SOFAStack

消息队列 SOFAMQ
运维指南

产品版本：AntStack Plus 1.11.0

文档版本：20230214

蚂蚁集团
ANT GROUP

法律声明

蚂蚁集团版权所有©2022，并保留一切权利。

未经蚂蚁集团事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

商标声明

蚂蚁集团 ANT GROUP及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

免责声明

由于产品版本升级、调整或其他原因，本文档内容有可能变更。蚂蚁集团保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在蚂蚁集团授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
`Courier字体`	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

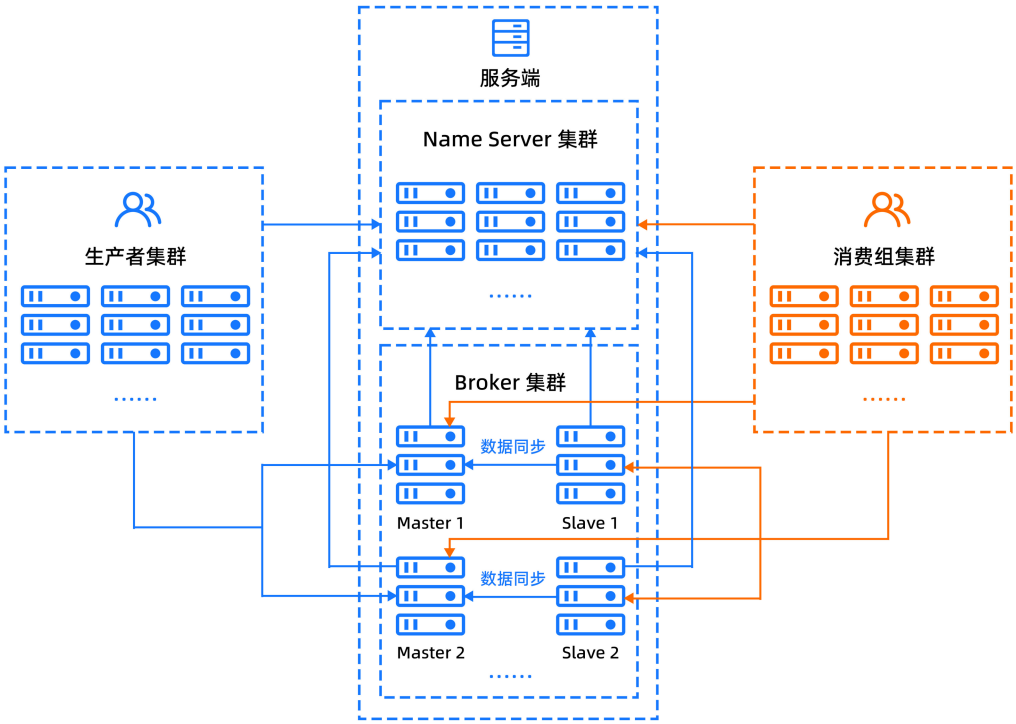
目录

1.系统组件 06
2.监控和预警 08
2.1. 监控指标和预警项 08
2.2. 预警项运维动作 08
3.系统日志 10
4.同城容灾 11
4.1. 产品架构 11
4.2. 容灾架构 12
4.2.1. 架构图 12
4.2.2. 产品依赖 14
4.2.3. 数据同步 14
4.2.4. 网络链路 14
4.2.5. 容灾能力 14
4.3. 热切换方案 15
4.4. 容灾场景处理 16
4.5. Region 收敛 16
5.日常运维 17
5.1. 告警处理 17
5.1.1. 容器异常告警 17
5.1.2. CPU 使用率异常告警 19
5.1.3. 内存使用率异常告警 19
5.1.4. 磁盘使用率异常告警 20
5.1.5. JVM 异常告警 20
5.2. 组件可用性运维 21
5.2.1. Broker 发消息异常 21
5.2.2. sofamq-namesrv 节点异常 22

5.2.3. sofamq-broker 节点异常 23
5.2.4. sofamq-console 节点异常 24
5.3. 服务巡检 24
5.4. 管理容器 26
5.4.1. 变更容器配置 26
5.4.2. 容器内存与 JVM 堆内存配置比 28
5.4.3. 重启容器 29
6.常见故障处理 31
6.1. 服务可用性故障 31
6.1.1. 服务可用性故障 31
6.1.2. Broker 运行状态故障 31
6.2. 业务连续性故障 32
7.常见问题 35
7.1. 路由表查询失败 35
7.2. 消息积压排查 35
8.管理控制台 37
8.1. 资源管理 37
8.1.1. 节点管理 37
8.2. 元数据管理 38
8.2.1. Topic 管理 38
8.2.2. Group 管理 39
8.3. 运维工具 40
8.3.1. 配置管理 40
8.3.2. KV 管理 42
8.4. 权限管理 44
8.4.1. 管理员列表 44

1.系统组件

组件角色说明

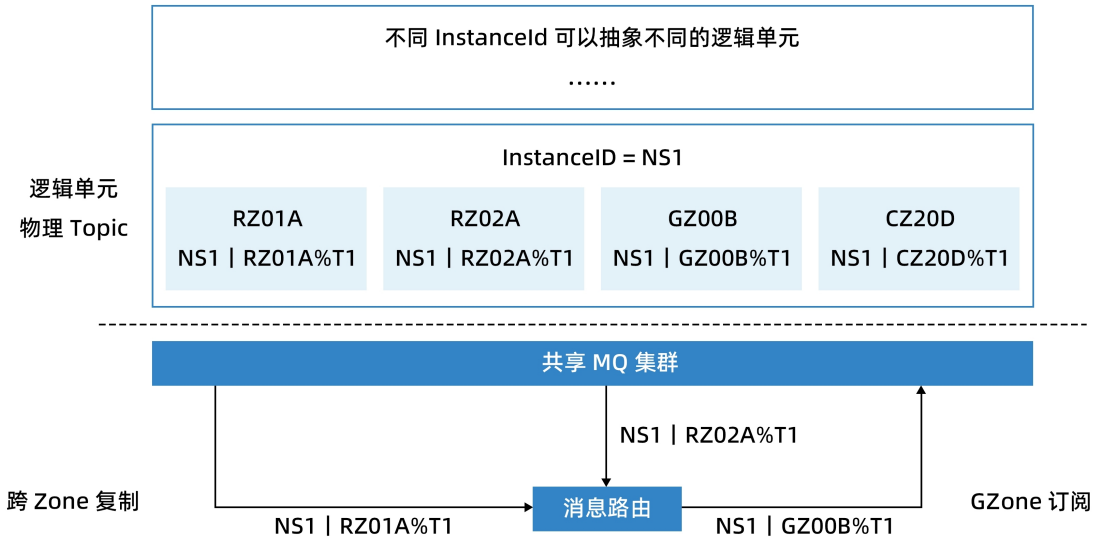


SOFAMQ 消息队列包括以下 4 个组件：

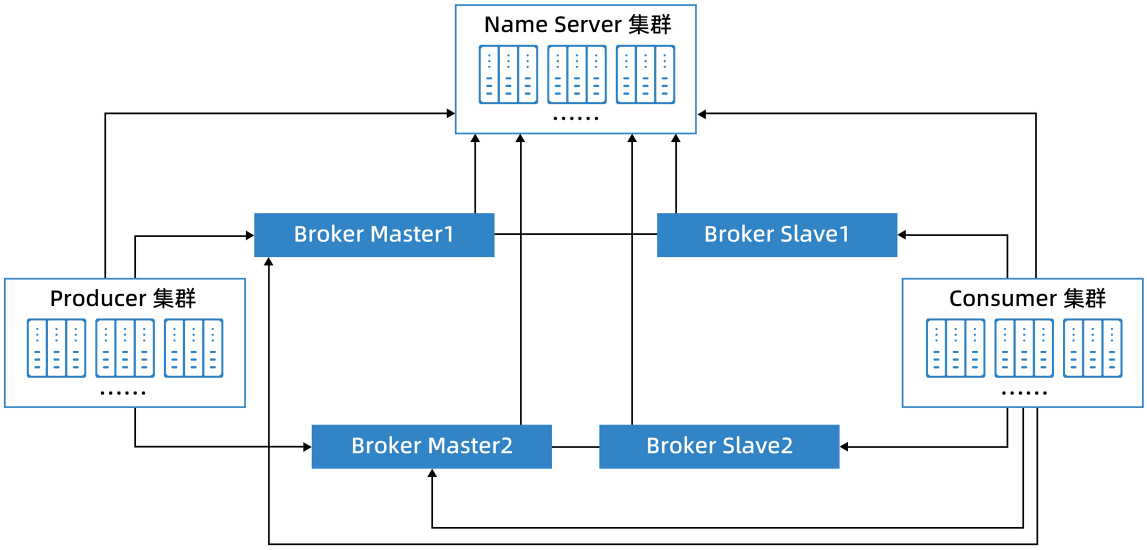
- **NameServer**：是一个几乎无状态节点，可集群部署，在消息队列中提供命名服务，该组件内置 zookeeper 用来负责 Broker 的选主和集群管理。
- **Broker**：消息中转角色，负责存储和转发消息。分为 Master Broker 和 Slave Broker，一个 Master Broker 可以对应多个 Slave Broker，但是一个 Slave Broker 只能对应一个 Master Broker。Broker 启动后需要完成一次将自己注册至 Name Server 的操作；随后每隔 30s 定期向 Name Server 上报 Topic 路由信息。
- **生产者 Producer**：与 Name Server 集群中的其中一个节点（随机）建立长连接（Keep-alive），定期从 Name Server 读取 Topic 路由信息，并与提供 Topic 服务的 Master Broker 建立长连接，且定时向 Master Broker 发送心跳。
- **消费者 Consumer**：与 Name Server 集群中的其中一个节点（随机）建立长连接，定期从 Name Server 拉取 Topic 路由信息，并向提供 Topic 服务的 Master Broker、Slave Broker 建立长连接，且定时向 Master Broker、Slave Broker 发送心跳。Consumer 既可以从 Master Broker 订阅消息，也可以从 Slave Broker 订阅消息，订阅规则由 Broker 配置决定。

SOFAMQ 消息队列还包含如下两个未在图中展示的组件，主要功能如下：

- **Console**：SOFAMQ 的图形化管理控制台。
- **Router**：LDC 场景下，根据配置的 Router 规则，进行逻辑单元之间消息转发，使 SOFAMQ 支持 LDC 能力的组件。



部署拓扑说明



资源清单说明

组件角色	资源依赖	服务端口
Name Server	2C4G	9876、2181
Broker	4C8G	10911、10909、9000
Console	2C4G	80、8080、8341

2.监控和预警

2.1. 监控指标和预警项

系统监控指标和预警项清单

巡检对象	描述	检测端口
NameServer	检查 NameServer 的 TCP 端口是否正常。	9876
Broker	检查 Broker 的 TCP 端口是否正常。	10911
Console	检查 Console 的 TCP 端口是否正常。	80

自定义监控指标和预警项清单

监控项	监控项说明	预警触发条件
发布者消息发布数量	统计消息队列接收到的发布者应用发布的消息数量。	-
订阅者消息投递数量	统计消息队列投递到订阅者应用的消息数量。	-
消息队列积压消息数量	统计消息队列消费积压消息数量，一般是消费下游处理消息慢或异常。 预警订阅角色：运维管理员。	每分钟超过 1 条。
消息队列接收失败消息数量	统计消息队接收的失败消息数量，一般是消息队列存储资源出现不可用异常或流量超过安全水位触发限流。 预警订阅角色：运维管理员。	每分钟超过 1 条。

2.2. 预警项运维动作

预警项	动作说明
TCP 端口访问异常	登录机器检查进程是否存在，如果进程存在，检查网络是否正常；不存在则重启节点。

预警项	动作说明
消息队列积压消息数量	检查对应消费节点正常或者处理消息是否过慢，尝试扩容消费者节点或优化消息处理性能。
消息队列接收失败消息数量	检查处理消息的业务逻辑或其他因素，保证能正确处理消息。

3.系统日志

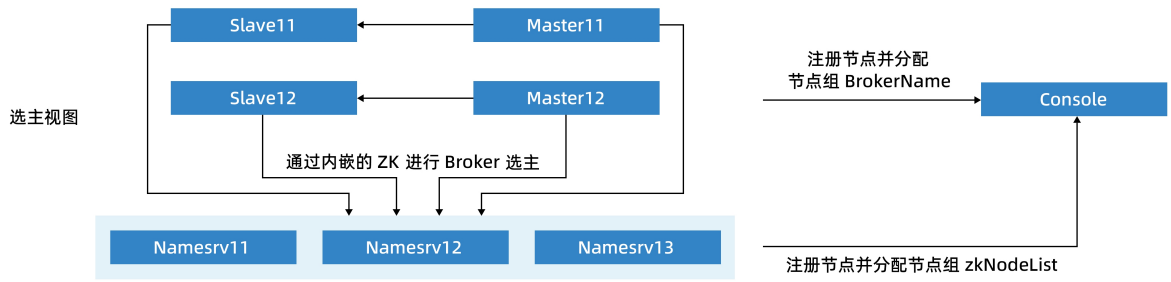
日志文件清单说明

日志文件名称	日志文件路径	日志文件描述
broker.log	/home/admin/logs/rocketmqlogs/broker.log	可以通过日志下的 register 信息确认 Broker 的注册状态，`Register broker to name server {ip} OK` 表示该 Broker 已在 nameserver 上注册，即 clusterlist 可查到该 Broker。
remoting.log	/home/admin/logs/rocketmqlogs/remoting.log	可以通过日志下的客户端 IP 确定网络断开的时间。
failover_controller.log	/home/admin/logs/rocketmqlogs/failover_controller.log	主备切换，关注状态机切换的时间点。
stats.log	/home/admin/logs/rocketmqlogs/stats.log	记录当前 Broker 的 TPS，用于查阅历史的 TPS 问题。资源报表，也是由这里的日志汇总得到。关键字说明如下： • `BROKER_PUT_NUMS` 每分钟写消息 TPS 流量监控。 • `BROKER_GET_NUMS` 每分钟读消息 TPS 流量监控。 可用于某个时间点发送消息失败/写消息失败的排查。
namesrv.log	/home/admin/logs/rocketmqlogs/namesrv.log	记录当前 namesrv 连接层相关信息。
console	/home/admin/logs/mqconsole	• `api-cost.log` ：接口调用耗时统计日志。 • `common-default.log` ：控制台日志（默认日志）。 • `common-error.log` ：错误日志。 • `oms.log` ：计费上报日志。 • `sla.log` ：异常上报日志。 • `zk_config.log` ：ZK 连接日志。

4.同城容灾

4.1. 产品架构

基础部署架构



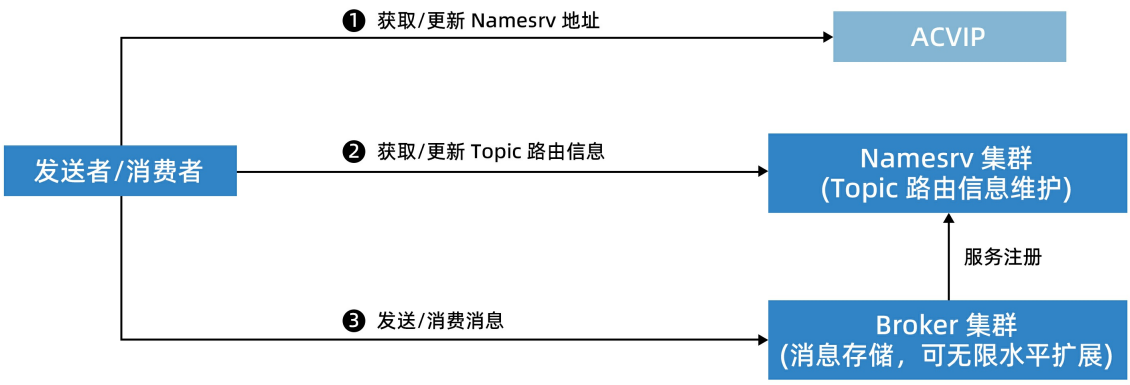
架构说明：

- **Console**：负责节点的注册和节点组的分配。
- **Broker**：负责消息存储，通过主备复制来达到单机容灾的数据高可靠。当主节点宕机时，NameServer 中内置的 Zookeeper 会根据主备复制的状态，将对应的备节点切换成主节点继续提供读写服务。
- **NameServer**：负责 Broker 的选主和 Topic 路由信息维护，三个节点组成 ZK 集群。

部署模式容灾表：

	【主】进程 Crash	【主】单机掉电	【主】磁盘损坏
单机房 HA ASYNC_FLUSH（默认）	RTO=0、RPO=0	RTO=0、RPO=0	RTO=0、RPO=0
单机房 HA SYNC_FLUSH	RTO=0、RPO=0	RTO=0、RPO=0	RTO=0、RPO=0

收发消息流程说明：



生产者或消费者收发消息的时候首先通过 ACVIP 获取 NameServer 的地址，然后从 NameServer 获取 Topic 的路由信息，最后选择路由信息中的 Broker 发送或消费消息。

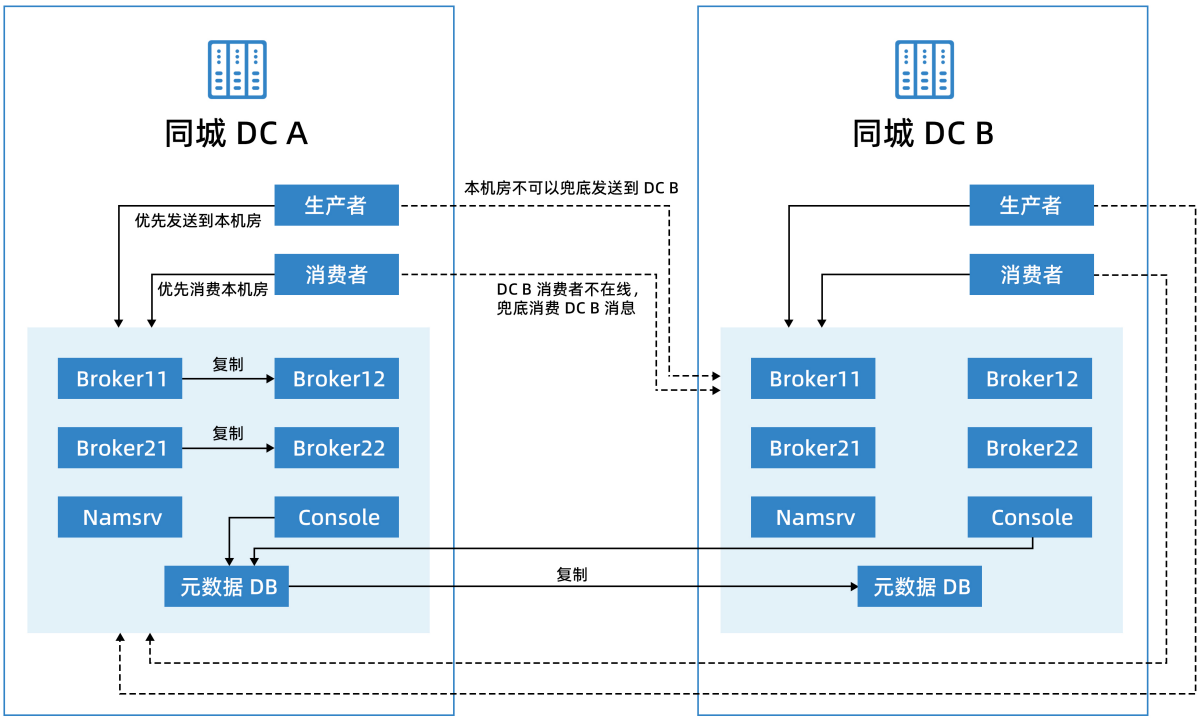
> **重要**
>
> 业务应用通过 ACVIP 查找 NameServer 地址，这是中间件通用逻辑，在上述架构图中省略。

4.2. 容灾架构

4.2.1. 架构图

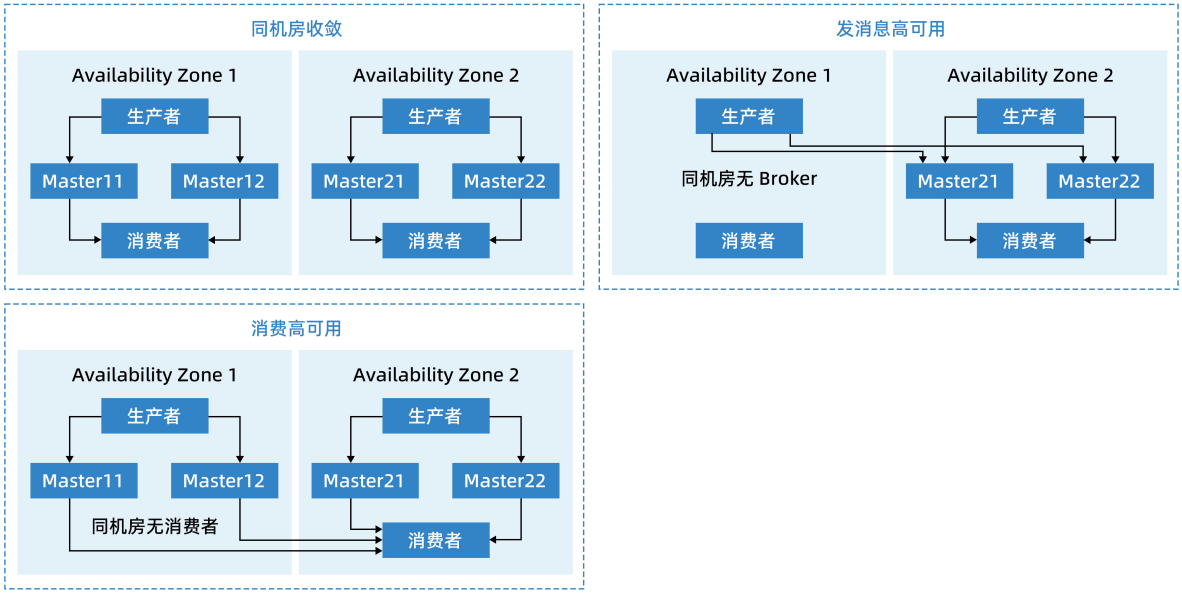
SOFAMQ 容灾架构有两种：机房收敛和本地优先。本节介绍本地优先的方案，这也是通常输出的方案。

Broker 节点向同城所有的 Namesrv 节点进行注册，并且生产者和消费者优先收发本机房的消息。



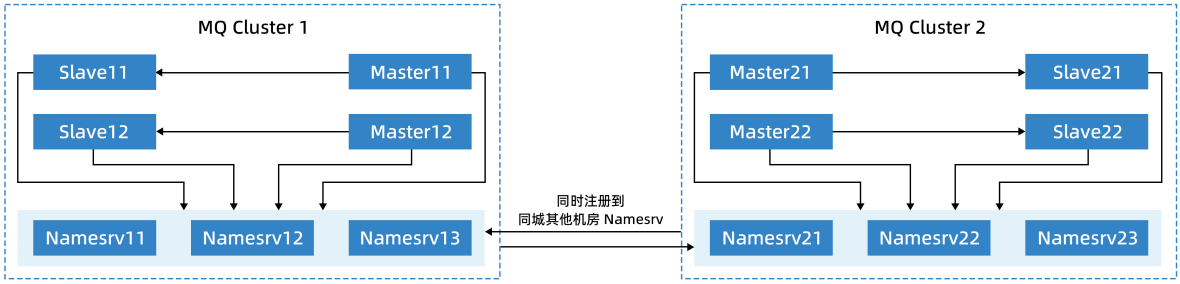
- **用户视图**

  生产者和消费者优先收发本机房的消息，在单机房 Broker 或者消费端不可用的时候，可以保障发送和消费高可用。



- **集群视图**

  Broker 同时注册到两个机房的 Namesrv，在单机房 Broker 或者消费端不可用的时候，可以保障发送和消费高可用。



容灾架构说明：

- 管控台在双机房部署，使用数据库作为元数据存储，管控页面统一通过 Intelliproxy 进行路由，容灾规则与 Intelliproxy 保持一致。
- 客户端通过自带的 DATACENTER 属性来完成正常情况单机房流量收敛。

容灾切换说明：

- **场景一：机房断电导致机房整体不可用或者网络孤岛导致机房对外网络入口关闭。**

  由于流量默认是单机房收敛的，且消息队列的主备是双机房部署的，如果机房断电，另一个机房仍是可用的。业务应用需切走故障机房的整个流量，将入口流量切换到正常提供服务的机房。

- **场景二：单机房的 Broker 全部不可用。**

  发送端检测到同机房不存在可用的 Broker时，会发送到另一个机房的 Broker，保障发送高可用。

- **场景三：单机房的消费者全部不可用。**

  消费端检测到同城其他机房不存在消费者时，会消费同城其他机房的数据，保障消费高可用。

- **场景四：单机房的 Namesrv 全部不可用。**

  Namesrv 至少有三个副本，且互斥部署。本地优先模式下，SOFAMQ Client 检测到同机房不存在可用的 Namesrv 时，会连接到另一个机房的 Namesrv，保障发送高可用；机房收敛模式下，如果 Namesrv 全部不可用相当于单机房宕机，处理方式同 **场景一：机房整体不可用**。

> 说明
>
> SOFAMQ Client 会优先连接本机房的 Namesrv，会过滤掉 ACVIP 中非本 datacenter 的链接地址。

- **场景五：元数据 DB 故障或者发生主备切换。**

  由于 DB 故障或者主备切换导致不可用时，SOFAMQ Console 无法使用，但是 Broker 可以正常使用，所以消息可以正常收发。这种情况需要尽快恢复 DB。

4.2.2. 产品依赖

外部产品依赖

- **ACVIP**：用于获取 Namesrv 地址。
- **SOFARegistry**：Console 将自身的地址注册到 SOFARegistry 上，供 Intelliproxy 获取调用。
- **Intelliproxy**、**OSP**：中间件控制台统一管理页面访问入口。

底座产品依赖

- **元集群**：元集群主备机房独立部署，无容灾要求。
- **内网 DNS**：
  - SOFAMQ 通过内网 DNS 提供的域名访问数据库，容灾切换时，需将数据库域名切换至备机房数据库域名。需依赖内网 DNS 的容灾能力。

    > 说明
    >
    > 公有云环境可直接使用数据库资源对外暴露的域名。

  - SOFAMQ 管控页面通过内网 DNS 提供的域名，挂载主备机房的内网负载均衡，并对外提供服务。
- **内网负载均衡**：主备机房各自通过内网负载均衡进行负载操作，并提供给业务应用进行访问。

4.2.3. 数据同步

- **元数据 DB**：主备机房共用元数据库 DB，数据同步能力依赖元数据库的容灾能力。
- **消息**：主备机房双写磁盘，只有都写成功后消息同步才算成功，以此保证机房宕机后数据不丢失。

4.2.4. 网络链路

- 业务应用通过 ACVIP 查询 SOFAMQ 服务地址，点对点通讯。
- 用户通过 Intelliproxy 访问 SOFAMQ 控制台。

4.2.5. 容灾能力

	【主】进程 Crash	【主】单机掉电	【主】磁盘损坏	机房掉电	机房断网*	机房不可恢复灾难*	机房消费者全宕机	机房 Broker 服务全宕机
单机房 HA ASYNC_FLUSH（默认）	RTO = 0、RPO = 0	RTO = 0、RPO = 0	RTO = 0、RPO = 0	RTO = 0、RPO = 刷盘间隔，默认是 500ms，有 UPS 的机房优雅关机的情况下可以做到 RPO = 0	RTO = 0、RPO = 0	RTO = 0、RPO = 积压未消费的数据	同城其他机房消费者进行消费，不会积压。	发送到其他机房。
单机房 HA SYNC_FLUSH	RTO = 0、RPO = 0	RTO = 0、RPO = 0	RTO = 0、RPO = 0	RTO = 0、RPO = 0	RTO = 0、RPO = 0	RTO = 0、RPO = 积压未消费的数据	同城其他机房消费者进行消费，不会积压。	发送到其他机房。

重要

- **机房掉电**：可能存在未被消费的消息，在机房恢复后，积压消息会被重新消费，此时由业务判断是否需要丢弃这些消息。
- **机房不可恢复灾难**：由于消息不会在主备机房同步，所以此时会丢失积压未消费的数据。

4.3. 热切换方案

切换逻辑

1. 切换元数据库 DB 和内网 DNS。
2. 切换 SOFAMQ 管控域名，摘除故障机房的流量。
3. 切换完毕后，验证故障机房 SOFAMQ 状态。

切换影响

切换数据库时，控制台可能存在短暂不可用。

切换原则

数据不丢失，对业务无影响。

4.4. 容灾场景处理

以下场景为真实故障，不涉及切换 SOFAMQ 内部服务。该场景下，只和外部依赖（元数据库和 OSP）相关。

常见容灾场景及处理方式如下：

容灾场景	容灾切换	容灾恢复	容灾回切
主机房断电	1. 切换元数据库 DB。 2. 切换 SOFAMQ 管控域名，摘除主机房流量。 3. 切换完毕后，验证备机房 SOFAMQ 状态。	确定 ACVIP、SoFaRegistry 恢复后，启动 SOFAMQ。	1. 切换元数据库 DB。 2. 切换 SOFAMQ 管控域名，挂载主机房的流量。 3. 切换完毕后，验证主机房 SOFAMQ 状态。
主机房网络孤岛	1. 切换元数据库 DB。 2. 切换 SOFAMQ 管控域名，摘除主机房流量。 3. 切换完毕后，验证备机房 SOFAMQ 状态。 说明 此时应无积压消息。	无需人工恢复。	1. 切换元数据库 DB。 2. 切换 SOFAMQ 管控域名，挂载主机房的流量。 3. 切换完毕后，验证主机房 SOFAMQ 状态。
备机房断电、备机房网络孤岛、脑裂	1. 切换元数据库 DB。 2. 切换 SOFAMQ 管控域名，摘除备机房流量。 3. 切换完毕后，验证主机房 SOFAMQ 状态。	确定 ACVIP、SoFaRegistry 恢复后，启动 SOFAMQ。	1. 切换元数据库 DB。 2. 切换 SOFAMQ 管控域名，挂载备机房的流量。 3. 切换完毕后，验证备机房 SOFAMQ 状态。

说明

网络恢复后会自动重连，在网络中断期间，Dsrconsole、ACVIP、OSP 及其他和管控台有关联的组件均无法使用。

4.5. Region 收敛

混合云场景支持跨地区访问，访问时，本机房延迟最低，其次是本地区的机房，最后是跨地区的机房。目前大部分组件支持本机房优先连接策略，但是当本机房宕机后，可能会出现服务随机连接其他机房的情况。

考虑网络延迟和架构合理性，SOFAMQ 消息队列支持 Region 流量收敛，当本机房宕机后，会优先将流量转发到本地机房，如果本地机房也无法连接，则随机选择其他地区的机房。

本地优先策略为：本机房 > 同地域其他机房 > 跨地域机房。

5.日常运维

5.1. 告警处理

5.1.1. 容器异常告警

在 SOFAStack 消息队列运维过程中，监控会对 SOFAStack 消息队列各组件的状态及指标进行巡检和监控。若巡检和监控异常，会触发告警。

容器异常告警

可能的原因

- 组件进程挂掉
- 容器挂掉

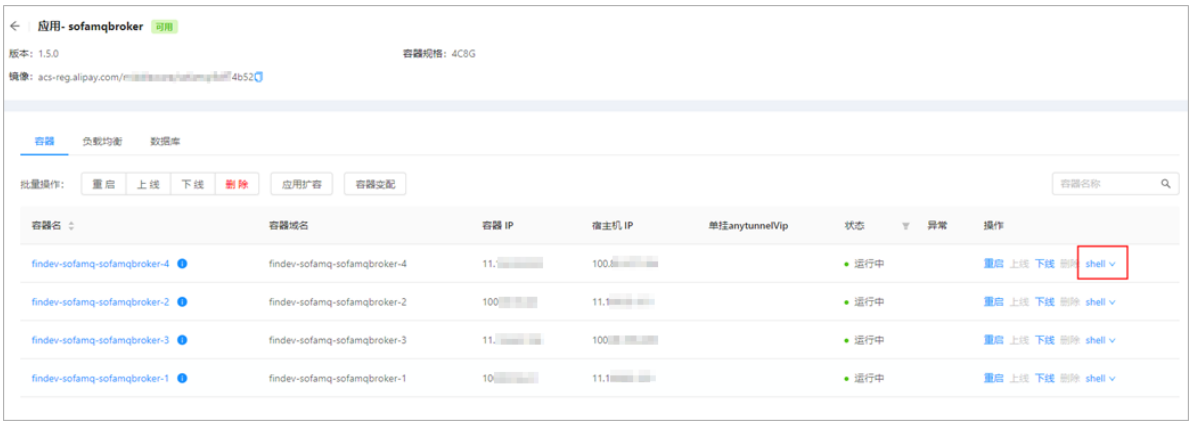
对系统的影响

- 单个容器挂掉：因为组件集群本身的高可用，对业务无影响。
- 容器全部挂掉：因组件不同而影响不同，详情参见 组件可用性运维 中各组件的相关描述。

处理方案

1. 登录当前环境的云游 Local 地址。
2. 在左侧导航栏上，选择 **产品运维 > 产品基线**。
3. 在右上角搜索框中搜索 **SOFAMQ**，单击进入到 SOFAMQ 的产品基线。

   在产品基线中，可以看到 SOFAMQ 的各个组件。
4. 单击异常组件进入详情页，可以看到组件的每一台容器。
5. 在 **容器** 页签，选择异常容器，单击操作列的 **Shell > bash** 进入到节点服务器。



6. 在节点服务器页面执行 `su admin` 命令切换至 admin 用户。

7. 执行 `JPS` 命令查看进程是否被拉起。

```
[root@findev-sofamq-sofamqbroker-4 /root]
# su admin

[admin@findev-sofamq-sofamqbroker-4 /root]
$ jps
5782 Jps
23 BrokerStartup

[admin@findev-sofamq-sofamqbroker-4 /root]
$
```

若JPS查看进程没有查到组件的进程，则按照以下操作步骤拉起进程。

i. 进入目录 `cd /home/admin/bin` 。

ii. 执行进程拉起命令。各组件进程拉起的命令如下：

组件	进程重启命令
Name Server	`/home/admin/sofamq/bin/deploy.sh`
Broker	`/home/admin/sofamq/bin/deploy.sh`
Console	`/home/admin/bin/deploy.sh`

若拉起进程无效，则可能容器挂掉，您可以尝试在云游上重启容器。

说明

您可以在云游上，选择对应的操作，如 **上线** 、**下线**、**重启** 等操作来重启容器。

结果验证

- 登录 RMS 监控控制台，查看相应的告警是否恢复。
- 登录云游控制台，查看容器的状态是否正常（绿色）。

5.1.2. CPU 使用率异常告警

CPU 使用率的阈值为 80%，如果最近 5 分钟 CPU 持续使用率超过 80%，系统会认为 CPU 使用率异常，触发告警。

可能的原因

- 并发访问量高
- 应用实例不足

对系统的影响

影响服务性能。

处理方案

1. 通过云游进入目标容器，具体操作可参考 容器异常告警。
2. 进入目标容器，执行 `top` 命令查看组件的 CPU 使用率。
   - 如果 CPU 使用率不高，结束排查。
   - 如果 CPU 使用率高，需要根据调用量做进一步排查。
3. 执行 `netstat -tnlp | grep -E "80|8080" | wc -l` 命令检查组件的调用情况。
   - 如果调用量偏高，并且符合业务状态，需要对实例进行扩容。
   - 如果调用量不高，需要执行下面的步骤排查 CPU 使用率高的原因。
     a. 执行 `top -Hp <组件进程号>` 命令，找到导致 CPU 或内存使用率高的进程。
     b. 执行 Jstack 命令找出具体的服务，提交工单申请处理。

结果验证

通过云游进入目标容器，执行 `top` 命令，查看 CPU 使用率是否恢复到正常范围（小于 80%）。

5.1.3. 内存使用率异常告警

内存使用率的阈值为 85%，如果内存使用率超过 85%，则认为内存使用率异常。

可能的原因

- 并发访问量高。
- 应用实例不足。

对系统的影响

影响服务性能。

处理方案

1. 通过云游进入目标容器，具体操作可参考 容器异常告警。
2. 执行 `top` 命令，查看组件的内存使用率。
   - 如果内存使用率不高，结束内存使用率异常排查，继续后续磁盘、JVM 的排查。
   - 如果内存使用率高，需要进行排查是否因为 JVM 导致。

3. 执行 `jmap` 命令观察内存使用情况。
4. 执行 `jstat -gcutil [pid]` 命令查看内存使用情况。
5. 执行 `vmstat` 命令分析、统计虚拟内存信息。
6. 执行 `netstat -tnlp | grep -E "80|8080" | wc -l` 命令检查组件的调用情况。
   - 如果调用量偏高，并且符合业务状态，需要对实例进行扩容。
   - 如果调用量不高，需要执行下面的步骤排查 CPU 使用率高的原因。
     a. 执行 `top -Hp <组件进程号>` 命令找出导致 CPU 或内存使用率高的进程。
     b. 执行 `Jstack` 命令找出具体的服务，提交工单申请处理。

结果验证

通过云游进入目标容器，执行 `top` 命令，查看内存使用率是否恢复到正常范围（小于 85%）。

5.1.4. 磁盘使用率异常告警

磁盘使用率的阈值为 80%，如果磁盘使用率超过 80%，则认为磁盘使用率异常。

可能的原因

- 并发访问量高。
- 应用实例不足。
- 磁盘空间不足或没有定期清理。

对系统的影响

影响服务性能。

处理方案

1. 通过云游进入目标容器，具体操作可参考 容器异常告警。
2. 执行 `df -lh` 命令查看各目录的磁盘使用情况，找出消耗过高、过快的目录。
3. 执行 `iostat` 命令查看写数据的情况。

   如果 `logs` 目录下的磁盘使用率过高，可以先进入磁盘清理。

结果验证

- 登录 RMS 监控控制台，查看相应的告警是否恢复。
- 通过云游进入目标容器，执行 `df -lh` 命令，查看磁盘使用率是否恢复到正常范围（小于 80%）。

5.1.5. JVM 异常告警

监控会监控容器 5 分钟内的 JVM 使用情况，如果超过设定的阈值则认为 JVM 异常。

背景信息

JVM 的监控项包括：

- FullGC 次数：1

可能的原因

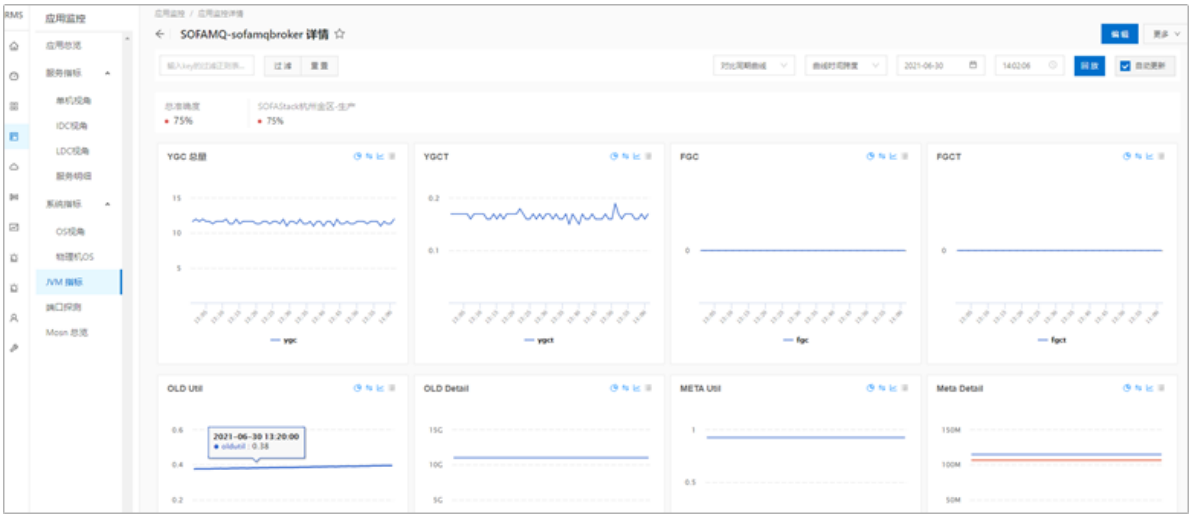
- 并发访问量高
- 应用实例不足
- 磁盘空间不足或没有定期清理

对系统的影响

影响服务性能。

处理方案

1. 登录 RMS 控制台。
2. 在左侧导航栏上，单击 **应用监控 > 全部应用 > SOFAMQ** 进入 SOFAMQ 监控页面。
3. 在左侧导航栏上，单击 **JVM指标** 查看FGC情况。



4. 通过云游进入目标容器，执行 `jmap -heap <pid>` 命令查看各内存区域的设置。
5. 执行 `jmap-histo:live <pid>` 命令，查看每个 class 的实例数目、内存使用率和类全名。每分钟执行一次，连续执行 5 次。
6. 保存收集到的 JVM 相关数据、信息，联系 MQ 技术支持人员处理。

结果验证

- 登录 RMS 监控控制台，查看相应的告警是否恢复。
- 登录 RMS 监控控制台，查看 JVM 的监控项是否在相应的阈值范围内。
- 通过云游进入目标容器，执行 `jmap-histo:live <pid>` 命令查看 JVM 各项监控数据是否恢复到正常范围。

5.2. 组件可用性运维

5.2.1. Broker 发消息异常

现象

- 用户侧反馈发消息 RT 变高，具体表现为在 `com.alipay.sofa.sofamq.client.SofaMQProducerAbstract#send(io.openmessaging.api.Message)` 接口发送耗时长。
- 用户侧反馈发消息失败，报 `[TIMEOUT_CLEAN_QUEUE]broker busy, start flow control for a while, period in queue: 901ms, size of queue: 20034` 错误。

处理方案

可快速通过 jstack 命令确认 `SendMessageThread_` 开头的多数线程卡的位置，快速定位发送消息处理慢的原因。

结果验证

- 隔离故障机器后，通过用户侧观察 `com.alipay.sofa.sofamq.client.SofaMQProducerAbstract#send(io.openmessaging.api.Message)` 接口耗时下降。
- 隔离故障机器后，通过用户侧观察写消息失败异常消失。

5.2.2. sofamq-namesrv 节点异常

现象

sofamq-namesrv 节点异常可能会接收到如下监控异常或告警：

- 监控控制台提示 sofamq-namesrv 组件异常，如果配置了告警将收到告警信息。
- 巡检管理收到 sofamq-namesrv 组件巡检告警。
- sofamq-namesrv 组件的 CPU、内存或 JVM 出现异常，将收到监控指标监控异常。

可能的原因

- 组件的 BUG。
- 严重的误操作。
- 所属的主机宕机。
- 容器、进程异常。
- 网络问题。

对系统的影响

- 单个容器宕机：因为组件集群本身的高可用，对业务无影响。
- 容器全部宕机：
  - 已启动的关联依赖实例因为有本地缓存，不受影响。新启动的生产者和消费者业务应用会异常。
  - 已启动的 Broker 不受影响，但如果已启动的 Broker 挂掉，不能通过 namesrv 通知业务应用，业务应用只能通过已缓存的信息逐个试错。新启动 Broker 时会报错。

处理方案

1. 参考 容器异常告警，处理容器监控异常。
2. 参考 CPU 使用率异常告警，检查 CPU 使用率是否正常。
3. 参考 内存使用率异常告警，检查内存使用率是否正常。

4. 参考 磁盘使用率异常告警，检查磁盘使用率是否征程。
5. 参考 JVM 异常告警，检查 JVM 相关指标是否正常。

结果验证

- 登录监控控制台，查看相关告警是否恢复。
- 登录监控控制台，查看组件、容器的状态是否恢复正常（绿色）。

5.2.3. sofamq-broker 节点异常

现象

sofamq-broker 节点异常可能会接收到如下监控异常或告警：

- 监控控制台提示 sofamq-broker 组件异常，如果配置了告警将收到告警信息。
- 巡检管理收到 sofamq-broker 组件巡检告警。
- sofamq-broker 组件的 CPU、内存或 JVM 出现异常，将收到监控指标监控异常。

可能的原因

- 组件的 BUG。
- 严重的误操作。
- 所属的主机宕机。
- 容器、进程异常。
- 网络问题。

对系统的影响

- Slave 宕机：对业务的收发都没有影响，连着该宕机的消费者会重连到可用的 Broker。
- Master 部分宕机：
  - 不会对消息发送产生影响
  - Group ID 消费端连着该宕机的 Broker 上的消息会被挂起投递，消费端将会根据缓存的 nameserver 同步过来的信息会自动重连其它的 Broker。
  - 宕机的 Broker 的 slave 会自动被 sofamq-namesrv 切换为 master。
- Broker 全部宕机：SOFAStack 消息队列没法对外提供服务，即客户端不能进行收发信息、不能创建 Topic 等操作。

处理方案

1. 参考 容器异常告警，处理容器监控异常
2. 参考 CPU 使用率异常告警，检查 CPU 使用率是否正常。
3. 参考 内存使用率异常告警，检查内存使用率是否正常。
4. 参考 磁盘使用率异常告警，检查磁盘使用率是否征程。
5. 参考 JVM 异常告警，检查 JVM 相关指标是否正常。

结果验证

- 登录监控控制台，查看相关告警是否恢复。

- 登录监控控制台，查看组件、容器的状态是否恢复正常。

5.2.4. sofamq-console 节点异常

现象

sofamq-console 节点异常可能会接收到如下监控异常或告警：

- 监控控制台提示 sofamq-console 组件异常，如果配置了告警将收到告警信息。
- 巡检管理收到 sofamq-console 组件巡检告警。
- sofamq-console 组件的 CPU、内存或 JVM 出现异常，将收到监控指标监控异常。

可能的原因

- 组件的 BUG。
- 严重的误操作。
- 所属的主机宕机。
- 容器、进程异常。
- 网络问题。

对系统的影响

- 单个容器宕机：因为组件集群本身的高可用，对业务无影响。
- 容器全部宕机：用户没法通过控制台执行 SOFAStack 消息队列的 Topic 和 Group ID 等的管理。

处理方案

1. 参考 容器异常告警，处理容器监控异常。
2. 参考 CPU 使用率异常告警，检查 CPU 使用率是否正常。
3. 参考 内存使用率异常告警，检查内存使用率是否正常。
4. 参考 磁盘使用率异常告警，检查磁盘使用率是否正常。
5. 参考 JVM 异常告警，检查 JVM 相关指标是否正常。

结果验证

- 登录监控控制台，查看相关告警是否恢复。
- 登录监控控制台，查看组件、容器的状态是否恢复正常（绿色）。

5.3. 服务巡检

系统组件监控检查

通过监控组件设置 HTTP、TCP、PING 和 DB SOFAMQ 的组件进行巡检。在巡检规则中通过状态码检测设置告警规则，通过告警描述设置告警内容。您也可以登录组件、服务部署的机器执行命令进行巡检。

基础监控检查

巡检对象	描述	指标（关键字）	告警阈值
NameServer	检查 NameServer 的 TCP 端口是否正常	9876	当前 alive 的值等于 0
Broker	检查 Broker 的 TCP 端口是否正常	10911	当前 alive 的值等于 0
Console	检查 Console 的 TCP 端口是否正常	80	当前 alive 的值等于 0
NameServer、Broker、Console	system 中 CPU 使用率监控	cpu_util	最近 5 分钟持续大于 80
	system 中内存使用率监控	mem_util	当前时间的值大于 85
	system 中磁盘使用率监控	max_partition_util	当前时间的值大于 80
	jvmgc 监控 jvm fullGC 情况	fgc_count	当前时间的值大于 1

自定义业务监控检查

监控项	监控项说明	预警触发条件
发布者消息发布数量	统计消息队列接收到的发布者应用发布的消息数量。	-
订阅者消息投递数量	统计消息队列投递到订阅者应用的消息数量。	-
消息队列积压消息数量	统计消息队列消费积压消息数量，一般是消费下游处理消息慢或异常。 预警订阅角色：运维管理员	每分钟超过 1 条。
消息队列异常消息恢复投递数量	统计消息队列针对异常消息的恢复投递数量。 预警订阅角色：运维管理员	每分钟超过 200 条。

监控项	监控项说明	预警触发条件
消息队列接收失败消息数量	统计消息队接收的失败消息数量，一般是消息队列存储资源出现不可用异常或流量超过安全水位触发限流。 预警订阅角色：运维管理员	每分钟超过 1 条。

5.4. 管理容器

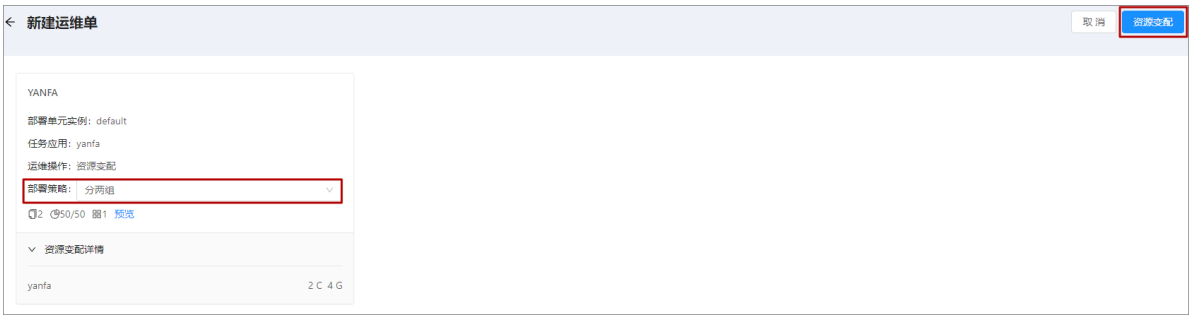
5.4.1. 变更容器配置

您可以通过云游 Local 控制台对应用的容器规格配置进行变更。

操作步骤

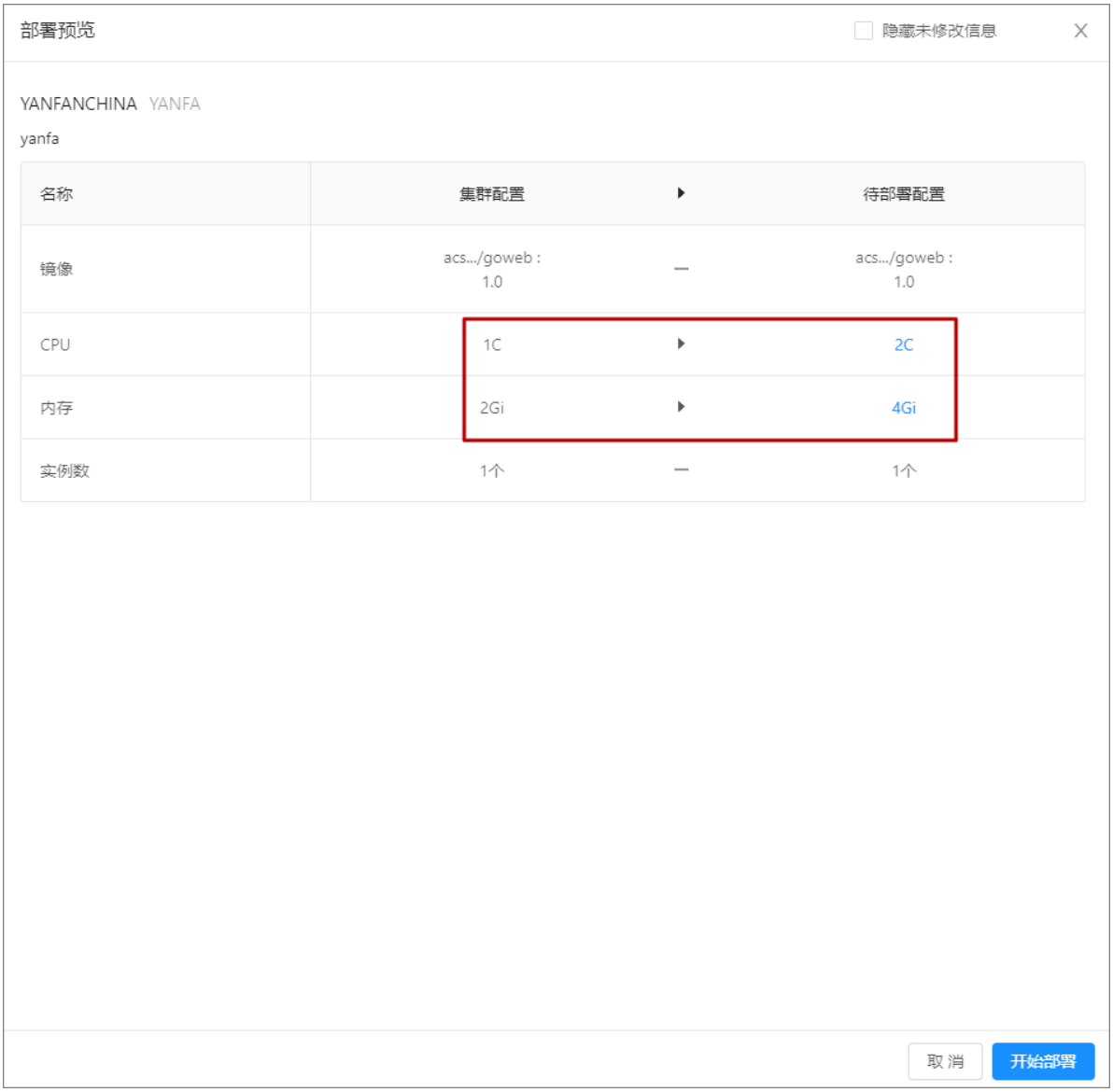
1. 登录云游 Local 控制台。
2. 在左侧导航栏，选择 **产品运维** > **产品基线**。
3. 在产品列表中，单击目标产品。
4. 单击 **应用** 页签，之后单击目标应用。
5. 单击 **容器** 页签，之后单击 **容器变配**。
6. 在 **应用变配规格** 对话框设置 **目标 CPU** 和 **目标内存** 参数，之后单击 **确定**。

7. 在 **新建运维单** 页面，将 **部署策略** 改为 **分两组**。



8. 单击 **资源变配**，在**部署预览**面板确认部署参数无误后单击 开始部署，之后单击 **确定**。

   这里主要确认当前实例的 CPU、内存和期望变更的 CPU、内存。



> **重要**
>
> 变配之后，AntStack Plus 底座环境，容器会自动进行滚动重启；AntStack 1X 和 2X 底座环境，您需要分批次手动重启容器。详情请参见 重启容器。

5.4.2. 容器内存与 JVM 堆内存配置比

中间件各产品容器内存与 JVM 堆内存的配置比如下：

产品名称	JVM 堆内存建议配置
动态配置 DRM	根据环境自动适配，适配方式如下： • 系统内存 = 2 GB；JVM = 1000 MB • 系统内存 = 4 GB；JVM = 3000 MB • 系统内存 = 8 GB；JVM = 6000 MB • 系统内存 = 10 GB；JVM = 9000 MB • 系统内存 = 14 GB；JVM = 12000 MB
注册中心 SOFARegistry	3/4 系统内存
DsrConsole	根据环境自动适配，适配方式如下： • 系统内存 = 2 GB；JVM = 1000 MB • 系统内存 = 4 GB；JVM = 3000 MB • 系统内存 = 8 GB；JVM = 6000 MB • 系统内存 = 10 GB；JVM = 9000 MB • 系统内存 = 14 GB；JVM = 12000 MB
任务调度 TS	根据环境自动适配，适配方式如下： • 系统内存 = 2 GB；JVM = 1000 MB • 系统内存 = 4 GB；JVM = 3000 MB • 系统内存 = 8 GB；JVM = 6000 MB • 系统内存 = 10 GB；JVM = 9000 MB • 系统内存 = 14 GB；JVM = 12000 MB
消息队列 MQ	系统自动调整，无需设置。
消息队列 DMS	Console：系统内存 = 4 GB；JVM = 3 GB Broker：系统内存 = 8 GB；JVM = 4 GB

产品名称	JVM 堆内存建议配置
分布式事务 DTX	JVM 堆内存小于系统内存。 例如系统内存为 8 GB，JVM < 8 GB。
数据访问代理 ODP	根据环境自动适配，适配方式如下： • 系统内存 < 2 GB；JVM 使用系统默认值（根据不同版本进行调整） • 2 GB < 系统内存 < 6 GB；JVM = 2048 MB • 6 GB < 系统内存 < 10 GB；JVM = 4800 MB • 10 GB < 系统内存 < 14 GB；JVM = 5600 MB • 系统内存 > 14 GB；JVM = 9600 MB
Mesh	不涉及
API 网关	不涉及
消息队列 SOFAKafka	系统自动调整，无需设置。

5.4.3. 重启容器

重启容器是指将指定容器（Pod）销毁后重新按期望调度出相同数量的新容器。

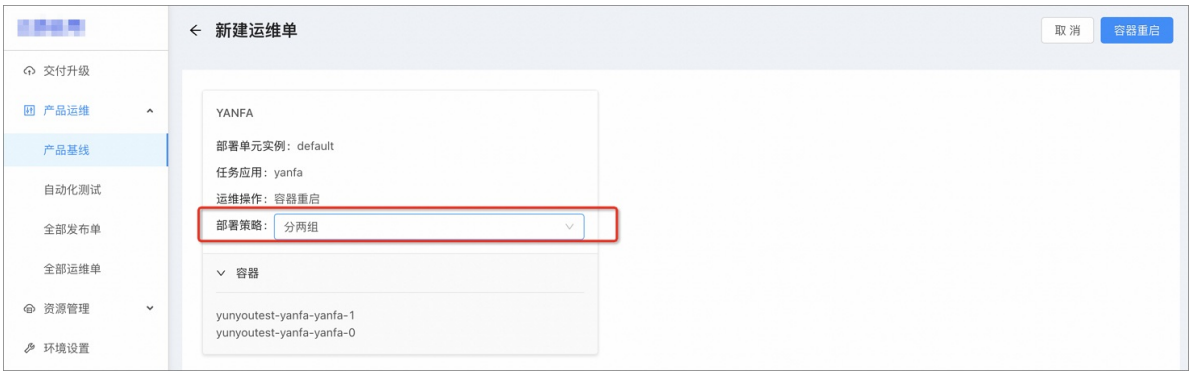
操作步骤

1. 登录云游 Local 控制台。
2. 在左侧导航栏，选择 **产品运维 > 产品基线**。
3. 在产品列表中，单击目标产品。
4. 单击 **应用** 页签，然后单击目标应用。

5. 单击 **容器** 页签，然后单击 **批量操作** 的 **重启** 按钮。

6. 选中一个或多个目标容器后单击 **确定**。
7. 在 **新建运维单** 页面单击 **容器重启**，然后单击 **确定**。

   如果您选择了多个容器，建议将 **部署策略** 修改为 **分两组**，以免所有运行容器同时销毁，导致业务中断。



6.常见故障处理

6.1. 服务可用性故障

6.1.1. 服务可用性故障

现象

用户侧反馈消息从发送到接收延迟较高。

可能的原因

用户的拉消息请求如果命中内存，消息直接从内存投递给消费者，速度快；用户的拉消息请求命中磁盘，消息需要先从文件加载到内存再投递给消费者，速度慢，容易成为瓶颈。

处理方案

可通过执行 `tail -n10000/home/admin/logs/rocketmqlogs/stats.log | grep GROUP_GET_LATENCY |sort -nk14` 命令判断具体由哪些 GID 拉取磁盘的堆积消息导致拉消息请求排队。

结果验证

用户侧反馈消息（非堆积消息）从发送到接收延迟下降。

6.1.2. Broker 运行状态故障

Broker 的 10911 端口探活

监控项

checkservice 告警

现象

1. 登录 RMS 控制台。
2. 在左侧导航栏上，单击 **应用监控 > 全部应用 > SOFAMQ** 进入 SOFAMQ 监控页面。
3. 在左侧导航栏上，单击 **端口探测** 查看端口存活情况。

可能的原因

- Broker 服务有问题。
- 巡检和 Broker 间的网络有问题。
- Broker 所在的服务器有问题。

处理方案

出现该错误通常有两类原因：服务端进程挂掉或者网络异常。

- 服务端进程挂掉：
  i. 登录 Broker 机器，执行 `netstat -anpl|grep 10911` 命令查看端口是否打开。
  ii. 执行 `ps -ef|grep java` 命令查看进程是否存在。

如果进程挂掉，查看日志。执行 `cd /home/admin/logs/rocketmqlogs` 命令分析进程挂掉的时间和原因。

- 网络异常：

  登录同网段机器，用 `telnet <broker ip> 10911` 远程确认 Broker 端口是否可通。如果一切都正常，则说明网络可能有闪断。

6.2. 业务连续性故障

Broker 宕机

现象

RMS 监控控制台的 checkservice 端口探活出现告警，说明 Broker 宕机。异常信息中带有宕机 Broker 的容器 IP。

可能的原因

引发 Broker 宕机的原因较多，比如人为下线 Broker 机器，或者 Broker 进程 OOM，物理机故障重启导致部署在上面的所有容器被关闭。

处理方案

部署底座会检测 Broker 进程的存活状态并自动将宕机的 Broker 重新拉起。Broker 被重新拉起后会发生主备切换，最终重新达到主备同步双写的状态。具体处理方案请参考 容器异常告警。

客户端启动失败

现象

客户端启动失败，报错 “Can not find name server with endpoint” 。

可能的原因

一般是设置的 instanceId、dataCenter（客户端会根据 dataCenter 参数过滤 ACVIP 返回的 sofamqnamesrv 地址）和 endpoint 参数不正确。

处理方案

1. 检查 instanceId、dataCenter 和 endpoint 参数设置。
2. 您可以尝试直接请求 acvip，检查返回的数据，如果出现以下两种情况说明数据有问题：

```
curl --location --request POST 'http://{替换成用户设置的 acvip endpoint}:9003/antcloud/antvip/instances/get' \ --header 'Content-Type: application/json' \ --data '{"allowPolling" : false,"from": "sofamq-client","clientVersion": "direct_http_json","vipDomainName2ChecksumMap": {"{替换成用户设置的 instanceId}-SOFAMQ": "N"}}'
```

   - 返回的地址列表为空，说明该用户没有在 portal 里面开通 SOFAMQ，所以地址没有加到对应的 ACVIP 域名。
   - 返回的地址列表不为空，但是没有任何地址的 dataCenter 属性和用户设置的 dataCenter 一致，两者必有一个是错的。具体的 dataCenter 值请登录云游 Local，在 **环境设置** > **环境变量** > **datacenter** 中查看。

消费端消息处理异常

现象

- 消息堆积增加。
- SOFAStack 消息队列的死信队列增加。
- 消费端的业务处理没有符合预期。

可能的原因

由于业务 BUG、接口故障等问题，导致 SOFAStack 消息队列的消息出现积压，或者出现消息处理错误，甚至出现死信的情况。

影响

业务消息发送缓慢，可能导致重要消息通知延迟，比如延迟收到采购订单报价通知等，影响用户感知。

处理方案

- 排查 Consumer 端的应用日志，找到导致消息处理变慢的根本原因，从而能够加快积压消息的处理。
- 使用 SOFAStack 消息队列的调用链分析工具查看消息的消费过程中的各操作步骤的耗时情况，根据等待外部资源（服务调用、数据库访问、缓存访问等），比如访问数据库耗时长等原因进行分析调优。
- 通过日志查看消费应用是否有异常信息，根据异常提示进行修复工作。
- 尝试增加 Consumer 端的处理进程数量，增大并发度。

Broker 主机磁盘问题导致 SOFAStack 消息队列响应慢

现象

磁盘 util 高。

可能的原因

消息消费时，如果集群中的 GID 都有消息堆积，此时其中几个消费消息再从磁盘中大量拉取冷消息（不在内存中），会导致磁盘的 util 过高。在其它的 SOFAStack 消息队列安装点中，这是相对容易出现的情况，主要原因如下：

- 安装点是机械盘（而非 SSD）。
- 机器磁盘相关的硬件故障（如 RAID 卡故障）。
- 对硬盘进行了热插拔操作。
- 在 Broker 主机进入了其它大量消耗 I/O 的维护性操作。
- inode 或者磁盘空间监控不及时。

> **说明**
>
> 消息发送时，消息发送为顺序写，不会导致 util 很高。

影响业务

消息发送缓慢，可能导致重要消息通知延迟，比如延迟收到采购订单报价通知等，影响用户感知。

处理方案

该情况不能通过重启解决，需要将 GID 分析出来，结合业务方的需求确定。根据业务需求和 GID 冷消息的情况决定 GID 中的消息是否支持延迟消费，如果支持延迟消费，请将该 GID 暂时禁读，等负载降下来再开放。建议 提交工单 申请处理。

完全顺序消息主备 Broker 都不可用

现象

- SOFAStack 消息队列巡检告警。
- 监控巡检、监控告警。
- 完全顺序消息涉及到的业务中断。

可能的原因

- 硬件故障。
- 黑客入侵、病毒等原因导致 2 台 SOFAStack 消息队列 Broker 不可用。
- 运维发生误操作导致 2 台 Broker 不可用，且不可通过重启恢复。

影响

完全顺序消息涉及的业务完全中断，未投递的消息丢失。

处理方案

在控制台上删除对应的完全顺序消息 Topic 后重建，新建的 Topic 会落在正常的那组 Broker 上。

7.常见问题

7.1. 路由表查询失败

现象

报错误日志“No route info of this topic, TP_XXXXX”。

可能的原因

- Endpoint 配置有误。
- NameServer 进程意外退出。
- Topic 没有配置或加载。
- Broker 进程意外退出。
- Broker 进程在但是没有写权限。

处理方案

1. 检查 Endpoint 是否配置正确。

   Client 会根据 Endpoint 拼凑出一个 antvip 地址，比如 `antvip://antvip-pool/sofamqnamesrv-pool.xxx.xxx.xxx.` 实际上会根据规则拼出 `sofamqnamesrv-pool.xxx.xxx.xxx` 这个 antvip 域名，所以可以根据 `~logs/antvip-java-client` 的日志来排查寻址是否正确。

2. 检查 NameServer 是否正确连接。

   通过 `netstat -an|grep 9876` 命令判断是否连接了正确的 NameServer。

3. 检查拉取的 Queue 列表信息是否正确。

   通过 `~/logs/sofamq/rocketmqlogs.log` 日志根据 Topic 查看是否拉取到了正确的 Queue 列表信息。

7.2. 消息积压排查

现象

登录消息队列控制台，在 **消费者状态** 页面，Group ID 的实时消息积压量的值高于预期，且性能明显下降。

可能的原因

部分线程状态异常。

处理方案

1. 在消息队列控制台，通过 查看消费者状态 获取消息堆积的消费者实例所对应的宿主机 IP 地址，并登录该宿主机或容器。

   线程状态的解释说明请参见 Java 官方文档。

2. 执行以下任一命令查看进程 pid。
   - `ps -ef |grep java`
   - `jps -lm`

3. 执行 `jstack -l pid > /tmp/pid.jstack` 命令查看堆栈信息。

4. 执行以下命令查看 `ConsumeMessageThread` 的信息，重点关注线程的状态及堆栈。`cat /tmp/pid.jstack|grep ConsumeMessageThread -A 10 --color` 命令回显如下图所示。

```
"ConsumeMessageThread_28" #906 daemon prio=5 os prio=0 tid=0x00007f08085d9000 nid=0x42c1 waiting for monitor entry [0x00007f07cc30c000]
   java.lang.Thread.State: BLOCKED (on object monitor)
        at com.taobao.txc.a.b.g.c(Unknown Source)
        - waiting to lock <0x0000000702f7d9d0> (a java.lang.Object)
        at com.taobao.txc.a.b.g.a(Unknown Source)
        at com.taobao.txc.a.b.g.b(Unknown Source)
        at com.taobao.txc.a.b.g.a(Unknown Source)
        at com.taobao.txc.client.a.a.a.a(Unknown Source)
        at com.taobao.txc.client.a.a.a.b(Unknown Source)
        at com.taobao.txc.client.a.a.a.a(Unknown Source)
        at com.taobao.txc.client.b.b.a(Unknown Source)
        at com.taobao.txc.client.aop.b.invoke(Unknown Source)
        at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:179)
        at org.springframework.aop.framework.JdkDynamicAopProxy.invoke(JdkDynamicAopProxy.java:207)
        at com.sun.proxy.$Proxy159.orderAutoAdaptation(Unknown Source)
        at com.xtep.o2o.order.adaptation.orderAutoAdaptation.mq.ConsumerOrderAdaptationMQ.consumerMassage(ConsumerOrderAdaptationMQ.java:84)
        at com.xtep.mq.OnsConsumerSpringBean$1.consume(OnsConsumerSpringBean.java:79)
        at com.aliyun.openservices.ons.api.impl.rocketmq.ConsumerImpl$MessageListenerImpl.consumeMessage(ConsumerImpl.java:179)
        at com.alibaba.rocketmq.client.impl.consumer.ConsumeMessageConcurrentlyService$ConsumeRequest.run(ConsumeMessageConcurrentlyService.java:414)
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
        at java.util.concurrent.FutureTask.run(FutureTask.java:266)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
        at java.lang.Thread.run(Thread.java:745)
```

线程状态的解释说明请参见 Java 官方文档。

更多信息

如按以上操作还未解决因消息堆积而导致的性能下降问题，然后 提交工单 获得技术支持。提交时，请附带以下信息：

- `heap.bin` 文件，该文件可通过执行 `jmap -dump:format=b,file=heap.bin [pid]` 命令获取，再执行 `gzip heap.bin` 命令生成压缩包。

  > **说明**
  >
  > [pid] 是消息堆积处理第一步中找到的进程 pid。

- 发生消息堆积的消费者客户端 `sofamq.log` 本地日志。
- 消费者客户端的版本。

8.管理控制台

8.1. 资源管理

8.1.1. 节点管理

您可以在 **节点管理** 页面查看所有节点的详细信息，包括 hostname、状态、vip、rip、集群、机房、数据中心以及节点组。本节介绍如何在 SOFA MQ 管理控制台查找节点和修改节点权限。

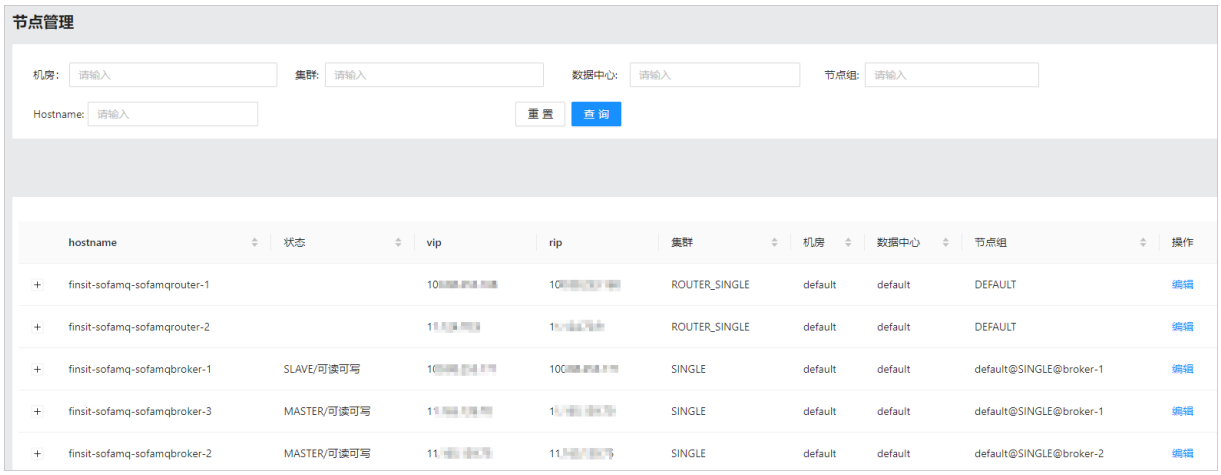
节点概述

消息队列包含 Broker、Namesrv、Router 节点，Broker 节点负责将 Topic 的路由信息上报至 Namesrv 节点，Namesrv 节点负责 Broker 的选主和集群管理，Router 节点通过复制消息流量实现跨逻辑寻址单元（Cell）通信。Broker、Namesrv、Router 节点的逻辑图请参见 组件角色说明。

查找节点

1. 登录 SOFA MQ 管理控制台。
2. 在左侧导航栏上，单击 **资源管理** > **节点管理** 进入 **节点管理** 页面。

   您可以根据机房、集群、数据中心、节点组、hostname 来查找目标节点。



修改节点权限

1. 在左侧导航栏上，单击 **资源管理** > **节点管理** 进入 **节点管理** 页面。
2. 单击目标节点 **操作** 列下的 **编辑**。
3. 在弹出的信息提示框中，修改节点权限。

- 0：不可读写
- 2：可写
- 4：可读
- 6：可读可写

> **说明**
>
> 除修改节点权限外，其他配置项不建议修改。

4. 单击 **提交**。

8.2. 元数据管理

8.2.1. Topic 管理

Topic（消息主题）是一级消息类型，一般通过 Topic 对消息进行分类，每个 Topic 下会由一到多个队列来存储消息。您可以在 **Topic 管理** 页面查看所有 Topic 的详细信息，包括实例 ID、Topic、集群、消息类型、读写权限、应用名、读分区数、写分区数、生效范围、版本以及创建时间。

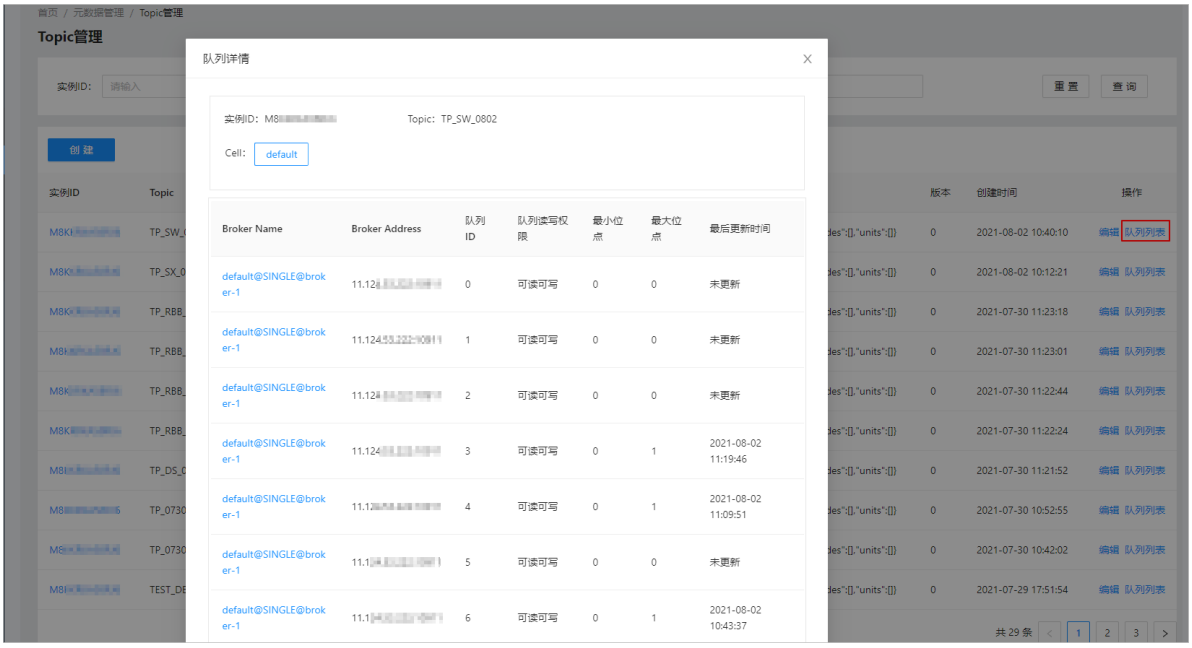
查询 Topic

1. 登录 SOFA MQ 管理控制台。
2. 在左侧导航栏上，单击 **元数据管理 > Topic 管理**。

   您可以根据实例 ID、集群、Topic 来查找目标 Topic。

查看订阅信息

1. 在左侧导航栏上，单击 **元数据管理** > **Topic 管理** 进入 **Topic 管理** 页面。
2. 单击目标 Topic 名称操作列下的 **队列列表** 查看该 Topic 下各个队列信息详情，包括 Broker Name、Broker Address、队列 ID、队列读写权限等。



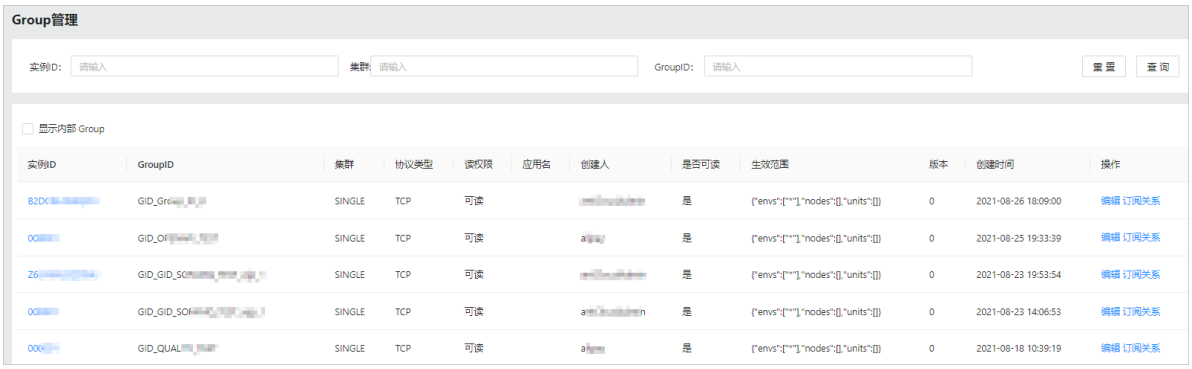
8.2.2. Group 管理

1 个 Group 下的生产者或消费者通常是生产或消费同一类消息，且消息发布或订阅的逻辑一致。您可以在 **Group 管理** 页面查看整个环境下各个工作空间的所有 Group 详细信息，包括实例 ID、Group ID、集群、协议类型、读权限、应用名、创建人、是否可读、生效范围、版本以及创建时间。

查询 Group

1. 登录 SOFA MQ 管理控制台。
2. 在左侧导航栏上，单击 **元数据管理** > **Group 管理**。

   您可以根据实例 ID、集群、Group ID 来查找目标 Group。



查看订阅信息

1. 在左侧导航栏上，单击 **元数据管理** > **Group 管理** 进入 **Group 管理** 页面。

2. 单击目标 GroupID 名称操作列下的 **订阅关系**，查看该 Group 的在线情况和订阅关系情况。

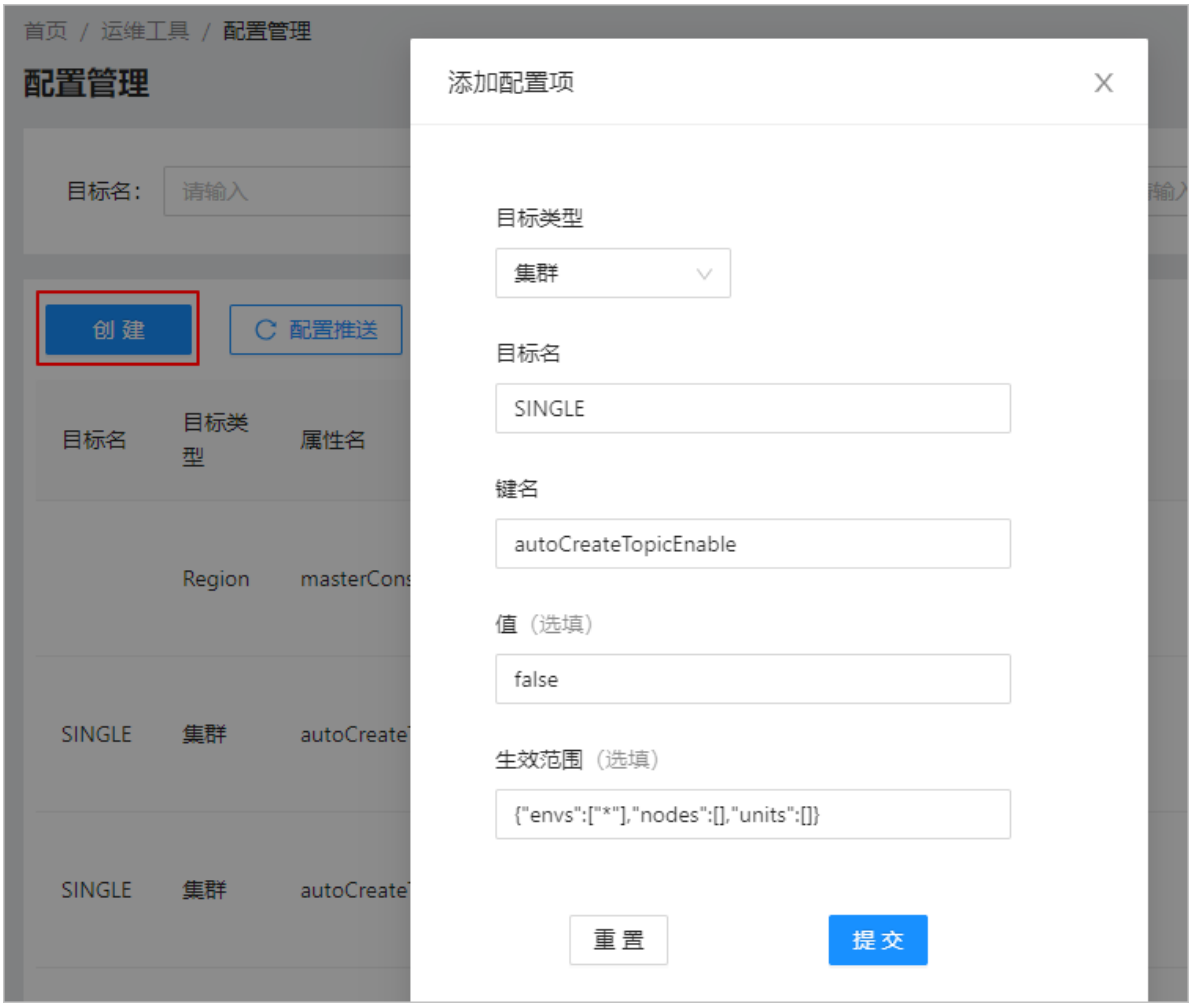
8.3. 运维工具

8.3.1. 配置管理

在 **配置管理** 页面，您可以根据目标名、目标类型和目标属性来查找目标。本文主要介绍如何对 Broker 推送配置。

创建配置项

1. 登录 SOFAMQ 管理控制台。
2. 在左侧导航栏上，单击 **运维工具** > **配置管理**。
3. 单击 **创建**，在弹出的页面中配置以下信息。



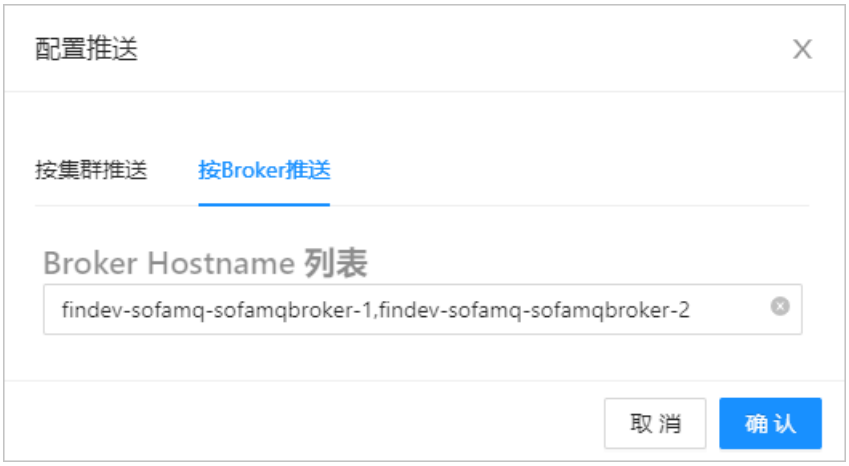
参数	是否必填	说明
目标类型	必填	可选 **集群**、**单机** 或 **Region**。本例中选择：`集群`。
目标名	必填	输入目标名。本例中输入：`SINGLE`。
键名	必填	输入键名。本例中输入：`autoCreateTopicEnable`。
值	选填	输入值。本例中输入：`false`。
生效范围	选填	输入生效范围。本例中输入：`{"envs":["*"],"nodes":[],"units":[]}`。

4. 单击 **提交**。

推送配置

1. 在 **配置管理** 页面，单击 **配置推送**。
2. 选择推送方式。
   - **按集群推送**：选择目标集群及机房。

   - **按 Broker 推送**：输入 Broker Hostname 列表，以逗号“,”间隔。



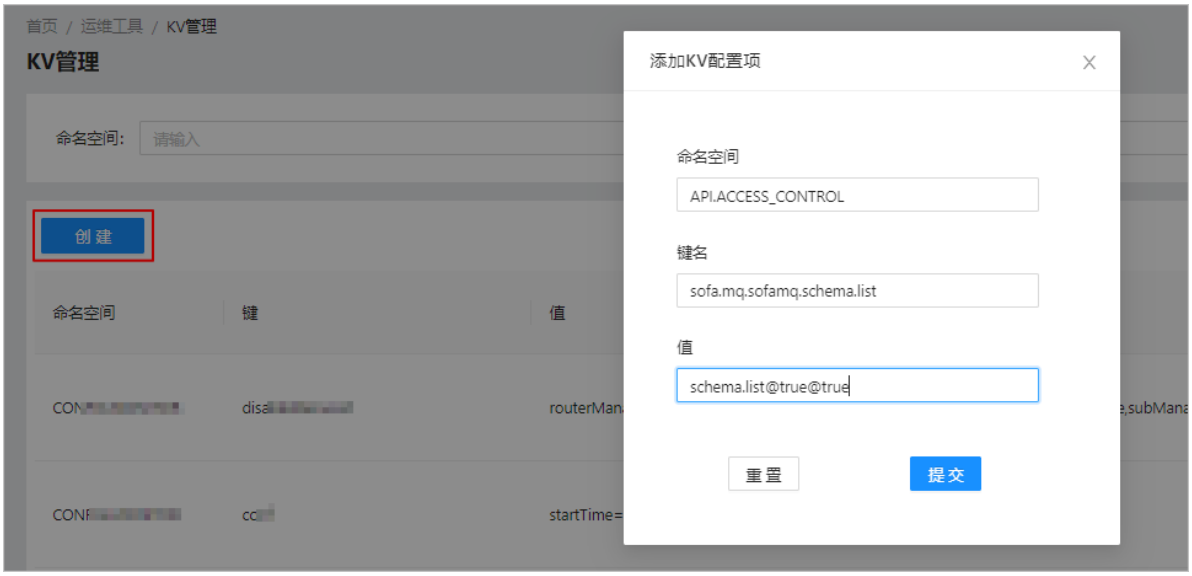
3. 单击 **确认**。

8.3.2. KV 管理

KV 配置主要用于支撑一些内部开发需求，例如根据 KV 配置控制控制台标签展示。您可以在 **KV 管理** 页面查看所有实例的详细信息，包括命名空间、键、KV 值、操作人、创建时间以及修改时间。

创建 KV 配置项

1. 登录 SOFA MQ 管理控制台。
2. 在左侧导航栏上，单击 **运维工具** > **KV 管理**。
3. 单击 **创建**，在弹出的页面中配置以下信息。

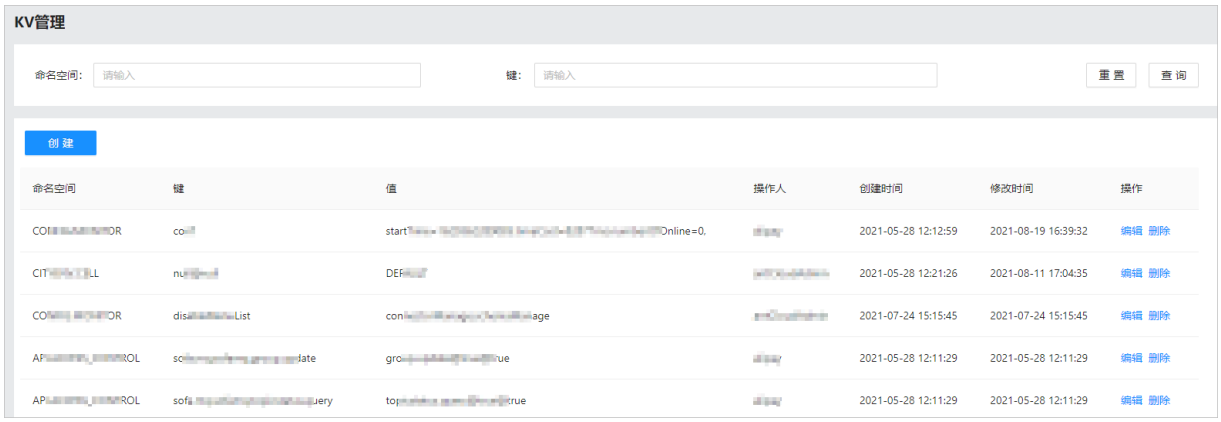


参数	说明
命名空间	输入命名空间名称。
键名	输入键名。
值	输入值。

4. 单击 **提交**。

更多操作

除了创建 KV 配置项，您还可以进行修改、删除和查询操作。



- 如需修改 KV 配置项，请单击 **操作** 列下的 **编辑**，在修改 KV 配置项页修改相关信息，然后单击 **提交**。
- 如需删除 KV 配置项，请单击 **操作** 列下的 **删除**，然后在弹出的提示对话框中单击 **确认**。
- 如需查询 KV 配置项，您可以在 **KV 管理** 页面根据命名空间、键来查找 KV 值。

8.4. 权限管理

8.4.1. 管理员列表

您可以在 **管理员列表** 页面查看所有用户的详细信息，包括用户名、用户类型、操作人以及创建时间。本文介绍如何在 SOFA MQ 管理控制台创建、删除、查询用户。

创建用户

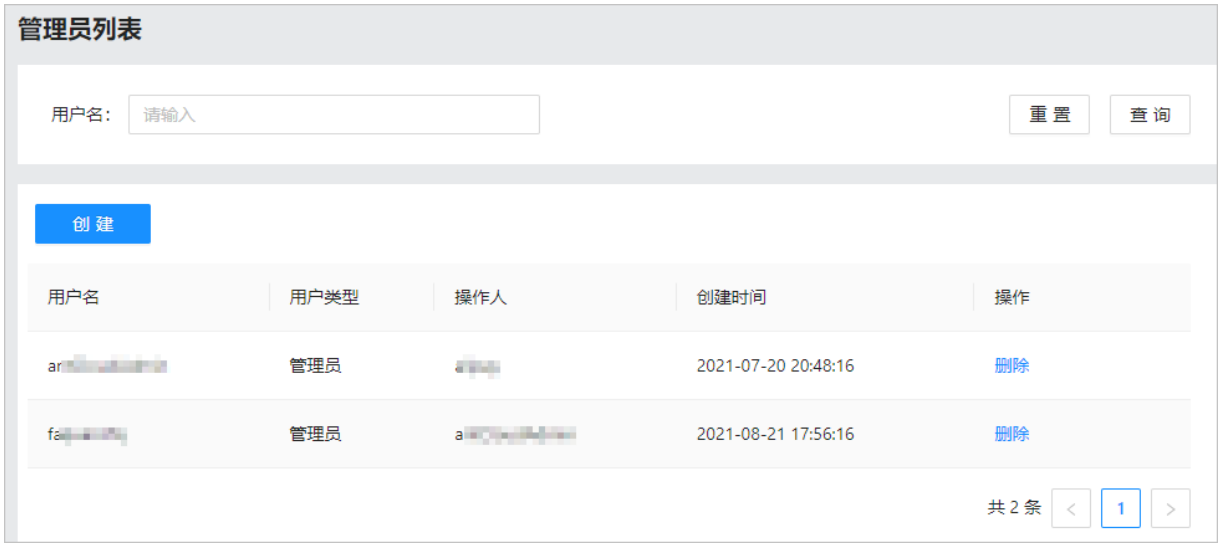
1. 登录 SOFA MQ 管理控制台。
2. 在左侧导航栏上，单击 **权限管理** > **管理员列表**。
3. 单击 **创建**，在弹出的对话框配置以下信息。

参数	说明
用户名	输入域名。
用户类型	选择用户类型，可选 **管理员** 或 **普通用户**。 ○ **管理员**： ■ 可以登录 SOFA MQ 管理控制台进行资源管理、元数据管理、日常运维操作。 ■ 可以登录消息队列控制台进行创建资源、收发消息等操作。 ○ **普通用户**： 可以登录消息队列控制台进行创建资源、收发消息等操作。

4. 单击 **提交**。

其他操作

除了创建用户，您还可以进行删除和查询操作。



- 如需删除用户，请目标用户名 **操作** 列下的 **删除**，然后在弹出的提示对话框中单击 **确认**。
- 如需查询用户，您可以在 **管理员列表** 页面根据用户名查找目标用户。