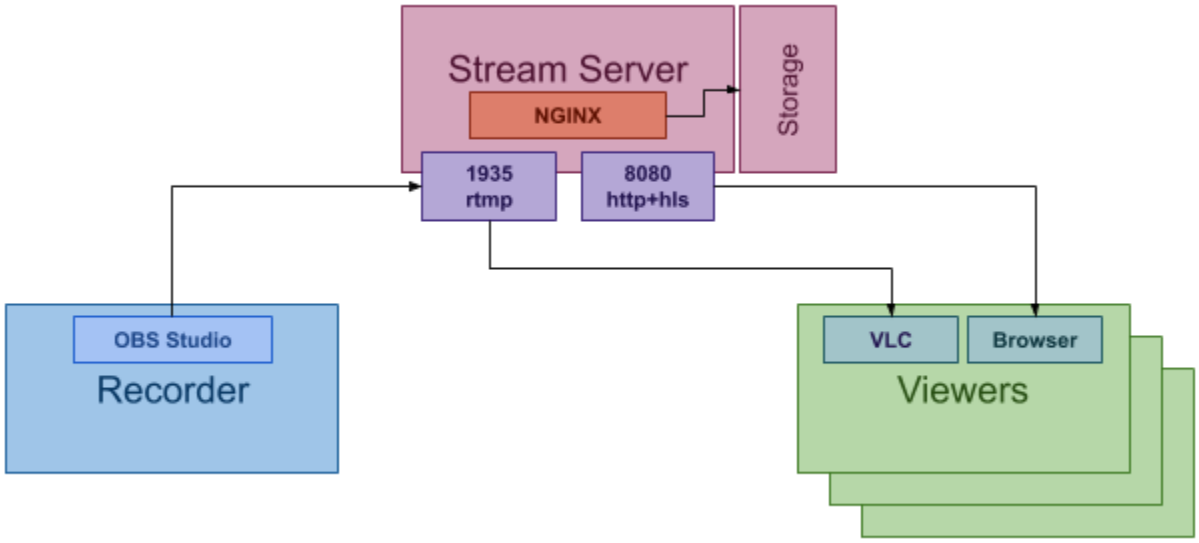
NGINX

Streaming Server

معماری سرور استریم ویدئو:



در این معماری سه عنصر اصلی وجود داره:

- ضبط کننده، که اطلاعات ویدئویی را ضبط کرده و برای سرور استریم ارسال می‌کند
- تماشاگران، که از طریق مرورگر یا یک مدیاپلیر استریم را تماشا می‌کنند
- سرور استریم که وظیفه دریافت استریم از ضبط کننده و در اختیار قرار دادن آن به تماشاگران است. همچنین وظیفه نگهداری تمام استریم را هم دارد.

برای راه اندازی سرور استریم از nginx و ماژول nginx-rtmp استفاده شده و با استفاده از یک برنامه ضبط ویدئو مثل OBS میتوان تصاویر را زنده پخش کرد و یا با کمک گرفت از پکیج ffmpeg ویدئو های از قبل ضبط شده را در پلیر ها و سایت پخش کرد که در ادامه به این موارد به طور کامل اشاره می‌شود.

ماژول RTMP موجود و استفاده شده در این پروژه توانایی استریم مدیا به وسیله RTMP و HLS و MPEG-DASH را دارد و همچنین از ویژگی های این ماژول نوشته شده می‌توان به پشتیبانی از H264/AAC و HTTP و HTTP callback و HTTP control module اشاره کرد.

مراحل اجرا و پیاده سازی:

نصب پیشنیاز های ماژول و nginx و اضافه کردن ریپازیتوری مورد نیاز.

```
# yum -y groupinstall 'Development Tools'

# yum -y install epel-release

# yum install -y  wget git unzip perl perl-devel perl-ExtUtils-Embed libxslt libxslt-devel libxml2 libxml2-devel gd gd-devel pcre-devel GeoIP GeoIP-devel
```

دانلود و نصب پکیج های تکمیل کننده و کتابخانه مورد نیاز ماژول rtmp و nginx نسخه پایدار و سازگار با این ماژول.

نکته: در هنگام دانلود ممکن است ارور مربوط به لایسنس ssl سایت های منبع دریافت کنید که با اضافه کردن کامند --no-check-certificate به آخر آدرس سایت میتوان مشکل را رفع کرد. پس از دانلود نیاز به extract کردن و unarchive کردن فایل ها است.

وارد دایرکتوری مربوط به سورس برنامه ها شده و دانلود را شروع کنید.

```
# cd /usr/local/src

# wget https://nginx.org/download/nginx-1.18.0.tar.gz
# tar -xzvf nginx-1.18.0.tar.gz

# wget https://ftp.pcre.org/pub/pcre/pcre-8.42.zip
# unzip pcre-8.42.zip

# wget https://www.zlib.net/zlib-1.2.11.tar.gz
# tar -xzvf zlib-1.2.11.tar.gz

# wget https://www.openssl.org/source/openssl-1.1.0h.tar.gz
# tar -xzvf openssl-1.1.0h.tar.gz
```

دریافت کلون سورس کد ماژول از سایت گیت هاب و نصب آن از اکانت فرد سازنده ماژول sergey-dryabzhinsky

نکته: ممکن است در هنگام دانلود کد ها به مشکل لایسنس امنیتی گیت هاب مواجه شوید که برای رد کردن این اخطار از کامند env GIT_SSL_NO_VERIFY=true قبل از نوشتن آدرس سایت وکامند git clone استفاده کرد.

```
# git clone https://github.com/sergey-dryabzhinsky/nginx-rtmp-module.git
```

پس از extract کردن تمامی موارد دانلودی و دسترسی به سورس کد های ماژول حال آمادگی لازم برای کامپایل کردن و کانفیگ ماژول RTMP به همراه NGINX با یکدیگر را داریم برای این کار از دستورات زیر استفاده می‌شود که کانفیگ تنظیمات انجام شود و متغییر های مورد نیاز خود را نیز اضافه کنیم و تنظیمات هنگام نصب نیز پیاده سازی شوند. بیشتر این پارامتر ها و کانفیگ ها مربوط به ماژول rtmp هستند.

وارد دایرکتوری یا فولدر ساخته شده پس از اکسترکت nginx شده و عمل کانفیگ تنظیمات انجام گیرد.

```
# ./configure –prefix=/etc/nginx \
        --sbin-path=/usr/sbin/nginx \
        --modules-path=/usr/lib64/nginx/modules \
        --conf-path=/etc/nginx/nginx.conf \
        --error-log-path=/var/log/nginx/error.log \
        --pid-path=/var/run/nginx.pid \
        --lock-path=/var/run/nginx.lock \
        --user=nginx \
        --group=nginx \
        --build=CentOS \
        --builddir=nginx-1.18.0 \
        --with-select_module \
        --with-poll_module \
        --with-threads \
        --with-file-aio \
        --with-http_ssl_module \
        --with-http_v2_module \
        --with-http_realip_module \
        --with-http_addition_module \
        --with-http_xslt_module=dynamic \
        --with-http_image_filter_module=dynamic \
```

```
--with-http_geoip_module=dynamic \
--with-http_sub_module \
--with-http_dav_module \
--with-http_flv_module \
--with-http_mp4_module \
--with-http_gunzip_module \
--with-http_gzip_static_module \
--with-http_auth_request_module \
--with-http_random_index_module \
--with-http_secure_link_module \
--with-http_degradation_module \
--with-http_slice_module \
--with-http_stub_status_module \
--http-log-path=/var/log/nginx/access.log \
--http-client-body-temp-path=/var/cache/nginx/client_temp \
--http-proxy-temp-path=/var/cache/nginx/proxy_temp \
--http-fastcgi-temp-path=/var/cache/nginx/fastcgi_temp \
--http-uwsgi-temp-path=/var/cache/nginx/uwsgi_temp \
--http-scgi-temp-path=/var/cache/nginx/scgi_temp \
--with-mail=dynamic \
--with-mail_ssl_module \
--with-stream=dynamic \
--with-stream_ssl_module \
--with-stream_realip_module \
--with-stream_geoip_module=dynamic \
--with-stream_ssl_preread_module \
--with-compat \
--with-pcre=../pcre-8.42 \
--with-pcre-jit \
--with-zlib=../zlib-1.2.11 \
--with-openssl=../openssl-1.1.0h \
--with-openssl-opt=no-nextprotoneg \
--add-module=../nginx-rtmp-module \
--with-debug
```

خروجی دستورات بالا برای تایید درست انجام شدن آن شبیه به موارد ذیل است.

```
Configuration summary
+ using threads
+ using PCRE library: ../pcre-8.42
+ using OpenSSL library: ../openssl-1.1.0h
+ using zlib library: ../zlib-1.2.11

nginx path prefix: “/etc/nginx”
nginx binary file: “/usr/sbin/nginx”
nginx modules path: “/usr/lib64/nginx/modules”
nginx configuration prefix: “/etc/nginx”
nginx configuration file: “/etc/nginx/nginx.conf”
nginx pid file: “/var/run/nginx.pid”
nginx error log file: “/var/log/nginx/error.log”
nginx http access log file: “/var/log/nginx/access.log”
nginx http client request body temporary files: “/var/cache/nginx/client_temp”
nginx http proxy temporary files: “/var/cache/nginx/proxy_temp”
nginx http fastcgi temporary files: “/var/cache/nginx/fastcgi_temp”
nginx http uwsgi temporary files: “/var/cache/nginx/uwsgi_temp”
nginx http scgi temporary files: “/var/cache/nginx/scgi_temp”
```

حال پس از اجرای کانفیگ های مورد نیاز باید کد های ماژول کامپایل شوند .

```
# make
# make install
```

بعد از تکمیل نصب باید بین دایرکتوری ماژول و nginx لینک ایجاد شود و یوزر و گروه برای nginx ساخته شود و cache directory جدیدی برای nginx ساخته شود.

```
# ln -s /usr/lib64/nginx/modules /etc/nginx/modules
```

```
# useradd -r -d /var/cache/nginx/ -s /sbin/nologin -U nginx
```

```
# mkdir -p /var/cache/nginx/
# chown -R nginx:nginx /var/cache/nginx/
```

و در بخش آخر این مرحله تست nginx و ماژول نصب شده بر روی آن از نظر تنظیمات اجرا شده باید انجام شود.

```
# nginx -t
# nginx -v
```

حال به علت اینکه ما nginx را به صورت پکیج و دستی نصب کرده این نیاز است که آن را به عنوان یک سرویس به سیستم عامل معرفی کنیم که امکان اجرا و استفاده از آن وجود داشته باشد. موارد ذیل برای انجام اینکار به صورت غیر اتوماتیک و دستی است:

```
# cd /lib/systemd/system/
# nano nginx.service
```

پس از ساخت فایل مورد نظر در دایرکتوری مربوطه systemd کانفیگ های زیر را در فایل می‌نویسیم:

```
[Unit]
Description=nginx - high performance web server
Documentation=https://nginx.org/en/docs/
After=network-online.target remote-fs.target nss-lookup.target
Wants=network-online.target
[Service]
Type=forking
PIDFile=/var/run/nginx.pid
ExecStartPre=/usr/sbin/nginx -t -c /etc/nginx/nginx.conf
ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf
```

```
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s TERM $MAINPID
[Install]
WantedBy=multi-user.target
```

حال سرویس اصلی systemd که یک برنامه مدیریت سرویس و جایگزین init است را ری لود می‌کنیم.

```
# systemctl daemon-reload
```

سرویس Nginx را فعال و آن را برای اجرای اتوماتیک بعد از بوت سیستم معرفی می‌کنیم.

```
# systemctl start nginx
# systemctl enable nginx
```

بعد از انجام تمامی موارد فوق باید فایل کانفیگ اصلی nginx را برای استفاده از ماژول rtmp و توانایی استریم ویدئو تغییر دهیم.

اول از فایل اصلی nginx.conf یک بکاپ می‌گیریم.

```
# cd /etc/nginx/
# mv nginx.conf  nginx.conf.asli
```

فایل جدید را ایجاد و کد های زیر را در آن مینویسیم. `# nano nginx.conf`

```
worker_processes  auto;
events {
   worker_connections  1024;
}
# RTMP configuration
rtmp {
   server {
```

```
        listen 1935; # Listen on standard RTMP port
        chunk_size 4000;
# Define the Application
        application show {
            live on;
            # Turn on HLS
            hls on;
            hls_path /mnt/hls/;
            hls_fragment 3;
            hls_playlist_length 60;
            # disable consuming the stream from nginx as rtmp
            deny play all;
        }
    }
   # RTMP video on demand for mp4 files
        application vod {
            play /mnt/mp4s;
        }
        # RTMP stream using OBS
        application stream {
            live on;
        }
}
http {
    sendfile off;
```

```
tcp_nopush on;
aio on;
directio 512;
default_type application/octet-stream;
server {
    listen 8080;
    location / {
        # Disable cache
        add_header 'Cache-Control' 'no-cache';
        # CORS setup
        add_header 'Access-Control-Allow-Origin' '*' always;
        add_header 'Access-Control-Expose-Headers' 'Content-Length';
        # allow CORS preflight requests
        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/dash+xml mpd;
            application/vnd.apple.mpegurl m3u8;
            video/mp2t ts;
        }
```

```
        root /mnt/;
      }
    }
}
```

و در قسمت آخر این مرحله تست کد های نوشته شده از نظر قابل اجرا بودن و ری استارت کردن سرویس nginx باید انجام شود.

```
# nginx -t
# systemctl restart nginx
```

در آخرین مرحله اجرایی و قبل از تست کلی میتوان برای فیلم هایی که باید به صورت vod پخش شوند یک دایرکتوری ایجاد کرد و آنهارا در آن ذخیره کرد و به یوزر nginx باید دسترسی خواندن اطلاعات داد.

```
# mkdir -p /mnt/mp4s
# chown -R nginx:nginx /mnt/mp4s
```

بقیه موارد مربوط به سیستم فرد استریمر است و باید با استفاده از آدرس rtmp و نرم افزار VLCویدئو های موجود در سرور را پخش کند و یا با نرم افزار OBS استریم صفحه نمایش سیستم خود را برای سرور ارسال کند و دیگران نیز آن را در پلیر خود پخش کنند. مثال:

rtmp://192.168.1.196:1935/vod/file.mp4