

README

Table of Contents

1. [Setup Guide](#)
 - [A Few Words](#)
 - [Project Information](#)
2. [Setting Your NGINX Server](#)
 - [Installing and Configuring Nginx-RTMP](#)
 - [Installing](#)
 - [Configuring](#)
 - [Docker](#)
 - [Render](#)
 - [Running](#)
3. [Setting Your Java Spring Boot Server](#)
 - [If You Are Pulling This Whole Project to Use](#)
 - [If You Are Building From Scratch](#)
 - [Packages Used](#)
 - [Documentation](#)
4. [Links](#)

SETUP guide

A few words

- **Hi there**, this is a small project to help you get familiar with setting **NGINX** to make it a rtmp livestreaming server and code to run it in the backend of your website
- You can use this as a reference

Project information

- **Java Spring Boot** for backend server
- **NGINX** on **Ubuntu** (I'm using Ubuntu 22.04 for this project) or any Unix-base operating system that support NGINX

| Author: **KayGV**

Setting your NGINX server

- Here i'm following a guide from **DigitalOcean** (Source: [DigitalOcean Guide](#)) with some tweaks for rtmp streaming

Installing and Configuring Nginx-RTMP

Installing

- Most of the time Nginx RTMP modules does not come along with Nginx but from Ubuntu 22.04 you can install it as a additional package
- In this project I'm building my Nginx from its source code with additional dependencies (RTMP)

```
sudo apt update
sudo apt install build-essential libpcre3 libpcre3-dev zlib1g zlib1g-dev
libssl-dev libgeoip-dev libxslt1-dev libgd-dev libperl-dev libaio-dev
libxml2-dev libexpat1-dev libmailutils-dev
wget http://nginx.org/download/nginx-1.27.3.tar.gz
tar -zxvf nginx-1.27.3.tar.gz
cd nginx-1.27.3
git clone https://github.com/arut/nginx-rtmp-module.git
./configure --add-module=./nginx-rtmp-module --with-cc-opt='-g -O2 -fno-
omit-frame-pointer -mno-omit-leaf-frame-pointer -ffile-prefix-
map=/build/nginx-D1MnQR/nginx-1.24.0. -flto auto -ffat-lto-objects -fstack-
protector-strong -fstack-clash-protection -Wformat -Werror-format-security -
fcf-protection -fdebug-prefix-map=/build/nginx-D1MnQR/ngin x-
1.24.0=/usr/src/nginx-1.24.0-2ubuntu7.1 -fPIC -Wdate-time -
D_FORTIFY_SOURCE=3' --with-ld-opt='-Wl,-Bsymbolic-functions -flto-auto -
ffat-lto-objects -Wl,-z, rel ro -Wl,-z, now -fPIC' --prefix=/usr/share/nginx
--conf-path=/etc/nginx/nginx.conf --http-log-path=/var/log/nginx/access.log
--error-log-path=stderr --lock-path=/var/lock/nginx.lock --pid-
path=/run/nginx.pid --modules-path=/usr/lib/nginx/modules --http-client-
body-temp-path=/var/lib/nginx/body --http-fastcgi-temp-
path=/var/lib/nginx/fastcgi --http-proxy-temp-path=/var/lib/nginx/proxy --
http-scgi-temp-path=/var/lib/nginx/scgi --http-uwsgi-temp-
path=/var/lib/nginx/uwsgi --with-compat --with-debug--with-pcre-jit --with-
http_ssl_module --with-http_stub_status_module --with-http_realip_module --
with-http_auth_request_module --with-http_v2_module --with-http_dav_module -
-with-http_slice_module --with-threads --with-http_addition_module --with-
http_flv_module --with-http_gunzip_module --with-http_gzip_static_module --
with-http_mp4_module --with-http_random_index_module --with-
http_secure_link_module --with-http_sub_module --with-mail_ssl_module --
with-stream_ssl_module --with-stream_ssl_preread_module --with-
stream_realip_module --with-http_geoip_module=dynamic --with-
http_image_filter_module=dynamic --with-http_perl_module=dynamic --with-
http_xslt_module=dynamic --with-mail=dynamic --with-stream=dynamic --with-
stream_geoip_module=dynamic --prefix=/etc/nginx
sudo make
```

```
sudo make install
sudo cp ./nginx-rtmp-module/stat.xsl /etc/nginx/html
```

Configuring

- After you have downloaded the package use the command below to start configuring Nginx to run RTMP server

```
# This will open the editor for you to edit the configuration file
# You can use vim or nvim for better experience
sudo nano /etc/nginx/nginx.conf
```

- Add this to the end of the file

```
# /etc/nginx/nginx.conf
. . .
rtmp {
    server {
        listen 1935;
        chunk_size 4096;

        # allow publish 127.0.0.1;
        # deny publish all;

        # Not recommended but you can do this in order to enable
multiple user
        # to stream at the same time
        allow publish all;

        application live {
            live on;
            record off;

            # HLS configuration
            hls on;
            hls_path /tmp/hls; # Path where HLS fragments are
stored
            hls_fragment 5s;
            hls_playlist_length 60;

        }
    }
}
. . .
http {
    server {
        <!-- Can use port 80 or 443 to use http/https -->
```

```

listen 8088;

location /hls {
    types {
        application/vnd.apple.mpegurl m3u8;
        video/mp2t ts;
    }
    alias /tmp/hls; # The path where HLS fragments are
stored

    add_header Cache-Control no-cache;
    add_header 'Access-Control-Allow-Origin' '*';
    add_header 'Access-Control-Allow-Methods' 'GET,
POST, OPTIONS, HEAD';
    add_header 'Access-Control-Allow-Headers'
'Authorization, Origin, X-Requested-With, Content-Type, Accept';
}

location /status {
    rtmp_stat all;
    rtmp_stat_stylesheet /stat.xsl;
    add_header 'Access-Control-Allow-Origin' '*';
    add_header 'Access-Control-Allow-Methods' 'GET,
POST, OPTIONS, HEAD';
    add_header 'Access-Control-Allow-Headers'
'Authorization, Origin, X-Requested-With, Content-Type, Accept';
}

location /stat.xsl {
    root /etc/nginx/html;
}

}
. . .
}

```

- `listen 1935` means that RTMP will be listening for connections on port 1935 (standard)
- `chunk_size 4096` means that RTMP will be sending data in 4KB blocks (standard)
- `allow publish 127.0.0.1` and `deny publish all` mean that the server will only allow video to be published from the same server, to avoid any other users pushing their own streams.
 - Alternatively you can add `allow publish all` to make everybody have the ability to access the rtmp server
- `application live` defines an application block that will be available at the /live URL path.
- `live on` enables live mode so that multiple users can connect to your stream concurrently, a baseline assumption of video streaming.

- `record off` disables Nginx-RTMP's recording functionality, so that all streams are not separately saved to disk by default
- If you are using Ngrok for simple use case run

```
# ngrok http --hostname=<static url provided by Ngrok> 80 --scheme http
ngrok http --hostname=marmoset-unbiased-logically.ngrok-free.app 80 --scheme
http,https
```

- Otherwise you can use Docker and push it to render to host

Docker

```
docker build -t csbu_software_design_2024 .
# docker build -t <Image name> .
docker login
docker tag csbu_software_design_2024 kaygv/csbu_software_design_2024:latest
# docker tag <Image name> <DockerHub Username>/<Image name>:<tag>
```

Render

- You can use Render.com to deploy your application.
- Create a new Webservice and choose deploy from DockerHub.
- Select the Docker image and tag that you created earlier.
- Voila

Running

- By default, it listens on port `1935`, which means you'll need to open that port in your firewall. If you configured `ufw` as part of your initial server setup run the following command

```
sudo ufw allow 1935
sudo ufw allow 8088
sudo ufw allow 80
```

- Check the Nginx config file syntax

```
sudo nginx -t
```

- Reload Nginx with changes

```
sudo systemctl start nginx.service
sudo systemctl enable nginx.service
```

```
sudo systemctl status nginx.service # to check status
```

Setting your JAVA SPRING BOOT server

If you are pulling this whole project to use

- The only thing you need to change is the application.properties in the resource folder
 - Change the streamserver.ip to your rtmp server ip
 - Change the databaase to yourown database link

If you are building from scratch

Packages used

- Spring Web
- Postgresql
- Jackson-core
- JWT
- JPA
- Websocket

Document

- [Java doc](#)

Backend for a live streaming platform deployed on AWS EC2 and Render, equipped with Nginx RTMP stream, live chat using websocket; account creation, modification, and deletion; secure user data using hash, light defense against XSS and SQL injection using patter recognition.

Technologies:

Java Spring Boot

Nginx with RTMP

PostgreSQL

Docker

- Links:
 - Source code: [Github source code](#)
 - Documentation: [Java Documentaion](#)