Apusic 金蝶天燕

生成时间：2026-05-07

命令行部署安装手册

金蝶Apusic分布式消息队列V2.0.6

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

Apusic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 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Apusic分布式消息队列V2.0.6	首次编写

2 命令行安装手册

2.1 概述说明

ADMQ包括管控台和引擎两部分，通常手动部署管控台，然后通过管控台部署引擎。但在某些场景下需要单独部署引擎，然后通过其他方式管理或者注册到ADMQ管控台管理。

本文档给出命令行方式部署ADMQ引擎的流程，用户可依据此文档完成ADMQ单机和集群的部署。

2.2 前期准备工作

在部署实施之前，首先获取软件包，准备部署环境。

软件环境

软件	*作用*	*其他依赖说明*
admq-manager-Vx.x.x	admq管控台，提供集群、主题等资源管理功能	JDK：JDK17数据库：默认使用h2内存数据库，生产环境建议使用国产数据库或mysql、postgressql等数据库
admq-Vx.x.x	admq核心引擎包，提供admq、rocketmq、kafka、rabbitmq、mqtt和jms等类型的消息收发服务。	JDK：jdk1.8及以上
rocketmq-Vx.x.x	rocketmq安装包，只提供rocketmq服务	JDK：jdk1.8及以上
jdk1.8及以上	软件运行依赖环境	
mysql 5.7.26及以上或其他数据库服务	提供管控台数据存储服务	可选。

2.2.1 系统要求

在开始部署之前，请确保您的环境满足以下要求：

- **操作系统**：64 位 Linux（推荐 CentOS 7+、Ubuntu 18.04+，国产Kylin v10+、OpenEuler 22.03+）
- **Java 版本**：推荐 JDK 17
- **硬件配置**：
    - 最小生产环境需要 6 台服务器（或虚拟机），推荐 9 台
        - 3 台运行 ZooKeeper（可以根据业务规模增加到5台）

        - 3 台运行 ADMQ Storage 和 Broker（推荐 Storage 和 Broker 各 3 台，也可以根据业务规模增加到5台或以上）
    - 测试环境可以在一台服务器上运行所有组件
- **网络**：所有节点之间可以相互通信，并具有稳定的网络连接
- **DNS**：可选，为所有 Broker 节点配置一个共同的 DNS 名称

2.2.2 硬件推荐

下表提供了不同规模集群的硬件推荐（每节点）：

组件	小型集群 (250 个主题)	中型集群 (1000 个主题)
ZooKeeper	• CPU: 4 核 • 内存: 16GB • 磁盘: 200GB SSD	• CPU: 8 核 • 内存: 32GB • 磁盘: 500GB SSD
Storage	• CPU: 8 核 • 内存: 32GB • 磁盘: 1TB SSD	• CPU: 16 核 • 内存: 64GB • 磁盘: 2TB SSD
Broker	• CPU: 8 核 • 内存: 32GB • 磁盘: 500GB SSD	• CPU: 16 核 • 内存: 64GB • 磁盘: 1TB SSD

硬件选择建议

- ZooKeeper 只用于元数据存储，可以使用相对较小的服务器
- Storage需要快速的存储设备，推荐使用 SSD 或带电池缓存的 RAID 控制器
- Broker 需要快速的 CPU 和网络，推荐使用 10Gbps 网络接口

2.3 部署管控台

管控台（admq-manager）是一个spring boot程序，可以直接解压运行。

获取软件包后，解压并进入解压目录

tar zxvf admq-manager-Vx.x-SNAPSHOT.tar.gz

cd admq-manager-Vx.x-SNAPSHOT

其中的目录及作用如下：

目录	*作用*	*其他说明*
bin	启动、停止等脚本目录	
config	配置文件目录，核心配置文件为application.properties	
data	数据存储目录，报错上传的软件包、部署配置等信息	

libs	程序依赖的所有jar包	
licenses	存放license文件	
logs	存放程序运行日志文件	
package	存放引擎服务安装包，程序第一次启动时会自动加载该目录中的安装包	
ui	前端页面文件	

修改配置文件

进入config目录，打开application.properties文件，修改数据库配置，选择对应使用的数据库，然后再打开对应config/other-config/路径下具体的数据库配置文件，修改服务端口和数据库地址等信息。常用的配置如下：

配置名称	*作用*	*默认值*
server.port	管控台访问端口，仅支持https访问	12306
server.http.port	管控台访问端口，仅支持http访问。建议不要使用此端口访问，容易造成安全	12305

	问题。	
config.dir	配置文件目录	./config/
spring.datasource.driver-class-name	使用的驱动名称，建议使用mysql等	org.h2.Driver
spring.datasource.url	数据库链接地址	jdbc:mysql://ip:port?useUnicode=true&characterEncoding=utf-8&allowMultiQueries=true&useSSL=false
spring.sql.init.schema-locations	数据库初始化sql文件路径	classpath:/META-INF/sql/h2-schema.sql
spring.datasource.username	数据库用户名称	sa
spring.datasource.password	数据库用户密码。使用ENC则表示加密，否则不加密。	ENC(79ZwI6lMamz8n3KxETrheBUA5uuWDwevUF6ZWIE1X4jXvJWQ+m

主要是修改数据库配置（测试环境可以不修改）。

启动

bin/admq-manager start

检查状态

浏览器访问http://ip:12305或者https://ip:12306，初始用户名密码：admq/11111111

停止

bin/admq-manager stop

2.4 部署ADMQ核心引擎

核心引擎包含三个组件：计算组件（broker）、存储组件（storage）和协调器组件，单机部署时三个组件的功能集成在一个进程中，不需要单独部署。但是集群部署时，需要单独部署、启动三个组件。

计算组件提供客户端连接服务，接收到消息后存储到存储组件中。协调器则负责保存topic等元数据信息。

2.4.1 单机部署

解压软件包并进入安装目录

tar zxvf admq-Vx.x.x-all.tar.gz

cd admq-Vx.x.x

修改配置文件

进入config目录，打开standalone.conf文件。常用的配置如下：

配置名称	*作用*	*默认值*
webServicePort	管理服务端口	8080
deployServerAddress	本机IP	127.0.0.1（需更改为本机IP）
advertisedAddress	本机IP	127.0.0.1（需更改为本机IP）
clusterName	集群名称	apusic-mq-standalone

放置License文件

把license文件拷贝到对应根目录。

启动

bin/admq-daemon start standalone

执行命令后，会出现部分在未节点启动前保存的日志内容和部分新增的日志内容，出现的部分警告和报错是正常现象，实际根据检查状态的命令确定是否正常。

检查状态

netstat -nltp检查端口是否正常启动

bin/admqctl admin clusters list

如果有结果则表示正常。

停止

bin/admq-daemon stop standalone

2.4.2 集群部署

组件部署顺序：协调器、存储组件、计算组件。

集群部署至少需要三台服务器，可以在每台服务器上都部署三种类型的组件。

2.4.2.1 部署协调器

协调器的内核是zookeeper，主要用于保存集群元数据信息。

创建部署目录并拷贝安装包

mkdir /admq/Vx.x.x/zk

cp admq-Vx.x.x-all.tar.gz /admq/Vx.x.x/localZk

解压软件包并进入目录

tar zxvf admq-Vx.x.x-all.tar.gz

cd admq-Vx.x.x

创建数据存储目录并生成节点ID（myid文件）

mkdir -p local_zookeeper/data

mkdir -p local_zookeeper/logs

echo 1 > local_zookeeper/data/myid

注：三个节点的id不能相同，而且需要和配置文件中保持一致。一般设置为1、2、3

修改配置文件

进入conf目录，打开zookeeper.conf文件。常用的配置如下：

配置名称	*作用*	*默认值*
clientPort	服务端口	2181
server.1	第一个节点的地址	192.168.0.1:2888:3888
server.2	第二个节点的地址	192.168.0.2:2888:3888
server.3	第三个节点的地址	192.168.0.3:2888:3888
forceSync	是否同步刷新，如果磁盘读写能力不好可设置成false，避免因此连接超时	true
metricsProvider.httpPort	Prometheus监控端口	18800

启动

bin/admq-daemon start zookeeper

停止

bin/admq-daemon stop zookeeper

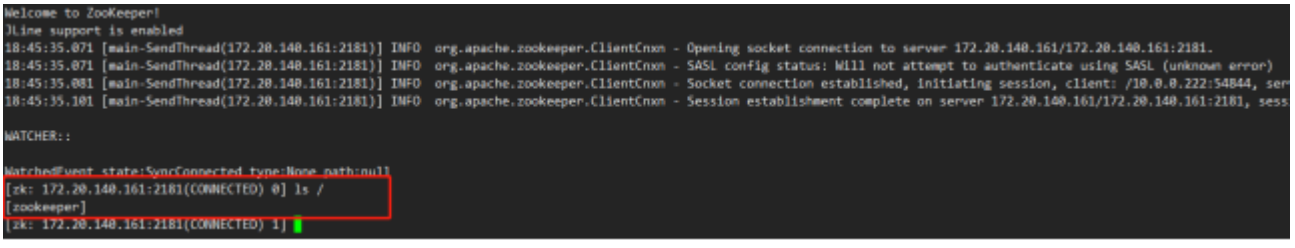
修改另外两个节点的配置并启动

只有myid的内容有区别，分别为1、2、3，其他都一样。

检查状态

bin/admqctl zk-cli -server ip1:2181

能连接则表示正常。



2.4.2.2 初始化元数据

需要把集群信息提前写入协调器中，执行以下命令：

bin/admqctl initialize-cluster-metadata \

--cluster apusic-mq \

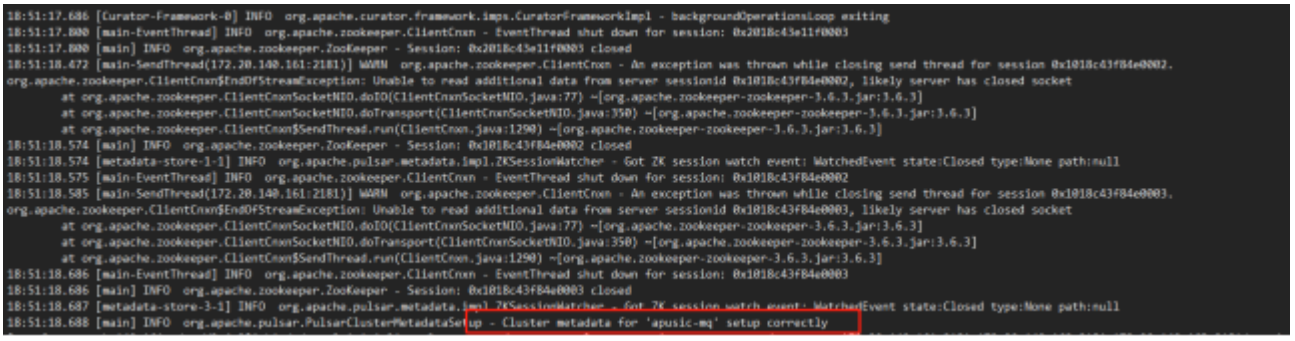
--metadata-store 172.20.140.161:2181,172.20.140.162:2181,172.20.140.163:2181/apusic-mq \

--configuration-metadata-store 172.20.140.161:2181,172.20.140.162:2181,172.20.140.163:2181/apusic-mq \

--web-service-url http://172.20.140.161:8080,172.20.140.162:8080,172.20.140.163:8080 \

--broker-service-url pulsar://172.20.140.161:6650,172.20.140.162:6650,172.20.140.163:6650

出现"Cluster metadata for 'apusic-mq' setup correctly"则表示执行成功



2.4.2.3 部署存储组件

存储组件主要用于保存集群数据。

创建部署目录并拷贝安装包

mkdir /admq/Vx.x.x/storage

cp admq-Vx.x.x-all.tar.gz /admq/Vx.x.x/storage

解压软件包并进入目录

tar zxvf admq-Vx.x.x-all.tar.gz

cd admq-Vx.x.x

放置License文件

把license文件拷贝到对应根目录。

修改配置文件

进入conf目录，打开storage.conf文件。常用的配置如下：

配置名称	*作用*	*默认值*
bookiePort	对外提供的服务端口	3181
deployServerAddress	本机IP	127.0.0.1（需更改为本机IP）
advertisedAddress	本机IP	127.0.0.1（需更改为本机IP）
journalDirectories	数据存储目录	/admq_journal
ledgerDirectories	数据存储目录	/admq_data
metadataServiceUri	协调器连接地址，其中的admq换成集群名称	zk+hierarchical://192.168.0.1:2181;192.168.0.2:2181;192.168.0.3:2181/admq
clusterName	集群名称；需要和元数据初始化中的保持一致	apusic-mq
httpServerPort	Prometheus监控服务端口	8000

需要创建journalDirectories和ledgerDirectories配置的目录，或者把上述配置改成已有目录。

启动

bin/admq-daemon start storage

停止

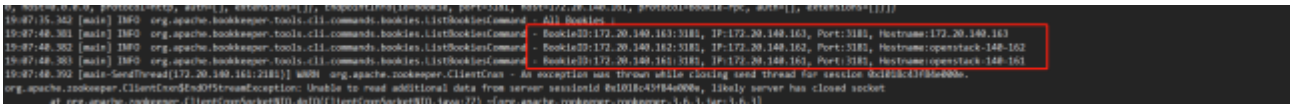
bin/admq-daemon stop storage

修改另外两个节点的配置并启动

只有IP地址有区别，其他都一样。

检查状态

bin/admqctl store-shell listbookies -a



可以查询到三个storage节点。

2.4.2.4 部署计算组件

计算组件主要用于提供客户端连接服务。

创建部署目录并拷贝安装包

mkdir /admq/Vx.x.x/broker

cp admq-Vx.x.x-all.tar.gz /admq/Vx.x.x/broker

解压软件包并进入目录

tar zxvf admq-Vx.x.x-all.tar.gz

cd admq-Vx.x.x

放置License文件

把license文件拷贝到对应根目录。

修改配置文件

进入conf目录，打开broker.conf文件。常用的配置如下：

配置名称	*作用*	*默认值*
metadataStoreUrl	协调器地址；需要把地址换成实际地址，同时把admq换成集群名称（和元数据初始化时保持一致）。	192.168.0.1:2181,192.168.0.2:2181,192.168.0.3:2181/admq
webServicePort	管理服务端口	8080
deployServerAddress	本机IP	127.0.0.1（需更改为本机IP）
advertisedAddress	本机IP	127.0.0.1（需更改为本机IP）
clusterName	集群名称	apusic-mq

启动

bin/admq-daemon start broker

停止

bin/admq-daemon stop broker

修改另外两个节点的配置并启动

只有IP地址有区别，其他都一样。

检查状态

bin/admqctl admin brokers list apusic-mq

其中apusic-mq是集群名称。

```
[root@openstack-140-161 admq-V2.4.2]# bin/admqctl admin brokers list apusic-mq
172.20.140.161:8080
172.20.140.162:8080
172.20.140.163:8080
[root@openstack-140-161 admq-V2.4.2]#
```

2.4.3 伪集群部署

在有些场景下需要在1台或者2台服务器上部署admq，这时候就会出现同一台服务器上多个相同的组件同时在运行。这种方式虽然也是集群部署，但可靠性相对于多台服务器部署的集群要低很多，因为服务器故障后会导致整个集群不可用。

下面针对单台服务器部署集群的方式进行说明。

2.4.3.1 创建目录并拷贝安装包

多服务器部署时，每一类进程在每台服务器上的目录基本都是相同的，但是单服务器部署集群时需要每个进程一个单独的目录。

部署一个三节点集群时，对于协调器、计算节点、存储节点分别创建对应的目录，如下：

mkdir -p node1/localZk

mkdir -p node2/localZk

mkdir -p node3/localZk

mkdir -p node1/broker

mkdir -p node2/broker

mkdir -p node3/broker

mkdir -p node1/storage

mkdir -p node2/storage

mkdir -p node3/storage

目录创建完成后解压安装包并把解压后的文件拷贝到node1目录中：

tar zxvf admq-Vx.x.x-all.tar.gz

cp -r admq-Vx.x.x node1/localZk

cp -r admq-Vx.x.x node1/broker

cp -r admq-Vx.x.x node1/storage

2.4.3.2 修改节点配置

伪集群部署时大部分配置都是相同的，因此可以只修改node1下三个节点的配置，然后把配置拷贝到node2和node3目录中。

修改协调器配置

首先创建存储目录

mkdir -p node1/localZk/admq-Vx.x.x/local_zookeeper/data

mkdir -p node1/localZk/admq-Vx.x.x/local_zookeeper/logs

生成节点标识：myid文件

echo 1 > node1/localZk/admq-Vx.x.x/local_zookeeper/data/myid

修改配置文件，修改其中的ip和端口，具体修改项如下

vi node1/localZk/admq-Vx.x.x/config/zookeeper.conf

配置名称	*默认值*	*修改后的值*
server.1	192.168.0.1:2888:3888	主机IP:12888:13888
server.2	192.168.0.2:2888:3888	主机IP:22888:23888
server.3	192.168.0.3:2888:3888	主机IP:32888:33888
clientPort	2181	12181

修改计算节点配置

vi node1/broker/admq-Vx.x.x/config/broker.conf

配置名称	*默认值*	*修改后的值*
metadataStoreUrl	192.168.0.1:2181,192.168.0.2:2181,192.168.0.3:2181/admq	主机IP:12181, 主机IP:22181, 主机IP:32181/apusic-mq
brokerServicePort	6650	16650
webServicePort	8080	18080
advertisedAddress	127.0.0.1	主机IP
deployServerAddress	127.0.0.1	主机IP

修改存储节点配置

vi node1/storage/admq-Vx.x.x/config/storage.conf

配置名称	*默认值*	*修*
advertisedAddress	127.0.0.1	主机
deployServerAddress	127.0.0.1	主机
bookiePort	3181	1318
journalDirectories	/admq_journal	data
ledgerDirectories	/admq_data	data
metadataServiceUri	zk+hierarchical://192.168.0.1:2181;192.168.0.2:2181;192.168.0.3:2181/admq/ledgers	zk+ 主机 IP:2 IP:3 mq/
httpServerPort	8000	180

2.4.3.3 拷贝安装包到其他节点目录并修改端口

可以把node1下三个组件所有配置拷贝到node2和node3下，然后再修改端口即可。

cp -r node1/* node2/

cp -r node1/* node3/

修改端口等信息，解决冲突问题

修改协调器端口

vi node2/localZk/admq-Vx.x.x/config/zookeeper.conf

配置名称	*当前值*	*修改后的值*
clientPort	12181	22181
admin.serverPort	19990	29990
metricsProvider.httpPort	18800	28800

vi node3/localZk/admq-Vx.x.x/config/zookeeper.conf

配置名称	*当前值*	*修改后的值*
clientPort	12181	32181
admin.serverPort	19990	39990
metricsProvider.httpPort	18800	38800

生成myid文件

echo 2 > node2/localZk/admq-Vx.x.x/local_zookeeper/data/myid

echo 3 > node3/localZk/admq-Vx.x.x/local_zookeeper/data/myid

修改计算节点端口

vi node2/broker/admq-Vx.x.x/config/broker.conf

配置名称	*当前值*	*修改后的值*
brokerServicePort	16650	26650
webServicePort	18080	28080

vi node3/broker/admq-Vx.x.x/config/broker.conf

配置名称	*当前值*	*修改后的值*
brokerServicePort	16650	36650
webServicePort	18080	38080

修改存储节点端口

vi node2/storage/admq-Vx.x.x/config/storage.conf

配置名称	*当前值*	*修改后的值*
bookiePort	13181	23181
httpServerPort	18000	28000

vi node3/storage/admq-Vx.x.x/config/storage.conf

配置名称	*当前值*	*修改后的值*
bookiePort	13181	33181
httpServerPort	18000	38000

2.4.3.4 启动协调器并初始化集群

分别进入三个协调器目录执行启动命令：

cd node1/localZk/admq-Vx.x.x/ && bin/admq-daemon start zookeeper

cd node2/localZk/admq-Vx.x.x/ && bin/admq-daemon start zookeeper

cd node3/localZk/admq-Vx.x.x/ && bin/admq-daemon start zookeeper

初始化集群

cd node1/localZk/admq-Vx.x.x

bin/admqctl initialize-cluster-metadata \

--cluster apusic-mq \

--zookeeper 主机IP:12181,主机IP:22181,主机IP:32181/apusic-mq \

--configuration-store 主机IP:12181,主机IP:22181,主机IP:32181/apusic-mq \

--web-service-url http://主机IP:18080,主机IP:28080,主机IP:38080 \

--broker-service-url pulsar://主机IP:16650,主机IP:26650,主机IP:36650

2.4.3.5 启动存储节点和计算节点

集群初始化完成后即可启动存储和计算节点，先启动存储在启动计算节点。

cd node1/storage/admq-Vx.x.x/ && bin/admq-daemon start storage

cd node2/storage/admq-Vx.x.x/ && bin/admq-daemon start storage

cd node3/storage/admq-Vx.x.x/ && bin/admq-daemon start storage

cd node1/broker/admq-Vx.x.x/ && bin/admq-daemon start broker

cd node2/broker/admq-Vx.x.x/ && bin/admq-daemon start broker

cd node3/broker/admq-Vx.x.x/ && bin/admq-daemon start broker

2.4.3.6 验证是否启动成功

执行以下命令可分别查看对应节点进程是否存在

ps -ef |grep MQZookeeper

ps -ef |grep MQStorage

ps -ef |grep MQBroker

随便进入一个安装目录，执行命令查看租户状态，如果正常输出则表示集群正常

（查看前需要修改node1/config/client.conf中webServiceUrl 和brokerServiceUrl，确保IP和端口号正确） cd node1/broker/admq-Vx.x.x

bin/admqctl admin tenants list

2.4.4 加载插件

通过修改config/broker.conf文件加载不同的插件。通用配置如下：

配置名称	*作用*	*默认值*
messagingProtocols	需要开启的插件；支持：kafka,mqtt,amqp,rocketmq四种	
protocolHandlerDirectory	插件目录	./protocols
narExtractionDirectory	插件解压目录	./nar

如果需要加载插件，首先要把上述配置加到standalone.conf或者broker.conf中。**添加前先检查配置项是否存在，如果存在则直接修改。**

加载插件后需要重启broker节点。

2.4.4.1 Kafka插件

配置名称	*作用*	*默认值*
kafkaTenant	数据保存租户	kafka-data
kafkaMetadataTenant	元数据保存租户	kafka-meta
kafkaListeners	监听地址	kafka://127.0.0.1:9092
kafkaProtocolMap	协议转换地址	kafka:PLAINTEXT
kafkaAdvertisedListeners	对外暴露地址	kafka://127.0.0.1:9092
entryFormat	数据转换类型	
brokerEntryMetadataInterceptors	数据存储之前的拦截器	空
allowAutoTopicCreationType	自动创建的topic类型，需要设置成：partitioned	空
kopEnableGroupLevelConsumerMetrics	是否开启订阅组监控	true
saslAllowedMechanisms	设置成PLAIN则开启sasl认证，设置成空则不开启	空

下面是一组示例配置：

kafkaTenant=kafka-data

kafkaMetadataTenant=kafka-meta

kafkaListeners=kafka://172.24.4.217:9092

kafkaProtocolMap=kafka:PLAINTEXT

kafkaAdvertisedListeners=kafka://172.24.4.217:9092

entryFormat=kafka

brokerEntryMetadataInterceptors=org.apache.pulsar.common.intercept.AppendIndexMetadataInterceptor

allowAutoTopicCreationType=partitioned

kopEnableGroupLevelConsumerMetrics=true

saslAllowedMechanisms=

2.4.4.2 AMQP(RabbitMQ)插件

配置名称	*作用*	*默认值*
amqpTenant	数据保存租户	amqp-data
amqpMetadataTenant	元数据保存租户	amqp-meta
amqpListeners	监听地址	amqp://127.0.0.1:5672
amqpMaxNoOfChannels	最大channel数量	2047
amqpMaxFrameSize	单个frame最大长度	4194304
amqpProxyEnable	是否开启代理	true
amqpProxyPort	代理服务端口	6672
amqpAuthenticationEnabled	是否开启认证	false
amqpAuthorizationEnabled	是否开启授权	false
amqpAdminPort	管理服务端口	15673
loadBalancerAutoBundleSplitEnabled	是否开启bundle自动分裂；需要设置成true	false
loadBalancerAutoUnloadSplitBundlesEnabled	是否开启开启bundle自动卸载；需要设置成true	false

下面是一组示例配置：

amqpTenant=amqp-data

amqpMetadataTenant=amqp-meta

amqpListeners=amqp://172.24.4.217:5672

amqpMaxNoOfChannels=2047

amqpMaxFrameSize=4194304

amqpProxyEnable=true

amqpProxyPort=6672

amqpAuthenticationEnabled=false

amqpAuthorizationEnabled=false

amqpAdminPort=15673

loadBalancerAutoBundleSplitEnabled=false

loadBalancerAutoUnloadSplitBundlesEnabled=false

2.4.4.3 MQTT插件

配置名称	*作用*	*默认值*
mqttListeners	监听地址	mqtt://127.0.0.1:1883
mqttProxyEnabled	是否开启代理	true
mqttProxyPort	代理端口	5682

下面是一组示例配置：

mqttListeners=mqtt://172.20.140.140:1883

mqttProxyEnabled=true

mqttProxyPort=5682

2.5 部署RocketMQ引擎

2.5.1 单机部署

单机需要启动一个nameserver和一个broker，参考下面集群部署。

2.5.2 集群部署

集群部署建议至少需要启动2个nameserver，启动三个broker。

在每个broker中配置多个nameserver的地址。

2.5.2.1 部署namesrv

主要用于保存集群元数据信息。

创建部署目录并拷贝安装包

mkdir /admq/Vx.x.x/nameserver

tar zxvf rocketmq-Vx.x.x-all.tar.gz

cp rocketmq-Vx.x.x-all.zip /admq/Vx.x.x/nameserver

解压软件包并进入目录

unzip rocketmq-Vx.x.x-all.zip

cd rocketmq-Vx.x.x-all

放置License文件

mkdir admq-licenses

把license文件拷贝到admq-licenses目录。

修改配置文件

进入conf目录，打开namesrv.properties文件。常用的配置如下：

配置名称	*作用*	*默认值*
listenPort	监听端口	9876
deployServerAddress	本机IP	127.0.0.1

启动

bin/rocketmq start nameserver

停止

bin/rocketmq stop nameserver

2.5.2.2 部署broker

主要用于保存集群元数据信息。

创建部署目录并拷贝安装包

mkdir /admq/Vx.x.x/broker

tar zxvf rocketmq-Vx.x.x-all.tar.gz

cp rocketmq-Vx.x.x-all.zip /admq/Vx.x.x/broker

解压软件包并进入目录

unzip rocketmq-Vx.x.x-all.zip

cd rocketmq-Vx.x.x-all

放置License文件

mkdir admq-licenses

把license文件拷贝到admq-licenses目录。

修改配置文件

进入conf目录，打开broker.conf文件。常用的配置如下：

配置名称	*作用*	*默认值*
listenPort	监听端口	10911
deployServerAddress	本机IP	127.0.0.1
namesrvAddr	nameserver地址	空，配置多个nameserver服务器时使用ip1:port;ip2:port;ip3:port形式
storePathRootDir	数据保存目录	空
storePathCommitLog	数据保存目录	空
brokerClusterName	集群名称	defaultCluster
aclEnable	是否开启认证	true
brokerName	计算节点名称，主从模式下，属于同一个主节点的所有从节点，名字必须和主节点相同；不同节点的名称不能相同，可以是broker-a，broker-b，broker-c	broker-a
brokerId	主从模式下，各个计算节点的id，0表示主节点，非0表示从节点	
restServerPort	监控端口	9888

下面是一组示例配置

deleteWhen=01 #每天凌晨1点执行磁盘文件的清理任务

fileReservedTime=72 #消息文件在磁盘上最多保留72小时，超过时间即使未消费也会被删除

brokerRole=ASYNC_MASTER #当前节点角色，async_master异步主节点

diskMaxUsedSpaceRatio=85 #当磁盘使用率超过85%，节点拒绝写入新消息

flushDiskType=ASYNC_FLUSH #异步刷盘，指定刷盘类型

tenantRateLimit=true #是否启用租户级别流控

sendRate=50000 #每秒最大发送消息数量

receiveRate=150000 #每秒最大拉去消息数量

autoCreateTopicEnable=false #是否启用自动创建主题

autoCreateSubscriptionGroup=false #是否启用自动创建订阅

aclEnable=true #是否开启acl权限认证

traceTopicEnable=true #是否启用主题消息轨迹跟踪（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）

brokerClusterName=rocketmq #broker节点集群名称

brokerName=broker-0 #节点id名称

brokerId=0 #broker节点id

restServerPort=9888 #rest协议服务端口

listenPort=10911 #监听端口

deployServerAddress=172.24.4.217

managerUrl=http://172.24.4.216:12307 #授权管理控制台IP地址

tls.enable=false #是否启用TLS加密通信

tls.server.keyPath=conf/auth/server.key #TLS认证服务端私钥文件路径

tls.server.keyPassword=123456 #TLS服务端私钥密码

tls.server.certPath=conf/auth/server.pem #broker节点证书文件（由CA签发）

tls.server.trustCertPath=conf/auth/ca.pem #受信任的CA根证书，用于验证客户端证书

tls.client.keyPath=conf/auth/client.key #客户端认证连接使用的私钥文件地址

tls.client.keyPassword=123456 #客户端私钥密钥

tls.client.certPath=conf/auth/client.pem #broker的客户端证书文件

tls.client.trustCertPath=conf/auth/ca.pem #用于验证服务端证书的CA根证书

namesrvAddr=172.24.4.217:9876

storePathRootDir=/admq/v2.4/data/rocketmq/rocketmq_broker/1gX49DA8

storePathCommitLog=/admq/v2.4/data/rocketmq/rocketmq_broker/1gX49DA8/commitlog

配置ACL文件（可选）

如果上边配置中开启了acl，则需要配置acl文件。修改conf/plan_acl.yml文件，如下：

accounts:

- accessKey: eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1xIn0.ybJge7zTfy_RDdAtB3w6nlPDHPT6-kbB6sNzgPt8sKQ （需根据实际使用生成对应的字符串）

secretKey: admq123456

admin: true

globalWhiteRemoteAddresses:

- 172.20.140.161

- 172.20.140.162

- 172.20.140.163

最下面的白名单建议配置broker和nameserver的ip地址。

启动

bin/rocketmq start broker

停止

bin/rocketmq stop broker

2.5.2.3 主从部署

主从模式和集群模式类似，在启动主节点的前提下在部署从节点即可。

从节点的brokerName和主节点保持一致，brokerId从1递增。

配置完成后启动：

bin/rocketmq start broker

2.5.2.4 验证是否启动成功

检查Nameserver是否启动

查看进程 jps | grep NamesrvStartup 或者检查端口（默认9876）

检查broker是否注册到Nameserver

bin/mqadmin clusterList -n localhost:9876

如内置的tools.sh测试消息发送接收

export NAMESRV_ADDR=127.0.0.1:9876

bin/tools.sh org.apache.rocketmq.example.quickstart.Consumer (-n localhost:9876 -t topic)

bin/tools.sh org.apache.rocketmq.example.quickstart.Producer (-n localhost:9876 -t topic)

2.6 管控台注册引擎服务

引擎服务部署好后可以直接通过集群注册功能注册服务，在【集群管理】页面选择【非容器化注册】按照提示内容填写即可，或者查看用户手册中对应说明。

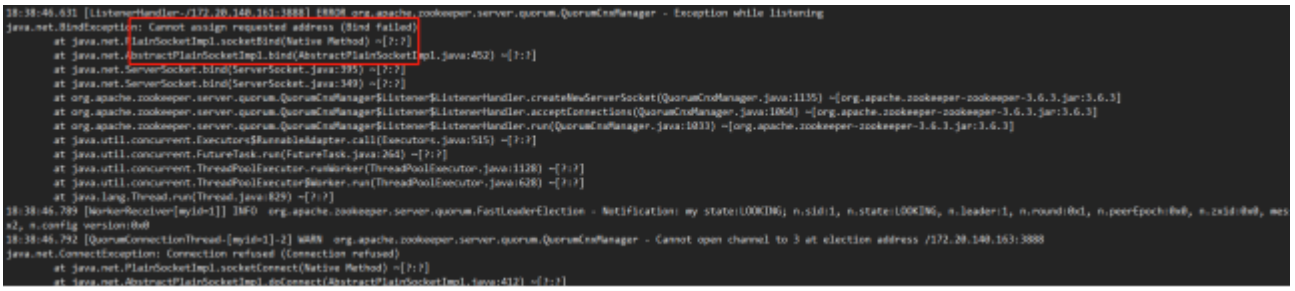
如果不想管理集群的启停和配置，则可以选择【容器化注册】。

2.7 常见问题

2.7.1 协调器启动失败

1、服务器多IP

查看logs/zookeeper—xx.log日志，出现以下日志则是该问题：



解决办法

修改conf/zookeeper.conf，把对应server的ip改成0.0.0.0，如下：

```
server.1=0.0.0.0:2888:3888
server.2=172.20.140.162:2888:3888
server.3=172.20.140.163:2888:3888
```

同理，在另外两台服务器上分别把对应id的ip改成0.0.0.0

2.7.2 rocketmq nameserver启动失败

1、jdk版本过高

关键词：

Unrecognized VM option 'UseParNewGC'

Unrecognized VM option 'UseGCLogFileRotation'

Unrecognized VM option 'UseCMSCompactAtFullCollection'

如：

```
[root@openstack-140-161 rocketmq-V4.9.2-all]# bin/rocketmq start nameserver
starting nameserver, logging to /admq/v2.4/rocketmq-V4.9.2-all/logs/namesrv.log
OpenJDK 64-Bit Server VM warning: Ignoring option PermSize; support was removed in 8.0
OpenJDK 64-Bit Server VM warning: Ignoring option MaxPermSize; support was removed in 8.0
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
Unrecognized VM option 'UseCMSCompactAtFullCollection'
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.
```

出现以上日志，则需要修改启动脚本：

bin/runserver.sh和bin/runbroker.sh中去掉上述不支持的jvm参数即可。

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