

Copy of: <https://obsproject.com/forum/resources/how-to-set-up-your-own-private-rtmp-server-using-nginx.50/>

Most people who stream enjoy using services such as Twitch.tv or Ustream to deliver video to viewers, and that works well enough. But sometimes you want some more control over your stream, or you want other people to be able to stream to you, or you want to stream to multiple places, or any number of things that requires you to have access to an actual RTMP stream from an RTMP server. This guide will cover the very basics of setting up a simple RTMP server on a Linux computer. Don't worry, it's not too complicated, but having familiarity with Linux will certainly help.

A couple things you can do with your own RTMP server that you might be interested in:

- Stream to multiple external channels
- Import other people's streams to incorporate for your own purposes (I use it in my casts for multiple camera angles, like in [this video I did of a Tribes Ascend broadcast](#) with multiple camera angles)

Alright, so how do you do these kinds of things?

Step 1: Get a Server Box

Believe it or not, RTMP is actually extremely light on system resources. Essentially it just grabs data from the input and forwards it on to the output, simple data transfer. Don't believe me? My RTMP server for a long time was a [Raspberry Pi](#), a \$35 mini-computer, sitting under my desk, and it was capable of hosting *at least* 3 simultaneous streams, and I never even stressed it to see how many more it could handle. So I assure you, even a cheap old box would suffice.

If you don't have your own box, a VPS can also work. I recommend Linode or Digital Ocean as providers. Just make sure you have enough bandwidth...remember that bandwidth usage will be (the size of a stream) * (the number of people uploading + the number of people downloading). So when I have 2 streamers stream to my server, and I download both of them, I can chew up 10GB of bandwidth in 2 hours.

I recommend using Ubuntu for the server software for the sake of ease, but you can obviously use whatever you want. As long as you get the dependencies for nginx somewhere besides apt, you can follow this guide just fine.

Note to Windows users: This guide focuses on using Linux. If you want to use Windows, you can find Windows binaries for nginx with the RTMP module already included here: <http://nginx-win.ecsds.eu/download/>

Note to Mac users: You can install nginx with the RTMP module via Homebrew:

<http://brew.sh/homebrew-nginx/>

If you are hosting your server in your home, you will have to forward TCP port 1935 to the box...this varies by router, so look up how to set up port forwarding for your router. Also, I recommend using a dynamic DNS service to overcome dynamic IP issues that come up with residential hosting.

Step 2: Installing nginx with RTMP module

Log into your box, and make sure you have the necessary tools to build nginx using the following command:

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

Now a bit of info about [nginx](#) (pronounced "engine-X"). nginx is an extremely lightweight web server, but someone wrote a [RTMP module](#) for it, so it can host RTMP streams too. However, to add the RTMP module, we have to compile nginx from source rather than use the apt package. Don't worry, it's really easy. Just follow these instructions. :)

From your home directory, download the nginx source code:

```
$ wget http://nginx.org/download/nginx-1.13.1.tar.gz
```

As of this writing, the latest stable version of nginx is 1.13.1. You can find the latest version on the [nginx download page](#).

Next, get the RTMP module source code from git:

```
$ wget https://github.com/arut/nginx-rtmp-module/archive/master.zip
```

Unpack/unzip them both, and enter the nginx directory:

```
$ tar -zxvf nginx-1.13.1.tar.gz
```

```
$ unzip master.zip
```

```
$ cd nginx-1.13.1
```

Now we build nginx:

```
$ ./configure --with-http_ssl_module --add-module=../nginx-rtmp-module-master
```

```
$ make
```

```
$ sudo make install
```

And nginx is installed! By default it installs to /usr/local/nginx, so to start the server run the following command:

```
$ sudo /usr/local/nginx/sbin/nginx
```

And to test to make sure nginx is running, point your browser to <http://<your server ip>> and you should get the "Welcome to nginx!" page.

Step 3: Configuring nginx to use RTMP

Open your config file, located by default at `/usr/local/nginx/conf/nginx.conf` and add the following at the very end of the file:

Code

```
rtmp {
    server {
        listen 1935;
        chunk_size 4096;

        application live {
            live on;
            record off;
        }
    }
}
```

This is an extremely basic configuration with a "live" application that simply forwards the RTMP stream on to whoever requests it. You can play with it some more later. [Here's the whole configuration guide](#), which shows you how to forward streams to other places (such as Twitch), save recordings of uploads, output stats, etc.

Restart nginx with:

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