

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN MÔN HỌC**  
**CÔNG NGHỆ TRUYỀN THÔNG ĐA PHƯƠNG TIỆN**  
**Đề tài: XÂY DỰNG HỆ THỐNG VIDEO STREAMING VỚI**  
**NGINX VÀ OWNCAST TRÊN MICROSOFT AZURE**

**Giảng viên hướng dẫn : TS. NGUYỄN TẤN HOÀNG PHƯỚC**  
**Lớp : NT536.O11.MMCL**

**NHÓM 11 – Sinh viên thực hiện:**

- |                      |                |
|----------------------|----------------|
| 1. Hồ Hải Dương      | MSSV: 21520202 |
| 2. Hồ Mạnh Đạt       | MSSV: 21520695 |
| 3. Nguyễn Phan Quyền | MSSV: 21522533 |
| 4. Võ Lê Thành Phát  | MSSV: 21522452 |

**THÀNH PHỐ HỒ CHÍ MINH – 2024**

## MỤC LỤC

A. LỜI MỞ ĐẦU .....	1
B. TỔNG QUAN ĐỀ TÀI .....	2
1. Giới thiệu Video Streaming .....	2
2. Lợi ích của Video Streaming .....	3
C. GIAO THỨC, CÔNG NGHỆ VÀ PHẦN MỀM .....	4
1. Các giao thức Streaming .....	4
1.1. Real-Time Messaging Protocol (RTMP) .....	4
1.2. Secure Reliable Transport (SRT) .....	4
2. Các công nghệ và phần mềm .....	5
2.1. Microsoft Azure .....	5
2.2. OBS Studio .....	5
2.3. Nginx với module nginx-rtmp.....	6
2.4. Owncast.....	6
2.5. VLC.....	7
2.6. Larix Broadcaster .....	7
2.7. SplitCam.....	8
D. CÀI ĐẶT, TRIỂN KHAI HỆ THỐNG.....	9
1. Mô hình hệ thống .....	9
2. Cài đặt và triển khai hệ thống .....	9
2.1. Cài đặt server trên Azure.....	9
2.2. Đối với module nginx .....	9
2.3. Ở máy stream video .....	11
2.4. Tại máy xem video.....	11
2.5. Tại Owncast Server .....	11
2.6. OBS Studio, SplitCam, Larix .....	13
TÀI LIỆU THAM KHẢO.....	17

## A. LỜI MỞ ĐẦU

Đồ án của nhóm chúng em là một đề tài về xây dựng ứng dụng video streaming với 100% mã nguồn mở dựa trên những yêu cầu thực tế của người dùng. Ứng dụng cho phép người streamer truyền tải trực tiếp video từ webcam của laptop (hoặc camera của điện thoại) cho 1 số lượng người xem nhất định. Ứng dụng còn có khả năng tùy chọn độ phân giải theo chất lượng đường truyền của người dùng đầu cuối. Ngoài ra, ứng dụng có thêm tính năng áp dụng các bộ lọc khác nhau (ví dụ filters: điều chỉnh màu, thêm sticker, thêm title,...) để chỉnh sửa nội dung video trước khi truyền đến người dùng.

Đồ án này đã đặt ra một thách thức lớn đối với nhóm chúng em về việc xây dựng, triển khai server cũng như mô hình ứng dụng. Tuy nhiên, nhờ có thầy Nguyễn Tấn Hoàng Phước đã tận tình giảng dạy, giúp đỡ chúng em về mặt lý thuyết để chúng em có thêm kiến thức và kỹ năng cần thiết để làm việc hiệu quả và giải quyết vấn đề.

Chúng em đã nỗ lực và cố gắng hoàn thành thật tốt đề tài được giao, nhưng cũng không thể tránh khỏi được những thiếu sót và những hạn chế trong quá trình hoàn thành đồ án. Mong thầy và các bạn thông cảm, góp ý thêm để đồ án của nhóm chúng em được hoàn thiện hơn.

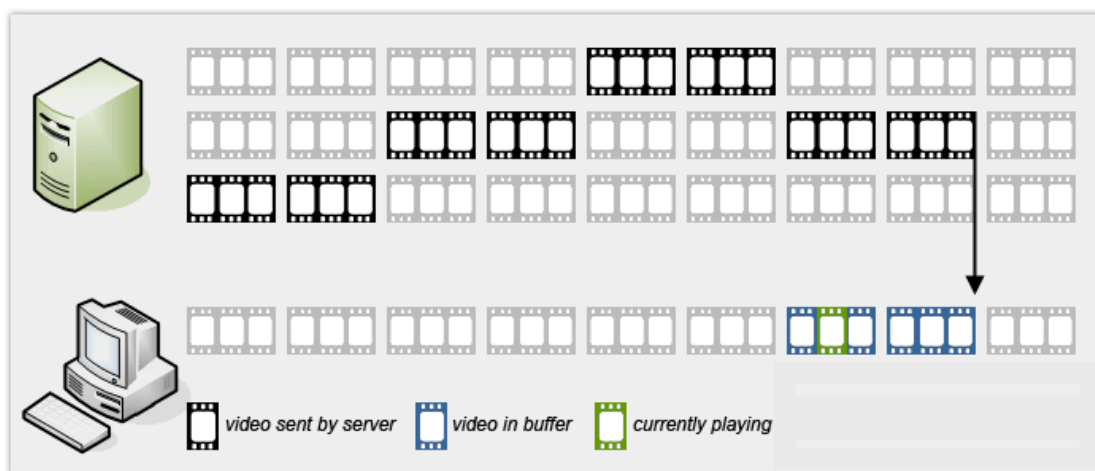
Sau đây, nhóm chúng em sẽ trình bày đồ án xây dựng ứng dụng video streaming với nginx và Owncast trên Microsoft Azure qua các chương sau:

- Tổng quan đề tài
- Tổng quan công nghệ
- Cài đặt, triển khai hệ thống.
- Demo
- Kết luận

## B. TỔNG QUAN ĐỀ TÀI

### 1. Giới thiệu Video Streaming

- Streaming media là kỹ thuật mà thực hiện liên tục quá trình nhận và hiển thị multimedia (video, audio,...) đến người dùng trong khi quá trình phân phối multimedia vẫn đang diễn ra.
- Video là một loại dữ liệu đa phương tiện, chủ yếu phục vụ cho truyền thông hoặc nhu cầu giải trí của con người trong nhiều thập niên.
- Streaming video là một kiểu của streaming media mà dữ liệu từ tệp video tiếp tục được truyền qua internet đến người dùng. Nó cho một video được xem trực tuyến mà không cần phải download về máy tính hay một thiết bị.
- Video truyền thống chỉ có thể hiển thị khi đã được tải về toàn bộ, việc làm này rất tiêu tốn thời gian còn Video Streaming tiết kiệm thời gian hơn cho người dùng bằng cách sử dụng các công nghệ giải nén kết hợp với hiển thị dữ liệu đồng thời trong lúc vẫn tiếp tục tải Video về (được gọi là kỹ thuật buffering).
- Kỹ thuật Buffering thay vì gửi 1 lần duy nhất, dữ liệu sẽ được chia ra thành từng gói nhỏ sau đó liên tục truyền những phần được chia ra. Bên nhận sẽ lấy 1 phần chia nhỏ của Video và hiển thị những phần đã nhận được, đồng thời sẽ lấy về các gói dữ liệu còn lại để kịp cho việc hiển thị tiếp theo. Minh họa ở hình dưới:



- Streaming Video được thể hiện dưới 2 loại:
  - **Video on demand – VoD (Video theo yêu cầu):** là các dữ liệu Video được lưu trữ trên máy chủ đa phương tiện và được truyền đến người dùng khi có yêu cầu, người dùng có toàn quyền để hiển thị cũng như thực hiện các thao tác (tua, dừng, quay lại...) với các đoạn dữ liệu này. Ví dụ: Xem phim, nhạc online trên zing, youtube,...

- **Live Streaming (Video thời gian thực):** là các dữ liệu Video được chuyển phát trực tiếp từ các nguồn cung cấp dữ liệu theo thời gian thực (máy camera, microphone, thiết bị phát dữ liệu Video...). Ví dụ: youtube live, facebook live, bigo,...

## 2. Lợi ích của Video Streaming

Video live streaming mang lại nhiều lợi ích cho người dùng, doanh nghiệp và giáo dục. Dưới đây là một số lợi ích chính của video live streaming:

- Tạo ra sự kết nối và tương tác: Video live streaming cho phép người xem tương tác trực tiếp với người phát sóng thông qua các tính năng như bình luận trực tiếp, thả tim hoặc tương tác khác.
- Tiết kiệm thời gian và chi phí: Video live streaming giúp tiết kiệm thời gian và chi phí cho các sự kiện trực tuyến như hội thảo, buổi biểu diễn âm nhạc, buổi phát sóng trực tiếp của trận đấu thể thao và các hoạt động giải trí khác.
- Truyền tải thông điệp nhanh chóng: Việc tạo ra một video live stream chỉ mất vài phút và người xem có thể truy cập và xem nó ngay lập tức.
- Tăng tương tác và tầm nhìn: Việc tạo ra một video live stream hoặc tham gia vào một video live stream có thể giúp tăng lượng truy cập và tương tác trên các kênh truyền thông xã hội của người dùng.
- Giải trí và giảng dạy trực tuyến: Video Streaming thường được dùng trong lĩnh vực giải trí và dạy học, dùng để lưu trữ các tập tin Video hoặc bài học, cung cấp cho người dùng các tiện ích như tìm kiếm, liệt kê, khả năng hiển thị hoặc hiển thị lại các dữ liệu Video theo yêu cầu.

## C. GIAO THỨC, CÔNG NGHỆ VÀ PHẦN MỀM

### 1. Các giao thức Streaming

#### 1.1. Real-Time Messaging Protocol (RTMP)

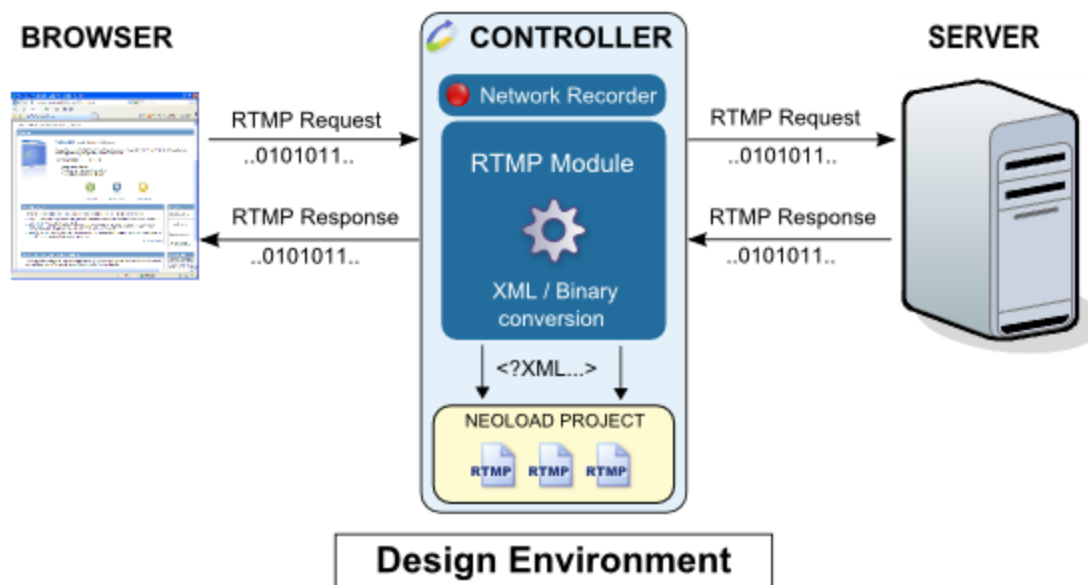
Là giao thức được sử dụng cho streaming các nội dung đa phương tiện qua internet, phát triển bởi tập đoàn Adobe. RTMP là cho phép streaming với độ trễ thấp, có thể dùng để phát theo kiểu VoD hoặc Live stream.

Ở chế độ tiêu chuẩn, RTMP chạy trên TCP sử dụng port 1935. Ngoài ra, nó còn hỗ trợ một số biến thể khác như: RTMPS- RTMP qua kết nối TLS/SSL

+ RTMPE - RTMP encrypted sử dụng kỹ thuật mã hóa của Adobe

+ RTMPT – RTMP Traverse sử dụng để đóng gói yêu cầu HTTP cho vượt firewall.

+ RTMFP – RTM Flow Protocol, stream sử dụng UDP thay cho TCP



Cú pháp sử dụng rtmp như sau:

```
rtmp://[username:password@]server[:port][[/app]][[/instance]][/playpath]
```

#### 1.2. Secure Reliable Transport (SRT)

Là một giao thức truyền tải video qua mạng Internet, giúp đảm bảo chất lượng video không bị giảm sút do mất mát dữ liệu.

## 2. Các công nghệ và phần mềm

### 2.1. Microsoft Azure

Microsoft Azure là nhà cung cấp dịch vụ điện toán đám mây hàng đầu, cung cấp cho người dùng một framework lớn để lưu trữ dữ liệu, phát triển phần mềm và các giải pháp đổi mới khác.



\* Lý do chọn Microsoft Azure:

- + Dễ dàng quản lý các hoạt động trên web
- + Dễ dàng mở rộng update trang web
- + Khả năng tương thích cao, có server ở tất cả mọi nơi trên thế giới
- + Đáp ứng các tác vụ trên một web tốc độ nhanh, cải thiện trải nghiệm người dùng
- + An toàn và bảo mật

### 2.2. OBS Studio

OBS (Open Broadcaster Software) là một phần mềm mã nguồn mở được sử dụng để ghi và phát trực tiếp (stream) nội dung từ máy tính cá nhân lên các nền tảng trực tuyến như Twitch, YouTube, Facebook Live và nhiều nền tảng khác thông qua giao thức như RTMP.



\* Lý do chọn OBS Studio:

- + Tính năng đa nền tảng
- + Bảng điều khiển đa chức năng
- + Giao diện đơn giản, dễ sử dụng

- + Hỗ trợ đa nguồn
- + Có thể điều chỉnh chất lượng video
- + Hỗ trợ đa kênh âm thanh
- + Hỗ trợ cộng đồng

### 2.3. Nginx với module nginx-rtmp

Nginx-rtmp là module mở rộng, kết hợp với Nginx để cho phép xây dựng máy chủ streaming media.



*\* Lý do chọn Nginx với module nginx-rtmp:*

- + Hiệu suất ổn định
- + Hỗ trợ nhiều tính năng streaming
- + Tính linh hoạt
- + Tương thích với nhiều ứng dụng.

### 2.4. Owncast

Owncast là một máy chủ trò chuyện web và phát trực tiếp mã nguồn mở miễn phí đang phát triển cho phép tạo nội dung đa phương tiện trực tiếp phục vụ cho khán giả.



*\* Lý do chọn Owncast:*

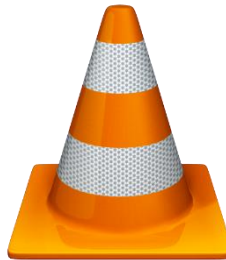
- + Mã nguồn mở, miễn phí, không quảng cáo
- + Cho phép chỉnh sửa, tùy chỉnh sâu



- + Tương thích với các phần mềm streaming hiện nay
- + Người dùng không cần tạo tài khoản cũng có thể tham gia live stream

## 2.5. VLC

VLC (Video LAN Client) là một phần mềm mã nguồn mở, là một trong những trình phát đa phương tiện phổ biến và sử dụng rộng rãi trên nhiều nền tảng.



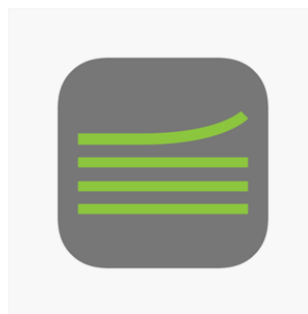
*\* Lý do chọn VLC:*

- + Hỗ trợ phát video và âm thanh với nhiều định dạng, độ phân giải khác nhau
- + Stream trực tuyến
- + Cung cấp bộ điều khiển hoàn chỉnh
- + Cho phép chuyển đổi định dạng video và âm thanh
- + Tiện ích mở rộng
- + Hỗ trợ đa nền tảng

## 2.6. Larix Broadcaster

Larix Broadcaster là một ứng dụng đơn giản và mạnh mẽ để truyền tải video và âm thanh từ xa. Nó cho phép truyền phát nội dung trực tiếp từ thiết bị di động của bạn trong thời gian thực qua WiFi, 3G, LTE, 5G thông qua SRT/RTMP/NDI/WebRTC/RTSP/RIST.

+ *Nhược điểm:* Có hình mờ trừ khi đăng ký Larix premium.



*\* Lý do chọn Larix Broadcaster:*

- + Hỗ trợ đa nền tảng
- + Hỗ trợ đa giao thức
- + Chất lượng cao
- + Tính năng phong phú

## **2.7. SplitCam**

SplitCam là một ứng dụng hữu ích dành cho những ai mong muốn thêm nhiều hiệu ứng độc đáo cho webcam để thực hiện cuộc gọi video với bạn bè của mình.

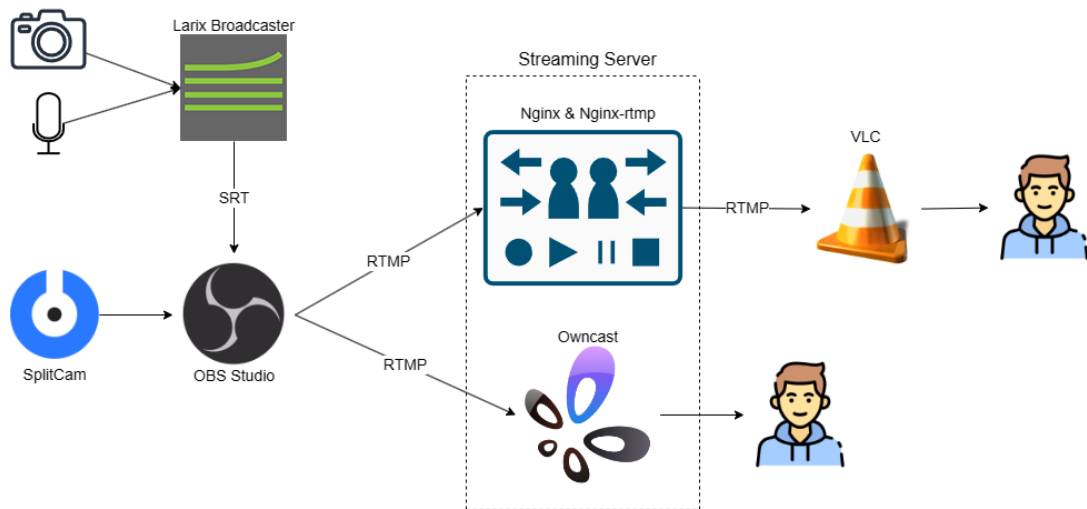


*\* Lý do chọn SplitCam:*

- + Cho phép tạo nhiều hiệu ứng độc đáo cho webcam
- + Hỗ trợ đa nền tảng
- + Hiệu suất cao
- + Tính năng phong phú

## D. CÀI ĐẶT, TRIỂN KHAI HỆ THỐNG

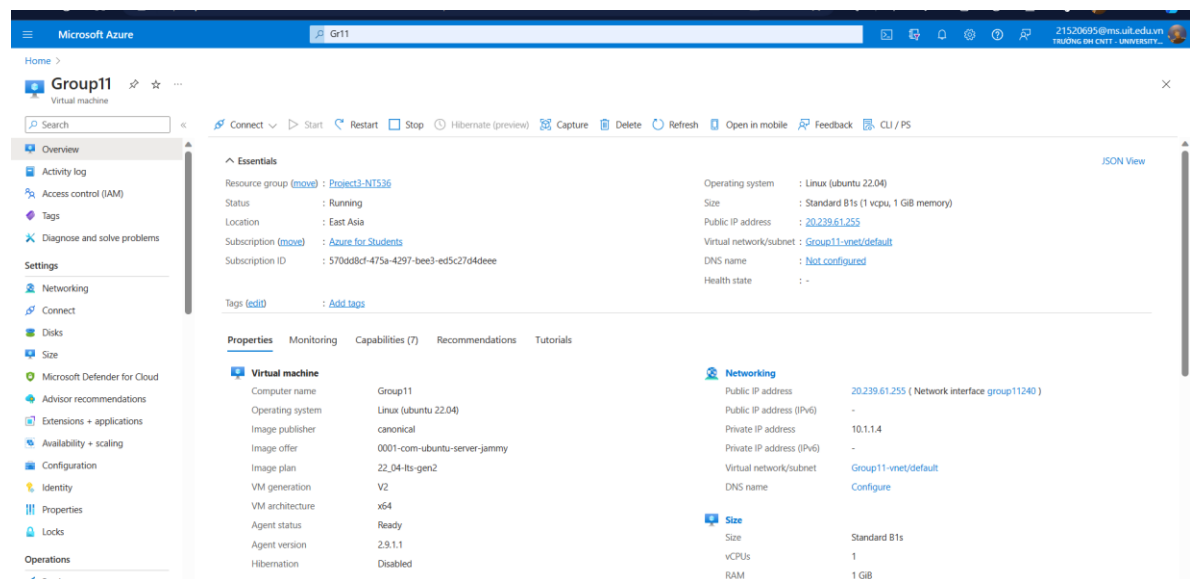
### 1. Mô hình hệ thống



### 2. Cài đặt và triển khai hệ thống

#### 2.1. Cài đặt server trên Azure

- Nhóm em sử dụng máy Ubuntu 22.04 rồi cài đặt server:



#### 2.2. Đối với module nginx

- Nhóm em cài đặt module nginx bằng lệnh:

```
sudo apt install libnginx-mod-rtmp
```

- Vào file config bằng lệnh:

```
sudo nano /etc/nginx/nginx.conf
```

- Sau đó thêm nội dung sau vào file config:

```
rtmp {  
    server {  
        listen 1935;  
        chunk_size 4096;  
        allow publish 127.0.0.1;  
        allow publish all;  
        application live {  
            live on;  
            record off;  
        }  
    }  
}
```

- Để bật giao thức rtmp, và sử dụng config allow publish all để OBS studio có thể kết nối đến server thông qua external IP của máy chủ trên Cloud, và listen trên port 1935 và sau đó em sẽ allow port 1935 với giao thức TCP thông qua lệnh: “sudo ufw allow 1935/tcp”, cuối cùng sử dụng lệnh: “sudo systemctl reload nginx.service”.
- Để gửi video đến server em sử dụng: “ffmpeg” một tiện ích âm thanh-video dòng lệnh phổ biến, để phát tệp video trực tiếp trên máy chủ.
- Đầu tiên em cài đặt Python và trình quản lý pip của nó “pip”:

```
sudo apt install python3-pip
```

- Sử dụng pip để cài đặt youtube-dl:

```
sudo pip install youtube-dl
```

- Sau đó là cài đặt ffmpeg:

```
sudo apt install ffmpeg
```

### 2.3. Ở máy stream video

Cài đặt OBS studio để stream, sau đó vào mục Setting => Stream để chọn máy chủ mà RTMP gửi thông tin về/;

Server: rtmp://your\_domain/live

Play Path/Stream Key: stream

Ở phần “yourdomain” em sẽ thay bằng External IP của máy chủ, còn phần stream key sẽ là tùy chọn của người stream ở đây, em sẽ chọn là “test”. Nếu khi bật bắt đầu stream mà OBS studio không báo lỗi => Đã kết nối thành công đến máy chủ.

### 2.4. Tại máy xem video

Cài đặt và sử dụng VLC => sau khi bật lên vào phần media => open networkstream => Nhập link theo cú pháp rtmp://your\_domain/live/stream\_key  
Tại đây em sẽ thay your\_domain ip bằng ip của máy chủ và stream\_key theo đúng như phần Play Path/Stream Key: đã set ở trên OBS Studio của máy phát stream.

### 2.5. Tại Owncast Server

Truy cập đường link sau: <https://owncast.online/quickstart/installer/>

Sao chép câu lệnh “curl -s https://owncast.online/install.sh | bash” và dán vào terminal ở máy linux.

Nếu máy linux chưa được cập nhật câu lệnh curl thì sẽ bị lỗi, khi đó nhập: “sudo apt install curl”, và sau đó nhập lại “curl -s https://owncast.online/install.sh | bash”.

Dòng thông báo sẽ hiện sau khi đã cài đặt thành công:

```
Success! Run owncast by changing to the owncast directory and run ./owncast.  
The default port is 8080 and the default streaming key is abc123.  
Visit https://owncast.online/docs/configuration/ to learn how to configure your  
new Owncast server.
```

Tiếp tục nhập câu lệnh điều hướng đến thư mục của owncast: “cd owncast/”

Tiếp tục nhập: “./owncast”

Khi đó có thể sẽ hiện lỗi như thế này:

```
nt536@pr3-gr11:~/owncast$ ./owncast  
INFO[2024-01-10T13:47:40Z] Owncast v0.1.2-linux-64bit (b2c0ab46621ed7832b3cead59  
4fb24cf21c395c5)  
INFO[2024-01-10T13:47:40Z] Web server is listening on port 8080.  
INFO[2024-01-10T13:47:40Z] Configure this server by visiting /admin.  
FATA[2024-01-10T13:47:40Z] listen tcp :1935: bind: address already in use
```

Lý do xuất hiện lỗi là có vài tiến trình đang được lắng nghe ở port 1935 (address already in use). Chúng ta phải tiến hành xóa (kill) nó đi:

- Trước hết cần kiểm tra xem có những tiến trình nào đang được lắng nghe tại port 1935 bằng câu lệnh: “sudo lsof -i: 1935”
- Sau khi đã hiện ID của những tiến trình đó, tiến hành kill nó bằng câu lệnh: “sudo kill -9 **ID của những tiến trình đó**”

Tiếp tục quay trở lại thực hiện các câu lệnh “cd owncast/” và “./owncast”

```
nt536@pr3-gr11:~/owncast$ ./owncast
INFO[2024-01-10T13:57:42Z] Owncast v0.1.2-linux-64bit (b2c0ab46621ed7832b3cead594fb24cf21c395c5)
INFO[2024-01-10T13:57:42Z] Web server is listening on port 8080.
INFO[2024-01-10T13:57:42Z] Configure this server by visiting /admin.
```

Lúc này hệ thống đã hoạt động bình thường.

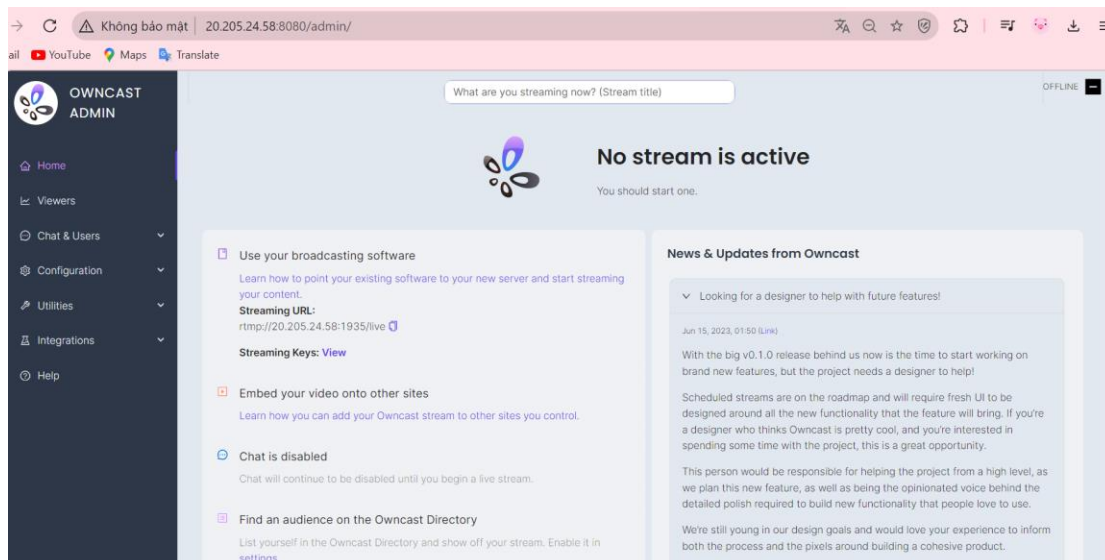
Kiểm tra xem địa chỉ IP public của máy linux hiện tại là bao nhiêu, ở đây địa chỉ của nhóm ở máy linux trên Azure là: 20.205.24.58

Connect ▾ ▶ Start ↺ Restart □ Stop ⌚ Hibernate (preview) 📷 Capture 🗑 Delete ↻ Refresh 📱 Open in mobile 🗨 Feedback 🖨 CLI / PS	
^ Essentials	
Resource group (move) :	NT536-Group11
Status :	Running
Location :	East Asia (Zone 1)
Subscription (move) :	Azure for Students
Subscription ID :	02fb02ba-e8ad-4596-8e76-6ad73679990b
Availability zone :	1
Tags (edit) :	Add tags
Operating system :	Linux (ubuntu 22.04)
Size :	Standard B1s (1 vcpu, 1 GiB memory)
Public IP address :	20.205.24.58
Virtual network/subnet :	project3-group11-vnet/default
DNS name :	Not configured
Health state :	-

Tiếp theo nhóm sẽ mở một trang web ở Chrome bằng câu lệnh “20.205.24.58:8080/admin/” để vào trang dành cho quản trị viên.

Khi đó sẽ được yêu cầu nhập thông tin đăng nhập, nhóm nhập username và password mặc định là “admin” và “abc123”.

Sau đó trang dành cho quản trị viên đã hiện lên:



Tại đây chúng ta có thể xem tất cả các thông tin cần thiết như số lượng người xem livestream, tin nhắn trong lúc live, các port mà server sử dụng, chỉnh sửa thông tin cơ bản, chỉnh sửa cấu hình khi livestream,...

## 2.6. OBS Studio, SplitCam, Larix

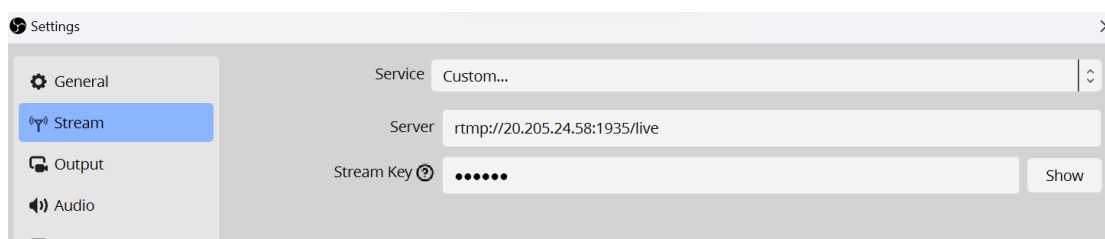
### 2.6.1. Tại OBS

Trước hết, vào lại trang owncast admin để lấy đường link livestream:

**Streaming URL:**  
rtmp://20.205.24.58:1935/live

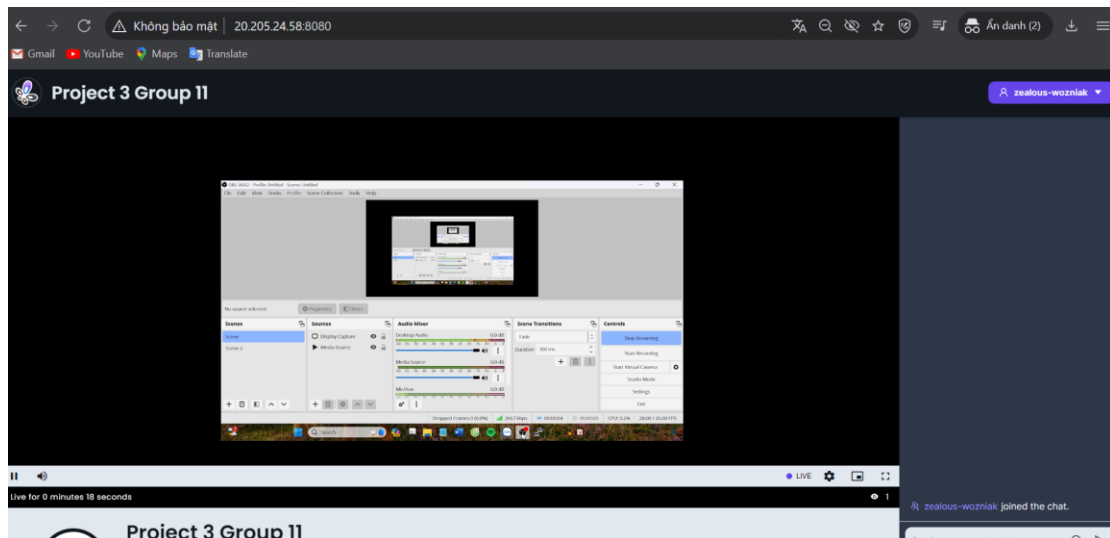
**Streaming Keys:** [View](#)

Sau đó vào OBS, chọn Settings, chọn Stream, và dán đường link vào, nhập password mặc định lúc ban đầu là “abc123” (nếu chưa có thay đổi gì), nhấn Apply:



Tiến hành livestream nhấn “Start Streaming”, khi đó mọi người có thể vào xem livestream bằng đường link: “20.205.24.58:8080”.

Giao diện khi mọi người vào xem live bằng đường dẫn:

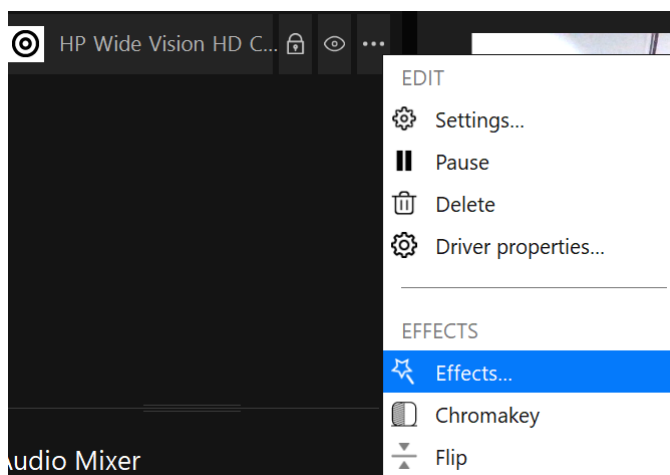


Bên phía người dùng có thể tương tác trực tiếp với nhau thông thanh tin nhắn ở bên phải màn hình, biết được khi nào kết thúc live, tăng giảm âm lượng, chỉnh chất lượng hình ảnh,...

Bên phía người livestream có thể kiểm soát được thông tin buổi livestream đó tại trang của quản trị viên, họ có thể kiểm soát được: số lượng người xem, đồ thị tăng giảm lượng views theo thời gian, tin nhắn giữa những người xem live với nhau, chất lượng buổi livestream,...

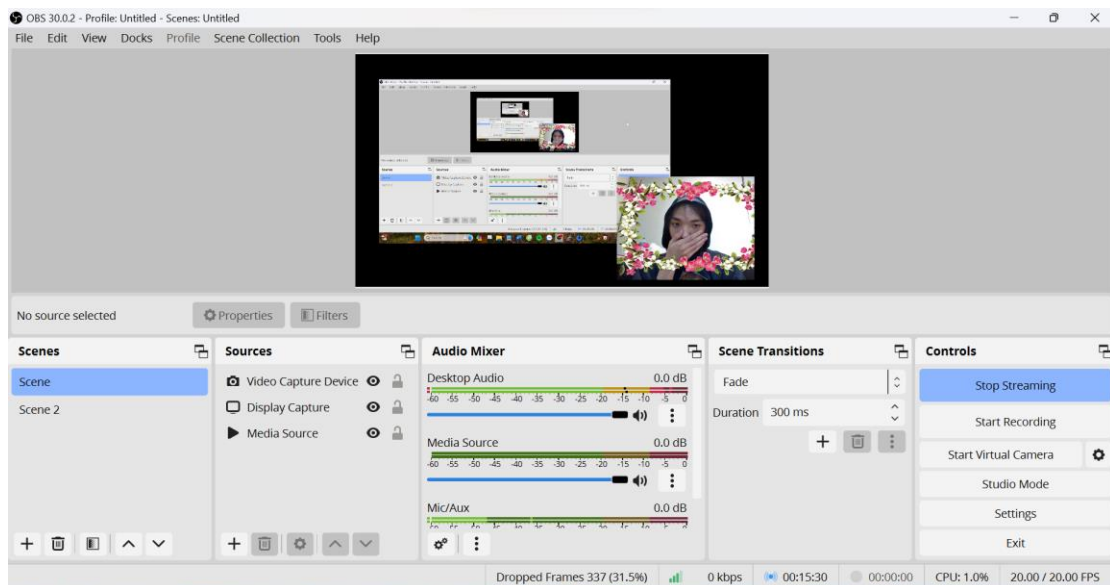
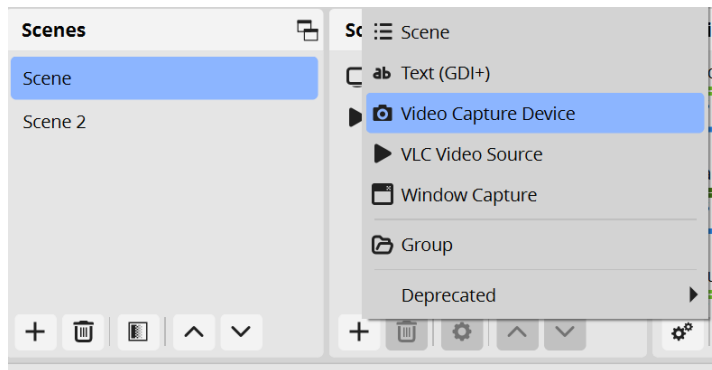
### 2.6.2. SplitCam

Với SplitCam có rất nhiều tính năng để chúng ta chỉnh sửa cho buổi livestream sao cho đẹp mắt nhất, và chúng ta có thể thay đổi màu sắc hiển thị cho hình ảnh chúng ta khi live, thêm filter cho nhìn sống động tại Effects:



Sau đó kết nối với OBS tại Video Capture Device:

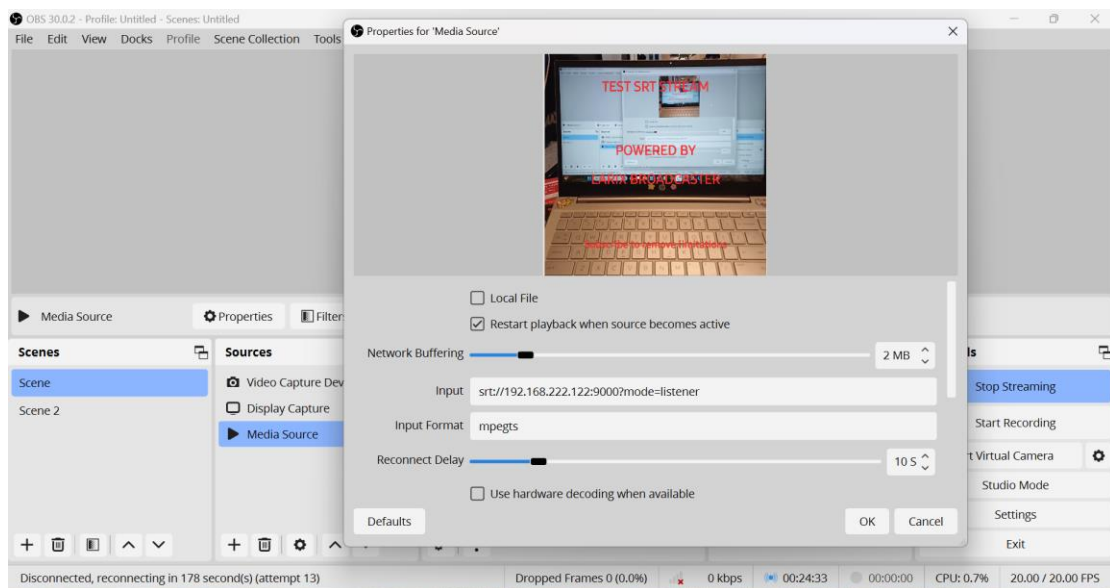




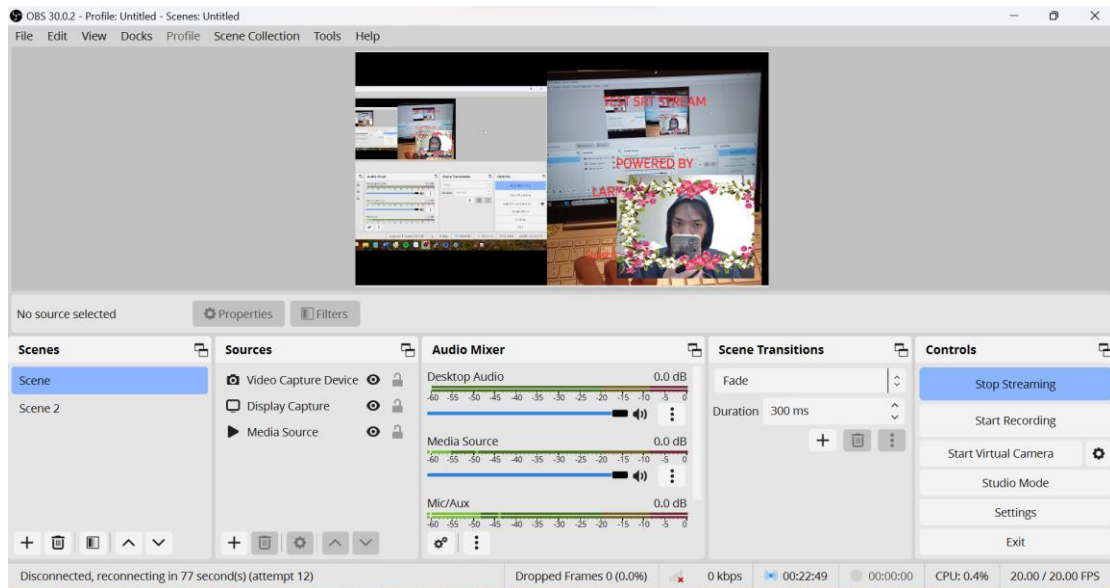
### 2.6.3. Larix

Với Larix chúng ta sẽ tải ứng dụng này trên điện thoại thông minh của mình, phải đảm bảo rằng cả máy tính và điện thoại đều kết nối chung 1 mạng.

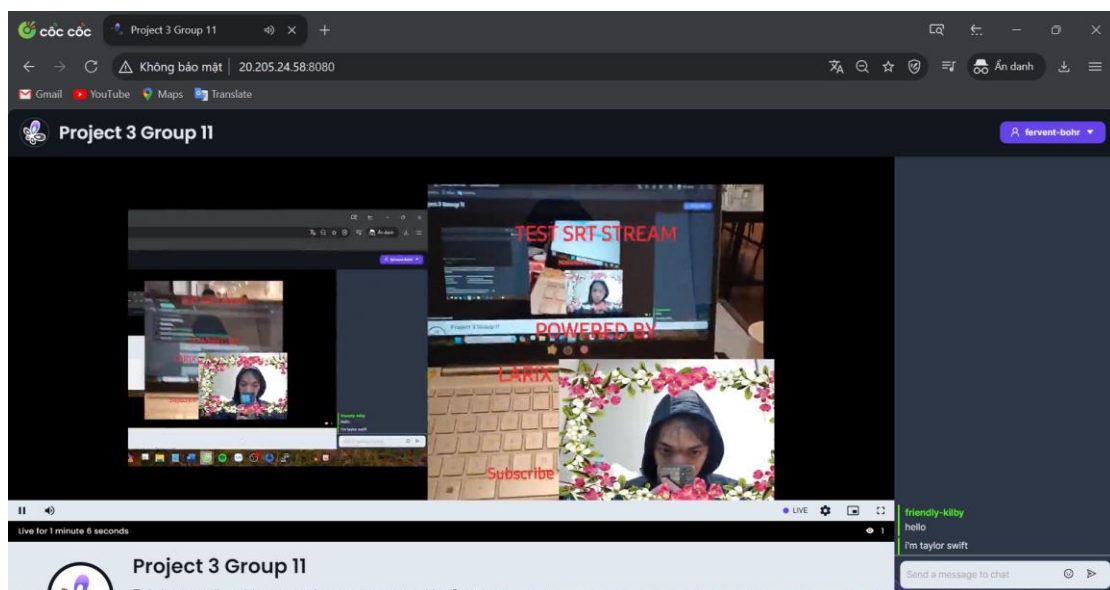
Sau đó nhập đường liên kết để kết nối với nhau tại Media Source:



Hình ảnh thu được từ điện thoại có Larix sẽ phát qua và hiển thị lên máy tính có OBS thông qua giao thức SRT:



Từ phía người dùng có thể xem đầy đủ những gì chúng ta đã hiển thị trên OBS:



## TÀI LIỆU THAM KHẢO

- [1] <https://github.com/keepwalking86/streaming/tree/master/docs>
- [2] <https://gostudio.co/vi/rtmp-la-gi-tim-hieu-ve-cong-nghe-livestream-va-cach-hoat-dong.310.html>
- [3] [https://en.wikipedia.org/wiki/OBS\\_Studio](https://en.wikipedia.org/wiki/OBS_Studio)
- [4] <https://en.wikipedia.org/wiki/Nginx>
- [5] [https://en.wikipedia.org/wiki/VLC\\_media\\_player](https://en.wikipedia.org/wiki/VLC_media_player)
- [6] <https://github.com/owncast/owncast>
- [7] [https://www.youtube.com/watch?v=eWE7HGJQI\\_4](https://www.youtube.com/watch?v=eWE7HGJQI_4)
- [8] <https://softvelum.com/larix/docs/>
- [9] <https://cctvapp.net/2021/06/22/huong-dan-tan-dung-camera-chuyen-thanh-webcam-voi-splitcam/>