

目录

目录	1
如何设置镜像队列	2
镜像队列简介	2
镜像对列设置	2
具体设置步骤	2
网络分区cluster_partition_handling	4
网络分区	4

如何设置镜像队列

镜像队列简介

RabbitMQ的集群创建后，默认消息只存在于队列所在节点，当此节点故障后，消息就丢失了。为了解决上述消息丢失的问题，引入RabbitMQ的镜像队列机制，将queue镜像到cluster中其他的节点之上。在该实现下，如果集群中的一个节点失效了，queue能自动地切换到镜像中的另一个节点以保证服务的可用性。

镜像对列设置

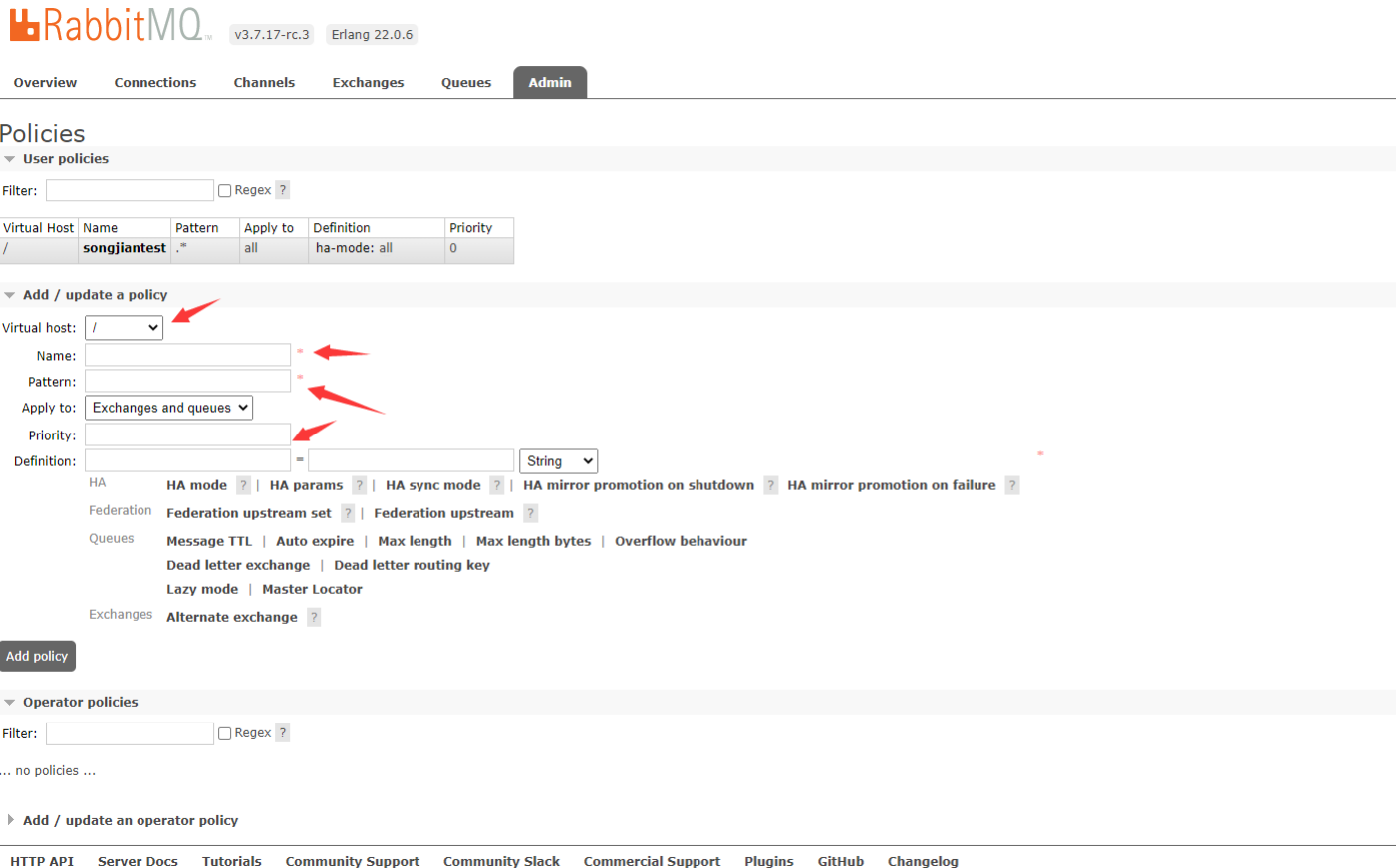
镜像队列，在单节点宕机时，提供高可用服务，保证数据不丢失。镜像队列是在购买服务后，由用户配置同步策略来实现的：

```
rabbitmqctl set_policy [-p Vhost] Name Pattern Definition [Priority]
-p Vhost: 可选参数，针对指定vhost下的queue进行设置
Name: policy的名称
Pattern: queue的匹配模式(正则表达式)
Definition: 镜像定义，包括三部分ha-mode, ha-params, ha-sync-mode
ha-mode:指明镜像队列的模式，有效值为 all/exactly/nodes
all: 表示在集群中所有的节点上进行镜像
exactly: 表示在指定个数的节点上进行镜像，节点的个数由ha-params指定
nodes: 表示在指定的节点上进行镜像，节点名称通过ha-params指定
ha-params: 作为参数，为ha-mode的补充
ha-sync-mode: 进行队列中消息的同步方式，有效值为automatic和manual
priority: 可选参数，policy的优先级

rabbitmqctl set_policy ha-all "" '{"ha-mode":"all"}' --apply-to all
```

具体设置步骤

1. 打开web ui管理页面：



2. 如对vhost /，中所有的队列设置则按以下内容填写，并点击Add policy:

Policies

▼ User policies

Filter: Regex ?

... no policies ...

▼ Add / update a policy

Virtual host: Name: Pattern: Apply to: Priority: Definition: = HA [HA mode ?](#) | [HA params ?](#) | [HA sync mode ?](#) | [HA mirror promotion on shutdown ?](#) | [HA mirror promotion on failure ?](#)Federation [Federation upstream set ?](#) | [Federation upstream ?](#)Queues [Message TTL](#) | [Auto expire](#) | [Max length](#) | [Max length bytes](#) | [Overflow behaviour](#)[Dead letter exchange](#) | [Dead letter routing key](#)[Lazy mode](#) | [Master Locator](#)Exchanges [Alternate exchange ?](#)**Add policy**

▼ Operator policies

Filter: Regex ?

... no policies ...

▶ Add / update an operator policy



3. 如果queue中有存量未消费的消息时，则需要手动触发同步才可进行数据的复制（同步期间会引起阻塞），红色箭头指向部分代表需要手动同步存量消息，如果没有则说明不需要单独同步存量消息。（如果policy设置后，消费者消费完了存量消息，则会自动变成正常状态，此时就无需手动同步了）

Queues

▼ All queues (3)

Pagination

Page of 1 - Filter: Regex ?

Overview				Messages			Message rates			+/-	
Virtual host	Name	Node	Features	State	Ready	Unacked	Total	incoming	deliver	get	ack
/	songjiantest	rabbit@vm10-72-48-67	+0   D songjiantest	idle	11	0	11	0.00/s	0.00/s	0.00/s	
/	songjiantest2	rabbit@vm10-72-48-67	+2 D songjiantest	idle	0	0	0				
/	songjiantest3	rabbit@vm10-72-48-67	+2 D songjiantest	idle	0	0	0				

▼ Add a new queue

Virtual host: Name: Durability: Node: Auto delete: Arguments: = Add [Message TTL ?](#) | [Auto expire ?](#) | [Max length ?](#) | [Max length bytes ?](#) | [Overflow behaviour ?](#)[Dead letter exchange ?](#) | [Dead letter routing key ?](#) | [Maximum priority ?](#)[Lazy mode ?](#) | [Master locator ?](#)**Add queue**

4. 点击队名列名，进入队列详情界面，可以看到如下信息，点击Synchronise则可发起同步

Message rates last minute ?

1.0/s
0.0/s
18:56:00 18:56:10 18:56:20 18:56:30 18:56:40 18:56:50

Publish 0.00/s
Deliver (manual ack) 0.00/s
Deliver (auto ack) 0.00/s

Consumer ack 0.00/s
Redelivered 0.00/s
Get (manual ack) 0.00/s
Get (auto ack) 0.00/s

Details

Features durable: true
Policy songjiantest
Operator policy
Effective policy definition ha-mode: all
Node rabbit@vm10-72-48-67
Mirrors rabbit@vm10-72-60-106 (unsynchronised)
rabbit@vm10-72-48-66 (unsynchronised)
Synchronise

State idle
Consumers 0
Consumer utilisation 0%

Messages 11
Message body bytes 221iB
Process memory 22kiB

Total 11
Ready 11
Unacked 0
In memory 11
Persistent 0
Transi 0

The shovel plugin can be used to move messages from this queue to another one. The form below will create a temporary shovel to move messages to another queue on the same virtual host, with default settings.

5. 当同步完成后，进入queues的界面，需要同步的队列的显示效果与其他队列相同

RabbitMQ v3.7.17-rc.3 Erlang 22.0.6

Overview Connections Channels Exchanges Queues Admin

Queues

All queues (3)

Overview

Virtual host	Name	Node	Features	State	Ready	Unacked	Total	Message rates
								incoming deliver / get ack
/	songjiantest	rabbit@vm10-72-48-67	+2 D songjiantest	idle	11	0	11	0.00/s 0.00/s 0.00/s
/	songjiantest2	rabbit@vm10-72-48-67	+2 D songjiantest	idle	0	0	0	
/	songjiantest3	rabbit@vm10-72-48-67	+2 D songjiantest	idle	0	0	0	

Add a new queue

Virtual host: /
Name:
Durability: Durable

更多参考请见[高可用队列设置](#)

网络分区cluster_partition_handling

网络分区

RabbitMQ提供了四种种处理网络分区的方式，即rabbitmq.config中的cluster_partition_handling参数配置，分别为：

- **ignore**: ignore的配置是当网络分区的时候，RabbitMQ不会自动做任何处理，即需要手动处理。该模式保证数据完整性，牺牲一致性以及部分可用性。
- **pause_minority**: 当发生网络分区时，集群中的节点在观察到某些节点down掉时，会自动检测其自身是否处于少数派（小于或者等于集群中一般的节点数）。少数派中的节点在分区发生时会自动关闭，当分区结束时又会启动。如果未设置镜像队列，发生分区时少数派上的队列将无法生产消费。此模式折中了可用性，选择了数据一致性。
- **autoheal**: 在autoheal模式下，当认为发生网络分区时，RabbitMQ会自动决定一个获胜的（winning）分区，然后重启不在这个分区中的节点以恢复网络分区。该模式在更看重服务连续性而不是跨节点数据一致性时使用。
- **pause_if_all_down**: 在pause_if_all_down模式下，RabbitMQ会自动关闭不能和list中节点通信的节点。语法为[pause_if_all_down, [nodes], ignore|autoheal]，其中[nodes]即为前面所说的list。

目前金山云采用cluster_partition_handling=ignore模式，后台系统监控告警分区情况，由金山侧运维处理。客户无法自行修改，客户可联系对应商务，提工单评估修改分区策略配置。

详细信息请参考[网络分区策略设置](#)