

Setting Up the Environment

Creating a PAI-DSW instance

[LMP-C01] LLM Engineer (Professional), makes use of **Platform for AI - Data Science Workshop (PAI-DSW)** and Model Studio to demonstrate the various functions and features of large language models (LLMs), as well as build RAGs, agents, and more. Before we get started with the course, you'll need to create your own PAI-DSW instance.

Instructions

1. Go to the [PAI-DSW console](#).

Note: Opens to the **Singapore** region by default, change to your region if required.

2. Click the **Create Instance** button.
3. Create an instance with the following configurations (minimum specs):

- **Chapters 1-7, 10, & 11**

These chapters use **Model Studio** in conjunction with **PAI-DSW**, so you only need a basic instance.

- **Instance Type:** ecs.g6.xlarge
- **Image:**
 - **Chip type:** CPU
 - **modelscope:** 1.28.0
 - **pytorch:** 2.3.1
 - **python:** 3.11
 - **OS:** ubuntu22.04

Note: You can use the selectors on the left to filter the images.

Alternatively, paste this into the search bar:

```
modelscope:1.28.0-pytorch2.3.1tensorflow2.16.1-gpu-py311-cu121-ubuntu22.04
```

- **Chapters 8 & 9**

These chapters will focus on fine-tuning and deploying LLMs on your **PAI-DSW** instance. We recommend that you use a GPU-accelerated instance to perform the labs for these chapters.

- **Instance Type:** ecs.gn7i-c8g1.2xlarge
- **Image:**
 - **Chip type:** GPU
 - **modelscope:** 1.28.0
 - **pytorch:** 2.3.1
 - **python:** 3.11

■ **OS:** ubuntu22.04

Note: You can use the selectors on the left to filter the images.

Alternatively, paste this into the search bar:

`modelscope:1.28.0-pytorch2.3.1tensorflow2.16.1-gpu-py311-cu121-ubuntu22.04`

4. Use the default settings for the rest of the configuration items, or set the up based on your preferences.

Checking environment prerequisites

[LMP-C01] LLM Engineer (Professional) requires Python 3.10 and above to run. You'll also need to make sure that your working folder is `/mnt/workspace` before continuing.

Note: You will need to run the commands in this tutorial in the PAI-DSW terminal.



Accessing the terminal in PAI-DSW

```
# Check Python version
python --version

# Check that we're in the /mnt/workspace directory
pwd

# If directory is not /mnt/workspace
cd /mnt/directory
```

Creating the environment

Once you've confirmed the prerequisites, run the following code in the **terminal**. It will set-up a virtual environment for the course.

```
# Create a virtual environment for the course
python3 -m venv LLM_pro

# Activate the virtual environment
source LLM_pro/bin/activate

# Upgrade pip
pip install --upgrade pip

# Install ipykernel
pip install ipykernel

# Registers a new Jupyter Notebook kernel and associate it with the
virtual environment
python -m ipykernel install --user --name LLM_pro --display-name "LLM
```

```
Enginner (Professional)"
```

```
# Install the requirements  
pip install -r requirements.txt
```

After everything is installed, you can opt to deactivate the environment. This step is not necessary.

```
# (Optional) Deactivate the virtual environment  
deactivate
```

Activating the PAI-DSW kernel

In PAI-DSW, the kernel is a separate process that runs and executes the code contained within the notebook. It acts as the computational engine for the notebook. Before you run any code in the course, check that the correct kernel is activated.



PAI-DSW showing the correct kernel

In the previous step, we created our kernel with the display name **LLM Enginner (Professional)**.

If the correct kernel is not activated, perform the following steps:

1. Click on the kernel name at the top-right of the PAI-DSW workspace.
2. A window appears. Select the correct kernel (default name: **LLM Enginner (Professional)**).
3. The kernel should now show the correct kernel.



Changing the kernel in PAI-DSW

Finding Your API Keys and AccessKeys

For most of the labs, you'll need access to the **Model Studio** API. To do that, you need to obtain an API Key. In some of the labs, you'll need to use your **RAM AccessKey**.

! Important:

Alibaba Cloud and Aliyun ("阿里云") have different account systems and endpoints. These are not cross-compatible.

Model Studio API key



Model Studio API-Key page

The copy API-Key icon is highlighted

1. Log on the [Model Studio console API-Key](#) page.

2. If you do not have an API key, click the **Create API Key** button at the top right of the screen.
3. Then, in the list of API keys, find the one you want to use, and click the **copy** icon.

Resource Access Management (RAM) AccessKey pair



RAM Users page

A RAM User is highlighted

1. Go to the [Users page](#) in the RAM console.
2. If you don't have an existing RAM User, click the **Create User** button, and follow the on-screen instructions to create a RAM User.
 - **Important:** During the create user process, you must select **Using permanent AccessKey to access** to obtain permissions on the API.
3. If you already have a RAM User, click on the name of the RAM User. You will be redirected to the details page.
4. On the RAM user details page, scroll to the bottom to see your AccessKey ID. You will not be able to see your AccessKey Secret.
5. To get a new AccessKey pair, click the **Create AccessKey** button.
6. In the window that appears, select **CLI** and click the **Continue** button.
7. A new window pops up with your **AccessKey pair**. You can download them and keep them in a secure place.



A window showing the created AccessKey pair