

live streaming hls using ffmpeg and nginx

REFERENCE LINK:-

<https://docs.peer5.com/guides/setting-up-hls-live-streaming-server-using-nginx/#options-2-from-local-webcamdifferent-rtmpfile>

TABLE OF CONTENTS

TABLE OF CONTENTS	2
Compile nginx with rtmp module	3
Create nginx configuration file	4
Start nginx	6
Creating Live Streaming	7

Compile nginx with rtmp module

Firstly, we'll need to compile nginx with the nginx-rtmp-module.

We recommend using [this](#) forked module. It's being actively worked on and contains more fixes and improvements over the [original one](#)

Clone nginx-rtmp-module

```
git clone
https://github.com/sergey-dryabzhinsky/nginx-rtmp-module.git
```

Install nginx dependencies

```
sudo apt-get install build-essential libpcre3 libpcre3-dev
libssl-dev
```

Download nginx

```
wget http://nginx.org/download/nginx-1.17.3.tar.gz
tar -xf nginx-1.17.3.tar.gz
cd nginx-1.17.3
```

Compile nginx

```
./configure --with-http_ssl_module
--add-module=../nginx-rtmp-module
make -j 1
sudo make install
```

1. Notice the `--add-module=../nginx-rtmp-module` argument, the path must point correctly to the cloned module
2. (Optional) replace `-j 1` with the amount of cpu's on your computer to accelerate the compilation

Create nginx configuration file

The default location for nginx conf is `/usr/local/nginx/conf/nginx.conf` or `/etc/nginx/nginx.conf`, take the backup of the default one and create the new one with the configuration written below:-

```
worker_processes  auto;
events {
    worker_connections  1024;
}

# RTMP configuration
rtmp {
    server {
        listen 1935; # Listen on standard RTMP port
        chunk_size 4000;

        application show {
            live on;
            # Turn on HLS
            hls on;
            hls_path /mnt/hls/;
            hls_fragment 3;
            hls_playlist_length 60;
            # disable consuming the stream from nginx as rtmp
            deny play all;
        }
    }
}
```

```

}

http {
    sendfile off;
    tcp_nopush on;
    # aio on;
    directio 512;
    default_type application/octet-stream;

    server {
        listen 8080;

        location / {
            # Disable cache
            add_header 'Cache-Control' 'no-cache';

            # 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/dash+xml mpd;
                application/vnd.apple.mpegurl m3u8;
                video/mp2t ts;
            }
        }
    }
}

```

```
    }  
    root /mnt/;  
  }  
}  
}
```

Start nginx

The nginx binary is located wherever you compiled it to -
/usr/local/nginx/sbin/nginx by default. Change it to reflect your path.

Test the configuration file

```
/usr/local/nginx/sbin/nginx -t
```

Start nginx in the background

```
/usr/local/nginx/sbin/nginx
```

Start nginx in the foreground

```
/usr/local/nginx/sbin/nginx -g 'daemon off;'
```

Reload the config on the go

```
/usr/local/nginx/sbin/nginx -t && nginx -s reload
```

Kill nginx

```
/usr/local/nginx/sbin/nginx -s stop
```

Creating Live Streaming

Download a video that you want to stream and scp that video file on the server (ec2 instance) and for initiating the live streaming you need ffmpeg.

Downloading ffmpeg

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

Stream file example-vid.mp4

Suppose you have a video with name example-vid.mp4, so for live streaming using ffmpeg you need to execute the below command:-

```
ffmpeg -re -i example-vid.mp4 -vcodec libx264 -vprofile baseline  
-g 30 -acodec aac -strict -2 -f flv rtmp://localhost/show/stream
```

It will initiate the process for streaming the video live on your terminal, for testing the livestreaming hit the below url on browser or on your video player that you are using, ex:- VLC

```
http://localhost:8080/hls/stream.m3u8
```

Wait for sometime, in this time the video will buffer and it will start playing.

