

SRS

SRS

SRS

SRS6.0: H.265

SRS

1SRS

1.1 Docker

1.2 Linux

1.3

1.4 K8s

2SRS

SRS

1SRS

SRS

1SRS + FFmpeg

SRS

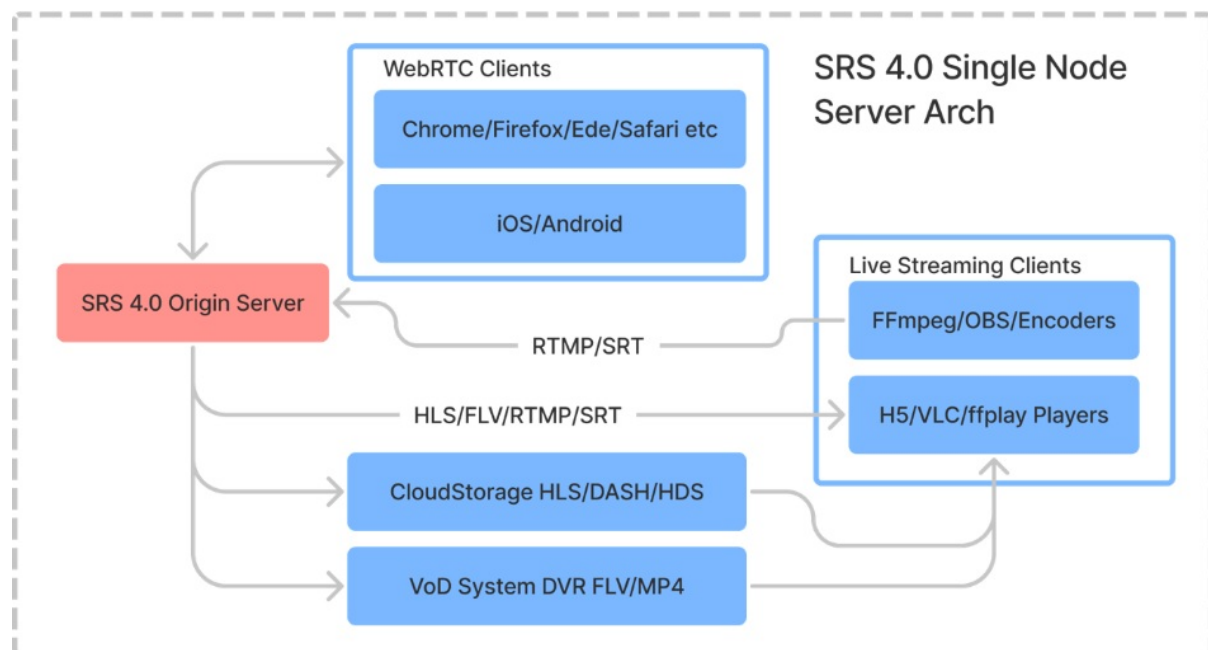
SRS(Simple Realtime Server)RTMPWebRTC HLSHTTP-FLVSRT

<https://ossrs.net/its/zh-cn/>

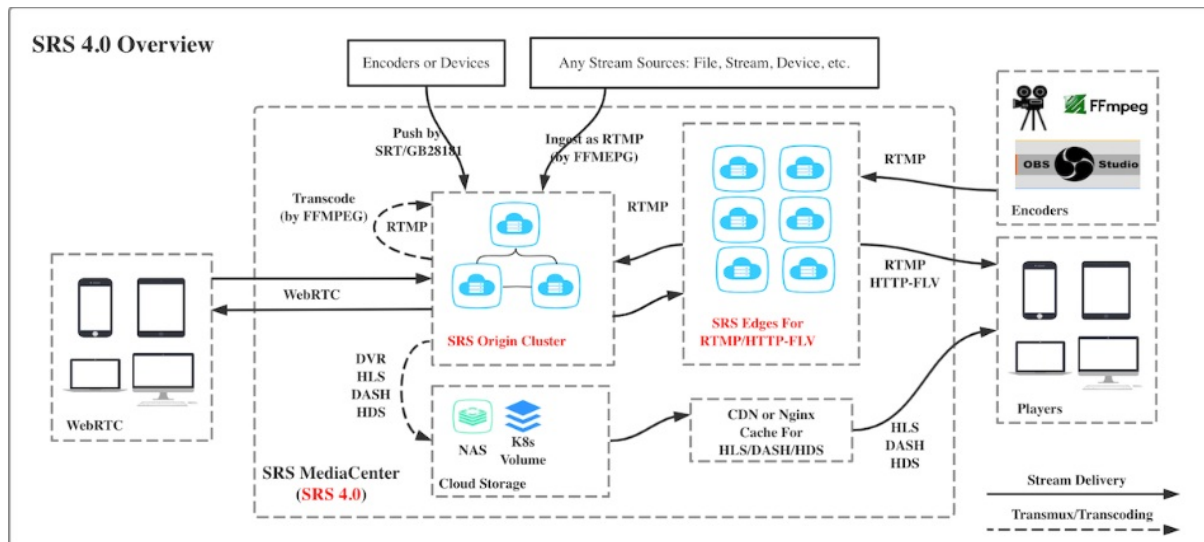
SRS Docker K8s SRS

SRS RTMP HLS WebRTC HTTP-FLV SRT

SRS 2~3



Note: [figma](#)



Note:

SRSWiki WikiIssue

<https://ossrs.net/lts/zh-cn/docs/v4/doc/introduction>

SRS

SRSStarWebRTCSRSSRSSRSSRS

https://mp.weixin.qq.com/s?__biz=MzIzMjY3MjYyOA==&mid=2247517482&idx=1&sn=533a042681c893ef365d34110ac198c4&chksm=e8939931dfe410273a1b4d750246b7de0096d55c76ddc88a7e50727f12cfae81153365caf02c&scene=21#wechat_redirect

SRS

<https://mp.weixin.qq.com/s/r2jn1GAchE08leTW32OyuQ>

SRS6.0: H.265

<https://mp.weixin.qq.com/s/uCvVswl0JSe0WMcMu9pG1A>

SRS6.0HEVC(H.265)RTMPFLVSRTTSHLSDASHGB28181WebRTC(Safari)
DVR FLVDVR MP4WordPress SrsPlayer

Why Important?

H.265H.264N10H.265H.264

H.265

8KH.2659MbpsH.26430Mbps8K

H.265

H.265H.264H.264

H.265H.265

Status of H.265

H.265Commit

Patch

- • FFmpegSRT
- • OBSSRT
- • GB28181
- • FFmpegRTMPPatch
- • SafariWebRTC
- • Chrome/FirefoxWebRTC
- • OBSRTMP

FFmpeg/ffplay

- • FFmpegHTTP-TS

- • FFmpegHLS
- • FFmpegMPEG-DASH
- • FFmpegSRT
- • ffmpegHTTP-TS
- • ffmpegHLS
- • ffmpegMPEG-DASH
- • ffmpegSRT
- • FFmpegRTMP
- • FFmpegHTTP-FLV
- • ffmpegRTMP
- • ffmpegHTTP-FLV

H5MSE

- • ChromeHTTP-TSSRSmpegts.js
- • ChromeHTTP-FLVSRSmpegts.js
- • SafariWebRTC
- • Chrome hls.jsHLSMSEhls.jsHLSfMP4
- • Chrome dash.jsDASHMSEdash.jsDASHfMP4
- • Chrome/FirefoxWebRTCChromeAV1

VLCTSMF4

- • VLCHTTP-TS
- • VLCSRT
- • VLCHLS

- • VLCMPEG-DASH

- • VLCRTMP

- • VLCHTTP-FLV

HEVC

- • FLV/MP4FLVMP4

- • HEVCHTTP API

- • HEVC

- • SRSFFmpegFFmpegSRSFFmpeg

- • WordPress plugin SrsPlayerHEVCSRS4.06.0

- • Update srs-cloud for HEVC.

- • Edge server supports publish HEVC stream to origin.

- • Edge server supports play HEVC stream from origin.

- • HTTP Callback takes HEVC metadata.

- • Prometheus Exporter supports HEVC metadata.

- • Improve coverage for HEVC.

- • Supports benchmark for HEVC by srs-bench.

iOSAndroid NativeFFmpeg

Chrome 105 MSEHEVC

Usage: Live

H.265

SRSHEVC(H.265)SRS 6.0.31+

```
git checkout develop./configure --h265=on && make
```

SRSSRTHTTP-FLVHLS


```
env SRS_LISTEN=1935 SRS_DAEMON=off SRS_LOG_TANK=console \ SRS_SRT_SERVER_ENABLED=on SRS_VHOST_SRT_ENABLED=on SRS_VHOST_SRT_TO_RTMP=on \ SRS_HTTP_SERVER_ENABLED=on SRS_VHOST_HTTP_REMUX_ENABLED=on \ SRS_VHOST_HTTP_REMUX_MOUNT=[vhost]/[app]/[stream].flv SRS_VHOST_HLS_ENABLED=on \ ./objs/srs -e
```

FFmpegSRTHEVC

```
ffmpeg -stream_loop -1 -re -i doc/source.flv -acodec copy -vcodec libx265 \ -pes_payload_size 0 -f mpegts 'srt://127.0.0.1:10080?streamid=#!::r=live/livestream,m=publish'
```

HTTP-FLVffplayVLCHLS

- <http://localhost:8080/live/livestream.flv>
- <http://localhost:8080/live/livestream.m3u8>

Note: H5HLSHls.jsH5HTTP-FLVHTTP-TSmpegts.js

SRS

Usage: WebRTC

WebRTC H.265SafariHEVCChromeFirefox

Safari > > WebRTC H265 codec

#465Safari WebRTC

WebRTCAV1Safari/Chrome/Firefox#1070MSEAV1

Note: Media Source Extensions (MSE)fMP4H5 videompegts.jsHls.jsdash.jsMSE
MDN: MSE

FFmpeg Patch

FFmpeg/ffplayHEVC over RTMP/HTTP-FLVFFmpegPatch

- (runner365)FFmpeg 4/5/6patchffmpeg_rtmp_h265SRSPatch
- Intel 0001-Add-SVT-HEVC-FLV-support-on-FFmpeg.patch
- FLVspecificationusage

SRSPatchFFmpegffplayffprobeSRS Docker

```
# For macOSdocker run --rm -it ossrs/srs:encoder ffmpeg -stream_loop -1
-re -i doc/source.flv \ -acodec copy -vcodec libx265 -f flv rtmp://host
.docker.internal/live/livestream# For linuxdocker run --net=host --rm -i
t ossrs/srs:encoder ffmpeg -stream_loop -1 -re -i doc/source.flv \ -aco
dec copy -vcodec libx265 -f flv rtmp://127.0.0.1/live/livestream
```

Note: Dockerfile

FFmpeg Tools

Known Issues

1. Safari HEVC WebRTCWebRTCSRS
2. Chrome/Firefox WebRTCHEVC
3. MSEiOSHEVC MSE
4. H5mpegs.jsHls.jsdash.js

HEVC

Thanks

H.265

- runner365 PRRTMPHLSSRT [265](#)
- yinjiaoyuan [265](#)bugGB28181 [265](#)Patch
- PieerePi [265](#)bug
- qichaoshen82 [265](#)bug
- ZSC714725 [265](#)bug
- bluestn [MP4](#)GB28181 [265](#)
- mapengfei53 [MP4](#) [265](#)
- chundonglinlin SRT [265](#)
- duiniuluantanqin GB28181 [265](#)
- panda1986 WordPress SrsPlayer [265](#)

Note: PRSRSSRSPRco-authorCommit



mpegs.jsH5HTTP-FLVHTTP-TSH.265flv.jsStar

SRS

1SRS

1.1 Docker

DockerSRS

<https://ossrs.net/zh-cn/docs/v5/doc/getting-started>

Live Streaming

SRS

DockerSRS:

```
docker run --rm -it -p 1935:1935 -p 1985:1985 -p 8080:8080 \
  registry.cn-hangzhou.aliyuncs.com/ossrs/srs:5 ./objs/srs -c conf/docker.conf
```

Note: Release

FFmpegDocker

```
docker run --rm -it registry.cn-hangzhou.aliyuncs.com/ossrs/srs:encoder
ffmpeg -stream_loop -1 -re -i doc/source.flv \
  -c copy -f flv rtmp://host.docker.internal/live/livestream
```

FFmpeg() OBS()

```
ffmpeg -re -i ./doc/source.flv -c copy -f flv rtmp://localhost/live/livestream
```

Note: ./doc/source.flvSRS

SRSlocalhostIP:

- RTMP (by VLC): `rtmp://localhost/live/livestream`
- H5(HTTP-FLV): <http://localhost:8080/live/livestream.flv>
- H5(HLS): <http://localhost:8080/live/livestream.m3u8>

WebRTC

SRSWebRTC

DockerSRS

```
CANDIDATE="192.168.1.10"
docker run --rm -it -p 1935:1935 -p 1985:1985 -p 8080:8080 -p 1990:1990
-p 8088:8088 \
    --env CANDIDATE=$CANDIDATE -p 8000:8000/udp \
    registry.cn-hangzhou.aliyuncs.com/ossrs/srs:5 ./objs/srs -c conf/doc
ker.conf
```

Note: IPSRSIP

Note: CANDIDATEWebRTC: CANDIDATE

SRSWebRTCWebRTC: Publish

Note: localhostSRSHTTPSWebRTC using HTTPS

WebRTCWebRTC: Play

Note:

WebRTC for Live Streaming

SRS WebRTCWebRTC

DockerSRS

```
CANDIDATE="192.168.1.10"
docker run --rm -it -p 1935:1935 -p 1985:1985 -p 8080:8080 \
    --env CANDIDATE=$CANDIDATE -p 8000:8000/udp \
    registry.cn-hangzhou.aliyuncs.com/ossrs/srs:5 ./objs/srs -c conf/rtm
p2rtc.conf
```

Note: IPSRSIP

Note: CANDIDATEWebRTC: CANDIDATE

Note: RTMPWebRTCrtp2rtc.conf

FFmpegDocker

```
docker run --rm -it registry.cn-hangzhou.aliyuncs.com/ossrs/srs:encoder
ffmpeg -stream_loop -1 -re -i doc/source.flv \
    -c copy -f flv rtmp://host.docker.internal/live/livestream
```

FFmpeg() OBS()

MinDoc

```
ffmpeg -re -i ./doc/source.flv -c copy -f flv rtmp://localhost/live/livestream
```

Note: ./doc/source.flvSRS

SRSlocalhostIP:

- WebRTC: webrtc://localhost/live/livestream
- H5(HTTP-FLV): <http://localhost:8080/live/livestream.flv>
- H5(HLS): <http://localhost:8080/live/livestream.m3u8>

WebRTC using HTTPS

WebRTCSSRSWebRTCHTTPS API

NoteWebRTCHTTPSlocalhostHTTPS

DockerSRS

```
CANDIDATE="192.168.1.10"
docker run --rm -it -p 1935:1935 -p 1985:1985 -p 8080:8080 -p 1990:1990
-p 8088:8088 \
  --env CANDIDATE=$CANDIDATE -p 8000:8000/udp \
  registry.cn-hangzhou.aliyuncs.com/ossrs/srs:5 ./objs/srs -c conf/https.docker.conf
```

Note: IPSRSIP

Note: CANDIDATEWebRTC: CANDIDATE

Remark: keycert HTTPS API HTTPS Callback HTTPS Live Streaming
HTTPSSRSNginxSRS

WebRTCSSRSWebRTC: Publish

WebRTCWebRTC: Play

thisisunsafe

Note:

SRT for Live Streaming

SRSSRTSRT

DockerSRS

```
docker run --rm -it -p 1935:1935 -p 1985:1985 -p 8080:8080 -p 10080:10080/udp \
registry.cn-hangzhou.aliyuncs.com/ossrs/srs:5 ./objs/srs -c conf/srt.conf
```

FFmpeg() OBS()

```
ffmpeg -re -i ./doc/source.flv -c copy -pes_payload_size 0 -f mpegts \
'srt://127.0.0.1:10080?streamid=#!::r=live/livestream,m=publish'
```

FFmpeg() OBS()

```
ffplay 'srt://127.0.0.1:10080?streamid=#!::r=live/livestream,m=request'
```

Multiple Streams

SRSSRSURL

- `rtmp://ip/live/livestream`
- `rtmp://ip/live/livestreamN`
- `rtmp://ip/liveN/livestreamN`
- `srt://ip:10080?streamid=#!::r=anyM/streamN,m=publish`
- `webrtc://localhost/anyM/streamN`
- `http://ip:8080/anyM/streamN.flv`
- `http://ip:8080/anyM/streamN.m3u8`
- `https://ip:8080/anyM/streamN.flv`
- `https://ip:8080/anyM/streamN.m3u8`

Note: RTMP URL

1.2Linux

1.3

<https://mp.weixin.qq.com/s/nutc5eJ73aUa4Hc23DbCwQ>

bt.cn[6]SRSCentOS 7+Ubuntu 20+CentOS 7

```
yum install -y wget &&
wget -O install.sh http://download.bt.cn/install/install_6.0.sh &&
sh install.sh ed8484bec
```

Note:

Note: SRS

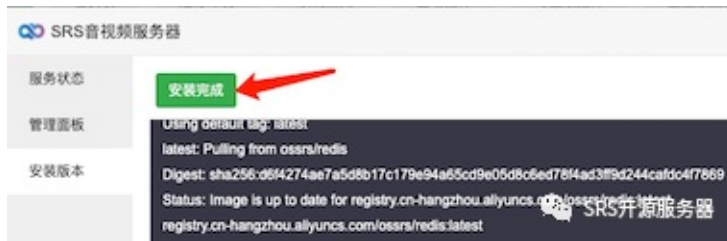
SRS5.0SRS



SRSSRSSRS



Note: SRS



SRSSRS



Note: SRSSRS[7]

SRS

1.4K8s

K8s

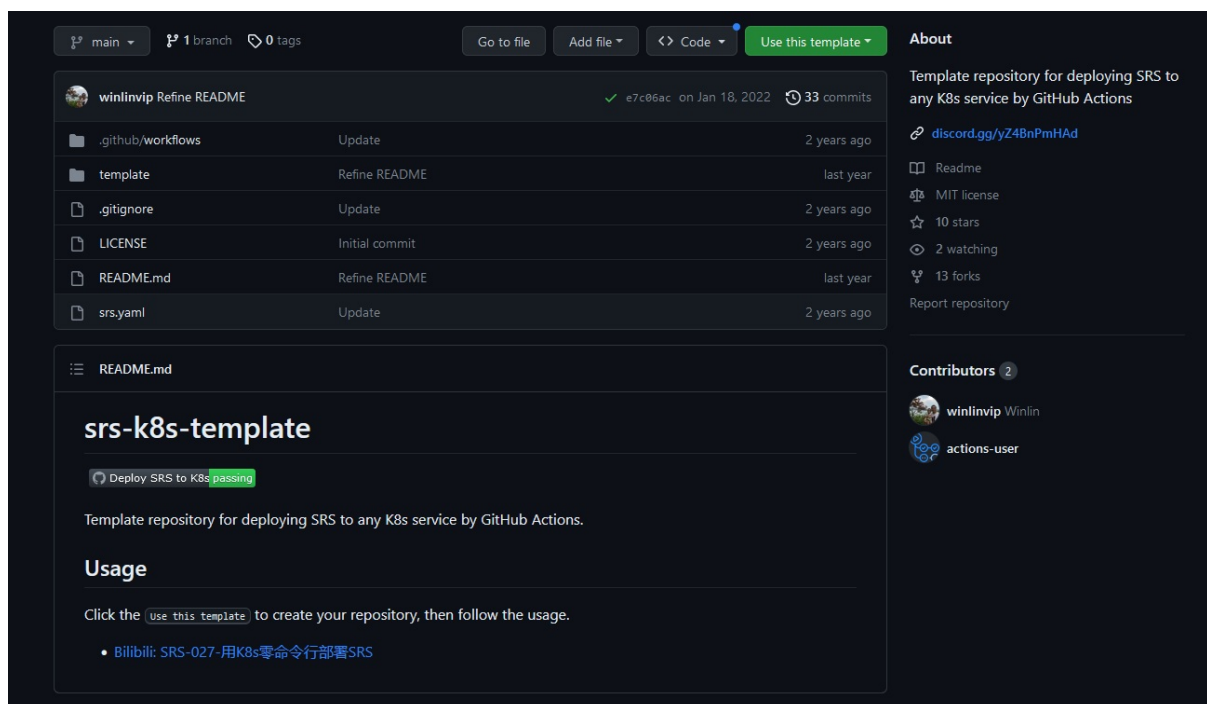
K8sSRSDeploy to Cloud PlatformsBilibili: SRS-027-K8sSRS

SRSK8s

- [TKE\(K8s\)](#)
- [K8s](#)
- [ACK\(K8s\)](#)
- [EKS\(AWS K8s\)](#)
- [AKS\(Azure K8s\)](#)

KubeSphere SRS

1 [K8s](#) GitHub



srs.yaml

```
git clone https://github.com/ossrs/srs-k8s-template.git
```

2 `srs.yaml` `k8s master`

```
#  
kubectl apply -f srs.yaml --namespace=  
# srs-live-prod  
kubectl apply -f srs.yaml --namespace=live-prod
```

3KubeSphere

2SRS

SRS

1SRS

SRS

1SRS + FFmpeg

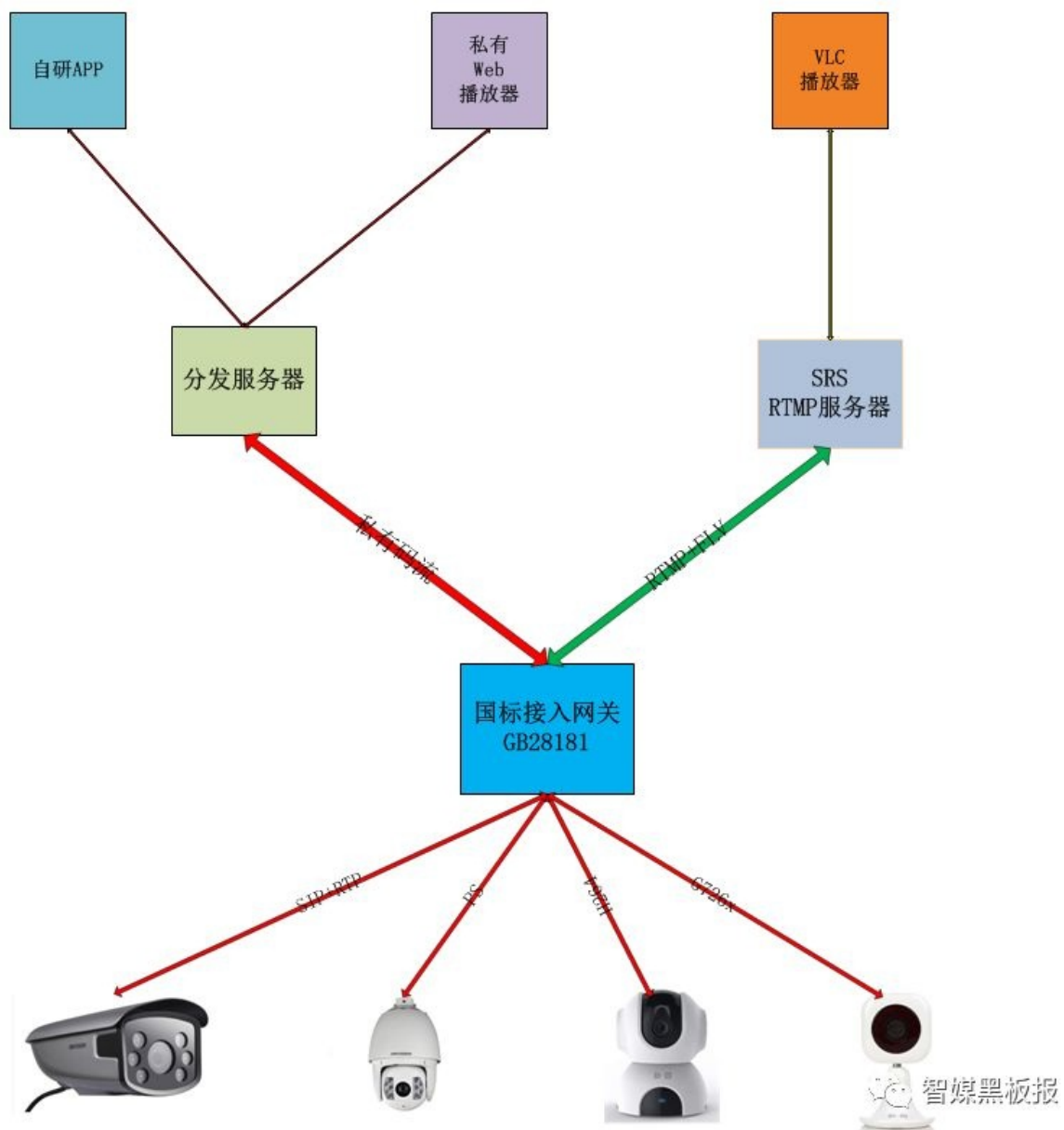
SRSFFmpeg

[https://mp.weixin.qq.com/mp/appmsgalbum?](https://mp.weixin.qq.com/mp/appmsgalbum?__biz=MzI0NTMxMjA1MQ==&action=getalbum&album_id=1574171727348170757&scene=173&from_msgid=2247484270&from_itemidx=1&count=3&nolastread=1#wechat_redirect)

[__biz=MzI0NTMxMjA1MQ==&action=getalbum&album_id=1574171727348170757&scene=173&from_msgid=2247484270&from_itemidx=1&count=3&nolastread=1#wechat_redirect](https://mp.weixin.qq.com/mp/appmsgalbum?__biz=MzI0NTMxMjA1MQ==&action=getalbum&album_id=1574171727348170757&scene=173&from_msgid=2247484270&from_itemidx=1&count=3&nolastread=1#wechat_redirect)

RTMPVLCRTMPFMSSRSwinlinux

git



- 1.
2. GBSIP
3. GBRTMP
4. SRSGBRTP->FLVSRS
- 5.

LinuxSRS:

SRSSRSGIT <https://github.com/ossrs/srs>

1. SRS

SRSlinux

A. git:

git clone <https://github.com/ossrs/srs.git>

B. Zip/usr/local/srs

```
[root@stg-cn-closetli-gb2181 trunk]# ll
total 80
drwxr-xr-x  5 root root  4096 Apr  5  2019 3rdparty
-rw-r--r--  1 root root  4380 Oct 17 20:47 Makefile
drwxr-xr-x  2 root root  4096 Apr  5  2019 auto
drwxr-xr-x  2 root root  4096 Apr  5  2019 conf
-rwxr-xr-x  1 root root 26091 Apr  5  2019 configure
drwxr-xr-x  2 root root  4096 Apr  5  2019 doc
drwxr-xr-x  3 root root  4096 Apr  5  2019 etc
drwxr-xr-x  7 root root  4096 Apr  5  2019 ide
drwxr-xr-x  2 root root  4096 Apr  5  2019 modules
drwxr-xr-x 14 root root  4096 Oct 17 20:49 objs
drwxr-xr-x 12 root root  4096 Apr  5  2019 research
drwxr-xr-x  2 root root  4096 Apr  5  2019 scripts
drwxr-xr-x  9 root root  4096 Apr  5  2019 src
```

智媒黑板报

C. trunk

./configure && make

./configure -h

D.

make

make install

E.

./objs/srs -c conf/srs.conf

F.

ps -ef|grep srs netstat -ano|grep 1935

```
[root@stg-cn-closeti-gb2181 trunk]# ps -ef|grep srs
root      8141 20338  0 15:26 pts/0    00:00:00 grep --color=auto srs
root      27836   1  0 Oct17 pts/0    00:06:20 ./objs/srs -c conf/srs.conf
[root@stg-cn-closeti-gb2181 trunk]# netstat -ano|grep 1935
tcp        0      0 0.0.0.0:1935          0.0.0.0:*              LISTEN      off (0.00/0/0)
tcp        0      0 127.0.0.1:1935        127.0.0.1:31140         ESTABLISHED keepalive (4694.67/0/0)
tcp        0      0 10.70.0.165:1935      112.17.80.218:13965     ESTABLISHED keepalive (5218.96/0/0)
tcp        0 32854 127.0.0.1:31140        127.0.0.1:1935          ESTABLISHED on (0.17/0/0)
tcp        0      0 127.0.0.1:1935        127.0.0.1:25948         ESTABLISHED keepalive (3187.34/0/0)
tcp        0      0 10.70.0.165:1935      112.17.80.218:45181     ESTABLISHED keepalive (2519.79/0/0)
tcp        0      0 127.0.0.1:25948       127.0.0.1:1935          ESTABLISHED off (0.00/0/0)
```

2.

FFmpegGBSIP RTP,

3. VLC



4. Video Player

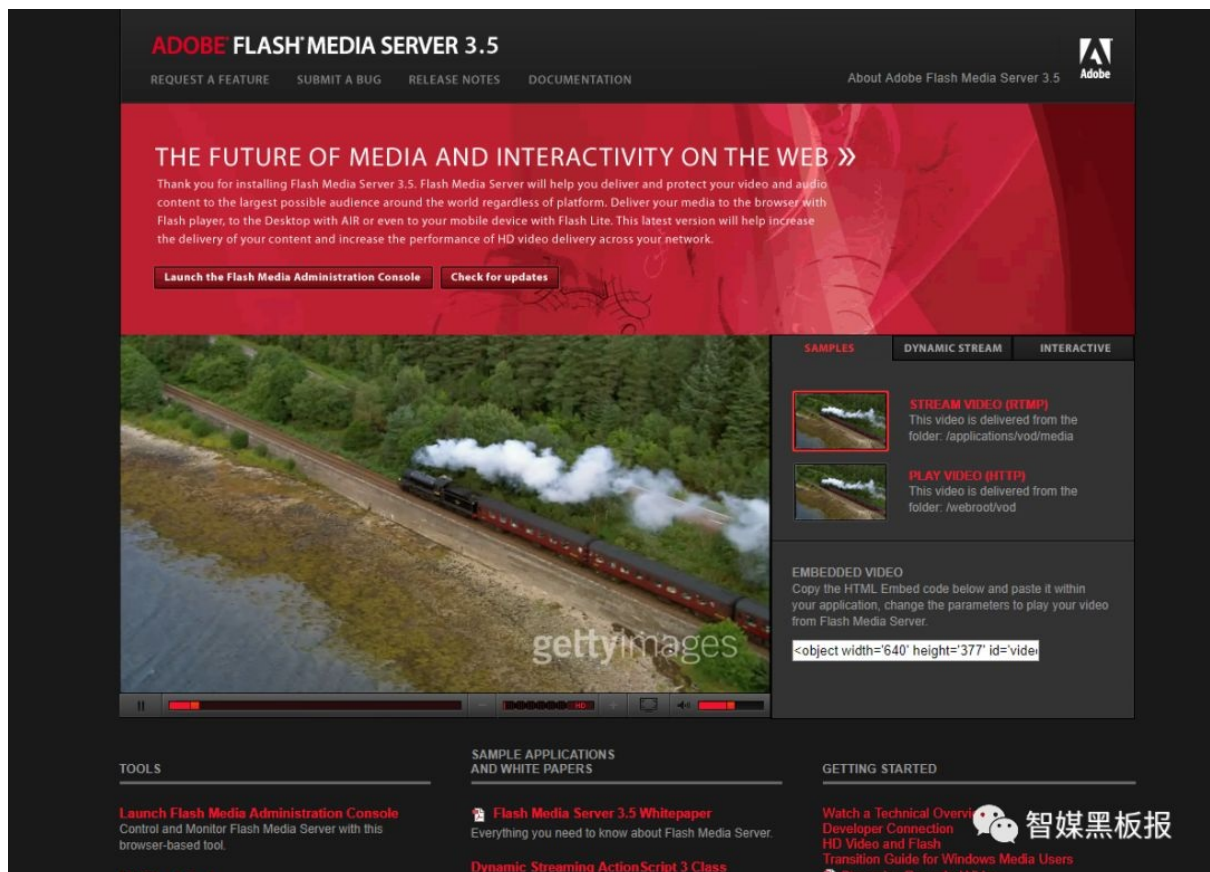


WindowsFMS

FMSAdobeRTMPWindowsFFmpegVLCVideo Palyer

1. FMS

19351935:



Flash Media Server 3.5(FMS) 1374-5568-0192-3684-3402-70241374-5232-8277-3901-0946-6727

2. FFmpeg

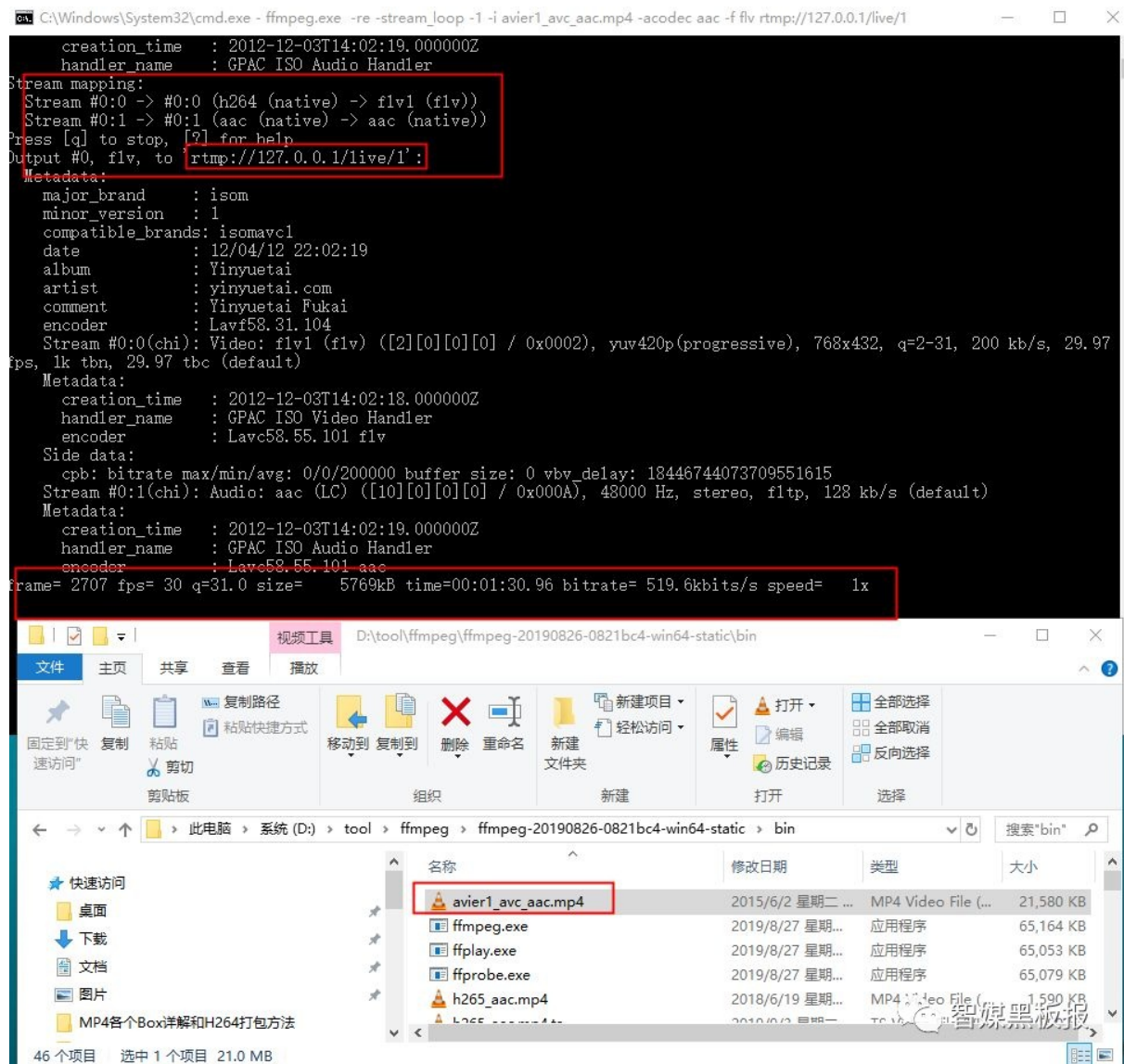
FLVMP4FFmpegFMS1935

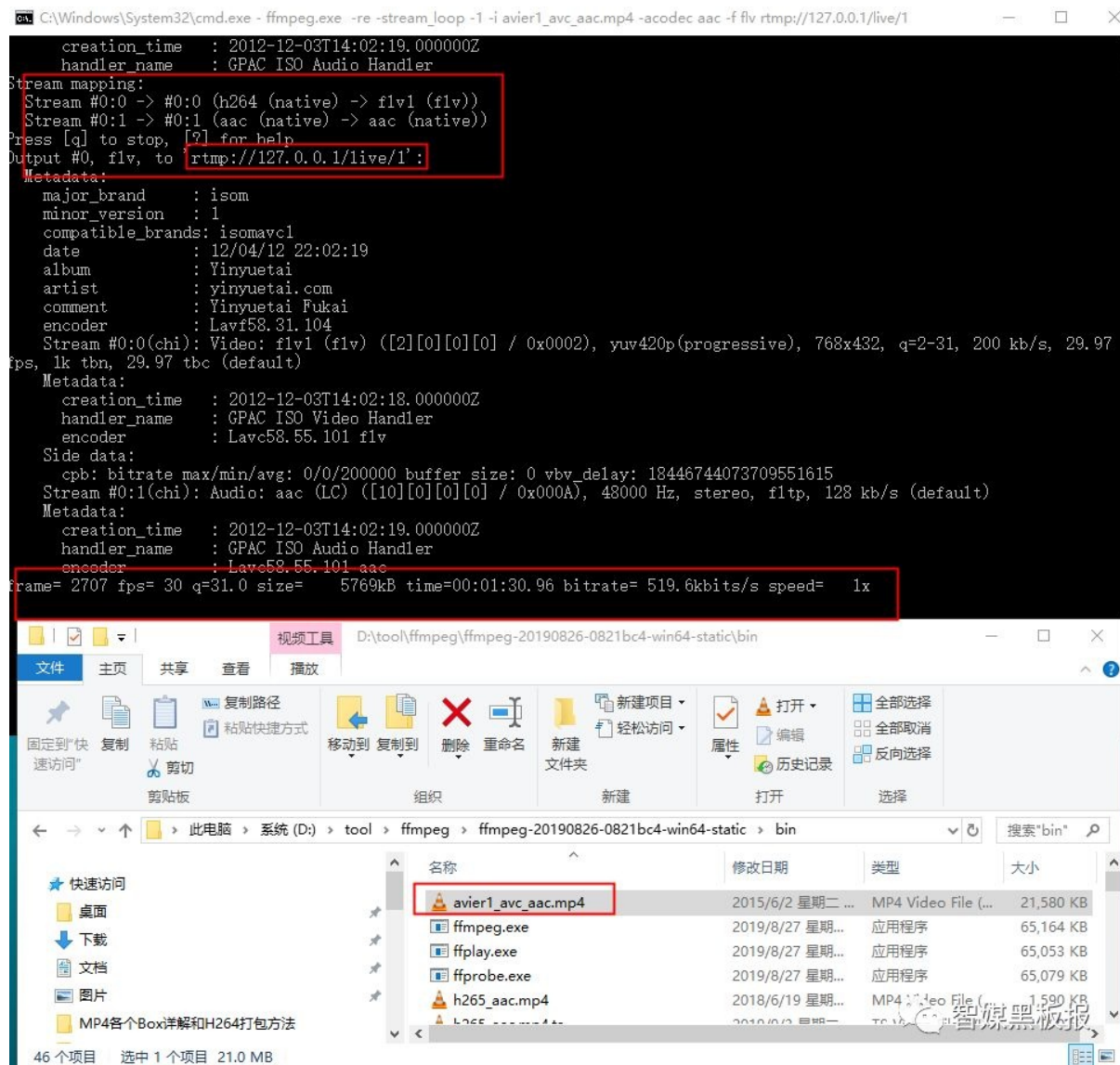
A. avier1_avc_aac.mp4FFmpeg

B.

ffmpeg.exe -re -stream_loop -1 -i avier1_avc_aac.mp4 -acodec aac -f flv rtmp://127.0.0.1:1935/live/1

FLVFMS1935



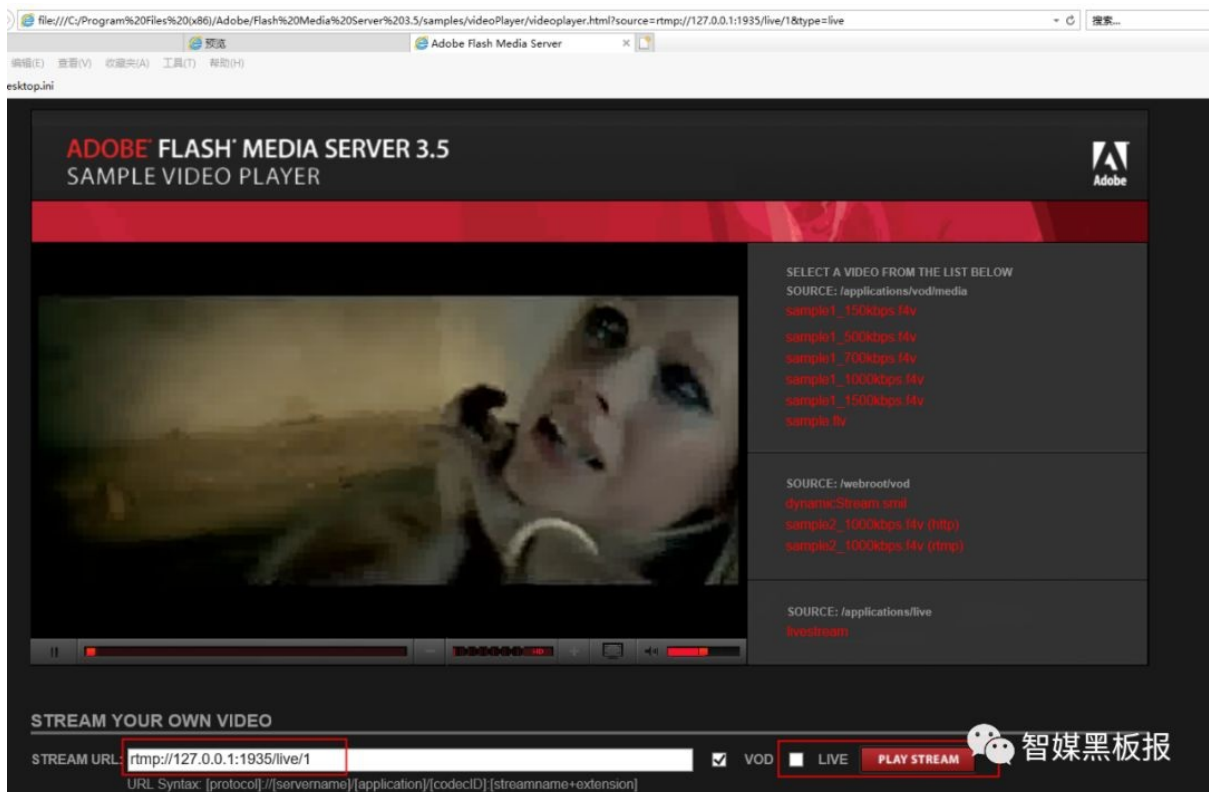


3.

A.URLVLC



B.URLVideo Player



WinLinuxFMSSRS&

<https://github.com/ty6815>

<https://github.com/ossrs/srs>

<https://www.jianshu.com/p/ce57f108fdaa>

<https://blog.csdn.net/u014359108/article/details/53114333>