1. 安装（docker安装方式）

1.1 默认安装

- 此镜像为github持续集成自动编译推送，跟代码(master分支)保持最新状态 docker run -id -p 1935:1935 -p 8080:80 -p 8443:443 -p 8554:554 -p 10000:10000 -p 10000:10000/udp -p 8000:8000/udp -p 9000:9000/udp zlmediakit/zlmediakit:master

1.2 指定配置安装

https://blog.csdn.net/u011374856/article/details/124802856

1.2.1 创建数据和配置存放目录

创建 zlmediakit 目录 mkdir -p /home/docker/zlmediakit && chmod 777 /home/docker/zlmediakit

1.2.2 安装并运行 ZLMediaKit

docker run -d -p 1935:1935 -p 8080:80 -p 8554:554
-p 10000:10000 -p 10000:10000/udp -p 8000:8000/udp
--name zlmediakit
zlmediakit/zlmediakit:master

-p 8554:554 ----> docker所安装主机8554端口映射到zlmediakit的554端口

1.2.3 复制 ZLMediaKit 相关文件

- 复制 zlmediakit 容器文件到指定目录 docker cp -a zlmediakit:/opt/media /home/docker/zlmediakit
- 删除 zlmediakit 容器 docker rm -f zlmediakit

1.2.4 编辑 ZLMediaKit 配置文件

配置文件详解及注释

- 进入 zlmediakit 配置文件目录 cd /home/docker/zlmediakit/media/conf
- 编辑单机版配置文件 vi config.ini

1.2.5 自定义启动 ZLMediaKit

- 此镜像为 zlmediakit 开发团队提供，推荐

docker run -d -p 1935:1935 -p 8080:80 -p 8554:554 \ -p 10000:10000 -p 10000:10000/udp -p 8000:8000/udp \ -p 30000-30500:30000-30500 -p 30000-30500:30000-30500/udp \ --name zlmediakit \ --restart=always \ --env MODE=standalone \ -e TZ="Asia/Shanghai" \ -v /home/docker/zlmediakit/media/bin:/opt/media/bin \ -v /home/docker/zlmediakit/media/conf:/opt/media/conf \ -v /home/hik:/home/hik \ zlmediakit/zlmediakit:master

- -v /home/hik:/home/hik 设置/index/api/addFFmpegSource接口mp4文件路径问题

1.2.6 访问地址

http://192.168.200.73:8080/

2.API接口

参考：
https://github.com/ZLMediaKit/ZLMediaKit/wiki/MediaServer%E6%94%AF%E6%8C%81%E7%9A%84HTTP-API

2.1 /index/api/addStreamProxy

功能：动态添加rtsp/rtmp/hls拉流代理(只支持H264/H265/aac/G711负载)

eg:http://192.168.200.73:8080/index/api/addStreamProxy?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&vhost=**defaultVhost**&app=504&stream=ch01&url=rtsp://admin:Grid12345@192.168.50.4:554/h264/ch01/main/av_stream

secret值为config中对应的值

返回： { "code": 0, "data": { "key": "**defaultVhost**/504/ch01" } }

转流之后使用方式： http://192.168.200.73:8080/504/ch01/hls.m3u8
http://192.168.200.73:8080/504/ch01/hls.live.flv

2.2 /index/api/addFFmpegSource

eg:http://192.168.200.73:8080/index/api/addFFmpegSource?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&src_url=/home/hik/video/test.mp4&dst_url=rtsp://192.168.200.73:8554/live/171&timeout_ms=10000

- src_url 地址必须在容器创建时候映射进去
- dst_url rtsp://192.168.200.73:8554/live/171 url中得ip，端口需为docker安装的机器的ip，已经zlmediakit的554端口映射之后的端口

2.3 /index/api/setServerConfig

eg：http://192.168.200.73:8080/index/api/setServerConfig?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&ffmpeg.cmd=%s -re -i %s -vcodec h264 -acodec libfdk_aac -f flv -r 50 %s

- ffmpeg.cmd = %s -re -i %s -vcodec h264 -f rtsp -rtsp_transport tcp %s
- 调用方法之后注意查看是否有乱码

3.播放url规则

3.1 URL的组成部分

以rtsp://somedomain.com:554/live/0?token=abcdefg&field=value为例,该url分为以下几个部分：

- 协议(scheam) : rtsp协议,默认端口554

- 虚拟主机(vhost) : somedomain.com,该字段既可以是域名也可以是ip，如果是ip则对应的虚拟主机为__defaultVhost__
- 服务端口号(port) : 554,如果不指定端口号，则使用协议默认端口号
- 应用名(app) : live
- 流ID(streamid) : 0
- 参数(args) : token=abcdefg&field=value

3.2 ZLMediaKit中的流媒体源

在ZLMediaKit中，流媒体源是一种可以被用于直播转发、推流转发等功能的数据对象，在本项目中被称作为MediaSource，目前支持5种类型的流媒体源，分别是RtspMediaSource、RtmpMediaSource、HlsMediaSource、TSMediaSource、FMP4MediaSource。

定位一个流媒体源，主要通过4个元素(我们后续称其为4元组)，分别是:

- 协议(scheam)
- 虚拟主机(vhost)
- 应用名(app)
- 流ID(streamid)
- RtspMediaSource支持 rtsp播放、rtsp推流、webrtc播放、webrtc推流。

RtmpMediaSource支持 rtmp推流/播放、http-flv播放、ws-flv播放。

HlsMediaSource支持 hls播放。

TSMediaSource 支持 http-ts播放、ws-ts播放。

FMP4MediaSource 支持 http-fmp4播放、ws-fmp4播放。

3.3 流媒体源对应的播放url

假定有一个RtspMediaSource，它的4元组分别为 rtsp(RtspMediaSource固定为rtsp)、somedomain.com、live、0 那么播放这个流媒体源的url对应为:

rtsp://somedomain.com/live/0 rtsps://somedomain.com/live/0 rtsp://127.0.0.1/live/0?vhost=somedomain.com rtsps://127.0.0.1/live/0?vhost=somedomain.com 如果有一个RtmpMediaSource，它的4元组分别为 rtmp(RtmpMediaSource固定为rtmp)、somedomain.com、live、0 那么播放这个流媒体源的url对应为:

rtmp://somedomain.com/live/0 rtmps://somedomain.com/live/0 rtmp://127.0.0.1/live/0?vhost=somedomain.com rtmps://127.0.0.1/live/0?vhost=somedomain.com rtmp类型的流媒体源也支持http-flv、websocket直播，对应的url如下：

注意: 老代码flv直播后缀为.flv,新代码才改成了.live.flv

http://somedomain.com/live/0.live.flv https://somedomain.com/live/0.live.flv http://127.0.0.1/live/0.live.flv?vhost=somedomain.com https://127.0.0.1/live/0.live.flv?vhost=somedomain.com ws://somedomain.com/live/0.live.flv wss://somedomain.com/live/0.live.flv ws://127.0.0.1/live/0.live.flv?vhost=somedomain.com wss://127.0.0.1/live/0.live.flv?vhost=somedomain.com 当然，ZLMediaKit一般会把rtsp、rtmp流媒体源互相转换，也会转换成hls/http-ts/ws-ts/http-fmp4/ws-fmp4，播放的url如下：

HLS

http://somedomain.com/live/0/hls.m3u8 https://somedomain.com/live/0/hls.m3u8 http://127.0.0.1/live/0/hls.m3u8?vhost=somedomain.com https://127.0.0.1/live/0/hls.m3u8?vhost=somedomain.com HTTP-TS/WS-TS(后缀为.live.ts,目的是为了解决与hls的冲突)

http://somedomain.com/live/0.live.ts https://somedomain.com/live/0.live.ts http://127.0.0.1/live/0.live.ts?vhost=somedomain.com https://127.0.0.1/live/0.live.ts?vhost=somedomain.com ws://somedomain.com/live/0.live.ts wss://somedomain.com/live/0.live.ts ws://127.0.0.1/live/0.live.ts?vhost=somedomain.com wss://127.0.0.1/live/0.live.ts?vhost=somedomain.com HTTP-fMP4/WS-fMP4(后缀为.live.mp4,目的是为了解决与mp4点播的冲突)

http://somedomain.com/live/0.live.mp4 https://somedomain.com/live/0.live.mp4 http://127.0.0.1/live/0.live.mp4?vhost=somedomain.com https://127.0.0.1/live/0.live.mp4?vhost=somedomain.com ws://somedomain.com/live/0.live.mp4 wss://somedomain.com/live/0.live.mp4 ws://127.0.0.1/live/0.live.mp4?vhost=somedomain.com wss://127.0.0.1/live/0.live.mp4?vhost=somedomain.com 一般而言，上述url在ZLMediaKit都有效，因为ZLMediaKit默认转换流媒体源。

3.4 点播url

ZLMediaKit的点播一般通过mp4文件来实现，推荐大家使用http mp4点播，这样是最简单，服务器也无需解复用mp4文件，当然ZLMediaKit目前也支持rtsp、rtmp、http-flv、websocket-flv的mp4点播， 对应的url跟直播url类似，不在赘述，这里只介绍区别。

ZLMediaKit对点播限制应用名，默认为record 假如一个mp4文件放置在http根目录record文件夹(www/record)下，他的相对路径为:www/record/0.mp4,那么点播url则为: rtsp://somedomain.com/record/0.mp4 rtmp://somedomain.com/record/0.mp4 http://somedomain.com/record/0.mp4(这里是通用的http文件点播，服务器不用解复用文件) http://somedomain.com/record/0.mp4.live.flv（这里是http-flv直播，不是http点播，服务器需要解复用文件） ws://somedomain.com/record/0.mp4.live.flv http://somedomain.com/record/0.mp4.live.ts（这里是http-ts直播，不是http点播，服务器需要解复用文件） ws://somedomain.com/record/0.mp4.live.ts http://somedomain.com/record/0.mp4.live.mp4（这里是http-fmp4直播，不是http点播，服务器需要解复用文件） ws://somedomain.com/record/0.mp4.live.mp4 如果开启了虚拟主机，那么点播文件需要放置在 www/somedomain.com/record/0.mp4。

3.5 webrtc推流/播放

webrtc播放跟上述方式不太一样，webrtc协议本身不定义信令交互协议，用户自己去实现sdp+icecandidate交换逻辑，所以webrtc并没有一个标准的播放器，需要自己使用js或native sdk去实现播放。

zlmediakit实现的webrtc sdp+icecandidate交换方式是http post方式，接口名为/index/api/webrtc, 该接口使用post content传递 offer sdp, 同时url query参数传递媒体源4元组中的app steam_id，由于http协议本身支持vhost，所以不需要另外指定vhost。 webrtc在zlmediakit中可以认为是rtsp协议的另外表现形式，他们推流、播放使用的数据源都相同，都是RtspMediaSource。

在webrtc推流时，交互webrtc sdp+icecandidate的http post接口类似为：http://127.0.0.1/index/api/webrtc?app=live&stream=test&type=push

在webrtc播放时，交互webrtc sdp+icecandidate的http post接口类似为：http://127.0.0.1/index/api/webrtc?app=live&stream=test&type=play。

zlmeiakit工程自带webrtc测试播放/推流器，用户启动zlmediakit后，浏览器访问http://127.0.0.1/webrtc/就可以访问之。

另外，zlmediakit也支持使用webrtc播放mp4文件，http post接口类似为：http://127.0.0.1/index/api/webrtc?app=record&stream=test.mp4&type=play。