CodeArts IDE Online

最佳实践

文档版本 01

发布日期 2025-07-23

华为云计算技术有限公司

版权所有 © 华为云计算技术有限公司 2025。 保留一切权利。

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

商标声明

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 最佳实践汇总 1
2 基于 CodeArts IDE Online 部署微服务到容器镜像服务 2
3 基于 CodeArts IDE Online 开发并发布 WeLink 应用 9
4 基于 CodeArts IDE Online 开发并使用模型 18
4.1 准备工作 18
4.2 导入和预处理训练数据集 20
4.3 开发深度学习模型 21

 版权所有 © 华为云计算技术有限公司

1 最佳实践汇总

本文汇总了基于CodeArts IDE Online常见应用场景的操作实践，为每个实践提供详细的方案描述和操作指导。

表 1-1 CodeArts IDE Online 最佳实践一览表

最佳实践	说明
基于CodeArts IDE Online部署微服务到容器镜像服务	本实践主要讲述在CodeArts IDE Online中如何快速将开发好的微服务部署到华为云容器服务并进行快速验证的过程。
基于CodeArts IDE Online开发并发布WeLink应用	本实践主要讲述基于CodeArts IDE Online快速开发、发布WeLink应用。
基于CodeArts IDE Online开发并使用模型	本实践主要讲解如何在CodeArts IDE Online中使用TensorFlow和Jupyter Notebook完成神经网络模型的训练，并利用该模型完成简单的图像分类。

版权所有 © 华为云计算技术有限公司

2 基于 CodeArts IDE Online 部署微服务到容器镜像服务

本章节主要介绍如何在CodeArts IDE Online中快速将开发好的微服务部署到华为云容器服务，并进行快速验证。

准备工作

- 登录到容器镜像服务（SWR）控制台。参考**创建组织**完成组织创建。
- 登录到**我的凭证**，在“项目列表”区域获取镜像仓库访问地址、区域项目名称。
  镜像仓库地址为：**swr.区域项目名称.myhuaweicloud.com**，如中国华北区4对应的镜像仓库地址为：**swr.cn-north-4.myhuaweicloud.com**。
- 获取访问密钥AK/SK，具体操作请参见**如何获取访问密钥AK/SK**。

操作步骤

步骤1 创建并启动IDE实例。

1. 登录到CodeArts IDE Online控制台，如果提示未开通则单击“OK”跳转至开通页面完成服务开通，成功登录后的页面如下：

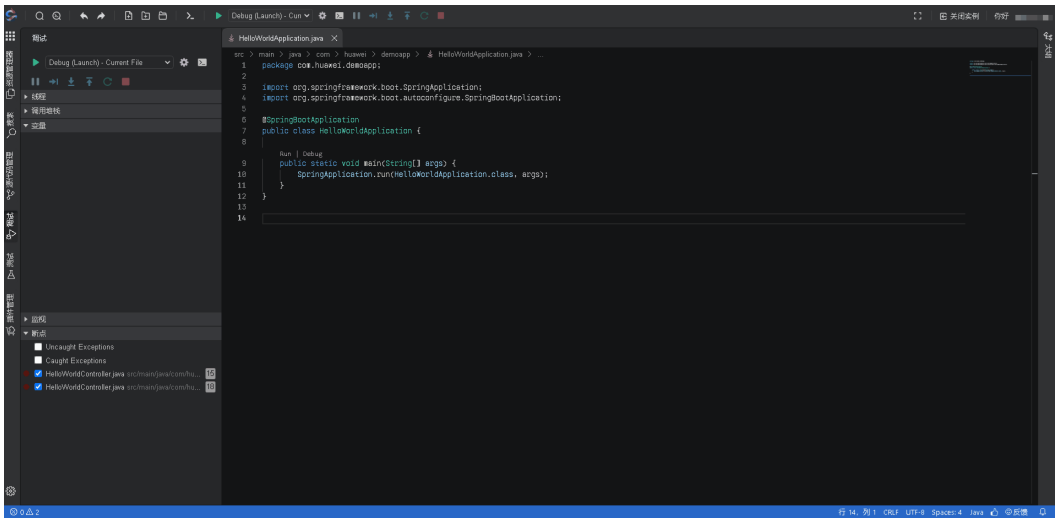
2. 单击“新建实例”，进入IDE实例模板配置页面并完成配置，已有实例并完成载入了项目工程则可以跳过该步骤。
   针对已有实例，则直接单击实例列表的启动按钮进入实例页面。

版权所有 © 华为云计算技术有限公司

步骤2 开发调试微服务代码

代码编写完成后，单击左侧Debug页签中的“Start Debug”或Main函数中的“Debug”启动工程，在代码文件行号前鼠标左键单击打上断点，完成监听端口的外部访问权开通即可边调试边在浏览器中查看代码修改的应用效果。

动图演示如下：

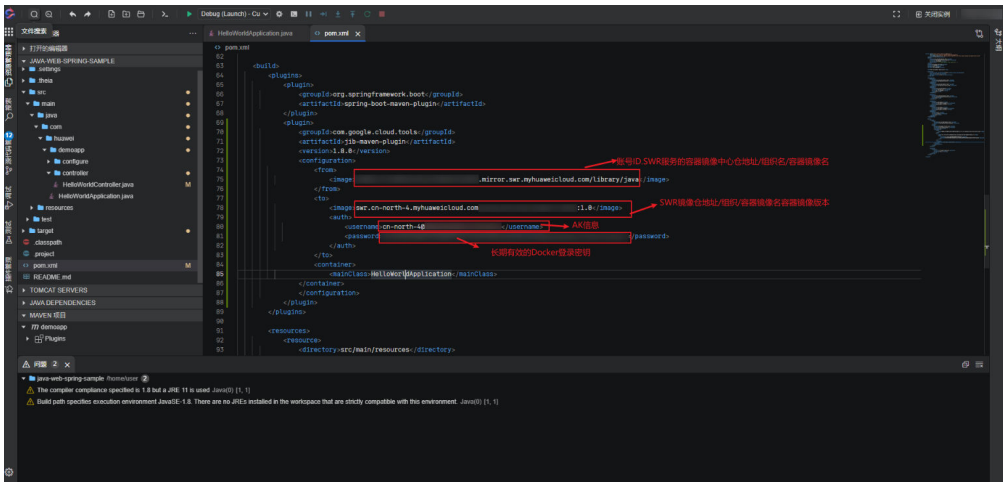


步骤3 制作微服务容器镜像

1. 编辑pom文件增加谷歌jib插件配置

   打开微服务工程中的“pom.xml”文件，在plugins标签下增加如下图内容：

版权所有 © 华为云计算技术有限公司



jib的maven配置格式：

```
<plugin>
    <groupId>com.google.cloud.tools</groupId>
    <artifactId>jib-maven-plugin</artifactId>
    <version>1.8.0</version>
    <configuration>
        <from>
            <image>用户账户ID.mirror.swr.myhuaweicloud.com/library/java</image>
        </from>
        <to>
            <image>swr.区域项目名称.myhuaweicloud.com/个人在SWR中创建的组织/springboot-
demo:1.0</image>
            <auth>
                <username>区域项目名称@申请的AK</username>
                <password>获取到的长期有效的Docker登录密钥</password>
            </auth>
        </to>
        <container>
            <mainClass>微服务入口类</mainClass>
        </container>
    </configuration>
</plugin>
```

其中用户账号ID、区域项目名称、AK/SK信息在**准备工作**获取。

- 长期有效的Docker登录密钥：可通过在Terminal中执行命令如下获取，*$AK*、*$SK*替换成**准备工作**章节获取到的真实AK/SK。

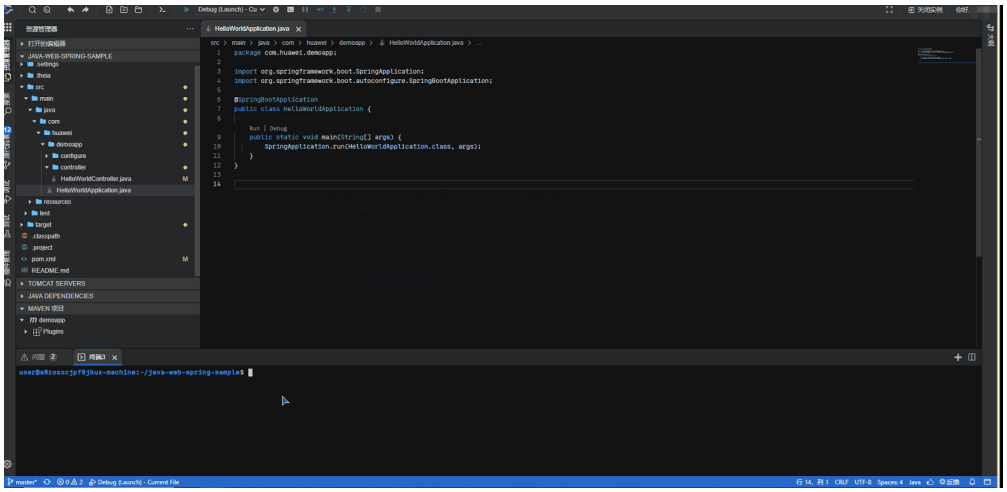
  ```
  printf "$AK" | openssl dgst -binary -sha256 -hmac "$SK" | od -An -vtx1 | sed 's/[ \n]//g' | sed
  'N;s/\n//'
  ```

- 个人在SWR中创建的组织：在**准备工作**已完成创建。
- “from”标签中的image标签内容为源镜像的地址，其中，“library”为容器镜像服务中镜像中心的组织名，该组织下分享了很多官方容器镜像，“java”为源镜像名称。

2. 编译打包并向SWR推送微服务镜像

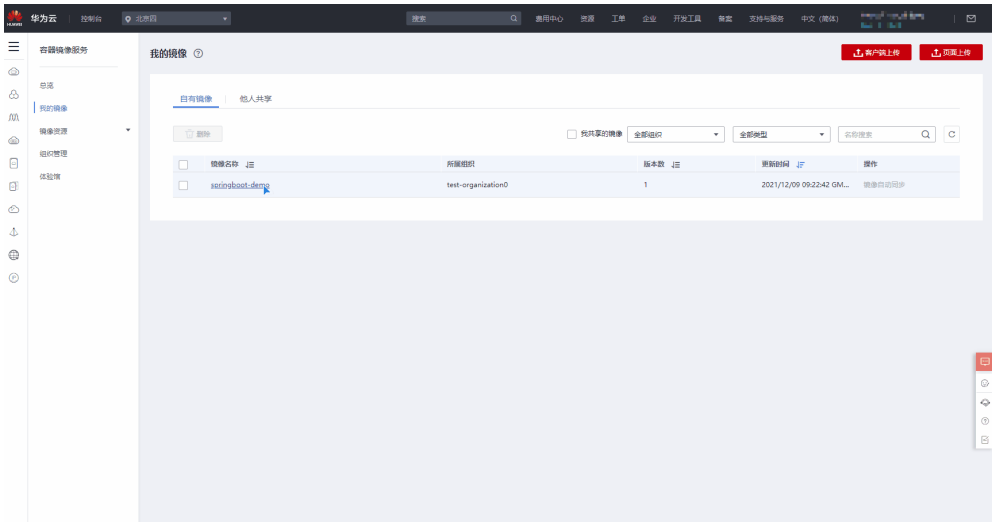
   完成jib配置后，在Terminal中输入“mvn package -Dmaven.test.skip=true jib:build”并回车，动图演示如下：

版权所有 © 华为云计算技术有限公司



3. 验证微服务镜像推送结果

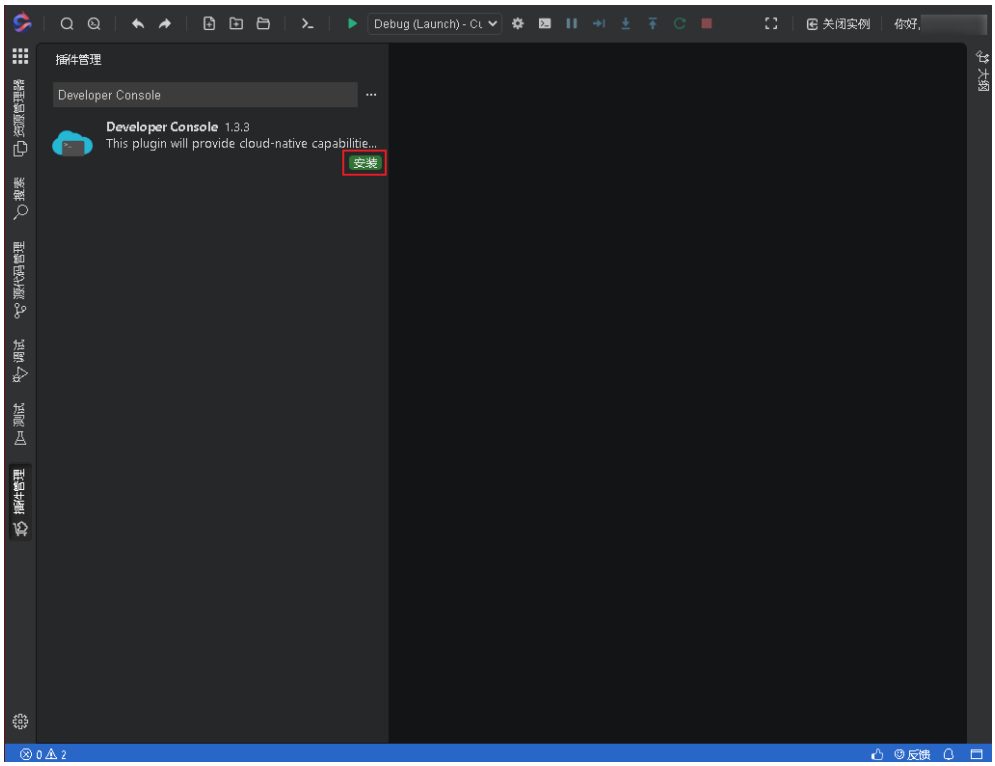
   登录SWR服务，验证查询容器镜像是否完成推送，动图演示如下：



步骤4 部署微服务容器镜像

1. 打开插件市场，搜索到“Developer Console”后，单击“安装”即可完成安装。

版权所有 © 华为云计算技术有限公司



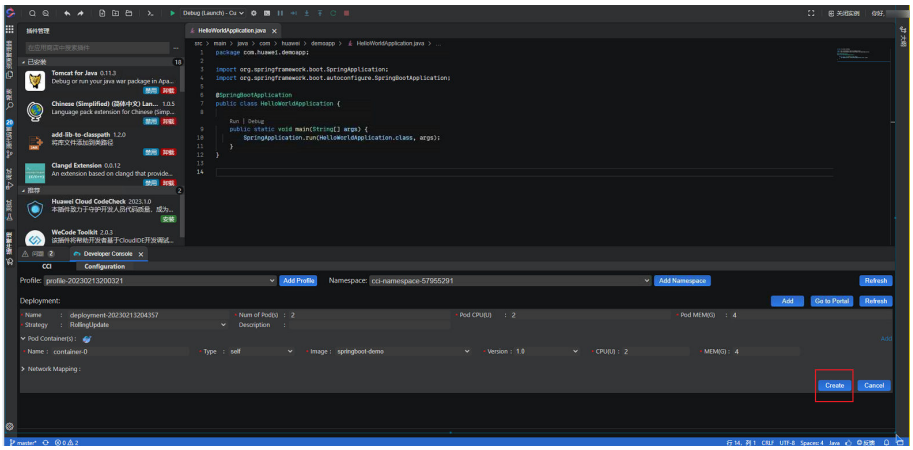
2. 部署微服务到云容器实例（ CCI ）

   a. 配置Profile

   Profile是一组API调用鉴权所需配置的抽象，Profile中的相关信息可通过访问**我的凭证**获取，通过如下信息就可以获取到镜像列表、vpc列表、cci负载/命名空间等数据。

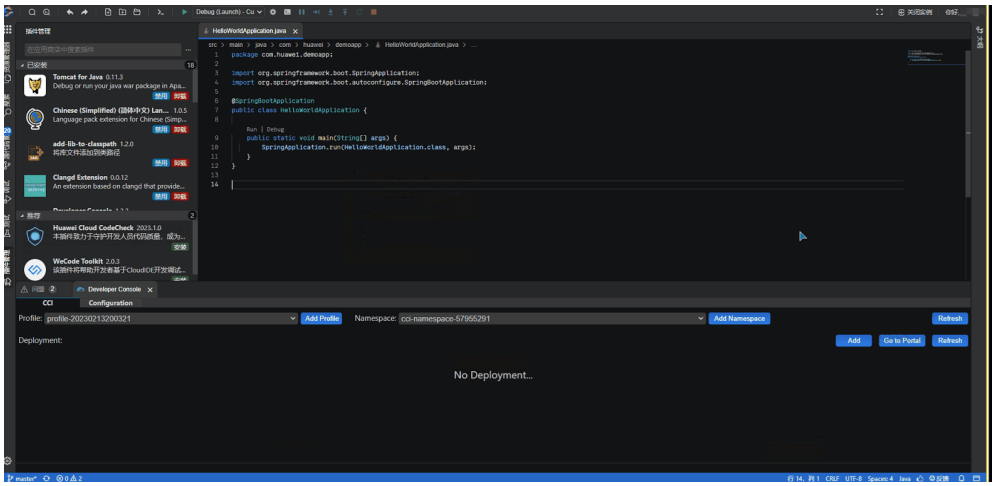
   - name：用于标识区分不同的Profile，只能输入大小写英文字母、下划线、中划线。
   - ProjectID：对应项目ID。
   - ProjectName：对应项目名称的英文。
   - Region：将控制台切换到区域项目所在region，网页URL中的region字段，如：cn-north-4。
   - UserID：用户ID。
   - UserName：用户名。
   - AK/SK：下载管理访问密钥页面中密钥excel文件，注意该文件的下载只有首次添加可以下载。

版权所有 © 华为云计算技术有限公司



3. 添加Namespace，具体请参见**命名空间**。

   选择不同的Profile，Namespace列表不尽相同，每一组profile对应不同的鉴权信息。



4. 创建Deployment

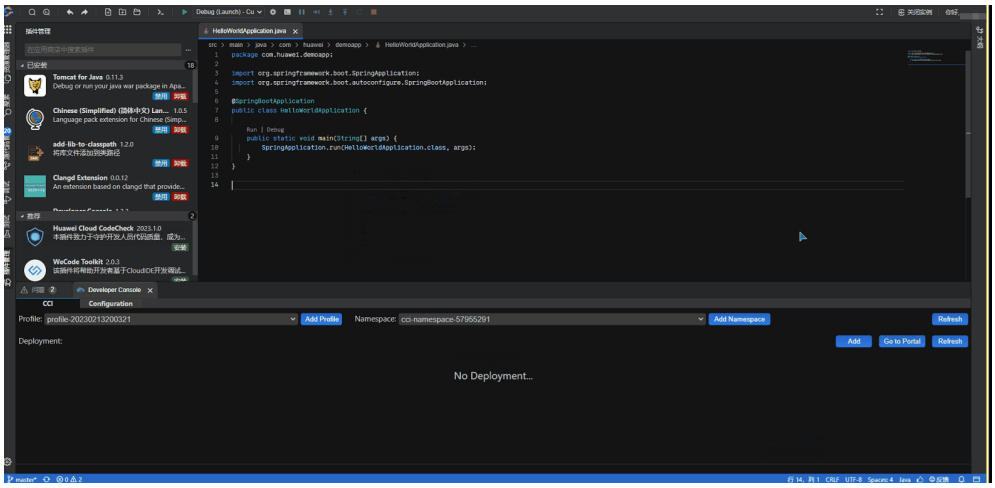
   a. 选择Profile和Namespace后，单击“Add”会出现deployment的配置输入，如果需要配置更多，则可以单击“Go to Portal”跳转至CCI页面进行创建。

      - 如果Pod中的镜像启动需要监听端口并且需要开放外部访问权时，则需要增加容器端口跟elb端口的映射关系，其中elb的端口一定要填写未被占用的。
      - 如果Pod中的镜像启动无监听端口或者不需要开发外部访问权时，则网络映射配置中的“Type”选择“Not enable”。

   b. 创建完成后，单击“Upgrade”将在IDE中直接升级负载，放弃升级则单击“Cancel”，如果希望跳转到CCI页面进行升级，则只需要单击与“Upgrade”平级的“Go to Portal”即可。

   c. 单击“Delete”，当前deployment则会被立即删除并附带删除配置的service数据。

版权所有 © 华为云计算技术有限公司



----结束

版权所有 © 华为云计算技术有限公司

3 基于 CodeArts IDE Online 开发并发布 WeLink 应用

本实践主要介绍如何在CodeArts IDE Online中快速开发、发布WeLink应用。

操作步骤

步骤1 注册和安装WeLink。

在手机应用市场上搜索找到“WeLink”并安装。

步骤2 创建Welink应用。

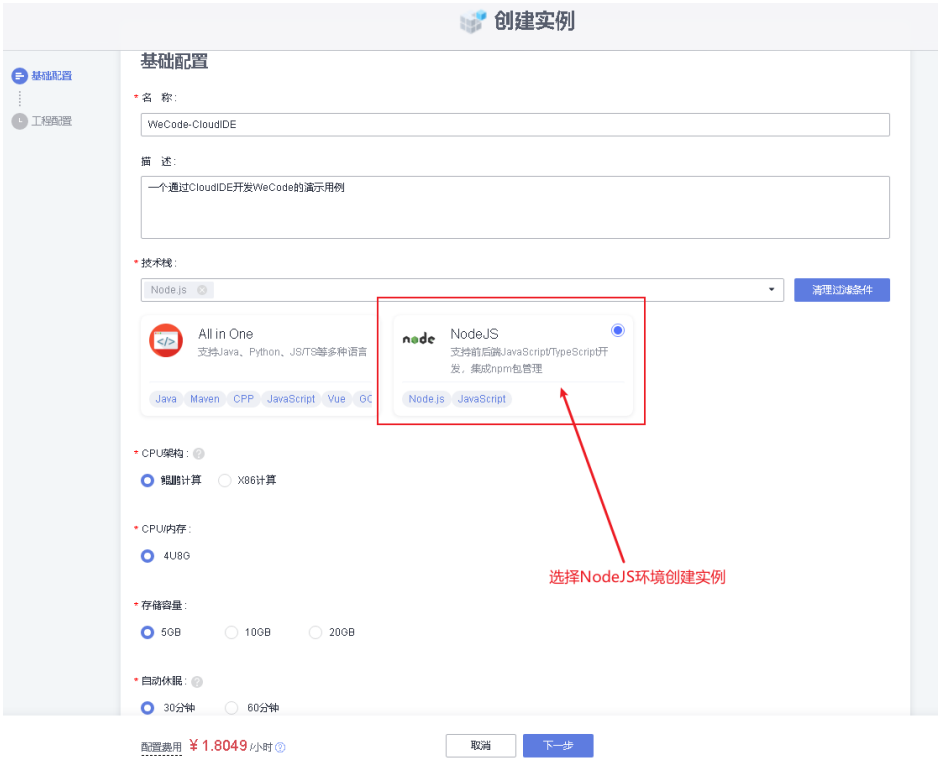
登录**WeLink开发者后台**，创建一个自己的应用，详细可参考**We码小程序**。

版权所有 © 华为云计算技术有限公司

图 3-1 WeLink 开发者后台

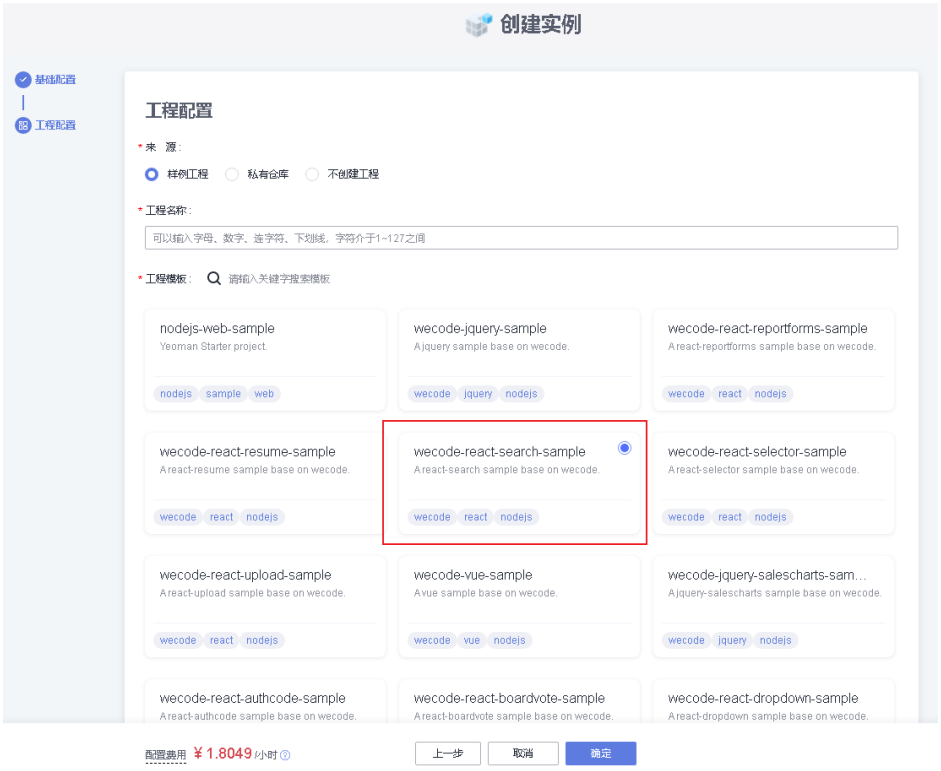
步骤3 准备We码小程序开发环境。

1. 登录到CodeArts IDE Online控制台，单击“新建实例”。
   如果提示未开通则根据提示跳转至开通页面完成服务开通。
2. 在“基础配置”页面，填写相关信息（其中，“技术栈”项选择“NodeJS”），单击“下一步”。



3. 进入“工程配置”页面，选择we码的模板工程（模板名以wecode开头）。

版权所有 © 华为云计算技术有限公司



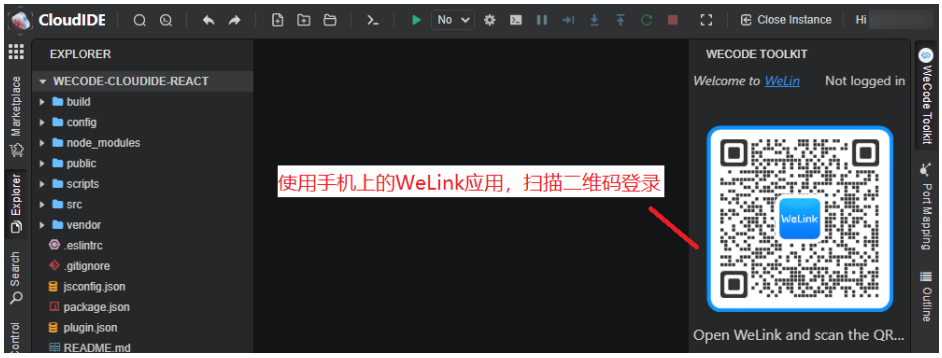
4. 配置完成后，单击“确定”，实例开始启动，浏览器会自动跳转到IDE实例。

步骤4 开发、预览We码小程序。

1. 登录WeLink。
   a. IDE实例启动后，会自动加载“WECODE TOOLKIT”插件，第一次启动时会要求用户登录，使用手机WeLink扫描右上角二维码即可登录。

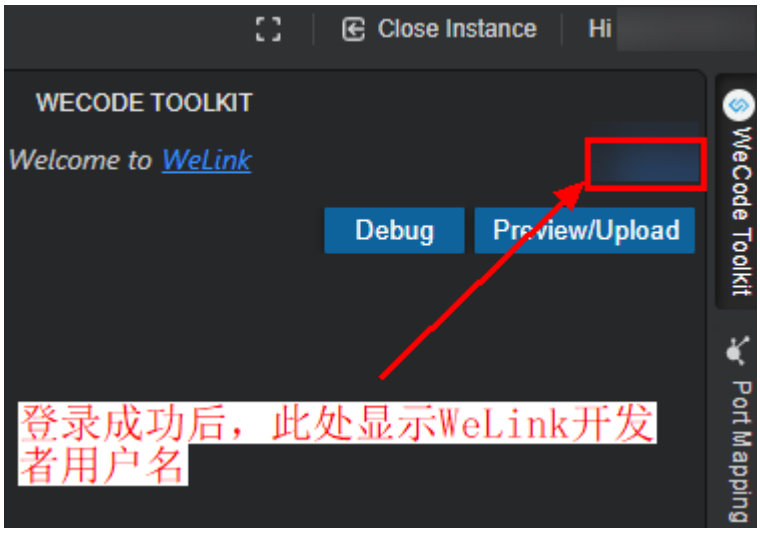
      说明

      在手机上安装WeLink后，使用WeLink扫描二维码完成WeLink账号登录，扫码之后单击“允许登录”。



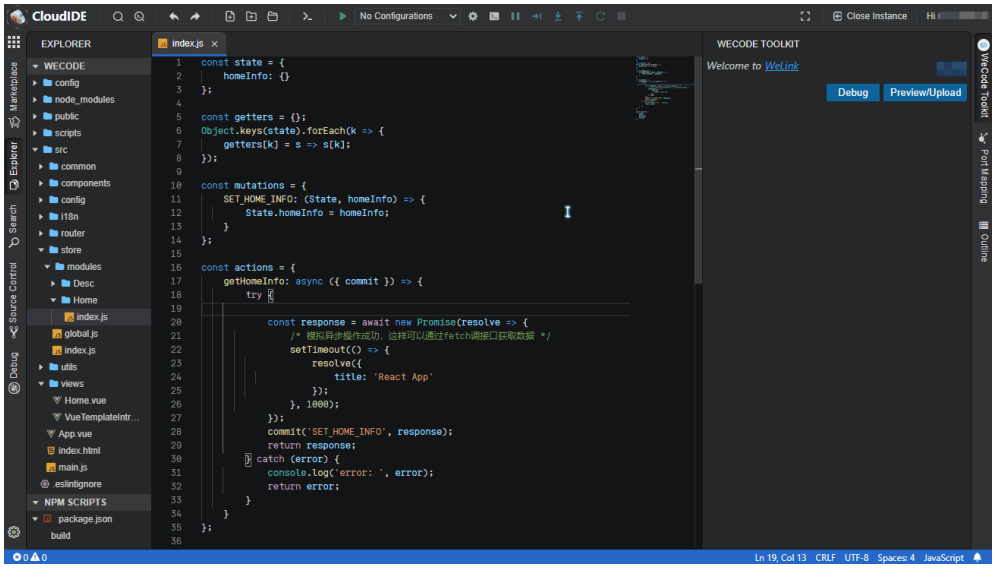
   b. 登录成功后，WeCode Toolkit插件显示WeLink开发者注册名。

版权所有 © 华为云计算技术有限公司



2. 编写程序代码

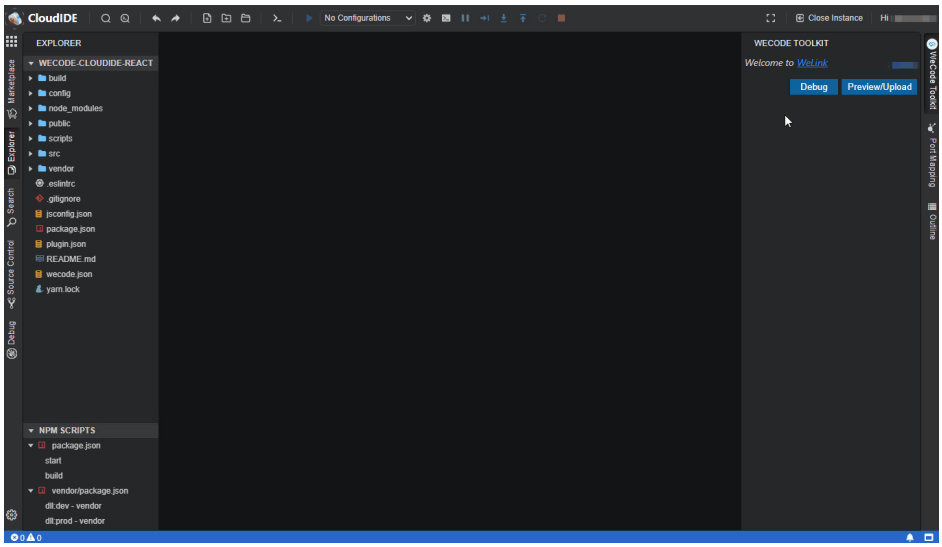
   使用CodeArts IDE Online实例可以编写程序逻辑代码，实现代码编写、语法高亮、自动补齐、语法校验、命令行访问等各种操作。



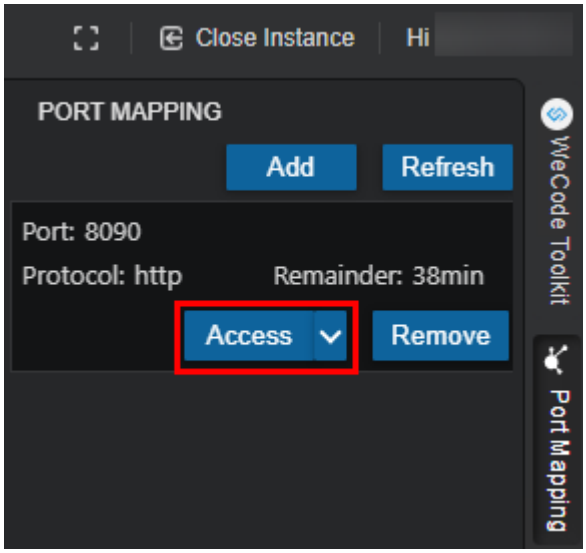
3. 调试代码

   a. 单击WeCode Toolkit插件上的“Debug”，等待WeCode应用完成自动编译。编译完成后，自动完成监听端口外部访问权的注册。

版权所有 © 华为云计算技术有限公司



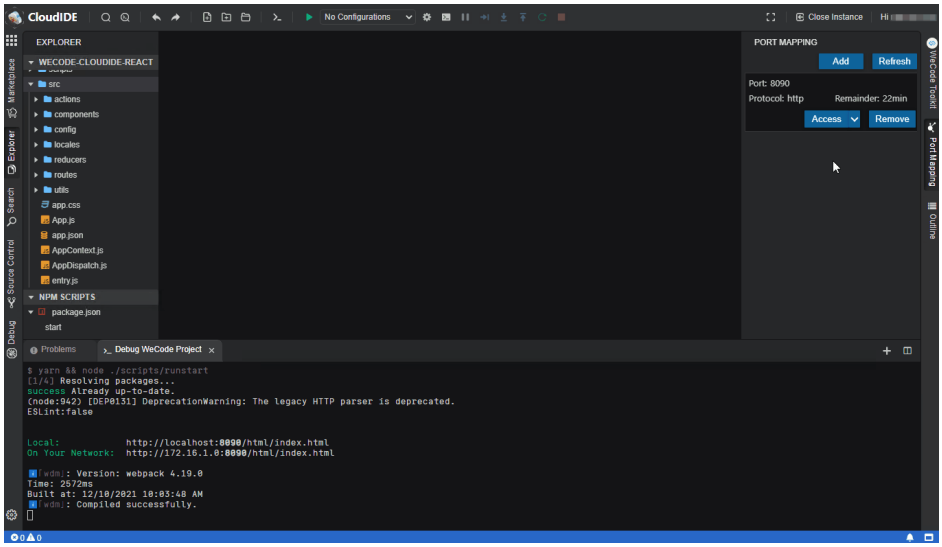
b. 切换至Port Mapping插件（端口映射），单击Access按钮访问。



c. 弹出app应用窗口，聚焦到app应用窗口上并按F12弹出网页调试窗口。

版权所有 © 华为云计算技术有限公司

图 3-2 在浏览器中调试 WeCode 应用

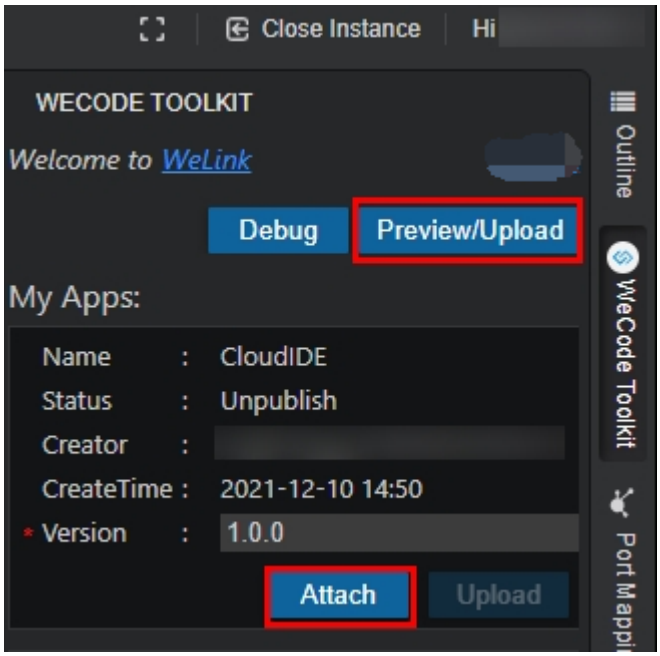


4. 小程序预览。

   程序开发完毕，通过使用WeCode Toolkit插件，CodeArts IDE Online中开发的We码小程序可被上传到临时仓库，并可以预览。

   a. 单击“Preview/Upload”，在“My Apps”列表中，选择刚刚新建的WeLink应用，并单击“Attach”。

   图 3-3 WeCode Toolkit 插件中上传预览



   b. 等待We码小程序上传至临时仓库后，可通过手机端WeLink扫码预览。

版权所有 © 华为云计算技术有限公司

图 3-4 "WeCode Toolkit"插件中扫描二维码以预览

c. 打开手机Welink扫码后，手机上就可以看到该应用的预览版。

版权所有 © 华为云计算技术有限公司

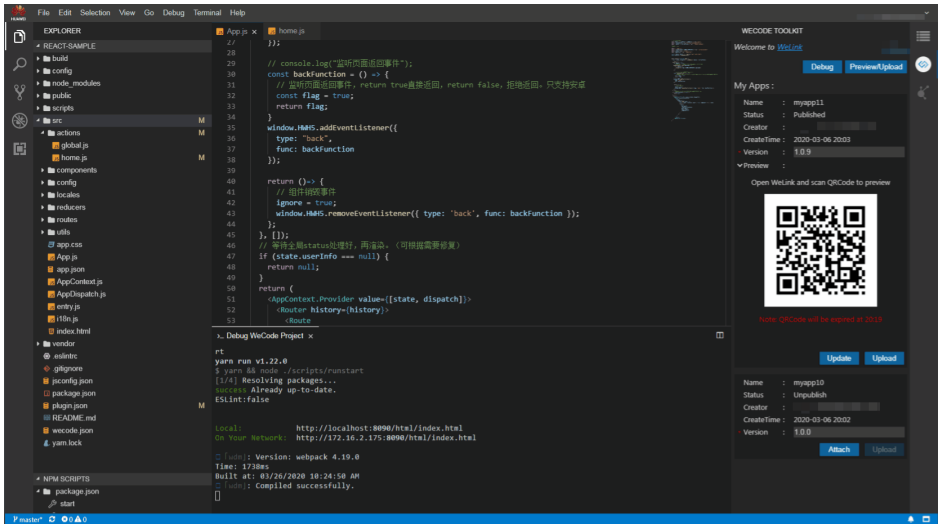
图 3-5 We 码小程序预览

步骤5 发布We码小程序。

通过CodeArts IDE Online上的WeCode Toolkit插件，可以将编写调试完毕的We码小程序上传发布到WeLink应用市场（这样手机上的WeLink应用可以搜索并安装这个已经发布的We码小程序）。

版权所有 © 华为云计算技术有限公司

单击“Upload”，将We码小程序发布到云端的发布仓库，然后进入“版本与发布”页面，单击“提交审核”，准备正式发布。



----结束

版权所有 © 华为云计算技术有限公司

4 基于 CodeArts IDE Online 开发并使用模型

本章节主要讲解如何在CodeArts IDE Online中使用TensorFlow和Jupyter Notebook完成神经网络模型的训练，并利用该模型完成简单的图像分类。

约束与限制

- 当前只对python3做了优化，python2下无法直接运行Jupyter Notebook。
- CodeArts IDE Online暂不支持GPU加速，建议安装tensorflow-cpu减小磁盘占用，并加快安装速度。
- 鲲鹏镜像暂时无法安装TensorFlow，敬请期待后续更新。

操作流程

1. **准备工作**
2. **导入和预处理训练数据集**
3. **开发深度学习模型**

4.1 准备工作

创建 IDE 实例

步骤1 登录到CodeArts IDE Online控制台，单击“创建实例”。

如果提示未开通则根据提示跳转至开通页面完成服务开通。

步骤2 进入“基础配置”页面，参数填写如**表4-1**所示，单击“下一步”。

表 4-1 基础配置参数说明

参数名	参数示例
名称	demo
描述	自定义对实例的描述，此处可以不填。
技术栈	Python

版权所有 © 华为云计算技术有限公司

参数名	参数示例
CPU架构	X86计算
CPU/内存	2U4G
存储容量	5GB
自动休眠	30分钟

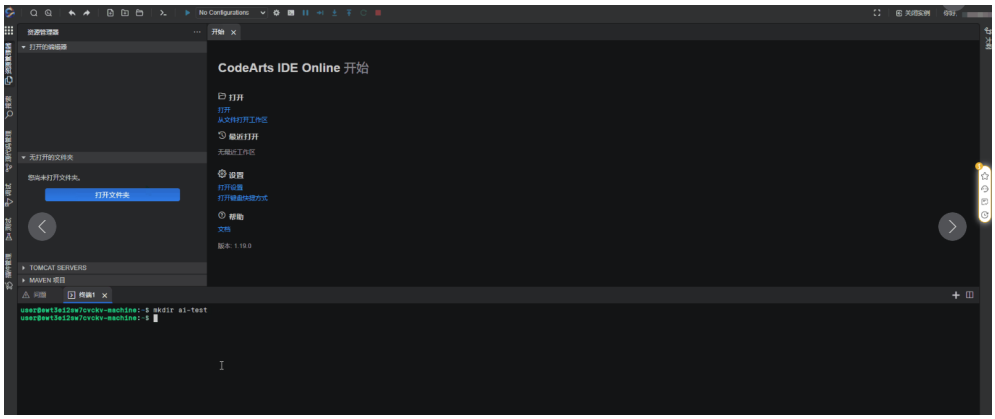
步骤3 进入“工程配置”页面，选择不创建工程，然后单击“确定”，完成实例创建。

----结束

安装 TensorFlow

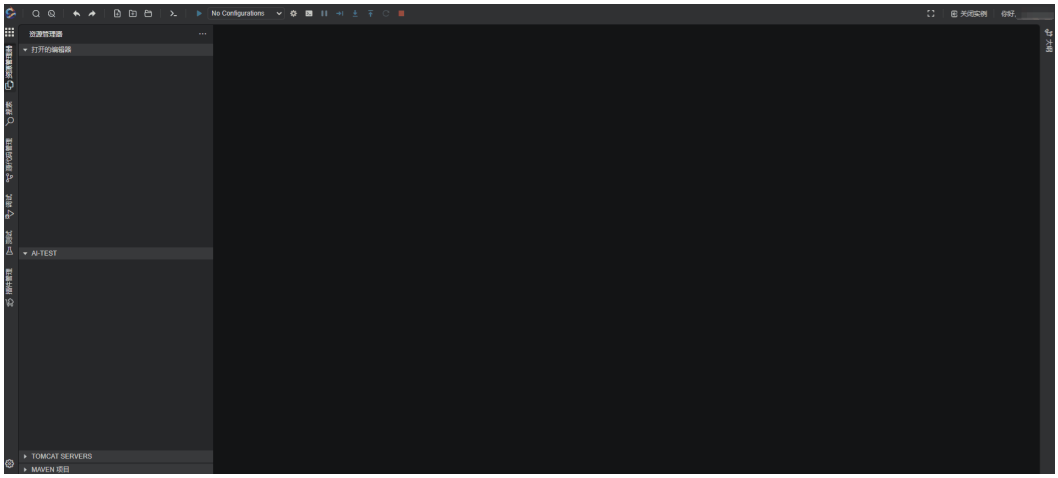
步骤1 进入CodeArts IDE Online实例，创建并打开一个空白工作目录，命令如下。

```
mkdir ai-test
```



步骤2 使用pip安装TensorFlow等依赖包，为加快安装速度此处安装的是tensorflow-cpu，命令如下。

```
python3 -m pip install tensorflow-cpu matplotlib numpy
pip install ipython==8.10.0
```

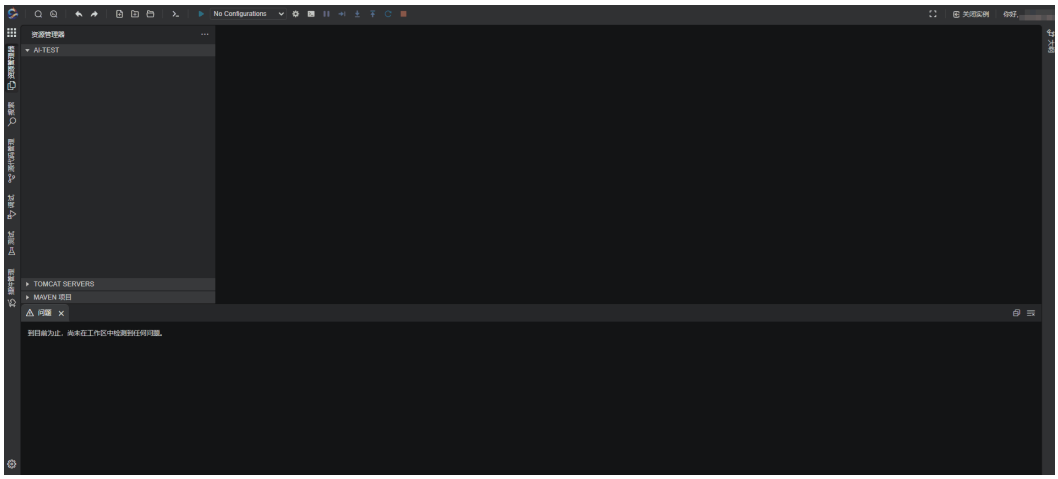


----结束

版权所有 © 华为云计算技术有限公司

进入 Notebook Editor

打开Jupyter Notebook（文件后缀名为“ipynb”），打开该文件后会出现一个Notebook Editor，可以在里面编辑和运行cell。

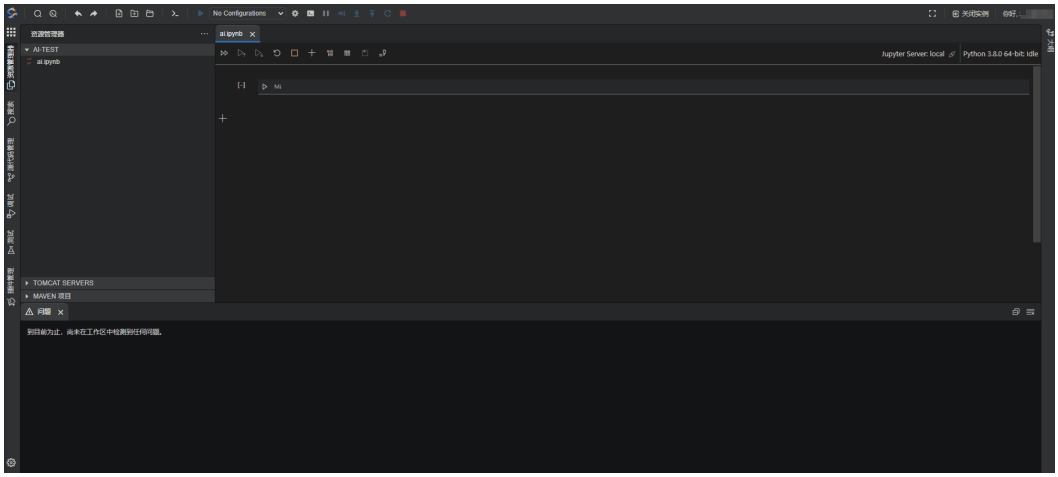


4.2 导入和预处理训练数据集

参考**TensorFlow官网的教程**，创建一个简单的图片分类模型。

查看当前TensorFlow版本，单击 或者敲击Shift+Enter运行cell。

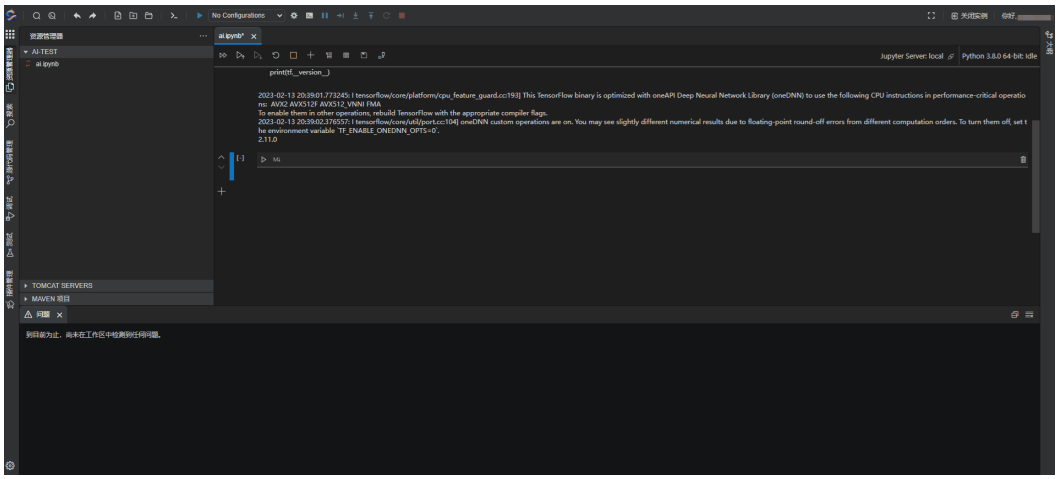
```
from __future__ import absolute_import, division, print_function, unicode_literals

# TensorFlow and tf.keras
import tensorflow as tf
from tensorflow import keras
# Helper libraries
import numpy as np
import matplotlib.pyplot as plt
# print tensorflow version
print(tf.__version__)
```



下载Fashion MNIST图片数据集，该数据集包含了10个类型共60000张训练图片以及10000张测试图片。

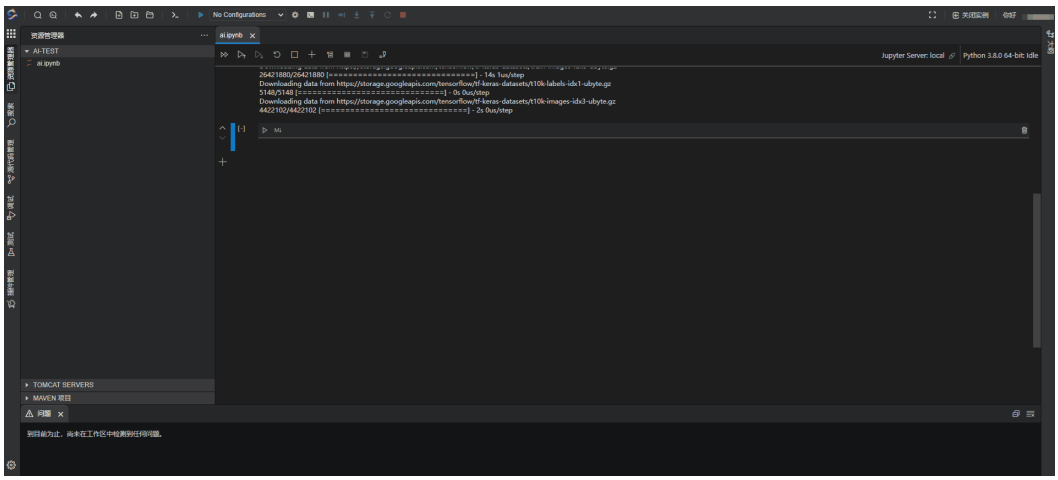
版权所有 © 华为云计算技术有限公司

```
# download Fashion MNIST dataset
fashion_mnist = keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```



对训练数据做预处理，并查看训练集中最开始的25个图片。

```
class_names = ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat',
               'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']
# preprocessing
train_images = train_images / 255.0
test_images = test_images / 255.0
# display first 25 images
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(train_images[i], cmap=plt.cm.binary)
    plt.xlabel(class_names[train_labels[i]])
plt.show()
```



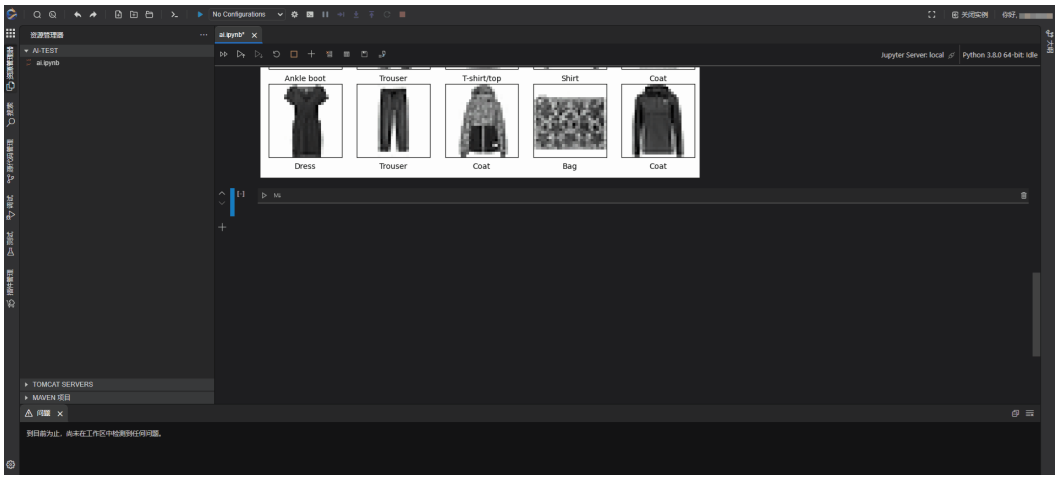
4.3 开发深度学习模型

创建和训练模型

使用如下命令创建并训练模型：

版权所有 © 华为云计算技术有限公司

```
# create model
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(10)
])
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
# training
model.fit(train_images, train_labels, epochs=10)
```



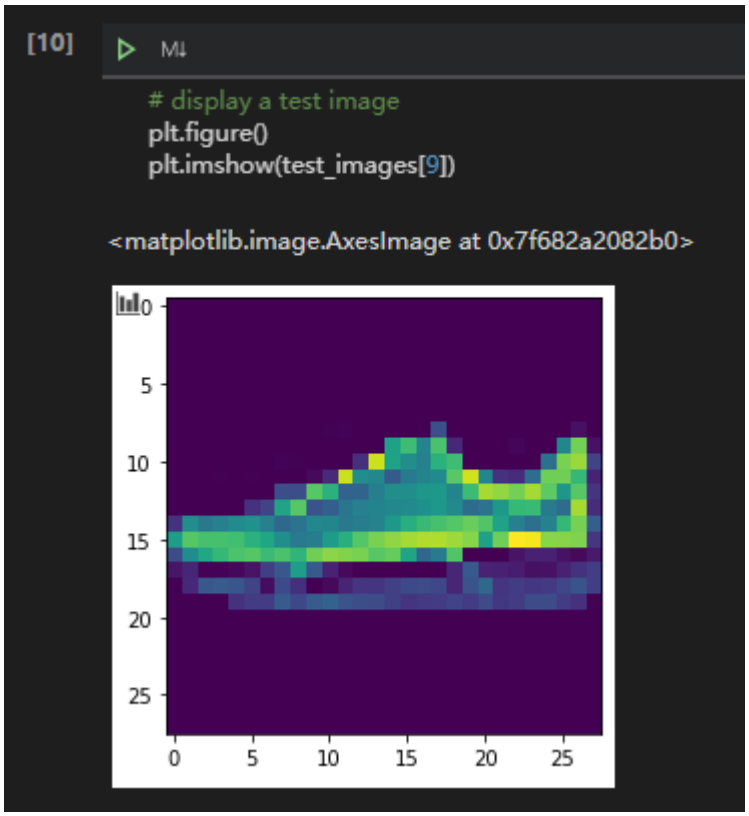
使用模型

用训练好的模型预测测试集中的某个图片属于什么类别，先显示这个图片，命令如下。

```
# display a test image
plt.figure()
plt.imshow(test_images[9])
```


版权所有 © 华为云计算技术有限公司

图 4-1 显示用以测试的图片

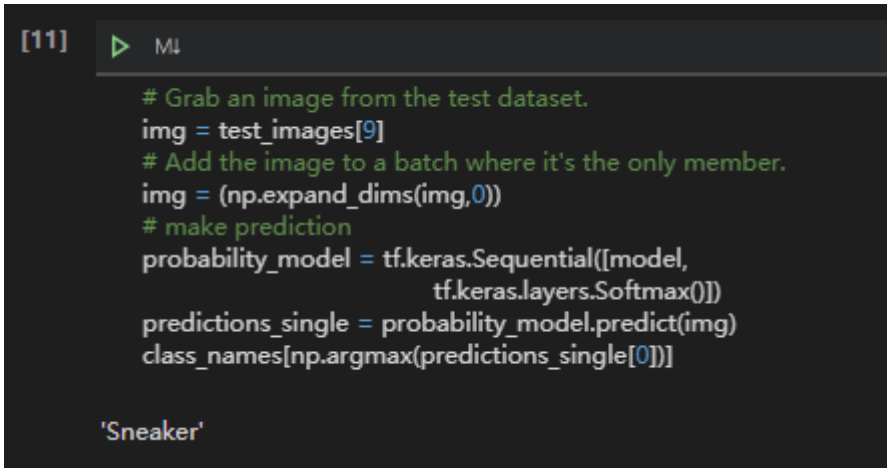


查看结果

查看预测结果，命令如下。

```
# Grab an image from the test dataset.
img = test_images[9]
# Add the image to a batch where it's the only member.
img = (np.expand_dims(img,0))
# make prediction
probability_model = tf.keras.Sequential([model,
                                         tf.keras.layers.Softmax()])
predictions_single = probability_model.predict(img)
class_names[np.argmax(predictions_single[0])]
```

图 4-2 查看预测结果



版权所有 © 华为云计算技术有限公司