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

**部署操作说明**

# 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

### 开通服务器的端口

服务器需要用到9500,9600,10085,10077,4022，9210, 9211端口，同时需要确保能访问我们的授权服务器6608端口。对使用云服务器的用户，需要确保上面端口开通，对装有防火墙的用户来说需要确保以上端口能通过防火墙。

### 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，要求开放的端口和windows保持一致，多一个4025端口作为内部测试端口。

## 安装JT1078视频服务器

安装前，请确保安装基础库：

Yum install –y libXv

然后按如下步骤完成对JT1078 视频服务的安装和部署

1. 解压veidio文件夹到/usr/local目录。
2. 执行install-libs.sh 安装动态库到目标机器
3. 添加服务到服务器。将vedio\_check 文件拷贝 /etc/init.d文件夹，确保有写权限

如果没有写权限将相关权限加上 chmod 777 vedio\_check

运行chkconfig --add vedio\_check

1. Chkconfig vedio\_check on 启动服务，如果没启动成功，请执行service vedio\_check start

注意：确保/usr/local/vedio下的可执行文件和脚本有可执行权限。

执行成功后会生成log.file.里面可以查看是否成功。

## 安装和运行nginx服务器

解压nginx\_V20200916\_1.0.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。

# 修改配置

## 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理解

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>9211</WSPort> // websocket端口

<WSSPort>9210</WSSPort> // 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

服务器跟web或者其他应用可以通过MQ进行通信。通信的ip和端口在configue.xml进行配置

<MQIP>127.0.0.1</MQIP> //将该端口指向应用服务器的MQ端口

<MQPort>61627</MQPort>

<MQProducerTopicName>ds\_vedio\_web\_1</MQProducerTopicName>

<MQConsumerTopicName>ds\_web\_vedio\_1</MQConsumerTopicName>

<MQUseTopics>0</MQUseTopics>

<MQSessionTransacted>0</MQSessionTransacted>

配置好IP和端口后，MQ功能将会被启动。应用程序可以按照MQ通信协议跟视频服务器进行通信

## 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>0</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

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

有关语音对讲的相关实现细节请与我们联系

### http2.0和wss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;

}