

HUAWEI 路网数字化服务

快速入门

文档版本 01
发布日期 2025-05-08



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI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云计算技术有限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，华为云计算技术有限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。

目 录

1 快速开通路网数字化服务.....	1
2 设备快速接入.....	4
2.1 概述.....	4
2.2 快速对接一个 Edge 设备示例.....	5
2.3 快速对接一个 RSU 设备示例.....	8
2.4 快速配置 RSU 设备类型.....	14
2.5 ITS800 连接到 V2Xserver.....	14
2.5.1 环境确认.....	15
2.5.2 清除 IEF.....	20
2.5.3 启动边缘节点安装命令.....	20
2.6 快速管理第三方边缘应用.....	21
2.7 在线调试.....	23
2.8 使用 MQTT.fx 调测.....	24
3 快速玩转路网数字化平台.....	27
3.1 手动事件下发示例.....	27

1 快速开通路网数字化服务

本节内容为您介绍购买路网数字化服务的方式，以及购买前注册华为云账户并实名认证的操作步骤。

注册华为云并实名认证

如果您已有一个华为云账户，请跳到下一个任务。如果您还没有华为云帐户，请参考以下步骤创建。

步骤1 打开[华为云网站](#)。

步骤2 单击“注册”，根据提示信息完成注册。注册成功后，系统会自动跳转至您的个人信息界面。

步骤3 个人或企业账号实名认证请参考[实名认证](#)。

----结束

购买路网数字化服务

步骤1 登录[管理控制台](#)。

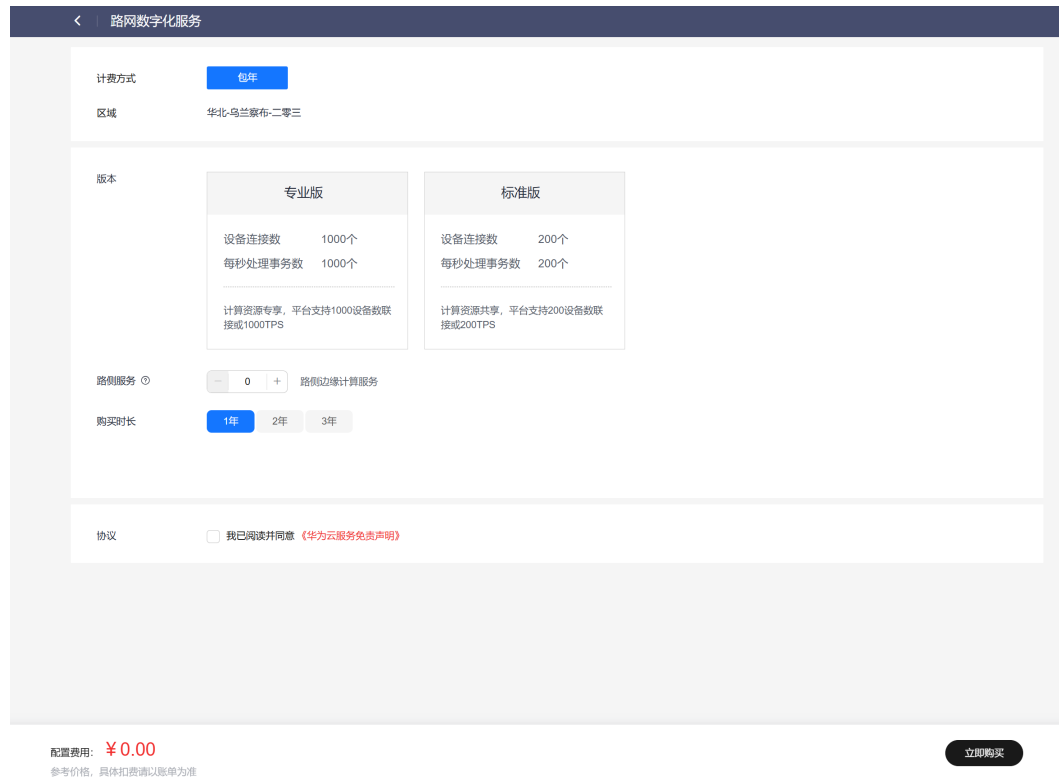
步骤2 单击管理控制台左上角的 ，选择“北京四”。

步骤3 在“服务列表”选择“物联网 > 路网数字化服务 DRIS”。



步骤4 单击“立即购买”，系统进入购买页。

计费方式默认“包年”，区域默认“北京四”。



步骤5 选择“版本”。

仅可选“专业版”。路网数字化服务专业版的资源独享，具有更高的数据可靠性和安全性。

步骤6 选择“路侧服务”数量。

根据需要设置路侧边缘计算服务。

步骤7 选择“购买时长”。

购买方式“包年”，最短1年，最长3年。系统会根据您选择的“服务配置”和“购买时长”自动计算费用。

步骤8 勾选同意协议，单击“立即购买 > 去支付 > 确认付款”。

----结束

2 设备快速接入

2.1 概述

路网数字化边缘服务通过摄像头、雷达设备识别路面交通事件，通过路网数字化服务下发到路侧单元，转发到车载单元。

路网数字化服务为您提供以下功能：

- 建立交通事件，下发至路侧单元，路侧单元转发给车载单元。主要场景为车内标
牌、恶劣天气类事件。
- 城市/区域交通实时监控，覆盖交通事故数、联接车辆数、拥堵路段、车型分布、
车流量、事故高发路段排行、在线设备数等重要指标。
- RSU（Road Side Unit，路侧单元）等设备状态实时监控，帮助企业随时掌握设备
情况，助力设备故障及时处理。
- 边云协同能力，可实现云上控制的路网数字化边缘服务的软件部署和更新。
- 数据存储和数据开放接口。

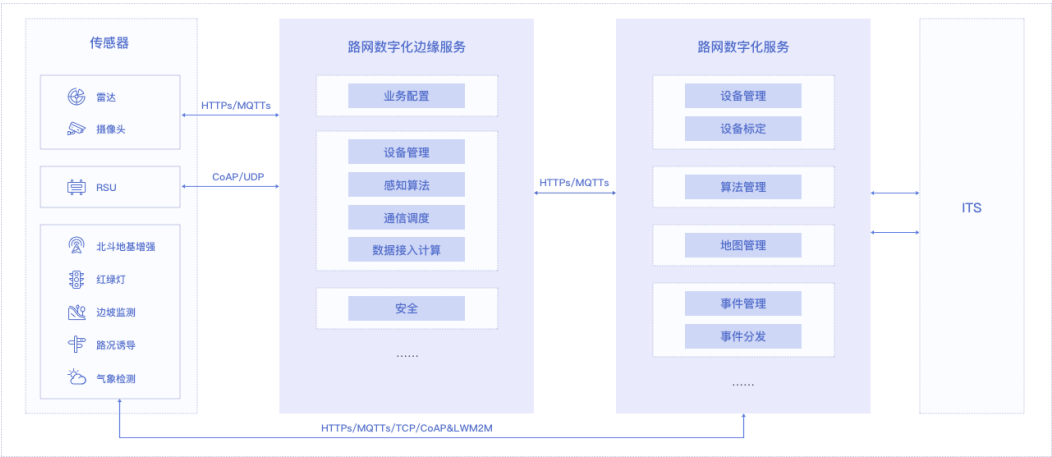


表 2-1 设备管理特性说明

特性分类	功能描述
设备模型	支持定义各种路侧设备的能力，为路侧设备提供边缘统一的设备模型。
设备接入	支持设备发放、认证、注册鉴权、配置、数据订阅、命令、数据存储归档服务等，保证合法设备相互通信以及传输信息的安全。
设备鉴权	支持一机一密、双向证书认证等多种认证方式，多种传输加密协议保障通道安全，数据隐私保护符合欧盟GDPR标准。

本节内容为您介绍在购买路网数字化服务后，如何快速对接设备，构筑路网数字化服务，使能智慧交通助力自动驾驶。

2.2 快速对接一个 Edge 设备示例

本节内容为您介绍在购买路网数字化服务后，如何在路网数字化平台快速对接一个边缘Edge设备。

操作步骤

步骤1 进入路网数字化服务控制台，选择“设备管理 > 设备总览 > 边缘Edge > 注册设备”。



步骤2 在弹出的对话框中，参考表 Edge设备信息填写Edge设备基础信息。

表 2-2 Edge 设备信息

参数类型	参数名称	说明
基本信息	设备名称	该边缘设备的名称，建议统一规划。

参数类型	参数名称	说明
	设备编码	该边缘设备的设备编码（ESN）。
	硬件类型	ITS800或ATLAS。
	协议类型	该边缘作为服务端的使用的协议类型，http和https两种
	SSL证书	该边缘作为服务端，https使用的SSL证书
	私钥	该边缘作为服务端，https使用的私钥
	位置编码	该边缘设备实际部署位置的位置编号，需统一规划。
	端口号	ITS800或者ATLAS的端口。
-	IP地址	该边缘设备设置的本地IP地址，需统一规划。
关联设备 Edge通用配置 Edge高级配置	经纬度	该边缘设备的经纬位置。
	用户名	该边缘设备注册的用户名。
	密码	该边缘设备注册的密码。
	描述	该边缘设备的描述信息，由用户自定义。
	关联的RSU AVP场景 RSM上报 时延补偿 隧道定位场景 应用日志 无	该Edge预期关联的RSU设备列表。 开启选项，该Edge将用于AVP（自动泊车）场景。 开启选项，该Edge将识别的RSM消息上传到云端。 开启选项，该Edge可以根据交通参与者的历史时刻的位置和速度等信息预测出当前时刻的信息，以弥补机器学习算法带来的目标识别时延。 开启选项，该Edge可以用于隧道场景，推送车辆航向信息、车辆速度信息、车牌信息、车辆位置。 开启选项，该Edge可以将日志上报至云平台AOM。 JSON格式，例如： <pre>{ "service_config":{ "event_match_para":{ "frame_num":60, "match_num":100 } } }</pre> 高级配置参数主要配置边缘应用运行和分析事件时所需要的复杂参数，主要包括：事件分析的阈值参数（比如，超速、慢速的速度判断阈值，停车判断的持续时长阈值，拥堵判断的速度阈值等），运维相关参数（如：日志开关等）和其他影响应用运行的参数。 所有参数的默认值都使用调优后的最佳值，一般不建议修改；如果需要修改，请联系华为工程师讨论修改方案、确定修改参数和具体的修改数值，获取最终修改指令后修改。

步骤3 获取安装命令并复制。

边缘Edge | 路侧RSU | 摄像头 | 雷达 | 车辆 | 信号机

边缘Edge注册成功后, 可进入详情页开通业务通道和部署应用。

注册设备 列表导出

Q 选择属性筛选, 或输入关键字搜索

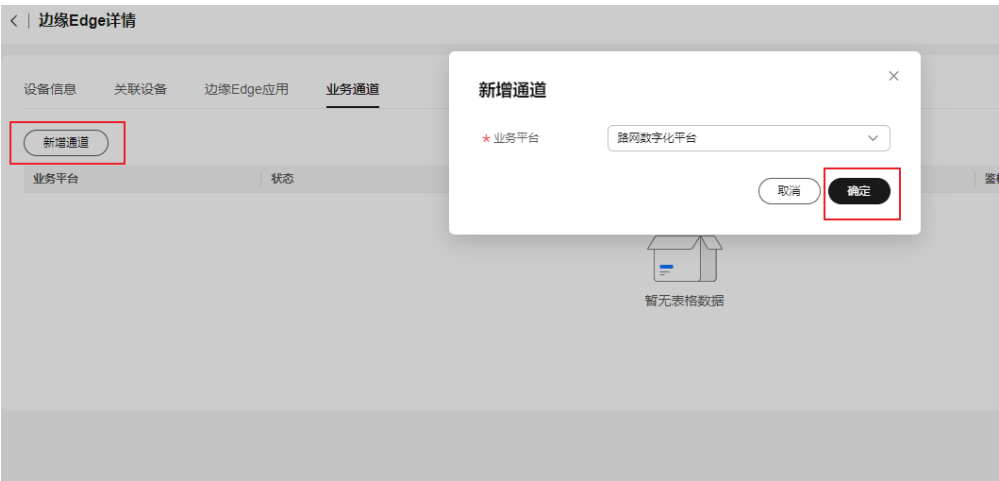
设备名称	设备编码	设备ID	设备状态	业务通道状态	创建时间	安装命令	操作
dwdad	sdasdqwr	7be1565f-5a2c-4291-9da...	在线	在线	2024/06/28 17:05:56 GM...	--	部署应用 详情 编辑 删除
qveqve	qveqveqr	912899a4-1104-4eca-b57...	部署中	--	2024/05/10 09:10:20 GM...	--	部署应用 详情 编辑 删除
dasdaada	qveqvedqsd	110a30c9-ecbd-4bd1-acb...	离线	未知	2024/04/28 09:53:08 GM...	--	部署应用 详情 编辑 删除
casdfdsa	sdasdasd	9582af01-e8e6-4ff3-9e4b...	离线	--	2024/04/28 09:44:25 GM...	--	部署应用 详情 编辑 删除
asdfsdf	fasdf	78bc7ad7-d3ac-4ff9-9e8c...	待部署	--	2024/04/08 15:41:52 GM...	📄	部署应用 详情 编辑 删除
sadfsdf	fasdf	ea5a8f09-683f-4a50-6284...	待部署	--	2024/04/01 17:02:37 GM...	📄	部署应用 详情 编辑 删除
dxc_test001	dxc_test001	90eca057-452f-4440-ab1f...	部署中	--	2024/04/01 10:06:09 GM...	--	部署应用 详情 编辑 删除
teset111111	teset111111	f5268d16-3fa5-4a97-9a31...	离线	未知	2024/03/21 10:53:18 GM...	--	部署应用 详情 编辑 删除
lrv003	9999999998	8ca2fe65-fb1a-4c32-ab23...	待部署	--	2024/01/26 19:43:56 GM...	📄	部署应用 详情 编辑 删除

步骤4 获取安装命令后，参考《[ITS800连接V2Xserver](#)》内的步骤启动边缘节点，完成线下部署设备后才能上线该Edge设备。

须知

该边缘节点安装命令的有效时间为生成后30分钟，请在30分钟内使用。

步骤5 添加业务通道，根据实际情况选择业务平台，单击“确定”完成注册。



步骤6 当Edge设备列表中的Edge状态为“在线”，表示该Edge设备和路网数字化服务对接成功。后续可去Edge详情页面[部署应用](#)。

----结束

部署应用

步骤1 边缘节点安装完成后，等待边缘节点的设备状态为“在线”。

步骤2 单击“详情 > 应用部署”，参照[应用部署信息](#)输入应用信息。

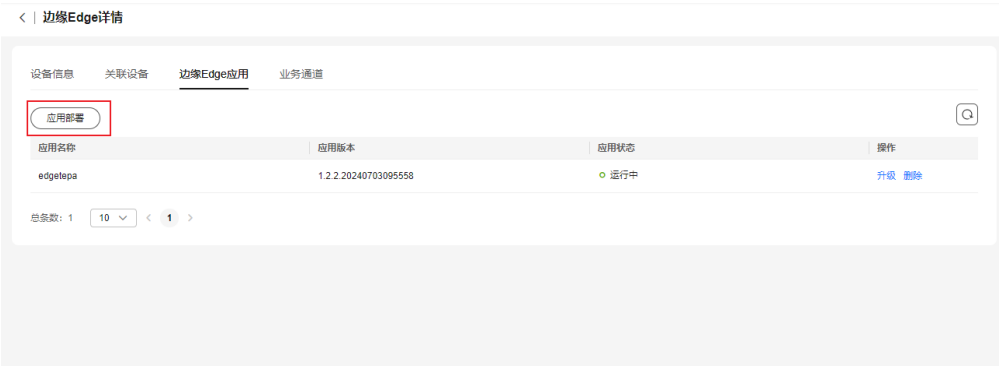


表 2-3 应用部署信息

参数名称	说明
应用名称	根据已添加的边缘应用选择，例如edgetepa。
应用版本	下拉列表，选择当前最高版本的边缘应用。

----结束

2.3 快速对接一个 RSU 设备示例

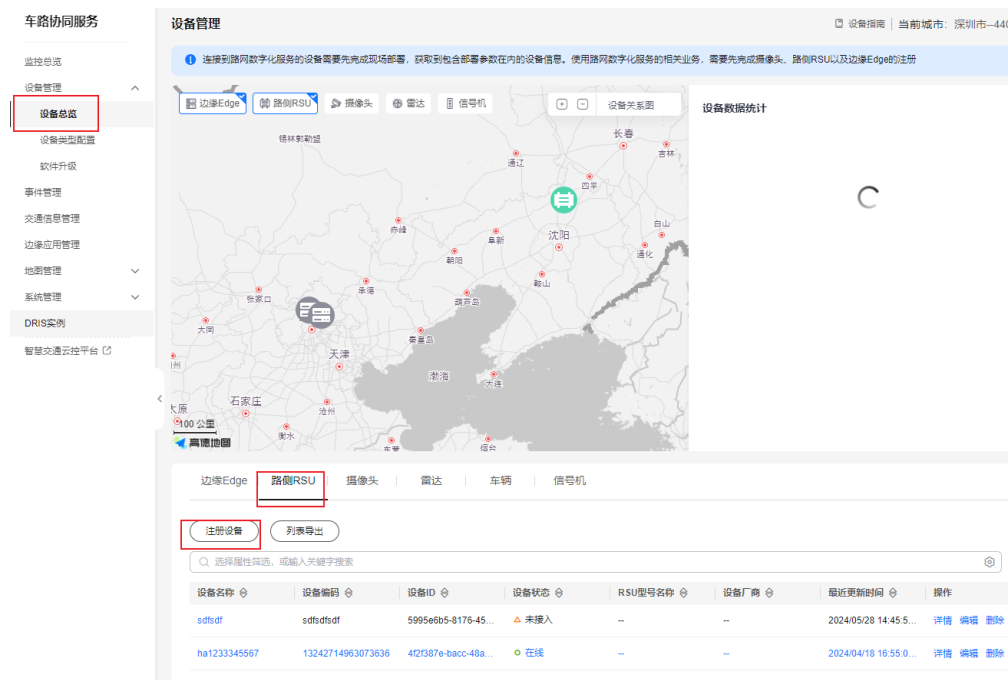
本节内容为您介绍在购买路网数字化服务后，如何在路网数字化平台快速对接一个路侧RSU设备。

前提条件

- 路网数字化服务的RSU设备完成现场部署后，获取RSU设备信息，其中设备编码必须和RSU设备一致，否则RSU无法完成接入。
- 获取路网数字化服务设备侧接入地址及端口，并在RSU上配置完成。

操作步骤

- 步骤1** 登录ManageOne运营面。
- 步骤2** 进入路网数字化服务页面，选择“设备管理 > 设备总览 > 路侧RSU > 注册设备”。

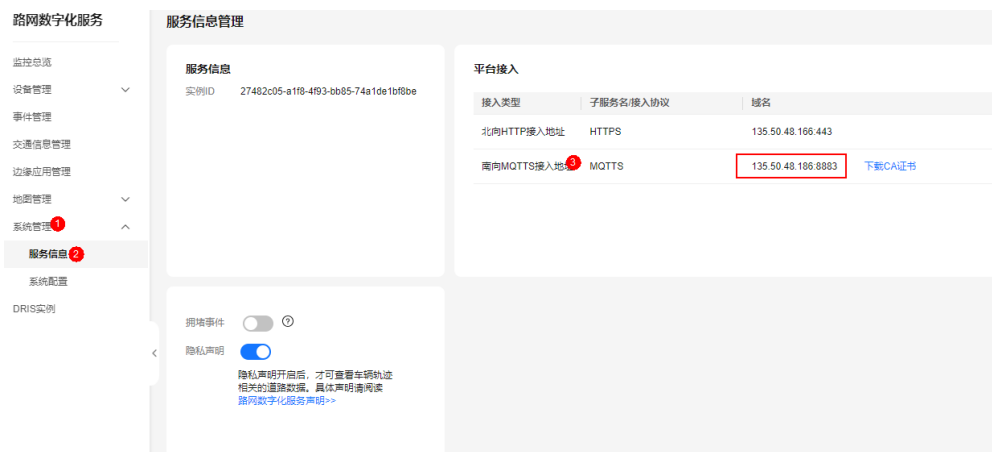


步骤3 在弹出的对话框中，参考RSU设备信息填写RSU设备基本信息。

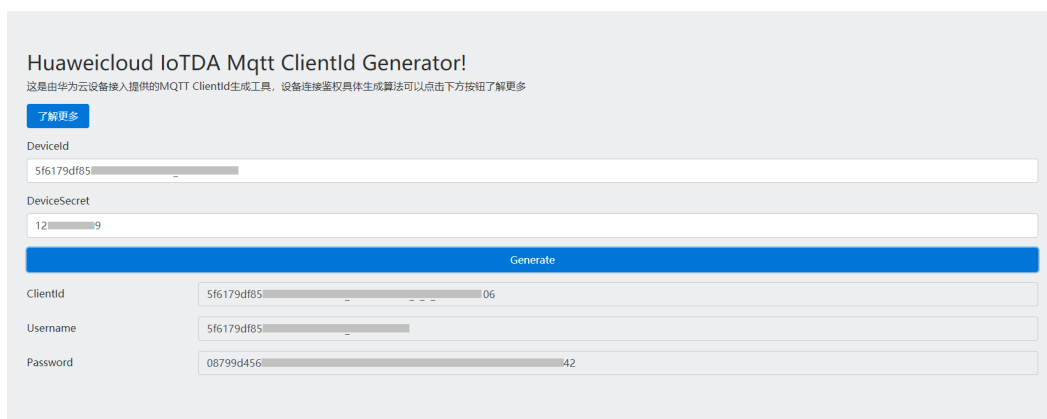
表 2-4 RSU 设备信息

参数类型	参数名称	说明
基本信息	设备名称	该RSU设备的名称，建议统一规划。输入中文、数字、字母、下划线、中划线，长度不能超出128。
	设备编码	该RSU的设备编码（ESN）。长度不超过64，只允许字母、数字以及_等字符的组合。
	位置编码	该RSU实际部署位置的位置编号，需统一规划。长度不低于1不超过128，只允许字母、数字、下划线(_)的组合。
	IP地址	该RSU设置的本地IP地址，需统一规划。
	路口id	在路网地图中，rsu设备所在区域对应的路口ID。
	Edge连接上限	表示该RSU最大连接的Edge个数，最大64个。
	描述	该RSU的描述信息，由用户自定义。长度不超过255，允许数字、字母、中文以及：？！，；：、,?!等字符的组合。
	RSU型号名称	需要提早在设备类型配置中进行RSU类型配置。下拉列表，选择RSU型号名称。
	RSU密钥	输入字母（a-f或A-F）或者数字，最少输入8位字符。 注意 该密钥很重要，请自行储存，为安全起见查询RSU时不再反显。

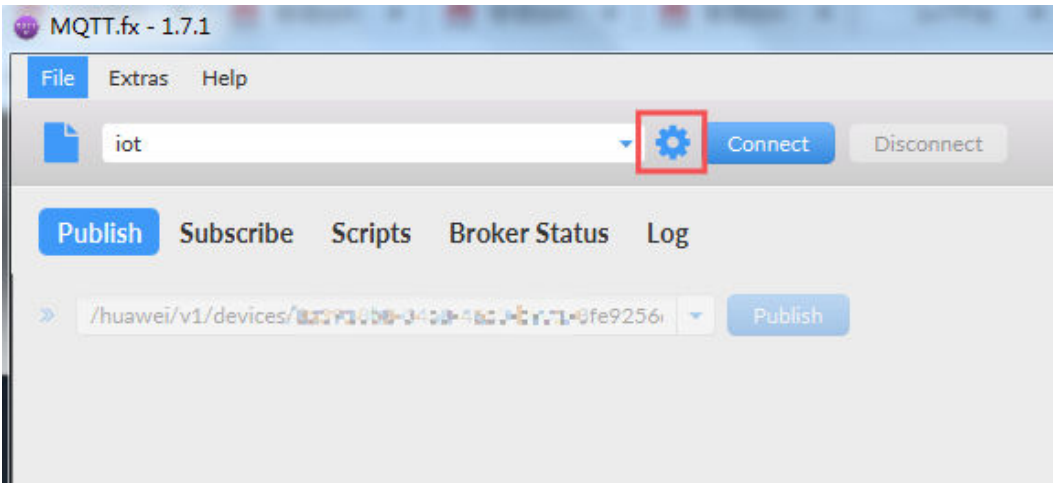
- 步骤4** 单击“确定”完成注册。在路侧RSU的设备列表中，当设备列表中RSU设备状态为“在线”，表示该设备和路网数字化服务对接成功。
- 步骤5** 如果没有实际的RSU设备，可以使用MQTT.fx工具模拟RSU注册上线。这里以MQTT.fx为例介绍如何模拟RSU设备注册上线，访问[MQTT.fx下载页面](#)，下载并安装最新版本的MQTT.fx工具。。
- 步骤6** 进入路网数字化服务页面，选择“系统管理->服务信息”。记录“平台接入”下的“南向MQTT接入地址”行的“域名”信息。



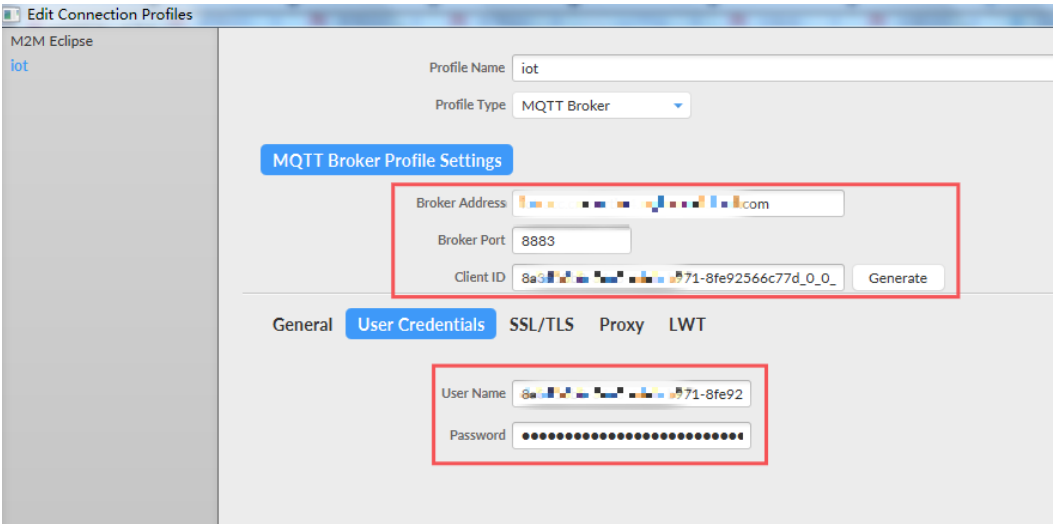
- 步骤7** 请访问[这里](#)，DeviceId为**步骤3**所填写的设备编码，DeviceSecret为**步骤3**所填写的RSU密钥，生成连接信息（ClientId、Username、Password）。



- 步骤8** 打开MQTT.fx软件，单击设置图标。



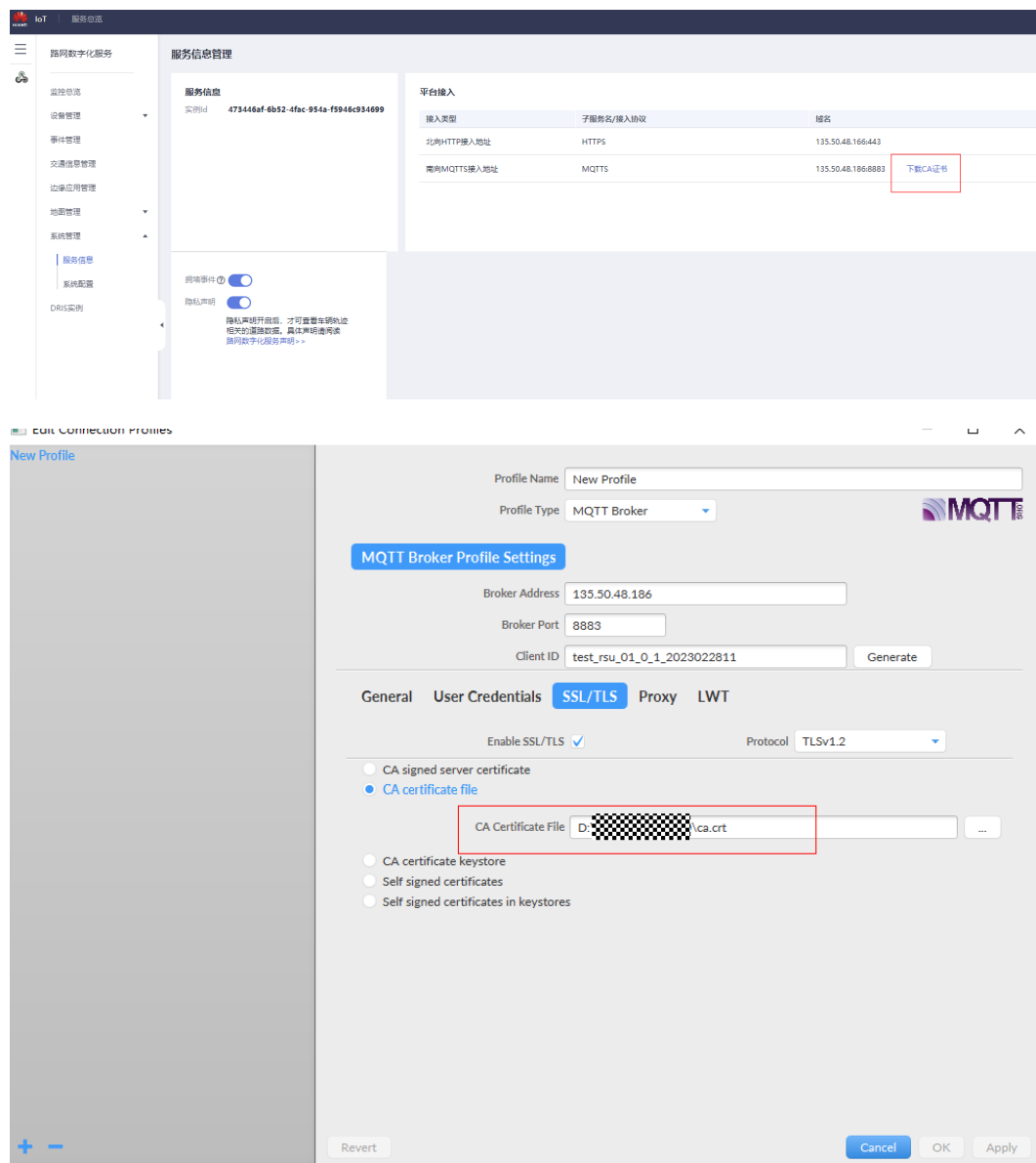
步骤9 填写路网数字化服务南向MQTT接入地址，并配置鉴权参数。



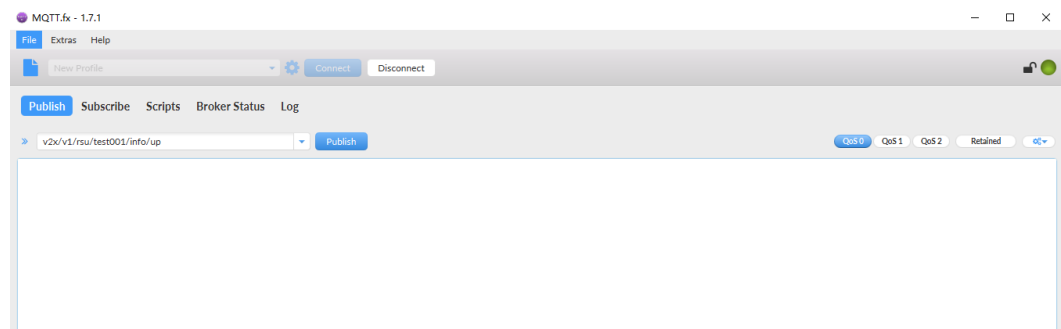
参数名称	说明
Broker Address	步骤1获取到的域名信息中的ip部分，例如135.131.3.6。
Broker Port	步骤1获取到的域名信息中的端口部分，为8883。
Client ID	步骤3生成的ClientId的值。
User Name	步骤3生成的Username的值。
Password	步骤3生成的Password的值。

步骤10 下载并配置证书

进入路网数字化服务页面，选择“系统管理->服务信息”，下载CA证书。



- 步骤11 单击“Connect”，设备鉴权。
- 步骤12 填写topic，此处以RSU注册消息为例，topic填写为v2x/v1/rsu/\${device_id}/info/up，其中\${device_id}为注册RSU设备时填写的设备编码。



- 步骤13 填写RSU注册信息。
- 请求参数

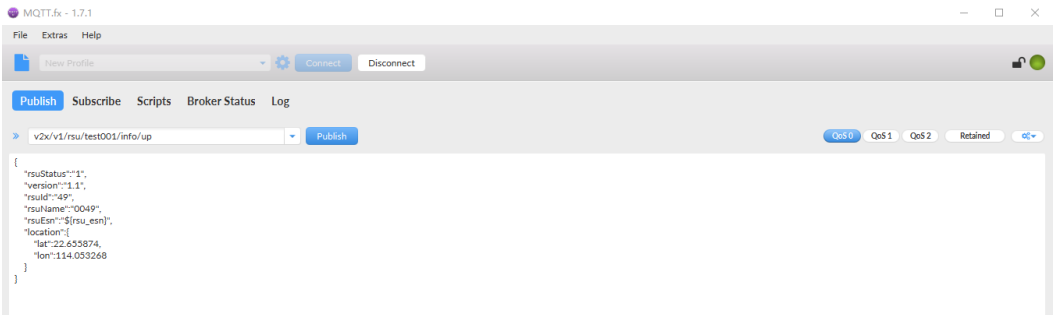
字段名	字段类型	是否可选	参数描述
rsuStatus	string	必选	RSU状态。此处填写为1，表示RSU在线。
version	string	必选	版本号。用于区分不同版本的消息，调测无需关注，按样例填写即可。
rsuld	string	必选	RSU设备ID。RSU设备的固有属性值，调测无需关注，按样例填写即可。
rsuName	string	必选	RSU设备名称。填写为注册RSU设备时的设备名称。
rsuESN	string	必选	RSU设备编码。填写为注册RSU设备时的设备编码。
location	location	必选	RSU设备的经纬度信息。事件下发时需要根据经纬度信息匹配事件发生地点影响半径内的RSU。

location参数结构定义：

字段名	类型	必选/可选	参数描述
lon	Int	必选	经度。例如：“lon”为114.061552
lat	Int	必选	纬度。例如：“lat”为22.651246

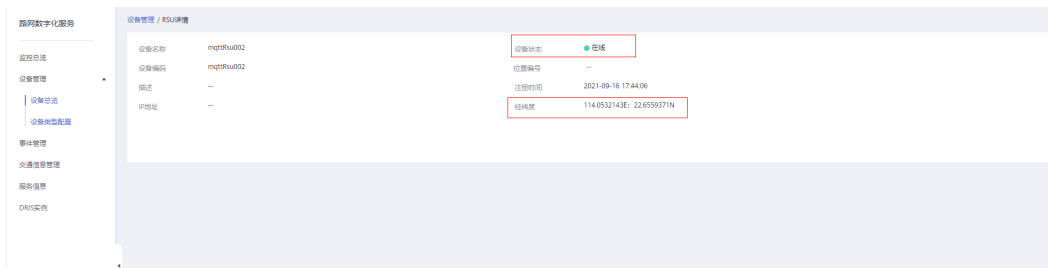
请求示例

```
{
  "rsuStatus": "1",
  "version": "1.1",
  "rsuld": "49",
  "rsuName": "0049",
  "rsuEsn": "${rsu_esn}",
  "location": {
    "lat": 22.655874,
    "lon": 114.053268
  }
}
```



The screenshot shows the MQTT.fx 1.7.1 application window. The 'Publish' tab is active, and the message content is displayed in a text area. The message is a JSON object with the following structure: { "rsuStatus": "1", "version": "1.1", "rsuld": "49", "rsuName": "0049", "rsuEsn": "\${rsu_esn}", "location": { "lat": 22.655874, "lon": 114.053268 } }. The 'Publish' button is visible, and the message is being sent to the topic 'v2x/v1/rsu/test001/info/up'.

- 步骤14** 单击“Publish”，即可在路网数字化服务中看到RSU设备处于在线状态，并且经纬度与注册消息的值相同。进入路网数字化服务页面，选择“设备管理 > 设备总览 > 路侧RSU”，选择前述步骤注册的RSU，点击“详情”，查看RSU的“设备状态”和“经纬度”，可以看到设备处于在线状态，并且经纬度刷新为注册消息中携带的值。



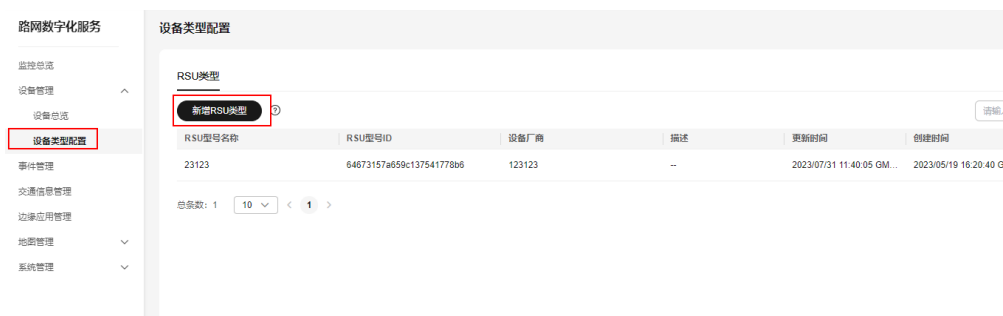
----结束

2.4 快速配置 RSU 设备类型

注册路侧RSU必须预先配置设备类型。

操作步骤

- 步骤1** 进入路网数字化服务控制台，单击“设备管理 > 设备类型配置 > RSU类型 > 新增RSU类型”。



- 步骤2** 填写“设备类型名称”。

输入中文、数字、字母、下划线、_?'#(),.&%@!-长度不能超出64位字符。

- 步骤3** 填写“设备厂商”。

输入中文、数字、字母、下划线、_?'#(),.&%@!-长度不能超出32位字符。

- 步骤4** 填写“描述”。

该设备的描述信息，由用户自定义。

- 步骤5** 单击“确认”，完成配置。

----结束

2.5 ITS800 连接到 V2Xserver

2.5.1 环境确认

进行注册边缘Edge之前，请先确认环境。

说明

适用版本：ITS800固件版本高于8.1.0

步骤1 确认操作系统和tar命令

1. 确认操作系统版本为openEuler22.03，查看操作系统版本，执行命令：

```
cat /etc/os-release
```

```
[root@localhost ~]# cat /etc/os-release
NAME="openEuler"
VERSION="22.03 LTS"
ID="openEuler"
VERSION_ID="22.03"
PRETTY_NAME="openEuler 22.03 LTS"
ANSI_COLOR="0;31"
```

2. 确认是否有tar命令，若没有tar命令则需要安装

- a. 下载tar包的rpm文件，下载地址。

<https://repo.openeuler.org/openEuler-22.03-LTS/everything/aarch64/Packages/>

- b. 使用rpm安装tar。

```
rpm -ivh tar-1.34-1.oe2203.aarch64.rpm
```

步骤2 确认系统是否安装docker。

步骤3 执行命令

```
docker --version
```

```
[root@v2xedge-wl1-6 TEPCS]# docker --version
Docker version 18.09.0, build 8f92b96
```

如果提示命令不存在，则需要安装docker

```
dxc@DESKTOP-IVA74KM:~$ docker --version
Command 'docker' not found, but can be installed with:
sudo apt install docker.io      # version 20.10.12-0ubuntu4, or
sudo apt install podman-docker  # version 3.4.4+ds1-1ubuntu1
```

docker软件包：docker-18.09.6.tgz

docker安装步骤如下：

1. 解压二进制包。

```
tar -xvf docker-18.09.6.tgz
```

```
docker/
docker/ctr
docker/containerd-shim
docker/containerd
docker/docker-proxy
docker/docker
docker/dockerd
docker/runc
docker/docker-init
```

2. 将解压后的docker二进制文件复制到/usr/bin目录下。

```
cp docker/* /usr/bin
```

3. systemd管理docker。

```
cat > /usr/lib/systemd/system/docker.service << EOF
```

```
[Unit]
Description=Docker Application Container Engine
Documentation=https://docs.docker.com
After=network-online.target firewalld.service
Wants=network-online.target

[Service]
Type=notify
ExecStart=/usr/bin/dockerd
ExecReload=/bin/kill -s HUP $MAINPID
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
TimeoutStartSec=0
Delegate=yes
KillMode=process
Restart=on-failure
StartLimitBurst=3
StartLimitInterval=60s

[Install]
WantedBy=multi-user.target
EOF
```

4. 启动并设置开机启动。

```
systemctl daemon-reload
systemctl start docker
systemctl enable docker
```

查看安装结果。

```
docker version
```

```
Client: Docker Engine - Community
 Version:           18.09.0
 API version:       1.39
 Go version:        go1.10.4
 Git commit:        4d60db4
 Built:             Wed Nov  7 00:46:51 2018
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:           18.09.0
  API version:       1.39 (minimum version 1.12)
  Go version:        go1.10.4
  Git commit:        4d60db4
  Built:             Wed Nov  7 00:52:55 2018
  OS/Arch:           linux/amd64
  Experimental:      false
```

步骤4 docker环境确认。

当前ITS800对容器间通信及提权操作做了限制，导致纳管时发生错误，按如下命令检查对应配置：

```
1 /usr/lib/systemd/system/docker.service
2 /usr/lib/systemd/system/docker.service
3 /usr/lib/systemd/system/docker.service
4 /usr/lib/systemd/system/docker.service
5 /usr/lib/systemd/system/docker.service
6 /usr/lib/systemd/system/docker.service
7 /usr/lib/systemd/system/docker.service
8 /usr/lib/systemd/system/docker.service
9 /usr/lib/systemd/system/docker.service
10 /usr/lib/systemd/system/docker.service
11 /usr/lib/systemd/system/docker.service
12 /usr/lib/systemd/system/docker.service
13 /usr/lib/systemd/system/docker.service
14 /usr/lib/systemd/system/docker.service
15 /usr/lib/systemd/system/docker.service
16 /usr/lib/systemd/system/docker.service
17 /usr/lib/systemd/system/docker.service
18 /usr/lib/systemd/system/docker.service
19 /usr/lib/systemd/system/docker.service
20 /usr/lib/systemd/system/docker.service
21 /usr/lib/systemd/system/docker.service
22 /usr/lib/systemd/system/docker.service
23 /usr/lib/systemd/system/docker.service
24 /usr/lib/systemd/system/docker.service
25 /usr/lib/systemd/system/docker.service
26 /usr/lib/systemd/system/docker.service
27 /usr/lib/systemd/system/docker.service
28 /usr/lib/systemd/system/docker.service
29 /usr/lib/systemd/system/docker.service
30 /usr/lib/systemd/system/docker.service
31 /usr/lib/systemd/system/docker.service
32 /usr/lib/systemd/system/docker.service
33 /usr/lib/systemd/system/docker.service
34 /usr/lib/systemd/system/docker.service
35 /usr/lib/systemd/system/docker.service
36 /usr/lib/systemd/system/docker.service
37 /usr/lib/systemd/system/docker.service
38 /usr/lib/systemd/system/docker.service
39 /usr/lib/systemd/system/docker.service
40 /usr/lib/systemd/system/docker.service
41 /usr/lib/systemd/system/docker.service
42 /usr/lib/systemd/system/docker.service
43 /usr/lib/systemd/system/docker.service
44 /usr/lib/systemd/system/docker.service
45 /usr/lib/systemd/system/docker.service
46 /usr/lib/systemd/system/docker.service
47 /usr/lib/systemd/system/docker.service
48 /usr/lib/systemd/system/docker.service
49 /usr/lib/systemd/system/docker.service
50 /usr/lib/systemd/system/docker.service
51 /usr/lib/systemd/system/docker.service
52 /usr/lib/systemd/system/docker.service
53 /usr/lib/systemd/system/docker.service
54 /usr/lib/systemd/system/docker.service
55 /usr/lib/systemd/system/docker.service
56 /usr/lib/systemd/system/docker.service
57 /usr/lib/systemd/system/docker.service
58 /usr/lib/systemd/system/docker.service
59 /usr/lib/systemd/system/docker.service
60 /usr/lib/systemd/system/docker.service
61 /usr/lib/systemd/system/docker.service
62 /usr/lib/systemd/system/docker.service
63 /usr/lib/systemd/system/docker.service
64 /usr/lib/systemd/system/docker.service
65 /usr/lib/systemd/system/docker.service
66 /usr/lib/systemd/system/docker.service
67 /usr/lib/systemd/system/docker.service
68 /usr/lib/systemd/system/docker.service
69 /usr/lib/systemd/system/docker.service
70 /usr/lib/systemd/system/docker.service
71 /usr/lib/systemd/system/docker.service
72 /usr/lib/systemd/system/docker.service
73 /usr/lib/systemd/system/docker.service
74 /usr/lib/systemd/system/docker.service
75 /usr/lib/systemd/system/docker.service
76 /usr/lib/systemd/system/docker.service
77 /usr/lib/systemd/system/docker.service
78 /usr/lib/systemd/system/docker.service
79 /usr/lib/systemd/system/docker.service
80 /usr/lib/systemd/system/docker.service
81 /usr/lib/systemd/system/docker.service
82 /usr/lib/systemd/system/docker.service
83 /usr/lib/systemd/system/docker.service
84 /usr/lib/systemd/system/docker.service
85 /usr/lib/systemd/system/docker.service
86 /usr/lib/systemd/system/docker.service
87 /usr/lib/systemd/system/docker.service
88 /usr/lib/systemd/system/docker.service
89 /usr/lib/systemd/system/docker.service
90 /usr/lib/systemd/system/docker.service
91 /usr/lib/systemd/system/docker.service
92 /usr/lib/systemd/system/docker.service
93 /usr/lib/systemd/system/docker.service
94 /usr/lib/systemd/system/docker.service
95 /usr/lib/systemd/system/docker.service
96 /usr/lib/systemd/system/docker.service
97 /usr/lib/systemd/system/docker.service
98 /usr/lib/systemd/system/docker.service
99 /usr/lib/systemd/system/docker.service
100 /usr/lib/systemd/system/docker.service
```

分别删除以下两行：

```
--no-new-privileges
--icc=false
```

重启docker.service，重启ITS800，此时若docker服务未正确启动（docker images返回全空）则启动docker.service。

重启docker.service命令：

```
systemctl daemon-reload
systemctl restart docker.service
```

步骤5 打开ip转发。

执行命令：

```
cat /proc/sys/net/ipv4/ip_forward
```

Ip转发配置如果显示为0，需手动设置为1。

步骤6 配置关闭docker.service自动还原功能。

修改“/home/ivstool/init.d/fix_docker_config.sh”文件，某些版本该脚本位于路径“/home/ivstool/bin/”路径下。

删除大红框标注的内容如下图：

```
function editDockerService () {
    cat /usr/lib/systemd/system/docker.service |grep default-umit
    if [ $? -ne 0 ]; then
        sed -i '/ExecReload/i\' --default-umit nproc=1024:2048 --default-umit nofile=65535:65535' /usr/lib/systemd/system/docker.service
    else
        sed -i "s~default-umit.*~default-umit nproc=1024:2048 --default-umit nofile=65535:65535~g" /usr/lib/systemd/system/docker.service
    fi

    cat /usr/lib/systemd/system/docker.service |grep icc
    if [ $? -ne 0 ]; then
        sed -i '/default-umit/i\' --icc=false \\' /usr/lib/systemd/system/docker.service
    fi

    cat /usr/lib/systemd/system/docker.service |grep graph
    if [ $? -ne 0 ]; then
        sed -i '/default-umit/i\' --graph /opt/third_algorithm_D \\' /usr/lib/systemd/system/docker.service
    fi

    cat /usr/lib/systemd/system/docker.service |grep userland-proxy
    if [ $? -ne 0 ]; then
        sed -i '/default-umit/i\' --userland-proxy=false \\' /usr/lib/systemd/system/docker.service
    fi

    cat /usr/lib/systemd/system/docker.service |grep no-new-privileges
    if [ $? -ne 0 ]; then
        sed -i '/default-umit/i\' --no-new-privileges \\' /usr/lib/systemd/system/docker.service
    fi

    sed -i '/INSECURE_REGISTRY/s/^.*$/INSECURE_REGISTRY \\' /usr/lib/systemd/system/docker.service
    systemctl daemon-reload
    service docker restart
    cont=$(ls /home/ivs_sudo/thirdApp/alg_model/)
    if [ -z "$cont" ]; then
        service docker stop
    fi
}
```

步骤7 配置docker信任仓库

修改“/etc/docker/daemon.json”。

如果没有则创建一个：

```
vi /etc/docker/daemon.json
```

在其中写入：

```
{
  "insecure-registries":["swr.cn-north-4.myhuaweicloud.com"]
}
```

保存后执行：

```
systemctl daemon-reload
systemctl restart docker
```

步骤8 检查dns及网络配置。

应确认设备能连上外网且能解析华为云的域名。

执行命令：

```
ping swr.cn-north-4.myhuaweicloud.com
```

正常情况下应能解析该地址，地址禁ping，ping不通为正常现象。

“ping www.baidu.com”能通说明外网连接正常。

无法解析域名的情况下：

执行命令：

```
cat /etc/resolv.conf
```

查看是否有配置有效dns，如果有有效dns但不能成功解析，则尝试将多余的dns地址删除掉，只保留有效dns。

步骤9 确认当前系统所在硬盘。

执行命令：

```
df -h
```

在打印出的信息中找到根目录(/) 和2块系统盘间的mount关系。

在系统版本8.1.0的ITS800上，该目录为“/dev/mmcblk0p3”与“/dev/mmcblk0p2”其中之一。

建议截图记录。后续若发生异常可用于定位分区倒换的问题。

```
Euler:/opt/v2xedge/log # df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.3G   0  3.3G   0% /dev
tmpfs           3.5G  548K  3.5G   1% /dev/shm
tmpfs           3.5G  129M  3.3G   4% /run
tmpfs           3.5G   0  3.5G   0% /sys/fs/cgroup
/dev/mmcblk0p3  2.0G  874M  961M  48% /
tmpfs           3.5G   40K  3.5G   1% /tmp
tmpfs          128M   0  128M   0% /var/IEF
tmpfs           64M   5.3M   59M   9% /var/log
/dev/mmcblk0p8  5.9G   24M   5.6G   1% /StaticFeature
tmpfs          128M   0  128M   0% /var/plog
tmpfs          128M   8.0K  128M   1% /var/dlog
tmpfs           64M   0   64M   0% /var/tmp
/dev/mmcblk0p6  6.4G   3.2G   2.9G  54% /home
tmpfs          128M   0  128M   0% /var/alog
/dev/mmcblk0p4  976M  406M  503M  45% /home/data
/dev/mmcblk0p7  9.8G   8.3G  1018M  90% /DB_DATA
/dev/mmcblk0p5  976M   65M  845M   8% /home/log
/dev/md0p1     129G   6.4G  116G   6% /opt
/dev/md0p2     21G    45M   19G   1% /opt/third_algorithm
/dev/md0p3     5.5G    23M   5.2G   1% /var/lib/docker
/dev/md0p4     25G    2.4G   21G  11% /opt/third_algorithm_D
overlay        25G    2.4G   21G  11% /opt/third_algorithm_D/overlay2/71305bff25218741
```

执行指令：

```
fw_printenv
```

可查看当前指定的启动分区与实际启动的分区是否一致，若不一致则可能出现了硬件问题。

```
Euler:/opt/v2xedge/log # fw_printenv
arch=arm
baudrate=115200
board=hi3559av100
board_name=hi3559av100
boot_index=b
bootargs=cma=90M mem=7168M console=ttyAMA0,115200
x4a000000,256M cmdlinepart.mtdparts=hi sfc:1M(mi
```

“/dev/mmcblk0p3” 对应 “boot_index=b”

“/dev/mmcblk0p2” 对应 “boot_index=a”

----结束

2.5.2 清除 IEF

如果该节点曾用IEF纳管，清除IEF。

说明

全新的机器或未使用过IEF纳管，则跳过此步。

执行命令：

```
cd /opt/edge-installer; sudo ./installer -op=uninstall  
systemctl status edgecore
```

提示无服务则完成。

清除容器命令：

```
docker ps
```

若查询到活动的edge相关镜像，则用输入以下命令将这些容器清除：

```
docker rm -f 容器id
```

2.5.3 启动边缘节点安装命令

用SSH工具登录ITS800后台，切换至root权限，并按文档最后的备注检查ITS800后台环境。

说明

请先[环境确认](#)再进行后续安装步骤。

步骤1 执行前检查环境上SSL是否正常，执行命令。

```
openssl
```

若提示“找不到libcrypto.so等库”，则先执行以下命令：

```
export LD_LIBRARY_PATH=/usr/lib64/
```

检查完SSL后，执行前一步中获取到的“边缘节点安装命令”进行安装。

步骤2 等待安装完成执行以下命令可看到容器。

```
docker ps
```

步骤3 安装完成后，增加网桥的可用性检查次数，避免网桥重复创建。

```
echo "60" > /opt/third_algorithm/IoTEdge/edgeDaemon/conf/network_check.file
```

步骤4 如果操作系统是openEuler22.03确认容器间网络是否是通的（容器都能安装成功，tepa连接mqtt失败），网络不通关闭防火墙。

```
firewall-cmd --zone=trusted --add-interface=br-<iot-edge-bridge网桥ID> --permanent  
firewall-cmd --zone=trusted --add-masquerade --permanent  
firewall-cmd --zone=public --add-masquerade --permanent  
firewall-cmd --reload
```

可以使用如下命令来获取实际的iot-edge-bridge网桥ID。

```
docker inspect iot-edge-bridge | grep "Id"
```

步骤5 创建占位文件，执行命令。

```
touch /home/ivs_sudo/thirdApp/alg_model/iotedge
```

----结束

2.6 快速管理第三方边缘应用

在部署边缘Edge应用前，需要先添加边缘应用。

添加应用

步骤1 进入路网数字化服务控制台，选择“边缘应用管理 > 添加应用”。



步骤2 填写应用名称及描述信息。

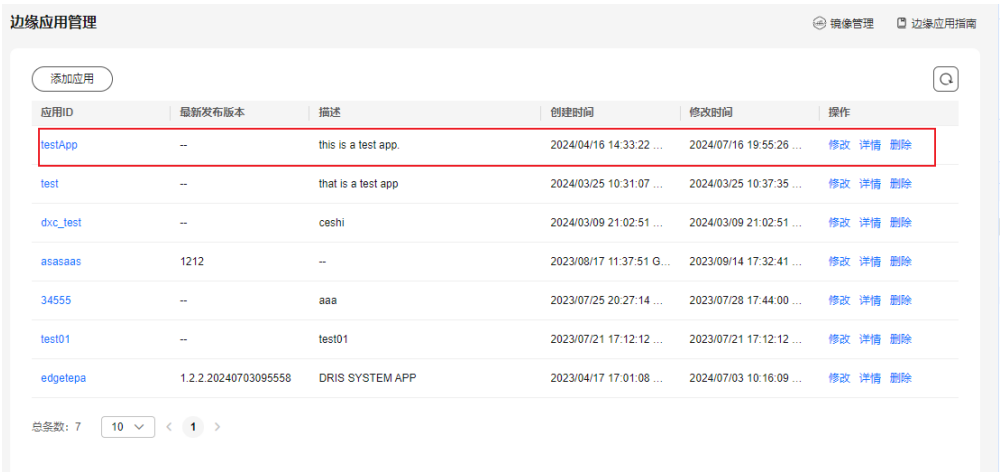
步骤3 单击“保存”，在边缘应用管理页面可查看创建好的应用，单击应用名称可查看应用详情。

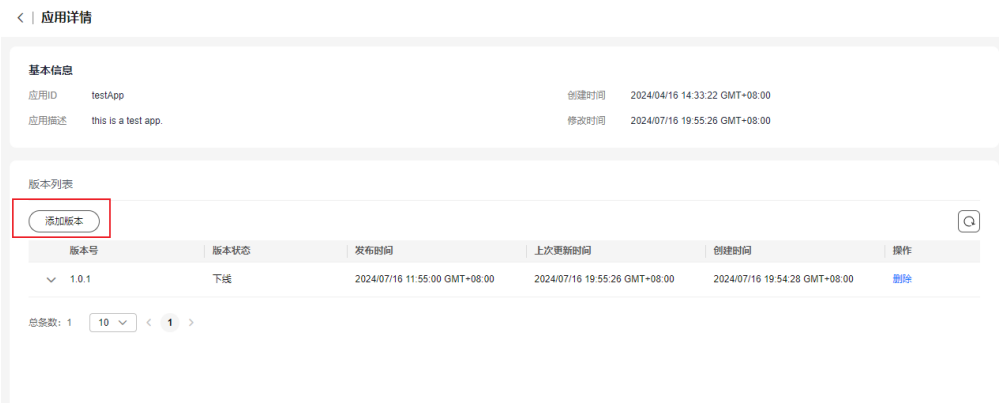
----结束

添加版本

应用添加成功后，需要设置应用对应的版本信息。

步骤1 选择“边缘应用管理 > 详情 > 添加版本”，设置版本基本信息。



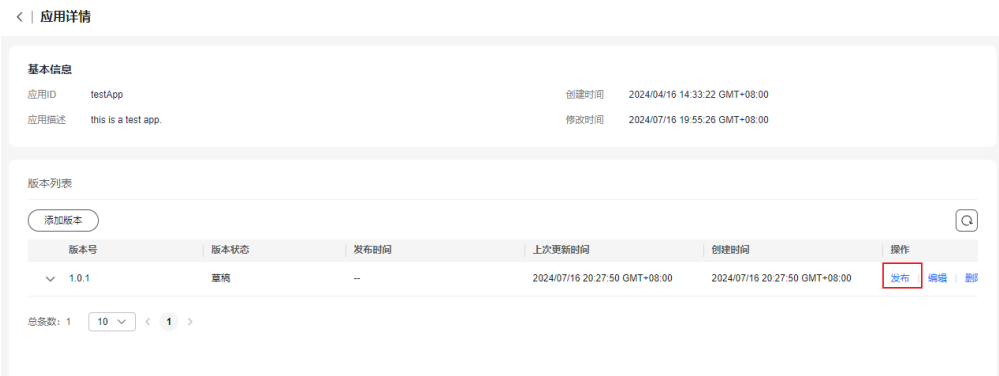


- 步骤2** 填写“应用版本”。
- 输入标识该应用的版本信息。
- 步骤3** 选择“镜像”。
- 下拉列表，根据实际情况选择我的镜像或他人共享的镜像。

说明

如果没有对应镜像，请单击容器镜像服务前往上传镜像，根据《[容器镜像服务相关指导](#)》上传镜像。

- 步骤4** 选择“支持架构”。
- 下拉列表选择架构信息。
- 步骤5** 设置“容器规格”。
- 根据实际情况配置CPU、内存、GPU、NPU的申请配额及限制配额。
- 步骤6** 单击“下一步”，进行部署配置。
- 步骤7** 设置“重启策略”，单击“提交”，完成版本的添加。
- 总是重启：当应用退出时，无论是正常退出还是异常退出，系统都会重新启动应用。
 - 失败时重启：当应用异常退出时，系统会重新启动应用。
 - 从不重启：当应用退出时，无论是正常退出还是异常退出，系统不会重新启动应用。
- 步骤8** 回到应用详情页面，单击“发布”。



说明

- 未发布的版本为草稿，支持编辑及发布。
- 已发布的版本，不可编辑，且版本为发布状态，无法删除应用。
- 发布的版本，可以申请下线，下线后，可重新发布。

----结束

2.7 在线调试

为了降低应用侧的开发难度、提升应用侧开发效率，物联网平台向应用侧开放了丰富的API。

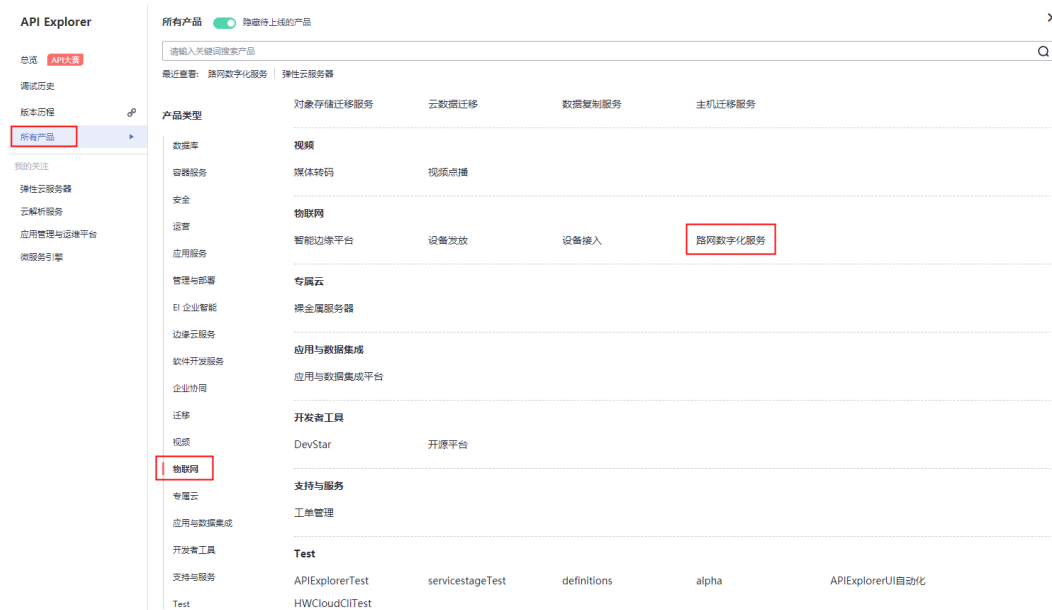
本文档以在线调试（即API Explorer）为例，使用API Explorer可以无码化快速调用接口。

API Explorer提供在线API检索及接口调试，您可以使用在线调试快速接入物联网平台。

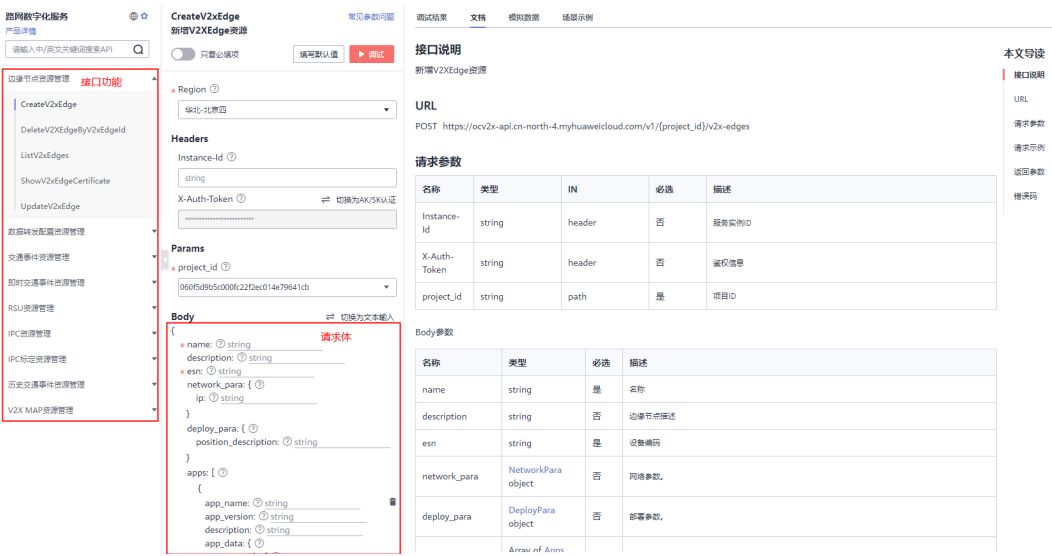
操作步骤

步骤1 打开API Explorer。

步骤2 选择“物联网 > 路网数字化服务”。



步骤3 选择需要调试的接口进行调试。

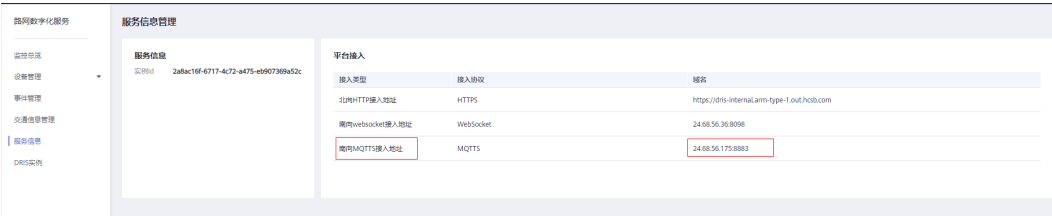


----结束

2.8 使用 MQTT.fx 调测

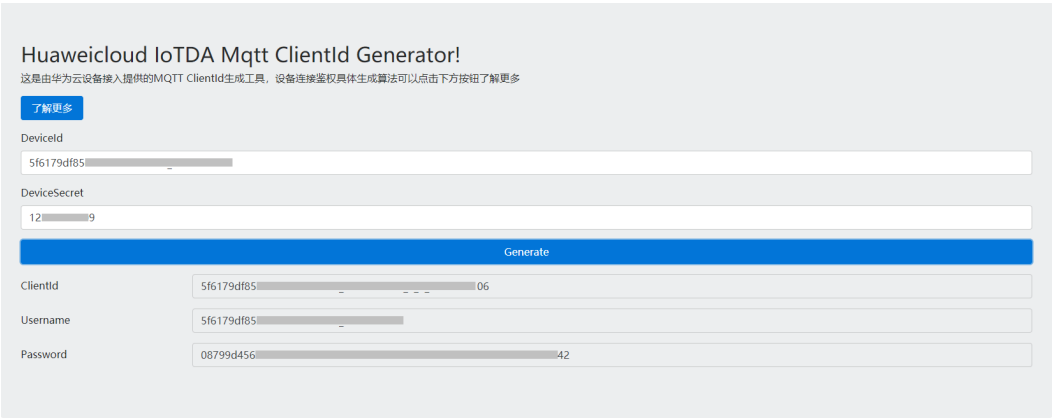
操作步骤

步骤1 进入路网数字化服务页面，选择“服务信息”。记录“平台接入”下的“南向MQTTS接入地址”行的“域名”信息。

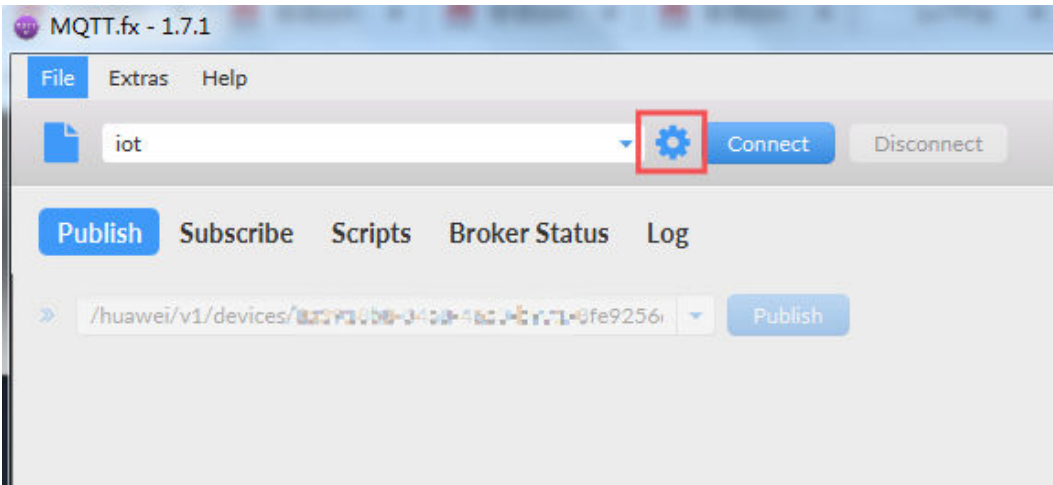


步骤2 访问[MQTT.fx下载页面](#)，下载并安装最新版本的MQTT.fx工具。

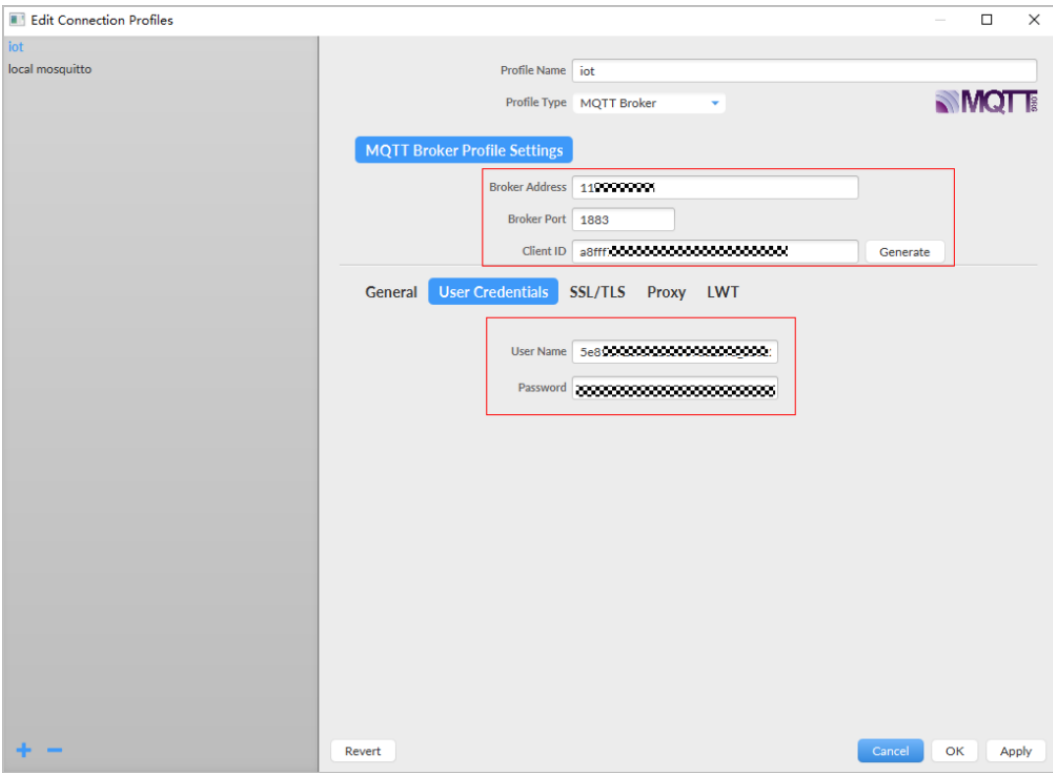
步骤3 请访问[这里](#)，填写DeviceId和DeviceSecret，生成连接信息（ClientId、Username、Password）。



步骤4 打开MQTT.fx软件，单击设置图标。



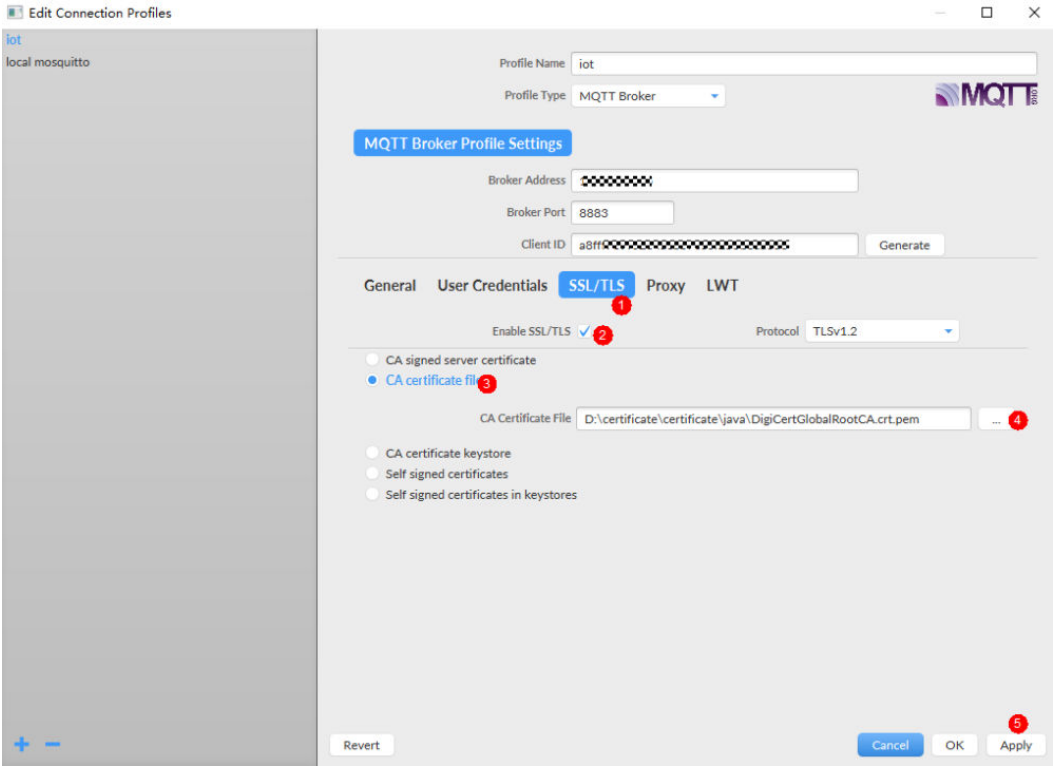
步骤5 填写路网数字化服务南向MQTT接入地址，并配置鉴权参数。



参数名称	说明
Broker Address	步骤1获取到的域名信息中的ip部分，例如135.131.3.6。
Broker Port	步骤1获取到的域名信息中的端口部分，为1883。
Client ID	步骤3生成的ClientId的值。
User Name	步骤3生成的Username的值。
Password	步骤3生成的Password的值。

 说明

如果您选择安全方式接入，Broker Port设置为8883，下载并获取[证书](#)，加载Java语言pem格式的证书。



步骤6 单击“Connect”，设备鉴权。
----结束

3 快速玩转路网数字化平台

3.1 手动事件下发示例

路网数字化服务场景中的大部分事件会由设备侧根据天气和路况等情况，自行识别交通事件上报至平台并转发给车载单元，同时路网数字化服务也支持用户手动下发事件。

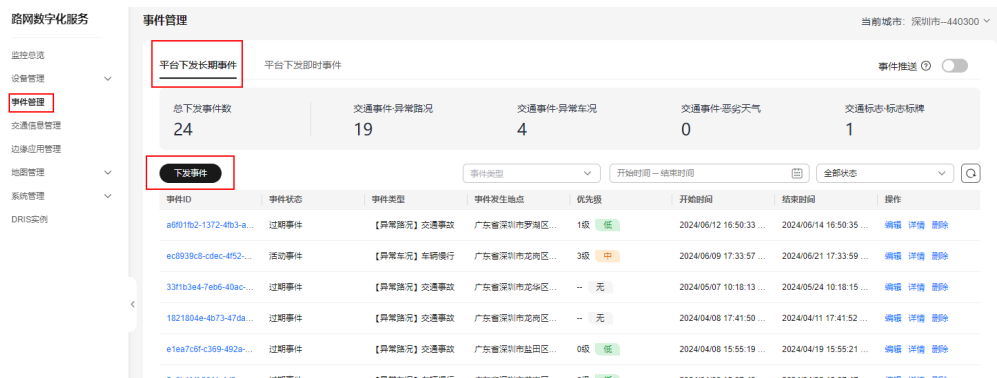
路网数字化服务覆盖《合作式ITS车用通信系统应用层及应用数据交互标准》中306种V2X事件（249种标志标牌及57种交通事件），典型交通事件单击[产品规格](#)查看。

本节内容为您介绍在对接完设备之后，如何使用路网数字化平台进行手动事件下发。

操作步骤

步骤1 下发事件需要完成[RSU设备接入](#)，并且已获取事件类型、事件发生地点、影响范围、事件发生时间以及事件等级。

步骤2 在左侧导航栏，选择“事件管理 > 平台下发长期事件 > 下发事件”。



步骤3 填写“事件类型”。

选择事件发生类型。若用户需要的类型未在界面上呈现，也可通过事件名称和编码进行搜索查询。

步骤4 填写“事件数值信息”。

事件数值信息会根据事件类型的选择而呈现不同的显示。

当交通事件类型为急转弯、道路最高限速、道路最低限速、沙尘暴时，该字段必填。

步骤5 填写“事件发生地点”。

定位事件发生的地点，有三种添加方式：

- 通过“事件发生地点”搜索框查找事件发生的大概位置，支持模糊搜索，搜索成功后左侧地图会自动定位到该区域。若搜索结果存在多个区域需要用户自定义选择在哪个区域。
- 在地图上单击鼠标右键，设置事件发生地点。
- 通过经纬度定位发生地点。

步骤6 填写“事件影响范围”。

设置事件影响的单个或多个范围，所有经过影响范围的已安装T-BOX的车辆都会收到事件预警信息。

在地图上单击鼠标右键分别设置事件发生范围的起点和终点。

步骤7 选择“优先级”。

优先级的划分：0-2级优先级为低，3-5级为中，6-7级为高。由低到高代表事件造成的影响从轻微拥堵到水泄不通。

步骤8 设置“事件发生时间”。

设置事件发生时间，分别设置起始时间和结束时间。

步骤9 填写“事件描述”。

填写事件描述信息，T-BOX侧收到的事件描述。

步骤10 填写“备注信息”。

填写事件备注信息，该字段只提供用户对该事件进行备注，不会下发到T-BOX。

步骤11 单击“确认下发”，完成事件下发。

----结束