# ModelArts

# 常见问题

| | |
|---|---|
| 文档版本 | 01 |
| 发布日期 | 2024-09-05 |

华为云计算技术有限公司

**商标声明**


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 一般性问题

## 1.1 什么是 ModelArts

ModelArts是面向AI开发者的一站式开发平台，提供海量数据预处理及半自动化标注、大规模分布式训练、自动化模型生成及模型按需部署能力，帮助用户快速创建和部署AI应用，管理全周期AI工作流。

“一站式”是指AI开发的各个环节，包括数据处理、算法开发、模型训练、创建AI应用、AI应用部署都可以在ModelArts上完成。从技术上看，ModelArts底层支持各种异构计算资源，开发者可以根据需要灵活选择使用，而不需要关心底层的技术。同时，ModelArts支持Tensorflow、MXNet等主流开源的AI开发框架，也支持开发者使用自研的算法框架，匹配您的使用习惯。

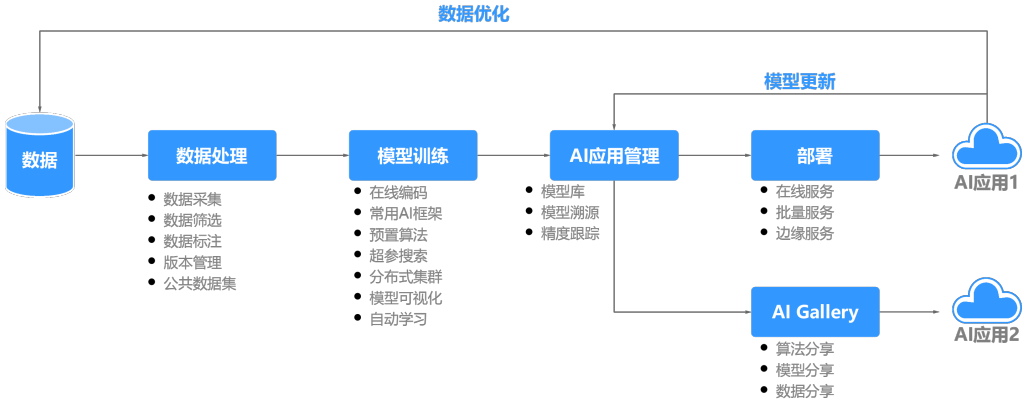
ModelArts的理念就是让AI开发变得更简单、更方便。面向不同经验的AI开发者，提供便捷易用的使用流程。例如，面向业务开发者，不需关注模型或编码，可使用自动学习流程快速构建AI应用；面向AI初学者，不需关注模型开发，使用预置算法构建AI应用；面向AI工程师，提供多种开发环境，多种操作流程和模式，方便开发者编码扩展，快速构建模型及应用。

### 产品架构

ModelArts是一个一站式的开发平台，能够支撑开发者从数据到AI应用的全流程开发过程。包含数据处理、模型训练、AI应用管理、AI应用部署等操作，并且提供AI Gallery功能，能够在市场内与其他开发者分享模型。

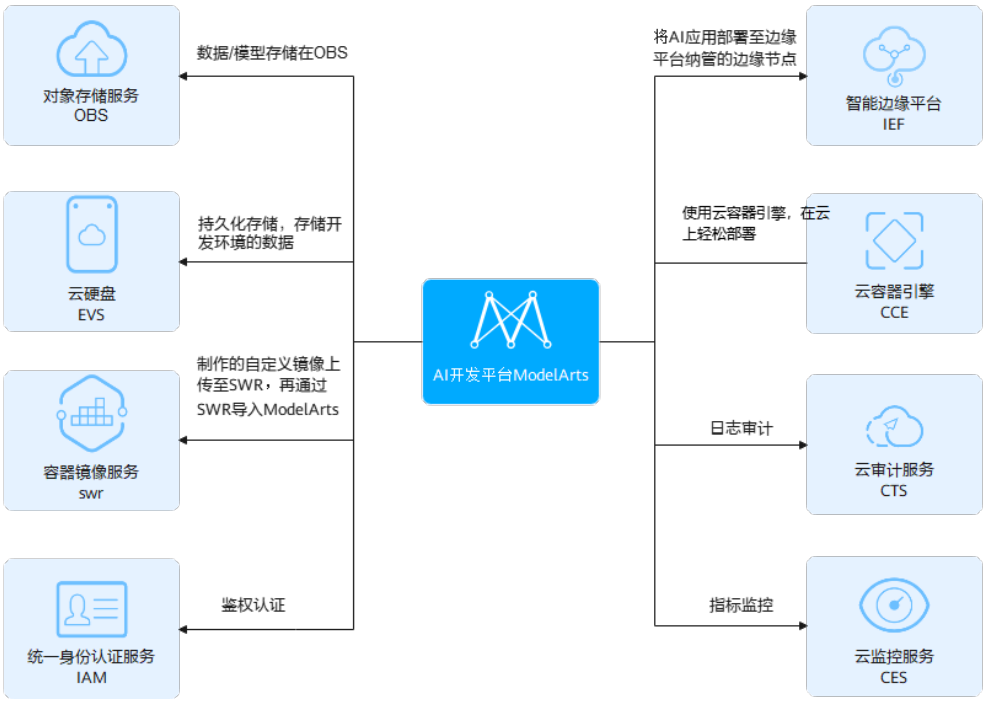
ModelArts支持图像分类、物体检测、视频分析、语音识别、产品推荐、异常检测等多种AI应用场景。

**图 1-1** ModelArts 架构




# 1.2 ModelArts 与其他服务的关系

**图 1-2** ModelArts 与其他服务的关系示意图




## 与统一身份认证服务的关系

ModelArts使用统一身份认证服务（Identity and Access Management，简称IAM）实现认证功能。IAM的更多信息请参见**《统一身份认证服务用户指南》**。

## 与对象存储服务的关系

ModelArts使用对象存储服务（Object Storage Service，简称OBS）存储数据和模型，实现安全、高可靠和低成本的存储需求。OBS的更多信息请参见**《对象存储服务控制台指南》**。

**表 1-1** ModelArts 各环节与 OBS 的关系

| 功能 | 子任务 | ModelArts与OBS的关系 |
| --- | --- | --- |
| 自动学习 | 数据标注 | ModelArts标注的数据存储在OBS中。 |
| | 自动训练 | 训练作业结束后，其生成的模型存储在OBS中。 |
| | 部署上线 | ModelArts将存储在OBS中的模型部署上线为在线服务。 |
| AI全流程开发 | 数据管理 | • 数据集存储在OBS中。<br>• 数据集的标注信息存储在OBS中。<br>• 支持从OBS中导入数据。 |
| | 开发环境 | Notebook实例中的数据或代码文件存储在OBS中。 |
| | 训练模型 | • 训练作业使用的数据集存储在OBS中。<br>• 训练作业的运行脚本存储在OBS中。<br>• 训练作业输出的模型存储在指定的OBS中。<br>• 训练作业的过程日志存储在指定的OBS中。 |
| | AI应用管理 | 训练作业结束后，其生成的模型存储在OBS中，创建AI应用时，从OBS中导入已有的模型文件。 |
| | 部署上线 | 将存储在OBS中的模型部署上线。 |
| 全局配置 | - | 获取访问授权（使用委托或访问密钥授权），以便ModelArts可以使用OBS存储数据、创建Notebook等操作。 |

## 与云硬盘的关系

ModelArts使用云硬盘服务（Elastic Volume Service，简称EVS）存储创建的Notebook实例。EVS的更多信息请参见《**云硬盘用户指南**》。

## 与云容器引擎的关系

ModelArts使用云容器引擎（Cloud Container Engine，简称CCE）部署模型为在线服务，支持服务的高并发和弹性伸缩需求。CCE的更多信息请参见**《云容器引擎用户指南》**。

## 与容器镜像服务的关系

当使用ModelArts不支持的AI框架构建模型时，可通过构建的自定义镜像导入ModelArts进行训练或推理。您可以通过容器镜像服务（Software Repository for Container，简称SWR）制作并上传自定义镜像，然后再通过容器镜像服务导入ModelArts。SWR的更多信息请参见**《容器镜像服务用户指南》**。

### 与智能边缘平台的关系

ModelArts可将模型部署至智能边缘平台（Intelligent EdgeFabric，简称IEF）纳管的边缘节点。IEF的更多信息请参见**《智能边缘平台用户指南》**。

### 与云监控的关系

ModelArts使用云监控服务（Cloud Eye Service， 简称CES）监控在线服务和对应模型负载，执行自动实时监控、告警和通知操作。CES的更多信息请参见**《云监控服务用户指南》**。

### 与云审计的关系

ModelArts使用云审计服务（Cloud Trace Service，简称CTS）记录ModelArts相关的操作事件，便于日后的查询、审计和回溯。CTS的更多信息请参见**《云审计服务指南》**。

## 1.3 ModelArts 与 DLS 服务的区别?

深度学习服务（DLS）是基于华为云强大高性能计算提供的一站式深度学习平台服务，内置大量优化的网络模型，以便捷、高效的方式帮助用户轻松使用深度学习技术，通过灵活调度按需服务化方式提供模型训练与评估。

但是，DLS服务仅提供深度学习技术，而ModelArts集成了深度学习和机器学习技术，同时ModelArts是一站式的AI开发平台，从数据标注、算法开发、模型训练及部署，管理全周期的AI流程。直白点解释，ModelArts包含并支持DLS中的功能特性。当前，DLS服务已从华为云下线，深度学习技术相关的功能可以直接在ModelArts中使用，如果您是DLS服务客户，也可以将DLS的数据迁移至ModelArts中使用。

## 1.4 如何购买或开通 ModelArts?

ModelArts是一个即开即用的平台，无需购买或开通，直接进入ModelArts管理控制台，完成全局配置，然后选择所需功能，直接使用即可。

ModelArts平台仅针对使用计算规格的功能才涉及计费，公共资源池全部为按需模式，根据选用规格以及作业运行时长收费。专属资源池可按需购买，也可选择包年包月购买，在运行训练作业或部署服务时，选择专属资源池，无需另外付费。

## 1.5 支持哪些型号的 Ascend 芯片?

目前支持Ascend Snt3和Snt9、Snt9、Snt9B、Snt9C。Ascend应用案例请参见**Ascend应用样例**。

- **模型训练**：ModelArts中支持使用Snt9、Snt9B训练模型。
- **模型推理**：在ModelArts中将模型部署上线为在线服务时，支持使用Snt3、Snt3P、Snt9、Snt9B规格资源进行模型推理。

# 1.6 如何获取访问密钥?

## 获取访问密钥

1. 登录华为云，在页面右上方单击“控制台”，进入华为云管理控制台。

   **图 1-3** 控制台入口

2. 在控制台右上角的账户名下方，单击“我的凭证”，进入“我的凭证”页面。

   **图 1-4** 我的凭证

3. 在“我的凭证”页面，选择“访问密钥>新增访问密钥”，如图1-5所示。

   **图 1-5** 单击新增访问密钥

4. 填写该密钥的描述说明，单击“确定”。根据提示单击“立即下载”，下载密钥。

**图 1-6** 新增访问密钥


5. 密钥文件会直接保存到浏览器默认的下载文件夹中。打开名称为“credentials.csv”的文件，即可查看访问密钥（Access Key Id和Secret Access Key）。

# 1.7 如何上传数据至 OBS?

使用ModelArts进行AI模型开发时，您需要将数据上传至对象存储服务（OBS）桶中。您可以登录OBS管理控制台创建OBS桶，并在您创建的OBS桶中创建文件夹，然后再进行数据的上传，OBS上传数据的详细操作请参见**《对象存储服务快速入门》**。

**说明**

- 您在创建OBS桶时，需保证您的OBS桶与ModelArts在同一个区域。如何查看OBS桶与ModelArts的所处区域，请参见**查看OBS桶与ModelArts是否在同一区域**。
- 建议根据业务情况及使用习惯，选择OBS使用方法。
  - 如果您的数据量较小（小于100MB）或数据文件少（少于100个），建议您使用控制台上传数据。控制台上传无需工具下载或多余配置，在少量数据上传时，更加便捷高效。
  - 如果您的数据量较大或数据文件较多，建议选择OBS Browser+或obsutil工具上传。OBS Browser+是一个比较常用的图形化工具，支持完善的桶管理和对象管理操作。推荐使用此工具创建桶或上传对象。obsutil是一款用于访问管理OBS的命令行工具，对于熟悉命令行程序的用户，obsutil是执行批量处理、自动化任务的好的选择。
  - 如果您的业务环境需要通过API或SDK执行数据上传操作，或者您习惯于使用API和SDK，推荐选择OBS的API或SDK方法创建桶和上传对象。

  上述说明仅罗列OBS常用的使用方式和工具，更多OBS工具说明，请参见《**OBS工具指南**》。

## 创建桶

桶是OBS中存储对象的容器，在上传对象前需要先创建桶。OBS提供多种使用方式，您可以根据使用习惯、业务场景选择不同的工具来创建桶。

**表 1-2** 不同访问方式创建桶的方法

| 访问方式 | 创建桶方法 |
| --- | --- |
| 控制台 | **通过控制台创建桶** |

| 访问方式 | 创建桶方法 |
| --- | --- |
| OBS Browser+ | 通过OBS Browser+创建桶 |
| obsutil | 通过obsutil创建桶 |
| SDK | 使用SDK创建桶，具体参考各语言开发指南的创建桶章节 |
| API | 通过API创建桶 |

### 上传对象

桶创建成功后，您可以通过以下多种方式将文件上传至桶，OBS最终将这些文件以对象的形式存储在桶中。

**表 1-3** 不同访问方式上传对象的方法

| 访问方式 | 上传对象方法 |
| --- | --- |
| 控制台 | 通过控制台上传对象 |
| OBS Browser+ | 通过OBS Browser+上传对象 |
| obsutil | 通过obsutil上传对象 |
| SDK | 使用SDK上传对象，具体参考各语言开发指南的上传对象章节 |
| API | PUT上传、POST上传 |

## 1.8 提示“上传的 AK/SK 不可用”，如何解决?

### 问题分析

AK与SK是用户访问OBS时需要使用的密钥对，AK与SK是一一对应，且一个AK唯一对应一个用户。如提示不可用，可能是由于账号欠费或AK与SK不正确等原因。

### 解决方案

1. 使用当前账号登录OBS管理控制台，确认当前账号是否能访问OBS。
   - 是，请执行步骤2。
   - 否，请执行步骤3。
2. 如能访问OBS，单击右上方登录的用户，在下拉列表中选择“我的凭证”。请根据“如何管理访问密钥”操作指导，确认当前AK/SK是否是当前账号创建的AK/SK。
   - 是，请联系提交工单处理。
   - 否，请根据“如何管理访问密钥”操作指导更换为当前账号的AK/SK。
3. 请确认当前账号是否欠费。

- 是，请给账号充值。操作指导请参见**账户充值**。
- 否，且提示资源已过保留期，需要**提工单**给OBS开通资源。

# 1.9 使用 ModelArts 时提示“权限不足”，如何解决?

当您使用ModelArts时如果提示权限不足，请您按照如下指导对相关服务和用户进行授权，并对用户权限进行检查操作。

由于ModelArts的使用权限依赖OBS服务的授权，您需要为用户授予OBS的系统权限。

- 如果您需要授予用户关于OBS的所有权限和ModelArts的基础操作权限，请参见**配置基础操作权限**。
- 如果您需要对用户使用OBS和ModelArts的权限进行精细化管理，进行自定义策略配置，请参见**创建ModelArts自定义策略**。

## 配置基础操作权限

使用ModelArts的基本功能，您需要为用户配置“作用范围”为“项目级服务”的“ModelArts CommonOperations”权限，由于ModelArts依赖OBS权限，您还**登录IAM管理控制台**需要为用户授予“作用范围”为“全局级服务”的“OBS Administrator”策略。

具体操作步骤如下：

**步骤1** 创建用户组。

**登录IAM管理控制台**，单击“用户组>创建用户组”。在“创建用户组”界面，输入“用户组名称”单击“确定”。

**步骤2** 配置用户组权限。

在用户组列表中，单击步骤1新建的用户组右侧的“授权”，在用户组“授权”页面，您需要配置的权限如下：

1. 配置“作用范围”为“项目级服务”的“ModelArts CommonOperations”权限，如下图所示，然后单击“确定”完成授权。

   **说明**

   区域级项目授权后只在授权区域生效，如果需要所有区域都生效，则所有区域都需要进行授权操作。

2. 配置“作用范围”为“全局级服务”的“OBS Administrator”权限，然后单击“确定”完成授权。

**步骤3** **创建用户并加入用户组**。

在IAM控制台创建用户，并将其加入步骤1中创建的用户组。

**步骤4** **用户登录**并验证权限。

新创建的用户登录控制台，切换至授权区域，验证权限：

- 在“服务列表”中选择ModelArts，进入ModelArts主界面，选择不同类型的专属资源池，在页面单击“创建”，如果无法进行创建（当前权限仅包含ModelArts CommonOperations），表“ModelArts CommonOperations”已生效。

- 在“服务列表”中选择除ModelArts外（假设当前策略仅包含ModelArts CommonOperations）的任一服务，如果提示权限不足，表示“ModelArts CommonOperations”已生效。
- 在“服务列表”中选择ModelArts，进入ModelArts主界面，单击“数据管理>数据集>创建数据 > 集”，如果可以成功访问对应的OBS路径，表示全局级服务的“OBS Administrator”已生效。

**----结束**

## 创建 ModelArts 自定义策略

如果系统预置的ModelArts权限不满足您的授权要求，或者您需要管理用户操作OBS的操作权限，可以创建自定义策略。更多关于创建自定义策略操作和参数说明请参见**创建自定义策略**。

目前华为云支持可视化视图创建自定义策略和JSON视图创建自定义策略，本章节将使用JSON视图方式的策略，以为ModelArts用户授予开发环境的使用权限并且配置ModelArts用户OBS相关的最小化权限项为例，指导您进行自定义策略配置。

**说明**

如果一个自定义策略中包含多个服务的授权语句，这些服务必须是同一属性，即都是全局级服务或者项目级服务。

由于OBS为全局服务，ModelArts为项目级服务，所以需要创建两条“作用范围”别为“全局级服务”以及“项目级服务”的自定义策略，然后将两条策略同时授予用户。

1. 创建ModelArts相关OBS的最小化权限的自定义策略。

   登录IAM控制台，在“权限管理>权限”页面，单击“创建自定义策略”。参数配置说明如下：

   - “策略名称”支持自定义。
   - “策略配置方式”为“JSON视图”。
   - “策略内容”请参见**ModelArts依赖的OBS权限自定义策略样例**，如果您需要了解更多关于OBS的系统权限，请参见**OBS权限管理**。

2. 创建ModelArts开发环境的使用权限的自定义策略。参数配置说明如下：

   - “策略名称”支持自定义。
   - “策略配置方式”为“JSON视图”。
   - “策略内容”请参见**ModelArts开发环境使用权限的自定义策略样例**，ModelArts自定义策略中可以添加的授权项（Action）请参见**《ModelArts API参考》>权限策略和授权项**。
   - 如果您需要对除ModelArts和OBS之外的其他服务授权，IAM支持服务的所有策略请参见**权限策略**。

3. 在IAM控制台**创建用户组并授权**。

   在IAM控制台创建用户组之后，将步骤**1**中创建的自定义策略授权给该用户组。

4. **创建用户并加入用户组**。

   在IAM控制台创建用户，并将其加入**3**中创建的用户组。

5. **用户登录**并验证权限。

   新创建的用户登录控制台，切换至授权区域，验证权限：

   - 在“服务列表”中选择ModelArts，进入ModelArts主界面，单击“数据管理 > 数据集”，如果无法进行创建（当前仅包含开发环境的使用权限），表示仅为ModelArts用户授予开发环境的使用权限已生效。

- 在“服务列表”中选择除ModelArts，进入ModelArts主界面，单击“开发环境>Notebook>创建”，如果可以成功访问“存储配置”项对应的OBS路径，表示为用户配置的OBS相关权限已生效。

## ModelArts 依赖的 OBS 权限自定义策略样例

如下示例为ModelArts依赖OBS服务的最小化权限项，包含OBS桶和OBS对象的权限。授予示例中的权限您可以通过ModelArts正常访问OBS不受限制。

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Action": [
            "obs:bucket:ListAllMybuckets",
            "obs:bucket:HeadBucket",
            "obs:bucket:ListBucket",
            "obs:bucket:GetBucketLocation",
            "obs:object:GetObject",
            "obs:object:GetObjectVersion",
            "obs:object:PutObject",
            "obs:object:DeleteObject",
            "obs:object:DeleteObjectVersion",
            "obs:object:ListMultipartUploadParts",
            "obs:object:AbortMultipartUpload",
            "obs:object:GetObjectAcl",
            "obs:object:GetObjectVersionAcl",
            "obs:bucket:PutBucketAcl",
            "obs:object:PutObjectAcl"
         ],
         "Effect": "Allow"
      }
   ]
}
```

## ModelArts 开发环境使用权限的自定义策略样例

```
{
   "Version": "1.1",
   "Statement": [

      {
         "Effect": "Allow",
         "Action": [
            "modelarts:notebook:list",
            "modelarts:notebook:create" ,
            "modelarts:notebook:get" ,
            "modelarts:notebook:update" ,
            "modelarts:notebook:delete" ,
            "modelarts:notebook:action" ,
            "modelarts:notebook:access"
         ]
      }
   ]
}
```

# 1.10 如何用 ModelArts 训练基于结构化数据的模型?

针对一般用户，ModelArts提供自动学习的预测分析场景来完成结构化数据的模型训练。

针对高阶用户，ModelArts在开发环境提供创建Notebook进行代码开发的功能，在训练作业提供创建大数据量训练任务的功能；用户在开发、训练流程中使用Scikit_Learn、XGBoost或Spark_MLlib引擎均可。

# 1.11 什么是区域、可用区?

## 什么是区域、可用区?

我们用区域和可用区来描述数据中心的位置，您可以在特定的区域、可用区创建资源。

- 区域（Region）：从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、VPC网络、弹性公网IP、镜像等公共服务。Region分为通用Region和专属Region，通用Region指面向公共租户提供通用云服务的Region；专属Region指只承载同一类业务或只面向特定租户提供业务服务的专用Region。
- 可用区（AZ，Availability Zone）：一个AZ是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。

图1-7阐明了区域和可用区之间的关系。

**图 1-7** 区域和可用区


目前，华为云已在全球多个地域开放云服务，您可以根据需求选择适合自己的区域和可用区。更多信息请参见**华为云全球站点**。

## 如何选择区域?

选择区域时，您需要考虑以下几个因素：

- 地理位置

  一般情况下，建议就近选择靠近您或者您的目标用户的区域，这样可以减少网络时延，提高访问速度。

  - 在除中国大陆以外的亚太地区有业务的用户，可以选择“中国-香港”、“亚太-曼谷”或“亚太-新加坡”区域。
  - 在非洲地区有业务的用户，可以选择“非洲-约翰内斯堡”区域。
  - 在欧洲地区有业务的用户，可以选择“欧洲-巴黎”区域。
  - 在拉丁美洲地区有业务的用户，可以选择“拉美-圣地亚哥”区域。

**说明**

“拉美-圣地亚哥”区域位于智利。

- 资源的价格

  不同区域的资源价格可能有差异，请参见**华为云服务价格详情**。

### 如何选择可用区?

是否将资源放在同一可用区内，主要取决于您对容灾能力和网络时延的要求。

- 如果您的应用需要较高的容灾能力，建议您将资源部署在同一区域的不同可用区内。
- 如果您的应用要求实例之间的网络延时较低，则建议您将资源创建在同一可用区内。

### 区域和终端节点

当您通过API使用资源时，您必须指定其区域终端节点。有关华为云的区域和终端节点的更多信息，请参阅**地区和终端节点**。

## 1.12 在 ModelArts 中如何查看 OBS 目录下的所有文件?

在使用Notebook或训练作业时，需要查看目录下的所有文件，您可以通过如下方式实现：

- 通过OBS管理控制台进行查看。

  使用当前账户登录OBS管理控制台，去查找对应的OBS桶、文件夹、文件。

- 通过接口判断路径是否存在。在已有的Notebook实例，或者创建一个Notebook，执行如下命令，检查路径是否存在。

  ```
  import moxing as mox
  mox.file.list_directory('obs://bucket_name', recursive=True)
  ```

  如果文件较多，请您耐心等待，最终文件路径信息会在提示信息之后显示。

## 1.13 ModelArts 数据集保存到容器的哪里?

ModelArts的数据集和数据存储位置对应的数据都保存在OBS中。

## 1.14 ModelArts 支持哪些 AI 框架?

ModelArts的开发环境Notebook、训练作业、模型推理（即AI应用管理和部署上线）支持的AI框架及其版本，不同模块的呈现方式存在细微差异，各模块支持的AI框架请参见如下描述。

### 统一镜像列表

ModelArts提供了ARM+Ascend规格的统一镜像，包括MindSpore、PyTorch。适用于开发环境，模型训练，服务部署，请参考**统一镜像列表**。**表1-4**、**表1-5所示镜像仅发布在西南-贵阳一区域**。

**表 1-4** MindSpore

| 预置镜像 | 适配芯片 | 适用范围 |
| --- | --- | --- |
| mindspore_2.2.0-cann_7.0.1-py_3.9-euler_2.10.7-aarch64-snt9b | Ascend snt9b | Notebook、训练、推理部署 |
| mindspore_2.1.0-cann_6.3.2-py_3.7-euler_2.10.7-aarch64-snt9b | Ascend snt9b | Notebook、训练、推理部署 |

**表 1-5** PyTorch

| 预置镜像 | 适配芯片 | 适用范围 |
| --- | --- | --- |
| pytorch_1.11.0-cann_6.3.2-py_3.7-euler_2.10.7-aarch64-snt9b | Ascend snt9b | Notebook、训练、推理部署 |

## 开发环境 Notebook

开发环境的Notebook，根据不同的工作环境，对应支持的镜像和版本有所不同。

**表 1-6** Notebook 支持的镜像

| 镜像名称 | 镜像描述 | 适配芯片 | 支持SSH远程开发访问 | 支持在线JupyterLab访问 |
| --- | --- | --- | --- | --- |
| pytorch1.8-cuda10.2-cudnn7-ubuntu18.04 | CPU、GPU通用算法开发和训练基础镜像，预置AI引擎PyTorch1.8 | CPU/GPU | 是 | 是 |
| mindspore1.7.0-cuda10.1-py3.7-ubuntu18.04 | CPU and GPU general algorithm development and training, preconfigured with AI engine MindSpore1.7.0 and cuda 10.1 | CPU/GPU | 是 | 是 |
| mindspore1.7.0-py3.7-ubuntu18.04 | CPU general algorithm development and training, preconfigured with AI engine MindSpore1.7.0 | CPU | 是 | 是 |

| 镜像名称 | 镜像描述 | 适配芯片 | 支持SSH远程开发访问 | 支持在线Jupyter Lab访问 |
| --- | --- | --- | --- | --- |
| pytorch1.10-cuda10.2-cudnn7-ubuntu18.04 | CPU and GPU general algorithm development and training, preconfigured with AI engine PyTorch1.10 and cuda10.2 | CPU/GPU | 是 | 是 |
| tensorflow2.1-cuda10.1-cudnn7-ubuntu18.04 | CPU、GPU通用算法开发和训练基础镜像，预置AI引擎TensorFlow2.1 | CPU/GPU | 是 | 是 |
| tensorflow1.13-cuda10.0-cudnn7-ubuntu18.04 | GPU通用算法开发和训练基础镜像，预置AI引擎TensorFlow1.13.1 | GPU | 是 | 是 |
| conda3-ubuntu18.04 | Clean user customized base image only include conda | CPU | 是 | 是 |
| pytorch1.4-cuda10.1-cudnn7-ubuntu18.04 | CPU、GPU通用算法开发和训练基础镜像，预置AI引擎PyTorch1.4 | CPU/GPU | 是 | 是 |
| conda3-cuda10.2-cudnn7-ubuntu18.04 | Clean user customized base image include cuda10.2, conda | CPU | 是 | 是 |
| tensorflow1.15-mindspore1.7.0-cann5.1.0-euler2.8-aarch64 | Ascend+ARM算法开发和训练基础镜像，AI引擎预置TensorFlow和MindSpore | Ascend | 是 | 是 |
| spark2.4.5-ubuntu18.04 | CPU algorithm development and training, prebuilt PySpark 2.4.5 and is able to attach to preconfigured spark cluster including MRS and DLI. | CPU | 否 | 是 |
| mindspore_1.10.0-cann_6.0.1-py_3.7-euler_2.8.3 | Ascend+ARM algorithm development and training. MindSpore is preset in the AI engine. | Ascend | 是 | 是 |

| 镜像名称 | 镜像描述 | 适配芯片 | 支持SSH远程开发访问 | 支持在线JupyterLab访问 |
| --- | --- | --- | --- | --- |
| mindspore_1.9.0-cann_6.0.0-py_3.7-euler_2.8.3 | Ascend+ARM algorithm development and training. MindSpore is preset in the AI engine. | Ascend | 是 | 是 |
| mindspore1.7.0-cann5.1.0-py3.7-euler2.8.3 | Ascend+ARM算法开发和训练基础镜像，AI引擎预置MindSpore | Ascend | 是 | 是 |
| tensorflow1.15-cann5.1.0-py3.7-euler2.8.3 | Ascend+ARM算法开发和训练基础镜像，AI引擎预置TensorFlow | Ascend | 是 | 是 |
| mindspore1.2.0-cuda10.1-cudnn7-ubuntu18.04 | GPU算法开发和训练基础镜像，预置AI引擎MindSpore-GPU | GPU | 是 | 是 |
| rlstudio1.0.0-ray1.3.0-cuda10.1-ubuntu18.04 | CPU、GPU强化学习算法开发和训练基础镜像，预置AI引擎 | CPU/GPU | 是 | 是 |
| mindquantum0.9.0-mindspore2.0.0-cuda11.6-ubuntu20.04 | MindSpore2.0.0 and MindQuantum0.9.0 | CPU | 是 | 是 |
| mindspore1.2.0-openmpi2.1.1-ubuntu18.04 | CPU算法开发和训练基础镜像，预置AI引擎MindSpore-CPU | CPU | 是 | 是 |
| cylp0.91.4-cbcpy2.10-ortools9.0-cplex20.1.0-ubuntu18.04 | CPU运筹优化求解器开发基础镜像，预置cylp，cbcpy，ortools及cplex | CPU | 是 | 是 |
| pytorch_2.1.0-cann_7.0.1.1-py_3.9-euler_2.10.7-aarch64-snt3p | Ascend_snt3p+ARM算法开发和训练基础镜像，AI引擎预置PyTorch2.1 | Ascend_snt3p | 是 | 是 |
| mindspore_2.2.12-cann_7.0.1.1-py_3.9-euler_2.10.7-aarch64-snt3p | IMAGE_MINDSPORE_ASCEND_310P_DESC | Ascend_snt3p | 是 | 是 |

## 训练作业

创建训练作业时，训练支持的AI引擎及对应版本如下所示。

预置引擎命名格式如下：

```
<训练引擎名称_版本号>-[cpu | <cuda_版本号 | cann_版本号 >]-<py_版本号>-<操作系统名称_版本号>-<x86_64 | aarch64>
```

**表 1-7** 训练作业支持的 AI 引擎

| 工作环境 | 系统架构 | 系统版本 | AI引擎与版本 | 支持的cuda或Ascend版本 |
| --- | --- | --- | --- | --- |
| **TensorFlow** | x86_64 | Ubuntu18.04 | tensorflow_2.1.0-cuda_10.1-py_3.7-ubuntu_18.04-x86_64 | cuda10.1 |
| **PyTorch** | x86_64 | Ubuntu18.04 | pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64 | cuda10.2 |
| **Ascend-Powered-Engine** | aarch64 | Euler2.8 | mindspore_1.7.0-cann_5.1.0-py_3.7-euler_2.8.3-aarch64 | cann 5.1.0 |
| | | | tensorflow_1.15-cann_5.1.0-py_3.7-euler_2.8.3-aarch64 | cann 5.1.0 |
| **MPI** | x86_64 | Ubuntu18.04 | mindspore_1.3.0-cuda_10.1-py_3.7-ubuntu_1804-x86_64 | cuda_10.1 |
| **Horovod** | x86_64 | ubuntu_18.04 | horovod_0.20.0-tensorflow_2.1.0-cuda_10.1-py_3.7-ubuntu_18.04-x86_64 | cuda_10.1 |
| | | | horovod_0.22.1-pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64 | cuda_10.2 |

**说明**

不同区域支持的AI引擎有差异，请以实际环境为准。

## 推理支持的 AI 引擎

在ModelArts创建AI应用时，若使用预置镜像“从模板中选择”或“从OBS中选择”导入模型，则支持如下常用引擎及版本的模型包。

**说明**

- 标注“推荐”的Runtime来源于统一镜像，后续统一镜像将作为主流的推理基础镜像。统一镜像中的安装包更齐全，详细信息可以参见**推理基础镜像列表**。
- 推荐将旧版镜像切换为统一镜像，旧版镜像后续将会逐渐下线。
- 待下线的基本镜像不再维护。
- 统一镜像Runtime的命名规范：<AI引擎名字及版本> - <硬件及版本：cpu或cuda或cann> - <python版本> - <操作系统版本> - <CPU架构>
- 当前支持自定义模型启动命令，预置AI引擎都有默认的启动命令，如非必要无需改动

**表 1-8** 支持的常用引擎及其 Runtime 以及默认启动命令

| 模型使用的引擎类型 | 支持的运行环境（Runtime） | 注意事项 |
| --- | --- | --- |
| TensorFlow | python3.6<br>python2.7（待下线）<br>tf1.13-python3.6-gpu<br>tf1.13-python3.6-cpu<br>tf1.13-python3.7-cpu<br>tf1.13-python3.7-gpu<br>tf2.1-python3.7（待下线）<br>tensorflow_2.1.0-cuda_10.1-py_3.7-ubuntu_18.04-x86_64（推荐） | • python2.7、python3.6的运行环境搭载的TensorFlow版本为1.8.0。<br>• python3.6、python2.7、tf2.1-python3.7，表示该模型可同时在CPU或GPU运行。其他Runtime的值，如果后缀带cpu或gpu，表示该模型仅支持在CPU或GPU中运行。<br>• 默认使用的Runtime为python2.7。<br>• 默认启动命令：sh /home/mind/run.sh |
| Spark_MLlib | python2.7（待下线）<br>python3.6（待下线） | • python2.7以及python3.6的运行环境搭载的Spark_MLlib版本为2.3.2。<br>• 默认使用的Runtime为python2.7。<br>• python2.7、python3.6只能用于运行适用于CPU的模型。<br>• 默认启动命令：bash /home/work/predict/bin/run.sh |
| Scikit_Learn | python2.7（待下线）<br>python3.6（待下线） | • python2.7以及python3.6的运行环境搭载的Scikit_Learn版本为0.18.1。<br>• 默认使用的Runtime为python2.7。<br>• python2.7、python3.6只能用于运行适用于CPU的模型。<br>• 默认启动命令：bash /home/work/predict/bin/run.sh |

| 模型使用的引擎类型 | 支持的运行环境（Runtime） | 注意事项 |
| --- | --- | --- |
| XGBoost | python2.7（待下线）<br>python3.6（待下线） | • python2.7以及python3.6的运行环境搭载的XGBoost版本为0.80。<br>• 默认使用的Runtime为python2.7。<br>• python2.7、python3.6只能用于运行适用于CPU的模型。<br>• 默认启动命令：bash /home/work/predict/bin/run.sh |
| PyTorch | python2.7（待下线）<br>python3.6<br>python3.7<br>pytorch1.4-python3.7<br>pytorch1.5-python3.7（待下线）<br>pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64（推荐） | • python2.7、python3.6、python3.7的运行环境搭载的PyTorch版本为1.0。<br>• python2.7、python3.6、python3.7、pytorch1.4-python3.7、pytorch1.5-python3.7，表示该模型可同时在CPU或GPU运行。<br>• 默认使用的Runtime为python2.7。<br>• 默认启动命令：sh /home/mind/run.sh |
| MindSpore | aarch64（推荐） | aarch64只能用于运行在Snt3芯片上。<br>• 默认启动命令：sh /home/mind/run.sh |

# 1.15 ModelArts 训练和推理分别对应哪些功能?

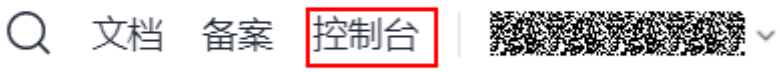
ModelArts训练包括自动学习、模型训练、专属资源池-训练/开发环境功能。

ModelArts推理包括AI应用管理、部署上线功能。

# 1.16 如何查看账号 ID 和 IAM 用户 ID

1. 使用IAM账号登录华为云。
2. 在页面右上方单击“控制台”，进入华为云管理控制台。

**图 1-8** 控制台入口

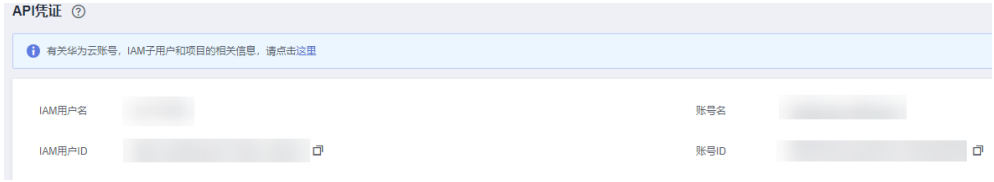



3. 在控制台右上角的账户名下方，单击“我的凭证”，进入“我的凭证”页面。

**图 1-9** 我的凭证


4. 在API凭证页面获取IAM用户名、用户ID、账号名和账号ID。

**图 1-10** 获取 IAM 用户名/用户 ID/账号名/账号 ID




# 1.17 ModelArts AI 识别可以单独针对一个标签识别吗?

标注多个标签进行训练而成的模型，最后部署成在线服务之后也是对标注的多个标签去进行识别的。如果只需要快速识别一种标签，建议单独训练识别此标签的模型使用，并选择较大的部署上线的规格也可以提供识别速度。

# 1.18 ModelArts 如何通过标签实现资源分组管理

ModelArts支持对接标签管理服务TMS，在ModelArts中创建资源消耗性任务（例如：创建Notebook、训练作业、推理在线服务）时，可以为这些任务配置标签，通过标签实现资源的多维分组管理。

ModelArts支持配置标签的任务有：创建训练作业任务、创建Notebook、创建推理在线服务。

## 使用流程

1. **Step1 在TMS上创建预定义标签**。
2. **Step2 在ModelArts任务中添加标签**。
3. **Step3 在TMS中根据资源类型查询ModelArts任务**。

## Step1 在 TMS 上创建预定义标签

登录TMS控制台，在预定义标签页面创建标签。此处创建的标签是全局标签，在华为云所有Region可见。

## Step2 在 ModelArts 任务中添加标签

在ModelArts中创建Notebook、创建训练作业、创建推理在线服务时，对这些任务配置标签。

- 在ModelArts的Notebook中添加标签。

  可以在创建Notebook页面添加标签，也可以在已经创建完成的Notebook详情页面的“标签”页签中添加标签。
- 在ModelArts的训练作业中添加标签。

  可以在创建训练作业页面添加标签，也可以在已经创建完成的训练作业详情页面的“标签”页签中添加标签。
- 在ModelArts的在线服务中添加标签。

  可以在创建在线服务页面添加标签，也可以在已经创建完成的在线服务详情页面的“标签”页签中添加标签。
- 在ModelArts的专属资源池中添加标签。

  可以在创建弹性集群的时候添加标签，也可以在已经创建完成的资源池详情页面的“标签”页签中添加标签。

**图 1-11** 添加标签


**说明**

用户也可以在ModelArts任务中添加标签时，创建新的标签，直接输入标签键和标签值即可。此处创建的标签仅当前的项目Project可见。不同的项目中查看不到。

## Step3 在 TMS 中根据资源类型查询 ModelArts 资源使用情况

登录TMS控制台，在资源标签页面根据资源类型和资源标签查询指定区域的资源任务。

- 区域：使用华为云的具体Region，区域概念请参见**什么是区域、可用区?**
- 资源类型：ModelArts支持查询的资源类型如**表1-9**所示。

- 资源标签：不填写标签时，表示查询所有资源，无论此资源是否有配置标签。选择相应标签查询资源，用户可以通过多个标签组合查询资源使用情况。

**表 1-9** ModelArts 的资源类型

| 资源类型 | 说明 |
| --- | --- |
| ModelArts-Notebook | ModelArts的开发环境Notebook对应的资源类型。 |
| ModelArts-TrainingJob | ModelArts的训练作业对应的资源类型。 |
| ModelArts-RealtimeService | ModelArts的推理在线服务对应的资源类型。 |
| ModelArts-ResourcePool | ModelArts的专属资源池对应的资源类型。 |

**说明**

如您的组织已经设定ModelArts的相关标签策略，则需按照标签策略规则为资源添加标签。标签如果不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。

# 1.19 为什么资源充足还是在排队？

- 如果是公共资源池，一般是由于其他用户占用资源导致，请耐心等待或根据**训练作业一直在等待中（排队）？**方法降低排队时间。
- 如果是专属资源池，建议您进行以下排查：
  a. 排查专属资源池中是否存在其他作业（包括推理作业、训练作业、开发环境作业等）。

     可通过总览页面，快速判断是否有其他模块的作业或实例在运行中，并进入到相关作业或实例上，判断是否使用了专属资源池。如判断相关作业或实例可停止，则可以停止，释放出更多的资源。

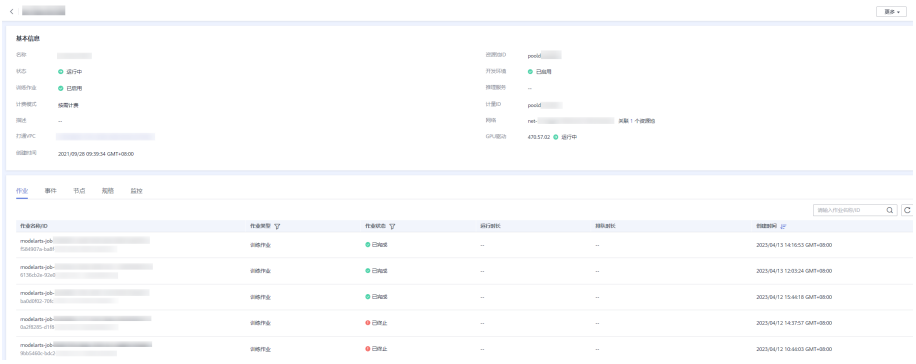
     **图 1-12** 总览

  b. 单击进入专属资源池详情页面，查看作业列表。

     观察队头是否有其他作业在排队，如果已有作业在排队，则新建的作业需要继续等待。

**图 1-13** 作业排队列表



c. 如果通过排查计算，发现资源确实足够，则考虑可能由于资源碎片化导致的。

例如，集群共2个节点，每个节点都空闲了4张卡，总剩余卡数为8张卡，但用户的作业要求为1节点8张卡，因此无法调度上。

# 1.20 规格中数字分别代表什么含义?

在创建作业时，若需选择资源规格，您可通过规格名称了解对应规格的相关信息，如加速卡显存、CPU核心数、内存、硬盘大小。

例如，“GPU: 1*GP-Vnt1(32GB) | CPU: 8 核 64GB 3200GB”中，32G为GPU显存、8核为CPU核心数、64GB为内存、3200GB为硬盘大小。

# 1.21 如何删除预置镜像中不需要的工具

预置的基础镜像中存在cpp、gcc等调试/编译工具，如果您不需要使用这些工具，可以通过运行脚本删除。

创建一个run.sh脚本文件，文件中的代码内容如下。然后在容器中执行sh run.sh命令运行脚本。

```
#!/bin/bash
delete_sniff_compiler()
{
    echo "[+][001] start remove debug tools"
    rm -rf /usr/bin/readelf
    rm -rf /usr/bin/gcc-nm
    #readelf根据需要决定是否删除
    #rm -rf /usr/local/Ascend/ascend-toolkit/latest/toolkit/toolchain/hcc/aarch64-target-linux-gnu/bin/readelf
    rm -rf /usr/bin/gcc
    rm -rf /usr/bin/cpp
    rm -rf /usr/bin/objdump
    echo "[+] complete"
}
hardening_ssh_config()
{
   sed -i "s/Subsystem/#Subsystem/g" /etc/ssh/sshd_config #关闭sftp服务
   sed -i "s/^.MaxAuthTries.*/MaxAuthTries 6/g" /etc/ssh/sshd_config #开启防暴力机制
   sed -i "s/^.ClientAliveInterval.*/ClientAliveInterval 300/g" /etc/ssh/sshd_config #开启会话超时机制
   systemctl restart /etc/ssh/sshd_config
   chmod 600 /home/ma-user/.ssh/id_rsa*  #收缩公私钥文件权限
   chmod 600 /home/ma-user/etc/ssh_host_rsa_key*  #收缩公私钥文件权限
   sed -i "s/ma-user/#ma-user/g" /etc/sudoers     #不允许ma-user用户免密执行所有命令
}
  delete_sniff_compiler
  hardening_ssh_config
```

Ascend镜像中存在hcc编译器，具体说明请参见昇腾社区提供的**HCC编译器说明文档**。

# 2 计费相关

## 2.1 如何查看 ModelArts 中正在收费的作业?

在ModelArts管理控制台，单击左侧菜单栏的“总览”，您可以在“总览”区域查看正在收费的作业。根据实际情况进入管理页面，停止实例。例如，Notebook正在计费，请前往“开发空间 > Notebook”页面，将状态为“运行中”的Notebook实例停止。

ModelArts使用过程中涉及到的具体收费项如下：

- Workflow：Workflow工作流运行时会收取费用，使用完请及时停止Workflow工作流、停止因运行Workflow工作流而创建的训练作业和部署的服务。同时，也需清理存储到OBS中的数据。
- 自动学习：自动学习运行时会收取费用，使用完请及时停止自动学习、停止因运行自动学习而创建的训练作业和部署的服务。同时，也需清理存储到OBS中的数据。
- Notebook实例：
  - 运行中的Notebook实例会收费，使用完成后请及时停止Notebook实例或删除。使用EVS做存储时，需同时清理存储到EVS中的数据。
  - CodeLab计费：在体验CodeLab时，切换为付费规格后会收费，使用完后请在JupyterLab界面及时停止Notebook实例。
- 训练作业：训练作业运行时会收取费用，使用完请及时停止训练作业。同时，也需清理存储到OBS中的数据。
- 部署上线：模型部署为在线服务、边缘服务时，会收取费用，使用完请及时停止服务。同时，也需清理存储到OBS中的数据。
- 专属资源池：在使用ModelArts进行AI全流程开发时，若购买了专属资源池，同时在运行自动学习作业、Workflow工作流、Notebook实例、模型训练和部署服务时选择使用已购买的专属资源池，则以上操作用到的计算资源会直接通过专属资源池来付费。按需计费的专属资源池，创建后会持续计费，不使用时请及时删除。

**⚠ 注意**

除了ModelArts总览页呈现的计费项之外，如果用户使用了OBS、云硬盘EVS存储，也会扣费。

- 请前往OBS控制台，及时清空OBS中的数据。
- 请在ModelArts控制台上，删除带有EVS存储的Notebook实例。前往EVS控制台，及时清空EVS中的数据。

# 2.2 如何查看 ModelArts 消费详情?

在“费用中心”，您可以根据需求按照账期、产品类型等查询ModelArts的消费详情。本章节以查询“账单详情”为例指导您查看计费情况，如需了解更多的账单情况，请参见**查看费用账单**。

查询方法：

1. 单击右上方的“费用中心 > 费用账单”进入费用中心详情页面，在左侧导航栏选择“账单管理 > 流水和明细账单”，在流水和明细账单页面，可切换“账单详情”和“明细账单”页签查看账单信息。
2. 在“流水账单”列表页，罗列该账号下各种产品类型，每个任务产生的费用详细。您可以单击“操作 > 详情”，查看使用量详情。可拖动详情下方的进度条，查看“使用量”、“应付金额”等信息。

**图 2-1** 流水账单


3. 在“明细账单”列表页，罗列了该账号下各种资源的计费模式、使用量和单价等信息。可以按账期、统计维度和统计周期筛选查看明细账单。

**图 2-2** 明细账单

# 2.3 ModelArts 上传数据集收费吗?

ModelArts中的数据集管理、标注等操作不收费，但是由于数据集存储在OBS中，因此会根据您使用的OBS桶进行收费。建议您前往OBS服务，了解**OBS计费详情**，创建相应的OBS桶用于存储ModelArts使用的数据。

# 2.4 ModelArts 标注完样本集后，如何保证退出后不再产生计费?

标注样本集本身不计费，数据集存储在OBS中，收取OBS的费用。建议您前往OBS控制台，删除存储的数据和OBS桶，即可停止收费。

# 2.5 ModelArts 自动学习所创建项目一直在扣费，如何停止计费?

- 对于使用**公共资源池**创建的自动学习作业：
  - 登录ModelArts控制台，在自动学习作业列表中，删除正在扣费的自动学习作业。在训练作业列表中，停止因运行自动学习作业而创建的训练作业。在在线服务列表中，停止因运行自动学习作业而创建的服务。操作完成后，ModelArts服务即停止计费。
  - 登录OBS控制台，进入自己创建的OBS桶中，删除存储在OBS中的数据。操作完成后，OBS服务即停止计费。
- 对于使用**专属资源池**创建的自动学习作业：
  - 登录ModelArts控制台，在自动学习作业列表中，删除正在扣费的自动学习作业。在训练作业列表中，停止因运行自动学习作业而创建的训练作业。在在线服务列表中，停止因运行自动学习作业而创建的服务。在资源池列表中，删除运行自动学习作业的专属资源池。操作完成后，ModelArts服务即停止计费。
  - 登录OBS控制台，进入自己创建的OBS桶中，删除存储在OBS中的数据。操作完成后，OBS服务即停止计费。

# 2.6 如果不再使用 ModelArts，如何停止收费?

在ModelArts中进行AI全流程开发时，主要包括存储费用、资源费用。如果不再使用ModelArts，需要停止/删除ModelArts中运行的服务；删除在OBS中存储的数据；删除在EVS中存储的数据。

## 清理存储数据

由于ModelArts的数据存储在OBS中，请前往OBS服务删除对应数据和目录，停止计费。

## 清理资源

请检查在ModelArts所创建运行中的作业，并停止或删除相关作业，即可停止计费。

**操作步骤：**

在ModelArts管理控制台，单击左侧菜单栏的“总览”，您可以在“总览”区域查看正在收费的作业。再根据实际情况进入管理页面，停止收费。

**图 2-3** 查看收费作业


- 进入“ModelArts>Workflow”页面，检查是否有“运行中”的Workflow列表。如果有，单击Workflow列表中“操作 > 删除”即可停止计费。
- 进入“ModelArts>自动学习”页面，检查是否有“运行中”的项目。如果有，单击项目列表中“操作 > 删除”即可停止计费。
- 进入“ModelArts>开发空间>Notebook”页面，检查是否有“运行中”的Notebook。如果有，单击Notebook列表右方操作下的“停止”即可停止Notebook计费。检查是否有带云硬盘EVS存储的Notebook。如果有，停止并删除该Notebook，即可停止EVS计费。
- 进入“ModelArts>模型训练>训练作业”页面，检查是否有“运行中”的训练作业。如果有，单击该作业列表右方操作下的“停止”即可停止计费。
- 进入“ModelArts>部署上线>在线服务”页面，检查是否有“运行中”的推理作业。如果有，单击该作业列表右方操作下的“停止”即可停止计费。
- 进入“ModelArts>部署上线>批量服务”页面，检查是否有“运行中”的推理作业。如果有，单击该作业列表右方操作下的“停止”即可停止计费。
- 进入“ModelArts>部署上线>边缘服务”页面，检查是否有“运行中”的推理作业。如果有，单击该作业列表右方操作下的“停止”即可停止计费。

# 2.7 训练作业如何收费?

- 如果您使用的是公共资源池，则根据您选择的规格、节点数、运行时长进行计费。计费规则为“规格单价×节点数×运行时长”（运行时长精确到秒）。
- 如果您使用的是专属资源池，则训练作业就不再进行单独计费。由专属资源池进行收费。

# 2.8 为什么项目删除完了，仍然还在计费?

如果ModelArts的自动学习项目、Notebook实例、训练作业或服务，都已经处于停止状态，即总览页面没看到收费项目，仍然发现账号还在计费。

有以下几种可能情况：

1. 因为您在使用ModelArts过程中，将数据上传至OBS进行存储，OBS会根据实际存储的数据进行计费。建议前往OBS管理控制台，清理您不再使用的数据、文件夹以及OBS桶，避免产生不必要的费用。

2. 您在创建Notebook时，选择了云硬盘EVS存储，该存储会单独收费，Notebook停止后，EVS还在计费，请及时删除该Notebook实例。
3. 您在体验CodeLab时，切换规格为付费的规格时会收费。请前往CodeLab界面单击右上角  停止Notebook实例。

# 2.9 欠费后，ModelArts 的资源是否会被删除?

欠费后，ModelArts的资源不会被立即删除。

欠费后，您可以在“费用中心”查看欠费详情。为了防止相关资源不会被停止服务或者逾期释放，您需要及时进行还款或充值。

## 查询欠费步骤

1. 登录管理控制台。
2. 单击页面右上角的“费用”进入“费用中心”页面。
3. 在“总览”页面可以查看到当前的欠费金额。
4. 如果存在欠费，请及时充值。更多关于欠费还款操作，请参见**如何进行欠费还款**。

# 2.10 部署后的 AI 应用是如何收费的?

ModelArts支持将AI应用按照业务需求部署为服务。训练类型不同，部署后的计费方式不同。

将AI应用部署为服务时，根据数据集大小评估模型的计算节点个数，根据实际编码情况选择计算模式。

具体计费方式请参见**ModelArts产品价格详情**。部署AI应用可选择按需计费，也可根据业务类型和需求**购买套餐包**。

为避免出现因购买套餐和使用套餐不一致产生多余计费的问题出现， 建议您注意核对在使用的套餐包资源规格是否和购买的套餐包资源规格一致。

# 2.11 Notebook 中的 EVS 存储可以使用套餐包吗?

无法使用套餐包。

# 3 自动学习（旧版）

## 3.1 功能咨询

### 3.1.1 什么是自动学习?

自动学习功能可以根据标注的数据自动设计模型、自动调参、自动训练、自动压缩和部署模型，不需要代码编写和模型开发经验。

自动学习功能主要面向无编码能力的用户，其可以通过页面的标注操作，一站式训练、部署，完成AI模型构建。

### 3.1.2 ModelArts 自动学习与 ModelArts PRO 的区别

ModelArts自动学习，提供了AI初学者，零编码、零AI基础情况下，可使用自动学习功能，开发用于图像分类、物体检测、预测分析、文本分类、声音分类等场景的模型。

而ModelArts PRO是一款为企业级AI应用打造的专业开发套件。用户可根据预置工作流生成指定场景模型，无需深究底层模型开发细节。ModelArts PRO底层依托ModelArts平台提供数据标注、模型训练、模型部署等能力。也可以理解过增强版的自动学习，提供行业AI定制化开发套件，沉淀行业知识，让开发者聚焦自身业务。

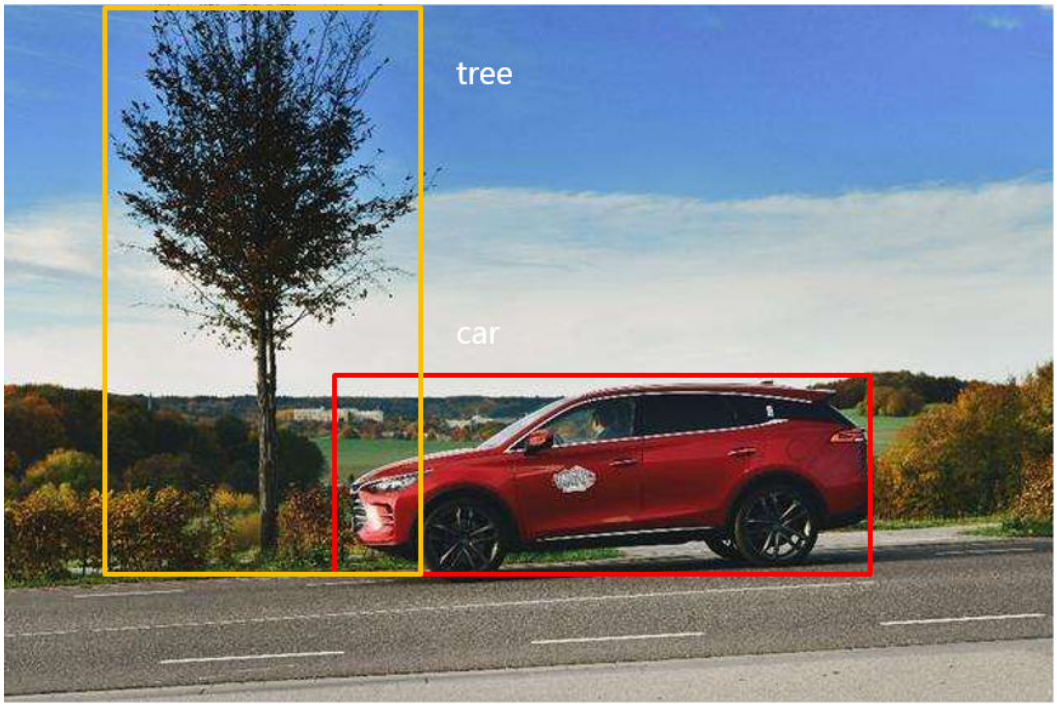
### 3.1.3 什么是图像分类和物体检测?

图像分类是根据各自在图像信息中所反映的不同特征，把不同类别的目标区分开来的图像处理方法。它利用计算机对图像进行定量分析，把图像或图像中的每个像元或区域划归为若干个类别中的某一种，以代替人的视觉判读。简单的说就是识别一张图中是否是某类/状态/场景，适合图中主体相对单一的场景，将下图识别为汽车的图片。

**图 3-1** 图像分类

物体检测是计算机视觉中的经典问题之一，其任务是用框去标出图像中物体的位置，并给出物体的类别。通常在一张图包含多个物体的情况下，定制识别出每个物体的位置、数量、名称，适合图片中有多个主体的场景，针对下图检测出图片包含树和汽车。

**图 3-2** 物体检测



## 3.1.4 自动学习和订阅算法有什么区别?

针对不同目标群体，ModelArts提供不同的AI开发方式。

- 如果您是新手，推荐您使用自动学习实现零代码模型开发。当您使用自动学习，系统会自动选择适合的算法和适合的参数进行模型训练。
- 如果您是AI开发进阶者，通过订阅算法进行模型训练有更多算法上的选择，并且您可以自定义训练所需的参数。

# 3.2 准备数据

## 3.2.1 自动学习的每个项目对数据有哪些要求?

### 图像分类对数据集的要求

- 文件名规范：不能有+、空格、制表符。
- 保证图片质量：不能有损坏的图片，目前支持的格式包括jpg、jpeg、bmp、png。
- 不要把明显不同的多个任务数据放在同一个数据集内。
- 每一类数据尽量多，尽量均衡。期望获得良好效果，图像分类项目中，至少有两种以上的分类，每种分类的样本不少于20张。
- 为了保证模型的预测准确度，训练样本跟真实使用场景尽量相似。
- 为保证模型的泛化能力，数据集尽量覆盖可能出现的各种场景。
- 在上传数据时，请选择非加密桶进行上传，否则会由于加密桶无法解密导致后期的训练失败。
- 用于训练的图片，至少有2种以上的分类，每种分类的图片数不少20张。

### 物体检测对数据集的要求

- 文件名规范，不能有中文，不能有+、空格、制表符。
- 保证图片质量：不能有损坏的图片；目前支持的格式包括jpg、jpeg、bmp、png。
- 不要把明显不同的多个任务数据放在同一个数据集内。
- 为了保证模型的预测准确度，训练样本跟真实使用场景尽量相似。
- 为保证模型的泛化能力，数据集尽量覆盖可能出现的各种场景。
- 物体检测数据集中，如果标注框坐标超过图片，将无法识别该图片为已标注图片。
- 在上传数据时，请选择非加密桶进行上传，否则会由于加密桶无法解密导致后期的训练失败。
- 用于训练的图片，至少有1种以上的分类，每种分类的图片数不少50张。

### 预测分析对数据集的要求

训练数据：

- 训练数据列数一致，总数据量不少于100条不同数据（有一个特征取值不同，即视为不同数据）。

- 训练数据列内容不能有时间戳格式（如：yy-mm-dd、yyyy-mm-dd等）的数据。
- 如果某一列的取值只有一种，会被视为无效列。请确保标签列的取值至少有两个且无数据缺失。

  **说明**

  标签列指的是在训练任务中被指定为训练目标的列，即最终通过该数据集训练得到模型时的输出（预测项）。
- 除标签列外数据集中至少还应包含两个有效特征列（列的取值至少有两个且数据缺失比例低于10%）。
- 当前由于特征筛选算法限制，预测数据列建议放在数据集最后一列，否则可能导致训练失败。

### 声音分类对数据集的要求

- 音频只支持16bit的WAV格式。支持WAV的所有子格式。
- 单条音频时长应大于1s，大小不能超过4MB。
- 适当增加训练数据，会提升模型的精度。声音分类建议每类音频至少20条，每类音频总时长至少5分钟。
- 建议训练数据和真实识别场景的声音保持一致并且每类的音频尽量覆盖真实环境的所有场景。
- 训练集的数据质量对于模型的精度有很大影响，建议训练集音频的采样率和采样精度保持一致。
- 标注质量对于最终的模型精度有极大的影响，标注过程中尽量不要出现误标情况。

### 文本分类对数据集的要求

- 文件格式要求为txt或者csv，文件大小不能超过8MB。
- 以换行符作为分隔符，每行数据代表一个标注对象。
- 文本分类目前只支持中文。

## 3.2.2 创建预测分析自动学习项目时，对训练数据有什么要求?

### 数据集要求

- 文件规范：名称由以字母数字及中划线下划线组成，以'.csv'结尾，且文件不能直接放在OBS桶的根目录下，应该存放在OBS桶的文件夹内。如：“/obs-xxx/data/input.csv”。
- 文件内容：文件保存为“csv”文件格式，文件内容以换行符（即字符“\n”，或称为LF）分隔各行，行内容以英文逗号（即字符“,”）分隔各列。文件内容不能包含中文字符，列内容不应包含英文逗号、换行符等特殊字符，不支持引号语法，建议尽量以字母及数字字符组成。
- 训练数据：
  - 训练数据列数一致，总数据量不少于100条不同数据（有一个特征取值不同，即视为不同数据）。
  - 训练数据列内容不能有时间戳格式（如：yy-mm-dd、yyyy-mm-dd等）的数据。

- 如果某一列的取值只有一种，会被视为无效列。请确保标签列的取值至少有两个且无数据缺失。

  **说明**

  标签列指的是在训练任务中被指定为训练目标的列，即最终通过该数据集训练得到模型时的输出（预测项）。
- 除标签列外数据集中至少还应包含两个有效特征列（列的取值至少有两个且数据缺失比例低于10%）。
- 训练数据的csv文件不能包含表头，否则会导致训练失败。

## 3.2.3 使用从 OBS 选择的数据创建表格数据集如何处理 Schema 信息?

Schema信息表示表格的列名和对应类型，需要跟导入数据的列数保持一致。

- 若您的原始表格中已包含表头，需要开启“导入是否包含表头”开关，系统会导入文件的第一行（表头）作为列名，无需再手动修改Schema信息。
- 若您的原始表格中没有表头，需要关闭“导入是否包含表头”开关，从OBS选择数据后，Schema信息的列名默认为表格中的第一行数据，请更改Schema信息中的“列名”为attr_1、attr_2、……、attr_n，其中attr_n为最后一列，代表预测列。

## 3.2.4 物体检测或图像分类项目支持对哪些格式的图片进行标注和训练?

图片格式支持JPG、JPEG、PNG、BMP。

# 3.3 创建项目

## 3.3.1 创建自动学习项目有个数限制吗?

ModelArts自动学习，包括图像分类项目、物体检测项目、预测分析项目、声音分类和文本分类项目。您最多只能创建100个自动学习项目。

## 3.3.2 创建项目的时候，数据集输入位置没有可选数据

### 可能原因

1. 创建的OBS桶与创建项目不在同一个区域。
2. 账号没有配置全局授权。
3. OBS桶里的数据格式不符合要求。

### 解决方法

查看ModelArts创建的项目与创建的OBS桶是否在同一区域。

1. 查看创建的OBS桶所在区域。
   a. 登录**OBS管理控制台**。

b. 进入“对象存储”界面，可在桶列表的“桶名称”列查找，或在右上方的搜索框中输入已经创建的桶名称搜索，找到您创建的OBS桶。
在“区域”列可查看创建的OBS桶的所在区域，如图1所示。

**图 3-3** OBS 桶所在区域

2. 查看ModelArts所在区域。
登录ModelArts控制台，在控制台左上角可查看ModelArts所在区域。
3. 比对您创建的OBS桶所在区域与ModelArts所在区域是否一致。务必保证OBS桶与ModelArts所在区域一致。

配置访问授权（全局配置）

1. 登录ModelArts管理控制台，在左侧导航栏选择“全局配置”，进入“全局配置”页面。
2. 单击“添加授权”，进入“访问授权”页面，根据参数说明进行配置。

**图 3-4** 查看权限列表

3. 然后勾选“我已经详细阅读并同意《ModelArts服务声明》”，单击“创建”，即完成委托配置。

# 3.4 数据标注

## 3.4.1 物体检测图片标注，一张图片是否可以添加多个标签?

可以，一张图片可添加多个标签。

## 3.4.2 在物体检测作业中上传已标注图片后，为什么部分图片显示未标注?

请您检查未标注图片的标注文件是否正确。如果标注框文件坐标超过图片，自动学习默认该图片未标注。

# 3.5 模型训练

## 3.5.1 创建图像分类自动学习项目并完成图片标注，训练按钮显示灰色，无法开始训练?

图像分类项目，图片标注至少需要两个类别，且每个类别至少5张图片，才可以开始自动训练。

## 3.5.2 自动学习项目中，如何进行增量训练?

在自动学习项目中，每训练一次，将自动产生一个训练版本。当前一次的训练结果不满意时（如对训练精度不满意），您可以适当增加高质量的数据，或者增减标签，然后再次进行训练。

**说明**

- 增量训练目前仅支持“图像分类”、“物体检测”、“声音分类”类型的自动学习项目。
- 为提升训练效果，建议在增量训练时，选择质量较高的数据，提升数据标注的质量。

### 增量训练的操作步骤

1. 登录ModelArts管理控制台，单击左侧导航栏的自动学习。
2. 在自动学习项目管理页面，单击对应的项目名称，进入此项目的自动学习详情页。
3. 在数据标注页面，单击未标注页签，在此页面中，您可以单击添加图片，或者增删标签。

   如果增加了图片，您需要对增加的图片进行重新标注。如果您增删标签，建议对所有的图片进行排查和重新标注。对已标注的数据， 也需要检查是否需要增加新的标签。
4. 在图片都标注完成后，单击右上角“开始训练”，在“训练设置”中，在“增量训练版本”中选择之前已完成的训练版本，在此版本基础上进行增量训练。其他参数请根据界面提示填写。

   设置完成后，单击“确定”，即进行增量训练。系统将自动跳转至“模型训练”页面，待训练完成后，您可以在此页面中查看训练详情，如“训练精度”、“评估结果”、“训练参数”等。

**图 3-5** 选择增量训练版本


## 3.5.3 自动学习训练后的模型是否可以下载?

不可以下载。但是您可以在AI应用管理页面查看，或者将此模型部署为在线服务。

## 3.5.4 自动学习为什么训练失败?

当自动学习项目训练失败时，请根据如下步骤排除问题。

1. 进入当前账号的费用中心，检查是否欠费。
   a. 是，建议您参考**华为云账户充值**，为您的账号充值。
   b. 否，执行**2**。
2. 检查存储图片数据的OBS路径。是否满足如下要求：
   - 此OBS目录下未存放其他文件夹。
   - 文件名称中无特殊字符，如~`@#$%^&*{}[]:;+=<>/

   如果OBS路径符合要求，请您按照服务具体情况执行**3**。
3. 自动学习项目不同导致的失败原因可能不同。
   - 图像识别训练失败请检查是否存在损坏图片，如有请进行替换或删除。
   - 物体检测训练失败请检查数据集标注的方式是否正确，目前自动学习仅支持矩形标注。
   - 预测分析训练失败请检查标签列的选取。标签列目前支持离散和连续型数据，只能选择一列。
   - 声音分类训练失败请检查音频格式是否为16bit的WAV格式。

   还无法排除故障，建议您提交**工单**，由专业工程师为您服务。

## 3.5.5 自动学习模型训练图片异常?

使用自动学习的图像分类或物体检测算法时，标注完成的数据在进行模型训练后，训练结果为图片异常。针对不同的异常情况说明及解决方案参见**表3-1**。

**表 3-1** 自动学习训练中图片异常情况说明（图像分类和物体检测）

| 序号 | 图片异常显示字段 | 图片异常说明 | 解决方案字段 | 解决方案说明 |
| --- | --- | --- | --- | --- |
| 1 | load failed | 图片无法被解码且不能修复 | ignore | 系统已自动过跳过这张图片，不需要用户处理。 |
| 2 | tf-decode failed | 图片无法被TensorFlow解码且不能修复 | ignore | 系统已跳过这张图片，不需要用户处理。 |
| 3 | size over | 图片大于5MB | resize to small | 系统已将图片压缩到5MB以内处理，不需要用户处理。 |
| 4 | mode illegal | 图片非RGB模式 | convert to rgb | 系统已将图片转成RGB格式处理，不需要用户处理。 |
| 5 | type illegal | 非图片文件，但可以转换成JPG | convert to jpg | 系统已将图片转换成JPG格式处理，不需要用户处理。 |

## 3.5.6 自动学习使用子账号单击开始训练出现错误 Modelarts.0010

用主账号给子账号配置ModelArts所使用的OBS桶的ACL权限即可。

## 3.5.7 自动学习中偏好设置的各参数训练速度大概是多少

偏好设置中：

performance_first：性能优先，训练时间较短，模型较小。对于TXT、图片类训练速度为10毫秒。

balance：平衡 。对于TXT、图片类训练速度为14毫秒 。

accuracy_first：精度优先，训练时间较长，模型较大。对于TXT、图片类训练速度为16毫秒。

## 3.5.8 自动学习声音分类预测报错 ERROR:input key sound is not in model

根据在线服务预测报错日志**ERROR：input key sound is not in model inputs**可知，预测的音频文件是空。预测的音频文件太小，换大的音频文件预测。

# 3.6 部署上线

## 3.6.1 自动学习中部署上线是将模型部署为什么类型的服务？

自动学习中部署上线是将模型部署为在线服务，您可以添加图片或代码进行服务测试，也可以使用URL接口调用。

部署成功后，您也可以在ModelArts管理控制台的“部署上线 > 在线服务”页面中，查看到正在运行的服务。您也可以在此页面停止服务或删除服务。

# 4 数据管理

## 4.1 添加图片时，图片大小有限制吗?

在数据管理功能中，针对“物体检测”或“图像分类”的数据集，在数据集中上传更多的图片时，是有限制的。要求单张图片大小不超过8MB，且只支持JPG、JPEG、PNG和BMP四种格式的图片。

请注意，针对自动学习功能中的添加图片，其图片大小限制不同，要求上传的图片大小不超过5MB。

**解决方案**:

- 方法1：使用导入功能。将图片上传至OBS任意目录，通过“从OBS目录导入”方式导入到已有数据集。
- 方法2：使用同步数据源功能。将图片上传到数据集输入目录下（或者其子目录），单击数据集详情页中的“同步数据源”将新增图片导入。需注意的是，同步数据源同时也会将OBS已删除的文件从数据集也删除，请谨慎操作。
- 方法3：新建数据集。将图片上传至OBS任意目录，可以直接使用这些图片目录作为数据集的输入目录，新建一个数据集。

## 4.2 数据集图片无法显示，如何解决?

### 问题现象

创建的数据集，在进行标注时无法显示图片，单击单张图片也无法查看。或者数据集中提示图片加载异常。

### 原因分析

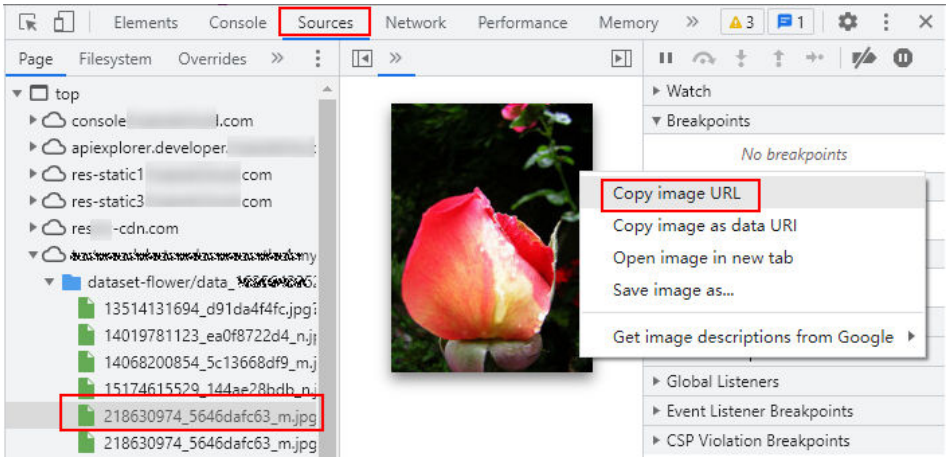
- 可能由于用户本地网络原因，无法正常访问OBS导致图片无法正常加载。
- 可能由于没有OBS桶的访问权限导致，请检查数据集输入位置所在的OBS桶，是否具有访问权限。
- 可能是OBS桶加密或者OBS文件加密导致。
- 可能跟OBS桶的存储类别有关，并行文件系统不支持图像处理，所以无法展示缩略图。

## 解决方案

1. 以Chrome浏览器为例，“F12”打开浏览器Console，锁定该图片，获取图片链接并复制。

**图 4-1** F12 获取图片链接




2. 在新的浏览器页面输入该链接，会出现提示“您的连接不是私密连接”，在该页面单击“高级”，然后选择继续前往目标链接页面。
3. 图片访问成功后再次返回ModelArts管理控制台访问数据集，即可成功查看图片。

# 4.3 如何将多个物体检测的数据集合并成一个数据集?

可以在OBS桶中创建一个父级目录，目录下面设置不同的文件夹，将多个数据集分别导出到这些文件夹里面，最后用父目录创数据集即可。

登录ModelArts管理控制台，选择“数据管理>数据集”进入数据集概览页，单击右上角“导出”，将对应的数据集到导出至OBS父级目录下的子文件夹中。

# 4.4 导入数据集失败

导入数据集失败可能原因为OBS桶类型选择错误，请您选择标准存储类型的桶导入。


# 4.5 表格类型的数据集如何标注

表格类型的数据集适合表格等结构化数据处理。数据格式支持csv。不支持标注，支持对部分表格数据进行预览，但是最多支持100条数据预览。

# 4.6 本地标注的数据，导入 ModelArts 需要做什么?

ModelArts支持通过导入数据集的操作，导入更多数据。本地标注的数据，当前支持从OBS目录导入或从Manifest文件导入两种方式。导入之后您还可以在ModelArts数据管理模块中对数据进行重新标注或修改标注情况。

从OBS目录导入或从Manifest详细操作指导和规范说明请参见**导入数据**。

# 4.7 为什么通过 Manifest 文件导入失败?

## 问题现象

针对已发布的数据集，使用此数据集的Manifest文件，重新导入，此时出现导入失败的错误。

## 原因分析

针对已发布的数据集，其对应的OBS目录下，发生了数据变化，如删除图片，导致此Manifest文件与当前OBS目录下的数据情况不符。使用此Manifest文件再次导入时，出现错误。

## 解决方案

- 方法1（推荐），建议将此数据集重新发布版本，然后再使用新版本的Manifest文件导入。

- 方法2，修改您本地的Manifest文件，查找OBS目录下的数据变更，根据变更同步修改Manifest。确保Manifest文件与OBS目录下的数据现状相同，然后使用修改后的Manifest文件导入。

# 4.8 标注结果存储在哪里?

ModelArts管理控制台，提供了数据可视化能力，您可以在控制台中查看详细数据以及标注信息。如需了解标注结果的存储路径，请参见如下说明。

## 背景说明

针对ModelArts中的数据集，在创建数据集时，需指定“数据集输入位置”和“数据集输出位置”。两个参数填写的均是OBS路径。

- “数据集输入位置”即原始数据存储的OBS路径。
- “数据集输出位置”，指在ModelArts完成数据标注后，执行数据集发布操作后，在此指定路径下，按数据集版本，生成相关目录。包含ModelArts中使用的Manifest文件（包含数据及标注信息）。详细文件说明可参见**数据集发布后，相关文件的目录结构说明**。

## 查看步骤

1. 在ModelArts管理控制台，进入“数据管理>数据集”。
2. 选择需查看数据集，单击名称左侧小三角，展开数据集详情。可获得“数据集输出位置”指定的OBS路径。

   **说明**

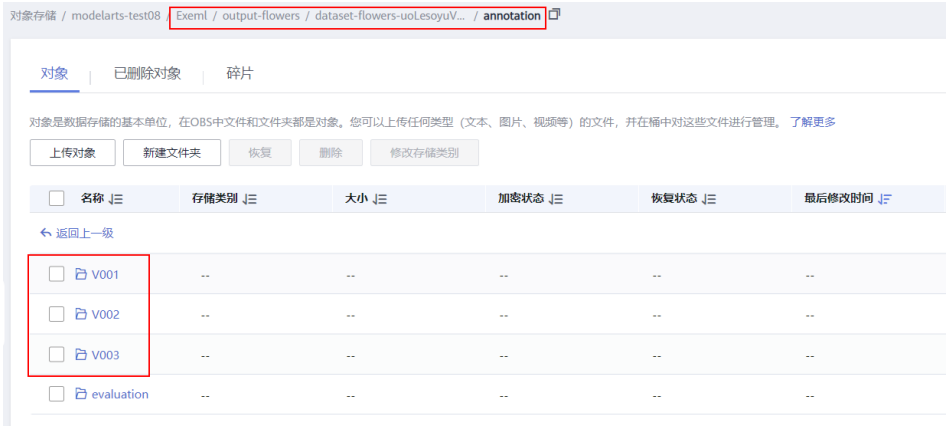
   获取标注信息前，需确保数据集已发布，至少有一个以上数据集版本。

   **图 4-2** 数据集详情

3. 进入OBS管理控制台，根据上述步骤获得的路径，找到对应版本号目录，即可获取数据集对应的标注结果。

**图 4-3** 获取标注结果




# 4.9 如何将标注结果下载至本地?

ModelArts数据集中的标注信息和数据在发布后，将以manifest格式存储在“数据集输出位置”对应的OBS路径下。

路径获取方式：

1. 在ModelArts管理控制台，进入“数据管理>数据集”。
2. 选择需查看数据集，单击名称左侧小三角，展开数据集详情。可获得“数据集输出位置”指定的OBS路径。
3. 进入OBS管理控制台，根据上述步骤获得的路径，找到对应版本号目录，即可获取数据集对应的标注结果。

如需将标注结果下载至本地，可前往manifest文件存储的OBS中，单击“下载”，即可将标注结果存储至本地。

**图 4-4** 下载标注结果


# 4.10 团队标注时，为什么团队成员收不到邮件?

团队标注时，成员收不到邮件的可能原因如下：

- 当数据集中的所有数据已完成标注，即“未标注”数据为空时，创建的团队标注任务，因为没有数据需要标注，不会给团队成员发送标注邮件。在发起团队标注任务时，请确保数据集中存在“未标注”数据。
- 只有当创建团队标注任务时，标注人员才会收到邮件。创建标注团队及添加标注团队的成员并不会发送邮件。
- 请确保您的邮箱已完成配置且配置无误。可参考**管理成员**，完成邮箱配置。
- 团队成员自检其邮箱是否有拦截设置。

## 4.11 可以两个账号同时进行一个数据集的标注吗?

可以多人同时标注，但多人同时对同一张图片标注的话，只会以最后一个保存的人的标注结果为最终标注结果。建议轮流标注并及时保存标注结果。

## 4.12 团队标注的数据分配机制是什么?

目前不支持用户自定义成员任务分配，数据是平均分配的。

- 当数量和团队成员人数不成比例，无法平均分配时，则将多余的几张图片，随机分配给团队成员。
- 如果样本数少于待分配成员时，部分成员会存在未分配到样本的情况。样本只会分配给labeler，比如10000张都是未标注，且5个都是labeler的话，那就是每个人分2000。

## 4.13 标注过程中，已经分配标注任务后，能否将一个 labeler 从标注任务中删除? 删除后对标注结果有什么影响? 如果不能删除 labeler，能否删除将他的标注结果从整体标注结果中分离出来?

目前不支持从标注任务中删除labeler。

labeler的标注必须通过审核后，才能同步到最终结果，不支持单独分离操作。

## 4.14 数据标注中，难例集如何定义? 什么情况下会被识别为难例?

难例是指难以识别的样本，目前只有图像分类和检测支持难例。

## 4.15 物体检测标注时，支持叠加框吗?

支持。

“物体检测”类型的数据集，在标注时，可在一张图片中添加多个标注框以及标签。需注意的是，标注框不能超过图片边缘。

## 4.16 如何将两个数据集合并?

目前不支持直接合并。

但是可以参考如下操作方式，将两个数据集的数据合并在一个数据集中。

例如需将数据集A和数据集B进行合并。

1. 分别将数据集A和数据集B进行发布。

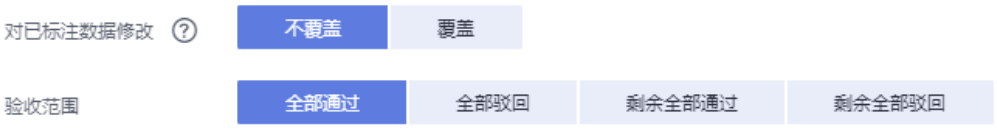
2. 发布后可获得数据集A和数据集B的Manifest文件。可通过数据集的“数据集输出位置”获得此文件。
3. 创建一个空数据集C，即无任何输出，其输入位置选择一个空的OBS文件夹。
4. 在数据集C中，执行导入数据操作，将数据集A和数据集B的Manifest文件导入。

   导入完成后，即将数据集A和数据集B的数据分别都合并至数据集C中。如需使用合并后的数据集，再针对数据集C执行发布操作即可。

# 4.17 智能标注是否支持多边形标注?

不支持。目前智能标注针对矩形框的标注类型，其他标注形式的样本，在智能标注的训练过程中，会跳过这部分。

# 4.18 团队标注的完成验收的各选项表示什么意思?




- 全部通过：被驳回的样本，也会通过。
- 全部驳回：已经通过的样本，需要重新标注，下次验收时重新进行审核。
- 剩余全部通过：已经驳回的会驳回，其余会自动验收通过。
- 剩余全部驳回：样本抽中的通过的，不需要标注了，未通过和样本未抽中的需要重新标注验收。

# 4.19 同一个账户，图片展示角度不同是为什么?

有的图片存在旋转角度等属性，不同的浏览器的处理策略不同，对浏览器的兼容性如表1和表2所示。

- L代表last，L3-产品版本上线时最新的3个稳定浏览器版本。
- 如果您当前使用的浏览器版本过低，将在一定程度上影响页面的显示效果，系统会提示您尽快对浏览器进行升级。
- 如果您当前使用的浏览器不支持访问管理控制台，系统会建议您对浏览器进行升级或安装支持的浏览器。

**表 4-1** PC 端浏览器兼容性一览表

| 浏览器类型 | 版本 | 操作系统 | 兼容性 |
| --- | --- | --- | --- |
| Internet Explorer | 11 | Windows 7 | 不承诺兼容。 |
| Microsoft Edge | L3 | Windows 10 | 完全兼容。 |
| | <79 | Windows 10 | 不承诺兼容。 |
| Mozilla Firefox | L3 | Windows 10 | 完全兼容。 |

| 浏览器类型 | 版本 | 操作系统 | 兼容性 |
| --- | --- | --- | --- |
| | L3 | CentOS 7+ | 部分兼容。<br>能确保基本交互操作，但在视觉、交互效果上可能存在兼容性问题。 |
| | L3 | Ubuntu 14.04 LTS+ | 部分兼容。<br>能确保基本交互操作，但在视觉、交互效果上可能存在兼容性问题。 |
| | L3 | macOS 10+ | 部分兼容。<br>能确保基本交互操作，但在视觉、交互效果上可能存在兼容性问题。 |
| Google Chrome | L3 | Windows 10 | 完全兼容。 |
| | L3 | CentOS 7+ | 部分兼容。<br>能确保基本交互操作，但在视觉、交互效果上可能存在兼容性问题。 |
| | L3 | Ubuntu 14.04 LTS+ | 部分兼容。<br>能确保基本交互操作，但在视觉、交互效果上可能存在兼容性问题。 |
| | L3 | macOS 10+ | 部分兼容。<br>能确保基本交互操作，但在视觉、交互效果上可能存在兼容性问题。 |
| Safari | L2 | macOS 10+ | 部分兼容。<br>能确保基本交互操作，但在视觉、交互效果上可能存在兼容性问题。 |

**表 4-2** 移动端浏览器兼容性一览表

| 浏览器类型 | 版本 | 操作系统 | 兼容性 |
| --- | --- | --- | --- |
| Chrome | L3 | Android | 完全兼容。 |
| Safari | L3 | IOS | 完全兼容。 |
| UC浏览器 | L3 | Android | 完全兼容。 |
| QQ浏览器 | L3 | Android | 完全兼容。 |
| 360浏览器 | L3 | Android | 完全兼容。 |

| 浏览器类型 | 版本 | 操作系统 | 兼容性 |
| --- | --- | --- | --- |
| 百度浏览器 | L3 | Android | 完全兼容。 |

# 4.20 智能标注完成后新加入数据是否需要重新训练?

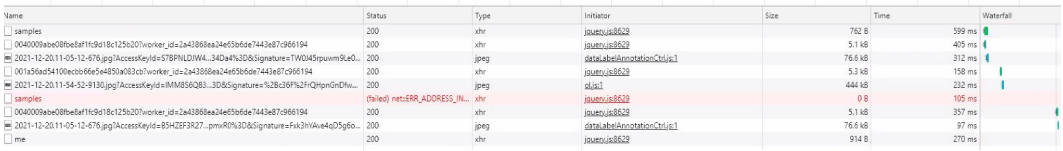
智能标注完成后，需要对标注结果进行确认。

- 如果未确认标注结果，直接加入新数据，重新智能标注，会将待确认的数据和新加入的数据全部重新训练。
- 如果确认标注结果后，再加入新数据，只重新训练标注新的数据。

# 4.21 为什么在 ModelArts 数据标注平台标注数据提示标注保存失败?

## 问题现象

以Chrome浏览器为例，同一张图片，第一次标注时，右上角弹窗提示标注保存失败，第二次提交相同的标注结果，又提示标注成功，此问题概率性发生。“F12”打开浏览器Console，单击network查看请求列表，请求状态显示为(failed)net::ERR_*ADDRESS*_IN_USE。




## 原因分析

可能是用户本地网络的原因，网速不稳定或者网络配置有问题，均可能导致保存失败。

## 解决方案

1. 切换为稳定的网络后重试。

2. 初始化网络配置，使用管理员权限启动CMD，输入netsh winsock reset指令，完成后重启电脑，再登录数据标注平台重试。

# 4.22 标注多个标签，是否可针对一个标签进行识别?

数据标注时若标注多个标签进行训练而成的模型，最后部署成在线服务之后也是对标注的多个标签去进行识别的。如果只需要快速识别一种标签，建议单独训练识别此标签的模型使用，并选择较大的部署上线的规格也可以提供识别速度。

# 4.23 使用数据处理的数据扩增功能后，新增图片没有自动标注

物体检测支持扩增后的图片自动标注，图像分类暂不支持。

# 4.24 视频数据集无法显示和播放视频

若无法显示和播放视频，请检查视频格式类型，目前只支持MP4格式。

# 4.25 使用样例的有标签的数据或者自己通过其他方式打好标签的数据放到 OBS 桶里，在 modelarts 中同步数据源以后看不到已标注，全部显示为未标注

OBS桶设置了自动加密会导致此问题，需要新建OBS桶重新上传数据，或者取消桶加密后，重新上传数据。

# 4.26 如何使用 soft NMS 方法降低目标框堆叠度

目前华为云AI市场订阅的算法中，yolo3可以使用该方法降低目标框堆叠度，yolo5 算法中没有看到相关支持的信息，需要在自定义算法进行使用。

# 4.27 ModelArts 标注数据丢失，看不到标注过的图片的标签

原因是删除了默认的标注作业，导致标签被删除。

# 4.28 如何将某些图片划分到验证集或者训练集?

目前只能指定切分比例，随机将样本划分到训练集或者验证集，不支持指定。

## 切分比例的指定：

在发布数据集时，仅“图像分类”、“物体检测”、“文本分类”和“声音分类”类型数据集支持进行数据切分功能。

一般默认不启用该功能。启用后，需设置对应的训练验证比例。

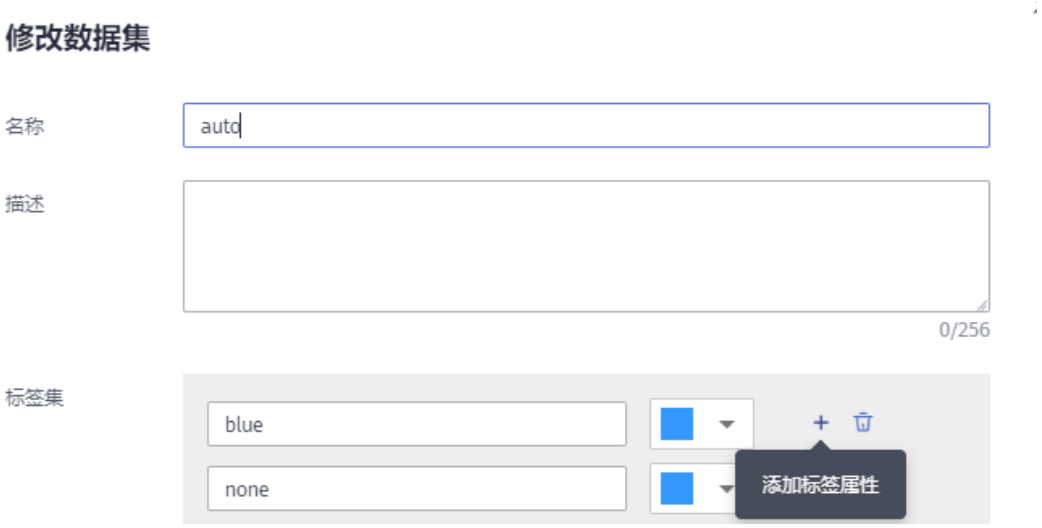
输入“训练集比例”，数值只能是0~1区间内的数。设置好“训练集比例”后，“验证集比例”自动填充。“训练集比例”加“验证集比例”等于1。

“训练集比例”即用于训练模型的样本数据比例；“验证集比例”即用于验证模型的样本数据比例。“训练验证比例”会影响训练模板的性能。

# 4.29 物体检测标注时除了位置、物体名字，是否可以设置其他标签，比如是否遮挡、亮度等?

可以通过修改数据集给标签添加自定义属性来设置一些自定义的属性。

**图 4-5** 修改数据集




# 4.30 ModelArts 数据管理支持哪些格式?

不同类型的数据集支持不同的功能。

| 数据集类型 | 标注类型 | 创建数据集 | 导入数据 | 导出数据 | 发布数据集 | 修改数据集 | 管理版本 | 自动分组 | 数据特征 |
|---|---|---|---|---|---|---|---|---|---|
| 文件型 | 图像分类 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 |
| | 物体检测 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 |
| | 图像分割 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 | - |
| | 声音分类 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |
| | 语音内容 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |
| | 语音分割 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |
| | 文本分类 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |

| 数据集类型 | 标注类型 | 创建数据集 | 导入数据 | 导出数据 | 发布数据集 | 修改数据集 | 管理版本 | 自动分组 | 数据特征 |
|---|---|---|---|---|---|---|---|---|---|
| | 命名实体 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |
| | 文本三元组 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |
| | 视频 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |
| | 自由格式 | 支持 | - | 支持 | 支持 | 支持 | 支持 | - | - |
| 表格型 | 表格 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |

## 4.31 旧版数据集中的数据是否会被清理?

旧版数据集中创建的数据不会被清理，旧版数据集中会自动关联一个数据标注任务。

但是在新版数据集中创建的数据，在旧版的数据集列表不会展示。

## 4.32 数据集版本管理找不到新建的版本

版本列表是可以缩放的，请缩小页面后查找。

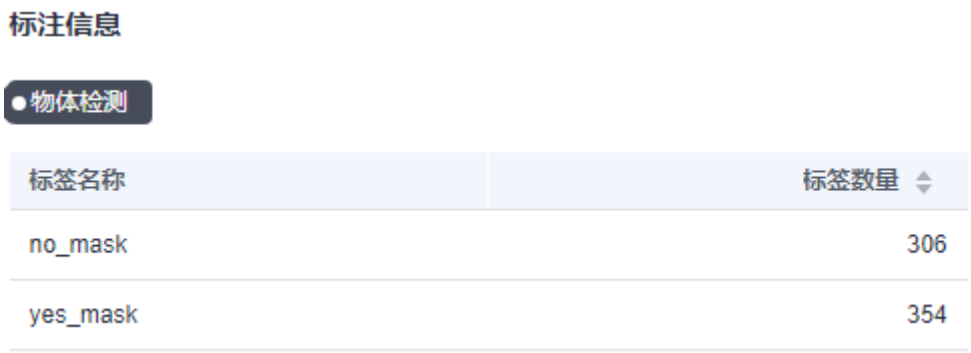
单击数据集名称，进入数据集概览页，在概览页选择“版本管理”，可对页面进行缩小。

# 4.33 如何查看数据集大小

数据管理目前只统计数据集的样本数量，无法查看数据集大小。

# 4.34 如何查看新版数据集的标注详情

1. 登录ModelArts管理控制台，左侧菜单栏选择“数据管理>数据集”。
2. 按照数据集名称，找到您想查看的数据集，单击该数据集名称，进入数据集概览页。
3. 在“概览”页签下，标注信息框，单击“查看标注详情”即可。




# 4.35 标注数据如何导出

只有“图像分类”、“物体检测”、“图像分割”类型的数据集支持导出功能。

- “图像分类”只支持导出txt格式的标注文件。
- “物体检测”只支持导出Pascal VOC格式的XML标注文件。
- “图像分割”只支持导出Pascal VOC格式的XML标注文件以及Mask图像。

其他类型的数据集可以使用**版本发布功能**。

# 4.36 找不到新创建的数据集

目前旧版数据集页面不展示新版数据集，新版数据集查看需跳转到新版的页面。


# 4.37 数据集配额不正确

当前每个账号支持的数据集配额为100，新版数据集页面显示所有已创建的数据集，但是旧版数据集页面不显示新版数据集。所以旧版页面存在显示不完整的情况，可以前往新版数据集页面查看。

# 4.38 数据集如何切分

在发布数据集时，仅“图像分类”、“物体检测”、“文本分类”和“声音分类”类型数据集支持进行数据切分功能。

一般默认不启用该功能。启用后，需设置对应的训练验证比例。

输入“训练集比例”，数值只能是0~1区间内的数。设置好“训练集比例”后，“验证集比例”自动填充。“训练集比例”加“验证集比例”等于1。

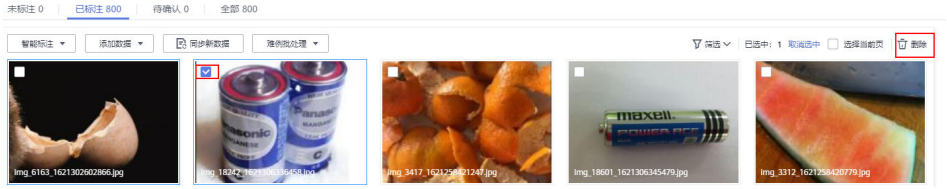
“训练集比例”即用于训练模型的样本数据比例；“验证集比例”即用于验证模型的样本数据比例。“训练验证比例”会影响训练模板的性能。

# 4.39 如何删除数据集图片

1. 登录ModelArts管理控制台，左侧菜单栏选择“数据管理>数据标注”，进入数据标注列表，单击需要删除图片的数据集，进入标注详情页。
2. 在“全部”、“未标注”或“已标注”页面中，依次选中需要删除的图片，或者“选择当前页”选中该页面所有图片，然后单击 删除。在弹出的对话框中，根据实际情况选择是否勾选“同时删除OBS源文件”，确认信息无误后，单击“确定”完成图片删除操作。

   其中，被选中的图片，其左上角将显示为勾选状态。如果当前页面无选中图片时， 按钮为灰色，无法执行删除操作。

**图 4-6** 删除数据集图片




# 4.40 从 AI Gallery 下载到桶里的数据集，再在 ModelArts 里创建数据集，显示样本数为 0

首先需要确认从AI Gallery下载的数据格式，比如压缩包、excel文件等会被忽略，支持格式详情：

| 数据集类型 | 标注类型 | 创建数据集 | 导入数据 | 导出数据 | 发布数据集 | 修改数据集 | 管理版本 | 自动分组 | 数据特征 |
|---|---|---|---|---|---|---|---|---|---|
| 文件型 | 图像分类 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 |
| | 物体检测 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 |
| | 图像分割 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 | 支持 | - |
| | 声音分类 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |
| | 语音内容 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |
| | 语音分割 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |
| | 文本分类 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |
| | 命名实体 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |
| | 文本三元组 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |
| | 视频 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |
| | 自由格式 | 支持 | - | 支持 | 支持 | 支持 | 支持 | - | - |
| 表格型 | 表格 | 支持 | 支持 | - | 支持 | 支持 | 支持 | - | - |

# 5 Notebook

## 5.1 规格限制

### 5.1.1 是否支持 sudo 提权?

出于安全考虑，Notebook不支持sudo提权操作。

### 5.1.2 是否支持 apt-get?

目前ModelArts开发环境的Terminal不支持使用“apt-get”。您可以使用**自定义镜像**来实现。

### 5.1.3 是否支持 Keras 引擎?

开发环境中的Notebook支持。训练作业和模型部署（即推理）暂时不支持。

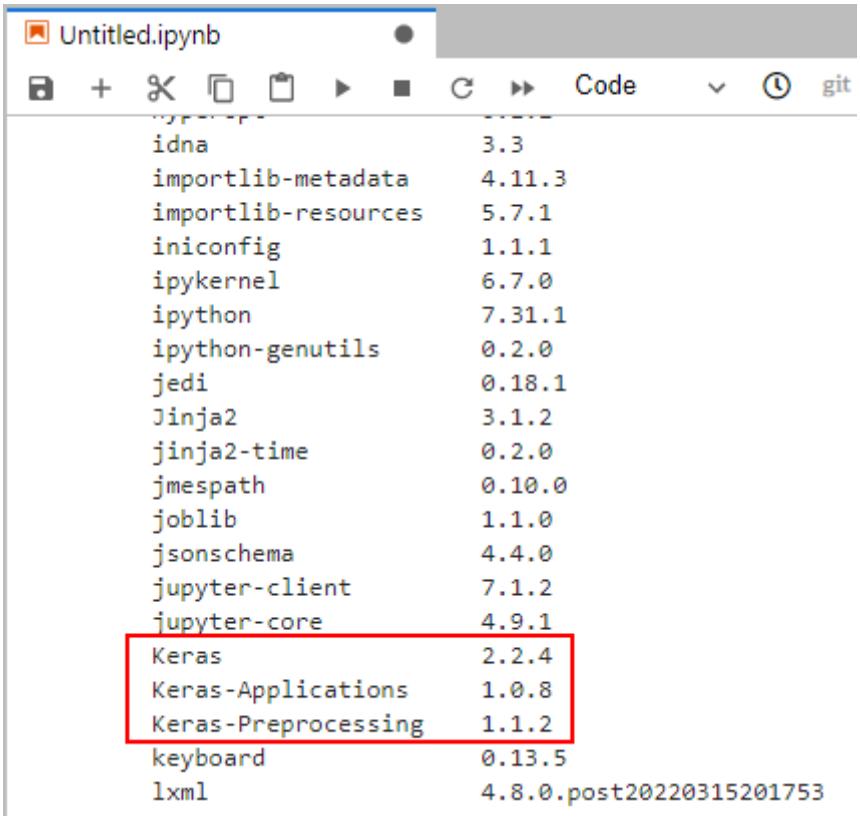
Keras是一个用Python编写的高级神经网络API，它能够以TensorFlow、CNTK或者Theano作为后端运行。Notebook开发环境支持“tf.keras”。

#### 如何查看 Keras 版本

1. 在ModelArts管理控制台，创建一个Notebook实例，镜像选择“TensorFlow-1.13”或“TensorFlow-1.15”。
2. 打开Notebook，在JupyterLab中执行**!pip list**查看Keras的版本。

图 5-1 查看 Keras 引擎版本




## 5.1.4 是否支持 caffe 引擎?

ModelArts的python2环境支持使用caffe，目前python3环境无法使用caffe。

## 5.1.5 是否支持本地安装 MoXing?

不支持，目前MoXing只支持在ModelArts里面使用。

## 5.1.6 Notebook 支持远程登录吗?

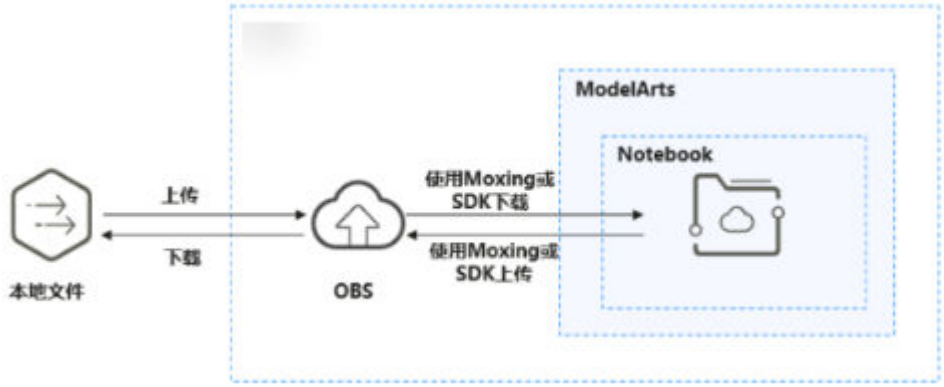
支持。创建Notebook时，可以开启SSH远程开发选项。在本地IDE通过**PyCharm专业版**或**VS Code**远程登录Notebook实例。

# 5.2 文件上传下载

## 5.2.1 如何在 Notebook 中上传下载 OBS 文件?

在Notebook中可以通过调用ModelArts的Moxing接口或者SDK接口与OBS交互，将Notebook中的文件上传至OBS，或者下载OBS中的文件至Notebook中。

**图 5-2** Notebook 中上传下载 OBS 文件




使用OBS客户端上传文件的操作指导：**上传文件**

## 方法一：在 Notebook 中通过 Moxing 上传下载 OBS 文件

MoXing是ModelArts自研的分布式训练加速框架，构建于开源的深度学习引擎TensorFlow、PyTorch等之上，使用MoXing API可让模型代码的编写更加简单、高效。

MoXing提供了一套文件对象API，可以用来读写OBS文件。

您可以通过MoXing API文档了解其与原生API对应关系，以及详细的接口调用示例，详细说明请参见**MoXing文件操作**。

示例代码：

```
import moxing as mox

#下载一个OBS文件夹sub_dir_0，从OBS下载至Notebook
mox.file.copy_parallel('obs://bucket_name/sub_dir_0', '/home/ma-user/work/sub_dir_0')
#下载一个OBS文件obs_file.txt，从OBS下载至Notebook
mox.file.copy('obs://bucket_name/obs_file.txt', '/home/ma-user/work/obs_file.txt')

#上传一个OBS文件夹sub_dir_0，从Notebook上传至OBS
mox.file.copy_parallel('/home/ma-user/work/sub_dir_0', 'obs://bucket_name/sub_dir_0')
#上传一个OBS文件obs_file.txt，从Notebook上传至OBS
mox.file.copy('/home/ma-user/work/obs_file.txt', 'obs://bucket_name/obs_file.txt')
```

## 方法二：在 Notebook 中通过 SDK 上传下载 OBS 文件

使用ModelArts SDK接口将OBS中的文件下载到Notebook后进行操作。

示例代码：将OBS中的文件file1.txt下载到Notebook的/home/ma-user/work/路径下。其中，桶名称、文件夹和文件的名称均可以按照业务需求自定义。

```
from modelarts.session import Session
session = Session()
session.obs.download_file(src_obs_file="obs://bucket-name/dir1/file1.txt", dst_local_dir="/home/ma-user/work/")
```

使用ModelArts SDK接口将OBS中的文件夹下载到Notebook后进行操作。

示例代码：将OBS中的文件夹dir1下载到Notebook的/home/ma-user/work/路径下。其中，桶名称和文件夹的名称均可以按照业务需求自定义。

```
from modelarts.session import Session
session = Session()
session.obs.download_dir(src_obs_dir="obs://bucket-name/dir1/", dst_local_dir="/home/ma-user/work/")
```

使用ModelArts SDK接口将Notebook中的文件上传到OBS后进行操作。

示例代码：将Notebook中的file1.txt文件上传到OBS桶路径obs://bucket-name/dir1/中。其中，桶名称、文件夹和文件的名称均可以按照业务需求自定义。

```
from modelarts.session import Session
session = Session()
session.obs.upload_file(src_local_file='/home/ma-user/work/file1.txt', dst_obs_dir='obs://bucket-name/dir1/')
```

使用ModelArts SDK接口将Notebook中的文件夹上传到OBS。

示例代码：将Notebook中的文件夹“/work/”上传至“bucket-name”桶的“dir1”文件夹下，路径为“obs://bucket-name/dir1/work/”。其中，桶名称和文件夹的名称均可以按照业务需求自定义。

```
from modelarts.session import Session
session = Session()
session.obs.upload_dir(src_local_dir='/home/ma-user/work/', dst_obs_dir='obs://bucket-name/dir1/')
```

### 异常处理

通过OBS下载文件到Notebook中时，提示Permission denied。请依次排查：

- 请确保读取的OBS桶和Notebook处于同一站点区域，例如：都在华北-北京四站点。不支持跨站点访问OBS桶。具体请参见**查看OBS桶与ModelArts是否在同一个区域**。
- 请确认操作Notebook的账号有权限读取OBS桶中的数据。如没有权限，请参见**在Notebook中，如何访问其他账号的OBS桶？**。

## 5.2.2 如何上传本地文件至 Notebook?

Notebook中JupyterLab的文件上传方式请参见**上传本地文件至JupyterLab**。

## 5.2.3 如何导入大文件到 Notebook 中?

- **大文件（大于100MB的文件）**

  针对大文件，建议使用OBS服务上传文件。使用OBS客户端，将本地文件上传至OBS桶中，然后使用ModelArts SDK从OBS下载文件至Notebook本地。

  使用OBS客户端上传文件的操作指导：**上传文件**。

  使用ModelArts SDK或Moxing接口从OBS下载文件请参见**如何在Notebook中上传下载OBS文件？**。

- **文件夹**

  将文件夹压缩成压缩包，上传方式与大文件相同。将文件上传至Notebook后，可在Terminal中解压压缩包。

  ```
  unzip xxx.zip  #在xxx.zip压缩包所在路径直接解压
  ```

  解压命令的更多使用说明可以在主流搜索引擎中查找Linux解压命令操作。

## 5.2.4 upload 后，数据将上传到哪里?

针对这个问题，有两种情况：

- 如果您创建的Notebook使用OBS存储实例时

  单击“upload”后，数据将直接上传到该Notebook实例对应的OBS路径下，即创建Notebook时指定的OBS路径。

- 如果您创建的Notebook使用EVS存储实例时

  单击“upload”后，数据将直接上传至当前实例容器中，即在“Terminal”中的“~/work”目录下。

## 5.2.5 如何下载 Notebook 中的文件到本地?

Notebook中JupyterLab下载文件到本地的方式，请参见**从JupyterLab下载文件至本地**。

## 5.2.6 如何将开发环境 Notebook A 的数据复制到 Notebook B 中?

目前不支持直接将Notebook A的数据复制到Notebook B，如果需要复制数据，可参考如下步骤操作：

1. 将Notebook A的数据上传至OBS；
2. 下载OBS中的数据至Notebook B。

文件的上传下载详细操作请参考**如何在Notebook中上传下载OBS文件?** 。

## 5.2.7 在 Notebook 中上传文件失败，如何解决?

### 问题现象

- 文件上传很快，但是上传失败。
- 上传文件到Notebook时，界面一直在转圈；使用Moxing命令上传，报错；上传OBS文件时，打开OBS浏览器也不显示桶，一直在“获取数据中”。


- 在JupyterLab界面通过 ModelArts Upload Files按钮上传文件时，显示“获取数据失败”。

**图 5-3** OBS 文件上传界面


查看Notebook日志（通常在/home/ma-user/log/下，notebook-<date>.log），报错“List objects failed, obs_client resp: {'status' : 403, 'reason' : 'Forbidden' , 'errorCode' : 'AccessDenied'”。

## 可能原因

第一种问题现象是通过华为内网上传时，文件大小受限，需要解决内网的问题。

其他问题现象的可能原因如下：

- 无OBS访问授权。
- 无OBS桶或文件的访问权限。
- OBS桶被删除。

## 解决方案

- **检查委托授权**

  请前往全局配置，查看是否具有OBS访问授权。如果没有，请参考**配置访问授权（全局配置）**。
- **请确认是否有OBS桶的访问权限**

  进入OBS控制台页面，可以看到所有的OBS桶列表，进入需要访问的桶，确认是否有权限访问，如果无权限则会报错。
- **进入OBS控制台页面，确认OBS桶是否存在。**

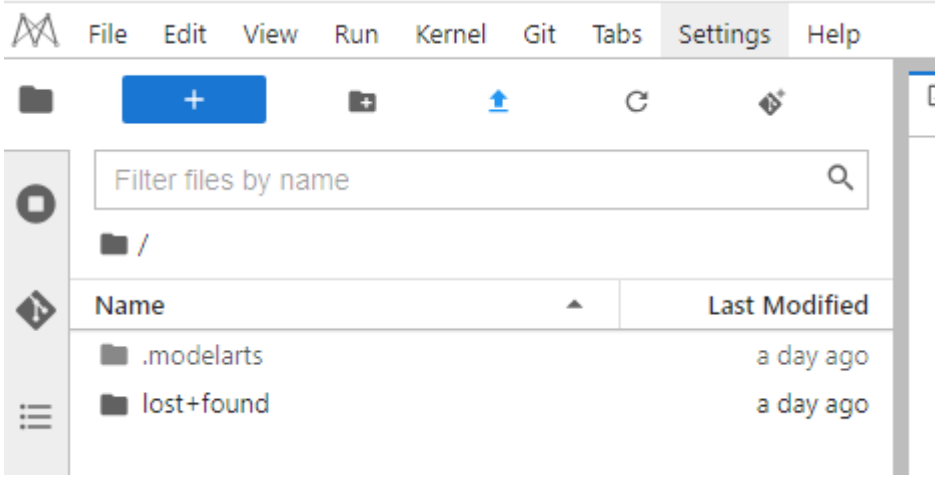
# 5.2.8 动态挂载 OBS 并行文件系统成功，但是在 Notebook 的 JupyterLab 中无法看到本地挂载点

## 问题现象

在Notebook中动态挂载OBS并行文件系统，本地挂载目录为/data/demo-yf/，实际在JupyterLab左侧导航看不到此目录。

**图 5-4** 本地挂载目录

**图 5-5** Notebook 的 JupyterLab

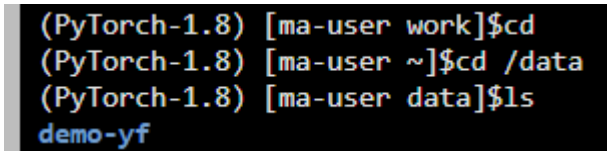



### 原因分析

本地挂载目录是在Notebook容器的“~/data”目录下创建的demo-yf文件夹，而JupyterLab左侧导航默认路径为“~/work”目录，相当于/data和/work是同一层级，所以在JupyterLab中看不到。

打开Terminal后，默认为~work目录，执行如下命令进入~data目录查看本地挂载路径：

```
(PyTorch-1.8) [ma-user work]$cd
(PyTorch-1.8) [ma-user ~]$cd /data
(PyTorch-1.8) [ma-user data]$ls
```




# 5.3 数据存储

## 5.3.1 如何对 OBS 的文件重命名?

由于OBS管理控制台不支持对OBS的文件重命名，当您需要对OBS文件进行重命名时需要通过调用MoXing API实现，在已有的或者新创建的Notebook中，执行如下命令，通过接口对OBS中的文件进行重命名。

具体操作如下：

如下示例为将文件“obs_file.txt”重命名为“obs_file_2.txt”。

```
import moxing as mox
mox.file.rename('obs://bucket_name/obs_file.txt', 'obs://bucket_name/obs_file_2.txt')
```

## 5.3.2 Notebook 停止或者重启后，“/cache”下的文件还存在么？如何避免重启？

“/cache”目录下存储的是临时文件，在Notebook实例停止或重启后，不会被保存。存储在“/home/ma-user/work”目录下的数据，在Notebook实例停止或重启后，会被保留。

为避免重启，请勿在开发环境中进行重型作业训练，如大量占用资源的作业。

## 5.3.3 如何使用 pandas 库处理 OBS 桶中的数据？

**步骤1** 参考**下载OBS文件到Notebook中**的指导，将OBS中的数据下载至Notebook本地处理。

**步骤2** 参考**pandas用户指南**处理pandas数据。

**----结束**

## 5.3.4 在 Notebook 中，如何访问其他账号的 OBS 桶？

创建Notebook时选择OBS存储，这种情况下只能访问到自己账号下的桶，无法访问到其他账号的OBS桶。

如果需要在Notebook中，访问其他账号的OBS文件，前提是，需获取目标OBS桶的读写权限。

1. 首先，请联系OBS桶的创建者，参考**对其他账号授予桶的读写权限**指导，授予当前账号OBS桶的读写权限。此操作指导是某一华为云账号将其OBS桶权限授予其他华为云账号。如果您的账号是IAM用户或其他场景时，请参见**《 OBS权限配置指南 》**> 典型场景配置案例，查找授予OBS桶权限的指导。
2. 获得OBS桶的读写权限后，您可以在Notebook中，使用moxing接口，访问对应的OBS桶，并读取数据。举例如下：
   ```
   import moxing as mox
   mox.file.copy_parallel('obs://bucket_1/dataset', 'obs://bucket_2/dataset')
   ```
   其中，“bucket_1”为其他账号的OBS桶，“bucket_2”为自己的OBS桶。

## 5.3.5 JupyterLab 默认工作路径是什么？

- **带OBS存储的Notebook实例**

  JupyterLab文件默认存储路径，为创建Notebook时指定的OBS路径。

  在文件列表的所有文件读写操作都是基于所选择的OBS路径下的内容操作的，跟当前实例空间没有关系。如果用户需要将内容同步到实例空间，需使用**JupyterLab上传下载功能**。

- **带EVS存储的Notebook实例**

  JupyterLab文件默认存储路径，为创建Notebook实例时，系统自动分配的EVS空间。

  在文件列表的所有文件读写操作都是基于所选择的EVS下的内容操作的。使用EVS类型的挂载，可将大数据挂载至“~/work”目录下。

# 5.4 环境配置相关

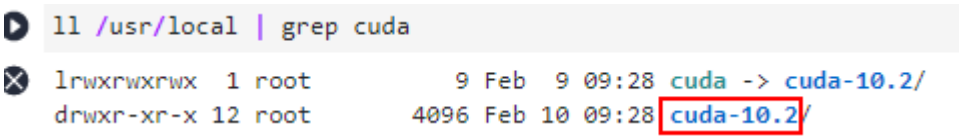
## 5.4.1 如何查看 Notebook 使用的 cuda 版本?

执行如下命令查看环境中的cuda版本。

```
ll /usr/local | grep cuda
```

举例：

**图 5-6** 查看当前环境的 cuda 版本

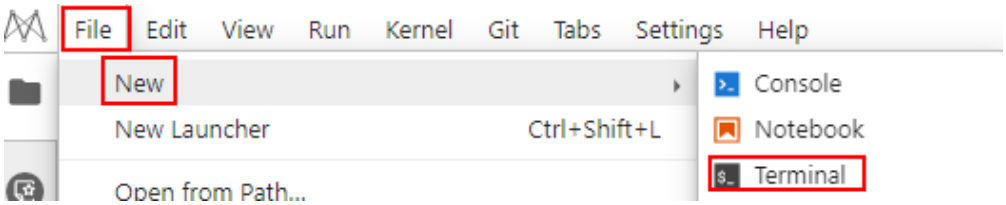



如图1所示，当前环境中cuda版本为10.2

## 5.4.2 如何打开 ModelArts 开发环境的 Terminal 功能?

1. 登录ModelArts管理控制台，选择“开发空间>Notebook”。
2. 创建Notebook实例，实例处于“运行中”，单击“操作”列的“打开”，进入“JupyterLab”开发页面。
3. 选择“Files > New > Terminal”，进入到Terminal界面。

**图 5-7** 进入 Terminal 界面




## 5.4.3 如何在 Notebook 中安装外部库?

ModelArts Notebook中已安装Jupyter、Python程序包等多种环境，包括TensorFlow、MindSpore、PyTorch、Spark等。您也可以使用pip install在Notobook或Terminal中安装外部库。

### 在 Notebook 中安装

例如，通过JupyterLab在“TensorFlow-1.8”的环境中安装Shapely。

1. 打开一个Notebook实例，进入到Launcher界面。
2. 在“Notebook”区域下，选择“TensorFlow-1.8”，新建一个ipynb文件。
3. 在新建的Notobook中，在代码输入栏输入如下命令。

   **!pip install Shapely**

### 在 Terminal 中安装

例如，通过terminal在“TensorFlow-1.8”的环境中使用**pip**安装Shapely。

1. 打开一个Notebook实例，进入到Launcher界面。

2. 在“Other”区域下，选择“Terminal”，新建一个terminal文件。
3. 在代码输入栏输入以下命令，获取当前环境的kernel，并激活需要安装依赖的python环境。

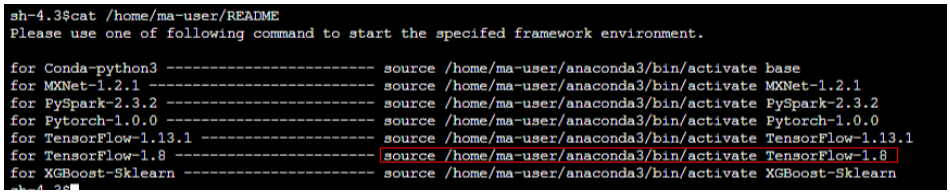
   **cat /home/ma-user/README**

   **source /home/ma-user/anaconda3/bin/activate TensorFlow-1.8**

   **说明**

   如果需要在其他python环境里安装，请将命令中“TensorFlow-1.8”替换为其他引擎。

**图 5-8** 激活环境




4. 在代码输入栏输入以下命令安装Shapely。

   **pip install Shapely**

## 5.4.4 如何获取本机外网 IP?

本机的外网IP地址可以在主流搜索引擎中搜索“IP地址查询”获取。

**图 5-9** 查询外网 IP 地址


## 5.4.5 如何解决“在 IOS 系统里打开 ModelArts 的 Notebook，字体显示异常”的问题?

### 问题现象

在IOS系统里打开ModelArts的Notebook时，字体显示异常。

### 解决方法

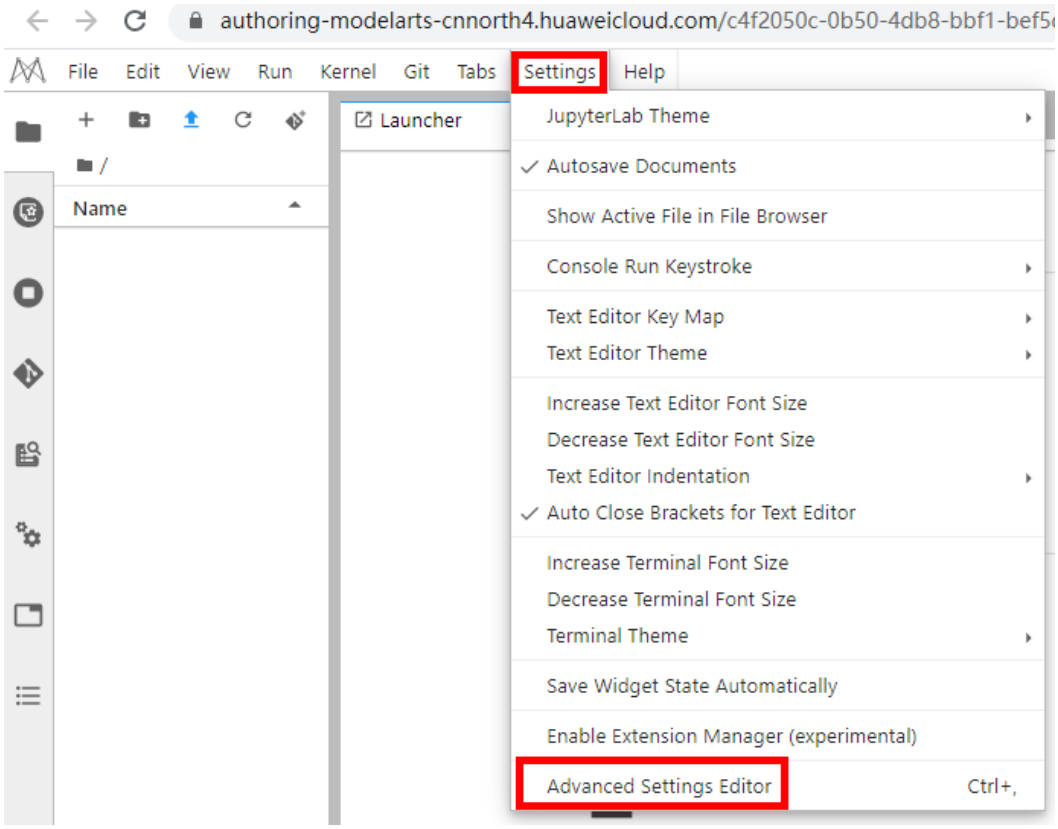
设置Terminal的“fontFamily”为“Menlo”。

## 操作步骤

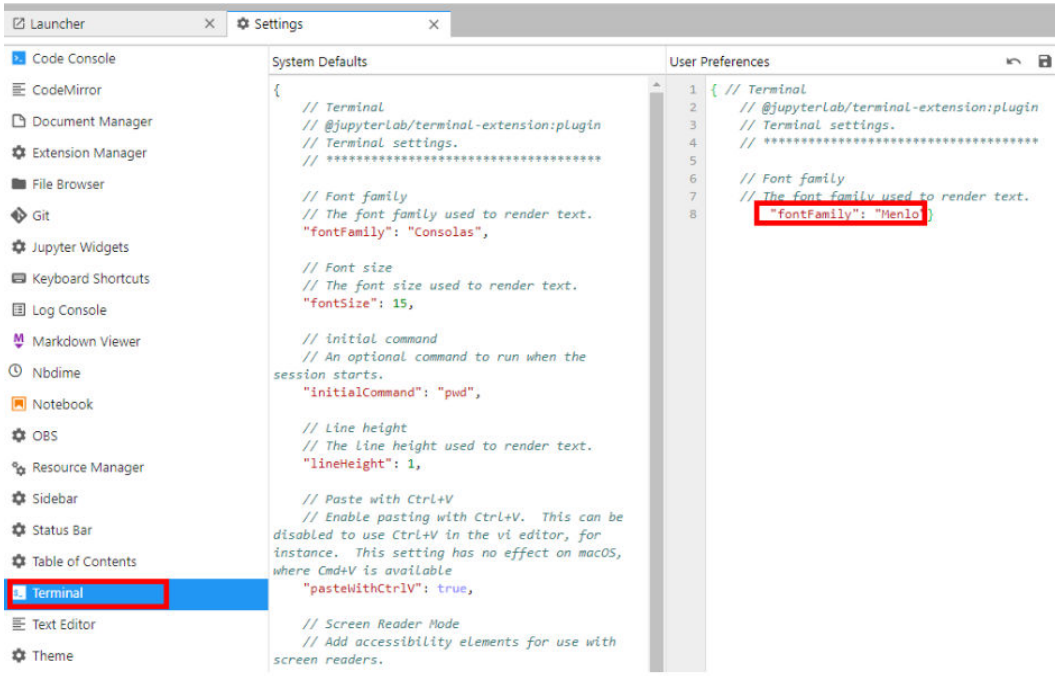
**步骤1** 登录ModelArts管理控制台，选择“开发空间>Notebook”。

**步骤2** 创建Notebook实例之后，在Notebook列表中，单击目标Notebook“操作”列的“打开”，进入“JupyterLab”开发页面。

**步骤3** 在“JupyterLab”页面，选择菜单栏的“Settings>Advanced Settings Editor”，进入“Settings”页面。




**步骤4** 单击Terminal，将“fontFamily”设置为“Menlo”。




**----结束**

## 5.4.6 Notebook 有代理吗？如何关闭？

Notebook有代理。

执行**env|grep proxy**命令查询Notebook代理。

执行**unset https_proxy unset http_proxy**命令关闭代理。

## 5.4.7 在 Notebook 中添加自定义 IPython Kernel

### 使用场景

当前Notebook默认内置的引擎环境不能满足用户诉求，用户可以新建一个conda env按需搭建自己的环境。本小节以搭建一个“python3.6.5和tensorflow1.2.0”的IPython Kernel为例进行展示。

### 操作步骤

1. 创建conda env。

   在Notebook的Terminal中执行如下命令。其中，my-env是虚拟环境名称，用户可自定义。conda详细参数可参考**conda官网**。

   ```
   conda create --quiet --yes -n my-env python=3.6.5
   ```

   创建完成后，执行**conda info --envs**命令查看现有的虚拟环境列表，可以看到my-env虚拟环境：

   ```
   sh-4.4$conda info --envs
   # conda environments:
   #
   base                  *  /home/ma-user/anaconda3
   TensorFlow-2.1           /home/ma-user/anaconda3/envs/TensorFlow-2.1
   my-env                   /home/ma-user/anaconda3/envs/my-env
   python-3.7.10            /home/ma-user/anaconda3/envs/python-3.7.10                /opt/conda/envs/my-env
   ```

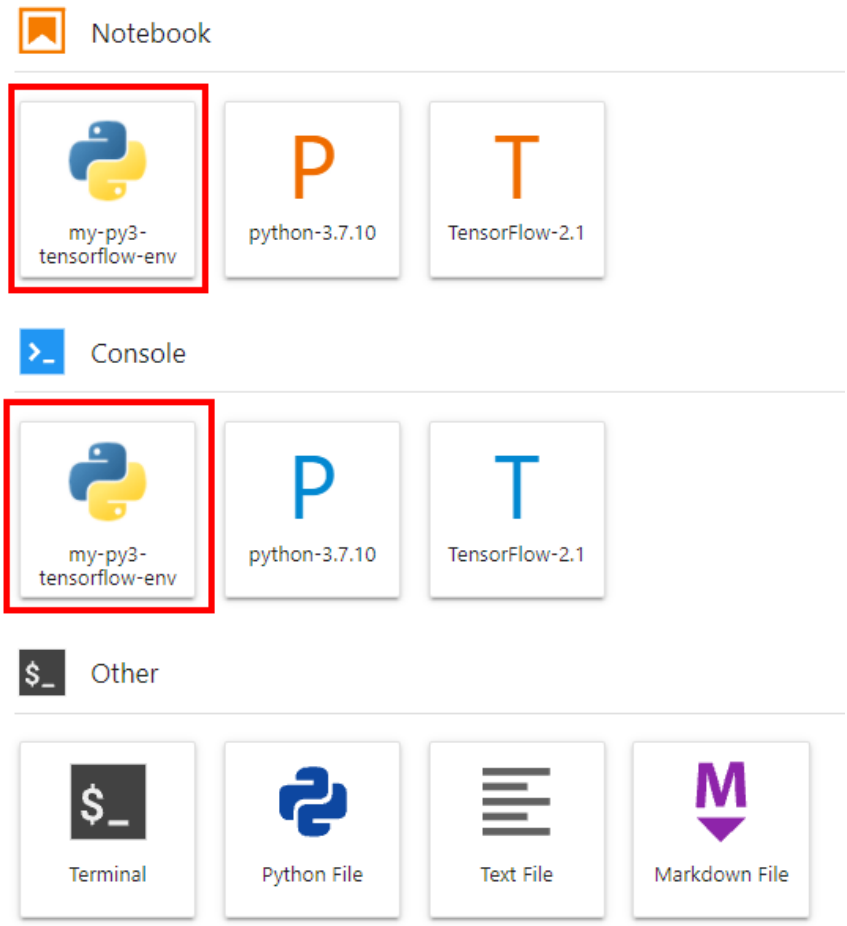
2. 执行如下命令进入conda env。
```
source /home/ma-user/anaconda3/bin/activate /home/ma-user/anaconda3/envs/my-env
```
3. 执行如下命令在my env里安装如下依赖包。
```
pip install jupyter
pip install jupyter_core==5.3.0
pip install jupyter_client==8.2.0
pip install ipython==8.10.0
pip install ipykernel==6.23.1
```
4. 执行下述命令添加虚拟环境为IPython Kernel。

   其中--name的值可自定义。
```
python3 -m ipykernel install --user --name "my-py3-tensorflow-env"
```
   执行完毕后，可以看到下述提示信息。
```
(my-env) sh-4.4$python3 -m ipykernel install --user --name "my-py3-tensorflow-env"
Installed kernelspec my-py3-tensorflow-env in /home/ma-user/.local/share/jupyter/kernels/my-py3-tensorflow-env
```
5. 自定义虚拟环境Kernel的环境变量。

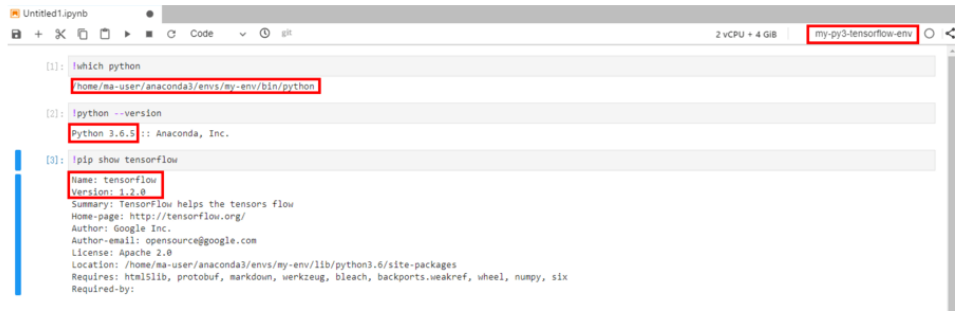
   执行**cat /home/ma-user/.local/share/jupyter/kernels/my-py3-tensorflow-env/kernel.json**，可以看到默认配置如下：
```
{
 "argv": [
  "/home/ma-user/anaconda3/envs/my-env/bin/python3",
  "-m",
  "ipykernel_launcher",
  "-f",
  "{connection_file}"
 ],
 "display_name": "my-py3-tensorflow-env",
 "language": "python"
}
```
   按需添加env字段的值，可参考下述配置。其中，PATH中增加了该虚拟环境python包所在路径：
```
{
 "argv": [
  "/home/ma-user/anaconda3/envs/my-env/bin/python3",
  "-m",
  "ipykernel_launcher",
  "-f",
  "{connection_file}"
 ],
 "display_name": "my-py3-tensorflow-env",
 "language": "python",
 "env": {
     "PATH": "/home/ma-user/anaconda3/envs/my-env/bin:/opt/conda/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/home/ma-user/modelarts/ma-cli/bin",
     "http_proxy": "http://proxy-notebook.modelarts-dev-proxy.com:8083",
     "https_proxy": "http://proxy-notebook.modelarts-dev-proxy.com:8083",
     "ftp_proxy": "http://proxy-notebook.modelarts-dev-proxy.com:8083",
     "HTTP_PROXY": "http://proxy-notebook.modelarts-dev-proxy.com:8083",
     "HTTPS_PROXY": "http://proxy-notebook.modelarts-dev-proxy.com:8083",
     "FTP_PROXY": "http://proxy-notebook.modelarts-dev-proxy.com:8083"
 }
}
```
6. 进入虚拟环境的IPython Kernel。

   刷新JupyterLab页面，可以看到自定义的虚拟环境Kernel。如下所示：




单击my-py3-tensorflow-env图标，验证是否为当前环境，如下所示：




7. 清理环境。

   删除虚拟环境的IPython Kernel。

   ```
   jupyter kernelspec uninstall my-py3-tensorflow-env
   ```

   删除虚拟环境。

   ```
   conda env remove -n my-env
   ```

## 5.5 Notebook 实例常见错误

# 5.5.1 创建 Notebook 实例后无法打开页面，如何处理?

如果您在创建Notebook实例之后，打开Notebook时，因报错导致无法打开页面，您可以根据以下对应的错误码来排查解决。

## 打开 Notebook 显示黑屏

Notebook打开后黑屏，由于代理问题导致，切换代理。

## 打开 Notebook 显示空白

- 打开Notebook时显示空白，请清理浏览器缓存后尝试重新打开。
- 检查浏览器是否安装了过滤广告组件，如果是，请关闭该组件。

## 报错 404

如果是IAM用户在创建实例时出现此错误，表示此IAM用户不具备对应存储位置（OBS桶）的操作权限。

解决方法：

1. 使用账号登录OBS，并将对应OBS桶的访问权限授予该IAM用户。详细操作指导请参见：**被授权用户**。
2. IAM用户获得权限后，登录ModelArts管理控制台，删除该实例，然后重新使用此OBS路径创建Notebook实例。

## 报错 503

如果出现503错误，可能是由于该实例运行代码时比较耗费资源。建议先停止当前Notebook实例，然后重新启动。

## 报错 504

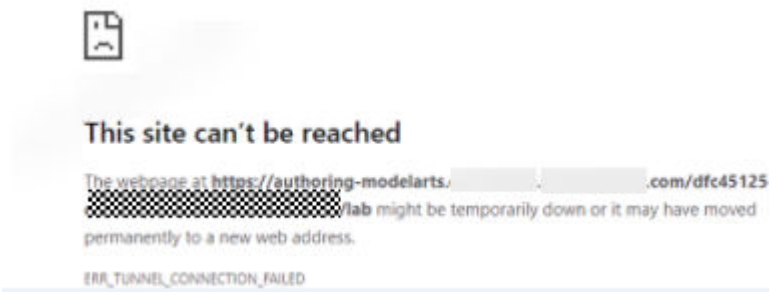
如果报此错误时，请提工单或拨打热线电话协助解决。提工单和热线电话请参见：**https://www.huaweicloud.com/service/contact.html**。

## 报错 500

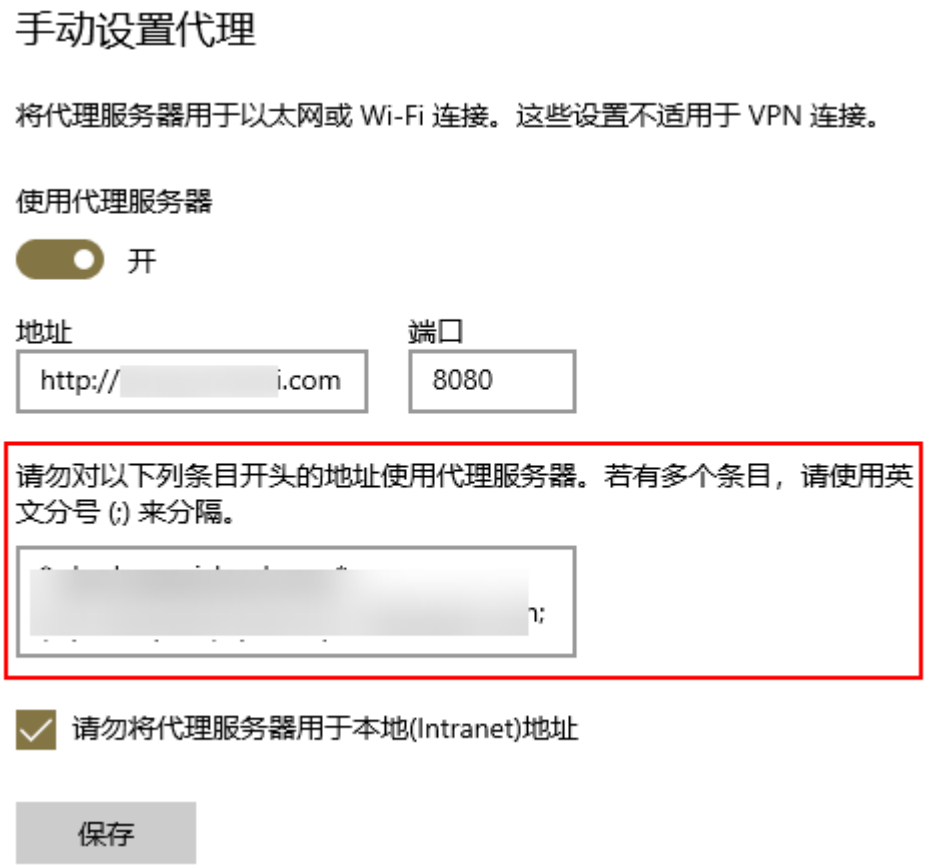
Notebook JupyterLab页面无法打开，报错500，可能是工作目录work下的磁盘空间满了，请参考**Notebook提示磁盘空间已满**排查并清理磁盘空间。

## 报错 This site can't be reached

创建完Notebook后，单击操作列的“打开”，报错如下：



解决方案：复制页面的域名，添加到windows代理“请勿对以下列条目开头的地址使用代理服务器”中，然后保存就可以正常打开。




## 5.5.2 使用 pip install 时出现“没有空间”的错误

### 问题现象

在Notebook实例中，使用**pip install**时，出现“No Space left...”的错误。

### 解决办法

建议使用**pip install --no-cache **** 命令安装，而不是使用**pip install ****。加上“--no-cache”参数，可以解决很多此类报错。

## 5.5.3 使用 pip install 提示 Read timed out

### 问题现象

在Notebook实例中，使用**pip install**时，提示“ReadTimeoutError...”或者“Read timed out...”的错误。

```
sh-4.3$pip install torch==1.7.0 torchvision==0.8.0 torchaudio==0.7.0 matplotlib pyyaml tqdm sklearn h5py tensorboard pandas
Looking in indexes: http://pip-notebook.modelarts.com:8888/repository/pypi/simple/
Collecting torch==1.7.0
  WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ReadTimeou
tError("HTTPConnectionPool(host='pip-notebook.modelarts.com', port=8888): Read timed out. (read timeout=15)")': /repository/pypi
/packages/torch/1.7.0/torch-1.7.0-cp37-cp37m-manylinux1_x86_64.whl
  WARNING: Retrying (Retry(total=3, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ReadTimeou
tError("HTTPConnectionPool(host='pip-notebook.modelarts.com', port=8888): Read timed out. (read timeout=15)")': /repository/pypi
/packages/torch/1.7.0/torch-1.7.0-cp37-cp37m-manylinux1_x86_64.whl
  WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ReadTimeou
tError("HTTPConnectionPool(host='pip-notebook.modelarts.com', port=8888): Read timed out. (read timeout=15)")': /repository/pypi
/packages/torch/1.7.0/torch-1.7.0-cp37-cp37m-manylinux1_x86_64.whl
^CERROR: Operation cancelled by user
WARNING: You are using pip version 21.0.1; however, version 21.1.2 is available.
You should consider upgrading via the '/opt/conda/bin/python -m pip install --upgrade pip' command.
```

### 解决办法

建议先尝试使用**pip install --upgrade pip**，再使用**pip install**。

## 5.5.4 出现“save error”错误，可以运行代码，但是无法保存

如果当前Notebook还可以运行代码，但是无法保存，保存时会提示“save error”错误。大多数原因是华为云WAF安全拦截导致的。

当前页面，即用户的输入或者代码运行的输出有一些字符被华为云拦截，认为有安全风险。出现此问题时，请提交工单，联系专业的工程师帮您核对并处理问题。

## 5.5.5 单击 Notebook 的打开按钮时报“请求超时”错误?

当Notebook容器因内存溢出等原因导致崩溃时，如果此时单击Notebook的打开按钮时，将会出现“请求超时”错误。

该种情况下，请耐心等待容器恢复，约几十秒，再重新单击打开按钮即可。

## 5.5.6 使用 CodeLab 时报错 kernel restart

报错是由于CPU满了，建议切换更高规格或使用付费规格的CPU。

**图 5-10** 切换规格或使用付费规格的 CPU

## 5.5.7 使用 SSH 工具连接 Notebook，服务器的进程被清理了，GPU 使用率显示还是 100%

原因是代码运行卡死导致被进程清理，GPU显存没有释放；或者代码运行过程中内存溢出导致程序被清理，需要释放下显存，清理GPU，然后重新启动。为了避免进程结束引起的代码未保存，建议您每隔一段时间保存下代码输出至OBS桶或者容器./work目录下。

## 5.5.8 Notebook 实例出现“Server Connection Error”错误

在Terminal中执行命令时，出现错误如**图1 报错信息截图**所示，此问题可能由于CPU/GPU或显存等占满，可在JupyterLab界面下方查看内存使用情况，如**图5-12**所示。

此时Kernel会自动重启，存储在“/home/ma-user/work”目录下的数据会被保留，其他目录的数据均不会保留。

**图 5-11** 报错信息截图

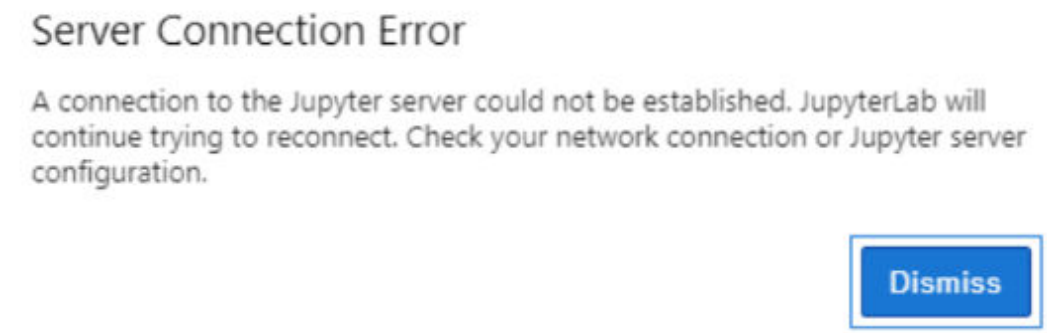
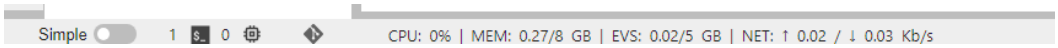



**图 5-12** 查看内存使用情况



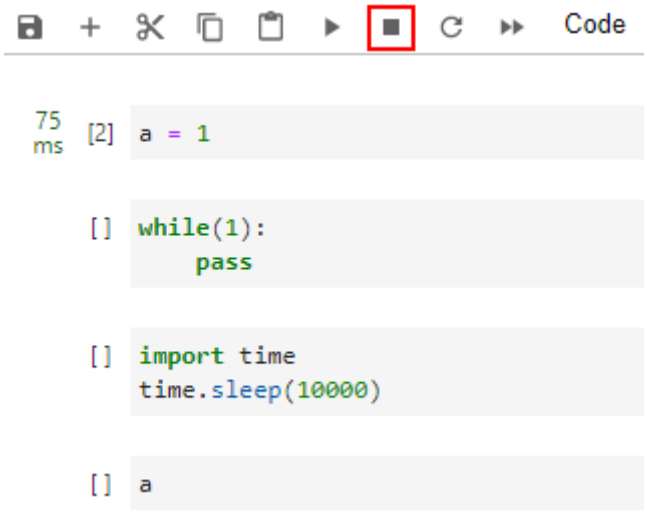

# 5.6 代码运行常见错误

## 5.6.1 Notebook 无法执行代码，如何处理?

当Notebook出现无法执行时，您可以根据如下几种情况判断并处理。

1. 如果只是Cell的执行过程卡死或执行时间过长，如**图5-13**中的第2个和第3个Cell，导致第4个Cell无法执行，但整个Notebook页面还有反应，其他Cell也还可以单击，则直接单击下图中红色方框处的“interrupt the kernel”，停止所有Cell的执行，同时会保留当前Notebook中的所有变量空间。

**图 5-13** 停止所有 Cell




2. 如果整个Notebook页面也已经无法使用，单击任何地方都无反应，则关闭Notebook页面，关闭ModelArts管理控制台页面。然后，重新打开管理控制台，打开之前无法使用的Notebook，此时的Notebook仍会保留无法使用之前的所有变量空间。
3. 如果重新打开的Notebook仍然无法使用，则进入ModelArts管理控制台页面的Notebook列表页面，“停止”此无法使用的Notebook。待Notebook处于“停止”状态后，再单击“启动”，重新启动此Notebook，并打开Notebook。此时，Notebook仍会保留无法使用之前的所有变量空间。

## 5.6.2 运行训练代码，出现 dead kernel，并导致实例崩溃

在Notebook实例中运行训练代码，如果数据量太大或者训练层数太多，亦或者其他原因，导致出现“内存不够”问题，最终导致该容器实例崩溃。

出现此问题后，系统将自动重启Notebook，来修复实例崩溃的问题。此时只是解决了崩溃问题，如果重新运行训练代码仍将失败。如果您需要解决“内存不够”的问题，建议您创建一个新的Notebook，使用更高规格的资源池，比如专属资源池来运行此训练代码。已经创建成功的Notebook不支持选用更高规格的资源规格进行扩容。

## 5.6.3 如何解决训练过程中出现的 cudaCheckError 错误?

### 问题现象

Notebook中，运行训练代码出现如下错误。

```
cudaCheckError() failed : no kernel image is available for execution on the device
```

### 原因分析

因为编译的时候需要设置setup.py中编译的参数arch和code和电脑的显卡匹配。

### 解决方法

对于GP Vnt1的显卡，GPU算力为**-gencode arch=compute_70,code=[sm_70,compute_70]**，设置setup.py中的编译参数即可解决。

## 5.6.4 开发环境提示空间不足，如何解决?

当提示空间不足时，推荐使用EVS类型的Notebook实例。

参考**如何在Notebook中上传下载OBS文件?** 操作指导，针对原有的Notebook，首先将代码和数据上传至OBS桶中。然后创建一个EVS类型的Notebook，将此OBS中的文件下载至Notebook本地（指新建的EVS类型Notebook）。

## 5.6.5 如何处理使用 opencv.imshow 造成的内核崩溃?

### 问题现象

当在Notebook中使用opencv.imshow后，会造成Notebook崩溃。

### 原因分析

opencv的cv2.imshow在jupyter这样的client/server环境下存在问题。 而matplotlib不存在这个问题。

### 解决方法

参考如下示例进行图片显示。注意opencv加载的是BGR格式， 而matplotlib显示的是RGB格式。

Python语言：

```
from matplotlib import pyplot as plt
import cv2
img = cv2.imread('图片路径')
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title('my picture')
plt.show()
```

## 5.6.6 使用 Windows 下生成的文本文件时报错找不到路径?

### 问题现象

当在Notebook中使用Windows下生成的文本文件时，文本内容无法正确读取，可能报错找不到路径。

### 原因分析

Notebook是Linux环境，和Windows环境下的换行格式不同，Windows下是CRLF，而Linux下是LF。

### 解决方法

可以在Notebook中转换文件格式为Linux格式。

shell语言：

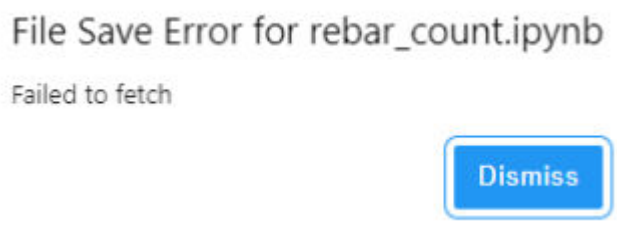
```
dos2unix  文件名
```

## 5.6.7 JupyterLab 中文件保存失败，如何解决?

### 问题现象

JupyterLab中保存文件时报错如下：




### 原因分析

- 浏览器安装了第三方插件proxy进行了拦截，导致无法进行保存。
- 在Notebook中的运行文件超过指定大小就会提示此报错。
- jupyter页面打开时间太长。
- 网络环境原因，是否有连接网络代理。

### 解决方法

- 关掉插件然后重新保存。
- 减少文件大小。
- 重新打开jupyter页面。
- 请检查网络。

# 5.7 CodeLab

## 5.7.1 如何将 git clone 的 py 文件变为 ipynb 文件

在ipynb文件中，执行**%load XXX.py**命令，即可将py文件内容加载到ipynb中。以“test.py”文件为例，下图展示了如何将“test.py”的文件内容加载到ipynb文件中。

**图 5-14** test.py 文件

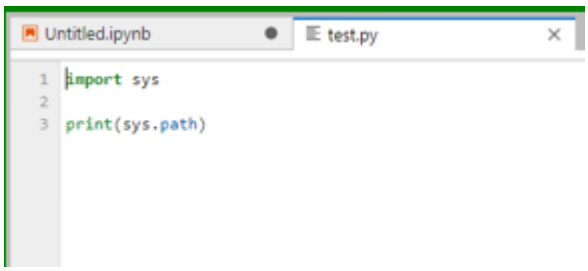


**图 5-15** 将“test.py”文件内容加载到.ipynb 文件里

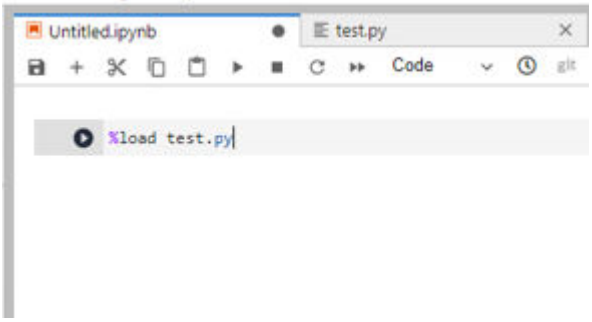
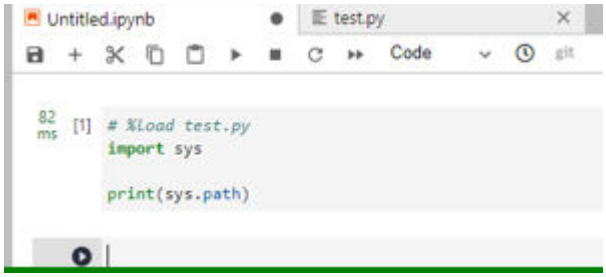



**图 5-16** 加载后的 ipynb 文件




## 5.7.2 Notebook 里面运行的实例，如果重启，数据集会丢失么?

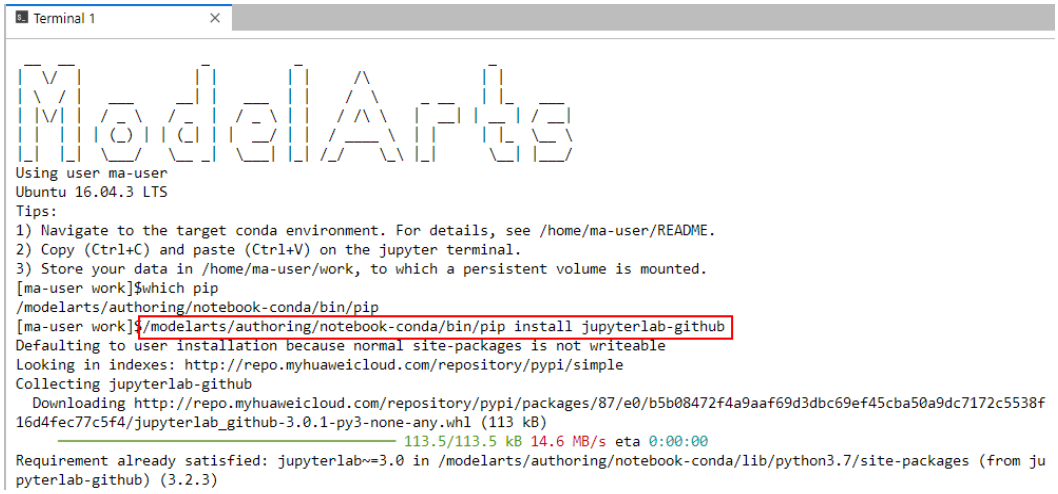
ModelArts提供的Notebook实例是以ma-user启动的，用户进入实例后，工作目录默认是“/home/ma-user/work”。

创建实例，“/home/ma-user/work”目录下挂载的数据，在实例停止、重新启动后依然保留，其他目录下的内容会还原。

## 5.7.3 Jupyter 可以安装插件吗?

Jupyter可以安装插件。

目前jupyter插件多数采用wheel包的形式发布，一次性完成前后端插件的安装，安装时注意使用jupyter服务依赖的环境“/modelarts/authoring/notebook-conda/bin/pip”进行安装，不要使用默认的anaconda（kernel依赖的python环境）的pip进行安装。



使用命令**jupyter labextension list --app-dir=/home/ma-user/.lab/console**查询

前端插件安装目录为：/home/ma-user/.local/share/jupyter/labextensions

```
[ma-user work]$jupyter labextension list --app-dir=/home/ma-user/.lab/console
JupyterLab v3.2.3
/home/ma-user/.local/share/jupyter/labextensions
        @jupyterlab/github v3.0.1 enabled OK (python, jupyterlab_github)

/modelarts/authoring/notebook-conda/share/jupyter/labextensions
        jupyter-matplotlib v0.10.5 enabled OK
        jupyterlab-plotly v5.6.0 enabled OK
        jupyterlab_pygments v0.2.2 enabled OK (python, jupyterlab_pygments)
        nbdime-jupyterlab v2.1.1 enabled OK
        @jupyter-widgets/jupyterlab-manager v3.1.1 enabled OK (python, jupyterlab_widgets)
```

后端插件代码安装目录：/home/ma-user/.local/lib/python3.7/site-packages

```
(PyTorch-1.8) [ma-user site-packages]$pwd
/home/ma-user/.local/lib/python3.7/site-packages
(PyTorch-1.8) [ma-user site-packages]$tree
.
├── jupyterlab_github
│   ├── __init__.py
│   ├── __pycache__
│   │   ├── __init__.cpython-37.pyc
│   │   └── _version.cpython-37.pyc
│   ├── _version.py
│   └── labextension
│       ├── package.json
│       ├── schemas
│       │   └── @jupyterlab
│       │       └── github
│       │           ├── drive.json
│       │           └── package.json.orig
│       └── static
│           ├── 060d51417beb0f47daa10a4dcb2e147d.png
│           ├── 534.06ec9cb816bf9170af66.js
│           ├── 742.c0343f89d61252c41791.js
│           ├── 784.7af3b56872a0f8721f0b.js
│           ├── remoteEntry.e0263a028853908ae4bb.js
│           └── style.js
└── jupyterlab_github-3.0.1.dist-info
    ├── INSTALLER
    ├── LICENSE
    ├── METADATA
    ├── RECORD
    ├── REQUESTED
    ├── WHEEL
    └── top_level.txt

8 directories, 20 files
```

配置文件目录：/home/ma-user/.jupyter/

```
(PyTorch-1.8) [ma-user jupyter_server_config.d]$pwd
/home/ma-user/.jupyter/jupyter_server_config.d
(PyTorch-1.8) [ma-user jupyter_server_config.d]$tree
.
├── jupyter_server_mathjax.json
├── jupyter_tensorboard.json
├── jupyterlab.json
├── jupyterlab_git.json
├── jupyterlab_github.json
├── modelarts_notebook_plugin.json
├── nbclassic.json
├── nbdime.json
└── notebook_shim.json

0 directories, 9 files
```

后端插件使用**jupyter server extension list**命令查询。

```
[ma-user work]$jupyter server extension list
Config dir: /home/ma-user/.jupyter
    jupyter_server_mathjax enabled
    - Validating jupyter_server_mathjax...
      jupyter_server_mathjax  OK
    jupyter_tensorboard enabled
    - Validating jupyter_tensorboard...
      jupyter_tensorboard 0.2.0 OK
    jupyterlab enabled
    - Validating jupyterlab...
      jupyterlab 3.2.3 OK
    jupyterlab_git enabled
    - Validating jupyterlab_git...
      jupyterlab_git 0.34.0 OK
    modelarts_notebook_plugin enabled
    - Validating modelarts_notebook_plugin...
      modelarts_notebook_plugin  OK
    nbclassic enabled
    - Validating nbclassic...
      nbclassic  OK
    nbdime enabled
    - Validating nbdime...
      nbdime 3.1.1 OK
    notebook_shim enabled
    - Validating notebook_shim...
      notebook_shim  OK
```

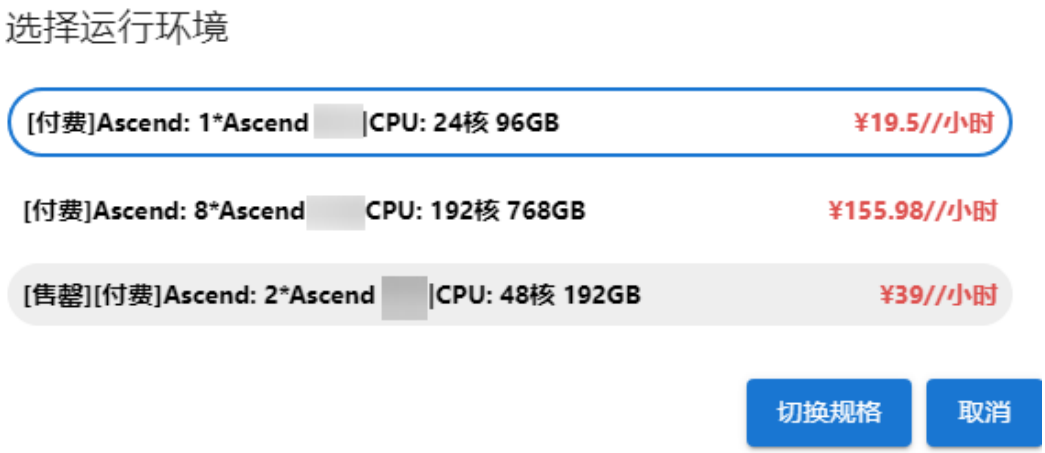
## 5.7.4 是否支持在 CodeLab 中使用昇腾的卡进行训练?

有两种情况。

第一种，在ModelArts控制台的“总览”界面打开CodeLab，使用的是CPU或GPU资源，无法使用昇腾卡训练。

第二种，如果是AI Gallery社区的Notebook案例，使用的资源是ASCEND的，“Run in ModelArts”跳转到CodeLab，就可以使用昇腾卡进行训练。


也支持切换规格




## 5.7.5 如何在 CodeLab 上安装依赖?

ModelArts CodeLab中已安装Jupyter、Python程序包等多种环境，您也可以使用**pip install**在Notebook或Terminal中安装依赖包。

### 在 Notebook 中安装

1. 在总览页面进入CodeLab。
2. 在“Notebook”区域下，新建一个ipynb文件。
3. 在新建的Notobook中，在代码输入栏输入如下命令。

   **!pip install xxx**

### 在 Terminal 中安装

在Terminal里激活需要的anaconda python环境后再进行安装。

例如，通过terminal在“TensorFlow-1.8”的环境中使用**pip**安装Shapely。

1. 在总览页面进入CodeLab。
2. 在“Other”区域下，选择“Terminal”，新建一个terminal文件。
3. 在代码输入栏输入以下命令，获取当前环境的kernel，并激活需要安装依赖的python环境。

   **cat /home/ma-user/README**

   **source /home/ma-user/anaconda3/bin/activate TensorFlow-1.8**

   **说明**

   如果需要在其他python环境里安装，请将命令中“TensorFlow-1.8”替换为其他引擎。
4. 在代码输入栏输入以下命令安装Shapely。

   **pip install Shapely**

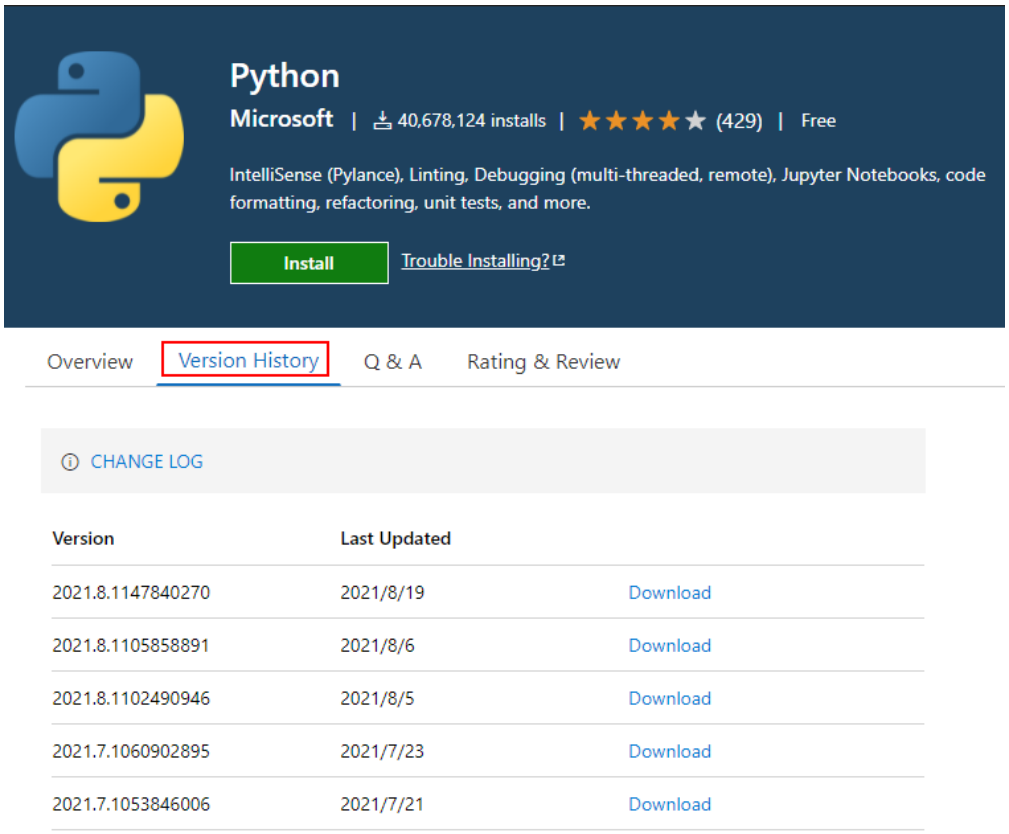
# 5.8 VS Code 使用技巧

## 5.8.1 安装远端插件时不稳定，需尝试多次

**方法一：**离线包安装方式（推荐）

1. 到VS Code插件官网vscode_marketplace搜索待安装的Python插件，**Python插件路径**。
2. 单击进入Python插件的Version History页签后，下载该插件的离线安装包，如图所示。

**图 5-17** Python 插件离线安装包




3. 在本地VS Code环境中，将下载好的.vsix文件拖动到远端Notebook中。
4. 右键单击该文件，选择Install Extension VSIX。

**方法二：**设置远端默认安装的插件

按照**VS Code中设置远端默认安装的插件**配置，即会在连接远端时自动安装，减少等待时间。

方法三：VS Code官网排查方式**https://code.visualstudio.com/docs/remote/troubleshooting**

小技巧（按需调整远端连接的相关参数）：

```
"remote.SSH.connectTimeout": 10,
"remote.SSH.maxReconnectionAttempts": null,
"remote.downloadExtensionsLocally": true,
"remote.SSH.useLocalServer": false,
"remote.SSH.localServerDownload": "always",
```

## 5.8.2 Notebook 实例重新启动后，需要删除本地 known_hosts 才能连接

可以在本地的ssh config文件中对这个Notebook配置参数“StrictHostKeyChecking no”和“UserKnownHostsFile=/dev/null”，如下参考所示：

```
Host roma-local-cpu
   HostName x.x.x.x #IP地址
   Port 22522
   User ma-user
   IdentityFile C:/Users/my.pem
```

```
StrictHostKeyChecking no
ForwardAgent yes
```

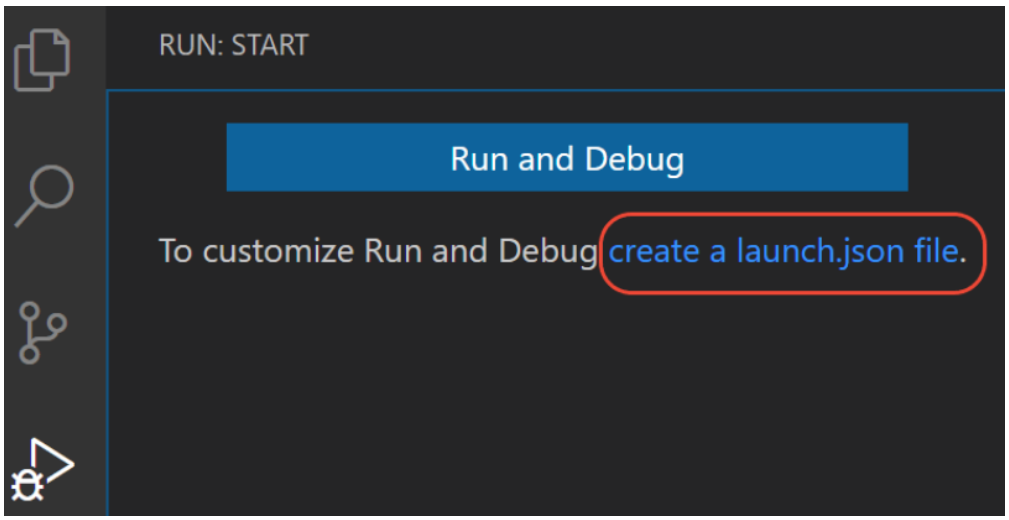
提示：因为SSH登录时会忽略known_hosts文件，有安全风险

## 5.8.3 使用 VS Code 调试代码时不能进入源码

如果已有launch.json文件，请直接看步骤三。

**步骤一：打开launch.json文件**

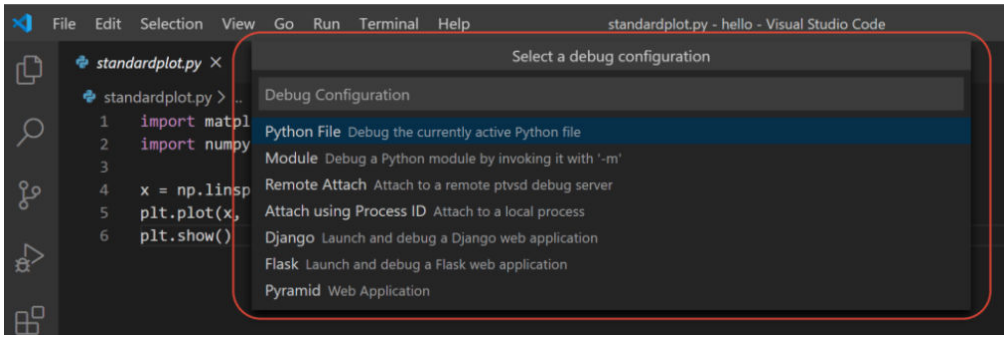
- 方法一：单击左侧菜单栏的Run（Ctrl+Shift+D）按钮，再单击create a launch.json file。如下图所示：




- 方法二：单击上侧菜单栏中的Run > Open configurations按钮

**步骤二：选择语言**

如果需要对Python语言进行设置，在弹出的Select a debug configuration中选择Python File，其他语言操作类似。如下图所示：

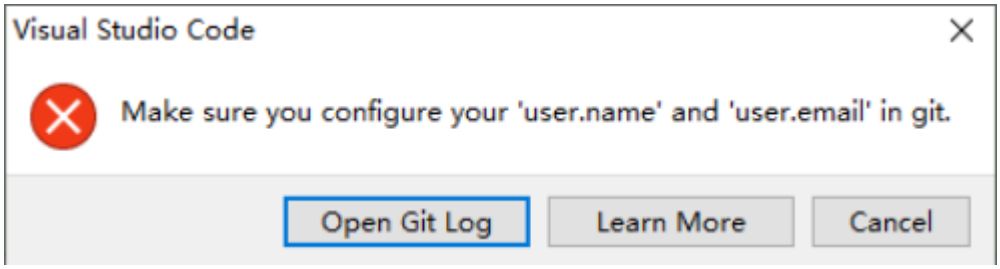



**步骤三：编辑launch.json，增加justMyCode": false配置，如下所示。**

```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "name": "Python: 当前文件",
            "type": "python",
            "request": "launch",
```

```
            "program": "${file}",
            "console": "integratedTerminal",
            "justMyCode": false
        }
    ]
}
```

## 5.8.4 使用 VS Code 提交代码时弹出对话框提示用户名和用户邮箱配置错误




1. 在VS Code环境中，执行Ctrl+Shift+P。
2. 搜Python: Select Interpreter，选择对应的Python环境。
3. 单击页面上方的“Terminal > New Terminal”，此时打开的命令行界面即为远端容器环境命令行。
4. 在VS Code的terminal中，执行如下述命令，再重试提交即可。

```
git config --global user.email xxx@xxx.com #改为你的用户邮箱
git config --global user.name xxx #改为你的用户名
```

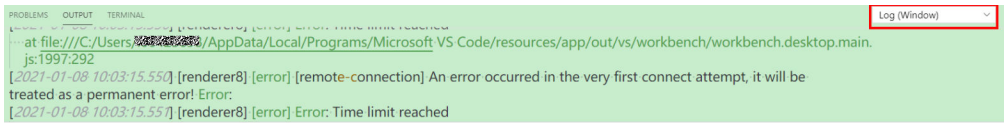
## 5.8.5 实例重新启动后，Notebook 内安装的插件丢失

请使用**镜像保存功能**。

## 5.8.6 VS Code 中查看远端日志

1. 在VS Code环境中执行Ctrl+Shift+P
2. 搜show logs
3. 选择Remote Server。

也可在如下截图的红框处切换至其他的Log




## 5.8.7 打开 VS Code 的配置文件 settings.json

1. 在VS Code环境中执行Ctrl+Shift+P
2. 搜Open Settings（JSON）

## 5.8.8 VS Code 背景配置为豆沙绿

在VS Code的配置文件settings.json中添加如下参数

```
"workbench.colorTheme": "Atom One Light",
"workbench.colorCustomizations": {
"[Atom One Light]": {
```

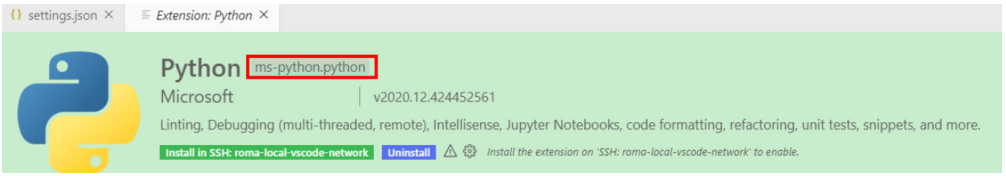
```
 "editor.background": "#C7EDCC",
"sideBar.background": "#e7f0e7",
"activityBar.background": "#C7EDCC",
    },
},
```

## 5.8.9 VS Code 中设置远端默认安装的插件

在VS Code的配置文件settings.json中添加remote.SSH.defaultExtensions参数，如自动安装Python和Maven插件，可配置如下。

```
"remote.SSH.defaultExtensions": [
    "ms-python.python",
    "vscjava.vscode-maven"
  ],
```

其中，插件名称可以单击VS Code的某个插件后获取，如下所示。




## 5.8.10 VS Code 中把本地的指定插件安装到远端或把远端插件安装到本地

1. 在VS Code的环境中执行Ctrl+Shift+P
2. 搜install local，按需选择即可

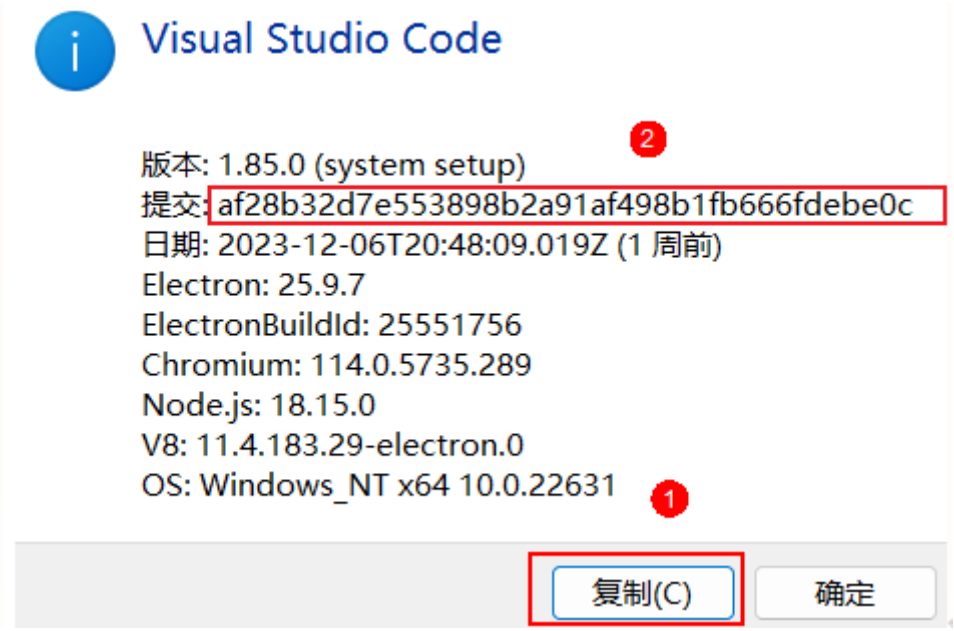
## 5.8.11 Notebook 如何离线安装 VS Code Server

### 背景介绍

VS Code执行remote-ssh远程连接时，会根据用户的VS Code版本去自动更新vscode-server和Vscode-client的版本，通过本地和远端尝试下载相关的安装脚本和包。当远端网络和本地网络不通时，可以手动下载对应版本的Vscode-server包，然后离线安装。

### 操作步骤

1. 打开VS Code，单击“Help > About”，弹出如下对话框，单击“复制”，然后记录提交码。




2. 替换如下链接的comment id（1获取的），使用浏览器下载相应版本的vscode-server-linux-x64.tar.gz文件，如果下载报错“InvalidUri”，切换国外代理或者检查网络。

   **下载URL：**

   - arm版本，下载vscode-server-linux-arm64.tar.gz

     https://update.code.visualstudio.com/commit:${commitID}/server-linux-arm64/stable

   - x86版本，下载vscode-server-linux-x64.tar.gz

     https://update.code.visualstudio.com/commit:${commitID}/server-linux-x64/stable

3. 将下载的vscode-server-linux-x64.tar.gz，上传到ModelArts实例的/home/ma-user/work目录下。

   ```
   (PyTorch-1.4) [ma-user work]$ls vscode-server*
   vscode-server-linux-x64.tar.gz
   (PyTorch-1.4) [ma-user work]$pwd
   /home/ma-user/work
   (PyTorch-1.4) [ma-user work]$
   ```

4. 执行下面命令，并指定commitId。

   ```
   commitId=<提交的ID码>
   mkdir -p /home/ma-user/.vscode-server/bin/$commitId
   tar -zxvf vscode-server-linux-x64.tar.gz -C /home/ma-user/.vscode-server/bin/$commitId  --strip=1
   chmod 750 -R /home/ma-user/.vscode-server/bin/$commitId
   ```

5. 关闭VS Code，重新进入Notebook实例列表页面打开VS Code。

# 5.9 VS Code 连接开发环境失败常见问题

# 5.9.1 在 ModelArts 控制台界面上单击 VS Code 接入并在新界面单击打开，未弹出 VS Code 窗口

## 原因分析

未安装VS Code或者安装版本过低。

## 解决方法

下载并安装VS Code（Windows用户请单击“Win”，其他用户请单击“其他”下载），安装完成后单击“刷新”完成连接。


# 5.9.2 在 ModelArts 控制台界面上单击 VS Code 接入并在新界面单击打开，VS Code 打开后未进行远程连接

**须知**

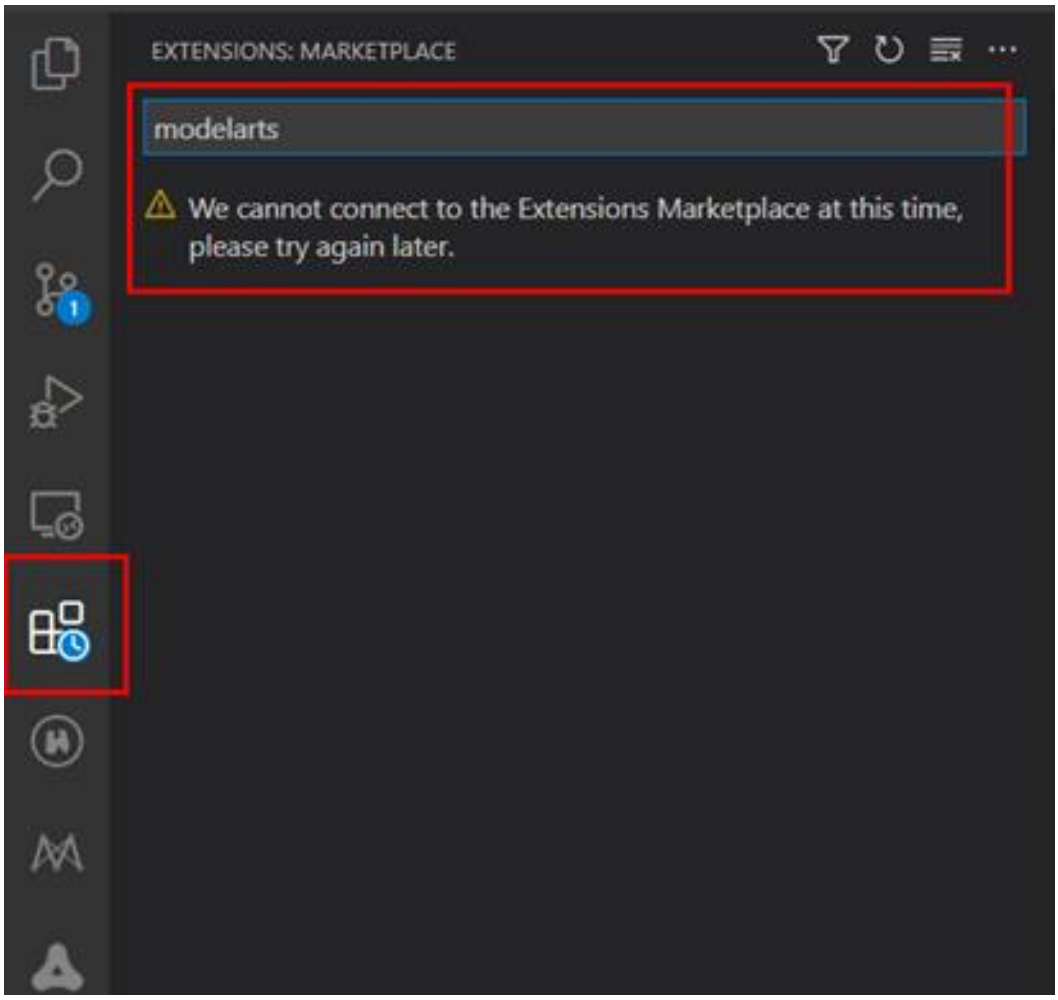
若本地为Linux系统，见原因分析二。

## 原因分析一

自动安装VS Code插件ModelArts-HuaweiCloud失败。

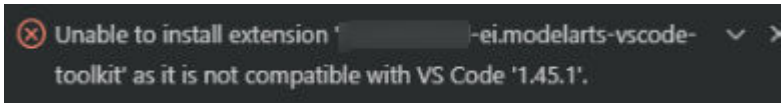
## 解决方法一

方法一：检查VS Code网络是否正常。在VS Code插件市场上搜索ModelArts-HuaweiCloud，如果显示如下则网络异常，请切换代理或使用其他网络。




操作完成后再次执行搜索，若显示如下则网络正常，请回到ModelArts控制台界面再次单击界面上的“VS Code接入”按钮。

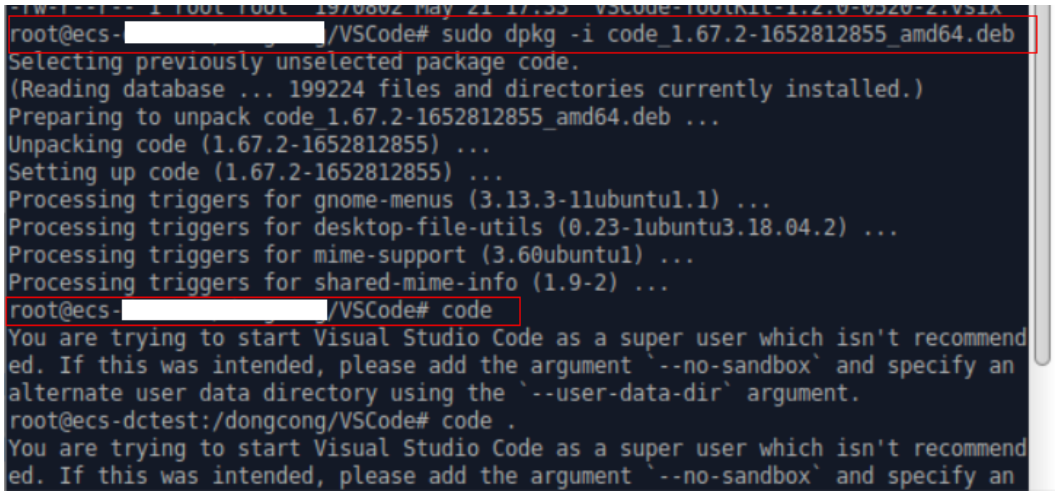

方法二：出现如下图报错，是由于VS Code版本过低，建议升级VS Code版本为1.57.1或者最新版。

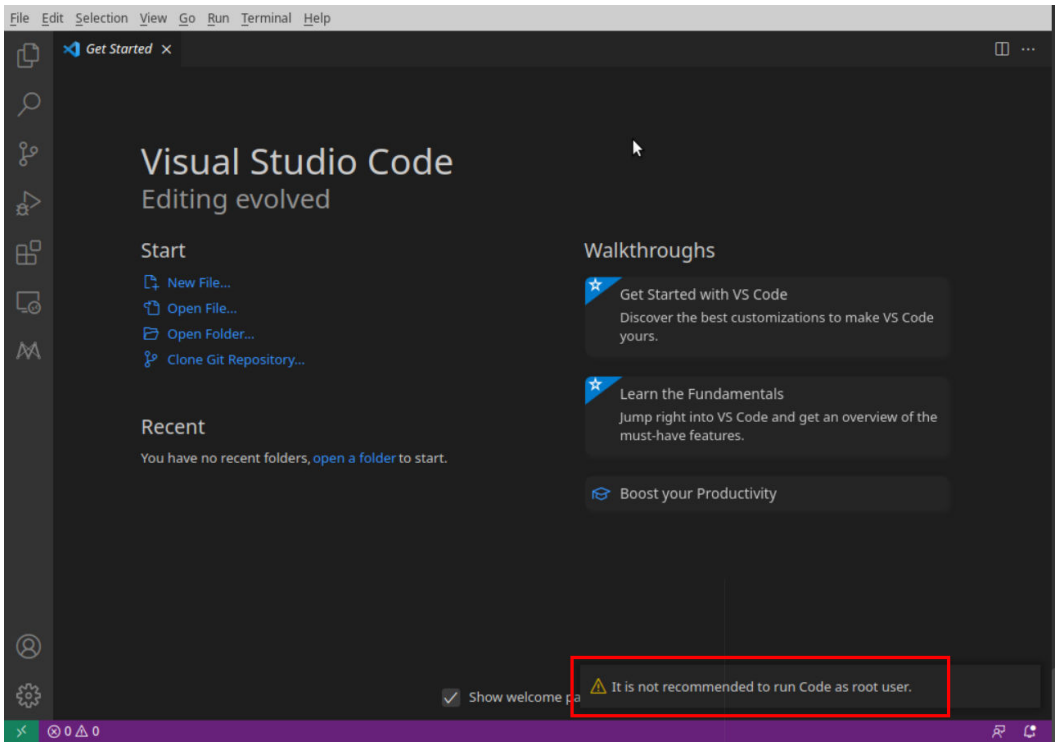



## 原因分析二

本地系统为Linux，由于使用root用户安装VS Code，打开VS Code显示信息It is not recommended to run Code as root user

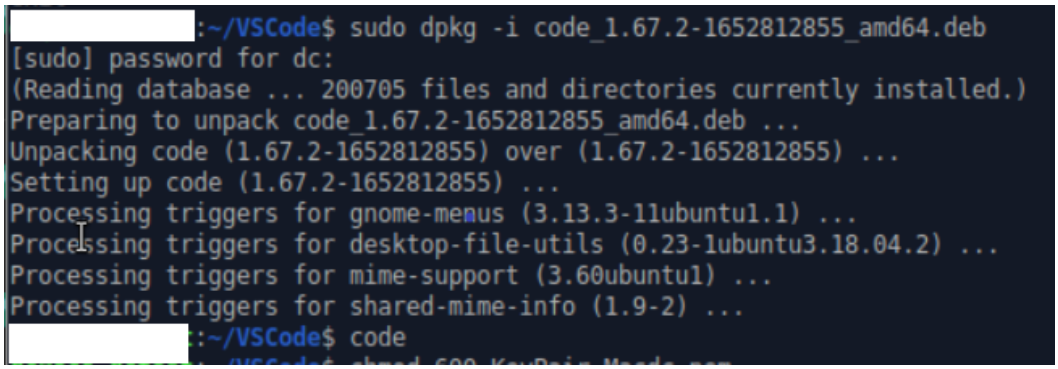


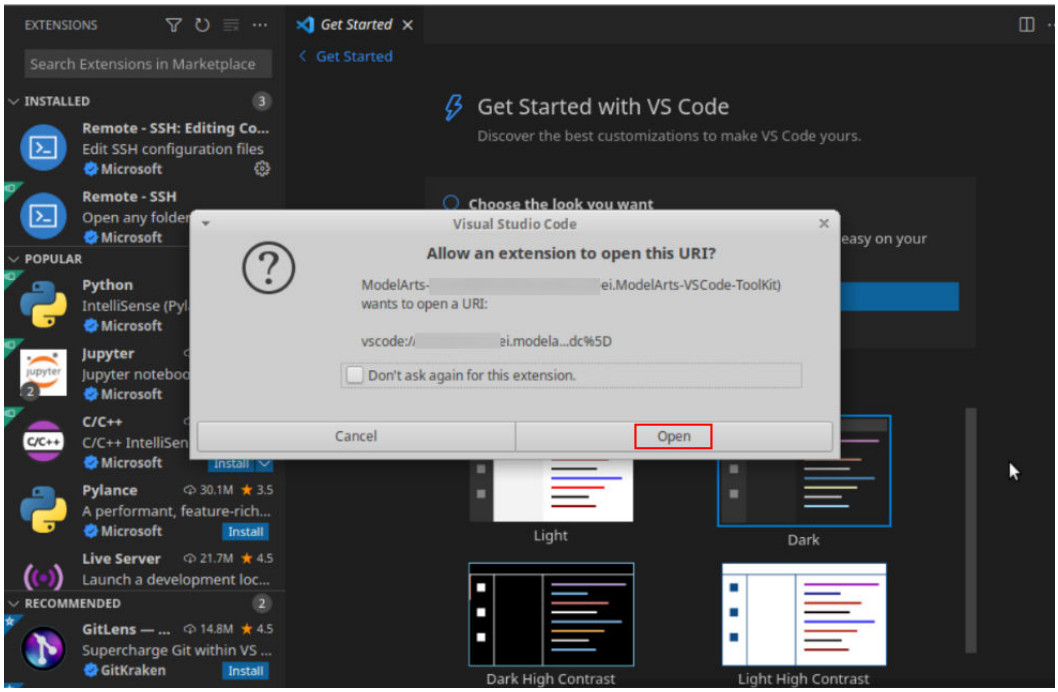




## 解决方法二

请使用非root用户安装VS Code后，回到ModelArts控制台界面再次单击界面上的“VS Code接入”按钮。

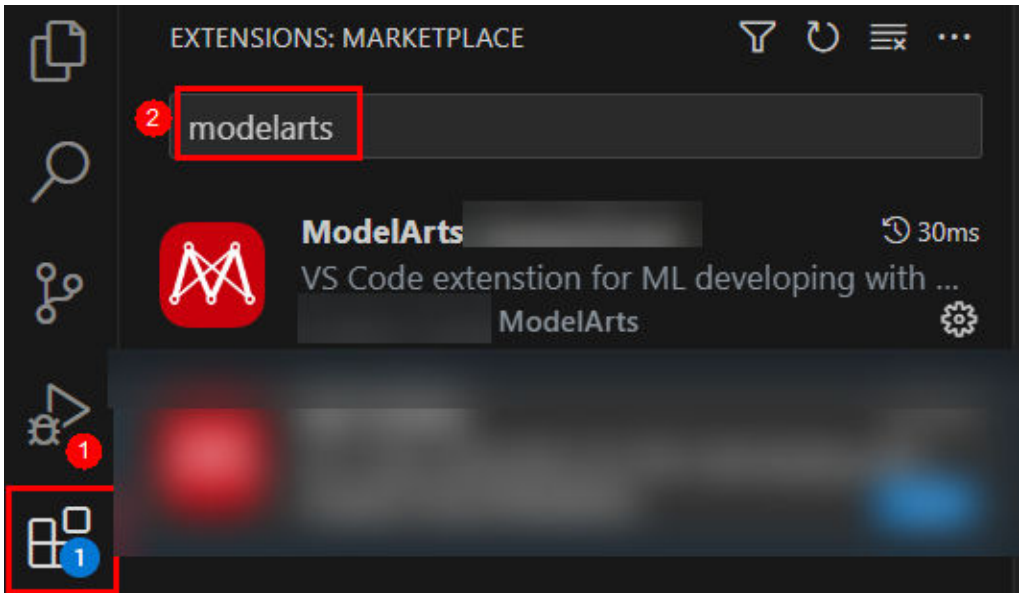



## 5.9.3 VS Code 连接开发环境失败时，请先进行基础问题排查

VS Code连接开发环境失败时，请参考以下步骤进行基础排查：

**步骤1** 排查插件包是否为最新版：在extensions中搜索，看是否需要升级。



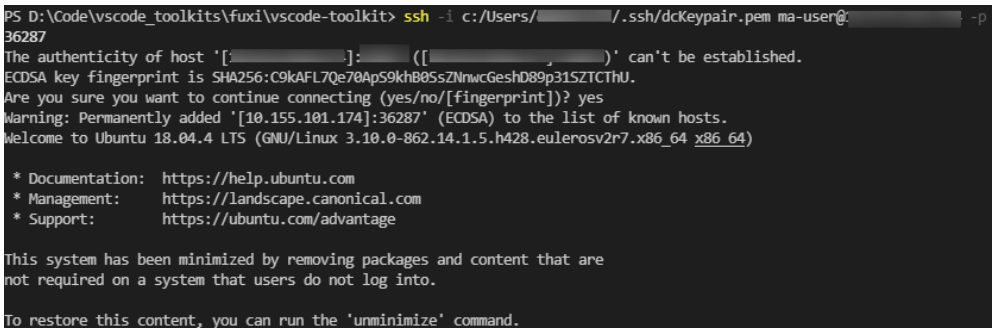

**步骤2** 检查实例状态是否为运行中，如果是，请执行下一步继续排查。

**步骤3** 在VS Code的Terminal中执行如下命令，连接到远端开发环境。

```
ssh -tt -o StrictHostKeyChecking=no -i ${IdentityFile} ${User}@${HostName} -p ${Port}
```

参数说明：

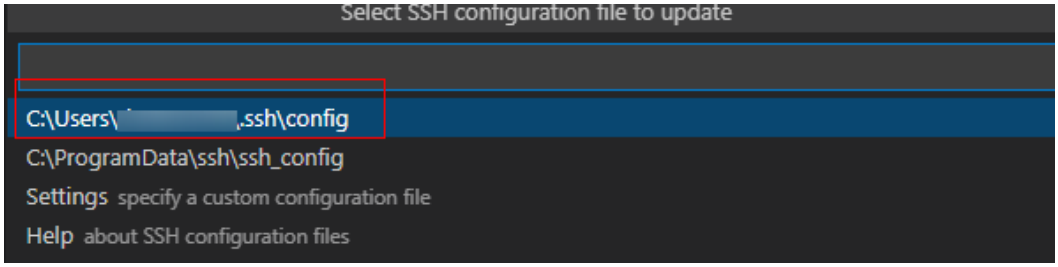
- IdentityFile：本地密钥路径
- User：用户名，例如：ma-user
- HostName：IP地址
- Port：端口号




如可以连接上，请执行下一步继续排查。

**步骤4** 检查配置是否正确，如果正确请执行下一步继续排查。

打开config文件进行检查，如图所示：



```
HOST remote-dev
    hostname <instance connection host>
    port <instance connection port>
    user ma-user
    IdentityFile ~/.ssh/test.pem
    StrictHostKeyChecking no
    UserKnownHostsFile /dev/null
    ForwardAgent yes
```

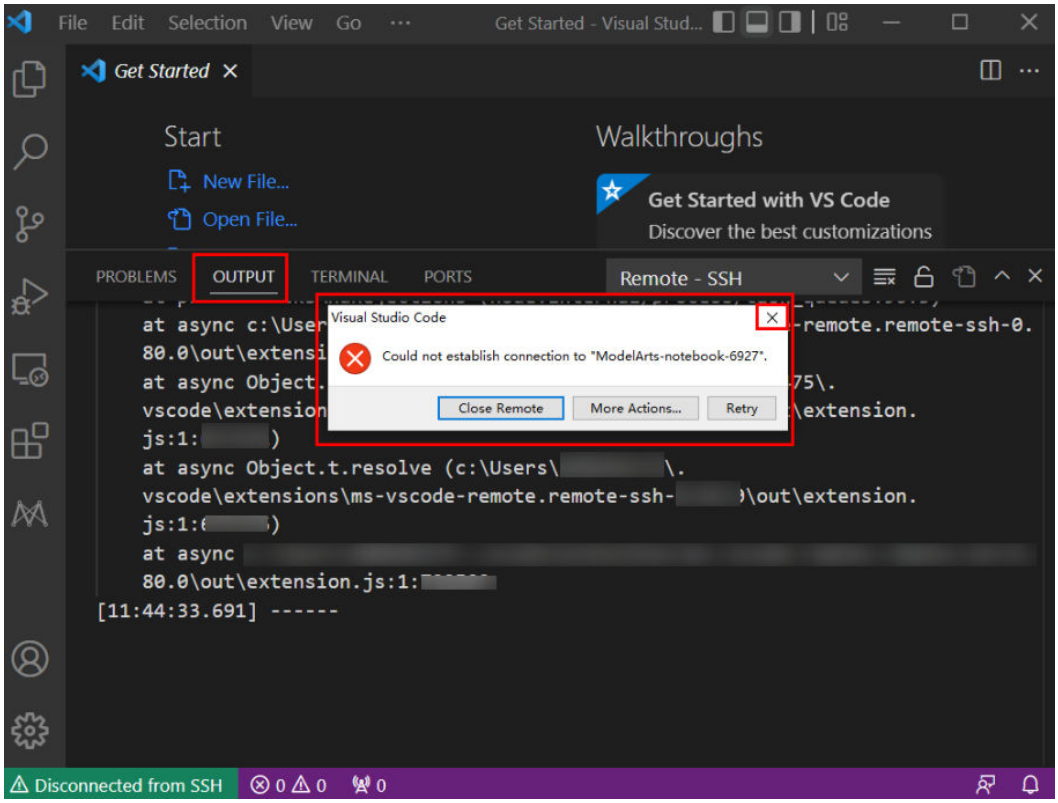
**步骤5** 查看密钥文件，建议放在C:\Users\xx.ssh下，并确保密钥文件无中文字符。

**步骤6** 如果还未解决，请参考后续章节的FAQ处理。

**----结束**

## 5.9.4 远程连接出现弹窗报错：Could not establish connection to xxx

### 问题现象




### 原因分析

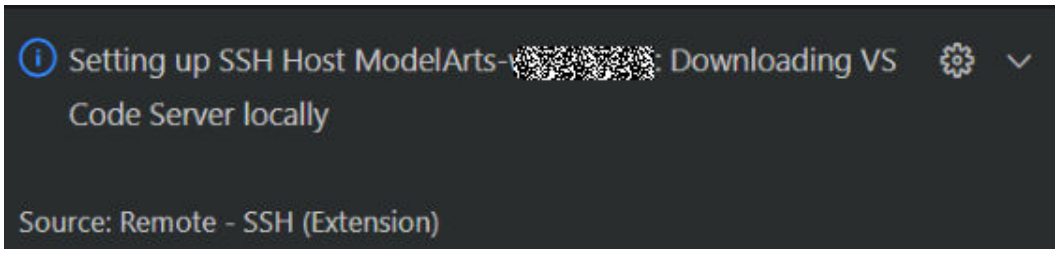
执行VS Code Remote SSH连接失败。

### 解决方法

单击弹窗右上角关闭弹窗，查看OUTPUT中的具体报错信息，并参考后续章节列举的几种常见报错解决问题。

# 5.9.5 连接远端开发环境时，一直处于"Setting up SSH Host xxx: Downloading VS Code Server locally"超过 10 分钟以上，如何解决?

## 问题现象



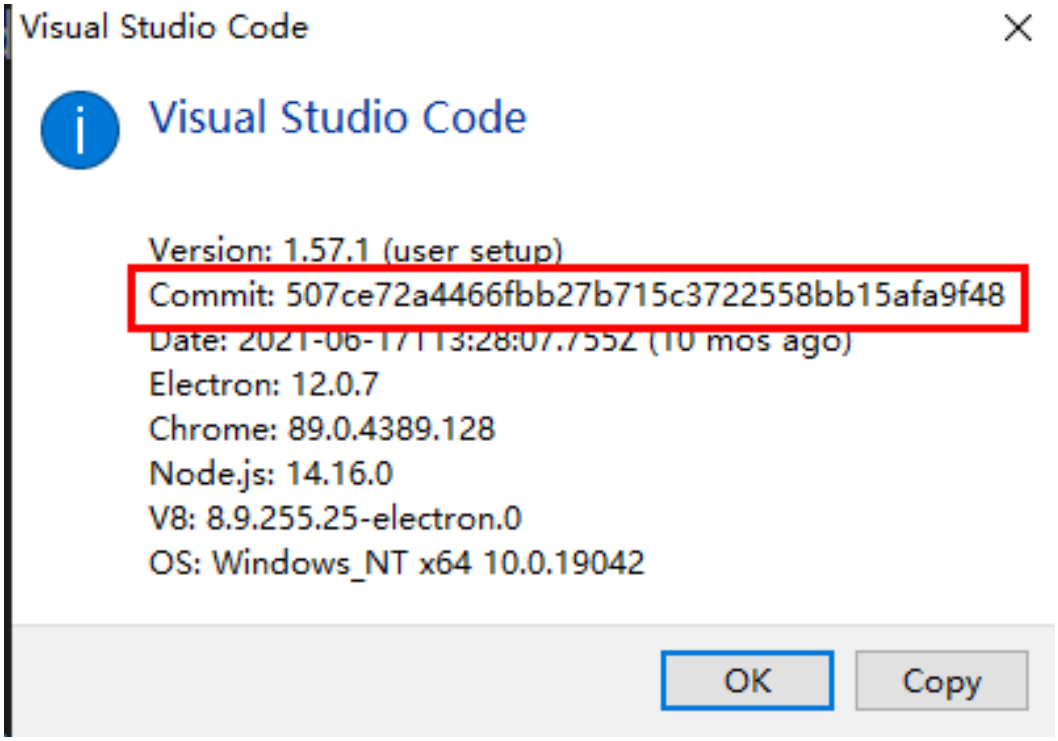

## 原因分析

当前本地网络原因，导致远程自动安装VS Code Server时间过长。

## 解决方法

手动安装vscode-server。

**步骤1** 获取VS Code的commitID




**步骤2** 下载相应版本vscode-server压缩包，请根据开发环境cpu架构选择arm版本或x86版本。

**说明**

替换下面链接中${commitID}为**步骤1 获取VS Code的commitID**中commitID。

- arm版本，下载vscode-server-linux-arm64.tar.gz

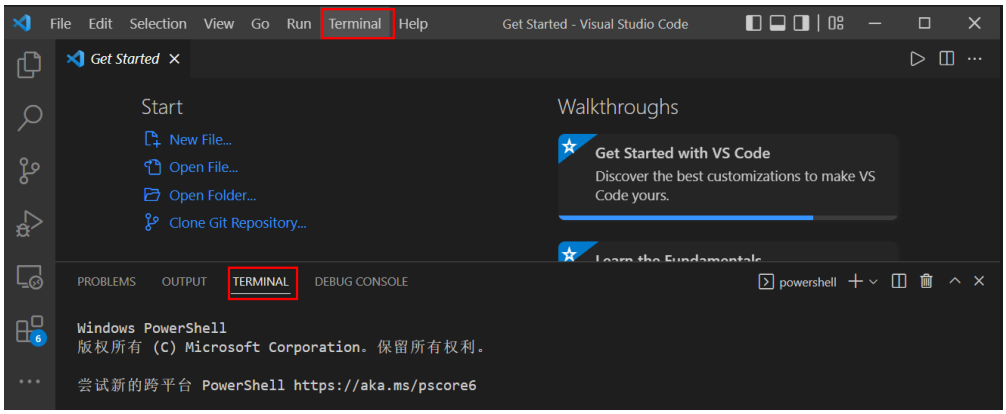
  https://update.code.visualstudio.com/commit:${commitID}/server-linux-arm64/stable

- x86版本，下载vscode-server-linux-x64.tar.gz

  https://update.code.visualstudio.com/commit:${commitID}/server-linux-x64/stable

**步骤3** 进入远程环境。

打开VS Code中的Terminal。




在VS Code的Terminal中执行如下命令，连接到远端开发环境。

```
ssh -tt -o StrictHostKeyChecking=no -i ${IdentityFile} ${User}@${HostName} -p ${Port}
```

参数说明：

- IdentityFile：本地密钥路径

- User：用户名，例如：ma-user

- HostName：IP地址

- Port：端口号


**步骤4** 手动安装vscode-server。

在VS Code的Terminal中执行如下命令，清空残留的vscode-server，注意替换命令中${commitID}为**步骤1 获取VS Code的commitID**中commitID。

```
rm -rf /home/ma-user/.vscode-server/bin/${commitID}/*
mkdir -p /home/ma-user/.vscode-server/bin/${commitID}
```

上传vscode-server压缩包到开发环境。执行如下命令：

```
exit
scp -i xxx.pem -P 31205 本地vscode-server压缩包路径 ma-user@xxx:/home/ma-user/.vscode-server/bin
ssh -tt -o StrictHostKeyChecking=no -i ${IdentityFile} ${User}@${HostName} -p ${Port}
```

参数说明：

- IdentityFile：本地密钥路径

- User：用户名，例如：ma-user

- HostName：IP地址

- Port：端口号

以arm版本为例，将vscode-server压缩包解压至$HOME/.vscode-server/bin文件夹，注意替换命令中${commitID}为**步骤1 获取VS Code的commitID**中commitID。

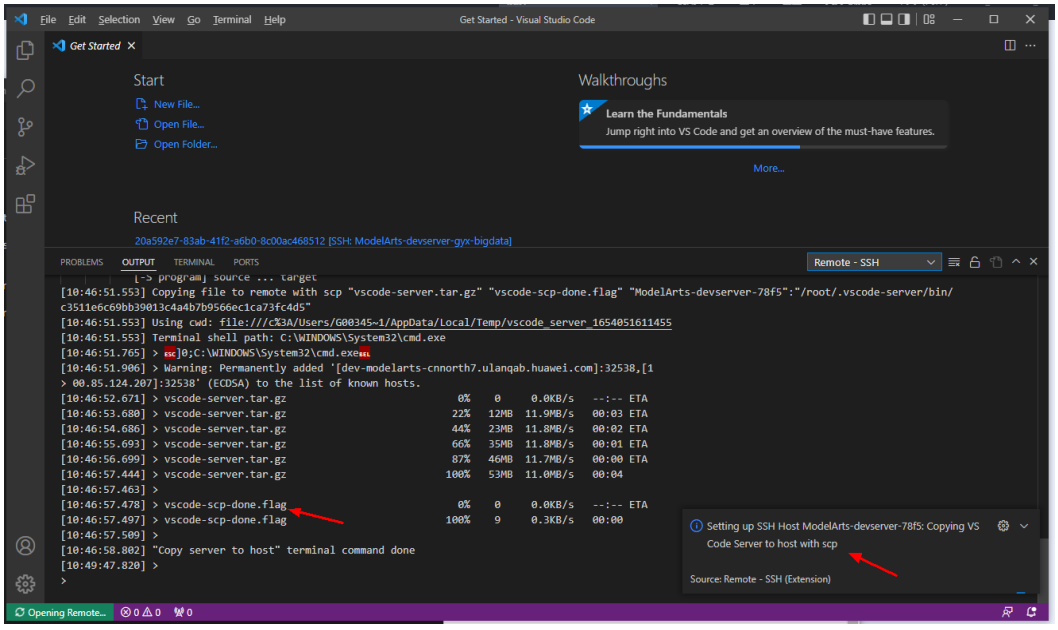
```
cd /home/ma-user/.vscode-server/bin
tar -zxf vscode-server-linux-arm64.tar.gz
mv vscode-server-linux-arm64/* ${commitID}
```

**步骤5** 重新远程连接。

**----结束**

# 5.9.6 连接远端开发环境时，一直处于"Setting up SSH Host xxx: Copying VS Code Server to host with scp"超过 10 分钟以上，如何解决?

## 问题现象




## 原因分析

通过查看日志发现本地vscode-scp-done.flag显示成功上传，但远端未接收到。

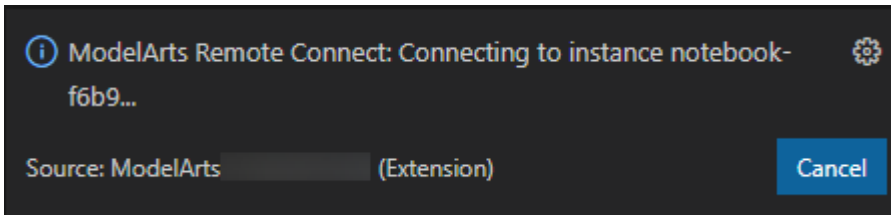
## 解决方法

关闭VS Code所有窗口后，回到ModelArts控制台界面再次单击界面上的“VS Code接入”按钮。

# 5.9.7 连接远端开发环境时，一直处于"ModelArts Remote Connect: Connecting to instance xxx..."超过 10 分钟以上，如何解决?

## 问题现象

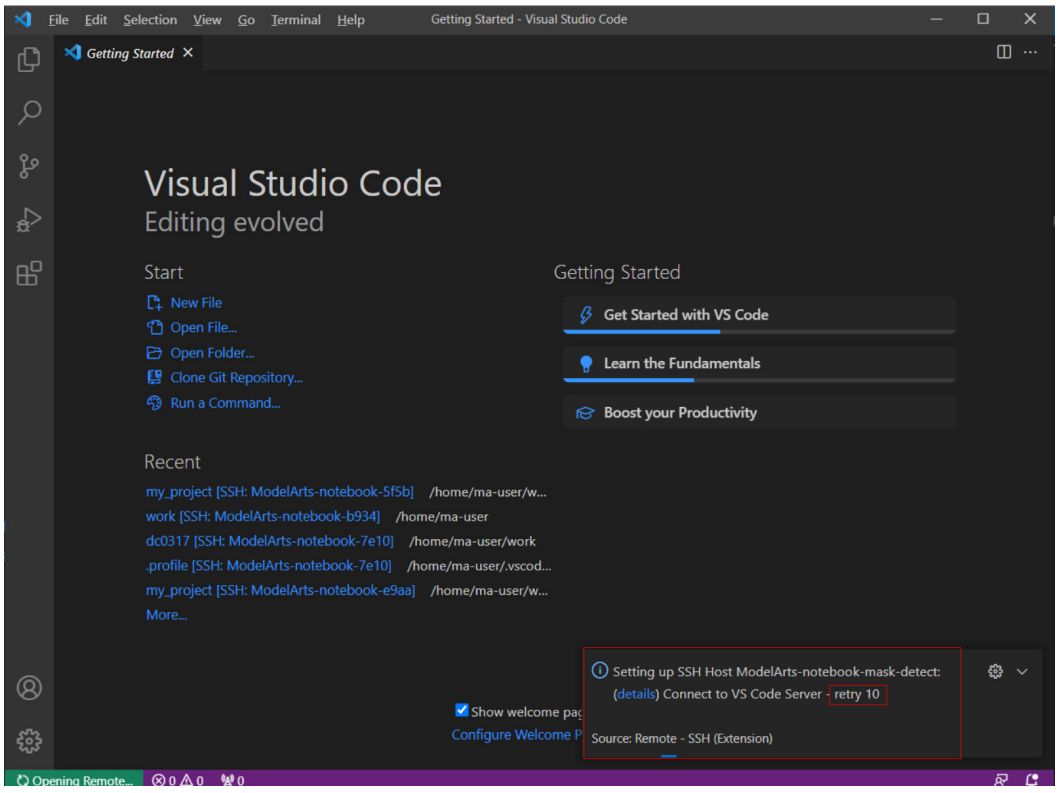



## 解决方法

单击“Canel”，并回到ModelArts控制台界面再次单击界面上的“VS Code接入”按钮。

# 5.9.8 远程连接处于 retry 状态如何解决?

## 问题现象



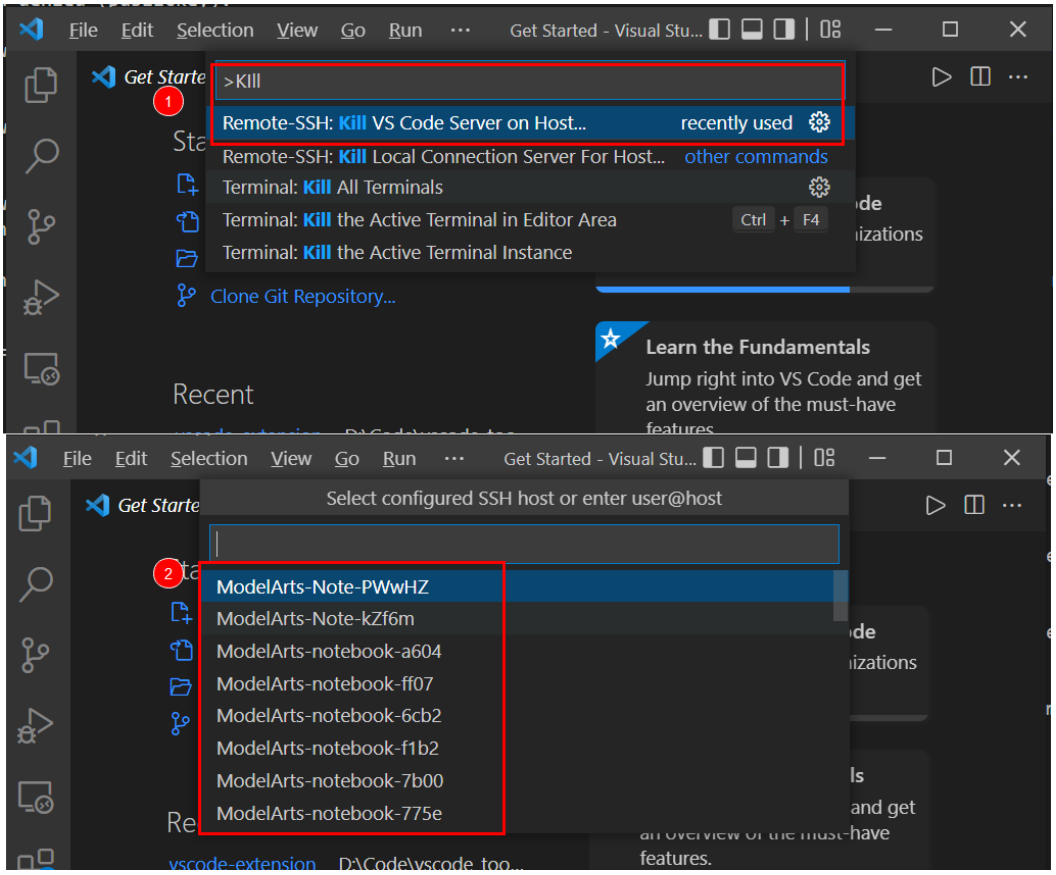

## 原因分析

之前下载VS Code server失败，有残留信息，导致本次无法下载。

## 解决方法

方法一（本地）：打开命令面板（Windows： Ctrl+Shift+P，macOS：Cmd+Shift+P），搜索“Kill VS Code Server on Host”，选择出问题的实例进行自动清除，然后重新进行连接。

**图 5-18** 清除异常的实例




方法二（远端）：在VS Code的Terminal中删除“/home/ma-user/.vscode-server/bin/”下正在使用的文件，然后重新进行连接。

```
ssh -tt -o StrictHostKeyChecking=no -i ${IdentityFile} ${User}@${HostName} -p ${Port}
rm -rf /home/ma-user/.vscode-server/bin/
```

参数说明：

- IdentityFile：本地密钥路径
- User：用户名，例如：ma-user
- HostName：IP地址
- Port：端口号

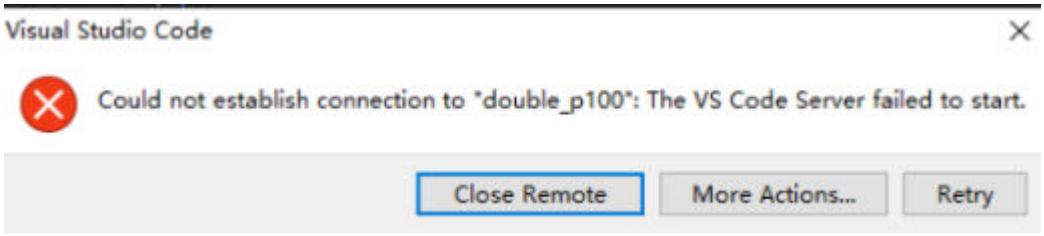
**说明**

vscode-server相关问题也可以使用上述的解决方法。

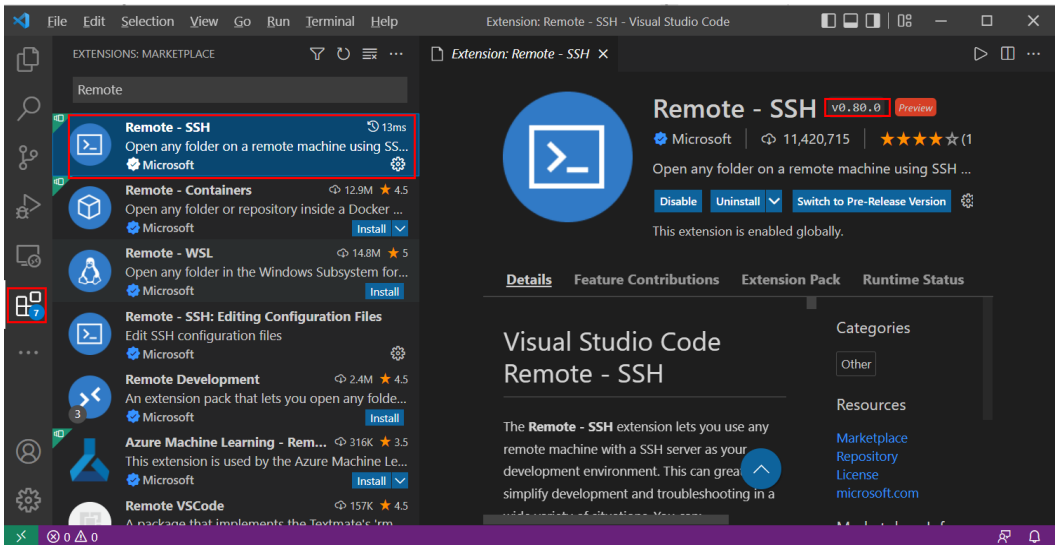
# 5.9.9 报错“The VS Code Server failed to start”如何解决?

## 问题现象



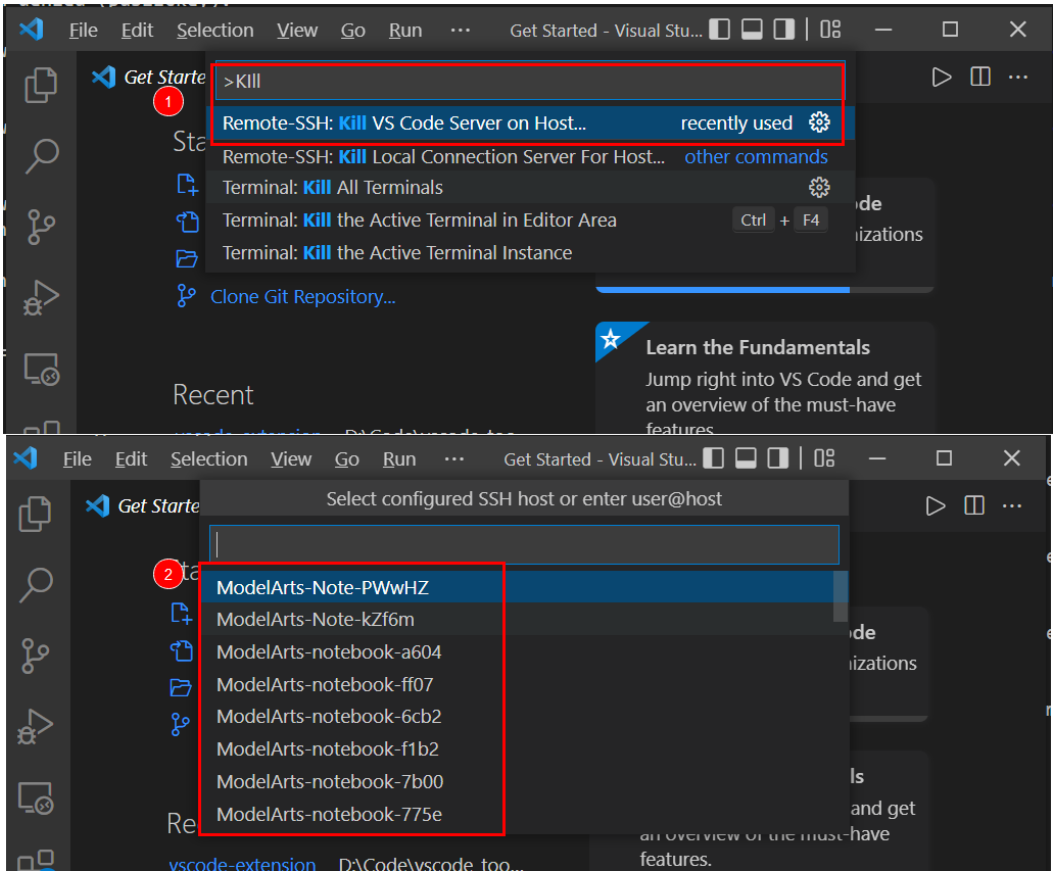

## 解决方法

**步骤1** 检查VS Code版本是否为1.78.2或更高版本，如果是，请查看Remote-SSH版本，如果Remote-SSH版本低于v0.76.1，请升级Remote-SSH。




**步骤2** 打开命令面板（Windows： Ctrl+Shift+P，macOS：Cmd+Shift+P），搜索“Kill VS Code Server on Host”，选择出问题的实例进行自动清除，然后重新进行连接。

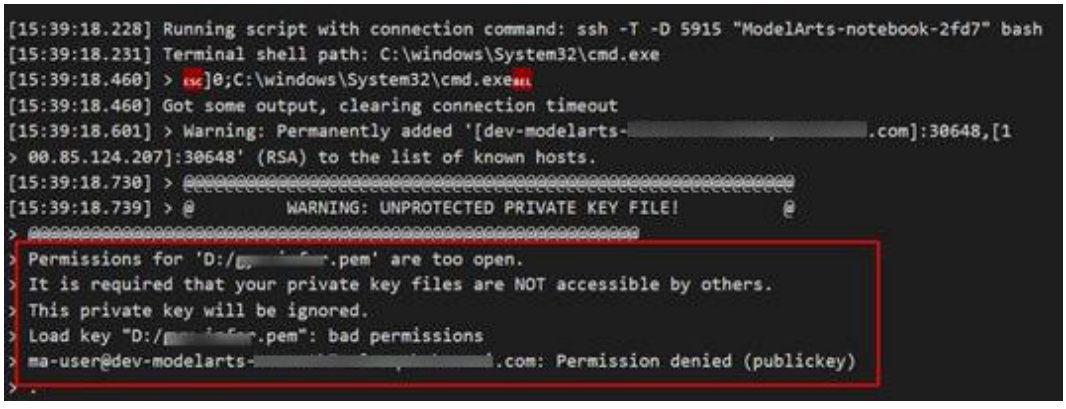
图 5-19 清除异常的实例




----结束

## 5.9.10 报错“Permissions for 'x:/xxx.pem' are too open”如何解决?

### 问题现象




### 原因分析

原因分析一：密钥文件未放在指定路径，详情请参考**安全限制**或**VS Code文档**。请参考解决方法一处理。

原因分析二：当操作系统为macOS/Linux时，可能是密钥文件或放置密钥的文件夹权限问题，请参考解决方法二处理。

## 解决方法

解决方法一：

请将密钥放在如下路径或其子路径下：

Windows：C:\Users\{{user}}

macOS/Linux： Users/{{user}}

解决方法二：

请**检查文件和文件夹权限**。

Local SSH file and folder permissions

**macOS / Linux:**

On your local machine, make sure the following permissions are set:

| Folder / File | Permissions |
| --- | --- |
| `.ssh` in your user folder | `chmod 700 ~/.ssh` |
| `.ssh/config` in your user folder | `chmod 600 ~/.ssh/config` |
| `.ssh/id_rsa.pub` in your user folder | `chmod 600 ~/.ssh/id_rsa.pub` |
| Any other key file | `chmod 600 /path/to/key/file` |

**Windows:**

The specific expected permissions can vary depending on the exact SSH implementation you are using. We recommend using the out of box Windows 10 OpenSSH Client.

In this case, make sure that all of the files in the `.ssh` folder for your remote user on the SSH host is owned by you and no other user has permissions to access it. See the Windows OpenSSH wiki for details.

For all other clients, consult your client's documentation for what the implementation expects.

# 5.9.11 报错“Bad owner or permissions on C:\Users\Administrator/.ssh/config”或“Connection permission denied (publickey)”如何解决?

## 问题现象

报错“Bad owner or permissions on C:\Users\Administrator/.ssh/config”或“Connection permission denied (publickey). Please make sure the key file is correctly selected and the file permission is correct. You can view the instance keypair information on ModelArts console.”

## 原因分析

文件夹“.ssh”的权限不仅是Windows当前用户拥有，或者当前用户权限不足，故修改权限即可。

## 解决方案

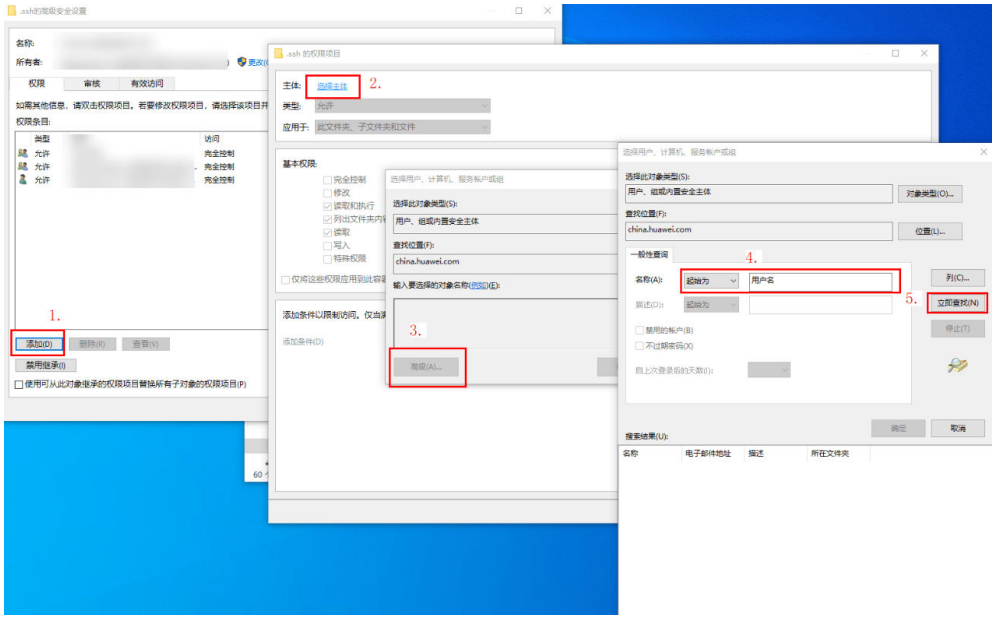
1. 找到.ssh文件夹。一般位于“C:\Users”，例如“C:\Users\xxx”。

   **说明**

   “C:\Users”目录下的文件名必须和Windows登录用户名完全一致。

2. 右键单击.ssh文件夹，选择“属性”。然后单击“安全”页签。
3. 单击“高级”，在弹出的高级安全设置界面单击“禁用继承”， 在弹出的“阻止继承”窗口单击“从此对象中删除所有继承的权限”。此时所有用户都将被删除。
4. 添加所有者：在同一窗口中，单击“添加”，在弹出的新窗口中，单击“主体”后面的“选择主体”，弹出“选择用户，计算机，服务账户或组”窗口，单击“高级”，输入用户名， 单击“立即查找”按钮，显示用户搜索结果列表。 选择您的用户账户，然后单击“确定”（大约四个窗口）以关闭所有窗口。

   **图 5-20** 添加所有者

5. 完成所有操作后，再次关闭并打开VS Code并尝试连接到远程SSH主机。备注：此时密钥需放到.ssh文件夹中。

# 5.9.12 报错“ssh: connect to host xxx.pem port xxxxx: Connection refused”如何解决?

## 问题现象


## 原因分析

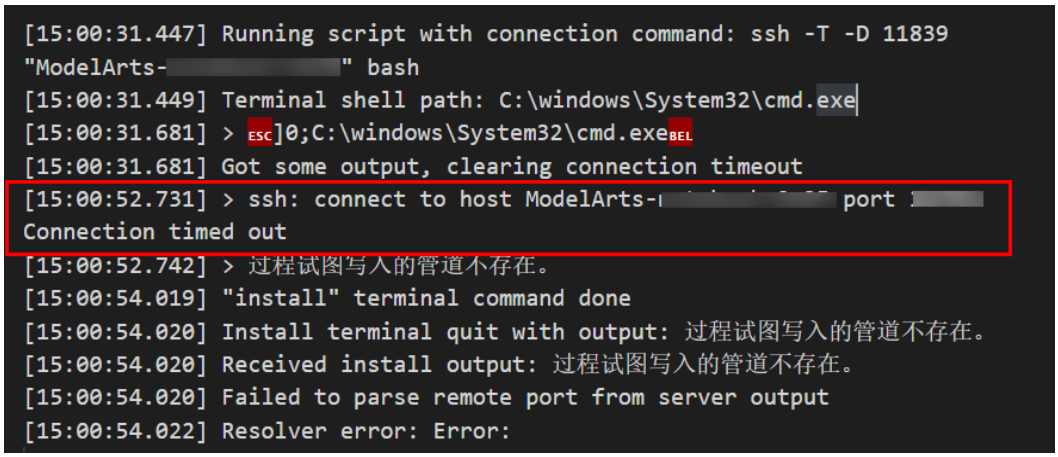
实例处于非运行状态。

## 解决方法

请前往ModelArts控制台查看实例是否处于运行状态，如果实例已停止，请执行启动操作，如果实例处于其他状态比如“错误”，请尝试先执行停止然后执行启动操作。待实例变为“运行中”后，再次执行远程连接。

# 5.9.13 报错"ssh: connect to host ModelArts-xxx port xxx: Connection timed out"如何解决?

## 问题现象




## 原因分析

原因分析一：实例配置的白名单IP与本地网络访问IP不符。

解决方法：请**修改白名单**为本地网络访问IP或者去掉白名单配置。

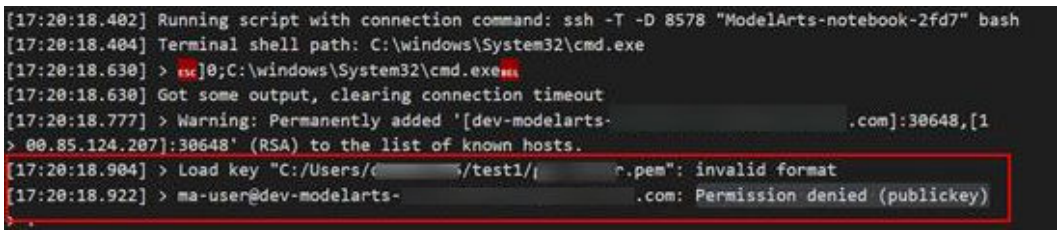
原因分析二：本地网络不通。

解决方法：检查本地网络以及网络限制。

## 5.9.14 报错“Load key "C:/Users/xx/test1/xxx.pem": invalid format”如何解决?

### 问题现象




### 原因分析

密钥文件内容不正确或格式不正确。

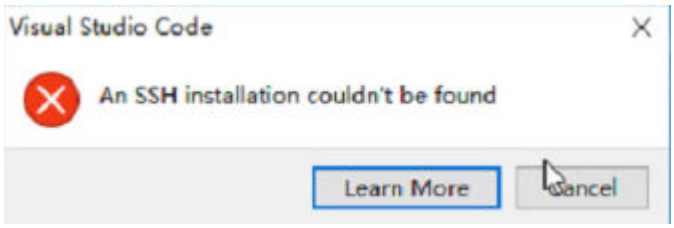
### 解决方法

请使用正确的密钥文件进行远程访问，如果本地没有正确的密钥文件或文件已损坏，可以尝试：

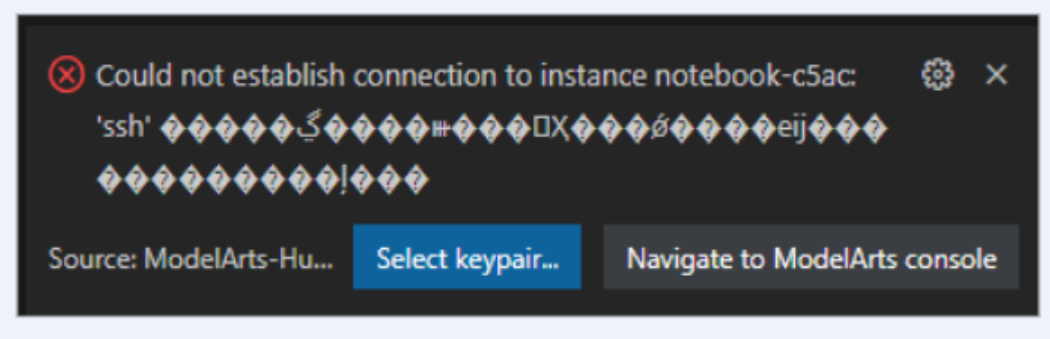
1. 登录控制台，搜索“数据加密服务DEW”，选择“密钥对管理 > 账号密钥对”页签，查看并下载正确的密钥文件。
2. 如果密钥不支持下载且已无法找到之前下载的密钥，建议创建新的开发环境实例并创建新的密钥文件。

## 5.9.15 报错“An SSH installation couldn't be found”或者“Could not establish connection to instance xxx: 'ssh' ...”如何解决?

### 问题现象




或




VS Code连接Notebook一直提示选择证书，且提示信息除标题外，都是乱码。选择证书后，如上图所示仍然没有反应且无法进行连接。

## 原因分析

当前环境未装OpenSSH或者OpenSSH未安装在默认路径下，详情请参考**VS Code文档**。

## 解决方法

- 如果当前环境未安装OpenSSH，请**下载并安装OpenSSH**。

  Installing a supported SSH client

  | OS | Instructions |
  | --- | --- |
  | Windows 10 1803+ / Server 2016/2019 1803+ | Install the Windows OpenSSH Client. |
  | Earlier Windows | Install Git for Windows. |
  | macOS | Comes pre-installed. |
  | Debian/Ubuntu | Run `sudo apt-get install openssh-client` |
  | RHEL / Fedora / CentOS | Run `sudo yum install openssh-clients` |

  VS Code will look for the `ssh` command in the PATH. Failing that, on Windows it will attempt to find `ssh.exe` in the default Git for Windows install path. You can also specifically tell VS Code where to find the SSH client by adding the `remote.SSH.path` property to `settings.json`.

  当通过“可选功能”未能成功安装时，请手动**下载OpenSSH安装包**，然后执行以下步骤：

  **步骤1** 下载zip包并解压放入“C:\Windows\System32”。

  **步骤2** 以管理员身份打开CMD，在“C:\Windows\System32\OpenSSH-xx”目录下，执行以下命令：

  ```
  powershell.exe -ExecutionPolicy Bypass -File install-sshd.ps1
  ```

  **步骤3** 添加环境变量：将“C:\Program Files\OpenSSH-xx”（路径中包含ssh可执行exe文件）添加到环境系统变量中。

  **步骤4** 重新打开CMD，并执行ssh，结果如下图即说明安装成功，如果还未装成功则执行**步骤5**和**步骤6**。

```
C:\windows\system32>ssh
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]
```

**步骤5** OpenSSH默认端口为22端口，开启防火墙22端口号，在CMD执行以下命令：

```
netsh advfirewall firewall add rule name=sshd dir=in action=allow protocol=TCP localport=22
```

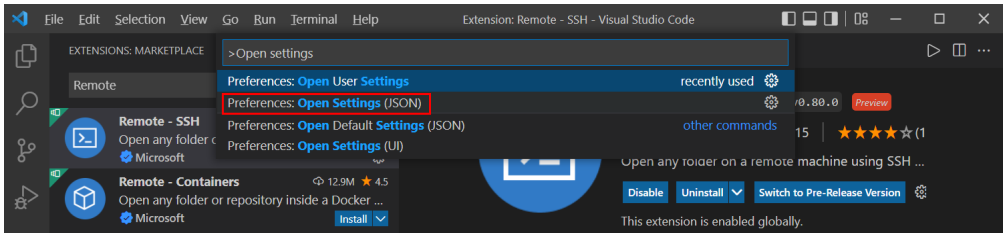
**步骤6** 启动OpenSSH服务，在CMD执行以下命令：

```
Start-Service sshd
```

**----结束**

- 若OpenSSH未安装在默认路径下，打开命令面板（Windows： Ctrl+Shift+P，macOS：Cmd+Shift+P），

  搜索“Open settings”。

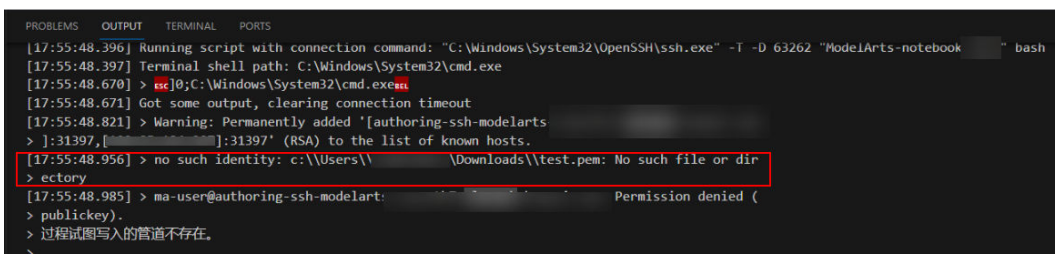



  然后将remote.SSH.path属性添加到settings.json中，例如："remote.SSH.path": "*本地OpenSSH的安装路径*"

```
{
    "extensions.autoCheckUpdates": false,
    "extensions.autoUpdate": false,
    "remote.SSH.remotePlatform": {
        "ModelArts-notebook-          ": "linux"
    },
    "remote.SSH.path": "D:/OpenSSH-Win64/ssh.exe"
}
```

# 5.9.16 报错“no such identity: C:/Users/xx /test.pem: No such file or directory”如何解决?

## 问题现象



## 原因分析

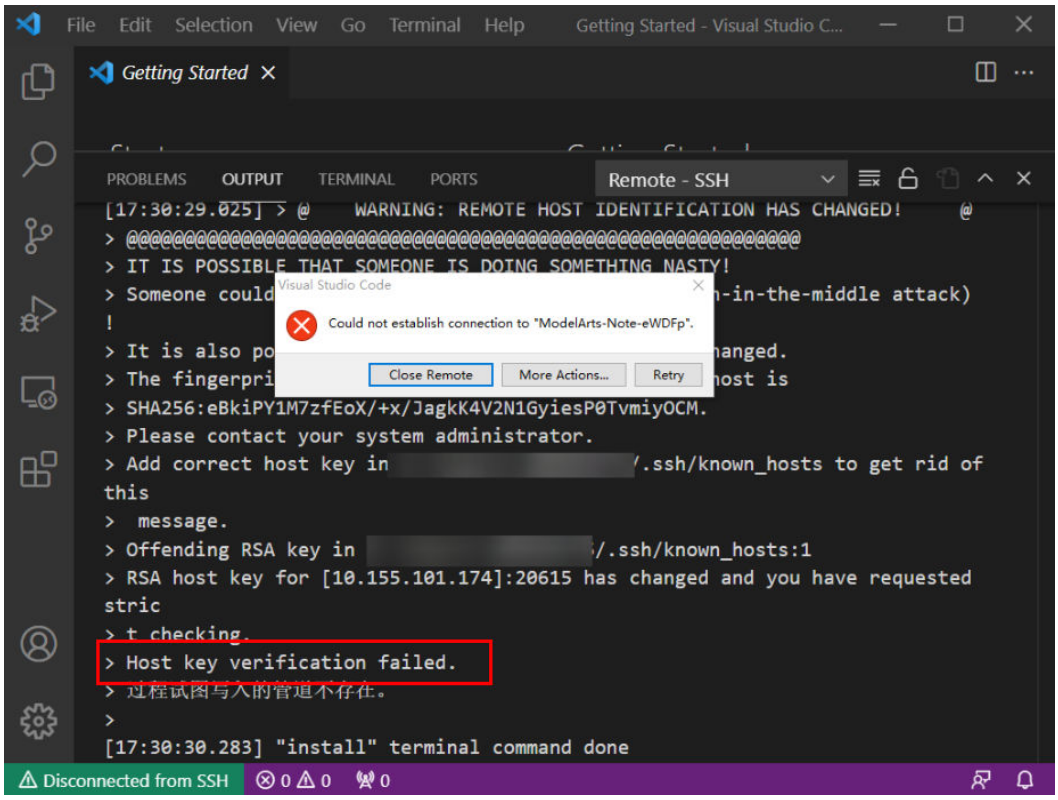
密钥文件不存在于该路径下，或者该路径下密钥文件名被修改。

## 解决方法

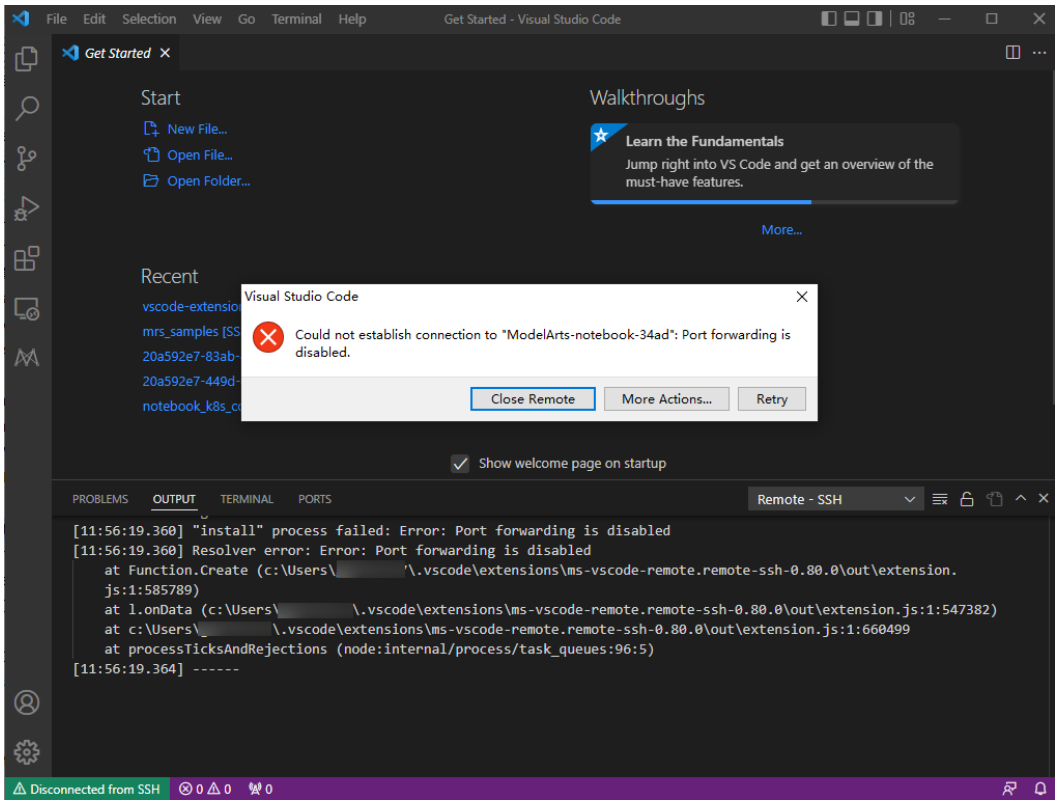
重新选择密钥路径。

# 5.9.17 报错“Host key verification failed.'或者'Port forwarding is disabled.”如何解决?

## 问题现象




或




## 原因分析

Notebook实例重新启动后，公钥发生变化，OpenSSH核对公钥发出警告。

## 解决方法

- 在VS Code中使用命令方式进行远程连接时，增加参数"-o StrictHostKeyChecking=no"

  ```
  ssh -tt -o StrictHostKeyChecking=no -i ${IdentityFile} ${User}@${HostName} -p ${Port}
  ```

  参数说明：

  - IdentityFile：本地密钥路径
  - User：用户名，例如：ma-user
  - HostName：IP地址
  - Port：端口号

- 在VS Code中手工配置远程连接时，在本地的ssh config文件中增加配置参数“StrictHostKeyChecking no”和“UserKnownHostsFile=/dev/null”

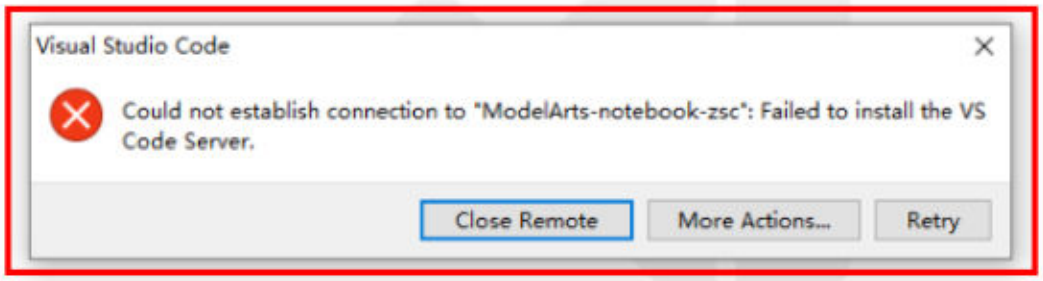
  ```
  Host xxx
    HostName x.x.x.x #IP地址
    Port 22522
    User ma-user
    IdentityFile C:/Users/my.pem
    StrictHostKeyChecking no
    UserKnownHostsFile=/dev/null
    ForwardAgent yes
  ```

提示：增加参数后SSH登录时会忽略known_hosts文件，有安全风险。

# 5.9.18 报错“Failed to install the VS Code Server.”或“tar: Error is not recoverable: exitng now.”如何解决?

## 问题现象




或

```
[17:53:24.382] > vscode-scp-done.flag                          100%    9     0.2KB/s   00:00
[17:53:24.756] > Found flag and server on host
[17:53:24.765] > d3aeabcaa9c5%%2%%
> tar --version:
[17:53:24.789] > tar (GNU tar) 1.30
> Copyright (C) 2017 Free Software Foundation, Inc.
> License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
> This is free software: you are free to change and redistribute it.
> There is NO WARRANTY, to the extent permitted by law.
>
> Written by John Gilmore and Jay Fenlason.
[17:53:24.796] > tar: This does not look like a tar archive
>
> gzip: stdin: unexpected end of file
> tar: Child returned status 1
> tar: Error is not recoverable: exiting now
[17:53:24.804] >
> ERROR: tar exited with non-0 exit code: 0
> Already attempted local download, failing
> d3aeabcaa9c5: start
> exitCode==37==
```

## 原因分析

可能为/home/ma-user/work磁盘空间不足。

## 解决方法

删除/home/ma-user/work路径下无用文件。

# 5.9.19 VS Code 连接远端 Notebook 时报错如“XHR failed”

## 原因分析

可能是所在环境的网络问题，请排查。

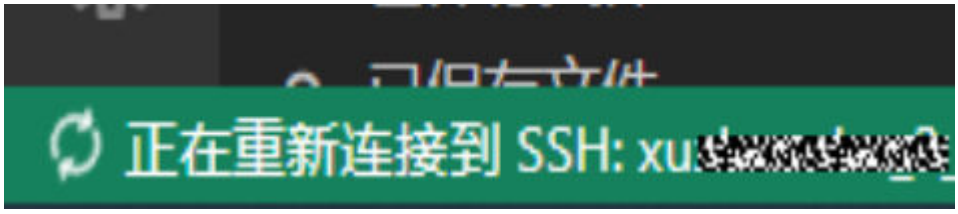
## 解决方法

参见**XHR failed常见排查思路**，进行排查。

# 5.9.20 VS Code 连接后长时间未操作，连接自动断开

## 问题现象

VS Code SSH连接后，长时间未操作，窗口未关闭，再次使用发现VS Code在重连环境，无弹窗报错。左下角显示如下图：




查看VS Code Remote-SSH日志发现，连接在大约2小时后断开了：

```
>
[21:32:39.136] Got some output, clearing connection timeout
[21:48:58.059] >
[21:49:12.060] >
[22:40:58.740] >
>
[23:32:49.341] > Connection reset by 139.159.152.36 port 32528
>
```

这会是正常的

这会断开了

## 原因分析

用户SSH交互操作停止后一段时间，防火墙对空闲链接进行了断开操作，SSH默认配置中不存在超时主动断连的动作，但是防火墙会关闭超时空闲连接（参考：http://bluebiu.com/blog/linux-ssh-session-alive.html），后台的实例运行是一直稳定的，重连即可再次连上。

## 解决方法

如果想保持长时间连接不断开，可以通过配置SSH定期发送通信消息，避免防火墙认为链路空闲而关闭。

- 客户端配置（用户可根据需要自行配置，不配置默认是不给服务端发心跳包），如图1，图2所示。

**图 5-21** 打开 VS Code ssh config 配置文件

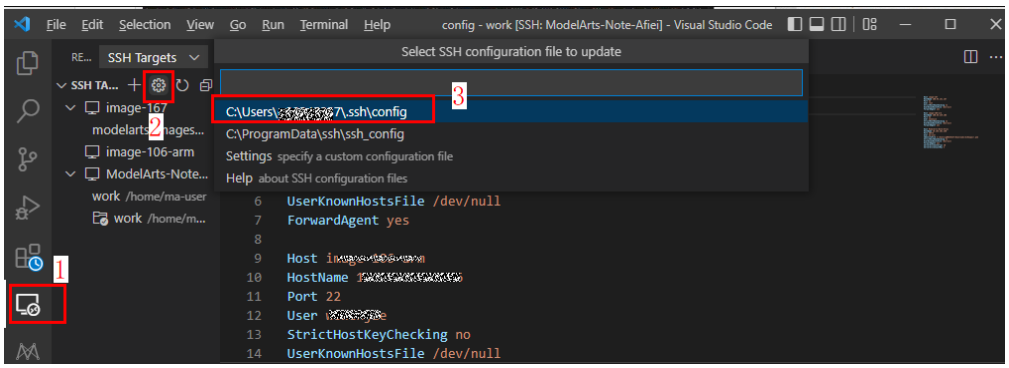
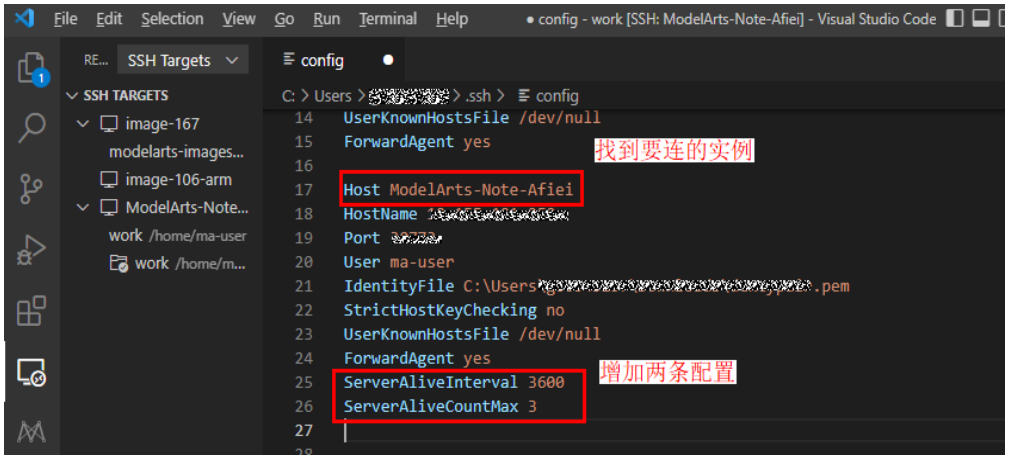


**图 5-22** 增加配置信息




配置信息示例如下：

```
Host ModelArts-xx
    ......
    ServerAliveInterval 3600  # 增加这个配置，单位是秒，每1h向服务端主动发个包
    ServerAliveCountMax 3  # 增加这个配置，3次发包均无响应会断开连接
```

比如防火墙配置是2小时空闲就关闭连接，那客户端配置ServerAliveInterval小于2小时（比如1小时），就可以避免防火墙将连接断开。

- 服务器端配置（Notebook当前已经配置，24h应该是长于防火墙的断连时间配置，该配置无需用户手工修改，写在这里仅是帮助理解ssh配置原理）配置文件路径：/home/ma-user/.ssh/etc/sshd_config

```
/modelarts/authoring(MindSpore) [ma-user work]$cat /home/ma-user/.ssh/etc/sshd_config |grep Client
ClientAliveInterval 1440m
ClientAliveCountMax 3
```

每24h向client端主动发个包，3次发包均无响应会断开连接

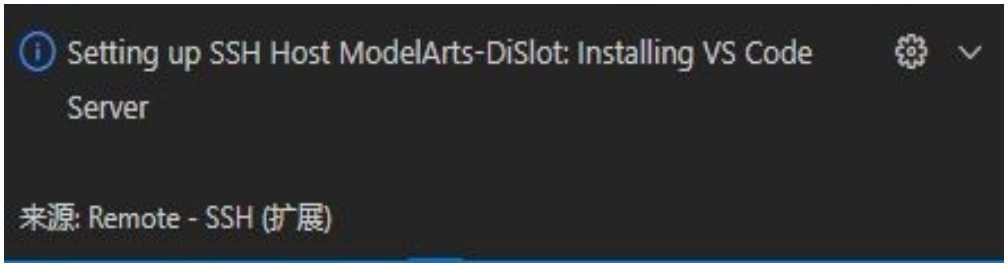
参考：https://unix.stackexchange.com/questions/3026/what-do-options-serveraliveinterval-and-clientaliveinterval-in-sshd-config-d

- 对于业务有影响的需要进行长链接保持的场景，尽量将日志写在单独的日志文件中，将脚本后台运行，例如：

```
nohup train.sh > output.log 2>&1 & tail -f output.log
```

# 5.9.21 VS Code 自动升级后，导致远程连接时间过长

## 问题现象




## 原因分析

由于VS Code自动升级，导致连接时需要重新下载新版vscode-server。

## 解决方法

禁止VS Code自动升级。单击左下角选择Settings项，搜索Update: Mode，将其设置为none。

**图 5-23** 打开 Settings

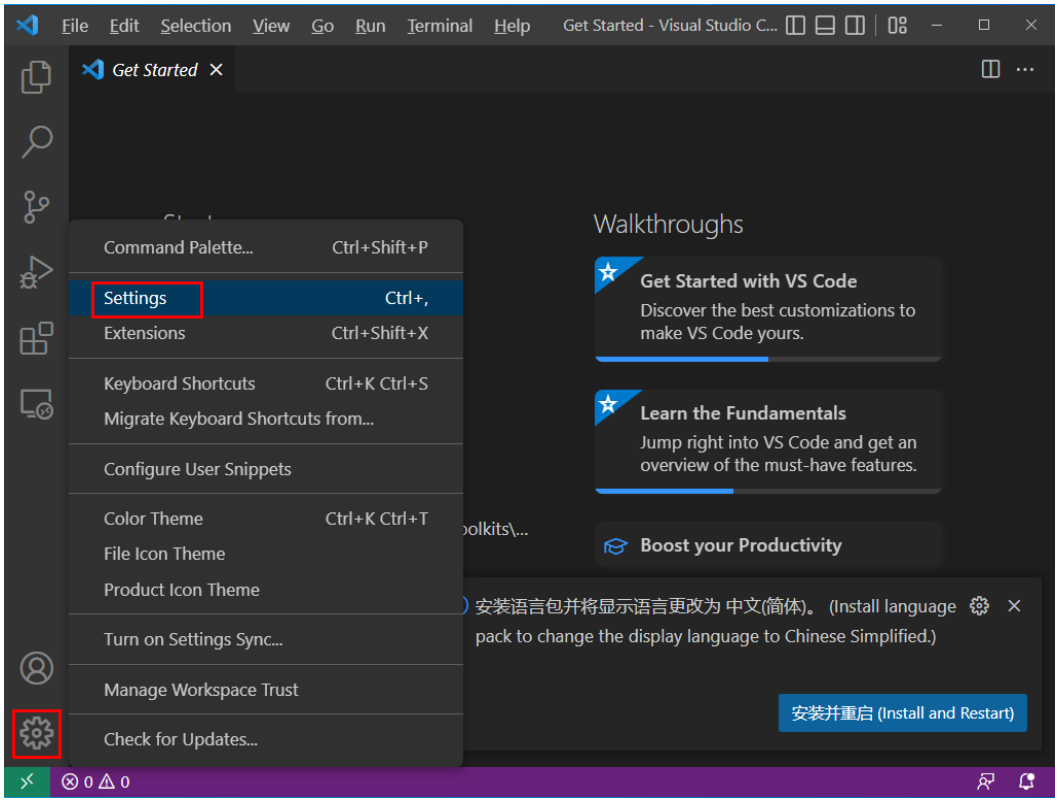
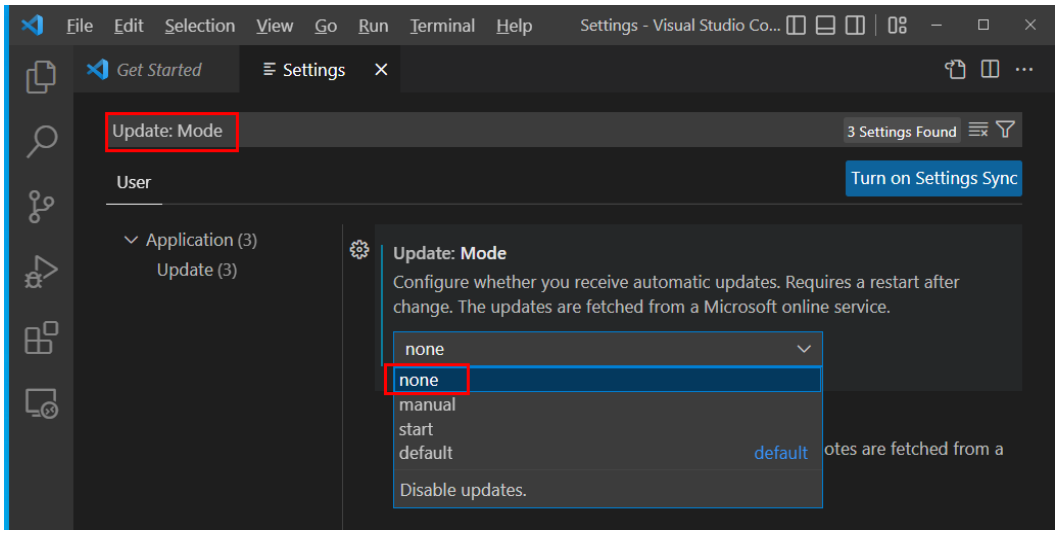



**图 5-24** 设置“Update: Mode”为“none”



# 5.9.22 使用 SSH 连接，报错“Connection reset”如何解决?

## 问题现象

```
C:\Users\      \.ssh>ssh -tt -o StrictHostKeyChecking=no -i KeyPair-      .pem ma-user@dev-modelarts
om -p 30
kex_exchange_identification: read: Connection reset
```

## 原因分析

可能是用户网络限制原因。比如部分企业网络的SSH是默认屏蔽的。

## 解决方法

用户重新进行申请SSH权限。

# 5.9.23 使用 MobaXterm 工具 SSH 连接 Notebook 后，经常断开或卡顿，如何解决?

## 问题现象

MobaXterm成功连接到开发环境后，过一段时间会自动断开。

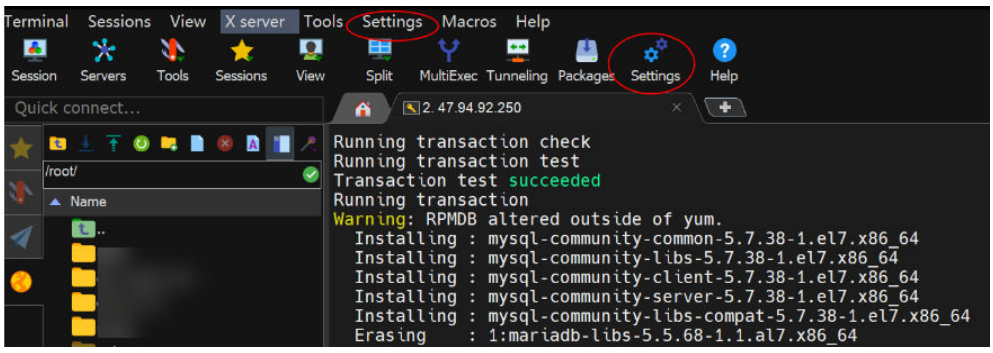
## 可能原因

配置MobaXterm工具时，没有勾选“SSH keepalive”或专业版MobaXterm工具的“Stop server after”时间设置太短。

## 解决方案

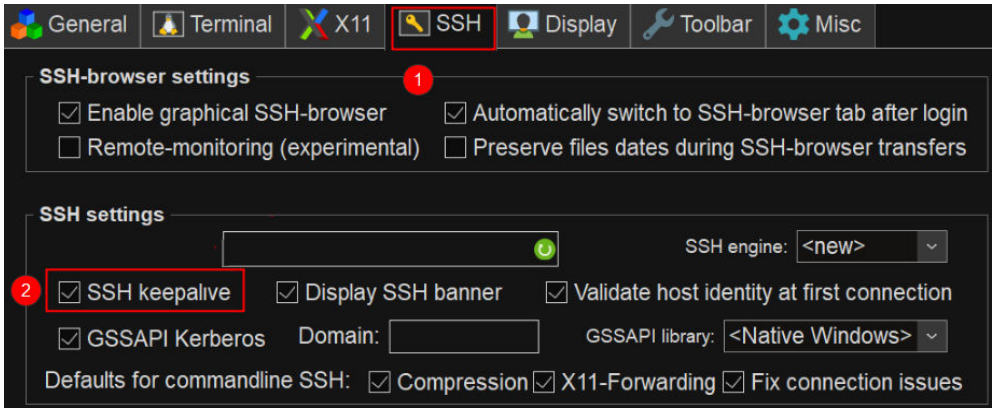
**步骤1** 打开MobaXterm，单击菜单栏“Settings”，如**图1 打开“Settings”**所示。

**图 5-25** 打开“Settings”




**步骤2** 在打开的“MobaXterm Configuration”配置页面，选择“SSH”选项卡，勾选“SSH keepalive”，如**图2 勾选“SSH keepalive”**所示。

**图 5-26** 勾选“SSH keepalive”

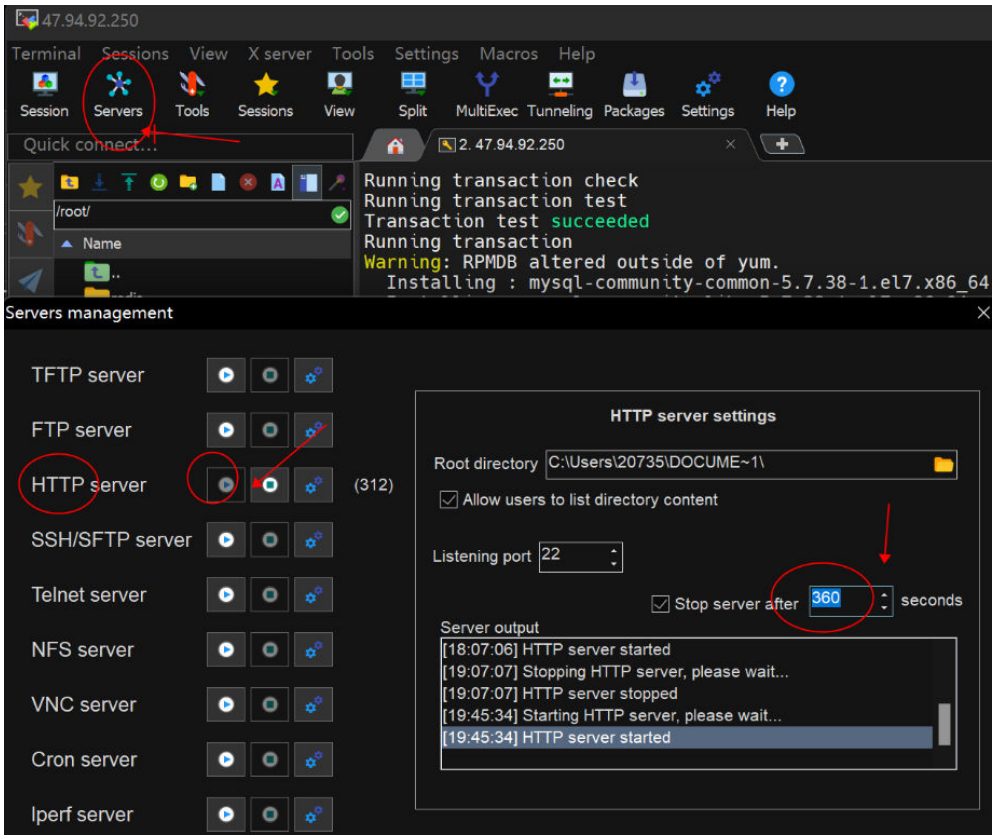



**说明**

如果使用的是专业版的MobaXterm工具，请执行**步骤3**。

**步骤3** 如果使用的是专业版的MobaXterm工具，请参考**图3 设置“Stop server after”**，此参数默认值为360s，将其设置为3600s或更大值。

**图 5-27** 设置“Stop server after”




**----结束**

## 5.9.24 VS Code 连接开发环境时报错 Missing GLIBC，Missing required dependencies

### 问题现象

VS Code连接开发环境时报错如下：

```
Warning: Missing GLIBC >= 2.28! from /lib/x86_64-linux-gnu/libc-2.27.so Error: Missing required dependencies. Please refer to our FAQ https://aka.ms/vscode-remote/faq/old-linux for additional information.
```

### 原因分析

该问题为用户使用VS Code 1.86版本软件导致的，需要用户使用较低版本的VS Code 。

### 解决方案

使用VS Code 1.85版本软件。下载链接：**https://code.visualstudio.com/updates/v1_85**。

## 5.9.25 使用 VSCode-huawei，报错：卸载了‘ms-vscode-remote.remot-sdh’，它被报告存在问题

### 问题现象

使用华为自研的VS Code软件时，报错“卸载了‘ms-vscode-remote.remot-sdh’，它被报告存在问题”。

### 原因分析

Remote - SSH只能在开源的VSCode软件中使用。

### 解决方案

推荐使用开源VS Code软件。

# 5.10 在 Notebook 中使用自定义镜像常见问题

## 5.10.1 不在同一个主账号下，如何使用他人的自定义镜像创建Notebook?

不是同一个主账号，用户A需要使用用户B的自定义镜像创建Notebook，此时需要用户B将此镜像共享给用户A，用户A将此共享镜像Pull下来注册后方可在Notebook中使用。详细操作如下：

**用户B的操作**：

1. 登录**容器镜像服务**控制台，进入“我的镜像”页面。
2. 单击需要共享的镜像名称，进入镜像详情页。

3. 在共享页签，单击“共享镜像”，在新窗口中输入共享的账号名称等，单击“确定”。

**用户A的操作**：

1. 登录**容器镜像服务**控制台，在“我的镜像>他人共享”页签下，查看用户B共享的镜像，单击镜像名称进入镜像详情。
2. 按照“Pull/Push指南”页签提供的操作方法，将用户B共享的镜像Pull下来，即作为自有镜像。


3. 进入ModelArts控制台，选择Pull下来的镜像进行镜像注册，注册成功后即可在Notebook界面使用此镜像。

# 5.11 更多功能咨询

## 5.11.1 在 Notebook 中，如何使用昇腾多卡进行调试?

昇腾多卡训练任务是多进程多卡模式，跑几卡需要起几个python进程。昇腾底层会读取环境变量：RANK_TABLE_FILE，开发环境已经设置，用户无需关注。比如跑八卡，可以如下片段代码：

```
export RANK_SIZE=8
current_exec_path=$(pwd)
echo 'start training'
for((i=0;i<=$RANK_SIZE-1;i++));
do
```

```
echo 'start rank '$i
mkdir ${current_exec_path}/device$i
cd ${current_exec_path}/device$i
echo $i
export RANK_ID=$i
dev=`expr $i + 0`
echo $dev
export DEVICE_ID=$dev
python train.py > train.log 2>&1 &
done
```

其中，train.py中设置环境变量DEVICE_ID：

```
devid = int(os.getenv('DEVICE_ID'))
 context.set_context(mode=context.GRAPH_MODE, device_target="Ascend", device_id=devid)
```

## 5.11.2 使用 Notebook 不同的资源规格，为什么训练速度差不多?

如果用户的代码中训练任务是单进程的，使用Notebook 8核64GB，72核512GB训练的速度是基本一致的，例如用户用的是2核4GB的资源，使用4核8GB，或者8核64GB效果是一样的。

如果用户的代码中训练任务是多进程的，使用Notebook 72核512GB训练速度要优于8核64GB。

## 5.11.3 使用 MoXing 时，如何进行增量训练?

在使用MoXing构建模型时，如果您对前一次训练结果不满意，可以在更改部分数据和标注信息后，进行增量训练。

### “mox.run”添加增量训练参数

在完成标注数据或数据集的修改后，您可以在“mox.run”中，修改“log_dir”参数，并新增“checkpoint_path”参数。其中“log_dir”参数建议设置为一个新的目录，“checkpoint_path”参数设置为上一次训练结果输出路径，如果是OBS目录，路径填写时建议使用“obs://”开头。

如果标注数据中的标签发生了变化，在运行“mox.run”前先执行**如果标签发生变化**的操作。

```
mox.run(input_fn=input_fn,
        model_fn=model_fn,
        optimizer_fn=optimizer_fn,
        run_mode=flags.run_mode,
        inter_mode=mox.ModeKeys.EVAL if use_eval_data else None,
        log_dir=log_dir,
        batch_size=batch_size_per_device,
        auto_batch=False,
        max_number_of_steps=max_number_of_steps,
        log_every_n_steps=flags.log_every_n_steps,
        save_summary_steps=save_summary_steps,
        save_model_secs=save_model_secs,
        checkpoint_path=flags.checkpoint_url,
        export_model=mox.ExportKeys.TF_SERVING)
```

### 如果标签发生变化

当数据集中的标签发生变化时，需要执行如下语句。此语句需在“mox.run”之前运行。

语句中的“logits”，表示根据不同网络中分类层权重的变量名，配置不同的参数。此处填写其对应的关键字。

```
mox.set_flag('checkpoint_exclude_patterns', 'logits')
```

如果使用的是MoXing内置网络，其对应的关键字需使用如下API获取。此示例将打印Resnet_v1_50的关键字，为“logits”。

```
import moxing.tensorflow as mox

model_meta = mox.get_model_meta(mox.NetworkKeys.RESNET_V1_50)
logits_pattern = model_meta.default_logits_pattern
print(logits_pattern)
```

您也可以通过如下接口，获取MoXing支持的网络名称列表。

```
import moxing.tensorflow as mox
print(help(mox.NetworkKeys))
```

打印出来的示例如下所示：

```
Help on class NetworkKeys in module
moxing.tensorflow.nets.nets_factory:

class NetworkKeys(builtins.object)
 |  Data descriptors defined here:
 |
 |  __dict__
 |      dictionary for instance variables (if defined)
 |
 |  __weakref__
 |      list of weak references to the object (if defined)
 |
 |  ----------------------------------------------------------------------
 |  Data and other attributes defined here:
 |
 |  ALEXNET_V2 = 'alexnet_v2'
 |
 |  CIFARNET = 'cifarnet'
 |
 |  INCEPTION_RESNET_V2 = 'inception_resnet_v2'
 |
 |  INCEPTION_V1 = 'inception_v1'
 |
 |  INCEPTION_V2 = 'inception_v2'
 |
 |  INCEPTION_V3 = 'inception_v3'
 |
 |  INCEPTION_V4 = 'inception_v4'
 |
 |  LENET = 'lenet'
 |
 |  MOBILENET_V1 = 'mobilenet_v1'
 |
 |  MOBILENET_V1_025 = 'mobilenet_v1_025'
 |
 |  MOBILENET_V1_050 = 'mobilenet_v1_050'
 |
 |  MOBILENET_V1_075 = 'mobilenet_v1_075'
 |
 |  MOBILENET_V2 = 'mobilenet_v2'
 |
 |  MOBILENET_V2_035 = 'mobilenet_v2_035'
 |
 |  MOBILENET_V2_140 = 'mobilenet_v2_140'
 |
 |  NASNET_CIFAR = 'nasnet_cifar'
 |
 |  NASNET_LARGE = 'nasnet_large'
```

```
NASNET_MOBILE = 'nasnet_mobile'
OVERFEAT = 'overfeat'
PNASNET_LARGE = 'pnasnet_large'
PNASNET_MOBILE = 'pnasnet_mobile'
PVANET = 'pvanet'
RESNET_V1_101 = 'resnet_v1_101'
RESNET_V1_110 = 'resnet_v1_110'
RESNET_V1_152 = 'resnet_v1_152'
RESNET_V1_18 = 'resnet_v1_18'
RESNET_V1_20 = 'resnet_v1_20'
RESNET_V1_200 = 'resnet_v1_200'
RESNET_V1_50 = 'resnet_v1_50'
RESNET_V1_50_8K = 'resnet_v1_50_8k'
RESNET_V1_50_MOX = 'resnet_v1_50_mox'
RESNET_V1_50_OCT = 'resnet_v1_50_oct'
RESNET_V2_101 = 'resnet_v2_101'
RESNET_V2_152 = 'resnet_v2_152'
RESNET_V2_200 = 'resnet_v2_200'
RESNET_V2_50 = 'resnet_v2_50'
RESNEXT_B_101 = 'resnext_b_101'
RESNEXT_B_50 = 'resnext_b_50'
RESNEXT_C_101 = 'resnext_c_101'
RESNEXT_C_50 = 'resnext_c_50'
VGG_16 = 'vgg_16'
VGG_16_BN = 'vgg_16_bn'
VGG_19 = 'vgg_19'
VGG_19_BN = 'vgg_19_bn'
VGG_A = 'vgg_a'
VGG_A_BN = 'vgg_a_bn'
XCEPTION_41 = 'xception_41'
XCEPTION_65 = 'xception_65'
XCEPTION_71 = 'xception_71'
```

## 5.11.4 在 Notebook 中如何查看 GPU 使用情况

创建Notebook时，当您选择的类型为GPU时，查看GPU使用情况具体操作如下：

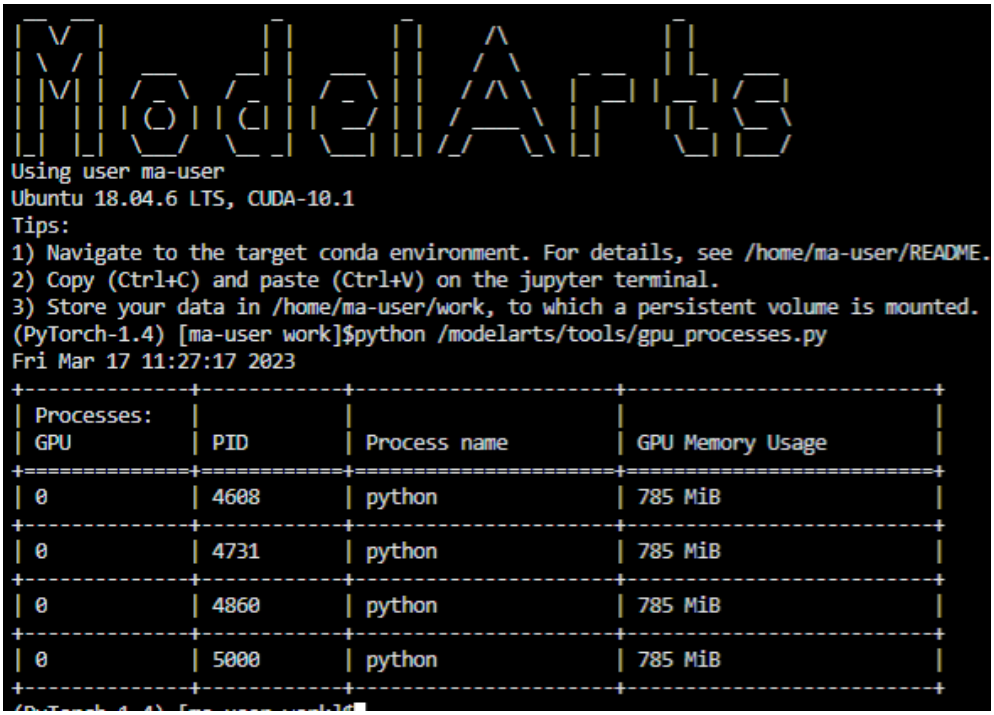
  

1. 登录ModelArts管理控制台，选择“开发空间>Notebook”。
2. 在Notebook列表中，单击目标Notebook“操作”列的“打开”，进入“Jupyter”开发页面。
3. 在Jupyter页面的“Files”页签下，单击“New”，然后选择“Terminal”，进入到Terminal界面。
4. 执行如下命令查看GPU使用情况。
   ```
   nvidia-smi
   ```
5. 查看当前Notebook实例中有哪些进程使用GPU。

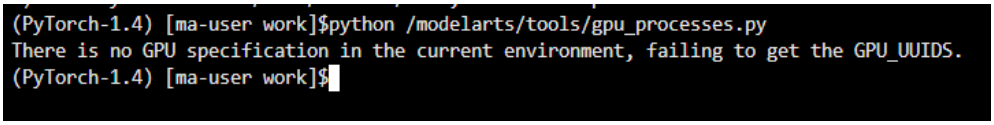
   方法一：

   ```
   python /modelarts/tools/gpu_processes.py
   ```

   如果当前进程使用GPU

   如果当前没有进程使用GPU

   方法二：

   打开文件“/resource_info/gpu_usage.json”，可以看到有哪些进程在使用GPU。

```
{
    <notebook name>: {
        <GPU0 UUID>: [
            {
                "pid": 2263,
                "processName": "python",
                "gpuMemoryUsage": "4935Mi"
            },
            {...}
        ]
        <GPU1 UUID>: [...]
    }
}
```

如果当前没有进程使用GPU，该文件可能不存在或为空。

## 5.11.5 如何在代码中打印 GPU 使用信息

用户可通过shell命令或python命令查询GPU使用信息。

### 使用 shell 命令

1. 执行nvidia-smi命令。

   依赖CUDA nvcc

   ```
   watch -n 1 nvidia-smi
   ```

   ```
   Every 1.0s: nvidia-smi

   Mon Oct 25 15:20:11 2021
   +-----------------------------------------------------------------------------+
   | NVIDIA-SMI 440.33.01    Driver Version: 440.33.01    CUDA Version: 10.2     |
   |-------------------------------+----------------------+----------------------+
   | GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
   | Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
   |===============================+======================+======================|
   |   0  Tesla V100-SXM2...  On   | 00000000:5F:00.0 Off |                    0 |
   | N/A   31C    P0    43W / 300W |      0MiB / 32510MiB |      0%      Default |
   +-------------------------------+----------------------+----------------------+
   |   1  Tesla V100-SXM2...  On   | 00000000:B5:00.0 Off |                    0 |
   | N/A   34C    P0    44W / 300W |      0MiB / 32510MiB |      0%      Default |
   +-------------------------------+----------------------+----------------------+

   +-----------------------------------------------------------------------------+
   | Processes:                                                       GPU Memory |
   |  GPU       PID   Type   Process name                             Usage      |
   |=============================================================================|
   |  No running processes found                                                 |
   +-----------------------------------------------------------------------------+
   ```

2. 执行gpustat命令。

   ```
   pip install gpustat
   gpustat -cp -i
   ```

   ```
   notebook-6a654129-698e-4635-b6be-67aedbdd4c54  Mon Oct 25 15:19:11 2021  440.33.01
   [0] Tesla V100-SXM2-32GB | 31'C,   0 % |     0 / 32510 MB |
   [1] Tesla V100-SXM2-32GB | 34'C,   0 % |     0 / 32510 MB |
   ```

   使用Ctrl+C可以退出。

## 使用 python 命令

1. 执行nvidia-ml-py3命令（常用）。

```
!pip install nvidia-ml-py3
import nvidia_smi
nvidia_smi.nvmlInit()
deviceCount = nvidia_smi.nvmlDeviceGetCount()
for i in range(deviceCount):
    handle = nvidia_smi.nvmlDeviceGetHandleByIndex(i)
    util = nvidia_smi.nvmlDeviceGetUtilizationRates(handle)
    mem = nvidia_smi.nvmlDeviceGetMemoryInfo(handle)
    print(f"|Device {i}| Mem Free: {mem.free/1024**2:5.2f}MB / {mem.total/1024**2:5.2f}MB | gpu-util:
{util.gpu:3.1%} | gpu-mem: {util.memory:3.1%} |")
```

```
Output:
|Device 0| Mem Free: 32510.44MB / 32510.50MB | gpu-util: 0.0% | gpu-mem: 0.0% |
|Device 1| Mem Free: 32510.44MB / 32510.50MB | gpu-util: 0.0% | gpu-mem: 0.0% |
```

2. 执行nvidia_smi + wapper + prettytable命令。

   用户可以将GPU信息显示操作看作一个装饰器，在模型训练过程中就可以实时的显示GPU状态信息。

```
def gputil_decorator(func):
    def wrapper(*args, **kwargs):
        import nvidia_smi
        import prettytable as pt

        try:
            table = pt.PrettyTable(['Devices','Mem Free','GPU-util','GPU-mem'])
            nvidia_smi.nvmlInit()
            deviceCount = nvidia_smi.nvmlDeviceGetCount()
            for i in range(deviceCount):
                handle = nvidia_smi.nvmlDeviceGetHandleByIndex(i)
                res = nvidia_smi.nvmlDeviceGetUtilizationRates(handle)
                mem = nvidia_smi.nvmlDeviceGetMemoryInfo(handle)
                table.add_row([i, f"{mem.free/1024**2:5.2f}MB/{mem.total/1024**2:5.2f}MB",
f"{res.gpu:3.1%}", f"{res.memory:3.1%}"])

        except nvidia_smi.NVMLError as error:
            print(error)

        print(table)
        return func(*args, **kwargs)
    return wrapper
```

```
Output:

+---------+-----------------------+----------+----------+
| Devices |        Mem Free       | GPU-util | GPU-mem |
+---------+-----------------------+----------+----------+
|    0    | 32510.44MB/32510.50MB |   0.0%   |   0.0%   |
|    1    | 32510.44MB/32510.50MB |   0.0%   |   0.0%   |
+---------+-----------------------+----------+----------+
```

3. 执行pynvml命令。

   nvidia-ml-py3可以直接查询nvml c-lib库，而无需通过nvidia-smi。因此，这个模块比nvidia-smi周围的包装器快得多。

```
from pynvml import *
nvmlInit()
handle = nvmlDeviceGetHandleByIndex(0)
info = nvmlDeviceGetMemoryInfo(handle)
print("Total memory:", info.total)
print("Free memory:", info.free)
print("Used memory:", info.used)
```

```
Output:
Total memory: 34089730048
Free memory: 34089664512
Used memory: 65536
```

4. 执行gputil命令。

```
!pip install gputil
import GPUtil as GPU
GPU.showUtilization()
```

```
Output:

| ID | GPU | MEM |
------------------
|  0 |  0% | 25% |
|  1 |  0% |  0% |
```

```
import GPUtil as GPU
GPUs = GPU.getGPUs()
for gpu in GPUs:
    print("GPU RAM Free: {0:.0f}MB | Used: {1:.0f}MB | Util {2:3.0f}% | Total
{3:.0f}MB".format(gpu.memoryFree, gpu.memoryUsed, gpu.memoryUtil*100, gpu.memoryTotal))
```

```
Output:
GPU RAM Free: 32510MB | Used: 0MB | Util   0% | Total 32510MB
GPU RAM Free: 32510MB | Used: 0MB | Util   0% | Total 32510MB
```

**注：用户在使用pytorch/tensorflow等深度学习框架时也可以使用框架自带的api进行查询。**

## 5.11.6 Ascend 上如何查看实时性能指标?

Ascend芯片上查看实时性能指标：npu-smi info，类似GPU的nvidia-smi。

## 5.11.7 不启用自动停止，系统会自动停掉 Notebook 实例吗？会删除 Notebook 实例吗?

针对此问题，需要根据选择的不同资源规格进行说明。

- 如果使用免费规格，Notebook实例将在运行1小时后，自动停止。如果72小时内没有再次启动，会释放资源，即删除此Notebook实例。因此使用免费规格时，关注运行时间并注意文件备份。
- 如果使用收费的公共资源池，未启用自动停止功能时，Notebook实例不会自动停止，且不会被删除。
- 如果使用专属资源池，Notebook实例也不会自动停止。但是如果专属资源池被删除，则会导致此Notebook实例不可用。

## 5.11.8 JupyterLab 目录的文件、Terminal 的文件和 OBS 的文件之间的关系

- JupyterLab目录的文件与Terminal中work目录下的文件相同。即用户在Notebook中新建的，或者是从OBS目录中同步的文件。
- 挂载OBS存储的Notebook，JupyterLab目录的文件可以与OBS的文件进行同步，使用JupyterLab文件上传下载功能。Terminal的文件与JupyterLab目录的文件相同。

- 挂载EVS存储的Notebook，JupyterLab目录的文件可使用Moxing接口或SDK接口，读取OBS中的文件。Terminal的文件与JupyterLab目录的文件相同。

## 5.11.9 ModelArts 中创建的数据集，如何在 Notebook 中使用

ModelArts上创建的数据集存放在OBS中，可以将OBS中的数据下载到Notebook中使用。

Notebook中读取OBS数据方式请参见**如何在Notebook中上传下载OBS文件？**。

## 5.11.10 pip 介绍及常用命令

pip是通用的python包的管理工具。它提供了对Python包的查找、下载、安装和卸载的功能。

pip常用命令如下：

```
pip --help#获取帮助
pip install SomePackage==XXXX #指定版本安装
pip install SomePackage #最新版本安装
pip uninstall SomePackage #卸载软件版本
```

其他命令请使用**pip --help**命令查询。

## 5.11.11 开发环境中不同 Notebook 规格资源“/cache”目录的大小

创建Notebook时，可以根据业务数据量的大小选择资源。

ModelArts会挂载硬盘至“/cache”目录，用户可以使用此目录来储存临时文件。“/cache”与代码目录共用资源，不同资源规格有不同的容量。

映射规则：当前不支持CPU配置cache盘；GPU与昇腾资源为单卡时，cache目录保持500G大小限制；除单卡外，cache盘大小与卡数有关，计算方式为卡数*500G，上限为3T。详细**表5-1**所示。

**表 5-1** 不同 Notebook 规格资源“/cache”目录的大小

| 规格类别 | cache盘大小 |
| --- | --- |
| GPU-0.25卡 | 500G*0.25 |
| GPU-0.5卡 | 500G*0.5 |
| GPU-单卡 | 500G |
| GPU-双卡 | 500G*2 |
| GPU-四卡 | 500G*4 |
| GPU-八卡 | 3T |
| 昇腾-单卡 | 500G |
| 昇腾-双卡 | 500G*2 |
| 昇腾-四卡 | 500G*4 |
| 昇腾-八卡 | 3T |

| 规格类别 | cache盘大小 |
| --- | --- |
| CPU | -- |

## 5.11.12 开发环境如何实现 IAM 用户隔离?

开发环境如果需要实现IAM用户隔离，即多个IAM用户之间无法查看、修改和删除他人创建的Notebook。

目前有两种方案：

- 方案一：删除modelarts:notebook:listAllNotebooks细粒度权限。
- 方案二：使用工作空间功能：目前工作空间功能是“受邀开通”状态，作为企业用户您可以通过您对口的技术支持申请开通。

## 5.11.13 资源超分对 Notebook 实例有什么影响?

Notebook超分，是指一个节点中CPU、内存共享的场景。为了充分利用资源，在专属池中存在超分情况。

举例：一个专属池中有1个8U64G的CPU节点，如创建2U8G规格的Notebook，因为超分最多可启动 8U/(2U*0.6)= 6.67个Notebook实例。这里的0.6就是超分比率。即启动该Notebook实例最少需要1.2U的CPU，运行Notebook时最大使用到2U的资源；内存同理，最少需要4.8G的内存，运行时最大使用到8U的内存。

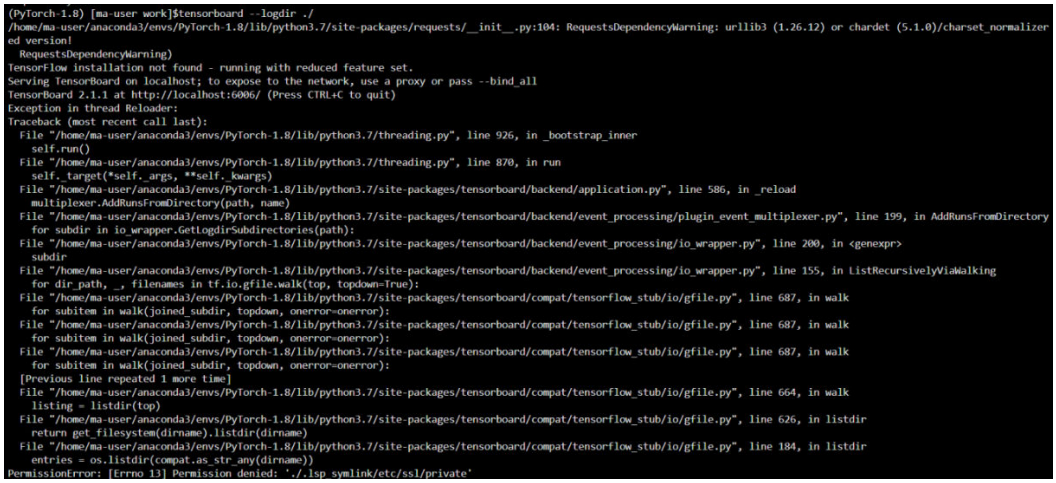
超分情况下会存在实例终止的风险。如1个8U的节点上同时启动了6个2U的实例，如果其中一个实例CPU使用增大到超过节点的上限（ 8U ）时，k8S会将使用资源最多的实例终止掉。

因此超分会带来实例重启的风险，请不要超分使用。

## 5.11.14 在 Notebook 中使用 tensorboard 命令打开日志文件报错 Permission denied

### 问题现象

在Notebook的Terminal中执行**tensorboard --logdir ./**命令，报错[Errno 13] Permission denied……。



## 原因分析

当前目录下包含没有权限的文件。

## 解决方法

建议用户新建一个文件夹（例如：tb_logs），将tensorboard的日志文件（例如：tb.events）放到新建的文件夹下，然后执行tensorboard命令。示例命令如下：

```
mkdir -p ./tb_logs
mv tb.events ./tb_logs
tensorboard --logdir ./tb_logs
```

# 6 训练作业

## 6.1 功能咨询

### 6.1.1 是否支持图像分割任务的训练?

支持。您可以使用以下三种方式实现图像分割任务的训练。

- 您可以在AI Gallery订阅相关图像分割任务算法，并**使用订阅算法完成训练**。
- 如果您在本地使用ModelArts支持的常用框架完成了训练脚本，可以**使用自定义脚本创建训练作业**。
- 如果您在本地开发的算法不是基于常用框架，您可以选择**使用自定义镜像创建训练作业**。

### 6.1.2 本地导入的算法有哪些格式要求?

ModelArts支持导入本地开发的算法，格式要求如下：

- 编程语言不限。
- 启动文件必须选择以“.py”结尾的文件。
- 文件数（含文件、文件夹数量）不超过1024个。
- 文件总大小不超过5GB。

### 6.1.3 欠拟合的解决方法有哪些?

1. 模型复杂化。
   - 对同一个算法复杂化。例如回归模型添加更多的高次项，增加决策树的深度，增加神经网络的隐藏层数和隐藏单元数等。
   - 弃用原来的算法，使用一个更加复杂的算法或模型。例如用神经网络来替代线性回归，用随机森林来代替决策树。
2. 增加更多的特征，使输入数据具有更强的表达能力。
   - 特征挖掘十分重要，尤其是具有强表达能力的特征，可以抵过大量的弱表达能力的特征。
   - 特征的数量并非重点，质量才是，总之强表达能力的特征最重要。
   - 能否挖掘出强表达能力的特征，还在于对数据本身以及具体应用场景的深刻理解，这依赖于经验。

3. 调整参数和超参数。
   - 神经网络中：学习率、学习衰减率、隐藏层数、隐藏层的单元数、Adam优化算法中的β1和β2参数、batch_size数值等。
   - 其他算法中：随机森林的树数量，k-means中的cluster数，正则化参数λ等。
4. 增加训练数据作用不大。

   欠拟合一般是因为模型的学习能力不足，一味地增加数据，训练效果并不明显。
5. 降低正则化约束。

   正则化约束是为了防止模型过拟合，如果模型压根不存在过拟合而是欠拟合了，那么就考虑是否降低正则化参数λ或者直接去除正则化项。

## 6.1.4 旧版训练迁移至新版训练需要注意哪些问题?

新版训练和旧版训练的差异主要体现在以下3点:

- **新旧版创建训练作业方式差异**
- **新旧版训练代码适配的差异**
- **新旧版训练预置引擎差异**

### 新旧版创建训练作业方式差异

- 旧版训练支持使用“算法管理”（包含已保存的算法和订阅的算法）、“常用框架”、“自定义”（即自定义镜像）方式创建训练作业。
- 新版训练支持使用“自定义算法”、“我的算法”、“我的订阅”方式来创建训练作业。

新版训练的创建方式有了更明确的类别划分，选择方式和旧版训练存在区别。

- 旧版中使用“算法管理”中已保存的算法创建训练作业的用户，可以在新版训练中使用“我的算法”创建训练作业。
- 旧版中使用“算法管理”中订阅的算法创建训练作业的用户，可以在新版训练中使用“我的订阅”创建训练作业。
- 旧版中使用“常用框架”创建训练作业的用户，可以在新版训练中使用“自定义算法”创建训练作业（启动方式选择“预置框架”）。
- 旧版中使用“自定义”（即自定义镜像）创建训练作业的用户，可以在新版训练中使用“自定义算法”创建训练作业（启动方式选择“自定义”）。

### 新旧版训练代码适配的差异

旧版训练中，用户需要在输入输出数据上做如下配置：

```
#解析命令行参数
import argparse
parser = argparse.ArgumentParser(description='MindSpore Lenet Example')
parser.add_argument('--data_url', type=str, default="./Data",
                    help='path where the dataset is saved')
parser.add_argument('--train_url', type=str, default="./Model", help='if is test, must provide\
                    path where the trained ckpt file')
args = parser.parse_args()
...
#下载数据参数至容器本地，在代码中使用local_data_path代表训练输入位置
mox.file.copy_parallel(args.data_url, local_data_path)
...
#上传容器本地数据至obs路径
mox.file.copy_parallel(local_output_path, args.train_url)
```

新版训练中，用户配置输入输出数据，无需书写下载数据的代码，在代码中把arg.data_url和arg.train_url当做本地路径即可，详情参考**开发自定义脚本**。

```
#解析命令行参数
import argparse
parser = argparse.ArgumentParser(description='MindSpore Lenet Example')
parser.add_argument('--data_url', type=str, default="./Data",
                    help='path where the dataset is saved')
parser.add_argument('--train_url', type=str, default="./Model", help='if is test, must provide\
                    path where the trained ckpt file')
args = parser.parse_args()
...
# 下载的代码无需设置，后续涉及训练数据和输出路径数据使用data_url和train_url即可
#下载数据参数至容器本地，在代码中使用local_data_path代表训练输入位置
#mox.file.copy_parallel(args.data_url, local_data_path)
...
#上传容器本地数据至obs路径
#mox.file.copy_parallel(local_output_path, args.train_url)
```

## 新旧版训练预置引擎差异

- 新版的预置训练引擎默认安装Moxing2.0.0及以上版本。
- 新版的预置训练引擎统一使用了Python3.7及以上版本。
- 新版镜像修改了默认的HOME目录，由“/home/work”变为“/home/ma-user”，请注意识别训练代码中是否有“/home/work”的硬编码。
- 提供预置引擎类型有差异。新版的预置引擎在常用的训练引擎上进行了升级。

  如果您需要使用旧版训练引擎，单击显示旧版引擎即可选择旧版引擎。新旧版支持的预置引擎差异请参考**表6-1**。详细的训练引擎版本说明请参考**新版训练和旧版训练分别支持的AI引擎**。

**表 6-1** 新旧版预置引擎差异

| 工作环境 | 预置训练I引擎与版本 | 旧版训练 | 新版训练 |
| --- | --- | --- | --- |
| **TensorFlow** | Tensorflow-1.8.0 | √ | x |
| | Tensorflow-1.13.1 | √ | 后续版本支持 |
| | Tensorflow-2.1.0 | √ | √ |
| **MXNet** | MXNet-1.2.1 | √ | x |
| **Caffe** | Caffe-1.0.0 | √ | x |
| **Spark_MLlib** | Spark-2.3.2 | √ | x |
| **Ray** | RAY-0.7.4 | √ | x |
| **XGBoost-Sklearn** | XGBoost-0.80-Sklearn-0.18.1 | √ | x |
| **PyTorch** | PyTorch-1.0.0 | √ | x |
| | PyTorch-1.3.0 | √ | x |
| | PyTorch-1.4.0 | √ | x |

| 工作环境 | 预置训练引擎与版本 | 旧版训练 | 新版训练 |
| --- | --- | --- | --- |
| | PyTorch-1.8.0 | x | √ |
| **Ascend-Powered-Engine** | Mindspore-1.3.0 | √ | x |
| | Mindspore-1.7.0 | x | √ |
| | Tensorflow-1.15 | √ | √ |
| **MPI** | MindSpore-1.3.0 | x | √ |
| **Horovod** | horovod_0.20.0-tensorflow_2.1.0 | x | √ |
| | horovod_0.22.1-pytorch_1.8.0 | x | √ |
| **MindSpore-GPU** | MindSpore-1.1.0 | √ | x |
| | MindSpore-1.2.0 | √ | x |

## 6.1.5 ModelArts 训练好后的模型如何获取?

使用自动学习产生的模型只能在ModelArts上部署上线，无法下载至本地使用。

使用自定义算法或者订阅算法训练生成的模型，会存储至用户指定的OBS路径中，供用户下载。

## 6.1.6 AI 引擎 Scikit_Learn0.18.1 的运行环境怎么设置?

在ModelArts的算法管理页面，创建算法时勾选“显示旧版镜像”，选择XGBoost-Sklearn引擎即可。

ModelArts创建算法操作请参见**创建算法**。

ModelArts创建训练作业操作请参见**创建训练作业**。

## 6.1.7 TPE 算法优化的超参数必须是分类特征（categorical features）吗

对于优化的超参数类型，TPE算法本身是没有限制的，但出于面对普通用户节省资源的目的，ModelArts在前端限制了TPE的超参数必须是float，如果想离散型和连续型参数混用的话，可以调用rest接口。

## 6.1.8 模型可视化作业中各参数的意义?

可视化作业通过TensorBoard提供能力，TensorBoard功能介绍请参见**TensorBoard官网资料**。

## 6.1.9 如何在 ModelArts 上获得 RANK_TABLE_FILE 进行分布式训练?

ModelArts会帮用户生成RANK_TABLE_FILE文件，可通过环境变量查看文件位置。

- 在Notebook中打开terminal，可以运行如下命令查看RANK_TABLE_FILE：
  ```
  env | grep RANK
  ```
- 在训练作业中，您可以在训练启动脚本的首行加入如下代码，把RANK_TABLE_FILE的值打印出来：
  ```
  os.system('env | grep RANK')
  ```

## 6.1.10 如何查询自定义镜像的 cuda 和 cudnn 版本?

查询cuda版本：

```
cat /usr/local/cuda/version.txt
```

查询cudnn版本：

```
cat /usr/local/cuda/include/cudnn.h | grep CUDNN_MAJOR -A 2
```

## 6.1.11 Moxing 安装文件如何获取?

Moxing安装文件不支持下载和用户自主安装。在ModelArts的Notebook和训练作业镜像中预置了Moxing安装包，用户可以直接引用。

## 6.1.12 如何使用 soft NMS 方法降低目标框堆叠度

目前华为云AI市场订阅的算法**YOLOv3-Ascend（物体检测/TensorFlow）**中可以使用soft NMS，YOLOv5算法文档中没有看到相关支持的信息，需要自定义算法进行使用。

## 6.1.13 多节点训练 TensorFlow 框架 ps 节点作为 server 会一直挂着，ModelArts 是怎么判定训练任务结束? 如何知道是哪个节点是 worker 呢?

TensorFlow框架分布式训练的情况下，会启动ps与worker任务组，worker任务组为关键任务组，会以worker任务组的进程退出码，判断训练作业是否结束。

通过task name判断的哪个节点是worker。下发的训练作业是一个volcano job，里边会有两个task：一个是ps、一个是worker。两个task的启动命令不同，会自动生成超参--task_name，ps的--task_name=ps，worker的 --task_name=worker。

## 6.1.14 训练作业的自定义镜像如何安装 Moxing?

为避免自动安装Moxing会影响用户自定义镜像中的包环境，所以自定义镜像需要用户手动安装Moxing。Moxing安装包会在作业启动后放在“/home/ma-user/modelarts/package/”目录下。可在使用Moxing功能前执行如下代码，进行Moxing的安装。

```
import os
os.system("pip install /home/ma-user/modelarts/package/moxing_framework-*.whl")
```

**说明**

本案例仅适用于训练作业环境。

## 6.1.15 子用户使用专属资源池创建训练作业无法选择已有的 SFS Turbo

由于权限不足，导致子用户无法看到已有的SFS Turbo，请为子用户所在用户组添加SFS FullAccess 、SFS Trubo FullAccess权限。

# 6.2 训练过程读取数据

## 6.2.1 在 ModelArts 上训练模型，输入输出数据如何配置?

ModelArts支持用户上传自定义算法创建训练作业。上传自定义算法前，请完成算法开发并上传至OBS桶。创建算法请参考**使用预置框架创建算法**。创建训练作业请参考**创建训练作业**指导。

### 解析输入路径参数、输出路径参数

运行在ModelArts的模型读取存储在OBS服务的数据，或者输出至OBS服务指定路径，输入和输出数据需要配置3个地方：

1. 训练代码中需解析输入路径参数和输出路径参数。ModelArts推荐以下方式实现参数解析。

```
import argparse
# 创建解析
parser = argparse.ArgumentParser(description="train mnist",
                                 formatter_class=argparse.ArgumentDefaultsHelpFormatter)
# 添加参数
parser.add_argument('--train_url', type=str,
                    help='the path model saved')
parser.add_argument('--data_url', type=str, help='the training data')
# 解析参数
args, unknown = parser.parse_known_args()
```

   完成参数解析后，用户使用“data_url”、“train_url”代替算法中数据来源和数据输出所需的路径。

2. 在使用预置框架创建算法时，根据1中的代码参数设置定义的输入输出参数。
   - 训练数据是算法开发中必不可少的输入。“输入”参数建议设置为“data_url”，表示数据输入来源，也支持用户根据1的算法代码自定义代码参数。
   - 模型训练结束后，训练模型以及相关输出信息需保存在OBS路径。“输出”数据默认配置为模型输出，代码参数为“train_url”，也支持用户根据1的算法代码自定义输出路径参数。
3. 在创建训练作业时，填写输入路径和输出路径。

   训练输入选择对应的OBS路径或者数据集路径，训练输出选择对应的OBS路径。

## 6.2.2 如何提升训练效率，同时减少与 OBS 的交互?

### 场景描述

在使用ModelArts进行自定义深度学习训练时，训练数据通常存储在对象存储服务（OBS）中，且训练数据较大时（如200GB以上），每次都需要使用GPU资源池进行训练，且训练效率低。

希望提升训练效率，同时减少与对象存储OBS的交互。可通过如下方式进行调整优化。

## 优化原理

对于ModelArts提供的GPU资源池，每个训练节点会挂载500GB的NVMe类型SSD提供给用户免费使用。此SSD挂载到“/cache”目录，“/cache”目录下的数据生命周期与训练作业生命周期相同，当训练作业运行结束以后“/cache”目录下面所有内容会被清空，腾出空间，供下一次训练作业使用。因此，可以在训练过程中将数据从OBS复制到“/cache”目录，然后每次从“/cache”目录读取数据，直到训练结束。训练结束以后“/cache”目录的内容会自动被清空。

## 优化方式

以TensorFlow代码为例。

优化前代码如下所示：

```
...
tf.flags.DEFINE_string('data_url', '', 'dataset directory.')
FLAGS = tf.flags.FLAGS
mnist = input_data.read_data_sets(FLAGS.data_url, one_hot=True)
```

优化后的代码示例如下，将数据复制至“/cache”目录。

```
...
tf.flags.DEFINE_string('data_url', '', 'dataset directory.')
FLAGS = tf.flags.FLAGS
import moxing as mox
TMP_CACHE_PATH = '/cache/data'
mox.file.copy_parallel('FLAGS.data_url', TMP_CACHE_PATH)
mnist = input_data.read_data_sets(TMP_CACHE_PATH, one_hot=True)
```

# 6.2.3 大量数据文件，训练过程中读取数据效率低?

当数据集存在较多数据文件（即海量小文件），数据存储在OBS中，训练过程需反复从OBS中读取文件，导致训练过程一直在等待文件读取，效率低。

## 解决方法

1. 建议将海量小文件，在本地压缩打包。例如打包成.zip格式。
2. 将此压缩后的文件上传至OBS。
3. 训练时，可直接从OBS下载此压缩文件至/cache目录。此操作仅需执行一次，无需训练过程反复与OBS交互导致训练效率低。

   如下示例，可使用mox.file.copy_parallel将zip文件下载至本地/cache目录并解压，然后再读取做训练。

   ```
   ...
   tf.flags.DEFINE_string('<obs_file_path>/data.zip', '', 'dataset directory.')
   FLAGS = tf.flags.FLAGS
   import os
   import moxing as mox
   TMP_CACHE_PATH = '/cache/data'
   mox.file.copy_parallel('FLAGS.data_url', TMP_CACHE_PATH)
   zip_data_path = os.path.join(TMP_CACHE_PATH, '*.zip')
   unzip_data_path = os.path.join(TEMP_CACHE_PATH, 'unzip')
   #也可以采用zipfile等Python包来做解压
   os.system('unzip '+ zip_data_path + ' -d ' + unzip_data_path)
   mnist = input_data.read_data_sets(unzip_data_path, one_hot=True)
   ```

## 6.2.4 使用 Moxing 时如何定义路径变量?

### 问题描述

```
mox.file.copy_parallel(src_obs_dir=input_storage,'obs://dyyolov8/yolov5_test/yolov5-7.0/datasets'),
```

**mox**这个函数怎么定义以变量的形式填写OBS路径?

### 解决方案

变量定义参考如下示例:

```
input_storage = './test.py'
import moxing as mox
mox.file.copy_parallel(input_storage,'obs://dyyolov8/yolov5_test/yolov5-7.0/datasets')
```

# 6.3 编写训练代码

## 6.3.1 训练模型时引用依赖包，如何创建训练作业?

ModelArts支持训练模型过程中安装第三方依赖包。在训练代码目录下放置“pip-requirements.txt”文件后，在训练启动文件被执行前系统会执行如下命令，以安装用户指定的Python Packages。

```
pip install -r pip-requirements.txt
```

仅使用**预置框架**创建的训练作业支持在训练模型时引用依赖包。

**说明**

pip-requirements.txt文件命名支持以下4种格式，文档中以pip-requirements为例说明。

- pip-requirement.txt
- pip-requirements.txt
- requirement.txt
- requirements.txt

- 代码目录位置请参考**在代码目录下提供安装文件**。
- pip-requirements文件写法请参考**安装文件规范**。

### 在代码目录下提供安装文件

- 如果使用“我的算法”创建训练作业，则在创建算法时，可以把相关文件放置在配置的“代码目录”下，算法的“启动方式”必须选择“预置框架”。
- 如果使用“自定义算法”创建训练作业，则可以把相关文件放置在配置的“代码目录”下，“启动方式”必须选择“预置框架”。

需要在创建训练作业前将相关文件上传至OBS路径下，文件打包要求请参见**安装文件规范**。

### 安装文件规范

请根据依赖包的类型，在代码目录下放置对应文件:

- **依赖包为开源安装包时**

  **说明**

  暂时不支持直接从github的源码中安装。

  在“代码目录”中创建一个命名为“pip-requirements.txt”的文件，并且在文件中写明依赖包的包名及其版本号，格式为“包名==版本号”。

  例如，“代码目录”对应的OBS路径下，包含模型文件，同时还存在“pip-requirements.txt”文件。“代码目录”的结构如下所示：

  ```
  |---模型启动文件所在OBS文件夹
      |---model.py              #模型启动文件。
      |---pip-requirements.txt  #定义的配置文件，用于指定依赖包的包名及版本号。
  ```

  “pip-requirements.txt”文件内容如下所示：

  ```
  alembic==0.8.6
  bleach==1.4.3
  click==6.6
  ```

- **依赖包为whl包时**

  如果训练后台不支持下载开源安装包或者使用用户编译的whl包时，由于系统无法自动下载并安装，因此需要在“代码目录”放置此whl包，同时创建一个命名为“pip-requirements.txt”的文件，并且在文件中指定此whl包的包名。依赖包必须为“.whl”格式的文件。

  例如，“代码目录”对应的OBS路径下，包含模型文件、whl包，同时还存在“pip-requirements.txt”文件。“代码目录”的结构如下所示：

  ```
  |---模型启动文件所在OBS文件夹
      |---model.py              #模型启动文件。
      |---XXX.whl               #依赖包。依赖多个时，此处放置多个。
      |---pip-requirements.txt  #定义的配置文件，用于指定依赖包的包名。
  ```

  “pip-requirements.txt”文件内容如下所示：

  ```
  numpy-1.15.4-cp36-cp36m-manylinux1_x86_64.whl
  tensorflow-1.8.0-cp36-cp36m-manylinux1_x86_64.whl
  ```

## 6.3.2 训练作业常用文件路径是什么?

训练环境的当前目录以及代码目录在容器的位置一般通过环境变量${MA_JOB_DIR}读取，${MA_JOB_DIR}变量对应的实际值是/home/ma-user/modelarts/user-job-dir。

## 6.3.3 如何安装 C++的依赖库?

在训练作业的过程中，会使用到第三方库。以C++为例，请参考如下操作步骤进行安装：

1. 将源码下载至本地并上传到OBS。使用OBS客户端上传文件的操作请参见**上传文件**。
2. 将上传到OBS的源码使用Moxing复制到开发环境Notebook中。

   以下为使用EVS挂载的开发环境，将数据复制至notebook中的代码示例：

   ```
   import moxing as mox
   mox.file.make_dirs('/home/ma-user/work/data')
   mox.file.copy_parallel('obs://bucket-name/data', '/home/ma-user/work/data')
   ```

3. 在Jupyter页面的“Files”页签下，单击“New”，打开“Terminal”。执行如下命令进入目标路径，确认源码已下载，即“data”文件是否存在。

   ```
   cd /home/ma-user/work
   ls
   ```

4. 在“Terminal”环境进行编译，具体编译方式请您根据业务需求进行。

5. 将编译结果使用Moxing复制至OBS中 。代码示例如下：
```
import moxing as mox
mox.file.make_dirs('/home/ma-user/work/data')
mox.file.copy_parallel('/home/ma-user/work/data', 'obs://bucket-name/file)
```
6. 在训练时，将OBS中的编译结果使用Moxing复制到容器中使用。代码示例如下：
```
import moxing as mox
mox.file.make_dirs('/cache/data')
mox.file.copy_parallel('obs://bucket-name/data', '/cache/data')
```

## 6.3.4 训练作业中如何判断文件夹是否复制完毕?

您可以在训练作业启动文件的脚本中，通过如下方式获取复制和被复制文件夹大小，根据结果判断是否复制完毕：

```
import moxing as mox
mox.file.get_size('obs://bucket_name/obs_file',recursive=True)
```

其中，“get_size”为获取文件或文件夹的大小。“recursive=True”表示类型为文件夹，“True”表示是文件夹，“False”为文件。

如果输出结果为一致，表示文件夹复制已完毕。如果输出结果不一致，表示复制未结束。

## 6.3.5 如何在训练中加载部分训练好的参数?

在训练作业时，需要从预训练的模型中加载部分参数，初始化当前模型。请您通过如下方式加载：

1. 通过如下代码，您可以查看所有的参数。
```
from moxing.tensorflow.utils.hyper_param_flags import mox_flags
print(mox_flags.get_help())
```
2. 通过如下方式控制载入模型时需要恢复的参数名。其中，“checkpoint_include_patterns”为需要恢复的参数，“checkpoint_exclude_patterns”为不需要恢复的参数。
```
checkpoint_include_patterns: Variables names patterns to include when restoring checkpoint. Such as: conv2d/weights.
checkpoint_exclude_patterns: Variables names patterns to include when restoring checkpoint. Such as: conv2d/weights.
```
3. 通过以下方式控制需要训练的参数列表。其中，“trainable_include_patterns”为需要训练的参数列表，“trainable_exclude_patterns”为不需要训练的参数列表。
```
--trainable_exclude_patterns: Variables names patterns to exclude for trainable variables. Such as: conv1,conv2.
--trainable_include_patterns: Variables names patterns to include for trainable variables. Such as: logits.
```

## 6.3.6 训练作业的启动文件如何获取训练作业中的参数?

训练作业参数有两种来源，包括后台自动生成的参数和用户手动输入的参数。具体获取方式如下：

1. 创建训练作业时，“输入”支持配置训练的输入参数名称（一般设置为“data_url”），以及输入数据的存储位置，“输出”支持配置训练的输出参数名称（一般设置为“train_url”），以及输出数据的存储位置。
2. 训练作业运行成功之后，在训练作业列表中，您可以单击作业名称，查看该作业的详情。在“日志”页签搜索输入输出参数名称获取参数信息。
3. 如果需在训练中获取“train_url”、“data_url”和“test”参数的值，可在训练作业的启动文件中添加以下代码获取：

```
import argparse
parser = argparse.ArgumentParser()
parser.add_argument('--data_url', type=str, default=None, help='test')
parser.add_argument('--train_url', type=str, default=None, help='test')
parser.add_argument('--test', type=str, default=None, help='test')
```

## 6.3.7 训练作业中使用 os.system('cd xxx')无法进入相应的文件夹?

当在训练作业的启动脚本中使用**os.system('cd xxx')**无法进入相应的文件夹时，建议使用如下方法：

```
import os
os.chdir('/home/work/user-job-dir/xxx')
```

## 6.3.8 训练作业如何调用 shell 脚本，是否可以执行.sh 文件?

ModelArts支持调用shell脚本，可以使用python调用“.sh”。具体操作步骤如下：

1. 上传“.sh”脚本至OBS桶，例如“.sh”所在存储位置为 “ /bucket-name/code/test.sh”。
2. 在本地创建“.py”文件，例如“test.py”。由于后台会自动将代码目录下载至容器的“/home/work/user-job-dir/”目录下，因此您可以在启动文件“test.py”中通过如下方式调用“.sh”文件：
   ```
   import os
   os.system('bash /home/work/user-job-dir/code/test.sh')
   ```
3. 将“test.py”文件上传至OBS中，则该文件存储位置为“/bucket-name/code/test.py”。
4. 创建训练作业时，指定的代码目录为“/bucket-name/code/”，启动文件目录为“/bucket-name/code/test.py”。

训练作业创建完成之后就可以使用python调用“.sh”文件。

## 6.3.9 训练代码中，如何获取依赖文件所在的路径?

由于用户本地开发的代码需要上传至ModelArts后台，训练代码中涉及到依赖文件的路径时，用户设置有误的场景较多。因此推荐通用的解决方案：使用os接口得到依赖文件的绝对路径，避免报错。

以下示例展示如何通过os接口获得其他文件夹下的依赖文件路径。

文件目录结构：

```
project_root                #代码根目录
 └─bootfile.py              #启动文件
 └─otherfileDirectory       #其他依赖文件所在的目录
     └─otherfile.py         #其他依赖文件
```

在启动文件代码中，建议用户参考以下方式获取其他依赖文件所在路径，即示例中的“otherfile_path”。

```
import os
current_path = os.path.dirname(os.path.realpath(__file__)) # 获得启动文件bootfile.py的路径
project_root = os.path.dirname(current_path) # 通过启动文件路径获得工程的根目录，对应ModelArts训练控制台上设置的代码目录
otherfile_path = os.path.join(project_root, "otherfileDirectory", "otherfile.py") # 通过工程的根目录得到依赖文件路径
```

## 6.3.10 自定义 python 包中如果引用 model 目录下的文件，文件路径怎么写

如果容器中的文件实际路径不清楚，可以使用Python获取当前文件路径的方法获取。

```
os.getcwd() #获取文件当前工作目录路径（绝对路径）
os.path.realpath(__ file __) #获得文件所在的路径（绝对路径）
```

也可在搜索引擎寻找其他获取文件路径的方式，使用获取到的路径进行文件读写。

# 6.4 创建训练作业

## 6.4.1 创建训练作业时提示“对象目录大小/数量超过限制”，如何解决?

### 问题分析

创建训练作业选择的代码目录有大小和文件个数限制。

### 解决方法

将代码目录中除代码以外的文件删除或存放到其他目录，保证代码目录大小不超过128MB，文件个数不超过4096个。

## 6.4.2 训练环境中不同规格资源“/cache”目录的大小

在创建训练作业时可以根据训练作业的大小选择资源。

ModelArts会挂载硬盘至“/cache”目录，用户可以使用此目录来储存临时文件。“/cache”与代码目录共用资源，不同资源规格有不同的容量。

**说明**

- k8s磁盘的驱逐策略是90%，所以可以正常使用的磁盘大小应该是“cache目录容量 x 0.9”。
- 裸机的本地磁盘为物理磁盘，无法扩容，如果存储的数据量大，建议使用SFS存放数据，SFS支持扩容。

- GPU规格的资源

  **表 6-2** GPU cache 目录容量

  | GPU规格 | cache目录容量 |
  | --- | --- |
  | GP Vnt1 | 800G |
  | 8*GP Vnt1 | 3T |
  | GP Pnt1 | 800G |

- CPU规格的资源

**表 6-3** CPU cache 目录容量

| CPU规格 | cache目录容量 |
| --- | --- |
| 2 核 8GiB | 50G |
| 8 核 32GiB | 50G |

- Ascend规格的资源

**表 6-4** Ascend cache 目录容量

| Ascend规格 | cache目录容量 |
| --- | --- |
| Ascend | 3T |

## 6.4.3 训练作业的“/cache”目录是否安全?

ModelArts训练作业的程序运行在容器中，容器挂载的目录地址是唯一的，只有运行时的容器能访问到。因此训练作业的“/cache”是安全的。

## 6.4.4 训练作业一直在等待中（排队）?

训练作业状态一直在等待中状态表示当前所选的资源池规格资源紧张，作业需要进行排队，请耐心等待。如想降低排队时间，根据您所选资源池的类型，有以下建议：

1. 公共资源池：

   公共资源池资源较少，高峰期如举办相关活动时会存在资源不足情况。有以下方法可以尝试：

   - 如果使用的是免费规格，可以换成收费规格，免费规格资源较少，排队概率高。
   - 规格选择卡数尽量少，如可以选择1卡，相比于选择8卡排队几率大大降低。
   - 可以尝试使用其他Region（如北京四切换为上海一）。
   - 如果有长期的资源使用诉求，可以购买独占使用的专属资源池。

2. 专属资源池：
   - 如有多个可用的专属资源池，可尝试选择其他较为空闲的资源池。
   - 可清理当前资源池下的其他资源，如停止长时间不使用的Notebook。
   - 在非高峰期时提交训练作业。
   - 如长期长时间排队可以联系该专属资源池的账号管理员，管理员可根据使用情况对资源池进行扩容。

相关问题：为什么资源充足还是在排队?

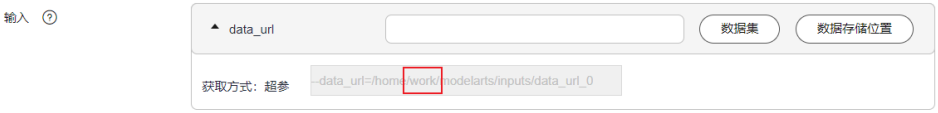
## 6.4.5 创建训练作业时，超参目录为什么有的是/work 有的是/ma-user?

### 问题描述

创建训练作业时，输入输出参数的超参目录有的是/work，有的是/ma-user。

**图 6-1** 目录是/ma-user


**图 6-2** 目录是/work

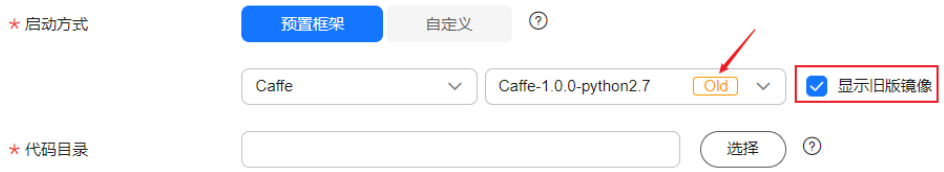



## 解决方案

这是创建训练作业选用的算法有差异导致的。

- 如果选择的算法是使用旧版镜像创建的，那么创建训练作业时输入输出参数的超参目录就是/work。

  **图 6-3** 创建算法

- 如果选择的算法不是使用旧版镜像创建的，那么创建训练作业时输入输出参数的超参目录就是/ma-user。

# 6.4.6 在 ModelArts 创建分布式训练时如何设置 NCCL 环境变量?

ModelArts训练平台预置了部分NCCL环境变量，如表6-5所示。这些环境变量建议保持默认值。

**表 6-5** 预置的环境变量

| 环境变量 | 说明 |
| --- | --- |
| NCCL_SOCKET_IFNAME | 指定通信的网卡名称。 |
| NCCL_IB_GID_INDEX | 系统设置的默认值为3，表示使用RoCE v2协议。 |
| NCCL_IB_TC | 系统设置的默认值为128，表示数据包走交换机的队列4，队列4使用PFC流控机制来保证网络是无损的。 |

如果训练时，需要提升通信稳定性，可以增加配置其他NCCL环境变量，如表6-6所示。

**表 6-6** 建议增加的环境变量

| 环境变量 | 建议值 | 说明 |
| --- | --- | --- |
| NCCL_IB_TIMEOUT | 18 | 用于控制IB通信超时时间，算法为“4.096 µs * 2 ^ timeout”。如出现NCCL通信超时问题可适当调大，最大可调整至22。较大的值可能会影响性能，设置为18相对平衡。 |
| NCCL_IB_RETRY_CNT | 15 | IB通信重试次数。建议设置为最大值15，减少IB通信失败的概率。 |

### 6.4.7 在 ModelArts 使用自定义镜像创建训练作业时如何激活 conda 环境?

由于训练作业运行时不是交互式的shell环境，因此无法直接使用“conda activate”命令激活指定的conda环境。但是，在自定义镜像中可参考以下命令激活conda环境：

```
source /home/ma-user/anaconda3/etc/profile.d/conda.sh; conda activate <conda_env>
```

## 6.5 管理训练作业版本

### 6.5.1 训练作业是否支持定时或周期调用?

ModelArts训练作业不支持定时周期化调用。当您的作业处于“运行中”状态时，可以按照业务需求进行调用。

## 6.6 查看作业详情

### 6.6.1 如何查看训练作业资源占用情况?

在ModelArts管理控制台，选择“模型训练>训练作业”，进入训练作业列表页面。在训练作业列表中，单击目标作业名称，查看该作业的详情。您可以在“资源占用情况”页签查看到如下指标信息。

- CPU：CPU使用率（cpuUsage）百分比（Percent）。
- MEM：物理内存使用率（memUsage）百分比（Percent）。
- GPU：GPU使用率（gpuUtil）百分比（Percent）。
- GPU_MEM：显存使用率（gpuMemUsage）百分比（Percent）。

### 6.6.2 如何访问训练作业的后台?

ModelArts不支持访问训练作业后台。

## 6.6.3 两个训练作业的模型都保存在容器相同的目录下是否有冲突?

ModelArts训练作业之间的存储目录相互不影响，每个环境之间彼此隔离，看不到其他作业的数据。

## 6.6.4 训练输出的日志只保留 3 位有效数字，是否支持更改 loss 值?

在训练作业中，训练输出的日志只保留3位有效数字，当loss过小的时候，显示为0.000。具体日志如下：

```
INFO:tensorflow:global_step/sec: 0.382191
INFO:tensorflow:step: 81600(global step: 81600) sample/sec: 12.098 loss: 0.000
INFO:tensorflow:global_step/sec: 0.382876
INFO:tensorflow:step: 81700(global step: 81700) sample/sec: 12.298 loss: 0.000
```

由于当前不支持更改loss值，您可以通过将loss的值乘以1000来规避此问题。

## 6.6.5 训练好的模型是否可以下载或迁移到其他账号？如何获取下载路径?

通过训练作业训练好的模型可以下载，然后将下载的模型上传存储至其他账号对应区域的OBS中。

### 获取模型下载路径

1. 登录ModelArts管理控制台，在左侧导航栏中选择“模型训练 > 训练作业”，进入“训练作业”列表。
2. 在训练作业列表中，单击目标训练作业名称，查看该作业的详情。
3. 在左侧获取“输出位置”下的路径，即为训练模型的下载路径。

### 模型迁移到其他账号

您可以通过如下两种方式将训练的模型迁移到其他账号。

- 将训练好的模型下载至本地后，上传至目标账号对应区域的OBS桶中。
- 通过对模型存储的目标文件夹或者目标桶配置策略，授权其他账号进行读写操作。详请参见**配置高级桶策略**。

# 7 推理部署

## 7.1 模型管理

### 7.1.1 导入模型

#### 7.1.1.1 如何将 Keras 的.h5 格式模型导入到 ModelArts 中

ModelArts不支持直接导入“.h5”格式的模型。您可以先将Keras的“.h5”格式转换为TensorFlow的格式，然后再导入ModelArts中。

从Keras转TensorFlow操作指导请参见其**官网指导**。

#### 7.1.1.2 导入模型时，模型配置文件中的安装包依赖参数如何编写?

**问题描述**

从OBS中或者从容器镜像中导入模型时，开发者需要编写模型配置文件。模型配置文件描述模型用途、模型计算框架、模型精度、推理代码依赖包以及模型对外API接口。配置文件为JSON格式。配置文件中的“dependencies”，表示配置模型推理代码需要的依赖包，需要提供依赖包名、安装方式和版本约束的信息，详细参数见**模型配置文件编写说明**。导入模型时，模型配置文件中的安装包依赖参数“dependencies”如何编写?

**解决方案**

安装包存在前后依赖关系。例如您在安装“mmcv-full”之前，需要完成“Cython”、“pytest-runner”、“pytest”的安装，在配置文件中，您需要把“Cython”、“pytest-runner”、“pytest”写在“mmcv-full”的前面。

示例如下：

```
"dependencies": [
  {
  "installer": "pip",
  "packages": [
    {
      "package_name": "Cython"
    },
    {
```

```
        "package_name": "pytest-runner"
      },
      {
        "package_name": "pytest"
      },
      {
        "restraint": "ATLEAST",
        "package_version": "5.0.0",
        "package_name": "Pillow"
      },
      {
        "restraint": "ATLEAST",
        "package_version": "1.4.0",
        "package_name": "torch"
      },
      {
        "restraint": "ATLEAST",
        "package_version": "1.19.1",
        "package_name": "numpy"
      },
      {
        "package_name": "mmcv-full"
      }
    ]
  }
]
```

当"mmcv-full"安装失败，原因可能是基础镜像中没有安装gcc，无法编译导致安装失败，此时需要用户使用线下wheel包安装。

示例如下：

```
"dependencies": [
   {
   "installer": "pip",
   "packages": [
      {
        "package_name": "Cython"
      },
      {
        "package_name": "pytest-runner"
      },
      {
        "package_name": "pytest"
      },
      {
        "restraint": "ATLEAST",
        "package_version": "5.0.0",
        "package_name": "Pillow"
      },
      {
        "restraint": "ATLEAST",
        "package_version": "1.4.0",
        "package_name": "torch"
      },
      {
        "restraint": "ATLEAST",
        "package_version": "1.19.1",
        "package_name": "numpy"
      },
      {
        "package_name": "mmcv_full-1.3.9-cp37-cp37m-manylinux1_x86_64.whl"
      }
    ]
   }
]
```

模型配置文件的“dependencies”支持多个“dependency”结构数组以list形式填入。

示例如下：

```
"dependencies": [
  {
  "installer": "pip",
  "packages": [
     {
        "package_name": "Cython"
     },
     {
        "package_name": "pytest-runner"
     },
     {
        "package_name": "pytest"
     },
     {
        "package_name": "mmcv_full-1.3.9-cp37-cp37m-manylinux1_x86_64.whl"
     }
   ]
  },
  {
  "installer": "pip",
  "packages": [
     {
        "restraint": "ATLEAST",
        "package_version": "5.0.0",
        "package_name": "Pillow"
     },
     {
        "restraint": "ATLEAST",
        "package_version": "1.4.0",
        "package_name": "torch"
     },
     {
        "restraint": "ATLEAST",
        "package_version": "1.19.1",
        "package_name": "numpy"
     },
   ]
  }
]
```

### 7.1.1.3 使用自定义镜像创建在线服务，如何修改默认端口

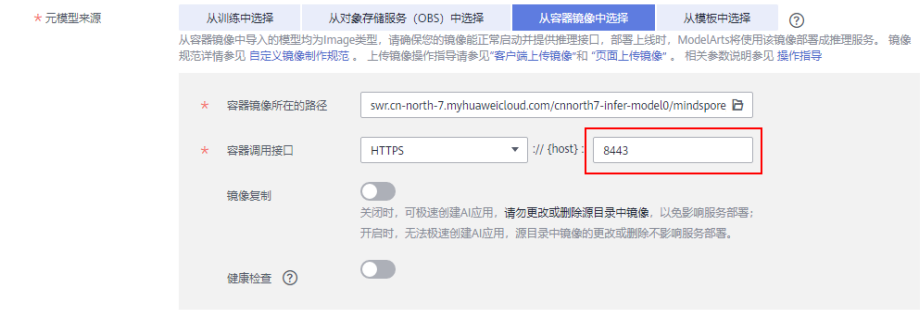
当模型配置文件中定义了具体的端口号，例如：8443，创建AI应用没有配置端口（默认端口号为8080），或者配置了其他端口号，均会导致服务部署失败。您需要把AI应用中的端口号配置为8443，才能保证服务部署成功。

修改默认端口号，具体操作如下：

1. 登录ModelArts控制台，左侧菜单选择“AI应用管理 > AI应用”；
2. 单击“创建”，进入创建AI应用界面，元模型选择“从容器镜像中选择”，选择自定义镜像；
3. 配置“容器调用接口”和端口号，端口号与模型配置文件中的端口保持一致；

图 7-1 修改端口号




4. 设置完成后，单击“立即创建”，等待AI应用状态变为“正常”；
5. 重新部署在线服务。

### 7.1.1.4 ModelArts 平台是否支持多模型导入

ModelArts平台从对象存储服务（OBS）中导入模型包适用于单模型场景。如果有多模型复合场景，推荐使用自定义镜像方式，通过从容器镜像（SWR）中选择元模型的方式创建AI应用部署服务。制作自定义镜像请参考**从0-1制作自定义镜像并创建AI应用**。

### 7.1.1.5 导入 AI 应用对于镜像大小的限制

ModelArts部署使用的是容器化部署，容器运行时有空间大小限制，当用户的模型文件或者其他自定义文件，系统文件超过容器引擎空间大小时，会提示镜像内空间不足。

当前，公共资源池容器引擎空间的大小最大支持50G，专属资源池容器引擎空间的默认为50G，专属资源池容器引擎空间可在创建资源池时自定义设置，设置专属资源池容器引擎空间不会造成额外费用增加。

如果使用的是OBS导入或者训练导入，则包含基础镜像、模型文件、代码、数据文件和下载安装软件包的大小总和。

如果使用的是自定义镜像导入，则包含解压后镜像和镜像下载文件的大小总和。

## 7.2 部署上线

## 7.2.1 功能咨询

### 7.2.1.1 ModelArts 支持将模型部署为哪些类型的服务?

支持在线服务、批量服务和边缘服务。

### 7.2.1.2 在线服务和批量服务有什么区别?

- 在线服务

  将模型部署为一个Web服务，您可以通过管理控制台或者API接口访问在线服务。
- 批量服务

  批量服务可对批量数据进行推理，完成数据处理后自动停止。

批量服务一次性推理批量数据，处理完服务结束。在线服务提供API接口，供用户调用推理。

### 7.2.1.3 在线服务和边缘服务有什么区别?

- 在线服务

  将模型部署为一个Web服务，您可以通过管理控制台或者API接口访问在线服务。

- 边缘服务

  云端服务是集中化的离终端设备较远，对于实时性要求高的计算需求，把计算放在云上会引起网络延时变长、网络拥塞、服务质量下降等问题。而终端设备通常计算能力不足，无法与云端相比。在此情况下，通过在靠近终端设备的地方建立边缘节点，将云端计算能力延伸到靠近终端设备的边缘节点，从而解决上述问题。

  智能边缘平台（Intelligent EdgeFabric）通过纳管您的边缘节点，提供将云上应用延伸到边缘的能力，联动边缘和云端的数据，满足客户对边缘计算资源的远程管控、数据处理、分析决策、智能化的诉求。

  ModelArts支持将模型通过智能边缘平台IEF，在边缘节点将模型部署为一个Web服务。您可以通过API接口访问边缘服务。

### 7.2.1.4 为什么选择不了 Ascend Snt3 资源?

由于Ascend Snt3资源有限，当资源售罄后，您在部署上线时，无法选择Ascend Snt3资源（公共资源池）进行推理，即在部署页面中，“Ascend: 1* Snt3 (8GB) | ARM: 3核 6GB”资源为灰色，无法选择。

**解决方案：**

- 方法1：如果您希望使用公共资源池下的Ascend Snt3，可以等待其他用户释放，即其他使用Ascend Snt3芯片的服务停止，您即可选择此资源进行部署上线。
- 方法2：如果专属资源池还有Ascend Snt3资源，您可以创建一个Ascend Snt3专属资源池使用。
- 方法3：如果专属资源池的Ascend Snt3资源也已售罄，则需等待其他用户删除Ascend Snt3实例后，您才可以创建Ascend Snt3的专属资源池进行使用。

### 7.2.1.5 线上训练得到的模型是否支持离线部署在本地?

通过ModelArts预置算法训练得到的模型是保存在OBS桶里的，模型支持下载到本地。

1. 在训练作业列表找到需要下载模型的训练作业，单击名称进入详情页，获取训练输出路径。

**图 7-2** 获取训练输出位置


2. 单击“输出路径”，跳转至OBS对象路径，下载训练得到的模型。
3. 在本地环境进行离线部署。

   具体请参见**模型调试**章节在本地导入模型，参见**服务调试**章节，将模型离线部署在本地并使用。

## 7.2.1.6 服务预测请求体大小限制是多少?

服务部署完成且服务处于运行中后，可以往该服务发送推理的请求，请求的内容根据模型的不同可以是文本，图片，语音，视频等内容。

当使用调用指南页签中显示的调用地址（华为云APIG网关服务的地址）预测时，对请求体的大小限制是12MB，超过12MB时，请求会被拦截。

如果是从ModelArts console的预测页签进行的预测，由于console的网络链路的不同，此时要求请求体的大小不超过8MB。

因此，尽量避免请求体大小超限。如果有高并发的大流量推理请求，请提工单联系专业服务支持。

## 7.2.1.7 在线服务部署是否支持包周期?

在线服务不支持包周期的计费模式。

## 7.2.1.8 部署服务如何选择计算节点规格?

部署服务时，用户需要指定节点规格进行服务部署，界面目前显示的节点规格是ModelArts根据用户的AI应用和资源池的节点规格计算得到，用户可以选择ModelArts提供的规格，也可以使用自定义规格（公共资源池不支持）。

计算节点规格主要是根据用户AI应用实际需要的资源进行选择，如AI应用正常运行需要3U10G的资源，那么需要选择大于3U10G的计算节点规格。确保服务能够部署成功正常运行。

**图 7-3** 选择计算节点规格


规格的使用注意事项如下：

1、权限控制

通用的计算节点规格是未做权限控制的，如modelarts.vm.cpu.2u，只要资源池有资源，就可以选择使用。一些特殊的规格需要联系系统管理员增加权限。

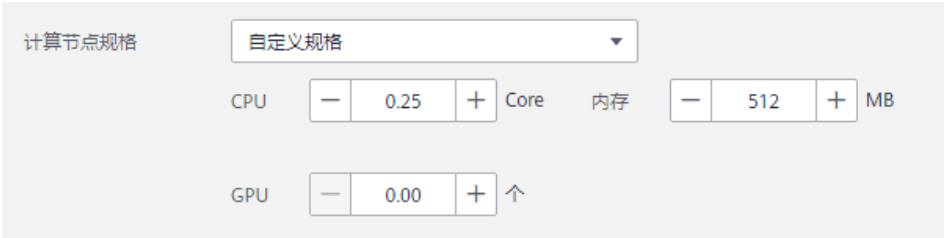
2、公共资源池的规格无法选择

共享池的资源是有限的，显示置灰表示当前规格的资源已经被用完。请选择未置灰的规格，也可以创建自己的专属资源池。

3、自定义规格

只有在专属资源池部署服务时，支持自定义资源规格。公共资源池部署服务不支持。

**图 7-4** 自定义规格



4、免费规格

只有公共资源池支持使用免费节点规格。免费规格有使用数量和使用时长的限制。目前仅在“华北-北京四”区域提供了免费规格。

### 7.2.1.9 部署 GPU 服务支持的 Cuda 版本是多少?

默认支持Cuda版本为10.2，如果需要更高的版本，可以提工单申请技术支持。

## 7.2.2 在线服务

### 7.2.2.1 部署在线服务时，自定义预测脚本 python 依赖包出现冲突，导致运行出错

导入模型时，需同时将对应的推理代码及配置文件放置在模型文件夹下。使用Python编码过程中，推荐采用相对导入方式（Python import）导入自定义包。

如果ModelArts推理框架代码内部存在同名包，而又未采用相对导入，将会出现冲突，导致部署或预测失败。

### 7.2.2.2 在线服务预测时，如何提高预测速度?

- 部署在线服务时，您可以选择性能更好的“计算节点规格”提高预测速度。例如使用GPU资源代替CPU资源。
- 部署在线服务时，您可以增加“计算节点个数”。

  如果节点个数设置为1，表示后台的计算模式是单机模式；如果节点个数设置大于1，表示后台的计算模式为分布式的。您可以根据实际需求进行选择。
- 推理速度与模型复杂度强相关，您可以尝试优化模型提高预测速度。

  ModelArts中提供了模型版本管理的功能，方便溯源和模型反复调优。

**图 7-5** 部署在线服务


### 7.2.2.3 调整模型后，部署新版本 AI 应用能否保持原 API 接口不变?

ModelArts提供多版本支持和灵活的流量策略，您可以通过使用灰度发布，实现模型版本的平滑过渡升级。修改服务部署新版本模型或者切换模型版本时，原服务预测API不会变化。

调整模型版本的操作可以参考如下的步骤。

## 前提条件

- 已存在部署完成的服务。

- 已完成模型调整，创建**AI应用新版本**。

## 操作步骤

1. 登录ModelArts管理控制台，在左侧导航栏中选择“部署上线 > 在线服务”，默认进入“在线服务”列表。
2. 在部署完成的目标服务中，单击操作列的“修改”，进入“修改服务”页面。
3. 在选择模型及配置中，单击“增加模型版本进行灰度发布”添加新版本。

**图 7-6** 灰度发布


4. 您可以设置两个版本的流量占比，服务调用请求根据该比例分配。其他设置可参考**参数说明**。完成设置后，单击下一步。
5. 确认信息无误后，单击“提交”部署在线服务。

## 7.2.2.4 在线服务的 API 接口组成规则是什么?

AI应用部署成在线服务后，用户可以获取API接口用于访问推理。

API接口组成规则如下：

```
https://域名/版本/infer/服务ID
```

示例如下：

```
https://6ac81cdfac4f4a30be95xxxbb682.apig.xxx.xxx.com/v1/infers/
468d146d-278a-4ca2-8830-0b6fb37d3b72
```

**图 7-7** API 接口

## 7.2.2.5 在线服务运行中但是预测失败时，如何排查报错是不是模型原因导致的

### 问题现象

在线服务启动后，当在线服务进入到“运行中”状态后，进行预测，预测请求发出后，收到的响应不符合预期，无法判断是不是模型的问题导致的不符合预期。

### 原因分析

在线服务启动后，ModelArts提供两种方式的预测：

- 方式1：在ModelArts的Console的预测页签进行预测；
- 方式2：在ModelArts的Console的调用指南页签获取到调用地址，然后通过cURL或者Postman等工具进行预测。

无论是方式1还是方式2，当推理请求发送出去后都有可能收到不符合预期的推理结果。

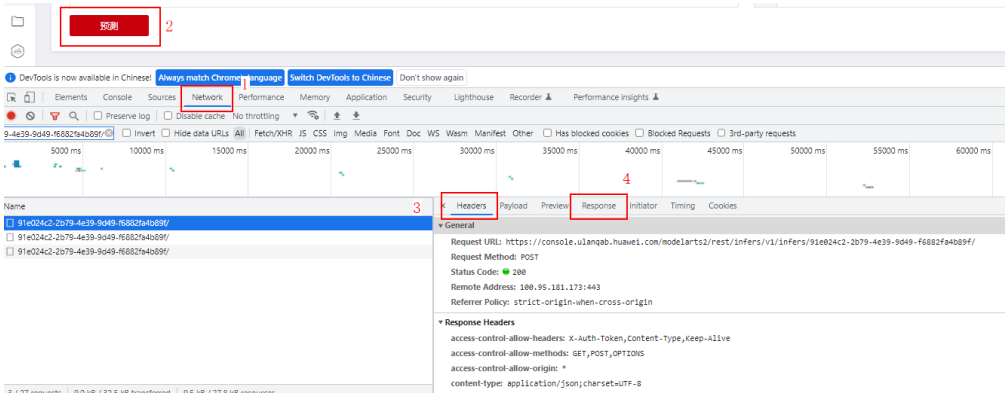
推理请求经过一系列传递后最终是会进入到模型服务中，模型服务可能是以自定义镜像的方式导入的，可能是因为模型服务在处理推理请求时候出现了问题导致结果不符合预期，能准确判断出来是否是在模型服务中出的问题对于快速解决问题帮助很大。

### 处理方法

不管是用方式1还是方式2，要判断是否是模型服务返回的不合预期的结果都需要获取到本次推理请求的response header及response body。

- 如果是方式1，可以通过浏览器的开发者工具获取到推理请求的response信息。以Chrome浏览器为例，可以使用快捷键F12打开开发者工具，然后选择“Network”页签，再单击“预测”，可以在Network页签窗口中看到本次推理请求的response信息如下图。

  **图 7-8** 推理请求的 response 信息

  在name栏找到推理请求，其中推理请求的URL包含“/v1/infers”的关键字，可以在header栏中的url看到完整url，分别在Headers页签及Response页签中查看response的信息。

- 如果是方式2可以根据不同的工具查看response header及body信息，比如CURL命令可以通过-I选项查看response header。

如果查看到的response header中Server字段为ModelArts且response body中没有显示ModelArts.XXXX的错误码，此时收到的response信息为模型服务返回的response信息，如果不符合预期，可以判断为模型服务返回的结果不符合预期。

## 建议与总结

鉴于模型服务有从对象存储服务(OBS)中导入，从容器镜像中导入，从AI Gallery中获取等多种途径，对于上述不同模型服务的来源所产生的常见问题及处理方法建议如下：

- 从容器镜像中导入：由于此种方式镜像为用户完全自定义的镜像，错误原因会因自定义镜像的不同而不同，建议查看模型日志确定错误原因。
- 从对象存储服务(OBS)中导入：如您收到的返回是MR系列错误码，如MR.0105，请查看在线服务详情页面的日志页签查看对应的错误日志。
- 从AI Gallery中获取：请咨询该模型在AI Gallery中的发布者。

# 7.2.2.6 在线服务处于运行中状态时，如何填写推理请求的 request header 和 request body

## 问题现象

部署在线服务完成且在线服务处于“运行中”状态时，通过ModelArts console的调用指南tab页签可以获取到推理请求的地址，但是不知道如何填写推理请求的header及body。

## 原因分析

在线服务部署完成且服务处于运行中状态后，可以通过调用指南页签的调用地址对模型发起预测请求，出于安全考虑，ModelArts会通过相关的认证鉴权机制避免在线服务被无关人员非法调用。所以在预测请求的header信息中包含的是调用者的身份信息，在body部分是需要进行预测的内容。

header的部分需要按照华为云的相关机制进行认证，body部分需要根据模型的要求如前处理脚本的要求，如自定义镜像的要求进行输入。

## 处理方法

- Header：

  在调用指南页签上最多可以获取到两个api地址，分别是支持IAM/AKSK认证的地址以及支持APP认证的地址，对于支持不同认证方式的地址， 对header的组织也不同，具体如下：

  - IAM/AKSK认证方式：需要在header的X-Auth-Token字段上填入该租户在该region的domain级别的token。具体指导参见连接：**获取IAM用户Token**。
  - APP认证的方式：APP认证方式又可以细分为AppCode认证和APP签名认证。
    - AppCode认证需要在header的X-Apig-AppCode字段上填入绑定给该在线服务的APP的AppCode。
    - APP签名认证需要在header的X-Sdk-Date和Authorization字段中填入通过sdk或者工具使用该在线服务绑定的APP的AppKey和AppSecret所生产的这两个字段的值，以完成对该请求的签名认证。具体指导参见链接：**访问在线服务（APP认证）**。

- Body：

  body的组装和模型强相关，不同来源的模型body的组装方式不同。

  - 模型为从容器镜像中导入的：需要按照自定义镜像的要求组织，请咨询该镜像的制作人。
  - 模型为从对象存储(OBS)导入的：此时对body的要求会在推理代码中体现，具体在推理代码的_preprocess方法中，该方法将输入的http body转换成模型期望的输入，具体的指导可以查看文档：**模型推理代码编写说明**。
  - 模型从AI Gallery中获取的：请查看AI Gallery中的调用说明或者咨询该模型的提供方。

## 建议与总结

无

## 7.2.2.7 作为调用发起方的客户端无法访问已经获取到的推理请求地址

## 问题现象

完成在线服务部署且服务处于“运行中”状态后，已经通过调用指南页面的信息获取到调用的server端地址，但是调用发起方的客户端访问该地址不通，出现无法连接、域名无法解析的现象。

## 原因分析

在调用指南页签中显示的调用地址都是华为云APIG（API网关服务）的地址。调用发起方的客户端和华为云网络不通。

## 处理方法

如果客户端位于华为云网络之外，保证客户端所处的网络环境可以连接Internet；

如果客户端位于华为云网络内，默认的网络配置即可以访问通这个地址，避免设置特殊的网络配置，例如防火墙规则等。

## 建议与总结

无

## 7.2.2.8 服务部署失败，报错 ModelArts.3520，服务总数超限

部署服务时，ModelArts报错“ModelArts.3520: 在线服务总数超限，限制为20”，接口返回“A maximum of xxx real-time services are allowed.”，表示服务数量超限。

正常情况下，单个用户最多可创建20个在线服务。可采取以下方式处理：

- 删除状态为“异常”的服务。
- 删除长期不使用的服务。
- 因业务原因需申请更大配额，可提工单申请扩容。

#### 7.2.2.9 配置了合理的服务部署超时时间，服务还是部署失败，无法启动

服务部署成功的标志是模型启动完成，如果没有配置健康检查，就无法检测到模型是否真实的启动。

在自定义镜像健康检查接口中，用户可以实现实际业务是否成功的检测。在创建AI应用时配置健康检查延迟时间，保证容器服务的初始化。

因此，推荐在创建AI应用时配置健康检查，并设置合理的延迟检测时间， 实现实际业务的是否成功的检测，确保服务部署成功。

### 7.2.3 边缘服务

#### 7.2.3.1 什么是边缘节点?

边缘节点是您自己的边缘计算设备，用于运行边缘应用，处理您的数据，并安全、便捷地和云端应用进行协同。

#### 7.2.3.2 更新 AI 应用版本时，边缘服务预测功能不可用?

针对某一部署的边缘服务，如果在更新AI应用版本时，即修改边缘服务，更新其使用的AI应用版本，导致此边缘服务的预测功能暂不可用。

针对此场景，由于更新了AI应用版本，边缘服务将重新部署，处于部署中的边缘服务，则无法使用预测功能。即更新AI应用版本，会导致预测功能中断。等待边缘服务重新处于运行中时，预测功能恢复正常。

#### 7.2.3.3 使用边缘节点部署边缘服务能否使用 http 接口协议?

系统默认使用https。如果您想使用http，可以采取以下两种方式：

- 方式一：在部署边缘服务时添加如下环境变量：
  ```
  MODELARTS_SSL_ENABLED = false
  ```

  **图 7-9** 添加环境变量

- 方式二：在使用自定义镜像导入模型时，创建AI应用页面中“容器调用接口”设置为“http”，再部署边缘服务。

# 8 资源池

## 8.1 ModelArts 支持使用 ECS 创建专属资源池吗?

不支持。创建资源池时，只能选择界面提供的“未售罄”节点规格进行创建。专属资源池的节点规格后台是对应的ECS资源，但是无法使用账号下购买的ECS，作为ModelArts专属资源池。

## 8.2 1 个节点的专属资源池，能否部署多个服务?

支持。

在部署服务时，选择专属资源池，在选择“计算节点规格”时选择“自定义规格”，设置小一些或者选择小规格的服务节点规格，当资源池节点可以容纳多个服务节点规格时，就可以部署多个服务。如果使用此方式进行部署推理，选择的规格务必满足模型的要求，当设置的规格过小，无法满足模型的最小推理要求时，则会出现部署失败或预测失败的情况。

**图 8-1** 设置自定义规格


## 8.3 专属资源池购买后，中途扩容了一个节点，如何计费?

华为云会重新计算一个增加了该节点的账单，付费以后才能使用。

## 8.4 共享池和专属池的区别是什么?

- 共享池是所有ModelArts共享的一个资源池，当使用人数比较多的时候，可能造成资源紧张而产生排队。
- 专属池是专属于您的资源池，不会因为资源紧张而产生排队，同时专属资源池支持打通自己的VPC，能和自己的资源网络互通。

## 8.5 如何通过 ssh 登录专属资源池节点?

ModelArts专属资源池不支持ssh登录节点。

## 8.6 训练任务的排队逻辑是什么?

当前训练任务排队的逻辑是先进先出，前面的任务没运行完后面的任务不会运行，有可能会造成小任务被“饿死”，需要用户注意。

**说明**

饿死指的是前面的任务被一个大的任务堵着（例如是64卡），需要等空闲64卡这个任务才能运行，64卡的任务后面跟着1卡的。即使现在空出来30卡，这个1卡的任务也排不上。

## 8.7 专属资源池下的在线服务停止后，启动新的在线服务，提示资源不足

停止在线服务后，需要等待几分钟等待资源释放。

## 8.8 不同实例的资源池安装的 cuda 和驱动版本号分别是什么?

专属资源池的cuda和驱动版本是可以根据用户的要求安装。如果需要调整，需提工单。

## 8.9 算法运行时需要依赖鉴权服务，公共资源池是否支持两者打通网络?

不支持，公共资源池不能打通网络。可通过专属资源池打通网络，使用ModelArts服务。

## 8.10 创建失败的专属资源池删除后，控制台为什么还能看到?

在控制台页面操作删除专属资源池后，后端服务需要进行资源实例释放。在资源实例释放过程中，用户依然可以查询到资源池。如果需要创建专属资源池，建议等待5min

后再创建，且不要使用已创建过的专属资源池名称来命名新建的专属资源池。如果做UI自动化测试，建议用例用随机串替代。

# 8.11 训练专属资源池如何与 SFS 弹性文件系统配置对等链接?

配置训练专属资源池与SFS弹性文件系统的对等链接，需要资源池打通VPC，使得资源池与SFS弹性文件系统所配置的VPC相同。配置完成后，在创建训练作业时，就可以看到SFS的配置选项。

打通VPC步骤请参考**打通VPC**。

# 9 AI Gallery

## 9.1 AI Gallery 的入口在哪里

- **控制台入口**
  a. 登录ModelArts管理控制台。
  b. 在左侧导航栏中选择“AI Gallery”跳转到AI Gallery首页。
- **直接网址访问**
  旧版AI Gallery将下线，已不再更新，建议使用新版AI Gallery。
  - 旧版AI Gallery地址：**https://developer.huaweicloud.com/develop/aigallery/home.html**
  - 新版AI Gallery地址：**https://pangu.huaweicloud.com/gallery/home.html**

## 9.2 在 AI Gallery 订阅商品失败怎么办?

AI Gallery是在ModelArts的基础上构建的开发者生态社区，提供模型、算法、HiLens技能、数据集等内容的共享。当您订阅商品失败可参照如下方式解决：

- 请检查您是否完成实名认证。
  账号注册成功后，您需要完成“实名认证”才可以正常使用服务。具体认证方式请参见**实名认证**。
- 进入当前账号的费用中心，检查是否欠费。
  如果欠费，建议您参考**华为云账户充值**，为您的账号充值。
- 如果以上都没问题，请尝试退出账号重新登录。
  单击页面右上角的账号，选择“退出登录”，并重新登录。

## 9.3 在 AI Gallery 订阅的数据集可以在 SDK 中使用吗?

支持。

将AI Gallery数据集下载至OBS，然后在SDK直接使用此OBS目录下的数据即可。详细操作步骤如下所示：

1. 将AI Gallery数据集下载至OBS。详细指导请参见**下载数据集**。

   数据集可以直接下载至OBS，也可以下载至ModelArts数据集中，不管任何方式，其最终的存储路径均为OBS目录。

   - 下载至OBS时，在下载任务完成后，数据将存储在下载时设置的OBS目录中。请注意下载任务中设置的区域，后续使用SDK或ModelArts控制台时，使用的区域需一致。
   - 下载至数据集时，可以在下载任务完成后，前往ModelArts控制台，发布此数据集，在“数据集输出位置”参数获取对应的OBS目录。即数据集存储的位置。

2. 在SDK中调用对应OBS目录下的数据。

   SDK的下载和使用，请参见**《SDK参考》**。

   可参考**从OBS下载文件**，通过接口直接使用上述步骤中下载的数据集。

# 9.4 AI Gallery 支持哪些区域?

首先，AI Gallery本身不区分区域，任意区域进入AI Gallery，其展示的商品和功能是一致的。

- 订阅算法或模型。

  从AI Gallery订阅的算法、模型，不区分区域，但是，在前往控制台使用订阅的产品时，需选择对应的区域。支持的区域与ModelArts相同，包含华北-北京一、华北-北京四、华东-上一、华南-广州（以界面上实际支持的区域为准）。

- 下载数据集。

  在AI Gallery中下载数据集时，不管是下载至OBS还是下载至数据集，均需设置对应的使用区域。支持的区域与ModelArts相同，包含华北-北京一、华北-北京四、华东-上一、华南-广州（以界面上实际支持的区域为准）。

# 9.5 AI Gallery 下载数据到 OBS 中使用的带宽是用户自己的还是华为云的?

AI Gallery下载数据到OBS中使用的带宽是华为云的。

# 10 API/SDK

## 10.1 ModelArts SDK、OBS SDK 和 MoXing 的区别?

### ModelArts SDK

ModelArts服务提供的SDK，可调用ModelArts功能。您可以下载SDK至本地调用接口，也可以在ModelArts Notebook中直接调用。

ModelArts SDK提供了OBS管理、训练管理、模型管理、服务管理等几个模块功能。目前，仅提供了Python语言的ModelArts SDK接口。

详细指导文档：《**ModelArts SDK参考**》

### OBS SDK

OBS服务提供的SDK，对OBS进行操作。由于ModelArts较多功能需使用OBS中存储的数据，用户可使用OBS SDK进行调用，使用OBS存储您的数据。

OBS提供了多种语言SDK供选择，开发者可根据使用习惯下载OBS SDK进行调用。使用OBS SDK前，需下载OBS SDK包，然后在本地开发环境中安装使用。

详细指导：《**OBS SDK参考**》

### MoXing

MoXing是ModelArts自研的组件，是一种轻型的分布式框架，构建于TensorFlow、PyTorch、MXNet、MindSpore等深度学习引擎之上，使得这些计算引擎分布式性能更高，同时易用性更好。MoXing包含很多组件，其中MoXing Framework模块是一个基础公共组件，可用于访问OBS服务，和具体的AI引擎解耦，在ModelArts支持的所有AI引擎(TensorFlow、MXNet、PyTorch、MindSpore等)下均可以使用。

MoXing Framework模块提供了OBS中常见的数据文件操作，如读写、列举、创建文件夹、查询、移动、复制、删除等。

在ModelArts Notebook中使用MoXing接口时，可直接调用接口，无需下载或安装SDK，使用限制比ModelArts SDK和OBS SDK少，非常便捷。

详细指导：《**MoXing开发指南**》

# 10.2 ModelArts 的 API 或 SDK 支持模型下载到本地吗?

ModelArts的API和SDK不支持模型下载到本地，但训练作业输出的模型是存放在对象存储服务（OBS）里面的，您可以通过OBS的API或SDK下载存储在OBS中的文件，具体请参见**从OBS下载文件**。

# 10.3 ModelArts 的 SDK 支持哪些安装环境?

ModelArts的SDK支持在Notebook或本地环境中使用，但是不同环境下的不同架构，支持情况不同，如**表10-1**所示。

**表 10-1** SDK 安装环境

| 开发环境 | 架构 | 是否支持 |
| --- | --- | --- |
| Notebook | ARM | 是 |
| | X86 | 是 |
| 本地环境 | ARM | 否 |
| | X86 | 是 |

# 10.4 ModelArts 通过 OBS 的 API 访问 OBS 中的文件，算内网还是公网?

在同一区域，ModelArts通过OBS的API访问OBS中的文件属于内网通信，不消耗公网流量费。

如果是通过互联网从OBS下载数据到本地，这时候会产生OBS公网流量费。OBS的详细计费说明可以参见**计费项**。

# 10.5 调用 API 提交训练作业后，能否绘制作业的资源占用率曲线?

调用API提交训练作业后，您可登录ModelArts控制台，在“模型训练 > 训练作业”中，单击“名称/ID”进入“训练作业详情”页面的“资源占用情况”模块，查看作业的资源占用率曲线。

# 10.6 如何使用 API 接口获取订阅算法的订阅 id 和版本 id?

调用API接口使用“我的订阅”方式创建训练作业时，请求参数需要填写算法的订阅id（algorithm.subscription_id）和版本id（algorithm.item_version_id）。可调用如下接口获取相关信息，如下以北京四为例：

1. 从AI Gallery获取订阅的算法列表

```
GET https://modelarts.cn-north-4.myhuaweicloud.com/v1/aihub/subscriptions?
content_types=algo&offset=0&limit=5&sort_dir=desc
```

获取订阅算法的subscription_id，假设为43b22aeb-5b28-4fad-9581-e3c16d5a3e68，该值即为算法的订阅id。

2. 根据subscription_id获取订阅算法的版本列表

```
GET https://modelarts.cn-north-4.myhuaweicloud.com/v1/aihub/subscriptions/
43b22aeb-5b28-4fad-9581-e3c16d5a3e68/versions
```

获取订阅算法的版本的version_id，该值即为算法的版本id。

# 10.7 使用 SDK 如何查看旧版专属资源池列表?

可参考如下代码查看旧版专属资源池列表：

```
from modelarts.session import Session
from modelarts.estimator import Estimator
algo_info = Estimator(modelarts_session=Session()).get_job_pool_list()print(algo_info)
```

# 10.8 调用 API 接口创建训练作业和部署服务时，如何填写资源池的参数?

- 调用API接口创建训练作业时，“pool_id”为“资源池ID”。
- 调用API接口部署在线服务时，“pool_name”为“资源池ID” 。

**图 10-1** 资源池 ID

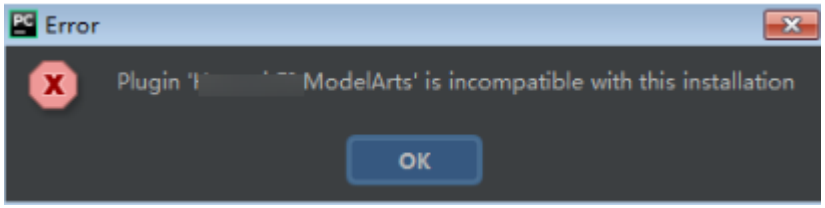
# 11 PyCharm Toolkit 使用

## 11.1 安装 ToolKit 工具时出现错误，如何处理?

### 问题现象

在安装ToolKit工具过程中，出现如下错误。

**图 11-1** 错误提示



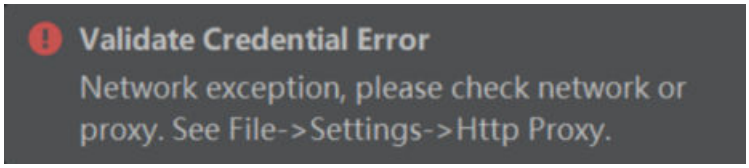

### 解决措施

此问题是因为插件版本和PyCharm版本不一致导致的，需要获取和PyCharm同一版本的插件安装，即2019.2或以上版本。

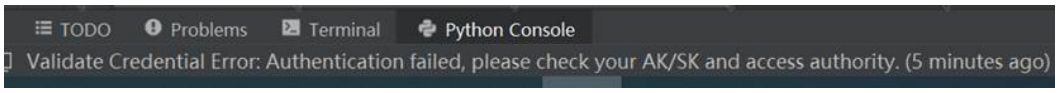
## 11.2 PyCharm ToolKit 工具中 Edit Credential 时，出现错误

### 问题现象

PyCharm ToolKit工具中Edit Credential时，提示Validate Credential error。




或




## 原因分析

- 可能原因一：Region等信息配置不正确
- 可能原因二：未配置hosts文件或者hosts文件信息配置不正确
- 可能原因三：网络代理设置
- 可能原因四：AK/SK不正确
- 可能原因五：电脑时间设置错误

## 解决措施

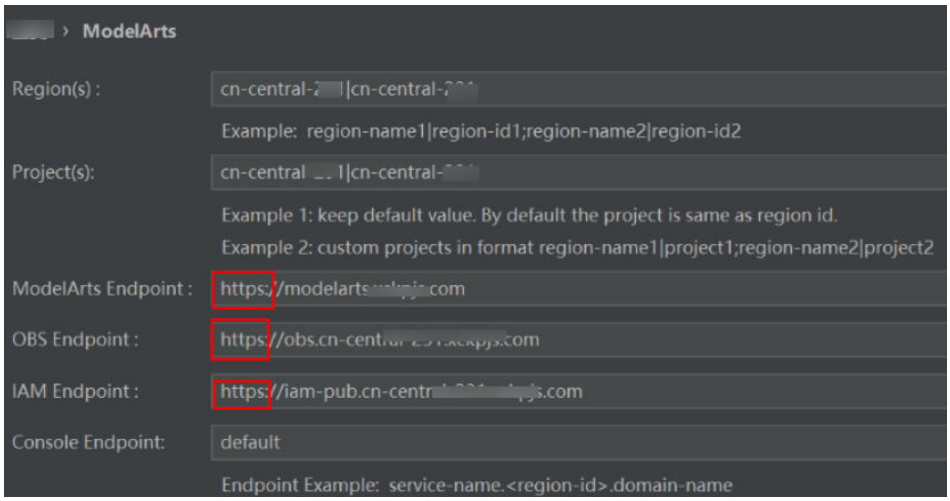
**一、Region等信息配置不正确**

配置正确的Region、Projects、Endpoint信息。

例如：Endpoint配置不正确也会导致认证失败。

错误示例：Endpoint参数前面带了https，正确的配置中不需要有https。

**图 11-2** 配置 ToolKit




**二、未配置hosts文件或者hosts文件信息配置不正确**

在本地PC的hosts文件中配置域名和IP地址的对应关系。

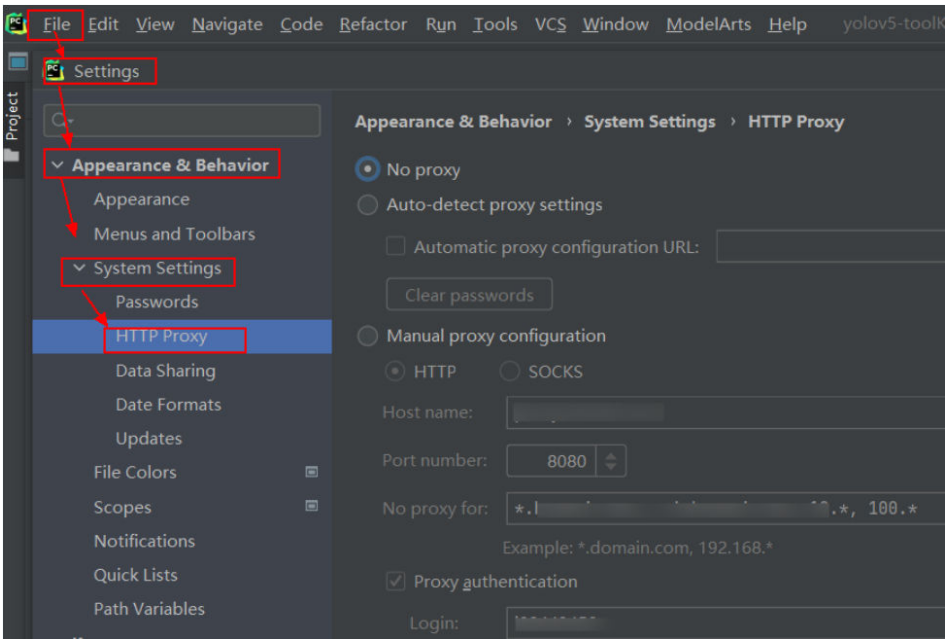
**三、网络代理设置**

如果用户使用的网络有代理设置要求，请检查代理配置是否正确。也可以使用手机热点网络连接进行测试排查。

检查代理配置是否正确。

**图 11-3** PyCharm 网络代理设置




**四、AK/SK不正确**

获取到的AK/SK信息不正确，请确认获取到正确的AK/SK信息再进行尝试，具体请参考**创建访问密钥（AK和SK）**。

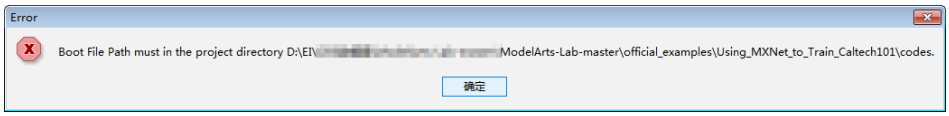
**五、电脑时间设置错误**

请设置电脑时间为正确时间。

# 11.3 为什么无法启动训练?

如果启动脚本选择了不属于本工程的代码，则无法启动训练，错误信息如下图所示。建议将启动脚本添加至本工程，或者是打开启动脚本所在工程后，再启动训练作业。

**图 11-4** 错误信息




# 11.4 提交训练作业时，出现 xxx isn't existed in train_version 错误

## 问题现象

提交训练作业时，出现xxx isn't existed in train_version错误，如下所示。

**图 11-5** xxx isn't existed in train_version 错误


## 原因分析

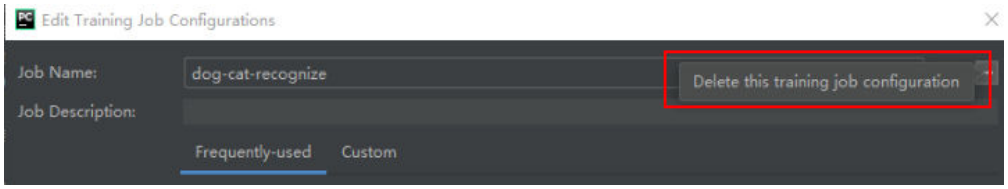
出现上述错误，是因为用户通过PyCharm ToolKit提交了训练作业后，又登录了ModelArts管理控制台删除了该训练作业导致的。

由于PyCharm ToolKit本地记录了云端ModelArts的训练作业ID，如果手动在ModelArts管理控制台删除，本地提交作业时候会报找不到该ID的作业。

## 解决措施

如果已经在ModelArts管理控制台删除了作业，需要在ToolKit也同步删除掉本地的配置信息，单击“Edit Training Configuration”，找到作业名称，单击右上角的减号并确认删除。

**图 11-6** 删除配置信息




在弹出的确认对话框中，确认信息无误后，单击“是”删除对应配置信息。删除后您可以创建新的训练作业配置并提交训练作业。

# 11.5 提交训练作业报错“Invalid OBS path”

当运行训练作业时，出现“Invalid OBS path”错误。

**图 11-7** 训练作业报错


该问题可以通过以下方法排查：

- 如果是首次使用ModelArts的新用户，需要登录ModelArts管理控制台，在全局配置页面完成访问授权配置，推荐使用委托授权的方式。完成全局配置后，再次提交作业即可。
- 请排查所填写的Data Path in OBS是否存在，文件夹下是否有数据文件，如果没有，需要在OBS创建目录并上传训练数据到该目录。

# 11.6 使用 PyCharm Toolkit 提交训练作业报错 NoSuchKey

## 问题现象

使用PyCharm Toolkit提交训练作业时，训练作业详情页的“日志”页签存在报错“errorCode:NoSuchKey”。

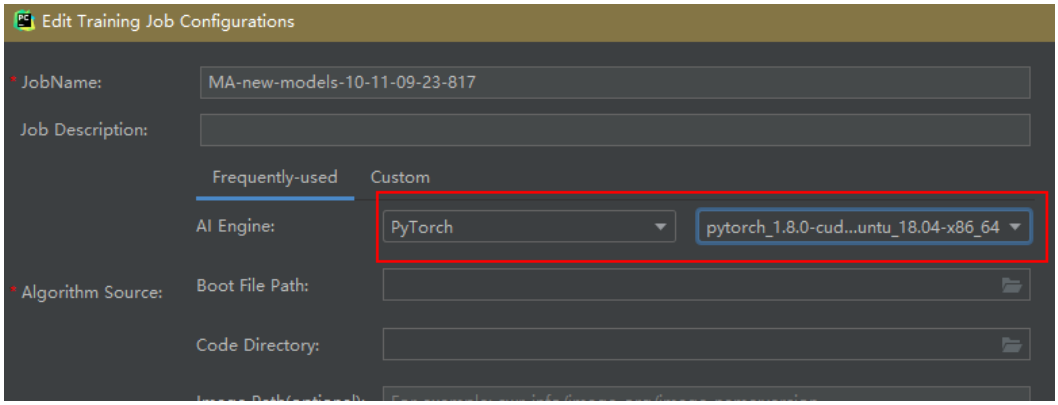
## 原因分析

检查配置后发现，是镜像版本太低，旧版的镜像与当前训练作业不兼容。

## 解决措施

使用PyCharm Toolkit提交训练作业时，常用框架选择训练作业支持的版本，具体支持哪些版本请参考**训练作业支持的AI引擎**。PyTorch的举例：不要选PyTorch-1.0.0、PyTorch-1.3.0、PyTorch-1.4.0。选择如下图：

**图 11-8** 选择训练作业支持的 AI 框架




# 11.7 部署上线时，出现错误

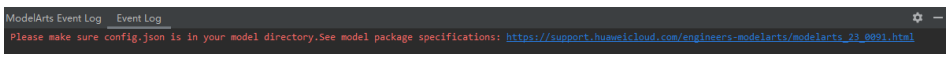
在部署上线前，您需要基于训练后的模型编写配置文件和推理代码。

如果您的模型存储路径下，缺少配置文件“confi.json”，或者缺少推理代码“customize_service.py”时，将出现错误，错误信息如下图所示。

解决方案：

请参考**模型包规范**写配置文件和推理代码，并存储至需部署的模型所在OBS目录下。

**图 11-9** 错误信息




# 11.8 如何查看 PyCharm ToolKit 的错误日志

PyCharm ToolKit的错误日志记录在PyCharm的“idea.log”中，以Windows为例，该文件的路径在“C:\Users\xxx\.IdeaIC2019.2\system\log\idea.log”。

在日志中搜索“modelarts”，可以查看所有和PyCharm ToolKit相关的日志。

# 11.9 如何通过 PyCharm ToolKit 创建多个作业同时训练?

PyCharm ToolKit一次只能运行一个作业，运行第二个作业时需要手动将第一个作业停止。

# 11.10 使用 PyCharm ToolKit ，提示 Error occurs when accessing to OBS

## 问题现象

**查看PyCharm ToolKit的日志**，报错信息为：Error occurs when accessing to OBS。

## 原因分析

可能是用户无OBS权限。

## 解决方法

判断用户是否有OBS权限。

**步骤1** 登录ModelArts控制台，进入“数据管理 > 数据集 ”，单击“创建数据集”，如果可以成功访问对应的OBS路径，表示用户有OBS权限。如果没有OBS权限，请执行**步骤2**配置OBS权限。

**步骤2** 如没有OBS权限，请**配置OBS权限**配置。

**----结束**

# 12 Lite Server

## 12.1 GPU A 系列裸金属服务器如何进行 RoCE 性能带宽测试?

### 场景描述

本文主要指导如何在GPU A系列裸金属服务器上测试RoCE性能带宽。

### 前提条件

GPU A系列裸金属服务器已经安装了IB驱动。（网卡设备名称可以使用ibstatus或者ibstat获取。华为云Ant8裸金属服务器使用Ubuntu20.04操作系统默认已经安装IB驱动。）

### 操作步骤

**方法1：使用mlx硬件计数器，估算ROCE网卡收发流量**

统计300s内流量，统计脚本如下：

```
x=$(cat /sys/class/infiniband/mlx5_2/ports/1/counters/port_rcv_data)
sleep 300
y=$(cat /sys/class/infiniband/mlx5_2/ports/1/counters/port_rcv_data)
res=$(($y-$x))
echo $res
```

上述获取的值*4/300 ，即为当前网卡的接收速率，单位Byte/s。

**方法2：使用ib_write_bw测试RDMA的读写处理确定带宽**

服务器A：服务端从mlx4_0网卡接收数据

```
ib_write_bw -a -d mlx5_0
```

服务器B：客户端向服务端mlx4_0网卡发送数据。

```
ib_write_bw -a -F 服务器A的IP -d mlx5_0 --report_gbits
```

**图 12-1** 服务器 A 执行结果

```
(base) root@devserver-gpu-[redacted]-roce-2:~# ib_write_bw -a -d mlx5_0

************************************
* Waiting for client to connect... *
************************************
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : mlx5_0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 PCIe relax order: ON
 ibv_wr* API     : ON
 CQ Moderation   : 100
 Mtu             : 4096[B]
 Link type       : Ethernet
 GID index       : 3
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
local address: LID 0000 QPN 0x00be PSN 0x101d60 RKey 0x189535 VAddr 0x007fc0683d9000
GID: 00:00:00:00:00:00:00:00:00:00:255:255:29:31:55:128
remote address: LID 0000 QPN 0x00be PSN 0x342cce RKey 0x189535 VAddr 0x007fb1d0f2f000
GID: 00:00:00:00:00:00:00:00:00:00:255:255:29:31:53:22
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]
 8388608    5000             73.96              64.82              0.000966
---------------------------------------------------------------------------------------
```

**图 12-2** 服务器 B 执行结果

```
(base) root@devserver-gpu-[redacted]-roce-3:~# ib_write_bw -a -F 192.168.102.236 -d mlx5_0 --report_gbits
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : mlx5_0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 PCIe relax order: ON
 ibv_wr* API     : ON
 TX depth        : 128
 CQ Moderation   : 100
 Mtu             : 4096[B]
 Link type       : Ethernet
 GID index       : 3
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x00be PSN 0x342cce RKey 0x189535 VAddr 0x007fb1d0f2f000
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:29:31:53:22
 remote address: LID 0000 QPN 0x00be PSN 0x101d60 RKey 0x189535 VAddr 0x007fc0683d9000
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:29:31:55:128
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[Gb/sec]    BW average[Gb/sec]   MsgRate[Mpps]
 2          5000           0.042403           0.020440             1.277489
 4          5000             0.14               0.14               4.327294
 8          5000             0.27               0.20               3.184282
 16         5000             0.54               0.50               3.925457
 32         5000             1.11               1.10               4.296528
 64         5000             1.97               0.76               1.483763
 128        5000             4.43               3.70               3.615876
 256        5000             8.81               5.88               2.872231
 512        5000             17.19              17.17              4.192633
 1024       5000             34.66              34.62              4.225802
 2048       5000             62.22              25.17              1.536477
 4096       5000             91.45              91.37              2.788505
 8192       5000             95.12              39.66              0.605207
 16384      5000             96.05              56.11              0.428078
 32768      5000             95.14              52.60              0.200663
 65536      5000             94.60              44.86              0.085564
 131072     5000             64.14              49.90              0.047587
 262144     5000             81.49              56.76              0.027066
 524288     5000             90.90              59.10              0.014090
 1048576    5000             88.51              65.10              0.007760
 2097152    5000             91.43              88.60              0.005281
 4194304    5000             90.91              86.47              0.002577
 8388608    5000             73.96              64.82              0.000966
---------------------------------------------------------------------------------------
```

# 12.2 GPU A 系列裸金属服务器节点内如何进行 NVLINK 带宽性能测试方法?

## 场景描述

本文指导如何进行节点内NVLINK带宽性能测试，适用的环境为：Ant8或者Ant1 GPU裸金属服务器， 且服务器中已经安装相关GPU驱动软件，以及Pytorch2.0。

GPU A系列裸金属服务器，单台服务器GPU间是走NVLINK，可以通过相关命令查询GPU拓扑模式：

```
nvidia-smi topo -m
```

**图 12-3** 查询 GPU 拓扑模式

| | GPU0 | GPU1 | GPU2 | GPU3 | GPU4 | GPU5 | GPU6 | GPU7 | NIC0 | NIC1 | NIC2 | NIC3 | NIC4 | NIC5 | NIC6 | NIC7 | Affinity |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPU0 | X | NV8 | NV8 | NV8 | NV8 | NV8 | NV8 | NV8 | PXB | PXB | NODE | NODE | SYS | SYS | SYS | SYS | |
| GPU1 | NV8 | X | NV8 | NV8 | NV8 | NV8 | NV8 | NV8 | PXB | PXB | NODE | NODE | SYS | SYS | SYS | SYS | |
| GPU2 | NV8 | NV8 | X | NV8 | NV8 | NV8 | NV8 | NV8 | NODE | NODE | PXB | PXB | SYS | SYS | SYS | SYS | |
| GPU3 | NV8 | NV8 | NV8 | X | NV8 | NV8 | NV8 | NV8 | NODE | NODE | PXB | PXB | SYS | SYS | SYS | SYS | |
| GPU4 | NV8 | NV8 | NV8 | NV8 | X | NV8 | NV8 | NV8 | SYS | SYS | SYS | SYS | PXB | PXB | NODE | NODE | |
| GPU5 | NV8 | NV8 | NV8 | NV8 | NV8 | X | NV8 | NV8 | SYS | SYS | SYS | SYS | PXB | PXB | NODE | NODE | |
| GPU6 | NV8 | NV8 | NV8 | NV8 | NV8 | NV8 | X | NV8 | SYS | SYS | SYS | SYS | NODE | NODE | PXB | PXB | |
| GPU7 | NV8 | NV8 | NV8 | NV8 | NV8 | NV8 | NV8 | X | SYS | SYS | SYS | SYS | NODE | NODE | PXB | PXB | |
| NIC0 | PXB | PXB | NODE | NODE | SYS | SYS | SYS | SYS | X | PIX | NODE | NODE | SYS | SYS | SYS | SYS | |
| NIC1 | PXB | PXB | NODE | NODE | SYS | SYS | SYS | SYS | PIX | X | NODE | NODE | SYS | SYS | SYS | SYS | |
| NIC2 | NODE | NODE | PXB | PXB | SYS | SYS | SYS | SYS | NODE | NODE | X | PIX | SYS | SYS | SYS | SYS | |
| NIC3 | NODE | NODE | PXB | PXB | SYS | SYS | SYS | SYS | NODE | NODE | PIX | X | SYS | SYS | SYS | SYS | |
| NIC4 | SYS | SYS | SYS | SYS | PXB | PXB | NODE | NODE | SYS | SYS | SYS | SYS | X | PIX | NODE | NODE | |
| NIC5 | SYS | SYS | SYS | SYS | PXB | PXB | NODE | NODE | SYS | SYS | SYS | SYS | PIX | X | NODE | NODE | |
| NIC6 | SYS | SYS | SYS | SYS | NODE | NODE | PXB | PXB | SYS | SYS | SYS | SYS | NODE | NODE | X | PIX | |
| NIC7 | SYS | SYS | SYS | SYS | NODE | NODE | PXB | PXB | SYS | SYS | SYS | SYS | NODE | NODE | PIX | X | |

## 操作步骤

**步骤1** 使用以下脚本测得GPU服务器内NVLINK带宽性能。

```
import torch
import numpy as np

device = torch.device("cuda")

n_gpus = 8
data_size = 1024 * 1024 * 1024  # 1 GB

speed_matrix = np.zeros((n_gpus, n_gpus))

for i in range(n_gpus):
    for j in range(i + 1, n_gpus):
        print(f"Testing communication between GPU {i} and GPU {j}...")
        with torch.cuda.device(i):
            data = torch.randn(data_size, device=device)
            torch.cuda.synchronize()
        with torch.cuda.device(j):
            result = torch.randn(data_size, device=device)
            torch.cuda.synchronize()
        with torch.cuda.device(i):
            start = torch.cuda.Event(enable_timing=True)
            end = torch.cuda.Event(enable_timing=True)
            start.record()
            result.copy_(data)
            end.record()
            torch.cuda.synchronize()
            elapsed_time_ms = start.elapsed_time(end)
        transfer_rate = data_size / elapsed_time_ms * 1000 * 8 / 1e9
        speed_matrix[i][j] = transfer_rate
        speed_matrix[j][i] = transfer_rate

print(speed_matrix)
```

**步骤2** 以Ant8 GPU裸金属服务器为例， 其理论GPU卡间带宽为：NVIDIA*NVLink*Bridge for 2GPUS: 400GB/s。使用上述测试脚本测得带宽性能进行如下分析。

- 正常模式-NVLINK全互通，带宽约为370GB。基本符合预期，且证明Ant GPU裸金属服务器内部GPU间确实走NVLINK模式，且完全互联。

**图 12-4** 正常模式带宽性能

| | GPU0 | GPU1 | GPU2 | GPU3 | GPU4 | GPU5 | GPU6 | GPU7 |
|---|---|---|---|---|---|---|---|---|
| GPU0 | 0.00 | 307.55 | 369.55 | 369.00 | 369.00 | 368.83 | 368.59 | 369.19 |
| GPU1 | 307.55 | 0.00 | 369.26 | 370.37 | 368.07 | 368.78 | 369.01 | 370.36 |
| GPU2 | 369.55 | 369.26 | 0.00 | 368.15 | 370.90 | 370.48 | 370.84 | 370.95 |
| GPU3 | 369.00 | 370.37 | 368.15 | 0.00 | 368.05 | 370.16 | 370.42 | 370.38 |
| GPU4 | 369.00 | 368.07 | 370.90 | 368.05 | 0.00 | 368.02 | 370.01 | 370.97 |
| GPU5 | 368.83 | 368.78 | 370.48 | 370.16 | 368.02 | 0.00 | 369.75 | 369.99 |
| GPU6 | 368.59 | 369.01 | 370.84 | 370.42 | 370.01 | 369.75 | 0.00 | 367.77 |
| GPU7 | 369.19 | 370.36 | 370.95 | 370.38 | 370.97 | 369.99 | 367.77 | 0.00 |
| 服务器1，116.204.125.186 | | | | | | | | |

- 异常模式-NVLINK部分互通，出现带宽波动较大的情况。如下图中GPU0和GPU4之间带宽远低于理论值， 存在问题。

**图 12-5** 异常模式带宽性能

| | GPU0 | GPU1 | GPU2 | GPU3 | GPU4 | GPU5 | GPU6 | GPU7 |
|---|---|---|---|---|---|---|---|---|
| GPU0 | 0.00 | 367.18 | 368.41 | 369.04 | **24.45** | 368.85 | 368.73 | 368.82 |
| GPU1 | 367.18 | 0.00 | 369.53 | 370.42 | 370.34 | 370.53 | 370.81 | 370.95 |
| GPU2 | 368.41 | 369.53 | 0.00 | 370.37 | 370.78 | 370.81 | 370.63 | 370.92 |
| GPU3 | 369.04 | 370.42 | 370.37 | 0.00 | 370.19 | 370.47 | 370.70 | 370.74 |
| GPU4 | **24.45** | 370.34 | 370.78 | 370.19 | 0.00 | 370.25 | 370.49 | 370.53 |
| GPU5 | 368.85 | 370.53 | 370.81 | 370.47 | 370.25 | 0.00 | 370.36 | 370.13 |
| GPU6 | 368.73 | 370.81 | 370.63 | 370.70 | 370.49 | 370.36 | 0.00 | 368.94 |
| GPU7 | 368.82 | 370.95 | 370.92 | 370.74 | 370.53 | 370.13 | 368.94 | 0.00 |

出现这种现象， 可尝试重装nvidia/cuda/nvidia-fabricmanager， 重装后再测试又恢复到了正式模式，GPU0和GPU4之间带宽恢复到370GB/s。

可能原因如下，仅供参考：

a. 驱动程序问题：可能是由于驱动程序没有正确安装或配置，导致NVLINK带宽受限。重新安装nvidia驱动、CUDA和nvidia-fabricmanager等软件后，驱动程序可能已经正确配置，从而解决了这个问题。

b. 硬件问题：如果GPU之间的NVLINK连接存在硬件故障，那么这可能会导致带宽受限。重新安装软件后，重启系统，可能触发了某种硬件自检或修复机制，从而恢复了正常的带宽。

c. 系统负载问题：最初测试GPU卡间带宽时，可能存在其他系统负载，如进程、服务等，这些负载会占用一部分网络带宽，从而影响NVLINK带宽的表现。重新安装软件后，这些负载可能被清除，从而使NVLINK带宽恢复正常。

**----结束**

# 12.3 如何将 Ubuntu20.04 内核版本从低版本升级至 5.4.0-144-generic?

## 场景描述

Ubuntu20.04内核版本从低版本升级至5.4.0-144-generic。

## 操作指导

**步骤1** 检查当前内核版本。

```
uname -r
```

**步骤2** 升级内核

```
apt-get install linux-headers-5.4.0-144-generic linux-image-5.4.0-144-generic
grub-mkconfig -o /boot/efi/EFI/ubuntu/grub.cfg
reboot
```

第一条命令为安装Linux内核头文件和内核镜像，其中版本为5.4.0-144-generic。

第二条命令为重新生成GRUB引导程序的配置文件，用于在启动计算机时加载操作系统， 命令将使用新安装的内核镜像更新GRUB的配置文件，以便在下次启动时加载新的内核。

**----结束**

# 12.4 如何禁止 Ubuntu 20.04 内核自动升级?

## 场景描述

在Ubuntu 20.04每次内核升级后，系统需要重新启动以加载新内核。如果您已经安装了自动更新功能，则系统将自动下载和安装可用的更新，这可能导致系统在不经意间被重启；如果使用的软件依赖于特定版本的内核，那么当系统自动更新到新的内核版本时，可能会出现兼容性问题。在使用Ubuntu20.04时，建议手动控制内核的更新。

**说明**

禁用自动更新可能会导致您的系统变得不安全，因为您需要手动安装重要的安全补丁。在禁用自动更新之前，请确保您已了解其中的风险。

## 操作步骤

在Ubuntu 20.04上禁止内核自动升级，步骤如下：

**步骤1** 禁用unattended-upgrades。

“unattended-upgrades”是一个用于安装安全更新的软件包。要禁用它，首先打开“/etc/apt/apt.conf.d/20auto-upgrades”文件：

```
vi /etc/apt/apt.conf.d/20auto-upgrades
```

将其中的“Unattended-Upgrade "1";”改为“Unattended-Upgrade "0";”以禁用自动更新，然后保存文件并退出。

**步骤2** 将当前内核版本锁定。

要禁止特定的内核版本更新，你可以使用“apt-mark”命令将其锁定。

首先，检查你当前的内核版本：

```
uname -r
```

例如，如果你的内核版本是“5.4.0-42-generic”，你需要锁定所有与此版本相关的软件包。可执行以下命令：

```
sudo apt-mark hold linux-image-5.4.0-42-generic linux-headers-5.4.0-42-generic linux-modules-5.4.0-42-generic linux-modules-extra-5.4.0-42-generic
```

**步骤3** 禁用自动更新

要禁用所有自动更新，首先打开“/etc/apt/apt.conf.d/10periodic”文件：

```
vi /etc/apt/apt.conf.d/10periodic
```

修改文件以将所有选项设置为“0”：

```
APT::Periodic::Update-Package-Lists "0";
APT::Periodic::Download-Upgradeable-Packages "0";
APT::Periodic::AutocleanInterval "0";
APT::Periodic::Unattended-Upgrade "0";
```

保存文件并退出。

执行完以上步骤后，您的Ubuntu 20.04系统将不会自动升级内核。

**----结束**

# 12.5 哪里可以了解 Atlas800 训练服务器硬件相关内容

## 场景描述

本文提供Atlas800训练服务器硬件相关指南，包括三维视图、备件信息、HCCL常用方法以及网卡配置信息。

## Atlas 800 训练服务器三维视图

Atlas 800 训练服务器（型号9000）是基于华为鲲鹏920+Snt9处理器的AI训练服务器，实现完全自主可控，广泛应用于深度学习模型开发和AI训练服务场景，可单击**此处**查看硬件三维视图。

## Atlas 800 训练服务器 HCCN Tool

**Atlas 800 训练服务器 1.0.11 HCCN Tool接口参考**主要介绍集群网络工具hccn_tool对外接口说明，包括配置RoCE网卡的IP、网关，配置网络检测对象IP和查询LLDP信息等。

## Atlas 800 训练服务器备件查询助手

**备件查询助手**可以帮助你查询服务器的所有部件、规格描述，数量等详细信息。

打开网站后请输入SN编码“2102313LNR10P5100077”， 如果失效可以提工单至华为云ModelArts查询。

## Atlas 800 训练服务器的网卡配置问题

1. 机头网卡配置是什么？

   有以下两类网卡：

   - 四个2*100GE网卡，为RoCE网卡，插在NPU板。
   - 一个4*25GE/10GE，为Hi1822网卡，插在主板上的。

2. ifconfig能看到的网卡信息吗

   能看到主板上的网卡信息，即VPC分配的私有IP。若要看RoCE网卡的命令需要执行“hccn_tools”命令查看，参考**Atlas 800 训练服务器 1.0.11 HCCN Tool接口参考**中的指导。

3. NPU上的网卡在哪里可以看到， 会健康检查吗？

8*NPU的网卡为机头上配置的四个2*100GE网卡。华为云有网卡健康状态监控机制。

# 12.6 使用 GPU A 系列裸金属服务器有哪些注意事项?

使用华为云A系列裸金属服务器时有如下注意事项：

1. nvidia-fabricmanager版本号必须和nvidia-driver版本号保持一致，可参考**安装nvidia-fabricmanag**方法。
2. NCCL必须和CUDA版本相匹配，可单击**此处**可查看配套关系和安装方法。
3. 使用该裸金属服务器制作自定义镜像时， 必须清除残留文件，请参考**清理文件**。

# 12.7 GPU A 系列裸金属服务器如何更换 NVIDIA 和 CUDA?

## 场景描述

当裸金属服务器预置的NVIDIA版本和业务需求不匹配时，需要更换NVIDIA驱动和CUDA版本。本文介绍华为云A系列GPU裸金属服务器（Ubuntu20.04系统）如何从“NVIDIA 525+CUDA 12.0”更换为“NVIDIA 515+CUDA 11.7”。

## 操作步骤

**步骤1** 卸载原有版本的NVIDIA和CUDA。

1. 查看使用apt包管理方式安装的nvidia软件包， 执行如下命令实现查看和卸载。
```
dpkg -l | grep nvidia
dpkg -l | grep cuda
sudo apt-get autoremove --purge nvidia-*
sudo apt-get autoremove --purge cuda-*
```
以上命令可以卸载nvidia-driver、cuda、nvidia-fabricmanager、nvidia-peer-memory四个软件。

但是如果nvidia和cuda是使用runfile(local)方式安装的，那么需要在下一步中再次卸载。
2. 若使用nvidia run包直接安装的驱动，需要找到对应的卸载命令。
```
sudo /usr/bin/nvidia-uninstall
sudo /usr/local/cuda-11.7/bin/cuda-uninstaller
```
3. 验证是否卸载完成。
```
nvidia-smi
nvcc -V
dpkg -l | grep peer
dpkg -l | grep fabricmanager
dpkg -l | grep nvidia
```

**步骤2** 卸载nccl相关软件。

由于nccl和cuda是配套关系， 当cuda版本从12.0更换为11.7的时候，libnccl和libnccl-dev都需要更换为和cuda11.7匹配的版本。因此必须卸载掉原版本。

```
sudo apt-get autoremove --purge *nccl*
```

**步骤3** 删除原nccl-test的编译后文件。

由于nccl-test make编译也是基于当前cuda12.0版本的。 当cuda版本更换后，需要重新编译， 因此删除它。默认该文件在/root/nccl-tests直接删除即可。

**步骤4** 从内核中卸载nvidia相关的所有进程。

在安装nvidia驱动时，必须把内核中加载nvidia相关的进程卸载，否则会失败。具体操作请参考**卸载nvidia驱动**。

**说明**

若遇到加载到内核的nvidia进程循环依赖，无法从内核中卸载nvidia，此时执行reboot命令重启服务器即可。

**步骤5** 安装NVIDIA-515和CUDA-11.7配套软件环境。具体步骤请参考**安装nvidia**。

**----结束**

# 13 Lite Cluster

## 13.1 Cluster 资源池如何进行 NCCl Test?

ModelArts提供AI诊断功能，用户可以通过NCCl Test，测试节点GPU状态，并且测试多个节点间的通信速度。

### 操作步骤

**步骤1** 单击资源池名称，进入资源池详情。

**步骤2** 单击左侧“AI组件管理 > AI诊断”。

**步骤3** 单击“诊断”，选择“日志上传路径”和NCCL Test节点，其余参数可保持默认值或根据实际需求修改。

- 测试使用的最大数据：取值范围[1, 1024]，单位可选为“B”、“KB”、“MB”、“GB”“TB”。测试使用的最大数据须大于开始测试使用的最小数据。
- 开始测试使用的最小数据：取值范围[1, 1024]，单位可选为“B”、“KB”、“MB”、“GB”“TB”。
- 日志上传路径：AI诊断日志上传路径。
- 数据增加方式：当前支持乘法方式。
- 乘法系数：数值范围[2, 100]。
- 超过时间：数值范围[150, 3600]。
- NCCL Test节点名称列表：不可为空，且被选择的节点须为可用状态。

**步骤4** 单击“确认”，即可开始诊断。

**----结束**