# 2-SRS 4.0配置支持WebRTC推拉流

- 0 SRS 4.0流媒体服务器入门系列
- 1 启动支持rtc的srs服务

异常处理 局域网webrtc可以拉流,正式环境黑屏

- 2 WebRTC拉流演示
  - 2.1 推送RTMP视频
  - 2.2 WebRTC拉流播放
- 3 WebRTC推流演示
  - 3.1 WebRTC推流
  - 3.2 WebRTC拉流播放
  - 3.3 ffplay和srs播放器拉流
- 4 Web端的srs rtc推拉流sdk
  - 4.1 推流
  - 4.2 拉流
  - 4.2 更进一步阅读

#### 版权归零声学院所有、侵权必究

音视频高级教程 - Darren老师: QQ326873713

课程链接: https://ke.qq.com/course/468797?tuin=137bb271

网页版本: https://www.yuque.com/docs/share/3415d552-e619-4c95-94b5-9182c3dd90f7?#

《2-SRS 4.0配置支持WebRTC推拉流》

## 0 SRS 4.0流媒体服务器入门系列

结合SRS官方Wiki以及本人对SRS的理解,推出《SRS 4.0流媒体服务器入门系列》,包括内容:

- 1. SRS 4.0 开发环境搭建
- 2. SRS 4.0 配置支持WebRTC推拉流
- 3. SRS 4.0 RTMP推拉流转发原理,包括延迟分析
- 4. SRS 4.0 支持WebRTC一对一通话,包括信令原理讲解(待录制)
- 5. SRS 4.0 支持WebRTC多人通话,包括信令原理讲解(待录制)

- 6. SRS 4.0 RTMP to WebRTC原理分析(待录制)
- 7. SRS 4.0 WebRTC to RTMP 原理分析(待录制)
- 8. SRS 4.0 配置支持GB28181推流(待录制)
- 9. ....其他待规划

每篇文章配合对应的视频,本文档配套视频: 2-SRS 4.0 配置支持WebRTC推拉流,大家可以扫码关注"零声学院"公众号,获取后续的更新。



零声学院

#### 云服务器: 阿里云Ubuntu 16.04

服务器: SRS(Simple Realtime Server, 支持RTMP、HTTP-FLV、HLS、WebRTC)

推流端: ffmpeg + OBS

拉流端: ffplay +VLC + srs播放器

SRS官网: http://www.ossrs.net/

微信官方公众号 srs-server



SRS开源服务器

微信扫描二维码, 关注我的公众号

# 1 启动支持rtc的srs服务

webrtc是默认支持的(--rtc=on),所以我们不需要重新编译:

我们直接使用默认的rtc配置(conf/rtc2rtmp.conf)跑起来:
cd srs.4.0/trunkconf/rtc2rtmp.conf
默认rtc2rtmp.conf配置如下

```
1 listen
                      1935;
2 max connections
                      1000;
3 daemon
                     off;
4 srs_log_tank
                     console;
6 http_server {
      enabled
                      on;
      listen
                      8080;
9
      dir
                      ./objs/nginx/html;
10 }
11
12 http_api {
13 enabled
                     on;
14 listen
                     1985;
15 }
16 stats {
17 network
                     0;
18 }
19 rtc_server {
20 enabled on;
21 # Listen at udp://8000
     listen 8000;
22
23
^{24} # The $CANDIDATE means fetch from env, if not configed, use *
  as default.
25
# The * means retrieving server IP automatically, from all ne
  twork interfaces,
      # @see https://github.com/ossrs/srs/wiki/v4_CN_RTCWiki#config
  -candidate
       #拉取流地址:使用本机地址或如下配置
```

```
#candidate $CANDIDATE;
    candidate 114.215.169.66; #设置为公网的地址
31 }
32
33 vhost __defaultVhost__ {
     rtc {
34
35
         enabled on;
        rtc_to_rtmp on; # 支持RTC推流, RTMP拉流
37
     }
38 http_remux {
         enabled on;
39
40
                  [vhost]/[app]/[stream].flv;
         mount
41
     }
42 }
```

一定要把对应的端口服务打开。

### 异常处理 局域网webrtc可以拉流,正式环境黑屏

原因

```
1 rtc_server {
2   enabled     on;
3   listen     8000;
4   candidate     $CANDIDATE;
5 }
```

rtc的**\$CANDIDATE**在云服务器一般默认会选中172(之类)的局域网络,若是推流地址和拉流地址不在一个局域中会异常,解决方法设置为公网ip

```
1 rtc_server {
2    enabled    on;
3    listen    8000;
4    candidate    114.215.xxx.xxx;# 设置为公网的地址
5 }
```

#### 启动后,可以看到rtc监听的端口信息

```
p89o][||] SRS/4.0.84 is not
[2021-04-22 16:31:00.766][Trace][13068][r5j3p89o] st_init success, use select 目前还不是稳定版本
[2021-04-22 16:31:00.767][Trace][13068][r5j3p89o] fingerprint=F6:36:7F:1D:C4:42:95:8C:04:26:02:32:2B:48:7B:7
E:BA:BA:7B:27:EF:F8:59:54:EA:94:20:4D:89:57:5B:CA
[2021-04-22 16:31:00.767][Trace][13068][r5j3p89o] RTC: Object cache init, rtp-cache=(enabled:1,pkt:64m-31w,p
ayload:16m-69w-41w), msg-cache=(enabled:1,obj:16m-41w,buf:512m-34w)
[2021-04-22 16:31:00.767][Trace][13068][r5j3p89o] http flv live stream, vhost=_ defaultVhost__, mount=[vhost
]/[app]/[stream].flv
[2021-04-22 16:31:00.767][Trace][13068][r5j3p89o] http: root mount to ./objs/nginx/html
[2021-04-22 16:31:00.767][Trace][13068][r5j3p89o] server main cid=r5j3p89o, pid=13068, ppid=12596, asprocess
[2021-04-22 16:31:00.767][Trace][13068][r5j3p89o] write pid=13068 to ./objs/srs.pid success!
[2021-04-22 16:31:00.767][Trace][13068][r5j3p89o] RTMP listen at tcp://0.0.0.0:1935, fd=6
[2021-04-22 16:31:00.767][Trace][13068][r5j3p89o] HTTP-API listen at tcp://0.0.0.0:1985, fd=7
[2021-04-22 16:31:00.767][Trace][13068][r5j3p890] HTTP-Server listen at tcp://0.0.0.0:8080, fd=8
[2021-04-22 16:31:00.767][Trace][13068][r5j3p89o] signal installed, reload=1, reopen=10, fast_quit=15, grace
quit=3
[2021-04-22 16:31:00.767][Trace][13068][r5j3p89o] http: api mount /console to ./objs/nginx/html/console
[2021-04-22 16:31:00.768][Trace][13068][r5j3p89o] rtc listen at udp://0.0.0.0:8000, fd=9 [2021-04-22 16:31:00.768][Trace][13068][r5j3p89o] Hybrid cpu=0.00%,0MB
[2021-04-22 16:31:00.768][Trace][13068][j1004634] TCP: connection manager run, conns=0
[2021-04-22 16:31:00.768][Trace][13068][v69h65s7] UDP #9 LISTEN at 0.0.0.0:8000, S0_SNDBUF(default=212992, e
xpect=10485760, actual=425984, r0=0), SO_RCVBUF(default=212992, expect=10485760, actual=425984, r0=0)
[2021-04-22 16:31:00.768][Trace][13068][i8405g81] RTC: connection manager run, conns=0
[2021-04-22 16:31:05.790][Trace][13068][r5j3p89o] Hybrid cpu=0.00%,12MB
2021-04-22 16:31:10.813][Trace][13068][r5j3p89o] Hybrid cpu=0.33%,12MB, cid=1,1, timer=53,0,0, clock=0,49,0
0,0,0,0,0,0
```

## 2 WebRTC拉流演示

我们通过RTMP进行推流,然后通过WebRTC进行拉流。

### 2.1 推送RTMP视频

这里采用ffmpeg命令进行推流

```
1 ffmpeg -re -i time.flv -vcodec copy -acodec copy -f flv -y rtmp ://114.215.169.66/live/livestream
```

### 2.2 WebRTC拉流播放

推送流成功之后,使用srs自带的rtc\_player播放器进行播放,直接请求srs服务的8080端口即可

http://114.215.169.66:8080/players/rtc\_player.html



## 3 WebRTC推流演示

### 3.1 WebRTC推流

http://114.215.169.66:8080/players/rtc\_publisher.html

因为我们现在使用使用ip地址进行测试,没有使用https+域名的方式,所以在使用WebRTC时需要修改Chrome的启动参数。

在使用Chrome浏览器推流时会报: TypeError: Cannot read property 'getUserMedia' of undefined 错误,这个错误主要是https证书问题。

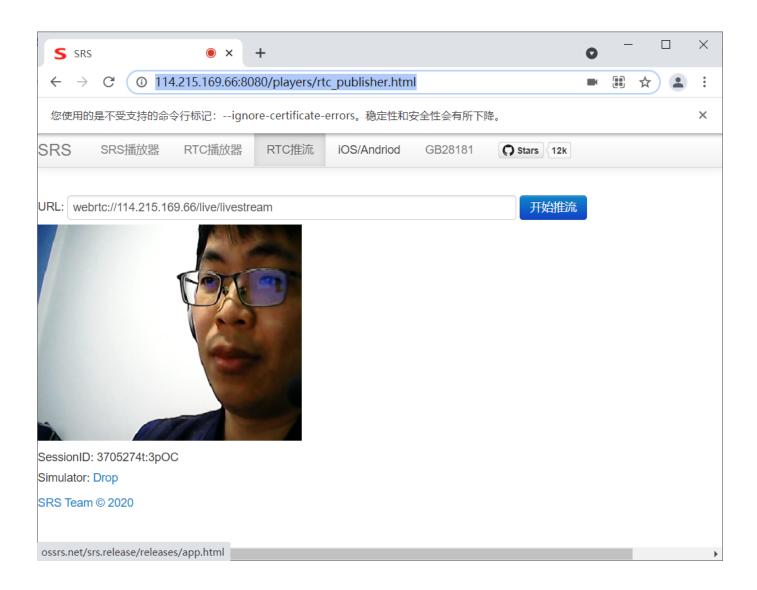
解决办法:先把chrome完全退出,右击桌面上chrome的快捷键,点击属性,在目标一栏添加如下内容,记着有个英文空格,IP换成自己的。

在目标加上以下参数(IP地址换成自己的):

1 --ignore-certificate-errors --allow-running-insecure-content --un
safely-treat-insecure-origin-as-secure="http://114.215.169.66:808"



然后重新打开chrome,输入自己的地址。

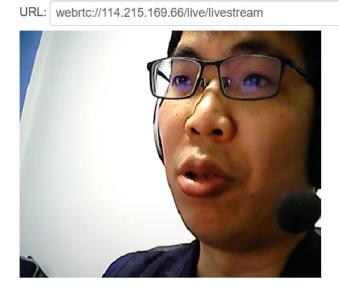


需要使用http://114.215.169.66:8080/players/rtc\_player.html去拉流。当前RTC推流还不支持RTMP/HTTP-FLV/HLS拉流。

## 3.2 WebRTC拉流播放

http://114.215.169.66:8080/players/rtc\_player.html





SessionID: 142z06sw:fsHE

Simulator: Drop SRS Team © 2020

# 3.3 ffplay和srs播放器拉流

ffplay rtmp://114.215.169.66/live/livestream

和RTMP推流时的播放是一样的。

SRS播放器同理。

# 4 Web端的srs rtc推拉流sdk

对应的js文件为 srs.sdk.js

### 4.1 推流

播放视频

```
Elements
                                                                                                                               X
                       Console
                                  Sources
                                           Network
                                                      Performance
                                                                    Memory
                                                                              Application
                                                                                           Security
                                                                                                       Lighthouse
  Page Filesystem Overrides >>>
                                                rtc_publisher.html
                                                                       jquery-1.10.2.min.js
                                                                                                            srs.sdk.js × >>
                                                                                                                              4
                                                                                            winlin.utility.is
                                                                                         rubt tsher.
 ▼ 🗖 top
                                                   29 function SrsRtcPublisherAsync()
                                                   30
                                                          var self = {};
   ▼ △ 114.215.169.66:8080
    ▼ players
                                                          // @see https://github.com/rtcdn/rtcdn-draft
                                                   32
      ► CSS
                                                   33
                                                          // @url The WebRTC url to play with, for example:
                                                   34
                                                                  webrtc://r.ossrs.net/live/livestream
      ▼ is
                                                   35
                                                          // or specifies the API port:
           adapter-7.4.0.min.js
                                                   36
                                                                  webrtc://r.ossrs.net:11985/live/livestream
                                                   37
                                                          // or autostart the publish:
           jquery-1.10.2.min.js
                                                   38
                                                          //
                                                                  webrtc://r.ossrs.net/live/livestream?autostart=true
             srs.page.js
                                                   39
                                                          // or change the app from live to myapp:
             srs.sdk.js
                                                   40
                                                                  webrtc://r.ossrs.net:11985/myapp/livestream
                                                   41
                                                          // or change the stream from livestream to mystream:
            winlin.utility.js
                                                   42
                                                                  webrtc://r.ossrs.net:11985/live/mystream
         rtc_publisher.html
                                                   43
                                                          // or set the api server to myapi.domain.com:
                                                   44
   ▶ △ img.shields.io
                                                                 webrtc://mvapi.domain.com/live/livestream
                                                   45
                                                          // or set the candidate(ip) of answer:
   ▶ △ ossrs.net
                                                   46
                                                                  webrtc://r.ossrs.net/live/livestream?eip=39.107.238.185
                                                   47
                                                          // or force to access https API:
                                                                  we brtc://r.ossrs.net/live/livestream?schema=https
                                                   48
                                                   49
                                                          // or use plaintext, without SRTP:
                                                                 webrtc://r.ossrs.net/live/livestream?encrypt=false
                                                   50
                                                   51
                                                          // or any other information, will pass-by in the query:
                                                   52
                                                                  webrtc://r.ossrs.net/live/livestream?vhost=xxx
                                                   53
                                                          //
                                                                  webrtc://r.ossrs.net/live/livestream?token=xxx
                                                   54
                                                          self.publish = async function (url) {
                                                   55
                                                              var conf = self.__internal.prepareUrl(url);
                                                   56
                                                              self.pc.addTransceiver("audio", {direction: "sendonly"});
                                                              self.pc.addTransceiver("video", {direction: "sendonly"});
                                                   57
                                                   58
                                                   59
                                                              var stream = await navigator.mediaDevices.getUserMedia(
                                                   60
                                                                  {audio: true, video: {width: {max: 320}}}}
                                                   61
                                                   62
                                                              // @see https://developer.mozilla.org/en-US/docs/Web/API/RTCPeerC
                                                   63
                                                              stream.getTracks().forEach(function (track) {
                                                   64
                                                                  self.pc.addTrack(track);
                                                   65
                                                              });
                                                   66
                                                   67
                                                              var offer = await self.pc.createOffer();
                                                              await self.pc.setLocalDescription(offer);
                                                   69
                                                              var session = await new Promise(function (resolve, reject) {
                                                   70
                                                                  // @see https://github.com/rtcdn/rtcdn-draft
                      67
                                 var offer = await self.pc.createOffer();
114.215.169.66:8080
                      68
                                 await self.pc.setLocalDescription(offer);
players
                      69
                                 var session = await new Promise(function (resolve, reject) {    resolve = f (), reject = f ()
                                     70
 ► CSS
                      71
 ▼ is
                      72
                                         api: conf.apiUrl, tid: conf.tid, streamurl: conf.streamUrl,
                      73
     adapter-7.4.0
                                         clientip: null, sdp: offer.sdp
      guery-1.10.2.
                                     console.log("Generated offer: ", data); data = {api: "http://114.215.169.66:1985/rtc/v1/publi
                      75
       srs.page.js
                                     ▶$.□ajax({
       srs.sdk.js
                                         type: "POST", url: conf.apiUrl, data: JSON.stringify(data),
contentType: 'application/json', dataType: 'json'
      winlin.utility.js
                      79
                                     }).done(function (data) {
   rtc_publisher.hti
                      80
                      81
                                         console.log("Got answer: ", data);
ima shields in
推流信令对应的URL: http://114.215.169.66:1985/rtc/v1/publish/
推流信令对应的data:
```

```
{ "api": "http://114.215.169.66:1985/rtc/v1/publish/",
   "tid": "1793820032d",
   "streamurl": "webrtc://114.215.169.66/live/livestream",
   "clientip": null,
```

```
"sdp": "v=0\no=- 170307602475242460 2 IN IP4 127.0.0.1\ns=-\nt=0 0\na=group:BUNDLE 0..\n" }
```

### 4.2 拉流

```
\dot{\Box}
              Elements
                                                 Network
                                                            Performance
                                                                            Memory
                          Console
                                      Sources
                                                                                       Application
                                                                                                     Security
                                                                                                                  Lic
  Page
                      Overrides
                                            | ■
                                                 rtc_publisher.html
                                                                      jquery-1.10.2.min.js
          Filesystem
                                                                                            winlin.utility.js
                                                                                                              srs.pac
                                                       self.pc = new RTCPeerConnection(null);
                                              254
 ▼ □ top
                                              255
   ▼ △ 114.215.169.66:8080
                                              256
                                                       return self;
                                              257 }
     players
                                              258
       ► CSS
                                              259 // Depends on adapter-7.4.0.min.js from https://github.com,
       ▼ js
                                              260 // Async-await-promise based SRS RTC Player.
                                              261 function SrsRtcPlayerAsync() {
              adapter-7.4.0.min.js
                                              262
                                                       var self = \{\};
             jquery-1.10.2.min.js
                                              263
              srs.page.is
                                              264
                                                       // @see https://github.com/rtcdn/rtcdn-draft
                                              265
                                                       // @url The WebRTC url to play with, for example:
               srs.sdk.js
                                              266
                                                                webrtc://r.ossrs.net/live/livestream
              winlin.utility.js
                                              267
                                                       // or specifies the API port:
                                                                webrtc://r.ossrs.net:11985/live/livestream
          rtc_publisher.html
                                              268
                                                       //
                                              269
                                                       // or autostart the play:
   ▶ △ img.shields.io
                                              270
                                                                webrtc://r.ossrs.net/live/livestream?autostart
   ▶ △ ossrs.net
                                              271
                                                       // or change the app from live to myapp:
                                              272
                                                                webrtc://r.ossrs.net:11985/myapp/livestream
                                              273
                                                       // or change the stream from livestream to mystream:
                                                       //
                                              274
                                                                webrtc://r.ossrs.net:11985/live/mystream
                                              275
                                                       // or set the api server to myapi.domain.com:
                              self.pc.addTransceiver("audio", {direction: "recvonly"});
self.pc.addTransceiver("video", {direction: "recvonly"});
 ▼ 📄 js
                   288
                   289
    adapter-7.4.0
                   290
    jquery-1.10.2.
                              var offer = await self.pc.createOffer();
                   291
                   292
                               await self.pc.setLocalDescription(offer);
       srs.page.js
                   293
                               var session = await new Promise(function(resolve, reject) { resolve = f (), reject = f ()
     srs.sdk.js
                   294
                                     Osee https://githuh.com/rtcdn/rtcdn-draf
     winlin.utility.js
                                  var data = { data = {api: "http://114.215.169.66:1985/rtc/v1/play/", tid: "17938243c61", streamu
                   295
                   296
                                      api: conf.apiUrl, tid: conf.tid, streamurl: conf.streamUrl,
  rtc_player.html
                                      clientip: null, sdp: offer.sdp
                   297
3 img.shields.io
                   298

→ ossrs.net

                   299
                                  300
                                  bajax({
    type: "POST", url: conf.apiUrl, data: JSON.stringify(data),
    contentType:'application/json', dataType: 'json'
                   301
                    302
                   303
                    304
                                  }).done(function(data) {
                   305
                                      console.log("Got answer: ", data);
                    306
                                      if (data.code) {
                    307
                                          reject(data); return;
拉流信令
    "api": "http://114.215.169.66:1985/rtc/v1/play/",
   "tid": "17938243c61",
   "streamurl": "webrtc://114.215.169.66/live/livestream",
   "clientip": null,
   "sdp": "v=0\no=- 8914052225280550591 2 IN IP4 127.0.0.1\ns=-\nt=0
0\na=group:BUNDLE... \n"
```

}

# 4.2 更进一步阅读

更详细的candidate讲解请阅读官方Wiki: https://github.com/ossrs/srs/wiki/v4\_CN\_WebRTC