

基因容器

CLI 参考

文档版本 01

发布日期 2020-10-10

华为技术有限公司



**版权所有 © 华为技术有限公司 2020。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，华为公司对本文档内容不做任何明示或默示的声明或保证。

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。

---

# 目录

---

<b>1 命令总览</b> .....	<b>1</b>
<b>2 安装命令行工具</b> .....	<b>3</b>
<b>3 示例：使用命令行执行简单脚本</b> .....	<b>7</b>
<b>4 gcs 命令详情</b> .....	<b>9</b>
4.1 gcs version.....	10
4.2 gcs config add.....	11
4.3 gcs config list.....	12
4.4 gcs get kubectl.....	13
4.5 gcs get enviroment.....	14
4.6 gcs set env.....	16
4.7 gcs get sfs.....	17
4.8 gcs set sfs.....	18
4.9 gcs get obscli.....	18
4.10 gcs obs upload.....	19
4.11 gcs get workflow.....	20
4.12 gcs create workflow.....	21
4.13 gcs del workflow.....	22
4.14 gcs update workflow.....	23
4.15 gcs sub workflow.....	24
4.16 gcs get execution.....	25
4.17 gcs create execution.....	27
4.18 gcs del execution.....	29
4.19 gcs set execution.....	30
4.20 gcs sub job.....	31
4.21 gcs sub repjob.....	33
4.22 gcs get tool.....	36
4.23 gcs sub wdl.....	38
4.24 gcs get wdl.....	39
4.25 gcs set wdl.....	40
4.26 get get fr.....	41
<b>5 config.ini 配置说明</b> .....	<b>42</b>

# 1 命令总览

## 基本概念

- workflow（模板）。即静态的模板内容。包括模板本身，以及对应的输入参数。
- execution（任务）。即模板对应的任务。基于一组输入参数，对某一workflow执行运行操作，得到的一组有序执行的任务。
- job（子任务）。即execution包含哪些 job，对应kubernetes的pods对象，可以通过kubectl get pods查看集群正在执行中的pods。
- repjob（并发子任务）。即并行执行一组job，这些job大部分参数相同，只有少数不一样，repjob可以将这样的多个job 归并成一个。
- wdl（Workflow Description Language，流程描述语言）。Cromwell引擎使用的流程描述语言。

## 注意事项

在使用命令行之前，请确保已配置AK/SK并赋予其可执行权限。

## 命令列表

表 1-1 命令列表

命令格式	功能描述
<code>gcs version</code>	获取命令行工具相关版本信息。
<code>gcs config add</code>	输入 AK/SK/region名称并选择计算集群，初始化命令行工具。该命令将信息储存在 ~/.gcs/config.ini 文件中。
<code>gcs config list</code>	显示存储在 ~/.gcs/config.ini 里面的 AK/SK/region 等信息。
<code>gcs get kubectl</code>	获取kubernetes 命令行工具kubectl 并根据对应集群进行配置。
<code>gcs get environment</code>	获取租户下的环境信息，包括集群名称、状态以及内存、CPU使用情况。
<code>gcs set env</code>	设置默认env，设置之后，后续的任务将默认投到改环境上。

命令格式	功能描述
<b>gcs get sfs</b>	获取当前设置环境对应的集群所关联的全部sfs pvc name。
<b>gcs set sfs</b>	设置环境默认使用的sfs pvc。
<b>gcs get obscli</b>	获取obs client 命令行工具并进行相关配置。
<b>gcs obs upload</b>	将本地文件上传到集群绑定的对象存储桶中。
<b>gcs get workflow</b>	查看全部的上传模板（YAML）ID、名称、使用者、最近上传时间。也可以根据 ID指定显示特定模板的信息。
<b>gcs create workflow</b>	上传workflow文件。
<b>gcs del workflow</b>	删除workflow模板。注意：执行前需要删除引用该模板的所有任务（execution）。
<b>gcs update workflow</b>	更新流程的相关信息。
<b>gcs sub workflow</b>	在集群中，根据模板内容，创建execution。
<b>gcs get execution</b>	查询全部的任务信息，也可以根据ID指定显示特定任务信息。
<b>gcs create execution</b>	根据workflow模板创建execution任务。
<b>gcs del execution</b>	删除某一任务。执行后需要等待半分钟，在此期间，需要保证 kubectl get pods中， pods 全部删除。
<b>gcs set execution</b>	设置execution启停状态。
<b>gcs sub job</b>	在集群中，执行单一 shell 脚本。
<b>gcs sub rejob</b>	在集群中，并发执行 shell 脚本中的逐行命令，脚本中每一行命令并发创建一个job。
<b>gcs get tool</b>	查询租户下的所有工具信息。
<b>gcs sub wdl</b>	提交单个WDL流程到Cromwell引擎。
<b>gcs get wdl</b>	查询提交的wdl流程执行结果。
<b>gcs set wdl</b>	终止指定的WDL流程。
<b>gcs get fr</b>	获取当前用户的CCI套餐包请况

# 2 安装命令行工具

## 下载命令行工具

基因容器命令行工具支持64位的Ubuntu、CentOS7及EulerOS系统，命令行工具无需安装，只需下载后将其放置在操作系统中的任意目录即可。

请登录[基因容器控制台](#)，在“总览”页面右侧的“工具”中下载“cloud-cli-linux-gcs-x.x.xx.zip”，下载包中的“gcs”即命令行工具。



## 配置环境变量

请将命令行工具所在路径设置为环境变量（在任意路径执行“`export PATH=DIR_TO_GCS:$PATH`”，其中加粗内容为命令行工具存放路径），并赋予其可执行权限（在任意路径执行“`chmod +x DIR_TO_GCS/gcs`”或在gcs命令行工具所在路径执行“`chmod +x gcs`”）。

## 配置代理（可选）

如果您所在的环境无法直接访问互联网，需要使用外网访问代理，请在环境变量中为命令行工具配置网络代理，以便命令行工具连接华为云环境。具体方法如下：

在任意路径执行如下命令。

```
export gcs_proxy=http://[username[:password]@]proxyURL:port/
```

其中加粗内容分别为：

- username，代理使用用户名。
- password，代理使用密码。本字段无需按照url规则转义。
- proxyURL，代理URL路径。
- port，代理URL端口。

## 配置命令行工具

**步骤1** 在[统一身份认证页面](#)获取 AK/SK。同一 AK/SK 只可以下载一次，请务必妥善保存。



**步骤2** 执行`gcs config add` 配置`ak,sk,region`，`ak,sk`获取方式见本小节上述步骤1，`region`可选值参见下表1，`region`如不设置，默认为`cn-north-1`。

```
gcs config add --ak MY_AK_VALUE --sk MY_SK_VALUE --region MY_REGION_VALUE
```

例如：

```
gcs config add --ak BQB6EAKLTHOCH3V7APCD --sk MLTKiwaAqBJQ6gImREvicGadKgckPjjiGyQhJxdJ --region cn-north-1
```

执行成功结果：

```
add ak succeed
add sk succeed
add region succeed
```

当前GCS支持的Region可以参考：



表 2-1 区域名称

区域	区域名称
华北-北京一	cn-north-1
华北-北京四	cn-north-4
华东-上海二	cn-east-2

----结束

# 3 示例：使用命令行执行简单脚本

基因容器命令行工具提供了上传原始数据和参考组数据至OBS桶内的功能。当然您还可以使用 OBS 客户端上传数据。

## 前提条件

请确保已完成命令行工具安装，且已创建集群资源。如未完成，请参见[2 安装命令行工具](#)完成操作。

## 上传本地数据文件

为了执行基因比对任务，需要将本地的原始数据文件及参考组文件上传至对象存储桶中。

基因容器服务提供原始测试数据供您使用，下载路径为：

- <http://opengene.org/data/R1.fq.gz>
- <http://opengene.org/data/R2.fq.gz>

此外，基因容器服务已在公共数据桶中上传hg19.fa\*参考组，您可以直接使用，无需下载或是上传至OBS桶中。公共参考组数据由基因容器自动挂载到对象存储的/ref目录下。

本示例中使用的是GCS提供的原始测试数据，将已下载到本地路径为“/LOCAL\_FILE”文件夹内的原始数据上传至OBS桶中的sample目录下。

如下示例以将原始数据文件传至 sample 目录，参考组文件传至 reference目录为例。执行以下命令上传基本文件，其中“LOCAL\_FILE”为本地文件路径。

```
gcs obs upload /LOCAL_FILE/R1.fq.gz --obs-path sample
gcs obs upload /LOCAL_FILE/R2.fq.gz --obs-path sample
gcs obs upload /LOCAL_FILE/hg19.fa --obs-path reference
gcs obs upload /LOCAL_FILE/hg19.fa.amb --obs-path reference
gcs obs upload /LOCAL_FILE/hg19.fa.ann --obs-path reference
gcs obs upload /LOCAL_FILE/hg19.fa.bwt --obs-path reference
gcs obs upload /LOCAL_FILE/hg19.fa.fai --obs-path reference
gcs obs upload /LOCAL_FILE/hg19.fa.pac --obs-path reference
gcs obs upload /LOCAL_FILE/hg19.fa.sa --obs-path reference
```

## 提交单个任务

使用命令行工具执行gcs sub job提交单个任务。执行单任务即指由基因容器服务在集群中选择某一个节点上的容器来运行输入参数FILENAME中的内容。

以执行脚本名为**bwa\_help.sh**为例，其中脚本中内容为查看bwa软件使用方法，并将输出重定向至/obs/log.err文件中。bwa软件为基因测序常用工具，作为公共工具集成在基因容器服务供您使用。

bwa\_help.sh文件中内容为：

```
#cat bwa_help.sh
sleep 5 && echo 123 >/obs/log.err
```

执行如下命令，为脚本bwa\_help.sh申请1G内存、0.5核CPU，并在基因容器中执行该脚本。

```
gcs sub job --memory 1g --cpu 0.5 --shell sh --tool bwa:0.7.12 bwa_help.sh
```

## 提交一组任务

以多个样本并行执行 bwa 软件使用方法为例。首先完成 [上传本地数据文件](#) 中的文件上传后，假设需要对上传的原始文件、参考基因组进行基因比对，有如下脚本 bwa\_mem.sh：

```
#cat bwa_mem.sh
fastq=$1
ref=$2

bwa mem $ref $fastq
```

此时，如果对这个脚本执行不同参数，可以额外写一个脚本 bwa\_mem\_work.sh。注意这里使用的 /ref 文件夹是由基因窗口提供的hg19.fa公共参考组数据，无需上传。

```
#cat bwa_mem_work.sh
sh bwa_mem.sh /obs/sample/R1.fq.gz /ref/reference_files/hg19.fa >/obs/sample1.sam
sh bwa_mem.sh /obs/sample/R2.fq.gz /ref/reference_files/hg19.fa >/obs/sample2.sam
```

若如[上传本地数据文件](#)所示，上传的自己本地文件，则这里 bwa\_mem\_work.sh 应写为：

```
#cat bwa_mem_work.sh
sh bwa_mem.sh /obs/sample/R1.fq.gz /obs/reference/hg19.fa >/obs/sample1.sam
sh bwa_mem.sh /obs/sample/R2.fq.gz /obs/reference/hg19.fa >/obs/sample2.sam
```

对这个额外脚本，需要使用 gcs sub repjob 投递。同样这里可以使用 bwa:0.7.12 这个工具：

```
gcs sub repjob -cpu 0.5 --memory 1g --shell sh --tool bwa:0.7.12 bwa_mem_work.sh
```

# 4 gcs 命令详情

---

- [4.1 gcs version](#)
- [4.2 gcs config add](#)
- [4.3 gcs config list](#)
- [4.4 gcs get kubectl](#)
- [4.5 gcs get enviroment](#)
- [4.6 gcs set env](#)
- [4.7 gcs get sfs](#)
- [4.8 gcs set sfs](#)
- [4.9 gcs get obscli](#)
- [4.10 gcs obs upload](#)
- [4.11 gcs get workflow](#)
- [4.12 gcs create workflow](#)
- [4.13 gcs del workflow](#)
- [4.14 gcs update workflow](#)
- [4.15 gcs sub workflow](#)
- [4.16 gcs get execution](#)
- [4.17 gcs create execution](#)
- [4.18 gcs del execution](#)
- [4.19 gcs set execution](#)
- [4.20 gcs sub job](#)
- [4.21 gcs sub repjob](#)
- [4.22 gcs get tool](#)
- [4.23 gcs sub wdl](#)

[4.24 gcs get wdl](#)

[4.25 gcs set wdl](#)

[4.26 get get fr](#)

## 4.1 gcs version

### 功能

查看命令行工具的软件版本以及编译平台。

### 语法

**gcs version**

version可以简写为ver，即：

**gcs ver**

### 参数说明

表 4-1 参数说明

参数名称	是否必选	简写	描述
--flags	否	-f	指定命令输出结果中所打印的基因容器logo的字母用组成字符。默认 ATGCU，可以输入任意由字母组成的字符串。
--update	否	-u	自动更新gcs命令行工具。

### 使用示例

查看软件版本：

```
gcs version
```

示例执行结果：

```
GCS:
Version : 3.0.9.1
Go version: go1.8.3
Git commit: a4a3fg7
Built : Fri Oct 20 17:56:04 UTC 2018
OS/Arch : linux/amd64

      .ATGCU!
    .!ATGC! !UATGC.
  .!UATGC!   !UATGC!.
.UATGC!     !UATGC!
UAT!.      !GCU
AT         .. GC.
!UA        !TGCUAT! !G!
CU.       !ATGCUA!TGCUATG!
.CU       !ATGCUA!!TG .CUATGCUA!
```

```

!#! .GCUATGCU. .ATG !CUA!TGC!
.UA! TGCU! !! ATG CUA !TG
!C! .UATGC! !! .. UAT. GC!
UA. TGC. UAT !! !! GCU !AT
GCU .ATG CUA! !TG. !. !CUA. .TG
!CU! !AT! !GCU !ATG .CUAT! GC!
.UAT GCUA. TG! CUA. TGCU !AT
GCU .AT!!!. !! .GCUAT! GC.
!UA!TGC. .UA !!. TGCU. ATG
CUATG CUA !!TGC! UA!
!TGCU!...ATGC!!!UATG! !CU!
ATGCUATGCUATGCUATGC UAT.
GCU! .....! .ATG
!CU !AT!
GCU ATG
CUAT!!!!!!!!!!!!GCU!
.ATGCUATGCUATGCUAT

.. . ...
!GCUA. !TGCUA! TGCUAT
.GC UA. !T! !GC U
.AT G!. CU AT.
.GC UA. TG !!. C .UA
!TGCU .!ATGC! !UATGCU
! !! !!!

```

## 4.2 gcs config add

### 功能

初始化命令行工具，在使用其它命令前，必需先执行本命令初始化命令行工具。该命令执行后，将生成命令行工具配置文件，路径为 `~/gcs/config.ini`，用于存储配置信息。配置文件的详细说明请参见[5 config.ini配置说明](#)。

若要修改配置文件，同样可以使用本命令，增加特定字段，指定任务执行时使用的存储、计算资源。执行本命令时，若 `~/gcs/config.ini` 文件已存在，则会根据命令的参数刷新配置文件中对应的参数。

### 语法

```
gcs config add [params]
```

### 参数说明

表 4-2 参数说明

参数名称	是否必选	简写	描述
--ak	否	-a	添加 AK。AK/ SK 在统一身份认证页面（ <a href="https://console.huaweicloud.com/iam/myCredential">https://console.huaweicloud.com/iam/myCredential</a> ）获取，仅可下载一次，请妥善保管。 <b>须知</b> 执行 <code>gcs config add</code> 命令初始化命令行工具时，该参数必须设置。

参数名称	是否必选	简写	描述
--sk	否	-s	添加 SK 。AK/ SK 在统一身份认证页面 ( https://console.huaweicloud.com/iam/myCredential ) 获取, 仅可下载一次, 请妥善保管。 <b>须知</b> 执行gcs config add命令初始化命令行工具时, 该参数必须设置。
--region	否	-r	服务区域名称。默认为华北区“cn-north-1”。
--sfs-pvc-name	否	-f	流程推荐使用 NAS 存储中间结果, NAS所对应的 PVC name。在基因容器页面, 推荐流程 > 开始测序, 运行某个流程时, 会提示新建存储, 继而显示“中间文件存储”所对应的 pvc-name。如果已有 NAS 存储, 直接使用对应 name 即可。 <b>须知</b> 执行gcs config add命令初始化命令行工具时, 该参数必须设置。
--cli-log-file	否	-l	日志路径, 不填写时默认为gcs命令行当前路径下 gcs-cli.log文件。
--workflow-files-dirpath	否	-w	自动化生成的 workflow 模板文件路径, 不填写时默认为gcs命令行当前路径workflows文件夹下。
--config-file	否	-n	用户指定的加载配置文件, 填写后将加载该路径指定的配置文件内容, 原配置文件内容失效, 不填写时默认为~/gcs/config.ini。

## 使用示例

配置参数:

```
gcs config add --ak MY_AK_VALUE --sk MY_SK_VALUE --region MY_REGION_VALUE --sfs-pvc-name cce-sfs-jl91vlyb-bl15
```

示例执行结果:

```
add ak succeed
add sk succeed
add region succeed
add sfs-pvc-name succeed配置参数:
```

## 4.3 gcs config list

```
gcs config list
```

### 功能

显示初始化命令行工具结果, 命令执行结果中的参数说明请参见[5 config.ini配置说明](#)。

## 语法

**gcs config list**

## 参数说明

无

## 使用示例

查看配置文件内容：

```
gcs config list
```

示例执行结果：

```
[common]
ak = MY_AK_VALUE
sk = MY_SK_VALUE
region = cn-north-1
cli-log-file =
workflow-files-dirpath =
gene-apiserver-addr =
iam-apiserver-addr =
config-file = /root/.gcs/config.ini
sfs-pvc-name = cce-sfs-jl91vlyb-bl15
cm-apiserver-addr =
obs-apiserver-addr =
```

## 4.4 gcs get kubectl

### 功能

下载kubectl命令行工具，并进行相关配置。

### 语法

**gcs get kubectl**

kubectl可以简写为kube，即：

**gcs get kube**

### 参数说明

无

### 注意事项

- 执行本命令前请先删除当前目录下当前目录下 kubectl\* 相关文件及目录。
- 命令执行完成后，根据使用场景，按需切换kubectl的访问模式。
  - VPC网络内接入访问请执行此命令。  
kubectl config use-context internal
  - 互联网接入访问请执行此命令（互联网访问前提是您已经绑定了弹性IP）  
kubectl config use-context external
- 设置完成后，可以通过以下命令查看kubernetes集群信息。



```
kubectl cluster-info
```

## 使用示例

```
gcs get kubectl
```

返回值:

```
start download kubectl https://cce-storage.obs.cn-north-1.myhuaweicloud.com/kubectl.zip
download kubectl.zip succeed
write kubeconfig.json succeed
```

## 4.5 gcs get environment

### 功能

获取当前用户下所有计算环境信息列表。

### 语法

```
gcs get environment [environmentID]
```

environment可以简写为env，即

```
gcs get env [environmentID]
```

### 参数说明

参数名称	是否必选	描述
environmentID	否	环境ID。当ID为空时，获取所有环境信息。
--source(-s)	否	环境类型，包括cromwell/cci/cce/sge/rampart五类，默认为cci

### 返回参数说明

参数	参数类型	描述
guid	string	环境ID
project_id	string	环境所属项目ID
domain_id	string	环境所属租户ID
name	string	环境名称
description	string	环境描述
status	string	环境运行状态
create_at	string	环境创建时间
update_at	string	环境更新时间

参数	参数类型	描述
cluster_id	string	环境所属集群ID
cluster_name	string	环境所属集群名称
cpu	int	环境Cpu核数
memory	int	环境内存量
stack_id	string	堆栈ID
is_gene_cluster	bool	是否为部署环境时新创建的集群
bucket_name	string	环境默认关联的桶名
default_env	bool	是否为默认环境
order_id	String	CCE集群的订单号
cromwell_secret	String	cromwell环境使用的密钥名称
cromwell_deployment	String	cromwell环境的deployment名称
cromwell_detail	String	cromwell环境详细信息
rds_instance_id	String	云数据库id
rds_database_name	String	云数据库的名称
environment_error	String	环境错误信息
rampart_deployment	String	rampart环境的deployment名称
rampart_detail	String	rampart环境的详细信息
enterprise_project:id	String	企业项目id
enterprise_project:name	String	企业项目名字

## 注意事项

集群资源不足，可以去云容器引擎（<https://console.huaweicloud.com/cce2.0/>）中，为集群添加更多、更高规格的服务器。



## 使用示例

查看集群环境：

```
gcs get environment -s "cce"
```

示例执行结果：

```
{
  "environments": [
    {
      "guid": "9f488195-92b5-bba8-7968-95ea81ac9560",
      "project_id": "b95272738324466d8fd7e8cb7ae85afa",
      "domain_id": "1c905dc27dc94aed90e637a07456de69",
      "name": "gcs-env-gcsdebug",
      "description": "Gene container environment form cce.",
      "status": "Running",
      "create_at": "2018-11-13T09:40:04Z",
      "update_at": "2018-11-19T03:20:41Z",
      "cluster_id": "0724c7ee-e726-11e8-9ec0-0255ac101e3e",
      "cluster_name": "gcsdebug",
      "cpu": 62,
      "memory": 132,
      "source": "cce",
      "stack_id": "",
      "is_gene_cluster": false,
      "bucket_name": "genomeref"
    }
  ],
  "total": 1
}
```

## 4.6 gcs set env

### 功能

设定environmentName参数的环境为命令行的默认环境。

## 语法

```
gcs set env environmentName
```

## 参数说明

表 4-3 参数说明

参数名称	是否必选	简写	描述
environment Name	是	不涉及	设置环境名称。环境名称可以通过 <a href="#">gcs get environment</a> 获取。

## 使用示例

- 设置默认环境为 “gcs-env-gcsdebug”

```
gcs set env gcs-env-gcsdebug
```

示例执行结果：

```
current environment is gcs-env-gcsdebug
```

```
please set sfs-pvc-name for environment gcs-env-gcsdebug
```

## 4.7 gcs get sfs

### 功能

获取默认环境下对应的sfs pvc name列表。

### 语法

```
gcs get sfs
```

### 参数说明

无

### 注意事项

无

### 使用示例

```
gcs get sfs
```

返回值：

```
[  
  "cce-sfs-import-jofkllwi-45rc"  
]
```

## 4.8 gcs set sfs

### 功能

设置环境默认使用的sfs pvc。

### 语法

```
gcs set sfs sfsPvcName
```

### 参数说明

表 4-4 参数说明

参数名称	是否必选	描述
sfsPvcName	是	环境默认使用的sfs pvc名称，可以通过 <a href="#">gcs get sfs</a> 获取sfs pvc name列表，从中指定默认使用的sfs pvc。

### 使用示例

```
gcs set sfs cce-sfs-xxx  
set environment gcs-env-gcsdebug default sfs-pvc-name cce-sfs-xxx succeed
```

## 4.9 gcs get obscli

### 功能

下载obs 命令行工具，并进行相关配置。

### 语法

```
gcs get obscli
```

obscli可以简写为obs，即：

```
gcs get obs
```

### 参数说明

无

### 注意事项

执行本命令前请先删除当前目录下obs\* 相关文件及目录。

### 使用示例

```
gcs get obscli
```

返回值:

```
start download obsCli http://static.huaweicloud.com/upload/files/obs/cmd.zip
download obsCli.zip succeed
config obs cli succeed
```

## 4.10 gcs obs upload

### 功能

上传文件到环境关联的obs桶中。

### 语法

**gcs obs upload** FILENAME [params]

upload可以简写为up, 即:

**gcs obs up** FILENAME [params]

### 参数说明

表 4-5 参数说明

参数名称	是否必选	简写	描述
FILENAME	是	不涉及	本地待上传的文件的的路径。
--obs-path	否	-o	本地待上传文件要上传到OBS桶中的目标路径。此处指OBS桶中的文件夹路径, 不是完整路径, 无需写桶名称。

### 注意事项

- 上传后未发现文件, 请在数据存储中, 确认是否使用了正确的桶。



- 不支持本地文件夹上传

## 使用示例

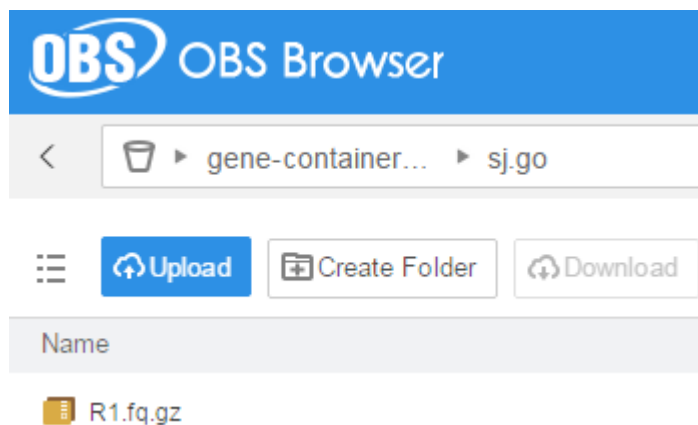
上传文件到OBS桶：

```
gcs obs upload R1.fq.gz --obs-path sample
```

示例执行结果：

```
upload file /root/CLionProjects/zlibfq/R1.fq.gz to obs sample succeed!
```

OBS 中显示上传文件：



## 4.11 gcs get workflow

### 功能

查看workflow信息。

### 语法

```
gcs get workflow WORKFLOWID [params]
```

workflow可以简写为wf，即：

```
gcs get wf WORKFLOWID [params]
```

### 参数说明

表 4-6 参数说明

参数名称	是否必选	简写	描述
WORKFLOW ID	否	不涉及	可以直接执行 <code>gcs get workflow</code> ，显示所有任务情况。也可以使用 <code>gcs get workflow workflowID</code> 指定显示单一workflow细节。
<code>--detail</code>	否	<code>-d</code>	默认值为不显示细节，如果希望显示 workflow 细节，则使用 <code>--detail</code> 。

## 使用示例

- 获取所有模板信息。

```
gcs get workflow
workflow_id      name      domain      last_update
xxxxxx          xxxx     xxxx        xxxx
xxxxxx          xxxx     xxxx        xxxx
xxxxxx          xxxx     xxxx        xxxx
330f6533-9024-3a11-ae94-a40f3d96d158 rep-demo gcs-developer 2018-08-25 09:35:33
xxxxxx          xxxx     xxxx        xxxx
xxxxxx          xxxx     xxxx        xxxx
```

- 获取所有模板详细信息。

```
gcs get workflow --detail
{
  "workflow": [
    {
      "id": "330f6533-9024-3a11-ae94-a40f3d96d158",
      "name": "gatk-best",
      "description": "",
      "vendor": "l00427178",
      "scope": "domain",
      "domain": "l00427178",
      "domain_id": "b0148a2cb4f347289b477bab429e48e7",
      "project_id": "c51567523b744d098a8a81ede51894ac",
      "create_at": "2018-08-25T09:35:33Z",
      "update_at": "2018-08-25T09:35:33Z",
      "workflow_url": "https://geneworkflows-149-
b0148a2cb4f347289b477bab429e48e7.192.149.46.200:443/gatk-best/gatk4-best-practices1.yaml",
      "logo": "",
      "enterprise_project": {"id": "0", "name": "default"}
    }
  ]
}
```

- 获取单一模板信息。

```
gcs get workflow 330f6533-9024-3a11-ae94-a40f3d96d158
{
  "workflow": {
    "id": "330f6533-9024-3a11-ae94-a40f3d96d158",
    "name": "gatk-best",
    "description": "",
    "vendor": "l00427178",
    "scope": "domain",
    "domain": "l00427178",
    "domain_id": "b0148a2cb4f347289b477bab429e48e7",
    "project_id": "c51567523b744d098a8a81ede51894ac",
    "create_at": "2018-08-25T09:35:33Z",
    "update_at": "2018-08-25T09:35:33Z",
    "workflow_url": "https://geneworkflows-149-
b0148a2cb4f347289b477bab429e48e7.192.149.46.200:443/gatk-best/gatk4-best-practices1.yaml",
    "logo": "",
    "image": "",
    "enterprise_project": {"id": "0", "name": "default"}
  }
}
```

## 4.12 gcs create workflow

### 功能

指定需要创建的模板名称，上传新的模板。同名模板上传两次会报错。

### 语法

```
gcs create workflow FILENAME [params]
```



workflow可以简写为wf，即：

```
gcs create wf WORKFLOWID [params]
```

## 参数说明

表 4-7 参数说明

参数名称	是否必选	简写	描述
FILENAME	是	不涉及	本地待上传workflow文件路径。
--name	否	-n	更新的模板名称。输入字符最大长度为64，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。默认为gcs-时间戳（YYYY-MMDD-hhmmss）格式。
--description	否	-d	对上传模板的描述。默认为空。
--enterprise_project_id	否	-e	上传workflow的企业项目id，如果为空，则默认为default企业项目（普通用户无需带此参数）

## 使用示例

创建 workflow。这里以 GATK 流程为例，首先，依据 [流程语法说明](#)，编写 YAML 文件。然后将模板上传至GCS，命名为 gatk-example：

```
gcs create workflow gatk4-example.yaml --name gatk-example --description "test " --enterprise_project_id "0"
```

创建 workflow 的示例执行结果：

```
create workflow succeedd !  
workflow id : 9b5e61a9-fdee-52cc-4ac8-dd07e1bf3ca6
```

## 4.13 gcs del workflow

### 功能

查看模板信息。

### 语法

```
gcs del workflow WORKFLOWID [WORKFLOWID2 WORKFLOWID3..]
```

workflow可以简写为wf，即：

```
gcs del wf WORKFLOWID [WORKFLOWID2 WORKFLOWID3..]
```

## 参数说明

表 4-8 参数说明

参数名称	描述
WORKFLOWID	待删除的workflow id。

## 注意事项

- 删除workflow 之前，需要删除所有引用该workflow 的 execution。
- 目前删除完成后，`gcs get workflow/execution` 仍会显示，但过一小段时间该 `workflow/execution` 将消失。需要使用 `kubectl` 命令具体确认任务状态。

## 使用示例

删除多个 workflow:

```
gcs del workflow 1d97cac3-7e45-2129-3314-115265abd91f 3b72d847-53c2-37ce-815e-d9558dd015ae
```

返回值:

```
delete workflow 1d97cac3-7e45-2129-3314-115265abd91f succeeded  
delete workflow 3b72d847-53c2-37ce-815e-d9558dd015ae succeeded
```

## 4.14 gcs update workflow

### 功能

更新流程的相关信息。

目前命令行只支持更改描述信息以及流程文件（已经使用中的流程不能更改流程文件）

### 语法

```
gcs update workflow workflow-id [params]
```

workflow可以简写为wf，即：

```
gcs update wf workflow-id [params]
```

### 参数说明

表 4-9 参数说明

参数名称	是否必选	简写	描述
workflow-id	是	不涉及	需要更改信息的流程的流程ID号。可以通过gcs get workflow获取。

参数名称	是否必选	简写	描述
--description	否	-d	要更新的workflow的描述信息。
--filepath	否	-f	要更新的workflow的文件。 <b>须知</b> 已经使用中的流程不能更改workflow文件。

## 使用示例

更新某个流程的描述信息

```
gcs update workflow 1ec895ca-18b3-162d-82d6-65e2d4955ca6 --description "t est for update"
update workflow succeed!
update workflow succeed! workflow id : 1ec895ca-18b3-162d-82d6-65e2d4955ca6
```

## 4.15 gcs sub workflow

### 功能

利用FILENAME参数提供的流程文件直接创建execution。

提示：该命令行不会新建workflow模板。如果已经有workflow模板，推荐直接使用[gcs create exection](#)。

### 语法

```
gcs sub workflow FILENAME [params]
```

workflow可以简写为wf，即：

```
gcs sub wf FILENAME [params]
```

### 参数说明

表 4-10 参数说明

参数名称	是否必选	描述
FILENAME	是	YAML 格式的模板文件，语法详见 <a href="#">流程语法参考</a> 。
--name	否	对此job指定名称。
--input	否	如果不希望使用 workflow YAML 文件默认的 inputs 值，可以在此指定一个 json 文件覆盖掉inputs输入。例如使用新样本时，可以在 json 文件中，定义新的样本名称。json文件示例：{"name":"gcsdemo","age":1}，其中"name"、"age"对应模板里的inputs参数名称。
--timeout	否	execution超时时间单位分钟，默认一天，1440分钟。

## 注意事项

- 可以在基因容器>示例流程 中，下载示例模板，并通过命令行提交。
- 示例流程中步骤直接由命令行书写。用户可以参考 [功能](#) 以及 [功能](#) 中的样例，调用 shell 脚本。
- 与 [功能](#) 以及 [功能](#) 中样例不同，**gcs sub workflow 命令，只上传 YAML 模板，不提供 shell 脚本上传功能。**推荐用户在调用 shell 脚本时，先使用 gcs sub job/ gcs sub repjob 命令单步测试通过，在此过程中完成 shell 脚本上传，再整合成一个 workflow 提交。

## 使用示例

首先在网页中，下载示例模板：



然后利用下载的示例模板创建execution：

```
gcs sub workflow gatk4-example.yaml --name gatk-example
```

注意这里使用正确的 YAML 模板路径，等待完成即可。结果可以在基因容器页面，执行结果中查看。

## 4.16 gcs get execution

### 功能

查看任务信息。

### 语法

```
gcs get execution EXECUTIONID [params]
```

execution可以简写为exec，即：

```
gcs get exec EXECUTIONID [params]
```

## 参数说明

表 4-11 参数说明

参数名称	是否必选	简写	描述
EXECUTIONID	否	不涉及	可以直接执行 <code>gcs get execution</code> ，显示所有任务情况。也可以 <code>gcs get execution EXECUTIONID</code> 指定显示单一 execution 细节。
--detail	否	-d	默认值为不显示细节，如果希望显示 execution 细节，则使用 --detail。

## 注意事项

- 可以直接执行 `gcs get execution`，显示所有任务情况。也可以 `gcs get execution executionID` 指定显示单一 execution 细节。
- 目前创建完成后，**Running** 状态只代表命令发给了集群，不代表程序已经开始。需要使用 `kubectl` 命令具体确认任务状态。

## 使用示例

- 获取所 任务信息。

```
gcs get execution
execution_id          execution_name
workflow_id          workflow_name      status      submit_time
duration submitter ep_name
3310a25e-fc1c-11e8-bcb5-0255ac10a106 deepvariant-12-10-093724
bbf1ffb1-7af4-7f15-70fd-c31cc4d711fc deepvariant      Succeeded    2018-12-10
09:37:48 0:11:41 default
9bf97d60-f9f3-11e8-bcb5-0255ac10a106
gcs-2018-1207-154212-57583
Failed 2018-12-07 15:42:12 23:59:52 root default
4dc64116-f911-11e8-bcb5-0255ac10a106 aaaa-12-06-124155
82afaafa-66cd-a5b8-ce9a-38c9e7ad0e5e aaaa      Succeeded    2018-12-06
12:42:14 0:22:03 default
189224b0-f911-11e8-bcb5-0255ac10a106 aaaa-12-06-124021
82afaafa-66cd-a5b8-ce9a-38c9e7ad0e5e aaaa      Succeeded    2018-12-06
12:40:45 0:10:41 default
e7644316-f910-11e8-bcb5-0255ac10a106 aaaa-12-06-123758
82afaafa-66cd-a5b8-ce9a-38c9e7ad0e5e aaaa      Succeeded    2018-12-06
12:39:23 0:10:41 default
afea104c-f910-11e8-bcb5-0255ac10a106 aaaa-12-06-123707
82afaafa-66cd-a5b8-ce9a-38c9e7ad0e5e aaaa      Succeeded    2018-12-06
12:37:50 0:10:31 default
7e8de091-f910-11e8-bcb5-0255ac10a106 aaaa2-12-06-123601
385346e4-8500-853e-9467-fb285c8f6caa aaaa2     Succeeded    2018-12-06
12:36:27 0:10:41 default
08dca565-f910-11e8-bcb5-0255ac10a106 aaaa-12-06-123234
82afaafa-66cd-a5b8-ce9a-38c9e7ad0e5e aaaa      Succeeded    2018-12-06
12:33:09 0:10:31 default
```

- 获取某个任务信息。

```
gcs get execution 2e1df238-adbd-11e8-bfb8-0255ac106796
job_name      namespace      status      submit_time      duration
bwa-mem-work-2018-0901-155928-2372c-0 default      Succeeded    2018-09-01 08:01:06
1260
bwa-mem-work-2018-0901-155928-2372c-1 default      Succeeded    2018-09-01 08:01:06
1350
```

- 获取所有任务详细信息。

```
gcs get execution --detail
{
  "execution": [
    {
      "guid": "2e1df238-adbd-11e8-bfb8-0255ac106796",
      "name": "gcs-2018-0901-155928-1-0-2018-0901-155928-86b7e",
      ...
    },
    {
      "id": "xxxx",
      ...
    }
  ],
  "enterprise_project": {
    "id": "ccc130bb-fa32-4af9-9370-7458470169cf",
    "name": "dts"
  }
}
```

- 获取某个任务详细信息。

```
gcs get execution 2e1df238-adbd-11e8-bfb8-0255ac106796 --detail
{
  "execution": {
    "guid": "2e1df238-adbd-11e8-bfb8-0255ac106796",
    "name": "gcs-2018-0901-155928-1-0-2018-0901-155928-86b7e",
    ...
  },
  {
    "id": "xxxx",
    ...
  },
  "enterprise_project": {
    "id": "ccc130bb-fa32-4af9-9370-7458470169cf",
    "name": "dts"
  }
}
```

## 4.17 gcs create execution

### 功能

创建execution。

### 语法

```
gcs create execution WORKFLOWID [params]
```

execution可以简写为exec，即：

```
gcs create exec WORKFLOWID [params]
```

## 参数说明

表 4-12 参数说明

参数名称	是否必选	简写	描述
WORKFLOWID	是	不涉及	待创建execution的workflow id。
--input	否	-i	用于创建execution的本地input.json文件路径，默认为空。
--name	否	-n	execution名称，输入字符最大长度为64字节，以小写字母开头，允许出现中划线(-)、数字和小写字母，且必须以小写字母或数字结尾。
--batch	否	-b	批量创建数量数量，如果batch 有定义具体数值且大于0 ,按照batch值进行创建,如果input长度小于batch 定义的数量，余下的execution则默认使用最后一个输入input文件，如果batch 为0,则取input中文件数量 为批量创建的数量,--input xx1,xx2,xx3,xx4，批量创建模式下，不支持自定义name,--name会变为前缀，比如--name test 则批量创建的execution名称为 ,name-execution-[1,2,3,4]，如果input 为空或者仅包含一个文件，且batch未指定，则认为单任务创建，否则为批量创建。
--timeout	否	-t	execution超时时间单位分钟，默认一天，1440分钟。
--email	否	-e	订阅通知消息的邮件地址，可以是多个，用逗号隔开。
--sms	否	-s	订阅通知消息的短信号码，可以是多个，用逗号隔开。
--events	否	-v	订阅通知消息的类型，exec_failed或者exec_succeeded，同时订阅用逗号隔开，默认是两个类型都订阅。
--policy	否	-p	镜像拉取策略参数，参数值可以是Always、Never和IfNotPresent，默认值是IfNotPresent。
--retry	否	-r	任务重试次数。参数值必须是0~9的整数。

## 注意事项

- 该命令的输入参数是 workflowID，可以通过 gcs get worflowID 查询后，根据 name/version获取。

- 目前创建完成后，Running状态只代表命令发给了集群，不代表程序已经开始。需要使用 `kubectl` 命令具体确认任务状态。
- `input` 文件是 json 格式，定义的变量值，需要与 流程模板中定义的 `inputs` 字段——对应。

## 使用示例

创建execution:

```
# 获取模板 ID
gcs get workflow
workflow_id      name      domain      last_update
xxxxxx          xxxx     xxxx        xxxx
xxxxxx          xxxx     xxxx        xxxx
xxxxxx          xxxx     xxxx        xxxx
330f6533-9024-3a11-ae94-a40f3d96d158 rep-demo  gcs-develper 2018-08-25 09:35:33
xxxxxx          xxxx     xxxx        xxxx
xxxxxx          xxxx     xxxx        xxxx

# 更换输入参数
cat run.json
{
  "GCS_DATA_PVC": "gene-pvc-gene-container-test",
  "GCS_REF_PVC": "gene-pvc-gcs-ref-southchina",
  "GCS_SFS_PVC": "gene-sfs-1111",
  "cpu": 1,
  "job-script": "run_work.sh",
  "memory": "0.5g",
  "shell": "sh"
}

# 使用新参数执行模板
gcs create execution 330f6533-9024-3a11-ae94-a40f3d96d158 --input run.json
```

示例执行结果:

```
create execution succeed
execution [0] : gcs-execution-2018-1007-145643 succeed
{
  "id": "573dcd38-c9fe-11e8-b18c-0255ac10a104"
}
batch create executions finish
```

## 4.18 gcs del execution

### 功能

查看任务信息。

### 语法

```
gcs del execution EXECUTIONID1 [EXECUTIONID2, EXECUTIONID3...]
```

`execution`可以简写为`exec`，即：

```
gcs del exec EXECUTIONID1 [EXECUTIONID2, EXECUTIONID3...]
```



## 参数说明

表 4-13 参数说明

参数名称	描述
EXECUTIONID	待删除的execution id。

## 注意事项

- 由于删除动作是异步的，所以删除完成后，`gcs get workflow/execution` 仍会显示，但稍后该 `workflow/execution` 将成功删除。需要使用 `kubectl` 命令具体确认任务状态。

## 使用示例

删除多个 execution:

```
gcs del execution aa1f8bdc-adb7-11e8-bfb8-0255ac106796 063417f0-ad8e-11e8-bebc-0255ac1066a1
```

返回值:

```
delete execution aa1f8bdc-adb7-11e8-bfb8-0255ac106796 succeeded  
delete execution 063417f0-ad8e-11e8-bebc-0255ac1066a1 succeeded
```

## 4.19 gcs set execution

### 功能

设置指定execution状态启停。

### 语法

```
gcs set execution executionId [params]
```

execution可以简写为exec，即：

```
gcs set exec executionId [params]
```

## 参数说明

表 4-14 参数说明

参数名称	是否必选	简写	描述
executionId	是	不涉及	execution guid，创建execution时生成，也可以通过 <a href="#">gcs get execution</a> 获取。

参数名称	是否必选	简写	描述
--action	是	-a	action 可以设置为： <ul style="list-style-type: none"> <li>cancel: 停止；</li> <li>retry: 重启，需要任务在失败情况触发。</li> </ul>

## 使用示例

```
gcs set execution xxxx --action cancel
current environment is gcs-env-gcsdebug
set execution xxxx cancel succeed
```

## 4.20 gcs sub job

### 功能

提交单个 shell 脚本，作为 job 在后台执行。

### 语法

**gcs sub job** FILENAME [params]

job可以简写为jb，即：

**gcs sub jb** FILENAME [params]

### 参数说明

表 4-15 参数说明

参数名称	是否必选	简写	描述
FILENAME	是	不涉及	执行job的脚本文件路径。 如果是cci环境，请将脚本上传到对应的sfs中，参数值使用sfs中的脚本的绝对路径。
--cpu	否	-c	所需CPU核数，单位为C。此处填写“数字”，数字支持小数，默认值为“1”。 <b>须知</b> 请确保请求CPU核数，小于容器集群中最大节点剩余核数大小。

参数名称	是否必选	简写	描述
--memory	否	-m	<p>所需内存资源大小，单位为G。此处填写“数字+单位”，默认值为“1G”。</p> <ul style="list-style-type: none"> <li>• 数字支持小数。</li> <li>• 单位为G或是g。</li> </ul> <p>例如，需要内存大小为4G，则此处可填写为“4G”或是“4g”。</p> <p><b>须知</b> 请确保请求内存量，小于容器集群中最大节点剩余内存大小。</p>
--shell	否	-s	job 使用的 Linux 脚本 shell 环境，如 sh / bash /zsh。默认值为“sh”。
--tool	否	-l	<p>job 使用的工具。工具支持公有或者私有容器镜像，为简化使用，基因容器服务工具仓库集成了常见镜像，您可以直接使用。</p> <p>默认值为“bwa:0.7.12”，即bwa工具0.7.12版本。</p>
--name	否	-n	对此 job 指定名称。
--timeout	否	-t	execution超时时间单位分钟，默认一天（1440分钟）。
--obs-path	否	-o	<p>支持多盘挂载，通过逗号分隔。</p> <p>--obs-path mountPath1:pvcName1,mountPath2:pvcName2</p> <p>mountPath 是容器内的挂载路径。 PvcName 是对应的pvc名称。</p>
--sfs-path	否	-f	<p>支持多盘挂载。</p> <p>--sfs-path mountPath1:pvcName1,mountPath2:pvcName2</p> <p>mountPath 是容器内的挂载路径。 PvcName 是对应的pvc名称。</p>
--ref-path	否	-r	<p>支持多盘挂载。</p> <p>--ref-path mountPath1:pvcName1,mountPath2:pvcName2</p> <p>mountPath 是容器内的挂载路径。 PvcName 是对应的pvc名称。</p>

## 使用示例

提交 `bwa_help.sh`，查看 `bwa` 软件版本，将结果重定向到 `/obs/log.err` 文件中。

```
# cat bwa_help.sh
# bwa -h > /obs/log.err

gcs sub job --cpu 1 --memory 1g --tool bwa:0.7.12 --shell sh bwa_help.sh
```

示例执行后，在OBS 桶中，`log.err` 文件，有如下内容：

```
Program: bwa (alignment via Burrows-Wheeler transformation)
Version: 0.7.12-r1039
Contact: Heng Li <lh3@sanger.ac.uk>

Usage: bwa <command> [options]

Command: index      index sequences in the FASTA format
mem          BWA-MEM algorithm
fastmap      identify super-maximal exact matches
permerge     merge overlapping paired ends (EXPERIMENTAL)
aln          gapped/ungapped alignment
samse        generate alignment (single ended)
sampe        generate alignment (paired ended)
bwasw        BWA-SW for long queries

shm          manage indices in shared memory
fa2pac       convert FASTA to PAC format
pac2bwt      generate BWT from PAC
pac2bwtgen   alternative algorithm for generating BWT
bwtupdate    update .bwt to the new format
bwt2sa       generate SA from BWT and Occ

Note: To use BWA, you need to first index the genome with `bwa index`.
There are three alignment algorithms in BWA: `mem`, `bwasw`, and
`aln/samse/sampe`. If you are not sure which to use, try `bwa mem`
first. Please `man ./bwa.1` for the manual.
```

## 4.21 gcs sub rejob

### 功能

提交一组并行的 `shell` 脚本，作为多个 `job` 在后台执行。

### 语法

```
gcs sub rejob FILENAME [params]
```

`rejob`可以简写为`rjb`，即：

```
gcs sub rjb FILENAME [params]
```

## 参数说明

表 4-16 参数说明

参数名称	是否必选	简写	描述
FILENAME	是	不涉及	<p>执行job的脚本文件路径。</p> <p>如果是cci环境，该脚本内的使用到的其他脚本需要上传到sfs上，这些脚本需要以sfs上的绝对路径的形式引用。</p>
--cpu	否	-c	<p>所需CPU核数，单位为C。此处填写“数字”，数字支持小数，默认值为“1”。</p> <p><b>须知</b> 请确保请求CPU核数，小于容器集群中最大节点剩余核数大小。</p>
--memory	否	-m	<p>所需内存资源大小，单位为G。此处填写“数字+单位”，默认值为“1G”。</p> <ul style="list-style-type: none"> <li>• 数字支持小数。</li> <li>• 单位为G或是g。</li> </ul> <p>例如，需要内存大小为4G，则此处可填写为“4G”或是“4g”。</p> <p><b>须知</b> 请确保请求内存量，小于容器集群中最大节点剩余内存大小。</p>
--shell	否	-s	<p>job 使用的 linux 脚本 shell 环境，如 sh / bash /zsh。默认值为“sh”。</p>
--tool	否	-l	<p>job 使用的工具。工具支持公有或者私有容器镜像，为简化使用，基因容器服务工具仓库集成了常见镜像，读者可以直接使用。</p> <p>默认使用“bwa:0.7.12”，安装了 bwa 0.7.12。</p>
--name	否	-n	<p>对此 job 指定名称。</p>
--timeout	否	-t	<p>execution超时时间单位分钟，默认一天（1440分钟）。</p>
--obs-path	否	-o	<p>支持多盘挂载，通过逗号分隔。</p> <p>--obs-path mountPath1:pvcName1,mountPath2:pvcName 2</p> <p>mountPath 是容器内的挂载路径。 PvcName 是对应的pvc名称。</p>

参数名称	是否必选	简写	描述
--sfs-path	否	-f	支持多盘挂载。 --obs-path mountPath1:pvcName1,mountPath2:pvcName 2 mountPath 是容器内的挂载路径。 PvcName 是对应的pvc名称。
--ref-path	否	-r	支持多盘挂载。 --obs-path mountPath1:pvcName1,mountPath2:pvcName 2 mountPath 是容器内的挂载路径。 PvcName 是对应的pvc名称。

## 注意事项

- 提交多个并行 shell 脚本具体格式如下：  
首先准备需要并行的单一脚本 bwa\_mem.sh，对 fastq 原始文件，执行 bwa mem 命令，进行基因序列比对。  

```
# cat bwa_mem.sh
fastq=$1
ref=$2

bwa mem $ref $fastq
```

然后准备调用单一脚本的并行脚本 bwa\_mem\_work.sh。其中，bwa\_mem.sh 可以是相对路径，要求与 bwa\_mem\_work.sh 在同一路径下，也可以是绝对路径。
- Tools 如果希望调用基因容器服务工具仓库，请在 <https://console.huaweicloud.com/gcs/?region=cn-north-1#/app/tools/list>，选择需要的软件。软件名称、版本，按照 [软件名称:软件版本] 格式填写即可。

## 使用示例

提交 bwa\_mem\_work.sh，执行 bwa 比对任务。

```
# cat bwa_mem_work.sh
sh bwa_mem.sh /obs/sample/R1.fq.gz /obs/reference/hg19.fa >/obs/R1.sam
sh /root/go/src/kubegene/cli/bin/bwa_mem.sh /obs/sample/R2.fq.gz /obs/reference/hg19.fa >/obs/R2.sam
```

示例执行结果在OBS 桶中，R1.sam、R2.sam 文件，如下所示。

<input type="checkbox"/>	R2.sam	标准存储	1.43 GB
<input type="checkbox"/>	R1.sam	标准存储	1.44 GB

## 4.22 gcs get tool

### 功能

获取所有tool、指定tool或者名称符合条件的tool的信息。

### 语法

```
gcs get tool TOOLID [params]
```

### 参数说明

表 4-17 参数说明

参数名称	是否必选	简写	描述
TOOLID	否	不涉及	需要获取的tool的ID。
--name	否	-n	需要获取的tool的名称，用于匹配符合条件的tool。
--exact	否	-e	查找名称时的匹配方式。指定--exact时，以完全匹配方式查找--name指定的tool。不指定是以部分匹配方式查找。

### 使用示例

- 获取所有模板信息。

```
gcs get tool
{
  "tools": [
    {
      "tool": [
        {
          "guid": "04e20b56-962d-24d8-0eb6-2c43ce35c957",
          "name": "nextdenovo",
          "description": "",
          "project_id": "b95272738324466d8fd7e8cb7ae85afa",
          "version": "v000",
          "scope": "domain",
          "domain": "tsjsdbd",
          "domain_id": "1c905dc27dc94aed90e637a07456de69",
          "vendor": "tsjsdbd",
          "image": "100.125.0.198:20202/hubqgene/nextdenovo:v0",
          "create_at": "2019-01-22T03:15:49Z",
          "update_at": "2019-01-22T03:15:49Z",
          "category": "environment",
          "instruction": "",
          "cpu": "2C",
          "memory": "8G",
          "command": "",
          "logo": "",
          "type": "container",
```

```

"enterprise_project": {
  "id": "ccc130bb-fa32-4af9-9370-7458470169cf",
  "name": "dts"
}
},
{
  "guid": "3a063961-27f5-8e8e-3200-4a45b744feaf",
  "name": "nextdenovo",
  "description": "",
  "project_id": "b95272738324466d8fd7e8cb7ae85afa",
  "version": "v0",
  "scope": "domain",
  "domain": "tsjsdbd",
  "domain_id": "1c905dc27dc94aed90e637a07456de69",
  "vendor": "huboqiang",
  "image": "100.125.0.198:20202/hubqgene/nextdenovo:v0",
  "create_at": "2019-01-09T05:40:15Z",
  "update_at": "2019-01-09T05:40:15Z",
  "category": "others",
  "instruction": "",
  "cpu": "2C",
  "memory": "8G",
  "command": "",
  "logo": "",
  "type": "container",
  "enterprise_project": {
    "id": "ccc130bb-fa32-4af9-9370-7458470169cf",
    "name": "dts"
  }
}
]
},
...

```

- 获取指定tool的信息

```
gcs get tool 04e20b56-962d-24d8-0eb6-2c43ce35c957
```

```

{
  "guid": "04e20b56-962d-24d8-0eb6-2c43ce35c957",
  "name": "nextdenovo",
  "description": "",
  "project_id": "b95272738324466d8fd7e8cb7ae85afa",
  "version": "v000",
  "scope": "domain",
  "domain": "tsjsdbd",
  "domain_id": "1c905dc27dc94aed90e637a07456de69",
  "vendor": "tsjsdbd",
  "image": "100.125.0.198:20202/hubqgene/nextdenovo:v0",
  "create_at": "2019-01-22T03:15:49Z",
  "update_at": "2019-01-22T03:15:49Z",
  "category": "environment",
  "instruction": "",
  "cpu": "2C",
  "memory": "8G",
  "command": "",
  "logo": "",
  "type": "container",
  "enterprise_project": {
    "id": "ccc130bb-fa32-4af9-9370-7458470169cf",
    "name": "dts"
  }
}
}

```

- 获取匹配某个名称的tool的信息

```
gcs get tool --name bwa
```

```

{
  "tools": [
    {
      "tool": [
        {
          "guid": "cafe1800-58f7-44e1-b73e-e26e29d68477",

```



```
"name": "bwa-sam-sambamba-bed",
"description": "",
"project_id": "b95272738324466d8fd7e8cb7ae85afa",
"version": "v0",
"scope": "domain",
"domain": "tsjsdbd",
"domain_id": "1c905dc27dc94aed90e637a07456de69",
"vendor": "tsjsdbd",
"image": "100.125.0.198:20202/hubqgene/bwa-sam-sambamba-bed:v0",
"create_at": "2018-12-13T08:15:43Z",
"update_at": "2018-12-13T08:15:43Z",
"category": "basic",
"instruction": "",
"cpu": "2C",
"memory": "8G",
"command": "",
"logo": "",
"type": "container",
"enterprise_project": {
  "id": "ccc130bb-fa32-4af9-9370-7458470169cf",
  "name": "dts"
}
}
]
}
],
"total": 1
}
```

## 4.23 gcs sub wdl

### 功能

提交单个WDL流程到Cromwell引擎。

### 语法

```
gcs sub wdl FILENAME [flags]
```

### 参数说明

表 4-18 参数说明

参数名称	是否必选	简写	描述
FILENAME	是	不涉及	需要执行的WDL流程文件。配置示例请参见 <a href="#">投递Cromwell任务</a> 。
--inputs	否	-i	流程输入文件。配置示例请参见 <a href="#">投递Cromwell任务</a> 。
--options	否	-o	流程运行时配置文件。配置示例请参见 <a href="#">投递Cromwell任务</a> 。
--depends	否	-d	依赖的子流程文件压缩包或者文件目录。

参数名称	是否必选	简写	描述
--server	否	-s	指定流程运行的环境名称，若不指定，则使用config中设置默认环境信息。

## 使用示例

提交wdl流程。

```
gcs sub wdl example.wdl -i example.inputs -s gcs-env-test
```

回显内容如下。

```
current environment is gcs-env-test
create wdl succeed
{
  "id":
  "aa215e0b-c896-4138-865e-6cf7921246b6",
  "status": "Submitted",
  "message": ""
}
```

## 4.24 gcs get wdl

### 功能

查询提交的wdl流程执行结果。

### 语法

```
gcs get wdl ID [flags]
```

### 参数说明

表 4-19 参数说明

参数名称	是否必选	简写	描述
ID	否	不涉及	指定要查询的执行结果的ID，若不指定则查询所有的执行结果。
--metadata	否	-m	查询指定执行结果的metadata信息，需要配合ID使用。

## 使用示例

- 查询单个执行结果详细信息。  
gcs get wdl aa215e0b-c896-4138-865e-6cf7921246b6  
返回内容如下。

```
{
  "execution": {
    "project_id": "",
    "domain_id": "",
    "guid": "aa215e0b-c896-4138-865e-6cf7921246b6",
    "name": "wf_chain-2019-11-05-15-34-35",
    "workflow_name": "wf_chain",
    "status": "Succeeded",
    "task_status": "",
    "metadata_json": {
      .....
    }
  }
}
```

- 查询单个执行结果的metadata信息。

```
gcs get wdl aa215e0b-c896-4138-865e-6cf7921246b6 -m
```

返回内容如下。

```
{
  "id": "aa215e0b-c896-4138-865e-6cf7921246b6",
  "workflowName": "wf_chain",
  "workflowRoot": "/sfs/wf_chain/aa215e0b-c896-4138-865e-6cf7921246b6",
  "submission": "2019-11-05T07:34:25.731Z",
  "start": "2019-11-05T07:34:41.184Z",
  "end": "2019-11-05T07:38:30.473Z",
  "status": "Succeeded",
  "labels": {
    "cromwell-workflow-id": "cromwell-aa215e0b-c896-4138-865e-6cf7921246b6"
  },
  "inputs": {
    "wf_chain.firstInput": ""
  },
  "outputs": {
    "wf_chain.result": ""
  },
  "calls": {}
}
```

## 4.25 gcs set wdl

### 功能

终止指定的WDL流程。

### 语法

```
gcs set wdl ID [flags]
```

### 参数说明

表 4-20 参数说明

参数名称	是否必选	简写	描述
ID	是	不涉及	指定要终止的执行结果的ID。

参数名称	是否必选	简写	描述
--action	是	-a	指定要执行的动作，当前仅支持 <b>abort</b> 。

## 使用示例

终止指定流程。

```
gcs set wdl aa215e0b-c896-4138-865e-6cf7921246b6 -a abort
```

返回内容如下。

```
set wdl f45e0304-77fb-46b5-a648-6234ecc001c9 abort succeed
```

## 4.26 get get fr

### 功能

获取CCI套餐包使用情况。

### 语法

```
gcs get fr
```

## 使用示例

查询套餐包情况。

```
gcs get fr
```

返回内容如下。

RESOURCE NAME TIME	RESOURCE TYPE REUSE CYCLE TYPE	RESOURCE INFO REUSE CYCLE	START TIME	END
CCI专业版资源套餐包 2020-07-26 23:59:59	包年 CPU资源	CCI资源套餐包 免费赠送CCI 0/50000(小时)	2019-07-26 14:13:06	
CCI专业版资源套餐包 2020-07-26 23:59:59	包年 内存资源	CCI资源套餐包 免费赠送CCI 74277.773/200000(小时)	2019-07-26 14:13:06	

### 📖 说明

- CPU资源单位为“核\*时”，上面回显中“0/50000”表示套餐包总共有50000核\*时，当前还剩余0核\*时可用。
- 内存资源单位为“GB\*时”，上面回显中“74277.773/200000”表示套餐包总共有200000GB\*时，当前还剩余74277.773GB\*时可用。

# 5 config.ini 配置说明

使用GCS使用gcs config add初始化后生成的配置文件~/gcs/config.ini，用于存储任务执行所涉及到的配置，如密钥、区域、集群信息、存储信息等。若控制界面修改了参数配置，请同时修改配置文件中的配置项，以确保命令行可以正确使用。修改配置文件，可以通过gcs config add命令或是手动修改config.ini文件，推荐通过gcs config add命令修改。

config.ini文件的配置参数说明请参见[5 config.ini配置说明](#)。

表 5-1 参数说明

参数名称	是否必选	描述
ak	必选	添加 AK 。AK/ SK 在统一身份认证页面（ <a href="https://console.huaweicloud.com/iam/myCredential">https://console.huaweicloud.com/iam/myCredential</a> ）获取，仅可下载一次，请妥善保管。
sk	必选	添加 SK 。AK/ SK 在统一身份认证页面（ <a href="https://console.huaweicloud.com/iam/myCredential">https://console.huaweicloud.com/iam/myCredential</a> ）获取，仅可下载一次，请妥善保管。
region	否	服务区域名称。默认为华北区 “ cn-north-1 ” 。 GCS当前支持的区域如下，请根据您所在区域就近选择： <ul style="list-style-type: none"><li>● 华北-北京： cn-north-1</li><li>● 华东-上海二： cn-east-2</li></ul>
sfs-pvc-name	必选	流程推荐使用 SFS 存储中间结果，SFS 所对应的 PVC name。在基因容器页面，推荐流程 > 开始测序，运行某个流程时，会提示新建存储，继而显示”中间文件存储”所对应的 pvc-name。如果已有 SFS 存储，直接使用对应 pvc-name 即可。
cli-log-file	否	日志路径，不填写时默认为gcs命令行当前路径下gcs-cli.log文件。
workflow-files-dirpath	否	自动化生成的 workflow 模板文件路径，不填写时默认为gcs命令行当前路径workflows文件夹下。

参数名称	是否必选	描述
config-file	否	用户指定的加载配置文件，填写后将加载该路径指定的配置文件内容，原配置文件内容失效，不填写时默认为~/gcs/config.ini。
gene-apiserver-addr	否	基因容器访问地址，无需修改，保持默认值即可。
iam-apiserver-addr	否	IAM访问地址，无需修改，保持默认值即可。
obs-apiserver-addr	否	IAM访问地址，无需修改，保持默认值即可。
cci-apiserver-addr	否	IAM访问地址，无需修改，保持默认值即可。
aom-apiserver-addr	否	AOM访问地址，无需修改，保持默认值即可。
cm-apiserver-addr	否	CCE集群访问地址，无需修改，保持默认值即可。