

**ModelArts**

# **Tool Guide**

**Issue**            01  
**Date**             2023-06-30



**Copyright © Huawei Technologies Co., Ltd. 2023. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

## **Trademarks and Permissions**



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

## **Notice**

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

## **Huawei Technologies Co., Ltd.**

Address: Huawei Industrial Base  
Bantian, Longgang  
Shenzhen 518129  
People's Republic of China

Website: <https://www.huawei.com>

Email: [support@huawei.com](mailto:support@huawei.com)

---

# Contents

---

<b>1 PyCharm Toolkit.....</b>	<b>1</b>
<b>2 Preparations.....</b>	<b>3</b>
2.1 Downloading and Installing PyCharm Toolkit.....	3
2.2 Creating Access Keys (AK and SK).....	4
2.3 Using Access Keys for Login.....	5
<b>3 Connecting to a Notebook Instance Through PyCharm Toolkit.....</b>	<b>8</b>
<b>4 PyCharm Toolkit (Latest Version).....</b>	<b>17</b>
4.1 Training a Model.....	17
4.1.1 Submitting a Training Job (New Version).....	17
4.1.2 Stopping a Training Job.....	21
4.1.3 Viewing Training Logs.....	21
<b>5 OBS-based Upload and Download.....</b>	<b>23</b>
<b>6 FAQs.....</b>	<b>27</b>
6.1 What Should I Do If an Error Occurs During ToolKit Installation?.....	27
6.2 An Error Occurs When You Edit a Credential in PyCharm Toolkit.....	28
6.3 Why Cannot I Start Training?.....	30
6.4 What Should I Do If Error "xxx isn't existed in train_version" Occurs When a Training Job Is Submitted .....	30
6.5 What Should I Do If an Error Occurs When I Submit a Training Job.....	31
6.6 What Should I Do If an Error Occurs During Service Deployment.....	31
6.7 How Do I View Error Logs of PyCharm ToolKit?.....	32

# 1 PyCharm Toolkit

## NOTE

This document is applicable to Cloud Alliance sites in Europe.

AI developers use PyCharm tools to develop algorithms or models. Therefore, ModelArts provides PyCharm Toolkit to help AI developers quickly submit locally developed code to a training environment on ModelArts. With PyCharm Toolkit, developers can quickly upload code, submit training jobs, and obtain training logs for local display so that they can better focus on local code development. For details about how to download and install PyCharm Toolkit, see [Downloading and Installing PyCharm Toolkit](#).

## Constraints

- Currently, only PyCharm 2019.2 or later is supported, including the community and professional editions.
- Only PyCharm of the professional edition can be used to access the notebook development environment.
- You can use a community or professional edition of PyCharm Toolkit to submit training jobs. PyCharm Toolkit 2.x can be used only to submit training jobs of the old version, and the latest version of PyCharm Toolkit can be used only to submit training jobs of the new version.
- PyCharm Toolkit supports PyCharm of the Windows, Linux, or Mac version.

## Available Functions

**Table 1-1** Toolkit functions of the latest version

Function	Description	Reference
Remote SSH	The notebook development environment can be accessed through remote SSH.	<a href="#">Configuring PyCharm Toolkit to Remotely Connect to a Notebook Instance</a>

Function	Description	Reference
Model training	Code developed locally can be quickly submitted to ModelArts and a training job of the new version is automatically created. During the running of the training job, training logs can be obtained and displayed on a local host.	<ul style="list-style-type: none"><li>• <a href="#">Submitting a Training Job (New Version)</a></li><li>• <a href="#">Stopping a Training Job</a></li><li>• <a href="#">Viewing Training Logs</a></li></ul>
OBS-based upload and download	Local files or folders can be uploaded to OBS and files or folders can be downloaded from OBS to a local directory.	<a href="#">OBS-based Upload and Download</a>

# 2 Preparations

---

[Downloading and Installing PyCharm Toolkit](#)

[Creating Access Keys \(AK and SK\)](#)

[Using Access Keys for Login](#)

## 2.1 Downloading and Installing PyCharm Toolkit

Before using PyCharm Toolkit, install and configure it in PyCharm by following the instructions provided in this section.

### Prerequisites

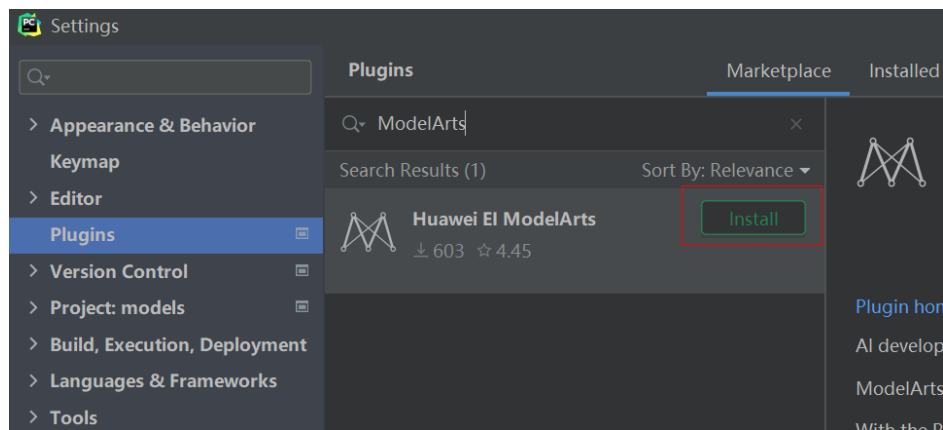
PyCharm community or professional 2019.2 or later has been installed locally.

- Only PyCharm of the professional edition can be used to access the notebook development environment.
- You can use a community or professional edition of PyCharm Toolkit to submit training jobs. PyCharm Toolkit 2.x can be used to submit only the old version of training jobs, and the latest version of PyCharm Toolkit can be used to submit only the new version of training jobs.

### Installing Through Marketplace

In PyCharm, choose **File > Settings > Plugins**, search for **ModelArts** in Marketplace, and click **Install**.

**Figure 2-1** Installing through Marketplace



**NOTE**

The version installed in Marketplace is the latest version.

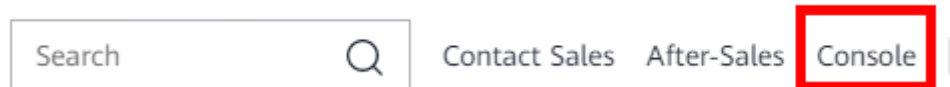
## 2.2 Creating Access Keys (AK and SK)

This section describes how to create access keys (AKs and SKs) on the ModelArts management console. A pair of AK and SK is used to encrypt the signature of a request, ensuring that the request is secure and integral, and that identities of the request sender and receiver are correct.

### Adding the AK/SK

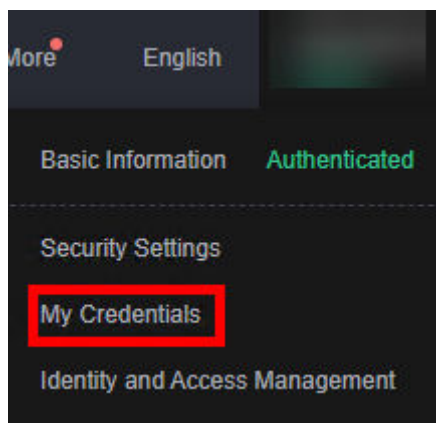
1. Log in to HUAWEI CLOUD and click **Console** in the upper right corner of the page to access the HUAWEI CLOUD management console.

**Figure 2-2** Console



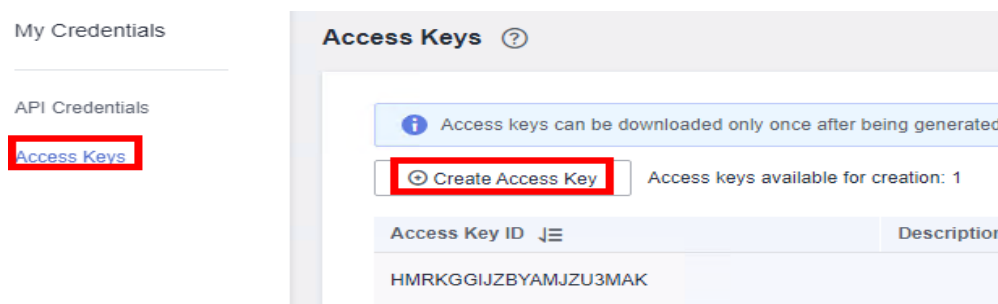
2. Hover the cursor over the account name in the upper right corner of the console and choose **My Credentials** from the drop-down list. The **API Credentials** page is displayed.

**Figure 2-3 My Credentials**



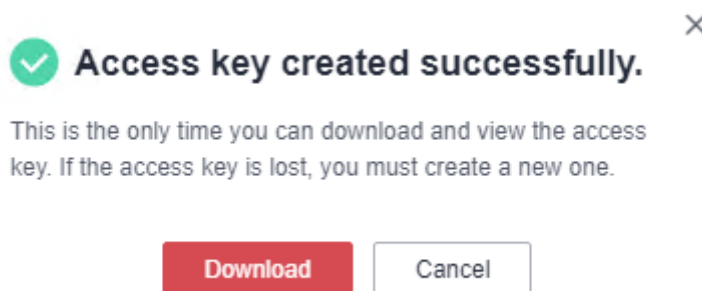
3. On the **API Credentials** page, choose **Access Keys > Create Access Key**. See [Figure 2-4](#).

**Figure 2-4 Clicking Create Access Key**



4. Enter the description of the key and click **OK**. Click **Download** to download the key.

**Figure 2-5 Create Access Key dialog box**



5. The access key file is saved in the default download folder of the browser. Open the **credentials.csv** file to view the AK and SK.

## 2.3 Using Access Keys for Login

To connect Toolkit to ModelArts, use the access keys of the current account for login authentication.



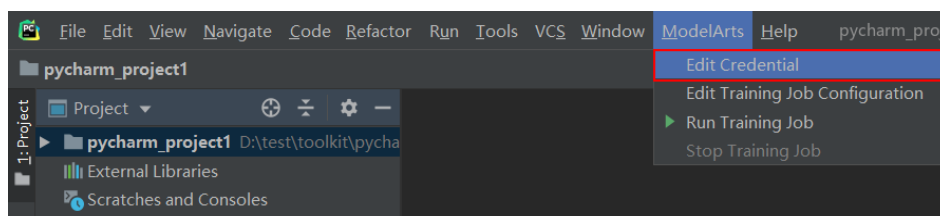
## Prerequisites

- Toolkit has been installed. If it is not installed, install it by referring to [Downloading and Installing PyCharm Toolkit](#).
- The access keys of the current account have been created, and the corresponding AK and SK have been obtained. If they are not created, create them by referring to [Creating Access Keys \(AK and SK\)](#).
- Before using Toolkit, go to the ModelArts console to configure access authorization. If the global configuration is not complete on the ModelArts management console, you cannot access and connect to ModelArts after logging in to Toolkit.

## Logging In to ModelArts

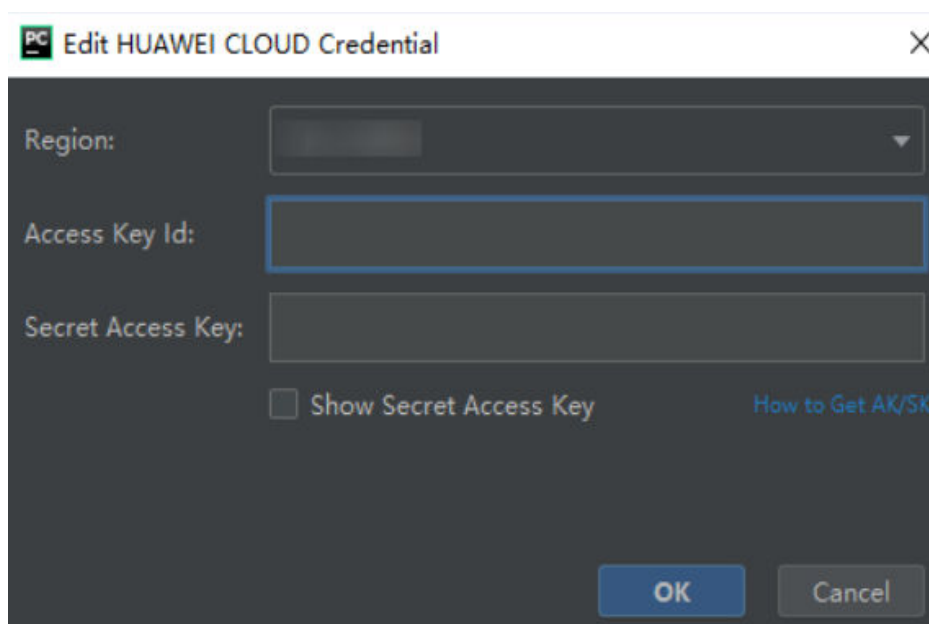
1. Open PyCharm with Toolkit installed. Choose **ModelArts > Edit Credential** from the menu bar.

Figure 2-6 Edit Credential



2. In the displayed dialog box, select the region where ModelArts is located, enter the AK and SK, and click **OK**.
  - **Region:** Select a region from the drop-down list.
  - **Access Key ID:** Enter the AK.
  - **Secret Access Key:** Enter the SK.

Figure 2-7 Entering the region and access keys



3. View the verification result.

In the **Event Log** area, if information similar to the following is displayed, the access key has been successfully added:

16:01 Validate Credential Success: The HUAWEI CLOUD credential is valid.

# 3 Connecting to a Notebook Instance Through PyCharm Toolkit

---

ModelArts provides the PyCharm plug-in PyCharm Toolkit for you to remotely access a notebook instance through SSH, upload code, submit a training job, and obtain training logs for local display.

## Prerequisites

PyCharm professional 2019.2 or later has been installed locally. Remote SSH applies only to the PyCharm professional edition. [Download PyCharm](#) and install it.

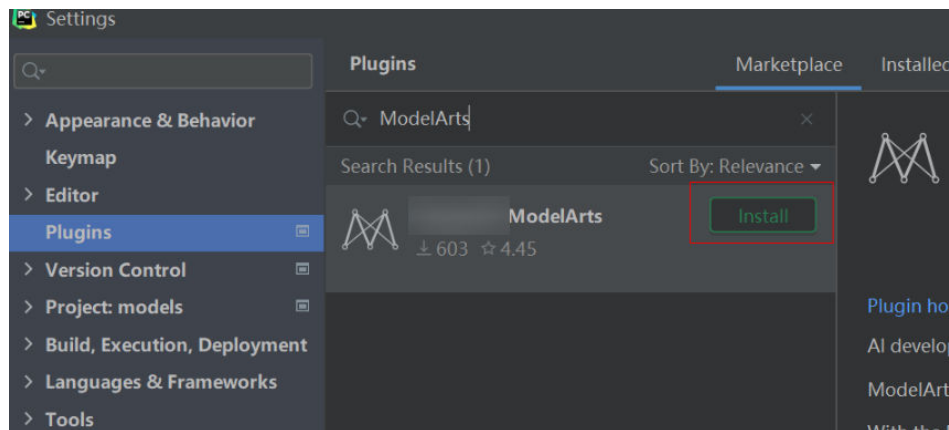
## Step 1 Create a Notebook Instance

Create a notebook instance with remote SSH enabled and whitelist configured. Ensure that the instance is running. For details, see [https://support.huaweicloud.com/eu/devtool-modelarts/devtool-modelarts\\_0004.html](https://support.huaweicloud.com/eu/devtool-modelarts/devtool-modelarts_0004.html).

## Step 2 Download and Install PyCharm Toolkit

### Installing through Marketplace

In PyCharm, choose **File > Settings > Plugins**, search for **ModelArts** in Marketplace, and click **Install**.

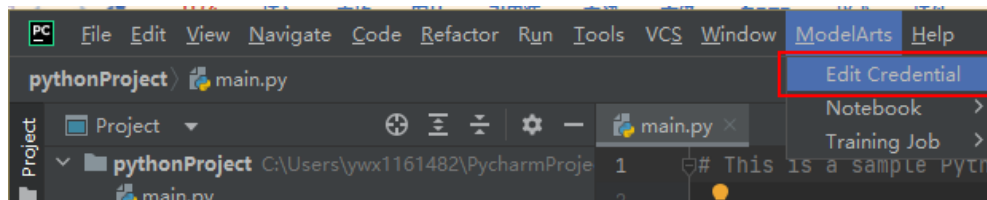
**Figure 3-1** Installing through Marketplace**NOTE**

The version installed in Marketplace is the latest version.

**Step 3 Log In to the Plug-in**

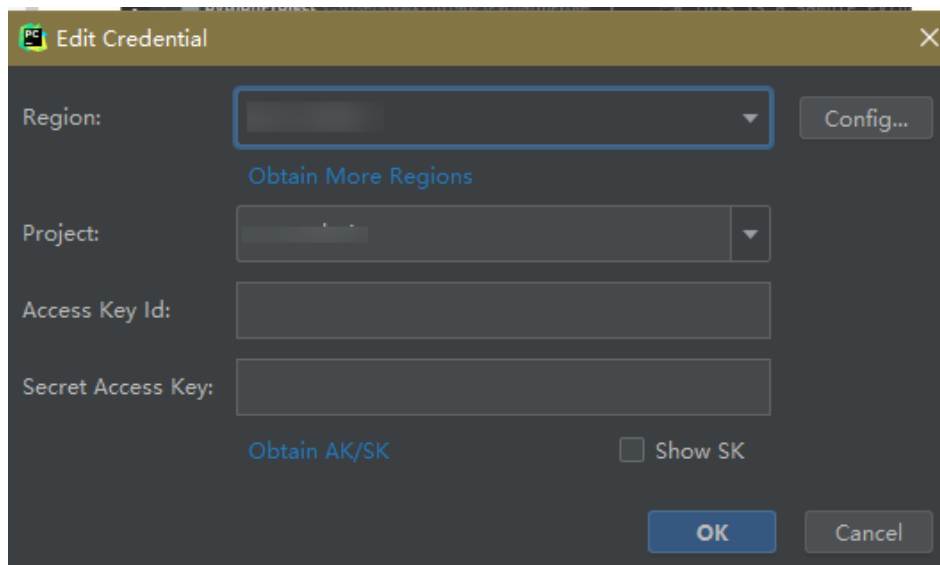
To use the AK/SK pair for login authentication, perform the following steps:

1. Open PyCharm with Toolkit installed. Choose **ModelArts > Edit Credential** from the menu bar.

**Figure 3-2** Edit Credential

2. In the displayed dialog box, select the region where ModelArts is located, enter the AK and SK, and click **OK**. For details about how to obtain the AK and SK, see [How Do I Obtain an Access Key?](#)
  - **Region:** Select a region from the drop-down list. It must be the same as the region of the ModelArts console.
  - **Project:** After the region is selected, the project is automatically filled.
  - **Access Key ID:** Enter the AK.
  - **Secret Access Key:** Enter the SK.

**Figure 3-3** Entering the region and access keys



3. View the verification result.

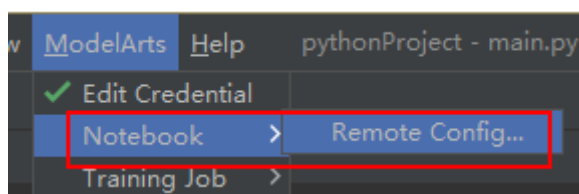
In the **Event Log** area, if information similar to the following is displayed, the access key has been successfully added:

16:01 Validate Credential Success: The HUAWEI CLOUD credential is valid.

## Step 4 Automatically Configure PyCharm Toolkit

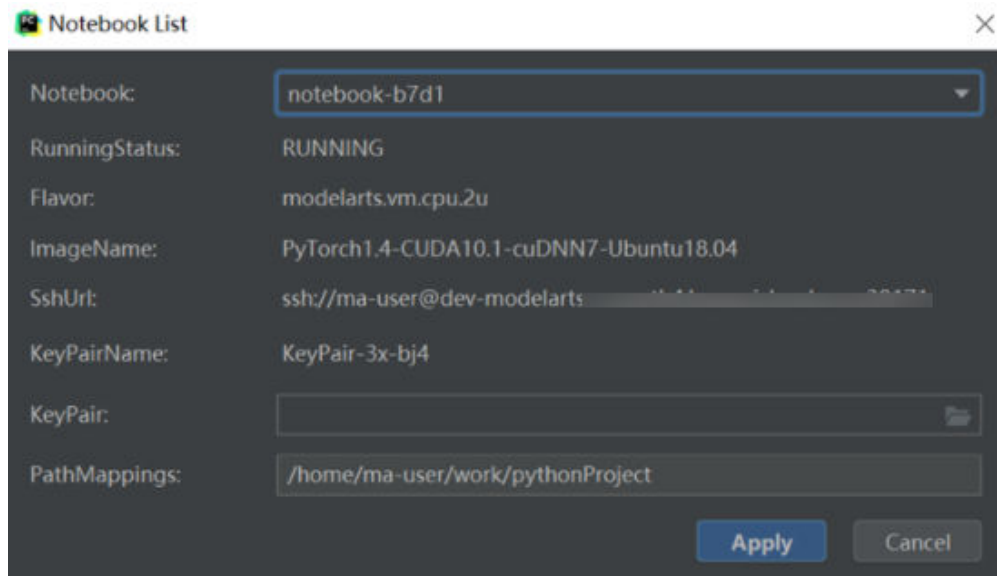
1. In the local PyCharm development environment, choose **ModelArts > Notebook > Remote Config...** and configure PyCharm Toolkit.

**Figure 3-4** Remotely connecting to PyCharm Toolkit



2. All notebook instances with remote SSH enabled under the account are displayed. Choose the target instance from the drop-down list.

**Figure 3-5** Notebook list

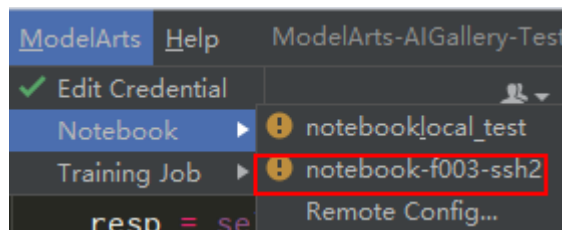


- **KeyPair:** Select the locally stored key pair of the notebook instance for authentication. The key pair created during the notebook instance creation is saved in your browser's default downloads folder.
  - **PathMappings:** Synchronization directory for the local IDE project and notebook, which defaults to `/home/ma-user/work/Project name` and is adjustable.
3. Click **Apply**. After the configuration is complete, restart the IDE for the configuration to take effect.  
After the restart, it takes about 20 minutes to update the Python interpreter for the first time.

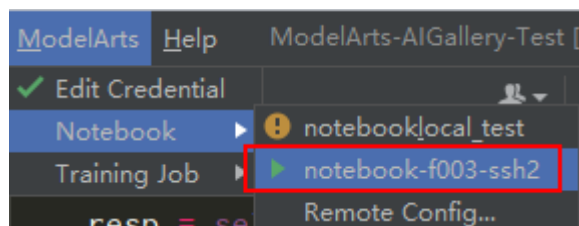
## Step 5 Access a Notebook Instance Through PyCharm Toolkit

Click the notebook instance name and connect it to the local IDE as prompted. The connection is kept for 4 hours by default.

**Figure 3-6** Starting the connection



To interrupt the connection, click the notebook name and disconnect it from the local IDE as prompted.

**Figure 3-7** Interrupting the connection

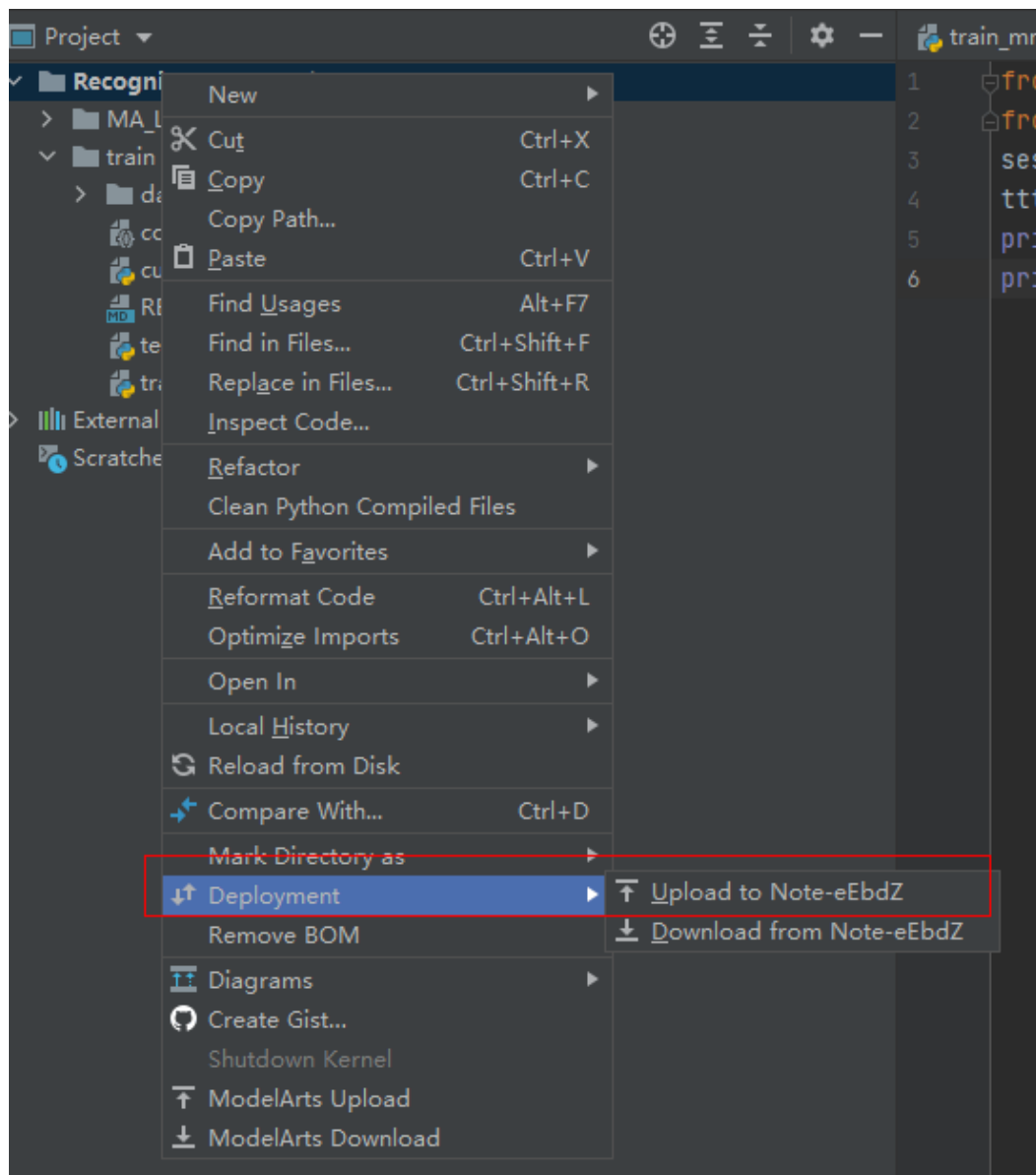
## Step 6 Upload Local Files to the Notebook Instance

Code in a local file can be copied to the local IDE, which will automatically synchronize the code to the in-cloud development environment.

### Initial synchronization

In the **Project** directory of the local IDE, right-click **Deployment** and choose **Upload to Notebook name** from the shortcut menu to upload the local project file to the specified notebook instance.

**Figure 3-8** Synchronizing local data to a notebook instance



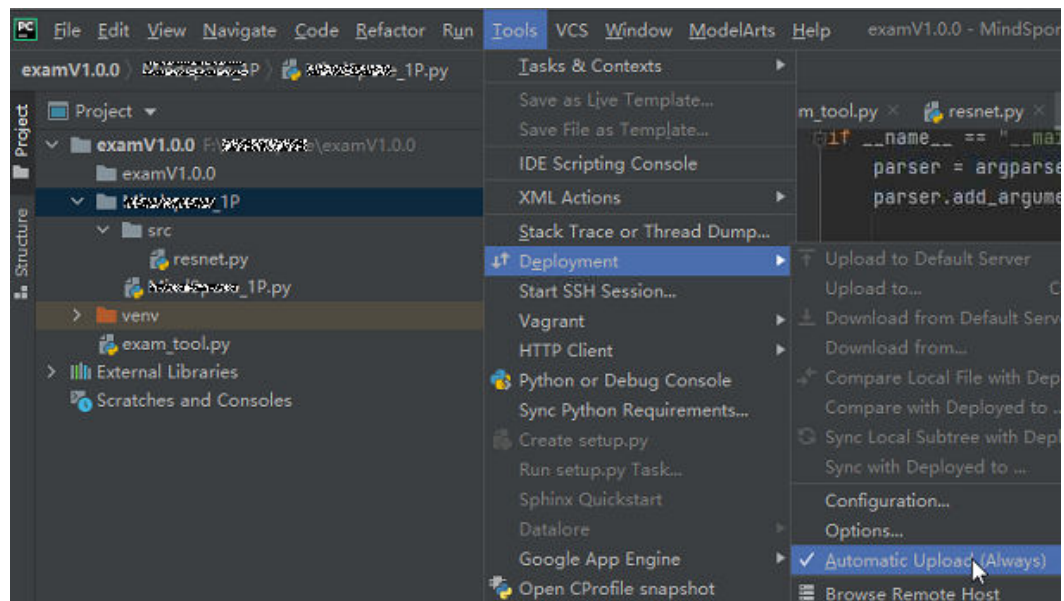
### Follow-up synchronization

After modifying the code, press **Ctrl+S** to save it. The local IDE will automatically synchronize the modification to the specified notebook instance.

After PyCharm Toolkit is installed, **Automatic Upload** is automatically enabled in the local IDE for automatically uploading the files in the local directory to the target notebook instance. If **Automatic Upload** is not enabled, enable it by referring to the following figure.



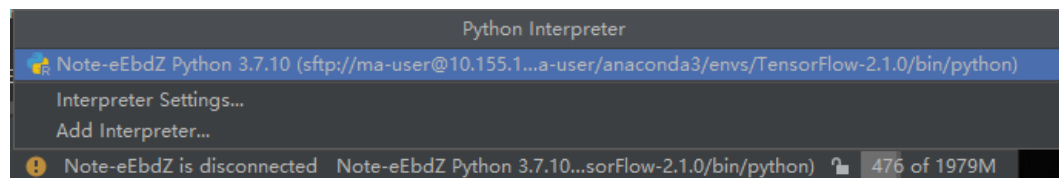
**Figure 3-9** Enabling Automatic Upload



## Step 7 Remotely Debug the Code

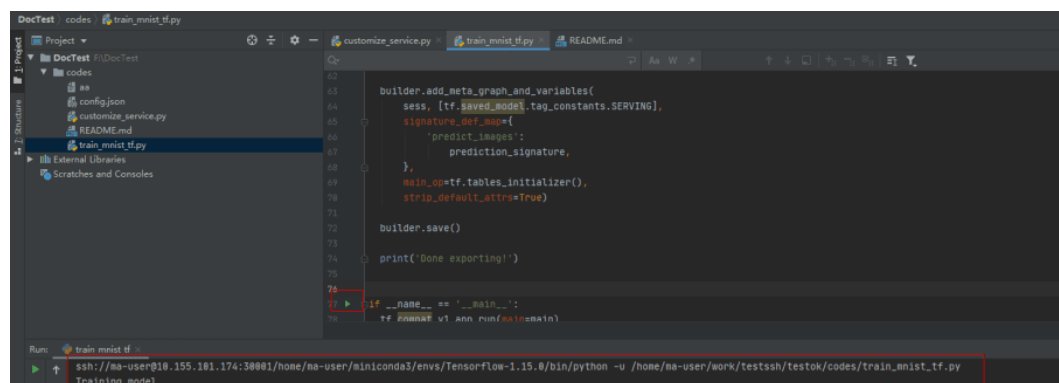
Click **Interpreter** in the lower right corner of the local IDE and select a notebook Python interpreter.

**Figure 3-10** Selecting a Python interpreter



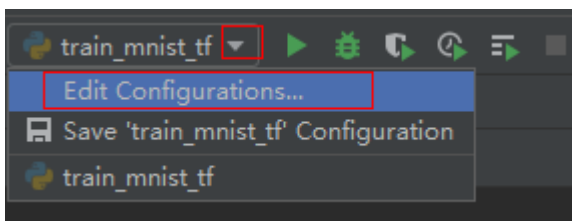
Run the code in the notebook instance. The logs are displayed locally.

**Figure 3-11** Runtime logs



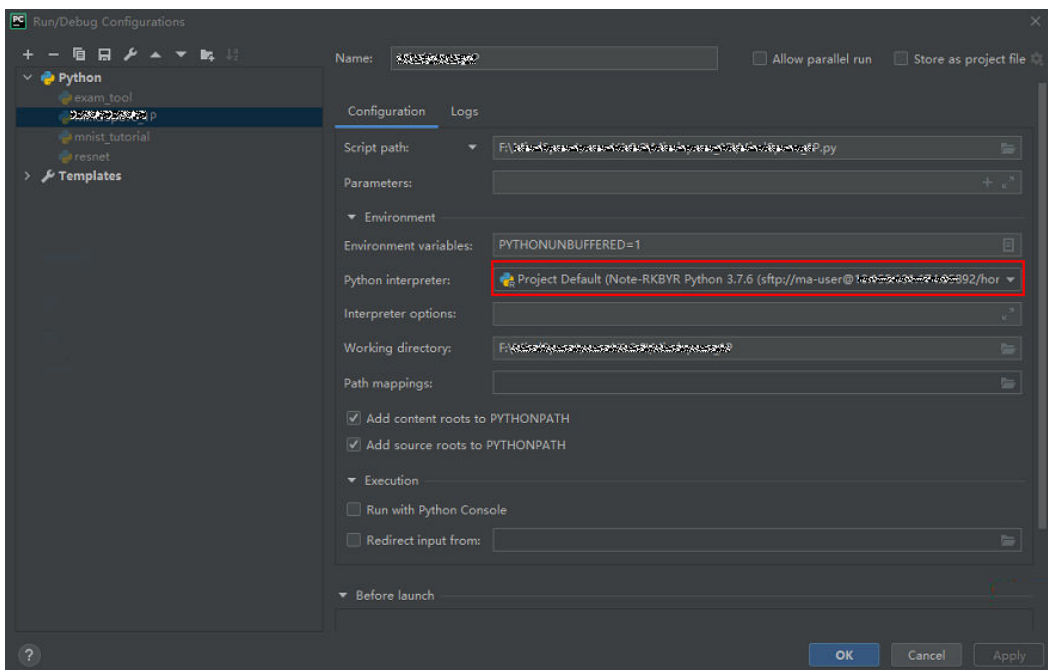
Click **Run/Debug Configurations** in the upper right corner of the local IDE to set runtime parameters.

**Figure 3-12** Setting runtime parameters (1)



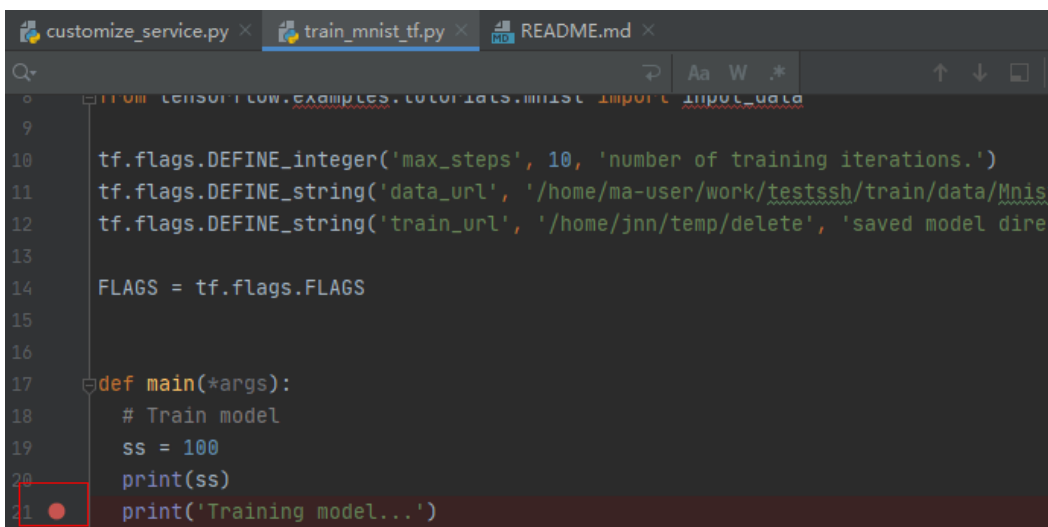
Select the Python interpreter that remotely connects to the target notebook instance.

**Figure 3-13** Setting runtime parameters (2)



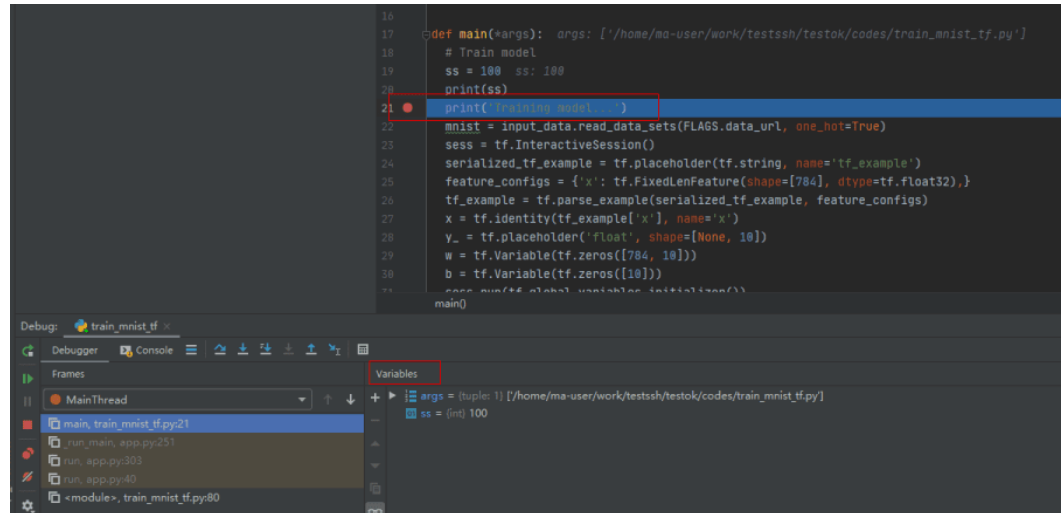
To debug code, set breakpoints and run the program in debug mode.

**Figure 3-14** Running the program in debug mode



In debug mode, the code execution is suspended in the specified line, and you can obtain variable values.

**Figure 3-15** Viewing variable values in debug mode



# 4 PyCharm Toolkit (Latest Version)

## Training a Model

### 4.1 Training a Model

#### 4.1.1 Submitting a Training Job (New Version)

You can use PyCharm Toolkit of the latest version to quickly submit the locally developed training code to ModelArts for training.

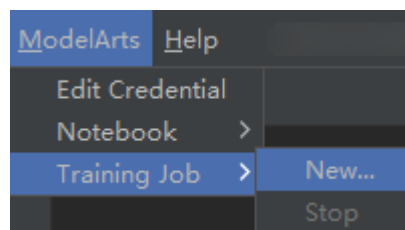
##### Prerequisites

- A training code project exists in the local PyCharm.
- You have created a bucket and folders in OBS for storing datasets and trained models. Data used by the training job has been uploaded to OBS.
- The credential has been configured. For details, see [Using Access Keys for Login](#).
- PyCharm Toolkit of the latest version is available for submitting a training job of the new version only.

##### Configuring Training Job Parameters

1. In PyCharm, open the training code project and training boot file, and choose **ModelArts > Training Job > New...** on the menu bar.

**Figure 4-1** Edit training job configuration



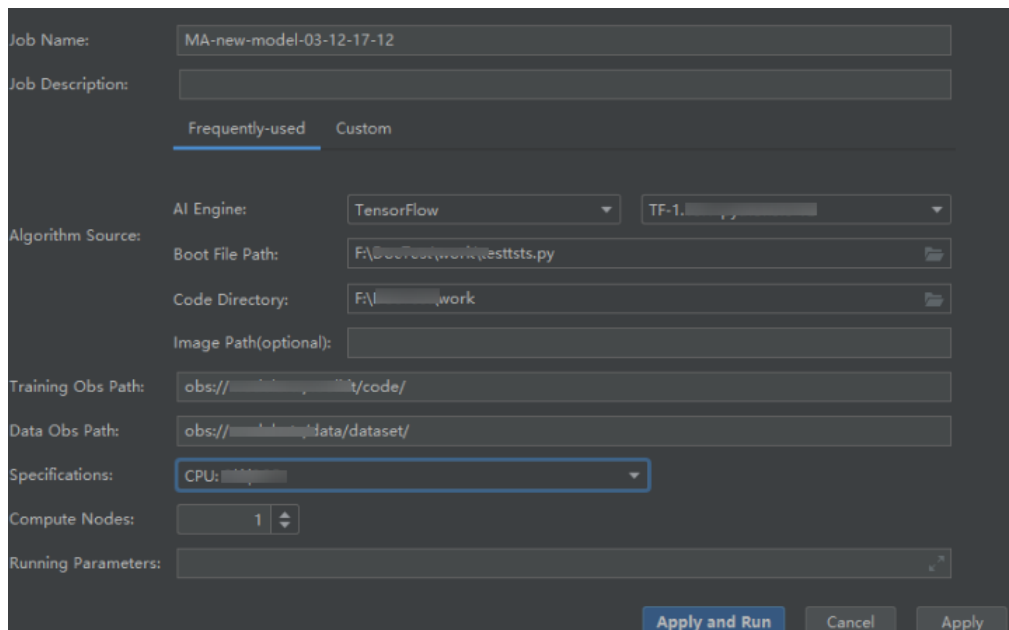
2. In the displayed dialog box, set the training job parameters. For details about the parameters, see [Table 4-1](#).

**Table 4-1** Training job parameters

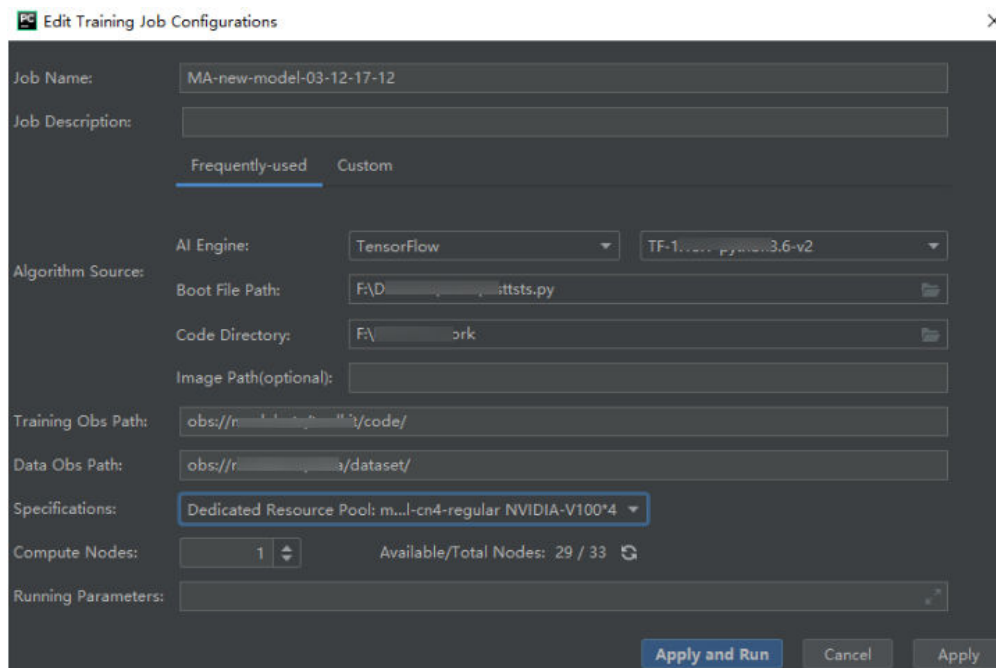
Parameter	Description
Job Name	Name of a training job
Job Description	Brief description of a training job
Algorithm Source	Source of the training algorithm. The options are <b>Frequently-used</b> and <b>Custom</b> . <b>Frequently-used</b> refers to the frequently-used AI engines supported by ModelArts Training Management. If the AI engine you use is not in the supported list, you are advised to create a training job using a custom image.
AI Engine	Select the AI engine and the version used in code. The supported AI engines are the same as the frequently-used frameworks supported by training jobs on the ModelArts management console.
Boot File Path	Training boot file. The selected boot file must be a file in the current PyCharm training project.
Code Directory	Training code directory. The system automatically sets this parameter to the directory where the training boot file is located. You can change the parameter value to a directory that is in the current project and contains the boot file.  If the algorithm source is a custom image and the training code has been built in the image, this parameter can be left blank.
Image Path (optional)	URL of the SWR image
Boot Command	Command for starting a training job, for example, <b>bash /home/work/run_train.sh python {Python boot file and parameters}</b> . This parameter is displayed if <b>Algorithm Source</b> is set to <b>Custom</b> .  If the command does not contain the <b>--data_url</b> or <b>--train_url</b> parameter, the tool automatically adds the two parameters to the end of the command when submitting the training job. The two parameters correspond to the OBS path for storing training data and the OBS path for storing training output, respectively.
Data Obs Path	OBS path for storing training data, for example, <b>/test-modelarts2/mnist/dataset-mnist/</b> , in which <b>test-modelarts2</b> indicates a bucket name.
Training Obs Path	OBS path. A directory is automatically created in the path for storing a trained model and training logs.

Parameter	Description
Running Parameters	Running parameters. If you want to add some running parameters to your code, add them here. Separate multiple running parameters with semicolons (;), for example, <b>key1=value1;key2=value2</b> . This parameter can be left blank.
Specifications	Type of resources used for training. Currently, public resource pools and dedicated resource pools are supported.  Dedicated resource pool specifications are identified by <b>Dedicated Resource Pool</b> . Dedicated resource pool specifications are displayed only for users who have purchased dedicated resource pools.
Compute Nodes	Number of compute nodes. If this parameter is set to <b>1</b> , the system runs in standalone mode. If this parameter is set to a value greater than 1, the distributed computing mode is used at the background.
Available/Total Nodes	When <b>Specifications</b> is set to a dedicated resource pool, the number of available nodes and the total number of nodes are displayed. The value of <b>Compute Nodes</b> cannot exceed the number of available nodes.

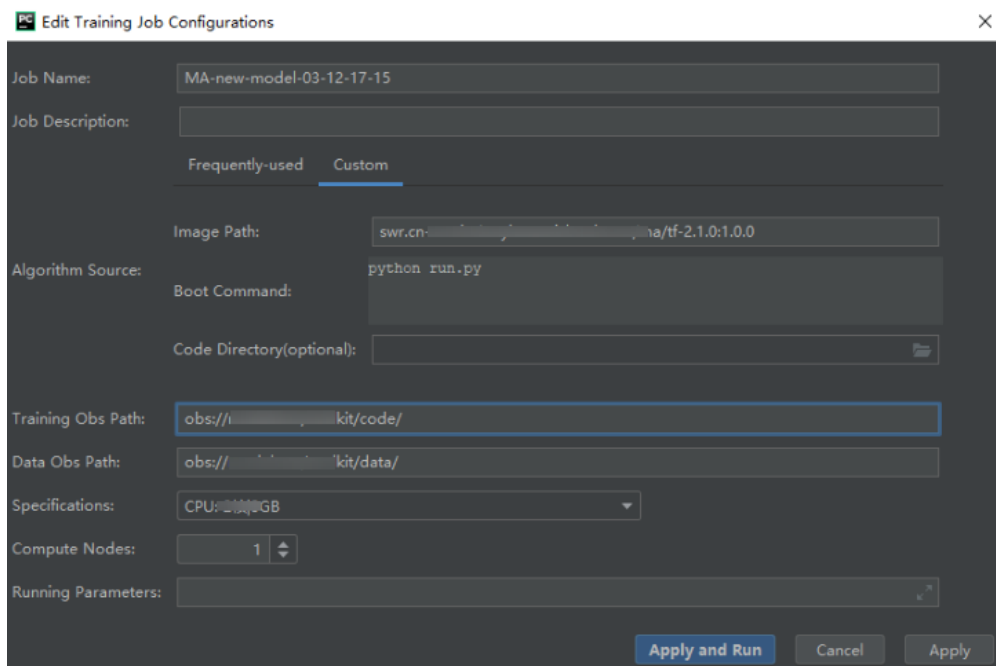
**Figure 4-2** Configuring training job parameter (public resource pool)



**Figure 4-3** Configuring training job parameter (dedicated resource pool)



**Figure 4-4** Configuring training job parameter (custom image)

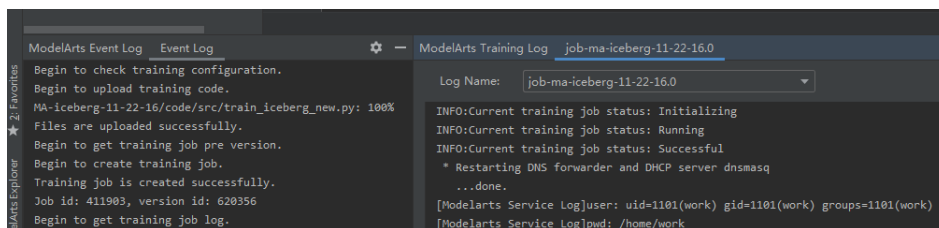


3. After setting the parameters, click **Apply and Run**. Then, local code is automatically uploaded to the cloud and training is started. The training job running status is displayed in the **Training Log** area in real time. If information similar to **Current training job status: Successful** is displayed in the training log, the training job has been successfully executed.

 **NOTE**

- After you click **Apply and Run**, the system automatically executes the training job. To stop the training job, choose **ModelArts > Training Job > Stop** on the menu bar.
- If you click **Apply**, the job is not started directly, and the training job settings are saved instead. To start the job, click **Apply and Run**.

**Figure 4-5** Training log example



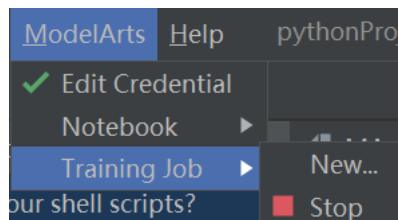
## 4.1.2 Stopping a Training Job

You can stop a running training job.

### Stopping a Job

When a training job is running, choose **ModelArts > Training Job > Stop** on the PyCharm menu bar to stop the job.

**Figure 4-6** Stopping a job



## 4.1.3 Viewing Training Logs

This section describes how to view training job logs.

### Viewing Training Logs in OBS

When you submit a training job, the system automatically creates a folder with the same name as the training job in the configured OBS path to store the model, logs, and code outputted after training is complete.

For example, when the **train-job-01** job is submitted, a folder named **train-job-01** is created in the **test-modelarts2** bucket. In this folder, three sub-folders (**output**, **log**, and **code**) are created to store the outputted model, logs, and training code, respectively. Sub-folders will be created in the **output** folder based on your training job version. The following is an example of the folder structure:

```
test-modelarts2
|---train-job-01
|---output
```

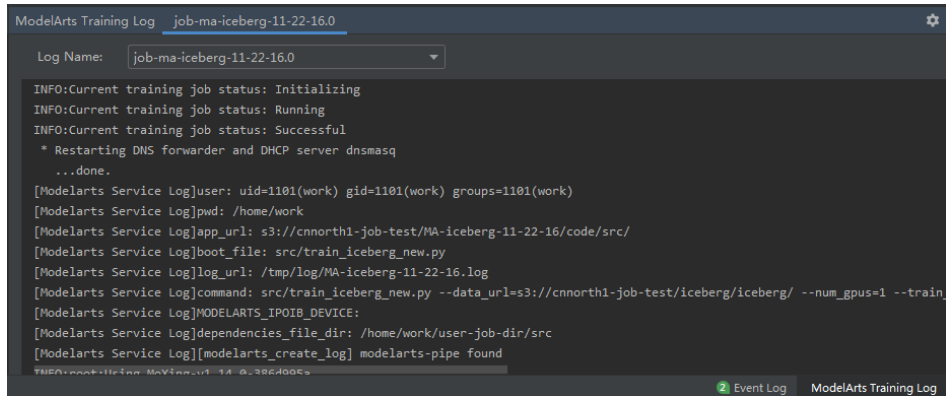


```
|---log  
|---code
```

## Viewing Training Logs in Toolkit

In PyCharm, click **ModelArts Training Log** in the lower right corner of the page. The training logs are displayed.

**Figure 4-7** Viewing Training Logs



# 5 OBS-based Upload and Download

---

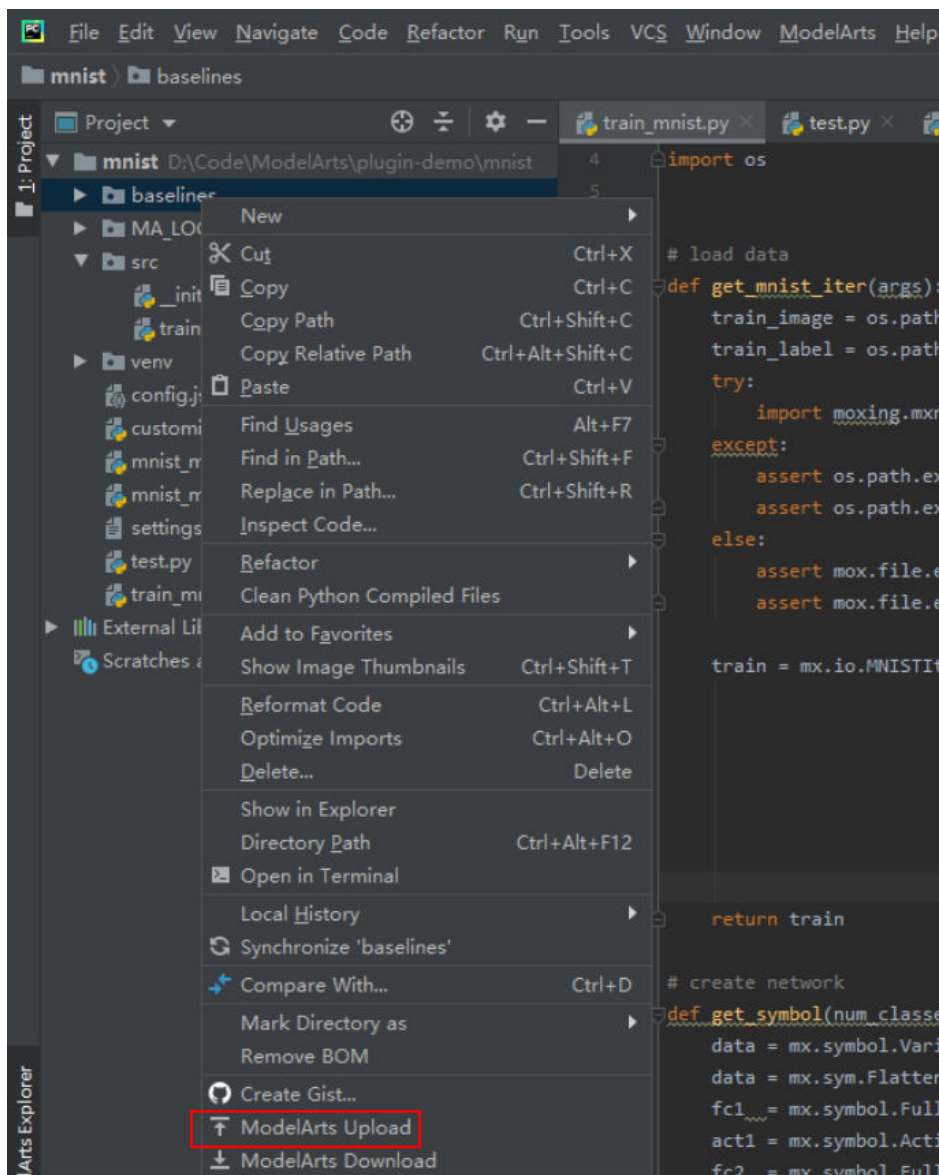
If data used for training needs to be stored to OBS, you can use the upload and download function provided by PyCharm Toolkit to upload local files or folders to OBS or download files or folders from OBS to the local host on the PyCharm page.

After files or folders are uploaded to OBS, you can directly define the OBS path in your training script.

## Uploading a Local File or Folder to OBS

1. Right-click in the **Project** window and choose **ModelArts Upload** from the shortcut menu.

Figure 5-1 ModelArts Upload

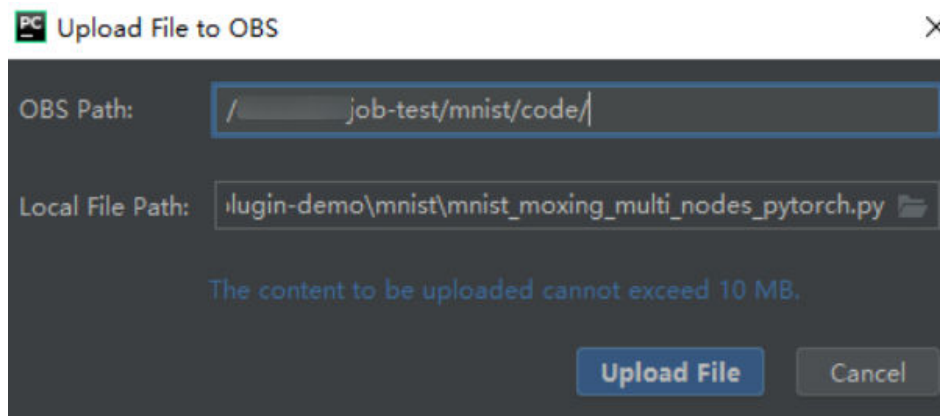


2. In the dialog box that is displayed, set **OBS Path** and **Local File Path**. It is recommended that the size of the file or folder to be uploaded be less than or equal to 10 MB. Click **Upload File**.

**OBS Path:** indicates the OBS path to which the local file or folder is to be uploaded.

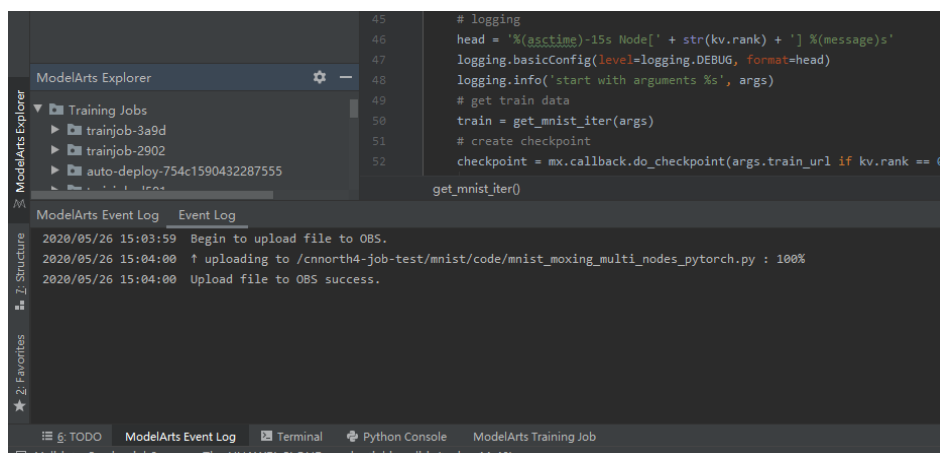
**Local File Path:** indicates the local directory where the file or folder to be uploaded is located. If a specific file is to be uploaded, set this parameter to the file name.

Figure 5-2 Entering file paths



3. In the **ModelArts Event Log** window, view the upload logs.

Figure 5-3 Viewing upload logs



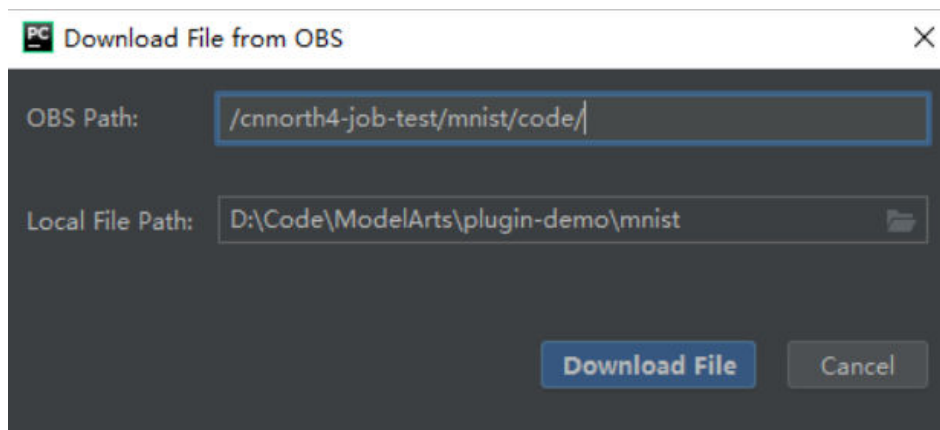
## Downloading a File or Folder from OBS

1. Right-click in the **Project** window and choose **ModelArts Download** from the shortcut menu.
2. In the dialog box that is displayed, set **OBS Path** and **Local File Path** and click **Download File**.

**OBS Path:** indicates the OBS path of the file or folder to be downloaded. If a specific file is to be downloaded, set this parameter to the file name.

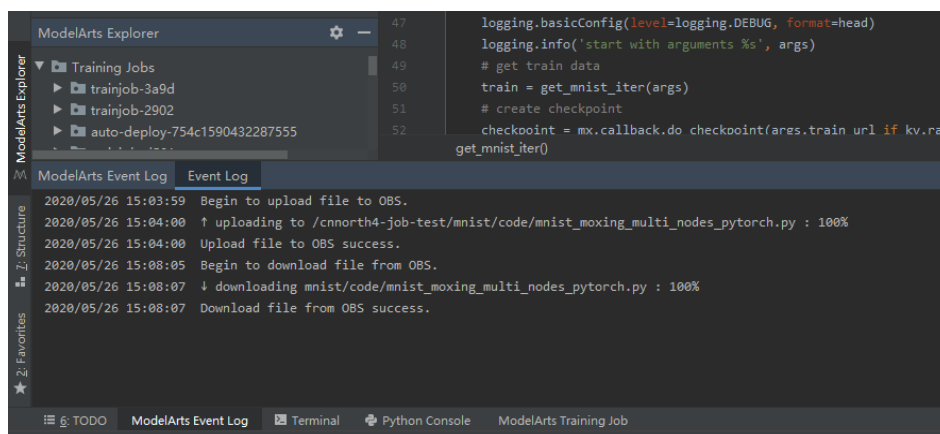
**Local File Path:** indicates the local path for storing downloaded files.

**Figure 5-4** Downloading a file from OBS



3. If a file with the same name already exists in the local directory, a message is displayed, asking you whether to overwrite the existing file. If you select **Yes**, the existing file will be overwritten. If you select **No**, the file conflicting with the existing file will not be downloaded and other files that do not conflict with the existing file will continue to be downloaded. Set this parameter based on the site requirements.
4. In the **ModelArts Event Log** window, view the download logs.

**Figure 5-5** Viewing download logs



# 6 FAQs

[What Should I Do If an Error Occurs During ToolKit Installation?](#)

[An Error Occurs When You Edit a Credential in PyCharm Toolkit](#)

[Why Cannot I Start Training?](#)

[What Should I Do If Error "xxx isn't existed in train\\_version" Occurs When a Training Job Is Submitted](#)

[What Should I Do If an Error Occurs When I Submit a Training Job](#)

[What Should I Do If an Error Occurs During Service Deployment](#)

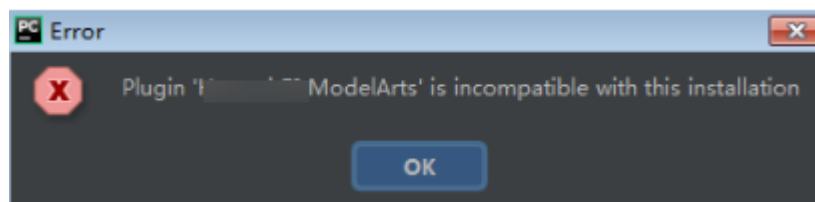
[How Do I View Error Logs of PyCharm Toolkit?](#)

## 6.1 What Should I Do If an Error Occurs During ToolKit Installation?

### Issue

The following error message is displayed during ToolKit installation.

**Figure 6-1** Error



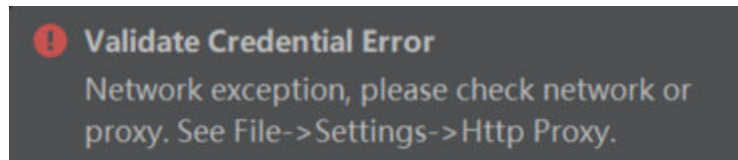
### Solution

This issue occurs because the plug-in version is inconsistent with the PyCharm version. You need to obtain the plug-in of the same version as the PyCharm version, that is, version 2019.2 or later.

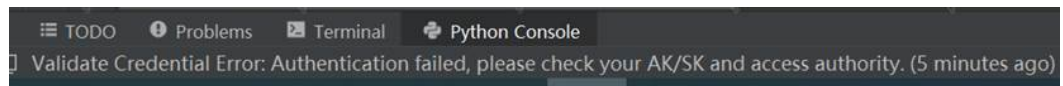
## 6.2 An Error Occurs When You Edit a Credential in PyCharm Toolkit

### Symptom

When you edit a credential in PyCharm Toolkit, the message "Validate Credential error" is displayed.



Or



### Possible Causes

- Possible cause 1: Information such as the region is incorrectly configured.
- Possible cause 2: The **hosts** file is not configured or is incorrectly configured.
- Possible cause 3: The network proxy settings are incorrect.
- Possible cause 4: The AK/SK is incorrect.
- Possible cause 5: The computer time is incorrectly set.

### Solution

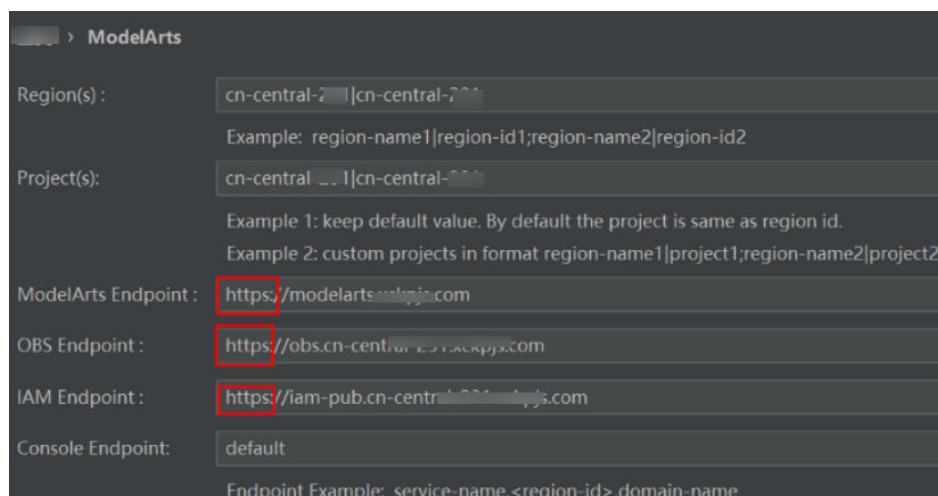
#### 1. Information such as the region is incorrectly configured.

Configure the correct information.

For example, if the endpoint is incorrect, the authentication fails.

Incorrect example: The endpoint is preceded by **https**.

**Figure 6-2** Configuring PyCharm Toolkit



**2. The hosts file is not configured or is incorrectly configured.**

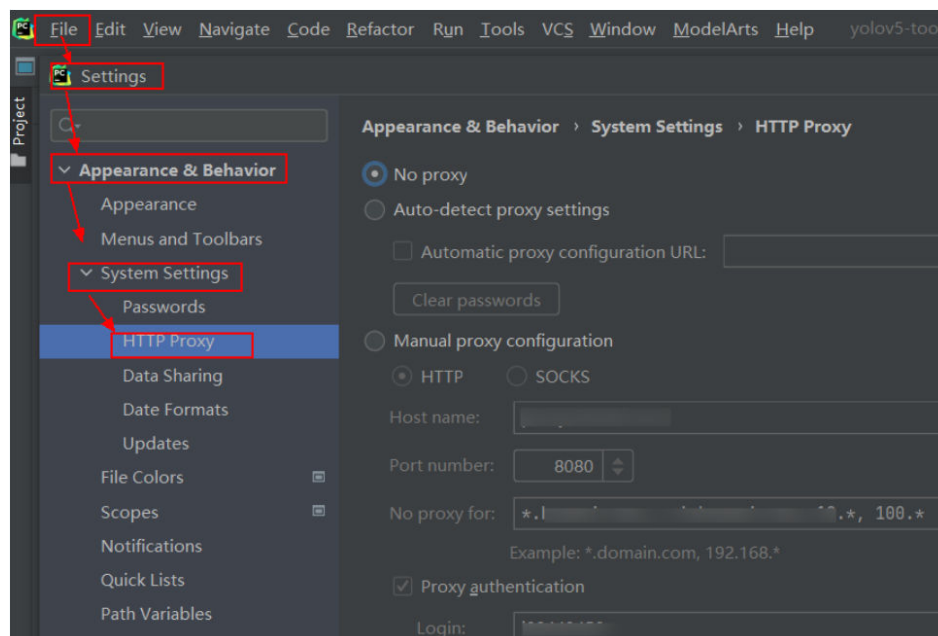
Configure the domain names and IP addresses in the **hosts** file on the local PC.

**Network proxy settings are incorrect.**

If the network requires proxy settings, check whether the proxy settings are correct. You can also use the mobile hotspot to test.

Check whether the proxy settings are correct.

**Figure 6-3** PyCharm network proxy settings



**4. The AK/SK is incorrect.**

The obtained AK/SK is incorrect. Obtain the correct AK/SK and try again. For details, see [Creating Access Keys \(AK and SK\)](#).



If you use a RightCloud account, contact the technical support of the region to obtain the AK/SK.

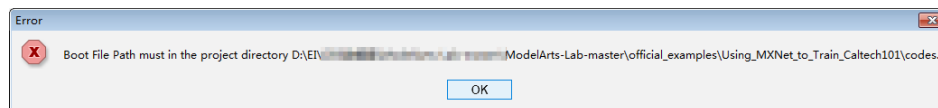
#### 5. The computer time is incorrectly set.

Set the computer time to the correct time.

## 6.3 Why Cannot I Start Training?

If code that does not belong to the used project is selected in a boot script, training cannot be started. The following figure shows error information. You are advised to add the boot script to the project or open the project where the boot script is located, and then start the training job.

Figure 6-4 Error



## 6.4 What Should I Do If Error "xxx isn't existed in train\_version" Occurs When a Training Job Is Submitted

### Symptom

Error "xxx isn't existed in train\_version" occurs when a training job is submitted. See the following figure.

Figure 6-5 Error "xxx isn't existed in train\_version"

```
Begin to check training configuration.
Begin to upload training code.
No modified code to upload.
Begin to get training job pre version.
Http Response Code : 400
{"is_success":false,"error_code":"ModelArts.0102","error_message":"193526 isn't existed in train_version."}
ModelArts Training is Finished.
```

### Possible Causes

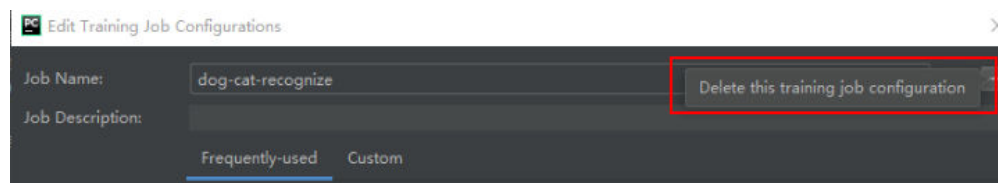
The preceding error occurs because the user logs in to the ModelArts management console and deletes the training job after submitting the training job using PyCharm ToolKit.

PyCharm Toolkit records the training job IDs of ModelArts on the cloud. If you manually delete the job on the ModelArts management console, a message is displayed indicating that the job with the ID cannot be found when you submit the job locally.

## Solution

If you have deleted a job on the ModelArts management console, you also need to delete the local configuration from ToolKit. To delete the local configuration, click **Edit Training Configuration**, find the job name, click the minus sign in the upper right corner, and confirm the deletion.

**Figure 6-6** Deleting the local configuration



In the displayed confirmation dialog box, confirm the information and click **Yes** to delete the configuration. After the deletion, you can create a training job configuration and submit the training job.

## 6.5 What Should I Do If an Error Occurs When I Submit a Training Job

When a training job is running, the "Invalid OBS path" error is reported.

**Figure 6-7** "Invalid OBS path" error

```
No modified code to upload.
Begin to get training job pre version.
Begin to create training job.
Http Response Code : 400
{"is_success":false,"error_code":"ModelArts.0404","error_message":"Invalid OBS path '/...'"}
Create training job failed.
ModelArts Training is Finished.
```

To locate the fault, perform the following operations:

- If you are using ModelArts for the first time, log in to the ModelArts management console and complete access authorization configuration. The agency authorization mode is recommended. After the global configuration is complete, submit the job again.
- Check whether the configured **Data Path in OBS** exists and whether data files exist in the directory. If the directory does not exist, create a directory on OBS and upload the training data to the directory.

## 6.6 What Should I Do If an Error Occurs During Service Deployment

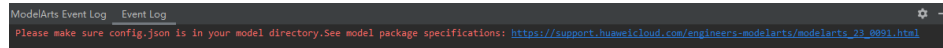
Before deploying a model as a service, you need to compile the configuration file and inference code based on the trained model.

If the **confi.json** configuration file or the **customize\_service.py** inference code is missing in the model storage path, an error is displayed, as shown in the following figure.

Solutions:

Write the configuration file and inference code, and save them to the OBS directory where the model to be deployed resides. For details, see [Introduction to Model Package Specifications](#).

**Figure 6-8** Error



## 6.7 How Do I View Error Logs of PyCharm ToolKit?

The error logs of PyCharm ToolKit are recorded in the **idea.log** file of PyCharm. For example, in the Windows operating system, the path of the **idea.log** file is **C:\Users\xxx\Idea\IC2019.2\system\log\idea.log**.

Search for **modelarts** in the log file to view all logs related to PyCharm ToolKit.