

文档说明

目录结构

安装要求

安装启动

安装步骤

修改配置文件

启动顺序

启动脚本

停止脚本

创建Topic

创建消费组

MQ服务器数据迁移

文档说明

本文档主要为RocketMQ的简介及安装部署手册，为后续运维人员部署RocketMQ提供规范和操作说明，请部署人员严格按照部署文档的操作规范进行部署，不能随意更改除指定更改点之外的地方，如确实需要更改，请与技术组相关负责人联系；

目录结构

目录	说明
\$HOME/store	Topic、消费组的配置数据和消息持久化数据
\$ROCKETMQ_HOME/bin	namesrv、broker的启动脚本，运维脚本
\$ROCKETMQ_HOME/conf	namesrv、broker的配置文件
\$HOME/logs/rocketmqlogs	namesrv、broker运行时的日志文件

- 环境变量\$HOME为用户目录
- 环境变量\$ROCKETMQ_HOME为rocketmq安装包的根目录，环境变量设置可参看 [安装步骤第2节环境变量设置](#)

安装要求

- 1.主机环境要求：
- 64bit OS, Linux/Unix

系统内存16GB（最低建议）

硬盘内存64GB（最低建议）
- 2.RocketMQ版本：
- 统一使用技术组提供的版本进行安装，核心版本为Apache RocetMQ release v4.1.0-incubating，禁止私自从github私自编译及从官网下载。

3.JDK版本：

64bit JDK 1.8（推荐）

64bit JDK 1.7（最低版本要求）

注：如果各分公司对JDK版本有严格规范要求，必须要求使用1.7的话，可使用JDK1.7，否则建议使用1.8

4.Rocketmq的JVM参数要求：

jdk1.7

Namesrv: "-server -Xms4g -Xmx4g -Xmn2g -XX:PermSize=128m -XX:MaxPermSize=320m"

Broker: "-server -Xms8g -Xmx8g -Xmn4g -XX:PermSize=128m -XX:MaxPermSize=320m"

jdk1.8

Namesrv: "-server -Xms4g -Xmx4g -Xmn2g -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"

Broker: "-server -Xms8g -Xmx8g -Xmn4g -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m"

注:安装包已配置好相关参数，请勿更改，直接启动即可，如果JVM参数需要调整，请与技术组确认。

安装启动

安装步骤

步骤列表：

步骤	操作	说明
1	下载&解压	根据部署mq的主机上安装的jdk版本下载相应指定的RocketMQ版本，建议在生产环境部署jdk8版本的RocketMQ，下载地址见下载安装解压包模块
2	ROCKETMQ_HOME	设置RocketMQ环境变量
3	修改配置文件	修改指定字段的属性
4	启动&停止	启动和停止namesrv和broker

1. 下载解压安装包

请先确定安装Rocketmq的主机上面jdk版本，根据jdk版本进行下载最新版本的rocketmq-4.1.0-masterA.tar.gz、rocketmq-4.1.0-masterB.tar.gz，具体下载信息如下：

jdk版本	下载地址
jdk7	http://192.168.100.9:8090/pages/viewpage.action?pageId=9143805
jdk8	http://192.168.100.9:8090/pages/viewpage.action?pageId=9143805

直接解压rocketmq-4.1.0-masterA.tar.gz、rocketmq-4.1.0-masterB.tar.gz到用户宿主目录下，解压完成后，不要改rocketmq根目录名称，如：

IP	用户	Master实例	角色
192.169.100.152	itframe	\$HOME/apache-rocketmq-masterA	master-A
192.169.100.153	itframe	\$HOME/apache-rocketmq-masterB	master-B

示例命令如下：

```
tar -zxvf apache-rocketmq-masterA.tar.gz
或
tar -zxvf apache-rocketmq-masterB.tar.gz
```

2. 设置环境变量

分别在机器A、B上设置ROCKETMQ_HOME环境变量，并加到PATH中

示例命令如下：

```
#设置环境变量
export ROCKETMQ_HOME=/home/itframe/apache-rocketmq-masterA
或
export ROCKETMQ_HOME=/home/itframe/apache-rocketmq-masterB
export PATH=$ROCKETMQ_HOME/bin:$PATH
```

设置环境变量注意以下要求：

1. 设置ROCKET_HOME环境变量前，需保证已设置JAVA_HOME环境变量，此变量设置不在此文档中体现
2. 必须设置该环境变量作用范围为当前用户环境变量，且持久化存储，如添加到用户目录下的.profile或.bash_profile文件中。不能直接拷贝上面的命令在命令行执行作为当前shell进程的局部变量，否则当下次重启MQ时，若该操作已不在当前进程，该变量已失效。

ROCKET_HOME环境变量设置如下，仅供参考：

```
#1. 进入宿主目录
cd ~
#2. 编辑.bash_profile文件，在文件最后面添加要配置的环境变量, 注意上文示例的安装包路径
(/home/itframe/apache-rocketmq-masterA) 修改为当前主机安装包的路径，不要直接复制
vi .bash_profile
#3. 使配置生效
source .bash_profile
#4. 查看环境变量是否正确配置，最好重新登录服务器，即不在当前shell进程中查看
echo $ROCKETMQ_HOME
#如果命令行展示的值非空，且路径正确便配置完成了
```

修改配置文件

1. 修改\$ROCKETMQ_HOME/conf/namesrv.properties配置文件

#namesrv监听端口，默认值9876，可根据运维部门的端口配置要求进行更改
listenPort=9876

2. 修改broker配置文件\$ROCKETMQ_HOME/conf/2m-noslave/(broker-*.properties)，注意masterA服务器这改broker-a.properties,MasterB服务器只改broker-b.properties,只对以下指定字段修改，其它配置项不能增加或修改，需要修改的字段如下：

key	默认value	规范说明
brokerClusterName	CLUSTER_DEFAULT	集群名：集群命名规范为CLUSTER_XXX（XXX为大写项目编码缩写）
brokerName	[broker-a/broker-b]	安装包中masterA、masterB对应的brokerName已经配置好，不要改动
namesrvAddr	192.169.100.2:9876;192.169.100.3:9876	namesrv地址及端口，多个地址以英文分号(;)分隔
brokerIP1	192.169.100.2	broker的地址
listenPort	10911	broker监听端口，默认10911，可根据运维部门的端口配置要求进行更改

配置实例：在192.168.100.152主机上部署masterA的broker-a.properties配置如下，broker-b.properties同理

```
#规范要求为CLUSTER_XXX，XXX为业务系统的应用编码或项目工程英文大写
brokerClusterName=CLUSTER_ITFRAME
brokerName=broker-a
#关于这个name sesrver集群地址，broker-a与broker-b的配置必须一模一样，如果name server是多个节点，按照规范配置多个name server地址，如本文示例即可。
#不要出现brokker配置namesrvAddr出现主机一一对应关系，比如broker-a和其中一个name server部署在152主机，而broker-b与另一个name server配置在另一个主机153上，配置namesrvAddr时，broker-a只配置152的name server地址，而broker-b只配置153的name server的地址，这是错误配置。
namesrvAddr=192.169.100.152:9876;192.169.100.153:9876
brokerIP1=192.169.100.152
listenPort=10911
brokerId=0
brokerRole=ASYNC_MASTER
flushDiskType=ASYNC_FLUSH
autoCreateTopicEnable=false
autoCreateSubscriptionGroup=false
```

启动顺序

集群模式为2m-noslave，分别启动masterA、masterB的namesrv、broker，启动顺序为：

启动masterA：先启动name server再启动broker

启动masterB：先启动name server再启动broker

注：**masterA**和**masterB**并没有严格的先后启动顺序，但是**namesrv**和**broker**应当遵守先后启动顺序，**namesrv**须先于**broker**启动

p.s. 如果对于启动顺序还有不明白的地方，请严格按照规范要求启动即可。

集群启动完成后，可通过集群列表查询命令查看集群列表，示例如下：

```
#-n 参数后面跟的是name server服务器的地址及端口，可使用其中任意一个name server的地址即可，如果想用多个地址，注意处理其中的分号，如将分号转义：192.168.100.152:9876\;192.168.100.153:9876
#刚启动集群完成，需要查询集群列表是，优先使用最先启动的name server的地址
#如果搭建MQ服务器的人员自定义脚本同时启动name server时，第一次查询集群列表没有broker信息，建议等待30秒后再查
#如果使用最后启动的name server的地址查询，看不到启动的第一个broker的信息，建议等待30秒后再查
sh mqadmin clusterList -n 192.168.100.152:9876
```

示例截图如下：

```
itframe@XQ6F-B0706-VM10: [/home/itframe/apache-rocketmq-masterA/bin] sh mqadmin clusterList -n 192.168.100.152:9876
#Cluster Name #Broker Name #PID #Addr #Version #InTPS(LOAD) #OutTPS(LOAD) #PCWait(ms) #Hour #SPACE
CLUSTER_ITFRAME broker-a 0 192.168.100.152:10911 V4_1_0 SNAPSHOT 0.00(0,0ms) 0.00(0,0ms) 0 424402.07 0.2309
CLUSTER_ITFRAME broker-b 0 192.168.100.153:10911 V4_1_0 SNAPSHOT 0.00(0,0ms) 0.00(0,0ms) 0 424402.07 0.2587
```

启动脚本

启动脚本在目录 `$ROCKETMQ_HOME/bin` 下

1. 启动namesrv

```
#启动mqnamesrv
nohup sh mqnamesrv -c ../conf/mqnamesrv.properties &
```

namesrv启动后需要查看启动日志，待namesrv启动成功后方可启动broker，查看日志命令如下：

```
tail -f nohup.out
```

启动name server及查看日志的命令操作截图如下：

```
itframe@XQ6F-B0706-VM10: [/home/itframe/apache-rocketmq-masterA/bin] nohup sh mqnamesrv -c ../conf/mqnamesrv.properties &
[1] 16298
itframe@XQ6F-B0706-VM10: [/home/itframe/apache-rocketmq-masterA/bin] nohup: 忽略输入并把输出追加到 "nohup.out"

itframe@XQ6F-B0706-VM10: [/home/itframe/apache-rocketmq-masterA/bin] tail -f nohup.out
load config properties file OK, ../conf/mqnamesrv.properties
The Name Server boot success. serializeType=JSON
█
```

如果显示图片中最后两行日志，即说明已正确启动，有可能nohup.out的日志内容比较多，tail -f命令默认展示最新的10行，所以其它的不用管，只看截图中指定的日志即可。

下文关于nohup.out的日志查看与此相同，只查看截图中指定的日志，打印的多余的无关日志不用关心。

2. 启动broker

启动broker的脚本命令如下：

```
nohup sh mqbroker -c ../conf/2m-noslave/broker-a.properties &  
或  
nohup sh mqbroker -c ../conf/2m-noslave/broker-b.properties &
```

查看启动日志的命令如下：

```
tail -f nohup.out
```

启动broker及查看日志的命令操作截图如下：

```
itframe@XQ6F-B0706-VM10: [/home/itframe/apache-rocketmq-masterA/bin] nohup sh mqbroker -c ../conf/2m-noslave/broker-a.properties &  
[2] 23234  
itframe@XQ6F-B0706-VM10: [/home/itframe/apache-rocketmq-masterA/bin] nohup: 忽略输入并把输出追加到 "nohup.out"  
itframe@XQ6F-B0706-VM10: [/home/itframe/apache-rocketmq-masterA/bin] tail -f nohup.out
```

在该版本，broker正确启动在nohup.out中应该是无法看到成功的日志，该日志被打印到日志文件中，所以无法重定向到指定输出流，但是如果相关配置或启动要求不合法，依然写到标准输出流，被重定向到当前的nohup.out中，如果启动失败，可以查看nohup.out定位问题。

如果想要查看启动成功的日志，可以查看用户目录下 `logs/rocketmqlogs/broker.log`，示例截图如下：

```
2018-06-04 15:19:58 INFO main - register broker to name server 192.168.100.152:9876 OK  
2018-06-04 15:19:58 INFO main - register broker to name server 192.168.100.153:9876 OK  
2018-06-04 15:19:58 INFO main - The broker[broker-a, 192.168.100.152:10911] boot success. serializeType=JSON and name server is 192.168.100.152:9876;192.168.100.153:9876
```

集群启动完成后，按照启动顺序里说明的查看集群列表的操作查看集群情况。

注：**broker**启动后，如果通过**netstat**查看的话，会发现**broker**除监听端口**10911**会额外出现**10912**、**10913**(在监听端口进行加1加2)，这两个端口不用关系，这是**broker**内部通信端口，属于正常端口，不用担心。

停止脚本

停止脚本在目录 `$ROCKETMQ_HOME/bin` 下

分别停止masterA、masterB下的broker、namesrv

```
sh mqshutdown broker  
sh mqshutdown namesrv
```

注意先停broker后停namesrv，重启时间间隔时间至少1分钟，否则可能出现端口冲突异常。

创建Topic

脚本在目录 `$ROCKETMQ_HOME/bin` 下，注意：只需在任意一台**MQ**服务器执行即可，不能两台都执行。

1. 创建普通消息Topic

```
sh mqadmin updateTopic -c CLUSTER_ITFRAME -n 192.168.100.152:9876 -t CUSTOMER_SMSNOTIC -r 10 -w 10
```

操作示例截图如下：

```
itframe@XQ6F-B0706-VM10: [/home/itframe/apache-rocketmq-masterA/bin] sh mqadmin updateTopic -c CLUSTER_ITFRAME -n 192.168.100.152:9876 -t CUSTOMER_SMSNOTIC -r 10 -w 10
create topic to 192.168.100.152:10911 success.
create topic to 192.168.100.153:10911 success.
TopicConfig [topicName=CUSTOMER_SMSNOTIC, readQueueNums=10, writeQueueNums=10, perm=RW-, topicFilterType=SINGLE_TAG, topicSysFlag=0, order=false]
itframe@XQ6F-B0706-VM10: [/home/itframe/apache-rocketmq-masterA/bin]
```

p.s. 该命令使用-c参数指定了集群名称，创建或更新topic的时候，会在该集群上注册的所有broker都创建或更新一份topic信息，所以该命令只用执行一次即可。

2. 创建顺序消息Topic

```
sh mqadmin updateTopic -c CLUSTER_ITFRAME -n 192.168.100.152:9876 -t CUSTOMER_SMSNOTIC -r 10 -w 10 -o true
```

操作示例截图如下：

```
<terA/bin] sh mqadmin updateTopic -c CLUSTER_ITFRAME -n 192.168.100.152:9876 -t CUSTOMER_SMSNOTIC -r 10 -w 10 -o true
create topic to 192.168.100.152:10911 success.
create topic to 192.168.100.153:10911 success.
set cluster orderConf. isOrder=true, orderConf=[broker-a:10;broker-b:10]
TopicConfig [topicName=CUSTOMER_SMSNOTIC, readQueueNums=10, writeQueueNums=10, perm=RW-, topicFilterType=SINGLE_TAG, topicSysFlag=0, order=true]
itframe@XQ6F-B0706-VM10: [/home/itframe/apache-rocketmq-masterA/bin]
```

p.s. 该命令使用-c参数指定了集群名称，创建或更新topic的时候，会在该集群上注册的所有broker都创建或更新一份topic信息，所以该命令只用执行一次即可。

注意以下几点要求：

1. Topic命名时需字母需大写,以业务功能命名，多单词以英文下划线（_）连接
2. 按照以上规范要求创建Topic,若读写队列需做调整请与技术组确认
3. 若该集群供多个业务系统使用，创建Topic时请以项目编码作为topic的前缀如：CUSTOMER_SMSNOTIC

创建消费组

脚本在目录 `$ROCKETMQ_HOME/bin` 下，注意：只需在任意一台MQ服务器执行即可，不能两台都执行。

1. 普通消息消费组

```
sh mqadmin updateSubGroup -g CONSUMER_CUSTOMER_SMSNOTIC_GROUP -r 1 -n 192.168.100.152:9876 -c CLUSTER_ITFRAME
```

操作示例截图如下：

```
<terA/bin] sh mqadmin updateSubGroup -g CONSUMER_CUSTOMER_SMSNOTIC_GROUP -r 1 -n 192.168.100.152:9876 -c CLUSTER_ITFRAME
create subscription group to 192.168.100.152:10911 success.
create subscription group to 192.168.100.153:10911 success.
SubscriptionGroupConfig [groupName=CONSUMER_CUSTOMER_SMSNOTIC_GROUP, consumeEnable=true, consumeFromMinEnable=false, consumeBroadcastEnable=false, retryQueueNums=1, retryMaxTimes=1, brokerId=0, whichBrokerWhenConsumeSlowly=1, notifyConsumerIdsChangedEnable=true]
itframe@XQ6F-B0706-VM10: [/home/itframe/apache-rocketmq-masterA/bin]
```

p.s. 该命令使用-c参数指定了集群名称，创建或更新订阅组的时候，会在该集群上注册的所有broker都创建或更新一份订阅组信息，所以该命令只用执行一次即可。

2. 广播消息消费组

```
sh mqadmin updateSubGroup -g CONSUMER_CUSTOMER_SMSNOTIC_GROUP -r 1 -d true -n 192.168.100.152:9876 -c CLUSTER_ITFRAME
```

操作示例截图如下：

```
terA/bin]sh mqadmin updateSubGroup -g CONSUMER_CUSTOMER_SMSNOTIC_GROUP -r 1 -d true -n 192.168.100.152:9876 -c CLUSTER_ITFRAME
create subscription group to 192.168.100.152:10911 success.
create subscription group to 192.168.100.153:10911 success.
SubscriptionGroupConfig [groupName=CONSUMER_CUSTOMER_SMSNOTIC_GROUP, consumeEnable=true, consumeFromMinEnable=false, consumeBroadcastEnable=true, retryQueueNums=1, retryMaxTimes=1, brokerId=0, whichBrokerWhenConsumeSlowly=1, notifyConsumerIdsChangedEnable=true]itframe@XQ6F-B0706-VM10:[/home/itframe/apache-rocketmq-masterA/bin]
```

p.s. 该命令使用-c参数指定了集群名称，创建或更新订阅组的时候，会在该集群上注册的所有broker都创建或更新一份订阅组信息，所以该命令只用执行一次即可。

注意以下几点要求：

1. 消费组命名规则，CONSUMER_XXX_GROUP，"XXX"为Topic名称(上节中TOPIC名称)，大写
2. 按照以上规范要求创建消费组，若相关参数需做调整请与技术组确认

MQ服务器数据迁移

注意：重新部署的MQ集群不涉及此项操作，请忽略

1. copy用户目录下store目录及其下所有子目录或文件信息到目标服务器用户目录下
2. 如果用户目录下有namesrv目录，copy该目录及其下所有文件到目标服务器用户目录下
3. 如果当前待迁移集群broker的配置文件中配置了store存储路径不是默认的用户目录下，则将配置路径下的store目录及其下所有子目录或文件信息复制到目标服务器下，路径说明：如果目标服务器也配置了路径，则为配置的路径，未配置则将store目录下的文件copy到用户目录的store目录下对应的位置。

注意：在数据复制过程中，即使store/commitlog目录下文件过大，也要保证全量复制