

# Hướng dẫn dùng docker để nộp bài cho Zalo AI Challenge 2025 - RoadBuddy - Understanding The Road Through Dashcam AI

1. Giới thiệu về Docker
2. Cài đặt Docker trên Ubuntu
3. Cài đặt Nvidia Docker / Driver
4. Nộp bài cho Zalo AI Challenge
5. Cấu trúc thư mục code
6. Bổ sung bắt buộc về Training Code & Tài liệu mô tả
7. Phụ lục: Cấu trúc viết Jupyter notebook để đo lường thời gian
8. Upload docker

## 1. Giới thiệu về Docker

**Docker** là một nền tảng giúp người dùng đóng gói và chạy chương trình của mình trên các môi trường khác nhau một cách nhanh nhất dựa trên các *container*.

**Docker Image** là một dạng tập hợp các tệp của ứng dụng, được tạo ra bởi Docker engine. Nội dung của các Docker image sẽ không bị thay đổi khi di chuyển. Docker image được dùng để chạy các Docker container.

**Docker Container** là một dạng runtime của các Docker image, dùng để làm môi trường chạy ứng dụng.

Hướng dẫn chi tiết các bạn có thể tham khảo tại: <https://docs.docker.com/get-started/>

## 2. Cài đặt Docker trên Ubuntu

Đối với các hệ điều hành khác, tham khảo cách cài đặt tại đây <https://docs.docker.com/install/overview/>

### 1. Cài đặt docker

```
| sudo apt-get install docker.io
```

### 2. Kiểm tra phiên bản docker

```
| sudo docker --version
```



Docker version 24.0.5, build ced0996

### 3. Chạy thử docker hello world

```
| sudo docker run hello-world
```

```
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

### 4. Một số câu lệnh phổ biến

- Liệt kê các images hiện có

```
| sudo docker images
```

- Liệt kê các container hiện có

```
| sudo docker ps -a
```

zdeploy@ZA_AI_GPU-34:~	\$	docker ps -a					
CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS	
1cef0b6d1f5a	hello-world		"/hello"	3 minutes ago	Exited (0) 3 minutes ago		

### 3. Cài đặt Nvidia Docker / Driver

Để sử dụng được GPU trên docker, bạn cần cài đặt Nvidia docker.

Hướng dẫn cài đặt Nvidia docker tại đây: <https://github.com/NVIDIA/nvidia-docker>

**Lưu ý:** Để tránh lỗi khi chấm, các đội cần cài đặt các thư viện **đồng bộ môi trường với server BTC**:

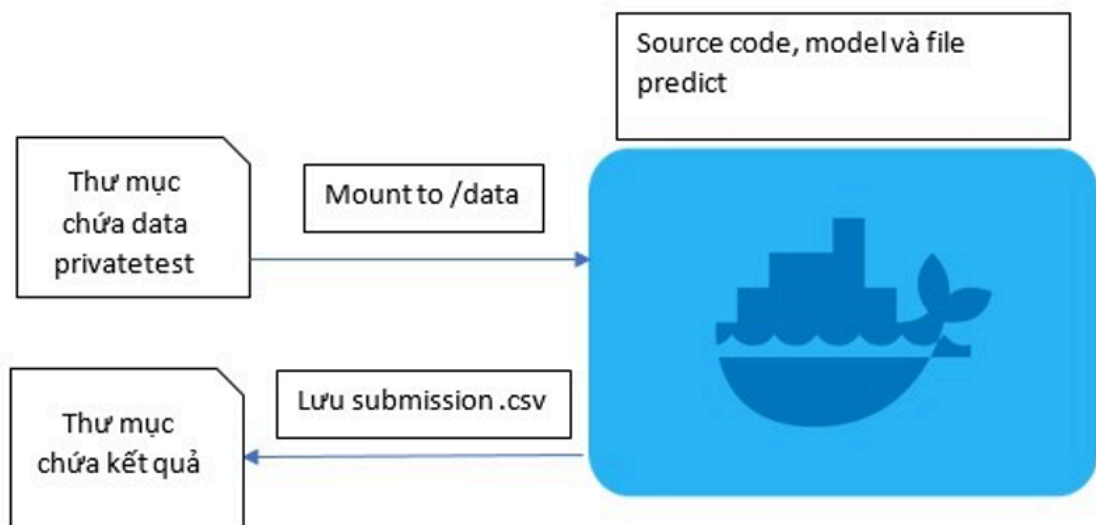
- **Driver Version: 535.86.10**
- **CUDA Version: 12.2**

Ví dụ nvidia docker image: **nvidia/cuda:12.2.0-devel-ubuntu22.04**

### 4. Nộp bài cho Zalo AI Challenge

**Các bước cần thiết để tạo được một Docker Image**

1. Tạo một docker container mới (hoặc sử dụng docker có sẵn).
2. Đưa model và source code đang có vào docker.
3. Install các packages và libraries cần thiết mà solution của bạn sử dụng để chạy.
4. Viết một file script tên predict.sh. File này chứa command các bước để chạy test. Nhận input từ **/data** và output ra **/result/submission.csv** (tùy theo format quy định của đề bài). Ngoài ra, xuất ra các thông số thời gian load model, thời gian predict ở stdout.
5. Nộp file predict **predict\_notebook.ipynb** để đo thời gian inference
6. Commit các thay đổi trong docker container
7. Save docker container thành 1 file image và nộp lên website cuộc thi.



## 5. Cấu trúc thư mục code

Ví dụ source code của bạn ở folder **/home/zdeploy/zac2025** với cấu trúc như sau.

```

|---- predict.py
|---- preprocessing.py
|---- saved_models
|    |---- models.safetensors # model cần đc copy vào trong docker
|---- train.py
|---- requirements.txt
|---- predict.sh
|---- start_jupyter.sh
|---- predict_notebook.ipynb # sử dụng để đánh giá thời gian, xem
|                             hướng dẫn viết notebook ở cuối
|---- training_code # chứa toàn bộ mã nguồn training và README.
|    |---- README.md
  
```

## 1. Khởi động docker container và đặt tên container với name **zac2025**

```
docker run --gpus '"device=0"' --network host -it --name zac2025  
nvidia/cuda:12.2.0-devel-ubuntu22.04 /bin/bash
```

- Bạn **bắt buộc** phải sử dụng image base có cuda 12 để tương thích với server chấm điểm của ban tổ chức.
- Cần phải có cờ **--network host** để thuận tiện run jupyter đo thời gian inference ở bước sau

```
$ docker run --gpus '"device=0"' --network host -it \  
> --name zac2025 \  
> --hostname zac2025 \  
> nvidia/cuda:12.2.0-devel-ubuntu22.04 \  
> /bin/bash  
  
=====  
== CUDA ==  
=====
```

CUDA Version 12.2.0

Container image Copyright (c) 2016-2023, NVIDIA CORPORATION & AFFILIATES. All rights reserved.

This container image and its contents are governed by the NVIDIA Deep Learning Container License.  
By pulling and using the container, you accept the terms and conditions of this license:  
<https://developer.nvidia.com/ngc/nvidia-deep-learning-container-license>

A copy of this license is made available in this container at /NGC-DL-CONTAINER-LICENSE for your convenience.

```
root@zac2025:/#
```

Lúc này trong container đang ở vị trí /

```
root@zac2025:/# pwd  
/
```

## 2. Kiểm tra container có sử dụng được GPU hay không bằng lệnh nvidia-smi

```
nvidia-smi
```

```

root@zac2025:/# nvidia-smi
Tue Nov 11 04:16:10 2025
+-----+
| NVIDIA-SMI 535.183.06                  Driver Version: 535.183.06      CUDA Version: 12.2   |
+-----+-----+
| GPU  Name           Persistence-M   Bus-Id        Disp.A    Volatile Uncorr. ECC |
| Fan  Temp    Perf       Pwr:Usage/Cap       Memory-Usage  GPU-Util  Compute M. |
|              |              |              |              |              |      MIG M. |
+-----+-----+
|  0  NVIDIA A30          On          00000000:30:00.0 Off      0%          0 |
| N/A   36C    P0          36W / 165W      824MiB / 24576MiB      Default |
|              |              |              |              |              |      Disabled |
+-----+-----+
+-----+
| Processes: |
| GPU   GI    CI          PID    Type    Process name                        GPU Memory |
|   ID   ID   ID              |    |              |              |      Usage |
+-----+-----+
root@zac2025:/#

```

3. **Mở một terminal mới**, copy source code ở folder hiện tại vào trong container với cú pháp.

```
sudo docker cp [source_path] [container_name]:[destination_path]
```

```
| sudo docker cp /home/zdeploy/AILab/zac2025/ zac2025:/code/
```

Lúc này toàn bộ source code ở ngoài đã được copy vào trong container ở vị trí /code trong **container**

```

root@zac2025:/# ls
NGC-DL-CONTAINER-LICENSE  boot  cuda-keyring_1.0-1_all.deb  etc  lib  lib64  media  opt  root  sbin  sys  usr
bin                        code  dev                          home  lib32  libx32  mnt  proc  run  srv  tmp  var
root@zac2025:/#

```

3. Cài python và các package cần thiết

```
| apt update; apt-get -y install libgl1-mesa-glx libglib2.0-0 vim; apt -y install python3-pip
```

Nhập câu trả lời cần thiết nếu được hỏi

```

1. Africa    3. Antarctica  5. Arctic    7. Atlantic  9. Indian    11. SystemV  13. Etc
2. America  4. Australia  6. Asia      8. Europe    10. Pacific  12. US
Geographic area: 6

Please select the city or region corresponding to your time zone.

1. Aden      16. Brunei      31. Hong_Kong  46. Kuala_Lumpur  61. Pyongyang  76. Tehran
2. Almaty    17. Chita       32. Hovd       47. Kuching        62. Qatar       77. Tel_Aviv
3. Amman     18. Choibalsan  33. Irkutsk    48. Kuwait         63. Qostanay    78. Thimphu
4. Anadyr    19. Chongqing   34. Istanbul  49. Macau          64. Qyzylorda   79. Tokyo
5. Aqttau    20. Colombo     35. Jakarta    50. Magadan        65. Rangoon     80. Toms
6. Aqtobe    21. Damascus    36. Jayapura   51. Makassar       66. Riyadh      81. Ujung_Pandang
7. Ashgabat  22. Dhaka       37. Jerusalem  52. Manila         67. Sakhalin    82. Ulaanbaatar
8. Atyrau    23. Dili        38. Kabul      53. Muscat         68. Samarkand   83. Urumqi
9. Baghdad   24. Dubai       39. Kamchatka  54. Nicosia        69. Seoul       84. Ust-Nera
10. Bahrain  25. Dushanbe    40. Karachi    55. Novokuznetsk  70. Shanghai    85. Vientiane
11. Baku     26. Famagusta   41. Kashgar    56. Novosibirsk    71. Singapore   86. Vladivostok
12. Bangkok  27. Gaza        42. Kathmandu  57. Omsk           72. Srednekolymsk 87. Yakutsk
13. Barnaul  28. Harbin      43. Khandyga   58. Oral           73. Taipei      88. Yangan
14. Beirut   29. Hebron      44. Kolkata    59. Phnom_Penh     74. Tashkent     89. Yekaterinburg
15. Bishkek  30. Ho_Chi_Minh 45. Krasnoyarsk 60. Pontianak      75. Tbilisi      90. Yerevan

Time zone: 30

Current default time zone: 'Asia/Ho_Chi_Minh'
Local time is now:      Fri Nov 24 14:52:01 +07 2023.
Universal Time is now:  Fri Nov 24 07:52:01 UTC 2023.

```

```

Selecting previously unselected package python3-setuptools.
Preparing to unpack .../14-python3-setuptools_59.6.0-1.2ubuntu0.22.04.3_all.deb ...
Unpacking python3-setuptools (59.6.0-1.2ubuntu0.22.04.3) ...
Selecting previously unselected package python3-wheel.
Preparing to unpack .../15-python3-wheel_0.37.1-2ubuntu0.22.04.1_all.deb ...
Unpacking python3-wheel (0.37.1-2ubuntu0.22.04.1) ...
Selecting previously unselected package python3-pip.
Preparing to unpack .../16-python3-pip_22.0.2+dfsg-1ubuntu0.7_all.deb ...
Unpacking python3-pip (22.0.2+dfsg-1ubuntu0.7) ...
Setting up javascript-common (11+nmu1) ...
Setting up python3.10 (3.10.12-1~22.04.11) ...
Setting up libexpat1-dev:amd64 (2.4.7-1ubuntu0.6) ...
Setting up zlib1g-dev:amd64 (1:1.2.11.dfsg-2ubuntu9.2) ...
Setting up libjs-jquery (3.6.0+dfsg+~3.5.13-1) ...
Setting up libpython3-stdlib:amd64 (3.10.6-1~22.04.1) ...
Setting up libjs-underscore (1.13.2~dfsg-2) ...
Setting up python3 (3.10.6-1~22.04.1) ...
running python rtupdate hooks for python3.10...
running python post-rtupdate hooks for python3.10...
Setting up libpython3.10-dev:amd64 (3.10.12-1~22.04.11) ...
Setting up libjs-sphinxdoc (4.3.2-1) ...
Setting up python3.10-dev (3.10.12-1~22.04.11) ...
Setting up python3-lib2to3 (3.10.8-1~22.04) ...
Setting up python3-pkg-resources (59.6.0-1.2ubuntu0.22.04.3) ...
Setting up python3-distutils (3.10.8-1~22.04) ...
Setting up libpython3-dev:amd64 (3.10.6-1~22.04.1) ...
Setting up python3-setuptools (59.6.0-1.2ubuntu0.22.04.3) ...
Setting up python3-wheel (0.37.1-2ubuntu0.22.04.1) ...
Setting up python3-dev (3.10.6-1~22.04.1) ...
Setting up python3-pip (22.0.2+dfsg-1ubuntu0.7) ...
root@zac2025:/# which python3
/usr/bin/python3
root@zac2025:/# python3 -V
Python 3.10.12
root@zac2025:/#

```

### 3. Cài đặt các thư viện cần thiết từ file requirements.txt hoặc tự install **trong container**

Trước hết chuyển tới thư mục code bằng lệnh

```
cd /code
```

```
pip install jupyterlab
```

```
pip install -r requirements.txt
```

```
pip install numpy
```

Lưu ý môi trường bắt buộc phải cài đặt **jupyterlab** để chấm thời gian inference

```
root@zac2025:/# pip install jupyterlab
Collecting jupyterlab
  Downloading jupyterlab-4.4.10-py3-none-any.whl (12.3 MB)
    12.3/12.3 MB 81.9 MB/s eta 0:00:00
Collecting jupyter-lsp>=2.0.0
  Downloading jupyter_lsp-2.3.0-py3-none-any.whl (76 kB)
    76.7/76.7 KB 52.3 MB/s eta 0:00:00
Collecting jupyter-server<3,>=2.4.0
  Downloading jupyter_server-2.17.0-py3-none-any.whl (388 kB)
    388.2/388.2 KB 117.4 MB/s eta 0:00:00
Collecting notebook-shim>=0.2
  Downloading notebook_shim-0.2.4-py3-none-any.whl (13 kB)
Collecting Jinja2>=3.0.3
  Downloading Jinja2-3.1.6-py3-none-any.whl (134 kB)
    134.9/134.9 KB 327.2 KB/s eta 0:00:00
Collecting tornado>=6.2.0
  Downloading tornado-6.5.2-cp39-abi3-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (443 kB)
    443.9/443.9 KB 49.0 MB/s eta 0:00:00
Collecting jupyter-core
  Downloading jupyter_core-5.9.1-py3-none-any.whl (29 kB)
Collecting ipykernel!=6.30.0,>=6.5.0
  Downloading ipykernel-7.1.0-py3-none-any.whl (117 kB)
    118.0/118.0 KB 41.3 MB/s eta 0:00:00
Collecting httpx<1,>=0.25.0
  Downloading httpx-0.28.1-py3-none-any.whl (73 kB)
    73.5/73.5 KB 39.7 MB/s eta 0:00:00
Collecting packaging
  Downloading packaging-25.0-py3-none-any.whl (66 kB)
    66.5/66.5 KB 33.2 MB/s eta 0:00:00
Collecting tomli>=1.2.2
```

```
root@zac2025:/code# pip install -r requirements.txt
Collecting transformers==4.57.0
  Downloading transformers-4.57.0-py3-none-any.whl (12.0 MB)
    12.0/12.0 MB 86.7 MB/s eta 0:00:00
Collecting torch
  Downloading torch-2.9.0-cp310-cp310-manylinux_2_28_x86_64.whl (899.8 MB)
    899.8/899.8 MB 4.6 MB/s eta 0:00:00
Collecting torchvision
  Downloading torchvision-0.24.0-cp310-cp310-manylinux_2_28_x86_64.whl (8.0 MB)
    8.0/8.0 MB 170.5 MB/s eta 0:00:00
Collecting accelerate
  Downloading accelerate-1.11.0-py3-none-any.whl (375 kB)
    375.8/375.8 KB 120.6 MB/s eta 0:00:00
Collecting vllm
  Downloading vllm-0.11.0-cp38-abi3-manylinux1_x86_64.whl (438.2 MB)
    438.2/438.2 MB 8.4 MB/s eta 0:00:00
Collecting filelock
  Downloading filelock-3.20.0-py3-none-any.whl (16 kB)
Collecting regex!=2019.12.17
  Downloading regex-2025.11.3-cp310-cp310-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl (791 kB)
    791.7/791.7 KB 165.8 MB/s eta 0:00:00
Collecting tokenizers<0.23.0,>=0.22.0
  Downloading tokenizers-0.22.1-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.3 MB)
    3.3/3.3 MB 39.9 MB/s eta 0:00:00
Collecting tqdm>=4.27
  Downloading tqdm-4.67.1-py3-none-any.whl (78 kB)
    78.5/78.5 KB 44.7 MB/s eta 0:00:00
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers==4.57.0->-r requirements.txt (line 1)) (2.32.5)
Collecting numpy>=1.17
  Downloading numpy-2.2.6-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.8 MB)
    16.8/16.8 MB 123.7 MB/s eta 0:00:00
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers==4.57.0->-r requirements.txt (line 1)) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers==4.57.0->-r requirements.txt (line 1)) (6.0.3)
Collecting huggingface-hub<1.0,>=0.34.0
```

#### 4. Chỉnh sửa file **/code/predict.sh**

Ví dụ có 2 bước để chạy bộ test ra kết quả.

```
python3 /code/preprocessing.py
```

```
python3 /code/predict.py
```



File predict.py phải ghi kết quả ở **/result/submission.csv** (trong container, **code phải tự tạo folder /result** nếu folder chưa tồn tại)

```
| sh /code/predict.sh
```

```
root@zac2025:/code# sh predict.sh
preprocessing...
output will be saved in /result/submission.csv
```

5. Chỉnh sửa file /code/start\_jupyter.sh

```
jupyter lab --port 9777 --ip 0.0.0.0 --NotebookApp.password='zac2025' -
-NotebookApp.token='zac2025' --allow-root --no-browser
```

6. Chỉnh sửa file **predict\_notebook.ipynb**: Xem phụ lục ở cuối

7. Sau khi hoàn thành predict tiến hành lưu lại các thay đổi trên container. (**thao tác ở terminal**)

```
sudo docker commit [CONTAINER ID] [image name]:[tag name]
```

```
| sudo docker commit zac2025 zac2025:v1
```

```
$ docker commit zac2025 zac2025:v1
sha256:20b5dbae68ca2e91290a3acb868db7ea81fa54170827d0e98335ea8a0aba2b75
```

10. Kiểm tra docker lần cuối

**10.1 Kiểm tra hàm predict để xuất submission.csv**

Chạy predict.sh (các tham số về hyperparameters phải được set default trong file **predict.py**, BTC không chịu trách nhiệm nếu các bạn muốn thay đổi tham số này sau khi nộp bài)

Cấu trúc thư mục /data như sau:



```
|---- private_test.json
|---- videos/ # thư mục chứa video
```

`sudo docker run -v [path to test data]:/data -v [current dir]:/result [docker name]`

```
sudo docker run --gpus '"device=0"' -v /data:/data -v
/home/zdeploy/zac2025:/result zac2025:v1 /bin/bash /code/predict.sh
```

```
$ docker run --gpus '"device=0"' -v /data:/data -v /home/zdeploy/zac2025:/result zac2025:v1 /bin/bash /code/predict.sh
=====
==  CUDA  ==
=====

CUDA Version 12.2.0

Container image Copyright (c) 2016-2023, NVIDIA CORPORATION & AFFILIATES. All rights reserved.

This container image and its contents are governed by the NVIDIA Deep Learning Container License.
By pulling and using the container, you accept the terms and conditions of this license:
https://developer.nvidia.com/ngc/nvidia-deep-learning-container-license

A copy of this license is made available in this container at /NGC-DL-CONTAINER-LICENSE for your convenience.

preprocessing...
output will be saved in /result/submission.csv
```

```
$ pwd && ls
/home/zdeploy/zac2025
predict.py predict.sh preprocessing.py requirements.txt submission.csv
```

Kết quả của file submission.csv sẽ được trả về ở thư mục  
**/home/zdeploy/zac2025**

## 10.2 Kiểm tra jupyter có chạy được không.

- Chạy start\_jupyter.sh

```
sudo docker run -it --gpus '"device=0"' -p 9777:9777 -v /data:/data -v
/home/zdeploy/zac2025:/result zac2025:v1 /bin/bash
/code/start_jupyter.sh
```

- Truy cập local host ở port 9777
- Nhập mật khẩu **zac2025**

Password or token:

Log in

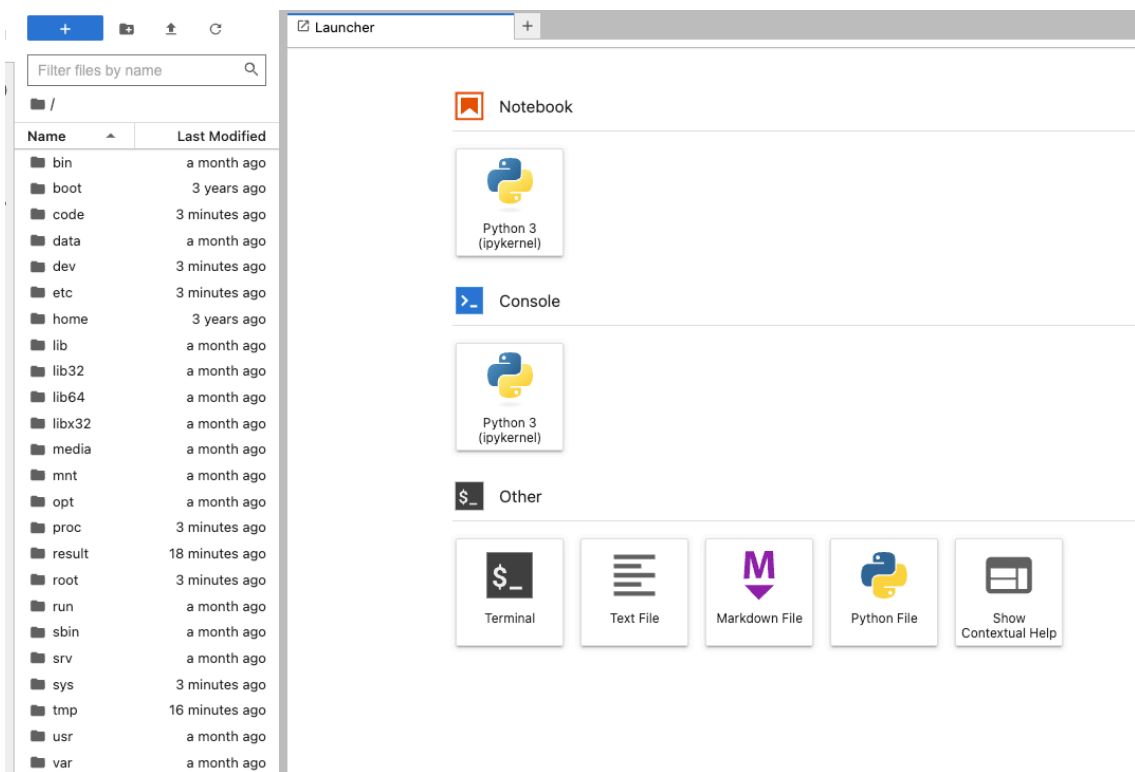
## Token authentication is enabled

If no password has been configured, you need to open the server with its login token in the URL, or paste it above. This requirement will be lifted if you [enable a password](#).

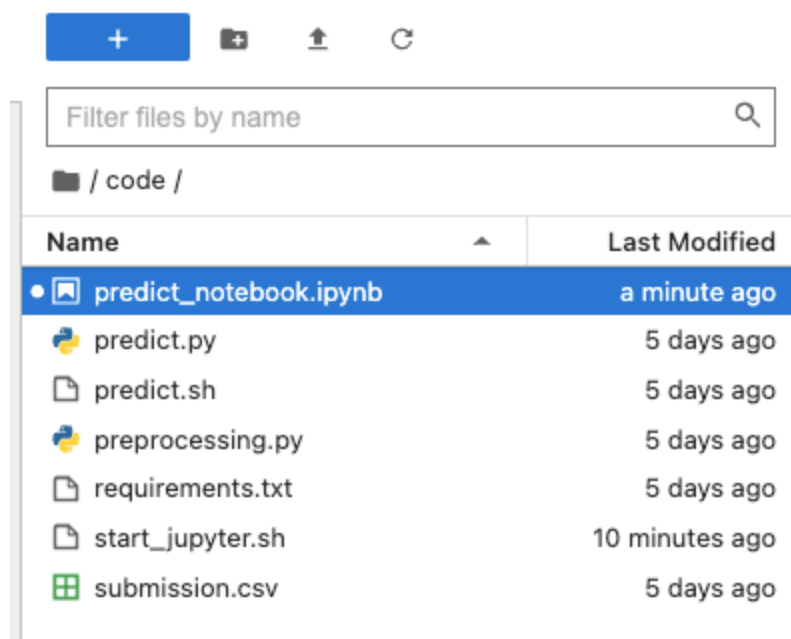
The command:

```
jupyter server list
```

- Giao diện cuối cùng sau khi cài đặt môi trường thành công



- Trong folder code phải có file predict\_notebook.ipynb với nội dung như phần **phụ lục**



8. Xuất image ra và nộp bài.

```
sudo docker save -o [output file] [image id]
```

```
sudo docker save -o zac2025_TeamName.tar.gz zac2025:v1
```

File **zac2025\_TeamName.tar.gz** là file bạn cần nộp.

```
$ ls
predict_notebook.ipynb  predict.py  predict.sh  preprocessing.py  requirements.txt  start_jupyter.sh  submission.csv  zac2025.tar.gz
```

## 6. Bổ sung bắt buộc về Training Code & Tài liệu mô tả

### 1. README.md mô tả ý tưởng

Các đội điền thêm **README.md** trình bày ngắn gọn về ý tưởng bao gồm training và inference của code bao gồm những gì?

### 2. Cung cấp đầy đủ Training Code & Data

BTC sẽ sử dụng code và data trong thư mục **training\_code/** để reproduce lại quá trình training nhằm kiểm tra tính nhất quán của kết quả.

Vì BTC sẽ chạy reproduce có internet, các đội có thể:

- Upload **training\_data** và **models** lên HuggingFace hoặc dịch vụ tương tự
- Trong training\_code, cần ghi rõ URL tải xuống.

**Lưu ý:** dữ liệu và mô hình trên các nền tảng không được thay đổi sau deadline nộp docker. **Base model** dùng trong quá trình training **không được đưa vào bên trong Docker** để tránh làm Docker quá nặng.

### 3. Cố định seed

Để bảo đảm reproducibility, các đội phải:

- Set seed cố định trong training và inference.
- Đảm bảo mô hình reproduce lại từ code/data cho ra kết quả giống mô hình dùng trong inference.

## 7. Phụ lục: Cấu trúc viết Jupyter notebook để đo lường thời gian

Bên cạnh các tài liệu hướng dẫn thực thi các mã nguồn huấn luyện và dự đoán. Mỗi team cần chuẩn bị một Jupyter notebook để thuận tiện cho việc đánh giá thời gian. BTC sẽ dùng notebook này để đo lường thời gian chạy của mỗi test case.

Nội dung của file notebook này sẽ tương tự như file **predict.py**, BTC sẽ tính điểm thời gian từ bước load file cho tới bước inference ra kết quả cho mỗi testcase.

Notebook này sẽ thể hiện các bước của việc dự đoán được thực hiện tách rời nhau như: (1) Nạp mô hình và tài nguyên; (2) đọc dữ liệu test cases (từng test case 1) (3) thực hiện dự đoán. Do vậy notebook này phải gồm ít nhất 3 ô (cell) tương ứng với 3 bước chính:

- Bước 1: **set seed cố định (đội thi tự chọn seed, điền seed vào trong notebook)** và BTC chỉ thực hiện run lấy kết quả

```
import os
import torch
import random
import numpy as np

def seed_everything(seed=42):
    random.seed(seed)
    os.environ['PYTHONHASHSEED'] = str(seed)
    np.random.seed(seed)
```

```
torch.manual_seed(seed)
torch.cuda.manual_seed(seed)
torch.cuda.manual_seed_all(seed)
torch.backends.cudnn.deterministic = True
torch.backends.cudnn.benchmark = False
```

```
seed_everything(42) # Ví dụ cho seed bằng 42
```

- Bước 2: Nạp mô hình và các tài nguyên cần thiết

```
# không dùng API ngoài, vì inference sẽ không có internet
model = ...
```

- Bước 3: Đọc nội dung các test case từ thư mục đầu vào

```
# read all test cases
test_cases = ...
```

- Bước 4: Thực hiện dự đoán và in ra kết quả cũng như thời gian.

```
#
all_predicted_time = []
all_result = []
for file_name in test_cases: # hoặc for item in test_cases tùy theo từng task
    t1 = time()
    input_ = preprocess(file_name) # convert file_path into expected input
    forward = model.predict(input_) # model forward
    result = postprocess(forward) # postprocess to expected format result
    t2 = time()
    predicted_time = int(t2*1000 - t1*1000)
    all_predicted_time.append((file_name, predicted_time))
    all_result.append(result)
```

```
write_predict_file(all_result) # output with the same format as predict.py
write_time_file(all_predicted_time) # write to time_submission.csv
```

### (Mã nguồn mang tính chất tham khảo tùy theo từng task)

**Lưu ý:** BTC chỉ mở file notebook và run all cells, nhiệm vụ của các bạn phải đảm bảo các cells khi chạy không bị lỗi ở bước nào, và cuối cùng phải xuất ra được 2 files:

1. **time\_submission.csv** với format gồm 3 cột là **id**, **answer**, và **time (millisecond)**
2. **jupyter\_submission.csv** với nội dung tương tự file **submission.csv**

Để tránh bị overwrite kết quả khi chạy bước predict.py, đối với các kết quả được xuất ra khi chạy jupyter, các bạn phải thêm prefix là **"jupyter\_"**

Các đội phải đảm bảo khi BTC chấm điểm thì điểm của file jupyter\_submission.csv sẽ phải giống với submission.csv thì mới bắt đầu chấm điểm qua phần time\_submission.csv

## 8. Upload docker

1. Lấy checksum MD5 của file docker **zac2025\_TeamName.tar.gz** đã nén ở mục

Tham khảo thêm cách check MD5 ở Windows và MacOS:

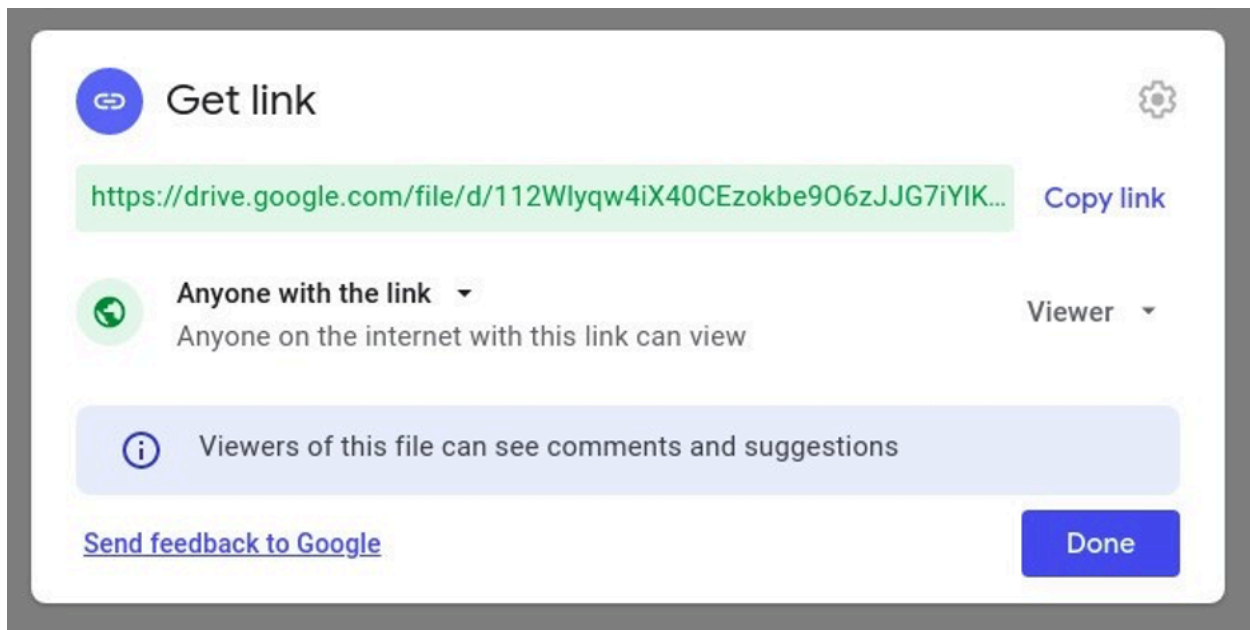
<https://portal.nutanix.com/page/documents/kbs/details?targetId=kA07V000000LWYqS>

Linux: <https://www.geeksforgeeks.org/md5sum-linux-command/>

2. Upload docker lên google drive

BTC sẽ download docker của các bạn về máy chủ để tiến hành bước kiểm tra kết quả cuối cùng.

Upload file docker đã nén lên dịch vụ storage của Google drive chỉnh sang chế độ share "Anyone with the link"



3. Nộp **link google drive** và **checksum** theo thông báo nộp bài của BTC  
Chúc các bạn thành công!