智能边缘平台

最佳实践

文档版本 02

发布日期 2024-11-26

华为技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：
https://www.huawei.com/cn/psirt/vul-response-process
如企业客户须获取漏洞信息，请参见如下网址：
https://securitybulletin.huawei.com/enterprise/cn/security-advisory

目 录

1 智慧园区人脸检测

1.1 环境准备

通常园区视频功能主要集中在存储和查看，视频分析和态势感知能力较弱。通过使用智能边缘平台与**视频智能分析服务（VIAS）**，提升视频分析和感知能力，实现智慧园区人脸识别检测功能。

> **说明**
>
> 本实践需要使用到视频分析服务的“边缘人脸提取”功能，使用前您需要确认该功能是否在您使用的区域已经上线。本实践以“华北-北京四”区域举例说明。

在开始使用之前，需要先完成相应的环境准备工作。

步骤1 配置边缘摄像头。

请参考摄像头相应型号官方配置文档，登录Web管理页面，配置IP地址，获取rtsp视频流地址。以海康摄像头为例，根据摄像头型号，按照说明书设置IP地址激活摄像机，激活后按照电脑IP配置摄像头IP地址，保持终端设备IP地址与电脑IP地址处于同一网段内；随后登录摄像头浏览器控制页面，添加用户，获得视频流地址，海康摄像头的rtsp地址格式为：XXXX，摄像头配置完成后，使用VLC（**https://www.videolan.org**）对摄像头rtsp流进行检验。使用VLC软件进行播放来检测是否有标准rtsp视频流。

步骤2 准备边缘节点服务器。

该场景下需要部署边缘智能视频算法，因此需要有一定的计算能力，该场景下边缘节点的最低要求如下：

表 1-1 边缘节点要求

项目	规格
OS	操作系统语言必须切换至**英文**。 ● x86_64架构 Ubuntu LTS (Xenial Xerus)、Ubuntu LTS (Bionic Beaver) 、CentOS、EulerOS、RHEL、银河麒麟、中兴新支点、中标麒麟、openEuler、uos（Unity Operating System）、ol（Oracle Linux）、hce（Huawei Cloud Euler）、openEuler 23.09 Edge ● armv7i（arm32）架构 Raspbian GNU/Linux (stretch) ● aarch64（arm64）架构 Ubuntu LTS (Xenial Xerus)、Ubuntu LTS (Bionic Beaver) 、CentOS、EulerOS、RHEL、银河麒麟、中兴新支点、中标麒麟、openEuler、uos（Unity Operating System）、ol（Oracle Linux）、hce（Huawei Cloud Euler）、openEuler 23.09 Edge **说明** 推荐使用面向边缘计算场景的openEuler 23.09 Edge操作系统。
内存	边缘软件开销约128MB，为保证业务的正常运行，建议边缘节点的内存大于256MB。
CPU	>= 1核
硬盘	>= 1GB
GPU（可选）	同一个边缘节点上的GPU型号必须相同。 **说明** 当前支持Nvidia Tesla系列P4、P40、T4等型号GPU。 含有GPU硬件的机器，作为边缘节点的时候可以不使用GPU。 如果边缘节点使用GPU，您需要在纳管前安装GPU驱动。 目前只有使用x86架构的GPU节点才能纳管到IEF中使用。
NPU（可选）	昇腾AI加速处理器。 **说明** 当前支持集成了昇腾处理器的边缘节点，如Atlas 300推理卡、Atlas 800推理服务器。同时支持昇腾310、昇腾310B。 如果边缘节点使用NPU，请确保边缘节点已安装驱动（目前昇腾310仅支持1.3.x.x和1.32.x.x的固件版本，例如1.3.2.B893，可用npu-smi info命令查看固件版本）（NPU驱动需不小于22.0.4版本，进入驱动所在路径如“/usr/local/Ascend/driver”，执行cat version.info命令查看）。如果没有安装驱动，请联系设备厂商获取支持。

项目	规格
容器引擎	Docker版本必须高于17.06。使用高于或等于1.23版本的docker时，需设置docker cgroupfs版本为1，不支持docker HTTP API v2。 （请勿使用18.09.0版本Docker，该版本存在严重bug，详见**https://github.com/docker/for-linux/issues/543**；如果已使用此版本，请尽快升级。） **须知** Docker安装完成后，请将Docker进程配置为开机启动，避免系统重启后Docker进程未启动引起的系统异常。 Docker Cgroup Driver必须设置为cgroupfs。详细配置方法请参考**在边缘节点安装Docker后，如何设置Docker Cgroup Driver?** 。
glibc	版本必须高于2.17。
端口使用	边缘节点需要使用8883端口，8883端口用于边缘节点内置MQTT broker监听端口，请确保该端口能够正常使用。
时间同步	边缘节点时间需要与UTC标准时间保持一致，否则会导致边缘节点的监控数据、日志上传出现偏差。您可以选择合适的NTP服务器进行时间同步，从而保持时间一致。详细配置方法请参见**如何同步NTP服务器?** 。

步骤3 安装Docker。

根据边缘计算节点的操作系统，安装对应版本的Docker。

说明

Docker安装完成后，请将Docker进程配置为开机启动，避免系统重启后Docker进程未启动引起的系统异常。

步骤4 安装GPU驱动并将GPU驱动文件拷贝到边缘节点指定目录下。

在园区人脸检测场景中，需要使用边缘节点上的GPU能力，所以需要提前在边缘节点上安装GPU驱动，缺少GPU驱动会导致人脸识别算法下发失败。

具体操作请参见**拷贝GPU驱动文件**。

步骤5 购买DIS通道。

人脸检测场景中，选择DIS作为数据传输通道，将边缘侧识别出的人脸图片及元数据上传云上进行分析。

在**DIS控制台**中，单击右上角“购买接入通道”，根据提示配置名称、通道类型等参数。DIS通道详细配置请参考**开通DIS通道**。

DIS通道创建完成后请记录通道名称，在下发人脸检测算法时需要选择该通道。

图 1-1 DIS 通道参考配置

----结束

1.2 IEF 服务配置

步骤1 注册边缘节点并纳管。

1. 登录**IEF管理控制台**。
2. 选择左侧导航栏的“边缘资源 > 边缘节点”，单击页面右上角的“注册边缘节点”。
3. 配置边缘节点基本信息。

   如**图1-2**所示，填写边缘节点的名称，AI加速卡选择“Nvidia GPU”，不绑定终端设备。

图 1-2 基本配置

如图1-3所示，为节点配置系统日志和应用日志。您可以自行选择是否开启云端日志（开启后，可在AOM服务中查看日志）。

图 1-3 日志配置

4. 阅读并勾选协议后单击“注册”，进入如下图页面，请下载配置文件和软件，在纳管边缘节点时将会用到。

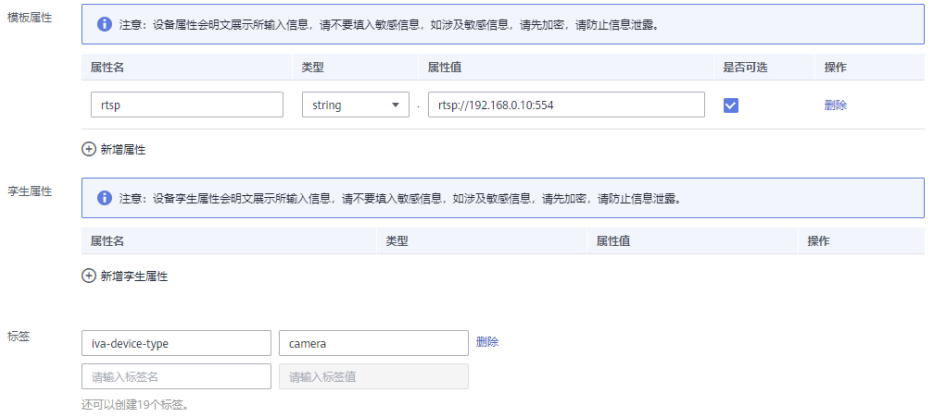
图 1-4 下载配置文件和边缘核心软件

5. 在右下角勾选“我已完成下载配置文件”，并单击“完成”，边缘节点注册完成。
6. 纳管边缘节点，具体操作请参见**纳管边缘节点**。

步骤2 创建设备模板。

1. 登录**IEF管理控制台**。
2. 选择左侧导航栏“边缘资源 > 终端设备”，单击页面右上角的“创建设备模板”。
3. 填写设备模板名称，增加模板属性和标签等。
   - 访问协议选择“MQTT”。
   - 模板属性的属性名请填写“rtsp”，类型为“string”，属性值请输入用户自己的rtsp视频地址，rtsp地址格式为“rtsp://IP:554”，例如“rtsp://192.168.0.10:554”。
   - 标签名请填写“iva-device-type”，标签值请填写“camera”。标签用于标识设备，视频分析服务通过标签识别关联的摄像头设备。

图 1-5 设备模板配置



须知

此处“rtsp”、“iva-device-type”和“camera”必须全部为小写。

4. 单击“创建”，即创建设备模板成功，返回到设备模板页面。

步骤3 创建边缘摄像头。

1. 登录**IEF管理控制台**。
2. 选择左侧导航栏“边缘资源 > 终端设备”，单击页面右上角的“注册终端设备”。
3. 填写设备参数。
   - 填写设备名称。
   - 访问协议选择“MQTT”。
   - 选择**步骤2**中创建的设备模板。
4. 单击“注册”完成一个终端设备的添加。

步骤4 给边缘节点绑定终端设备。

1. 登录**IEF管理控制台**。
2. 选择左侧导航栏“边缘资源 > 边缘节点”。
3. 选择**步骤1**中注册的边缘节点，单击进入节点详情页，选择“设备”页签。
4. 单击“绑定设备”，在弹出的对话框中勾选需要添加的终端设备，填写设备与节点的关系（请填写“camera”）以及备注，然后单击“确定”。
   完成以上操作就可以为您的边缘节点添加一个终端设备，如**图1-6**所示。

   图 1-6 绑定设备

5. 绑定设备之后，单击终端设备名称，进入终端设备详情页面，可以修改终端设备的属性信息，查看终端设备关联的节点、设备孪生信息、标签等。

----结束

1.3 下发人脸检测算法

步骤1 购买边缘人脸检测算法包。

1. 登录**视频分析服务控制台**，并选择与边缘节点相同的区域。
2. 在页面左侧导航栏中选择“服务 > 园区智能体”，进入视觉能力包列表，在边缘人脸检测算法包所在行单击“购买”。

图 1-7 购买边缘人脸检测算法包

3. 选择“购买时长”和“视频路数”，单击“立即购买”。

步骤2 在“服务 > 园区智能体”页面，单击已购买算法包操作栏的“使用”。

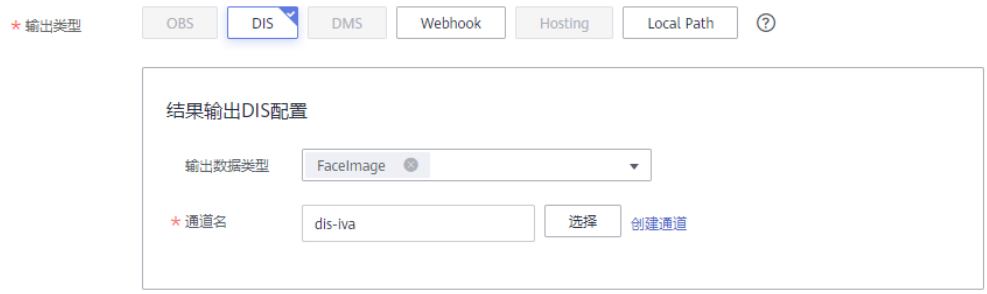
步骤3 在目标服务的操作栏，单击“创建作业”，进入创建视觉分析作业页面，设置作业参数。

步骤4 输入数据选择“Edge Camera”，获取摄像头设备列表，选择相应的摄像头。位于右侧的摄像头名称表示已选择摄像头，如图1-8所示。

图 1-8 选择终端设备

步骤5 输出类型选择“DIS”，通道名选择步骤5中购买的DIS通道，如图1-9所示。

图 1-9 填入通道名称



步骤6 单击页面右下角“立即创建”。

创建完成后，就会自动下发作业，由于需要拷贝2G左右的镜像到边缘服务器上，所以需要一定的时间。可以通过查看作业的状态判断是否成功。

步骤7 查看人脸检测结果。

1. 登录DIS控制台，单击DIS通道名称进入详情页，在“监控”页签查看输入流是否有数据。
2. 使用DIS接口查看检测到的人脸数据。

----结束

2 通过专线或 VPN 连接 IEF

操作场景

线下边缘节点无法通过公网访问IEF时，可以选择通过**云专线（DC）**或**VPN**连接华为云VPC，然后通过**VPC终端节点**在VPC提供私密安全的通道连接IEF，从而使得线下边缘节点在无法访问公网时连接IEF。

连接方案说明

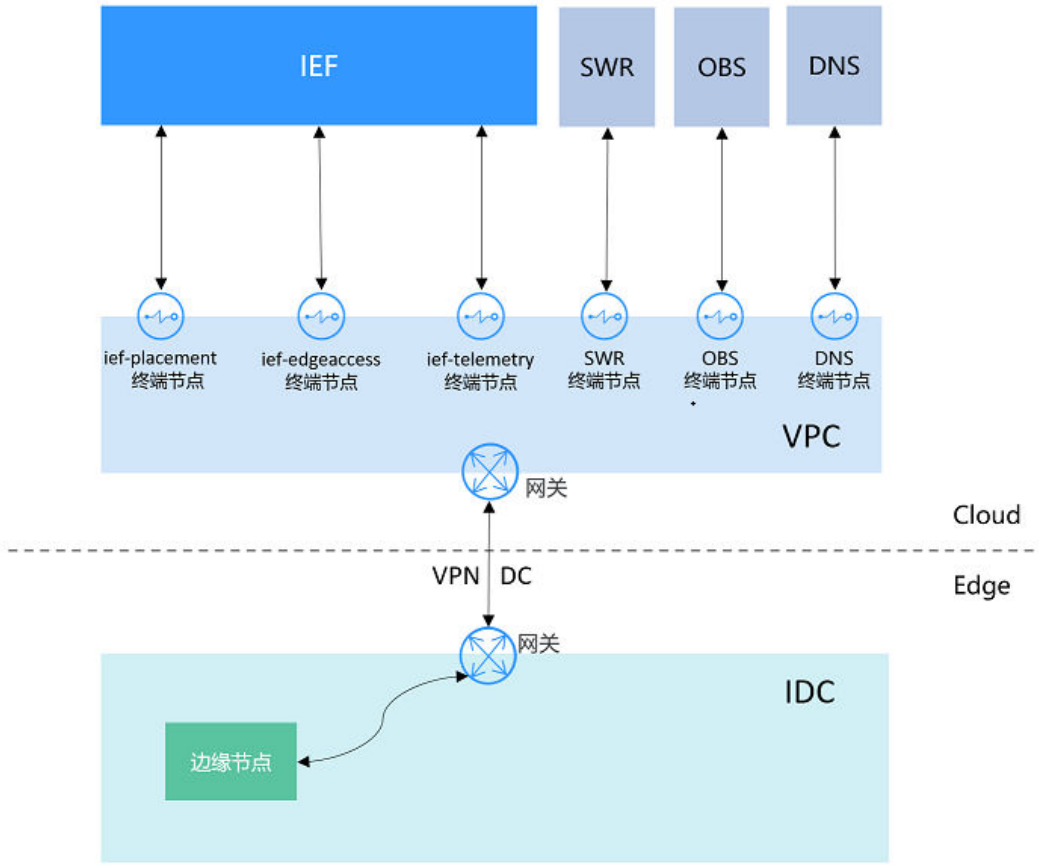
纳管边缘节点部署应用时，需要能够与IEF、SWR、OBS通信，在无法通过公网连接的情况下，可以先通过VPN或专线（DC）与华为云VPC连接，然后通过VPC终端节点服务，让VPC能够在内网访问IEF、SWR和OBS，具体连接方案如**图2-1**所示。

与IEF连接需要创建三个终端节点，分别为如下三个。

- ief-placement：用于边缘节点的纳管和升级。
- ief-edgeaccess：用于边缘节点与IEF发送边云消息。
- ief-telemetry：边缘节点上传监控和日志数据。

与SWR连接需要创建一个终端节点，与OBS通信需要创建OBS和DNS两个终端节点（OBS只能通过域名访问，需要通过DNS动态解析OBS的地址才能访问到）。

图 2-1 通过专线或 VPN 连接 IEF



操作步骤

步骤1 创建VPC。

创建VPC的方法请参见**创建虚拟私有云和子网**。

您也可以使用已有VPC。

须知

VPC网段不能与IDC的网段重复。

步骤2 使用DC或VPN连接VPC。

具体连接方法请参见如下链接。

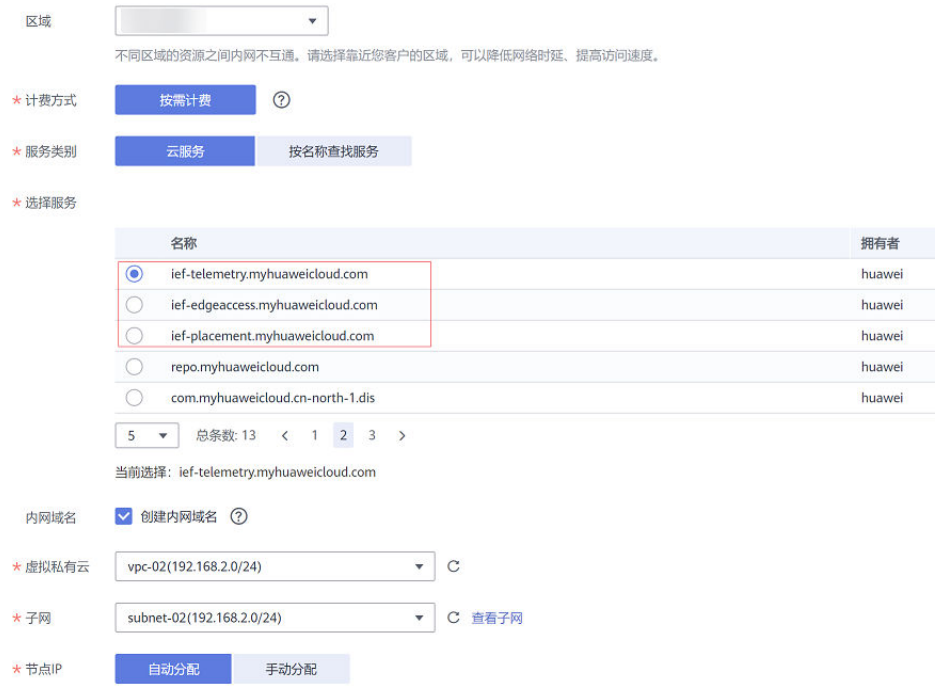
- VPN：**https://support.huaweicloud.com/qs-vpn/vpn_qs_00003.html**
- DC：**https://support.huaweicloud.com/qs-dc/zh-cn_topic_0145790541.html**

步骤3 创建IEF终端节点，使得边缘节点能够与IEF连接。

共需要创建三个终端节点，分别为ief-placement、ief-edgeaccess和ief-telemetry。具体创建步骤如下。

1. 登录**VPCEP控制台**，单击右上角的“购买终端节点”。
2. 选择IEF的终端节点和虚拟私有云。

图 2-2 创建 IEF 终端节点

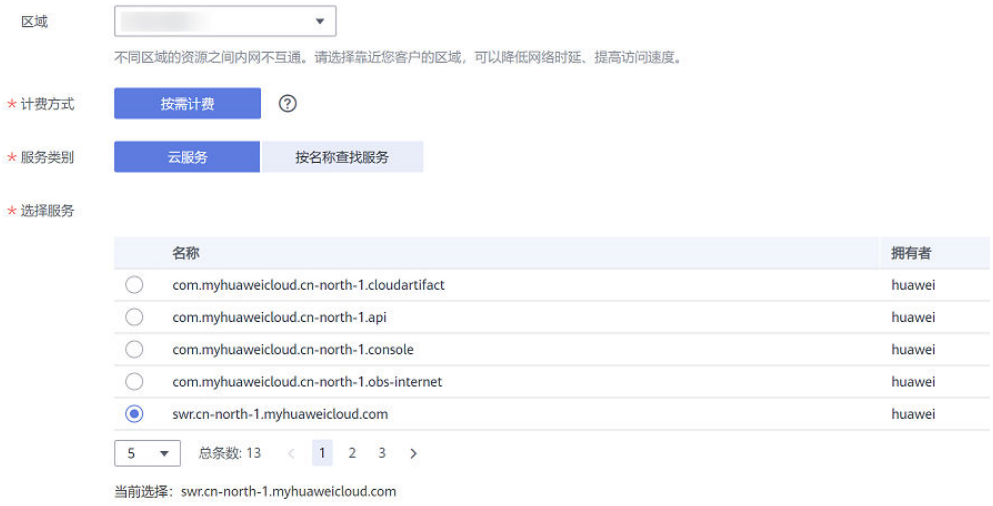


3. 单击“立即购买”，确认信息无误后单击“提交”，完成创建。

步骤4 创建SWR终端节点，使得边缘节点能够从SWR拉取容器镜像。

创建方法与**创建IEF终端节点**相同。

图 2-3 创建 SWR 终端节点



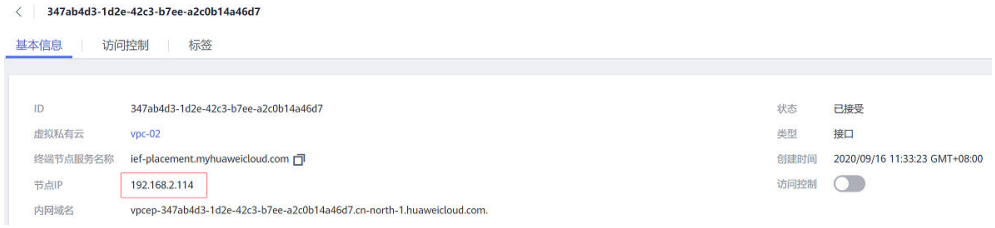
步骤5 创建DNS和OBS终端节点，使得边缘节点能够访问OBS。

具体方法请参见**访问OBS**。

步骤6 给边缘节点添加hosts配置。

查询IEF和SWR的终端节点IP地址，共4个IP地址，配置到边缘节点的“/etc/hosts”文件中。

图 2-4 查询终端节点 IP 地址



打开“/etc/hosts”文件，在文件末尾加入如下配置，使得访问IEF和SWR的域名指向终端节点的IP地址。

须知

此处IP地址和域名需要根据实际情况修改，IP地址为上面步骤查询到的地址，不同区域的域名不相同，具体请参见**域名地址**。

```
192.168.2.20    ief2-placement.cn-north-1.myhuaweicloud.com
192.168.2.142    ief2-edgeaccess.cn-north-1.myhuaweicloud.com
192.168.2.106   ief2-telemetry.cn-north-1.myhuaweicloud.com
192.168.2.118   swr.cn-north-1.myhuaweicloud.com
```

步骤7 注册并纳管边缘节点，具体步骤请参见**边缘节点概述**。

----结束

域名地址

说明

铂金版ief-edgeaccess有单独的地址，请在IEF控制台“总览”页面查询，云端接入域名的取值即为edgeaccess域名。

区域	名称	域名
华北-北京一	ief-placement	ief2-placement.cn-north-1.myhuaweicloud.com
	ief-edgeaccess	ief2-edgeaccess.cn-north-1.myhuaweicloud.com
	ief-telemetry	ief2-telemetry.cn-north-1.myhuaweicloud.com
	swr	swr.cn-north-1.myhuaweicloud.com
华北-北京四	ief-placement	ief2-placement.cn-north-4.myhuaweicloud.com
	ief-edgeaccess	ief2-edgeaccess.cn-north-4.myhuaweicloud.com

区域	名称	域名
	ief-telemetry	ief2-telemetry.cn-north-4.myhuaweicloud.com
	swr	swr.cn-north-4.myhuaweicloud.com
华南-广州	ief-placement	ief-placement.cn-south-1.myhuaweicloud.com
	ief-edgeaccess	ief-edgeaccess.cn-south-1.myhuaweicloud.com
	ief-telemetry	ief-telemetry.cn-south-1.myhuaweicloud.com
	swr	swr.cn-south-1.myhuaweicloud.com
华东-上海一	ief-placement	ief-placement.cn-east-3.myhuaweicloud.com
	ief-edgeaccess	ief-edgeaccess.cn-east-3.myhuaweicloud.com
	ief-telemetry	ief-telemetry.cn-east-3.myhuaweicloud.com
	swr	swr.cn-east-3.myhuaweicloud.com
华东-上海二	ief-placement	ief2-placement.cn-east-2.myhuaweicloud.com
	ief-edgeaccess	ief2-edgeaccess.cn-east-2.myhuaweicloud.com
	ief-telemetry	ief2-telemetry.cn-east-2.myhuaweicloud.com
	swr	swr.cn-east-2.myhuaweicloud.com

3 使用开源 C 语言库连接 MQTT Broker

操作场景

MQTT是一种发布/订阅模式的消息协议，能够在硬件性能低下的远程设备以及网络状况糟糕的情况下工作。

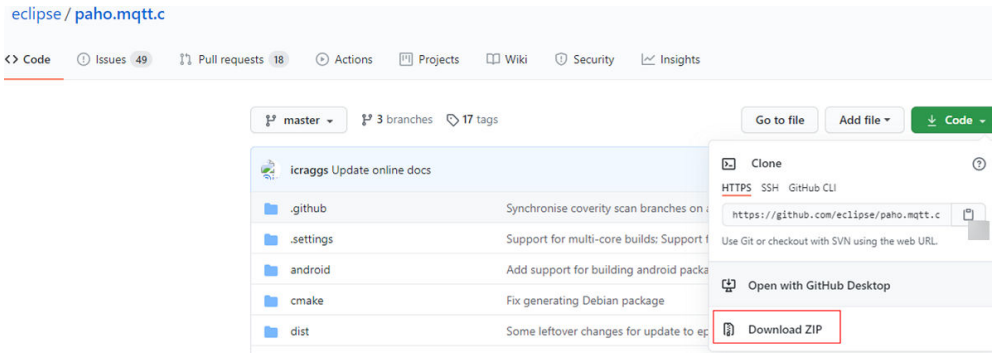
本文介绍一种开源的C语言库Eclipse Paho C Client Library连接使用IEF的内置MQTT Broker。

操作步骤

步骤1 准备一台Linux机器，下载源码。

git clone https://github.com/eclipse/paho.mqtt.c.git

或直接在https://github.com/eclipse/paho.mqtt.c页面下载zip包，然后解压。



步骤2 安装编译依赖工具。

Ubuntu系统执行如下命令。

apt-get install build-essential gcc make cmake cmake-gui cmake-curses-gui

apt-get install fakeroot fakeroot devscripts dh-make lsb-release

apt-get install libssl-dev

apt-get install ninja-build

CentOS系统执行如下命令。

yum install build-essential gcc make cmake cmake-gui cmake-curses-gui

yum install fakeroot fakeroot devscripts dh-make lsb-release

yum install openssl-devel

yum install ninja-build

步骤3 修改示例程序代码。

在源码的“src/samples/ paho_cs_pub.c”文件中，增加如下行。

ssl_opts.enableServerCertAuth =0;

```
 94                ssl_opts.CApath = opts.capath;
 95                ssl_opts.keyStore = opts.cert;
 96                ssl_opts.trustStore = opts.cafile;
 97                ssl_opts.privateKey = opts.key;
 98                ssl_opts.privateKeyPassword = opts.keypass;
 99                ssl_opts.enabledCipherSuites = opts.ciphers;
100                ssl_opts.enableServerCertAuth = 0;
101                conn_opts.ssl = &ssl_opts;
```

在“src/samples/paho_cs_sub.c”文件中增加如下行。

```
91                ssl_opts.CApath = opts.capath;
92                ssl_opts.keyStore = opts.cert;
93                ssl_opts.trustStore = opts.cafile;
94                ssl_opts.privateKey = opts.key;
95                ssl_opts.privateKeyPassword = opts.keypass;
96                ssl_opts.enabledCipherSuites = opts.ciphers;
97                ssl_opts.enableServerCertAuth = 0;
98                conn_opts.ssl = &ssl_opts;
```

步骤4 编译示例程序。

mkdir /tmp/build.paho

cd /tmp/build.paho

cmake -GNinja -DPAHO_BUILD_STATIC=TRUE -DPAHO_BUILD_SHARED=FALSE -DPAHO_WITH_SSL=TRUE -DPAHO_BUILD_SAMPLES=TRUE *{paho.mqtt.c目录}*

ninja package

其中 *{paho.mqtt.c目录}* 为paho.mqtt.c源码所在的目录，如“/root/work/paho.mqtt.c”。

步骤5 进入编译之后的目录，将编译生成的二进制文件“paho_cs_pub”和“paho_cs_sub”拷贝至边缘节点上。

cd /tmp/build.paho/src/samples/

```
(base) root@cci-clustermanager-xsw:/tmp/build.paho/src/samples# ls
CMakeFiles          MQTTAsync_publish       MQTTAsync_subscribe  MQTTClient_publish_async  paho_c_pub  paho_cs_sub
cmake_install.cmake  MQTTAsync_publish_time  MQTTClient_publish    MQTTClient_subscribe      paho_cs_pub  paho_c_sub
```

步骤6 下载边缘节点证书。

1. 登录IEF控制台，在左侧选择“边缘资源 > 边缘节点”，在右侧单击边缘节点名称，进入边缘节点详情页。选择“证书”页签，单击“添加证书”。

图 3-1 添加证书

2. 在弹出的窗口中输入证书名称，单击“确定”。
3. 将下载好的证书，拷贝至边缘节点，并解压。

步骤7 运行示例程序。

以发布消息到指定topic为示例，查看“paho_cs_pub”命令指导。

```
(base) root@cci-clustermanager-xsw:/tmp/build.paho/src/samples# ./paho_c_pub
Eclipse Paho MQTT C publisher

Library information:
Product name: Eclipse Paho Asynchronous MQTT C Client Library
Version: 1.3.8
Build level: 2021-01-20T14:05:21Z
OpenSSL version: OpenSSL 1.1.0l  10 Sep 2019
OpenSSL flags: compiler: gcc -DDSO_DLFCN -DHAVE_DLFCN_H -DNDEBUG -DOPENSSL_THREADS -DOPENSSL_NO_STATIC_ENGI
BN_ASM_MONT -DOPENSSL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DRC4_ASM -DM
H_ASM -DECP_NISTZ256_ASM -DPADLOCK_ASM -DPOLY1305_ASM -DOPENSSLDIR="\"/usr/local/ssl\"" -DENGINESDIR="\"/us
OpenSSL build timestamp: built on: reproducible build, date unspecified
OpenSSL platform: platform: linux-x86_64
OpenSSL directory: OPENSSLDIR: "/usr/local/ssl"

Usage: paho_c_pub [topicname] [-t topic] [-c connection] [-h host] [-p port]
       [-q qos] [-i clientid] [-u username] [-P password] [-k keepalive_timeout]
       [-V MQTT-version] [--quiet] [--trace trace-level]
       [-r] [-n] [-m message] [-f filename]
       [--maxdatalen len] [--message-expiry seconds] [--user-property name value]
       [--will-topic topic] [--will-payload message] [--will-qos qos] [--will-retain]
       [--cafile filename] [--capath dirname] [--cert filename] [--key filename]
       [--keypass string] [--ciphers string] [--insecure]

  -t (--topic)        : MQTT topic to publish to
  -c (--connection)   : connection string, overrides host/port e.g wss://hostname:port/ws.  Use this option
                        rather than host/port to connect with TLS and/or web sockets. No default.
  -h (--host)         : host to connect to.  Default is localhost.
  -p (--port)         : network port to connect to. Default is 1883.
  -q (--qos)          : MQTT QoS to publish with (0, 1 or 2). Default is 0.
  -V (--MQTTversion)  : MQTT version (31, 311, or 5).  Default is 311.
  --quiet             : do not print error messages.
  --trace             : print internal trace ("error", "min", "max" or "protocol").
```

发布消息示例如下：

./paho_cs_pub -c ssl://127.0.0.1:8883 -q 0 **-m** "xxx" **-t** "aaa" **--cert** /root/mqtt_cert/xOEMIsYVpw_private_cert.crt **--key** /root/mqtt_cert/xOEMIsYVpw_private_cert.key

这条命令向内置MQTT Broker名为“aaa”的Topic发送了内容为“xxx”的消息，其中“127.0.0.1:8883”为边缘节点内置MQTT Broker的地址，“/root/mqtt_cert/xOEMIsYVpw_private_cert.crt”和“/root/mqtt_cert/xOEMIsYVpw_private_cert.key”为边缘节点证书。

----结束

