Báo cáo LAB 1

Thành viên:

22520825 – Nguyễn Đức Luân

22521110 – Đào Hoàng Phúc

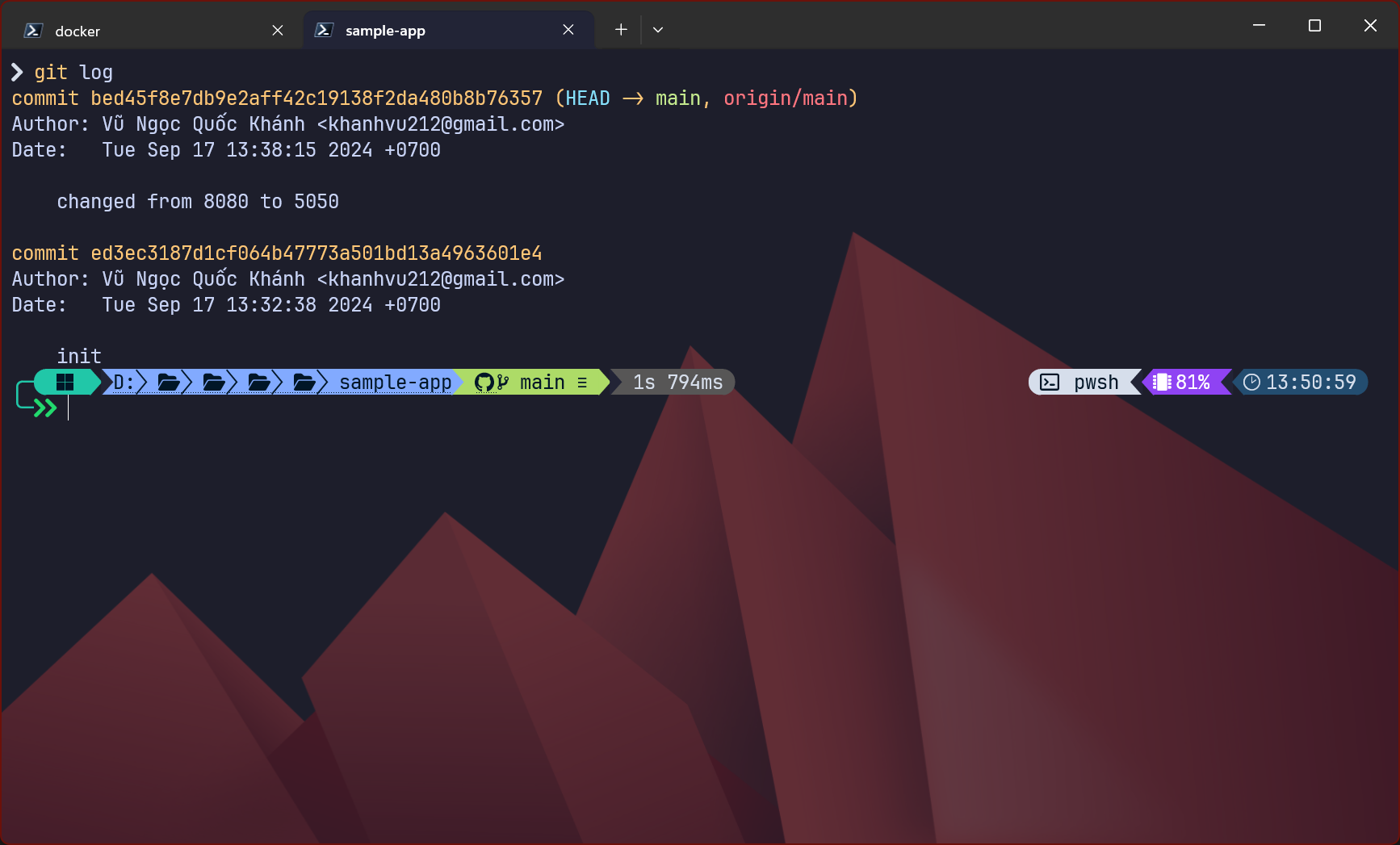
22520661 – Vũ Ngọc Quốc Khánh

# Câu 1: Cho biết thông tin về lần commit vừa thực hiện:

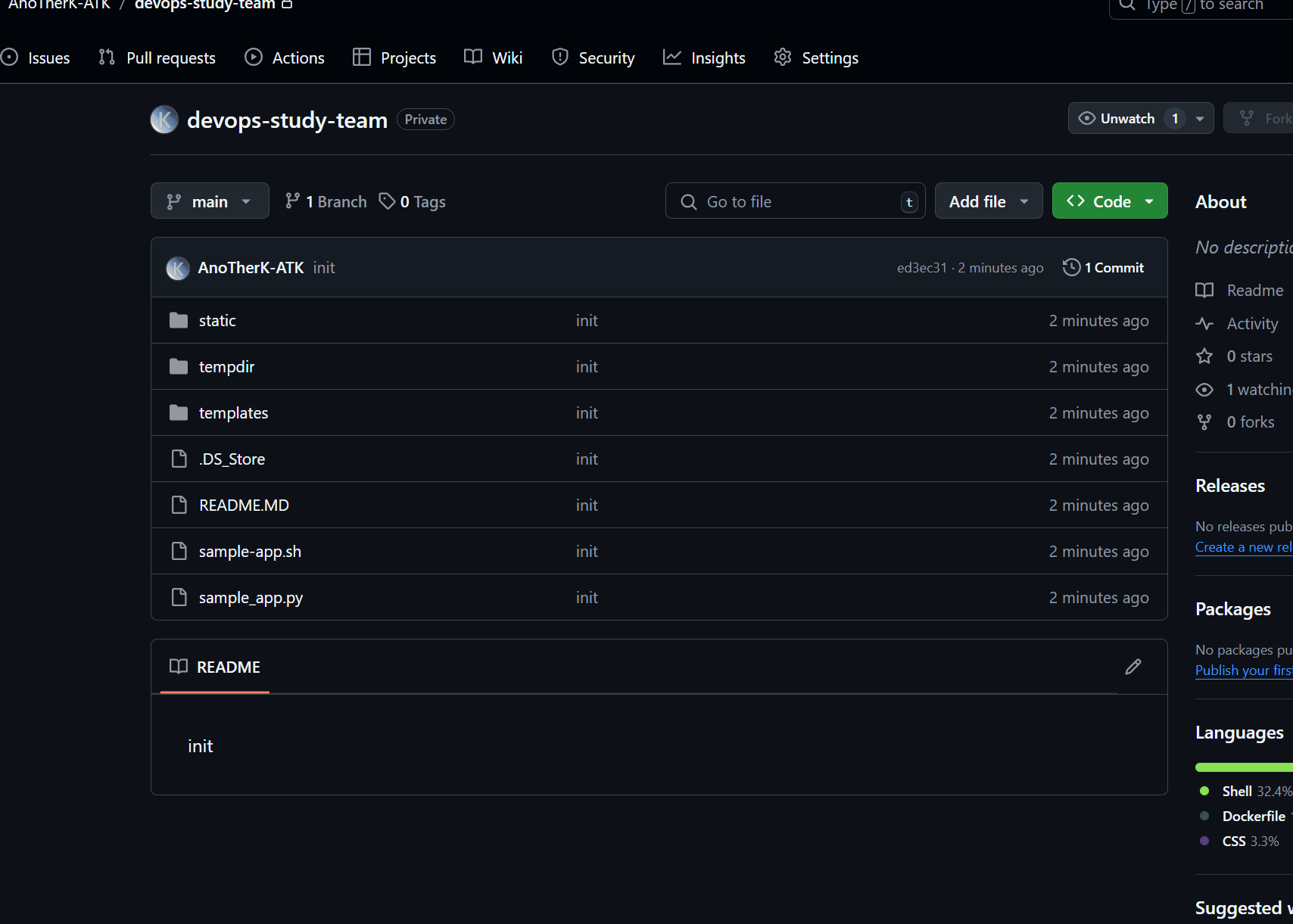
Commit ID: bed45f8e7db9e2aff42c19138f2da480b8b76357

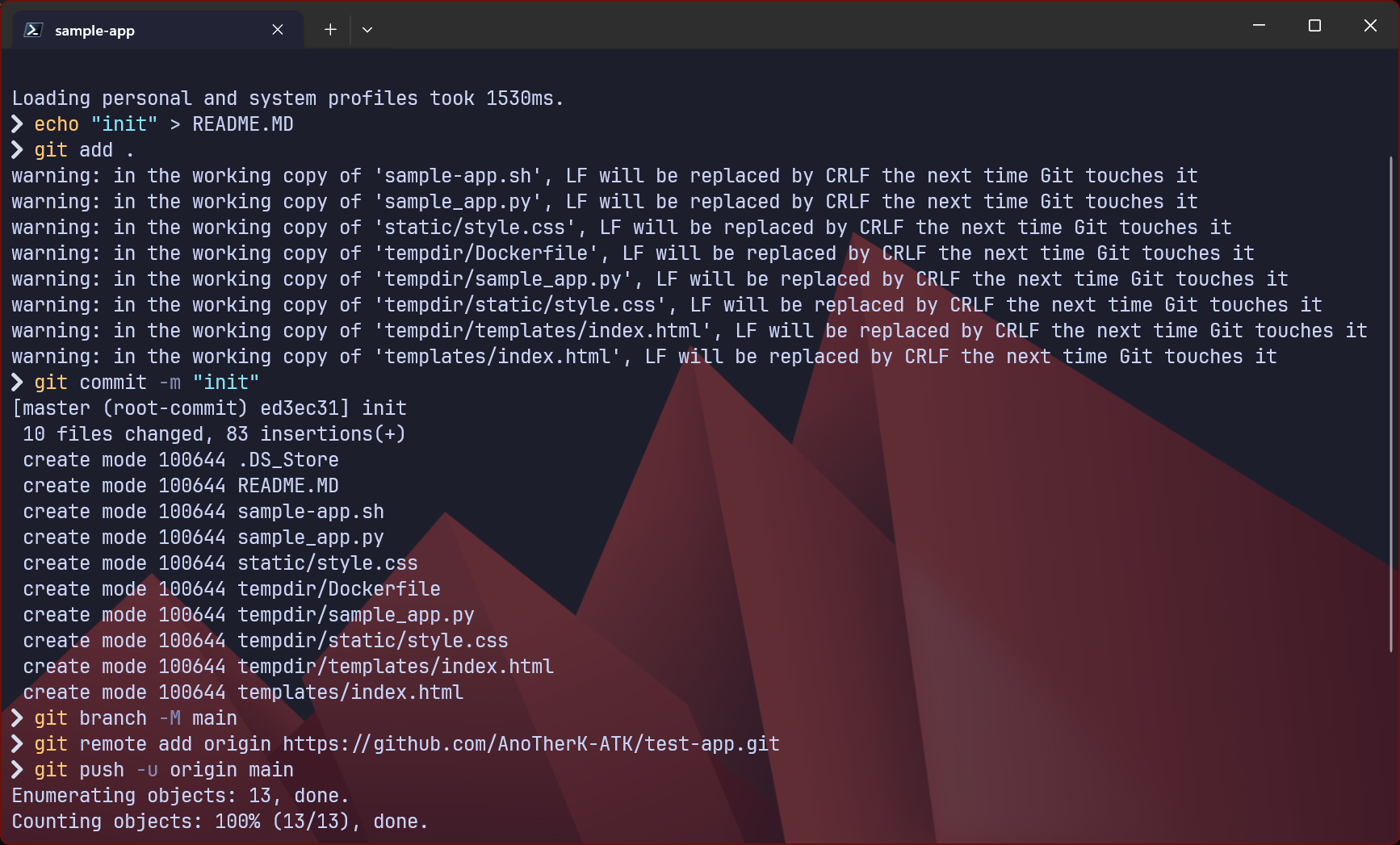
Ngày giờ commit: 17/09/2024 lúc 13:38:15

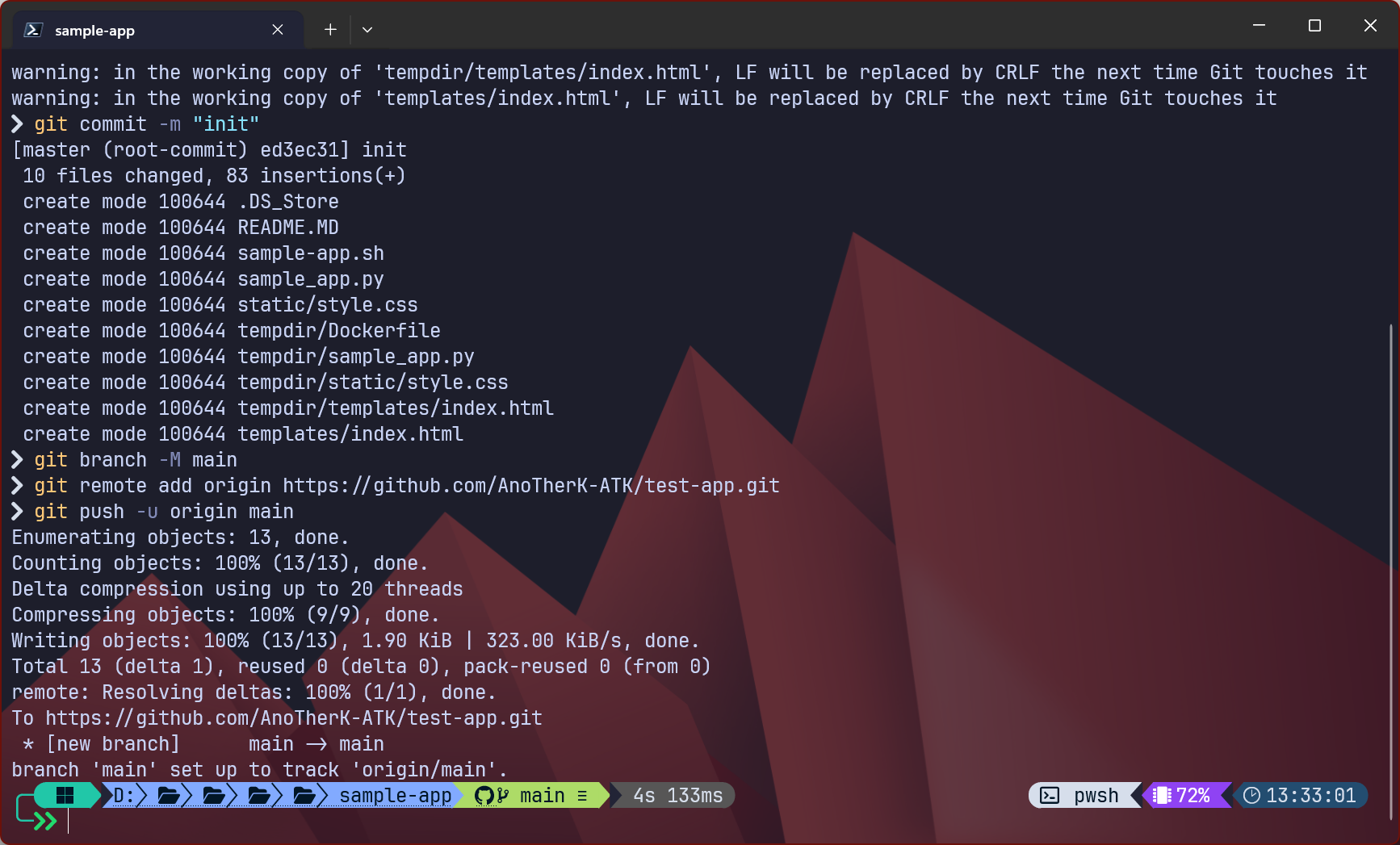
Thông điệp commit: changed from 8080 to 5050



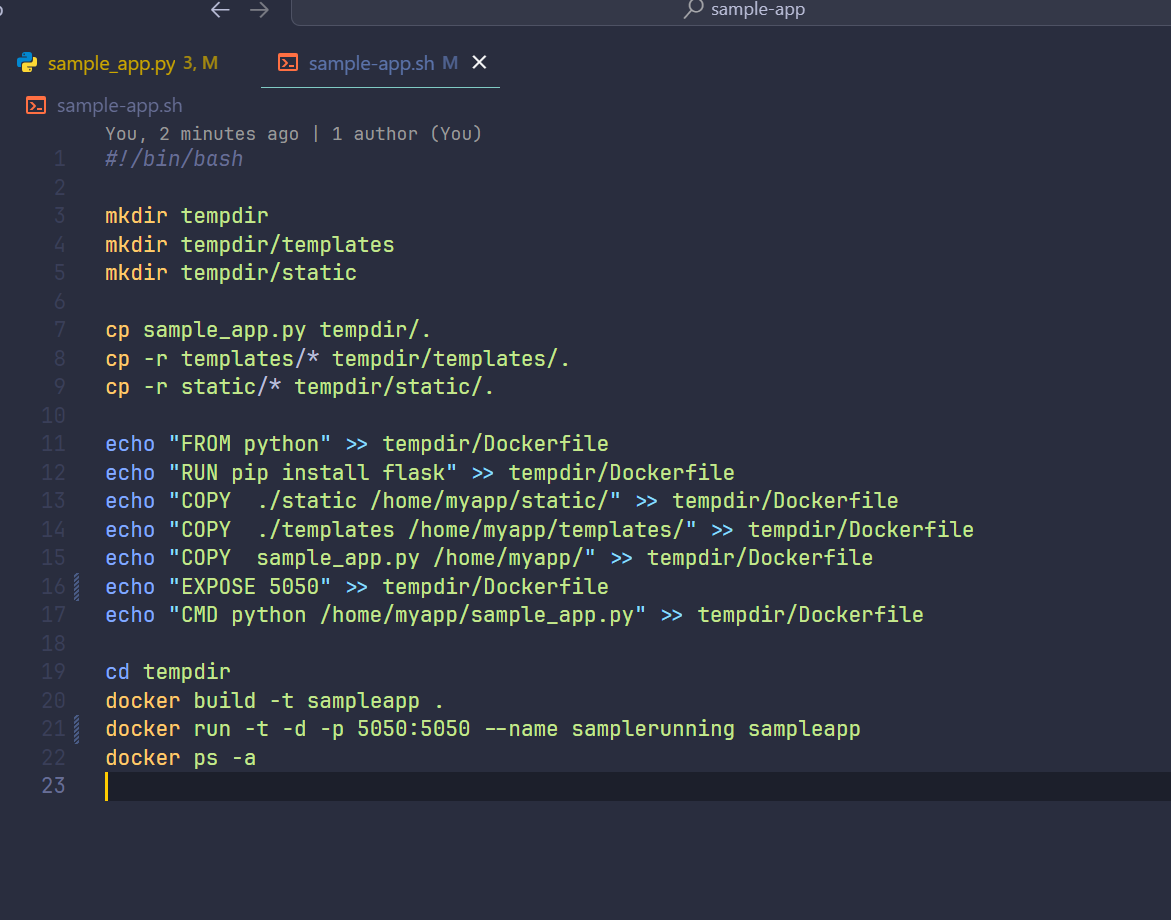
# Câu 2: Git và Github

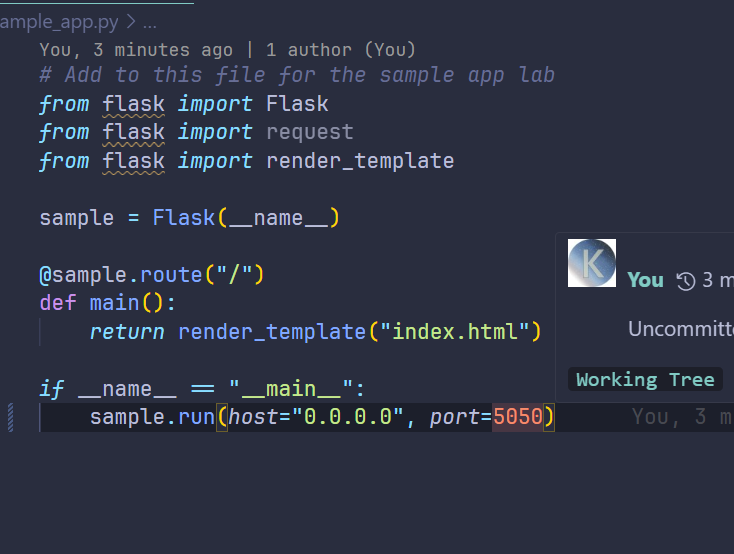


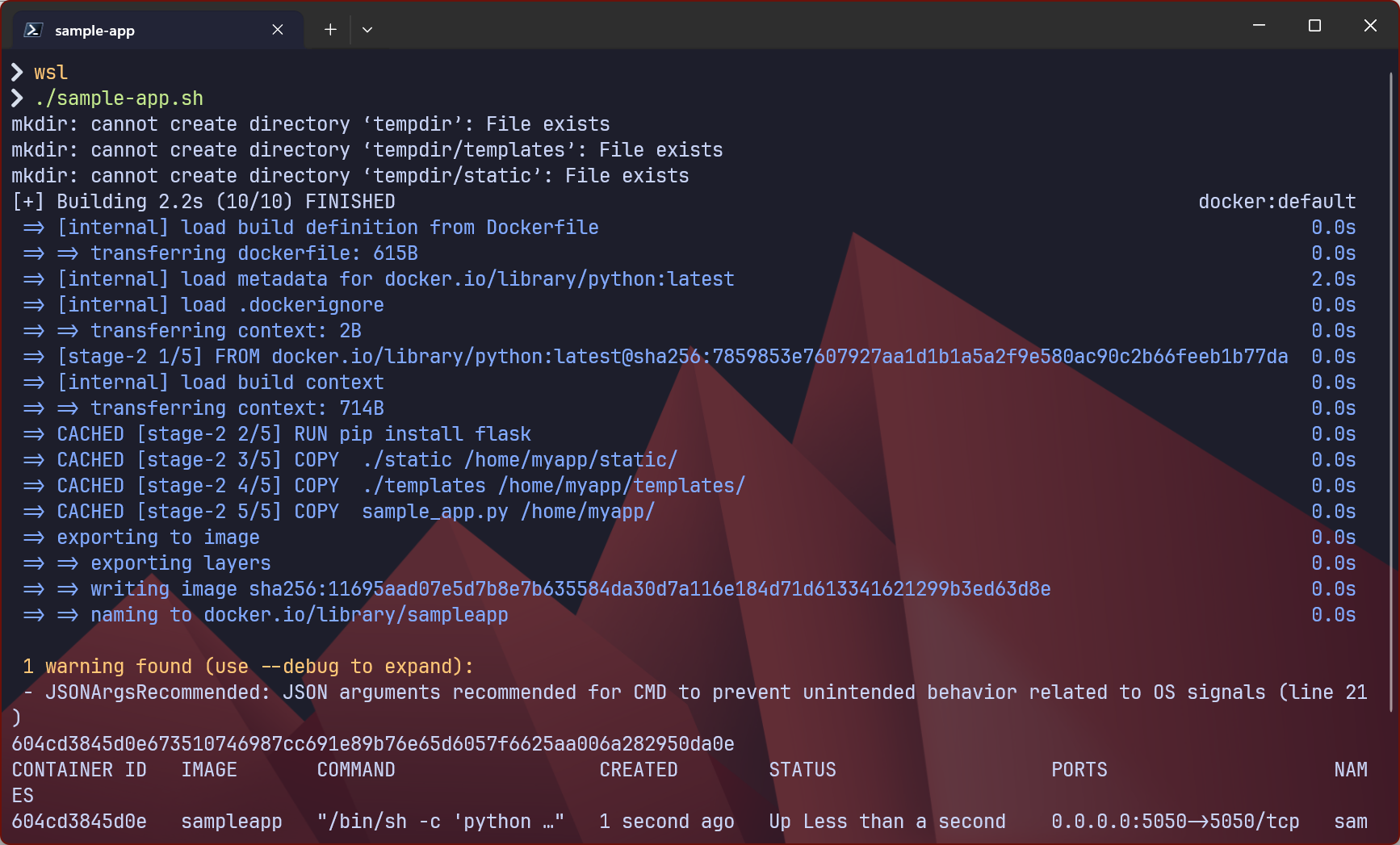


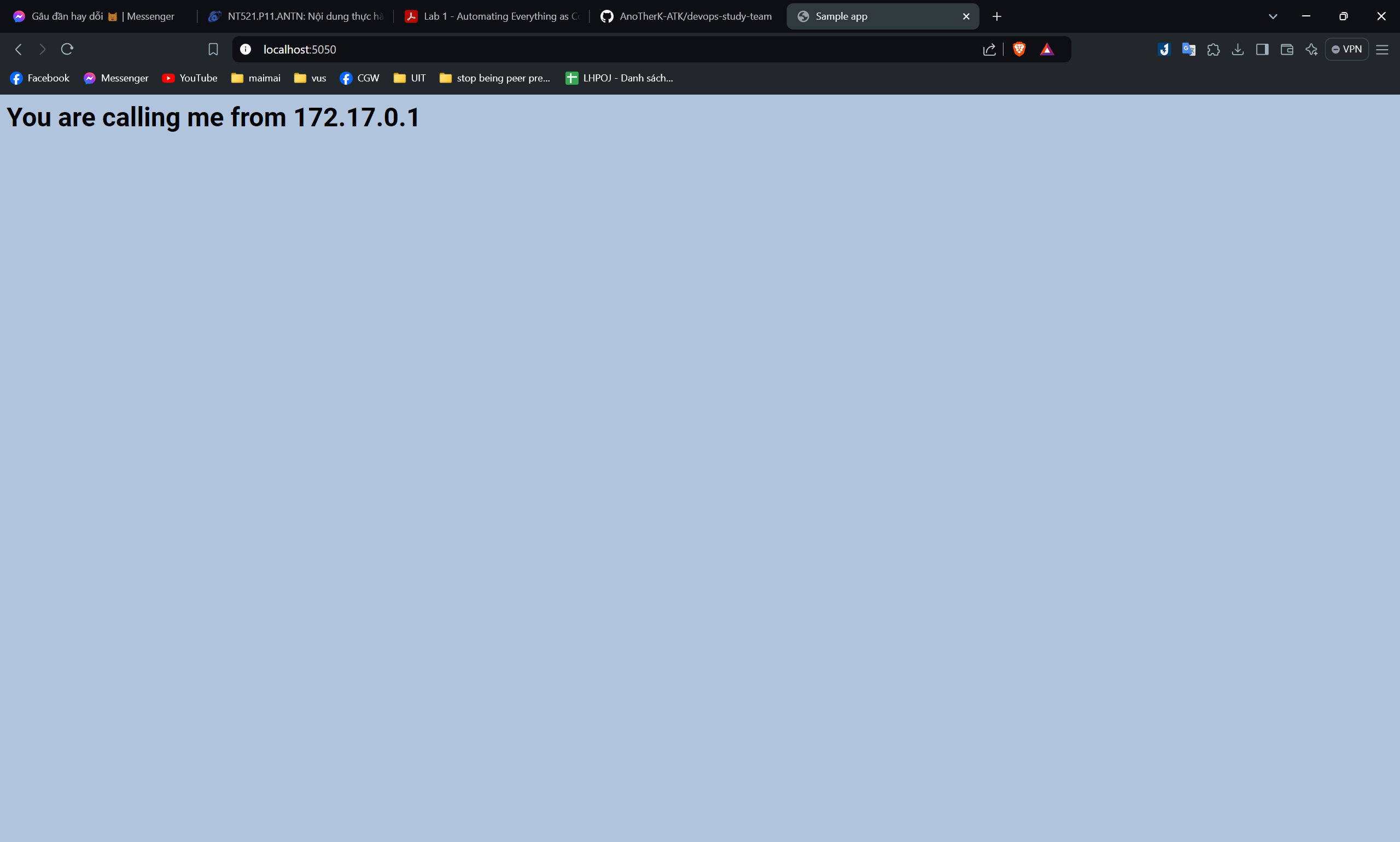


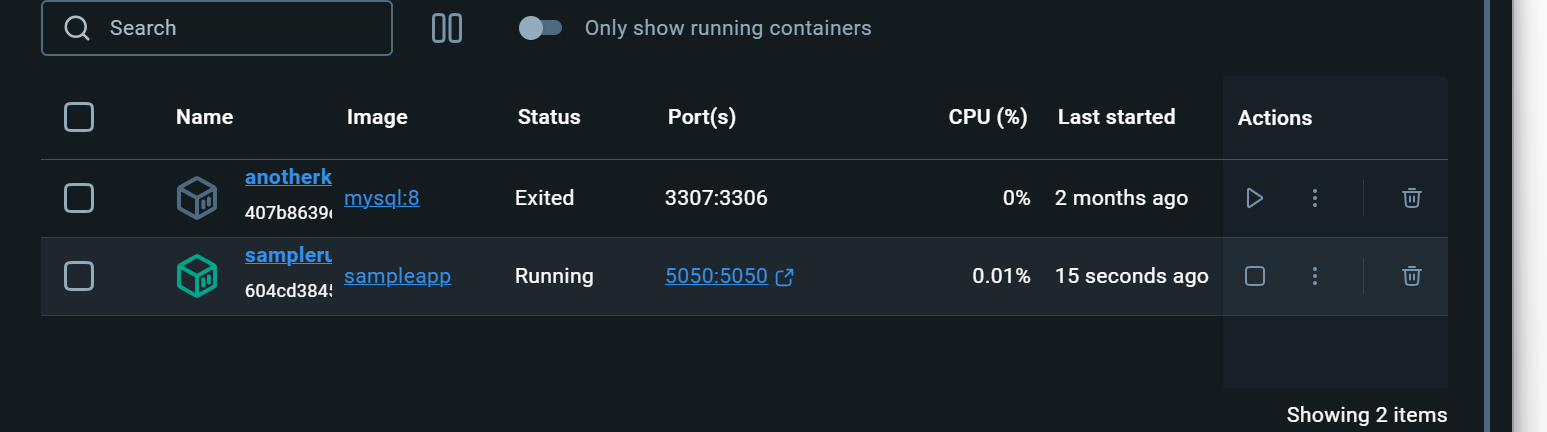
# Câu 3: Thực hiện commit và chuyển port

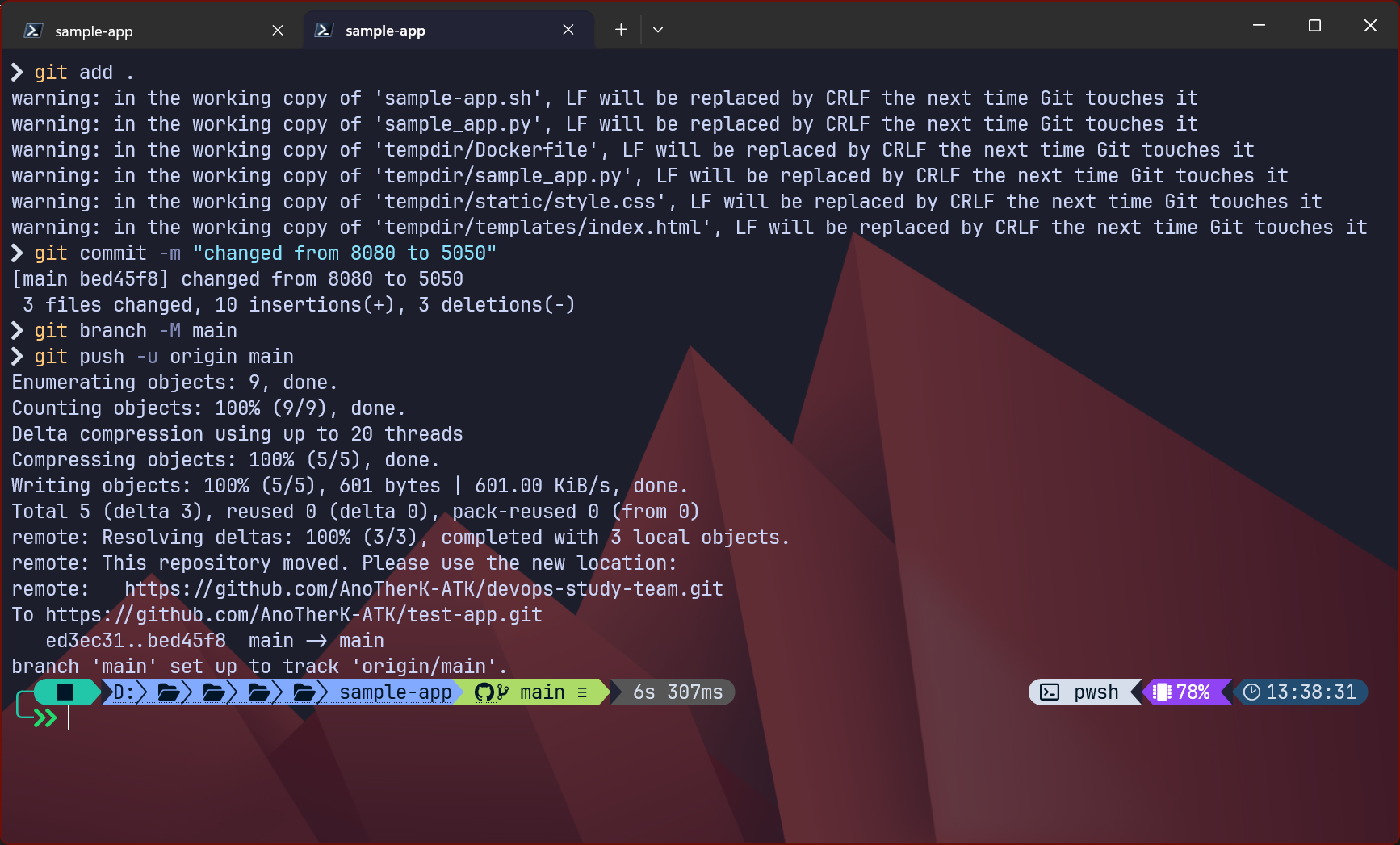


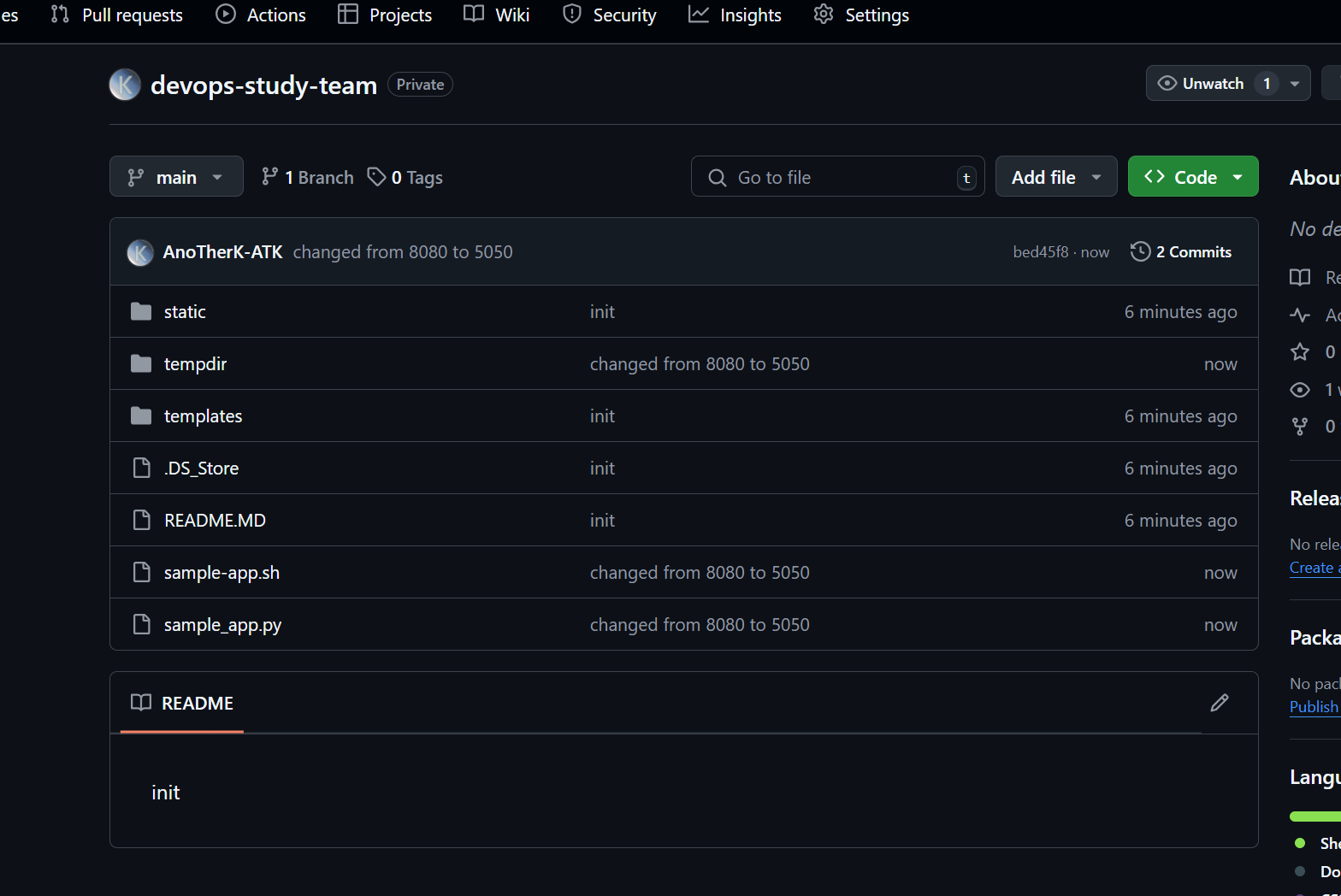






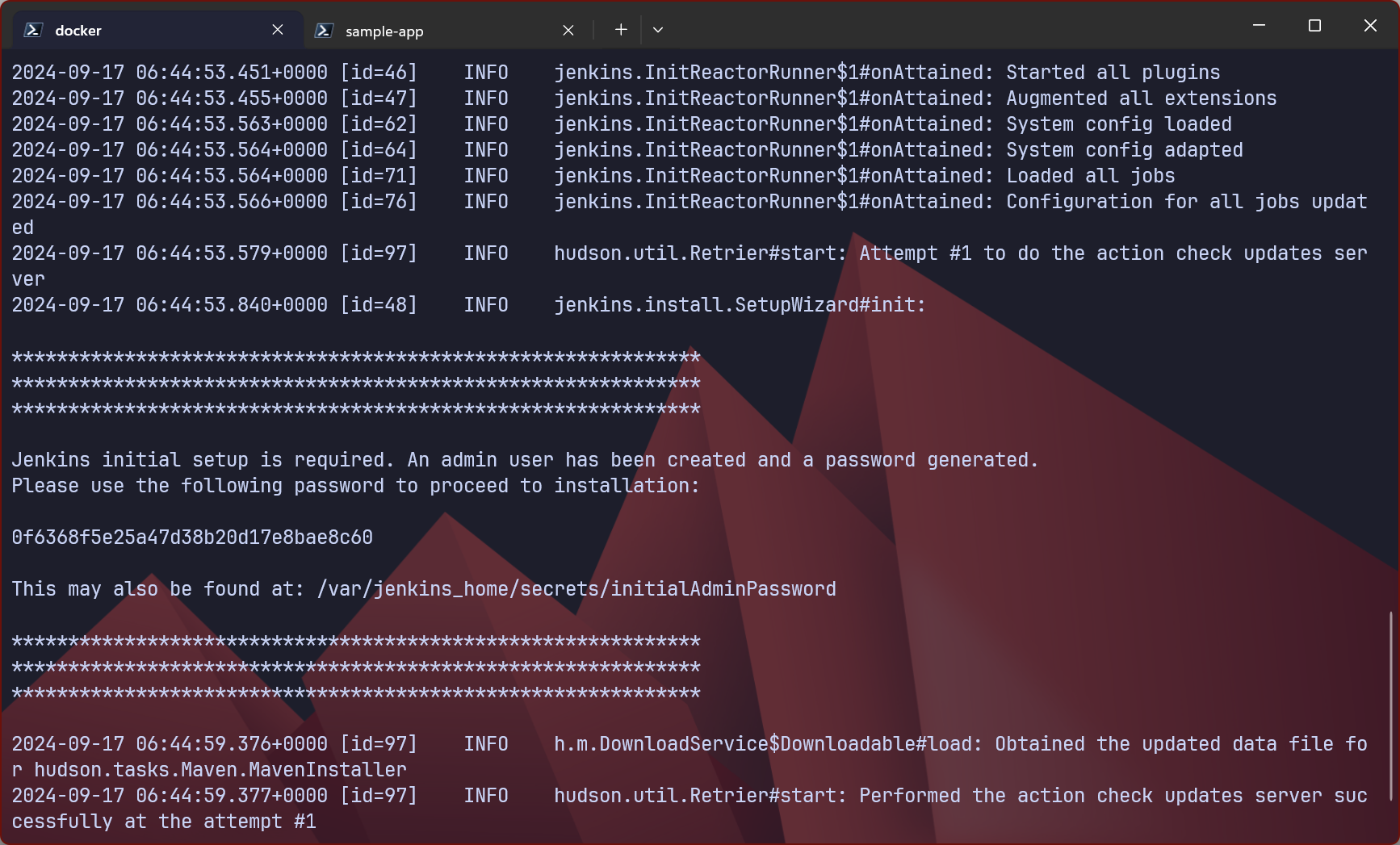


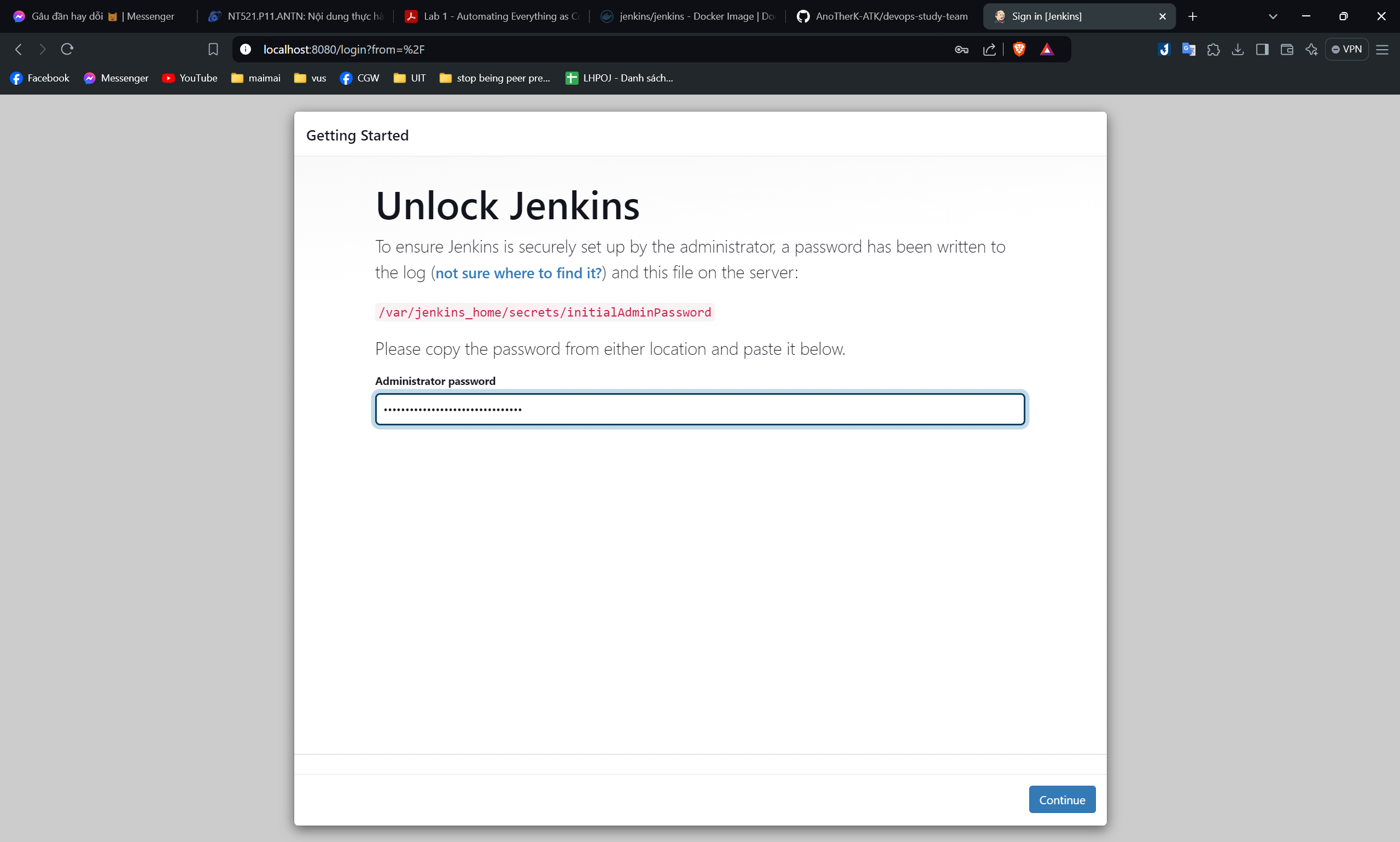


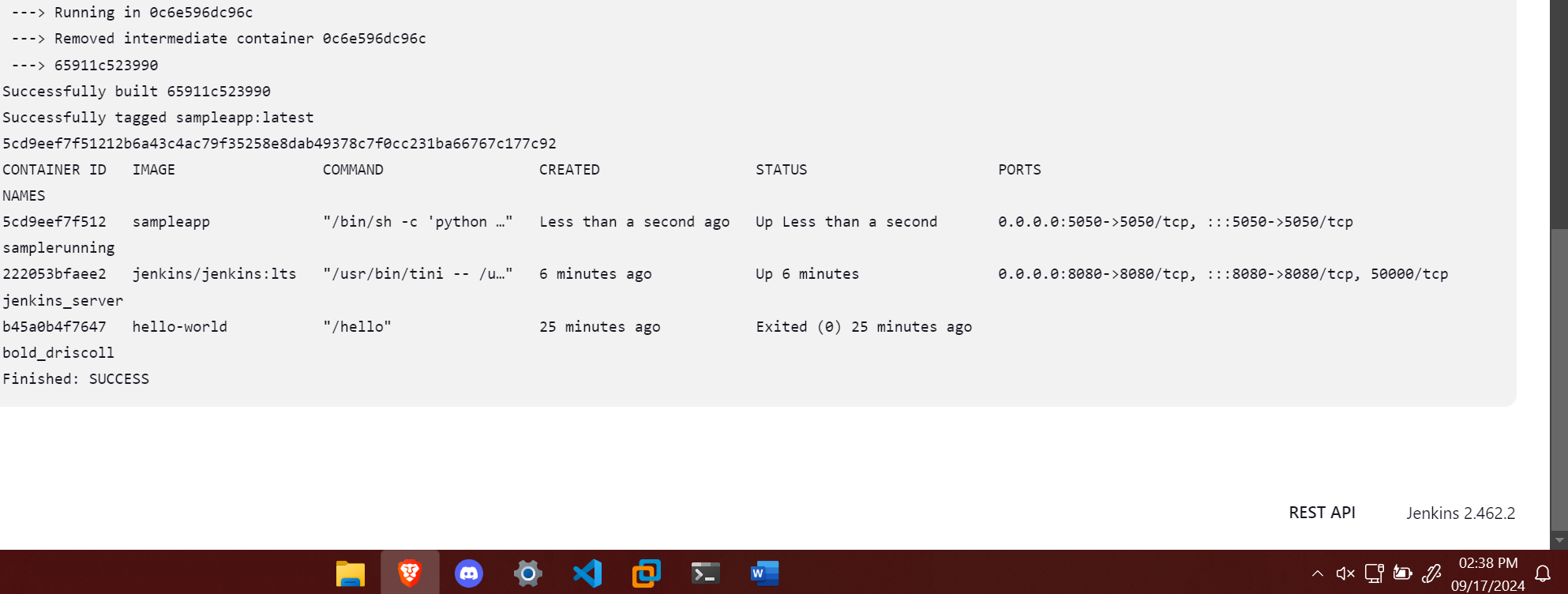


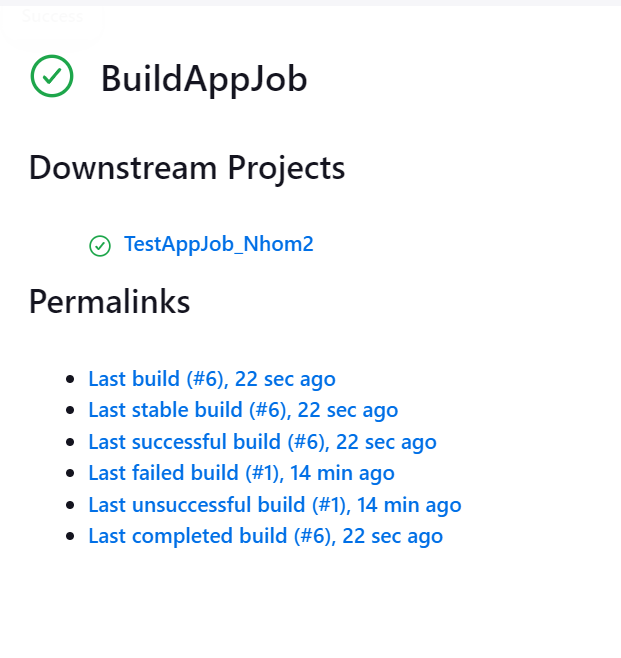
# Câu 4: Sinh viên tự tìm hiểu và giải thích tuỳ chọn -v ở lệnh chạy Docker

-v: mount volume ra một folder

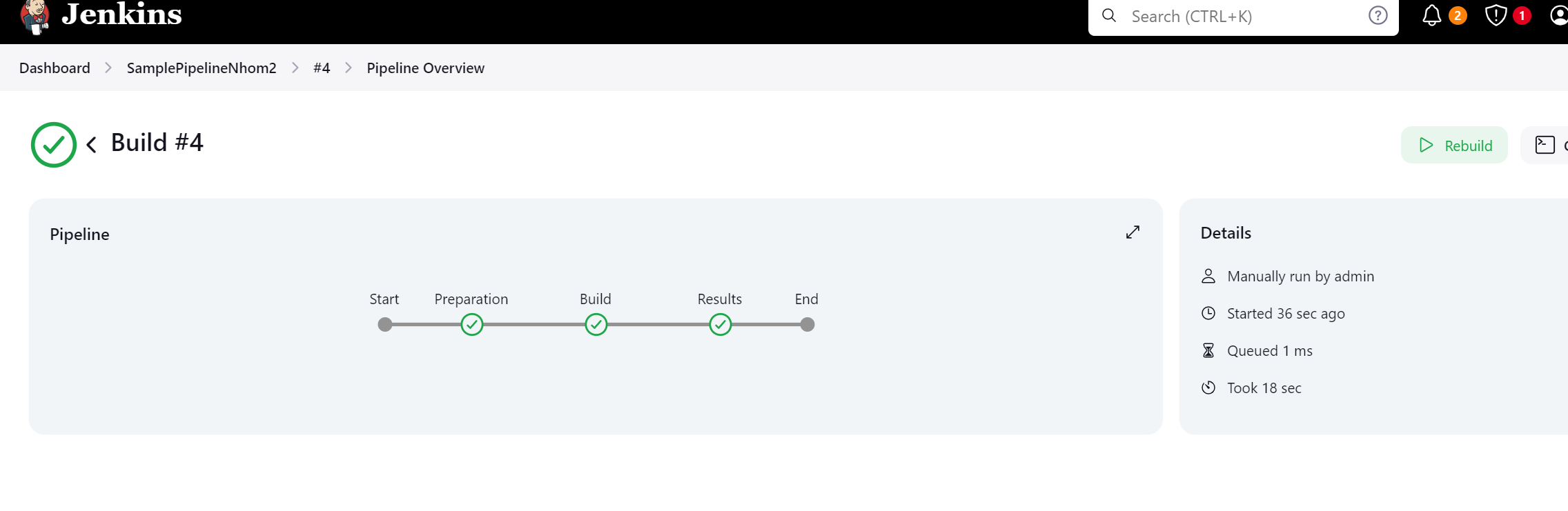




Câu 5: Sinh viên hoàn thành job kiểm tra bản dựng ứng dụng tự động

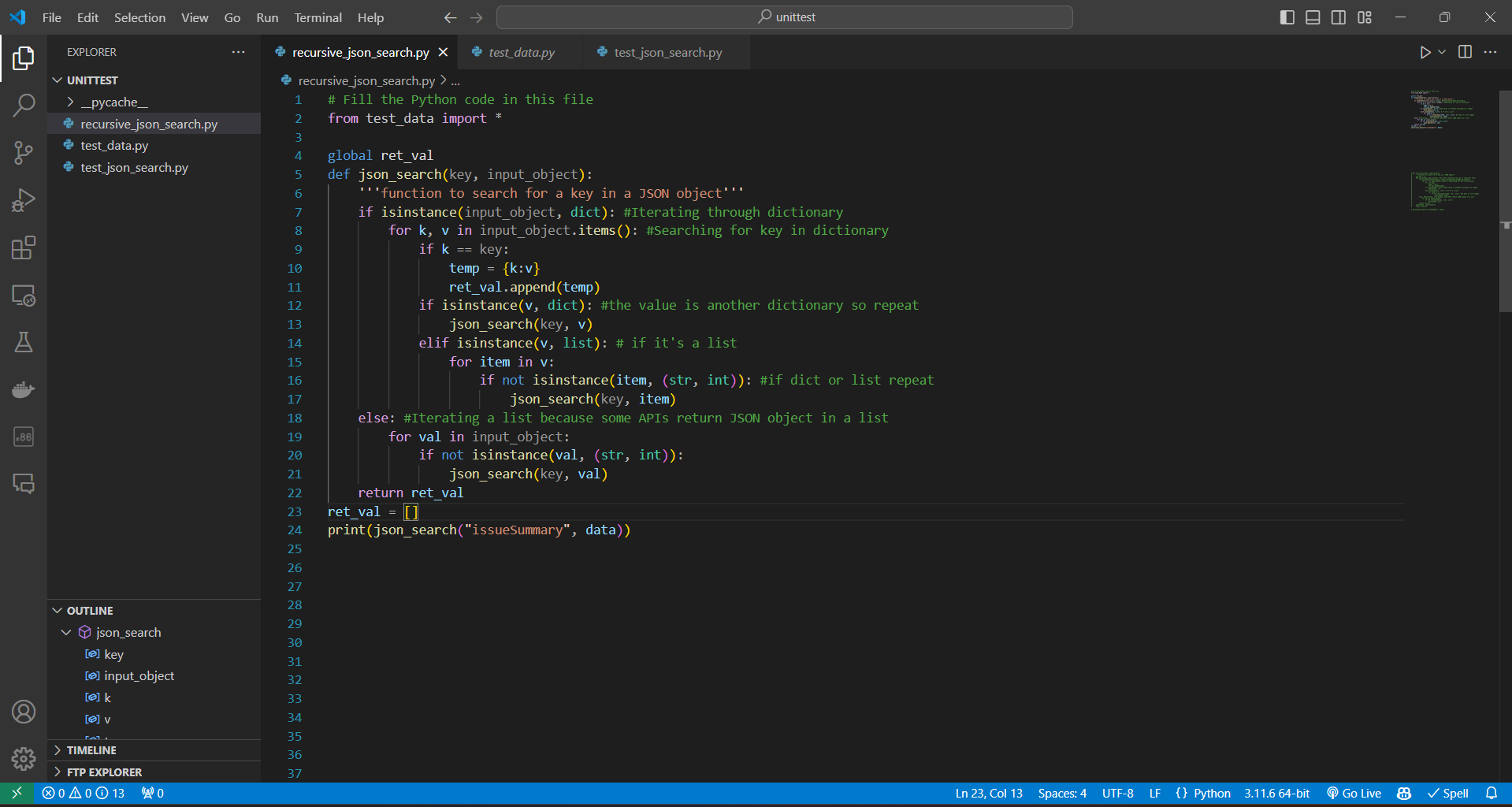
 Câu 6: Sinh viên hoàn thành pipeline của 2 ứng dụng Buid và Test



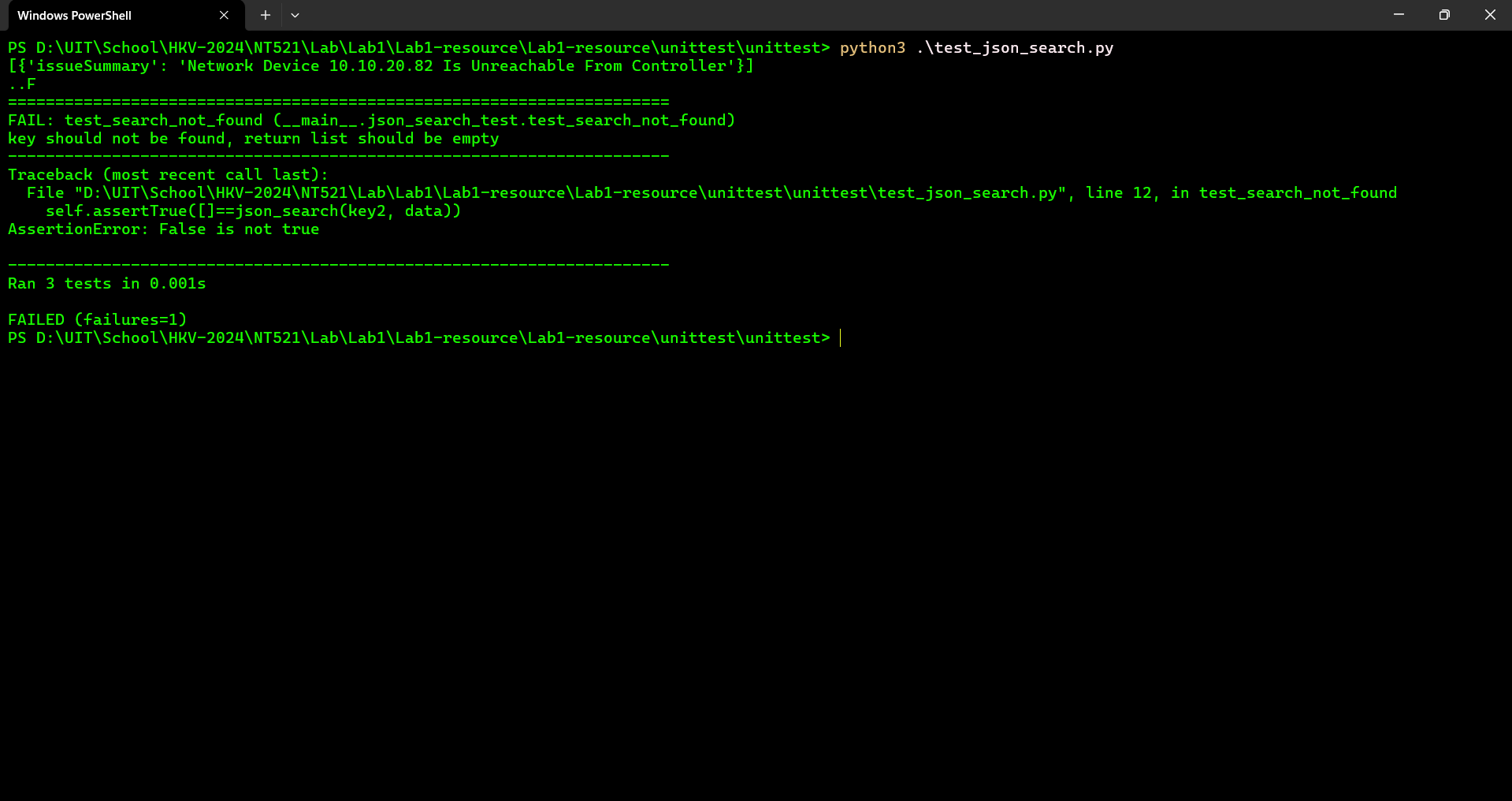


# Câu 7: Sửa code bằng cách thay đổi vị trí của ret\_val

Hướng xử lý: Em khởi tạo ret\_val ở ngoài và dùng global ret\_val để nó truy cập vào value trong function, qua đó nó sẽ không bị reset dữ liệu mỗi khi đệ quy được gọi



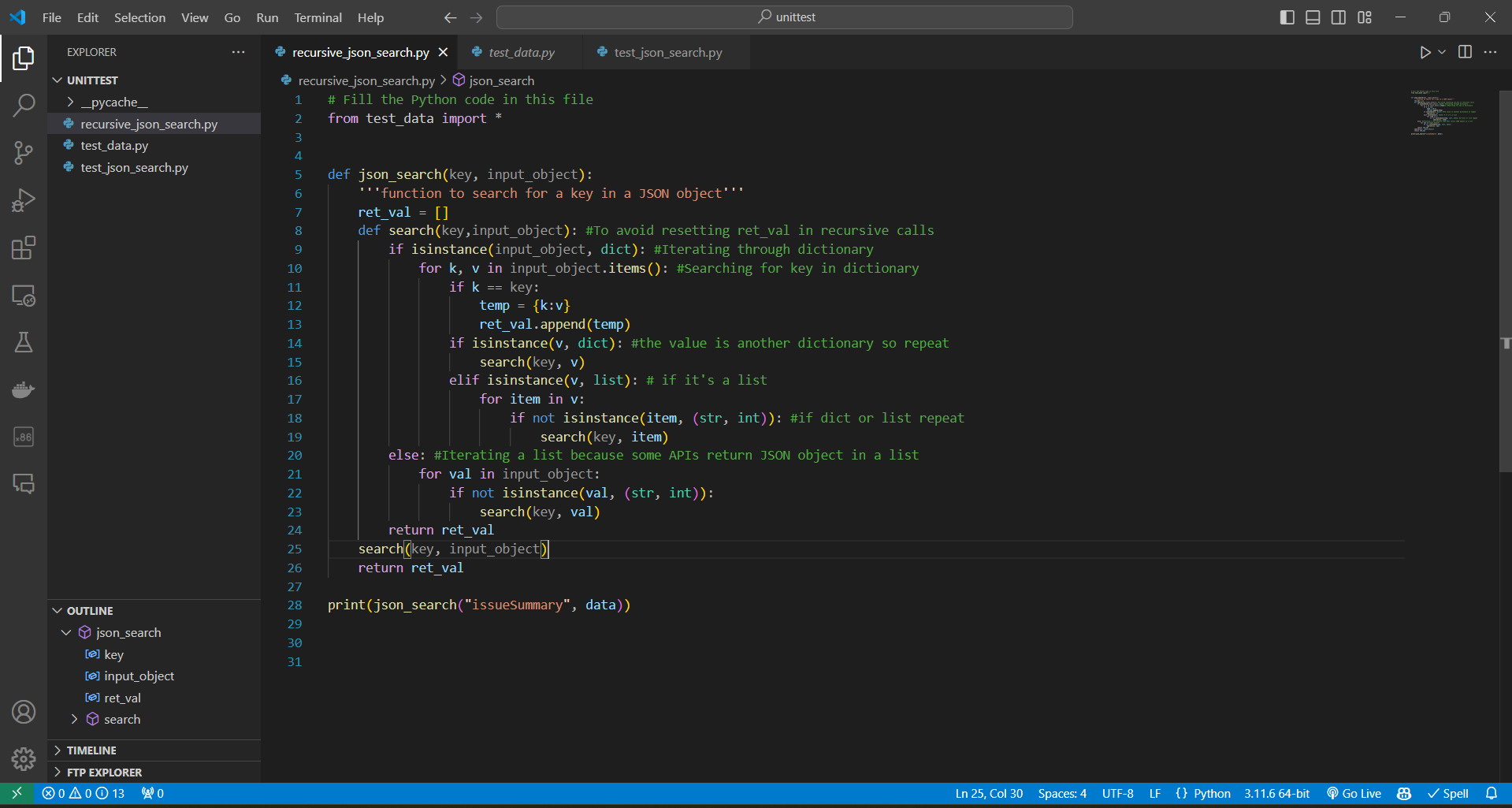
**Kết quả: Pass hết các test trừ test của function “test\_search\_not\_found” khi mà key2 không tồn tại nhưng lại trả về có.**



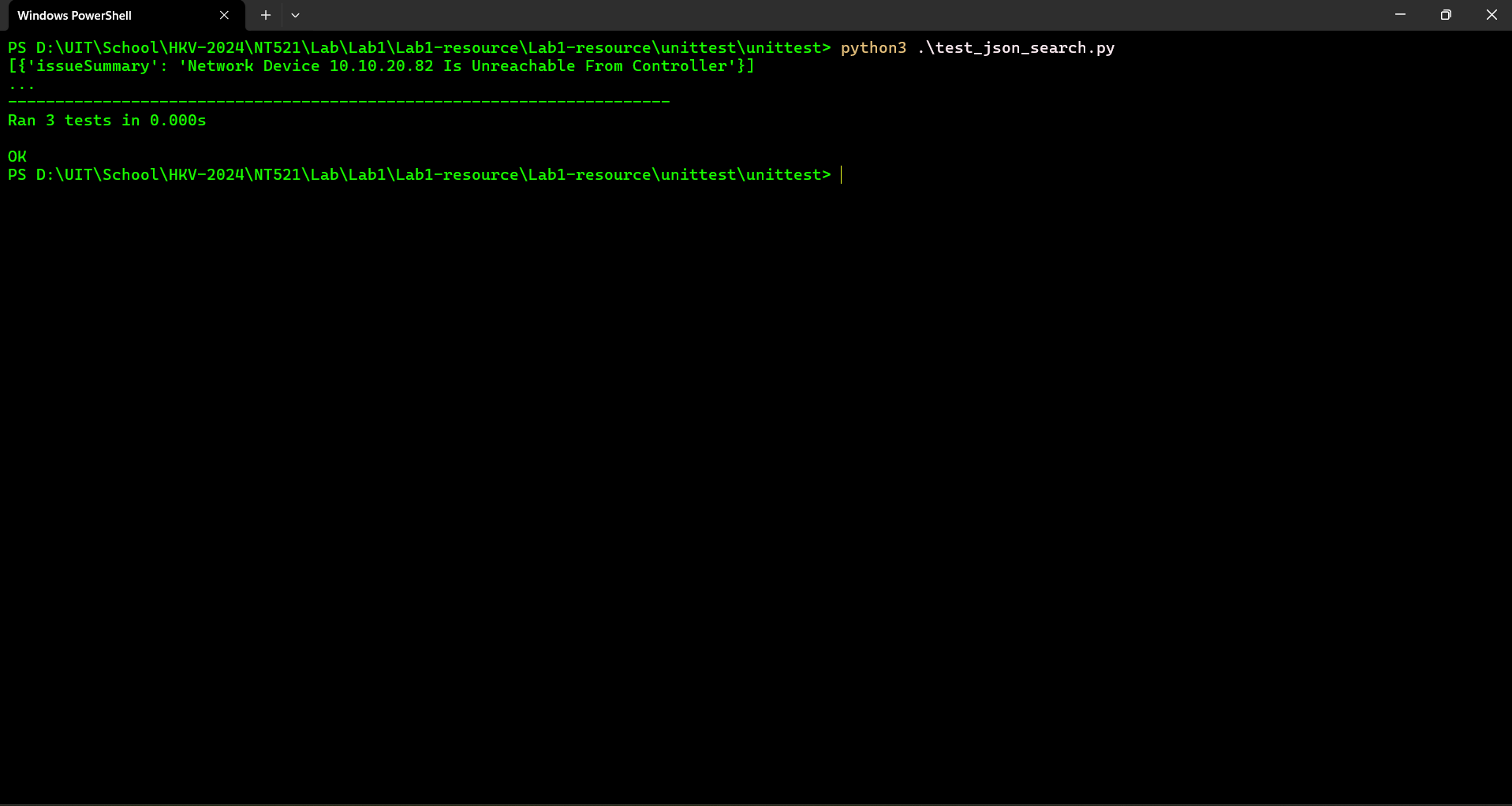
# Câu 8: Sửa lỗi lần nữa

Tìm ra vấn đề: Code ở trên sai vì nếu để global ret\_val, tất cả các value của ret\_val sẽ sử dụng chung 1 giá trị, có nghĩa là nếu đã gọi hàm lần đầu thì kết quả đó sẽ san sẻ cho những lần gọi sau. Ở đây, khi test function “test\_search\_found” nó trả về true, đồng thời value của key1 được store trong ret\_val thì khi khi sang function “test\_search\_not\_found” dù không tìm ra gì nhưng do không reset lại dữ liệu của ret\_val nên nó cũng sẽ trả về chuỗi ở function đầu tiên và do đó không pass được test.

Cách sửa lỗi:



Sửa lại vị trí đưa vào phạm vi scope của function và tạo 1 function trong function json\_search không chứa ret\_val để khi nó đệ quy thì nó sẽ không ảnh hưởng gì tới ret\_val, và khi gọi function mới thì giá trị của ret\_val sẽ trở lại thành [].



**Kết quả chạy:**

**Đã pass hết 3 test và thành công.**