



APUSIC
固若长城
睿比世界

核心引擎使用者手冊

金蝶Apusic分布式消息队列V2.0.6_for_kafka

版權所有 © 深圳市金蝶天燕雲計算股份有限公司2026。保留所有權利。

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 核心引擎目錄說明
- 3 Broker 配置說明
 - 3.1 基礎配置 (Basic Configuration)
 - 3.2 副本與可靠性 (Replication & Reliability)
 - 3.3 日誌保留策略 (Log Retention)
 - 3.4 效能調優 (Performance Tuning)
 - 3.5 ADCC for zk 連線 (ADCC for zk Connection)
 - 3.6 安全 (Security)
 - 3.7 其他 (Others)
- 4 ADCC for zk 配置說明
 - 4.1 基礎配置 (Basic Timing)
 - 4.2 資料儲存 (Data Storage)
 - 4.3 連線資訊 (Connection Info)
 - 4.4 叢集配置 (Cluster Config)
 - 4.5 自動清理 (Auto Purge)
 - 4.6 TCP 網路最佳化 (Network Tuning)
 - 4.7 監控和管理 (Admin & 4lw)
 - 4.8 其他配置 (Others)
- 5 常見命令
 - 5.1 Topic 管理
 - 5.1.1 建立 Topic
 - 5.1.2 檢視 Topic 清單
 - 5.1.3 檢視 Topic 詳情 (分割區、副本分佈)
 - 5.1.4 增加分割區數 (只能增加，不能減少)
 - 5.1.5 刪除 Topic
 - 5.2 生產與消費 (資料測試)
 - 5.2.1 啟動控制檯生產者 (Producer)
 - 5.2.2 啟動控制檯消費者 (Consumer)

- 5.3 消費者組管理 (Consumer Groups)
 - 5.3.1 檢視消費者組清單
 - 5.3.2 檢視組內詳情 (延遲、Offset)
 - 5.3.3 重置 Offset (謹慎使用)
- 5.4 叢集中繼資料管理
 - 5.4.1 檢視叢集 Broker 資訊
 - 5.4.2 首選副本選舉 (Preferred Leader Election)
 - 5.4.3 檢視叢集配置
- 6 常見問題
 - 6.1 常見錯誤排查

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）訊息引擎的使用者手冊，詳細介紹了ADMQ for Kafka訊息引擎的功能使用、設定方法及管理操作等內容。

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 核心引擎目錄說明

目錄名	說明
bin	存放啟動、停止及管理指令稿
config	存放設定檔 (如 kafka-standalone.conf)
data	存放 Kafka 訊息資料及 ADCC for zk 資料
jdk	內建的 Java 執行環境，無需額外設定系統 JDK
logs	存放執行日誌 (Broker 日誌、ADCC for zk 日誌等)
service	系統服務註冊相關檔案 (可選)

3 Broker 配置說明

3.1 基礎配置 (Basic Configuration)

核心作用：定義 Broker 的身份、網路暴露方式及儲存路徑。

參數	說明	生產環境建議
<code>broker.id</code>	叢集唯一標識。 Kafka 透過此 ID 在 ADCC for zk 中註冊自己。	必須唯一。 若重啟後 ID 改變，Broker 會被視為新節點，導致舊資料不可見或叢集中繼資料混亂。通常寫死在配置檔中，不要動態產生。
<code>listeners</code>	監聽位址。 定義 Broker 繫結哪些 IP 和連線埠，以及使用什麼協定。格式：協定://IP:連線埠。	- 內網部署 ： <code>PLAINTEXT://0.0.0.0:9092</code> (繫結所有網卡) 或指定內網 IP。 - 多網卡/混合雲 ：可設定多個，如 <code>INTERNAL://192.168.1.10:9092,EXTERNAL://10.0.0.5:9093</code> 。
<code>advertised.listeners</code>	廣播位址。 用戶端連線時，Broker 傳回給用戶端的位址。 至關重要。	- 如果用戶端和 Broker 不在同一網路（如 Docker、雲伺服器）， 必須 設定此項為用戶端可存取的公網 IP 或網域名稱。 - 若不設定，預設取 <code>listeners</code> 的值，可能導致用戶端連線內網 IP 失敗。
<code>log.dirs</code>	資料儲存目錄。 訊息日誌檔案存放路徑。	- 強烈建議使用高效能 SSD。 - 可設定多個目錄（逗號分隔）以利用多塊磁碟 IO，如 <code>/data1/kafka,/data2/kafka</code> 。

<code>num.partitions</code>	預設分割區數。 建立 Topic 時未指定分割區數時的預設值。	建議設定為 3~6（與叢集 Broker 數量匹配或略多），以提高平行消費能力。
<code>auto.create.topics.enable</code>	自動建立 Topic。 當生產者向不存在的 Topic 發送訊息時，是否自動建立。	**生產環境強烈建議設為 <code>****false**</code>。 原因：防止因程式碼拼寫錯誤建立大量無用 Topic，且自動建立的 Topic 參數（分割區/副本）往往不符合生產標準。

3.2 副本與可靠性 (Replication & Reliability)

核心作用：保證資料不遺失、服務高可用。

參數	說明	生產環境建議
<code>default.replication.factor</code>	預設副本數。 新建 Topic 時的預設副本數量。	建議 ≥ 3 。允許同時掛掉 2 臺機器而不遺失資料。需確保叢集 Broker 數量 $\geq$ 該值。
<code>min.insync.replicas</code>	最小同步副本數。 配合 <code>acks=all</code> 使用，表示寫入成功至少需要多少個副本確認。	- 若設為 1：只要 Leader 寫入成功即返回，Leader 掛掉可能丟資料。 - 推薦設為 2 (配合 3 副本)：保證即使掛掉 1 臺，仍有 1 個完整副本存活，且寫入時需 2 個節點成功，兼顧安全與可用。
<code>offsets.topic.replication.factor</code>	<code>__consumer_offsets</code> 內部 Topic 的副本數。	必須 ≥ 2 (最好 3)。若該 Topic 掛了，所有消費者無法提交位移，導致重複消費或消費停滯。
<code>replica.lag.time.max.ms</code>	ISR 踢出閾值。 Follower 落後 Leader 超過此時間，將被移出 ISR 清單。	預設 30s。網路抖動大可適當調大，但過大會導致故障切換變慢。
<code>unclean.leader.election.enable</code>	非 ISR 選舉。 允許不在 ISR 中的副本（資料落後的）成為 Leader。	必須設為 <code>**false**</code> (預設)。 若設為 <code>true</code> ：當所有 ISR 掛掉，強制讓資料落

		後的副本上位，會導致資料遺失。生產環境嚴禁開啟。
<code>controller.quorum.voters</code>	控制器投票節點。僅用於 KRaft 模式 (無 ADCC for zk)。	ADCC for zk 模式下留空。若遷移到 KRaft 模式，需設定 Controller 節點的 ID 和位址。

3.3 日誌保留策略 (Log Retention)

核心作用：控制磁碟空間使用，清理舊資料。

參數	說明	生產環境建議
<code>log.retention.hours</code>	按時間保留。日誌檔案保留的小時數。	預設 168h (7天)。根據業務需求調整，如合規要求保留 30 天則設為 720。
<code>log.retention.bytes</code>	按大小保留。每個 Partition 最大保留的位元組數。	註解該配置則預設為 -1 (不限)。建議設定上限 (如 50GB)，防止單 Topic 寫爆磁碟。
<code>log.segment.bytes</code>	段檔案大小。日誌切分的大小閾值。	預設 1GB。太大導致清理粒度粗，太小導致檔案控制代碼過多。一般保持預設即可。
<code>log.retention.check.interval.ms</code>	檢查間隔。背景執行緒檢查是否要刪除舊日誌的頻率。	預設 5 分鐘。無需頻繁修改。

3.4 效能調優 (Performance Tuning)

核心作用：平衡吞吐量與延遲，適配硬體資源。

參數	說明	生產環境建議
<code>num.network.threads</code>	網路執行緒數。處理 Socket 讀寫、編解碼的執行緒數。	預設 8。高並發場景 (如萬級 TPS) 可調至更大或與 CPU 核心數匹配。
<code>num.io.threads</code>	IO 執行緒數。處理磁碟讀寫請求的執行緒數。	預設 16。建議設定為磁碟數量的 2 倍或 CPU 核心數，如 16。
<code>socket.send/receive.buffer.bytes</code>	Socket 緩衝區。發送/接收緩衝區大小。	預設 1024KB。高吞吐場景可適當調大 (如 2MB)，依賴 OS 核心參數配合。

<code>log.flush.interval.messages/ms</code>	刷盤策略。 多少條訊息或多少毫秒強制刷盤到磁碟。	預設值極大 (即不主動強制刷盤)。 最佳實踐： 依賴 OS 的 Page Cache 非同步刷盤，效能最高。僅在極度追求資料零遺失且能接受效能大幅下降時才調小此值。
---	---------------------------------	--

3.5 ADCC for zk 連線 (ADCC for zk Connection)

核心作用：連線中繼資料管理中心（僅限 ADCC for zk 模式）。

參數	說明	生產環境建議
<code>zookeeper.connect</code>	ADCC for zk 位址清單。格式 <code>host:port</code> ，多節點用逗號分隔。	填寫所有 ADCC for zk 節點位址。支援 chroot 路徑。
<code>zookeeper.session.timeout.ms</code>	工作階段超時。 Broker 與 ADCC for zk 心跳遺失多久判定為下線。	預設 18s-30s。調小可加快故障發現，但網路抖動易導致誤判重新平衡；調大則故障恢復慢。
<code>zookeeper.chroot.path</code>	根路徑。 將 Kafka 中繼資料儲存在 ADCC for zk 的某個子路徑下。	多叢集共用 ADCC for zk 時必配。 例如 <code>/kafka-cluster-01</code> ，避免不同 Kafka 叢集中繼資料衝突。

3.6 安全 (Security)

核心作用：認證與加密（註解部分為範例）。

參數	說明	生產環境建議
<code>security.inter.broker.protocol</code>	Broker 間通訊協定。	生產環境建議啟用 <code>SASL_SSL</code> 或至少 <code>SASL_PLAINTEXT</code> ，防止內部竊聽或偽造節點。
<code>sasl.enabled.mechanisms</code>	認證機制。	常用 <code>PLAIN</code> (簡單帳號密碼), <code>SCRAM-SHA-256/512</code> (更安全), <code>GSSAPI</code> (Kerberos)。
<code>listener.name...jaas.config</code>	JAAS 配置。	定義使用者名稱密碼。建議將敏感資訊移至獨立的 JAAS 檔案，而非直接寫在 <code>properties</code> 中。

3.7 其他 (Others)

參數	說明	生產環境建議
----	----	--------

<code>delete.topic.enable</code>	刪除 Topic 開關。	**建議 **<code>true</code>**。 方便運維清理測試資料或廢棄業務線資料。若為 <code>false</code> ，刪除命令僅標記無效，不釋放磁碟。
<code>JMX_PORT</code>	監控告警。	必須設定。 用於 Prometheus/JConsole 採集指標。需在啟動指令稿中透過 <code>-D</code> 參數設定，並注意防火牆放行。
<code>replica.fetch.wait.max.ms</code>	Follower 拉取等待時間。	預設 500ms。適當調大可減少 Follower 對 Leader 的空輪詢壓力，但會增加同步延遲。一般保持預設。

4 ADCC for zk 配置說明

4.1 基礎配置 (Basic Timing)

核心作用：定義心跳節奏和叢集容錯時間視窗，直接影響故障檢測速度。

參數	說明	生產環境建議
tickTime	心跳基準時間 (ms)。ADCC for zk 內部所有時間計算的基礎單位。	預設 2000ms (2秒)。 - 設太小：網路抖動易導致誤判節點下線。 - 設太大：故障檢測慢。 建議保持 2000 ，除非網路極差可調至 3000-4000。
initLimit	初始化同步時限 。Follower 啟動時從 Leader 同步資料的最大 tickTime 倍數。	預設 10 (即 20秒)。若資料量大或磁碟慢，需調大 (如 20)，否則 Follower 啟動會逾時失敗。
syncLimit	心跳同步時限 。Follower 與 Leader 心跳響應的最大 tickTime 倍數。	預設 5 (即 10秒)。超過此時間未回應，Leader 認為該 Follower 已死，將其踢出叢集。 網路不穩定時可適當調大至 8-10 。

4.2 資料儲存 (Data Storage)

核心作用：控制資料落盤策略，直接影響 IO 效能和磁碟壽命。

參數	說明	生產環境建議
dataDir	快照資料目錄 。儲存記憶體資料庫的快照 (snapshot) 和 myid 檔案。	必須設定在高效能 SSD 上 。不要與交易日誌 (dataLogDir) 混用，避免 IO 爭搶。
dataLogDir	交易日誌目錄 。儲存寫入操作的交易日誌 (transaction log)。	關鍵效能點： 1. 務必獨立掛載磁碟 (與 dataDir 物理隔離)。 2. ADCC for zk 對順序寫要求高，獨立的 SSD 能極大提升吞吐量。
preAllocSize	預分配檔案大小 (KB)。交易日誌檔案預分配的大小。	預設 65536 (64MB)。配置中設為 131072 (128MB)。 作用： 減少檔案系統碎片和頻繁的檔案擴充系統呼叫。大檔案有利於順序寫效能。
snapCount	快照觸發頻率 。每處理多少事務產生一次快照。	預設 100,000。配置中設為 200,000。 作用： 減少快照產生頻率，降低磁碟 IO 壓力。但會導致重啟恢復時間變長 (因為要重播更多日誌)。生產環境可適當調大。

<code>fsync.warningthresholdms</code>	刷盤警告閾值 (ms) 。當 <code>fsync</code> 耗時超過此值時記錄警告日誌。	用於監控磁碟健康狀況。若頻繁出現警告，說明磁碟 IO 瓶頸嚴重，需更換硬體或最佳化負載。
---------------------------------------	---	--

4.3 連線資訊 (Connection Info)

核心作用：定義用戶端接入點和並發控制。

參數	說明	生產環境建議
<code>clientPort</code>	用戶端監聽連線埠 。Kafka Broker 和業務用戶端連線 ADCC for zk 的連線埠。	預設 2181。防火牆需開放此連線埠。
<code>maxClientCnxns</code>	單 IP 最大連線數 。限制單個 IP 位址能建立的最大連線數。	預設 60。配置中設為 0 (表示無限制)。 風險 ：若某臺機器異常發起大量連線，可能耗盡 ADCC for zk 資源。 生產環境建議設定合理上限 (如 200-500)，防止 DDoS 或程式碼無窮迴圈。

4.4 叢集配置 (Cluster Config)

核心作用：定義叢集拓撲和節點角色。

參數	說明	生產環境建議
<code>server.x=A:B:C:D;E</code>	叢集節點定義 。 <code>x</code> : MyID (對應 <code>myid</code> 檔案)。 <code>A</code> : IP。 <code>B</code> : 2888 (Follower 與 Leader 通訊)。 <code>C</code> : 3888 (選舉通訊)。 <code>D</code> : 角色 (participant/observer)。 <code>E</code> : 用戶端連線埠 (可選，新版支援)。	- 奇數節點 ：叢集節點數必須為奇數 (3, 5, 7)，以容忍 $(N-1)/2$ 臺故障。 - Observer ：若需增加讀效能而不增加投票負擔，可將部分節點設為 <code>observer</code> 。
<code>electionPortBindRetry</code>	選舉連線埠繫結重試次數 。	預設 3。若連線埠被佔用，重試幾次後退出。一般無需修改。

4.5 自動清理 (Auto Purge)

核心作用：防止磁碟被歷史快照和日誌填滿。

參數	說明	生產環境建議
<code>autopurge.snapRetainCount</code>	保留快照數量 。保留最近的 N 個快照檔案。	配置中設為 3。 建議 ：至少保留 3 個，以防最新快照損壞時有舊版本可回滾。

<code>autopurge.purgeInterval</code>	清理間隔 (小時)。每隔多久執行一次清理任務。	配置中設為 6 (小時)。 建議 ：根據資料量調整。高寫入場景可設為 1-2 小時，低負載可設為 12-24 小時。
--------------------------------------	-------------------------	---

4.6 TCP 網路最佳化 (Network Tuning)

核心作用：應對高並發連線和工作階段管理。

參數	說明	生產環境建議
<code>globalOutstandingLimit</code>	全域待處理請求上限。防止請求堆積導致 OOM。	預設 1000。配置中設為 5000。 注意 ：調大可提高吞吐量，但若後端處理慢，可能導致記憶體暴漲。需配合監控觀察記憶體使用率。
<code>minSessionTimeout</code>	最小工作階段超時 (ms)。用戶端請求的 timeout 低於此值將被強制設為此值。	預設 2 * tickTime。設太小會導致網路微抖動就斷連。建議 $\geq 4000\text{ms}$ 。
<code>maxSessionTimeout</code>	最大工作階段超時 (ms)。	預設 20 * tickTime。配置中設為 60000 (60秒)。允許長連線用戶端設定更長的超時時間，減少重新連線頻率。

4.7 監控和管理 (Admin & 4lw)

核心作用：運維監控與安全控制。

參數	說明	生產環境建議
<code>admin.enableServer</code>	是否開啟 AdminServer (HTTP 介面)。	預設 true (新版)。配置中設為 false。 安全建議 ：若不需要 HTTP 監控介面， 建議關閉 以減少攻擊面。可透過 JMX 或 4lw 命令監控。
<code>4lw.commands.whitelist</code>	四字命令白名單。ZK 透過 <code>echo cmd nc ip 2181</code> 進行監控。	

4.8 其他配置 (Others)

參數	說明	生產環境建議
<code>skipACL</code>	跳過 ACL 檢查。	預設 no。若設為 yes，將忽略所有許可權檢查，大幅提升效能，但失去安全性。僅在完全可信的內網且追求極致效能時使用。

leaderServers	Leader 是否處理讀請求。	配置中 no (可能是舊版本參數或特定發行版)。標準 ADCC for zk 中 Leader 預設處理所有寫請求和部分讀請求。若設為不參與服務，可能影響讀吞吐量。通常保持預設。
forceSync	是否強制每次事務都 fsync。	預設 no (使用 OS 緩衝)。若設為 yes，資料最安全但效能急劇下降。 **生產環境務必保持 ***no**。
sslQuorum / portUnification	SSL 加密與連線埠統一。	若需啟用 SSL 加密集群通訊，需設為 true 並設定憑證。明文內網通訊可保持 false。

5 常見命令

5.1 Topic 管理

5.1.1 建立 Topic

建立一個名為 `test-topic` 的 Topic，包含 3 個分割區，3 個副本。

```
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` 不能超過叢集中 Broker 的數量。

5.1.2 檢視 Topic 清單

```
# 列出所有 Topic  
kafka/bin/kafka-topics.sh --list --bootstrap-server  
192.168.1.10:9092  
  
# 過濾特定 Topic (支援正規表示式)  
kafka/bin/kafka-topics.sh --list --bootstrap-server  
192.168.1.10:9092 | grep "test"
```

5.1.3 檢視 Topic 詳情 (分割區、副本分佈)

這是排查資料傾斜和副本狀態最重要的命令。

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

輸出解讀：

- `Leader`：目前負責讀寫的副本 ID。

- `Replicas` : 該分割區所有副本所在的 Broker ID 清單。
- `Isr` (In-Sync Replicas): 目前與 Leader 保持同步的副本清單。如果 `Isr` 數量少於 `Replicas`, 說明有副本落後或宕機。

5.1.4 增加分割區數 (只能增加, 不能減少)

將 `test-topic` 的分割區數從 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 刪除 Topic

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

- 前提: 設定檔 `server.properties` 中 `delete.topic.enable=true`。

5.2 生產與消費 (資料測試)

5.2.1 啟動控制檯生產者 (Producer)

向 `test-topic` 傳送訊息。輸入一行按回車即傳送一條訊息。

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

- 進階用法: 指定 Key 傳送 (用於測試分割區策略)

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

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

5.2.2 啟動控制檯消費者 (Consumer)

從 `test-topic` 消費訊息。

- 消費最新資料：

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

- **從頭開始消費 (歷史資料)**：

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

- 顯示 Key 和分割區資訊：

```
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
```

- 限制消費數量 (測試用)：

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

5.3 消費者組管理 (Consumer Groups)

5.3.1 檢視消費者組清單

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

5.3.2 檢視組內詳情 (延遲、Offset)

運維最常用命令，用於檢視消費積壓 (Lag)。

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

輸出關鍵列：

- `CURRENT-OFFSET`：目前已提交的位置。
- `LOG-END-OFFSET`：分割區最新的訊息位置。
- `LAG`：**積壓量** = `LOG-END-OFFSET` - `CURRENT-OFFSET`。如果 LAG 持續增大，說明消費速度跟不上生產速度。

5.3.3 重置 Offset (謹慎使用)

將消費者組的 Offset 重置到最早、最新或指定時間。

- 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** (相當於忽略歷史積壓):

```
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:** 執行重置時，消費者組必須處於**非活躍狀態**（即沒有正在執行的消費者例項）。

5.4 叢集中繼資料管理

5.4.1 檢視叢集 Broker 資訊

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

5.4.2 首選副本選舉 (Preferred Leader Election)

如果因為故障導致 Leader 分佈不均勻（例如所有 Leader 都在 Node 1），可以觸發重新選舉，讓 Leader 回到 `Replicas` 清單中的第一個節點（通常是理想分佈）。

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

5.4.3 檢視叢集配置

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

6 常見問題

6.1 常見錯誤排查

- ADCC for zk 啟動失敗，日誌報 `Connection refused` 或 `Session expired`
 - 檢查 `myid` 檔案是否存在且內容正確。
 - 檢查 `zookeeper.properties` 中 `server.x` 的 IP 是否可達，連線埠 2888/3888 是否被防火牆攔截。
- Kafka 啟動失敗，日誌報 `Broker ID 0 is already registered`
 - 檢查是否有殘留的舊 ADCC for zk 資料。如果是新部署，清空 ADCC for zk 的 `dataDir` 和 Kafka 的 `log.dirs` 後重試。
 - 確認各節點 `broker.id` 不重複。
- ISR 集合不全 (Isr: 0,1 而不是 0,1,2)
 - 說明某個 Follower 同步太慢或網路不通。檢查該節點の日誌和網路頻寬。
- Kafka 啟動失敗，日誌報** `Failed to bind to port 9092` **
 - 檢查 9092 連線埠是否被其他程式佔用（使用 `netstat -tulpn | grep 9092` 或 `lsof -i:9092` 命令）。
 - 檢查 `server.properties` 中 `listeners` 配置的連線埠是否與實際要使用的連線埠一致，避免配置衝突。
 - 確認防火牆未攔截 9092 連線埠，或已在防火牆規則中放行該連線埠。
- Kafka 生產者傳送訊息失敗，報 `Leader not available`
 - 檢查對應的 Topic 是否存在，若不存在需先建立（`kafka-topics.sh --create` 命令）。
 - 檢查 Kafka 叢集是否正常啟動，Broker 節點是否全部線上。
 - 檢查 Topic 的副本配置是否合理，若副本數大於可用 Broker 數，會導致 Leader 無法選舉。
- Kafka 消費者消費失敗，報 `No offset found for partition`
 - 若為新消費者組，首次消費無偏移量屬於正常情況，可設定 `auto.offset.reset` 為 `earliest` 或 `latest`。
 - 若為舊消費者組，檢查是否手動刪除過 ADCC for zk/ Kafka 中的偏移量資料，或消費者組長時間未消費導致偏移量過期（需調整 `offsets.retention.minutes` 配置）。
- ADCC for zk 叢集啟動後，部分節點無法加入叢集，日誌報 `Cannot open channel to x at election address`
 - 檢查所有 ADCC for zk 節點的 `zoo.cfg` 中 `server.x` 配置的 IP / 連線埠是否一致，無拼寫錯誤。
 - 檢查各節點之間的時間是否同步（時間差超過 2000ms 會導致選舉失敗），可透過 `ntpd` 同步時間。

- 確認 dataDir 目錄許可權為 ADCC for zk 執行使用者可讀寫，避免許可權不足導致 myid 檔案無法讀取。
- Kafka 日誌報 Not enough replicas，訊息無法持久化
 - 檢查 Topic 的 min.insync.replicas 配置值是否大於目前可用的同步副本數（ISR）。
 - 若為單 Broker 部署，需將 min.insync.replicas 改為 1（預設 1，若手動修改過需調整）。
 - 檢查 Follower 節點是否正常同步，排查網路延遲、磁碟 IO 過高導致同步慢的問題。
- Kafka 啟動後，JVM 記憶體溢位（OOM），程式直接崩潰
 - 檢查 kafka-server-start.sh 中的 JVM 配置（KAFKA_HEAP_OPTS），預設堆記憶體可能過大 / 過小。
 - 單機測試環境可調整為 -Xmx512M -Xms512M，生產環境根據伺服器記憶體配置（建議不超過物理記憶體的 50%）。
 - 檢查是否開啟了過多的 Topic / 分割區，導致記憶體佔用過高，需合理規劃分割區數。
- 連線 Kafka 時報 SSL handshake failed
 - 若開啟了 SSL 認證，檢查用戶端 / 服務端的 SSL 憑證路徑、密碼配置是否正確。
 - 確認憑證未過期，且服務端 listeners 配置為 SSL:// 而非 PLAINTEXT://。
 - 檢查用戶端是否正確配置了 SSL 相關參數（如 security.protocol=SSL）。

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。

Apusic
金蝶天燕

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

