

华为认证 AI 系列教程

# HCIP-AI-MindSpore

## Developer

### 实验环境搭建及恢复指南

版本：1.0



华为技术有限公司

版权所有 © 华为技术有限公司 2021。保留一切权利。

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

## 商标声明



和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

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

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。

## 华为技术有限公司

地址：          深圳市龙岗区坂田华为总部办公楼          邮编：518129

网址：          <http://e.huawei.com>

---

## 华为认证体系介绍

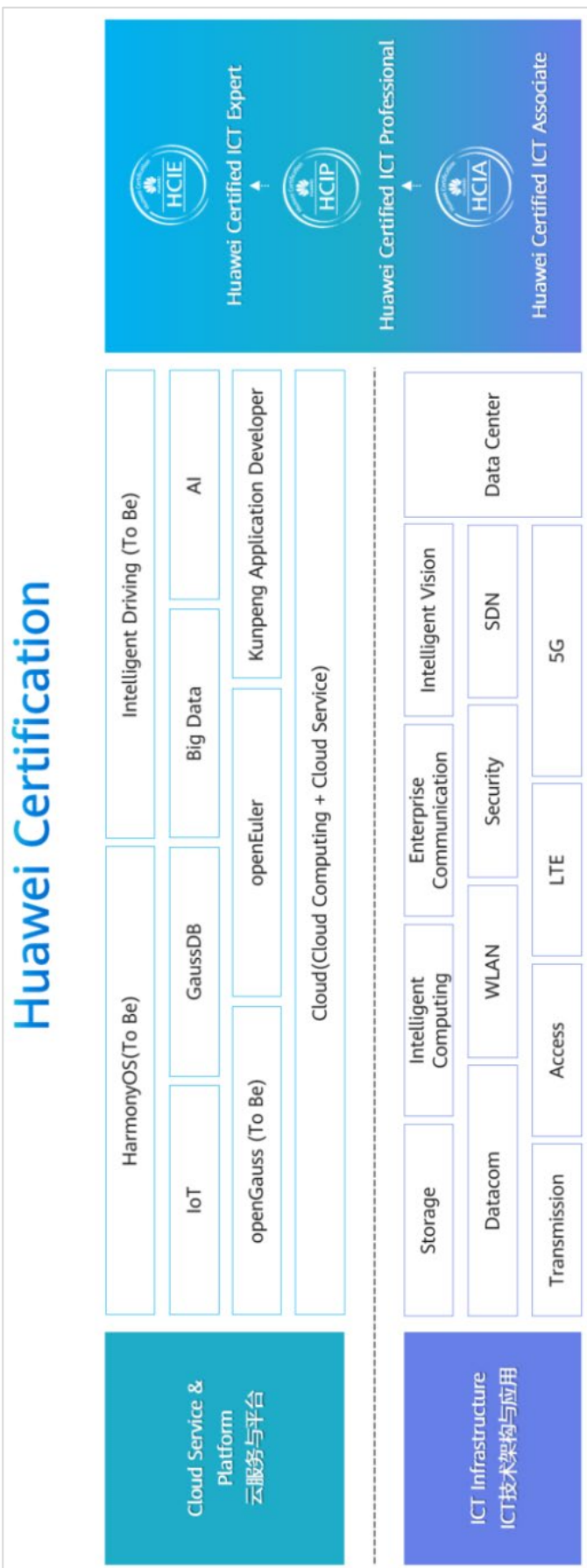
华为认证是华为公司基于“平台+生态”战略，围绕“云-管-端”协同的新ICT技术架构，打造的覆盖ICT（Information and Communications Technology 信息技术）全技术领域的认证体系，包含ICT技术架构与应用认证、云服务与平台认证两类认证。

根据ICT从业者的学习和进阶需求，华为认证分为工程师级别、高级工程师级别和专家级别三个认证等级。

华为认证覆盖ICT全领域，符合ICT融合的技术趋势，致力于提供领先的人才培养体系和认证标准，培养数字化时代新型ICT人才，构建良性ICT人才生态。

HCIP-AI-MindSpore Developer V1.0 定位于培养和认证具备使用华为深度学习框架MindSpore进行人工智能开发能力的算法工程师。

通过HCIP-AI-MindSpore Developer V1.0认证，您将掌握MindSpore和MindSpore Lite的架构与特性相关知识，理解MindSpore网络迁移、分布式训练、端云侧推理与部署等相关特性；具备使用MindSpore完成图像处理任务和自然语言处理任务的能力；能够胜任人工智能算法工程师、深度学习工程师、图像算法工程师、自然语言处理算法工程师等岗位。



# 前言

---

## 简介

本书为 HCIP-AI-MindSpore Developer V1.0 中文版认证培训教程，适用于准备参加 HCIP-AI-MindSpore 考试的学员或者希望了解 AI 基础知识及 MindSpore 编程基础的读者。以下三个部分请学员根据各自电脑的环境进行配置。

## 内容描述

本实验指导书共包含 4 个部分。

- 第一部分为 Windows 实验环境搭建。
- 第二部分为 Ubuntu 实验环境搭建。
- 第三部分为华为云使用指导手册。

## 读者知识背景

本课程为华为认证基础课程，为了更好地掌握本书内容，阅读本书的读者应首先具备以下基本条件：

- 具有基本的 python 知识背景，同时熟悉 MindSpore 基本概念，了解基本 MindSpore 编程知识。

# 目录

<b>前 言</b>	<b>3</b>
简介	3
内容描述	3
读者知识背景	3
<b>1 Windows 实验环境配置</b>	<b>6</b>
1.1 实验介绍	6
1.1.1 关于本实验	6
1.1.2 实验目的	6
1.1.3 实验所需模块	6
1.2 Anaconda 安装	6
1.2.1 Anaconda 简介	6
1.2.2 安装步骤	7
1.3 更换国内源	7
1.3.1 Conda 命令更换国内源	7
1.3.2 pip 命令更换国内源	8
1.4 安装 MindSpore	8
1.4.1 MindSpore 简介	8
1.4.2 MindSpore1.3 版本安装	9
<b>2 Ubuntu 实验环境配置</b>	<b>10</b>
2.1 Miniconda 安装	10
2.2 创建虚拟环境	14
2.3 pip 换源	14
2.4 安装 MindSpore	15
<b>3 华为云使用指导手册</b>	<b>17</b>
3.1 实验介绍	17
3.1.1 关于本实验	17
3.1.2 实验目的	17
3.1.3 实验所需模块	17
3.2 华为云账号申请与实名认证	17
3.2.1 账户用途	17



---

3.2.2 注册步骤.....	17
3.3 对象存储服务 OBS.....	19
3.3.1 简介 .....	19
3.4 ModelArts.....	22
3.4.2 自动学习服务.....	24
3.5 notebook 开发.....	24

# 1

## Windows 实验环境配置

---

### 1.1 实验介绍

#### 1.1.1 关于本实验

本章实验主要用于指导学员基于 Windows 系统搭建适配 HCIP-AI 所有实验的开发环境，包括 Anaconda 下载安装，python 版本选择，以及 MindSpore1.3.0 的下载。

#### 1.1.2 实验目的

基于 Windows 系统搭建 CPU 版本的 HCIP-AI 实验开发环境。

#### 1.1.3 实验所需模块

Anaconda 3.7 或更高版本

虚拟环境（MindSpore）python 版本 3.7.5

MindSpore 1.3.0

### 1.2 Anaconda 安装

#### 1.2.1 Anaconda 简介

Anaconda 是一个用于科学计算的 python 发行版，支持 Linux, Mac, Windows 系统，提供了包管理与环境管理的功能，可以很方便地解决多版本 python 并存、切换以及各种第三方包安装问题。Anaconda 利用工具/命令 conda 来进行 package 和 environment 的管理（也可以使用 pip），并且已经包含了 python 和相关的配套工具。Anaconda 是适用于企业级大数据分析的 python 工具。其包含了 720 多个数据科学相关的开源包，在数据可视化、机器学习、深度学习等多方面都有涉及。不仅可以做数据分析，甚至可以用在大数据和人工智能领域。

Anaconda 有如下特性将极大方便开发者的使用：

1. 安装完了 Anaconda 就不需要再安装 python 了，且 python 版本会在下载时进行选择。



2. 开发时，有时需要不同的框架进行支撑，只需要对 Anaconda 添加虚拟环境，就可以在不同环境中完成开发，而不需要顾及兼容性问题，可以为特殊项目配置相应的环境，方便管理。

## 1.2.2 安装步骤

### 步骤 1 下载 Anaconda

登陆 Anaconda 官网下载安装包 <https://www.Anaconda.com/distribution/#download-section>，根据系统不同，可以选择 Windows、Linux 版本，这里选择 Windows，选择 python3.8 version，点击 64-Bit Graphical Installer 下载（32 位电脑无法匹配该课程）。

### 步骤 2 安装 Anaconda

双击下载好的 Anaconda3-x.x.x-Windows-x86\_64.exe 文件，出现如下界面，点击 Next，点击 I Agree。

Install for: Just me 还是 All Users,这里直接 Just Me,继续点击 Next 。

### 步骤 3 设置安装路径

选择软件安装地址，继续 Next。

### 步骤 4 配置环境变量

两个都勾选，第一个是加入环境变量，第二个是默认使用 python 3.8，可以减少我们后续配置的工作，点击 Install 开始安装。（注意：如果本机器上已经装有 python 的其他版本，建议先删除，然后安装 Anaconda。如未删除，建议两个选项都不勾选，否则容易引发路径错误）。

等待安装完成，点击 finish 即可。

## 1.3 更换国内源

### 1.3.1 Conda 命令更换国内源

通常情况下，使用 conda 命令或者 pip 命令都是从国外的服务器上下载我们需要的模块包的，这在网速不佳的情况下会消耗大量的时间。所以这里我们建议更换国内的源来进行模块下载，速度大大提升。

在命令提示符界面输入以下命令将 conda 命令的源更换为清华源。

```
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/Anaconda/pkgsg/free/  
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/Anaconda/pkgsg/main/
```

注：在安装 Jupyter Notebook 或者 Spyder 时，建议使用 conda 安装，因为 conda 安装后在开始菜单中会出现对应的图标，且名称后的括号内会注明该 IDE 属于哪一个虚拟环境，方便启动。

## 1.3.2 pip 命令更换国内源

上一小节我们介绍了如何给 conda 换源，这里我们介绍如何给 pip 换源。pip 源有两种方式可以更换，一种是临时更换，仅限于当前的这一条命令，另一种为默认更换，更换后所有命令默认以更换后的源来进行资源下载。建议在下载安装 TensorFlow 和 MindSpore 之前进行换源。

### 1.3.2.1 临时更换 pip 源

在每一条命令后面加 -i 源网址，即可更换，如下所示：

```
pip install tensorflow -i http://pypi.douban.com/simple
```

有些时候也会报错。此时请在命令后加 --trusted-host 源网址，如下所示：

```
pip install tensorflow -i http://pypi.douban.com/simple --trusted-host pypi.douban.com
```

### 1.3.2.2 默认更换 pip 源

在 “C:\Users\用户名” 这个目录下创建 pip 文件夹，在 pip 文件夹内创建一个 txt 文件，输入以下代码：

```
[global]
index-url = http://pypi.douban.com/simple/
[install]
trusted-host =pypi.douban.com
```

然后，将这个文件另存为文件名是 pip.ini，保存类型是“所有文件”。

## 1.4 安装 MindSpore

### 1.4.1 MindSpore 简介

MindSpore 是端边云全场景按需协同的华为自研 AI 计算框架，提供全场景统一 API，为全场景 AI 的模型开发、模型运行、模型部署提供端到端能力。

MindSpore 采用端-边-云按需协作分布式架构、微分原生编程新范式以及 AI Native 新执行模式，实现更好的资源效率、安全可信，同时降低行业 AI 开发门槛、释放昇腾芯片算力，助力普惠 AI。

## 1.4.2 MindSpore1.3 版本安装

步骤 1 新建一个命令行窗口，新建一个虚拟环境（参考 1.5 节），保证 python 版本为 3.7.5。可以使用如下命令：`conda create -n MindSpore python==3.7.5`。红色斜体部分是环境名称，可以自行定义。同时，注意 python 版本必须为 3.7.5 版本。其他版本暂时不支持。

步骤 2 使用 `acitivate` 命令激活该虚拟环境（红色部分为自己的环境名）。

```
activate MindSpore
```

步骤 3 步骤 3 输入以下命令安装 MindSpore 1.3 版本，因为版本更新较快，可以参考官网安装不同的版本：<https://www.mindspore.cn/install>。

```
pip install https://ms-release.obs.cn-north-4.myhuaweicloud.com/1.3.0/MindSpore/cpu/x86_64/mindspore-1.3.0-cp37-cp37m-win_amd64.whl --trusted-host ms-release.obs.cn-north-4.myhuaweicloud.com -i https://pypi.tuna.tsinghua.edu.cn/simple
```

步骤 4 安装成功后输入 `python`，在命令行中进入开发环境，输入以下命令导入 MindSpore，如果没有报错则安装成功。

```
import mindspore
```

# 2 Ubuntu 实验环境配置

## 2.1 Miniconda 安装

从 <https://docs.conda.io/en/latest/miniconda.html> 下载 miniconda 的 Linux 版本对应的 64 位安装包，由于官方源下载速度慢，实验所用安装包为清华源下载，带有 x86\_64 的为 64 位安装包。

步骤 1 找到下载的文件，然后右击文件，点击 Properties，然后点击 permissions，下方执行那里打勾，添加文件执行权限。

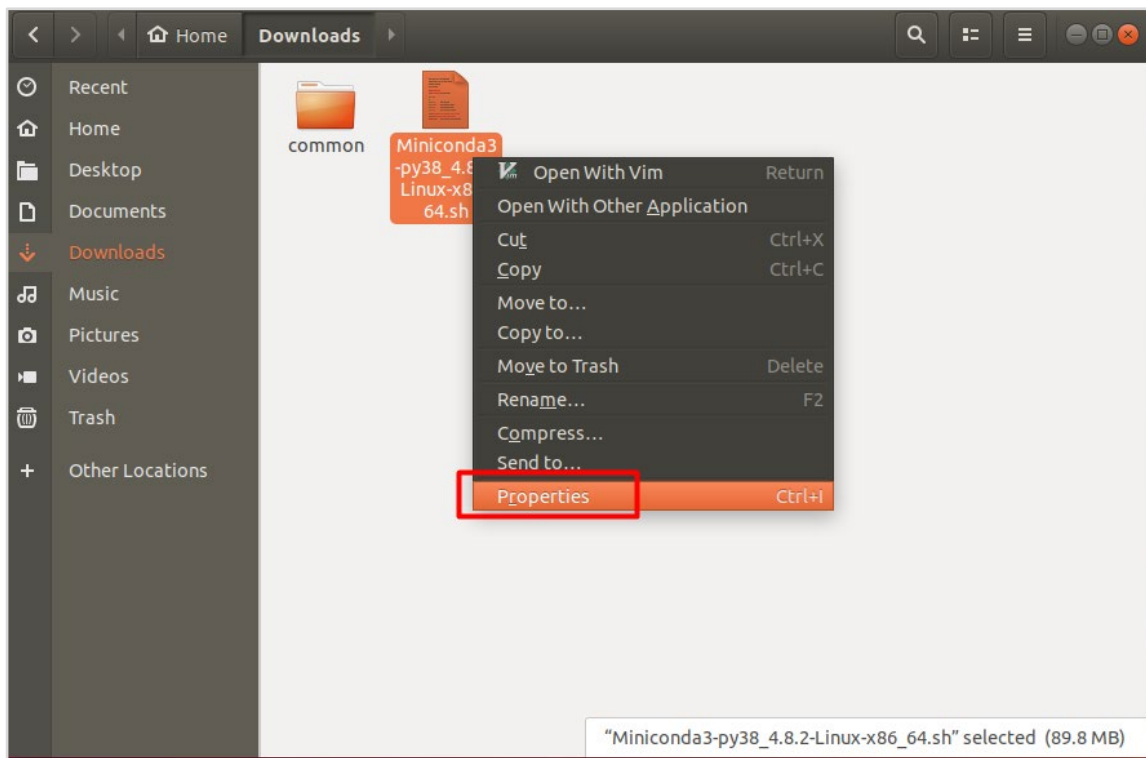


图2-1 文件执行权限配置界面（1）

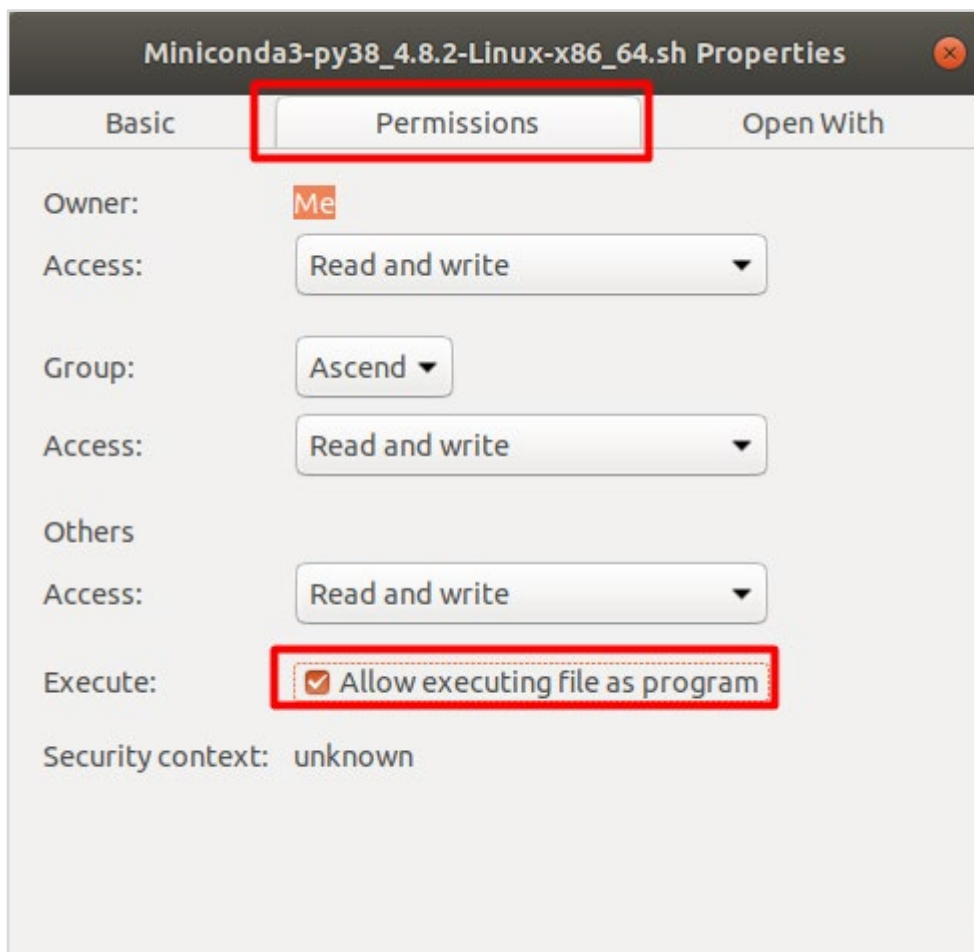


图2-2 文件执行权限配置界面（2）

步骤 2 在文件所在文件夹位置右击空白处，新建一个终端，如果使用普通用户登录，需要输入以下命令切换到 bash 模式。

```
bash
```

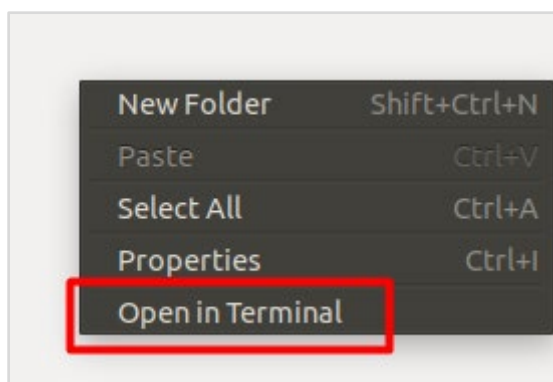


图2-3 新建终端界面

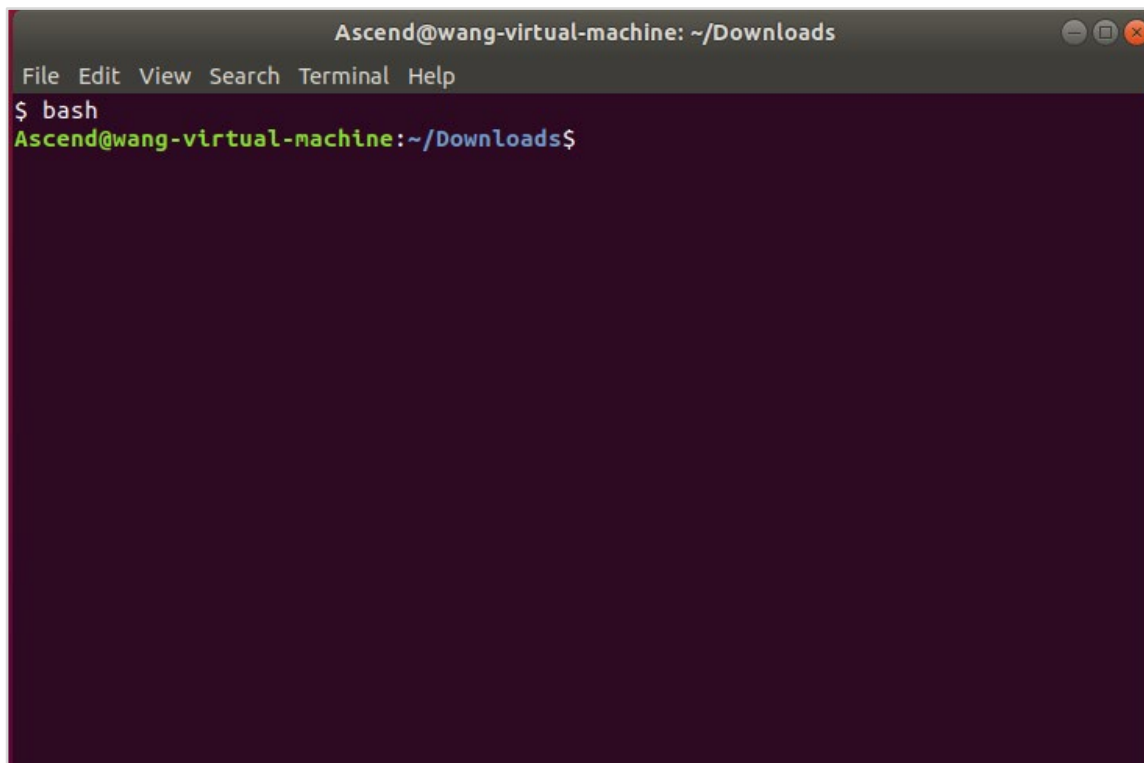


图2-4 切换终端到 bash 模式

步骤 3 输入以下命令执行安装文件，红色部分以实际下载文件名称为准：

```
./Miniconda3-py38_4.8.2-Linux-x86_64.sh
```

步骤 4 安装过程需要同意安装协议，默认为 no，需要手动输入 yes。

```
Ascend@wang-virtual-machine: ~/Downloads
File Edit View Search Terminal Help

Anaconda has self-classified this software as Export Commodity Control Number (ECCN) 5D992.c, which includes mass market information security software using or performing cryptographic functions with asymmetric algorithms. No license is required for export of this software to non-embargoed countries. The Intel Math Kernel Library contained in Anaconda Individual Edition is classified by Intel as ECCN 5D992.c with no license required for export to non-embargoed countries.

The following packages are included in this distribution that relate to cryptography:

...skipping 23 lines
libsodium
    A software library for encryption, decryption, signatures, password hashing and more.

pynacl
    A Python binding to the Networking and Cryptography library, a crypto library with the stated goal of improving usability, security and speed.

Last updated February 25, 2020

Do you accept the license terms? [yes|no]
[no] >>> yes
```

图2-5 同意安装协议界面

步骤 5 安装结束需要输入 yes 初始化 Miniconda。

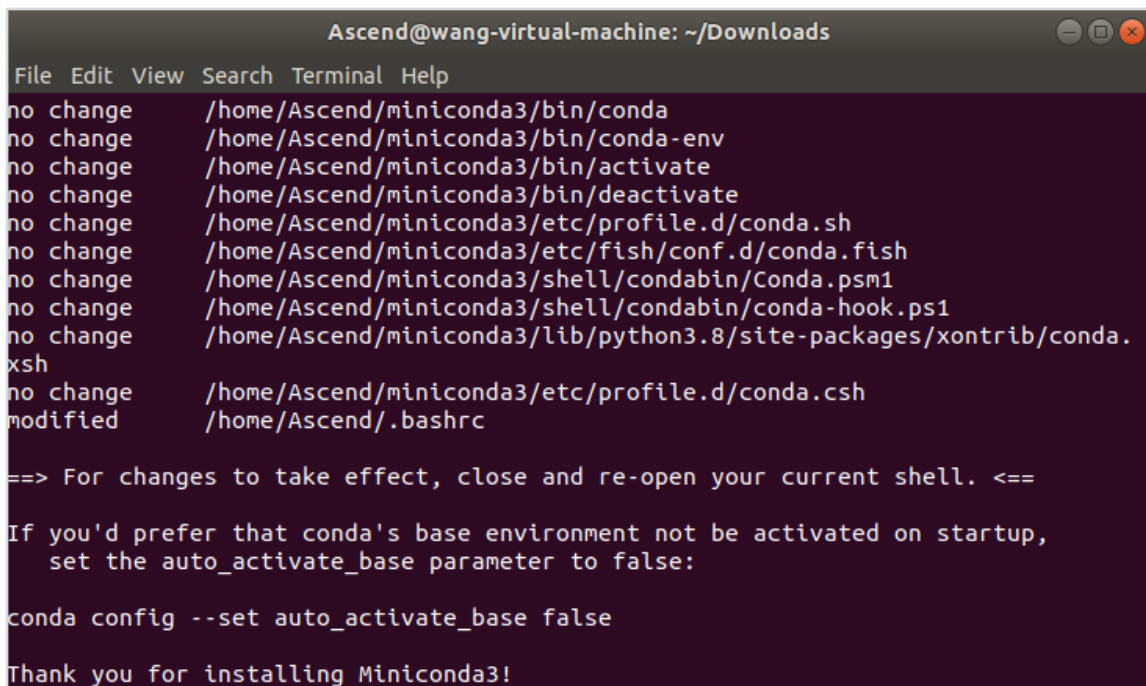
```
Ascend@wang-virtual-machine: ~/Downloads
File Edit View Search Terminal Help

pyopenssl          pkgs/main/linux-64::pyopenssl-19.1.0-py38_0
pysocks            pkgs/main/linux-64::pysocks-1.7.1-py38_0
python             pkgs/main/linux-64::python-3.8.1-h0371630_1
readline          pkgs/main/linux-64::readline-7.0-h7b6447c_5
requests          pkgs/main/linux-64::requests-2.22.0-py38_1
ruamel_yaml       pkgs/main/linux-64::ruamel_yaml-0.15.87-py38h7b6447c_0
setuptools        pkgs/main/linux-64::setuptools-45.2.0-py38_0
six               pkgs/main/linux-64::six-1.14.0-py38_0
sqlite            pkgs/main/linux-64::sqlite-3.31.1-h7b6447c_0
tk                pkgs/main/linux-64::tk-8.6.8-hbc83047_0
tqdm              pkgs/main/noarch::tqdm-4.42.1-py_0
urllib3           pkgs/main/linux-64::urllib3-1.25.8-py38_0
wheel             pkgs/main/linux-64::wheel-0.34.2-py38_0
xz                pkgs/main/linux-64::xz-5.2.4-h14c3975_4
yaml              pkgs/main/linux-64::yaml-0.1.7-had09818_2
zlib              pkgs/main/linux-64::zlib-1.2.11-h7b6447c_3

Preparing transaction: done
Executing transaction: done
installation finished.
Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>> yes
```

图2-6 Miniconda 初始化确认界面

步骤 6 完成以上操作后 Miniconda 安装成功，需要关闭当前终端，新建一个终端完成后续操作。



```
Ascend@wang-virtual-machine: ~/Downloads
File Edit View Search Terminal Help
no change /home/Ascend/miniconda3/bin/conda
no change /home/Ascend/miniconda3/bin/conda-env
no change /home/Ascend/miniconda3/bin/activate
no change /home/Ascend/miniconda3/bin/deactivate
no change /home/Ascend/miniconda3/etc/profile.d/conda.sh
no change /home/Ascend/miniconda3/etc/fish/conf.d/conda.fish
no change /home/Ascend/miniconda3/shell/condabin/Conda.psm1
no change /home/Ascend/miniconda3/shell/condabin/conda-hook.ps1
no change /home/Ascend/miniconda3/lib/python3.8/site-packages/xontrib/conda.
xsh
no change /home/Ascend/miniconda3/etc/profile.d/conda.csh
modified /home/Ascend/.bashrc

==> For changes to take effect, close and re-open your current shell. <==

If you'd prefer that conda's base environment not be activated on startup,
set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Miniconda3!
```

图2-7 Miniconda 安装完成界面

## 2.2 创建虚拟环境

步骤 1 打开命令行窗口之后，输入以下命令创建虚拟环境，因为 MindSpore 和 TensorFlow 都需要大量的依赖包，如果安装在同一个环境可能会出现冲突，所以需要为不同的框架创建不同的虚拟环境，输入以下命令为 MindSpore 创建虚拟环境，指定下载版本节省下载时间，创建过程需要输入 y 确认。

```
conda create -n MindSpore python==3.7.5
```

步骤 2 输入以下命令可以激活对应虚拟环境：

```
conda activate MindSpore
```

## 2.3 pip 换源

python 可以通过 pip 和 conda 两种方式来安装包，但是两者所安装的包并不完全兼容，在实际使用过程中建议只选择一种方式来安装包，本实验使用的是 pip。

步骤 1 新建一个终端，然后逐行输入以下命令，更新索引，安装 vim。

```
sudo apt-get update
```



```
sudo apt-get install vim
```

步骤 2 依次输入以下命令，创建并编辑 pip 配置文件：

```
mkdir ~/.pip/  
touch ~/.pip/pip.conf  
vim ~/.pip/pip.conf
```

步骤 3 按 i 键进入编辑模式，然后将以下内容拷贝到文件中，随后按 Esc 键，输入 :wq 按回车键进行保存并退出的操作：

```
[global]  
index-url = https://mirrors.huaweicloud.com/repository/pypi/simple  
trusted-host = mirrors.huaweicloud.com  
timeout = 120
```

步骤 4 更多 python 换源的内容可参考以下链接：

```
https://mirrors.huaweicloud.com/
```



图2-8 华为开源镜像站界面

## 2.4 安装 MindSpore

步骤 1 新建一个命令行窗口，激活已经创建好的虚拟环境（参考 1.5 节），使用 activate 命令激活该虚拟环境。

```
activate MindSpore
```

步骤 2 输入以下命令安装 MindSpore 1.3 版本，因为版本更新较快，可以参考官网安装不同的版本：<https://www.mindspore.cn/install>。

```
pip install https://ms-release.obs.cn-north-4.myhuaweicloud.com/1.3.0/MindSpore/cpu/x86_64/mindspore-1.3.0-cp37-cp37m-linux_x86_64.whl --trusted-host ms-release.obs.cn-north-4.myhuaweicloud.com -i https://pypi.tuna.tsinghua.edu.cn/simple
```

步骤 3 安装成功后输入 python，在命令行中进入开发环境，输入以下命令导入 MindSpore，如果没有报错则安装成功。

注意：由于 GLIBC 包的依赖问题，ubuntu16.04 系统不支持 MindSpore 需要 ubuntu18 以上的版本。

```
import mindspore
```

# 3 华为云使用指导手册

## 3.1 实验介绍

### 3.1.1 关于本实验

华为云提供了非常丰富的云资源以及云服务，可以满足开发者的需要。本手册主要聚焦于如何申请华为云上的相应资源，主要包括了：账号申请与实名认证，AK 和 SK 的获取，华为云主要服务介绍。

### 3.1.2 实验目的

- 理解人工智能开发环境的具体内容。
- 申请人工智能开发环境。
- 基于搭建的环境进行功能测试。

### 3.1.3 实验所需模块

华为云平台：<http://www.huaweicloud.com/>

## 3.2 华为云账号申请与实名认证

### 3.2.1 账户用途

- 用于在华为云官网开通相应服务，确保课程实验部分顺利进行。
- 用于体验华为云提供的各种便捷便宜的云服务，包括深度学习，机器学习，图像识别，语音识别，自然语言处理等。

### 3.2.2 注册步骤

步骤 1 进入华为云官网；

进入华为云官网：<http://www.huaweicloud.com/>，如下图所示：



图3-1 华为云官网

## 步骤 2 点击首页右上角“注册”

点击右上角的“注册”按钮，进行账号注册，结果如下：

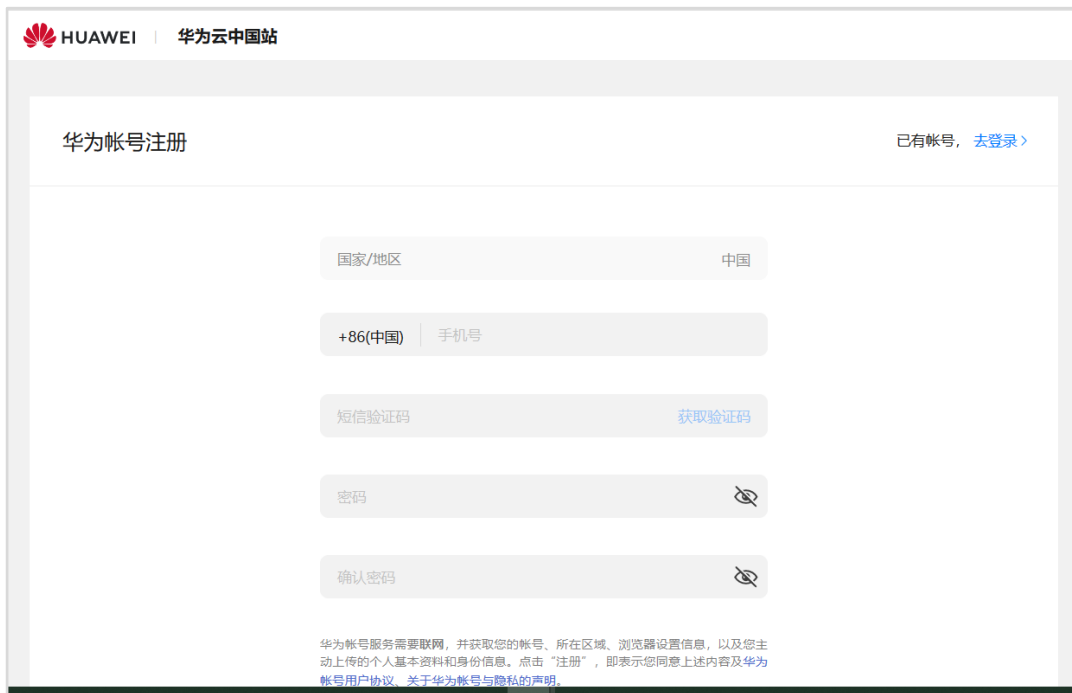


图3-2 注册账号界面

## 步骤 3 实名认证

选择“账号中心”进入下一项，点击“实名认证”按钮进行账户的实名认证，：



图3-3 完成实名认证界面

## 3.3 对象存储服务 OBS

### 3.3.1 简介

对象存储服务（Object Storage Service）是一款稳定、安全、高效、易用的云存储服务，具备标准 Restful API 接口，可存储任意数量和形式的非结构化数据。

### 3.3.2 创建 obs 桶

步骤 1 打开华为云官网并点击 OBS 服务

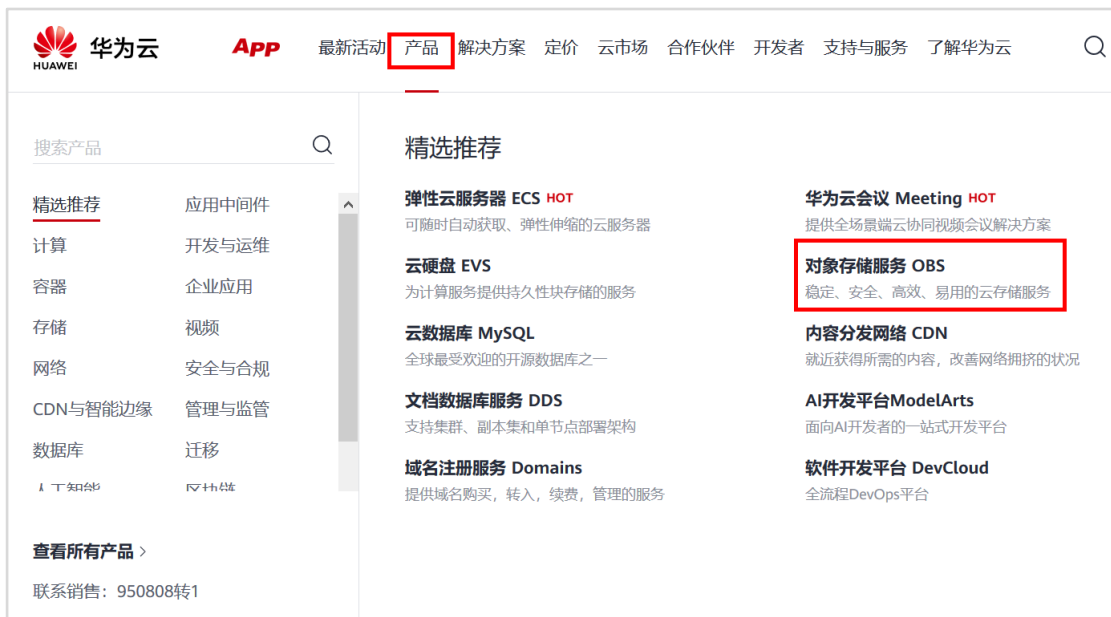


图3-4 产品选择界面

OBS 服务是华为云中最常用的服务，通常都会搭配 ModelArts 服务一起使用。

步骤 2 点击管理控制台，进入服务。

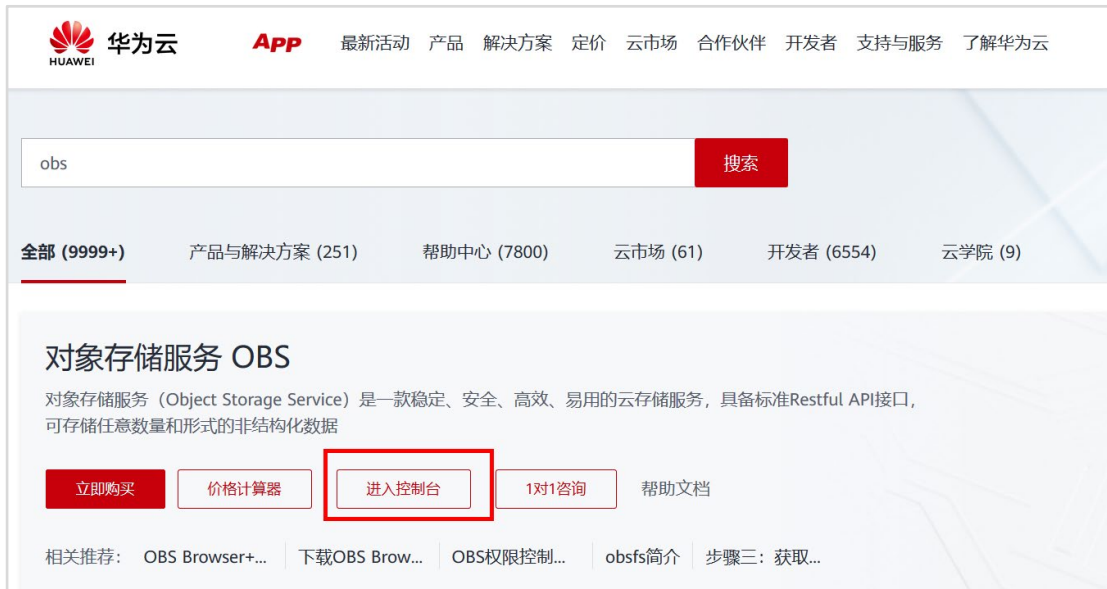


图3-5 选择进入 OBS 控制台界面

步骤 3 点击右上角的创建桶，来创建自己的 OBS 桶。

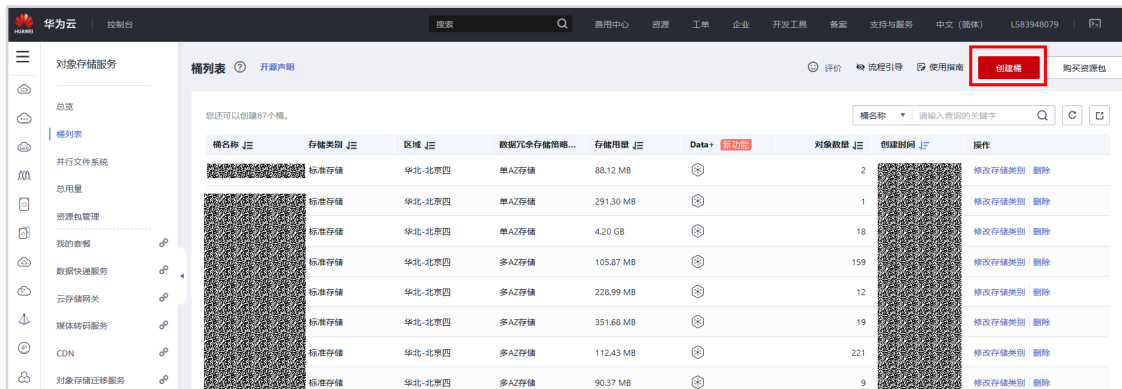


图3-6 创建 OBS 桶界面

步骤 4 桶的具体配置情况，可以由自己设定，区域自行选择，建议选择“北京四”，桶名称自定义，储存类别为标准储存。桶策略请自行定义，如果需要别人也读取你的文件，请选择“公共读”，如果只供自己使用，建议选择“私有”，之后点击“立即创建”。



图3-7 选择 OBS 桶配置界面

步骤 5 之后，点击进入新创的桶，点击“对象”，进行数据的存储。



图3-8 选择侧边栏对象界面

步骤 6 接下来就可以上传文件或者创建新的文件夹。



图3-9 在 OBS 对象栏中进行上传等操作

### 3.3.3 计费模式

OBS 提供按需计费和包年包月两种计费模式，用户可以根据实际需求选择资源的计费模式。

- 按需计费

OBS 服务默认为按需计费模式，即按实际使用的时长收费，以小时为单位，每小时整点结算，不设最低消费标准。

- 包年包月

您也可以通过包年包月（资源包）的方式，提前购买使用额度和时长，获取更多的优惠。超出资源包的部分自动转按需计费。

建议使用按需计费，如果停止使用请及时删除对象以及数据桶，以免造成不必要的扣费。

## 3.4 ModelArts

ModelArts 是面向开发者的一站式 AI 开发平台，为机器学习与深度学习提供海量数据预处理及半自动化标注、大规模分布式 Training、自动化模型生成，及端-边-云模型按需部署能力，帮助用户快速创建和部署模型，管理全周期 AI 工作流。

步骤 1 在“EI 企业智能”中选择“AI 开发平台 ModelArts”。



图3-10 选择 AI 开发平台 ModelArts

步骤 2 进入管理控制台，也可在“帮助文档”中查看详细介绍。



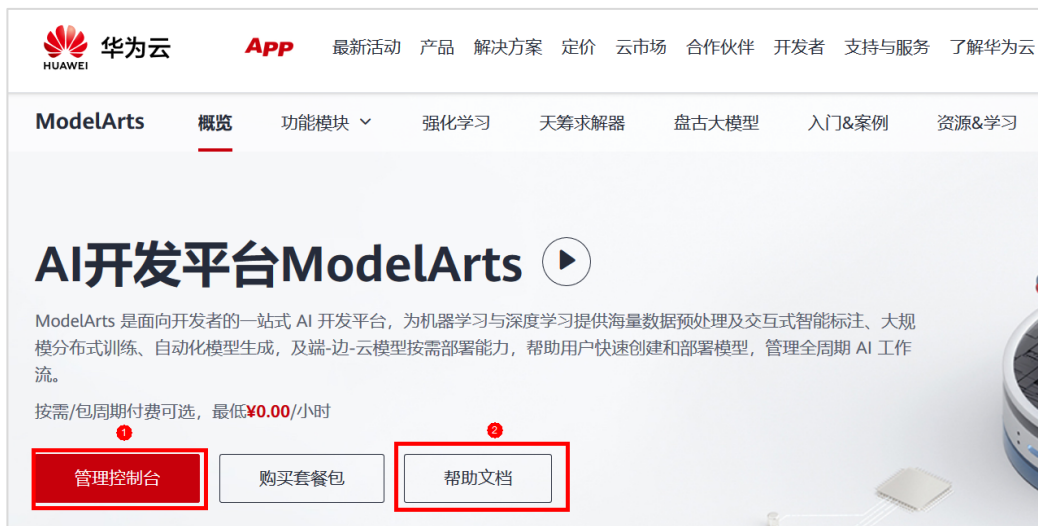


图3-11 进入管理控制台

步骤 3 进入“全局配置”，进行访问授权，并点击“访问授权”，如果委托对象无内容，可选择“自动创建”，最后点击“同意授权”。



图3-12 访问授权（1）



图3-13 访问授权（2）

## 3.4.2 自动学习服务

步骤 1 进入自动学习服务。



图3-14 授权提示界面

如果有授权提示，可以点击上面的蓝色字体“此处”进行对应的授权操作。

步骤 2 选择自动学习的不同场景。

自动学习目前有如下场景可供选择：

- 图像分类
- 物体检测
- 预测分析
- 声音分类
- 文本分类

具体学习任务可以查看对应实验手册。

## 3.5 notebook 开发

步骤 1 进入 notebook 开发界面，并进行 notebook 的创建。

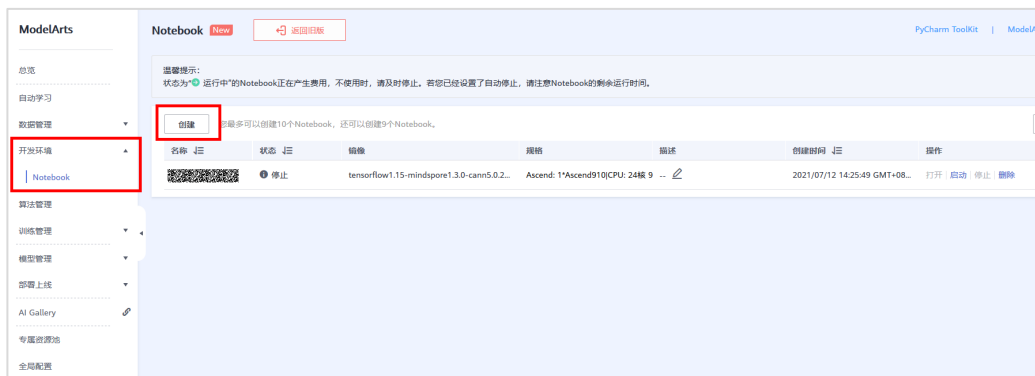


图3-15 开发环境 Notebook 列表界面

步骤 2 选择本次认证课程所需要的 notebook 配置-MindSpore1.3.0。

\* 名称: notebook-c3fd

描述: 0/256

\* 自动停止: ☒ 开启该选项后, 该Notebook实例将在运行时长超出您所选择的时长后, 自动停止。

☒ 1小时 ☐ 2小时 ☐ 4小时 ☐ 6小时 ☐ 自定义

图3-16 创建 notebook

- 在“名称”处填写自己的名称。
- 在“自动停止”处选择对应的服务时长，建议勾选，防止产生不必要的扣费。

公共镜像		自定义镜像
名称	描述	
<input type="radio"/> 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	
<input type="radio"/> 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	
<input type="radio"/> 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	
<input type="radio"/> 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	
<input type="radio"/> mlstudio-pyspark2.3.2-ubuntu16.04	CPU算法开发和训练基础镜像, 包含可以图形化机器学习算法开发和调测MLStudio...	
<input type="radio"/> mindstudio3.0.2-ascend910-cann5.0.2.1-ubuntu18.04-aarch64	Ascend算子开发基础镜像, 预置专业级算子开发工具MindStudio, 仅支持SSH链接	
<input checked="" type="radio"/> tensorflow1.15-mindspore1.3.0-cann5.0.2-euler2.8-aarch64	Ascend+ARM算法开发和训练基础镜像, AI引擎预置TensorFlow和MindSpore	
<input type="radio"/> modelbox1.0.9.2-tensorrt7.1.3-pytorch1.8.1-cuda10.2-cudnn8-eu	AI应用开发基础镜像, 预置AI应用编排引擎ModelBox、AI引擎PyTorch、TensorRT...	
<input type="radio"/> 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。	
<input type="radio"/> cylv0.91.4-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cylv, cbcpy, ortools及cplex。	

图3-17 选择 MindSpore1.3.0

★ 资源池 **公共资源池** 专属资源池

★ 类型 **ASCEND**

★ 规格 Ascend: 1\*Ascend910|CPU: 24核 96GB  
昇腾910(32GB显存)单卡规格, 搭配ARM处理器, 适合深度学习场景下的模型训练和调测

★ 存储配置 **默认存储** 云硬盘EVS 弹性文件服务SFS  
针对探索、实验等非正式生产场景, 提供免费的50GB共享网络存储

SSH远程开发 ☐

图3-18 其他配置选择

步骤 3 进行 notebook 的创建, 并提交。

产品名称	产品规格	计费模式	价格
	描述	--	
	自动停止	1小时	
	镜像	tensorflow1.15-mindspore1.3.0-cann5.0.2-euler2.8-aarch64	
	资源池	公共资源池	
notebook-88d7	规格	Ascend: 1*Ascend910 CPU: 24核 96GB	按需计费 Notebook: ¥ 19.5/小时
	存储配置	默认存储	
	存储空间	50 GB	
	SSH远程开发	--	
	远程访问白名单	--	

配置费用: **¥19.50/小时**  
优先扣减免费套餐用量, [了解更多](#)

上一步 提交

图3-19 创建 notebook

**注意:** notebook 资源在停止使用时, 请务必选择**停止使用**, 否则会产生不必要的扣费。

Notebook **New** [+3 返回旧版](#) PyCharm ToolKit | ModelArts SDK

温馨提示:  
状态为“运行中”的Notebook正在产生费用, 不使用时, 请及时停止。若您已设置自动停止, 请注意Notebook的剩余运行时间。

**创建** 您最多可以创建10个Notebook, 还可以创建9个Notebook.

名称	状态	镜像	规格	描述	创建时间	操作
notebook-1e03	停止	tensorflow1.15-mindspore1.3.0-cann5.0.2-	Ascend: 1*Ascend910 CPU: 24核 96GB		2021/07/12 14:25:49 GMT+08...	打开 启动 <b>停止</b> 删除

图3-20 停止服务

# 4 实验环境恢复

## 4.1 实验环境恢复

### 4.1.1 本地环境恢复

本认证用到的本地环境不需要恢复，如果本地有多个开发环境，建议使用 Anaconda 虚拟环境来保证环境纯净独立，避免依赖包版本不对的问题。

### 4.1.2 云端环境恢复

本认证使用的 ModelArts 云端环境主要需要启动环境来恢复原有时间环境。当不使用时，需要停止环境或删除环境。创建环境的步骤在 3.5 已经有描述。

#### 步骤 1 恢复环境

点击已有环境中的“启动”按钮，恢复实验环境。

名称	状态	镜像	规格	描述	创建时间	操作
notebook-cef8	停止	tensorflow1.15-mindspore1.3.0-cann5.0.2-euler2.8-aarch64	Ascend: 1*Ascend910(CPU: 24核 96GB)	-- 编辑	2021/09/26 10:07:22 GMT+08:00	打开 启动 停止 更多
notebook-98e0	停止	mindspore1.2.0-openmpi2.1.1-ubuntu18.04	CPU: 2核 8GB	-- 编辑	2021/09/23 17:38:57 GMT+08:00	打开 启动 停止 更多
notebook-4524	停止	tensorflow2.1-cuda10.1-cudnn7-ubuntu18.04	GPU: 1*V100(32GB)/CPU: 8核 64GB	-- 编辑	2021/08/09 16:34:05 GMT+08:00	打开 启动 停止 更多
notebook-9af0	停止	tensorflow2.1-cuda10.1-cudnn7-ubuntu18.04	CPU: 2核 8GB	-- 编辑	2021/08/09 15:53:50 GMT+08:00	打开 启动 停止 更多

图4-1 恢复环境

#### 步骤 2 自动停止设置

建议在使用时设置自动停止时间，这可以有效避免超时扣费。根据自身训练时长需要，设定该实例的自动停止时间。



图4-2 自动停止设置

#### 步骤 3 延长自动停止时间

在代码编辑界面的右上角有按钮可以实现自动停止时间的设置，可以延长时间。

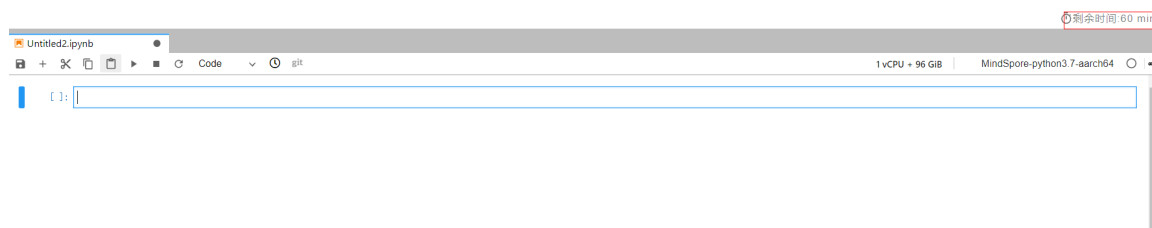


图4-3 延长自动停止时间

手动输入需要延长的时间。

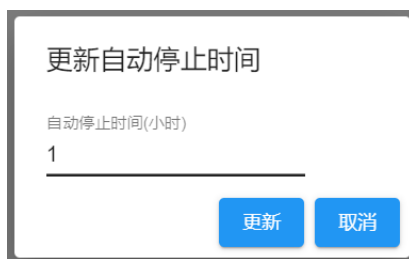


图4-4 延长时间设置

#### 步骤 4 关闭和停止环境

点击“停止”按钮可以停止实例。也可以点击“更多”按钮中的“删除”来删除环境。

名称	状态	镜像	规格	描述	创建时间	操作
notebook-cell8	运行中 (56分钟后自动停止...)	tensorflow1.15-mindspore1.3.0-cann5.0.2-euler2.8-aarch64	Ascend: 1*Ascend910(CPU: 24核 96GB)	--	2021/09/26 10:07:22 GMT+08:00	打开   启动   停止   更多
notebook-98e9	停止	mindspore1.2.0-openmpi2.1.1-ubuntu18.04	CPU: 2核 8GB	--	2021/09/23 17:38:57 GMT+08:00	打开   启动   停止   更多
notebook-4524	停止	tensorflow2.1-cuda10.1-cudnn7-ubuntu18.04	GPU: 1*V100(32GB)(CPU: 8核 64GB)	--	2021/08/09 16:34:05 GMT+08:00	打开   启动   停止   更多
notebook-9afd	停止	tensorflow2.1-cuda10.1-cudnn7-ubuntu18.04	CPU: 2核 8GB	--	2021/08/09 15:53:50 GMT+08:00	打开   启动   停止   更多

图4-5 停止环境

名称	状态	镜像	规格	描述	创建时间	操作
notebook-cell8	运行中 (56分钟后自动停止...)	tensorflow1.15-mindspore1.3.0-cann5.0.2-euler2.8-aarch64	Ascend: 1*Ascend910(CPU: 24核 96GB)	--	2021/09/26 10:07:22 GMT+08:00	打开   启动   停止   更多
notebook-98e9	停止	mindspore1.2.0-openmpi2.1.1-ubuntu18.04	CPU: 2核 8GB	--	2021/09/23 17:38:57 GMT+08:00	打开   启动   停止   更多
notebook-4524	停止	tensorflow2.1-cuda10.1-cudnn7-ubuntu18.04	GPU: 1*V100(32GB)(CPU: 8核 64GB)	--	2021/08/09 16:34:05 GMT+08:00	打开   启动   停止   更多
notebook-9afd	停止	tensorflow2.1-cuda10.1-cudnn7-ubuntu18.04	CPU: 2核 8GB	--	2021/08/09 15:53:50 GMT+08:00	打开   启动   停止   更多

图4-6 删除环境