



This repository Search

Pull requests Issues Gist



arut / nginx-rtmp-module

Watch 523 Star 4,421 Fork 1,359

Code Issues 471 Pull requests 51 Projects 0 Wiki Pulse Graphs

NGINX-based Media Streaming Server <http://nginx-rtmp.blogspot.com>



948 commits 150 branches 109 releases 10 contributors BSD-2-Clause

Branch: master New pull request Create new file Upload files Find file Clone or download

arut	support for nginx 1.11.5-style cache-manager	Latest commit 5150993 on Oct 7, 2016
dash	support for nginx 1.11.5-style cache-manager	4 months ago
doc	added documentation link	4 years ago
hls	support for nginx 1.11.5-style cache-manager	4 months ago
test	removed trailing space	3 years ago
AUTHORS	improved frame timing & added meta files	5 years ago
LICENSE	updated year	3 years ago
README.md	Use same path in the comments and in the code	3 years ago
config	fixed rtmp as a dynamic module	6 months ago
ngx_rtmp.c	fixed rtmp as a dynamic module	6 months ago
ngx_rtmp.h	fixed compilation with nginx 1.7.11	2 years ago

<a href="#">ngx_rtmp_access_module.c</a>	removed trailing spaces	3 years ago
<a href="#">ngx_rtmp_amf.c</a>	improved Envivio fix: now key length can be unfinalized with object	3 years ago
<a href="#">ngx_rtmp_amf.h</a>	updated copyright line	3 years ago
<a href="#">ngx_rtmp_auto_push_module.c</a>	fixed rtmp as a dynamic module	6 months ago
<a href="#">ngx_rtmp_bandwidth.c</a>	removed trailing spaces	3 years ago
<a href="#">ngx_rtmp_bandwidth.h</a>	added audio and video bw to stat	3 years ago
<a href="#">ngx_rtmp_bitop.c</a>	implemented avc sps parser & improved dash & stats	3 years ago
<a href="#">ngx_rtmp_bitop.h</a>	implemented avc sps parser & improved dash & stats	3 years ago
<a href="#">ngx_rtmp_cmd_module.c</a>	implemented play2 time continuation	3 years ago
<a href="#">ngx_rtmp_cmd_module.h</a>	shared fill args func	3 years ago
<a href="#">ngx_rtmp_codec_module.c</a>	fixed windows compilation	3 years ago
<a href="#">ngx_rtmp_codec_module.h</a>	added aac-he & aac-hev2 support	3 years ago
<a href="#">ngx_rtmp_control_module.c</a>	fixed errors in control output; now returning http 206 instead of emp...	3 years ago
<a href="#">ngx_rtmp_core_module.c</a>	fixed rtmp as a dynamic module	6 months ago
<a href="#">ngx_rtmp_eval.c</a>	removed trailing spaces	3 years ago
<a href="#">ngx_rtmp_eval.h</a>	merged exec pull feature	3 years ago
<a href="#">ngx_rtmp_exec_module.c</a>	added filename, basename & dirname to exec_record_done	3 years ago
<a href="#">ngx_rtmp_flv_module.c</a>	implemented default buflen setting	3 years ago
<a href="#">ngx_rtmp_handler.c</a>	reset byte counter at 0xf0000000 to fix publishing from fmle	3 years ago
<a href="#">ngx_rtmp_handshake.c</a>	removed trailing spaces	3 years ago
<a href="#">ngx_rtmp_init.c</a>	added queue init	2 years ago

<a href="#">ngx_rtmp_limit_module.c</a>	removed trailing spaces	3 years ago
<a href="#">ngx_rtmp_live_module.c</a>	reverted changes in live module	3 years ago
<a href="#">ngx_rtmp_live_module.h</a>	reverted changes in live module	3 years ago
<a href="#">ngx_rtmp_log_module.c</a>	fixed issue with multiple access_logs	2 years ago
<a href="#">ngx_rtmp_mp4_module.c</a>	fixed skipping first key frame in mp4 streamer	3 years ago
<a href="#">ngx_rtmp_netcall_module.c</a>	added clientid to notifications & control	3 years ago
<a href="#">ngx_rtmp_netcall_module.h</a>	removed trailing spaces	3 years ago
<a href="#">ngx_rtmp_notify_module.c</a>	added epoch argument to on_connect	3 years ago
<a href="#">ngx_rtmp_play_module.c</a>	fixed compilation	2 years ago
<a href="#">ngx_rtmp_play_module.h</a>	removed trailing spaces	3 years ago
<a href="#">ngx_rtmp_proxy_protocol.c</a>	fixed compilation with old nginx	3 years ago
<a href="#">ngx_rtmp_proxy_protocol.h</a>	added proxy protocol support	3 years ago
<a href="#">ngx_rtmp_receive.c</a>	removed trailing spaces	3 years ago
<a href="#">ngx_rtmp_record_module.c</a>	fixed rtmp as a dynamic module	6 months ago
<a href="#">ngx_rtmp_record_module.h</a>	recorder now writes correct flv track mask	3 years ago
<a href="#">ngx_rtmp_relay_module.c</a>	enabled static relay cleanup to fix static_relay+auto_push combination	2 years ago
<a href="#">ngx_rtmp_relay_module.h</a>	removed trailing spaces	3 years ago
<a href="#">ngx_rtmp_send.c</a>	removed trailing spaces	3 years ago
<a href="#">ngx_rtmp_shared.c</a>	removed trailing spaces	3 years ago
<a href="#">ngx_rtmp_stat_module.c</a>	fixed formatting error in stat module	3 years ago
<a href="#">ngx_rtmp_streams.h</a>	updated copyright line	3 years ago

 <a href="#">ngx_rtmp_version.h</a>	version bump	3 years ago
 <a href="#">stat.xsl</a>	moved bytes/bw block to the right of stat page	3 years ago

## README.md

# NGINX-based Media Streaming Server

---

## nginx-rtmp-module

---

### Project blog

<http://nginx-rtmp.blogspot.com>

### Wiki manual

<https://github.com/arut/nginx-rtmp-module/wiki/Directives>

### Google group

<https://groups.google.com/group/nginx-rtmp>

<https://groups.google.com/group/nginx-rtmp-ru> (Russian)

### Donation page (Paypal etc)

<http://arut.github.com/nginx-rtmp-module/>

## Features

- RTMP/HLS/MPEG-DASH live streaming
- RTMP Video on demand FLV/MP4, playing from local filesystem or HTTP
- Stream relay support for distributed streaming: push & pull models
- Recording streams in multiple FLVs
- H264/AAC support
- Online transcoding with FFmpeg
- HTTP callbacks (publish/play/record/update etc)
- Running external programs on certain events (exec)
- HTTP control module for recording audio/video and dropping clients
- Advanced buffering techniques to keep memory allocations at a minimum level for faster streaming and low memory footprint
- Proved to work with Wirecast, FMS, Wowza, JWPlayer, FlowPlayer, StrobeMediaPlayback, ffmpeg, avconv, rtmpdump, flvstreamer and many more
- Statistics in XML/XSL in machine- & human- readable form
- Linux/FreeBSD/MacOS/Windows

## Build

cd to NGINX source directory & run this:

```
./configure --add-module=/path/to/nginx-rtmp-module  
make  
make install
```

Several versions of nginx (1.3.14 - 1.5.0) require `http_ssl_module` to be added as well:

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

For building debug version of nginx add `--with-debug`

```
./configure --add-module=/path/to/nginx-rtmp-module --with-debug
```

[Read more about debug log](#)

## Windows limitations

Windows support is limited. These features are not supported

- execs
- static pulls
- auto\_push

## RTMP URL format

```
rtmp://rtmp.example.com/app[/name]
```

app - should match one of application {} blocks in config

name - interpreted by each application can be empty

## Multi-worker live streaming

Module supports multi-worker live streaming through automatic stream pushing to nginx workers. This option is toggled with `rtmp_auto_push` directive.

## Example nginx.conf

```
rtmp {  
  
    server {  
  
        listen 1935;  
  
        chunk_size 4000;  
  
        # TV mode: one publisher, many subscribers  
        application mytv {  
  
            # enable live streaming  
            live on;  
  
            # record first 1K of stream  
            record all;  
            record_path /tmp/av;  
            record_max_size 1K;  
  
            # append current timestamp to each flv  
            record_unique on;  
  
            # publish only from localhost
```

```
        allow publish 127.0.0.1;
        deny publish all;

        #allow play all;
    }

    # Transcoding (ffmpeg needed)
    application big {
        live on;

        # On every published stream run this command (ffmpeg)
        # with substitutions: $app/${app}, $name/${name} for application & stream name.
        #
        # This ffmpeg call receives stream from this application &
        # reduces the resolution down to 32x32. The stream is then published to
        # 'small' application (see below) under the same name.
        #
        # ffmpeg can do anything with the stream like video/audio
        # transcoding, resizing, altering container/codecs etc
        #
        # Multiple exec lines can be specified.

        exec ffmpeg -re -i rtmp://localhost:1935/$app/$name -vcodec flv -acodec copy -s 32x32
            -f flv rtmp://localhost:1935/small/${name};
    }

    application small {
        live on;
        # Video with reduced resolution comes here from ffmpeg
    }

    application webcam {
        live on;
```



```
    # Stream from local webcam
    exec_static ffmpeg -f video4linux2 -i /dev/video0 -c:v libx264 -an
        -f flv rtmp://localhost:1935/webcam/mystream;
}

application mypush {
    live on;

    # Every stream published here
    # is automatically pushed to
    # these two machines
    push rtmp1.example.com;
    push rtmp2.example.com:1934;
}

application mypull {
    live on;

    # Pull all streams from remote machine
    # and play locally
    pull rtmp://rtmp3.example.com pageUrl=www.example.com/index.html;
}

application mystaticpull {
    live on;

    # Static pull is started at nginx start
    pull rtmp://rtmp4.example.com pageUrl=www.example.com/index.html name=mystream static;
}

# video on demand
application vod {
    play /var/flvs;
}
```

```
application vod2 {
    play /var/mp4s;
}

# Many publishers, many subscribers
# no checks, no recording
application videochat {

    live on;

    # The following notifications receive all
    # the session variables as well as
    # particular call arguments in HTTP POST
    # request

    # Make HTTP request & use HTTP retcode
    # to decide whether to allow publishing
    # from this connection or not
    on_publish http://localhost:8080/publish;

    # Same with playing
    on_play http://localhost:8080/play;

    # Publish/play end (repeats on disconnect)
    on_done http://localhost:8080/done;

    # All above mentioned notifications receive
    # standard connect() arguments as well as
    # play/publish ones. If any arguments are sent
    # with GET-style syntax to play & publish
    # these are also included.
    # Example URL:
    #   rtmp://localhost/myapp/mystream?a=b&c=d
```

```
    # record 10 video keyframes (no audio) every 2 minutes
    record keyframes;
    record_path /tmp/vc;
    record_max_frames 10;
    record_interval 2m;

    # Async notify about an flv recorded
    on_record_done http://localhost:8080/record_done;

}

# HLS

# For HLS to work please create a directory in tmpfs (/tmp/hls here)
# for the fragments. The directory contents is served via HTTP (see
# http{} section in config)
#
# Incoming stream must be in H264/AAC. For iPhones use baseline H264
# profile (see ffmpeg example).
# This example creates RTMP stream from movie ready for HLS:
#
# ffmpeg -loglevel verbose -re -i movie.avi -vcodec libx264
#   -vprofile baseline -acodec libmp3lame -ar 44100 -ac 1
#   -f flv rtmp://localhost:1935/hls/movie
#
# If you need to transcode live stream use 'exec' feature.
#
application hls {
    live on;
    hls on;
    hls_path /tmp/hls;
}
```

```
        # MPEG-DASH is similar to HLS

        application dash {
            live on;
            dash on;
            dash_path /tmp/dash;
        }
    }
}

# HTTP can be used for accessing RTMP stats
http {

    server {

        listen      8080;

        # This URL provides RTMP statistics in XML
        location /stat {
            rtmp_stat all;

            # Use this stylesheet to view XML as web page
            # in browser
            rtmp_stat_stylesheet stat.xsl;
        }

        location /stat.xsl {
            # XML stylesheet to view RTMP stats.
            # Copy stat.xsl wherever you want
            # and put the full directory path here
            root /path/to/stat.xsl/;
        }
    }
}
```

```
    location /hls {
        # Serve HLS fragments
        types {
            application/vnd.apple.mpegurl m3u8;
            video/mp2t ts;
        }
        root /tmp;
        add_header Cache-Control no-cache;
    }

    location /dash {
        # Serve DASH fragments
        root /tmp;
        add_header Cache-Control no-cache;
    }
}
}
```

## Multi-worker streaming example

```
rtmp_auto_push on;

rtmp {
    server {
        listen 1935;

        application mytv {
            live on;
        }
    }
}
```

---

© 2017 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)



[Contact GitHub](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)