



APUSIC
固若长城
睿比世界

開發手冊

金蝶Apusic分布式消息队列V2.0.6_for_kafka

版權聲明

本文件所涉及的軟件著作權、版權等知識產權已依法進行了註冊，由金蝶天燕雲計算股份有限公司合法擁有。受《中華人民共和國著作權法》《計算機軟件保護條例》《知識產權保護條例》和相關國際版權條約、法律、法規以及其它知識產權法律和條約的保護。未經授權許可，不得非法使用。

免責聲明

本文件包含的版權資訊由金蝶天燕雲計算股份有限公司合法擁有，受法律的保護，金蝶天燕雲計算股份有限公司對本文件可能涉及到的非金蝶天燕雲計算股份有限公司的資訊不承擔任何責任。在法律允許的範圍內，您可以查閱並僅能夠在《中華人民共和國著作權法》規定的合法範圍內複製和打印本文件。任何單位和個人未經金蝶天燕雲計算股份有限公司書面授權許可，不得使用、修改、再發佈本文件的任何部分和內容，否則將被視為侵權，金蝶天燕雲計算股份有限公司有依法追究其責任的權利。

本文件如有更新，不另行通知。對本文件中的問題您可向金蝶天燕雲計算股份有限公司告知或查詢。未經本公司明確授予的任何權利均予保留。

商標聲明

 是深圳市金蝶天燕雲計算股份有限公司向中華人民共和國國家商標局申請註冊的註冊商標，註冊商標專用權由金蝶天燕合法擁有，受法律保護。未經金蝶天燕的書面許可，任何單位及個人不得以任何方式或理由對該商標的任何部分進行使用、複製、修改、傳播、抄錄或與其它產品捆綁使用銷售。凡侵犯金蝶天燕商標權的，金蝶天燕將依法追究其法律責任。本文件提及的其他所有商標或註冊商標，由各自的所有人擁有。

目錄

- 1 前言
 - 1.1 適用對象
 - 1.2 相關文件
 - 1.3 技術支援
- 2 核心架構
 - 2.1 主題與分割區
 - 2.2 副本機制
 - 2.3 高可用與故障轉移
- 3 安裝部署
 - 3.1 單機部署
- 4 用戶端開發
 - 4.1 Maven 依賴
 - 4.2 生產者 (Producer) 核心配置
 - 4.2.1 生產者範例程式碼
 - 4.3 消費者 (Consumer) 核心配置
 - 4.3.1 消費者範例程式碼
- 5 生產環境最佳實踐
 - 5.1 Topic 設計
 - 5.2 效能調優
 - 5.3 監控指標
- 6 安全配置
 - 6.1 SASL/PLAIN 認證配置
 - 6.2 ACL 範例
- 7 常用命令
 - 7.1 Topic 管理
 - 7.2 消費者組管理
 - 7.3 KRaft 模式相關
- 8 故障排查
 - 8.1 常見問題
 - 8.2 日誌位置

1 前言

本文檔為金蝶Apusic分散式訊息佇列for Kafka（Apusic Distributed Message Queue，簡稱：ADMQ for Kafka）的開發使用說明，幫助使用者快速學習如何使用金蝶Apusic分散式訊息佇列for Kafka進行開發。

1.1 適用對象

本文檔適用於IT資訊化業務負責人、研發經理、軟體專案經理、軟體架構師、運維工程師。

1.2 相關文件

了解更多ADMQ for Kafka產品相關的資訊，請參閱以下ADMQ for Kafka產品手冊文件集：

序號	手冊文件	說明
1	金蝶Apusic分散式訊息佇列for Kafka快速使用手冊	簡單介紹了如何快速上手使用ADMQ for Kafka。
2	金蝶Apusic分散式訊息佇列for Kafka安裝手冊	詳細介紹如何在各作業系統上安裝ADMQ for Kafka，以及ADMQ for Kafka服務啟停等操作。
3	金蝶Apusic分散式訊息佇列for Kafka訊息引擎使用者手冊	詳細介紹ADMQ for Kafka訊息引擎相關功能的使用、設定、管理及配套工具的使用方法。
4	金蝶Apusic分散式訊息佇列for Kafka管控台使用者手冊	詳細介紹ADMQ for Kafka管控台相關功能的使用和操作說明。
5	金蝶Apusic分散式訊息佇列for Kafka開發手冊	詳細介紹基於各開發語言進行ADMQ for Kafka用戶端應用開發的說明。
6	金蝶Apusic分散式訊息佇列for Kafka遷移手冊	詳細介紹從Kafka遷移到ADMQ for Kafka的說明。
7	金蝶Apusic分散式訊息佇列for Kafka運維手冊	詳細介紹ADMQ for Kafka的監控、運維、安全加固等運維說明。
8	金蝶Apusic分散式訊息佇列for Kafka效能最佳化手冊	詳細介紹ADMQ for Kafka效能調優的說明。

1.3 技術支援

ADMQ for Kafka產品提供全面的技術支援服務，您可以透過以下方式獲得技術支援：

- 網址: www.apusic.com
- 電話: 400-855-5800
- 信箱: support@apusic.com
- 金蝶雲社群: <https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

您在取得技術支援時，請提供如下資訊：

1. 您的姓名
2. 公司資訊與聯絡方式
3. 作業系統及其版本
4. 產品版本號
5. 出現異常及錯誤的日誌、截圖等詳細資訊

2 核心架構

2.1 主題與分割區

- Topic (主題) : 訊息的邏輯分類
- Partition (分割區) : Topic 的物理分片, 實現平行讀寫與水平擴充
 - 單個分割區內的訊息嚴格有序
 - 不同分割區之間無全域性順序保證
 - 分割區數應根據吞吐量需求設定, 典型值為 10-50

2.2 副本機制

- Leader 副本: 處理所有讀寫請求
- Follower 副本: 僅從 Leader 同步資料, 不處理用戶端請求
- ISR (In-Sync Replicas) : 與 Leader 保持同步的副本集合
 - 只有 ISR 中的副本才有資格成為新 Leader
 - 透過 `replica.lag.time.max.ms` 控制同步超時 (預設 30 秒)

2.3 高可用與故障轉移

- Controller 監控 Broker 狀態
- Leader 故障時, 從 ISR 中選舉新 Leader (優先選擇偏移量最高的副本)
- 更新叢集中繼資料, 用戶端自動重新連線新 Leader
- 預設禁止非 ISR 副本成為 Leader (`unclean.leader.election.enable=false`)

3 安裝部署

3.1 單機部署

解壓安裝包

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

修改配置

```
cd /apusic/admq-kafka  
vi config/kafka-standalone.conf  
...  
# 下面位址改成伺服器實際 IP  
listeners=PLAINTEXT://192.168.1.10:9092  
...
```

啟動

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

4 用戶端開發

4.1 Maven 依賴

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

4.2 生產者 (Producer) 核心配置

配置項	推薦值	說明
bootstrap.servers	host1:9092,host2:9092	叢集位址清單
acks	all	等待所有 ISR 副本確認
enable.idempotence	true	啟用冪等性，防止重複
compression.type	lz4 或 zstd	壓縮演算法
linger.ms	5-10	批次傳送延遲
batch.size	16384	批次大小（位元組）
retries	Integer.MAX_VALUE	重試次數
max.in.flight.requests.per.connection	5	單連線未確認請求數（冪等性啟用時 ≤ 5）

4.2.1 生產者範例程式碼

```
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 消費者 (Consumer) 核心配置

配置項	推薦值	說明
bootstrap.servers	host1:9092,host2:9092	叢集位址
group.id	自訂	消費者組 ID
enable.auto.commit	false	停用自動提交，手動控制
auto.offset.reset	earliest 或 latest	無位移時策略
max.poll.records	500	單次 poll 最大記錄數
session.timeout.ms	30000	工作階段超時
heartbeat.interval.ms	10000	心跳間隔（應小於工作階段超時的 1/3）

4.3.1 消費者範例程式碼

```
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 生產環境最佳實踐

5.1 Topic 設計

- 分割區數計算：單分割區吞吐 $\times$ 分割區數 $\geq$ 目標吞吐
- 副本數：生產環境推薦 3 副本
- 清理策略：
 - `cleanup.policy=delete`：按時間/大小清理
 - `cleanup.policy=compact`：日誌壓縮，保留最新 Key

5.2 效能調優

- 生產者側：
 - 啟用壓縮（`lz4` 或 `zstd`）
 - 合理設定 `linger.ms` 和 `batch.size`
 - 啟用冪等性和事務（Exactly-Once 語義）
- 消費者側：
 - 調整 `max.poll.records` 避免單次處理過慢
 - 合理設定 `fetch.min.bytes` 和 `fetch.max.wait.ms`

5.3 監控指標

指標	說明	告警閾值
UnderReplicatedPartitions	未充分同步的分割區數	> 0
OfflinePartitionsCount	離線分割區數	> 0
RequestLatencyAvg	平均請求延遲	$> 100\text{ms}$
MessagesInPerSec	每秒訊息數	根據業務設定
BytesInPerSec	每秒寫入位元組數	根據業務設定

6 安全配置

6.1 SASL/PLAIN 認證配置

```
# 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 配置檔 (以下密碼僅為範例，生產環境請使用強密碼)
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 範例

```
# 授予 alice 對 topic my-topic 的讀寫許可權
kafka-acls.sh --bootstrap-server localhost:9092 \
    --add --allow-principal User:alice \
    --operation Read --operation Write \
    --topic my-topic

# 授予 alice 對消費者組 my-group 的操作許可權
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，位於admq-kafka/kafka/bin/目錄下

7.1 Topic 管理

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

# 建立 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

# 檢視 topic 詳情
kafka-topics.sh --bootstrap-server localhost:9092 --describe --topic
my-topic

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

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

7.2 消費者組管理

```
# 列出消費者組
kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list

# 檢視消費者組詳情
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 模式相關

格式化儲存目錄（僅首次）

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

檢視仲裁狀態

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

動態新增控制器節點

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

8 故障排查

8.1 常見問題

問題	可能原因	解決方案
消費者頻繁重新平衡	<code>session.timeout.ms</code> 過小或處理時間過長	增大工作階段超時，最佳化消費邏輯
訊息遺失	<code>acks=1</code> 或未啟用幂等性	設定 <code>acks=all</code> ，啟用 <code>enable.idempotence</code>
重複消費	位移提交失敗	使用手動提交，處理重複邏輯
生產延遲高	網路或磁碟瓶頸	檢查網路頻寬，最佳化磁碟 I/O
分割區不可用	Broker 故障或副本不足	檢查 Broker 狀態，確保 ISR 足夠

8.2 日誌位置

- Broker 日誌: `logs/server.log`
- Controller 日誌: `logs/controller.log`
- 狀態變更日誌: `logs/state-change.log`

全國統一服務熱線
4008-555-800

金蝶天燕雲計算股份有限公司（簡稱“金蝶天燕”）成立於2000年，前身為“金蝶中間件公司”，是金蝶集團旗下基礎軟件平台提供商，雲計算國家標準制定企業，國家信創產業核心軟件企業。金蝶天燕是國家863重點研發計劃與核高基重大專項承接企業，也是“兩網一站四庫十二金”國家重點工程的基礎平台提供商，產品廣泛應用於政府、軍工、金融、能源等關鍵行業，累計服務客戶總數超過10萬家。



雲計算國家標準制定企業
金蝶集團旗下基礎軟件企業
資訊技術應用創新核心企業
官網：www.apusic.com

