




DZone (/) > Web Dev Zone (/web-development-programming-tutorials-tools-news) > HLS Streaming of RTSP Stream by Nginx and Apache Tomcat

# HLS Streaming of RTSP Stream by Nginx and Apache Tomcat



by **sagar patel** (/users/3424034/sagarpatel-1.html) · Sep. 28, 18 · Web Dev Zone

(/web-development-programming-tutorials-tools-news) · Tutorial

 Like (5)  Comment (2)  Save  Tweet

Hello coders!

In my previous article (<https://dzone.com/articles/hls-streaming-protocol>), I tried to explain the basics of HLS.

Today, I am writing this article on how to configure Nginx (<https://www.nginx.com/>) for RTSP ([https://en.wikipedia.org/wiki/Real\\_Time\\_Streaming\\_Protocol](https://en.wikipedia.org/wiki/Real_Time_Streaming_Protocol)) to HLS (<https://dzone.com/articles/hls-streaming-protocol>) streams and both Apache 2 and Apache Tomcat for HLS streams.

We have many ways to convert RTSP streams to HLS streams. I have mentioned two ways to convert RTSP to HLS using the Nginx server and Apache server. We will discuss both ways, one-by-one. These two ways are easy and less time-consuming than setting up an environment to convert RTSP to HLS. There are just a few small steps required to configure Nginx and an Apache server.

## 1. Setting Up an Nginx Server for HLS Live Stream From an RTSP Stream

First of all, Nginx can't enable us to stream from RTSP to direct HLS. First, we need to convert RTSP to RTMP. RTMP streams can be converted into HLS by Nginx.

Required steps:

1. RTSP to RTMP (configure nginx.conf for RTSP to RTMP conversion by ffmpeg).



3. RTMP stream becomes an HLS stream (same name as we used in the pull request).

4. HLS streams from RTSP are accessible in the browser or you can use it in a flash or HTML player.

5. Required ffmpeg binary.

**NOTE:** Requires a pull client (like VLC or any other custom player) for RTMP stream.

### Example nginx.conf file

Find the RTMP tag in the file and configure that tag as shown below:

```
1 rtmp {
2     server {
3         listen 1935;
4         chunk_size 4096;
5
6         application live {
7             live on;
8             record off;
9             hls on;
10            hls_nested on;
11            hls_path /usr/local/nginx/html/hls/;
12            hls_fragment 3;
13            hls_playlist_length 60;
14
15            exec_pull /usr/bin/ffmpeg -i rtsp://10.103.0.77:8050/$name -vcodec copy -acodec copy -f
16        }
17    }
18 }
```

### Here:

- The `ffmpeg` command is used to convert RTSP streams (for example, the local RTSP stream from a camera) to RTMP. The client is required to pull the RTMP stream.
- **hls\_path:** The path where you want to store the HLS file (m3u8 file).
- **listen 1935:** Listen to this port for the client. For example, you can pull an RTMP



stream from VLC by using this port. This is configurable, so we can assign a different port. (1)



(/users/login.html)



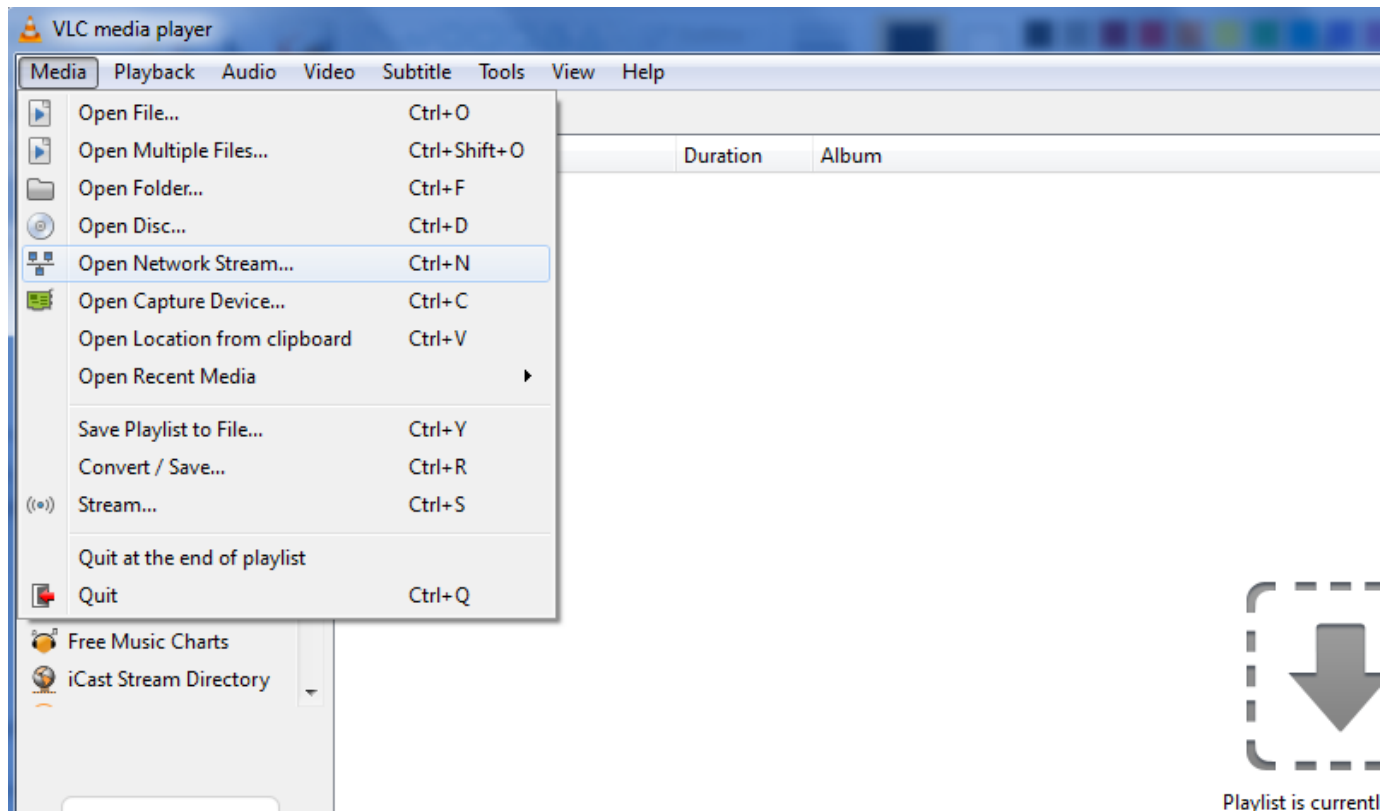
(/search)

**REFCARDZ** (/refcardz) **TREND REPORTS** (/trendreports) **WEBINARS** (/webinars) **ZONES** ▾

- **hls\_fragment**: Each .ts file has a 3-second duration to update the m3u8 file for HLS.

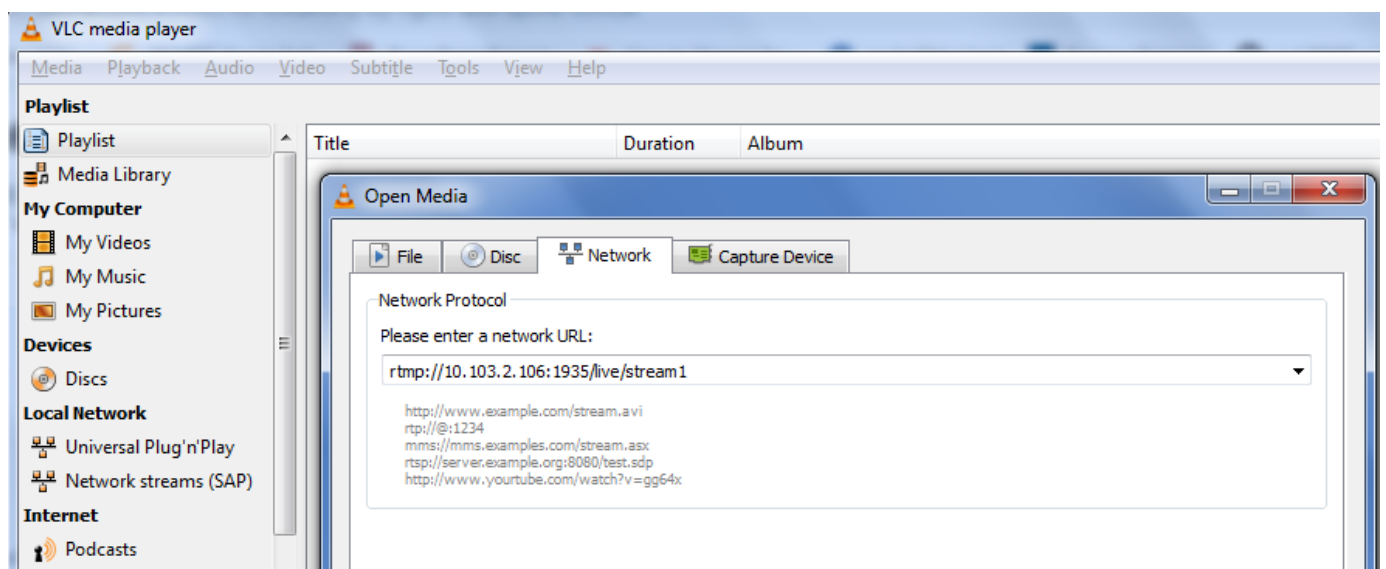
Now, we can pull RTMP streams from the Nginx server via port 1935 (listening port).

The below image is the pull stream that streams data from VLC.



Playlist is currentl

Select the network stream and enter the stream as shown below. The stream URL will be the same as that given in the configuration file.



## Configure a Dynamic Name for the RTMP Stream

Here, I have used `$name` for the dynamic name of the RTMP stream.

Now, you can see the RTMP stream in the VLC player; thus we can see the RTMP stream. The Nginx server starts to generate a m3u8 file on the given `hls_path`.

## To See the HLS Stream in the Browser

Now, in the browser, we can see an HLS stream at **url: `http://server_ip/stream1`**.

Here, `server_ip` is the Nginx server IP where Nginx is installed.

## Limitations

The client is required to pull RTMP continuously, so once the client stops pulling, the HLS stream goes offline.

## 2. Setting Up an Apache 2 Server for HLS Live Stream From RTSP Stream

For Apache 2, only one step is required to set the path where your m3u8 file is generated. You can see an HLS stream via the server IP and the Apache port (given in the `httpd.conf` file).

In the browser, URL will look like

this: **`http://server_ip:port/file_location/file.m3u8`**.

### Example `httpd.conf` file

```
1 #
2 # Listen: Allows you to bind Apache to specific IP addresses and/or
3 # ports, instead of the default. See also the <VirtualHost>
4 # directive.
5 #
6 # Change this to Listen on specific IP addresses as shown below to
7 # prevent Apache from glomming onto all bound IP addresses.
8 #
9 #Listen 12.34.56.78:80
10 Listen *:80
11
12 #
13 # DocumentRoot: The directory out of which you will serve your
14 # documents. By default, all requests are taken from this directory, but
15 # symbolic links and aliases may be used to point to other locations.
```



Here DocumentRoot will be your m3u8 file location.

For Linux, the same configuration will work. It's all about Apache 2 configuration.

## 2. Setting Up Apache Tomcat Server for HLS Live Stream From RTSP Stream.

In Tomcat 8, we are required to set the HLS path and folder location where m3u8 files will be stored.

The path will be like this: **/tomcat8/webapps/ROOT/hls**

Upload the HLS file in this location, and apply the filter shown below in the server.xml file. Tomcat will automatically recognize the .m3u8 file folder in the ROOT directory.

In the browser, you can see the HLS stream via this URL:


**http://tomcat\_serverIp:port/hls/filename.m3u8.**

It's required to allow Apache Tomcat servers to serve cross-browser requests, so I have to use the below filter for HLS configurations called cross-browser filters.

```

1 <filter>
2   <filter-name>CorsFilter</filter-name>
3   <filter-class>org.apache.catalina.filters.CorsFilter</filter-class>
4   <init-param>
5     <param-name>cors.allowed.origins</param-name>
6     <param-value>*</param-value>
7   </init-param>
8   <init-param>
9     <param-name>cors.allowed.methods</param-name>
10    <param-value>GET,POST,HEAD,OPTIONS,PUT</param-value>
11  </init-param>
12  <init-param>
13    <param-name>cors.allowed.headers</param-name>
14    <param-value>Content-Type,X-Requested-With,accept,Origin,Access-Control-Request-Method,
15  </init-param>
16  <init-param>
17    <param-name>cors.exposed.headers</param-name>
18    <param-value>Access-Control-Allow-Origin,Access-Control-Allow-Credentials</param-value>
19  </init-param>
20  <init-param>
21    <param-name>cors.support.credentials</param-name>
22    <param-value>>false</param-value>
23  </init-param>
24  <init-param>
25    <param-name>cors.preflight.maxage</param-name>
26    <param-value>10</param-value>

```



27 </init-param>

28 </filter> (/)

29 </filter-mapping>

30 <filter-name>CorsFilter</filter-name>

31 <url-pattern>/\*</url-pattern>


32 </filter-mapping>


REFCARDZ (/refcardz)

TREND REPORTS (/trendreports)

WEBINARS (/webinars)

ZONES ▾

 (/users/login.html)

 (/search)

I hope I have covered all the information related to configuration you were hoping to learn. If you have any questions, please feel free to ask.

**Previous article:** <https://dzone.com/articles/hls-streaming-protocol>  
(<https://dzone.com/articles/hls-streaming-protocol>)

Thanks!

Stream (Computing)   Apache Tomcat

Opinions expressed by DZone contributors are their own.

## Popular on DZone

- [Kubernetes Monitoring With Prometheus \(/articles/kubernetes-monitoring-with-prometheus?fromrel=true\)](/articles/kubernetes-monitoring-with-prometheus?fromrel=true)
- [Why Your Database Needs a Machine Learning Brain \(/articles/why-your-database-needs-a-machine-learning-brain?fromrel=true\)](/articles/why-your-database-needs-a-machine-learning-brain?fromrel=true)
- [The Top Security Strategies in Custom Software Development \(/articles/the-top-security-strategies-in-custom-software-dev-2?fromrel=true\)](/articles/the-top-security-strategies-in-custom-software-dev-2?fromrel=true)
- [How a Low-Code API Platform Delivers Developer Productivity \(/articles/how-a-low-code-api-platform-delivers-developer-productivity?fromrel=true\)](/articles/how-a-low-code-api-platform-delivers-developer-productivity?fromrel=true)

### ABOUT US

[About DZone \(/pages/about\)](/pages/about)

[Send feedback \(mailto:support@dzone.com\)](mailto:support@dzone.com)

[Careers \(https://careers.dzone.com/\)](https://careers.dzone.com/)

[Sitemap \(/sitemap\)](/sitemap)

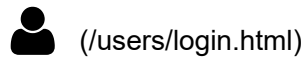
### ADVERTISE

[Advertise with DZone \(https://advertise.dzone.com\)](https://advertise.dzone.com)

### CONTRIBUTE ON DZONE

[Article Submission Guidelines \(/articles/dzones-article-submission-guidelines\)](/articles/dzones-article-submission-guidelines)

[MVB Program \(/pages/mvb\)](/pages/mvb)



**REGARDZ** (/refcardz) **TREND REPORTS** (/trendreports) **WEBINARS** (/webinars) **ZONES** ▾

**Terms of Service** (/pages/tos)

**Privacy Policy** (/pages/privacy)

## CONTACT US

600 Park Offices Drive

Suite 300

Durham, NC 27709

[support@dzzone.com](mailto:support@dzzone.com) (mailto:support@dzzone.com)

+1 (919) 678-0300 (tel:+19196780300)

Let's be friends:    

(/pages/subscribe) (<https://www.dzzone.com/dzzone-company/dzzone/>)

DZone.com is powered by



(<https://devada.com/answerhub/>)