

AI 科学计算服务

常见问题

文档版本 01
发布日期 2025-07-08



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

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

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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

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

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 账号类	1
1.1 如何进行实名认证	1
2 认证信息类	3
2.1 如何获取项目 ID、项目、账号信息	3
2.2 如何获取 AK/SK	3
3 镜像类	5
3.1 如何搭建 Docker 环境	5
3.2 如何将生物信息学软件封装为镜像并上传	8
4 Notebook 类	13
4.1 如何使用 Notebook 的 Terminal 功能	13
4.2 如何将 Notebook 中的数据下载至本地	13
5 流程、作业类	16
5.1 应用的参数和镜像启动命令如何设置	16
5.2 直接挂载 OBS 目录进行大规模计算，如何解决偶现报错	16
5.3 作业日志不显示，如何配置 LTS	17
6 小分子药物设计类	19
6.1 为什么下载的部分靶点文件，显示不完整	19
6.2 CSS 常见问题	20

1 账号类

1.1 如何进行实名认证

实名认证入口

1. 使用已注册的华为账号登录“[华为云](#)”。
2. 单击账号名下拉框中的“账号中心”。
进入“账号中心”页面。

图 1-1 账号中心



3. 单击左侧导航中的“实名认证”，选择认证类型（个人认证/企业认证），根据页面提示进行实名认证。

图 1-2 实名认证



实名认证类型介绍

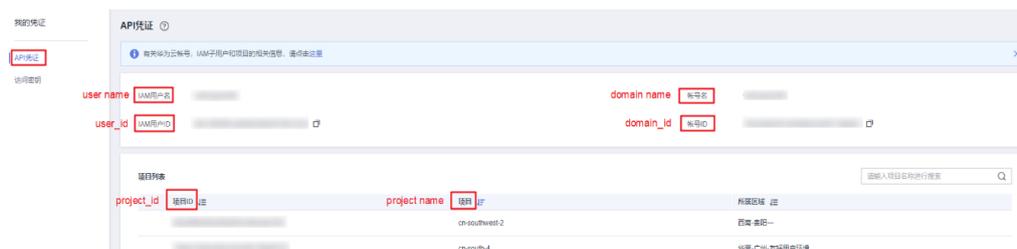
表 1-1 实名认证详细介绍

账号类型	认证类型（任选一种类型）	详细操作指导
个人账号	推荐 扫码认证（即时完成认证）	请参见 如何进行扫码认证 。
	银行卡认证（即时完成认证）	请参见 如何进行银行卡认证 。
	证件认证（1-3个工作日）	请参见 如何进行证件认证 。
企业账号	推荐 银行对公账户认证（最快30分钟）	请参见 如何进行企业银行对公账户打款认证 。
	企业证件认证（0-3个工作日）	请参见 如何进行企业证件认证 。

2 认证信息类

2.1 如何获取项目 ID、项目、账号信息

项目ID、项目名称、账号名（domain name）、账号id（domain id）可登录控制台“我的凭证”页面获取。



2.2 如何获取 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。

操作步骤

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。

3 镜像类

3.1 如何搭建 Docker 环境

制作Docker镜像，有以下两种方法。

- **快照方式制作镜像**（偶尔制作的镜像）：在基础镜像上，比如Ubuntu，先登录镜像系统并安装Docker软件，然后整体制作快照，即可得到所需软件的Docker镜像。
- **Dockerfile方式制作镜像**（经常更新的镜像）：将软件安装的流程写成DockerFile，使用Docker build构建成为Docker镜像。

📖 说明

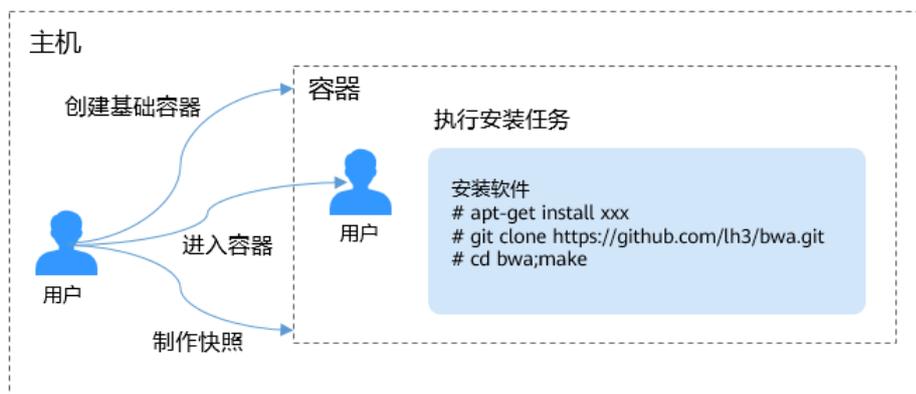
建议设置Docker镜像开机启动，防止出现“docker: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?”报错。

设置Docker开机启动命令：**systemctl enable docker**

启动Docker命令：**systemctl start docker**

快照方式制作镜像

如果后续镜像没有变化，可通过快照方式制作镜像。



快照方式制作镜像示例：

本示例中使用华为云**弹性云服务器（ECS）**创建一台云服务器，并使用快照方式制作bwa镜像。

1. 购买弹性云服务器。
2. 云服务器创建成功后，在**图3-1**页，选中待登录的弹性云服务器。单击“远程登录”，输入ECS初始账号，登录ECS。

图 3-1 云服务器列表



3. 安装容器引擎。

例如，在Linux操作系统下，可以使用如下命令快速安装容器引擎。

```
curl -fsSL get.docker.com -o get-docker.sh  
sh get-docker.sh
```

4. 启动一个空白的基础容器，并进入容器。

例如，启动一个CentOS容器。

```
docker run -it centos
```

5. 安装依赖包。

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm  
yum -y install git  
yum -y install gcc automake autoconf libtool make  
yum install -y zlib zlib-devel
```

6. 安装bwa软件，在github上下载bwa的源代码，并使用make编译。

```
yum install bwa  
git clone https://github.com/lh3/bwa.git  
cd bwa;make
```

📖 说明

请预先安装好Git，并检查本机是否有ssh key设置。

7. 输入**exit**退出容器。

8. 查询容器id。

```
docker ps -a
```

9. 制作快照。

```
docker commit -m "xx" -a "tsj" container-id tsj/image:tag
```

例如：`docker commit -m "test" -a "username" adb1127979a1 bwa:v0.7`

- `-a`：提交的镜像作者，例如tsj。
- `container-id`：容器id。
- `-m`：提交时的说明文字，例如xx。
- `tsj/image:tag`：仓库名/镜像名:TAG名，名称可自定义。

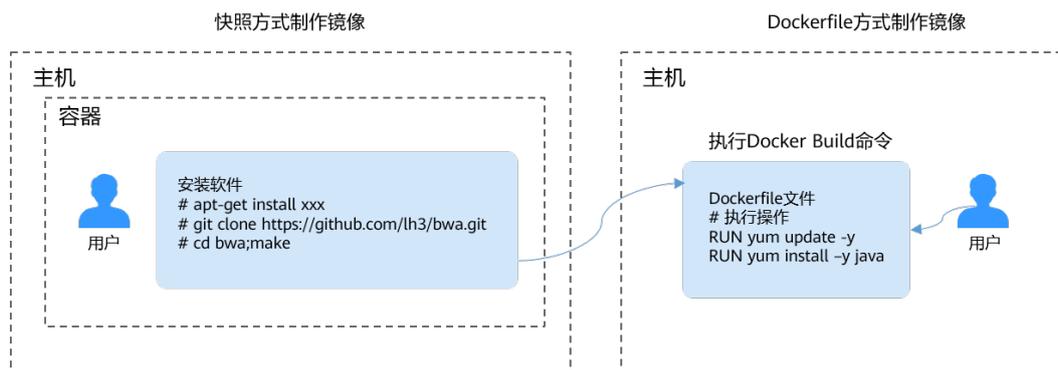
10. 执行**docker images**查看制作完成的Docker镜像。

Dockerfile 方式制作镜像

如果后续镜像经常变更（例如某个软件更新版本），建议使用Dockerfile方式制作镜像。如果采用快照方式制作镜像，则每次变更都需要执行操作命令，制作过程较为繁琐。

Dockerfile方式制作镜像是将快照制作的方式用Dockerfile文件写出来，然后执行 `docker build -t tsj/image:tag .`命令，自动完成镜像制作。

命令中“.”表示DockerFile文件的路径，“tsj/image:tag”表示仓库名/镜像名:TAG名。



Dockerfile方式制作镜像常用命令示例：

本例中使用华为云弹性云服务器（ECS）创建一台云服务器，制作centos镜像，并在镜像中放入图片文件。

1. [购买弹性云服务器](#)。
2. 云服务器创建成功后，在[图3-2](#)页，选中待登录的弹性云服务器。单击“远程登录”，输入ECS初始账号，登录ECS。

图 3-2 云服务器列表



3. 安装容器引擎。
例如，在Linux操作系统下，可以使用如下命令快速安装容器引擎。

```
curl -fsSL get.docker.com -o get-docker.sh  
sh get-docker.sh
```

4. 创建一个名为workdir的目录。
`mkdir workdir`
5. 在该目录中创建一个Dockerfile文件。
`cd ./workdir`
`touch Dockerfile`
6. 在该目录中创建一个空白文件abc.txt，创建一个webapp目录。
`touch abc.txt`
`mkdir webapp`

7. 在该目录中下载一个网络图片。
wget https://www.baidu.com/img/bd_logo1.png
8. 执行**vi Dockerfile**命令，进入Dockerfile文件中，编写文件。

```
#Version 1.0.1
FROM centos:latest
MAINTAINER ***u "***u@xxx.com"

#设置root用户为后续命令的执行者
USER root

#执行操作
RUN yum update -y
RUN yum install -y java

#使用&&拼接命令
RUN touch test.txt && echo "abc" >>abc.txt

#对外暴露端口
EXPOSE 80 8080 1038

#复制文件
COPY abc.txt /opt/

#复制文件夹
COPY /webapp /opt/webapp

#复制图片文件
COPY bd_logo1.png /opt/

#设置环境变量
ENV WEBAPP_PORT=9090

#设置工作目录
WORKDIR /opt/

#设置启动命令
ENTRYPOINT ["ls"]

#设置启动参数
CMD ["-a", "-l"]

#设置卷
VOLUME ["/data", "/var/www"]

#设置子镜像的触发操作，该命令在创建镜像时不会执行，只有在后续依照该镜像创建新镜像时才会执行
ONBUILD ADD . /app/src
ONBUILD RUN echo "on build excuted" >> onbuild.txt
```

9. 按Esc键，并执行:wq退出Dockerfile。

10. 制作镜像。
docker build -t centos:v01 .

详细的Dockerfile指令请参见[Dockerfile参考](#)、《[容器镜像服务 编写高效的Dockerfile](#)》。

3.2 如何将生物信息学软件封装为镜像并上传

本章节提供了在AI科学计算平台创建FastQC应用的样例，帮助您快速熟悉平台的使用方法。

FastQC是一款高通量序列数据的质量检查工具，此样例基于开源的FastQC软件，将软件制作成镜像，上传至平台，并基于此镜像创建应用。应用创建完成后可以直接使用FastQC应用，或将其编排至流程，和其他应用一起使用。

创建FastQC应用时，需要您熟悉该软件的使用，并具备一定的开发经验。FastQC的详细介绍请参考[FastQC介绍](#)。使用AI科学计算平台创建FastQC应用的详细步骤如下所示：

步骤1：搭建Docker环境

步骤2：制作镜像

步骤3：上传镜像

步骤4：创建应用

步骤 1：搭建 Docker 环境

1. 搭建Docker环境，您可以任选以下两种方式搭建Docker环境。

- 使用自己的电脑搭建Docker环境。
- 使用华为云弹性云服务器ECS搭建Docker环境。

本示例中使用华为云弹性云服务器ECS，并通过ECS搭建Docker环境。在创建ECS时，可以选择ECS的操作系统。例如，在Linux操作系统下，可以使用如下命令快速安装容器引擎。

```
curl -fsSL get.docker.com -o get-docker.sh  
sh get-docker.sh
```

2. 检查安装结果。

执行**docker --version**命令，如果显示如下类似信息，表示Docker安装成功。

图 3-3 Docker 安装成功

```
root@ecs-c134:~# docker --version  
Docker version 19.03.13, build 4484c46d9d
```

步骤 2：制作镜像

方法1：直接下载官方的FastQC镜像。

执行如下命令下载FastQC镜像。

```
docker pull biocontainers/fastqc:v0.11.5
```

方法2：通过Dockerfile制作FastQC镜像。

1. 执行**vi Dockerfile**命令，进入Dockerfile文件中，编写文件。

```
FROM ubuntu:16.04  
# FastQC依赖java运行，需安装java环境。安装执行下载、解压缩的软件包  
RUN apt-get update && apt-get upgrade -y \  
    && apt-get install -y default-jre perl wget zip  
# 下载FastQC，解压缩，设置FastQC可执行权限  
RUN wget https://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.11.5.zip \  
    && unzip fastqc_v0.11.5.zip \  
    && rm fastqc_v0.11.5.zip \  
    && chmod +x /FastQC/fastqc  
# 将FastQC添加到环境变量中  
ENV PATH "/FastQC:$PATH
```

2. 按Esc键，并执行**wq**退出Dockerfile。

3. 制作镜像。

```
docker build -t fastqc:v0.11.5 .
```

详细的Dockerfile指令请参见[Dockerfile参考](#)。

步骤 3：上传镜像

1. [下载命令行工具并进行初始化配置](#)。
2. 使用**ai4s switch project**命令进入到所需的空间中。

```
# 命令结构
ai4s switch project <project-name>
# 命令示例，例如进入到名为ai4s-user-project的空间中
ai4s switch project ai4s-user-project
```

3. 使用命令行工具上传镜像到AI科学计算平台。
 - a. 使用**ai4s docker tag**命令给要上传的镜像打标签。

```
ai4s docker tag biocontainers/fastqc:v0.11.5 fastqc:v0.11.5
```
 - b. 使用**ai4s docker push**命令上传镜像。

```
ai4s docker push fastqc:v0.11.5
```

执行**ai4s docker images**命令查看已有的镜像。

详细的命令介绍请参见“[命令行工具 > 镜像管理命令](#)”章节。

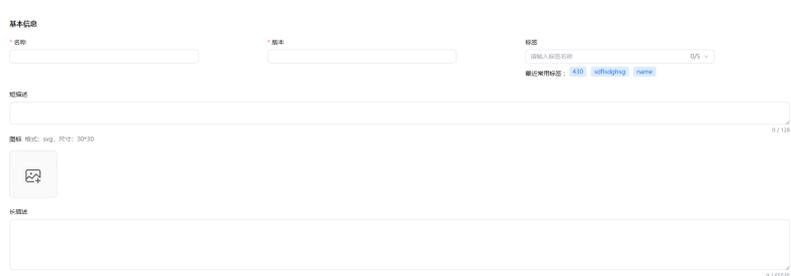
4. 登录AI科学计算平台，在镜像列表中查看已上传的镜像。

步骤 4：创建应用

1. 在“我的空间”页面“应用”页签中，单击“创建应用”。
2. 填写应用的基本信息。

“名称”填写fastqc，“版本”填写v0.11.5.2。“短描述”、“图标”、“长描述”、“标签”可选填。

图 3-4 填写基本信息



The screenshot shows a form titled '基本信息' (Basic Information) for creating an application. It contains several input fields: '名称' (Name) with a value of 'fastqc', '版本' (Version) with a value of 'v0.11.5.2', '短描述' (Short Description), '图标' (Icon) with a placeholder image, and '长描述' (Long Description). There are also buttons for '取消' (Cancel) and '确定' (Confirm).

3. 选择镜像。

单击“选择镜像”，在镜像列表中选择fastqc镜像和镜像版本。
4. 依据[FastQC命令说明](#)填写镜像启动命令。

镜像启动命令需要引用输入、输出参数中的变量，并以大括号扩起，以\$符号进行引用。

fastqc软件输入参数填写为input-file、threads，输出参数为output-dir，则镜像启动命令如下所示。

使用-t命令，指定运行所需的线程数量。-o命令，指定存放输出结果的文件夹。输入文件夹已在填写参数时指定。

```
fastqc -t ${threads} -o ${output-dir} ${input-file}
```
5. 选择“X86”CPU架构，CPU需求建议0.2起。GPU类型选择“无”。
6. 按需填写内存大小，单位为GB。FastQC运行中所需内存大小依赖于输入数据大小，建议至少1GB。

图 3-5 CPU、内存、GPU

^ 资源参数

CPU架构

X86 ARM

CPU需求

Memory需求

GB

GPU类型

无 GPU Snt9

GPU需求

0

计算节点标签 ?

请选择

7. 填写参数。

- a. 通过阅读[FastQC命令说明](#)，了解命令。

图 3-6 FastQC 命令

```
my $result = GetOptions('version' => \%$version,
                        'help' => \%$help,
                        'quiet' => \%$quiet,
                        'nogroup' => \%$nogroup,
                        'expgroup' => \%$expgroup,
                        'outdir=s' => \%$outdir,
                        'extract!' => \%$unzip,
                        'format=s' => \%$format,
                        'threads=i' => \%$threads,
                        'kmers=i' => \%$kmer_size,
                        'casava' => \%$casava,
                        'nano' => \%$nano,
                        'nofilter' => \%$nofilter,
                        'contaminants=s' => \%$contaminant,
                        'adapters=s' => \%$adapter,
                        'limits=s' => \%$limits,
                        'dir=s' => \%$temp_directory,
                        'javas' => \%$java_bin,
                        'min_length=i' => \%$min_length,
                        );

The options for the program as follows:
-h --help      Print this help file and exit
-v --version   Print the version of the program and exit
-o --outdir    Create all output files in the specified output directory.
               Please note that this directory must exist as the program
               will not create it. If this option is not set then the
               output file for each sequence file is created in the same
               directory as the sequence file which was processed.
--casava       Files come from raw casava output. Files in the same sample
               group (differing only by the group number) will be analysed
               as a set rather than individually. Sequences with the filter
               flag set in the header will be excluded from the analysis.
               Files must have the same names given to them by casava
               (including being gzipped and ending with .gz) otherwise they
               won't be grouped together correctly.
--nano         Files come from nanopore sequences and are in fast5 format. In
               this mode you can pass in directories to process and the program
               will take in all fast5 files within those directories and produce
               a single output file from the sequences found in all files.
--nofilter     If running with --casava then don't remove read flagged by
               casava as poor quality when performing the QC analysis.
```

- b. 填写所需的输入参数。

图 3-7 输入参数

c. 填写所需的输出参数。

因镜像启动命令中指定了输出参数，设置输出参数时，需勾选“必传”，并填写“默认值”。例如，输出结果默认存放在fastqc_output文件夹中。

图 3-8 输出参数



8. 单击“立即创建”，完成fastqc应用的创建。

创建完成后的应用，将显示在应用列表中，您可以使用该应用创建分析作业。

4 Notebook 类

4.1 如何使用 Notebook 的 Terminal 功能

对于习惯编码的开发者可以使用Terminal功能进行开发、调试和运行分析任务。

在“Files”页签下，单击右上角“New”，然后选择“Terminal”，进入Terminal界面。

图 4-1 Terminal



例如，您可以执行**wget**命令在公开数据集中下载基因组测序数据。

图 4-2 执行命令

```
root@notebook-3b8a3h:~# wget https://cg.10xgenomics.com/samples/genome/2.0.0/NA12878_WGS/NA12878_WGS_fastqs.tar
--2020-10-16 11:45:13-- https://cg.10xgenomics.com/samples/genome/2.0.0/NA12878_WGS/NA12878_WGS_fastqs.tar
Resolving cg.10xgenomics.com (cg.10xgenomics.com)... 104:18.0.173, 104:18.1.173, 2606:4700::6812:ad, ...
Connecting to cg.10xgenomics.com (cg.10xgenomics.com)|104.18.0.173|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 73424855040 (68G) [application/x-tar]
Saving to: 'NA12878_WGS_fastqs.tar'

NA12878_WGS_fastqs.tar      3%[>] 2.41G 16.8MB/s eta 85m 30s
```

4.2 如何将 Notebook 中的数据下载至本地

在Notebook页面，可以通过“Upload”和“Download”上传和下载文件。当在Notebook中上传文件提示大小受限时，您可以根据以下不同场景将大文件上传或下载到Notebook中。

图 4-3 上传和下载文件



OBS 存储类型的 Notebook

在创建 Notebook 时，如果“存储配置”选择的是“OBS”。Notebook 列表的所有文件读写操作是基于所选择的 OBS 路径下的内容操作，即 Notebook 中的数据 and OBS 中的数据是同步的。在 OBS 路径中创建文件夹、上传数据，会同步到 Notebook 中，Notebook 中的操作也会同步到 OBS 中，如图 4-4 所示。

图 4-4 通过 OBS 同步数据

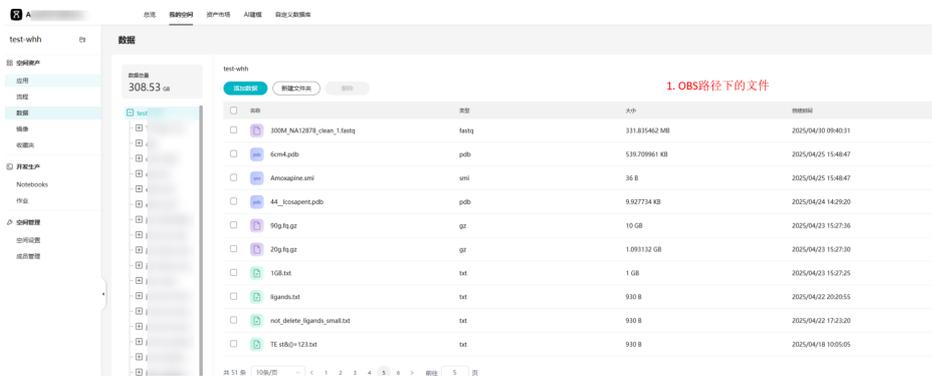
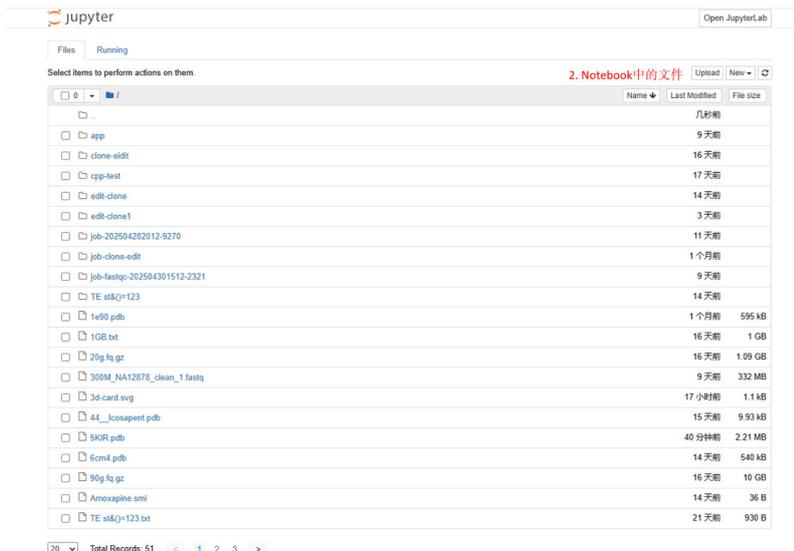


图 4-5 Notebook 中的文件



“Upload”上传数据大小受限时，您可以通过以下多种方式进行文件上传到 OBS 中，通过 OBS 与 Notebook 进行数据同步。

表 4-1 上传数据方法

上传方法	说明
“数据”页面上传	通过“数据”页面上传数据，支持上传最大为1GB的单个文件。
使用obsutil工具上传	您可以使用 obsutil工具 对OBS进行常用的配置管理操作，如创建桶、上传文件/文件夹、下载文件/文件夹、删除文件/文件夹等。

5 流程、作业类

5.1 应用的参数和镜像启动命令如何设置

创建应用时，需要设置应用的输入、输出参数和镜像的启动命令。需要您熟悉所制作的生物信息学软件的使用并具备一定的开发经验。

例如，设置FastQC应用的参数和镜像启动命令时，首先通过阅读[FastQC介绍](#)和[FastQC命令说明](#)了解软件的使用。并依照FastQC的调用命令设置参数和镜像启动命令。

图 5-1 FastQC 命令

```
my $result = GetOptions('version' => \%$version,
    'help' => \%$help,
    'quiet' => \%$quiet,
    'nogroup' => \%$nogroup,
    'exogroup' => \%$exogroup,
    'outdir' => \%$outdir,
    'extract!' => \%$unzip,
    'format!' => \%$format,
    'threads!' => \%$threads,
    'kmer!' => \%$kmer_size,
    'casava' => \%$casava,
    'nano' => \%$nano,
    'nofilter' => \%$nofilter,
    'contaminants!' => \%$contaminant
    'adapters!' => \%$adapters,
    'limits!' => \%$limits,
    'dir!' => \%$step_directory,
    'java!' => \%$java_bin,
    'min_length!' => \%$min_length,
    );

The options for the program as as follows:
-h --help      Print this help file and exit
-v --version   Print the version of the program and exit
-o --outdir    Create all output files in the specified output directory.
               Please note that this directory must exist as the program
               will not create it. If this option is not set then the
               output file for each sequence file is created in the same
               directory as the sequence file which was processed.
--casava       Files come from raw casava output. Files in the same sample
               group (differing only by the group number) will be analysed
               as a set rather than individually. Sequences with the filter
               flag set in the header will be excluded from the analysis.
               Files must have the same names given to them by casava
               (including being gzipped and ending with .gz) otherwise they
               won't be grouped together correctly.
--nano        Files come from nanopore sequences and are in fast5 format. In
               this mode you can pass in directories to process and the program
               will take in all fast5 files within those directories and produce
               a single output file from the sequences found in all files.
--nofilter     If running with --casava then don't remove read flagged by
               casava as poor quality when performing the QC analysis.
```

5.2 直接挂载 OBS 目录进行大规模计算，如何解决偶现报错

问题现象

运行作业时，作业直接挂载OBS目录进行大规模计算。偶现“异常应用”，并日志报错input/output error或file xxx not exists。

问题原因

OBS集群的IOPS达到了上限。

解决方案

- 更改分析存储介质，例如使用更高性能的IO加速方案（SFS Turbo、EVS），如使用SFS Turbo加速，在投递作业时可以选择“IO加速”。



The screenshot shows a configuration panel for a job. The 'IO Acceleration' (IO加速) option is highlighted with a red box. Other visible options include 'None' (无) and 'Local Disk Acceleration' (本地盘加速). The 'IO Acceleration' option is currently selected.

- 降低吞吐量运行，进而降低带宽、IO需求，使得带宽、IO满足生产需求。
- 优化软件算法，如使用内存做缓存等，降低软件的IO需求。
- 针对特定的项目，对带宽或IO性能进行扩容。
- 提交工单或联系服务技术支持。

5.3 作业日志不显示，如何配置 LTS

问题现象

作业完成时，作业日志显示为空。

问题原因

没有对接LTS日志采集配置。

解决方案

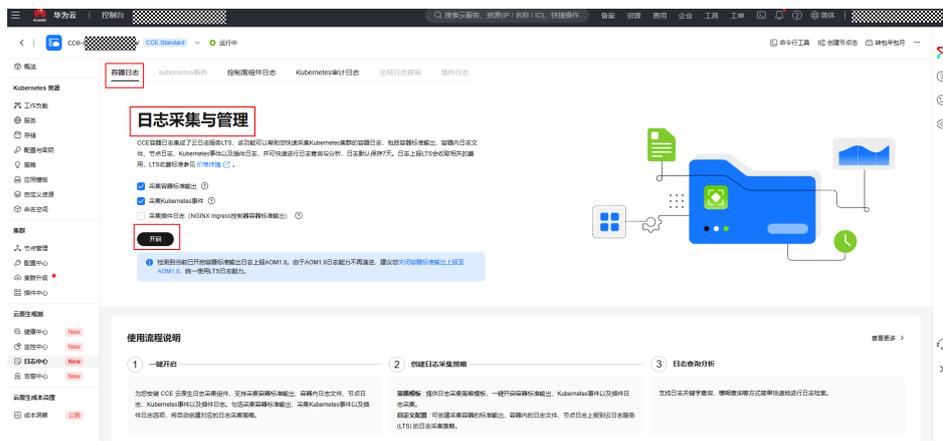
- 单击集群名称，找到日志中心-->授权说明弹窗，单击“确定”按钮。

图 5-2 授权说明



- 容器日志界面，日志采集与管理下方，单击“开启”按钮。

图 5-3 开启日志采集配置



说明

作业子任务Task运行时长小于10s（不区分任务状态），无日志输出。

6 小分子药物设计类

6.1 为什么下载的部分靶点文件，显示不完整

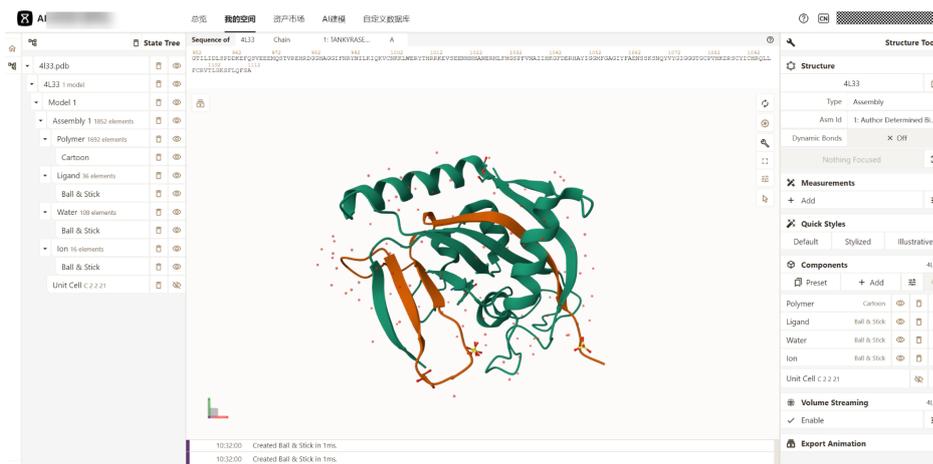
由于molstar插件自身问题，部分靶点文件中存在REMARK行，会导致受体展示不完整。可通过手动删除文件中REMARK行来解决该问题。

如下所示：

分子优化靶点设置界面，受体展示正常。



但是下载该靶点文件后，使用通用工具Mol 3D Viewer打开，会出现蛋白显示不完整的情况，如下图所示。



此时可将受体文件中的REMARK行进行删除，即可解决该问题。

6.2 CSS 常见问题

- 如果子用户看不到“CSS资源列表”，需要授权[子用户CSS权限](#)。
- 如果在创建和追加数据库的过程，显示资源不足，需要[扩容CSS集群](#)。
- 若CSS集群显示“不可绑定”。
 - 检查购买的CSS资源状态是不是正常的。
 - 检查购买的CSS资源是不是[绑定了公网IP](#)。
- 已绑定的集群修改了密码或者公网IP。
- 在平台侧需要解绑并重新绑定。
- CSS资源到期。
 - 在云搜索服务的[集群管理列表页](#)，找到需要续订的计费模式为“包年/包月”的集群。
 - 单击操作列的“更多 > 续费”，确定后进入续费页面。
 - 选择续费时长，并支付费用，完成续费。