

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN**



ĐỒ ÁN TỐT NGHIỆP

**HƯỚNG DẪN KẾT NỐI CONTAINER TRONG LABTAINER VỚI
OLLAMA TRÊN KAGGLE TRONG CÁC BÀI THỰC HÀNH LLM**

Sinh viên thực hiện:

B21DCAT111

Lý Quốc Khánh

Khóa: 2021 – 2026

Hệ: Đại học chính quy, ngành An toàn thông tin

Giảng viên hướng dẫn: PGS.TS. Nguyễn Ngọc Điệp

HÀ NỘI 12-2025

MỤC LỤC

MỤC LỤC	ii
DANH MỤC CÁC HÌNH VẼ.....	iii
1.1 Truy cập link Kaggle	1
1.2 Cài đặt mô hình chạy trên ollama của Kaggle.....	2
1.2.1 Trường hợp có 1 mô hình.....	2
1.2.2 Trường hợp có 2 mô hình.....	3
1.3 Cài đặt để chạy được ngrok	3
1.4 Khởi chạy file code trên Kaggle	4
1.5 Thay thế giá trị ngrok url vào labtainer	5

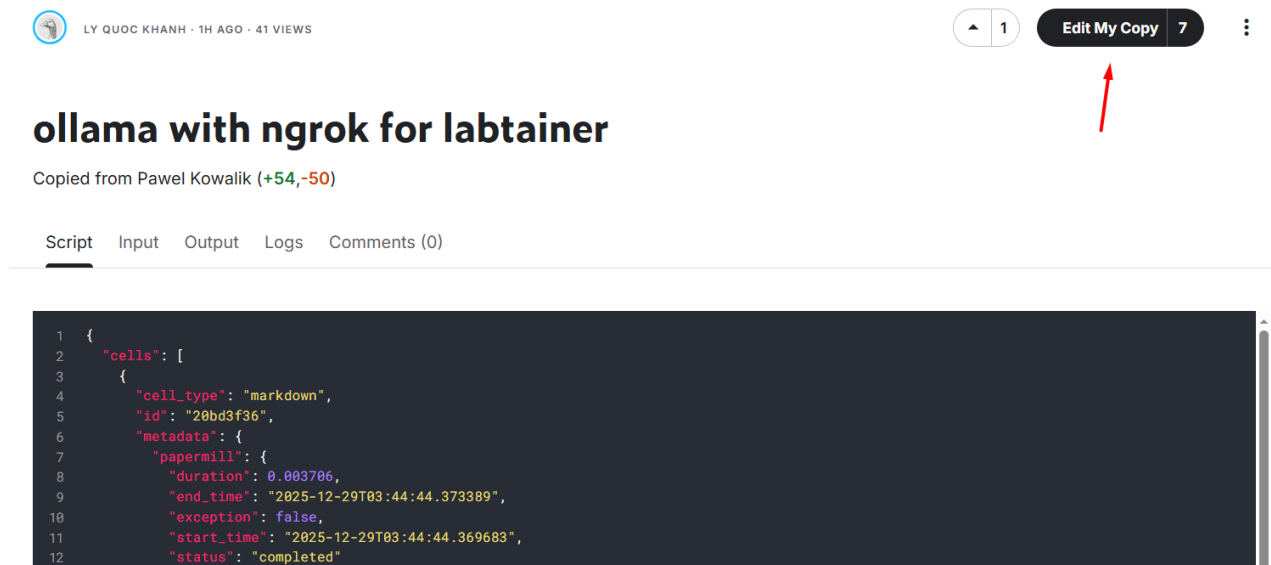
DANH MỤC CÁC HÌNH VẼ

Hình 1 : Truy cập vào đường link	1
Hình 2 : Tạo thành công file code Kaggle	1
Hình 3 : Mục môi trường trong bài hướng dẫn xác định mô hình ngôn ngữ sử dụng trên Kaggle	2
Hình 4 : Cấu hình tên mô hình trên Kaggle dựa theo mô hình trong bài hướng dẫn thực hành	2
Hình 5 : Mục môi trường trong bài hướng dẫn xác định mô hình mô hình ngôn ngữ sử dụng trên Kaggle	3
Hình 6 : Cấu hình tên mô hình trên Kaggle dựa theo mô hình trong bài hướng dẫn thực hành	3
Hình 7 : Thay thế mã ngrok để chạy được Ngrok trên Kaggle.....	4
Hình 8 : Lựa chọn GPU chạy trên Kaggle	4
Hình 9 : Chạy tất cả đoạn code trong file code Kaggle	4
Hình 10 : Thu được giá trị url ngrok sau khi chạy code hoàn tất	5
Hình 11 : Tìm file proxy_server.py.....	5
Hình 12 : Thay thế giá trị url ngrok vào biến OLLAMA_HOST	6
Hình 13 : Chạy file proxy_server.py	6
Hình 14 : Phản hồi 200 có nghĩa là bạn đã kết nối thành công.....	6
Hình 15 : Vị trí token ngrok sau khi đăng nhập thành công tài khoản ngrok.....	7

1.1 Truy cập link Kaggle

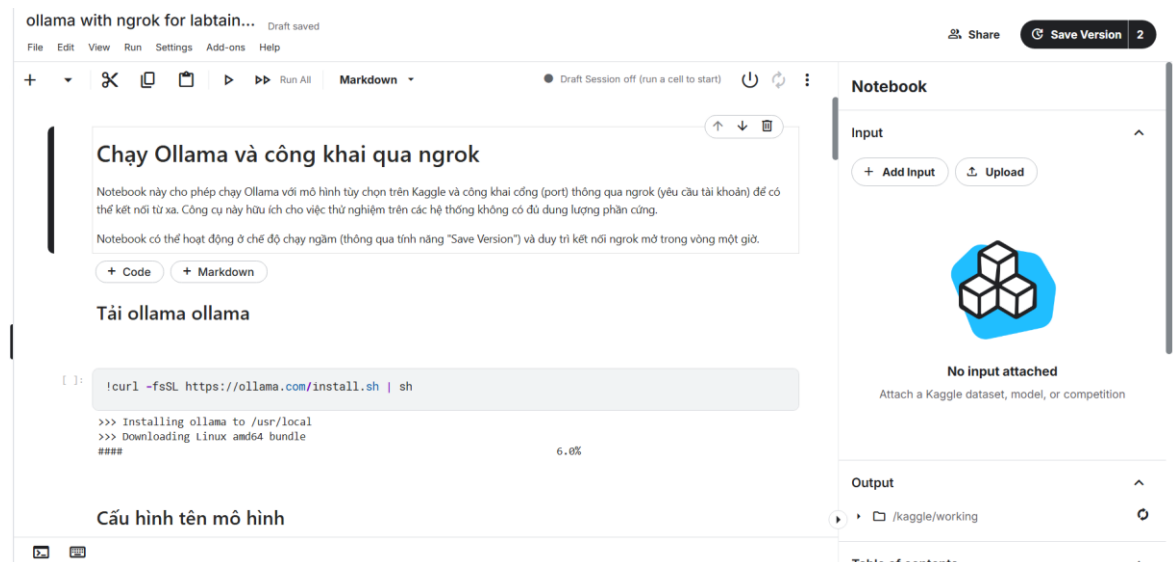
Ta truy cập vào đường link : <https://www.kaggle.com/code/lyquockhanh/ollama-with-ngrok-for-labtainr>

Truy cập vào đường dẫn thấy được project , click nút Edit My Copy để tạo ra bản sao



Hình 1 : Truy cập vào đường link

Tạo thành công bản copy



Hình 2 : Tạo thành công file code Kaggle

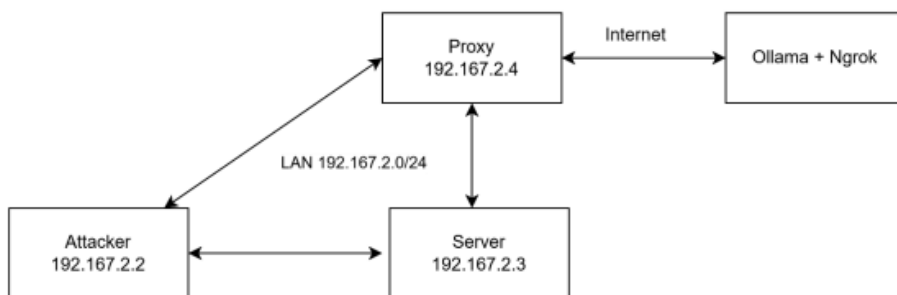
1.2 Cài đặt mô hình chạy trên ollama của Kaggle

Trong bài hướng dẫn ta đọc mục môi trường thấy tên mô hình ngôn ngữ lớn sử dụng trong bài lab

1.2.1 Trường hợp có 1 mô hình

1.1.3 Môi trường

- Mô hình ngôn ngữ lớn sử dụng : **llama3.1:8b** ←
- Sơ đồ mạng



Hình 1 : Sơ đồ mạng

Hình 3 : Mục môi trường trong bài hướng dẫn xác định mô hình ngôn ngữ sử dụng trên Kaggle

Đưa tên mô hình trong bài hướng dẫn thực hành vào mục Cấu hình tên mô hình của Kaggle

Cấu hình tên mô hình

Kiểm tra mục môi trường trong hướng dẫn của bài thực hành để xác định tên mô hình và thay vào biến .

Nếu bài lab chỉ cần 1 mô hình thì bạn hãy để trống 1 biến

```
] : OLLAMA_MODEL='llama3.1:8b'
    OLLAMA_MODEL1=''
    import os
    os.environ['OLLAMA_MODEL'] = OLLAMA_MODEL
    os.environ['OLLAMA_MODEL1'] = OLLAMA_MODEL1
    !echo $OLLAMA_MODEL
    !echo $OLLAMA_MODEL1
```

Hình 4 : Cấu hình tên mô hình trên Kaggle dựa theo mô hình trong bài hướng dẫn thực hành

1.2.2 Trường hợp có 2 mô hình

1.1.3 Môi trường

- Mô hình ngôn ngữ lớn kiểm thử : **llama3.1:8b**
- Mô hình ngôn ngữ lớn đánh giá : **gemma3:12b**

Sơ đồ mạng



Hình 2 : Sơ đồ mạng

Hình 5 : Mục môi trường trong bài hướng dẫn xác định mô hình mô hình ngôn ngữ sử dụng trên Kaggle

Đưa tên mô hình trong bài hướng dẫn thực hành vào mục **Cấu hình tên mô hình** của Kaggle

Cấu hình tên mô hình

Kiểm tra mục môi trường trong hướng dẫn của bài thực hành để xác định tên mô hình và thay vào biến .

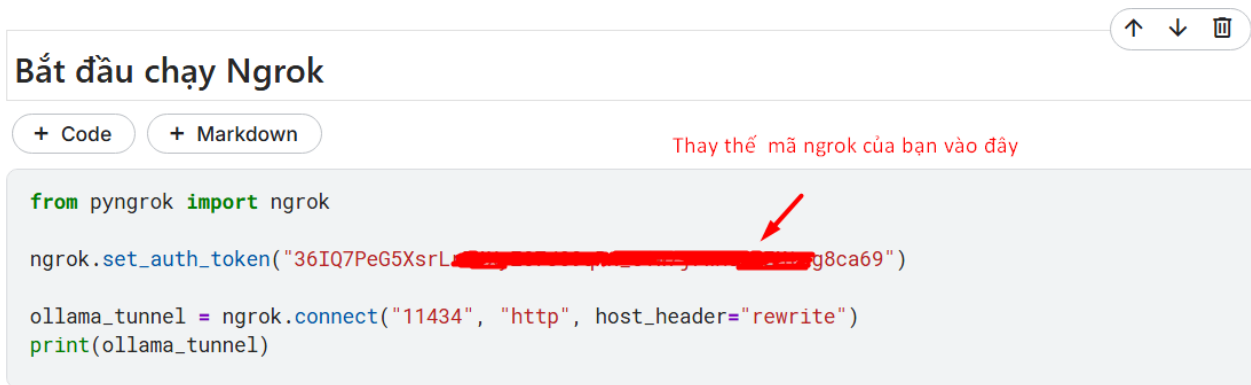
Nếu bài lab chỉ cần 1 mô hình thì bạn hãy để trống 1 biến

```
> OLLAMA_MODEL='llama3.1:8b'
OLLAMA_MODEL1='gemma3:12b'
import os
os.environ['OLLAMA_MODEL'] = OLLAMA_MODEL
os.environ['OLLAMA_MODEL1'] = OLLAMA_MODEL1
!echo $OLLAMA_MODEL
!echo $OLLAMA_MODEL1
```

Hình 6 : Cấu hình tên mô hình trên Kaggle dựa theo mô hình trong bài hướng dẫn thực hành

1.3 Cài đặt để chạy được ngrok

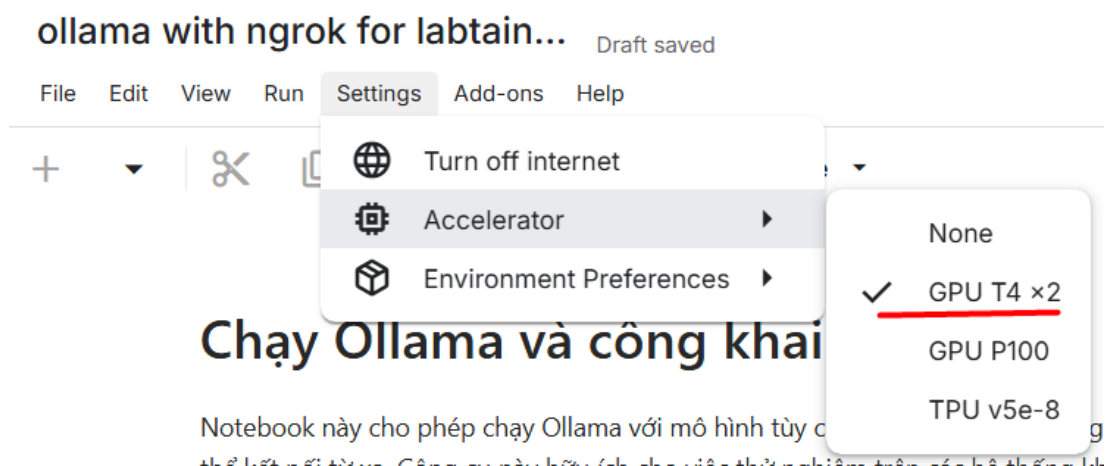
Để có thể khởi chạy và sử dụng ngrok, sinh viên cần thay thế mã xác thực (ngrok auth token) bằng token của chính tài khoản ngrok cá nhân.



Hình 7 : Thay thế mã ngrok để chạy được Ngrok trên Kaggle

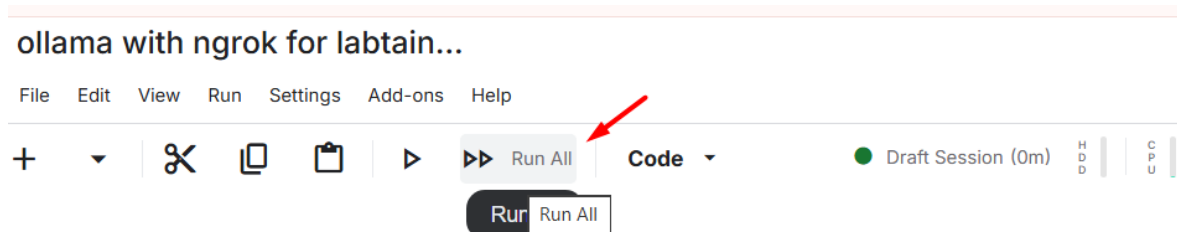
1.4 Khởi chạy file code trên Kaggle

Chọn mục Settings -> Accelerator -> GPU T4x2



Hình 8 : Lựa chọn GPU chạy trên Kaggle

Sinh viên click nút Run All để chạy tất cả các code theo thứ tự .



Chạy Ollama và công khai qua ngrok

Notebook này cho phép chạy Ollama với mô hình tùy chọn trên Kaggle và công khai cổng (port) thông qua thể kết nối từ xa. Công cụ này hữu ích cho việc thử nghiệm trên các hệ thống không có đủ dung lượng phần

Notebook có thể hoạt động ở chế độ chạy ngầm (thông qua tính năng "Save Version") và duy trì kết nối ng



Hình 9 : Chạy tất cả đoạn code trong file code Kaggle

Quá trình chạy hoàn tất sinh viên sẽ thu được một ngrok url

Bắt đầu chạy Ngrok

```
[7]: from pyngrok import ngrok

ngrok.set_auth_token("36IQ7PeG5XsrLrB4XyZ07J09qRx_51W7j7WMGZtZMtxg8ca69")

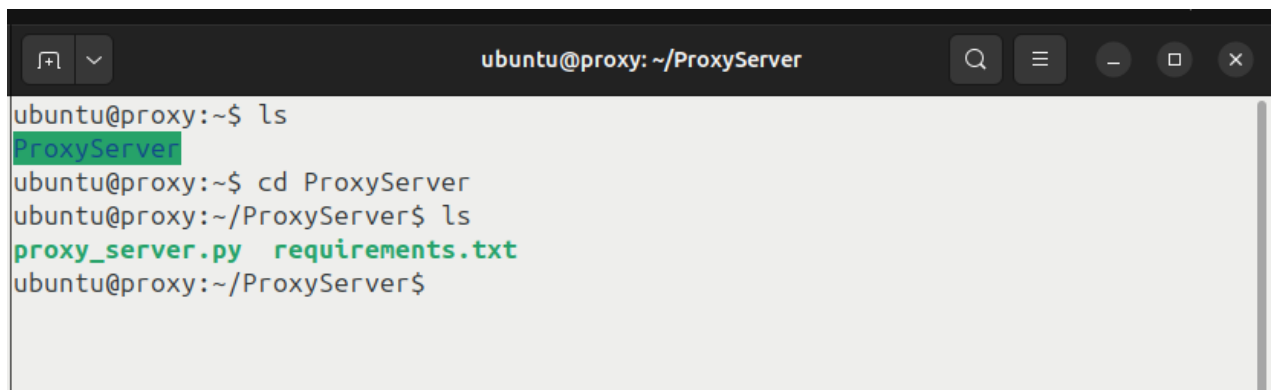
ollama_tunnel = ngrok.connect("11434", "http", host_header="rewrite")
print(ollama_tunnel)

NgrokTunnel: "https://postosseous-superjudicially-shavon.ngrok-free.dev" -> "http://localhost:11434"
```

Hình 10 : Thu được giá trị url ngrok sau khi chạy code hoàn tất

1.5 Thay thế giá trị ngrok url vào labtainer

Sinh viên mở bài labtainer , trong terminal ProxyServer di chuyển tới thư mục ProxyServer



```
ubuntu@proxy: ~/ProxyServer
ubuntu@proxy:~$ ls
ProxyServer
ubuntu@proxy:~$ cd ProxyServer
ubuntu@proxy:~/ProxyServer$ ls
proxy_server.py  requirements.txt
ubuntu@proxy:~/ProxyServer$
```

Hình 11 : Tìm file proxy_server.py

Chỉnh sửa file proxy_server.py thay thế giá trị **url ngrok** vào biến **OLLAMA_HOST**

```
ubuntu@proxy: ~/ProxyServer
GNU nano 4.8 proxy_server.py
from flask import Flask, request, jsonify
import requests
import os

app = Flask(__name__)

OLLAMA_HOST = "https://postosseous-superjudicially-shavon.ngrok-free.dev"

@app.route("/api/chat", methods=["POST"])
def proxy_chat():
    """
    Forward request to Flask server to Ollama through Ngrok.
    """
    try:
        data = request.get_json()
        resp = requests.post(f"{OLLAMA_HOST}/api/chat", json=data, timeout=40)
        resp.raise_for_status()

        return jsonify(resp.json())
    except requests.exceptions.RequestException as e:
        print(f"Error when forwarding request: {e}")
        return jsonify({"error": f"Error when forwarding request: {e}"}), 503
    except Exception as e:
        return jsonify({"error": f"Error in Proxy: {e}"}), 500

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=29310)
```

Hình 12 : Thay thế giá trị url ngrok vào biến OLLAMA_HOST

Khởi chạy file proxy_server.py thông qua câu lệnh :

python3 proxy_server.py

```
ubuntu@proxy: ~/ProxyServer
ubuntu@proxy:~$ ls
ProxyServer
ubuntu@proxy:~$ cd ProxyServer
ubuntu@proxy:~/ProxyServer$ ls
proxy_server.py  requirements.txt
ubuntu@proxy:~/ProxyServer$ nano proxy_server.py
ubuntu@proxy:~/ProxyServer$ nano proxy_server.py
ubuntu@proxy:~/ProxyServer$ python3 proxy_server.py
* Serving Flask app 'proxy_server'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:29310
* Running on http://192.167.2.4:29310
Press CTRL+C to quit
```

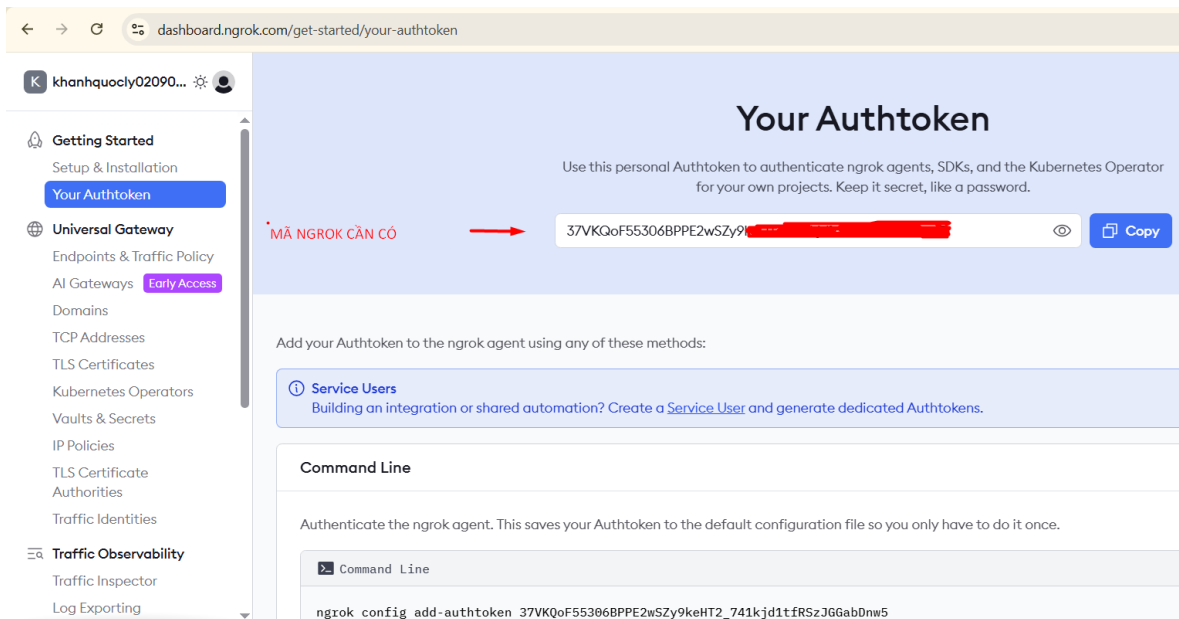
Hình 13 : Chạy file proxy_server.py

```
ubuntu@proxy: ~/ProxyServer
ubuntu@proxy:~/ProxyServer$ ls
proxy_server.py  requirements.txt
ubuntu@proxy:~/ProxyServer$ python3 proxy_server.py
* Serving Flask app 'proxy_server'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:29310
* Running on http://192.167.2.4:29310
Press CTRL+C to quit
192.167.2.3 - - [29/Dec/2025 04:59:22] "POST /api/chat HTTP/1.1" 200 -
192.167.2.3 - - [29/Dec/2025 04:59:27] "POST /api/chat HTTP/1.1" 200 -
192.167.2.3 - - [29/Dec/2025 04:59:32] "POST /api/chat HTTP/1.1" 200 -
192.167.2.3 - - [29/Dec/2025 04:59:52] "POST /api/chat HTTP/1.1" 200 -
```

Hình 14 : Phản hồi 200 có nghĩa là bạn đã kết nối thành công

Chú thích

Khi đăng ký đăng nhập thành công tài khoản ngrok thì đây chính là mã ngrok cần lấy



Hình 15 : Vị trí token ngrok sau khi đăng nhập thành công tài khoản ngrok