

IoT 边缘 IoTEdge

最佳实践

文档版本 1.4
发布日期 2024-06-27



版权所有 © 华为云计算技术有限公司 2024。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 操作前准备	1
2 基于 IoT 边缘实现进程包镜像插件部署	6
2.1 实验简介	6
2.2 前提条件	6
2.2.1 前提概要	6
2.2.2 上传进程包到 OBS	6
2.3 操作步骤	8
2.3.1 创建边缘节点	8
2.3.2 (可选) 部署边缘应用	15
2.3.3 创建自定义应用	15
2.3.4 部署进程包插件	18
2.3.5 资源释放	20
3 基于 IoT 边缘实现容器应用镜像插件部署	23
3.1 实验简介	23
3.2 前提条件	23
3.2.1 前提概要	23
3.2.2 上传容器应用镜像到 SWR	23
3.3 操作步骤	24
3.3.1 创建边缘节点	24
3.3.2 (可选) 部署边缘应用	31
3.3.3 创建自定义应用	31
3.3.4 部署容器应用插件	36
3.3.5 资源释放	37
4 基于 IoT 边缘实现模拟 MQTT 设备的快速接入	40
4.1 实验简介	40
4.2 前提条件	40
4.2.1 前提概要	40
4.2.2 安装 MQTT 设备模拟器	41
4.3 操作步骤	41
4.3.1 创建边缘节点	41
4.3.2 (可选) 部署边缘应用	48
4.3.3 设备接入边缘节点 (MQTT)	48

4.3.4 资源释放.....	59
5 基于 IoT 边缘服务实现边缘插件的生态拓展.....	62
5.1 实验简介.....	62
5.2 前提条件.....	62
5.2.1 环境要求.....	62
5.2.2 下载 SDK.....	62
5.2.3 开发插件.....	63
5.2.4 打包工程.....	64
5.2.5 打包容器镜像.....	65
5.2.6 上传镜像.....	66
5.3 操作步骤.....	67
5.3.1 注册边缘节点.....	67
5.3.2 添加插件应用.....	74
5.3.3 远程部署.....	78
5.3.4 接入模拟设备.....	79
6 基于 IoT 边缘实现 OT 数采（OPCUA 协议）.....	88
6.1 实验简介.....	88
6.2 操作步骤.....	88
6.2.1 创建边缘节点.....	88
6.2.2 准备数据源.....	96
6.2.3 部署边缘数采应用.....	98
6.2.4 设备建模&发放.....	100
6.2.5 OT 数采配置.....	106
6.2.6 查看采集结果.....	115
6.2.7（可选）点位拆分.....	116
6.2.8（可选）点位计算.....	120
6.2.9（可选）点位阈值.....	125
7 基于 IoT 边缘实现模拟 IEC104 网关的接入.....	128
7.1 实验简介.....	128
7.2 操作步骤.....	128
7.2.1 创建边缘节点.....	128
7.2.2 准备数据源.....	136
7.2.3 部署边缘数采应用.....	138
7.2.4 设备建模&发放.....	139
7.2.5 OT 数采配置.....	145
7.2.6 查看采集结果.....	155
8 基于 IoT 边缘实现视频数采（Onvif 协议）.....	156
8.1 实验简介.....	156
8.2 操作步骤.....	156
8.2.1 创建边缘节点.....	156
8.2.2 部署边缘视频数采应用.....	164

8.2.3 添加视频设备.....	165
8.2.4 视频预览.....	167
8.2.5 流媒体服务器搭建（可选）.....	169
8.2.5.1 安装 nginx+http-flv(rtmp 服务器).....	169
8.2.6 监控视频数据推送（可选）.....	172
9 基于 IoT 边缘实现 OT 数采（Modbus-TCP）.....	174
9.1 实验简介.....	174
9.2 操作步骤.....	174
9.2.1 创建边缘节点.....	174
9.2.2 准备数据源.....	182
9.2.3 部署边缘数采应用.....	185
9.2.4 设备建模&发放.....	186
9.2.5 OT 数采配置.....	191
9.2.6 查看采集结果.....	203
9.3 常见问题.....	204
9.3.1 连接状态未连接.....	204
9.3.1.1 服务端点格式不正确.....	204
9.3.1.2 配置信息配置有误.....	204
9.3.1.3 有效地址范围重复.....	205
9.3.2 网络问题排查.....	206
9.3.2.1 连接拒绝.....	206
9.3.2.2 连接超时.....	206
9.3.3 点位采集失败.....	207
9.3.3.1 点位数据类型与寄存器类型不匹配或点位未下发.....	207
9.3.3.2 点位超出有效地址范围.....	208
9.3.3.3 坏点.....	208
9.3.4 点位下写失败.....	209
9.3.4.1 下写数据类型与配置类型不匹配.....	209
9.3.4.2 下写不可写寄存器.....	210
9.3.4.3 未知数据类型下写.....	210
9.3.4.4 下写数据结果与预期不一致.....	210
9.3.5 修改数据配置并下发，数据源配置未更新.....	210
10 基于 IoT 边缘实现业务应用自定义配置.....	212
10.1 实验简介.....	212
10.2 操作步骤.....	213
10.2.1 创建边缘节点.....	213
10.2.2 导入自定义业务应用配置模板.....	220
10.2.3 新增自定义应用以及版本.....	223
10.2.4 部署自定义应用.....	233
10.2.5 应用模块配置.....	234
10.2.6 查看配置结果.....	239

11 基于 IoT 边缘实现云原生应用部署	241
11.1 实验简介	241
11.2 前提条件	241
11.2.1 前提概要	241
11.2.2 上传应用镜像到 SWR	241
11.3 操作步骤	243
11.3.1 创建边缘集群	243
11.3.2 创建应用模板	246
11.3.3 添加应用版本	247
11.3.4 部署云原生应用到边缘集群	249
11.3.5 资源释放	252
12 基于 IoT 边缘企业版实现集群本地运维管理	255
12.1 实验简介	255
12.2 前提条件	255
12.3 操作步骤	257
12.3.1 部署本地控制台网关	257
12.3.2 部署监控告警组件	258
12.3.3 部署本地镜像仓库	259
12.3.4 部署物联接入	262
12.3.5 登录边缘本地控制台	264

1 操作前准备

1. 已注册[华为云](#)账号，且已完成实名认证。
2. 已开通[设备接入IoTDA服务](#)和[IoT边缘IoTEdge服务](#)。
访问[设备接入IoTDA服务](#)产品首页，单击“立即使用”，即可自助开通服务。
访问[IoT边缘IoTEdge服务](#)产品首页，单击“管理控制台”，进入IoT边缘服务控制台，单击页面右上角“开通服务”，即可自助开通服务。
3. 已准备一台边缘节点服务器（需[安装Docker](#)，版本>17.06），例如Atlas 500。如果没有，可购买弹性云服务器ECS作为边缘节点，参考[（可选）购买边缘节点硬件](#)。

（可选）购买边缘节点硬件

边缘节点硬件需要满足一定的规格要求，具体请参见[边缘节点硬件规格要求](#)。本实验通过购买弹性云服务器ECS作为边缘节点服务器。

步骤1 访问[弹性云服务器 ECS](#)，单击“购买”，进入购买页面。

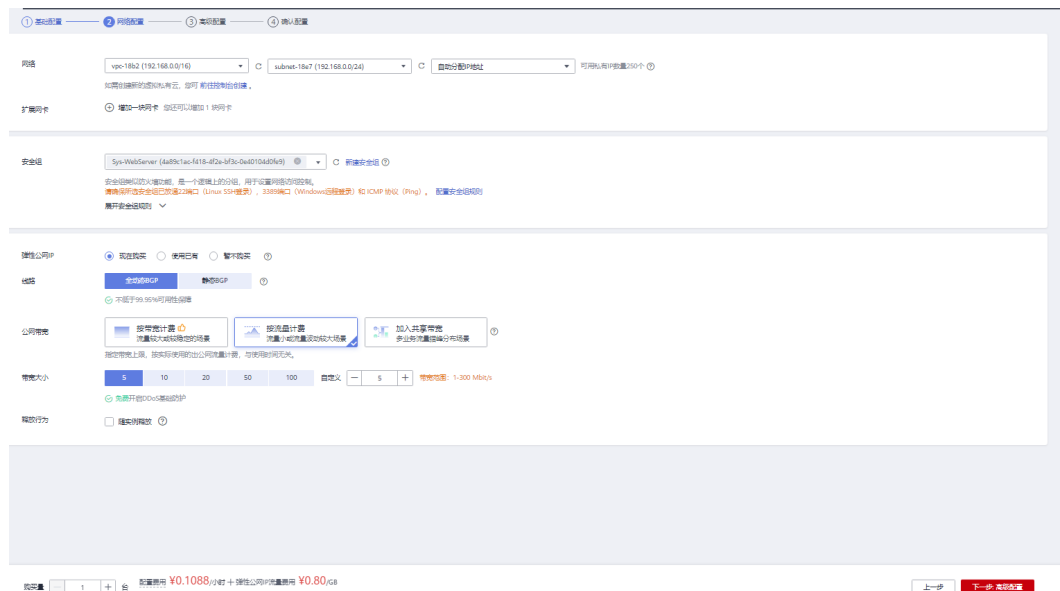
步骤2 填写基础配置信息后，单击“下一步：网络配置”。

The screenshot shows the ECS purchase configuration interface. Key elements include:

- 计费模式:** 包年/包月 (selected), 按需计费, 竞价计费.
- 区域:** 华北-北京四 (selected), 华东-上海二, 华南-广州, etc.
- 可用区:** 默认分配, 可用区1, 可用区2, etc.
- CPU架构:** x86计算 (selected), 鲲鹏计算.
- 规格:** 最新系列, vCPUs: 全部, 内存: 全部. A table lists instance types like c7.large.2 with 2vCPUs | 4 GiB memory.
- 镜像:** 公共镜像 (selected), 私有镜像, 共享镜像, 市场镜像. Image: Ubuntu 16.04 server 64bit(40GB).
- 主机安全:** 开通主机安全 (optional).
- 系统盘:** 高IO, 40 GB, IOPS上限2,120, IOPS突发上限5,000.
- 购买量:** 1, 配置费用 ¥0.5196/小时.
- 下一步: 网络配置** (button).

参数名称	取值示例
计费模式	按需计费
区域	华北-北京四
可用区	随机分配
CPU架构	x86计算
规格	为降低实验成本，可选择“通用入门型 t6.large.2”
镜像	公共镜像 Ubuntu Ubuntu 16.04 server 64bit
主机安全	不勾选“开通主机安全”
系统盘	高IO 40GB

步骤3 填写网络配置信息后，单击“下一步：高级配置”。



基本信息：

- 网络：请刷新后，选择已创建虚拟私有云VPC（vpc-iot）和子网（subnet-iot），自动分配IP地址
- 扩展网卡：默认配置
- 安全组：默认配置
- 弹性公网IP：现在购买
- 路线：默认配置
- 公网宽带：按流量计算
- 宽带大小：为降低实验成本，可降低至5Mbit/s
- 释放行为：不勾选

步骤4 填写高级配置信息后，单击“下一步：确认配置”。

云服务器名称 允许重名

购买多台云服务器时，名称自动按序增加4位数字后缀。例如：输入ecs，从ecs-0001开始命名；若已有ecs-0010，从ecs-0011开始命名。

登录凭证 **密码** 密钥对 创建后设置

用户名 root

密码 请牢记密码，如忘记密码可登录ECS控制台重置密码。

确认密码

云备份 使用云备份服务，需购买备份存储库，存储库是存放服务器产生的备份副本的容器。
 ?

云服务器组（可选） **反亲和性** ?

--请选择云服务器组-- C

[新建云服务器组](#)

高级选项 现在配置

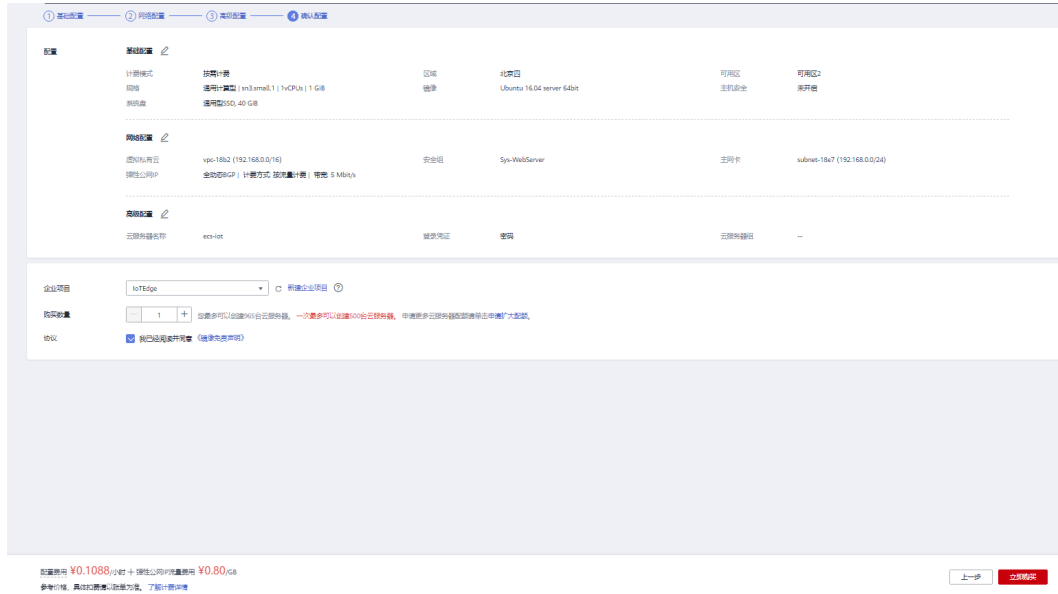
参数名称	说明
云服务器名称	自定义，如ecs-iot
登录凭证	密码
密码/确认密码	自定义，设置远程登录ECS用的密码，如Huawei@123
其他参数	默认配置

步骤5 确认配置信息后，单击“立即购买”。

企业项目：IoTEdge

数量：根据要求填

协议：勾选“我已经阅读并同意（镜像免责声明）”。



步骤6 主机创建完成后，进行主机授信配置。请记录“弹性公网IP”信息，用于后续配置。

1. 单击主机名，如ecs-iot，进入主机详情页面，选择“安全组”页签。
2. 展开Sys-default安全组，单击“更改安全组规则”，选择“入方向规则”，单击“添加规则”。
 - SSH协议端口（22）的入方向TCP协议默认开放，无需修改。
 - 边缘节点需要使用7883和8943端口，为了方便省事，可以选择全部放通。
 - 出方向规则默认不做限制，无需修改。
3. 填写参数信息后，单击“确定”。

添加入方向规则 教我设置



- 协议端口：全部放通
- 源地址：默认配置，IP地址、0.0.0.0/0

步骤7 安装Docker。

1. 单击“远程登录”。



2. 输入用户名和密码，登录Linux系统。

```
Ubuntu 16.04.6 LTS ecs-iot tty1
ecs-iot login: root
Password:
```

- 用户名: root
- 密码: 购买弹性云服务器时配置的用户密码（注：命令行不显示输入的密码，输入后回车即可）

3. 参考<https://docs.docker.com/engine/install/ubuntu/>，安装Docker Engine。下面演示为Ubuntu系统安装docker，其他系统请参考docker官方文档进行安装。

```
//1.更新apt资源
$ sudo apt-get update
//2.安装相关依赖资源
$ sudo apt-get install \
ca-certificates \
curl \
gnupg \
lsb-release
//3.添加Docker官方的GPG key
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/
keyrings/docker-archive-keyring.gpg
//4.添加稳定源
$ echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
//5.安装最新版本的Docker
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
//6.运行hello-world校验Docker是否安装成功
$ sudo docker run hello-world
```

----结束

2 基于 IoT 边缘实现进程包镜像插件部署

2.1 实验简介

本实验将指导您搭建一个基于IoT边缘服务的边缘节点，并在边缘节点上进行进程包插件部署，将用户自定义的进程包插件部署到边缘节点上运行。

通过本实验，您将能够：

- 了解并掌握IoT边缘服务的功能特点与使用方法。
- 掌握通过IoTEdge云服务将进程包插件部署到边缘节点的方法。

2.2 前提条件

2.2.1 前提概要

1. **前提：**体验需要三方配合，首先要开通IoTEdge，用于节点管理和应用管理，再开通IoTDA，用于设备接入与管理，还得有边缘服务器作为节点被IoTEdge纳管。当前章节讲的是进程包，进程包需要开通OBS，将准备的进程包上传至OBS。
2. **边缘服务器：**第一点提到的IoTEdge, IoTDA, OBS是华为云服务，使用华为云租户账号直接免费开通可用，边缘服务器属于硬件，可以购买低配的ECS代替，其EIP则作为边缘服务器的节点IP地址。
3. **在IoTEdge上注册节点：**在购买的ECS上执行安装脚本命令，即自动部署好2个系统应用，agent用于节点应用管理，hub用于数据通信。
4. **进程包插件：**是集成了边缘SDK的应用程序打包而成，可作为边缘节点的自定义应用部署到边缘服务器上，以进程方式运行，可在节点上进行设备数据协议解析、子系统数据解析、数据业务处理、智能计算等。

概述：用户将进程包上传到OBS云服务，通过IoTEdge将该进程包配置为用户自定义应用，按需部署到指定边缘节点上运行起来。

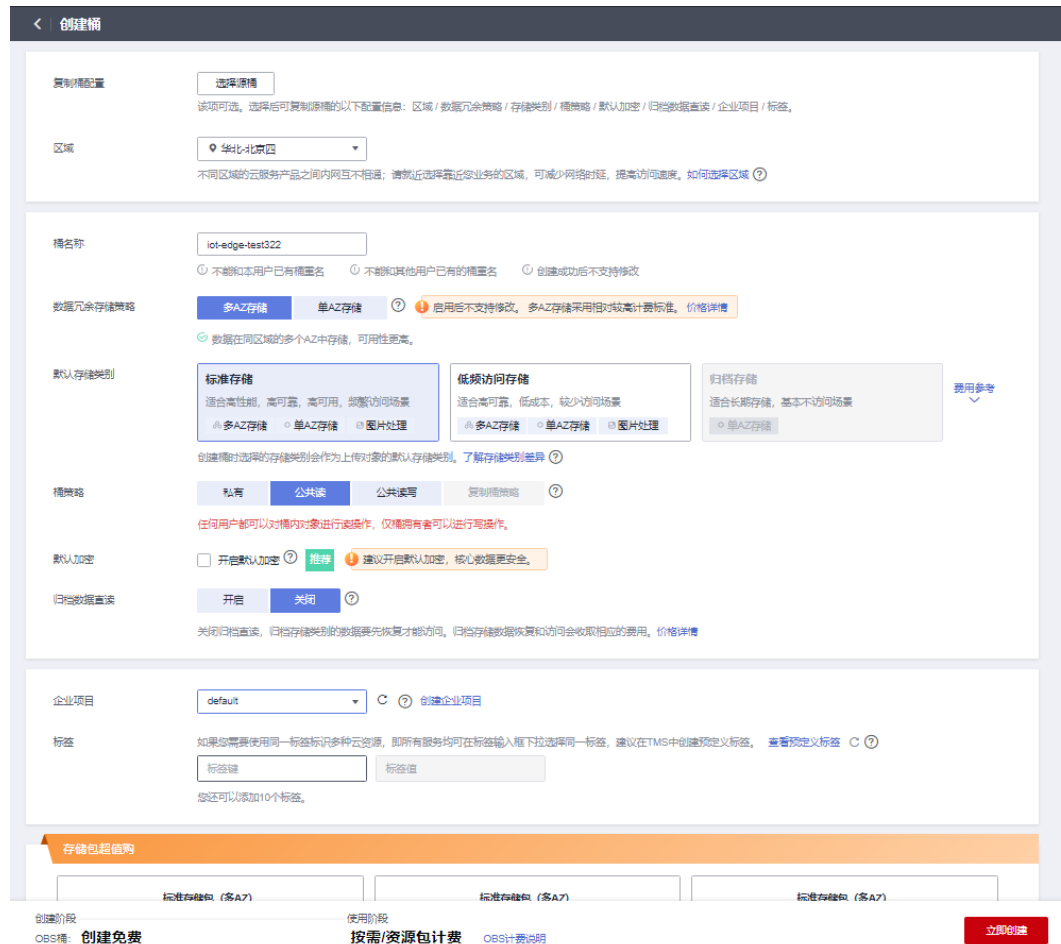
2.2.2 上传进程包到 OBS

步骤1 进程包插件由用户/伙伴开发，需集成边缘SDK，根据需要可分别基于ItClient/OAClient/DriverClient进行开发。此处仅演示部署步骤，故假设已有一个进程包。

ProcessWithStartBash.tar.gz

步骤2 访问**对象存储服务 OBS**，单击“进入控制台”，进入对象存储服务控制台。

步骤3 在左侧导航栏选择“桶列表”单击右上角“创建桶”，创建用于存储对象的桶。



- 区域：选择“华北-北京四”
- 数据冗余存储策略：“多AZ存储”的可靠性较高，出于实验成本，此处可选择“单AZ存储”
- 桶名称：自定义，如 iot-edge-test
- 存储类别：标准存储
- 桶策略：请务必选择“公开读”
- 默认加密：关闭
- 归档数据直读：关闭
- 企业项目：default

步骤4 在对象存储列表中选择刚创建的桶iot-edge-test进入详情界面，选择左侧菜单栏“对象”，单击“上传对象”，将进程包上传到OBS。



步骤5 上传完毕后，可在页面上查看到桶下的对象列表。



----结束

2.3 操作步骤

2.3.1 创建边缘节点

注册专业版节点

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 选择左侧导航栏“边缘节点 > 节点管理”单击右上角“注册节点”。
- 步骤3** 根据页面提示填写参数，勾选右下角“《华为云服务等级协议》”单击“下一步”配置节点数据。

图 2-1 专业版

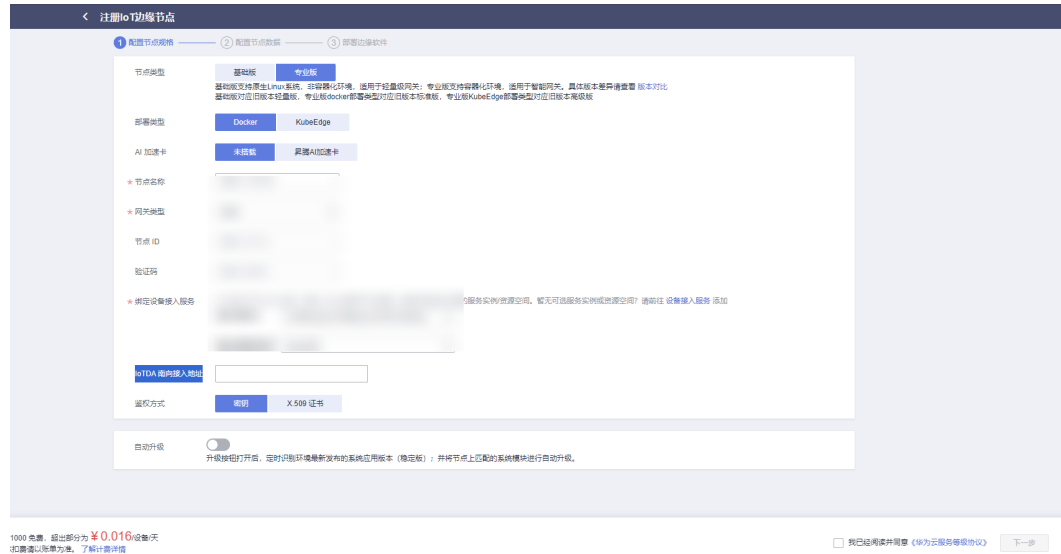


表 2-1 节点规格参数说明

参数名称	参数说明
节点类型	专业版。
部署类型	Docker。
AI加速卡	<ul style="list-style-type: none"> 未搭载：边缘节点未使用AI加速卡时选择。 昇腾AI加速卡：支持华为昇腾AI加速设备，边缘节点的服务器搭载相关昇腾AI加速设备时选择。
昇腾驱动地址	可选，AI加速卡选择昇腾AI加速卡时需要保证安装边缘节点的服务器上已经安装昇腾设备驱动，并根据前端提示填入驱动地址。若改值为空，默认填入Atlas300驱动默认安装地址： /usr/local/Ascend/driver/lib64/。
节点名称	自定义节点名称，如“test”。 说明 节点名称可以在：“节点概览” - “基本信息”中编辑修改。
网关类型	other。
节点ID	自定义（可选填写）。只允许中文，英文字母、数字、中划线（-），下划线（_），最小长度1，最大长度64。 说明 边缘节点ID自定义(不输入默认生成)。
验证码	输入自定义验证码（可选填写）。验证码，执行安装边缘软件命令时与云端通讯的凭证。
设备归属实例	选择默认归属实例。
设备归属资源空间	选择默认资源空间，或为本实验单独新建一个资源空间，如edge。

参数名称	参数说明
IoTDA南向接入地址	选择网关归属的IoTDA实例的南向设备mqtt接入地址，也可以选择自定义的南向接入地址，支持“ip:端口”或“域名:端口”形式。
鉴权方式	密钥：默认生成。 X.509证书：请参考： 基于MQTT.fx的X.509证书接入指导 。 警告 鉴权方式为“密钥”的节点，请勿在IoTDA界面重置节点的密钥，重置密钥会导致网关与IoTDA通信异常，且不可恢复。
指纹	证书文件生成的字符串（可选填写）。只能为40位十六进制字符串或64位六进制字符串。
本地路径	输入证书文件和密钥文件在设备上的存储路径，只允许字母、数字、下划线（_）和斜线（/），以/开头并且以.pem或.crt结尾。
自动升级	可选，开启后节点将自动升级系统应用模块。 注意 升级过程中可能导致节点数据上传中断。

步骤4 填写配置节点数据信息后，单击“立即创建”。

图 2-2 配置节点数据

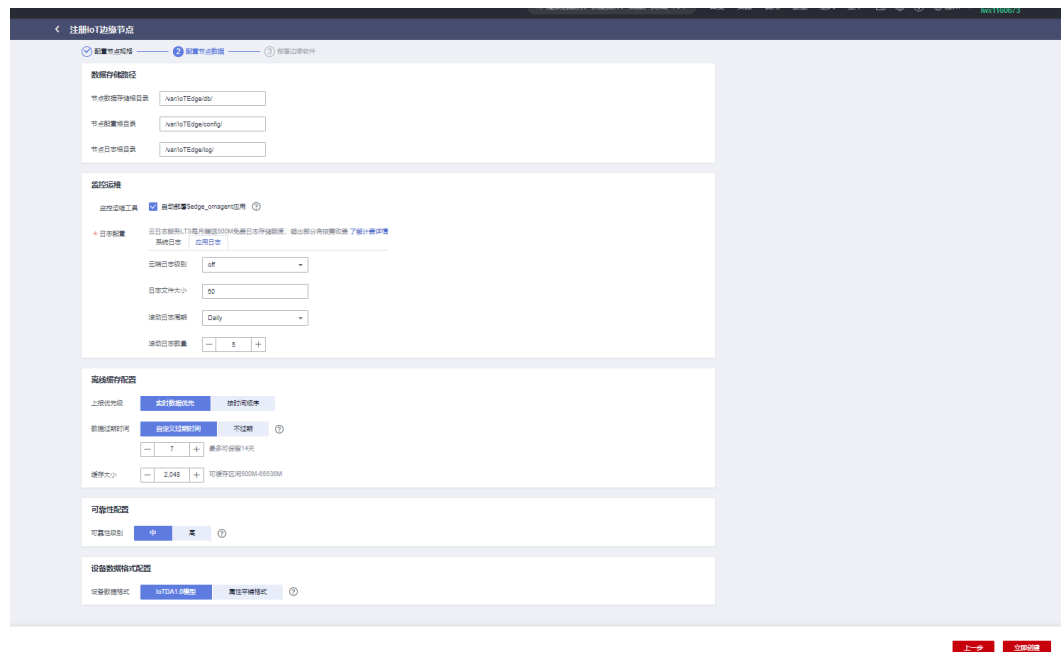


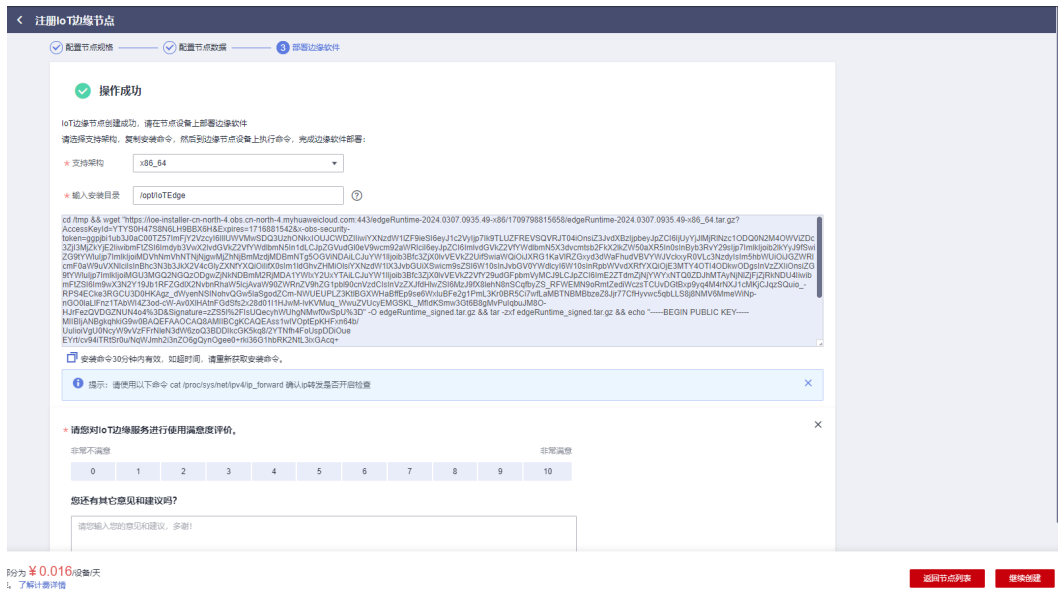
表 2-2 专业版节点信息

参数名称	说明
数据存储路径	默认（允许修改存储路径）。
监控运维	默认勾选“监控运维工具”。
\$edge_omagent应用	负责边缘节点远程监控和运维，提供日志上报、远程SSH、CPU等指标上报、文件上传下载能力。
日志配置	<p>系统日志：边缘节点上已部署的系统应用产生的日志。</p> <p>应用日志：边缘节点上已部署的用户定义应用所产生的日志。</p> <ul style="list-style-type: none">• 云端日志级别：on, off, trace, debug, info, warn, error, fatal。其中，on表示上传所有日志至云端，off表示不上传日志至云端，其余可选项代表按日志级别采集，选择后将只采集目标级别及以上的日志（模糊匹配）。日志级别的严重程度等级由低到高依次为：trace, debug, info, warn, error, fatal。• 日志文件大小：日志文件大小限制，单位MB，默认50，取值范围10-1000。某个日志文件如果达到大小限制，则会将文件转储保存至日志目录下。• 滚动日志周期：日志转储周期，可选项：daily, weekly, monthly, yearly。某个日志文件如果达到转储周期，则会将文件转储保存至日志目录下。日志文件大小和滚动日志周期是同时生效的，满足任何一个条件都会进行日志转储。• 滚动日志数量：日志文件转储个数，默认5，取值范围1-10。边缘节点保存的转储日志数量如果达到限制，则会删除最早的转储文件。 <p>注意</p> <ol style="list-style-type: none">1. 当前“云端日志级别”对日志等级采取模糊匹配的策略，若日志消息中存在与日志级别标识相同的关键字，则也可能被采集。2. 若要开启“云端日志级别”中的trace, debug, info, warn, error, fatal选项，则必须满足以下三个前置条件：<ol style="list-style-type: none">1. 目标节点是基础版节点或部署类型为Docker的专业版节点。需注意，部署类型为KubeEdge的专业版节点不支持此功能。2. 已安装\$edge_omagent应用，且版本不低于1-1-15-xxx。3. 应用日志需遵循如下规范打印，以便进行过滤：<ul style="list-style-type: none">○ 日志级别标识需打印为（大小写均可）：trace、debug、info、warn、error、fatal。○ 每条日志消息以YYYY-MM-DD日期格式作为前缀。 <p>说明</p> <p>专业版（部署类型为Docker）和基础版节点只涉及日志级别这一个参数，其他参数只对专业版（部署类型为KubeEdge）有效。</p>
离线缓存配置	可对离线节点上报的数据进行缓存，定义缓存周期（可配置为不过期）跟缓存大小、上报的优先级、路由转发是否开启离线缓存。仅当上报优先级为“按时间顺序”时支持路由转发开启离线缓存。

参数名称	说明
可靠性配置	可靠性级别分为“中”和“高”2种。中等级别：当应用上报属性和点位的频率超过流控值时，超过部分会被丢弃；高级别：当应用上报属性和点位的频率超过流控值时，通过控制模块的发送速度，降低上报频率，报文不会被丢弃。（注意：高可靠性级别的配置需要定制应用的配合，同时会造成性能的损失）。
设备数据格式配置	设备的数据格式：即单网关节点中设备的标准数据格式，包含 IoTDA1.0物模型格式和属性平铺格式。IoTDA1.0物模型中包含了设备、服务和属性三层结构，具体参考IoTDA网关子设备属性上报报文格式。属性平铺格式，包含设备、属性2层结构，属性结构可以包含自定义结构的拼接，在json中作为一项平铺展开，具体格式请参考 步骤6设备数据格式配置 。 须知 属性平铺格式功能，需要edge_hub应用1-1-38-xx版本及以上支持

步骤5 弹出如下对话框，您可以“返回节点列表”也可以“继续创建”。

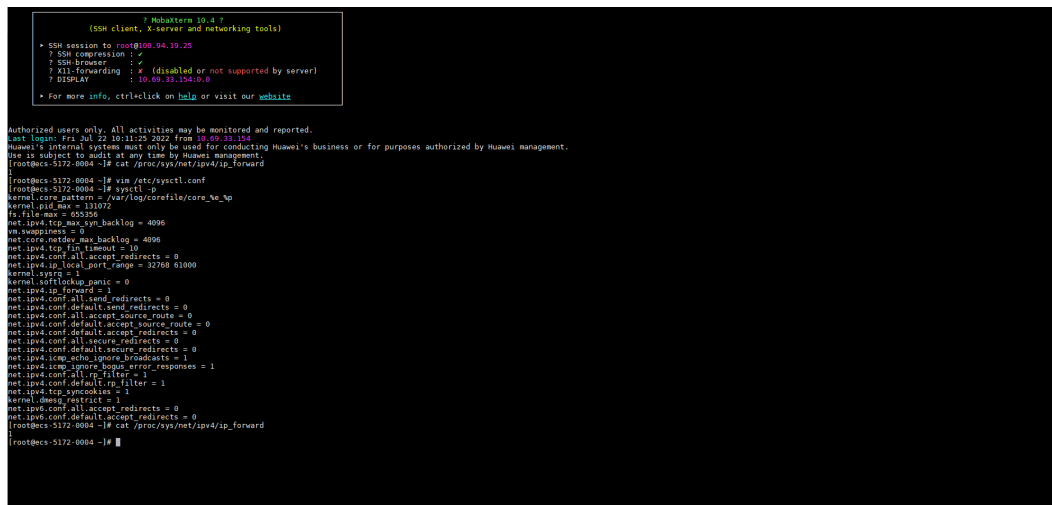
图 2-3 注册成功



说明

安装部署节点前，请确认机器的IP转发已开启并且已开启自启动，以及7883和8943端口开放。其中，IP转发规则配置如下：

图 2-4 IP 转发规则配置



1. 检查机器ip_forward in/proc文件系统的值。
`cat /proc/sys/net/ipv4/ip_forward`
 若值等于0，则需要添加配置IP转发规则。
 2. 检查配置文件/etc/sysctl.conf是否包含net.ipv4.ip_forward = 1。
 如果配置文件中没有net.ipv4.ip_forward参数项，请新增一行net.ipv4.ip_forward = 1。
`vim /etc/sysctl.conf`
 按“i”进入编辑模式，修改完成后按“Esc”退出编辑模式，然后按“Shift” + “;”组合键，输入wq并回车，保存修改并退出。
 3. 重新加载此文件的值。
`sysctl -p`
 4. 检查7883和8943端口是否开放，如果是华为云服务器，请参考：[安全组](#)。
- 结束

安装专业版节点

步骤1 在“节点管理”的节点列表选择您需要安装的节点，单击右侧“安装”，获得提示。

图 2-5 安装

节点名称	节点类型	资源包类型	资源包ID	状态	主机名/网络	操作
	专业版	--	--	未安装		删除 安装
	专业版	--	--	离线	172.30.7.64,172.17.0.1,1...	删除
	专业版	--	--	已安装		删除

步骤2 选择支持架构和安装目录，然后单击 复制安装命令，使用SSH工具以root用户登录边缘节点服务器后台系统，执行安装命令。

图 2-8 安装命令执行成功

```

2024-05-28T14:44:29,581321 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:PENDING, reason:(zero), ]
2024-05-28T14:44:31,632105 | INFO | saveBasePathInfo | 78 | config base path start
2024-05-28T14:44:31,632291 | INFO | saveBasePathInfo | 83 | config log path:/var/IoTEdge/log/, db_path:/var/IoTEdge/db/, config_path:/var/IoTEdge/config/
2024-05-28T14:44:31,632457 | INFO | saveBasePathInfo | 95 | config base path success.
2024-05-28T14:44:31,632535 | INFO | addAgentCustomConfigs | 201 | addCustomConfigs start.
2024-05-28T14:44:31,632603 | INFO | addAgentCustomConfigs | 205 | addCustomConfigs end.
2024-05-28T14:44:31,632654 | INFO | pullImage | 319 | start pull image, deploy type is cloud
2024-05-28T14:44:35,888519 | INFO | createModule | 76 | create Module, name:sys_edge_agent
2024-05-28T14:44:35,939013 | INFO | createModule | 94 | module create, CreateResp[ moduleId:sys_edge_agent, moduleName:sys-edge-agent, appId:sedge_agent, appVersion:test-tpm-2-x86, containerId:4b3e43cee7eac9bc282b994377abfb48a284389a36856f3cacc0338743036e, processId:(zero), type:docker, ]
2024-05-28T14:44:35,939163 | INFO | startModule | 160 | startModule moduleId:sys_edge_agent
2024-05-28T14:44:42,390994 | INFO | listModule | 259 | listModule
2024-05-28T14:44:42,440775 | INFO | checkModulesDo | 340 | start check modules, time:1
2024-05-28T14:44:42,440908 | INFO | listModule | 259 | listModule
2024-05-28T14:44:42,491163 | INFO | printModulesAndReport | 404 | ***success:1 failed:0
2024-05-28T14:44:42,491339 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:RUNNING, reason:(zero), ]
2024-05-28T14:44:42,542192 | INFO | reportEvent | 158 | Begin reporting event: {"events":[{"resource":"node.installation","location":"zzzzzz","event":"succeeded","event_severity":"Info","event_time":"20240528T064442Z","notify_data":{"host_tag":"DEFAULT","edge_node_id":"zzzzzz"}}]}
2024-05-28T14:44:42,592933 | INFO | reportModuleStatus | 138 | reportModuleStatus, modulesInfo:ModuleStatusInfo[stl host_tag:DEFAULT, deploy_id:1716878669308, modules:[ModuleStatusInfo[ module_id:sys_edge_agent, app_id:sedge_agent, app_version:test-tpm-2-x86, container_state:RUNNING, liveness_state:(zero), update_time:1716878669305, ]], ]
2024-05-28T14:44:43,043689 | INFO | checkBridgeNetwork | 560 | begin checking bridge network
2024-05-28T14:44:50,524377 | INFO | install | 494 | install edge finish
2024-05-28T14:44:50,524559 | INFO | stop | 545 | edgeinstatter stop.
/root
    
```

步骤3 单击“我知道了”，等待边缘节点的状态变为“在线”，表示该节点已安装并连接成功。

步骤4 您可以单击“节点名称”，查看该节点详细信息，关于节点详情描述可参见[节点概览](#)。

----结束

2.3.2 (可选) 部署边缘应用

在边缘节点安装成功后可部署边缘应用，本示例演示Modbus设备接入。

步骤1 访问[IoT边缘](#)，单击“管理控制台”进入IoT边缘控制台。

步骤2 选择左侧导航栏“边缘节点 > 节点管理”进入页面。

步骤3 选择您的边缘节点，单击“节点名称”进入节点详情页。

步骤4 选择左侧导航栏“应用模块 > 模块管理”页签，查看已部署边缘应用。

专业版（Docker）默认部署sys_edge_hub和sys_edge_agent，专业版（KubeEdge）默认部署sys_edge_hub。

实例状态显示为“运行中”，表示应用运行正常。

模块名称	模块图标	所属应用	版本	应用类型	部署状态	操作
sys_edge_agent		sedge_agent	1-17-standard-008	系统应用	运行中	详情
sys_edge_hub		sedge_hub	1-17-standard-008	系统应用	运行中	详情

----结束

2.3.3 创建自定义应用

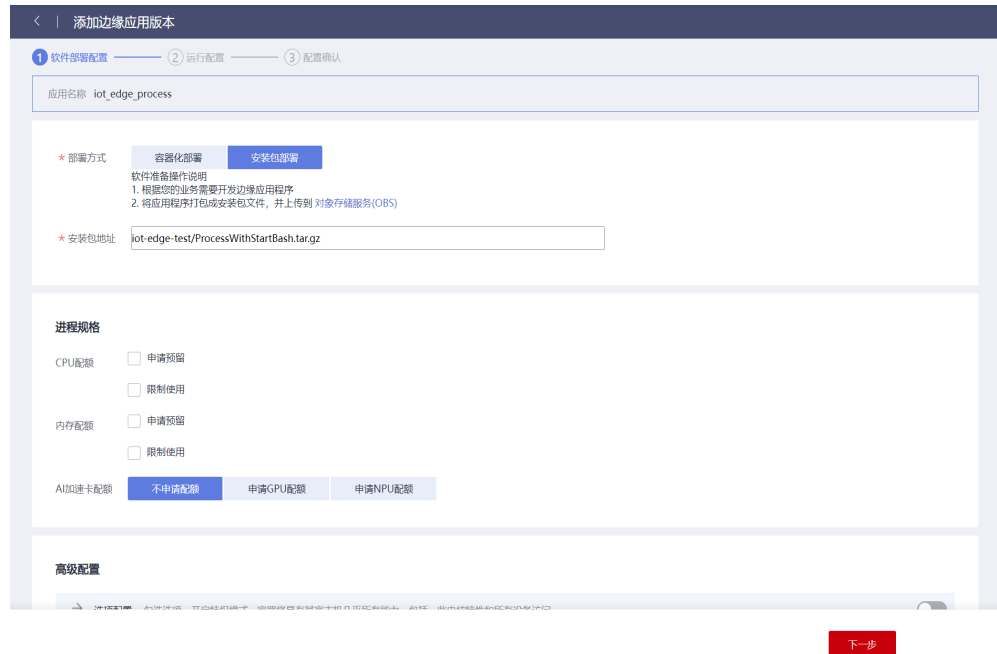
在IoT边缘控制台上添加应用和应用版本，即可将OBS云服务中的进程包插件通过IoT边缘部署到指定边缘节点中。

步骤1 访问[IoT边缘](#)，单击“管理控制台”进入IoT边缘控制台。

步骤2 选择左侧导航栏“边缘节点 > 应用管理”进入页面。

步骤3 单击右上角的“添加应用”，按照页面提示填写信息。

1. 软件部署配置



- 部署方式：选择“安装包部署”
- 安装包地址：格式为“OBS桶名称/进程包对象名称”请参见[上传进程包到OBS](#)步骤，将进程包上传到OBS后，获取“安装包地址”此处配置“iot-edge-test/ProcessWithStartBash.tar.gz”，其他配置保持默认即可。
- cpu配额，cpu资源占用率限制
- 内存配额，内存资源占用率限制

📖 说明

申请预留，可理解为最低占用。限制使用，可理解为最高占用

AI加速卡：NPU、GPU等加速卡资源占用率限制。需要正确安装驱动及镜像支持调用。

高级配置

运行命令：启动容器时执行的命令

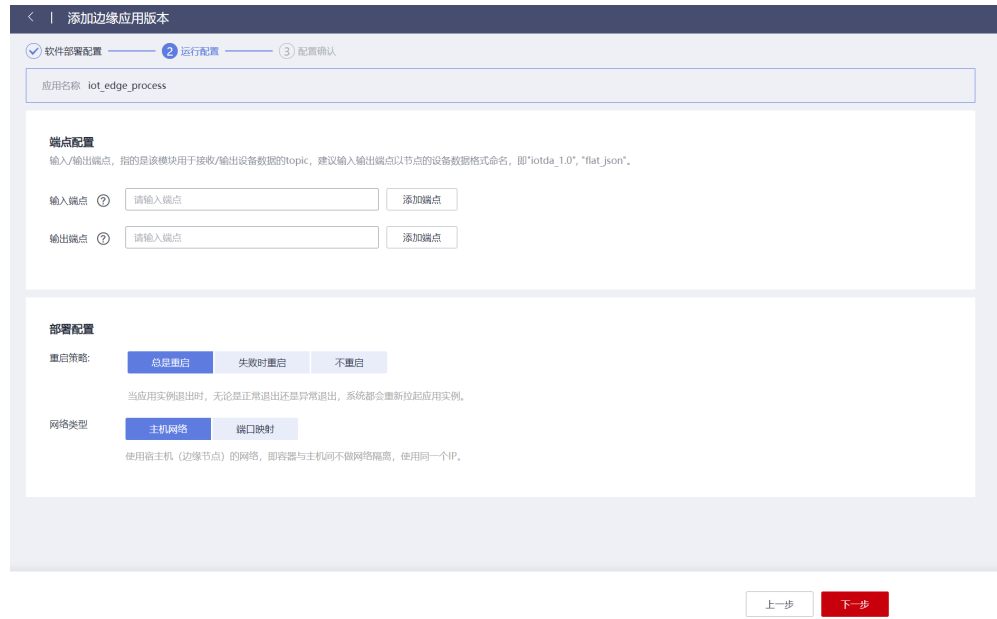
选项配置：容器特权选项，开启后容器可以访问GPU等组件。

数据存储：将宿主机文件目录挂载到容器中，可以实现配置保存，日志保存等持久化。

外挂设备：将串口、摄像头等设备添加到容器。

健康检查：提供接口检查容器业务是否正常，边缘软件检测到不正常会自动重启容器。

2. 运行配置

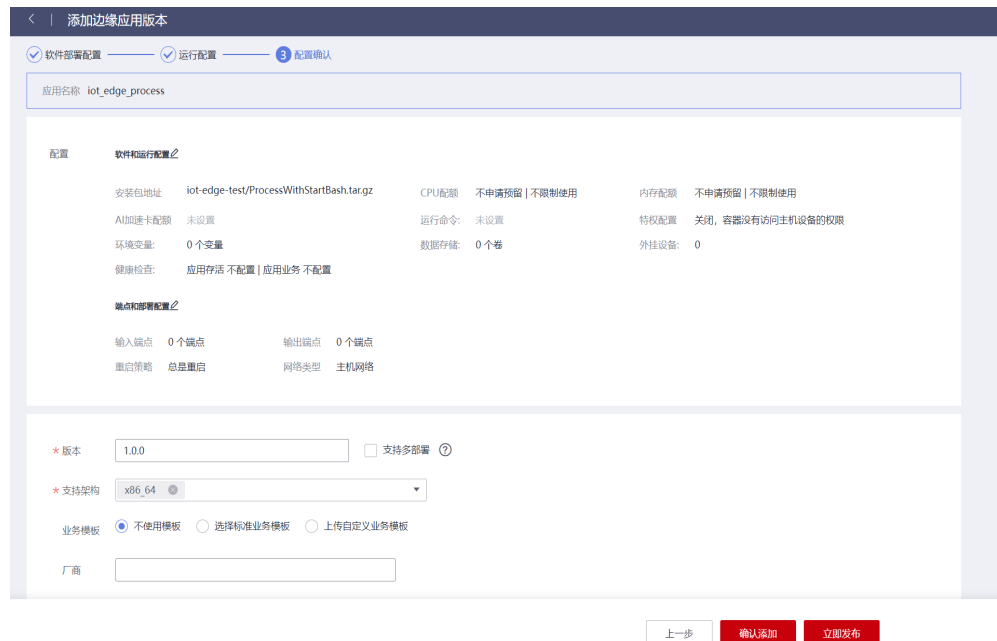


输入/出端点：若进程包插件集成了边缘SDK，并配置了输入/输出端点，则该容器进程包可与hub建立MQTT连接，并可以从该输入端点获得设备数据，进行自定义的数据处理后，从输出端点将数据流回hub。

在“添加应用”时，也需配置输入/输出端点，为后续配置边端流转规则提供前提条件。

本实验使用的进程包没有集成边缘SDK，故不配置输入/输出端点。

3. 配置确认



- 功能用途：根据容器应用的功能用途自行选择，本实验选择“数据处理”。
数据处理：将边缘接收到的数据处理后（去重，聚合等）再上报云端
- 协议解析：通过集成sdk接入第三方协议
- 本地子系统集成：将erp等系统数据接入云端

- 网关管理：开发中
 - 混合应用：含多个功能（如含协议解析和数据处理两个功能）的应用
 - 数据采集：将数据库数据接入云端
 - 应用描述：自定义，可不填。
 - 版本：自定义，如“1.0.0”
 - 支持架构：根据容器应用镜像支持的架构自行选择，本实验采用的镜像只支持x86架构，故选择“x86_64”。
4. 单击“确认添加”后，则创建自定义应用成功(单击立即发布可直接部署)。
 5. 返回“应用管理”可查看到创建的应用，单击“应用名称”进入详情页，此时在版本列表中选择发布指定版本即可。



📖 说明

只有状态为“已发布”的应用版本才可以部署到边缘节点中。

----结束

2.3.4 部署进程包插件

在IoT边缘控制台上将已创建的自定义应用部署到指定边缘节点中，即进程包插件将在边缘节点上以进程身份运行。

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 选择左侧导航栏的“边缘节点>节点管理”，选择您的边缘节点名称，进入节点详情页。
- 步骤3** 选择左侧导航栏“应用模块 > 模块管理”页签，单击“部署应用”，将制作的自定义应用部署到边缘节点上运行。



边缘应用：选择上述制作的应用名称，如：iot_edge_process

选择版本：选择已发布的与边缘节点相同架构的应用版本，如：1.0.0

模块名称：自定义，如：iot_edge_process

步骤4 等待5秒后，刷新应用界面，可看到部署的自定义应用的实例状态为“运行中”，表明该进程包插件已运行起来。



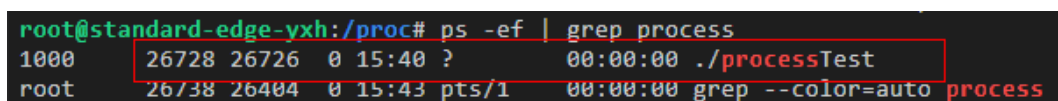
说明

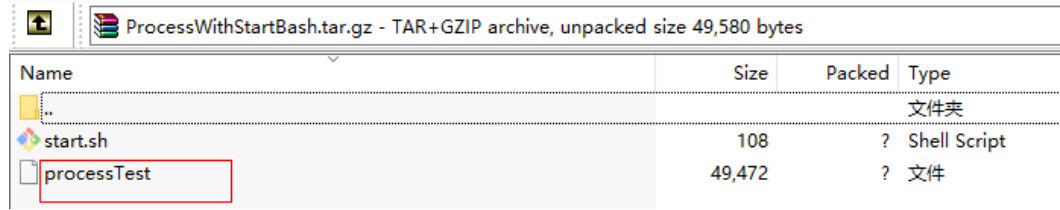
若部署应用后实例状态为“实例异常”，请检查上传进程包到OBS桶时，是否已将桶策略设置为“公开读”。

步骤5 登录边缘节点服务器后台系统，执行命令。

```
//process为进程包中的脚本名称，根据实际情况进行调整  
$ ps -ef | grep process
```

可查看边缘节点上运行的进程。





----结束

2.3.5 资源释放

实验完成，您可以按如下步骤删除在实验过程中，产生在您的华为云账号下的相关资源。

注：所有资源删除动作，在“华北-北京四”区域下进行。

删除弹性云服务器 ECS

步骤1 访问[弹性云服务器 ECS](#)，单击“**管理控制台**”，进入云服务器控制台。

步骤2 在页面左上角切换区域至“华北-北京四”。

步骤3 在左侧导航中选择“弹性云服务器”，按需勾选待删除ECS服务器，如ecs-iot，选择“更多 > 删除”。

请勾选“释放云服务器绑定的弹性公网IP地址”和“删除云服务器挂载的数据盘”。



----结束

删除自定义应用

步骤1 访问[IoT边缘](#)，单击“**管理控制台**”进入IoT边缘控制台。

步骤2 选择左侧导航栏“边缘节点 > 应用管理”进入页面。

步骤3 根据应用名称选择需要删除的自定义应用，如iot_edge_process，单击操作列的“删除”即可。

📖 说明

下线状态的版本且未部署到节点上的应用版本才允许删除。

----结束

删除边缘节点

步骤1 访问[IoT边缘](#)，单击“**管理控制台**”进入IoT边缘控制台。

步骤2 选择左侧导航栏“边缘节点 > 节点管理”进入页面。

步骤3 根据节点名称选择，单击操作列的“删除”，并在弹出的窗口中单击“确认”。

删除节点需要一个过程，不超出1分钟，请耐心等待。



----结束

删除虚拟私有云 VPC

步骤1 访问[虚拟私有云 VPC](#)，单击“立即使用”进入网络控制台。

步骤2 选择左侧导航栏“子网”，删除子网，如subnet-iot，请根据节点名称选择，单击操作列的“删除”，并在弹出的窗口中单击“是”。

步骤3 选择左侧导航栏“虚拟私有云”，删除虚拟私有云，如vpc-iot，请根据节点名称选择，单击操作列的“删除”，并在弹出的窗口中单击“确定”。

----结束

删除 OBS 对象

步骤1 访问[对象存储服务 OBS](#)，单击“立即购买”，进入对象存储服务控制台。

步骤2 左侧导航栏单击“桶列表”选择桶名称，如：iot-edge-test进入桶详情界面。

步骤3 在左侧菜单栏选择“对象”，选中待删除的对象，如：ProcessWithStartBash.tar.gz，单击操作列的“更多 > 删除”按钮。

步骤4 返回对象存储控制台，删除桶iot-edge-test。

 **说明**

桶里面的对象个数为0才允许删除桶。

----**结束**

3 基于 IoT 边缘实现容器应用镜像插件部署

3.1 实验简介

本实验将指导您搭建一个基于IoT边缘服务的边缘节点，并在边缘节点上进行容器应用镜像插件部署，将用户自定义的容器镜像部署到边缘节点上，并实现节点上应用之间的数据流转。

通过本实验，您将能够：

- 了解并掌握IoT边缘服务的功能特点与使用方法。
- 掌握通过IoTEdge云服务将自定义容器镜像部署到边缘节点的方法。

3.2 前提条件

3.2.1 前提概要

1. **前提：**体验需要三方配合，首先要开通IoTEdge，用于节点管理和应用管理，再开通IoTDA，用于设备接入与管理，还得有边缘服务器作为节点被IoTEdge纳管。此外还需开通SWR，用于存储已开发并想部署在边缘节点上运行的Docker镜像。
2. **边缘服务器：**第一点提到的IoTEdge, IoTDA, SWR是华为云服务，使用华为云租户账号直接免费开通可用，边缘服务器属于硬件，可以购买低配的ECS代替，其EIP则作为边缘服务器的节点IP地址。
3. **在IoTEdge上注册节点：**在购买的ECS上执行安装脚本命令，即自动部署好2个系统应用，agent用于节点应用管理，hub用于数据通信。
4. **Docker镜像：**是集成了边缘SDK的应用程序打包成的容器镜像，可作为边缘节点的自定义应用部署到边缘服务器上运行，可在节点上进行设备数据协议解析、子系统数据解析、数据业务处理、智能计算等。

概述：用户将Docker镜像打包上传到SWR云服务，通过IoTEdge将该容器镜像配置为用户自定义应用，按需部署到指定边缘节点上运行起来。

3.2.2 上传容器应用镜像到 SWR

步骤1 应用镜像需集成边缘SDK，根据需要可分别基于ItClient/OAClient/DriverClient进行开发。此处仅演示部署步骤，故可直接拉取一个现有的可公开拉取到的镜像。

步骤2 登录边缘节点硬件（例如步骤2.1购买的ECS），执行拉取镜像的命令：

```
//1.拉取镜像
$ docker pull swr.cn-north-4.myhuaweicloud.com/huawei-iot-edge/edgerule:1.1.0001.20200428202651.x86
//2.查看拉取到的镜像
$ docker images
```

步骤3 访问[容器镜像服务 SWR](#)，单击“立即使用”，进入容器镜像服务控制台。

步骤4 单击右上角“创建组织”，创建容器镜像所属的组织。

组织名称：自定义，如iot_edge_test

步骤5 单击右上角“登录指令”，复制在边缘节点硬件上连接容器镜像服务的临时指令。



步骤6 登录边缘节点服务器后台系统，输入复制的登录指令并回车，提示“Login Succeeded”则表明该硬件已通过SWR云服务的鉴权认证，可进行上传/下载镜像。

步骤7 上传docker镜像到SWR云服务

1. 为本地镜像打标签，格式为docker tag [镜像名称:版本名称] [镜像仓库地址]/[组织名称]/[镜像名称:版本名称]

```
$ docker tag swr.cn-north-4.myhuaweicloud.com/huawei-iot-edge/edgerule:1.1.0001.20200428202651.x86 swr.cn-north-4.myhuaweicloud.com/iot_edge_test/app:1.0.0
```

2. 上传镜像到SWR，格式为docker push [镜像仓库地址]/[组织名称]/[镜像名称:版本名称]

```
$ docker push swr.cn-north-4.myhuaweicloud.com/iot_edge_test/app:1.0.0
```

- 镜像名称/版本名称：swr.cn-north-4.myhuaweicloud.com/huawei-iot-edge/edgerule:1.1.0001.20200428202651.x86为用户自定义的容器应用镜像，本实验中使用一个公开的数据清洗镜像，即**步骤1**中拉取到ECS的镜像。
- 镜像仓库地址：本实验在北京四环境，则该地址为swr.cn-north-4.myhuaweicloud.com
- 组织名称：上述**步骤4**在SWR云服务上创建的组织名称，即iot_edge_test
- 镜像名称/版本：自定义，该名称即为上传到SWR存储时镜像的名称/版本

步骤8 在SWR控制台选择“我的镜像”，可查看上传的自定义镜像app。

----结束

3.3 操作步骤

3.3.1 创建边缘节点

注册专业版节点

步骤1 访问[IoT边缘](#)，单击“管理控制台”进入IoT边缘控制台。

步骤2 选择左侧导航栏“边缘节点 > 节点管理”单击右上角“注册节点”。

步骤3 根据页面提示填写参数，勾选右下角“《华为云服务等级协议》”单击“下一步”配置节点数据。

图 3-1 专业版

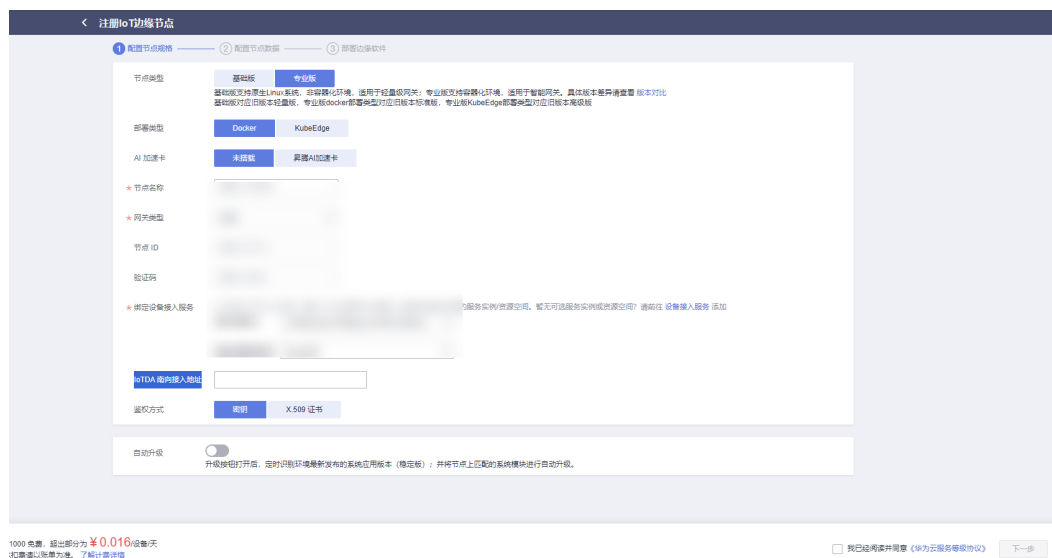


表 3-1 节点规格参数说明

参数名称	参数说明
节点类型	专业版。
部署类型	Docker。
AI加速卡	<ul style="list-style-type: none"> 未搭载：边缘节点未使用AI加速卡时选择。 昇腾AI加速卡：支持华为昇腾AI加速设备，边缘节点的服务器搭载相关昇腾AI加速设备时选择。
昇腾驱动地址	可选，AI加速卡选择昇腾AI加速卡时需要保证安装边缘节点的服务器上已经安装昇腾设备驱动，并根据前端提示填入驱动地址。若改值为空，默认填入Atlas300驱动默认安装地址： /usr/local/Ascend/driver/lib64/。
节点名称	自定义节点名称，如“test”。 说明 节点名称可以在：“节点概览” - “基本信息”中编辑修改。
网关类型	other。
节点ID	自定义（可选填写）。只允许中文，英文字母、数字、中划线（-），下划线（_），最小长度1，最大长度64。 说明 边缘节点ID自定义(不输入默认生成)。
验证码	输入自定义验证码（可选填写）。验证码，执行安装边缘软件命令时与云端通讯的凭证。

参数名称	参数说明
设备归属实例	选择默认归属实例。
设备归属资源空间	选择默认资源空间，或为本实验单独新建一个资源空间，如 edge。
IoTDA南向接入地址	选择网关归属的IoTDA实例的南向设备mqtt接入地址，也可以选择自定义的南向接入地址，支持“ip:端口”或“域名:端口”形式。
鉴权方式	密钥：默认生成。 X.509证书：请参考： 基于MQTT.fx的X.509证书接入指导 。 警告 鉴权方式为“密钥”的节点，请勿在IoTDA界面重置节点的密钥，重置密钥会导致网关与IoTDA通信异常，且不可恢复。
指纹	证书文件生成的字符串（可选填写）。只能为40位十六进制字符串或64位六进制字符串。
本地路径	输入证书文件和密钥文件在设备上的存储路径，只允许字母、数字、下划线（_）和斜线（/），以/开头并且以.pem或.crt结尾。
自动升级	可选，开启后节点将自动升级系统应用模块。 注意 升级过程中可能导致节点数据上传中断。

步骤4 填写配置节点数据信息后，单击“立即创建”。

图 3-2 配置节点数据

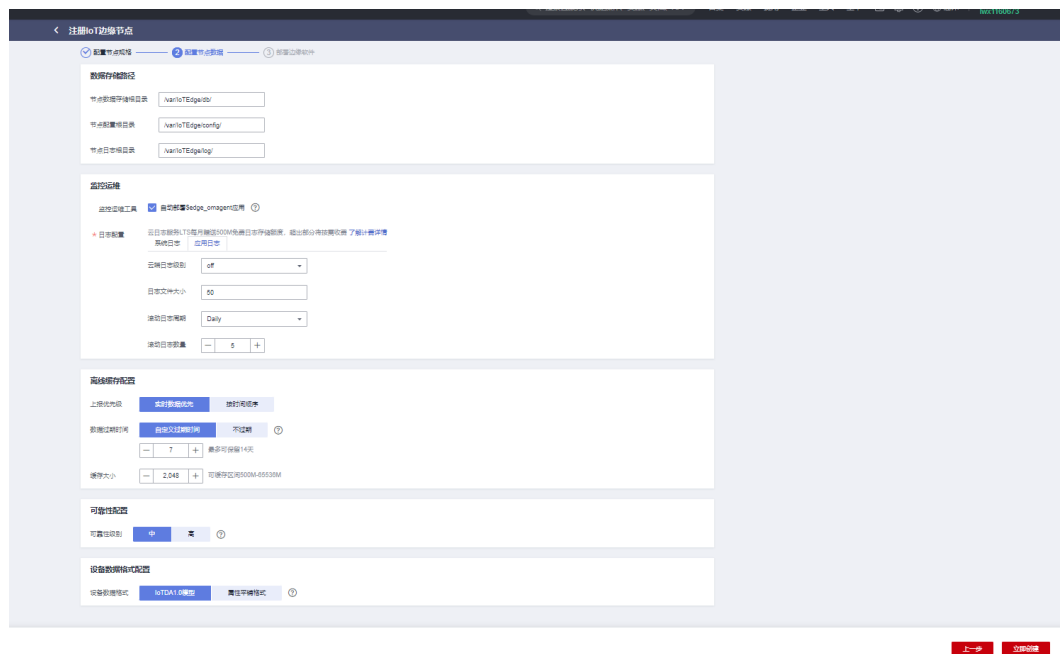


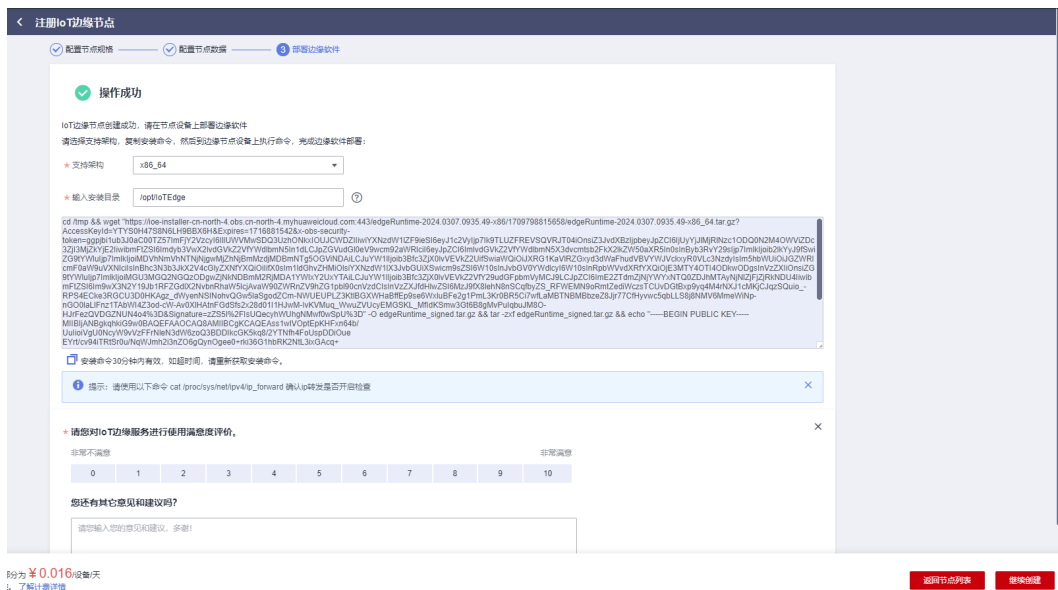
表 3-2 专业版节点信息

参数名称	说明
数据存储路径	默认（允许修改存储路径）。
监控运维	默认勾选“监控运维工具”。
\$edge_omagent应用	负责边缘节点远程监控和运维，提供日志上报、远程SSH、CPU等指标上报、文件上传下载能力。
日志配置	<p>系统日志：边缘节点上已部署的系统应用产生的日志。</p> <p>应用日志：边缘节点上已部署的用户定义应用所产生的日志。</p> <ul style="list-style-type: none"> • 云端日志级别：on, off, trace, debug, info, warn, error, fatal。其中，on表示上传所有日志至云端，off表示不上传日志至云端，其余可选项代表按日志级别采集，选择后将只采集目标级别及以上的日志（模糊匹配）。日志级别的严重程度等级由低到高依次为：trace, debug, info, warn, error, fatal。 • 日志文件大小：日志文件大小限制，单位MB，默认50，取值范围10-1000。某个日志文件如果达到大小限制，则会将文件转储保存至日志目录下。 • 滚动日志周期：日志转储周期，可选项：daily, weekly, monthly, yearly。某个日志文件如果达到转储周期，则会将文件转储保存至日志目录下。日志文件大小和滚动日志周期是同时生效的，满足任何一个条件都会进行日志转储。 • 滚动日志数量：日志文件转储个数，默认5，取值范围1-10。边缘节点保存的转储日志数量如果达到限制，则会删除最早的转储文件。 <p>注意</p> <ol style="list-style-type: none"> 1. 当前“云端日志级别”对日志等级采取模糊匹配的策略，若日志消息中存在与日志级别标识相同的关键字，则也可能被采集。 2. 若要开启“云端日志级别”中的trace, debug, info, warn, error, fatal选项，则必须满足以下三个前置条件： <ol style="list-style-type: none"> 1. 目标节点是基础版节点或部署类型为Docker的专业版节点。需注意，部署类型为KubeEdge的专业版节点不支持此功能。 2. 已安装\$edge_omagent应用，且版本不低于1-1-15-xxx。 3. 应用日志需遵循如下规范打印，以便进行过滤： <ul style="list-style-type: none"> ○ 日志级别标识需打印为（大小写均可）：trace、debug、info、warn、error、fatal。 ○ 每条日志消息以YYYY-MM-DD日期格式作为前缀。 <p>说明</p> <p>专业版（部署类型为Docker）和基础版节点只涉及日志级别这一个参数，其他参数只对专业版（部署类型为KubeEdge）有效。</p>
离线缓存配置	可对离线节点上报的数据进行缓存，定义缓存周期（可配置为不过期）跟缓存大小、上报的优先级、路由转发是否开启离线缓存。仅当上报优先级为“按时间顺序”时支持路由转发开启离线缓存。

参数名称	说明
可靠性配置	可靠性级别分为“中”和“高”2种。中等级别：当应用上报属性和点位的频率超过流控值时，超过部分会被丢弃；高级别：当应用上报属性和点位的频率超过流控值时，通过控制模块的发送速度，降低上报频率，报文不会被丢弃。（注意：高可靠性级别的配置需要定制应用的配合，同时会造成性能的损失）。
设备数据格式配置	设备的数据格式：即单网关节点中设备的标准数据格式，包含 IoTDA1.0物模型格式和属性平铺格式。IoTDA1.0物模型中包含了设备、服务和属性三层结构，具体参考IoTDA网关子设备属性上报报文格式。属性平铺格式，包含设备、属性2层结构，属性结构可以包含自定义结构的拼接，在json中作为一项平铺展开，具体格式请参考 步骤6设备数据格式配置 。 须知 属性平铺格式功能，需要edge_hub应用1-1-38-xx版本及以上支持

步骤5 弹出如下对话框，您可以“返回节点列表”也可以“继续创建”。

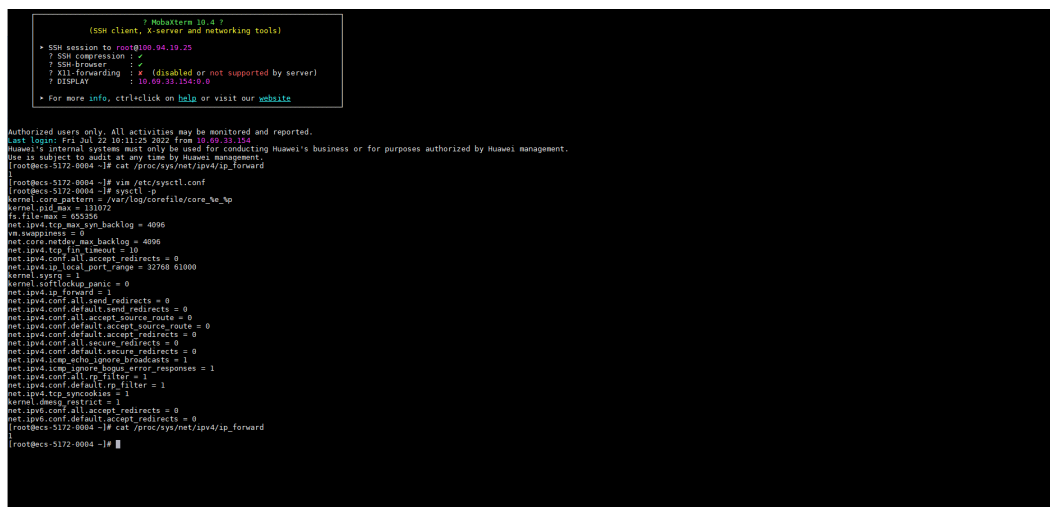
图 3-3 注册成功



说明

安装部署节点前，请确认机器的IP转发已开启并且已开启自启动，以及7883和8943端口开放。其中，IP转发规则配置如下：

图 3-4 IP 转发规则配置



1. 检查机器ip_forward in/proc文件系统的值。
`cat /proc/sys/net/ipv4/ip_forward`
 若值等于0，则需要添加配置IP转发规则。
 2. 检查配置文件/etc/sysctl.conf是否包含net.ipv4.ip_forward = 1。
 如果配置文件中没有net.ipv4.ip_forward参数项，请新增一行net.ipv4.ip_forward = 1。
`vim /etc/sysctl.conf`
 按“i”进入编辑模式，修改完成后按“Esc”退出编辑模式，然后按“Shift” + “;”组合键，输入wq并回车，保存修改并退出。
 3. 重新加载此文件的值。
`sysctl -p`
 4. 检查7883和8943端口是否开放，如果是华为云服务器，请参考：[安全组](#)。
- 结束

安装专业版节点

步骤1 在“节点管理”的节点列表选择您需要安装的节点，单击右侧“安装”，获得提示。

图 3-5 安装

节点名称	节点类型	资源包类型	资源包ID	状态	主机名/网络	操作
	专业版	--	--	未安装		删除 安装
	专业版	--	--	离线	172.30.7.64,172.17.0.1,1...	删除
	专业版	--	--	已安装		删除


步骤2 选择支持架构和安装目录，然后单击  复制安装命令，使用SSH工具以root用户登录边缘节点服务器后台系统，执行安装命令。

图 3-8 安装命令执行成功

```
2024-05-28T14:44:29,581321 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:PENGOING, reason:(zero) ]
2024-05-28T14:44:31,632185 | INFO | saveBasePathInfo | 78 | config base path start.
2024-05-28T14:44:31,632291 | INFO | saveBasePathInfo | 83 | config log path:/var/IoTEdge/log/, db_path:/var/IoTEdge/db/, config_path:/var/IoTEdge/config/
2024-05-28T14:44:31,632457 | INFO | saveBasePathInfo | 95 | config base path success.
2024-05-28T14:44:31,632535 | INFO | addAgentCustomConfigs | 261 | addCustomConfigs start.
2024-05-28T14:44:31,632603 | INFO | addAgentCustomConfigs | 285 | addCustomConfigs end.
2024-05-28T14:44:31,632654 | INFO | pullImage | 319 | start pull image, deploy type is cloud
2024-05-28T14:44:35,888319 | INFO | createModule | 76 | create Module, name:sys_edge_agent
2024-05-28T14:44:35,939013 | INFO | createModule | 94 | module created, CreateRespBody: ModuleCreateResp[ moduleId:sys_edge_agent, moduleName:sys-edge-agent, appId:sedge_agent, appVersion:test-tpm-2-x86, containerId:4b3e43ce7eac9bc282b994377abfb148a264389a36856f3cacc9338743036e, processId:(zero), type:docker, ]
2024-05-28T14:44:35,939163 | INFO | startModule | 160 | startModule moduleId:sys_edge_agent
2024-05-28T14:44:42,398934 | INFO | listModule | 250 | listModule
2024-05-28T14:44:42,440775 | INFO | checkModulesBo | 340 | start check modules, time:1
2024-05-28T14:44:42,440909 | INFO | listModule | 250 | listModule
2024-05-28T14:44:42,491163 | INFO | printModulesAndReport | 484 | ***success:1 failed:0
2024-05-28T14:44:42,491339 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:RUNNING, reason:(zero) ]
2024-05-28T14:44:42,542192 | INFO | reportEvent | 158 | Begin reporting event: {"events":[{"resource":"node.installation","location":"zzzzz","event":"succeeded","event_severity":"Info", "event_time":"20240528T044422"}, {"notify_data":{"host_tag":"DEFAULT","edge_node_id":"zzzzz"}}]}
2024-05-28T14:44:42,529233 | INFO | reportModuleStatus | 138 | reportModuleStatus, modulesInfo:ModuleStatusInfoList[ host_tag:DEFAULT, deploy_id:1716878669308, modules:[ModuleStatusInfo module_id:sys_edge_agent, app_id:sedge_agent, app_version:test-tpm-2-x86, container_state:RUNNING, liveness_state:(zero), update_time:1716878669305], ]
2024-05-28T14:44:43,643689 | INFO | checkBridgeNetwork | 556 | Begin checking bridge network
2024-05-28T14:44:50,524137 | INFO | install | 495 | install edge finish
2024-05-28T14:44:50,524559 | INFO | stop | 545 | edgeinstaller stop.
/root
```

步骤3 单击“我知道了”，等待边缘节点的状态变为“在线”，表示该节点已安装并连接成功。

步骤4 您可以单击“节点名称”，查看该节点详细信息，关于节点详情描述可参见[节点概览](#)。

----结束

3.3.2 (可选) 部署边缘应用

在边缘节点安装成功后可部署边缘应用，本示例演示Modbus设备接入，除了安装节点完毕后自动部署的2个系统应用（hub, agent）。

步骤1 访问[IoT边缘](#)，单击“管理控制台”进入IoT边缘控制台。

步骤2 选择左侧导航栏“边缘节点 > 节点管理”进入页面。

步骤3 选择您的边缘节点，单击“节点名称”进入节点详情页。

步骤4 选择“应用模块”页签，查看已部署边缘应用。

专业版（Docker）默认部署sys_edge_hub和sys_edge_agent，专业版（KubeEdge）默认部署sys_edge_hub。

实例状态显示为“运行中”，表示应用运行正常。

模块id	模块名称	所属应用	版本	应用类型	实例状态	操作
sys_edge_access	sys_edge_access	Sedge_access	1-0-13-x86	系统可选	运行中	删除 升级
sys_edge_agent	sys_edge_agent	Sedge_agent	1-1-7-8be-x86_64	系统必选	运行中	升级
sys_edge_hub	sys_edge_hub	Sedge_hub	1-1-12-8be-x86_64	系统必选	运行中	升级
sys_edge_omagent	sys_edge_omagent	Sedge_omagent	1-1-11-8be-x86_64	系统可选	运行中	删除 升级
sys_edge_rule	sys_edge_rule	Sedge_rule	1-1-0-standard-x86	系统可选	运行中	删除 升级
sys_ot_dc_opcua	sys_ot_dc_opcua	Sot_dc_opcua	1-0-2-standard-x86	系统可选	运行中	删除 升级
sys_ot_dc_yk	sys_ot_dc_yk	Sot_dc_yk	1-0-3-x86	系统可选	运行中	删除 升级

----结束

3.3.3 创建自定义应用

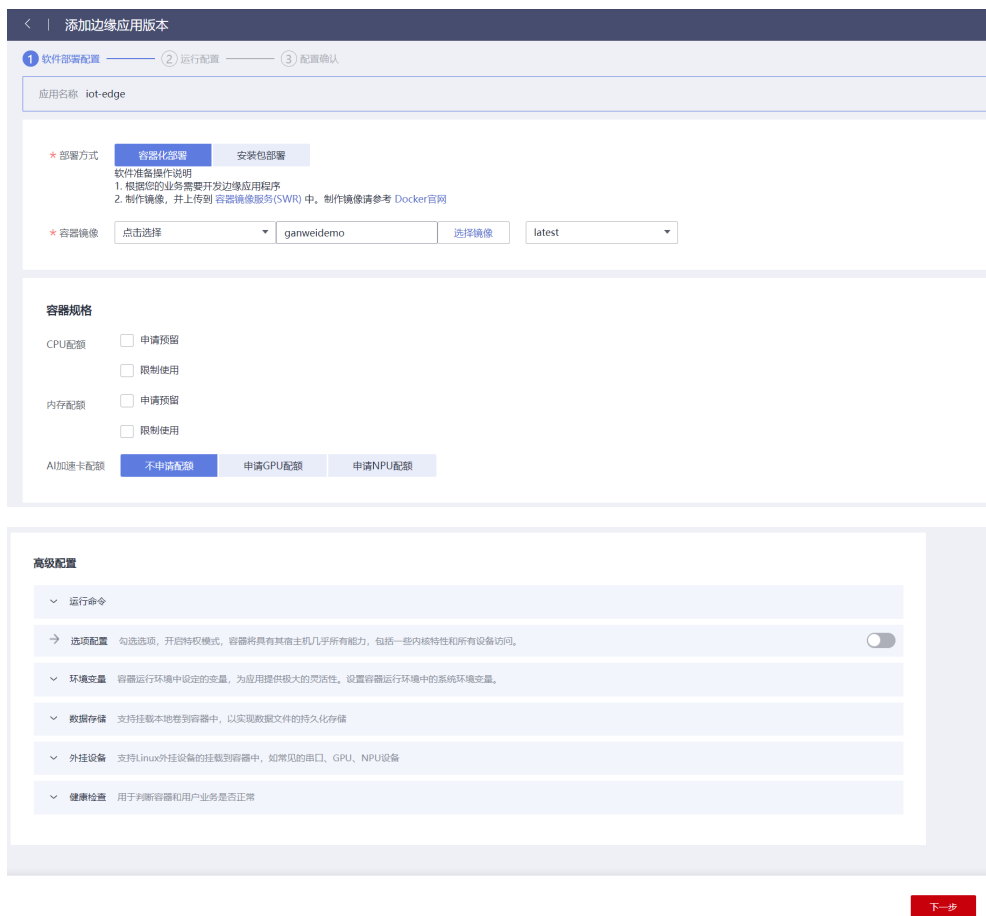
在IoT边缘控制台上添加应用和应用版本，即可将SWR云服务中的镜像通过IoT边缘部署到指定边缘节点中。

步骤1 访问[IoT边缘](#)，单击“管理控制台”进入IoT边缘控制台。

步骤2 选择左侧导航栏“边缘节点 > 应用管理”进入页面。

步骤3 单击右上角的“添加应用”，按照页面提示填写信息。

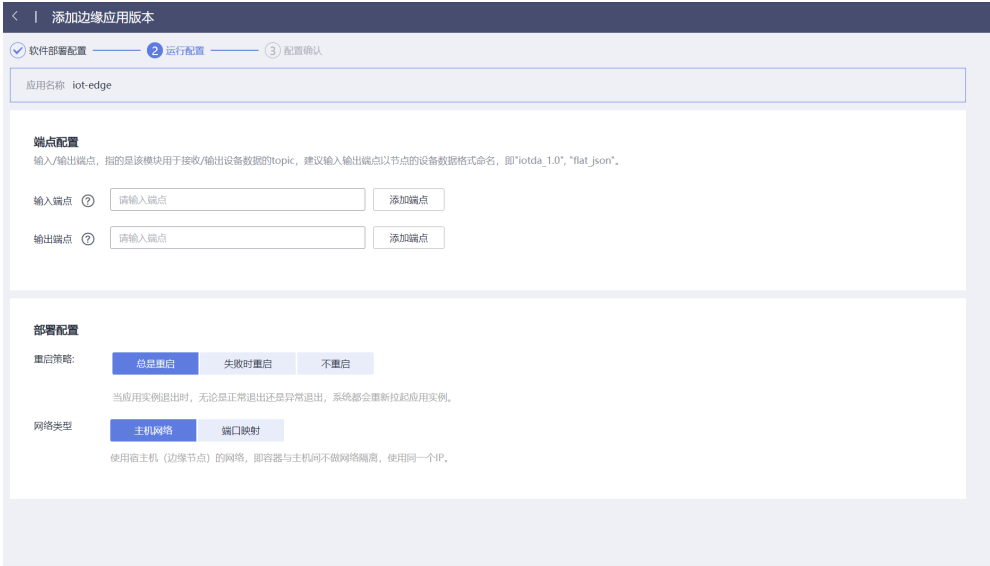
1. 软件部署配置



参数名称	说明
部署方式	选择“容器化部署”。
容器镜像	选择“单击选择”。 选择镜像：单击“选择镜像”按钮后，在右侧展开栏中选择SWR中的自定义容器镜像
容器规格	
CPU配额：	容器规格中CPU/内存配额，可以配置该容器的运行资源限制，即最小最大资源占用限制；此处不勾选则使用默认值。
内存配额：	
AI加速卡配额	不申请配额。
高级配置	
运行命令	启动程序的运行命令。

参数名称	说明
环境变量	<p>可按需配置环境变量,如下图所示:</p> 
数据存储	<p>可将主机某个目录挂载到容器中。卷类型有4中类型,分别为DB, LOG, CONFIG, 其它。本次实验配置3个卷如下图所示。</p>  <p>说明 本地卷的类型选择LOG/DB/CONFIG时,会给主机目录默认添加前缀: "/var/IoTEdge/{log db config}"。 若需要将容器目录直接挂载到主机某个目录,本地卷的类型需选择为“其他”。</p>
外挂设备	支持Linux外挂设备的挂载到容器中,如常见的串口、GPU、NPU 设备。
健康检查	用于判断容器和用户业务是否正常。

2. 运行配置



添加边缘应用版本

应用名称:

端点配置
输入/输出端点,指的是该模块用于接收/输出设备数据的topic,建议输入输出端点以节点的设备数据格式命名,即"iota_1.0", "flat_json".

输入端点:

输出端点:

部署配置

重启策略: 总是重启 失败时重启 不重启

网络类型: 主机网络 端口映射

使用宿主主机(边缘节点)的网络,即容器与主机间不做网络隔离,使用同一个IP.

上一步

参数名称	说明
端点配置	
输入端点	输入/出端点：若容器应用镜像集成了边缘SDK，并配置了输入/输出端点，则该容器应用可以从该输入端点获得设备数据，进行自定义的数据处理后，从输出端点将数据流回边缘处理中心（系统应用hub）。在“添加应用”时，也需配置输入/输出端点，为后续配置边缘流转规则提供前提条件。
输出端点	
部署配置	
重启策略	选择“总是重启”。
网络类型	选择“主机网络”。

📖 说明

配置数据流转规则需要选择应用的输入/输出端点；

自定义的容器应用想从hub获取设备数据，需配置响应的数据流转规则。

3. 配置确认

The screenshot shows the '添加边缘应用版本' (Add Edge Application Version) configuration page. It is divided into three main sections: '配置' (Configuration), '端点和部署配置' (Endpoint and Deployment Configuration), and '部署选项' (Deployment Options). The '配置' section includes fields for image address, AI acceleration, environment variables, and health checks. The '端点和部署配置' section includes input/output endpoints, restart strategy, and network type. The '部署选项' section includes version (1.0.0), architecture (x86_64), and business template selection.

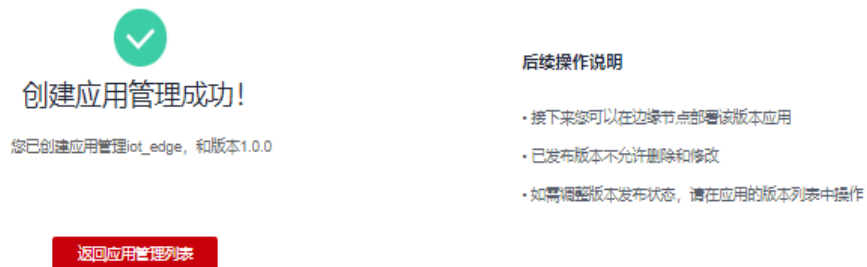
参数名称	说明
应用名称	自定义，如iot-edge
功能用途	根据容器应用的功能用途自行选择，本实验采用的镜像是进行数据清洗，故选择“数据处理”。
应用描述	自定义，可不填。
SDK版本	集成边缘SDK的版本号（ ModuleSDK下载链接 ）。
版本	选择镜像版本，如“1.0.0”，不勾选“支持多部署”。

参数名称	说明
支架架构	根据容器应用镜像支持的架构自行选择，本实验采用的镜像只支持x86架构，故选择“x86_64”。

📖 说明

配置数据流转规则需要选择应用的输入/输出端点；
自定义的容器应用想从hub获取设备数据，需配置响应的数据流转规则。

步骤4 单击“确认添加”后，则创建自定义应用成功。



步骤5 单击“返回应用管理列表”可查看到创建的应用，选择左侧版本列表，单击右侧的“版本信息 > 发布”。



步骤6 返回“应用管理”可查看到创建的应用，单击左侧的“版本列表”可查看已发布的应用版本。

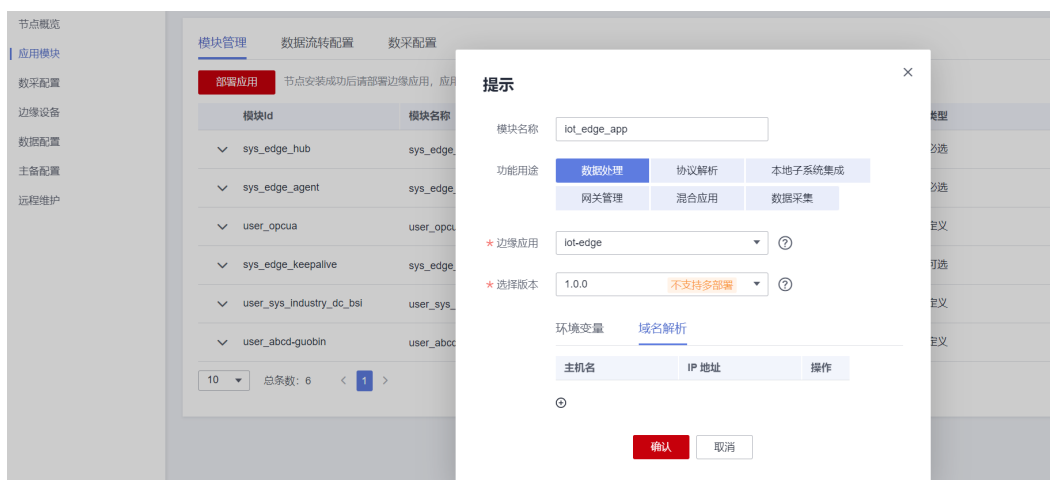


----结束

3.3.4 部署容器应用插件

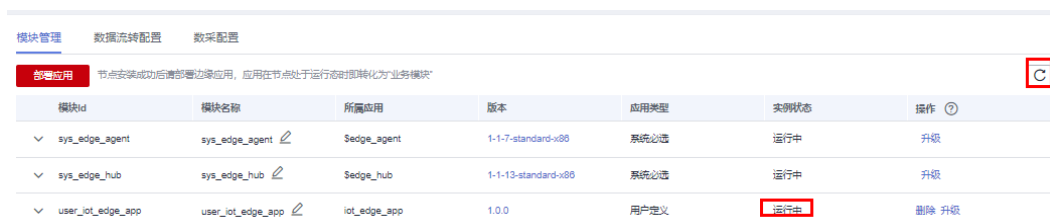
在IoT边缘控制台上将已创建的自定义应用部署到指定边缘节点中，即应用插件将在边缘节点上以容器身份运行，并在配置相应的流转规则后，可对设备数据进行清洗等数据处理。

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 选择左侧导航栏的“边缘节点 > 节点管理”，选择您的边缘节点名称，进入节点详情页。
- 步骤3** 在左侧导航栏选择“应用模块 > 模块管理”页签，单击“部署应用”，将制作的自定义应用部署到边缘节点上运行。



- 边缘应用：选择上述制作的应用名称，如：iot_edge_app
- 选择版本：选择已发布的与边缘节点相同架构的应用版本，如1.0.0
- 模块名称：自定义。

- 步骤4** 等待3秒后，刷新应用界面，可看到部署的自定义应用的实例状态为“运行中”，表明该应用容器已运行起来。



- 步骤5** 登录边缘节点服务器后台系统，执行命令。

```
> docker ps
```

可查看边缘节点上运行的容器列表，检查自定义应用容器已启动运行。若配置了卷挂载，则可在主机相应目录下查看日志/数据库等信息。也可通过命令 **docker exec -it {容器ID} env** 查看创建应用时配置的环境变量是否在容器的环境变量中。

```
root@standard-edge-yxh:~# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
52b0d2486d69   swr.cn-north-4.myhuaweicloud.com/iot_edge_test/app:1.0.0   "/opt/iot/edge_rule/..." 6 minutes ago  Up 6 minutes
ba9bd368e8b   swr.cn-north-4.myhuaweicloud.com/huawei-iot-edge/edgehub:2021.0323.2125.01  "/opt/iot/edge_hub/..." 38 hours ago   Up 38 hours   0.0.0.0:7883
83/tcp, 0.0.0.0:8943->8943/tcp   sys-edge-hub
0fe22802680   swr.cn-north-4.myhuaweicloud.com/huawei-iot-edge/edgeagent_x86_64:0.3.072.20210309103747  "/bin/sh -c 'sh /opt/..." 38 hours ago   Up 38 hours
                                sys-edge-agent
root@standard-edge-yxh:~# cd /var/IoTEdge/log/
root@standard-edge-yxh:/var/IoTEdge/log# ll
total 32
drwxr-xr-x 8 root root 4096 Apr 15 09:39 ./
drwxr-xr-x 5 root root 4096 Apr 13 19:26 ../
drwxr-xr-x 3 root root 4096 Apr 15 08:08 edge_daemon/
drwx----- 3 root root 4096 Apr 13 19:26 edge_installer/
drwx----- 3 root root 4096 Apr 14 18:35 sys_edge_access/
drwx----- 3 root root 4096 Apr 13 19:26 sys_edge_agent/
drwx----- 3 root root 4096 Apr 13 19:27 sys_edge_hub/
drwx----- 3 root root 4096 Apr 15 09:39 user-iot-edge-app/
```

```
root@standard-edge-yxh:/var/IoTEdge/log# docker exec -it 52b0d2486d69 env
PATH=/usr/local/jre/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=standard-edge-yxh
TERM=xterm
storage_period=0
deploy_type=cloud
edge_http_server_ip=127.0.0.1
edge_http_server_port=8943
edge_mqtt_server_port=7883
device_id=579429216366637056
verify_code=oS9KN0h79jdpmGM-w0rsTLCZvu1ov2AaIc7881Ave14=
edge_mqtt_server_ip=127.0.0.1
product_id=5eb52033c235d506b92b8fe4
security_level=HIGH
module_id=user-iot-edge-app
cloud_http_server_ip=iotedge-south.cn-north-4.myhuaweicloud.com
cloud_http_server_port=8943
node_type=standard
project_id=05068467bb8026bc2f64c010b1e6fb8c
edge_host_ip=127.0.0.1
module_cfg_dir=/opt/iot/edge_rule/secret
app_env=for_test
JRE_HOME=/usr/local/jre
CLASSPATH=./usr/local/jre/lib/rt.jar
HOME=/home/edge
```

----结束

3.3.5 资源释放

实验完成，您可以按如下步骤删除在实验过程中，产生在您的华为云账号下的相关资源。

注：所有资源删除动作，在“华北-北京四”区域下进行。

删除弹性云服务器 ECS

- 步骤1** 访问[弹性云服务器 ECS](#)，单击“进入控制台”，进入云服务器控制台。
- 步骤2** 在页面左上角切换区域至“华北-北京四”。
- 步骤3** 在左侧导航中选择“弹性云服务器”，按需勾选待删除ECS服务器，如ecs-iot，选择右边操作列“更多 > 删除”。

请勾选“释放云服务器绑定的弹性公网IP地址”和“删除云服务器挂载的数据盘”。



----结束

删除自定义应用

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 选择左侧导航栏“边缘节点 > 应用管理”进入页面。
- 步骤3** 根据应用名称选择需要删除的自定义应用，如iot_edge_process，单击操作列的“删除”即可。

----结束

删除边缘节点

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 选择左侧导航栏“边缘节点 > 节点管理”进入页面。
- 步骤3** 根据节点名称选择，如TestEdge01，单击右边操作列的“删除”，并在弹出的窗口中单击“确认”。

删除节点需要一个过程，不超出1分钟，请耐心等待。



----结束

删除虚拟私有云 VPC

- 步骤1** 访问[虚拟私有云 VPC](#)，单击“访问控制台”进入网络控制台。
- 步骤2** 选择左侧导航栏“子网”，删除子网，如subnet-iot，请根据节点名称选择，单击操作列的“删除”，并在弹出的窗口中单击“是”。
- 步骤3** 选择左侧导航栏“虚拟私有云”，删除虚拟私有云，如vpc-iot，请根据节点名称选择，单击操作列的“删除”，并在弹出的窗口中单击“确定”。

----结束

删除 SWR 镜像

- 步骤1** 访问[容器镜像服务 SWR](#)，单击“立即使用”，进入容器镜像服务控制台。
- 步骤2** 选择左侧菜单栏“我的镜像”，选中不需要的镜像，如app，单击左上角“删除”。

----结束

4 基于 IoT 边缘实现模拟 MQTT 设备的快速接入

4.1 实验简介

本实验将指导您搭建一个基于IoT边缘服务的边缘节点，并通过边缘节点实现设备通过原生MQTT协议以设备直连边缘节点模式接入。

通过本实验，您将能够：

- 了解并掌握IoT边缘服务的功能特点与使用方法。
- 掌握设备通过IoT边缘节点接入物联网平台的方法。
- 掌握MQTT.fx模拟器的用法。

4.2 前提条件

4.2.1 前提概要

1. **前提：** 体验需要三方配合，首先要开通IoTEdge，用于节点管理和应用管理，再开通IoTDA，用于设备接入与管理，还得有边缘服务器作为节点被IoTEdge纳管。
2. **边缘服务器：** 第一点提到的IoTEdge和IoTDA是华为云服务，使用华为云租户账号直接免费开通可用，边缘服务器属于硬件，可以购买低配的ECS代替，其EIP则作为边缘服务器的节点IP地址。
3. **在IoTEdge上注册节点：** 在购买的ECS上执行安装脚本命令，即自动部署好2个系统应用，agent用于节点应用管理，hub用于数据通信。
4. **网关设备：** 在上一步注册节点的同时，IoTEdge会自动在IoTDA侧创建一个网关设备，即IoTDA把整个边缘节点（也即边缘网关）当成一个网关设备。该网关设备用于边缘节点与IoTDA之间的数据交互。
5. **边缘设备：** 在网关设备下注册边缘设备，则相当于边缘设备是网关下的子设备，用于边缘设备与边缘节点之间的数据交互。本示例采用MQTT.fx工具模拟边缘设备。
6. **数据上报：** 边缘设备上报的数据是到边缘节点（即购买的ECS服务器），然后由节点上的系统应用（hub）负责json格式解析，再根据该边缘设备与网关设备的父子关系，重新组装json数据格式，用网关设备上报给IoTDA。

概述：边缘设备上报数据到边缘节点，然后节点上的系统应用把数据解析重新拼装，以网关设备上报的方式把数据报给IoTDA。数据上报流程不经过IoTEdge云服务，IoTEdge主要做节点和应用管理。

4.2.2 安装 MQTT 设备模拟器

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

MQTT.fx 1.7.0及旧版本对带有\$的主题（Topic）处理存在问题，请使用最新版本进行测试。

在设备对接边缘节点场景中，需要在设备侧集成相应证书。

证书类型，证书格式，以及用途详见下表。

证书名称	证书类型	证书格式	说明	下载
plt-device-ca	设备侧证书	pem	用于设备校验边缘节点的身份。	证书文件

4.3 操作步骤

4.3.1 创建边缘节点

注册专业版节点

步骤1 访问[IoT边缘](#)，单击“管理控制台”进入IoT边缘控制台。

步骤2 选择左侧导航栏“边缘节点 > 节点管理”单击右上角“注册节点”。

步骤3 根据页面提示填写参数，勾选右下角“《华为云服务等级协议》”单击“下一步”配置节点数据。

图 4-1 专业版

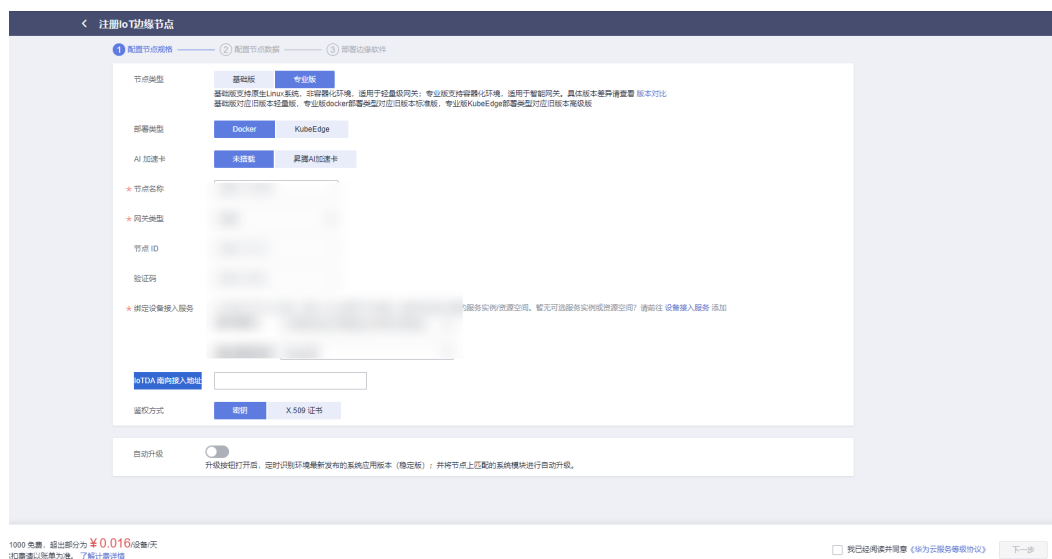


表 4-1 节点规格参数说明

参数名称	参数说明
节点类型	专业版。
部署类型	Docker。
AI加速卡	<ul style="list-style-type: none"> 未搭载：边缘节点未使用AI加速卡时选择。 昇腾AI加速卡：支持华为昇腾AI加速设备，边缘节点的服务器搭载相关昇腾AI加速设备时选择。
昇腾驱动地址	可选，AI加速卡选择昇腾AI加速卡时需要保证安装边缘节点的服务器上已经安装昇腾设备驱动，并根据前端提示填入驱动地址。若改值为空，默认填入Atlas300驱动默认安装地址： /usr/local/Ascend/driver/lib64/。
节点名称	自定义节点名称，如“test”。 说明 节点名称可以在：“节点概览”-“基本信息”中编辑修改。
网关类型	other。
节点ID	自定义（可选填写）。只允许中文，英文字母、数字、中划线（-），下划线（_），最小长度1，最大长度64。 说明 边缘节点ID自定义(不输入默认生成)。
验证码	输入自定义验证码（可选填写）。验证码，执行安装边缘软件命令时与云端通讯的凭证。
设备归属实例	选择默认归属实例。
设备归属资源空间	选择默认资源空间，或为本实验单独新建一个资源空间，如edge。
IoTDA南向接入地址	选择网关归属的IoTDA实例的南向设备mqtt接入地址，也可以选择自定义的南向接入地址，支持“ip:端口”或“域名:端口”形式。
鉴权方式	密钥：默认生成。 X.509证书：请参考： 基于MQTT.fx的X.509证书接入指导 。 警告 鉴权方式为“密钥”的节点，请勿在IoTDA界面重置节点的密钥，重置密钥会导致网关与IoTDA通信异常，且不可恢复。
指纹	证书文件生成的字符串（可选填写）。只能为40位十六进制字符串或64位六进制字符串。
本地路径	输入证书文件和密钥文件在设备上的存储路径，只允许字母、数字、下划线（_）和斜线（/），以/开头并且以.pem或.crt结尾。
自动升级	可选，开启后节点将自动升级系统应用模块。 注意 升级过程中可能导致节点数据上传中断。

步骤4 填写配置节点数据信息后，单击“立即创建”。

图 4-2 配置节点数据

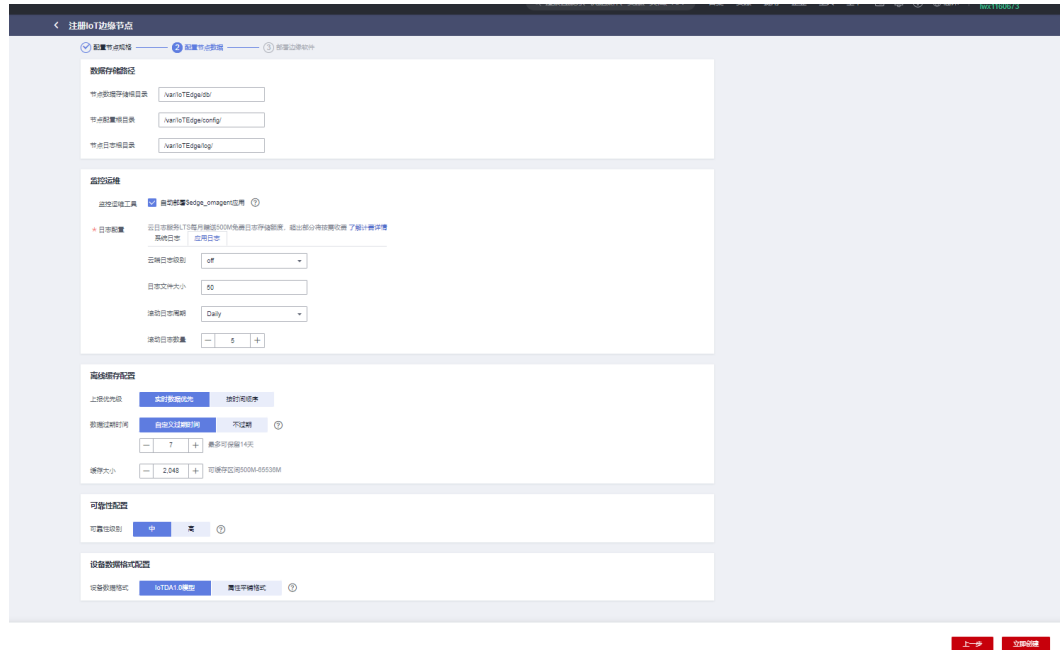


表 4-2 专业版节点信息

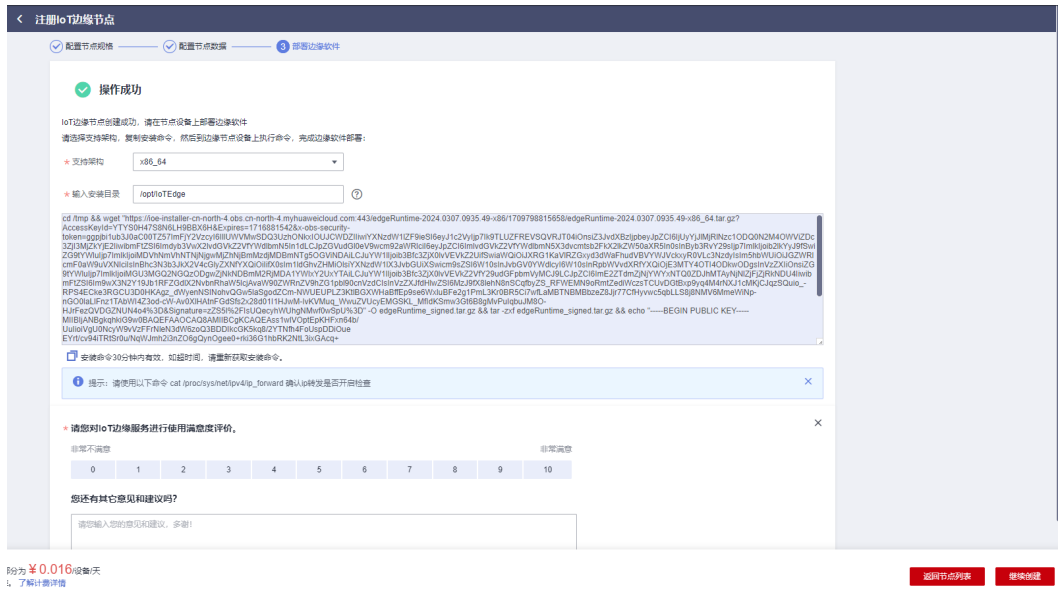
参数名称	说明
数据存储路径	默认（允许修改存储路径）。
监控运维	默认勾选“监控运维工具”。
\$edge_omagent应用	负责边缘节点远程监控和运维，提供日志上报、远程SSH、CPU等指标上报、文件上传下载能力。

参数名称	说明
日志配置	<p>系统日志：边缘节点上已部署的系统应用产生的日志。</p> <p>应用日志：边缘节点上已部署的用户定义应用所产生的日志。</p> <ul style="list-style-type: none">• 云端日志级别：on, off, trace, debug, info, warn, error, fatal。其中，on表示上传所有日志至云端，off表示不上传日志至云端，其余可选项代表按日志级别采集，选择后将只采集目标级别及以上的日志（模糊匹配）。日志级别的严重程度等级由低到高依次为：trace, debug, info, warn, error, fatal。• 日志文件大小：日志文件大小限制，单位MB，默认50，取值范围10-1000。某个日志文件如果达到大小限制，则会将文件转储保存至日志目录下。• 滚动日志周期：日志转储周期，可选项：daily, weekly, monthly, yearly。某个日志文件如果达到转储周期，则会将文件转储保存至日志目录下。日志文件大小和滚动日志周期是同时生效的，满足任何一个条件都会进行日志转储。• 滚动日志数量：日志文件转储个数，默认5，取值范围1-10。边缘节点保存的转储日志数量如果达到限制，则会删除最早的转储文件。 <p>注意</p> <ol style="list-style-type: none">1. 当前“云端日志级别”对日志等级采取模糊匹配的策略，若日志消息中存在与日志级别标识相同的关键字，则也可能被采集。2. 若要开启“云端日志级别”中的trace, debug, info, warn, error, fatal选项，则必须满足以下三个前置条件：<ol style="list-style-type: none">1. 目标节点是基础版节点或部署类型为Docker的专业版节点。需注意，部署类型为KubeEdge的专业版节点不支持此功能。2. 已安装\$edge_omagent应用，且版本不低于1-1-15-xxx。3. 应用日志需遵循如下规范打印，以便进行过滤：<ul style="list-style-type: none">○ 日志级别标识需打印为（大小写均可）：trace、debug、info、warn、error、fatal。○ 每条日志消息以YYYY-MM-DD日期格式作为前缀。 <p>说明</p> <p>专业版（部署类型为Docker）和基础版节点只涉及日志级别这一个参数，其他参数只对专业版（部署类型为KubeEdge）有效。</p>
离线缓存配置	<p>可对离线节点上报的数据进行缓存，定义缓存周期（可配置为不过期）跟缓存大小、上报的优先级、路由转发是否开启离线缓存。仅当上报优先级为“按时间顺序”时支持路由转发开启离线缓存。</p>
可靠性配置	<p>可靠性级别分为“中”和“高”2种。中级别：当应用上报属性和点位的频率超过流控值时，超过部分会被丢弃；高级别：当应用上报属性和点位的频率超过流控值时，通过控制模块的发送速度，降低上报频率，报文不会被丢弃。（注意：高可靠性级别的配置需要定制应用的配合，同时会造成性能的损失）。</p>

参数名称	说明
设备数据格式配置	<p>设备的数据格式：即单网关节点中设备的标准数据格式，包含 IoTDA1.0物模型格式和属性平铺格式。IoTDA1.0物模型中包含了设备、服务和属性三层结构，具体参考IoTDA网关子设备属性上报报文格式。属性平铺格式，包含设备、属性2层结构，属性结构可以包含自定义结构的拼接，在json中作为一项平铺展开，具体格式请参考步骤6设备数据格式配置。</p> <p>须知 属性平铺格式功能，需要edge_hub应用1-1-38-xx版本及以上支持</p>

步骤5 弹出如下对话框，您可以“返回节点列表”也可以“继续创建”。

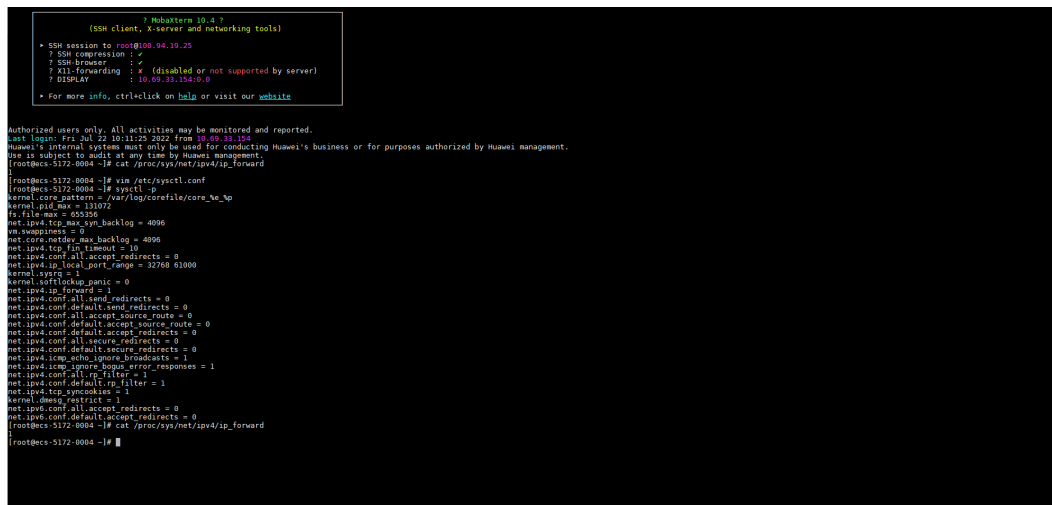
图 4-3 注册成功



说明

安装部署节点前，请确认机器的IP转发已开启并且已开启自启动，以及7883和8943端口开放。其中，IP转发规则配置如下：

图 4-4 IP 转发规则配置



1. 检查机器ip_forward in/proc文件系统的值。
`cat /proc/sys/net/ipv4/ip_forward`
 若值等于0，则需要添加配置IP转发规则。
2. 检查配置文件/etc/sysctl.conf是否包含net.ipv4.ip_forward = 1。
 如果配置文件中没有net.ipv4.ip_forward参数项，请新增一行net.ipv4.ip_forward = 1。
`vim /etc/sysctl.conf`
 按“i”进入编辑模式，修改完成后按“Esc”退出编辑模式，然后按“Shift” + “;”组合键，输入wq并回车，保存修改并退出。
3. 重新加载此文件的值。
`sysctl -p`
4. 检查7883和8943端口是否开放，如果是华为云服务器，请参考：[安全组](#)。

----结束

安装专业版节点

步骤1 在“节点管理”的节点列表选择您需要安装的节点，单击右侧“安装”，获得提示。

图 4-5 安装

节点名称	节点类型	资源包类型	资源包ID	状态	主机名/网络	操作
	专业版	--	--	未安装		删除 安装
	专业版	--	--	离线	172.30.7.64,172.17.0.1,1...	删除
	专业版	--	--	已安装		删除


步骤2 选择支持架构和安装目录，然后单击  复制安装命令，使用SSH工具以root用户登录边缘节点服务器后台系统，执行安装命令。

图 4-8 安装命令执行成功

```
2024-05-28T14:44:29,581321 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:PENGOING, reason:(zero) ]
2024-05-28T14:44:31,632185 | INFO | saveBasePathInfo | 78 | config base path start.
2024-05-28T14:44:31,632291 | INFO | saveBasePathInfo | 83 | config log_path:/var/IoTEdge/log/, config_path:/var/IoTEdge/config/
2024-05-28T14:44:31,632457 | INFO | saveBasePathInfo | 95 | config base path success.
2024-05-28T14:44:31,632535 | INFO | addAgentCustomConfigs | 261 | addCustomConfigs start.
2024-05-28T14:44:31,632603 | INFO | addAgentCustomConfigs | 285 | addCustomConfigs end.
2024-05-28T14:44:31,632654 | INFO | pullImage | 319 | start pull image, deploy type is cloud
2024-05-28T14:44:35,888319 | INFO | createModule | 76 | create Module, name:sys_edge_agent
2024-05-28T14:44:35,939013 | INFO | createModule | 94 | module created, CreateRespBody: ModuleCreateResp[ moduleId:sys_edge_agent, moduleName:sys-edge-agent, appId:sedge_agent, appVersion:test-tpm-2-x86, containerId:4b3e43cee7eac9bc282b994377abfb148a264389a36856f3cacc9338743036e, processId:(zero), type:docker, ]
2024-05-28T14:44:35,939162 | INFO | startModule | 160 | startModule moduleId:sys_edge_agent
2024-05-28T14:44:42,398934 | INFO | listModule | 250 | listModule
2024-05-28T14:44:42,440775 | INFO | checkModulesBo | 340 | start check modules, time:1
2024-05-28T14:44:42,440908 | INFO | listModule | 250 | listModule
2024-05-28T14:44:42,491163 | INFO | printModulesAndReport | 484 | ***success:1 failed:0
2024-05-28T14:44:42,491339 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:RUNNING, reason:(zero) ]
2024-05-28T14:44:42,542192 | INFO | reportEvent | 158 | Begin reporting event: {"events":[{"resource":"node.installation","location":"zzzzz","event":"succeeded","event_severity":"Info", "event_time":"20240528T044442Z","notify_data":{"host_tag":"DEFAULT","edge_node_id":"zzzzz"}}]}
2024-05-28T14:44:42,529233 | INFO | reportModuleStatus | 138 | reportModuleStatus, modulesInfo:ModuleStatusInfoList[ host_tag:DEFAULT, deploy_id:1716878669308, modules:[ModuleStatusInfo module_id:sys_edge_agent, app_id:sedge_agent, app_version:test-tpm-2-x86, container_state:RUNNING, liveness_state:(zero), update_time:1716878669305,]], ]
2024-05-28T14:44:43,643689 | INFO | checkBridgeNetwork | 556 | Begin checking bridge network
2024-05-28T14:44:50,524137 | INFO | install | 494 | install edge finish
2024-05-28T14:44:50,524559 | INFO | stop | 545 | edgeinstaller stop.
/root
```

步骤3 单击“我知道了”，等待边缘节点的状态变为“在线”，表示该节点已安装并连接成功。

步骤4 您可以单击“节点名称”，查看该节点详细信息，关于节点详情描述可参见[节点概览](#)。

----结束

4.3.2 (可选) 部署边缘应用

在边缘节点安装成功后可部署边缘应用，本示例只演示MQTT设备接入，在安装边缘节点时会自动部署好2个系统应用，无需操作。

步骤1 访问[IoT边缘](#)，单击“管理控制台”进入IoT边缘控制台。

步骤2 选择左侧导航栏“边缘节点 > 节点管理”进入页面。

步骤3 选择您的边缘节点，单击“节点名称”进入节点详情页。

步骤4 选择左侧导航栏“应用模块 > 模块管理”页签，查看已部署边缘应用。

专业版（Docker）默认部署sys_edge_hub和sys_edge_agent，专业版（KubeEdge）默认部署sys_edge_hub。

实例状态显示为“运行中”，表示应用运行正常。

名称	模块名称	模块ID	版本	容器ID	实例状态	操作
sys_edge_agent	sys_edge_agent	sedge_agent	1-17-standard-c88	标准应用	运行中	删除
sys_edge_hub	sys_edge_hub	sedge_hub	1-13-standard-c88	标准应用	运行中	删除

----结束

4.3.3 设备接入边缘节点（MQTT）

IoT边缘应用部署完成后，边缘节点即作为远端物联网平台在设备侧的延伸，云边协同，共同完成设备管理。边缘节点可以为近端连接的设备就近提供计算和管理服务，如低时延业务本地管理、与云端断链时的本地控制和规则执行等。设备通过接入边缘节点，最终通过边缘节点将数据上传到远端物联网平台。

创建产品

对于设备通过边缘节点上报至物联网平台的数据，云端物联网平台要能够解析设备上报的数据。故需要在设备接入服务侧完成设备的产品模型，编解码插件的开发。

步骤1 访问[IoT边缘](#)，进入IoT边缘控制台。

步骤2 选择左侧导航栏“边缘节点 > 设备建模”进入页面。

步骤3 单击右上角的“创建产品”，按照页面提示填写信息，然后单击“立即创建”。



创建产品

* 所属资源空间 ?

* 产品名称

协议类型

* 数据格式 ?

厂商名称

* 设备类型 ?

立即创建 取消

参数名称	说明
所属资源空间	所属资源空间：选择创建边缘节点时，绑定设备接入服务时选择的设备归属资源空间，如edge。
产品名称	自定义，如iot_edge。
协议类型	MQTT。
数据格式	JSON。
厂商名称	自定义，如Huawei。
设备类型	自定义，如MQTT_Device。

步骤4 单击“产品名称”下的产品iot_edge进入产品详情页面。

步骤5 在“模型定义”页签下，单击“自定义模型”。

产品模型是用来描述设备能力的文件，通过JSON的格式定义了设备的基本属性、上报数据和下发命令的消息格式。定义产品模型，即在物联网平台构建一款设备的抽象模型，使平台理解该款设备支持的属性信息。



步骤6 填写服务ID、服务类型，单击“确认”。

服务ID/服务类型：自定义，如sensor

步骤7 展开服务列表，单击“添加属性”，分别添加温度、湿度两个属性。

添加“温度”属性，属性名称为“temp”，如下图所示：

新增属性 ×

* 属性名称 必选

属性描述

* 数据类型

* 访问权限

取值范围 -

步长

单位

添加“湿度”属性，属性名称为“humi”，如下图所示：

新增属性 ×

* 属性名称 必选

属性描述

* 数据类型

* 访问权限

取值范围 -

步长

单位

步骤8 单击“添加命令”，添加一条控制命令。

1. 填写命令名称，自定义，如control_light

新增命令

* 命令名称

下发参数

参数名称	数据类型	是否必选	操作	描述
暂无表格数据				

响应参数

参数名称	数据类型	是否必选	操作	描述
暂无表格数据				

- 单击“新增下发参数”，填写参数信息，并单击“确认”。
 - 参数名称: light
 - 数据类型: string(字符串)
 - 长度: 3
 - 枚举值: ON,OFF

新增参数

* 参数名称 必选

参数描述

* 数据类型

* 长度

枚举值

- 单击“新增响应参数”，填写参数信息，并单击“确认”。
 - 参数名称: light_state
 - 数据类型: int(整型)
 - 取值范围: 0-1

新增参数

* 参数名称 必填

参数描述

* 数据类型

取值范围 -

步长

单位

4. 在“新增命令”窗口，单击“确认”，完成命令添加。

步骤9 模型定义完成，参数信息预览如下图所示。

属性名称	数据类型	是否必填	初始化式	描述	操作
temp	int	true	可选可不写		删除 编辑
hum	int	true	可选		删除 编辑

命令名称	下发参数	接收参数	操作
control_light	light	light_state	删除 编辑

---结束

添加边缘设备

步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。

步骤2 选择左侧导航栏的“边缘节点 > 节点管理”，选择您的边缘节点名称，进入节点详情页。

步骤3 选择左侧导航栏“边缘设备”页签，单击“添加边缘设备”，填写参数后，单击“确认”。

- 所属产品：已创建的产品，如iot_edge
- 设备标识码：自定义，如iot_edge_test_0319
- 设备名称：自定义，如iot_edge_test_0319
- 模块ID：不填 [添加边缘设备的什么时候填写模块ID?](#)
- password：自定义，如12345678

提示 ×

接入设备需统一纳入物联网平台管理，并和当前边缘节点归属同一个设备接入服务实例/资源空间。
归属服务实例 IoTDA默认基础版
归属资源空间 DefaultApp_hwstaff_pub_loTEdge_1ot

* 所属产品 C
没有可选产品？请前往设备接入服务增加自定义产品，并定义设备功能前往添加产品

* 设备标识码

* 设备名称 ?

模块 ID ?

password

确认 取消

步骤4 添加设备成功，请保存好设备ID和密钥，用于设备接入平台认证。

×

✓ **操作成功**

设备注册成功，请保存好以下信息，用于设备接入平台认证。

设备ID

设备密钥

我知道了

步骤5 完成添加设备后，可以在设备列表中查看设备状态，并管理边缘设备。

图 4-9 边缘设备列表

设备名称	设备ID (单击进入设备详情)	所属产品	接入协议	操作
<input type="checkbox"/> iot_edge_test_0319	<input type="checkbox"/> iot_edge_test_0319	MQTT	MQTT	配置 删除 管理子设备

也可以单击“设备ID（单击进入设备详情）”跳转至“设备接入服务控制台”进行设备管理。

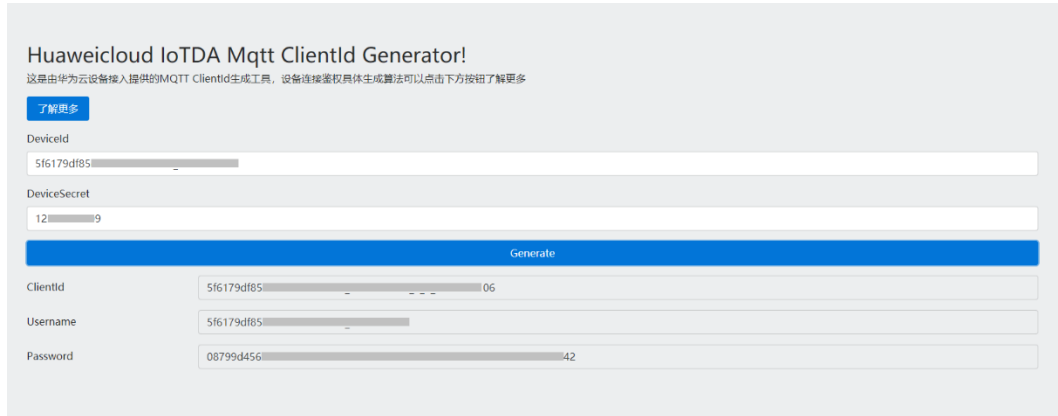
边缘设备是作为子设备挂在节点设备下的。

----结束

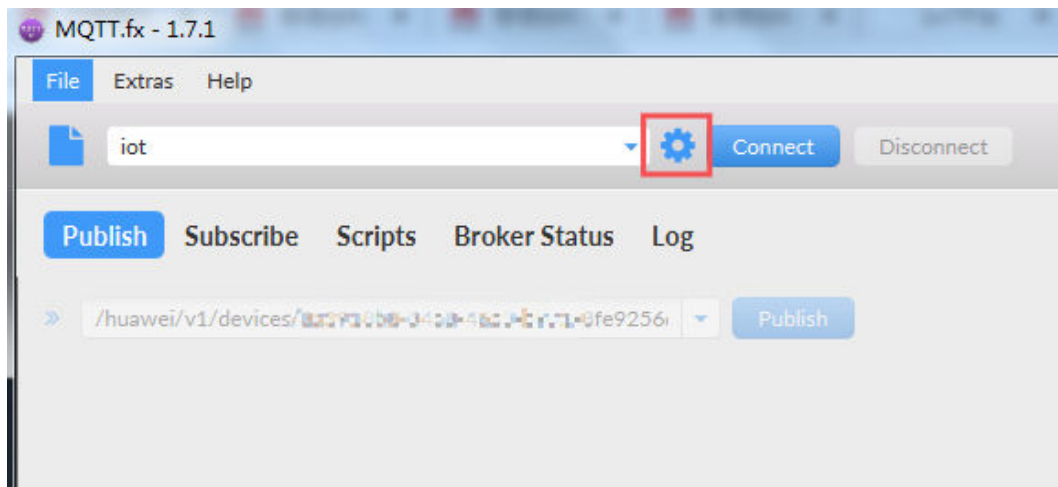
设备接入调测

本章节以MQTT.fx为例，介绍以MQTT原生协议接入IoT边缘节点，通过边缘节点上报设备属性至物联网平台。MQTT.fx是目前主流的MQTT客户端，可以快速验证是否可以与物联网平台服务交互发布或订阅消息。

步骤1 访问[这里](#)，填写**添加边缘设备**后生成的设备ID和设备密钥，生成连接信息（ClientId、Username、Password）。

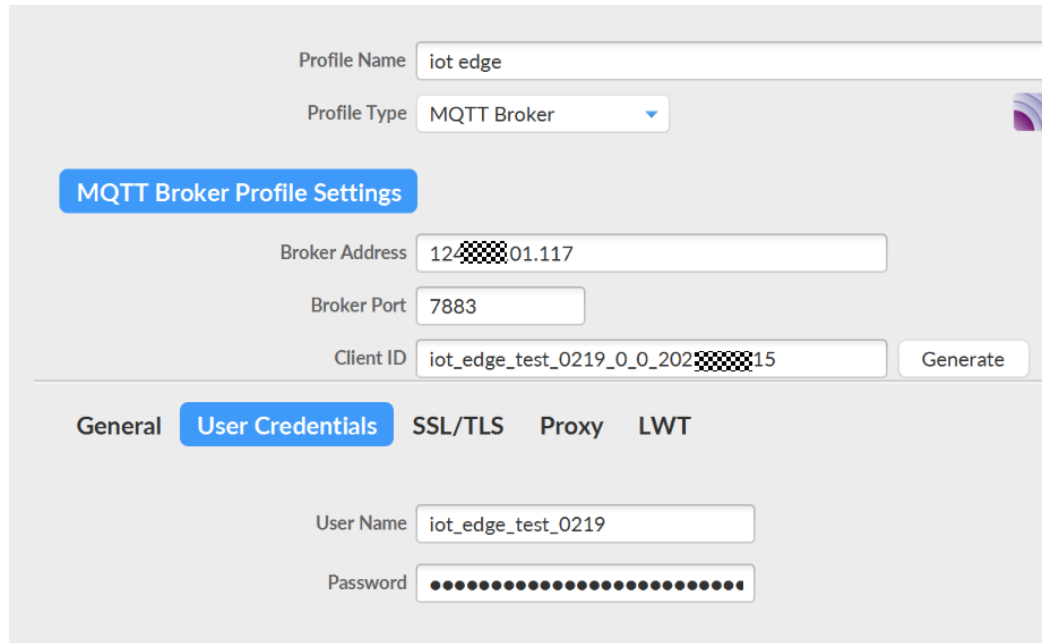


步骤2 打开MQTT.fx软件，单击设置图标。

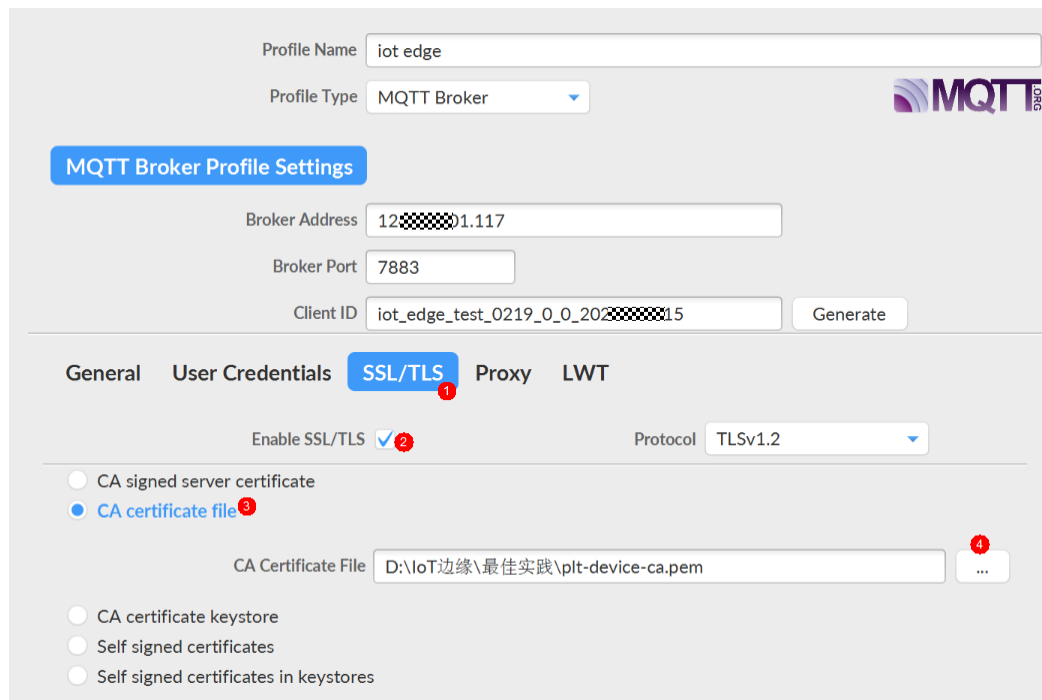


步骤3 参考截图中配置鉴权参数，然后单击截图中“Apply”。

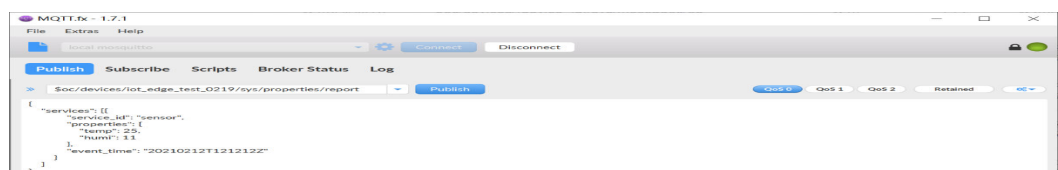
- Broker Address: 填写边缘节点服务器公网IP地址
- Broker Port: 填写边缘节点提供给南向MQTT(S)设备接入的端口，即7883
- Client ID: 设备cliendID，参考1获取
- User Name: 即设备ID，参考1获取
- Password: 加密后的设备密钥，参考1获取



在设备对接边缘节点场景中，需要在设备侧集成相应证书（[证书文件](#)），用于设备校验边缘节点的身份。



步骤4 单击“Connect”，设备鉴权成功后，在IoT边缘控制台和物联网平台可以看到设备处于在线状态。如果模拟器右侧的红灯变成绿灯，说明与物联网平台的连接成功。



IoTEdge云服务上边缘节点显示子设备在线：

边缘节点 / TestEdge01 / iot_edge_test_0219

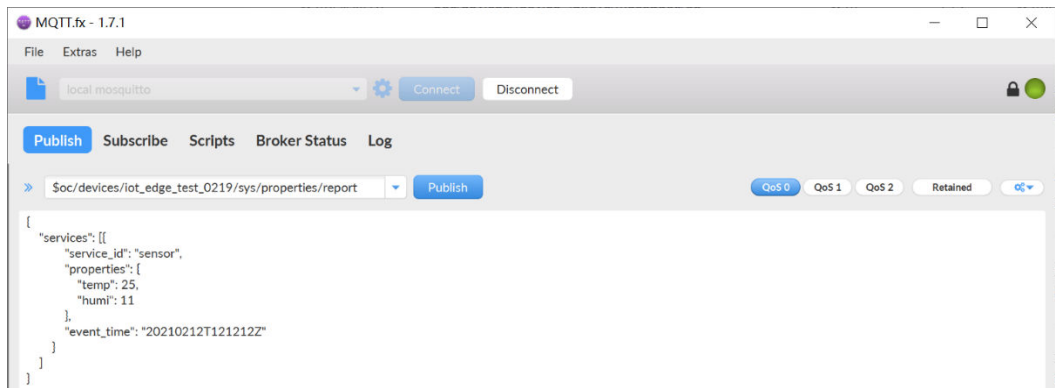


物联网平台显示边缘节点、子设备在线：



步骤5 通过MQTT.fx模拟设备上报属性，并在物联网平台查看设备上报数据。

在Publish页签，填写设备属性上报Topic，及JSON消息体，单击“Publish”完成上报。

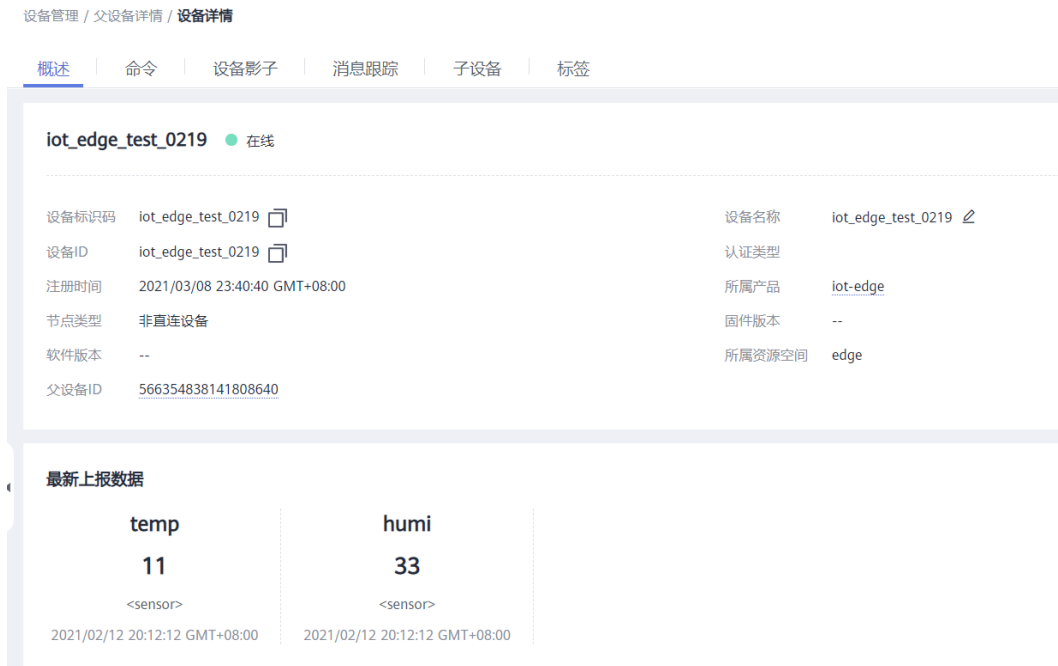


- Topic: \$oc/devices/iot_edge_test_0219/sys/properties/report
其中，iot_edge_test_0219为设备ID，请替换为实际值。

- 消息体：

```
{
  "services": [
    {
      "service_id": "sensor",
      "properties": {
        "temp": 25,
        "humi": 11
      },
      "event_time": "20210212T121212Z"
    }
  ]
}
```

在物联网平台查看设备上报数据。

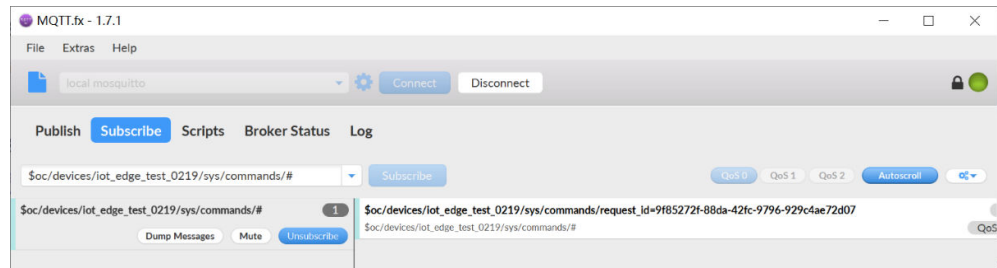


步骤6 通过MQTT.fx订阅命令下发Topic，并在物联网平台下发控制命令，查看设备是否收到平台下发的命令。

1. 在Subscribe页签，填写命令下发Topic，并单击“Subscribe”完成订阅。

Topic: \$oc/devices/iot_edge_test_0219/sys/commands/#

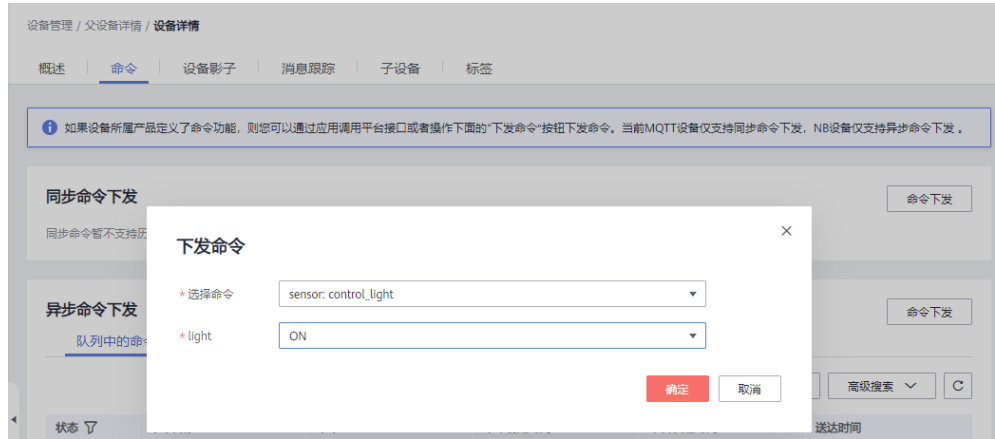
其中，iot_edge_test_0219为设备ID，请替换为实际值。



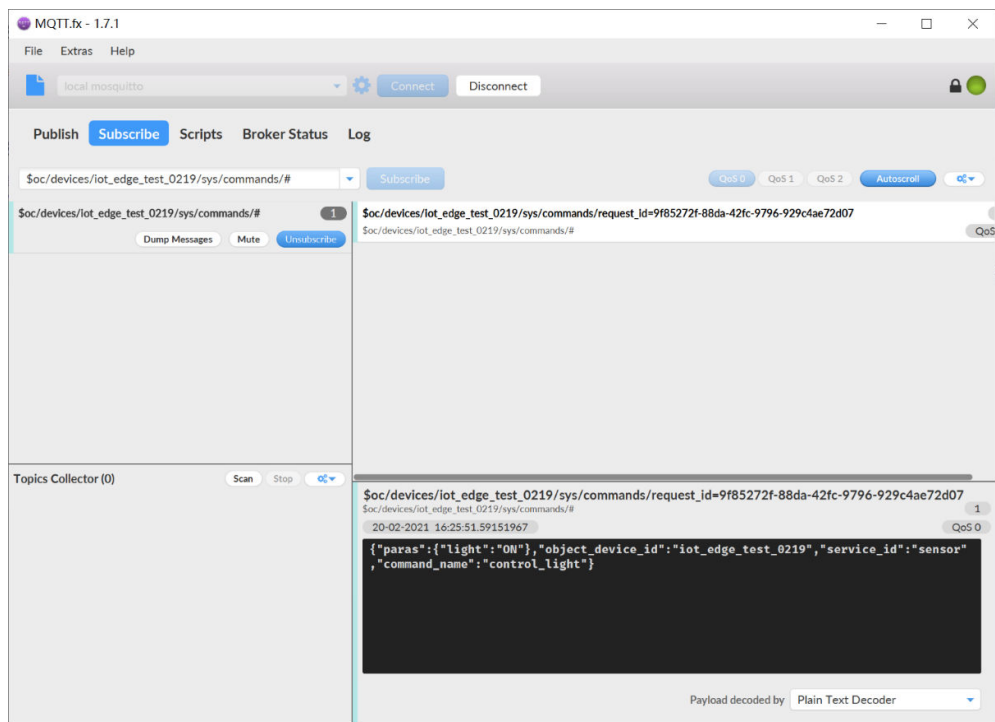
2. 在物联网平台执行命令下发。

进入设备详情页，选择“命令”页签，单击同步命令下发区域的“命令下发”，设置参数并单击“确定”完成命令下发。

- 选择命令：sensor: control_light
- light: ON



在MQTT.fx客户端查看，平台下发命令接收成功。



----结束

4.3.4 资源释放

实验完成，您可以按如下步骤删除在实验过程中，产生在您的华为云账号下的相关资源。

注：所有资源删除动作，在“华北-北京四”区域下进行。

删除弹性云服务器 ECS

- 步骤1** 访问[弹性云服务器 ECS](#)，单击“进入控制台”，进入云服务器控制台。
- 步骤2** 在页面左上角切换区域至“华北-北京四”。
- 步骤3** 在左侧导航中选择“弹性云服务器”，按需勾选待删除ECS服务器，如ecs-iot，选择右边操作列“更多 > 删除”。

请勾选释放弹性公网IP和删除数据盘。

×

删除



删除云服务器会同时删除系统盘对应快照，确定要进行删除操作吗？

删除的云服务器和磁盘无法恢复。云服务器删除完成后，对应的磁盘需要1分钟左右才能完成删除。此时不要对磁盘有任何操作，否则可能导致云服务器故障或磁盘删除失败，需要重新执行删除操作。

删除云服务器时保留关联的云服务器备份，该备份继续收费，可在云服务器备份页面执行删除操作。

名称	状态	EIP释放数量	磁盘释放数量	备注
实例名称	→ 运行中	1	1	--

未释放的弹性公网IP、磁盘会继续收费。

释放云服务器绑定的弹性公网IP地址 删除云服务器挂载的数据盘

本次随实例释放的弹性公网IP共1个，磁盘共1个（快照随对应的磁盘同时删除）。

是 否

---结束

删除边缘设备

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 选择左侧导航栏“边缘节点 > 节点管理”进入页面，单击节点名称进入节点详情页。
- 步骤3** 根据设备名称和设备标识码选择设备删除，如iot_edge_test_0319。
- 步骤4** 退出节点详情页，选择左侧导航栏“边缘节点 > 设备建模”进入页面。
- 步骤5** 根据产品名称和所属资源空间选择产品删除，如iot_edge。

请勿删除*opcua_server、*modbus_server和*edge_node三个产品，否则对应资源空间再次在IoT边缘注册节点时会报错。

----结束

删除边缘节点

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 选择左侧导航栏“边缘节点 > 节点管理”进入页面。
- 步骤3** 根据节点名称选择，如TestEdge01，单击右边操作列的“删除”，并在弹出的窗口中单击“确认”。

删除节点需要一个过程，不超出1分钟，请耐心等待。



----结束

删除虚拟私有云

- 步骤1** 访问[虚拟私有云 VPC](#)，单击“访问控制台”进入网络控制台。
- 步骤2** 选择左侧导航栏“子网”，删除子网，如subnet-iot，请根据节点名称选择，单击操作列的“删除”，并在弹出的窗口中单击“是”。
- 步骤3** 选择左侧导航栏“虚拟私有云”，删除虚拟私有云，如vpc-iot，请根据节点名称选择，单击操作列的“删除”，并在弹出的窗口中单击“确定”。

----结束

5 基于 IoT 边缘服务实现边缘插件的生态拓展

5.1 实验简介

本实验将指导您使用边缘ModuleSDK开发一个数据处理插件以使您的设备发送的数据在IoT边缘预先处理后再上报数据到IoTDA。

通过本实验，您将能够：

- 了解并掌握IoT边缘ModuleSDK的使用。
- 掌握在边缘处理设备发送的数据。

5.2 前提条件

5.2.1 环境要求

安装边缘节点的服务器需要与设备在同一网域，因此您可以

- 局域网：使用两台电脑（一台Windows，一台Linux），或者在一台Windows电脑上使用虚拟化软件安装Linux（网卡需要设置为桥接模式）。
- 公网：分别使用两台拥有公网的Window和Linux电脑，推荐购买两台ECS服务器。

其他要求：

- 安装边缘节点的电脑需为Linux，推荐使用CentOS 7（不推荐CentOS 8，因CentOS 8缺少边缘软件依赖组件）。
- 安装边缘节点的电脑需要安装docker且内存不小于1G，docker要求版本大于17.06.0，推荐18.06.3，不推荐18.09.0（该版本有多处bug）。
- 开放7883端口。

5.2.2 下载 SDK

SDK下载地址：[ModuleSDK_Java](#)。

5.2.3 开发插件

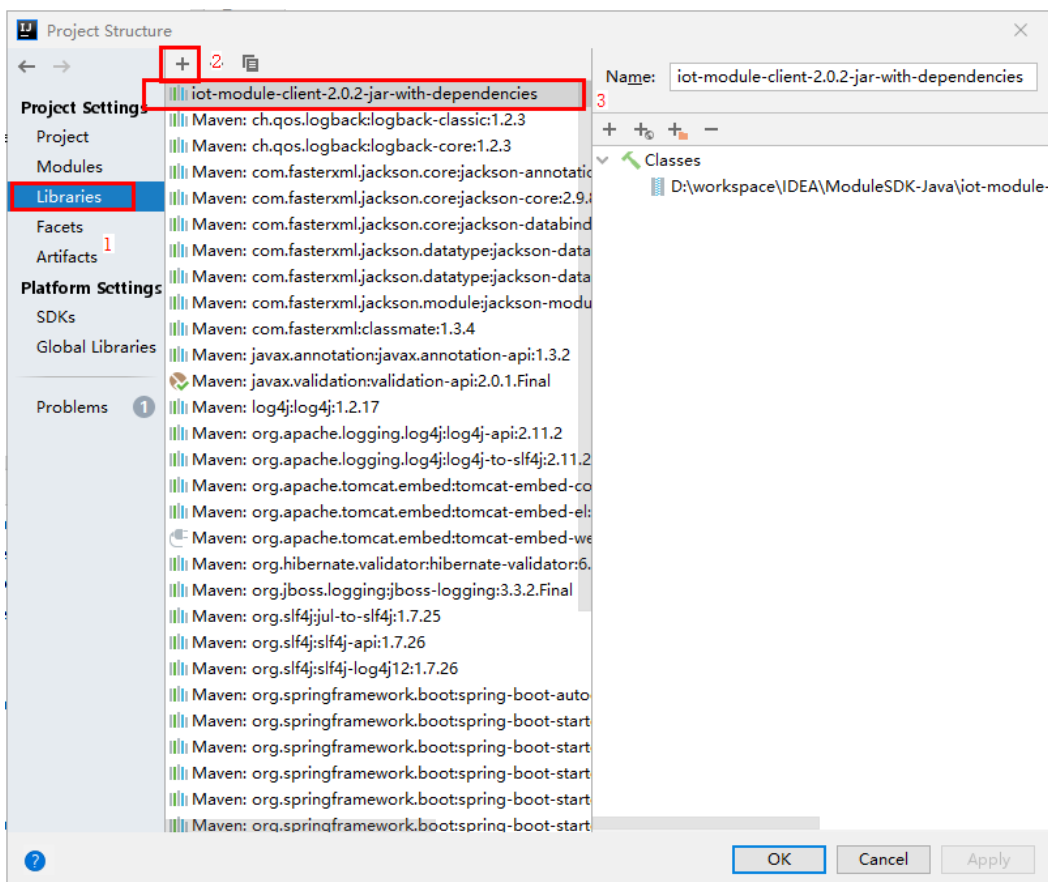
根据您的需要开发插件，提供了一个Demo工程来协助您更快的了解SDK。

步骤1 下载Demo

下载地址：[使用ModulSDK开发插件Demo](#)。

步骤2 导入ModuleSDK依赖包

选择项目名右键单击Open Project Structure，选择Libraries后，单击中间框上方“+”号，选择下载好的ModuleSDK，添加到工程模块。



说明

该Demo包含三个模块：

- erp-integration：此模块主要演示It集成服务。
- modbus-driver：此模块主要演示协议转换。
- monitor-app：此模块主要演示数据处理。

以其中的数据处理样例即monitor-app模块为例演示如何开发一个插件并远程部署到边缘上进行数据处理。

如果您有兴趣，可以参考[开发指南-插件侧开发](#)，该文档详细讲述了实践的架构和业务。

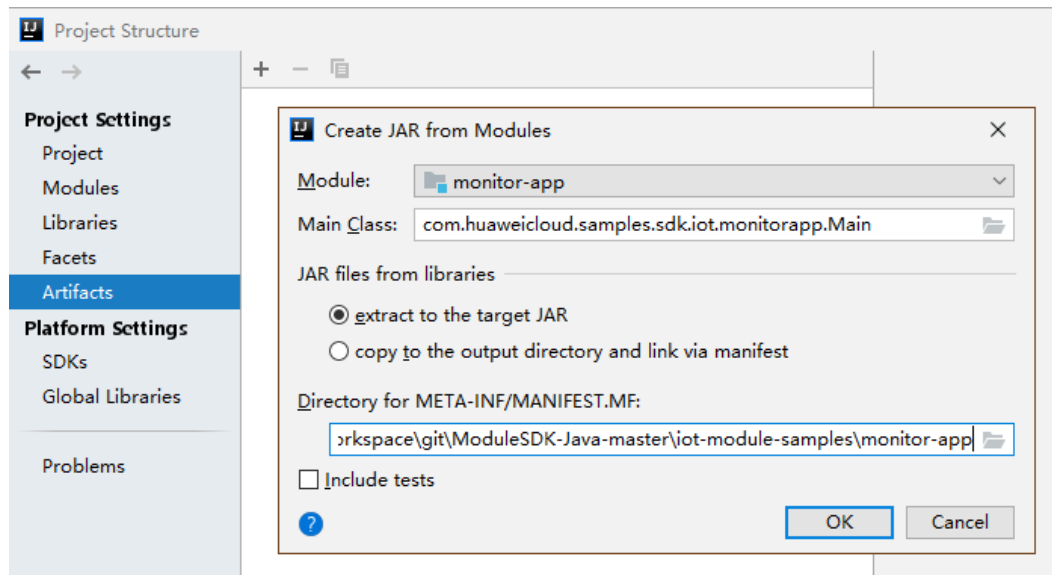
----结束

5.2.4 打包工程

根据您的需求进行相关代码的开发，并将项目打包，以编辑器IDEA为例：

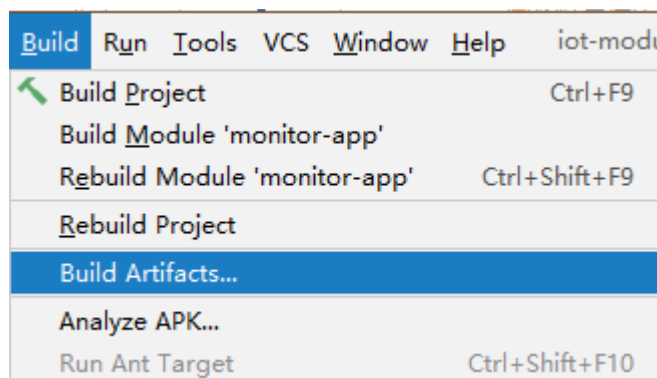
步骤1 选中项目->右键open Module setting

步骤2 Artifacts->单击+号>JAR->From modules with dependencies->模块选择monitor-app, 选择monitorapp的Main入口类,注意MANIFEST.MF位置选择模块根目录->单击apply。



步骤3 单击上方build选项->选择build Artifacts->monitor-app:jar->build

步骤4 打包完成得到monitor-app.jar文件。(如运行遇到错误(Invalid signature file digest for Manifest main attributes)请使用压缩软件进入打开jar包META-INF目录删除.rsa和.mf文件，或者配置maven排除.RSA和.sf文件)



📖 说明

您也可以借助Maven工具进行项目打包
Maven打包命令 `mvn clean compile package`

----结束

5.2.5 打包容器镜像

⚠ 注意

若制作镜像包以容器化方式部署应用，不支持在一个容器内运行多个集成ModuleSDK的软件进程或者重启集成ModuleSDK的软件进程，会导致鉴权失败等问题。

步骤1 上传打包的项目。

将jar文件上传到联网的linux机器上，如目录 (/home/monitor) 中

步骤2 安装docker。

请确认你使用的系统已经安装docker(docker版本需要高于17.06，推荐18.06),安装参照[docker 安装教程](#)。

步骤3 制作镜像。

1. 搜索基础镜像，基础镜像需要集成jre。

```
docker search jre8
```

2. 选择合适的镜像（镜像需要集成版本不低于8的jre）

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
livingobjects/jre8	Jre8 image	4	[OK]	

📖 说明

livingobjects/jre8镜像是docker hub第三方提供的镜像，非华为公司发布，且华为公司未提供任何官方镜像。该镜像在此仅做示例，华为对该镜像的安全性不作保证。强烈建议用户自己封装镜像！

3. 拉取镜像

```
docker pull livingobjects/jre8
```

4. 编写dockerfile制作镜像

dockerfile内容参照如下（具体可参考[编写高效的Dockerfile](#)）

```
#Version 1.0.0
#基础镜像来源
FROM livingobjects/jre8
#授权
RUN mkdir -p /opt/iot/edge/monitor / && chmod -R 777 /opt/
#复制文件到指定目录,此dockerfile文件位置:/home,jar包位置:/home/monitor
COPY monitor /opt/iot/edge/monitor
#用户
USER root
#开放端口
EXPOSE 8080
#运行命令
CMD ["java", "-jar", "/opt/iot/edge/monitor/monitor-app.jar", "run"]
```

目录树结构如下：

```
home
├── dockerfile
├── monitor
└── monitor-app.jar
```

5. 构建镜像

```
docker build -t edge_monitor:1.0.0 /home --no-cache
```

6. 查看打包完成的镜像

```
docker images
```

回显信息:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
edge_monitor	1.0.0	93f9d964bcea	12 seconds ago	243MB

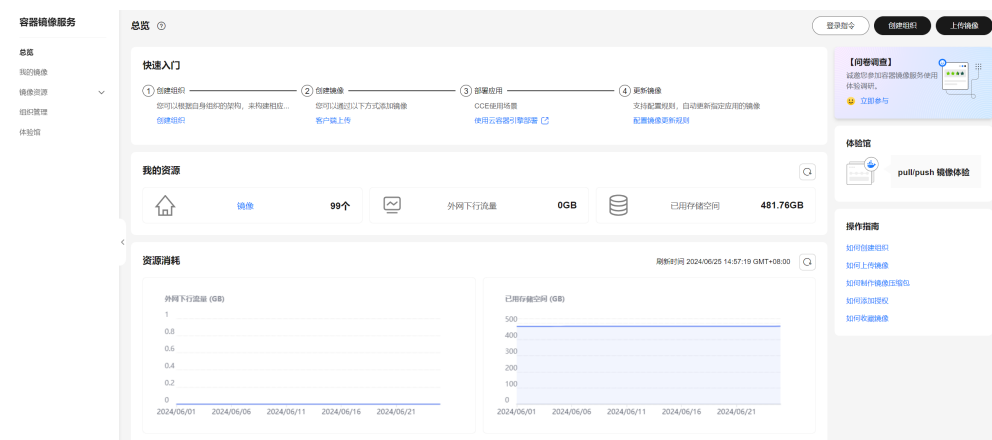
----结束

5.2.6 上传镜像

镜像上传需要使用SWR镜像容器服务，开通及使用请参考[容器镜像服务 SWR_用户指南](#)。

步骤1 上传镜像

访问[SWR容器镜像服务](#)，在左侧导航栏选择“总览”，然后单击右上方“创建组织”。



创建名称为hw_swr的组织

步骤2 获取SWR登录指令

访问[SWR容器镜像服务](#)，在左侧导航栏选择“总览”，然后单击右上角“登录指令”。



复制登录指令到您推送镜像的本地机器上执行

```
[root@ecs-eia2 ~]# docker login -u cn-north-4@VSMNV5R0KW30IRQ6UI98 -p 32c07e71f554201717e396034b1a7e4f6cc3921c6210a340690c26ffe8840969 swr.cn-north-4.myhuaweicloud.com
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[root@ecs-eia2 ~]#
```

步骤3 tag镜像

使用tag命令将镜像重命名以确定上传位置和组织，获取上传位置和组织前缀请参考[pull/push命令](#)

```
docker tag edge_monitor:1.0.0 swr.cn-north-4.myhuaweicloud.com/hw_swr/edge_monitor:1.0.0
```

步骤4 上传镜像

```
docker push swr.cn-north-4.myhuaweicloud.com/hw_swr/edge_monitor:1.0.0
```

步骤5 查看上传结果

镜像名称	所属组织	版本号	更新时间	操作
edge_monitor	hw_swr	1	2021/05/07 22:11:25 GM...	镜像自动同步

注意

上传镜像后请在SWR将镜像设置为公开。

----结束

5.3 操作步骤

5.3.1 注册边缘节点

注册专业版节点

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 选择左侧导航栏“边缘节点 > 节点管理”单击右上角“注册节点”。
- 步骤3** 根据页面提示填写参数，勾选右下角“《华为云服务等级协议》”单击“下一步”配置节点数据。

图 5-1 专业版

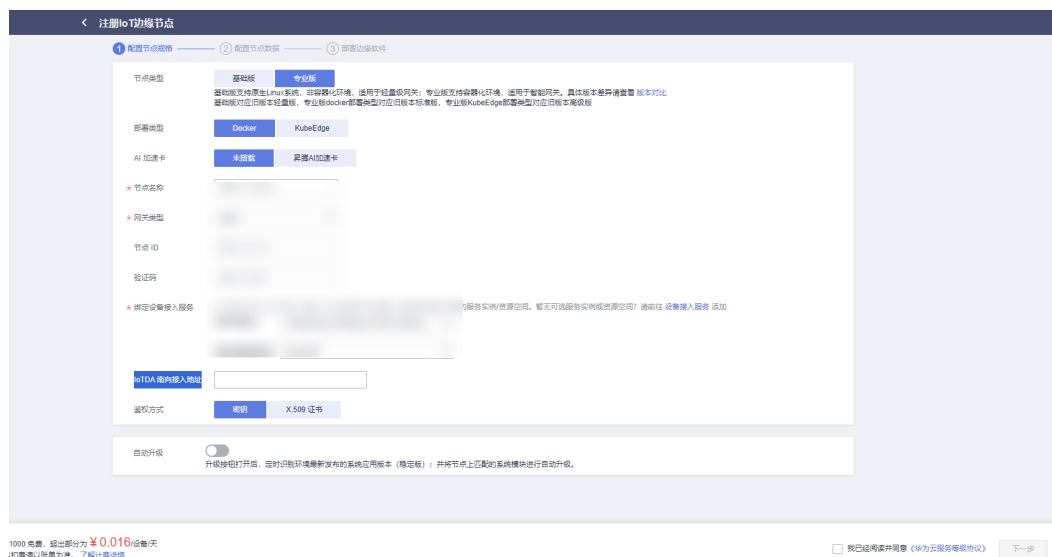


表 5-1 节点规格参数说明

参数名称	参数说明
节点类型	专业版。
部署类型	Docker。
AI加速卡	<ul style="list-style-type: none"> 未搭载：边缘节点未使用AI加速卡时选择。 昇腾AI加速卡：支持华为昇腾AI加速设备，边缘节点的服务器搭载相关昇腾AI加速设备时选择。
昇腾驱动地址	可选，AI加速卡选择昇腾AI加速卡时需要保证安装边缘节点的服务器上已经安装昇腾设备驱动，并根据前端提示填入驱动地址。若改值为空，默认填入Atlas300驱动默认安装地址： /usr/local/Ascend/driver/lib64/。
节点名称	自定义节点名称，如“test”。 说明 节点名称可以在：“节点概览”-“基本信息”中编辑修改。
网关类型	other。
节点ID	自定义（可选填写）。只允许中文，英文字母、数字、中划线（-），下划线（_），最小长度1，最大长度64。 说明 边缘节点ID自定义(不输入默认生成)。
验证码	输入自定义验证码（可选填写）。验证码，执行安装边缘软件命令时与云端通讯的凭证。
设备归属实例	选择默认归属实例。
设备归属资源空间	选择默认资源空间，或为本实验单独新建一个资源空间，如edge。
IoTDA南向接入地址	选择网关归属的IoTDA实例的南向设备mqtt接入地址，也可以选择自定义的南向接入地址，支持“ip:端口”或“域名:端口”形式。
鉴权方式	密钥：默认生成。 X.509证书：请参考： 基于MQTT.fx的X.509证书接入指导 。 警告 鉴权方式为“密钥”的节点，请勿在IoTDA界面重置节点的密钥，重置密钥会导致网关与IoTDA通信异常，且不可恢复。
指纹	证书文件生成的字符串（可选填写）。只能为40位十六进制字符串或64位六进制字符串。
本地路径	输入证书文件和密钥文件在设备上的存储路径，只允许字母、数字、下划线（_）和斜线（/），以/开头并且以.pem或.crt结尾。
自动升级	可选，开启后节点将自动升级系统应用模块。 注意 升级过程中可能导致节点数据上传中断。

步骤4 填写配置节点数据信息后，单击“立即创建”。

图 5-2 配置节点数据

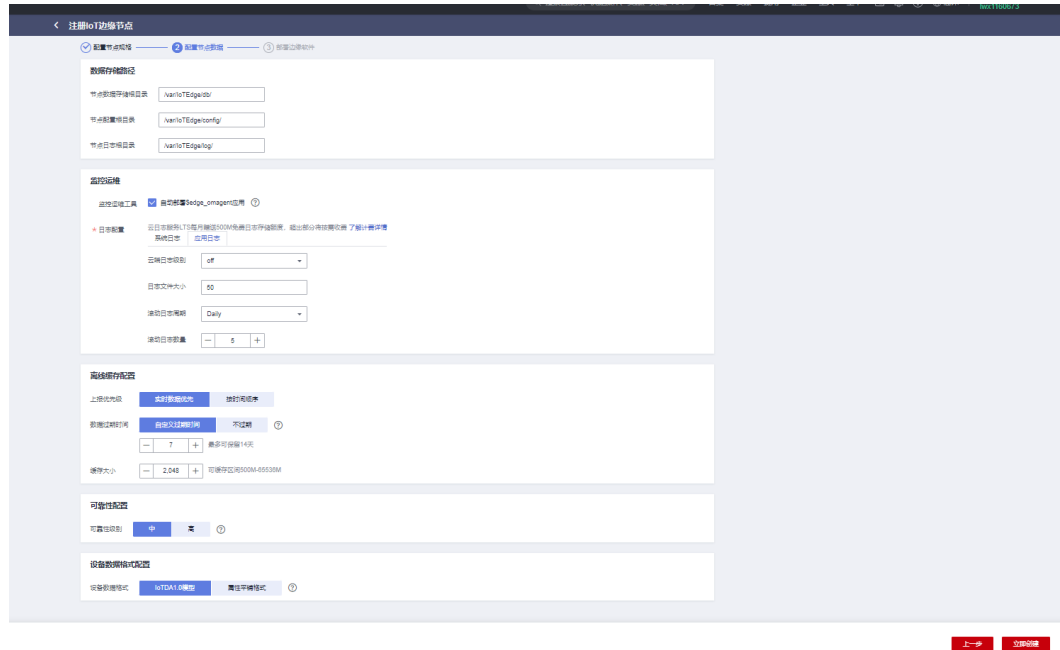


表 5-2 专业版节点信息

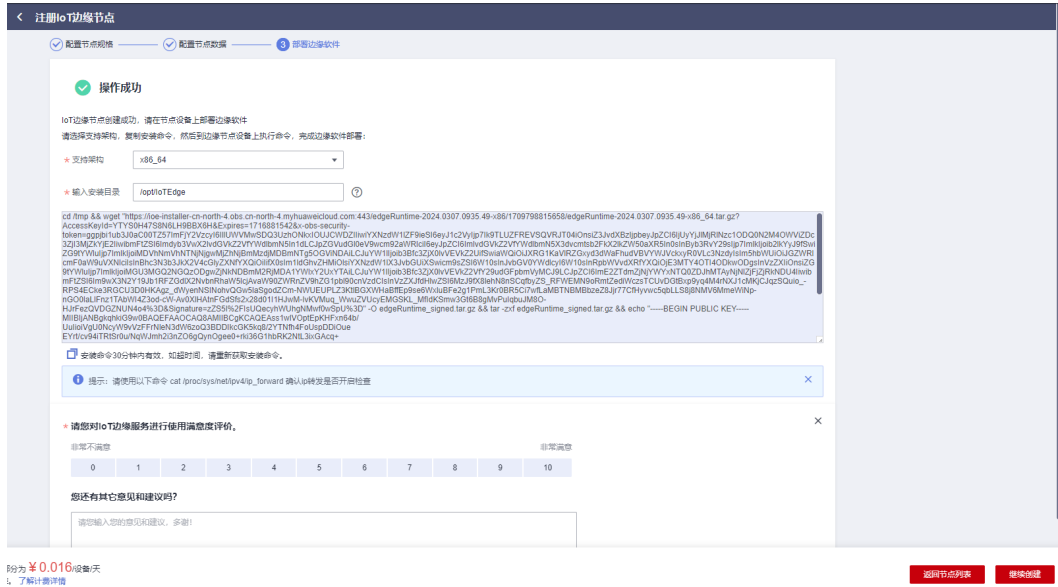
参数名称	说明
数据存储路径	默认（允许修改存储路径）。
监控运维	默认勾选“监控运维工具”。
\$edge_omagent应用	负责边缘节点远程监控和运维，提供日志上报、远程SSH、CPU等指标上报、文件上传下载能力。

参数名称	说明
日志配置	<p>系统日志：边缘节点上已部署的系统应用产生的日志。 应用日志：边缘节点上已部署的用户定义应用所产生的日志。</p> <ul style="list-style-type: none"> • 云端日志级别：on, off, trace, debug, info, warn, error, fatal。其中，on表示上传所有日志至云端，off表示不上传日志至云端，其余可选项代表按日志级别采集，选择后将只采集目标级别及以上的日志（模糊匹配）。日志级别的严重程度等级由低到高依次为：trace, debug, info, warn, error, fatal。 • 日志文件大小：日志文件大小限制，单位MB，默认50，取值范围10-1000。某个日志文件如果达到大小限制，则会将文件转储保存至日志目录下。 • 滚动日志周期：日志转储周期，可选项：daily, weekly, monthly, yearly。某个日志文件如果达到转储周期，则会将文件转储保存至日志目录下。日志文件大小和滚动日志周期是同时生效的，满足任何一个条件都会进行日志转储。 • 滚动日志数量：日志文件转储个数，默认5，取值范围1-10。边缘节点保存的转储日志数量如果达到限制，则会删除最早的转储文件。 <p>注意</p> <ol style="list-style-type: none"> 1. 当前“云端日志级别”对日志等级采取模糊匹配的策略，若日志消息中存在与日志级别标识相同的关键字，则也可能被采集。 2. 若要开启“云端日志级别”中的trace, debug, info, warn, error, fatal选项，则必须满足以下三个前置条件： <ol style="list-style-type: none"> 1. 目标节点是基础版节点或部署类型为Docker的专业版节点。需注意，部署类型为KubeEdge的专业版节点不支持此功能。 2. 已安装\$edge_omagent应用，且版本不低于1-1-15-xxx。 3. 应用日志需遵循如下规范打印，以便进行过滤： <ul style="list-style-type: none"> ○ 日志级别标识需打印为（大小写均可）：trace、debug、info、warn、error、fatal。 ○ 每条日志消息以YYYY-MM-DD日期格式作为前缀。 <p>说明 专业版（部署类型为Docker）和基础版节点只涉及日志级别这一个参数，其他参数只对专业版（部署类型为KubeEdge）有效。</p>
离线缓存配置	<p>可对离线节点上报的数据进行缓存，定义缓存周期（可配置为不过期）跟缓存大小、上报的优先级、路由转发是否开启离线缓存。仅当上报优先级为“按时间顺序”时支持路由转发开启离线缓存。</p>
可靠性配置	<p>可靠性级别分为“中”和“高”2种。中级别：当应用上报属性和点位的频率超过流控值时，超过部分会被丢弃；高级别：当应用上报属性和点位的频率超过流控值时，通过控制模块的发送速度，降低上报频率，报文不会被丢弃。（注意：高可靠性级别的配置需要定制应用的配合，同时会造成性能的损失）。</p>

参数名称	说明
设备数据格式配置	设备的数据格式：即单网关节点中设备的标准数据格式，包含 IoTDA1.0物模型格式和属性平铺格式。IoTDA1.0物模型中包含了设备、服务和属性三层结构，具体参考IoTDA网关子设备属性上报报文格式。属性平铺格式，包含设备、属性2层结构，属性结构可以包含自定义结构的拼接，在json中作为一项平铺展开，具体格式请参考 步骤6设备数据格式配置 。 须知 属性平铺格式功能，需要edge_hub应用1-1-38-xx版本及以上支持

步骤5 弹出如下对话框，您可以“返回节点列表”也可以“继续创建”。

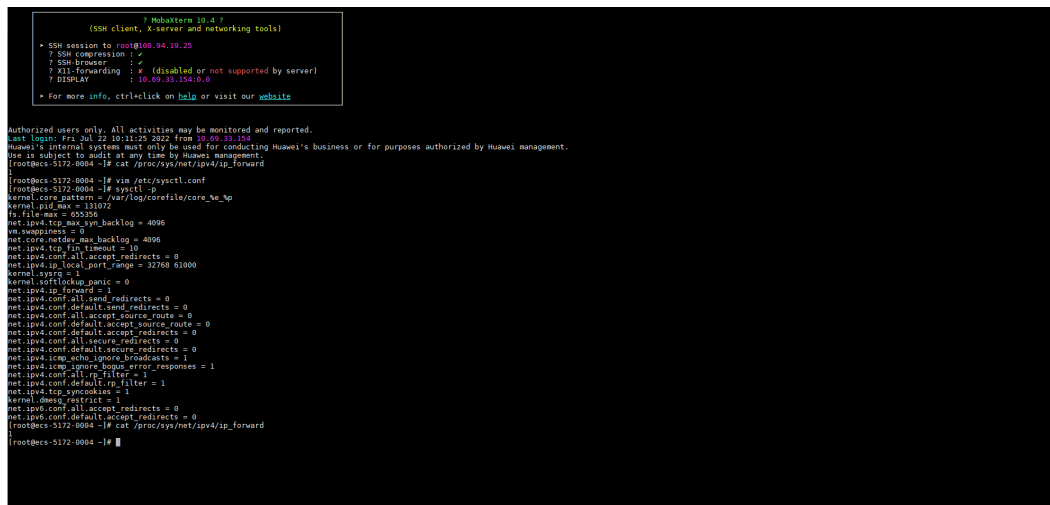
图 5-3 注册成功



说明

安装部署节点前，请确认机器的IP转发已开启并且已开启自启动，以及7883和8943端口开放。其中，IP转发规则配置如下：

图 5-4 IP 转发规则配置



1. 检查机器ip_forward in/proc文件系统的值。
`cat /proc/sys/net/ipv4/ip_forward`
 若值等于0，则需要添加配置IP转发规则。
 2. 检查配置文件/etc/sysctl.conf是否包含net.ipv4.ip_forward = 1。
 如果配置文件中没有net.ipv4.ip_forward参数项，请新增一行net.ipv4.ip_forward = 1。
`vim /etc/sysctl.conf`
 按“i”进入编辑模式，修改完成后按“Esc”退出编辑模式，然后按“Shift” + “;”组合键，输入wq并回车，保存修改并退出。
 3. 重新加载此文件的值。
`sysctl -p`
 4. 检查7883和8943端口是否开放，如果是华为云服务器，请参考：[安全组](#)。
- 结束

安装专业版节点

步骤1 在“节点管理”的节点列表选择您需要安装的节点，单击右侧“安装”，获得提示。

图 5-5 安装

节点名称	节点类型	资源包类型	资源包ID	状态	主机名/网络	操作
	专业版	--	--	未安装		删除 安装
	专业版	--	--	离线	172.30.7.64,172.17.0.1,1...	删除
	专业版	--	--	已安装		删除


步骤2 选择支持架构和安装目录，然后单击  复制安装命令，使用SSH工具以root用户登录边缘节点服务器后台系统，执行安装命令。

图 5-8 安装命令执行成功

```
2024-05-28T14:44:29,581321 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:PENDING, reason:(zero), ]
2024-05-28T14:44:31,632105 | INFO | saveBasePathInfo | 78 | config base path start.
2024-05-28T14:44:31,632291 | INFO | saveBasePathInfo | 83 | config log path:/var/IoTEdge/log/, db_path:/var/IoTEdge/db/, config_path:/var/IoTEdge/config/
2024-05-28T14:44:31,632457 | INFO | saveBasePathInfo | 95 | config base path success.
2024-05-28T14:44:31,632535 | INFO | addAgentCustomConfigs | 201 | addCustomConfigs start.
2024-05-28T14:44:31,632603 | INFO | addAgentCustomConfigs | 205 | addCustomConfigs end.
2024-05-28T14:44:31,632654 | INFO | pullImage | 319 | start pull image, deploy type is cloud
2024-05-28T14:44:35,888519 | INFO | createModule | 76 | create Module, name:sys_edge_agent
2024-05-28T14:44:35,939013 | INFO | createModule | 84 | module created, CreateResp[ moduleId:sys_edge_agent, moduleName:sys-edge-agent, appId:sedge_agent, appVersion:test-tpm-2-x86, containerId:4b3e43cee7eac9bc282b994377abfb48a284389a36856f3cacc0338743036e, processId:(zero), type:docker, ]
2024-05-28T14:44:35,939163 | INFO | startModule | 160 | startModule moduleId:sys_edge_agent
2024-05-28T14:44:42,390394 | INFO | listModule | 259 | listModule
2024-05-28T14:44:42,440775 | INFO | checkModulesDo | 340 | start check modules, time:1
2024-05-28T14:44:42,440908 | INFO | listModule | 259 | listModule
2024-05-28T14:44:42,491163 | INFO | printModulesAndReport | 404 | ***success:1 failed:0
2024-05-28T14:44:42,491339 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:RUNNING, reason:(zero), ]
2024-05-28T14:44:42,542192 | INFO | reportEvent | 158 | Begin reporting event: {"events":[{"resource":"node.installation","location":"zzzzz","event":"succeeded","event_severity":"Info","event_time":"20240528T064442Z","notify_data":{"host_tag":"DEFAULT","edge_node_id":"zzzzz"}}]}
2024-05-28T14:44:42,592933 | INFO | reportModuleStatus | 138 | reportModuleStatus, modulesInfo:ModuleStatusInfo[! host_tag:DEFAULT, deploy_id:1716878669308, modules:[ModuleStatusInfo[ module_id:sys_edge_agent, app_id:sedge_agent, app_version:test-tpm-2-x86, container_state:RUNNING, liveness_state:(zero), update_time:1716878669305, ]], ]
2024-05-28T14:44:43,043689 | INFO | checkBridgeNetwork | 560 | begin checking bridge network
2024-05-28T14:44:39,524137 | INFO | install | 494 | install edge finish
2024-05-28T14:44:30,524559 | INFO | stop | 545 | edgeinstatter stop.
/root
```

步骤3 单击“我知道了”，等待边缘节点的状态变为“在线”，表示该节点已安装并连接成功。

步骤4 您可以单击“节点名称”，查看该节点详细信息，关于节点详情描述可参见节点概览。

----结束

5.3.2 添加插件应用

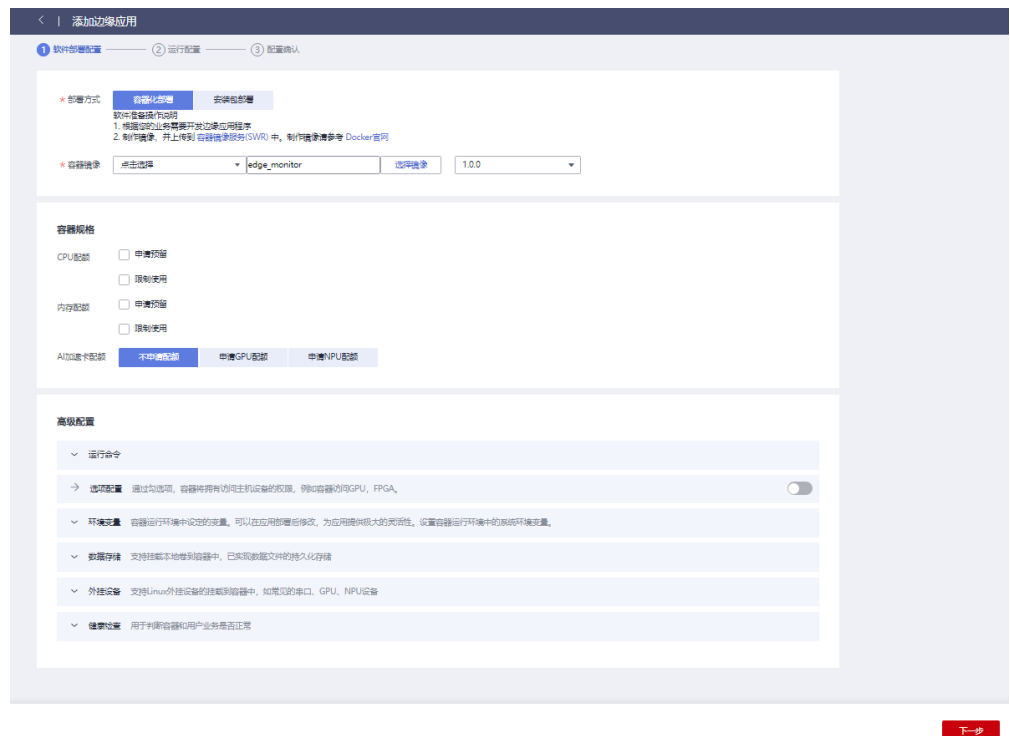
步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。

步骤2 选择左侧导航栏“边缘节点 > 应用管理”进入页面。

步骤3 单击右上角的“添加应用”，按照页面提示填写信息。

1. 软件部署配置

选择上传到SWR服务的镜像，如未发现镜像，请检查镜像是否为公开，设置镜像为公开方式：容器镜像服务SWR->我的镜像->单击镜像ID进入详情->右上方编辑。



参数名称	说明
部署方式	选择“容器化部署”。
容器镜像	选择“单击选择”。 选择镜像：单击“选择镜像”按钮后，在右侧展开栏中选择SWR中的自定义容器镜像
容器规格	
CPU配额：	容器规格中CPU/内存配额，可以配置该容器的运行资源限制，即最小最大资源占用限制；此处不勾选则使用默认值。
内存配额：	
AI加速卡配额	不申请配额。
高级配置	
运行命令	启动程序的运行命令。
环境变量	可按需配置环境变量。
数据存储	可将主机某个目录挂载到容器中。
外挂设备	支持Linux外挂设备的挂载到容器中，如常见的串口、GPU、NPU设备。
健康检查	用于判断容器和用户业务是否正常。

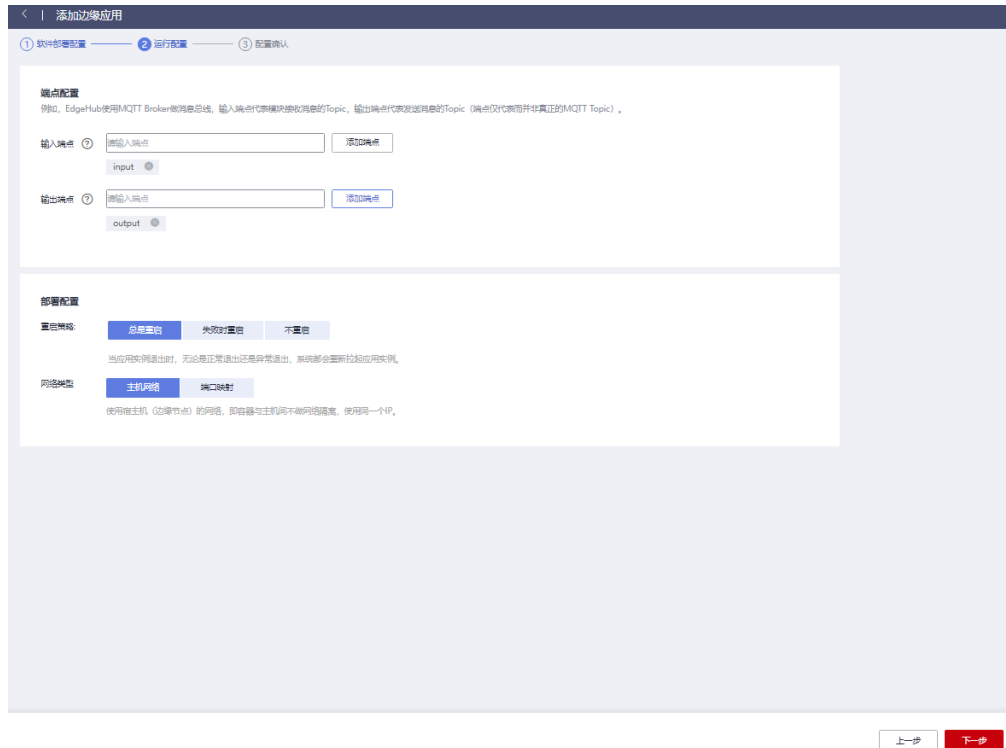
2. 根据需要进行运行配置

输入端点/输出端点与demo中代码定义的端点对应，如monitor-app中输入与输出端点设置为input和output，则配置为：

- 输入端点：input
- 输出端点：output

说明

输入输出端点是必需配置的，当有数据流转时需要配置，如OT应用（数据处理）。
驱动类应用和IT应用一般不需要配置。



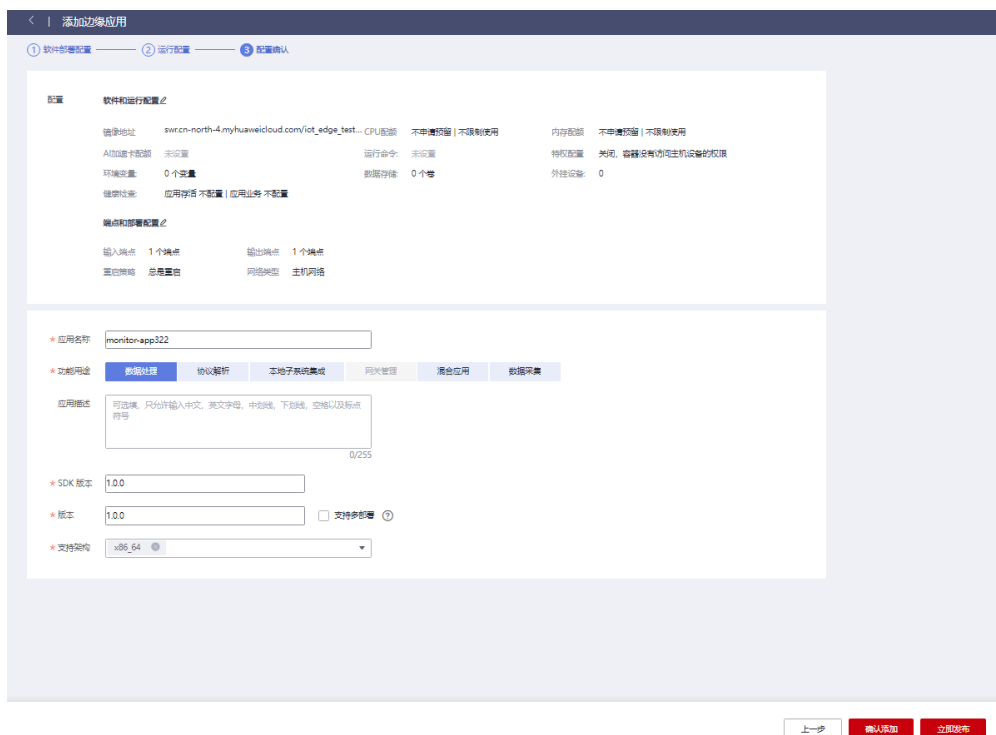
参数名称	说明
端点配置	
输入端点	输入/出端点：若容器应用镜像集成了边缘SDK，并配置了输入/输出端点，则该容器应用可以从该输入端点获得设备数据，进行自定义的数据处理后，从输出端点将数据流回边缘处理中心（系统应用hub）。在“添加应用”时，也需配置输入/输出端点，为后续配置边缘流转规则提供前提条件。
输出端点	
部署配置	
重启策略	选择“总是重启”。
网络类型	选择“主机网络”。

注意

集成ModuleSDK后，在高级配置中必须挂载config卷，主机目录和容器目录均为自定义，可参考下图。模块身份等信息均由SDK存储在config卷中，如果不配置挂载config卷，会导致自定义应用重启后无法与EdgeHub成功建立MQTT连接。



3. 确认配置



参数名称	说明
应用名称	自定义，如iot_edge
功能用途	根据容器应用的功能用途自行选择，本实验采用的镜像是进行数据清洗，故选择“数据处理”。
应用描述	自定义，可不填。
SDK版本	集成边缘SDK的版本号。
版本	选择镜像版本，如“1.0.0”，不勾选“支持多部署”。
支架架构	根据容器应用镜像支持的架构自行选择，本实验采用的镜像只支持x86架构，故选择“x86_64”。

步骤4 单击“确认添加”后，则创建自定义应用成功。



步骤5 单击“返回应用管理列表”可查看到创建的应用，单击右侧的“版本列表 > 发布”。

应用详情



步骤6 返回“应用管理”可查看到创建的应用，单击右侧的“版本列表”可查看已发布的应用版本。



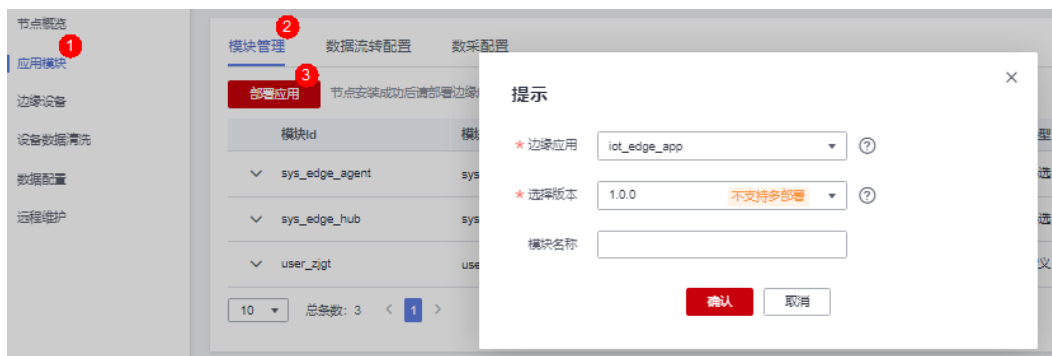
----结束

5.3.3 远程部署

步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。

步骤2 选择左侧导航栏的“边缘节点 > 节点管理”，选择您的边缘节点名称，进入节点详情页。

步骤3 在左侧导航栏选择“应用模块 > 模块管理”页签，单击“部署应用”。



- 边缘应用：选择上述制作的应用名称，如：iot_edge_app
- 选择版本：选择已发布的与边缘节点相同架构的应用版本，如1.0.0
- 模块名称：自定义。

步骤4 单击“确认”。

等待3秒后，刷新应用界面，可看到部署的自定义应用的实例状态为“运行中”，表明该应用容器已运行起来。

模块id	所属应用	版本	应用类型	实例状态	操作
sys_edge_agent	sedge_agent	1-0-12-x86	系统必选	运行中	升级
sys_edge_hub	sedge_hub	1-0-28-x86	系统必选	运行中	升级
user_edge_modbus-server	edge_modbus-server	1.0.6	用户定义	运行中	删除 升级
user_monitor	monitor	1.0.5	用户定义	运行中	删除 升级

部署后应用状态为部署中，等待一段时间后单击刷新，可以看到应用为运行中。

步骤5 查看应用在边缘运行状态

输入命令docker ps

```
[root@ecs-e1a2 ~]# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STA
TUS           PORTS
b25d0869a969   swr.cn-north-4.myhuaweicloud.com/huawei-iot-edge-node/edge_monitor_x86:1.0.5   user-monitor           6 minutes ago Up
6 minutes
07b1c87074ff   swr.cn-north-4.myhuaweicloud.com/iotedge/edge_modbus-driver:1.0.6           user-edge-modbus-server About an hour ago Up
About an hour
38bfc6a610b8   swr.cn-north-4.myhuaweicloud.com/huawei-iot-edge/edgehub:2021.0705.1003.43    "/opt/iot/edge_hub/s..." 39 hours ago Up
39 hours
c580b587e119   swr.cn-north-4.myhuaweicloud.com/huawei-iot-edge/edgeagent_x86_64:0.3.072-20210709163857 "/bin/sh -c 'sh /opt..." 39 hours ago Up
39 hours
```

可以看到插件应用已经在边缘运行了。

步骤6 部署应用后还需要配置数据流转规则，以使您的应用能够接受和发送数据

规则名称	消息来源	消息目标	操作
monitor_input	设备端	user_monitor-app / input	删除
monitor_output	user_monitor-app / output	云端	删除

----结束

5.3.4 接入模拟设备

创建产品

步骤1 选择左侧导航栏“边缘节点 > 设备建模”进入页面，单击右上角“创建产品”。

The screenshot shows the 'IoT Edge' management console. On the left is a navigation menu with '设备建模' (Device Modeling) selected. The main area is titled '产品' (Product) and contains a '功能简介' (Feature Introduction) section with a diagram showing the process from '定义Profile' (Define Profile) to '注册设备' (Register Device). Below this is a table listing created products.

产品名称	产品ID	资源空间	设备类型	协议类型	操作
				MQTT	详情 删除
				MQTT	详情 删除
				MQTT	详情 删除

步骤2 创建产品

创建产品

★ 所属资源空间 ?

★ 产品名称

协议类型

★ 数据格式 ?

厂商名称

★ 设备类型 ?

- 所属资源空间：与节点所属资源空间一致
- 产品名称：自定义
- 协议类型：MQTT
- 数据格式：JSON
- 厂商名称：自定义
- 设备类型：MQTT_Device

步骤3 在单击“产品名称”下的产品hw_iotedge_mqtt进入产品详情页面。

步骤4 在“模型定义”页签下，单击“自定义模型”。



步骤5 填写服务ID、服务类型，单击“确认”。

新增服务 ×

* 服务ID

* 服务类型

服务描述

- 服务ID: power
- 服务类型: power

步骤6 新增属性

新增属性 ×

* 属性名称 必选

属性描述

* 数据类型

* 访问权限

* 长度

枚举值

- 属性名称: status
- 属性描述: 自定义
- 数据类型: string
- 访问权限: 可读可写
- 长度: 10
- 枚举值: normal, error

步骤7 新增命令



- 命令名称：power_control
- 下发参数
- 参数名称：restart
- 数据类型：string
- 描述：自定义

----结束

添加边缘设备

步骤1 单击“边缘节点 > 节点管理 > 节点名称”进入节点详情页，然后单击“边缘设备 > 添加边缘设备”。

图 5-9 添加边缘设备



步骤2 输入设备信息。

提示 ×

接入设备需统一纳入物联网平台管理, 并和当前边缘节点归属同一个设备接入服务实例/资源空间。
归属服务实例 IoTDA默认基础版
归属资源空间 DefaultApp_hwstaff_pub_IoTEdge_iot

* 所属产品 MQTT C
没有可选产品? 请前往设备接入服务增加自定义产品, 并定义设备功能前往添加产品

* 设备标识码

* 设备名称 ?

模块 ID ?

password

确认 取消

- 所属产品: 即前面创建的产品
- 设备标识码: 自定义
- 设备名称: 自定义
- password: 自定义

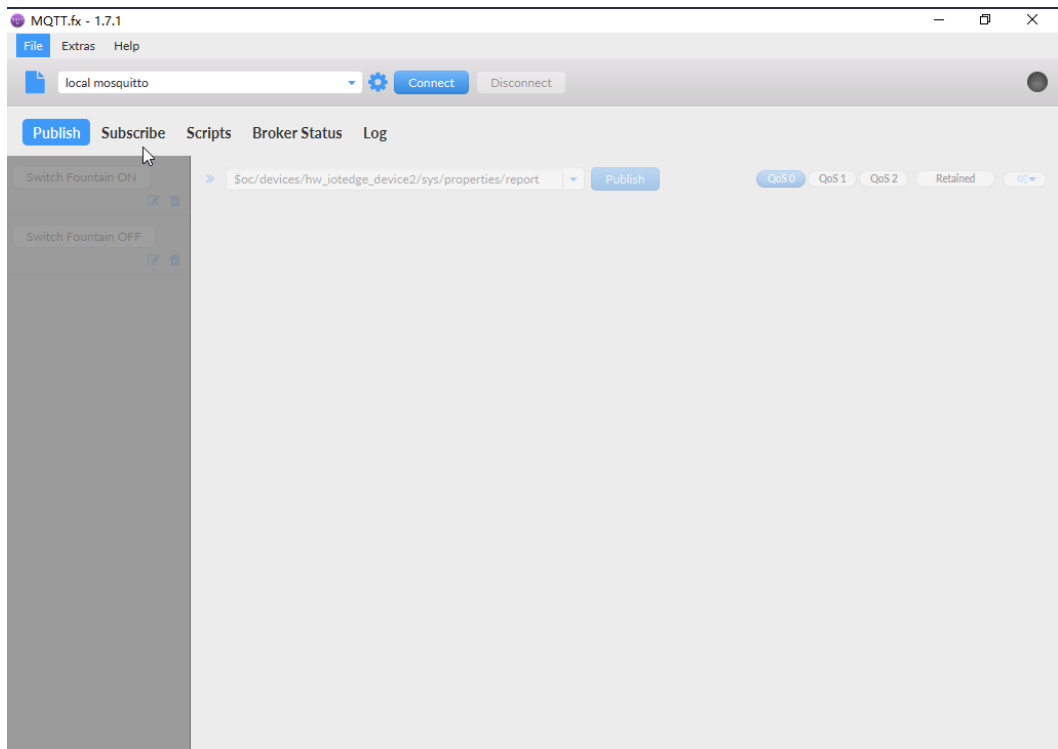
----结束

接入设备

使用MQTT.fx模拟设备接入。

步骤1 下载MQTT.fx及证书, [证书下载地址](#)。

安装完成后打开, MQTT.fx软件界面如下:

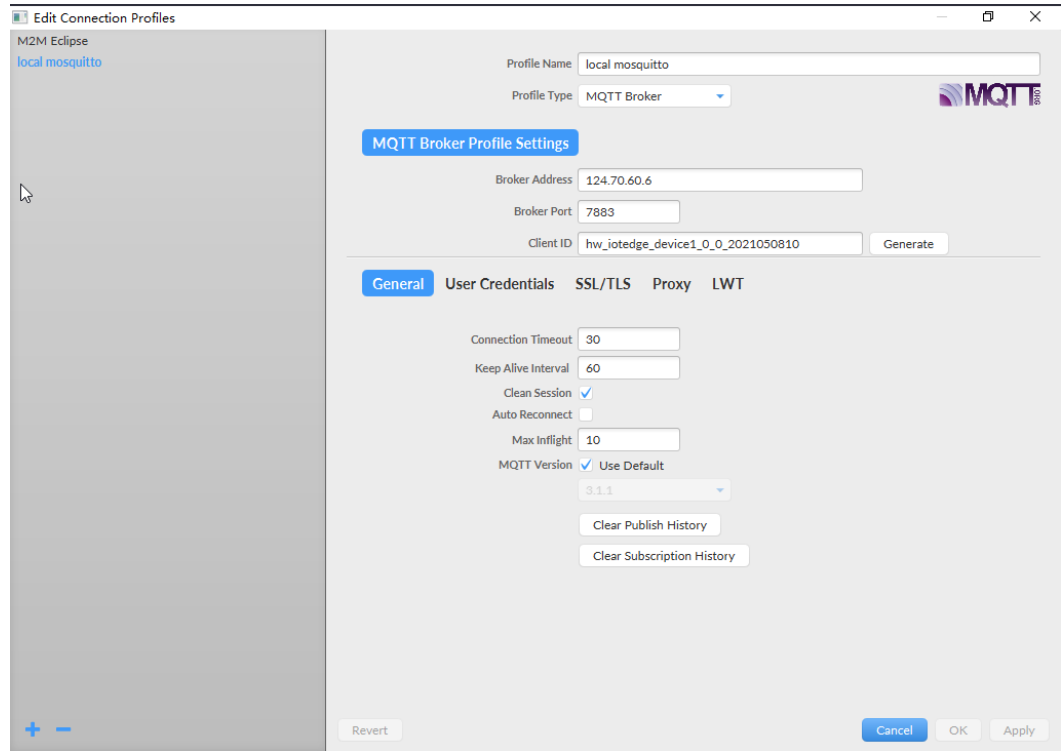


注意

Connect左边的蓝色齿轮为设置。

Publish是消息发送，Subscribe为消息接受，Log可查看日志。

步骤2 单击设置-General，输入以下信息

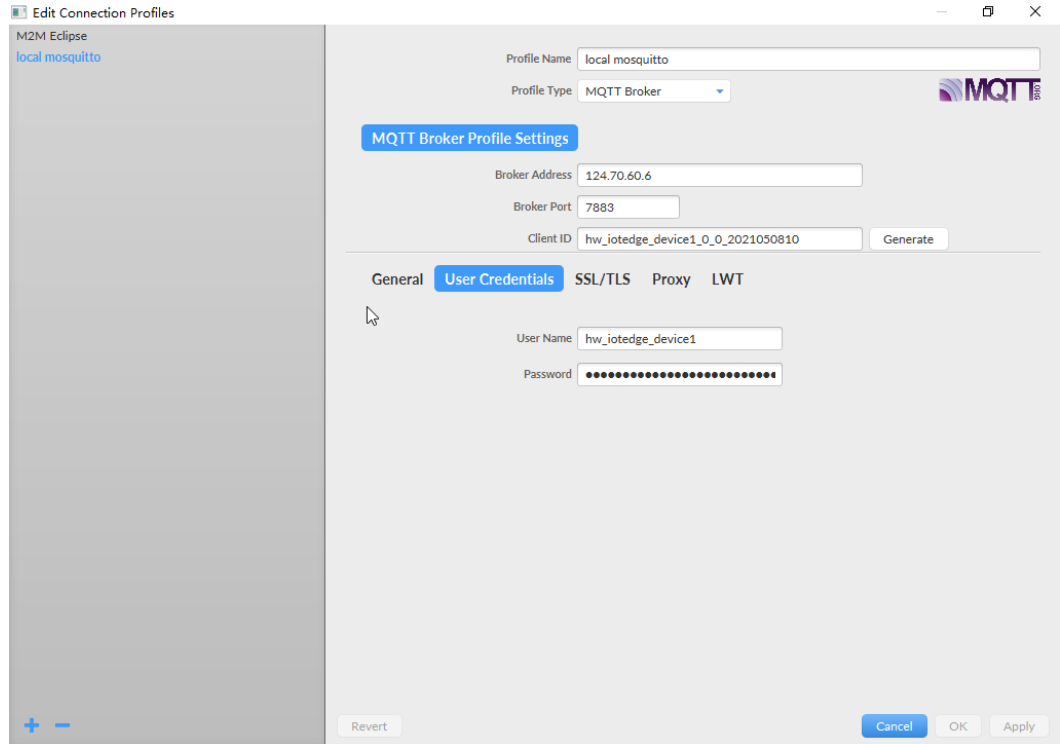


- Broker Address：输入节点的公网地址。
- Broker Port：MQTT(S)协议使用的端口，默认为7883。
- 尝试连接时间和保持连接时间等自定义。

步骤3 单击设置-User Credentials

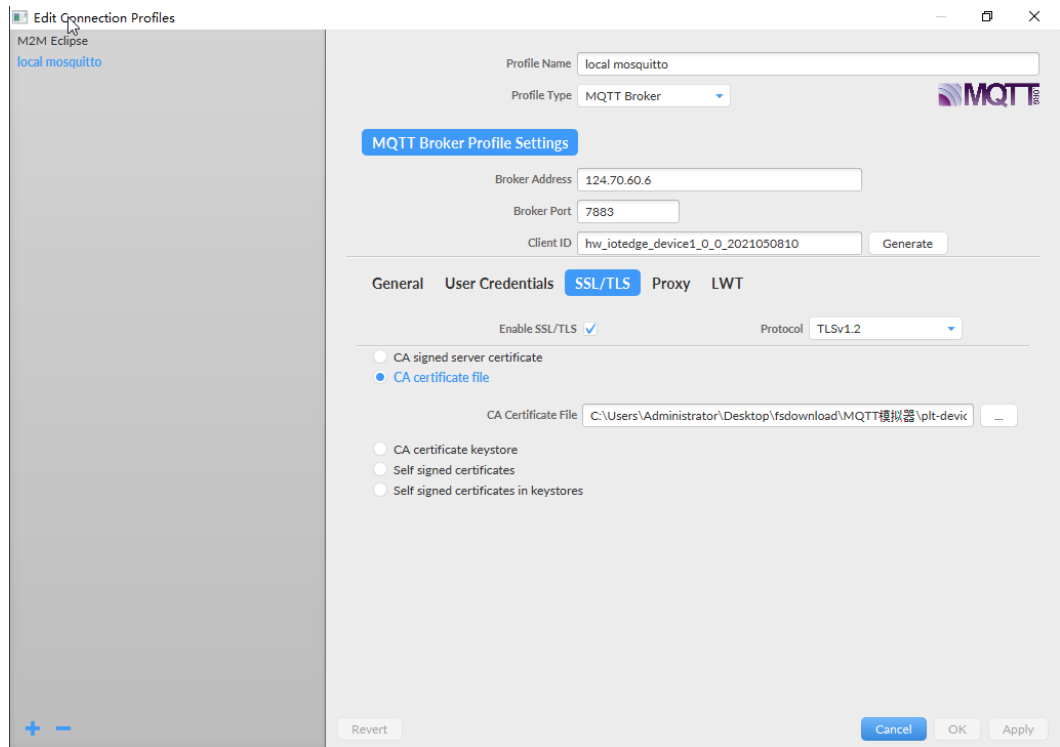
Client ID和密码需要工具进行转换。利用[网页转换工具](#)进行转换。

填写[添加边缘设备](#)后生成的设备ID和设备密钥，生成连接信息（ClientId、Username、Password）。



步骤4 单击设置-SSL/TLS

勾选Enable SSL/TLS，单击CA certificate，选择下载的证书文件。



单击Apply应用设置后返回。

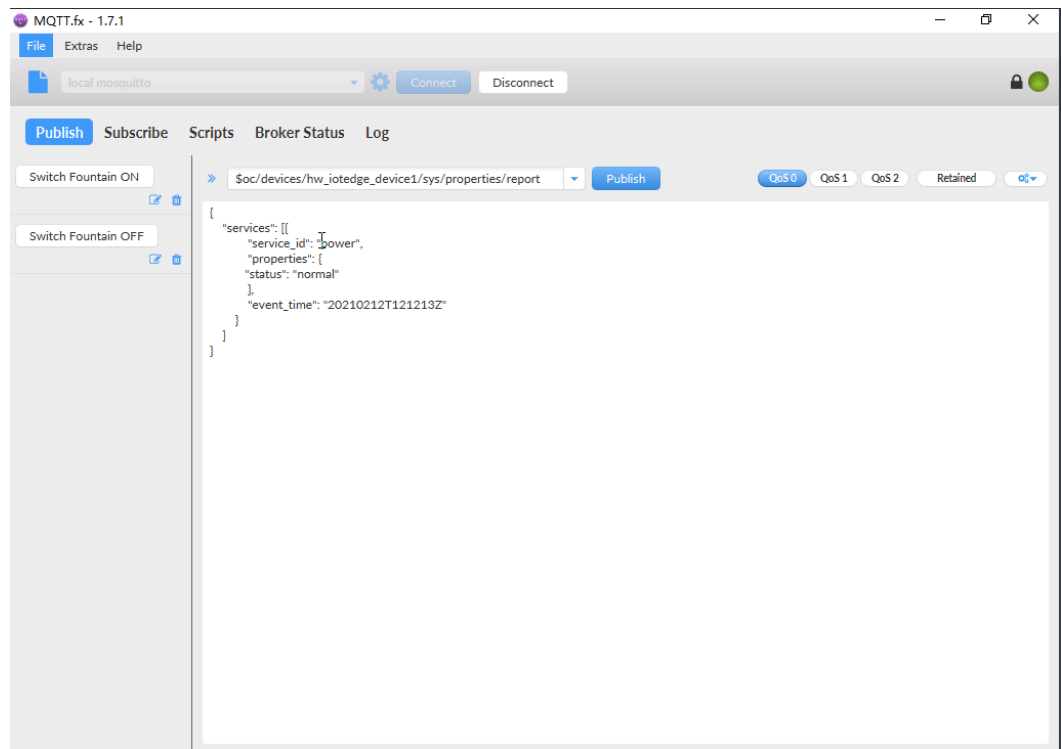
单击Connect连接，连接成功后右边红点会变成绿色，IoTDA也会显示设备在线。

步骤5 选择publish输入topic地址。

Topic: \$oc/devices/hwiotedgedevice2/sys/properties/report

其中，hwiotedgedevice2为设备ID，请替换为实际值，可在IoTDA->设备->所有设备列表中查看。消息体输入：

```
{
  "services": [{
    "service_id": "power",
    "properties": {
      "status": "error"
    },
    "event_time": "20210508T173342Z"
  } ]
}
```

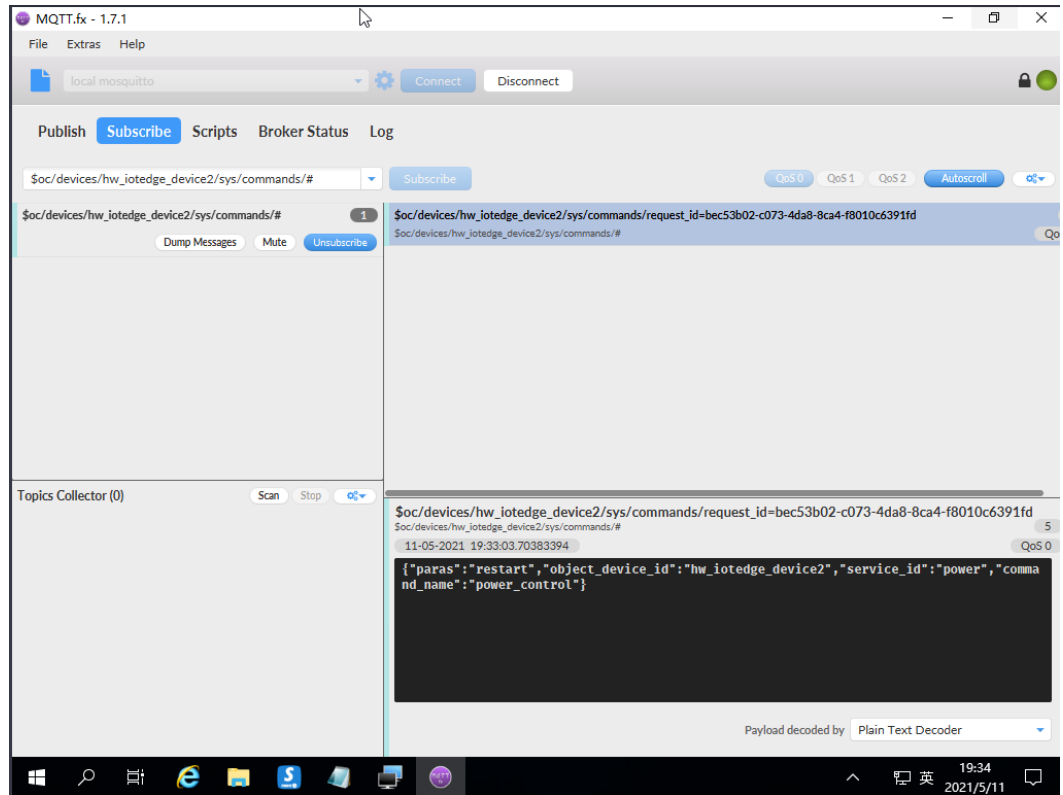


步骤6 输入订阅的topic，可在IoTDA->产品->topic管理中查看。

Topic: \$oc/devices/hw_iotedged_device2/sys/commands/#

其中，hw_iotedged_device2为设备ID，请替换为实际值，设备ID请进入设备详情查看。

返回到publish，单击publish按钮后进入Subscribe，可以看到订阅命令收到一条command。



步骤7 进入边缘设备查看数据上报情况

进入IoTDA单击设备，进入概览发现并无数据上报，说明设备发送的数据在节点本地被集成SDK的monitor-app应用拦截，当数据内容为error时，在本地向hub调用重启命令。应用实现了数据处理和命令下发的功能。



----结束

6 基于 IoT 边缘实现 OT 数采 (OPCUA 协议)

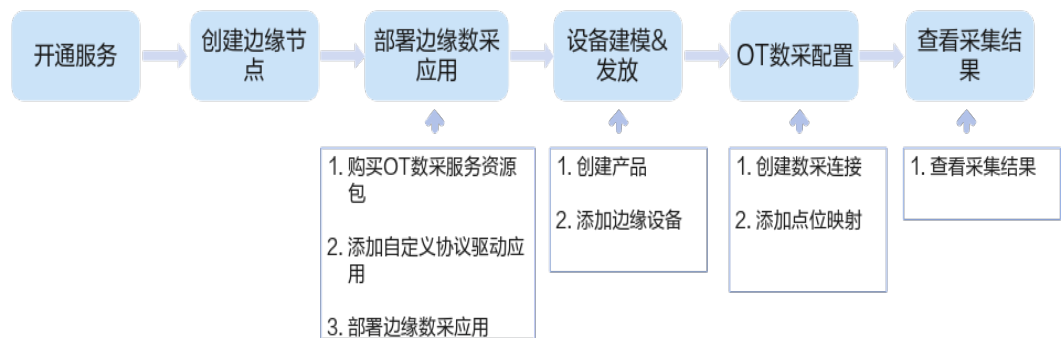
6.1 实验简介

OT数采配置，提供了在IoT边缘控制台简单配置，实现OT数采（DCS、PLC、OPCUA等）。

通过本实验，您将能够：

- 了解并掌握IoT边缘服务的功能特点与使用方法。
- 掌握将OT设备的数据上报云端的方法。

OT 数采流程介绍



6.2 操作步骤

6.2.1 创建边缘节点

前提条件

- 已注册[华为云](#)账号，且已完成实名认证。
- 已开通[IoT边缘服务](#)和[IoTDA设备接入服务](#)。
- 在创建边缘节点前，检查配置要符合[硬件规格](#)要求，如磁盘、内存和docker版本，切换root用户执行安装命令。

注册专业版节点

- 步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2 选择左侧导航栏“边缘节点 > 节点管理”单击右上角“注册节点”。
- 步骤3 根据页面提示填写参数，勾选右下角“《华为云服务等级协议》”单击“下一步”配置节点数据。

图 6-1 专业版

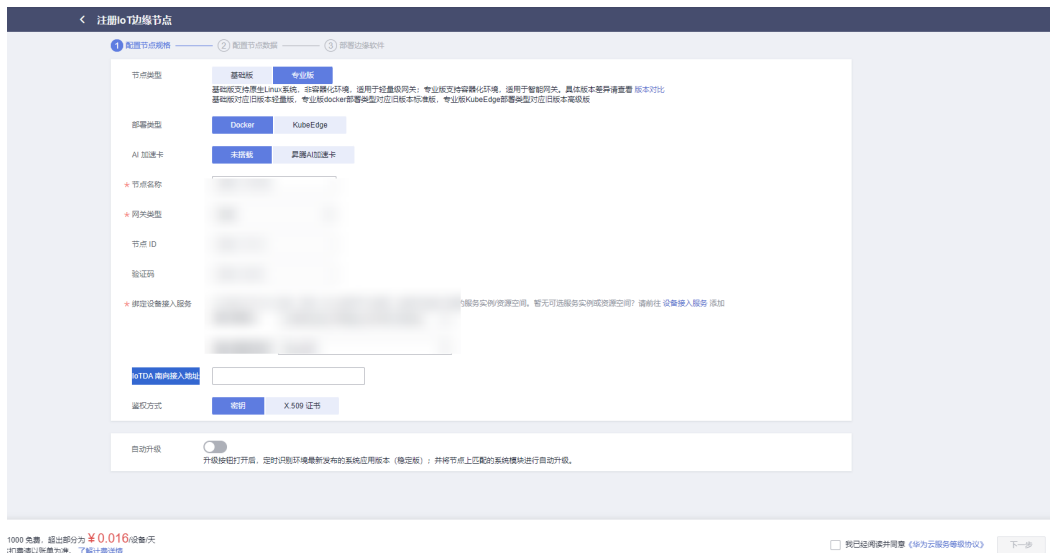


表 6-1 节点规格参数说明

参数名称	参数说明
节点类型	专业版。
部署类型	Docker。
AI加速卡	<ul style="list-style-type: none"> 未搭载：边缘节点未使用AI加速卡时选择。 昇腾AI加速卡：支持华为昇腾AI加速设备，边缘节点的服务器搭载相关昇腾AI加速设备时选择。
昇腾驱动地址	可选，AI加速卡选择昇腾AI加速卡时需要保证安装边缘节点的服务器上已经安装昇腾设备驱动，并根据前端提示填入驱动地址。若改值为空，默认填入Atlas300驱动默认安装地址： /usr/local/Ascend/driver/lib64/。
节点名称	自定义节点名称，如“test”。 说明 节点名称可以在：“节点概览”-“基本信息”中编辑修改。
网关类型	other。

参数名称	参数说明
节点ID	自定义 (可选填写)。只允许中文, 英文字母、数字、中划线 (-), 下划线 (_), 最小长度1, 最大长度64。 说明 边缘节点ID自定义(不输入默认生成)。
验证码	输入自定义验证码 (可选填写)。验证码, 执行安装边缘软件命令时与云端通讯的凭证。
设备归属实例	选择默认归属实例。
设备归属资源空间	选择默认资源空间, 或为本实验单独新建一个资源空间, 如 edge。
IoTDA南向接入地址	选择网关归属的IoTDA实例的南向设备mqtt接入地址, 也可以选择自定义的南向接入地址, 支持“ip:端口”或“域名:端口”形式。
鉴权方式	密钥: 默认生成。 X.509证书: 请参考: 基于MQTT.fx的X.509证书接入指导 。 警告 鉴权方式为“密钥”的节点, 请勿在IoTDA界面重置节点的密钥, 重置密钥会导致网关与IoTDA通信异常, 且不可恢复。
指纹	证书文件生成的字符串 (可选填写)。只能为40位十六进制字符串或64位六进制字符串。
本地路径	输入证书文件和密钥文件在设备上的存储路径, 只允许字母、数字、下划线 (_) 和斜线 (/), 以/开头并且以.pem或.crt结尾。
自动升级	可选, 开启后节点将自动升级系统应用模块。 注意 升级过程中可能导致节点数据上传中断。

步骤4 填写配置节点数据信息后, 单击“立即创建”。

图 6-2 配置节点数据

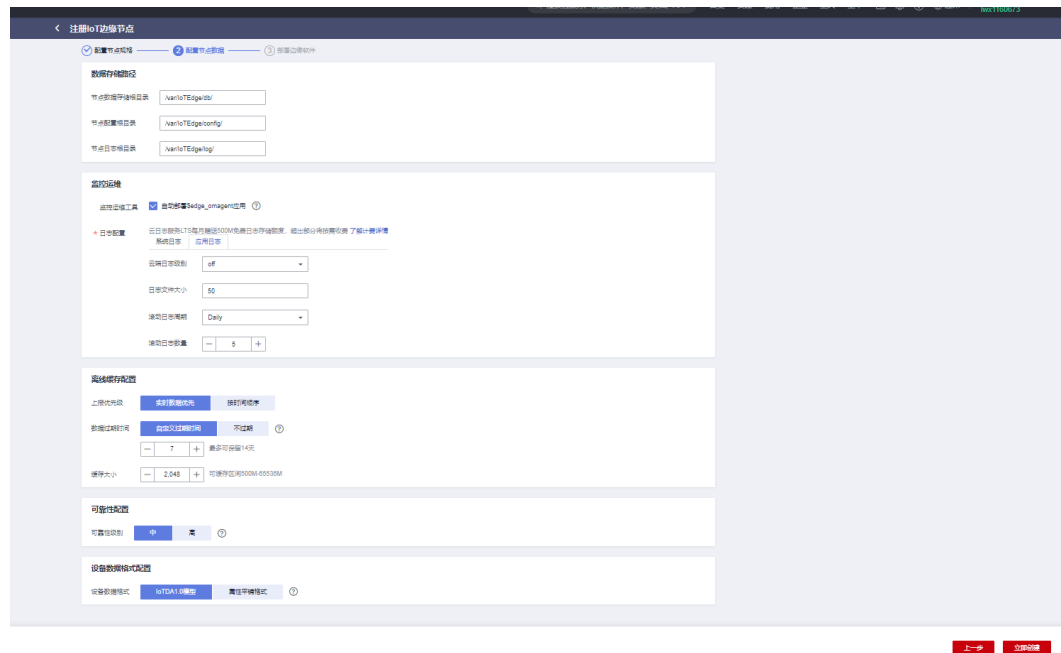


表 6-2 专业版节点信息

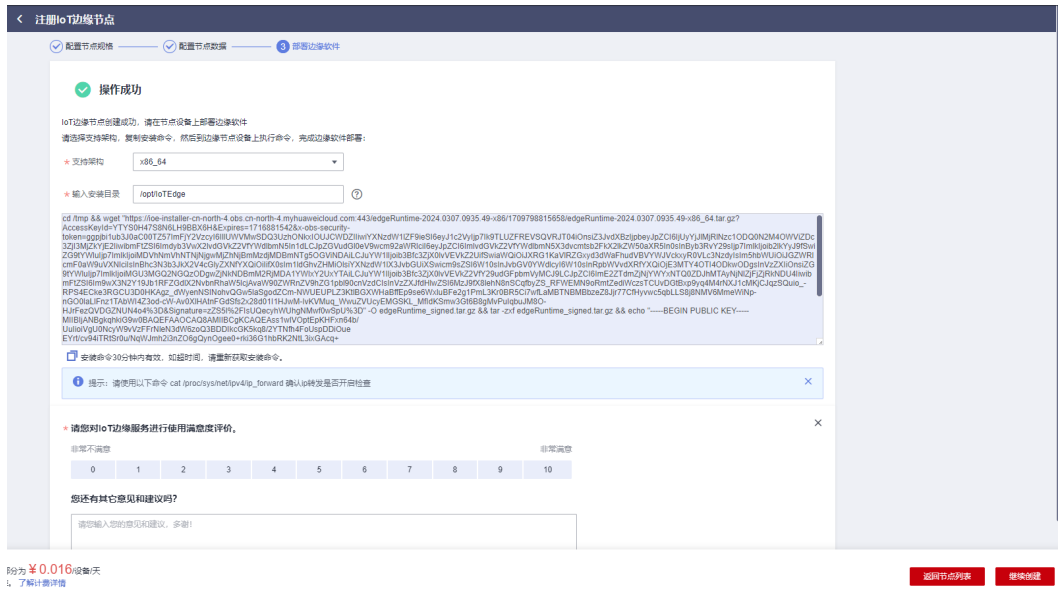
参数名称	说明
数据存储路径	默认（允许修改存储路径）。
监控运维	默认勾选“监控运维工具”。
\$edge_omagent应用	负责边缘节点远程监控和运维，提供日志上报、远程SSH、CPU等指标上报、文件上传下载能力。

参数名称	说明
日志配置	<p>系统日志：边缘节点上已部署的系统应用产生的日志。 应用日志：边缘节点上已部署的用户定义应用所产生的日志。</p> <ul style="list-style-type: none"> • 云端日志级别：on, off, trace, debug, info, warn, error, fatal。其中，on表示上传所有日志至云端，off表示不上传日志至云端，其余可选项代表按日志级别采集，选择后将只采集目标级别及以上的日志（模糊匹配）。日志级别的严重程度等级由低到高依次为：trace, debug, info, warn, error, fatal。 • 日志文件大小：日志文件大小限制，单位MB，默认50，取值范围10-1000。某个日志文件如果达到大小限制，则会将文件转储保存至日志目录下。 • 滚动日志周期：日志转储周期，可选项：daily, weekly, monthly, yearly。某个日志文件如果达到转储周期，则会将文件转储保存至日志目录下。日志文件大小和滚动日志周期是同时生效的，满足任何一个条件都会进行日志转储。 • 滚动日志数量：日志文件转储个数，默认5，取值范围1-10。边缘节点保存的转储日志数量如果达到限制，则会删除最早的转储文件。 <p>注意</p> <ol style="list-style-type: none"> 1. 当前“云端日志级别”对日志等级采取模糊匹配的策略，若日志消息中存在与日志级别标识相同的关键字，则也可能被采集。 2. 若要开启“云端日志级别”中的trace, debug, info, warn, error, fatal选项，则必须满足以下三个前置条件： <ol style="list-style-type: none"> 1. 目标节点是基础版节点或部署类型为Docker的专业版节点。需注意，部署类型为KubeEdge的专业版节点不支持此功能。 2. 已安装\$edge_omagent应用，且版本不低于1-1-15-xxx。 3. 应用日志需遵循如下规范打印，以便进行过滤： <ul style="list-style-type: none"> ○ 日志级别标识需打印为（大小写均可）：trace、debug、info、warn、error、fatal。 ○ 每条日志消息以YYYY-MM-DD日期格式作为前缀。 <p>说明 专业版（部署类型为Docker）和基础版节点只涉及日志级别这一个参数，其他参数只对专业版（部署类型为KubeEdge）有效。</p>
离线缓存配置	<p>可对离线节点上报的数据进行缓存，定义缓存周期（可配置为不过期）跟缓存大小、上报的优先级、路由转发是否开启离线缓存。仅当上报优先级为“按时间顺序”时支持路由转发开启离线缓存。</p>
可靠性配置	<p>可靠性级别分为“中”和“高”2种。中级别：当应用上报属性和点位的频率超过流控值时，超过部分会被丢弃；高级别：当应用上报属性和点位的频率超过流控值时，通过控制模块的发送速度，降低上报频率，报文不会被丢弃。（注意：高可靠性级别的配置需要定制应用的配合，同时会造成性能的损失）。</p>

参数名称	说明
设备数据格式配置	设备的数据格式：即单网关节点中设备的标准数据格式，包含 IoTDA1.0物模型格式和属性平铺格式。IoTDA1.0物模型中包含了设备、服务和属性三层结构，具体参考IoTDA网关子设备属性上报报文格式。属性平铺格式，包含设备、属性2层结构，属性结构可以包含自定义结构的拼接，在json中作为一项平铺展开，具体格式请参考 步骤6设备数据格式配置 。 须知 属性平铺格式功能，需要edge_hub应用1-1-38-xx版本及以上支持

步骤5 弹出如下对话框，您可以“返回节点列表”也可以“继续创建”。

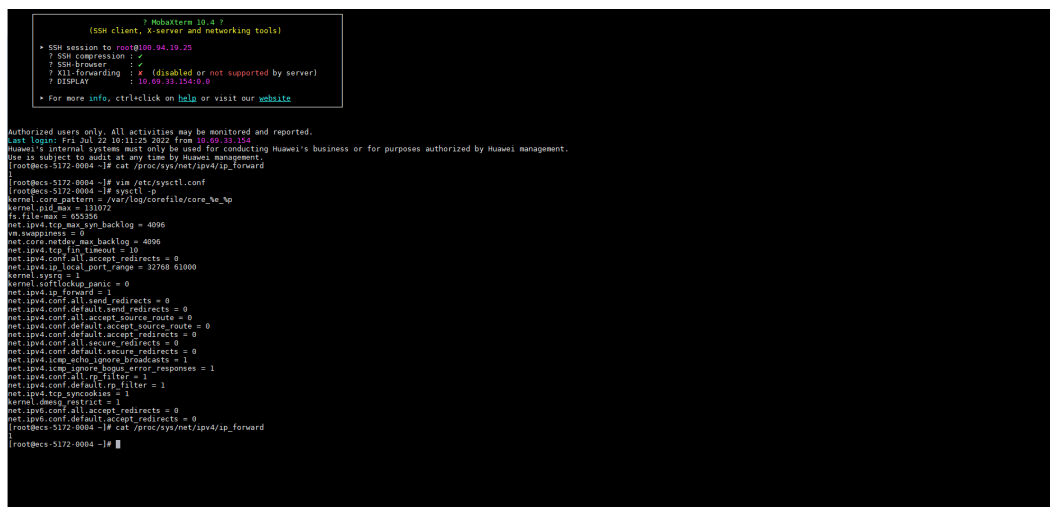
图 6-3 注册成功



说明

安装部署节点前，请确认机器的IP转发已开启并且已开启自启动，以及7883和8943端口开放。其中，IP转发规则配置如下：

图 6-4 IP 转发规则配置



1. 检查机器ip_forward in/proc文件系统的值。
`cat /proc/sys/net/ipv4/ip_forward`
 若值等于0，则需要添加配置IP转发规则。
 2. 检查配置文件/etc/sysctl.conf是否包含net.ipv4.ip_forward = 1。
 如果配置文件中没有net.ipv4.ip_forward参数项，请新增一行net.ipv4.ip_forward = 1。
`vim /etc/sysctl.conf`
 按“i”进入编辑模式，修改完成后按“Esc”退出编辑模式，然后按“Shift” + “;”组合键，输入wq并回车，保存修改并退出。
 3. 重新加载此文件的值。
`sysctl -p`
 4. 检查7883和8943端口是否开放，如果是华为云服务器，请参考：[安全组](#)。
- 结束

安装专业版节点

步骤1 在“节点管理”的节点列表选择您需要安装的节点，单击右侧“安装”，获得提示。

图 6-5 安装

节点名称	节点类型	资源包类型	资源包ID	状态	主机名/网络	操作
	专业版	--	--	未安装		删除 安装
	专业版	--	--	离线	172.30.7.64,172.17.0.1,1...	删除
	专业版	--	--	已安装		删除


步骤2 选择支持架构和安装目录，然后单击  复制安装命令，使用SSH工具以root用户登录边缘节点服务器后台系统，执行安装命令。

图 6-8 安装命令执行成功

```

2024-05-28T14:44:29.581321 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:PENDING, reason:(zero) ]
2024-05-28T14:44:31.632105 | INFO | saveBasePathInfo | 76 | config base path start.
2024-05-28T14:44:31.632291 | INFO | saveBasePathInfo | 83 | config log path:/var/IoTEdge/log/, config_path:/var/IoTEdge/config/
2024-05-28T14:44:31.632457 | INFO | saveBasePathInfo | 95 | config base path success.
2024-05-28T14:44:31.632535 | INFO | addAgentCustomConfigs | 261 | addCustomConfigs start.
2024-05-28T14:44:31.632603 | INFO | addAgentCustomConfigs | 285 | addCustomConfigs end.
2024-05-28T14:44:31.632654 | INFO | pullImage | 319 | start pull image, deploy type is cloud
2024-05-28T14:44:35.888319 | INFO | createModule | 76 | create Module, name:sys_edge_agent
2024-05-28T14:44:35.939013 | INFO | createModule | 94 | module created, CreateRespBody: ModuleCreateResp[ moduleId:sys_edge_agent, moduleName:sys-edge-agent, appId:sedge_agent, appVersion:test-tpm-2-x86, containerId:4b3e43cee7eac9bc282b994377abfbf48a264389a36856f3cacc0338743036e, processId:(zero), type:docker, ]
2024-05-28T14:44:42.398934 | INFO | startModule | 160 | startModule moduleId:sys_edge_agent
2024-05-28T14:44:42.440775 | INFO | listModule | 250 | listModule
2024-05-28T14:44:42.440775 | INFO | checkModulesDo | 340 | start check modules, time:1
2024-05-28T14:44:42.440908 | INFO | printModulesAndReport | 404 | ***success:1 failed:0
2024-05-28T14:44:42.491163 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:RUNNING, reason:(zero) ]
2024-05-28T14:44:42.542192 | INFO | reportEvent | 158 | Begin reporting event: {"events":[{"resource":"node.installation","location":"zzzzzz","event":"succeeded","event_severity":"Info", "event_time":"20240528T064442Z","notify_data":{"host_tag":"DEFAULT","edge_node_id":"zzzzzz"}}]}
2024-05-28T14:44:42.592932 | INFO | reportModuleStatus | 138 | reportModuleStatus, modulesInfo:ModuleStatusInfoList[ host_tag:DEFAULT, deploy_id:1716878669308, modules:[ModuleStatusInfo module_id:sys_edge_agent, app_id:sedge_agent, app_version:test-tpm-2-x86, container_state:RUNNING, liveness_state:(zero), update_time:1716878669305,], ]
2024-05-28T14:44:43.043689 | INFO | checkBridgeNetwork | 556 | Begin checking bridge network
2024-05-28T14:44:50.524137 | INFO | install | 494 | install edge finish.
2024-05-28T14:44:50.524559 | INFO | stop | 545 | edgeinstaller stop.
/root
    
```

步骤3 单击“我知道了”，等待边缘节点的状态变为“在线”，表示该节点已安装并连接成功。

步骤4 您可以单击“节点名称”，查看该节点详细信息，关于节点详情描述可参见[节点概览](#)。

----结束

6.2.2 准备数据源

安装 OPC-UA 服务器模拟器

步骤1 访问[Prosyst OPC UA Simulation Server 下载页面](#)，下载并安装最新版本的Prosyst OPC UA Simulation Server 模拟器。

步骤2 安装完毕后，启动Prosyst OPC UA Simulation Server 模拟器。

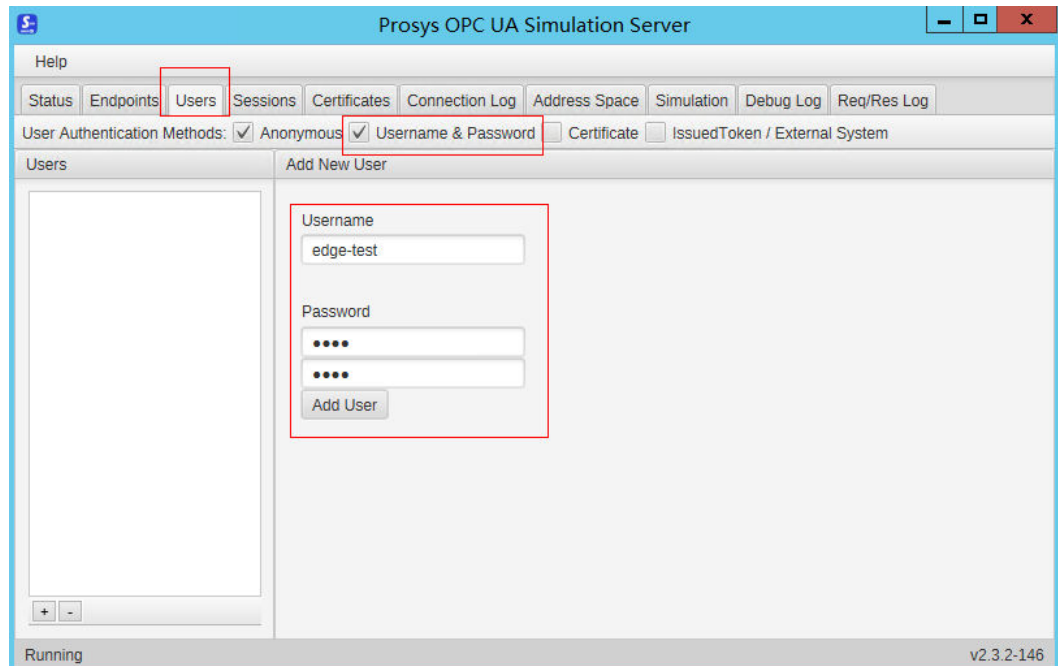


步骤3 Connection Address即是该OPC-UA服务端提供给OPC-UA Client端进行连接的地址。

其暴露的URL通常格式如“opc.tcp://{主机名}:53530/OPCUA/SimulationServer”。需把主机名替换为该服务器的公网IP。如opc.tcp://121.36.62.255:53530/OPCUA/SimulationServer

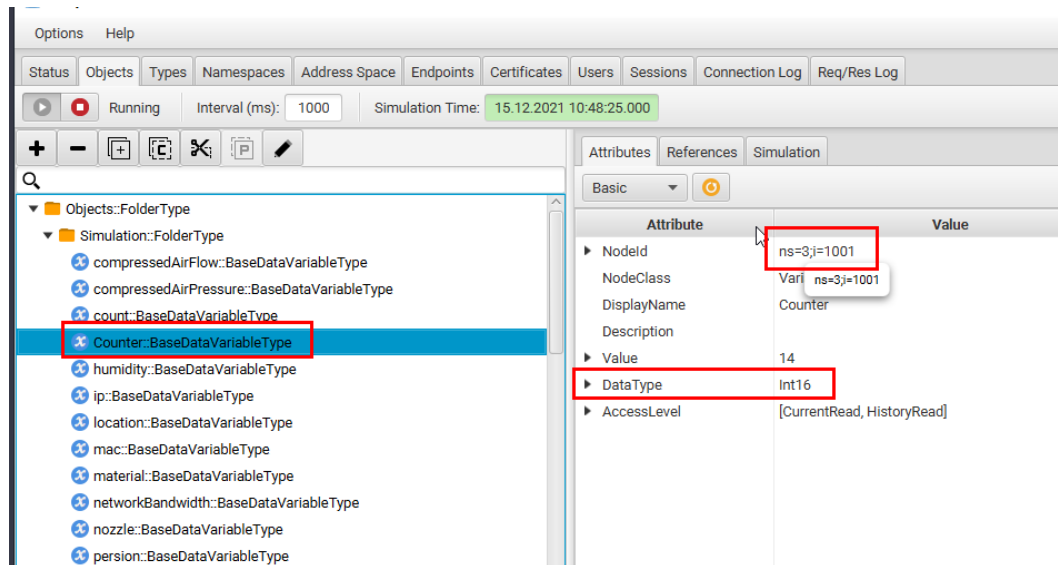
步骤4 配置OPC-UA服务端的用户身份，添加一个用户名和密码，如

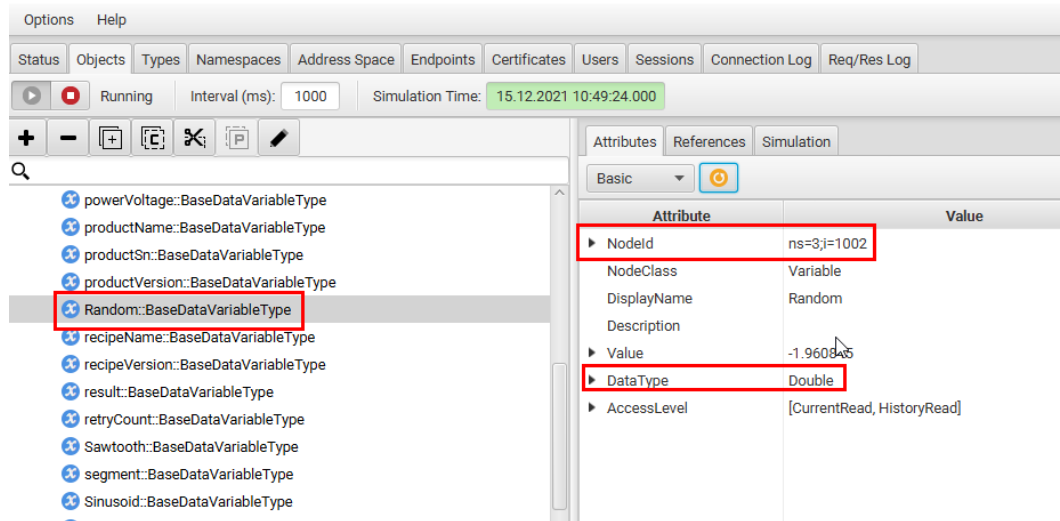
- 用户名: edge-test
- 密码: 1234



步骤5 定义数据源属性。

例如在OPC UA服务端模拟器的地址空间中，可认为Simulation是接入OPC UA服务端的一个设备，该设备有 Counter, Random等属性数据，地址为ns=3;i=1001和 ns=3;i=1002。





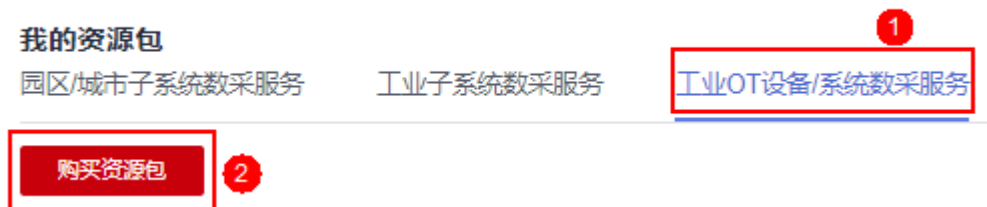
----结束

6.2.3 部署边缘数采应用

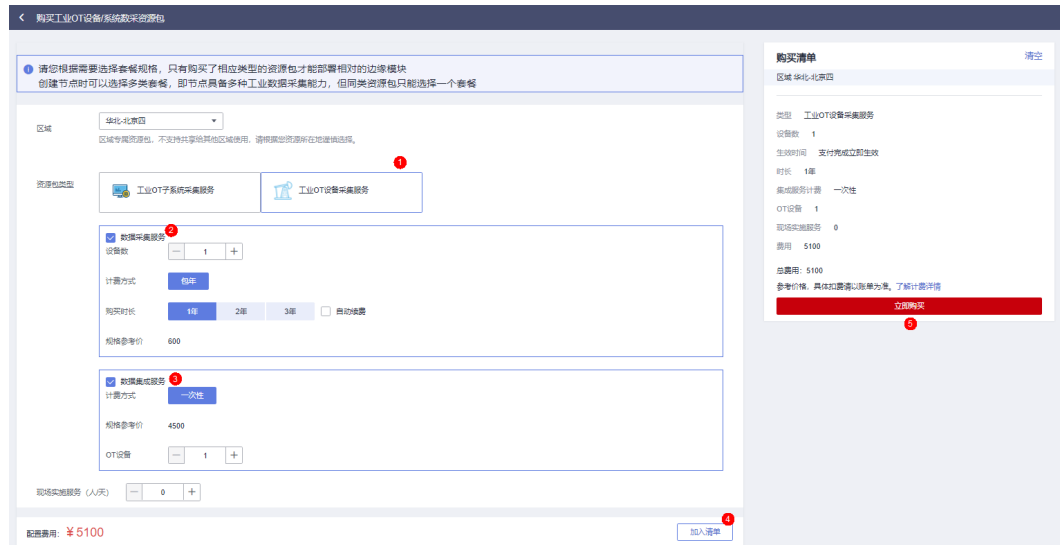
购买 OT 数采服务资源包

用户需先购买服务，购买完成后即可使用IoT边缘数采功能。

- 步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2 在左侧导航栏选择“总览”，单击右上角“我的数采服务”。
- 步骤3 选择“工业OT设备/系统数采服务”，单击“购买资源包”。



- 步骤4 选择“工业OT设备采集服务”，根据需求选择合适的规格加入清单，勾选“数据采集服务”和“数据集成服务”，配置完成后单击“加入清单”再“立即购买”。



步骤5 数据集成服务付款成功后，可在华为云页面右上角“支持与服务 > 专业服务 > 我的服务单”中查看。



说明

成功购买数据集成服务后，将有专业物联网工程师与您联系进行数据采集实施对接。
未购买数据集成服务时，在节点下部署行业数采应用将会失败。

----结束

部署边缘数采应用

如未添加边缘数采应用，可参考[添加驱动应用](#)。

步骤1 在左侧导航栏选择“边缘节点 > 节点管理”，选择之前创建的边缘节点“test_loT10”，单击“详情”进入节点详情页。

步骤2 选择左侧“应用模块 > 模块管理”，单击“部署应用”填写基本信息后“确认”。

- 功能用途: 数据采集
- 边缘应用: \$ot_dc_opcua
- 选择版本: 当前最新的版本



步骤3 边缘应用部署任务下发成功后，应用状态为已停止。

模块ID	模块名称	所属应用	版本	应用类型	实例状态	操作
sys_edge_access	sys_edge_access	\$edge_access	2022-02-25-standard-x86	系统可选	运行中	删除 升级
sys_edge_agent	sys_edge_agent	\$edge_agent	1-1-8-standard-x86	系统可选	运行中	升级
sys_edge_apigw	sys_edge_apigw	\$edge_apigw	1-1-2-x86	系统可选	运行中	删除 升级
sys_edge_hub	sys_edge_hub	\$edge_hub	1-1-20-standard-x86	系统可选	运行中	升级
sys_ot_dc_opcua	sys_ot_dc_opcua	\$ot_dc_opcua	1-0-4-standard-x86	系统可选	已停止	删除 升级

说明

数采应用部署成功后或者升级数采应用之后，应用状态都显示为已停止，只有在数采配置启用了数采连接后，应用状态会变更为运行中。

----结束

6.2.4 设备建模&发放

创建产品

使用物联网平台的第一步就是在控制台创建产品。产品是设备的集合，是指某一类具有相同能力或特征的设备的合集被称为一款产品。

步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。

步骤2 在左侧导航栏选择“边缘节点 > 设备建模”进入页面。

步骤3 单击右上角“创建产品”，按照页面提示填写信息，然后单击“确定”。

×

创建产品

* 所属资源空间 ?

* 产品名称

协议类型

* 数据格式 ?

厂商名称

* 设备类型 ?

立即创建
取消

参数名称	说明
所属资源空间	与之前创建边缘节点时选择的资源空间保持一致。
产品名称	自定义，如test_loT10。
协议类型	Other 说明 使用OT数采能力时，建模协议类型必须选择Other。
数据格式	默认值，JSON。
厂商名称	自定义，如Huawei。
设备类型	自定义。

步骤4 单击“设备建模 > 产品”，选择刚才创建的产品“test_loT10”，单击详情进入产品详情页面。

步骤5 在“模型定义”页签下，单击“自定义模型”。



步骤6 填写“服务ID”、“服务类型”，单击“确定”。

- 服务ID、服务类型自定义。

×

新增服务

* 服务ID

* 服务类型

服务描述

确认 取消

步骤7 展开服务列表，单击“添加属性”，分别添加电流、电压两个属性。

模型定义

添加服务上传模型文件Excel导入

^ 服务ID: Motor 🗑

服务类型: Motor

服务描述:

添加属性

属性名称	数据类型	是否必选

添加命令

命令名称	下发参数

添加“电流”属性，属性名称为“Current”，如下图所示：

新增属性

×

* 属性名称 必选

属性描述

* 数据类型

* 访问权限

取值范围 -

步长

单位

添加“电压”属性，属性名称为“Voltage”，如下图所示：

修改属性 ✕

* 属性名称

属性描述
0/128

* 数据类型

* 访问权限

* 取值范围 -

步长

单位

步骤8 模型定义完成，参数信息预览如下图所示。

属性名称	数据类型	访问方式	描述	操作
<input type="checkbox"/> Current	decimal(小数)	可读,可写		复制 修改 删除
<input type="checkbox"/> Voltage	int(整型)	可读,可写		复制 修改 删除

10 总条数: 2 < 1 >

----结束

📖 说明

支持将本地开发的产品模型导入到平台。

导入步骤：选择“设备建模 > test_loT10”，单击“上传模型文件”，在弹窗中添加模型文件，单击“确定”完成上传。



添加边缘设备

用户创建产品后，需要在边缘节点上添加对应的边缘设备，用于接收边缘节点实际采集的OT设备数据。

- 步骤1** 在左侧导航栏选择“边缘节点 > 节点管理”选择之前创建的边缘节点“test_loT10”，单击节点进入详情页。
- 步骤2** 选择左侧导航栏“边缘设备”，单击“添加边缘设备”，根据提示填写参数后，单击“确认”。



参数名称	说明
所属产品	选择步骤3创建的产品“test_loT10”。
设备标识码	自定义。只允许英文字母、数字，下划线，中划线。最大长度64。
设备名称	自定义。设备名称长度为1~256位，只能由数字,字母,中文,逗号,括号,#,.,?!,-,@,%,&!组成。

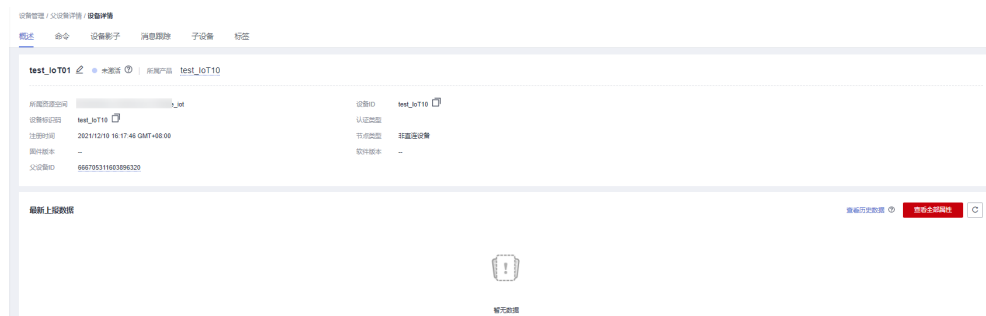
参数名称	说明
模板ID	为空。 说明 使用Other数采能力设备必须选择协议类型为Other的产品，且模块id必须为空。

步骤3 添加设备成功后，可以在设备列表中查看“test_loT10”设备状态，并管理边缘设备。此时设备处于未激活状态，需完成设备部署接入后，可在物联网平台查看设备上报的数据。

图 6-9 设备列表



单击对应的“设备ID”跳转至“设备接入服务控制台”进行设备管理。



----结束

6.2.5 OT 数采配置

配置数据源

数据源提供OT数据的设备和系统，如DCS、SCADA、PLC、数据库等。

创建数采连接

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_loT10”进入页面。
- 步骤3** 在左侧导航栏选择“数采配置 > 创建数采连接”。

图 6-10 创建数采连接



步骤4 填写“数采连接参数”后单击“确认”保存，此时配置仅在云平台存在，未下发到边缘，数采连接的“同步状态”为“待同步”。

表 6-3 连接参数信息

参数名称	参数说明
通道标识	opcua
通道名称	opcua通道 (自定义)
驱动模块	sys_ot_dc_opcua
服务端点:	opc.tcp://121.36.62.255:53530/OPCUA/SimulationServer
用户名	与准备数据源 步骤4 保持一致
密码	与准备数据源 步骤4 保持一致
采集周期	1000毫秒

说明

服务端点: 填写OPC-UA服务端提供的连接地址，常见的OPC-UA Server有SCADA系统，Kepware等软件。在本实验**准备数据源**中使用Prosys OPC UA Simulation Server来模拟一个OPC-UA服务端，服务端点填写该模拟器的url。

---结束

添加点位映射

将点位上报转换为设备属性上报，将设备属性读写转换为点位读写操作，将设备命令转化为点位写操作。

步骤1 访问**IoT边缘**，单击“管理控制台”进入IoT边缘控制台。

步骤2 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_loT10”进入页面。

步骤3 在左侧导航栏选择“数采配置 > 选择对应的数采连接”。

步骤4 单击“添加点位”在“添加点位映射”弹窗中填写配置信息，如表6-4所示，然后单击“确认”。



表 6-4 添加点位映射

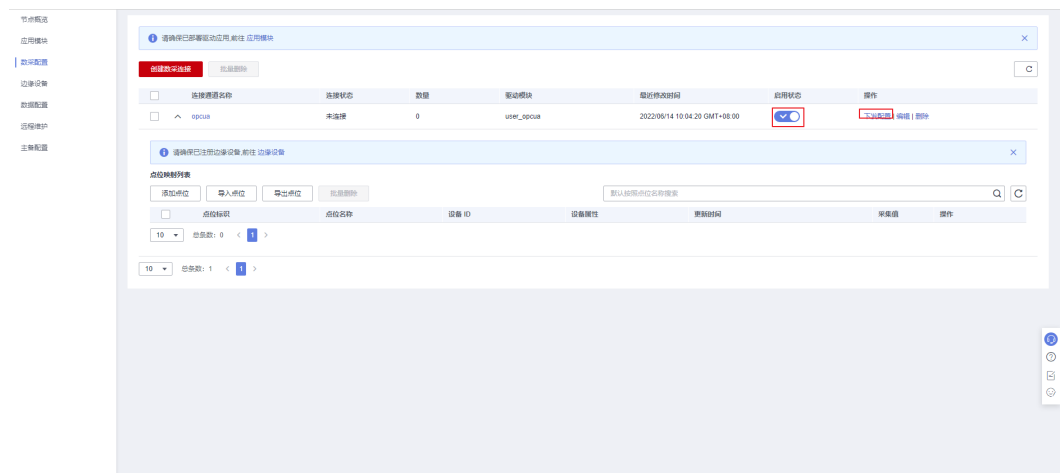
基本信息	
点位标识	点位的唯一标识。
点位名称	按照规范填写。
设备ID	边缘设备页签里添加设备的ID。
点位数据类型	<p>当前支持如下数据类型：</p> <ul style="list-style-type: none"> int(整型) uint(无符号整型) int64(64 位整型) short(短整型) ushort(无符号短整型) long(长整型) ulong(无符号长整型) bool(布尔类型) float(单精度浮点型) double(双精度浮点型) decimal(小数) string(字符串) object(对象类型) <p>取值范围：</p> <ul style="list-style-type: none"> -2147483648~2147483647 0~4294967295 -2147483648~2147483647 -32768~32767 0~65535 -2147483648~2147483647 0~4294967295 0/1 -3.4*10³⁸~+3.4*10³⁸ -1.7*10³⁰⁸~1.7*10³⁰⁸ -1.7*10³⁰⁸~1.7*10³⁰⁸
设备属性	设备属性由物模型中service_id/property_name格式构成。
采集配置	
点位地址	填写OT设备及系统的实际点位地址“ns=3;i=1001和ns=3;i=1002”。

基本信息		
点位值的数据类型	当前支持如下数据类型： int(整型) uint(无符号整型) int64(64 位整型) short(短整型) ushort(无符号短整型) long(长整型) ulong(无符号长整型) bool(布尔类型) float(单精度浮点型) double(双精度浮点型) decimal(小数) string(字符串) object(对象类型)	取值范围： -2147483648~2147483647 0~4294967295 -2147483648~2147483647 -32768~32767 0~65535 -2147483648~2147483647 0~4294967295 0/1 -3.4*10 ³⁸ ~+3.4*10 ³⁸ -1.7*10 ³⁰⁸ ~1.7*10 ³⁰⁸ -1.7*10 ³⁰⁸ ~1.7*10 ³⁰⁸
点位采集周期	1000毫秒。	

注意

不同点位请勿配置同一设备的相同属性。

步骤5 单击“下发配置”，把点位配置下发至边缘侧，此时数采连接的“同步状态”为“已同步”。



---结束

说明

1. 修改点位信息后需重新单击下发配置，配置才会生效。
2. 模块的运行状态由启用状态决定，当启用了数采连接后，模块会显示为运行中。

添加数采模板

描述数采配置项的元数据，数采配置包括数据源配置元数据、点位采集配置元数据。

系统数据源模板导入

数据源模板描述了数据源配置项的元数据，用户在配置数据源之前，需要先导入数据源模板。IoT边缘服务当前已经预置通用数据源模板，可以直接使用，系统也支持用户自定义导入模板。

步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。

步骤2 在左侧导航栏选择“边缘节点 > 应用管理”进入页面，

步骤3 选择“驱动应用 > 数采模板管理 > 添加数采模板”。



步骤4 添加文件，单击“确认”。

- 本地准备一份OT数采模板文件，文件内容合规。
- IoT边缘云平台当前已经预制2个业界主流通用数采模板，可以通过导入通用模板后使用。
- 支持用户自定义导入模板，导入模板后缀.json，满足json格式规范，样例及字段解释如下：

示例

```
{
  "tpl_id": "sys_general_opcua",
  "name": "opcua通用数采模板",
  "description": "opcua通用数采模板",
  "datasource_meta": {
    "config_tabs": [
      {
        "key": "connection_info",
        "name": "连接信息",
        "description": "连接信息",
        "config_items": [
          {
            "key": "endpoint",
            "name": "服务端点",
            "description": "服务完整URL",
            "data_type": "string",
            "required": true,
            "crypted": false,
            "max_length": 128,
            "example": "opc.tcp://127.0.0.1:53530/OPCUA"
          }
        ]
      }
    ]
  }
}
```



```
    }, {
      "key": "username",
      "name": "用户名",
      "description": "OPCUA服务器认证用户名",
      "data_type": "string",
      "required": false,
      "crypted": false,
      "max_length": 128,
      "example": "admin"
    }, {
      "key": "password",
      "name": "密码",
      "description": "OPCUA服务器认证密码",
      "data_type": "string",
      "required": false,
      "crypted": true,
      "max_length": 512,
      "example": "*****"
    }
  ]
}, {
  "key": "collection_paras",
  "name": "连接附加参数",
  "description": "连接附加参数",
  "config_items": [
    {
      "key": "default_cycle",
      "name": "采集周期",
      "description": "采集周期",
      "data_type": "int",
      "required": true,
      "crypted": false,
      "max_length": 65535,
      "example": 10000
    }
  ]
},
"default_values": {
  "drivername": "OPCUA"
}
},
"point_meta": {
  "config_items": [
    {
      "key": "address",
      "name": "点位地址",
      "description": "采集点位的地址",
      "data_type": "string",
      "required": true,
      "crypted": false,
      "max_length": 256,
      "example": "ns=2;s=Root/Motor/Voltage"
    }
  ], {
    "key": "data_type",
    "name": "点位值的数据类型",
    "description": "点位地址存储值的数据类型",
    "data_type": "string",
    "required": true,
    "crypted": false,
    "max_length": 128,
    "example": "int, decimal"
  }, {
    "key": "cycle",
    "name": "点位采集周期",
    "description": "点位采集周期",
    "data_type": "int",
    "required": true,
    "crypted": false,
    "max_length": 65535,
    "example": 10000
  ]
}
```

```
}
}
```

表 6-5 OT 数采模板

key	类型	描述
tpl_id	String	数采模板标识, 租户下唯一 pattern: '^[a-zA-Z0-9_-]*\$' 1-64
name	String	数据源模板名称, 支持中英文 1-64
description	String	数据源模板描述, 支持中英文, 0-128
datasource_meta	Object	数据源配置元数据
point_meta	Object	点位采集配置元数据

表 6-6 datasource_meta

key	类型	描述
config_tabs	Array	数据源配置表格列表
default_values	Object	数据源默认的值, 通常用于描述默认的驱动协议

表 6-7 config_tabs

key	类型	描述
key	String	数据源配置表格键值, pattern: 'connection_info collection_paras'
name	String	数据源配置表格名称, 1-64
description	String	描述, 0-255
config_items	Array	配置项列表

表 6-8 config_items

key	类型	描述
key	String	配置项key, 1-32
name	String	默认项名称,1-64
description	String	配置项描述, 0-128
data_type	String	配置项类型, 支持short ushort int int64 uint long ulong float double bool string object decimal。
required	Boolean	配置项是否必传项
crypted	Boolean	配置项是否加密
max_length	Integer	当data_type= string时配置有效, 校验输入的字符串最大长度
example	String	配置项举例, 值在界面输入框作为示例灰色显示, 0-256

表 6-9 default_values

key	类型	描述
drivername	String	用于描述默认的驱动协议, 如OPCUA。

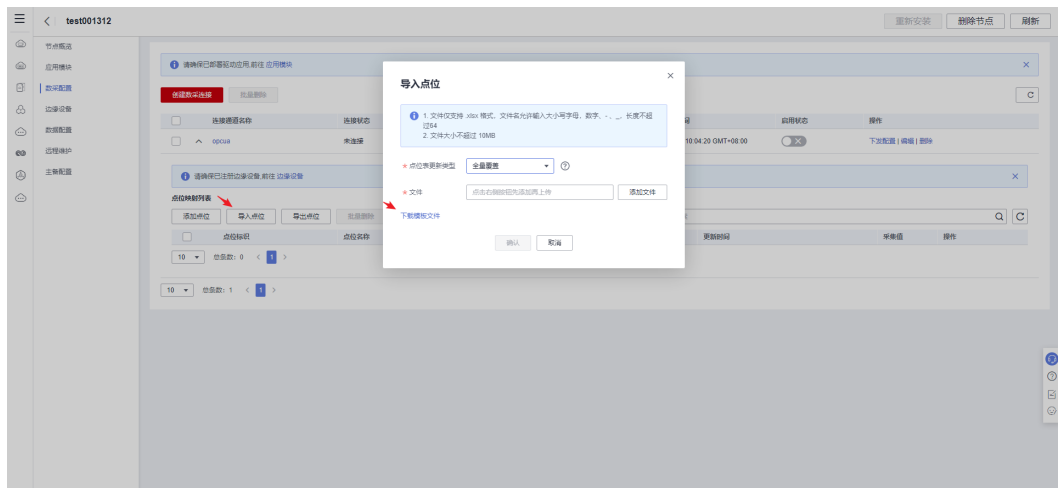


----结束

附录：批量导入/导出点位映射

批量导入点位映射

- 步骤1** 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_loT10”进入页面。
- 步骤2** 在左侧导航栏选择“数采配置”->对应的连接通道。
- 步骤3** 单击“导入点位”单击“下载模板文件”，打开“模板文件”内容填写。



- 步骤4** 选择“添加文件”导入**步骤3**下载的文件后，单击“确认”完成批量导入点位映射。



----结束

批量导出点位映射

- 步骤1** 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_loT10”进入页面。
- 步骤2** 在左侧导航栏选择“应用模块 > 数采配置 > 对应的连接通道”。
- 步骤3** 单击“导出点位”完成批量导出点位。



----结束

6.2.6 查看采集结果

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 在左侧导航栏选择“边缘节点 > 节点管理”，选择之前创建的边缘节点“test_loT10”，单击节点进入详情页。
- 步骤3** 选择“边缘设备”页签，单击添加边缘设备中添加的设备ID“test_loT10”，跳转至“设备接入服务控制台”查看采集的数据。

图 6-11 设备列表



说明

OT数采设备状态平台不感知，即统一显示为“未激活”，请进入设备详情页，查看最新上报数据。



----结束

6.2.7 (可选) 点位拆分

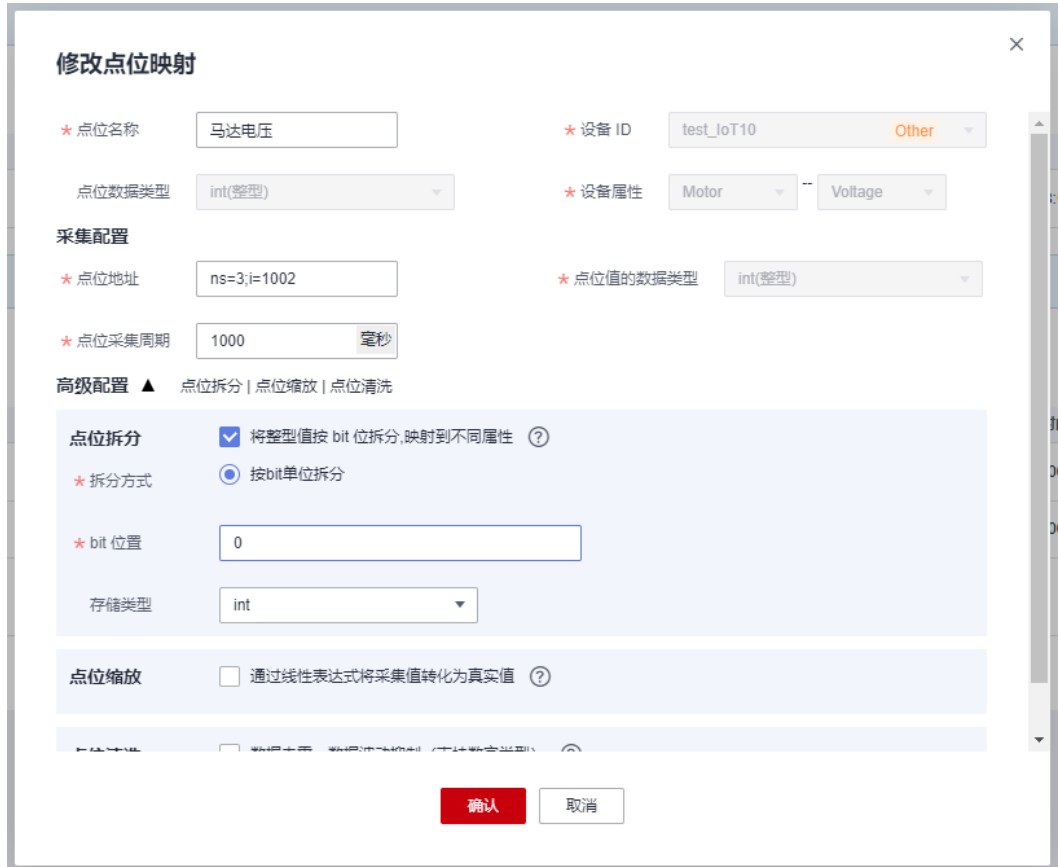
点位拆分只支持数据类型为整型的点位，即int、uint、int64、short、ushort、long、ulong这7种类型。

按比特位拆分

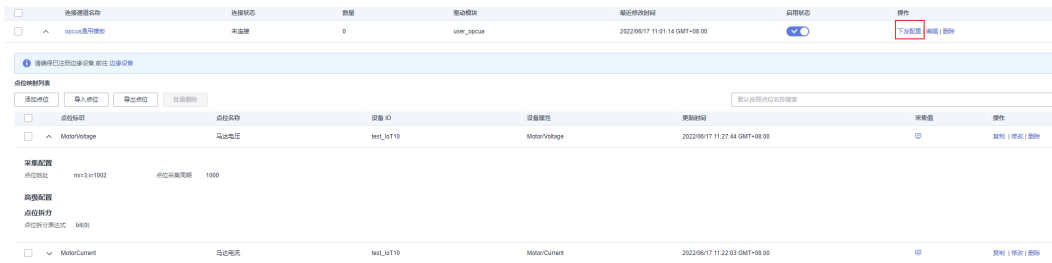
增加点位拆分配置，在基于IoT边缘实现OT数采章节的配置上，可以选择“高级配置 > 点位拆分”。

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_loT10”进入页面。
- 步骤3** 在左侧导航栏选择“数采配置 > 选择对应的数采连接”。
- 步骤4** 选择已有点位MotorVoltage，操作列--单击修改，勾选“点位拆分”选项，bit位置设置为“0”，存储类型选择“int”，即上报的数据为该点位“比特0”的值，单击“确认”

只有点位的数据类型为“整型”才支持点位拆分，即int、uint、int64、short、ushort、long、ulong这7种类型



步骤5 修改点位配置后，单击“下发”，将配置更新给边缘网关。



步骤6 查看结果

1. 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
2. 在左侧导航栏选择“边缘节点 > 节点管理”，选择之前创建的边缘节点“test_loT10”，单击节点进入详情页。
3. 选择“边缘设备”页签，单击添加边缘设备中添加的设备ID“test_loT10”，跳转至“设备接入服务控制台”查看采集的数据。
4. 上报的数据如下图所示，Voltage的值为MotorVoltage点位“比特0”的值为“1”。



----结束

按比特位拆分并转换成布尔类型

增加点位拆分配置，在基于IoT边缘实现OT数采章节的配置上，可以选择“高级配置 > 点位拆分”。

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_loT10”进入页面。
- 步骤3** 在左侧导航栏选择“数采配置 > 选择对应的数采连接”。
- 步骤4** 选择已有点位MotorVoltage，操作列--单击修改，勾选“点位拆分”选项，bit位置设置为“0”，存储类型选择“bool”，即上报的数据为该点位“比特0”的值，单击“确认”只有点位的数据类型为“整型”才支持点位拆分，即int、uint、int64、short、ushort、long、ulong这7种类型

步骤5 修改点位配置后，单击“下发配置”，将配置更新给边侧。

连接名称	连接状态	数据	驱动模块	最近修改时间	启用状态	操作
opcua通用模块	已连接	2	user_opcua	2022/06/17 11:29:47 GMT+08:00	<input checked="" type="checkbox"/>	下发配置 编辑 删除

点位标识	点位名称	设备 ID	设备属性	更新时间	采集值	操作
MotorVoltage	马达电压	test_loT10	MotorVoltage	2022/06/17 11:35:05 GMT+08:00	0	复制 修改 删除

采集配置
点位地址: ns=3;j=1002 点位采集周期: 1000

高级配置
点位拆分
点位拆分表达式: bit(0;bool)

步骤6 查看结果

1. 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
2. 在左侧导航栏选择“边缘节点 > 节点管理”，选择之前创建的边缘节点“test_loT10”，单击节点进入详情页。
3. 选择“边缘设备”页签，单击添加边缘设备中添加的设备ID“test_loT10”，跳转至“设备接入服务控制台”查看采集的数据。
4. 上报的数据如下图所示，Voltage的值为MotorVoltage点位“比特0”的值为“true”。



----结束

6.2.8 (可选) 点位计算

点位计算特性为可选项，支持点位清洗和点位缩放功能。具体描述请参见点位计算。

⚠ 注意

使用点位计算特性，会造成一定的性能消耗，性能消耗在10%以内。

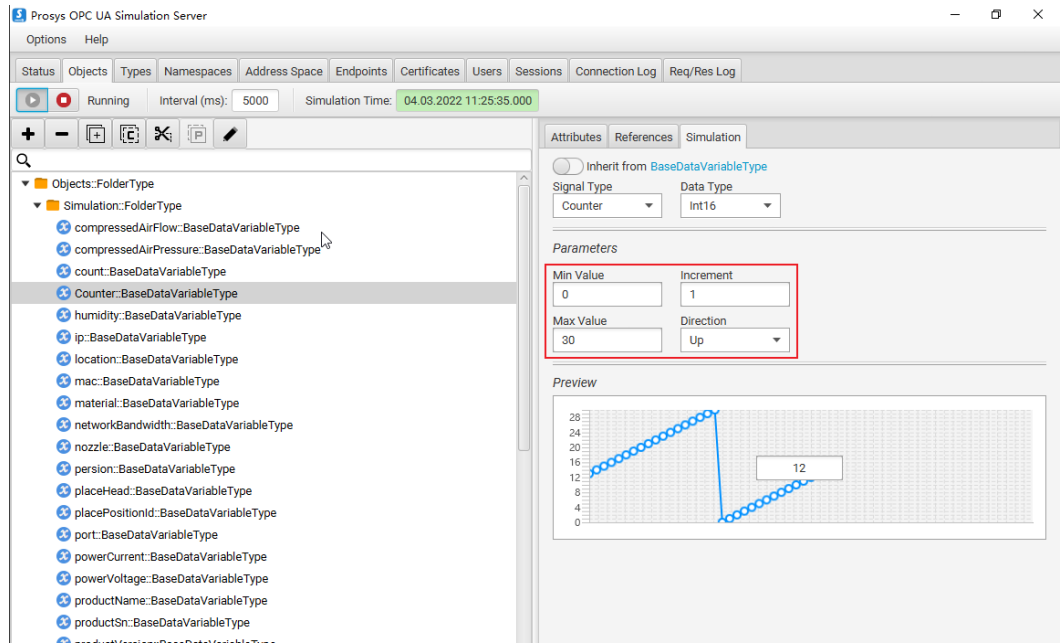
点位缩放

增加点位缩放配置，在基于IoT边缘实现OT数采章节的配置上，可以选择增强功能“高级配置 > 点位缩放”。

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_loT10”进入页面。
- 步骤3** 在左侧导航栏选择“数采配置 > 选择对应的数采连接”。
- 步骤4** 选择已有点位MotorCurrent，操作列--单击修改，勾选“点位缩放”选项，配置缩放倍率为10倍，基准值为10，即公式为 $Y = 10 * X + 10$ ，单击“确定”。
 - 缩放倍率 (ratio)**：浮点型，取值范围[-10000, +10000]，最大支持小数点后7位。
 - 基准值 (base)**：浮点型，取值范围[-10000, +10000]，最大支持小数点后7位。
 - 精度 (accuracy)**：缩放后结果的精度，精确到小数点后几位，直接截取，不四舍五入。整数类型，取值范围[-1, 8]。-1或界面不输入，表示不对小数位进行处理。

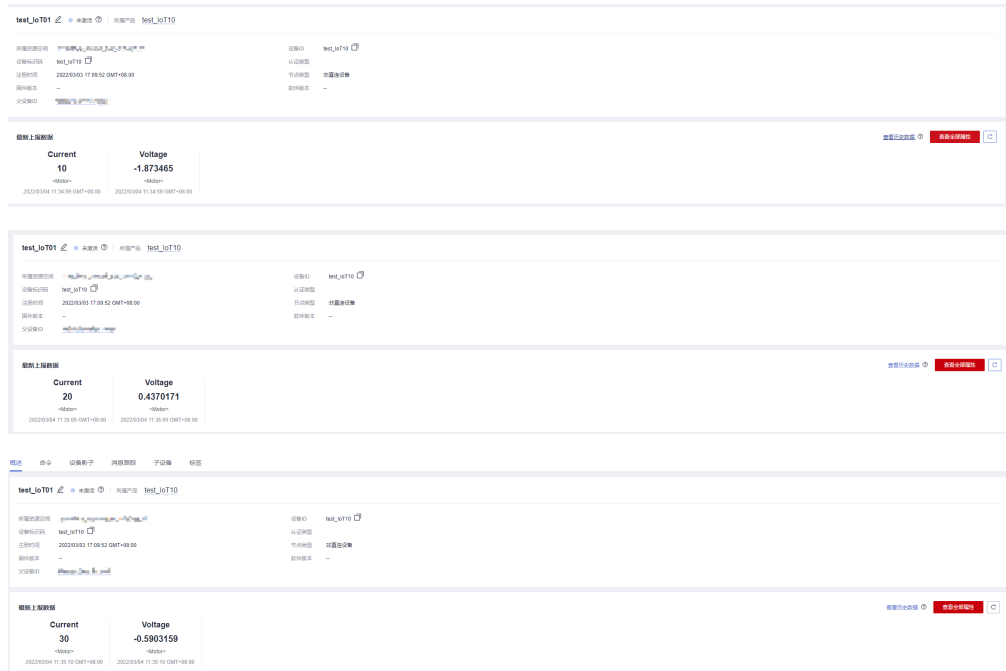
步骤5 修改点位配置后，单击“下发”，将配置更新给边缘网关。

步骤6 修改数据源配置，为了更清晰的看到点位缩放的效果，将数据源的配置进行修改，对点位MotorCurrent对应的数据源地址的值变换规则参数修改。将Min Value=0、Max Value=30、Increment=1，即该点位的数据以0为起点，每秒增加1的频率上升，最大为30。



步骤7 查看结果

1. 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
2. 在左侧导航栏选择“边缘节点 > 节点管理”，选择之前创建的边缘节点“test_loT10”，单击节点进入详情页。
3. 选择“边缘设备”页签，单击添加边缘设备中添加的设备ID“test_loT10”，跳转至“设备接入服务控制台”查看采集的数据。
4. 上报的数据如下图所示，数据源上报的值为0、1、2...，平台收到的值为10、20、30...，采集值通过点位缩放规则 $Y = 10 * X + 10$ 进行运算。



---结束

点位清洗

增加点位清洗配置，在基于IoT边缘实现OT数采章节的配置上，可以选择增强功能“高级配置 > 点位清洗”。

步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。

步骤2 在左侧导航栏选择“边缘节点”，节点名称选择“test_loT10”进入页面。

步骤3 在左侧导航栏选择“数采配置 > 选择对应的数采连接”。

步骤4 选择已有点位MotorCurrent，操作列--单击修改，勾选“点位清洗”选项，配置静默时间窗为100秒，偏差为10%，单击“确定”。

偏差 (Deviation)：震荡的偏差区间值，在该区间的值将视作正常的波动，数据将不进行上报，取值范围[0~100]，类型浮点型。如偏差值为10时，表示在±10% ([-10%, +10%]) 的区间内值将视为正常的波动。

静默窗口 (SilentWin)：静默时间窗口，在静默时间窗口内没有触发上报条件的数据，将不会上报。取值范围(0, 86400]，单位秒，当值为0时，表示时间窗口无穷大。

步骤5 修改点位配置后，单击“下发配置”，将配置更新给边侧。

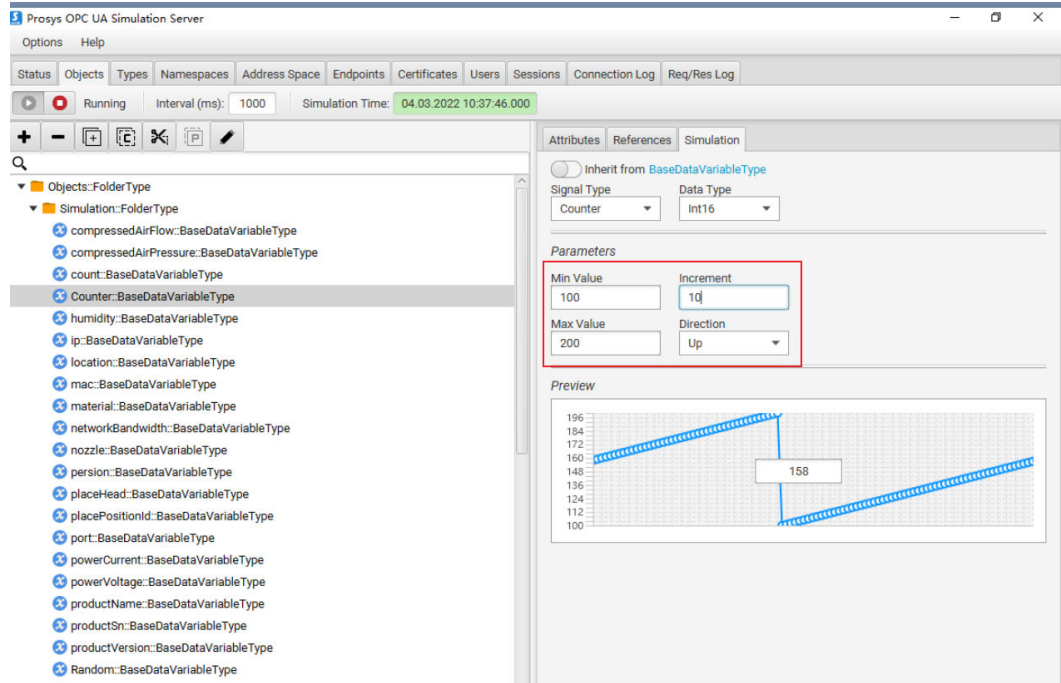
选择配置名称	选择状态	数据	驱动地址	最后修改时间	应用状态	操作
opcu4通用驱动	未选择	0	user_opcu4	2022/06/17 11:01:14 GMT+08:00	<input checked="" type="checkbox"/>	下发配置 删除 刷新

请确保已启用边缘设备，前往 边缘设备						
点位映射列表						
添加点位	导入点位	导出点位	批量删除	默认按照点名称升序		
<input type="checkbox"/>	点位名称	设备 ID	设备属性	更新时间	采集值	操作
<input checked="" type="checkbox"/>	MotorVoltage	test_loT10	MotorVoltage	2022/06/17 11:18:53 GMT+08:00	0	删除 修改 刷新
<input checked="" type="checkbox"/>	MotorCurrent	test_loT10	MotorCurrent	2022/06/17 11:20:10 GMT+08:00	0	删除 修改 刷新

采集配置	
点位地址	ns=3,i=1001
点位采集周期	5000

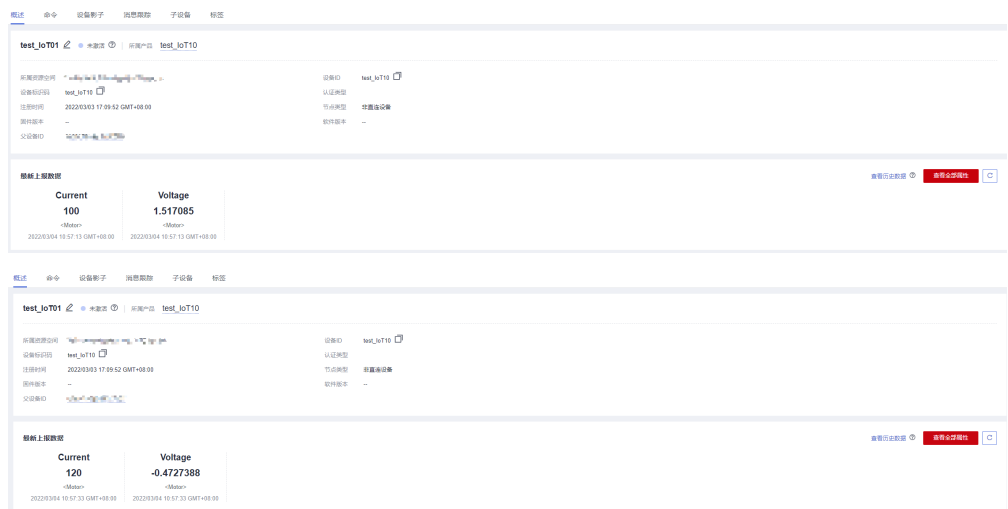
高级配置	
点位清洗	<input checked="" type="checkbox"/> 数据去重、数据波动抑制 (支持数字类型) ?
静默时间窗	100 秒
偏差	10.0000000 %

步骤6 修改数据源配置，为了更清晰的看到点位清洗的效果，将数据源的配置进行修改，对点位MotorCurrent对应的数据源地址的值变换规则参数修改。将Min Value=100、Max Value=200、Increment=10，即该点位的数据以100为起点，每秒增加10的频率上升，最大为200。



步骤7 查看结果

1. 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
2. 在左侧导航栏选择“边缘节点 > 节点管理”，选择之前创建的边缘节点“test_loT10”，单击节点进入详情页。
3. 选择“边缘设备”页签，单击添加边缘设备中添加的设备ID“test_loT10”，跳转至“设备接入服务控制台”查看采集的数据。
4. 上报的数据如下图所示，数据源上报的值为100、110、120...，平台收到的值为100、120...，即值110由于 $(110-100) \div 100 \leq 10\%$ ，没有超过偏差区间，则被清洗。



----结束

6.2.9 (可选) 点位阈值

增加点位清洗配置，在基于IoT边缘实现OT数采章节的配置上，可以选择增强功能“高级配置 > 点位清洗”。

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_IoT10”进入页面。
- 步骤3** 在左侧导航栏选择“数采配置 > 选择对应的数采连接”。
- 步骤4** 选择已有点位MotorCurrent，操作列--单击修改，勾选“点位阈值”选项，配置最大值为10，最小值为0，单击“确定”。

图 6-12 点位阈值修改

修改点位映射

点位数据... int(整型) * 设备属性 Motor/Current

采集配置

* 点位地址 ns=3;i=1002 * 点位值的... int(整型)

* 点位采集... 1000 毫秒

高级配置 ▲ 点位拆分|点位缩放|点位清洗|点位阈值 ?

点位拆分 将整型值按 bit 位拆分,映射到不同属性 ?

点位缩放 通过线性表达式将采集值转化为真实值 ?

点位清洗 数据去重、数据波动抑制 (支持数字类型) ?

点位阈值 仅数据类型为 int 时,呈现点位阈值 ?

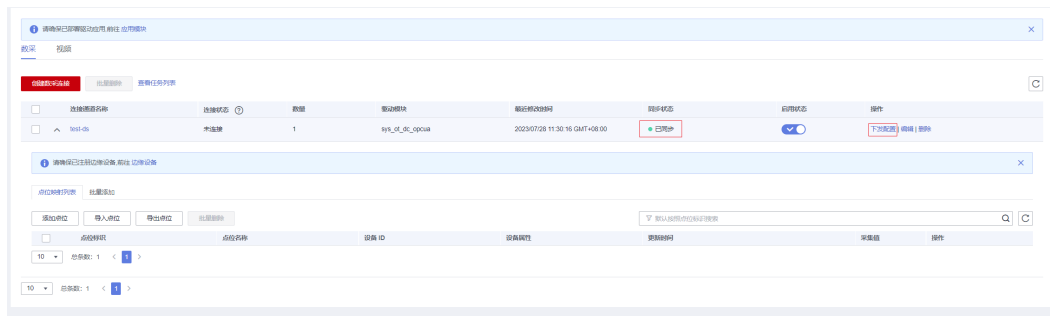
* 最小值 0

* 最大值 10

确认 取消

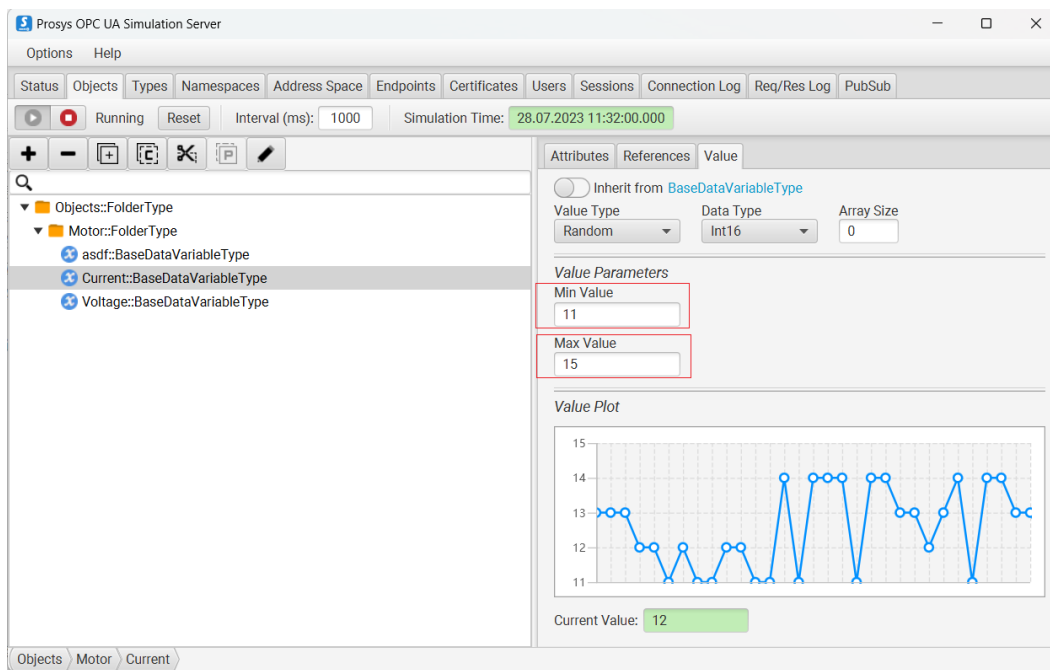
- 步骤5** 修改点位配置后，单击“下发配置”，将配置更新给边侧。

图 6-13 同步下发配置



步骤6 修改数据源配置，为了更清晰的看到点位阈值的效果，将数据源的配置进行修改，对点位MotorCurrent对应的数据源地址的值进行修改，保持上报值11~15。

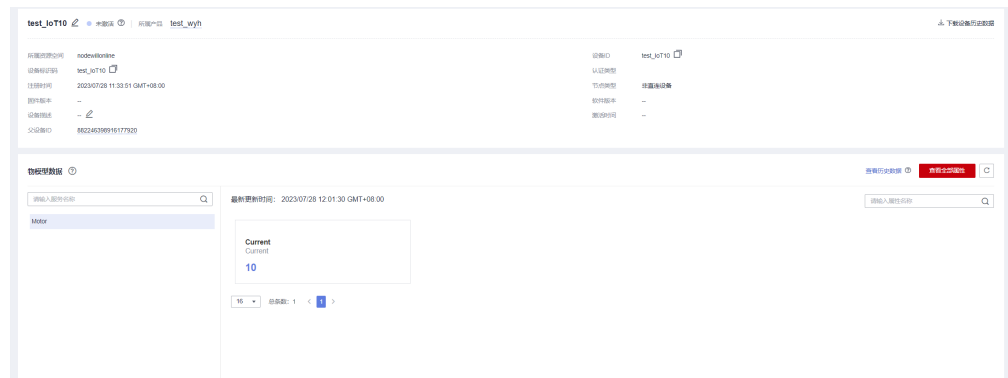
图 6-14 点位上报值修改



步骤7 查看结果。

1. 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
2. 在左侧导航栏选择“边缘节点 > 节点管理”，选择之前创建的边缘节点“test_loT10”，单击节点进入详情页。
3. 选择“边缘设备”页签，单击添加边缘设备中添加的设备ID“test_loT10”，跳转至“设备接入服务控制台”查看采集的数据。
4. 上报的数据如下图所示，数据源上报的值为11~15之间的随机值，平台仅收到了调整前收到的10，超出0~10范围范围的异常值都被过滤掉了

图 6-15 平台查看上报数据



----结束

7 基于 IoT 边缘实现模拟 IEC104 网关的接入

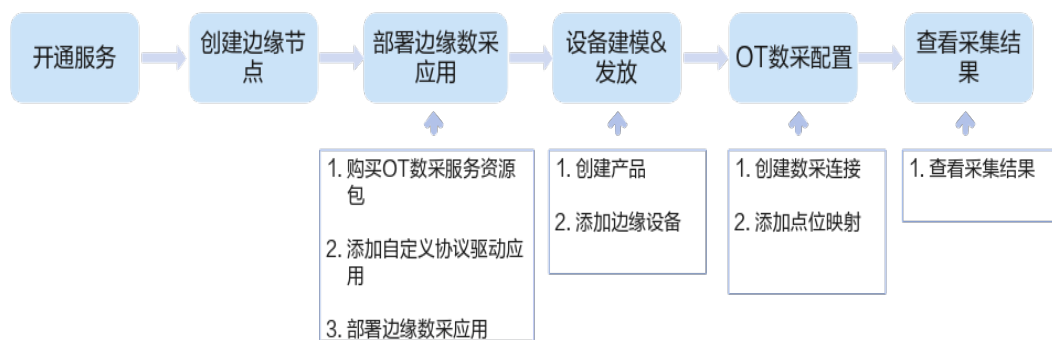
7.1 实验简介

OT数采配置，提供了在IoT边缘控制台简单配置，实现OT数采（DCS、PLC、OPCUA等）。

通过本实验，您将能够：

- 了解并掌握IoT边缘服务的功能特点与使用方法。
- 掌握将OT设备的数据上报云端的方法。

OT 数采流程介绍



7.2 操作步骤

7.2.1 创建边缘节点

前提条件

- 已注册[华为云](#)账号，且已完成实名认证。
- 已开通[IoT边缘服务](#)和[IoTDA设备接入服务](#)。
- 在创建边缘节点前，检查配置要符合[硬件规格](#)要求，如磁盘、内存和docker版本，切换root用户执行安装命令。

注册专业版节点

- 步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2 选择左侧导航栏“边缘节点 > 节点管理”单击右上角“注册节点”。
- 步骤3 根据页面提示填写参数，勾选右下角“《华为云服务等级协议》”单击“下一步”配置节点数据。

图 7-1 专业版

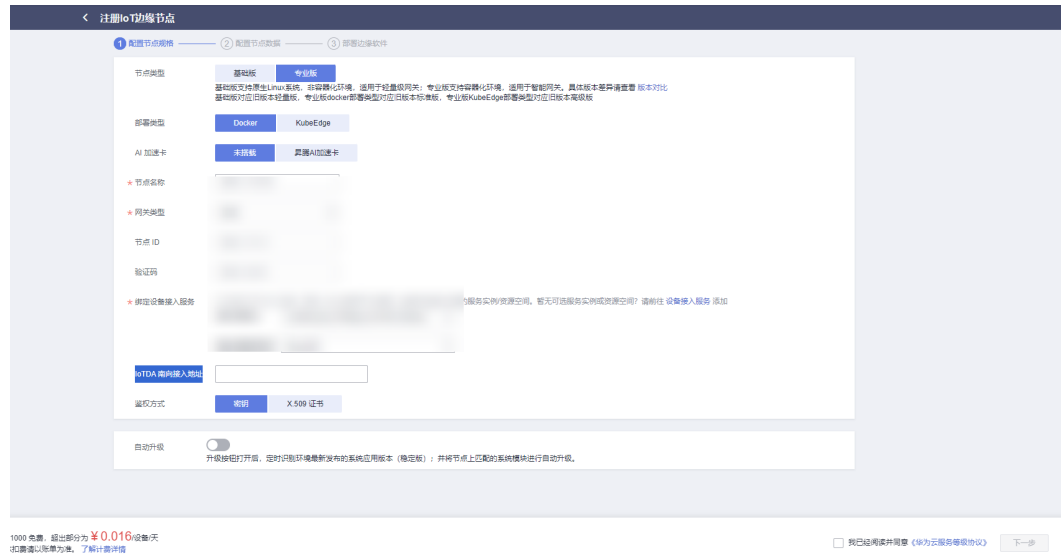


表 7-1 节点规格参数说明

参数名称	参数说明
节点类型	专业版。
部署类型	Docker。
AI加速卡	<ul style="list-style-type: none"> 未搭载：边缘节点未使用AI加速卡时选择。 昇腾AI加速卡：支持华为昇腾AI加速设备，边缘节点的服务器搭载相关昇腾AI加速设备时选择。
昇腾驱动地址	可选，AI加速卡选择昇腾AI加速卡时需要保证安装边缘节点的服务器上已经安装昇腾设备驱动，并根据前端提示填入驱动地址。若改值为空，默认填入Atlas300驱动默认安装地址： /usr/local/Ascend/driver/lib64/。
节点名称	自定义节点名称，如“test”。 说明 节点名称可以在：“节点概览”-“基本信息”中编辑修改。
网关类型	other。

参数名称	参数说明
节点ID	自定义（可选填写）。只允许中文，英文字母、数字、中划线（-），下划线（_），最小长度1，最大长度64。 说明 边缘节点ID自定义(不输入默认生成)。
验证码	输入自定义验证码（可选填写）。验证码，执行安装边缘软件命令时与云端通讯的凭证。
设备归属实例	选择默认归属实例。
设备归属资源空间	选择默认资源空间，或为本实验单独新建一个资源空间，如 edge。
IoTDA南向接入地址	选择网关归属的IoTDA实例的南向设备mqtt接入地址，也可以选择自定义的南向接入地址，支持“ip:端口”或“域名:端口”形式。
鉴权方式	密钥：默认生成。 X.509证书：请参考： 基于MQTT.fx的X.509证书接入指导 。 警告 鉴权方式为“密钥”的节点，请勿在IoTDA界面重置节点的密钥，重置密钥会导致网关与IoTDA通信异常，且不可恢复。
指纹	证书文件生成的字符串（可选填写）。只能为40位十六进制字符串或64位六进制字符串。
本地路径	输入证书文件和密钥文件在设备上的存储路径，只允许字母、数字、下划线（_）和斜线（/），以/开头并且以.pem或.crt结尾。
自动升级	可选，开启后节点将自动升级系统应用模块。 注意 升级过程中可能导致节点数据上传中断。

步骤4 填写配置节点数据信息后，单击“立即创建”。

图 7-2 配置节点数据

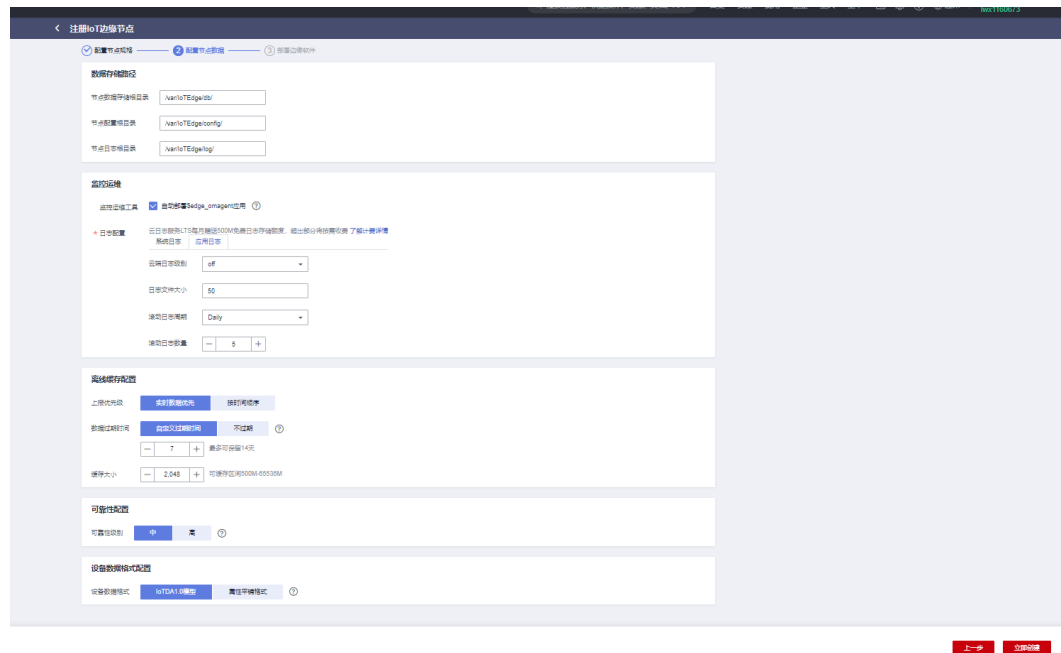


表 7-2 专业版节点信息

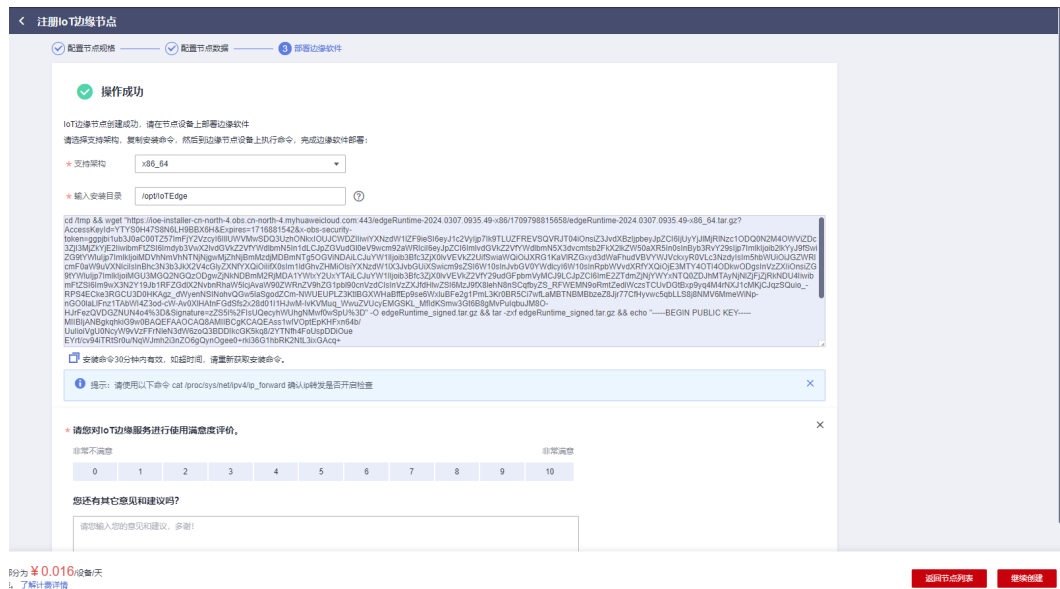
参数名称	说明
数据存储路径	默认（允许修改存储路径）。
监控运维	默认勾选“监控运维工具”。
\$edge_omagent应用	负责边缘节点远程监控和运维，提供日志上报、远程SSH、CPU等指标上报、文件上传下载能力。

参数名称	说明
日志配置	<p>系统日志：边缘节点上已部署的系统应用产生的日志。 应用日志：边缘节点上已部署的用户定义应用所产生的日志。</p> <ul style="list-style-type: none"> • 云端日志级别：on, off, trace, debug, info, warn, error, fatal。其中，on表示上传所有日志至云端，off表示不上传日志至云端，其余可选项代表按日志级别采集，选择后将只采集目标级别及以上的日志（模糊匹配）。日志级别的严重程度等级由低到高依次为：trace, debug, info, warn, error, fatal。 • 日志文件大小：日志文件大小限制，单位MB，默认50，取值范围10-1000。某个日志文件如果达到大小限制，则会将文件转储保存至日志目录下。 • 滚动日志周期：日志转储周期，可选项：daily, weekly, monthly, yearly。某个日志文件如果达到转储周期，则会将文件转储保存至日志目录下。日志文件大小和滚动日志周期是同时生效的，满足任何一个条件都会进行日志转储。 • 滚动日志数量：日志文件转储个数，默认5，取值范围1-10。边缘节点保存的转储日志数量如果达到限制，则会删除最早的转储文件。 <p>注意</p> <ol style="list-style-type: none"> 1. 当前“云端日志级别”对日志等级采取模糊匹配的策略，若日志消息中存在与日志级别标识相同的关键字，则也可能被采集。 2. 若要开启“云端日志级别”中的trace, debug, info, warn, error, fatal选项，则必须满足以下三个前置条件： <ol style="list-style-type: none"> 1. 目标节点是基础版节点或部署类型为Docker的专业版节点。需注意，部署类型为KubeEdge的专业版节点不支持此功能。 2. 已安装\$edge_omagent应用，且版本不低于1-1-15-xxx。 3. 应用日志需遵循如下规范打印，以便进行过滤： <ul style="list-style-type: none"> ○ 日志级别标识需打印为（大小写均可）：trace、debug、info、warn、error、fatal。 ○ 每条日志消息以YYYY-MM-DD日期格式作为前缀。 <p>说明 专业版（部署类型为Docker）和基础版节点只涉及日志级别这一个参数，其他参数只对专业版（部署类型为KubeEdge）有效。</p>
离线缓存配置	<p>可对离线节点上报的数据进行缓存，定义缓存周期（可配置为不过期）跟缓存大小、上报的优先级、路由转发是否开启离线缓存。仅当上报优先级为“按时间顺序”时支持路由转发开启离线缓存。</p>
可靠性配置	<p>可靠性级别分为“中”和“高”2种。中级别：当应用上报属性和点位的频率超过流控值时，超过部分会被丢弃；高级别：当应用上报属性和点位的频率超过流控值时，通过控制模块的发送速度，降低上报频率，报文不会被丢弃。（注意：高可靠性级别的配置需要定制应用的配合，同时会造成性能的损失）。</p>

参数名称	说明
设备数据格式配置	<p>设备的数据格式：即单网关节点中设备的标准数据格式，包含 IoTDA1.0物模型格式和属性平铺格式。IoTDA1.0物模型中包含了设备、服务和属性三层结构，具体参考IoTDA网关子设备属性上报报文格式。属性平铺格式，包含设备、属性2层结构，属性结构可以包含自定义结构的拼接，在json中作为一项平铺展开，具体格式请参考步骤6设备数据格式配置。</p> <p>须知 属性平铺格式功能，需要edge_hub应用1-1-38-xx版本及以上支持</p>

步骤5 弹出如下对话框，您可以“返回节点列表”也可以“继续创建”。

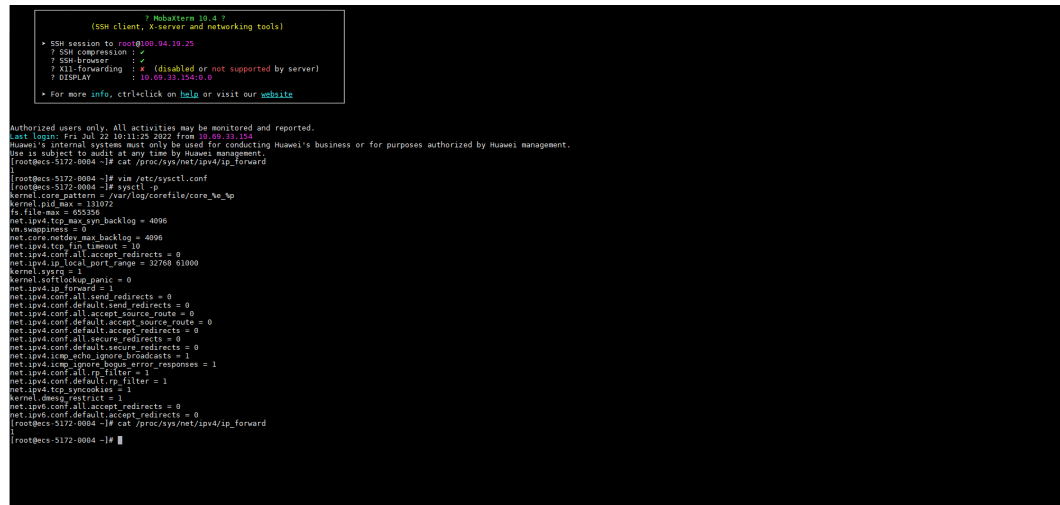
图 7-3 注册成功



说明

安装部署节点前，请确认机器的IP转发已开启并且已开启自启动，以及7883和8943端口开放。其中，IP转发规则配置如下：

图 7-4 IP 转发规则配置



1. 检查机器ip_forward in/proc文件系统的值。
`cat /proc/sys/net/ipv4/ip_forward`
 若值等于0，则需要添加配置IP转发规则。
 2. 检查配置文件/etc/sysctl.conf是否包含net.ipv4.ip_forward = 1。
 如果配置文件中没有net.ipv4.ip_forward参数项，请新增一行net.ipv4.ip_forward = 1。
`vim /etc/sysctl.conf`
 按“i”进入编辑模式，修改完成后按“Esc”退出编辑模式，然后按“Shift” + “;”组合键，输入wq并回车，保存修改并退出。
 3. 重新加载此文件的值。
`sysctl -p`
 4. 检查7883和8943端口是否开放，如果是华为云服务器，请参考：[安全组](#)。
- 结束

安装专业版节点

步骤1 在“节点管理”的节点列表选择您需要安装的节点，单击右侧“安装”，获得提示。

图 7-5 安装

节点名称	节点类型	资源包类型	资源包ID	状态	主机名/网络	操作
	专业版	--	--	未安装		删除 安装
	专业版	--	--	离线	172.30.7.64,172.17.0.1,1...	删除
	专业版	--	--	已安装		删除


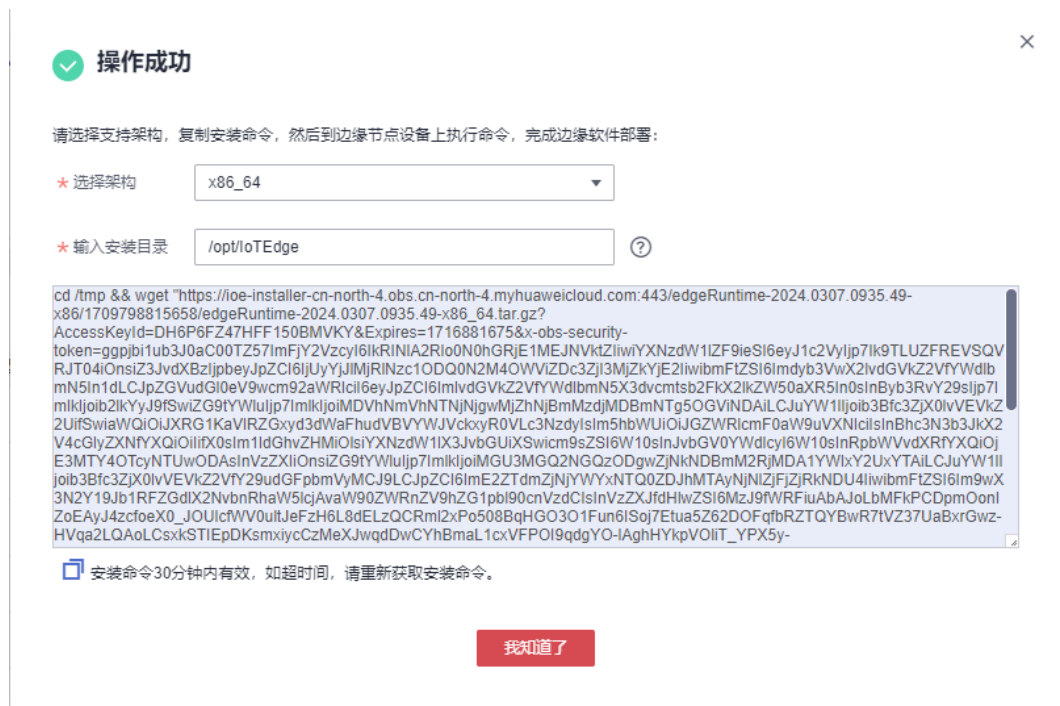
步骤2 选择支持架构和安装目录，然后单击  复制安装命令，使用SSH工具以root用户登录边缘节点服务器后台系统，执行安装命令。

图 7-6 提示



说明

安装命令的选择需要与购买的ecs的架构保持一致。

图 7-7 执行安装

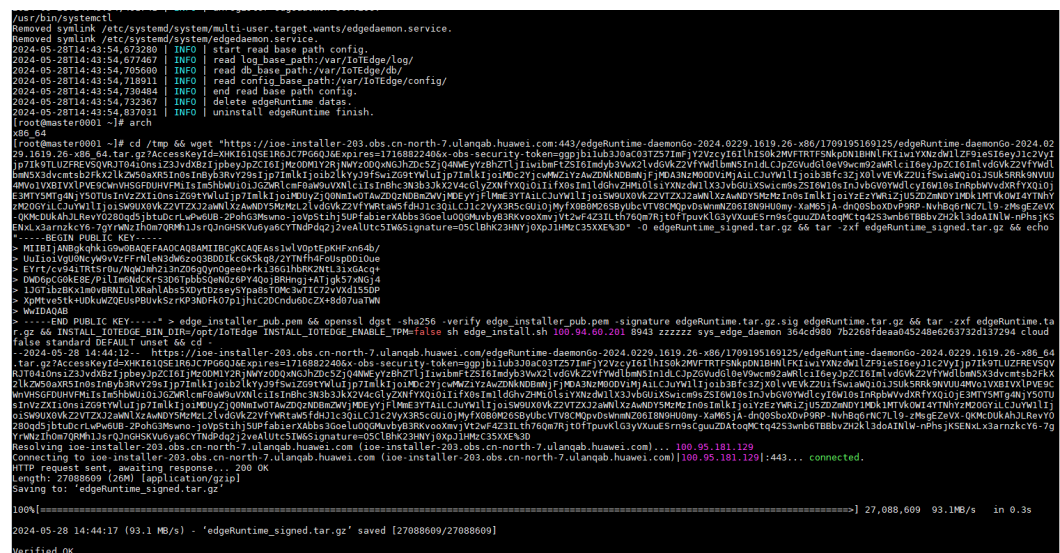


图 7-8 安装命令执行成功

```

2024-05-28T14:44:29,581321 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:PENDING, reason:(zero) ]
2024-05-28T14:44:31,632185 | INFO | saveBasePathInfo | 78 | config base path start.
2024-05-28T14:44:31,632291 | INFO | saveBasePathInfo | 83 | config log_path:/var/IoTEdge/log/, config_path:/var/IoTEdge/config/
2024-05-28T14:44:31,632457 | INFO | saveBasePathInfo | 95 | config base path success.
2024-05-28T14:44:31,632535 | INFO | addAgentCustomConfigs | 261 | addCustomConfigs start.
2024-05-28T14:44:31,632603 | INFO | addAgentCustomConfigs | 285 | addCustomConfigs end.
2024-05-28T14:44:31,632654 | INFO | pullImage | 319 | start pull image, deploy type is cloud
2024-05-28T14:44:35,888319 | INFO | createModule | 76 | create Module, name:sys_edge_agent
test-tpm-2-x86, containerId:4b3e43ce7eac9bc282b094377abfb148a284389a36856f3cacc0338743036e, processID:(zero), type:docker,
2024-05-28T14:44:35,898163 | INFO | startModule | 160 | startModule moduleId:sys_edge_agent
2024-05-28T14:44:42,390394 | INFO | listModule | 259 | listModule
2024-05-28T14:44:42,440775 | INFO | checkModulesbo | 340 | start check modules, time:1
2024-05-28T14:44:42,440808 | INFO | listModule | 259 | listModule
2024-05-28T14:44:42,491163 | INFO | printModulesAndReport | 404 | ***success:1 failed:0
2024-05-28T14:44:42,491339 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:RUNNING, reason:(zero) ]
2024-05-28T14:44:42,542192 | INFO | reportEvent | 158 | Begin reporting event: {"events":[{"resource":"node.installation","location":"zzzzz","event":"succeeded","event_severity":"Info",event_time":"20240528T064442Z","notify_data":{"host_tag":"DEFAULT","edge_node_id":"zzzzz"}}]}
2024-05-28T14:44:42,529233 | INFO | reportModuleStatus | 138 | reportModuleStatus, modulesInfo:ModuleStatusInfoList[ host_tag:DEFAULT, deploy_id:1716878669308, modules:[ModuleStatusInfo module_id:sys_edge_agent, app_id:sedge_agent, app_version:test-tpm-2-x86, container_state:RUNNING, liveness_state:(zero), update_time:1716878669305,]]]
2024-05-28T14:44:43,643089 | INFO | checkBridgeNetwork | 508 | begin checking bridge network
2024-05-28T14:44:50,524137 | INFO | install | 494 | install edge finish
2024-05-28T14:44:50,524559 | INFO | stop | 545 | edgeinstaller stop.
/root
    
```

步骤3 单击“我知道了”，等待边缘节点的状态变为“在线”，表示该节点已安装并连接成功。

步骤4 您可以单击“节点名称”，查看该节点详细信息，关于节点详情描述可参见[节点概览](#)。

----结束

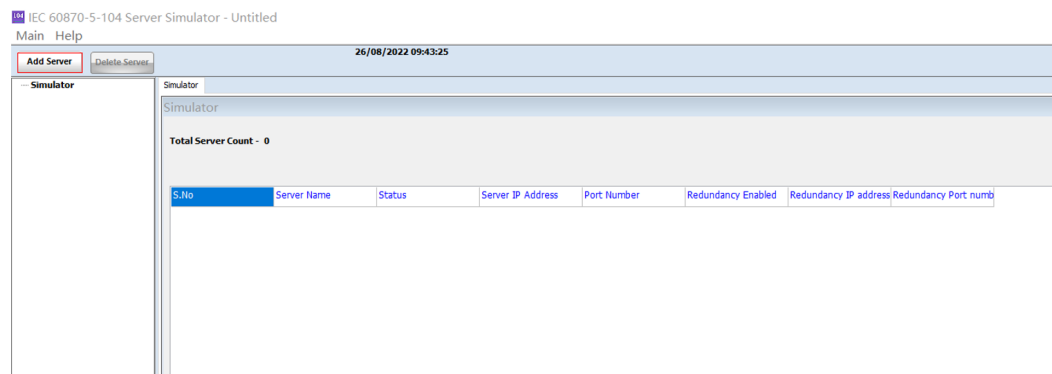
7.2.2 准备数据源

安装 IEC104 Server Simulator 模拟器

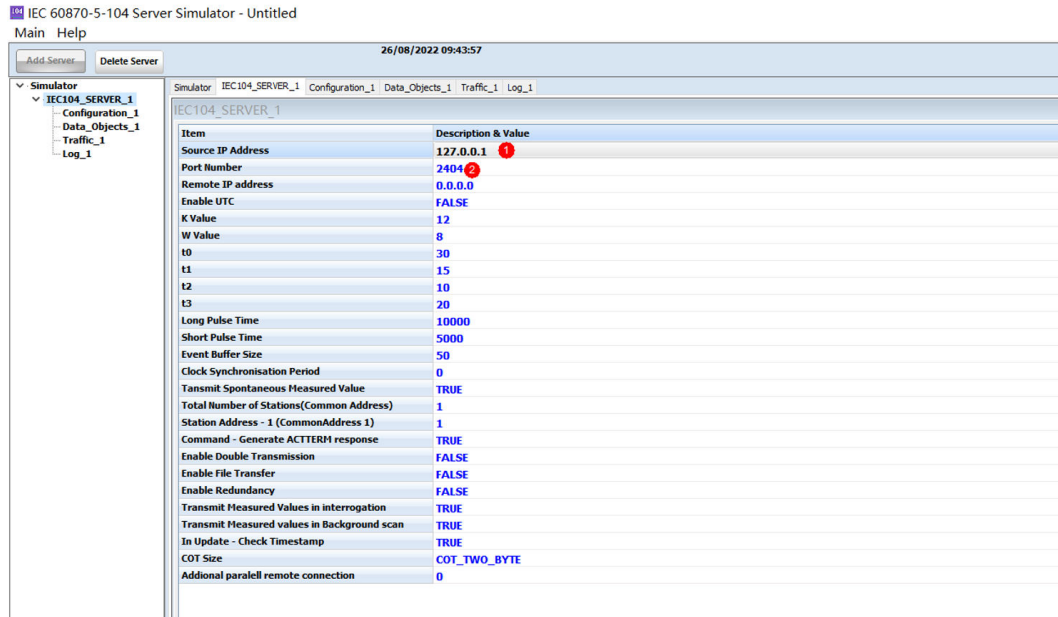
步骤1 下载[IEC104 Server Simulator模拟器](#)。

步骤2 安装完毕后，启动IEC104 Server Simulator模拟器

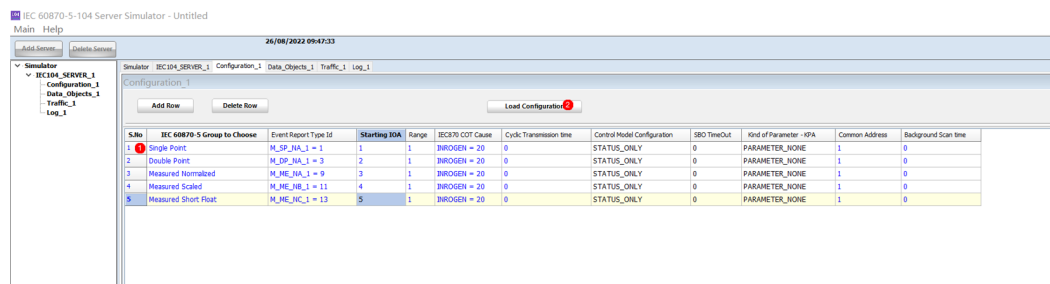
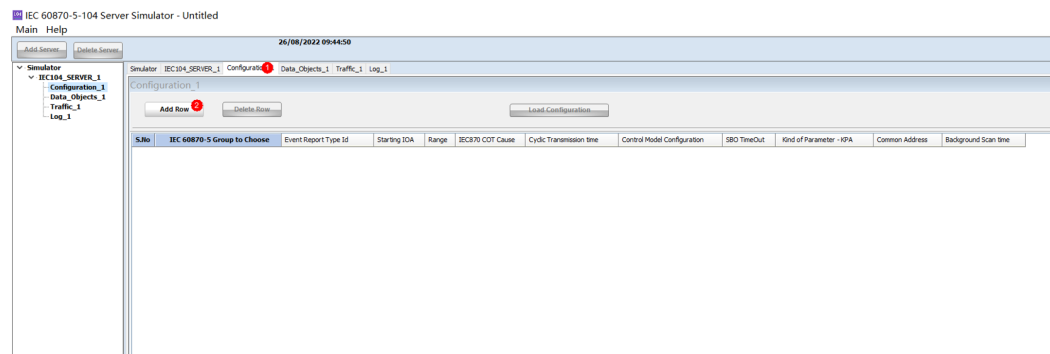
步骤3 单击Add Server 添加server信息



步骤4 修改server的ip地址和端口号（改为模拟器所在的设备的IP地址，确认该设备的端口正常开放）。

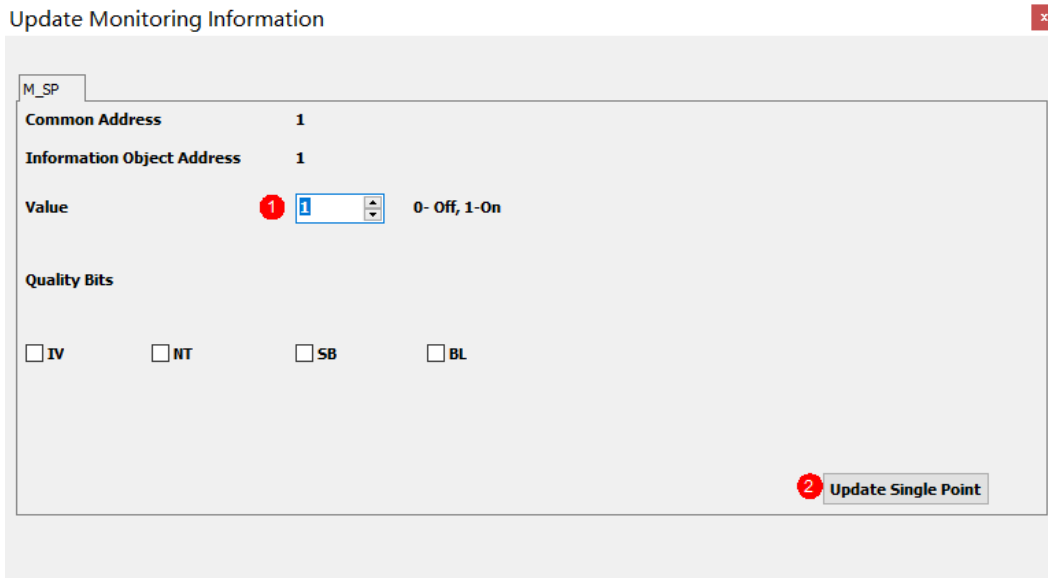


步骤5 在模拟器上添加采集点的信息，信息添加完成后载入配置。



步骤6 打开模拟器，右键对应的数据值进行数据更新

S.No	Common Address	Event Report Type ID	IOA	Value	Quality Bits	Time Stamp	IEC870 COT Cause	Control Model	SBO TimeOut	Kind of Parameter - KPA	Mapped Point CA	Mapped Point Type ID	Mapped Point IOA	Toggle with
1	M_SP_NA_1	1	0	GD	09:48:12:614 - 26/08/2022	INTROGEN	STATUS_ONLY	0	PARAMETER_NONE	0	0	0	0	
2	M_DP_NA_1	2	0	GD	09:48:12:614 - 26/08/2022	INTROGEN	STATUS_ONLY	0	PARAMETER_NONE	0	0	0	0	
3	M_ME_NA_1	3	0	GD	09:48:12:614 - 26/08/2022	INTROGEN	STATUS_ONLY	0	PARAMETER_NONE	0	0	0	0	
4	M_ME_NB_1	4	0	GD	09:48:12:614 - 26/08/2022	INTROGEN	STATUS_ONLY	0	PARAMETER_NONE	0	0	0	0	
5	M_ME_NC_1	5	0	GD	09:48:12:614 - 26/08/2022	INTROGEN	STATUS_ONLY	0	PARAMETER_NONE	0	0	0	0	



----结束

7.2.3 部署边缘数采应用

部署边缘数采应用

如未添加边缘数采应用，可参考[添加驱动应用](#)。

步骤1 在左侧导航栏选择“边缘节点 > 节点管理”“选择之前创建的边缘节点“test_loT10”，单击“详情”进入节点详情页。

步骤2 选择左侧“应用模块 > 模块管理”，单击“部署应用”填写基本信息后“确认”。

- 功能用途: 数据采集
- 边缘应用: iec104
- 选择版本: 当前最新的版本



步骤3 边缘应用部署任务下发成功后，应用状态为已停止。

模块ID	模块名称	所属应用	版本	应用类型	实例状态	操作
sys_edge_agent	sys_edge_agent	Sedge_agent	1.526.x86	系统必选	运行中	升级
sys_edge_hub	sys_edge_hub	Sedge_hub	2022-07-19-53-standard-x86	系统必选	运行中	升级
sys_edge_omagent	sys_edge_omagent	Sedge_omagent	2-3-41-cpu-lite-x86_B4	系统可选	运行中	删除 升级
user_iec104	user_iec104	iec104	0.0.19-dev	用户定义	已停止	删除 升级

说明

数采应用部署成功后或者升级数采应用之后，应用状态都显示为已停止，只有在数采配置启用了数采连接后，应用状态会变更为运行中。

---结束

7.2.4 设备建模&发放

创建产品

使用物联网平台的第一步就是在控制台创建产品。产品是设备的集合，是指某一类具有相同能力或特征的设备的合集被称为一款产品。

- 步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2 在左侧导航栏选择“边缘节点 > 设备建模”进入页面。
- 步骤3 单击右上角“创建产品”，按照页面提示填写信息，然后单击“确定”。

创建产品 ×

* 所属资源空间 ?

* 产品名称

协议类型

* 数据格式 ?

厂商名称

* 设备类型 ?

参数名称	说明
所属资源空间	与之前创建边缘节点时选择的资源空间保持一致。

参数名称	说明
产品名称	自定义，如test_loT10。
协议类型	Other 说明 使用OT数采能力时，建模协议类型必须选择Other。
数据格式	默认值，JSON。
厂商名称	自定义，如Huawei。
设备类型	自定义。

步骤4 单击“设备建模 > 产品”，选择刚才创建的产品“test_loT10”，单击详情进入产品详情页面。

步骤5 在“模型定义”页签下，单击“自定义模型”。



📖 说明

支持将本地开发的产品模型导入到平台。

导入步骤：选择“设备建模 > test_loT10”，单击“上传模型文件”，在弹窗中添加模型文件，单击“确定”完成上传。



步骤6 填写“服务ID”、“服务类型”，单击“确定”。

- 服务ID：自定义，如service0
- 服务类型：自定义，如service0

添加服务



* 服务ID

服务类型 ?

服务描述
0/128

步骤7 展开服务列表，单击“添加属性”，根据实际需要分别添加单点信息、双点信息等属性。

修改属性



* 属性名称

属性描述
0/128

* 数据类型

* 访问权限

修改属性



* 属性名称

属性描述
0/128

* 数据类型

* 访问权限 可读 可写

枚举值
0/1,024

确定

取消

修改属性

✕

* 属性名称

属性描述
0/128

* 数据类型

* 访问权限 可读 可写

* 取值范围 -

步长

单位

步骤8 模型定义完成，参数信息预览如下图所示。



----结束

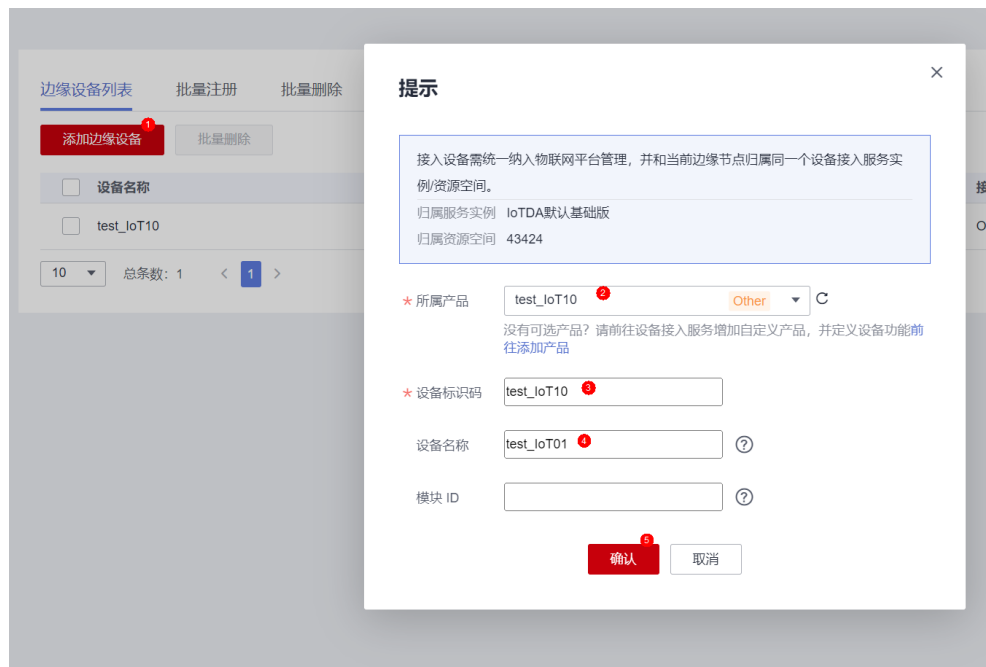
添加边缘设备

用户创建产品后，需要在边缘节点上添加对应的边缘设备，用于接收边缘节点实际采集的OT设备数据。

步骤1 在左侧导航栏选择“边缘节点 > 节点管理”选择之前创建的边缘节点“test_loT10”，单击节点进入详情页。

步骤2 选择左侧导航栏“边缘设备”，单击“添加边缘设备”，根据提示填写参数后，单击“确认”。

图 7-9 创建设备



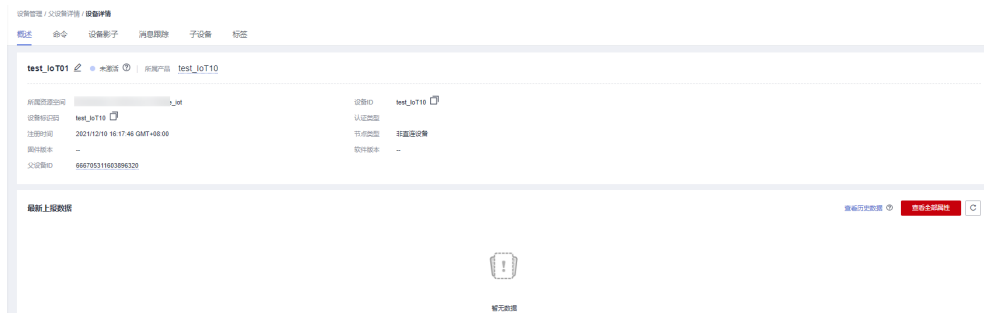
参数名称	说明
所属产品	选择步骤3创建的产品“test_loT10”。
设备标识码	自定义。只允许英文字母、数字，下划线，中划线.最大长度64。
设备名称	自定义。设备名称长度为1~256位，只能由数字,字母,中文,逗号,括号,#,.,,?,!,-,_,@,%,&,!组成。
模板ID	为空。 说明 使用Other数采能力设备必须选择协议类型为Other的产品，且模块id必须为空。

步骤3 添加设备成功后，可以在设备列表中查看“test_loT10”设备状态，并管理边缘设备。此时设备处于未激活状态，需完成设备部署接入后，可在物联网平台查看设备上报的数据。

图 7-10 设备列表



单击对应的“设备ID”跳转至“设备接入服务控制台”进行设备管理。



----结束

7.2.5 OT 数采配置

添加数采模板

描述数采配置项的元数据，数采配置包括数据源配置元数据、点位采集配置元数据。

系统数据源模板导入

数据源模板描述了数据源配置项的元数据，用户在配置数据源之前，需要先导入数据源模板。IoT边缘服务当前已经预置通用数据源模板，可以直接使用，系统也支持用户自定义导入模板。

步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。

步骤2 在左侧导航栏选择“边缘节点 > 应用管理”进入页面，

步骤3 选择“驱动应用 > 数采模板管理 > 添加数采模板”。



步骤4 添加文件，单击“确认”。

- 本地准备一份数采模板文件，文件内容合规。
- IoT边缘云平台当前已经预制2个业界主流通用数采模板，可以通过导入通用模板后使用。
- 支持用户自定义导入模板，导入模板后缀.json，满足json格式规范，样例及字段解释如下：

示例

```
{
  "tpl_id": "iec104_template",
  "name": "IEC104通用数采模板",
  "description": "IEC104通用数采模板",
  "datasource_meta": {
    "config_tabs": [
```

```
{
  "key": "connection_info",
  "name": "连接信息",
  "description": "连接信息",
  "config_items": [
    {
      "key": "host",
      "name": "服务地址",
      "description": "受控站服务地址",
      "data_type": "string",
      "required": true,
      "crypted": false,
      "max_length": 15,
      "example": "127.0.0.1"
    },
    {
      "key": "port",
      "name": "服务端口",
      "description": "受控站监听端口",
      "data_type": "int",
      "required": true,
      "crypted": false,
      "max_length": 15,
      "example": "2404"
    }
  ]
},
{
  "key": "collection_paras",
  "name": "连接附加参数",
  "description": "连接附加参数",
  "config_items": [
    {
      "key": "timeout_t0",
      "name": "连接建立超时时间t0 (秒)",
      "description": "Time-out of connection establishment",
      "data_type": "int",
      "required": false,
      "crypted": false,
      "max_length": 128,
      "example": "30"
    },
    {
      "key": "timeout_t1",
      "name": "APDU发送超时时间t1 (秒)",
      "description": "Time-out of send or test APDUs",
      "data_type": "int",
      "required": false,
      "crypted": false,
      "max_length": 128,
      "example": "15"
    },
    {
      "key": "timeout_t2",
      "name": "消息确认超时时间t2 (秒)",
      "description": "Time-out for acknowledges in case of no data messages t2 < t1",
      "data_type": "int",
      "required": false,
      "crypted": false,
      "max_length": 128,
      "example": "10"
    },
    {
      "key": "timeout_t3",
      "name": "心跳数据超时时间t3 (秒)",
      "description": "Time-out for sending test frames in case of a long idle state",
      "data_type": "int",
      "required": false,
      "crypted": false,

```

```
        "max_length": 128,
        "example": 20
    },
    {
        "key": "send_seq_depth",
        "name": "发送队列深度k",
        "description": "Maximum difference receive sequence number to send state variable",
        "data_type": "int",
        "required": false,
        "crypted": false,
        "max_length": 128,
        "example": 12
    },
    {
        "key": "rcv_seq_depth",
        "name": "接收队列深度w",
        "description": "Latest acknowledge after receiving w l format APDUs",
        "data_type": "int",
        "required": false,
        "crypted": false,
        "max_length": 128,
        "example": 8
    }
]
},
"default_values": {
    "drivername": "IEC104"
}
},
"point_meta": {
    "config_items": [
        {
            "key": "common_address",
            "name": "公共地址(dec)",
            "description": "十进制公共地址",
            "data_type": "ushort",
            "required": true,
            "crypted": false,
            "max_length": 256,
            "example": 1
        },
        {
            "key": "point_address",
            "name": "点位地址(dec)",
            "description": "十进制点位地址",
            "data_type": "ushort",
            "required": true,
            "crypted": false,
            "max_length": 256,
            "example": 1
        },
        {
            "key": "point_type_id",
            "name": "点位数据类型",
            "description": "协议所规范的点位类型",
            "data_type": "string",
            "required": true,
            "crypted": false,
            "max_length": 256,
            "example": "M_SP_NA_1"
        }
    ]
}
}
```

表 7-3 OT 数采模板

key	类型	描述
tpl_id	String	数采模板标识，租户下唯一 pattern: '^([a-zA-Z0-9_-]*)\$' 1-64
name	String	数据源模板名称，支持中英文 1-64
description	String	数据源模板描述，支持中英文，0-128
datasource_meta	Object	数据源配置元数据
point_meta	Object	点位采集配置元数据

表 7-4 datasource_meta

key	类型	描述
config_tabs	Array	数据源配置表格列表
default_values	Object	数据源默认的值，通常用于描述默认的驱动协议

表 7-5 config_tabs

key	类型	描述
key	String	数据源配置表格键值，pattern: 'connection_info collection_paras'
name	String	数据源配置表格名称，1-64
description	String	描述，0-255
config_items	Array	配置项列表

表 7-6 config_items

key	类型	描述
key	String	配置项key，1-32
name	String	默认项名称,1-64
description	String	配置项描述，0-128

key	类型	描述
data_type	String	配置项类型，支持short ushort int int64 uint long ulong float double bool string object decimal。
required	Boolean	配置项是否必传项
encrypted	Boolean	配置项是否加密
max_length	Integer	当data_type= string时配置有效，校验输入的字符串最大长度
example	String	配置项举例，值在界面输入框作为示例灰色显示，0-256

表 7-7 default_values

key	类型	描述
drivername	String	用于描述默认的驱动协议，如 OPCUA。

----结束

创建数采连接

- 步骤1** 访问[IoT边缘](#)，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_IoT10”进入页面。
- 步骤3** 在左侧导航栏选择“数采配置 > 创建数采连接”。



- 步骤4** 填写“数采连接参数”后单击“确认”保存，此时配置仅在云平台存在，未下发到边缘，此时数采连接的“同步状态”为“待同步”。
 - 通道标识：iec104
 - 通道名称：iec104（自定义）
 - 驱动模块：user_iec104
 - 受控端地址：模拟器所在的设备的ip地址

- 受控端端口：默认2404，可以根据实际使用填写（与模拟器端口号对应）
- 连接附加参数：根据实际情况填写，无参数写入时默认使用协议默认参数

📖 说明

服务端点：填写IEC104受控端提供的连接地址。本实验[准备数据源](#)中使用IEC104 Server Simulator来模拟一个IEC104受控端，受控端地址与端口号填写该模拟器的url。

创建数采连接

通道信息

* 通道标识

* 通道名称

配置数据源

* 驱动模块 暂无?前往 [应用管理](#)

连接信息

* 服务地址

* 服务端口

连接附加参数

连接建立超时时间t0 (秒)

APDU发送超时时间t1 (秒)

消息确认超时时间t2 (秒)

心跳数据超时时间t3 (秒)

发送队列深度k

...

---结束

添加点位映射

将点位上报转换为设备属性上报，将设备属性读写转换为点位读写操作，将设备命令转化为点位写操作。

- 步骤1** 访问**IoT边缘**，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_IoT10”进入页面。
- 步骤3** 在左侧导航栏选择“数采配置 > 选择对应的数采连接”。
- 步骤4** 单击“添加点位”在“添加点位映射”弹窗中填写配置信息，如**表7-8**所示，然后单击“确认”。

表 7-8 添加点位映射

基本信息	
点位标识	点位的唯一标识。
点位名称	按照规范填写。
设备ID	边缘设备页签里添加设备的ID。

基本信息		
点位数据类型	当前支持如下数据类型： int(整型) uint(无符号整型) int64(64 位整型) short(短整型) ushort(无符号短整型) long(长整型) ulong(无符号长整型) bool(布尔类型) float(单精度浮点型) double(双精度浮点型) decimal(小数) string(字符串) object(对象类型)	取值范围： -2147483648~2147483647 0~4294967295 -2147483648~2147483647 -32768~32767 0~65535 -2147483648~2147483647 0~4294967295 0/1 -3.4*10 ³⁸ ~+3.4*10 ³⁸ -1.7*10 ³⁰⁸ ~1.7*10 ³⁰⁸ -1.7*10 ³⁰⁸ ~1.7*10 ³⁰⁸
设备属性	设备属性由物模型中service_id/property_name格式构成。	
采集配置		
公共地址	对应IEC104协议中的Common Address，即从站地址	
点位地址	对应IEC104协议中的Information Object Address，即信息体地址	
点位数据类型	对应IEC104协议中数据上报时报文帧的TypeId	

采集配置部分点位数据类型支持的几种点位类型如下表：

表 7-9 点位数据类型配置

点位类型	对应数据类型
M_SP_NA_1	单点信息，布尔值
M_DP_NA_1	双点信息，枚举值
M_ME_NA_1	归一化测量值，浮点数值
M_ME_NB_1	标度化测量值，整型
M_ME_NC_1	短浮点数值

 注意

不同点位请勿配置同一设备的相同属性。

步骤5 单击“下发配置”，把点位配置下发至边缘侧。配置下发成功后启用通道，可以看到通道连接状态变为“已连接”，此时数采连接的“同步状态”为“已同步”。



说明

- 修改点位信息后需重新单击下发配置，配置才会生效。
- 模块的运行状态由启用状态决定，当启用了数采连接后，模块会显示为运行中。

----结束

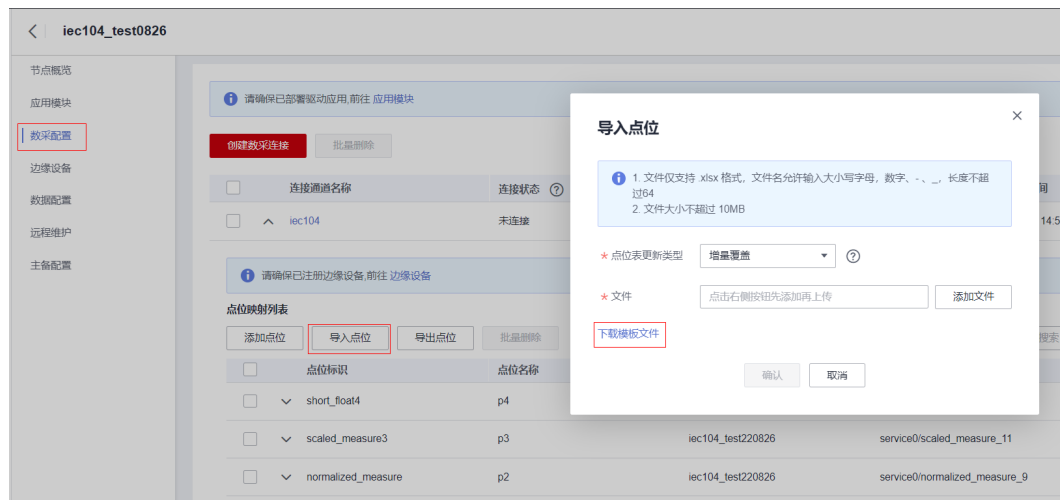
附录：批量导入/导出点位映射

批量导入点位映射

步骤1 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_loT10”进入页面。

步骤2 在左侧导航栏选择“数采配置”->对应的连接通道。

步骤3 单击“导入点位”单击“下载模板文件”，打开“模板文件”内容填写。



步骤4 选择“添加文件”导入步骤3下载的文件后，单击“确认”完成批量导入点位映射。



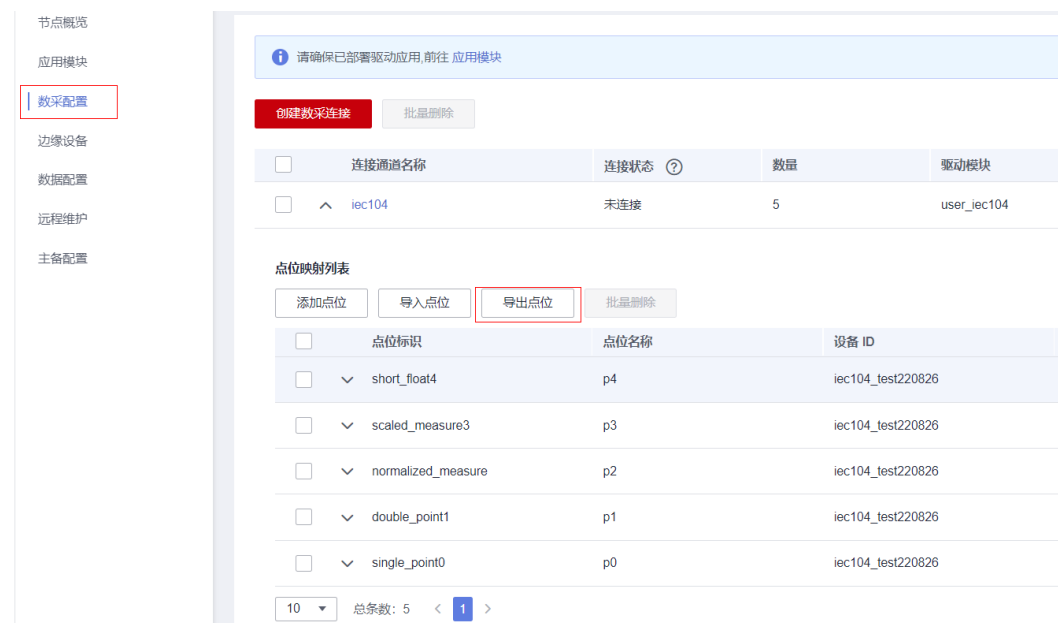
----结束

批量导出点位映射

步骤1 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_loT10”进入页面。

步骤2 在左侧导航栏选择“应用模块 > 数采配置 > 对应的连接通道”。

步骤3 单击“导出点位”完成批量导出点位。



----结束

7.2.6 查看采集结果

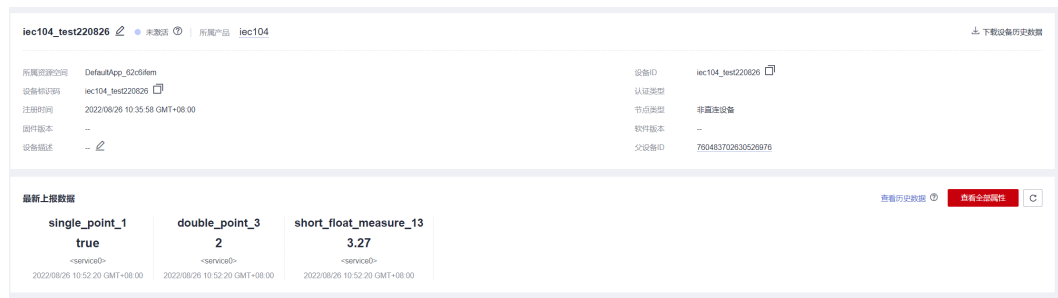
- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 在左侧导航栏选择“边缘节点 > 节点管理”，选择之前创建的边缘节点“test_loT10”，单击节点进入详情页。
- 步骤3** 选择“边缘设备”页签，单击设备ID“test_loT10”，跳转至“设备接入服务控制台”查看采集的数据。

图 7-11 设备列表



说明

数采设备状态平台不感知，即统一显示为“未激活”，请进入设备详情页，查看最新上报数据。



----结束

8 基于 IoT 边缘实现视频数采（Onvif 协议）

8.1 实验简介

📖 说明

当前为公测阶段，实验室环境仅验证过海康DS-2CD2Q10FD-IW摄像头，其余型号当前无验证不保证支持。

视频数采，提供网络监控摄像头的接入能力，通过Onvif协议实现和设备的交互，实现监控视频数据的采集、推送、预览等。

8.2 操作步骤

8.2.1 创建边缘节点

前提条件

- 已注册[华为云](#)账号，且已完成实名认证。
- 已开通[IoT边缘服务](#)和[IoTDA设备接入服务](#)。
- 在创建边缘节点前，检查配置要符合[硬件规格](#)要求，如磁盘、内存和docker版本，切换root用户执行安装命令。

注册专业版节点

- 步骤1** 访问[IoT边缘](#)，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 选择左侧导航栏“边缘节点 > 节点管理”单击右上角“注册节点”。
- 步骤3** 根据页面提示填写参数，勾选右下角“《华为云服务等级协议》”单击“下一步”配置节点数据。

图 8-1 专业版

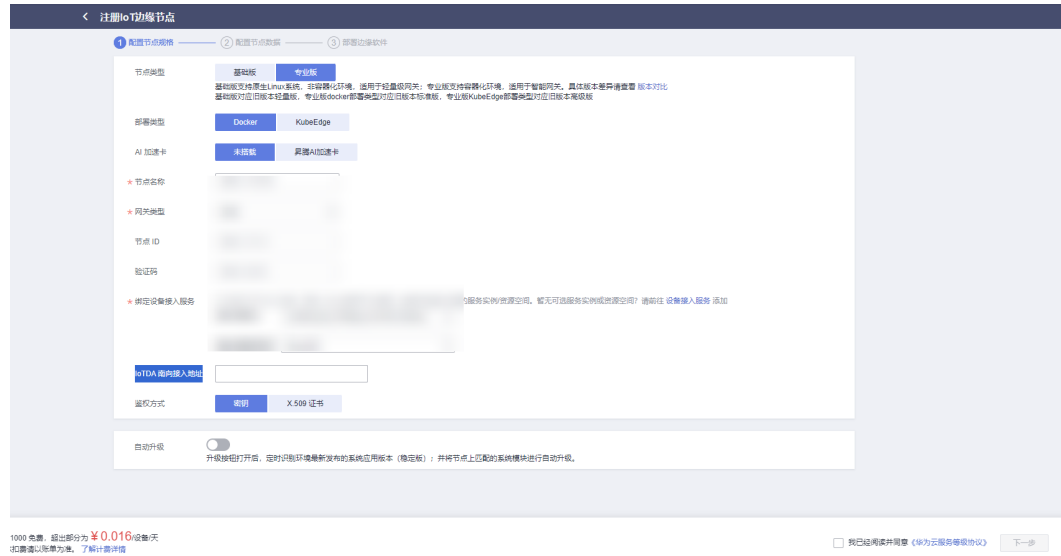


表 8-1 节点规格参数说明

参数名称	参数说明
节点类型	专业版。
部署类型	Docker。
AI加速卡	<ul style="list-style-type: none"> 未搭载：边缘节点未使用AI加速卡时选择。 昇腾AI加速卡：支持华为昇腾AI加速设备，边缘节点的服务器搭载相关昇腾AI加速设备时选择。
昇腾驱动地址	可选，AI加速卡选择昇腾AI加速卡时需要保证安装边缘节点的服务器上已经安装昇腾设备驱动，并根据前端提示填入驱动地址。若改值为空，默认填入Atlas300驱动默认安装地址： /usr/local/Ascend/driver/lib64/。
节点名称	自定义节点名称，如“test”。 说明 节点名称可以在：“节点概览” - “基本信息”中编辑修改。
网关类型	other。
节点ID	自定义（可选填写）。只允许中文，英文字母、数字、中划线（-），下划线（_），最小长度1，最大长度64。 说明 边缘节点ID自定义(不输入默认生成)。
验证码	输入自定义验证码（可选填写）。验证码，执行安装边缘软件命令时与云端通讯的凭证。
设备归属实例	选择默认归属实例。
设备归属资源空间	选择默认资源空间，或为本实验单独新建一个资源空间，如edge。

参数名称	参数说明
IoTDA南向接入地址	选择网关归属的IoTDA实例的南向设备mqtt接入地址，也可以选择自定义的南向接入地址，支持“ip:端口”或“域名:端口”形式。
鉴权方式	密钥：默认生成。 X.509证书：请参考： 基于MQTT.fx的X.509证书接入指导 。 警告 鉴权方式为“密钥”的节点，请勿在IoTDA界面重置节点的密钥，重置密钥会导致网关与IoTDA通信异常，且不可恢复。
指纹	证书文件生成的字符串（可选填写）。只能为40位十六进制字符串或64位六进制字符串。
本地路径	输入证书文件和密钥文件在设备上的存储路径，只允许字母、数字、下划线（_）和斜线（/），以/开头并且以.pem或.crt结尾。
自动升级	可选，开启后节点将自动升级系统应用模块。 注意 升级过程中可能导致节点数据上传中断。

步骤4 填写配置节点数据信息后，单击“立即创建”。

图 8-2 配置节点数据

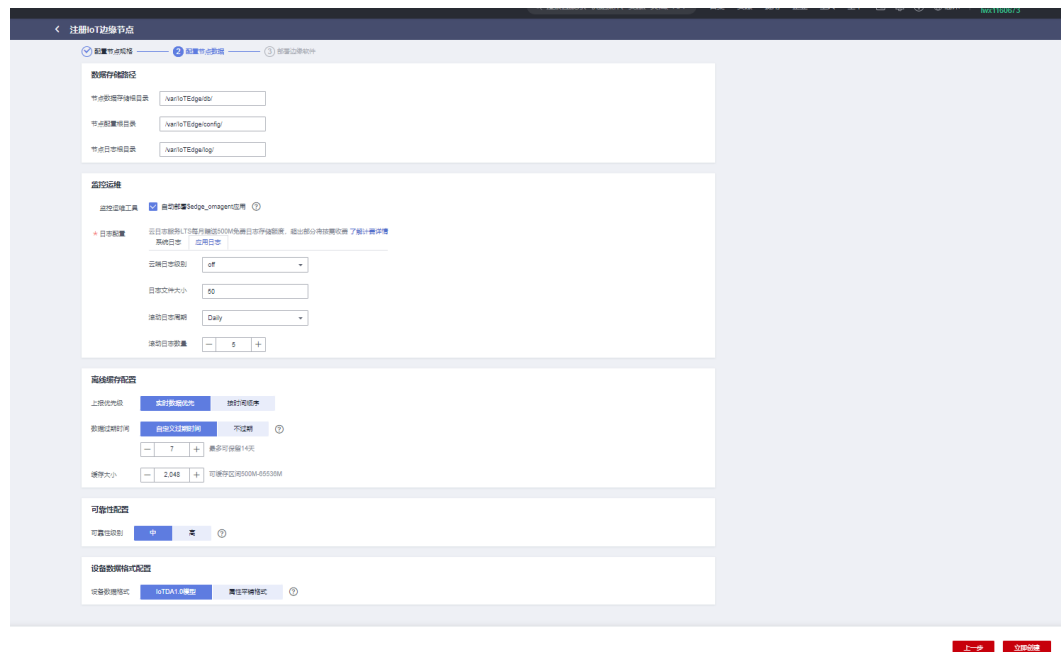


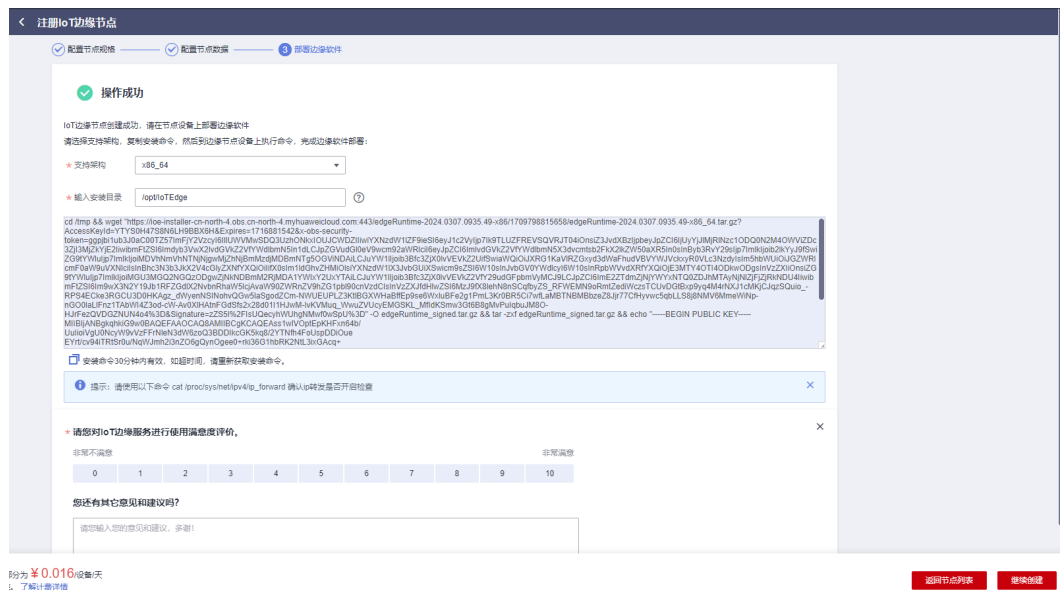
表 8-2 专业版节点信息

参数名称	说明
数据存储路径	默认（允许修改存储路径）。
监控运维	默认勾选“监控运维工具”。
\$edge_omagent应用	负责边缘节点远程监控和运维，提供日志上报、远程SSH、CPU等指标上报、文件上传下载能力。
日志配置	<p>系统日志：边缘节点上已部署的系统应用产生的日志。</p> <p>应用日志：边缘节点上已部署的用户定义应用所产生的日志。</p> <ul style="list-style-type: none"> • 云端日志级别：on, off, trace, debug, info, warn, error, fatal。其中，on表示上传所有日志至云端，off表示不上传日志至云端，其余可选项代表按日志级别采集，选择后将只采集目标级别及以上的日志（模糊匹配）。日志级别的严重程度等级由低到高依次为：trace, debug, info, warn, error, fatal。 • 日志文件大小：日志文件大小限制，单位MB，默认50，取值范围10-1000。某个日志文件如果达到大小限制，则会将文件转储保存至日志目录下。 • 滚动日志周期：日志转储周期，可选项：daily, weekly, monthly, yearly。某个日志文件如果达到转储周期，则会将文件转储保存至日志目录下。日志文件大小和滚动日志周期是同时生效的，满足任何一个条件都会进行日志转储。 • 滚动日志数量：日志文件转储个数，默认5，取值范围1-10。边缘节点保存的转储日志数量如果达到限制，则会删除最早的转储文件。 <p>注意</p> <ol style="list-style-type: none"> 1. 当前“云端日志级别”对日志等级采取模糊匹配的策略，若日志消息中存在与日志级别标识相同的关键字，则也可能被采集。 2. 若要开启“云端日志级别”中的trace, debug, info, warn, error, fatal选项，则必须满足以下三个前置条件： <ol style="list-style-type: none"> 1. 目标节点是基础版节点或部署类型为Docker的专业版节点。需注意，部署类型为KubeEdge的专业版节点不支持此功能。 2. 已安装\$edge_omagent应用，且版本不低于1-1-15-xxx。 3. 应用日志需遵循如下规范打印，以便进行过滤： <ul style="list-style-type: none"> ○ 日志级别标识需打印为（大小写均可）：trace、debug、info、warn、error、fatal。 ○ 每条日志消息以YYYY-MM-DD日期格式作为前缀。 <p>说明</p> <p>专业版（部署类型为Docker）和基础版节点只涉及日志级别这一个参数，其他参数只对专业版（部署类型为KubeEdge）有效。</p>
离线缓存配置	可对离线节点上报的数据进行缓存，定义缓存周期（可配置为不过期）跟缓存大小、上报的优先级、路由转发是否开启离线缓存。仅当上报优先级为“按时间顺序”时支持路由转发开启离线缓存。

参数名称	说明
可靠性配置	可靠性级别分为“中”和“高”2种。中等级别：当应用上报属性和点位的频率超过流控值时，超过部分会被丢弃；高级别：当应用上报属性和点位的频率超过流控值时，通过控制模块的发送速度，降低上报频率，报文不会被丢弃。（注意：高可靠性级别的配置需要定制应用的配合，同时会造成性能的损失）。
设备数据格式配置	设备的数据格式：即单网关节点中设备的标准数据格式，包含 IoTDA1.0物模型格式和属性平铺格式。IoTDA1.0物模型中包含了设备、服务和属性三层结构，具体参考IoTDA网关子设备属性上报报文格式。属性平铺格式，包含设备、属性2层结构，属性结构可以包含自定义结构的拼接，在json中作为一项平铺展开，具体格式请参考 步骤6设备数据格式配置 。 须知 属性平铺格式功能，需要edge_hub应用1-1-38-xx版本及以上支持

步骤5 弹出如下对话框，您可以“返回节点列表”也可以“继续创建”。

图 8-3 注册成功



说明

安装部署节点前，请确认机器的IP转发已开启并且已开启自启动，以及7883和8943端口开放。其中，IP转发规则配置如下：

图 8-4 IP 转发规则配置

```

? MobaXterm 20.4.7
(SSH client, X-server and networking tools)
* SSH session to root@100.04.10.25
* SSH compression :
* SSH browser :
* X11 forwarding : x (disabled or not supported by server)
* DISPLAY : 10.69.33.154:0.0
* For more info, ctrl+click on help or visit our website

Authorized users only. All activities may be monitored and reported.
Last login: Fri Jul 22 10:11:25 2022 from 10.69.33.154
Huawei's internal systems must only be used for conducting Huawei's business or for purposes authorized by Huawei management.
Use is subject to audit at any time by Huawei management.
root@ecs-5172-0004 ~# cat /proc/sys/net/ipv4/ip_forward
1
root@ecs-5172-0004 ~# vim /etc/sysctl.conf
root@ecs-5172-0004 ~# sysctl -p
kernel.core_pattern = /var/log/corefiles/core_%p
kernel.pid_max = 131072
fs.file-max = 409200
net.ipv4.tcp_max_syn_backlog = 4096
mmap_min_size = 0
net.core.netdev_max_backlog = 4096
net.ipv4.tcp_fq.kernel = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.local_port_range = 32768 61000
kernel.sysrq = 1
kernel.softlockup_panic = 0
net.ipv4.ip_forward = 1
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.icmp_ignore_bogus_error_responses = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.tcp_syncookies = 1
kernel.dmesg_restrict = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0
root@ecs-5172-0004 ~# cat /proc/sys/net/ipv4/ip_forward
1
root@ecs-5172-0004 ~#
    
```

1. 检查机器ip_forward in/proc文件系统的值。
cat /proc/sys/net/ipv4/ip_forward
若值等于0，则需要添加配置IP转发规则。
 2. 检查配置文件/etc/sysctl.conf是否包含net.ipv4.ip_forward = 1。
如果配置文件中没有net.ipv4.ip_forward参数项，请新增一行net.ipv4.ip_forward = 1。
vim /etc/sysctl.conf
按“i”进入编辑模式，修改完成后按“Esc”退出编辑模式，然后按“Shift” + “;”组合键，输入wq并回车，保存修改并退出。
 3. 重新加载此文件的值。
sysctl -p
 4. 检查7883和8943端口是否开放，如果是华为云服务器，请参考：[安全组](#)。
- 结束

安装专业版节点

步骤1 在“边缘节点”的节点列表选择您需要安装的节点，单击右侧“安装”，获得提示。

图 8-5 安装

节点名称	节点类型	资源包类型	资源包ID	状态	主机名/网维	操作
lilili	高级版	--	--	未安装		删除
test-centos8	标准版	--	--	离线	192.168.0.235,172.17.0.1,1...	删除
jhhjyu	标准版	泛协议网关资源包1000设备	18160545-a0df-429a-aca5-...	未安装		删除 安装


步骤2 选择边缘节点设备类型，并设备安装目录，然后单击  复制安装命令，使用SSH工具以root用户登录边缘节点服务器后台系统，执行安装命令。

图 8-6 提示

操作成功

请选择支持架构，复制安装命令，然后到边缘节点设备上执行命令，完成边缘软件部署：

* 选择架构

* 输入安装目录

```
cd /tmp && wget "https://ioe-installer.obs.cn-north-4.myhuaweicloud.com:443/edgeRuntime-0.3.072.20220218152640-x86/1645171422610/edgeRuntime-0.3.072.20220218152640-x86_64.tar.gz?AccessKeyId=XRMQYJEYZFLLTWFP6MZV&Expires=1646817715&Signature=UtEXqMEZ11RZacThxUmm2urE%2BQ8%3D" -O edgeRuntime_signed.tar.gz && tar -zxf edgeRuntime_signed.tar.gz && echo "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAss1wIVOptEpkHFxn64b/\nUulioiVgU0NcyW9vVzFFrNleN3dW6zoQ3BDDlkGK5kq8/2YTNfh4FoUspDDiOue\nEYr/cv94ITRtSr0u/NqWJmh2i3nZO6gQynOgee0+rki36G1hbRK2Nl.3ixGAcq+\nDWD6pCG0kE8E/Pillm6NdCKrS3D6TpbbsQeNOz6PY40jBRHngj+ATjgk57xNGj4\n1JGTibzBKx1m0vBRNluXRahAbs5XDyDzseySYpa8sTOMc3wTIC72vVXd155DP\nXpMfve5tk+UDkuWZQEUsPBuVksZrKP3NDFkO7p1jhiC2DCndu6DcZX+8d07uaTWN\nWwIDAQAB\n-----END PUBLIC KEY-----" > edge_installer_pub.pem && openssl dgst -sha256 -verify edge_installer_pub.pem -signature\nedgeRuntime.tar.gz.sig edgeRuntime.tar.gz && tar -zxf edgeRuntime.tar.gz && INSTALL_IOTEDGE_BIN_DIR=/opt/IoTEdge sh\nedge_install.sh iotedge-south.cn-north-4.myhuaweicloud.com 8943 698958664815620096 sys_edge_daemon bad0d7d\n0a2c2f36800f3f42f9e00c76740f5cloud.tps.standard&& ed
```

安装命令30分钟内有效，如超时间，请重新获取安装命令。

我知道了

说明

安装命令的选择需要与购买的ecs的架构保持一致。

图 8-7 执行安装

```
2021-02-07T10:31:47 | INFO | delete edgeRuntime datas.\n2021-02-07T10:31:47 | INFO | delete edgeInstaller\n2021-02-07T10:31:47 | INFO | uninstall edgeRuntime finish.\nroot@ecs-5271:~# docker ps\nCONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES\nroot@ecs-5271:~# arch\nx86_64\nroot@ecs-5271:~# cd /tmp && wget "https://ioe-installer.obs.cn-north-4.myhuaweicloud.com:443/edgeRuntime-0.3.072.20210111180508-x86\n/161036406063/edgeRuntime-0.3.072.20210111180508-x86_64.tar.gz?AccessKeyId=XRMQYJEYZFLLTWFP6MZU&Expires=1612668719&Signature=Xu\nGwUqHk3k6Tz2FAthJvz2B4UD1Fn0z3D" -O edgeRuntime_signed.tar.gz && tar -zxf edgeRuntime_signed.tar.gz && echo "-----BEGIN PUBLIC\nKEY-----\n> MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAss1wIVOptEpkHFxn64b/\n> UulioiVgU0NcyW9vVzFFrNleN3dW6zoQ3BDDlkGK5kq8/2YTNfh4FoUspDDiOue\n> EYr/cv94ITRtSr0u/NqWJmh2i3nZO6gQynOgee0+rki36G1hbRK2Nl.3ixGAcq+\n> DWD6pCG0kE8E/Pillm6NdCKrS3D6TpbbsQeNOz6PY40jBRHngj+ATjgk57xNGj4\n> 1JGTibzBKx1m0vBRNluXRahAbs5XDyDzseySYpa8sTOMc3wTIC72vVXd155DP\n> XpMfve5tk+UDkuWZQEUsPBuVksZrKP3NDFkO7p1jhiC2DCndu6DcZX+8d07uaTWN\n> WwIDAQAB\n> -----END PUBLIC KEY-----" > edge_installer_pub.pem && openssl dgst -sha256 -verify edge_installer_pub.pem -signature edgeRunTi\nme.tar.gz.sig edgeRuntime.tar.gz && tar -zxf edgeRuntime.tar.gz && sh install.sh iotedge-south.cn-north-4.myhuaweicloud.com 8943\n555739634131546112 sys_edge_daemon WT3MQ99d-yUG54rGDSEnMhzjGGfIchTrzsEMzURBe08-f3dc9b178ea144b586b1213957cd33c9 cloud && cd -\n---2021-02-07 10:35:05--- https://ioe-installer.obs.cn-north-4.myhuaweicloud.com/edgeRuntime-0.3.072.20210111180508-x86_16103640606\n3/edgeRuntime-0.3.072.20210111180508-x86_64.tar.gz?AccessKeyId=XRMQYJEYZFLLTWFP6MZU&Expires=1612668719&Signature=XuGwUqHk3k6Tz2\nFAthJvz2B4UD1Fn0z3D\nResolving ioe-installer.obs.cn-north-4.myhuaweicloud.com (ioe-installer.obs.cn-north-4.myhuaweicloud.com)... 100.125.80.29\nConnecting to ioe-installer.obs.cn-north-4.myhuaweicloud.com (ioe-installer.obs.cn-north-4.myhuaweicloud.com) |100.125.80.29|:443... co\nnected.\nHTTP request sent, awaiting response... 200 OK\nLength: 27279194 (26M) [application/gzip]\nSaving to: 'edgeRuntime_signed.tar.gz'\n\nedgeRuntime_signed.tar.gz      100%[=====] 26.01M  95.4MB/s   in 0.3s\n\n2021-02-07 10:35:06 (95.4 MB/s) - 'edgeRuntime_signed.tar.gz' saved [27279194/27279194]\n\nVerified OK\n2021-02-07T10:35:07 | INFO | Current Work dir is /tmp.\n2021-02-07T10:35:07 | INFO | Log written to /tmp/installer.log.\n2021-02-07T10:35:07 | INFO | Start to install edgeDaemon.\n2021-02-07T10:35:07 | INFO |\n2021-02-07T10:35:07 | INFO | (1/5) check docker version
```

图 8-8 安装命令执行成功

```
2022-08-24T17:22:46.621888 | INFO | getDeploymentInfo | 156 | deploymentInfo schema_version:1.0 runtime_runtime settings_runtimeConfig loginOptions:zero, base_path:/cloudbasePath/config log_base_p...
2022-08-24T17:22:46.621888 | INFO | saveBasePathInfo | 74 | config base path start.
2022-08-24T17:22:46.621888 | INFO | saveBasePathInfo | 79 | config log_path:/var/IoTEdge/log, db_path:/var/IoTEdge/db, config_path:/var/IoTEdge/config
2022-08-24T17:22:46.621888 | INFO | addAgentCustomConfigs | 206 | addCustomConfigs start.
2022-08-24T17:22:46.621888 | INFO | pullImage | 305 | start pull image, deploy type is cloud
2022-08-24T17:22:46.621888 | INFO | createModule | 75 | create Module, name:sys_edge_agent
2022-08-24T17:22:46.621888 | INFO | startModule | 159 | startModule created, createModuleResp moduleId:sys_edge_agent, moduleName:sys-edge-agent, appId:sedge_agent, appVersion:1.526-x86, cont...
2022-08-24T17:22:46.621888 | INFO | checkModuleStatus | 284 | start check modules time.
2022-08-24T17:22:46.621888 | INFO | listModule | 245 | listModule
2022-08-24T17:22:46.621888 | INFO | printModulesAndReport | 349 | **success:1 failed:0
2022-08-24T17:22:46.621888 | INFO | reportDeploymentState | 130 | reportDeploymentState deploymentState:DeploymentState node_id:759866899487272960, deploy_id:1661332967180, deploy_state:RUNNING, reason:
2022-08-24T17:22:46.621888 | INFO | reportModuleInstallEvent | 152 | report event, msg:{events:[{"resource":{"node.installation","location":"759866899487272960","event":"succeeded","event_severity":"Info"}
2022-08-24T17:22:46.621888 | INFO | reportModuleStatus | 131 | reportModuleStatus, modulesInfo:ModuleStatusInfoList[host_tag:DEFAULT, deploy_id:1661332967180, modules:[ModuleStatusInfo module_id:sys_e...
2022-08-24T17:22:46.621888 | INFO | install | 432 | install edge finish.
2022-08-24T17:22:46.621888 | INFO | stop | 483 | edgeInstaller stop.
front
root@iotedge-zhshou-test:~# update-agent -l
```

步骤3 单击“我知道了”，等待边缘节点的状态变为“在线”，表示该节点已安装并连接成功。

步骤4 您可以单击“节点名称”，查看该节点详细信息，关于节点详情描述可参见**基本信息**。



创建节点后会在该账号对应的资源空间下创建edge_onvif产品模型，务必不要修改以及删除。其中，创建的产品模型信息如下：

产品名称	产品ID	资源空间	设备类型	协议类型	操作
mqtt_test01	62e49f550278a1265430b02	DefaultApp_82c6fem	MQTT_Device	MQTT	详情 删除
stduy_test1_edge	62e49f260278a1265430b0a	DefaultApp_82c6fem	MQTT_Device	MQTT	详情 删除
mqtt_test	62e49f14895735465526102	DefaultApp_82c6fem	MQTT_Device	MQTT	详情 删除
edge_node	62e30e43583a07274e893f	DefaultApp_82c6fem	edge_node	MQTT	详情 删除
modbus1	62e9008f3563a07274e89540	DefaultApp_82c6fem	modbus	Modbus	详情 删除
edge_onvif	62e7a5ba586e453778516c2b	DefaultApp_82c6fem	edge_onvif	ONVIF	详情 删除
k2x_opcua	62e5599586e453778516bdf	DefaultApp_82c6fem	k2x_opcua	OPC-UA	详情 删除
k2xMQTT	62e91a0e886e453778516b89	DefaultApp_82c6fem	k2xMQTT	MQTT	详情 删除
k2xopcua	62e914c0446c11877e0804b	DefaultApp_82c6fem	k2xopcua	OPC-UA	详情 删除
Modbus_k2x	62e912e4446c11877e0804b	DefaultApp_82c6fem	Modbus_k2x	Modbus	详情 删除

产品 / edge_onvif

edge_onvif ID: 62e7a5ba586e453778516c2b | 注册时间: 23

产品名称	edge_onvif	所属资源空间	DefaultApp_82c6fem
设备类型	edge_onvif	协议类型	ONVIF
数据格式	json	创建时间	2022/07/20 14:50:34 GMT+08:00
厂商名称	-		

模型定义

- 视频ID: Scentig 删除模型
- 视频ID: BaseInfo 删除模型
- 视频ID: Video 删除模型

----结束

8.2.2 部署边缘视频数采应用

操作步骤

如未添加边缘视频数采应用，可参考[添加驱动应用](#)。

步骤1 在左侧导航栏选择“边缘节点 > 节点管理”，选择之前创建的边缘节点“test-edge-video-001”，单击“详情”进入节点详情页。

步骤2 选择左侧“应用模块 > 模块管理”，单击“部署应用”填写基本信息后“确认”。

- 功能用途: 数据采集
- 边缘应用: \$vds_onvif
- 选择版本: 当前最新的版本

提示

功能用途	数据处理	协议解析	本地子系统集成
	网关管理	混合应用	数据采集

* 边缘应用: \$vds_onvif

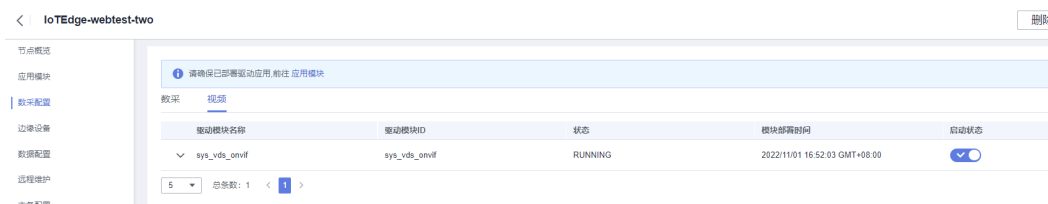
* 选择版本: 1-1-17-standard-x86 支持多部署

模块名称: _____

端口映射: 容器端口 ? 主机端口 ? 操作

确认 取消

步骤3 数采配置中启动该视频数采模块。



----结束

8.2.3 添加视频设备

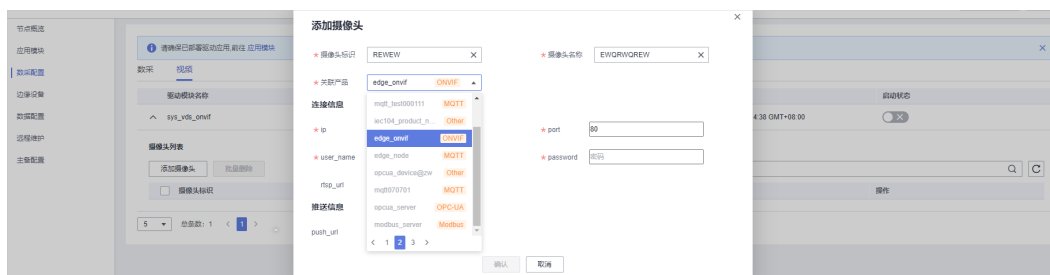
用户启动视频数采模块后，需要在摄像头列表下添加对应的视频设备，用于监控视频数据的采集、推送、预览。

方式一：

步骤1 在左侧导航栏选择“边缘节点 > 节点管理”选择之前创建的边缘节点“test-edge-video-001”，单击节点进入详情页。

步骤2 选择左侧导航栏“数采配置”，单击“视频”，摄像头列表-添加摄像头。

步骤3 关联产品时，仅支持选择onvif协议的产品模型。



步骤4 按照如下图示依次输入监控设备的接入信息。

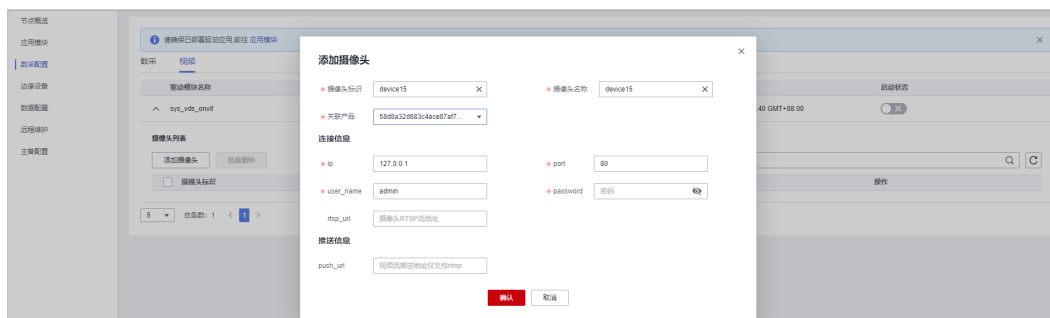


表 8-3 参数说明：

参数名称	说明	示例
ip	视频设备的接入IP	127.0.0.1
port	视频设备的接入端口port	5506
user_name	Onvif协议接入时的用户名	admin
password	密码（onvif协议接入时的密码或者GB设备的认证密码）	password
rtsp_url	监控设备的rtsp取流地址。默认不配置由数采应用自行获取设备的主码流地址，如果配置则以配置的地址为视频数据的采集地址	rtsp://10.58.239.64:554/stream01

参数名称	说明	示例
push_url	表示视频数据的推送地址。当前仅支持 rtmp，如果为空代表不推送视频数据	rtmp:// 100.93.18.171:1985/ edge_video/ device_test_push_0001

添加成功后显示如下图所示:



说明

- 如果设备参数填写无误，可以在设备接入云服务查看该设备的运行状态。
- 设备在线，则表示边缘节点已经和对应的设备能够进行通信，且默认会上报该设备的基本信息。

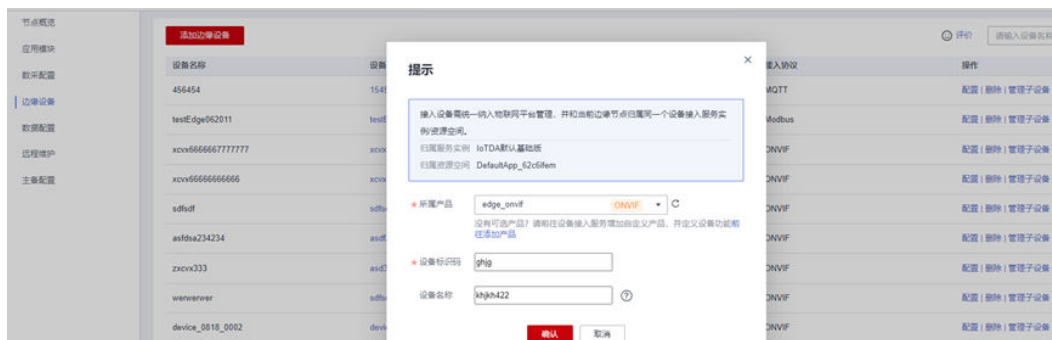


----结束

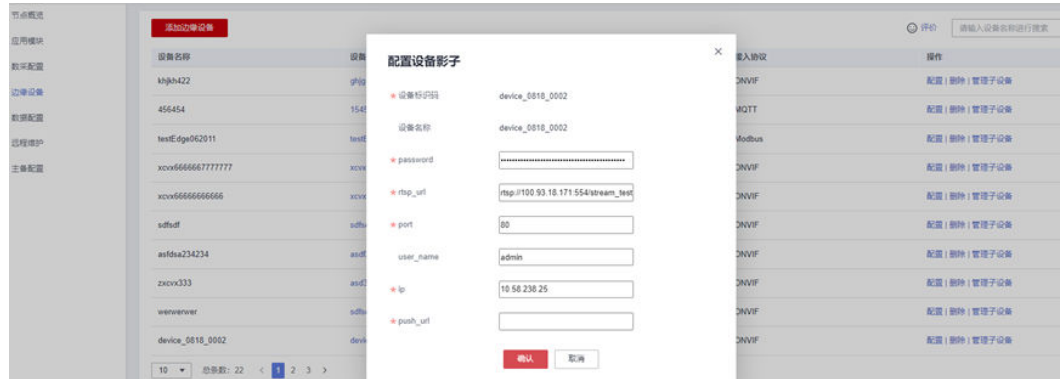
方式二:

步骤1 在左侧导航栏选择“边缘节点 > 节点管理”选择之前创建的边缘节点“test-edge-video-001”，单击节点进入详情页。

步骤2 选择左侧导航栏“边缘设备”，单击“添加边缘设备”。



步骤3 选择左侧导航栏“边缘设备”，单击“添加边缘设备”，单击配置，可同步编辑设备。



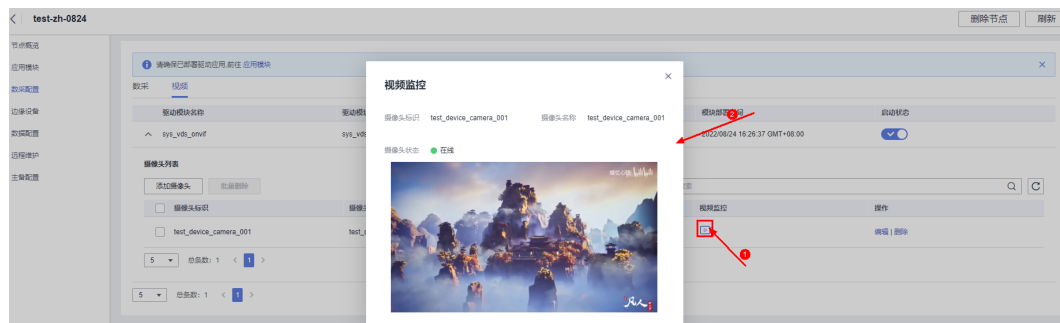
----结束

8.2.4 视频预览

操作步骤

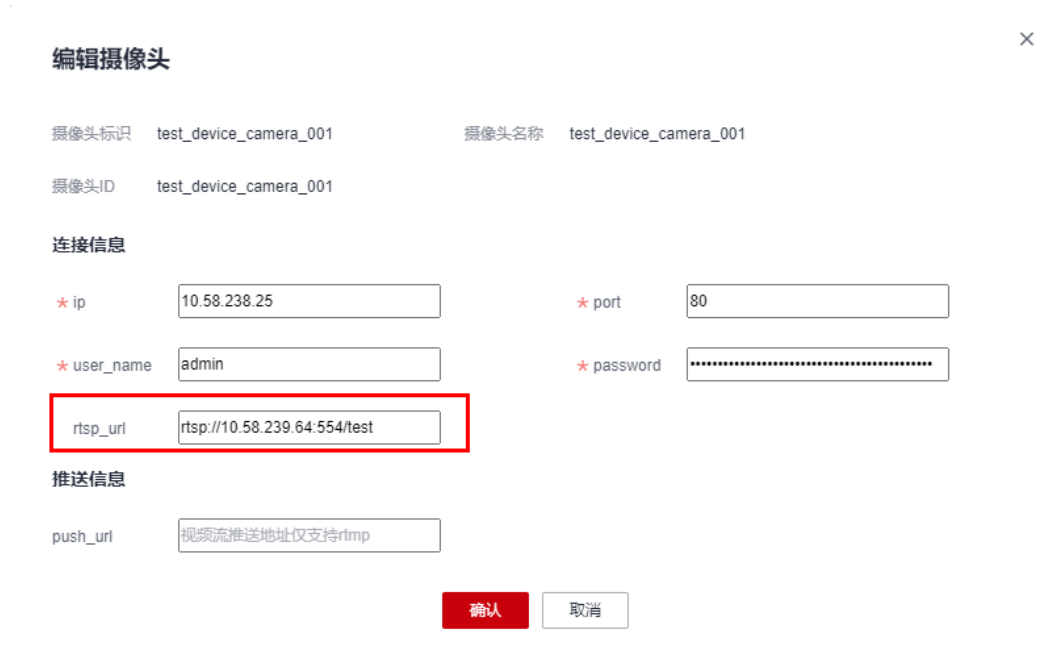
步骤1 在左侧导航栏选择“边缘节点 > 节点管理”“选择之前创建的边缘节点“test-edge-video-001”，单击“详情”进入节点详情页。

步骤2 选择左侧“数采配置 > 视频”，单击“视频监控”。

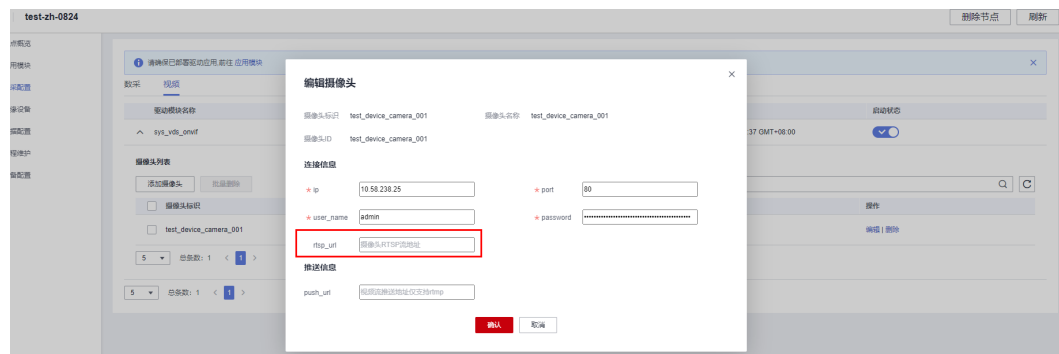


说明

此处使用了自定义的rtsp地址来模拟摄像头的视频数据，使用过程中如果没有配置该参数则由数采应用自行获取网络摄像头的主码流rtsp地址，例如：rtsp://10.58.238.25:554/Streaming/Channels/101?transportmode=unicast&profile=Profile_1。



删除自定义的rtsp地址进行测试：



删除rtsp的配置数据，再次进行预览，如下图所示：



说明

1. 预览视频时不支持音频播放。
2. 使用本地视频源文件进行视频预览，画面加载将近6S；使用真实摄像头预览时画面加载平均4S。
3. 同时预览同一设备，预览时其中一个通道关闭，另外一个通道也强制关闭。

----结束

8.2.5 流媒体服务器搭建（可选）

8.2.5.1 安装 nginx+http-flv(rtmp 服务器)

操作步骤

下载[安装指导](#)。

步骤1 安装依赖项openssl,下载路径<https://www.openssl.org/source/old/1.1.1/>，这里选择1.1.1g。

Old 1.1.1 Releases		
KBytes	Date	File
9629	2022-Jun-21 14:03:44	openssl-1.1.1p.tar.gz (SHA256) (PGP sign) (SHA1)
9625	2022-May-03 14:02:38	openssl-1.1.1o.tar.gz (SHA256) (PGP sign) (SHA1)
9619	2022-Mar-15 15:24:36	openssl-1.1.1n.tar.gz (SHA256) (PGP sign) (SHA1)
9616	2021-Dec-14 16:28:09	openssl-1.1.1m.tar.gz (SHA256) (PGP sign) (SHA1)
9603	2021-Aug-24 13:46:31	openssl-1.1.1l.tar.gz (SHA256) (PGP sign) (SHA1)
9593	2021-Mar-25 13:41:15	openssl-1.1.1k.tar.gz (SHA256) (PGP sign) (SHA1)
9592	2021-Feb-16 15:52:38	openssl-1.1.1j.tar.gz (SHA256) (PGP sign) (SHA1)
9578	2020-Dec-08 14:39:04	openssl-1.1.1i.tar.gz (SHA256) (PGP sign) (SHA1)
9580	2020-Sep-22 13:11:49	openssl-1.1.1h.tar.gz (SHA256) (PGP sign) (SHA1)
9571	2020-Apr-21 13:01:56	openssl-1.1.1g.tar.gz (SHA256) (PGP sign) (SHA1)
9563	2020-Mar-31 12:41:55	openssl-1.1.1f.tar.gz (SHA256) (PGP sign) (SHA1)
9563	2020-Mar-17 14:56:32	openssl-1.1.1e.tar.gz (SHA256) (PGP sign) (SHA1)
8638	2019-Sep-10 13:53:14	openssl-1.1.1d.tar.gz (SHA256) (PGP sign) (SHA1)
8656	2019-May-28 13:26:28	openssl-1.1.1c.tar.gz (SHA256) (PGP sign) (SHA1)
8021	2019-Feb-26 14:34:59	openssl-1.1.1b.tar.gz (SHA256) (PGP sign) (SHA1)
8154	2018-Nov-20 14:07:08	openssl-1.1.1a.tar.gz (SHA256) (PGP sign) (SHA1)
8142	2018-Sep-11 13:21:29	openssl-1.1.1.tar.gz (SHA256) (PGP sign) (SHA1)
8213	2018-Aug-21 12:31:28	openssl-1.1.1-pre9.tar.gz (SHA256) (PGP sign) (SHA1)
8139	2018-Jun-20 14:58:10	openssl-1.1.1-pre8.tar.gz (SHA256) (PGP sign) (SHA1)
8114	2018-May-29 12:29:20	openssl-1.1.1-pre7.tar.gz (SHA256) (PGP sign) (SHA1)
8092	2018-May-01 13:00:55	openssl-1.1.1-pre6.tar.gz (SHA256) (PGP sign) (SHA1)
8094	2018-Apr-17 13:50:45	openssl-1.1.1-pre5.tar.gz (SHA256) (PGP sign) (SHA1)
8065	2018-Apr-03 13:32:24	openssl-1.1.1-pre4.tar.gz (SHA256) (PGP sign) (SHA1)
6398	2018-Mar-20 13:55:55	openssl-1.1.1-pre3.tar.gz (SHA256) (PGP sign) (SHA1)
6333	2018-Feb-27 13:50:44	openssl-1.1.1-pre2.tar.gz (SHA256) (PGP sign) (SHA1)
6256	2018-Feb-13 13:57:58	openssl-1.1.1-pre1.tar.gz (SHA256) (PGP sign) (SHA1)

假设将下载文件放至"/home"目录下并解压，依次执行以下命令进行openssl的编译安装。

```
./config no-asm -shared  
make -j4  
make install
```

步骤2 下载nginx和Nginx-http-flv-module。

安装指导以及下载链接: <https://github.com/winshining/nginx-http-flv-module/blob/master/README.CN.md>

📖 说明

上述的包均为源码, 需要将nginx和nginx-http-flv-module一起编译, 接下来进行编译。

步骤3 将上述的压缩包分别解压到"/home"目录下。

```
total 41628
-rw-r----- 1 root root 19338737 Dec 28 2020 20200709175313976edge-installer_1.0.9_x86_64.tar.gz
-rw-r----- 1 root root 81 Feb 23 2021 fail.txt
-rw-r----- 1 root root 303555 Feb 23 2021 fail.txt.bnk
drwxr-xr-x 2 root root 4096 Jul 22 14:13 ffmpeg
-rw-r----- 1 root root 18887963 Jul 22 11:06 ffmpeg-release-arm64-static.tar.xz
drwxr-xr-x 8 1001 1001 4096 Apr 17 2018 nginx-1.14.0
-rw-r----- 1 root root 1016272 Jul 22 14:46 nginx-1.14.0.tar.gz
drwxr-xr-x 8 root root 4096 Feb 20 13:47 nginx-http-flv-module-master
-rw-r----- 1 root root 3033568 Jul 22 14:46 nginx-http-flv-module-master.zip
drwxr-xr-x 2 opsadmin admingroup 4096 Mar 17 2020 opsadmin
drwxr-xr-x 2 root root 4096 Jul 22 14:36 rtsp_server
drwxr-xr-x 2 service servicegroup 4096 Mar 10 2020 service
-rw-r----- 1 root root 333 Feb 23 2021 test.file
```

步骤4 使用以下命令进行编译。

```
./configure --add-module=/home/nginx-http-flv-module-master
make -j4
make install
```

默认nginx安装在/usr/local/nginx

步骤5 配置nginx。

配置文件为nginx.conf

Nginx的配置目录为/usr/local/nginx/conf/

可以使用下面的文件直接覆盖nginx.conf

```
rtmp {
    server {
        listen 1985;
        application edge_video {
            live on;
            gop_cache on;
        }
        application hls {
            live on;
            hls on;
            hls_path /tmp/hls;
        }
    }
}
```

上面两个配置分别表示rtmp服务器地址监听地址以及appname。

配置文件:

```
#user nobody;
worker_processes 1;
#error_log logs/error.log;
#error_log logs/error.log notice;
#error_log logs/error.log info;
#pid logs/nginx.pid;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    #log_format main '$remote_addr - $remote_user [$time_local] "$request" '
    # '$status $body_bytes_sent "$http_referer" ';
```

```
# "$http_user_agent" "$http_x_forwarded_for";
#access_log logs/access.log main;
sendfile on;
#tcp_nopush on;
#keepalive_timeout 0;
keepalive_timeout 65;
#gzip on;
server {
    listen 8080;
    server_name localhost;
    #charset koi8-r;
    #access_log logs/host.access.log main;

    location / {
        root html;
        index index.html index.htm;
    }

    #error_page 404 /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }

    location /live {
        flv_live on;
        chunked_transfer_encoding on; #支持 'Transfer-Encoding: chunked' 方式回复
        add_header 'Access-Control-Allow-Origin' '*'; #添加额外的 HTTP 头
        add_header 'Access-Control-Allow-Credentials' 'true'; #添加额外的 HTTP 头
    }
    location /hls {
        types {
            application/vnd.apple.mpegurl m3u8;
            video/mp2t ts;
        }

        root /tmp;
        add_header 'Cache-Control' 'no-cache';
    }
    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass http://127.0.0.1;
    #}

    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    #
    #location ~ \.php$ {
    #    root html;
    #    fastcgi_pass 127.0.0.1:9000;
    #    fastcgi_index index.php;
    #    fastcgi_param SCRIPT_FILENAME /scripts$fastcgi_script_name;
    #    include fastcgi_params;
    #}

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny all;
    #}
}

# another virtual host using mix of IP-, name-, and port-based configuration
#
```

```
#server {
# listen 8000;
# listen somename:8080;
# server_name somename alias another.alias;

# location / {
# root html;
# index index.html index.htm;
# }
#}

# HTTPS server
#
#server {
# listen 443 ssl;
# server_name localhost;

# ssl_certificate cert.pem;
# ssl_certificate_key cert.key;

# ssl_session_cache shared:SSL:1m;
# ssl_session_timeout 5m;

# ssl_ciphers HIGH:!aNULL:!MD5;
# ssl_prefer_server_ciphers on;

# location / {
# root html;
# index index.html index.htm;
# }
#}
}
rtmp {
server {
listen 1985;
application edge_video{
live on;
gop_cache on;
}
application hls {
live on;
hls on;
hls_path /tmp/hls;
}
}
}
```

步骤6 运行nginx。

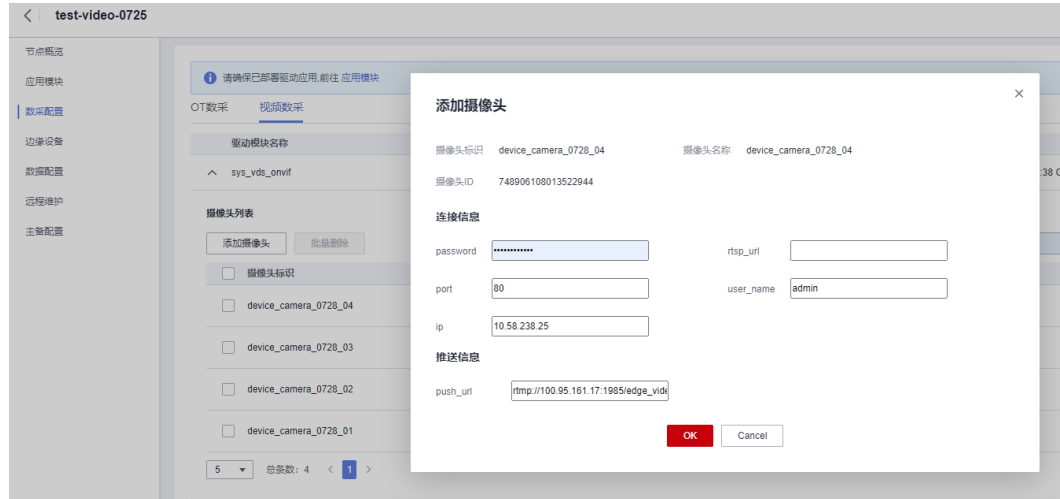
```
export LD_LIBRARY_PATH=/usr/local/lib64:${LD_LIBRARY_PATH}
cd /usr/local/nginx/sbin
./nginx
```

----结束

8.2.6 监控视频数据推送（可选）

操作步骤

步骤1 选择左侧“数采配置 > 视频”，单击“编辑”。



其中, Push_url 设置为 `rtmp://100.95.161.17:1985/edge_video/device_camera_0728_04`。

使用 vlc 测试 [步骤1](#) 中看到的推送地址是否有视频数据, vlc 下载地址: <https://www.videolan.org/>。



可以正常预览说明视频数据推送正常。

----结束

9 基于 IoT 边缘实现 OT 数采 (Modbus-TCP)

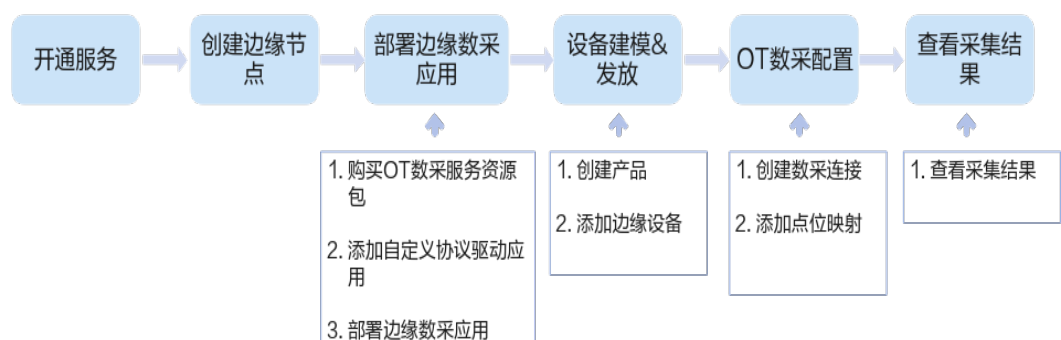
9.1 实验简介

OT数采配置，提供了在IoT边缘控制台简单配置，实现OT数采（DCS、PLC、OPCUA等）。

通过本实验，您将能够：

- 了解并掌握IoT边缘服务的功能特点与使用方法。
- 掌握将OT设备的数据上报云端的方法。

OT 数采流程介绍



9.2 操作步骤

9.2.1 创建边缘节点

前提条件

- 已注册[华为云](#)账号，且已完成实名认证。
- 已开通[IoT边缘服务](#)和[IoTDA设备接入服务](#)。

- 在创建边缘节点前，检查配置要符合**硬件规格**要求，如磁盘、内存和docker版本，切换root用户执行安装命令。

注册专业版节点

- 步骤1** 访问**IoT边缘**，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 选择左侧导航栏“边缘节点 > 节点管理”单击右上角“注册节点”。
- 步骤3** 根据页面提示填写参数，勾选右下角“《华为云服务等级协议》”单击“下一步”配置节点数据。

图 9-1 专业版

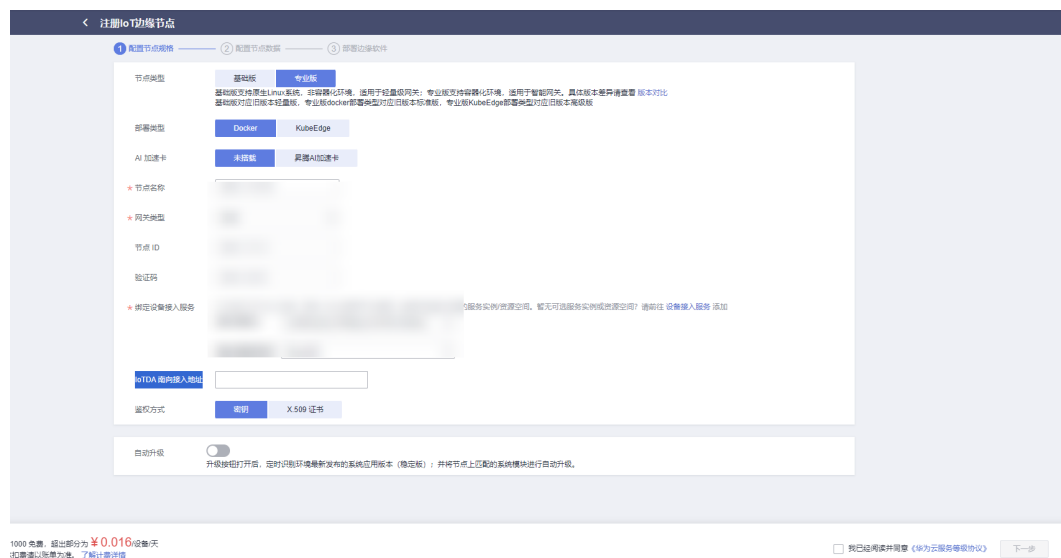


表 9-1 节点规格参数说明

参数名称	参数说明
节点类型	专业版。
部署类型	Docker。
AI加速卡	<ul style="list-style-type: none"> 未搭载：边缘节点未使用AI加速卡时选择。 昇腾AI加速卡：支持华为昇腾AI加速设备，边缘节点的服务器搭载相关昇腾AI加速设备时选择。
昇腾驱动地址	可选，AI加速卡选择昇腾AI加速卡时需要保证安装边缘节点的服务器上已经安装昇腾设备驱动，并根据前端提示填入驱动地址。若改值为空，默认填入Atlas300驱动默认安装地址： /usr/local/Ascend/driver/lib64/。
节点名称	自定义节点名称，如“test”。 说明 节点名称可以在：“节点概览” - “基本信息”中编辑修改。
网关类型	other。

参数名称	参数说明
节点ID	自定义 (可选填写)。只允许中文, 英文字母、数字、中划线 (-), 下划线 (_), 最小长度1, 最大长度64。 说明 边缘节点ID自定义(不输入默认生成)。
验证码	输入自定义验证码 (可选填写)。验证码, 执行安装边缘软件命令时与云端通讯的凭证。
设备归属实例	选择默认归属实例。
设备归属资源空间	选择默认资源空间, 或为本实验单独新建一个资源空间, 如 edge。
IoTDA南向接入地址	选择网关归属的IoTDA实例的南向设备mqtt接入地址, 也可以选择自定义的南向接入地址, 支持“ip:端口”或“域名:端口”形式。
鉴权方式	密钥: 默认生成。 X.509证书: 请参考: 基于MQTT.fx的X.509证书接入指导 。 警告 鉴权方式为“密钥”的节点, 请勿在IoTDA界面重置节点的密钥, 重置密钥会导致网关与IoTDA通信异常, 且不可恢复。
指纹	证书文件生成的字符串 (可选填写)。只能为40位十六进制字符串或64位六进制字符串。
本地路径	输入证书文件和密钥文件在设备上的存储路径, 只允许字母、数字、下划线 (_) 和斜线 (/), 以/开头并且以.pem或.crt结尾。
自动升级	可选, 开启后节点将自动升级系统应用模块。 注意 升级过程中可能导致节点数据上传中断。

步骤4 填写配置节点数据信息后, 单击“立即创建”。

图 9-2 配置节点数据

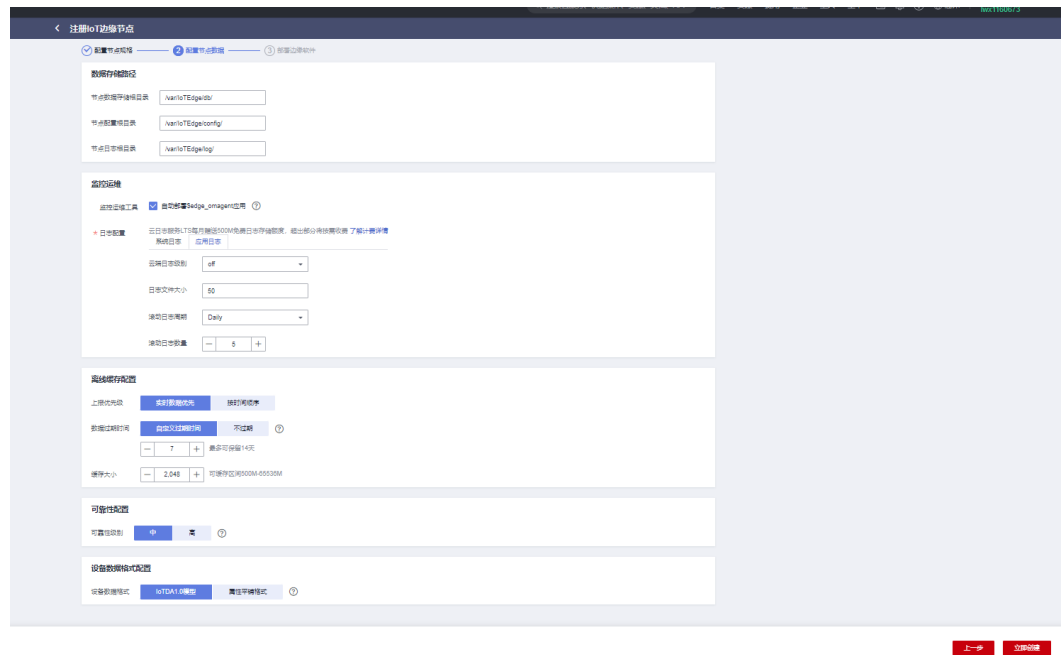


表 9-2 专业版节点信息

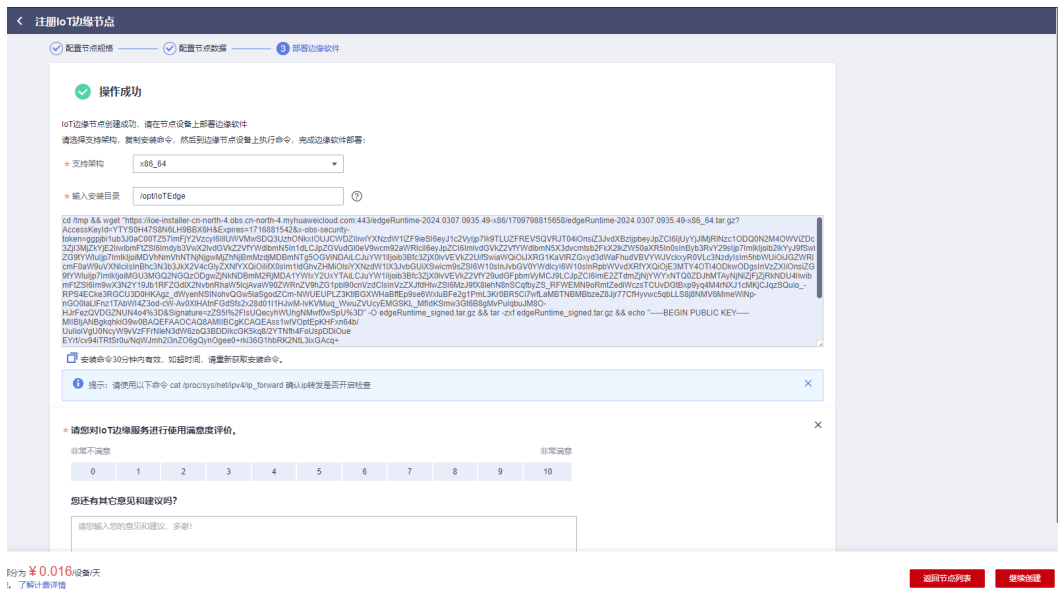
参数名称	说明
数据存储路径	默认（允许修改存储路径）。
监控运维	默认勾选“监控运维工具”。
\$edge_omagent应用	负责边缘节点远程监控和运维，提供日志上报、远程SSH、CPU等指标上报、文件上传下载能力。

参数名称	说明
日志配置	<p>系统日志：边缘节点上已部署的系统应用产生的日志。 应用日志：边缘节点上已部署的用户定义应用所产生的日志。</p> <ul style="list-style-type: none"> • 云端日志级别：on, off, trace, debug, info, warn, error, fatal。其中，on表示上传所有日志至云端，off表示不上传日志至云端，其余可选项代表按日志级别采集，选择后将只采集目标级别及以上的日志（模糊匹配）。日志级别的严重程度等级由低到高依次为：trace, debug, info, warn, error, fatal。 • 日志文件大小：日志文件大小限制，单位MB，默认50，取值范围10-1000。某个日志文件如果达到大小限制，则会将文件转储保存至日志目录下。 • 滚动日志周期：日志转储周期，可选项：daily, weekly, monthly, yearly。某个日志文件如果达到转储周期，则会将文件转储保存至日志目录下。日志文件大小和滚动日志周期是同时生效的，满足任何一个条件都会进行日志转储。 • 滚动日志数量：日志文件转储个数，默认5，取值范围1-10。边缘节点保存的转储日志数量如果达到限制，则会删除最早的转储文件。 <p>注意</p> <ol style="list-style-type: none"> 1. 当前“云端日志级别”对日志等级采取模糊匹配的策略，若日志消息中存在与日志级别标识相同的关键字，则也可能被采集。 2. 若要开启“云端日志级别”中的trace, debug, info, warn, error, fatal选项，则必须满足以下三个前置条件： <ol style="list-style-type: none"> 1. 目标节点是基础版节点或部署类型为Docker的专业版节点。需注意，部署类型为KubeEdge的专业版节点不支持此功能。 2. 已安装\$edge_omagent应用，且版本不低于1-1-15-xxx。 3. 应用日志需遵循如下规范打印，以便进行过滤： <ul style="list-style-type: none"> ○ 日志级别标识需打印为（大小写均可）：trace、debug、info、warn、error、fatal。 ○ 每条日志消息以YYYY-MM-DD日期格式作为前缀。 <p>说明 专业版（部署类型为Docker）和基础版节点只涉及日志级别这一个参数，其他参数只对专业版（部署类型为KubeEdge）有效。</p>
离线缓存配置	<p>可对离线节点上报的数据进行缓存，定义缓存周期（可配置为不过期）跟缓存大小、上报的优先级、路由转发是否开启离线缓存。仅当上报优先级为“按时间顺序”时支持路由转发开启离线缓存。</p>
可靠性配置	<p>可靠性级别分为“中”和“高”2种。中级别：当应用上报属性和点位的频率超过流控值时，超过部分会被丢弃；高级别：当应用上报属性和点位的频率超过流控值时，通过控制模块的发送速度，降低上报频率，报文不会被丢弃。（注意：高可靠性级别的配置需要定制应用的配合，同时会造成性能的损失）。</p>

参数名称	说明
设备数据格式配置	<p>设备的数据格式：即单网关节点中设备的标准数据格式，包含 IoTDA1.0物模型格式和属性平铺格式。IoTDA1.0物模型中包含了设备、服务和属性三层结构，具体参考IoTDA网关子设备属性上报报文格式。属性平铺格式，包含设备、属性2层结构，属性结构可以包含自定义结构的拼接，在json中作为一项平铺展开，具体格式请参考步骤6设备数据格式配置。</p> <p>须知 属性平铺格式功能，需要edge_hub应用1-1-38-xx版本及以上支持</p>

步骤5 弹出如下对话框，您可以“返回节点列表”也可以“继续创建”。

图 9-3 注册成功



说明

安装部署节点前，请确认机器的IP转发已开启并且已开启自启动，以及7883和8943端口开放。其中，IP转发规则配置如下：

图 9-4 IP 转发规则配置



1. 检查机器ip_forward in/proc文件系统的值。
`cat /proc/sys/net/ipv4/ip_forward`
 若值等于0，则需要添加配置IP转发规则。
2. 检查配置文件/etc/sysctl.conf是否包含net.ipv4.ip_forward = 1。
 如果配置文件中没有net.ipv4.ip_forward参数项，请新增一行net.ipv4.ip_forward = 1。
`vim /etc/sysctl.conf`
 按“i”进入编辑模式，修改完成后按“Esc”退出编辑模式，然后按“Shift” + “;”组合键，输入wq并回车，保存修改并退出。
3. 重新加载此文件的值。
`sysctl -p`
4. 检查7883和8943端口是否开放，如果是华为云服务器，请参考：[安全组](#)。

----结束

安装专业版节点

步骤1 在“节点管理”的节点列表选择您需要安装的节点，单击右侧“安装”，获得提示。

图 9-5 安装

节点名称	节点类型	资源包类型	资源包ID	状态	主机名/网络	操作
	专业版	--	--	未安装		删除 安装
	专业版	--	--	离线	172.30.7.64,172.17.0.1,1...	删除
	专业版	--	--	已安装		删除


步骤2 选择支持架构和安装目录，然后单击  复制安装命令，使用SSH工具以root用户登录边缘节点服务器后台系统，执行安装命令。

图 9-8 安装命令执行成功

```

2024-05-28T14:44:29.581321 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:PENDING, reason:(zero), ]
2024-05-28T14:44:31.632105 | INFO | saveBasePathInfo | 78 | config base path start.
2024-05-28T14:44:31.632291 | INFO | saveBasePathInfo | 83 | config log path:/var/IoTEdge/log/, db_path:/var/IoTEdge/db/, config_path:/var/IoTEdge/config/
2024-05-28T14:44:31.632457 | INFO | saveBasePathInfo | 95 | config base path success.
2024-05-28T14:44:31.632535 | INFO | addAgentCustomConfigs | 201 | addCustomConfigs start.
2024-05-28T14:44:31.632603 | INFO | addAgentCustomConfigs | 205 | addCustomConfigs end.
2024-05-28T14:44:31.632654 | INFO | pullImage | 319 | start pull image, deploy type is cloud
2024-05-28T14:44:35.888519 | INFO | createModule | 76 | create Module, name:sys_edge_agent
2024-05-28T14:44:35.939013 | INFO | createModule | 94 | module created, CreateResp[ moduleId:sys_edge_agent, moduleName:sys-edge-agent, appId:sedge_agent, appVersion:test-tpm-2-x86, containerId:4b3e43cee7eac9bc282b994377abfb48a284389a36856f3acc0338743036e, processID:(zero), type:docker, ]
2024-05-28T14:44:35.939163 | INFO | startModule | 160 | startModule moduleId:sys_edge_agent
2024-05-28T14:44:42.390994 | INFO | listModule | 259 | listModule
2024-05-28T14:44:42.440775 | INFO | checkModulesDo | 340 | start check modules, time:1
2024-05-28T14:44:42.440909 | INFO | listModule | 259 | listModule
2024-05-28T14:44:42.491163 | INFO | printModulesAndReport | 404 | ***success:1 failed:0
2024-05-28T14:44:42.491339 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:RUNNING, reason:(zero), ]
2024-05-28T14:44:42.542192 | INFO | reportEvent | 158 | Begin reporting event: {"events":[{"resource":"node.installation","location":"zzzzz","event":"succeeded","event_severity":"Info","event_time":"20240528T064442Z","notify_data":{"host_tag":"DEFAULT","edge_node_id":"zzzzz"}}]}
2024-05-28T14:44:42.592933 | INFO | reportModuleStatus | 138 | reportModuleStatus, modulesInfo:ModuleStatusInfo[ host_tag:DEFAULT, deploy_id:1716878669308, modules:[ModuleStatusInfo[ module_id:sys_edge_agent, app_id:sedge_agent, app_version:test-tpm-2-x86, container_state:RUNNING, liveness_state:(zero), update_time:1716878669305, ]], ]
2024-05-28T14:44:43.043689 | INFO | checkBridgeNetwork | 560 | begin checking bridge network
2024-05-28T14:44:39.024137 | INFO | install | 494 | install edge flash
2024-05-28T14:44:30.524559 | INFO | stop | 545 | edgeinstaller stop.
/root
    
```

步骤3 单击“我知道了”，等待边缘节点的状态变为“在线”，表示该节点已安装并连接成功。

步骤4 您可以单击“节点名称”，查看该节点详细信息，关于节点详情描述可参见[节点概览](#)。

----结束

9.2.2 准备数据源

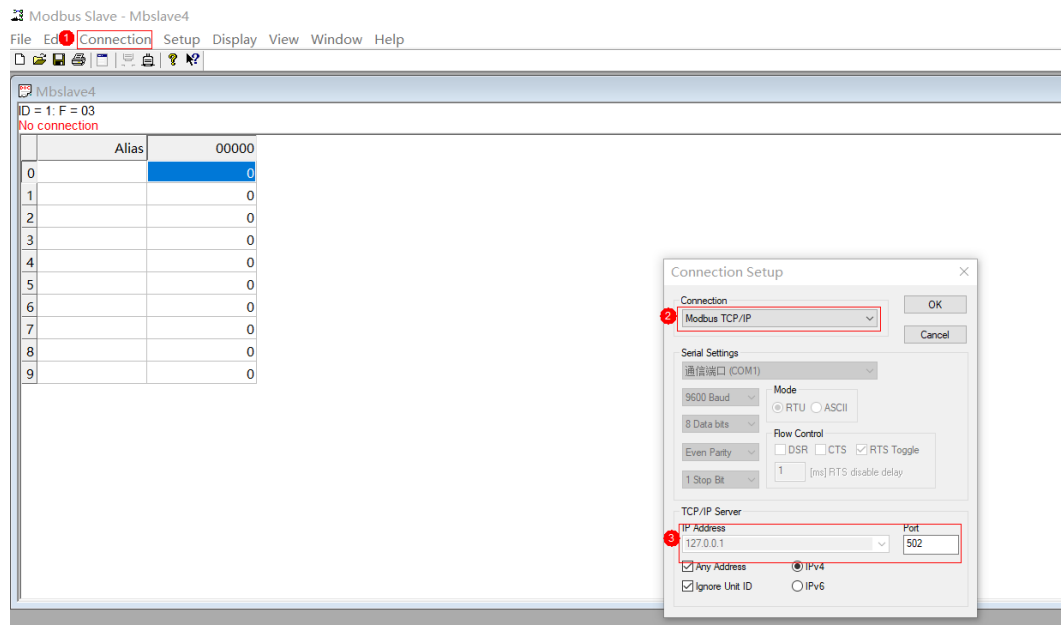
在部署数采应用之前需要先准备数据源，数据源可以是Modbus Server的对应服务器，也可以是类似Modbus Slave的模拟器网关，此处以Modbus Slave模拟器举例。

安装 Modbus Slave 模拟器

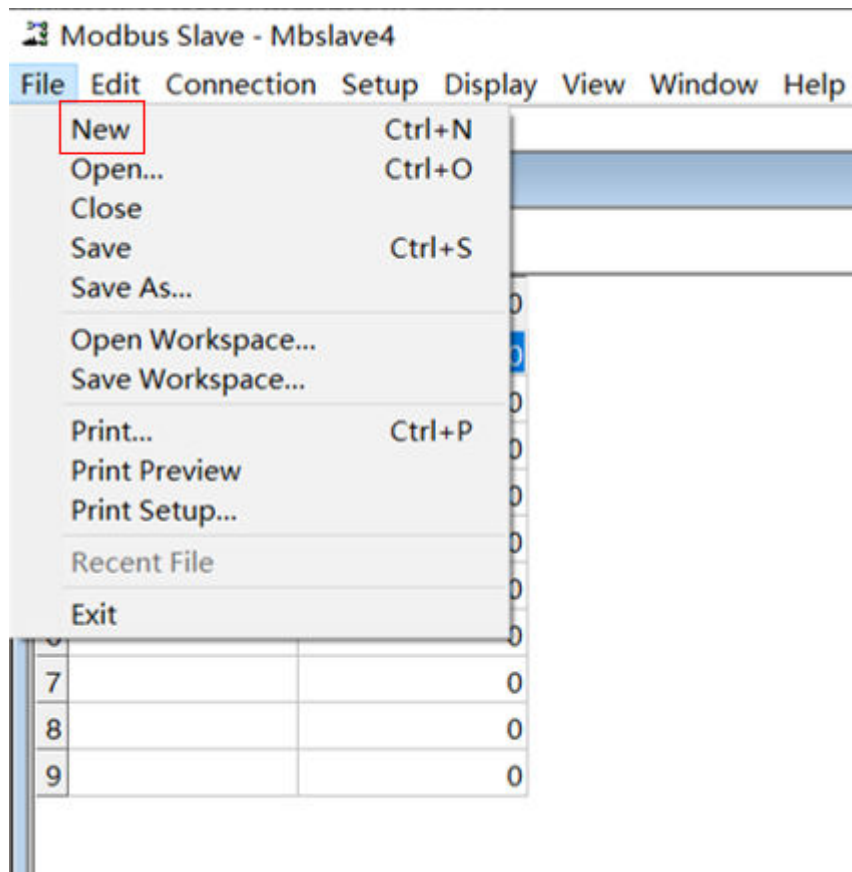
步骤1 下载[Modbus Slave模拟器](#)。

步骤2 安装完毕后，启动Modbus Slave模拟器。

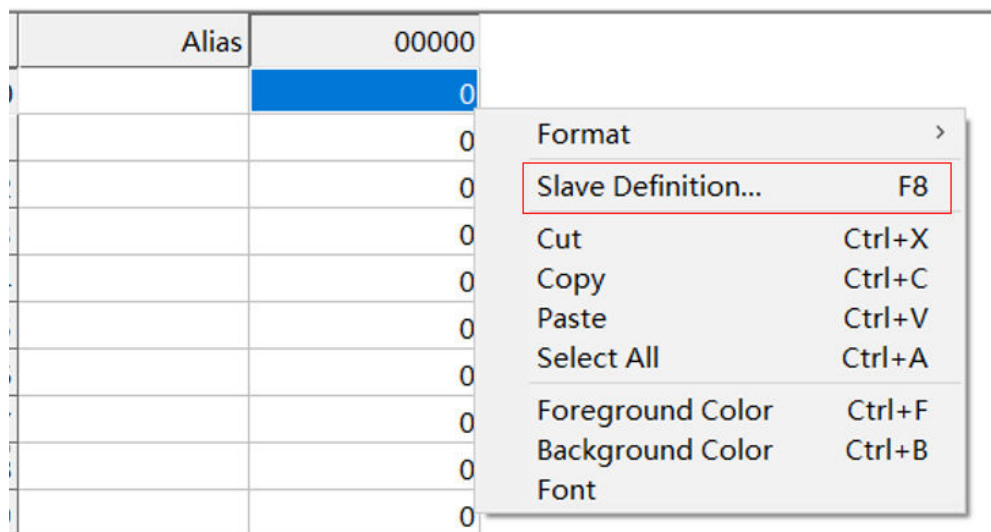
步骤3 单击“Connection”添加连接信息，将连接方式设置为“Modbus TCP/IP”，添加ip地址与端口号。



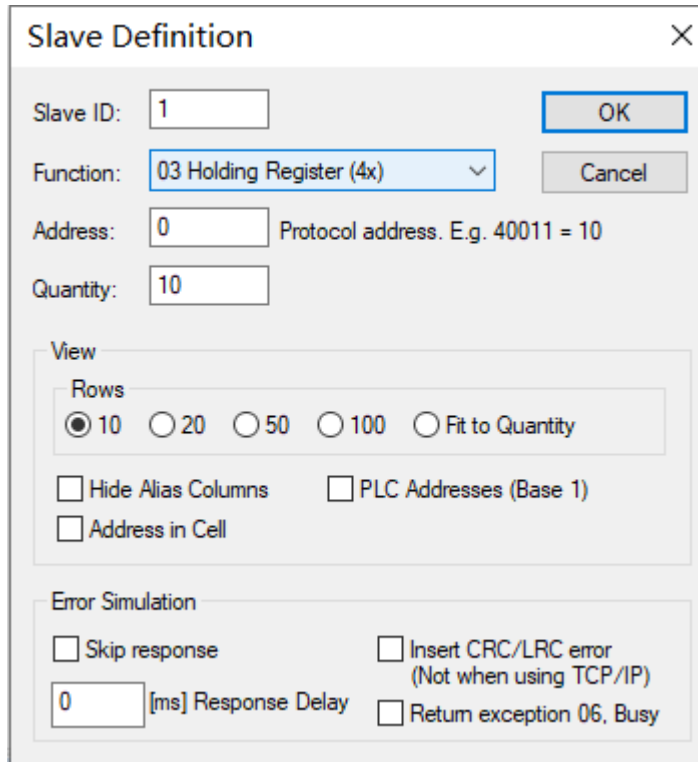
步骤4 添加新的slave配置页，并进行点位配置。



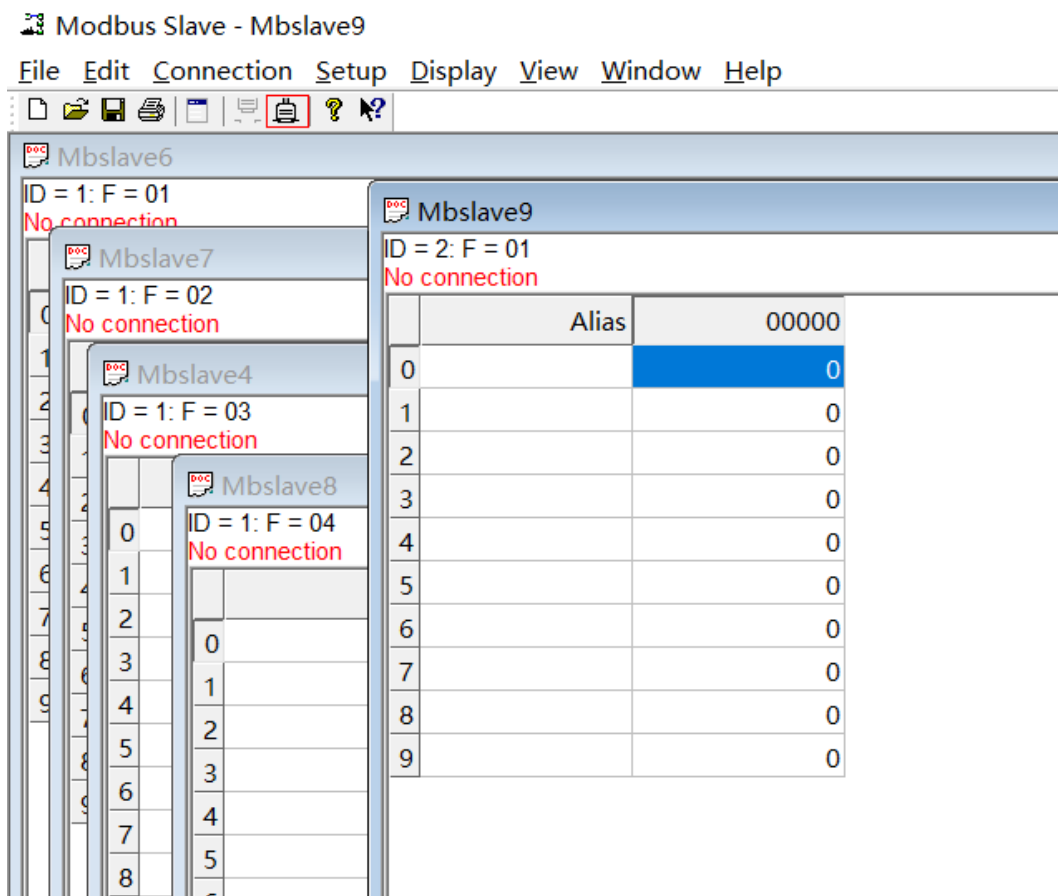
步骤5 单击鼠标右键进入slave配置页面。



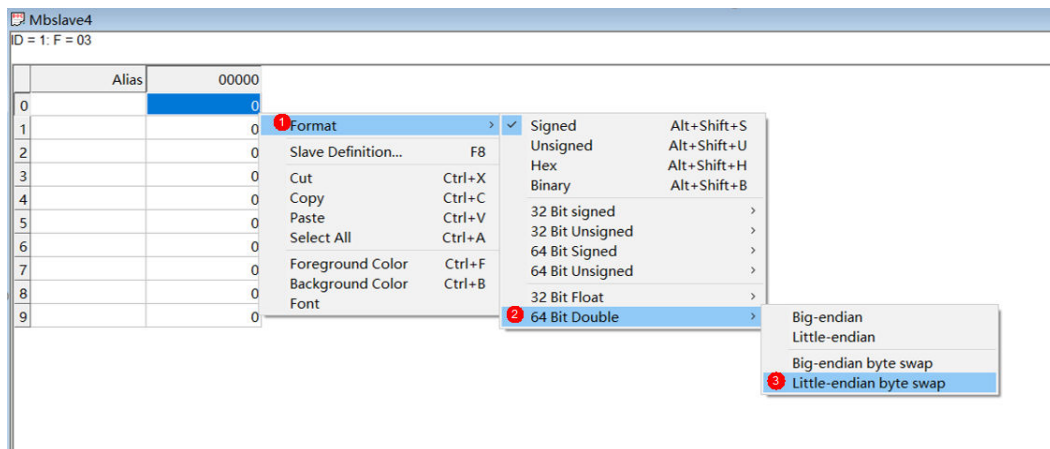
步骤6 配置从站号，寄存器类型，起始地址，寄存器数量等信息。



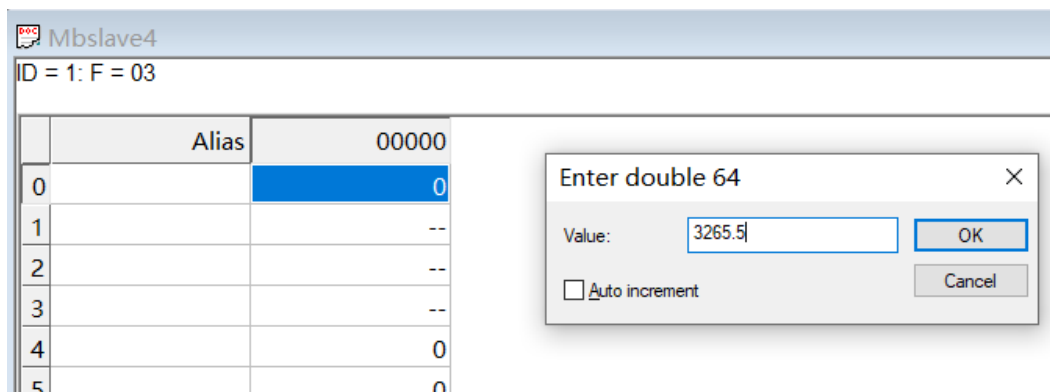
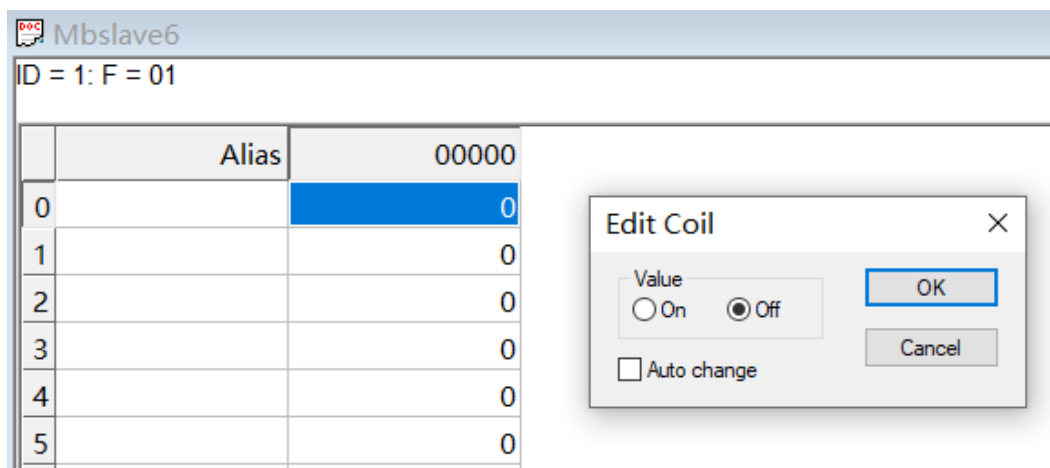
步骤7 打开“Modbus Slave”连接。(ID代表从站号，F为Function代表寄存器类型)



步骤8 单击鼠标右键，选择修改寄存器数据格式。（寄存器通过临接组合的方式表示复杂类型数据，协议为大端格式协议，寄存器内字节间与寄存器间顺序可交换。）



步骤9 双击对应位置修改对应点位数据（可设置数据自动改变）。



----结束

9.2.3 部署边缘数采应用

部署边缘数采应用

如自定义边缘数采应用，可参考[添加驱动应用](#)。

步骤1 在左侧导航栏选择“边缘节点 > 节点管理”“选择之前创建的边缘节点“test_loT10”，单击“详情”进入节点详情页。

步骤2 选择左侧“应用模块 > 模块管理”，单击“部署应用”填写基本信息后“确认”。



- 功能用途: 数据采集
- 边缘应用: ot_dc_modbus_tcp
- 选择版本: 当前最新的版本

步骤3 边缘应用部署任务下发成功后，应用状态为已停止。

模块id	模块名称	所属应用	版本	应用类型	实例状态	操作
sys_edge_hub	sys_edge_hub	\$edge_hub		系统必选	运行中	升级
sys_edge_agent	sys_edge_agent	\$edge_agent		系统必选	运行中	升级
sys_ot_dc_modbus_tcp	sys_ot_dc_modbus_tcp	\$ot_dc_modbus_tcp	1.0-0-standard-x86	系统可选	已停止	启用 删除 升级

📖 说明

数采应用部署成功后或者升级数采应用之后，应用状态都显示为已停止，只有在数采配置启用了数采连接后，应用状态会变更为运行中。

---结束

9.2.4 设备建模&发放

创建产品

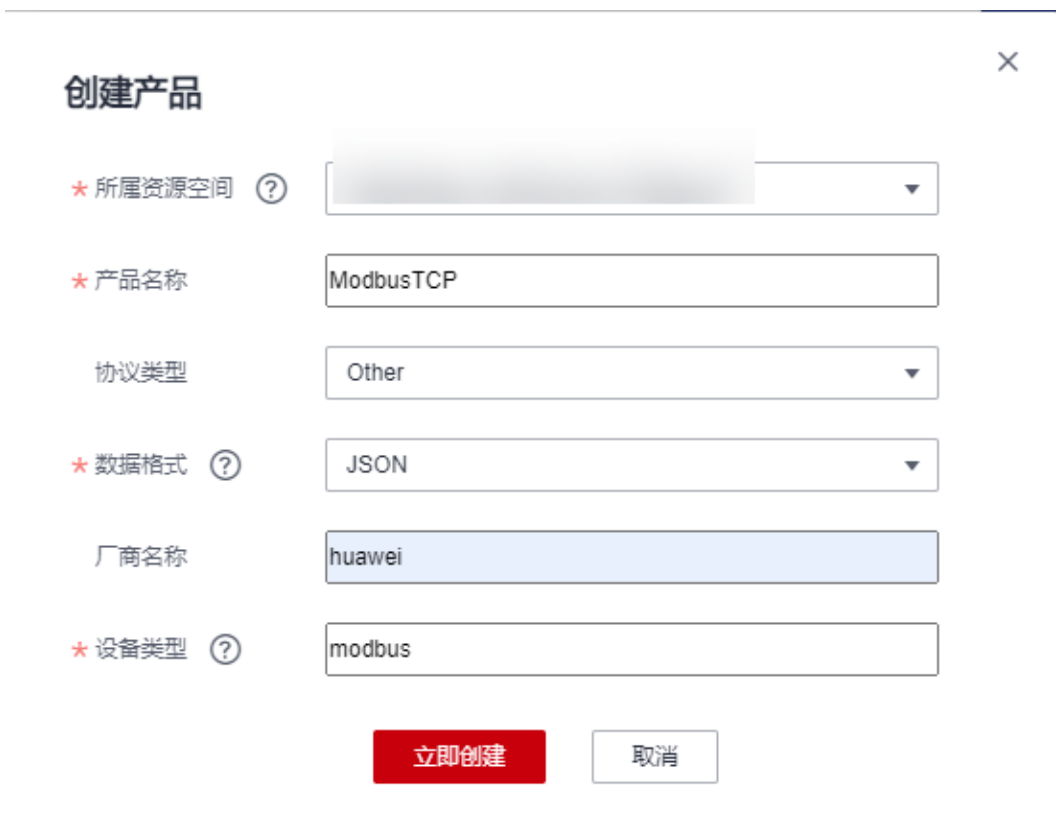
使用物联网平台的第一步就是在控制台创建产品。产品是设备的集合，是指某一类具有相同能力或特征的设备的合集被称为一款产品。

步骤1 访问[IoT边缘](#)，单击“管理控制台”进入IoT边缘控制台。

步骤2 在左侧导航栏选择“边缘节点 > 设备建模”进入页面。



步骤3 单击右上角“创建产品”，按照页面提示填写信息，然后单击“确定”。



参数名称	说明
所属资源空间	与之前创建边缘节点时选择的资源空间保持一致。
产品名称	自定义，如ModbusTCP。
协议类型	Other。 说明 自定义应用使用OT数采能力时，建模协议类型必须选择Other。
数据格式	默认值，JSON。
厂商名称	自定义，如Huawei。
设备类型	自定义。

步骤4 单击“设备建模 > 产品”，选择刚才创建的产品“ModbusTCP”，单击详情进入产品详情页面。

步骤5 在“模型定义”页签下，单击“自定义模型”。



说明

支持将本地开发的产品模型导入到平台。

导入步骤：选择“设备建模 > test_loT10”，单击“上传模型文件”，在弹窗中添加模型文件，单击“确定”完成上传。



步骤6 填写“服务ID”、“服务类型”，单击“确定”。

- 服务ID：自定义，如ModbusCollection
- 服务类型：自定义，如ModbusCollection

新增服务 ×

* 服务ID

* 服务类型

服务描述

步骤7 展开服务列表，单击“添加属性”，根据实际需要分别添加属性。

新增属性 ×

* 属性名称 必选

属性描述

* 数据类型

* 访问权限

取值范围 -

步长

单位

步骤8 模型定义完成，参数信息预览如下图所示。

属性名称	数据类型	是否必填	访问方式	描述	操作
bool读写	boolean	false	可读,可写		编辑 删除
bool只读	boolean	false	可读		编辑 删除
integer只读	int	false	可读		编辑 删除
integer读写	int	false	可读,可写		编辑 删除
decimal读写	decimal	false	可读,可写		编辑 删除
decimal只读	decimal	false	可读		编辑 删除

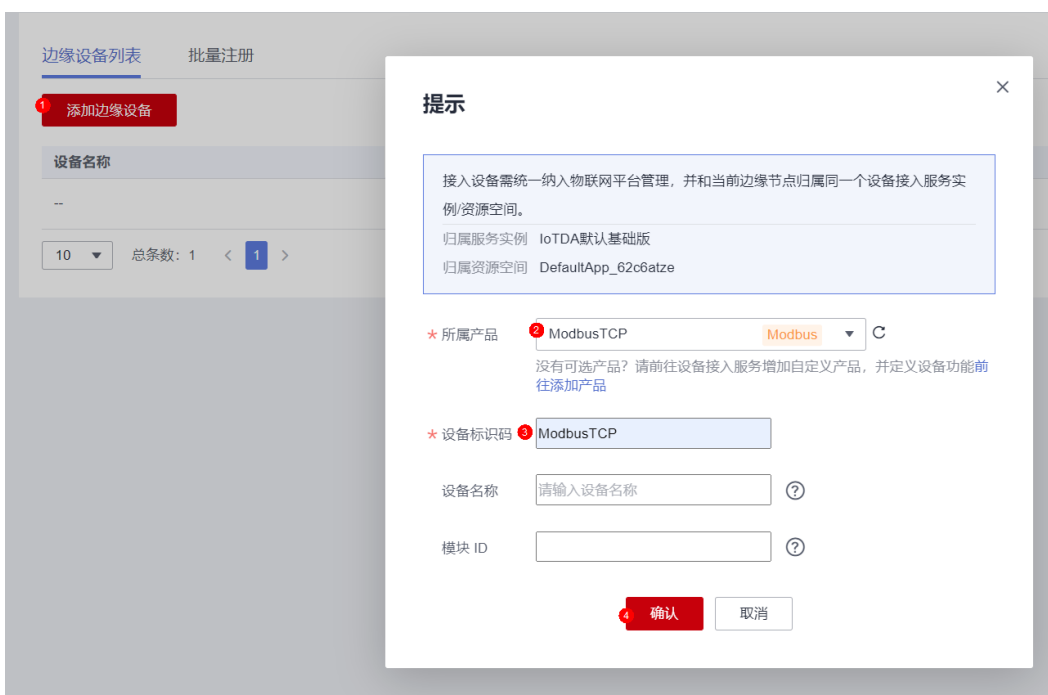
----结束

添加边缘设备

用户创建产品后，需要在边缘节点上添加对应的边缘设备，用于接收边缘节点实际采集的OT设备数据。

步骤1 在左侧导航栏选择“边缘节点 > 节点管理”选择之前创建的边缘节点“test_loT10”，单击节点进入详情页。

步骤2 选择左侧导航栏“边缘设备”，单击“添加边缘设备”，根据提示填写参数后，单击“确认”。



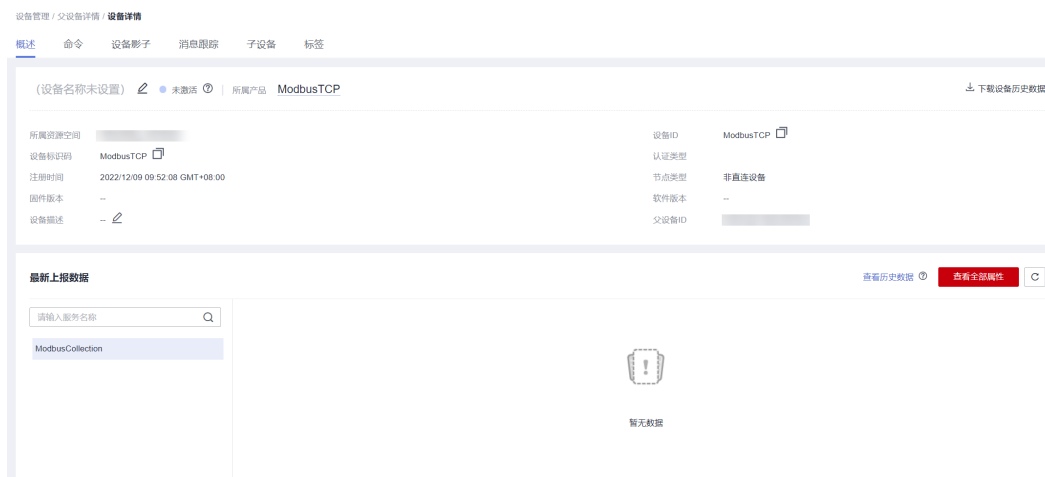
参数名称	说明
所属产品	选择步骤3创建的产品“ModbusTCP”。
设备标识码	自定义。只允许英文字母、数字，下划线，中划线。最大长度64。
设备名称	自定义。设备名称长度为1~256位，只能由数字,字母,中文,逗号,括号,#,.,,?,!,-,_,@,%,&,!组成。

参数名称	说明
模板ID	为空。 说明 使用Other数采能力设备必须选择协议类型为Other的产品，且模块id必须为空。

步骤3 添加设备成功后，可以在设备列表中查看“ModbusTCP”设备状态，并管理边缘设备。此时设备处于未激活状态，需完成设备部署接入后，可在物联网平台查看设备上报的数据。



单击对应的“设备ID”跳转至“设备接入服务控制台”进行设备管理。



----结束

9.2.5 OT 数采配置

添加数采模板

数采模板是描述数采配置项的元数据，数采配置包括数据源配置元数据、点位采集配置元数据。当用户添加自定义数采应用时，可通过添加自定义数采模板完成元数据的定义。

应用数据源模板导入

数据源模板描述了数据源配置项的元数据，用户在配置数据源之前，需要先导入数据源模板。IoT边缘服务当前已经预置通用数据源模板，可直接使用，系统也支持用户自定义导入模板。以下为用户添加自定义模板的详细流程及关键配置信息解释。

步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。

步骤2 在左侧导航栏选择“边缘节点 > 应用管理”进入页面，

步骤3 选择“驱动应用 > 数采模板管理 > 添加数采模板”。



步骤4 添加文件，单击“确认”。



- 本地准备一份数采模板文件，文件内容合规。
- IoT边缘云平台当前已经预制多个业界主流通用数采模板，可以通过导入通用模板后使用。
- 支持用户自定义导入模板，导入模板后缀.json，满足json格式规范，样例及字段解释如下：

示例

```
{
  "tpl_id": "sys_general_modbus_tcp",
  "name": "ModbusTcp通用数采模板",
  "description": "ModbusTcp通用数采模板",
  "datasource_meta": {
    "config_tabs": [{
      "key": "connection_info",
      "name": "连接信息",
      "description": "连接信息",
      "type": "object",
      "config_items": [{
        "key": "endpoint",
        "name": "服务端点",
        "description": "服务完整URL",
        "data_type": "string",
        "required": true,
        "crypted": false,
        "max_length": 128,
        "example": "127.0.0.1:502"
      }]
    }]
  }
}
```

```
},
{
  "key": "array_info",
  "name": "组配置信息",
  "description": "组配置信息",
  "type": "array",
  "quantity": 20,
  "config_items": [
    {
      "key": "slave_id",
      "name": "从站号",
      "description": "从站号",
      "data_type": "int",
      "required": false,
      "crypted": false,
      "max_length": 256,
      "example": 1
    },
    {
      "key": "coil_status_valid_address",
      "name": "线圈有效地址范围",
      "description": "线圈有效地址范围",
      "data_type": "string",
      "required": false,
      "crypted": false,
      "max_length": 65535,
      "example": "0x0-0x10,0x1a-0x2f"
    },
    {
      "key": "input_status_valid_address",
      "name": "输入状态有效地址范围",
      "description": "输入状态有效地址范围",
      "data_type": "string",
      "required": false,
      "crypted": false,
      "max_length": 65535,
      "example": "0x0-0x10,0x1a-0x2f"
    },
    {
      "key": "holding_registers_valid_address",
      "name": "保持寄存器有效地址范围",
      "description": "保持寄存器有效地址范围",
      "data_type": "string",
      "required": false,
      "crypted": false,
      "max_length": 65535,
      "example": "0x0-0x10,0x1a-0x2f"
    },
    {
      "key": "input_registers_valid_address",
      "name": "输入寄存器有效地址范围",
      "description": "输入寄存器有效地址范围",
      "data_type": "string",
      "required": false,
      "crypted": false,
      "max_length": 65535,
      "example": "0x0-0x10,0x1a-0x2f"
    },
    {
      "key": "byte_swap",
      "name": "寄存器高低位交换",
      "description": "寄存器高低位交换",
      "data_type": "bool",
      "required": false,
      "crypted": false,
      "max_length": 256,
      "default": false,
      "example": true
    }
  ]
}
```

```
        "key": "register_swap",
        "name": "寄存器顺序交换",
        "description": "寄存器顺序交换",
        "data_type": "bool",
        "required": false,
        "crypted": false,
        "max_length": 256,
        "default": false,
        "example": true
    }
}
],
"default_values": {
    "drivername": "ModbusTcp"
}
},
"point_meta": {
    "data_type_enums": [
        "bool",
        "short",
        "ushort",
        "int",
        "uint",
        "long",
        "ulong",
        "float",
        "double"
    ],
    "config_items": [{
        "key": "slave_id",
        "name": "从站号",
        "description": "modbus传感器从站号",
        "data_type": "int",
        "required": true,
        "crypted": false,
        "max_length": 256,
        "example": 1
    },
    {
        "key": "register_type",
        "name": "寄存器类型",
        "description": "寄存器类型",
        "data_type": "string",
        "required": true,
        "crypted": false,
        "max_length": 256,
        "enum_value": "Coil_Status|Input_Status|Holding_Registers|Input_Registers",
        "example": "Coil_Status"
    },
    {
        "key": "register_address",
        "name": "起始地址",
        "description": "寄存器起始地址",
        "data_type": "string",
        "required": true,
        "crypted": false,
        "max_length": 256,
        "example": "0x0"
    },
    {
        "key": "cycle",
        "name": "采集周期",
        "description": "点位采集周期",
        "data_type": "int",
        "required": true,
        "crypted": false,
        "max_length": 65535,
        "example": 10000
    }
    ]
}
```

```

    }
  ]
}
}

```

表 9-3 OT 数采模板

key	类型	描述
tpl_id	String	数采模板标识，租户下唯一 pattern: '^ [a-zA-Z0-9_-]*\$' 1-64。
name	String	数据源模板名称，支持中英文 1-64。
description	String	数据源模板描述，支持中英文， 0-128。
datasource_meta point_meta	Object Object	数据源配置元数据 点位采集配置元数据（内可配置 data_typr_enums用来限制点位数据类型）。

表 9-4 datasource_meta

key	类型	描述
config_tabs	Array	数据源配置表格列表。
default_values	Object	数据源默认的值，通常用于描述默认 的驱动协议。

表 9-5 config_tabs

key	类型	描述
key	String	数据源配置表格键值，pattern: 'connection_info collection_paras'。
name	String	数据源配置表格名称，1-64。
description	String	描述，0-255。
config_items	Array	配置项列表。
type (非必填)	String	配置表类型，可为“object”或 “array”，默认为“object”，为 array”时可传输多组配置项列表。

key	类型	描述
quantity (非必填)	Int	配置项列表组数量, 当配置表类型为“array时有效”。

表 9-6 config_items

key	类型	描述
key	String	配置项key, 1-32。
name	String	默认项名称, 1-64。
description	String	配置项描述, 0-128。
data_type	String	配置项类型, 受限于“data_type_enum”(非必填), 支持short ushort int int64 uint long ulong float double bool。
required	Boolean	配置项是否必传项。
crypted	Boolean	配置项是否加密。
max_length	Integer	当data_type= string时配置有效, 校验输入的字符串最大长度。
example	String	配置项举例, 值在界面输入框作为示例灰色显示, 0-256。
enum_value (非必填)	String	配置项枚举量, 可生成下拉框, 减少手工输入错误。

表 9-7 default_values

key	类型	描述
drivername	String	用于描述默认的驱动协议, 如OPCUA。

----结束

创建数采连接

- 步骤1 访问[IoT边缘](#), 单击“管理控制台”进入IoT边缘控制台。
- 步骤2 在左侧导航栏选择“边缘节点 > 节点管理”, 节点名称选择“test_IoT10”进入页面。
- 步骤3 在左侧导航栏选择“数采配置 > 创建数采连接”。



步骤4 填写“数采连接参数”后单击“确认”保存，此时配置仅在云平台存在，未下发到边缘，此时数采连接的“同步状态”为“待同步”。



- 通道标识：ModbusTCP。
- 通道名称：ModbusTCP。
- 驱动模块：sys_ot_dc_modbus_tcp。
- 服务端点：模拟器或Modbus服务器网关的ip地址+端口号,中间用“:”分隔，如“127.0.0.1:502”（半角）。
- 配置信息：用于配置从站的寄存器地址可用范围与寄存器顺序等信息，共可以添加20组。不填写配置信息代表默认对应从站点位映射地址全部有效，同时对应从站字节序与寄存器顺序默认为大端。

×

创建数采连接

驱动模块 sys_ot_dc_modbus_tcp 暂无?前往 [应用管理](#)

连接信息

服务端点 127.0.0.1:502

配置信息 ?

▼ **配置信息 1** 🗑️

从站号

线圈有效地址范围

输入状态有效地址范围

保持寄存器有效地址范围

输入寄存器有效地址范围

寄存器高低位交换

寄存器顺序交换

+ 添加配置信息

确认
取消

表 9-8 配置信息说明

配置项	说明
从站号	配置项的从站号，范围0-255，需唯一。
线圈有效地址范围	地址量为十六进制，格式“0x”+地址量，起始地址与结束地址用“-”连接，地址范围间用“,”分隔，范围不可重叠。例如“0x0-0x20,0x2a-0xc5,0xd0-0xff”（半角）。
输入状态有效地址范围	地址量为十六进制，格式“0x”+地址量，起始地址与结束地址用“-”连接，地址范围间用“,”分隔，范围不可重叠。例如“0x0-0x20,0x2a-0xc5,0xd0-0xff”（半角）。
保持寄存器有效地址范围	地址量为十六进制，格式“0x”+地址量，起始地址与结束地址用“-”连接，地址范围间用“,”分隔，范围不可重叠。例如“0x0-0x20,0x2a-0xc5,0xd0-0xff”（半角）。

配置项	说明
输入寄存器有效地址范围	地址量为十六进制，格式“0x”+地址量，起始地址与结束地址用“-”连接，地址范围间用“,”分隔，范围不可重叠。例如“0x0-0x20,0x2a-0xc5,0xd0-0xff”（半角）。
寄存器高低位交换	一个保持寄存器或输入寄存器对应2字节的寄存器空间，配置该项为true则交换对应寄存器内字节顺序。例如“byte1,byte2”交换后顺序为“byte2,byte1”。该项配置仅对32位及以上的数据有效。
寄存器顺序交换	Modbus协议默认为大端，配置该项为true则交换对应寄存器间顺序。例如“r1,r2,r3,r4”交换后顺序为“r4,r3,r2,r1”。该项配置仅对32位及以上的数据有效。

📖 说明

服务端点：填写Modbus服务器所在的网关地址，本实验中为[准备数据源处](#)的Modbus salve 模拟器。

寄存器高低位交换、寄存器顺序交换：Modbus协议默认寄存器顺序为大端，高低位交换与顺序交换同时配置为true时，数据部分传输顺序变为小端模式。以64bits的数据底层字节顺序为“0x1,0x2,0x3,0x4,0x5,0x6,0x7,0x8”为例，配置后数据转换结果如[表9-9](#)。

未添加配置信息，默认所有地址范围有效且寄存器顺序为大端模式。可以以从站为单位配置对应有效地址范围与寄存器顺序。

表 9-9 寄存器数据交换配置表

配置	数据顺序展示
高低位交换false，顺序交换false（原始数据）	0x1,0x2,0x3,0x4,0x5,0x6,0x7,0x8
高低位交换true，顺序交换false	0x2,0x1,0x4,0x3,0x6,0x5,0x8,0x7
高低位交换false，顺序交换true	0x7,0x8,0x5,0x6,0x3,0x4,0x1,0x2
高低位交换true，顺序交换true	0x8,0x7,0x6,0x5,0x4,0x3,0x2,0x1

----结束

添加点位映射

将点位上报转换为设备属性上报，将设备属性读写转换为点位读写操作，将设备命令转化为点位写操作。

步骤1 访问[IoT边缘](#)，单击“管理控制台”进入IoT边缘控制台。

步骤2 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_IoT10”进入页面。

步骤3 在左侧导航栏选择“数采配置 > 选择对应的数采连接”。



步骤4 单击“添加点位”在“添加点位映射”弹窗中填写配置信息，如表9-10所示，然后单击“确认”。



表 9-10 添加点位映射

基本信息	
点位标识	点位的唯一标识。
点位名称	按照规范填写。
设备ID	边缘设备页签里添加设备的ID。

基本信息		
点位数据类型	当前支持如下数据类型： int(整型) uint(无符号整型) short(短整型) ushort(无符号短整型) long(长整型) ulong(无符号长整型) bool(布尔类型) float(单精度浮点型) double(双精度浮点型)	取值范围： -2147483648~2147483647 0~4294967295 -32768~32767 0~65535 -2147483648~2147483647 0~4294967295 true/false -3.4*10 ³⁸ ~+3.4*10 ³⁸ -1.7*10 ³⁰⁸ ~1.7*10 ³⁰⁸
设备属性	设备属性由物模型中service_id/property_name格式构成。	
采集配置		
从站号	点位所属从站号，范围0-255。	
寄存器类型	点位对应寄存器类型，枚举字符串值，取值范围如右侧框。	Coil_Status Input_Status Holding_Registers Input_Registers
起始地址	点位对应数据起始地址，十六进制，格式：“0x” + (地址量)。	例如：“0xa3”
采集周期	点位采集周期，最小200ms，低于200ms则默认为200ms。	范围0-65535

注意

不同点位请勿配置同一设备的相同属性。

步骤5 单击“下发配置”，把点位配置下发至边缘侧。配置下发成功后启用通道，可以看到通道连接状态变为“已连接”，此时数采连接的“同步状态”为“已同步”。



说明

- 修改点位信息后需重新单击下发配置，配置才会生效。
- 模块的运行状态由启用状态决定，当启用了数采连接后，模块会显示为运行中。

---结束

附录：批量导入/导出点位映射

批量导入点位映射

- 步骤1** 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_loT10”进入页面。
- 步骤2** 在左侧导航栏选择“数采配置”->对应的连接通道。
- 步骤3** 单击“导入点位”选择“批量添加”，单击“下载模板文件”，打开“模板文件”内容填写。



- 步骤4** 选择“添加文件”导入步骤3下载的文件后，单击“确认”完成批量导入点位映射。

----结束

批量导出点位映射

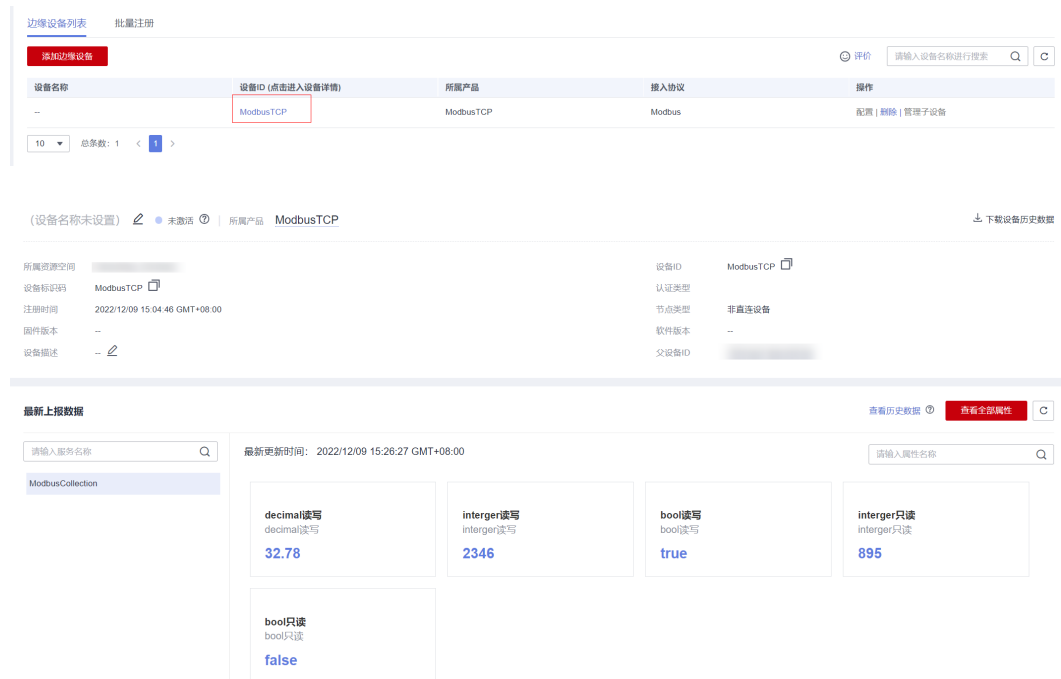
- 步骤1** 在左侧导航栏选择“边缘节点 > 节点管理”，节点名称选择“test_loT10”进入页面。
- 步骤2** 在左侧导航栏选择“应用模块 > 数采配置 > 对应的连接通道”。
- 步骤3** 单击“导出点位”完成批量导出点位。



----结束

9.2.6 查看采集结果

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 在左侧导航栏选择“边缘节点 > 节点管理”，选择之前创建的边缘节点“test_loT10”，单击节点进入详情页。
- 步骤3** 选择“边缘设备”页签，单击设备ID“ModbusTCP”，跳转至“设备接入服务控制台”查看采集的数据。



📖 说明

数采设备状态平台不感知，即统一显示为“未激活”，请进入设备详情页，查看最新上报数据。

----结束

9.3 常见问题

9.3.1 连接状态未连接

9.3.1.1 服务端点格式不正确

报错信息：

```
2022-12-09T08:15:28.480Z | ERROR | ModbusTCP/pkg/proxy/handler.go:75 | Illegal Endpoint : 127.0.0.1
```

排查位置：

修改数采连接

通道信息

- * 通道标识: ModbusTCP
- * 通道名称: ModbusTCP

连接信息

- * 服务端点: 127.0.0.1:502

配置信息

⊕ 添加配置信息

确认 取消

正确格式：

正确服务端点格式为ip+ “.” (半角)+端口号。

9.3.1.2 配置信息配置有误

报错信息：

```
2022-12-09T08:19:27.885Z | ERROR | ModbusTCP/pkg/proxy/handler.go:63 | Illegal Range limit config : 123-234
```

排查位置：

请确保已部署启动应用,前往应用模块

数采 报警

创建数采连接 批量删除 查看任务列表

<input type="checkbox"/>	连接源名称	连接状态	数量	驱动模块	最近修改时间	同步状态	启用状态	操作
<input type="checkbox"/>	ModbusTCP	已连接	5	user_ModbusTCP	2022/12/09 16:19:27 GMT+0...	已同步	<input checked="" type="checkbox"/>	下发配置 编辑 删除

10 总数: 1 < 1 >

配置信息 ?

配置信息 1 🗑️

从站号

线圈有效地址范围

输入状态有效地址范围

保持寄存器有效地址范围

输入寄存器有效地址范围

寄存器高低位交换

寄存器顺序交换

确认 取消

正确格式:

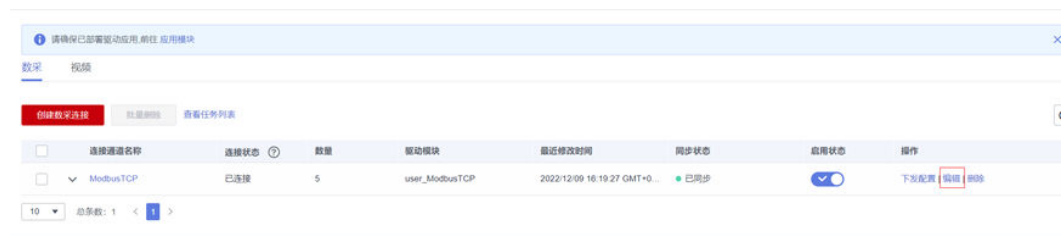
地址为十六进制数，格式为“0x”+地址量。起始地址与结束地址以“-”（半角）连接，范围之间用“,”（半角）分隔。地址范围不可重复，否则报地址范围重复错误。

9.3.1.3 有效地址范围重复

报错信息:

```
2022-12-09T08:24:30.210Z | INFO | ModbusTCP/pkg/proxy/handler.go:152 | repeated address range
```

排查位置:



配置信息 (?)

配置信息 1 🗑️

从站号	<input type="text" value="1"/>
线圈有效地址范围	<input type="text" value="0x0-0x10,0x9-0x13"/>
输入状态有效地址范围	<input type="text" value="0x0-0x10,0x1a-0x2f"/>
保持寄存器有效地址范围	<input type="text" value="0x0-0x10,0x1a-0x2f"/>
输入寄存器有效地址范围	<input type="text" value="0x0-0x10,0x1a-0x2f"/>
寄存器高低位交换	<input type="text" value=""/>
寄存器顺序交换	<input type="text" value=""/>

正确格式:

地址为十六进制数, 格式为“0x”+地址量。起始地址与结束地址以“-”(半角)连接, 范围之间用“,”(半角)分隔。地址范围不可重复。

9.3.2 网络问题排查

9.3.2.1 连接拒绝

现象描述:

```
2022-12-12T06:46:15.319Z | ERROR | ModbusTCP/pkg/modbus/worker/connect.go:41 | Connect failed : dial tcp connect: connection refused
```

报错连接拒绝, ping对应ip可以ping通。

```
root@ecs-23fb:~# ping 192.168.1.100
PING 192.168.1.100: 56(84) bytes of data:
64 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=43.8 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=43.9 ms
64 bytes from 192.168.1.100: icmp_seq=3 ttl=64 time=43.9 ms
64 bytes from 192.168.1.100: icmp_seq=4 ttl=64 time=43.9 ms
64 bytes from 192.168.1.100: icmp_seq=5 ttl=64 time=43.9 ms
64 bytes from 192.168.1.100: icmp_seq=6 ttl=64 time=43.8 ms
```

排查方法:

优先考虑对端服务器未开启或ip地址/端口填写错误, 无相应程序监听对应端口。检查网关服务器是否正常开启, 若正常开启则检查网关对应ip与端口是否与配置相符合。

9.3.2.2 连接超时

现象描述:

```
2022-12-12T06:47:55.369Z | ERROR | ModbusTCP/pkg/modbus/worker/connect.go:41 | Connect failed : dial tcp 192.168.1.100:502: i/o timeout
```

报i/o timeout, ping对应ip不通。


```
root@ecs-23fb:~# ping
PING
PING: 56(84) bytes of data.
^C
--- ping statistics ---
9 packets transmitted, 0 received, 100% packet loss, time 8200ms
```

排查方法:

对端网络不可达, 通常为连接配置填写错误。

9.3.3 点位采集失败

9.3.3.1 点位数据类型与寄存器类型不匹配或点位未下发

报错信息:



```
2022-12-09T08:40:48.164Z | ERROR | ModbusTCP/pkg/client/subscribe.go:65 | point p6 hasn't published yet , or dumped for illegal configuration
```

排查位置:

根据报错信息, 查看对应点位的数据类型与寄存器类型是否匹配, 也可查看上一次下发配置时的metadata报错信息。

```
2022-12-12T08:34:43.297Z | DEBUG | ModbusTCP/pkg/modbus/worker/worker.go:124 | 10.38.110.50:502 Telv frame close
2022-12-12T08:34:43.297Z | INFO | ModbusTCP/pkg/proxy/point.go:66 | report l metas , detail : map[p6:[bad dumped point for invalid configuration]]
2022-12-12T08:34:43.297Z | DEBUG | ModbusTCP/pkg/client/subscribe.go:65 | syscall read block close by ctrl cancel
```

若数据类型与寄存器类型配置无误, 且检查上次下发配置是报错信息无对应点位, 可初步判断点位未下发。点位未下发时, 单击下发配置后刷新, 再次单击采集查看结果。

连接通道名称	连接状态	数量	驱动模块	最近修改时间	同步状态	启用状态	操作
ModbusTCP	已连接	2	user_ModbusTCP	2022/12/12 10:41:38 GMT+0	已同步	<input checked="" type="checkbox"/>	下发配置 编辑 删除

正确格式:

表 9-11 参数

寄存类型	支持的数据类型
Coil_Status	Bool
Input_Status	Bool
Holding_Registers	Short,Ushort,Int,Uint,Float,Long,Ulong,Double
Input_Registers	Short,Ushort,Int,Uint,Float,Long,Ulong,Double

9.3.3.2 点位超出有效地址范围

报错信息：

无报错信息，但设备详情页面周期上报数据，单击采集后有数据。

排查位置：

找到该点位对应的从站配置信息，查看该点位地址是否在配置的有效范围。

⚠ 注意

一个值需要多寄存器存储时需要考虑是否所有涉及寄存器都满足有效范围，例如：数据类型为double的值需要占4个16bits的寄存器，起始地址为0x0则需要满足0x0-0x3都在有效地址范围内。

配置信息 1 🗑

从站号

线圈有效地址范围

采集配置

从站号	1	寄存器类型	Coil_Status	起始地址	0x10
数据类型	bool	采集周期	10000		

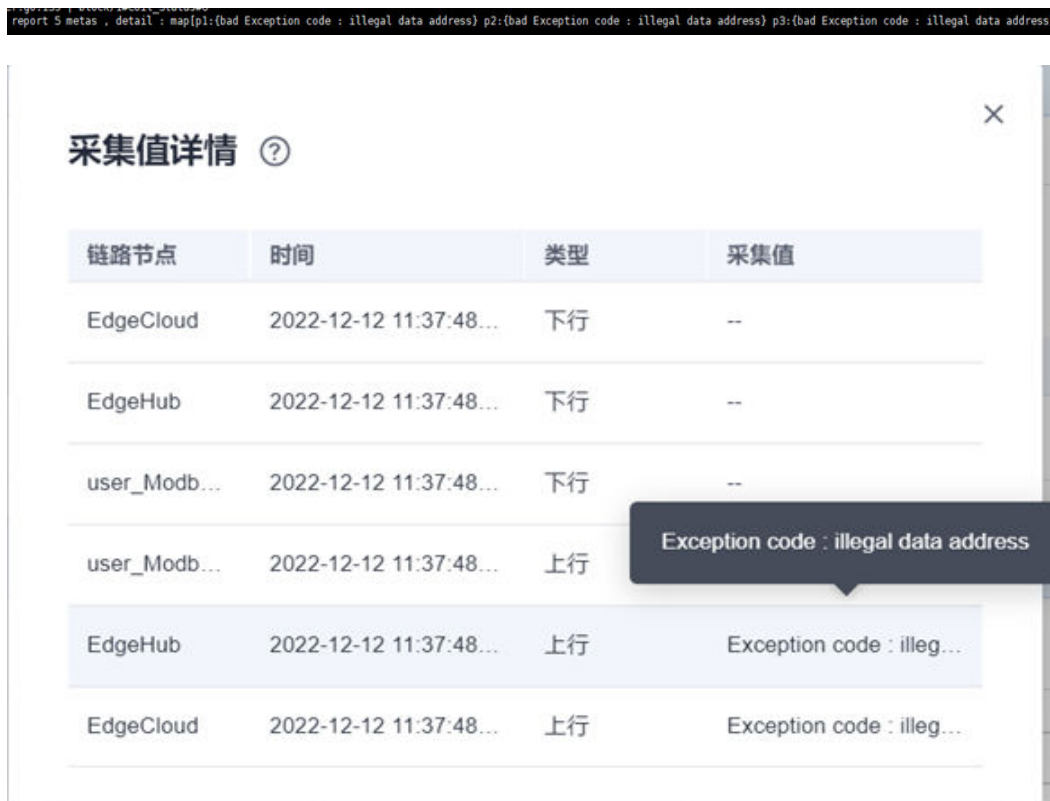
如有效地址是相邻的两段，则应合并为一个有效地址段，否则同样会导致该错误。
如：有效地址范围为“0x0-0xa,0xb-0x13”应合并为“0x0-0x13”。

正确格式：

满足所有需要周期采集的数据地址都在有效地址范围。

9.3.3.3 坏点

报错信息：



单击采集时数据大部分点位正常采集，但个别点位报错为Exception Code（结合各厂家自己定义的报错方式与协议Exception Code分析），优先考虑出现坏点导致采集失败。

排查位置：

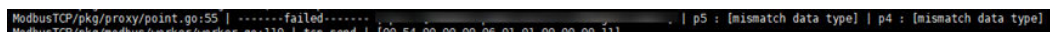
参考告警接收到的报错信息，或驱动日志里的metadata信息，找到对应的点组。挨个单击采集，直到找出上报信息为Exception Code的点位，即为坏点。

正确格式：

可通过修改坏点所在的从站配置中的有效地址范围缩小采集范围，避开坏点进行周期采集让业务正常运行后排查设备故障点位并修复。

9.3.4 点位下写失败

9.3.4.1 下写数据类型与配置类型不匹配



现象描述：

下写失败，日志写数据报错mismatch data type。

问题排查：

下写数据类型应与配置数据类型相同，如配置为bool则只能有true或false两个值，下写123（int）则会导致此类错误，反之亦然。排查时根据报错信息找到对应点位，查看其数据类型与下写数据类型是否一致。

正确格式：

下写数据类型与配置类型一致。

9.3.4.2 下写不可写寄存器

现象描述:

下写失败, 如图报错couldn't write for register type_____。

```
2022-12-12T09:31:07.693Z | ERROR | ModbusTCP/pkg/client/builtin.go:43 | couldn't write point for register type : Input_Registers  
2022-12-12T09:31:12.694Z | INFO | ModbusTCP/pkg/proxy/point.go:54 | -----failed----- | p5 : [couldn't write Input_Registers ]
```

问题排查:

排查报错对应点位配置, 查看是否存在写不可以寄存器的行为。

正确格式:

Modbus协议中的寄存器分为四种, Coil_Status与Holding_Registers为可读可写, Input_Status与Input_Registers为只读寄存器。应尽量避免对不可写寄存器进行写操作。

9.3.4.3 未知数据类型下写

现象描述:

数据下写失败。报错如图 abandon point set for illegal write 。

```
ModbusTCP/pkg/proxy/point.go:55 | -----failed----- | p1 : [Abandon point set for illegal write ]
```

问题排查:

下写不支持的数据类型导致, 查看数据类型是否符合要求。

正确格式:

下写数据量可以为bool类型、整型、小数, 不可以下写其他值如string类型等。

9.3.4.4 下写数据结果与预期不一致

现象描述:

下写数据为小数, 写成功后发现下写值为整型。

问题排查:

排查配置对应点位时, 数据类型是否为整型, 若配置整型但下写小数, 则会取整。

正确格式:

将对应配置修改为正确的对应参数。

9.3.5 修改数据配置并下发, 数据源配置未更新

修改数据源配置并下发后, 驱动会接收到更新过的数据影子。在连接建立成功的前提下, 若修改端点信息或组配置信息下发后修改项未生效, 优先考虑配置错误的可能。

- 采集未避开坏点

现象描述: 在出坏点问题后, 通过主动采集的方式感知坏点地址并配置组配置信息内的对应寄存器有效地址后, 坏点信息仍继续上报、相邻点无法成功上报数据。

排查方式: 查看驱动日志可以看到Illegal Range limit config报错日志, 也可以自行排查组配置处坏点所在从站寄存器有效地址范围是否按对应格式配置。

```
2023/01/09 11:50:20 ERROR | Illegal Range limit config : 12-465
2023/01/09 11:50:20 ERROR | Invalid config , update process failed
```

正确格式: 地址为十六进制数, 格式为“0x”+地址量。起始地址与结束地址以“-” (半角) 连接, 范围之间用“, ” (半角) 分隔。地址范围不可重复。

- **更新端点信息后驱动仍采集原ip地址**

现象描述: 更新端点信息后驱动仍采集原ip地址

排查方式: 查看日志能看到Illegal EndPoint对应错误日志。

```
2023/01/09 11:43:44 ERROR | Illegal Endpoint : 
2023/01/09 11:43:44 ERROR | Invalid config , update process failed
```

解决方法: 按ip+端口号的格式配置端点信息。

10 基于 IoT 边缘实现业务应用自定义配置

10.1 实验简介

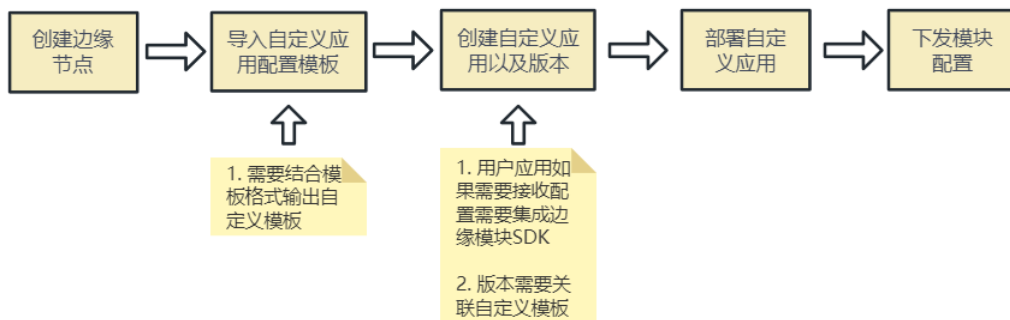
业务应用自定义配置，提供了在IoT边缘控制台，能够基于模板对自己所管理的模块进行配置。

通过本实验，您将能够：

- 了解并掌握创建节点的详细流程。
- 了解并掌握自定义业务应用模板的创建、更新等管理动作，以及自定义模板的书写格式。
- 了解并掌握业务应用与业务模板建立关联关系的操作步骤。
- 了解并掌握通过业务应用模板进行模块自定义数据的配置以及下发。

整体操作流程介绍：

图 10-1 整体流程



前提概要

1. **前提：**体验需要三方配合，首先要开通IoTEdge，用于节点管理和应用管理，再开通IoTDA，用于设备接入与管理，还得有边缘服务器作为节点被IoTEdge纳管。此外还需开通SWR或者OBS服务（取决于驱动应用的开发方式），用于存储已开发并想部署在边缘节点上运行的Docker镜像或者进程包。

2. **边缘服务器**：第一点提到的IoTEdge, IoTDA, SWR, OBS是华为云服务，使用华为云租户账号直接免费开通可用，边缘服务器属于硬件，可以购买低配的ECS代替，其EIP则作为边缘服务器的节点IP地址。
3. **在IoTEdge上注册节点**：在购买的ECS上执行安装脚本命令，即自动部署好2个系统应用，agent用于节点应用管理，hub用于数据通信。
4. 业务应用需要集成边缘SDK，才能够接受云端下发的配置，集成SDK方式见[插件侧开发](#)。
5. **进程包插件**：是集成了边缘SDK的应用程序打包而成，可作为边缘节点的自定义应用部署到边缘服务器上，以进程方式运行，可在节点上进行设备数据协议解析、子系统数据解析、数据业务处理、智能计算等。进程包开发方式见[基于IoT边缘实现进程包镜像插件部署](#)。
6. **Docker镜像**：是集成了边缘SDK的应用程序打包成的容器镜像，可作为边缘节点的自定义应用部署到边缘服务器上运行，可在节点上进行设备数据协议解析、子系统数据解析、数据业务处理、智能计算等。镜像包开发方式见[基于IoT边缘实现容器应用镜像插件部署](#)。

10.2 操作步骤

10.2.1 创建边缘节点

前提条件

- 已注册[华为云](#)账号，且已完成实名认证。
- 已开通[IoT边缘服务](#)和[IoTDA设备接入服务](#)。
- 在创建边缘节点前，检查配置要符合[硬件规格](#)要求，如磁盘、内存和docker版本，切换root用户执行安装命令。

注册专业版节点

- 步骤1** 访问[IoT边缘](#)，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 选择左侧导航栏“边缘节点 > 节点管理”单击右上角“注册节点”。
- 步骤3** 根据页面提示填写参数，勾选右下角“《华为云服务等级协议》”单击“下一步”配置节点数据。

图 10-2 专业版

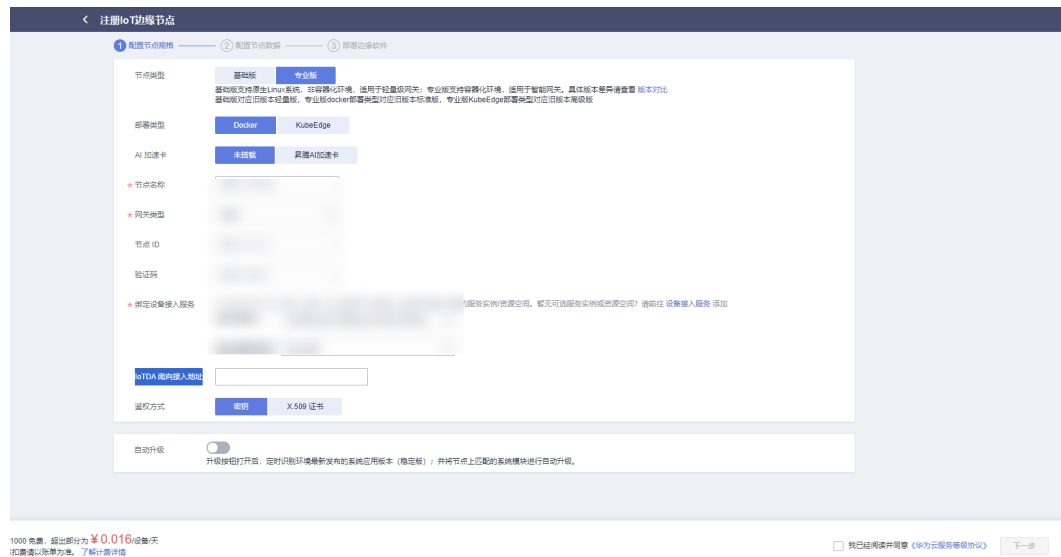


表 10-1 节点规格参数说明

参数名称	参数说明
节点类型	专业版。
部署类型	Docker。
AI加速卡	<ul style="list-style-type: none"> 未搭载：边缘节点未使用AI加速卡时选择。 昇腾AI加速卡：支持华为昇腾AI加速设备，边缘节点的服务器搭载相关昇腾AI加速设备时选择。
昇腾驱动地址	可选，AI加速卡选择昇腾AI加速卡时需要保证安装边缘节点的服务器上已经安装昇腾设备驱动，并根据前端提示填入驱动地址。若改值为空，默认填入Atlas300驱动默认安装地址： <code>/usr/local/Ascend/driver/lib64/</code> 。
节点名称	自定义节点名称，如“test”。 说明 节点名称可以在：“节点概览”-“基本信息”中编辑修改。
网关类型	other。
节点ID	自定义（可选填写）。只允许中文，英文字母、数字、中划线（-），下划线（_），最小长度1，最大长度64。 说明 边缘节点ID自定义(不输入默认生成)。
验证码	输入自定义验证码（可选填写）。验证码，执行安装边缘软件命令时与云端通讯的凭证。
设备归属实例	选择默认归属实例。
设备归属资源空间	选择默认资源空间，或为本实验单独新建一个资源空间，如edge。

参数名称	参数说明
IoTDA南向接入地址	选择网关归属的IoTDA实例的南向设备mqtt接入地址，也可以选择自定义的南向接入地址，支持“ip:端口”或“域名:端口”形式。
鉴权方式	密钥：默认生成。 X.509证书：请参考： 基于MQTT.fx的X.509证书接入指导 。 警告 鉴权方式为“密钥”的节点，请勿在IoTDA界面重置节点的密钥，重置密钥会导致网关与IoTDA通信异常，且不可恢复。
指纹	证书文件生成的字符串（可选填写）。只能为40位十六进制字符串或64位六进制字符串。
本地路径	输入证书文件和密钥文件在设备上的存储路径，只允许字母、数字、下划线（_）和斜线（/），以/开头并且以.pem或.crt结尾。
自动升级	可选，开启后节点将自动升级系统应用模块。 注意 升级过程中可能导致节点数据上传中断。

步骤4 填写配置节点数据信息后，单击“立即创建”。

图 10-3 配置节点数据

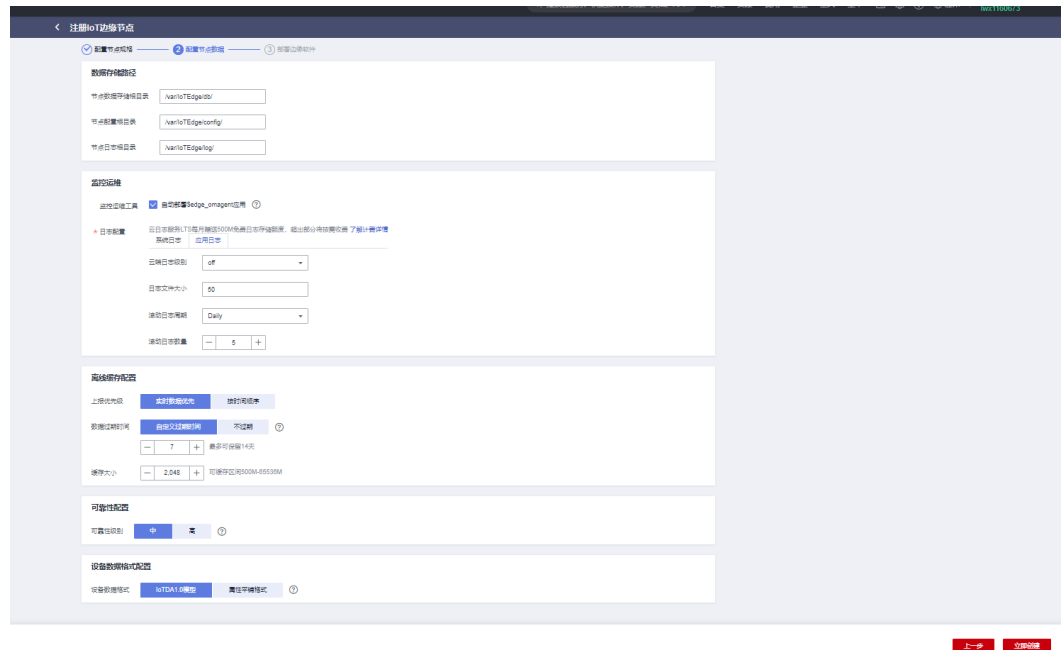


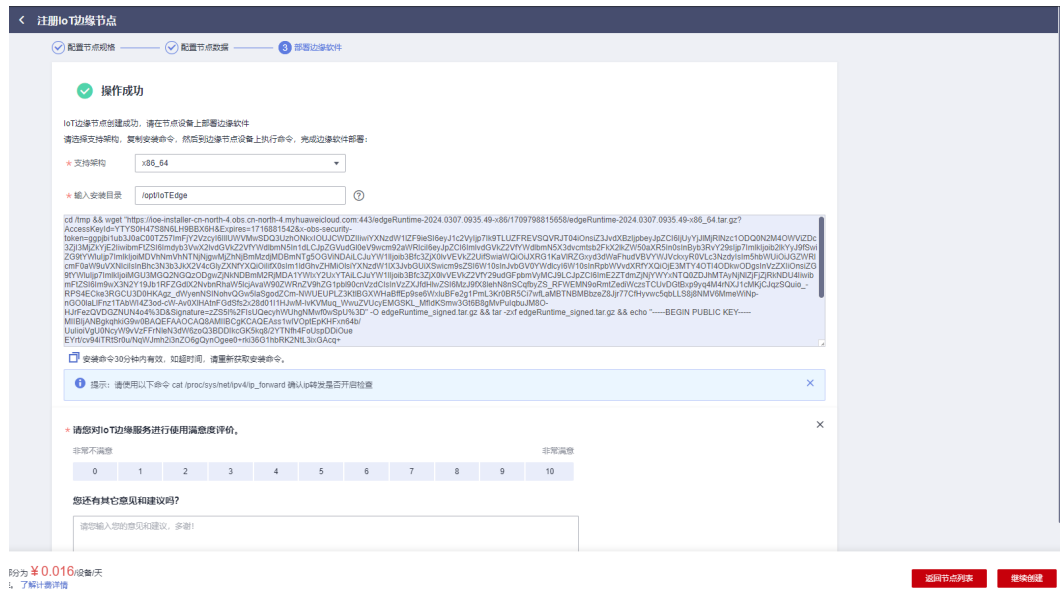
表 10-2 专业版节点信息

参数名称	说明
数据存储路径	默认（允许修改存储路径）。
监控运维	默认勾选“监控运维工具”。
\$edge_omagent应用	负责边缘节点远程监控和运维，提供日志上报、远程SSH、CPU等指标上报、文件上传下载能力。
日志配置	<p>系统日志：边缘节点上已部署的系统应用产生的日志。</p> <p>应用日志：边缘节点上已部署的用户定义应用所产生的日志。</p> <ul style="list-style-type: none"> • 云端日志级别：on, off, trace, debug, info, warn, error, fatal。其中，on表示上传所有日志至云端，off表示不上传日志至云端，其余可选项代表按日志级别采集，选择后将只采集目标级别及以上的日志（模糊匹配）。日志级别的严重程度等级由低到高依次为：trace, debug, info, warn, error, fatal。 • 日志文件大小：日志文件大小限制，单位MB，默认50，取值范围10-1000。某个日志文件如果达到大小限制，则会将文件转储保存至日志目录下。 • 滚动日志周期：日志转储周期，可选项：daily, weekly, monthly, yearly。某个日志文件如果达到转储周期，则会将文件转储保存至日志目录下。日志文件大小和滚动日志周期是同时生效的，满足任何一个条件都会进行日志转储。 • 滚动日志数量：日志文件转储个数，默认5，取值范围1-10。边缘节点保存的转储日志数量如果达到限制，则会删除最早的转储文件。 <p>注意</p> <ol style="list-style-type: none"> 1. 当前“云端日志级别”对日志等级采取模糊匹配的策略，若日志消息中存在与日志级别标识相同的关键字，则也可能被采集。 2. 若要开启“云端日志级别”中的trace, debug, info, warn, error, fatal选项，则必须满足以下三个前置条件： <ol style="list-style-type: none"> 1. 目标节点是基础版节点或部署类型为Docker的专业版节点。需注意，部署类型为KubeEdge的专业版节点不支持此功能。 2. 已安装\$edge_omagent应用，且版本不低于1-1-15-xxx。 3. 应用日志需遵循如下规范打印，以便进行过滤： <ul style="list-style-type: none"> ○ 日志级别标识需打印为（大小写均可）：trace、debug、info、warn、error、fatal。 ○ 每条日志消息以YYYY-MM-DD日期格式作为前缀。 <p>说明</p> <p>专业版（部署类型为Docker）和基础版节点只涉及日志级别这一个参数，其他参数只对专业版（部署类型为KubeEdge）有效。</p>
离线缓存配置	可对离线节点上报的数据进行缓存，定义缓存周期（可配置为不过期）跟缓存大小、上报的优先级、路由转发是否开启离线缓存。仅当上报优先级为“按时间顺序”时支持路由转发开启离线缓存。

参数名称	说明
可靠性配置	可靠性级别分为“中”和“高”2种。中等级别：当应用上报属性和点位的频率超过流控值时，超过部分会被丢弃；高级别：当应用上报属性和点位的频率超过流控值时，通过控制模块的发送速度，降低上报频率，报文不会被丢弃。（注意：高可靠性级别的配置需要定制应用的配合，同时会造成性能的损失）。
设备数据格式配置	设备的数据格式：即单网关节点中设备的标准数据格式，包含 IoTDA1.0物模型格式和属性平铺格式。IoTDA1.0物模型中包含了设备、服务和属性三层结构，具体参考IoTDA网关子设备属性上报报文格式。属性平铺格式，包含设备、属性2层结构，属性结构可以包含自定义结构的拼接，在json中作为一项平铺展开，具体格式请参考 步骤6设备数据格式配置 。 须知 属性平铺格式功能，需要edge_hub应用1-1-38-xx版本及以上支持

步骤5 弹出如下对话框，您可以“返回节点列表”也可以“继续创建”。

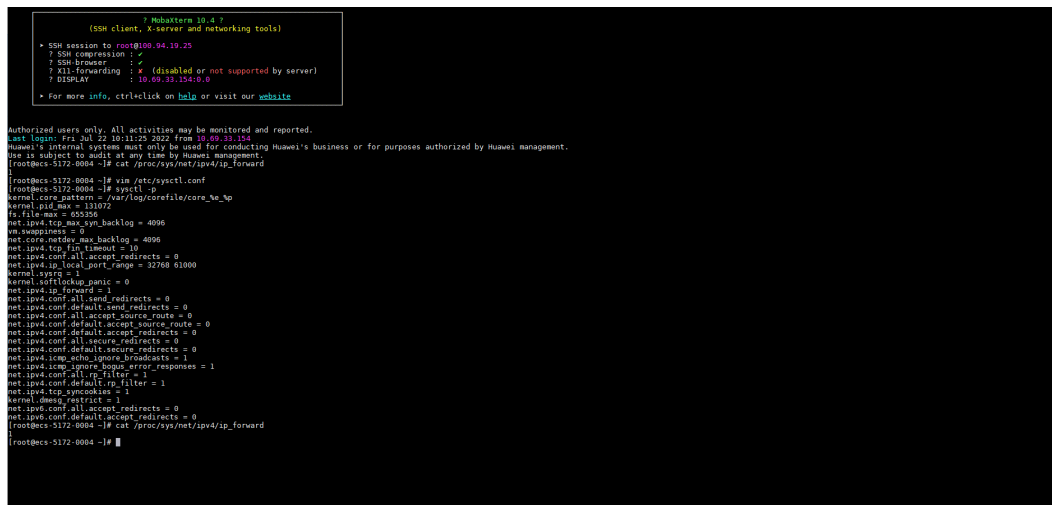
图 10-4 注册成功



说明

安装部署节点前，请确认机器的IP转发已开启并且已开启自启动，以及7883和8943端口开放。其中，IP转发规则配置如下：

图 10-5 IP 转发规则配置



1. 检查机器ip_forward in/proc文件系统的值。
`cat /proc/sys/net/ipv4/ip_forward`
 若值等于0，则需要添加配置IP转发规则。
 2. 检查配置文件/etc/sysctl.conf是否包含net.ipv4.ip_forward = 1。
 如果配置文件中没有net.ipv4.ip_forward参数项，请新增一行net.ipv4.ip_forward = 1。
`vim /etc/sysctl.conf`
 按“i”进入编辑模式，修改完成后按“Esc”退出编辑模式，然后按“Shift” + “;”组合键，输入wq并回车，保存修改并退出。
 3. 重新加载此文件的值。
`sysctl -p`
 4. 检查7883和8943端口是否开放，如果是华为云服务器，请参考：[安全组](#)。
- 结束

安装专业版节点

步骤1 在“节点管理”的节点列表选择您需要安装的节点，单击右侧“安装”，获得提示。

图 10-6 安装

节点名称	节点类型	资源包类型	资源包ID	状态	主机名/网络	操作
	专业版	--	--	未安装		删除 安装
	专业版	--	--	离线	172.30.7.64,172.17.0.1,1...	删除
	专业版	--	--	已安装		删除

步骤2 选择支持架构和安装目录，然后单击 复制安装命令，使用SSH工具以root用户登录边缘节点服务器后台系统，执行安装命令。

图 10-9 安装命令执行成功

```
2024-05-28T14:44:29.581321 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:PENDING, reason:(zero), ]
2024-05-28T14:44:31.632105 | INFO | saveBasePathInfo | 78 | config base path start.
2024-05-28T14:44:31.632291 | INFO | saveBasePathInfo | 83 | config log path:/var/IoTEdge/db/, config_path:/var/IoTEdge/config/
2024-05-28T14:44:31.632457 | INFO | saveBasePathInfo | 95 | config base path success.
2024-05-28T14:44:31.632535 | INFO | addAgentCustomConfigs | 201 | addCustomConfigs start.
2024-05-28T14:44:31.632603 | INFO | addAgentCustomConfigs | 205 | addCustomConfigs end.
2024-05-28T14:44:31.632654 | INFO | pullImage | 319 | start pull image, deploy type is cloud
2024-05-28T14:44:35.888519 | INFO | createModule | 76 | create Module, name:sys_edge_agent
2024-05-28T14:44:35.939013 | INFO | createModule | 84 | module created, CreateResp[ moduleId:sys_edge_agent, moduleName:sys-edge-agent, appId:sedge_agent, appVersion:test-tpm-2-x86, containerId:4b3e43ce7eac9bc282b994377abfbf48a284389a36856f3cacc0338743036e, processID:(zero), type:docker, ]
2024-05-28T14:44:35.939153 | INFO | startModule | 160 | startModule moduleId:sys_edge_agent
2024-05-28T14:44:42.390994 | INFO | listModule | 259 | listModule
2024-05-28T14:44:42.440775 | INFO | checkModulesDo | 340 | start check modules, time:1
2024-05-28T14:44:42.440908 | INFO | listModule | 259 | listModule
2024-05-28T14:44:42.491163 | INFO | printModulesAndReport | 404 | ***success:1 failed:0
2024-05-28T14:44:42.491339 | INFO | reportDeploymentState | 81 | reportDeploymentState, deploymentState:DeploymentState[ node_id:zzzzz, deploy_id:1716878669308, deploy_state:RUNNING, reason:(zero), ]
2024-05-28T14:44:42.542192 | INFO | reportEvent | 158 | Begin reporting event: {"events":[{"resource":"node.installation","location":"zzzzzz","event":"succeeded","event_severity":"Info","event_time":"20240528T064442Z","notify_data":{"host_tag":"DEFAULT","edge_node_id":"zzzzzz"}}]}
2024-05-28T14:44:42.592933 | INFO | reportModuleStatus | 138 | reportModuleStatus, modulesInfo:ModuleStatusInfo[! host_tag:DEFAULT, deploy_id:1716878669308, modules:[ModuleStatusInfo[ module_id:sys_edge_agent, app_id:sedge_agent, app_version:test-tpm-2-x86, container_state:RUNNING, liveness_state:(zero), update_time:1716878669305, ]]]
2024-05-28T14:44:43.643689 | INFO | checkBridgeNetwork | 566 | begin checking bridge network
2024-05-28T14:44:39.624137 | INFO | install | 494 | install edge finash.
2024-05-28T14:44:30.524559 | INFO | stop | 545 | edgeinstatter stop.
/root
```

步骤3 单击“我知道了”，等待边缘节点的状态变为“在线”，表示该节点已安装并连接成功。

步骤4 您可以单击“节点名称”，查看该节点详细信息，关于节点详情描述可参见[节点概览](#)。

----结束

10.2.2 导入自定义业务应用配置模板

添加业务应用配置模板

业务应用配置模板是描述业务应用自定义配置项的元数据，应用配置包括基本配置项元数据、默认数据。当用户添加自定义应用时，可通过添加自定义应用配置模板完成元数据的定义。

应用数据源模板导入

数据源模板描述了数据源配置项的元数据，用户在配置数据源之前，需要先导入数据源模板。IoT边缘服务当前已经预置通用数据源模板，可直接使用，系统也支持用户自定义导入模板。以下为用户添加自定义模板的详细流程及关键配置信息解释。

步骤1 访问[IoT边缘](#)，单击“管理控制台”进入IoT边缘控制台。

步骤2 在左侧导航栏选择“边缘节点 > 应用管理”进入页面，

步骤3 选择“业务应用 > 业务模板管理 > 添加业务模板”。

图 10-10 添加业务模板



步骤4 添加文件，单击“确认”。

图 10-11 添加业务模板文件



- 本地准备一份业务模板文件，文件内容合规。
- IoT边缘云平台当前已经1个系统应用的配置模板，如果需要对对应的应用进行配置需要导入通用模板后使用。
- 支持用户自定义导入模板，导入模板后缀.json，满足json格式规范，样例及字段解释如下：

示例

```
{
  "tpl_id": "app_general_tpl",
  "name": "应用模板",
  "description": "应用模板",
  "config_tabs": [{
    "key": "connect_info",
    "name": "连接信息",
    "description": "连接信息",
    "type": "object",
    "config_items": [{
      "key": "endpoint",
      "name": "服务端点",
      "description": "服务完整URL",
      "data_type": "string",
      "required": true,
      "crypted": false,
      "max_length": 128,
      "example": "127.0.0.1:502"
    }]
  }],
  {
    "key": "channels",
    "name": "通道配置信息",
    "description": "通道配置信息",
    "type": "array",
    "quantity": 20,
    "config_items": [{
      "key": "device_id",
      "name": "设备id",
      "description": "和Edge网关交互的鸿蒙设备id",
      "data_type": "string",
      "required": true,
      "crypted": false,
      "max_length": 200,
      "example": "e08a4c6953dc193405c2c5a2e8bf2c4"
    }]
  }
}
```

```

    },
    {
      "key": "name",
      "name": "name",
      "description": "通道名称",
      "data_type": "string",
      "required": true,
      "crypted": false,
      "max_length": 200,
      "example": "test_name"
    }
  ],
  "default_values": {
    "app_name": "user_app"
  }
}

```

表 10-3 业务应用配置模板

key	类型	描述
tpl_id	String	数采模板标识，租户下唯一 pattern: '^([a-zA-Z0-9_-]*)\$' 1-64
name	String	模板名称，支持中英文 1-64
description	String	模板描述，支持中英文，0-128
config_tabs	List<config_tab>	配置项元数据
default_values	Map<String, Object>	默认值,用于承载一些不需要动态配置的数据

表 10-4 config_tab

key	类型	描述
key	String	配置表格键值
name	String	配置表格名称，1-64
description	String	描述，0-255
config_items	List<config_item>	配置项列表
type	String	配置表类型，可为"object"或"array"，默认为"object"。为"array"时可传输多组配置项列表。
quantity	Int	配置项列表组数量，当配置表类型为"array"时有效。

表 10-5 config_item

key	类型	描述
key	String	配置项key, 1-32
name	String	默认项名称,1-64
description	String	配置项描述, 0-128
data_type	String	配置项类型, 支持short ushort int int64 uint long ulong float double bool
required	Boolean	配置项是否必传项
crypted	Boolean	配置项是否加密
max_length	Integer	当data_type= string时配置有效, 校验输入的字符串最大长度, 如果配置项需要加密, 请注意字段长度max_length应当满足加密后字段的长度要求, 因为实际加密后会将加密后的字段返回给前端展示, 如果按照加密前的最大长度进行判定会出现展示异常问题。
example	String	配置项举例, 值在界面输入框作为示例灰色显示, 0-256
enum_value (非必填)	String	配置项枚举量, 可生成下拉框, 减少手工输入错误。

表 10-6 default_values

key	类型	描述
app_name	String	用于配置一些默认值, 不需要进行动态配置的数据。(此项仅为示例)

----结束

10.2.3 新增自定义应用以及版本

这里添加您的自定义边缘应用。

前提说明:

1. 需要实现了一个集成了边缘模块SDK的自定义应用。sdk集成方法见[插件侧开发](#)。
2. 驱动应用已经上传至镜像仓或者上传至OBS。

操作步骤

步骤1 访问**IoT边缘**，单击“**管理控制台**”进入IoT边缘控制台。

步骤2 选择左侧导航栏“**边缘节点 > 应用管理**”进入页面，选择业务应用，单击“**添加应用**”。

应用名称：自定义边缘应用的名称。

功能用途：您可以根据实际情况选择。

数据处理：提供设备数据处理能力。

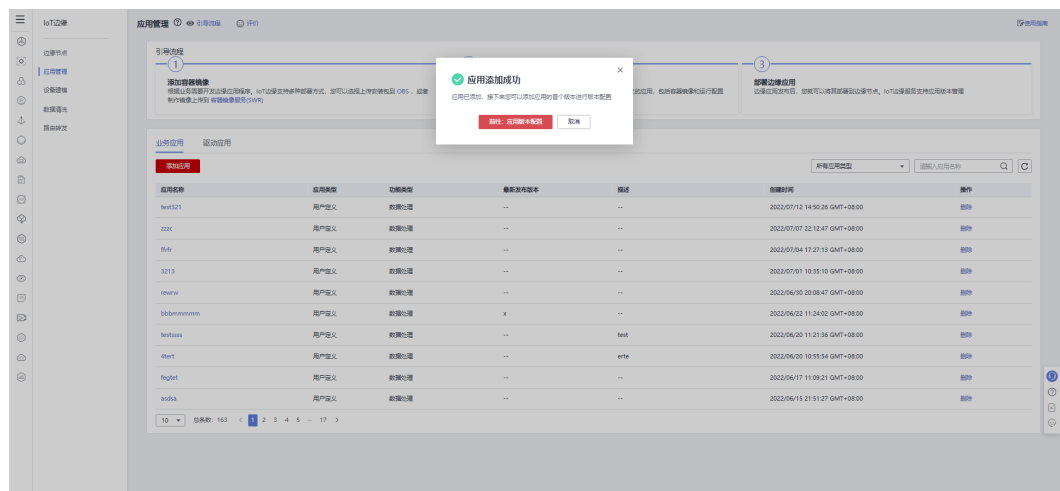
协议解析：提供设备接入能力。

本地子系统集成：提供IT子系统集成能力。

业务应用均支持进行配置模板的关联。



步骤3 添加完成后，单击“**前往：应用版本配置**”。



步骤4 第一步“**软件部署配置**”。

1. 选择部署方式，参考**表10-7**填写基本信息。

图 10-12 软件部署配置



表 10-7 部署配置

参数名称	说明
部署方式	<p>容器化部署：在边缘节点是以Docker容器运行；添加容器化部署的应用需先确保容器镜像已上传到容器镜像服务(SWR)，且设置其类型为“公开”，例如图10-13所示。</p> <p>安装包部署：在边缘节点是以进程形式运行；添加安装包部署的应用需先确保安装包已按照打包规范进行压缩打包，并上传到对象存储服务(OBS)。</p>
容器镜像	部署方式为容器化部署时，参考 表10-8 填写。
安装包部署	部署方式为安装包部署时，填写您此前上传到对象存储服务OBS中的安装包地址。

表 10-8 选择镜像输入方式

参数名称	说明
手动输入	输入镜像地址：统一在 容器镜像服务(SWR) 中管理。
单击选择	<p>选择镜像：选择需要部署的镜像，单击“确认”。</p> <ul style="list-style-type: none"> - 我的镜像：展示了您在容器镜像服务中创建的所有镜像。 - 他人共享：展示了其他用户共享的镜像。 <p>版本：选择需要部署的镜像版本。</p>

2. 选择容器规格，根据您的需要选择。

图 10-13 容器规格配置



3. 高级配置，按照如下步骤进行配置。

图 10-14 高级配置



- a. 运行命令（只有容器化部署的方式可以配置）

图 10-15 运行命令



- 运行命令：输入可执行的命令，例如“/run/start”。
 - 若可执行命令有多个，多个命令之间用换行进行分隔，且末尾不能有空行。
- 命令参数：输入控制容器运行命令参数，例如-port=8080。
 - 若参数有多个，多个参数以换行分隔，且末尾不能有空行。


说明

启动命令和参数复杂时，建议编写启动脚本，打包到镜像中，如start.sh，启动命令设置为 ./shart.sh。

若启动命令已经定义在dockerfile中，则该项可以不进行配置。

若应用部署实例异常，可参考配置指导：https://support.huaweicloud.com/iotedge_faq/iotedge_04_0029.html

b. 选项配置

打开勾选项 ，容器开启特权模块，将拥有访问主机设备的权限，可以访问主机上的设备（如GPU、FPGA），容器在特性模式下运行会存在一些安全风险，默认关闭该选项。

c. 环境变量

图 10-16 环境变量



容器运行环境中设定的一个变量。可以在部署应用时修改。

单击  添加环境变量，输入变量名称和变量值，添加环境变量。

说明

IoTEdge平台不会对用户输入的环境变量进行加密。

如果用户配置的环境变量涉及敏感信息，用户需要自行加密后再填入，并在应用中自己完成解密过程。

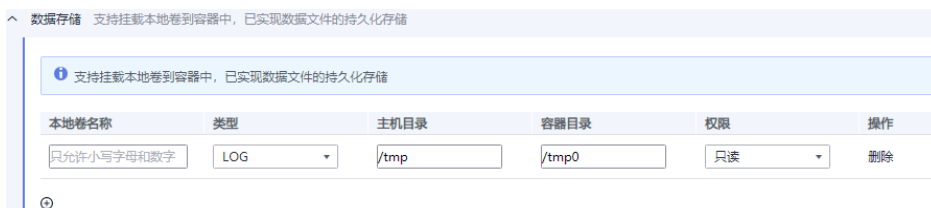
IoTEdge平台服务不提供任何加解密工具，如果您需要设置加密密文，可以使用其他平台的加解密工具。

d. 数据存储

卷是指容器运行过程中使用的存储卷。

将主机某个目录挂载到容器中。主机目录是一种持久化存储，应用删除后 hostPath 里面的内容依然存在于边缘节点本地硬盘目录中，如果后续重新创建应用，挂载后依然可以读取到之前写入的内容。

图 10-17 数据存储



说明

- 请不要挂载在系统目录下，如“/”、“/var/run”等，会导致容器异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响容器启动的文件，否则文件会被替换，导致容器启动异常，应用创建失败。
- 挂载高危目录的情况下，建议使用低权限账号启动，否则可能会造成宿主机高危文件被破坏。
- 本地卷的类型选择LOG/DB/CONFIG时，会给主机目录默认添加前缀：“/var/IoTEdge/{log|db|config}”。若需要将容器目录直接挂载到主机某个目录，本地卷的类型需选择为“其他”。
- 当挂载的目录磁盘满，会导致节点异常，无法使用，请及时清理。

e. 健康检查

健康检查是指容器运行过程中根据用户需要定时检查容器健康状况或是容器中负载的健康状况。

- 应用存活探针：应用存活探针用于探测容器是否正常工作，不正常则重启实例。当前支持发送HTTP请求和执行命令检查，通过检测容器响应是否正常。
- 应用业务探针：应用业务探针用于探测业务是否就绪，如果业务还未就绪，就不会将流量转发到当前实例。

图 10-18 健康检查



表 10-9 检查方式

参数名称	说明
HTTP请求检查	<p>向容器发送HTTP GET请求，如果探针收到2xx或3xx，说明容器是健康的。</p> <p>例如图10-19的配置，会在容器启动10秒（延迟时间）后，发送HTTP GET请求到“http://{实例IP}/health:8080”，如果在2秒（超时时间）内没有响应则视为检查失败；如果请求响应的状态码为2xx或3xx，则说明容器是健康的。</p> <p>说明 这里无需填写主机地址，默认直接使用实例的IP（即往容器发送请求），除非您有特殊需求。</p>

参数名称	说明
执行命令检查	<p>探针执行容器中的命令并检查命令退出的状态码，如果状态码为0则说明健康。</p> <p>例如图10-20的配置，会在容器启动10秒（延迟时间）后，在容器中执行“cat /tmp/healthy”命令，如果在2秒（超时时间）内没有响应，则视为检查失败；如果命令成功执行并返回0，则说明容器是健康的。</p>
TCP检查	<p>例如图10-21的配置，检查TCP端口是否正常可以访问。如果可以，说明容器是正常的。</p>

图 10-19 HTTP 请求检查

应用存活探针 检查容器是否正常，不正常则返回实例

* 检查方式 不配置 HTTP请求检查 执行命令检查 TCP检查

* 路径/端口

主机地址

HTTP协议 HTTP HTTPS

* 延迟时间 ① 秒

* 探测超时时间 秒

图 10-20 执行命令检查

应用存活探针 检查容器是否正常，不正常则返回实例

* 检查方式 不配置 HTTP请求检查 执行命令检查 TCP检查

* 执行命令

* 延迟时间 ① 秒

* 探测超时时间 秒

图 10-21 TCP 检查

* 检查方式 不配置 HTTP请求检查 执行命令检查 TCP检查

* 端口

* 延迟时间 ① 秒

* 探测超时时间 秒

步骤5 第二步“运行配置”。

图 10-22 端点和部署配置



1. 端点配置。

EdgeHub与其他模块之间的通讯采用MQTT作为消息总线，其中MQTT作为Broker可与各模块进行数据流转。

- 输入端点：模块订阅的指定正则MQTT Topic的后缀，EdgeHub向该模块发送数据时则根据该模块配置的输入端点构造MQTT Topic并发送数据。
- 输出端点：从模块端向EdgeHub发送数据时使用的MQTT Topic后缀；即EdgeHub接收该模块配置的输出端点对应的MQTT Topic的数据。

说明

只有配置了消息输入输出端点的应用实例化的模块才能够使用数据流转功能。

2. 部署配置。

表 10-10 部署配置

参数名称	说明
重启策略	<p>根据您的需要选择。</p> <ul style="list-style-type: none"> - 总是重启：当应用实例退出时，无论是正常退出还是异常退出，系统都会重新拉起应用实例。 - 失败时重启：当应用实例异常退出时，系统会重新拉起应用实例，正常退出时，则不再拉起应用实例。 - 不重启：当应用实例退出时，无论是正常退出还是异常退出，系统都不再重新拉起应用实例。

参数名称	说明
网络类型	<p>容器访问支持主机网络和端口映射两种方式。</p> <ul style="list-style-type: none"> - 主机网络 使用宿主机（边缘节点）的网络，即容器与主机间不做网络隔离，使用同一个IP。 - 端口映射 容器网络虚拟化隔离，容器拥有单独的虚拟网络，容器与外部通信需要与主机做端口映射。配置端口映射后，流向主机端口的流量会映射到对应的容器端口。例如容器端口80与主机端口8080映射，那主机8080端口的流量会流向容器的80端口。

步骤6 第三步配置确认，参考表10-11 填写基本信息

图 10-23 应用版本基本配置

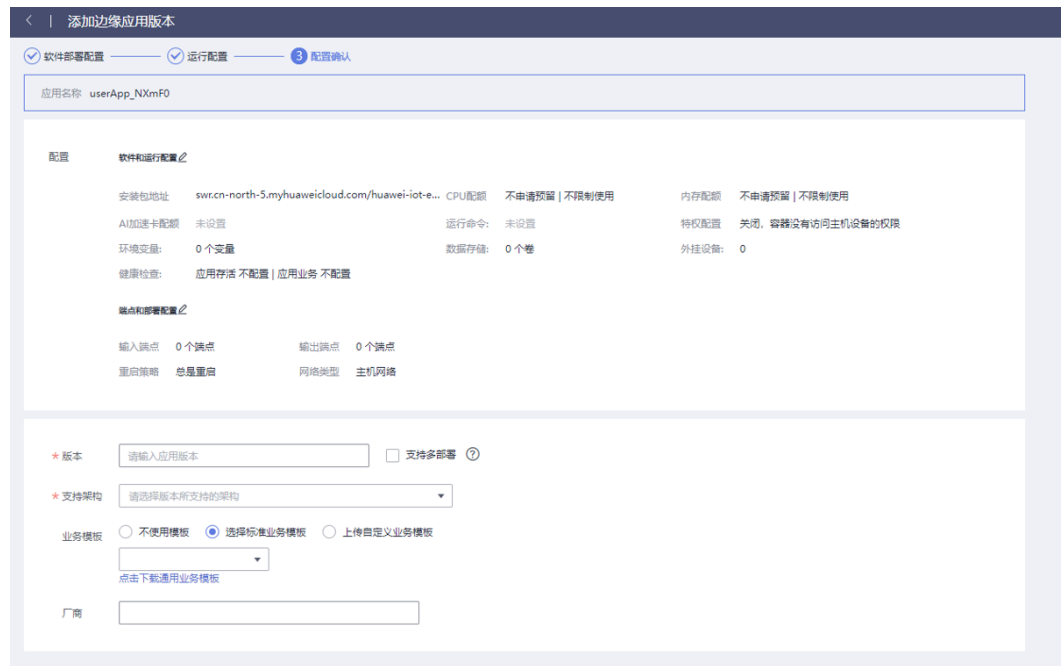


表 10-11 应用版本基本配置

参数名称	说明
应用名称	自定义边缘应用的名称。

参数名称	说明
功能用途	您可以根据实际情况选择： 数据处理：提供设备数据处理能力。 协议解析：提供设备接入能力。 本地子系统集成：提供IT子系统集成能力。 网关管理：扩展边缘节点的协议接入能力，当前支持Modbus，OPC-UA协议接入。 混合应用：包含数据处理、协议转换、子系统接入、网关管理、数据采集其中两个及以上功能的应用。 数据采集：提供OT数采能力。
应用描述	输入对应用的描述信息，方便与其他应用进行区分。
SDK版本	集成边缘SDK的版本号。
版本	自定义您的应用版本号。可选择是否“立即发布”。
支持架构	选择应用可支持的架构。有x86_64、arm32、arm64的架构，支持多选。
支持多部署	应用版本支持在一个边缘节点上部署多个实例。
业务模板	业务应用使用的配置模板，模板配置可参考 导入自定义业务应用配置模板 。也可以不关联模板。

步骤7 单击“确认添加”，完成应用的创建；单击“立即发布”，完成应用创建并发布新版本。

图 10-24 创建成功



步骤8 单击“返回应用管理列表”。

您可以看到应用类型为“用户定义”，区别于系统预置的应用。

图 10-25 应用管理列表

应用名称	应用类型	功能类型	最新发布版本	描述	创建时间	操作
cxxxx	用户定义	数据处理	1	--	2022/05/27 09:45:35 GMT+08:00	删除
ot_11111111	用户定义	数据处理	123	--	2022/05/26 17:10:40 GMT+08:00	删除
userApp_4mLioL	用户定义	数据处理	appVersion_TuRMe	--	2022/05/20 07:45:27 GMT+08:00	删除
App_1MOMI	用户定义	数据处理	--	--	2022/05/20 07:45:27 GMT+08:00	删除
testApp43522	用户定义	数据处理	--	test	2022/05/19 00:37:19 GMT+08:00	删除
testApp26988	用户定义	数据处理	--	test	2022/05/19 00:33:59 GMT+08:00	删除
testApp71586	用户定义	数据处理	--	test	2022/05/19 00:31:40 GMT+08:00	删除
testApp99064	用户定义	数据采集	--	test	2022/05/19 00:29:09 GMT+08:00	删除
testApp63361	用户定义	数据处理	--	test	2022/05/19 00:24:19 GMT+08:00	删除
testApp48475	用户定义	协议解析	--	test	2022/05/18 00:36:05 GMT+08:00	删除

----结束

10.2.4 部署自定义应用

在边缘节点安装成功后可部署边缘应用。

操作步骤

- 步骤1** 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。
- 步骤2** 选择左侧导航栏“边缘节点 > 节点管理”进入页面。
- 步骤3** 选择您的边缘节点，单击“节点名称”进入节点详情页。
- 步骤4** 在左侧导航栏选择“应用模块 > 模块管理”页签，单击“部署应用”。

图 10-26 部署边缘应用



- 步骤5** 根据页面提示填写参数信息后，单击“确认”。

图 10-27 部署应用



这里部署新添加的应用以及应用版本。

步骤6 弹出“操作成功”对话框，再单击“确认”返回部署边缘应用列表。

图 10-28 确认



步骤7 单击“刷新”，当应用的实例状态由“部署中”转为“运行中”表示部署成功。

模块ID	模块名称	所属应用	版本	应用类型	实例状态	操作
sys_edge_agent	sys_edge_agent	Edge_agent	1.0-0-standard-x86	系统必备	运行中	升级
sys_edge_hub	sys_edge_hub	Edge_hub	1.0-0-standard-x86	系统必备	运行中	升级
user_test_app_multiployment	user_test_app_multiployment	test_app_multiployment	1.0.0	用户定义	运行中	删除 升级

----结束

10.2.5 应用模块配置

应用部署成功之后，可以进行应用配置。

图 10-29 应用配置入口位置



根据应用是否关联业务模板呈现不同的配置页面。

关联业务模板的应用配置

如果对应的应用关联了业务应用模板，则呈现如下配置参数页面：

图 10-30 配置页面展示



图 10-31 添加自定义配置数据

✕

配置参数

模块 ID	所属应用	版本
user_test_app_config	test_app_config	1.0.0

*** 业务模块**

连接信息

*** 服务端点**

配置信息 ?

∨ **配置信息1** 🗑

*** 设备id**

*** 加密数据**

∨ **配置信息2** 🗑

*** 设备id**

*** 加密数据**

+ 添加配置信息

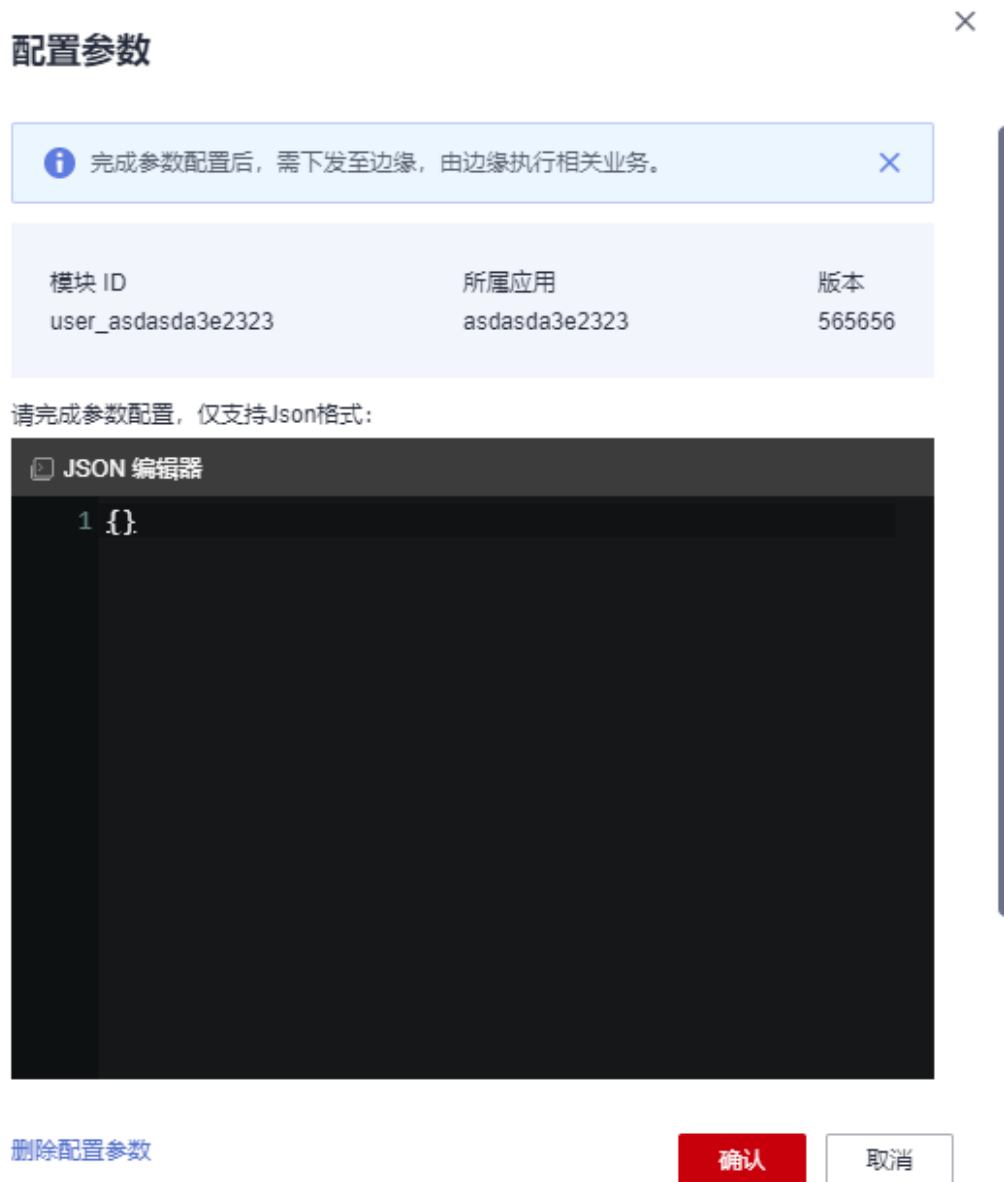
删除配置参数 确认 取消

这里在配置信息中添加两个配置项。单击确认进行配置保存并同时下发至边缘模块。

未关联业务模板的应用配置

如果该应用没有关联业务模板，则呈现如下图所示的配置页面：

图 10-32 未关联业务模板的应用配置



须知

1. 如果未关联业务模板，应用的配置数据为增量修改，本次修改会和之前的配置数据按key进行比对，不一致进行修改，本次新增key为新增数据。
2. 如果需要删除之前的配置项，需要将对应的配置项置为null。
3. 如果需要删除所有配置数据，可以直接单击“删除配置参数”进行删除。

例子 1：新增数据

原数据：

```
{  
  "key1": "value1"  
}
```

配置数据:

```
{  
  "key1": "value1",  
  "key2": "value2"  
}
```

结果:

```
{  
  "key1": "value1",  
  "key2": "value2"  
}
```

例子 2: 删除数据

原数据:

```
{  
  "key1": "value1",  
  "key2": "value2"  
}
```

配置数据:

```
{  
  "key1": "value1",  
  "key2": null  
}
```

结果:

```
{  
  "key1": "value1"  
}
```

例子 3: 修改数据

原数据:

```
{  
  "key1": "value1",  
  "key2": "value2"  
}
```

配置数据:

```
{  
  "key1": "value1",  
  "key2": "value2_1"  
}
```

结果:

```
{  
  "key1": "value1",  
  "key2": "value2_1"  
}
```

例子 4: 删除所有配置数据

如果需要删除所有配置数据，则可以单击如下图所示中按钮进行删除。

图 10-33 删除配置参数



删除后的配置数据为空，如下结构：

```
{}
```

10.2.6 查看配置结果

当完成模块配置后，可以在已部署的应用模块中确定对应的配置数据是否下发至边缘。

登录网关节点，并打开自定义应用日志，进行观察：

图 10-34 影子数据

```
2023-04-14T12:41:06.379 | INFO | 767fc700 | edge_http_get(edge_http.c:144) | get url: [http://localhost:72/modules/user_test_app_config/sign
2023-04-14T12:41:06.483 | INFO | 767fc700 | edge_http_delete(edge_http.c:131) | http: 0x189b2c0 delete
2023-04-14T12:41:06.483 | INFO | 767fc700 | edge_daemon_http_conns_sign(edge_http_connections.c:475) | rsp.expires_at:16814766380
2023-04-14T12:41:06.482 | INFO | 767fc700 | edge_daemon_http_sign_runner(edge_http_connections.c:384) | expires_at:16814766380, cur time:1681476666, sleep_time:10800
2023-04-14T12:41:06.482 | INFO | 767fc700 | edge_base_schedule_job_runner(edge_base_schedule_job.c:70) | jobName:edge_daemon_http_token_sleep_10800s.
2023-04-14T12:54:09.568 | DEBUG | 744f9700 | edge_mqtt_message_received_callback(edge_mqtt.c:209) | mqtt_rcv_message_topic:modules/user_test_app_config/shadow/notify, len:392
2023-04-14T12:54:09.568 | INFO | 744f9700 | edge_mqtt_conns_recv_msg(edge_mqtt_connections.c:172) | recv msg, topic:*, payload:len:392
shadowCb:{"properties":{"default_values":{"app_name":"user_app"},"connect_info":{"endpoint":"127.0.0.1"},"channels":[{"device_id":"device_01",
{"device_id":"device_02",
"test_data":"nTeYaqO1H1QEC8xrdL5JrRcL0LN0mXTVLfrfxl1lIKo="}]}},
"properties_update_time":{"default_values":1681476843981,
"connect_info":1681476843981,
"channels":1681476843981}
}
```

数据详情:

```
{
  "properties": {
    "default_values": {
      "app_name": "user_app"
    },
    "connect_info": {
      "endpoint": "127.0.0.1"
    },
    "channels": [
      {
        "device_id": "device_01",
        "test_data": "4dvuazeD4QCRW1Y3-PzWQzvm9OJStv36ZctLcwPzDV8="
      },
      {
        "device_id": "device_02",
        "test_data": "nTeYaqO1H1QEC8xrdL5JrRcL0LN0mXTVLfrfxl1lIKo="
      }
    ]
  },
  "properties_update_time": {
    "default_values": 1681476843981,
    "connect_info": 1681476843981,
    "channels": 1681476843981
  }
}
```

其中test_data为加密数据，需要调用sdk接口进行解密。其余数据可直接使用。

11 基于 IoT 边缘实现云原生应用部署

11.1 实验简介

本实验将指导您搭建一个基于IoT边缘服务的边缘集群，并在边缘集群上部署用户自定义的云原生应用。

通过本实验，您将能够：

- 了解并掌握IoT边缘服务的功能特点与使用方法。
- 掌握通过IoTEdge云服务将自定义云原生应用部署到边缘集群的方法。

11.2 前提条件

11.2.1 前提概要

1. **开通服务：**体验需要三方配合，首先要开通IoTEdge，用于集群管理和应用管理。还需开通SWR，用于存储已开发并想部署到边缘集群上运行的云原生应用Docker镜像。
2. **准备边缘服务器：**IoTEdge、SWR是华为云服务，使用已开通华为云的账号直接免费开通可用。边缘服务器属于硬件，可以购买低配的ECS代替，其EIP则作为边缘服务器的节点IP地址。后续若创建单机版集群仅需要一台边缘服务器，若创建高可用集群则推荐准备五台在同一VPC下的边缘服务器，且需要申请一个vip（虚拟IP地址）。
3. **准备镜像和Chart包：**用户将应用镜像打包上传到SWR云服务，并制作好应用的Chart包，其中values.yaml中的镜像地址填写上传到SWR的应用镜像地址。

11.2.2 上传应用镜像到 SWR

步骤1 准备一个制作好的云原生应用镜像，登录边缘服务器，拉取镜像。此实验仅演示部署步骤，故直接拉取一个现有的镜像为例。

```
//1.拉取镜像
$ docker pull swr.cn-north-4.myhuaweicloud.com/app_test/redis:0.1.1
//2.查看拉取到的镜像
$ docker images
```

- 步骤2** 访问[容器镜像服务 SWR](#)，单击“立即使用”，进入容器镜像服务控制台。
- 步骤3** 单击右上角“创建组织”，创建容器镜像所属的组织，组织名称：自定义，如 `iot_edge_test`。
- 步骤4** 单击右上角“登录指令”，复制在边缘集群服务器上连接容器镜像服务的临时指令。

图 11-1 获取登录命令



- 步骤5** 登录边缘集群服务器后台系统，输入复制的登录指令并回车，提示“Login Succeeded”则表明该硬件已通过SWR云服务的鉴权认证，可进行上传/下载镜像。

- 步骤6** 上传Docker镜像到SWR云服务。

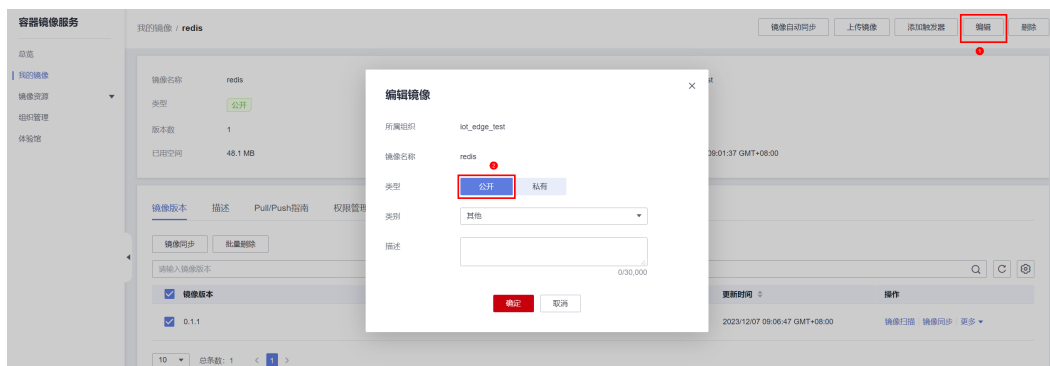
- 为本地镜像打标签，格式为 `docker tag [镜像名称:版本名称] [镜像仓库地址]/[组织名称]/[镜像名称:版本名称]`。该本地镜像为**步骤1**中拉取到ECS的redis镜像。

```
$ docker tag swr.cn-north-4.myhuaweicloud.com/app_test/redis:0.1.1 swr.cn-north-4.myhuaweicloud.com/iot_edge_test/redis:0.1.1
```
 - 上传镜像到SWR，格式为 `docker push [镜像仓库地址]/[组织名称]/[镜像名称:版本名称]`。

```
$ docker push swr.cn-north-4.myhuaweicloud.com/iot_edge_test/redis:0.1.1
```
- 镜像仓库地址：本实验在北京四环境，则该地址为：`swr.cn-north-4.myhuaweicloud.com`。
 - 组织名称：上述**步骤3**在SWR云服务上创建的组织名称，即 `iot_edge_test`。
 - 镜像名称/版本：自定义，该名称即为上传到SWR存储时镜像的名称/版本，即 `redis:0.1.1`。

- 步骤7** 在SWR控制台选择“我的镜像”，单击上传的自定义镜像名称进入详情，单击右上方的“编辑”，将该镜像类型设置成“公开”。

图 11-2 设置镜像类型为“公开”



注意

本实验需将上传到SWR中的应用镜像类型设置为“公开”，若镜像类型为“私有”，则需要应用Chart包中values.yaml中的imagePullSecrets字段填写密钥内容，不在本实验的操作范围内。

----结束

11.3 操作步骤

11.3.1 创建边缘集群

步骤1 访问[IoT边缘](#)，单击“**管理控制台**”进入IoT边缘控制台。

步骤2 选择左侧导航栏“**边缘集群 > 集群管理**”进入页面，单击右上角“**创建边缘集群**”。

图 11-3 创建边缘集群

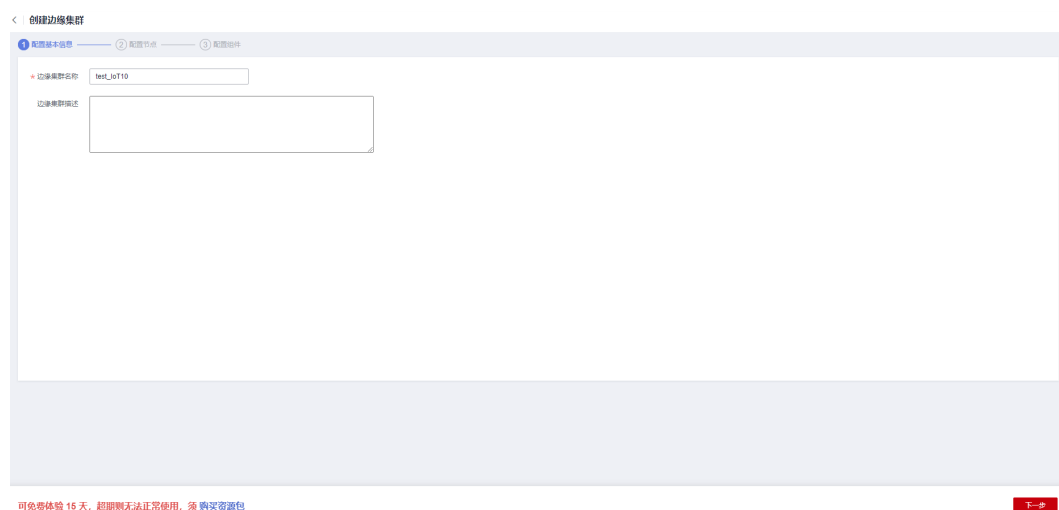


步骤3 据页面提示填写基本信息后，单击右下角“**下一步**”配置节点信息参数。

边缘集群名称：自定义边缘集群名称，如“testApp”。

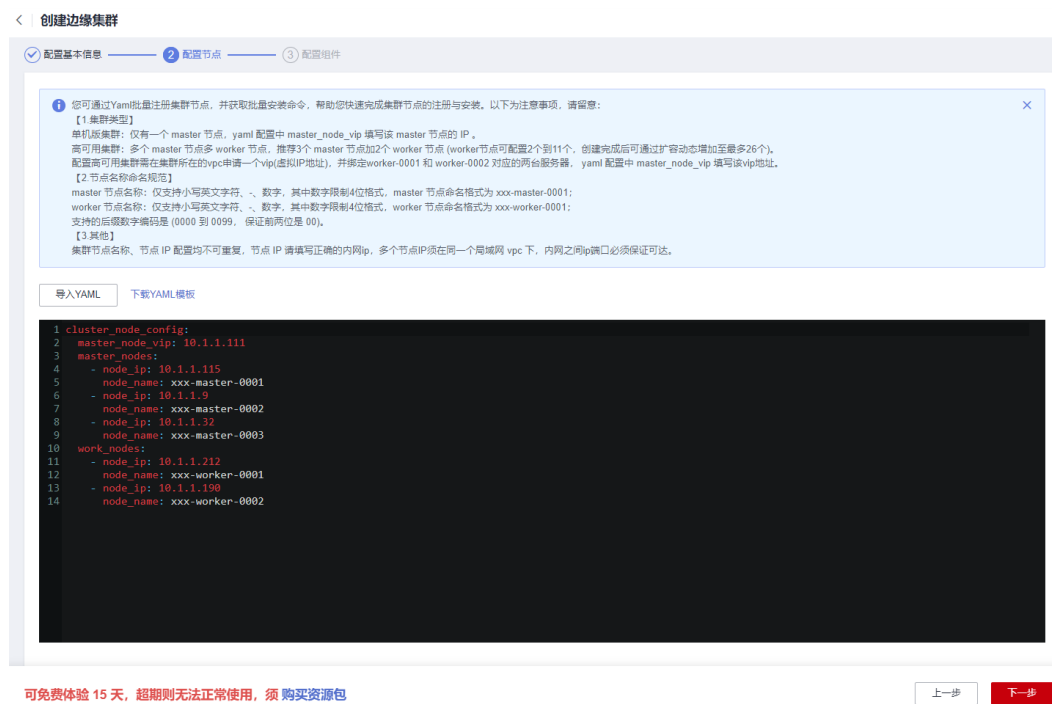
边缘集群描述：根据实际情况描述。

图 11-4 集群名称填写



步骤4 配置集群节点信息参数，可以通过console页面进行填写配置参数；也可以通过下载YAML模板填写，然后再导入YAML文件。如果创建高可用集群，执行步骤5；如果创建单节点集群，执行步骤6。

图 11-5 集群 yaml 配置



⚠ 注意

您可通过Yaml批量注册集群节点，并获取批量安装命令，帮助您快速完成集群节点的注册与安装。

● 集群类型

单机版集群：仅有一个master节点，Yaml 配置中master_node_vip填写该master节点的IP。

高可用集群：多个master节点多worker节点，推荐3个master节点加2个worker节点（worker节点可配置2个到11个，创建完成后可通过扩容动态增加至最多26个）。配置高可用集群需在集群所在的VPC申请一个vip（虚拟IP地址），并绑定“worker-0001”和“worker-0002”对应的两台服务器，Yaml 配置中master_node_vip填写该vip地址。

● 节点名称命名规范

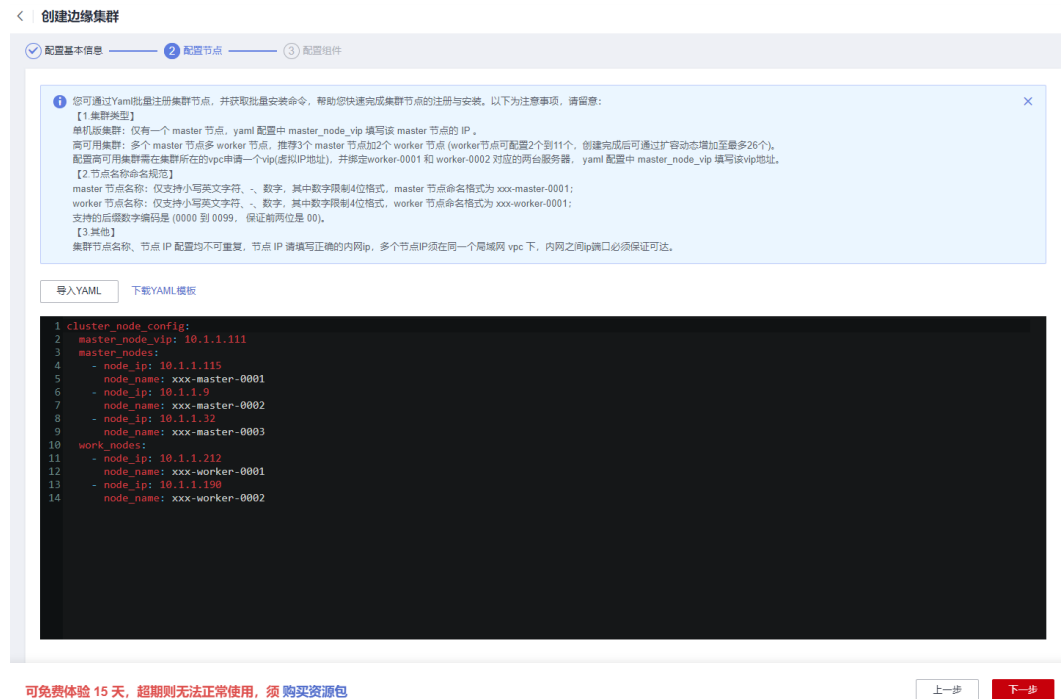
master节点名称：仅支持小写英文字符、-、数字，其中数字限制4位格式，master节点命名格式为 xxx-master-0001；worker节点名称：仅支持小写英文字符、-、数字，其中数字限制4位格式，worker节点命名格式为：xxx-worker-0001；支持的后缀数字编码是（0000到0099，保证前两位是00）。

● 其他

集群节点名称、节点IP配置均不可重复，节点IP请填写正确的内网IP，多个节点IP须在同一个局域网VPC下，内网之间IP端口必须保证可达。

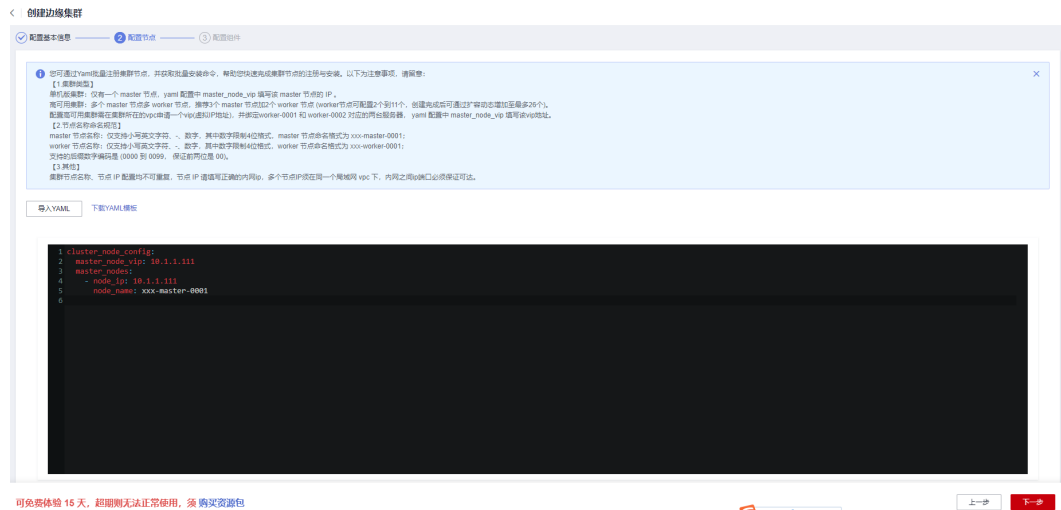
步骤5 根据集群的节点和IP规划，额外申请一个虚拟ip，填入master_node_vip，并配置其他参数后，单击“下一步”，执行步骤7。

图 11-6 高可用集群配置填写



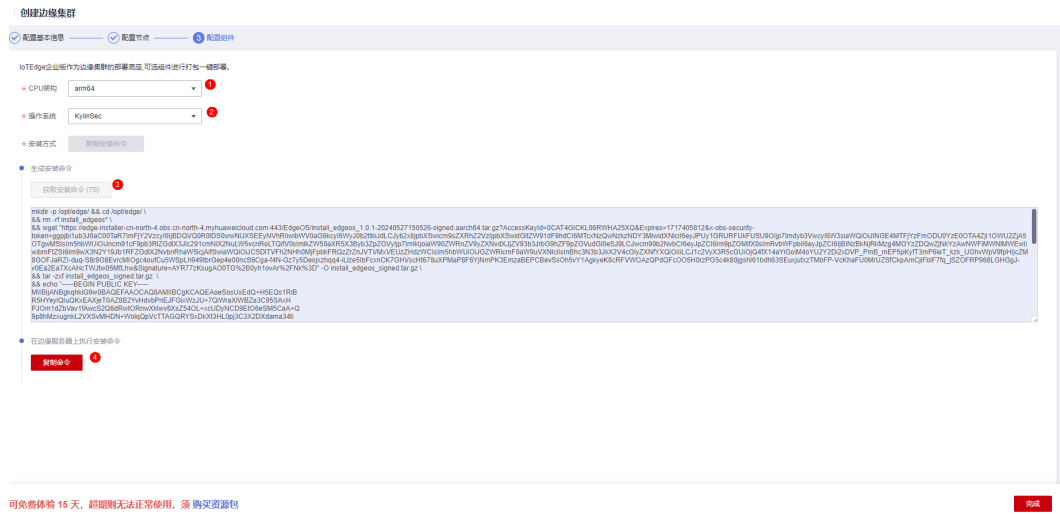
步骤6 将master_node_vip的IP与master节点IP保持一致，单击“下一步”，执行步骤7。

图 11-7 单机版集群配置填写



步骤7 选择对应的CPU架构和操作系统，单击“获取安装命令”，安装命令生成后，单击“复制命令”，到xxx-master-0001机器上执行安装命令，进行集群安装，这个步骤大概10~15min。安装成功后，单击“完成”即完成集群的安装。

图 11-8 生成对应操作系统架构的安装命令



步骤8 可登录边缘集群xxx-master-0001机器通过命令查看集群状态，若各Pod状态都是“Running”则说明集群正常运行。

```
$ kubectl get pods -A
```

----结束

11.3.2 创建应用模板

步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。

步骤2 在左侧导航栏选择“边缘集群 > 应用模板”，单击右上角“创建应用模板”。

图 11-9 创建应用模板



步骤3 在“添加应用模板”弹窗中填写信息后，单击“确认”即完成“创建模板”。

模板名称：自定义。

应用类型：云服务。

功能用途：默认“云原生应用”。

描述：可选填。

图 11-10 添加应用模板

✕

添加应用模板

★ 模板名称

★ 应用类型

功能用途

描述

步骤4 在应用模板也可以查看到已创建的应用模板。

图 11-11 查看已创建的应用模板



---结束

11.3.3 添加应用版本

步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。

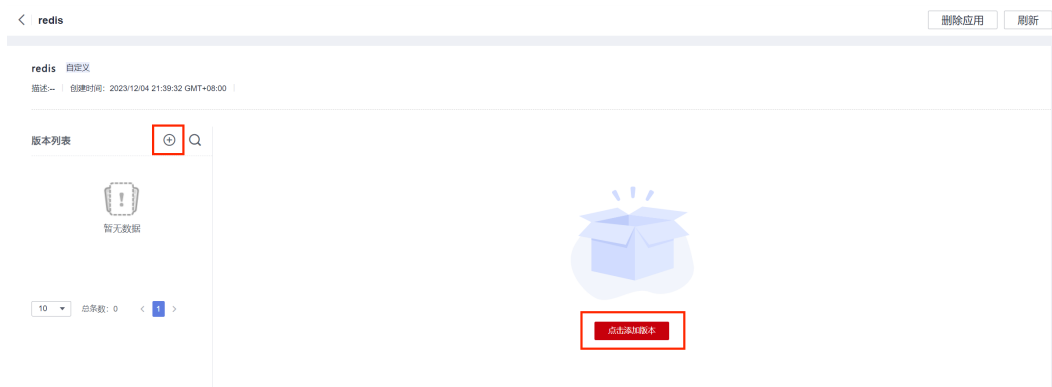
步骤2 在左侧导航栏选择“边缘集群 > 应用模板”，在对应的“应用模板名称”右侧单击“版本管理”进入页面。

图 11-12 版本管理



步骤3 单击左侧版本列表的“+”或者单击“添加版本”为该应用模板添加新的版本。

图 11-13 添加版本



步骤4 在“添加应用版本”弹窗填写版本号，上传制作好的云原生应用的Chart包文件，其中 values.yaml中的应用镜像地址填写上传镜像到SWR的应用镜像地址，且版本保持一致，本实验填写如图3所示，单击“确认”完成应用版本号添加。

说明

添加应用版本时，上传的Chart包需要满足如下条件：

1. Chart包为合法的tgz包；
2. Chart包名称和版本（Chart.yaml中的name、version字段），与应用模板的app_id和app_version必须一一对应；
3. Chart包的文件结构需要满足规范，必须包含Chart.yaml、values.yaml。

图 11-14 values.yaml 中的镜像内容填写示例

```
image:
  pullPolicy: Always
  repository: swr.cn-north-4.myhuaweicloud.com/iot_edge_test/redis
  tag: 0.1.1
```

图 11-15 添加应用版本



添加应用版本

仅支持 *.tgz 和 *.tar.gz 格式，文件大小不超过 4M

* 模板名称: redis

* 版本号: 0.1.1

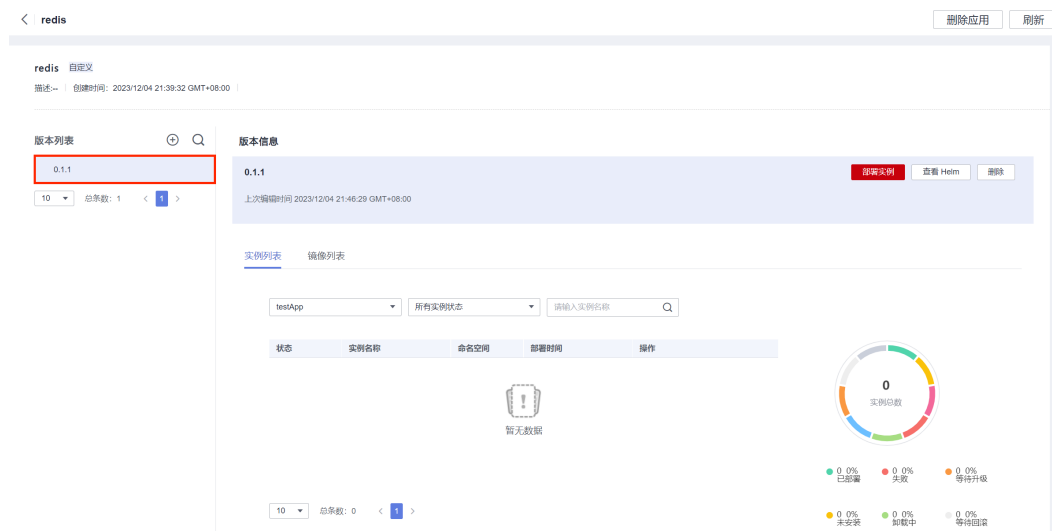
* 上传文件: redis-0.1.1.tgz (4.54KB) [X] [添加文件]

完整性校验 安装应用时，对应用包含的镜像进行完整性校验

[确认] [取消]

步骤5 添加新版本成功后，在左侧应用版本列表即可查看添加的应用版本。

图 11-16 查看添加的应用版本



----结束

11.3.4 部署云原生应用到边缘集群

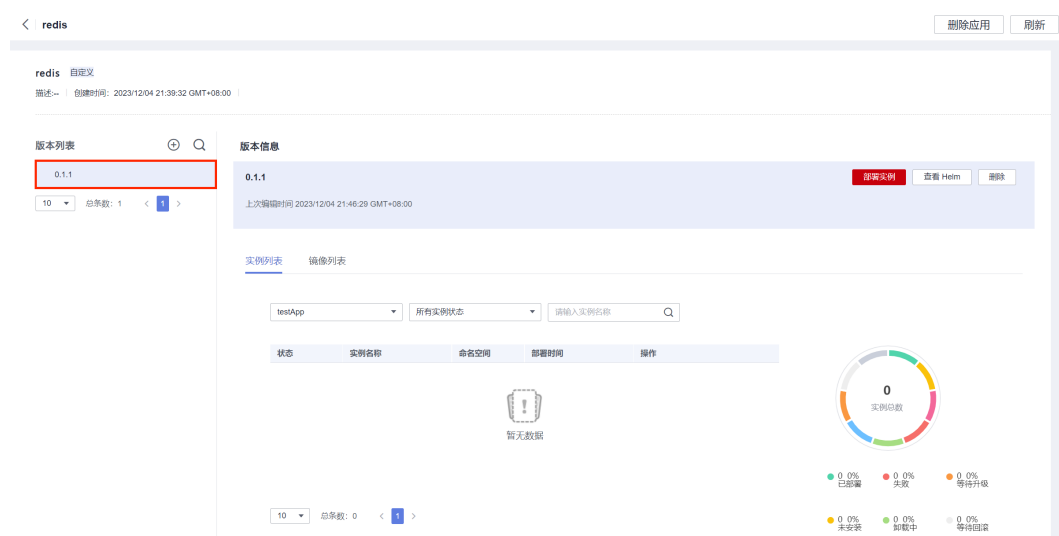
步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。

步骤2 在左侧导航栏“边缘集群 > 应用模板”，在对应的“应用模板名称”右侧单击“版本管理”可以查看该应用的所有版本。这里选择之前创建的应用模板“redis”为例。

图 11-17 版本管理



图 11-18 查看应用版本



步骤3 选择需要部署的版本，单击右上角“部署实例”。

图 11-19 部署实例



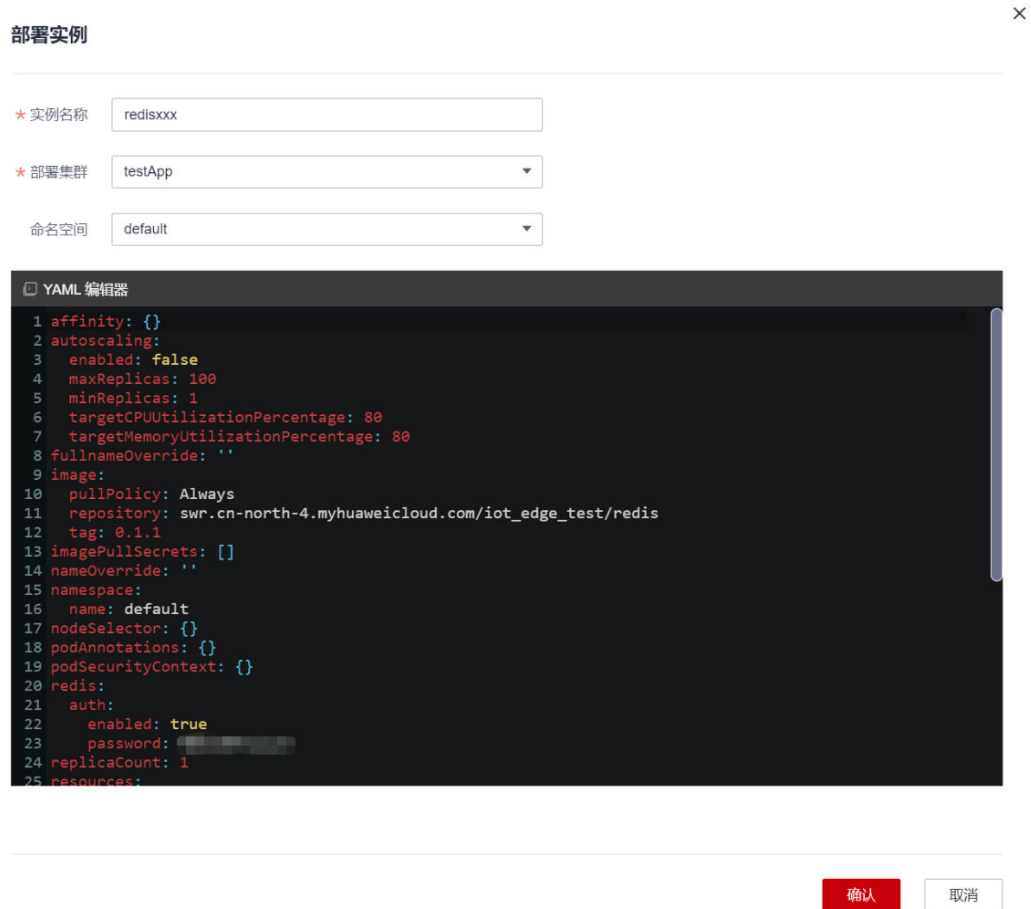
步骤4 在“部署实例”弹窗中填写信息后，单击“确认”即可完成该应用的部署。

实例名称：自定义。

部署集群：根据实际情况选择要部署实例的集群。

命名空间：根据实际情况选择命名空间。

图 11-20 部署实例



步骤5 在上一步选择部署的集群的“应用实例”中可看到应用实例状态显示“已部署”。

图 11-21 应用实例状态显示“已部署”



步骤6 可登录边缘集群xxx-master-0001机器通过执行命令，可查看到应用已成功运行。

```
$ kubectl get pods -A
```

图 11-22 查看应用已成功运行



----结束

11.3.5 资源释放

实验完成，您可以按如下步骤删除在实验过程中，产生在您的华为账号下的相关资源。

⚠ 注意

所有资源删除动作，在“华北-北京四”区域下进行。

删除自定义应用

步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。

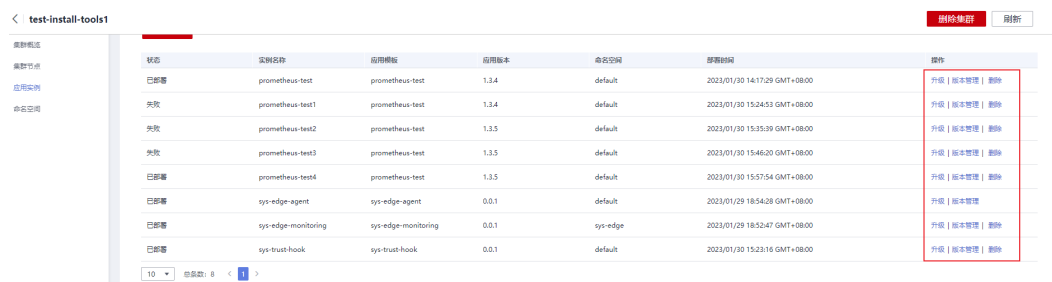
步骤2 在左侧导航栏单击“边缘集群 > 集群管理”，选择要删除应用的“边缘集群名称”，进入集群概览页。

图 11-23 集群管理



步骤3 选择“应用实例”，在需要删除的应用右侧，单击“删除”。

图 11-24 删除应用实例



📖 说明

系统必选的应用实例不支持删除，支持删除系统非必选或者用户自定义应用实例。

----结束

删除边缘集群

步骤1 访问IoT边缘，单击“管理控制台”进入IoT边缘控制台。

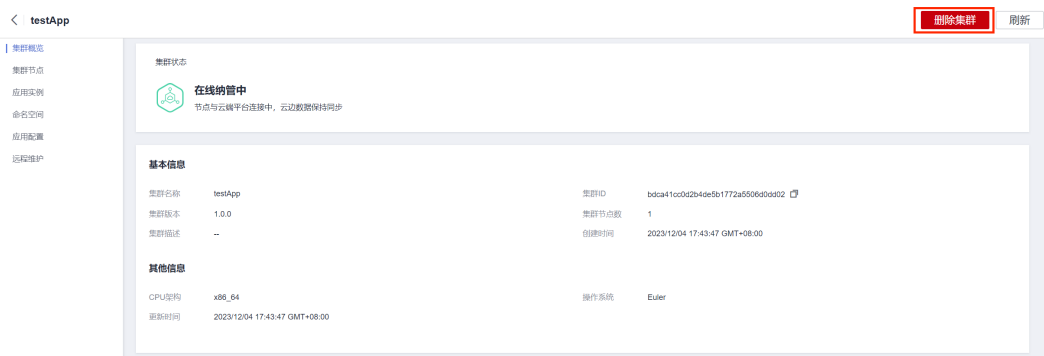
步骤2 在左侧导航栏选择“边缘集群 > 集群管理”，选择需要删除的“边缘集群名称”，单击右侧“删除”再单击“确认”即可删除集群。如果该集群已经安装，执行步骤3。

图 11-25 删除边缘集群



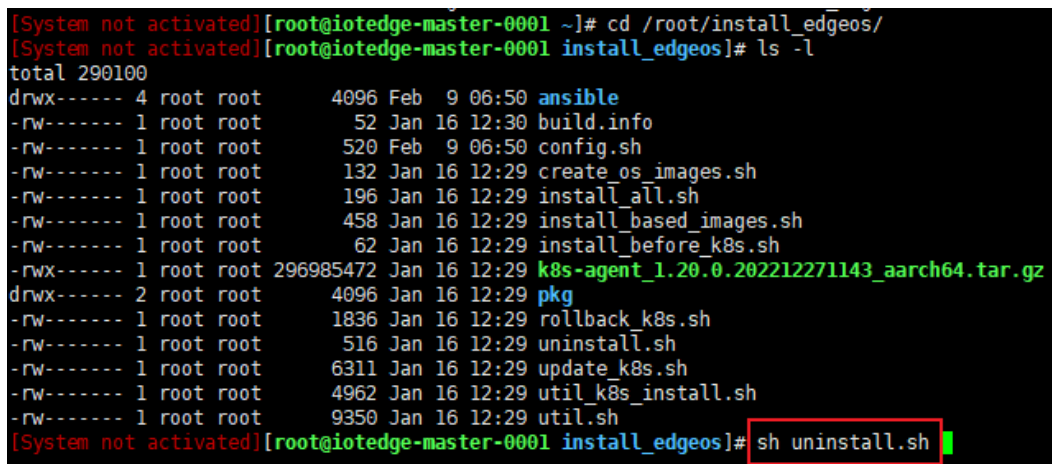
也可以单击“边缘集群名称”进入集群概览页，单击右上角的“删除集群”。

图 11-26 删除边缘集群



步骤3 在要删除集群的xxx-master-0001机器上，执行删除集群的脚本uninstall.sh，删除该集群。

图 11-27 执行删除集群的脚本



----结束

删除弹性云服务器 ECS

步骤1 访问弹性云服务器 ECS，单击“管理控制台”，进入云服务器控制台。

步骤2 在页面左上角切换区域至“华北-北京四”。

步骤3 在左侧导航中选择“弹性云服务器”，按需勾选待删除ECS服务器，如ecs-iot，选择“更多 > 删除”。

请勾选“释放云服务器绑定的弹性公网IP地址”和“删除云服务器挂载的数据盘”。

图 11-28 删除云服务器



----结束

12 基于 IoT 边缘企业版实现集群本地运维管理

12.1 实验简介

本实验将指导您在一个已经搭建好的（且满足硬件资源规格的）边缘集群（IoTEdge企业版）上部署IoTEdge内置的系统应用，用于实现边缘集群本地自治，包括本地监控运维、本地镜像仓库、和物联接入等功能。

通过本实验，您将能够：

- 了解并掌握如何基于IoT边缘集群如何部署系统应用。
- 掌握通过IoTEdge云服务如何部署边缘本地控制台。

本地控制台应用及依赖说明

在IoT边缘企业版中安装本地控制台应用，用于实现集群本地运维管理，涉及应用包括：

- `sys-console-gateway`：本地控制台网关。所有其他本地控制台应用的基础，用于集成其他本地控制台应用，包括统一入口，认证鉴权，`session`管理，前后端转发。所有其他本地控制台应用都依赖此应用，所以须优先安装此应用。
- `sys-lwops`：监控运维。
- `sys-edge-harbor`：本地镜像仓库。
- `sys-iotedge`：物联接入。依赖 `sys-edge-harbor` 的安装。

12.2 前提条件

开通服务：

体验需要三方配合，首先要开通IoTEdge，用于集群管理和应用管理。还需开通SWR，用于存储已开发并想部署到边缘集群上运行的云原生应用Docker镜像。

准备边缘服务器硬件资源：

边缘本地控制台全套系统应用的安装需要大量硬件资源，可以通过购买ECS替代实现，具体可参考如下 硬件资源规划。

在IoT边缘企业版中安装本地控制台应用，需要企业版Kubernetes集群中的工作节点满足应用运行所需要的硬件资源，包括内存、cpu、磁盘。以下是一套建议的配置。

表 12-1 资源规划

节点名称	安装本地控制台应用	cpu/内存	系统盘大小	挂载 /data/ 盘
iotedge-master-0001	无	2U/8G	300G	无
iotedge-master-0002	无	2U/8G	100G	无
iotedge-master-0003	无	2U/8G	100G	无
iotedge-worker-0001	sys-console-gateway sys-lwops	8U/32G	300G	1T
iotedge-worker-0002	sys-edge-harbor (主)	8U/32G	300G	1T
iotedge-worker-0003	sys-edge-harbor (备)	8U/32G	300G	1T
iotedge-worker-0004	sys-iotedge	8U/32G	300G	1T
iotedge-worker-0005	sys-iotedge	8U/32G	300G	1T
iotedge-worker-0006	sys-iotedge	8u/32G	300G	1T

须知

硬件资源不能低于建议的配置，否则应用可能因为硬件资源不够导致部署失败。若只是实验环境可以不用挂载 /data/ 数据盘。生产环境建议严格按照表格所示规划好硬件资源。

部署好边缘集群：

在准备好各台ECS服务器，且硬件资源满足上面条件后，可以部署边缘集群了。流程步骤参考[创建边缘集群](#)。

安装成功后检查如下：

- 单击“集群名称 > 集群概览”，查看集群节点信息，确认集群状态显示为“在线纳管中”。

The screenshot displays the 'Cluster Overview' page in the IoT Edge console. The left sidebar contains navigation options: 'Cluster Overview', 'Cluster Nodes', 'Application Instances', 'Naming Space', 'Application Configuration', and 'Remote Maintenance'. The main content area is divided into three sections:

- Cluster Status:** Shows 'Online Management' (在线纳管中) with a green icon and the text 'Nodes are connected to the cloud platform, and cloud data is kept synchronized.' (节点与云端平台连接中，云边数据保持同步)
- Basic Information:** Lists 'Cluster Name' (test-ha), 'Cluster Version' (1.0.0), and 'Cluster Description' (Local control console test, do not delete).
- Other Information:** Lists 'CPU Architecture' (arm64) and 'Update Time' (2024/04/02 09:04:25 GMT+08:00).

- 单击“集群名称 > 集群节点”查看集群节点信息，确认集群节点列表状态显示为“Ready”。

The screenshot shows the 'Cluster Nodes' page with a table of nodes. A red banner at the top indicates 'Expansion Work Item' (扩容工作项). The table columns are: Node Name, Node Type, IP Address, CPU Usage/Limit, Memory Usage/Limit (Total MB), Container Group Usage/Limit, Status, and Action.

节点名称	节点类型	IP地址	CPU 申请 / 限制	内存 申请 / 限制 / 总额(单位MB)	容器组 申请 / 限制	状态	操作
edges-master...	master	10.15.7.14	62.50% / 70.00...	14.9% / 25.33% / 6710.38	11 / 110	Ready	详情
edges-master...	master	10.15.7.53	52.50% / 35.00...	10.43% / 14.9% / 6710.38	9 / 110	Ready	详情
edges-master...	master	10.15.7.22	62.50% / 45.00...	14.25% / 18.72% / 6710.38	10 / 110	Ready	详情
edges-worker...	worker	10.15.7.86	66.25% / 8.75...	72.11% / 5.37% / 14893.25	7 / 110	Ready	详情
edges-worker...	worker	10.15.7.217	30.00% / 95.00...	14.16% / 43.19% / 6710.38	6 / 110	Ready	详情
edges-worker...	worker	10.15.8.124	60.31% / 117.5...	58.29% / 71.99% / 31261.25	11 / 110	Ready	详情
edges-worker...	worker	10.15.8.56	50.31% / 88.75...	55.47% / 56.43% / 31261.25	10 / 110	Ready	详情
edges-worker...	worker	10.15.8.5	41.25% / 90.00...	44.18% / 45.14% / 31261.25	10 / 110	Ready	详情
edges-worker...	worker	10.15.7.18	7.63% / 15.00...	2.84% / 4.66% / 31261.25	7 / 110	Ready	详情

12.3 操作步骤

12.3.1 部署本地控制台网关

sys-console-gateway等本地控制台应用，属于系统应用，已在“边缘集群>应用模板”中预置了系统应用模板，模板名称均以“sys-”开头。

操作步骤

- 步骤1** 左侧导航栏中选择“边缘集群 > 应用模板”，可以查看应用模板列表，可以查看到来源为“系统内置”的所有应用，其中包括了sys-console-gateway。

应用模板名称	应用类型	来源	功能用途
sys-iotedge	云服务	系统内置	云原生应用
sys-edge-harbor	云服务	系统内置	云原生应用
sys-lwops	云服务	系统内置	云原生应用
sys-console-gateway	云服务	系统内置	云原生应用
sys-ops-agent	云服务	系统内置	云原生应用
sys-edge-monitoring	云服务	系统内置	云原生应用

步骤2 左侧导航栏中选择“边缘集群>集群管理”，单击集群名称跳转到集群详情后，左侧导航栏中选择“应用实例>部署实例”。填写如下对应的实例名称、命名空间（选择 sys-edge）、应用模板，单击“部署”。关注部署状态，如果显示已部署，则部署完毕。

状态	实例名称	应用模板	应用版本
已部署	sys-console-gateway	sys-console-gateway	0.0.3
已部署	sys-edge-agent	sys-edge-agent	0.0.1
已部署	sys-edge-harbor	sys-edge-harbor	0.0.2
已部署	sys-edge-monitoring	sys-edge-monitoring	0.0.3
已部署	sys-iotedge	sys-iotedge	0.1.2
已部署	sys-lwops	sys-lwops	0.0.1

部署实例

* 部署集群: test-ha

* 实例名称: sys-console-gateway

* 命名空间: sys-edge

* 应用模板: sys-consol... 0.0.3

部署参数

```
1 global:
2   image:
3     tag: '20240311152544'
4   repository: registry-cbu.huawei.com
5   repositoryLocation: cloud
6   imagePullSecrets: []
7 namespace:
8   name: sys-edge
9 pullPolicy: IfNotPresent
10 replicaCount: 1
11 roles: primary
12 serviceConfig:
13   nodePort: 30270
```

----结束

12.3.2 部署监控告警组件

sys-lwops 是监控告警组件，也是系统应用。在部署完本地控制台网关后即可部署 sys-lwops。

步骤1 选中“边缘集群 > 应用模板”，可以查看应用模板列表，可以查看到来源为“系统内置”的所有应用，其中包括了 sys-lwops。

应用模板名称	应用类型	来源	功能用途
sys-iotedge	云服务	系统内置	云原生应用
sys-edge-harbor	云服务	系统内置	云原生应用
sys-lwops	云服务	系统内置	云原生应用
sys-console-gateway	云服务	系统内置	云原生应用
sys-ops-agent	云服务	系统内置	云原生应用
sys-edge-monitoring	云服务	系统内置	云原生应用

步骤2 左侧导航栏中选择“边缘集群 > 集群管理”，单击集群名称跳转到集群详情后，左侧导航栏中选择“应用实例 > 部署实例”。填写如下对应的实例名称、命名空间（选择 sys-edge）、应用模板。部署lwops还需要修改“部署参数”，deploy.node1 指定lwops部署在哪个节点上面。根据硬件资源规划，这里填写iotedge-worker-0001节点名称，确保镜像组织名称 imageNamespace，镜像名称 name，镜像版本 version 配置正确（从SWR上面是否能拉取到该镜像）。修改完成后单击“部署”，sys-lwops应用实例就会自动创建并部署到您的集群上。在这个页面可以关注到部署状态的变化，如果显示已部署，则应用已经部署成功。

状态	实例名称	应用模板	应用版本
已部署	sys-console-gateway	sys-console-gateway	0.0.3
已部署	sys-edge-agent	sys-edge-agent	0.0.1
已部署	sys-edge-harbor	sys-edge-harbor	0.0.2
已部署	sys-edge-monitoring	sys-edge-monitoring	0.0.3
已部署	sys-iotedge	sys-iotedge	0.1.2
已部署	sys-lwops	sys-lwops	0.0.1

部署实例

* 部署集群: test-ha

* 实例名称: sys-lwops

* 命名空间: sys-edge

* 应用模板: sys-lwops 0.0.1

部署参数

```

YAML 编辑器
20 memory: 4Gi
21 serviceName: zabbix-server
22 deploy:
23   enable_opration: false
24   node1: 'iotedge-worker-0001'
25 global:
26   imageNamespace: cloud-lwops
27   repositoryLocation: cloud
28 images:
29   lwops:
30     name: itops
31     version: '20231115175214'
32   postgresql:
33     name: postgresql
                
```

部署 取消

步骤3 lwops部署完毕后，需要等待3~5分钟，完成整个程序的初始化和启动，可执行命令查看lwops的pod容器是否已经处于READY状态。

```
$ kubectl get pods -n sys-edge | grep lwops
```

```

[System not activated][root@iotedge-master-0001 ~]# kubectl get pods -n sys-edge | grep lwops
lwops-c7b47777-c48v2          1/1   Running   0          29h
lwops-postgres-69d8d7d9bc-6v2qj  1/1   Running   0          29h
                
```

----结束

12.3.3 部署本地镜像仓库

sys-edge-harobr 是边缘本地镜像仓库，也是系统应用。后面安装的物联接入应用依赖本地镜像仓库。

步骤1 选择“边缘集群 > 应用模板”，可以查看应用模板列表，可以查看到来源为“系统内置”的所有应用，其中包括了 sys-edge-harbor。

应用模板名称	应用类型	来源	功能用途
sys-iotedge	云服务	系统内置	云原生应用
sys-edge-harbor	云服务	系统内置	云原生应用
sys-lwops	云服务	系统内置	云原生应用
sys-console-gateway	云服务	系统内置	云原生应用
sys-ops-agent	云服务	系统内置	云原生应用
sys-edge-monitoring	云服务	系统内置	云原生应用

步骤2 单击左上方“部署实例”，填写如下对应的实例名称、命名空间（sys-edge）、应用模板。

- 部署sys-edge-harbor时需要修改部署参数deploy.node1和 deploy.node2，指定 sys-edge-harbor部署的集群节点。根据硬件资源规划表，这里填写 iotedge-worker-0002和iotedge-worker-0003 节点名称。

部署实例

状态	实例名称	应用模板	应用版本
已部署	sys-console-gateway	sys-console-gateway	0.0.3
已部署	sys-edge-agent	sys-edge-agent	0.0.1
已部署	sys-edge-harbor	sys-edge-harbor	0.0.2
已部署	sys-edge-monitoring	sys-edge-monitoring	0.0.3
已部署	sys-iotedge	sys-iotedge	0.1.2
已部署	sys-lwops	sys-lwops	0.0.1

部署实例配置：

- * 部署集群: test-ha
- * 实例名称: sys-edge-harbor
- * 命名空间: sys-edge
- * 应用模板: sys-edge-harbor 0.0.3

部署参数

```

YAML 编辑器
56 podAnnotations: {}
57 type: internal
58 deploy:
59   node1: 'iotedge-worker-0002'
60   node2: 'iotedge-worker-0003'
61   existingSecretSecretKey: ''
62 fullname: edge-harbor-cluster
63 harborAdminPassword: Harbor#sre2023
64 hostName: artifact.local.edgeos.com
65 imagePullPolicy: IfNotPresent
66 imagePullSecrets: null
67 images:
68   core:
    
```

- 部署regionID字段的值配置成对应部署区域的ID值，默认值是北京4的regionID（cn-north-5），若在其他区域部署，则修改为其它区域对应的regionID即可。regionSelectType字段已经列出了所有支持的区域ID，可参考其填写。

```

YAML 编辑器
64  hostName: artifact.local.edges.com
65  hwcCloudDeploy:
66    regionID: cn-north-5
67    regionSelectType: >-
68      cn-north-5(乌兰201)/cn-north-4(北京4)/cn-east-3(上海
        1)/cn-south-1(华南广州)/cn-south-4(广州友好)
69  imagePullPolicy: IfNotPresent
70  imagePullSecrets: null
71  images:
72    core:
73      name: huawei-iot-edge-app/harbor-core
74      tag: v2.6.0
75    db:
76      name: huawei-iot-edge-app/harbor-db
77      tag: v2.6.0
78    jobservice:
79      name: huawei-iot-edge-app/harbor-jobservice
80      tag: v2.6.0
81    postjob:
82      name: huawei-iot-edge-app/edge_harbor_tools
83      tag: '20231106103644'
84    proxy:
85      name: huawei-iot-edge-app/edge-harbor-proxy
86      tag: 2023-11-21-1656-32
    
```

3. 所有镜像组织名由默认的“system”改为“huawei-iot-edge-app”。

The screenshot shows a deployment management interface. On the left is a table of deployment instances:

状态	实例名称	应用模板	应用版本
已部署	sys-console-gateway	sys-console-gateway	0.0.3
已部署	sys-edge-agent	sys-edge-agent	0.0.1
已部署	sys-edge-harbor	sys-edge-harbor	0.0.2
已部署	sys-edge-monitoring	sys-edge-monitoring	0.0.3
已部署	sys-iotedge	sys-iotedge	0.1.2
已部署	sys-ivops	sys-ivops	0.0.1

On the right, the configuration for the 'sys-edge-harbor' instance is shown. The '部署实例' (Deployment Instance) section includes:

- 部署集群: lesl-ha
- 实例名称: sys-edge-harbor
- 命名空间: sys-edge
- 应用模板: sys-edge-h...
- 应用版本: 0.0.3

The '部署参数' (Deployment Parameters) section shows a YAML editor with the following content:

```

79  tag: '20231106103644'
80  proxy:
81    name: huawei-iot-edge-app/edge-harbor-proxy
82    tag: 2023.11.21.1656.32
83  redisphoton:
84    name: huawei-iot-edge-app/redis-photon
85    tag: v2.6.0
86  registryctl:
87    name: huawei-iot-edge-app/harbor-registryctl
88    tag: v2.6.0
89  registryphoton:
90    name: huawei-iot-edge-app/registry-photon
91    tag: v2.6.0
    
```

4. 修改后单击“部署”，因为sys-edge-harbor部署时间较长，不会立即返回部署成功，稍等后会提示“应用部署中，请刷新页面查看...”。

The screenshot shows a notification message with a red 'X' icon and the text: "应用正在部署中，请刷新页面查看实例部署详情。 IoTEdge.06103033". Below the notification, the text "部署实例" (Deployment Instance) is visible.

步骤3 sys-edge-harbor部署完毕后，需要等待3~5分钟，才能完成整个程序的初始化和启动，可执行命令查看sys-edge-harbor的pod容器是否已经处于 READY状态。

```
$ kubectl get pods -n sys-edge|grep edge-harbor
```

```
[System not activated][root@iotedge-master-0001 ~]# kubectl get pods -n sys-edge | grep edge-harbor
edge-harbor-1-fd8c847f6-5ngr8      6/6      Running   2          151m
edge-harbor-2-6fc95d4669-gtwh4    6/6      Running   2          5h58m
edge-harbor-proxy-85fb58fc57-4449x 1/1      Running   0          5h58m
sys-edge-harbor-harbor-init-8r4j5   0/1      Completed 0          5h58m
```

----结束

12.3.4 部署物联接入

sys-iotedge 物联接入也是系统应用。部署物联接入应用需要依赖前面的本地镜像仓库，因此确保您已经安装了本地镜像仓库。

步骤1 选中“边缘集群 > 应用模板”，可以查看应用模板列表，可以查看到来源为“系统内置”的所用应用，其中包括了sys-iotedge。



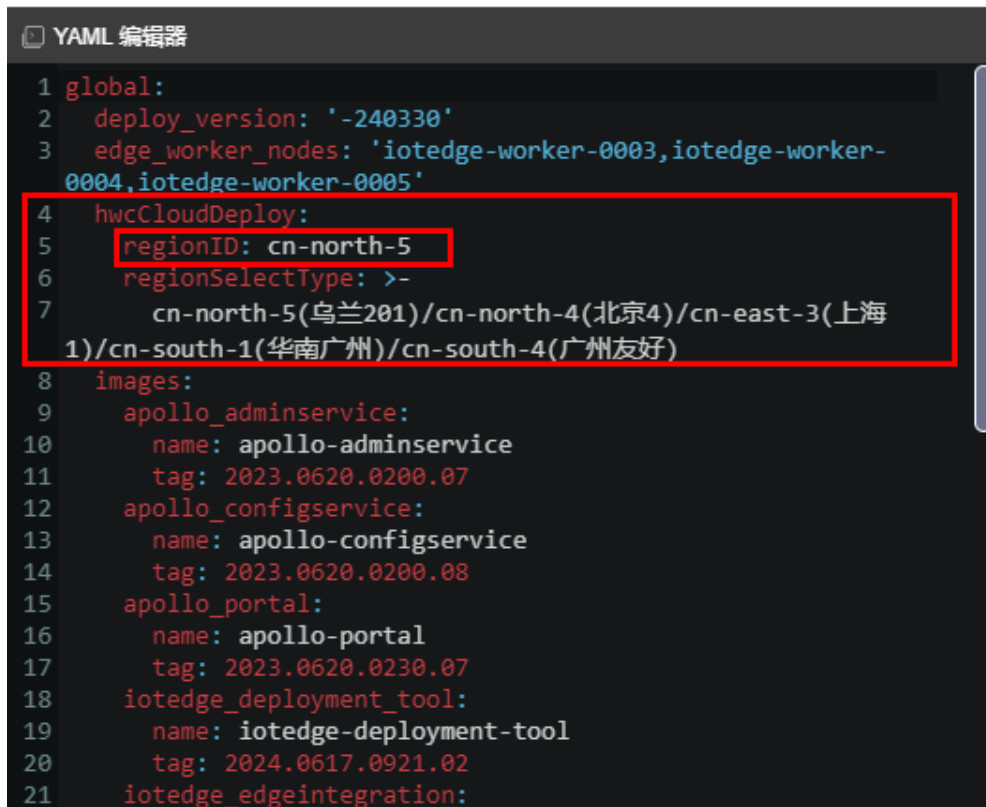
应用模板名称	应用类型	来源	功能用途
sys-iotedge	云服务	系统内置	云原生应用
sys-edge-harbor	云服务	系统内置	云原生应用
sys-lwops	云服务	系统内置	云原生应用
sys-console-gateway	云服务	系统内置	云原生应用
sys-ops-agent	云服务	系统内置	云原生应用
sys-edge-monitoring	云服务	系统内置	云原生应用

步骤2 左侧导航栏中选择“边缘集群 > 集群管理”，单击集群名称跳转到集群详情后，左侧导航栏中选择“应用实例 > 单击部署实例”，填写如下对应的实例名称、命名空间(sys-edge)、应用模板。

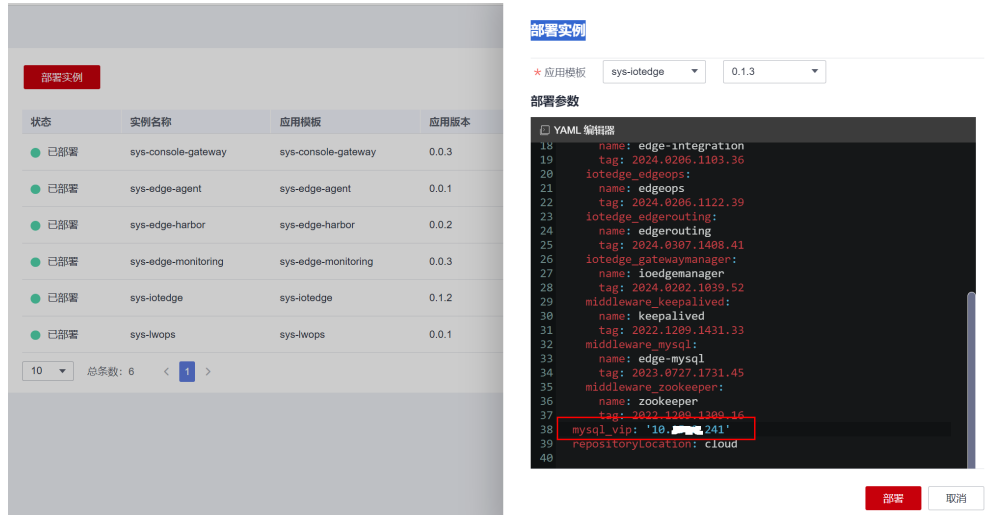
1. 部署sys-iotedge时需要修改部署参数 global.edge_worker_nodes: 指定sys-iotedge部署在三个集群节点上。根据硬件资源规划填写“iotedge-worker-0004,iotedge-worker-0005,iotedge-worker-0006”。



- 部署regionID字段的值配置成对应部署区域的ID值，默认值是北京4的regionID（cn-north-5），若在其他区域部署，则修改为其它区域对应的regionID即可。regionSelectType字段已经列出了所有区域ID，可参考其填写。



- global.mysql_vip 改为规划的虚拟IP地址，虚拟IP地址需要在华为云上申请并绑定到规划部署sys-iotedge的三个节点的前两个节点上。请参考[申请虚拟IP地址](#)申请虚拟IP并绑定指定服务器。



4. 修改后单击“部署”，因为sys-iotedge部署时间较长，不会立即返回部署成功，而是稍后会提示类似“部署中，稍后请刷新列表..”



部署实例

- 步骤3 sys-iotedge部署完毕后，需要等待10~15分钟，完成整个程序的初始化和启动，可以在边缘集群中使用命令行查看sys-iotedge的pod容器是否已经处于READY状态。

```
[System not activated][root@iotedge-master-0001 ~]# kubectl get pod -n iot
NAME                                READY   STATUS    RESTARTS   AGE
deployment-edge-integration-5d87f7fcb5-2kq95    1/1     Running   1           6h12m
deployment-edge-integration-5d87f7fcb5-ftbtd    1/1     Running   1           6h12m
deployment-edge-ops-5bf7b488f5-gbds5           1/1     Running   0           121m
deployment-edge-ops-5bf7b488f5-s554v           1/1     Running   0           118m
deployment-edge-routing-6fcb6595f4-m2fjp       1/1     Running   1           6h12m
deployment-edge-routing-6fcb6595f4-q4tdw       1/1     Running   1           6h12m
deployment-gateway-manager-dd86bf9d9-5z4vv     1/1     Running   0           6h12m
deployment-gateway-manager-dd86bf9d9-lwsjf     1/1     Running   0           6h12m
deployment-keepalived-mysql-5d7f94cd88-5sz7p   1/1     Running   0           6h12m
deployment-keepalived-mysql-5d7f94cd88-7rzkp   1/1     Running   0           6h12m
iotedge-apollo-adminservice-7cb868c6d5-xpjgw   1/1     Running   1           6h12m
iotedge-apollo-adminservice-7cb868c6d5-zng4x   1/1     Running   1           6h12m
iotedge-apollo-configservice-79db6b4b85-jpk2p  1/1     Running   2           6h12m
iotedge-apollo-configservice-79db6b4b85-qlhbx  1/1     Running   2           6h12m
iotedge-apollo-portal-5bc768c794-nwkpz        1/1     Running   1           6h12m
iotedge-apollo-portal-5bc768c794-v2vx5        1/1     Running   1           6h12m
statefulset-mysql-0                           1/1     Running   0           6h12m
statefulset-mysql-1                           1/1     Running   0           6h11m
statefulset-zookeeper-0                       1/1     Running   0           6h12m
statefulset-zookeeper-1                       1/1     Running   0           6h5m
statefulset-zookeeper-2                       1/1     Running   0           6h4m
```

----结束

12.3.5 登录边缘本地控制台

本地浏览器输入https://{公网ip}:30070/，使用 Admin账号登录后检查本地控制台各个功能是否OK。

📖 说明

- 公网ip可以是master01的公网IP地址，也可以是任意节点的公网IP，或者是vip绑定的公网IP。
- 首次登录需修改密码。

