

Cài đặt Nginx RTMP và HLS

Mục đích

Để có thể sử dụng được tính năng streaming video trên server, ta cần phải setup server của mình thành một nơi vừa có thể chứa video và vừa có thể phát video trong danh sách video đó. Để có thể sử dụng tính năng này, ta cần cài đặt giao thức RTMP và HLS cho server.

RTMP là một giao thức dựa trên TCP, duy trì các kết nối liên tục và cho phép giao tiếp với độ trễ thấp. Để phân phối luồng một cách trơn tru và truyền càng nhiều thông tin càng tốt, nó chia luồng thành các đoạn (fragments) và kích thước của chúng được thỏa thuận tùy vào thỏa thuận giữa máy khách và máy chủ. Đôi khi, kích thước đó được giữ nguyên. Nginx cung cấp cho chúng ta cách thức cấu hình rtmp thông qua phiên bản có tính phí [nginx plus](#) nhưng ta cũng có thể cài đặt phiên bản free thông qua [nginx - nginx-rtmp-module](#)

HLS (HTTP live streaming) là một giao thức truyền bitrate dựa trên HTTP. HLS được lập trình bởi Apple Inc. Là một giao thức truyền phát nội dung đa phương tiện khá phổ biến. HLS giống với MPEG-DASH ở chỗ nó hoạt động bằng cách chia luồng tổng thể thành một chuỗi các tệp tải xuống dựa trên HTTP, mỗi lần tải xuống tải một đoạn ngắn của một luồng truyền tải tổng thể có khả năng không bị ràng buộc. Một danh sách các luồng khả dụng, được mã hóa ở các tốc độ bit khác nhau, được gửi đến máy client bằng cách sử dụng danh sách phát m3u mở rộng.

Chuẩn bị

- Server Ubuntu 20.04

Các bước tiến hành

Đầu tiên, ta tiến hành cài đặt ffmpeg để phân giải video về dạng m3u mở rộng

```
sudo add-apt-repository ppa:jonathonf/ffmpeg-4
```

```
sudo apt-get update
```

```
sudo apt-get install -y ffmpeg
```

Tiếp theo, ta cài đặt nginx. Vì ta cần import thêm module **rtmp** phiên bản free vào nginx nên không thể cài bằng apt được mà phải tự build một file cài đặt. Ta tiến hành tải các module cần thiết.

```
sudo apt install build-essential -y
```

```
wget http://nginx.org/download/nginx-1.14.2.tar.gz
```

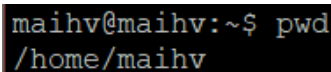
```
tar xzvf nginx-1.14.2.tar.gz
wget https://ftp.pcre.org/pub/pcre/pcre-8.40.tar.gz
tar xzvf pcre-8.40.tar.gz
wget http://www.zlib.net/zlib-1.2.11.tar.gz
tar xzvf zlib-1.2.11.tar.gz
wget https://www.openssl.org/source/openssl-
1.1.0l.tar.gz
tar xzvf openssl-1.1.0l.tar.gz
```

Tiến hành tải module *nginx-rtmp* về

```
wget https://github.com/arut/nginx-rtmp-
module/archive/master.zip
sudo apt install -y unzip
unzip master.zip
```

Bây giờ ta cần kiểm tra xem được dẫn hiện tại mà ta vừa tải các gói về là gì, gõ lệnh:

```
pwd
```



```
maihv@maihv:~$ pwd
/home/maihv
```

Như vậy các module vừa tải nằm trong thư mục */home/maihv*. Bạn phải kiểm tra kĩ phần này để có thể thay các giá trị đó vào phần config *--add-module, --with-openssl, --with-pcre, --with-zlib* của câu lệnh sau.

```
cd nginx-1.14.2
./configure --prefix=/usr/share/nginx --sbin-
path=/usr/sbin/nginx --modules-path=/usr/lib/nginx/modules --
conf-path=/etc/nginx/nginx.conf --error-log-
path=/var/log/nginx/error.log --http-log-
path=/var/log/nginx/access.log --pid-path=/run/nginx.pid --
lock-path=/var/lock/nginx.lock --user=www-data --group=www-data
--build=Ubuntu --http-client-body-temp-path=/var/lib/nginx/body
--http-fastcgi-temp-path=/var/lib/nginx/fastcgi --http-proxy-
temp-path=/var/lib/nginx/proxy --http-scgi-temp-
path=/var/lib/nginx/scgi --http-uwsgi-temp-
path=/var/lib/nginx/uwsgi --with-openssl=/home/maihv/openssl-
1.1.0l --with-openssl-opt=enable-ec_nistp_64_gcc_128 --with-
openssl-opt=no-nextprotoneg --with-openssl-opt=no-weak-ssl-
ciphers --with-openssl-opt=no-ssl3 --with-
pcre=/home/maihv/pcre-8.40/ --with-pcre-jit --with-
zlib=/home/maihv/zlib-1.2.11/ --with-compat --with-file-aio --
```

```
with-threads --with-http_addition_module --with-
http_auth_request_module --with-http_dav_module --with-
http_flv_module --with-http_gunzip_module --with-
http_gzip_static_module --with-http_mp4_module --with-
http_random_index_module --with-http_realip_module --with-
http_slice_module --with-http_ssl_module --with-http_sub_module
--with-http_stub_status_module --with-http_v2_module --with-
http_secure_link_module --with-mail --with-mail_ssl_module --
with-stream --with-stream_realip_module --with-
stream_ssl_module --with-stream_ssl_preread_module --with-debug
--add-module='/home/maihv/nginx-rtmp-module-master/'
```

```
sudo make
```

```
sudo make install
```

Lưu ý câu lệnh configure bên trên là 1 dòng lệnh.

Sau khi chạy thành công, ta có thể kiểm tra xem nginx đã cài đặt thành công hay chưa bằng lệnh

```
sudo nginx -V
```

```
maihv@maihv:~/nginx-1.14.2$ sudo nginx -V
nginx version: nginx/1.14.2 (Ubuntu)
built by gcc 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)
built with OpenSSL 1.1.0f 10 Sep 2019
TLS SNI support enabled
configure arguments: --prefix=/usr/share/nginx --sbin-path=/usr/sbin/nginx --mod
=/usr/lib/nginx/modules --conf-path=/etc/nginx/nginx.conf --error-log-path=/var/
/error.log --http-log-path=/var/log/nginx/access.log --pid-path=/run/nginx.pid -
h=/var/lock/nginx.lock --user=www-data --group=www-data --build=Ubuntu --http-cl
-temp-path=/var/lib/nginx/body --http-fastcgi-temp-path=/var/lib/nginx/fastcgi -
xy-temp-path=/var/lib/nginx/proxy --http-scgi-temp-path=/var/lib/nginx/scgi --ht
temp-path=/var/lib/nginx/uwsgi --with-openssl=/home/maihv/openssl-1.1.0f --with-
pt=enable-ec nistp_64 gcc 128 --with-openssl-opt=no-nextprotoneg --with-openssl-
ak-ssl-ciphers --with-openssl-opt=no-ssl3 --with-pcre=/home/maihv/pcre-8.40/ --w
jit --with-zlib=/home/maihv/zlib-1.2.11/ --with-compat --with-file-aio --with-th
ith-http_addition_module --with-http_auth_request_module --with-http_dav_module
tp_flv_module --with-http_gunzip_module --with-http_gzip_static_module --with-ht
module --with-http_random_index_module --with-http_realip_module --with-http_slice
--with-http_ssl_module --with-http_sub_module --with-http_stub_status_module --wi
2_module --with-http_secure_link_module --with-mail --with-mail_ssl_module --wit
--with-stream_realip_module --with-stream_ssl_module --with-stream_ssl_preread_m
ith-debug --add-module=/home/maihv/nginx-rtmp-module-master/
maihv@maihv:~/nginx-1.14.2$
```

Sau đó, ta cần tạo file service cho nginx để có thể chạy thành tiến trình.

```
sudo vim /etc/systemd/system/nginx.service
```

Copy phần bên dưới vào file service:

```
[Unit]
Description=A high performance web server and a reverse
proxy server
After=network.target

[Service]
Type=forking
PIDFile=/run/nginx.pid
```

```
ExecStartPre=/usr/sbin/nginx -t -q -g 'daemon on;
master_process on;'
ExecStart=/usr/sbin/nginx -g 'daemon on; master_process on;'
ExecReload=/usr/sbin/nginx -g 'daemon on; master_process
on;' -s reload
ExecStop=-/sbin/start-stop-daemon --quiet --stop --retry
QUIT/5 --pidfile /run/nginx.pid
TimeoutStopSec=5
KillMode=mixed

[Install]
WantedBy=multi-user.target
```

Tiến hành tạo các thư mục sau để cho service được hoạt động bình thường

```
sudo mkdir -p /var/lib/nginx
```

```
sudo mkdir -p /var/lib/nginx/body
```

```
sudo mkdir -p /var/lib/nginx/fastcgi
```

Lưu lại và thoát ra, ta tiến hành start service để kiểm tra tình trạng hoạt động của service:

```
sudo systemctl start nginx
```

Tiếp theo, ta cần tạo config cho nginx trên firewall để mở các port cần thiết

```
sudo vim /etc/ufw/applications.d/nginx
```

Dán đoạn đoạn sau vào file sau đó lưu lại:

```
[Nginx HTTP]
title=Web Server (Nginx, HTTP)
description=Small, but very powerful and efficient web
server
ports=80/tcp

[Nginx HTTPS]
title=Web Server (Nginx, HTTPS)
description=Small, but very powerful and efficient web
server
ports=443/tcp

[Nginx Full]
title=Web Server (Nginx, HTTP + HTTPS)
description=Small, but very powerful and efficient web
server
ports=80,443/tcp
```

Sau khi import được module rtmp vào trong nginx, ta thêm block code của rtmp vào config của nginx là được. Ta gõ lệnh:

```
cd /etc/nginx
```

```
sudo vim nginx.conf
```

Dán đoạn sau vào cuối cùng của file:

```
rtmp {
    server {
        listen 1935;
        application myvideos {
            play /home/maihv/videos/;
        }
    }
}
```

*Lưu ý: **/home/maihv/video/** là thư mục chứa video hiện tại của mình, bạn có thể thay đổi đường dẫn video sao cho phù hợp với cây thư mục của bạn*

Tiến hành restart lại nginx:

```
sudo systemctl restart nginx
```

```
maihv@maihv:/etc/nginx$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/etc/systemd/system/nginx.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-10-05 07:43:15 UTC; 2min 21s ago
     Process: 74709 ExecStartPre=/usr/sbin/nginx -t -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
     Process: 74721 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Main PID: 74725 (nginx)
       Tasks: 2 (limit: 4583)
      Memory: 2.2M
    CGroup: /system.slice/nginx.service
            └─74725 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
               └─74726 nginx: worker process

Oct 05 07:43:15 maihv systemd[1]: Starting A high performance web server and a reverse proxy server...
Oct 05 07:43:15 maihv systemd[1]: Started A high performance web server and a reverse proxy server.
maihv@maihv:/etc/nginx$
```

Tiếp theo, ta cần 1 video trên server để có thể check kết quả. Ở đây, mình sử dụng WinSCP để chuyển file video vào server tại thư mục **/home/maihv/videos**

Sau khi chuyển xong, ta cần convert video về định dạng **.ts** và một playlist dưới dạng file **.m3u8** để có thể phát video qua hls. Ta sử dụng lệnh sau:

```
ffmpeg -i /home/maihv/videos/video.mp4 -profile:v baseline -level 3.0 -s 720x400 -start_number 0 -hls_time 10 -hls_list_size 0 -f hls /home/maihv/videos/hls/video.m3u8
```

```
maihv@maihv:/etc/nginx$ ffmpeg -i /home/maihv/videos/video.mp4 -profile:v baseline -level 3.0 -s 720x400 -start_number 0 -hls_time 10 -hls_list_size 0 -f hls /home/maihv/videos/hls/video.m3u8
ffmpeg version 4.2.4-lubuntu0.1 Copyright (c) 2000-2020 the FFmpeg developers
  built with gcc 9 (Ubuntu 9.3.0-10ubuntu2)
  configuration: --prefix=/usr --extra-version=lubuntu0.1 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x86_64-linux-gnu --arch=amd64 --enable-gpl --disable-strip ping --enable-avresample --disable-filter-resample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libaom --enable-libass --enable-libbluray --enable-libs2b --enable-libcaca --enable-libcdio --enable-libcodecs2 --enable-libflite --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libgme --enable-libgsm --enable-libjack --enable-libmp3lame --enable-libmvs ofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-libsvg --enable-librubberband --enable-libshine --enable-lbsnappy --enable-libsoxr --enable-libspeex --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-lv2 --enable-omx --enable-opengl --enable-openc1 --enable-opengl --enable-sdl2 --enable-libcd1394 --enable-libdrm --enable-libiec61883 --enable-nvenc --enable-chromaprint --enable-frei0r --enable-libx264 --enable-shared
  libavutil 56. 31.100 / 56. 31.100
  libavcodec 58. 54.100 / 58. 54.100
  libavformat 58. 29.100 / 58. 29.100
  libavdevice 58.  8.100 / 58.  8.100
  libavfilter 7. 57.100 / 7. 57.100
  libavresample 4.  0.  0 / 4.  0.  0
  libswscale 5.  5.100 / 5.  5.100
  libswresample 3.  5.100 / 3.  5.100
  libpostproc 55.  5.100 / 55.  5.100
```

*Lưu ý: **/home/maihv/videos/video.mp4** là đường dẫn chứa video gốc của mình và **/home/maihv/videos/hls/video.m3u8** là đường dẫn chứa video đã convert*

Sau khi convert xong, tiến hành tạo file config cho hls, bạn sử dụng lệnh:

```
cd /etc/nginx/  
sudo mkdir conf.d  
cd conf.d/  
sudo vim hls.conf
```

Paste đoạn code sau vào file sau đó lưu lại:

```
server {  
    listen 80;  
    root /var/www/html;  
    server_name videos.maihv.xyz;  
    location /hls {  
        # CORS setup  
        add_header 'Access-Control-Allow-Origin' '*' always;  
        add_header 'Access-Control-Expose-Headers' 'Content-  
Length';  
        # Allow CORS preflight requests  
        if ($request_method = 'OPTIONS') {  
            add_header 'Access-Control-Allow-Origin' '*';  
            add_header 'Access-Control-Max-Age' 1728000;  
            add_header 'Content-Type' 'text/plain  
charset=UTF-8';  
            add_header 'Content-Length' 0;  
            return 204;  
        }  
        types {  
            application/vnd.apple.mpegurl m3u8;  
            video/mp2t ts;  
        }  
        add_header Cache-Control no-cache;  
        alias /home/maihv/videos/hls/;  
    }  
}
```

*Lưu ý, phần **alias** là đường dẫn chứa thư mục video đã convert, bạn cần thay đổi đường dẫn cho phù hợp. Phần **server_name** cũng cần đổi lại cho đúng với domain mà bạn đang sở hữu, nếu như chưa mua domain, bạn có thể nhập tùy ý.*

Bây giờ, ta cần thêm phần config của hls vào config của nginx để tích hợp. Ta gõ lệnh sau:

```
cd /etc/nginx  
sudo vim nginx.conf
```

Paste dòng include vào trong block code của http sau đó lưu lại

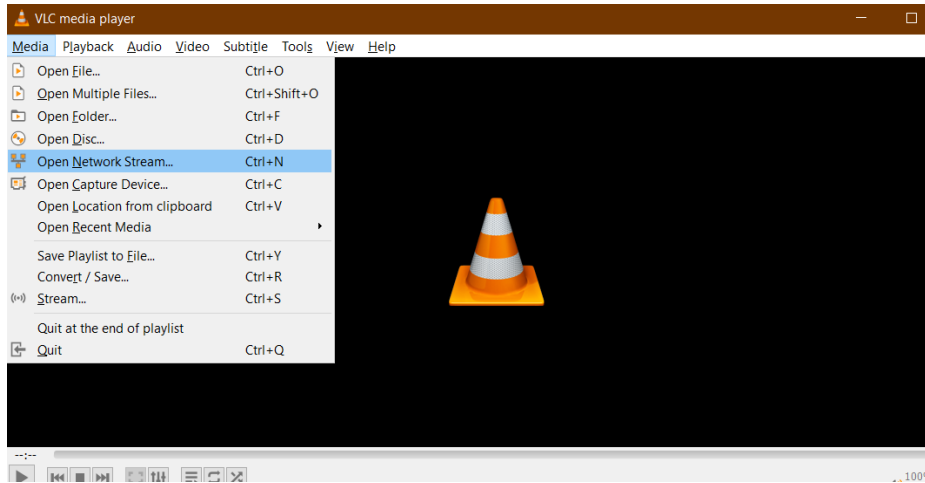
```
http {  
    ...  
    include /etc/nginx/conf.d/*;  
    ...  
}
```

Giờ thì restart lại nginx để xác nhận các thay đổi.

```
sudo systemctl restart nginx
```

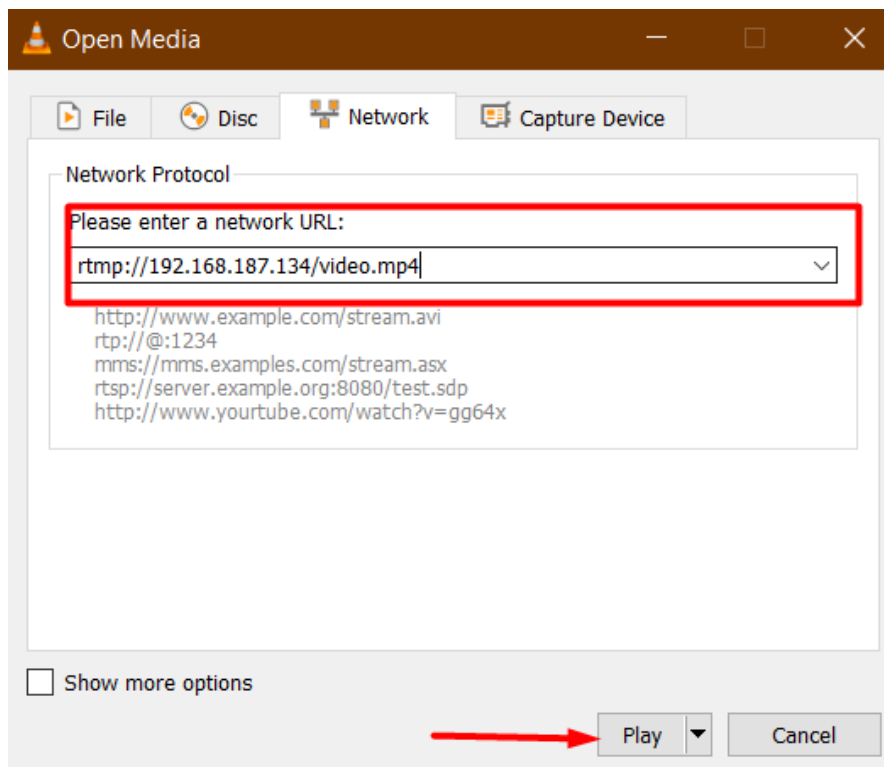
Kiểm tra kết quả

Để kiểm tra tính năng rtmp xem đã được hay chưa, ta sử dụng **vlc media player** để mở video trên server lên. Ta chọn **Media -> Open Network Stream**.



Ở trong mục network, chọn đường dẫn đến thư mục video sau đó nhấn play để chạy thử

```
rtmp://[YOUR_IP or YOUR_DNS]/myvideos/[VIDEO_NAME]
```



Ta được kết quả



Để test tính năng hls, ta cần chuẩn bị một file html để có thể sử dụng được chức năng này. Trên máy khách, ta tạo một file *test.html*. Chèn đoạn mã sau và lưu lại:

```
<!-- CSS -->
<link href="https://vjs.zencdn.net/7.2.3/video-js.css"
rel="stylesheet">

<!-- HTML -->
<video id='hls-example' class="video-js vjs-default-
skin" width="400" height="300" controls>
<source type="application/x-mpegURL"
src="http://192.168.187.134/hls/video.m3u8">
</video>

<!-- JS code -->
<!-- If you'd like to support IE8 (for Video.js versions
prior to v7) -->
<script src="https://vjs.zencdn.net/ie8/ie8-
version/videojs-ie8.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/videojs-
contrib-hls/5.14.1/videojs-contrib-hls.js"></script>
<script
src="https://vjs.zencdn.net/7.2.3/video.js"></script>

<script>
var player = videojs('hls-example');
player.play();
</script>
```


Lưu ý, phần source của video có dạng như sau

```
http://[YOUR_IP or YOUR_DNS]/hls/[YOUR_VIDEO].m3u8
```

Mở file html đó lên, ta được kết quả như hình



Tài liệu tham khảo

- [Streaming videos server sử dụng NGINX RTMP và HLS](#)