



分布式消息服务 MQTT

用户使用指南

中国电信股份有限公司云计算分公司

目 录

目录

1. 产品概述	3
1.1 产品定义	3
1.2 基本概念	3
1.3 产品架构	4
1.3.1 产品形态.....	4
1.3.2 消息分类存储	5
1.3.3 数据生态.....	6
1.4 应用场景	6
2. 快速入门	8
2.1 创建实例	8
2.2 连接实例	8
2.3 使用须知	8
2.3.1 公网接入.....	8
2.3.2 消息数据存储	9
2.3.3 会话机制.....	9
2.3.4 离线消息.....	9
2.3.5 系统主题.....	9
2.3.6 名称解释.....	10

2.3.7	使用限制	10
2.4	开发指南	11
2.4.1	SDK 支持	11
2.4.2	主题规则	12
2.4.3	消息收发	13
3.	控制台使用	15
3.1	实例生命周期管理	15
3.2	实例详情	15
3.3	消息存储	15
3.4	认证授权	18
3.5	连接查询	19
3.6	设备轨迹	19
3.7	资源报表	20
3.8	监控告警	22
4.	常见问题	23
4.1	终端设备一直处于断线状态	23
4.2	终端断线重连后一直未收到消息	23
4.3	客户端连接异常断开	23
4.4	MQTT BROKER 连接数为 0	23
4.5	离线消息缺失	23

1. 产品概述

1.1 产品定义

分布式消息服务 MQTT 是面向移动互联网以及物联网领域的轻量级消息中间件，可以在有限的资源条件下，为连接远程设备提供实时可靠的消息服务并支持数据高效分类存储、再处理，实现终端设备与云端应用互通。

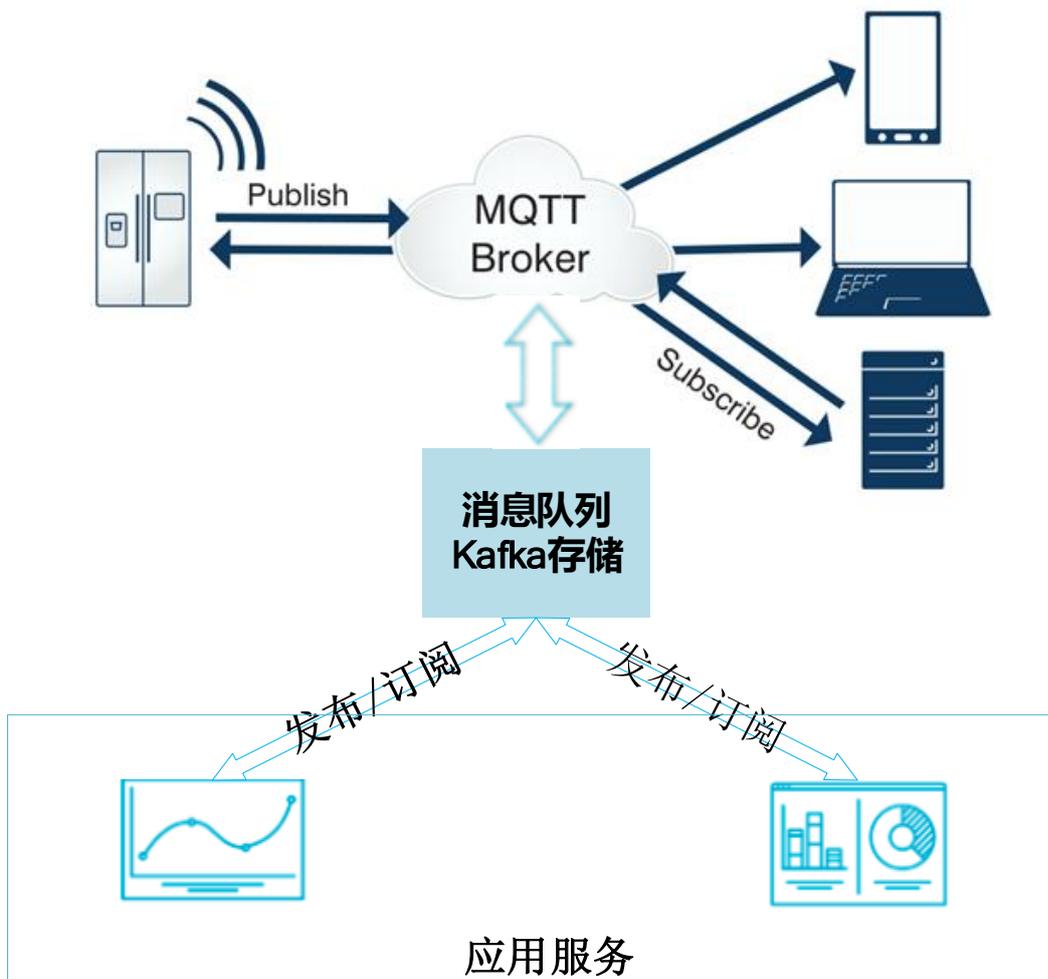
1.2 基本概念

实例	创建购买消息队列 MQTT 服务的实体单元，包含 MQTT Broker 和 kafka 集群。
MQTT Broker	MQTT 协议交互的服务端节点，承担 MQTT 客户端连接接入、管理、数据转发、消息数据存储至 kafka 消息队列。
MQTT 客户端	和 MQTT Broker 交互的终端节点
父 Topic	MQTT 协议，topic 存在层级性，定义第一层级 topic 为父 topic，
子 Topic	除第一层父 topic 外，MQTT 协议 topic 后续层级统称子 topic
ClientID	每个 MQTT 客户端唯一标识
GroupID	若终端设备需要分类，可创建 GroupId；对应 ClientID 则以 GroupID@@开头，可按 GroupId 提供在线数查询。非强制要求，用户自行决定是否使用。
消息队列 kafka	MQTT Broker 对接收的终端消息数据按父 topic 分类存储至 kafka 集群队列；云端应用系统通过 kafka 集中处理分析和指令下发

云端应用服务	用户后端消费数据系统
--------	------------

1.3 产品架构

1.3.1 产品形态



MQTT 客户端

MQTT Broker 交互的 pub 和 sub 设备。

MQTT Broker

承担移动端连接接入、连接管理、数据转发至订阅的设备和 kafka 消息队列。

消息队列 kafka

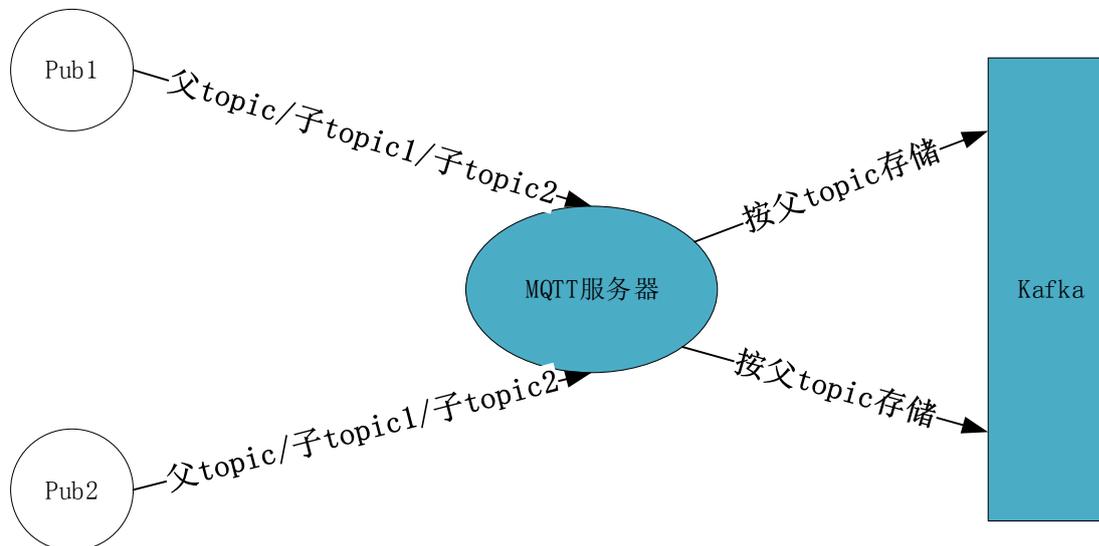
存储 MQTT 消息，接收后端应用服务系统指令转发至 MQTT broker。

应用服务

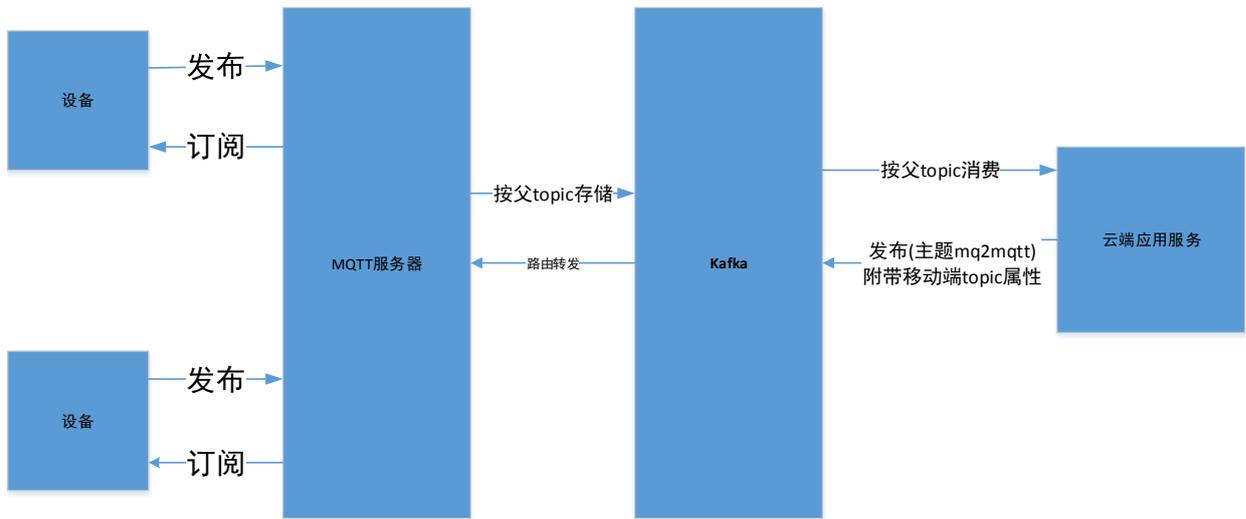
分析、处理 kafka 的数据和向其下发指令。

1.3.2 消息分类存储

主题是多级形式的，消息按第一级即父 topic 分类存储至 kafka 队列。



1.3.3 数据生态

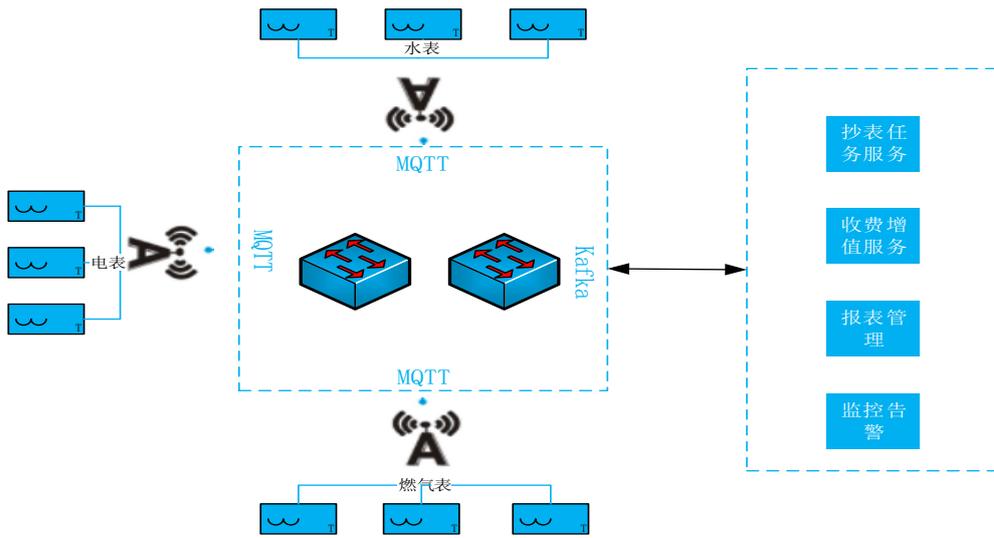


1.4 应用场景

广泛应用于移动互联网以及物联网领域，车联网、工业设备、智能家居、即时聊天等多种应用场景。

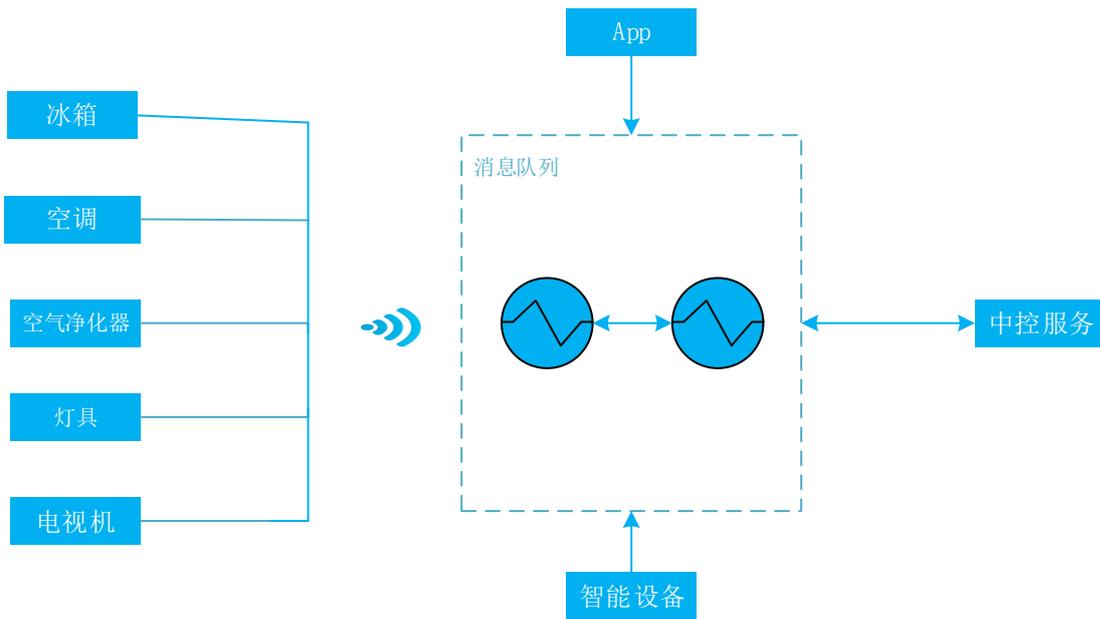
场景一

智能抄表。电表、水表、燃气表已是每个家庭必备，集成消息队列 MQTT 系统后，再无需进行人工抄表，所有住户的用电、用水、燃气、供暖数据都可由数据后台一键抄取，省时省力避免上门催收的尴尬，助力能源管理部门提供更优质服务



场景二

智能家居，通过门监测住户出入行为，同时结合实时气温与时间，控制空调、灯具、电视机、音箱等的启动和关闭



2. 快速入门

2.1 创建实例

- 1、登入天翼云门户
- 2、选择相应的资源池
- 3、进入服务列表，选择分布式缓存，进入分布式消息服务 MQTT 控制台实例列表页点击“订购实例”进入开通页进行实例创建。

2.2 连接实例

- 1、创建用户名和密码，需要数据存储创建父主题，需要分组管理创建 GroupId；
- 2、对用户进行主题授权
- 3、公网接入若未绑定弹性公网 ip 需先进行购买弹性 ip 并进行绑定
- 4、终端设备使用 MQTT 客户端通过终端连接地址接入；
- 5、云端应用使用 KAFKA 客户端通过服务端连接地址接入。
- 6、支持 SSL

2.3 使用须知

2.3.1 公网接入

公网访问需购买弹性 ip，提供以下两种方式实现：

- 1、开通时，开通页面选择已购买未使用的弹性 ip，自动绑定
- 2、先开通 MQTT 实例，后通过控制台提供的公网绑定入口进行绑定操作

弹性 ip 带宽大小计算规则可参照：带宽= 报文大小*TPS*120%，建议按 120%购买，应对突发流程。

如基础版规格，报文大小 1KB，TPS 2W/s，则带宽=20000*1000*8=160Mb/s，建议 200Mb/s

2.3.2 消息数据存储

终端消息数据按父 topic 存储至 kafka 队列，需先在控制台创建父 topic；对未创建父 topic 的消息可正常收发，但不会存储至 kafka 队列。

Kafka 存储内容格式：{

"clientId": 设备 clientId,

"topic": 主题,

"payload": 消息内容,

"ts": 发送的时间戳

}

2.3.3 会话机制

终端 clean session=true,断线后会话信息清除，再次上线后之前所有的订阅关系以及离线消息丢失。clean session=false 断开连接的情况下，MQTT Broker 也会为断连客户端保存一个会话，默认 2 小时，超期未重连订阅关系清除；对于 clean session=false 的客户端断线重连后可接收 Qos>0 的离线消息。对于客户端因网络等各种原因断线，需要加上重连和订阅关系重新订购机制。

2.3.4 离线消息

对于 clean session=false 的客户端在未超出会话失效期断线重连后可接收 Qos>0 的离线消息

2.3.5 系统主题

系统主题	说明
mq2mqtt	用于云端服务向终端发送消息。发往该主题消息会转发至 MQTT Broker 实现云端与移动端互通
mqtt-device-connect	设备上线主题，内容

系统主题	说明
	{"clientid":客户端 ID, "ts":上线时间戳 }
mqtt-device-disconnect	设备下线主题, 内容 {"clientid":客户端 ID, "ts":下线时间戳 }

2.3.6 名称解释

名词	说明
终端连接地址	即 MQTT Broker 端接入地址, 设备端使用;
服务端连接地址	即 kafka 集群连接地址, 云端应用服务使用
订阅关系	终端设备每订阅一个主题即一个订阅关系

2.3.7 使用限制

限制项	限制值	说明
最大报文	128KB	允许的 MQTT 报文最大长度
Topic 和 Client ID 可用字符	仅限数字 0-9, 字母 a-z 或 A-Z, 和符号 “-”	Client ID 和 MQTT 的 Topic 不允许使用其他非常规字符, 否则可能导致无法连接和收发消息

限制项	限制值	说明
Client ID 长度	128 个字符	Client ID 长度不得超过该限制，否则会导致连接被断开。
用户消息收发 TPS		根据实例规格不同进行限制
用户连接数		根据实例规格不同进行限制
最大 QoS 等级	2	请合理使用 QoS>0 等级，以免避免内存超额。
单个设备订阅 Topic 数量	无	
每个持久化会话离线消息存储数量	10 条	服务端会限制每个持久化会话的离线消息数量。超过该限制后，服务端会从最早的消息开始清理。因此，请合理使用持久化订阅模式，以免避免内存超额。
顺序消息		指每个客户端发送消息的顺序性
Kafka 分区数	3	

2.4 开发指南

2.4.1 SDK 支持

分布式消息服务 MQTT 支持标准的 MQTT 协议，理论上适配所有的 MQTT 客户端 SDK。

推荐对应的第三方 SDK 如下表：

语言/平台	推荐的第三方 SDK
Java	Eclipse Paho SDK
iOS	MQTT-Client-Framework
Android	Eclipse Paho SDK
JavaScript	Eclipse Paho JavaScript
Python	Eclipse Paho Python SDK
C	Eclipse Paho C SDK
C#	Eclipse Paho C# SDK
Golang	Eclipse Paho Golang SDK
Node.js	MQTT-JS
PHP	Mosquitto-PHP

2.4.2 主题规则

主题形式：父 topic/子级 topic1/子级 topic2…。(父 topic 需要先创建)

使用 MQTT 消息队列发消息,会把消息以父 Topic 主题分类保存在 kafka 上,应用服务可通过 kafka 客户端以父 Topic 为主题消费消息。

云端应用服务统一发送到 kafka topic 为 mq2mqtt 的主题队列上, **移动端 topic**、会话属性 **Qos** 和 **cleansession** 保存在 **Record Header** 中, MQTT 设备通过订阅**移动端 topic**, 实现云端到移动端通讯。

Kakfa header 与 **mqtt 属性** 映射如下表

Kafka Header	MQTT 属性
qoslevel	Qos
cleanSession	cleanSession
mqttTopic	主题

2.4.3 消息收发

2.4.3.1 MQTT 客户端收发

使用 MQTT SDK 接入终端连接地址进行消息生产消费。

2.4.3.2 MQTT 发送 kafka 接收

终端设备使用 MQTT SDK 接入终端连接地址进行消息发布，云端应用服务使用 kafka sdk 接入服务端连接地址按父主题进行数据消费。

2.4.3.3 MQTT 发送顺序消息 kafka 接收顺序消息

创建父主题，类型分区顺序，父主题以 orderMsg2mq-开头。

终端设备使用 MQTT SDK 接入终端连接地址进行消息发布，云端应用服务使用 kafka sdk 接入服务端连接地址按父主题进行分区顺序数据消费

2.4.3.4 Kafka 发送 MQTT 接收

云端应用服务使用 kafka sdk 接入服务端连接地址往系统主题 mq2mqtt 下发指令，终端设备使用 MQTT SDK 接入终端连接地址接收；

示例：

```
public static void main(String[] args) {
    Properties properties = new Properties();
    properties.put("bootstrap.servers", "xxx.xx.xxx:port");
    properties.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer");
    properties.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer");
    Producer<String, String> producer = null;
    try {
        producer = new KafkaProducer<String, String>(properties);
        for (int i = 0; i < 100000; i++) {
            String msg = "Message 1" ;
            Headers headers=new RecordHeaders();
            headers.add(new RecordHeader("qoslevel",String.valueOf(1).getBytes()));
            headers.add(new RecordHeader("cleanSession","false".getBytes()));
            headers.add(new RecordHeader("mqttTopic","/test3/xxx".getBytes()));
            producer.send(new ProducerRecord<String, String>(topic:"mq2mqtt", partition: null, key: null,msg,headers));
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        producer.close();
    }
}
```

3. 控制台使用

3.1 实例生命周期管理

进入分布式消息服务 MQTT 控制台实例列表页，可进行实例的开通、续费、退订；

实例到期不可访问，续费可恢复；退订后实例默认保留半个月后进行销毁。

3.2 实例详情

实例详情包含基本的实例规格信息、连接信息、当前性能指标信息以及各 MQTT Broker 节点信息。

规格信息					
连接数上限: 100000	TPS上限: 20000				
订阅关系数上限: 100000					
基本信息					
实例ID: 1abc223a12ad4dbeb1d7b1696343b81a	实例名称: MQTT-2af8				
实例类型: 基础版	专有网络: vpc-ipv63				
子网: subnet-245b	安全组: default				
创建时间: 2020-09-17 13:32:42	服务端连接地址: 192.168.0.144:9092,192.168.0.68:9092,192.168.0.183:9092				
终端连接地址: tcp://192.168.0.67:1883	终端SSL连接地址: tcp://192.168.0.67:8085 ↓				
终端PV6连接地址: tcp://240e:980:2f00:71:364:1883	终端PV6 SSL连接地址: tcp://240e:980:2f00:71:364:8085				
公网IP: 未绑定					
会话信息					
主题数: 3040	连接数: 3003				
订阅关系数: 5994	pub Tps(/min): 10477				
	sub Tps(/min): 20909				
MQTT节点信息					
节点名称	连接数	历史最大连接数	订阅关系数	订阅者数	节点状态
MQTT2@192.168.0.183	2	2	0	0	● 运行中
MQTT3@192.168.0.68	2999	2999	5994	5994	● 运行中
MQTT1@192.168.0.144	2	3	0	0	● 运行中

3.3 消息存储

消息存储页提供父主题配置，主题消息堆积信息，以及 kafka 集群实时状态。

主题管理 Kafka集群状态

系统主题

测试

主题名	描述	操作
mqtt-device-connect	connect主题记录设备每次连接MQTT Broker时的信息	详情
mqtt-device-disconnect	disconnect主题存储设备每次断开MQTT Broker连接时的信息	详情
mq2mqtt	mq2mqtt主题用于云端向移动端发送消息，云端应用服务往该主题发送消息会转发至MQTT Broker实现云端与移动端互通	详情

移动端父主题

新增

主题名	消息类型	分区数	备注	创建时间	操作
test001	普通消息	3	1	2020-09-17 01:01:25	删除 分区状态 详情

创父主题分两类：普通消息主题和分区顺序消息主题；分区顺序只是每个客户端发送消息的顺序性。创建的时候可指定消息类型

新增主题 ×

* 主题名

* 消息类型

* 备注

普通消息

分区顺序消息

主题消息堆积情况，消息堆积数=截止 offset-当前 offset

分区状态

×

分区ID	消息数	当前offset	起始offset	截止offset
0	3490489	0	0	3490489
2	3492411	0	0	3492411
1	3492780	0	0	3492780

关闭

主题状态

主题状态 ×

Metrics	mean	1 min	5 min	15 min
Messages in /sec	0.00	0.00	0.00	0.00
Bytes out /sec	0.00	0.00	0.00	0.00
Bytes in /sec	0.00	0.00	0.00	0.00
Failed fetch request /sec	0.00	0.00	0.00	0.00
Failed produce request /sec	0.00	0.00	0.00	0.00
Bytes rejected /sec	0.00	0.00	0.00	0.00

关闭

Kafka 集群状态信息

Kafka节点状态			
broker id	主机IP	端口	节点状态
1	192.168.0.144	9092	● 运行中
2	192.168.0.183	9092	● 运行中
3	192.168.0.68	9092	● 运行中

Kafka集群状态				
Metrics	mean	1 min	5 min	15 min
Messages in /sec	241.11	208.99	220.77	229.92
Bytes out /sec	38104.53	31109.34	33258.89	34817.85
Bytes in /sec	39849.45	35167.73	36807.22	38348.17
Failed fetch request /sec	0.00	0.00	0.00	0.00
Failed produce request /sec	0.00	0.00	0.00	0.00
Bytes rejected /sec	0.00	0.00	0.00	0.00

3.4 认证授权

终端设备连接 MQTT 队列需要先创建用户密码并对用户名进行主题授权，客户端方可正常收发。

← 基础版-1

实例详情 消息存储 认证授权 Group管理 连接查询 轨迹查询 资源报表 监控告警

认证用户 (用于客户端连接MQTT Broker)

新增

用户名	备注	创建时间	操作
test	1	2020-09-26 01:18:53	删除 授权 已授权列表

共 1 条 10条/页 < 1 > 前往 1 页

授权页面，主题权限包含 3 种：pub、sub、pubsub，支持通配符，#代表所有主题。

用户授权

认证用户名:

* 主题:

* 动作:

*注: 支持通配符, #代表所有主题

3.5 连接查询

提供按 Client ID 查询客户端的连接信息和订阅关系，进行剔除（下线）和主题退订操作；

连接查询

ClientID:

连接详情

属性	值
ClientID	client_00711112114
客户端IP	172.31.208.105
连接协议	MQTT
连接时间	2020-09-18 18:18:20
cleanSession	false
认证用户	system
当前订阅数	1
发送消息数	0
接收消息数	0
是否在线	true 删除
连接断开时间	

订阅关系

主题	QOS	操作
testaaa/test12	0	退订

3.6 设备轨迹

按 clientId 按天维度查询设备轨迹。

← MQTT-2af8

实例详情 消息存储 MQTT用户 Group管理 连接查询 **轨迹查询** 资源报表 监控告警

设备轨迹

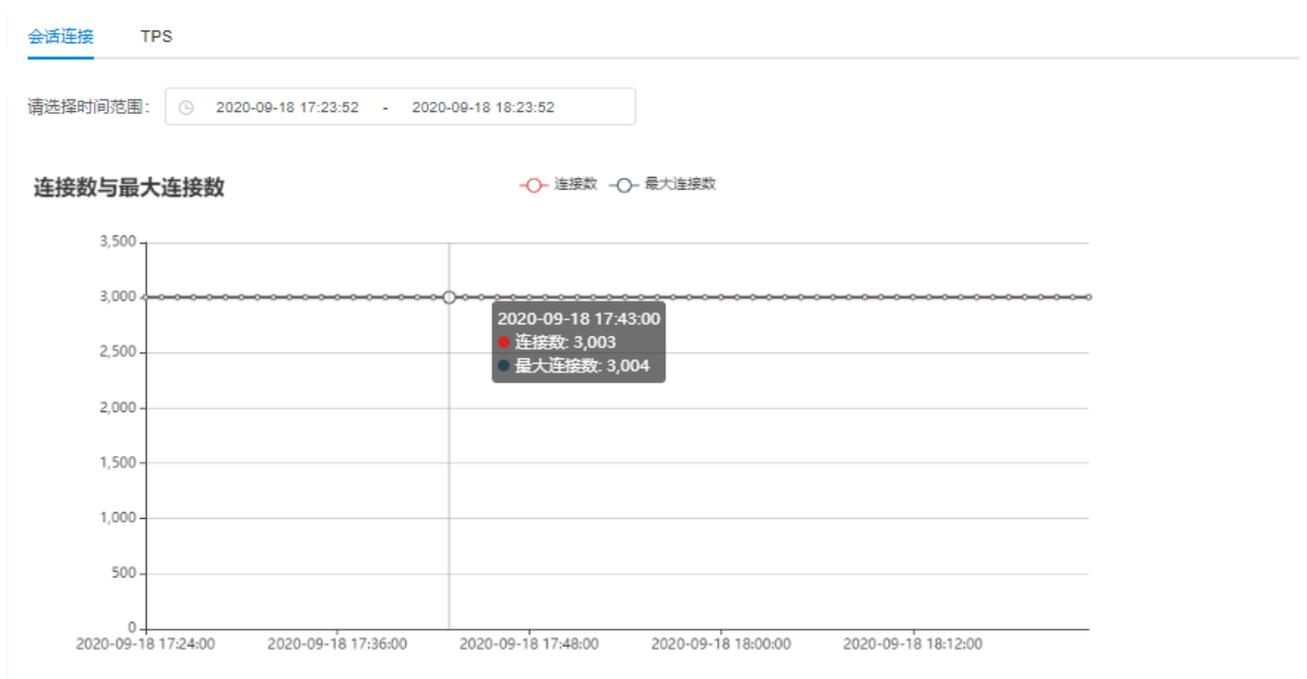
Client ID: 时间范围:

时间	Client ID	动作
2020-09-18 18:08:11	client_00711112114	connect
2020-09-18 18:08:15	client_00711112114	subscribed
2020-09-18 18:16:06	client_00711112114	disconnect
2020-09-18 18:16:16	client_00711112114	subscribed
2020-09-18 18:16:16	client_00711112114	resumed
2020-09-18 18:16:16	client_00711112114	connect

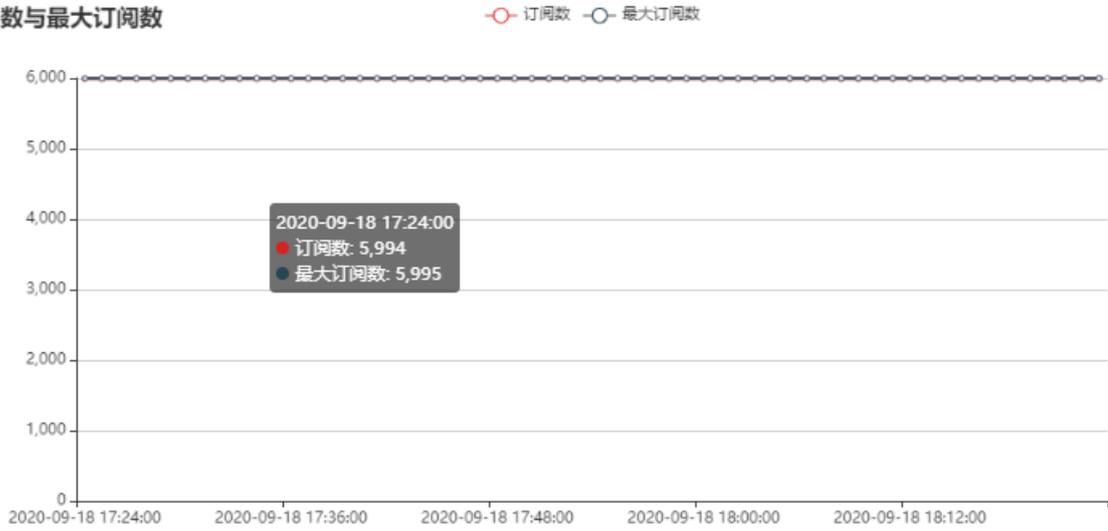
字段	说明
时间	动作发生的日期时间
Client ID	设备 id
动作	connect: 建立连接 subscribed: 主题订阅 resumed: 会话恢复 disconnect: 连接断开 discarded: 会话丢弃, cleanSession=true 客户端连接如果服务端中已存在该客户端的会话, 丢弃旧的会话 takeovered: 会话接管, cleanSession=false 客户端连接如果服务端中已存在该客户端的会话, 旧的会话被该连接接管

3.7 资源报表

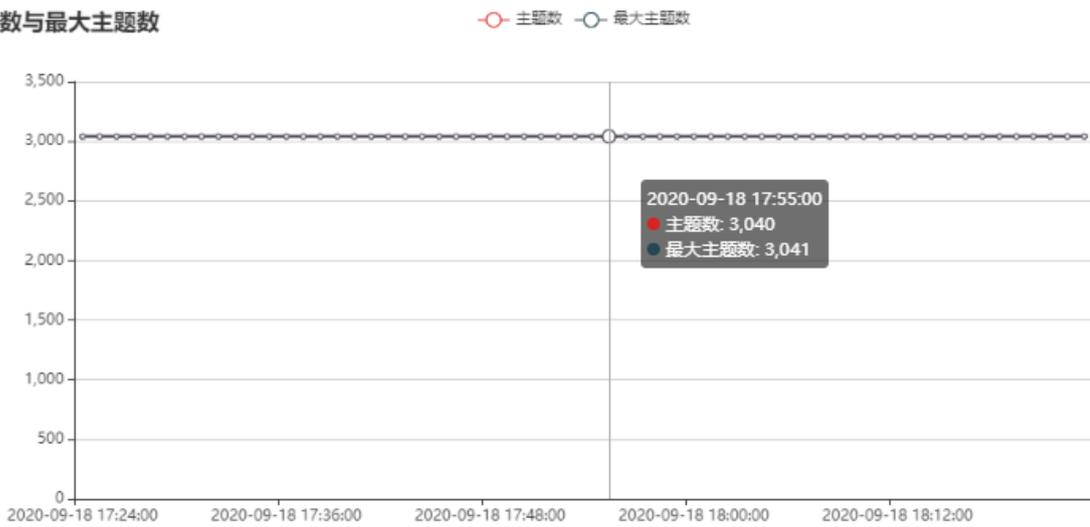
提供连接数、订阅关系数、主题数、收发 tps 历史指标数据的查询

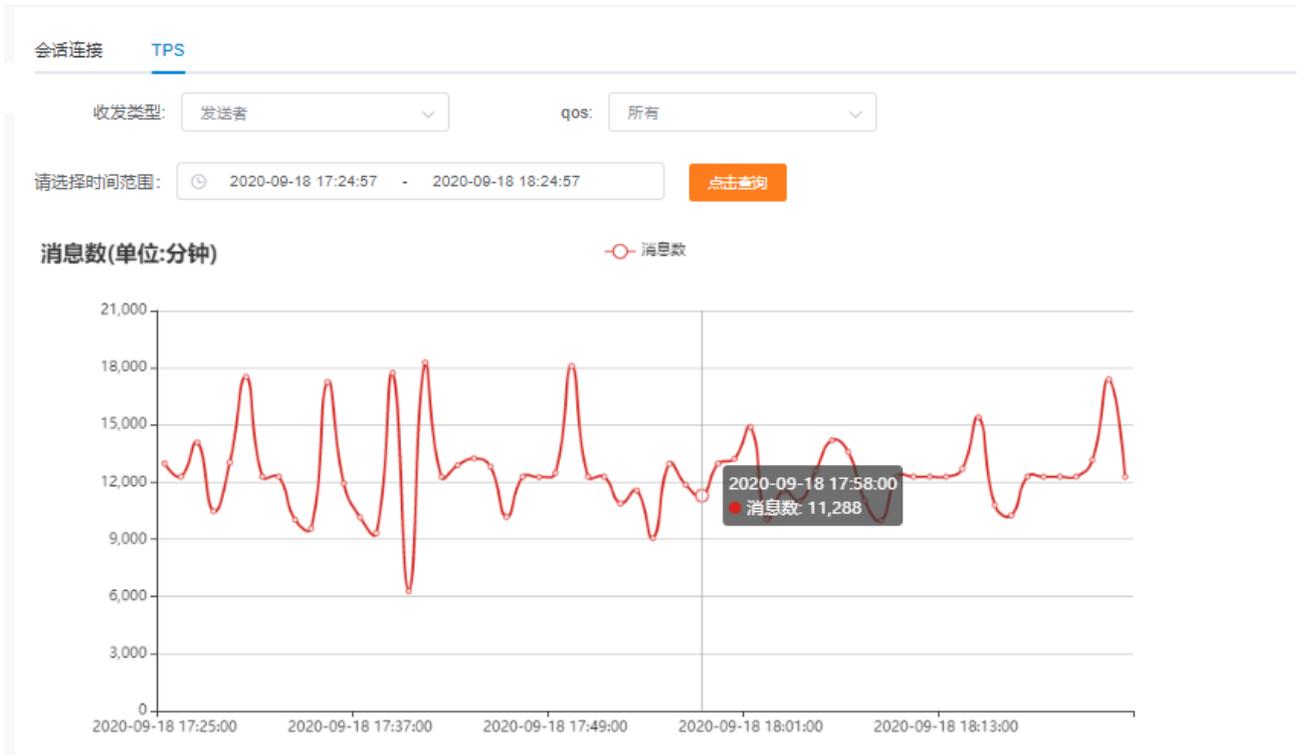


订阅数与最大订阅数



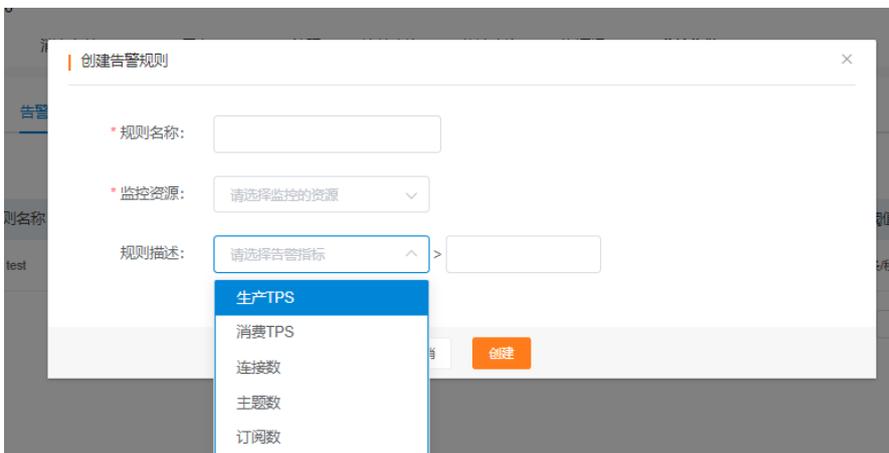
主题数与最大主题数





3.8 监控告警

提供连接数、订阅关系数、主题数、tps 自定义阈值配置，并在指标阈值触发时进行告警显示，记录了每个阈值告警的开始时间和结束时间。



告警情况 告警规则

告警资源	告警指标	当前值	告警阈值	状态	开始时间	结束时间	持续分钟数	操作
实例	生产TPS	12826	1	告警中	2020-09-18 05:27:00	-	780	忽略

共 1 条 10条/页 < 1 > 前往 1 页

4. 常见问题

4.1 终端设备一直处于断线状态

检查终端客户端使用是否进行了重连机制。

4.2 终端断线重连后一直未收到消息

对于 `cleanSession=true` 的客户端，断线重连后需要重新主题订阅；

对于 `cleanSession=false` 的客户端，断线在 2h 时间范围内重新成功无需进行主题订购，超过也需主题重新订阅。

4.3 客户端连接异常断开

- 1、不同的客户端使用相同的 Client ID
- 2、报文是否超过最大限制
- 3、权限不匹配

4.4 MQTT Broker 连接数为 0

MQTT Broker 节点宕机重连，客户端重连至其他节点，正常现象。

4.5 离线消息缺失

对于 `clean session=false` 的客户端在未超出会话失效期断线重连后可接收 `Qos>0` 的离线消息，离线消息限制不超过 10 条。