



开发手册

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 核心架构
 - 2.1 主题与分区
 - 2.2 副本机制
 - 2.3 高可用与故障转移
- 3 安装部署
 - 3.1 单机部署
- 4 客户端开发
 - 4.1 Maven 依赖
 - 4.2 生产者 (Producer) 核心配置
 - 4.2.1 生产者示例代码
 - 4.3 消费者 (Consumer) 核心配置
 - 4.3.1 消费者示例代码
- 5 生产环境最佳实践
 - 5.1 Topic 设计
 - 5.2 性能调优
 - 5.3 监控指标
- 6 安全配置
 - 6.1 SASL/PLAIN 认证配置
 - 6.2 ACL 示例
- 7 常用命令
 - 7.1 Topic 管理
 - 7.2 消费者组管理
 - 7.3 KRaft 模式相关
- 8 故障排查
 - 8.1 常见问题
 - 8.2 日志位置

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）的开发使用说明，帮助用户快速学习如何使用金蝶Apusic分布式消息队列for Kafka进行开发。

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 核心架构

2.1 主题与分区

- Topic (主题) : 消息的逻辑分类
- Partition (分区) : Topic 的物理分片, 实现并行读写与水平扩展
 - 单个分区内的消息严格有序
 - 不同分区之间无全局顺序保证
 - 分区数应根据吞吐量需求设置, 典型值为 10-50

2.2 副本机制

- Leader 副本: 处理所有读写请求
- Follower 副本: 仅从 Leader 同步数据, 不处理客户端请求
- ISR (In-Sync Replicas) : 与 Leader 保持同步的副本集合
 - 只有 ISR 中的副本才有资格成为新 Leader
 - 通过 `replica.lag.time.max.ms` 控制同步超时 (默认 30 秒)

2.3 高可用与故障转移

- Controller 监控 Broker 状态
- Leader 故障时, 从 ISR 中选举新 Leader (优先选择偏移量最高的副本)
- 更新集群元数据, 客户端自动重连新 Leader
- 默认禁止非 ISR 副本成为 Leader (`unclean.leader.election.enable=false`)

3 安装部署

3.1 单机部署

解压安装包

```
mkdir /apusic  
tar -zxvf ADMQ-V2.0.6.391-Kafka-20260319.tar.gz -C /apusic
```

修改配置

```
cd /apusic/admq-kafka  
vi config/kafka-standalone.conf  
...  
# 下面地址改成服务器实际 IP  
listeners=PLAINTEXT://192.168.1.10:9092  
...
```

启动

```
bin/admq-daemon start kafka zk  
bin/admq-daemon start kafka standalone
```

4 客户端开发

4.1 Maven 依赖

```
<dependency>
  <groupId>org.apache.kafka</groupId>
  <artifactId>kafka-clients</artifactId>
  <version>3.9.1</version>
</dependency>
```

4.2 生产者 (Producer) 核心配置

配置项	推荐值	说明
bootstrap.servers	host1:9092,host2:9092	集群地址列表
acks	all	等待所有 ISR 副本确认
enable.idempotence	true	启用幂等性，防止重复
compression.type	lz4 或 zstd	压缩算法
linger.ms	5-10	批量发送延迟
batch.size	16384	批次大小（字节）
retries	Integer.MAX_VALUE	重试次数
max.in.flight.requests.per.connection	5	单连接未确认请求数（幂等性启用时 ≤ 5）

4.2.1 生产者示例代码

```
Properties props = new Properties();
props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG,
"localhost:9092");
props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG,
StringSerializer.class.getName());
props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG,
```

```

StringSerializer.class.getName());
props.put(ProducerConfig.ACKS_CONFIG, "all");
props.put(ProducerConfig.ENABLE_IDEMPOTENCE_CONFIG, "true");
props.put(ProducerConfig.RETRIES_CONFIG, Integer.MAX_VALUE);
props.put(ProducerConfig.COMPRESSION_TYPE_CONFIG, "lz4");

KafkaProducer<String, String> producer = new KafkaProducer<>(props);

ProducerRecord<String, String> record = new ProducerRecord<>("my-
topic", "key1", "value1");
producer.send(record, (metadata, exception) -> {
    if (exception != null) {
        exception.printStackTrace();
    } else {
        System.out.println("Sent to partition " +
metadata.partition() +
        ", offset " + metadata.offset());
    }
});

producer.close();

```

4.3 消费者 (Consumer) 核心配置

配置项	推荐值	说明
bootstrap.servers	host1:9092,host2:9092	集群地址
group.id	自定义	消费者组 ID
enable.auto.commit	false	禁用自动提交，手动控制
auto.offset.reset	earliest 或 latest	无位移时策略
max.poll.records	500	单次 poll 最大记录数
session.timeout.ms	30000	会话超时
heartbeat.interval.ms	10000	心跳间隔（应小于会话超时的 1/3）

4.3.1 消费者示例代码

```
Properties props = new Properties();
props.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
"localhost:9092");
props.put(ConsumerConfig.GROUP_ID_CONFIG, "my-group");
props.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG,
StringDeserializer.class.getName());
props.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG,
StringDeserializer.class.getName());
props.put(ConsumerConfig.ENABLE_AUTO_COMMIT_CONFIG, "false");
props.put(ConsumerConfig.AUTO_OFFSET_RESET_CONFIG, "earliest");

KafkaConsumer<String, String> consumer = new KafkaConsumer<>(props);
consumer.subscribe(Collections.singletonList("my-topic"));

try {
    while (true) {
        ConsumerRecords<String, String> records =
consumer.poll(Duration.ofMillis(100));
        for (ConsumerRecord<String, String> record : records) {
            System.out.println("Received: " + record.value());
        }
        consumer.commitSync();
    }
} finally {
    consumer.close();
}
```

5 生产环境最佳实践

5.1 Topic 设计

- 分区数计算：单分区吞吐 $\times$ 分区数 $\geq$ 目标吞吐
- 副本数：生产环境推荐 3 副本
- 清理策略：
 - `cleanup.policy=delete`：按时间/大小清理
 - `cleanup.policy=compact`：日志压缩，保留最新 Key

5.2 性能调优

- 生产者侧：
 - 启用压缩（`lz4` 或 `zstd`）
 - 合理设置 `linger.ms` 和 `batch.size`
 - 启用幂等性和事务（Exactly-Once 语义）
- 消费者侧：
 - 调整 `max.poll.records` 避免单次处理过慢
 - 合理设置 `fetch.min.bytes` 和 `fetch.max.wait.ms`

5.3 监控指标

指标	说明	告警阈值
UnderReplicatedPartitions	未充分同步的分区数	> 0
OfflinePartitionsCount	离线分区数	> 0
RequestLatencyAvg	平均请求延迟	$> 100\text{ms}$
MessagesInPerSec	每秒消息数	根据业务设定
BytesInPerSec	每秒写入字节数	根据业务设定

6 安全配置

6.1 SASL/PLAIN 认证配置

```
# broker 配置
listeners=SASL_PLAINTEXT://0.0.0.0:9092
security.inter.broker.protocol=SASL_PLAINTEXT
sasl.mechanism.inter.broker.protocol=PLAIN
sasl.enabled.mechanisms=PLAIN

authorizer.class.name=kafka.security.authorizer.AclAuthorizer
allow.everyone.if.no.acl.found=false

# JAAS 配置文件（以下密码仅为示例，生产环境请使用强密码）
KafkaServer {
    org.apache.kafka.common.security.plain.PlainLoginModule required
    username="admin"
    password="admin-secret"
    user_admin="admin-secret"
    user_alice="alice-secret";
};
```

6.2 ACL 示例

```
# 授予 alice 对 topic my-topic 的读写权限
kafka-acls.sh --bootstrap-server localhost:9092 \
    --add --allow-principal User:alice \
    --operation Read --operation Write \
    --topic my-topic

# 授予 alice 对消费者组 my-group 的操作权限
kafka-acls.sh --bootstrap-server localhost:9092 \
```

```
--add --allow-principal User:alice \  
--operation Read --group my-group
```

7 常用命令

ADMQ for Kafka提供原生命令，位于admq-kafka/kafka/bin/目录下

7.1 Topic 管理

```
cd /apusic/admq-kafka/kafka/bin/

# 创建 topic
kafka-topics.sh --bootstrap-server localhost:9092 --create \
  --topic my-topic --partitions 3 --replication-factor 3

# 列出所有 topic
kafka-topics.sh --bootstrap-server localhost:9092 --list

# 查看 topic 详情
kafka-topics.sh --bootstrap-server localhost:9092 --describe --topic
my-topic

# 修改分区数 (只能增加)
kafka-topics.sh --bootstrap-server localhost:9092 --alter \
  --topic my-topic --partitions 6

# 删除 topic
kafka-topics.sh --bootstrap-server localhost:9092 --delete --topic
my-topic
```

7.2 消费者组管理

```
# 列出消费者组
kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list

# 查看消费者组详情
kafka-consumer-groups.sh --bootstrap-server localhost:9092 \
```

```
--describe --group my-group
```

重置位移

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

7.3 KRaft 模式相关

格式化存储目录 (仅首次)

```
kafka-storage.sh format -t <cluster-id> -c  
config/kraft/server.properties
```

查看仲裁状态

```
kafka-metadata-quorum.sh --bootstrap-server localhost:9092 describe  
--status
```

动态添加控制器节点

```
kafka-metadata-quorum.sh --bootstrap-server localhost:9092 add \  
--controller-quorum-voter <node-id>@<host>:<port>
```

8 故障排查

8.1 常见问题

问题	可能原因	解决方案
消费者频繁重平衡	<code>session.timeout.ms</code> 过小或处理时间过长	增大会话超时，优化消费逻辑
消息丢失	<code>acks=1</code> 或未启用幂等性	设置 <code>acks=all</code> ，启用 <code>enable.idempotence</code>
重复消费	位移提交失败	使用手动提交，处理重复逻辑
生产延迟高	网络或磁盘瓶颈	检查网络带宽，优化磁盘 I/O
分区不可用	Broker 故障或副本不足	检查 Broker 状态，确保 ISR 足够

8.2 日志位置

- Broker 日志: `logs/server.log`
- Controller 日志: `logs/controller.log`
- 状态变更日志: `logs/state-change.log`

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

