**How to make an RTMP Streaming Server and Player with a Raspberry Pi**

[How to make an RTMP Streaming Server and Player with a Raspberry Pi • Aaron Parecki](https://aaronparecki.com/2020/09/07/7/raspberry-pi-streaming-server)

1. Install Raspberry Pi OS lite (32-bit). The full version works too
2. Config OS as needed (e.g., turn off wifi): sudo raspi-config
3. Next we need to configure the video mode so we know what kind of signal the Raspberry Pi will be sending on the HDMI port.

sudo vim /boot/config.txt. add the following if not existing

# Make sure the image fits the whole screen

disable\_overscan=1

# Set HDMI output mode to Consumer Electronics Association mode

hdmi\_group=1

# Enable audio over HDMI

hdmi\_drive=2

# Set the output resolution and frame rate to your desired option

# 1920x1080 60fps

hdmi\_mode=16

# 1920x1080 25fps

hdmi\_mode=33

# 1920x1080 30fps

hdmi\_mode=34

1. reboot to make the changes take effect. Sudo reboot
2. We'll be using nginx with the RTMP module as the RTMP server, and then connect omxplayer to play out the stream over the HDMI port

sudo apt update

sudo apt install omxplayer nginx libnginx-mod-rtmp

1. give nginx permission to use the video port

sudo usermod -aG video www-data

1. Add rtmp server in nginx.conf

sudo vim /etc/nginx/nginx.conf

Add the following to the very bottom of the file:

rtmp {

server {

listen 1935;

application live {

# Enable livestreaming

live on;

# Disable recording

record off;

# Allow only this machine to play back the stream

allow play 127.0.0.1;

deny play all;

# Start omxplayer and play the stream out over HDMI

exec omxplayer -o hdmi rtmp://127.0.0.1:1935/live/$name;

}

}

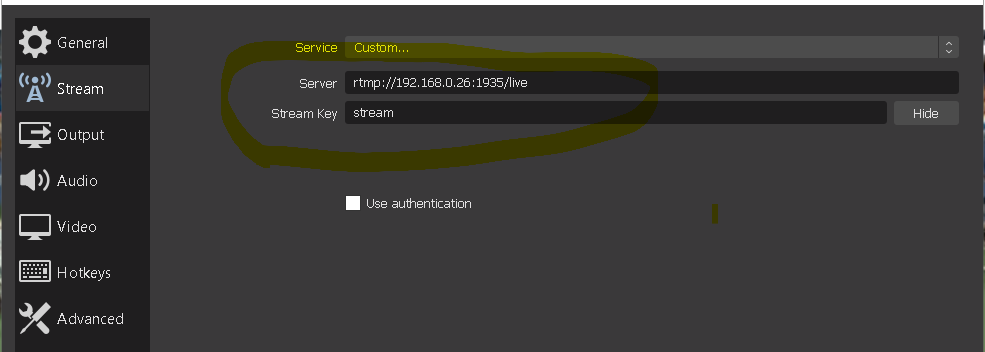
}

The magic sauce here is the exec line that starts omxplayer. omxplayer is an application that can play an RTMP stream out over the Raspberry Pi's HDMI port. The exec line will run this command whenever a new RTMP stream is received. The stream key will be set to the $name variable. Note that this means any stream key will work, there is no access control the way we have it configured here. You can read up on the [RTMP module](https://github.com/arut/nginx-rtmp-module/wiki/Directives) if you'd like to learn how to lock down access to only specific stream keys or if you want to enable recording the stream.

1. Sudo nginx -t to test if there are configuration errors
2. Sudo nginx -s reload
3. Use hostname -I to check the streaming url:

use the RTMP URL: rtmp://YOUR\_IP\_ADDRESS/live and anything as the stream key

1. Start OBS studio, and set streaming server to: rtmp://192.168.0.26:1935/live



Full file content:

**/etc/nginx/nginx.conf:**

user www-data;

worker\_processes auto;

pid /run/nginx.pid;

include /etc/nginx/modules-enabled/\*.conf;

events {

worker\_connections 768;

# multi\_accept on;

}

http {

##

# Basic Settings

##

sendfile on;

tcp\_nopush on;

tcp\_nodelay on;

keepalive\_timeout 65;

types\_hash\_max\_size 2048;

# server\_tokens off;

# server\_names\_hash\_bucket\_size 64;

# server\_name\_in\_redirect off;

include /etc/nginx/mime.types;

default\_type application/octet-stream;

##

# SSL Settings

##

ssl\_protocols TLSv1 TLSv1.1 TLSv1.2; # Dropping SSLv3, ref: POODLE

ssl\_prefer\_server\_ciphers on;

##

# Logging Settings

##

access\_log /var/log/nginx/access.log;

error\_log /var/log/nginx/error.log;

##

# Gzip Settings

##

gzip on;

# gzip\_vary on;

# gzip\_proxied any;

# gzip\_comp\_level 6;

# gzip\_buffers 16 8k;

# gzip\_http\_version 1.1;

# gzip\_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;

##

# Virtual Host Configs

##

include /etc/nginx/conf.d/\*.conf;

include /etc/nginx/sites-enabled/\*;

}

rtmp {

server {

listen 1935;

chunk\_size 4096;

application live {

# Enable live streaming

live on;

interleave on;

# Disable recording

record off;

hls on;

hls\_path /var/www/winn-live/hls;

hls\_fragment 3;

dash on;

dash\_path /var/www/winn-live/dash;

dash\_fragment 15s;

# Allow only this machine to playback the stream

allow play 127.0.0.1;

deny play all;

# Start omxplayer and play the stream out over the HDMI

exec omxplayer -o hdmi rtmp://127.0.0.1/live/$name;

}

}

}

**/etc/nginx/sites-enabled/winn-live-port81.conf:**

# For winn-live-stream

server {

listen 81;

listen [::]:81;

server\_name winn-live.com;

location /hls {

# Disable cache

add\_header Cache-Control non-cache;

# CORS setup

add\_header 'Access-Control-Allow-Origin' '\*' always;

add\_header 'Access-Control-Expose-Headers' 'Content-Length';

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;

application/dash+xml mpd;

text/html html;

}

root /var/www/winn-live;

}

root /var/www/winn-live;

index index.html;

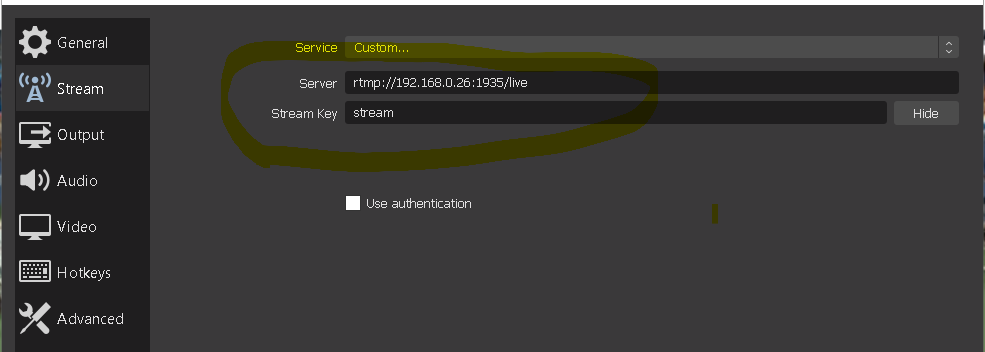
location / {

try\_files $uri $uri/ =404;

}

}

## Creating the following folders for HLS and DASH.



## Content of the index.html file:

<!DOCTYPE html>

<html lang="en">

<head>

<title>Winn Live Stream</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- CSS -->

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

<style>

\* {

box-sizing: border-box;

font-family: Arial, Helvetica, sans-serif;

}

body {

margin: 0;

font-family: Arial, Helvetica, sans-serif;

}

/\* Style the top navigation bar \*/

.topnav {

overflow: hidden;

background-color: #333;

}

/\* Style the topnav links \*/

.topnav a {

float: left;

display: block;

color: #f2f2f2;

font-size: 30px;

font-weight: bold;

text-align: center;

padding: 14px 16px;

text-decoration: none;

}

/\* Change color on hover \*/

.topnav a:hover {

background-color: #ddd;

color: black;

}

/\* Style the content \*/

.content {

background-color: #ddd;

padding: 10px;

}

/\* Style the footer \*/

.footer {

background-color: #f1f1f1;

padding: 10px;

}

.flex-container {

display: flex;

flex-wrap: wrap;

flex-direction: row;

}

.flex-container > div {

background-color: DodgerBlue;

/\* background-color: #f1f1f1; \*/

margin: 10px;

padding: 10px;

font-size: 15px;

flex: 40%;

}

/\* Responsive layout - makes a one column layout instead of a two-column layout \*/

@media (max-width: 800px) {

.flex-item-right, .flex-item-left {

flex: 10%;

}

}

</style>

</head>

<body>

<div class="topnav">

<a href="#">Winter Winn Live Stream</a>

</div>

<div class="content">

<div class="flex-container">

<div>

<p>Winn Live Strem #1</p>

<video id='winn-live1' class="video-js vjs-default-skin" width="500" height="400" controls>

<source type="application/x-mpegURL" src="/hls/stream.m3u8">

</video>

<!--

<a href="index2.html">Watch Live #2</a>

-->

</div>

<div>

<h5>Winn Live Strem #2</h5>

<video id='winn-live2' class="video-js vjs-default-skin" width="500" height="400" controls>

<source type="application/x-mpegURL" src="/hls2/stream2.m3u8">

</video>

</div>

<div>

<h5>Winn Live Strem #3</h5>

<video id='winn-live3' class="video-js vjs-default-skin" width="500" height="400" controls>

<source type="application/x-mpegURL" src="/hls3/stream3.m3u8">

</video>

</div>

<div>

<h5>Winn Live Strem #4</h5>

<video id='winn-live4' class="video-js vjs-default-skin" width="500" height="400" controls>

<source type="application/x-mpegURL" src="/hls4/stream4.m3u8">

</video>

</div>

</div>

</div>

<div class="footer">

<p>Winter Winn Live Stream 24/7 support line: 778-winter1</p>

</div>

<!-- 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 player1 = videojs('winn-live1');

player1.play();

var player2 = videojs('winn-live2');

player2.play();

var player3 = videojs('winn-live3');

player3.play();

var player4 = videojs('winn-live4');

player4.play();

</script>

</body>

</html>