**Car-eye JT/T 1078 media server**

**部署操作说明**

# **开放端口**

|  |  |  |
| --- | --- | --- |
| 端口 | 使用说明 | 说明 |
| 9500 | 设备上行实时流端口 | 必选 |
| 9600 | 设备上行回放流端口 | 必选 |
| 9213 | 1078 视频服务器自带的ws拉流端口 | 自带的拉流端口 |
| 9214 | 1078 视频服务器自带的wss拉流端口 | 自带的SSL拉流端口 |
| 10077 | RTMP 服务器拉流端口 | 必选 |
| 4022 | Ws/http FLV 拉流端口 | （nginx 拉流端口，根据项目选择） |
| 4023 | Wss/HTTS flv 拉流端口 | （nginx SSL拉流端口，根据项目选择） |
| 9210 | 服务器自带的web 对讲端口 | 可选 |
| 7749 | 服务器自带的通信端口 | 可选 |
| 10085 | Rtmp服务器状态检测端口 | Nginx RTMP 状态检测口 |
| 10086 | 服务器自带的状态检测端口 | 可选 |
| 9000 | 文件服务器端口 | 可选 |

注：请根据业务需要开通相应的端口, 比如不需要客户端对讲的就不需要配置CS客户端对讲端口，nginx rtmp 拉流和1078 视频服务器拉流通常只需要一个就好了。

# Windows 部署说明

## Windows 系统要求

要求最好是windwos 2008 server或者以上版本

## 操作步骤

### 解压压安装系统补丁

将V20200916\_1.0.rar（可能版本不同名称不同）拷贝到目标机器并解压

双击MSVBCRT.AIO.2019.05.08.X86 X64.exe将运行环境补丁安装到机器上

### 运行JT/1078 程序

前台运行：双击CarEyeMediaServer.exe 程序运行程序。

后台运行：install service.bat 启动服务。

对后台服务运行的，需要定时对服务器做保护。双击CarEyeMediaServer.bat启动服务保护程序。需要将保护程序做到开机启动的，请按照以下步骤进行操作。

1. WinKey + C 调出超级按钮，点搜索

2. 搜索框里输入shell:startup，然后点击第一个搜索结果。 然后启动文件夹就会出现，接下去就可以把需要开机自动启动的程序（CarEyeMediaServer.bat）的快捷方式拷贝到这个文件夹里。

### 运行nginx-server

到nginx目录运行start.bat即可。

# Linux 部署说明

## 系统要求

系统需要运行在centos 7.x 或以上

## 安装JT1078视频服务器

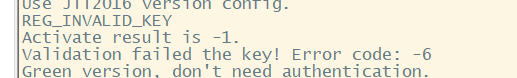
然后按如下步骤完成对JT1078 视频服务的安装和部署

1. 解压软件压缩文件MediaServer.tar.gz到目录某个工作目录。
2. 进入MediaServer执行install.sh 安装程序
3. 安装完成后会自动启动进程和服务。
4. 在/home/CarEye/MediaServer/log 目录下查看最新的日志。

如发现如下提示



说明安装成功。如发现打印出



请与我们联系进行授权。

## 安装和运行nginx服务器

解压nginx.tar.gz 到/usr/local目录。如果服务器有安装别的nginx建议先停，相关功能集成到本服务器。

进入nginx/sbin，运行启动nginx视频服务器

./nginx –c /usr/local/conf/nginx.conf

启动后在log目录下看是否有错误存在确保服务器已经启动。

# 配置说明

## 修改相关端口

一般情况下我们不建议修改端口。当用户需要修改端口时，需要对配置文件configure.xml

JT1078 入流服务器的端口配置

<Server\_Port>10077</Server\_Port> //这是nginx RTMP端口

<Linsten\_Port>9500</Linsten\_Port> //设备上行视频流端口用于查看实时视频

<Playback\_Port>9600</Playback\_Port> //设备上行视频流端口用于回放视频

<WSPort>9213</WSPort> // 自带websocket端口

<WSSPort>9214</WSSPort> //自带SSL wss 端口

配置nginx.conf

server {

listen 4022; //http, ws 拉流端口

location /flv {

flv\_live on;

chunked\_transfer\_encoding on;

add\_header 'Access-Control-Allow-Origin' '\*';

add\_header 'Access-Control-Allow-Credentials' 'true';

}

server {

listen 10085; //服务器状态检测端口

location /stat {

rtmp\_stat all;

rtmp\_stat\_format json;

}

}

rtmp {

out\_queue 4096;

out\_cork 16;

max\_streams 128;

activation\_url http://localhost:4022/;

server {

listen 10077; // rtmp 端口，注意该端口必须要和JT1078 视频服务器里面的相关端口保持一致

## 增加服务器与后台通信功能

服务器通过HTTP协议和其他客户端进行通信

<HttpsAPIPort>7749</HttpsAPIPort> 配置HTTP通信端口

## 语音对讲wss连接和证书

语音对讲功能可以是websocket链接，也可以按照TCP协议进行通信。Websocket 协议进行通信需要提供SSL证书。将证书放到服务器的某个位置并配置相关参数，就可以跟服务器进行语音对讲通信

<EnableWSS>1</EnableWSS> //1 开通语音对讲功能

<WSSPort>9210</WSSPort> // 语音对讲的端口

<WSSPem>ssl/server.pem</WSSPem> //证书的位置 请填写绝对路径

<WSSKey>ssl/server.key</WSSKey>

有关语音对讲的相关实现细节请与我们联系

## 录像服务器配置

JT/T 1078 服务器本身有录像功能。可以将视频流录制成MP4文件。目前使用nginx做文件服务器下载

<http\_server>car-eye.cn</http\_server> //配置文件服务器IP

<http\_port>9000</http\_port> // 服务器端口

<http\_root>/home/record/</http\_root> //确保相关路径存在

在nginx下配置文件服务器。

server {

listen 9000 ssl; #端口

server\_name car-eye.cn;

charset utf-8; # 避免中文乱码

ssl\_certificate "cert/car-eye.cn.pem";

ssl\_certificate\_key "cert/car-eye.cn.key";

ssl\_session\_cache shared:SSL:5m;

ssl\_session\_timeout 5m;

ssl\_ciphers HIGH:!aNULL:!MD5;

ssl\_prefer\_server\_ciphers on;

root /home/record; #显示的根索引目录，注意这里要改成你自己的，目录要存在，跟1078 视频服务路径保持一致

location / {

autoindex on; #开启索引功能

autoindex\_exact\_size off; # 关闭计算文件确切大小（单位bytes）

autoindex\_localtime on; # 显示本机时间而非 GMT 时间

add\_header 'Access-Control-Allow-Origin' '\*';

add\_header 'Access-Control-Allow-Credentials' 'true';

}

}

### **4.5 SSL在nginx内部配置**

server {

listen 4026 ssl;

server\_name www.car-eye.cn;

ssl\_certificate "cert/car-eye.cn.pem";

ssl\_certificate\_key "cert/car-eye.cn.key";

ssl\_session\_cache shared:SSL:5m;

ssl\_session\_timeout 5m;

ssl\_ciphers HIGH:!aNULL:!MD5;

ssl\_prefer\_server\_ciphers on;

location /flv {

flv\_live on;

chunked\_transfer\_encoding on;

add\_header 'Access-Control-Allow-Origin' '\*';

add\_header 'Access-Control-Allow-Credentials' 'true';

}

location /ws {

websocket on srv\_ping\_interval=5s cli\_ping\_interval=500ms srv\_ping\_timeout=20s;

ws\_live on;

}