

<https://docs.nginx.com/nginx/admin-guide/content-cache/content-caching/>

<https://serverfault.com/questions/223509/how-can-i-see-which-flags-nginx-was-compiled-with>

Configuration summary

- + using system PCRE library
- + using system OpenSSL library
- + using system zlib library

nginx path prefix: "/usr/local/nginx"
nginx binary file: "/usr/local/nginx/sbin/nginx"
nginx modules path: "/usr/local/nginx/modules"
nginx configuration prefix: "/usr/local/nginx/conf"
nginx configuration file: "/usr/local/nginx/conf/nginx.conf"
nginx pid file: "/usr/local/nginx/logs/nginx.pid"
nginx error log file: "/usr/local/nginx/logs/error.log"
nginx http access log file: "/usr/local/nginx/logs/access.log"
nginx http client request body temporary files: "client_body_temp"
nginx http proxy temporary files: "proxy_temp"
nginx http fastcgi temporary files: "fastcgi_temp"
nginx http uwsgi temporary files: "uwsgi_temp"
nginx http scgi temporary files: "scgi_temp"

-ldl -lpthread -lcrypt -lpcres -lssl -lcrypto -ldl -lpthread -lz \
-Wl,-E
sed -e "s|%%PREFIX%%|usr/local/nginx|" \
-e "s|%%PID_PATH%%|usr/local/nginx/logs/nginx.pid|" \
-e "s|%%CONF_PATH%%|usr/local/nginx/conf/nginx.conf|" \
-e "s|%%ERROR_LOG_PATH%%|usr/local/nginx/logs/error.log|" \
< man/nginx.8 > objs/nginx.8
make[1]: Leaving directory '/root/nginx-1.15.1'

root@iptv:~/nginx-1.15.1# make install
make -f objs/Makefile install
make[1]: Entering directory '/root/nginx-1.15.1'
test -d '/usr/local/nginx' || mkdir -p '/usr/local/nginx'
test -d '/usr/local/nginx/sbin' \
|| mkdir -p '/usr/local/nginx/sbin'

```
test ! -f '/usr/local/nginx/sbin/nginx' \  
  || mv '/usr/local/nginx/sbin/nginx' \  
    '/usr/local/nginx/sbin/nginx.old'  
cp objs/nginx '/usr/local/nginx/sbin/nginx'  
test -d '/usr/local/nginx/conf' \  
  || mkdir -p '/usr/local/nginx/conf'  
cp conf/koi-win '/usr/local/nginx/conf'  
cp conf/koi-utf '/usr/local/nginx/conf'  
cp conf/win-utf '/usr/local/nginx/conf'  
test -f '/usr/local/nginx/conf/mime.types' \  
  || cp conf/mime.types '/usr/local/nginx/conf'  
cp conf/mime.types '/usr/local/nginx/conf/mime.types.default'  
test -f '/usr/local/nginx/conf/fastcgi_params' \  
  || cp conf/fastcgi_params '/usr/local/nginx/conf'  
cp conf/fastcgi_params \  
  '/usr/local/nginx/conf/fastcgi_params.default'  
test -f '/usr/local/nginx/conf/fastcgi.conf' \  
  || cp conf/fastcgi.conf '/usr/local/nginx/conf'  
cp conf/fastcgi.conf '/usr/local/nginx/conf/fastcgi.conf.default'  
test -f '/usr/local/nginx/conf/uwsgi_params' \  
  || cp conf/uwsgi_params '/usr/local/nginx/conf'  
cp conf/uwsgi_params \  
  '/usr/local/nginx/conf/uwsgi_params.default'  
test -f '/usr/local/nginx/conf/scgi_params' \  
  || cp conf/scgi_params '/usr/local/nginx/conf'  
cp conf/scgi_params \  
  '/usr/local/nginx/conf/scgi_params.default'  
test -f '/usr/local/nginx/conf/nginx.conf' \  
  || cp conf/nginx.conf '/usr/local/nginx/conf/nginx.conf'  
cp conf/nginx.conf '/usr/local/nginx/conf/nginx.conf.default'  
test -d '/usr/local/nginx/logs' \  
  || mkdir -p '/usr/local/nginx/logs'  
test -d '/usr/local/nginx/logs' \  
  || mkdir -p '/usr/local/nginx/logs'  
test -d '/usr/local/nginx/html' \  
  || cp -R html '/usr/local/nginx'  
test -d '/usr/local/nginx/logs' \  
  || mkdir -p '/usr/local/nginx/logs'  
make[1]: Leaving directory '/root/nginx-1.15.1'
```

<https://www.dreamvps.com/en/tutorials/article/hls-and-nginx-on-ubuntu>

VVV

<https://www.techbeginner.in/2020/01/how-to-install-nginx-rtmp-server-with.html>

```
-----  
root@iptv:/usr/local/nginx/conf# cat nginx.conf
```

```
#user nobody;
```

```
worker_processes 1;
```

```
#error_log logs/error.log;
```

```
#error_log logs/error.log notice;
```

```
#error_log logs/error.log info;
```

```
#pid logs/nginx.pid;
```

```
events {
```

```
    worker_connections 1024;
```

```
}
```

```
rtmp {
```

```
server {
```

```
listen 1935; # Listen on standard RTMP port
```

```
chunk_size 4000;
```

```
application hls {
```

```
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;
```

```
}
```

```
}
```

```
}
```

```
http {
```

```
include mime.types;
```

```
default_type application/octet-stream;
```

```
#log_format main '$remote_addr - $remote_user [$time_local] "$request" '
#                 '$status $body_bytes_sent "$http_referer" '
#                 '"$http_user_agent" "$http_x_forwarded_for"';

#access_log logs/access.log main;

sendfile        on;
#tcp_nopush     on;

#keepalive_timeout 0;
keepalive_timeout 65;

#gzip on;

server {
listen 8080;

location /hls {
# Disable cache
add_header Cache-Control no-cache;

# CORS setup
add_header 'Access-Control-Allow-Origin' '*';
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/vnd.apple.mpegurl m3u8;
video/mp2t ts;
}

root /mnt;
}
}
```

```
server {
listen 80;
server_name localhost;

#charset koi8-r;

#access_log logs/host.access.log main;

location / {
root html;
index index.html index.htm;
}

#error_page 404          /404.html;

# redirect server error pages to the static page /50x.html
#
error_page 500 502 503 504 /50x.html;
location = /50x.html {
root html;
}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80
#
#location ~ \.php$ {
#    proxy_pass http://127.0.0.1;
#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
#
#location ~ \.php$ {
#    root          html;
#    fastcgi_pass 127.0.0.1:9000;
#    fastcgi_index index.php;
#    fastcgi_param SCRIPT_FILENAME /scripts$fastcgi_script_name;
#    include       fastcgi_params;
#}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
#location ~ /\.ht {
#    deny all;
}
```

```

#}
}

# another virtual host using mix of IP-, name-, and port-based configuration
#
#server {
#    listen 8000;
#    listen somename:8080;
#    server_name somename alias another.alias;

#    location / {
#        root html;
#        index index.html index.htm;
#    }
#}

# HTTPS server
#
#server {
#    listen 443 ssl;
#    server_name localhost;

#    ssl_certificate      cert.pem;
#    ssl_certificate_key  cert.key;

#    ssl_session_cache    shared:SSL:1m;
#    ssl_session_timeout  5m;

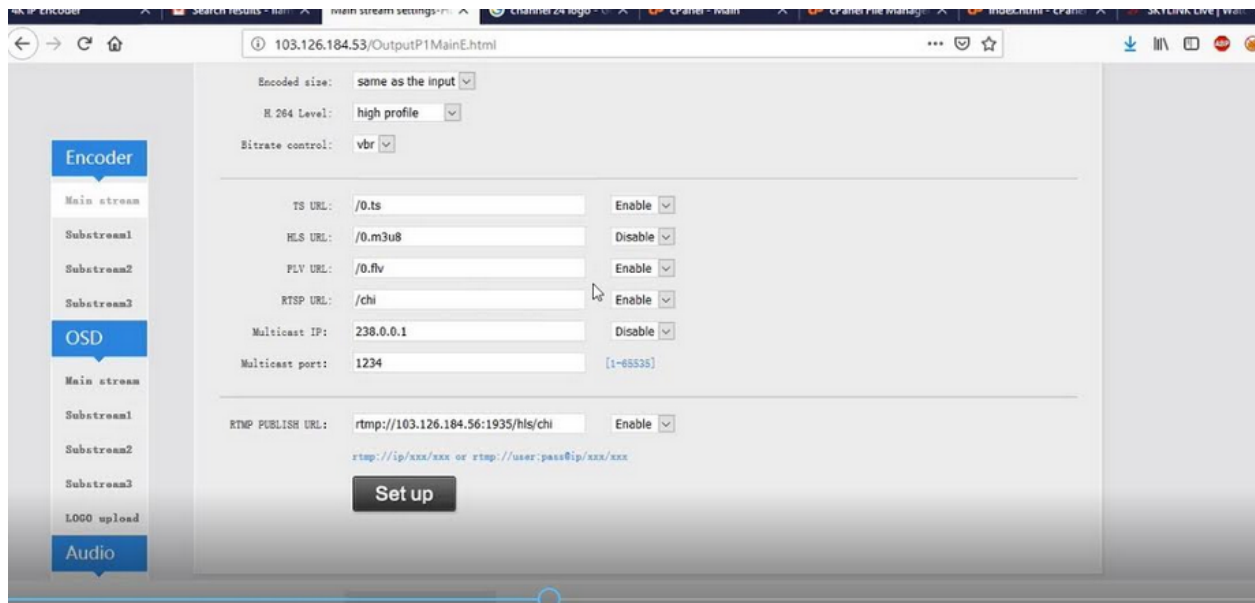
#    ssl_ciphers HIGH:!aNULL:!MD5;
#    ssl_prefer_server_ciphers on;

#    location / {
#        root html;
#        index index.html index.htm;
#    }
#}
}

```

VLC TEST

<http://103.152.237.243:8080/hls/chi.m3u8>



How To Install Nginx RTMP Server with HLS on Ubuntu 16.04 | 18.04 | 20.04

[Aftab Ali January 09, 2020 nginx.](#)

Share This

AddThis Sharing Buttons

[Share to Facebook](#)

[Share to Twitter](#)[Share to WhatsApp](#)[Share to More](#)

Nginx RTMP is a TCP-based convention intended to keep up low-dormancy associations for sound and video spilling. To expand the measure of information that can be easily transmitted, streams are part into littler sections called parcels. RTMP additionally characterizes a few virtual channels that work autonomously of one another for bundles to be conveyed on. This implies video and sound are conveyed on discrete channels all the while, To get more information click [here](#) to visit the official website.

Install Dependencies

To install Nginx RTMP server on ubuntu use the following commands with using root privileges.

©

```
sudo -i
```

```
apt-get update
```

```
apt-get install ffmpeg libpcre3 unzip libssl-dev build-essential  
libpcre3-dev -y
```

Download Nginx and RTMP Modules

To download the nginx and rtmp module use the following commands.

©

```
cd /tmp
```

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

```
wget http://nginx.org/download/nginx-1.14.0.tar.gz
```

Extract and Compile the Nginx with RTMP module

After downloaded nginx with RTMP module, You need to extract and unzip the master.zip packages and compile the nginx rtmp module by following the commands.

©

```
tar -zxvf nginx-1.14.0.tar.gz
```

```
unzip master.zip
```

```
cd nginx-1.14.0
```



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

```
make
```

```
make install
```

Configuration of Nginx Daemon

To control nginx daemon, We need to download the pre-define the service of nginx and make executable it by following the commands.

©

```
cd /tmp
```

```
wget
```

```
https://raw.githubusercontent.com/JasonGiedymin/nginx-init-ubuntu/master/nginx  
-O /etc/init.d/nginx
```

```
chmod +x /etc/init.d/nginx
```

On boot enable nginx service

If you want start the nginx service on boot the server use the following commands.

©

```
update-rc.d nginx defaults
```

Configuration of RTMP protocol in nginx config file by using the following the commands, Create a backup file of nginx.conf and than edit the original nginx.conf

©

```
cp -p /usr/local/nginx/conf/nginx.conf nginx.conf_backup
```

Open the nginx.conf with nano editor .

©

```
nano /usr/local/nginx/conf/nginx.conf
```

Add the following configuration.

©

```
#user nobody;
```

```
worker_processes 1;
```

```
error_log logs/rtmp_error.log debug;
```

```
pid logs/nginx.pid;
```

```
events {
```

```
    worker_connections 1024;
```

```
}
```

```
http {
```

```
    server {
```

```
        listen 80;
```

```
        server_name localhost;
```

```
location /hls {

    # Serve HLS fragments

    # 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/vnd.apple.mpegurl m3u8;

        video/mp2t ts;
```

```
    }

    root /tmp;

    add_header Cache-Control no-cache;

}

}

}

rtmp {

    server {

        listen 1935;

        chunk_size 8192;

        application hls {

            live on;

            meta copy;

            hls on;

            hls_path /tmp/hls;

        }

    }

}
```

Save and Exit from nano editor.

Create required directory for Stream and recording data by following the commands.

©

mkdir /mnt/hls

```
mkdir /HLS
```

```
mkdir /HLS/live
```

```
mkdir /HLS/mobile
```

```
mkdir /video_recordings
```

```
chmod -R 777 /video_recordings
```

Restart the nginx service

©

```
systemctl restart the nginx.service
```

Update the UFW firewall

If you have enable ufw firewall so then you need to allow the port 80 and port 1935 for rtmp protocol to access from the network, Use the following commands to open port 80 and port 1935.

©

```
ufw allow 80
```

```
ufw allow 1935
```

```
ufw status
```

Check the nginx's RTMP service using netstate commands.

©

```
netstate -plntu | grep 1935
```

Now you can stream with any key using OBS and Webcam, Use the given details.

©

```
rtmp://localhost/hls
```

To view the live HLS stream open you online player and enter the given url.

©

```
http://localhost/HLS/stream_name.m3u8
```