阿里云 物联网边缘计算

用户指南

文档版本: 20190322

为了无法计算的价值 | [] 阿里云

<u>法律声明</u>

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读 或使用本文档,您的阅读或使用行为将被视为对本声明全部内容的认可。

- 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档,且仅能用于自身的合法 合规的业务活动。本文档的内容视为阿里云的保密信息,您应当严格遵守保密义务;未经阿里云 事先书面同意,您不得向任何第三方披露本手册内容或提供给任何第三方使用。
- 未经阿里云事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分 或全部,不得以任何方式或途径进行传播和宣传。
- 3. 由于产品版本升级、调整或其他原因,本文档内容有可能变更。阿里云保留在没有任何通知或者 提示下对本文档的内容进行修改的权利,并在阿里云授权通道中不时发布更新后的用户文档。您 应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
- 4. 本文档仅作为用户使用阿里云产品及服务的参考性指引,阿里云以产品及服务的"现状"、"有缺陷"和"当前功能"的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,阿里云不承担任何法律责任。在任何情况下,阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使阿里云已被告知该等损失的可能性)。
- 5. 阿里云网站上所有内容,包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计,均由阿里云和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外,未经阿里云事先书面同意,任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独为或以组合形式包含"阿里云"、Aliyun"、"万网"等阿里云和/或其关联公司品牌,上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
- 6. 如若发现本文档存在任何错误,请与阿里云取得直接联系。

通用约定

格式	说明	样例
•	该类警示信息将导致系统重大变更甚至 故障,或者导致人身伤害等结果。	禁止: 重置操作将丢失用户配置数据。
A	该类警示信息可能导致系统重大变更甚 至故障,或者导致人身伤害等结果。	▲ 警告: 重启操作将导致业务中断,恢复业务所需 时间约10分钟。
	用于补充说明、最佳实践、窍门等,不 是用户必须了解的内容。	道 说明: 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定。
courier 字体	命令。	执行 cd /d C:/windows 命令,进 入Windows系统文件夹。
##	表示参数、变量。	bae log listinstanceid Instance_ID
[]或者[a b]	表示可选项,至多选择一个。	ipconfig[-all -t]
{}或者{a b }	表示必选项,至多选择一个。	<pre>swich {stand slave}</pre>

目录

法律声明I
通用约定I
1 环境搭建1
1.1 专业版环境搭建1
1.1.1 基于Ubuntu 16.04搭建环境1
1.1.2 基于Windows搭建环境11
1.2 标准版环境搭建21
1.2.1 基于Ubuntu 16.04搭建环境21
1.2.2 基于树莓派搭建环境32
1.2.3 基于阿里云Cloud Shell搭建环境 43
1.3 轻量版环境搭建54
1.3.1 轻量版环境搭建54
2 设备接入64
2.1 设备接入简介64
2.2 示例驱动66
2.3 官方驱动
2.4 驱动开发75
3 场景联动
3.1 什么是场景联动
3.2 云端场景联动97
3.3 边缘场景联动101
3.4 分配场景联动到边缘实例107
4 断网续传111
5 流数据分析117
5.1 什么是边缘流数据分析117
5.2 分配流数据分析到边缘实例118
5.3 边缘端流数据分析SQL120
5.3.1 概述120
5.3.2 创建源表121
5.3.3 创建结果表124
5.3.4 创建维表127
5.4 使用示例128
5.4.1 概述128
5.4.2 高温报警128
5.4.3 温度平均值计算133
5.4.4 连续高温报警144
5.4.5 订单结算148
5.4.6 人流重茂示159
6 云贠源迈问170

1环境搭建

1.1 专业版环境搭建

1.1.1 基于Ubuntu 16.04搭建环境

本文介绍如何在Ubuntu 16.04的系统中搭建专业版(LE Pro)的Docker运行环境,实现网关与 云端连接的步骤。

专业版(LE Pro)规格的详细说明请参见产品规格。

准备工作

LE Pro版需要您提前安装好Docker环境,请参考Docker_{官方文档}安装使用您Ubuntu 16.04系统的Docker客户端。要求Docker版本大于v17.03。

创建边缘实例和网关

1. 物联网平台控制台,选择边缘计算 > 边缘实例。

- 2. 创建一个边缘实例。
 - a. 单击新增实例, 在弹出窗口中设置实例名称。
 - b. 在网关产品后单击新建网关产品,为实例创建网关。

物联网边缘计算中的网关,承载边缘计算能力,每个实例必须分配一个网关设备,并且该网 关设备同一时间只能被分配到一个边缘实例。

物联网平台	边缘实例 💿					
快速入门						
设备管理	实例列表				A	前新建实例
规则引擎	实例 ∨ 请输入实例名称		×			
数据分析	实例名称	新增实例		实例状态	创建时间	操作
边缘计算	10	* 实例名称:		 未部署 	2018-12-13 10:52:57	查看 删除
边缘实例		LinkloTEdge_Node				
驱动管理	10000000	* 网关产品:		• 部署成功	2018-12-20 23:59:45	查看 删除
开发服务	10000012-	请选择网关产品	> 新建网关产品	● 部署成功	2018-12-21 10:39:08	查看 删除
应用托管	test, propri	₩215次: +新增标签		● 部署失败	2018-09-21 20:26:51	查看 删除
视频服务	waters;			◎ 未部署	2018-11-27 16:34:50	查看 删除
产品文档	1,010		職定 取消	 未部署 	2018-11-16 17:29:21	查看 删除

c. 在新建产品页面中,设置网关产品参数,然后单击完成。

物联网边缘计算中的新建网关产品功能继承物联网平台设备管理 > 产品中,高级版产品的功能。已自动为您简化创建适合物联网边缘计算中使用的网关产品步骤。

新建产			×
产品	信息		
	* 产品名称 LinkloTEdge_Gateway		
	* 所属分类 💿		古能会い
			ATTACK
更多	信息 产品描述		
	请输入产品描述		
		0/100	
使用文	文档		取消 完成

参数说明如下:

参数	说明
产品名称	为网关产品设置名称,用于后续的查询及识别网关产品。支持中 文、英文字母、数字和下划线,长度限制4~30,一个中文汉字算2 位。

参数	说明
所属分类	选择品类,为该产品定义 <u>物模型</u> 。 可选择为:
	 ・自定义品类:需根据实际需要,定义产品功能。 ・任一既有功能模板。
	选择任一物联网平台预定义的品类,快速完成产品的功能定义。 选择产品模板后,您可以在该模板基础上,编辑、修改、新增功 能。
	若您需要的网关没有特殊功能定义,建议您选择自定义品类。

参数	说明
产品描述	可输入文字,用来描述产品信息。字数限制为100。可以为空。

产品创建成功后,页面自动跳转回新增实例页面,并且网关产品参数下自动分配了刚创建的 网关产品。

d. 在新增实例页面,单击网关设备后的新建网关设备为网关产品添加设备。

物联网边缘计算中的新建网关设备功能继承物联网平台设备管理 > 设备的功能。

新增实例	>
* 实例名称:	
LinkloTEdge_Node	
* 网关产品:	
LinkloTEdge_Gateway	→ 新建网关产品
* 网关设备:	
请选择网关设备	→ 新建网关设备
标签信息:	
+新增标签	
	确定 取消

e. 根据界面提示设置参数后,单击确认。

参数说明如下:

参数	描述
产品	系统已自动关联上一步创建的网关产品。

参数	描述
设备名称	为该网关设备命名。设备名称需保持产品内唯一。如不填写,系统 将自动生成。
	 说明: 设备名称支持大写字母[A-Z]、小写字母[a-z]、数字[0-9]和下划 线(_)。且不能以下划线开头或结尾。

- f. (可选)在新增实例页面,单击新增标签,可以设置实例标签。通过标签您可以更加有效地 归类及识别实例。您也可以不设置标签。
- 3. 实例参数设置完成后,单击确定,至此您已创建边缘实例和网关。
- 4. 在实例详情 > 实例信息页面, 网关页签下, 单击网关名称右侧的查看, 获取网关设备信息。

边缘实例 > 实例详											
LinkloTEdge_N	Node 未部署								查看部署日志	重置	部署
CPU 使用率: - 查看		内存使用率: - 查看		存储	諸使用率: - 查看			服务版本号: -			
实例信息	子设备 子设备通信通道	场景联动 函数	计算 流数据分	浙 消息路由	日志服务	设置					
实例信息											编辑
实例名称	LinkloTEdge_Node		部署状态	未部署			云监控状态				
CPU 使用率	- 查看		内存使用率	- 查看			存储使用率	- 查看			
实例进程	查看		服务版本号	-			创建时间	2019-03-12 15	:51:20		
修改时间	2019-03-12 15:51:20										
实例标签											
网关										分配	
网关名称		产品名称			状态	最近	后上线时间	操作			
gateway_01		LinkloTE	dge_Gateway		 未激活 	£ -		<u> 音</u> 看 移	SR 远程连接 远格		

5. 系统跳转到网关设备的设备详情页面,在设备详情页面获取网关设备的设备证

书(ProductKey、DeviceName、DeviceSecret),用于后续启动网关。

设备管理 → 设备详情							
gateway_01	活						
产品:LinkloTEdge_Gate	产品:LinkloTEdge_Gateway 查看 ProductKey: 复制 DeviceSecret:******* 显示						
设备信息 Topic3	设备信息 Topic列表 运行状态 事件管理 服务调用 日志服务 子设备管理 子设备通道管理						
设备信息							
产品名称	LinkloTEdge_Gateway	ProductKey	复制]	区域	-	
节点类型	网关	DeviceName	复制]	DeviceSecret	******** 显示	
当前状态	未激活	IP地址	-		固件版本	-	
添加时间	2019/01/11 17:06:58	激活时间			最后上线时间		
实时延迟 💿	्रध्नम्ब्रह्म 🦉						
标签信息	标签信息						
设备标签:无标签信息,	设备标签:无标签信息,立即添加						

启动Link IoT Edge

- 1. 登录您的Ubuntu系统的机器。
- 2. 执行如下命令下载启动网关的脚本。

```
wget http://link-iot-edge-packet.oss-cn-shanghai.aliyuncs.com/config
/link-iot-edge.sh
chmod +x link-iot-edge.sh
```

3. 执行如下命令, 启动网关。

```
./link-iot-edge.sh {version} {YourProductKey} {YourDeviceName} {
YourDeviceSecret}
```



请将{version}替换为最新的发布版本号,将{YourProductKey} {YourDeviceName} { YourDeviceSecret}替换为已获取的网关设备的设备证书信息。 ./link-iot-edge.sh v1.8.1 a1*****gs gateway 2Px**********************************

如果您第一次启动网关,则需要完成如下交互式配置,您可以直接按Enter键使用默认配置。

- ・确认启动版本
- · 确认函数计算的runtime类型, 默认为进程版
- ·确认是否启动流式计算,默认开启流式计算
- ·确认是否卸载之前已安装的版本,默认卸载

拉取Docker镜像完成并启动可能需要等待5~10分钟,启动完成后通过docker ps命令查看相关Docker容器是否已启动,若系统显示如下图所示信息,表示启动成功。

weixiaeweixia-virtualbox: \$ weixia@weixia-VirtualBox: \$./link-iot-edge.sh v1.8.1 alphR16zqNZ gateway_01 [sudo] password for weixia:				
% Total % Received % Xferd Average Speed Time Time Time Current				
Dload Upload Total Spent Left Speed				
100 359 100 359 0 0 3243 0 = := := := := := := := := := := := := :				
Please confirm the version of Linkedge (v1.8.1) you want to start (v/n, default: v):				
Supported Fucntion Compute runtimes are: 0 Process-Runtime-All 1 Docker-Runtime	e-All 2 Docker-Runtime	e-Pvthon3 3 Dock	er-Runtime-Node is8	(Each c
osts about 200MB disk space. Choose 0 to disable Function Compute)				
Which runtime do you want to install? (Please input the number and use '' to split.	default: 0):			
Do you want to install Stream Compute (need 1GB Disk Space) (v/n, default: v):				
Do you want to uinstall previously installed Linkedge (v/n, default; v):				
Stopping scedge				
284a238a56c2				
Stopping feedge daemon				
Stopping feedge runtimes				
Stopping linkedge				
Installing linkedge				
Installing registry, cn-hangzhou, alivuncs, com/iotedge/edge x86 alpine:v1.8.1				
image already exists				
Function Compute Process Runtime in this version				
Installing Stream Compute Image				
Installing registry on-hangzhou, alivuncs, com/iotedge/edge x86 sc:v1.8.1				
image already exists				
Starting linkedge				
Initializing linkedge				
caa77c77baafd681f9908d9f01dfa3d450ed4a4c76c39c779c71c195d7cd4b8d				
2cbc4eaa4f61261874a0a4d9228d370860a3d70365e23ea697afb98c687476dd				
Start linkedge success!				
weixia@weixia=VirtualBox: `\$				
weixia@weixia-VirtualBox:~\$ docker ps				
CONTAINER ID IMAGE	COMMAND	CREATED	STATUS	PORTS
NAMES				
dfff6d58aacf registry, cn-hangzhou, alivuncs, com/iotedge/edge_x86_sc;v1.8,1	"/linkedge/stream-co…"	12 minutes ago	Up 12 minutes	
edge-stream-compute				
2cbc4eaa4f61 registry.cn-hangzhou.aliyuncs.com/iotedge/edge x86 alpine:v1.8.1	"/linkedge/gateway/b…"	13 minutes ago	Up 13 minutes	
linkedge				
weixia@weixia-VirtualBox:~\$				

在物联网控制台,选择边缘计算 > 边缘实例,在已创建好的边缘实例右侧单击查看进入实例详 情页面,查看网关状态。

边缘实例》实例详	den la constanti de								
LinkloTEdge_N	lode 未部署							查看部署日志	重置部署
CPU 使用率: - 查看		内存使用率: - 查看		存储	使用率:- 查看		服务版本号:-		
实例信息	子设备 子设备通信通道	场景联动 函数)	计算 流数据分析	忻 消息路由	日志服务 设置				
实例信息									编辑
实例名称	LinkloTEdge_Node		部署状态	未部署		云监控状态			
CPU 使用率	- 查看		内存使用率	- 查看		存储使用率	- 查看		
实例进程	查看		服务版本号	-		创建时间	2019-03-12 15:5	51:20	
修改时间	2019-03-13 17:11:00								
实例标签									_
									2
网关									分配网关
网关名称		产品名称			状态	最后上线时间	操作		
gateway_01		LinkloTE	dge_Gateway		 在线 	2019-03-14 15:23	3:44 查看 移	除 远程连接 远程3	之件管理

- 5. 在实例详情页面中,查看CPU使用率、内存使用率、存储使用率以及实例进程需要授权访问阿 里云云监控(CloudMonitor)服务。
 - a. 请参考云资源访问内容,在RAM控制台,创建授信IoT物联网的服务角色,并为该角色添加名 为AliyunCloudMonitorFullAccess的访问云监控服务的权限。
 - b. 在实例信息页签下打开云监控状态,如下图所示。

边缘实例 > 实例详	情						
LinkloTEdge_N	Node 部署成功					直看部署日志	重置部署
CPU 使用率: 20.63 %	% 查看	内存使用率: 48.57	% 查看	存储使用率: 38 % 查看		服务版本号: v1.8.2	
实例信息	文例信息 子设备 子设备通信通道 场景联动 函数计算 流数据分析 消息路由 日志服务 设置						
实例信息							编辑
实例名称	LinkloTEdge_Node		部署状态	部署成功	云监控状态		
CPU 使用率	20.63 % 查看		内存使用率	48.57 % 查看	存储使用率	38 % 查看	
实例进程	查看		服务版本号	v1.8.2	创建时间	2019-03-12 15:51:20	
修改时间	2019-03-13 17:11:00						
实例标签							

c. 在实例进程后,单击查看,可查看实例各个进程的信息及CPU、内存使用率。

边缘实例 > 实例详	情							
LinkloTEdge_N	Node 部署成功							查看部署日志 重置 部署
CPU 使用率: 13.32 % 查看 实际研程						服务版本号: v1.8.2		
实例信息	子设备 子设备					Blief		
立例信息						M33V1		從相
大四百心		进程 ID	进程名	CPU	内存			3614
实例名称	LinkloTEdge_No	280	task-dispatcher	5.17 %	0.61 %		云监控状态	
CPU 使用率	13.32 % 查看	431	node	1.46 %	3.41 %		存储使用率	38 % 查看
实例进程	直看	446	fc-base	1.13 %	1.91 %		创建时间	2019-03-12 15:51:20
修改时间	2019-03-13 17:11	204	logger	0.73 %	0.2 %			
实例标签		261	message-router	0.93 %	0.27 %			
网关						关闭		

 您可以在实例详情页面,网关名称右侧的操作栏中单击远程连接或者远程文件管理,方便您远程 控制网关设备或对网关设备上的文件进行管理。详细说明请参见远程运维管理。

Link IoT Edge的其它操作

・ 重新配置 Link IoT Edge。

如果您希望对当前已安装的 Link IoT Edge版本配置进行修改可以使用如下命令:

```
./link-iot-edge.sh --reconfig {Version}
```

其中, {Version}替换为目标版本号, 例如目标版本号为v1.8.1, 则实际命令为./link-iotedge.sh --reconfig v1.8.1。 ・停止Link IoT Edge。

使用如下命令可以停止所有Link IoT Edge运行的容器,但是不会删除。

./link-iot-edge.sh --stop

重新启动Link IoT Edge。

在容器已存在且没有运行的状态下,执行如下命令可重新启动Link IoT Edge。

./link-iot-edge.sh --restart {Version}

其中, {Version}替换为目标版本号, 例如目标版本号为v1.8.1, 则实际命令为./link-iotedge.sh --restart v1.8.1。

・清理Link IoT Edge。

执行如下命令可停止当前运行的Link IoT Edge相关容器,并会删除所有已安装的相关镜像,删除相关数据卷以及启动配置文件。

./link-iot-edge.sh --clean

・提取Link IoT Edge的日志。

执行如下命令可打包Link IoT Edge的所有日志,并拷贝到当前目录。

./link-iot-edge.sh --packagelog

1.1.2 基于Windows搭建环境

本文介绍如何在Windows7和Windows10的系统中搭建专业版(LE Pro)的Docker运行环境,实现网关与云端连接的步骤。

专业版(LE Pro)规格的详细说明请参见产品规格。

准备工作

 LE Pro版需要您提前安装好Docker环境, Windows7系统请参见Install Docker Toolbox on Windows, Windows10系统请参见Install Docker for Windows。要求Docker版本大 于v17.03。

约束条件

- · 目前仅支持在Windows7和Windows10的系统中搭建专业版运行环境。
- · LE Pro版需要bash的运行环境,用于运行Link IoT Edge的脚本工具,请务必安装好git bash

0

创建边缘实例和网关

1. 物联网平台控制台,选择边缘计算 > 边缘实例。

- 2. 创建一个边缘实例。
 - a. 单击新增实例, 在弹出窗口中设置实例名称。
 - b. 在网关产品后单击新建网关产品,为实例创建网关。

物联网边缘计算中的网关,承载边缘计算能力,每个实例必须分配一个网关设备,并且该网 关设备同一时间只能被分配到一个边缘实例。

物联网平台	边缘实例 💿					
快速入门						
设备管理	实例列表				扇	新建实例
规则引擎	实例 ~ 请输入实例名称		×			
数据分析	实例名称	新增实例		实例状态	创建时间	操作
边缘计算	444	* 实例名称:		 未部署 	2018-12-13 10:52:57	查看 删除
驱动管理	10000710	* 网关产品:		● 部署成功	2018-12-20 23:59:45	查看 删除
开发服务	1000000121	请选择网关产品	> 新建网关产品	● 部署成功	2018-12-21 10:39:08	查看 删除
应用托管	test, propert	标签信息: +新增标签		● 部署失败	2018-09-21 20:26:51	查看 删除
视频服务	100000			 未部署 	2018-11-27 16:34:50	查看 删除
产品文档	1,010		确定 取消	◎ 未部署	2018-11-16 17:29:21	查看 删除

c. 在新建产品页面中,设置网关产品参数,然后单击完成。

物联网边缘计算中的新建网关产品功能继承物联网平台设备管理 > 产品中,高级版产品的功能。已自动为您简化创建适合物联网边缘计算中使用的网关产品步骤。

新建产品			×
产品信息			
* 产品名称			
LinkloTEdge_Gateway			
* 所属分类 💿			
自定义品类	\sim	功能定义	
更多信息			
产品描述			
请输入产品描述			
	0/100		
使用文档		取消	完成

参数说明如下:

参数	说明
产品名称	为网关产品设置名称,用于后续的查询及识别网关产品。支持中 文、英文字母、数字和下划线,长度限制4~30,一个中文汉字算2 位。

参数	说明
所属分类	选择品类,为该产品定义 <u>物模型</u> 。 可选择为:
	 ・自定义品类:需根据实际需要,定义产品功能。 ・任一既有功能模板。
	选择任一物联网平台预定义的品类,快速完成产品的功能定义。 选择产品模板后,您可以在该模板基础上,编辑、修改、新增功 能。
	若您需要的网关没有特殊功能定义,建议您选择自定义品类。

参数	说明
产品描述	可输入文字,用来描述产品信息。字数限制为100。可以为空。

产品创建成功后,页面自动跳转回新增实例页面,并且网关产品参数下自动分配了刚创建的 网关产品。

d. 在新增实例页面,单击网关设备后的新建网关设备为网关产品添加设备。

物联网边缘计算中的新建网关设备功能继承物联网平台设备管理 > 设备的功能。

新增实例	×
* 实例名称:	
LinkloTEdge_Node	
* 网关产品:	
LinkloTEdge_Gateway	→ 新建网关产品
* 网关设备:	
请选择网关设备	→ 新建网关设备
标签信息:	
+新增标签	
	确定 取消

e. 根据界面提示设置参数后,单击确认。

参数说明如下:

参数	描述
产品	系统已自动关联上一步创建的网关产品。

参数	描述
设备名称	为该网关设备命名。设备名称需保持产品内唯一。如不填写,系统 将自动生成。
	 说明: 设备名称支持大写字母[A-Z]、小写字母[a-z]、数字[0-9]和下划 线(_)。且不能以下划线开头或结尾。

- f. (可选)在新增实例页面,单击新增标签,可以设置实例标签。通过标签您可以更加有效地 归类及识别实例。您也可以不设置标签。
- 3. 实例参数设置完成后,单击确定,至此您已创建边缘实例和网关。
- 4. 在实例详情 > 实例信息页面, 网关页签下, 单击网关名称右侧的查看, 获取网关设备信息。

边缘实例 > 实例详	情										
LinkloTEdge_N	Node 未部署								查看部署日志	重置	部署
CPU 使用率: - 查看		内存使用率: - 查看		存储	諸使用率: - 查看			服务版本号: -			
实例信息	子设备 子设备通信通道	场景联动 函数	计算 流数据分	浙 消息路由	日志服务	设置					
实例信息											编辑
实例名称	LinkloTEdge_Node		部署状态	未部署			云监控状态				
CPU 使用率	- 查看		内存使用率	- 查看			存储使用率	- 查看			
实例进程	查看		服务版本号	-			创建时间	2019-03-12 15	:51:20		
修改时间	2019-03-12 15:51:20										
实例标签											
网关										分配	
网关名称		产品名称			状态	最近	后上线时间	操作			
gateway_01		LinkloTE	dge_Gateway		 未激活 	£ -		<u> 音</u> 看 移	SR 远程连接 远格		

5. 系统跳转到网关设备的设备详情页面,在设备详情页面获取网关设备的设备证

书(ProductKey、DeviceName、DeviceSecret),用于后续启动网关。

设备管理 > 设备详情						
gateway_01	活					
产品:LinkloTEdge_Ga	teway 查看	ProductKey :	复制		DeviceSecret : ********	显示
设备信息 Topic	列表 运行状态 事件管理 服务调	用日志服务子	设备管理 子设备通道	管理		
设备信息						
产品名称	LinkloTEdge_Gateway	ProductKey	复制]	区域	-
节点类型	网关	DeviceName	复制]	DeviceSecret	******** 显示
当前状态	未激活	IP地址	-		固件版本	-
添加时间	2019/01/11 17:06:58	激活时间			最后上线时间	
实时延迟 🔘	测试					
标签信息						
设备标签:无标签信息,	立即添加					

启动Link IoT Edge

- 1. 登录您的Windows7或Windows10系统的机器。
- 2. 下载启动网关的脚本到机器上。

link-iot-edge.sh

3. 执行如下命令, 启动网关。

./link-iot-edge.sh {version} {YourProductKey} {YourDeviceName} {
YourDeviceSecret}



请将{version}替换为最新的发布版本号,将{YourProductKey} {YourDeviceName} { YourDeviceSecret}替换为已获取的网关设备的设备证书信息。

```
./link-iot-edge.sh v1.8.1 a1*****gs gateway 2Px***********************************
```

如果您第一次启动网关,则需要完成如下交互式配置,您可以直接按Enter键使用默认配置。

- ・ 确认启动版本
- ·确认函数计算的runtime类型,默认为进程版
- ·确认是否启动流式计算,默认开启流式计算
- ·确认是否卸载之前已安装的版本,默认卸载

拉取Docker镜像完成并启动可能需要等待5~10分钟,启动完成后通过docker ps命令查看相关Docker容器是否已启动,若系统显示如下图所示信息,表示启动成功。



在物联网控制台,选择边缘计算 > 边缘实例,在已创建好的边缘实例右侧单击查看进入实例详 情页面,查看网关状态。

边缘实例》实例详	den la constanti de								
LinkloTEdge_N	lode 未部署							查看部署日志	重置部署
CPU 使用率: - 查看		内存使用率: - 查看		存储	使用率:- 查看		服务版本号: -		
实例信息	子设备 子设备通信通道	场景联动 函数)	计算 流数据分析	忻 消息路由	日志服务 设置				
实例信息									编辑
实例名称	LinkloTEdge_Node		部署状态	未部署		云监控状态			
CPU 使用率	- 查看		内存使用率	- 查看		存储使用率	- 查看		
实例进程	查看		服务版本号	-		创建时间	2019-03-12 15:5	51:20	
修改时间	2019-03-13 17:11:00								
实例标签									_
									2
网关									分配网关
网关名称		产品名称			状态	最后上线时间	操作		
gateway_01		LinkloTE	dge_Gateway		 在线 	2019-03-14 15:23	3:44 查看 移	除 远程连接 远程3	之件管理

- 5. 在实例详情页面中,查看CPU使用率、内存使用率、存储使用率以及实例进程需要授权访问阿 里云云监控(CloudMonitor)服务。
 - a. 请参考云资源访问内容,在RAM控制台,创建授信IoT物联网的服务角色,并为该角色添加名 为AliyunCloudMonitorFullAccess的访问云监控服务的权限。
 - b. 在实例信息页签下打开云监控状态,如下图所示。

边缘实例 > 实例词	前								
LinkloTEdge_	Node 部署成功							查看部署日志	重置部署
CPU 使用率: 20.63	% 查看	内存使用率: 48.57	% 查看	存住	诸使用率: 38 % 查	酒		服务版本号: v1.8.2	
实例信息	子设备 子设备通信通道	场景联动 函数	计算 流数据分	淅 消息路由	日志服务	设置			
实例信息									编辑
实例名称	LinkloTEdge_Node		部署状态	部署成功			云监控状态		
CPU 使用率	20.63 % 查看		内存使用率	48.57 % 直看			存储使用率	38 % 查看	
实例进程	查看		服务版本号	v1.8.2			创建时间	2019-03-12 15:51:20	
修改时间	2019-03-13 17:11:00								
实例标签									

c. 在实例进程后,单击查看,可查看实例各个进程的信息及CPU、内存使用率。

边缘实例 > 实例详	情								
LinkloTEdge_N	Node 部署成功							查看部署日志 重置 寬	瑨
CPU 使用率: 13.32 9	% 查看	2例进程				- ×		服务版本号: v1.8.2	
实例信息	子设备 子设备					RIPC			
والم الم الم						//big/1			
头例信息		进程 ID	进程名	CPU	内存			<u>編輯</u>	
实例名称	LinkloTEdge_No	280	task-dispatcher	5.17 %	0.61 %		云监控状态		
CPU 使用率	13.32 % 查看	431	node	1.46 %	3.41 %		存储使用率	38 % 直看	
实例进程	查看	446	fc-base	1.13 %	1.91 %		创建时间	2019-03-12 15:51:20	
修改时间	2019-03-13 17:11	204	logger	0.73 %	0.2 %				
实例标签		261	message-router	0.93 %	0.27 %				
网关						关闭			

 您可以在实例详情页面,网关名称右侧的操作栏中单击远程连接或者远程文件管理,方便您远程 控制网关设备或对网关设备上的文件进行管理。详细说明请参见远程运维管理。

Link IoT Edge的其它操作

・ 重新配置 Link IoT Edge。

如果您希望对当前已安装的 Link IoT Edge版本配置进行修改可以使用如下命令:

```
./link-iot-edge.sh --reconfig {Version}
```

其中, {Version}替换为目标版本号, 例如目标版本号为v1.8.1, 则实际命令为./link-iotedge.sh --reconfig v1.8.1。 ·停止Link IoT Edge。

使用如下命令可以停止所有Link IoT Edge运行的容器,但是不会删除。

./link-iot-edge.sh --stop

重新启动Link IoT Edge。

在容器已存在且没有运行的状态下,执行如下命令可重新启动Link IoT Edge。

```
./link-iot-edge.sh --restart {Version}
```

其中, {Version}替换为目标版本号, 例如目标版本号为v1.8.1, 则实际命令为./link-iotedge.sh --restart v1.8.1。

・清理Link IoT Edge。

执行如下命令可停止当前运行的Link IoT Edge相关容器,并会删除所有已安装的相关镜像,删除相关数据卷以及启动配置文件。

```
./link-iot-edge.sh --clean
```

· 提取Link IoT Edge的日志。

执行如下命令可打包Link IoT Edge的所有日志,并拷贝到当前目录。

./link-iot-edge.sh --packagelog

1.2 标准版环境搭建

1.2.1 基于Ubuntu 16.04搭建环境

本文介绍基于Ubuntu 16.04系统,如何快速将网关连接到物联网边缘计算控制台,并将网关数据 上传至云端。

Link IoT Edge标准版软件包支持在Ubuntu 16.04 ~ Ubuntu 18.04系统上运行,并在下列平台上进行测试和验证:

架构	操作系统
x86_64	Ubuntu 16.04 ~ Ubuntu 18.04
ARMv7	Ubuntu 16.04 ~ Ubuntu 18.04
ARMv8 (AArch64)	Ubuntu 16.04 ~ Ubuntu 18.04

尽管Link IoT Edge可以在其它版本的Ubuntu操作系统上运行,但为了最佳的稳定性和安全

性,建议您选择在官方支持的系统版本上运行。接下来,我们将基于x86_64 Ubuntu 16.04的平

台上为您介绍Link IoT Edge标准版安装部署的方法。Ubuntu16.04~18.04系统在其它架构上的标准版软件包您可以在发布历史中获取。

准备工作

准备Ubuntu 16.04系统的PC或者虚拟机,并且该PC或者虚拟机能够正常连接网络。如果您不熟悉Ubuntu 16.04的安装,可参考Ubuntu 16.04安装指南。

环境设置

在x86_64 Ubuntu 16.04机器上需要设置Link IoT Edge运行所依赖的环境。

1. 在x86_64 Ubuntu 16.04机器的本地终端窗口或者SSH终端窗口执行以下命令,下载环境检查 工具并运行:

```
wget http://iotedge-web.oss-cn-shanghai.aliyuncs.com/public/
testingTool/link-iot-edge_env-check.sh
sudo chmod +x ./link-iot-edge_env-check.sh
sudo ./link-iot-edge_env-check.sh
```

📋 说明:

link-iot-edge_env-check.sh脚本在Link IoT Edge支

持的平台需要以root权限运行并需要以下Linux系统命令:

printf, echo, cat, ls, expr, grep, test, uname, set, head, sort, cut, uniq, xargs, ifcom

按照运行环境检查工具的提示在您的机器上安装所有必需的依赖项,当检查工具成功运行完成 后,返回如下图信息,表示Link IoT Edge能够在您的机器上成功运行。

```
Linux Kernel Version:
for ARMv7:
RAM:
FLASH:
                  >= 128MB
CPU Architecture: x86 64, ARMv7, ARMv7 VFPv3, ARMv8 64
Check Depended Commands:
            Checking for command wget
            Checking for command realpath
            Checking for command unzip
Checking for command readlink
            Checking for command pidof
             Checking for command grep
            Checking for command ps
            Checking for command xargs
            Checking for command awk
System configuration:
Kernel architecture:
Kernel version: 4.15
Total memory: 7.7g
Available flash space:
C library version: 2.27
C library: Ubuntu GLIBC 2.27-3ubuntu1
Init process: /lib/systemd/systemd
Loopback is: up
1. It looks like the kernel uses 'systemd' as the init process.
Please using systemd to manage Link IoT Edge service.
```

创建边缘实例和网关

1. 物联网平台控制台,选择边缘计算 > 边缘实例。

- 2. 创建一个边缘实例。
 - a. 单击新增实例, 在弹出窗口中设置实例名称。
 - b. 在网关产品后单击新建网关产品,为实例创建网关。

物联网边缘计算中的网关,承载边缘计算能力,每个实例必须分配一个网关设备,并且该网 关设备同一时间只能被分配到一个边缘实例。

物联网平台	边缘实例 💿					
快速入门						
设备管理	实例列表					刷新新建实例
规则引擎	实例 > 请输入实例名称		×			
数据分析	实例名称	新增实例		实例状态	创建时间	操作
边缘计算	and the second s	◆ 实例名称:		 未部署 	2018-12-13 10:52:57	查看 删除
辺線尖例	10000000	* 网关产品:		 部署成功 	2018-12-20 23:59:45	查看 删除
开发服务		请选择网关产品	~ 新建网关产品	 部署成功 	2018-12-21 10:39:08	查看 删除
应用托管	100,0000	标签信息: +新增标签		 部署失敗 	2018-09-21 20:26:51	查看 删除
视频服务	waterio			 未部署 	2018-11-27 16:34:50	查看 删除
产品文档	4,914		确定现消	 未部署 	2018-11-16 17:29:21	查看 删除

c. 在新建产品页面中,设置网关产品参数,然后单击完成。

物联网边缘计算中的新建网关产品功能继承物联网平台设备管理 > 产品中,高级版产品的功能。已自动为您简化创建适合物联网边缘计算中使用的网关产品步骤。

新建产品		×
产品信息		
∗ 产品名称 LinkloTEdge_Gateway		
* 所属分类 • 所属分类 自定义品类		功能定义
雨久信自		
产品描述		
请输入产品描述		
	0/100	
使用文档		取消 完成

参数说明如下:

参数	说明
产品名称	为网关产品设置名称,用于后续的查询及识别网关产品。支持中 文、英文字母、数字和下划线,长度限制4~30,一个中文汉字算2 位。

参数	说明
所属分类	选择品类,为该产品定义 _{物模型} 。 可选择为:
	 ・自定义品类:需根据实际需要,定义产品功能。 ・任一既有功能模板。
	选择任一物联网平台预定义的品类,快速完成产品的功能定义。 选择产品模板后,您可以在该模板基础上,编辑、修改、新增功 能。
	若您需要的网关没有特殊功能定义,建议您选择自定义品类。

参数	说明
产品描述	可输入文字,用来描述产品信息。字数限制为100。可以为空。

产品创建成功后,页面自动跳转回新增实例页面,并且网关产品参数下自动分配了刚创建的 网关产品。

d. 在新增实例页面,单击网关设备后的新建网关设备为网关产品添加设备。

物联网边缘计算中的新建网关设备功能继承物联网平台设备管理 > 设备的功能。

新增实例	×
* 实例名称:	
LinkloTEdge_Node	
* 网关产品:	
LinkloTEdge_Gateway	→ 新建网关产品
* 网关设备:	
请选择网关设备	~ 新建网关设备
标签信息:	
+新增标签	
	确定 取消

e. 根据界面提示设置参数后,单击确认。

参数说明如下:

参数	描述
产品	系统已自动关联上一步创建的网关产品。

参数	描述
设备名称	为该网关设备命名。设备名称需保持产品内唯一。如不填写,系统 将自动生成。
	 说明: 设备名称支持大写字母[A-Z]、小写字母[a-z]、数字[0-9]和下划 线(_)。且不能以下划线开头或结尾。

- f. (可选)在新增实例页面,单击新增标签,可以设置实例标签。通过标签您可以更加有效地 归类及识别实例。您也可以不设置标签。
- 3. 实例参数设置完成后,单击确定,至此您已创建边缘实例和网关。
- 4. 在实例详情 > 实例信息页面, 网关页签下, 单击网关名称右侧的查看, 获取网关设备信息。

边缘实例 → 实例详 资											
LinkloTEdge_Node 未認習 重置 部署											部署
CPU 使用率: - 查看		内存使用率: - 查看	内存使用率:- 查看 存储使用率:-			i摔: - 查看 服务版					
实例信息	子设备 子设备通信通道	场景联动 函数	计算 流数据分	浙 消息路由	日志服务	设置					
实例信息	实例信息										编辑
实例名称	LinkloTEdge_Node		部署状态	未部署			云监控状态				
CPU 使用率	- 查看		内存使用率	- 查看			存储使用率	- 查看			
实例进程	查看		服务版本号	-			创建时间	2019-03-12 15	:51:20		
修改时间	2019-03-12 15:51:20										
实例标签											
网关										分配	
网关名称		产品名称			状态	最近	后上线时间	操作			
gateway_01		dge_Gateway		 未激活 	£ -		<u> 音</u> 看 移	除 远程连接 远相			

5. 系统跳转到网关设备的设备详情页面,在设备详情页面获取网关设备的设备证

书(ProductKey、DeviceName、DeviceSecret),用于后续启动网关。

设备管理 > 设备详情										
gateway_01 (未認活										
产品:LinkloTEdge_Gateway 查看 ProductKey: 复制 DeviceSecret:******* 显示										
设备信息 Topic列表 运行状态 事件管理 服务调用 日志服务 子设备管理 子设备通道管理										
设备信息										
产品名称	LinkloTEdge_Gateway	ProductKey	复制]	区域	•				
节点类型	网关	DeviceName	复制]	DeviceSecret	******** 显示				
当前状态	未激活	IP地址	-		固件版本	-				
添加时间	2019/01/11 17:06:58	激活时间			最后上线时间					
实时延迟 🔘	测试									
标签信息										
设备标签:无标签信息,立即添加										

启动Link IoT Edge

- 1. 登录您的x86_64 Ubuntu 16.04机器。
- 2. 下载Link IoT Edge软件包。

执行如下命令下载适用于x86_64 Ubuntu 16.04系统的Link IoT Edge软件包。

```
wget http://link-iot-edge-packet.oss-cn-shanghai.aliyuncs.com/x86-64
-linux-gnu/link-iot-edge-x86-64-v1.8.1.tar.gz
```

🗾 说明:

以上命令中使用的下载软件包地址为x86_64 Ubuntu 16.04系统v1.8.1版本软件包地址,其他版本LE Standard软件包下载地址请见发布历史。

3. 以root用户执行以下命令解压并安装Link IoT Edge软件包。

该命令在网关设备的根目录中创建 /linkedge/gateway目录, 并将软件包安装到/linkedge /gateway/build目录。

```
sudo tar xzvf link-iot-edge-x86-64-v1.8.1.tar.gz -C /
```

4. 配置网关初始化参数到安全存储。

```
cd /linkedge/gateway/build/script
sudo ./set_gw_triple.sh {YourProductKey} {YourDeviceName} {
YourDeviceSecret}
```

📕 说明:

请将{YourProductKey} {YourDeviceName} {YourDeviceSecret}替换为已在本地保存的网关设备的设备证书信息。

5. 启动Link IoT Edge核心服务。

cd /linkedge/gateway/build/script

sudo ./iot_gateway_start.sh

6. 执行如下命令查看Link IoT Edge核心服务的运行状态。

sudo ./iot_gateway_status.sh

若系统显示如下信息,表示Link IoT Edge核心服务启动成功。

ro	root@arm:/linkedge/gateway/build/script# ./iot_gateway_status.sh								
			Link IoT Edge Service Status						
[Service mbusd.conf is active, pid is 2307						
[Service config-manager is active, pid is 2312						
[Service logger is active, pid is 2319						
[Service data-manager is active, pid is 2343						
[Service watch-dog is active, pid is 2346						
[Service cloud-proxy is active, pid is 2342						
[Service dimu is active, pid is 2357						
[Service message-router is active, pid is 2384						
[Service fota is active, pid is 2391						
[Service log-uploader is active, pid is 2395						
[Service gateway-monitor is active, pid is 2398						
[Service task-dispatcher is active, pid is 2404						
[Service file-uploader is active, pid is 2409						
[Service wss-driver is active, pid is 2414						
[Service fc-base is active, pid is 2586						
[Service ifttt is active, pid is 2512						

您也可以在物联网控制台,选择边缘计算 > 边缘实例,在已创建好的边缘实例右侧单击查看进 入实例详情页面,查看网关状态。

2.5%文例 → 交例洋情											
LinkloTEdge_Node 末端 重置 部署											部署
CPU 使用率: - 查看		存储使用率: - 宣看				服务版本号: -					
实例信息	子设备 子设备通信通道	场景联动 函数	计算 流数据分	析 消息路由	日志服务	设置					
实例信息										篇辑	
实例名称	LinkloTEdge_Node		部署状态	素部署			云监控状态				
CPU 使用率	- 查香		内存使用率	 查看 			存储使用率	 查看 			
实例进程	查看		服务版本号	-			创建时间	2019-03-12 15	:51:20		
修改时间	2019-03-13 17:11:00										
实例标签											
											咨询
网关										分配	网关 <mark>建</mark> 议
网关名称		产品名称	t		状态	II.	后上线时间	操作			
gateway_01		LinkloTE	dge_Gateway		 ● 在約 	戋 20	019-03-14 15:23:44	查看《	多除 远程连接 远程	文件管理	
- 7. 在实例详情页面中,查看CPU使用率、内存使用率、存储使用率以及实例进程需要授权访问阿 里云云监控(CloudMonitor)服务。
 - a. 请参考云资源访问内容,在RAM控制台,创建授信IoT物联网的服务角色,并为该角色添加名 为AliyunCloudMonitorFullAccess的访问云监控服务的权限。
 - b. 在实例信息页签下打开云监控状态,如下图所示。

边缘实例 → 实例评情											
LinkloTEdge_Node 部署成功 查看部署日志 重置 部署											
CPU使用率: 20.63% 查看 内存使用率: 48.57		内存使用率: 48.57	% 查看	6 查看 存储使用率: 38 % 查看			服务版本号: v1.8.2				
实例信息	文例信息 子设备 子设备通信通道 场景联动 函数		计算 流数据分	计算 流数据分析 消息路由 日志服务 设置							
实例信息											编辑
实例名称	LinkloTEdge_Node		部署状态	部署成功			云监控状态				
CPU 使用率	20.63 % 查看		内存使用率	48.57 % 查看			存储使用率	38 % 查看			
实例进程	查看		服务版本号	v1.8.2			创建时间	2019-03-12 15:51:2	20		
修改时间	2019-03-13 17:11:00										
实例标签											

c. 在实例进程后,单击查看,可查看实例各个进程的信息及CPU、内存使用率。

边缘实例 > 实例详情								
LinkloTEdge_Node 部署成功 查看部署日志 重置 部署								
CPU 使用率 13.32% 查看							服务版本号: v1.8.2	
实例信息	子设备 子设备					Blief		
立例信息						M33V1		從相
大四百心		进程 ID	进程名	CPU	内存			3614
实例名称	LinkloTEdge_No	280	task-dispatcher	5.17 %	0.61 %		云监控状态	
CPU 使用率	13.32 % 查看	431	node	1.46 %	3.41 %		存储使用率	38 % 查看
实例进程	直看	446	fc-base	1.13 %	1.91 %		创建时间	2019-03-12 15:51:20
修改时间	2019-03-13 17:11	204	logger	0.73 %	0.2 %			
实例标签		261	message-router	0.93 %	0.27 %			
网关						关闭		

 您可以在实例详情页面,网关名称右侧的操作栏中单击远程连接或者远程文件管理,方便您远程 控制网关设备或对网关设备上的文件进行管理。详细说明请参见远程运维管理。

使用systemd管理Link IoT Edge

您可以使用systemd来管理Link IoT Edge服务的启动(start)、停止(stop)和查看状态(status)。

Link IoT Edge的systemd service如下所示:

```
[Unit]
Description=Link IoT Edge
```

```
[Service]
Type=forking
Restart=on-failure
```

```
ExecStart=/linkedge/gateway/build/script/iot_gateway_start.sh
ExecReload=/linkedge/gateway/build/script/iot_gateway_start.sh
ExecStop=/linkedge/gateway/build/script/iot_gateway_stop.sh
```

[Install] WantedBy=multi-user.target

您可运行如下命令下载该service文件,并拷贝到/etc/systemd/system/目录。

```
wget http://iotedge-web.oss-cn-shanghai.aliyuncs.com/public/testingToo
l/LinkIoTEdge.service
sudo cp LinkIoTEdge.service /etc/systemd/system/LinkIoTEdge.service
```

可以使用如下命令启动或者重启服务。

- ・ 启动命令: sudo systemctl start LinkIoTEdge.service
- ・重启命令: sudo systemctl restart LinkIoTEdge.service

可以使用如下命令停止服务。

sudo systemctl stop LinkIoTEdge.service

可以使用如下命令设置开机自动启动服务。

sudo systemctl enable LinkIoTEdge.service

1.2.2 基于树莓派搭建环境

本文介绍基于树莓派(Raspberry Pi)系统,如何快速将网关连接到物联网边缘计算控制台,并 将网关数据上传至云端。

Link IoT Edge标准版软件包支持在树莓派系统上运行,并在下面平台上进行了测试和验证:

硬件	操作系统
树莓派3B	Raspbian

理论上Link IoT Edge也可以在其它版本的树莓派操作系统(树莓派2B/3B/3B+)上运行,但为了 最佳的稳定性和安全性,建议您选择在官方支持的系统版本上运行。

准备工作

准备符合如下表格中要求的树莓派设备。

项目	规格
树莓派	树莓派3B
操作系统	Raspbian Stretch with desktop
内核	≥ 4.13

项目	规格
Micro SD卡	≥ 8GB
外设	HDMI线、键盘、鼠标、MicroUSB线

环境设置

在树莓派设备上需要设置Link IoT Edge运行所依赖的环境,如果您的树莓派设备上已经安装好 Raspbian,请跳过步骤1至步骤3。

- 1. 下载Raspbian镜像和SD镜像烧写工具。
 - ·操作系统镜像: Raspbian Stretch with desktop
 - · SD卡镜像烧写工具: Etcher



下载适合您计算机操作系统的SD卡镜像烧写工具。

2. 使用Etcher将Raspbian镜像烧写到SD卡。

将Micro SD卡插入读卡器,并接入到您的计算机。请参考树莓派操作系统镜像安装指 南将Raspbian镜像烧写到Micro SD卡。

3. 启动并配置Raspbian操作系统。

将Micro SD卡插入您的树莓派设备上,连接好鼠标和键盘,通过HDMI线连接到监视器上,使用MicroUSB线进行供电,启动您的树莓派设备。

a. 通过WiFi网络或者以太网将树莓派设备连接到Internet,配置方法请参考树莓派配置^{WiFi}无线网络。

b. 配置SSH服务。

- ・执行sudo systemctl start ssh命令开启SSH服务。
- ·执行sudo systemctl status ssh命令查看SSH服务。
- · 执行sudo systemctl enable ssh设置开机自动启动SSH服务。

在启动SSH服务之后,执行hostname -I命令获取树莓派的IP地址,通过网络SSH远程登录的 方式登录树莓派设备。

- 4. 检查Link IoT Edge运行环境依赖。
 - a. 在树莓派设备的本地终端窗口或者SSH终端窗口执行以下命令,下载环境检查工具并运行:

```
wget http://iotedge-web.oss-cn-shanghai.aliyuncs.com/public/
testingTool/link-iot-edge_env-check.sh
sudo chmod +x ./link-iot-edge_env-check.sh
```

sudo ./link-iot-edge_env-check.sh

说明:

link-iot-edge_env-check.sh脚本在Link IoT Edge支

持的平台需要以root权限运行并需要以下Linux系统命令:

printf, echo, cat, ls, expr, grep, test, uname, set, head, sort, cut, uniq, xargs, ifc

b. 按照运行环境检查工具的提示在您的机器上安装所有必需的依赖项,当检查工具成功运行完成后,返回如下图信息,表示Link IoT Edge能够在您的机器上成功运行。

******************************** Eink IoT Edge v1.8 Run Environment Check************************************						
System configuration:						
C library version: 2.23						
C library: Ubuntu GLIBC 2.23-Oubuntu10						
Init process: /lib/systemd/systemd						
Check Depended Software Version:						
OpenSSL version: 1.0.1						
Python version: 3,5.2						
Check Depended Commands:						
(ves) Checking for command weet						
(ves) Checking for command realpath						
(yes] Checking for command tar						
[ves] Checking for command unzip						
[ves] Checking for command readlink						
[ves] Checking for command basename						
(ves) Checking for command dirname						
(ves) Checking for command pidof						
(ves) Checking for command df						
[ves] Checking for command grep						
[yes] Checking for command ps						
[yes] Checking for command kill						
[yes] Checking for command dirname						
[yes] Checking for command xargs						
[yes] Checking for command umount						
[yes] Checking for command unshare						
[yes] Checking for command awk						
[yes] Checking for command mkdir						
**************Link IoT Edge v1.8 Run Environment Check Result:************						
Suggestion:						
1. It looks like the kernel uses 'systemd' as the init process.						
Please using systemd to manage Link IoT Edge service.						
You can install the Link loT Edge v1.8 software on the device.						

创建边缘实例和网关

1. 物联网平台控制台,选择边缘计算 > 边缘实例。

- 2. 创建一个边缘实例。
 - a. 单击新增实例, 在弹出窗口中设置实例名称。
 - b. 在网关产品后单击新建网关产品,为实例创建网关。

物联网边缘计算中的网关,承载边缘计算能力,每个实例必须分配一个网关设备,并且该网 关设备同一时间只能被分配到一个边缘实例。

物联网平台	边缘实例 💿					
快速入门						
设备管理	实例列表					刷新新建实例
规则引擎	实例 ∨ 请输入实例名称		×			
数据分析	实例名称	新增实例		实例状态	创建时间	操作
边缘计算	data.	* 实例名称: LinkloTEdge Node		 未部署 	2018-12-13 10:52:57	查看 删除
驱动管理	1000010	* 网关产品:		 部署成功 	2018-12-20 23:59:45	查看 删除
开发服务	100000121	请选择网关产品	◇ 新建网关产品	 部署成功 	2018-12-21 10:39:08	查看 删除
应用托管	test, property	恢金信息: +新婚祝签		 部署失敗 	2018-09-21 20:26:51	查看 删除
监控运维	10000			 未部署 	2018-11-27 16:34:50	查看 删除
产品文档	1,010		确定取消	• 未部署	2018-11-16 17:29:21	查看 删除

c. 在新建产品页面中,设置网关产品参数,然后单击完成。

物联网边缘计算中的新建网关产品功能继承物联网平台设备管理 > 产品中,高级版产品的功能。已自动为您简化创建适合物联网边缘计算中使用的网关产品步骤。

新建产品			×
产品信息			
* 产品名称			
LinkloTEdge_Gateway			
* 所属分类 💿			
自定义品类	\sim	功能定义	
更多信息			
产品描述			
请输入产品描述			
	0/100		
使用文档		取消	完成

参数说明如下:

参数	说明
产品名称	为网关产品设置名称,用于后续的查询及识别网关产品。支持中 文、英文字母、数字和下划线,长度限制4~30,一个中文汉字算2 位。

参数	说明
所属分类	选择品类,为该产品定义 <u>物模型</u> 。 可选择为:
	 ・自定义品类:需根据实际需要,定义产品功能。 ・任一既有功能模板。
	选择任一物联网平台预定义的品类,快速完成产品的功能定义。 选择产品模板后,您可以在该模板基础上,编辑、修改、新增功 能。
	若您需要的网关没有特殊功能定义,建议您选择自定义品类。

参数	说明
产品描述	可输入文字,用来描述产品信息。字数限制为100。可以为空。

产品创建成功后,页面自动跳转回新增实例页面,并且网关产品参数下自动分配了刚创建的 网关产品。

d. 在新增实例页面,单击网关设备后的新建网关设备为网关产品添加设备。

物联网边缘计算中的新建网关设备功能继承物联网平台设备管理 > 设备的功能。

新增实例	×
* 实例名称:	
LinkloTEdge_Node	
* 网关产品:	
LinkloTEdge_Gateway	∨ 新建网关产品
* 网关设备:	
请选择网关设备	→ 新建网关设备
标签信息:	
+新增标签	
	确定 取消

e. 根据界面提示设置参数后,单击确认。

参数说明如下:

参数	描述
产品	系统已自动关联上一步创建的网关产品。

参数	描述
设备名称	为该网关设备命名。设备名称需保持产品内唯一。如不填写,系统 将自动生成。
	 说明: 设备名称支持大写字母[A-Z]、小写字母[a-z]、数字[0-9]和下划 线(_)。且不能以下划线开头或结尾。

- f. (可选)在新增实例页面,单击新增标签,可以设置实例标签。通过标签您可以更加有效地 归类及识别实例。您也可以不设置标签。
- 3. 实例参数设置完成后,单击确定,至此您已创建边缘实例和网关。
- 4. 在实例详情 > 实例信息页面, 网关页签下, 单击网关名称右侧的查看, 获取网关设备信息。

边缘实例》 实例详	NH F										
LinkloTEdge_N	lode 未部署								查看部署日志	重置	部署
CPU 使用率: - 查看		内存使用率:- 查看		存储	諸使用率: - 查看			服务版本号: -			
实例信息	子设备 子设备通信通道	场景联动 函数	计算 流数据分	浙行 消息路由	日志服务	设置					
实例信息											编辑
实例名称	LinkloTEdge_Node		部署状态	未部署			云监控状态				
CPU 使用率	- 查看		内存使用率	- 查看			存储使用率	- 查看			
实例进程	查看		服务版本号	-			创建时间	2019-03-12 1	5:51:20		
修改时间	2019-03-12 15:51:20										
实例标签											
网关										分香	
网关名称		产品名称			状态	最近	后上线时间	操作			
gateway_01		LinkloTEd	dge_Gateway		 未激演 	£ -		查看	多除 远程连接 远移		

5. 系统跳转到网关设备的设备详情页面,在设备详情页面获取网关设备的设备证

书(ProductKey、DeviceName、DeviceSecret),用于后续启动网关。

设备管理 > 设备详情							
gateway_01	gateway_01 未激活						
产品:LinkloTEdge_Ga	teway 查看	ProductKey :	复制		DeviceSecret : ********	显示	
设备信息 Topic	列表 运行状态 事件管理 服务调	用 日志服务 子	设备管理 子设备通道	管理			
设备信息							
产品名称	LinkloTEdge_Gateway	ProductKey	复制]	区域	•	
节点类型	网关	DeviceName	复制]	DeviceSecret	******* 显 示	
当前状态	未激活	IP地址			固件版本	-	
添加时间	2019/01/11 17:06:58	激活时间			最后上线时间		
实时延迟 🔘	测试						
标签信息							
设备标签:无标签信息,	立即添加						

启动Link IoT Edge

1. 下载Link IoT Edge软件包。

执行如下命令下载适用于Raspbian系统的Link IoT Edge软件包。

wget http://link-iot-edge-packet.oss-cn-shanghai.aliyuncs.com/armlinux-gnueabihf/link-iot-edge-armv7-hf-v1.8.1.tar.gz



以上命令中使用的下载软件包地址为Raspbian系统v1.8.1版本软件包地址,其他版本LE Standard软件包下载地址请见发布历史。

2. 以root用户执行以下命令解压并安装Link IoT Edge软件包。

该命令在网关设备的根目录中创建 /linkedge/gateway目录, 并将软件包安装到/linkedge

/gateway/build目录。

sudo tar xzvf link-iot-edge-armv7-hf-v1.8.1.tar.gz -C /

3. 配置网关初始化参数到安全存储。

```
cd /linkedge/gateway/build/script
sudo ./set_gw_triple.sh {YourProductKey} {YourDeviceName} {
YourDeviceSecret}
```

〕 说明:

请将{YourProductKey} {YourDeviceName} {YourDeviceSecret}替换为已在本地保存的网关设备的设备证书信息。

4. 启动Link IoT Edge核心服务。

cd /linkedge/gateway/build/script

sudo ./iot_gateway_start.sh

5. 执行如下命令查看Link IoT Edge核心服务的运行状态。

sudo ./iot_gateway_status.sh

若系统显示如下信息,表示Link IoT Edge核心服务启动成功。

ro	ot@a	arm	:/linkedge/gateway/build/script# ./iot_gateway_status.	sh
			Link IoT Edge Service Status	
[Service mbusd.conf is active, pid is 2307	
[Service config-manager is active, pid is 2312	
[Service logger is active, pid is 2319	
[Service data-manager is active, pid is 2343	
[Service watch-dog is active, pid is 2346	
[Service cloud-proxy is active, pid is 2342	
[Service dimu is active, pid is 2357	
[Service message-router is active, pid is 2384	
[Service fota is active, pid is 2391	
[Service log-uploader is active, pid is 2395	
[Service gateway-monitor is active, pid is 2398	
[Service task-dispatcher is active, pid is 2404	
[Service file-uploader is active, pid is 2409	
[Service wss-driver is active, pid is 2414	
[Service fc-base is active, pid is 2586	
[Service ifttt is active, pid is 2512	

您也可以在物联网控制台,选择边缘计算 > 边缘实例,在已创建好的边缘实例右侧单击查看进 入实例详情页面,查看网关状态。

边缘实例 > 实例详	隋										
LinkloTEdge_N	lode 未部署								查看部署日志	重置	部署
CPU 使用率: - 查看		内存使用率: - 查看		存(诸使用率: - 查看			服务版本号: -			
实例信息	子设备 子设备通信通道	场景联动 函数	计算 流数据分	沂 消息路由	日志服务	设置					
实例信息										ŝ	謝
实例名称	LinkloTEdge_Node		部署状态	未部署			云监控状态				
CPU 使用率	- 查看		内存使用率	 查看 			存储使用率	- 查看			
实例进程	查看		服务版本号	-			创建时间	2019-03-12 15	:51:20		
修改时间	2019-03-13 17:11:00										
实例标签											
											咨询
网关										分配网	· 建
网关名称		产品名称	t		状态		后上线时间	操作			
gateway_01		LinkloTE	dge_Gateway		 在线 	20)19-03-14 15:23:44	查看 利	多除 远程连接 远程	文件管理	

- 6. 在实例详情页面中,查看CPU使用率、内存使用率、存储使用率以及实例进程需要授权访问阿 里云云监控(CloudMonitor)服务。
 - a. 请参考云资源访问内容,在RAM控制台,创建授信IoT物联网的服务角色,并为该角色添加名 为AliyunCloudMonitorFullAccess的访问云监控服务的权限。
 - b. 在实例信息页签下打开云监控状态,如下图所示。

边缘实例》 实例详	情										
LinkloTEdge_N	Node 部署成功								看部署日志	重置	部署
CPU 使用率: 20.63 9	% 查看	内存使用率: 48.57	% 查看	存储的	起用率: 38 % 查 ?	6		服务版本号: v1.8.2			
实例信息	子设备 子设备通信通道	场景联动 函数	计算 流数据分	淅 消息路由	日志服务	设置					
实例信息											编辑
实例名称	LinkloTEdge_Node		部署状态	部署成功			云监控状态				
CPU 使用率	20.63 % 查看		内存使用率	48.57 % 查看			存储使用率	38 % 查看			
实例进程	查看		服务版本号	v1.8.2			创建时间	2019-03-12 15:51:2	20		
修改时间	2019-03-13 17:11:00										
实例标签											

c. 在实例进程后,单击查看,可查看实例各个进程的信息及CPU、内存使用率。

边缘实例 > 实例详	情							
LinkloTEdge_N	Node 部署成功							查看部署日志 重置 部署
CPU 使用率: 13.32 9	% 查看	2例进程				- ×		服务版本号: v1.8.2
实例信息	子设备 子设备					Blief		
立例信息						M33V1		伦 博
		进程 ID	进程名	CPU	内存			3614
实例名称	LinkloTEdge_No	280	task-dispatcher	5.17 %	0.61 %		云监控状态	
CPU 使用率	13.32 % 查看	431	node	1.46 %	3.41 %		存储使用率	38 % 查看
实例进程	直看	446	fc-base	1.13 %	1.91 %		创建时间	2019-03-12 15:51:20
修改时间	2019-03-13 17:11	204	logger	0.73 %	0.2 %			
实例标签		261	message-router	0.93 %	0.27 %			
网关						关闭		

 您可以在实例详情页面,网关名称右侧的操作栏中单击远程连接或者远程文件管理,方便您远程 控制网关设备或对网关设备上的文件进行管理。详细说明请参见远程运维管理。

使用systemd管理Link IoT Edge

您可以使用systemd来管理Link IoT Edge服务的启动(start)、停止(stop)和查看状态(status)。

Link IoT Edge的systemd service如下所示:

```
[Unit]
Description=Link IoT Edge
[Service]
Type=forking
Restart=on-failure
```

```
ExecStart=/linkedge/gateway/build/script/iot_gateway_start.sh
ExecReload=/linkedge/gateway/build/script/iot_gateway_start.sh
ExecStop=/linkedge/gateway/build/script/iot_gateway_stop.sh
```

[Install] WantedBy=multi-user.target

您可运行如下命令下载该service文件,并拷贝到/etc/systemd/system/目录。

```
wget http://iotedge-web.oss-cn-shanghai.aliyuncs.com/public/testingToo
l/LinkIoTEdge.service
sudo cp LinkIoTEdge.service /etc/systemd/system/LinkIoTEdge.service
```

可以使用如下命令启动或者重启服务。

- ・启动命令: sudo systemctl start LinkIoTEdge.service
- ・重启命令: sudo systemctl restart LinkIoTEdge.service

可以使用如下命令停止服务。

sudo systemctl stop LinkIoTEdge.service

可以使用如下命令设置开机自动启动服务。

sudo systemctl enable LinkIoTEdge.service

1.2.3 基于阿里云Cloud Shell搭建环境

本文介绍基于阿里云云命令行(Cloud Shell),快速将网关连接到物联网边缘计算控制台,并将 网关数据上传至云端。Cloud Shell是阿里云提供的网页版命令行工具。您可以在任意浏览器上运 行云命令行管理阿里云资源。

然而在Cloud Shell中操作时,若连续15分钟不做任何操作,系统会自动退出,并清空您保存 在Cloud Shell上的资料,因此不建议您使用Cloud Shell作为开发生产环境,仅用于快速了解并 无任何门槛地使用Link IoT Edge。在开始操作本章内容前,请您仔细阅读*Cloud Shell介绍及使*用 限制。

Cloud Shell仅支持搭建v1.8.2及以上版本的Link IoT Edge标准版(LE Standard)软件包。

创建边缘实例和网关

1. 物联网平台控制台,选择边缘计算 > 边缘实例。

- 2. 创建一个边缘实例。
 - a. 单击新增实例, 在弹出窗口中设置实例名称。
 - b. 在网关产品后单击新建网关产品,为实例创建网关。

物联网边缘计算中的网关,承载边缘计算能力,每个实例必须分配一个网关设备,并且该网 关设备同一时间只能被分配到一个边缘实例。

物联网平台	边缘实例 💿					
快速入门						
设备管理	实例列表				F	制新新建实例
规则引擎	实例 ~ 请输入实例名称		×			
数据分析	实例名称	新增实例		实例状态	创礁时间	操作
边缘计算	ana .	* 实例名称: LinkloTEdge Node		 未部署 	2018-12-13 10:52:57	查看 删除
驱动管理	10000010	* 网关产品:		 部署成功 	2018-12-20 23:59:45	查看 删除
开发服务	1000000-02-	请选择网关产品	→ 新建网关产品	● 部署成功	2018-12-21 10:39:08	查看 删除
应用托管	ted, prost	标验信息: +新婚初整		• 部署失败	2018-09-21 20:26:51	查看 删除
视频服务				 未部署 	2018-11-27 16:34:50	查看 删除
产品文档	1,010		确定 取消	 未部署 	2018-11-16 17:29:21	查看 删除

c. 在新建产品页面中,设置网关产品参数,然后单击完成。

物联网边缘计算中的新建网关产品功能继承物联网平台设备管理 > 产品中,高级版产品的功能。已自动为您简化创建适合物联网边缘计算中使用的网关产品步骤。

新建产品		×
产品信息		
∗ 产品名称 LinkloTEdge_Gateway		
* 所属分类 • 所属分类 自定义品类		功能定义
雨久/自		
产品描述		
请输入产品描述		
	0/100	
使用文档		取消 完成

参数说明如下:

参数	说明
产品名称	为网关产品设置名称,用于后续的查询及识别网关产品。支持中 文、英文字母、数字和下划线,长度限制4~30,一个中文汉字算2 位。

参数	说明
所属分类	选择品类,为该产品定义 <u>物模型</u> 。 可选择为:
	 ・自定义品类:需根据实际需要,定义产品功能。 ・任一既有功能模板。
	选择任一物联网平台预定义的品类,快速完成产品的功能定义。 选择产品模板后,您可以在该模板基础上,编辑、修改、新增功 能。
	若您需要的网关没有特殊功能定义,建议您选择自定义品类。

参数	说明
产品描述	可输入文字,用来描述产品信息。字数限制为100。可以为空。

产品创建成功后,页面自动跳转回新增实例页面,并且网关产品参数下自动分配了刚创建的 网关产品。

d. 在新增实例页面,单击网关设备后的新建网关设备为网关产品添加设备。

物联网边缘计算中的新建网关设备功能继承物联网平台设备管理 > 设备的功能。

新增实例	×
* 实例名称:	
LinkloTEdge_Node	
* 网关产品:	
LinkloTEdge_Gateway	→ 新建网关产品
* 网关设备:	
请选择网关设备	→ 新建网关设备
标签信息:	
+新增标签	
	确定 取消

e. 根据界面提示设置参数后,单击确认。

参数说明如下:

参数	描述
产品	系统已自动关联上一步创建的网关产品。

参数	描述
设备名称	为该网关设备命名。设备名称需保持产品内唯一。如不填写,系统 将自动生成。
	 说明: 设备名称支持大写字母[A-Z]、小写字母[a-z]、数字[0-9]和下划 线(_)。且不能以下划线开头或结尾。

- f. (可选)在新增实例页面,单击新增标签,可以设置实例标签。通过标签您可以更加有效地 归类及识别实例。您也可以不设置标签。
- 3. 实例参数设置完成后,单击确定,至此您已创建边缘实例和网关。
- 4. 在实例详情 > 实例信息页面, 网关页签下, 单击网关名称右侧的查看, 获取网关设备信息。

边缘实例 > 实例详	情										
LinkloTEdge_N	Node 未部署								查看部署日志	重置	部署
CPU 使用率: - 查看		内存使用率: - 查看		存储	諸使用率: - 查看			服务版本号: -			
实例信息	子设备 子设备通信通道	场景联动 函数	计算 流数据分	浙 消息路由	日志服务	设置					
实例信息											编辑
实例名称	LinkloTEdge_Node		部署状态	未部署			云监控状态				
CPU 使用率	- 查看		内存使用率	- 查看			存储使用率	- 查看			
实例进程	查看		服务版本号	-			创建时间	2019-03-12 15	:51:20		
修改时间	2019-03-12 15:51:20										
实例标签											
网关										分配	
网关名称		产品名称			状态	最近	后上线时间	操作			
gateway_01		LinkloTE	dge_Gateway		 未激活 	£ -		<u> 音</u> 看 移	SR 远程连接 远格		

5. 系统跳转到网关设备的设备详情页面,在设备详情页面获取网关设备的设备证

书(ProductKey、DeviceName、DeviceSecret),用于后续启动网关。

设备管理 > 设备详情										
gateway_01 未熟活										
产品: LinktoTEdge_Gateway 查看 ProductKey: 复制 DeviceSecret: ******** 显示										
设备信息 Topic	设备信息 Topic列表 运行状态 事件管理 服务调用 日志服务 子设备管理 子设备通道管理									
设备信息										
产品名称	LinkloTEdge_Gateway	ProductKey	复制]	区域	•				
节点类型	网关	DeviceName	复制]	DeviceSecret	******** 显示				
当前状态	未激活	IP地址			固件版本	-				
添加时间	2019/01/11 17:06:58	激活时间			最后上线时间					
्रमम्बर अ त										
标签信息										
设备标签:无标签信息,	, 立即添加									

启动Link IoT Edge

- 1. 以阿里云账号登录 Cloud Shell。
- 2. 在授权提示窗口中,单击确认授权Cloud Shell获取临时的会话访问密钥。



3. 在挂载存储空间提示窗口中,选择暂不创建。



等待15~30秒后系统成功登录Cloud Shell,操作界面如下图所示:



4. 执行如下命令,查看Cloud Shell运行环境的CPU。

cat /proc/cpuinfo

ር 🕸 🗘 🔞		
Requesting a Cl Connecting term	ou ina	d Shell <i>Succeeded.</i> al
Welcome to Alib	aba	a Cloud Shell
Type "aliyun" t	0 1	nse Alibaba Cloud CLI
shell@Alicloud:	~\$	cat /proc/cpuinfo
processor		0
vendor id		GenuineIntel
cpu family		6
model		79
model name		Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50GHz
stepping		1
microcode		0x1
cou MHz		2499,996
cache size		40960 KB
physical id		0
siblings		1
core id		0
cpu cores		1
apicid		0
initial apicid		0
fpu		ves
fpu exception		ves
cpuid level		13
wp		yes
flags		fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp
1m constant tsc	re	ep good nopl eagerfpu pni pclmulqdq ssse3 fma cx16 pcid sse4 1 sse4 2 x2apic movbe popcnt tsc deadline timer aes xsave avx f16c rdr
and hypervisor	lal	hf lm abm 3dnowprefetch ibrs ibpb stibp fsgsbase tsc adjust bmil hle avx2 smep bmi2 erms invpcid rtm rdseed adx smap xsaveopt spec
ctrl intel stib	ρ	
bogomips		4999.99
clflush size		64
cache alignment		64
address sizes		46 bits physical, 48 bits virtual
power management	t	

CPU为Intel E5-2682,因此使用Link IoT Edge标准版(LE Standard)x86_64架构安装 包。

5. 执行如下命令下载适用于x86_64架构的Link IoT Edge软件包。

```
wget http://link-iot-edge-packet.oss-cn-shanghai.aliyuncs.com/x86-64
-linux-gnu/link-iot-edge-x86-64-v1.8.2.tar.gz
```

📃 说明:

上述命令中使用的下载软件包地址为x86_64系统v1.8.2版本软件包地址,其他高于v1.8.2版本

的LE Standard软件包下载地址请见发布历史。

6. 执行如下命令解压并安装Link IoT Edge软件包。

```
tar xzvf link-iot-edge-x86-64-v1.8.2.tar.gz -C /tmp/ && mv /tmp/
linkedge/* /linkedge && rm -rf /tmp/linkedge
```

7. 配置网关初始化参数到安全存储。

```
cd /linkedge/gateway/build/script
./set_gw_triple.sh {YourProductKey} {YourDeviceName} {YourDevice
Secret}
```

▋ 说明:

请将{YourProductKey} {YourDeviceName} {YourDeviceSecret}替换为已在本地保

存的网关设备的设备证书信息。

./set_gw_triple.sh a1*****gs gateway 2Px*********H1S

- 8. 启动Link IoT Edge核心服务。
 - cd /linkedge/gateway/build/script

./iot_gateway_start.sh

9. 执行如下命令查看Link IoT Edge核心服务的运行状态。

./iot_gateway_status.sh

若系统显示如下信息,表示Link IoT Edge核心服务启动成功。

<pre>shell@Alicloud:/linkedge/gateway/build/script\$./iot gateway status.sh</pre>							
Link IoT Edge Service Status							
<pre>[+] Service mbusd.conf is active, pid is 10053</pre>							
<pre>[+] Service config-manager is active, pid is 10061</pre>							
<pre>[+] Service data-manager is active, pid is 10120</pre>							
[+] Service watch-dog is active, pid is 10129							
<pre>[+] Service cloud-proxy is active, pid is 10096</pre>							
[+] Service dimu is active, pid is 10163							
[+] Service message-router is active, pid is 10178							
[+] Service fota is active, pid is 10189							
<pre>[+] Service gateway-monitor is active, pid is 10198</pre>							
<pre>[+] Service task-dispatcher is active, pid is 10216</pre>							
<pre>[+] Service credential is active, pid is 10087</pre>							
<pre>[+] Service fc-base is active, pid is 10262</pre>							
<pre>[+] Service ifttt is active, pid is 10537</pre>							
<pre>[+] Service redis-server is active, pid is 10052</pre>							
[+] Service logger is active, pid is 10071							
[+] Service service-monitor is active, pid is 10278							
[+] Service RemoteTerminalDaemon is active, pid is 2357							

您也可以在物联网控制台,选择边缘计算 > 边缘实例,在已创建好的边缘实例右侧单击查看进 入实例详情页面,查看网关状态。

也缘实例 > 实例详	時								
LinkloTEdge_N	lode 未部署						直看	部署日志	重置部
CPU 使用率: - 查看		内存使用率: - 查看		存储使用	率:- 查看		服务版本号: -		
实例信息	子设备 子设备通信通道	场景联动 函数)	十算 流数据分	析 消息路由 日調	志服务 设置				
实例信息									编辑
实例名称	LinkloTEdge_Node		部署状态	未部署		云监控状态			
CPU 使用率	- 查看		内存使用率	- 查看		存储使用率	- 查看		
实例进程	查看		服务版本号	-		创建时间	2019-03-12 15:51:20		
修改时间	2019-03-13 17:11:00								
实例标签									
网关									分配网关
网关名称		产品名称			状态	最后上线时间	操作		
gateway_01		LinkloTE	dge_Gateway		● 在线	2019-03-14 15:23:44	查看 移除 远	程连接 远程3	て件管理

10.在实例详情页面中,查看CPU使用率、内存使用率、存储使用率以及实例进程需要授权访问阿 里云云监控(CloudMonitor)服务。

- a. 请参考云资源访问内容,在RAM控制台,创建授信IoT物联网的服务角色,并为该角色添加名 为AliyunCloudMonitorFullAccess的访问云监控服务的权限。
- b. 在实例信息页签下打开云监控状态,如下图所示。

边缘实例》 实例详	情										
LinkloTEdge_N	Node 部署成功								看部署日志	重置	部署
CPU 使用率: 20.63 9	CPU 使用率: 20.63 % 查看 内存使用率: 48.57 %		% 查看	查看 存储使用率: 38% 查看				服务版本号: v1.8.2			
实例信息	子设备 子设备通信通道	场景联动 函数	计算 流数据分	淅 消息路由	日志服务	设置					
实例信息											编辑
实例名称	LinkloTEdge_Node		部署状态	部署成功			云监控状态				
CPU 使用率	20.63 % 查看		内存使用率	48.57 % 查看			存储使用率	38 % 查看			
实例进程	查看		服务版本号	- v1.8.2 创建时间		创建时间	2019-03-12 15:51:20				
修改时间	2019-03-13 17:11:00										
实例标签											

c. 在实例进程后,单击查看,可查看实例各个进程的信息及CPU、内存使用率。

边缘实例 > 实例洋情									
LinkloTEdge_N	lode 部署成功	查看部署日志	重置部署						
CPU 使用率: 13.32% 查看								服务版本号: v1.8.2	
实例信息	子设备 子设备					DIFF			
立例信息						ANGENT			信借
		进程 ID	进程名	CPU	内存				265144
实例名称	LinkloTEdge_No	280	task-dispatcher	5.17 %	0.61 %		云监控状态		
CPU 使用率	13.32 % 查看	431	node	1.46 %	3.41 %		存储使用率	38 % 查看	
实例进程	直看	446	fc-base	1.13 %	1.91 %		创建时间	2019-03-12 15:51:20	
修改时间	2019-03-13 17:11	204	logger	0.73 %	0.2 %				
实例标签		261	message-router	0.93 %	0.27 %				
网关						关闭			

11.您可以在实例详情页面,网关名称右侧的操作栏中单击远程连接或者远程文件管理,方便您远程 控制网关设备或对网关设备上的文件进行管理。详细说明请参见远程运维管理。

1.3 轻量版环境搭建

1.3.1 轻量版环境搭建

本文介绍基于x86 Linux系统的两种搭建Link IoT Edge轻量版本(Lite版本)运行环境的方法。 有二进制软件包和源码编译两种方式搭建Link IoT Edge轻量版本运行环境。

项目	二进制软件包	源码编译
CPU平台	ARM v7/v8、x86	ARM、MIPS、PPC
操作系统	Linux、MacOS、Windows、 Android	Linux、MacOS、Windows、 Android、OpenWRT
Flash资源	8 MB	Linux(< 1 MB)、Windows(< 200 MB)
内存资源	< 1 MB	< 1 MB
优势	开箱即用,超低集成成本	源码开放,支持裁剪,支持多平台
劣势	目前支持的平台较少,且不支持自定 义裁剪,Flash占用较多	可能碰到一些交叉编译问题,需要自 行解决

- · Lite版源码请参考GitHub源码库。
- · Lite版本上云通道依赖物联网平台SDK, 需要您参考这里自行移植。
- ·Windows版本支持图形界面,您可以自行下载。

二进制软件包方式

- 1. 登录您的x86 Linux系统机器。
- 2. 执行如下命令,将Link IoT Edge轻量版本软件包下载到/tmp目录。

```
cd /tmp/
curl -0 http://iotedge-packet-daily-build.oss-cn-hangzhou.aliyuncs.
com/x86-64-linux-gnu/iot-edge-lite-x86-64.tar.gz
```

▋ 说明:

```
以上命令中使用的下载软件包地址为x86 Linux系统v1.8.1版本软件包地址,其余系统的其他版本LE Lite软件包下载地址请见发布历史。
```

3. 执行如下命令,查看软件包是否已下载成功。

```
ls -alh iot-edge-lite-x86-64.tar.gz
```

系统显示类似如下信息,表示软件包下载成功。

```
-rw-rw-r-- 1 test test 7.4M 1月 10 09:58 iot-edge-lite-x86-64.tar.gz
```

4. 执行如下命令, 解压软件包到根目录并确认解压结果。

```
sudo tar zxvf iot-edge-lite-x86-64.tar.gz -C /
```

ls -alh /linkedge

- 5. 根据如下步骤完成边缘实例和网关的创建。
 - a. 物联网平台控制台,选择边缘计算 > 边缘实例。
 - b. 创建一个边缘实例。
 - A. 单击新增实例,在弹出窗口中设置实例名称。
 - B. 在网关产品后单击新建网关产品,为实例创建网关。

物联网边缘计算中的网关,承载边缘计算能力,每个实例必须分配一个网关设备,并且该 网关设备同一时间只能被分配到一个边缘实例。

物联网平台	边缘实例 💿							
快速入门								
设备管理	实例列表							
规则引擎	实例 ∨ 请输入实例名称		×					
数据分析	实例名称	新增实例		实例状态	创建时间	操作		
边缘计算	444	★ 实例名称:		 未部署 	2018-12-13 10:52:57	查看 删除		
边缘实例				- 00000-000-	2010 12 20 22 50 45	atar atra		
驱动管理		• 网天产品: · · · · · · · · · · · · · · · · · · ·	1 AT2800 4 35 D	● 司内省(700-97)	2010-12-20 23:59:45	三個 研究		
开发服务	10000010-		✓ ensewsty on	• 部署成功	2018-12-21 10:39:08	查看 删除		
应用托管	test, propri	你应信息: +新增标签		• 部署失败	2018-09-21 20:26:51	查看 删除		
视频服务	10000			 未部署 	2018-11-27 16:34:50	查看 删除		
产品文档	1.010		職定取消	 未部署 	2018-11-16 17:29:21	查看 删除		

C. 在新建产品页面中,设置网关产品参数,然后单击完成。

物联网边缘计算中的新建网关产品功能继承物联网平台设备管理 > 产品中,高级版产品的功能。已自动为您简化创建适合物联网边缘计算中使用的网关产品步骤。

新建产			
产品信	言息		
	* 产品名称		
	LinkloTEdge_Gateway		
	* 所属分类 💿		
	自定义品类	\sim	功能定义
更多信	言息		
	产品描述		
	请输入产品描述		
		0/100	
使用文	档		取消 完成

参数说明如下:

参数	说明
产品名称	为网关产品设置名称,用于后续的查询及识别网关产品。支持中 文、英文字母、数字和下划线,长度限制4~30,一个中文汉字算 2位。

参数	说明
所属分类	选择品类,为该产品定义 <u>物模型</u> 。 可选择为:
	 ・自定义品类:需根据实际需要,定义产品功能。 ・任一既有功能模板。
	选择任一物联网平台预定义的品类,快速完成产品的功能定 义。选择产品模板后,您可以在该模板基础上,编辑、修改、 新增功能。
	若您需要的网关没有特殊功能定义,建议您选择自定义品类。

参数	说明
产品描述	可输入文字,用来描述产品信息。字数限制为100。可以为空。

产品创建成功后,页面自动跳转回新增实例页面,并且网关产品参数下自动分配了刚创建 的网关产品。

D. 在新增实例页面,单击网关设备后的新建网关设备为网关产品添加设备。

物联网边缘计算中的新建网关设备功能继承物联网平台设备管理 > 设备的功能。

新增实例		
* 实例名称:		
LinkloTEdge_Node		
* 网关产品:		
LinkloTEdge_Gateway	\sim	新建网关产品
* 网关设备:		
请选择网关设备	~	新建网关设备
标签信息:		
+新增标签		
		确定 取消

E. 根据界面提示设置参数后,单击确认。

参数说明如下:

参数	描述
产品	系统已自动关联上一步创建的网关产品。

参数	描述
设备名称	为该网关设备命名。设备名称需保持产品内唯一。如不填写,系 统将自动生成。
	 说明: 设备名称支持大写字母[A-Z]、小写字母[a-z]、数字[0-9]和下划 线(_)。且不能以下划线开头或结尾。

- F. (可选)在新增实例页面,单击新增标签,可以设置实例标签。通过标签您可以更加有效 地归类及识别实例。您也可以不设置标签。
- c. 实例参数设置完成后,单击确定,至此您已创建边缘实例和网关。
- d. 在实例详情 > 实例信息页面, 网关页签下, 单击网关名称右侧的查看, 获取网关设备信息。

边缘尖例 → 实例注情											
LinkloTEdge_N	lode 未部署								查看部署日志	重置	部署
CPU 使用率 - 查看 内存使用率 - 查看 服务版本号: -											
实例信息	子设备 子设备通信通道	场景联动 函數	计算 流数据分	沂 消息路由	日志服务	设置					
实例信息	实例信息										
实例名称	LinkloTEdge_Node		部署状态	未部署			云监控状态				
CPU 使用率	- 查香		内存使用率	- 查看			存储使用率	- 查看			
实例进程	查看		服务版本号	-			创建时间	2019-03-12 1	5:51:20		
修改时间	2019-03-12 15:51:20										
实例标签											
网关										分音	阙关
网关名称		产品名称			状态	最后	后上线时间	操作			
gateway_01		LinkloTE	dge_Gateway		 未激活 	-		查看	除 远程连接 远移		

e. 系统跳转到网关设备的设备详情页面,在设备详情页面获取网关设备的设备证书(ProductKey、DeviceName、DeviceSecret),用于后续启动网关。

设备管理 > 设备评情									
gateway_01 未满适									
产品:LinkloTEdge_Gateway 查看 ProductKey: 复制 DeviceSecret:******* 显示									
设备信息 Topic	设备信息 Topic列表 运行状态 事件管理 服务调用 日志服务 子设备管理 子设备通道管理								
设备信息	设备信息								
产品名称	LinkloTEdge_Gateway	ProductKey	复制]	区域	-			
节点类型	网关	DeviceName	复制]	DeviceSecret	******* 显示			
当前状态	未激活	IP地址			固件版本	-			
添加时间	2019/01/11 17:06:58	激活时间			最后上线时间				
实时延迟 🎱	测试								
设备标签:无标签信息,	立即添加								

6. 在x86 Linux系统机器上执行以下命令, 启动Link IoT Edge。

```
sudo /linkedge/gateway/build/script/iot_gateway_start_lite.sh {
YourProductKey} {YourDeviceName} {YourDeviceSecret}
```

📋 说明:

请将{YourProductKey} {YourDeviceName} {YourDeviceSecret}替换为已在本地保

存的网关设备的设备证书信息。

例如,网关设备证书信息为ProductKey: a1*****gs、DeviceName: gateway、

DeviceSecret: 2Px*********H1S, 则执行的实际命令如下:

系统显示类似如下信息,表示启动成功。



7. 执行如下命令,查看网关的运行状态。

sudo /linkedge/gateway/build/script/iot_gateway_status_lite.sh

系统显示类似如下信息,表示已激活网关。

pi,	in.	@io	t:/linkedge/gateway/build/script\$ sudo /linkedge/gateway/build/script/iot_gateway_status_lite.sh
			Link IoT Edge Lite Service Status
[[+ +]	Service RemoteTerminalDaemon is active, pid is 6439 Service MQTTSample is active, pid is 6447

您也可以在物联网控制台,选择边缘计算 > 边缘实例,在已创建好的边缘实例右侧单击查看进 入实例详情页面,查看网关状态。

边缘实例 → 实例详情								
LinkloTEdge_N	Node 未部署						查看部署日志	雪置 部署
CPU 使用率: - 查看		内存使用率: - 查看		存储使用率:	- 查看		服务版本号: -	
实例信息	子设备 子设备通信通道	场景联动 函数)	十算 流数据分	析 消息路由 日志服	务设置			
实例信息								编辑
实例名称	LinkloTEdge_Node		部署状态	未部署		云监控状态		
CPU 使用率	- 查看		内存使用率	 查看 		存储使用率	- 查看	
实例进程	查看		服务版本号	-		创建时间	2019-03-12 15:51:20	
修改时间	2019-03-13 17:11:00							
实例标签								
								译
网关								分配网关议
网关名称		产品名称			状态	最后上线时间	操作	
gateway_01		LinkloTE	dge_Gateway		 	2019-03-14 15:23:44	查看 移除 远程连接 远程	这件管理

 您可以在实例详情页面,网关名称右侧的操作栏中单击远程连接或者远程文件管理,方便您远程 控制网关设备或对网关设备上的文件进行管理。详细说明请参见远程运维管理。

源码编译方试

源码编译方试主要包含如下两个部分:

- · 设备数据上云通道(C语言SDK)。详细操作请参考iotkit-embedded内容。
- · 设备远程运维通道。详细操作请参考iot_remote_access内容。

您需要根据需求,参考以上两个链接内容自行完成交叉编译生成二进制。您也可以参考开源Link Kit C-SDK内容,将设备数据上云通道和设备远程运维通道整合到一起编译。

📕 说明:

· Windows版本的下载和使用,请参考windows版本内容。

· 设备数据上云通道的其它开发语言操作,请参考SDK不同语言/平台功能汇总内容。

2 设备接入

2.1 设备接入简介

设备接入是Link IoT Edge提供的基础能力,设备接入模块在Link IoT Edge中称为驱动(driver)或设备接入驱动。所有连接到Link IoT Edge的设备都需要通过驱动实现接入。

设备接入驱动在Link IoT Edge框架位置如图所示。



设备接入驱动

一个完整的驱动(设备接入模块)由设备的连接管理、设备的数据(协议)转换和设备的数据交互 三个模块组成。

・连接管理

指设备与网关建立通信连接。Link IoT Edge不限制建立通信连接的协议,您可根据业务需求灵 活选择。

・数据转换

指设备接入驱动将获取到的终端设备数据转换为符合阿里云IoT物模型规范的数据格式,并上报 到阿里云IoT Cloud。阿里云物联网平台物模型规范请参考物模型。

・数据与命令处理

指驱动可以处理云端对于设备的操作请求,并完成对设备的服务调用和处理调用结果,最终将结 果返回到阿里云物联网平台。

设备接入驱动是Link IoT Edge中独立的服务模块,您可以根据业务协议需求开发自定义设备接入 驱动。下图展示了自定义驱动的功能和数据流向,并指出了开发一个自定义驱动需要做的开发工 作。



驱动使用介绍

Link IoT Edge针对不同用户需求提供了两种形态的驱动。

- · 一是由Link IoT Edge提供常用协议驱动(又称官方驱动),如Modbus、OPC UA驱动。
- ・二是提供设备接入驱动开发SDK,设备厂商或开发者使用驱动开发SDK开发私有协议驱动(又称自定义驱动)。



您可通过示例驱动章节提供的两个官方示例驱动快速熟悉驱动使用流程。

Link IoT Edge提供官方驱动,官方驱动是由阿里云提供的通信协议驱动,包括Modbus、OPC UA官方驱动。您可以参照_{官方驱动}章节内容,使用官网驱动做设备接入。

Link IoT Edge也提供设备接入SDK,您可以使用设备接入SDK,完成自定义协议驱动的开发。关于自定义驱动开发的详细介绍,请参考<u>驱动开发</u>。

2.2 示例驱动

本章介绍Link IoT Edge提供的Light和LightSensor两个示例驱动,通过这两个示例驱动帮助您 快速熟悉驱动的使用流程。

Light驱动是Link IoT Edge提供的一款智能灯的模拟驱动。Light驱动支持智能灯的开关操作,可以用于模拟灯设备。

LightSensor驱动是Link IoT Edge提供的一款光照度传感器的模拟驱动,光照强度数据按照一定周期重复,可以用于模拟光照度传感器设备。

前提条件

在开始操作本章内容前,请您确保已根据环境搭建内容完成了边缘实例的创建。
示例驱动使用方法

1. 添加子设备到边缘实例。

通过物联网平台控制台创建光照度传感器和智能灯设备,并将两个设备添加到边缘实例中作为子 设备,方便后续的控制管理。

a. 在边缘实例页面,选择已创建的LinkIoTEdge_Node实例,单击右侧的查看。

b. 在实例详情页面,选择子设备,单击分配子设备。

A. 在分配子设备页面中,单击新建子设备。

边缘实例 > 实例详情								
LinkloTEdge_Node				设备名称 搜索			刷新新	建子设备
CPU 使用率: - 查看	内存使用率: - 宣看	存储使	设备名称	产品名称	状态	最后上线时间	操作	
实例信息 子设备通信通道	i 场景联动 函数计算 流数	据分析 消息路	10.000	No. of Concession, Name	-81		分配	
子设备管理 请输入设备名称 搜索								
设备名称	产品名称							

B. 在新建子设备页面,单击新建产品,创建光照度传感器产品。

新建子设备

特别说明:deviceName可以为空,当为空时,阿里云会颁发全局唯一标识符作为deviceName。	
* 产品: 请选择产品 ~ 新建产品 设备名称:	
确认 耳	又消

C. 在创建产品页面设置参数后,单击确认。

 \times

新建产品	×
产品信息	
* 产品名称	
 光照度传感器 * 所属分类 ● 	
智能生活 / 家居安防 / 光照度传感器	功能定义
连网与数据	
接入网关协议	
自定义	0
更多信息	
产品描述	
请输入产品描述	
0/100	

使用文档

取消 完成

参数	描述
产品名称	此处设置为光照度传感器。
所属分类	选择品类,为该产品定义 _{物模型} 。此处选择智能生活 > 家居安防 > 光照度传感器。

参数	描述
接入网关协议	此处选择自定义。

D. 在新建子设备页面,产品自动分配已创建的光照度传感器产品,设备名称输 入LightSensor后单击确认,为光照度传感器添加设备。

 \times

特别说明:deviceName可以为空,当为空时,阿里云会颁发全局唯一称为deviceName。	示识符作
* 产品:	
光照度传感器	→ 新建产品
设备名称:	
LightSensor	
	确认 取迷

E. 参考创建光照度传感器产品及设备的步骤,创建客厅灯产品和名称为Light的设备。

其中,客厅灯产品参数设置如下:

参数	描述
产品名称	设置为客厅灯。

完成

参数	描述
所属分类	此处选择智能生活 > 电工照明 > 灯。

c. 在分配子设备页面,分别分配光照度传感器产品下的LightSensor设备和客厅灯产品下的Light设备到实例中。

分配子设备					×
光照度传感器 ~	设备名称	搜索		刷新 新建	子设备
设备名称	产品名称	状态	最后上线时间	操作	
LightSensor	光照度传感器	●未激活	-	分配	

d. 单击完成,至此您已为LinkIoTEdge_Node实例分配了LightSensor和Light两个子设备。

2. 为子设备分配驱动。

驱动用于将两个边缘实例的子设备接入到边缘网关中。

a. 在实例详情页面,选择子设备,单击已分配的子设备右侧的驱动配置,分别为两个子设备分 配LightSensor官方示例驱动和Light官方示例驱动。

边缘实例 > 实例详情 LinkloTEdge_Node (未認) CPU (使用語: 46.26 % 書葉 (た)	7001日道: 02.65 % 普通 770	Adduma: 32 % 表示		部要详情 筆置 部署
实例信息 子设备 子设备通信通道	场景联动 函数计算 流数据分析	消息路由 日志服务 设置		
子设备管理 研始入设备名称 搜索	驱动配置 + 选择驱动: LightSensor 查方示例	×		刷新 分配子设备
设备名称	自定义配置:	分留状态	最后上线时间	操作
LightSensor	请输入自定义配置	◎ 未激活		驱动配置 查看 移除
Light		确定 取消 未激活		驱动配置 查看 移除

参数	描述
选择驱动	为LightSensor设备选择LightSensor官方示例驱动。 为Light设备选择Light官方示例驱动。

参数	描述
自定义配置	此处无需设置。

b. 单击确定,至此您已成功为LightSensor设备和Light设备分别配置了驱动。

3. 配置消息路由。

为边缘实例分配消息路由,用于将子设备的数据上传到云端(IoT)。

- a. 在实例详情页面,选择消息路由,单击添加路由。
- b. 在添加消息路由页面中, 配置参数, 此处配置的参数是将子设备数据发送至云端。

边缘实例 > 实例详情												
LinkloTEdge_Node ***										部署详情	重置部署	l
CPU 使用率: - 查看	内存使用率:	- 宣晋		存储使用率	1: - 宣晋							l
实例信息 子设备 子设备通信通道	场景联动	函数计算	流数据分析	消息路由	日志服务	设置						
		添加消息路由					×					l
消息路由		* 消息来源:									添加路由	
消息来源	消息主题过	设备	~	全部产品	~			分配时间	操作			l
		* 消息主题过	滤:									l
		全部			~							l
		* 消息目标:										l
		IoT Hub			~							l
		*服务级别:										
		1			~	0						l
								2间进行流转,				ł
						确定取消	4					

按照界面提示,设置如下参数。

参数	描述
消息来源	此处选择设备,选择全部产品。
消息主题过滤	此处选择全部。
消息目标	此处选择IoT Hub。
服务级别	此处选择1。

c. 单击确定,至此您已为LinkIoTEdge_Node实例添加了消息路由。

4. 部署边缘实例。

在实例详情页面,单击部署后在弹出框中单击确认,部署边缘实例。

您可以通过单击部署详情来查看部署进度及结果。

边缘实例 > 实例详情							
LinkloTEdge_Node						○ 部署	中, 查看 重量 部署
CPU 使用率: - 查看 内存使用	率: • 查看	存储使用率:	· 宣晋				
实例信息 子设备 子设备通信通道 场景联	〕 部署详情			×			
子设备管理							刷新分配子设备
· 诸给 \ 心久夕行 · · · · · · · · · · · · · · · · · · ·	动作内容	状态 〇	详情	_			
MI-MIX/CLAC MI TEHTO	message routing	如果由	-				
设备名称	路由			2备状;	态 最后上线时间	操作	
LightSensor	Light 驱动	• 部署中	-	未激清	ŝ	驱动配置 查看 移除	
Light	: LightSensor 驱动	• 部署中	-	未激活	5	驱动配置 查看 移除	
				关闭			

5. 查看设备上线状态。

部署实例后,您可以在实例的子设备列表中看到Light设备和LightSensor设备状态已显示在

线。

边缘实例 > 实例详情					
LinkloTEdge_Node 🏁	成功				部署详情 重置 部署
CPU 使用率: 29.35 % 查看	内存使用率: 92.51 % 查看	存储使用率: 31% 查看			
实例信息 子设备 子设备通	信通道 场景联动 函数计算	流数据分析 消息路由 日志服务	设置		
子设备管理					刷新 分配子设备
请输入设备名称	搜索				
设备名称	产品名称	驱动名称	设备状态 最	后上线时间 操作	
Light	客厅灯	Light 首方示例	●在线	驱动配置 重	适 移除
LightSensor	光照度传感器	LightSensor 宣方示例	●在线	驱动配置 重	酒 移除

6. 查看设备运行状态。

在物联网平台,选择设备管理 > 设备,分别单击LightSensor和Light设备右侧操作栏中的查 看,进入设备详情页面,查看设备状态。您可以查看具体设备详情,选择运行状态,看到设备数 据已经发送至云端。

LightSensor运行状态

物联网平台	设备管理 > 设备评博
付達入口	LightSensor
	产品:光翻廣傳器碼 音看 Productikey: 雪茄 DeviceSecret: ******* 显示
产品	设备信息 Topic列表 运行状态 事件管理 服务调用 日志服务
i045	
分组	运行状态 设备数据上级均衡新期性值,点击 古雪勒斯市 可以西雪勒法 黑性的历史数据 文刘刚新 ① 表格 图表
规则引擎	光照度检测值 直看数据
数据分析	200 Lux
边缘计算	
开发服务 •	2013/01/14 10.44.32
应用管理	

Light设备状态

设备管理 > 设备详情			
Light 在线			
产品: 客厅灯 童看		ProductKey: 复制	DeviceSecret: ******* 显示
设备信息 Topic列表	运行状态 事件管理	服务调用 日志服务	
运行状态 设备数据上报的最新属性	值,点击"查看数据"可以查看指定属	性的历史数据	实动影新 👥 表格 能表
主灯开关	查看数据		
1			
2019/01/28 17:19:44			

至此,您已经完成示例驱动设备接入操作流程。如果您想进一步熟悉官方驱动和驱动开发,请参 考官方驱动和驱动开发章节。如果您想了解更多业务逻辑的开发和运行框架,可参考函数计算、场 景联动和流数据分析章节内容学习。

2.3 官方驱动

Link IoT Edge提供Modbus和OPC UA两种官方驱动,用于支持设备使用Modbus和OPC UA这两种工业领域应用广泛的通信协议。通过两个官方驱动,您可以简单快速地把Modbus设备或者OPC UA设备接入到Link IoT Edge。

Modbus驱动介绍

Modbus是常用的应用层数据通信协议,阿里云官方Modbus驱动(以下简称Modbus驱动)支持 Modbus-RTU和Modbus-TCP两种交互。

Modbus驱动可以直接连接Modbus从设备,详情见下图。



Modbus驱动也可以通过Modbus网关连接Modbus从设备,详情见下图。



Modbus驱动支持的功能有读取输入状态和输入寄存器,读/写线圈状态和保持寄存器。您可以直接 从控制台部署Modbus驱动到网关,也可以从控制台下载Modbus驱动代码进行修改,作为您的自 定义驱动使用。

Modbus驱动使用步骤如下:

- 1. 创建高级版产品,并选择接入网关协议为Modbus,具体创建产品步骤请参见创建产品(高级版)。
- 2. 为产品添加设备,具体添加设备步骤请参见单个创建设备和批量创建设备。
- 3. 为产品定义物模型,具体定义方法请参见新增物模型。
- 4. 完成上述设备配置后,需要配置设备与网关的交互方式。
 - a. 创建子设备通道,具体方法请参见子设备通道管理。
 - b. 添加子设备,具体方法请参见子设备管理。
- 5. 到边缘实例中分配相关设备和网关,选择Modbus驱动进行部署。

OPC UA驱动介绍

Link IoT Edge产品提供用于接入OPC UA设备的驱动(简称OPC UA驱动)。您可以直接从控制 台部署OPC UA驱动到网关,也可以从控制台下载OPC UA驱动代码进行修改,作为您的自定义驱 动使用。 OPC UA驱动和OPC UA设备的连接是通过OPC UA服务器关联的,OPC UA驱动通过操作OPC UA 服务器对外暴露的协议接口操作OPC UA设备,详情见下图。



OPC UA驱动使用步骤如下:

- 1. 创建高级版产品,并选择设备协议为OPC UA,具体创建产品步骤请参见创建产品(高级版)。
- 2. 为产品添加设备,具体添加设备步骤请参见单个创建设备和批量创建设备。
- 3. 为产品定义物模型,具体定义方法请参见新增物模型。
- 4. 完成上述设备配置后,需要配置设备与网关的交互方式,目前仅支持OPC UA交互。
 - a. 创建子设备通道,具体方法请参见子设备通道管理。
 - b. 添加子设备,具体方法请参见子设备管理。
- 5. 到边缘实例中分配相关设备和网关,选择OPC UA驱动进行部署。

2.4 驱动开发

Link IoT Edge提供设备接入SDK(Link Edge Device Access,简称LEDA)方便您开发自己的 驱动,SDK目前支持C、Node.js和Python三种版本的语言。本章节通过实现一个LED灯设备接 入Link IoT Edge,介绍自定义驱动开发流程。

设备接入SDK支持的三种开发语言详细内容,请参见C、Node.js和Python。

背景信息

本驱动示例实现如下场景:通过自定义驱动,实现一个LED灯设备接入Link IoT Edge,您可以通 过物联网平台控制台控制LED灯设备的开关操作,也可以主动查询灯设备的当前温度。



在Link IoT Edge官方驱动中并没有可以完成该需求的驱动,因此需要在Link IoT Edge上开发 LED设备驱动。

本文将基于C语言SDK,详细介绍LED设备驱动开发过程。

前提条件

在开始操作本章内容前,请您确保已根据基于Ubuntu 16.04搭建标准版环境内容完成边缘实例的创

建,为C语言SDK开发的设备驱动准备运行环境。

自定义驱动开发

- 1. 编码阶段
- 初始化驱动资源。通过调用leda_get_module_name接口,获取设备接入驱动名称(该名称为 在物联网平台控制台上传驱动时填写的驱动名称),驱动名称获取成功后,调用leda_init接 口完成资源初始化。

```
int main(int argc, char** argv)
ſ
                             = LE_SUCCESS;
    int
                 ret
    char*
                 module_name = NULL;
    pthread_t
                thread_id;
    log_i(LED_TAG_NAME, "demo startup\r\n");
    /* init driver */
    module_name = leda_get_module_name();
    if (NULL == module_name)
    {
        \log_e(\text{LED}_TAG_NAME, \text{"the driver no deploy or deploy failed\r"})
\n");
        return LE_ERROR_UNKNOWN;
    }
    if (LE_SUCCESS != (ret = leda_init(module_name, 5)))
    {
        log_e(LED_TAG_NAME, "leda_init failed\r\n");
        return ret;
    if (LE_SUCCESS != (ret = get_and_parse_deviceconfig(module_name
)))
    {
        log_e(LED_TAG_NAME, "parse device config failed\r\n");
        return ret;
    }
```

```
if (0 != pthread_create(&thread_id, NULL, thread_device_data,
NULL))
    {
        log_e(LED_TAG_NAME, "create thread failed\r\n");
        return LE_ERROR_UNKNOWN;
    ł
    prctl(PR_SET_NAME, "demo_led_main");
    while (1)
    {
        sleep(5);
    }
    /* exit driver */
    leda_exit();
    log_i(LED_TAG_NAME, "demo exit\r\n");
  return LE_SUCCESS;
}
```

解析驱动配置,完成设备上线。通过leda_get_config_size接口和leda_get_config接
 口获取设备配置,按照驱动配置格式解析,根据解析后的驱动配置连接设备并调
 用leda_register_and_online_by_device_name接口完成设备注册上线到阿里云物联网

平台。

驱动配置格式定义请参考驱动开发注意事项中驱动配置部分的内容。

```
static int get_and_parse_deviceconfig(const char* module_name)
Ł
    int
                                                     = LE_SUCCESS;
                                   ret
    int
                                   size
                                                     = 0;
    char
                                   *device_config = NULL;
                                                    = NULL;
                                   *productKey
    char
                                   *deviceName
                                                   = NULL;
    char
    cJSON
                                   *root
                                                    = NULL;
    cJSON
                                   *object
                                                    = NULL;
    cJSON
                                   *item
                                                    = NULL;
    cJSON
                                   *result
                                                    = NULL;
    leda_device_callback_t
                                   device_cb;
    device_handle_t
                                   dev_handle
                                                    = -1;
    /* get device config */
size = leda_get_config_size(module_name);
    if (size >0)
    {
        device_config = (char*)malloc(size);
        if (NULL == device_config)
         {
             log_e(LED_TAG_NAME, "allocate memory failed\r\n");
return LE_ERROR_INVAILD_PARAM;
        if (LE_SUCCESS != (ret = leda_get_config(module_name,
device_config, size)))
         {
             log_e(LED_TAG_NAME, "get device config failed\r\n");
             return ret;
        }
    }
    root = cJSON_Parse(device_config);
    if (NULL == root)
    {
         log_e(LED_TAG_NAME, "device config parser failed\r\n");
         return LE_ERROR_INVAILD_PARAM;
    }
```

```
object = cJSON_GetObjectItem(root, "deviceList");
   cJSON_ArrayForEach(item, object)
   {
       if (cJSON_Object == item->type)
       {
           /* parse device config */
           result = cJSON_GetObjectItem(item, "productKey");
           productKey = result->valuestring;
           result = cJSON_GetObjectItem(item, "deviceName");
           deviceName = result->valuestring;
           /* TODO: 解析设备自定义配置信息custom, 该字段内容来源在物联网平
台控制台的驱动配置项。由于该字段内容
              为字符串的json内容,所以在去除custom的value值后,需要再次进行
json解析操作。
           */
           /* register and online device */
           device_cb.get_properties_cb
                                                  = get_proper
ties_callback_cb;
           device_cb.set_properties_cb
                                                  = set_proper
ties_callback_cb;
           device_cb.call_service_cb
                                                  = call_servi
ce_callback_cb;
           device_cb.service_output_max_count = 5;
           dev_handle = leda_register_and_online_by_device_name(
productKey, deviceName, &device_cb, NULL);
           if (dev_handle < 0)
               log_e(LED_TAG_NAME, "product:%s device:%s register
failed\r\n", productKey, deviceName);
               continue;
           }
           g_dev_handle_list[g_dev_handle_count++] = dev_handle;
           log_i(LED_TAG_NAME, "product:%s device:%s register
success\r\n", productKey, deviceName);
    if (NULL != root)
    {
       cJSON_Delete(root);
   free(device_config);
   return LE SUCCESS;
}
```

3. 将收到的设备数据转换为阿里云IoT物模型格式并上报到物联网平台,调

用leda_report_properties接口上报设备属性数据,调用leda_report_event接口上报 设备事件。

说明:

此处使用虚拟设备,直接构造物模型格式的数据进行上报。

```
static void *thread_device_data(void *arg)
{
    int i = 0;
    prctl(PR_SET_NAME, "demo_led_data");
    while (1)
```

```
{
        for (i = 0; i < g_dev_handle_count; i++)</pre>
             /* report device properties */
             leda_device_data_t dev_proper_data[1] =
             {
                 {
                      .type = LEDA_TYPE_INT,
                      .key = {"temperature"},
                      .value = {"20"}
                 }
             };
             leda_report_properties(g_dev_handle_list[i], dev_proper
_data, 1);
             /* report device event */
             leda_device_data_t dev_event_data[1] =
             {
                 {
                      .type = LEDA_TYPE_INT,
.key = {"temperature"},
                      .value = {"50"}
                 }
             };
             leda_report_event(g_dev_handle_list[i], "high_tempe
rature", dev_event_data, 1);
        sleep(5);
    }
    pthread_exit(NULL);
    return NULL;
}
```

- 4. 处理云端服务请求。实现服务请求的三个回调函数如下所示:
 - · get接口:处理获取设备属性的请求。
 - · set接口:处理设置设备属性的请求。
 - · service接口:处理调用设备自定义方法的请求。

📃 说明:

本示例使

用get_properties_callback_cb、set_properties_callback_cb和call_service_callback_ 个回调函数实现服务请求。

```
log_i(LED_TAG_NAME, "%s\r\n", properties[i].value);
         }
    }
    return LE_SUCCESS;
}
static int set_properties_callback_cb(device_handle_t dev_handle,
                                   const leda_device_data_t properties
[],
                                   int properties_count,
                                   void *usr_data)
{
     int i = 0;
    for (i = 0; i < properties_count; i++)</pre>
     ł
/* 作为演示, 仅打印出设置属性信息 */
log_i(LED_TAG_NAME, "set_property type:%d %s: %s\r\n",
properties[i].type, properties[i].key, properties[i].value);
     ł
    return LE_SUCCESS;
}
static int call_service_callback_cb(device_handle_t dev_handle,
                                   const char *service_name,
const leda_device_data_t data[],
                                   int data_count,
                                   leda_device_data_t output_data[],
                                   void *usr_data)
{
    int i = 0;
     /* service_name为该驱动物模型自定义方法名称 */
    log_i(LED_TAG_NAME, "service_name: %s\r\n", service_name);
     /* 获取service_name方法的参数名称和值信息 */
    for (i = 0; i < data_count; i++)</pre>
         log_i(LED_TAG_NAME, "input_data %s: %s\r\n", data[i].key,
data[i].value);
     /* 此处错位演示并没有执行真正的自定义方法 */
    return LE_SUCCESS;
}
```

■ 说明:

本示例的完整工程源码请参见Github源码库LED设备驱动。

2. 调试与发布阶段

设备接入驱动编码完成后,进入调试阶段。调试阶段包括:编译打包、上传驱动、搭建环境、创建 物模型、创建设备、部署驱动、查看调试信息、调试数据交互和本地替换更新驱动。

1. 编译打包。

a. 下载示例代码。

git clone https://github.com/aliyun/linkedge-thing-access-sdk-c.
git

b. 进入linkedge-thing-access-sdk-c目录,完成示例编译。

```
cd linkedge-thing-access-sdk-c
make prepare && make && make install
```

c. 查看示例编译产物。

```
ls -l build/bin/demo/led/
```

系统显示类似如下图表示已完成编译并打包。

[root@5246ddd4b5cc link-iot-edge-access-sdk-c]#ls -l build/bin/demo/led/ total 112 -rw-r--r-- 1 root root 113702 Jan 4 12:03 demo_led.zip [root@5246ddd4b5cc link-iot-edge-access-sdk-c]# [root@5246ddd4b5cc link-iot-edge-access-sdk-c]#

2. 上传驱动。

- a. 在物联网平台控制台左侧导航栏中,单击边缘计算 > 驱动管理。
- b. 在驱动管理页面,单击右侧新增驱动。

快速入门	驱动管理列表				刷新新措驱动
设备管理	请输入驱动名称	搜索			
产品	驱动名称	驱动语言	驱动描述	修改时间	操作
设备 分组	LightSensor 官方示例	nodejs8	此驱动是一款光线传		查看 下载
边缘计算					
边缘实例	Light 官方示例	nodejs8	此驱动是一款智能灯	-	查看 下载
驱动管理 规则引擎	Modbus 官方	python3	modbus协议转换驱		查看 下载

c. 在新增驱动界面,根据提示设置驱动参数,上传已完成编译并打包的demo_led.zip文件。

新增驱动	×
* 语言类型: C	~
* 驱动名称: demo_driver	0
驱动描述: 驱动开发示例	
6/10	0
* 驱动文件: 重新上传 ②	
demo_led.zip上传成功 X	
	确定取消

d. 完成参数的设置并上传成功驱动文件后,单击确认。您可以在驱动管列表中看到刚刚创建的 驱动。

驱动管理列表				刷新新增驱动
请输入驱动名称	搜索			
驱动名称	驱动语言	驱动描述	修改时间	操作
LightSensor 官方示例	nodejs8	此驱动是一款光线传	-	查看 下载
Light 官方示例	nodejs8	此驱动是一款智能灯	-	查看 下载
Modbus 官方	python3	modbus协议转换驱	-	查看 下载
OPCUA 官方	python3	opcua协议转换驱动	-	查看 下载
demo_driver	С	驱动开发实例	2018-12-28 10:01:03	编辑 删除 上传 下载

- 3. 创建产品。
 - a. 参考创建产品(高级版), 创建一个产品。

物联网平台	产品管理		产品信息				
数据概览 快速入门	全部(4) 基础版(0) 高级版(i(4)	◆ 产品名称 led_demo				
设备管理	产品列表		 所属分类 ② 边缘计算/其他设备 	→ 功能定义			別新 创建产品
产品设备	请输入产品名称查询	搜索	节点类型				
分组	产品名称	产品版本	★ 节点类型		设备数	源加时间	操作
边缘计算 边缘实例	30/	高级版	 ● 设备 ● 设备 ● 是否接入网关 		1	2018-12-28 14:26:16	童香 删除
驱动管理	c net	高级版	 夏 否 连网与数据 		1	2018-12-28 10:08:28	査者 删除
规则引擎 数据分析		高级版	接入网关协议		1	2018-11-23 17:29:34	查看 删除
扩展服务 服务管理	upuun_gin /	高级版	国定又数据格式		1	2018-11-23 17:22:27	查看 删除
本日 大 裕			ICA 标准数据格式 (Alink JSON)	~ 0			

b. 参考新增物模型,为产品添加属性。

产品管理 > 产品详情	添加自定义功能 ×	
Ied_demo 應過版 ProductKey: 算制 产品信息 消息通信 ①施定义 服务端订算	 功能灵型: 属性 服务 事件 ● 功能名称: 	发布 设备投:1 前往管理
标准功能 ◎ 功能类型 功能名称	emperature emperature exg銀型: int32	号入物模型 查看物模型 <mark>液加功能</mark> 数据类型 数据定义 操作
	 取值范囲: -20 -20<!--</th--><th></th>	
	······ 摄氏度 / °C	
自定义功能 💿	读写类型: ● 读写 ○ 尺速	(Inclusional)
功能类型 功能名称	指还 评输入描述	数据类型 数据定义 操作

c. 参考新增物模型,为产品添加事件。

led_demo 高级版		发布
ProductKey:	DenduatSaard: ******* 🛢 🖛	设备数:1 前往管理
产品信息 消息通信 功能定义 服务端订阅	添加自定义功能	×
标准功能 ⑧	◆功能类型: 属性 服务 事件 ◎	导入物模型 童者物模型 落加功能
功能类型 功能名称	◆功能名称: high_temperature	数据类型 数据定义 操作
	★ 标识符: high_temperature	
	 ● 信息 ○ 告賢 ○ 故障 ◎ 	
	输出参数: +增加参数	
	描述 请输入描述	
自定义功能 💿	0/100	(and an

为事件增加参数。

编辑参数

 \times

* 参数名称:				
temperature				0
*标识符:				
temperature				0
* 数据类型:				
int32			\ \	~
* 取值范围:				
-20	~	80		
∗ 步长:				
1				
单位:				
摄氏度 / ℃			```	~

4. 添加设备。

参考单个创建设备,为产品添加设备。

设备管理 へ	设备列表 批次管理					
产品						
设备	设备列表					批量添加 泰加设备
分组	请输入DeviceName 请选择设备标签	添加设备 💿 🛛 👋				
边缘计算 へ 边缘实例	DeviceName	 特别说明:deviceName可以为空,当为空时,阿里云会颁发全局唯一标识符作为deviceName, 	节点类型	状态/启用状态 -	最后上线时间	操作
驱动管理		• /*B:	网关	- 未激活 🛑	-	查看 删除 子设备 (1) 配置下发
规则引擎		Led demo	设备	• 未激活 🚺	-	査看 删除
蚊掘分研 ✓ 扩展服务 ∧	- *****	led_demo	设备	• 离线 💽	2018/12/19 16:38:19	查看 删除
服务管理						

5. 部署驱动。

部署驱动依赖边缘实例,即您在<mark>前提条件</mark>中完成的环境搭建操作。

- a. 在物联网平台控制台,选择边缘计算 > 边缘实例,在已创建好的边缘实例右侧单击查看进 入实例详情页面。
- b. 在实例详情页面,选择子设备,单击分配子设备。
- c. 在分配子设备页面中,将之前创建的产品和设备分配至实例中,单击设备右侧的分配。

边缘实例 > 实例详情				分配子设备					×
demo 未部署 CPU 使用率: - 查看	内存使用率:-查考			led_demo ~	设备名称	搜索		刷新	新建子设备
实例信息 子设备	子设备通信通道 场景联动	函数计算 注	売数据 分	设备名称	产品名称	状态	最后上线时间	操作	
子设备管理				led_device	led_demo	●未激活	-	分配	
请输入设备名称	搜索								
设备名称	产品名称								
									完成

- d. 单击完成,完成设备的分配。
- e. 单击已分配的子设备右侧的驱动配置,为设备分配上传驱动步骤中完成的驱动。

边缘实例 > 实例详情				
demo 👬				部署详情重置部署
CPU 使用率: - 查看	内存使用率:- 查看	存储使用率: - 查看		
实例信息 子设备	子设备通信通道 场景联动 函数计算	流数据分析 消息路由 日志服务 设		
子设备管理	驱动配置	×		刷新 分配子设备
请输入设备名称	搜索 * 选择驱动:			
设备名称	了自定义配置:	stTetE36.44)	备状态 最后上线时间	操作
led_device	0	۲	未激活	驱动配置 查看 移除
		确定取消		■

- f. 在实例详情页面,选择消息路由,单击添加路由。
- g. 在添加消息路由页面中, 配置参数, 您可以通过设置消息路由查看具体配置方式。

实例信息	子设备	子设备通信通道	规则计算	函数计算	流数据分析	消息路由	日志服务	设置				
消息路由				添加消息路由					×			添加路由
消息来源			消息主题过	* X 1 * X	^{肖息} 来源: Q备 肖息主题过滤:	~	全部产品	~		分配时间	操作	
				*)	全部 ^{肖息目标:} oT Hub			~				
				* 1	₩ 予 级 別 :			~	0	2间进行流转,		
								确定	取消			

h. 在实例详情页面,右上角单击部署,部署边缘实例。部署成功后边缘实例名称后显示部署成功。

6. 查看调试信息。

实例部署成功后,可以查看设备连接状态和运行状态,这些信息可以在阿里云物联网平台查 看,也可以通过在边缘网关运行环境查看驱动运行日志。

云端查看:

a. 在实例详情页面,实例详情页签下查看设备连接状态。

demo 部署成功					部署详情 重置 部署
CPU 使用率: - 宣看	内存使用率: - 查看	存储使用率: - 查看			
实例信息 子设备 子设备通信通道	规则计算 函数计算 流發	如据分析 消息路由 日志服务	设置		
子设备管理					刷新 分配子设备
设备名称	产品名称	驱动名称	设备状态 最后上线时	间操作	
led_device	led_demo	demo_driver	 在线 2018-12-30 16:18:23 	驱动配量 查	香 移 除

b. 在物联网平台,选择设备管理>设备,单击led_device设备右侧操作栏中的查看,进入设备 详情页面,选择运行状态查看设备上报的数据。

led_device 在线		
产品: led_demo 查看	ProductKey: a复制	DeviceSecret:
设备信息 Topic列表 事件管理 服务调用 运行状	日志服务	
运行状态 设备数据上报的最新属性值,点击"查看数据"可以查看指定属性的历史题	R.	实时别新 〇〇 表格 图表
temperature		
20.00		
20°C		
更新的同:2018/12/28 15:17:52		
查看数据		

c. 在物联网平台,选择监控运维 > 日志服务,在日志服务页面,产品参数后选择led_demo产品,查看设备行为分析。

物联网平台	日志服务				
快速入门	产品: led_demo >>				
设备管理					
产品	日志服务 ②				
设备	设备行为分析物模型数据分析	上行消息分析 下行消息分析 消息内容查询			
分组	请输入DeviceName	11/189 🗸			授素 重置
规则引擎					
数据分析	时间	DeviceName	内容(全部)	状态以及原因分析	
边缘计算	2019/01/08 09:44:35	led_device	bnline, client/p=42.120.237.158	成功	
边缘实例					
驱动管理					共有1条 < 1 >
扩展服务					
监控运维					
实时监控					
在线调试					
日志服务					

d. 在日志服务页面,选择上行消息分析查看设备数据上报日志。

物联网平台	日志服务					
快速入门	产品: led_demo ~					
设备管理						
产品	日志服务 💿					
设备	设备行为分析 物模型数据分析	上行消息分析 下行消息分析 消息内容重	至询			
分组	请输入DeviceName	训输入MessageID 全部划	☆ ✓ 1小时 ✓			搜索 重置
规则引擎						
数据分析	∨ 时间	MessageID	DeviceName	内容(全部)	状态以及原因分析	
边缘计算	2019/01/08 10:02:00	1082457359965573120	led_device	Publish message to topic:/sys/u.utwizioxc	成功	
边缘实例	2019/01/08 10:02:00	1082457359999045633	led_device	Publish message to topic:/sys/a1UXxZxxc	成功	
驱动管理	2019/01/08 10:01:55	1082457339044421633	led_device	Publish message to topic:/sys/a1UXxZxxc	成功	
监控运维	2019/01/08 10:01:55	1082457339006640128	led_device	Publish message to topic:/sys/a1UXxZxxc	成功	
实时监控	2019/01/08 10:01:50	1082457318010023936	led_device	Publish message to topic:/sys/a1UXxZxxc	成功	
在线调试	2019/01/08 10:01:50	1082457318052069376	led_device	Publish message to topic:/sys/a1UXxZxxc	成功	
口心成务					-	

本地查看:

a. 通过./fctl show命令,可以查看到部署驱动的具体位置。

```
cd /linkedge/gateway/build/bin/
./fctl show
```

系统显示类似如下图:

ootebc60a124b6e6 bin]#./fctl show
Type: Driver
RunMode: LongLived
ServiceName: demo_driver
FunctionName: demo_driver
JobID: 297900
FunctionID: e8c2bb9c9dad413ca1eb913be1f7c93e
Environments: {"FC_DRIVER_CONFIG":"{\"driverConfig\":\"\",\"serverList\":[{\"customJson\":\"{}\",\"serverId\":\"ACE38D509C7D4C91868953D510F3B29D\"}],\"deviceList\":
custom\":\"{}\",\"productKey\":\'usin inc3Y\",\"deviceName\":\"led_device\",\"serverId\":\"ACE38D509C7D4C91868953D510F3B29D\"}],\"driverType\":1}"}
DriverName: demo_driver
Deploy: Success
Process: Running
Process PID: 23159
Process MemoryUsage: 18548 KB
ProcessStartTime: 2018-12-28 15:15:41.683994
ProcessLastInvokeTime: 1970-01-01 08:00:00.000000
GroupID: 142091
MemoryLimitSize: 1073741824 Bytes
TimeOut: 60
Category: 0
Version: 1545979754905
PackageType: zip
CodePath: /linkedge/run/downloaded-job/nodeengine/237900
ConfigFile: /Linkedge/run/downLoaded-job/nodeengine/297900/config.json
CheckSum: codeCheckSumDefault
Accounting accounting and
Regional Predictional default
Forelgnkey: aPtver#142091#e8c2009c9dad413cd1e09130e1f7C93e
Iniggeniype:
Cron:
Hanaler: main.hanaler

参数说明如下:

字段名称	字段解释
DriverName	驱动名称,该名称为在阿里云物联网平台上传驱动时填写名称。
CodePath	驱动部署到边缘网关中的位置路径。

字段名称	字段解释
Process PID	驱动启动后的进程ID。

b. 驱动在运行过程中会产生运行日志,通过查看运行日志可以了解驱动运行状态,每个自定义 驱动生成的日志文件统一放在/linkedge/run/logger/fc-base路径下。每个驱动日志文 件路径为/linkedge/run/logger/fc-base/xxxx/xxxx.txt。

〕 说明:

驱动日志文件路径中的xxxx为在上传驱动时填写的名称。

```
cd /linkedge/run/logger/fc-base/demo_driver
ls -l
```

详情如下:

		diamo an an					
_ [root@bc60a1	root@bc60a124b6e6_demo_driver]#pwd						
/linkedge/ru	/linkedge/run/logger/fc-base/demo_driver						
[root@bc60a1	24b6e6	demo_driv	er]#				
[root@bc60a1	24b6e6	demo_driv	er]#ls -l				
total 76							
lrwxrwxrwx	1 roo	ot root	. 70	Dec 28	15:15	log.INFO -> /linkedge/run/logger/fc-base/demo_driver/log_2018-12-28-15-15-41-1.txt	
-rw-rr	1 roo	ot root	4134	Dec 28	15:09	log_2018-12-28-14-57-08-1.txt	
-rw-rr	1 roo	ot root	3571	Dec 28	15:13	log_2018-12-28-15-10-34-1.txt	
-rw-rr	1 roo	ot root	15283	Dec 28	15:15	log_2018-12-28-15-14-36-1.txt	
-rw-rr	1 roo	ot root	37830	Dec 28	15:20	log_2018-12-28-15-15-41-1.txt	
[root@bc60a124b6e6 demo_driver]#							

驱动运行日志可以查看大部分驱动的运行状态,但是有时还需要查看其它日志配合了解当前 驱动的运行,比如设备上线异常时可以查看dimu日志,日志文件路径为/linkedge/run/ logger/dimu/xxxxx.txt。

```
logger/almu/xxxxx.txt。
```

```
cd /linkedge/run/logger/dimu
ls -l
```

详情如下:

 /linkedge/ru [root@bc60a1 [root@bc60a1	24b6e6 dim n/logger/d 24b6e6 dim 24b6e6 dim	u]# pwd imu u]# u]# ls -l	
	1 noot	noot	FE Das 28 15:15 Jag TNEO //inkadas/mum/laggan/dimu/lag 2018 12 28 15 15 40 1 tyt
LIMXLWXLWX	I POOL	root	55 Dec 28 15.15 \log_{10} -> / [[[Redge/Pun]/[logger/athur/log_2018-12-28-15-15-40-1.1x]]
-rw-rr	1 root	root	1459 Dec 28 11:11 log_2018-12-28-11-10-26-1.txt
-rw-rr	1 root	root	3793 Dec 28 14:47 log_2018-12-28-14-47-04-1.txt
-rw-rr	1 root	root	3638 Dec 28 14:55 log_2018-12-28-14-47-57-1.txt
-rw-rr	1 root	root	3899 Dec 28 15:10 log_2018-12-28-14-57-07-1.txt
-rw-rr	1 root	root	3807 Dec 28 15:13 log_2018-12-28-15-10-33-1.txt
-rw-rr	1 root	root	5475 Dec 28 15:15 log_2018-12-28-15-14-35-1.txt
-rw-rr	1 root	root	3204 Dec 28 15:15 log_2018-12-28-15-15-40-1.txt
[root@bc60a1	.24b6e6 dim	u]#	

设备上线成功后,如果设备数据上报有异常,则可以查看cloud_proxy日

志。cloud_proxy产生的运行日志文件路径格式和内容为/linkedge/run/logger/

cloud-proxy/xxxxx.txt。

cd /linkedge/run/logger/cloud-proxy

ls -ll

详情如下:

[root@bc60a1	24b6e6 clo	ud-proxy]#p	owd					
/linkedge/ru	linkedge/run/logger/cloud-proxy							
[root@bc60a1	24b6e6 clo	ud-proxy]#						
Froot@bc60a1	24b6e6 clo	ud-proxy]#1	.s -l					
total 1128								
lrwxrwxrwx	1 root	root	62	Dec 28	15:15	log.INFO -> /linkedge/run/logger/cloud-proxy/log_2018-12-28-15-15-40-1.txt		
-rw-rr	1 root	root	7299	Dec 28	11:11	log_2018-12-28-11-10-25-1.txt		
-rw-rr	1 root	root	30451	Dec 28	14:47	log_2018-12-28-14-47-04-1.txt		
-rw-rr	1 root	root	165951	Dec 28	14:55	log_2018-12-28-14-47-57-1.txt		
-rw-rr	1 root	root	472630	Dec 28	15:10	log_2018-12-28-14-57-07-1.txt		
-rw-rr	1 root	root	136951	Dec 28	15:13	log_2018-12-28-15-10-33-1.txt		
-rw-rr	1 root	root	76663	Dec 28	15:15	log_2018-12-28-15-14-35-1.txt		
-rw-rr	1 root	root	247473	Dec 28	15:21	log_2018-12-28-15-15-40-1.txt		
root@bc60a124b6e6 cloud-proxy]#								

7. 调试数据交互。

如果设备上线成功,则可以使用阿里云物联网平台在线调试功能调试驱动和设备,该功能页面还 可以实时查看该设备上报的数据信息,也可以触发对设备的方法请求调用。

a. 在物联网平台,选择监控运维 > 在线调试,在在线调试页面,选择调试产品和调试设备。

自动刷新 〕 刷新 清屏

b. 选择调试设备待调功能和方法,进行调试,并查看设备实时运行日志。

在线调试 调试设备: [ed_demo ·] [ed_device ·]		
编辑设备 例试真实设备 虚拟真实设备 功能: temperature(t ~) 方法: 设置 ~ /	实时日志 ● 检测到设备在线 类型 / 时间 设备上报数据 2019-01-08-10-15-05	自动規新 ①
2 "temperature": 0 3 }	设备上报数据 2019-01-08 10:19:05	vent/poperty/post\'\trunklagdi':1'082461699881001472\'\tid\'1397\'\'parma':1'(1'moperturu')'1'me':1546913945473\'vau w'20)}\' MessageMethod:''sys `````A''a', 'BizTpe': "PrOFERTY_EPCPT, "MessageResult:''200', 'DeviceNam': ''ed_device', 'MessageID': '971', 'uuid': '25dff291-2bd1-a995-6671-3c6e1ceedab'} { 'ActionType': 'upstream', 'MessageI': 'success,', 'TransformData': '[', 'LogTime': '1546913945', 'MessageParams': '[\'otid \'\'UkdBBb22Gaseg5YNL0010170300\'\'method':\'timg event.hgh_temperture.post\'\'topic'\'yes' ething/event/hgh_temperture/s0)}\', 'MessagBU': '872', 'uuid': '26dF2064066', 'udi' \'326JF207', 'bernam'; '('method':'15469139457' 8\'value'\'('temperature':50))', 'MessagBWthod': '/ysd', 'ucacucco'.Med', 'BizType': 'EVENT_REPORT, 'MessagBeResult': '20 o', 'DeviceName': 'Ied_device', ''MessagBU': '272', 'udi': '2632666-dH4-a684-a686-a68-a68-a68-a68-a68-a68-a68-a68-a6

8. 本地替换驱动。

如果在设备接入驱动调试过程中发现问题,需要修改驱动代码重新生成新的驱动进行调试,这时 仅需要将修改编译后的驱动直接在本地替换。

a. 找到驱动的位置,可以通过fctl命令进行查找,具体字段解释查看调试信息。

```
cd /linkedge/gateway/build/bin/
```

./fctl show

	[rootEbc60a124b6e6 bin]#./fctl show iC run in [Process] Mode.
ļ	Show Function [1]
	Configuration information:
	Type: Driver
	Executor: c
	RunMode: LongLived
	ServiceName: demo_driver
	FunctionName: demo_driver
	JobID: 297900
	FunctionID: e8c2bb9c9dad413ca1eb913be1f7c93e
	Environments: {"FC_DRIVER_CONFIG":"{\"driverConfig\":\"\", "serverList\":[{\"customJson\":\"{}\", "serverId\":\"ACE38D509C7D4C91868953D510F3B29D\"}],\"deviceList\":[
1	<pre>\"custom\":\"{}\",\"productKey\":\"uconcomec31\",\"deviceName\":\"led_device\",\"serverId\":\"ACE38D509C7D4C91868953D510F3B29D\"}],\"driverType\":1}"}</pre>
	DriverName: demo_driver
	Deploy: Success
	Process: Running
	Process PID: 23159
	Process MemoryUsage: 18548 KB
	ProcessStartTime: 2018-12-28 15:15:41.683994
	ProcessLastInvokeTime: 1970-01-01 08:00:00.000000
	GroupID: 142091
	MemoryLimitSize: 1073741824 Bytes
	TimeOut: 60
	(afegory: 0
	Veriston: 15459/9/54905
	Package / yee: ztp
	Coderath: / thrkedge/run/downloaded-job/hodeengine/29/900
	Contignite: Vitakedge/vun/admitodaed=job/nodeengine/29/300/contig.json
	CheckSum: codecheckSumberduit
	Regionaby regionable unit
	(roy-
	Handler: main.handler

- b. 根据CodePath找到驱动所在位置路径,完成替换。
- c. 根据Process PID得到驱动进程ID,使用kill -9 Pid命令重新启动驱动。

kill -9 Pid #Pid为驱动进程ID,即通过fctl命令查看得到的Process PID

驱动开发注意事项

驱动配置

驱动配置是在边缘实例中添加设备时产生的。驱动配置在边缘实例部署成功后,其内容以json格式 存储在Link IoT Edge配置中心。

驱动配置格式定义:

```
{
    "deviceList": [{
        "custom": "{\"ip\":\"192.168.1.1\",\"port\":12345}", //设备配
置, 支持自定义配置
        "deviceName": "dn", //设备名
        "productKey": "pk", //产品id
    }]
}
```

驱动配置参数说明:

配置名称	配置解释
deviceList	设备列表
deviceName	设备名称
productKey	产品标识
custom	自定义配置

驱动依赖

设备接入驱动根据协议和业务场景的不同,可能会涉及第三方库依赖。Link IoT Edge针对开发设备接入驱动所用不同开发语言,分别制定了第三方库依赖规则。

・C版本SDK:

C语言属于编译型语言,如果编译目标环境和运行环境不一致,则很可能导致不可运行。所以对 于使用设备接入C版本SDK开发驱动,需要保证开发编译目标环境和运行环境相同。

驱动包中包含驱动程序和依赖动态库。如果该驱动依赖于第三方库,则需要将动态库和驱动程序 一起打包生成最终驱动程序包。

· Node.js版本SDK:

使用设备接入SDK Node.js版本开发驱动时,若依赖第三方库,可在项目根目录使用如下命令 安装依赖。

npm install #####

· Python版本SDK:

使用设备接入SDK Python版本开发驱动时,若依赖第三方库,需要在驱动目录使用如下命 令,将第三方库与驱动安装在同级目录。

pip3 install -t . #####

驱动打包

基于Link IoT Edge提供的SDK开发驱动并完成调试后,需将产物打包为.zip包,并确保驱动Binary或index源文件在.zip包第一级目录。

每个版本SDK开发的驱动在打包时,有不同的打包规则:

· 基于C SDK开发的驱动

对于C语言开发的驱动,驱动包中包含驱动程序和驱动依赖的动态库。如果驱动程序包含依赖 库,则需要将依赖库放置指定的位置,即在驱动程序当前路径下的lib文件夹下。具体操作步骤 如下:

- 1. 规定驱动程序需命名为main。
- 2. 在main当前路径下创建lib文件夹。
- 3. 将main依赖的动态库全部拷贝到lib文件夹下。
- 4. 使用zip命令对当前路径下的main和lib进行压缩处理生成zip包。

```
zip -r your_driver_name.zip main lib
```

・基于Python SDK开发的驱动

驱动包文件中须包含index.py,并且在该文件中定义handler函数。

· 基于Node.js SDK开发的驱动

驱动包文件中须包含index.js,并且在该文件中定义handler函数。

附录

 · leda_register_and_online_by_device_name(const char *product_key, const char *device_name, leda_device_callback_t *device_cb, void *usr_data
)接口的device_cb作用是什么?

答:

device_cb是用于接收来自云端或其他边缘应用对设备的服务请求调用。分别为请求获取设备 属性、请求设置设备属性,请求调用设备服务。

typedef struct leda_device_callback { get_properties_callback
get_properties_cb; /* 设备属性获取回调 */ set_properties_callback
set_properties_cb; /* 设备属性设置回调 */ call_service_callback
call_service_cb; /* 设备服务回调 */ int service_output_max_count; /
* 设备服务回调结果数组最大长度 */ } leda_device_callback_t;

字段service_output_max_count表示什么意思?

答:

service_output_max_count是用来设置call_service_cb回调接口output_data数组的 最大长度。 leda_init(const char *module_name, int worker_thread_nums)接口的
 worker_thread_nums是用来做什么的?

答:

worker_thread_nums是用来初始驱动的线程池,初始的线程池用来处理来自云端或边缘应用 对设备的请求调用,具体大小可根据接入设备量进行合理设置。

leda_register_and_online_by_device_name和leda_register_and_on
 line_by_local_name有什么区别?

答:

by_device_name表示用此接口实现设备注册的设备名称device_name必须与在阿里云物联网 平台创建设备时的名称一样;

- by_local_name表示用此接口实现设备注册的设备名称device_name可使用自定义名称;注 意针对同一ProductKey下的设备注册,必须保证只能使用同一接口,即by_device_name和 by_local_name接口不允许在同一ProductKey下同时使用。
- ・设备正常上线且从设备所在驱动日志中看到设备数据正常上报,但是在阿里云物联网平台没有改 设备数据上报内容?

答:

确认是否在阿里云物联网平台配置消息路由,详细操作可参考设置消息路由内容。

3场景联动

3.1 什么是场景联动

场景联动是规则引擎中,一种开发自动化业务逻辑的可视化编程方式,您可以通过可视化的方式定 义设备之间联动规则,将规则部署至云端或者边缘端。

首先,您需在物联网平台控制台,规则引擎 > 场景联动页面中创建场景联动规则。每个场景联动规 则由触发条件(Trigger)、过滤条件(Condition)、执行动作(Action)三个部分组成。这种 规则模型称为 TCA 模型。

当触发条件触发这条场景联动规则后,系统通过判断过滤条件是否已满足,来决定是否执行规则中 定义的执行动作。如果满足过滤条件,则直接执行定义的执行动作;反之则不执行。

例如,您每天18:00下班回家。在炎热的夏天,您希望您到家后,家里的温度是凉爽、舒适的。您 可以创建一条规则,使空调设备自动化,实现这个需求。

空调自动化 描述:						编辑
场聚场跟联动 触发条件: ● 触发条件1 定时触发 ~	0 18***	٥				副除
+ 新增就发展件 过途条件: @ 过途条件1 设备过波 ~	温度传感器	V mydevice2	>) 室内温度	v .	√ 26	8059:
+ 新聞过途条件 + 执行动作: 	空调	V mydevice1	◇】能源开关	→ 开启-1	~	HMPs:
执行动作2 设备输出 >	空调	V mydevice1	◇□空陽温度	√ 26		删除
T 新HADATI JOJF						

参数设置如下图。

参数说明如下:

参数	描述
触发条件	定时为每天18:00。时间的cron表达式写作方法,请参见CRONTAB网
	页。
过滤条件	温度传感器探测到室内温度高于 26 摄氏度。

参数	描述
执行动作	空调开关设置为打开;空调控制温度设置为 26 摄氏度。

创建场景联动规则的更多设置说明,请参见云端管理场景联动。

3.2 云端场景联动

场景联动类型的规则是一种开发自动化业务逻辑的可视化编程方式,可以通过设备或时间维度的条件触发,经过过滤条件过滤,执行预定的业务逻辑,输出数据到设备或者其他规则,实现海量设备 根据场景的联动。

目前场景联动只支持高级版产品。

操作步骤

- 1. 单击物联网平台控制台左侧导航栏中的规则引擎 > 场景联动。
- 2. 单击创建规则。

物联网平台	规则引擎				
快速入门	数据流转 场景联动				
设备管理					
规则引擎	场景联动列表				创建规则
数据分析	请输入规则名称	搜索			
边缘计算	规则名称	规则描述	创建时间	状态	操作
开发服务 New					
应用托管			2019-01-25 16:16:49	• 未启动	管理 启动 日志 删除
监控运维			2018-04-19 20:09:58	●运行中	管理停止 触发日志
/*************************************					删除

3. 设置参数, 然后单击确认。

参数	描述
规则名称	设置具体规则的名称。支持中文、英文字母、数字、下划线和短划 线,长度限制 1 ~ 30个字符,中文字算两位字符。
规则描述	为规则添加描述,可以为空。

4. 完成场景联动的创建后在弹窗中单击前往编辑,管理配置场景联动。

您也可以在场景联动名称右侧单击管理,管理配置场景联动。

以空调设备自动化为例:在12:00至23:59之间,当温度传感器上报的室内温度低于16摄氏度时,空调设备执行控制室内温度为26摄氏度。

具体参数设置,请见下图。

规则引擎 > 场景联动详情				
我的家自动化场景				编辑
描述:				
触发条件: 💿				
触发条件1				删除
设备触发	~	温度传感器	mydevice2 ~	
室内温度	~	< v	16	
+新增触发条件				
过滤条件: 💿				
hand the fall of				
1辺28元1年1		10.00	00:50	删除
H31H312388	~	12.00	20.09	
+ 新增过滤条件				
*执行动作:				
执行动作1				an Re-
设备输出		· 空调 · · · · · · · · · · · · · · · · · · ·	mydevice1	and Poly
空阔温度	<u> </u>	26		
1. 新博林 行动作				
T THERE WITE				
仮 左 取消				
48/15				

单击页面右上角编辑,可更改场景联动规则名称,其余参数说明请见下表:

参数	描述
参 <u>奴</u> 触发条件	
	上图示例中,设置为设备触发:以温度传感器上报的室内温度低于16摄氏度作为触发条件。

参数	描述
过滤条件	过滤条件集。只有满足过滤条件的数据,才能触发执行动作。可设置 为设备过滤或时间过滤。可以为一个规则创建多个过滤条件,过滤条 件之间是和(and)关系。
	 · 设置为设备过滤,则需选择已创建的产品名称、设备名称、和设备 功能中的属性或事件。 · 设置为时间过滤,则需设置起始时间和结束时间。
	上图示例中,设置为时间过滤:时间在 12:00 至 23:59 之间,则触发 执行动作。
执行动作	需执行的动作。可设置为设备输出或规则输出。您可以设置多个动 作。某一动作执行失败时,不影响其他动作。
	 · 设置为设备输出,则需选择已创建的产品名称、设备名称、和设备 属性或服务(只有可写的属性或服务才能被设为执行动作)。当触 发条件和过滤条件均被满足时,执行已定义的设备属性或服务的相 关动作。
	 · 设置为规则输出,则需嵌套另外一个规则,即调用其他规则。被调用规则中的触发条件将被跳过,直接进行过滤条件检查。若过滤条件满足,则执行该规则中定义的执行动作。
	上图示例中,设置为设备输出:指定的空调设备执行控制温度为26摄 氏度。

运行场景联动

场景联动创建成功后,您可在规则引擎>场景联动页面中,启动此场景联动。

启动场景联动操作:

- 1. 单击物联网平台控制台左侧导航栏中规则引擎 > 场景联动。
- 2. 找到要启动的场景联动,单击右侧操作栏中的启动,使规则状态为运行中。

规则引擎 数据流转 场景联系	ih				
场景联动列表					创建规则
规则名称	规则描述		状态	创建时间	操作
			• 未启动	2018-12-24 16:16:45	管理 启动 日志 删除
我的家自动化场景			●运行中	2018-12-24 15:53:09	管理停止 触发日志 删除

当设备有数据上报,并且上报数据满足触发条件时,该场景联动便会在云端运行。

启动场景联动后:

- · 若场景联动在云端运行,则需要为场景联动中的设备配置消息路由,使得设备的属性和事件能够
 发送到IoT Hub(云端)。消息路由的配置请参考设置消息路由中。
- ・若场景联动在边缘端运行,则需要将场景联动分配到边缘实例中,分配方法请参考本文下方场景
 联动其他操作。

查看日志

您可以查看该场景联动的日志,并且可在详情中查看运行结果。



若某条场景联动即在云端运行又在边缘端运行,那么在物联网平台控制台规则引擎 > 场景联 动中,查看到的日志为云端运行日志和边缘端运行日志。

- 1. 单击物联网平台控制台左侧导航栏中规则引擎 > 场景联动。
- 2. 找到要查看日志的场景联动,单击右侧操作栏中的日志。

规则引擎			
数据流转 场展联动			
场景联动列表			创建规则
规则名称 规则描述	状态	创建时间	操作
我的家自动化场展 -	●运行中	2018-12-24 15:53:09	管理停止 触发日志 删除

3. 单击详情, 查看该条日志的详情信息。

规则计算 > 运行日志					
执行状态: 全部状态 🗸 201	8-06-25 21:11:1(- 2018-06-25 21:26:1⊗	搜索			刷新
执行时间	执行状态			操作	
2018-06-25 21:26:12	成功			① 详情	
2018-06-25 21:26:12	牛服			详情	
2018-06-25 21:26:12	日志详情		×	详情	
2018-06-25 21:26:12	执行设备	执行状态	失败原因	详情	
2018-06-25 21:26:12	a17y32gj7	r#74		详情	
2018-06-25 21:26:12	ditioning	102-101	success	详情	
2018-06-25 21:26:12			~	详情	
2018-06-25 21:26:12		(2) ^{关闭}	详情	

场景联动其他操作

- ·删除场景联动:
 - 1. 在规则引擎 > 场景联动页面中, 找到需要删除的场景联动规则名称。
 - 2. 单击规则名称右侧的删除, 删除该条场景联动规则。

· 触发场景规则:

在启动场景联动规则后, 方可显示触发操作按钮。

- 1. 在规则引擎 > 场景联动页面中, 找到需要已启动, 并需要触发的场景联动规则名称。
- 2. 单击规则名称右侧的触发,表示手动触发规则一次,即忽略已管理配置的触发条件,直接执行所有过滤条件和执行动作。
- · 在边缘实例节点中运行场景联动:

您需要根据如下步骤,将场景联动部署到实例中。

- 1. 左侧导航栏选择边缘计算 > 边缘实例,在环境搭建中完成的边缘实例右侧单击查看。
- 2. 在实例详情页面,选择场景联动,单击分配规则。
- 3. 在分配规则页面,单击待分配场景联动规则名称后的分配,然后单击完成。

边缘实例 → 实例详情	分配规则	×
LinkloTEdge_Node		
CPU 使用率: - 查看 内存使用率: - 查看	请输入规则名称 搜索	刷新 添加场景联动
实例信息 子设备 子设备通信通道 场景联动 函数计算 流数据分	规则名称	操作
	我的家自动化场景	分配
场景联动		分配
规则名称 分配时间	18	分配
		完成

4. 分配场景联动后,重新部署边缘实例。

3.3 边缘场景联动

部署至边缘实例的场景联动能够快速响应本地的消息,并且可在断网环境下正常运行。本文介绍 利用规则计算和设置消息路由,实现当光照度传感器(LightSensor设备)检测到光照度大于等 于500流明的时候,控制灯(Light设备)关闭,当光照度小于等于200流明的时候,控制灯打开。 如下图所示:



前提条件

- ·请您确保已根据<mark>环境搭建</mark>内容完成边缘实例的创建。
- ·请根据_{示例驱动}内容,使用官方示例驱动完成LightSensor和Light设备的创建,并将设备分配 到边缘实例中。

操作步骤
- 1. 创建场景联动规则。
 - a) 在边缘实例页面,选择已在<mark>前提条件</mark>中完成创建的LinkIoTEdge_Node实例,单击右侧的查 看。
 - b) 在实例详情页面, 选择场景联动, 单击分配规则。

您可以为实例分配已创建的场景联动规则,若没有可分配的规则,请继续参考下方的步骤创 建新的场景联动规则。

c) 在分配规则页面中,单击添加场景联动。系统自动跳转到规则引擎 > 场景联动中的创建场景 联动规则页面中。

· 边缘实例 > 实例详情		分配规则	×
LinkloTEdge_Node 部階成功		请输入规则名称 搜索	刷新 添加场景联动
CPU使用率,查看 内存使用率, 立例信息 子设备 子设备通信通道 场景联动	· 宣看 函数计算 法数	规则名称	操作
	E1971134 Didg	Sauge_Ine	分配
场景联动		Next, 1497, See	分配
规则名称	分配时间		
ToT	Edge可将简单规则从 i		
			关闭

- d) 参考云端场景联动,创建名为Rule_Light_Open的场景联动规则。
- e) 创建完成后,管理配置场景联动规则,具体参数设置如下图所示。

规则引擎 > 场景联动详情				
Rule_Light_Open 描述:				编辑
触发条件: 🕘				
触发条件1				删除
设备触发	~ 光照度传感器	 ✓ LightSensor 	×	
光照度检测值	<=	~ 200		
+ 新增触发条件				
过滤条件: 🔘				
过滤条件1				删除
请选择过滤条件	~			
+新增过滤条件				
* 执行动作:				
执行动作1				删除
设备输出	~ 客厅灯	 ✓ Light 	\checkmark	
主灯开关	→ 开启-1	~		
+ 新增执行动作				
保存 取消				

f) 配置场景联动后保存设置内容,并在规则引擎 > 场景联动页面,启动Rule_Light_Open规则,使状态变为运行中。

规则引擎 数据流转 场景联动				
场景联动列表 请输入规则名称	搜索			创成主规则
规则名称	规则描述	创建时间	状态	操作
Rule_Light_Open		2019-02-24 14:52:23	●运行中	管理停止触发日志 删除

- g) 参考创建Rule_Light_Open场景联动的步骤, 创建名为Rule_Light_Close的场景联动。
- h) 管理配置Rule_Light_Close场景联动,具体参数如下图所示。

规则引擎 > 场累联动详情 Rule_Light_Close 描述:				编辑
触发条件:● 触发条件1 设备触发	~ [光照度传感器	LightSensor	制除
ノuttuberiluogila + 新増競技条件 过途条件: 辺途条件1 通途条件1 通送条件		~ ~ ~		删除
 新增过滤条件 执行动作: 执行动作1 设备输出 主灯开关 	~ [~	客厅灯 、 关闭-0 、	light V	删除
+ 新增执行动作 保存 取消				

i) 配置完成场景联动后,启动Rule_Light_Close规则。

- 2. 分配场景联动到边缘实例。
 - a) 返回到实例详情 > 场景联动页面, 分配规则弹窗中。
 - b) 单击刷新, 刷新规则名称列表。
 - c)分别在已创建的Rule_Light_Open和Rule_Light_Close规则后单击分配,将场景联动规则 分配到边缘实例。然后单击完成。

边缘实例 > 实例详情	分配规则	×
LinkloTEdge_Node 舒霜成功		
CPU 使用率: - 查看 内存使用率: - 查看	请输入规则名称 搜索	刷新添加场景联动
实例信息 子设备 子设备通信通道 场景联动 函数计算 流数据分	规则名称	操作
	Rule_Light_Close	分配
场景联动	Rule_Light_Open	分配
规则名称 分配时间		
loT Edge可将简单规则从云 可以		
		完成

完成场景联动的添加后,便可在该实例的实例详情页面上,查看已分配的场景联动信息。

边缘实例 > 实例详情			
LinkloTEdge_Node 部署成功	部署详情 重置 部署		
CPU 使用率: - 查看 内存使用率: - 查看	存储使用率: - 查看		
实例信息 子设备 子设备通信通道 场景联动 函数	计算 流数据分析 消息路由 日志服务 设置		
场景联动			分配规则
规则名称	分配时间	操作	
Rule_Light_Open	2019-02-24 15:26:19	查看 日志 移除	
Rule_Light_Close	2019-02-24 15:26:18	查看 日志 移除	

d) 在分配规则页面, 分别单击Rule_Light_Open和Rule_Light_Close场景联动规则名称后的分配, 然后单击完成。

- 3. 部署边缘实例。
 - a) 在实例详情页面, 单击右上角部署后在弹出框中单击确定, 部署边缘实例。

边缘实例 > 实例详情	部署进程	×		
LinkloTEdge_Node 部署中 CPU 使用率: 66.28 % 查看	※ 部署开始中	0%	率: 31% 查看	查看部署日志 重置 部署 服务版本号: v18.2
实例信息 子设备 子设备通	部署进程		5服务 设置	
	 部署开始 			
场赏联动	· 即看表記 · 部署打包			SYACHURAN CONTRACTOR
规则名称	·部署分发	等待中		操作
Rule_Light_Open	 部署结束 	等待中		管理 日志 立即触发 移除
Rule_Light_Close		取消部署 关闭		管理 日志 立即触发 移除

b) 当部署状态显示为部署成功, 表示部署实例完成。您可以单击查看日志, 查看部署详情。

边缘实例 > 实例详情			
LinkloTEdge_Node 部署成功	部署进程	×	查看部署日志 重置 部署
CPU 使用率: 69.17 % 查看 实例信息 子设备 子设备	✓ 实例部署完成		服务版本号: v1.8.2 设置
场景联动	部署进程 • 部署开始	2019-02-28 21:18:01	<u>分散現明</u>
规则名称	 部署装配 ・部署打包 	 ✓ 完成 ○ 完成 	操作
Rule_Light_Open	 部署分发 ・部署结束 	✓ 完成 2019-02-28 21:18:07	管理日志 立即触发 移除
		度有日志 关闭	2010年1月11日日 一日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日

如果部署失败,在部署详情页面,可以单击部署失败的资源名称前的+号图标,进行查看具

体失败原因。

2019-02-28 19:16:04	•	部署详情	×
2019-02-28 19:01:39	•	222回かび5 4日か- 14日大 244年	
2019-02-28 19:01:20	•		
2019-02-28 18:12:38	•	Lid device_config 子设备能置 部督 ● 成切	
2019-02-28 18:10:58	•	☐ tsl_config 物模型配置 部署 ● 失败 101207	
2019-02-28 17:53:18	•	2019-02-27 16:06:28 [INFO] [tsl_config] [0] init success 2019-02-27 16:06:30 [INFO] [tsl_config] [0] assembly success 2019-02-27 16:06:32 [INFO] [tsl_config] [0] package success	
2019-02-28 17:52:55	•	2019-02-27 16:06:43 [ERROR] [tsl_config] [101207] Service not exist	
2019-02-28 15:51:24	•	□ driver_config 驱动配置 部署 ● 失败 101207	
2019-02-27 16:09:06	•	2019-02-27 16:06:28 [INFO] [driver_config] [0] init success 2019-02-27 16:06:30 [INFO] [driver_config] [0] assembly success 2019-02-27 16:06:32 [INFO] [driver_config] [0] package success	
2019-02-27 16:06:27	•	2019-02-27 16:06:43 [ERROR] [driver_config] [101207] Service not exist	
2019-02-27 16:02:08	•	① message_route_config 部署 ● 成功 消息路由配置	
共有302条 〈 上	一页	Instance_config 实例配置 部署 ●成功	
		*	长闭

部署实例成功后,已分配到该实例的场景联动规则将会在边缘端开始运行、监听及触发执 行。

4. 查看边缘实例场景联动日志。

📋 说明:

若某条场景联动既在云端运行又在边缘端运行,那么在物联网平台控制台规则引擎 > 场景联 动中,查看到的日志为云端运行日志和边缘端运行日志。

a) 在实例详情 > 场景联动页面,单击规则名称右侧操作栏中的日志。

边缘实例 > 实例详情			
LinkloTEdge_Node 部署成功			查看部署日志 重置 部署
CPU 使用率: 65 % 查看	内存使用率: 92.79 % 查看	存储使用率: 32% 查看	嚴务版本号: v18.2
实例信息 子设备 子设备通信通道	场景联动 函数计算 流数据分析	消息路由 日志服务 设置	
场景联动			分配规则
规则名称	分配时间		操作
Rule_Light_Open	2019-03-06 21:01:	32	管理 日志 立即触发 移除
Rule_Light_Close	2019-03-06 21:01:	29	管理 日志 立即触发 移除

b) 在运行日志页面,单击详情查看日志详情。

规则计算 > 运行日志				
执行状态: 全部状态 🗸 2018-06-25	21:11:1/ - 2018-06-25 21:26:1⑧			刷新
执行时间	执行状态		操作	
2018-06-25 21:26:12	成功		详情	
2018-06-25 21:26:12	牛股		详情	
2018-06-25 21:26:12	日志详情	×	详情	
2018-06-25 21:26:12	执行设备 执行状态	失败原因	详情	
2018-06-25 21:26:12	and the state	0100000	详情	
2018-06-25 21:26:12	10x+1)	Success	详情	
2018-06-25 21:26:12		+ 42	详情	
2018-06-25 21:26:12		大团	详情	

3.4 分配场景联动到边缘实例

边缘实例中若使用规则计算处理设备数据,则需要为边缘实例添加和部署场景联动。部署至边缘实 例的场景联动能够快速响应本地的消息,并且可在断网环境下正常运行。

为实例配置场景联动

为边缘实例部署场景联动之前,需先创建场景联动和边缘实例。创建方法,请参见云端管理场景联 动。

场景联动部署步骤:

- 1. 登录物联网平台控制台。
- 2. 单击左侧导航栏中边缘计算 > 边缘实例。
- 3. 找到要添加场景联动的实例,单击对应操作栏中的查看。

物联网平台	边缘实例 💿						
数据概览 快速入门	实例列表						刷新新增实例
设备管理	文例 ~ 前部	入买例/网天名称	搜索				
边缘计算	实例名称	网关名称	网关产品	实例状态	创建时间	修改时间	操作
驱动管理	myhome	linshi_gw	gateway_linkedge	● 未部署	2018-06-06 10:40:50	2018-10-19 10:51:51	查看 删除
规则引擎 应用管理							
数据分析							
扩展服务 产品文档							

4. 在实例详情页面,选择场景联动,然后单击分配规则或分配规则计算。

边缘实例 > 实例详情				
myhome 未部署				部臀详情重置都臀
CPU 使用率: - 查看	内存使用率: - 查看	存储使用率: - 查查		
实例信息 子设备 子设备通信通道	场景联动 函数计算 流数据分析	消息路由 日志服务 设置		
场景联动				分配规则
规则名称	分配时间		操作	

尚未添加任何场景联动 IoT Edge可將簡单规则人云扩展至边缘,簡单规则是一种可以在断网情况下持续运行的应用程序 可以帮助您实现本地及时处理的简单业务场景。 分离规则 了解语展联动

5. 在弹出的分配规则页面中,选择要分配给实例的规则,单击分配。分配成功后,单击完成。

行配规则		
请输入规则名称	搜索	添加场景联动
规则名称		操作
空调自动化		分配
10101010		分配
101001-0100		分配

完成场景联动添加后,便可在该实例的实例详情页面上,查看该实例的场景联动信息。

辺塚其例 〉 其例律情						
myhome 未部署				部署详情	重置	部臀
CPU 使用率: - 直吞	内存使用率: - 查容	存储使用率: - 查召				
实例信息 子设备 子设备通信通道	场景联动 函数计算 流数据分析	消息路由 日志服务 设置				
场景联动					分配	规则
规则名称	分配时间		操作			
空调自动化	2018-12-25 23:30:02		查看 日志 移除			



您可以单击场景联动右侧的日志,快速查询该场景联动的相关日志。

6. 部署实例。为实例分配场景联动后,单击页面右上角部署,并在弹出对话框中,单击确认。

您可以在本页面中查看部署状态和部署详情。

边缘实例 > 实例详情					
myhome 部署中				编辑 🔅 部署中 , 查看	重置部署
CPU 使用率: - 查看	内存使用率: - 查看	存储使用率: - 查看		,	
实例信息 子设备	子设备通信通道 场景联动 函数计算	流数据分析 消息路由 日	日志服务 设置		
场景联动					分配规则
规则名称	分配时间		操作		
空调自动化	2018-12-25 23:30:02		查看 日志	移除	

查看实例场景联动日志

在物联网平台控制台,查看已同步至云端的日志。



若某条场景联动既在云端运行又在边缘端运行,那么在物联网平台控制台规则引擎 > 场景联 动中,查看到的日志为云端运行日志和边缘端运行日志。

- 1. 登录物联网平台控制台。
- 2. 单击左侧导航栏中边缘计算 > 边缘实例。
- 3. 找到要查看日志的实例,单击对应操作栏中的查看按钮。
- 4. 在实例详情页面,选择场景联动。
- 5. 单击规则名称对应操作栏中日志。

边缘实例 > 💈	实例详情										
myhome	部署成功								部署详情	重置	部署
CPU 使用率: -	查看	内存	字使用率: - 查看		存储	諸使用率: - 查看					
实例信息	子设备	子设备通信通道	场景联动	函数计算	流数据分析	消息路由	日志服务	设置			
场景联动										分配	現则
规则名称				分配时间				操作			
空调自动化				2018-12-25 23:30:02				查看 日志 移除			

6. 在运行日志页面,单击详情查看日志详情。

规则计算 > 运行日志				
执行状态: 全部状态 🗸 2018-06-25 21:11:11 - 20	118-06-25 21:26:1⊗ 搜索			RDI SUF
执行时间	执行状态		操作	
2018-06-25 21:26:12	成功		① 详情	
2018-06-25 21:26:12	牛酸	~	详情	
2018-06-25 21:26:12	日志详情		详情	
2018-06-25 21:26:12	执行设备 执行状态	失败原因	详情	
2018-06-25 21:26:12	a17y32gj7 Wb/AirCon 5∜⊺th	SUCCASS	详情	
2018-06-25 21:26:12	ditioning	3000033	详情	
2018-06-25 21:26:12			详情	
2018-06-25 21:26:12			详情	

4 断网续传

物联网边缘计算提供断网续传服务。当本地断网或数据上传云端后超过15秒未接收到回复,上传云端的数据将在本地进行持久化。网络恢复后,历史数据将重新上传。使用断网续传功能,需要在物 联网平台上配置消息路由,同时您也可以自定义边缘端持久化设置。

物联网平台配置消息路由

边缘端的消息可以有多个来源,例如设备、函数计算、流数据分析等。物联网边缘计算支持为每个 消息来源配置独立的消息路由。若希望某个来源的消息被本地持久化、支持断网续传,则您需要为 该消息配置消息路由,且必须将消息目标设置为IoT Hub,服务级别设置为1(即QoS=1)。

本文以操作使用示例驱动接入设备文档内容为例,说明如何配置断网续传。

在操作使用示例驱动接入设备过程中, 配置消息路由这一步, 必须将消息目标设置为IoT Hub, 将服务级别设置为1。

	边缘实例 > 实例详情				
快速入门	LinkloTEdae Node				部署详情 軍置 部署
设备管理	CDIL 体田学· 杏手	添加消息路由	×		
规则引擎	0-0 12用率 且有		- 1		
数据分析	实例信息 子设备 子	* 消息来源:	务	设置	
边缘计算		设备 🗸 全部产品 🗸			
20000101	消息路由	* 消息主题过滤:			法加路由
22383年199	112.071	全部 ~			
驱动管理	消息来源		间		操作
开发服务 New		* 消息目标:			
应用托管					
监控运维		* 服务级别:			
stra-tiketst		1 🗸			
-X+10 mltr					
在线调试			- 1		
日志服务		确定 取	び消		

边缘端持久化配置项

边缘端持久化涉及文件存储目录、节点个数和节点存储空间三个配置项。

| ■ 说明:

整个持久化文件最大存储空间 = 持久化节点个数 x 持久化节点大小。

当整个持久化文件占用存储空间 > 阈值, 最久的数据持久化节点将会被删除。

默认配置:

配置项	默认配置
文件存储目录	/linkedge/run/history_record
gw_history_record_storage_path	

配置项	默认配置
节点最大个数 gw_history_record_node_num	10个
节点最大存储空间 gw_history_record_node_size	104857600 (100 MB)

自定义配置:

配置项	操作
获取文件存储目录	 命令: /linkedge/gateway/build/bin/tool_config -g gw_history_record_storage_path 返回值: Set config success. /linkedge/storage表示获取配置项 成功,存储路径为/linkedge/storage Get config fail, code:100100表示持久化存储路径未配 置,将使用默认路径/linkedge/run/history_record
设置文件存储目录	 命令: /linkedge/gateway/build/bin/tool_config -s gw_history_record_storage_path \$存储路径 返回值: Set config success表示设置成功。 Get config fail, code:100100表示设置失败,请检查Link IoT Edge服务状态。 边缘计算服务未对实际的存储路径进行限制,如果该路径下当前用 户无读写权限,会造成message-router服务异常。
获取节点个数	 命令: /linkedge/gateway/build/bin/tool_config -g gw_history_record_node_num 返回值: Set config success. 20表示获取配置项成功,节点个 数20。 Get config fail, code:100100表示节点个数未配置,将使 用默认值10。

配置项	操作
设置节点个数	 命令: /linkedge/gateway/build/bin/tool_config -s gw_history_record_node_num \$节点个数 返回值: Set config success表示设置成功。 Get config fail, code:100100表示设置失败,请检查Link IoT Edge服务状态。 节点个数可配范围[1-INT_MAX],如果配置越界,日志将会打 印the node num must be between 1 and INT32_MAX错 误。
获取节点最大存储空间	 命令: /linkedge/gateway/build/bin/tool_config -g gw_history_record_node_size 返回值: Set config success. 10485760表示获取配置项成功,节点 最大存储空间10485760 (10 MB)。 Get config fail, code:100100表示节点最大存储空间未配 置,将使用默认值104857600 (100 MB)。
设置节点最大存储空间	 命令: /linkedge/gateway/build/bin/tool_config -s gw_history_record_node_size \$节点最大存储空间 返回值: Set config success表示设置成功。 Get config fail, code:100100表示设置失败,请检查Link IoT Edge服务状态。 存储空间可配范围[1-LONGLONG_MAX],如果配置越界,日 志将会打印the node size must be between 1 and INT64_MAX错误。

功能验证

以下将使用_{光照传感器设备}为例,验证断网续传功能。

该实例的消息路由目标已设置为IoT Hub, 服务级别已设置为1;边缘端持久化配置项使用默认配置。

1. 查看历史数据空间占用

查看当前持久化文件占用空间,数据结构会占用一部分磁盘空间。

送明:

持久化文件目录空间 != 0并不意味着有历史数据存在。

#du /linkedge/run/history_record/ -d 0 -h
28.0K /linkedge/run/history_record/

2. 仿真网络异常状态

拔掉网线,间隔几秒时间执行命令,可以看到持久化文件目录空间占用持续递增。

#du /linkedge/run/history_record/ -d 0 -h
28.0K /linkedge/run/history_record/

#du /linkedge/run/history_record/ -d 0 -h
56.0K /linkedge/run/history_record/

#du /linkedge/run/history_record/ -d 0 -h

92.0K /linkedge/run/history_record/

3. 恢复网络

重新插上网线,网络恢复后,登录控制台查看光照传感器运行状态页面, 检查断网期间是否有 数据丢失。

- a. 登录控制台。
- b. 在左侧导航栏中,单击边缘计算 > 边缘实例。
- c. 选中之前创建的实例,单击查看。本示例中,查看LinkIoTEdge_Node实例。
- d. 选择子设备页面,单击查看之前创建的子设备。本示例中,查看LightSensor设备。

	边缘实例 > 实例详情					
快速入门	LinkloTEdae Nod	e 部署成功				部署详情 重置 部署
设备管理		内存使田室: - 奇看	存储使用率: - 1	看		
规则引擎			ca lational 23 1 -			
数据分析	实例信息 子设备	子设备通信通道 场景联动	函数计算 流数据分析 消息路日	日志服务	设置	
边缘计算						
边缘实例	子设备管理					刷新 分配子设备
驱动管理	请输入设备名称	搜索				
开发服务 New						
应用托管	设备名称	产品名称	驱动名称	设备状态	最后上线时间	操作
监控运维	LightSensor	光照度传感器	LightSensor 官方示例	 在线 	2019-02-23	驱动配置 查看 移除
产品文档					14.11.01	—

e. 在设备详情页,选择运行状态,打开实时刷新,查看设备上报的最新属性数据。

设备管理 > 设	设备管理 > 设备详情									
LightSenso	LightSensor 在线									
产品:光照度传	感器 查看			ProductKey	a13s 6vN	复制		DeviceSecret:******** 显示		
设备信息	Topic列表	运行状态	事件管理	服务调用	日志服务					
运行状态 设备	函数据上报的最新属	性值,点击"查看数1	图"可以查看指定属	計的历史数据					实时刷新 🦲	表格图表
光照度检测(直	查看数据								
300 Lux	x									
2019/02/23	14:22:27									

f. 单击查看数据,可以看到该属性的历史数据。您可以根据时间戳判断断网期间是否有数据。 本示例中,断网期间的数据,也上传到了云端。



常见问题

· 断网续传会对磁盘IO有影响吗?

答:断网期间Link IoT Edge会对上云数据进行本地持久化操作,具体视消息量大小。

· 网络恢复后,在控制台查看发现断网期间有数据未上报?

答:目前对历史数据上报限流为10条数据/秒。如果边缘端有大量的历史数据等待上报,完成这 一动作将会花费较长的时间。

·如何自定义持久化文件存储路径及最大存储空间?

- 答:可参考配置边缘端章节。
- ·断网期间,数据持久化文件占用空间达到设定的阈值,新的数据会保存吗?
 - 答: 会的。当历史数据持久化文件占用空间达到阈值, 最久的历史数据将会被删除。
- · 如何查看断网续传模块日志?
 - 答:
 - cd /linkedge/run/logger/message-router/
 - tail -f log.INFO

5 流数据分析

5.1 什么是边缘流数据分析

流数据分析是一种使用流的方法快速实时处理数据的计算方式。边缘计算中的流数据分析,继承了 物联网平台的流数据分析能力。

在物联网平台创建_{流数据分析任务},并将该任务下发到边缘端,通过边缘设备实时运行。运行结果 可以存储在边缘端也可以直接传输至云端。

边缘端流数据分析主要特点如下:

- ・运行在边缘端,不依赖网络,低时延。
- ・对数据进行采集、清洗、加工、聚合之后再上云、大大减少数据传输成本。
- ・提供和阿里云流式计算完全相同的SQL语法。您开发一次SQL,既可以在云上执行、也可以在边缘执行。
- ・提供了内建的字符串处理和时间、统计等各类计算函数。
- · 支持对消息乱序的处理。
- ・提供了流数据分析开发的图形化控制台。



5.2 分配流数据分析到边缘实例

本文介绍创建流数据分析任务、发布任务和分配流数据分析任务到边缘实例的整个步骤。

操作步骤

- 1. 登录物联网平台控制台。
- 2. 参考流数据分析,创建、设置并发布数据分析任务。

其中,执行任务需选择边缘端。

说明:图片以组件编排任务为例。		
创建任务		×
 * 任务类型: ● 组件编排 ○ SQL 		
* 任务名称: test_local_db * 任务描述:		
边缘端执行任务		
* 执行任务: 〇 云端		
		航定 取消

3. 单击左侧导航栏中边缘计算 > 边缘实例。

找到要分配流数据分析任务的实例,	单击对应操作栏中的查看。
------------------	--------------

物联网平台	边缘实例 🔵						
数据概览	and BURNING.						
快速入门	头例列表	(入实例)网关名称	搜索				刷新新增实例
设备管理							
边缘实例	实例名称	网关名称	网关产品	实例状态	创建时间	修改时间	操作
驱动管理	myhome	linshi_gw	gateway_linkedge	● 未部署	2018-06-06 10:40:50	2018-10-19 10:51:51	查看 删除
规则引擎							
应用管理							
数据分析							
扩展服务							
产品又档							

4. 在实例详情页面,选择流数据分析,然后单击分配任务或分配流数据分析任务。

边缘实例 > 实例能	自情											
myhome12	20 部署成功									部響详情	重置	部署
CPU 使用率: - 查看		内存使用率:	查看		存储使用率	1: - 查看						
实例信息 子	设备 子设备通信通道	道 场景联动	函数计算	流数据分析	消息路由	日志服务	设置					
流数据分析					_						分配	E\$
任务名称			任务状态				分配时间		操作			
尚未添加任何流数据分析任务												
			loT I	Edge可将流数据分析	可以帮助	边缘,流数据分 您实现本地及时 数据分析任务	析是一种可通过设备上报数据实现 处理的数据加工厂家。 了解流数据分析	持续營道式數据计算				● 咨询、辨议

5. 在弹出的分配任务页面中,选择要分配给实例的任务,单击分配。

本例中,流数据分析任务名称为test_local_db。

分配成功后,单击完	已成。			
分配任务				×
请输入任务名称	搜索		刷新	新增任务
任务名称				操作
test_local_db				分配
				完成

完成规则添加后,便可在该实例的实例详情页面上,查看该实例的流数据分析任务信息。

6. 部署实例。为实例分配任务后,单击页面右上角部署,并在弹出对话框中,单击确认。

您可以在本页面中查看部署状态和部署详情。

边缘实例 → 实例详情		
myhome (新寶中)	编辑 🔅 部署中, 查看 重置 部署	
最近修改:2018-10-19 10:51:51 实例标签:		
网关 子设备 规则计算 函数计算 🥻	流数据分析 消息路由 子设备通信通道 设置	
流數撮分析		分散任务
任务名称	分配时间	操作
test_local_db	2018-10-30 21.06.54	查看 移除



当任务在流数据分析页面撤回时,会通知到边缘计算,并在边缘计算中边缘实例 > 实例详情 > 流数据分析页签下,该任务的状态会显示为已撤回。

- ・如果不重新部署实例,将不会影响流数据任务继续处理设备数据。
- ·如果重新部署了实例,那么流数据任务将无法处理设备数据,需要重新分配已发布的流数据 任务到边缘实例,并部署实例。

5.3 边缘端流数据分析SQL

5.3.1 概述

您可以创建SQL类型的流数据分析任务,并在边缘端执行该任务。

边缘端SQL语法与云端Flink SQL语法完全一致,语法说明请参见Flink SQL。

在流数据分析中,数据存储用源表、维表、结果表来表示。但由于运行环境不一样,边缘端与云端 支持的数据存储大部分是不相同的。

类型	描述
源表	流式数据存储,驱动流数据分析的运行。每个流数据分析任务必须提供 至少一个流式数据存储。 边缘端支持的类型如下:
	 EdgeBus:边缘端消息总线(边缘端跨进程通信机制),EdgeBus 的数据作为输入源,会根据消息路由(管理控边缘端消息流向)的配置将数据输入到流作业中,目前支持设备数据流向流作业。

类型	描述
结果表	目的数据存储,用于存储流数据分析的计算结果。每个流数据分析任务 必须提供至少一个目的数据存储。 边缘端支持的类型如下: EdgeBusSink:边缘端消息总线,会把流作业的处理结果输出到 EdgeBus,可通过消息路由的配置决定流作业处理结果的具体流 向,目前支持流向函数计算、IoTHub。 RDS:关系型数据库(MySQL),会将流数据分析任务的处理结果 输出到阿里云关系型数据库中。 File:会将流数据分析任务的处理结果输出到边缘网关的文件中,主 要用于本地调试。
	关系型数据存储,必须显式定义数据变化周期,常用来做数据连接(Join)。 边缘端支持的类型如下: ·RDS:会把关系型数据库(MySQL)的数据当成维表输入。

5.3.2 创建源表

流数据分析的源表是指流式数据存储,流式数据存储驱动流数据分析的运行。因此,每个流数据分 析任务必须提供至少一个流式数据存储。

创建源表的语法如下:

边缘端消息总线(EdgeBus)

示例:

```
create table property (
    propertyName varchar,
    propertyValue varchar,
    productKey varchar,
    deviceName varchar,
    gmtCreate varchar,
    ts varchar,
    tstamp as to_timestamp (cast (ts as bigint)),
    WATERMARK wk FOR tstamp as withOffset (tstamp, 2000)
) with (
    type = 'edgebus',
    jsonParser = 'device_property'
);
```

WITH参数说明如下:

参数	是否必选	描述
type	是	定义源表类型为edgebus。
jsonParser	否	定义消息解析器,默认为default,取值如下:
		 device_property:属性消息,不符合属性消息格式 的消息将会被丢弃 device_event:事件消息,不符合事件消息格式的消息将会被丢弃 default:不使用解析器,直接使用json格式字符串,不丢弃任何消息
		不同的jsonParser对应不同的字段,详情请见下文字段说 明内容。

字段说明:

```
• jsonParser = 'device_property':
```

示例:

```
create table property (
    propertyName varchar,
    propertyValue varchar,
    productKey varchar,
    deviceName varchar,
    gmtCreate varchar,
    ts varchar,
    tstamp as to_timestamp (cast (ts as bigint)),
    WATERMARK wk FOR tstamp as withOffset (tstamp, 2000)
) with (
    type = 'edgebus',
    jsonParser = 'device_property'
);
```

字段名	类型	描述
productKey	String	产品的唯一标识ProductKey。
deviceName	String	设备名称,该产品下设备唯一标识。
propertyName	String	属性名。
propertyValue	String	属性值。
ts	String	消息产生时间。
gmtCreate	String	流数据分析接收到消息的时间。

• jsonParser = 'device_event':

示例:

```
create table event (
     eventCode varchar,
```

```
params varchar,
productKey varchar,
deviceName varchar,
gmtCreate varchar,
ts varchar,
tstamp as to_timestamp (cast (ts as bigint)),
WATERMARK wk FOR tstamp as withOffset (tstamp, 2000)
) with (
   type = 'edgebus',
   jsonParser = 'device_event'
);
```

字段名	类型	描述
productKey	String	产品的唯一标识ProductKey。
deviceName	String	设备名称,该产品下设备唯一标识。
eventCode	String	事件标识符。
params	String	事件参数。
ts	String	事件产生的时间。
gmtCreate	String	流数据分析收到事件消息的时间。

• jsonParser = 'default':

示例:

```
create table property (
    json varchar,
    jsonType varchar,
    gmtCreate as to_timestamp (
        cast (json_value (json, '$.gmtCreate') as bigint)
    ),
    deviceName as json_value (json, '$.deviceName'),
    productKey as json_value (json, '$.productKey'),
    ts as to_timestamp (
        cast (
            json_value (json, '$.items.temperature.time') as bigint
        )
    ),
    temperature as cast (
       json_value (json, '$.items.temperature.value') as int
    )
) with (
   type = 'edgebus'
);
```

字段名	描述
json	从消息路由获取到的消息内容,该消息内容的格式为字符串型json格 式,详细示例请参见本文下方Json格式内容。

字段名	描述
jsonType	从消息路由获取到的消息类型。
	 device_property device_event

Json格式:

- jsonType = device_event

5.3.3 创建结果表

流数据分析使用CREATE TABLE作为输出结果数据的格式定义,同时定义数据如何输出到目的数据存储中。

创建结果表的语法如下:

```
CREATE TABLE tableName
(columnName dataType [, columnName dataType ]*)
```

```
[ WITH (propertyName=propertyValue [, propertyName=propertyValue
]*) ];
```

边缘消息总线(EdgeBusSink)

示例:

```
create table edgehub_output (
    productKey varchar,
    deviceName varchar,
    temperature int,
    productTime timestamp,
    inputTime timestamp,
    delayTime bigint
) with (
    type = 'edgebus_sink',
    topic = '/sys/streamCompute/highTemperature'
);
```

WITH参数说明如下:

参数	是否必选	描述
type	是	定义结果表类型为edgebus_sink。
topic	是	定义消息主题,必须以斜线"/"开头,支持英文大小 写、数字、斜线"/"、下划线"_"和中划线"-",不超 过256字符。
		 注意: 若topic需要流转到IoTHub(云端),则需要在网关的 产品里定义topic,且该topic必须与SQL中的topic保持 一致。

关系型数据库(RDS)

目前仅支持MySQL数据存储类型,数据库中必须要有真实的表存在。

示例:

```
create table db (
productKey varchar,
deviceName varchar,
ts timestamp,
temperature int
) with (
type = 'rds',
url='jdbc:mysql://your_db_host:your_db_port/your_db_name',
tableName='your_table_name',
userName='config://local_db_username', -- local_db_username为在配置中存
储的用户名的key
password='config://local_db_password' -- local_db_password为在配置中存储
的密码的key
```

);

!) 注意:

在实际使用过程中需要把所有your_xxxxx参数值,替换为实际的数据库相关参数值,并根据如下步骤,在配置中心存入数据库的用户名密码。

- 1. 在物联网平台控制台左侧导航栏中,单击边缘计算 > 边缘实例。
- 在相应的边缘实例右侧单击查看,进入实例详情页面,在实例信息页签下的网关右侧单击远程
 连接,登录远程控制台,执行如下命令:

```
cd /linkedge/gateway/build/bin
./tool_config -s local_db_username root
./tool_config -s local_db_password xxxxxxx
```

WITH参数说明如下:

参数	描述
type	固定值rds,表示结果表的类型为RDS。
host	MySQL数据库(若为localhost,则为本地数据库)的host地址。
port	MySQL数据库端口号。
dbName	数据库名。
tableName	表名。
userName	数据库用户名。
password	数据库密码。

更多参数介绍,请参考创建云数据库结果表中RDS相关参数说明。

边缘文件系统(File)

示例:

```
create table print_sink (
    productKey varchar,
    deviceName varchar,
    eventCode varchar,
    ts timestamp,
    temperature int
) with (
    type = 'file'
    tag = 'prop_filter_b',
    filePath = '/linkedge/run/debug/prop_filter_b.txt'
);
```

with参数说明如下:

参数	是否必选	描述
type	是	定义结果表类型为file。
tag	否	为输出的消息设置标签。
		 · 若设置该参数,表示每条输出的消息前都附带该参数的 取值。 · 若无需为输出消息设置标签,则删除参数。
filePath	是	输出消息的目的文件路径,不同的流数据分析任务不可共 用一个文件路径。

示例表示,将消息输出到本地/linkedge/run/debug/prop_filter_b.txt文件中,并在每条 输出消息前带prop_filter_b标签。

5.3.4 创建维表

流数据分析没有专门为维表设计的DDL语法,只在标准的CREATE TABLE语法基础上增加一 行PERIOD FOR SYSTEM_TIME的声明。该声明定义了维表的变化周期,即表明维表是一张会变 化的表。

关系型数据库(RDS)

目前仅支持MySQL数据存储类型,数据库中必须要有真实的表存在。

示例:

```
CREATE TABLE rds_dim_table(
    id int,
    len int,
    content VARCHAR,
    PRIMARY KEY (id), -- 用作维表时,必须有声明的主键。
    PERIOD FOR SYSTEM_TIME--定义维表的变化周期,表明该表是一张会变化的表。
) with (
    type='rds',
    url='jdbc:mysql://your_db_host:your_db_port/your_db_name',
    tableName='your_table_name',
    userName='config://local_db_username', -- local_db_username为在配置
中存储的用户名的key
    password='config://local_db_password' -- local_db_password为在配置中
存储的密码的key
);
```

📕 说明:

· 声明一个维表时,必须指明主键。在进行维表JOIN的时候,JOIN ON的条件必须包含所有主键 的等值条件。RDS的主键可以定义为表的主键或唯一索引列。

- ・ 在实际使用过程中需要把所有your_xxxxx参数值, 替换为实际的数据库相关参数值, 并根据 如下步骤, 在配置中心存入数据库的用户名密码。
 - 1. 在物联网平台控制台左侧导航栏中,单击边缘计算 > 边缘实例。
 - 在相应的边缘实例右侧单击查看,进入实例详情页面,在实例信息页签下的网关右侧单击远 程连接,登录远程控制台,执行如下命令:

cd /linkedge/gateway/build/bin ./tool_config -s local_db_username root ./tool_config -s local_db_password xxxxxxx

WITH参数

与Flink SQL创建RDS维表的WITH参数一致,详情请参见创建云数据库#RDS和DRDS#维表。

5.4 使用示例

5.4.1 概述

流数据分析提供了连接边缘端消息总线的能力,您可以通过配置消息路由将设备数据传入到流数据 分析中,利用流数据分析方便地对设备数据进行实时计算,并将结果输出到RDS、本地文件、函数 计算或IoTHub中。

本使用示例将会从多个使用场景来讲述流数据分析的能力及使用方法。阅读本使用示例,需要您了 解SQL的基本知识。下表展示了每个示例的输入、输出,以及使用到的流数据分析算子,您可以根 据需求选择性阅读。

示例名称	输入	流数据分析算子	输出
高温报警	设备属性	过滤	边缘端文件
温度平均值计算	设备属性	过滤、聚合	边缘端文件、MySQL 数据库
连续高温报警	设备属性	复杂事件处理	边缘端文件
订单结算	设备事件、MySQL数 据库	聚合、数据连接(Join)	函数计算
人流量展示	设备事件	聚合	函数计算

5.4.2 高温报警

本文以高温报警(传感器温度>80度时,输出报警信息到本地文件中)的场景为例,详细说明如何 使用SQL进行流数据分析作业的编写以及流数据分析作业如何运作。



前提条件

在开始操作本章内容前,请您确保已根据环境搭建内容创建完成边缘实例。

操作步骤

1. 添加设备。

参考<mark>设备模拟器</mark>章节中使用设备模拟器驱动部分的内容,创建一个使用设备模拟器驱动的设备,并将设备分配到边缘实例中。



此处请先不要部署边缘实例,待完成其他操作后统一部署边缘实例。

2. 添加流数据分析作业。

- a)参考分配流数据分析到边缘实例,创建、设置并发布数据分析任务。
 - 其中,任务类型选择SQL,执行任务选择边缘端。

创建任务	×
* 任务类型:	
─ 组件编排 ● SQL	
*任务名称:	
高温报警	
*任务描述:	
当传感器温度大于80时输出报警信息到本地文件中	
* 执行任务:	
○ 云端	

b) 在任务列表中,已创建的SQL任务名称右侧单击查看,进入SQL编辑器。

复制如下SQL内容到SQL编辑器编辑框中。

温度>80时,输出报警信息到本地文件中	
定义数据源表,源表会驱动流数据分析作业的运行。	
每个流数据分析作业必须提供至少一个流式数据存储。	
create table property (
定义源表输入字段	
propertyName varchar, 属性名	
propertyValue varchar, 属性值	
productKey varchar,	
deviceName varchar,	
ts varchar, 数据产生的时间,由设备产生,即设备属性变化发生的时间	
gmtCreate varchar 流引擎接收到数据的时间,大于事件时间	
) with (
type = 'edgebus', 定义了源表类型, edgebus表示从边缘消息总线获取数	友
据 (json格式的字符串)	
jsonParser = 'device_property' 定义了json字符串解析器,	
device_property表示使用设备属性json解析器解析,不符合设备属性json格式的数	1
据会被丢弃	
);	

确定

取消

```
-- 定义数据结果表,用于存储流数据分析的计算结果。
-- 每个流数据分析作业必须提供至少一个结果表。
create table file (
   -- 定义结果表输出字段
   productKey varchar,
   deviceName varchar,
temperature double, -- 温度值
   ts timestamp, -- 数据产生的时间
   gmtCreate timestamp -- 流引擎接收到数据的时间
) with (
   type = 'file', -- 定义了结果表类型, edgebus表示将结果输出到文件中
   filePath = '/linkedge/run/debug/case01_high_temperature_alarm.
txt' -- 定义了输出的文件路径
);
-- insert语句定义具体的执行逻辑、源表收到数据后会触发这段逻辑的执行
insert into
   file -- 将结果输出到结果表(文件)
select
   productKey,
   deviceName,
   cast (propertyValue as double), -- propertyValue默认类型是字
符,温度值为浮点类型,需要做数据转化
to_timestamp (cast (ts as bigint)), -- 将数据产生的utc时间转为
timestamp
   to_timestamp (cast (gmtCreate as bigint)) -- 将流引擎接收到数据的
时间转为timestamp
from
   property -- 将源表接收到的设备属性数据拿出来处理
where
   propertyName = 'temperature' -- 通过属性名过滤出包含温度值的数据
   and cast (propertyValue as double) > 80; -- 塞选出温度值>80的数据
```

c) 保存任务并发布。

d) 参考分配流数据分析到边缘实例,将该SQL任务分配到边缘实例中。

3. 添加消息路由。

在边缘实例的实例详情页面,选择消息路由,将温度传感器的属性变化数据路由到高温报警的流 数据分析作业中。

添加消息路由	×
 * 消息来源: 设备 ∨ 温度传感器 ∨ temperatur ∨ 	
* 消息主题过滤: 属性 ~	
 * 消息目标: 流数据分析 ∨ 高温报警 ∨ 	



文档版本: 20190322

确定

取消

- 4. 部署边缘实例并查看设备运行结果。
 - a) 在边缘实例的实例详情页面,单击部署,将子设备、流数据分析作业及消息路由下发到边缘端。
 - b) 在实例详情 > 实例信息页面,单击网关页签下的远程连接,打开两个远程控制台,例如远程 控制台1和远程控制台2。远程控制台1用于改变温度传感器温度值,远程控制台2用于查看计 算结果。
 - c) 在远程控制台1, 进入/linkedge/gateway/build/bin目录, 执行如下命令:

```
./ds_ctrl property alWuxHrhojr temperatureSensor01 '{"temperature
":70}'
```

d) 在远程控制台2,执行tail -f /linkedge/run/debug/case01_high_temperat
 ure_alarm.txt命令,查看输出结果。

此时因为设备温度只达到了70,因此不会输出任何信息。

e) 在远程控制台1, 如下命令:

```
./ds_ctrl property a1WuxHrhojr temperatureSensor01 '{"temperature
":90}'
```

此时温度已经超过了80,可在远程控制台2可看到如下信息。

2019-01-24 18:09:48.490 -> a1WuxHrhojr,temperatureSensor01,90.0, 2019-01-24 18:09:48.456,2019-01-24 18:09:48.464

说明:

- · 流数据分析作业部署到边缘端后,会先解析 create table语句,即源表和结果表的定义。在本例中,源表定义解析完成之后会开始监听边缘消息总线,结果表定义解析后会尝试 打开文件。
- ・流数据分析作业会一直处于运行的状态,只要接收到边缘消息路由的数据(在本例中为温度 传感器的属性变化数据)就会立即执行具体的逻辑,即执行insert语句。

5.4.3 温度平均值计算

本文以计算五秒温度平均值为例,详细说明如何使用流数据分析预置的算子(统计、窗口等)实 现五秒温度平均值的计算并存入数据库中。本文操作中,步骤1~步骤4主要说明如何计算平均温 度,步骤5~步骤7主要说明如何将结果保存到数据库中,步骤8~步骤9主要说明如何根据需求修改 分组维度。步骤5~步骤9为可选步骤,可根据自身需求选择操作。



前提条件

在开始操作本章内容前,请您确保已根据环境搭建内容创建完成边缘实例。

背景信息

物联网数据有频率高、数据量大、数据变化小、数据价值较低等特点,将数据直接存储或全部上云的性价比非常低。为减少数据存储及传输成本,需要先将数据进行过滤、聚合之后再存储或上云。

例如,在工厂中有一台机器上安装了多个温度传感器,用于检测机器各部位的温度变化,现在需要把这些传感器的状态变化记录下来,以便进行数据分析及问题追溯。由于传感器每次产生的变化都 非常小,为节约成本,需要每五秒(真实场景中可能为一分钟甚至会更长,为了让读者更快看到计 算结果,本例采用5秒)求一次温度的平均值再存储到数据库中。

操作步骤

1. 添加设备。

参考<mark>设备模拟器章节中使用设备模拟器驱动部分的内容,创建一个使用设备模拟器驱动的温度传</mark> 感器产品和设备,并将设备分配到边缘实例中。



· 设备分配到边缘实例后,请先不要部署边缘实例,待完成其他操作后统一部署。

·若已操作过<u>高温报警</u>示例内容,并创建了温度传感器产品和设备,则无需重复创建。

2. 添加流数据分析任务。

a)参考分配流数据分析到边缘实例,创建、设置并发布数据分析任务。

其中,任务类型选择SQL,执行任务选择边缘端。

创建任务	\times
* 任务类型:	
○ 组件编排 ● SQL	
* 任务名称:	
平均温度	
* 任务描述:	
计算每五秒的温度平均值	
* 执行任务:	
○ 云端 ● 边缘端	
确定取	消

b) 在任务列表中,已创建的SQL任务名称右侧单击查看,进入SQL编辑器。

复制如下SQL内容到SQL编辑器编辑框中。

```
-- 计算5秒平均温度,并将结果输出到本地文件中
-- 定义数据源表
create table property(
    propertyName varchar,
    propertyValue varchar,
    productKey varchar,
    deviceName varchar,
    ts varchar, -- 属性变化的时间
    tstamp as to_timestamp (cast (ts as bigint)), -- ts为long型字符
串,单位为毫秒,需要转化为时间截格式
    WATERMARK wk FOR tstamp as withOffset (tstamp, 2000) -- 使用时
间窗口必须要在源表定义时声明Watermark。Watermark是插入到数据流中的一种特殊
的数据,Watermark上带了一个时间截,其含义是:在这个之后不会收到小于或等于该时
间截的数据。假设数据的乱序程度是1分钟,也就是说等1分钟之后,绝大部分迟到的数据
都到了,那么我们就可以定义watermark计算方式为偏移1分钟。2000表示偏移为2秒
) with (
    type = 'edgebus',
    jsonParser = 'device_property'
```

```
);
-- 定义数据结果表
create table output (
    productKey varchar,
    deviceName varchar,
    avg_temperature double,
    t_start timestamp,
    t_end timestamp
) with (
    type = 'file', -- 定义了结果表类型, file表示将结果输出到文件中
    filePath = '/linkedge/run/debug/case02_avg_temperature.txt'
 -- 定义了输出的文件路径
);
-- 计算平均温度并写入结果表中
insert into
    output
select
    productKey,
    deviceName,
    avg (temperature) as avg_temperature, -- 计算平均温度
    tumble_start (tstamp, interval '5' second), -- 时间窗口开始时
间(时间窗口长度为5秒)
    tumble_end (tstamp, interval '5' second) -- 时间窗口结束时间(时
间窗口长度为5秒)
from (
       select
           productKey,
           deviceName,
           cast (propertyValue as int) as temperature,
           tstamp
        from
           property
       where
           propertyName = 'temperature' -- 筛选出温度属性
    )
where temperature >= 0 and temperature <=100 -- 数据过滤,只计算合法
数据
group by
    tumble (tstamp, interval '5' second), -- 按时间窗口维度分组计
算(时间窗口长度为5秒)
   productKey, -- 按productKey维度分组计算
deviceName; -- 按deviceName维度分组计算
```

c) 保存任务并发布。

d) 参考分配流数据分析到边缘实例,将该SQL任务分配到边缘实例中。

3. 添加消息路由。

在边缘实例的实例详情页面,选择消息路由,将温度传感器的属性变化数据路由到平均温度的流 数据分析任务中。

添加消息路由	\times
 * 消息来源: 设备 ∨ 温度传感器 ∨ temperatur ∨ 	
* 消息主题过滤:	
全部	
* 消息目标: 流数据分析	
确定	取消
注意:若边缘实例中有关于将温度传感器到IoTHub的消息路由,请将其删除。	

- 4. 部署边缘实例并查看设备运行结果。
 - a) 在边缘实例的实例详情页面,单击部署,将子设备、流数据分析任务及消息路由下发到边缘端。
 - b) 在实例详情 > 实例信息页面,单击网关页签下的远程连接,打开两个远程控制台,例如远程 控制台1和远程控制台2。远程控制台1用于改变温度传感器温度值,远程控制台2用于查看计 算结果。
 - c) 在远程控制台2,执行tail -f /linkedge/run/debug/case01_high_temperat ure_alarm.txt命令,查看输出结果。

在有数据产生的情况下,计算结果每五秒产生一次。

d) 在远程控制台1,进入/linkedge/gateway/build/bin目录,多次执行如下命令改变温度 传感器状态,温度值可以在30、31两个数值间不断变化,5秒后将会在远程控制台2可看到如 下信息:

2019-01-30 15:46:43.045 -> a1WuxHrhojr,temperatureSensor01,30.5, 2019-01-30 15:46:35.0,2019-01-30 15:46:40.0

执行25秒后可看到如下信息:

```
2019-01-30 15:46:43.045 -> alWuxHrhojr,temperatureSensor01,30.5,
2019-01-30 15:46:35.0,2019-01-30 15:46:40.0
2019-01-30 15:46:49.783 -> alWuxHrhojr,temperatureSensor01,30.5,
2019-01-30 15:46:53.096 -> alWuxHrhojr,temperatureSensor01,30.
6666666666666666668,2019-01-30 15:46:45.0,2019-01-30 15:46:50.0
2019-01-30 15:46:58.119 -> alWuxHrhojr,temperatureSensor01,30.
33333333333332,2019-01-30 15:46:50.0,2019-01-30 15:46:55.0
2019-01-30 15:47:02.710 -> alWuxHrhojr,temperatureSensor01,30.
8333333333332,2019-01-30 15:46:55.0,2019-01-30 15:47:00.0
```

- 5. 在数据库中创建表。
 - a) (可选)新建一个RDS实例,或者在搭建边缘环境的机器上安装一个MySQL数据库(也可以 跨机器安装,但必须保证两台机器网络互通)。安装命令如下:


若已有MySQL数据库(本地数据库或RDS),可直接跳过本步骤。

sudo docker run --name mysql -p your_db_port:3306 -e
MYSQL_ROOT_PASSWORD=your_db_password -d mysql:5.6.35

其中, your_db_port为在宿主机(即启动mysql docker的机器)访问数据库的端口 号, your_db_password为数据库root账户的密码,请根据实际情况更改。例如,端口号 为1234,密码为xxxxxxx,则实例执行的命令为:

```
sudo docker run --name mysql -p 1234:3306 -e MYSQL_ROOT_PASSWORD=
xxxxxxxx -d mysql:5.6.35
```

- b)为了保证数据库用户名密码的安全性,您需要把数据库用户名密码保存在配置中心(会加密 后再存储)。按如下方式设置:
 - A. 在物联网平台控制台左侧导航栏中,单击边缘计算>边缘实例。
 - B. 在相应的边缘实例右侧单击查看,进入实例详情页面,在实例信息页签下的网关右侧单 击远程连接,登录远程控制台,执行如下命令:

```
cd /linkedge/gateway/build/bin
./tool_config -s local_db_username root
./tool_config -s local_db_password xxxxxxx
```

- c) 查看数据库host, 即安装MySQL的宿主机IP。在宿主机上执行ifconfig命令可看到如下结
 - 果:

```
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
  options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
  inet 127.0.0.1 netmask 0xff000000
 inet6 ::1 prefixlen 128
 inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
  nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
XHCO: flags=0<> mtu 0
XHC20: flags=0<> mtu 0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu
 1500
  ether f0:18:98:37:81:b4
  inet6 fe80::1c6e:7a87:1c27:724d%en0 prefixlen 64 secured scopeid
 0x6
  inet 30.43.83.169 netmask 0xfffff000 broadcast 30.43.95.255
  nd6 options=201<PERFORMNUD,DAD>
 media: autoselect
```

status: active

其中, *inet*参数后的30.43.83.169为宿主机对外IP, 即数据库host(用于流数据分析任 务中替换your_db_host)。

d) 下载mysql-client命令行工具或者mysql-client图形化工具(如Navicat、Sequel Pro

```
),进行MySQL数据库的访问,并创建iot_data数据库,用于在流数据分析任务中替换 your_db_name参数。
```

```
CREATE DATABASE iot_data;
USE iot_data;
```

e) 创建名为table_case_02的表,用于在流数据分析任务会中替换your_table_name参数。

```
CREATE TABLE table_case_02 (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    productKey varchar(255),
    deviceName varchar(255),
    avg_temperature double NOT NULL,
    t_start timestamp NOT NULL,
    t_end timestamp NOT NULL
);
```

6. 更新流数据分析任务,并将结果存入数据库。

在物联网平台控制台,选择数据分析 > 流数据分析,单击任务名称后的查看,进入SQL编辑器页 面撤回任务,并将任务内容改为如下SQL后保存并发布。

```
-- 计算5秒平均温度,并将结果输出到数据库中
-- 定义数据源表
create table property (
   propertyName varchar,
   propertyValue varchar,
   productKey varchar,
   deviceName varchar,
   ts varchar, -- 属性变化的时间
   tstamp as to_timestamp (cast (ts as bigint)), -- ts为long型字符
串, 单位为毫秒, 需要转化为时间戳格式
WATERMARK wk FOR tstamp as withOffset (tstamp, 2000) -- 使用时间窗
口必须要在源表定义时声明Watermark。Watermark是插入到数据流中的一种特殊的数据,
Watermark上带了一个时间戳,其含义是:在这个之后不会收到小于或等于该时间戳的数
据。假设数据的乱序程度是1分钟,也就是说等1分钟之后,绝大部分迟到的数据都到了,那
么我们就可以定义watermark计算方式为偏移1分钟。2000表示偏移为2秒
) with (
   type = 'edgebus',
   jsonParser = 'device_property'
);
-- 定义数据结果表[此处为变化部分,由之前的文件改成了数据库]
create table output (
   productKey varchar,
   deviceName varchar,
   avg_temperature double,
   t_start timestamp,
   t_end timestamp
) with (
```

```
type = 'rds',
    url='jdbc:mysql://30.43.83.169:1234/iot_data', -- 格式: 'jdbc:
mysql://your_db_host:your_db_port/your_db_name'
    tableName = 'table_case_02', -- your_table_name
userName = 'config://local_db_username', -- local_db_username为在
配置中存储的用户名的key
    password = 'config://local_db_password' -- local_db_password为在
配置中存储的密码的key
);
-- 计算平均温度并写入结果表中
insert into
   output
select
   productKey,
    deviceName,
    avg (temperature) as avg_temperature, -- 计算平均温度
    tumble_start (tstamp, interval '5' second), -- 时间窗口开始时间 (时
间窗口长度为5秒)
    tumble_end (tstamp, interval '5' second) -- 时间窗口结束时间 (时间窗
口长度为5秒)
from (
       select
           productKey,
           deviceName,
           cast (propertyValue as int) as temperature,
           tstamp
        from
            property
       where
           propertyName = 'temperature' -- 筛选出温度属性
    )
where temperature >= 0 and temperature <=100
group by
    tumble (tstamp, interval '5' second), -- 按时间窗口维度分组计算(时间
窗口长度为5秒)
    productKey, -- 按productKey维度分组计算
    deviceName; -- 按deviceName维度分组计算
```

- 7. 部署边缘实例并查看设备运行结果。
 - a) 在边缘实例的实例详情页面,单击部署,将子设备、流数据分析任务及消息路由下发到边缘端。
 - b) 在实例详情 > 实例信息页面,单击网关页签下的远程连接进入远程控制台。
 - c) 在远程控制台中,进入/linkedge/gateway/build/bin目录,多次执行如下命令,每次 执行时改变temperature参数后的值:

```
./ds_ctrl property a1WuxHrhojr temperatureSensor01 '{"temperature
":30}'
```

5秒后可在数据库中看到类似如下数据:

```
mysql> select * from table_case_02;
+----+-----+
| id | productKey | deviceName | avg_temperature |
t_start | t_end |
```

++	
+	·+
1 a1WuxHrhojr temperatureSensor01	31 2019-
01-50 10.05.50 2019-01-50 10.05.55	20 2010
2 alwuxHrnojr temperatureSensorol	30 2019-
01-30 16:05:35 2019-01-30 16:05:40	
3 alWuxHrhojr temperatureSensor01	31.6 2019-
01-30 16:05:40 2019-01-30 16:05:45	
++	
+	·+
3 rows in set (0.00 sec)	

- 8. 添加一个温度传感器。
 - a) 参考步骤1内容,再添加一个温度传感器设备temperatureSensor02。
 - b) 将温度传感器temperatureSensor02的属性变化数据,路由到平均温度的流数据分析任务中。
 - c) 参考步骤4部署边缘实例并查看设备运行状态。

需多次改变温度传感器temperatureSensor01和temperatureSensor02状态,每5秒可在数据库中看到两条结果(每个传感器产生一条结果)。

<pre>mysql> select * from table_case_02 order by t_start desc lin</pre>	nit 10;
++	
id productKey deviceName avg_temperature	
t_start t_end	
+++++++	
24 alWuxHrhoir temperatureSensor02 31	2019-
01-30 16:31:30 2019-01-30 16:31:35	2010
25 a1WuxHrhojr temperatureSensor01 31	2019-
01-30 16:31:30 2019-01-30 16:31:35	
23 alWuxHrhojr temperatureSensor01 30	2019-
01-30 10:31:25 2019-01-30 10:31:30 22 alWuyHrhoir temperatureSensor02 30	2019-
01-30 16:31:25 2019-01-30 16:31:30	2015
21 alWuxHrhojr temperatureSensor02 30.5	2019-
01-30 16:31:20 2019-01-30 16:31:25	
20 a1WuxHrhojr temperatureSensor01 30.5	2019-
01-30 16:31:20 2019-01-30 16:31:25	2010
19 a1wuxHrno]r temperatureSensor02 31	2019-
18 alWuxHrhoir temperatureSensor01 31	2019-
01-30 16:31:15 2019-01-30 16:31:20	
17 a1WuxHrhojr temperatureSensor01 31	2019-
01-30 16:31:10 2019-01-30 16:31:15	
16 a1WuxHrhojr temperatureSensor02 31	2019-
· · · · · · · · · · · · · · · · · · ·	
10 rows in set (0.00 sec)	

9. 求所有传感器的平均温度。

将本文中的SQL按时间窗口、productKey、deviceName三个维度进行分组计算,可以计算 出每个温度传感器5秒的平均温度。若在某些不需要关注每个温度的平均值,只需要知道机器温 度的平均值(即所有温度传感器的平均值)的场景中可以将流数据分析任务的SQL改为如下内

容(从分组维度中去掉productKey和deviceName):

```
-- 计算5秒平均温度,并将结果输出到数据库中
-- 定义数据源表
create table property (
   propertyName varchar,
   propertyValue varchar,
   productKey varchar,
   deviceName varchar,
   ts varchar, -- 属性变化的时间
   tstamp as to timestamp (cast (ts as bigint)), -- ts为long型字符
串, 单位为毫秒, 需要转化为时间戳格式
WATERMARK wk FOR tstamp as withOffset (tstamp, 2000) -- 使用时间窗
口必须要在源表定义时声明Watermark。Watermark是插入到数据流中的一种特殊的数据,
Watermark上带了一个时间戳,其含义是:在这个之后不会收到小于或等于该时间戳的数
据。假设数据的乱序程度是1分钟,也就是说等1分钟之后,绝大部分迟到的数据都到了,那
么我们就可以定义watermark计算方式为偏移1分钟。2000表示偏移为2秒
) with (
   type = 'edgebus',
   jsonParser = 'device_property'
);
-- 定义数据结果表[此处为变化部分,由之前的文件改成了数据库]
create table output (
   productKey varchar,
   deviceName varchar,
   avg_temperature double,
   t_start timestamp,
   t_end timestamp
) with (
   type = 'rds',
   url='jdbc:mysql://30.43.83.169:1234/iot_data', -- 格式: 'jdbc:
mysql://your_db_host:your_db_port/your_db_name'
   tableName = 'table_case_02', -- your_table_name
   userName = 'config://local_db_username', -- local_db_username为在
配置中存储的用户名的key
   password = 'config://local_db_password' -- local_db_password为在
配置中存储的密码的key
);
-- 计算平均温度并写入结果表中
insert into
   output
select
    11
   · , ,
   avg (temperature) as avg_temperature, -- 计算平均温度
   tumble_start (tstamp, interval '5' second), -- 时间窗口开始时间 (时
间窗口长度为5秒)
   tumble_end (tstamp, interval '5' second) -- 时间窗口结束时间 (时间窗
口长度为5秒)
from (
       select
           cast (propertyValue as int) as temperature,
           tstamp
       from
           property
       where
           propertyName = 'temperature' -- 筛选出温度属性
   )
```

```
where temperature >= 0 and temperature <=100
group by
    tumble (tstamp, interval '5' second); -- 按时间窗口维度分组计算(时间
窗口长度为5秒)
```

运行结果如下:

mysql> select * from table_case_02 order by t_start desc limit 5; | id | productKey | deviceName | avg_temperature | t_start | t_end ____+ -+----30.5 | 2019-01-30 16:40: 35 55 | 2019-01-30 16:41:00 | 34 31 | 2019-01-30 16:40: 50 | 2019-01-30 16:40:55 | | 33 | 31 | 2019-01-30 16:40: 40 | 2019-01-30 16:40:45 | 32 30.5 | 2019-01-30 16:40: 35 | 2019-01-30 16:40:40 | 30 | 2019-01-30 16:40: 31 05 | 2019-01-30 16:40:10 | 5 rows in set (0.00 sec)

5.4.4 连续高温报警

现实生活中,温度单次超出阈值并不会对设备造成影响,但连续多次超出阈值可能会引发机器故障 等风险。本文以连续高温报警(连续3次超过80度)为例,详细说明如何使用流数据分析复杂事件 处理的功能实现连续高温事件的判断,并将结果输出到文件中。



前提条件

在开始操作本章内容前,请您确保已根据环境搭建内容创建完成边缘实例。

操作步骤

1. 添加设备。

参考<mark>设备模拟器</mark>章节中使用设备模拟器驱动部分的内容,创建一个使用设备模拟器驱动的设 备,并将设备分配到边缘实例中。

📕 说明:

· 设备分配到边缘实例后,请先不要部署边缘实例,待完成其他操作后统一部署。

- ·若已操作过<u>高温报警</u>示例内容,并创建了温度传感器产品和设备,则无需重复创建。
- 2. 添加流数据分析任务。
 - a)参考分配流数据分析到边缘实例,创建、设置并发布连续高温报警的流数据分析任务。

其中,任务类型选择SQL,执行任务选择边缘端。

创建任务

 \times

- * 任务类型:
- ─ 组件编排 💿 SQL
- * 任务名称:

连续高温报警

* 任务描述:

连续高温报警

* 执行任务 :) 云端 • 边缘端

确定	取消
----	----

b) 在任务列表中,已创建的SQL任务名称右侧单击查看,进入SQL编辑器。

复制如下SQL内容到SQL编辑器编辑框中。若想了解更多复杂事件处理(CEP)的语法,请 参考Flink CEP。

-- 连续3次温度>80时,输出报警信息到本地文件中

```
create table property (
    propertyName varchar,
```

```
propertyValue varchar,
    productKey varchar,
    deviceName varchar,
    ts varchar,
    tstamp as to_timestamp (cast (ts as bigint)),
    WATERMARK wk FOR tstamp as withOffset (tstamp, 0)
) with (
    type = 'edgebus',
    jsonParser = 'device_property'
);
create table file (
    productKey varchar,
    deviceName varchar,
    temperatureList varchar,
    startTime timestamp,
    endTime timestamp
) with (
    type = 'file',
filePath = '/linkedge/run/debug/case05_continue_high_temperatu
re_alarm.txt'
);
insert into
file
select
    productKey,
    deviceName,
    temperatureList, -- 每次的温度示数
    startTime, -- 开始时间
    endTime -- 结束时间
from
    property
    MATCH_RECOGNIZE (
        PARTITION BY productKey, deviceName
        ORDER BY tstamp -- 在窗口内, 对事件时间进行排序
        MEASURES -- 定义如何根据匹配成功的输入事件构造输出事件
CONCAT(LAST(e.propertyValue, 2), '|', LAST(e.
propertyValue, 1), '|', e.propertyValue) as temperatureList,
            LAST(e.tstamp, 2) as startTime,
            e.tstamp as endTime
        ONE ROW PER MATCH -- 匹配成功输出一条
        AFTER MATCH SKIP TO NEXT ROW -- 匹配跳转到下一行后继续处理
        PATTERN (e{3}) -- e连续发生3次则输出一个事件
        DEFINE -- 定义在PATTERN中出现的patternVariable的具体含义
            e as e.propertyName = 'temperature' and cast (e.
propertyValue as double) > 80
    );
```

c) 保存任务并发布。

d) 参考分配流数据分析到边缘实例,将该SQL任务分配到边缘实例中。

确定

取消

3. 添加消息路由。

在边缘实例的实例详情页面,选择消息路由,将温度传感器的属性变化数据路由到连续高温报警 的流数据分析作业中。

添加消息路由	\times
 * 消息来源: 设备 ∨ 温度传感器 ∨ temperatur ∨ 	
* 消息主题过滤: 属性 ~	
 * 消息目标: 流数据分析 ∨ 连续高温报警 ∨ 	

- 4. 部署边缘实例并查看设备运行结果。
 - a) 在边缘实例的实例详情页面,单击部署,将子设备、流数据分析作业及消息路由下发到边缘端。
 - b) 在实例详情 > 实例信息页面,单击网关页签下的远程连接,打开两个远程控制台,例如远程 控制台1和远程控制台2。远程控制台1用于改变温度传感器温度值,远程控制台2用于查看计 算结果。
 - c) 在远程控制台1,进入/linkedge/gateway/build/bin目录,执行3次如下命令,每次更改temperature参数的值,分别改为80、81、82:

./ds_ctrl property a1WuxHrhojr temperatureSensor01 '{"temperature ":80}'

d) 在远程控制台2,执行tail -f /linkedge/run/debug/case05_continue_high
 _temperature_alarm.txt命令,查看输出结果。

此时因为设备温度只是连续两次超过80度,因此不会输出任何信息。

e) 在远程控制台1, 多次执行如下命令, 每次更改temperature参数的

值,按83~89、78~83每次增加一度的规律执行命令:

./ds_ctrl property a1WuxHrhojr temperatureSensor01 '{"temperature ":90}'

命令执行完成后,可在远程控制台2可看到如下信息。

2019-02-01 14:51:23.890 -> a1WuxHrhojr,temperatureSensor01,81|82| 83,2019-02-01 14:51:20.1,2019-02-01 14:51:23.723 2019-02-01 14:51:30.472 -> a1WuxHrhojr,temperatureSensor01,82|83| 84,2019-02-01 14:51:22.053,2019-02-01 14:51:30.401 2019-02-01 14:51:32.582 -> a1WuxHrhojr,temperatureSensor01,83|84| 85,2019-02-01 14:51:23.723,2019-02-01 14:51:32.509 2019-02-01 14:51:34.494 -> a1WuxHrhojr,temperatureSensor01,84|85| 86,2019-02-01 14:51:30.401,2019-02-01 14:51:34.398 2019-02-01 14:51:36.602 -> a1WuxHrhojr,temperatureSensor01,85|86| 87,2019-02-01 14:51:32.509,2019-02-01 14:51:36.559 2019-02-01 14:51:38.817 -> a1WuxHrhojr,temperatureSensor01,86|87| 88,2019-02-01 14:51:34.398,2019-02-01 14:51:38.802 2019-02-01 14:51:41.031 -> a1WuxHrhojr,temperatureSensor01,87|88| 89,2019-02-01 14:51:36.559,2019-02-01 14:51:40.967 2019-02-01 14:51:59.790 -> a1WuxHrhojr,temperatureSensor01,81|82| 83,2019-02-01 14:51:53.35,2019-02-01 14:51:59.68

5.4.5 订单结算

本文以订单结算为例,详细说明如何通过流数据分析的维表Join功能,将物联网设备产生的数据(识别器识别的商品及订单编号)和数据库中的数据(商品价格表)进行连接。



前提条件

在开始操作本章内容前,请您确保已根据<mark>环境搭建</mark>内容创建完成边缘实例。

背景信息

物联网设备的产生的数据大多都是一些动态的属性信息,很多时候我们需要将设备的动态属性和更 多维度的数据(如设备产地、型号,商品的颜色、价格等)进行关联,才能计算出我们想要的结 果。

例如,某无人小店有一台商品自助结算机,包含商品识别器及价格显示屏。识别器每识别到一个商 品会将商品及订单编号通过事件的方式上报,显示屏实时显示当前累积总价,而当前累积总价需要 将识别器上报的数据和数据库里的商品价格表关联后计算得出。

操作步骤

1. 添加设备。

参考设备模拟器章节中使用设备模拟器驱动部分的内容,完成如下步骤:

- ・添加商品识别器设备:
 - a. 新建商品识别器产品,在产品自定义功能处添加名为 recognizedGoods的事件,事件包含两个int型参数,商品编号goodsId和订单编号。
 - b. 为商品识别器产品添加设备,并分配进边缘实例中。
 - c. 将商品识别器设备的驱动设为设备模拟器驱动。
- ・添加显示器设备:
 - a. 新建显示器产品,在产品自定义功能处添加名为 content的字符类型属性。
 - b. 为显示器产品添加设备,并分配进边缘实例中。
 - c. 将显示器设备的驱动设为设备模拟器驱动。

送 说明:

将设备分配到边缘实例后,请先不要部署边缘实例,待完成其他操作后统一部署。

确定

取消

2. 添加流数据分析作业。

a)参考分配流数据分析到边缘实例,创建、设置并发布订单结算的流数据分析任务。

其中,任务类型选择SQL,执行任务选择边缘端。

创建任务	×
* 任务类型:	
○ 组件编排 ● SQL	
* 任务名称:	
订单结算	
* 任务描述:	
订单结算任务	
* 执行任务:	
○ 云端 ● 边缘端	

b) 在任务列表中,已创建的SQL任务名称右侧单击查看,进入SQL编辑器。

复制如下SQL内容到SQL编辑器编辑框中。

```
create table event (
    eventCode varchar,
    params varchar,
    ts varchar,
    tstamp as to_timestamp (cast (ts as bigint)),
    WATERMARK wk FOR tstamp as withOffset (tstamp, 2000)
) with (
    type = 'edgebus',
    jsonParser = 'device_event'
);
CREATE TABLE goods_info(
    id bigint,
    name varchar,
    price double,
    PRIMARY KEY(id),
    PERIOD FOR SYSTEM_TIME)
WITH(
    type= 'rds',
```

```
url='jdbc:mysql://30.43.83.73:1234/iot_data', -- 请将
your_db_host、your_db_port、your_db_name替换成真实值
userName='config://local_db_username', -- 通过./tool_config -s
 local_db_username xxx 设置
    password='config://local_db_password', -- 通过./tool_config -s
 local_db_password xxx 设置
    tableName='table_case_06_goods_info',
    cache='LRU',
    cacheSize='1000'
    cacheTTLMs='600000'
);
CREATE TABLE edgebus_sink (
    orderId bigint,
    orderPrice double
) WITH (
    type='edgebus_sink',
    topic = '/sys/streamCompute/orderPrice'
);
create view scan_goods as
select
    cast (json_value(params, '$.goodsId') as bigint) as goodsId,
cast (json_value(params, '$.orderId') as bigint) as orderId
from
    event
where
    eventCode = 'recognizedGoods';
insert into edgebus_sink
select
    orderId,
    sum(goods_info.price)
from
    scan_goods
join
    LATERAL goods_info FOR SYSTEM_TIME AS OF PROCTIME()
ON
    goods_info.id = scan_goods.goodsId
group by
    orderId;
```

c) 保存任务并发布。

d) 参考分配流数据分析到边缘实例,将该SQL任务分配到边缘实例中。

- 3. 添加函数。
 - a) 请参考函数计算使用流程内容创建服务、并在该服务下创建名为DisplayPrice的使用Nodejs8运行环境的函数。
 - b) 创建函数完成后,在线编辑代码,添加如下内容:

📕 说明:

请将your_displaydevice_productKey以及your_displaydevice_deviceName换成实际网关设备的ProductKey和DeviceName。

const leSdk = require('linkedge-core-sdk');

```
const iotData = new leSdk.IoTData();
module.exports.handler = function(event, context, callback) {
    const payload = JSON.parse(JSON.parse(event.toString()).
payload);
    const time = payload.time;
    const params = payload.params;
    iotData.setThingProperties({
         productKey: 'your_displaydevice_productKey',
deviceName: 'your_displaydevice_deviceName',
         payload: {
             content: `orderId: ${params.orderId}, orderPrice: ${
params.orderPrice}, time: ${time}
         }
    }, (err) => {
         if (err) {
             console.log(`display price err: ${err}`);
         }
    });
};
```

c) 在物联网平台控制台边缘实例的实例详情页面,选择函数计算,单击分配函数,将已创建的 函数分配到边缘实例。

边缘实例 > 实例详情				
demo 未部署				部著详情重置部署
CPU 使用率: - 查看	分配函数	×		
实例信息 子设备 子设备通信通	* 地域:			
函數计算	华东2(上海) ~ · · · · · · · · · · · · · · · · · ·			分配函数
区域服务名	DeviceScript ~	创建服务	分配时间	操作
	* 函数: DisplayPrice ~	创建函数		
	◆授权: AliyunIOTAccessingFCRole ~	创建RAM角色		
		分配 取消		
	回水が加加 loTEdge可将函数计算从云扩展至边缘,函数计算是可以帮助您实现本地	士19世纪安队17 异 一种可以在断网情况下按需运行或 设时处理的复杂业务场景。	者持续运行的应用程序,	
	分配函数	了解函数计算		

d) 配置函数。

参数	说明
运行模式	选择按需运行。
内存限制	设置为50 MB。
超时限制	设置为5秒。

4. 添加消息路由。

a) 在物联网平台控制台边缘实例的实例详情页面,选择消息路由,将商品识别器的数据路由到 订单结算的流数据分析任务中。

添加消息路由	×
 * 消息来源: 设备 ∨ 商品识别器 ∨ displayDevi ∨ 	
* 消息主题过滤: 事件 ~	
* 消息目标: 流数据分析 ~ 订单结算 ~	
	旋 取消

b) 将订单结算的流数据分析任务路由到DisplayPrice函数计算中。消息主题过滤设置为/sys/ streamCompute/orderPrice。 添加消息路由

 \times

* 消息来源:				
流数据分析 ~	订单结算	\sim		
* 消息主题过滤:				
/sys/streamCompute/orderPri	ce			
* 消息目标:				
函数计算 >	DisplayPrice	\sim		
			确定	取消

c)将显示器的属性变换数据路由到IoTHub中,方便在云端查看显示器设备的状态。

添加消息路由	×
 * 消息来源: 设备 ∨ 显示器 ∨ displayDevi ∨ 	
* 消息主题过滤:	
属性]
* 消息目标:	
loT Hub 🗸]
* 服务级别:	
1 ~	0
	确定 取消

5. 创建商品价格表。

若您已经操作过<u>温度平均值计算</u>示例并完成了5. <u>在数据库中创建表</u>这一步内容,则在本步骤中可 以直接跳到子步骤v操作。

a) (可选)新建一个RDS实例,或者在搭建边缘环境的机器上安装一个MySQL数据库(也可以 跨机器安装,但必须保证两台机器网络互通)。安装命令如下:



若已有MySQL数据库(本地数据库或RDS),可直接跳过本步骤。

sudo docker run --name mysql -p your_db_port:3306 -e
MYSQL_ROOT_PASSWORD=your_db_password -d mysql:5.6.35

其中, your_db_port为在宿主机(即启动mysql docker的机器)访问数据库的端口 号, your_db_password为数据库root账户的密码,请根据实际情况更改。例如,端口号 为1234,密码为xxxxxxx,则实例执行的命令为:

```
sudo docker run --name mysql -p 1234:3306 -e MYSQL_ROOT_PASSWORD=
xxxxxxxx -d mysql:5.6.35
```

- b)为了保证数据库用户名密码的安全性,您需要把数据库用户名密码保存在配置中心(会加密 后再存储)。按如下方式设置:
 - A. 在物联网平台控制台左侧导航栏中,单击边缘计算>边缘实例。
 - B. 在相应的边缘实例右侧单击查看,进入实例详情页面,在实例信息页签下的网关右侧单 击远程连接,登录远程控制台,执行如下命令:

```
cd /linkedge/gateway/build/bin
./tool_config -s local_db_username root
./tool_config -s local_db_password xxxxxxx
```

- c) 查看数据库host, 即安装MySQL的宿主机IP。在宿主机上执行ifconfig命令可看到如下结
 - 果:

```
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
  options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
  inet 127.0.0.1 netmask 0xff000000
 inet6 ::1 prefixlen 128
 inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
  nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
XHCO: flags=0<> mtu 0
XHC20: flags=0<> mtu 0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu
 1500
  ether f0:18:98:37:81:b4
 inet6 fe80::1c6e:7a87:1c27:724d%en0 prefixlen 64 secured scopeid
 0x6
  inet 30.43.83.169 netmask 0xfffff000 broadcast 30.43.95.255
  nd6 options=201<PERFORMNUD,DAD>
 media: autoselect
```

status: active

其中, *inet*参数后的30.43.83.169为宿主机对外IP, 即数据库host(用于流数据分析任 务中替换your_db_host)。

d) 下载mysql-client命令行工具或者mysql-client图形化工具(如Navicat、Sequel Pro

),进行MySQL数据库的访问,并创建iot_data数据库,用于在流数据分析任务中替换 your_db_name参数。

CREATE DATABASE iot_data; USE iot_data;

e) 创建商品价格表:

```
CREATE TABLE table_case_06_goods_info(
    id INT UNSIGNED NOT NULL PRIMARY KEY,
    name varchar(128),
    price double
);
INSERT INTO table_case_06_goods_info
(id, name, price)
VALUES
(1, 'Coke', 3.00),
(2, 'Sprite', 4.00),
(3, 'Juice', 5.00),
(4, 'Coffee', 6.00),
```

(5, 'Tea', 7.00);

- 6. 部署边缘实例并查看设备运行结果。
 - a) 在边缘实例的实例详情页面, 单击部署, 将子设备、流数据分析作业及消息路由下发到边缘端。
 - b) 在实例详情 > 实例信息页面, 单击网关页签下的远程连接, 登录远程控制台。
 - c) 进入/linkedge/gateway/build/bin目录,执行如下命令上报识别到的商品识别器事件:

```
./ds_ctrl event alivc15wcrn goodsRecognizer01 recognizedGoods '{"
orderId": 12, "goodsId": 1}'
```

d) 在物联网平台控制台,选择设备管理>设备,在相应设备名称右侧单击查看。

019-01-14及17公司・	※//방치 With 는 나와 변수 명수 이 특별 이 특		
设备管理 > 设	查看数据		×
displayDe	设备状态		
产品:显示器	1小时 🗸 2019-01-31 09:48:10	- 2019-01-31 10:48:10 曲	图表
设备信息	时间	原始值	
运行状态设备	2019/01/31 10:47:43	orderld: 12, orderPrice: 3, time: 1548902863231	
显示内容	2019/01/31 10:46:59	orderld: 165, orderPrice: 6, time: 1548902819054	
2011 J H		全部加载完毕	
orderl			
2019/01/31			
			关闭

在设备详情页面查看显示器设备的运行状态以及数据。

5.4.6 人流量展示

在物联网场景中,经常需要根据设备的状态分析结果及时作出响应。例如某会场需要实时统计最近 十秒人流量并立即更新显示屏上的数据。本文以人流量实时显示为例,详细说明如何使用流数据分 析进行人流量的统计,并将计算结果输出到函数计算中,在函数计算中更新显示屏内容。



前提条件

在开始操作本章内容前,请您确保已根据<mark>环境搭建</mark>内容创建完成边缘实例。

操作步骤

1. 添加设备。

参考设备模拟器章节中使用设备模拟器驱动部分的内容,完成如下步骤:

- ・添加闸口产品和设备:
 - a. 新建闸口产品,在产品自定义功能处添加识别符为 peopel Passed的事件。

其中,事件包含一个int32类型、标识符为count的关于通过人数统计的参数。

添加自定义功能	
* 功能类型: 属性 服务 事件 📀	
* 功能名称:	_
检测到人通过	0
* 标识符:	
peopelPassed	0
*事件类型:	
● 信息 ─ 告警 ─ 故障 @	
输出参数:	
- 参数名称: 通过人数	编辑 删除
+増加参数	
描述	
请输入描述	
	0/100
	UNIN BX

b. 为闸口产品添加gate01设备,并分配到边缘实例中。

c. 将闸口产品gate01设备的驱动设为设备模拟器驱动。

・添加显示器设备:

a. 新建显示器产品,在产品自定义功能处添加名为 content的字符串(text)类型属性。

- b. 为显示器产品添加设备,并分配到边缘实例中。
- c. 将显示器设备的驱动设为设备模拟器驱动。

📕 说明:

将设备分配到边缘实例后,请先不要部署边缘实例,待完成其他操作后统一部署。

- 2. 添加流数据分析作业。
 - a)参考分配流数据分析到边缘实例,创建、设置并发布计算人流量的流数据分析任务。

其中,任务类型选择SQL,执行任务选择边缘端。

创建任务		×
* 任务类型: ○ 组件编排 ● SQL		
* 任务名称:		
计算人流量		
* 任务描述: 计算人流量		
* 执行任务: 〇 云端		
	确定	取消

b) 在任务列表中,已创建的SQL任务名称右侧单击查看,进入SQL编辑器。

复制如下SQL内容到SQL编辑器编辑框中。

```
-- 计算每个闸口每十秒的人流量并显示到屏幕上
-- 每个闸口检测到人会上报peopelPassed事件,并附带经过的人数count
create table event(
    eventCode varchar,
    params varchar,
    productKey varchar,
    deviceName varchar,
    gmtCreate varchar,
    ts varchar,
    tstamp as to_timestamp (cast (ts as bigint)),
    WATERMARK wk FOR tstamp as withOffset (tstamp, 2000)
) with (
    type = 'edgebus',
```

```
jsonParser = 'device_event'
);
create table edgebus_sink (
    productKey varchar,
    deviceName varchar,
    flowrate int,
    t_start timestamp,
    t_end timestamp
) with (
    type = 'edgebus_sink',
    topic = '/sys/streamCompute/peopleFlowrate'
);
insert into
    edgebus_sink
select
    productKey,
    deviceName,
sum (cast (json_value (params, '$.count') as int)),
tumble_start (tstamp, interval '10' second),
    tumble_end (tstamp, interval '10' second)
from
    event
where
    eventCode = 'peopelPassed'
group by
    tumble (tstamp, interval '10' second),
    productKey,
    deviceName;
```

- c) 保存任务并发布。
- d) 参考分配流数据分析到边缘实例,将该SQL任务分配到边缘实例中。
- 3. 添加函数。
 - a) 请参考函数计算使用流程内容创建服务、并在该服务下创建名为DisplayFlowrate的使用Nodejs8运行环境的函数。
 - b) 创建函数完成后,在线编辑代码,添加如下内容:

] 说明:

```
请将your_displaydevice_productKey以及your_displaydevice_deviceName换
```

成实际网关设备的ProductKey和DeviceName。

```
const leSdk = require('linkedge-core-sdk');
const iotData = new leSdk.IoTData();
module.exports.handler = function(event, context, callback) {
    const payload = JSON.parse(JSON.parse(event.toString()).
payload);
    const params = payload.params;
    iotData.setThingProperties({
        productKey: 'your_display_device_productKey',
        deviceName: 'your_display_device_deviceName',
        payload: {
            content: `flowrate: ${params.flowrate}`
```

```
}
}, (err) => {
    if (err) {
        console.log(`display flowrate err: ${err}`);
      }
};
```

c) 在物联网平台控制台边缘实例的实例详情页面,选择函数计算,单击分配函数,将已创建 的DisplayFlowrate函数分配到边缘实例。

边缘实例 > 实例详情				
demo 未部署				部署详情重置部署
CPU 使用率: - 查看	分配函数	×		
实例信息 子设备 子设备通信	道 * 地域:			
	华东2(上海) ~			
函数计算	*服务:			分配函数
区域服务	DeviceScript ~	创建服务	分配时间	操作
	* 函数:	_		
	DisplayFlowrate	创建函数		
	* 授权:			
	AliyunIOTAccessingFCRole	创建RAM角色		
		分配 取消		
	回木祢加	江川四奴订异		
	IoTEdge可将函数计算从云扩展至边缘,函数计算。 可以帮助您实现本:	是一种可以在断网情况下按需运行或 也及时处理的复杂业务场景。	6者持续运行的应用程序,	
	分配函数	了解函数计算		

d) 配置函数。

参数	说明
运行模式	选择按需运行。
内存限制	设置为20 MB。
超时限制	设置为5秒。

- 4. 添加消息路由。
 - a) 在物联网平台控制台边缘实例的实例详情页面,选择消息路由,将闸口设备的数据路由到人 流量计算的流数据分析任务中。

添加消息路由	>
 * 消息来源: 设备 ∨ 闻口 ∨ displayDevi ∨ 	
* 消息主题过滤:	
事件 ~	
* 消息目标:	
流数据分析 〜 计算人流量 〜	
确	定 取消

b) 将人流量计算的流数据分析任务路由到DisplayFlowrate函数计算中。消息主题过滤设置 为/sys/streamCompute/peopleFlowrate。

添加消息路由				
* 消息来源: 流数据分析 ~	计算人流量	\sim		
* 消息主题过滤: /sys/streamCompute/peopleF	lowrate			
* 消息目标:				
函数计算 🗸 🗸 🗸	DisplayFlowrate	\sim		
			确定	取消

c)将显示器的属性变换数据路由到IoTHub中,方便在云端查看显示器设备的状态。

添加消息路由		×
* 消息来源: 设备		
唐性 ✓ * 消息目标: IoT Hub ✓		
* 服务级别: 1	0	
	确定	取消

- 5. 部署边缘实例并查看设备运行结果。
 - a) 在边缘实例的实例详情页面,单击部署,将子设备、流数据分析作业及消息路由下发到边缘端。
 - b) 在实例详情 > 实例信息页面, 单击网关页签下的远程连接, 登录远程控制台。
 - c) 进入/linkedge/gateway/build/bin目录,执行如下命令上报有人员通过的事件:

./ds_ctrl event a1Wn0t8NulA gate01 peopelPassed '{"count": 1}'

d) 在物联网平台控制台,选择设备管理>设备,在相应设备名称右侧单击查看。

在设备详情页面查看显示器设备的运行状态以及数据。

查看数据		\times
设备状态		
1小时 ~ 2019-02-13 11:20:36	- 2019-02-13 12:20:36 曲 表格 图	表
时间	原始值	
2019/02/13 12:20:22	flowrate: 14	
2019/02/13 12:20:03	flowrate: 12	
2019/02/13 12:19:53	flowrate: 7	
2019/02/13 12:19:45	flowrate: 2	

6 云资源访问

边缘实例中的函数计算、云监控等服务在网关上运行时,若需要调用阿里云其他云资源(例如调 用OSS的API),边缘实例使用RAM角色来实现对云资源权限的访问控制。

您可以参考RAM_{角色身份}和权限管理内容,了解RAM角色以及创建RAM角色和为角色授权的操作步骤。

操作步骤

- 1. 以阿里云账号登录物联网平台控制台。
- 左侧导航栏选择边缘计算>边缘实例,找到需要分配角色的边缘实例,并单击对应操作栏中的查看。
- 3. 在实例详情页面,选择设置,单击添加角色。
- 4. 在添加角色页面,单击创建服务角色进入RAM访问控制控制台的RAM角色管理页面。
- 5. 单击新建RAM角色,创建一个阿里云服务角色。

RAM访问控制	RAM访问控制 / RAM角色管理			新建 RAM 角色	\times
概览	RAM角色管理			选择可信实体类型	
人気繁琐 ヘ 用户頃 设置 収売管理 ヘ 授収	代金星RAM角色? RAMG色机制造向設備任的实体(eg.RAM用中、第个应用或用量运搬等) - 必定物下下的一个RAM用中(可能是代表一个移动App的的函数等) - 地域云明中中的RAM用中(電能云顶影中的函数的目) - StackLinchen用甲体的(G.C.S.S.P.P.P.P.P.P.P.P.P.P.P.P.P.P.P.P.P	进行授权的一种安全方法,根据不同应用场景,受信任的实体可能合加T 5月90股的方法。	5一些例子:	 ● 阿里元秋号 受信元素号下的自用户可以通过投演はAAM指色来切问等 の方面度、交結元準可以是当為元素号,也可以是対応 元素号 ● 河里元般 受信元都只可以通过投演和AAM自色来切问他的云面度 * 选择安值服务 ● 古田安田の 	
权限策略管理	RAM角色不同于传统的教科书式角色(其含义是指一组权限集)。如果您需	需要使用教科书式角色的功能,请参考RAM授权策略(Policy)。		*RAM角色名称	
RAM角色管理	新建RAM角色 输入角色名称或备注 Q			不超过64个字符,允许英文字母、数字,或"-"	
OAuth应用管理(公测…	RAM角色名称	备注	创建时间	备注	
		And the second s	2018年4月12日		
		the fight states a monthly state.	2018年7月20日		
		And in	2018年3月30년		
			2018年5月7日		
		STREET, STREET	2018年3月19日	确定 关闭	

其中,参数说明如下:

参数	描述
选择可信实体类型	选择阿里云服务。
选择授信服务	选择IoT物联网。

参数	描述
RAM角色名称	设置一个RAM角色名称。

参数设置完成后,单击确定,完成RAM角色的创建。

更多操作及参数说明,可以参考管理RAM角色内容。

- 6. 成功创建IoT物联网服务角色后,为角色添加您需要使用的云资源的访问权限。
 - a. 在RAM_角色管理页面,找到上一步创建的IoT物联网服务角色,在右侧操作栏中单击添加权限。
 - ・ 在添加权限页面, 左侧权限策略名称列中勾选需要授予当前角色的权限策略, 该权限策略
 会被添加到右侧区域框中。

	说明:		
可使用]关键字查询权限策略名称。		

LinkIoTEdgeRole@role.alivun-	document.onaliyunservice.com ×				
,,,,,,,					
择权限				1	
系统权限策略 > 日志		8	Q	已选择(1)	清防
权限策略名称	备注			AliyunLogFullAccess	×
AliyunLogFullAccess	管理日志服务(Log)的权限				
AliyunLogReadOnlyAccess	只读访问日志服务(Log)的权限				
AliyunYundunSLMFullAccess	管理云盾安全日志源管理(SLM)的权限				

- · 在右侧区域框, 选择某条策略并单击×, 可撤销该策略
- b. 单击确定,完成给角色授权。

、天力ロオフル日



后续如果在使用物联网边缘计算过程中,需要访问更多的云资源,则继续为已创建的IoT物联网服务角色添加相应服务的权限。更多操作方法可参考RAM 授权内容。

7. 返回物联网平台控制台,在实例详情页面,添加角色弹窗中,配置角色信息,授予边缘实例操作 对应云资源的能力。

边缘分组 → 分组详情					
myhome 未部署		编辑	部署详情	重置	部署分组
最近修改:2018-09-27 11:04:51 分组标签:					
网关 设备 规则计	添加角色 ×				
分组角色管理	 选择已经存在的角色,可以授予分组操作对应云资源的能力 *角色: 				添加角色
角色名称	请选择角色 · · · · · · · · · · · · · · · · · · ·	操作			
	◆系统模板授权: 权限将根据角色自动选取				
	職定 取消				
	尚未添加任何角色 IoT Edge可通过或动连接边缘上的IoT设备并将这些设备数据转换为IoT标曲的物模型设备 您可将这些数据通过消息路由流转至边缘函数计算中,也可将数据发往IoT Hub。 添加角色				

参数	描述					
角色	选择已创建的IoT物联网服务角色。					
系统模板授权	权限根据已选的角色,自动选取。					

8. 单击确定, 您已成功为实例添加了角色。

9. 添加角色完成后单击页面右上角部署,并在弹出对话框中,单击确认。

7远程运维管理

远程运维管理包括远程连接和远程文件管理。物联网边缘计算支持使用远程连接方式登录到网 关,方便您控制网关设备。远程文件管理是对边缘网关上文件的管理,可以将本地PC上的文件上传 到边缘网关或将边缘网关上的文件下载到本地PC上。远程连接和远程文件管理是基于SSH技术实现 的,因此使用的用户名和密码是Docker镜像或网关上SSH服务的用户名与密码。

远程运维管理功能已默认集成到Link IoT Edge的v1.8及以上版本,同时允许您单独集成,目前代 码已开源,详见github开源库。

前提条件

・网关必须开启SSH服务,且端口默认为22。开启SSH服务方法请参考OpenSSH Server。

如果端口有变更,需要局域网登录网关,并手动修改/linkedge/gateway/build/bin/ remote_terminal.json中的listen_port字段为更新的端口,同时网关必须支持127.0. 0.1本地回环端口的访问。

· 已在物联网平台控制台成功创建边缘实例并分配网关。

远程连接

- 1. 在物联网平台控制台,选择边缘计算 > 边缘实例。
- 2. 选择已创建的实例,单击右侧的查看。
- 3. 实例详情页面,选择实例信息,在网关设备右侧操作栏中,单击远程连接。

myhome1220 部署成功 部署详情 重置 部署							重置部署				
CPU 使用率: -	查看	内存使用率: - 查看		存储使用率: - 查看							
实例信息	子设备 子设备通信	通道 场景联动 函数	如计算 流数	据分析 消息路由	日志服务	设置					
实例信息											
实例名称	myhome1220		部属状态	部署成功			云监控状态				
CPU 使用率	- 查看		内存使用率	- 查看			存储使用率	- 查看			
实例进程	查看		创建时间	2018-12-20 23:59:45							
修改时间	2018-12-21 00:15:31										
实例标签											
网关											分配网关
网关名称		产品名称			状态	最	后上线时间		操作		
mygateway1	220	边缘计算	节点1220		●在线	-			查看 移除 远程	连接远程文件	+管理

 在弹出对话框中,输入远程登录网关的用户名和密码(即Docker镜像的用户名和密码,初始值 为root/iiot),单击确认。

连接成功后,您将看到如下远程连接页面。

!) 注意:

远程登录网关,具有高级权限,可进行更新、删除、改动、查看各类网关设备的信息和文件。 请谨慎使用。

辺線変例 → 実例详備 → 返程控制台
Welcome to Ubuntu 16.04.5 LTS (ONU/Linux 4.15.0-36-generic x86_54)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
85 packages can be updated.
0 updates are security updates.
Now release 18.04.1 LIS' available.
Run 'do-release-upgrade' to upgrade to it.
And System result requires and requires and results an
Last Login: Lue Lue: 4 10:14:100 /2013 FTCH 394.35/2/.34
Sea Sity role 24 do exotit leveratemi-20color do dash
yueniggut raise is a th
-fw-rw-r 1 0 12/3 1 11:27 8 11). Rey
-rk-rk-r 1 2/4 12/ 1 22/18 atc. key
Ghwanxi-x 2 4.0k 11.7 23 14:35 addstasdt
-ru-ru-r 1 249 12/5 1 17:40 5.Key

远程文件管理

- 1. 实例详情页面,选择实例信息,在网关设备右侧操作栏中,单击远程文件管理。
- 2. 在弹出对话框中,输入远程登录网关的用户名和密码(即Docker镜像的用户名和密码,初始值 为root/iiot),单击确认。

连接成功后,您将看到如下远程文件管理页面。

边缘实例 > 实例详情 > 远程文件管理							
deviceaccess 当前目录: /			刷新新增目录 上作文件				
文件名称	文件大小	更新时间	操作				
	0 B	2018-11-25 22:13:32	下载 删除				
	4.00 KB	2018-09-17 21:23:38	查看 删除				
	12.00 KB	2018-12-04 06:55:39	查看 删除				
	4.00 KB	2018-07-20 16:37:11	查看 删除				
		共有34条 〈上一页	1 2 3 4 下一页 > 每页显示: 10 >				

目前远程文件管理提供目录浏览与创建,文件上传与下载等功能。
3. (可选)在远程文件管理界面,文件对应的操作栏中单击下载,可下载网关设备文件到本地。

边缘实例 > 实例详情 > 远程文件管理				
deviceaccess 当前目录: /home/ / 1_回上级				刷新 新增目录 上传文件
文件名称	文件大小	更新时间	操作	
	21.52 KB	2018-12-04 19:27:43	下載的除	
E	下载文件		下载 删除	
	文件加载中, 请超后		下載 删除	
	•	7%	下载 删除	
			查看 删除	
		共有180条 〈上一页 1 2 3 4 …	18 下一页 > 3/18 到第	页 确定 每页显示: 10 ~

4. (可选)在远程文件管理界面,单击上传文件,可上传本地文件到网关设备当前目录。

边缘实例 > 实例详情 > 远程文件管理				
deviceaccess 当前目录: /home/ 1_回上级				別新 新増目录 上传文件
文件名称	文件大小	更新时间	操作	
	上传文件	×	下载 删除	
	*上传文件	1	下载 删除	
	上传中		下載 删除	
	linkedge_x86_64.tar.gz - 20% ×		下载 删除	
	L		查看 删除	
		确定取消	下載 删除	
	共有180条	〈上一页 1 2 3	4 18 下一页 > 3/18 到第	页 确定 每页显示: 10 ~

8 设备模拟器

物联网边缘计算(Link IoT Edge)提供了场景联动、函数计算、流数据分析等多种服务,帮助用 户构建物联网应用程序。考虑到用户没有大量可连接的实际设备来测试使用应用程序,物联网边缘 计算同时提供一套设备模拟器(DeviceSimulator)的解决方案。

设备模拟器包含两部分:

- · 驱动: 在边缘端可以模拟所有类型的设备(设备的属性、事件、服务取决于物模型的定义)。
- · 控制工具:改变设备模拟器的属性值、触发设备模拟器进行事件上报。功能与真实物理设备的按 键类似。

您只需要将设备和设备模拟器的驱动绑定,并部署至边缘端,结合控制工具,便可以脱离真实的物 理设备进行应用程序的测试。下文将会对设备模拟器的使用进行详细介绍。

前提条件

- · 下载设备模拟器驱动。
- · 设备模拟器只适用于Link IoT Edge标准版和专业版v1.7及以上版本环境中使用,因此开始操作本章内容前,请确保您已根据专业版环境搭建或标准版环境搭建内容完成边缘实例的创建。

使用设备模拟器驱动

- 1. 上传驱动。
 - a. 在物联网平台控制台左侧导航栏中,单击边缘计算 > 驱动管理。
 - b. 在驱动管理页面,单击右侧新增驱动。
 - c. 在新增驱动界面,根据提示设置驱动参数,上传<mark>前提条件</mark>中下载到本地的设备模拟器驱动文件。

物联网平台	驱动管理						
快速入门					100		
设备管理	驱动管理列表		新增驱动	×			刷新新增驱动
规则引擎		1	* 语言类型:				
数据分析	驱动名称	驱动语	Node js 8	/		修改时间	操作
边缘计算			* 驱动名称:				
边缘实例	LightSensor 宣方示例	Node.j	DeviceSimulator	0	里提供的	-	查看 源码
驱动管理			驱动描述:				
开发服务	Light 官方示例	Node.j:	设备模拟器		提供的用		查看 源码
应用托管	Modbus 官方	Python			iΩmodbus	_	查看 源码
行业服务			* 驱动文件:				
监控运维	OPCUA 宣方	Python	重新上传 🔘		server		查看 源码
产品文档			DeviceSimulator.zip上传成功 X				
	The statements	Node.j:				2019-02-13 16:37:36	编辑 删除 上传 源码
	No. (1999)	Node.j:		确定取消		2018-11-16 17:58:14	编辑 删除 上传 源码

d. 完成参数的设置并上传成功驱动文件后,单击确定。您可以在驱动管列表中看到刚刚创建的 驱动。 2. 添加子设备到边缘实例。

为前提条件中完成的边缘实例分配子设备。

- a. 在实例详情页面,选择子设备,单击分配子设备。
- b. 在分配子设备页面中, 单击新建子设备。
- c. 在新建子设备页面, 单击新建产品, 创建产品。

边缘实例 > 实例详情		分配子设备			×
demo 未部署					
CPU 使用率: - 查看	内存使用率:- 查看	TEST01291 ~ 设备名称	搜索		刷新新建子设备
实例信息 子设备 子设备通信通	道场景联动 函数计算 流数据分	设备名称 产品名称	状态	最后上线时间	操作
	新建子设备	×			
子设备管理	 特别说明:deviceName可以为空,当为空时 为deviceName。 	,阿里云会颁发全局唯一标识符作	暂无设备可分配,请前往新政	圭子设备	
设备名称	• 产品:	 新額产品 総认 取消 			
			_		
					元成

d. 在创建产品页面设置参数后,单击确认。

新建产品		×
产品信息		
* 产品名称		
温度传感器		
* 所属分类 ②		
边缘计算 / 其他设备	\sim	功能定义
连网与数据 接入网关协议		
自定义	\sim	?
更多信息		
产品描述		
请输入产品描述		
0/*	100	
0/	.00	

e. 在新建子设备页面,产品自动分配已创建的产品,输入设备名称后单击确认,为产品添加设备。

新建子设备

 \times

特别说明:deviceName可以为空,当为空时,阿里为deviceName。	云会颁发全局唯一标识符作
产品:	
温度传感器	→ 新建产品
设备名称:	
temperatureSensor01	

f. 在分配子设备页面,分配温度传感器产品下的temperatureSensor01设备到实例中,并单 击完成。

分配子设备					×
温度传感器 🗸 🗸	设备名称	搜索		刷新 新	建子设备
设备名称	产品名称	状态	最后上线时间	操作	
temperatureSensor0 1	温度传感器	●未激活	-	分配	



确认

取消

3. 为子设备配置驱动。

在已分配的子设备右侧单击驱动配置,为设备分配步骤1步骤中上传的驱动。

ジ焼変例 > 次例評構 demo 未認定 CPU使用率 - 直着 「 次例信息 子设备 子设备通信通道	9存使用率:-宣誓 场帮联动 函数计算 流数	存储使用率 - 宣音 据分析 消息路由 日志服务 设	꺱	部署许俏 重重 部署
子设备管理 请输入设备名称	黎动配置 * 边球驱动: DeviceSimulator	× -> 新建驱动		刷新 分配子设备
设备名称 temperatureSensor01	7 自定义配置: 0	● 報定 取消	段备状态 最后上线时间	操作 驱动配置 查看 移除

4. 为产品配置物模型。

为已创建的温度传感器产品配置物模型。

a. 参考新增物模型,为产品添加temperature属性。

* 功能类型:	
属性 服务 事件 ?	
* 功能名称:	
温度	?
* 标识符:	
temperature	?
* 数据类型:	
int32 (整数型) ~	
* 取值范围:	
0 ~ 100	
* 步长:	
1	
单位:	
度/° ~	
读写类型:	
● 读写○ 只读	

b. 参考新增物模型,为产品添加high_temperature事件。

添加自定义功能

۰.		1	
	4	r	
	ጥ		
/		`	

* 功能类型:	
属性服务事件?	
* 功能名称:	
高温报警	0
* 标识符:	
high_temperature	0
*事件类型:	
🔵 信息 💿 告警 🗌 故障 🛛 😰	
输出参数:	
+增加参数	
描述	
请输入描述	
	0/100

为事件增加参数。

新增参数

 \times

温度		0
* 标识符:		
temperature		0
* 数据类型:		
int32 (整数型)	\sim]
* 取值范围:		
0 ~ 1	100]
* 步长 :		
1		
单位:		
摄氏度/℃	~	

确认 取消

5. 添加消息路由。

为边缘实例配置消息路由,用于将温度传感器的数据上传到云端。

- a. 在实例详情页面,选择消息路由,单击添加路由。
- b. 在添加消息路由页面中, 配置参数。

边缘实例 > 实例详情			
demo(未部署)			部響详情重置部署
CPU 使用率: - 查若 内存使用率:	查看 存储使用率: - 查看		
实例信息 子设备 子设备通信通道 场景联动	函数计算 流数据分析 <mark>消息路由</mark> 日志服务 设置		
消息路由	添加满意踏由		添加路由
消息来源 消息主题:	* 消息来源:	分配时间 操作	
	- 设备		
	* 消息主题过滤:		
	全部 🗸		
	* 消息目标:		
	loT Hub \checkmark		
	* 服务级别:		
	1 ~ 0	≥间进行流转,	
	職證 取消		

6. 部署边缘实例。

在实例详情页面,右上角单击部署,部署边缘实例。部署成功后边缘实例名称后显示部署成功。

使用设备模拟器控制工具

设备模拟器控制工具的功能和真实物理设备的按键类似。

- 1. 登录你的远程终端。登录操作详情请见远程运维管理。
- 2. (可选)执行下载命令:

目前v1.8.1(不含)以上版本的Link IoT Edge标准版和专业版软件已经默认集成该工具,不 需要下载。

·若登录远程终端时,使用root账号,执行:

```
platform=Your_Platform && output=/linkedge/gateway/build/
bin/ds_ctrl && wget http://link-iot-edge-packet.oss-cn-
shanghai.aliyuncs.com/tool/ds_ctrl_$platform -0 $output && chmod
+x $output
```

·若登录远程终端时,使用其他账号(需要支持sudo命令),执行:

```
platform=Your_Platform && output=/linkedge/gateway/build/
bin/ds_ctrl && sudo wget http://link-iot-edge-packet.oss-cn-
```

shanghai.aliyuncs.com/tool/ds_ctrl_\$platform -0 \$output && sudo
 chmod +x \$output

(!) 注意:

其中, Your_Platform取决于您使用的物联网边缘计算软件的版本,请参考下表示例:

版本	使用环境	Your_Platform
标准版(LE Standard)v1.8.1及	ARMv7 Linux	arm7
	ARMv7 VPFv3 Linux	arm7
	ARMv8_64 Linux	arm8
	x86_64 Linux	x86
专业版(LE Pro)v1.7及以上	x86_64 Docker环境	x86

例如,您的物联网边缘计算软件版本版本为专业版,且使用root账号登录远程终端,则实际执行的命令为:

```
platform=x86 && output=/linkedge/gateway/build/bin/ds_ctrl && wget
    http://link-iot-edge-packet.oss-cn-shanghai.aliyuncs.com/tool/
    ds_ctrl_$platform -0 $output && chmod +x $output
```

3. 使用设备模拟器控制工具控制设备行为。

进入/linkedge/gateway/build/bin 目录,执行 ./ds_ctrl命令即可看到工具的使用方

式。

Usage: ds_ctrl <command>
where <command> is one of:
 property, event
ds_ctrl property productKey deviceName params Change
property
ds_ctrl event productKey deviceName eventCode params Trigger
event
For example(productKey = xxx, deviceName = yyy):
 ds_ctrl property xxx yyy '{"temperature":30}'

```
ds_ctrl event xxx yyy alarm '{"temperature":90}'
```

- · 可改变设备属性:
 - a. 改变温度。

使用工具将温度传感器温度改为30:

```
./ds_ctrl property alWuxHrhojr temperatureSensor01 '{"
temperature":30}'
```

系统返回如下信息,表示温度设置成功:

Send property success!

b. 在物联网平台控制台,选择设备管理>设备,在相应设备名称右侧单击查看。

在设备详情页面查看设备的运行状态。

物联网平台	2019-01-14发布公告:物联网平台收费变更!查看详情
	设备管理 > 设备详情
快速入门	temperatureSensor01 在线
设备管理	
产品	广品:温度传感器 查看 ProductKey: 复制
设备	设备信息 Topic列表 运行状态 事件管理 服务调用 日志服务
分组	
规则引擎	运行状态 设备数据上报的最新属性值,点击"查看数据"可以查看指定属性的历史数据
数据分析 🛛 🗸 🗸	温度
边缘计算 >	30 •
应用管理	2019/01/24 15:47:37
扩展服务	

c. (可选)模拟多次温度变化。

连续执行如下命令5次,每次将温度值+1。

```
./ds_ctrl property a1WuxHrhojr temperatureSensor01 '{"
temperature":31}'
```

可在控制台中,设备的运行状态页面下查看数据变化。

2019-01-14发布公告:	查看数据	×
设备管理 > 设	设备状态	
temperatu	1小时 2019-01-24 14:53:16 - 2019-01-24 15:53:16 中市 表格	图表
产品:温度传感		ISIAX
设备信息	35	
	34	
运行状态设备	33	
温度	32	
35 °		
2019/01/24	31	
		关闭

・可触发设备事件:

a. 上报高温事件。

./ds_ctrl event a1WuxHrhojr temperatureSensor01 high_tempe rature '{"temperature":90}'

b. 在物联网平台控制台,选择设备管理>设备,在已相应设备名称右侧单击查看。

在设备详情页面查看设备的事件管理内容。

temperatureSensor01	在线			
产品:温度传感器 查看	ProductKey	al Marenhap 复制	DeviceSecret: ******** 📆	π
设备信息 Topic列表	运行状态 事件管理 服务调	用 日志服务		
事件管理				
请输入事件标识符	搜索 全部类型 > 1小时	\sim		刷新
时间	标识符	事件名称	事件类型	输出参数
2019/01/24 16:04:57	high_temperature	高温报警	alert	{"temperature":90}

在线调试设备

1. 在物联网平台,选择监控运维 > 在线调试,在在线调试页面,选择调试产品和调试设备。

设备	
分组	
规则引擎	调试设备: 温度传感器 ··· temperatureSensor01 ···
数据分析	
边缘计算	编辑设备
应用管理	调试真实设备 虚拟真实设备
扩展服务	功能: 请洗择 方法: 请洗择 🗸
监控运维	1
实时监控 1	
在线调试	
日志服务	
固件升级	
远程配置	
产品文档	发送指令 重置

2. 获取设备属性。

选择功能、方法后,单击发送指令,即可在输入框中看到当前温度值为35。



3. 设置设备属性。

将温度设为40,单击发送指令后可在设备运行状态查看温度传感器温度值已变更为40。

调试真实设备 虚拟真实设备	1
功能: ₂ 温度(temperat… ~	方法: 设置 🗸
1 { 2 "temperature": 40	
5	
3	
发送指令 重置	

4. 其他自定义服务,会将相关信息打印到驱动日志中,并返回如下信息。

```
{"code":0,"message":"success"}
```

9日志服务

Link IoT Edge 日志服务是通过集成阿里云日志服务云产品来完成边缘端日志的管理功能,包括边缘日志自动同步至云端存储、日志分级别显示等。

物联网边缘计算中的日志服务使用的是阿里云_{日志服务}产品功能。阿里云日志服务在超过免费额度 后会产生一定的费用,在使用前请您仔细阅读该服务的<mark>计费方式</mark>。

前提条件

在开始操作本章内容前,请您确保已根据<mark>环境搭建</mark>内容完成了边缘实例的创建,并根据云资源访 向内容为创建的实例添加了服务角色。

操作步骤

- 在物联网平台控制台左侧导航栏中选择边缘计算 > 边缘实例,找到要查看日志的实例,单击对应 操作栏中的查看。
- (可选)若您第一次使用日志服务功能,则需要开通服务。若已开通服务,则跳过本步骤进行下 一步操作。
 - a. 在实例详情页面,选择日志服务,单击开通日志服务。
 - b. 使用日志服务需要先进行授权,在系统弹窗中单击前往授权,跳转到RAM访问控制中的RAM角色管理页面。

请参考云资源访问内容,为以创建的IoT物联网服务角色添加一个名为AliyunLogFullAccess的访问阿里云日志服务的权限。

边缘实例 > 实例详情	
LinkloTEdge_Node	部署详情重置部署
CPU 使用率: • 直看 存储使用率: • 直看	
实例信息 子设备 子设备通信通道 场景联动 函数计算 流数据分析 消息路由 日志服务 设置	
日志服务 日志服务省次开通需要1分钟左右的时间创建表引,期间查询功能不可用。	● 启用状态: 日志存储策略
全部限制× × 2019-01-21 15:37:40 - 2019-01-28 15:37:40 +	
序号 日志类型 级别 ●使用日志服务需要先进行授权,是否前往授权? × 前¥社授权 取消	时间
尚未开通日志服务	
Link to T Edge 日本服务完通过集成无端日本服务产品来完成边缘日本的管理功能, 包括边缘日志自动同步云端存储、日志分级别显示等。	
开通日志服务	

3. 在实例详情页面日志服务页签下,单击日志存储策略,设置您的日志存储策略。

日志存储使用LOG(阿里云日志服务)产品能力,因此日志存储时间越长所需空间就越多,请 按需设置存储时间。

边缘实例 > 实例详情		
LinkloTEdge_Node 部署成功	1	部署详情 重置 部署
CPU 使用率: - 查看	内存使用率 - 查看 存储使用率 - 查看	
实例信息 子设备 子设备通信	系述 括照联洲 动物计结 治教师公析 迷自敗由 口 主服徒 识婴 日志存储策略 ————————————————————————————————————	
日志服务 日本 日本<	 日志存儲需要使用SLS产品能力,日志存儲加间越长所需空间越多,请按需设置存儲加间 日志保存时间: 	自用状态: 10
序号 日志类型	7 × 天	Bitta
	• 日志吸到: 错误 🗸 🖉	
	戦定取消	

参数说明如下:

参数	描述
日志保存时间	网关日志被保存的时间长度。
日志级别	选择存储哪一个级别及以上的日志,日志级别优先级如下: 错误>警告>信息>调试 例如,选择信息,表示存储信息、警告、错误三个级别的日志。

4. 单击确定完成日志存储策略的设置。

当网关产生日志后,物联网边缘计算会根据您日志存储策略,过滤日志的内容并在日志服务页面 中展示。

(可选)您还可以根据日志级别和时间范围来搜索查询日志服务中的日志,展示在日志服务页面中。

10 固件升级

物联网平台提供固件升级服务,您只需在控制台上传新的固件,即可将升级消息推送到设备端,完成设备在线升级。

您需要在发布历史中获取新的固件包。

物联网边缘计算使用固件升级时,有以下限制条件。

设备端软件包要求

- · 仅支持v1.8.1及以上版本的Link IoT Edge标准版(LE Standard)软件包使用固件升级服务。
- ・要求Link IoT Edge软件包的运行路径必须位于/linkedge目录中,即启动脚本位于 / linkedge/gateway/build/script/iot_gateway_start.sh。
- ·要求/linkedge/run/binary-ota/的剩余空间不少于自包含包大小及解压缩后的大小之和。
- ・要求/linkedge/gateway/build/的剩余空间不少于自包含包解压缩后的大小。

控制台升级设置

・新增固件时:

表 10-1: 参数设置要求

参数	描述
固件版本号	必须使用/linkedge/gateway/build/config/fota/ fota_config.ini文件中version字段的参数值。示例如图所示。
	<pre>fota_config.ini [fota] version=v1.8.1</pre>
签名算法	仅支持MD5方式。

・验证固件、批量升级固件时:

表 10-2: 参数设置要求

参数	描述
升级方式	仅支持整包升级。

11 产线烧录

本文主要介绍物联网边缘计算产品的产线烧录流程,帮助您快速完成批量生产和产线激活。

背景信息

为保障设备安全,阿里云物联网平台为设备颁发证书,包括产品证书(ProductKey和 ProductSecret)与设备证书(ProductKey、DeviceName和DeviceSecret)。其中,设备证 书与真实设备一一对应,以确保设备的唯一合法性。设备通过协议接入阿里云之前,需依据不同的 认证方案,使用产品证书或设备证书计算签名,通过云端认证后,方可接入物联网平台。

物联网边缘计算服务的产品管理能力复用自物联网平台,支持两种设备认证方式:

·一机一密:逐台为每台设备烧录单独的设备证书。安全级别高。

· 一型一密: 批量为同一产品类型的设备烧录相同的产品证书。批量烧录, 操作方便。

详细信息可参考物联网平台设备安全认证文档。

准备工作

引导程序

引导程序用于为边缘计算产品配置设备证书信息。一机一密和一型一密两种方案都需使用。

该程序必须开机启动。获取设备证书信息后,需做简单的合法性校验,如:是否为空,长度是否符 合要求(ProductKey长度11位,ProductSecret长度32位)等。

程序逻辑示例如下:



产线烧录工具

若您使用一型一密认证方案,阿里云为您提供了源码及二进制产线烧录工具。输入产品证书信息(ProductKey、ProductSecret)及DeviceName后,您可以获得设备证书信息(ProductKey、 DeviceName和DeviceSecret)。

```
运行工具,命令后面跟二元组和DeviceName信息:
$/linkedge/gateway/build/bin/tool_generate_3triple ProductKey
DeviceName ProductSecret
返回结果:
$DeviceInfo: ProductKey DeviceName DeviceSecret
```

1 说明:

- ・产线烧录工具仅用于一型一密认证方案。
- 一型一密方案里,一个设备只能通过产品证书(ProductKey和ProductSecret)获取唯一一次设备证书信息(ProductKey、DeviceName和DeviceSecret)。因此,请您务必将烧录 工具获取到的设备信息存储到掉电/重置不丢失的Flash安全存储区。
- ・开发方式:
 - 您可以发送邮件到link-iot-edge-cert@list.alibaba-inc.com, 讲明使用缘由, 获 取工具和源码进行二次开发。
 - 您也可以使用Link Kit SDK自行实现该逻辑。示例代码请参考LinkKit Sample

一机一密产线烧录

一机一密认证方法,即预先为每个设备烧录其唯一的设备证书(ProductKey、DeviceName和 DeviceSecret)。当设备与物联网平台建立连接时,物联网平台对其携带的设备证书信息进行认 证。



- 参考物联网平台创建产品(高级版)与批量创建设备章节,批量创建边缘计算/边缘网关类型的网关 设备,并记录设备证书信息(包含ProductKey, DeviceName和DeviceSecret)。
- 2. 在产线上将设备证书信息烧录至实际设备中。
 - a. 将引导程序烧录到设备的某一分区。
 - b. 将设备证书信息(ProductKey、DeviceName和DeviceSecret)烧录到设备的其它分区,确保安全存储。

3. 为设备插上网线,开机后确保设备可以访问公网。

Ê	说明:	
您可り	【在设备上执行以下命令,	确认网络环境:

- · \$ping iot-as-mqtt.cn-shanghai.aliyuncs.com: 确认外部网络状态。
- · \$ping 127.0.0.1: 确认内部网络状态。
- ・\$/linkedge/gateway/build/scripts/iot_gateway_status.sh: 确认引导程序运行正常,物联网边缘计算服务开启。
- 4. 云端查看设备状态。若设备状态变为在线,表明烧录成功。

物联网平台	设备管理 > 设备详情								
快速入门	gateway_01 在线								
设备管理へ	产品:LinkloTEc	lge_Gateway 查看	ProductKey	:a1hR YJ7 复制	DeviceSecret : *****	DeviceSecret: ******** 显示			
产品	设备信息	设备信息 Topic列表 运行状态 事件管理 服务调用 日志服务 子设备管理 子设备通道管理							
设备	况存在中								
分组	收 會 信息								
规则引擎	产品名称	LinkloTEdge_Gateway	ProductKe	y a1hF YJ7 复制	区域	华东2(上海)			
数据分析 🛛 🗸 🗸	节点类型	网关	DeviceNar	me gateway_01 复制	DeviceSecret	******* 显 示			

一型一密产线烧录

一型一密安全认证方式,即为同一产品下所有设备烧录相同固件(固件中写入产品证书,即 ProductKey和ProductSecret)。选用此种方式,您也需要预先在物联网平台上,开启动态注册 开关,保存产品证书。设备基于产品证书发送激活请求时,物联网平台根据产品证书进行认证。认 证通过,物联网平台下发该设备对应的DeviceSecret,设备便可以使用设备证书信息接入物联网 平台。



创建边缘计算/边缘网关类型的网 关产品和设备;获取产品证书。

控制台手动开启设备动态注册

- 参考物联网平台创建产品(高级版)与批量创建设备章节,批量创建边缘计算/边缘网关类型的网关 设备,并记录产品证书信息(包含ProductKey和ProductSecret)。
- 2. 手动开启产品动态注册按钮。



- ・开启动态注册需通过手机验证。
- ·若设备发出激活请求时,系统校验发现该开关未开启,将拒绝新设备的动态激活请求。已激 活设备不受影响。

・建议所有设备完成激活后,关闭动态注册开关。

物联网平台	产品管理 > 产品详情									
快速入门	LinkloTEdge_Gateway 高级版 发布									
设备管理	ProductKey : a1hR	YJ7 复制	ProductSecret : *****	*** 显示	设备数:3 前往管理					
产品	产品信息	Topic类列表 功能定义 服务端证	丁阅 日志服务	在线调试						
设备										
分组	产品信息									
规则引擎	产品名称	LinkloTEdge_Gateway	节点类型	网关	创建时间	2019/01/11 16:45:13				
数据分析	产品版本	高级版	所属分类	边缘网关	数据格式	ICA 标准数据格式 (Alink JSON)				
边缘计算	动态注册 💿	已开启 💽	ProductSecret	******** 显示						
开友服务 New 应用托管	状态	 ● 开发中 	是否接入网关	否	连网协议	WiFi				

- 3. 在产线上将产品证书信息(ProductKey和ProductSecret)烧录至实际设备中。
 - a. 将烧录工具和引导程序烧录到设备的某一分区。
 - b. 将产品证书信息,烧录到设备的其它分区,确保安全存储。
 - c. 运行烧录工具,通过产品证书信息获取设备证书信息。即云端下发DeviceSecret至设备端。

```
$/linkedge/gateway/build/bin/tool_generate_3triple ProductKey
DeviceName ProductSecret
```

- d. 将设备证书信息存储到烧录工具和引导程序以外的其它分区,确保安全存储。
- 4. 为设备插上网线,开机后确保设备可以访问公网。

您可以在设备上执行以下命令,确认网络环境:

- · \$ping iot-as-mqtt.cn-shanghai.aliyuncs.com: 确认外部网络状态。
- · \$ping 127.0.0.1: 确认内部网络状态。
- ・\$/linkedge/gateway/build/scripts/iot_gateway_status.sh: 确认引导程序运行正常,物联网边缘计算服务开启。
- 5. 云端查看设备状态。若设备状态变为在线,表明烧录成功。

物联网平台		设备管理 > 设备详情										
快速入门		gateway_01 在线										
设备管理		产品:LinkloTEdge	e_Gateway	話		ProductKey:a1hR YJ7 复制				DeviceSecret:******* 显示		
产品		设备信息	Topic列表	运行状态	事件管理	服务调用	日志服	务 子设备管理	子设备通道管	理		
设备		还有信白										
分组		· 按會信息										
规则引擎		产品名称	LinkloTE	Edge_Gateway		ProductKey		a1hF YJ7 🧮		区域	华东2(上海)	
数据分析	/	节点类型	网关			DeviceName	9	gateway_01 复制		DeviceSecret	******* 显示	

常见问题

· 一型一密方案获取设备证书信息失败怎么办?

答:应用程序获取设备证书成功后,须妥善保存至设备的安全存储区域,谨防丢失。

若获取设备证书失败,需要在物联网平台上,手动删除该设备后,方能重新获取。删除设备需经 过手机信息验证。