计算节点标签**

```
health edit compute-resources -t add-labels -l "[h1\","h2\]" -n 9b1727a3-2102-4d92-95b9-7eaa86b398ac
```

the edit compute-resources using add-labels mode is successful
- 设置节点可调度**

```
health edit compute-resources -t schedulable -n 9b1727a3-2102-4d92-95b9-7eaa86b398ac
```

the edit compute-resources using schedulable mode is successful
- 停止计算节点**

```
health edit compute-resources -t stop -n 9b1727a3-2102-4d92-95b9-7eaa86b398ac
```

The stop operation is started successfully...

3.8 修改性能加速资源

修改性能加速资源节点。

命令结构

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

表 3-7 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。
-- schedulab le	-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


```
health 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!
```

3.9 查询消息及已完成作业保留设置

使用`get`命令查询消息及已完成作业保留数目。

命令结构

```
health get retention [flags]
```

命令示例

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

```
health get retention
# 执行成功返回结果如下
Name          Retain Number
message       500000

Name          Retain Number
completed-jobs 6000000
```

3.10 修改消息及已完成作业保留配置

使用`update`或`edit`命令修改消息及已完成作业保留数目。

命令结构

```
health update retention <retention-name> [flags] 或者 health edit retention <retention-name> [flags]
```

表 3-8 参数说明

参数	简写	是否必选	说明
retention-name	无	是	名称，支持message或者completed-jobs。
--number	-n	否	保存条数，范围：10000-10000000。

命令示例

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

- 修改消息的保留数目

```
health edit retention message --number 10000
# 执行成功返回结果如下
set the retention of message successfully!
```
- 修改已完成作业的保留数目

```
health edit retention completed-jobs --number 10000
# 执行成功返回结果如下
set the retention of completed-jobs successfully!
```

3.11 获取供应商 logo 和名称设置

使用get命令获取供应商logo和名称配置，默认将供应商logo图片下载到本地路径。

命令结构

```
health 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eihealth-toolkit的使用过程，Linux和macOS环境使用方法基本相同，可参考。

```
health 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!
```

3.12 修改供应商 logo 和名称配置

使用update或edit命令修改供应商logo和名称配置。

命令结构

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

表 3-9 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eihealth-toolkit的使用过程，Linux和macOS环境使用方法基本相同，可参考。

- 设置供应商名称

```
health 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

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

3.13 获取系统配额信息

使用get命令获取系统配额信息。

命令结构

```
health get quota [flags]
```

命令示例

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

```
health get quota
# 执行成功返回结果如下
Name          Used   Total  Unit
Total projects 27     100   NUM
Applications per project --     500   NUM
Backups per project --     1000  NUM
Instances per project --     500   NUM
Images per project --     500   NUM
Notebooks per project --     10    NUM
Templates per project --     500   NUM
Workflows per project --     500   NUM
Total users    31     100   NUM
Projects per user --     10    NUM
```

3.14 获取消息中心消息列表

使用get命令获取消息中心消息列表。

命令结构

```
health get message [flags]
```

表 3-10 参数说明

参数	简写	是否必选	说明
--message-type	-m	否	消息类型，根据消息类型筛选消息列表。
--resource-type	-r	否	资源类型，根据资源类型筛选消息列表。
--operator	-e	否	操作人，根据操作人筛选消息列表。
--project-name	-p	否	项目名称，根据项目名称筛选消息列表。
--status	-s	否	消息状态，根据消息状态筛选消息列表。
--limit	-l	否	返回条数，默认20，范围0-1000。
--offset	-o	否	偏移量，默认0，范围0-100000000。

命令示例

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

```
health get message --limit 5
# 执行成功返回结果如下
```

Resource Name	Resource Type	Message Type	Project Name	Status	Operator	Operated
lmx-project-01	system	PROJECT_DELETE xxx		SUCCEED	lmx-test-01	2021-02-01 11:11:27
lmx-job-01	job	JOB_STATUS xxx		CANCELLED	lmx-test-01	2021-02-01 11:11:27
lmx-database-01	data	DATABASE_IMPORT xxx		PROCESSING	lmx-test-01	2021-02-01 11:11:27
lmx-data-01	data	SUBSCRIBE_DATA xxx		FAILED	lmx-test-01	2021-02-01 11:11:27
lmx-data-02	data	DELETE_DATA xxx		SUCCEED	lmx-test-01	2021-02-01 11:11:27

4 项目管理命令

4.1 获取项目

使用`get`命令获取当前用户有权限访问的所有项目列表，以及项目信息，包含项目名称、ID、所有者、项目角色、存储大小、状态、创建时间、更新时间。

在使用该命令前，您需要通过平台[创建项目](#)。

命令结构

```
health 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	项目名称，在EIHealth平台创建项目时，由用户填写。 <ul style="list-style-type: none">指定具体项目名称时，返回此项目的详情信息。为空时，获取当前用户有权限访问的项目列表。
--current	-c	否	同时使用project-name和--current，查询当前所在的项目详情信息。
--policy	-p	否	查看项目的数据权限策略。

命令示例

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

- 不指定参数。
执行`health get project`命令获取当前用户有权限访问的项目列表。

```
C:\Users\ >health get project
```

Project Name	Project Id	Owner	Role	Status	Size	Created	Modified
kixix	3a19c	ei_eihe	Owner	ACTIVE	13.476B	2021-10-14 17:52:14	2021-10-14 17:52:15
ei-lab-test	4e818	ei_eihe	Owner	ACTIVE	25.896B	2021-07-05 15:53:18	2021-10-22 15:59:29
test-deletion	761f4	ei_eihe	Owner	ACTIVE	29.386B	2021-07-02 18:14:17	2021-08-27 10:41:57
wwx-project-viewer	43b6f	ei_eihe	Owner	ACTIVE	0.08B	2021-05-10 11:00:38	2021-10-25 16:50:18
wwx-project-uploader	ca57f	ei_eihe	Owner	ACTIVE	0.08B	2021-05-10 10:59:53	2021-10-08 15:46:58
wwx-project-developer	ca92a	ei_eihe	Owner	ACTIVE	939.88B	2021-05-10 10:59:22	2021-10-09 15:46:34
wwx-project-admin	8258e	ei_eihe	Owner	ACTIVE	362.726B	2021-05-10 10:58:30	2021-10-14 14:33:11
wwx-project-owner	737e3	wwx-test	Administrator	ACTIVE	0.08B	2021-05-08 11:30:42	2021-11-18 10:32:43
test-zh-0507	31879	ei_eihe	Owner	ACTIVE	15.846B	2021-05-07 16:16:41	2021-09-16 09:24:27
test-zh-0507	f8811	ei_eihe	Owner	ACTIVE	146.866B	2021-05-07 15:54:48	2021-09-16 11:25:22

- 指定“project-name”参数。
例如查看名为“test-zh-0507”项目信息，执行health get project test-zh-0507命令。

```
C:\Users\ >health get project test-zh-0507
```

```
{
  "id": " ",
  "name": "test-zh-0507",
  "creator": "ei_eiheal ",
  "role": "Owner",
  "roles": [
    {
      "role_type": "Administrator",
      "users": [
        {
          "id": " ",
          "name": "wwx-test-002"
        }
      ]
    },
    {
      "role_type": "Owner",
      "users": [
        {
          "id": " ",
          "name": "ei_eiheal "
        }
      ]
    }
  ],
  "size": 156829437924,
  "status": "ACTIVE",
  "tags": [
    "-123-",
    "m6ayhe54y96hbzyvafu3tyxqsvumnyl",
    "test-biaoqian",
    "1"
  ],
  "create_time": "2021-05-07T07:54:48Z",
  "update_time": "2021-09-16T03:25:22Z"
}
```

- 指定“--current”和“--policy”参数。
先执行health switch project test-zh-0507，切换到“test-zh-0507”项目，再执行health get project --current --policy命令，查询当前所在项目的详细信息、数据权限策略。

```
C:\Users\ >health get project --current --policy
{
  "id": " ",
  "name": "test-zh-0507",
  "creator": "ei_eihealth ",
  "role": "Owner",
  "roles": [
    {
      "role_type": "Administrator",
      "users": [
        {
          "id": " ",
          "name": "wx-test-002"
        }
      ]
    },
    {
      "role_type": "Owner",
      "users": [
        {
          "id": " ",
          "name": "ei_eihealth "
        }
      ]
    }
  ],
  "size": 156829437924,
  "status": "ACTIVE",
  "tags": [
    "-123-",
    "m6ayhe54y96bhbzyvafu3tyxqsvumny1",
    "test-biaoqian",
    "1"
  ],
  "create_time": "2021-05-07T07:54:48Z",
  "update_time": "2021-09-16T03:25:22Z"
}
{
  "data_delete": true,
  "data_download": true,
  "data_encrypted": false,
  "data_share": true
}
```

4.2 切换项目

使用switch命令切换到指定的项目中。

使用该命令前，您需要先创建项目，再使用switch命令进入该项目。

命令结构

```
health 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	需要进入的项目名称，名称在创建项目时由用户填写。

命令示例

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

执行如下命令，切换到名为“lmx-test-01”的项目。

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

4.3 创建项目

使用**create**或**submit**命令创建项目，并可设置项目级数据权限策略。

命令结构

```
health create project <project-name> [flags] 或 health 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	项目名称。
--description	-d	否	项目描述。
-tags	-t	否	项目标签，多个标签使用分号(;)分隔。
--policy	-p	否	设置项目级数据权限策略。 用数字0和1表示关闭和打开，以data-share、data-download、data-delete、data-encrypted的顺序排列，例如全部打开则为1111，全部关闭则为0000，默认1110。

命令示例

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

执行如下命令，创建项目。

```
health create project demo-project -d description -t 'tag1;tag2;tag3' -p 0101
# 执行成功返回结果如下
create project successfully! project id is xxx.
```

4.4 更新项目

使用**update**或**edit**命令更新项目描述信息、标签或状态。

命令结构

```
health update project <project-name> [flags] 或 health 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	项目名称，不填时默认为当前所在项目。

参数	简写	是否必选	说明
--description	-d	否	项目描述。 只有项目所有者和管理员能修改。
--tags	-t	否	项目标签，多个标签使用分号(;)分隔。 只有项目所有者和管理员能修改。
--policy	-p	否	设置项目级数据权限策略。只有项目所有者能修改。 用数字0和1表示关闭和打开，以data-share、data-download、data-delete、data-encrypted的顺序排列，例如全部打开则为1111，全部关闭则为0000，默认1110。
--status	-s	否	修改项目的状态，支持ACTIVE、INACTIVE和TO_BE_DELETED，其中只有ACTIVE和TO_BE_DELETED状态项目能被修改为INACTIVE，只有INACTIVE和TO_BE_DELETED状态项目能被修改为ACTIVE，且只有项目所有者能冻结、解冻、恢复。

命令示例

- 更新项目描述和标签信息。
health update project demo-project --description description --tags 'a;b;c'
执行成功返回结果如下
edit project successfully!
- 修改项目级数据权限策略。
health update project demo-project --policy 1111
执行成功返回结果如下
edit project successfully!
- 修改项目状态为ACTIVE。
health update project demo-project --status ACTIVE
执行成功返回结果如下
edit project successfully!

4.5 删除项目

使用**delete**命令删除项目。项目的所有者有权限执行该操作。

为了防止项目误删，执行项目删除命令后工具会提示用户进行二次确认输入要删除的项目，只有二次输入的项目名称和命令中的项目名称一致，命令行工具才会执行删除动作。

命令结构

```
health 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	项目名称，不填时默认为当前所在项目。

命令示例

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

执行以下命令，删除项目demo-project。

```
health delete project demo-project
# 执行命令后，二次确认输入要删除的项目
Are you sure to delete the project, please input the project name to be deleted again to confirm:
demo-project
# 再次手动输入要删除的项目，返回结果如下
delete project successfully! The project will retain for seven days and you can restore it before perform a
real delete.
```

4.6 转移项目

使用transfer命令转移项目给项目admin成员。项目的所有者有权限执行该操作。

命令结构

```
health 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	项目名称，不填时默认为当前所在项目。
--user	-u	是	转移用户名称。

命令示例

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

```
health 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 添加项目成员角色

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

命令结构

```
health 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	项目名称，不填时默认为当前所在项目。
--role	-r	是	用户项目角色，只支持Administrator, Developer, Uploader或者Viewer。

命令示例

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

```
health 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 修改项目成员角色

使用**update**或**edit**命令修改项目成员角色。项目的所有者和管理员有权限执行该操作。

命令结构

```
health update member <member-name> [flags] 或 health 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	项目名称，不填时默认为当前所在项目。
--role	-r	是	用户项目角色，只支持Administrator, Developer, Uploader或者Viewer。

命令示例

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

```
health 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 移除项目成员的角色

使用`delete`命令删除项目成员的角色，并将成员移除项目。项目的所有者和管理员有权限执行该操作。

命令结构

```
health 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	项目名称，不填时默认为当前所在项目。

命令示例

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

```
health 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 列举对象

使用ls命令查询EIHealth项目中的对象，返回的对象名称按照字典序排列。同时，该命令支持显示引用的其他项目的数据。

在使用该命令前，需要使用switch命令进入待操作的项目，才可以执行数据相关操作，使用逻辑与EIHealth平台相同。

命令结构

```
health ls <path> [flags]
```

表 5-1 参数说明

参数	简写	是否必选	描述
path	无	否	路径名称。 <ul style="list-style-type: none"> 不指定path时，列举当前目录下的所有对象。 指定path为根目录时，列举根目录下本项目的对象和引用的对象。返回的查询结果中，引用的对象使用*标识。如果使用<project-name>:<path>方式列举，project-name必须是已经引用了的，要求path格式是绝对路径。如果path是文件夹，则后面要加/标识。当cd到其他项目路径时，如果要查看本项目内容，使用绝对路径即可，如：health ls /abc/。
--limit	-l	否	列举查询结果的最大个数，小于等于0表示列举所有结果，不设置时默认为最大值1000。引用数据会列举全部，不受该参数影响。
--simple	-s	否	以精简格式显示查询结果，返回结果只包含对象名。

参数	简写	是否必选	描述
--recursive	-r	否	递归列举本项目文件夹中的所有文件和子文件夹。
--v	-v	否	列举桶内多版本对象，列举结果包含最新版本的对象和历史版本的对象。
--marker	-M	否	列举桶内对象的起始位置，返回结果是按照字典序排序后该参数之后的所有对象，具体可参考 列举示例 。
--versionIdMarker	-V	否	列举桶内多版本对象的起始位置，返回结果是对象名和版本号按照字典序排序后该参数之后的所有对象。 必须与--v和--marker配合使用。 如果--versionIdMarker的版本号不包含在--marker返回结果对应的版本号中，则该参数无效。
--bf	-b	否	列举结果中字节数的显示格式。取值范围[human-readable, raw]。 如果未设置该参数，则列举结果中字节数的显示格式由配置文件中的humanReadableFormat参数决定。
--format	-F	否	指定以自定义格式打印列举结果。当前仅支持值[default]，指定列举结果在一行显示。
--a	-a	否	同时列举桶内对象和桶内分段上传任务。
--m	-m	否	列举桶内分段上传任务。
--e	-P	否	指定终端节点。
--i	-i	否	指定用户的AK。
--k	-k	否	指定用户的SK。
--t	-n	否	指定用户的securitytoken。
--uploadIdMarker	-u	否	列举桶内分段上传任务的起始位置，必须与--marker配合使用。返回结果是对象名和UploadId按照字典序排序后该参数之后的所有分段上传任务。

命令示例

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

假设“lmx-test-01”项目引用了来自“lmx-test-02”项目的数据，使用**health switch project lmx-test-01**命令进入“lmx-test-01”项目，使用**health ls**命令列举该项目下的对象。

- 指定“path”参数为根目录，查询根目录下所有对象。

```
health ls / -r
# 返回结果如下
# *表示引用的其他项目中的数据
*lmx-test-02:/test1.txt
*lmx-test-02:/test2.txt
*lmx-test-02:/src1/
*lmx-test-02:/src1/test1.txt

# Folder list为本项目中的数据
Folder list:
/src1/
/src2/

File list:
key          size  last_Modified
/test1.txt   6.57MB  2020/11/09 10:32:12
/test2.txt   6.57MB  2020/11/09 10:32:12
/test3.txt   6.57MB  2020/11/09 10:32:12
/test4.txt   6.57MB  2020/11/09 10:32:12
/test5.txt   6.57MB  2020/11/09 10:32:12
/test6.txt   6.57MB  2020/11/09 10:32:12
/src1/test7.txt 6.57MB  2020/11/09 10:32:12
/src2/test8.txt 6.57MB  2020/11/09 10:32:12

Total size of current file is: 52.56MB
Folder number is: 4
File number is: 8
```

- 指定“limit”参数，并只列举项目中的3个对象。

```
health ls --limit 3
# 返回结果如下
Folder list:
/src1/
/src2/

File list:
key          Size  LastModified
/test1.txt   6.57MB  2020/11/09 10:32:12

Total size of current file is: 6.57MB
Folder number is: 2
File number is: 1
```

- 指定“path”参数，列举某指定路径中的对象。

```
health ls /src2/ # 列举路径中的文件夹对象时，需使用/xxx/格式
# 返回结果如下
Folder list:
/src2/ # 路径中无文件夹时，返回为空

File list:
key          Size  LastModified
/src2/test8.txt 6.57MB  2020/11/09 10:32:12

Total size of current file is: 6.57MB
Folder number is: 1
File number is: 1
```

- 指定“simple”参数，以简略格式显示对象。

```
health ls --simple
# 返回结果如下
Folder list:
/src1/
/src2/

File list:
/test1.txt
/test2.txt
/test3.txt
/test4.txt
```

```
/test5.txt  
/test6.txt  
/src1/test7.txt  
/src2/test8.txt
```

```
Total size of current file is: 52.56MB  
Folder number is: 2  
File number is: 8
```

- 指定引用项目。

```
health ls lmx-test-02:/src1/ --simple
```

```
Start at 2022-08-23 08:28:43.3699574 +0000 UTC
```

```
Listing objects .
```

```
Folder list:
```

```
Object list:  
/src1/test1.txt
```

5.2 显示当前目录

使用**pwd**命令显示当前所在的EIHealth项目中的目录。同时，该命令支持显示引用项目的目录。

命令结构

```
health pwd
```

命令示例

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

```
health pwd  
# 若在本项目目录下，返回结果如下  
/src2/  
# 若在引用项目目录下，返回结果如下  
project_b:/src2/
```

5.3 切换路径

使用**cd**命令在EIHealth项目中切换数据路径。

命令结构

```
health cd <path>
```

命令示例

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

- 当前目录，不进行跳转。

```
health cd .
```

- 切换到上一级路径。

```
health cd ..
```


- 切换到指定的路径，如切换到src1。
health cd /src1
- 回到本项目根路径。
health cd /
- 回到本项目下DataSet路径。
health cd gene-assent:/DataSet/
- 切换到引用数据的路径，例如引用了gene-assent项目中的DataSet数据，使用health cd命令切换到该路径，并使用health ls名称查看详细数据。
当由引用数据的路径，切换到自己项目数据时，需指明具体的路径。
health cd gene-assent:/DataSet/
health ls

5.4 文件拷贝

使用cp命令将源路径中的文件拷贝到指定路径。同时，该命令支持拷贝其他项目中的数据。

命令结构

```
health cp <srcdir> <destdir> [flags]
```

表 5-2 参数说明

参数	简写	是否必选	说明
srcdir	无	是	源路径，支持本项目和其他项目的路径。当源路径为其他项目时，与EIHealth平台数据导入功能对应，即拷贝其他项目文件至本项目。
destdir	无	是	目的路径，只支持本项目路径。
--rename	-e	否	重命名，复制文件时该参数可选。
--recursive	-r	否	递归复制文件夹中所有文件和子文件夹，复制文件夹时该参数必选。
--force	-f	否	强制操作，不进行询问提示，复制文件夹时该参数可选。不加-f，会被询问是否执行操作，需选择yes或no。
--flat	-l	否	复制文件夹时，只复制文件夹下的所有内容，复制文件夹时该参数可选。
--update	-u	否	增量复制，设置该参数后，复制时会判断是否有同名文件，若有同名文件，则跳过不进行复制。
--fr	-R	否	复制对象时生成结果清单文件。
--dryRun	-y	否	测试模式运行，不执行实际的复制操作。

参数	简写	是否必选	说明
--crr	-c	否	<p>复制时使用客户端跨区域复制模式，以通过数据流的方式从源桶直接复制数据到目标桶，且两个桶可以是任意两个OBS服务的桶。</p> <p>若设置了该参数，必须确保更新了配置文件中客户端跨区域复制的相关配置信息，具体可参考更新配置文件。复制时源桶对应的配置信息为配置文件中的：akCrr/skCrr/tokenCrr/endpointCrr，目标桶对应的配置信息为配置文件中的：ak/sk/token/endpoint。</p> <p>开启该模式后，会占用上传和下载的双向带宽。</p>
--vlength	-v	否	<p>复制完成后，验证目标桶对象大小是否与源桶中对象的大小一致。</p> <p>必须与--crr参数配合使用。</p>
--vmd5	-M	否	<p>复制完成后，验证目标桶的MD5值是否与源桶中对象的MD5值一致。</p> <p>必须与--crr参数配合使用。源桶中对象必须包含元数据x-obs-md5chksum，否则会忽略MD5值校验。MD5值校验通过后，会将该值设置为目标对象元数据x-obs-md5chksum，用于下载或复制时校验MD5。</p>
--parallel	-p	否	<p>每个分段复制任务的最大并发数，默认为配置文件中的defaultParallels。</p>
--threshold	-t	否	<p>开启分段复制任务的阈值，单位：字节，默认为配置文件中的defaultBigfileThreshold。</p> <p>如果待复制的源对象小于该阈值，则使用直接复制模式；否则使用分段复制模式。直接复制模式不会产生断点记录文件，不支持断点续传。支持带容量单位配置，例如，配置1MB代表1048576字节。</p>
--versionId	-V	否	<p>复制单个对象时可指定的源对象版本号。</p>
--acl	-a	否	<p>复制对象时可指定的目标对象的预定义访问策略。</p> <p>支持的值：</p> <ul style="list-style-type: none"> • private：私有读写 • public-read：公共读 • public-read-write：公共读写 • bucket-owner-full-control：桶拥有者完全控制

参数	简写	是否必选	说明
--meta	-E	否	<p>复制对象时可指定的目标对象的自定义元数据。</p> <p>格式为： key1:value1#key2:value2#key3:value3。</p> <p>以上值表示复制对象完成后，桶内目标对象包含三组自定义元数据，分别为： key1:value1、key2:value2、key3:value3。</p>
--ps	-s	否	<p>每个分段复制任务的段大小，单位：字节，取值范围是100KB~5GB，默认为配置文件中的defaultPartSize。</p> <p>支持带容量单位配置，例如，配置1MB代表1048576字节。</p> <p>支持配置为auto，此时obsutil会根据源对象大小自动设置每个分段任务的段大小。</p>
--cpd	-C	否	<p>生成断点记录文件的文件夹，默认为运行obsutil命令的用户目录的子文件夹obsutil_checkpoint。</p> <p>每个分段复制任务会产生唯一对应的断点记录文件并保存至该文件夹的copy子文件夹下，分段任务执行成功后，对应的断点记录文件会被自动删除；分段任务执行失败或被中断后，下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。</p>
--jobs	-j	否	<p>批量复制时的最大并发数，默认为配置文件中的defaultJobs。</p>
--exclude	-x	否	<p>不包含源对象的匹配模式，如：*.txt。</p> <p>支持“*”匹配多个任意字符和“?”匹配单个任意字符，例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</p> <p>您可以使用“*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。如果待复制的对象名匹配该参数，则跳过该对象的复制。</p> <p>建议使用引号传递该匹配模式（macOS/Linux操作系统使用单引号，Windows操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。</p> <p>该匹配模式作用于对象全路径（含从根路径开始的对象前缀和对象名，例如，桶内对象路径为obs://bucket/src1/src2/test.txt，则对象的全路径为src1/src2/test.txt）。该匹配模式仅适用于对象名非“/”结尾的对象。</p> <p>支持指定多个exclude参数，如 --exclude *.xxx --exclude *.xxx。</p>

参数	简写	是否必选	说明
--include	-I	否	<p>包含源对象的匹配模式，如：*.jpg。</p> <p>支持“*”匹配多个任意字符和“?”匹配单个任意字符。</p> <p>您可以使用“*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。首先执行exclude的匹配规则，如果待复制的对象名不匹配exclude，则判断待复制的对象名是否匹配该参数，如果匹配则复制该对象，否则跳过该文件的复制。</p> <p>建议使用引号传递该匹配模式（macOS/Linux操作系统使用单引号，Windows操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。</p> <p>该匹配模式作用于对象全路径（含从根路径开始的对象前缀和对象名，例如，桶内对象路径为obs://bucket/src1/src2/test.txt，则对象的全路径为src1/src2/test.txt）。该匹配模式仅适用于对象名非“/”结尾的对象。</p> <p>支持指定多个include参数，如--include *.xxx --include *.xxx。</p>
--timeRange	-T	否	<p>复制对象时的时间段匹配模式，仅复制最后修改时间在该时间段内的对象。该匹配模式优先级低于对象名匹配模式：exclude和include，优先执行对象名匹配模式后才会执行该匹配模式。</p> <p>使用“time1-time2”，代表匹配的时间段，其中time1必须小于等于time2，格式为yyyyMMddHHmmss。时间格式支持自动补全，例如yyyyMMdd等价于yyyyMMdd000000，yyyyMM等价于yyyyMM01000000。</p> <p>使用“*-time2”，代表匹配最后修改时间在time2之前的所有文件，使用“time1-*”，代表匹配最后修改时间time1之后的所有文件。</p> <p>该匹配模式表示的时间是UTC时间。该匹配模式仅适用于对象名非“/”结尾的对象。</p>
--mf	-m	否	<p>设置名称匹配模式（include、exclude）和时间匹配模式（timeRange）对对象名以“/”结尾的对象也生效。</p>

参数	简写	是否必选	说明
--o	-o	否	生成结果清单文件的文件夹，命令执行完成后，会在该文件夹下生成结果清单文件（可能包含成功结果、失败结果和警告结果三个文件），默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。 结果清单文件命名规则：cp_{succeed failed warning}_report_时间_TaskId.txt。 单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024，可在配置文件中通过recordMaxLogSize和recordBackups分别配置。当涉及多文件/文件夹时，如需确认失败任务的具体错误信息，可参考结果清单文件夹下的失败结果清单文件“cp_failed_report_时间_TaskId.txt”及日志路径下的 日志文件 。
--e	-P	否	指定终端节点。
--i	-i	否	指定用户的AK。
--k	-k	否	指定用户的SK。
--t	-n	否	指定用户的securitytoken。
--recover	-z	否	待恢复复制任务结果清单文件的任务号。注意：恢复任务，不支持bucket-owner-full-control权限，因此，用于作业的数据abort后，不要使用恢复的方式，建议重新执行操作，或手动修改数据权限。

命令示例

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

📖 说明

- 列举路径中的文件夹对象时，需使用/xxx/格式，如示例中的/src1/和/src2/。
如果路径中带有特殊字符比如()之类的，运行的时候需要将整个路径用""括起来，或者用\将字符转义，例如：health cp "/src2/abs (1).txt" /src1/ 或者 health cp /src2/abs \ (1) .txt /src1/ 。
- cp命令源路径支持../写法。
- 指定具体文件进行拷贝，将src2中的abc.txt文件拷贝至src1中。
health cp /src2/abs.txt /src1/ -f
返回结果如下
copy project file successfully!
- 不指定具体文件进行批量拷贝，将src1中的所有文件拷贝至src2中。
health cp /src1/ /src2/ -r -f
返回结果如下
copy project file successfully!

- 拷贝其他项目中的文件，将project1项目中的zip文件拷贝至本项目src1中。拷贝其他项目文件，需要是该项目的成员，并具备数据操作权限，详细权限介绍请参见[项目成员和权限](#)。

```
health cp project1:/test/test.zip /src1/ -r -f
```

5.5 创建子目录

使用**mkdir**命令，在当前目录下创建子目录，此命令不支持在引用的项目中创建子目录。

命令结构

```
health mkdir <name>
```

表 5-3 参数说明

参数	简写	是否必选	说明
name	无	是	子目录的名称。
--e	-P	否	指定终端节点。
--i	-i	否	指定用户的AK。
--k	-k	否	指定用户的SK。
--t	-n	否	指定用户的securitytoken。

命令示例

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

若当前所在目录为src2，新增子目录dataset。

```
health mkdir dataset
# 创建成功不进行提示，可以使用health ls命令查看目录
```

5.6 上传数据

使用**upload**命令，将本地数据上传到EIHealth平台。该命令不支持将数据上传到引用目录。

最小可以上传**0Byte**的空文件或文件夹，最大可以上传**48.8TB**的单个文件。

数据在上传的过程中，受网络影响可能出现损坏，上传命令默认会在上传完成后，验证项目中数据的MD5值与本地数据的MD5值的一致性，以及验证项目中数据的大小与本地数据大小一致性。

命令结构

```
health upload <srcdir> <destdir> [flags]
```

表 5-4 参数说明

参数	简写	是否必选	说明
srcdir	无	是	源路径。
destdir	无	是	目的路径。
--rename	-e	否	重命名，上传文件时可选。
--recursive	-r	否	递归上传文件夹中所有的文件和子文件夹，上传文件夹时必选。
--force	-f	否	强制操作，不进行询问提示，上传文件夹时可选。
--flat	-l	否	上传文件夹时，只上传该文件夹下的所有内容，上传文件夹时可选。
--update	-u	否	增量上传操作，设置该参数后上传每个文件时会比对平台数据路径中的文件，仅在以下情况时上传数据： <ul style="list-style-type: none"> • 文件不存在。 • 待上传文件大小不与平台文件大小一致。 • 文件的最后修改时间不一致。
--parallel	-p	否	上传文件时，每个分段上传任务数的最大并发数，默认为5。取值范围[1,10]。
--jobs	-j	否	上传文件夹时，批量任务的最大并发数，默认为5。取值范围[1,10]。
--fr	-R	否	上传对象时生成结果清单文件。
--arcDir	-g	否	上传文件成功后的归档路径，上传成功后的文件会移动到该本地路径下。
--dryRun	-y	否	测试模式运行，不执行实际的上传操作。
--link	-K	否	上传软链接文件/文件夹指向的真实路径。 如果未指定该参数，而待上传的文件是一个软链接且该软链接指向的目标文件不存在，则Windows操作系统会产生“The system cannot find the file specified”的上传失败异常，macOS/Linux操作系统会产生“No such file or directory”的上传失败异常。 文件夹软链接不能形成环，否则上传会以panic的形式退出；如果不希望产生panic，请在配置文件中配置panicForSymbolicLinkCircle为false。
--vlength	-v	否	上传完成后，验证桶中对象的大小是否与本地文件大小一致。

参数	简写	是否必选	说明
--vmd5	-M	否	<p>上传完成后，验证桶中对象的MD5值是否与本地文件的MD5值一致。</p> <p>如果待上传的本地文件较大，使用该参数将会因为计算MD5而导致整体性能下降。MD5值校验通过后，会将该值设置为对象元数据x-obs-md5chksum，用于下载或上传时校验MD5。</p>
--threshold	-t	否	<p>开启分段上传任务的阈值，单位：字节，默认为配置文件中的defaultBigfileThreshold。</p> <p>如果待上传的文件小于该阈值，则使用直接上传模式；否则使用分段上传模式。直接上传模式不会产生断点记录文件，不支持断点续传。支持带容量单位配置，例如，配置1MB代表1048576字节。</p>
--acl	-a	否	<p>上传对象时可指定的目标对象的预定义访问策略。</p> <p>支持的值：</p> <ul style="list-style-type: none"> • private：私有读写 • public-read：公共读 • public-read-write：公共读写 • bucket-owner-full-control：桶拥有者完全控制
--meta	-E	否	<p>复制对象时可指定的目标对象的自定义元数据。</p> <p>格式为： key1:value1#key2:value2#key3:value3。</p> <p>以上值表示上传对象完成后，桶内目标对象包含三组自定义元数据，分别为： key1:value1、key2:value2、key3:value3。</p>
--ps	-s	否	<p>每个分段上传任务的段大小，单位：字节，取值范围是100KB~5GB，默认为配置文件中的defaultPartSize。</p> <p>支持带容量单位配置，例如，配置1MB代表1048576字节。</p> <p>支持配置为auto，此时obsutil会根据源对象大小自动设置每个分段任务的段大小。</p>

参数	简写	是否必选	说明
--cpd	-C	否	<p>生成断点记录文件的文件夹，默认为运行obsutil命令的用户目录的子文件夹obsutil_checkpoint。</p> <p>每个分段上传任务会产生唯一对应的断点记录文件并保存至该文件夹的copy子文件夹下，分段任务执行成功后，对应的断点记录文件会被自动删除；分段任务执行失败或被中断后，下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。</p>
--msm	-N	否	<p>开启多文件/文件夹上传模式，支持的值：[1, 2]。</p> <ul style="list-style-type: none"> • 如果msm=1则代表上传的URL是一组文件/文件夹列表（以英文逗号分隔）。如果文件/文件夹名本身包含英文逗号，请不要使用msm=1的模式。 • 如果msm=2则代表上传的URL是一个包含文件/文件夹列表的文件。 <p>多文件/文件夹上传时必选，如果没有设置--recursive参数，则列表中的文件夹不会被上传。</p>
--exclude	-x	否	<p>不包含源对象的匹配模式，如：*.txt。</p> <p>支持“*”匹配多个任意字符和“?”匹配单个任意字符，例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</p> <p>您可以使用“*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。如果待上传的对象名匹配该参数，则跳过该对象的上传。</p> <p>建议使用引号传递该匹配模式（macOS/Linux操作系统使用单引号，Windows操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。</p> <p>该匹配模式作用于对象全路径（含从根路径开始的对象前缀和对象名，例如，桶内对象路径为obs://bucket/src1/src2/test.txt，则对象的全路径为src1/src2/test.txt）。该匹配模式仅适用于对象名非“/”结尾的对象。</p> <p>支持指定多个exclude参数，如 --exclude *.xxx --exclude *.xxx。</p>

参数	简写	是否必选	说明
--include	-I	否	<p>包含源对象的匹配模式，如：*.jpg。</p> <p>支持“*”匹配多个任意字符和“?”匹配单个任意字符。</p> <p>您可以使用“*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。首先执行exclude的匹配规则，如果待上传的对象名不匹配exclude，则判断待上传的对象名是否匹配该参数，如果匹配则上传该对象，否则跳过该文件的上传。</p> <p>建议使用引号传递该匹配模式（macOS/Linux操作系统使用单引号，Windows操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。</p> <p>该匹配模式作用于对象全路径（含从根路径开始的对象前缀和对象名，例如，桶内对象路径为obs://bucket/src1/src2/test.txt，则对象的全路径为src1/src2/test.txt）。该匹配模式仅适用于对象名非“/”结尾的对象。</p> <p>支持指定多个include参数，如--include *.xxx --include *.xxx。</p>
--at	-A	否	<p>指定按照文件最后访问时间作为--timeRange匹配模式，仅上传最后访问时间满足--timeRange时间段的文件。</p> <p>需要与--timeRange配合使用。</p>
--disableDirObject	-D	否	<p>上传文件夹时候，文件夹本身不会作为一个对象上传，在有很多空文件夹场景时候可以避免无用文件夹上传到对象存储桶中；文件夹下有文件时候，文件夹下文件依然会上传并保持原有路径格式，不受影响。</p>
--timeRange	-T	否	<p>上传文件时的时间段匹配模式，仅上传最后修改时间在该时间段内的文件。该匹配模式优先级低于文件名匹配模式：exclude和include，优先执行文件名匹配模式后才会执行该匹配模式。</p> <p>使用“time1-time2”，代表匹配的时间段，其中time1必须小于等于time2，格式为yyyyMMddHHmmss。时间格式支持自动补全，例如yyyyMMdd等价于yyyyMMdd000000，yyyyMM等价于yyyyMM01000000。</p> <p>使用“*-time2”，代表匹配最后修改时间在time2之前的所有文件，使用“time1-*”，代表匹配最后修改时间time1之后的所有文件。</p> <p>该匹配模式表示的时间是UTC时间。</p>

参数	简写	是否必选	说明
--mf	-m	否	设置名称匹配模式（include、exclude）和时间匹配模式（timeRange）对文件夹也生效。
--o	-o	否	生成结果清单文件的文件夹，命令执行完成后，会在该文件夹下生成结果清单文件（可能包含成功结果、失败结果、警告结果三个文件），默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。 结果清单文件命名规则：up_{succeed failed warning}_report_时间_TaskId.txt。单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024，可在配置文件中通过recordMaxLogSize和recordBackups分别配置。 当涉及多文件/文件夹时，如需确认失败任务的具体错误信息，可参考结果清单文件夹下的失败结果清单文件“up_failed_report_时间_TaskId.txt”及日志路径下的 日志文件 。
--e	-P	否	指定终端节点。
--i	-i	否	指定用户的AK。
--k	-k	否	指定用户的SK。
--t	-n	否	指定用户的securitytoken。
--recover	-z	否	待恢复上传任务结果清单文件的任务号。 任务号（TaskId）可在每次上传任务运行完毕后获取，或者通过结果清单文件名查询（文件名除去后缀.txt后的后36位）。 待恢复的上传任务会从结果清单的文件夹中查找，结果清单文件夹的路径参考附加参数--o。

命令示例

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

上传数据前需要使用**switch命令**进入待操作的项目。

说明

- 列举路径中的文件夹对象时，需使用/xxx/格式，如示例中的/src/。
- 列举本地路径中的文件夹对象时，需要使用路径/文件名或路径\文件名格式，如示例中的D:\local。
- 如果路径中带有特殊字符比如（ ）之类的，运行的时候需要将整个路径用""括起来，或者用\将字符转义，例如：health cp "/src2/abs (1) .txt" /src1/ 或者 health cp /src2/abs \\(1)\.txt /src1/ 。

- 将本地D:\local\data\test.txt路径下的txt文件上传至EIHealth平台项目中的src文件夹中。
上传完成后，会在EIHealth平台数据管理页面的src文件中，生成test.txt文件。
`health upload D:\local\data\test.txt /src/`
- 上传文件并重命名为abc.txt。
`health upload D:\local\data\test.txt /src/ --rename abc.txt`
- 递归上传本地D:\local路径下的所有文件和文件夹至项目src文件夹中（不上传local文件夹本身）。
`health upload D:\local /src/ -r -f -l`
- 上传本地D:\local文件夹至项目src文件夹中（上传local文件夹本身）
`health upload D:\local /src/ -r -f`
- 增量上传文件，同步本地D:\local\test.txt文件至项目src文件夹中。
`health upload D:\local\test.txt /src/ --update -r`

5.7 清理本地记录的上传文件

上传对象时，支持断点续传功能，本地因此可能会残留上传异常产生的文件，可以使用该命令对异常文件进行清理。

命令结构

```
health clear [flags]
```

表 5-5 参数说明

参数	简写	是否必选	说明
upload	-u	是	上传文件时，本地生成的断点目录。

命令示例

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

```
health clear -u
# 返回结果如下
delete the upload dir for checkpoint successfully!
```

5.8 下载数据

使用**download**命令将EIHealth平台的数据下载到本地，此命令不支持下载引用项目中的数据。

数据在下载的过程中，受网络影响可能出现损坏，下载命令默认会在下载完成后，验证项目中数据的MD5值与本地数据的MD5值的一致性，以及验证项目中数据的大小与本地数据大小一致性。

须知

下载数据操作会产生流量费用，计费方式为按需计费，计费详情请参考[OBS数据下载费用](#)。

命令结构

```
health download <srcdir> <destdir> [flags]
```

表 5-6 参数说明

参数	简写	是否必选	描述
srcdir	无	是	源路径。可以为已引用的数据，查看引用数据时，需要使用绝对路径。
destdir	无	是	目的路径。
--rename	-e	否	重命名，下载文件时可选。
--recursive	-r	否	按指定的对象名前缀批量下载，批量下载时必须选。
--force	-f	否	强制操作，不进行询问提示，批量下载时可选。
--flat	-l	否	批量下载时，不包含上一级父对象名前缀，批量下载时可选。
--update	-u	否	增量下载操作，设置该参数后，下载每个文件时会比对本地的文件，仅在以下情况时下载数据： <ul style="list-style-type: none"> 文件不存在。 下载文件大小与本地文件大小不一致。 文件的最后修改时间不一致。
--parallel	-p	否	每个分段复制任务的最大并发数。
--jobs	-j	否	下载文件夹时，批量任务的最大并发数，默认为5。取值范围[1,10]。
--fr	-R	否	下载对象时生成结果清单文件。
--tempFileDir	-D	否	分段下载时保存临时文件的文件夹，默认为配置文件中的defaultTempFileDir。 如果该值为空，且配置文件中的defaultTempFileDir也为空，则分段下载时生成的临时文件会保存在待下载文件的同目录下并以.obs.temp结尾。由于分段下载时生成的临时文件会保存在该目录，请确保运行obsutil的用户对该路径有写权限。该路径的可用空间需要大于待下载对象的大小。
--dryRun	-y	否	测试模式运行，不执行实际的下载操作。

参数	简写	是否必选	描述
--vlength	-v	否	下载完成后，验证本地文件大小是否与桶中对象的大小一致。
--vmd5	-M	否	下载完成后，验证本地文件的MD5值是否与桶中对象的MD5值一致。 桶中对象必须包含元数据x-obs-md5chksum，否则会忽略MD5值校验。
--threshold	-t	否	开启分段下载任务的阈值，单位：字节，默认为配置文件中的defaultBigfileThreshold。 如果待下载的对象小于该阈值，则使用直接下载模式；否则使用分段下载模式。直接下载模式不会产生断点记录文件，不支持断点续传。支持带容量单位配置，例如，配置1MB代表1048576字节。
--versionId	-V	否	下载单个对象时可指定的对象版本号。
--ps	-s	否	每个分段下载任务的段大小，单位：字节，默认为配置文件中的defaultPartSize。 支持带容量单位配置，例如，配置1MB代表1048576字节。支持配置为auto，此时obsutil会根据源对象大小自动设置每个分段任务的段大小。
--cpd	-C	否	生成断点记录文件的文件夹，默认为运行obsutil命令的用户目录的子文件夹obsutil_checkpoint。 每个分段下载任务会产生唯一对应的断点记录文件并保存至该文件夹的down子文件夹下，分段任务执行成功后，对应的断点记录文件会被自动删除；分段任务执行失败或被中断后，下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。

参数	简写	是否必选	描述
--exclude	-x	否	<p>不包含源对象的匹配模式，如：*.txt。</p> <p>支持“*”匹配多个任意字符和“?”匹配单个任意字符，例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</p> <p>您可以使用“*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。如果待下载的对象名匹配该参数，则跳过该对象的复制。</p> <p>建议使用引号传递该匹配模式（macOS/Linux操作系统使用单引号，Windows操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。该匹配模式作用于对象全路径（含从根路径开始的对象前缀和对象名，例如，桶内对象路径为obs://bucket/src1/src2/test.txt，则对象的全路径为src1/src2/test.txt）。该匹配模式仅适用于对象名非“/”结尾的对象。</p> <p>支持指定多个exclude参数，如--exclude *.xxx --exclude *.xxx。</p>
--include	-I	否	<p>包含源对象的匹配模式，如：*.jpg。</p> <p>支持“*”匹配多个任意字符和“?”匹配单个任意字符。</p> <p>您可以使用“*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。首先执行exclude的匹配规则，如果待下载的对象名不匹配exclude，则判断待下载的对象名是否匹配该参数，如果匹配则下载该对象，否则跳过该文件的下载。</p> <p>建议使用引号传递该匹配模式（macOS/Linux操作系统使用单引号，Windows操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。该匹配模式作用于对象全路径（含从根路径开始的对象前缀和对象名，例如，桶内对象路径为obs://bucket/src1/src2/test.txt，则对象的全路径为src1/src2/test.txt）。该匹配模式仅适用于对象名非“/”结尾的对象。</p> <p>支持指定多个include参数，如--include *.xxx --include *.xxx。</p>

参数	简写	是否必选	描述
--timeRange	-T	否	<p>下载对象时的时间段匹配模式，仅下载最后修改时间在该时间段内的对象。该匹配模式优先级低于对象名匹配模式：exclude和include，优先执行对象名匹配模式后才会执行该匹配模式。</p> <p>使用“time1-time2”，代表匹配的时间段，其中time1必须小于等于time2，格式为yyyyMMddHHmss。时间格式支持自动补全，例如yyyyMMdd等价于yyyyMMdd000000，yyyyMM等价于yyyyMM01000000。</p> <p>使用“*-time2”，代表匹配最后修改时间在time2之前的所有文件，使用“time1-*”，代表匹配最后修改时间time1之后的所有文件。</p> <p>该匹配模式表示的时间是UTC时间。该匹配模式仅适用于对象名非“/”结尾的对象。</p>
--mf	-m	否	<p>设置名称匹配模式（include、exclude）和时间匹配模式（timeRange），对对象名以“/”结尾的对象也生效。</p>
--o	-o	否	<p>生成结果清单文件的文件夹，命令执行完成后，会在该文件夹下生成结果清单文件（可能包含成功结果、失败结果和警告结果三个文件），默认为运行obsutil命令的用户目录的子文件夹obsutil_output。</p> <p>结果清单文件命名规则：cp_{succeed failed warning}_report_时间_TaskId.txt。单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024，可在配置文件中通过recordMaxLogSize和recordBackups分别配置。</p> <p>当涉及多文件/文件夹时，如需确认失败任务的具体错误信息，可参考结果清单文件夹下的失败结果清单文件“cp_failed_report_时间_TaskId.txt”及日志路径下的日志文件。</p>
--e	-P	否	指定终端节点。
--i	-i	否	指定用户的AK。
--k	-k	否	指定用户的SK。
--t	-n	否	指定用户的securitytoken。

参数	简写	是否必选	描述
--recover	-z	否	待恢复下载任务结果清单文件的任务号。 任务号 (TaskId) 可在每次下载任务运行完毕后获取, 或者通过结果清单文件名查询 (文件名除去后缀.txt后的后36位)。 待恢复的下载任务会从结果清单的文件夹中查找, 结果清单文件夹的路径参考附加参数--O。

命令示例

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

📖 说明

列举路径中的文件夹对象时, 需使用/xxx/格式, 如示例中的/src/。

如果路径中带有特殊字符比如 () 之类的, 运行的时候需要将整个路径用""括起来, 或者用\将字符转义, 例如: health cp "/src2/abs (1).txt" /src1/ 或者 health cp /src2/abs\(1\).txt /src1/。

- 下载EIHealth平台bucket-test项目中src/abc.txt文件到本地的data路径中。
health download /src/abc.txt D:\local\data
- 下载项目中的abc.txt文件到本地, 重命名为abc1.txt。
health download /src/abc.txt D:\local\data --rename abc1.txt
- 递归下载项目中src文件夹中的所有文件和文件夹 (包含src文件夹本身) 至本地data路径, 且下载过程中不进行询问提示。
health download /src/ D:\local\data -r -f
- 递归下载项目中src文件夹中的所有文件和文件夹 (不包含src文件夹本身) 至本地data路径, 且下载过程中不进行询问提示。
health download /src/ D:\local\data -r -f --flat
- 增量下载文件。同步项目中src文件夹下所有的文件和文件夹至本地data路径中。
health download /src/ D:\local\data --update -r
- 使用引用方式下载。
引用了bucket-test项目, 下载引用项目根目录的test1.txt文件至本地的src1文件夹中, 命令如下:
health download bucket-test:/test1.txt /src1

5.9 删除数据

使用rm命令删除EIHealth平台的文件或目录, 此命令不支持删除引用的数据。

命令结构

```
health rm <destdir> [flags]
```

表 5-7 参数说明

参数	简写	是否必选	说明
destdir	无	是	目的文件或目录。
--force	-f	否	强制操作，不进行询问提示。
--recursive	-r	否	按指定的对象名前缀批量删除，批量删除时必须。
--jobs	-j	否	批量删除对象时，批量任务的最大并发数，默认为5。取值范围[1,10]。
--fr	-R	否	删除单个对象时生成结果清单文件。
--versionId	-V	否	待删除对象的版本号。
--o	-o	否	生成结果清单文件的文件夹，命令执行完成后，会在该文件夹下生成结果清单文件（可能包含成功结果和失败结果两个文件），默认为运行obsutil命令的用户目录的子文件夹obsutil_output。 结果清单文件命名规则：rm_{succeed failed}_report_时间_TaskId.txt。单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024，可在配置文件中通过recordMaxLogSize和recordBackups分别配置。 当涉及多文件/文件夹时，如需确认失败任务的具体错误信息，可参考结果清单文件夹下的失败结果清单文件“rm_failed_report_时间_TaskId.txt”及日志路径下的 日志文件 。
--e	-P	否	指定终端节点。
--i	-i	否	指定用户的AK。
--k	-k	否	指定用户的SK。
--t	-n	否	指定用户的securitytoken。
--v	-v	否	按指定的对象名前缀批量删除多版本对象和多版本删除标记。

命令示例

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

- 删除单个数据
health rm /src2/test.txt
- 删除单个文件夹
health rm /src2 -r -f

5.10 数据导入

使用import命令引用数据到当前所在项目或者导入网上数据。

命令结构

```
health import data <src-dir> <dest-dir> [flags]
```

表 5-8 参数说明

参数	简写	是否必选	说明
src-dir	无	是	源路径，支持四种格式，分别是医疗项目、外部桶、外部并行文件系统或者网络数据下载链接，对应格式分别为ProjectName:/path、OBS:BucketName:/path、OBSFS:ObsfsName:/path、下载url。 须知 源路径中如果要填写多个路径，路径之间用空格分隔。如果路径中有特殊字符或者空格，用""包裹单个路径；如果路径中有""则使用"包裹，如路径中有"则使用""包裹。
dest-dir	无	否	要导入的本项目路径，导入网上数据时必选，路径样式为：path/；引用数据时目的存放路径为本项目的根目录，故无需指定destdir。
--md5s	-m	否	网络下载链接对应的md5列表，值用；分隔，和网络数据链接一一对应。值用于下载完后对文件进行md5校验。
--project	无	否	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

- 从医疗项目中引用数据
health import data eihealth-project-name:/path1/
返回结果如下
reference data successfully!
- 从外部桶中引用数据
health import data OBS:obs-buckt-name:/path1/
返回结果如下
reference data successfully!
- 从外部并行文件系统中引用数据
health import data OBSFS:obsfs-buckt-name:/path1/
返回结果如下
reference data successfully!

执行上述三个命令成功后，项目根目录层级显示如下：

```

|---lmx-test-project      # 本项目名称
|   |---path1
|   |---path2
|   |---.....
|---eihealth-project-name # 引用的项目名
|   |---path1           # 引用的项目的源路径
|---OBS:obs-bucket-name  # 引用的外部桶名
|   |---path1           # 引用的外部桶的源路径
|---OBSFS:obsfs-bucket-name: # 引用的外部并行系统名称
|   |---path1           # 引用的外部并行系统的源路径
    
```

- 导入网络数据

Windows:

```

health import data https://eihealth-lmx-test-01-aad9f09b.obs.cn-north-4.myhuaweicloud.com/
test1.csv? https://eihealth-lmx-test-02-aad9f09b.obs.cn-north-4.myhuaweicloud.com/test2.csv path/ --
md5=your-md5-value1,your-md5-value2
# 返回结果如下
import data form network successfully! the result import path is xxxx.
    
```

Linux:

```

./health import data https://eihealth-lmx-test-01-aad9f09b.obs.cn-north-4.myhuaweicloud.com/
test1.csv? https://eihealth-lmx-test-02-aad9f09b.obs.cn-north-4.myhuaweicloud.com/test2.csv path/ --
md5=your-md5-value1,your-md5-value2
# 返回结果如下
import data form network successfully! the result import path is xxxx.
    
```

5.11 创建归档

使用**create**命令创建数据归档。

命令结构

```

health create archive <archive-name> [flags]
# 命令中的archive可替换为backup
    
```

表 5-9 参数说明

参数	简写	是否必选	说明
archive-name	无	是	归档名称。
--path	-p	是	归档的数据路径，多个对象用分号(;)分隔，开头不需要加/。例如，"xxx";"yyy"。
--description	-d	否	归档描述。
--delete	-l	否	是否删除已归档数据（默认false）。
--project	无	否	指定项目名。未填写则使用配置文件中的项目名。
--storage-type	-s	否	归档的存储类型，支持STANDARD和COLD。

命令示例

本节以Windows为例介绍eihealth-toolkit的使用过程。

```
health create backup h-test2 --path lmx-test-data/ -s COLD --description test
# 返回结果如下
create archive h-test2 successfully!
```

📖 说明

Linux和macOS使用时, 多个路径使用;进行分隔, 整个路径使用"引起。例如:

```
./health create archive lmx-test-cli --path 'newjob;/output/'
```

5.12 获取归档数据

使用get命令查看归档列表或者下载归档的全部数据清单。

命令结构

```
health get archive <archive-id> [flags]
# 命令中的archive可替换为backup
```

表 5-10 参数说明

参数	简写	是否必选	说明
archive-id	无	否	归档id, 不指定id表示获取归档列表, 指定id表示获取归档详情。
--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], 获取归档列表时生效。
--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], 获取归档列表时生效。
--sort-dir	-s	否	降序或升序(分别对应desc和asc, 默认为desc), 获取归档列表时生效。
--sort-key	-k	否	排序字段, 支持type和end_time, 默认为end_time, 获取归档列表时生效。
--data-list	-d	否	获取指定归档的全部数据清单, 需要与archive-id一起使用才生效。

命令示例

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

- 查看归档列表

```
health get archive -l 2 -o 3
# 返回结果如下
Name      Region      ID                                     Type      Size
Operator  Start Time  End Time                               Archive Days Storage Type
ngs       region01    c51ac7df-56a2-4b1c-b654-71103edfc630  manual    13.17GB
ei_eihealth_02  2022-12-23 18:15:01 2022-12-23 18:16:02 20 COLD
fsfs     region01    fe67eeaf-68e6-4ed5-9245-a48340a9efba  manual    282.60KB
ei_eihealth_02  2022-11-25 16:24:26 2022-11-25 16:24:36 48 STANDARD
```

- 获取指定归档详情

```
health get archive id-123455677
# 返回结果如下
{
  "id": "c51ac7df-56a2-4b1c-b654-71103edfc630",
  "name": "ngs",
  "type": "manual",
  "region": "region01",
  "paths": [
    "NGS-small/"
  ],
  "start_time": "2022-12-23T10:15:01Z",
  "end_time": "2022-12-23T10:16:02Z",
  "archive_days": 20,
  "size": 14144732346,
  "description": "",
  "operator_name": "ei_eihealth_02",
  "storage_type": "COLD"
}
```

- 获取指定归档的全部数据清单

```
health get archive id-123455677 --data-list
# 返回结果如下
asset/demo01/test01.pdb
asset/demo01/test02.pdb
asset/demo01/test03.pdb
asset/demo01/test04.pdb
asset/demo01/test05.pdb
```

5.13 删除归档

使用**delete**命令删除归档数据。

命令结构

```
health delete archive <archive-id> [flags]
# 命令中的archive可替换为backup
```

表 5-11 参数说明

参数	简写	是否必选	说明
archive-id	无	是	归档id。
--project	无	否	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

```
health delete archive id-12345678
# 返回结果如下
delete archive id-12345678 successfully!
```

5.14 恢复归档

使用**restore**命令恢复归档数据。

命令结构

```
health restore archive <archive-id> [flags]
# 命令中的archive可替换为backup
```

表 5-12 参数说明

参数	简写	是否必选	说明
--destdir	-d	是	目标路径。
archive-id	无	是	归档id。

命令示例

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

- 恢复归档数据到根目录

```
health restore archive 123456 -d www-test-dev:/ 或者 health restore archive 123456 -d /
# 返回结果如下
restore archive 123456 successfully!
```
- 恢复归档数据到test-folder

```
health restore archive 123456 -d test-folder/
# 返回结果如下
restore archive 123456 successfully!
```

5.15 获取数据归档列表

使用get命令获取数据归档列表。

命令结构

```
health get archive-config
# 命令中的archive-config可替换为backup-config
```

命令示例

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

```
health get archive-config
# 返回结果如下
Region Id          Current Region
cn-north-7         true
```

5.16 修改归档区域

使用edit命令修改归档区域。

命令结构

```
health edit archive-region <archive-region> 或者 health update archive-region <archive-region>
# 命令中的archive-region可替换为backup-region
```

表 5-13 参数说明

参数	简写	是否必选	说明
archive-region	无	是	归档区域。

命令示例

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

```
health edit archive-region cn-north-7 或者 health update archive-region cn-north-7
# 返回结果如下
update archive region successfully!
```

5.17 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

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

表 5-14 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： <ul style="list-style-type: none"> CLONE_DATA DELETE_DATA IMPORT_DATA IMPORT_NETWORK_DATA SUBSCRIBE_DATA BACKUP_DATA BACKUP_DATA_DELETE RESTORE_DATA DATABASE_IMPORT
--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。

参数	简写	是否必选	说明
--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。
--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。
--log	-g	否	在界面显示数据作业执行日志，需要与data-job-id一起使用。 说明 <ul style="list-style-type: none"> 参数--log和--file不能同时使用。 参数--log的优先级高于--file。如果命令中同时包含参数--log和--file，默认只执行--log。
--file	-f	否	本地文件夹路径及自定义文件名称，下载详细数据作业执行日志到本地，需要与data-job-id一起使用。
--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	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

- 获取作业列表

```
health 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
job-03 xxxxx  DATABASE_IMPORT  RUNNING  LMX-01   2021-02-01 11:11:27  2s
```

- 获取作业详情

```
health 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": "BACKUP_DATA"
}
```

- 获取数据作业执行日志

```
health get data-job 12345678 --log
# 返回结果如下
2021-02-01 11:11:27
Complete import file lmx-test-01:/lmx-database/data1.txt

2021-02-01 11:11:27
Import row 2-4 of file lmx-test-01:/lmx-database/data1.txt
```

- 下载数据作业执行日志

```
health get data-job 12345678 --file d:/log.txt
# 返回结果如下
download the log file of data job to d:/log.txt successfully!
```

5.18 删除数据作业

使用**delete**命令删除数据作业。

命令结构

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

表 5-15 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	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

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

5.19 重试数据作业

使用retry命令重试数据作业。

命令结构

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

表 5-16 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	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

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

5.20 取消数据作业

使用cancel命令取消数据作业。

命令结构

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

表 5-17 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	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

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

5.21 移动对象

使用mv命令移动对象或批量移动对象。

命令结构

```
health mv <object> <object-dest> [flags]
```

表 5-18 参数说明

参数	简写	是否必选	说明
object	无	<ul style="list-style-type: none"> 移动对象时必选 批量移动时可选 	<p>移动对象时的源对象名，或批量移动时源对象名前缀。</p> <p>规则如下：</p> <ul style="list-style-type: none"> 移动对象时，该值不能为空；如果object-dest为空，则将源对象移动到目标桶的根目录；如果object-dest以“/”结尾，则目标对象名为object-dest+源对象名；否则，目标对象名为object-dest。 批量移动时，如果该值为空，则移动源桶中的所有对象；否则，移动源桶中以该值为对象名前缀的所有对象。目标对象名的确认规则为： <ul style="list-style-type: none"> 如果object-dest以“/”结尾，目标对象名为object-dest+源对象名。 如果object-dest不以“/”结尾，目标对象名为object-dest+源对象名。 <p>说明</p> <ul style="list-style-type: none"> 当移动对象是文件夹时，如果object以“/”结尾，object-dest没有以“/”结尾，会将object中的所有文件移动至object-dest中，文件夹名称与object-dest一致；如果object没有以“/”结尾，object-dest结尾无论加“/”或不加“/”，都会将object移动到object-dest目录下。 可参考命令结构了解该参数的使用方式。
object-dest	无	是	目标路径。
--fr	-R	否	移动对象时生成结果清单文件。移动对象时该参数可选。
--flat	-l	否	移动时，不包含上一级父对象名前缀。批量移动时该参数可选。
--dryRun	-y	否	测试模式运行，不执行实际的移动操作。

参数	简写	是否必选	说明
--update	-u	否	增量移动操作，设置该参数后，移动每个源对象时会对比目标桶中对应路径的目标对象，仅在目标对象不存在，或者目标对象大小与源对象大小不一致，或者目标对象的最后修改时间早于源对象的最后修改时间时进行移动。 如果目标对象与源对象大小以及修改时间都一致，此时会直接删除源对象，而不进行移动操作。
--parallel	-p	否	每个分段移动任务的最大并发数，默认为配置文件中的defaultParallels。
--threshold	-t	否	开启分段移动任务的阈值，单位：字节，默认为配置文件中的defaultBigfileThreshold。 如果待移动的源对象小于该阈值，则使用直接移动模式；否则使用分段移动模式。直接移动模式不会产生断点记录文件，不支持断点续传。支持带容量单位配置，例如，配置1MB代表1048576字节。
--versionId	-V	否	移动单个对象时可指定的源对象版本号。 POSIX桶无version概念，故POSIX桶不支持versionId选项。 移动对象时该参数可选。
--acl	-a	否	移动对象时可指定的目标对象的预定义访问策略。支持的值： <ul style="list-style-type: none"> private：私有读写 public-read：公共读 public-read-write：公共读写 bucket-owner-full-control：桶拥有者完全控制
--meta	-E	否	移动对象时可指定的目标对象的自定义元数据。格式为： key1:value1#key2:value2#key3:value3。 以上值表示移动对象完成后，桶内目标对象包含三组自定义元数据，分别为： key1:value1、key2:value2、key3:value3。
--ps	-s	否	每个分段移动任务的段大小，单位：字节，取值范围是100KB~5GB，默认为配置文件中的defaultPartSize。 支持带容量单位配置，例如，配置1MB代表1048576字节。支持配置为auto，此时obsutil会根据源对象大小自动设置每个分段任务的段大小。

参数	简写	是否必选	说明
--cpd	-C	否	<p>生成断点记录文件的文件夹，默认为运行obsutil命令的用户目录的子文件夹obsutil_checkpoint。</p> <p>每个分段移动任务会产生唯一对应的断点记录文件并保存至该文件夹的copy子文件夹下，分段任务执行成功后，对应的断点记录文件会被自动删除；分段任务执行失败或被中断后，下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。</p>
--recursive	-r	否	按指定源桶的对象名前缀批量移动对象。批量移动时该参数可选。
--force	-f	否	强制操作，不进行询问提示。批量移动时该参数可选。
--jobs	-j	否	<p>批量移动时的最大并发数，默认为配置文件中的defaultJobs。**说明：工具会保证该值至少为1。</p> <p>批量移动时该参数可选。</p>
--exclude	-x	否	<p>不包含源对象的匹配模式，如：*.txt。批量移动时该参数可选。</p> <p>支持“*”匹配多个任意字符和“?”匹配单个任意字符。例如，abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</p> <p>您可以使用“*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。如果待移动的对象名匹配该参数，则跳过该对象的移动。</p> <p>建议使用引号传递该匹配模式（macOS/Linux操作系统使用单引号，Windows操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。该匹配模式作用于对象全路径（含从根路径开始的对象前缀和对象名，例如，桶内对象路径为obs://bucket/src1/src2/test.txt，则对象的全路径为src1/src2/test.txt）。该匹配模式仅适用于对象名非“/”结尾的对象。</p> <p>支持指定多个exclude参数，如--exclude *.xxx --exclude *.xxx。</p>

参数	简写	是否必选	说明
--include	-l	否	<p>包含源对象的匹配模式，如：*.jpg。批量移动时该参数可选。</p> <p>支持“*”匹配多个任意字符和“?”匹配单个任意字符。</p> <p>您可以使用“*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。首先执行exclude的匹配规则，如果待移动的对象名不匹配exclude，则判断待移动的对象名是否匹配该参数，如果匹配则移动该对象，否则跳过该文件的复制。</p> <p>建议使用引号传递该匹配模式（macOS/Linux操作系统使用单引号，Windows操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。该匹配模式作用于对象全路径（含从根路径开始的对象前缀和对象名，例如，桶内对象路径为obs://bucket/src1/src2/test.txt，则对象的全路径为src1/src2/test.txt）。该匹配模式仅适用于对象名非“/”结尾的对象。</p> <p>支持指定多个include参数，如--include *.xxx --include *.xxx。</p>
--timeRange	-T	否	<p>移动对象时的时间段匹配模式，仅移动最后修改时间在该时间段内的对象。该匹配模式优先级低于对象名匹配模式：exclude和include，优先执行对象名匹配模式后才会执行该匹配模式。</p> <p>使用“time1-time2”，代表匹配的时间段，其中time1必须小于等于time2，格式为yyyyMMddHHmms。时间格式支持自动补全，例如yyyyMMdd等价于yyyyMMdd000000，yyyyMM等价于yyyyMM01000000。</p> <p>使用“*-time2”，代表匹配最后修改时间在time2之前的所有文件。</p> <p>使用“time1-*”，代表匹配最后修改时间time1之后的所有文件。</p> <p>该匹配模式表示的时间是UTC时间。该匹配模式仅适用于对象名非“/”结尾的对象。</p>
--mf	-m	否	<p>设置名称匹配模式（include、exclude）和时间匹配模式（timeRange）对对象名以“/”结尾的对象也生效。</p>

参数	简写	是否必选	说明
--o	-o	否	生成结果清单文件的文件夹，命令执行完成后，会在该文件夹下生成结果清单文件（可能包含成功结果、失败结果和警告结果三个文件），默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。 结果清单文件命名规则：mv_{succeed failed warning}_report_时间_TaskId.txt。 单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024，可在配置文件中通过recordMaxLogSize和recordBackups分别配置。 当涉及多文件/文件夹时，如需确认失败任务的具体错误信息，可参考结果清单文件夹下的失败结果清单文件“mv_failed_report_时间_TaskId.txt”及日志路径下的 日志文件 。
--e	-P	否	指定终端节点。
--i	-i	否	指定用户的AK。
--k	-k	否	指定用户的SK。
--t	-n	否	指定用户的securitytoken。

命令示例

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

移动文件夹

示例1：将tmp文件夹中的所有文件移动至t1文件夹下，命令中tmp文件夹结尾需要加“/”，t1文件夹结尾无需加“/”。

```
health mv /tmp/ /t1 -r -f --dryRun
# 返回结果如下
Start at 2022-07-07 09:21:39.973288 +0000 UTC
Move successfully, obs://eihealth-6d95566b-f0c6-4615-b876-678686ec96e3/tmp/ --> obs://eihealth-6d95566b-f0c6-4615-b876-678686ec96e3/t1/, cost [332], status [200], request id [00000181DB8D6A334012AD144BC14321]
```

示例2：将tmp文件夹移动至t1文件夹下，命令中tmp文件夹结尾无需加“/”，t1文件夹结尾可以加或不加“/”。

```
health mv /tmp /t1 -r -f --dryRun 或者 health mv /tmp /t1/ -r -f --dryRun
# 返回结果如下
Start at 2022-07-07 09:25:32.739281 +0000 UTC
Move successfully, obs://eihealth-6d95566b-f0c6-4615-b876-678686ec96e3/tmp --> obs://eihealth-6d95566b-f0c6-4615-b876-678686ec96e3/t1/tmp, cost [356], status [200], request id [00000181DB8D6A334012AD144AA17E01]
```

5.22 增量同步

使用sync命令让本地源路径下的所有内容和OBS指定目标对象进行数据同步，使两边内容保持一致。

- 增量：依次比较源文件和目标对象，只上传存在变化的源文件。
- 同步：命令执行完成后，保证本地源路径是OBS指定目标桶的子集，即本地源路径下的所有文件均能在OBS指定目标桶中找到对应对象。

命令结构

```
health sync <src> <dest> [flags]
```

表 5-19 参数说明

参数	简写	是否必选	说明
src	无	是	被同步的目录或文件。
destdir	无	是	目的目录或者文件。
--type	-H	是	增量同步类型：copy/download/upload。
--crr	-c	否	复制时使用客户端跨区域复制模式，从通过数据流的方式从源桶直接复制数据到目标桶，且两个桶可以是任意两个OBS服务的桶。 若设置了该参数，必须确保更新了配置文件中客户端跨区域复制的相关配置信息，具体可参考 更新配置文件 。复制时源桶对应的配置信息为配置文件中的：akCrr/skCrr/tokenCrr/endpointCrr，目标桶对应的配置信息为配置文件中的：ak/sk/token/endpoint。 开启该模式后，会占用上传和下载的双向带宽。
--fr	-R	否	同步上传文件时生成结果清单文件。同步上传文件时该参数可选。
--arcDir	-g	否	同步上传文件成功后的归档路径，上传成功后的文件会移动到该本地路径下。
--dryRun	-y	否	测试模式运行，不执行实际的上传操作。
--link	-k	否	上传软链接文件/文件夹指向的真实路径。 如果未指定该参数，而待上传的文件是一个软链接且该软链接指向的目标文件不存在，则Windows操作系统会产生“Thesystemcannotfindthefilespecified”的上传失败异常，而macOS/Linux操作系统会产生“Nosuchfileordirectory”的上传失败异常。 文件夹软链接不能形成环，否则上传会以panic的形式退出；如果不希望产生panic，请在配置文件中配置panicForSymbolicLinkCircle为false。
--vlength	-v	否	同步上传完成后，验证上传到桶中对象的大小是否与本地文件大小一致。

参数	简写	是否必选	说明
--vmd5	-M	否	同步上传完成后，验证上传到桶中对象的MD5值是否与本地文件的MD5值一致。 如果待上传的本地文件较大，使用该参数将会因为计算MD5而导致整体性能下降。MD5值校验通过后，会将该值设置为对象元数据x-obs-md5chksum，用于下载或复制时校验MD5。
--parallel	-p	否	每个分段上传任务的最大并发数，默认为配置文件中的defaultParallels。
--threshold	-t	否	开启分段上传任务的阈值，单位：字节，默认为配置文件中的defaultBigfileThreshold。 如果待上传的文件小于该阈值，则使用直接上传模式；否则使用分段上传模式。直接上传模式不会产生断点记录文件，不支持断点续传。支持带容量单位配置，例如，配置1MB代表1048576字节。
--acl	-a	否	同时上传文件时可指定的预定义访问策略。支持的值： <ul style="list-style-type: none"> • private：私有读写 • public-read：公共读 • public-read-write：公共读写 • bucket-owner-full-control：桶拥有者完全控制
--meta	-E	否	上传文件时可指定的自定义元数据。格式为：key1:value1#key2:value2#key3:value3。 以上值表示上传文件完成后，桶内对象包含三组自定义元数据，分别为：key1:value1、key2:value2、key3:value3。
--tempFileDir	-z	否	同步下载时保存临时文件的文件夹，默认为配置文件中的defaultTempFileDir。
--ps	s-	否	每个分段上传任务的段大小，单位：字节，取值范围是100KB~5GB，默认为配置文件中的defaultPartSize。 支持带容量单位配置，例如，配置1MB代表1048576字节。支持配置为auto，此时obsutil会根据源文件大小自动设置每个分段任务的段大小。

参数	简写	是否必选	说明
--cpd	-C	否	<p>生成断点记录文件的文件夹，默认为运行 obsutil 命令的用户目录的子文件夹 obsutil_checkpoint。</p> <p>每个分段上传任务会产生唯一对应的断点记录文件并保存至该文件夹的 upload 子文件夹下，分段任务执行成功后，对应的断点记录文件会被自动删除；分段任务执行失败或被中断后，下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。</p>
--jobs	-j	否	<p>同步上传文件夹时批量任务的最大并发数，默认为配置文件中的 defaultJobs。说明：工具会保证该值至少为 1。</p> <p>同步文件夹时该参数可选。</p>
--exclude	-x	否	<p>不包含文件的匹配模式，如：*.txt。</p> <p>支持“*”匹配多个任意字符和“?”匹配单个任意字符，例如 abc*.txt 代表匹配以 abc 开头以 .txt 结尾的任意文件。</p> <p>您可以使用“*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。如果待上传的文件名匹配该参数，则跳过该文件的上传。</p> <p>建议使用引号传递该匹配模式（macOS/Linux 操作系统使用单引号，Windows 操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。该匹配模式作用于文件全路径（含文件名和文件目录）。该匹配模式仅对文件夹中的文件生效。</p> <p>支持指定多个 exclude 参数，如 --exclude *.xxx --exclude *.xxx。</p> <p>同步文件夹时该参数可选。</p>

参数	简写	是否必选	说明
--include	-I	否	<p>包含文件的匹配模式，如：*.jpg。</p> <p>支持“*”匹配多个任意字符和“?”匹配单个任意字符。</p> <p>您可以使用“*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。首先执行exclude的匹配规则，如果待上传的文件名不匹配exclude，则判断待上传的文件名是否匹配该参数，如果匹配则上传该文件，否则跳过该文件的上传。</p> <p>建议使用引号传递该匹配模式（macOS/Linux操作系统使用单引号，Windows操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。该匹配模式作用于文件全路径（含文件名和文件目录）。该匹配模式仅对文件夹中的文件生效。</p> <p>支持指定多个include参数，如--include *.xxx --include *.xxx。</p> <p>同步文件夹时该参数可选。</p>
--at	-A	否	<p>同步上传文件时指定按照文件最后访问时间作为timeRange匹配模式，仅上传最后访问时间满足timeRange时间段的文件。</p> <p>需配合--timeRange参数使用。</p> <p>同步上传文件夹时该参数可选。</p>
--disableDirObject	-D	否	<p>同步上传文件夹时，文件夹本身不会作为一个对象上传，在有很多空文件夹场景时候可以避免无用文件夹上传到对象存储桶中；文件夹下有文件时候，文件夹下文件依然会上传并保持原有路径格式，不受影响。</p> <p>同步上传文件夹时该参数可选。</p>

参数	简写	是否必选	说明
--timeRange	-T	否	<p>同步上传文件时的时间段匹配模式，仅上传最后修改时间在该时间段内的文件。该匹配模式优先级低于文件名匹配模式：exclude和include，优先执行文件名匹配模式后才会执行该匹配模式。该匹配模式表示的时间是UTC时间。</p> <p>使用“time1-time2”，代表匹配的时间段，其中time1必须小于等于time2，格式为yyyyMMddHHmmss。时间格式支持自动补全，例如yyyyMMdd等价于yyyyMMdd000000，yyyyMM等价于yyyyMM01000000。</p> <p>使用“*-time2”，代表匹配最后修改时间在time2之前的所有文件。</p> <p>使用“time1-*”，代表匹配最后修改时间time1之后的所有文件。</p> <p>同步上传文件夹时该参数可选。</p>
--mf	-m	否	<p>设置名称匹配模式（include、exclude）和时间匹配模式（timeRange）对文件夹也生效。</p>
--o	-o	否	<p>生成结果清单文件的文件夹，命令执行完成后，会在该文件夹下生成结果清单文件（可能包含成功结果、失败结果和警告结果三个文件），默认为运行obsutil命令的用户目录的子文件夹.obsutil_output</p> <p>结果清单文件命名规则：sync_{succeed failed warning}_report_时间_TaskId.txt。单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024，可在配置文件中通过recordMaxLogSize和recordBackups分别配置。当涉及多文件/文件夹时，如需确认失败任务的具体错误信息，可参考结果清单文件夹下的失败结果清单文件“sync_failed_report_时间_TaskId.txt”及日志路径下的日志文件。</p>
--e	-P	否	指定终端节点。
--i	-i	否	指定用户的AK。
--k	-k	否	指定用户的SK。
--t	-n	否	指定用户的securitytoken。

命令示例

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

- 增量同步上传

```
health sync ./bb2/ /tmp/bb3/ --type=upload --fr -j
# 返回结果如下
Start at 2022-04-11 12:04:45.0966584 +0000 UTC
Parallel: 5      Jobs: 2
Threshold: 50.00MB      PartSize: auto
VerifyLength: false      VerifyMd5: false
CheckpointDir: C:\Users\xxx\.obsutil_checkpoint
OutputDir: C:\Users\xxx\.obsutil_output

[-----] 100.00% tps:0.00 ?/s 1/1 0B/0B 435ms
Succeed count is: 1      Failed count is: 0      Skip count is: 1
Succeed bytes is: 0B
Metrics [max cost:n/a, min cost:n/a, average cost:0.00 ms, average tps:0.00, transfered size:0B]

Task id is: 1b3cdbdc-972f-40cd-9ef0-f233991207c3
```

5.23 删除分段上传任务

使用**abort**命令通过分段上传任务的ID，删除指定桶中的分段上传任务；或按指定的对象名前缀批量删除分段上传任务。

命令结构

```
health abort <destdir> [flags]
```

表 5-20 参数说明

参数	简写	是否必选	说明
destdir	无	是	待删除分段上传任务的对象名，或批量删除分段上传任务的对象名前缀。
--u	-u	否	待删除分段上传任务的ID。可参考 列举分段上传任务 获取。 删除单个分段上传任务时该参数必选。
--fr	-R	否	删除单个分段上传任务时生成结果清单文件。 删除单个分段上传任务时该参数可选。
--force	-f	否	强制操作，不进行询问提示。
--recursive	-r	否	按指定的对象名前缀批量删除分段上传任务。 批量删除分段上传任务时该参数必选。
--jobs	-j	否	批量删除分段上传任务的最大并发数，默认为配置文件中的defaultJobs。 批量删除分段上传任务时该参数可选。

参数	简写	是否必选	说明
--o	-o	否	生成结果清单文件的文件夹，命令执行完成后，会在该文件夹下生成结果清单文件（可能包含成功结果和失败结果两个文件），默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。 结果清单文件命名规则：abort_{succeed failed}_report_时间_TaskId.txt。单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024，可在配置文件中通过recordMaxLogSize和recordBackups分别配置。
--e	-P	否	指定终端节点。
--i	-i	否	指定用户的AK。
--k	-k	否	指定用户的SK。
--t	-n	否	指定用户的securitytoken。

命令示例

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

- 删除某个文件的分段任务

```
health abort /tmp/a -r -f
# 返回结果如下
Start at 2022-04-11 11:36:44.7982446 +0000 UTC

[ ] ?% tps:0.00 0/0 236ms
Warn: No task to run
```

5.24 修改权限

修改目录、文件权限。

命令结构

```
health chattri [flags]
```

表 5-21 参数说明

参数	简写	是否必选	说明
--path	-p	附加参数，可选。	修改权限的目录或文件绝对路径。
--delete-policy	-d	附加参数，可选。	设置目录或文件是否允许删除。取值包括：allow、deny。

命令示例

设置路径不可删除。

```
chattri -p /path/1/ -d deny
chattri /path/1/ deny is successful
```

5.25 查看对象属性

查看对象属性。

命令格式

```
health stat <obj> [flags]
```

表 5-22 参数说明

参数	简写	是否必选	说明
obj	不涉及	是	查看的文件、文件夹对象。 注意 如果为引用数据，必须要使用绝对路径（文件夹最后要加/）。未引用的数据不能查看。
--bf	-b	附加参数，可选	设置是否以人方便阅读方式输出结果。取值包括：human-readable、raw。
--e	-P	附加参数，可选	指定终端节点。
--i	-i	附加参数，可选	指定用户的AK。
--k	-k	附加参数，可选	指定用户的SK。
--t	-n	附加参数，可选	指定用户的securitytoken。

命令示例

```
health stat huo:/huo/ --bf human-readable
Start at 2022-09-01 06:53:01.0466497 +0000 UTC
```

```
Key:
  obs://xxx/huo/
LastModified:
  2022-08-30T11:13:48Z
Size:
  0B
StorageClass:
  standard
ETag:
  6c2d85292e26b513c2670998ecf8427e
ContentType:
  application/x-directory
Type:
```



```
folder  
Metadata:  
mtime=1661858028  
mode=16832
```

6 数据库管理命令

6.1 创建数据库模板

使用**create**命令创建数据库模板。

命令结构

```
health create database template <template-name> [flags] 或者 health create db template <template-name> [flags]
```

表 6-1 参数说明

参数	简写	是否必选	说明
template-name	无	否	模板名称，带此参数时会覆盖yaml文件里面的模板名称。
--description	-d	否	描述，带此参数时会覆盖yaml文件里面的模板描述。
--yaml	-y	是	本地模板yaml文件。
--project	无	否	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

```
health create db template --yaml /root/template/demo.yaml  
# 返回结果如下  
create template succussfully! template id is xxx.
```

6.2 获取数据库模板

使用**get**命令获取数据库模板列表、详情或者示例yaml文件。

命令结构

```
health get database template <template-id> [flags] 或者 health get db template <template-id> [flags]
```

表 6-2 参数说明

参数	简写	是否必选	说明
template-id	无	否	template id, 不填写时表示获取template列表, 填写时表示获取template详情。
--sample	-s	否	显示示例yaml文件内容, 以yaml格式打印到控制台, 需要直接获取示例文件可使用Linux输出流重定向。
--project	无	否	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

- 获取模板列表

```
health get db template
# 返回结果如下
Template ID Source Project Creator Created
shennongProject xxx drug-screening EIHealth 2021-02-01 11:11:27
ligandAnnotation xxx drug-screening EIHealth 2021-02-01 11:11:27
```

- 获取模板详情

```
health get db template xxxx
# 返回结果如下
{
  "id": "xxxx",
  "name": "ligandAnnotation",
  "description": "commom template for ligand annotation",
  "source_project_name": "drug-screening-demo-01",
  "source_project_id": "xxxx",
  "source_template_id": "",
  "creator": "EIHealth",
  "columns": [
    {
      "name": "DrugBankID",
      "type": "String",
      "description": "The ID of the drug in DrugBank",
      "nullable": false,
      "primary": true,
      "searchable": true,
      "unique": true,
      "tips": ""
    },
    {
      "name": "GenericName",
      "type": "String",
      "description": "The generic name of the drug",
      "nullable": false,
      "primary": false,
      "searchable": true,
      "unique": false,
      "tips": ""
    }
  ],
  {

```

```
"name": "GroupOfDrug",
"type": "String",
"description": "The group of the drug",
"nullable": false,
"primary": false,
"searchable": false,
"unique": false,
"tips": ""
},
{
  "name": "Formula",
  "type": "String",
  "description": "The chemical formula of the drug",
  "nullable": false,
  "primary": false,
  "searchable": false,
  "unique": false,
  "tips": ""
},
{
  "name": "SMILES",
  "type": "String",
  "description": "The smiles of the drug",
  "nullable": false,
  "primary": false,
  "searchable": false,
  "unique": false,
  "tips": ""
},
{
  "name": "Target",
  "type": "String",
  "description": "The target of the drug",
  "nullable": false,
  "primary": false,
  "searchable": false,
  "unique": false,
  "tips": ""
},
{
  "name": "Description",
  "type": "String",
  "description": "The description of the drug",
  "nullable": false,
  "primary": false,
  "searchable": false,
  "unique": false,
  "tips": ""
},
{
  "name": "FractionCsp3",
  "type": "Double",
  "description": "The ratio of sp3 hybridized carbons over the total carbon count of the drug",
  "nullable": false,
  "primary": false,
  "searchable": false,
  "unique": false,
  "tips": ""
},
{
  "name": "MW",
  "type": "Double",
  "description": "The MW of the drug",
  "nullable": false,
  "primary": false,
  "searchable": false,
  "unique": false,
  "tips": ""
},
}
```

```
{
  "name": "XLOGP3",
  "type": "Double",
  "description": "The XLOGP3 of the drug",
  "nullable": false,
  "primary": false,
  "searchable": false,
  "unique": false,
  "tips": ""
},
{
  "name": "RotatableBondNum",
  "type": "Double",
  "description": "The number of rotatable bonds for flexibility",
  "nullable": false,
  "primary": false,
  "searchable": false,
  "unique": false,
  "tips": ""
},
{
  "name": "TPSA",
  "type": "Double",
  "description": "TPSA for polarity",
  "nullable": false,
  "primary": false,
  "searchable": false,
  "unique": false,
  "tips": ""
}
],
"create_time": "2021-06-01T12:49:15Z",
"primary_key": "DrugBankID",
"is_prefab": true
}
```

- 获取模板示例文件

```
health get db template --sample
```

返回结果如下

database:

name: demo # 模板名称, 必填, 长度范围[3,32], 只能包含字母、数字、中划线、下划线, 必须以字母开头

description: This is demo template # 模板描述, 非必填, 最大长度允许1024

columns: # 数据库列定义列表, 必填, 最多不能超过100个

- name: id # 列名, 必填, 长度[1,55], 必须以字母开头, 只能包含字母、数字、下划线

description: identifier # 列描述, 非必填, 最大长度1024

type: Long # 列类型, 必填, 支持Long, String, Double三种类型

nullable: false # 列是否允许为空, 必填

unique: false # 列是否增加唯一约束, 必填

primary: true # 列是否作为主键, 必填, 若有多个列设为true, 则多个列组成复合主键

searchable: true # 列是否可查询, 必填

tips: uuid # 列的查询提示, 非必填, 若该列为可查询列, 可提供查询提示

- name: user_name

description: user name

type: String

nullable: false

unique: false

primary: false

searchable: true

tips: string

- name: score

description: score

type: Double

nullable: false

unique: false

primary: false

searchable: true

tips: string

- 将示例文件存储到当前所在路径的app.yaml文件中
health get db template --sample > app.yaml

6.3 删除数据库模板

使用delete命令删除数据库模板。

命令结构

```
health delete database template <template-id> [flags] 或者 health delete db template <template-id> [flags]
```

表 6-3 参数说明

参数	简写	是否必选	说明
template-id	无	是	模板id。
--project	无	否	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

```
health delete db template 123345
# 返回结果如下
delete template successfully!
```

6.4 导入数据库模板

使用import命令导入别的项目的数据库模板到当前项目，暂只支持导入单个模板。

命令结构

```
health import database template <template-id> [flags] 或者 health import db template <template-id> [flags]
```

表 6-4 参数说明

参数	简写	是否必选	说明
template-id	无	是	源模板id。
--project	-p	是	源项目名称。
--rename	-r	否	重命名导入模板名称，不填时与源模板名称保持一致。
--current-project	无	否	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

```
health import db template 1223454 --project lmx-test-01 --rename demo
# 返回结果如下
import database template finished! result is:
[
  {
    "source_project_id": "xxxx",
    "source_template_id": "xxxx",
    "destination_template_id": "",
    "destination_template_name": "lmx-cli-template",
    "failed_reason": "",
    "status": "success"
  }
]
```

6.5 创建数据库实例

使用create命令创建数据库实例。

命令结构

```
health create database instance <instance-name> [flags] 或者 health create db instance <instance-name> [flags]
```

表 6-5 参数说明

参数	简写	是否必选	说明
instance-name	无	是	数据库实例名称。
--description	-d	否	描述。
--template-id	-t	是	模板id。
--skip-lines	-s	否	跳过的header行数。不指定参数值时默认为0。
--delimiter	-e	否	常用的三种分隔符为(;\t)，也可以根据导入数据的文件格式设定其他分隔符。
--files	-f	否	生成实例数据的文件列表，多个文件用分号(;)分隔。
--project	无	否	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

```
health create db instance demo --description desc --template-id 123456 --skip-lines 1 --delimiter '\t' --files project-01:/test1.txt;project-01:/test2/txt
```

```
# 返回结果如下
create database instance successfully! instance id is xxx.
```

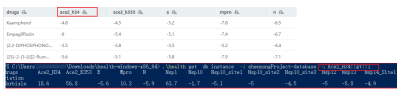
6.6 获取数据库实例

使用get命令获取数据库实例列表、详情或者查询数据库数据。

命令结构

```
health get database instance <instance-id> [flags] 或者 health get db instance <instance-id> [flags]
```

表 6-6 参数说明

参数	简写	是否必选	说明
instance-id	无	否	数据库实例 id，不填写时表示获取数据库实例列表，填写时表示获取数据库实例详情。
--data	-d	否	是否查询数据库数据，需要和database-id参数一起使用。
--limit	-l	否	限制量，默认为10，取值范围[0,1000]，只有在--data参数使用时才生效。
--offset	-o	否	偏移量，默认为0，取值范围[0,100000000]，只有在--data参数使用时才生效。
--query	-q	否	<p>查询条件，格式为{column}::{operator}::{number}。::分割一个表达式中的列名、操作符和值。不同类型支持的操作符如下： String支持like（模糊搜索）、equal（精确搜索）、notlike（不包含）、notequal（不等于），Double支持gt（大于）、lt（小于），Long支持gte（大于等于）、lte（小于等于）、gt（大于）、lt（小于）。只有在--data参数使用时才生效。</p> <p>例如查询下图数据库中第2列，数值大于1数据。 -q Ace2_H34::gt::1</p> 
--sort-key	-k	否	排序字段，只有在--data参数使用时才生效。
--sort-dir	-i	否	排序方向，升序或降序，即ASC和DESC，默认DESC，只有在--data参数使用时才生效。
--columns	-c	否	当某个数据库实例列数非常多时，查看数据时会由于机器显示界面限制导致换行难以查看，添加此参数可以只显示想要查看数据的列名，多个列名用分号隔开(;)，只有在--data参数使用时才生效。

参数	简写	是否必选	说明
--project	无	否	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

- 获取数据库实例列表

```
health get db instance
# 返回结果如下
Database ID Template Source Project Creator Number of Records Created Modified
demo-01 xxx template1 project-01 EIHealth 1234 xxx xxx
demo-02 xxx template1 project-02 EIHealth 347 xxx xxx
```

- 获取模板详情

```
health get db instance 12345
# 返回结果如下
{
  "id": "xxxx",
  "name": "job-1622803262953",
  "description": "database instance for drug screening, do not delete",
  "template": {
    "id": "xxxx",
    "name": "job-1622803262953-d69d2162",
    "description": "template for drug screening, study name is study-rbqr5mvgrqm086kz, job name is job-1622803262953",
    "source_project_name": "drug-screening-demo-01",
    "source_project_id": "xxxx",
    "source_template_id": "",
    "creator": "ei_eihealth_02",
    "columns": [
      {
        "name": "drugs",
        "type": "String",
        "description": "The name of the drug",
        "nullable": false,
        "primary": true,
        "searchable": true,
        "unique": true,
        "tips": ""
      },
      {
        "name": "Nsp10_6zct_pure",
        "type": "Double",
        "description": "The binding energy between the ligand and receptor",
        "nullable": true,
        "primary": false,
        "searchable": false,
        "unique": false,
        "tips": ""
      },
      {
        "name": "NSP13_pure",
        "type": "Double",
        "description": "The binding energy between the ligand and receptor",
        "nullable": true,
        "primary": false,
        "searchable": false,
        "unique": false,
        "tips": ""
      }
    ]
  }
}
```

```

    "name": "Average",
    "type": "Double",
    "description": "The average of the binding energy",
    "nullable": false,
    "primary": false,
    "searchable": false,
    "unique": false,
    "tips": ""
  },
  {
    "name": "Standard_Deviation",
    "type": "Double",
    "description": "The standard deviation of the binding energy",
    "nullable": true,
    "primary": false,
    "searchable": false,
    "unique": false,
    "tips": ""
  }
],
"create_time": "2021-06-04T10:41:32Z",
"primary_key": "drugs",
"is_prefab": false
},
"creator": "ei_eihealth_02",
"create_time": "2021-06-04T10:43:25Z",
"update_time": "2021-06-04T10:43:26Z",
"data_count": 2,
"source_project_name": "drug-screening-demo-01",
"source_project_id": "xxxx",
"source_id": "",
"is_prefab": false
}

```

- 查询数据

```

health get db instance 12345 --data --limit 2
# 返回结果如下
drugs      nsp3  mpro  average  standard_deviation  _row_num
Pyridoxal-phosphate -5.1 -4.7 -4.9  0.2      8496
Tetrahydrofolic-acid -7.8 -8.3 -8.05 0.25     8495

```

- 按列查询数据

```

health get db instance 12345 --data --limit 2 --columns drugs;nsp3;mpro
# 返回结果如下
drugs      nsp3  mpro  _row_num
Pyridoxal-phosphate -5.1 -4.7 8496
Tetrahydrofolic-acid -7.8 -8.3 8495

```

6.7 删除数据库实例

使用delete命令删除数据库实例。

命令结构

```
health delete database instance <instance-id> [falgs] 或者 health delete db instance <instance-id> [falgs]
```

表 6-7 参数说明

参数	简写	是否必选	说明
instance-id	无	是	数据库实例id。
--project	无	否	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

```
health delete database instance 12345678
# 返回结果如下
delete database successfully!
```

6.8 引用数据库或者导入数据到指定数据库

使用import命令引用数据库实例到当前所在项目或者导入数据到指定数据库。

命令结构

```
health import database instance <instance-id> [flags] 或者 health import db instance <instance-id> [flags]
```

表 6-8 参数说明

参数	简写	是否必选	说明
instance-id	无	是	数据库实例id，其中引用数据库实例时可以引用多个数据库实例，用分号(;)分隔；导入数据到数据库时不支持多个实例id。
--project	-p	否	源项目名称，使用此参数时代表引用数据库实例。
--skip-lines	-s	否	跳过的header行数，导入数据到数据库时此参数必填。
--delimiter	-e	否	分隔符(;\t)，导入数据到数据库时此参数必填。
--files	-f	否	导入文件列表，多个文件用分号(;)分隔，导入数据到数据库时此参数必填。
--current-project	无	否	指定项目名。未填写则使用配置文件中的项目名。

命令示例

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

- 引用单个数据库实例

```
health import db instance 12345 --project demo-project
# 返回结果如下
reference database finished! result is:
[
  {
    "source_project_id": "879d952e-93c6-4b09-.....",
    "source_database_id": "5ea7d345-d1b4-4520-.....",
    "destination_database_id": "a94e72de-1305-4cc5-.....",
    "destination_database_name": "test001",
    "failed_reason": "",
    "status": "success"
  }
]
```

- ```

 }
]

```
- 引用多个数据库实例**

```

health import database instance 12345;123333;56777 --project demo-project
返回结果如下
reference database finished! result is:
[
 {
 "source_project_id": "879d952e-93c6-4b09-ae6-....",
 "source_database_id": "e06f784f-c561-4323-92cd-....",
 "destination_database_id": "0589ab48-d685-483f-92b7-....",
 "destination_database_name": "test003",
 "failed_reason": "",
 "status": "success"
 },
 {
 "source_project_id": "879d952e-93c6-4b09-ae6-c5cf....",
 "source_database_id": "cac4cfb9-0160-4221-b7c4-092f3da8....",
 "destination_database_id": "a1d7b16f-92b7-4344-a74d-e9eb0abe....",
 "destination_database_name": "test002",
 "failed_reason": "",
 "status": "success"
 }
]

```
  - 导入数据到指定数据库**

```

health import database instance 12345 --skip-lines 1 --delimiter \t --files project-01:/test1.txt;project-01:/test2/txt
返回结果如下
import data to database successfully!

```

## 6.9 修改数据库实例

使用 `edit` 命令修改数据库实例。

### 命令结构

```
health edit database instance <instance-id> [flags] 或者 health edit db instance <instance-id> [flags]
```

表 6-9 参数说明

| 参数          | 简写 | 是否必选 | 说明                                                  |
|-------------|----|------|-----------------------------------------------------|
| instance-id | 无  | 是    | 数据库实例id。                                            |
| --update    | -u | 否    | 表示执行更新某行操作。--update、--delete、--insert必须要传其中一个，三选一。  |
| --delete    | -d | 否    | 表示执行删除某行操作。--update、--delete、--insert必须要传其中一个，三选一。  |
| --insert    | -i | 否    | 表示执行插入数据行操作。--update、--delete、--insert必须要传其中一个，三选一。 |
| --row-num   | -n | 否    | 行号。执行更新、删除操作时，必须指定该参数。                              |

| 参数        | 简写 | 是否必选 | 说明                                                                                                                                                                         |
|-----------|----|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| --values  | -v | 否    | 修改列名和对应的值，用反单引号引用列名和值，格式为`column-name1:value1`;`column-name2:value2`，更新或插入多个列值时使用分号。<br><b>说明</b><br>对于极其特殊的值，例如：`qqq\'"22, qqq\'"22, aa`;`aaa:4444`，因为与上述要求的格式冲突，暂时不支持操作。 |
| --project | 无  | 否    | 指定项目名。未填写则使用配置文件中的项目名。                                                                                                                                                     |

## 命令示例

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

- 更新单列**  

```
health edit database instance xxxx --update -v "`USER_NAME:column1`;`volume1:1`" -n 19;
返回结果如下
database xxxx update successfully!
```
- 插入**  

```
health edit db instance 14e4b880-e3bc-42f1-a35e-7c31d561ac57 --insert --values "`USER_NAME:column1`;`volume1:1`"
返回结果如下
database xxxx insert successfully!
```
- 删除**  

```
health edit db instance xxxx --delete --row-num 19
返回结果如下
database xxx delete successfully!
```

# 7 镜像管理命令

## 7.1 标记镜像

使用health 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项目，需先使用切换项目命令进入想要上传镜像的项目。

### 命令结构

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

表 7-1 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

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

- 若当前所在项目为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。

- 若当前不在某个项目内，会提示：  
Error: you're currently not in any project. run the switch project command to enter a project first  
Usage:  
health switch project <project-name> [flags]  
Flags:  
-h, --help help for switch

## 7.2 上传镜像

通过health docker push命令将镜像上传至EIHealth平台项目中。

### 命令结构

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

表 7-2 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    | 指定项目名。未填写则使用配置文件中的项目名。                 |

### 命令示例

- Windows  
health docker push demo-image:v1.0 -d "this is a desc" -t APP
- Linux  
./health 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
  - 待上传项目不存在

 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。

## 7.3 下载镜像

通过health docker pull命令下载项目中的镜像。

### 命令结构

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

表 7-3 参数说明

| 参数           | 简写 | 是否必选 | 说明                               |
|--------------|----|------|----------------------------------|
| project-name | 无  | 否    | 源项目名称，要下载的镜像所在的源项目名称，不填时默认为当前项目。 |
| image-name   | 无  | 是    | 镜像名称。                            |
| tag-name     | 无  | 是    | 镜像tag名称。                         |
| --project    | 无  | 否    | 指定项目名。未填写则使用配置文件中的项目名。           |

### 命令示例

```
health 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
```

## 7.4 查询镜像

使用health docker images命令查询项目中的镜像。

### 命令结构

```
health docker images [flags]
```

表 7-4 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    | 指定项目名。未填写则使用配置文件中的项目名。 |

## 命令示例

- 查询当前所在项目demo-project下面的类型为APP的镜像。

```
health 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
```

- 查询当前项目下面的镜像名为demo-image的所有tag信息。

```
health 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
```

## 7.5 导入镜像

使用import命令从源项目导入镜像到当前项目。

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项目。

## 命令结构

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

表 7-5 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项目名称。       |
| 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    | 指定项目名。未填写则使用配置文件中的项目名。 |

## 命令示例

从源项目lmx-project-02导入镜像到当前项目。

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

## 7.6 更新镜像

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

### 命令结构

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

表 7-6 参数说明

| 参数            | 简写 | 是否必选 | 说明                               |
|---------------|----|------|----------------------------------|
| project-name  | 无  | 否    | 要更新的镜像的源项目名称，不填写时默认为当前所在的项目。     |
| 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    | 指定项目名。未填写则使用配置文件中的项目名。           |

### 命令示例

- Linux  
./health docker update lmx-project-01/demo-image --type APP --description 'this is a demo'
- Windows  
health docker update lmx-project-01/demo-image --type APP --description "this is a demo"
- 不填写项目名时，若当前所在项目是lmx-test-01，则执行如下命令后会更新当前项目中源项目为lmx-test-01的镜像demo-image的类型。  
health docker update demo-image --type APP --chip ARM  
# 执行成功返回结果如下  
update image successfully!

## 7.7 删除镜像标签

使用**rmi**命令删除当前项目中指定镜像标签。

- 对于本项目的私有镜像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。
- 对于其他项目的导入镜像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

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

表 7-7 参数说明

| 参数           | 简写 | 是否必选 | 说明                         |
|--------------|----|------|----------------------------|
| project-name | 无  | 否    | 要删除的镜像的源项目名称，不填时默认为当前所在项目。 |
| 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    | 指定项目名。未填写则使用配置文件中的项目名。     |

## 命令示例

假设当前所在项目为lmx-project-01，命令及返回结果示例：

- 删除其他项目lmx-project-02的导入镜像tag。

```
health 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。

```
health docker rmi [AssetMarket]/demo-image:v1.1
执行成功返回结果如下
untagged:[AssetMarket]/demo-image:v1.1
```
- 删除本项目的私有镜像tag，可不填源项目名。

```
health docker rmi demo-image:v1.1 或 health 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
```

# 8 应用管理命令

## 8.1 应用配置文件说明

EIHealth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EIHealth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EIHealth平台。

### 获取应用模板

使用`health 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EIHealth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。

修改模板时，需填写镜像地址，获取方法请参见[获取镜像地址](#)。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(-)、下划线(_)、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
 - health.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|D310。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:
 - '/output'

```

## 获取镜像地址

对于已上传或平台上已有的镜像, 您可以参考以下步骤拼接镜像地址。

1. 登录医疗智能体平台, 进入镜像所在的项目。
2. 在页面上方单击“镜像”, 进入镜像管理页面。
3. 单击镜像名称, 进入镜像详情页。在该页面查看“源项目”、“镜像名称”、“镜像版本”, 按照以下规则拼接出应用配置模板中的image参数。

```
{project-name}/{image-repo}:{image-version} #源项目/镜像名称:镜像版本, 例如htt_test-01/
qvina_w:1.0.0
```



## 8.2 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

```
health create app [flags]
```

表 8-1 参数说明

| 参数                         | 简写              | 是否必选 | 说明                                               |
|----------------------------|-----------------|------|--------------------------------------------------|
| <code>--yaml</code>        | <code>-y</code> | 否    | 本地的应用模板路径。获取应用模板方法请参见 <a href="#">应用配置文件说明</a> 。 |
| <code>--description</code> | <code>-d</code> | 否    | 应用的详细描述信息。                                       |
| <code>--project</code>     | 无               | 否    | 指定项目名。未填写则使用配置文件中的项目名。                           |

### 命令示例

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

使用本地已填写好的应用模板app.yaml创建应用。

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

## 8.3 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

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

表 8-2 参数说明

| 参数            | 简写 | 是否必选 | 说明                                                                                                                                                                                                                                                |
|---------------|----|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID            | 无  | 是    | 可选： <ul style="list-style-type: none"> <li>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<a href="#">创建应用命令示例</a>。</li> <li>应用的名称、版本、所在项目名称 ( app-name:version:srcproject )，srcproject为源项目名称，可选。不指定srcproject时，默认为当前项目。例如 GATK:4.0:projectname。</li> </ul> |
| --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 <a href="#">创建FastQC应用样例</a> 。                                                                                                                                                                                                     |
| --image       | -i | 否    | 镜像地址。获取方法请参见 <a href="#">获取镜像地址</a> 。                                                                                                                                                                                                             |
| --labels      | -l | 否    | 标签，多个标签使用;符号分隔开。                                                                                                                                                                                                                                  |
| --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63字符，必须以health.开头。                                                                                                                                                                                      |
| --project     | 无  | 否    | 指定项目名。未填写则使用配置文件中的项目名。                                                                                                                                                                                                                            |

### 📖 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eihealth-toolkit的使用过程，Linux和macOS环境使用方法基本相同，可参考。

- 通过指定“app-id”修改应用。  

```
health 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。  

```
health 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!
```

## 8.4 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

```
health get app ID [flags]
```

表 8-3 参数说明

| 参数             | 简写 | 是否必选 | 说明                                                                                                                                                                                                   |
|----------------|----|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID             | 无  | 否    | <ul style="list-style-type: none"> <li>不选此参数时，列出当前所在项目的所有应用信息。</li> <li>指定app-id或app-name:version:srcproject，列出具体的应用信息，srcproject为源项目名称，可选。不指定srcproject时，默认为当前项目。例如GATK:4.0:projectname。</li> </ul> |
| --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    | 指定项目名。未填写则使用配置文件中的项目名。                                                                                                                                                                               |
| --label        | -l | 否    | 根据label标签搜索应用。                                                                                                                                                                                       |

### 命令示例

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

- 使用health 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health get app命令查询指定应用信息。  

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



```
health 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_x00356764_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: sdfsf
 values:
 - /sdsds
 license: ""
```

```
health 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，项目内唯一
 # 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版本在一个项目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'
 # 可选 # 'CPU数量'
 cpu: 2C
 # 可选 # '内存大小'
 memory: 4G
 # 可选 # 'GPU类型'
 gpu_type:
 # 可选 # 'GPU数量'
 gpu:
 # 可选 # 输入参数列表
inputs:
 # 参数名称
- name: 'input_name1'
 # 参数类型 # STRING|FILE|DIRECTORY|ENUM
 type: FILE
 # 参数模式, 界面展示, 不做强校验
 pattern: '*.fastq'
 # 是否必填
 required: true
 # 参数描述
 description: ""
 # 枚举类型时, 枚举值列表
 enum:
 - xxx
 # 参数值, 非数组类型仅允许填写单个值
 values:
 - xxx
outputs:
 # 可选 # 输出参数列表, app创建、修改时指定
- name: 'output_name1'
 # FILE和DIRECTORY两种类型, 指定输出的相对路径
 type: FILE
 # 参数模式, 界面展示, 不做强校验
 pattern: '*'
 # 是否必填
 required: true
 # 描述
 description: ""
 values:
 - 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
 #app的许可证
 license: xxx

health 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

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

根据label标签搜索应用
health 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 wwx-test-admin 2024-01-04
09:42:49 2024-01-04 09:42:49

```

## 8.5 删除应用

使用**delete**命令删除指定应用。

### 命令结构

```
health delete app ID [flags]
```

表 8-4 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、所在项目名称 ( app-name:version:srcproject )。srcproject为源项目名称, 可选。不指定srcproject时, 默认为当前项目。 |
| --project | 无  | 否    | 指定项目名。未填写则使用配置文件中的项目名。                                                                                          |

## 命令示例

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

通过指定应用的ID或名称和版本删除应用。

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

## 8.6 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项目。

## 命令结构

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

表 8-5 参数说明

| 参数                      | 简写 | 是否必选 | 说明                        |
|-------------------------|----|------|---------------------------|
| source_proj<br>ect_name | 无  | 是    | 要导入的应用当前所在的源项目名称。         |
| 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    | 指定项目名。未填写则使用配置文件中的项目名。    |

## 命令示例

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

从源项目lmx-project-01导入应用，并重命名应用。

```
health import app ae81debc-c209-11eb-8a0d-fa163e3ddba1:lmx-project-01 --rename lmx-test-import-app
或 health 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"
 }
]
```

# 9 EIHealth 流程管理命令

## 9.1 流程配置文件说明

EIHealth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工作流。

在EIHealth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EIHealth平台。

### 获取流程模板

使用**health 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。

# 详情说明可参考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(-)、下划线(_)、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;
 # 其他场景, app_id为系统分配的唯一标识
```

```

display_name: '步骤1' # 流程的子任务展示名称, 最大长度64
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|D310。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}'

```

## 9.2 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

```
health create workflow [flags]
```

表 9-1 参数说明

| 参数                         | 简写              | 是否必选 | 说明                                               |
|----------------------------|-----------------|------|--------------------------------------------------|
| <code>--yaml</code>        | <code>-y</code> | 否    | 本地的流程模板路径。获取流程模板方法请参见 <a href="#">流程配置文件说明</a> 。 |
| <code>--description</code> | <code>-d</code> | 否    | 流程的详细描述信息。                                       |
| <code>--project</code>     | 无               | 否    | 指定项目名。未填写则使用配置文件中的项目名。                           |

### 命令示例

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

使用本地已填写好的流程模板workflow.yaml创建流程。

```
health create workflow D:\workflow.yaml 或 health create workflow -y D:\workflow.yaml
返回结果如下
create workflow succeed! workflow id : 65b7a404-3f4f-11eb-868a-fa163e3ddba1
```

## 9.3 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

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

表 9-2 参数说明

| 参数            | 简写 | 是否必选 | 说明                                                                                                                                                                                                                                                 |
|---------------|----|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID            | 无  | 是    | 可选： <ul style="list-style-type: none"> <li>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<a href="#">创建流程命令示例</a>。</li> <li>流程的名称、版本、所在项目名称（workflow-name:version:srcproject），srcproject为源项目名称，可选。不指定srcproject时，默认为当前项目。例如ngs:1.0:srcproject。</li> </ul> |
| --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，以当前项目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    | 指定项目名。未填写则使用配置文件中的项目名。                                                                                                                                                                                                                             |

#### 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eihealth-toolkit的使用过程，Linux和macOS环境使用方法基本相同，可参考。

- 通过指定“workflow-id”修改流程。

```
health 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。

```
health 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!
```

## 9.4 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

```
health get workflow ID [flags]
```

表 9-3 参数说明

| 参数             | 简写 | 是否必选 | 说明                                                                                                                                                                                      |
|----------------|----|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID             | 无  | 否    | <ul style="list-style-type: none"> <li>不选此参数时，列出当前所在项目的所有流程信息。</li> <li>指定workflow-id或workflow-name:version:srcproject，列出具体的应用信息，srcproject为源项目名称，可选。不指定srcproject时，默认为当前项目。</li> </ul> |
| --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    | 指定项目名。未填写则使用配置文件中的项目名。                                                                                                                                                                  |
| --label        | -l | 否    | 根据label标签搜索workflow                                                                                                                                                                     |

## 命令示例

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

- 使用health 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 **health get workflow** 命令查询指定的流程信息。

```
health get workflow 550e8400-e29b-41d4-a716-446655440000
health 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:
 - SDFSDF
 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_x00356764_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_x00356764_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

health 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版本在一个项目内保持唯一
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or订阅的资源是需要填写，为空表示本项目
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
health get workflow 3df8c00e-0291-11ed-973f-fa163e507c84 -d ./a
download workflow detail successfully!

根据label标签搜索workflow
health 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 test-project-new-permissions-700wwx-test-
admin 2024-01-04 19:56:36 2024-01-04 19:56:36
4dd32f12-aaf7-11ee-a057-fa163ef319da cli-demo-workflow 2.0.0
labelA,labelB test-project-new-permissions-700wwx-test-
admin 2024-01-04 19:49:23 2024-01-04 19:49:23
```

## 9.5 删除流程

使用`delete`命令删除指定流程。

### 命令结构

```
health delete workflow ID [flags]
```

表 9-4 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、所在项目名称 ( workflow-name:version:srcproject )，srcproject为源项目名称，可选。不指定srcproject时，默认为当前项目。 |
| --project | 无  | 否    | 指定项目名。未填写则使用配置文件中的项目名。                                                                                                  |

### 命令示例

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

通过指定流程的ID或名称和版本删除流程。

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

## 9.6 导入流程

使用`import`命令从源项目导入流程到当前项目。

订阅和已导入的流程不支持再被导入到别的项目。

### 命令结构

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

表 9-5 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项目名称。        |
| 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    | 指定项目名。未填写则使用配置文件中的项目名。   |

## 命令示例

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

从源项目lmx-project导入流程，并重命名流程。

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

# 10 EIHealth 作业管理命令

## 10.1 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EIHealth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EIHealth平台。

### 获取作业模板

使用`health 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EIHealth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}::{源项目名称}; 正常id: 流程id
 tool_type # 作业依赖的组件类型, 仅支持填写workflow
 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}::{源项目名称}; 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
 - health.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|D310'。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'

```

## 10.2 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

```

health create job [flags]
create和submit作用相同
health submit job [flags]

```

表 10-1 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 <a href="#">作业配置文件说明</a> 。                                                                                                           |
| --workflow    | -w | 否    | 基于流程ID创建分析作业, 可使用health get workflow workflow-name:version:srcproject命令查询流程ID, srcproject为源项目名称, 可选。不指定srcproject时, 默认为当前项目。<br>--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    | <p>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。</p> <p>选择时，input和input-file二选一。</p> <p>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。</p> <p>task间使用;;分隔，input的参数路径间使用;;分隔。</p> <p>例如<br/>task1.input1=datapath1;;datapath2;;task1.input2=xxx;xxx;;task2.input1=xxx;;xxx</p>                                                                                                                                                         |
| --input-file | -f | 否    | <p>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。</p> <p>选择时，input和input-file二选一。</p> <p>格式为：{"task0": {"input-1": "datapath"}, "task1": {"inputs": "datapath"}}。</p> <p>同时，输入参数可以设置多个参数值。例如：</p> <pre>--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。 {   "task1":{     "data-01":"projectname:/test.txt",     "data-02":"projectname:/test2.txt;projectname:/test3.txt"   } }</pre> |
| --output_dir | -o | 否    | <p>输出路径（EIHealth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。</p>                                                                                                                                                                                                                                                                                                                                                |
| --timeout    | -t | 否    | <p>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。</p>                                                                                                                                                                                                                                                                                                                                                                                |
| --priority   | -p | 否    | <p>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。</p>                                                                                                                                                                                                                                                                                                                                                                                     |
| --name       | -n | 否    | <p>作业的名称。</p> <p>通过流程ID（workflow id）启动作业时，必选。</p>                                                                                                                                                                                                                                                                                                                                                                                    |

| 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。<br><br>• 获取性能加速资源列表<br><pre>health get resource -t performance # 执行成功返回结果如下 Id      Name      Specifications xxx    sfs-turbo-01  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。<br><br>标签数量取值范围[0,1]。单个标签最大长度为63字符，名称必须以health.开头。                                                                                                                                                                                                         |
| --project      | 无  | 否    | 指定项目名。未填写则使用配置文件中的项目名。                                                                                                                                                                                                                                                                             |
| --io-acc-tasks | -s | 否    | 设定任务的I/O加速类型<br>例如：`taskname1:EVS`;`taskname3:SFS`<br>不写的task默认不带加速类型                                                                                                                                                                                                                              |

## 命令示例

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

### 📖 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。

```

C:\EIHealth>health-windows-v1.0.0-03_64\health.exe create job -w fdf9714-b96e-11ec-8c0a-000000000000 -n ssa
submit job succeed! job id : 453ccc73-b35b-11ec-8c0a-000000000000
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-cl1 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。  

```
health 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。  

```
health 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。  

```
health 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。

```
health 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-fa163e3ddb1
```

- 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。

```
health 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-fa163e3ddb1
```

## 10.3 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

```
health get job ID [flags]
```

表 10-2 参数说明

| 参数                 | 简写 | 是否必选 | 说明                                                                                                                                                                                                                                                    |
|--------------------|----|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID                 | 无  | 否    | <ul style="list-style-type: none"> <li>● 不选此参数时，列出当前所在项目的所有作业信息。</li> <li>● 指定job-id时，列出具体的作业信息。可以同时指定多个job-id。 <ul style="list-style-type: none"> <li>- 不带--detail参数，以yaml格式展示作业基本信息。</li> <li>- 带--detail参数，以json格式展示作业基本信息。</li> </ul> </li> </ul> |
| --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eihealth-toolkit的使用过程，Linux和macOS环境使用方法基本相同，可参考。

- 使用**health 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。

```
health 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。

```
health get job f17a3542-3f7c-11eb-868a-fa163e3ddb1 --detail
{
```

```

"jobs": [{
 "id": "2",
 "name": "zx-1030-mkdir",
 "description": "测试文件创建",
 "priority": 0,
 "timeout": 1440,
 "output_dir": "",
 "status": "SUCCEDED",
 "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_x00356764_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_x00356764_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。

health get job #不带任何参数默认获取100条

| job_id                               | user_name      | job_name            | tool_name         | tool_version | tool_type |
|--------------------------------------|----------------|---------------------|-------------------|--------------|-----------|
| 4b682e15-ab92-11ee-a057-fa163ef319da |                | cli-demo-job        | cli-demo-job      | cp-test      | 2.0.0     |
| 58a8f13b-aaf3-11ee-a057-fa163ef319da | wwx-test-admin | cli-demo-job-import | cli-demo-workflow |              |           |
| e7e55c6e-aaf6-11ee-a057-fa163ef319da | wwx-test-admin | job-6685            | cli-demo-workflow | 4.0.0        |           |
| ae9e91a-aaf6-11ee-a057-fa163ef319da  | wwx-test-admin | job                 | cp-test           | 2.0.0        |           |
| 35ff73b3-aaf3-11ee-a057-fa163ef319da | wwx-test-admin | job                 | cp-test           | 2.0.0        |           |
| 24b72eee-aaf3-11ee-a057-fa163ef319da | wwx-test-admin | job                 | cp-test           | 2.0.0        |           |
| 4cce1fb-aaf2-11ee-a057-fa163ef319da  | wwx-test-admin | job                 | cp-test           | 2.0.0        |           |
| 70f1baa8-ab96-11ee-a057-fa163ef319da | wwx-test-admin | cli-demo-job        | cli-demo-job      | cp-test      | 2.0.0     |
| 6c6098f0-ab96-11ee-a057-fa163ef319da | wwx-test-admin | cli-demo-job        | cli-demo-job      | cp-test      | 2.0.0     |

health get job -j cli-demo-job

| job_id                               | user_name      | job_name     | tool_name    | tool_version | tool_type |
|--------------------------------------|----------------|--------------|--------------|--------------|-----------|
| 70f1baa8-ab96-11ee-a057-fa163ef319da | wwx-test-admin | cli-demo-job | cli-demo-job | cp-test      | 2.0.0     |
| 6c6098f0-ab96-11ee-a057-fa163ef319da | wwx-test-admin | cli-demo-job | cli-demo-job | cp-test      | 2.0.0     |

health get job -l 3 同 health 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

health get job -o 10 同 health 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

health 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。

health 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。

```
health 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 |
|----------------------------------------|-----------|----------------------|----------------|--------------|
| 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。

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

Task event list:

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

Task instances list:

| Name                                             | PodIP       | Node                | RestartCount | Request/ |
|--------------------------------------------------|-------------|---------------------|--------------|----------|
| task-1-test-bingfasmial-1-59620029ac100038-jkdpt | 172.16.3.37 | 192.168.54.255      | 0            | 1/1      |
|                                                  |             | 2022-06-24 16:37:20 |              | 1G/1G    |

- 获取作业某一task日志。

```
health 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。

```
health 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

```

## 10.4 删除指定作业

使用`delete`命令删除指定作业。

### 命令结构

```
health delete job ID [flags]
```

表 10-3 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    | 指定项目名。未填写则使用配置文件中的项目名。 |

### 命令示例

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

```

health delete job f17a3542-3f7c-11eb-868a-fa163e3ddbba1
返回结果如下
delete job success! job id : f17a3542-3f7c-11eb-868a-fa163e3ddbba1

```

## 10.5 重试作业

使用`retry`命令重试作业。

### 命令结构

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

表 10-4 参数说明

| 参数        | 简写 | 是否必选 | 描述                     |
|-----------|----|------|------------------------|
| job-id    | 无  | 是    | 作业ID。                  |
| --project | 无  | 否    | 指定项目名。未填写则使用配置文件中的项目名。 |

### 命令示例

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

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

## 10.6 取消作业

使用cancel命令取消或强制终止作业。

### 命令结构

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

表 10-5 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    | 指定项目名。未填写则使用配置文件中的项目名。 |

### 命令示例

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

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

## 10.7 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
health export job --ids <job-id> [flags]
使用job-id文件导出作业
health export job --file <job-id-file> [flags]
```

表 10-6 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。<br>导出多个job-id时，用分号(;) 隔开。<br>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。<br>文件内容格式：{"ids":"id1;id2;id3"}。    |
| --project   | 无  | 否    | 指定项目名。未填写则使用配置文件中的项目名。                             |

## 命令示例

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

- 使用job-id导出作业  
health export job -i f17a3542-3f7c...  
# 返回结果如下  
export job successfully!
- 使用job-id文件导出作业  
health export job -f /user/path/ids.txt  
# 返回结果如下  
export job successfully!

## 10.8 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

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

表 10-7 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    | 指定项目名。未填写则使用配置文件中的项目名。               |

## 命令示例

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

- 获取json格式实例详情

```
health 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

```
health 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

```
health 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:
 ...
```

# 11 Nexflow 流程管理命令

## 11.1 创建流程

### 功能描述

通过引用本地标准的workflow yaml文件创建workflow。

### 命令结构

```
health nexflow create workflow [params]
```

表 11-1 参数说明

| 参数            | 简写 | 是否必选 | 说明                          |
|---------------|----|------|-----------------------------|
| --workflow    | -w | 是    | 本地workflow文件路径，可以是zip或nf文件。 |
| --name        | -n | 是    | 流程名称。                       |
| --description | -d | 否    | workflow的详细描述信息。            |
| --labels      | -l | 否    | 流程标签列表。以","分隔，如："a,b"。      |
| --main_file   | -m | 否    | 流程主文件名，默认为：main.nf。         |
| --params      | -p | 否    | 本地json/yaml格式参数文件路径。        |

### 命令示例

```
health nexflow create workflow -n h2 -w "D:\testzip.zip" -d "h test" -l "a,b" -m "m.nf" -p "D:\param1.json"
{
 "id": "5b5cd09c-bc14-11ed-b7f6-fa163e504fdd"
}
```

```
create nextflow workflow h2 successful
```

## 11.2 修改流程

### 功能描述

修改指定workflow内容（名称不支持修改）。

### 命令结构

```
health nextflow edit workflow ID [params]
```

表 11-2 参数说明

| 参数            | 简写 | 是否必选 | 说明                          |
|---------------|----|------|-----------------------------|
| ID            | 无  | 是    | 流程id。                       |
| --workflow    | -w | 否    | 本地workflow文件路径，可以是zip或nf文件。 |
| --description | -d | 否    | workflow的详细描述信息。            |
| --labels      | -l | 否    | 流程标签列表，用","分割，如"a,b,c"。     |
| --main_file   | -m | 否    | 流程主文件名。                     |
| --params      | -p | 否    | 本地json/yaml格式参数文件路径。        |

### 命令示例

```
health nextflow edit workflow 5b5cd09c-bc14-11ed-b7f6-fa163e504fdd -w "D:\testzip.zip" -d "h test 2" -l "a,b,c" -m "m1.nf" -p "D:\param1.json"
```

```
edit nextflow workflow 5b5cd09c-bc14-11ed-b7f6-fa163e504fdd successful
```

## 11.3 查询流程详情

### 功能描述

查询指定workflow详情。

### 命令结构

```
health nextflow get workflow ID [flags]
```

表 11-3 参数说明

| 参数           | 简写  | 是否必选 | 说明                                     |
|--------------|-----|------|----------------------------------------|
| ID           | 不涉及 | 否    | 流程ID，指定后获取流程ID对应的流程详情。不指定时，获取的是流程列表。   |
| --param-file | -p  | 否    | 参数文件保存路径，以yaml格式保存。如果没指定，则不会显示和保存参数信息。 |

## 命令示例

```
health nexflow get workflow 550e8400-e29b-41d4-a716-446655440000 -p a.yaml

wirte nexflow workflow 5b5cd09c-bc14-11ed-b7f6-fa163e504fdd params file h3 success
{
 "create_time": "2023-03-06T11:45:13Z",
 "description": "h test 2",
 "id": "5b5cd09c-bc14-11ed-b7f6-fa163xxx",
 "labels": [
 "a",
 "b",
 "c"
],
 "main_file": "m1.nf",
 "name": "h3",
 "params_file": "h3-params.yaml",
 "source_resource_id": "",
 "update_time": "2023-03-06T11:51:13Z",
 "workflow_file": "testzip.zip",
 "workflow_file_url": "https://xxx/_nextflow_/assets/e6dcd289-dadb-48d9-b53b-e0c6c256932e/h3-compress/testzip.zip?AccessKeyId=VQ52M3Q714WPWPDU85XZ\u0026Expires=1678139581\u0026Signature=XmQTSYDCq2Xl2pVDVE4S%2BoMBH3M%3D"
}

health nexflow get workflow
{
 "workflows": [
 {
 "id": "5b5cd09c-bc14-11ed-b7f6-fa163exxx",
 "name": "h3",
 "description": "h test 2",
 "labels": [
 "a",
 "b",
 "c"
],
 "create_time": "2023-03-06T11:45:13Z",
 "update_time": "2023-03-06T11:51:13Z",
 "creator": "ei_eihealth_h00541446_01",
 "source_resource_id": ""
 },
 {
 "id": "cff53a66-b7d3-11ed-b7f6-fa163xxx",
 "name": "dsxs",
 "description": "xxx",
 "labels": [],
 "create_time": "2023-03-01T01:53:07Z",
 "update_time": "2023-03-01T01:53:07Z",
 "creator": "ei_eihealth_h00541446_01",
 "source_resource_id": ""
 }
]
}
```

```
],
 "count": 2
}
```

## 11.4 删除流程

### 功能描述

删除指定workflow。

### 命令结构

```
health nextflow delete workflow ID
```

表 11-4 参数说明

| 参数 | 简写  | 是否必选 | 说明   |
|----|-----|------|------|
| ID | 不涉及 | 是    | 模板id |

### 命令示例

```
health nextflow delete workflow 550e8400-e29b-41d4-a716-446655440000
```

```
delete nextflow workflow success,workflow id : 4f712af9-3f51-11eb-868a-fa163e3ddba1
```

# 12 Nextflow 作业管理命令

## 12.1 启动作业

### 功能描述

通过引用本地参数文件，启动nextflow作业。

### 命令结构

```
health nextflow create job [flags]
```

表 12-1 参数说明

| 参数            | 简写 | 是否必选 | 说明                               |
|---------------|----|------|----------------------------------|
| --workflow-id | -w | 是    | nextflow作业ID。                    |
| --name        | -n | 是    | 作业名称。                            |
| --description | -d | 否    | 作业的详细描述信息。                       |
| --labels      | -l | 否    | 作业标签列表。以","分隔，如："a,b"。           |
| --add-command | -a | 否    | 作业额外执行的命令，如：--param1 hello。      |
| --params      | -p | 否    | 本地json/yaml格式参数文件路径。             |
| --priority    | -o | 否    | 作业优先级，支持输入的参数为0-9，默认值为0，0为最低优先级。 |

### 命令示例

```
health nextflow create job -n "j1" -w "113b2ee6-bcb3-11ed-b7f6-fa163e504fdd" -d "htest" -l "a,b,c" -a "hello" -p "/param1.json"
```

```

输出结果:
{
 "id": "fe88c9fe-2623-4b08-8f75-df60bb216072"
}
create nextflow job j1 successful

```

## 12.2 查询作业

### 功能描述

查询指定job详情。

### 命令结构

```
health nextflow get job ID [flags]
```

表 12-2 参数说明

| 参数        | 简写  | 是否必选 | 说明                                                                                                                                                                                 |
|-----------|-----|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID        | 不涉及 | 否    | job id, 不指定job-id时, 列出当前project的全部job的基本信息。                                                                                                                                        |
| --type    | -t  | 否    | 查询类型, 默认detail。此参数用于获取作业列表。<br>取值: logs、detail、reports 。<br><ul style="list-style-type: none"> <li>logs: 获取job日志。</li> <li>detail: 获取job详情。</li> <li>reports: 获取job报告 。</li> </ul> |
| --sortDir | -d  | 否    | 排序规则, 按创建时间排序, 默认DESC。此参数用于获取作业列表。<br>取值: DESC、ASC。                                                                                                                                |
| --sortKey | -y  | 否    | 指定需要排序的key, 如: create_time。此参数用于获取作业列表。<br>取值:<br>create_time,finish_time,name,status,u ser_name,workflow_name                                                                     |
| --limit   | -l  | 否    | 代表当次请求获取的最大条数。此参数用于获取作业列表。<br>默认值: 100。                                                                                                                                            |
| --offset  | -o  | 否    | 代表从哪个位置开始。默认为0。此参数用于获取作业列表。                                                                                                                                                        |
| --jobName | -j  | 否    | 作业名。此参数用于获取作业列表。                                                                                                                                                                   |
| --labels  | -a  | 否    | 按标签筛选列表, 标签间用, 分隔。例子: label1,label2。此参数用于获取作业列表。                                                                                                                                   |

| 参数                | 简写 | 是否必选 | 说明                                                                                 |
|-------------------|----|------|------------------------------------------------------------------------------------|
| --status          | -s | 否    | 作业状态。此参数用于获取作业列表。<br>取值: CANCELLED, COMPLETED, FAILED, RUNNING, SUBMITTED, UNKNOWN |
| --workflowName    | -w | 否    | 流程名。此参数用于获取作业列表。                                                                   |
| --userName        | -u | 否    | 用户名。此参数用于获取作业列表。                                                                   |
| --createStartTime | -c | 否    | 最小创建时间, 格式: YYYY-MM-DD HH:MM:SS。例子: "2006-01-02 15:04:05"。此参数用于获取作业列表。             |
| --createEndTime   | -e | 否    | 最大创建时间, 格式: YYYY-MM-DD HH:MM:SS。例子: "2006-01-02 15:04:05"。此参数用于获取作业列表。             |
| --finishStartTime | -f | 否    | 最小完成时间, 格式: YYYY-MM-DD HH:MM:SS。例子: "2006-01-02 15:04:05"。此参数用于获取作业列表。             |
| --finishEndTime   | -m | 否    | 最大完成时间, 格式: YYYY-MM-DD HH:MM:SS。例子: "2006-01-02 15:04:05"。此参数用于获取作业列表。             |

## 命令示例

- 获取作业列表

```
health nextflow get job --limit 2 --offset 0 --sortDir DESC --sortKey create_time --jobName j1 --
labels "a" --status "SUBMITTED" --workflowName "h5" --userName "ei_eihealth_01" --
createStartTime "2023-03-07 07:47:54"
```

```
{
 "jobs": [
 {
 "id": "a37c22fc-0c8f-4ca2-a354-97f997a5ab96",
 "name": "j1",
 "description": "hstest",
 "labels": [
 "a",
 "b",
 "c"
],
 "status": "SUBMITTED",
 "has_ignore_failed_tasks": false,
 "create_time": "2023-03-07T07:47:54Z",
 "failed_message": "",
 "failed_reason": "",
 "user_name": "ei_eihealth_01",
 "workflow_name": "h5",
 "workflow_id": "113b2ee6-bcb3-11ed-b7f6-fa163e504fdd"
 }
],
 "count": 4
}
```



- 按创建时间由小到大获取作业列表

```
health nextflow get job --sortKey "create_time" --sortDir ASC
```

```
{
 "jobs": [
 {
 "id": "a37c22fc-0c8f-4ca2-a354-97f99xxx",
 "name": "j1",
 "description": "hstest",
 "labels": [
 "a",
 "b",
 "c"
],
 "status": "SUBMITTED",
 "has_ignore_failed_tasks": false,
 "create_time": "2023-03-07T07:47:54Z",
 "failed_message": "",
 "failed_reason": "",
 "user_name": "ei_eihealth_01",
 "workflow_name": "h5",
 "workflow_id": "113b2ee6-bcb3-11ed-b7f6-fa1xxx"
 },
 {
 "id": "020cf2a1-36ca-4c2e-9be6-aa070325f2e5",
 "name": "j1",
 "description": "hstest",
 "labels": [
 "d"
],
 "status": "SUBMITTED",
 "has_ignore_failed_tasks": false,
 "create_time": "2023-03-07T09:44:44Z",
 "failed_message": "",
 "failed_reason": "",
 "user_name": "ei_eihealth_01",
 "workflow_name": "h5",
 "workflow_id": "113b2ee6-bcb3-11ed-b7f6-fa163xxx"
 }
],
 "count": 4
}
```

- 获取作业日志

```
获取作业日志
```

```
health nextflow get job "a37c22fc-0c8f-4ca2-a354-97f997a5ab96" -t logs
```

```
{
 "count": 14,
 "logs": [
 "Mar-07 15:48:04.575 [main] DEBUG nextflow.cli.Launcher - $> nextflow -log ./e6dcd289-dadb-48d9-b53b-e0c6c256932e/q07ae83e-b12c-4137-b417-0fa3ac61094c/nextflow.log run ./e6dcd289-dadb-48d9-b53b-e0c6c256932e/q07ae83e-b12c-4137-b417-0fa3ac61094c/h5/m.nf -name q07ae83e-b12c-4137-b417-0fa3ac61094c -w /nextflow/_nextflow_work/_e6dcd289-dadb-48d9-b53b-e0c6c256932e/q07ae83e-b12c-4137-b417-0fa3ac61094c -c ./e6dcd289-dadb-48d9-b53b-e0c6c256932e/q07ae83e-b12c-4137-b417-0fa3ac61094c/nextflow.config -params-file ./e6dcd289-dadb-48d9-b53b-e0c6c256932e/q07ae83e-b12c-4137-b417-0fa3ac61094c/params.json -with-trace -with-report -with-timeline -with-dag -with-weblog 'http://127.0.0.1:8445/v1.0/nextflow/web-log'",
 "Mar-07 15:48:05.285 [main] INFO nextflow.cli.CmdRun - N E X T F L O W ~ version 22.10.6",
 "Mar-07 15:48:05.309 [main] DEBUG nextflow.plugin.PluginsFacade - Setting up plugin manager > mode=prod; embedded=false; plugins-dir=/home/service/.nextflow/plugins; core-plugins: nf-amazon@1.11.3,nf-azure@0.14.2,nf-codecommit@0.1.2,nf-console@1.0.4,nf-ga4gh@1.0.4,nf-google@1.4.5,nf-tower@1.5.6,nf-wave@0.5.3",
 "Mar-07 15:48:05.390 [main] INFO org.pf4j.DefaultPluginStatusProvider - Enabled plugins: []",
 "Mar-07 15:48:05.415 [main] INFO org.pf4j.DefaultPluginStatusProvider - Disabled plugins: []",
 "Mar-07 15:48:05.420 [main] INFO org.pf4j.DefaultPluginManager - PF4J version 3.4.1 in 'deployment' mode",
]
}
```

```

"Mar-07 15:48:05.718 [main] INFO org.pf4j.AbstractPluginManager - No plugins",
"Mar-07 15:48:05.869 [main] DEBUG nextflow.scm.ProviderConfig - Using SCM config
path: /home/service/.nextflow/scm",
"Mar-07 15:48:05.880 [main] DEBUG nextflow.cli.Launcher - Operation aborted",
"nextflow.exception.AbortOperationException: Cannot find script file: ./e6dcd289-
dadb-48d9-b53b-e0c6c256932e/q07ae83e-b12c-4137-b417-0fa3ac61094c/h5/m.nf",
"\tat nextflow.cli.CmdRun.getScriptFile(CmdRun.groovy:448)",
"\tat nextflow.cli.CmdRun.run(CmdRun.groovy:300)",
"\tat nextflow.cli.Launcher.run(Launcher.groovy:487)",
"\tat nextflow.cli.Launcher.main(Launcher.groovy:646)"
],
"download_url": "https://xxx/_nextflow_run_/e6dcd289-dadb-48d9-b53b-e0c6c256932e/
q07ae83e-b12c-4137-b417-0fa3ac61094c/nextflow.log?
AccessKeyId=VQ52M3Q7I4WPWPDUB5XZ&Expires=1678190993&Signature=oRiADod2TExnsP56Q9k
%2FSgm4h9g%3D"
}

```

- 获取作业详情

```

获取作业详情
health nextflow get job "a37c22fc-0c8f-4ca2-a354-97f997a5ab96"
{
 "id": "a37c22fc-0c8f-4ca2-a354-97f997axxx",
 "name": "j1",
 "description": "hstest",
 "labels": [
 "a",
 "b",
 "c"
],
 "status": "SUBMITTED",
 "has_ignore_failed_tasks": false,
 "create_time": "2023-03-07T07:47:54Z",
 "failed_message": "",
 "failed_reason": "",
 "workflow_name": "h5",
 "workflow_id": "113b2ee6-bcb3-11ed-b7f6-fa163e5xxx",
 "command_line": "nextflow run m.nf",
 "params": [],
 "config_context": ""
}

```

## 12.3 删除指定作业

### 功能描述

删除指定作业

### 命令结构

```
health nextflow delete job ID
```

表 12-3 参数说明

| 参数 | 简写  | 是否必选 | 说明   |
|----|-----|------|------|
| ID | 不涉及 | 是    | 作业id |

## 命令示例

```
health nextflow delete job f17a3542-3f7c-11eb-868a-fa163e3ddb1
delete job 3650d348-440e-4ee7-b043-a44481dc6211 success
```

## 12.4 重试作业

### 功能描述

重试作业。

### 命令结构

```
health nextflow retry job <job-id> [flags]
```

表 12-4 参数说明

| 参数        | 简写  | 是否必选 | 说明                  |
|-----------|-----|------|---------------------|
| job-id    | 不涉及 | 是    | 作业ID                |
| -- params | -p  | 否    | 本地json/yaml格式参数文件路径 |

### 命令示例

```
health nextflow retry job f17a3542-3f7c-11eb-868a-fa163e3ddb1
retry job 3650d348-440e-4ee7-b043-a44481dc6211 success
```

## 12.5 停止作业

### 功能描述

停止作业。

### 命令结构

```
health nextflow stop job <job-id> [flags]
```

表 12-5 参数说明

| 参数     | 简写  | 是否必选 | 说明   |
|--------|-----|------|------|
| job-id | 不涉及 | 是    | 作业id |

### 命令示例

```
health nextflow stop job f17a3542-3f7c-11eb-868a-fa163e3ddb1
stop job 3650d348-440e-4ee7-b043-a44481dc6211 success
```

## 12.6 获取任务信息

### 功能描述

获取任务相关信息，如任务日志、详情。

### 命令结构

```
health nextflow get task <task-id> --jobId <job-id> [flags]
```

表 12-6 参数说明

| 参数      | 简写  | 是否必选 | 说明                                                                                                                                             |
|---------|-----|------|------------------------------------------------------------------------------------------------------------------------------------------------|
| task-id | 不涉及 | 否    | 任务id，如果不传，那么获取的是任务列表。                                                                                                                          |
| --type  | -t  | 否    | 查询类型。<br>默认detail。<br>取值：logs、detail。<br><ul style="list-style-type: none"> <li>logs：获取task日志。</li> <li>detail：获取task详情，用于获取task列表。</li> </ul> |
| --jobId | -j  | 是    | 作业Id。                                                                                                                                          |

### 命令示例

- 获取task列表

```
health nextflow get task -j "7991e0b4-bffe-4166-ac2e-45a261592dcc"
{
 "tasks": [
 {
 "task_id": "1",
 "process": "readfile",
 "hash": "f2/a0f16c",
 "status": "RUNNING",
 "container": "swr.xxx/eihealth-0659ea54-5ae6-4e9d-b014-a2b041a26f62/
admet:2.0.0.1659942728536",
 "pod_name": "nf-f2a0f16c66bbe8e28824e8ccabbddd25",
 "submit": "2023-03-09T08:50:37Z"
 }
],
 "count": 1
}
```

- 获取task详情

```
health nextflow get task 1 -j "7991e0b4-bffe-4166-ac2e-45a261592dcc"
{
 "id": "1",
 "command": "\n echo /test;sleep 1000;echo \"end /test\";\n ",
 "status": "RUNNING",
 "exit": 2147483647,
 "work_dir": "/nextflow/___nextflow_work___/e6dcd289-dadb-48d9-b53b-e0c6c256932e/"
}
```

```
mb79e5e6-1b14-41b4-ac3b-e8fb0122d64d/f2/a0f16c66bbe8e28824e8ccabddd25",
 "module": [],
 "container": "swr.xxx/eihealth-0659ea54-5ae6-4e9d-b014-a2b041a26f62/
admet:2.0.0.1659942728536",
 "attempt": 1,
 "execution_time": {
 "submit": "2023-03-09T08:50:37Z",
 "start": "2023-03-09T08:50:40Z"
 },
 "resource_requested": {
 "container": "swr.xxx/eihealth-0659ea54-5ae6-4e9d-b014-a2b041a26f62/
admet:2.0.0.1659942728536",
 "cpus": 1
 },
 "resource_usage": {}
}
```

- 获取task日志

```
health nextflow get task 1 -j "7991e0b4-bffe-4166-ac2e-45a261592dcc" -t logs

{
 "count": 1,
 "logs": [
 "2023-03-09T08:50:44Z /test\n"
]
}
```

# 13 notebook 命令

## 13.1 创建 notebook

使用create命令创建notebook。

### 命令结构

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

表 13-1 参数说明

| 参数            | 简写 | 是否必选 | 说明                                                                                                                                                               |
|---------------|----|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| notebook-name | 无  | 是    | notebook名称。<br>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或者指定项目中镜像（格式为SourceProjectName/ ImageName:TagName）。<br>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。 |
| --flavor      | -f | 否    | flavor规格，格式为cpu:gpu:memory，不填默认为1:0:2，也即其中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]。                                                          |

| 参数             | 简写 | 是否必选 | 说明                                                                                 |
|----------------|----|------|------------------------------------------------------------------------------------|
| --storage-path | -s | 是    | notebook存储路径。<br>包含本项目桶最多挂载6个，不包含本项目桶最多挂载5个。<br>格式为ProjectName:/path，多个路径用分号(;)隔开。 |
| --project      | 无  | 否    | 指定项目名。未填写则使用配置文件中的项目名。                                                             |

## 命令示例

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

- 创建SYSTEM镜像类型的notebook

```
health 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

```
health 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
```

## 13.2 获取 notebook 信息

使用get命令获取notebook列表、详情。

### 命令结构

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

表 13-2 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    | 指定项目名。未填写则使用配置文件中的项目名。                            |

### 命令示例

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

- 获取notebook列表

```
health 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

```
health 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": 1.0,
 "gpu": 0.0,
 "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/health-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

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

## 13.3 删除 notebook

使用delete命令删除notebook。

### 命令结构

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



表 13-3 参数说明

| 参数          | 简写 | 是否必选 | 说明                     |
|-------------|----|------|------------------------|
| notebook-id | 无  | 是    | notebook id。           |
| --project   | 无  | 否    | 指定项目名。未填写则使用配置文件中的项目名。 |

## 命令示例

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

```
health delete notebook xxxx
返回结果如下
delete notebook xxxx successfully!
```

## 13.4 编辑 notebook

使用update或edit命令更新notebook描述信息。

### 命令结构

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

表 13-4 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eihealth-toolkit的使用过程，Linux和macOS环境使用方法基本相同，可参考。

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

## 13.5 启动 notebook

使用start命令启动notebook。

### 命令结构

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

表 13-5 参数说明

| 参数          | 简写 | 是否必选 | 说明                     |
|-------------|----|------|------------------------|
| notebook-id | 无  | 是    | notebook id。           |
| --project   | 无  | 否    | 指定项目名。未填写则使用配置文件中的项目名。 |

## 命令示例

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

```
health start notebook xxxx
返回结果如下
start notebook xxxx successfully!
```

## 13.6 停止 notebook

使用**stop**命令停止notebook。

## 命令结构

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

表 13-6 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eihealth-toolkit的使用过程，Linux和macOS环境使用方法基本相同，可参考。

```
health stop notebook xxxx
返回结果如下
stop notebook xxxx successfully!
```