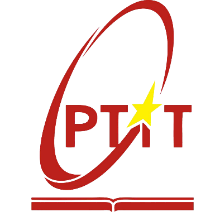
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

-------🙞🙜🕮🙞🙜-------



**CÁC KỸ THUẬT GIẤU TIN**

**Đề bài: Giấu tin trên miền nén video chất lượng cao**

|  |  |
| --- | --- |
| **HỌ TÊN:** | **NGUYỄN THÀNH LONG** |
| **MÃ SINH VIÊN:** | **B21DCAT120** |

**Giảng viên hướng dẫn: PGS.TS. Đỗ Xuân Chợ**

**HÀ NỘI, 04/2025**

**Mục lục**

[**I.** **Giới thiệu chung về bài thực hành** 4](#_Toc197089288)

[**II.** **Nội dung và hướng dẫn bài thực hành** 5](#_Toc197089289)

[**1.** **Mục đích** 5](#_Toc197089290)

[**2.** **Yêu cầu đối với sinh viên** 5](#_Toc197089291)

[**3.** **Nội dung thực hành** 5](#_Toc197089292)

[**III.** **Cài đặt và cấu hình máy ảo** 7](#_Toc197089293)

[**IV.** **Tích hợp và triển khai** 9](#_Toc197089294)

[**1.** **Docker hub** 9](#_Toc197089295)

[**2.** **Github** 9](#_Toc197089296)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Giao diện lab edit 7](#_Toc197089404)

[Hình 2. Cấu hình result 7](#_Toc197089405)

[Hình 3. Docker file bob 8](#_Toc197089406)

[Hình 4. Docker file Alice 9](#_Toc197089407)

[Hình 5.Docker hub 9](#_Toc197089408)

[Hình 6. Github 10](#_Toc197089409)

1. **Giới thiệu chung về bài thực hành**

Video chất lượng cao (High-Quality Video) là khái niệm dùng để chỉ những video có độ phân giải, tốc độ bit (bitrate), độ sắc nét, và độ chân thực màu sắc vượt trội so với các chuẩn video thông thường. Các yếu tố chính xác định một video chất lượng cao bao gồm:

* **Độ phân giải (Resolution)**: Số lượng pixel trong mỗi khung hình video. Các mức phổ biến:
  + HD (1280x720 pixels)
  + Full HD (1920x1080 pixels)
  + 4K Ultra HD (3840x2160 pixels) trở lên.
* **Tốc độ bit (Bitrate)**: Lượng dữ liệu được truyền mỗi giây, đơn vị thường tính bằng Mbps (Megabit per second). Bitrate cao cho phép giữ được nhiều chi tiết hình ảnh hơn, giảm nhiễu và artifacts.
* **Độ sâu màu (Color Depth)**: Số lượng bit dùng để biểu diễn mỗi màu trong một pixel. Độ sâu màu lớn (ví dụ 10-bit, 12-bit) giúp video hiển thị màu sắc mượt mà, tự nhiên hơn.
* **Tỷ lệ khung hình (Frame Rate)**: Số lượng khung hình hiển thị mỗi giây (fps). Tốc độ 60fps hoặc cao hơn thường được xem là tiêu chuẩn cho video chất lượng cao, đặc biệt trong các ứng dụng cần chuyển động mượt mà như thể thao hoặc game.
* **Thuật toán nén tiên tiến**: Video chất lượng cao thường được nén bằng các chuẩn hiệu quả như H.264/AVC, H.265/HEVC, nhằm cân bằng giữa kích thước tệp nhỏ và chất lượng hình ảnh tối ưu.

Trong kỹ thuật steganography trên miền nén, việc sử dụng video chất lượng cao đem lại nhiều lợi thế:

* **Tăng khả năng giấu tin**: Do số lượng dữ liệu lớn và cấu trúc nén phức tạp, các thay đổi nhỏ sẽ khó bị phát hiện hơn.
* **Bảo toàn thông điệp khi nén**: Các chuẩn nén tiên tiến như H.264 vẫn giữ được phần lớn thông tin ngay cả khi bitrate giảm nhẹ.
* **Độ bền cao**: Thông tin giấu có khả năng sống sót tốt hơn qua các quá trình chuyển mã hoặc tái nén.

1. **Nội dung và hướng dẫn bài thực hành**
   1. **Mục đích**

Bài thực hành nhằm giúp sinh viên hiểu và ứng dụng kỹ thuật **steganography** (giấu tin) trên **video nén chất lượng cao**, cụ thể là trên định dạng **H.264/AVC**.  
Thông qua bài lab, sinh viên sẽ:

* Nắm vững quy trình giấu và giải mã thông tin trong luồng video nén.
* Thực hành phân tích cấu trúc bitstream của video H.264.
* Thực hiện chèn thông điệp ẩn vào miền nén (thay vì miền ảnh thô như các phương pháp truyền thống).
* Đánh giá tính ổn định và khả năng phục hồi thông điệp khi video bị nén lại hoặc tái mã hóa.

Mục tiêu cuối cùng là giúp sinh viên tiếp cận kỹ thuật steganography hiện đại trên nền tảng video có tính ứng dụng thực tế cao.

* 1. **Yêu cầu đối với sinh viên**
* Có kiến thức cơ bản về kỹ thuật giấu tin trong đa phương tiện (steganography), đặc biệt là trên video.
* Hiểu rõ khái niệm, cấu trúc luồng dữ liệu và các ứng dụng của chuẩn video H264-AVC trong thực tế.
* Thành thạo việc sử dụng các công cụ xử lý media và các script Python hỗ trợ xử lý dữ liệu H264.
  1. **Nội dung thực hành**

Sinh viên khởi động bài lab

Chạy lệnh:

labtainer -r steg\_code2\_high-quality-video

*(Chú ý: sinh viên sử dụng <TÊN\_TÀI\_KHOẢN> của mình để nhập thông tin người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm.)*

Sau khi khởi động bài lab,có 2 container hiện lên là bob và alice. Sinh viên thực hiện làm lab theo yêu cầu trên 2 container này

TASK 1: Tiền xử lý video chất lượng cao

* Sau khi khởi động xong bài lab, có 2 terminal Bob và Alice
* Trong terminal Alice, đã có sẵn một file video chất lượng cao video\_input.mp4 và file python embed\_bmp.py để giấu thông điệp kèm theo thông điệp trong file secret.txt. Còn lại trong terminal Bob sẽ chứa file python extract\_bmp.py để có thể giải thông điệp
* Sinh viên cài thư viện Pillow của Python vào trong 2 terminal trên để có thể chạy được code

*python3 -m pip install Pillow*

* Sinh viên sử dụng công cụ ffmpeg để xử lý video nhằm lấy ra được khung I
* Trong lab này, ta sẽ giấu thông điệp vào khung I đầu tiên của video. Chạy lệnh sau để có thể lấy được khung I, việc sử dụng lệnh này sẽ cho phép chúng ta lấy được khung I đồng thời giải nén khung I đó luôn

*ffmpeg -i video\_input.mp4 -frames:v 1 i\_frame.bmp*

TASK 2: Giấu thông điệp vào khung ảnh I

* Tiến hành giấu thông điệp secret.txt vào frame i\_frame.bmp bằng code python

*python3 embed\_bmp.py i\_frame.bmp secret.txt stego\_iframe.bmp*

TASK 3: Ghép lại khung I chứa thông điệp vào video gốc

* Trong lab này ta sẽ sử dụng công cụ AtomicParsley để gắn ảnh stego\_iframe.bmp đã được giấu thông điệp vào vùng metadata của video gốc
* Do AtomicParsley hỗ trợ tốt với định dạng ảnh png, nên phải đổi ảnh ở dạng bmp về png

*ffmpeg -i stego\_iframe.bmp -compression\_level 0 stego\_iframe.png*

* Lệnh trên cho phép chuyển đổi ảnh mà không làm mất dữ liệu LSB đã giấu
* Giấu ảnh vào metadata của video bằng AtomicParsley để tạo ra video giống hệt với video gốc nhưng chứa thông điệp đã được giấu

*AtomicParsley video\_input.mp4 --artwork stego\_iframe.png -o final\_stego.mp4*

TASK 4: Giải mã thông điệp

* Khởi động dịch vụ ssh bên máy Bob

*sudo service ssh start*

* Gửi video hoàn chỉnh vừa tạo được từ Alice sang cho Bob với mật khẩu là “password”

*scp final\_stego.mp4 bob@"ip máy bob":/home/bob*

* Extract video để lấy khung I chứa thông điệp

*ffmpeg -i final\_stego.mp4 -map 0:v:1 -c copy extracted.png*

* Sử dụng python chạy code để extract thông điệp được giấu trong ảnh

*python3 extract\_bmp.py extracted.png*

Kết thúc lab:

* Trên terminal khởi động lab, sinh viên sử dụng lệnh:

***Stoplab***

* Khi bài lab kết thúc, một tệp lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab. Sinh viên cần nộp file .lab để chấm điểm.
* Để kiểm tra kết quả khi trong khi làm bài thực hành sử dụng lệnh:

***checkwork <tên bài thực hành>***

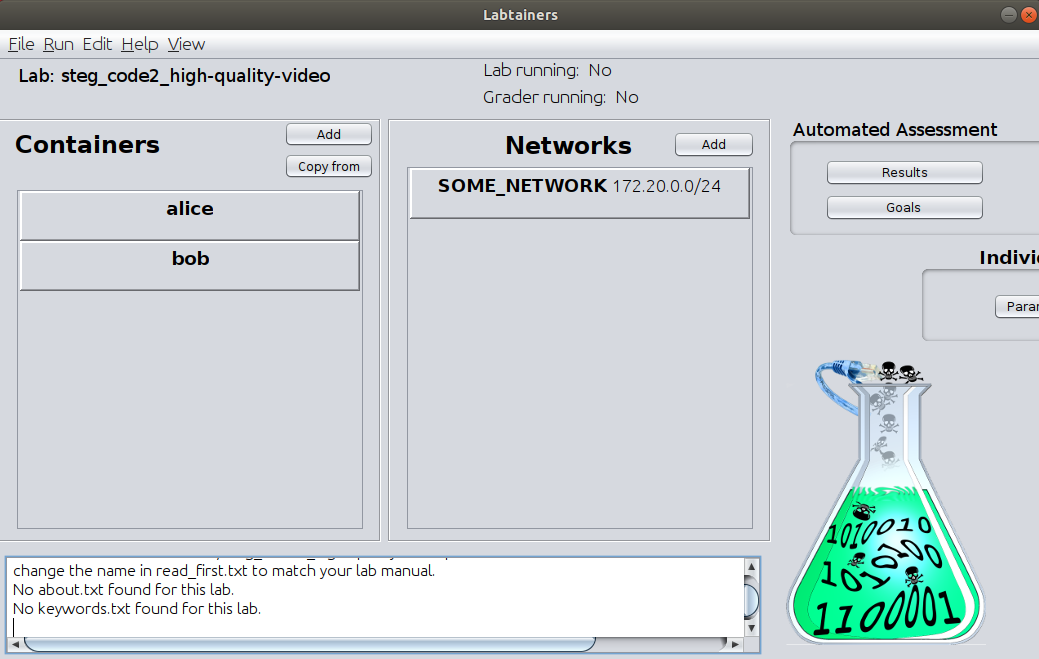
* Sinh viên cần nộp file .lab để chấm điểm.
* Kiểm tra kết quả trong quá trình làm bài:

***checkwork <tên bài lab>***

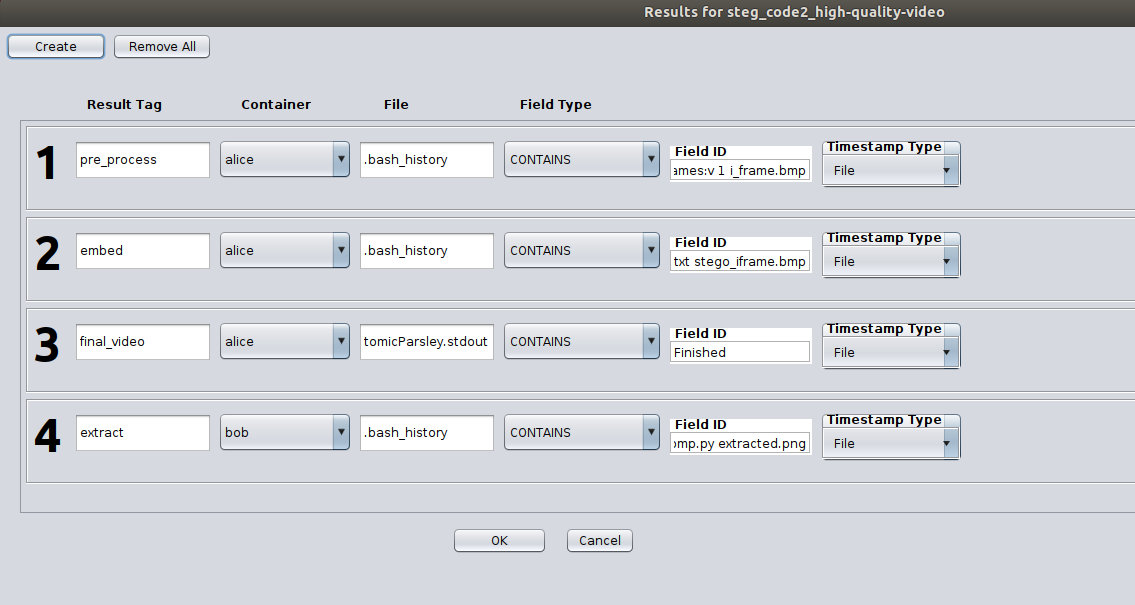
* Khởi động lại bài lab: trong quá trình làm bài sinh viện cần thực hiện lại bài lab, dùng câu lệnh:

***labtainer -r steg\_code2\_high-quality-video***

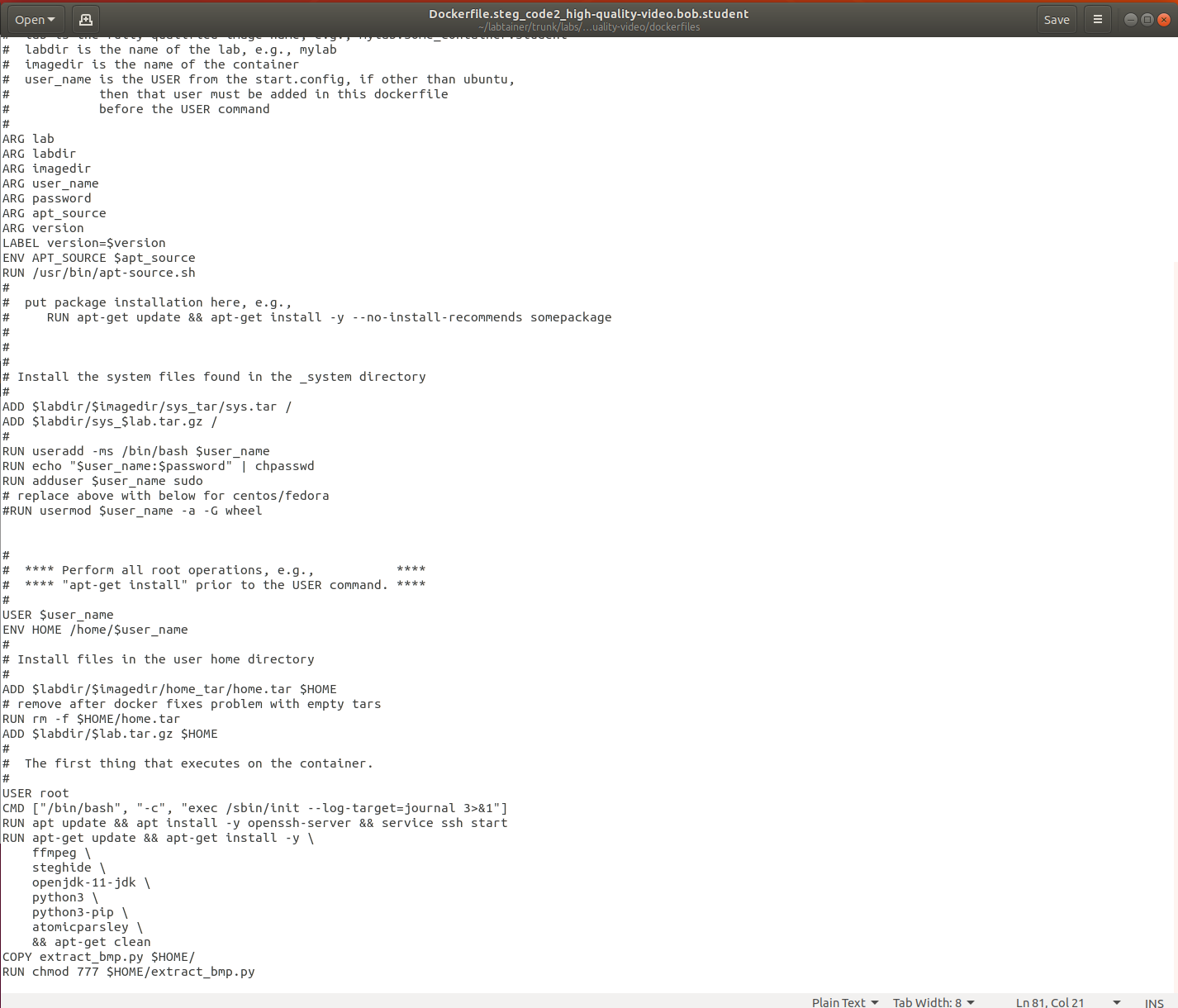
1. **Cài đặt và cấu hình máy ảo**



Hình 1. Giao diện lab edit



Hình 2. Cấu hình result

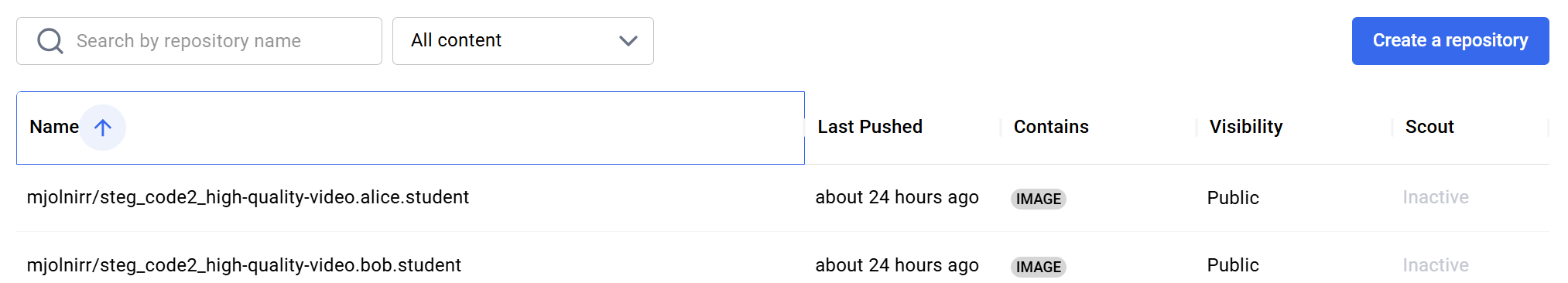


Hình 3. Docker file bob



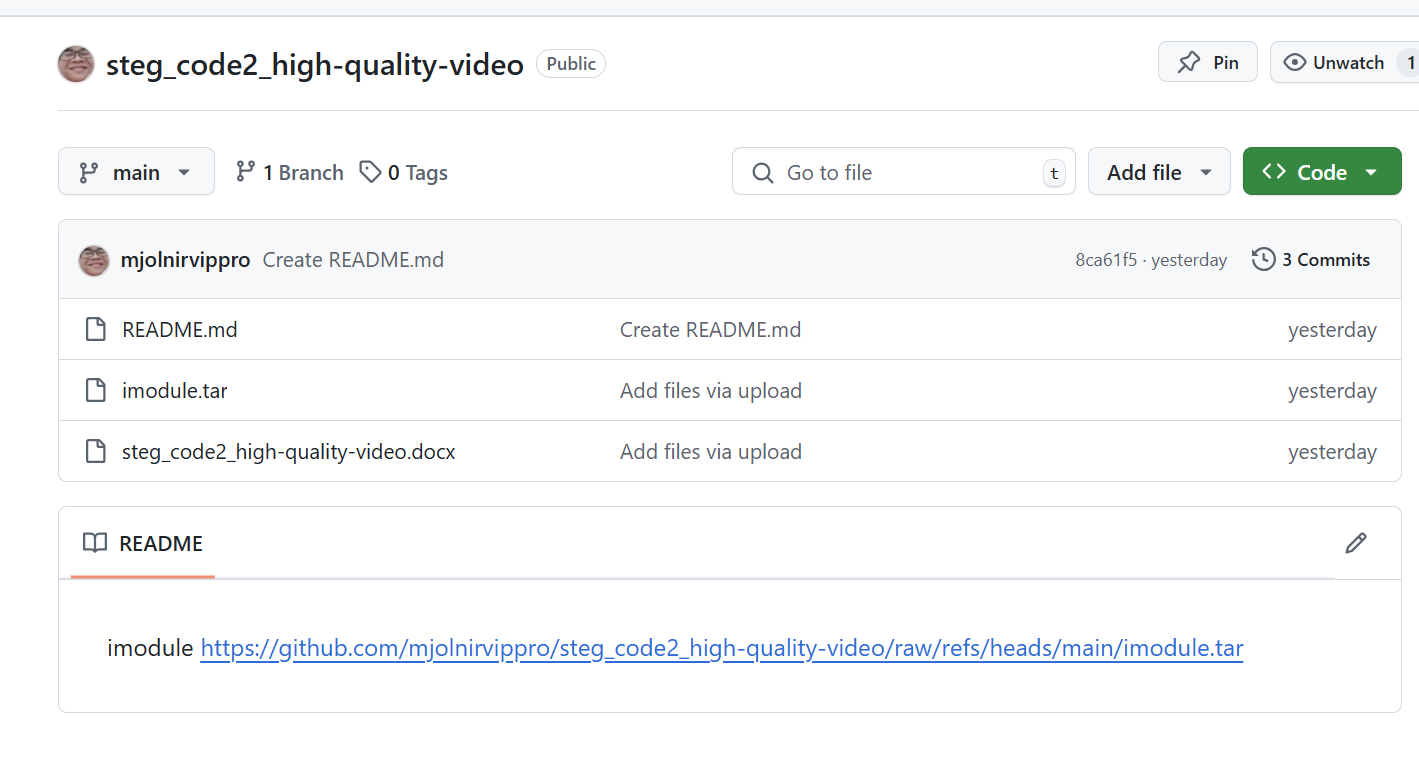
Hình 4. Docker file Alice

1. **Tích hợp và triển khai**
   1. **Docker hub**

****

Hình 5.Docker hub

* 1. **Github**

****

Hình 6. Github