



引擎用户手册

金蝶Apusic分布式消息队列V2.0.6_for_kafka

版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

商标声明

 是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。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。

目录

- 1 前言
 - 1.1 适用对象
 - 1.2 相关文档
 - 1.3 技术支持
- 2 核心引擎目录说明
- 3 Broker 配置说明
 - 3.1 基础配置 (Basic Configuration)
 - 3.2 副本与可靠性 (Replication & Reliability)
 - 3.3 日志保留策略 (Log Retention)
 - 3.4 性能调优 (Performance Tuning)
 - 3.5 ADCC for zk 连接 (ADCC for zk Connection)
 - 3.6 安全 (Security)
 - 3.7 其他 (Others)
- 4 ADCC for zk 配置说明
 - 4.1 基础配置 (Basic Timing)
 - 4.2 数据存储 (Data Storage)
 - 4.3 连接信息 (Connection Info)
 - 4.4 集群配置 (Cluster Config)
 - 4.5 自动清理 (Auto Purge)
 - 4.6 TCP 网络优化 (Network Tuning)
 - 4.7 监控和管理 (Admin & 4lw)
 - 4.8 其他配置 (Others)
- 5 常见命令
 - 5.1 Topic 管理
 - 5.1.1 创建 Topic
 - 5.1.2 查看 Topic 列表
 - 5.1.3 查看 Topic 详情 (分区、副本分布)
 - 5.1.4 增加分区数 (只能增加, 不能减少)
 - 5.1.5 删除 Topic
 - 5.2 生产与消费 (数据测试)
 - 5.2.1 启动控制台生产者 (Producer)
 - 5.2.2 启动控制台消费者 (Consumer)

- 5.3 消费者组管理 (Consumer Groups)
 - 5.3.1 查看消费者组列表
 - 5.3.2 查看组内详情 (延迟、Offset)
 - 5.3.3 重置 Offset (慎用)
- 5.4 集群与元数据管理
 - 5.4.1 查看集群 Broker 信息
 - 5.4.2 首选副本选举 (Preferred Leader Election)
 - 5.4.3 查看集群配置
- 6 常见问题
 - 6.1 常见错误排查

1 前言

本文档为金蝶Apusic分布式消息队列for Kafka（Apusic Distributed Message Queue，简称：ADMQ for Kafka）消息引擎的用户手册，详细介绍了ADMQ for Kafka消息引擎的功能使用、配置方法及管理操作等内容。

1.1 适用对象

本文档适用于IT信息化业务负责人、研发经理、软件项目经理、软件架构师、运维工程师。

1.2 相关文档

了解更多ADMQ for Kafka产品相关的信息，请参阅以下ADMQ for Kafka产品手册文档集：

序号	手册文档	说明
1	金蝶Apusic分布式消息队列for Kafka快速使用手册	简单介绍了如何快速上手使用ADMQ for Kafka。
2	金蝶Apusic分布式消息队列for Kafka安装手册	详细介绍如何在各操作系统上安装ADMQ for Kafka，以及ADMQ for Kafka服务启停等操作。
3	金蝶Apusic分布式消息队列for Kafka消息引擎用户手册	详细介绍 ADMQ for Kafka 消息引擎相关功能的使用、配置、管理及配套工具的使用方法。
4	金蝶Apusic分布式消息队列for Kafka管控台用户手册	详细介绍ADMQ for Kafka管控台相关功能的使用和操作说明。
5	金蝶Apusic分布式消息队列for Kafka开发手册	详细介绍基于各开发语言进行ADMQ for Kafka客户端应用开发的说明。
6	金蝶Apusic分布式消息队列for Kafka迁移手册	详细介绍从Kafka迁移到ADMQ for Kafka的说明。
7	金蝶Apusic分布式消息队列for Kafka运维手册	详细介绍ADMQ for Kafka的监控、运维、安全加固等运维说明。
8	金蝶Apusic分布式消息队列for Kafka性能优化手册	详细介绍ADMQ for Kafka性能调优的说明。

1.3 技术支持

ADMQ for Kafka产品提供全面的技术支持服务，您可以通过以下方式获得技术支持：

- 网址: www.apusic.com
- 电话: 400-855-5800
- 邮箱: support@apusic.com
- 金蝶云社区: <https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

您在取得技术支持时, 请提供如下信息:

1. 您的姓名
2. 公司与联系方式
3. 操作系统及其版本
4. 产品版本号
5. 出现异常及错误的日志、截图等详细信息

2 核心引擎目录说明

目录名	说明
bin	存放启动、停止及管理脚本
config	存放配置文件 (如 <code>kafka-standalone.conf</code>)
data	存放 Kafka 消息数据及 ADCC for zk 数据
jdk	内置的 Java 运行环境, 无需额外配置系统 JDK
logs	存放运行日志 (Broker 日志、ADCC for zk 日志等)
service	系统服务注册相关文件 (可选)

3 Broker 配置说明

3.1 基础配置 (Basic Configuration)

核心作用：定义 Broker 的身份、网络暴露方式及存储路径。

参数	说明	生产环境建议
<code>broker.id</code>	集群唯一标识。 Kafka 通过此 ID 在 ADCC for zk 中注册自己。	必须唯一。 若重启后 ID 改变，Broker 会被视为新节点，导致旧数据不可见或集群元数据混乱。通常写死在配置文件中，不要动态生成。
<code>listeners</code>	监听地址。 定义 Broker 绑定哪些 IP 和端口，以及使用什么协议。格式：协议://IP:端口。	- 内网部署 ：PLAINTEXT://0.0.0.0:9092 (绑定所有网卡) 或指定内网 IP。 - 多网卡/混合云 ：可配置多个，如 INTERNAL://192.168.1.10:9092,EXTERNAL://10.0.0.5:9093。
<code>advertised.listeners</code>	广播地址。 客户端连接时，Broker 返回给客户端的地址。 至关重要。	- 如果客户端和 Broker 不在同一网络（如 Docker、云服务器）， 必须 配置此项为客户端可访问的公网 IP 或域名。 - 若不配，默认取 <code>listeners</code> 的值，可能导致客户端连接内网 IP 失败。
<code>log.dirs</code>	数据存储目录。 消息日志文件存放路径。	- 强烈建议使用高性能 SSD。 - 可配置多个目录（逗号分隔）以利用多块磁盘 IO，如 <code>/data1/kafka,/data2/kafka</code> 。
<code>num.partitions</code>	默认分区数。 创建 Topic 时未指定分区数时的默认值。	建议设置为 3~6（与集群 Broker 数量匹配或略多），以提高并行消费能力。

<code>auto.create.topics.enable</code>	自动创建 Topic。当生产者向不存在的 Topic 发消息时，是否自动创建。	**生产环境强烈建议设为 <code>****false****</code>** 。原因：防止因代码拼写错误创建大量无用 Topic，且自动创建的 Topic 参数（分区/副本）往往不符合生产标准。
--	---	--

3.2 副本与可靠性 (Replication & Reliability)

核心作用：保证数据不丢失、服务高可用。

参数	说明	生产环境建议
<code>default.replication.factor</code>	默认副本数 。新建 Topic 时的默认副本数量。	建议 ≥ 3 。允许同时挂掉 2 台机器而不丢失数据。需确保集群 Broker 数量 $\geq$ 该值。
<code>min.insync.replicas</code>	最小同步副本数 。配合 <code>acks=all</code> 使用，表示写入成功至少需要多少个副本确认。	- 若设为 1：只要 Leader 写入成功即返回，Leader 挂掉可能丢数据。 - 推荐设为 2 (配合 3 副本)：保证即使挂掉 1 台，仍有 1 个完整副本存活，且写入时需 2 个节点成功，兼顾安全与可用。
<code>offsets.topic.replication.factor</code>	<code>__consumer_offsets</code> 内部 Topic 的副本数。	必须 ≥ 2 (最好 3)。若该 Topic 挂了，所有消费者无法提交位移，导致重复消费或消费停滞。
<code>replica.lag.time.max.ms</code>	ISR 踢出阈值 。Follower 落后 Leader 超过此时间，将被移出 ISR 列表。	默认 30s。网络抖动大时可适当调大，但过大会导致故障切换变慢。
<code>unclean.leader.election.enable</code>	非 ISR 选举 。允许不在 ISR 中的副本（数据落后的）成为 Leader。	必须设为 <code>****false****</code> (默认) 。若设为 <code>true</code> ：当所有 ISR 挂掉，强制让数据落后的副本上位，会导致数据丢失。生产环境严禁开启。
<code>controller.quorum.voters</code>	控制器投票节点 。仅用于 KRaft 模式 (无 ADCC for zk)。	ADCC for zk 模式下留空。若迁移到 KRaft 模式，需配置 Controller 节点的 ID 和地址。

3.3 日志保留策略 (Log Retention)

核心作用：控制磁盘空间使用，清理旧数据。

参数	说明	生产环境建议
log.retention.hours	按时间保留。日志文件保留的小时数。	默认 168h (7天)。根据业务需求调整，如合规要求保留 30 天则设为 720。
log.retention.bytes	按大小保留。每个 Partition 最大保留的字节数。	注释该配置则默认为 -1 (不限制)。建议设置上限（如 50GB），防止单 Topic 写爆磁盘。
log.segment.bytes	段文件大小。日志切分的大小阈值。	默认 1GB。太大导致清理粒度粗，太小导致文件句柄过多。一般保持默认即可。
log.retention.check.interval.ms	检查间隔。后台线程检查是否要删除旧日志的频率。	默认 5 分钟。无需频繁修改。

3.4 性能调优 (Performance Tuning)

核心作用：平衡吞吐量与延迟，适配硬件资源。

参数	说明	生产环境建议
num.network.threads	网络线程数。处理 Socket 读写、编解码的线程数。	默认 8。高并发场景（如万级 TPS）可调至更大或与 CPU 核数匹配。
num.io.threads	IO 线程数。处理磁盘读写请求的线程数。	默认 16。建议设置为 磁盘数量的 2 倍 或 CPU 核数，如 16。
socket.send/receive.buffer.bytes	Socket 缓冲区。发送/接收缓冲区大小。	默认 1024KB。高吞吐场景可适当调大（如 2MB），依赖 OS 内核参数配合。
log.flush.interval.messages/ms	刷盘策略。多少条消息或多少毫秒强制刷盘到磁盘。	默认值极大 (即不主动强制刷盘)。 最佳实践 ：依赖 OS 的 Page Cache 异步刷盘，性能最高。仅在极度追求数据零丢失且能接受性能大幅下降时才调小此值。

3.5 ADCC for zk 连接 (ADCC for zk Connection)

核心作用：连接元数据管理中心（仅限 ADCC for zk 模式）。

参数	说明	生产环境建议
<code>zookeeper.connect</code>	ADCC for zk 地址列表。格式 <code>host:port</code> ，多节点用逗号分隔。	填写所有 ADCC for zk 节点地址。支持 chroot 路径。
<code>zookeeper.session.timeout.ms</code>	会话超时。Broker 与 ADCC for zk 心跳丢失多久判定为下线。	默认 18s-30s。调小可加快故障发现，但 网络抖动易导致误判重平衡；调大则故障 恢复慢。
<code>zookeeper.chroot.path</code>	根路径。将 Kafka 元数据存储 在 ADCC for zk 的某个子路径 下。	多集群共用 ADCC for zk 时必配。例如 <code>/kafka-cluster-01</code> ，避免不同 Kafka 集群元数据冲突。

3.6 安全 (Security)

核心作用：认证与加密（注释部分为示例）。

参数	说明	生产环境建议
<code>security.inter.broker.protocol</code>	Broker 间通信 协议。	生产环境建议启用 <code>SASL_SSL</code> 或至少 <code>SASL_PLAINTEXT</code> ，防止内部窃听或伪造节点。
<code>ssl.enabled.mechanisms</code>	认证机制。	常用 <code>PLAIN</code> (简单账号密码), <code>SCRAM-SHA-256/512</code> (更安全), <code>GSSAPI</code> (Kerberos)。
<code>listener.name...jaas.config</code>	JAAS 配置。	定义用户名密码。建议将敏感信息移至独立的 JAAS 文 件，而非直接写在 <code>properties</code> 中。

3.7 其他 (Others)

参数	说明	生产环境建议
<code>delete.topic.enable</code>	删除 Topic 开 关。	**建议 ** <code>true</code> ** 。方便运维清理测试数据或废弃业务线数据。若为 <code>false</code> ，删除命令仅标记无效，不释放磁盘。
<code>JMX_PORT</code>	监控端口。	必须配置 。用于 Prometheus/JConsole 采集指标。需在启动 脚本中通过 <code>-D</code> 参数设置，并注意防火墙放行。

<code>replica.fetch.wait.max.ms</code>	Follower 拉取等待时间。	默认 500ms。适当调大可减少 Follower 对 Leader 的空轮询压力，但会增加同步延迟。一般保持默认。
--	-------------------------	--

4 ADCC for zk 配置说明

4.1 基础配置 (Basic Timing)

核心作用：定义心跳节奏和集群容错时间窗口，直接影响故障检测速度。

参数	说明	生产环境建议
tickTime	心跳基准时间 (ms)。ADCC for zk 内部所有时间计算的基础单位。	默认 **2000ms ** (2秒)。- 设太小：网络抖动易导致误判节点下线。- 设太大：故障检测慢。 建议保持 2000 ，除非网络极差可调至 3000-4000。
initLimit	初始化同步时限 。Follower 启动时从 Leader 同步数据的最大 tickTime 倍数。	默认 10 (即 20秒)。若数据量大或磁盘慢，需调大 (如 20)，否则 Follower 启动会超时失败。
syncLimit	心跳同步时限 。Follower 与 Leader 心跳响应的最大 tickTime 倍数。	默认 5 (即 10秒)。超过此时间未响应，Leader 认为该 Follower 已死，将其踢出集群。 网络不稳定时可适当调大至 8-10 。

4.2 数据存储 (Data Storage)

核心作用：控制数据落盘策略，直接影响 IO 性能和磁盘寿命。

参数	说明	生产环境建议
dataDir	快照数据目录 。存储内存数据库的快照 (snapshot) 和 myid 文件。	必须配置在高性能 SSD 上 。不要与事务日志 (dataLogDir) 混用，避免 IO 争抢。
dataLogDir	事务日志目录 。存储写入操作的事务日志 (transaction log)。	关键性能点： 1. 务必独立挂载磁盘 (与 dataDir 物理隔离)。2. ADCC for zk 对顺序写要求高，独立的 SSD 能极大提升吞吐量。
preAllocSize	预分配文件大小 (KB)。事务日志文件预分配的大小。	默认 65536 (64MB)。配置中设为 131072 (128MB)。 作用： 减少文件系统碎片和频繁的文件扩展系统调用。大文件有利于顺序写性能。
snapCount	快照触发频率 。每处理多少事务生成一次快照。	默认 100,000。配置中设为 200,000。 作用： 减少快照生成频率，降低磁盘 IO 压力。但会导致重启恢复时间变长 (因为要重放更多日志)。生产环境可适当调大。

<code>fsync.warningthresholdms</code>	刷盘警告阈值 (ms) 。当 <code>fsync</code> 耗时超过此值时记录警告日志。	用于监控磁盘健康状况。若频繁出现警告，说明磁盘 IO 瓶颈严重，需更换硬件或优化负载。
---------------------------------------	---	---

4.3 连接信息 (Connection Info)

核心作用：定义客户端接入点和并发控制。

参数	说明	生产环境建议
<code>clientPort</code>	客户端监听端口 。Kafka Broker 和 业务客户端 连接 ADCC for zk 的端口。	默认 2181。防火墙需开放此端口。
<code>maxClientCnxns</code>	单 IP 最大连接数 。限制单个 IP 地址能建立的最大连接数。	默认 60。配置中设为 0 (表示无限制)。 风险 ：若某台机器异常发起大量连接，可能耗尽 ADCC for zk 资源。 生产环境建议设置合理上限 (如 200-500)，防止 DDoS 或代码死循环。

4.4 集群配置 (Cluster Config)

核心作用：定义集群拓扑和节点角色。

参数	说明	生产环境建议
<code>server.x=A:B:C:D;E</code>	集群节点定义 。 <code>x</code> : MyID (对应 <code>myid</code> 文件)。 <code>A</code> : IP。 <code>B</code> : 2888 (Follower 与 Leader 通信)。 <code>C</code> : 3888 (选举通信)。 <code>D</code> : 角色 (participant/observer)。 <code>E</code> : 客户端端口 (可选，新版支持)。	- 奇数节点 ：集群节点数必须为奇数 (3, 5, 7)，以容忍 $(N-1)/2$ 台故障。 - Observer ：若需增加读性能而不增加投票负担，可将部分节点设为 <code>observer</code> 。
<code>electionPortBindRetry</code>	选举端口绑定重试次数 。	默认 3。若端口被占用，重试几次后退出。一般无需修改。

4.5 自动清理 (Auto Purge)

核心作用：防止磁盘被历史快照和日志填满。

参数	说明	生产环境建议
<code>autopurge.snapRetainCount</code>	保留快照数量。保留最近的 N 个快照文件。	配置中设为 3。建议：至少保留 3 个，以防最新快照损坏时有旧版本可回滚。
<code>autopurge.purgeInterval</code>	清理间隔 (小时)。每隔多久执行一次清理任务。	配置中设为 6 (小时)。建议：根据数据量调整。高写入场景可设为 1-2 小时，低负载可设为 12-24 小时。

4.6 TCP 网络优化 (Network Tuning)

核心作用：应对高并发连接和会话管理。

参数	说明	生产环境建议
<code>globalOutstandingLimit</code>	全局待处理请求上限。防止请求堆积导致 OOM。	默认 1000。配置中设为 5000。注意：调大可提高吞吐，但若后端处理慢，可能导致内存暴涨。需配合监控观察内存使用率。
<code>minSessionTimeout</code>	最小会话超时 (ms)。客户端请求的 timeout 低于此值将被强制设为此值。	默认 2 * tickTime。设太小会导致网络微抖动就断连。建议 ≥ 4000ms。
<code>maxSessionTimeout</code>	最大会话超时 (ms)。	默认 20 * tickTime。配置中设为 60000 (60秒)。允许长连接客户端设置更长的超时时间，减少重连频率。

4.7 监控和管理 (Admin & 4lw)

核心作用：运维监控与安全控制。

参数	说明	生产环境建议
<code>admin.enableServer</code>	是否开启 AdminServer (HTTP 接口)。	默认 true (新版)。配置中设为 false。安全建议：若不需要 HTTP 监控接口，建议关闭以减少攻击面。可通过 JMX 或 4lw 命令监控。
<code>4lw.commands.whitelist</code>	四字命令白名单。ZK 通过 `echo cmd`	nc ip 2181` 进行监控。

4.8 其他配置 (Others)

参数	说明	生产环境建议
skipACL	跳过 ACL 检查。	默认 no。若设为 yes，将忽略所有权限检查，大幅提升性能，但失去安全性。仅在完全可信的内网且追求极致性能时使用。
leaderServers	Leader 是否处理读请求。	配置中 no (可能是旧版本参数或特定发行版)。标准 ADCC for zk 中 Leader 默认处理所有写请求和部分读请求。若设为不参与服务，可能影响读吞吐。通常保持默认。
forceSync	是否强制每次事务都 fsync。	默认 no (使用 OS 缓冲)。若设为 yes，数据最安全但性能急剧下降。 **生产环境务必保持 ***no** 。
sslQuorum / portUnification	SSL 加密与端口统一。	若需启用 SSL 加密集群通信，需设为 true 并配置证书。明文内网通信可保持 false。

5 常见命令

5.1 Topic 管理

5.1.1 创建 Topic

创建一个名为 `test-topic` 的 Topic, 包含 3 个分区, 3 个副本。

```
kafka/bin/kafka-topics.sh --create \  
  --bootstrap-server 192.168.1.10:9092 \  
  --topic test-topic \  
  --partitions 3 \  
  --replication-factor 3
```

- 注意: `replication-factor` 不能超过集群中 Broker 的数量。

5.1.2 查看 Topic 列表

```
# 列出所有 Topic  
kafka/bin/kafka-topics.sh --list --bootstrap-server  
192.168.1.10:9092  
  
# 过滤特定 Topic (支持正则)  
kafka/bin/kafka-topics.sh --list --bootstrap-server  
192.168.1.10:9092 | grep "test"
```

5.1.3 查看 Topic 详情 (分区、副本分布)

这是排查数据倾斜和副本状态最重要的命令。

```
kafka/bin/kafka-topics.sh --describe \  
  --bootstrap-server 192.168.1.10:9092 \  
  --topic test-topic
```

输出解读:

- `Leader` : 当前负责读写的副本 ID。

- `Replicas` : 该分区所有副本所在的 Broker ID 列表。
- `Isr` (In-Sync Replicas): 当前与 Leader 保持同步的副本列表。如果 `Isr` 数量少于 `Replicas`, 说明有副本落后或宕机。

5.1.4 增加分区数 (只能增加, 不能减少)

将 `test-topic` 的分区数从 3 增加到 6。

```
kafka/bin/kafka-topics.sh --alter \  
  --bootstrap-server 192.168.1.10:9092 \  
  --topic test-topic \  
  --partitions 6
```

5.1.5 删除 Topic

```
kafka/bin/kafka-topics.sh --delete \  
  --bootstrap-server 192.168.1.10:9092 \  
  --topic test-topic
```

- 前提: 配置文件 `server.properties` 中 `delete.topic.enable=true`。

5.2 生产与消费 (数据测试)

5.2.1 启动控制台生产者 (Producer)

向 `test-topic` 发送消息。输入一行按回车即发送一条消息。

```
kafka/bin/kafka-console-producer.sh \  
  --bootstrap-server 192.168.1.10:9092 \  
  --topic test-topic
```

- 高级用法: 指定 Key 发送 (用于测试分区策略)

```
kafka/bin/kafka-console-producer.sh --bootstrap-server  
192.168.1.10:9092 --topic test-topic --property "parse.key=true" --
```

```
property "key.separator=:"
# 输入格式: key:value (例如 user1:login)
```

5.2.2 启动控制台消费者 (Consumer)

从 `test-topic` 消费消息。

- 消费最新数据:

```
kafka/bin/kafka-console-consumer.sh \
  --bootstrap-server 192.168.1.10:9092 \
  --topic test-topic
```

- **从头开始消费 (历史数据)**:

```
kafka/bin/kafka-console-consumer.sh \
  --bootstrap-server 192.168.1.10:9092 \
  --topic test-topic \
  --from-beginning
```

- 显示 Key 和 分区信息:

```
kafka/bin/kafka-console-consumer.sh \
  --bootstrap-server 192.168.1.10:9092 \
  --topic test-topic \
  --from-beginning \
  --property print.key=true \
  --property partition=true \
  --property print.timestamp=true
```

- 限制消费数量 (测试用):

```
kafka/bin/kafka-console-consumer.sh --bootstrap-server
192.168.1.10:9092 --topic test-topic --max-messages 10
```

5.3 消费者组管理 (Consumer Groups)

5.3.1 查看消费者组列表

```
kafka/bin/kafka-consumer-groups.sh \  
  --bootstrap-server 192.168.1.10:9092 \  
  --list
```

5.3.2 查看组内详情 (延迟、Offset)

运维最常用命令，用于查看消费积压 (Lag)。

```
kafka/bin/kafka-consumer-groups.sh \  
  --bootstrap-server 192.168.1.10:9092 \  
  --group my-consumer-group \  
  --describe
```

输出关键列：

- `CURRENT-OFFSET`：当前已提交的位置。
- `LOG-END-OFFSET`：分区最新的消息位置。
- `LAG`：**积压量** = `LOG-END-OFFSET` - `CURRENT-OFFSET`。如果 LAG 持续增大，说明消费速度跟不上生产速度。

5.3.3 重置 Offset (慎用)

将消费者组的 Offset 重置到最早、最新或指定时间。

- **重置到最早：**

```
kafka/bin/kafka-consumer-groups.sh --bootstrap-server  
192.168.1.10:9092 \  
  --group my-consumer-group \  
  --topic test-topic \  
  --reset-offsets --to-earliest \  
  --execute
```

- **重置到最新** (相当于忽略历史积压):

```
kafka/bin/kafka-consumer-groups.sh --bootstrap-server
192.168.1.10:9092 \
  --group my-consumer-group \
  --topic test-topic \
  --reset-offsets --to-latest \
  --execute
```

- **注意:** 执行重置时, 消费者组必须处于**非活跃状态** (即没有正在运行的消费者实例)。

5.4 集群与元数据管理

5.4.1 查看集群 Broker 信息

```
kafka/bin/kafka-broker-api-versions.sh --bootstrap-server
192.168.1.10:9092
```

5.4.2 首选副本选举 (Preferred Leader Election)

如果因为故障导致 Leader 分布不均匀 (例如所有 Leader 都在 Node 1), 可以触发重新选举, 让 Leader 回到 `Replicas` 列表中的第一个节点 (通常是理想分布)。

```
kafka/bin/kafka-leader-election.sh --bootstrap-server
192.168.1.10:9092 \
  --election-type preferred \
  --all-topic-partitions
```

5.4.3 查看集群配置

```
kafka/bin/kafka-configs.sh --bootstrap-server 192.168.1.10:9092 \
  --entity-type brokers \
  --entity-name 0 \
  --describe
```

6 常见问题

6.1 常见错误排查

- ADCC for zk 启动失败，日志报 **Connection refused** 或 **Session expired**
 - 检查 `myid` 文件是否存在且内容正确。
 - 检查 `zookeeper.properties` 中 `server.x` 的 IP 是否可达，端口 2888/3888 是否被防火墙拦截。
- Kafka 启动失败，日志报 **Broker ID 0 is already registered**
 - 检查是否有残留的旧 ADCC for zk 数据。如果是新部署，清空 ADCC for zk 的 `dataDir` 和 Kafka 的 `log.dirs` 后重试。
 - 确认各节点 `broker.id` 不重复。
- ISR 集合不全 (Isr: 0,1 而不是 0,1,2)
 - 说明某个 Follower 同步太慢或网络不通。检查该节点的日志和网络带宽。
- Kafka 启动失败，日志报** `Failed to bind to port 9092` **
 - 检查 9092 端口是否被其他进程占用（使用 `netstat -tulpn | grep 9092` 或 `lsof -i:9092` 命令）。
 - 检查 `server.properties` 中 `listeners` 配置的端口是否与实际要使用的端口一致，避免配置冲突。
 - 确认防火墙未拦截 9092 端口，或已在防火墙规则中放行该端口。
- Kafka 生产者发送消息失败，报 `Leader not available`
 - 检查对应的 Topic 是否存在，若不存在需先创建（`kafka-topics.sh --create` 命令）。
 - 检查 Kafka 集群是否正常启动，Broker 节点是否全部在线。
 - 检查 Topic 的副本配置是否合理，若副本数大于可用 Broker 数，会导致 Leader 无法选举。
- Kafka 消费者消费失败，报 `No offset found for partition`
 - 若为新消费者组，首次消费无偏移量属于正常情况，可配置 `auto.offset.reset` 为 `earliest` 或 `latest`。
 - 若为旧消费者组，检查是否手动删除过 ADCC for zk/ Kafka 中的偏移量数据，或消费者组长时间未消费导致偏移量过期（需调整 `offsets.retention.minutes` 配置）。
- ADCC for zk 集群启动后，部分节点无法加入集群，日志报 `Cannot open channel to x at election address`
 - 检查所有 ADCC for zk 节点的 `zoo.cfg` 中 `server.x` 配置的 IP / 端口是否一致，无拼写错误。
 - 检查各节点之间的时间是否同步（时间差超过 2000ms 会导致选举失败），可通过 `ntpd` 同步时间。

- 确认 dataDir 目录权限为 ADCC for zk 运行用户可读写，避免权限不足导致 myid 文件无法读取。
- Kafka 日志报 Not enough replicas，消息无法持久化
 - 检查 Topic 的 min.insync.replicas 配置值是否大于当前可用的同步副本数（ISR）。
 - 若为单 Broker 部署，需将 min.insync.replicas 改为 1（默认 1，若手动修改过需调整）。
 - 检查 Follower 节点是否正常同步，排查网络延迟、磁盘 IO 过高导致同步慢的问题。
- Kafka 启动后，JVM 内存溢出（OOM），进程直接崩溃
 - 检查 kafka-server-start.sh 中的 JVM 配置（KAFKA_HEAP_OPTS），默认堆内存可能过大 / 过小。
 - 单机测试环境可调整为 -Xmx512M -Xms512M，生产环境根据服务器内存配置（建议不超过物理内存的 50%）。
 - 检查是否开启了过多的 Topic / 分区，导致内存占用过高，需合理规划分区数。
- 连接 Kafka 时报 SSL handshake failed
 - 若开启了 SSL 认证，检查客户端 / 服务端的 SSL 证书路径、密码配置是否正确。
 - 确认证书未过期，且服务端 listeners 配置为 SSL:// 而非 PLAINTEXT://。
 - 检查客户端是否正确配置了 SSL 相关参数（如 security.protocol=SSL）。。

全国统一服务热线
4008-555-800



金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年,前身为“金蝶中间件公司”,是金蝶集团旗下新一代软件基础云平台服务商,云计算国家标准制定企业,国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业,也是“两网一站四库十二金”国家重点工程的基础平台提供商,产品广泛应用于政府、军工、金融、能源等关键行业,累计服务客户总数超过10万家。



云计算国家标准制定企业
金蝶集团旗下基础软件企业
信息技术应用创新核心企业
官网: www.apusic.com

