



APUSIC  
固若长城  
睿比世界

# 配置参数说明文档

金蝶Apusic分布式消息队列V2.0.6

版权所有 © 深圳市金蝶天燕云计算股份有限公司2026。保留所有权利。

## 版权声明

本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。

本档如有更新，不另行通知。对本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 版本更新说明

## 2 ADMQ 配置参数说明

### •2.1 Broker配置参数

#### •2.1.1 认证参数

#### •2.1.2 通用参数

#### •2.1.3 第三方传输协议适配参数

##### •2.1.3.1 mqtt

##### •2.1.3.2 kafka

##### •2.1.3.3 amqp

##### •2.1.3.4 rocketmq

#### •2.1.4 TLS相关配置参数

#### •2.1.5 存储配置参数

#### •2.1.6 日志配置参数

#### •2.1.7 监控配置参数

### •2.2 Zookeeper配置参数

### •2.3 Storage配置参数

# 1 版本更新说明

本文档最新版本包含历史修改记录如下：

更新日期	手册版本	适用产品	更新说明
2025年12月	V1.0	金蝶Apsic分布式消息队列V2.0.6	首次编写

## 2 ADMQ 配置参数说明

### 2.1 Broker配置参数

#### 2.1.1 认证参数

- authenticationEnabled=true: 是否开启身份认证
- authenticationProviders=org.apache.pulsar.broker.authentication.AuthenticationProviderToken: 身份认证服务类配置
- authorizationEnabled=true: 是否开启权限认证
- authorizationProvider=org.apache.pulsar.broker.authorization.PulsarAuthorizationProvider: 权限验证服务类配置
- superUserRoles=admq: 超级管理员默认角色配置
- brokerClientAuthenticationPlugin=org.apache.pulsar.client.impl.auth.AuthenticationToken: 客户端与其他 broker/BookKeeper 通信时使用的认证插件类
- brokerClientAuthenticationParameters=token:eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1xLn0.ybJge7zTfy\_RDdAtB3w6nIPDHPT6-kbB6sNzgPt8sKQ: 客户端与其他 broker/BookKeeper 通信时使用的认证参数
- tokenSecretKey=config/auth/token.key: JWT认证的密钥文件相对地址

#### 2.1.2 通用参数

- allowAutoTopicCreationType=partitioned: 自动创建主题时的默认类型 (参数可选, partitioned表示默认创建分区主题, non-partitioned表示非分区主题), 需配合allowAutoTopicCreation=true生效, 否则无法创建成功
- allowAutoTopicCreation=false: 是否允许自动创建主题
- allowAutoSubscriptionCreation=true: 是否允许自动创建订阅
- brokerDeleteInactiveTopicsEnabled=false: 是否允许自动删除无效的主题
- brokerDeleteInactiveTopicsFrequencySeconds=60: 无效主题检测周期配置
- brokerDeleteInactiveTopicsMode=delete\_when\_no\_subscriptions: 无效主题判定规则设置 ('delete\_when\_no\_subscriptions': 没有订阅且没有活跃的生产者; 'delete\_when\_subscriptions\_caught\_up': 所有订阅都消费完了, 没有消息堆积, 而且没有生产者和消费者)
- forceDeleteTenantAllowed=false: 是否允许强制删除
- forceDeleteNamespaceAllowed=true: 是否允许强制删除命令空间
- subscriptionTypesEnabled=Exclusive,Shared,Failover,Key\_Shared: 允许使用的订阅模式
- subscriptionKeySharedUseConsistentHashing=false: key分发模式是否采用一致性hash算法
- subscriptionKeySharedConsistentHashingReplicaPoints=100: 一致性hash环的大小
- systemTopicEnabled=true: 是否开启系统主题
- topicLevelPoliciesEnabled=true: 是否开启主题级别的策略
- defaultMessageSize=5242880: 消息最大长度的设置 (默认5M)
- nettyMaxFrameSizeBytes=5242880: 消息最大长度的设置 (默认5M)
- defaultRetentionTimeInMinutes=240: 设置默认消息保留时间, 单位分钟
- defaultRetentionSizeInMB=16384: 设置默认消息保留大小, 单位MB
- messageTrackEnabled=true: 是否启用消息轨迹跟踪 (记录消息的生产、消费、存储等链路信息, 需配合日志或监控系统使用)
- managerUrl=<http://localhost:12305>: [管理控制台地址](#)

- metadataStoreUrl=0.0.0.0:2181: 集群元数据存储的 ZooKeeper 地址
- configurationStoreServers=: 全局配置存储的 ZooKeeper 地址
- brokerServicePort=6650: 数据服务监听端口
- brokerServicePortTls=6651: TLS数据服务监听端口
- webServicePort=8080: 管理服务监听端口
- webServicePortTls=8081: TLS管理服务监听端口
- bindAddress=0.0.0.0: 服务绑定的ip地址
- deployServerAddress=127.0.0.1: 本机IP
- advertisedAddress=127.0.0.1: 对外提供服务的IP地址
- isRunningStandalone=true: 是否单机模式启动
- clusterName=apusic-mq-standalone: 集群名称

### 2.1.3 第三方传输协议适配参数

- messagingProtocols=kafka,mqtt,amqp,rocketmq: 启用的第三方传输协议
- protocolHandlerDirectory=./protocols: 插件软件包所在的地址路径
- narExtractionDirectory=./nar: 插件nar包所在的地址路径

#### 2.1.3.1 mqtt

- mqttListeners=mqtt://127.0.0.1:1883: mqtt监听端口
- mqttProxyEnabled=true: 是否开启mqtt代理服务
- mqttProxyPort=5682: mqtt代理服务端口
- mqttAuthenticationEnabled=true: 是否开启mqtt身份认证
- mqttAuthorizationEnabled=true: 是否开启mqtt权限认证
- mqttAuthenticationMethods=token: mqtt身份认证方式配置（可填选Token或者Basic）

#### 2.1.3.2 kafka

- kafkaTenant=kafka-data: kafka 协议适配时，客户端消息对应的默认ADMQ租户
- kafkaMetadataTenant=kafka-meta: kafka元数据默认租户
- kafkaListeners=kafka://127.0.0.1:9092: kafka监听端口
- kafkaProtocolMap=kafka:PLAINTEXT: kafka插件协议名称，定义协议名称与安全协议的映射关系，确保客户端通过正确的协议连接ADMQ
- kafkaAdvertisedListeners=kafka://127.0.0.1:9092: kafka对外访问端口
- entryFormat=kafka: 用于控制存储条目的序列化格式保持kafka原始格式
- brokerEntryMetadataInterceptors=org.apache.pulsar.common.intercept.AppendIndexMetadataInterceptor: 元数据拦截器，以实现消息偏移量管理、消息追踪等功能
- kopEnableGroupLevelConsumerMetrics=true: 是否开启kafka消费者组监控
- saslAllowedMechanisms=: 设置kafka认证方式

#### 2.1.3.3 amqp

- amqpTenant=amqp-data: amqp协议适配时，客户端消息对应的默认ADMQ租户
- amqpMetadataTenant=amqp-meta: amqp默认元数据租户
- amqpListeners=amqp://127.0.0.1:5672: AMQP 协议的非 TLS 监听端口
- amqpMaxNoOfChannels=2047: amqp允许创建的最大通道数量

- amqpMaxFrameSize=4194304: AMQP 协议帧的最大大小 (单位字节, 4194304 = 4MB), 超过该大小的消息会被拒绝
- amqpProxyEnable=true: 是否开启amqp代理
- amqpProxyPort=6672: amqp代理端口
- amqpAdminPort=15673: amqp命令端口
- amqpAuthenticationEnabled=false: 是否开启amqp身份认证
- amqpAuthorizationEnabled=false: 是否开启amqp权限认证

#### 2.1.3.4 rocketmq

- rocketmqTenant=rocketmq-data: rocketmq 协议适配时, 客户端消息对应的默认ADMQ租户
- rocketmqMetadataTenant=rocketmq-meta: rocketmq元数据默认租户
- rocketmqListeners=rocketmq://127.0.0.1:9876: rocketmq监听端口
- rocketmqListenerPortMap=9876:rocketmq: rocketmq插件协议名称, 当客户端使用某个协议使用rocketmq协议处理对应的请求
- ropBrokerReplicationNum=1: RocketMQ 主题对应的ADMQ分区副本数
- ropTraceTopicEnable=false: 是否开启rocketmq主题轨迹
- ropRestServerPort=9888: rocketmq监控信息采集服务端口
- ropAclEnable=false: 是否开启rocketmq Acl

#### 2.1.4 TLS相关配置参数

- tlsEnabled=false: 是否启用 broker 的 TLS 加密 (true 表示开启, false 关闭)
- tlsRequireTrustedClientCertOnConnect=false: 是否要求客户端提供可信证书 (双向 TLS 认证, true 表示强制校验客户端证书)
- brokerClientTlsEnabled=false: broker 作为客户端与其他组件 (如 BookKeeper、ZooKeeper) 通信时是否启用 TLS 加密
- brokerClientTrustCertsFilePath=config/auth/ca.cert.pem: broker 客户端 TLS 认证的可信 CA 证书路径 (用于验证服务端证书)
- tlsKeyFilePath=config/auth/broker.key-pk8.pem: 服务端密钥证书路径
- tlsCertificateFilePath=config/auth/broker.cert.pem: 服务端证书路径
- tlsTrustCertsFilePath=config/auth/ca.cert.pem: 用于验证客户端证书的 CA 证书路径 (双向 TLS 时生效)
- tlsProtocols=TLSv1.2,TLSv1.1,TLSv1: TLS协议版本
- tlsCiphers=TLS\_DH\_RSA\_WITH\_AES\_256\_GCM\_SHA384,TLS\_DH\_RSA\_WITH\_AES\_256\_CBC\_SHA: TLS加密算法

#### 2.1.5 存储配置参数

- journalWriteData=true: 是否将消息数据写入 BookKeeper 的 Journal (预写日志, 确保数据持久化)
- journalSyncData=false: false 表示异步刷盘, 提升性能; true 表示同步刷盘, 确保数据不丢失, 牺牲性能)
- managedLedgerMaxEntriesPerLedger=5253120: 每个 Managed Ledger 的最大条目数 (达到阈值后创建新的 Ledger)

#### 2.1.6 日志配置参数

- logLevel=info: 日志等级
- logSize=250M: 单个日志文件的最大大小 (超过后滚动生成新文件)
- logKeepTime=30d: 日志保留时间
- logKeepSize=10G: 日志保留大小

## 2.1.7 监控配置参数

- monitor\_managerAddr=<http://127.0.0.1:12305>: 管理控制台监控地址
- monitor\_clusterName=apusic-mq: 监控集群名称
- monitor\_role=standalone: 监控目标的集群部署模式
- monitor\_serviceIp=127.0.0.1: 被监控的ADMQ Broker 服务 IP 地址 (即 Broker 绑定的 advertisedAddress 或 bindAddress, 用于监控系统定位目标服务)
- monitor\_servicePort=6650: 被监控的ADMQ数据服务端口 (对应 Broker 的 brokerServicePort, 仅用于标识监控目标的服务端口, 非监控指标暴露端口)
- monitor\_username= \_\_sys\_\_monitor: 监控默认用户
- monitor\_password=11111111: 监控默认密码

## 2.2 Zookeeper配置参数

- tickTime=10000: 每个计时单位所对应的毫秒数
- initLimit=10: 初始同步阶段允许的最大时间 (单位为 tickTime, 即  $10 * tickTime$ , 此处为  $10 * 10000ms = 100s$ ), 用于从节点同步主节点数据
- syncLimit=5: Leader 与 Follower 服务器之间正常交互的最长等待时间 (以tickTime为单位)
- dataDir=local\_zookeeper/data: zk数据存储目录
- dataLogDir=local\_zookeeper/logs: zk日志存储目录
- clientPort=2181: 服务端口
- server.1=192.168.0.1:2888:3888: ZooKeeper 集群节点配置 (格式: server.id=ip:通信端口:选举端口, 2888 用于主从通信, 3888 用于选举)
- admin.enableServer=true: 是否开启管理员监听服务
- admin.serverPort=9990: 管理员服务的监听端口 (通过 HTTP 提供 ZooKeeper 状态查询等功能)
- maxClientCnxns=300: 单个客户端 IP 允许的最大连接数
- autopurge.snapRetainCount=3: 在“dataDir”目录中保留的快照数量
- autopurge.purgeInterval=1: 清除任务间隔 (以小时为单位)
- forceSync=yes: 在完成更新处理之前, 是否需要将更新同步到事务日志的存储介质中
- sslQuorum=false: 是否在每个节点上启用强认证 TLS
- portUnification=false: 是否在客户端端口 (clientPort) 启用端口统一模式 (同时接受明文和 TLS 加密连接), false 表示仅接受明文
- metricsProvider.className=org.apache.zookeeper.metrics.prometheus.PrometheusMetricsProvider: 监控服务类
- metricsProvider.httpPort=18800: 监控服务端口
- metricsProvider.exportJvmInfo=true: 是否开启监控jvm

## 2.3 Storage配置参数

- bookiePort=3181: 存储服务监听端口
- journalDirectories=/admjournal: BookKeeper Journal (预写日志) 的存储目录 (用于持久化消息写入记录, 确保数据可靠性)
- ledgerDirectories=/admdata: BookKeeper Ledger (消息数据块) 的存储目录 (实际存储消息内容的目录)

- metadataServiceUri=zk+hierarchical://192.168.0.1:2181;192.168.0.2:2181;192.168.0.3:2181/admq/ledgers: BookKeeper 元数据存储的 ZooKeeper 地址 (zk+hierarchical 表示使用分层命名空间模式, 多个地址用分号分隔)
- clusterMode=cluster: BookKeeper 的运行模式 (cluster 表示集群模式, 单机模式可设为 standalone)
- clusterName=apusic-mq: 集群信息, 集群名称
- enableStatistics=false: 是否启用 BookKeeper 的统计指标收集 (用于监控读写性能、队列长度等)
- httpServerEnabled=true: 是否开启http服务, 用于管控台管理
- httpServerPort=8000: BookKeeper HTTP 管理服务的监听端口

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。

**Apusic**  
金蝶天燕

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](http://www.apusic.com)

