

分布式消息服务 Kafka

# 用户指南

文档版本 08  
发布日期 2025-09-03



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

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

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

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

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

# 目录

<b>1 产品简介</b>	<b>1</b>
1.1 什么是分布式消息服务 Kafka	1
1.2 产品优势	1
1.3 典型应用场景	2
1.4 Kafka 实例规格	4
1.4.1 Kafka 集群实例	4
1.5 与 RabbitMQ、RocketMQ 的差异	7
1.6 与开源 Kafka 的差异	8
1.7 约束与限制	10
1.8 与其他云服务的关系	13
1.9 Kafka 相关概念	14
1.10 权限管理	16
<b>2 快速入门</b>	<b>21</b>
2.1 快速连接 Kafka 并生产消费消息	21
<b>3 Kafka 业务使用流程</b>	<b>30</b>
<b>4 权限管理</b>	<b>32</b>
4.1 创建用户并授权使用 DMS for Kafka	32
<b>5 购买 Kafka 实例</b>	<b>36</b>
<b>6 配置 Topic</b>	<b>49</b>
6.1 创建 Kafka Topic	49
6.2 设置 Kafka Topic 权限	55
6.3 Topic 管理	61
6.3.1 查看 Kafka Topic 详情	61
6.3.2 查看 Kafka Topic 日志	63
6.3.3 修改 Kafka Topic 配置	66
6.3.4 修改 Kafka Topic 分区数	69
6.3.5 修改 Kafka Topic 副本数	74
6.3.6 导出 Kafka Topic 列表	77
6.3.7 修改 Kafka 分区平衡	78
6.3.8 配置 Kafka 自动创建 Topic	88
6.3.9 删除 Kafka Topic	89

<b>7 连接实例</b>	<b>92</b>
7.1 配置 Kafka 网络连接	92
7.1.1 连接 Kafka 网络要求	92
7.1.2 使用 VPCEP 实现跨 VPC 访问 Kafka	93
7.2 配置 Kafka 访问控制	99
7.2.1 配置 Kafka ACL 用户	99
7.3 配置 Kafka 客户端	102
7.3.1 Kafka 客户端参数配置建议	102
7.3.2 Kafka 客户端使用建议	106
7.4 使用客户端连接 Kafka（明文接入）	107
7.5 使用客户端连接 Kafka（密文接入）	109
7.6 使用控制台连接 Kafka	116
<b>8 管理消息</b>	<b>117</b>
8.1 查看 Kafka 消息	117
8.2 修改 Kafka 消息老化时间	120
8.3 删除 Kafka 消息	122
<b>9 管理消费组</b>	<b>125</b>
9.1 创建 Kafka 消费组	125
9.2 查看 Kafka 消费组信息	126
9.3 查看 Kafka 消费者信息	128
9.4 查看和重置 Kafka 消费进度	131
9.5 查看 Kafka 重平衡日志	135
9.6 导出 Kafka 消费组列表	138
9.7 删除 Kafka 消费组	139
9.8 取消 Kafka 消费组和 Topic 的订阅关系	141
<b>10 配置流控</b>	<b>142</b>
10.1 配置 Kafka 流控	142
10.2 查看 Kafka 流控统计	146
<b>11 管理实例</b>	<b>148</b>
11.1 查看和修改 Kafka 实例基本信息	148
11.2 查看 Kafka 磁盘使用量	151
11.3 查看 Kafka 实例后台任务	152
11.4 查看 Kafka 生产消费示例代码	153
11.5 修改 Kafka 实例配置参数	154
11.6 配置 Kafka 实例标签	158
11.7 配置 Kafka 实例回收站策略	160
11.8 导出 Kafka 实例列表	162
11.9 重启 Kafka 实例	163
11.10 删除 Kafka 实例	163
11.11 管理 Kafka Manager	164
11.11.1 访问 Kafka Manager	164

11.11.2 重置 Kafka Manager 密码.....	170
11.11.3 重启 Kafka Manager.....	171
11.11.4 关闭 Kafka Manager.....	172
<b>12 变更实例规格.....</b>	<b>173</b>
12.1 变更 Kafka 集群实例规格.....	173
<b>13 迁移数据.....</b>	<b>179</b>
13.1 Kafka 数据迁移概述.....	179
13.2 使用 MirrorMaker 跨集群同步数据.....	182
<b>14 申请扩大 Kafka 配额.....</b>	<b>186</b>
<b>15 查看监控指标与配置告警.....</b>	<b>187</b>
15.1 查看 Kafka 监控数据.....	187
15.2 Kafka 支持的监控指标.....	187
15.3 配置 Kafka 监控告警.....	196
<b>16 查看 Kafka 审计日志.....</b>	<b>199</b>
<b>17 常见问题.....</b>	<b>203</b>
17.1 实例问题.....	203
17.1.1 为什么可用区不能选择 2 个? .....	203
17.1.2 创建实例时为什么无法查看子网和安全组等信息? .....	203
17.1.3 如何选择 Kafka 实例的存储空间? .....	203
17.1.4 Kafka 实例的超高 IO 和高 IO 如何选择? .....	204
17.1.5 如何选择 Kafka 实例存储容量阈值策略? .....	204
17.1.6 Kafka 服务端支持版本是多少? .....	204
17.1.7 Kafka 实例的 ZK 地址是什么? .....	204
17.1.8 创建的 Kafka 实例是集群模式么? .....	204
17.1.9 Kafka 实例是否支持修改访问端口? .....	204
17.1.10 Kafka 实例的 SSL 证书有效期多长? .....	205
17.1.11 如何将 Kafka 实例中的数据同步到另一个 Kafka 实例中? .....	206
17.1.12 Kafka 实例的 SASL_SSL 开关如何修改? .....	206
17.1.13 SASL 认证机制如何修改? .....	206
17.1.14 修改企业项目, 是否会导致 Kafka 重启? .....	207
17.1.15 Kafka 服务和 ZK 是部署在相同的虚拟机中, 还是分开部署? .....	207
17.1.16 Kafka 支持哪些加密套件? .....	207
17.1.17 购买实例时选择的单 AZ, 怎样可以扩展为多 AZ? .....	207
17.1.18 Kafka 是否支持跨 AZ 容灾? 已经购买的实例在哪里查看是否为跨 AZ? .....	207
17.1.19 Kafka 支持磁盘加密吗? .....	207
17.1.20 Kafka 实例创建后, 能修改 VPC 和子网吗? .....	207
17.1.21 有没有 Kafka Stream 的案例? .....	208
17.1.22 Kafka 实例版本可以升级吗? .....	208
17.1.23 配置了“自动删除”策略, 磁盘容量到达 95%时, 为什么没有自动删除最早的消息? .....	208
17.2 实例规格变更问题.....	208

17.2.1 Kafka 扩容会影响业务吗? .....	208
17.2.2 扩容过程中涉及数据迁移吗? .....	209
17.2.3 扩容过程中生产消息失败.....	209
17.2.4 扩容失败提示资源不足.....	210
17.3 连接问题.....	210
17.3.1 选择和配置安全组.....	210
17.3.2 Kafka 实例是否支持公网访问? .....	212
17.3.3 Kafka 实例的连接地址默认有多少个? .....	212
17.3.4 是否支持跨 Region 访问? .....	213
17.3.5 Kafka 实例是否支持跨 VPC 访问? .....	213
17.3.6 Kafka 实例是否支持不同的子网? .....	213
17.3.7 Kafka 是否支持 Kerberos 认证, 如何开启认证? .....	213
17.3.8 Kafka 实例是否支持无密码访问? .....	213
17.3.9 Kafka 支持服务端认证客户端吗? .....	213
17.3.10 连接开启 SASL_SSL 的 Kafka 实例时, ssl truststore 文件可以用 PEM 格式的吗? .....	213
17.3.11 下载的证书 JKS 和 CRT 有什么区别? .....	214
17.3.12 Kafka 支持哪个版本的 TLS? .....	214
17.3.13 Kafka 实例连接数有限制吗? .....	214
17.3.14 客户端单 IP 连接的个数为多少? .....	214
17.3.15 Kafka 实例的内网连接地址可以修改吗? .....	214
17.3.16 不同实例中, 使用的 SSL 证书是否一样? .....	214
17.3.17 为什么不建议使用 Sarama 客户端收发消息? .....	215
17.4 Topic 和分区问题.....	216
17.4.1 Kafka 实例的 Topic 数量是否有限制? .....	216
17.4.2 为什么限制 Topic 的总分区数? .....	216
17.4.3 Kafka 支持减少分区数吗? .....	217
17.4.4 Kafka 实例创建 Topic 失败.....	217
17.4.5 Kafka 实例支持批量导入 Topic 功能么? 或者是自动生成 Topic 功能? .....	217
17.4.6 Kafka 实例是否支持查看单个 Topic 占用磁盘空间? .....	218
17.4.7 Topic 是否支持 ACL 权限配置? .....	218
17.4.8 消息被消费后, 没有删除, 导致 Kafka 存储空间占满? .....	218
17.4.9 如何扩总分区? .....	218
17.4.10 修改自动创建 Topic 的配置, 会触发重启吗? .....	218
17.4.11 消费者消费 Topic 失败, 提示没有权限? .....	218
17.4.12 分区数可以单独买吗? .....	219
17.5 消费组问题.....	219
17.5.1 Kafka 实例是否需要创建消费组、生产者和消费者? .....	219
17.5.2 如果消息组中没有在线的消费者(如 empty 状态), 是否 14 天后会自动被删除? .....	219
17.5.3 为什么删除消费组不生效? .....	220
17.5.4 实例消费正常, 为什么看不到消费者? .....	220
17.5.5 Kafka 可以删除消费组下不用的 Topic 吗? .....	220
17.5.6 为什么创建的消费组不见了? .....	220

17.6 消息问题.....	220
17.6.1 Kafka 生产消息的最大长度是多少? .....	220
17.6.2 为什么 Kafka 实例常常发生 Rebalance (重平衡), 导致消息拉取失败? .....	221
17.6.3 为什么 Console 页面上, 消息查询查不到消息? .....	221
17.6.4 Kafka 消息堆积了怎么办? .....	222
17.6.5 消息超过老化时间, 消息仍存在的原因.....	222
17.6.6 Kafka 实例是否支持延迟消息? .....	222
17.6.7 如何查看堆积消息数? .....	222
17.6.8 为什么消息创建时间显示 1970? .....	223
17.6.9 如何修改 message.max.bytes 参数? .....	223
17.6.10 为什么 offset 不连续? .....	223
17.7 Kafka Manager 问题.....	224
17.7.1 登录 Kafka Manager 的账号是否可以设置为只读账号? .....	224
17.7.2 登录到 Kafka Manager 页面, 为什么获取不到节点信息? .....	224
17.7.3 Yikes! Insufficient partition balance when creating topic : projectman_project_enterprise_project Try again.....	224
17.7.4 Kafka Manager 能否查询到消息的正文? .....	224
17.7.5 Kafka Manager WebUI 的端口能否修改? .....	224
17.7.6 在 Kafka Manager 上支持修改 Topic 的哪些属性? .....	224
17.7.7 Kafka Manager 和云监控显示的信息不一致.....	225
17.7.8 Kafka Manager 如何修改 Topic 的分区 Leader? .....	225
17.7.9 实例版本在控制台和 Kafka Manager 上显示不一致? .....	228
17.7.10 为什么实例中存在默认名为__trace 和__consumer_offsets 的 Topic? .....	228
17.7.11 客户端删除消费组后, 在 Kafka Manager 中仍可以看到此消费组? .....	229
17.8 监报告警问题.....	229
17.8.1 云监控无法展示 Kafka 监控数据.....	229
17.8.2 Kafka 监控显示消息堆积数跟实例里的消息数不一致? .....	230
17.8.3 Kafka 的消费组删除了, 怎么监控页面还可以看到这个消费组? .....	230
17.8.4 为什么磁盘读/写流量、磁盘平均读/写操作耗时和 CPU 使用率出现明显波动? .....	230
17.8.5 为什么 JVM 堆内存使用率出现明显波动? .....	230
17.8.6 生产消息数不为 0, 为什么生产速率为 0? .....	230
17.8.7 消费进度显示有堆积消息, 为什么监控中显示消息堆积数为 0? .....	231
<b>18 故障排除.....</b>	<b>232</b>
18.1 Kafka 连接问题排查和解决.....	232
18.2 消息从生产到消费时延高达 6 分钟.....	234
18.3 生产消息失败.....	236
18.4 删除 Topic 失败.....	236
18.5 在 Windows 服务器中无法登录 Kafka Manager.....	236
18.6 生产或消费消息时, 报 Topic {{topic_name}} not present in metadata after 60000 ms 错误.....	237
18.7 Flink 1.15 消费进度提交失败.....	238
<b>A 修订记录.....</b>	<b>239</b>

# 1 产品简介

## 1.1 什么是分布式消息服务 Kafka

Kafka是一个拥有高吞吐、可持久化、可水平扩展，支持流式数据处理等多种特性的分布式消息流处理中间件，采用分布式消息发布与订阅机制，在日志收集、流式数据传输、在线/离线系统分析、实时监控等领域有广泛的应用。

分布式消息服务Kafka是一款基于开源社区版Kafka提供的消息队列服务，向用户提供计算、存储和带宽资源独占式的Kafka专享实例。使用分布式消息服务Kafka，资源按需申请，即买即用，您将有更多精力专注于业务快速开发，不用考虑部署和运维。

### 关于 Kafka 的帮助手册阅读指引

受限于篇幅，我们提供的Kafka帮助手册重点描述产品相关的内容，以及与开源社区版Kafka的差异部分，例如Kafka的产品规格、控制台操作、客户端对接等。

如果您需要了解Kafka入门知识或消息生产、消费等方面的技术细节，请查阅[Kafka官网资料](#)。

## 1.2 产品优势

分布式消息服务Kafka完全兼容开源社区版本，旨在为用户提供便捷高效的消息队列。业务无需改动即可快速迁移上云，为您节省维护和使用成本。

- 一键式部署，免去集群搭建烦恼  
您只需要在实例管理界面选好规格配置，提交订单。后台将自动创建部署完成一整套Kafka实例。
- 兼容开源，业务零改动迁移上云  
兼容社区版Kafka的API，具备原生Kafka的所有消息处理特性。  
业务系统基于开源的Kafka进行开发，只需加入少量认证安全配置，即可使用分布式消息服务Kafka，做到无缝迁移。

#### 📖 说明

Kafka实例兼容开源社区Kafka 1.1.0、2.3.0、2.7、3.x版本。在客户端使用上，推荐使用和服务端版本一致的版本。

- 安全保证  
独有的安全加固体系，提供业务操作云端审计，消息传输加密等有效安全措施。在网络通信方面，除了提供SASL（Simple Authentication and Security Layer）认证，还借助虚拟私有云（VPC）和安全组等加强网络访问控制。
- 数据高可靠  
Kafka实例支持消息持久化，多副本存储机制。副本间消息同步、异步复制，数据同步或异步落盘多种方式供您自由选择。
- 集群架构与跨AZ部署，服务高可用  
Kafka后台为多集群部署，支持故障自动迁移和容错，保证业务的可靠运行。Kafka实例支持跨AZ部署，代理部署在不同的AZ，进一步保障服务高可用。不同AZ之间基于Kafka ISR（in-sync replica）进行数据同步，Topic需要选择数据多副本并且将不同副本分布到不同的ISR上，在ISR正常同步状态下，故障RPO（Recovery Point Objective）趋近于0。
- 无忧运维  
云服务平台提供一整套完整的监控告警等运维服务，故障自动发现和告警，避免7\*24小时人工值守。Kafka实例自动上报相关监控指标，如分区数、主题数、堆积消息数等，并支持配置监控数据发送规则，您可以在第一时间通过短信、邮件等获得业务消息队列的运行使用和负载状态。
- 海量消息堆积与弹性扩容  
内建的分布式集群技术，使得服务具有高度扩展性。分区数可配置多达200个，存储空间、代理数量和代理规格支持弹性扩展，保证在高并发、高性能和大规模场景下的访问能力，轻松实现百亿级消息的堆积和访问能力。
- 多规格灵活选择  
Kafka实例的带宽与存储资源可灵活配置，并且自定义Topic的分区数、副本数。

## 1.3 典型应用场景

Kafka作为一款热门的消息队列中间件，具备高效可靠的消息异步传递机制，主要用于不同系统间的数据交流和传递，在企业解决方案、金融支付、电信、电子商务、社交、即时通信、视频、物联网、车联网等众多领域都有广泛应用。

对于同步链路，需要实时返回调用结果的场景，建议使用RPC（Remote Procedure Call）调用方案。

### 异步通信

将业务中属于非核心或不重要的流程部分，使用消息异步通知的方式发给目标系统，这样主业务流程无需同步等待其他系统的处理结果，从而达到系统快速响应的目的。

如网站的用户注册场景，在用户注册成功后，还需要发送注册邮件与注册短信，这两个流程使用Kafka消息服务通知邮件发送系统与短信发送系统，从而提升注册流程的响应速度。

图 1-1 串行发送注册邮件与短信流程



图 1-2 借助消息队列异步发送注册邮件与短信流程

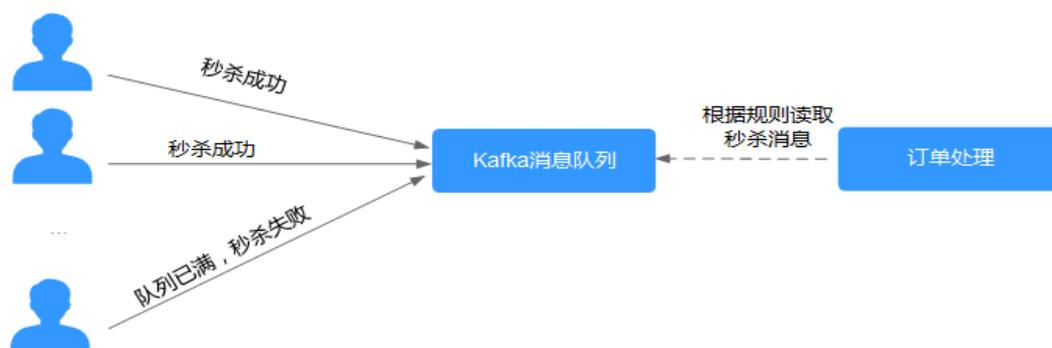


## 错峰流控与流量削峰

在电子商务系统或大型网站中，上下游系统处理能力存在差异，处理能力高的上游系统的突发流量可能会对处理能力低的某些下游系统造成冲击，需要提高系统的可用性的同时降低系统实现的复杂性。电商大促销等流量洪流突然来袭时，可以通过队列服务堆积缓存订单等信息，在下游系统有能力处理消息的时候再处理，避免下游订阅系统因突发流量崩溃。消息队列提供亿级消息堆积能力，3天的默认保留时长，消息消费系统可以错峰进行消息处理。

另外，在商品秒杀、抢购等流量短时间内暴增场景中，为了防止后端应用被压垮，可在前后端系统间使用Kafka消息队列传递请求。

图 1-3 消息队列应对秒杀大流量场景



## 日志同步

在大型业务系统设计中，为了快速定位问题，全链路追踪日志，以及故障及时预警监控，通常需要将各系统应用的日志集中分析处理。

Kafka设计初衷就是为了应对大量日志传输场景，应用通过异步方式将日志消息同步到消息服务，再通过其他组件对日志做实时或离线分析，也可用于关键日志信息收集进行应用监控。

日志同步主要有三个关键部分：日志采集客户端，Kafka消息队列以及后端的日志处理应用。

1. 日志采集客户端，负责用户各类应用服务的日志数据采集，以消息方式将日志“批量”、“异步”发送到Kafka客户端。  
Kafka客户端批量提交和压缩消息，对应用服务的性能影响非常小。
2. Kafka将日志存储在消息文件中，实现日志的持久化存储。
3. 日志处理应用，如Logstash，订阅并消费Kafka中的日志消息，最终供文件搜索服务检索日志，或者由Kafka将消息传递给Hadoop等其他大数据应用系统化存储与分析。

图 1-4 日志同步示意图



**说明**

上图中Logstash、ElasticSearch分别为日志分析和检索的开源工具，Hadoop表示大数据分析系统。

## 1.4 Kafka 实例规格

### 1.4.1 Kafka 集群实例

#### Kafka 集群实例规格

Kafka实例兼容开源Kafka 1.1.0、2.3.0、2.7和3.x，实例规格以实例提供的ECS规格分类，包括kafka.2u4g.cluster、kafka.4u8g.cluster、kafka.8u16g.cluster、kafka.12u24g.cluster和kafka.16u32g.cluster共5种类型。

**说明**

关于下表中TPS的说明：

- **TPS ( Transaction per second )**：在Kafka场景中，指每秒能写入到Kafka实例的最大消息数量。
- **TPS性能**：指以1KB大小的消息为例的每秒处理消息条数。
- **测试场景**：使用内网连接未开启SASL、磁盘类型为超高I/O的实例。

表 1-1 Kafka 集群实例规格

规格名称	代理个数范围	单个代理TPS	单个代理分区上限	单个代理建议消费组数	单个代理客户端总连接数上限	存储空间范围 ( GB )	单个代理流量规格 ( MB/s )
kafka.2u4g.cluster	3~30	30000	250	20	2000	300~300000	100
kafka.4u8g.cluster	3~30	10000	500	100	4000	300~600000	200
kafka.8u16g.cluster	3~50	15000	1000	150	4000	300~150000	375

规格名称	代理个数范围	单个代理TPS	单个代理分区上限	单个代理建议消费组数	单个代理客户端总连接数上限	存储空间范围 (GB)	单个代理流量规格 (MB/s)
kafka.12u24g.cluster	3~50	200000	1500	200	4000	300~1500000	625
kafka.16u32g.cluster	3~50	250000	2000	200	4000	300~1500000	750

## 实例规格和网络带宽说明

Kafka实例的网络带宽主要由以下两个部分组成：

1. 实例Broker对应的网络带宽
2. 实例Broker的磁盘所对应的带宽值（不同类型的磁盘对应的带宽值不同，具体参考：[如何选择磁盘类型](#)）

注意事项：

- Kafka默认情况下测试均为尾读场景（即仅消费最新生产的数据），而不是冷读场景（即从头开始消费历史数据的场景）。
- 老规格实例（即实例规格为100MB/s等）的带宽指的是实例所有Broker对应的网络带宽总和

新规格实例（即实例规格为kafka.2u4g.cluster等）的流量规格测算模型说明如下：

- 测试模型读写比例为1:1
- 默认Topic的副本数为3
- 实例网络总流量 = 单个代理流量规格 \* 代理数量
- 实例整体流量 = 业务流量 + 代理节点间数据复制流量

参考上述测算模型说明，假如当前规格为kafka.2u4g.cluster，单个代理流量规格为100MB/s，代理数量为3，实例网络总流量、最大读流量和最大写流量分别为多少？

1. 实例网络总流量 = 单个代理流量规格 \* 代理数量 = 100MB/s \* 3 = 300MB/s
2. 最大读流量 = 实例网络总流量 / 默认副本数 / 2 = 300MB/s / 3 / 2 = 50MB/s
3. 最大写流量 = 实例网络总流量 / 默认副本数 / 2 = 300MB/s / 3 / 2 = 50MB/s

## 新老规格对应关系

2种Kafka实例规格对比，新老规格的对应关系如[表1-2](#)所示。

表 1-2 Kafka 实例新老规格对应关系

老规格		对应的新规格	
规格类型	实例网络总流量	规格类型	实例网络总流量
300MB/s	300MB/s	kafka.2u4g.cluster * 3	300MB/s
600MB/s	600MB/s	kafka.4u8g.cluster * 3	600MB/s
1200MB/s	1200MB/s	kafka.4u8g.cluster * 6	1250MB/s

新老规格区别如下：

- 老规格使用的非独享资源，在高负载情况下容易出现资源抢占情况。新规格使用的独占资源，性能更优，性价比更高。
- 新规格支持最新的功能，例如：分区平衡等。
- 新规格支持规格灵活变更，例如：Broker规格的扩容。
- 新规格的磁盘大小选择更加灵活，磁盘大小不与实例规格进行绑定，仅与Broker数量相关。
- 新规格选择粒度更细，根据Broker规格和数量进行灵活的规格选择，并且最大规格可以达到10000MB/s以上。

## Kafka 实例规格参考

- kafka.2u4g.cluster，三个代理  
Kafka客户端连接数在6000以内，消费组个数在60个以内，业务TPS为90000以内时推荐选用。
- kafka.4u8g.cluster，三个代理  
Kafka客户端连接数在12000以内，消费组个数在300个以内，业务TPS为300000以内时推荐选用。
- kafka.8u16g.cluster，三个代理  
Kafka客户端连接数在12000以内，消费组个数在450个以内，业务TPS为450000以内时推荐选用。
- kafka.12u24g.cluster，三个代理  
Kafka客户端连接数在12000以内，消费组个数在600个以内，业务TPS为600000以内时推荐选用。
- kafka.16u32g.cluster，三个代理  
Kafka客户端连接数在12000以内，消费组个数在600个以内，业务TPS为750000以内时推荐选用。

## Kafka 实例的存储空间估算参考

Kafka实例支持多副本存储，存储空间用于存储消息（包括副本中的消息）、日志和元数据。您在创建Kafka实例，选择初始存储空间时，建议根据业务消息体积预估、副本数量以及预留磁盘大小选择合适的存储空间。每个Kafka代理会预留33GB的磁盘空间，用于存储日志和元数据。

例如：业务消息体积预估100GB，副本数为2，Kafka实例的代理数为3，则磁盘容量最少应为100GB\*2+33GB\*3=299GB。

Kafka实例支持对存储进行扩容，根据业务增长，随时扩容，节约成本。

## Kafka 实例 Topic 数量计算

Kafka实例对Topic分区数之和设置了上限，当达到上限之后，用户无法继续创建Topic。

所以，Topic数量和实例分区数上限、每个Topic的分区数有关，其中，每个Topic分区数可在创建Topic时设置，实例分区数上限参考表1-1。

**kafka.2u4g.cluster \* 3 broker实例的分区数上限为750。**

- 如果该实例下每个Topic的分区个数都为3，则Topic个数为750/3=250个。
- 如果该实例下每个Topic的分区个数都为1，则Topic个数为750/1=750个。

## 1.5 与 RabbitMQ、RocketMQ 的差异

表 1-3 功能差异

功能项	RocketMQ	Kafka	RabbitMQ
优先级队列	不支持	不支持	支持。建议优先级大小设置在0-10之间。
延迟队列	支持	不支持	支持
死信队列	支持	不支持	支持
消息重试	支持	不支持	不支持
消费模式	支持客户端主动拉取和服务端推送两种方式。	客户端主动拉取。	支持客户端主动拉取和服务端推送两种模式。
广播消费	支持	支持	支持
消息回溯	支持	支持。Kafka支持按照offset和timestamp两种维度进行消息回溯。	不支持。RabbitMQ中消息一旦被确认消费就会被标记删除。
消息堆积	支持	支持。考虑吞吐因素，Kafka的堆积效率比RabbitMQ总体上要高。	支持
持久化	支持	支持	支持

功能项	RocketMQ	Kafka	RabbitMQ
消息追踪	支持	不支持	支持 RabbitMQ中可以采用Firehose或者rabbitmq_tracing插件实现，但开启rabbitmq_tracing插件会影响性能，建议只在定位问题过程中开启。
消息过滤	支持	支持	不支持，但可以自行封装。
多租户	支持	支持	支持
多协议支持	兼容RocketMQ协议。	只支持Kafka自定义协议。	RabbitMQ基于AMQP协议实现。
跨语言支持	支持多语言的客户端。	采用Scala和Java编写，支持多种语言的客户端。	支持多种语言的客户端。
流量控制	RocketMQ 5.x支持基于实例规格的流量控制。	支持client、user和Topic级别，通过主动设置可将流控作用于生产者或消费者。	RabbitMQ的流控基于Credit-Based算法，是内部被动触发的保护机制，作用于生产者层面。
消息顺序性	单队列（queue）内有序。	支持单分区（partition）级别的顺序性。	单线程发送、单线程消费并且不采用延迟队列、优先级队列等一些高级功能时，才能实现消息有序。
安全机制	支持SSL认证。	支持SSL、SASL身份认证和读写权限控制。	支持SSL认证。
事务性消息	支持	支持	支持

## 1.6 与开源 Kafka 的差异

分布式消息服务Kafka在兼容开源Kafka基础上，对版本特性做了一定程度的定制和增强，所以，除了拥有开源Kafka的优点，分布式消息服务Kafka提供了更多可靠、实用的特性。

表 1-4 分布式消息服务 Kafka 与开源 Kafka 的差异说明

对比类	对比项	分布式消息服务Kafka	开源Kafka
简单易用	立等可用	即开即用，可视化操作，自助创建，自动化部署，分钟级创建实例，立即使用，实时查看和管理消息实例。	<ul style="list-style-type: none"> <li>自行准备服务器资源，安装配置必要的软件并进行配置，等待时间长。</li> <li>易出错。</li> </ul>
	简单API	提供简单的实例管理 RESTful API，使用门槛低。	无
成本低廉	按需使用	提供多种规格，按需使用，支持一键式在线进行实例代理个数、磁盘存储空间和代理规格扩容。	搭建消息服务本身需要费用，而且即使没有使用，所占资源本身依旧要收费。
	完全托管	租户不需要单独采购硬件资源，直接使用就绪的服务，无需额外成本。	需要购买硬件资源，自行搭建整个消息服务，使用和维护成本高。
实践验证	成熟度高	经受电商网站大规模访问考验，并且已经在云服务平台许多产品中使用，广泛部署运行在分布于世界各地的电信级客户云业务系统里。满足严苛的电信级故障模式库标准。紧随社区主流版本，修复开源bug，持续上线新功能，进行版本升级。	使用开源软件成熟度低，无法保证关键业务，商业案例少；自研周期长，并需要长时间进行验证。
	能力强	100%兼容开源，支持一键扩容，深度优化开源代码提升性能和可靠性，支持消息查询等高级特性。	功能不完善，需额外投入进行开发。
稳定可靠	稳定高可用	支持跨AZ部署，提升可靠性。故障自动发现并上报告警，保证用户关键业务的可靠运行。	需要自己开发或基于开源实现，开发成本高昂，无法保证业务可靠运行。
	无忧运维	后台运维对租户完全透明，整个服务运行具有完备的监控和告警功能。有异常可以及时通知相关人员。避免7*24小时人工值守。	需要自行开发完善运维功能，尤其是告警及通知功能，否则只能人工值守。
	安全保证	VPC隔离，支持SSL通道加密。	需要自行进行安全加固。

## 1.7 约束与限制

本章节介绍分布式消息服务Kafka产品功能的约束和限制，您在使用Kafka实例时注意不要超过相应的约束和限制，以免程序出现异常。

### 须知

如果因为您的实例配置超过相应的约束和限制而导致的任何不稳定情况，不在SLA承诺和赔偿范围内。

### 实例

表 1-5 实例约束与限制

限制项	约束与限制
Kafka ZooKeeper	Kafka集群依赖ZooKeeper进行管理，开放ZooKeeper可能引发误操作导致业务受损，当前仅Kafka内部使用，不对外提供服务。
版本	<ul style="list-style-type: none"> <li>当前服务端版本为1.1.0、2.3.0、2.7和3.x。实例创建后，服务端版本不支持升级。</li> <li>兼容0.10以上的客户端版本，推荐使用和服务端一致的版本。</li> </ul>
登录Kafka节点所在机器	不能登录。
存储空间	<ul style="list-style-type: none"> <li>集群实例只支持扩大存储空间，不支持减小存储空间。</li> <li>扩容存储空间有次数限制，最多扩容20次。</li> </ul>
基准带宽/代理数量	集群实例只支持增加基准带宽/代理数量，不支持减小基准带宽/代理数量。
代理规格	<ul style="list-style-type: none"> <li>集群实例支持扩容代理规格，不支持缩容代理规格。</li> <li>若Topic为单副本，扩容期间无法对该Topic生产消息或消费消息，会造成业务中断。</li> <li>若Topic为多副本，扩容代理规格不会造成服务中断，但可能会导致消费的分区消息发生乱序，请谨慎评估业务影响，建议您在业务低峰期扩容。</li> <li>扩容代理规格的过程中，节点滚动重启造成分区Leader切换，会发生秒级连接闪断，在用户网络环境稳定的前提下，Leader切换时长一般为1分钟以内。多副本的Topic需要在生产客户端配置重试机制。</li> </ul>
修改VPC/子网/可用区	实例创建后，不支持修改VPC/子网/可用区。
是否支持Kerberos认证	不支持

限制项	约束与限制
客户端单IP连接数	Kafka实例的每个代理允许客户端单IP连接的个数默认为1000个，如果超过了，会出现连接失败问题。

## Topic

表 1-6 Topic 约束与限制

限制项	约束与限制
Topic总分区数	Topic总分区数和实例规格有关，具体请参考 <a href="#">Kafka集群实例</a> 。 Kafka以分区为粒度管理消息，分区多导致生产、存储、消费都碎片化，影响性能稳定性。在使用过程中，当Topic的总分区数达到上限后，用户无法继续创建Topic。
单个Topic分区数	<ul style="list-style-type: none"><li>按照开源Kafka现有逻辑，单个Topic分区数只支持增加，不支持减少。</li><li>出于性能考虑，建议单个Topic的分区数设置为200以内。</li></ul>
Topic数量	Topic数量和Topic总分区数、每个Topic的分区数有关，具体请参考 <a href="#">Kafka集群实例</a> 。

限制项	约束与限制
是否支持自动创建Topic	<p>支持。开启自动创建Topic表示生产或消费一个未创建的Topic时，系统会自动创建此Topic，此Topic的默认参数值如下：</p> <ul style="list-style-type: none"> <li>• 分区数为3。</li> <li>• 副本数为3。</li> <li>• 老化时间为72小时。</li> <li>• 不开启同步复制和同步落盘。</li> <li>• 消息时间戳类型为CreateTime。</li> <li>• 批处理消息最大值为10485760字节。</li> </ul> <p>集群实例如果在“实例管理 &gt; 配置参数”中修改“log.retention.hours”（老化时间）、“default.replication.factor”（副本数）或“num.partitions”（分区数）的参数值，此后自动创建的Topic参数值为修改后的参数值。</p> <p>例如：“num.partitions”修改为“5”，自动创建的Topic参数值如下：</p> <ul style="list-style-type: none"> <li>• 分区数为5。</li> <li>• 副本数为3。</li> <li>• 老化时间为72小时。</li> <li>• 不开启同步复制和同步落盘。</li> <li>• 消息时间戳类型为CreateTime。</li> <li>• 批处理消息最大值为10485760字节。</li> </ul>
同步复制	Topic副本数为1时，不能选择同步复制功能。
副本数	集群实例不建议使用单副本。实例节点出现故障的情况下，单副本Topic查询消息时可能会报“内部服务错误”，因此不建议使用单副本Topic。
老化时间	<p>如果Topic已经设置了老化时间，此时“实例管理 &gt; 配置参数”中的log.retention.hours值将不对此Topic生效。仅在Topic中未设置老化时间时，log.retention.hours值才会对此Topic生效。</p> <p>例如：Topic01设置的老化时间为60小时，log.retention.hours值为72小时，此时Topic01实际的老化时间为60小时。</p>
批量导入/导出Topic	支持批量导出，不支持批量导入。
Topic名称	Topic名称开头包含特殊字符，例如#号“#”，会导致监控数据无法展示。
是否支持延迟队列	不支持
代理故障场景	实例中部分代理故障时，无法创建、修改和删除Topic，只能查询Topic。

## 消费组

表 1-7 消费组约束与限制

限制项	约束与限制
是否需要创建消费组、消费者、生产者	<ul style="list-style-type: none"> <li>“auto.create.groups.enable”为“true”时，不需要单独创建消费组、生产者和消费者，在使用时自动生成，实例创建后，直接使用即可。</li> <li>“auto.create.groups.enable”为“false”时，需要手动创建消费组，不需要单独创建生产者和消费者。</li> </ul>
重置消费进度	重置消费进度可能会导致重复消费。
消费组名称	消费组名称开头包含特殊字符，例如#号“#”，会导致监控数据无法展示。
代理故障场景	实例中部分代理故障时，无法创建、删除消费组，以及重置消费进度，只能查询消费组。

## 消息

表 1-8 消息约束与限制

限制项	约束与限制
消息大小	生产消息的最大长度为10MB，超过10MB会导致生产失败。

## 用户

表 1-9 用户约束与限制

限制项	约束与限制
创建用户的数量	一个Kafka实例最多创建的用户数量在控制台存在两种限制，一种为20个，另一种为500个，具体以控制台为准。
代理故障场景	实例中部分代理故障时，无法创建、删除用户，以及重置密码，只能查询用户。

## 1.8 与其他云服务的关系

- 云审计（Cloud Trace Service）  
云审计为您提供云服务资源的操作记录，记录内容包括您从管理控制台或者开放API发起的云服务资源操作请求以及每次请求的结果，供您查询、审计和回溯使用。

当前CTS记录的操作，请参考[查看Kafka审计日志](#)。

- 虚拟私有云（Virtual Private Cloud）  
Kafka实例运行于虚拟私有云，需要使用虚拟私有云创建的IP和带宽。通过虚拟私有云安全组的功能可以增强访问Kafka实例的安全性。
- 弹性云服务器（Elastic Cloud Server）  
弹性云服务器是由CPU、内存、操作系统、云硬盘组成的基础的计算组件。Kafka实例运行在弹性云服务器上，一个代理对应一台弹性云服务器。
- 云硬盘（Elastic Volume Service）  
云硬盘为云服务器提供块存储服务，Kafka的所有数据（如消息、元数据和日志等）都保存在云硬盘中。
- 统一身份认证（Identity and Access Management）  
统一身份认证提供了权限管理功能，可以帮助您安全地控制云服务和资源的访问权限。您可以为不同的用户设置不同的访问Kafka实例的权限，以达到不同用户之前的权限隔离。
- 云监控（Cloud Eye）  
云监控是一个开放性的监控平台，提供资源的实时监控、告警、通知等服务。

#### 📖 说明

Kafka实例向CloudEye上报监控数据的更新周期为1分钟。

- 标签管理服务（Tag Management Service）  
标签管理服务是一种快速便捷将标签集中管理的可视化服务，提供跨区域、跨服务的集中标签管理和资源分类功能。  
为Kafka实例添加标签，可以方便用户识别和管理拥有的实例资源。
- VPC终端节点（VPC Endpoint）  
客户端和Kafka实例在同一区域的不同VPC中，利用VPC终端节点在不同VPC间建立跨VPC的连接通道，实现客户端通过内网访问Kafka实例。
- NAT网关（NAT Gateway）  
使用NAT网关的DNAT通过端口映射方式，将弹性IP地址映射到Kafka实例指定端口，实现Kafka实例和客户端通过公网通信。

## 1.9 Kafka 相关概念

云服务平台使用Kafka作为消息引擎，以下概念基于Kafka进行描述。

### Topic

消息主题。消息的生产与消费，围绕消息主题进行生产、消费以及其他消息管理操作。

Topic也是消息队列的一种发布与订阅消息模型。生产者向消息主题发布消息，多个消费者订阅该消息主题的消息，生产者与消费者彼此并无直接关系。

### 生产者（Producer）

向Topic（消息主题）发布消息的一方。发布消息的最终目的在于将消息内容传递给其他系统/模块，使对方按照约定处理该消息。

## 消费者 (Consumer)

从Topic (消息主题) 订阅消息的一方。订阅消息最终目的在于处理消息内容, 如日志集成场景中, 监报告警平台 (消费者) 从主题订阅日志消息, 识别出告警日志并发送告警消息/邮件。

## 代理 (Broker)

即Kafka集群架构设计中的单个Kafka进程, 一个Kafka进程对应一台服务器, 因此手册中描述的代理, 还包括对应的存储、带宽等服务器资源。

## 分区 (Partition)

为了实现水平扩展与高可用, Kafka将Topic划分为多个分区, 消息被分布式存储在分区中。

## 副本 (Replica)

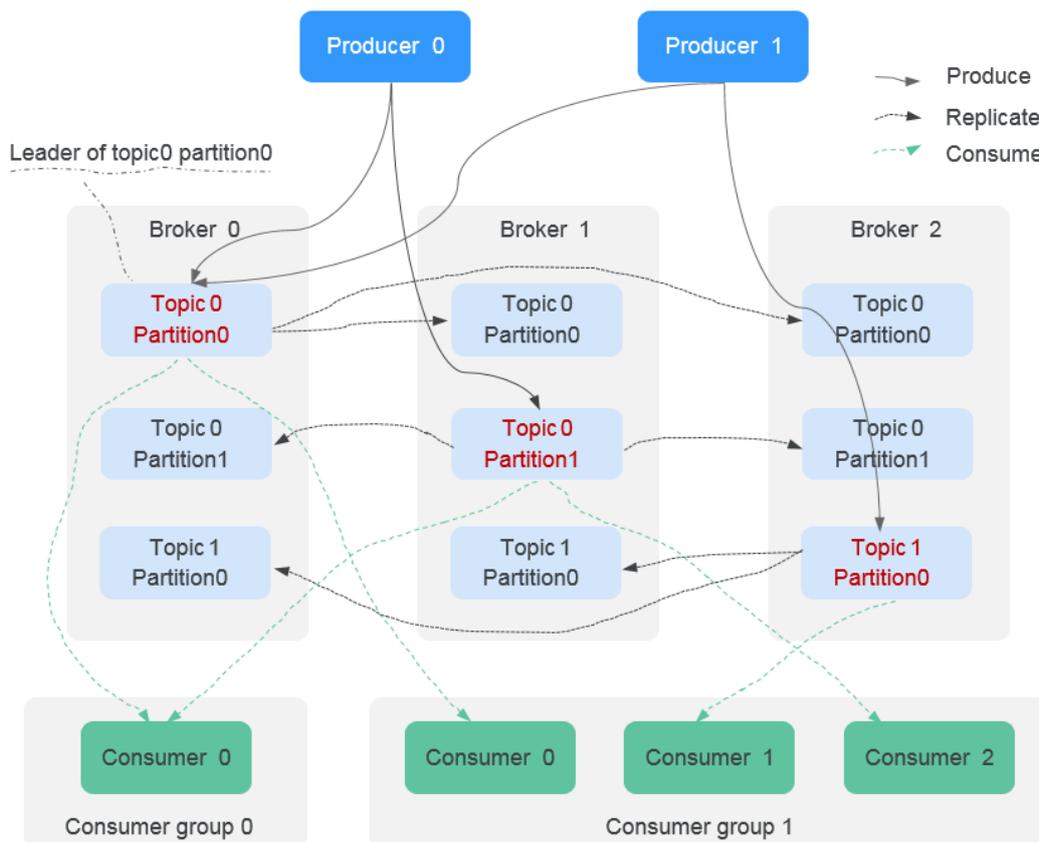
消息的备份存储。为了确保消息可靠, Kafka创建Topic时, 每个分区会分别从代理中选择1个或多个, 对消息进行冗余存储。

Topic的所有消息分布式存储在各个分区上, 分区在每个副本存储一份全量数据, 副本之间的消息数据保持同步, 任何一个副本不可用, 数据都不会丢失。

每个分区都随机挑选一个副本作为Leader, 该分区所有消息的生产与消费都在Leader副本上完成, 消息从Leader副本复制到其他副本 (Follower)。

Kafka的主题和分区属于逻辑概念, 副本与代理属于物理概念。下图通过消息的生产与消费流向, 解释了Kafka的分区、代理与主题间的关系。

图 1-5 Kafka 消息流



## 老化时间

消息的最长保留时间，消费者必须在此时间结束前消费消息，否则消息将被删除。删除的消息，无法被消费。

## 1.10 权限管理

如果您需要对云服务平台上购买的DMS for Kafka资源，给企业中的员工设置不同的访问权限，以达到不同员工之间的权限隔离，您可以使用统一身份认证服务（Identity and Access Management，简称IAM）进行精细的权限管理。该服务提供用户身份认证、权限分配、访问控制等功能，可以帮助您安全的控制云服务资源的访问。

通过IAM，您可以在账号中给员工创建IAM用户，并使用策略来控制他们对云服务资源的访问范围。例如您的员工中有负责软件开发的人员，您希望他们拥有DMS for Kafka的使用权限，但是不希望他们拥有删除Kafka实例等高危操作的权限，那么您可以使用IAM为开发人员创建用户，通过授予仅能使用Kafka实例，但是不允许删除Kafka实例的权限策略，控制他们对DMS for Kafka资源的使用范围。

如果账号已经能满足您的要求，不需要创建独立的IAM用户进行权限管理，您可以跳过本章节，不影响您使用DMS for Kafka的其它功能。

IAM是云服务平台提供权限管理的基础服务，无需付费即可使用，您只需要为您账号中的资源进行付费。

关于IAM的详细介绍，请参见[IAM产品简介](#)。

## 说明

DMS for Kafka的权限与策略基于分布式消息服务DMS，因此在IAM服务中为Kafka分配用户与权限时，请选择并使用“DMS”的权限与策略。

## DMS for Kafka 权限

默认情况下，管理员创建的IAM用户没有任何权限，需要将其加入用户组，并给用户组授予策略或角色，才能使得用户组中的用户获得对应的权限，这一过程称为授权。授权后，用户就可以基于被授予的权限对云服务进行操作。

DMS for Kafka部署时通过物理区域划分，为项目级服务。授权时，“作用范围”需要选择“区域级项目”，然后在指定区域对应的项目中设置相关权限，并且该权限仅对此项目生效；如果在“所有项目”中设置权限，则该权限在所有区域项目中都生效。访问DMS for Kafka时，需要先切换至授权区域。

权限根据授权精细程度分为角色和策略。

- 角色：IAM最初提供的一种根据用户的工作职能定义权限的粗粒度授权机制。该机制以服务为粒度，提供有限的服务相关角色用于授权。由于云服务平台各服务之间存在业务依赖关系，因此给用户授予角色时，可能需要一并授予依赖的其他角色，才能正确完成业务。角色并不能满足用户对精细化授权的要求，无法完全达到企业对权限最小化的安全管控要求。
- 策略：IAM最新提供的一种细粒度授权的能力，可以精确到具体服务的操作、资源以及请求条件等。基于策略的授权是一种更加灵活的授权方式，能够满足企业对权限最小化的安全管控要求。例如：针对DMS for Kafka服务，管理员能够控制IAM用户仅能对实例进行指定的管理操作。多数细粒度策略以API接口为粒度进行权限拆分，DMS for Kafka支持的API授权项请参见[细粒度策略支持的授权项](#)。

如表1-10所示，包括了DMS for Kafka的所有系统权限。

表 1-10 DMS for Kafka 系统权限

系统角色/策略名称	描述	类别	依赖关系
DMS FullAccess	分布式消息服务管理员权限，拥有该权限的用户可以操作所有分布式消息服务的功能。	系统策略	无
DMS UserAccess	分布式消息服务普通用户权限（没有实例创建、修改、删除、扩容）。	系统策略	无
DMS ReadOnlyAccess	分布式消息服务的只读权限，拥有该权限的用户仅能查看分布式消息服务数据。	系统策略	无
DMS VPCAccess	分布式消息服务租户委托时需要授权的VPC操作权限。	系统策略	无
DMS KMSAccess	分布式消息服务租户委托时需要授权的KMS操作权限。	系统策略	无
DMS ELBAccess	分布式消息服务租户委托时需要授权的ELB操作权限。	系统策略	无

系统角色/策略名称	描述	类别	依赖关系
DMS VPCEndpointAccess	分布式消息服务租户委托时需要授权的VPCEndpoint操作权限。	系统策略	无
DMSAgencyCheckAccessPolicy	分布式消息服务检查租户委托权限需要授权的IAM操作权限。	系统策略	无
DMS BSSAccess	分布式消息服务租户委托时需要授权的BSS操作权限。	系统策略	无
DMS OBSAccess	分布式消息服务租户委托时需要授权的OBS操作权限。	系统策略	无

### 📖 说明

系统策略有包含OBS授权项，由于缓存的存在，对用户、用户组以及企业项目授予OBS相关的系统策略后，大概需要等待5分钟系统策略才能生效。

**表1-11**列出了DMS for Kafka常用操作与系统策略的授权关系，您可以参照该表选择合适的系统策略。

**表 1-11** 常用操作与系统策略的关系

操作	DMS FullAccess	DMS UserAccess	DMS ReadOnlyAccess
创建实例	√	×	×
修改实例	√	×	×
删除实例	√	×	×
变更实例规格	√	×	×
重启实例	√	√	×
查询实例信息	√	√	√

除**表1-11**以外的系统策略，没有任何DMS for Kafka常用操作的权限。

## 细粒度授权

使用自定义细粒度策略，请使用管理员用户登录IAM控制台，按需选择DMS的细粒度权限进行授权操作。DMS for Kafka细粒度权限依赖说明请参见**表1-12**。

表 1-12 DMS for Kafka 细粒度权限依赖说明

权限名称	权限描述	权限依赖
dms:instance:list	查看实例列表	无
dms:instance:get	查看实例详情信息	无
dms:instance:create	创建实例	<ul style="list-style-type: none"> <li>• vpc:vpcs:get</li> <li>• vpc:ports:create</li> <li>• vpc:securityGroups:get</li> <li>• vpc:ports:get</li> <li>• vpc:subnets:get</li> <li>• vpc:vpcs:list</li> <li>• vpc:publicIps:get</li> <li>• vpc:publicIps:list</li> <li>• vpc:ports:update</li> <li>• vpc:publicIps:update</li> <li>• vpc:ports:delete</li> </ul>
dms:instance:getBackgroundTask	查看实例后台任务详情	无
dms:instance:deleteBackgroundTask	删除实例后台任务	无
dms:instance:modifyStatus	重启实例	无
dms:instance:resetAuthInfo	重置实例访问密码	无
dms:instance:modifyAuthInfo	修改实例访问密码	无
dms:instance:modify	修改实例	<ul style="list-style-type: none"> <li>• vpc:vpcs:get</li> <li>• vpc:ports:create</li> <li>• vpc:securityGroups:get</li> <li>• vpc:ports:get</li> <li>• vpc:subnets:get</li> <li>• vpc:vpcs:list</li> <li>• vpc:publicIps:get</li> <li>• vpc:publicIps:list</li> <li>• vpc:ports:update</li> <li>• vpc:publicIps:update</li> <li>• vpc:ports:delete</li> </ul>

权限名称	权限描述	权限依赖
dms:instance:scale	实例开启扩容功能	<ul style="list-style-type: none"><li>• vpc:vpcs:get</li><li>• vpc:ports:create</li><li>• vpc:securityGroups:get</li><li>• vpc:ports:get</li><li>• vpc:subnets:get</li><li>• vpc:vpcs:list</li><li>• vpc:publicIps:get</li><li>• vpc:publicIps:list</li><li>• vpc:ports:update</li><li>• vpc:publicIps:update</li></ul>
dms:instance:delete	删除实例	无

## 相关链接

- [IAM产品简介](#)
- [创建用户组、用户并授予DMS for Kafka权限](#)
- [细粒度策略支持的授权项](#)

# 2 快速入门

## 2.1 快速连接 Kafka 并生产消费消息

本章节将为您介绍分布式消息服务Kafka版入门的使用流程，以创建一个开启密文接入的Kafka实例，客户端使用内网通过同一个VPC连接Kafka实例生产消费消息为例，帮助您快速上手Kafka。

图 2-1 Kafka 使用流程



### 1. 准备工作

Kafka实例运行于虚拟私有云（Virtual Private Cloud，简称VPC）中，在创建实例前需要确保有可用的虚拟私有云。

Kafka实例创建后，您需要在弹性云服务器中下载和安装Kafka开源客户端，然后才能进行生产消息和消费消息。

### 2. 创建Kafka实例

在创建实例时，您可以根据需求选择需要的实例规格和数量，并开启密文接入。

### 3. 创建Topic

Topic用于存储消息，供生产者生产消息以及消费者订阅消息。

本文以在控制台创建Topic为例介绍。

#### 4. 连接实例

在连接开启SASL的Kafka实例时，需要下载证书，并在客户端配置文件中设置连接信息。

### 步骤一：准备工作

**步骤1** 为用户添加Kafka实例的操作权限。

如果您需要对云上的资源进行精细管理，请使用统一身份认证服务（Identity and Access Management，简称IAM）创建IAM用户及用户组，并授权，以使得IAM用户获得Kafka实例的操作权限。具体操作请参考[创建用户并授权使用DMS for Kafka](#)。

**步骤2** 创建VPC和子网。

在创建Kafka实例前，确保已存在可用的VPC和子网。创建VPC和子网的具体步骤，请参考[创建虚拟私有云和子网](#)。

**创建的VPC与Kafka实例必须在相同的区域。**

**步骤3** 创建安全组并添加安全组规则。

在创建Kafka实例前，确保已存在可用的安全组。创建安全组的具体步骤，请参考[创建安全组](#)。

**创建的安全组与Kafka实例必须在相同的区域。**

连接Kafka实例前，请添加[表2-1](#)所示安全组规则，其他规则请根据实际需要添加。

表 2-1 安全组规则

方向	协议	端口	源地址	说明
入方向	TCP	9093	0.0.0.0/0	使用内网通过同一个VPC访问Kafka实例（密文接入）。

#### 📖 说明

安全组创建后，系统默认添加入方向“允许安全组内的弹性云服务器彼此通信”规则和出方向“放通全部流量”规则。此时使用内网通过同一个VPC访问Kafka实例，无需添加[表2-1](#)的规则。

**步骤4** 构建生产消费客户端。

本文以Linux系统的弹性云服务器（Elastic Cloud Server，简称ECS）作为生产消费客户端。在创建Kafka实例前，请先创建开启弹性公网IP的ECS、安装JDK、配置环境变量以及下载Kafka开源客户端。

1. 登录管理控制台，在左上角单击，选择“计算 > 弹性云服务器”，创建一个ECS实例。  
创建ECS的具体步骤，请参考[创建弹性云服务器](#)。如果您已有可用的ECS，可重复使用，不需要再次创建。
2. 使用root用户登录ECS。
3. 安装Java JDK，并配置JAVA\_HOME与PATH环境变量。
  - a. 下载Java JDK。

## 说明

ECS默认自带的JDK可能不符合要求，例如OpenJDK，需要配置为Oracle的JDK，可至Oracle官方下载页面[下载Java Development Kit 1.8.111及以上版本](#)。

- b. 解压Java JDK。  

```
tar -zxvf jdk-8u321-linux-x64.tar.gz
```

“jdk-8u321-linux-x64.tar.gz”为JDK的版本，请根据实际情况修改。
  - c. 打开“.bash\_profile”文件。  

```
vim ~/.bash_profile
```
  - d. 在键盘上按“i”，添加如下内容。  

```
export JAVA_HOME=/root/jdk1.8.0_321
export PATH=$JAVA_HOME/bin:$PATH
```

“/root/jdk1.8.0\_321”为JDK的安装路径，请根据实际情况修改。
  - e. 按“Esc”，然后输入以下命令，按“Enter”，保存并退出“.bash\_profile”文件。  

```
:wq
```
  - f. 执行如下命令使修改生效。  

```
source .bash_profile
```
  - g. 查看Java JDK是否安装成功。  

```
java -version
```

回显信息中包含如下信息，表示Java JDK安装成功。  

```
java version "1.8.0_321"
```
4. 下载开源的[Kafka客户端](#)。  

```
wget https://archive.apache.org/dist/kafka/2.7.2/kafka_2.12-2.7.2.tgz
```
  5. 解压Kafka客户端文件。  

```
tar -zxf kafka_2.12-2.7.2.tgz
```

---结束

## 步骤二：创建 Kafka 实例

**步骤1** 登录分布式消息服务Kafka控制台，单击页面右上方的“购买Kafka实例”。

**步骤2** 在“快速购买”页签，设置实例基础配置，配置详情请参考[表2-2](#)。

表 2-2 设置实例基础配置

参数	说明
计费模式	选择“按需计费”，即先使用再付费，按照Kafka实例实际使用时长计费，秒级计费，按小时结算。
区域	不同区域的云服务产品之间内网互不相通。请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。 选择“马来西亚-吉隆坡”。
可用区	可用区指在同一区域下，电力、网络隔离的物理区域，可用区之间内网互通，不同可用区之间物理隔离。 选择“可用区1、可用区2、可用区3”。

**步骤3** 设置实例规格和存储空间，配置详情请参考[表2-3](#)。

表 2-3 设置实例规格和存储空间

参数	说明
套餐规格	选择“综合推荐”中的“进阶规格”。进阶规格适用于性能和时延要求较高的在线生产系统等。
类型	根据实际需要选择存储Kafka数据的磁盘类型。 Kafka实例创建后，磁盘类型不支持修改。 选择“超高I/O”。
容量	根据实际需要选择存储Kafka数据的磁盘大小。 实例总存储空间 = 单个代理磁盘容量 * 代理数量。 选择“100GB”。

步骤4 设置实例网络环境信息，配置详情请参考表2-4。

表 2-4 设置实例网络环境信息

参数	说明
虚拟私有云	虚拟私有云在Kafka实例创建完成后，不支持修改。 选择 <b>准备工作</b> 中设置好的虚拟私有云。
子网	子网在Kafka实例创建完成后，不支持修改。 选择 <b>准备工作</b> 中设置好的子网。
内网IP地址	选择“自动分配”，即系统自动分配子网中可用的IP地址。
安全组	选择 <b>准备工作</b> 中设置好的安全组。

步骤5 设置实例的访问方式，配置详情请参考表2-5。

表 2-5 设置实例的访问方式

参数	说明
接入方式	选择“密文接入”，即客户端连接Kafka实例时，需要进行SASL认证。
kafka安全协议	选择“SASL_SSL”，即采用SASL方式进行认证，数据通过SSL证书进行加密传输，安全性更高。
SASL认证机制	勾选“PLAIN”。密文接入成功开启后，SASL机制不支持修改。 开启PLAIN机制后，同时支持SCRAM-SHA-512机制和PLAIN机制，根据实际情况选择其中任意一种配置连接。

参数	说明
SSL用户名	<p>用户名输入“test”，密文接入成功开启后，用户名不支持修改。</p> <p>用户名需要符合以下命名规则：由英文字母开头，且只能由英文字母、数字、中划线、下划线组成，长度为4~64个字符。</p>
密码	<p>设置连接实例的密码，密码需要符合以下命名规则：</p> <ul style="list-style-type: none"> <li>• 长度为8~32个字符。</li> <li>• 至少包含以下字符中的3种：大写字母、小写字母、数字、特殊字符`~!@#\$%^&amp;*()-_+=\ []{};:'",&lt;.&gt;?`和空格，并且不能以-开头。</li> <li>• 不能与用户名或倒序的用户名相同。</li> </ul>

**步骤6** 设置实例高级配置，配置详情请参考表2-6，其他参数保持默认设置。

表 2-6 设置实例高级配置

参数	说明
实例名称	<p>实例名称支持自定义，但需要符合命名规则：长度为4~64个字符，由英文字母开头，只能由英文字母、数字、中划线、下划线组成。</p> <p>输入“kafka-test”。</p>
企业项目	<p>该参数针对企业用户使用。企业项目是对企业不同项目间资源的分组和管理，属于逻辑隔离。</p> <p>选择“default”。</p>
容量阈值策略	<p>选择“自动删除”，即磁盘达到95%的容量阈值后，可以正常生产和消费消息，但是会删除最早的10%的消息，以保证磁盘容量充足。该场景优先保障业务不中断，数据存在丢失的风险。</p>
Manager Username	<p>Kafka Manager名称支持自定义，但需要符合命名规则：长度为4~64个字符，由英文字母开头，只能由英文字母、数字、中划线、下划线组成。</p> <p>输入“manager-test”。</p>
Password	<p>设置Kafka Manager的密码。</p> <p>密码需要符合以下命名规则：</p> <ul style="list-style-type: none"> <li>• 长度为8~32个字符。</li> <li>• 至少包含以下字符中的3种：大写字母、小写字母、数字、特殊字符`~!@#\$%^&amp;*()-_+=\ []{};:'",&lt;.&gt;?`和空格，并且不能以-开头。</li> <li>• 不能与用户名或倒序的用户名相同。</li> </ul>

**步骤7** 单击“确认订单”，进入规格确认页面。

**步骤8** 确认实例信息无误后，提交请求。

**步骤9** 单击“返回Kafka实例列表”，查看Kafka实例是否创建成功。

创建实例大约需要3到15分钟，此时实例的“状态”为“创建中”。

- 当实例的“状态”变为“运行中”时，说明实例创建成功。
- 当实例的“状态”变为“创建失败”，请删除创建失败的实例，重新创建。如果重新创建仍然失败，请联系客服。

#### 说明

创建失败的实例，不会占用其他资源。

**步骤10** 实例创建成功后，单击实例名称，进入实例详情页。

**步骤11** 在“连接信息”区域，查看并记录连接地址。

**图 2-2** 使用内网通过同一个 VPC 访问 Kafka 实例的连接地址

内网连接地址 IPv4 192.168.0.239:9093,192.168.0.182:9093,192.168.0.57:9093 

----结束

## 步骤三：创建 Topic

**步骤1** 在“Kafka实例”页面，单击Kafka实例的名称，进入实例详情页面。

**步骤2** 在左侧导航栏单击“实例管理 > Topic管理”，进入Topic列表页。

**步骤3** 单击“创建Topic”，弹出“创建Topic”对话框。

**步骤4** 填写Topic名称和配置信息，配置详情请参考表2-7，单击“确定”，完成创建Topic。

表 2-7 Topic 参数说明

参数	说明
Topic名称	<p>名称支持自定义，但需要符合命名规则：以英文字母、数字、下划线开头，且只能由英文字母、数字、句点、中划线、下划线组成，长度为3~200个字符。</p> <p>名称不能为以下内置Topic：</p> <ul style="list-style-type: none"> <li>• __consumer_offsets</li> <li>• __transaction_state</li> <li>• __trace</li> <li>• __connect-status</li> <li>• __connect-configs</li> <li>• __connect-offsets</li> <li>• __cloud_event</li> <li>• __site_event</li> <li>• __dms_dial_test</li> <li>• __remote_log_metadata</li> <li>• __remote_log_compact_metadata</li> <li>• __remote_log_metadata_offset</li> </ul> <p>创建Topic后不能修改名称。 输入“topic-01”。</p>
分区数	<p>如果分区数与消费者数一致，分区数越大消费的并发度越大。 输入“3”。</p>
副本数	<p>Kafka会自动在每个副本上备份数据，当其中一个Broker故障时数据依然是可用的，副本数越大可靠性越高。 输入“3”。</p>
老化时间（小时）	<p>消息的最长保留时间，消费者必须在此时间结束前消费消息，否则消息将被删除。删除的消息，无法被消费。 输入“72”。</p>
同步复制	<p>选择“不开启”，即Leader副本接收到消息并成功写入到本地日志后，就马上向客户端发送写入成功的消息，不需要等待所有Follower副本复制完成。</p>
同步落盘	<p>选择“不开启”，即生产的消息存在内存中，不会立即写入磁盘。</p>
消息时间戳类型	<p>选择“CreateTime”，即生产者创建消息的时间。</p>
批处理消息最大值（字节）	<p>Kafka允许的最大批处理大小，如果在生产客户端配置文件或代码中启用消息压缩，则表示压缩后的最大批处理大小。 输入“10485760”。</p>

----结束

## 步骤四：连接实例生产消费消息

### 步骤1 配置生产消费配置文件。

1. 登录Linux系统的ECS。
2. 下载client.truststore.jks证书并上传到ECS的“/root”目录下。

获取证书的方法如下：在Kafka控制台单击Kafka实例名称，进入实例详情页面，在“连接信息 > SSL证书”所在行，单击“下载”。解压压缩包，获取压缩包中的客户端证书文件：client.truststore.jks。

#### 📖 说明

“/root”为证书存放路径，请根据实际情况修改。

3. 进入Kafka客户端文件的“/config”目录下。  
`cd kafka_2.12-2.7.2/config`
4. 在“consumer.properties”和“producer.properties”文件中分别增加如下行（示例以PLAIN机制为例介绍）。

```
sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \  
username="*****" \  
password="*****";  
sasl.mechanism=PLAIN
```

```
security.protocol=SASL_SSL  
ssl.truststore.location={ssl_truststore_path}  
ssl.truststore.password=dms@kafka  
ssl.endpoint.identification.algorithm=
```

#### 参数说明：

- username和password为创建Kafka实例过程中开启密文接入时填入的用户名和密码。
- ssl.truststore.location配置为**步骤1.2**证书的存放路径。
- **ssl.truststore.password**为服务器证书密码，不可更改，需要保持为**dms@kafka**。
- ssl.endpoint.identification.algorithm为证书域名校验开关，为空则表示关闭。这里需要**保持关闭状态，必须设置为空**。

### 步骤2 进入Kafka客户端文件的“/bin”目录下。

```
cd ../bin
```

### 步骤3 生产消息。

```
./kafka-console-producer.sh --broker-list ${连接地址} --topic ${Topic名称} --producer.config ../config/  
producer.properties
```

#### 参数说明如下：

- 连接地址：从**创建实例**中获取的连接地址。
- Topic名称：从**创建Topic**中获取的Topic名称。

示例如下，“192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093”为Kafka实例连接地址。

执行完命令后，输入需要生产的消息内容，按“Enter”发送消息到Kafka实例，输入的每一行内容都将作为一条消息发送到Kafka实例。

```
[root@ecs-kafka bin]#./kafka-console-producer.sh --broker-list  
192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093 --topic topic-01 --producer.config ../config/  
producer.properties  
>Hello  
>DMS
```

```
>Kafka!  
>^C[root@ecs-kafka bin]#
```

如需停止生产使用**Ctrl+C**命令退出。

#### 步骤4 消费消息。

```
./kafka-console-consumer.sh --bootstrap-server ${连接地址} --topic ${Topic名称} --from-beginning --  
consumer.config ../config/consumer.properties
```

参数说明如下：

- 连接地址：从[创建实例](#)中获取的连接地址。
- Topic名称：从[创建Topic](#)中获取的Topic名称。

示例如下：

```
[root@ecs-kafka bin]# ./kafka-console-consumer.sh --bootstrap-server  
192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093 --topic topic-01 --from-beginning --  
consumer.config ../config/consumer.properties  
Hello  
Kafka!  
DMS  
^CProcessed a total of 3 messages  
[root@ecs-kafka bin]#
```

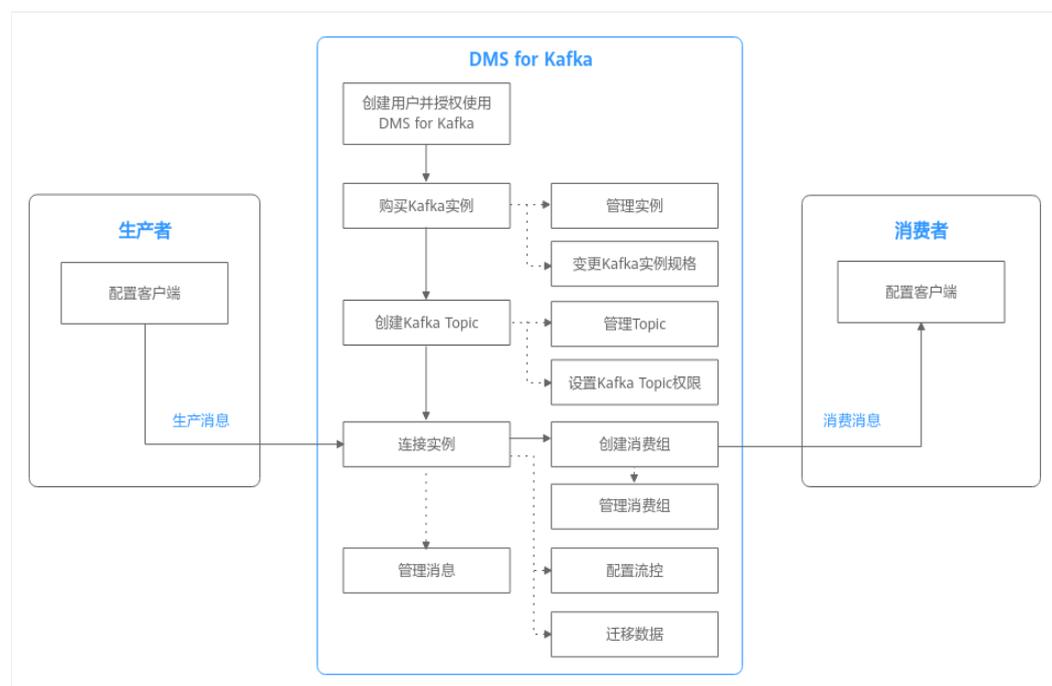
如需停止消费使用**Ctrl+C**命令退出。

----结束

# 3 Kafka 业务使用流程

分布式消息服务Kafka是一款基于开源社区版Kafka提供的消息队列服务，向用户提供计算、存储和带宽资源独占式的Kafka实例。使用Kafka实例生产消费消息的流程如下图所示。

图 3-1 Kafka 业务使用流程



- 1. 创建用户并授权使用DMS for Kafka**  
创建IAM用户，并授予DMS for Kafka的权限，以达到用户之间权限的隔离。
- 2. 购买Kafka实例**  
Kafka实例采用物理隔离的方式部署，租户独占Kafka实例。
- 3. 创建Kafka Topic**  
创建用于存储消息的Topic，供生产者生产消息以及消费者订阅消息。
- 4. 连接实例**  
客户端使用内网连接Kafka实例，并生产消费消息。



# 4 权限管理

## 4.1 创建用户并授权使用 DMS for Kafka

如果您需要对您所拥有的DMS for Kafka服务进行精细的权限管理，您可以使用统一身份认证服务（Identity and Access Management，简称IAM），通过IAM，您可以：

- 根据企业的业务组织，在您的账号中，给企业中不同职能部门的员工创建IAM用户，让员工拥有唯一安全凭证，并使用DMS for Kafka资源。
- 根据企业用户的职能，设置不同的访问权限，以达到用户之间的权限隔离。
- 将DMS for Kafka资源委托给更专业、高效的其他账号或者云服务，这些账号或者云服务可以根据权限进行代运维。

如果账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用DMS for Kafka服务的其它功能。

本章节为您介绍对用户授权的方法，操作流程如[图4-1](#)所示。

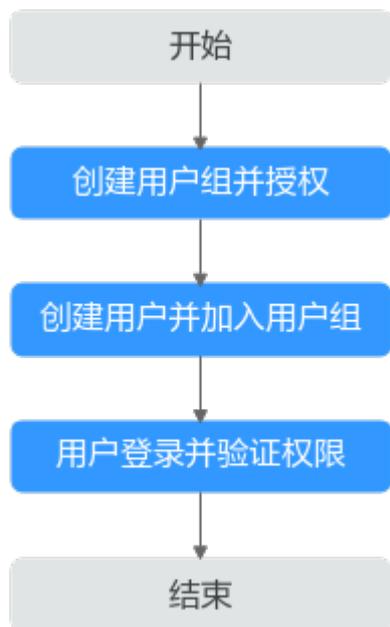
### 前提条件

给用户组授权之前，请您了解用户组可以添加的DMS for Kafka系统策略，并结合实际需求进行选择，DMS for Kafka支持的系统策略及策略间的对比，请参见：[DMS for Kafka系统策略](#)。若您需要对除DMS for Kafka之外的其它服务授权，IAM支持服务的所有策略请参见[系统权限](#)。

**DMS for Kafka的权限与策略基于分布式消息服务DMS，因此在IAM服务中为用户组授予DMS for Kafka权限时，请选择并使用“DMS”的权限与策略。**

## 示例流程

图 4-1 给用户授权 DMS for Kafka 权限流程



### 1. 创建用户组并授权

在IAM控制台创建用户组，并授予DMS for Kafka的只读权限“DMS ReadOnlyAccess”。

### 2. 创建用户并加入用户组

在IAM控制台创建用户，并将其加入1中创建的用户组。

### 3. 用户登录并验证权限

新创建的用户登录到控制台，切换至授权区域，验证权限：

- 在“服务列表”中选择分布式消息服务Kafka，进入Kafka实例主界面，单击右上角“购买Kafka实例”，尝试购买Kafka实例，如果无法购买Kafka实例（假设当前权限仅包含DMS ReadOnlyAccess），表示“DMS ReadOnlyAccess”已生效。
- 在“服务列表”中选择云硬盘（假设当前策略仅包含DMS ReadOnlyAccess），若提示权限不足，表示“DMS ReadOnlyAccess”已生效。
- 在“服务列表”中选择分布式消息服务Kafka（假设当前策略仅包含DMS ReadOnlyAccess），进入Kafka实例主界面，如果能够查看Kafka实例列表，表示“DMS ReadOnlyAccess”已生效。

## DMS for Kafka 自定义策略样例

如果系统预置的DMS for Kafka权限，不满足您的授权要求，可以创建自定义策略。自定义策略中可以添加的授权项（Action）请参考[细粒度策略支持的授权项](#)。

目前支持以下两种方式创建自定义策略：

- 可视化视图创建自定义策略：无需了解策略语法，按可视化视图导航栏选择云服务、操作、资源、条件等策略内容，可自动生成策略。

- JSON视图创建自定义策略：可以在选择策略模板后，根据具体需求编辑策略内容；也可以直接在编辑框内编写JSON格式的策略内容。

具体创建步骤请参见：[创建自定义策略](#)。下面为您介绍常用的DMS for Kafka自定义策略样例。

- 示例1：授权用户删除实例和重启实例

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dms:instance:modifyStatus",
        "dms:instance:delete"
      ]
    }
  ]
}
```

- 示例2：拒绝用户删除实例

拒绝策略需要同时配合其他策略使用，否则没有实际作用。用户被授予的策略中，一个授权项的作用如果同时存在Allow和Deny，则遵循**Deny优先原则**。

如果您给用户授予DMS FullAccess的系统策略，但不希望用户拥有DMS FullAccess中定义的删除实例权限，您可以创建一条拒绝删除实例的自定义策略，然后同时将DMS FullAccess和拒绝策略授予用户，根据Deny优先原则，则用户可以对DMS for Kafka执行除了删除实例外的所有操作。拒绝策略示例如下：

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "dms:instance:delete"
      ]
    }
  ]
}
```

## DMS for Kafka 资源

资源是服务中存在的对象。在DMS for Kafka中，资源包括：kafka，您可以在创建自定义策略时，通过指定资源路径来选择特定资源。

表 4-1 DMS for Kafka 的指定资源与对应路径

指定资源	资源名称	资源路径
kafka	实例	<p><b>【格式】</b> DMS:*:*kafka:实例ID</p> <p><b>【说明】</b> 对于实例资源，IAM自动生成资源路径前缀<b>DMS:*:*kafka:</b>通过实例ID指定具体的资源路径，支持通配符*。例如： DMS:*:*kafka:*表示任意Kafka实例。</p>

## DMS for Kafka 请求条件

您可以在创建自定义策略时，通过添加“请求条件”（Condition元素）来控制策略何时生效。请求条件包括条件键和运算符，条件键表示策略语句的Condition元素，分为全局级条件键和服务级条件键。**全局级条件键**（前缀为g:）适用于所有操作，服务级条件键（前缀为服务缩写，如dms:）仅适用于对应服务的操作。运算符与条件键一起使用，构成完整的条件判断语句。

DMS for Kafka通过IAM预置了一组条件键，例如，您可以先使用dms:ssl条件键检查Kafka实例是否开启SASL，然后再允许执行操作。下表显示了适用于DMS for Kafka服务特定的条件键。

表 4-2 DMS for Kafka 请求条件

DMS for Kafka条件键	运算符	描述
dms:publicIP	Bool	是否开启公网
dms:ssl	Bool	是否开启SSL

# 5 购买 Kafka 实例

Kafka实例采用物理隔离的方式部署，租户独占Kafka实例。支持用户自定义规格和自定义特性，您可以根据业务需要定制相应计算能力和存储空间的Kafka实例。

## 准备实例依赖资源

创建Kafka实例前，请提前准备好如表5-1所示资源。

表 5-1 Kafka 实例依赖资源

资源名称	要求	创建指导
VPC和子网	<p>Kafka实例可以使用当前账号下已创建的VPC和子网，也可以使用新创建的VPC和子网，请根据实际需要进行配置。</p> <p>在创建VPC和子网时应注意：</p> <ul style="list-style-type: none"><li>• 创建的VPC与Kafka实例在相同的区域。</li><li>• 子网开启IPv6后，Kafka实例支持IPv6功能。Kafka实例开启IPv6后，客户端可以使用IPv6地址连接实例。</li></ul>	<p>创建VPC和子网的操作指导请参考《虚拟私有云 用户指南》的创建虚拟私有云和子网。</p>
安全组	<p>不同的Kafka实例可以重复使用相同的安全组，也可以使用不同的安全组，请根据实际需要进行配置。</p> <p>安全组与Kafka实例必须在相同的区域。</p> <p>连接Kafka实例前，请根据连接方式配置对应的安全组，具体请参考表7-2。</p>	<p>创建安全组的操作指导请参考《虚拟私有云 用户指南》的创建安全组，为安全组添加规则的操作指导请参考《虚拟私有云 用户指南》的添加安全组规则。</p>

## 快速购买 Kafka 集群实例

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择区域。

不同区域的云服务产品之间内网互不相通。请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入 Kafka实例列表页面。

**步骤4** 单击页面右上方的“购买Kafka实例”。

**步骤5** 在“快速购买”页签中，设置实例基础配置。

表 5-2 实例基础配置参数说明

参数名称	说明
计费模式	<ul style="list-style-type: none"> <li>按需付费：一种后付费模式，即先使用再付费，按照Kafka实例实际使用时长计费，秒级计费，按小时结算。</li> </ul>
区域	不同区域的云服务产品之间内网互不相通。请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。
可用区	<p>可用区指在同一区域下，电力、网络隔离的物理区域，可用区之间内网互通，不同可用区之间物理隔离。</p> <p>根据实际情况选择1个或者3个及以上可用区。实例创建成功后，不支持修改可用区。</p>

**步骤6** 设置套餐规格。

分布式消息服务Kafka版已经预设了几种规格，请根据需要选择。磁盘类型和磁盘大小根据实际需求进行设置。**Kafka实例创建后，磁盘类型不支持修改。**

存储空间用于存储消息（包括副本中的消息）、日志和元数据，建议根据业务消息体积、副本数量以及预留磁盘大小选择存储空间大小。每个Kafka代理会预留33GB的磁盘空间，用于存储日志和元数据。

创建实例时会进行磁盘格式化，磁盘格式化会导致实际可用磁盘为总磁盘的93%~95%。

**步骤7** 设置网络信息。

表 5-3 实例网络参数说明

参数名称	说明
虚拟私有云	<p>选择已经创建好的或共享的虚拟私有云。</p> <p>虚拟私有云可以为您的Kafka实例构建隔离的、能自主配置和管理的虚拟网络环境。您可以单击右侧的“管理虚拟私有云”，跳转到“虚拟私有云”页面，查看或创建虚拟私有云。</p> <p>虚拟私有云在Kafka实例创建完成后，不支持修改。</p>

参数名称	说明
子网	选择已经创建好的子网。 子网在Kafka实例创建完成后，不支持修改。
IPv6	子网开启IPv6后，页面才显示此参数。开启IPv6后，客户端可以使用IPv6地址连接实例。 实例创建成功后，不支持修改IPv6开关。
内网IP地址	根据实际需求选择“自动分配”或“手动选择”。 <ul style="list-style-type: none"> <li>自动分配：系统自动分配子网中可用的IP地址。</li> <li>手动选择：在下拉框中勾选指定的IP地址。如果指定的IP地址数量少于代理数量时，系统会自动为剩余代理分配IP地址。</li> </ul> 开启IPv6后，控制台不显示此参数。
安全组	选择已经创建好的安全组。 安全组是一组对Kafka实例的访问规则的集合。您可以单击右侧的“创建安全组”，跳转到“安全组”页面，查看或创建安全组。 客户端连接Kafka实例前，请根据不同的连接方式配置对应的安全组规则，否则会连接失败。具体的安全组规则请参考 <a href="#">表 7-2</a> 。

**步骤8** 设置实例访问方式。

**表 5-4** 实例访问方式参数说明

参数名称	说明
接入方式	接入方式分为以下两种： <ul style="list-style-type: none"> <li>明文接入：表示客户端连接Kafka实例时，无需进行SASL认证。</li> <li>密文接入：表示客户端连接Kafka实例时，需要进行SASL认证。 开启“密文接入”后，需要设置SSL用户名、密码。</li> </ul> 内网访问不支持关闭，明文接入和密文接入至少开启一个。
kafka安全协议	<ul style="list-style-type: none"> <li>SASL_SSL：采用SASL方式进行认证，数据通过SSL证书进行加密传输，安全性更高。</li> <li>SASL_PLAINTEXT：采用SASL方式进行认证，数据通过明文传输，性能更好。由于明文传输，建议使用SCRAM-SHA-512机制。</li> </ul>

参数名称	说明
SASL认证机制	<ul style="list-style-type: none"> <li>未勾选“PLAIN”时，使用SCRAM-SHA-512机制对账号密码进行认证。</li> <li>勾选“PLAIN”后，同时支持SCRAM-SHA-512机制和PLAIN机制，根据实际情况选择其中任何一种配置连接。</li> </ul> <p>密文接入成功开启后，SASL PLAIN机制不支持修改。</p> <p><b>什么是SCRAM-SHA-512机制和PLAIN机制？</b></p> <ul style="list-style-type: none"> <li>SCRAM-SHA-512机制：采用哈希算法对用户名与密码生成凭证，进行身份校验的安全认证机制，比PLAIN机制安全性更高。</li> <li>PLAIN机制：一种简单的用户名密码校验机制。</li> </ul>
SSL用户名	<p>客户端用于连接Kafka实例的用户名。</p> <p>用户名需要符合以下命名规则：由英文字母开头，且只能由英文字母、数字、中划线、下划线组成，长度为4~64个字符。</p> <p>密文接入成功开启后，用户名不支持修改。</p>
密码	<p>客户端用于连接Kafka实例的密码。</p> <p>密码需要符合以下规则：</p> <ul style="list-style-type: none"> <li>长度为8~32个字符。</li> <li>至少包含以下字符中的3种：大写字母、小写字母、数字、特殊字符`~!@#\$%^&amp;*()-_+=\ [{]`;&lt;.:&gt;?和空格，并且不能以-开头。</li> <li>不能与用户名或倒序的用户名相同。</li> </ul>

**步骤9** 设置高级配置。

表 5-5 高级配置参数说明

参数名称	说明
实例名称	<p>名称支持自定义，但需要符合命名规则：长度为4~64个字符，由英文字母开头，只能由英文字母、数字、中划线、下划线组成。</p>
企业项目	<p>该参数针对企业用户使用。</p> <p>企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理，默认项目为default。</p>

参数名称	说明
容量阈值策略	<p>设置磁盘使用达到容量阈值后的消息处理策略，容量阈值为95%。</p> <ul style="list-style-type: none"> <li>● 自动删除：可以正常生产和消费消息，但是会删除最早的10%的消息，以保证磁盘容量充足。该场景优先保障业务不中断，数据存在丢失的风险。</li> <li>● 生产受限：无法继续生产消息，但可以继续消费消息。该场景适用于对数据不能丢的业务场景，但是会导致生产业务失败。</li> </ul>
Kafka自动创建Topic	<p>选择是否开启Kafka自动创建Topic功能。</p> <p>开启Kafka自动创建Topic功能表示向一个未创建的Topic生产或消费消息时，系统会自动创建此Topic，此Topic的默认参数值如表5-6所示。</p> <p>集群实例如果在“配置参数”中修改“log.retention.hours”（老化时间）、“default.replication.factor”（副本数）或“num.partitions”（分区数）的参数值，此后自动创建的Topic参数值为修改后的参数值。例如：“num.partitions”修改为“5”，自动创建的Topic参数值如表5-6所示。</p>
标签	<p>标签用于标识云资源，当您拥有相同类型的许多云资源时，可以使用标签按各种维度（例如用途、所有者或环境）对云资源进行分类。</p> <p>如您的组织已经设定分布式消息服务Kafka版的相关标签策略，则需按照标签策略规则为Kafka实例添加标签。标签不符合标签策略的规则，则可能会导致Kafka实例创建失败，请联系组织管理员了解标签策略详情。</p> <ul style="list-style-type: none"> <li>● 如果您已经预定义了标签，在“标签键”和“标签值”中选择已经定义的标签键值对。另外，您可以单击“创建预定义标签”，跳转到标签管理服务页面，查看已经预定义的标签，或者创建新的标签。</li> <li>● 您也可以直接在“标签键”和“标签值”中设置标签。</li> </ul> <p>当前每个Kafka实例最多支持设置20个不同标签，标签的命名规格，请参考配置Kafka实例标签章节。</p>
Manager Username	<p>Kafka Manager名称支持自定义，但需要符合命名规则：长度为4~64个字符，由英文字母开头，只能由英文字母、数字、中划线、下划线组成。</p> <p>输入“manager-test”。</p>
Password	<p>设置Kafka Manager的密码。</p> <p>密码需要符合以下命名规则：</p> <ul style="list-style-type: none"> <li>● 长度为8~32个字符。</li> <li>● 至少包含以下字符中的3种：大写字母、小写字母、数字、特殊字符~!@#%&amp;*()-_+=\ [\{\}]:";'&lt;.&gt;?和空格，并且不能以-开头。</li> <li>● 不能与用户名或倒序的用户名相同。</li> </ul>

参数名称	说明
描述	设置实例的描述信息，长度为0~1024个字符。

表 5-6 Topic 参数值

参数名称	默认值	修改后的值
分区数	3	5
副本数	3	3
老化时间	72小时	72小时
同步复制	不开启	不开启
同步落盘	不开启	不开启
消息时间戳类型	CreateTime	CreateTime
批处理消息最大值	10485760字节	10485760字节

**步骤10** 单击“确认订单”，进入“确认配置”页面。

**步骤11** 确认实例信息无误后，单击“提交”。

**步骤12** 在实例列表页面，查看Kafka实例是否创建成功。

创建实例大约需要3到15分钟，此时实例的“状态”为“创建中”。

- 当实例的“状态”变为“运行中”时，说明实例创建成功。
- 当实例的“状态”变为“创建失败”，请参考[删除Kafka实例](#)，删除创建失败的实例，然后重新创建。如果重新创建仍然失败，请联系客服。

#### 说明

创建失败的实例，不会占用其他资源。

----结束

## 自定义购买 Kafka 集群实例

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择区域。

不同区域的云服务产品之间内网互不相通。请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击页面右上方的“购买Kafka实例”。

**步骤5** 在“自定义购买”页签中，设置实例基础配置。

**表 5-7** 实例基础配置参数说明

参数名称	说明
计费模式	<ul style="list-style-type: none"> <li>按需付费：一种后付费模式，即先使用再付费，按照Kafka实例实际使用时长计费，秒级计费，按小时结算。</li> </ul>
区域	不同区域的云服务产品之间内网互不相通。请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。
可用区	<p>可用区指在同一区域下，电力、网络隔离的物理区域，可用区之间内网互通，不同可用区之间物理隔离。</p> <p>根据实际情况选择1个或者3个及以上可用区。实例创建成功后，不支持修改可用区。</p>

**步骤6** 设置实例规格信息。

**表 5-8** 实例规格参数说明

参数名称	说明
版本	Kafka的版本号，支持1.1.0、2.3.0、2.7、3.x。 <b>实例创建后，版本号不支持修改。</b>
部署架构	选择“集群”。
代理规格	<p>请根据业务需求选择相应的代理规格。</p> <p>单个代理最大分区数*代理数量=实例分区数上限。当所有Topic的总分区数大于实例分区数上限时，创建Topic失败。</p>
代理数量	选择代理数量。
类型	选择存储Kafka数据的磁盘类型。 <b>Kafka实例创建后，磁盘类型不支持修改。</b>
容量	<p>选择存储Kafka数据的磁盘大小。</p> <p>存储空间用于存储消息（包括副本中的消息）、日志和元数据，建议根据业务消息体积、副本数量以及预留磁盘大小选择存储空间大小。每个Kafka代理会预留33GB的磁盘空间，用于存储日志和元数据。</p> <p>创建实例时会进行磁盘格式化，磁盘格式化会导致实际可用磁盘为总磁盘的93%~95%。</p>

图 5-1 实例规格

**Custom**

Specifications

**Default**

Version

**3.x** 2.7 2.3.0 1.1.0

Software version of the instance. This setting is fixed once the instance is created. Ideally, use server and client with the same version.

Instance Type

**Default**

Architecture

**Cluster**

Broker Flavor

Flavor Name	TPS Limit per Broker	Maximum Partitions per Broker	Recommended Consumer Gr...	Traffic per Broker (MB/s)
<input checked="" type="radio"/> kafka.2u4g.cluster	30,000	250	20	100
<input type="radio"/> kafka.4u8g.cluster	100,000	500	100	200
<input type="radio"/> kafka.8u16g.cluster	150,000	1,000	150	375
<input type="radio"/> kafka.12u24g.cluster	200,000	1,500	200	625
<input type="radio"/> kafka.16u32g.cluster	250,000	2,000	200	750

**Specifications** Flavor Name: kafka.2u4g.cluster | TPS Limit per Broker: 30,000 | Maximum Partitions per Broker: 250 | Recommended Consumer Groups per Broker: 20 | Traffic per Broker (MB/s): 100

Brokers

- 3 +

**Note**

For more about total traffic and service traffic, see . Reserve 30% of traffic to accommodate for fluctuations. Total network bandwidth: 300 MB/s; Recommended service bandwidth: 50 MB/s for read and 50 MB/s for write.

Storage Space per Broker

The system disk type and size **cannot be changed**. The storage space **cannot be decreased**. Select a disk type for your required service I/O. [Learn more](#)

Type Capacity

Ultra-high I/O - 100 +

Total storage space = 300 GB (3 brokers × 100 GB per broker) Disk bandwidth per broker = 170 MB/s. Total disk bandwidth = 510 MB/s (Disk bandwidth per broker × Number of brokers)

**步骤7 设置网络信息。**

表 5-9 实例网络参数说明

参数名称	说明
虚拟私有云	选择已经创建好的或共享的虚拟私有云。 虚拟私有云可以为您的Kafka实例构建隔离的、能自主配置和管理的虚拟网络环境。您可以单击右侧的“管理虚拟私有云”，跳转到“虚拟私有云”页面，查看或创建虚拟私有云。 虚拟私有云在Kafka实例创建完成后，不支持修改。
子网	选择已经创建好的子网。 子网在Kafka实例创建完成后，不支持修改。
IPv6	子网开启IPv6后，页面才显示此参数。开启IPv6后，客户端可以使用IPv6地址连接实例。 实例创建成功后，不支持修改IPv6开关。

参数名称	说明
内网IP地址	<p>根据实际需求选择“自动分配”或“手动选择”。</p> <ul style="list-style-type: none"> <li>自动分配：系统自动分配子网中可用的IP地址。</li> <li>手动选择：在下拉框中勾选指定的IP地址。如果指定的IP地址数量少于代理数量时，系统会自动为剩余代理分配IP地址。</li> </ul> <p>开启IPv6后，控制台不显示此参数。</p>
安全组	<p>选择已经创建好的安全组。</p> <p>安全组是一组对Kafka实例的访问规则的集合。您可以单击右侧的“创建安全组”，跳转到“安全组”页面，查看或创建安全组。</p> <p>客户端连接Kafka实例前，请根据不同的连接方式配置对应的安全组规则，否则会连接失败。具体的安全组规则请参考表 7-2。</p>

**步骤8** 设置实例访问方式。

表 5-10 实例访问方式参数说明

参数名称	说明
接入方式	<p>接入方式分为以下两种：</p> <ul style="list-style-type: none"> <li>明文接入：表示客户端连接Kafka实例时，无需进行SASL认证。</li> <li>密文接入：表示客户端连接Kafka实例时，需要进行SASL认证。开启“密文接入”后，需要设置SSL用户名、密码。</li> </ul> <p>内网访问不支持关闭，明文接入和密文接入至少开启一个。</p>
kafka安全协议	<ul style="list-style-type: none"> <li>SASL_SSL：采用SASL方式进行认证，数据通过SSL证书进行加密传输，安全性更高。</li> <li>SASL_PLAINTEXT：采用SASL方式进行认证，数据通过明文传输，性能更好。由于明文传输，建议使用SCRAM-SHA-512机制。</li> </ul>
SASL认证机制	<ul style="list-style-type: none"> <li>未勾选“PLAIN”时，使用SCRAM-SHA-512机制对账号密码进行认证。</li> <li>勾选“PLAIN”后，同时支持SCRAM-SHA-512机制和PLAIN机制，根据实际情况选择其中任意一种配置连接。</li> </ul> <p>密文接入成功开启后，SASL PLAIN机制不支持修改。</p> <p><b>什么是SCRAM-SHA-512机制和PLAIN机制？</b></p> <ul style="list-style-type: none"> <li>SCRAM-SHA-512机制：采用哈希算法对用户名与密码生成凭证，进行身份校验的安全认证机制，比PLAIN机制安全性更高。</li> <li>PLAIN机制：一种简单的用户名密码校验机制。</li> </ul>

参数名称	说明
SSL用户名	<p>客户端用于连接Kafka实例的用户名。</p> <p>用户名需要符合以下命名规则：由英文字母开头，且只能由英文字母、数字、中划线、下划线组成，长度为4~64个字符。</p> <p>密文接入成功开启后，用户名不支持修改。</p>
密码	<p>客户端用于连接Kafka实例的密码。</p> <p>密码需要符合以下规则：</p> <ul style="list-style-type: none"> <li>• 长度为8~32个字符。</li> <li>• 至少包含以下字符中的3种：大写字母、小写字母、数字、特殊字符`~!@#%\$^&amp;*()-_+=+\\[{}];:","&lt;.&gt;?和空格，并且不能以-开头。</li> <li>• 不能与用户名或倒序的用户名相同。</li> </ul>

**步骤9** 设置高级配置。

**表 5-11** 高级配置参数说明

参数名称	说明
实例名称	<p>名称支持自定义，但需要符合命名规则：长度为4~64个字符，由英文字母开头，只能由英文字母、数字、中划线、下划线组成。</p>
企业项目	<p>该参数针对企业用户使用。</p> <p>企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理，默认项目为default。</p>
容量阈值策略	<p>设置磁盘使用达到容量阈值后的消息处理策略，容量阈值为95%。</p> <ul style="list-style-type: none"> <li>• 自动删除：可以正常生产和消费消息，但是会删除最早的10%的消息，以保证磁盘容量充足。该场景优先保障业务不中断，数据存在丢失的风险。</li> <li>• 生产受限：无法继续生产消息，但可以继续消费消息。该场景适用于对数据不能丢的业务场景，但是会导致生产业务失败。</li> </ul>
Kafka自动创建Topic	<p>选择是否开启Kafka自动创建Topic功能。</p> <p>开启Kafka自动创建Topic功能表示向一个未创建的Topic生产或消费消息时，系统会自动创建此Topic，此Topic的默认参数值如表5-12所示。</p> <p>集群实例如果在“配置参数”中修改“log.retention.hours”（老化时间）、“default.replication.factor”（副本数）或“num.partitions”（分区数）的参数值，此后自动创建的Topic参数值为修改后的参数值。例如：“num.partitions”修改为“5”，自动创建的Topic参数值如表5-12所示。</p>

参数名称	说明
标签	<p>标签用于标识云资源，当您拥有相同类型的许多云资源时，可以使用标签按各种维度（例如用途、所有者或环境）对云资源进行分类。</p> <p>如您的组织已经设定分布式消息服务Kafka版的相关标签策略，则需按照标签策略规则为Kafka实例添加标签。标签如果不符合标签策略的规则，则可能会导致Kafka实例创建失败，请联系组织管理员了解标签策略详情。</p> <ul style="list-style-type: none"> <li>如果您已经预定义了标签，在“标签键”和“标签值”中选择已经定义的标签键值对。另外，您可以单击“查看预定义标签”，跳转到标签管理服务页面，查看已经预定义的标签，或者创建新的标签。</li> <li>您也可以直接在“标签键”和“标签值”中设置标签。</li> </ul> <p>当前每个Kafka实例最多支持设置20个不同标签，标签的命名规格，请参考<a href="#">配置Kafka实例标签</a>章节。</p>
Manager Username	<p>Kafka Manager名称支持自定义，但需要符合命名规则：长度为4~64个字符，由英文字母开头，只能由英文字母、数字、中划线、下划线组成。</p> <p>输入“manager-test”。</p>
Password	<p>设置Kafka Manager的密码。</p> <p>密码需要符合以下命名规则：</p> <ul style="list-style-type: none"> <li>长度为8~32个字符。</li> <li>至少包含以下字符中的3种：大写字母、小写字母、数字、特殊字符`~!@#\$%^&amp;*()-_+=\ {[];:~",&lt;.&gt;?`和空格，并且不能以-开头。</li> <li>不能与用户名或倒序的用户名相同。</li> </ul>
描述	<p>设置实例的描述信息，长度为0~1024个字符。</p>

表 5-12 Topic 参数值

参数名称	默认值	修改后的值
分区数	3	5
副本数	3	3
老化时间	72小时	72小时
同步复制	不开启	不开启
同步落盘	不开启	不开启
消息时间戳类型	CreateTime	CreateTime
批处理消息最大值	10485760字节	10485760字节

**步骤10** 在页面右侧“配置概要”中，查看已选择的实例配置。

**步骤11** 确认配置无误后，单击“确认订单”，进入“确认配置”页面。

**步骤12** 确认实例信息无误后，单击“提交”。

**步骤13** 在实例列表页面，查看Kafka实例是否创建成功。

创建实例大约需要3到15分钟，此时实例的“状态”为“创建中”。

- 当实例的“状态”变为“运行中”时，说明实例创建成功。
- 当实例的“状态”变为“创建失败”，请参考[删除Kafka实例](#)，删除创建失败的实例，然后重新创建。如果重新创建仍然失败，请联系客服。

#### 说明

创建失败的实例，不会占用其他资源。

----结束

## 购买相同配置的 Kafka 实例

若您需购买与当前Kafka实例配置一致的新实例，可通过“购买相同配置”功能，一键复用当前实例配置，快速完成新实例购买，大幅节省配置时间。

**步骤1** 登录管理控制台。

**步骤2** 在左上角单击  ，选择区域。

不同区域的云服务产品之间内网互不相通。请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。

**步骤3** 在管理控制台左上角单击  ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 选择目标Kafka实例，单击“操作”列下的“更多 > 购买相同配置”，进入“购买Kafka实例”页面。

**步骤5** 系统已自动复制目标Kafka实例的参数配置，根据需要调整待购买Kafka实例的参数配置。参数配置的详细内容，请参见[自定义购买Kafka集群实例](#)。

为安全起见，当复制的Kafka实例存在如下场景时，系统不会自动复制相应的参数配置，需要您手动配置：

- 开启密文接入的Kafka实例，需要重新设置SSL用户名和密码。
- 开启公网访问的Kafka实例，需要重新设置公网IP地址。
- 指定内网IP地址的Kafka实例，需要在“内网IP地址”中重新设置为“手动选择”，并指定相应的内网IP地址。
- 目标Kafka实例的名称不会被复制，需要重新设置Kafka实例名称。

**步骤6** 在页面右侧“配置概要”中，查看已选择的实例配置。

**步骤7** 确认配置无误后，单击“确认订单”，进入“确认配置”页面。

**步骤8** 确认实例信息无误后，单击“提交”。

**步骤9** 在实例列表页面，查看Kafka实例是否创建成功。

创建实例大约需要3到15分钟，此时实例的“状态”为“创建中”。

- 当实例的“状态”变为“运行中”时，说明实例创建成功。
- 当实例的“状态”变为“创建失败”，请参考[删除Kafka实例](#)，删除创建失败的实例，然后重新创建。如果重新创建仍然失败，请联系客服。

#### 说明

创建失败的实例，不会占用其他资源。

----结束

# 6 配置 Topic

## 6.1 创建 Kafka Topic

Topic，即消息主题，用于存储消息，供生产者生产消息以及消费者订阅消息。创建 Kafka 实例成功后，如果未开启“Kafka 自动创建 Topic”功能，则需要手动创建 Topic。如果实例开启了“Kafka 自动创建 Topic”，则该操作为可选。

“Kafka 自动创建 Topic”表示向一个未创建的 Topic 生产或消费消息时，系统会自动创建此 Topic，此 Topic 的默认参数值如表 6-1 所示。

如果在“配置参数”中修改“log.retention.hours”（老化时间）、“default.replication.factor”（副本数）或“num.partitions”（分区数）的参数值，此后自动创建的 Topic 参数值为修改后的参数值。例如：“num.partitions”修改为“5”，自动创建的 Topic 参数值如表 6-1 所示。

表 6-1 Topic 参数值

参数名称	默认值	修改后的值
分区数	3	5
副本数	3	3
老化时间	72小时	72小时
同步复制	不开启	不开启
同步落盘	不开启	不开启
消息时间戳类型	CreateTime	CreateTime
批处理消息最大值	10485760字节	10485760字节

本文主要介绍手动创建 Topic 的操作，有以下几种方式，您可以根据实际情况选择任意一种方式：

- [在控制台创建 Topic](#)

- [在Kafka Manager创建Topic](#)
- [在客户端创建Topic](#)

## 约束与限制

- Kafka实例对Topic的总分区数设置了上限，**当Topic的总分区数达到上限后，用户就无法继续创建Topic**。不同规格配置的Topic总分区数不同，具体请参考[Kafka产品规格说明](#)。
- 已开启密文接入的实例，“allow.everyone.if.no.acl.found”为“false”时，初始用户（即首次开启密文接入时设置的用户）可以通过客户端创建Topic，其他用户无法通过客户端创建Topic。
- Topic名称开头包含特殊字符，例如“#”号时，监控数据无法展示。
- 由于Kafka内核限制，无法区分句点和下划线，因此无法创建名称中只有句点和下划线不同的Topic。例如，已创建Topic\_1，在新创建Topic.1时，报错如下：Topic 'topic.1' collides with existing topics: topic\_1。

## 在控制台创建 Topic

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > Topic管理”，单击“创建Topic”，弹出“创建Topic”对话框。

**步骤6** 填写Topic名称和配置信息，单击“确定”，完成创建Topic。

图 6-1 创建 Topic

### 创建Topic

Topic 名称	<input type="text" value="topic-1161178702"/>
分区数 <span>?</span>	<input type="button" value="-"/> <input type="text" value="3"/> <input type="button" value="+"/> 取值范围：1-200 Topic创建后，分区数不支持扩容。
副本数 <span>?</span>	<input type="button" value="-"/> <input type="text" value="3"/> <input type="button" value="+"/> 取值范围：1-3，建议取3副本 副本数需要小于等于代理个数。
老化时间（小时）	<input type="button" value="-"/> <input type="text" value="72"/> <input type="button" value="+"/> 取值范围：1-720 Topic中消息的过期时间，超过时间的消息将被删除，无法被消费。
同步复制 <span>?</span>	<input type="checkbox"/>
同步落盘 <span>?</span>	<input type="checkbox"/>
消息时间戳类型 <span>?</span>	<input type="text" value="LogAppendTime"/>
批处理消息最大值（字节） <span>?</span>	<input type="button" value="-"/> <input type="text" value="10,485,760"/> <input type="button" value="+"/>

表 6-2 Topic 参数说明

参数	说明
Topic名称	<p>名称支持自定义，但需要符合命名规则：以英文字母、数字、下划线开头，且只能由英文字母、数字、句点、中划线、下划线组成，长度为3~200个字符。</p> <p>名称不能与以下内置Topic名称一致：</p> <ul style="list-style-type: none"> <li>• <code>__consumer_offsets</code></li> <li>• <code>__transaction_state</code></li> <li>• <code>__trace</code></li> <li>• <code>__connect-status</code></li> <li>• <code>__connect-configs</code></li> <li>• <code>__connect-offsets</code></li> <li>• <code>__cloud_event</code></li> <li>• <code>__site_event</code></li> <li>• <code>__dms_dial_test</code></li> <li>• <code>__remote_log_metadata</code></li> <li>• <code>__remote_log_compact_metadata</code></li> <li>• <code>__remote_log_metadata_offset</code></li> </ul> <p>创建Topic后不能修改名称。</p> <p>由于Kafka内核限制，无法区分句点和下划线，因此无法创建名称中只有句点和下划线不同的Topic。例如，已创建Topic_1，在新创建Topic.1时，报错如下：Topic 'topic.1' collides with existing topics: topic_1。</p>
分区数	<p>Topic的分区数。</p> <p>如果分区数与消费者数一致，分区数越大消费的并发度越大。</p> <p>该参数设置为1时，消费消息时会按照先入先出的顺序进行消费。</p> <p>取值范围：1~200。</p>
副本数	<p>您可以为每个Topic设置副本的数量，Kafka会自动在每个副本上备份数据，当其中一个Broker故障时数据依然是可用的，副本数越大可靠性越高。</p> <p>该参数设置为1时，表示只有一份数据。</p> <p>取值范围：1~实例的代理数量。</p> <p><b>说明</b> 实例节点出现故障的情况下，单副本Topic查询消息时可能会报“内部服务错误”，因此不建议使用单副本Topic。</p>
老化时间（小时）	<p>消息的最长保留时间，消费者必须在此时间结束前消费消息，否则消息将被删除。删除的消息，无法被消费。</p> <p>取值范围：1~720。</p>

参数	说明
同步复制	表示后端收到生产消息请求并复制给所有副本后，才返回客户端。 开启同步复制后，需要在生产者客户端的配置文件或者生产代码中配置acks=all或者-1，否则无效。 当副本数为1时，不能选择同步复制功能。
同步落盘	表示生产的每条消息都会立即写入磁盘，可靠性更高。关闭同步落盘后，生产的消息存在内存中，不会立即写入磁盘。
消息时间戳类型	定义消息中的时间戳类型，取值如下： <ul style="list-style-type: none"> <li>• CreateTime：生产者创建消息的时间。</li> <li>• LogAppendTime：broker将消息写入日志的时间。</li> </ul>
批处理消息最大值（字节）	Kafka允许的最大批处理大小，如果在生产客户端配置文件或代码中启用消息压缩，则表示压缩后的最大批处理大小。 如果增加“批处理消息最大值”，且存在消费者版本早于0.10.2，此时消费者的“fetch size”值也必须增加，以便消费者可以获取增加后的批处理大小。 取值范围：0~10485760。

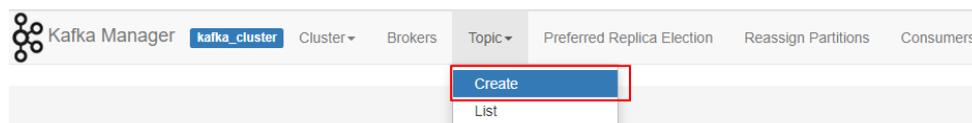
Topic创建完成后，在Topic列表页面查看新创建的Topic。

---结束

## 在 Kafka Manager 创建 Topic

登录Kafka Manager后，在页面顶部选择“Topic > Create”，然后按照界面参数填写即可。出于性能考虑，建议单个Topic的分区数设置为200以内。

图 6-2 在 Kafka Manager 中创建 Topic



## 在客户端创建 Topic

Kafka客户端版本为2.2以上时，支持通过kafka-topics.sh创建Topic，以及管理Topic的各类参数。

- 未开启密文接入的Kafka实例，在Kafka客户端的“/bin”目录下，通过以下命令创建Topic。  

```
./kafka-topics.sh --create --topic {topic-name} --bootstrap-server {connection-address} --partitions {number-of-partitions} --replication-factor {number-of-replicas}
```

表 6-3 创建 Topic 参数说明

参数名称	说明
topic-name	Topic名称，支持自定义。
connection-address	Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。
number-of-partitions	Topic的分区数。 出于性能考虑，建议单个Topic的分区数设置为 <b>200以内</b> 。
number-of-replicas	Topic的副本数。

示例如下：

```
[root@ecs-kafka bin]# ./kafka-topics.sh --create --topic topic-01 --bootstrap-server
192.168.xx.xx:9092,192.168.xx.xx:9092,192.168.xx.xx:9092 --partitions 3 --replication-factor 3
Created topic topic-01.
[root@ecs-kafka bin]#
```

- 已开启密文接入的Kafka实例，通过以下步骤创建Topic。
  - (可选) 如果已经设置了用户名和密码，以及SSL证书配置，请跳过此步骤，执行**b**。否则请执行以下操作。

在Kafka客户端的“/config”目录中创建“ssl-user-config.properties”文件，参考**步骤3**增加用户名和密码，以及SSL证书配置。

- 在Kafka客户端的“/bin”目录下，通过以下命令创建Topic。

```
./kafka-topics.sh --create --topic {topic-name} --bootstrap-server {connection-address} --
partitions {number-of-partitions} --replication-factor {number-of-replicas} --command-config ../
config/{ssl-user-config.properties}
```

表 6-4 创建 Topic 参数说明

参数名称	说明
topic-name	Topic名称，支持自定义。
connection-address	Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。
number-of-partitions	Topic的分区数。 出于性能考虑，建议单个Topic的分区数设置为 <b>200以内</b> 。
number-of-replicas	Topic的副本数。
ssl-user-config.properties	配置文件的名称，此配置文件用于存放用户名和密码，以及SSL证书配置信息。

示例如下：

```
[root@ecs-kafka bin]# ./kafka-topics.sh --create --topic topic-01 --bootstrap-server
192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --partitions 3 --replication-factor 3 --
command-config ../config/ssl-user-config.properties
```

```
Created topic topic-01.  
[root@ecs-kafka bin]#
```

## 6.2 设置 Kafka Topic 权限

Kafka实例开启密文接入后，支持对Topic进行ACL（Access Control List）权限管理，您可以在Topic中为不同的用户设置不同的生产消费消息的权限，以达到用户之间的权限隔离。

本章节主要介绍Kafka实例开启密文接入后，如何对Topic进行用户授权。

### 约束与限制

- 如果“allow.everyone.if.no.acl.found”为“true”，且Topic未授权任何用户，此时所有的用户都可以订阅/发布此Topic。如果Topic已授权某一个或多个用户，此时只有授权的用户可以订阅/发布此Topic，其他未授权的用户不能订阅/发布此Topic。“allow.everyone.if.no.acl.found”在[配置参数](#)中修改。
- 如果“allow.everyone.if.no.acl.found”为“false”，初始用户（即创建实例时设置的用户）拥有订阅/发布Topic的权限，其他用户在未授权时不能订阅/发布Topic。“allow.everyone.if.no.acl.found”在[配置参数](#)中修改。
- 如果Topic同时设置了“默认用户”权限和单个用户权限，取两者的并集。
- 批量设置Topic权限后，会覆盖Topic之前的权限。
- 修改Topic权限过程中，可能存在权限短暂失效的情况，客户端报“AuthorizationException”错误，建议在客户端设置重试机制，具体方法请参考[Kafka客户端使用建议](#)。

### 前提条件

- Kafka实例已开启密文接入。
- 已[创建用户](#)。

### 查看 Topic 的权限信息

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 单击Topic名称，进入Topic详情页面。

**步骤7** 单击“用户权限”，进入“用户权限”页签。

**步骤8** 查看该Topic的所有授权用户及其对应权限。

表 6-5 用户权限

参数名称	说明
用户名	拥有发布/订阅Topic权限的用户。
权限	该用户对应的权限。

----结束

## 设置单个 Topic 权限

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“实例管理 > Topic管理”，进入Topic列表页面。

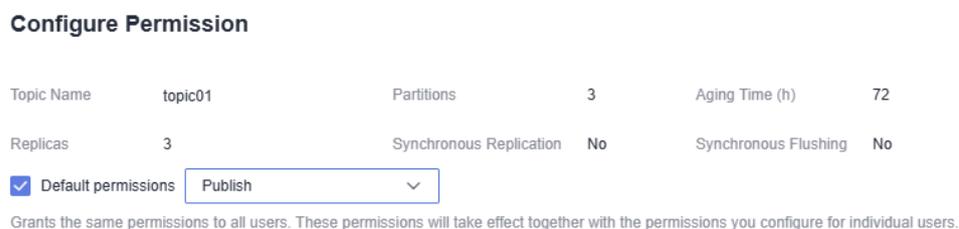
**步骤6** 通过以下任意一种方法，进入设置用户权限页面。

- 在待设置用户权限的Topic所在行，单击“用户权限”，弹出“用户权限”对话框。
- 单击待设置用户权限的Topic名称，进入Topic详情页面。单击右上角的“设置权限”，弹出“用户权限”对话框。
- 单击待设置用户权限的Topic名称，进入Topic详情页面。单击“用户权限”，进入“用户权限”页签。单击“设置权限”，弹出“用户权限”对话框。

**步骤7** 为用户设置Topic权限。

- 如果需要为所有用户设置相同的权限，勾选“默认用户”，并设置权限。如下图所示，所有用户都拥有发布此Topic的权限。

图 6-3 为所有用户设置相同的权限



- 如果需要为某个用户设置权限，不勾选“默认用户”，在“用户列表”区域，勾选需要订阅/发布/发布+订阅此Topic的用户名。如果用户过多，在搜索框中输入用户名，快速查找用户。在“Topic权限”区域，为用户设置不同的权限。如下图所示，此时只有“test”、“send”和“receive”用户可以订阅/发布此Topic，“send\_receive”用户无法订阅/发布此Topic。

图 6-4 为某个用户设置权限

The screenshot shows two panels. The left panel, titled 'Users Select Users', has a search input 'Enter a username.' and a list of users with checkboxes: 'receive' (checked), 'test' (checked), 'send' (checked), and 'send\_receive' (unchecked). The right panel, titled 'Topic Permissions', has a search input 'Enter a username.' and a table with columns 'Username', 'Permission', and 'Operation'. The table contains three rows: 'send' with 'Publish' permission and 'Delete' operation; 'test' with 'Publish/Subscribe' permission and 'Delete' operation; and 'receive' with 'Subscribe' permission and 'Delete' operation.

同时设置了“默认用户”权限和单个用户权限时，取两者的并集。如下图所示，“test”和“receive”用户可以订阅+发布此Topic，其他用户只能发布此Topic。

图 6-5 为用户设置 Topic 权限

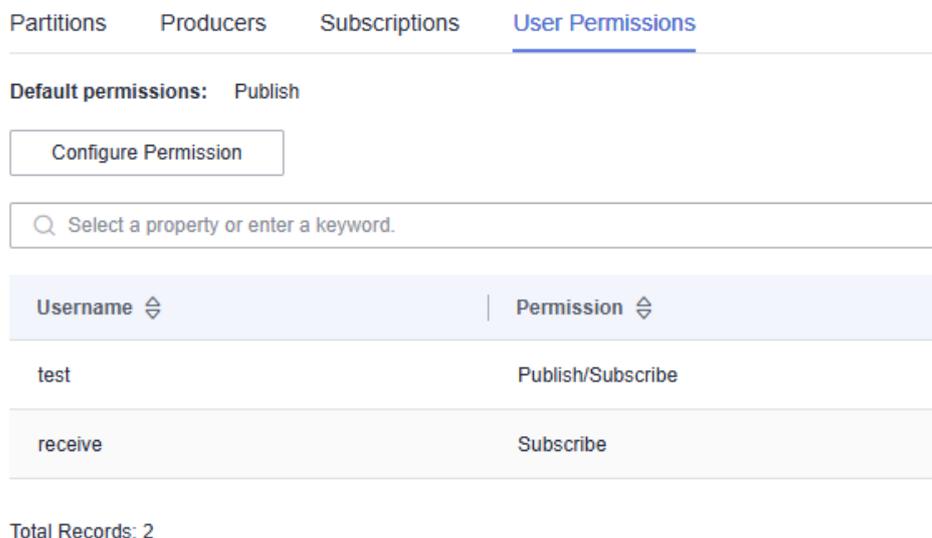
The screenshot shows two panels. The top panel has a checked box for 'Default permissions' and a dropdown menu set to 'Publish'. Below it is a text note: 'Grants the same permissions to all users. These permissions will take effect together with the permissions you configure for individual users.' The bottom panel, titled 'Users Select Users', has a search input 'Enter a username.' and a list of users with checkboxes: 'receive' (checked), 'test' (checked), 'send' (unchecked), and 'send\_receive' (unchecked). The right panel, titled 'Topic Permissions', has a search input 'Enter a username.' and a table with columns 'Username', 'Permission', and 'Operation'. The table contains two rows: 'test' with 'Publish/Subscribe' permission and 'Delete' operation; and 'receive' with 'Subscribe' permission and 'Delete' operation.

**步骤8** 在“用户权限”对话框底部，单击“一键输入”，系统自动在输入框中输入“MODIFY”，确认修改权限。然后单击“确定”，完成为用户的授权。

**步骤9** 验证权限设置是否正确。

1. 单击已设置用户权限的Topic名称，进入Topic详情页面。
2. 单击“用户权限”，进入“用户权限”页签。
3. 查看已设置的用户权限。

图 6-6 查看已经授权的用户及其对应的权限



----结束

## 批量设置 Topic 权限

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 在Topic名称左侧的方框中，勾选待设置用户权限的Topic，单击页面上方的“设置用户权限”，弹出“批量设置权限”对话框。

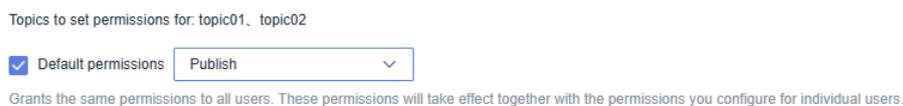
**步骤7** 为用户批量设置Topic权限。

**批量设置权限页面不展示已设置的权限，且设置之后会覆盖之前的权限。**例如：

“test”用户初始拥有“发布+订阅”Topic01权限，在批量设置Topic权限时，将“test”用户设置为“发布”权限，最终“test”用户只拥有“发布”Topic01权限。

- 如果需要为所有用户设置相同的权限，勾选“默认用户”，并设置权限。如下图所示，所有用户都拥有发布topic01和topic02的权限。

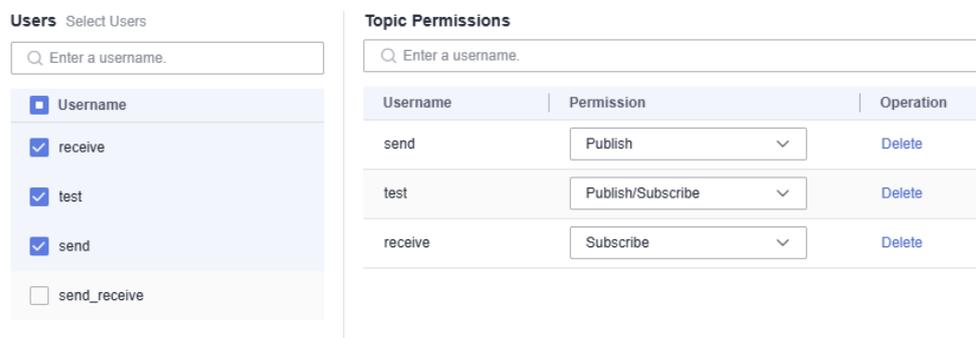
图 6-7 为所有用户设置相同的权限



- 如果需要为某个用户设置权限，不勾选“默认用户”，在“用户列表”区域，勾选待授权的用户名。如果用户过多，在搜索框中输入用户名，快速查找用户。在“Topic权限”区域，为用户设置不同的权限。如下图所示，此时只有“test”、

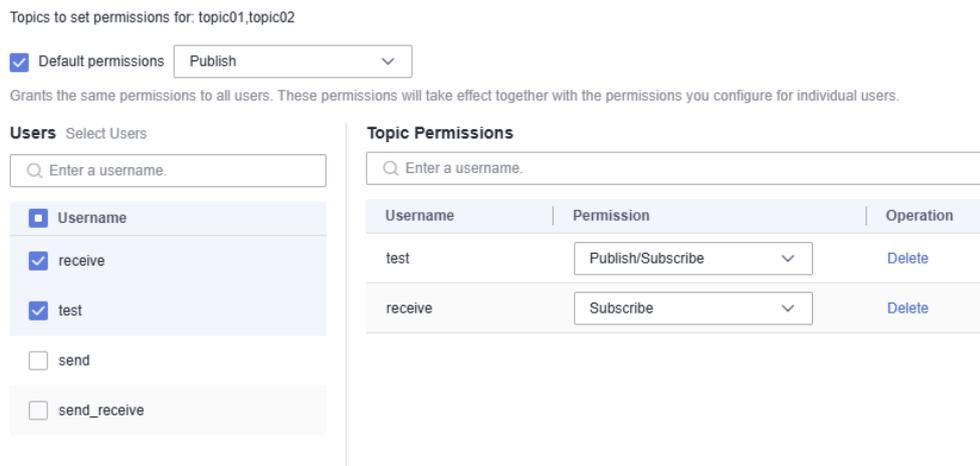
“send”和“receive”用户可以订阅/发布topic01和topic02，“send\_receive”用户无法订阅/发布topic01和topic02。

图 6-8 为多个用户设置权限



同时设置了“默认用户”权限和单个用户权限时，取两者的并集。如下图所示，“test”和“receive”用户可以订阅+发布topic01和topic02，其他用户只能发布topic01和topic02。

图 6-9 为用户设置 Topic 权限

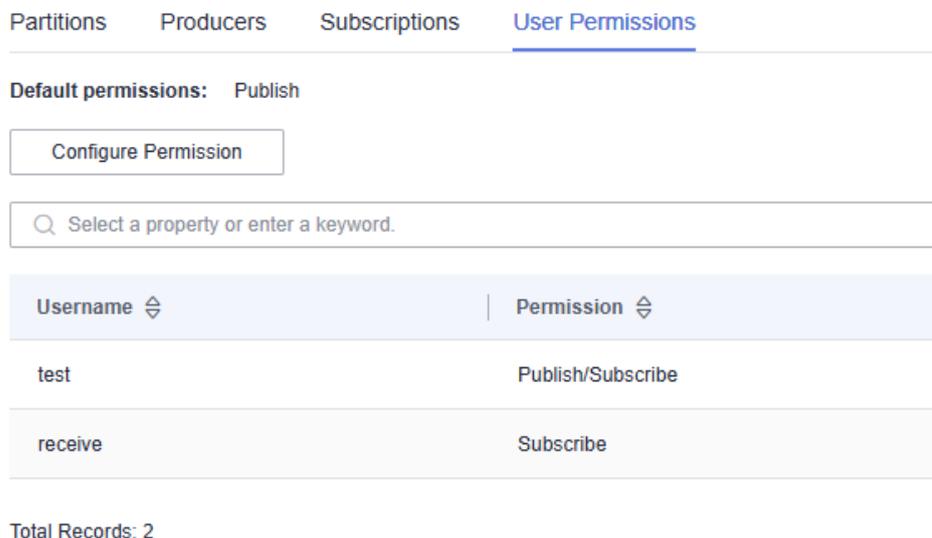


**步骤8** 在“批量设置权限”对话框底部，单击“一键输入”，系统自动在输入框中输入“MODIFY”，确认修改权限。然后单击“确定”，完成为用户的授权。

**步骤9** 验证权限设置是否正确。

1. 单击已设置用户权限的Topic名称，进入Topic详情页面。
2. 单击“用户权限”，进入“用户权限”页签。
3. 查看已设置的用户权限。

图 6-10 查看已经授权的用户及其对应的权限



----结束

## 删除单个 Topic 权限

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 通过以下任意一种方法，进入设置用户权限页面。

- 在待删除用户权限的Topic所在行，单击“用户权限”，弹出“用户权限”对话框。
- 单击待删除用户权限的Topic名称，进入Topic详情页面。单击右上角的“设置权限”，弹出“用户权限”对话框。
- 单击待删除用户权限的Topic名称，进入Topic详情页面。单击“用户权限”，进入“用户权限”页签。单击“设置权限”，弹出“用户权限”对话框。

**步骤7** 在“Topic权限”区域，在待删除权限的用户所在行，单击“删除”。

**步骤8** 在“用户权限”对话框底部，单击“一键输入”，系统自动在输入框中输入“MODIFY”，确认修改权限。然后单击“确定”。

**步骤9** 验证权限是否已删除。

1. 单击已删除用户权限的Topic名称，进入Topic详情页面。
2. 单击“用户权限”，进入“用户权限”页签。
3. 已删除的用户不再显示在“用户权限”中，表明权限已成功删除。

----结束

## 6.3 Topic 管理

### 6.3.1 查看 Kafka Topic 详情

在Topic详情页面，您可以查看其基本信息、分区信息、生产者信息和订阅关系。

#### 查看 Topic 详情

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 单击Topic名称，进入Topic详情页面。

Topic详情页面显示**Topic基本信息**、**分区信息**、**生产者信息**和**订阅关系**。

----结束

#### Topic 基本信息

图 6-11 Topic 基本信息

Topic 名称	topic-1128625419	关联代理	broker-0,broker-1,broker-2
分区数	3	创建时间	2025/07/03 16:24:30 GMT+08:00

表 6-6 Topic 基本信息

参数名称	说明
Topic名称	该Topic的名称。
关联代理	该Topic已经部署到关联代理中。
分区数	该Topic的分区数。
创建时间	<p>该Topic的创建时间。</p> <p><b>注意</b></p> <p>创建Topic的时间或方式满足以下任意一种情况时，Topic详情页面不显示创建时间。</p> <ul style="list-style-type: none"> <li>很久前创建的Topic，在Topic详情页面不显示创建时间，具体以控制台显示为准。</li> <li>自动创建的Topic、在客户端使用命令行创建的Topic、以及在客户端通过代码创建的Topic。</li> </ul>

## 分区信息

图 6-12 分区信息

Partition	Minimum Offset	First Updated	Maximum Offset	Last Updated	Messages
0	0	Jul 01, 2025 20:28:52 GMT...	1	Jul 01, 2025 20:28:58 GMT...	2
1	0	--	0	--	0
2	0	Jul 01, 2025 20:28:38 GMT...	0	Jul 01, 2025 20:28:38 GMT...	1

表 6-7 Topic 的分区信息

参数名称	说明
分区编号	该Topic的分区号。
最小偏移量	该分区的最小偏移量。
最早更新时间	该分区下最早一条消息的更新时间。 <b>注意</b> 当分区中的消息都已老化，或尚未生产消息时，“最早更新时间”显示“--”。
最大偏移量	该分区的最大偏移量。
最近更新时间	该分区下最近一条消息的更新时间。 <b>注意</b> 当分区中的消息都已老化，或尚未生产消息时，“最近更新时间”显示“--”。
消息数	该分区中的消息数。

## 生产者信息

**注意**

很久前创建的Topic，在Topic详情页面不显示“生产者”页签，具体以控制台显示为准。

图 6-13 生产者信息

Broker Address	Producer Address	Producer Connected
192.168.0.79:9094	1: 41:41092	Jul 01, 2025 20:40:14 GMT+08:00

表 6-8 Topic 的生产者信息

参数名称	说明
Broker地址	生产者连接的Kafka实例Broker地址。
生产者地址	生产者客户端的地址。
加入时间	生产者连接Kafka实例的时间。

## 订阅关系



**警告**

如果一个实例中消费组数量超过10000，则不支持查询Topic的订阅关系。

图 6-14 订阅关系

Consumer Group Name	Status	Coordinator (ID)	Accumulated Messages
test	STABLE	0	0

表 6-9 Topic 的订阅关系

参数名称	说明
消费组名称	订阅该Topic的消费组名称。 单击消费组名称，跳转到对应消费组的详情页面，查看消费者列表和消费进度。
消费组状态	消费组当前的状态。 <ul style="list-style-type: none"> <li>DEAD：消费组内没有任何成员，且没有任何元数据。</li> <li>EMPTY：消费组内没有任何成员，存在元数据。</li> <li>PREPARING_REBALANCE：准备开启Rebalance。</li> <li>COMPLETING_REBALANCE：所有成员加入消费组。</li> <li>STABLE：消费组内成员可以正常消费。</li> </ul>
Coordinator(ID)	Coordinator组件所在的代理。
堆积数	消费组中剩余可以消费的消息个数。

### 6.3.2 查看 Kafka Topic 日志

Topic日志记录Topic分区选举Leader的详情，包括选举Leader时间、选举Leader的Topic分区和Leader的选举次数等。本章节介绍如何在控制台查看Topic日志。

Topic日志存储在云日志服务（Log Tank Service，简称LTS）中，由LTS提供查询功能。

## 约束与限制

- 老实例不支持Topic日志功能，具体以控制台为准。
- Topic日志默认存储时间为七天，如果需要延长保留天数，可以在[LTS控制台修改日志组的存储时间](#)。
- Topic日志开启后会在LTS控制台创建对应的日志组、日志流和仪表盘。
- 频繁产生Topic日志，可能会影响实例性能。
- 不同Kafka实例使用相同的日志组和日志流保存Topic日志时，在每个Kafka实例中都能看到使用同一个日志组和日志流的所有Kafka实例的Topic日志。

## 前提条件

- 开启Topic日志需要您的账号拥有LTS服务的创建日志组和日志流的权限。
- Kafka实例处于“运行中”时，才能开启/关闭Topic日志。

## 开启 Topic 日志

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“分析与诊断 > Topic日志”，进入“Topic日志”页面。

**步骤6** 单击“开启日志”，弹出“开启日志”对话框。

**步骤7** 单击“确定”，弹出“日志配置”对话框。

**步骤8** 根据实际情况决定是否开启日志配置，单击“确定”。自动跳转到“后台任务管理”页面，开启日志的任务状态为“成功”时，表示开启Topic日志成功。

- 不开启：LTS后台会自动创建一个日志组以及对应的日志流。
- 开启：选择存储日志文件“topic.log”的日志组和日志流。您还可以单击右侧的“查看日志组”，跳转到LTS控制台，查看或创建日志组和日志流。

----结束

## 查看 Topic 日志

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“分析与诊断 > Topic日志”，进入“Topic日志”页面。

**步骤6** 在“日志查询”页签，查看Topic日志。

如果您需要搜索日志，请参考[进入搜索LTS日志页面](#)进行操作。

Topic日志示例如下：

```
{
  "level": "INFO",
  "timestamp": "2024-12-27 17:26:13,361",
  "message": {
    "topicPartition": "topic-0",
    "targetState": "OnlinePartition",
    "leaderAndIsr": "LeaderAndIsr(leader=1, leaderEpoch=3, isr=List(1, 0),
leaderRecoveryState=RECOVERED, partitionEpoch=3)",
    "partitionState": "OnlinePartition",
    "topic": "topic",
    "type": "ELECT_LEADER"
  }
}

{
  "level": "INFO",
  "timestamp": "2024-12-27 17:26:13,491",
  "message": {
    "leader": "1",
    "startOffset": "0",
    "topic": "topic",
    "type": "MAKE_LEADER",
    "topicPartition": "topic-0",
    "epoch": "3"
  }
}
```

参数说明如表6-10所示。

**表 6-10** Topic 日志参数说明

参数名称	说明
level	Topic日志的等级，只有“INFO”一种等级。
timestamp	Topic分区选举Leader或确定Leader的时间。
topicPartition	Topic分区。
targetState	目标状态，取值如下： <ul style="list-style-type: none"> <li>● NewPartition：表示分区处于新建状态。</li> <li>● OnlinePartition：表示分区处于正常工作状态。</li> <li>● OfflinePartition：表示分区处于下线状态。</li> <li>● NonExistentPartition：表示分区不存在或被删除。</li> </ul>
leaderAndIsr	leaderAndIsr请求的信息。

参数名称	说明
partitionState	分区状态，取值如下： <ul style="list-style-type: none"><li>• NewPartition：表示分区处于新建状态。</li><li>• OnlinePartition：表示分区处于正常工作状态。</li><li>• OfflinePartition：表示分区处于下线状态。</li><li>• NonExistentPartition：表示分区不存在或被删除。</li></ul>
topic	Topic名称。
type	Leader所处的阶段，取值如下： <ul style="list-style-type: none"><li>• ELECT_LEADER：选举Leader。</li><li>• MAKE_LEADER：确定Leader。</li></ul>
leader	Leader所在分区。
startOffset	Leader在对应Epoch上写入第一条消息的Offset。 每个Epoch对应一个startOffset。
epoch	Leader的选举次数，初始值为0。Leader每发生一次选举，Epoch值加一。

----结束

## 关闭 Topic 日志

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“分析与诊断 > Topic日志”，进入“Topic日志”页面。

**步骤6** 在页面右上角单击“关闭日志”，弹出确认关闭对话框。

**步骤7** 单击“确定”，自动跳转到“后台任务管理”页面，“状态”为“成功”时，表示关闭Topic日志成功。

关闭Topic日志只是停止Topic日志上报功能，LTS控制台的日志组和日志流仍然保留，还会继续收费，如不需要保留该日志，可以在LTS控制台删除对应的[日志组](#)和[日志流](#)。

----结束

## 6.3.3 修改 Kafka Topic 配置

本章节指导您在控制台修改如[表6-11](#)所示Kafka Topic的配置。

修改同步复制、同步落盘、消息时间戳类型和批处理消息最大值，不会重启实例。

表 6-11 Kafka Topic 的配置参数

参数名称	说明
分区数	Topic的分区数，修改方法请参见 <a href="#">修改Kafka Topic分区数</a> 。
老化时间	消息的最长保留时间，修改方法请参见 <a href="#">修改Kafka消息老化时间</a> 。
副本数	Topic每个分区的副本数量，修改方法请参见 <a href="#">修改Kafka Topic副本数</a> 。
同步复制	后端收到生产消息请求并复制给所有副本后，才返回客户端。
同步落盘	<ul style="list-style-type: none"> <li>开启：生产的每条消息都会立即写入磁盘，可靠性更高。</li> <li>关闭：生产的消息存在内存中，不会立即写入磁盘。</li> </ul>
消息时间戳类型	定义消息中的时间戳类型，取值如下： <ul style="list-style-type: none"> <li>CreateTime：生产者创建消息的时间。</li> <li>LogAppendTime：broker将消息写入日志的时间。</li> </ul>
批处理消息最大值	Kafka允许的最大批处理大小，如果启用消息压缩，则表示压缩后的最大批处理大小。 如果增加“批处理消息最大值”的值，且存在消费者版本早于0.10.2，此时消费者的“fetch size”值也必须增加，以便消费者可以获取增加后的批处理大小。

## 约束与限制

- 当副本数为1时，不能开启同步复制功能。
- 开启同步复制后，需要在客户端配置acks=all或者-1，否则无效。
- 批量修改Topic配置时，每次最多修改50个Topic的配置。

## 修改单个 Topic 配置

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 通过以下任意一种方法，修改Topic配置。

- 勾选Topic名称左侧的方框，可选一个或多个，单击信息栏左上侧的“编辑Topic”。

- 在待修改配置的Topic所在行，单击“编辑”。

**步骤7** 在“编辑Topic”对话框中，修改Topic的配置，单击“确定”。

设置成功后，在Topic列表页面，查看修改后的配置。

----结束

## 批量修改 Topic 配置

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 勾选待修改配置的Topic，单击信息栏左上侧的“批量编辑Topic”，弹出“批量编辑Topic”对话框。

**步骤7** 在“批量操作”区域，勾选如表6-12所示配置。在“批量数据预览”区域，查看修改前和修改后的配置信息。确认无误后，单击“确定”。

图 6-15 批量修改 Topic 配置

**Batch Edit Topic**

Selected settings will be synced to all selected topics.

**Batch Operations**

Partitions

Aging Time (h)

Synchronous Replication

Synchronous Flushing

Message Timestamp CreateTime ▼

Max. Message Size (bytes) - | 10485759 | +

**Preview Change**

Topic Name	Message Timestamp Before/After	Max. Message Size (bytes) Before/After
topic01	CreateTime	10485759
	LogAppendTime	10485760
topic02	CreateTime	10485759
	LogAppendTime	10485760

Total Records: 2 10 ▼ < 1 >

表 6-12 Topic 配置参数

参数名称	操作说明
同步复制	勾选“同步复制”，并开启或关闭同步复制功能。
同步落盘	勾选“同步落盘”，并开启或关闭同步落盘功能。
消息时间戳类型	勾选“消息时间戳类型”，并在下拉框中选择“CreateTime” / “LogAppendTime”。
批处理消息最大值（字节）	勾选“批处理消息最大值（字节）”，并输入批处理消息最大值。

设置成功后，在Topic列表页面，查看修改后的配置。

----结束

### 6.3.4 修改 Kafka Topic 分区数

创建Kafka实例Topic成功之后，您可以根据业务需要修改Topic的分区数。**修改分区数，不会重启实例，不会影响业务。**

修改Topic分区数的方法如下：

- [在控制台修改单个Topic分区数](#)
- [在Kafka Manager修改Topic分区数](#)
- [在客户端修改Topic分区数](#)

#### 约束与限制

- 分区数只支持增加，不支持减少。
- Kafka实例对Topic的总分区数设置了上限，**当Topic的总分区数达到上限后，用户就无法继续创建Topic**。不同规格配置的Topic总分区数不同，具体请参考[Kafka产品规格说明](#)。
- 已开启密文接入的实例，“allow.everyone.if.no.acl.found”为“false”时，初始用户（即创建实例时设置的用户）可以通过客户端修改Topic分区数，其他用户无法通过客户端修改Topic分区数。
- 批量修改分区数时，每次最多修改50个Topic的分区数。

#### 在控制台修改单个 Topic 分区数

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 通过以下任意一种方法，修改Topic分区数。

- 勾选Topic名称左侧的方框，可选一个或多个，单击信息栏左上侧的“编辑Topic”。
- 在待修改分区数的Topic所在行，单击“编辑”。

**步骤7** 在“编辑Topic”对话框中，参考表6-13修改分区数，然后单击“确定”。

表 6-13 分区数参数说明

参数	说明
分区数	输入分区数。分区数只支持增加，不支持减少。 出于性能考虑，Kafka控制台限定单个Topic的分区数上限为200。
新增分区分配	设置新增分区所在代理。 <ul style="list-style-type: none"> <li>• 自动分配：Kafka服务自动为新增分区分配代理。</li> <li>• 手动分配：在下拉框中为新增分区分配代理，分配的代理数量不能小于Topic的副本数。</li> </ul>

设置成功后，在Topic列表页面，查看修改后的分区数。

----结束

## 在控制台批量修改 Topic 分区数

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > Topic管理”，进入Topic列表页面。

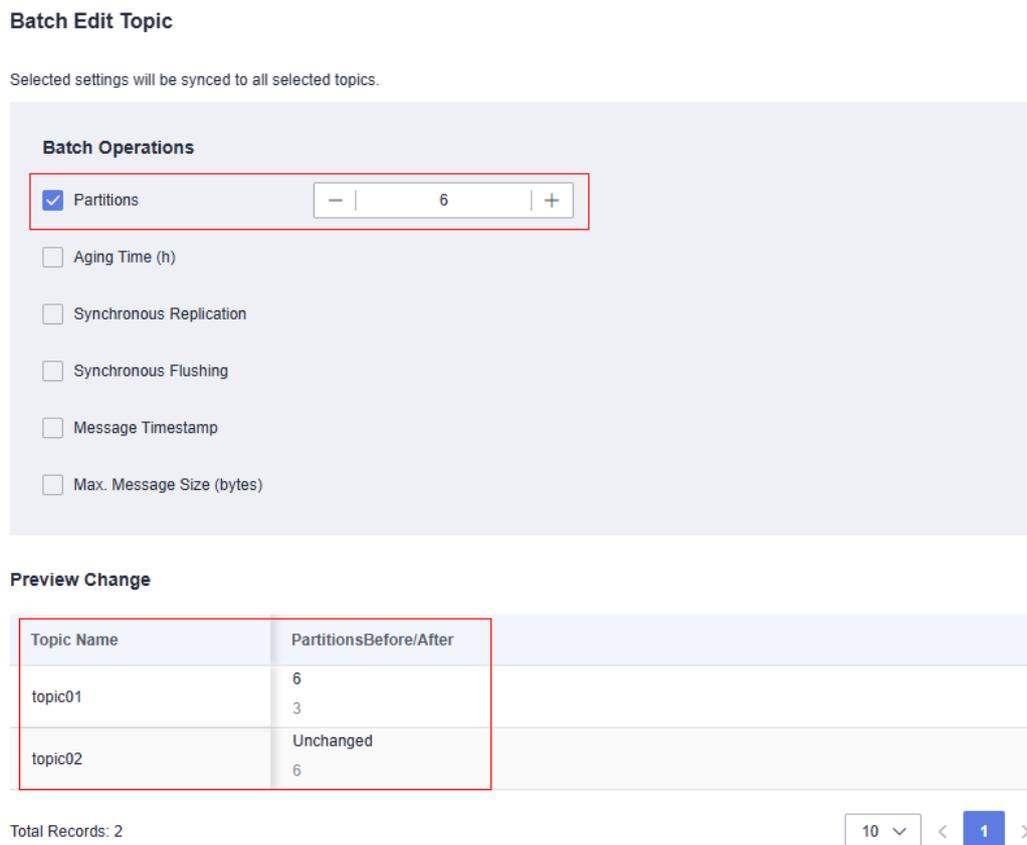
**步骤6** 勾选待修改分区数的Topic，单击信息栏左上侧的“批量编辑Topic”，弹出“批量编辑Topic”对话框。

**步骤7** 在“批量操作”区域，勾选“分区数”，并输入分区数。在“批量数据预览”区域，查看修改前和修改后的分区数。确认无误后，单击“确定”。

分区数的取值范围受以下几个条件约束：

- 出于性能考虑，Kafka控制台限定单个Topic的分区数上限为200。
- 分区数不能小于待设置分区数的Topic中最大分区数的值。例如，Topic01的分区数为3，Topic02的分区数为6，批量修改分区数时，设置的分区数不能小于6。

图 6-16 批量修改分区数



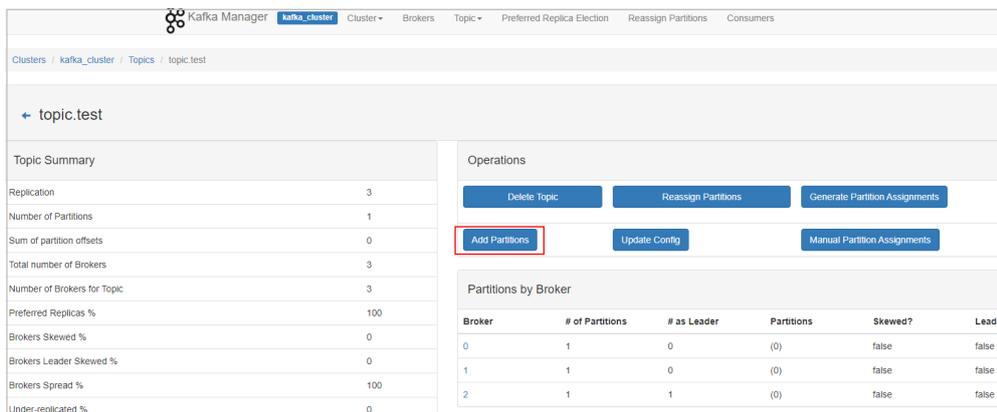
设置成功后，在Topic列表页面，查看修改后的分区数。

----结束

## 在 Kafka Manager 修改 Topic 分区数

- 步骤1** 登录Kafka Manager。
- 步骤2** 在Kafka Manager中，单击“Topic > List”，进入Topic列表界面。
- 步骤3** 单击Topic名称，进入Topic详情界面。
- 步骤4** 单击“Add Partitions”，进入增加分区界面。

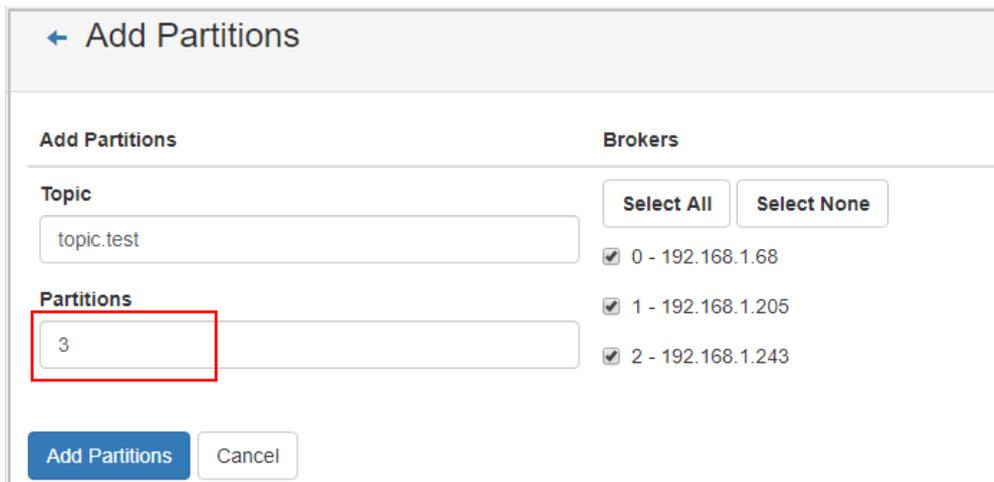
图 6-17 Topic 详情界面



步骤5 输入分区数，单击“Add Partitions”。

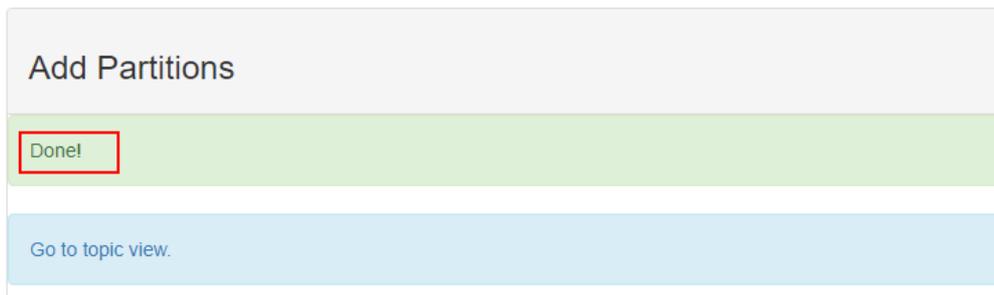
出于性能考虑，建议单个Topic的分区数设置为200以内。

图 6-18 增加分区数量



显示“Done”，表示分区增加成功。

图 6-19 分区增加成功



----结束

## 在客户端修改 Topic 分区数

Kafka客户端版本为2.2以上时，支持通过kafka-topics.sh修改Topic分区数。

- 未开启密文接入的Kafka实例，在Kafka客户端的“/bin”目录下，通过以下命令修改Topic分区数。

```
./kafka-topics.sh --bootstrap-server {connection-address} --topic {topic-name} --alter --partitions {number-of-partitions}
```

表 6-14 修改 Topic 分区数参数说明

参数名称	说明
connection-address	Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。
topic-name	Topic名称。
number-of-partitions	Topic的分区数。 出于性能考虑，建议单个Topic的分区数设置为 <b>200以内</b> 。

示例如下：

```
[root@ecs-kafka bin]# ./kafka-topics.sh --bootstrap-server 192.168.xx.xx:9092,192.168.xx.xx:9092,192.168.xx.xx:9092 --topic topic-01 --alter --partitions 6 [root@ecs-kafka bin]#
```

- 已开启密文接入的Kafka实例，通过以下步骤修改Topic分区数。
  - （可选）如果已经设置了用户名和密码，以及SSL证书配置，请跳过此步骤，执行**b**。否则请执行以下操作。

在Kafka客户端的“/config”目录中创建“ssl-user-config.properties”文件，参考**步骤3**增加用户名和密码，以及SSL证书配置。

- 在Kafka客户端的“/bin”目录下，通过以下命令修改Topic分区数。

```
./kafka-topics.sh --bootstrap-server {connection-address} --topic {topic-name} --alter --partitions {number-of-partitions} --command-config ../config/{ssl-user-config.properties}
```

表 6-15 修改 Topic 分区数参数说明

参数名称	说明
connection-address	Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。
topic-name	Topic名称。
number-of-partitions	Topic的分区数。 出于性能考虑，建议单个Topic的分区数设置为 <b>200以内</b> 。
ssl-user-config.properties	配置文件的名称，此配置文件用于存放用户名和密码，以及SSL证书配置信息。

示例如下：

```
[root@ecs-kafka bin]# ./kafka-topics.sh --bootstrap-server
192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --topic topic-01 --alter --partitions 6 --
command-config ../config/ssl-user-config.properties
[root@ecs-kafka bin]#
```

## 6.3.5 修改 Kafka Topic 副本数

创建Kafka实例Topic成功之后，您可以根据业务需要修改Topic的副本数。

通过修改分区平衡可以实现修改副本数，Kafka控制台提供两种修改副本数的方法：[使用自动平衡修改副本数](#)和[使用手动平衡修改副本数](#)。

### 约束与限制

- 分区平衡任务启动后，无法手动停止任务，需要等到任务完成。
- 已设置了一个定时分区平衡任务，在此任务未执行前，无法对本实例内的任何Topic执行其他分区平衡任务。

### 操作影响

- 对数据量大的Topic进行分区平衡会占用大量的网络和存储带宽，业务可能会出现请求超时或者时延增大，建议在业务低峰期时操作。对Topic进行分区平衡前，根据Kafka实例规格对比当前实例负载情况，评估是否可以进行分区平衡，建议预留足够的带宽进行分区平衡，CPU使用率在90%以上时，不建议进行分区平衡。Topic的数据量和CPU使用率可以通过监控页面的“队列数据容量”和“CPU使用率”查看，具体步骤请参考[查看Kafka监控数据](#)。
- 带宽限制是指设定Topic进行副本同步的带宽上限，确保不会对该实例上的其他Topic造成流量冲击。但需要注意，带宽限制不会区分是正常的生产消息造成的副本同步还是分区平衡造成的副本同步，如果带宽限制设定过小，可能会影响正常的生产消息，且可能会造成分区平衡一直无法结束。如果分区平衡一直无法结束，请联系客服处理。
- 分区平衡任务启动后，不能删除正在进行分区平衡的Topic，否则会导致分区平衡任务无法结束。
- 分区平衡后Topic的metadata会改变，如果生产者不支持重试机制，会有少量的请求失败，导致部分消息生产失败。
- 数据量大的Topic进行分区平衡的时间会比较长。Topic的数据量可以通过监控页面的“队列数据容量”查看，具体步骤请参考[查看Kafka监控数据](#)。在不影响业务的前提下，建议适当[调小Topic老化时间](#)并等待消息老化，减少迁移数据，加快迁移速度。分区平衡任务结束后可重新调整为初始值。

### 前提条件

确保分区平衡的目标Broker磁盘容量充足，在[磁盘存储统计](#)中查看每个Broker的可用磁盘容量。如果目标Broker磁盘剩余容量接近分区平衡迁移到该Broker上的数据量，为了给目标Broker上的消息生产预留存储空间，应先进行[磁盘扩容](#)，再进行分区平衡。

### 使用自动平衡修改副本数

**步骤1** 登录管理控制台。

- 步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。
- 步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。
- 步骤4** 单击Kafka实例的名称，进入实例详情页面。
- 步骤5** 在左侧导航栏单击“实例管理 > Topic管理”，进入Topic列表页面。
- 步骤6** 通过以下任意一种方法，进入“自动平衡”页面。
- 勾选Topic名称左侧的方框，可选一个或多个，单击信息栏左上侧的“一键平衡 > 自动平衡”。
  - 在待修改副本数的Topic所在行，单击“更多 > 一键平衡 > 自动平衡”。
- 步骤7** 修改副本数。

表 6-16 自动平衡参数说明

参数名称	说明
Broker名称	勾选目标Broker的名称，Topic分区的副本即将迁移至目标Broker上。
副本数	输入Topic的副本数，副本数须小于等于Broker的数量。
带宽限制	输入带宽大小，默认值为“-1”，表示不限制带宽。 如果实例负载较低（例如实例规格为300MB/s，当前实例负载为30MB/s），建议不设置带宽限制。如果需要设置带宽限制，建议该参数值不小于待分区平衡Topic的总生产带宽 * 待分区平衡Topic的最大副本数。 带宽限制值的估算方法，请参考 <a href="#">带宽限制值估算方法</a> 。
执行时间	选择分区平衡任务执行的时间。 <ul style="list-style-type: none"> <li>• 立即执行：表示立即执行分区平衡任务。</li> <li>• 定时执行：表示在指定的时间执行分区平衡任务。</li> </ul>

- 步骤8** （可选）单击“一键测算”，在“预估耗时”中，显示执行自动平衡大概需要的时间。
- 一键测算功能对Kafka实例性能没有影响。
- 步骤9** 单击“确定”。

定时任务和非定时任务查看分区平衡是否完成的方法不同，具体如下：

表 6-17 查看分区平衡结果

任务类型	分区平衡结果
非定时任务	在Topic列表页面左上方单击“查看变更任务”，进入“后台任务管理 > 即时任务”页面，当任务的“状态”为“成功”时，表示分区平衡完成，即修改副本数成功。

任务类型	分区平衡结果
定时任务	<ol style="list-style-type: none"> <li>系统自动跳转到“后台任务管理 &gt; 定时任务”页面，此页面的状态仅表示定时任务是否开始执行，并非表示任务是否执行成功。 <ul style="list-style-type: none"> <li>当“状态”为“待执行”时，表示定时分区平衡任务未执行。</li> <li>当“状态”为“成功”时，表示定时分区平衡任务开始执行。</li> <li>当“状态”为“取消”时，表示已取消定时分区平衡任务。</li> </ul> </li> <li>单击“即时任务”，进入“即时任务”页签，查看任务的状态。当任务的“状态”为“成功”时，表示分区平衡完成，即修改副本数成功。</li> </ol>

----结束

## 使用手动平衡修改副本数

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 通过以下任意一种方法，进入“手动平衡”页面。

- 勾选Topic名称左侧的方框，手动平衡每次只能勾选一个Topic，单击信息栏左上侧的“一键平衡 > 手动平衡”。
- 在待修改副本数的Topic所在行，单击“更多 > 一键平衡 > 手动平衡”。

**步骤7** 修改副本数。

- 在“手动平衡”对话框右上角，单击“减少副本”/“添加副本”，为Topic的每个分区减少/增加副本数。
- 在待修改分区平衡的副本名称下，单击Broker名称或 ，选择目标Broker的名称，副本即将迁移至目标Broker上。同一分区下的不同副本需要分配在不同的Broker上。
- 在“带宽限制”中，输入带宽大小，默认值为“-1”，表示不限制带宽。如果实例负载较低（例如实例规格为300MB/s，当前实例负载为30MB/s），建议不设置带宽限制。如果需要设置带宽限制，建议该参数值不小于待分区平衡Topic的总生产带宽 \* 待分区平衡Topic的最大副本数。带宽限制值的估算方法，请参考[带宽限制值估算方法](#)。
- 在“执行时间”中，选择分区平衡任务执行的时间。“立即执行”表示立即执行分区平衡任务。“定时执行”表示在指定的时间执行分区平衡任务。

**步骤8** (可选) 单击“一键测算”，在“预估耗时”中，显示执行手动平衡大概需要的时间。

一键测算功能对Kafka实例性能没有影响。

**步骤9** 单击“确定”。

定时任务和非定时任务查看分区平衡是否完成的方法不同，具体如下：

表 6-18 查看分区平衡结果

任务类型	分区平衡结果
非定时任务	在Topic列表页面左上方单击“查看变更任务”，进入“后台任务管理 > 即时任务”页面，当任务的“状态”为“成功”时，表示分区平衡完成，即修改副本数成功。
定时任务	<ol style="list-style-type: none"> <li>系统自动跳转到“后台任务管理 &gt; 定时任务”页面，此页面的状态仅表示定时任务是否开始执行，并非表示任务是否执行成功。 <ul style="list-style-type: none"> <li>当“状态”为“待执行”时，表示定时分区平衡任务未执行。</li> <li>当“状态”为“成功”时，表示定时分区平衡任务开始执行。</li> <li>当“状态”为“取消”时，表示已取消定时分区平衡任务。</li> </ul> </li> <li>单击“即时任务”，进入“即时任务”页签，查看任务的状态。当任务的“状态”为“成功”时，表示分区平衡完成，即修改副本数成功。</li> </ol>

----结束

## 6.3.6 导出 Kafka Topic 列表

本章节指导您在控制台导出Topic列表，支持批量导出。

### 前提条件

已创建Topic。

### 导出 Topic 列表

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 通过以下任意一种方法，导出Topic列表。

- 勾选待导出的Topic，单击“导出 > 导出已选中数据到XLSX”，导出指定Topic列表。
- 单击“导出 > 导出全部数据到XLSX”，导出全部Topic列表。

Topic列表中包含如下信息：Topic名称、分区数、副本数、老化时间、是否同步复制或同步落盘、消息时间戳类型、批处理消息最大值。

----结束

## 6.3.7 修改 Kafka 分区平衡

分区平衡是指将分区的副本重新分配到不同的代理上，解决代理负载不均衡问题。

需要进行分区平衡的场景如下：

- 实例扩容代理个数后，新创建的Topic才会分布在新代理上，原有Topic还分布在原有代理上，造成分区分布不均衡。通过修改分区平衡，将原有Topic分区的副本迁移到新代理上。
- 将高负载代理上的Leader分区切换为Follower。
- 在修改分区平衡过程中，可以配置增加/减少Topic的副本数。

分布式消息服务Kafka版控制台提供两种分区平衡的方法：[自动平衡](#)和[手动平衡](#)，建议选择自动平衡，确保分区Leader的均匀分布。

### 约束与限制

- 分区平衡任务启动后，无法修改Topic的分区数。
- 分区平衡任务启动后，无法手动停止任务，需要等到任务完成。
- 已设置了一个定时分区平衡任务，在此任务未执行前，无法对本实例内的任何Topic执行其他分区平衡任务。

### 操作影响

- 对数据量大的Topic进行分区平衡会占用大量的网络和存储带宽，业务可能会出现请求超时或者时延增大，建议在业务低峰期时操作。对Topic进行分区平衡前，根据Kafka实例规格对比当前实例负载情况，评估是否可以分区平衡，建议预留足够的带宽进行分区平衡，CPU使用率在90%以上时，不建议进行分区平衡。Topic的数据量和CPU使用率可以通过监控页面的“队列数据容量”和“CPU使用率”查看，具体步骤请参考[查看Kafka监控数据](#)。
- 带宽限制是指设定Topic进行副本同步的带宽上限，确保不会对该实例上的其他Topic造成流量冲击。但需要注意，带宽限制不会区分是正常的生产消息造成的副本同步还是分区平衡造成的副本同步，如果带宽限制设定过小，可能会影响正常的生产消息，且可能会造成分区平衡一直无法结束。如果分区平衡一直无法结束，请联系客服处理。
- 分区平衡任务启动后，不能删除正在进行分区平衡的Topic，否则会导致分区平衡任务无法结束。
- 分区平衡后Topic的metadata会改变，如果生产者不支持重试机制，会有少量的请求失败，导致部分消息生产失败。
- 数据量大的Topic进行分区平衡的时间会比较长。Topic的数据量可以通过监控页面的“队列数据容量”查看，具体步骤请参考[查看Kafka监控数据](#)。在不影响业务的

前提下，建议适当调小Topic老化时间并等待消息老化，减少迁移数据，加快迁移速度。分区平衡任务结束后可重新调整为初始值。

## 前提条件

确保分区平衡的目标Broker磁盘容量充足，在[磁盘存储统计](#)中查看每个Broker的可用磁盘容量。如果目标Broker磁盘剩余容量接近分区平衡迁移到该Broker上的数据量，为了给目标Broker上的消息生产预留存储空间，应先进行[磁盘扩容](#)，再进行分区平衡。

## 自动平衡

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 通过以下任意一种方法，修改分区平衡。

- 勾选Topic名称左侧的方框，可选一个或多个，单击信息栏左上侧的“一键平衡 > 自动平衡”。
- 在待修改分区平衡的Topic所在行，单击“更多 > 一键平衡 > 自动平衡”。

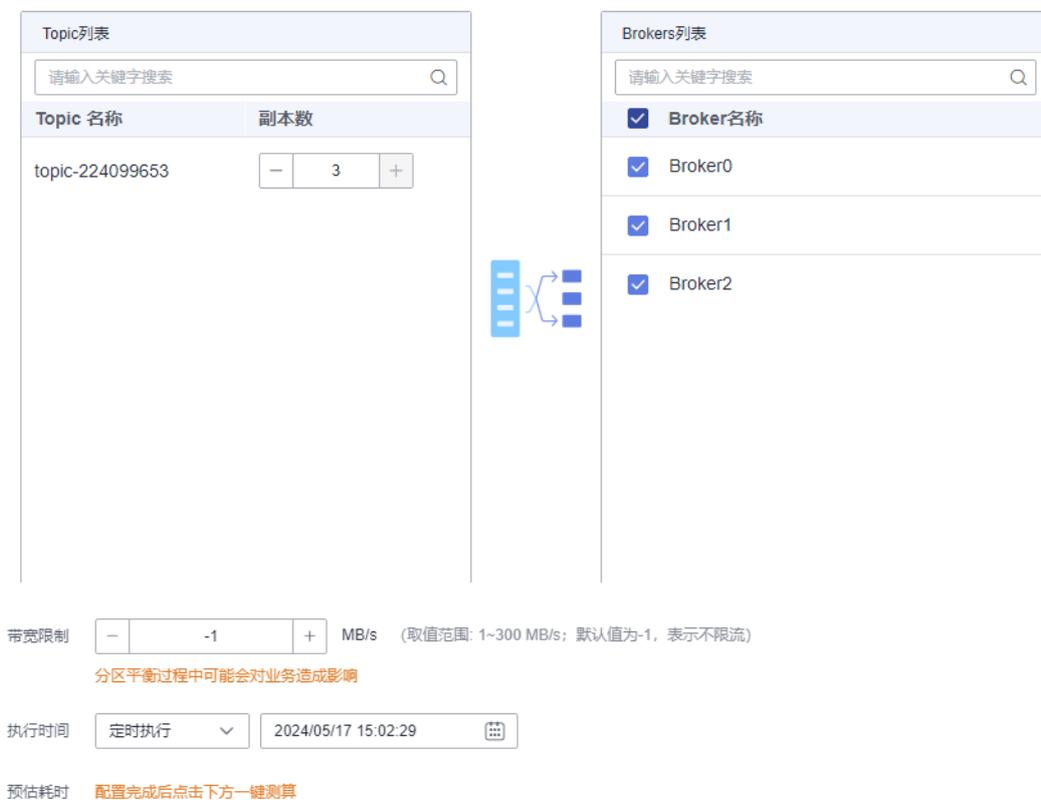
**步骤7** 设置自动平衡参数。

表 6-19 自动平衡参数说明

参数名称	说明
Broker名称	勾选目标Broker的名称，Topic分区的副本即将迁移至目标Broker上。
副本数	输入Topic需要自动平衡的副本数，副本数须小于等于Broker的数量。
带宽限制	输入带宽大小，默认值为“-1”，表示不限制带宽。 如果实例负载较低（例如实例规格为300MB/s，当前实例负载为30MB/s），建议不设置带宽限制。如果需要设置带宽限制，建议该参数值不小于待分区平衡Topic的总生产带宽 * 待分区平衡Topic的最大副本数。 带宽限制值的估算方法，请参考 <a href="#">带宽限制值估算方法</a> 。
执行时间	选择分区平衡任务执行的时间。 <ul style="list-style-type: none"> <li>• 立即执行：表示立即执行分区平衡任务。</li> <li>• 定时执行：表示在指定的时间执行分区平衡任务。</li> </ul>

图 6-20 设置自动平衡参数

自动平衡



**步骤8** (可选) 单击“一键测算”，在“预估耗时”中，显示执行自动平衡大概需要的时间。

一键测算功能对Kafka实例性能没有影响。

**步骤9** 单击“确定”。

定时任务和非定时任务查看分区平衡是否完成的方法不同，具体如下：

表 6-20 查看分区平衡结果

任务类型	分区平衡结果
非定时任务	在Topic列表页面左上方单击“查看变更任务”，进入“后台任务管理 > 即时任务”页面，当任务的“状态”为“成功”时，表示分区平衡完成。

任务类型	分区平衡结果
定时任务	<ol style="list-style-type: none"> <li>系统自动跳转到“后台任务管理 &gt; 定时任务”页面，此页面的状态仅表示定时任务是否开始执行，并非表示任务是否执行成功。 <ul style="list-style-type: none"> <li>当“状态”为“待执行”时，表示定时分区平衡任务未执行。</li> <li>当“状态”为“成功”时，表示定时分区平衡任务开始执行。</li> <li>当“状态”为“取消”时，表示已取消定时分区平衡任务。</li> </ul> </li> <li>单击“即时任务”，进入“即时任务”页签，查看任务的状态。当任务的“状态”为“成功”时，表示分区平衡完成。</li> </ol>

### 📖 说明

- 分区平衡任务启动后，不能删除正在进行分区平衡的Topic，否则会导致分区平衡任务无法结束。
- 分区平衡任务启动后，无法修改Topic的分区数。
- 分区平衡任务启动后，无法手动停止任务，需要等到任务完成。
- 已设置了一个定时分区平衡任务，在此任务未执行前，无法对本实例内的任何Topic执行其他分区平衡任务。

---结束

## 手动平衡

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 通过以下任意一种方法，修改分区平衡。

- 勾选Topic名称左侧的方框，手动平衡每次只能勾选一个Topic，单击信息栏左上侧的“一键平衡 > 手动平衡”。
- 在待修改分区平衡的Topic所在行，单击“更多 > 一键平衡 > 手动平衡”。

**步骤7** 设置手动平衡参数。

- 在“手动平衡”对话框右上角，单击“减少副本”/“添加副本”，为Topic的每个分区减少/增加副本数。
- 在待修改分区平衡的副本名称下，单击Broker名称或 ，选择目标Broker的名称，副本即将迁移至目标Broker上。同一分区下的不同副本需要分配在不同的Broker上。

- 在“带宽限制”中，输入带宽大小，默认值为“-1”，表示不限制带宽。如果实例负载较低（例如实例规格为300MB/s，当前实例负载为30MB/s），建议不设置带宽限制。如果需要设置带宽限制，建议该参数值不小于待分区平衡Topic的总生产带宽 \* 待分区平衡Topic的最大副本数。带宽限制值的估算方法，请参考[带宽限制值估算方法](#)。
- 在“执行时间”中，选择分区平衡任务执行的时间。“立即执行”表示立即执行分区平衡任务。“定时执行”表示在指定的时间执行分区平衡任务。

图 6-21 设置手动平衡参数

手动平衡

i 每个分区下的副本1为leader副本

分区 0

副本 1  
Broker 2

副本 2  
Broker 1

副本 3  
Broker 0

分区 1

副本 1  
Broker 0

副本 2  
Broker 2

副本 3  
Broker 1

带宽限制  MB/s (取值范围: 1~300 MB/s; 默认值为-1, 表示不限流)

分区平衡过程中可能会对业务造成影响

执行时间 定时执行 2024/05/17 15:05:17 ✕

预估耗时 配置完成后点击下方一键测算

**步骤8** （可选）单击“一键测算”，在“预估耗时”中，显示执行手动平衡大概需要的时间。

一键测算功能对Kafka实例性能没有影响。

**步骤9** 单击“确定”。

定时任务和非定时任务查看分区平衡是否完成的方法不同，具体如下：

表 6-21 查看分区平衡结果

任务类型	分区平衡结果
非定时任务	在Topic列表页面左上方单击“查看变更任务”，进入“后台任务管理 > 即时任务”页面，当任务的“状态”为“成功”时，表示分区平衡完成。

任务类型	分区平衡结果
定时任务	<ol style="list-style-type: none"> <li>系统自动跳转到“后台任务管理 &gt; 定时任务”页面，此页面的状态仅表示定时任务是否开始执行，并非表示任务是否执行成功。 <ul style="list-style-type: none"> <li>当“状态”为“待执行”时，表示定时分区平衡任务未执行。</li> <li>当“状态”为“成功”时，表示定时分区平衡任务开始执行。</li> <li>当“状态”为“取消”时，表示已取消定时分区平衡任务。</li> </ul> </li> <li>单击“即时任务”，进入“即时任务”页签，查看任务的状态。当任务的“状态”为“成功”时，表示分区平衡完成。</li> </ol>

### 说明

- 分区平衡任务启动后，不能删除正在进行分区平衡的Topic，否则会导致分区平衡任务无法结束。
- 分区平衡任务启动后，无法修改Topic的分区数。
- 分区平衡任务启动后，无法手动停止任务，需要等到任务完成。
- 已设置了一个定时分区平衡任务，在此任务未执行前，无法对本实例内的任何Topic执行其他分区平衡任务。

----结束

## 修改定时分区平衡任务

**步骤1** 在“实例管理 > 后台任务管理”页面的“定时任务”页签中，单击页面左上角下拉框，选择时间段，在搜索对话框中输入待修改定时分区平衡任务的Topic名称，按“Enter”，实现快速查找定时分区平衡任务。

图 6-22 查找定时分区平衡任务



**步骤2** 在待修改的定时分区平衡任务后，单击“修改”。

**步骤3** 在弹出的“修改定时任务”对话框中，您可以修改定时分区平衡任务的时间，还可以取消定时分区平衡任务，具体操作如下。

- 修改定时分区平衡任务的时间：修改时间，单击“确定”。
- 取消定时分区平衡任务：选择“取消”，如图6-23所示，单击“确定”。

图 6-23 取消定时分区平衡任务



----结束

## 带宽限制值估算方法

带宽限制值受分区平衡任务执行时间、分区副本Leader/Follower分布情况以及消息生产速率等因素影响，具体分析如下。

- 带宽限制值作用范围为整个Broker，对该Broker内所有副本同步的分区进行带宽限流。
- 带宽限制会将分区平衡后新增的副本视为Follower副本限流，分区平衡前的Leader副本视为Leader副本限流，Leader副本的限流和Follower副本限流分开计算。
- 带宽限制不会区分是正常的生产消息造成的副本同步还是分区平衡造成的副本同步，因此两者的流量都会被限流统计。

假设分区平衡任务需要在200s内完成，每个副本的数据量为100MB，在以下几种场景中，估算带宽限制值。

**场景一：** Topic1有2分区2副本，Topic2有1分区1副本，所有Leader副本在同一个Broker上，如表6-22所示，Topic1和Topic2分别需要新增1个副本，如表6-23所示。

表 6-22 分区平衡前 Topic 分区的副本分布

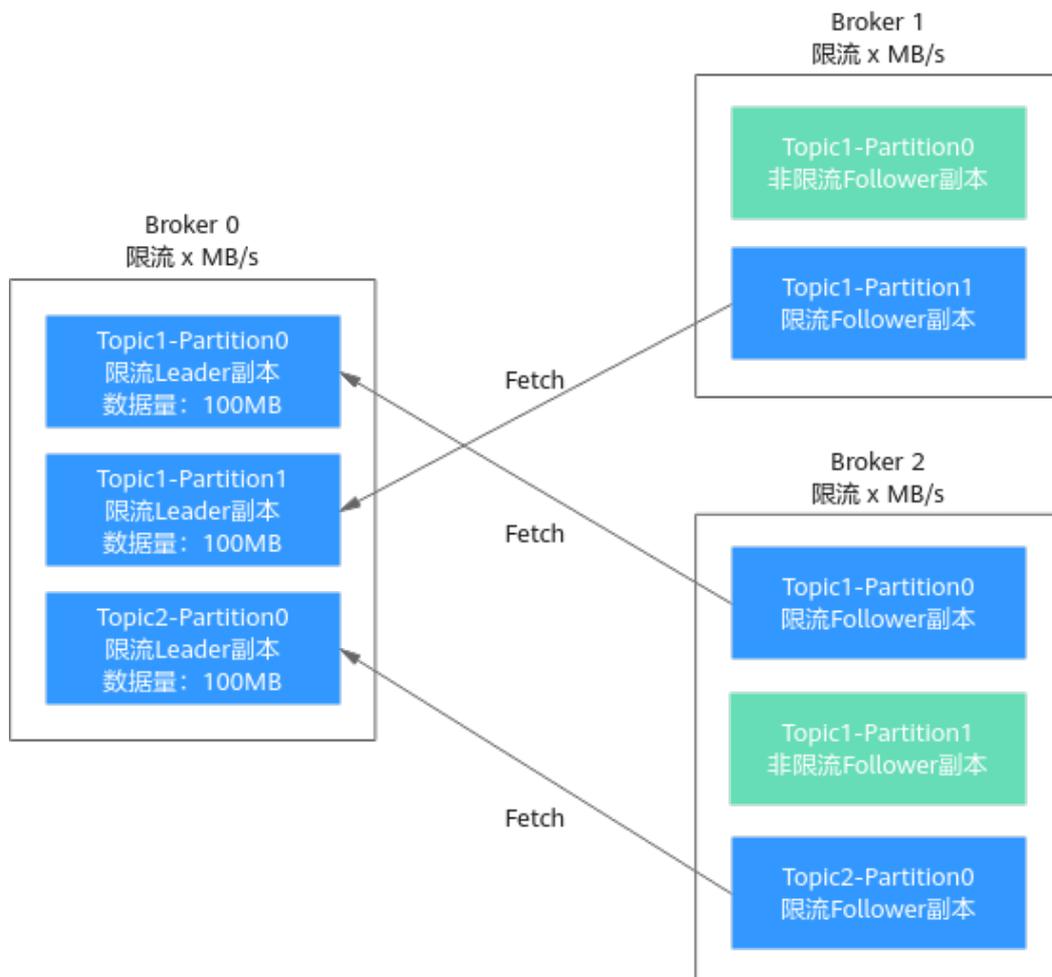
Topic名称	分区名称	Leader副本所在Broker	Follower副本所在Broker
Topic1	0	0	0, 1
Topic1	1	0	0, 2
Topic2	0	0	0

表 6-23 分区平衡后 Topic 分区的副本分布

Topic名称	分区名称	Leader副本所在Broker	Follower副本所在Broker
Topic1	0	0	0, 1, 2
Topic1	1	0	0, 1, 2

Topic名称	分区名称	Leader副本所在Broker	Follower副本所在Broker
Topic2	0	0	0, 2

图 6-24 场景一分区平衡图



如图6-24所示，有3个副本需要从Broker 0拉取数据，Broker 0中每个副本的数据量为100MB，Broker 0中只有Leader副本，Broker 1和Broker 2中只有Follower副本，由此得出以下数据：

- Broker 0在200s内完成分区平衡所需的带宽限制值=  $(100+100+100) / 200=1.5\text{MB/s}$
- Broker 1在200s内完成分区平衡所需的带宽限制值=  $100/200=0.5\text{MB/s}$
- Broker 2在200s内完成分区平衡所需的带宽限制值=  $(100+100) / 200=1\text{MB/s}$

综上所述，若想要在200s内完成分区平衡任务，带宽限制值应设为大于等于1.5MB/s。由于控制台的带宽限制只能是整数，因此带宽限制值应设为大于等于2MB/s。

**场景二：Topic1有2分区1副本，Topic2有2分区1副本，Leader副本分布在不同Broker上，如表6-24所示，Topic1和Topic2分别需要新增1个副本，如表6-25所示。**

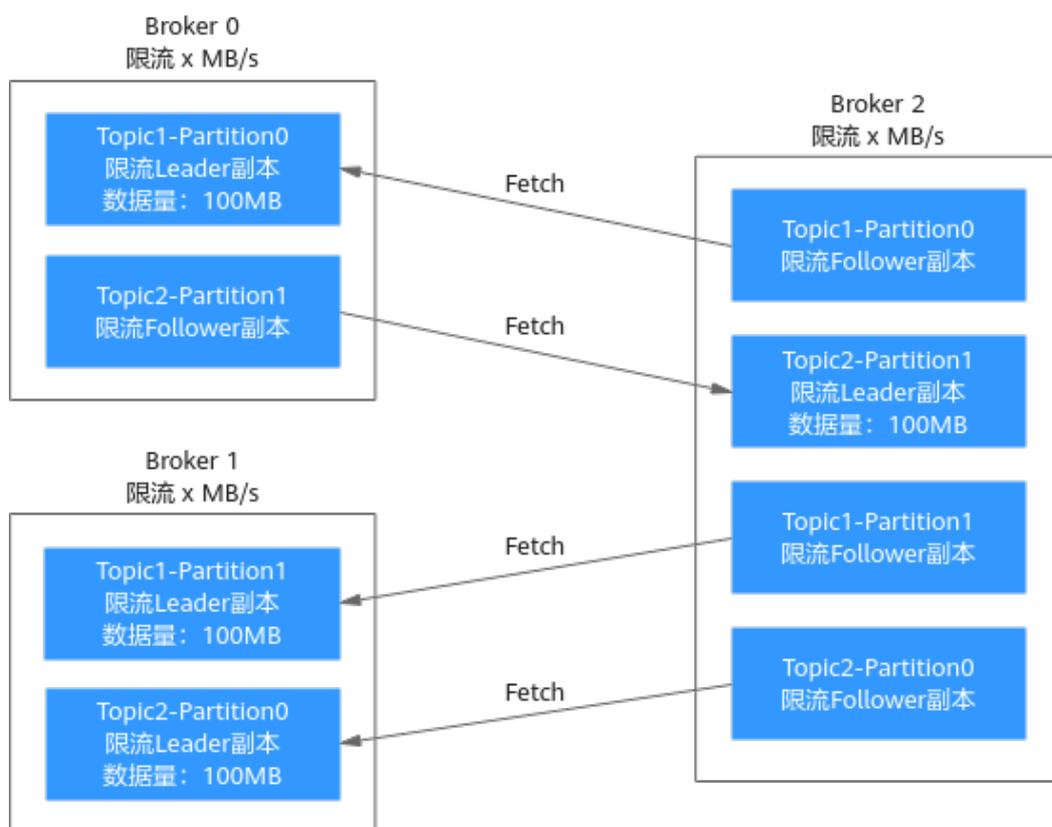
表 6-24 分区平衡前 Topic 分区的副本分布

Topic名称	分区名称	Leader副本所在Broker	Follower副本所在Broker
Topic1	0	0	0
Topic1	1	1	1
Topic2	0	1	1
Topic2	1	2	2

表 6-25 分区平衡后 Topic 分区的副本分布

Topic名称	分区名称	Leader副本所在Broker	Follower副本所在Broker
Topic1	0	0	0, 2
Topic1	1	1	1, 2
Topic2	0	1	1, 2
Topic2	1	2	2, 0

图 6-25 场景二分区平衡图



如图6-25所示，Broker 1中只有Leader副本，Broker 0和Broker 2中存在Leader副本和Follower副本，由于Leader副本的限流和Follower副本限流分开计算，Broker 0和Broker 2需要分别计算Leader副本的限流和Follower副本限流。由此得出以下数据：

- Broker 0作为Leader副本在200s内完成分区平衡所需的带宽限制值  
=100/200=0.5MB/s
- Broker 0作为Follower副本在200s内完成分区平衡所需的带宽限制值  
=100/200=0.5MB/s
- Broker 1在200s内完成分区平衡所需的带宽限制值= ( 100+100 ) /200=1MB/s
- Broker 2作为Leader副本在200s内完成分区平衡所需的带宽限制值  
=100/200=0.5MB/s
- Broker 2作为Follower副本在200s内完成分区平衡所需的带宽限制值=  
( 100+100+100 ) /200=1.5MB/s

综上所述，若想要在200s内完成分区平衡任务，带宽限制值应设为大于等于1.5MB/s。由于控制台的带宽限制只能是整数，因此带宽限制值应设为大于等于2MB/s。

**场景三：Topic1有1分区2副本，Topic2有1分区2副本，所有Leader副本在同一个Broker上，如表6-26所示，Topic1需要新增1个副本，Topic1上有生产消息造成的副本同步，如表6-27所示。**

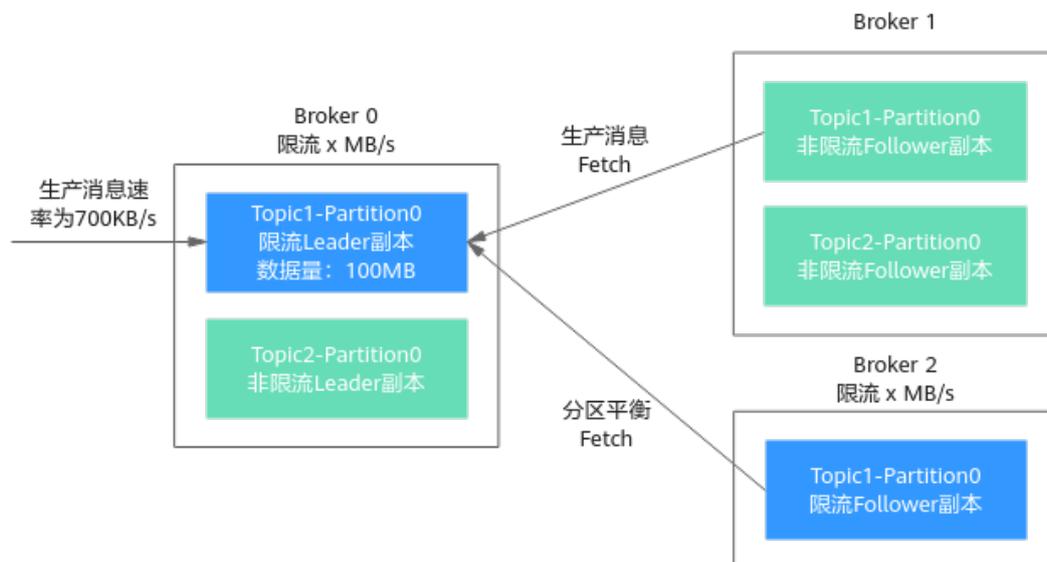
表 6-26 分区平衡前 Topic 分区的副本分布

Topic名称	分区名称	Leader副本所在Broker	Follower副本所在Broker
Topic1	0	0	0, 1
Topic2	0	0	0, 1

表 6-27 分区平衡后 Topic 分区的副本分布

Topic名称	分区名称	Leader副本所在Broker	Follower副本所在Broker
Topic1	0	0	0, 1, 2
Topic2	0	0	0, 1

图 6-26 场景三分区平衡图



如图6-26所示，有1个副本由于分区平衡需要从Broker 0拉取数据，另1个副本由于生产消息需要从Broker 0拉取数据，带宽限制不会区分是正常的生产消息造成的副本同步还是分区平衡造成的副本同步，因此两者的流量都会被限流统计。由此得出以下数据：

- Broker 0在200s内完成分区平衡所需的带宽限制值= ( 100MB+700KB/s\*200s ) / 200s+700KB/s=1.9MB/s
- Broker 2在200s内完成分区平衡所需的带宽限制值=100/200=0.5MB/s

综上所述，若想要在200s内完成分区平衡任务，带宽限制值应设为大于等于1.9MB/s。由于控制台的带宽限制只能是整数，因此带宽限制值应设为大于等于2MB/s。

### 6.3.8 配置 Kafka 自动创建 Topic

“Kafka自动创建Topic”表示向一个未创建的Topic生产或消费消息时，系统会自动创建此Topic，此Topic的默认参数值如表6-28所示。

如果在“配置参数”中修改“log.retention.hours”（老化时间）、“default.replication.factor”（副本数）或“num.partitions”（分区数）的参数值，此后自动创建的Topic参数值为修改后的参数值。例如：“num.partitions”修改为“5”，自动创建的Topic参数值如表6-28所示。

表 6-28 Topic 参数值

参数名称	默认值	修改后的值
分区数	3	5
副本数	3	3
老化时间	72小时	72小时
同步复制	不开启	不开启

参数名称	默认值	修改后的值
同步落盘	不开启	不开启
消息时间戳类型	CreateTime	CreateTime
批处理消息最大值	10485760字节	10485760字节

## 约束与限制

开启/关闭“Kafka自动创建Topic”可能会导致实例重启，具体以控制台显示为准。

## 配置 Kafka 自动创建 Topic

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例名称，进入“概览”页面。

**步骤5** 在“实例信息”区域的“Kafka自动创建Topic”后，单击  或 ，弹出“确认”对话框。

**步骤6** 单击“确定”，跳转到“后台任务管理”页面。任务状态为“成功”时，表示开启/关闭Kafka自动创建Topic成功。

----结束

## 6.3.9 删除 Kafka Topic

本章节指导您删除Topic，请根据实际情况选择以下任意一种方式：

- [删除Kafka Topic（控制台）](#)
- [删除Kafka Topic（Kafka客户端）](#)

## 约束与限制

- 删除Topic会导致Topic中的数据也被删除，并且无法找回，同时会影响使用该Topic的相关业务。
- 已开启密文接入的实例，“allow.everyone.if.no.acl.found”设置为“false”时，初始用户（即创建实例时设置的用户）可以通过客户端删除Topic，其他用户无法通过客户端删除Topic。

## 前提条件

只有运行中的实例才可以删除Topic。

## 删除 Kafka Topic (控制台)

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 通过以下任意一种方法，删除Topic。

- 勾选Topic名称左侧的方框，可选一个或多个，单击信息栏左上侧的“删除Topic”。
- 在待删除Topic所在行，单击“更多 > 删除”。
- 单击待删除的Topic名称，进入Topic详情页。单击右上角的“删除”。

**步骤7** 在“删除Topic”对话框中，单击“确定”。

已删除的Topic不再显示在Topic列表中，表明Topic已成功删除。

----结束

## 删除 Kafka Topic (Kafka 客户端)

Kafka客户端版本为2.2以上时，支持通过**kafka-topics.sh**删除Topic。

- 未开启密文接入的Kafka实例，在Kafka客户端的“/bin”目录下，通过以下命令删除Topic。

```
./kafka-topics.sh --bootstrap-server {connection-address} --delete --topic {topic-name}
```

表 6-29 删除 Topic 参数说明

参数名称	说明
connection-address	Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。
topic-name	Topic名称。

示例如下：

```
[root@ecs-kafka bin]# ./kafka-topics.sh --bootstrap-server
192.168.xx.xx:9092,192.168.xx.xx:9092,192.168.xx.xx:9092 --delete --topic topic-01
[root@ecs-kafka bin]#
```

- 已开启密文接入的Kafka实例，通过以下步骤删除Topic。
  - (可选) 如果已经设置了SSL证书配置，请跳过此步骤，执行**b**。否则请执行以下操作。  
在Kafka客户端的“/config”目录中创建“ssl-user-config.properties”文件，参考**步骤3**增加SSL证书配置。

- b. 在Kafka客户端的“/bin”目录下，通过以下命令删除Topic。
- ```
./kafka-topics.sh --bootstrap-server {connection-address} --delete --topic {topic-name} --command-config ../config/{ssl-user-config.properties}
```

**表 6-30** 删除 Topic 参数说明

| 参数名称                       | 说明                                     |
|----------------------------|----------------------------------------|
| connection-address         | Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。 |
| topic-name                 | Topic名称。                               |
| ssl-user-config.properties | 配置文件的名称，此配置文件用于存放用户名和密码，以及SSL证书配置信息。   |

示例如下：

```
[root@ecs-kafka bin]# ./kafka-topics.sh --bootstrap-server
192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --delete --topic topic-01 --command-
config ../config/ssl-user-config.properties
[root@ecs-kafka bin]#
```

# 7 连接实例

## 7.1 配置 Kafka 网络连接

### 7.1.1 连接 Kafka 网络要求

客户端通过内网连接Kafka实例时，注意以下几点：

- 如果客户端和Kafka实例部署在同一个VPC内，网络默认互通。
- 如果客户端和Kafka实例部署在不同VPC中，由于VPC之间逻辑隔离，客户端和Kafka实例不能直接通信，需要打通VPC之间的网络。

客户端连接Kafka实例的方式如[表7-1](#)所示。

表 7-1 连接方式说明

| 连接方式 | 实现方式                                                                 | 参考文档                                  |
|------|----------------------------------------------------------------------|---------------------------------------|
| 内网连接 | 客户端和Kafka实例部署在同一个VPC中，此时网络默认互通。                                      | -                                     |
|      | 客户端和Kafka实例部署在同一个Region的不同VPC中，利用VPC终端节点在不同VPC间建立跨VPC的连接通道，实现跨VPC访问。 | <a href="#">使用VPCPEP实现跨VPC访问Kafka</a> |
|      | 客户端和Kafka实例部署在同一个Region的不同VPC中，利用VPC对等连接将两个VPC间的网络打通，实现跨VPC访问。       | 《虚拟私有云 用户指南》的“VPC对等连接”章节              |

客户端连接Kafka实例前，需要在Kafka实例所属安全组中配置如下安全组规则，否则会连接失败。

#### 📖 说明

创建安全组后，系统默认添加入方向“允许安全组内的弹性云服务器彼此通信”规则和出方向“放通全部流量”规则，此时使用内网通过同一个VPC访问Kafka实例，无需添加[表7-2](#)的规则。

表 7-2 安全组规则

| 方向  | 协议  | 类型   | 端口   | 源地址                 | 说明                                                                                                                 |
|-----|-----|------|------|---------------------|--------------------------------------------------------------------------------------------------------------------|
| 入方向 | TCP | IPv4 | 9092 | Kafka客户端所在的IP地址或地址组 | <ul style="list-style-type: none"> <li>使用内网通过同一个VPC访问Kafka实例（明文接入）。</li> <li>使用对等连接跨VPC访问Kafka实例（明文接入）。</li> </ul> |
| 入方向 | TCP | IPv6 | 9192 | Kafka客户端所在的IP地址或地址组 | 通过IPv6地址访问Kafka实例（关闭SSL加密）。                                                                                        |
| 入方向 | TCP | IPv4 | 9093 | Kafka客户端所在的IP地址或地址组 | <ul style="list-style-type: none"> <li>使用内网通过同一个VPC访问Kafka实例（密文接入）。</li> <li>使用对等连接跨VPC访问Kafka实例（密文接入）。</li> </ul> |
| 入方向 | TCP | IPv6 | 9193 | Kafka客户端所在的IP地址或地址组 | 通过IPv6地址访问Kafka实例（开启SSL加密）。                                                                                        |
| 入方向 | TCP | IPv4 | 9011 | 198.19.128.0/17     | 通过VPC终端节点实现跨VPC访问Kafka实例（密文接入和明文接入都适用）。                                                                            |

## 7.1.2 使用 VPCEP 实现跨 VPC 访问 Kafka

Kafka客户端和Kafka实例在同一区域的不同VPC中，由于VPC之间逻辑隔离，Kafka客户端和Kafka实例不能直接通信。您可以通过以下任意一个方式实现跨VPC访问：

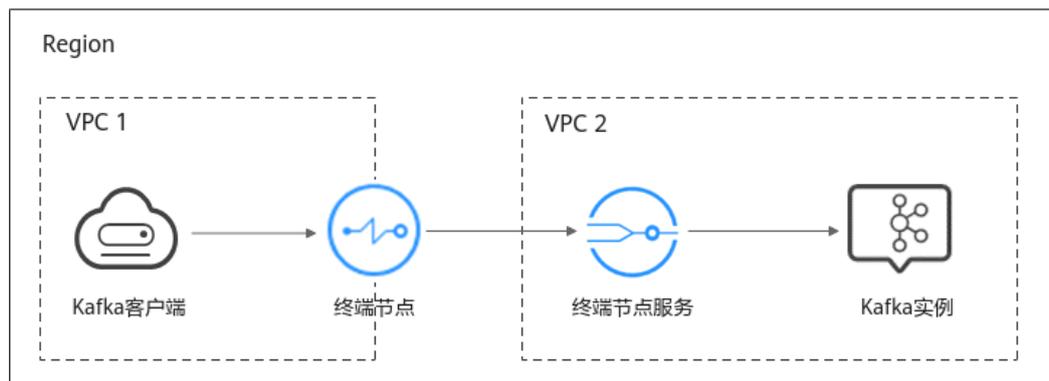
- 创建VPC对等连接，将两个VPC的网络打通，实现跨VPC访问。具体步骤请参考[对等连接](#)。
- 利用VPC终端节点在不同VPC间建立跨VPC的连接通道，实现Kafka客户端通过内网访问Kafka实例。

本章节主要介绍通过VPC终端节点实现跨VPC访问的方法。

VPC终端节点由“终端节点服务”和“终端节点”两种资源实例组成。

- 终端节点服务：将Kafka实例配置为VPC终端节点支持的服务，可以被终端节点连接和访问。
- 终端节点：用于在VPC和终端节点服务之间建立便捷、安全、私密的连接通道。

图 7-1 跨 VPC 访问 Kafka 实例



## 客户端使用 VPCEP 跨 VPC 访问 Kafka 时，使用的是明文连接还是密文连接？

使用明文连接还是密文连接，取决于“跨VPC访问协议”。跨VPC访问协议是在创建Kafka实例时设置的，实例创建成功后，不支持修改。

跨VPC访问协议的取值如下：

- PLAINTEXT：表示客户端访问Kafka实例时，无需认证，数据通过明文传输。
- SASL\_SSL：表示客户端访问Kafka实例时，使用SASL认证，数据通过SSL证书加密传输。
- SASL\_PLAINTEXT：表示客户端访问Kafka实例时，使用SASL认证，数据通过明文传输。

## 步骤一：创建终端节点服务

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例名称，进入实例详情页面。

**步骤5** 在“概览”页面的“高级配置”区域，获取Kafka实例的网卡地址（listeners IP）和对应的Port ID。

图 7-2 Kafka 实例的网卡地址（listeners IP）和对应的 Port ID

Cross-VPC Access Modify

| listeners IP Address | advertised.listeners IP Address/Domain Name | Port | Port ID                              |
|----------------------|---------------------------------------------|------|--------------------------------------|
| 192.168.0.194        | 192.168.0.194                               | 9011 | c7af42a5-3586-429d-a173-c76e9cf25e29 |
| 192.168.0.32         | 192.168.0.32                                | 9011 | d69f3c5f-0d78-41ee-ba1a-3df64c4f8d3e |
| 192.168.0.245        | 192.168.0.245                               | 9011 | 8d5f77e4-e9bc-467a-8309-d513d8fc6666 |

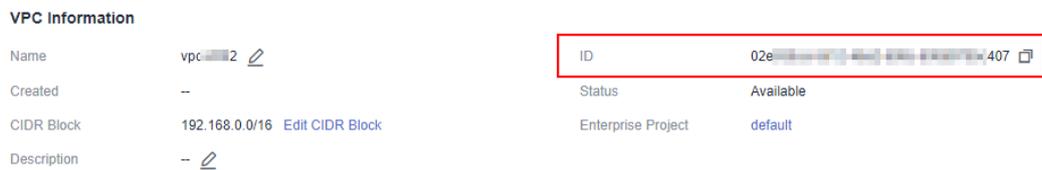
**步骤6** 在“概览”页面的“网络”区域，查看Kafka实例所在的VPC名称。

图 7-3 查看 Kafka 实例所在的 VPC 名称



**步骤7** 单击VPC名称，进入VPC基本信息页签，获取VPC ID。

图 7-4 获取 VPC ID



**步骤8** 调用VPC终端节点的API创建终端节点服务。参考表7-3设置请求参数，其他参数根据实际情况设置。API详情请参考[创建终端节点服务](#)。

POST https://{endpoint}/v1/{project\_id}/vpc-endpoint-services

表 7-3 创建终端节点服务参数说明

| 参数名称        | 说明                                         |
|-------------|--------------------------------------------|
| port_id     | 终端节点服务后端资源的ID。<br>输入步骤5中获取的其中一个Port ID。    |
| vpc_id      | 终端节点服务对应后端资源所在的VPC的ID。<br>输入步骤7中获取的VPC ID。 |
| server_type | 资源类型。<br>输入“VM”。                           |
| client_port | 终端节点访问的端口。<br>输入“9011”。                    |
| server_port | 终端节点服务的端口。<br>输入“9011”。                    |
| protocol    | 端口映射协议。<br>输入“TCP”。                        |

| 参数名称             | 说明                                                   |
|------------------|------------------------------------------------------|
| approval_enabled | 是否需要审批。<br>输入“false”，表示不需要审批，创建的终端节点连接直接为accepted状态。 |
| service_type     | 终端节点服务类型。<br>输入“interface”。                          |
| endpoint         | 获取VPC终端节点的终端节点，区域必须与Kafka实例保持一致。                     |
| project_id       | 获取VPC终端节点的项目ID，区域必须与Kafka实例保持一致。                     |

记录响应信息中的“service\_name”参数值，此参数表示终端节点服务的名称。

**步骤9** 参考**步骤8**，为**步骤5**中其他Port ID创建终端节点服务，并记录终端节点服务的名称。

----结束

## （可选）步骤二：添加白名单

VPCEP允许跨账号的终端节点连接终端节点服务，通过设置终端节点服务的白名单实现。

Kafka客户端和Kafka实例属于不同账号时，将Kafka客户端所在账号的ID添加到终端节点服务的白名单中，完成跨账号终端节点的访问授权，具体操作步骤请参考[添加白名单](#)。

## 步骤三：创建终端节点

**步骤1** 在管理控制台左上角单击，选择“网络 > VPC终端节点”，进入终端节点页面。

**步骤2** 单击“Create VPC Endpoint”，进入“Create VPC Endpoint”页面。

**步骤3** 参考**表7-4**设置参数，其他参数保持默认，如果想要了解更多的参数信息，请参考[创建终端节点](#)。

图 7-5 终端节点参数设置

The screenshot shows a configuration form for terminal nodes. Key elements include:

- Region:** A dropdown menu with a warning icon.
- Service Category:** Two buttons: "Cloud service" and "Find a service by name".
- VPC Endpoint Service Name:** A text input field containing a service name, a "Verify" button, and a help icon. Below it, a green checkmark indicates "Service name found. Service Type: Interface".
- Create a Private Domain Name:** A checked checkbox with a help icon.
- VPC:** A dropdown menu showing "vpc-.../16" and a "View VPCs" link.
- Subnet:** A dropdown menu showing "subnet-.../24" and a "View Subnets" link with "Available IP Addresses: 245".
- IPv4 Address:** Two buttons: "Automatically assign IP address" (selected) and "Manually specify IP address".
- Access Control:** A toggle switch.
- Tag:** A section with a note: "It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags". It includes "Tag key" and "Tag value" input fields and a note "You can add 10 more tags.".
- Description:** A large text area with a character count "0/512".

表 7-4 创建终端节点参数说明

| 参数名称        | 说明                                                                                                                                                  |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 区域          | 终端节点所在区域。与Kafka实例保持一致。                                                                                                                              |
| 服务类别        | <ul style="list-style-type: none"> <li>云服务：当您要连接的终端节点服务为云服务时，需要选择“云服务”。</li> <li>按名称查找服务：当您要连接的终端节点服务为用户私有服务时，需要选择“按名称查找服务”。选择“按名称查找服务”。</li> </ul> |
| 服务名称        | 输入步骤8中记录的终端节点服务名称，单击“验证”。显示“已找到服务”后，继续后续操作。                                                                                                         |
| (可选) 创建内网域名 | 如果您想以域名方式访问终端节点，请勾选“创建内网域名”。终端节点创建完成后，便能通过内网域名直接访问终端节点。                                                                                             |
| 虚拟私有云       | 终端节点所属的虚拟私有云。选择Kafka客户端所属的VPC。                                                                                                                      |
| 子网          | 终端节点所属的子网。选择Kafka客户端所属的子网。                                                                                                                          |
| IPv4地址      | 终端节点的IPv4地址。选择“自动分配IPv4地址”。                                                                                                                         |

**步骤4** 单击“Create Now”，进入规格确认页面。

**步骤5** 确认无误后，提交请求。

**步骤6** 创建成功后，返回终端节点页面，查看终端节点状态是否为“已接受”，“已接受”表示终端节点已成功连接至终端节点服务。

图 7-6 查看终端节点状态

| ID        | 虚拟私有云     | 状态  | 终端节点服务名称 | 类型 | 创建时间                        | 操作 |
|-----------|-----------|-----|----------|----|-----------------------------|----|
| 8fded5... | vpc-kafka | 已接受 | cn-...   | 接口 | 2021/02/01 14:32:35 GMT+... | 删除 |

**步骤7** 单击终端节点ID，在“基本信息”页签，查看并记录节点IP。您可以使用该IP访问终端节点，进行跨VPC资源通信。

如果您想通过域名访问终端节点，进行跨VPC资源通信，在“基本信息”页签查看并记录内网域名。

**注意**

复制内网域名时，不要复制最后一个点。正确的域名格式为“vpcep-5xxx7-e302-4ab4-bc6c-54xxxe52.xx-xx-xx.xxx.com”。

**步骤8** 参考**步骤1~步骤7**，为**步骤9**中创建的终端节点服务创建终端节点，查看并记录节点IP。

如果您想通过域名访问终端节点，请记录内网域名。

----结束

## 步骤四：修改 advertised.listeners IP

**步骤1** 在管理控制台左上角单击☰，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤2** 单击Kafka实例名称，进入实例详情页面。

**步骤3** 在“概览”页面的“高级配置”区域，单击“修改”，将“跨VPC访问”的“advertised.listeners IP/域名”修改为**步骤7**和**步骤8**中记录的节点IP。节点IP必须与Port ID一一对应，否则会导致网络不通。修改完后，单击“保存”。

如果您想通过域名访问终端节点，在“概览”页面的“高级配置”区域，单击“修改”。在“advertised.listeners IP/域名”列，单击=，切换到域名模式。将“advertised.listeners IP/域名”修改为**步骤7**和**步骤8**中记录的内网域名。内网域名必须与Port ID一一对应，否则会导致网络不通。

图 7-7 修改 advertised.listeners IP

Advanced Settings

Cross-VPC Access Protocol SASL\_SSL

Cross-VPC Access ⓘ

Save Cancel

| listeners IP Address |        | advertised.listeners IP Address/Domain Name | Port | Port ID                              |
|----------------------|--------|---------------------------------------------|------|--------------------------------------|
| 192.168.0.194        | = IPv4 | 192 · 168 · 0 · 211                         | 9011 | c7af42e5-3586-429d-a173-c76e9c25e29  |
| 192.168.0.32         | = IPv4 | 192 · 168 · 0 · 97                          | 9011 | d69f3c5f-0d78-41ee-ba1a-3df64c4f8d3e |
| 192.168.0.245        | = IPv4 | 192 · 168 · 0 · 17                          | 9011 | 8d5f77e4-e9bc-467a-8309-0513d8fc6666 |

----结束

## 步骤五：验证接口连通性

参考[使用客户端连接Kafka（明文接入）](#)或者[使用客户端连接Kafka（密文接入）](#)，测试是否可以生产和消费消息。

测试接口连通性时，注意以下几点：

- 连接Kafka实例的地址为“advertised.listeners IP:9011”，以图7-7为例，连接Kafka实例的地址为“192.168.0.211:9011,192.168.0.97:9011,192.168.0.17:9011”。  
如果使用域名访问终端节点，连接Kafka实例的地址为“域名:9011”，例如“vpcep-5xxx7-e302-4ab4-bc6c-54xxe52.xx-xx-xx.xxx.com:9011”。
- 在Kafka实例安全组的入方向规则中放通9011端口，以及198.19.128.0/17网段的地址。
- 如果Kafka实例的子网配置了网络ACL功能，需要在网络ACL的入方向规则中放通198.19.128.0/17网段的地址，以及VPC终端节点涉及的子网。

### 说明

198.19.128.0/17是为VPC终端节点分配的网段，使用VPC终端节点需要放通此网段。

## 7.2 配置 Kafka 访问控制

### 7.2.1 配置 Kafka ACL 用户

Kafka实例开启密文接入后，支持对Topic进行ACL（Access Control List）权限管理，您可以在Topic中为不同的用户设置不同的生产消费消息的权限，以达到用户之间的权限隔离。

本章节主要介绍Kafka实例开启密文接入后，如何创建用户、重置用户密码和删除用户。创建用户后，对Topic进行用户授权的操作，请参见[设置Kafka Topic权限](#)。

在控制台创建用户的方法有两种，两种方法对应的重置密码的步骤有区别。

- 初始用户：在创建实例时设置的用户，如果忘记了密码，请参考[重置用户密码（初始用户）](#)，重置密码。
- 非初始用户：在“实例管理 > 用户管理”页面创建的用户，如果忘记了密码，请参考[重置用户密码（非初始用户）](#)，重置密码。

### 约束与限制

- 创建Kafka实例时设置的用户无法删除。
- 重置用户密码会导致业务中断，您需要及时在客户端配置文件或代码中修改用户密码。
- 一个Kafka实例最多创建的用户数量在控制台存在两种限制，一种为20个，另一种为500个，具体以控制台为准。

### 前提条件

- Kafka实例已开启密文接入。
- Kafka实例的“状态”为“运行中”。

## 创建用户

- 步骤1** 登录管理控制台。
- 步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。
- 步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。
- 步骤4** 单击Kafka实例名称，进入实例详情页面。
- 步骤5** 在左侧导航栏单击“实例管理 > 用户管理”，进入用户列表页面。
- 步骤6** 在页面左上方，单击“创建用户”。
- 步骤7** 在弹出的“创建用户”对话框中，参考表7-5，设置用户信息。

表 7-5 创建用户参数说明

| 参数  | 说明                                                                                                                                                                                                                |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 用户名 | 连接Kafka实例的用户名，用户名支持自定义，但需要符合命名规则：长度为4~64个字符，由英文字母开头，只能由英文字母、数字、中划线、下划线组成。                                                                                                                                         |
| 密码  | 连接Kafka实例的密码，密码需要符合以下规则： <ul style="list-style-type: none"> <li>长度为8~32个字符。</li> <li>至少包含以下字符中的3种：大写字母、小写字母、数字、特殊字符`~!@#\$%^&amp;*()-_+=\ [{ }];:","&lt;.&gt;?`和空格，并且不能以-开头。</li> <li>不能与用户名或倒序的用户名相同。</li> </ul> |

- 步骤8** 单击“确定”，完成用户的创建。
- 在用户列表页面，查看新创建的用户。
- 用户创建成功后，参考[设置Kafka Topic权限](#)为用户授权。

----结束

## 重置用户密码（初始用户）

- 步骤1** 登录管理控制台。
- 步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。
- 步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。
- 步骤4** 通过以下任意一种方法，重置初始用户的密码。
  - 在需要重置密码的Kafka实例右侧，单击“更多 > 重置Kafka密码”，弹出“重置Kafka密码”对话框。

- 单击Kafka实例名称，进入实例详情页面。单击右上角的“... > 重置Kafka密码”，弹出“重置Kafka密码”对话框。
- 单击Kafka实例名称，进入实例详情页面。在“概览”页面，在“连接信息 > 用户名”后，单击“重置密码”，弹出“重置Kafka密码”对话框。
- 单击Kafka实例名称，进入实例详情页面。在“实例管理 > 用户管理”页面，在待重置密码的用户所在行，单击“重置密码”，弹出“重置密码”对话框。

**步骤5** 输入“新密码”，单击“确定”，完成密码重置。

- 如果重置密码成功，界面提示重置实例的密码成功。
- 如果重置密码失败，界面提示重置实例的密码失败，请重新尝试重置密码操作。如果多次重置失败，请联系客服处理。

**只有所有代理都重置密码成功，系统才会提示重置密码成功，否则会提示重置失败。**

----结束

## 重置用户密码（非初始用户）

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击待重置用户密码的Kafka实例名称，进入实例详情页面。

**步骤5** 在“实例管理 > 用户管理”页面，在待重置密码的用户所在行，单击“重置密码”。

**步骤6** 输入“新密码”，单击“确定”，完成密码重置。

- 如果重置密码成功，界面提示重置实例的密码成功。
- 如果重置密码失败，界面提示重置实例的密码失败，请重新尝试重置密码操作。如果多次重置失败，请联系客服处理。

### 说明

只有所有代理都重置密码成功，系统才会提示重置密码成功，否则会提示重置失败。

----结束

## 删除用户

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > 用户管理”，进入用户列表页面。

**步骤6** 通过以下任意一种方法，删除用户。

- 在待删除的用户所在行，单击“删除”。
- 勾选用户名左侧的方框，可选一个或多个，单击信息栏左上侧的“删除”。

**步骤7** 在弹出的“删除用户”对话框中，单击“确定”，完成用户的删除。

已删除的用户不再显示在用户列表中，表明用户已成功删除。

----结束

## 导出用户列表

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > 用户管理”，进入用户列表页面。

**步骤6** 通过以下任意一种方法，导出用户列表。

- 勾选待导出的用户，单击“导出 > 导出已选中数据到XLSX”，导出指定用户列表。
- 单击“导出 > 导出全部数据到XLSX”，导出全部用户列表。

请注意：初始用户无法被选中，如果要导出初始用户，请使用“导出全部数据到XLSX”。

----结束

## 7.3 配置 Kafka 客户端

### 7.3.1 Kafka 客户端参数配置建议

Kafka客户端的配置参数很多，以下提供Producer和Consumer几个常用参数配置。不同版本的Kafka客户端参数名称可能不同，以下配置参数适用于1.1.0及以上版本。其他参数和版本配置，请参考[Kafka配置](#)。

表 7-6 Producer 参数

| 参数                 | 默认值   | 推荐值                    | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------|-------|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| acks               | 1     | 高可靠: all或者-1<br>高吞吐: 1 | <p>收到Server端确认信号个数, 表示producer需要收到多少个这样的确认信号, 算消息发送成功。acks参数代表了数据备份的可用性。常用选项:</p> <p>acks=0: 表示producer不需要等待任何确认收到的信息, 副本将立即加到socket buffer并认为已经发送。没有任何保障可以保证此种情况下server已经成功接收数据, 同时重试配置不会发生作用(因为客户端不知道是否失败)回馈的offset会总是设置为-1。</p> <p>acks=1: 这意味着至少要等待leader已经成功将数据写入本地log, 但是并没有等待所有follower是否成功写入。如果follower没有成功备份数据, 而此时leader又无法提供服务, 则消息会丢失。</p> <p>acks=all或者-1: 这意味着leader需要等待ISR中所有备份都成功写入日志。只要任何一个备份存活, 数据都不会丢失。min.insync.replicas指定必须确认写入才能被认为成功的副本的最小数量。</p> |
| retries            | 0     | 结合实际业务调整               | <p>客户端发送消息的重试次数。值大于0时, 这些数据发送失败后, 客户端会重新发送。</p> <p>注意, 这些重试与客户端接收到发送错误时的重试没有什么不同。允许重试将潜在的改变数据的顺序, 如果这两个消息记录都是发送到同一个partition, 则第一个消息失败第二个发送成功, 则第二条消息会比第一条消息出现要早。</p> <p>针对网络闪断场景, 生产者建议配置重试能力, 推荐重试次数retries=3, 重试间隔retry.backoff.ms=1000。</p>                                                                                                                                                                                                                           |
| request.timeout.ms | 30000 | 结合实际业务调整               | <p>设置一个请求最大等待时间(单位为ms), 超过这个时间则会抛Timeout异常。</p> <p>超时时间如果设置大一些, 如127000(127秒), 高并发的场景中, 能减少发送失败的情况。</p>                                                                                                                                                                                                                                                                                                                                                                   |

| 参数                   | 默认值                                                                                    | 推荐值                 | 说明                                                                                                                                                                                                                      |
|----------------------|----------------------------------------------------------------------------------------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| block.on.buffer.full | TRUE                                                                                   | TRUE                | TRUE表示当我们内存用尽时，停止接收新消息记录或者抛出错误。<br>默认情况下，这个设置为TRUE。然而某些阻塞可能不值得期待，因此立即抛出错误更好。如果设置为false，则producer抛出一个异常错误：BufferExhaustedException                                                                                       |
| batch.size           | 16384                                                                                  | 262144              | 默认的批量处理消息字节数上限。producer将试图批处理消息记录，以减少请求次数。这将改善client与server之间的性能。不会试图处理大于这个字节数的消息字节数。<br>发送到brokers的请求将包含多个批量处理，其中会包含对每个partition的一个请求。<br>较小的批量处理数值比较少用，并且可能降低吞吐量（0则会禁用批量处理）。较大的批量处理数值将会浪费更多内存空间，这样就需要分配特定批量处理数值的内存大小。 |
| buffer.memory        | 33554432                                                                               | 67108864            | producer可以用来缓存数据的内存大小。如果数据产生速度大于向broker发送的速度，producer会阻塞或者抛出异常，以“block.on.buffer.full”来表明。<br>这项设置将和producer能够使用的总内存相关，但并不是一个硬性的限制，因为不是producer使用的所有内存都是用于缓存。一些额外的内存会用于压缩（如果引入压缩机制），同样还有一些用于维护请求。                       |
| enable.idempotence   | <ul style="list-style-type: none"> <li>3.0之前版本：false</li> <li>3.0及之后版本：true</li> </ul> | 如果无幂等需求，建议设置为false。 | 在生产者客户端中开启幂等，然后生产消息，此时您会在消费者客户端或Kafka控制台的“实例管理 > 消息查询”中观察到消息offset不连续的现象。这是因为开启了幂等后，在生产消息时会产生一些元数据控制消息，这些控制消息也会生产到该Topic中，且它们对消费者不可见，从而造成offset不连续的现象。                                                                  |

表 7-7 Consumer 参数

| 参数                      | 默认值     | 推荐值                                                    | 说明                                                                                                                                                                                                                                                                                                     |
|-------------------------|---------|--------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| auto.commit.enable      | TRUE    | FALSE                                                  | <p>如果为真，consumer所fetch的消息的offset将会自动的同步到zookeeper。这项提交的offset将在进程无法提供服务时，由新的consumer使用。</p> <p>约束：设置为false后，需要先成功消费再提交，这样可以避免消息丢失。</p>                                                                                                                                                                 |
| auto.offset.reset       | latest  | earliest                                               | <p>没有初始化offset或者offset被删除时，可以设置以下值：</p> <ul style="list-style-type: none"> <li>• earliest：自动复位offset为最早</li> <li>• latest：自动复位offset为最新</li> <li>• none：如果没有发现offset，则向消费者抛出异常</li> <li>• anything else：向消费者抛出异常。</li> </ul> <p>如果将此配置设置为latest，新增分区时，生产者可能会在消费者重置初始偏移量之前开始向新增加的分区发送消息，从而导致部分消息丢失。</p> |
| connections.max.idle.ms | 600000  | 30000                                                  | 空连接的超时时间（单位为ms），设置为30000可以在网络异常场景下减少请求卡顿的时间。                                                                                                                                                                                                                                                           |
| max.poll.records        | 500     | 需小于max.poll.interval.ms时间内的消费处理能力。                     | 消费者单次从Broker中拉取的最大消息条数。                                                                                                                                                                                                                                                                                |
| max.poll.interval.ms    | 300000  | 如果在两次poll之间存在复杂、耗时的逻辑，需要延长该参数值。                        | 消费者两次拉取消息的最大时间间隔，单位为ms。如果两次拉取消息的时间间隔超过该间隔时间，就视为此次消费失败，消费者被踢出消费组，触发Rebalance。                                                                                                                                                                                                                           |
| heartbeat.interval.ms   | 3000    | >=3000                                                 | 消费者和Kafka之间的心跳间隔，单位为ms。                                                                                                                                                                                                                                                                                |
| session.timeout.ms      | 10000   | 将该参数值设置为heartbeat.interval.ms的3倍以上。                    | 使用消费组管理offset时，消费者与Broker的会话超时时间，单位为ms。                                                                                                                                                                                                                                                                |
| fetch.max.bytes         | 1000000 | max.request.size < message.max.bytes < fetch.max.bytes | 消费者单次从Broker中拉取消息的最大字节数。                                                                                                                                                                                                                                                                               |

## 7.3.2 Kafka 客户端使用建议

### consumer 使用建议

1. consumer的owner线程需确保不会异常退出，避免客户端无法发起消费请求，阻塞消费。
2. 确保处理完消息后再做消息commit，避免业务消息处理失败，无法重新拉取处理失败的消息。
3. 通常不建议对每条消息都进行commit，如果对每条消息都进行了commit，会导致OFFSET\_COMMIT请求过多，进而导致CPU使用率过高。例如：如果一个消费请求拉取1000条消息，每条都commit，则commit请求TPS是消费的1000倍，消息体越小，这个比例越大。建议隔一定条数或时间，批量commit，或打开enable.auto.commit，这样设置会存在一个缺点，即在客户端故障时，可能丢失一部分缓存的消费进度，导致重复消费。请根据业务实际情况，设置批量commit。
4. consumer不能频繁加入和退出group，频繁加入和退出，会导致consumer频繁做rebalance，阻塞消费。
5. 同一消费组内consumer数量不能超过该消费组订阅的分区总数，否则会有consumer拉取不到消息。
6. consumer需周期poll，维持和server的心跳，避免心跳超时，导致consumer频繁加入和退出，阻塞消费。
7. consumer拉取的消息本地缓存应有大小限制，避免OOM（Out of Memory）。
8. consumer session设置为30秒，session.timeout.ms=30000。
9. Kafka不能保证消费重复的消息，业务侧需保证消息处理的幂等性。
10. 消费线程退出要调用consumer的close方法，避免同一个组的其他消费者阻塞session.timeout.ms的时间。
11. 消费组名称开头不使用特殊字符（如#），使用特殊字符可能会导致云监控无法展示此消费组的监控数据。
12. 在消费逻辑中处理AuthorizationException，设置重试消费。如在SpringKafka中，可以在配置文件中增加以下配置。

```
# 设置授权异常重试间隔为10秒  
spring.kafka.listener.authorizationExceptionRetryInterval=10000
```

### producer 使用建议

1. 同步复制客户端需要配合使用：acks=all
2. 配置发送失败重试：retries=3
3. 发送优化：对于时延敏感的信息，设置linger.ms=0。对于时延不敏感的信息，设置linger.ms在100~1000之间。
4. 生产端的JVM内存要足够，避免内存不足导致发送阻塞。
5. 时间戳设置为与当地时间一致，避免时间戳为未来时间导致消息无法老化。
6. 尽量复用producer，不要频繁创建producer。当producer开启幂等时（生产者客户端3.0及之后的版本默认开启幂等），生产消息会在服务端创建生产者状态对象，若频繁创建producer，会导致服务端创建大量生产者状态对象后无法及时回收，服务端内存占用飙升，进而导致服务端性能急剧下降。如果不需要使用幂等功能，请将“enable.idempotence”设置为“false”。
7. 捕获AuthorizationException异常，建议在业务侧重试生产消息，可以采取有限重试及退避策略实现自愈。

## Topic 使用建议

配置要求：推荐3副本，同步复制，最小同步副本数为2，且同步副本数不能等于Topic副本数，否则宕机1个副本会导致无法生产消息。

创建方式：支持选择是否开启Kafka自动创建Topic的开关。选择开启后，表示向一个未创建的Topic生产或消费消息时，系统会自动创建此Topic。

## 其他建议

连接数限制：3000

消息大小：不能超过10MB

使用SASL\_SSL协议访问Kafka：确保DNS具有反向解析能力，或者在hosts文件配置Kafka所有节点IP和主机名映射，避免Kafka client做反向解析，阻塞连接建立。

磁盘容量申请超过业务量 \* 副本数的2倍，即保留磁盘空闲50%左右。

业务进程JVM内存使用确保无频繁FGC，否则会阻塞消息的生产和消费。

## 7.4 使用客户端连接 Kafka（明文接入）

本章节介绍如何使用开源的Kafka客户端访问关闭SASL的Kafka实例的方法。关闭SASL表示客户端访问Kafka实例时，无需进行认证，数据通过明文传输，性能更好。

### 约束与限制

Kafka实例的每个代理允许客户端单IP连接的个数为1000个，如果超过了，会出现连接失败问题。您可以通过[修改Kafka实例配置参数](#)来修改单IP的连接数，即修改max.connections.per.ip参数值。

### 前提条件

- 已连接客户端和Kafka实例之间的网络，具体网络要求请参考[连接Kafka网络要求](#)。
- 已配置正确的安全组。  
客户端访问关闭密文接入的Kafka实例前，Kafka实例需要配置正确的安全组规则，否则会连接失败。安全组规则的配置请参考[表7-2](#)。
- 已获取连接Kafka实例的地址。  
在Kafka控制台的“概览 > 连接信息”中获取实例连接地址。  
使用内网通过同一个VPC访问，Kafka连接地址如下图所示。

图 7-8 使用内网通过同一个 VPC 访问 Kafka 实例的连接地址

内网连接地址 IPv4 192.168.0.24:9092,192.168.0.224:9092,192.168.0.197:9092 

- 如果Kafka实例未开启自动创建Topic功能，在连接实例前，请先[创建Topic](#)，否则会连接失败。
- 已下载对应版本的Kafka命令行工具，确保Kafka实例版本与命令行工具版本相同。

- [Kafka命令行工具1.1.0版本](#)
- [Kafka命令行工具2.3.0版本](#)
- [Kafka命令行工具2.7.2版本](#)
- [Kafka命令行工具3.4.0版本](#)
- 已在服务器中安装[Java Development Kit 1.8.111或以上版本](#)，并配置 JAVA\_HOME与PATH环境变量，具体方法如下：  
使用执行用户在用户家目录下修改“.bash\_profile”，添加如下行。其中“/opt/java/jdk1.8.0\_151”为JDK的安装路径，请根据实际情况修改。  

```
export JAVA_HOME=/opt/java/jdk1.8.0_151
export PATH=$JAVA_HOME/bin:$PATH
```

 执行source .bash\_profile命令使修改生效。

## 命令行模式连接实例

以下操作命令以Linux系统为例进行说明：

### 步骤1 解压Kafka命令行工具。

进入文件压缩包所在目录，然后执行以下命令解压文件。

```
tar -zxf {kafka_tar}
```

其中，{kafka\_tar}表示命令行工具的压缩包名称。

例如：

```
tar -zxf kafka_2.12-2.7.2.tgz
```

### 步骤2 进入Kafka命令行工具的“/bin”目录下。

注意，Windows系统下需要进入“/bin/windows”目录下。

```
cd {kafka_tar}/bin
```

### 步骤3 执行如下命令进行生产消息。

```
./kafka-console-producer.sh --broker-list {连接地址} --topic {Topic名称}
```

表 7-8 生产消息参数说明

| 参数名称    | 说明                                                                                   |
|---------|--------------------------------------------------------------------------------------|
| 连接地址    | Kafka实例的连接地址。<br>从 <a href="#">前提条件</a> 中获取连接地址。                                     |
| Topic名称 | Kafka实例下创建的Topic名称。<br>如果Kafka实例开启了自动创建Topic功能，此参数值可以填写已创建的Topic名称，也可以填写未创建的Topic名称。 |

以获取的Kafka实例连接地址为

“10.xx.xx.45:9092,10.xx.xx.127:9092,10.xx.xx.103:9092”为例。执行完命令后输入内容，按“Enter”发送消息到Kafka实例，输入的每一行内容都将作为一条消息发送到Kafka实例。

```
[root@ecs-kafka bin]# ./kafka-console-producer.sh --broker-list
10.xx.xx.45:9092,10.xx.xx.127:9092,10.xx.xx.103:9092 --topic topic-demo
>Hello
>DMS
>Kafka!
>^C[root@ecs-kafka bin]#
```

如需停止生产使用**Ctrl+C**命令退出。

#### 步骤4 执行如下命令消费消息。

```
./kafka-console-consumer.sh --bootstrap-server {连接地址} --topic {Topic名称} --group {消费组名称} --from-
beginning
```

表 7-9 消费消息参数说明

| 参数名称    | 说明                                                |
|---------|---------------------------------------------------|
| 连接地址    | Kafka实例的连接地址。<br>从 <a href="#">前提条件</a> 中获取连接地址。  |
| Topic名称 | Kafka实例下创建的Topic名称。                               |
| 消费组名称   | 根据您的业务需求，设定消费组名称。消费组名称开头包含特殊字符，例如#号“#”时，监控数据无法展示。 |

消费消息示例如下：

```
[root@ecs-kafka bin]# ./kafka-console-consumer.sh --bootstrap-server
10.xx.xx.45:9092,10.xx.xx.127:9092,10.xx.xx.103:9092 --topic topic-demo --group order-test --from-beginning
Kafka!
DMS
Hello
^CProcessed a total of 3 messages
[root@ecs-kafka bin]#
```

如需停止消费使用**Ctrl+C**命令退出。

----结束

## 7.5 使用客户端连接 Kafka（密文接入）

本章节介绍如何使用开源的Kafka客户端访问开启SASL的Kafka实例的方法。开启SASL表示客户端访问Kafka实例时，需要进行SASL认证且数据加密传输，安全性更高。

### 约束与限制

- 由于安全问题，支持的加密套件为  
TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256，  
TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256和  
TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256。
- Kafka实例的每个代理允许客户端单IP连接的个数默认为1000个，如果超过了，会出现连接失败问题。您可以通过[修改Kafka实例配置参数](#)来修改单IP的连接数，即修改max.connections.per.ip参数值。

## 前提条件

- 已连接客户端和Kafka实例之间的网络，具体网络要求请参考[连接Kafka网络要求](#)。
- 已配置正确的安全组。  
客户端访问开启密文接入的Kafka实例时，实例需要配置正确的安全组规则，否则会连接失败。安全组规则的配置请参考[表7-2](#)。
- 已获取连接Kafka实例的地址。  
在Kafka控制台的“概览 > 连接信息”中获取实例连接地址。  
使用内网通过同一个VPC访问，Kafka连接地址如下图所示。

图 7-9 使用内网通过同一个 VPC 访问 Kafka 实例的连接地址

内网连接地址 IPv4 192.168.0.239:9093,192.168.0.182:9093,192.168.0.57:9093 

- 已获取开启的SASL认证机制。  
在Kafka实例详情页的“连接信息”区域，查看“开启的SASL认证机制”。很久前创建的Kafka实例在详情页如果未显示“开启的SASL认证机制”，默认使用PLAIN机制。

图 7-10 开启的 SASL 认证机制

开启的SASL认证机制 SCRAM-SHA-512

- 已获取启用的安全协议。  
在Kafka实例详情页的“连接信息”区域，查看“启用的安全协议”。很久前创建的Kafka实例在详情页如果未显示“启用的安全协议”，默认使用SASL\_SSL协议。
- 如果Kafka实例未开启自动创建Topic功能，在连接实例前，请先[创建Topic](#)，否则会连接失败。
- 已下载client.truststore.jks证书。如果没有，在控制台单击Kafka实例名称，进入实例详情页面，在“连接信息 > SSL证书”所在行，单击“下载”。下载压缩包后解压，获取压缩包中的客户端证书文件：client.truststore.jks。
- 已下载对应版本的Kafka命令行工具，确保Kafka实例版本与命令行工具版本相同。
  - [Kafka命令行工具1.1.0版本](#)
  - [Kafka命令行工具2.3.0版本](#)
  - [Kafka命令行工具2.7.2版本](#)
  - [Kafka命令行工具3.4.0版本](#)
- 已在服务器中安装[Java Development Kit 1.8.111或以上版本](#)，并配置JAVA\_HOME与PATH环境变量，具体方法如下：

使用执行用户在用户家目录下修改“.bash\_profile”，添加如下行。其中“/opt/java/jdk1.8.0\_151”为JDK的安装路径，请根据实际情况修改。

```
export JAVA_HOME=/opt/java/jdk1.8.0_151
export PATH=$JAVA_HOME/bin:$PATH
```

执行source .bash\_profile命令使修改生效。

## 命令行模式连接实例

以下操作命令以Linux系统为例进行说明。

**步骤1** 在客户端所在主机的“/etc/hosts”文件中配置host和IP的映射关系，以便客户端能够快速解析实例的Broker。

1. 执行以下命令，编辑“/etc/hosts”文件。  
`vim /etc/hosts`
2. 在键盘上按“i”，在“/etc/hosts”文件中增加host和IP的映射关系。  
`ip1 主机名称1`  
`ip2 主机名称2`  
`ip3 主机名称3`

**表 7-10** “/etc/hosts”文件参数说明

| 参数名称 | 说明                               |
|------|----------------------------------|
| IP地址 | Kafka实例的连接地址，从前提条件中获取。           |
| 主机名称 | 每个实例主机的名称。<br>主机的名称由您自行设置，但不能重复。 |

例如：

```
10.154.48.120 server01
10.154.48.121 server02
10.154.48.122 server03
```

3. 按“Esc”，然后输入以下命令，按“Enter”，保存并退出“/etc/hosts”文件。  
`:wq`

**步骤2** 解压Kafka命令行工具。

进入文件压缩包所在目录，然后执行以下命令解压文件。

```
tar -zxf {kafka_tar}
```

其中，`{kafka_tar}`表示命令行工具的压缩包名称。

例如：

```
tar -zxf kafka_2.12-2.7.2.tgz
```

**步骤3** 修改Kafka命令行工具配置文件。

配置文件包括生产者配置文件“producer.properties”和消费者配置文件“consumer.properties”，需要分别修改。

### 修改“producer.properties”文件：

1. 进入Kafka命令行工具的“/config”目录。  
`cd {kafka_tar}/config`
2. 编辑“producer.properties”文件。  
`vim producer.properties`
3. 在键盘上按“i”，然后在“producer.properties”文件中添加如下内容。
  - a. Kafka实例包含两种认证机制，根据前提条件中获取的**SASL认证机制**，将对**应认证机制的内容添加到配置文件中**。如果SCRAM-SHA-512和PLAIN都开启了，根据实际情况选择其中任意一种配置连接。

■ **PLAIN机制:**

```
sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \
username="*****" \
password="*****";
sasl.mechanism=PLAIN
```

表 7-11 SASL 认证机制参数说明

| 参数名称     | 说明                             |
|----------|--------------------------------|
| username | 首次开启密文接入时设置的用户名，或者创建用户时设置的用户名。 |
| password | 首次开启密文接入时设置的密码，或者创建用户时设置的密码。   |

■ **SCRAM-SHA-512机制:**

```
sasl.jaas.config=org.apache.kafka.common.security.scram.ScramLoginModule required \
username="*****" \
password="*****";
sasl.mechanism=SCRAM-SHA-512
```

表 7-12 SASL 认证机制参数说明

| 参数名称     | 说明                             |
|----------|--------------------------------|
| username | 首次开启密文接入时设置的用户名，或者创建用户时设置的用户名。 |
| password | 首次开启密文接入时设置的密码，或者创建用户时设置的密码。   |

- b. Kafka实例包含两种安全协议，根据前提条件中获取的[安全协议](#)，将对应安全协议的内容添加到配置文件中。

■ **SASL\_SSL:**

```
security.protocol=SASL_SSL
ssl.truststore.location={ssl_truststore_path}
ssl.truststore.password=dms@kafka
ssl.endpoint.identification.algorithm=
```

表 7-13 SASL\_SSL 参数说明

| 参数名称                    | 说明                                                                                               |
|-------------------------|--------------------------------------------------------------------------------------------------|
| ssl.truststore.location | client.truststore.jks证书的存放路径。<br>注意，Windows系统下证书路径中也必须使用“/”，不能使用Windows系统中复制路径时的“\”，否则客户端获取证书失败。 |
| ssl.truststore.password | Kafka客户端证书密码。<br>默认为dms@kafka，不可更改。                                                              |

| 参数名称                                  | 说明                                  |
|---------------------------------------|-------------------------------------|
| ssl.endpoint.identification.algorithm | 证书域名校验开关，为空则表示关闭。这里需要保持关闭状态，必须设置为空。 |

▪ **SASL\_PLAINTEXT:**

```
security.protocol=SASL_PLAINTEXT
```

- 按“Esc”，然后输入以下命令，按“Enter”，保存并退出“producer.properties”文件。

```
:wq
```

**修改“consumer.properties”文件:**

- 编辑“consumer.properties”文件。

```
vim consumer.properties
```

- 在键盘上按“i”，然后在“consumer.properties”文件中添加如下内容。

- Kafka实例包含两种认证机制，根据前提条件中获取的**SASL认证机制**，将**对应认证机制的内容添加到配置文件中**。如果SCRAM-SHA-512和PLAIN都开启了，根据实际情况选择其中任意一种配置连接。

▪ **PLAIN机制:**

```
sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \
username="*****" \
password="*****";
sasl.mechanism=PLAIN
```

表 7-14 SASL 认证机制参数说明

| 参数名称     | 说明                             |
|----------|--------------------------------|
| username | 首次开启密文接入时设置的用户名，或者创建用户时设置的用户名。 |
| password | 首次开启密文接入时设置的密码，或者创建用户时设置的密码。   |

▪ **SCRAM-SHA-512机制:**

```
sasl.jaas.config=org.apache.kafka.common.security.scram.ScramLoginModule required \
username="*****" \
password="*****";
sasl.mechanism=SCRAM-SHA-512
```

表 7-15 SASL 认证机制参数说明

| 参数名称     | 说明                             |
|----------|--------------------------------|
| username | 首次开启密文接入时设置的用户名，或者创建用户时设置的用户名。 |
| password | 首次开启密文接入时设置的密码，或者创建用户时设置的密码。   |

- Kafka实例包含两种安全协议，根据前提条件中获取的**安全协议**，将**对应安全协议的内容添加到配置文件中**。

▪ **SASL\_SSL:**

```
security.protocol=SASL_SSL
ssl.truststore.location={ssl_truststore_path}
ssl.truststore.password=dms@kafka
ssl.endpoint.identification.algorithm=
```

表 7-16 SASL\_SSL 参数说明

| 参数名称                                  | 说明                                                                                               |
|---------------------------------------|--------------------------------------------------------------------------------------------------|
| ssl.truststore.location               | client.truststore.jks证书的存放路径。<br>注意，Windows系统下证书路径中也必须使用“/”，不能使用Windows系统中复制路径时的“\”，否则客户端获取证书失败。 |
| ssl.truststore.password               | Kafka客户端证书密码。<br><b>默认为dms@kafka，不可更改。</b>                                                       |
| ssl.endpoint.identification.algorithm | 证书域名校验开关，为空则表示关闭。这里需要保持关闭状态，必须设置为空。                                                              |

▪ **SASL\_PLAINTEXT:**

```
security.protocol=SASL_PLAINTEXT
```

- 按“Esc”，然后输入以下命令，按“Enter”，保存并退出“consumer.properties”文件。

```
:wq
```

**步骤4** 进入Kafka命令行工具的“/bin”目录下。

注意，Windows系统下需要进入“/bin/windows”目录下。

```
cd ../bin
```

**步骤5** 执行如下命令进行生产消息。

```
./kafka-console-producer.sh --broker-list {连接地址} --topic {Topic名称} --producer.config ../config/producer.properties
```

表 7-17 生产消息参数说明

| 参数名称    | 说明                                                                                   |
|---------|--------------------------------------------------------------------------------------|
| 连接地址    | Kafka实例的连接地址。<br>从 <b>前提条件</b> 中获取连接地址。                                              |
| Topic名称 | Kafka实例下创建的Topic名称。<br>如果Kafka实例开启了自动创建Topic功能，此参数值可以填写已创建的Topic名称，也可以填写未创建的Topic名称。 |

如下示例，Kafka实例连接地址为

“10.xx.xx.45:9093,10.xx.xx.127:9093,10.xx.xx.103:9093”。

执行完命令后，输入需要生产的消息内容，按“Enter”发送消息到Kafka实例，输入的每一行内容都将作为一条消息发送到Kafka实例。

```
[root@ecs-kafka bin]# ./kafka-console-producer.sh --broker-list
10.xx.xx.45:9093,10.xx.xx.127:9093,10.xx.xx.103:9093 --topic topic-demo --producer.config ../config/
producer.properties
>Hello
>DMS
>Kafka!
>^C[root@ecs-kafka bin]#
```

如需停止生产使用Ctrl+C命令退出。

#### 步骤6 执行如下命令消费消息。

```
./kafka-console-consumer.sh --bootstrap-server {连接地址} --topic {Topic名称} --group {消费组名称} --from-
beginning --consumer.config ../config/consumer.properties
```

表 7-18 消费消息参数说明

| 参数名称    | 说明                                                                                                                                                                     |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 连接地址    | Kafka实例的连接地址。<br>从前提条件中获取连接地址。                                                                                                                                         |
| Topic名称 | Kafka实例下创建的Topic名称。                                                                                                                                                    |
| 消费组名称   | 根据您的业务需求，设定消费组名称。消费组名称开头包含特殊字符，例如#号“#”时，监控数据无法展示。<br><b>警告</b><br>配置文件（即consumer.properties）中已经指定了消费组名称，请确保命令行中的消费组名称与配置文件中的相同，否则可能消费失败。<br>查看并修改配置文件中消费组名称的方法参考步骤6.1。 |

查看并修改配置文件中消费组名称的方法如下：

1. 进入Kafka命令行工具的“/config”目录下。  
cd ../config
2. 编辑“consumer.properties”文件。  
vim consumer.properties
3. 在键盘上按“i”，查看并修改消费组名称。  
“group.id”表示消费组的名称，您可以根据实际情况修改。  
# consumer group id  
group.id=test-consumer-group
4. 按“Esc”，然后输入以下命令，按“Enter”，保存并退出“consumer.properties”文件。  
:wq

消费消息示例如下：

```
[root@ecs-kafka bin]# ./kafka-console-consumer.sh --bootstrap-server
10.xx.xx.45:9093,10.xx.xx.127:9093,10.xx.xx.103:9093 --topic topic-demo --group order-test --from-beginning
--consumer.config ../config/consumer.properties
Hello
DMS
Kafka!
^CProcessed a total of 3 messages
[root@ecs-kafka bin]#
```

如需停止消费使用Ctrl+C命令退出。

----结束

## 7.6 使用控制台连接 Kafka

本章节介绍Kafka实例如何在控制台生产消息。此功能通过将指定消息发送给Kafka实例，验证业务逻辑是否正常。

### 前提条件

- 仅在实例的“状态”为“运行中”时，才可以在Topic中生产消息。
- 已[创建Topic](#)。

### 在控制台生产消息

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 在待生产消息的Topic所在行，单击“更多 > 生产消息”（实例已开启SASL）/“生产消息”（实例未开启SASL），弹出“生产消息”对话框。

**步骤7** 参考[表7-19](#)，设置消息的相关参数。

表 7-19 消息参数

| 参数名称    | 说明                                                                                                                                         |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 消息内容    | 输入消息的内容。长度为0~128000B。                                                                                                                      |
| 消息Key   | 输入消息的Key。                                                                                                                                  |
| 发送到指定分区 | 选择是否开启消息发送到指定的分区。 <ul style="list-style-type: none"><li>• 关闭：消息基于Key的hash值选择分区。</li><li>• 开启：消息将发送到指定的分区内。开启“发送到指定分区”后，需要选择分区ID。</li></ul> |

**步骤8** 单击“确定”，完成消息的发送。

消息发送成功后，您可以在“实例管理 > 消息查询”页面，查看已成功发送的消息。

----结束

# 8 管理消息

## 8.1 查看 Kafka 消息

客户端连接Kafka实例生产消费过程中，如果遇到消息丢失或消息消费失败等问题，可以通过消息查询功能来查看特定消息的内容和相关属性，帮助排查问题根因。

您可以查看指定Topic不同分区的偏移量、消息大小、创建时间以及消息正文。

### 约束与限制

- 当Topic中的数据量比较大时，单副本Topic查询消息可能会报“内部服务错误”，建议根据数据量适当减小查询时间范围。
- 使用消息内容或KEY查询消息时，由于查询资源和性能限制，最大搜索范围为10000条消息且所有消息的总大小不超过200MB，最多返回满足查询要求的前10条消息。
- 在所有分区查询消息时，由于查询资源和性能限制，最多返回500条数据。在单分区查询消息时，没有此限制。
- 控制台只能展示4KB以内的消息，超过4KB的消息，请单击“下载消息”，下载到本地查看消息完整的内容。

### 查看 Kafka 消息

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > 消息查询”，进入消息列表页面。

**步骤6** Kafka实例支持两种查询消息的方法：按偏移量查询和按创建时间查询。

- 如果您知道消息所在Topic的分区和偏移量，可以使用按偏移量查询消息，具体查询参数参见[表8-1](#)。

表 8-1 按偏移量查询消息

| 参数名称    | 说明               |
|---------|------------------|
| Topic名称 | 选择待查询消息的Topic名称。 |
| 分区      | 输入消息所在的分区。       |
| 偏移量     | 输入消息的偏移量。        |

- 如果您知道消息发送时间段，但是不知道消息偏移量，可以使用按创建时间查询消息，具体查询参数参见表8-2。

表 8-2 按创建时间查询消息

| 参数名称    | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Topic名称 | 选择待查询消息的Topic名称。                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 分区      | <p>输入消息所在的分区。</p> <p>如果未设置具体分区，查询结果显示Topic所有分区的消息。</p> <p>基于分区的消息查询限制说明：</p> <ul style="list-style-type: none"> <li><b>所有分区查询限制：</b>由于查询资源和性能限制，最多返回<b>500条</b>消息，且所有消息总大小不超过<b>200MB</b>。</li> <li><b>500条消息的获取过程：</b>后台按分区顺序（从分区0开始）依次拉取各分区最新的500条消息，若总大小超过200MB则立即终止查询。然后将已查询到的消息按时间排序，最终返回<b>时间最新的500条消息</b>。</li> <li><b>单分区查询限制：</b>没有500条/200MB的限制，可以完整获取该分区内符合时间条件的<b>全部消息</b>。</li> </ul> <p><b>大数据量查询建议：</b>通过<b>缩小查询时间范围</b>或<b>指定分区</b>查询，提升查询效率。</p> |
| KEY查询   | <p>输入消息的KEY，查询结果为包含KEY的所有消息。</p> <p>例如：Topic包含两条消息，KEY分别为“abc”和“abcd”，在“KEY查询”中输入“abc”，查询结果会同时返回这两条消息。</p> <p>基于KEY的消息查询限制说明：由于查询资源和性能限制，最大搜索<b>10000条</b>消息且所有消息总大小不超过<b>200MB</b>，最多返回<b>包含KEY的前10条消息</b>。</p>                                                                                                                                                                                                                                                  |

| 参数名称 | 说明                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 内容查询 | <p>支持设置多个查询条件，查询结果为满足所有查询条件的消息。</p> <p>设置查询条件的方法如下：</p> <ol style="list-style-type: none"> <li>1. 在“查询内容”输入框中，单击鼠标左键，选择“包含”/“不包含”。</li> <li>2. 输入消息正文的关键字，按Enter。</li> <li>3. 如果需要设置多个查询条件，重复执行<a href="#">步骤6.a</a>和<a href="#">步骤6.b</a>。</li> </ol> <p>基于内容的消息查询限制说明：由于查询资源和性能限制，最大搜索<b>10000</b>条消息且所有消息总大小不超过<b>200MB</b>，最多返回符合关键字的<b>前10</b>条消息。</p> <p>查询超大消息（即单条消息超过20KB）或时间跨度大的消息时，建议将消息下载到本地，在本地进行关键字查询。</p> |
| 创建时间 | <p>消息的创建时间，设置待查询消息的时间范围。</p> <p><b>警告</b><br/>当Topic中的数据量比较大时，单副本Topic查询消息可能会报“内部服务错误”，建议根据数据量适当减小查询时间范围。</p>                                                                                                                                                                                                                                                                                                           |

**步骤7** 单击“搜索”，查询消息。

查询结果如下：

**图 8-1** 查询 Topic 消息

| Topic名称  | 分区 | 偏移量 | 消息大小 (B) | 创建时间                          | 操作                     |
|----------|----|-----|----------|-------------------------------|------------------------|
| topic-01 | 1  | 7   | 5        | 2021/03/19 11:27:30 GMT+08:00 | <a href="#">查看消息正文</a> |
| topic-01 | 2  | 7   | 3        | 2021/03/19 11:27:19 GMT+08:00 | <a href="#">查看消息正文</a> |
| topic-01 | 1  | 6   | 4        | 2021/03/19 11:27:11 GMT+08:00 | <a href="#">查看消息正文</a> |
| topic-01 | 2  | 6   | 4        | 2021/03/19 11:27:09 GMT+08:00 | <a href="#">查看消息正文</a> |
| topic-01 | 1  | 5   | 6        | 2021/03/19 11:27:06 GMT+08:00 | <a href="#">查看消息正文</a> |

**表 8-3** 消息的参数说明

| 参数名称    | 说明                                                                   |
|---------|----------------------------------------------------------------------|
| Topic名称 | 消息所在的Topic名称。                                                        |
| 分区      | 消息所在的分区。                                                             |
| 偏移量     | 消息在分区中的位置。                                                           |
| 消息大小    | 消息存入磁盘的大小，单位为B。                                                      |
| 创建时间    | 消息的创建时间。创建时间由生产客户端在生产消息时通过CreateTime指定的，如果生产消息时没有设置此参数，创建时间会默认为1970。 |

**步骤8** 单击“查看消息正文”，弹出“查看消息正文”对话框，查看消息的内容，包括Topic名称、分区、偏移量、创建时间和消息正文。

控制台只能展示4KB以内的消息，超过4KB的消息，请单击“下载消息”，下载到本地查看消息完整的内容。

**步骤9** （可选）如果需要恢复默认设置，单击“重置”。

----结束

## 8.2 修改 Kafka 消息老化时间

老化时间即消息的最长保留时间，消费者必须在此时间结束前消费消息，否则消息将被删除。已删除的消息，无法被消费。

Topic老化时间默认为72小时，创建Topic成功之后，您可以根据业务需要修改Topic的老化时间。修改老化时间，不会影响业务。

修改老化时间可以通过以下方式实现：

- 在“实例管理 > Topic管理”中，修改老化时间。
- 在“实例管理 > 配置参数”中，修改“log.retention.hours”参数值，具体步骤请参考[修改Kafka实例配置参数](#)。

如果Topic已设置老化时间，则“实例管理 > 配置参数”中的log.retention.hours值对此Topic不生效。仅当Topic未设置老化时间时，该参数值才会生效。例如：Topic01设置的老化时间为60小时，log.retention.hours值为72小时，Topic01实际的老化时间仍为60小时。

### 约束与限制

- 批量修改老化时间时，每次最多修改50个Topic的老化时间。

### 修改单个 Topic 消息老化时间

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 通过以下任意一种方法，修改Topic老化时间。

- 勾选Topic名称左侧的方框，可选一个或多个，单击信息栏左上侧的“编辑Topic”。
- 在待修改老化时间的Topic所在行，单击“编辑”。

**步骤7** 在“编辑Topic”对话框中，输入老化时间（取值范围：1~720），单击“确定”。

设置成功后，在Topic列表页面，查看修改后的老化时间。

----结束

## 批量修改 Topic 消息老化时间

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

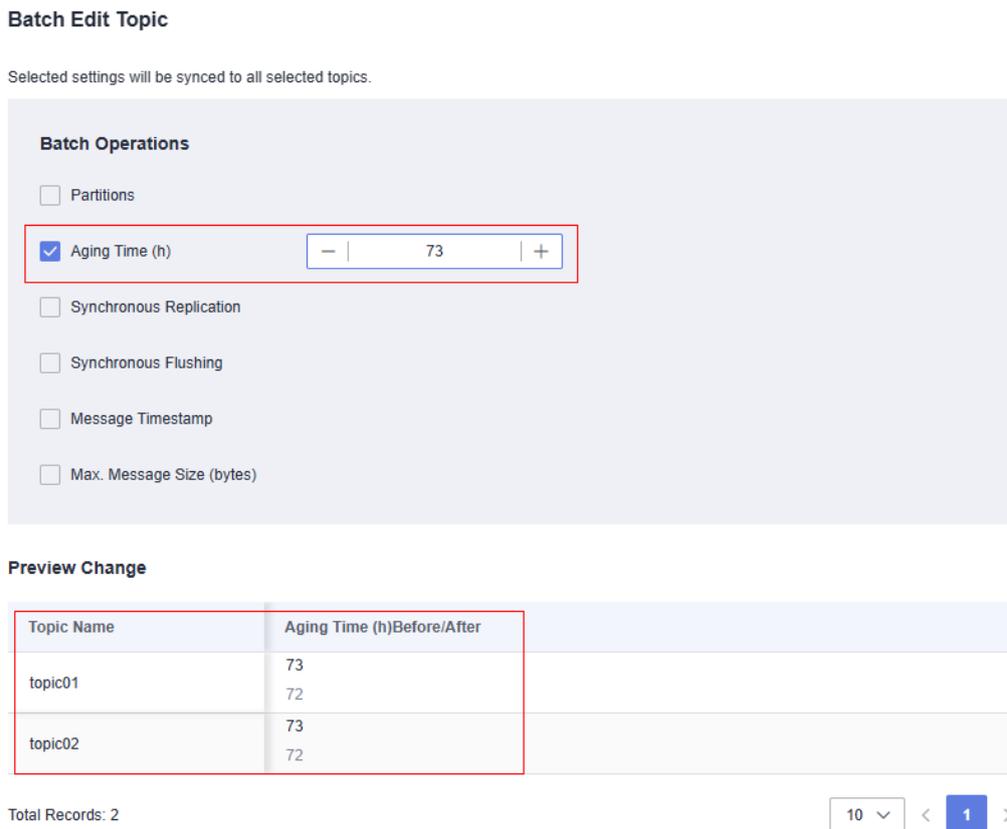
**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 勾选待修改老化时间的Topic，单击信息栏左上侧的“批量编辑Topic”，弹出“批量编辑Topic”对话框。

**步骤7** 在“批量操作”区域，勾选“老化时间（小时）”，并输入老化时间（取值范围：1~720）。在“批量数据预览”区域，查看修改前和修改后的老化时间。确认无误后，单击“确定”。

图 8-2 批量修改老化时间



设置成功后，在Topic列表页面，查看修改后的老化时间。

----结束

## 8.3 删除 Kafka 消息

本章节介绍如何在控制台删除Topic中存储的消息。

### 约束与限制

消息删除后无法恢复，请谨慎操作。

### 前提条件

删除消息前，请先在消费代码中设置“auto.offset.reset”参数。“auto.offset.reset”用来指定当Kafka中没有初始偏移量或者当前偏移量不存在（例如当前偏移量已被删除）时，消费者的消费策略。取值如下：

- latest：偏移量自动被重置到最晚偏移量。
- earliest：偏移量自动被重置到最早偏移量。
- none：向消费者抛出异常。

如果将此配置设置为latest，新增分区时，生产者可能会在消费者重置初始偏移量之前开始向新增加的分区发送消息，从而导致部分消息丢失。

### 删除 Kafka 消息

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 在待删除消息的Topic所在行，单击“更多 > 消息删除”，弹出“消息删除”对话框。

**步骤7** 设置消息删除参数，如表8-4所示。

图 8-3 消息删除

### 消息删除

您正在对Topic: topic-01进行消息删除, 请选择分区并输入偏移量进行删除 (每次最多可选择10个分区)

| 分区  | 偏移量 | 操作 |
|-----|-----|----|
| 分区0 | 10  | 删除 |

⊕ 添加分区

**⚠️** 1. 请先确认消息的偏移量区间, 若您输入的偏移量在区间之外, 消息将不会被删除。  
 2. 删除消息前, 请先在客户端中设置“auto.offset.reset”参数。“auto.offset.reset”用来指定当Kafka中没有初始偏移量或者当前偏移量不存在 (例如当前偏移量已被删除) 时, 消费者的消费策略。取值如下:

- latest: 偏移量自动被重置到最晚偏移量
- earliest: 偏移量自动被重置到最早偏移量
- none: 向消费者抛出异常

3. 消息删除后无法恢复, 请谨慎操作。

表 8-4 消息删除参数说明

| 参数  | 说明                                                                                                                                                                                                                         |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 分区  | 选择消息所在的分区编号。                                                                                                                                                                                                               |
| 偏移量 | 输入偏移量, 最早偏移量到输入偏移量之前的数据将被删除。<br>例如: 最早偏移量为2, 输入的偏移量为5, 此时会删除偏移量为2到4的消息。<br><b>警告</b> <ul style="list-style-type: none"> <li>• 如果“偏移量”设置为“-1”, 表示删除分区中所有的消息。</li> <li>• 如果您输入的偏移量不在指定分区的最早偏移量和最晚偏移量之间, 消息将不会被删除。</li> </ul> |

如果需要删除多个分区的消息, 单击“添加分区”, 设置需要删除消息的分区和偏移量。每次最多可选择10个分区。

**步骤8** 单击“确定”, 弹出“清理结果”对话框, 单击“确定”, 完成消息的删除。

图 8-4 清理结果

×

#### 清理结果

| 分区  | 偏移量 | 清理结果  |
|-----|-----|-------|
| 分区0 | -1  | ✔️ 成功 |

确定

**步骤9** 验证消息是否已成功删除。

在“消息查询”页面使用偏移量查询，输入如下参数，如果查询不到消息，表示已成功删除消息。

- 分区：输入[步骤7](#)中的分区编号。
- 偏移量：如果[步骤7](#)中的偏移量设置为“-1”，随机输入一个偏移量。如果[步骤7](#)中的偏移量不为“-1”，输入小于此偏移量的整数。

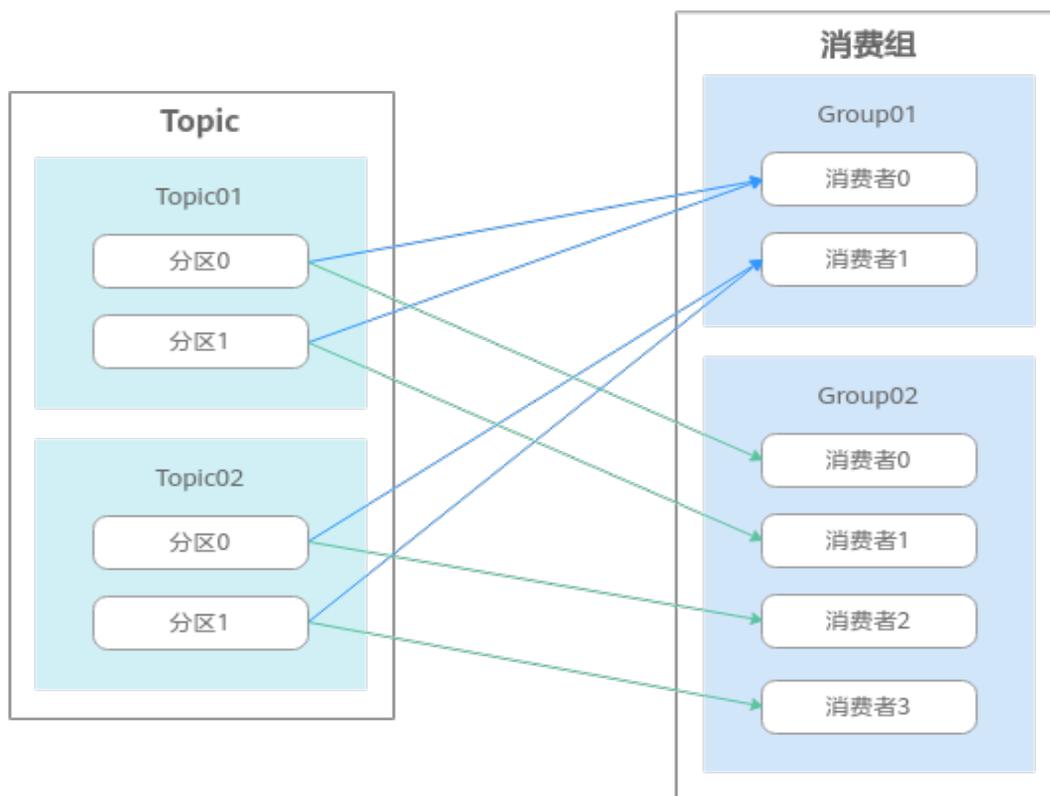
----结束

# 9 管理消费组

## 9.1 创建 Kafka 消费组

消费者是从Topic订阅消息的一方，消费组是由一个或多个消费者组成的。同一个消费组内，一个消费者可同时消费多个分区，但一个分区在同一时刻只能被一个消费者消费。

图 9-1 消费关系示例图



“`auto.create.groups.enable`”表示在消费者进入未创建的消费组时，会自动创建此消费组。

- 如果在配置参数中设置“auto.create.groups.enable”为“false”，在消费消息前，需要先创建消费组，否则会导致消费失败。
- 如果在配置参数中设置“auto.create.groups.enable”为“true”，系统会自动创建消费组，无需手动创建。

本章节介绍如何在控制台创建消费组，此操作不会导致Kafka实例重启。

## 约束与限制

- 如果“auto.create.groups.enable”设置为“true”，当消费组的状态为“EMPTY”且从未提交过offset，系统将在十分钟后自动删除该消费组。
- 如果“auto.create.groups.enable”设置为“false”，系统不会自动删除消费组。如果需要删除消费组，需要您手动删除。
- 如果消费组从未提交过offset，当Kafka实例重启后，该消费组会被删除。

## 创建 Kafka 消费组

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > 消费组管理”，进入消费组列表页面。

**步骤6** 单击“创建消费组”，弹出“创建消费组”对话框。

**步骤7** 参考表9-1设置消费组参数，单击“确定”。

表 9-1 消费组参数说明

| 参数    | 说明                                                                                                        |
|-------|-----------------------------------------------------------------------------------------------------------|
| 消费组名称 | 自定义名称，但需要符合命名规则：只能由英文字母、数字、下划线开头，且只能英文字母、数字、句点、中划线、下划线组成，长度3~64个字符。<br>消费组名称开头包含特殊字符，例如#号“#”，会导致监控数据无法展示。 |

在消费组列表页面，查看新创建的消费组。

----结束

## 9.2 查看 Kafka 消费组信息

消费组创建成功后，可查看消费组配置信息和状态信息。

## 查看消费组列表（控制台）

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > 消费组管理”，进入消费组列表页面。

消费组列表页面显示消费组名称、消费组状态和Coordinator (ID)。Coordinator (ID) 显示的数字表示Coordinator组件所在的代理，消费组的状态如下。

- DEAD：消费组内没有任何成员，且没有任何元数据。
- EMPTY：消费组内没有任何成员，存在元数据。
- PREPARING\_REBALANCE：准备开启Rebalance。
- COMPLETING\_REBALANCE：所有成员加入消费组。
- STABLE：消费组内成员可以正常消费。

图 9-2 消费组列表

| <input type="checkbox"/> 消费组名称 | 消费组状态 | Coordinator (ID) | 操作 |
|--------------------------------|-------|------------------|----|
| <input type="checkbox"/> test  | EMPTY | 2                | 删除 |

**步骤6** （可选）如果需要搜索某个消费组，在搜索框中，输入消费组名称、消费组状态、Coordinator (ID)或者关键字，按“Enter”。

**步骤7** （可选）如果需要刷新消费组列表，在右上角单击 。

----结束

## 查看消费组列表（Kafka 客户端）

- 未开启密文接入的Kafka实例，在Kafka客户端的“/bin”目录下，通过以下命令查询消费组列表。

```
./kafka-consumer-groups.sh --bootstrap-server {connection-address} --list
```

connection-address表示Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。

示例如下：

```
[root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
192.168.xx.xx:9092,192.xx.xx.212:9092,192.xx.xx.147:9092 --list
test
__consumer-group-dial-test
[root@ecs-kafka bin]#
```

- 已开启密文接入的Kafka实例，通过以下步骤查询消费组列表。
  - （可选）如果已经设置了用户名和密码，以及SSL证书配置，请跳过此步骤，执行**b**。否则请执行以下操作。  
在Kafka客户端的“/config”目录中创建“ssl-user-config.properties”文件，参考**步骤3**增加用户名和密码，以及SSL证书配置。

- b. 在Kafka客户端的“/bin”目录下，通过以下命令查询消费组列表。  
`./kafka-consumer-groups.sh --bootstrap-server {connection-address} --list --command-config ../config/{ssl-user-config.properties}`

表 9-2 查询消费组列表参数说明

| 参数名称                       | 说明                                     |
|----------------------------|----------------------------------------|
| connection-address         | Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。 |
| ssl-user-config.properties | 配置文件的名称，此配置文件用于存放用户名和密码，以及SSL证书配置信息。   |

示例如下：

```
[root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --list --command-config ../config/ssl-
user-config.properties
test
__consumer-group-dial-test
[root@ecs-kafka bin]#
```

## 9.3 查看 Kafka 消费者信息

如果消费组中存在正在连接Kafka实例的消费者，可查看消费组中该消费者的连接信息。

### 约束与限制

Kafka Manager由于缓存原因，显示的消费者连接地址可能不是当前消费者连接地址，建议重启Kafka Manager解决。

### 前提条件

消费组中的消费者处于连接Kafka实例时（即消费组状态为“STABLE”），才可以查看消费者列表和消费者连接地址。

### 查看消费者列表（控制台）

- 步骤1 登录管理控制台。
- 步骤2 在管理控制台左上角单击 ，选择Kafka实例所在的区域。
- 步骤3 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。
- 步骤4 单击Kafka实例的名称，进入实例详情页面。
- 步骤5 在左侧导航栏选择“实例管理 > 消费组管理”，进入消费组列表页面。
- 步骤6 单击待查看消费者列表的消费组名称，进入消费组详情页。
- 步骤7 在“消费者列表”页签，查看消费者列表。

在消费者列表中，可以查看消费者ID、消费者地址和客户端ID。

**步骤8** （可选）如果需要查询某个消费者，在搜索框中，输入消费者ID，按“Enter”。

---结束

## 查看消费者列表（客户端）

- 未开启密文接入的Kafka实例，在Kafka客户端的“/bin”目录下，通过以下命令查询消费者列表。

```
./kafka-consumer-groups.sh --bootstrap-server {connection-address} --group {group-name} --members --describe
```

**表 9-3** 查看消费者列表参数说明

| 参数名称               | 说明                                     |
|--------------------|----------------------------------------|
| connection-address | Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。 |
| group-name         | 消费组名称。                                 |

示例如下：

```
[root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
192.168.xx.xx:9092,192.168.xx.xx:9092,192.168.xx.xx:9092 --group test --members --describe
GROUP          CONSUMER-ID                                     HOST          CLIENT-ID
#PARTITIONS
test           console-consumer-571a64fe-b0c4-47ce-833d-9e0da5a88d14 /192.168.0.215 console-
consumer 3
[root@ecs-kafka bin]#
```

- 已开启密文接入的Kafka实例，通过以下步骤查询消费者列表。
  - （可选）如果已经设置了用户名和密码，以及SSL证书配置，请跳过此步骤，执行**b**。否则请执行以下操作。  
在Kafka客户端的“/config”目录中创建“ssl-user-config.properties”文件，参考**步骤3**增加用户名和密码，以及SSL证书配置。
  - 在Kafka客户端的“/bin”目录下，通过以下命令查询消费者列表。  
./kafka-consumer-groups.sh --bootstrap-server {connection-address} --group {group-name} --members --describe --command-config ../config/{ssl-user-config.properties}

**表 9-4** 查看消费者列表参数说明

| 参数名称                       | 说明                                     |
|----------------------------|----------------------------------------|
| connection-address         | Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。 |
| group-name                 | 消费组名称。                                 |
| ssl-user-config.properties | 配置文件的名称，此配置文件用于存放用户名和密码，以及SSL证书配置信息。   |

示例如下：

```
[root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --group test --members --describe --
command-config ../config/ssl-user-config.properties

GROUP          CONSUMER-ID          HOST          CLIENT-ID
#PARTITIONS
test          console-consumer-566d0c82-07d3-4d87-9a6e-f57a9bc9fc69 /192.168.0.215 console-
consumer 3
[root@ecs-kafka bin]#
```

## 查看消费者连接地址（控制台）

- 步骤1 登录管理控制台。
- 步骤2 在管理控制台左上角单击 ，选择Kafka实例所在的区域。
- 步骤3 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。
- 步骤4 单击Kafka实例的名称，进入实例详情页面。
- 步骤5 在左侧导航栏选择“实例管理 > 消费组管理”，进入消费组列表页面。
- 步骤6 单击待查看消费者连接地址的消费组名称，进入消费组详情页。
- 步骤7 在“消费者列表”页签，查看消费者连接地址。

图 9-3 消费者列表



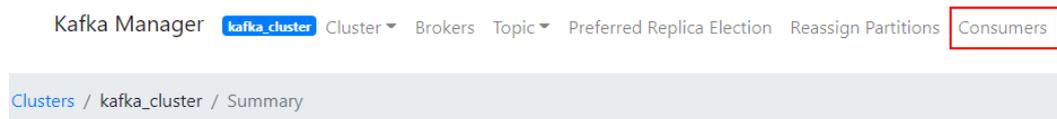
| 消费者ID                                           | 消费者地址            | 客户端ID      |
|-------------------------------------------------|------------------|------------|
| consumer-1-6883fa94-d355-4403-b5d8-251a16f2b53d | /128.128.128.183 | consumer-1 |

----结束

## 查看消费者连接地址（Kafka Manager）

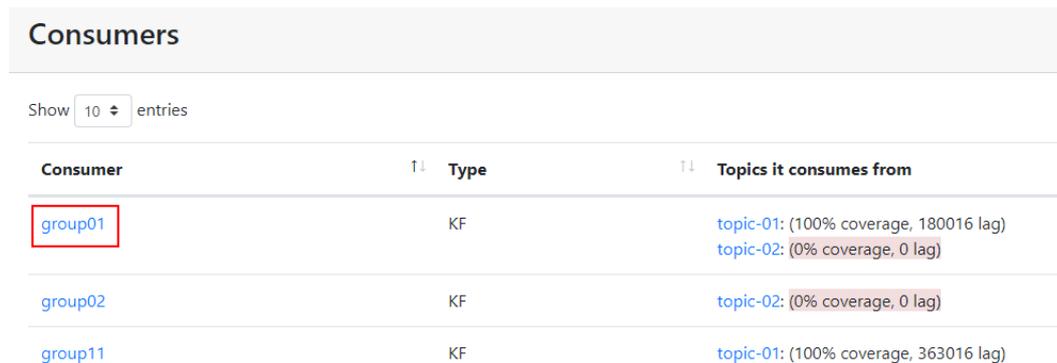
- 步骤1 登录Kafka Manager。
- 步骤2 单击“kafka\_cluster”，进入集群详情页。
- 步骤3 在顶部导航栏单击“Consumers”，进入消费组列表页面。

图 9-4 导航栏



- 步骤4 单击待查看消费者连接地址的消费组名称，进入消费组订阅的Topic列表页面。

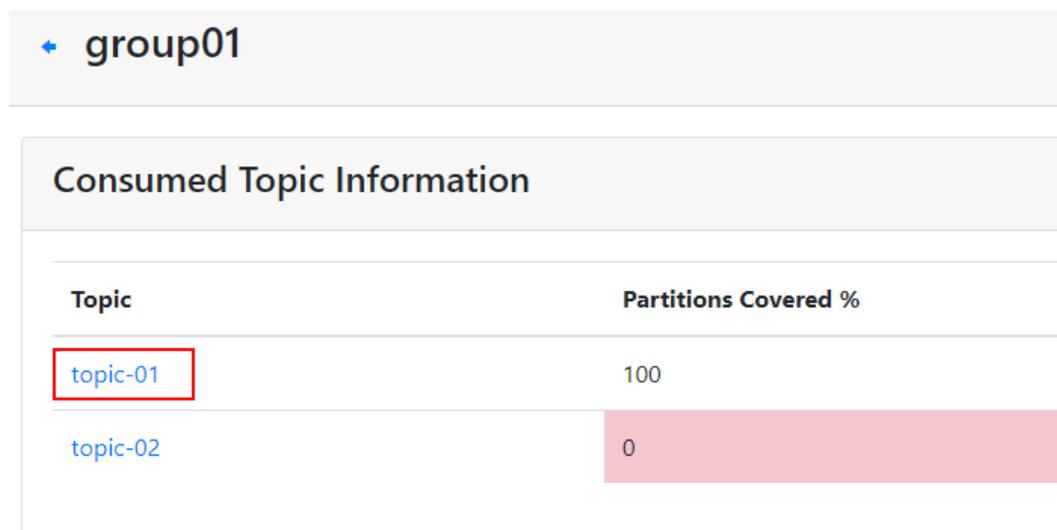
图 9-5 消费组列表页面



| Consumer | Type | Topics it consumes from                                                 |
|----------|------|-------------------------------------------------------------------------|
| group01  | KF   | topic-01: (100% coverage, 180016 lag)<br>topic-02: (0% coverage, 0 lag) |
| group02  | KF   | topic-02: (0% coverage, 0 lag)                                          |
| group11  | KF   | topic-01: (100% coverage, 363016 lag)                                   |

步骤5 单击待查看消费者连接地址的Topic名称，进入Topic详情页。

图 9-6 消费组订阅的 Topic 列表页面



| Topic    | Partitions Covered % |
|----------|----------------------|
| topic-01 | 100                  |
| topic-02 | 0                    |

步骤6 在“Consumer Instance Owner”中，查看消费者连接地址。

图 9-7 Topic 详情页



| Partition | LogSize | Consumer Offset | Lag    | Consumer Instance Owner                                       |
|-----------|---------|-----------------|--------|---------------------------------------------------------------|
| 0         | 33,333  | 0               | 33,333 | consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1/10.234.177.60 |
| 1         | 33,334  | 0               | 33,334 | consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1/10.234.177.60 |
| 2         | 33,333  | 0               | 33,333 | consumer-1-5d096c5f-159d-468d-8b10-7961dc6f49d1/10.234.177.60 |

----结束

## 9.4 查看和重置 Kafka 消费进度

消费进度表示消费者的消费位置，本章节介绍如何查看和重置消费进度，重置消费进度即修改消费者的消费位置。

## 约束与限制

重置消费进度可能会导致重复消费，请谨慎操作。

## 前提条件

Kafka实例不支持在线重置消费进度，请先将待重置消费进度的消费组停止消费，然后重置消费进度。停止待重置消费进度的消费者客户端后，需要等待一段时间（即ConsumerConfig.SESSION\_TIMEOUT\_MS\_CONFIG配置的时间，默认为1000毫秒），服务端才会认为此消费者已下线。

## 查看消费进度（控制台）

- 步骤1 登录管理控制台。
- 步骤2 在管理控制台左上角单击 ，选择Kafka实例所在的区域。
- 步骤3 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。
- 步骤4 单击Kafka实例的名称，进入实例详情页面。
- 步骤5 在左侧导航栏选择“实例管理 > 消费组管理”，进入消费组列表页面。
- 步骤6 单击待查看消费进度的消费组名称，进入消费组详情页。
- 步骤7 在“消费进度”页签，查看此消费组订阅的Topic列表、Topic的分区数、Topic中消息的总堆积数、Topic每个分区的消费进度。

表 9-5 消费进度参数说明

| 参数名称    | 说明                                                                                                                                 |
|---------|------------------------------------------------------------------------------------------------------------------------------------|
| Topic名称 | 消费组订阅的Topic名称。                                                                                                                     |
| 分区数     | Topic的分区数。                                                                                                                         |
| 总堆积数    | Topic中未被消费组消费的消息数。<br>总堆积数为采集时刻的瞬时值，而监控指标中的堆积数采集周期为1分钟，由于采集周期不同，可能会导致数值存在差异。具体现象请参见 <a href="#">消费进度显示有堆积消息，为什么监控中显示消息堆积数为0？</a> 。 |
| 分区编号    | Topic的分区号。                                                                                                                         |
| 堆积数     | 该分区中未被消费组消费的消息数。                                                                                                                   |
| 偏移量     | 该分区的偏移量。                                                                                                                           |
| 最大消息位置  | 该分区的最大消息位置。                                                                                                                        |
| 消费者ID   | 消费该分区消息的消费者ID。                                                                                                                     |
| 消费者地址   | 消费该分区消息的消费者地址。                                                                                                                     |

| 参数名称  | 说明                               |
|-------|----------------------------------|
| 客户端ID | 客户端标识符，该客户端用于连接Kafka实例并消费该分区的消息。 |

**步骤8** (可选) 如果需要查询某个Topic的消费进度，在搜索框中，输入Topic名称，按“Enter”。

**步骤9** (可选) 如果需要导出消费进度到本地，参考以下任意一种方法操作。

- 勾选待导出的Topic，单击“导出 > 导出已选中数据到XLSX”，导出指定Topic的消费进度。
- 单击“导出 > 导出全部数据到XLSX”，导出全部Topic的消费进度。

----结束

## 查看消费进度 (Kafka 客户端)

- 未开启密文接入的Kafka实例，在Kafka客户端的“/bin”目录下，通过以下命令查询消费进度。

```
./kafka-consumer-groups.sh --bootstrap-server ${connection-address} --offsets --describe --all-groups
```

connection-address表示Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。

示例如下：

```
[root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
192.168.xx.xx:9092,192.168.xx.xx:9092,192.168.xx.xx:9092 --offsets --describe --all-groups

Consumer group '_consumer-group-dial-test' has no active members.

GROUP          TOPIC          PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG
CONSUMER-ID    HOST           CLIENT-ID
_consumer-group-dial-test __dms_dial_test 0      350             350             0      -
-              -              -
_consumer-group-dial-test __dms_dial_test 1      350             350             0      -
-              -              -
_consumer-group-dial-test __dms_dial_test 2      350             350             0      -
-              -              -

Consumer group 'test' has no active members.

GROUP          TOPIC          PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG
CONSUMER-ID    HOST           CLIENT-ID
test           topic-01      0          5               5               0      -
test           topic-01      1          3               3               0      -
test           topic-01      2          10              10              0      -
[root@ecs-kafka bin]#
```

- 已开启密文接入的Kafka实例，通过以下步骤查询消费进度。
  - (可选) 如果已经设置了用户名和密码，以及SSL证书配置，请跳过此步骤，执行**b**。否则请执行以下操作。

在Kafka客户端的“/config”目录中创建“ssl-user-config.properties”文件，参考**步骤3**增加用户名和密码，以及SSL证书配置。

- 在Kafka客户端的“/bin”目录下，通过以下命令查询消费进度。

```
./kafka-consumer-groups.sh --bootstrap-server {connection-address} --offsets --describe --all-groups --command-config ../config/{ssl-user-config.properties}
```

表 9-6 查看消费进度参数说明

| 参数名称                       | 说明                                     |
|----------------------------|----------------------------------------|
| connection-address         | Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。 |
| ssl-user-config.properties | 配置文件的名称，此配置文件用于存放用户名和密码，以及SSL证书配置信息。   |

示例如下：

```
[root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --offsets --describe --all-groups --
command-config ../config/ssl-user-config.properties

Consumer group '_consumer-group-dial-test' has no active members.

GROUP          TOPIC          PARTITION  CURRENT-OFFSET  LOG-END-OFFSET
LAG          CONSUMER-ID    HOST        CLIENT-ID
_consumer-group-dial-test _dms_dial_test 0           347             347           0
-            -              -           -               -             -
_consumer-group-dial-test _dms_dial_test 1           347             347           0
-            -              -           -               -             -
_consumer-group-dial-test _dms_dial_test 2           347             347           0
-            -              -           -               -             -

Consumer group 'test' has no active members.

GROUP          TOPIC          PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG
CONSUMER-ID    HOST          CLIENT-ID
test           topic-01      0           5               0               -
test           topic-01      1           3               0               -
test           topic-01      2           10              0               -
[root@ecs-kafka bin]#
```

## 重置消费进度

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > 消费组管理”，进入消费组列表页面。

**步骤6** 单击待重置消费进度的消费组名称，进入消费组详情页。

**步骤7** 在“消费进度”页签，通过以下方法，重置消费进度。

- 重置单个Topic所有分区的消费进度：在待重置消费进度的Topic后，单击“重置消费进度”。
- 重置单个Topic中单个分区的消费进度：在待重置消费进度的Topic分区后，单击“重置消费进度”。
- 重置所有Topic所有分区的消费进度：单击“重置所有消费进度”。

**步骤8** 在弹出的“重置消费进度”对话框中，参考表9-7，设置重置消费进度参数。

**表 9-7** 重置消费进度参数说明

| 参数   | 说明                                                                                                                                                              |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 重置类型 | 选择重置类型： <ul style="list-style-type: none"> <li>• 时间：重置消费进度到指定的时间。</li> <li>• 偏移量：重置消费进度到指定的偏移量。</li> </ul> “重置所有消费进度”只支持重置消费进度到指定时间。                            |
| 时间   | 当“重置类型”为“时间”时，需要设置此参数。<br>选择重置消费进度的时间点，重置完成后，将从此时间点开始消费。 <ul style="list-style-type: none"> <li>• 最早：最早偏移量</li> <li>• 自定义：自定义时间点</li> <li>• 最晚：最晚偏移量</li> </ul> |
| 偏移量  | 当“重置类型”为“偏移量”时，需要设置此参数。<br>选择重置消费进度的偏移量，此偏移量不能小于0，重置完成后，将从此偏移量开始消费。                                                                                             |

**步骤9** 单击“确定”，弹出确认对话框。

**步骤10** 单击“是”，完成消费进度的重置。

在“消费进度”页签，在已重置消费进度的Topic前单击 ，在“偏移量”列查看重置后的值。

---结束

## 9.5 查看 Kafka 重平衡日志

Rebalance是重新分配消费组中的消费者和Topic分区之间订阅关系的过程。在Rebalance过程中消费组下的所有消费者都会停止消费，等待Rebalance完成。

可能会触发Rebalance的原因：

- 消费组成员个数发生变化，例如有新的消费者加入，或者有消费者退出。
- 消费组订阅的Topic个数发生变化。
- 消费组订阅的Topic分区数发生变化。

重平衡日志记录Rebalance的详情，包括Rebalance时间、原因和触发Rebalance的客户端等。本章节介绍如何在控制台查看重平衡日志。

重平衡日志存储在云日志服务（Log Tank Service，简称LTS）中，由LTS提供查询功能。

## 约束与限制

- 重平衡日志默认存储时间为七天，如果需要延长保留天数，可以在[LTS控制台修改日志组的存储时间](#)。
- 重平衡日志开启后会在LTS控制台创建对应的日志组、日志流和仪表盘。
- 不同Kafka实例使用相同的日志组和日志流保存重平衡日志时，在每个Kafka实例中都能看到使用同一个日志组和日志流的所有Kafka实例的重平衡日志。

## 前提条件

- 开启重平衡日志需要您的账号拥有LTS服务的创建日志组和日志流的权限。
- Kafka实例处于“运行中”时，才能开启/关闭重平衡日志。

## 开启重平衡日志

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“分析与诊断 > 重平衡日志”，进入“重平衡日志”页面。

**步骤6** 单击“开启日志”，弹出“开启日志”对话框。

**步骤7** 单击“确定”，弹出“日志配置”对话框。

**步骤8** 根据实际情况决定是否开启日志配置，单击“确定”，自动跳转到“后台任务管理”页面，“状态”为“成功”时，表示开启重平衡日志成功。

- 不开启：LTS后台会自动创建一个日志组以及对应的日志流。
- 开启：选择存储日志文件“coordinator.log”的日志组和日志流。您还可以单击右侧的“查看日志组”，跳转到LTS控制台，查看或创建日志组和日志流。

----结束

## 查看重平衡日志

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“分析与诊断 > 重平衡日志”，进入“重平衡日志”页面。

**步骤6** 在“仪表盘”页签，查看消费组重平衡次数和原因。在“日志查询”页签，查看重平衡日志。

如果您需要搜索日志，请参考[进入搜索LTS日志页面](#)进行操作。

重平衡日志示例如下：

```
{
  "level":"INFO",
  "timestamp":"2023-03-23 17:23:22,906",
  "message":{
    "leaderId":"consumer-1-177817b6-1f29-4717-8a83-dda8eaab1635",
    "generationId":"1",
    "reason":"Assignment received from leader for group KMOffsetCache-dms-vm-fa3cf9d6-manager-shared-server-0 for generation 1",
    "groupId":"KMOffsetCache-dms-vm-fa3cf9d6-manager-shared-server-0",
    "coordinatorId":"0",
    "type":"END_REBALANCE",
    "group":"GroupMetadata(groupId=KMOffsetCache-dms-vm-fa3cf9d6-manager-shared-server-0, generation=1, protocolType=Some(consumer), currentState=CompletingRebalance, members=Map(consumer-1-177817b6-1f29-4717-8a83-dda8eaab1635 -&gt; MemberMetadata(memberId=consumer-1-177817b6-1f29-4717-8a83-dda8eaab1635, clientId=consumer-1, clientHost=/172.31.2.168, sessionTimeoutMs=10000, rebalanceTimeoutMs=300000, supportedProtocols=List(range, )))"
  }
}
```

参数说明如表9-8所示。

表 9-8 重平衡参数说明

| 参数名称          | 说明                                                                                                                                                                                                                                                                                                                                                       |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| level         | 重平衡日志的等级，只有“INFO”一种等级。                                                                                                                                                                                                                                                                                                                                   |
| timestamp     | Rebalance时间。                                                                                                                                                                                                                                                                                                                                             |
| leaderId      | 消费者Leader ID。                                                                                                                                                                                                                                                                                                                                            |
| generationId  | 消费组Generation ID。Generation等同于消费组执行Rebalance的次数，每次Rebalance完成后，Generation都会增加1。                                                                                                                                                                                                                                                                          |
| reason        | 触发Rebalance的原因。                                                                                                                                                                                                                                                                                                                                          |
| groupId       | 消费组ID。                                                                                                                                                                                                                                                                                                                                                   |
| coordinatorId | Coordinator组件所在的Broker。                                                                                                                                                                                                                                                                                                                                  |
| type          | 触发Rebalance的操作，取值如下： <ul style="list-style-type: none"> <li>JOIN_GROUP：新消费者加入消费组。</li> <li>OVER_CAPACITY：超过组限制。</li> <li>UPDATE_MEMBER：消费者更新元数据。</li> <li>PROTOCOL_CHANGE：协议变化。</li> <li>HEARTBEAT_EXPIRED：消费者心跳超时。</li> <li>SYNC_GROUP：同步分配计划。</li> <li>END_REBALANCE：重平衡结束。</li> <li>LEAVE_GROUP：消费者主动离开消费组。</li> <li>DELETE_GROUP：用户删除消费组。</li> </ul> |
| group         | 消费组中消费者的信息。                                                                                                                                                                                                                                                                                                                                              |

----结束

## 关闭重平衡日志

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“分析与诊断 > 重平衡日志”，进入“重平衡日志”页面。

**步骤6** 在页面右上角单击“关闭日志”，弹出确认关闭对话框。

**步骤7** 单击“确定”，自动跳转到“后台任务管理”页面，当重平衡日志任务的“状态”为“成功”时，表示关闭重平衡日志成功。

关闭重平衡日志只是停止重平衡日志上报功能，LTS控制台的日志组和日志流仍然保留，还会继续收费，如不需要保留该日志，可以在LTS控制台删除对应的[日志组](#)和[日志流](#)。

----结束

## 9.6 导出 Kafka 消费组列表

消费组创建成功后，可导出Kafka实例中消费组的列表信息。

### 导出 Kafka 消费组列表

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > 消费组管理”，进入消费组列表页面。

**步骤6** 通过以下任意一种方法，导出消费组列表。

- 勾选待导出的消费组，单击“导出 > 导出已选中数据到XLSX”，导出指定消费组列表。
- 单击“导出 > 导出全部数据到XLSX”，导出全部消费组列表。

----结束

## 9.7 删除 Kafka 消费组

Kafka实例支持通过以下两种方式删除消费组，您可以根据实际情况选择任意一种方式。

- 在控制台删除消费组
- 在Kafka客户端使用命令行工具删除消费组（确保Kafka实例版本与命令行工具版本相同）

### 约束与限制

- 若“auto.create.groups.enable”设置为“true”，当消费组的状态为“EMPTY”且从未提交过offset，系统将在十分钟后自动删除该消费组。
- 若“auto.create.groups.enable”设置为“false”，系统不会自动删除消费组。如果需要删除消费组，需要您手动删除。
- 若消费组从未提交过offset，当Kafka实例重启后，该消费组会被删除。
- 删除消费组会导致消费进度丢失，造成重新消费或重复消费等情况。

### 前提条件

待删除消费组的状态为“EMPTY”。

### 删除消费组（控制台）

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > 消费组管理”，进入消费组列表页面。

**步骤6** 通过以下任意一种方法，删除消费组。

- 勾选消费组名称左侧的方框，可选一个或多个，单击信息栏左上侧的“删除消费组”。
- 在待删除消费组所在行，单击“删除”。
- 单击待删除消费组名称，进入消费组详情页。在页面右上角单击“删除”。

**步骤7** 弹出“删除消费组”对话框，单击“确定”，完成消费组的删除。

已删除的消费组不再显示在消费组列表中，表明消费组已成功删除。

----结束

### 删除消费组（Kafka 客户端）

以下操作命令以Linux系统为例进行说明：

- 未开启密文接入的Kafka实例，在Kafka客户端的“/bin”目录下，通过以下命令删除消费组。  
`./kafka-consumer-groups.sh --bootstrap-server {connection-address} --delete --group {group-name}`

表 9-9 删除消费组参数说明

| 参数名称               | 说明                                     |
|--------------------|----------------------------------------|
| connection-address | Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。 |
| group-name         | 消费组名称。                                 |

示例如下：

```
[root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
192.168.xx.xx:9092,192.168.xx.xx:9092,192.168.xx.xx:9092 --delete --group group-01
Deletion of requested consumer groups ('group-01') was successful.
[root@ecs-kafka bin]#
```

- 已开启密文接入的Kafka实例，通过以下步骤删除消费组。
  - （可选）如果已经设置了SSL证书配置，请跳过此步骤，执行**b**。否则请执行以下操作。  
在Kafka客户端的“/config”目录中创建“ssl-user-config.properties”文件，参考[步骤3](#)增加SSL证书配置。
  - 在Kafka客户端的“/bin”目录下，通过以下命令删除消费组。  
`./kafka-consumer-groups.sh --bootstrap-server {connection-address} --delete --group {group-name} --command-config ../config/{ssl-user-config.properties}`

表 9-10 删除消费组参数说明

| 参数名称                       | 说明                                     |
|----------------------------|----------------------------------------|
| connection-address         | Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。 |
| group-name                 | 消费组名称。                                 |
| ssl-user-config.properties | 配置文件的名称，此配置文件用于存放用户名和密码，以及SSL证书配置信息。   |

示例如下：

```
[root@ecs-kafka bin]# ./kafka-consumer-groups.sh --bootstrap-server
192.168.xx.xx:9093,192.168.xx.xx:9093,192.168.xx.xx:9093 --delete --group group-02 --command-
config ../config/ssl-user-config.properties
Deletion of requested consumer groups ('group-02') was successful.
[root@ecs-kafka bin]#
```

## 9.8 取消 Kafka 消费组和 Topic 的订阅关系

消费者订阅Topic消费消息后，在消费组详情页可以查看消费进度。如果消费者不再消费某个Topic，您可以取消消费组和此Topic的订阅关系，避免由于此Topic的消息堆积，导致的告警误报。

### 约束与限制

- 已创建的1.1.0/2.3.0版本的Kafka实例可能不支持取消订阅功能。
- 取消消费组和Topic的订阅关系会删除消费组对该Topic的消费位点信息，删除后无法恢复，请谨慎操作。

### 取消 Kafka 消费组和 Topic 的订阅关系

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > 消费组管理”，进入消费组列表页面。

**步骤6** 单击待取消订阅关系的消费组名称，进入消费组详情页。

**步骤7** 在“消费进度”页签中，勾选待取消订阅关系的Topic，每次最多勾选**50个**，单击信息栏左上侧的“取消订阅”。

**步骤8** 在“取消订阅”对话框中，单击“是”。

取消订阅成功后，在“消费进度”页签，不再显示取消订阅的Topic。

----结束

# 10 配置流控

## 10.1 配置 Kafka 流控

Kafka支持对用户/客户端/Topic配置流量控制，控制生产/消费消息的上限速率。  
用户/客户端的流控作用范围是整个Broker，Topic的流控作用范围是指定Topic。

### 操作影响

- 当流控值达到上限后，会导致生产/消费的时延增大。
- 设置的流控值较小且生产者速率较大时，可能会造成生产超时、消息丢失，导致部分消息生产失败。
- 初始生产/消费的流量较大，如果设置一个较小的流控值，会导致生产/消费的时延增大、部分消息生产失败。建议逐次减半设置流控值，待生产/消费稳定后继续减半设置，直到设置为目标流控值。例如初始生产流量100MB/s，可先设置生产流控为50MB/s，待稳定后再修改为25MB/s，直到目标流控值。

### 前提条件

- 如果需要对用户进行流量控制，请在创建Kafka实例时，开启SASL\_SSL功能。然后在控制台的“实例管理 > 用户管理”页面，获取用户名。
- 如果需要对指定客户端进行流量控制，请在客户端配置中获取client ID。
- 如果需要对指定Topic进行流量控制，请在控制台的“实例管理 > Topic管理”页面，获取Topic名称。

### 创建流控

以下介绍创建用户/客户端/Topic流控的步骤。

#### 创建用户/客户端流控

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

- 步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入 Kafka实例列表页面。
- 步骤4** 单击Kafka实例的名称，进入实例详情页面。
- 步骤5** 在左侧导航栏单击“实例管理 > 流控管理”，进入流控列表页面。
- 步骤6** 在页面顶端单击“用户/客户端”，进入“用户/客户端”页签。
- 步骤7** 在页面左上角单击“创建流控”，弹出“创建流控”对话框。
- 步骤8** 设置流控参数。

表 10-1 流控参数说明

| 参数名称   | 说明                                                                                           |
|--------|----------------------------------------------------------------------------------------------|
| 用户名    | 输入 <b>前提条件</b> 中获取的用户名，对此用户进行流控。如果需要对所有用户进行流控，在“用户名”后，单击“选择默认”。<br>流控创建完后，无法修改“用户名”。         |
| 客户端ID  | 输入 <b>前提条件</b> 中获取的客户端ID，对此客户端进行流控。如果需要对所有客户端进行流控，在“客户端ID”后，单击“选择默认”。<br>流控创建完后，无法修改“客户端ID”。 |
| 生产上限速率 | 设置生产上限速率，单位为MB/s。为空时，表示不设置速率。                                                                |
| 消费上限速率 | 设置消费上限速率，单位为MB/s。为空时，表示不设置速率。                                                                |

 **注意**

- 未开启密文接入的实例，在“创建流控”对话框中，不显示“用户名”。
- “用户名”和“客户端ID”不可同时为空。
- “生产上限速率”和“消费上限速率”不可同时为空。

- 步骤9** 单击“确定”，跳转到“后台任务管理”页面，当流控任务的“状态”为“成功”时，表示流控创建成功。

进入“实例管理 > 流控管理”页面，在“用户/客户端”页签中，通过以下方法查看新创建的流控。

- 未开启密文接入的实例：在搜索框中输入新创建的流控名称，按“Enter”。
- 已开启密文接入的实例：单击页面左上角的“仅设置了用户名”/“仅设置了客户端ID”/“设置了用户名和客户端ID”，选择新创建的流控所属的分类，然后在搜索框中输入新创建的流控名称，按“Enter”

图 10-1 查看新创建的流控



----结束

## 创建 Topic 流控

- 步骤1 登录管理控制台。
- 步骤2 在管理控制台左上角单击 ，选择Kafka实例所在的区域。
- 步骤3 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。
- 步骤4 单击Kafka实例的名称，进入实例详情页面。
- 步骤5 在左侧导航栏单击“实例管理 > 流控管理”，进入流控列表页面。
- 步骤6 在页面顶端单击“Topic”，进入“Topic”页签。
- 步骤7 在页面左上角单击“创建流控”，弹出“创建流控”对话框。
- 步骤8 设置流控参数。

表 10-2 流控参数说明

| 参数名称    | 说明                                                |
|---------|---------------------------------------------------|
| Topic名称 | 输入指定Topic名称，对此Topic进行流控。<br>流控创建完后，无法修改“Topic名称”。 |
| 生产上限速率  | 设置生产上限速率，单位为MB/s。为空时，表示不设置速率。                     |
| 消费上限速率  | 设置消费上限速率，单位为MB/s。为空时，表示不设置速率。                     |

### 注意

“生产上限速率”和“消费上限速率”不可同时为空。

- 步骤9 单击“确定”，跳转到“后台任务管理”页面，当流控任务的“状态”为“成功”时，表示流控创建成功。

进入“实例管理 > 流控管理”页面，在“Topic”页签的搜索框中输入新创建的流控名称，单击“Enter”，查看新创建的流控。

----结束

## 编辑流控

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“实例管理 > 流控管理”，进入流控列表页面。

**步骤6** 在待修改的流控所在行，单击“编辑”，弹出“编辑流控”对话框。

**步骤7** 修改生产上限速率或者消费上限速率，单击“确定”，跳转到“后台任务管理”页面，当流控任务的“状态”为“成功”时，表示流控修改成功。

进入“实例管理 > 流控管理”页面，查看修改后流控的生产上限速率或者消费上限速率。

### 说明

“生产上限速率”和“消费上限速率”不可同时为空。

----结束

## 导出流控列表

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“实例管理 > 流控管理”，进入流控列表页面。

**步骤6** 导出流控列表。

- 指定用户/客户端流控：在“用户/客户端”页签，勾选待导出的用户/客户端流控，单击“导出 > 导出已选中数据到XLSX”。
- 全部用户/客户端流控：在“用户/客户端”页签，单击“导出 > 导出全部数据到XLSX”。
- 指定Topic流控：在“Topic”页签，勾选待导出的Topic流控，单击“导出 > 导出已选中数据到XLSX”。
- 全部Topic流控：在“Topic”页签，单击“导出 > 导出全部数据到XLSX”。

----结束

## 删除流控

- 步骤1** 登录管理控制台。
- 步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。
- 步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。
- 步骤4** 单击Kafka实例的名称，进入实例详情页面。
- 步骤5** 在左侧导航栏单击“实例管理 > 流控管理”，进入流控列表页面。
- 步骤6** 在待删除的流控所在行，单击“删除”，弹出“删除流控”对话框。
- 步骤7** 单击“确定”，跳转到“后台任务管理”页面，当流控任务的“状态”为“成功”时，表示成功删除流控。

----结束

## 10.2 查看 Kafka 流控统计

如果Kafka配置了流量控制，可在控制台查看当前流控策略下，每个代理上用户/客户端/Topic的带宽使用量。

### 查看带宽使用量

- 步骤1** 登录管理控制台。
- 步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。
- 步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。
- 步骤4** 单击Kafka实例的名称，进入实例详情页面。
- 步骤5** 在左侧导航栏单击“监控 > 流控监控”，进入流控监控页面。
- 步骤6** 设置查询带宽使用量的参数。

图 10-2 查询带宽使用量的参数

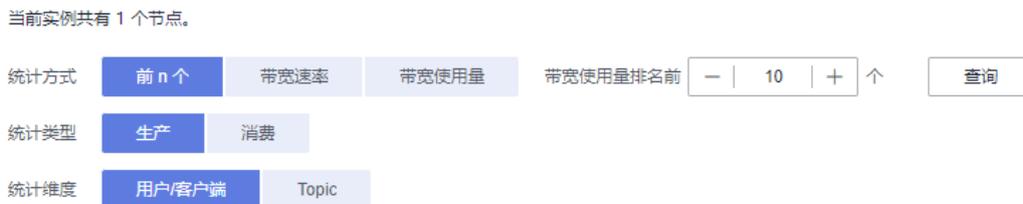


表 10-3 查询带宽使用量的参数说明

| 参数   | 说明                                                                                                                                                                                                     |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 统计方式 | 设置带宽使用量的统计方式。 <ul style="list-style-type: none"><li>前n个：统计带宽使用量排名前x个的用户/客户端/Topic，x由您自行输入。</li><li>带宽速率：统计带宽速率大于x MB/s的用户/客户端/Topic，x由您自行输入。</li><li>带宽使用量：统计带宽使用量超过x%的用户/客户端/Topic，x由您自行输入。</li></ul> |
| 统计类型 | 设置带宽使用量的统计类型。 <ul style="list-style-type: none"><li>生产：统计生产消息的带宽使用量。</li><li>消费：统计消费消息的带宽使用量。</li></ul>                                                                                                |
| 统计维度 | 设置带宽使用量的统计维度。 <ul style="list-style-type: none"><li>用户/客户端：统计用户/客户端的带宽使用量。</li><li>Topic：统计Topic的带宽使用量。</li></ul>                                                                                      |

**步骤7** 单击“查询”，查看每个代理上用户/客户端/Topic的带宽使用量。

----结束

# 11 管理实例

## 11.1 查看和修改 Kafka 实例基本信息

创建Kafka实例成功后，您可以在控制台查看Kafka实例的详细信息，并根据自己的业务情况对Kafka实例的部分参数进行调整，包括实例名称、描述、安全组和容量阈值策略等。

### 前提条件

Kafka实例的状态为“运行中”时，才能修改实例的基本信息。

### 查看 Kafka 实例的详细信息

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** Kafka实例支持通过筛选功能查询目标实例。当前支持的筛选条件为名称、状态、版本、实例类型、规格、已用/可用存储空间、分区上限、计费模式、可用区、实例ID、内网连接地址、公网连接地址、企业项目和资源标签。

- 企业用户才能使用企业项目进行筛选。
- Kafka实例状态请参见[表11-1](#)。

表 11-1 Kafka 实例状态说明

| 状态   | 说明                               |
|------|----------------------------------|
| 创建中  | 创建Kafka实例后，在Kafka实例状态进入运行中之前的状态。 |
| 创建失败 | Kafka实例处于创建失败状态。                 |

| 状态   | 说明                                                              |
|------|-----------------------------------------------------------------|
| 运行中  | Kafka实例正常运行状态。<br>在这个状态的实例可以运行您的业务。                             |
| 故障   | Kafka实例处于故障状态。                                                  |
| 重启中  | Kafka实例正在进行重启操作。                                                |
| 变更中  | Kafka实例正在进行规格变更操作。                                              |
| 变更失败 | Kafka实例处于规格变更操作失败的状态。<br>实例处于变更失败状态时，您无法重启实例、删除实例和修改实例，请联系客服处理。 |
| 回收中  | Kafka实例正在进行临时删除操作。                                              |
| 回收站中 | Kafka实例处于回收站中。                                                  |
| 恢复中  | Kafka实例正在从回收站中恢复。                                               |

**步骤5** 单击Kafka实例的名称，进入该Kafka实例的“概览”页面，查看Kafka实例的详细信息。

**表11-2**为连接实例的相关参数，其他参数，请查看页面显示。

**表 11-2** 连接参数说明

| 信息类型 | 参数             | 说明                                                                                                     |
|------|----------------|--------------------------------------------------------------------------------------------------------|
| 连接信息 | 用户名            | 实例开启SASL_SSL后，连接实例的用户名。                                                                                |
|      | Kafka SASL_SSL | 是否开启SASL_SSL开关。                                                                                        |
|      | 启用的安全协议        | 密文接入开启后，才显示此参数。                                                                                        |
|      | 开启的SASL认证机制    | 开启SASL_SSL后，实例已开启的SASL认证机制。                                                                            |
|      | SSL证书          | 单击“下载”，下载连接实例的SSL证书。                                                                                   |
|      | 内网连接地址         | 连接实例的地址。<br>显示的连接地址个数为实例代理个数。                                                                          |
|      | Kafka Manager  | 是否关闭Kafka Manager。<br>关闭Kafka Manager后，控制台不显示Kafka Manager连接地址，不支持重置Kafka Manager密码和重启Kafka Manager功能。 |
|      | Manager用户名     | 连接Kafka Manager的用户名。                                                                                   |
|      | Manager内网访问地址  | 连接Kafka Manager的地址。                                                                                    |

----结束

## 修改 Kafka 实例的基本信息

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 以下参数支持修改。

表 11-3 Kafka 实例支持修改的参数

| 参数名称   | 修改方法                                                                                                                                                                                                                                             | 查看修改结果                           |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|
| 实例名称   | 单击  ，输入修改后的实例名称，单击  。<br>实例名称需要符合命名规则：长度为4~64个字符，由英文字母开头，只能由英文字母、数字、中划线、下划线组成。 | 页面上方直接提示修改结果。                    |
| 描述     | 单击  ，输入修改后的描述，单击  。<br>长度为0~1024个字符。                                       | 页面上方直接提示修改结果。                    |
| 企业项目   | 单击  ，在下拉框中选择新的企业项目，单击  。<br>该参数针对企业用户使用。修改企业项目，不会重启实例。                     | 页面上方直接提示修改结果。                    |
| 安全组    | 单击  ，在下拉框中选择新的安全组，单击  。<br>修改安全组，不会重启实例。                                   | 页面上方直接提示修改结果。                    |
| 容量阈值策略 | 单击待选择的容量阈值策略，弹出“确认”对话框，单击“确定”。<br>修改容量阈值策略，不会重启实例。<br>磁盘使用量达到容量阈值时会触发对应的阈值处理策略，可能会导致消息被删除或无法继续生产消息。                                                                                                                                              | 系统跳转到“后台任务管理”页面，并显示当前任务的操作进度和结果。 |

| 参数名称           | 修改方法                                                                                | 查看修改结果                           |
|----------------|-------------------------------------------------------------------------------------|----------------------------------|
| Kafka自动创建Topic | 开启/关闭Kafka自动创建Topic，弹出“确认”对话框，单击“确定”。<br>开启/关闭“Kafka自动创建Topic”可能会导致实例重启，具体以控制台显示为准。 | 系统跳转到“后台任务管理”页面，并显示当前任务的操作进度和结果。 |
| 跨VPC访问         | 修改方法请参考 <a href="#">使用VPCEP实现跨VPC访问Kafka</a> 。                                      | 页面上方直接提示修改结果。                    |

----结束

## 11.2 查看 Kafka 磁盘使用量

本章节指导您在控制台查看每个代理上磁盘的使用量，以了解Kafka实例的磁盘使用情况。

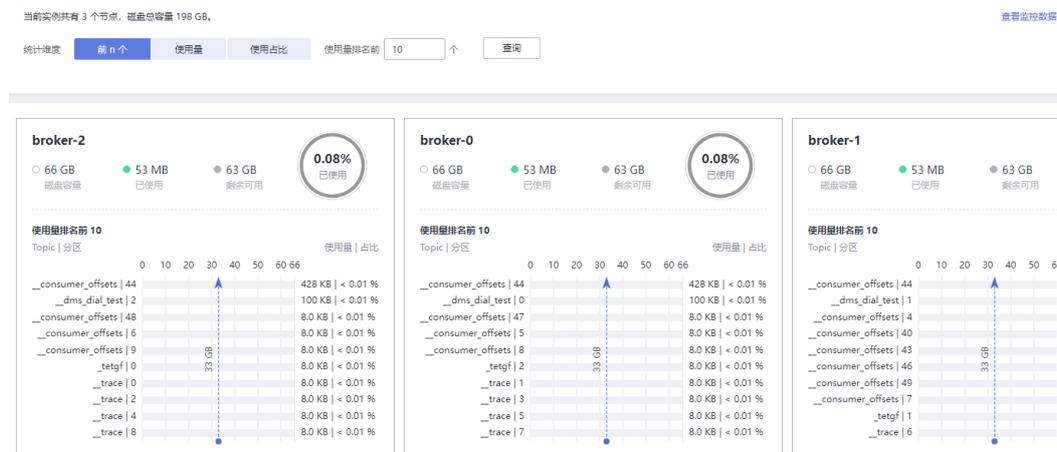
### 查看 Kafka 磁盘使用量

- 步骤1** 登录管理控制台。
- 步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。
- 步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。
- 步骤4** 单击Kafka实例的名称，进入实例的“概览”页面。
- 步骤5** 在“监控 > 磁盘存储统计”页面，查看磁盘的使用量。

页面支持从以下三个方面对代理中的Topic磁盘使用量进行查询：使用量排名前xx个、使用量达到xx GB和使用占比达xx%。

在页面右上角，单击“查看监控数据”，跳转到云监控服务的监控指标页面。在监控指标页面，您可以查看Kafka实例的各项监控指标。

图 11-1 查看磁盘的使用量



----结束

## 11.3 查看 Kafka 实例后台任务

对实例的一些操作，如表11-4所示，会启动一个后台任务，您可以在后台任务页，查看该操作的状态等信息，同时可通过删除操作，清理任务信息。

表 11-4 后台任务列表

| 任务名         | 说明                                                                                                       |
|-------------|----------------------------------------------------------------------------------------------------------|
| 创建实例        | 创建一个Kafka实例。                                                                                             |
| 重启实例        | 重启Kafka实例。                                                                                               |
| Kafka配置参数修改 | <ul style="list-style-type: none"> <li>修改Kafka的配置参数。</li> <li>开启/关闭自动创建Topic。</li> </ul>                 |
| 容量阈值策略变更    | 修改Kafka实例的容量阈值策略。                                                                                        |
| 变更规格        | <ul style="list-style-type: none"> <li>扩容存储空间</li> <li>扩容代理数量</li> <li>扩容基准带宽</li> <li>扩容代理规格</li> </ul> |
| 创建流控        | 创建用户/客户端/Topic流控。                                                                                        |
| 编辑流控        | 修改流控参数。                                                                                                  |
| 删除流控        | 删除用户/客户端/Topic流控。                                                                                        |
| Kafka分区平衡   | 修改Topic的分区平衡。                                                                                            |
| 开启日志        | 开启Kafka重平衡日志/Topic日志。                                                                                    |
| 关闭日志        | 关闭Kafka重平衡日志/Topic日志。                                                                                    |

| 任务名       | 说明                    |
|-----------|-----------------------|
| 设置Topic权限 | 在Topic中为不同的用户设置不同的权限。 |

## 查看 Kafka 实例后台任务

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例的“概览”页面。

**步骤5** 在左侧导航栏单击“实例管理 > 后台任务管理”，进入后台任务管理页面。

**步骤6** 在“即时任务”/“定时任务”页签，单击时间段下拉框，选择搜索时间，在搜索框中输入关键字，按“Enter”，界面显示相应时间段内启动的任务。

在后台任务管理页面，您还可以执行以下操作：

- 单击 ，刷新任务状态。
- 单击“删除”，弹出“删除任务记录”对话框，单击“确定”，清理任务信息。  
只有在任务已经执行完成，即任务状态为成功或者失败，或者定时任务已经取消，即任务状态为取消，才能执行删除操作。

----结束

## 11.4 查看 Kafka 生产消费示例代码

分布式消息服务Kafka版支持在控制台查看Java、Go和Python语言的生产消费消息示例代码，您可以参考示例代码快速完成Kafka客户端的集成，并进行消息的生产和消费。

### 查看 Kafka 生产消费示例代码

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在左侧导航栏选择“实例管理 > Topic管理”，进入Topic列表页面。

**步骤6** 单击“查看示例代码”，弹出“生产消费示例代码”对话框。

示例代码根据语言分为：Java、Go和Python。根据接入类型分为：PlainText、SASL\_SSL和SASL\_PLAINTEXT。

- PlainText：客户端通过明文连接Kafka实例。
- SASL\_SSL：客户端通过密文连接Kafka实例，且连接时需要进行SASL认证。SASL认证分为PLAIN和SCRAM-SHA-512。
- SASL\_PLAINTEXT：客户端通过明文连接Kafka实例，且连接时需要进行SASL认证。SASL认证分为PLAIN和SCRAM-SHA-512。

----结束

## 11.5 修改 Kafka 实例配置参数

分布式消息服务Kafka版为实例、Topic、消费者提供了几个常用配置参数的默认值，您可以根据实际业务需求，在控制台自行修改参数值。其他在控制台未列出的配置参数，请参考[Kafka配置](#)进行修改。

Kafka实例的参数分为动态参数和静态参数：

- 动态参数：动态参数修改成功后，无需重启实例，立即生效。
- 静态参数：静态参数修改成功后，需要手动重启实例才能生效。

### 约束与限制

部分老实例不支持修改配置参数，具体以控制台为准，请联系客服解决。

### 前提条件

Kafka实例的状态为“运行中”时，才能修改配置参数。

### 修改 Kafka 实例配置参数

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击Kafka实例的名称，进入实例详情页面。

**步骤5** 在“实例管理 > 配置参数”页面，修改配置参数。

- 修改指定参数：在待修改参数所在行，单击“编辑”。
- 批量修改参数：单击“批量编辑”。
- 恢复参数默认值：单击“编辑”/“批量编辑”，然后单击“恢复默认”。

1.1.0版本实例的参数说明如[表11-5](#)和[表11-6](#)所示，2.3.0/2.7/3.x版本实例的参数说明如[表11-7](#)和[表11-8](#)所示。

表 11-5 动态参数说明（1.1.0 版本）

| 参数                        | 参数说明                                                                                                                       | 参数范围                  | 默认值   |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------|-----------------------|-------|
| auto.create.groups.enable | 是否开启自动创建消费组功能。<br>部分老实例在控制台不显示此参数，实例默认开启自动创建消费组功能，无法在控制台关闭此功能。                                                             | true/false            | true  |
| offsets.retention.minutes | 消费位点保留时间是消费位点最大保留时间，从提交消费位点的时间开始计算，超过该时间的消费位点将被删除。消费者每次向Topic分区提交一次消费位点时，该消费位点的保留时间将会被重置为0。<br>此参数在部分老实例中显示为静态参数，具体以控制台为准。 | 1440 ~ 30240<br>单位：分钟 | 20160 |

表 11-6 静态参数说明（1.1.0 版本）

| 参数                             | 参数说明                                                                                | 参数范围                   | 默认值      |
|--------------------------------|-------------------------------------------------------------------------------------|------------------------|----------|
| min.insync.replicas            | 当producer将acks设置为“all”（或“-1”）时，此配置指定必须确认写入才能被认为成功的副本的最小数量。                          | 1 ~ 3                  | 1        |
| message.max.bytes              | 单条消息的最大长度。                                                                          | 0 ~ 10485760<br>单位：字节  | 10485760 |
| unclean.leader.election.enable | 指示是否启用不在ISR集合中的副本选为领导者作为最后的手段，即使这样做可能导致数据丢失。                                        | true/false             | false    |
| connections.max.idle.ms        | 此参数用来指定在多少毫秒之后，关闭空闲的连接。                                                             | 5000 ~ 600000<br>单位：毫秒 | 600000   |
| log.retention.hours            | 实例级消息老化时间（小时为单位）。<br>如果Topic已经设置了老化时间，则此参数对此Topic不生效。仅在Topic未设置老化时间时，此参数才对此Topic生效。 | 1 ~ 168<br>单位：小时       | 72       |
| max.connections.per.ip         | 每个IP允许的最大连接数。超过此连接数的连接请求将被丢弃。                                                       | 100 ~ 20000            | 1000     |

| 参数                             | 参数说明                                                                                                                                                                                                                                      | 参数范围                    | 默认值     |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|---------|
| group.max.session.timeout.ms   | consumer允许的最大会话超时时间。超时时间越长，consumer就能在心跳探测周期内有更多时间处理消息，但也会使故障检测花费更长时间。                                                                                                                                                                    | 6000 ~ 1800000<br>单位：毫秒 | 1800000 |
| default.replication.factor     | 自动创建Topic时的默认副本个数。                                                                                                                                                                                                                        | 1 ~ 3                   | 3       |
| allow.everyone.if.no.acl.found | <ul style="list-style-type: none"> <li>如果设置为true，没有给资源配置ACL时，允许所有用户访问。</li> <li>如果设置为false，初始用户拥有所有操作的权限，其他用户需要配置权限才能进行对应的操作。所有操作包括修改Topic权限、创建和删除Topic、以及修改Topic分区数。</li> </ul> <p>仅在实例开启SASL时，存在此参数。</p> <p>部分老实例不支持修改此参数，具体以控制台为准。</p> | true/false              | true    |
| num.partitions                 | 自动创建Topic时的默认分区数。                                                                                                                                                                                                                         | 1 ~ 200                 | 3       |
| group.min.session.timeout.ms   | consumer允许的最小会话超时时间。超时时间越短，consumer的心跳探测越频繁，可以使故障检测更快，但会导致broker被抢占更多的资源。                                                                                                                                                                 | 6000 ~ 300000<br>单位：毫秒  | 6000    |

表 11-7 动态参数说明（2.3.0/2.7/3.x 版本）

| 参数                        | 参数说明                                                           | 参数范围                  | 默认值      |
|---------------------------|----------------------------------------------------------------|-----------------------|----------|
| min.insync.replicas       | 当producer将acks设置为“all”（或“-1”）时，此配置指定必须确认写入才能被认为成功的副本的最小数量。     | 1 ~ 3                 | 1        |
| message.max.bytes         | 单条消息的最大长度。                                                     | 0 ~ 10485760<br>单位：字节 | 10485760 |
| auto.create.groups.enable | 是否开启自动创建消费组功能。<br>部分老实例在控制台不显示此参数，实例默认开启自动创建消费组功能，无法在控制台关闭此功能。 | true/false            | true     |

| 参数                                 | 参数说明                                                                                                                           | 参数范围                             | 默认值   |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|----------------------------------|-------|
| max.connections.p<br>er.ip         | 每个IP允许的最大连接数。超过此连接数的连接请求将被丢弃。                                                                                                  | 100 ~<br>20000                   | 1000  |
| unclean.leader.ele<br>ction.enable | 指示是否启用不在ISR集合中的副本选为领导者作为最后的手段，即使这样做可能导致数据丢失。                                                                                   | true/false                       | false |
| offsets.retention.<br>minutes      | 消费位点保留时间是消费位点最大保留时间，从提交消费位点的时间开始计算，超过该时间的消费位点将被删除。消费者每次向Topic分区提交一次消费位点时，该消费位点的保留时间将会被重置为0。<br><br>此参数在部分老实例中显示为静态参数，具体以控制台为准。 | 1440 ~<br>30240<br><br>单位：分<br>钟 | 20160 |

表 11-8 静态参数说明（2.3.0/2.7/3.x 版本）

| 参数                               | 参数说明                                                                                                   | 参数范围                               | 默认值     |
|----------------------------------|--------------------------------------------------------------------------------------------------------|------------------------------------|---------|
| connections.max.i<br>dle.ms      | 此参数用来指定在多少毫秒之后，关闭空闲的连接。                                                                                | 5000 ~<br>600000<br><br>单位：毫<br>秒  | 600000  |
| log.retention.hour<br>s          | 实例级消息老化时间（小时为单<br>位）<br><br>如果Topic已经设置了老化时间，则<br>此参数对此Topic不生效。仅在Topic<br>未设置老化时间时，此参数才对此<br>Topic生效。 | 1 ~ 168<br><br>单位：小<br>时           | 72      |
| group.max.session<br>.timeout.ms | consumer允许的最大会话超时时间。超时时间越长，consumer就能<br>在心跳探测周期内有更多时间处理<br>消息，但也会使故障检测花费更长<br>时间。                     | 6000 ~<br>1800000<br><br>单位：毫<br>秒 | 1800000 |
| default.replication<br>.factor   | 自动创建Topic时的默认副本个数。                                                                                     | 1 ~ 3                              | 3       |

| 参数                             | 参数说明                                                                                                                                                                                                                                      | 参数范围                   | 默认值  |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|------|
| allow.everyone.if.no.acl.found | <ul style="list-style-type: none"> <li>如果设置为true，没有给资源配置ACL时，允许所有用户访问。</li> <li>如果设置为false，初始用户拥有所有操作的权限，其他用户需要配置权限才能进行对应的操作。所有操作包括修改Topic权限、创建和删除Topic、以及修改Topic分区数。</li> </ul> <p>仅在实例开启密文接入时，存在此参数。</p> <p>部分老实例不支持修改此参数，具体以控制台为准。</p> | true/false             | true |
| num.partitions                 | 自动创建Topic时的默认分区数。                                                                                                                                                                                                                         | 1 ~ 200                | 3    |
| group.min.session.timeout.ms   | consumer允许的最小会话超时时间。超时时间越短，consumer的心跳探测越频繁，可以使故障检测更快，但会导致broker被抢占更多的资源。                                                                                                                                                                 | 6000 ~ 300000<br>单位：毫秒 | 6000 |

**步骤6** 单击“保存”，完成参数的修改。

动态参数修改成功后，无需重启实例，立即生效。**静态参数修改成功后，需要手动重启实例才能生效。**

在“配置参数”页面的“参数运行值”列，查看修改后的参数值。

----结束

## 11.6 配置 Kafka 实例标签

标签是Kafka实例的标识，为Kafka实例添加标签，可以方便用户识别和管理拥有的Kafka实例资源。

如您的组织已经设定分布式消息服务Kafka版的相关标签策略，则需按照标签策略规则为Kafka实例添加标签。新添加的标签不符合标签策略的规则，会添加失败。

标签共由两部分组成：“标签键”和“标签值”，其中，“标签键”和“标签值”的命名规则如表11-9所示。

表 11-9 标签命名规则

| 参数名称 | 规则                                                                                                                                                                                                             |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 标签键  | <ul style="list-style-type: none"> <li>不能为空。</li> <li>对于同一个实例，Key值唯一。</li> <li>长度为1~128个字符（中文也可以输入128个字符）。</li> <li>由任意语种字母、数字、空格和字符组成，字符仅支持 _ . : = + - @</li> <li>首尾字符不能为空格。</li> <li>不能以_sys_开头。</li> </ul> |
| 标签值  | <ul style="list-style-type: none"> <li>长度为0~255个字符（中文也可以输入255个字符）。</li> <li>由任意语种字母、数字、空格和字符组成，字符仅支持 _ . : = + - @</li> <li>创建实例时设置的标签值，首尾字符不能为空格。</li> </ul>                                                  |

## 约束限制

一个Kafka实例最多添加20个标签。

## 配置单个 Kafka 实例标签

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 单击实例名称，进入实例详情页面。

**步骤5** 在左侧导航栏单击“实例管理 > 标签”，进入标签列表页面。

**步骤6** 单击“编辑标签”，弹出“编辑标签”对话框。

**步骤7** 您可以根据实际需要，添加/删除/编辑标签。

表 11-10 标签操作说明

| 操作类型 | 操作步骤                                                                                                                                                                                    |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 添加标签 | <ol style="list-style-type: none"> <li>单击“添加标签”，在“标签键”和“标签值”中，输入标签的键/值。如果您已经预定义了标签，在“标签键”和“标签值”中选择已经定义的标签键/值。最多可以添加<b>20个标签</b>。</li> <li>单击“确定”，成功为实例添加标签。在标签列表页面，查看新添加的标签。</li> </ol> |

| 操作类型 | 操作步骤                                                               |
|------|--------------------------------------------------------------------|
| 编辑标签 | 修改标签键/值，单击“确定”，完成标签的修改。<br>在标签列表页面，查看修改后的标签键/值。                    |
| 删除标签 | 在待删除的标签所在行，单击“删除”。然后单击“确定”，完成标签的删除。<br>已删除的标签不再显示在标签列表中，表明标签已成功删除。 |

----结束

## 11.7 配置 Kafka 实例回收站策略

开启回收站策略的实例在被删除后，会被临时存入回收站中，此时实例中的数据尚未被彻底删除，在保留天数内支持从回收站中恢复此实例。超过保留天数的实例会被彻底删除，无法恢复。

回收站策略默认是关闭状态。

### 约束与限制

- 回收站中的按需实例不会收取实例的费用，但是会收取存储空间的费用。

### 开启回收站策略

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 在左侧导航栏选择“回收站”，进入“回收站”页面。

**步骤5** 单击“回收站策略”，弹出“回收站策略”对话框。

**步骤6** 开启启用回收站，并设置保留天数（只能设置为1~7天），单击“确定”。

----结束

### 恢复 Kafka 实例

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 在左侧导航栏选择“回收站”，进入“回收站”页面。

**步骤5** 通过以下任意一种方法，恢复Kafka实例。

- 勾选Kafka实例名称左侧的方框，可选一个或多个，单击信息栏左上侧的“恢复”。
- 在待恢复Kafka实例所在行，单击“恢复”。

**步骤6** 在弹出的“恢复实例”对话框中，单击“确定”。

恢复实例大约需要3~10分钟，在“Kafka实例”页面查看已恢复的实例。

----结束

## 修改保留天数

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 在左侧导航栏选择“回收站”，进入“回收站”页面。

**步骤5** 单击“回收站策略”，弹出“回收站策略”对话框。

**步骤6** 修改保留天数（只能设置为1~7天），单击“确定”。

修改保留天数仅对修改后删除的实例生效，修改前已在回收站中的实例依然按照修改前的天数保留实例。

----结束

## 导出回收站中的实例

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 在左侧导航栏选择“回收站”，进入“回收站”页面。

**步骤5** 通过以下任意一种方法，导出实例列表。

- 勾选待导出的实例，单击“导出 > 导出已选中数据到XLSX”，导出指定实例列表。
- 单击“导出 > 导出全部数据到XLSX”，导出全部实例列表。

----结束

## 删除回收站中的实例

**步骤1** 登录管理控制台。

- 步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。
- 步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。
- 步骤4** 在左侧导航栏选择“回收站”，进入“回收站”页面。
- 步骤5** 通过以下任何一种方法，删除实例。
- 勾选Kafka实例名称左侧的方框，可选一个或多个，单击信息栏左上侧的“删除”。
  - 在待删除Kafka实例所在行，单击“删除”。
- 步骤6** 在弹出的“删除实例”对话框中，输入“DELETE”，单击“确定”，完成删除Kafka实例。

回收站中的Kafka实例删除后，实例中原有的数据将被删除，且没有备份，请谨慎操作。

----结束

## 关闭回收站策略

- 步骤1** 登录管理控制台。
- 步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。
- 步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。
- 步骤4** 在左侧导航栏选择“回收站”，进入“回收站”页面。
- 步骤5** 单击“回收站策略”，弹出“回收站策略”对话框。
- 步骤6** 关闭启用回收站，单击“确定”。

----结束

## 11.8 导出 Kafka 实例列表

分布式消息服务Kafka版支持在控制台导出实例列表信息，供用户查看和使用。

### 导出 Kafka 实例列表

- 步骤1** 登录管理控制台。
- 步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。
- 步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。
- 步骤4** 通过以下任何一种方法，导出实例列表。
- 勾选待导出的实例，单击“导出 > 导出已选中数据到XLSX”，导出指定实例列表。

- 单击“导出 > 导出全部数据到XLSX”，导出全部实例列表。

----结束

## 11.9 重启 Kafka 实例

分布式消息服务Kafka版支持在控制台重启单个Kafka实例，或批量重启Kafka实例。

### 约束与限制

- 在Kafka实例重启过程中，客户端的生产与消费消息等请求会被拒绝。
- 在客户端配置重试机制，避免因为重启实例导致的业务断连。

### 前提条件

只有当Kafka实例处于“运行中”或“故障”状态，才能执行重启操作。

### 重启 Kafka 实例

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 通过以下任何一种方法，重启Kafka实例。

- 勾选Kafka实例名称左侧的方框，可选一个或多个，单击信息栏左上侧的“重启”。
- 在待重启Kafka实例所在行，单击“更多 > 重启”。
- 单击Kafka实例名称，进入实例详情页面。单击右上角的“... > 重启”。

**步骤5** 在“重启实例”对话框中，单击“确定”，完成重启Kafka实例。

重启Kafka实例大约需要3到15分钟。Kafka实例重启成功后，实例状态切换为“运行中”。

#### 说明

重启Kafka实例只会重启实例进程，不会重启实例所在虚拟机。

----结束

## 11.10 删除 Kafka 实例

分布式消息服务Kafka版管理控制台支持删除Kafka实例，且可实现批量删除Kafka实例。

删除的实例能否恢复取决于是否开启回收站策略，如果未开启回收站策略，实例删除后，实例中原有的数据将被删除，且没有备份。回收站策略默认关闭，如果需要开启，具体步骤请参见[开启回收站策略](#)。

## 前提条件

Kafka实例状态为运行中、故障、创建失败的实例才能执行删除操作。

## 删除 Kafka 实例

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 通过以下任意一种方法，删除按需计费的Kafka实例。

- 勾选Kafka实例名称左侧的方框，可选一个或多个，单击信息栏左上侧的“删除”。
- 在待删除Kafka实例所在行，单击“更多 > 删除”。
- 单击Kafka实例名称，进入实例详情页面。单击右上角的“... > 删除”。

**步骤5** 在“删除实例”对话框中，输入“DELETE”，单击“确定”，完成删除Kafka实例。

删除Kafka实例大约需要1到60秒。

已删除的Kafka实例不再显示在实例列表中，表明Kafka实例已成功删除。

----结束

## 11.11 管理 Kafka Manager

### 11.11.1 访问 Kafka Manager

Kafka Manager是开源的Kafka集群管理工具，需要通过浏览器才能访问Kafka Manager的地址。在Kafka Manager页面，您可以查看Kafka集群的监控、代理等信息。

表 11-11 Kafka Manager 功能对应的 Kafka 实例控制台功能

| Kafka Manager功能 | Kafka实例控制台功能                                        |
|-----------------|-----------------------------------------------------|
| 查看实例的Topic      | 在“实例管理 > Topic管理”页面，查看Topic列表。                      |
| 查看队列基本信息        | 在“实例管理 > Topic管理”页面，查看每个Topic的基本信息，包括副本数、分区数和老化时间等。 |
| Topic分区平衡       | 在“实例管理 > Topic管理”页面，修改分区平衡，支持手动平衡和自动平衡。             |
| 更新Topic配置       | 在“实例管理 > Topic管理”页面，修改Topic配置参数。                    |
| 查看消费组列表         | 在“实例管理 > 消费组管理”页面，查看消费组列表。                          |
| 查看消费组详情页        | 在“实例管理 > 消费组管理”页面，单击消费组名称，进入消费组详情页，查看消费者列表和消费进度。    |

| Kafka Manager功能 | Kafka实例控制台功能                                                                             |
|-----------------|------------------------------------------------------------------------------------------|
| 查看消费组队列详情页      | 在“实例管理 > 消费组管理”页面，单击消费组名称，进入消费组详情页。在“消费进度”页签中，查看消费组消费的所有Topic列表以及每个Topic的总堆积数和每个分区的消费状态。 |
| 集群监控信息/队列监控信息   | 在“监控 > 监控详情”页面，查看实例的监控信息。                                                                |

## 前提条件

已配置如表11-12所示安全组。

表 11-12 安全组规则

| 方向  | 协议  | 端口   | 源地址                 | 说明               |
|-----|-----|------|---------------------|------------------|
| 入方向 | TCP | 9999 | Kafka客户端所在的IP地址或地址组 | 访问Kafka Manager。 |

## 登录 Kafka Manager

- 步骤1** 创建一台与Kafka实例相同VPC和相同安全组的Windows服务器，详细步骤请参考[创建弹性云服务器](#)。
- 步骤2** 在实例详情信息页面，获取Kafka Manager地址。
- 步骤3** 在浏览器中输入Kafka Manager的地址，进入Kafka Manager登录页面。
- 步骤4** 输入创建实例时设置的Kafka Manager用户名和密码，即可管理Kafka集群。

----结束

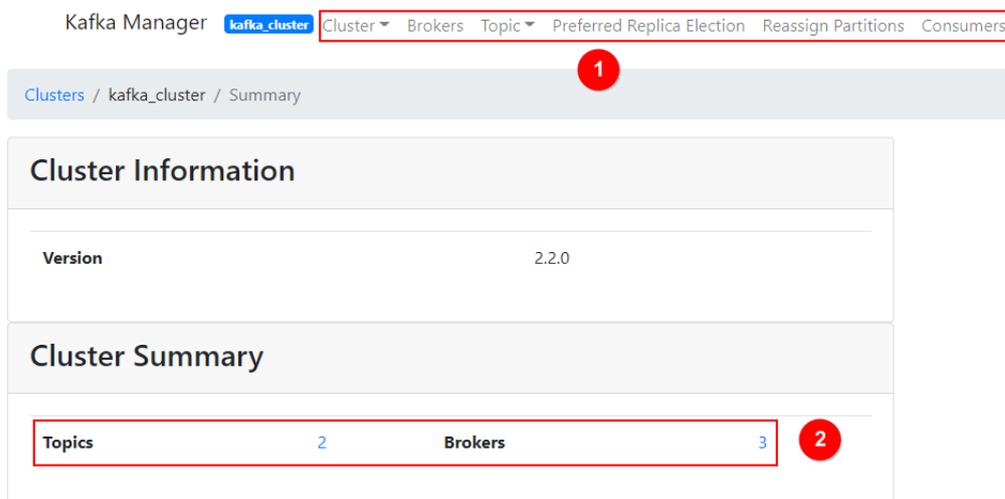
## 查看 Kafka Manager

在进入Kafka Manager集群管理页面后，您可以查看Kafka集群的监控、代理等信息。

- 集群信息页
  - 单击**Clusters**中的集群列表，即可进入集群信息页。如图11-2所示。
    - 图中①区域表示功能导航栏。
      - **Cluster**: 集群，统计集群列表和集群详情。
      - **Brokers**: 代理，统计当前集群中各代理的状态信息。
      - **Topic**: 队列，统计当前集群中的kafka队列。
      - **Preferred Replica Election**: 强制进行一次队列leader的最优选举（不建议用户操作）。

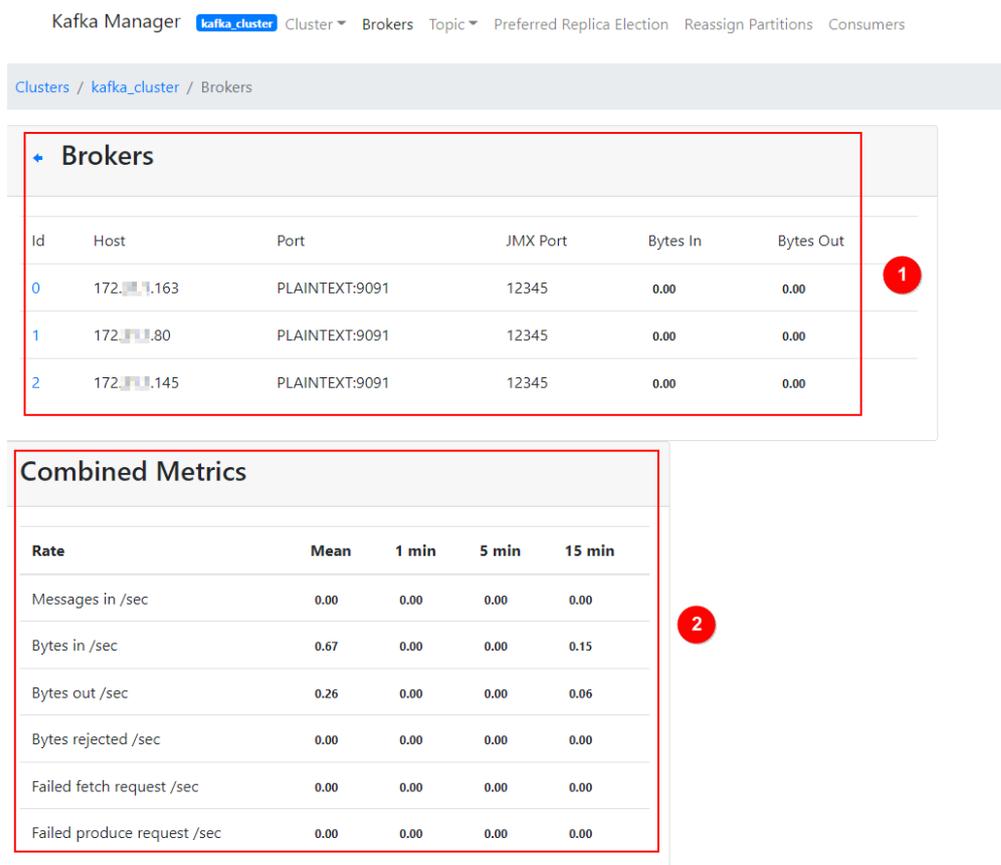
- **Reassign Partitions:** 进行分区副本的重分配（不建议用户操作）。
- **Consumers:** 统计集群中的消费组状态。
- 图中②区域表示集群信息统计，包含集群的Topic数和集群的代理数。

图 11-2 集群信息页



- 集群所有代理统计页  
单击功能导航栏中的**Brokers**，即可进入代理统计页。如图11-3所示。
  - 图中①区域代理列表，包含总的字节流入和字节流出。
  - 图中②集群监控信息。

图 11-3 所有代理统计页

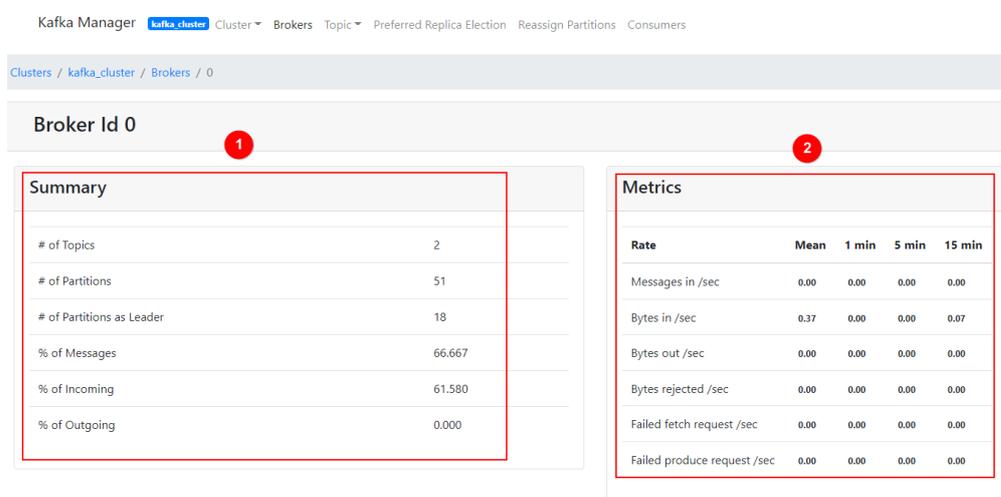


• 具体代理统计页

单击id列表中具体的Broker，即可查看对应代理的统计信息。如图11-4所示。

- 图中①区域表示对应代理总的统计信息，包括队列数、分区数、分区leader数、消息速率占比、写入字节占比以及流出字节占比。
- 图中②区域表示代理监控信息。

图 11-4 具体 Broker 信息



• 查看实例的Topic

在导航栏选择**Topic**，并在下拉列表中选择**List**。页面如图11-5所示，展示了队列列表以及分区数等。

列表中以“\_”开头的队列为内部队列，严禁操作，否则可能导致业务问题。

图 11-5 查看实例的 Topic

The screenshot shows the Kafka Manager interface for the 'kafka\_cluster' instance. The 'Topics' section is active, displaying a table of topics. The table has the following columns: Topic, # Partitions, # Brokers, Brokers Spread %, Brokers Skew %, Brokers Leader Skew %, # Replicas, Under Replicated %, Producer Message/Sec, and Summed Recent Offsets. Three topics are listed: '\_consumer\_offsets', '\_trace', and 'topic-test'. The '\_trace' row is highlighted in red, indicating it is an internal topic.

| Topic             | # Partitions | # Brokers | Brokers Spread % | Brokers Skew % | Brokers Leader Skew % | # Replicas | Under Replicated % | Producer Message/Sec | Summed Recent Offsets |
|-------------------|--------------|-----------|------------------|----------------|-----------------------|------------|--------------------|----------------------|-----------------------|
| _consumer_offsets | 50           | 3         | 100              | 0              | 0                     | 3          | 0                  | 0.00                 | 3                     |
| _trace            | 9            | 3         | 100              | 66             | 66                    | 1          | 0                  | 0.00                 | 0                     |
| topic-test        | 3            | 3         | 100              | 0              | 0                     | 3          | 0                  | 0.00                 | 0                     |

● 队列详情页

单击具体的Topic名称，进入如图11-6所示页面。

- 图中①区域表示队列基本信息，包括副本数(Replication)，分区数(Number of Partitions)，消息数(Sum of partition offsets)等。
- 图中②区域表示代理与队列分区的对应关系。
- 图中③区域表示该队列的消费组列表。单击消费组名称可进入该消费组的详情页。
- 图中④区域表示队列的配置信息。详情参考[Kafka队列官方配置文档](#)。
- 图中⑤区域表示队列监控数据统计。
- 图中⑥区域表示队列分区信息，包括分区消息数(Latest Offset)，分区 leader(Leader)，副本列表(Replicas)，同步副本列表(In Sync Replicas)。

图 11-6 队列详情页

topic-test

**Topic Summary**

|                             |     |
|-----------------------------|-----|
| Replication                 | 3   |
| Number of Partitions        | 3   |
| Sum of partition offsets    | 0   |
| Total number of Brokers     | 3   |
| Number of Brokers for Topic | 3   |
| Preferred Replicas %        | 100 |
| Brokers Skewed %            | 0   |
| Brokers Leader Skewed %     | 0   |
| Brokers Spread %            | 100 |
| Under-replicated %          | 0   |

**Config**

| Config       | Value     |
|--------------|-----------|
| retention.ms | 259200000 |

**Metrics**

| Rate                        | Mean | 1 min | 5 min | 15 min |
|-----------------------------|------|-------|-------|--------|
| Messages in /sec            | 0.00 | 0.00  | 0.00  | 0.00   |
| Bytes in /sec               | 0.00 | 0.00  | 0.00  | 0.00   |
| Bytes out /sec              | 0.00 | 0.00  | 0.00  | 0.00   |
| Bytes rejected /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   |

**Operations**

Delete Topic Reassign Partitions Generate Partition Assignments

Add Partitions Update Config Manual Partition Assignments

**Partitions by Broker**

| Broker | # of Partitions | # as Leader | Partitions | Skewed? | Leader Skewed? |
|--------|-----------------|-------------|------------|---------|----------------|
| 0      | 3               | 1           | (0,1,2)    | false   | false          |
| 1      | 3               | 1           | (0,1,2)    | false   | false          |
| 2      | 3               | 1           | (0,1,2)    | false   | false          |

**Consumers consuming from this topic**

| group | KF |
|-------|----|
| group | KF |

**Partition Information**

| Partition | Latest Offset | Leader | Replicas | In Sync Replicas | Preferred Leader? | Under Replicated? |
|-----------|---------------|--------|----------|------------------|-------------------|-------------------|
| 0         | 0             | 1      | (1,0,2)  | (1,0,2)          | true              | false             |
| 1         | 0             | 0      | (0,2,1)  | (0,2,1)          | true              | false             |
| 2         | 0             | 2      | (2,1,0)  | (2,1,0)          | true              | false             |

- 查看消费组列表

导航栏中单击**Consumers**，即可查看当前集群中的消费组列表。此页面只显示14天内有消费记录的消费组。

图 11-7 集群的消费组列表

Kafka Manager kafka\_cluster Cluster Brokers Topic Preferred Replica Election Reassign Partitions Consumers

Clusters / kafka\_cluster / Consumers

**Consumers**

Show 10 entries Search:

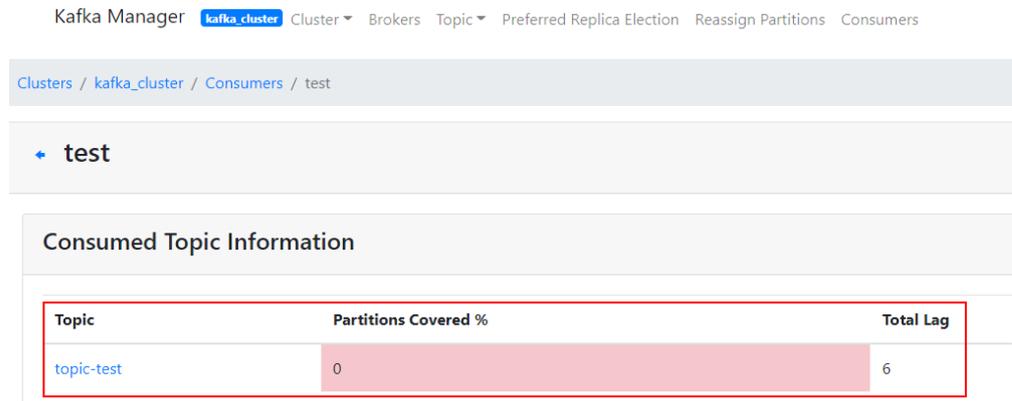
| Consumer | Type | Topics it consumes from         |
|----------|------|---------------------------------|
| group    | KF   | topic-test (0% coverage, 6 lag) |
| test     | KF   | topic-test (0% coverage, 0 lag) |

Showing 1 to 2 of 2 entries Previous 1 Next

- 查看消费组详情页

单击消费组名称可进入消费组详情页面，展示消费组消费的所有队列列表以及每个队列的可消费数（Total Lag）。

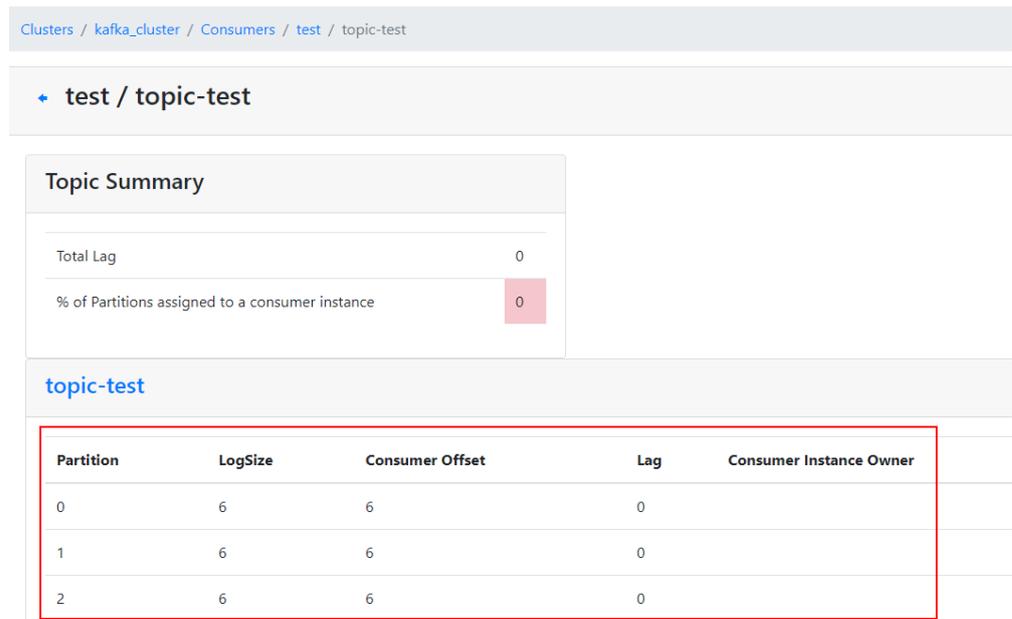
图 11-8 消费组详情页面



- 查看消费组队列详情页

单击队列名称，即可进入详情页，查看消费组消费在队列中每个分区的消费状态。包括分区编号（Partition）、分区消息数（LogSize）、分区消费进度（Consumer Offset）、分区剩余可消费数（Lag）和最近消费该分区的消费者（Consumer Instance Owner）。

图 11-9 消费组队列详情页



## 11.11.2 重置 Kafka Manager 密码

如果您忘记了创建实例时设置的Kafka Manager密码，通过重置Manager密码功能，重新设置一个新的密码，可使用新密码登录Kafka Manager。

### 前提条件

已创建Kafka实例，且实例处于“运行中”。

## 重置 Kafka Manager 密码

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 通过以下任意一种方法，重置Kafka Manager密码。

- 在待重置Manager密码的Kafka实例所在行，单击“更多 > 重置Manager密码”。
- 单击Kafka实例名称，进入实例详情页面。单击右上角的“... > 重置Manager密码”。
- 单击Kafka实例名称，进入实例详情页面。在“概览”页面，在“连接信息 > Manager用户名”后，单击“重置Manager密码”。

**步骤5** 输入“新密码”和“确认密码”，单击“确定”，完成密码重置。

- 重置Manager密码成功，界面提示重置实例的密码成功。
- 重置Manager密码失败，界面提示重置实例的密码失败，请重新尝试重置密码操作。如果多次重置失败，请联系客服处理。

### 说明

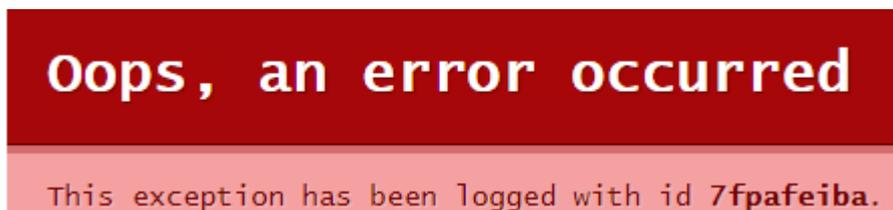
只有所有代理都重置密码成功，系统才会提示重置密码成功，否则会提示重置失败。

---结束

## 11.11.3 重启 Kafka Manager

当Kafka Manager无法登录或者无法使用时，例如下图中的报错，可以通过重启Kafka Manager，使Kafka Manager恢复正常。

图 11-10 报错信息



### 约束与限制

重启Kafka Manager不会影响业务。

## 重启 Kafka Manager

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入 Kafka实例列表页面。

**步骤4** 通过以下任意一种方法，重启Kafka Manager。

- 在待重启Manager的Kafka实例所在行，单击“更多 > 重启Manager”。
- 单击Kafka实例名称，进入实例详情页面。单击右上角的“... > 重启Manager”。

**步骤5** 单击“是”。

您可以在实例的“实例管理 > 后台任务管理”页面，查看当前任务的操作进度。任务状态为“成功”，表示重启成功。

----结束

## 11.11.4 关闭 Kafka Manager

Kafka Manager会占用额外的内存和CPU资源，关闭Kafka Manager可以减少这部分资源的占用。本章节指导您在控制台关闭Kafka Manager。

### 约束与限制

- Kafka Manager关闭后，无法再次开启。
- 关闭Kafka Manager，不会重启实例。

### 关闭 Kafka Manager

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入 Kafka实例列表页面。

**步骤4** 单击待关闭Kafka Manager的Kafka实例名称，进入实例详情页面。

**步骤5** 在“概览”页面的“连接信息 > Kafka Manager”后，单击 ，关闭Kafka Manager。

关闭Kafka Manager后，控制台不显示Kafka Manager连接地址，不支持重置Kafka Manager密码和重启Kafka Manager功能。

----结束

# 12 变更实例规格

## 12.1 变更 Kafka 集群实例规格

Kafka实例创建成功后，您可以根据业务需要扩容，Kafka实例支持的变更配置如表 12-1 所示。代理数量、基准带宽、存储空间和代理规格不能同时变更，每次变更只支持一种配置类型的变更。

表 12-1 变更配置列表

| 实例规格类型 | 变更配置类型 | 是否支持扩容 | 是否支持缩容 |
|--------|--------|--------|--------|
| 新规格实例  | 代理数量   | √      | ×      |
|        | 存储空间   | √      | ×      |
|        | 代理规格   | √      | ×      |
| 老规格实例  | 基准带宽   | √      | ×      |
|        | 存储空间   | √      | ×      |
|        | 代理规格   | ×      | ×      |

### 怎样判断老规格实例和新规格实例？

- 老规格实例：在实例列表中，“规格”显示为带宽（如100MB/s）的实例。
- 新规格实例：在实例列表中，“规格”显示为ECS规格和代理个数（如 kafka.2u4g.cluster\*3 broker）的实例。

图 12-1 实例列表

| 名称                                                                                   | 状态  | 版本    | 规格                            | 已用/可用存储空间 (GB) ... |
|--------------------------------------------------------------------------------------|-----|-------|-------------------------------|--------------------|
| kafka-doc <a href="#">🔗</a> <a href="#">🔗</a><br>c6f705cc-358b-470f-a71c-e18...      | 运行中 | 2.7   | kafka.2u4g.cluster * 3 broker | 0/300              |
| kafka-974006966 <a href="#">🔗</a> <a href="#">🔗</a><br>252b248f-501e-41ed-9832-48... | 运行中 | 2.3.0 | 100 MB/s                      | 0/186              |

## 变更实例规格的影响

表 12-2 变更实例规格的影响

| 变更配置类型    | 影响                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 基准带宽/代理数量 | <ul style="list-style-type: none"> <li>扩容基准带宽/代理数量不会影响原来的代理，<b>业务也不受影响</b>。</li> <li>扩容基准带宽/代理数量时，系统会根据当前磁盘大小进行相应比例的存储空间扩容。例如扩容前实例的代理数为3，每个代理的磁盘大小为200GB，扩容后实例的代理数为10，此时代理的磁盘大小依旧为200GB，但是总磁盘大小为2000GB。</li> <li><b>新创建的Topic才会分布在新代理上</b>，原有Topic还分布在原有代理上，造成分区分布不均匀。通过<a href="#">修改Kafka分区平衡</a>，实现将原有Topic分区的副本迁移到新代理上。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 存储空间      | <ul style="list-style-type: none"> <li>扩容存储空间有次数限制，只能扩容<b>20次</b>。</li> <li><b>扩容存储空间不会影响业务</b>。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 代理规格      | <ul style="list-style-type: none"> <li>若Topic为<b>单副本</b>，扩容期间无法对该Topic生产消息或消费消息，会造成<b>业务中断</b>。</li> <li>若Topic为<b>多副本</b>，扩容代理规格<b>不会造成服务中断</b>，但可能会导致消费的分区分区消息发生乱序，请谨慎评估业务影响，建议在业务低峰期扩容。</li> <li>扩容代理规格的过程中，节点滚动重启造成分区Leader切换，会发生<b>秒级连接闪断</b>，在用户网络环境稳定的前提下，Leader切换时长一般为<b>1分钟以内</b>。多副本的Topic需要在生产客户端配置<b>重试机制</b>，方法如下： <ul style="list-style-type: none"> <li>生产客户端为<b>Kafka开源客户端</b>时，检查是否配置retries参数，建议此参数值设置为3~5。</li> <li>生产客户端为<b>Flink客户端</b>时，检查是否配置重启策略，配置重启策略可以参考如下代码。 <pre>StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment(); env.setRestartStrategy(RestartStrategies.fixedDelayRestart(3, Time.seconds(20)));</pre> </li> </ul> </li> <li>变更代理规格的时长和代理数量有关，单个代理的变更时长一般在<b>5~10分钟</b>，代理数量越多，规格变更时间越长。</li> </ul> |

## 扩容代理规格的过程

在扩容代理规格的过程中，代理采用滚动重启的方式进行实例变更，具体过程如下（以3个代理为例介绍）：

1. 停止Broker 0的Kafka进程。
2. 扩容Broker 0的规格。
3. 重启Broker 0的Kafka进程。
4. 重复**1~3**，扩容Broker 1的规格。

5. 重复1~3，扩容Broker 2的规格。

图 12-2 扩容代理规格过程



## 扩容代理数量/基准带宽

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 在Kafka实例所在行，单击“变更规格”，进入“分布式消息服务Kafka版变更规格”页面。

**步骤5** 根据实际情况选择扩容代理数量或基准带宽。

- **扩容基准带宽（如何判断老规格实例）**

- a. 在“规格”中，选择扩容后的带宽，单击“下一步”。
- b. 确认扩容信息无误后，单击“提交”。
- c. 在实例列表页面，查看变更是否成功？

- 实例状态从“变更中”变为“运行中”时，表示变更成功，在“规格”中查看扩容后的带宽大小。

- 实例状态从“变更中”变为“变更失败”时，表示变更失败，把鼠标放在“变更失败”上，显示变更失败的原因。

实例状态为“变更失败”时，无法对实例进行重启、修改和删除等操作。等一段时间实例状态会自动从“变更失败”变为“运行中”，此时可以继续对实例进行操作。如果未变为“运行中”，请联系客服处理。

- d. 扩容基准带宽后，为了提高可靠性，需要在客户端连接配置中增加新扩容代理的IP地址。

- **扩容代理数量（如何判断新规格实例）**

- a. 在“变更类型”中，选择“代理数量”。
- b. 在“代理数量”中，选择扩容后的代理数量。不同实例规格对应的代理数量范围不同，具体请参考表1-1。
- c. 如果已开启公网访问，在“弹性IP地址”中，选择新扩容代理的公网IP地址。
- d. “子网”保持默认设置。

- e. 在“内网IP地址”中，选择“自动分配”或者“手动选择”。
  - 自动分配：系统自动分配子网中可用的IP地址。
  - 手动选择：在下拉框中为新扩容代理勾选指定的IP地址。如果指定的IP地址数量少于代理数量时，系统会自动为剩余代理分配IP地址。
- f. 单击“下一步”，进入变更信息确认页面。
- g. 确认扩容信息无误后，单击“提交”。
- h. 查看变更进度和预计剩余时间。
  - i. 在实例列表页面，单击实例名称，进入实例详情页。
  - ii. 在左侧导航栏单击“实例管理 > 后台任务管理”，进入“即时任务”页签。
  - iii. 单击“任务名”为“变更规格”的任务，弹出“变更规格任务详情”对话框。
  - iv. 在顶部区域，查看变更进度和预计剩余时间。在“任务执行步骤”中，查看执行步骤的名称和开始执行时间，以及结束时间。
- i. 查看变更是否成功。
  - 变更规格任务的状态为“成功”时，表示变更成功。在实例列表页面的“规格”中查看扩容后的代理数量。
  - 变更规格任务的状态为“失败”时，表示变更失败。把鼠标放在“失败”上，显示变更失败的原因。也可以在“任务执行步骤”中查看变更失败的原因。

变更失败后，实例状态会变为“变更失败”，此时无法对实例进行重启、修改和删除等操作。等一段时间实例状态会自动从“变更失败”变为“运行中”，此时可以继续对实例进行操作。如果未变为“运行中”，请联系客服处理。
- j. 扩容代理数量后，为了提高可靠性，需要在客户端连接配置中增加新扩容代理的IP地址。

---结束

## 扩容存储空间

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 在Kafka实例所在行，单击“变更规格”，进入“分布式消息服务Kafka版变更规格”页面。

**步骤5** 根据实际情况选择扩容存储空间。

- **扩容存储空间（如何判断老规格实例）**

- a. 在“存储空间”中，选择扩容后的存储空间大小，单击“下一步”。
- b. 确认扩容信息无误后，单击“提交”。

- c. 在实例列表页面，查看变更是否成功？
  - 实例状态从“变更中”变为“运行中”时，表示变更成功，在“已用/可用存储空间”中查看扩容后的存储空间大小。
  - 实例状态从“变更中”变为“变更失败”时，表示变更失败，把鼠标放在“变更失败”上，显示变更失败的原因。  
实例状态为“变更失败”时，无法对实例进行重启、修改和删除等操作。等一段时间实例状态会自动从“变更失败”变为“运行中”，此时可以继续对实例进行操作。如果未变为“运行中”，请联系客服处理。
- **扩容存储空间（如何判断新规格实例）**
  - a. 在“变更类型”中，选择“存储空间”。
  - b. 在“单个代理存储空间”中，选择扩容后的单个代理的存储空间大小，不同实例规格对应的存储空间范围不同，具体请参考表1-1，单击“下一步”。
  - c. 确认扩容信息无误后，单击“提交”。
  - d. 查看变更进度和预计剩余时间。
    - i. 在实例列表页面，单击实例名称，进入实例详情页。
    - ii. 在左侧导航栏单击“实例管理 > 后台任务管理”，进入“即时任务”页签。
    - iii. 单击“任务名”为“变更规格”的任务，弹出“变更规格任务详情”对话框。
    - iv. 在顶部区域，查看变更进度和预计剩余时间。在“任务执行步骤”中，查看执行步骤的名称和开始执行时间，以及结束时间。
  - e. 查看变更是否成功。
    - 变更规格任务的状态为“成功”时，表示变更成功。在实例列表页面的“可用存储空间”中查看扩容后的存储空间大小（即扩容后的单个代理的存储空间\*代理数量）。
    - 变更规格任务的状态为“失败”时，表示变更失败。把鼠标放在“失败”上，显示变更失败的原因。也可以在“任务执行步骤”中查看变更失败的原因。  
变更失败后，实例状态会变为“变更失败”，此时无法对实例进行重启、修改和删除等操作。等一段时间实例状态会自动从“变更失败”变为“运行中”，此时可以继续对实例进行操作。如果未变为“运行中”，请联系客服处理。

---结束

## 扩容代理规格

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 在Kafka实例所在行，单击“变更规格”，进入“分布式消息服务Kafka版变更规格”页面。

**步骤5** 根据实际情况选择扩容代理规格。

1. 在“变更类型”中，选择“代理规格”。
2. 在“代理规格”中，选择扩容后的代理规格。
3. 在“风险检查”区域，查看扩容前的检查项是否正常。  
如果检查项有异常，请先根据提示信息修改，然后单击“重新检查”。如果确认异常的检查项无需修改，请勾选“我已知晓风险。”。
4. 单击“下一步”，确认扩容信息无误后，单击“提交”。
5. 查看扩容进度和预计剩余时间。
  - a. 在实例列表页面，单击实例名称，进入实例详情页。
  - b. 在左侧导航栏单击“实例管理 > 后台任务管理”，进入“即时任务”页签。
  - c. 单击“任务名”为“变更规格”的任务，弹出“变更规格任务详情”对话框。
  - d. 在顶部区域，查看变更进度和预计剩余时间。在“任务执行步骤”中，查看执行步骤的名称和开始执行时间，以及结束时间。
6. 查看变更是否成功。
  - 变更规格任务的状态为“成功”时，表示变更成功。在实例列表页面的“规格”中查看扩容后的代理规格。
  - 变更规格任务的状态为“失败”时，表示变更失败。把鼠标放在“失败”上，显示变更失败的原因。也可以在“任务执行步骤”中查看变更失败的原因。  
变更失败后，实例状态会变为“变更失败”，此时无法对实例进行重启、修改和删除等操作。等一段时间实例状态会自动从“变更失败”变为“运行中”，此时可以继续对实例进行操作。如果未变为“运行中”，请联系客服处理。

---结束

# 13 迁移数据

## 13.1 Kafka 数据迁移概述

Kafka迁移是指将生产与消费消息的客户端切换为连接新Kafka，部分还涉及将持久化的消息文件迁移到新Kafka。主要涉及到以下2类场景：

- 业务上云且不希望业务有中断。  
在上云过程中，连续性要求高的业务，需要平滑迁移，不能有长时间的中断。
- 在云上变更业务部署。  
单AZ部署的Kafka实例，不具备AZ之间的容灾能力。用户对可靠性要求提升后，需要迁移到多AZ部署的实例上。

### 迁移方案概述

表 13-1 迁移方案概述

| 迁移方案             | 迁移工具 | 优点                                                                                                                          | 缺点                                                                                                                              |
|------------------|------|-----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| 先迁生产，再迁消费（不迁移数据） | -    | <ul style="list-style-type: none"><li>• 业界通用的迁移方案，操作步骤简单，无需安装其他插件。</li><li>• 迁移过程由业务侧自主控制。</li><li>• 可确保消息消费的顺序性。</li></ul> | <ul style="list-style-type: none"><li>• 切换过程存在延时，消费端需要先消费完原Kafka消息，然后才能消费新Kafka消息。</li><li>• 迁移完消费业务后，新Kafka可能存在消息积压。</li></ul> |
| 同时消费，后迁生产（不迁移数据） | -    | <ul style="list-style-type: none"><li>• 操作步骤简单，无需安装其他插件。</li><li>• 同时消费原Kafka和新Kafka消息，确保消息能及时消费，无消息积压。</li></ul>           | 在迁移生产的开始阶段，同时消费原Kafka与新Kafka，会导致部分消息之间的生产顺序无法保证，存在消息乱序的问题。                                                                      |

| 迁移方案                             | 迁移工具         | 优点                               | 缺点                                  |
|----------------------------------|--------------|----------------------------------|-------------------------------------|
| 使用 MirrorMaker 先迁移数据，然后迁消费，最后迁生产 | Mirror Maker | 新Kafka同步原Kafka的全量历史数据和实时生产的增量数据。 | 消费端迁移到新Kafka后，会重复消费历史消息，消费端需支持消息幂等。 |

## 迁移准备

- 配置网络环境。  
使用内网连接Kafka实例前，需要配置如下安全组。

表 13-2 安全组规则

| 方向  | 协议  | 端口   | 源地址                 | 说明                                                                                                                       |
|-----|-----|------|---------------------|--------------------------------------------------------------------------------------------------------------------------|
| 入方向 | TCP | 9092 | Kafka客户端所在的IP地址或地址组 | <ul style="list-style-type: none"> <li>使用内网通过同一个VPC访问Kafka实例（关闭SSL加密）。</li> <li>使用对等连接跨VPC访问Kafka实例（关闭SSL加密）。</li> </ul> |
| 入方向 | TCP | 9093 | Kafka客户端所在的IP地址或地址组 | <ul style="list-style-type: none"> <li>使用内网通过同一个VPC访问Kafka实例（开启SSL加密）。</li> <li>使用对等连接跨VPC访问Kafka实例（开启SSL加密）。</li> </ul> |

- 创建目标Kafka实例。  
目标Kafka的规格不能低于原业务使用的Kafka规格。具体请参考[购买Kafka实例](#)。
- 在目标Kafka实例中创建Topic。  
在目标Kafka实例上创建与原Kafka实例相同配置的Topic，包括Topic名称、副本数、分区数、消息老化时间，以及是否同步复制和落盘等。具体请参考[创建Kafka Topic](#)。

## 迁移方案一：先迁生产，再迁消费（不迁移数据）

先将生产消息的业务迁移到新Kafka，原Kafka不会有新消息生产。待原Kafka的消息全部消费完后，再将消费消息业务迁移到新Kafka，开始消费新Kafka的消息。

本方案为业界通用的迁移方案，操作步骤简单，迁移过程由业务侧自主控制，整个过程中消息不会存在乱序问题，**适用于对消息顺序有要求的场景**。但是该方案中需要等待消费业务执行完毕，存在一个时间差的问题，部分数据可能存在较大的端到端时延。

- 步骤1** 将生产客户端的Kafka连接地址修改为新Kafka实例的连接地址。
- 步骤2** 重启生产业务，使得生产者将新的消息发送到新Kafka实例中。
- 步骤3** 观察各消费组在原Kafka的消费进度，直到原Kafka中数据都已经被消费完毕。
- 步骤4** 将消费客户端的Kafka连接地址修改为新Kafka实例的连接地址。
- 步骤5** 重启消费业务，使得消费者从新Kafka实例中消费消息。
- 步骤6** 观察消费者是否能正常从新Kafka实例中获取数据。
- 步骤7** 迁移结束。

----结束

## 迁移方案二：同时消费，后迁生产（不迁移数据）

消费业务启用多个消费客户端，分别向原Kafka和新Kafka消费消息，然后将生产业务切换到新Kafka实例，这样能确保所有消息都被及时消费。

本方案中消费业务会在一段时间内同时消费原Kafka和新Kafka消息。由于在迁移生产业务之前，已经有消费业务运行在新Kafka实例上，因此不会存在端到端时延的问题。但在迁移生产的开始阶段，同时消费原Kafka与新Kafka，会导致部分消息之间的生产顺序无法保证，存在消息乱序的问题。此场景适用于对端到端时延有要求，但对消息顺序不敏感的业务。

- 步骤1** 启动新的消费客户端，配置Kafka连接地址为新Kafka实例的连接地址，消费新Kafka实例中的数据。  
原有消费客户端需继续运行，消费业务同时消费原Kafka与新Kafka实例的消息。
- 步骤2** 修改生产客户端，Kafka连接地址改为新Kafka实例的连接地址。
- 步骤3** 重启生产客户端，将生产业务迁移到新Kafka实例中。
- 步骤4** 生产业务迁移后，观察连接新Kafka实例的消费业务是否正常。
- 步骤5** 等待原Kafka中数据消费完毕，关闭原有消费业务客户端。
- 步骤6** 迁移结束。

----结束

## 迁移方案三：使用 MirrorMaker 先迁移数据，然后迁消费，最后迁生产

首先通过MirrorMaker同步两个Kafka的消息，其次将消费端迁移到新Kafka，最后将生产端迁移到新Kafka。

本方案依赖于MirrorMaker，MirrorMaker实时同步原Kafka和新Kafka的数据。待数据同步完成，先将消费端迁移到新Kafka，然后将生产端迁移到新Kafka。此场景适用于生产端不可停止，端到端有时延要求，但是可以兼容少量重复消费的业务。

- 步骤1** 使用MirrorMaker同步两个Kafka的消息。具体步骤请参见[使用MirrorMaker跨集群同步数据](#)。
- 步骤2** 将消费客户端的Kafka连接地址修改为新Kafka实例的连接地址。
- 步骤3** 重启消费业务，使得消费者从新Kafka实例中消费消息。

- 步骤4 观察消费者是否能正常从新Kafka实例中获取数据。
  - 步骤5 修改生产客户端，Kafka连接地址改为新Kafka实例的连接地址。
  - 步骤6 重启生产客户端，将生产业务迁移到新Kafka实例中。
  - 步骤7 生产业务迁移后，观察连接新Kafka实例的消费业务是否正常。
  - 步骤8 迁移结束。
- 结束

## 常见问题：如何将持久化数据也一起迁移

如果需要将原Kafka的已消费数据也迁移到Kafka实例，可以使用开源工具 [MirrorMaker](#)，模拟成原Kafka的消费客户端，以及新Kafka实例的生产客户端，将Kafka所有消息数据迁移到新的Kafka实例，具体步骤请参考[使用MirrorMaker跨集群同步数据](#)。

## 13.2 使用 MirrorMaker 跨集群同步数据

### 方案概述

#### 应用场景

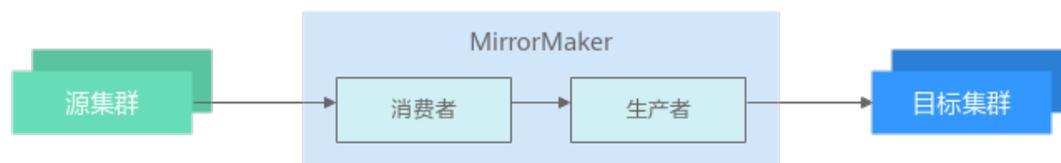
在以下场景，使用MirrorMaker进行不同集群间的数据同步，可以确保Kafka集群的可用性和可靠性。

- 备份和容灾：企业存在多个数据中心，为了防止其中一个数据中心出现问题，导致业务不可用，会将集群数据同步备份在多个不同的数据中心。
- 集群迁移：当今很多企业将业务迁移上云，迁移过程中需要确保线下集群和云上集群的数据同步，保证业务的连续性。

#### 方案架构

使用MirrorMaker可以实现将源集群中的数据镜像复制到目标集群中。其原理如[图 13-1](#)所示，MirrorMaker本质上也是生产消费消息，首先从源集群中消费数据，然后将消费的数据生产到目标集群。如果您需要了解更多关于MirrorMaker的信息，请参见[Mirroring data between clusters](#)。

图 13-1 MirrorMaker



### 约束与限制

- 源集群中节点的IP地址和端口号不能和目标集群中节点的IP地址和端口号相同，否则会导致数据在Topic内无限循环复制。
- 使用MirrorMaker同步数据，至少需要有两个或以上集群，不可在单个集群内部使用MirrorMaker，否则会导致数据在Topic内无限循环复制。

## 实施步骤

**步骤1** 购买一台弹性云服务器，确保弹性云服务器与源集群、目标集群网络互通。具体购买操作，请参考[创建弹性云服务器](#)。

**步骤2** 登录弹性云服务器，安装Java JDK，并配置JAVA\_HOME与PATH环境变量，使用执行用户在用户家目录下修改“.bash\_profile”，添加如下行。其中“/opt/java/jdk1.8.0\_151”为JDK的安装路径，请根据实际情况修改。

```
export JAVA_HOME=/opt/java/jdk1.8.0_151
export PATH=$JAVA_HOME/bin:$PATH
```

执行source .bash\_profile命令使修改生效。

### 说明

弹性云服务器默认自带的JDK可能不符合要求，例如OpenJDK，需要配置为Oracle的JDK，可至Oracle官方下载页面[下载Java Development Kit 1.8.111及以上版本](#)。

**步骤3** 下载[Kafka 3.3.1版本的二进制软件包](#)。

```
wget https://archive.apache.org/dist/kafka/3.3.1/kafka_2.12-3.3.1.tgz
```

**步骤4** 解压二进制软件包。

```
tar -zxvf kafka_2.12-3.3.1.tgz
```

**步骤5** 进入二进制软件包目录，修改“config”目录下的“connect-mirror-maker.properties”的配置文件，在配置文件中指定源集群和目标集群的IP地址和端口以及其他配置。

```
# 指定两个集群
clusters = A, B
A.bootstrap.servers = A_host1:A_port, A_host2:A_port, A_host3:A_port
B.bootstrap.servers = B_host1:B_port, B_host2:B_port, B_host3:B_port

# 指定数据同步方向，可以单向同步也可互相同步
A->B.enabled = true

# 指定同步的Topic，支持正则匹配，默认复制全部Topic，如："foo-.*"
A->B.topics = .*

# 取消以下两个配置的注释表示A、B两个集群互相复制同步
#B->A.enabled = true
#B->A.topics = .*

# 设置副本个数，如果需要同步多个Topic且副本数各不相同，建议先创建同名同副本数的Topic再启动
MirrorMaker
replication.factor=3

# 设置消费进度同步方向，可以单向同步也可互相同步
A->B.sync.group.offsets.enabled=true

##### Internal Topic Settings #####
# The replication factor for mm2 internal topics "heartbeats", "B.checkpoints.internal" and
# "mm2-offset-syncs.B.internal"
# 测试环境可以为1，生产环境建议以下配置大于1，比如设为3
checkpoints.topic.replication.factor=3
heartbeats.topic.replication.factor=3
offset-syncs.topic.replication.factor=3

# The replication factor for connect internal topics "mm2-configs.B.internal", "mm2-offsets.B.internal" and
# "mm2-status.B.internal"
# 测试环境可以为1，生产环境建议以下配置大于1，比如设为3
offset.storage.replication.factor=3
status.storage.replication.factor=3
config.storage.replication.factor=3

# customize as needed
```

```
# replication.policy.separator = _
# sync.topic.acls.enabled = false
# emit.heartbeats.interval.seconds = 5
# 设置目标集群中的Topic名称和源集群相同
# replication.policy.class = org.apache.kafka.connect.mirror.IdentityReplicationPolicy
```

**步骤6** 在二进制软件包目录下，启动MirrorMaker，进行数据同步。

```
./bin/connect-mirror-maker.sh config/connect-mirror-maker.properties
```

**步骤7** （可选）MirrorMaker开启后，如果在源集群上新建了Topic，如需对此Topic进行数据同步，则需重启MirrorMaker，重启步骤参考**步骤6**。也可配置自动同步新增Topic，按需增加如**表13-3**所示配置后，无需重启MirrorMaker，即可周期性同步新增Topic。其中，“refresh.topics.interval.seconds”为必选，其他参数根据实际情况选择。

**表 13-3** MirrorMaker 配置参数

| 参数名                               | 默认值                                                      | 说明                                       |
|-----------------------------------|----------------------------------------------------------|------------------------------------------|
| sync.topic.configs.enabled        | true                                                     | 是否监控源集群的配置更改                             |
| sync.topic.acls.enabled           | true                                                     | 是否监控源集群ACL的更改                            |
| emit.heartbeats.enabled           | true                                                     | 连接器应定期发出心跳                               |
| emit.heartbeats.interval.seconds  | 5秒                                                       | 心跳频率                                     |
| emit.checkpoints.enabled          | true                                                     | 连接器应定期发出消费端偏移量信息                         |
| emit.checkpoints.interval.seconds | 5秒                                                       | 检查点的频率                                   |
| refresh.topics.enabled            | true                                                     | 连接器应定期检查新主题                              |
| refresh.topics.interval.seconds   | 5秒                                                       | 检查源集群中是否有新主题的频率                          |
| refresh.groups.enabled            | true                                                     | 连接器应定期检查新的消费组                            |
| refresh.groups.interval.seconds   | 5秒                                                       | 检查源集群新的消费组频率                             |
| replication.policy.class          | org.apache.kafka.connect.mirror.DefaultReplicationPolicy | 使用LegacyReplicationPolicy模仿旧版MirrorMaker |
| heartbeats.topic.retention.ms     | 1天                                                       | 首次创建心跳主题时使用                              |
| checkpoints.topic.retention.ms    | 1天                                                       | 首次创建检查点主题时使用                             |
| offset.syncs.topic.retention.ms   | max long                                                 | 首次创建偏移同步主题时使用                            |

----结束

## 验证数据是否同步

**步骤1** 在目标集群中查看Topic列表，确认是否有源集群Topic。

### 📖 说明

“replication.policy.class”的默认值为  
“org.apache.kafka.connect.mirror.DefaultReplicationPolicy”，此时目标集群中的Topic名称和源集群相比，多了前缀（如A.），这是MirrorMaker为了防止Topic循环备份进行的设置。如果想要Topic名称保持一致，请将“replication.policy.class”设置为  
“org.apache.kafka.connect.mirror.IdentityReplicationPolicy”。

**步骤2** 在源集群生产并消费消息，在目标集群查看消费进度，确认数据是否已从源集群同步到了目标集群。

如果目标集群为云服务平台Kafka实例的话，在分布式消息服务Kafka版控制台的“实例管理 > 消费组管理 > 消费进度”中，查看消费进度。

----结束

# 14 申请扩大 Kafka 配额

## 什么是配额？

为防止资源滥用，平台限定了各服务资源的配额，对用户的资源数量和容量做了限制。如您最多可以创建多少个Kafka实例。

如果当前资源配额限制无法满足使用需要，您可以申请扩大配额。

## 怎样查看我的配额？

1. 登录管理控制台。
2. 单击管理控制台左上角的 ，选择区域和项目。
3. 单击页面右上角的“My Quota”图标 。  
系统进入“服务配额”页面。
4. 您可以在“服务配额”页面，查看各项资源的总配额及使用情况。  
如果当前配额不能满足业务要求，请参考后续操作，申请扩大配额。

## 如何申请扩大配额？

目前系统暂不支持在线调整配额大小。如您需要调整配额，请拨打热线或发送邮件至客服，客服会及时为您处理配额调整的需求，并以电话或邮件的形式告知您实时进展。

在拨打热线或发送邮件之前，请您准备好以下信息：

- Account Name、Project Name、Project ID。  
获取方式如下：登录云账户管理控制台，在右上角单击账户名，选择“我的认证”，在“我的认证”页面获取相关信息。
- 配额信息，包括：
  - 服务名
  - 配额类别
  - 需要的配额值

如您需要调整配额，请联系管理员。

# 15 查看监控指标与配置告警

## 15.1 查看 Kafka 监控数据

云监控对Kafka实例的运行状态进行日常监控，可以通过控制台直观的查看Kafka实例各项监控指标。

### 前提条件

已创建Kafka实例，且实例中有可消费的消息。

### 查看 Kafka 监控数据

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 通过以下任意一种方法，查看监控数据。

- **查看所有监控数据**：在Kafka实例名称后，单击“查看监控数据”。跳转到云监控页面，查看监控数据，数据更新周期为1分钟。
- **查看所有监控数据**：单击Kafka实例名称，进入实例详情页。在左侧导航栏单击“监控 > 监控详情”，进入监控页面，查看监控数据，数据更新周期为1分钟。

Kafka实例支持查看实例、节点、主题和消费组的监控数据。

----结束

## 15.2 Kafka 支持的监控指标

### 功能说明

本章节定义了分布式消息服务Kafka版上报云监控服务的监控指标的命名空间，监控指标列表和维度定义，您可以通过云监控服务的管理控制台或[API接口](#)来检索Kafka实例

产生的监控指标和告警信息，也可以通过分布式消息服务Kafka版控制台提供的“监控详情”页面来检索Kafka实例产生的监控指标。

## 命名空间

SYS.DMS

## 实例监控指标

表 15-1 实例支持的监控指标

| 指标ID                    | 指标名称  | 指标含义                                     | 取值范围         | 单位     | 进制        | 测量对象(维度) | 监控周期(原始指标) |
|-------------------------|-------|------------------------------------------|--------------|--------|-----------|----------|------------|
| current_partitions      | 分区数   | 该指标用于统计Kafka实例中已经使用的分区数量。                | 0~100000     | Count  | 不涉及       | Kafka实例  | 1分钟        |
| current_topics          | 主题数   | 该指标用于统计Kafka实例中已经创建的主题数量。                | 0~100000     | Count  | 不涉及       | Kafka实例  | 1分钟        |
| group_msgs              | 消息堆积数 | 该指标用于统计Kafka实例中所有消费组中总堆积消息数。             | 0~1000000000 | Count  | 不涉及       | Kafka实例  | 1分钟        |
| instance_bytes_in_rate  | 生产流量  | 统计实例中每秒生产的字节数。<br>部分存量实例不支持此监控，具体以控制台为准。 | 0~1000000    | Byte/s | 1024(IEC) | Kafka实例  | 1分钟        |
| instance_bytes_out_rate | 消费流量  | 统计实例中每秒消费的字节数。<br>部分存量实例不支持此监控，具体以控制台为准。 | 0~1000000    | Byte/s | 1024(IEC) | Kafka实例  | 1分钟        |

## 节点监控指标

表 15-2 节点支持的监控指标

| 指标 ID                   | 指标名称       | 指标含义                             | 取值范围        | 单位      | 进制        | 测量对象 (维度) | 监控周期 (原始指标) |
|-------------------------|------------|----------------------------------|-------------|---------|-----------|-----------|-------------|
| broker_data_size        | 节点数据容量     | 该指标用于统计节点当前的消息数据大小。              | 0~500000000 | Byte    | 1024(IEC) | Kafka实例节点 | 1分钟         |
| broker_messages_in_rate | 消息生产速率     | 该指标用于统计每秒生产的消息数量。                | 0~50000     | Count/s | 不涉及       | Kafka实例节点 | 1分钟         |
| broker_bytes_out_rate   | 消费流量       | 该指标用于统计每秒消费的字节数。                 | 0~50000000  | Byte/s  | 1024(IEC) | Kafka实例节点 | 1分钟         |
| broker_bytes_in_rate    | 生产流量       | 该指标用于统计每秒生产的字节数。                 | 0~50000000  | Byte/s  | 1024(IEC) | Kafka实例节点 | 1分钟         |
| broker_fetch_mean       | 消费请求平均处理时长 | 统计Broker节点处理消费请求平均时长。            | 0~10000     | ms      | 不涉及       | Kafka实例节点 | 1分钟         |
| broker_produce_mean     | 生产请求平均处理时长 | 统计Broker节点处理生产请求平均时长。            | 0~10000     | ms      | 不涉及       | Kafka实例节点 | 1分钟         |
| broker_cpu_core_load    | CPU核均负载    | 该指标为从Kafka节点虚拟机层面采集的CPU每个核的平均负载。 | 0~20        | 不涉及     | 不涉及       | Kafka实例节点 | 1分钟         |
| broker_disk_usage       | 磁盘容量使用率    | 该指标为从Kafka节点虚拟机层面采集的磁盘容量使用率。     | 0~100       | %       | 不涉及       | Kafka实例节点 | 1分钟         |

| 指标 ID                       | 指标名称      | 指标含义                             | 取值范围                                                                       | 单位     | 进制        | 测量对象 (维度) | 监控周期 (原始指标) |
|-----------------------------|-----------|----------------------------------|----------------------------------------------------------------------------|--------|-----------|-----------|-------------|
| broker_memory_usage         | 内存使用率     | 该指标为Kafka节点虚拟机层面采集的内存使用率。        | 0~100                                                                      | %      | 不涉及       | Kafka实例节点 | 1分钟         |
| broker_heap_usage           | JVM堆内存使用率 | 该指标从Kafka节点Kafka进程JVM中采集的堆内存使用率。 | 0~100                                                                      | %      | 不涉及       | Kafka实例节点 | 1分钟         |
| broker_alive                | 节点存活状态    | 表示Kafka节点是否存活。                   | <ul style="list-style-type: none"> <li>• 1: 存活</li> <li>• 0: 离线</li> </ul> | 不涉及    | 不涉及       | Kafka实例节点 | 1分钟         |
| broker_connections          | 连接数       | 统计Kafka节点当前所有TCP连接数量。            | 0~65535                                                                    | Count  | 不涉及       | Kafka实例节点 | 1分钟         |
| broker_cpu_usage            | CPU使用率    | 统计Kafka节点虚拟机的CPU使用率。             | 0~100                                                                      | %      | 不涉及       | Kafka实例节点 | 1分钟         |
| broker_disk_read_await      | 磁盘平均读操作耗时 | 该指标用于统计磁盘在测量周期内平均每个读IO的操作时长。     | >0                                                                         | ms     | 不涉及       | Kafka实例节点 | 1分钟         |
| broker_disk_write_await     | 磁盘平均写操作耗时 | 该指标用于统计磁盘在测量周期内平均每个写IO的操作时长。     | >0                                                                         | ms     | 不涉及       | Kafka实例节点 | 1分钟         |
| broker_total_bytes_in_rate  | 网络入流量     | 统计Broker节点每秒网络访问流入流量。            | 0~100000000                                                                | Byte/s | 1024(IEC) | Kafka实例节点 | 1分钟         |
| broker_total_bytes_out_rate | 网络出流量     | 统计Broker节点每秒网络访问流出流量。            | 0~100000000                                                                | Byte/s | 1024(IEC) | Kafka实例节点 | 1分钟         |

| 指标 ID                   | 指标名称    | 指标含义     | 取值范围  | 单位     | 进制        | 测量对象 (维度) | 监控周期 (原始指标) |
|-------------------------|---------|----------|-------|--------|-----------|-----------|-------------|
| broker_disk_read_rate   | 磁盘读流量   | 磁盘读操作流量。 | >=0   | Byte/s | 1024(IEC) | Kafka实例节点 | 1分钟         |
| broker_disk_write_rate  | 磁盘写流量   | 磁盘写操作流量。 | >=0   | Byte/s | 1024(IEC) | Kafka实例节点 | 1分钟         |
| network_bandwidth_usage | 网络带宽利用率 | 网络带宽利用率。 | 0~100 | %      | 不涉及       | Kafka实例节点 | 1分钟         |

## 主题监控指标

表 15-3 主题支持的监控指标

| 指标 ID                | 指标名称 | 指标含义                                                | 取值范围       | 单位     | 进制        | 测量对象 (维度) | 监控周期 (原始指标) |
|----------------------|------|-----------------------------------------------------|------------|--------|-----------|-----------|-------------|
| topic_bytes_in_rate  | 生产流量 | 该指标用于统计每秒生产的字节数。<br>在“主题”页签，当“监控类型”为“基本监控”时，才包含该指标。 | 0~50000000 | Byte/s | 1024(IEC) | Kafka实例队列 | 1分钟         |
| topic_bytes_out_rate | 消费流量 | 该指标用于统计每秒消费的字节数。<br>在“主题”页签，当“监控类型”为“基本监控”时，才包含该指标。 | 0~50000000 | Byte/s | 1024(IEC) | Kafka实例队列 | 1分钟         |

| 指标 ID                  | 指标名称   | 指标含义                                                   | 取值范围         | 单位      | 进制        | 测量对象 (维度) | 监控周期 (原始指标) |
|------------------------|--------|--------------------------------------------------------|--------------|---------|-----------|-----------|-------------|
| topic_data_size        | 队列数据容量 | 该指标用于统计队列当前的消息数据大小。<br>在“主题”页签，当“监控类型”为“基本监控”时，才包含该指标。 | 0~5000000000 | Byte    | 1024(IEC) | Kafka实例队列 | 1分钟         |
| topic_messages         | 队列消息总数 | 该指标用于统计队列当前的消息总数。<br>在“主题”页签，当“监控类型”为“基本监控”时，才包含该指标。   | ≥ 0          | Count   | 不涉及       | Kafka实例队列 | 1分钟         |
| topic_messages_in_rate | 消息生产速率 | 该指标用于统计每秒生产的消息数量。<br>在“主题”页签，当“监控类型”为“基本监控”时，才包含该指标。   | 0~500000     | Count/s | 不涉及       | Kafka实例队列 | 1分钟         |
| partition_messages     | 分区消息数  | 该指标用于统计分区中当前的消息个数。<br>在“主题”页签，当“监控类型”为“分区监控”时，才包含该指标。  | ≥ 0          | Count   | 不涉及       | Kafka实例队列 | 1分钟         |

| 指标 ID             | 指标名称  | 指标含义                                                 | 取值范围 | 单位    | 进制  | 测量对象 (维度) | 监控周期 (原始指标) |
|-------------------|-------|------------------------------------------------------|------|-------|-----|-----------|-------------|
| produced_messages | 生产消息数 | 该指标用于统计目前生产的消息总数。<br>在“主题”页签，当“监控类型”为“分区监控”时，才包含该指标。 | ≥ 0  | Count | 不涉及 | Kafka实例队列 | 1分钟         |

## 消费组监控指标

表 15-4 消费组支持的监控指标

| 指标 ID              | 指标名称     | 指标含义                                                                        | 取值范围 | 单位    | 进制  | 测量对象 (维度)   | 监控周期 (原始指标) |
|--------------------|----------|-----------------------------------------------------------------------------|------|-------|-----|-------------|-------------|
| messages_consumed  | 分区已消费消息数 | 该指标用于统计当前消费组已经消费的消息个数。<br>在“消费组”页签，“主题”为指定的 Topic 名称，“监控类型”为“分区监控”时，才包含该指标。 | ≥ 0  | Count | 不涉及 | Kafka实例的消费组 | 1分钟         |
| messages_remaining | 分区可消费消息数 | 该指标用于统计消费组可消费的消息个数。<br>在“消费组”页签，“主题”为指定的 Topic 名称，“监控类型”为“分区监控”时，才包含该指标。    | ≥ 0  | Count | 不涉及 | Kafka实例的消费组 | 1分钟         |

| 指标 ID                       | 指标名称              | 指标含义                                                                           | 取值范围                 | 单位    | 进制  | 测量对象 (维度)    | 监控周期 (原始指标) |
|-----------------------------|-------------------|--------------------------------------------------------------------------------|----------------------|-------|-----|--------------|-------------|
| topic_messages_remaining    | 队列可消费消息数          | 该指标用于统计消费组指定队列可以消费的消息个数。<br>在“消费组”页签，“主题”为指定的 Topic 名称，“监控类型”为“基本监控”时，才包含该指标。  | 0~<br>( $2^{63}-1$ ) | Count | 不涉及 | Kafka 实例的消费组 | 1分钟         |
| topic_messages_consumed     | 队列已消费消息数          | 该指标用于统计消费组指定队列当前已经消费的消息数。<br>在“消费组”页签，“主题”为指定的 Topic 名称，“监控类型”为“基本监控”时，才包含该指标。 | 0~<br>( $2^{63}-1$ ) | Count | 不涉及 | Kafka 实例的消费组 | 1分钟         |
| consumer_messages_remaining | 消息堆积数 (消费组可消费消息数) | 该指标用于统计消费组剩余可以消费的消息个数。<br>在“消费组”页签，“主题”为“全部队列”时，才包含该指标。                        | 0~<br>( $2^{63}-1$ ) | Count | 不涉及 | Kafka 实例的消费组 | 1分钟         |
| consumer_messages_consumed  | 消费组已消费消息数         | 该指标用于统计消费组当前已经消费的消息数。<br>在“消费组”页签，“主题”为“全部队列”时，才包含该指标。                         | 0~<br>( $2^{63}-1$ ) | Count | 不涉及 | Kafka 实例的消费组 | 1分钟         |

| 指标 ID                              | 指标名称    | 指标含义                                                                                               | 取值范围       | 单位        | 进制  | 测量对象 (维度)    | 监控周期 (原始指标) |
|------------------------------------|---------|----------------------------------------------------------------------------------------------------|------------|-----------|-----|--------------|-------------|
| messages_consumed_per_min          | 分区消费速率  | 统计消费组指定队列分区每分钟的消费数。<br>在“消费组”页签，“主题”为指定的 Topic 名称，“监控类型”为“分区监控”时，才包含该指标。<br>部分存量实例不支持此监控，具体以控制台为准。 | 0~30000000 | Count/min | 不涉及 | Kafka 实例的消费组 | 1分钟         |
| topic_messages_consumed_per_min    | 队列消费速率  | 统计消费组指定队列每分钟的消费数。<br>在“消费组”页签，“主题”为指定的 Topic 名称，“监控类型”为“基本监控”时，才包含该指标。<br>部分存量实例不支持此监控，具体以控制台为准。   | 0~30000000 | Count/min | 不涉及 | Kafka 实例的消费组 | 1分钟         |
| consumer_messages_consumed_per_min | 消费组消费速率 | 统计消费组每分钟的消费数。<br>在“消费组”页签，“主题”为“全部 Topic”时，才包含该指标。<br>部分存量实例不支持此监控，具体以控制台为准。                       | 0~30000000 | Count/min | 不涉及 | Kafka 实例的消费组 | 1分钟         |

## 维度

| Key                     | Value         |
|-------------------------|---------------|
| kafka_instance_id       | Kafka实例       |
| kafka_broker            | Kafka实例节点     |
| kafka_topics            | Kafka实例主题     |
| kafka_partitions        | Kafka实例分区     |
| kafka_groups-partitions | Kafka实例分区的消费组 |
| kafka_groups_topics     | Kafka实例队列的消费组 |
| kafka_groups            | Kafka实例的消费组   |

## 15.3 配置 Kafka 监控告警

本章节主要介绍部分监控指标的告警策略，以及配置操作。在实际业务中，建议按照以下告警策略，配置监控指标的告警规则。

表 15-5 Kafka 实例的告警策略和处理建议

| 指标ID                | 指标名称    | 测量对象 | 告警策略                                | 指标说明                         | 告警处理建议                                                                 |
|---------------------|---------|------|-------------------------------------|------------------------------|------------------------------------------------------------------------|
| broker_disk_usage   | 磁盘容量使用率 | 节点   | 告警阈值：原始值>80%<br>连续触发次数：1<br>告警级别：紧急 | 该指标为从Kafka节点虚拟机层面采集的磁盘容量使用率。 | 出现该告警时，需要修改实例 <b>存储空间</b> 。具体操作，请参考 <a href="#">变更实例规格</a> 。           |
| broker_memory_usage | 内存使用率   | 节点   | 告警阈值：原始值>90%<br>连续触发次数：3<br>告警级别：紧急 | 该指标为Kafka节点虚拟机层面采集的内存使用率。    | 出现该告警时，需要修改实例 <b>基准带宽/代理个数</b> ，即扩节点。具体操作，请参考 <a href="#">变更实例规格</a> 。 |

| 指标ID                   | 指标名称     | 测量对象 | 告警策略                                                                                     | 指标说明                         | 告警处理建议                                                                                                                |
|------------------------|----------|------|------------------------------------------------------------------------------------------|------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| current_partitions     | 分区数      | 实例   | 告警阈值：原始值>分区数上限的90%，不同实例规格分区数上限不同，具体参考 <a href="#">Kafka实例规格</a> 。<br>连续触发次数：1<br>告警级别：重要 | 该指标用于统计Kafka实例中已经使用的分区数量。    | 出现该告警时，如果业务后续还需要新增Topic，则需要修改实例 <b>基准带宽/代理个数</b> 或将业务拆分至多个实例。修改实例 <b>基准带宽/代理个数</b> 的具体操作，请参考 <a href="#">变更实例规格</a> 。 |
| broker_cpu_usage       | CPU使用率   | 节点   | 告警阈值：原始值>90%<br>连续触发次数：3<br>告警级别：重要                                                      | 统计Kafka节点虚拟机的CPU使用率。         | 出现该告警时，先检查该监控是否长期处于接近或超过告警阈值状态，如果是，需要修改实例 <b>基准带宽/代理个数</b> ，即扩节点。具体操作，请参考 <a href="#">变更实例规格</a> 。                    |
| group_msgs             | 堆积消息数    | 实例   | 告警阈值：原始值>积压上限的90%，积压上限由您根据业务实际情况设定<br>连续触发次数：1<br>告警级别：重要                                | 该指标用于统计Kafka实例中所有消费组中总堆积消息数。 | 出现该告警时，首先排查是否有闲置消费组，如果有，则删除。其次，可以考虑加快消费速度，例如增加组内消费者数量等。                                                               |
| topic_messages_remaind | 队列可消费消息数 | 消费组  | 告警阈值：原始值>积压上限的90%，积压上限由您根据业务实际情况设定<br>连续触发次数：1<br>告警级别：重要                                | 该指标用于统计消费组指定队列可以消费的消息个数。     | 出现该告警时，首先排查消费者代码逻辑是否有误，例如消费者出现了异常不再消费等。其次，可以考虑加快消息的消费，例如增加队列消费者，并确保分区数大于或等于消费者数。                                      |

## 配置 Kafka 告警规则

以下步骤指导您配置指定Kafka实例的告警规则。

**步骤1** 登录管理控制台。

**步骤2** 在管理控制台左上角单击 ，选择Kafka实例所在的区域。

**步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 Kafka”，进入Kafka实例列表页面。

**步骤4** 通过以下任意一种方法，进入监控页面。

- 在Kafka实例名称后，单击“查看监控数据”，进入云监控该实例的监控指标页面。
- 单击Kafka实例名称，进入实例详情页。在左侧导航栏单击“监控 > 监控详情”，进入监控页面。

**步骤5** 在实例监控指标页面中，找到需要创建告警的指标项，鼠标移动到指标区域，然后单击指标右上角的 ，跳转到创建告警规则页面。

**步骤6** 在告警规则页面，设置告警信息。

创建告警规则操作，请查看[创建告警规则](#)。

1. 设置告警名称和告警的描述。
2. 设置告警策略。

如下图所示，在进行指标监控时，如果连续3个周期，磁盘容量使用率原始值  $\geq 85\%$ ，则产生告警，如果未及时处理，则产生告警通知。

**图 15-1** 设置告警策略和告警级别



3. 设置“发送通知”开关。  
当开启时，设置告警生效时间、产生告警时通知的对象以及触发的条件。
4. 单击“立即创建”，等待创建告警规则成功。

----结束

# 16 查看 Kafka 审计日志

通过云审计服务（Cloud Trace Service, CTS），您可以记录与分布式消息服务Kafka版相关的操作事件，便于日后的查询、审计和回溯。

## 前提条件

已开通CTS。

## CTS 支持的 DMS for Kafka 操作

表 16-1 云审计服务支持的 DMS for Kafka 操作列表

| 操作名称            | 资源类型  | 事件名称                                   |
|-----------------|-------|----------------------------------------|
| 创建DMS实例任务执行成功   | kafka | createDMSInstanceTaskSuccess           |
| 创建DMS实例任务执行失败   | kafka | createDMSInstanceTaskFailure           |
| 删除创建失败的DMS实例成功  | kafka | deleteDMSCreateFailureInstancesSuccess |
| 删除创建失败的DMS实例失败  | kafka | deleteDMSCreateFailureInstancesFailure |
| 删除DMS实例任务执行成功   | kafka | deleteDMSInstanceTaskSuccess           |
| 删除DMS实例任务执行失败   | kafka | deleteDMSInstanceTaskFailure           |
| 批量删除DMS实例任务     | kafka | batchDeleteDMSInstanceTask             |
| 提交批量删除DMS实例请求成功 | kafka | batchDeleteDMSInstanceSuccess          |
| 批量删除DMS实例任务执行成功 | kafka | batchDeleteDMSInstanceTaskSuccess      |

| 操作名称            | 资源类型     | 事件名称                               |
|-----------------|----------|------------------------------------|
| 提交批量删除DMS实例请求失败 | kafka    | batchDeleteDMSInstanceFailure      |
| 批量删除DMS实例任务执行失败 | kafka    | batchDeleteDMSInstanceTaskFailure  |
| 提交扩容实例请求成功      | kafka    | extendDMSInstanceSuccess           |
| 扩容DMS实例任务执行成功   | kafka    | extendDMSInstanceTaskSuccess       |
| 提交扩容实例请求失败      | kafka    | extendDMSInstanceFailure           |
| 扩容DMS实例任务执行失败   | kafka    | extendDMSInstanceTaskFailure       |
| 提交重置DMS实例密码请求成功 | kafka    | resetDMSInstancePasswordSuccess    |
| 提交重置DMS实例密码请求失败 | kafka    | resetDMSInstancePasswordFailure    |
| 提交重启DMS实例请求成功   | kafka    | restartDMSInstanceSuccess          |
| 重启DMS实例任务执行成功   | kafka    | restartDMSInstanceTaskSuccess      |
| 提交重启DMS实例请求失败   | kafka    | restartDMSInstanceFailure          |
| 重启DMS实例任务执行失败   | kafka    | restartDMSInstanceTaskFailure      |
| 提交批量重启DMS实例请求成功 | instance | batchRestartDMSInstanceSuccess     |
| 批量重启DMS实例任务执行成功 | kafka    | batchRestartDMSInstanceTaskSuccess |
| 提交批量重启DMS实例请求失败 | instance | batchRestartDMSInstanceFailure     |
| 批量重启DMS实例任务执行失败 | kafka    | batchRestartDMSInstanceTaskFailure |
| 提交修改DMS实例信息请求成功 | kafka    | modifyDMSInstanceInfoSuccess       |
| 修改DMS实例信息任务执行成功 | kafka    | modifyDMSInstanceInfoTaskSuccess   |
| 提交修改DMS实例信息请求失败 | kafka    | modifyDMSInstanceInfoFailure       |

| 操作名称              | 资源类型  | 事件名称                               |
|-------------------|-------|------------------------------------|
| 修改DMS实例信息任务执行失败   | kafka | modifyDMSInstanceInfoTaskFailure   |
| 删除后台任务成功          | kafka | deleteDMSBackendJobSuccess         |
| 删除后台任务失败          | kafka | deleteDMSBackendJobFailure         |
| Kafka实例创建Topic成功  | kafka | Kafka_create_topicSuccess          |
| Kafka实例创建Topic失败  | kafka | Kafka_create_topicFailure          |
| Kafka实例删除Topic成功  | kafka | Kafka_delete_topicsSuccess         |
| Kafka实例删除Topic失败  | kafka | Kafka_delete_topicsFailure         |
| 开启自动创建Topic成功     | kafka | enable_auto_topicSuccess           |
| 开启自动创建Topic失败     | kafka | enable_auto_topicFailure           |
| 修改Topic成功         | kafka | Kafka_alter_topicsSuccess          |
| 修改Topic失败         | kafka | Kafka_alter_topicsFailure          |
| 执行Topic分区平衡任务成功   | kafka | kafka_reassignmentTaskSuccess      |
| 执行Topic分区平衡任务失败   | kafka | kafka_reassignmentTaskFailure      |
| 提交执行Topic分区平衡请求成功 | kafka | kafka_reassignmentSuccess          |
| 提交执行Topic分区平衡请求失败 | kafka | kafka_reassignmentFailure          |
| 重置消费组偏移量成功        | kafka | Kafka_reset_consumer_offsetSuccess |
| 重置消费组偏移量失败        | kafka | Kafka_reset_consumer_offsetFailure |
| 批量删除消费组成功         | kafka | Kafka_batch_delete_groupSuccess    |
| 批量删除消费组失败         | kafka | Kafka_batch_delete_groupFailure    |
| 创建用户成功            | kafka | createUserSuccess                  |
| 创建用户失败            | kafka | createUserFailure                  |

| 操作名称      | 资源类型  | 事件名称                          |
|-----------|-------|-------------------------------|
| 删除用户成功    | kafka | deleteUserSuccess             |
| 删除用户失败    | kafka | deleteUserFailure             |
| 更新用户策略成功  | kafka | updateUserPoliciesTaskSuccess |
| 更新用户策略失败  | kafka | updateUserPoliciesTaskFailure |
| 回收站恢复实例成功 | kafka | out_recycleTaskSuccess        |
| 回收站恢复实例失败 | kafka | out_recycleTaskFailure        |

## 查看审计日志

查看DMS for Kafka云审计日志，请参考[查询审计事件](#)。

# 17 常见问题

## 17.1 实例问题

### 17.1.1 为什么可用区不能选择 2 个？

如果您需要提高Kafka实例的可靠性，在创建实例时，建议选择3个或以上的可用区，不支持选择2个可用区。原因如下：

每个Kafka实例包含3个Zookeeper节点，Zookeeper集群用来管理Kafka实例的配置，如果Zookeeper集群出现问题，Kafka实例将无法正常运行。至少2个Zookeeper节点正常运行，才能保证Zookeeper集群正常运行。

假设选择2个可用区，可用区1有1个Zookeeper节点，可用区2有2个Zookeeper节点。如果可用区1故障，则Kafka实例能正常使用；如果可用区2故障，则不能正常使用。Kafka实例可用的场景只有50%，所以不支持选择2个可用区。

### 17.1.2 创建实例时为什么无法查看子网和安全组等信息？

创建实例时，如果无法查看虚拟私有云、子网、安全组、弹性IP，可能原因是该用户的用户组无Server Administrator和VPC Administrator权限。增加用户组权限的详细步骤，请参考[修改用户组权限](#)。

### 17.1.3 如何选择 Kafka 实例的存储空间？

存储空间主要是指用于存储消息（包括副本中的消息）、日志和元数据所需要的空间。选择存储空间时，需要选择磁盘类型和磁盘大小。更多磁盘信息，请参考[如何选择磁盘类型](#)。

假设业务存储数据保留天数内磁盘大小为100GB，则磁盘容量最少为**100GB\*副本数 + 预留磁盘大小100GB**。Kafka集群中，每个Kafka节点会使用33GB的磁盘作为日志和Zookeeper数据的存储，因而实际可用存储会小于购买存储。

其中，**副本数**在创建Topic时可以选择，默认为3副本存储。如果开启了Kafka自动创建Topic功能，自动创建的Topic默认为3副本，副本数可以通过“实例管理 > 配置参数”页签中的“default.replication.factor”修改。

## 17.1.4 Kafka 实例的超高 IO 和高 IO 如何选择？

- 高IO：平均时延1-3ms，最大带宽150MB/s（读+写）。
- 超高IO：平均时延1ms，最大带宽350MB/s（读+写）。

建议选择超高IO，云硬盘服务端压力大场景，都不能达到最大带宽，但是超高IO可达到的带宽比高IO高很多。

## 17.1.5 如何选择 Kafka 实例存储容量阈值策略？

支持以下两种策略：

- 生产受限策略  
该策略场景下一旦磁盘使用达到容量阈值95%，会导致后续生产失败，但保留了当前磁盘中的数据，直至数据自然老化（Kafka原有的老化机制，数据默认保留3天）。该场景适用于对数据不能丢的业务场景，但是会导致生产业务失败。
- 自动删除策略  
该策略场景下磁盘使用到达容量阈值95%后，依旧可以正常生产和消费消息，但是会删除最早的10%的消息，以保证磁盘容量充足。该场景优先保障业务不中断，数据可能会丢失。

以上两种策略需要基于业务对数据和业务的可靠性来进行选择，只能作为极端场景下的一种处理方式。建议业务购买时保证有充足的磁盘容量，避免磁盘的使用达到容量阈值。

## 17.1.6 Kafka 服务端支持版本是多少？

Kafka 1.1.0、2.3.0、2.7和3.x版本。

如果您想要创建Kafka实例，具体步骤请参考[购买Kafka实例](#)。

## 17.1.7 Kafka 实例的 ZK 地址是什么？

Kafka实例依赖ZK进行管理，开放ZK可能引发误操作导致业务受损，ZK仅Kafka内部使用，不对外提供服务。

您可使用Kafka开源客户端连接Kafka实例，调用原生API生产与消费消息。

## 17.1.8 创建的 Kafka 实例是集群模式么？

一个Kafka实例即为一个集群实例，Kafka实例由3个及以上代理组成。

## 17.1.9 Kafka 实例是否支持修改访问端口？

Kafka实例的访问端口固定，不支持修改。

- 如果是访问未开启SASL的Kafka实例  
支持的访问方式如下：同一VPC内访问、跨VPC访问。
  - 同一VPC内使用IPv4地址访问，访问端口为**9092**。
  - 同一VPC内使用IPv6地址访问，访问端口为**9192**。
  - 使用VPCEP跨VPC访问，访问端口为**9011**。
  - 使用对等连接跨VPC访问，访问端口为**9092**。

- 如果是访问开启SASL的Kafka实例支持的访问方式如下：同一VPC内访问、跨VPC访问。
  - 同一VPC内使用IPv4地址访问，访问端口为**9093**。
  - 同一VPC内使用IPv6地址访问，访问端口为**9193**。
  - 使用VPCEP跨VPC访问，访问端口为**9011**。
  - 使用对等连接跨VPC访问，访问端口为**9093**。

在访问Kafka实例之前，需要确保安全组是否配置正确，配置要求，请参考[选择和配置安全组](#)。

### 17.1.10 Kafka 实例的 SSL 证书有效期多长？

Kafka实例开启SASL\_SSL时，需进行单向认证，证书有效期足够长（超过15年），客户端不需要关注证书过期风险。

查看SSL证书有效期的方法如下：

- 步骤1** 解压从Kafka实例控制台下载的压缩包，获取压缩包中的“phy\_ca.crt”。
- 步骤2** 双击“phy\_ca.crt”，弹出“证书”对话框。
- 步骤3** 在“常规”页签中，获取证书有效期。

图 17-1 证书有效期



---结束

### 17.1.11 如何将 Kafka 实例中的数据同步到另一个 Kafka 实例中?

如果需要做Kafka实例迁移,可以同时向两个Kafka实例生产消息,源实例中的消息可继续消费,待源实例的消息数据全部被消费完或老化后,业务可迁移到新的Kafka实例。

### 17.1.12 Kafka 实例的 SASL\_SSL 开关如何修改?

Kafka SASL\_SSL开关不支持创建实例后修改,在创建时,请慎重选择。如果创建后需要修改,需要重新创建实例。

### 17.1.13 SASL 认证机制如何修改?

Kafka实例的SASL认证机制包括SCRAM-SHA-512和PLAIN。

实例创建后,不支持修改SASL认证机制。如果需要修改,请重新创建实例。

### 17.1.14 修改企业项目，是否会导致 Kafka 重启？

修改企业项目不会导致Kafka重启。

### 17.1.15 Kafka 服务和 ZK 是部署在相同的虚拟机中，还是分开部署？

Kafka服务和ZK部署在相同的虚拟机中。

### 17.1.16 Kafka 支持哪些加密套件？

由于安全问题，支持的加密套件为  
TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256,  
TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256和  
TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256。

### 17.1.17 购买实例时选择的单 AZ，怎样可以扩展为多 AZ？

已购买的实例无法扩展AZ，请重新购买多AZ的实例。

### 17.1.18 Kafka 是否支持跨 AZ 容灾？已经购买的实例在哪里查看是否为跨 AZ？

Kafka实例支持跨AZ容灾，在购买实例时，选择多个可用区即可实现跨AZ容灾。

已购买的实例，在Kafka控制台“概览”页面的“网络”区域，查看可用区数量。多于一个可用区时，表示已实现跨AZ容灾。

图 17-2 实例网络信息

|       |                                                                                              |
|-------|----------------------------------------------------------------------------------------------|
| 网络    |                                                                                              |
| 可用区   | 可用区1,可用区2,可用区3                                                                               |
| 虚拟私有云 | vpc-kafka                                                                                    |
| 子网    | subnet-kafka                                                                                 |
| 安全组   | sg-kafka  |

### 17.1.19 Kafka 支持磁盘加密吗？

Kafka实例不支持磁盘加密。

### 17.1.20 Kafka 实例创建后，能修改 VPC 和子网吗？

不能修改VPC和子网。

## 17.1.21 有没有 Kafka Stream 的案例？

Kafka Stream的案例请参见[Kafka Stream的官网](#)。

## 17.1.22 Kafka 实例版本可以升级吗？

Kafka实例创建成功后，实例版本不支持升级。您可以重新创建Kafka实例，实现升级Kafka实例的版本。

## 17.1.23 配置了“自动删除”策略，磁盘容量到达 95%时，为什么没有自动删除最早的消息？

### 问题现象

Kafka实例每个代理（即broker）的存储空间为100GB，配置了“自动删除”策略，并创建了一个180分区的Topic，磁盘容量到达95%时，没有自动删除最早的消息。

### 可能原因

Kafka实例每个代理会使用33GB的磁盘作为日志和Zookeeper数据的存储，且创建实例时磁盘格式化会占用部分磁盘空间，因此实际数据存储空间大约为66GB。Topic的每个分区都是由多个segment文件组成，每个segment文件的最大存储容量为500MB。Kafka删除消息是删除segment文件，而不是删除一条消息。Kafka要求每个分区至少保留一个segment文件用来存储消息，如果分区中仅剩一个segment文件，此segment文件不会被删除。在分区数到达132个（ $66\text{GB}/500\text{MB}=132$ ）及以上时，可能会达到磁盘存储空间使用上限。

磁盘容量到达95%（即62.7GB）时，由于Topic配置了180个分区，每个分区中可能只存在一个segment文件，此segment文件无法被删除，其存储的消息也不会被删除。

### 解决方法

建议扩容磁盘空间，具体操作请参考[变更Kafka集群实例规格](#)。

## 17.2 实例规格变更问题

### 17.2.1 Kafka 扩容会影响业务吗？

扩容Kafka实例的影响如[表17-1](#)所示。

表 17-1 变更实例规格的影响

| 变更配置类型    | 影响                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 基准带宽/代理数量 | <ul style="list-style-type: none"> <li>扩容基准带宽/代理数量不会影响原来的代理，<b>业务也不受影响</b>。</li> <li>扩容基准带宽/代理数量时，系统会根据当前磁盘大小进行相应比例的存储空间扩容。例如扩容前实例的代理数为3，每个代理的磁盘大小为200GB，扩容后实例的代理数为10，此时代理的磁盘大小依旧为200GB，但是总磁盘大小为2000GB。</li> <li><b>新创建的Topic才会分布在新代理上</b>，原有Topic还分布在原有代理上，造成分区分布不均匀。通过<b>修改Kafka分区平衡</b>，实现将原有Topic分区的副本迁移到新代理上。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 存储空间      | <ul style="list-style-type: none"> <li>扩容存储空间有次数限制，只能扩容<b>20次</b>。</li> <li><b>扩容存储空间不会影响业务</b>。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 代理规格      | <ul style="list-style-type: none"> <li>若Topic为<b>单副本</b>，扩容期间无法对该Topic生产消息或消费消息，会造成<b>业务中断</b>。</li> <li>若Topic为<b>多副本</b>，扩容代理规格<b>不会造成服务中断</b>，但可能会导致消费的分区消息发生乱序，请谨慎评估业务影响，建议在业务低峰期扩容。</li> <li>扩容代理规格的过程中，节点滚动重启造成分区Leader切换，会发生<b>秒级连接闪断</b>，在用户网络环境稳定的前提下，Leader切换时长一般为<b>1分钟以内</b>。多副本的Topic需要在生产客户端配置<b>重试机制</b>，方法如下： <ul style="list-style-type: none"> <li>生产客户端为<b>Kafka开源客户端</b>时，检查是否配置retries参数，建议此参数值设置为3~5。</li> <li>生产客户端为<b>Flink客户端</b>时，检查是否配置重启策略，配置重启策略可以参考如下代码。 <pre>StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment(); env.setRestartStrategy(RestartStrategies.fixedDelayRestart(3, Time.seconds(20)));</pre> </li> </ul> </li> <li>变更代理规格的时长和代理数量有关，单个代理的变更时长一般在<b>5~10分钟</b>，代理数量越多，规格变更时间越长。</li> </ul> |

## 17.2.2 扩容过程中涉及数据迁移吗？

扩容过程中不会迁移数据。

## 17.2.3 扩容过程中生产消息失败

**可能原因：**在扩容代理规格的过程中，代理采用滚动重启的方式进行实例变更。在重启过程中，分区Leader会进行切换，此时生产客户端的元数据缓存中保存的分区Leader为旧分区Leader ID，仍然会向旧分区Leader发送消息，导致生产消息失败。

**解决方法：**在生产客户端配置重试机制。建议设置retries=Integer.MAX\_VALUE

## 17.2.4 扩容失败提示资源不足

**问题现象：**扩容失败，提示底层ECS/EVS资源不足，但在ECS控制台可以购买对应规格的ECS。

**可能原因：**底层资源配额与控制台界面显示的可用规格配额存在差异。

**解决方法：**联系客服增加配额数。

## 17.3 连接问题

### 17.3.1 选择和配置安全组

Kafka实例支持使用内网通过同一个VPC访问，访问实例前，需要配置安全组。

#### 使用内网通过同一个 VPC 访问实例

**步骤1** 客户端和实例是否使用相同的安全组？

- 是，如果保留了创建安全组后，系统默认添加的入方向“允许安全组内的弹性云服务器彼此通信”规则和出方向“放通全部流量”规则，则无需添加其他规则。否则，请添加表17-2所示规则。

表 17-2 安全组规则

| 方向  | 协议  | 类型   | 端口   | 源地址                 | 说明                              |
|-----|-----|------|------|---------------------|---------------------------------|
| 入方向 | TCP | IPv4 | 9092 | Kafka客户端所在的IP地址或地址组 | 使用内网通过同一个VPC访问Kafka实例（关闭SSL加密）。 |
| 入方向 | TCP | IPv6 | 9192 | Kafka客户端所在的IP地址或地址组 | 使用内网通过同一个VPC访问Kafka实例（关闭SSL加密）。 |
| 入方向 | TCP | IPv4 | 9093 | Kafka客户端所在的IP地址或地址组 | 使用内网通过同一个VPC访问Kafka实例（开启SSL加密）。 |
| 入方向 | TCP | IPv6 | 9193 | Kafka客户端所在的IP地址或地址组 | 使用内网通过同一个VPC访问Kafka实例（开启SSL加密）。 |

- 否，执行**步骤2**。

**步骤2** 参考如下配置安全组规则。

假设客户端和Kafka实例的安全组分别为：sg-53d4、Default\_All。以下规则，远端可使用安全组，也可以使用具体的IP地址，本章节以安全组为例介绍。

客户端所在安全组需要增加如下规则，以保证客户端能正常访问Kafka实例。

表 17-3 安全组规则

| 方向  | 策略 | 协议端口 | 目的地址        |
|-----|----|------|-------------|
| 出方向 | 允许 | 全部   | Default_All |

图 17-3 配置客户端安全组



Kafka实例所在安全组需要增加如下规则，以保证能被客户端访问。

表 17-4 安全组规则

| 方向  | 策略 | 协议端口 | 源地址     |
|-----|----|------|---------|
| 入方向 | 允许 | 全部   | sg-53d4 |

图 17-4 配置 Kafka 实例安全组



----结束

## 跨 VPC 访问和通过 DNAT 访问实例

请按照表17-5设置安全组规则。

表 17-5 安全组规则

| 方向  | 协议  | 端口   | 源地址                 | 说明                                            |
|-----|-----|------|---------------------|-----------------------------------------------|
| 入方向 | TCP | 9011 | 198.19.128.0/17     | 通过VPC终端节点实现跨VPC访问Kafka实例（开启SSL加密和关闭SSL加密都适用）。 |
| 入方向 | TCP | 9092 | Kafka客户端所在的IP地址或地址组 | 使用对等连接跨VPC访问Kafka实例（关闭SSL加密）。                 |
| 入方向 | TCP | 9093 | Kafka客户端所在的IP地址或地址组 | 使用对等连接跨VPC访问Kafka实例（开启SSL加密）。                 |

### 17.3.2 Kafka 实例是否支持公网访问？

暂时不支持公网访问。您必须通过同一虚拟私有云下的ECS来访问Kafka实例，以确保安全。

### 17.3.3 Kafka 实例的连接地址默认有多少个？

Kafka实例的连接地址个数和实例的代理个数有关，连接地址个数即为代理个数。每类实例规格对应的代理个数如下表所示。

表 17-6 Kafka 集群实例规格

| 规格名称                  | 代理个数范围 | 单个代理TPS | 单个代理分区上限 | 单个代理建议消费组数 | 单个代理客户端总连接数上限 | 存储空间范围 (GB) | 单个代理流量规格 (MB/s) |
|-----------------------|--------|---------|----------|------------|---------------|-------------|-----------------|
| kafka.2u4g.clusters   | 3~30   | 30000   | 250      | 20         | 2000          | 300~300000  | 100             |
| kafka.4u8g.clusters   | 3~30   | 100000  | 500      | 100        | 4000          | 300~600000  | 200             |
| kafka.8u16g.clusters  | 3~50   | 150000  | 1000     | 150        | 4000          | 300~1500000 | 375             |
| kafka.12u24g.clusters | 3~50   | 200000  | 1500     | 200        | 4000          | 300~1500000 | 625             |

| 规格名称                  | 代理个数范围 | 单个代理TPS | 单个代理分区上限 | 单个代理建议消费组数 | 单个代理客户端总连接数上限 | 存储空间范围 (GB) | 单个代理流量规格 (MB/s) |
|-----------------------|--------|---------|----------|------------|---------------|-------------|-----------------|
| kafka.16u32g.clusters | 3~50   | 250000  | 2000     | 200        | 4000          | 300~1500000 | 750             |

### 17.3.4 是否支持跨 Region 访问?

Kafka可以跨Region访问，但是跨Region目前只能拉专线的方式。

### 17.3.5 Kafka 实例是否支持跨 VPC 访问?

Kafka实例支持跨VPC访问，您可以通过以下任意一个方式实现跨VPC访问：

- 创建VPC对等连接，将两个VPC的网络打通，实现跨VPC访问。具体步骤请参考[对等连接](#)。
- 利用VPC终端节点在不同VPC间建立跨VPC的连接通道，实现Kafka客户端通过内网访问Kafka实例。具体步骤请参考[使用VPC对等连接实现跨VPC访问Kafka](#)。

### 17.3.6 Kafka 实例是否支持不同的子网?

支持。

客户端与实例在相同VPC内，可以跨子网段访问。同一个VPC内的子网默认可以进行通信。

### 17.3.7 Kafka 是否支持 Kerberos 认证，如何开启认证?

Kafka支持SASL客户端认证、调用接口支持Token和AK/SK两种认证，Kerberos认证目前不支持。

如果使用SASL认证方式，则在开源客户端基础上使用分布式消息服务Kafka版提供的证书文件。具体操作参考[使用客户端连接Kafka（密文接入）](#)。

### 17.3.8 Kafka 实例是否支持无密码访问?

支持，连接未开启SASL的Kafka实例时，无需密码。具体操作，请参考[使用客户端连接Kafka（明文接入）](#)。

### 17.3.9 Kafka 支持服务端认证客户端吗?

不支持。

### 17.3.10 连接开启 SASL\_SSL 的 Kafka 实例时，ssl truststore 文件可以用 PEM 格式的吗?

使用Java语言连接实例时，只能使用JKS格式的证书，不支持转成PEM格式。

### 17.3.11 下载的证书 JKS 和 CRT 有什么区别？

使用Java语言连接实例时，需要用JKS格式的证书。使用Python语言连接实例时，需要用CRT格式的证书。

### 17.3.12 Kafka 支持哪个版本的 TLS？

Kafka支持TLS 1.2。

### 17.3.13 Kafka 实例连接数有限制吗？

不同规格的Kafka实例，连接数限制如下：

表 17-7 老规格 Kafka 实例的连接数

| 基准带宽     | 连接数上限 |
|----------|-------|
| 100MB/s  | 3000  |
| 300MB/s  | 10000 |
| 600MB/s  | 20000 |
| 1200MB/s | 20000 |

表 17-8 新规格 Kafka 实例的连接数

| 实例规格                 | 单个代理客户端总连接数上限 |
|----------------------|---------------|
| kafka.2u4g.cluster   | 2000          |
| kafka.4u8g.cluster   | 4000          |
| kafka.8u16g.cluster  | 4000          |
| kafka.12u24g.cluster | 4000          |
| kafka.16u32g.cluster | 4000          |

### 17.3.14 客户端单 IP 连接的个数为多少？

Kafka实例的每个代理允许客户端单IP连接的个数默认为1000个，如果超过了，会出现连接失败问题。您可以通过[修改Kafka实例配置参数](#)来修改单IP的连接数。

### 17.3.15 Kafka 实例的内网连接地址可以修改吗？

不支持修改，且不支持指定IP地址。

### 17.3.16 不同实例中，使用的 SSL 证书是否一样？

Kafka实例中的SSL证书是通用的，不区分用户或者实例，即不同的用户或者实例，使用的SSL证书是同一个。

获取SSL证书的方法如下：

**步骤1** 在Kafka控制台，单击实例名称，进入实例详情页。

**步骤2** 在“连接信息 > SSL证书”所在行，单击“下载”。

----结束

## 17.3.17 为什么不建议使用 Sarama 客户端收发消息？

### 问题现象

使用Sarama客户端收发消息可能存在以下问题：

- Sarama客户端无法感知分区变化，当Topic分区数增加时，需要重启客户端后才能正常消费。
- Sarama客户端消息最大处理时间（MaxProcessingTime）默认值为100ms，超过最大处理时间可能导致消费者无法消费。
- 当消费位点重置策略设置为Oldest(earliest)时，当客户端重启时，偏移量重置后可能从最小位点开始重复消费所有消息。
- 消费者同时订阅多个Topic时，部分分区可能无法消费到消息。

### 解决方法

建议您使用[Confluent-Kafka-go](#)作为Kafka客户端库。

以下为您提供常用Golang客户端的对比，具体参见[表17-9](#)。

表 17-9 常用 Golang 客户端对比

| 客户端                | 优点                                                                                                                                                                  | 缺点                                                                                                                                                     |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| Confluent-Kafka-go | <ul style="list-style-type: none"> <li>• Confluent-Kafka-go是由Confluent提供的官方Kafka客户端库，与Kafka完全兼容，支持所有的Kafka特性。</li> <li>• 稳定性高，基于librdkafka，具有高性能和低延迟的特点。</li> </ul> | <p>增加编译复杂度。由于导入C++库，Golang编译器需要引入额外编译配置，增加了编译依赖，提高编译复杂度。</p>                                                                                           |
| Kafka-go           | <ul style="list-style-type: none"> <li>• kafka-go是一个简单、轻量级的Kafka客户端库，易于学习和使用。</li> <li>• kafka-go的代码库相对较小，依赖较少，可以减少应用程序的体积和依赖关系。</li> </ul>                         | <ul style="list-style-type: none"> <li>• Kafka-go相对于Confluent-Kafka-go来说，功能较为有限，不支持一些高级特性和复杂的配置选项。</li> <li>• 性能和吞吐量较低，适用于一些对性能要求不高的简单应用场景。</li> </ul> |

| 客户端    | 优点                                     | 缺点                                                                                                                    |
|--------|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| Sarama | Sarama采用原生Golang语言编写，对于异步以及高并发操作支持度较好。 | <ul style="list-style-type: none"> <li>问题较多，文档相对较少。</li> <li>Sarama在处理大量消息时，会占用较多的内存资源，可能会对应用程序的性能造成一定的影响。</li> </ul> |

## 17.4 Topic 和分区问题

### 17.4.1 Kafka 实例的 Topic 数量是否有限制？

Topic数量和Topic总分区数、每个Topic的分区数有关，Kafka实例对Topic总分区数设置了上限，当达到上限后，会导致用户无法继续创建Topic。

不同规格配置的Topic总分区数不同，如下表所示。

表 17-10 Kafka 集群实例规格

| 规格名称                 | 代理个数范围 | 单个代理TPS | 单个代理分区上限 | 单个代理建议消费组数 | 单个代理客户端总连接数上限 | 存储空间范围 (GB) | 单个代理流量规格 (MB/s) |
|----------------------|--------|---------|----------|------------|---------------|-------------|-----------------|
| kafka.2u4g.cluster   | 3~30   | 30000   | 250      | 20         | 2000          | 300~300000  | 100             |
| kafka.4u8g.cluster   | 3~30   | 100000  | 500      | 100        | 4000          | 300~600000  | 200             |
| kafka.8u16g.cluster  | 3~50   | 150000  | 1000     | 150        | 4000          | 300~1500000 | 375             |
| kafka.12u24g.cluster | 3~50   | 200000  | 1500     | 200        | 4000          | 300~1500000 | 625             |
| kafka.16u32g.cluster | 3~50   | 250000  | 2000     | 200        | 4000          | 300~1500000 | 750             |

### 17.4.2 为什么限制 Topic 的总分区数？

Kafka以分区为粒度管理消息，分区多导致生产、存储、消费都碎片化，影响性能稳定性。在使用过程中，当Topic的总分区数达到上限后，用户就无法继续创建Topic。

不同规格配置的Topic总分区数不同，如下表所示。

表 17-11 Kafka 集群实例规格

| 规格名称                 | 代理个数范围 | 单个代理TPS | 单个代理分区上限 | 单个代理建议消费组数 | 单个代理客户端总连接数上限 | 存储空间范围 (GB) | 单个代理流量规格 (MB/s) |
|----------------------|--------|---------|----------|------------|---------------|-------------|-----------------|
| kafka.2u4g.cluster   | 3~30   | 30000   | 250      | 20         | 2000          | 300~300000  | 100             |
| kafka.4u8g.cluster   | 3~30   | 100000  | 500      | 100        | 4000          | 300~600000  | 200             |
| kafka.8u16g.cluster  | 3~50   | 150000  | 1000     | 150        | 4000          | 300~1500000 | 375             |
| kafka.12u24g.cluster | 3~50   | 200000  | 1500     | 200        | 4000          | 300~1500000 | 625             |
| kafka.16u32g.cluster | 3~50   | 250000  | 2000     | 200        | 4000          | 300~1500000 | 750             |

### 17.4.3 Kafka 支持减少分区数吗？

Kafka不支持减少分区数，您可以通过删除原先的Topic，然后创建新Topic，重新设置分区数。

### 17.4.4 Kafka 实例创建 Topic 失败

可能原因：已创建的Topic，分区数之和达到实例规格的分区数上限。不同规格实例配置的分区数上限不同，具体请参考[Kafka实例规格](#)。

解决方案：对Kafka实例扩容，或者删除不需要的Topic。

### 17.4.5 Kafka 实例支持批量导入 Topic 功能么？或者是自动生成 Topic 功能？

支持自动生成Topic功能，但不支持Topic批量导入功能，仅支持批量导出Topic功能。

通过以下任意一种方法，开启自动生成Topic功能：

- 创建实例时，开启Kafka自动创建Topic。
- 创建实例后，在实例详情页开启Kafka自动创建Topic。

## 17.4.6 Kafka 实例是否支持查看单个 Topic 占用磁盘空间?

支持。通过以下任意一种方法，查看单个Topic占用磁盘空间大小。

- 在Kafka实例名称所在行，单击“查看监控数据”，跳转到云监控页面。在“主题”页签中，“主题”选择待查看磁盘空间大小的Topic名称，“监控类型”选择“基本监控”，查看“队列数据容量”，该指标表示该队列当前的消息数据大小。
- 单击Kafka实例名称，进入实例详情页。在左侧导航栏选择“监控 > 监控详情”，进入监控页面。在“主题”页签中，“主题”选择待查看磁盘空间大小的Topic名称，“监控类型”选择“基本监控”，查看“队列数据容量”，该指标表示该队列当前的消息数据大小。

## 17.4.7 Topic 是否支持 ACL 权限配置?

Kafka实例已开启Kafka SASL\_SSL功能，此时Topic支持配置ACL权限。在Kafka控制台的“实例管理 > Topic管理”页面，在需要设置用户权限的Topic所在行，单击“用户权限”，为用户设置不同的权限。

具体操作请参考[设置Kafka Topic权限](#)。

## 17.4.8 消息被消费后，没有删除，导致 Kafka 存储空间占满?

消息被消费后，并不会被删除，只有超过老化时间，才会被删除。

您可以通过减小老化时间或者扩容存储空间，解决此问题。

## 17.4.9 如何扩总分区?

增加基准带宽/代理数量/代理规格，可以扩大总分区数。具体操作请参考[变更Kafka集群实例规格](#)。

## 17.4.10 修改自动创建 Topic 的配置，会触发重启吗?

开启或者关闭“Kafka自动创建Topic”，可能会导致Kafka重启，具体以控制台显示为准。

## 17.4.11 消费者消费 Topic 失败，提示没有权限?

**问题现象：**同一个消费组内有多个消费者，为每个消费者授权不同的Topic访问权限，某一消费者消费其中一个Topic时，提示消费失败，报错信息如下：Not authorized to access topics。

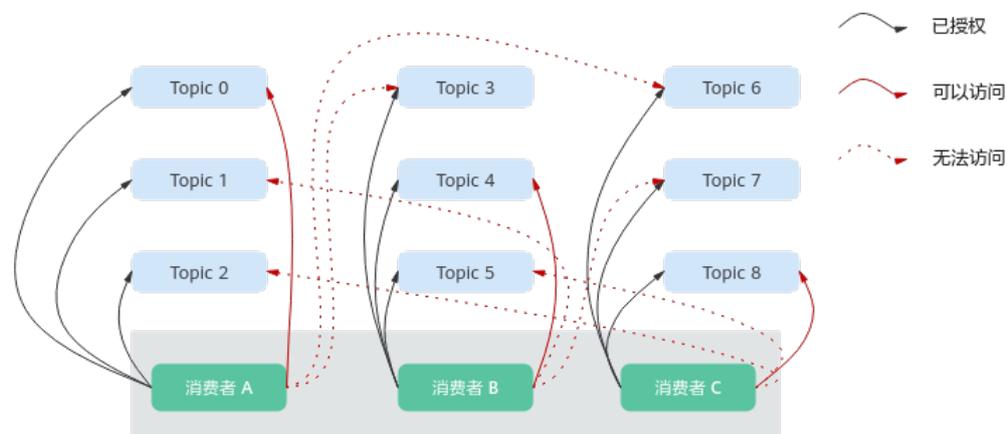
```
Tests passed: 1 of 1 test - 40 s 771 ms
"C:\Program Files\Huanwei\jdk1.8.0_242\bin\java.exe" ...
the numbers of topic:0
org.apache.kafka.common.errors.TopicAuthorizationException: Not authorized to access topics: [huanwei-gongce-basic, topic-huanwei-data]
```

**问题原因：**消费组的leader在进行分区分配时，不会考虑某一个消费者的授权和订阅信息，只会根据消费组整体的订阅情况进行分区分配，此种情况下可能会给消费者分配到未授权的Topic，从而导致了上述问题的出现。

例如：消费组中有消费者A、B、C，A订阅并授权Topic 0、Topic 1、Topic 2，B订阅并授权Topic 3、Topic 4、Topic 5，C订阅并授权Topic 6、Topic 7、Topic 8，假设以上

Topic都只有一个分区，消费组的leader会根据策略进行分区分配，分配的结果可能变成：A消费Topic 0、Topic 3、Topic 6，B消费Topic 1、Topic 4、Topic 7，C消费Topic 2、Topic 5、Topic 8。此时A对Topic 3和Topic 6是没有授权的，因此会出现“Not authorized to access topics”的报错。

图 17-5 消费者访问权限



#### 处理方法：

- 如果业务要求所有消费者在同一个消费组内，即group.id相同，解决方法：为所有消费者授权相同的Topic访问权限。
- 如果消费者不需要在同一个消费组内，解决方法：修改group.id，让每个消费者单独在一个消费组内。

### 17.4.12 分区数可以单独买吗？

Topic分区数不能单独购买，您可以通过增加基准带宽/代理数量/代理规格，实现增加总分区数。具体操作请参考[变更Kafka集群实例规格](#)。

## 17.5 消费组问题

### 17.5.1 Kafka 实例是否需要创建消费组、生产者和消费者？

- “auto.create.groups.enable”为“true”时，不需要单独创建消费组、生产者和消费者，在使用时自动生成，实例创建后，直接使用即可。
- “auto.create.groups.enable”为“false”时，需要手动创建消费组，不需要单独创建生产者和消费者。

修改“auto.create.groups.enable”的方法，请参考[修改Kafka实例配置参数](#)。

连接Kafka实例后，生产消息和消费消息，请参考[向Kafka实例生产消息和消费消息](#)。

### 17.5.2 如果消息组中没有在线的消费者（如 empty 状态），是否 14 天后会自动被删除？

消息组中没有在线的消费者（如empty状态），14天后是否会自动被删除与offsets.retention.minutes、auto.create.groups.enable参数有关：

- 很久前创建的实例，`auto.create.groups.enable`默认为true，消费组多久后会自动被删除取决于`offsets.retention.minutes`，`offsets.retention.minutes`支持在控制台修改，具体修改方法请参考[修改Kafka实例配置参数](#)。
- 新创建的实例，分为以下几种情况：
  - `auto.create.groups.enable`为false时，消费组不会被删除，如果需要删除消费组，您可以手动删除。
  - `auto.create.groups.enable`为true时，如果消费组中从未提交过offset，消费组十分钟后自动被删除。
  - `auto.create.groups.enable`为true时，如果消费组中提交过offset，消费组多久后会自动被删除取决于`offsets.retention.minutes`，`offsets.retention.minutes`支持在控制台修改，具体修改方法请参考[修改Kafka实例配置参数](#)。

**原因如下：** Kafka通过`offsets.retention.minutes`参数控制消费组中offsets保留时间，在此时间内如果没有提交offset，offsets将会被删除。Kafka判定消息组中没有在线的消费者（如empty状态），且没有offsets时，将会删除此消费组。

### 17.5.3 为什么删除消费组不生效？

**可能原因：** 您开启了自动创建消费组功能，且您的业务正在连接该消费组消费消息。所以，如果没有停止您的业务，删除了消费组后，消费消息时还是会自动创建该消费组。

**解决办法：** 关闭自动创建消费组功能，即在Kafka控制台的“实例管理 > 配置参数”页面，把“`auto.create.groups.enable`”设置为“false”，然后再删除消费组。部分实例在“实例管理 > 配置参数”中没有“`auto.create.groups.enable`”参数，具体以控制台为准，此时建议修改业务代码，使业务代码不再连接该消费组，然后再删除消费组。修改“`auto.create.groups.enable`”参数的方法请参考[修改Kafka实例配置参数](#)。

### 17.5.4 实例消费正常，为什么看不到消费者？

检查是否使用Flink进行消费。Flink使用的是assign模式，消费者定义在客户端，因此在Kafka实例控制台看不到消费者。

### 17.5.5 Kafka 可以删除消费组下不用的 Topic 吗？

在Kafka客户端取消订阅该Topic，即可达到在消费组下删除该Topic的效果。

### 17.5.6 为什么创建的消费组不见了？

如果您发现已创建的消费组在控制台查询不到这一现象，可能的原因如下：

- “`auto.create.groups.enable`”设置为“true”，消费组的状态为“EMPTY”且从未提交过offset，系统将在消费组创建成功的十分钟后自动删除该消费组。
- 消费组从未提交过offset，当Kafka实例重启后，该消费组会被删除。

## 17.6 消息问题

### 17.6.1 Kafka 生产消息的最大长度是多少？

生产消息的最大长度为10MB。

## 17.6.2 为什么 Kafka 实例常常发生 Rebalance（重平衡），导致消息拉取失败？

消费组的Rebalance就是对Topic分区的重新分配。

正常情况下消费组内加入新的消费者或老的消费者退出都会导致Rebalance，这种情况是无法避免的。但是某些特殊情况下，消费者会被服务端认为异常从而被踢出消费组，此时可能会导致消费异常，需要重点关注。

消费者被服务端认为异常从而被踢出消费组的场景如下：

### 1. 未能及时发送心跳请求。

消费者以设置的heartbeat.interval.ms为间隔向broker发送心跳请求，如果broker在session.timeout.ms时间内没有收到消费者的心跳请求，broker会认为消费者异常，从而将其从消费组中踢出，然后开始新一轮的Rebalance。

### 2. 消费者消费时间间隔过长。

消费者每次最多消费max.poll.records条消息，多数情况下客户端都会把一次消费到的数据处理完后才会开始下一次消费，如果单次消费的消息太多导致无法在max.poll.interval.ms时间内处理完或消息处理流程发生了异常（如需要写入后端数据库，后端数据库压力太大，慢SQL，时延增加等）导致消费时间增加，在max.poll.interval.ms时间内消费者没有发起下一次消费请求，broker认为消费者不活跃而将其踢出消费组，然后开始新一轮的Rebalance。

## 解决方法/排查思路

### 场景一：未能及时发送心跳请求

**排查思路：**如果是Java语言客户端，建议排查GC日志是否存在长时间FullGC的情况。该场景可能会导致心跳线程阻塞，服务端检查心跳失败，进入Rebalance状态。

**解决方法：**如果存在长时间FullGC的情况，需要您排查客户端相关问题（FullGC可能是客户端存在内存泄露）。

### 场景二：消费者消费时间间隔过长

#### 排查思路：

1. 检查单条消息的处理时间是多久，处理max.poll.records条消息会不会超过max.poll.interval.ms时间。
2. 消息处理流程是否有网络行为，如写数据库、调用后端API等，在发生Rebalance的场景下消费者下游系统是否正常。

**解决方法：**建议在消费者客户端将max.poll.records值减小，如果消息处理时间较长可以适当增加max.poll.interval.ms。

## 17.6.3 为什么 Console 页面上，消息查询查不到消息？

- **可能原因1：**消息已被老化。

**解决方法：**修改老化时间。

- **可能原因2：**消息的createTime时间戳不对。

Console页面是根据时间查询的，所以查不到。时间戳是由客户端生成，不同客户端有不同的处理策略，有的客户端默认值会是0或者-1，则查询不到消息。

**解决方法：**检查客户端消息的createTime设置是否正确。

- **可能原因3:** 磁盘容量超过95%，且“容量阈值策略”设置为“自动删除”。  
“容量阈值策略”设置为“自动删除”，表示磁盘容量达到95%时，系统会删除最早的10%的消息，以保证磁盘容量充足。当磁盘容量超过95%时，未到达老化时间的消息也会被删除，所以可能会导致部分消息查询不到。  
**解决方法:** 修改容量阈值策略或扩大磁盘容量。“容量阈值策略”设置为“生产受限”，表示一旦磁盘使用达到容量阈值95%，会导致后续生产失败，但保留了当前磁盘中的数据，直至数据自然老化。该场景适用于对数据不能丢的业务场景，但是会导致生产业务失败。

## 17.6.4 Kafka 消息堆积了怎么办？

**问题现象:** 实例的监控指标“消息堆积数”产生了告警。

**处理方法:**

1. 在Kafka实例控制台，单击产生告警的实例名称，进入实例详情页。
2. 在左侧导航栏，单击“监控 > 监控详情”，进入监控页面。
3. 在“消费组”页签中，查看“消息堆积数（消费组可消费消息数）”，找出消息堆积的消费组。
4. 在左侧导航栏，单击“实例管理 > 消费组管理”，进入消费组列表页面。
5. 查看消息堆积的消费组是否有消费者在消费，如果有，让业务方加快消费效率，如果没有，让客户酌情删掉不使用的消费组。

## 17.6.5 消息超过老化时间，消息仍存在的原因

**问题现象:** 消息超过设置的老化时间（如果Topic已设置老化时间，则“实例管理 > 配置参数”中的log.retention.hours值对此Topic不生效。仅当Topic未设置老化时间时，该参数值才会生效。），消息仍存在。

**可能原因1:** Topic的每个分区都是由多个大小相同的segment文件组成，每个segment文件的大小为500MB，当segment文件存储的消息大小到达500MB后，才会新建下一个segment文件。Kafka删除消息是删除segment文件，而不是删除一条消息。Kafka要求至少保留一个segment文件用来存储消息，如果正在使用的segment文件中包含超过老化时间的消息，由于此时segment文件不会被删除，所以超过老化时间的消息也不会被删除。

**处理方法:** 等待segment文件被使用完，或者删除超过老化时间的消息所在的Topic。

**可能原因2:** Topic中存在一条create time为未来时间的消息（例如当前时间为1月1日，create time设置成了2月1日），此消息在72小时后，并不会被老化，导致在此消息后创建的其他消息都不会被老化。

**处理方法:** 删除create time为未来时间的消息所在的Topic。

## 17.6.6 Kafka 实例是否支持延迟消息？

不支持延迟消息。

## 17.6.7 如何查看堆积消息数？

通过以下任意一种方法，查看堆积消息数。

- 在Kafka控制台的“实例管理 > 消费组管理”页面，单击待查看堆积消息的消费组名称，进入消费组详情页。在“消费进度”页签，查看消费组中每个Topic的总堆积数。具体步骤，请参考[查询消费进度](#)。

- 在Kafka控制台的“监控 > 监控详情”页面的“消费组”页签中，“消费组”选择待查看堆积消息数的消费组名称，“主题”选择“全部Topic”，“消费组可消费消息数”表示此消费组中所有Topic的堆积消息数之和。查看监控数据的具体步骤，请参考[查看Kafka监控数据](#)。
- 在云监控页面的“消费组”页签中，“消费组”选择待查看堆积消息数的消费组名称，“主题”选择“全部Topic”，“消费组可消费消息数”表示此消费组中所有Topic的堆积消息数之和。查看监控数据的具体步骤，请参考[查看Kafka监控数据](#)。
- 在Kafka客户端的“/bin”目录下，通过以下命令查看消费组中每个Topic的堆积消息数。

**未开启SASL:**

```
./kafka-consumer-groups.sh --bootstrap-server {connection-address} --describe --group {group-name}
```

**已开启SASL:**

```
./kafka-consumer-groups.sh --bootstrap-server {connection-address} --describe --group {group-name} --command-config {ssl-user-config.properties}
```

表 17-12 参数说明

| 参数名称                       | 说明                                     |
|----------------------------|----------------------------------------|
| connection-address         | Kafka实例的连接地址，在Kafka控制台的“概览 > 连接信息”中获取。 |
| group-name                 | 消费组名称。                                 |
| ssl-user-config.properties | 配置文件的名称，此配置文件用于存放用户名和密码，以及SSL证书配置信息。   |

图 17-6 查看每个 Topic 的总堆积数

```
[root@vm-246f011-kafka-share-zk-server-2 bin]# ./kafka-consumer-groups.sh --bootstrap-server 172.17.137.137 --group console-consumer-54209 --describe
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/dms/version/2.7/kafka_2.13-2.7.1/libs/slf4j-log4j12-1.7.25.jar/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/dms/version/2.7/kafka_2.13-2.7.1/libs/slf4j-log4j12-1.7.26.jar/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

GROUP             TOPIC             PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG             CONSUMER-ID                                     HOST              CLIENT-ID
console-consumer-54209 test2             0          -                0                0              consumer-console-consumer-54209-1-b88da1b5-4664-4f52-836a-73105901160d /172.31.7.137    consumer-console-consumer-54209-1
console-consumer-54209 test2             1          -                0                0              consumer-console-consumer-54209-1-b88da1b5-4664-4f52-836a-73105901160d /172.31.7.137    consumer-console-consumer-54209-1
console-consumer-54209 test2             2          -                0                0              consumer-console-consumer-54209-1-b88da1b5-4664-4f52-836a-73105901160d /172.31.7.137    consumer-console-consumer-54209-1
```

返回信息中的“LAG”表示每个Topic的总堆积数。

### 17.6.8 为什么消息创建时间显示 1970?

消息创建时间是由生产客户端在生产消息时通过CreateTime指定的，如果生产消息时没有设置此参数，消息创建时间会默认为1970。

### 17.6.9 如何修改 message.max.bytes 参数?

在Kafka控制台的“实例管理 > 配置参数”页面修改message.max.bytes参数，具体步骤请参见[修改Kafka实例配置参数](#)。

message.max.bytes参数最大只能修改为10485760字节，如果您实际的业务场景中单条消息的大小超过此值，建议使用压缩算法或者业务逻辑做切分。

### 17.6.10 为什么 offset 不连续?

在生产者客户端中开启幂等或事务，然后生产消息，此时您会在消费者客户端或Kafka控制台的“实例管理 > 消息查询”中观察到消息offset不连续的现象。这是因为开启了

幂等或事务后，在生产消息时会产生一些元数据控制消息，这些控制消息也会生产到该Topic中，且它们对消费者不可见，从而造成offset不连续的现象。

Kafka生产者客户端3.0及之后的版本会默认开启幂等，如果不需要使用该功能，请将“enable.idempotence”设置为“false”。

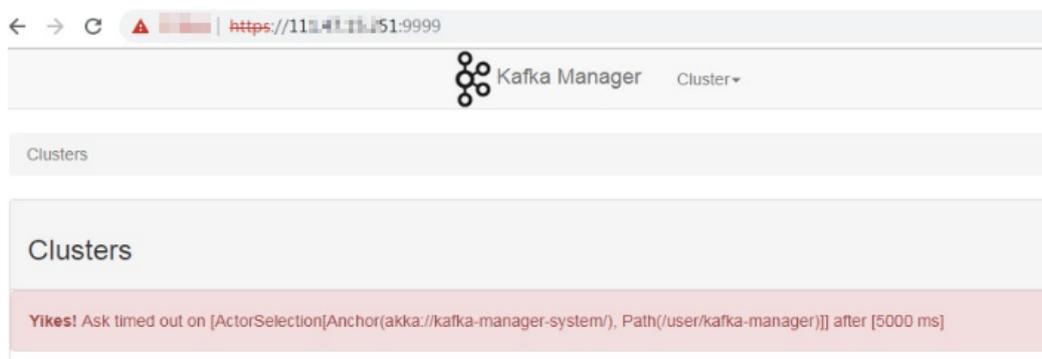
## 17.7 Kafka Manager 问题

### 17.7.1 登录 Kafka Manager 的账号是否可以设置为只读账号？

不支持设置为只读账号。

### 17.7.2 登录到 Kafka Manager 页面，为什么获取不到节点信息？

**问题现象：**正常登录Kafka Manager页面，但是页面无法显示节点信息，如下所示：



**问题原因：**这是Kafka开源问题，需要联系客服，重启Kafka Manager。

### 17.7.3 Yikes! Insufficient partition balance when creating topic : projectman\_project\_enterprise\_project Try again.

**问题现象：**

在Kafka Manager中无法创建topic，报错：Yikes! Insufficient partition balance when creating topic : projectman\_project\_enterprise\_project Try again.

**问题原因：**Topic分区超过限制，不能继续创建Topic。

**处理方法：**建议扩大实例规格，实例规格增加，分区数也会相应增加。

### 17.7.4 Kafka Manager 能否查询到消息的正文？

Kafka Manager无法查询到消息的正文。

### 17.7.5 Kafka Manager WebUI 的端口能否修改？

Kafka Manager WebUI的端口不支持修改。

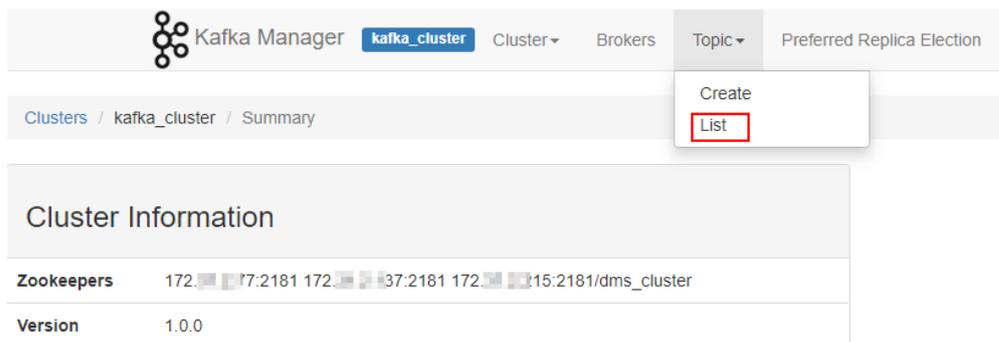
### 17.7.6 在 Kafka Manager 上支持修改 Topic 的哪些属性？

在Kafka Manager上，支持修改如下的Topic属性：max.message.bytes、segment.index.bytes、segment.jitter.ms、min.cleanable.dirty.ratio、

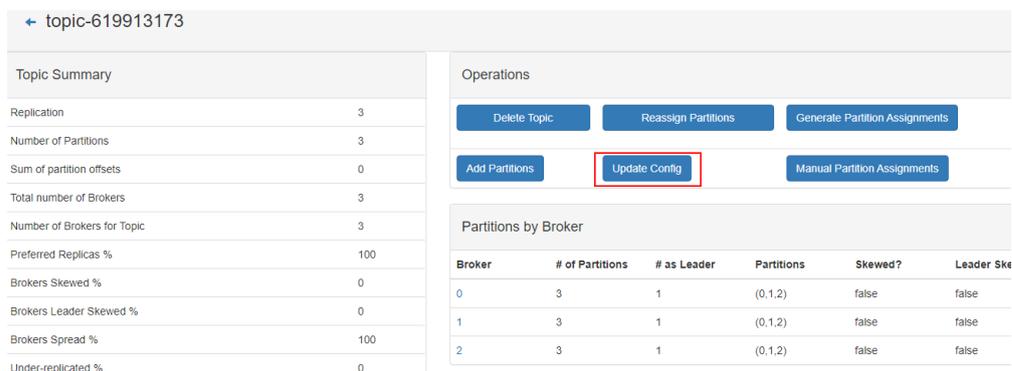
retention.bytes、file.delete.delay.ms、compression.type、flush.ms、cleanup.policy、unclean.leader.election.enable、flush.messages、retention.ms、min.insync.replicas、delete.retention.ms、preallocate、index.interval.bytes、segment.bytes、segment.ms。

具体修改方法如下：

1. [登录Kafka Manager](#)。
2. 单击“kafka\_cluster”，进入“Summary”界面。
3. 单击“Topic > List”，进入“Topics”界面。



4. 单击需要修改属性的Topic名称，进入Topic详情界面。
5. 单击“Update Config”，修改Topic属性。



### 17.7.7 Kafka Manager 和云监控显示的信息不一致

**问题现象：**在后台删除某个消费组后，在云监控已经不显示此消费组，但是Kafka Manager中依然存在。

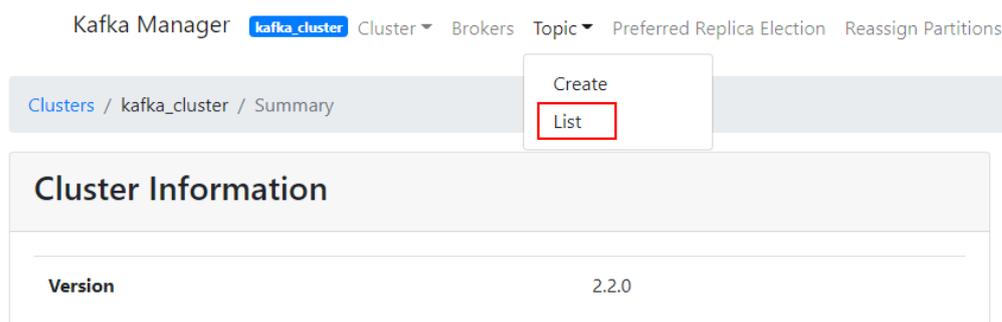
**问题原因：**Kafka Manager存在缓存。

**处理方法：**登录Kafka控制台，在实例所在行，单击“更多 > 重启Manager”。

### 17.7.8 Kafka Manager 如何修改 Topic 的分区 Leader?

在Kafka Manager中修改Topic的分区Leader的步骤如下：

1. [登录Kafka Manager](#)。
2. 选择“Topic > List”，进入Topic列表页面。



- 单击待修改分区Leader的Topic名称（以topic-test为例介绍），进入Topic详情页面。

### Topics

Show 10 entries

| Topic                             | # Partitions | # Brokers | Brokers Spread % | Brokers Skew % | Brokers Leader Skew % |
|-----------------------------------|--------------|-----------|------------------|----------------|-----------------------|
| <a href="#">_consumer_offsets</a> | 50           | 3         | 100              | 0              | 0                     |
| <a href="#">_trace</a>            | 9            | 3         | 100              | 0              | 0                     |
| <a href="#">topic-test</a>        | 3            | 3         | 100              | 0              | 0                     |

Showing 1 to 3 of 3 entries

- 单击“Manual Partition Assignments”，进入修改分区Leader页面。

图 17-7 Topic 详情页面

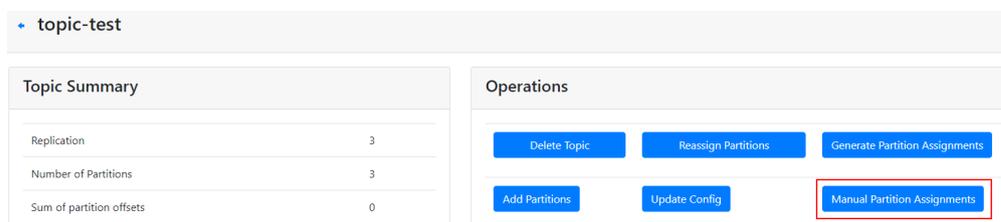
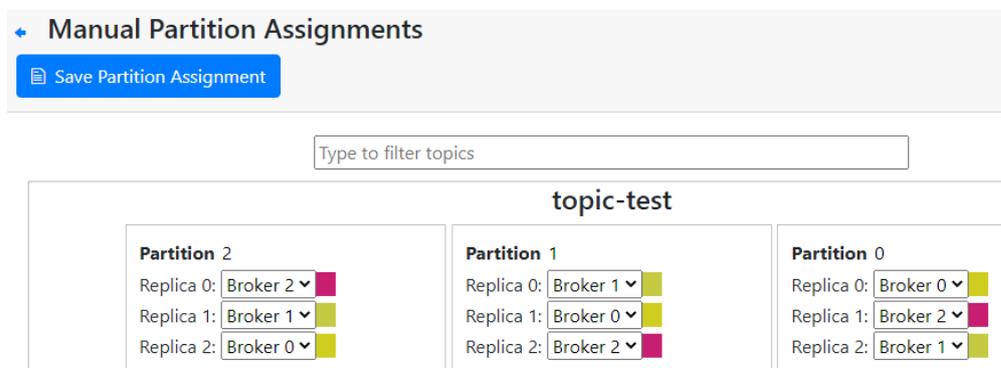
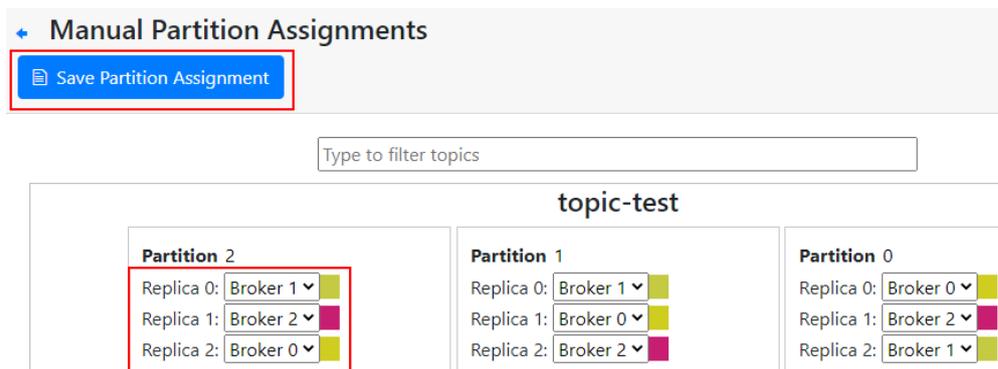


图 17-8 修改分区 Leader 页面

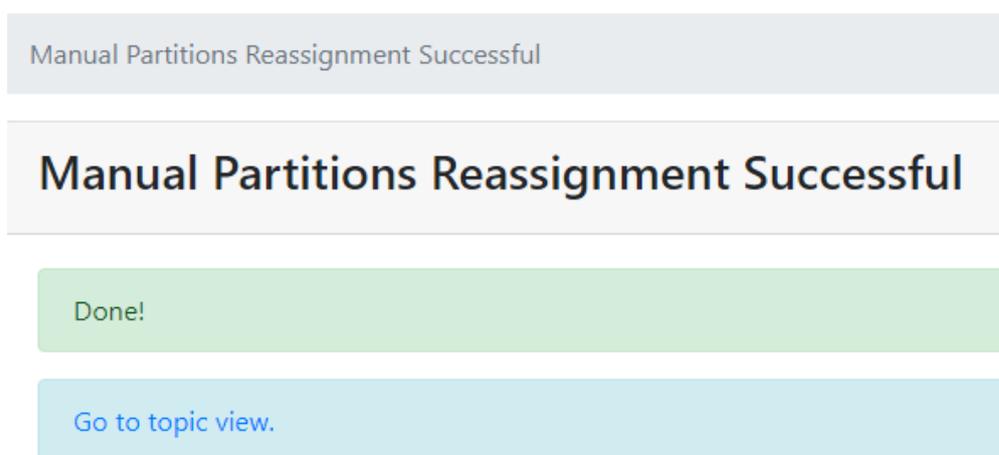


以修改分区2的Leader为例介绍（Replica 0即为分区Leader），如上图所示，分区2的Leader为2。

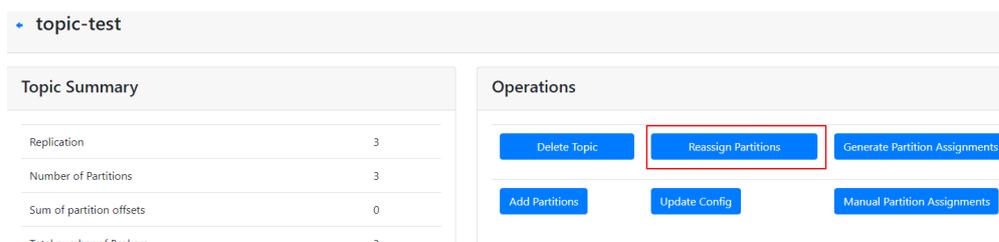
5. 修改分区Leader，单击“Save Partition Assignment”。



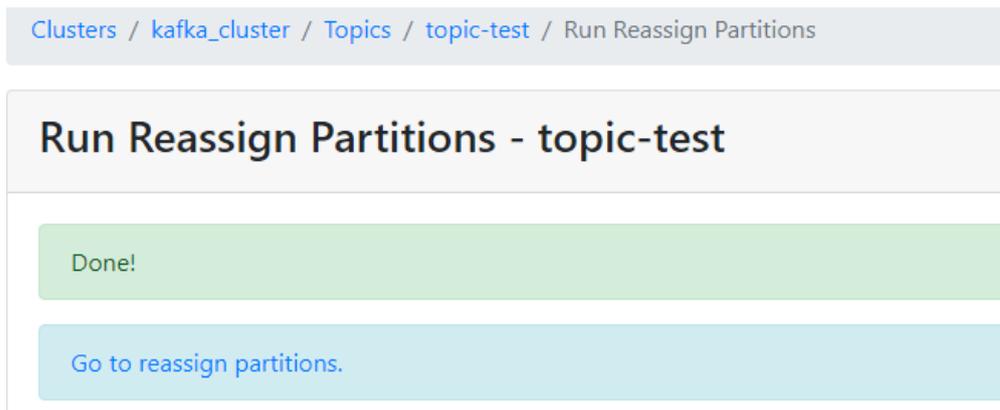
修改成功后，返回如下图所示信息。



6. 单击“Go to topic view”，进入Topic详情页面。
7. 单击“Reassign Partitions”，保存对Topic分区Leader的修改。



保存成功后，返回如下图所示信息。



8. 在导航栏单击Topic名称，进入Topic详情页面，查看分区详情。

| Partition Information |               |        |          |                  |
|-----------------------|---------------|--------|----------|------------------|
| Partition             | Latest Offset | Leader | Replicas | In Sync Replicas |
| 0                     | 0             | 0      | (0,2,1)  | (0,2,1)          |
| 1                     | 0             | 1      | (1,0,2)  | (1,0,2)          |
| 2                     | 0             | 1      | (1,2,0)  | (2,1,0)          |

如上图所示，分区2的Leader已经从2改为1。

### 17.7.9 实例版本在控制台和 Kafka Manager 上显示不一致?

Kafka实例的版本以控制台为准。Kafka Manager使用了开源Kafka 2.2.0的公共信息配置，所以显示的版本是2.2.0，和购买的Kafka实例版本没有关系。

### 17.7.10 为什么实例中存在默认名为\_\_trace 和\_\_consumer\_offsets 的 Topic?

**问题现象：** Kafka Manager中存在默认名为\_\_trace和\_\_consumer\_offsets的Topic。

## Topics

Show  entries

| Topic                              | ↑↓ # Partitions | ↑↓ # Brokers | ↑↓ Brokers Spread % |
|------------------------------------|-----------------|--------------|---------------------|
| <a href="#">__consumer_offsets</a> | 50              | 3            | 100                 |
| <a href="#">__trace</a>            | 9               | 3            | 100                 |
| <a href="#">topic-01</a>           | 3               | 3            | 100                 |

**处理方法：**\_\_trace和\_\_consumer\_offsets是Kafka实例内部预留的Topic，不建议删除这两个Topic，删除后可能导致实例无法使用。

### 17.7.11 客户端删除消费组后，在 Kafka Manager 中仍可以看到此消费组？

客户端删除消费组后，此消费组已经被删除了。在Kafka Manager中仍可以看到此消费组，是因为Kafka Manager存在缓存。

通过以下任意一种方法解决此问题：

- 重启Kafka Manager。
- Kafka Manager只显示14天内有消费记录的消费组，如果您不想重启Kafka Manager，可以等待14天后Kafka Manager自动清除此消费组。

## 17.8 监控告警问题

### 17.8.1 云监控无法展示 Kafka 监控数据

Topic监控数据无法展示，可能原因如下：

- Topic名称开头包含特殊字符，例如#号“#”。
- 在云监控中特殊字符“.”按5个字符计算长度。如果Topic名称包含“.”，且经此规则计算后的总长度超过256个字符。
- Kafka实例中没有创建Topic。

解决方法如下：

- 删除带特殊字符的Topic。
- 删除带特殊字符“.”的Topic。
- 创建Topic。

消费组监控数据无法展示，可能原因如下：

- 消费组名称开头包含特殊字符，例如#号“#”。
- 在云监控中特殊字符“.”按5个字符计算长度。如果消费组名称包含“.”，且经此规则计算后的总长度超过256个字符。
- 此消费组从未有消费者连接。

解决方法如下：

- 删除带特殊字符的消费组。
- 删除带特殊字符“.”的消费组。
- 使用此消费组消费消息。

云监控中特殊字符点“.”按5个字符计算长度。如果遇到其他监控数据无法显示时，建议优先排查监控对象的名称是否包含“.”，且经此规则计算后的总长度超过256个字符。

## 17.8.2 Kafka 监控显示消息堆积数跟实例里的消息数不一致？

**问题现象：**监控显示消息堆积数为8.1亿+，Kafka控制台显示实例中6个Topic的消息数总和为1亿+，两者不一致。

**问题结论：**两者统计方式不同，Kafka控制台显示的消息数为实例中未消费的消息个数，而监控显示的消息堆积数=Topic中的消息积压数\*消费组数。

## 17.8.3 Kafka 的消费组删除了，怎么监控页面还可以看到这个消费组？

监控数据是每分钟进行采集上报，上报的数据经过整理后才会显示在监控页面上，此过程大约需要几分钟到十几分钟，建议您在删除消费组后，过一段时间再去监控页面查看。

## 17.8.4 为什么磁盘读/写流量、磁盘平均读/写操作耗时和 CPU 使用率出现明显波动？

磁盘读流量、磁盘写流量、磁盘平均读操作耗时、磁盘平均写操作耗时和CPU使用率这几个监控指标采集的是瞬时值，仅作为系统资源评估参考。它们出现明显波动通常情况下是由于Kafka数据采用异步落盘会消耗磁盘I/O和CPU导致的，这种波动不会对业务产生影响。

## 17.8.5 为什么 JVM 堆内存使用率出现明显波动？

JVM堆内存使用率在监控页面呈现锯齿状属于正常现象，在扩容规格或升级Kafka实例内核版本后，可能导致JVM堆内存使用率最大值升高、波动幅度变大，这种波动属于JVM堆内存回收的正常机制，对业务无影响。

## 17.8.6 生产消息数不为 0，为什么生产速率为 0？

消息生产速率的原始指标数据为每分钟上报一次，但是它的单位为“个/秒”，所以消息生产速率=一分钟内生产的消息数/60，其值向下取整。当一分钟内生产的消息数小于60个时，生产速率会显示为0。

## 17.8.7 消费进度显示有堆积消息，为什么监控中显示消息堆积数为0？

### 问题现象

在消费组详情页的“消费进度”页签中，显示Topic有堆积消息（即总堆积数不为0），但是在消费组监控页面显示“消息堆积数（消费组可消费消息数）”为0。

### 问题结论

因为“消费进度”页签中总堆积数和消费组监控页面中“消息堆积数（消费组可消费消息数）”的采样周期不同。

“消费进度”页签中总堆积数是采集的瞬时值，而消费组监控页面中“消息堆积数（消费组可消费消息数）”的采集周期是一分钟。导致以上问题的原因为您在“消费进度”中查看到的是当前时间点的堆积消息数，而在一分钟内消费者消费完了堆积消息，所以“消息堆积数（消费组可消费消息数）”显示为0。

# 18 故障排除

## 18.1 Kafka 连接问题排查和解决

### 概述

本章节主要描述Kafka连接问题的排查和解决。

### 问题分类

当您发现与Kafka实例连接出现异常时，可以根据本文的内容，从以下几个方面进行排查。

- [检查网络](#)
- [检查Consumer和Producer的配置](#)
- [客户端连接问题-Java客户端](#)
- [客户端连接问题-Go客户端](#)

### 检查网络

在连接Kafka实例前，需要确保客户端与Kafka实例之间可以相互连接，如果无法连接，建议先检查网络连接是否正常。

已开启sasl\_ssl访问的Kafka实例为例，执行如下命令：

```
curl -kv {ip}:{port}
```

- 网络正常，如下：

```
[root@ecs-5d2f ~]# curl -kv 192.168.0.52:9093
* Rebuilt URL to: 192.168.0.52:9093/
* Trying 192.168.0.52...
* TCP_NODELAY set
* Connected to 192.168.0.52 (192.168.0.52) port 9093 (#0)
> GET / HTTP/1.1
> Host: 192.168.0.52:9093
> User-Agent: curl/7.61.1
> Accept: */*
>
Warning: Binary output can mess up your terminal. Use "--output -" to tell
Warning: curl to output it to your terminal anyway, or consider "--output
Warning: <FILE>" to save to a file.
* Failed writing body (0 != 7)
* Closing connection 0
```

- 网络异常/不通，如下：

```
[root@ecs-5d2f ~]# curl -kv 192.168.0.52:9093
* Rebuilt URL to: 192.168.0.52:9093/
* Trying 192.168.0.52...
* TCP_NODELAY set
* connect to 192.168.0.52 port 9093 failed: Connection timed out
* Failed to connect to 192.168.0.52 port 9093: Connection timed out
* Closing connection 0
curl: (7) Failed to connect to 192.168.0.52 port 9093: Connection timed out
```

解决方案：

1. 检查是否同一个VPC。如果不是，请参考[跨VPC访问](#)。
2. 检查安全组规则是否配置正确。请参考[如何选择和配置安全组](#)。

## 检查 Consumer 和 Producer 的配置

查看日志，检查Consumer和Producer初始化时打印的参数是否符合预期，是否和自己在配置文件中设置的一致。

如果不一致，请检查配置文件的写法。

## Java 客户端常见报错

- 未关闭域名校验

出现下图的错误信息：

```
Caused by: java.lang.Thread.run(Thread.java:748)
Caused by: javax.net.ssl.SSLHandshakeException: General SSLEngine problem
    at sun.security.ssl.Alerts.getSSLException(Alerts.java:192)
    at sun.security.ssl.SSLEngineImpl.fatal(SSLEngineImpl.java:1789)
    at sun.security.ssl.Handshaker.fatalISE(Handshaker.java:318)
    at sun.security.ssl.Handshaker.fatalISE(Handshaker.java:318)
    at sun.security.ssl.ClientHandshaker.serverCertificate(ClientHandshaker.java:1639)
    at sun.security.ssl.ClientHandshaker.processMessage(ClientHandshaker.java:223)
    at sun.security.ssl.Handshaker.processLoop(Handshaker.java:1837)
    at sun.security.ssl.Handshaker$1.run(Handshaker.java:978)
    at sun.security.ssl.Handshaker$1.run(Handshaker.java:967)
    at java.security.AccessController.doPrivileged(Native Method)
    at sun.security.ssl.Handshaker$DelegatedTask.run(Handshaker.java:1459)
    at org.apache.kafka.common.network.SslTransportLayer.runDelegatedTasks(SslTransportLayer.java:482)
    at org.apache.kafka.common.network.SslTransportLayer.handshakeUnwrap(SslTransportLayer.java:484)
    at org.apache.kafka.common.network.SslTransportLayer.doHandshake(SslTransportLayer.java:348)
    ... 7 more
Caused by: java.security.cert.CertificateException: No subject alternative names matching IP address 10.166.37.165 found
    at sun.security.util.HostnameChecker.matchIP(HostnameChecker.java:168)
    at sun.security.util.HostnameChecker.match(HostnameChecker.java:94)
    at sun.security.ssl.X509TrustManagerImpl.checkIdentity(X509TrustManagerImpl.java:462)
    at sun.security.ssl.X509TrustManagerImpl.checkIdentity(X509TrustManagerImpl.java:442)
    at sun.security.ssl.X509TrustManagerImpl.checkTrusted(X509TrustManagerImpl.java:261)
    at sun.security.ssl.X509TrustManagerImpl.checkServerTrusted(X509TrustManagerImpl.java:144)
    at sun.security.ssl.ClientHandshaker.serverCertificate(ClientHandshaker.java:1626)
    ... 16 more
(kafka.admin.TopicCommand$)
```

解决方案：检查consumer.properties和producer.properties文件，ssl.endpoint.identification.algorithm参数必须设置为空，关闭域名校验开关。

```
ssl.endpoint.identification.algorithm=
```

- 加载SSL证书文件失败

出现下图的错误信息：

```
[2020-05-28T06:35:38.654][ERROR][logstash.outputs.kafka ] Unable to create Kafka producer from given configuration {:kafka_error_message=>org.apache.kafka.common.KafkaException: Failed to load SSL keystore /opt/cloud/logstash/openssl/ssl/openssl.cnm-north-4/client-truststore.jks of type jks}
```

解决方案：

- a. 请检查对应地址的client.truststore.jks文件是否存在。
- b. 检查进程和文件所属权限。
- c. 检查consumer.properties和producer.properties文件中的ssl.truststore.password参数是否配置正确。

ssl.truststore.password为服务器证书密码，不可更改，需要保持为dms@kafka

```
ssl.truststore.password=dms@kafka
```

- Topic名称错误

出现下图的错误信息：

```
020-05-11 01:11:23,584 INFO [eventpull-thread30] [impl.KafkaClientImpl 267] ... ready poll, topic is CSBPromotionManagementService_PromotionTopic
020-05-11 01:11:23,784 INFO [eventpull-thread30] [kafka.pull.callback] [771] pull event from kafka cost time 200, topic CSBPromotionManagementService_PromotionTopic, eventList []
020-05-11 01:11:24,629 ERROR [PublishEventToKafka-Thread] [impl.KafkaClientImpl 268] send event to kafka failed, topic=[CSBPromotionCouponService_CouponTopic], eventId = [01780-99999]
#####54d8c5-01 ex w
# qps@kafka:~/kafka/common/errors: TimeoutException: Topic=CSBPromotionCouponService_CouponTopic not present in metadata after 60000 ms.
020-05-11 01:11:24,717 INFO [pool-20-thread-1] [impl.KafkaClientImpl 100] ready.getTopicList
020-05-11 01:11:24,724 INFO [pool-20-thread-1] [impl.KafkaClientImpl 107] getTopicList cost time 6
020-05-11 01:11:24,724 INFO [pool-20-thread-1] [impl.KafkaClientImpl 113] end.getTopicList
020-05-11 01:11:24,863 INFO [eventpull-thread34] [impl.KafkaClientImpl 267] ... ready poll, topic is CSBPromotionCouponService_CouponTopic
```

解决方案：用户重新创建Topic或者打开自动创建Topic开关。

## Go 客户端常见报错

使用GO客户端无法连接Kafka SSL，报“first record does not look like a TLS handshake”错误。

解决方案：go默认是关闭的，需要打开 TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256加密套件。

## 18.2 消息从生产到消费时延高达 6 分钟

### 问题现象

消息从生产到消费的端到端时延偶现到达6分钟，业务对消息时延较为敏感。

### 分析过程

1. 怀疑是业务堆积，处理慢导致时延。  
通过查看监控数据，发现业务请求并不是很多，堆积最多时也只有50条消息，同时每秒新增消息只有10条左右，应该还没有到达处理极限。
2. 排查EIP流量情况，发现入流量存在下降的场景。  
联系EIP服务技术人员协助排查，未发现问题。
3. 从业务日志中分析消费组行为。  
通过查看服务端日志，消费组存在大量rebalance动作，大部分rebalance都会秒级完成，但偶尔会有分钟级别的rebalance耗时，而rebalance过程中是无法正常消费的，只有在rebalance动作完成才可以进行消费。  
该现象与问题现象描述的偶现长时间时延行为相吻合，问题确定。

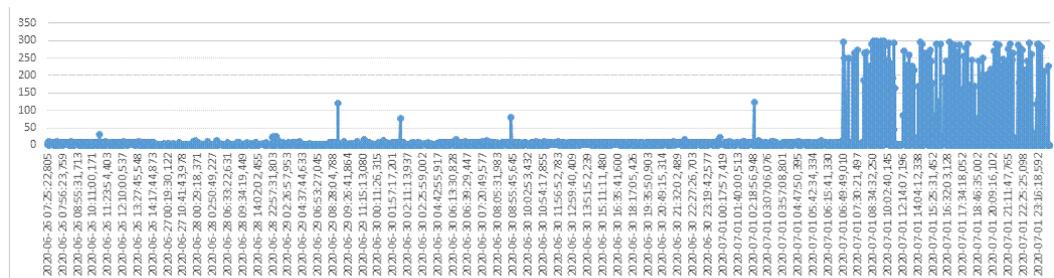
### 详细分析

查看用户消费组行为日志文件，文件中存在以下三种日志：

- Preparing to rebalance group 1  
表示消费组开始进行rebalance，消费组状态变更为REBALANCING。
- Stabilized group  
表示消费组rebalance完成，消费组状态变更为STABILIZED。
- Member consumer-xxx in group 1 has failed  
表示消费组内的某一个消费者超时退出，一般触发场景是消费者长时间没有与服务端通信导致，比如处理消息耗时太久阻塞等。

对文件中每次Preparing到Stabilized完成的时间进行计算得到以下结果图。图中时间为UTC+0时间。

图 18-1 消费组 rebalance 图



从以上图中可以看出，消费组rebalance的性能在7月1号06:49以后存在明显劣化的情况，导致客户端异常。

## 根因描述

用户业务中偶尔会存在某一个消费者无法及时响应rebalance的动作，导致整个消费组消费阻塞，一直到该消费者响应rebalance动作为止。

## 问题规避

1. 建议用户根据业务区分不同消费组使用，降低单消费者阻塞导致的影响访问。
2. `max.poll.interval.ms`用于设置消费组请求消费消息的最大时间间隔，如果消费者在超时前没有发起下一次消费请求，服务端会触发rebalance。调大`max.poll.interval.ms`的默认配置，降低问题频率。

## 问题解决措施

1. 区分业务使用不同消费组。
2. 需要客户侧排查自身业务，优化自身业务处理逻辑，提高处理效率，降低阻塞时间。

## 背景知识介绍

消费组可以简单认为有两种状态REBALANCING和STABILIZED。

- REBALANCING：消费组元数据发生变化，该状态下消费组中的所有消费者都**无法进行正常的业务消费**，该场景触发场景为消费组内有新的消费者加入或有已经建立连接的消费者退出。
- STABILIZED：rebalance完成，消费组处于稳定状态，该状态下消费组中的消费者**可以进行正常的业务消费**，触发条件是，当前消费组内的所有消费者都同步完成新的消费组元数据，包括之前已经同步过的消费者，也需要重新同步。

消费组简单流程如下：

1. 有新的消费者加入或退出，服务端记录的消费组元数据更新，服务端更新消费组进入REBALANCING状态。
2. 服务端等待所有消费者（包含已有的消费者）同步最新的元数据。
3. 所有消费者同步完最新的元数据后，服务端更新消费组状态为STABILIZED。
4. 消费者开始正常的消费业务。

## 18.3 生产消息失败

### 问题现象

系统提示“Disk error when trying to access log file on the disk”错误。

### 根因描述

节点磁盘使用率超高导致生产消息失败。

### 问题解决措施

扩容实例的磁盘空间，具体操作请参考[变更Kafka集群实例规格](#)。

## 18.4 删除 Topic 失败

### 问题现象

删除Topic不生效，删除后该Topic仍然存在。

### 根因描述

实例开启了自动创建Topic功能，且有消费者正在连接该Topic。如果没有停止业务，删除Topic后，还会有消息生产行为，并自动创建Topic。

### 问题解决措施

关闭Kafka自动创建Topic功能，才可以正常删除Topic。关闭方法请参考[配置Kafka自动创建Topic](#)。

## 18.5 在 Windows 服务器中无法登录 Kafka Manager

### 问题现象

在Windows服务器的浏览器中输入Kafka Manager地址，无法登录Kafka Manager，报错信息如下图。



## 根因描述

1. Windows服务器和Kafka实例不在同一个VPC同一个子网内，或者安全组配置不正确，导致网络无法连通。
2. Kafka Manager异常。

## 问题解决措施

1. 检查Windows服务器和Kafka实例是否在同一个VPC同一个子网内？
  - 是，执行2。
  - 否，修改Windows服务器的VPC和子网，与Kafka实例一致。
2. 检查安全组是否配置正确？安全组的配置请参考[如何选择和配置安全组](#)。
  - 是，执行3。
  - 否，修改安全组配置。
3. 在Kafka实例控制台，重启Kafka Manager，具体操作请参考[重启Kafka Manager](#)。

## 18.6 生产或消费消息时，报 Topic `{{topic_name}}` not present in metadata after 60000 ms 错误

### 问题现象

多可用区的Kafka实例，在某个可用区故障后，Kafka客户端在生产或消费消息时，可能会报Topic `{{topic_name}}` not present in metadata after 60000 ms的错误，如下图所示。

```
ssl.secure.random.implementation = null
ssl.trustmanager.algorithm = PKIX
ssl.truststore.location = null
ssl.truststore.password = null
ssl.truststore.type = JKS
transaction.timeout.ms = 60000
transactional.id = null
value.serializer = class org.apache.kafka.common.serialization.StringSerializer
(org.apache.kafka.clients.producer.ProducerConfig)
[2021-10-29 15:44:44,141] INFO Kafka version: 2.3.0 (org.apache.kafka.common.utils.AppInfoParser)
[2021-10-29 15:44:44,141] INFO Kafka commitId: fclaa116b661c8a (org.apache.kafka.common.utils.AppInfoParser)
[2021-10-29 15:44:44,141] INFO Kafka startTimeMs: 1635493484139 (org.apache.kafka.common.utils.AppInfoParser)
[2021-10-29 15:45:44,146] ERROR produce message failed. error msg: Topic topic-test not present in metadata after 60000 ms. (org.example
.Producer)
[2021-10-29 15:46:44,247] ERROR produce message failed. error msg: Topic topic-test not present in metadata after 60000 ms. (org.example
.Producer)
[2021-10-29 15:46:51,418] WARN [Producer clientId=producer-1] Connection to node -3 (/100.85.120.91:9094) could not be established. Brok
er may not be available. (org.apache.kafka.clients.NetworkClient)
[2021-10-29 15:46:51,684] INFO [Producer clientId=producer-1] Cluster ID: t0R4RgFHTN2pjUhiJqkFPQ (org.apache.kafka.clients.Metadata)
[2021-10-29 15:46:51,733] INFO produce message success. partition: 1, offset: 9335 (org.example.Producer)
[2021-10-29 15:46:51,809] INFO produce message success. partition: 4, offset: 9336 (org.example.Producer)
[2021-10-29 15:46:51,920] INFO produce message success. partition: 5, offset: 9335 (org.example.Producer)
[2021-10-29 15:46:52,005] INFO produce message success. partition: 2, offset: 9336 (org.example.Producer)
[2021-10-29 15:46:52,112] INFO produce message success. partition: 3, offset: 9327 (org.example.Producer)
[2021-10-29 15:46:52,206] INFO produce message success. partition: 8, offset: 9324 (org.example.Producer)
[2021-10-29 15:46:52,308] INFO produce message success. partition: 9, offset: 9332 (org.example.Producer)
[2021-10-29 15:46:52,410] INFO produce message success. partition: 6, offset: 9332 (org.example.Producer)
[2021-10-29 15:46:52,508] INFO produce message success. partition: 7, offset: 9335 (org.example.Producer)
[2021-10-29 15:46:52,608] INFO produce message success. partition: 0, offset: 9335 (org.example.Producer)
[2021-10-29 15:46:52,709] INFO produce message success. partition: 1, offset: 9336 (org.example.Producer)
[2021-10-29 15:46:52,808] INFO produce message success. partition: 4, offset: 9337 (org.example.Producer)
```

## 问题解决措施

以下三种措施都可以解决此问题，请根据实际情况任意选择一种。

- 升级Kafka客户端的版本到2.7或以上版本，并设置“socket.connection.setup.timeout.ms”大于1s，且小于“request.timeout.ms/Kafka服务端节点数”。
- 修改Kafka客户端的“request.timeout.ms”大于“127s”。
- 修改Kafka客户端Linux系统的网络参数“net.ipv4.tcp\_syn\_retries”为“3”。

## 18.7 Flink 1.15 消费进度提交失败

### 问题现象

Flink 1.15版本消费Kafka时，消费进度提交失败，持续报错：COORDINATOR\_NOT\_AVAILABLE。

### 根因描述

Flink 1.15版本使用的Kafka客户端存在bug，当某次消费进度提交失败后，客户端将coordinator置为不可用且无法自动恢复。

### 问题解决措施

- 重启Flink job。
- 升级Flink至1.16及以上版本。

# A 修订记录

| 发布日期       | 修订记录                                                                                                                                                                                                                                                                                                                                                                       |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2025-08-29 | 本次变更如下： <ul style="list-style-type: none"><li>● 新增IPv6功能，主要涉及<a href="#">购买Kafka实例</a>章节。</li><li>● 新增一键购买相同配置实例功能，主要涉及<a href="#">购买Kafka实例</a>章节。</li><li>● 控制台导航栏优化，主要涉及<a href="#">创建Kafka Topic</a>章节。</li></ul>                                                                                                                                                      |
| 2025-04-30 | 本次变更如下： <ul style="list-style-type: none"><li>● 新增<a href="#">查看Kafka Topic日志</a>、<a href="#">查看Kafka重平衡日志</a>、<a href="#">关闭Kafka Manager</a>和<a href="#">配置Kafka实例回收站策略</a>章节。</li><li>● 新增Key查询，主要涉及<a href="#">查看Kafka消息</a>章节。</li><li>● 购买页面改版，并新增“内网IP地址”参数，主要涉及<a href="#">购买Kafka实例</a>章节。</li><li>● 新增Topic批量授权，主要涉及<a href="#">设置Kafka Topic权限</a>章节。</li></ul> |
| 2025-01-24 | 本次变更如下： <ul style="list-style-type: none"><li>● 新增Kafka 3.x版本，主要涉及<a href="#">购买Kafka实例</a>、<a href="#">Kafka集群实例</a>和<a href="#">Kafka服务端支持版本是多少？</a>章节。</li><li>● 新增扩容代理规格，主要涉及<a href="#">变更Kafka集群实例规格</a>章节。</li><li>● 新增跨VPC访问，主要涉及<a href="#">使用VPCEP实现跨VPC访问Kafka</a>章节。</li></ul>                                                                                 |
| 2024-06-26 | 本次变更如下：<br>新增标签策略功能，修改 <a href="#">购买Kafka实例</a> 和 <a href="#">配置Kafka实例标签</a> 章节。                                                                                                                                                                                                                                                                                         |

| 发布日期       | 修订记录                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2024-05-28 | <p>本次变更如下：</p> <ul style="list-style-type: none"> <li>● 新增按关键字查询消息，主要涉及<a href="#">查看Kafka消息</a>章节。</li> <li>● 新增定时分区平衡，主要涉及<a href="#">修改Kafka分区平衡</a>章节。</li> <li>● 创建用户时，用户数上限增加到500，主要涉及<a href="#">配置Kafka ACL用户</a>章节。</li> <li>● 在控制台新增创建消费组、生产消息和删除消息功能，主要涉及<a href="#">创建Kafka消费组</a>、<a href="#">使用控制台连接Kafka</a>和<a href="#">删除Kafka消息</a>。</li> <li>● Topic的老化时间上限增加到720小时，主要涉及<a href="#">创建Kafka Topic</a>章节。</li> </ul>                                                                                                                |
| 2023-05-09 | <p>本次变更如下：</p> <ul style="list-style-type: none"> <li>● 新增<a href="#">快速入门</a>。</li> <li>● 新增<a href="#">查看Kafka磁盘使用量</a>、<a href="#">修改Kafka分区平衡</a>、<a href="#">查看和重置Kafka消费进度</a>和<a href="#">配置流控</a>等章节。</li> <li>● 实例详情页改版，主要涉及<a href="#">重启Kafka实例</a>、<a href="#">删除Kafka实例</a>和<a href="#">使用客户端连接Kafka（明文接入）</a>等章节。</li> <li>● 新增SASL机制，主要涉及<a href="#">购买Kafka实例</a>和<a href="#">使用客户端连接Kafka（密文接入）</a>章节。</li> <li>● 购买实例时，使用新版实例规格，主要涉及<a href="#">Kafka集群实例</a>、<a href="#">购买Kafka实例</a>和<a href="#">变更Kafka集群实例规格</a>章节。</li> </ul> |
| 2022-10-27 | <p>本次变更如下：</p> <ul style="list-style-type: none"> <li>● 新增<a href="#">故障排除</a>。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 2022-08-12 | 第一次正式发布。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |