

Nginx



```
//  
yum install prel  
yum install pcre pcre-devel  
sudo yum install libxml2 libxml2-devel libxslt libxslt-devel  
sudo yum install gd gd-devel  
  
//  
nginx -V  
  
//  
make clean  
  
//  
./configure \  
--prefix=/usr/share/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 \  
--http-log-path=/var/log/nginx/access.log \  
--http-client-body-temp-path=/var/lib/nginx/tmp/client_body \  
--http-proxy-temp-path=/var/lib/nginx/tmp/proxy \  
--http-fastcgi-temp-path=/var/lib/nginx/tmp/fastcgi \  
--http-uwsgi-temp-path=/var/lib/nginx/tmp/uwsgi \  
--http-scgi-temp-path=/var/lib/nginx/tmp/scgi \  
--pid-path=/run/nginx.pid \  
--lock-path=/run/lock/subsys/nginx \  
--user=nginx \  
--group=nginx \  
--with-compat \  
--with-debug \  
--with-file-aio \  
--with-http_addition_module \  
--with-http_auth_request_module \  

```

```

--with-http_dav_module \
--with-http_degradation_module \
--with-http_flv_module \
--with-http_gunzip_module \
--with-http_gzip_static_module \
--with-http_image_filter_module=dynamic \
--with-http_mp4_module \
--with-http_perl_module=dynamic \
--with-http_random_index_module \
--with-http_realip_module \
--with-http_secure_link_module \
--with-http_slice_module \
--with-http_ssl_module \
--with-http_stub_status_module \
--with-http_sub_module \
--with-http_v2_module \
--with-http_xslt_module=dynamic \
--with-mail=dynamic \
--with-mail_ssl_module \
--with-pcre \
--with-pcre-jit \
--with-stream_ssl_module \
--with-stream_ssl_preread_module \
--with-stream \
--with-threads \
--with-cc-opt='-O2 -flto=auto -ffat-lto-objects -fexceptions -g -grecord-gcc-switches -pipe -Wall -Werror=format-
security -Wp,-D_FORTIFY_SOURCE=2 -Wp,-D_GLIBCXX_ASSERTIONS -specs=/usr/lib/rpm/redhat/redhat-
hardened-cc1 -fstack-protector-strong -specs=/usr/lib/rpm/redhat/redhat-annobin-cc1 -m64 -march=x86-64-v2 -
mtune=generic -fasynchronous-unwind-tables -fstack-clash-protection -fcf-protection' \
--with-ld-opt='-Wl,-z,relro -Wl,--as-needed -Wl,-z,now -specs=/usr/lib/rpm/redhat/redhat-hardened-ld -
specs=/usr/lib/rpm/redhat/redhat-annobin-cc1 -Wl,-E' \
--add-module=/home/nginx-http-flv-module-master \
--add-module=/home/nginx-vod-module-master
//[] Makefile[]
make
//[]
make install

```

dhparam

```
// Nginx SSL/TLS DH
openssl dhparam -out /etc/nginx/ssl/dhparam.pem 2048
```

nginx-http-flv-module

```
# http server
server {
    listen 443 ssl;
    listen [::]:443 ssl;
    http2 on;
    server_name rtmp.lingyanspace.com;

    location /live {
        flv_live on;
        # 'Transfer-Encoding: chunked'
        chunked_transfer_encoding on;
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Credentials' 'true';
    }

    location /vod {
        types {
            application/vnd.apple.mpegurl m3u8;
            video/mp2t ts;
        }
        alias /home/lingyanassets/vods;
        expires -1;
    }

    # control
    location /control {
        rtmp_control all;
    }

    location /stat {
        #
        rtmp_stat all;
        rtmp_stat_stylesheet stat.xml;
    }
}
```

```

location /stat.xsl {
    root /home/nginx-http-flv-module-master;
}
}
#rtmp server
server {
    listen 1935;
    chunk_size 4096;
    application live {
        #
        live on;
        #
        allow publish all;
        #
        allow play all;
        #
        record off;
        #  HLS
        hls on;
        # HLS
        hls_path /home/lingyanassets/lives;
        #  10
        hls_fragment 10s;
        #  60
        hls_playlist_length 60m;
        #
        hls_continuous on;
        #
        hls_cleanup off;
    }
    #  'video'
    application vod {
        play /home/lingyanassets/vods; #
    }
}

```

nginx-vod-module-master

```
server {

    listen 443 ssl;
    listen [::]:443 ssl;
    http2 on;
    server_name hls.lingyanspace.com;


    # 512MB
    vod_metadata_cache metadata_cache 512m;
    # 256MB
    vod_response_cache response_cache 256m;
    # VOD
    vod_mode local;
    # 2
    vod_segment_duration 2000;
    #
    vod_align_segments_to_key_frames on;


    #

    location /LimitVideo/ {

        auth_request /test_auth;


        alias /home/lingyanspace/LimitVideo/;

        vod hls;

        add_header Access-Control-Allow-Headers '*';

        add_header Access-Control-Expose-Headers 'Server,range,Content-Length,Content-Range';

        add_header Access-Control-Allow-Methods 'GET, HEAD, OPTIONS';

        add_header Access-Control-Allow-Origin '*';
```

```

        expires 100d;

    }

    location =/test_auth {

        proxy_pass http://127.0.0.1:7777/LimitVideo;

        proxy_pass_request_body off;

        proxy_set_header X-Original-URI $request_uri;

        proxy_set_header X-Original-Method $request_method;

        proxy_set_header X-Forwarded-Proto $scheme;

    }

    access_log /var/log/nginx/hls.lingyanspace.com_access.log;

    error_log /var/log/nginx/hls.lingyanspace.com_error.log;

}

```

nginx

[Unit]

Description=The nginx HTTP and reverse proxy server

After=network-online.target remote-fs.target nss-lookup.target

Wants=network-online.target

[Service]

Type=forking

PIDFile=/run/nginx.pid

Nginx will fail to start if /run/nginx.pid already exists but has the wrong

SELinux context. This might happen when running `nginx -t` from the cmdline.

```
# https://bugzilla.redhat.com/show_bug.cgi?id=1268621
ExecStartPre=/usr/bin/rm -f /run/nginx.pid
ExecStartPre=/usr/sbin/nginx -t
ExecStart=/usr/sbin/nginx
ExecReload=/usr/sbin/nginx -s reload
KillSignal=SIGQUIT
TimeoutStopSec=5
KillMode=mixed
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

nginx 和 signalr

```
location ~* /(lingyanchat)$ {
    proxy_pass http://lingyanasp;

    # 设置 Host 头
    proxy_set_header Host $host:$server_port;
    # 设置 HTTP 版本
    proxy_http_version 1.1;
    # 设置 WebSocket 和 HTTP/2 支持
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}
```

nginx 配置

```
user nginx; # nginx 用户和组

worker_processes auto; # 设置 worker 进程数 (默认 4 或 8G 内存)
worker_cpu_affinity auto; # 设置 worker 进程的 CPU 亲和性

error_log /var/log/nginx/error.log warn; # 设置日志级别为 warn
```

```
pid /var/run/nginx.pid;
```

```
worker_rlimit_nofile 16384;#  nginx  -- worker_processes * worker_connections
```

```
events {
```

```
    use epoll;
```

```
    worker_connections 4096; # worker_processes *
```

```
1024
```

```
    multi_accept on; #
```

```
}#
```

```
http {
```

```
    include mime.types;#  MIME
```

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

```
    charset utf-8; #
```

```
#
```

```
log_format main '{"@timestamp":'$time_iso8601','
```

```
    "host":'$server_addr','
```

```
    "clientip":'$remote_addr','
```

```
    "size":$body_bytes_sent,'
```

```
    "responsetime":$request_time,'
```

```
    "upstreamtime":'$upstream_response_time','
```

```
    "upstreamhost":'$upstream_addr','
```

```
    "http_host":'$host','
```

```
    "url":'$uri','
```

```
    "xff":'$http_x_forwarded_for','
```

```
    "referer":'$http_referer','
```

```
    "agent":'$http_user_agent','
```

```
    "status":'$status'}';
```

```
access_log /var/log/nginx/access.log main;
```

```
sendfile on; #  sendfile
```

```
tcp_nopush on; #  TCP_CORK
```

```
tcp_nodelay on; #  TCP_NODELAY  Nagle
```

```
server_names_hash_bucket_size 128; #  server_names 128
```

```
server_names_hash_max_size 512; #  server_names 512
```

```

keepalive_timeout 65; # HTTP 保持连接时间 65 s
client_header_timeout 15M; # 客户端请求头接收超时时间 15 M
client_body_timeout 15M; # 客户端请求体接收超时时间 15 M
send_timeout 60s; # 发送超时时间 60 s

limit_conn_zone $binary_remote_addr zone=perip:10m; # 按IP地址限制连接数 10MB
limit_conn_zone $server_name zone=perserver:10m; # 按服务器限制连接数 10MB
limit_conn perip 2; # 按IP地址限制连接数 2
limit_conn perserver 20; # 按服务器限制连接数 20
limit_rate 300k; # 限制速率 300KB/s

proxy_cache_path /data/nginx-cache levels=1:2 keys_zone=nginx-cache:20m max_size=50g inactive=168h;
# 代理缓存路径

client_body_buffer_size 512k; # 客户端请求体缓冲区大小 512KB
client_header_buffer_size 4k; # 客户端请求头缓冲区大小 4KB
client_max_body_size 1G; # 客户端最大请求体大小 512KB
large_client_header_buffers 2 8k; # 大客户端请求头缓冲区数及大小
proxy_connect_timeout 5s; # 代理连接超时时间 5 s
proxy_send_timeout 120s; # 代理发送超时时间 120 s
proxy_read_timeout 120s; # 代理接收超时时间 120 s
proxy_buffer_size 16k; # 代理缓冲区大小
proxy_buffers 4 64k; # 代理缓冲区数及大小
proxy_busy_buffers_size 128k; # 代理忙缓冲区大小
proxy_temp_file_write_size 128k; # 代理临时文件写入大小
proxy_next_upstream http_502 http_504 http_404 error timeout invalid_header; #
# 代理下一个上游的条件

gzip on; # 是否开启gzip压缩
gzip_min_length 1k; # 最小压缩长度 1KB
gzip_buffers 4 16k; # gzip缓冲区数及大小
gzip_http_version 1.1; # HTTP/1.1 版本
gzip_comp_level 4; # gzip压缩级别 4
gzip_types text/plain text/css application/json application/x-javascript text/xml application/xml
application/xml+rss text/javascript; # 支持的MIME类型
gzip_vary on; # 是否开启Vary: Accept-Encoding头

fastcgi_buffers 64 4K; # FastCGI 缓冲区数及大小

```

```

open_file_cache max=2000 inactive=10m; # 2000 10
open_file_cache_valid 4m; # 4
open_file_cache_min_uses 2; # 2
open_file_cache_errors on; #
aio on; # I/O

proxy_set_header Host $host; # Host
proxy_set_header X-Real-IP $remote_addr; # IP
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; # IP
proxy_set_header X-Forwarded-Proto $scheme; # HTTP HTTPS

# *.lingyanspace.com HTTP
server {
    listen 80;
    listen [::]:80;
    #
    server_name *.lingyanspace.com;
    # http https
    return 301 https://$host$request_uri;
}

# *.lingyanspace.com HTTPS
server {

    listen 443 ssl;
    listen [::]:443 ssl;
    http2 on;
    server_name *.lingyanspace.com;

    # SSL
    ssl_certificate /etc/nginx/ssl/lingyanspace/GeneralFullChainCert.pem;
    ssl_certificate_key /etc/nginx/ssl/lingyanspace/GeneralPrivateKey.pem;
    ssl_dhparam /etc/nginx/ssl/dhparam.pem;

    # SSL
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;

    ssl_protocols TLSv1.2 TLSv1.3;

```

```
ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
ssl_prefer_server_ciphers on;

#ocsp stapling

ssl_stapling on;
ssl_stapling_verify on;

resolver 223.5.5.5 223.6.6.6 119.29.29.29 119.28.28.28 180.76.76.76 1.2.4.8 114.114.114.114 1.1.1.1
1.0.0.1 valid=300s;# DNS cloudflare
resolver_timeout 5s;
}

# conf.d .http.conf
include conf.d/*.http.conf;

}
```

Nginx

```
ps aux | grep nginx | grep worker | awk '{print $2}' | xargs -l {} cat /proc/{}/limits | grep
"Max open files"
```

		#21		
			31	2024 20:46:26
			13	2024 16:55:34