

## Cài đặt môi trường MinerAI

Tài liệu này hướng dẫn người dùng cài đặt môi trường python với các thư viện được sử dụng ở server, giúp cho việc chạy thử nghiệm code được chính xác hơn.

Trong này, chúng tôi sẽ hướng dẫn các bạn cài đặt môi trường theo 2 cách:

- Cài đặt môi trường trực tiếp trên máy tính:
  - o Ưu điểm: dễ thực hiện, quen thuộc với các bạn đã từng code và cài đặt python
  - o Nhược điểm: có thể sẽ có chênh lệch so với môi trường chạy thực tế do hệ điều hành khác nhau (Môi trường thực tế mà server sẽ run code của các bạn là Ubuntu Server 18.04)
- Cài đặt sử dụng docker:
  - o Ưu điểm: sao chép được môi trường code giống với môi trường chạy thực tế
  - o Nhược điểm: việc cài đặt docker có thể sẽ có khó khăn với một số hệ điều hành cũ

Nội dung cài đặt:

- Python 3.6.9 (Ubuntu) – python 3.7.4 (windows)
- Tensorflow 1.14.0
- Keras 2.3.1
- Numpy 1.18.4
- Pandas 0.15
- PyTorch 1.5.0

### 1. Run trực tiếp

#### a. Windows

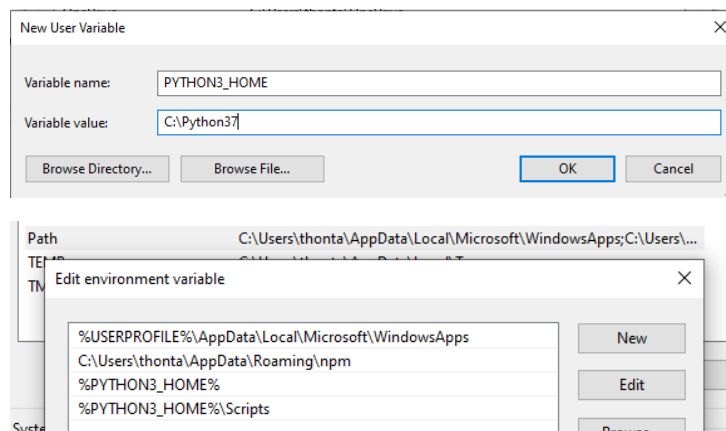
##### i. Cài đặt python 3.7.4:

Download bản executable installer cho OS tương ứng từ:

<https://www.python.org/downloads/release/python-374/>

(Link download được đặt ở mục Files phía cuối trang)

Thực hiện cài đặt và setup biến môi trường windows đến PYTHON\_HOME/, PYTHON\_HOME/Scripts



Kiểm tra cài đặt, đảm bảo version là 3.7.4

```
D:\>python37 -V
Python 3.7.4
```

#### ii. Cài đặt pip3

Chạy lệnh sau để thực hiện cài đặt: ***python37 -m pip install --upgrade pip***  
Kiểm tra cài đặt, đảm bảo pip3 được cài đặt trong Python37

```
D:\>which pip3
/cygdrive/c/Python37/Scripts/pip3
```

#### iii. Cài đặt virtualenv

Chạy lệnh sau để thực hiện cài đặt: ***pip3 install virtualenv***  
Kiểm tra cài đặt, đảm bảo virtualenv được cài đặt trong Python37

```
D:\>which virtualenv
/cygdrive/c/Python37/Scripts/virtualenv
```

#### iv. Cài đặt thư viện

Để không ảnh hưởng đến môi trường python chung, việc cài đặt sẽ được thực hiện trên môi trường ảo.

- Chuyển thư mục hiện thời về thư mục bạn muốn cài đặt và thực hiện Tạo môi trường ảo:

***virtualenv -p python37 {env\_name}***

{env\_name}: là tên môi trường do bạn tự chọn. Ví dụ, nếu bạn muốn đặt tên môi trường là miner thì lệnh cài đặt sẽ là: ***virtualenv -p python37 miner***

- Kích hoạt môi trường ảo  
***.\{env\_name}\Scripts\activate***
- Cài đặt thư viện:

***pip3 install numpy==1.18.4***

***pip3 install keras==2.3.1***

***pip3 install pandas==1.0.4***

***pip3 install tensorflow==1.14.0***

***pip3 install torch==1.5.0+cpu torchvision==0.6.0+cpu -f  
https://download.pytorch.org/whl/torch\_stable.html***

- Run code: code được run ở môi trường ảo mà bạn cài ở step trước, vì thế đừng quên kích hoạt môi trường ảo trước khi run nhé!

### b. Ubuntu 18.04

#### i. Cài đặt python 3.6.9

Các lệnh cài đặt:

***apt-get update***

***apt-get install python3***

(Tham khảo thêm cách cài đặt ở các version OS khác ở:

<https://askubuntu.com/questions/865554/how-do-i-install-python-3-6-using-apt-get>)

Kiểm tra cài đặt đúng version:

```
root@8ab45972b8ea:/# python3 -V
Python 3.6.9
```

## ii. Cài đặt pip

Run lệnh sau để cài đặt: ***python3 -m pip install --upgrade pip***

Hoặc: ***sudo apt install python3-pip***

## iii. Cài đặt virtualenv

Run lệnh sau để cài đặt: ***pip3 install virtualenv***

## iv. Cài đặt thư viện

Để không ảnh hưởng đến môi trường python chung, việc cài đặt sẽ được thực hiện trên môi trường ảo.

- Chuyển thư mục hiện thời về thư mục bạn muốn cài đặt và thực hiện Tạo môi trường ảo:

***virtualenv -p python3 {env\_name}***

{env\_name}: là tên môi trường do bạn tự chọn. Ví dụ, nếu bạn muốn đặt tên môi trường là miner thì lệnh cài đặt sẽ là: ***virtualenv -p python3 miner***

- Kích hoạt môi trường ảo

***cd {env\_name}/bin***

***source ./activate***

- Cài đặt thư viện:

***pip3 install numpy==1.18.4***

***pip3 install keras==2.3.1***

***pip3 install pandas==0.15***

***pip3 install tensorflow==1.14.0***

***pip3 install torch==1.5.0+cpu torchvision==0.6.0+cpu -f***

***[https://download.pytorch.org/whl/torch\\_stable.html](https://download.pytorch.org/whl/torch_stable.html)***

- Run code: code được run ở môi trường ảo mà bạn cài ở step trước, vì thế đừng quên kích hoạt môi trường ảo trước khi run nhé!

## 2. **Sử dụng docker**

Chúng tôi cung cấp cho các bạn docker image mà đã cài đặt đủ các môi trường giống như ở môi trường server thật.

Ở phần này, chúng tôi sẽ hướng dẫn cho bạn cách cài đặt docker, và cách sử dụng image mà chúng tôi cung cấp.

### a. **Cài đặt Docker**

Việc cài đặt Docker được hướng dẫn khá đầy đủ và chi tiết ở nhiều nguồn trên internet, vì thế chúng tôi xin phép chỉ cung cấp các link hướng dẫn mà chúng tôi đã thử nghiệm thực hiện theo thành công.

#### i. Windows:

- Windows 10: truy cập vào đường link dưới đây để download bộ cài và thực hiện cài đặt trên máy tính.

<https://hub.docker.com/editions/community/docker-ce-desktop-windows/>

- Windows 7: thực hiện việc cài đặt theo đường link sau <https://webme.ie/how-to-install-docker-on-windows-7/>

#### ii. Ubuntu:

- Ubuntu 18.04: Thực hiện cài đặt theo hướng dẫn ở đường link sau:

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04>

- Ubuntu 16.04: Thực hiện cài đặt theo hướng dẫn ở đường link sau:

<https://docs.docker.com/engine/install/ubuntu/>

## b. Sử dụng docker image

### i. Pull image:

Thực hiện lệnh sau để pull image về: **docker pull codelearnio/miner-ai:training**

Kiểm tra image đã được pull về thành công bằng lệnh: **docker images -a**

Thông tin được hiển thị ra sẽ bao gồm docker image có thông tin dưới đây:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
codelearnio/miner-ai	training	b71e2ea7dec6	12 minutes ago	2.45GB

### ii. Sử dụng image

Ở phần này, chúng tôi hướng dẫn bạn sử dụng 1 số lệnh cơ bản với docker container.

Để tìm hiểu thêm các lệnh khác, các bạn có thể tham khảo ở đường link sau:

<https://docs.docker.com/engine/reference/commandline/docker/>

#### 1. Tạo mới và run docker container từ image có sẵn:

Tạo và run docker container với docker image được cung cấp bằng lệnh sau: **docker run -it -v**

**{WORKING\_DIR}:/v b71e2ea7dec6**

Note: {WORKING\_DIR} là path dẫn đến thư mục chứa source của bạn;

Ví dụ: bạn để source code ở thư mục **D:\MinerAI** thì lệnh run của bạn sẽ là: **docker run -it -v**

**D:\MinerAI:/v b71e2ea7dec6**

Bạn có thể đặt tên container bằng cách thêm tham số: **--name={name}**

Chuyển thư mục hiện thời về thư mục đã được bind: **cd /v**

Kiểm tra các file đã được bind vào container: **ls**

```
D:\>docker run -it -v D:\projects\GameAI\Miner-Training-Local-CodeSample:/v --name=miner b71e2ea7dec6
root@f39ab8375c62:/# cd /v
root@f39ab8375c62:/v# ls
DONModel.py  GameSocketDummy.py  Memory.py  MinerState.py  TrainingClient.py  bot2.py
bot1.py      MinerEnv.py          LocalModel.py  bot1.py        bot3.py
```

Tại đây bạn có thể thực hiện lệnh run **python3** với source code của bạn mà không cần phải cài đặt gì thêm.

Ví dụ: **python3 TrainingClient.pyt**

#### 2. Kiểm tra các container đang có: **docker container ls -a**

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2f3d9797c028	b71e2ea7dec6	"/bin/bash"	6 seconds ago	Up 6 seconds		modest_sutherland
f39ab8375c62	b71e2ea7dec6	"/bin/bash"	2 minutes ago	Exited (0) 16 seconds ago		miner

Với hiển thị như ở hình trên, ta thấy đang có 2 container được khởi tạo từ image

**b71e2ea7dec6**, container **2f3d9797c028** đang running, container **f39ab8375c62** đã stop

#### 3. Attach docker đang chạy: **docker attach {container\_id}**

```
D:\>docker attach 2f3d9797c028
root@2f3d9797c028:/#
```

Note: bạn có thể thay thế {container\_id} bằng {container\_name}

#### 4. Start docker đang stop: **docker start -a {container\_id}**

```
D:\>docker start -a miner
root@f39ab8375c62:/#
```

#### 5. Stop container đang run: **docker stop {container\_id}**

#### 6. Remove docker container: **docker rm {container\_id}**