Yêu cầu :

@All Các bạn sẽ bắt đầu step tiếp theo: deploy model

Tutorials:

1. Gọi API: [tutorials/Quick\_Deploy/ONNX/README.md at main · triton-inference-server/tutorials](https://github.com/triton-inference-server/tutorials/blob/main/Quick_Deploy/ONNX/README.md)
2. Hosting: [server/docs/getting\_started/quickstart.md at main · triton-inference-server/server](https://github.com/triton-inference-server/server/blob/main/docs/getting_started/quickstart.md)
3. Đo performance: [server/docs/perf\_benchmark/perf-analyzer-README.rst at main · triton-inference-server/server](https://github.com/triton-inference-server/server/blob/main/docs/perf_benchmark/perf-analyzer-README.rst)

Reference:

[GitHub - triton-inference-server/server: The Triton Inference Server provides an optimized cloud and](https://github.com/triton-inference-server/server?tab=readme-ov-file)

[GitHub - triton-inference-server/tutorials: This repository contains tutorials and examples for Trit](https://github.com/triton-inference-server/tutorials/tree/main)

Yêu cầu:

1. down được image của triton server và hosting
2. viết file config để hosting model
3. dùng Tutorials link 1 để gọi API và chạy inference -> ra được kết quả là done
4. push code github (có gì push nấy)
5. (Extra - 10đ) Đo performance với toàn bộ config có thể chỉnh sửa từ 2) - Tutorials Link 3

Deadline là Thứ 4 (19/2/2025)

\*\*1. Tải và chạy Triton Inference Server:\*\*

1:đăng nhập <https://catalog.ngc.nvidia.com/orgs/nvidia/containers/tritonserver> với tài khoản

2+ dùng lệnh docker login nvcr.io

3: tạo API <https://org.ngc.nvidia.com/setup/api-key>

Và đăng nhập

Username: $oauthtoken

Password: <Your Key>

4: docker pull nvcr.io/nvidia/tritonserver:24.02-py3-min để tải image

Hiện Digest: sha256:f5aa202c99df0259198cab8def151f9c68f6add2e6f0a8d38a197bf1d39e8550

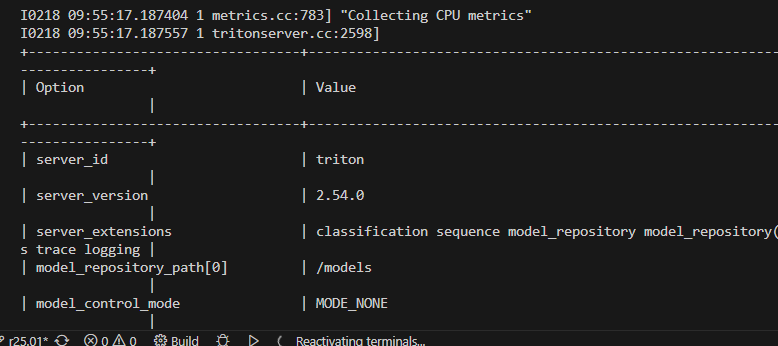
Status: Downloaded newer image for nvcr.io/nvidia/tritonserver:24.02-py3-min

nvcr.io/nvidia/tritonserver:24.02-py3-min là thành công cho 25.01

5: ta chạy lệnh

docker run --gpus=1 --rm --net=host -v D:\SourceCode\ProjectOJT\OJT\_TASK3\_DEPLOY\server\docs\examples\model\_repository:/models nvcr.io/nvidia/tritonserver:25.01-py3 tritonserver --model-repository=/models

ta đươc :



6:đăng nhập docker:

docker login --username hoanguyenthanh07 --password dckr\_pat\_6gGjC3VNUuXl3mPq5DJ1G\_jRkLw

Chào bạn, tôi sẽ giúp bạn thực hiện các yêu cầu trên. Đây là hướng dẫn chi tiết:

\*\*1. Download và Hosting Triton Server Image:\*\*

Bạn đã tải Triton 25.01, vậy là xong bước này. Nếu bạn muốn tải lại hoặc kiểm tra, đây là lệnh:

```bash

docker pull nvcr.io/nvidia/tritonserver:25.01-py3

```

Để hosting Triton, bạn cần chạy container từ image này. Lệnh như sau (đã được chỉnh sửa để dễ sử dụng):

```bash

docker run --gpus all --rm -p8000:8000 -p8001:8001 -p8002:8002 -v /path/to/your/model\_repository:/models nvcr.io/nvidia/tritonserver:25.01-py3 tritonserver --model-repository=/models

```

\* \*\*`--gpus all`\*\*: Cho phép Triton sử dụng tất cả GPU có sẵn. Nếu bạn muốn chỉ định số lượng GPU, hãy thay bằng `--gpus <number>`. Nếu không có GPU, bỏ tùy chọn này để chạy trên CPU (sẽ chậm hơn).

\* \*\*`--rm`\*\*: Xóa container sau khi dừng.

\* \*\*`-p8000:8000 -p8001:8001 -p8002:8002`\*\*: Map các cổng (ports) của container ra host. 8000 là HTTP, 8001 là GRPC, 8002 là metrics.

\* \*\*`-v /path/to/your/model\_repository:/models`\*\*: Gắn (mount) thư mục chứa model repository của bạn vào thư mục `/models` trong container. \*\*Quan trọng:\*\* Thay `/path/to/your/model\_repository` bằng đường dẫn thực tế đến thư mục chứa models của bạn trên máy tính.

\* \*\*`tritonserver --model-repository=/models`\*\*: Chạy lệnh `tritonserver` và chỉ định đường dẫn đến model repository bên trong container.

\*\*2. Viết File Config cho Model (model.config):\*\*

Triton sử dụng file `config.pbtxt` (hoặc `model.config`) để định nghĩa các thông tin về model. Bạn cần tạo file này trong thư mục của mỗi model. Ví dụ, nếu bạn có một model tên là `my\_model`, thì file config sẽ nằm ở `/path/to/your/model\_repository/my\_model/1/config.pbtxt` (thư mục `1` là version của model).

Đây là một ví dụ đơn giản cho TensorFlow:

```protobuf

name: "my\_model"

platform: "tensorflow\_savedmodel"

max\_batch\_size: 16

input [

{

name: "input\_1"

data\_type: TYPE\_FP32

dims: [ 1, 224, 224, 3 ] # Ví dụ: batch size, height, width, channels

}

]

output [

{

name: "output\_1"

data\_type: TYPE\_FP32

dims: [ 1000 ] # Ví dụ: 1000 classes

}

]

```

\* \*\*`name`\*\*: Tên model (phải khớp với tên thư mục).

\* \*\*`platform`\*\*: Loại framework. Các giá trị phổ biến: `tensorflow\_savedmodel`, `onnxruntime\_onnx`, `pytorch\_libtorch`.

\* \*\*`max\_batch\_size`\*\*: Kích thước batch tối đa. `0` nghĩa là không có batching.

\* \*\*`input`\*\*: Mảng các định nghĩa input.

\* `name`: Tên input (quan trọng, phải khớp với tên input trong model của bạn).

\* `data\_type`: Kiểu dữ liệu. Các giá trị phổ biến: `TYPE\_FP32`, `TYPE\_INT32`, `TYPE\_STRING`.

\* `dims`: Kích thước (dimensions) của input.

\* \*\*`output`\*\*: Mảng các định nghĩa output. Tương tự như `input`.

\*\*Lưu ý quan trọng:\*\*

\* Bạn cần điều chỉnh file config này cho phù hợp với \*model\* của bạn. Hãy kiểm tra model của bạn để biết tên input/output, kiểu dữ liệu và kích thước. Nếu không, Triton sẽ báo lỗi.

\* Bạn có thể tìm thêm ví dụ config trong thư mục `docs/examples/model\_repository` của Triton (nếu bạn đã clone source code).

\*\*3. Gọi API và Chạy Inference (Tutorials Link 1):\*\*

Tutorials Link 1 thường là Quick Deploy. Bạn sẽ sử dụng `image\_client` example.

Đầu tiên, bạn cần tải Triton SDK image:

```bash

docker pull nvcr.io/nvidia/tritonserver:25.01-py3-sdk

```

Sau đó, chạy container SDK:

```bash

docker run -it --rm --net=host nvcr.io/nvidia/tritonserver:25.01-py3-sdk

```

Bên trong container SDK, bạn có thể chạy `image\_client` (hoặc một client khác) để gửi request đến Triton server.

```bash

/workspace/install/bin/image\_client -m my\_model -c 3 -s INCEPTION /workspace/images/mug.jpg

```

\* \*\*`-m my\_model`\*\*: Tên model (phải khớp với `name` trong `config.pbtxt`). Thay `my\_model` bằng tên model của bạn.

\* \*\*`-c 3`\*\*: Yêu cầu top 3 classifications.

\* \*\*`-s INCEPTION`\*\*: Preprocessing scheme. Có thể cần thay đổi tùy thuộc vào model của bạn. Thường có các lựa chọn như `NONE`, `INCEPTION`, `RESNET`.

\* `/workspace/images/mug.jpg`: Đường dẫn đến ảnh. Thay bằng đường dẫn đến ảnh bạn muốn dùng. Hoặc copy ảnh vào `/workspace/images` trong container.

Nếu mọi thứ hoạt động, bạn sẽ thấy kết quả trả về.

\*\*4. Push Code lên Github:\*\*

Tạo một repository trên Github. Sau đó, trong thư mục chứa model repository của bạn, chạy:

```bash

git init

git add .

git commit -m "Initial commit"

git remote add origin <your\_github\_repository\_url>

git push -u origin main

```

Thay `<your\_github\_repository\_url>` bằng URL của repository Github của bạn.

\*\*5. (Extra) Đo Performance (Tutorials Link 3):\*\*

Tutorials Link 3 thường liên quan đến Performance Analyzer.

\* \*\*Tải Performance Analyzer Image:\*\*

```bash

docker pull nvcr.io/nvidia/tritonserver:25.01-py3-performance

```

\* \*\*Chạy Performance Analyzer Container:\*\*

```bash

docker run -it --rm --net=host nvcr.io/nvidia/tritonserver:25.01-py3-performance

```

\* \*\*Sử Dụng Performance Analyzer:\*\*

Bên trong container, bạn có thể sử dụng công cụ `perf\_analyzer`. Ví dụ:

```bash

/workspace/install/bin/perf\_analyzer -m my\_model -u localhost:8001 -i grpc -b 1 -p 1000

```

\* \*\*`-m my\_model`\*\*: Tên model.

\* \*\*`-u localhost:8001`\*\*: URL của Triton server (GRPC port).

\* \*\*`-i grpc`\*\*: Giao thức (protocol) là GRPC. Có thể dùng `http` cho HTTP.

\* \*\*`-b 1`\*\*: Batch size ban đầu.

\* \*\*`-p 1000`\*\*: Số lượng inference request.

\*\*Các Config Có Thể Chỉnh Sửa (từ `model.config`):\*\*

\* \*\*`max\_batch\_size`\*\*: Ảnh hưởng lớn đến throughput và latency. Thử các giá trị khác nhau (nhớ restart Triton sau khi thay đổi).

\* \*\*`instance\_group`\*\*: Cho phép bạn chỉ định số lượng instances của model và nơi chạy (GPU hoặc CPU). Ví dụ:

```protobuf

instance\_group [

{

count: 1

kind: KIND\_GPU

gpus: [ 0 ] # GPU ID

}

]

```

\* \*\*`dynamic\_batching`\*\*: Cho phép Triton tự động gom các request nhỏ thành batch lớn hơn để tăng hiệu suất.

\*Thực hiện theo các bước trên và điều chỉnh các thông số cấu hình cho phù hợp với model và phần cứng của bạn để đạt được hiệu suất tốt nhất.\*