

# HTTP Live Streaming

*A Practical Guide*

## Streaming HLS with Nginx's RTMP Module

In this post I'm going to show you how to use Nginx's RTMP module to stream a live video (over RTMP) from your desktop and then play it back with HLS. I'll also cover how to encrypt the video.

[RTMP](#) was initially a proprietary protocol by Adobe for streaming video and audio between a Flash player and a server but is still widely used today for broadcasting video despite the demise of Flash.

Note: I'm working on a live streaming solution that runs in your own cloud. Right now, only AWS is supported. A hosted option will also be available. Features will include, among other things, low-latency HLS and encryption. If you're interested, send me a message, and I'll let you know when it's released.

### Installation

I built the RTMP module on Ubuntu 18.04 but the installation instructions should work for any Debian-based Linux distribution.

The first thing to do is install Nginx if you haven't already. You can install it from your distribution's default package repositories but the provided version is usually outdated. A better option is to install it from the official Nginx repository. The current stable version is 1.18.

To install Nginx, run the following command: `$ sudo apt install nginx`

To build the RTMP module, first install the required dependencies then check out the source code of the module. We also need the source code of the version of Nginx that is installed. To build the module, run the following commands:

```
$ sudo apt install build-essential dpkg-dev libpcre3 libpcre3-dev libssl-dev zlib1g zlib1g-dev
$ git clone https://github.com/arut/nginx-rtmp-module.git
$ apt-get source nginx
$ cd nginx-1.18.0/
$ ./configure --with-compat --add-dynamic-module=./nginx-rtmp-module
$ make modules
```

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

This will build the RTMP module as a dynamic module. To load the module at runtime, add the following line to the Nginx configuration file (`/etc/nginx/nginx.conf`):

```
5
6  load_module modules/ngx_rtmp_module.so;
7
```

nginx.conf hosted with ❤ by GitHub

[view raw](#)

Now that Nginx has been installed and the RTMP module has been built, we need to configure it.

## Configuring RTMP

The first thing we need to do is define an RTMP section in the Nginx configuration file:

```
12  rtmp {
13      access_log /var/log/nginx/rtmp.log;
14
15      server {
16          listen 1935;
17
18          application live {
19              live on;
20
21              hls on;
22              hls_path /srv/www;
23              hls_fragment 6s;
24
25              hls_keys on;
26              hls_key_path /srv/www/keys;
27              hls_key_url /keys/;
28              hls_fragments_per_key 10;
29          }
30      }
31  }
```

nginx.conf hosted with ❤ by GitHub

[view raw](#)

The server will listen for connections on port 1935. Within the `server` block, we define an application called `live`. (You can call it whatever you like, I just happened to call it `live`.) We'll refer back to this later on when we start streaming. To enable HLS, set `hls` to `on`. The next two parameters determine the location where the playlist and the segments will be written and the segment duration. In this instance, the playlist and segments are written to the `/srv/www` directory and the segment duration is set to 6 sec-

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

To enable encryption (optional) set `hls_keys` to `on`. The next two values specify the directory where the keys will be written, and the base URL of the keys. Finally, we specify the key rotation period. This determines how many segments are encrypted with a key before a new key is generated. For example, if `hls_fragments_per_key` is set to 10 and the segment duration is 6 seconds, the encryption key will be rotated every minute.

The next step is to configure Nginx to serve the HLS playlist and segments. We do that by defining an `http` block in the configuration file:

```
33 http {
34     include mime.types;
35     default_type application/octet-stream;
36     access_log /var/log/nginx/access.log;
37
38     map $sent_http_content_type $expires {
39         default 1d;
40         application/vnd.apple.mpegurl epoch;
41     }
42
43     server {
44         listen 80;
45
46         location / {
47             root /srv/www;
48             expires $expires;
49         }
50     }
51 }
```

nginx.conf hosted with ❤ by GitHub

[view raw](#)

The root location is set to `/srv/www`, which you'll recall is the same location where the playlist and segments are written to. The `expires` parameter (line 48) specifies how long the playlist and the segments can be cached for, which is determined by looking up the value from a map based on the content type (line 38). In this case, the expiry time of the playlist (`application/vnd.apple.mpegurl`) is set to some time in the past (`epoch`) so it *won't* be cached. Segments can be cached for 24 hours.

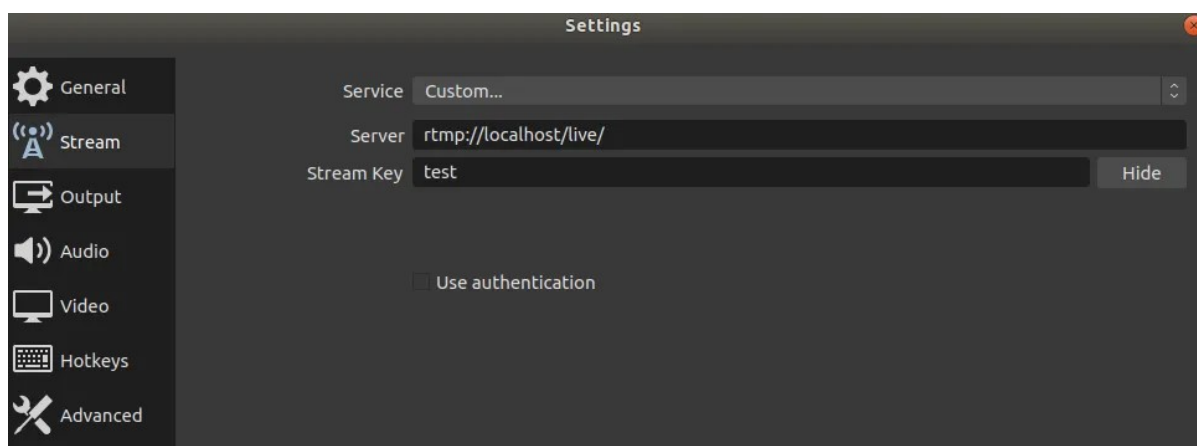
Note: Everything is served over HTTP. If you are encrypting the video then you should serve the keys over HTTPS. This is left as an exercise for the reader.

To start Nginx, run the following: `$ sudo nginx`

If you want to stream the output of a webcam, I suggest downloading and installing [OBS Studio](#). Alternatively, you can use `ffmpeg` to “live” stream a video.

However you decide to set-up your live stream, the format of the RTMP URL is the same. It must be in the following format: `rtmp://<host>/<application name>/<stream key>`

The host is either the IP address or the name of the server. As we are running it locally, it will be `localhost`. The application name *must* match the name in the Nginx configuration file, which in this example is `live`. The stream key can be anything – we’ll use `test` – and is used by the RTMP module to name the playlist and the segments. For example, if the stream key is `test` the playlist will be called `test.m3u8` and the segments will be `test-0.ts`, `test-1.ts`, `test-2.ts`, and so on.



Setting the RTMP URL in OBS

Here’s an example of how to use `ffmpeg` to stream a video over RTMP:

```
$ ffmpeg -re -i big_buck_bunny_720p_h264.mov -c copy -f flv rtmp://localhost/live/test
```

Start the live stream. After a few seconds you should see some files in `/srv/www`.

Next, copy the following HTML and save it as `index.html` in the `/srv/www` directory:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <link href="https://vjs.zencdn.net/7.8.4/video-js.css" rel="stylesheet" />
5 </head>
6 <body>
7   <video
8     id="my-video"
9     class="video-js"
10    controls
11    preload="auto"
```

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it. Ok

```
15     <source src="test.m3u8" type="application/vnd.apple.mpegurl" />
16     <p class="vjs-no-js">
17         To view this video please enable JavaScript, and consider upgrading to a
18         web browser that
19         <a href="https://videojs.com/html5-video-support/" target="_blank">supports HTML5 video
20     </p>
21 </video>
22 <script src="https://vjs.zencdn.net/7.8.4/video.js"></script>
23 </body>
24 </html>
```

index.html hosted with ❤ by GitHub

[view raw](#)

(If you are using a different stream key, you'll need to change the name of the playlist in the HTML.)

To play the live video, open up a browser and navigate to <http://localhost/>.

If you have any questions, feel free to leave a comment.

---

#### Share this:



Facebook



LinkedIn



Twitter

---

#### Like this:

Like

Be the first to like this.

---

#### Related

[Playing HLS Video in the Browser](#)

February 23, 2016

In "Uncategorised"

[Playing HLS Video in the Browser - Part 2](#)

December 14, 2017

Similar post

[Segmenting Video with ffmpeg - Part 2](#)

August 20, 2016

In "ffmpeg"

This entry was posted in Uncategorized on September 17, 2020 [<https://hlsbook.net/hls-nginx-rtmp-module/>].

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.