



APUSIC
固若长城
睿比世界

用户手册

金蝶Apusic分布式消息队列V2.0.5

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 产品简介
 - 2.1 产品概述
 - 2.2 产品功能
 - 2.3 产品架构
 - 2.3.1 概述
 - 2.3.2 发布订阅
 - 2.3.3 跨地域复制
 - 2.3.4 多租户
 - 2.3.5 权限控制
 - 2.3.6 事务支持
 - 2.3.7 消息分块传输
 - 2.3.8 传输安全
 - 2.3.9 与其他系统集成
 - 2.3.10 多副本存储
 - 2.3.11 多集群传输通道
 - 2.3.12 黑白名单
 - 2.3.13 可视化管控台
 - 2.3.14 监控运维
 - 2.4 产品优势
 - 2.4.1 计算与存储分离
 - 2.4.2 高性能低延迟
 - 2.4.3 无限制的主题存储
 - 2.4.4 无缝故障恢复
 - 2.4.5 队列和流模型的结合
 - 2.4.6 内置轻量级计算引擎
 - 2.4.7 统一认证和授权
 - 2.4.8 运维监控
 - 2.4.9 多租户隔离
 - 2.4.10 消息查询
 - 2.4.11 国密SSL传输
 - 2.4.12 多协议接入
 - 2.5 应用场景
 - 2.5.1 系统解耦
 - 2.5.2 异步处理
 - 2.5.3 削峰填谷
 - 2.5.4 基于优先级的消息推送
 - 2.5.5 跨地域数据复制
 - 2.5.6 分级存储
 - 2.5.7 数据分析
 - 2.5.8 顺序消息
 - 2.5.9 分布式事务
 - 2.6 相关概念
 - 2.6.1 消息

- 2.6.2 主题
 - 2.6.3 计算节点
 - 2.6.4 存储节点
 - 2.6.5 发布订阅
- ### 3 产品安装
- 3.1 基本概念
 - 3.1.1 ADMQ 集群
 - 3.1.2 Broker
 - 3.1.3 元数据存储
 - 3.1.4 持久化存储
 - 3.1.5 管理控制台
 - 3.2 产品部署说明
 - 3.2.1 部署架构图
 - 3.2.2 部署硬件需求(单机最低配置)
 - 3.2.3 部署硬件需求(集群最低配置)
 - 3.2.4 部署硬件需求(推荐集群配置)
 - 3.2.5 安装介质说明
 - 3.3 授权模式说明
 - 3.4 安装管理控制台
 - 3.4.1 安装前准备
 - 3.4.2 开始安装
 - 3.4.2.1 管理控制台安装
 - 3.4.2.2 启动管理控制台服务
 - 3.4.2.3 停止管理控制台服务
 - 3.4.3 验证
 - 3.5 通过管控台安装ADMQ
 - 3.5.1 安装前准备
 - 3.5.1.1 管控台导入授权
 - 3.5.1.2 上传软件包
 - 3.5.1.3 新增可用区 (可选)
 - 3.5.1.4 新增服务器信息
 - 3.5.1.5 确认License中允许的节点数量
 - 3.5.2 安装ADMQ
 - 3.5.2.0.1 新增集群
 - 3.5.2.0.2 部署集群
 - 3.5.2.0.3 启动集群节点
 - 3.5.3 插件安装
 - 3.5.3.1 插件加载方式安装 (推荐使用)
 - 3.5.3.2 配置文件方式安装 (不推荐使用)
 - 3.5.3.2.1 Kafka 插件
 - 3.5.3.2.2 RocketMQ 插件
 - 3.5.3.2.3 RabbitMQ 插件
 - 3.5.3.2.4 MQTT 插件
 - 3.5.3.3 插件使用
 - 3.5.3.3.1 Kafka 插件
 - 3.5.3.3.2 RocketMQ 插件

- 3.5.3.3.3 RabbitMQ 插件
 - 3.5.3.3.4 MQTT 插件
 - 3.6 通过命令行安装ADMQ
 - 3.6.1 节点授权
 - 3.6.2 单机部署
 - 3.6.3 集群部署
 - 3.6.3.1 部署协调器
 - 3.6.3.2 初始化元数据
 - 3.6.3.3 存储组件
 - 3.6.3.4 计算组件
- ## 4 快速入门
- 4.1 客户端使用
- ## 5 管控台使用
- 5.1 登录管理控制台
 - 5.2 系统配置
 - 5.2.1 服务器管理
 - 5.2.1.1 新增服务器
 - 5.2.1.2 编辑服务器
 - 5.2.1.3 删除服务器
 - 5.2.1.4 查询服务器
 - 5.2.1.5 部署采集器
 - 5.2.2 License管理
 - 5.2.2.1 导入License
 - 5.2.2.2 导出License
 - 5.2.2.3 删除License
 - 5.2.3 软件包管理
 - 5.2.3.1 上传软件包
 - 5.2.3.2 删除软件包
 - 5.2.4 默认配置
 - 5.2.4.1 新增配置信息
 - 5.2.4.2 修改配置信息
 - 5.2.4.3 删除配置信息
 - 5.2.4.4 启用配置参数
 - 5.2.4.5 停用配置信息
 - 5.2.4.6 下载配置文件
 - 5.2.4.7 查询配置项信息
 - 5.2.5 操作日志
 - 5.2.5.1 操作日志查询
 - 5.2.6 事件管理
 - 5.2.6.1 查看事件详细信息
 - 5.2.6.2 查询事件信息
 - 5.3 集群
 - 5.3.1 集群管理
 - 5.3.1.1 新增集群
 - 5.3.1.2 容器化注册 (docker)
 - 5.3.1.3 容器化注册 (k8s)

- 5.3.1.4 非容器化注册
- 5.3.1.5 配置集群信息
 - 5.3.1.5.1 ADMQ集群配置
 - 5.3.1.5.1.1 基础配置
 - 5.3.1.5.1.2 资源配置
 - 5.3.1.5.1.3 安全配置
 - 5.3.1.5.1.4 其他配置
 - 5.3.1.5.2 RocketMQ的配置页面
 - 5.3.1.5.2.1 基础配置
 - 5.3.1.5.2.2 安全配置
 - 5.3.1.5.2.3 存储设置
- 5.3.1.6 删除集群
- 5.3.1.7 查询集群详细信息
- 5.3.1.8 证书管理
 - 5.3.1.8.1 替换证书
 - 5.3.1.8.2 下发证书
 - 5.3.1.8.3 下载证书
 - 5.3.1.8.4 编辑证书
- 5.3.1.9 依赖管理
 - 5.3.1.9.1 新增依赖
 - 5.3.1.9.2 替换依赖
 - 5.3.1.9.3 下发依赖
 - 5.3.1.9.4 下载依赖
 - 5.3.1.9.5 编辑依赖
 - 5.3.1.9.6 删除依赖
- 5.3.1.10 插件管理
 - 5.3.1.10.1 加载插件
 - 5.3.1.10.2 替换插件
 - 5.3.1.10.3 下发插件
 - 5.3.1.10.4 编辑插件
- 5.3.1.11 资源隔离
- 5.3.1.12 元数据初始化
- 5.3.1.13 命令行操作
- 5.3.1.14 节点相关操作
 - 5.3.1.14.1 部署节点
 - 5.3.1.14.2 重新部署节点
 - 5.3.1.14.3 启动节点
 - 5.3.1.14.4 停止节点
 - 5.3.1.14.5 重启节点
 - 5.3.1.14.6 编辑节点
- 5.3.1.15 集群扩缩容
 - 5.3.1.15.1 删除节点
 - 5.3.1.15.2 新增节点
- 5.3.1.16 配置节点参数
 - 5.3.1.16.1 新增配置参数
 - 5.3.1.16.2 修改配置参数

- 5.3.1.16.3 删除配置参数
- 5.3.1.16.4 启用配置参数
- 5.3.1.16.5 停用配置参数
- 5.3.1.16.6 下载配置文件
- 5.3.1.16.7 查询配置项
- 5.3.1.17 一键操作
 - 5.3.1.17.1 一键部署
 - 5.3.1.17.2 一键启动
 - 5.3.1.17.3 一键停止
- 5.3.1.18 节点资源监控
- 5.4 用户
 - 5.4.1 用户管理
 - 5.4.1.1 新增用户
 - 5.4.1.2 编辑用户
 - 5.4.1.3 删除用户
 - 5.4.1.4 修改用户密码
 - 5.4.1.5 查看用户令牌
 - 5.4.1.6 查看用户令牌权限
 - 5.4.1.7 普通用户新增令牌
 - 5.4.1.7.1 普通用户新增令牌流程
 - 5.4.1.7.2 新增普通用户
 - 5.4.1.7.3 新增角色
 - 5.4.1.7.4 角色授权
 - 5.4.1.7.5 普通用户分配角色权限
 - 5.4.1.7.6 新增令牌
- 5.5 角色
 - 5.5.1 角色管理
 - 5.5.1.1 新增角色
 - 5.5.1.2 编辑角色
 - 5.5.1.3 删除角色
 - 5.5.1.4 角色授权
 - 5.5.1.4.1 分配菜单权限
 - 5.5.1.4.2 分配数据权限
 - 5.5.1.5 角色分配
- 5.6 资源管理
 - 5.6.1 租户
 - 5.6.1.1 新增租户
 - 5.6.1.2 编辑租户
 - 5.6.1.3 删除租户
 - 5.6.1.4 查询租户
 - 5.6.1.5 副本设置
 - 5.6.2 命名空间
 - 5.6.2.1 新增命名空间
 - 5.6.2.2 编辑命名空间
 - 5.6.2.3 删除命名空间
 - 5.6.2.4 查询命名空间

- 5.6.2.5 新增命名空间权限
- 5.6.2.6 编辑命名空间权限
- 5.6.2.7 删除命名空间权限
- 5.6.2.8 副本设置
- 5.6.2.9 开启消息轨迹
- 5.6.2.10 关闭消息轨迹
- 5.6.3 主题
 - 5.6.3.1 新增主题
 - 5.6.3.2 编辑主题
 - 5.6.3.3 删除主题
 - 5.6.3.4 查询主题
 - 5.6.3.5 主题分区扩容
 - 5.6.3.6 查看主题详情
 - 5.6.3.7 新增主题权限
 - 5.6.3.8 编辑主题权限
 - 5.6.3.9 删除主题权限
- 5.6.4 订阅组
 - 5.6.4.1 新增订阅组
 - 5.6.4.2 编辑订阅组
 - 5.6.4.3 删除订阅组
 - 5.6.4.4 查询订阅组
 - 5.6.4.5 查看订阅组详情
 - 5.6.4.6 消息回溯
 - 5.6.4.7 下载未消费
 - 5.6.4.8 下载未确认
- 5.7 通道
 - 5.7.1 通道管理
 - 5.7.1.1 新增通道
 - 5.7.1.1.1 原生ADMQ通道
 - 5.7.1.1.2 原生RocketMQ通道
 - 5.7.1.2 编辑通道
 - 5.7.1.3 删除通道
- 5.8 消息查询
 - 5.8.1 消息查询
 - 5.8.1.1 新增消息查询
 - 5.8.1.2 查看消息内容
 - 5.8.1.3 查看消息轨迹
 - 5.8.1.4 查询已创建的消息查询
 - 5.8.1.5 删除已创建的消息查询
- 5.9 系统运维
 - 5.9.1 运维监控
 - 5.9.2 系统告警
 - 5.9.3 功能验证
 - 5.9.3.1 新增功能验证
 - 5.9.3.2 编辑功能验证
 - 5.9.3.3 执行功能验证

- 5.9.3.4 删除功能验证
- 5.9.3.5 查看历史执行功能验证
- 5.9.4 数据迁移
 - 5.9.4.1 元数据迁移_导出
 - 5.9.4.2 元数据迁移_导入
 - 5.9.4.3 数据迁移_导出
 - 5.9.4.4 数据迁移_导入
- 5.9.5 元数据同步
- 5.9.6 节点日志
 - 5.9.6.1 下载节点日志
 - 5.9.6.2 查看节点日志
 - 5.9.6.3 删除节点日志
 - 5.9.6.4 查询节点日志
- 5.10 插件管理
 - 5.10.1 RocketMQ管理
 - 5.10.1.1 租户
 - 5.10.1.1.1 新增租户
 - 5.10.1.1.2 编辑租户
 - 5.10.1.1.3 删除租户
 - 5.10.1.1.4 查询租户
 - 5.10.1.1.5 查看租户详情
 - 5.10.1.2 命名空间
 - 5.10.1.2.1 新增命名空间
 - 5.10.1.2.2 编辑命名空间
 - 5.10.1.2.3 删除命名空间
 - 5.10.1.2.4 查询命名空间
 - 5.10.1.2.5 新增命名空间权限
 - 5.10.1.2.6 编辑命名空间权限
 - 5.10.1.2.7 删除命名空间权限
 - 5.10.1.3 主题
 - 5.10.1.3.1 新增主题
 - 5.10.1.3.2 编辑主题
 - 5.10.1.3.3 删除主题
 - 5.10.1.3.4 查询主题
 - 5.10.1.3.5 查看主题详情
 - 5.10.1.3.6 新增主题权限
 - 5.10.1.3.7 编辑主题权限
 - 5.10.1.3.8 删除主题权限
 - 5.10.1.3.9 发送消息
 - 5.10.1.4 订阅组
 - 5.10.1.4.1 新增订阅组
 - 5.10.1.4.2 编辑订阅组
 - 5.10.1.4.3 删除订阅组
 - 5.10.1.4.4 查询订阅组
 - 5.10.1.4.5 查看订阅组详情
 - 5.10.1.4.6 重置消费位点

- 5.10.1.5 消息查询
 - 5.10.1.5.1 新增消息查询
 - 5.10.1.5.2 查看消息详情信息
 - 5.10.1.5.3 查询已创建的消息查询
- 5.10.2 Kafka管理
 - 5.10.2.1 租户
 - 5.10.2.1.1 新增租户
 - 5.10.2.1.2 编辑租户
 - 5.10.2.1.3 删除租户
 - 5.10.2.1.4 查询租户
 - 5.10.2.1.5 查看租户详情
 - 5.10.2.2 命名空间
 - 5.10.2.2.1 新增命名空间
 - 5.10.2.2.2 编辑命名空间
 - 5.10.2.2.3 删除命名空间
 - 5.10.2.2.4 查询命名空间
 - 5.10.2.2.5 发布消费权限
 - 5.10.2.2.5.1 新增权限
 - 5.10.2.2.5.2 编辑权限
 - 5.10.2.2.5.3 删除权限
 - 5.10.2.3 主题
 - 5.10.2.3.1 新增主题
 - 5.10.2.3.2 编辑主题
 - 5.10.2.3.3 删除主题
 - 5.10.2.3.4 查询主题
 - 5.10.2.3.5 查看主题详情
 - 5.10.2.3.6 发布消费权限
 - 5.10.2.3.6.1 新增权限
 - 5.10.2.3.6.2 编辑权限
 - 5.10.2.3.6.3 删除权限
 - 5.10.2.3.7 发送消息
 - 5.10.2.4 订阅组
 - 5.10.2.4.1 新增订阅组
 - 5.10.2.4.2 编辑订阅组
 - 5.10.2.4.3 删除订阅组
 - 5.10.2.4.4 查询订阅组
 - 5.10.2.4.5 查看订阅组详情
 - 5.10.2.4.6 重置消费位点
 - 5.10.2.5 Kafka认证
 - 5.10.2.5.1 认证设置
 - 5.10.2.5.2 SASL/SCRAM机制
- 5.10.3 RabbitMQ管理
 - 5.10.3.1 租户
 - 5.10.3.1.1 新增租户
 - 5.10.3.1.2 编辑租户
 - 5.10.3.1.3 删除租户

- 5.10.3.1.4 查询租户
- 5.10.3.1.5 查看租户详情
- 5.10.3.2 虚拟主机
 - 5.10.3.2.1 新增虚拟主机
 - 5.10.3.2.2 编辑虚拟主机
 - 5.10.3.2.3 删除虚拟主机
 - 5.10.3.2.4 查询虚拟主机
 - 5.10.3.2.5 发布消费权限
 - 5.10.3.2.5.1 新增权限
 - 5.10.3.2.5.2 编辑权限
 - 5.10.3.2.5.3 删除权限
- 5.10.3.3 交换机
 - 5.10.3.3.1 新增交换机
 - 5.10.3.3.2 编辑交换机
 - 5.10.3.3.3 删除交换机
 - 5.10.3.3.4 查询交换机
 - 5.10.3.3.5 查看交换机详情
 - 5.10.3.3.6 发送消息
- 5.10.3.4 队列
 - 5.10.3.4.1 新增队列
 - 5.10.3.4.2 编辑队列
 - 5.10.3.4.3 删除队列
 - 5.10.3.4.4 查询队列
 - 5.10.3.4.5 查看队列详情
 - 5.10.3.4.6 清空队列信息
- 5.10.3.5 路由
 - 5.10.3.5.1 新增路由信息
 - 5.10.3.5.2 编辑路由信息
 - 5.10.3.5.3 删除路由信息
 - 5.10.3.5.4 查询路由
 - 5.10.3.5.5 查看路由详情
 - 5.10.3.5.6 重新绑定
- 5.10.3.6 RabbitMQ插件认证授权
 - 5.10.3.6.1 认证
 - 5.10.3.6.2 授权
 - 5.10.3.6.3 TLS/SSL连接
- 5.11 监控运维
 - 5.11.1 管控台集成AUC和AMP
 - 5.11.1.1 AUC登录
 - 5.11.1.2 注册ADMQ服务并跳转到ADMQ界面
 - 5.11.1.3 节点跳转到监控页面
 - 5.11.1.4 各个组件监控界面
 - 5.11.2 监控指标
 - 5.11.2.1 zookeeper监控指标
 - 5.11.2.2 storage监控指标
 - 5.11.2.3 broker监控指标

- 5.11.3 AMP配置告警策略
 - 5.11.3.1 配置zookeeper告警策略
 - 5.11.3.2 配置storage告警策略
 - 5.11.3.3 配置broker告警策略
- 5.11.4 AMP监控指标展示
 - 5.11.4.1 按核心组件维度展示
 - 5.11.4.2 按业务视图维度展示
 - 5.11.4.2.1 创建ADMQ监控业务
 - 5.11.4.2.2 创建ADMQ应用
 - 5.11.4.2.3 为ADMQ应用绑定设施
 - 5.11.4.2.4 查看ADMQ监控业务
 - 5.11.4.3 自定义仪表盘
- 6 性能测试
- 7 信创适配
- 8 常见问题
 - 8.1 概念相关
 - 8.1.1 是否支持国产数据库
 - 8.1.2 是否支持通过其他消息中间件的客户端发送消息
 - 8.1.3 能够支持其他消息中间件的版本
 - 8.1.4 是否能够同时支持多种消息协议
 - 8.1.5 是否有权限管理功能
 - 8.1.6 消费者的四种订阅模式区别
 - 8.1.7 消息持久化和非持久化区别
 - 8.1.8 死信队列
 - 8.1.9 批量消息发送
 - 8.1.10 消息加密
 - 8.1.11 消息保留策略
 - 8.1.12 支持发送哪些类型的消息
 - 8.1.13 租户、命名空间、主题、用户的作用和说明
 - 8.1.14 消息查询
 - 8.2 部署相关
 - 8.2.1 单机启动失败
 - 8.2.2 节点启动失败
 - 8.2.3 没有java执行权限
 - 8.2.4 管控台启动节点报JAVA_HOME not set
 - 8.2.5 协调器连接超时或失败
 - 8.2.6 存储节点日志报Failed to get cluster instance id
 - 8.3 操作相关
 - 8.3.1 不同格式的消息发送
 - 8.3.2 查看历史消息
 - 8.3.3 license文件
 - 8.3.4 权限
 - 8.3.5 消息确认
 - 8.3.6 怎么设置不同的租户
 - 8.3.7 客户端启动后没有收到消息
 - 8.3.8 同一个订阅名称是否能设置不同的订阅模式

- 8.3.9 使用事务时程序报错
- 8.3.10 使用java客户端收发消息样例
- 8.4 测试相关
 - 8.4.1 有哪些第三方测试认证
- 8.5 其他材料

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
2024年12月	V1.0	金蝶Apsic分布式消息队列V2.0.5	首次编写

2 产品简介

2.1 产品概述

金蝶天燕分布式消息中间件（ADMQ）是一款云原生消息中间件，采用计算和存储分离的基础架构，增强了系统的伸缩性。ADMQ在数据一致性上做了重点研究，能严格验证各个副本的数据强一致性，从而能更好的服务于金融等对数据敏感的行业。

表1-主要能力

主要能力	解释
消息发布订阅	一对多消费模式，发布者可以将消息发送到主题，被一个或多个消费者同时消费。
多租户支持	在共享服务器资源的情况下，实现不同租户间的数据隔离，保障用户隐私。
多协议接入	支持原生Kafka、AMQP、MQTT和JMS协议的接入，无需修改代码即可完成到ADMQ的迁移。
多客户端支持	提供Java、Go、Python、C++和C#等多语言客户端。
数据一致性保证	数据多副本存储，在数据写入成功后保证各个副本数据的一致性，提供可重复读级别的一致性保证。
跨地区数据同步	支持同步、异步两种方式的异地数据同步。
计算和存储分离	无状态的计算节点+分层存储，集群在理论上支持无限制的扩展。

2.2 产品功能

ADMQ是一个分布式消息中间件，核心功能点包含以下几个方面。

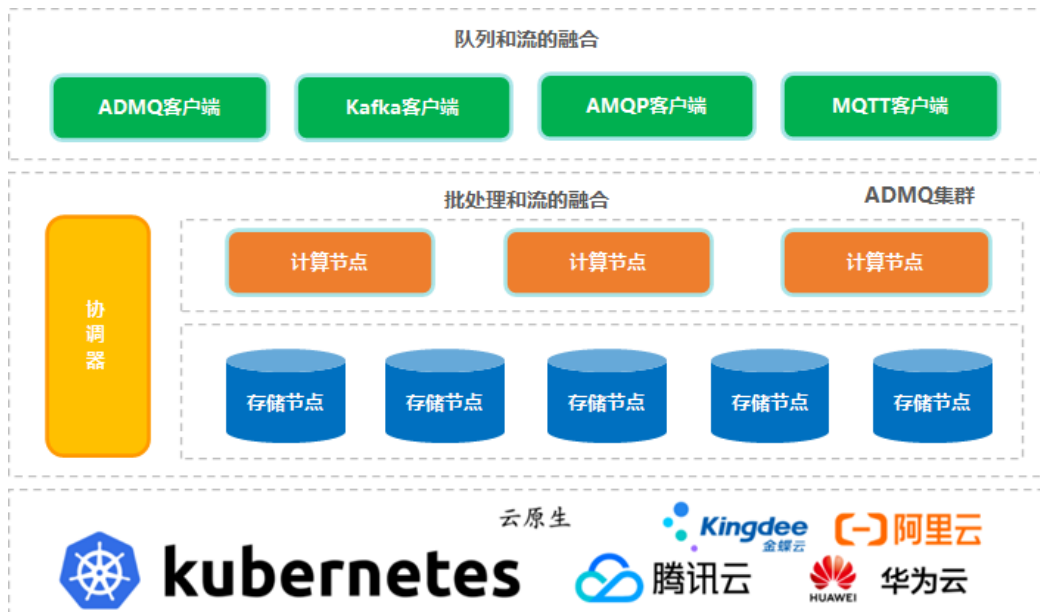
表2-核心功能

核心功能	解释
消息接收	同步接收、异步接收、消息确认、批量确认、累计确认、重复消费、死信主题、消息订阅、多主题订阅、多种订阅模式、消费历史消息
消息存储	消息持久化、数据备份、同步策略、多级存储、集群间消息复制、单点多磁盘存储、数据清理、自动切换、同城多活、异地多活
消息传输	支持消息解压缩、消息可靠传输、断点续传。
文件传输	支持文件可靠传输、断点续传。
事务控制	原子性写入和读取、读取事务消息、跨主题事务
认证授权	身份认证、身份鉴权、多级账号、加密传输、数据隔离
多协议适配	Kafka On ADMQ、AMQP On ADMQ、JMS On ADMQ、MQTT On ADMQ
集群管理	节点管理、负载均衡、动态扩展
消息查询	开启消息轨迹记录功能后可查询主题下消息的接收、存储和消费情况
优先级队列	在同一主题内可设置消息的优先级，高优先级消息有更高的接收权重
国密SSL	在消息传输过程中支持国密SSL加密传输
跨地域复制	支持持久化的消息在多个集群间备份，实现数据同步功能
黑白名单	支持IP、网段的黑白名单过滤，严格控制客户端访问的IP或网段
可视化管控台	提供集群自动化部署、资源管理、可视化监控运维等功能

2.3 产品架构

2.3.1 概述

ADMQ由三部分组成：无状态的服务层，负责处理生产者发出的消息，并将这些消息分派给消费者；存储层负责消息的持久化存储；Zookeeper进行元数据存储、集群配置和协调。



2.3.2 发布订阅

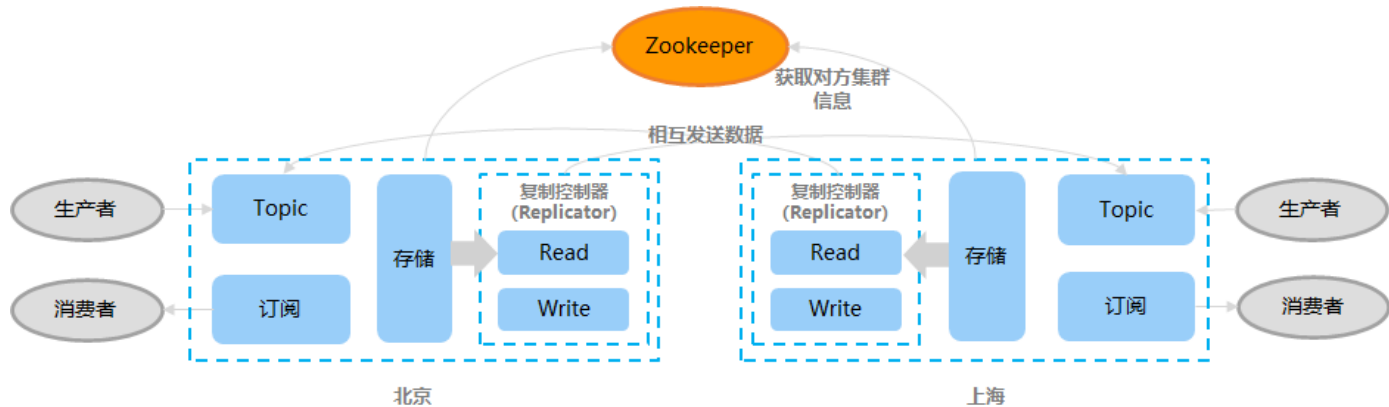
通过“订阅”抽象出了统一的消费模型: 生产者-主题-订阅-消费者。ADMQ的消息模型既支持队列模型，也支持流模型。消息在主题上仅存储一次，但是用户可以有不同的订阅方式来消费这些消息：

- 消费者被组合在一起消费消息，每个消费组是一个订阅。
- 每个主题可以有不同的消费组。
- 每组消费者都是对主题的一个订阅。
- 每组消费者可以拥有自己不同的消费模型
 - 独占 (Exclusive)：在任何时间，一个消费者组中有且只有一个消费者来消费主题中的消息
 - 故障切换 (Failover)：多个消费者可以附加到同一订阅。但是一个订阅中的所有消费者，只会有一个消费者被选为该订阅的主消费者，其他消费者将被指定为故障转移消费者。
 - 共享 (Share)：同一个订阅，用户按照应用的需求挂载任意多的消费者。订阅中的所有消息以循环分发形式发送给多个消费者，并且一个消息仅传递给一个消费者。
 - key共享：同一个订阅，用户可根据给定的映射规则，订阅中相同key的消息发送给同一个消费者。

2.3.3 跨地域复制

跨地域复制是将ADMQ中持久化的消息在多个集群间备份。ADMQ在核心服务上构建了跨地域复制功能，用户可以在部署时选择同步或异步配置，并且可以按主题配置复制机制。生产者可以从任何地区写入共享主题，ADMQ负责确保这些信息对各地的消费者均可见。

ADMQ只需进行简单配置，即可在多个方向复制单个主题到任意数量的外部数据中心。ADMQ在后台进行数据同步，由存储层处理，不会影响服务层正在处理的其他工作负载。



北京和上海的集群共用一个全局Zookeeper，用于相互同步集群信息。

生产者发送消息到集群后并存储后，集群会启动复制流程，从存储中读取消息，然后发送到另外的集群。

2.3.4 多租户

每个租户都可以有单独的认证和授权机制。租户也是存储配额、消息 TTL 和隔离策略的管理单元。多租户基础设施可以跨多个用户和组织共享，同时保证它们彼此隔离。一个租户的活动不会影响其他租户的安全或SLA。

多租户可以从两个方面降低成本。首先，简单地共享单个租户没有充分利用的基础设施——将组件的成本分摊到所有用户。第二，通过简化管理——当有几十、几百或几千个租户时，管理一个实例明显简单许多。

2.3.5 权限控制

支持在租户和命名空间上管理和分配用户的权限。当消息生产者和消费者连接到集群之间，需要对生产者和消费者使用的资源进行权限的分配，保障数据的安全性。

支持主-子账户权限模型，子账户拥有的权限是主账户的子集。

2.3.6 事务支持

多个消息发送和接收可包含在一个事务里，ADMQ保证一个事务中的所有消息发送和消息接收的单元操作要么全部成功，要么全部失败。通过对事务的支持，ADMQ可以将多个消息的发送包含进一个事务中，从而轻松地实现诸如关联交易等业务。

2.3.7 消息分块传输

ADMQ对单条消息的长度没有严格限制，在消息传输协议中使用4字节存储消息负载长度，理论上最大支持传输2G 的消息内容。

为了保证传输性能，每条消息的长度最好限制在10M以内。对于比较大的消息，ADMQ支持分块传输的方式，在客户端侧把消息切割成多个消息块，当所有的消息块都传输成功后，ADMQ再按顺序把所有消息块组装成一条完整的消息发送给订阅客户端。对于基于其他例如MQTT协议传输的消息，消息最大长度是协议本身限制的，ADMQ不会做任何限制。

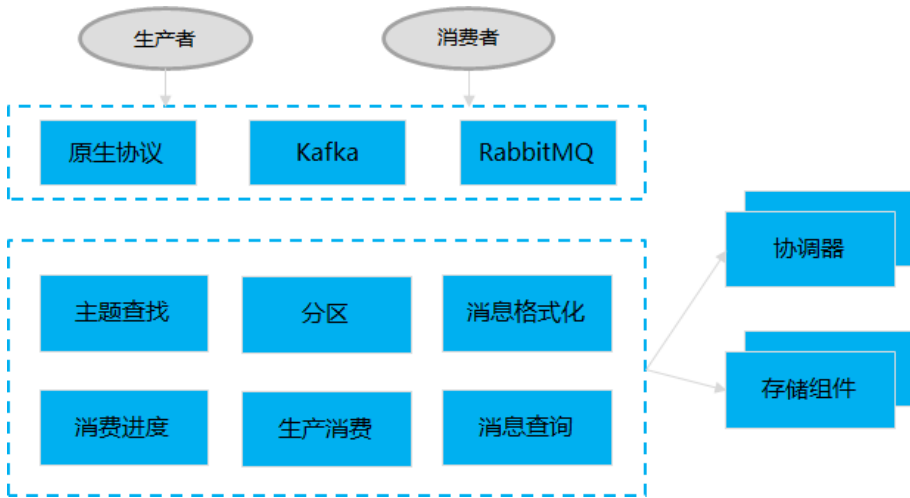
2.3.8 传输安全

支持TLS协议，从而实现客户端与服务器之间、以及服务器与服务器之间的传输通道安全保护；支持国密SSL。

支持传输过程中的消息进行加密。生产者和消费者可以提前约定私钥，生产者发送消息时对消息进行加密，消费者通过私钥对消息进行解密，从而保证消息在传输过程中的安全性，也能避免当其他无密钥的消费者解析消息。

2.3.9 与其他系统集成

ADMQ在消息接收和处理中间添加了中间层 - 协议处理框架，可以处理其他协议的客户端发送的消息。例如用户可以通过Kafka、RabbitMQ客户端发送和接收消息，从而在不改变原来代码的情况下无缝转换到ADMQ。



计算和存储是相互独立的，存储组件的核心功能就是消息的存储和查询，不会涉及到业务相关的逻辑。

计算部分把涉及到消息处理的部分抽象出API接口，包括消息的读写、Topic发布和管理、分区处理、消息内容格式化、消费进度和消息查询等功能API的抽象。当客户端使用其他协议接入时，就可以很方便的把对应的操作转换成抽象API中的方法。

这种方式能很容易的适配其他协议的客户端，目前支持Kafka协议、AMQP协议、JMS协议和MQTT协议。

2.3.10 多副本存储

ADMQ接收到消息后可以把消息副本存储到多个节点上，同时可基于地域或者机房的策略存储，保证副本平分到不同的地域和机房，从而提高数据的可用性。

2.3.11 多集群传输通道

ADMQ内置了跨集群消息传输功能，当其中一个集群收到消息后自动转发到其他集群，从而实现多集群、跨网络的消息传输功能。

支持租户、命名空间的维度配置消息的转发目的地。

2.3.12 黑白名单

ADMQ支持IP、网段的黑白名单过滤，严格控制客户端访问的IP或网段。

2.3.13 可视化管控台

集群管理：部署管理ADMQ、RocketMQ集群；证书管理、依赖管理、插件管理，节点管理；配置集群信息、配置节点参数；自动扩容；节点资源监控等。

用户管理：新增用户信息、编辑用户信息、删除用户信息、修改用户密码、查看令牌、查看权限。

通道管理：跨集群间的消息复制可视化界面，包括：新增通道信息、编辑通道信息、删除通道信息。

资源管理：管理监控租户、命名空间、主题。

消息查询：监控查询消息的轨迹。

系统运维：监控运维、系统告警、功能验证、节点日志。

系统配置：服务器管理、license管理、软件包管理、默认配置管理、操作日志、事件管理。

插件管理：RocketMQ管理、Kafka管理、RabbitMQ管理。

2.3.14 监控运维

管控台自带的监控运维模块可用于监控管理集群信息并以图表的形式直观显示出来。

集成auc登录平台和amp监控平台，集群节点可通过管控台监控自动跳转到amp监控图标界面，对节点资源情况进行监控。

对于ROP插件、KOP插件，管控台分别从租户、主题、订阅组三个维度对资源生产消费情况进行监控；对于AOP插件，管控台从租户、交换机、队列三个维度对资源生产消费情况进行监控。

提供HTTP监控接口。

2.4 产品优势

2.4.1 计算与存储分离

采用计算与存储分离的云原生架构，将消息的存储和服务分开，可实现存储层和服务层的独立扩展。扩容过程无需任何数据再平衡，不会将旧数据从现有存储节点复制到新存储节点，从而降低了网络带宽和I/O的消耗。

2.4.2 高性能低延迟

能够高效支持百万级消息生产和消费，海量消息堆积且消息堆积容量不设上限。单集群 QPS 超过10万，同时在时耗方面有保护机制来保证低延迟，满足业务性能需求。

2.4.3 无限制的主题存储

主题被分割成分片，以分布式方式存储，因此主题的容量不受单一节点容量的限制。

2.4.4 无缝故障恢复

由于服务层是无状态节点，所以某一个节点宕机对整体的生产和消费没有任何影响，集群会根据负载将主题重新分配给可用的节点。

一条消息在被确认之前会被持久化到N个存储节点，数据在多个节点上都可以被访问，并且可以在N-1个节点故障中存活。通过添加新的存储节点来替换失败的节点，不影响整个集群的可用性。

2.4.5 队列和流模型的结合

队列模型主要是采用无序或者共享的方式来消费消息，而流模型要求消息的消费严格排序或独占消息消费。

ADMQ 提供了统一的消息模型和API，做到了队列模型和流模型的统一。在主题级别只需保存一份数据，可被多次消费，为用户提供了更多灵活性，方便用户程序以最匹配模式来使用消息系统。

2.4.6 内置轻量级计算引擎

内置轻量级计算引擎，让开发人员可以使用Java或Python实现函数（处理逻辑），为用户提供了一个部署简单、运维方便的 FaaS（Function as a service）平台。此功能使用户可以享受无服务器计算（serverless）的好处，类似于AWS的Lambda计算。

2.4.7 统一认证和授权

提供统一的可视化界面，管理租户、命名空间、主题的创建和销毁，并分配用户相应的使用权限，保证用户的数据安全。

2.4.8 运维监控

拥有管控一体的可视化平台，提供服务可视化部署，简化部署流程，集群资源实现多维度监控告警功能，方便运维管理。

2.4.9 多租户隔离

提供多租户间隔离机制，保证租户之间互不干扰，保障用户隐私，同时对每个租户的资源进行管理监控，实时查看生产消费情况。

2.4.10 消息查询

支持在命名空间层开启消息轨迹记录功能，开启后会自动记录消息的接收、存储、消费和消费确认状态，在管控台可查询消息轨迹。

2.4.11 国密SSL传输

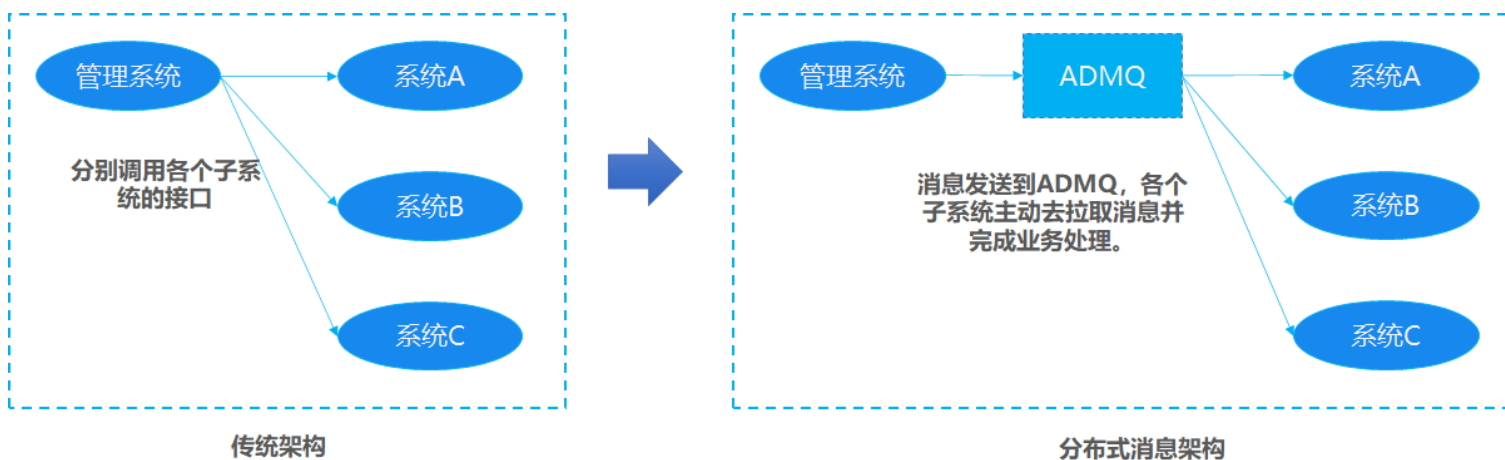
在消息传输过程中支持国密SSL加密传输。

2.4.12 多协议接入

客户端支持 Java、C++、Python、GO 多种语言；另外，还支持 Kafka、RocketMQ、RabbitMQ 客户端的接入，若用户只是利用消息队列的基础功能对消息进行生产和消费，可以不用修改代码就能完成到ADMQ的迁移工作；同时还支持MQTT、JMS 等协议接入。

2.5 应用场景

2.5.1 系统解耦



- 对接多个子系统，适配难度增加。
- 多个子系统的交互耦合在一起，系统整体可用性降低。

- 解耦合，提高系统灵活性。
- 提高并行度，减少业务耗时。

业务描述：

假如有一个灾害指挥中心，遇到告警时需要通知各个灾害处理子系统进行相应的灾害处理。

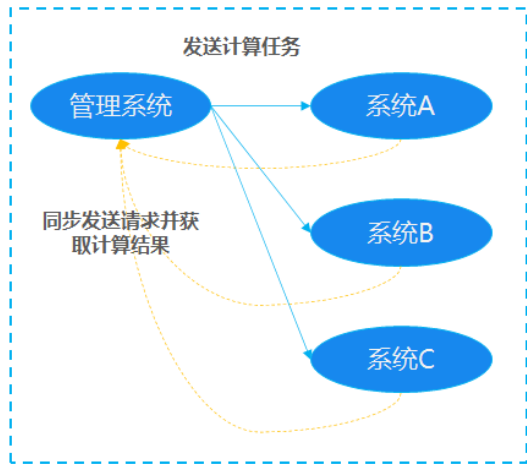
传统架构：

需要给每一个子系统发送一条通知，如果其中某个子系统连接不上，就会影响后续的处理流程。

消息队列架构：

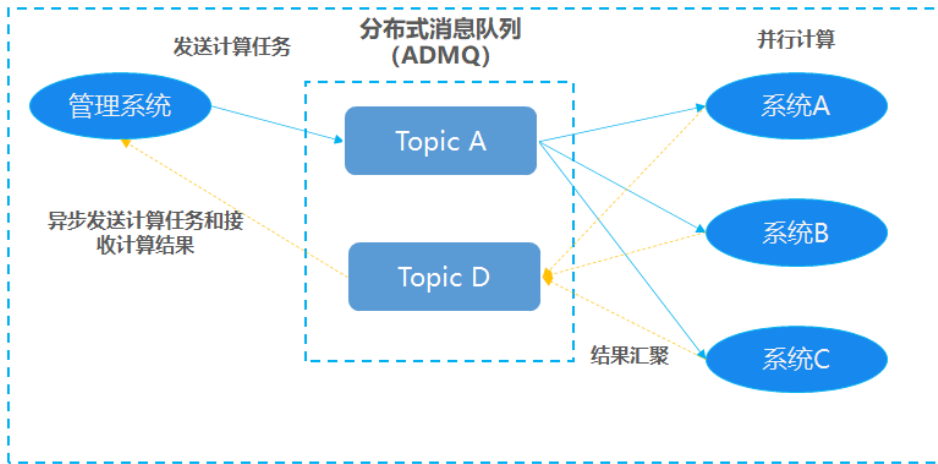
在有消息独队列的架构下，管理系统只需要把通知发送到消息队列，由消息队列复制消息给每一个子系统。这种方式提高了系统的灵活性，当有新的子系统接入时或者子系统的消息接收接口改变时，其他子系统不需要做任何改动，只需要相应的子系统自己做适配就好。

2.5.2 异步处理



传统架构

需要分别发送计算任务到三个子系统中，然后等待各个子系统的计算结果并汇聚结果，同时要处理各子系统任务接收失败的情况。



分布式消息架构

把计算任务发送到消息队列，各个子系统拉取任务后独立计算并把结果发送到消息队列中另外一个Topic上，管理系统可以订阅该Topic获取计算结果。

和系统解耦类似，有一点区别就是消息通知到各个子系统后需要得到子系统的反馈。

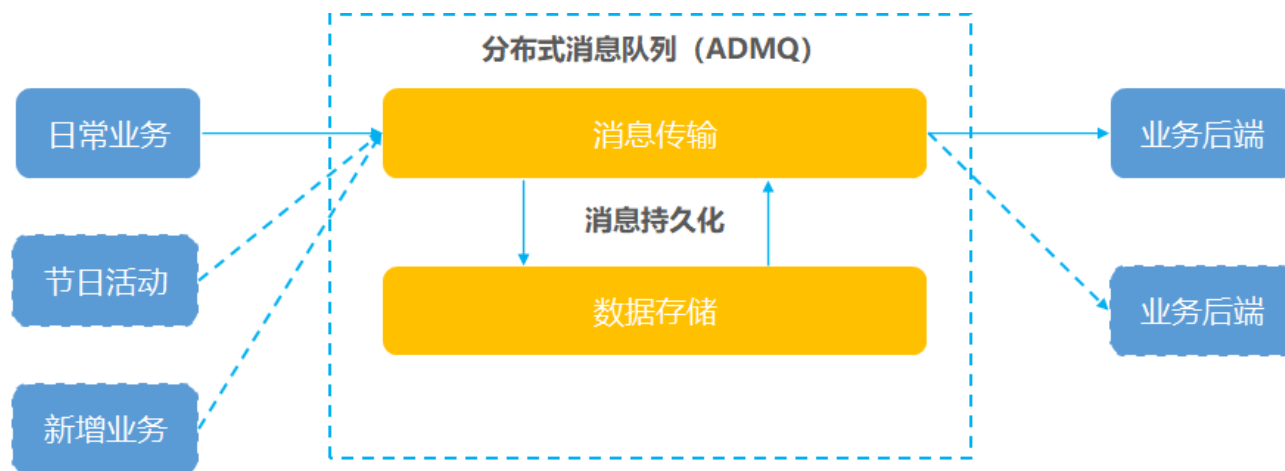
传统架构：

传统架构下就是在发送消息后阻塞后续流程等待子系统处理完毕后反馈结果。如果子系统处理失败了，还需要重新发送消息，也就是管理系统需要知道各个子系统的状态，这种方式增加了管理系统的复杂度。

消息队列架构：

在有消息队列的情况下，管理系统可以把消息发送到其中一个主题，子系统从该主题订阅消息，处理完成后反馈结果到另外一个主题上。管理系统通过另外一个主题获取子系统处理结果。这种异步处理的方式使得管理系统和各个子系统都不需要关注对方的状态，只要保证能从消息队列拿到消息并成功把消息或者处理结果发送到消息队列即可。

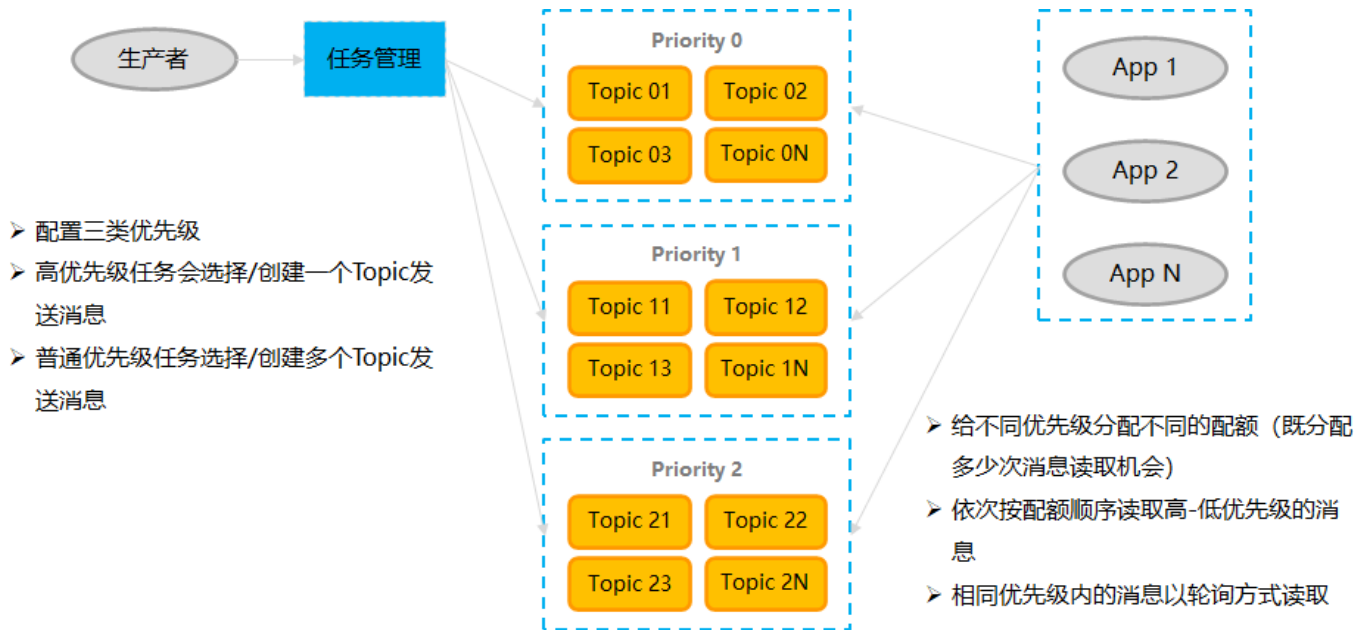
2.5.3 削峰填谷



上游因节日活动、新增业务等因素造成系统压力过大时，依赖于分布式消息队列的缓存能力缓存突增的流量，下游业务后端可以根据自己的处理能力从消息队列中拉取消息并处理，有效保障业务的可用性。

消息队列本身就是系统之间的队列，天然有消息缓存能力。当消息发送方在某个时间点突然增大消息量时，消息队列可以把接收到的消息持久化到硬盘上，消息接收方仍然按照自己的处理能力从消息队列拉取消息进行处理，避免因系统要处理的流量突增导致系统崩溃。

2.5.4 基于优先级的消息推送



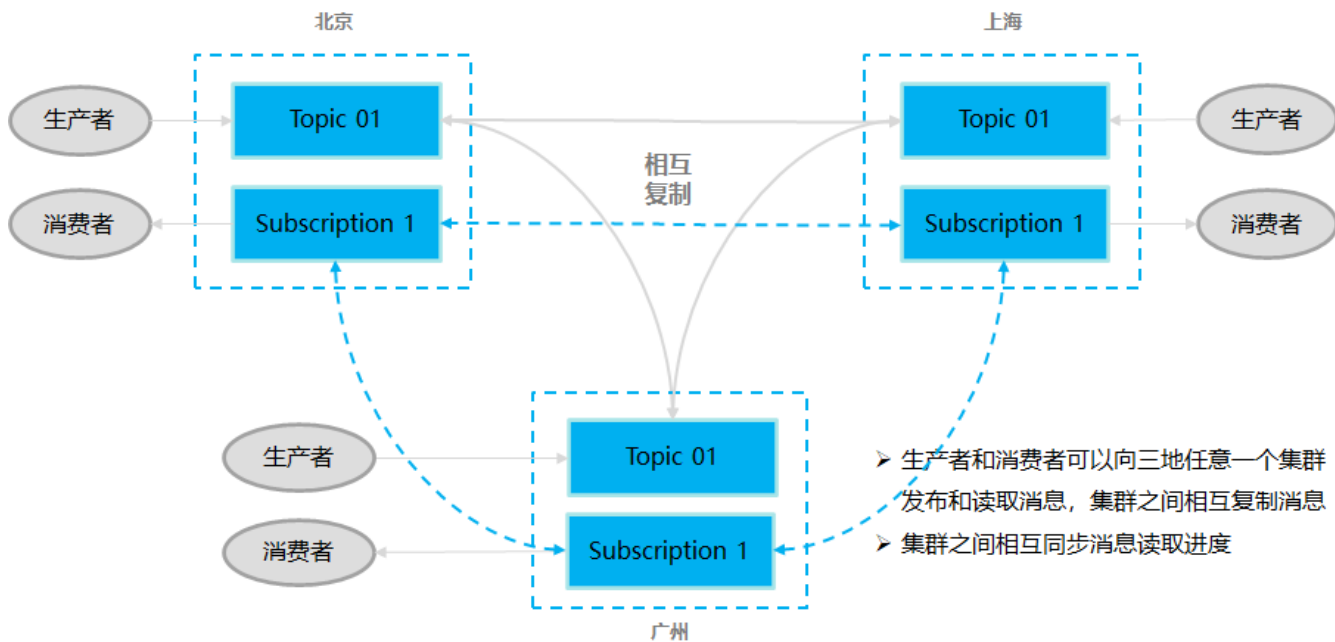
业务场景：

推送新闻到用户的APP上。

处理逻辑：

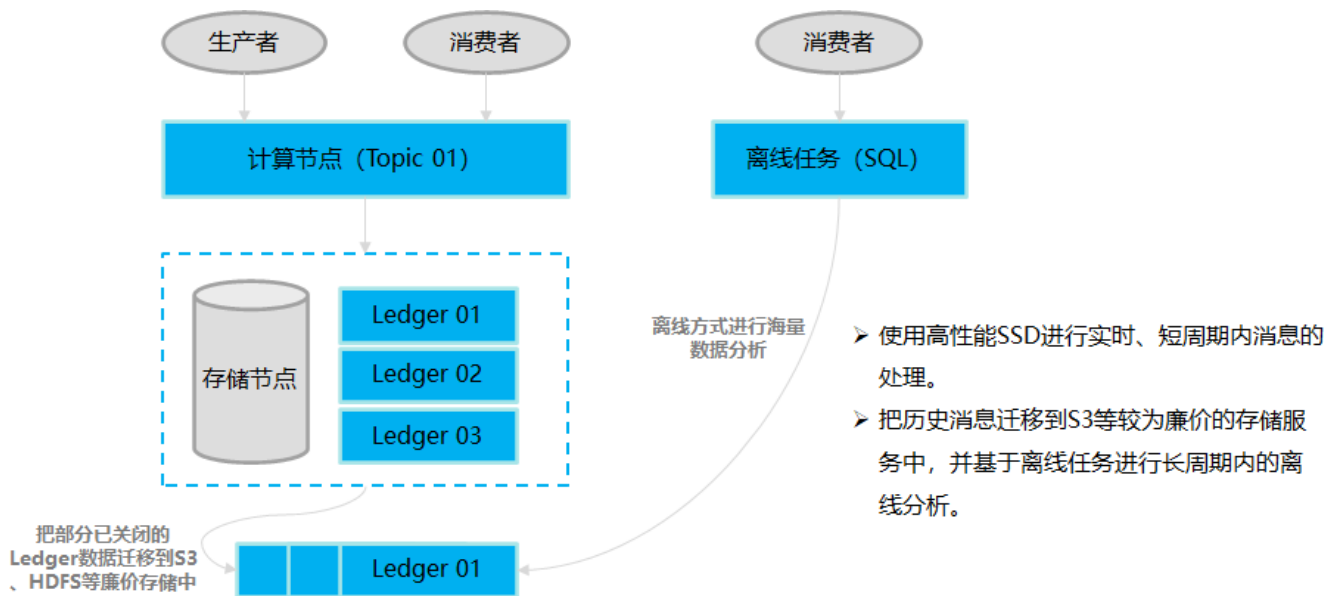
消息推送中涉及到优先级，有些紧急的消息需要优先于其他消息推送给用户。我们可以预先设置好几个优先级，每个优先级包含一类主题（比如用特定前缀或者给定范围），同时给每个优先级分配一定的读取配额。高优先级任务发布消息给高优先级的主题，低优先级任务发布消息给低优先级的主题，这样消费者可以按优先级顺序依次一定读取一定数量的消息。如果三个优先级按照4:2:1的比例分配读取配额，那就是高优先级先读取4条，中优先级读取2条，低优先级读取1条，从而实现基于优先级的消息推送功能。

2.5.5 跨地域数据复制



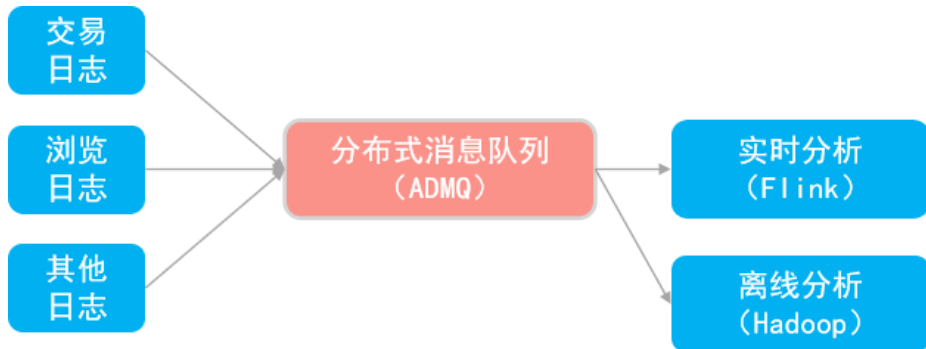
对于某些数据敏感的系统，通常需要在多个城市建立数据中心保存系统产生的数据，避免一个城市因自然灾害等原因出现整个数据中心都不能访问时，业务能切换到其他城市的数据中心，继续给用户提供服务。ADMQ内置了数据跨集群复制的功能，我们可以在北上广三地分别部署ADMQ集群，当一个地区的集群收到消息后，会通过异步的方式复制到其他地区的集群上，这样当广州集群出现问题后，业务系统仍能够通过北京和上海的集群读取数据。

2.5.6 分级存储



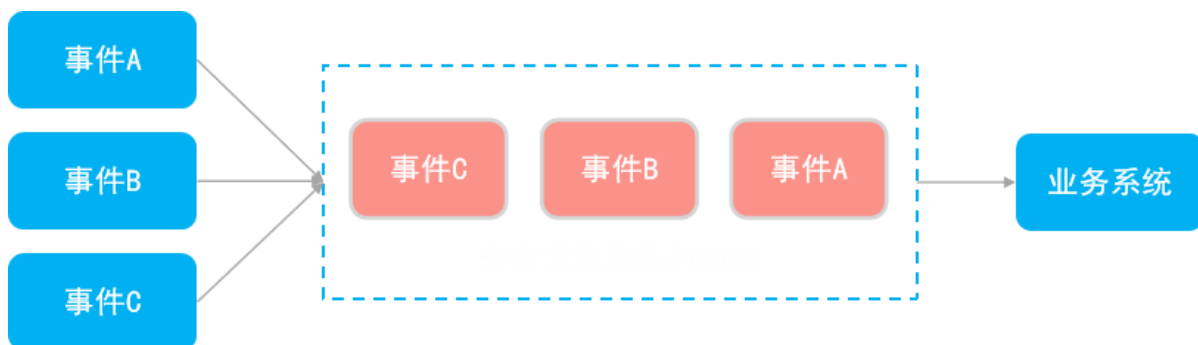
消息的分发一般对时效性要求比较高，所以不需要把所有的历史消息都存在本地磁盘。生产者发送的数据通过Broker发送到Bookie存储，Bookie把接收到的数据存储到文件里，我们可以把文件和ledger对应起来，就是每一个ledger对应一个文件，随着接收到的消息越来越多，文件大小也会主逐渐变大，当达到一定大小时就会生成一个新的文件存储数据，把之前的文件关掉。然后我们就可以把这些已经关闭的文件放到HDFS或者云存储环境里，只把当前使用的文件放到SSD盘上，这样就能更充分的利用SSD的读写优势。对于迁移到其他存储的数据，我们也可以通过其他方式读取里面的数据。

2.5.7 数据分析



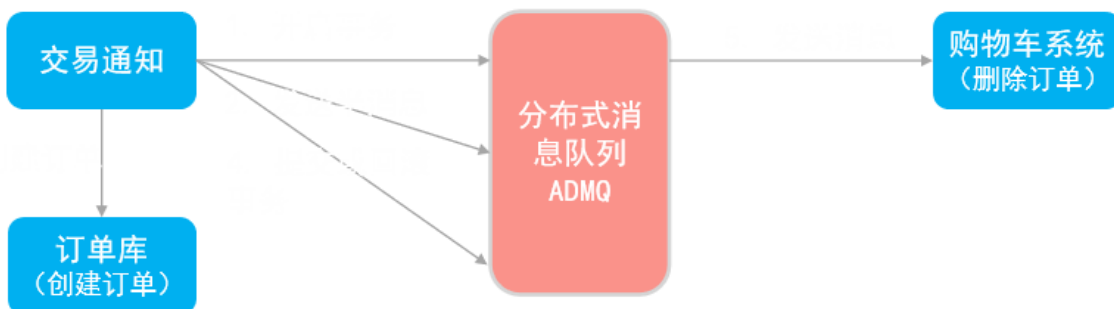
数据在“流动”中产生价值，传统数据分析大多是基于批量计算模型，而无法做到实时的数据分析，利用 ADMQ 与流式计算引擎相结合，可以很方便地实现业务数据的实时分析。

2.5.8 顺序消息



消息的顺序性有很多的业务场景，比如证券交易过程时间优先原则，交易系统中的订单创建、支付、退款等流程，航班中的旅客登机消息处理等等。与 FIFO 原理类似，ADMQ 提供的顺序消息即保证消息的先进先出。

2.5.9 分布式事务



将核心链路业务与可异步化处理的分支链路进行拆分，将大事务拆分成小事务，减少系统间的交互，既高效又可靠；ADMQ 的可靠传输与多副本技术能确保消息不丢失，At-Least-Once 特性确保数据的最终一致性。

2.6 相关概念

2.6.1 消息

系统的基本数据单元，生产者发送消息到ADMQ，ADMQ把消息转发给消费者。

2.6.2 主题

消息转发通道，标识一组消息，区分消息的不同类型，是个逻辑概念。生产者发送消息时必须指定主题，消费者接收消息时也必须指定主题。

2.6.3 计算节点

和生产者、消费者直接对接的程序，提供一个消息接收和发送服务。

2.6.4 存储节点

存储ADMQ接收到的消息内容到磁盘中。

2.6.5 发布订阅

一种消息转发模型，生产者发送消息到ADMQ后，多个消费者可以通过不同订阅组分别获取全量消息。

3 产品安装

3.1 基本概念

在正确使用应用服务器来部署、管理应用之前，需要先理解以下几个基本概念：

3.1.1 ADMQ 集群

一个ADMQ实例由一个或多个ADMQ集群组成。集群又由以下部分组成。

- 一个或者多个brokers
- 一个或者多个ZooKeeper协调器，用于集群级别的配置和协调
- 一组BookKeeper的Bookies用于消息的持久化存储
- 一个管理控制台，用于监控管理用户和相关资源的创建和分配，也是broker和bookkeeper启动的依赖。

3.1.2 Broker

broker是一个无状态组件，主要负责运行另外的两个组件：

- 一个 HTTP 服务器，它暴露了 REST 系统管理接口以及在生产者 and 消费者之间进行 Topic 查找的API。
- 一个调度分发器，它是异步的TCP服务器，通过自定义二进制协议应用于所有相关的数据传输。
- 为了支持全局Topic异地复制，Broker会控制Replicators追踪本地发布的条目，并把这些条目用Java 客户端重新发布到其他区域。

3.1.3 元数据存储

ADMQ使用Zookeeper进行元数据存储、集群配置和协调。在一个ADMQ实例中：

- 配置与仲裁存储：存储租户，命名域和其他需要全局一致的配置项
- 每个集群有自己独立的ZooKeeper保存集群内部配置和协调信息，例如归属信息，broker负载报告等等。

3.1.4 持久化存储

ADMQ为应用程序提供有保障的消息传递。如果一个消息成功地到达ADMQ Broker，它将被送达其预定的目标。为了提供这种保证，未确认送达的消息需要持久化存储直到它们被确认送达。ADMQ用BookKeeper作为持久化存储。BookKeeper是一个分布式的预写日志（WAL）系统。

3.1.5 管理控制台

管理控制台是一个基于Web的GUI管理和监控工具，用于管理用户、权限、租户、命名空间、主题、订阅、broker、部署集群、插件，并支持多个环境的动态配置。

下表列出了金蝶Apusic分布式消息队列V2.0.5的默认管理值和各端口的默认值。

端口	对应组件	作用	访问范围
12306	adm-q-manager	管控台访问端口	外部访问
12305	adm-q-manager	license认证端口	内部访问
8080	broker	admin管理端口	外部访问
6650	broker	数据传输端口	外部访问
9092	broker	kafka插件服务端口	外部访问
5672	broker	rabbtmq单节点服务端口	外部访问
6672	broker	rabbtmq集群代理服务端口	外部访问

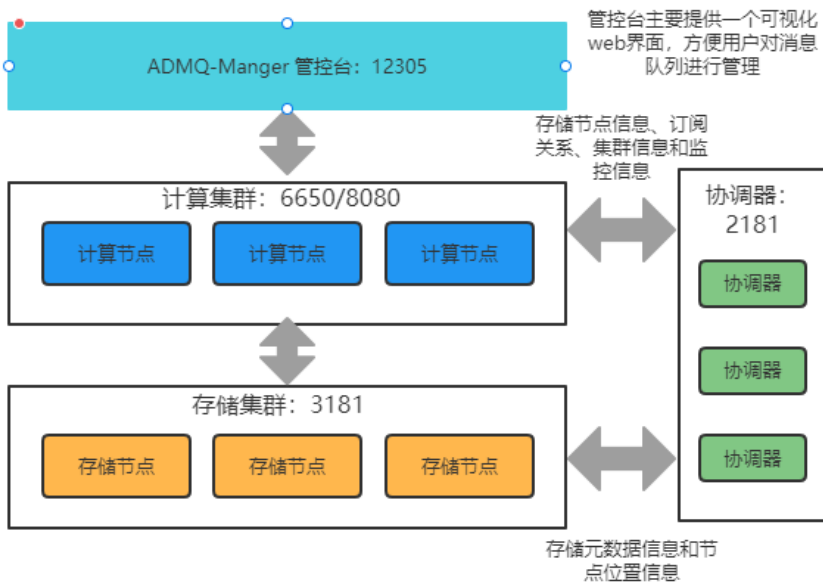
15673	broker	rabbitmqHTTP接口端口	外部访问
9876	broker	rocketmq插件服务端口	外部访问
1888	broker	mqtt单节点插件服务端口	外部访问
5682	broker	mqtt集群代理插件服务端口	外部访问
3181	storage	数据存储端口	内部访问
8000	storage	监控数据采集端口	内部访问
2181	zookeeper	zookeeper服务端口	内部访问
2888	zookeeper	zookeeper服务端口	内部访问
3888	zookeeper	zookeeper服务端口	内部访问
9990	zookeeper	zookeeper admin管理端口	内部访问
18800	zookeeper	监控数据采集端口	内部访问

如果使用tls, 需要开放 6651, 8081。

3.2 产品部署说明

3.2.1 部署架构图

ADMQ集群安装至少需要四台服务器, 1管控+3节点 (计算、存储、协调器)



3.2.2 部署硬件需求(单机最低配置)

用途	硬件配置	软件配置
ADMQ节点	CPU: 8核 内存: 16G 硬盘HDD: 500G	ADMQ v2.0.5, JDK8

3.2.3 部署硬件需求(集群最低配置)

用途	硬件配置	软件配置
----	------	------

ADMQ管控台	CPU: 4核 内存: 8G 硬盘HDD: 500G	ADMQ v2.4.14 manager, JDK17
ADMQ节点1	CPU: 8核 内存: 16G 硬盘HDD: 500G	ADMQ v2.4.5, JDK8
ADMQ节点2	CPU: 8核 内存: 16G 硬盘HDD: 500G	ADMQ v2.4.5, JDK8
ADMQ节点3	CPU: 8核 内存: 16G 硬盘HDD: 500G	ADMQ v2.4.5, JDK8

3.2.4 部署硬件需求(推荐集群配置)

用途	硬件配置	软件配置
ADMQ管控台	CPU: 8核 内存: 16G 硬盘SSD: 500G	ADMQ v2.4.14 manager, JDK17
ADMQ节点1	CPU: 8核 内存: 32G 硬盘SSD: 500G	ADMQ v2.4.5, JDK8
ADMQ节点2	CPU: 8核 内存: 32G 硬盘SSD: 500G	ADMQ v2.4.5, JDK8
ADMQ节点3	CPU: 8核 内存: 32G 硬盘SSD: 500G	ADMQ v2.4.5, JDK8

3.2.5 安装介质说明

安装介质ADMQ-V2.0.5-20241230.tar.gz, 解压后包含

- admq-manager-V2.4.14.tar.gz: ADMQ管控台安装包
- admq-V2.4.5-all.tar.gz: ADMQ核心包 (包括协调器、计算节点、存储节点)

3.3 授权模式说明

ADMQ 包含管理控制台和引擎节点, 引擎节点分为计算、存储和协调器三种类型。授权文件只限制部署的计算和存储两种类型的节点数量, 管理控制台本身不占用授权节点数。

- 授权方式

每个授权文件都包含一个站点的授权, 每个站点包含两个节点。即每个授权文件都可以部署两个节点, 不限制节点类型。比如用户可选择部署 1 个计算节点和 1 个存储节点, 或者部署 2 个计算节点, 或者部署 2 个存储节点。

- 部署说明

单机部署时占用 2 个节点数, 即 1 个计算节点和 1 个存储节点。

集群部署时通常需要占用 6 个节点数 (3 个站点的授权), 即 3 个计算节点和 3 个存储节点, 协调器不占用节点数。

3.4 安装管理控制台

3.4.1 安装前准备

- 获取安装包: 从 <http://www.apusic.com/下载金蝶> Apusic 分布式消息队列 V2.0.5 安装包, 或从金蝶Apusic 分布式消息队列 V2.0.5 产品光盘中获得相应的安装包文件。
- 系统要求: Java环境 (管控台JDK17以上, 节点JDK1.8及以上版本)、内存 (16GB+)、硬盘空间 (100GB+)、浏览器 (IE8及以上, FireFox, Chrome)

3.4.2 开始安装

3.4.2.1 管理控制台安装

解压安装包

```
mv ADMQ-V2.4.14.tar.gz /opt/
cd /opt
tar -zxvf admq-manager-V2.4.14.tar.gz
```

修改数据库类型 (可选)

默认使用H2数据库，如果正式环境部署需要换成其他数据库，例如MySQL。

/opt/admq-manager-V2.4.14/config/application.properties 文件中添加了多种数据库配置，可以把使用的数据库配置打开，注释掉其他不使用的数据库配置项。

1. 注释掉H2数据库相关配置，打开mysql的相关配置

```
# 数据库配置 (多选一)
#spring.config.import.03=file:./config/other-config/db-h2.properties
spring.config.import.03=file:./config/other-config/db-mysql.properties
#spring.config.import.03=file:./config/other-config/db-kingbase.properties
#spring.config.import.03=file:./config/other-config/db-oscar.properties
#spring.config.import.03=file:./config/other-config/db-dm.properties
#spring.config.import.03=file:./config/other-config/db-pgsql.properties
```

2. 配置MYSQL的地址、端口、数据库名称、用户名、密码，路径如上图 config/other-config/db-mysql.properties

```
# mysql
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://ip:port?useUnicode=true&characterEncoding=utf-8&allowMultiQueries=true&useSSL=false
spring.sql.init.schema-locations=classpath:/META-INF/sql/mysql-schema.sql
spring.datasource.username=root
spring.datasource.password=root
```

3. 管控台启动的时候会自动加载数据库初始化脚本

其他数据库参照此步骤即可

3.4.2.2 启动管理控制台服务

```
cd /opt/admq-manager/
bin/admq-manager start
```

3.4.2.3 停止管理控制台服务

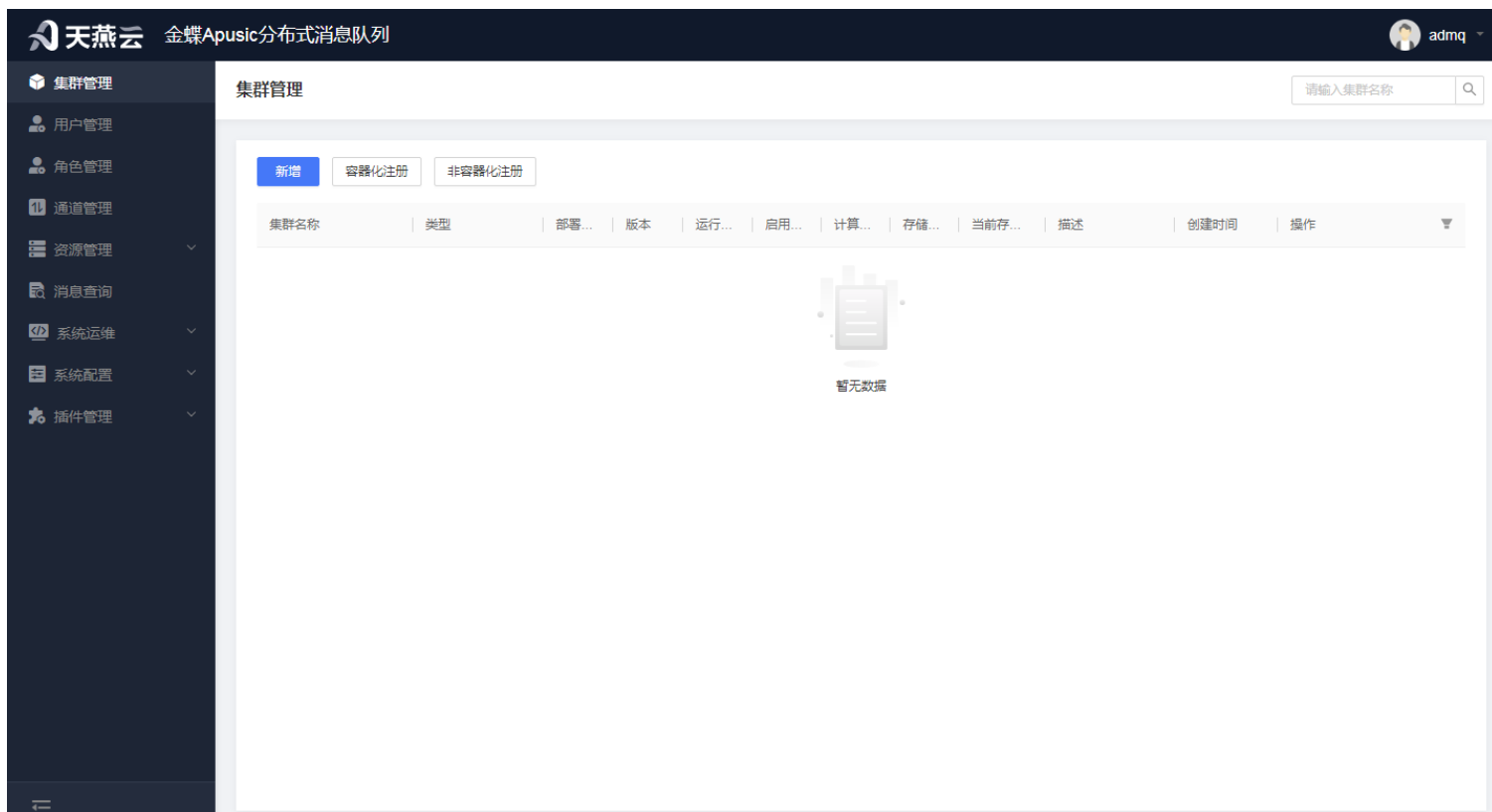
```
cd /opt/admq-manager/
bin/admq-manager stop
```

3.4.3 验证

- 1.在浏览器中键入URL: <https://ip:12306> 或者<http://ip:12305>
- 2.在登录界面输入用户名 (admq) ,密码(11111111)进行登录，首次登录需要修改密码。



3.若登录成功，会跳转到ADMQ管控台首页，则表示安装成功。



3.5 通过管控台安装ADMQ

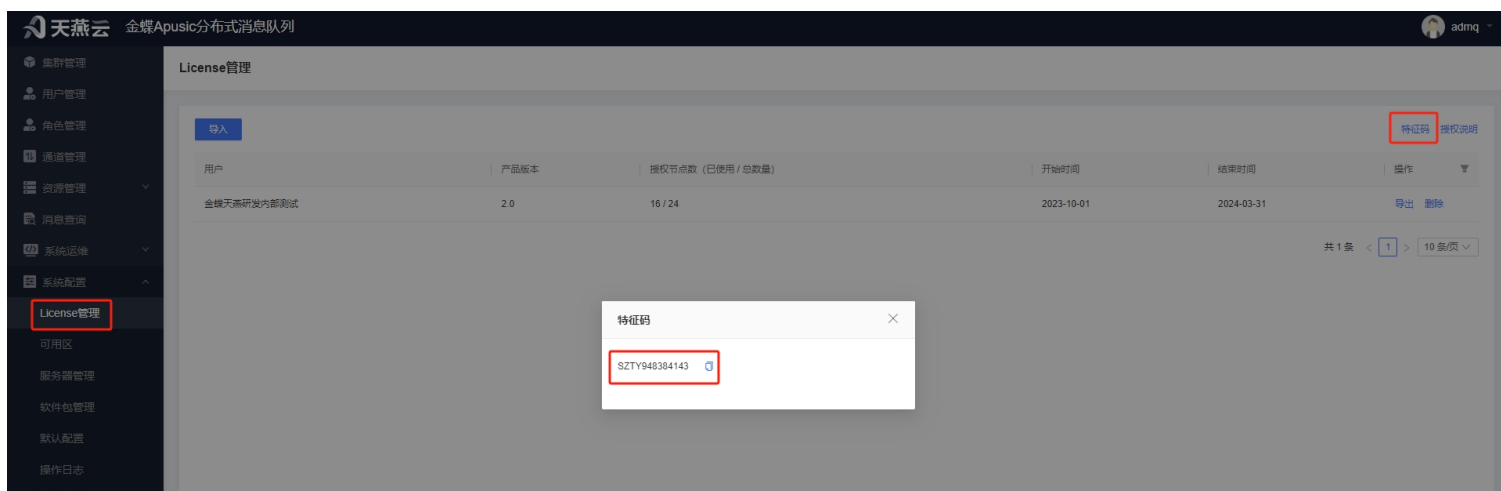
3.5.1 安装前准备

3.5.1.1 管控台导入授权

管控台目前支持两种授权模式：KBC模式、授权中心模式，可选择其中一种授权模式进行授权。

1、KBC模式

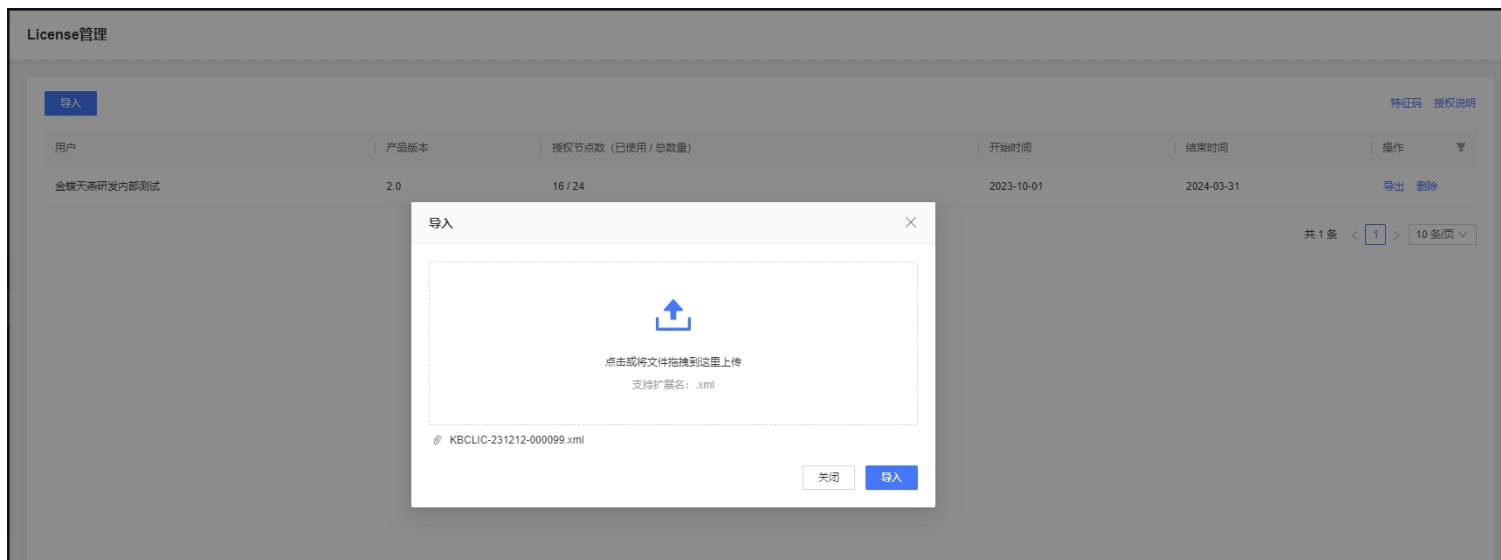
获取管控台特征码



用特征码去KBC系统激活和下载许可文件

将下载的许可文件改成xml后缀导入到管控台中

名称	修改日期	类型	大小
KBCLIC-231212-000099.lic	2023/12/22 17:53	LIC 文件	3 KB
KBCLIC-231212-000099.xml	2023/12/22 17:53	XML 源文件	3 KB



2、授权中心模式

在管控台安装目录vi config/application.properties

```
#是否开启授权中心
apusic_acls_enable=false
#授权中心地址 geip1:port1, ip2:port2
apusic_acls_authUrls=
#租户, 可选
apusic_acls_tenant=
#命名空间, 可选
apusic_acls_ns=
```

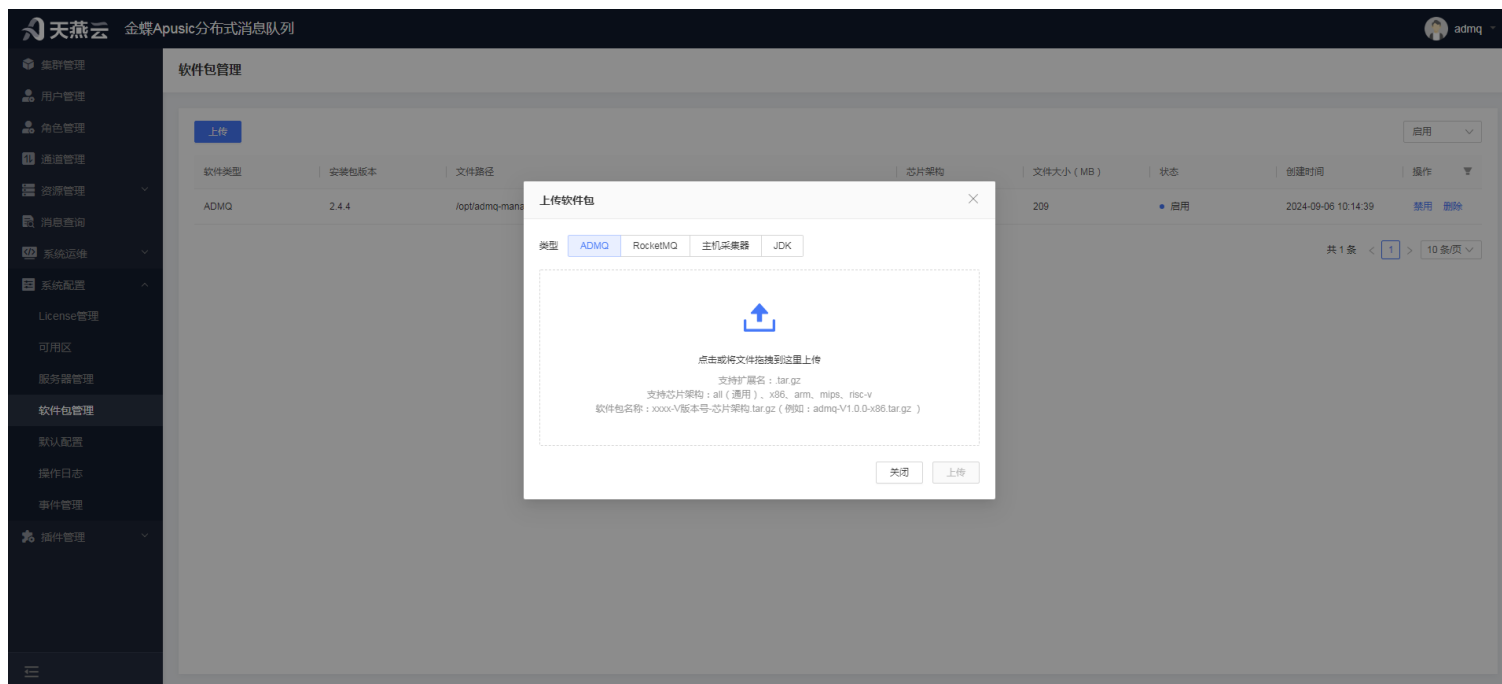
```
apusic_acls_enable=true
apusic_acls_authUrls=ip:port
apusic_acls_tenant=public
apusic_acls_ns=XXXX有限公司
```

重启管控台即可连上授权中心

3.5.1.2 上传软件包

部署集群前，需要引用admq软件包信息，需要上传软件包信息。

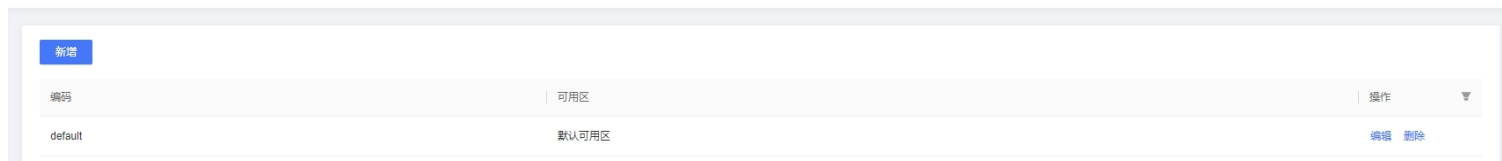
进入系统配置->软件包管理->上传，将获取的admq-V2.4.5-all.tar.gz软件包或者rocketmq-V4.9.2-all.tar.gz软件包上传。



3.5.1.3 新增可用区 (可选)

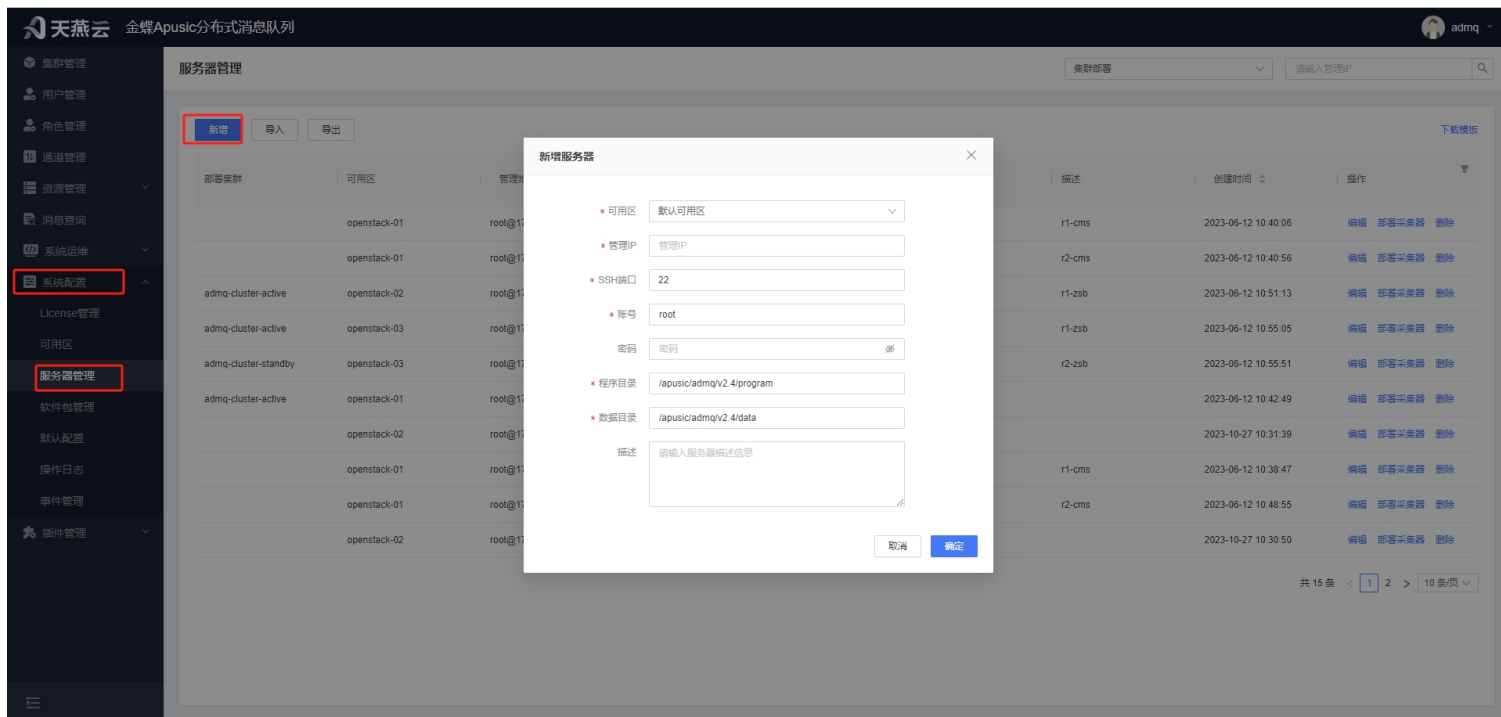
系统默认会初始化一个默认可用区，如果有多个数据中心可以根据需要添加。

可用区



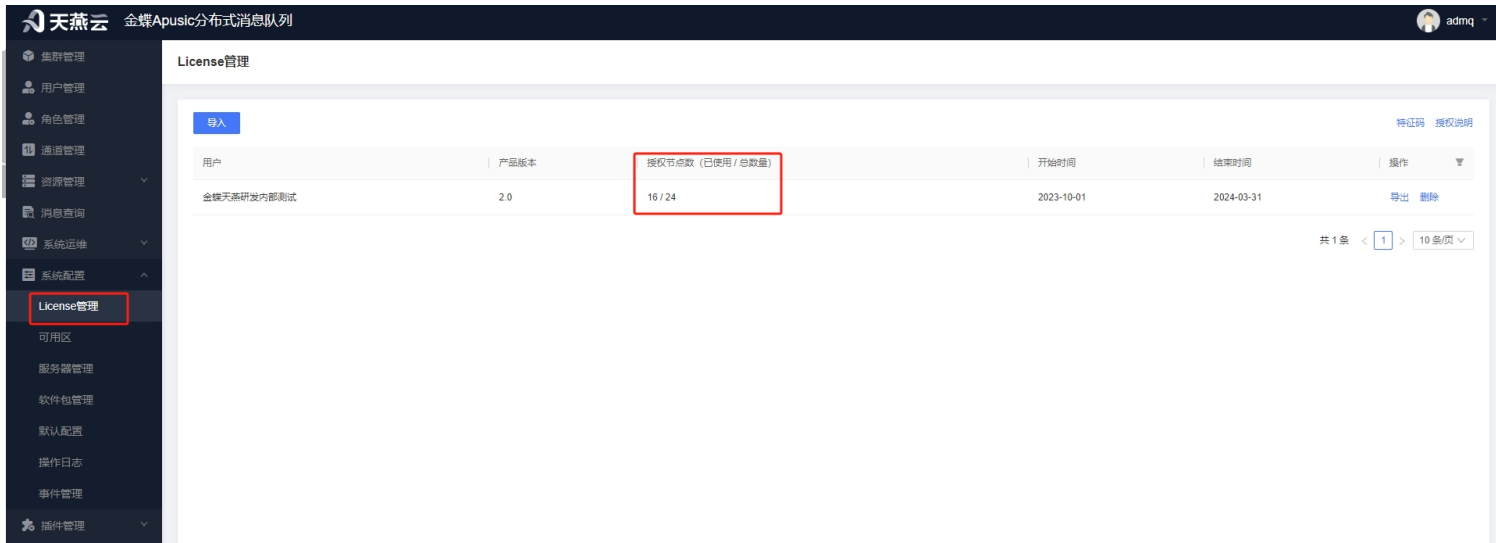
3.5.1.4 新增服务器信息

部署集群环境之时，需要选择计算程序地址、存储程序地址、本地协调器地址，所以需要新增服务器信息，待新建集群信息引用。部署单机环境时，只需要创建一个服务器信息即可，若需要创建集群环境，至少需要新建3个服务器信息作为集群的远程节点服务器（计算程序地址、存储程序地址、本地协调器地址）。



3.5.1.5 确认License中允许的节点数量

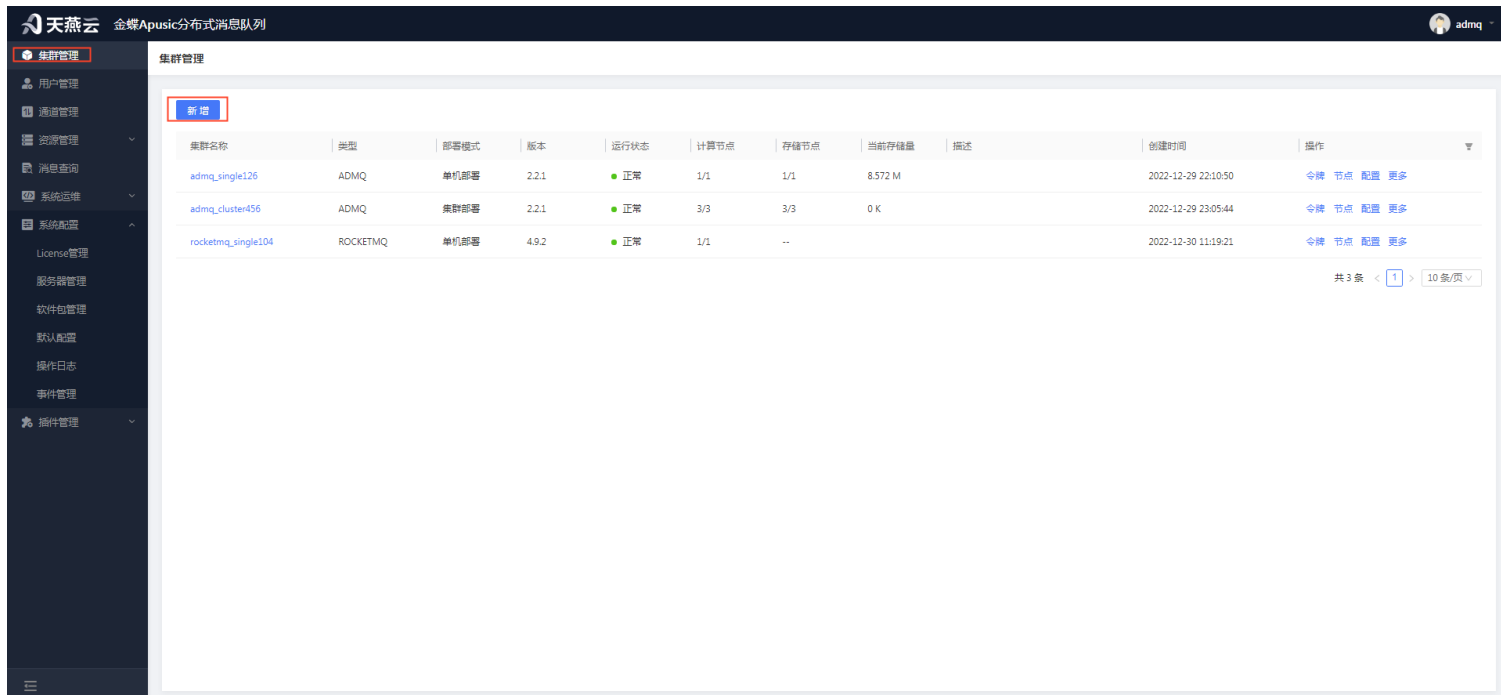
集群服务器节点启动前需要license进行授权，若在集群启动的节点数量超过License允许的最大计算节点数量和最大存储节点数量，则启动集群节点起不来，需要再上传license（上传license后，最大存储节点数与最大计算节点数会进行累加）。若需要部署单机环境，最大节点数和最大存储节点数需要 $>=1$ 。若需要部署集群环境，最大节点数和最大存储节点数需要 $>=3$ 。（具体数量需要根据节点数量来确认）



3.5.2 安装ADMQ

3.5.2.0.1 新增集群

1.进入集群管理->新增，即可进入安装页面



2.输入集群名称。

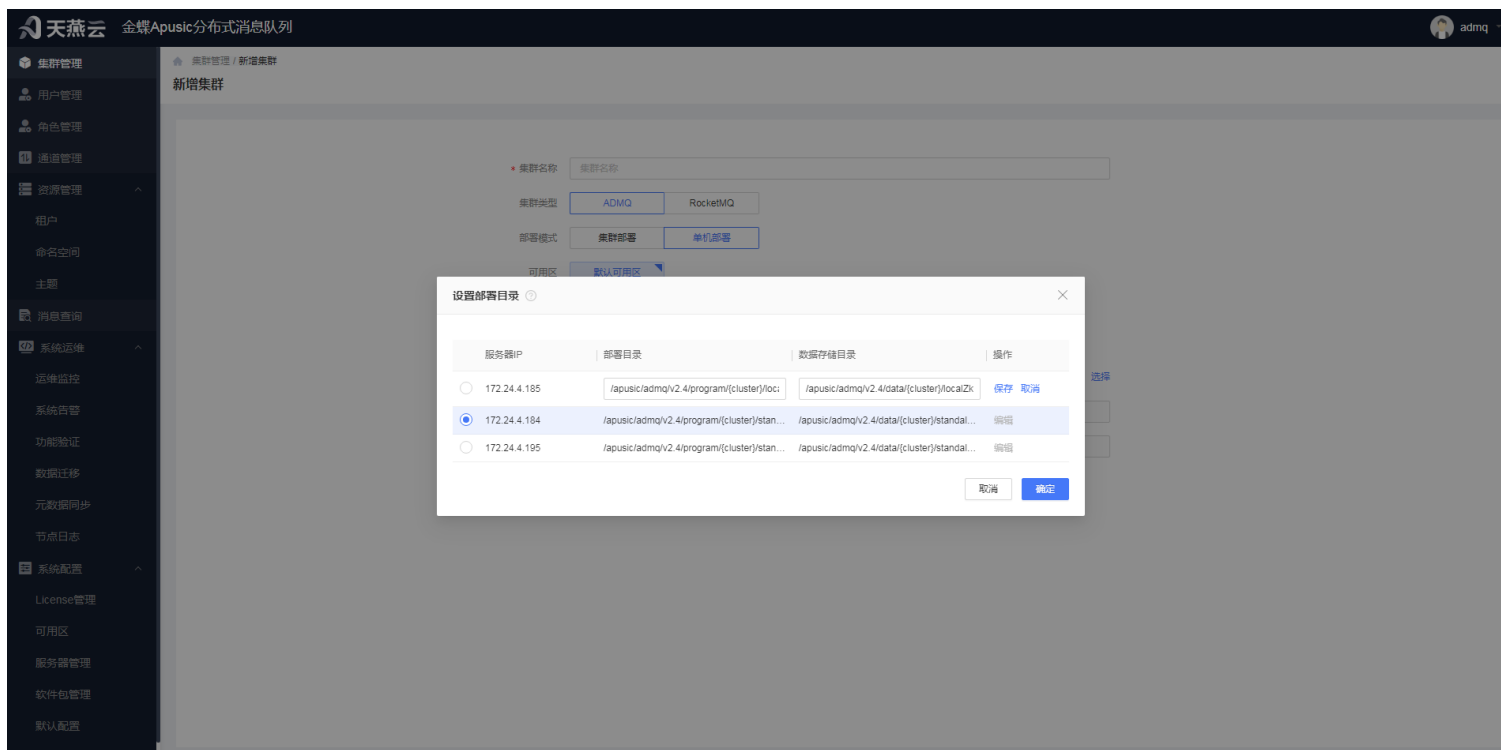
3.选择部署模式，目前支持单机部署和集群部署两种模式。

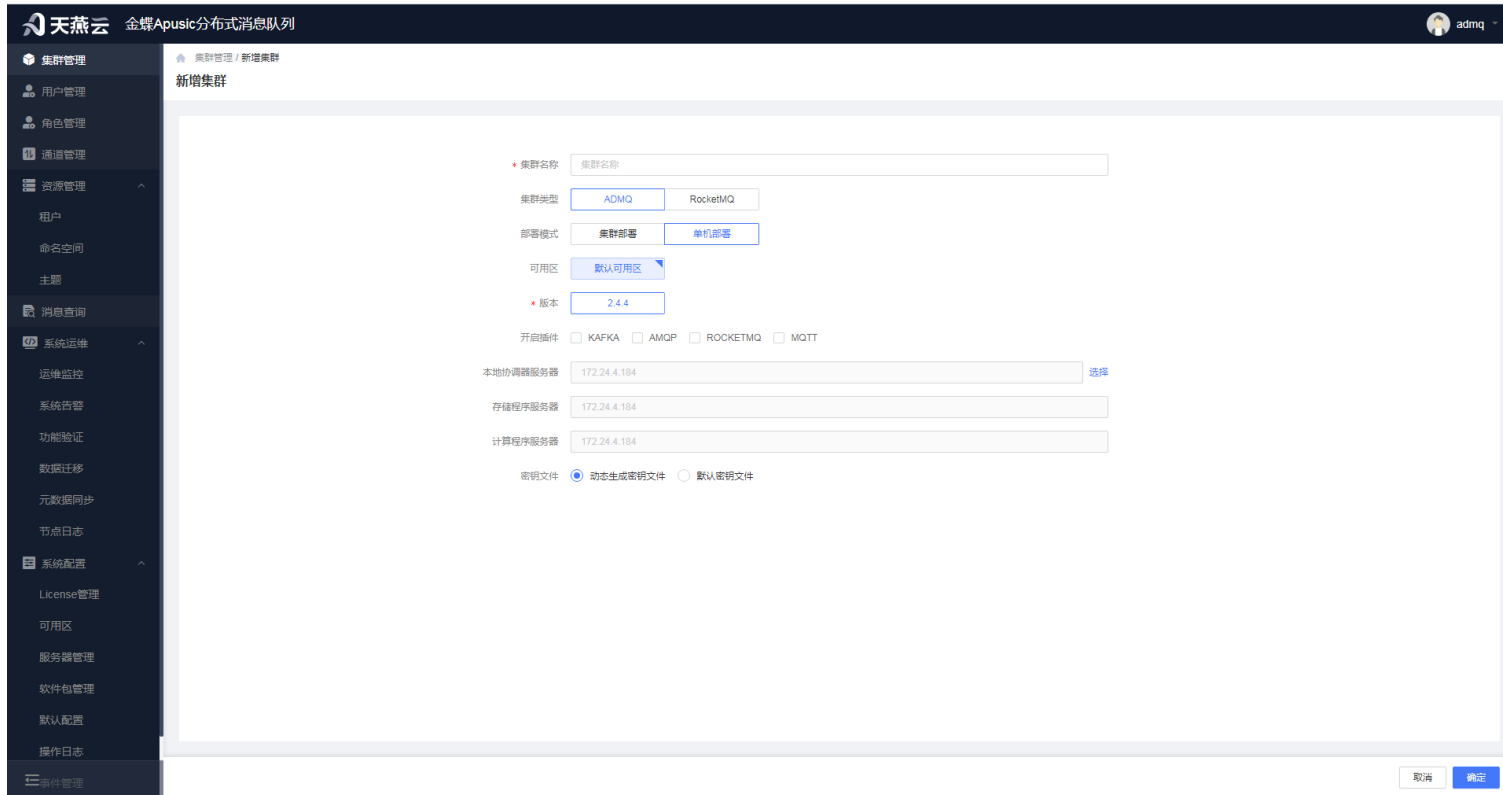
4.选择可用区。

5.勾选需要开启的插件。

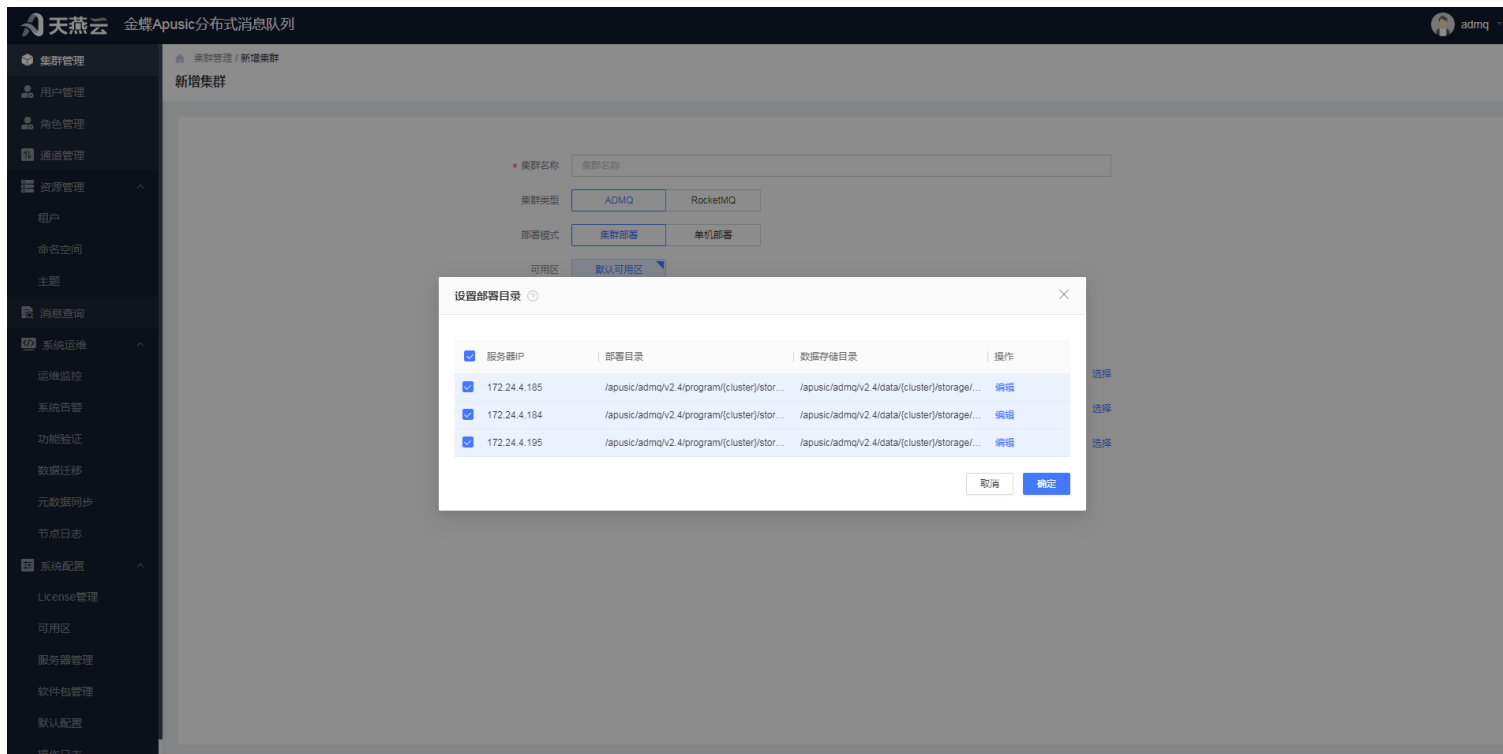
6.设置协调器、存储程序、计算程序部署服务器、部署目录、数据存储目录，其中部署目录、数据存储目录建议使用默认配置。

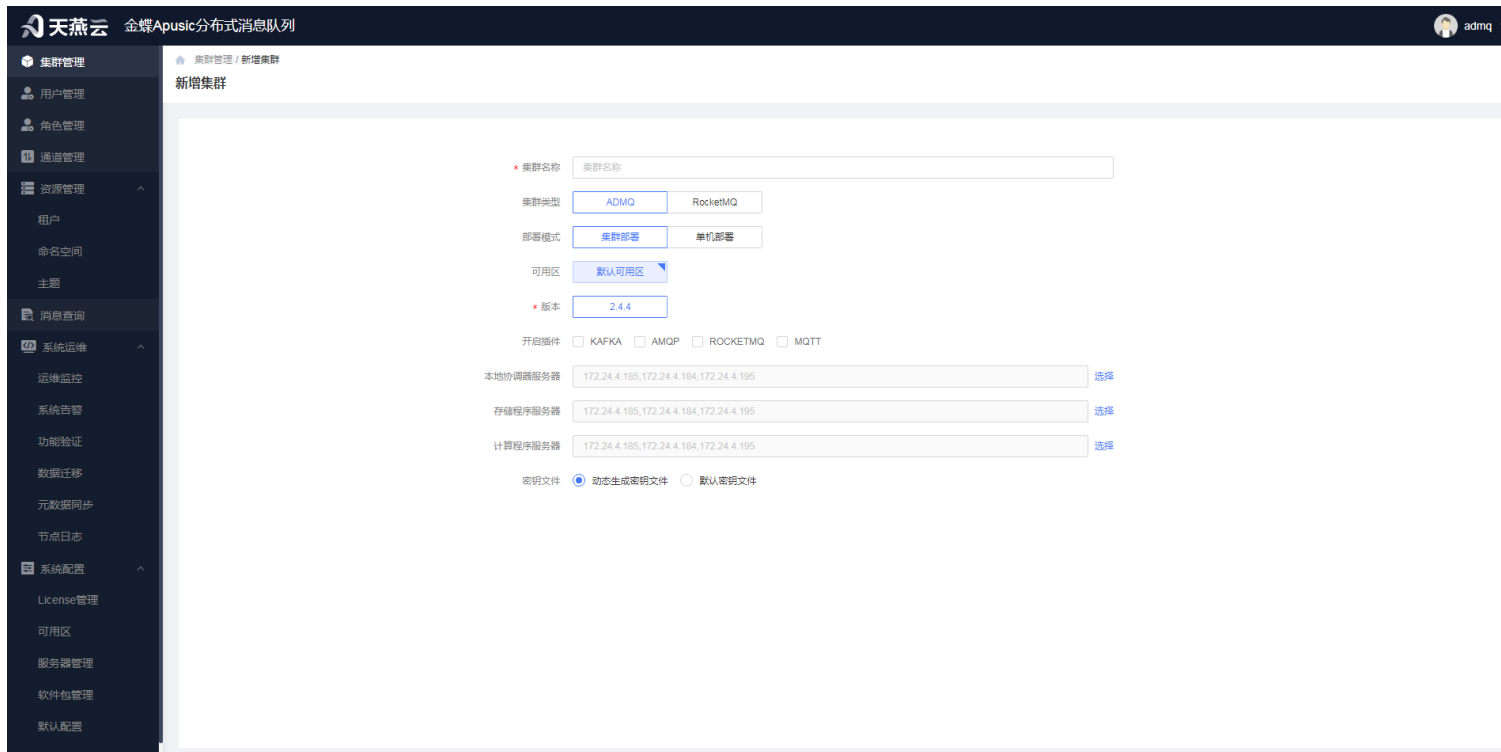
- 单机部署时，选择一台服务器即可。





- 集群部署时，每个组件都要进行服务器选择，且至少选择3台服务器。





6. 密钥文件默认动态生成密钥文件，点击【确定】按钮，即可新增集群。

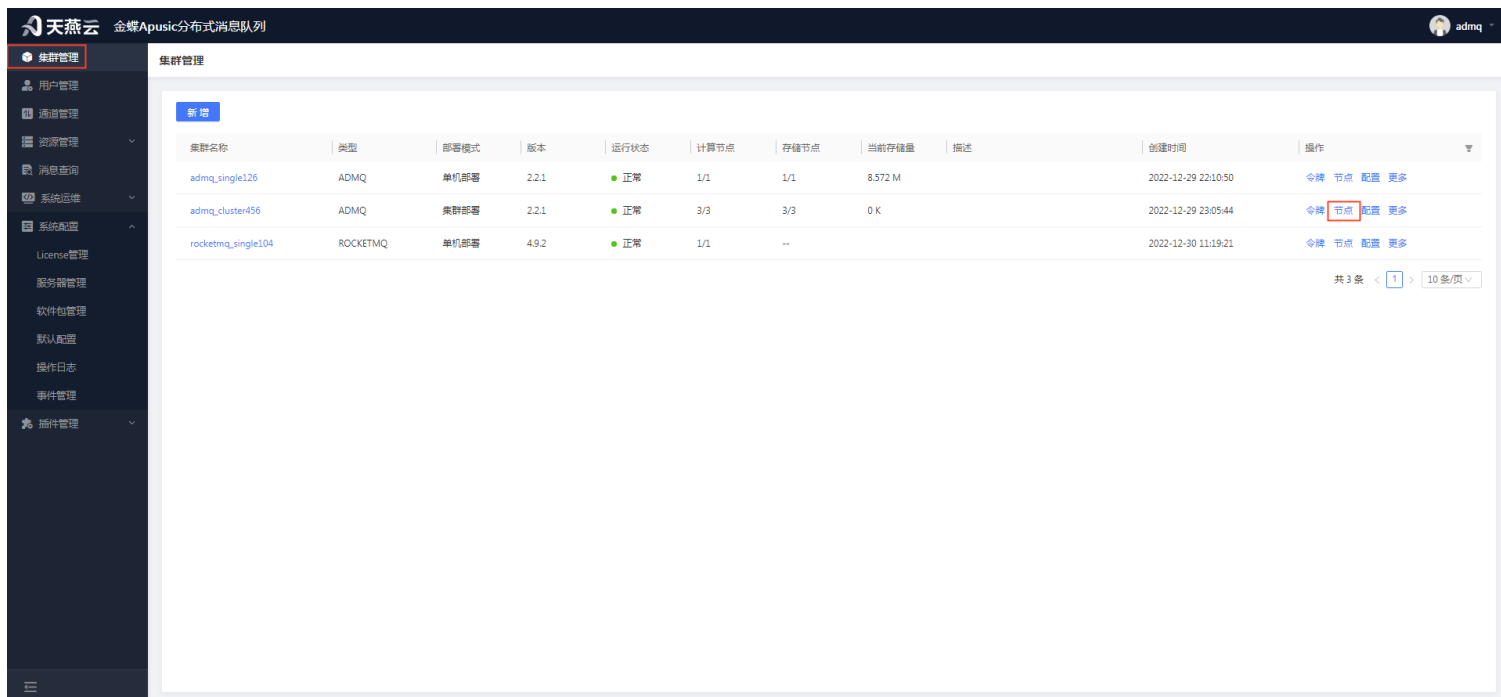
3.5.2.0.2 部署集群

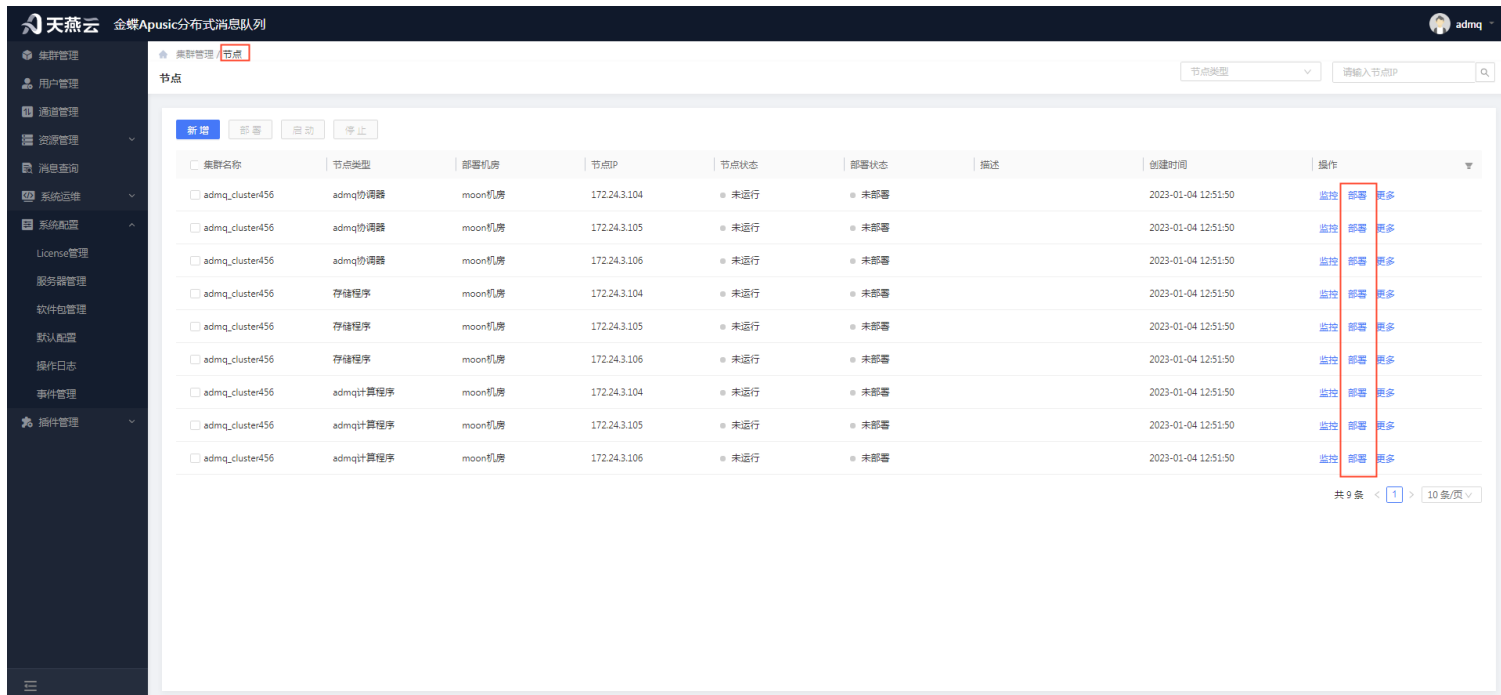
部署前检查：

集群部署会使用默认配置文件中一些默认端口，部署前需要开放某些端口，管控台不负责处理，由用户自己处理：存储程序:2181,2888,3888 全局协调器：2184,2889, 3889 :3181,8000, 计算程序:6650,8080,如果使用tls，需要开放6651,8081。

先查看防火墙是否开启：systemctl status firewalld 如果没开启，要先开启：systemctl start firewalld 查看端口是否对外开放：firewall-cmd --query-port=端口号/tcp 如果没开放，就执行：firewall-cmd --add-port=端口号/tcp --permanent 重新加载防火墙的端口：firewall-cmd --reload

新增集群信息后，可部署集群服务器节点。

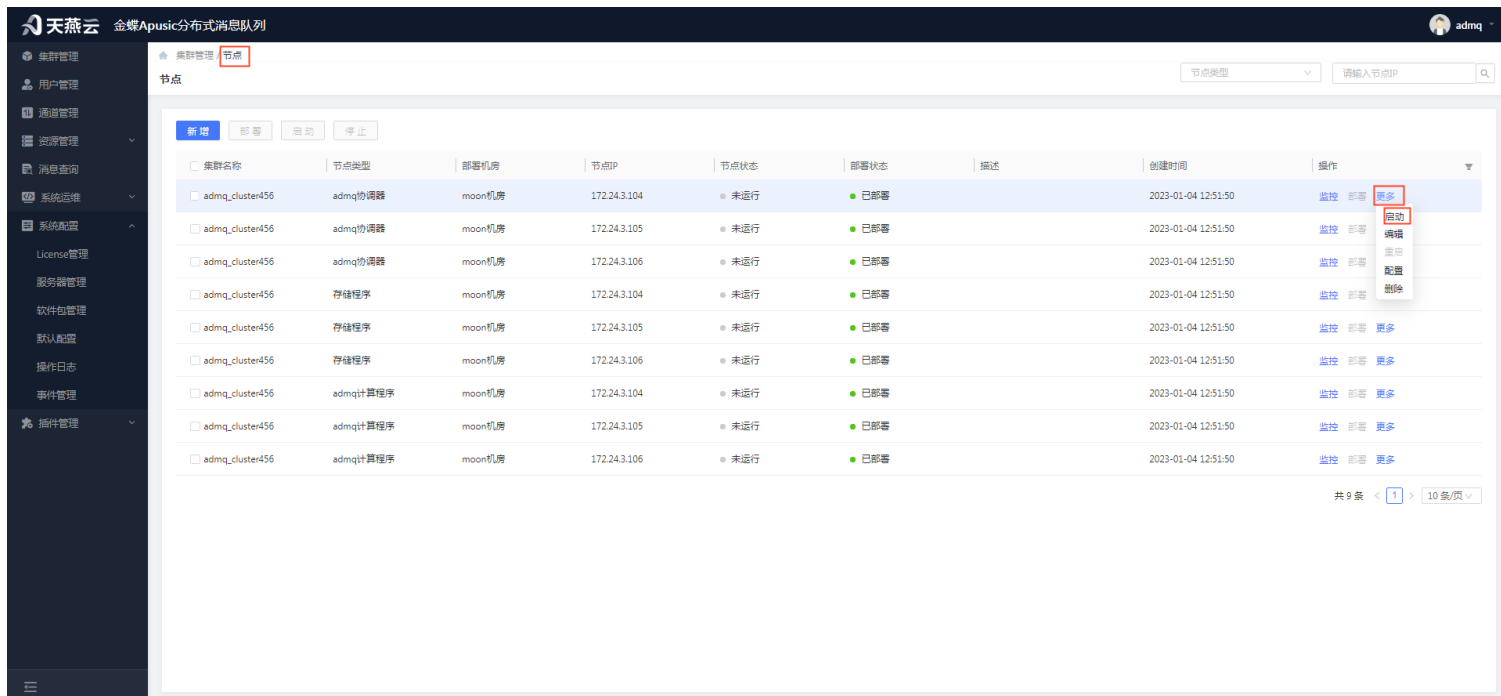




部署过程需要稍等几分钟左右。

3.5.2.0.3 启动集群节点

部署节点后，可以启动集群的各个节点。



启动所有节点，集群环境就部署成功了。

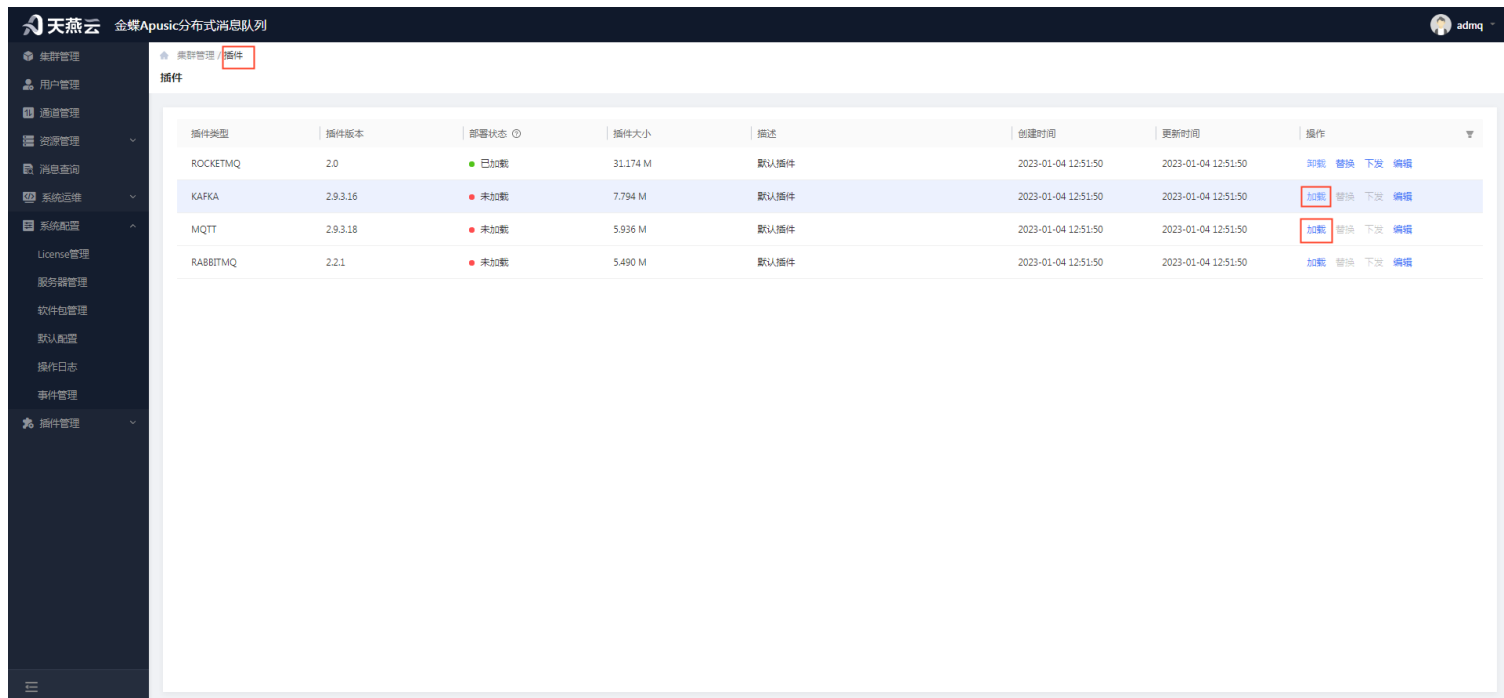
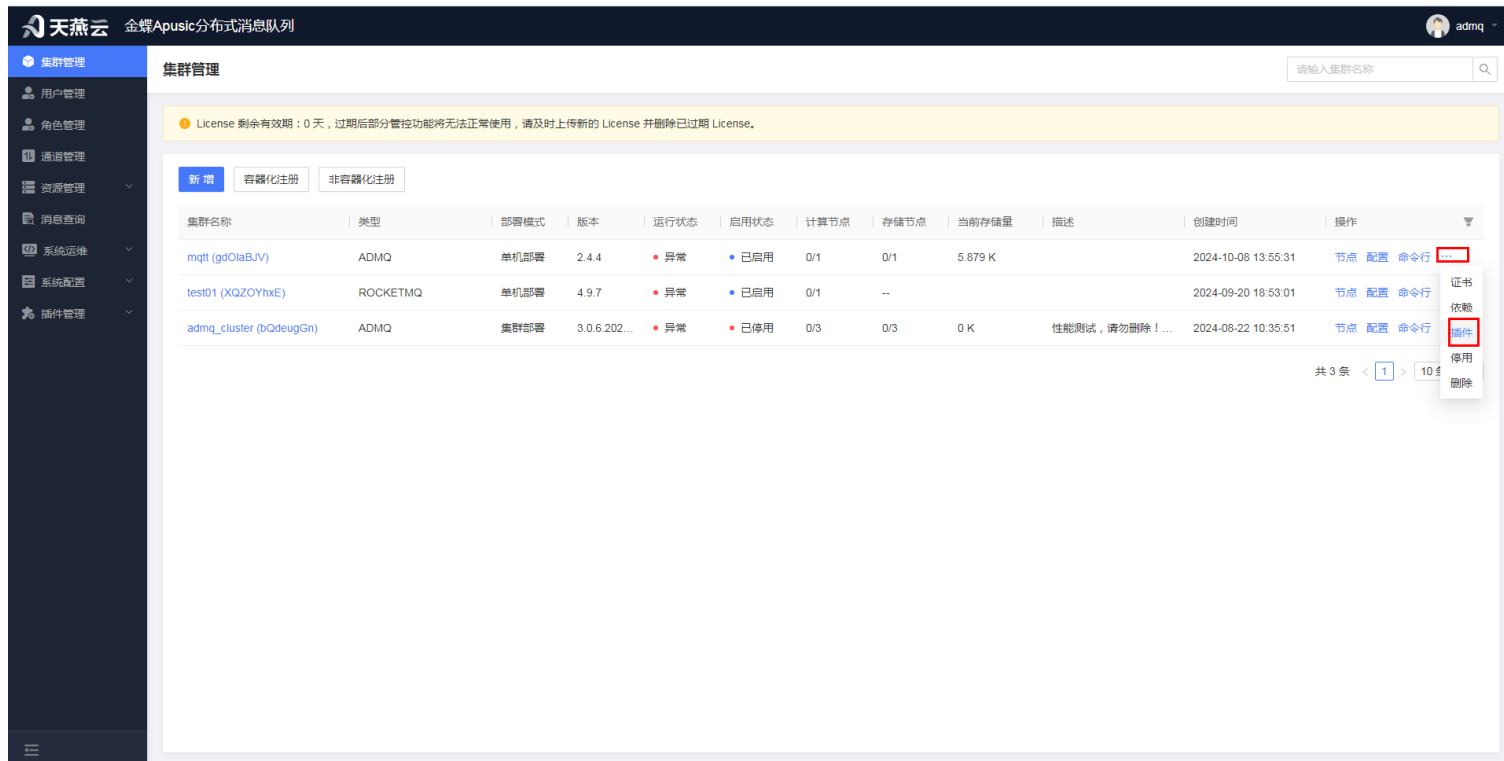
3.5.3 插件安装

ADMQ支持通过加载Kafka、RocketMQ、RabbitMQ等插件的方式适配其他客户端协议，使原先使用其他消息中间件的系统可在不修改代码的情况下快速迁移到ADMQ上。

3.5.3.1 插件加载方式安装（推荐使用）

通过管控台界面加载KOP、MOP、ROP、AOP插件，进行插件管理，包括：插件加载、卸载、替换、下发，简化插件启动、停用流程，方便用户操作。

具体步骤：集群管理->选择要加载插件的集群选择“...”按钮->插件，进入到插件管理页面后，可选择相对应的插件进行加载。注意：可同时加载部分或全部插件，加载后需要重启集群。



3.5.3.2 配置文件方式安装（不推荐使用）

需要提前确认计算节点的adm_q-V2.2.1/protocols/目录（插件协议路径）下是否有适配各个插件的文件，若没有需获取插件文件放置到该目录下。

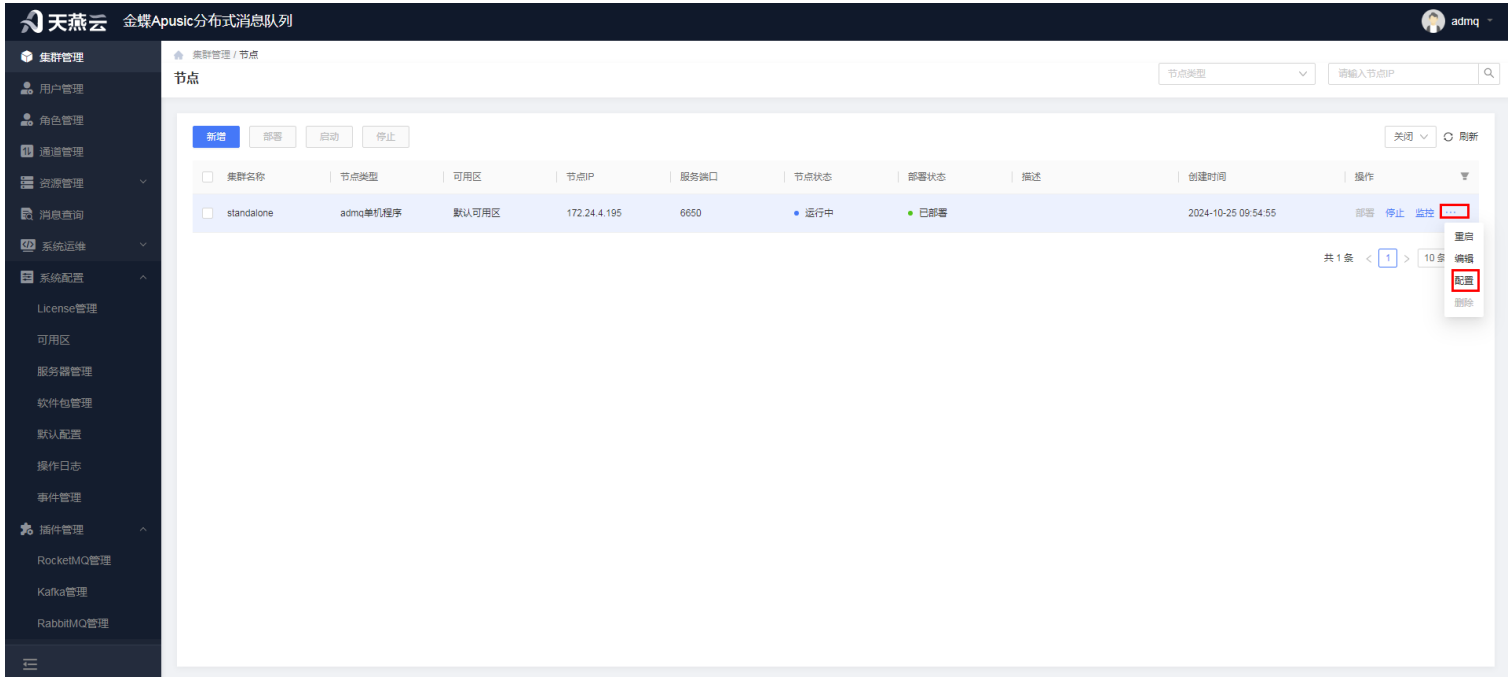
插件对应关系如下：

- Kafka插件: plugin_kafka-{version}.nar

- RocketMQ插件: plugin_rocketmq-{version}.nar
- RabbitMQ插件: plugin_amqp-{version}.nar
- MQTT插件: plugin_mqtt-{version}.nar

3.5.3.2.1 Kafka 插件

在部署完节点后，可在管控台集群节点页面修改集群每个计算节点的配置项，其他协调器和存储程序不需要修改。



点“配置”按钮进入配置信息页面后，查找如下配置项，如果有则更新，没有则添加。

```

messagingProtocols=kafka
# 插件协议路径 ({程序目录}/{集群名称}/{节点类型}/admq-V2.1.0/protocols)
# 程序目录: 添加服务器时设置的程序目录
# 集群名称: 添加集群时设置的集群名称
# 节点类型: 如果是单机部署, 则是standalone; 如果是集群部署则是broker
protocolHandlerDirectory=/home/admq/pro/apusic-mq/broker/admq-V2.2.1/protocols
narExtractionDirectory=/home/admq/pro/apusic-mq/broker/admq-V2.2.1/nar

kafkaTenant=kafka-data
kafkaMetadataTenant=kafka-meta
# 改为服务器IP
kafkaListeners=kafka://172.20.140.140:9092
kafkaProtocolMap=kafka:PLAINTEXT
# 改为服务器IP
kafkaAdvertisedListeners=kafka://172.20.140.140:9092
entryFormat=kafka
# 偏移管理
brokerEntryMetadataInterceptors=org.apache.pulsar.common.intercept.AppendIndexMetadataIntercepto
# 设置自动创建分区主题
allowAutoTopicCreationType=partitioned

kopEnableGroupLevelConsumerMetrics=true

```

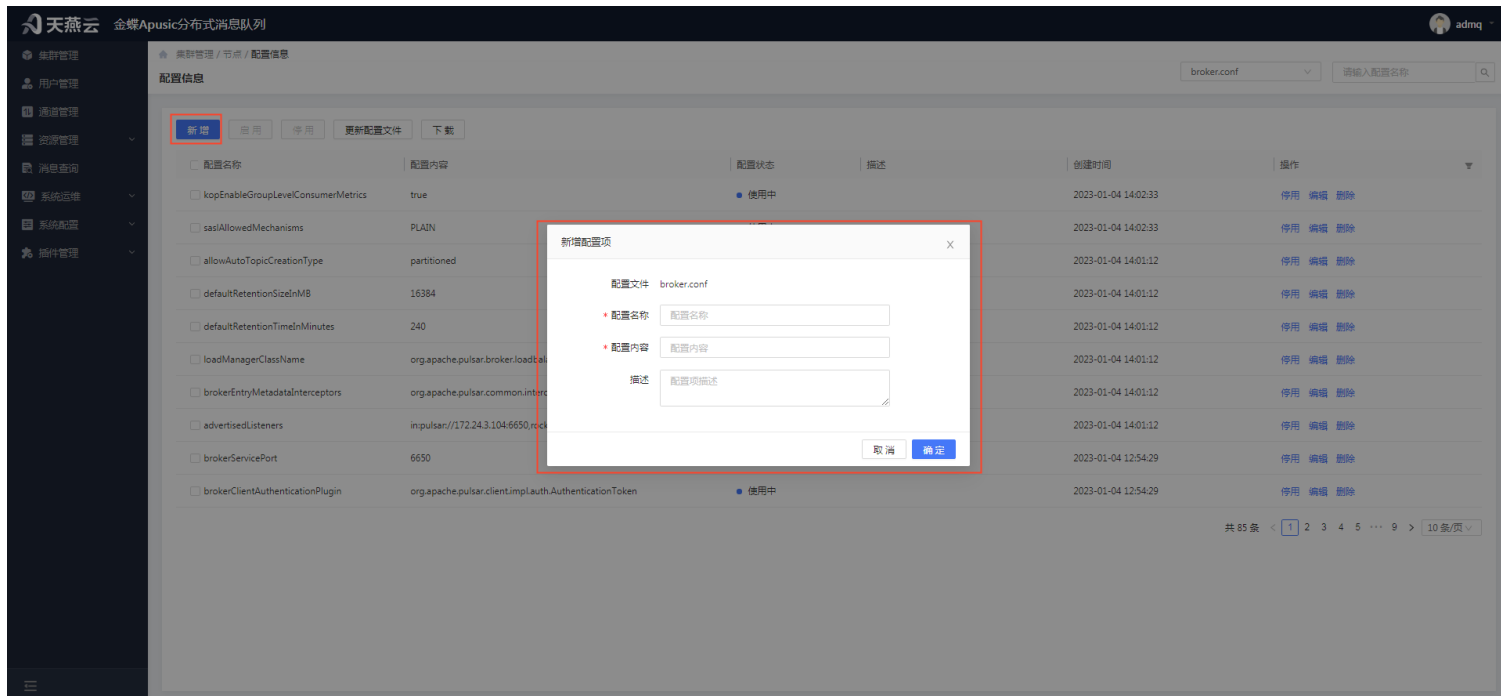
```

saslAllowedMechanisms=PLAIN
loadManagerClassName=org.apache.pulsar.broker.loadbalance.impl.ModularLoadManagerImpl
defaultRetentionTimeInMinutes=240
defaultRetentionSizeInMB=16384

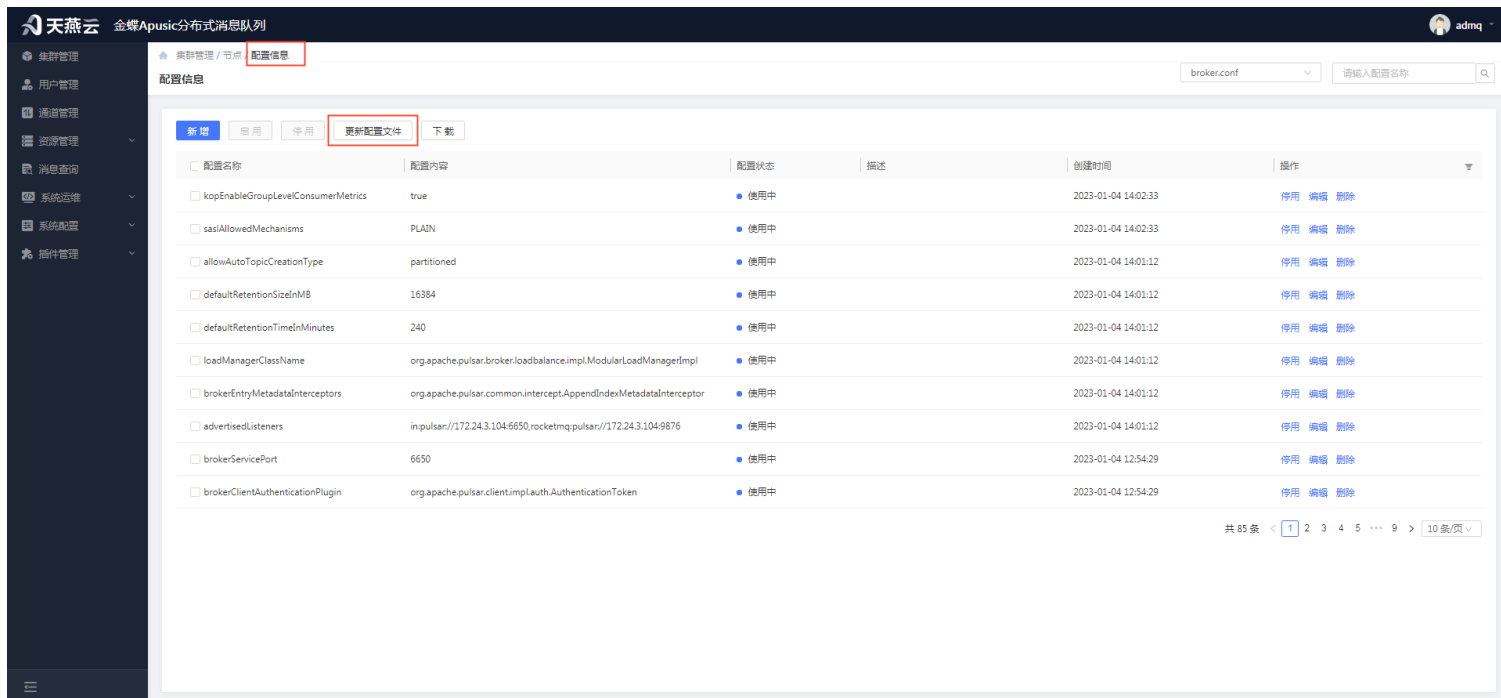
# 设置不自动删除非活跃主题 (可不添加)
brokerDeleteInactiveTopicsEnabled=false

```

添加配置项：



添加配置项完成后需要更新配置文件到节点：



备注：更新配置文件到节点后，需要重启节点。

3.5.3.2.2 RocketMQ 插件

方式同Kafka，涉及的配置项如下：

```

messagingProtocols=rocketmq
# 插件协议路径 ({程序目录}/{集群名称}/{节点类型}/admq-V2.1.0/protocols)
# 程序目录：添加服务器时设置的程序目录
# 集群名称：添加集群时设置的集群名称
# 节点类型：如果是单机部署，则是standalone；如果是集群部署则是broker
protocolHandlerDirectory=/home/admq/pro/apusic-mq/broker/admq-V2.2.1/protocols
narExtractionDirectory=/home/admq/pro/apusic-mq/broker/admq-V2.2.1/nar

# rocketmq
rocketmqTenant=rocketmq-data
rocketmqMetadataTenant=rocketmq-meta
rocketmqListeners=rocketmq://172.20.140.140:9876
rocketmqListenerPortMap=9876:rocketmq

loadManagerClassName=org.apache.pulsar.broker.loadbalance.impl.ModularLoadManagerImpl
brokerEntryMetadataInterceptors=org.apache.pulsar.common.intercept.AppendIndexMetadataInterceptors

# 设置消息保留时间，单位：分钟
defaultRetentionTimeInMinutes=2400
# 设置消息保留大小，单位：MB
defaultRetentionSizeInMB=16384
# 设置自动创建分区主题
allowAutoTopicCreationType=partitioned
# 设置不自动删除非活跃主题（可不加）
brokerDeleteInactiveTopicsEnabled=false

advertisedListeners=INTERNAL:pulsar://172.20.140.140:6650,rocketmq:pulsar://172.20.140.140:9876
ropBrokerReplicationNum=1
ropTraceTopicEnable=false
ropRestServerPort=9888
ropAclEnable=false

```

3.5.3.2.3 RabbitMQ 插件

方式同Kafka，涉及的配置项如下：

```

messagingProtocols=amqp
# 插件协议路径 ({程序目录}/{集群名称}/{节点类型}/admq-V2.1.0/protocols)
# 程序目录：添加服务器时设置的程序目录
# 集群名称：添加集群时设置的集群名称
# 节点类型：如果是单机部署，则是standalone；如果是集群部署则是broker
protocolHandlerDirectory=/home/admq/pro/apusic-mq/broker/admq-V2.2.1/protocols
narExtractionDirectory=/home/admq/pro/apusic-mq/broker/admq-V2.2.1/nar
amqpTenant=amqp-data
amqpMetadataTenant=amqp-meta
amqpListeners=amqp://172.20.140.140:5672

```

```
amqpMaxNoOfChannels=2047
amqpMaxFrameSize=4194304
amqpProxyEnable=true
amqpProxyPort=6672
```

3.5.3.2.4 MQTT 插件

方式同Kafka，涉及的配置项如下：

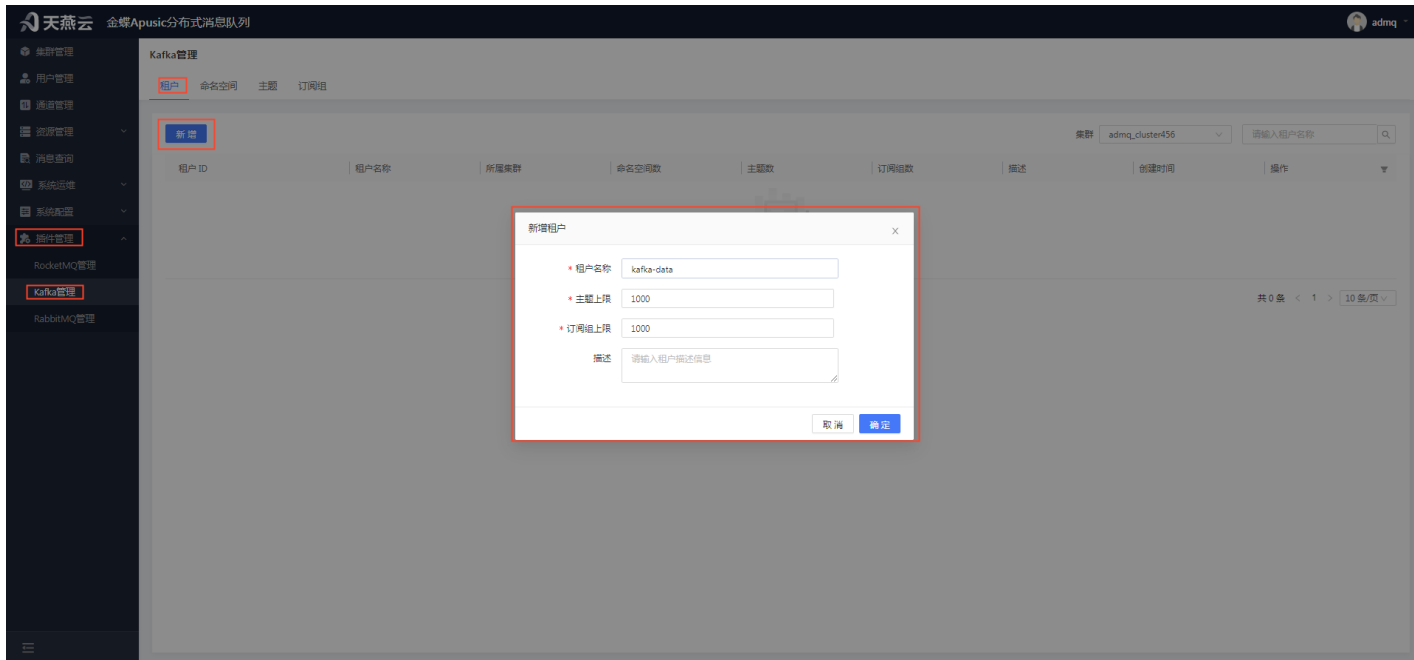
```
messagingProtocols=mqtt
# 插件协议路径 ({程序目录}/{集群名称}/{节点类型}/admq-V2.1.0/protocols)
# 程序目录：添加服务器时设置的程序目录
# 集群名称：添加集群时设置的集群名称
# 节点类型：如果是单机部署，则是standalone；如果是集群部署则是broker
protocolHandlerDirectory=/home/admq/pro/apusic-mq/broker/admq-V2.2.1/protocols
narExtractionDirectory=/home/admq/pro/apusic-mq/broker/admq-V2.2.1/nar
mqttListeners=mqtt://172.20.140.140:1883
mqttProxyEnabled=true
mqttProxyPort=5682
```

3.5.3.3 插件使用

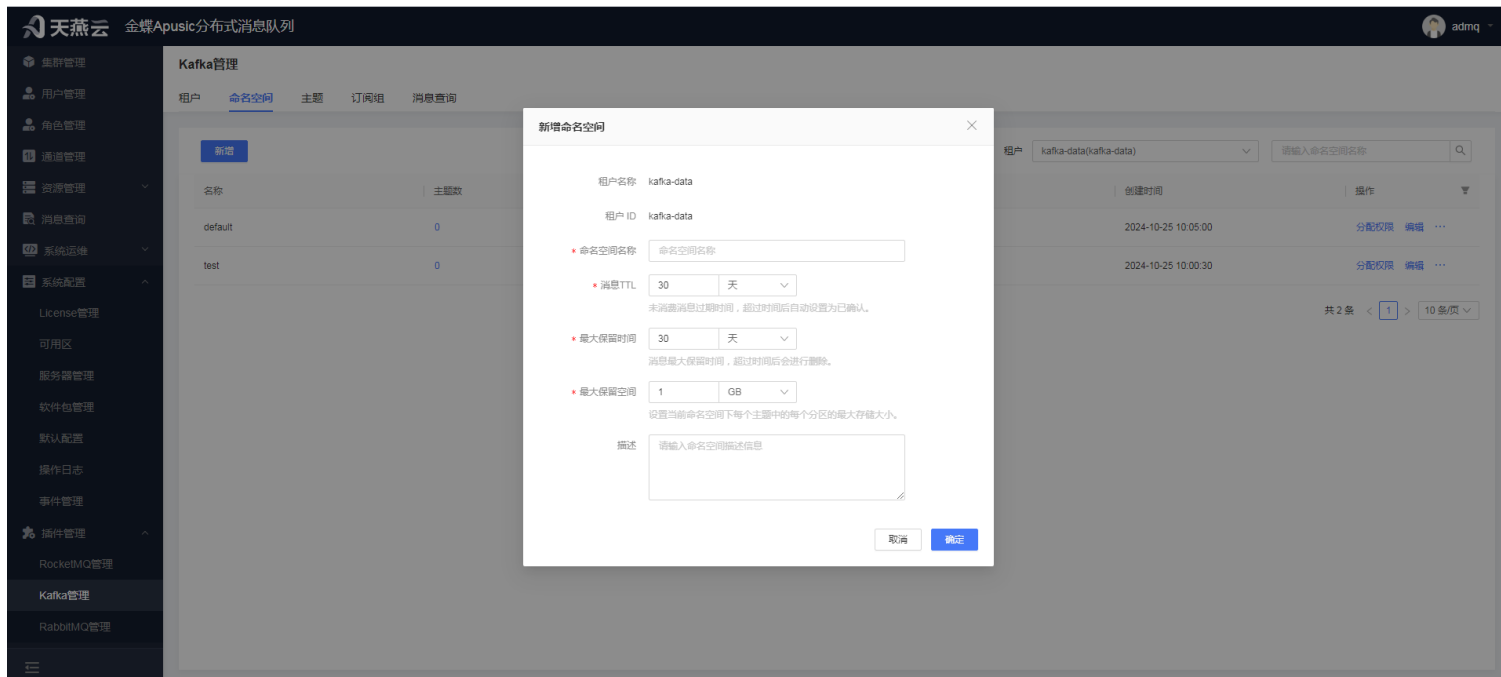
3.5.3.3.1 Kafka 插件

在插件管理 - Kafka管理 - 租户页面创建租户：kafka-data，在该租户下创建命名空间：default。

- 租户创建：



- 命名空间创建：



备注：客户端使用的租户名是租户ID，租户ID是以kafka-开头，创建租户后租户ID会自动生成且以“kafka-”开头。

之后可以使用kafka客户端连接到 IP1:9092;IP2:9092;IP3:9092服务地址（单机服务地址：IP:9092）进行发送和接收消息，demo如下：

```
public void sendMessage() {
    long time = System.currentTimeMillis();
    int limit = speed < 1 ? Integer.MAX_VALUE : speed;
    LongAdder curCount = new LongAdder();

    Properties props = new Properties();
    props.put("bootstrap.servers", service);
    props.put("acks", "all");
    props.put("retries", 0);
    props.put("batch.size", 16384);
    props.put("key.serializer", StringSerializer.class.getName());
    props.put("value.serializer", StringSerializer.class.getName());

    KafkaProducer<String, String> producer = new KafkaProducer<>(props);

    int i = 0;
    while (repeat == -1 || i < repeat) {
        String messageValue = message + "_" + i;

        ProducerRecord<String, String> record = new ProducerRecord<>(topic, messageValue);
        producer.send(record);

        log.info("send message: {}", messageValue);
        if (interval > 0) {
            Thread.sleep(interval);
        }
        time = speedLimit(curCount, limit, time);
    }
}
```

```
        i ++;
    }

    producer.close();
}

public void receiveMessage() {
    List<String> topics = new ArrayList<>();
    topics.add(topic);

    Properties props = new Properties();
    props.put("bootstrap.servers", service);
    props.put("group.id", "admq");
    props.put("auto.offset.reset", "earliest");
    props.put("key.deserializer", StringDeserializer.class.getName());
    props.put("value.deserializer", StringDeserializer.class.getName());

    KafkaConsumer<String, String> consumer = new KafkaConsumer<>(props);
    consumer.subscribe(topics);

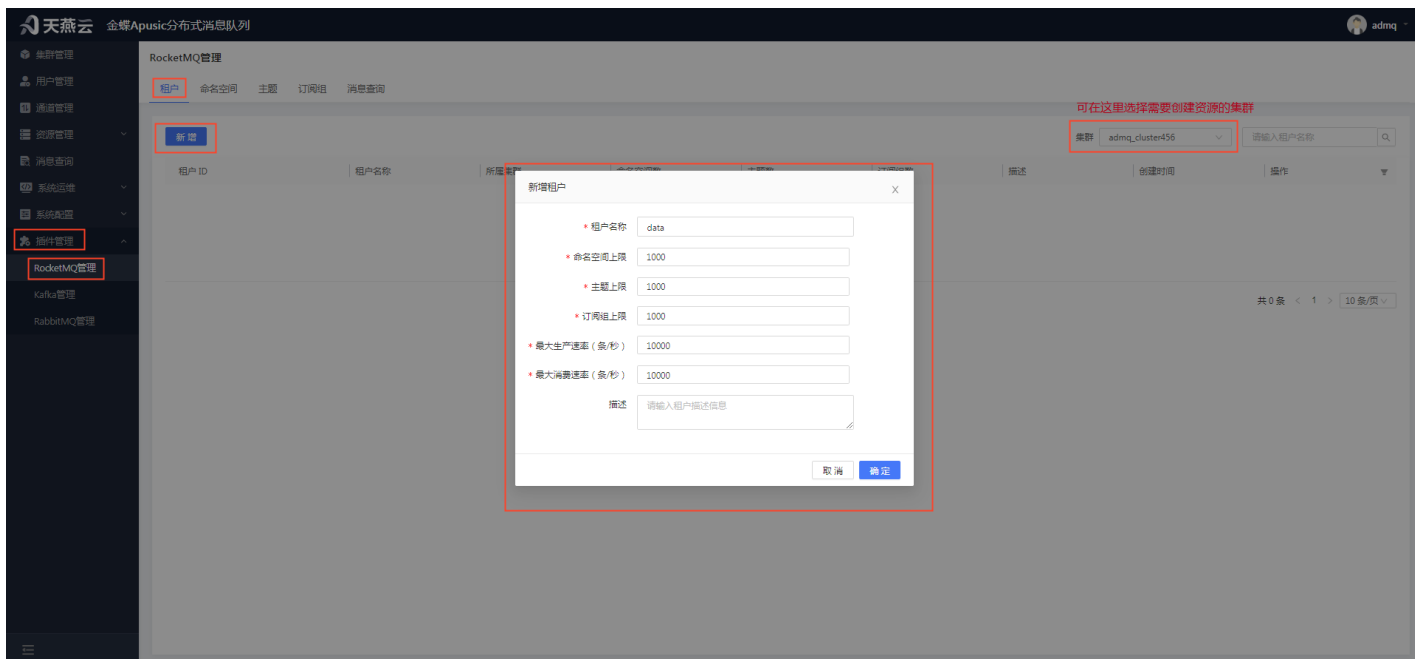
    while (true) {
        ConsumerRecords<String, String> records = consumer.poll(Duration.ofSeconds(120));
        if (records == null || records.isEmpty()) {
            break;
        }
        for (ConsumerRecord<String, String> record : records) {
            log.info("receive message: {}", record.value());
        }
    }

    consumer.close();
}
```

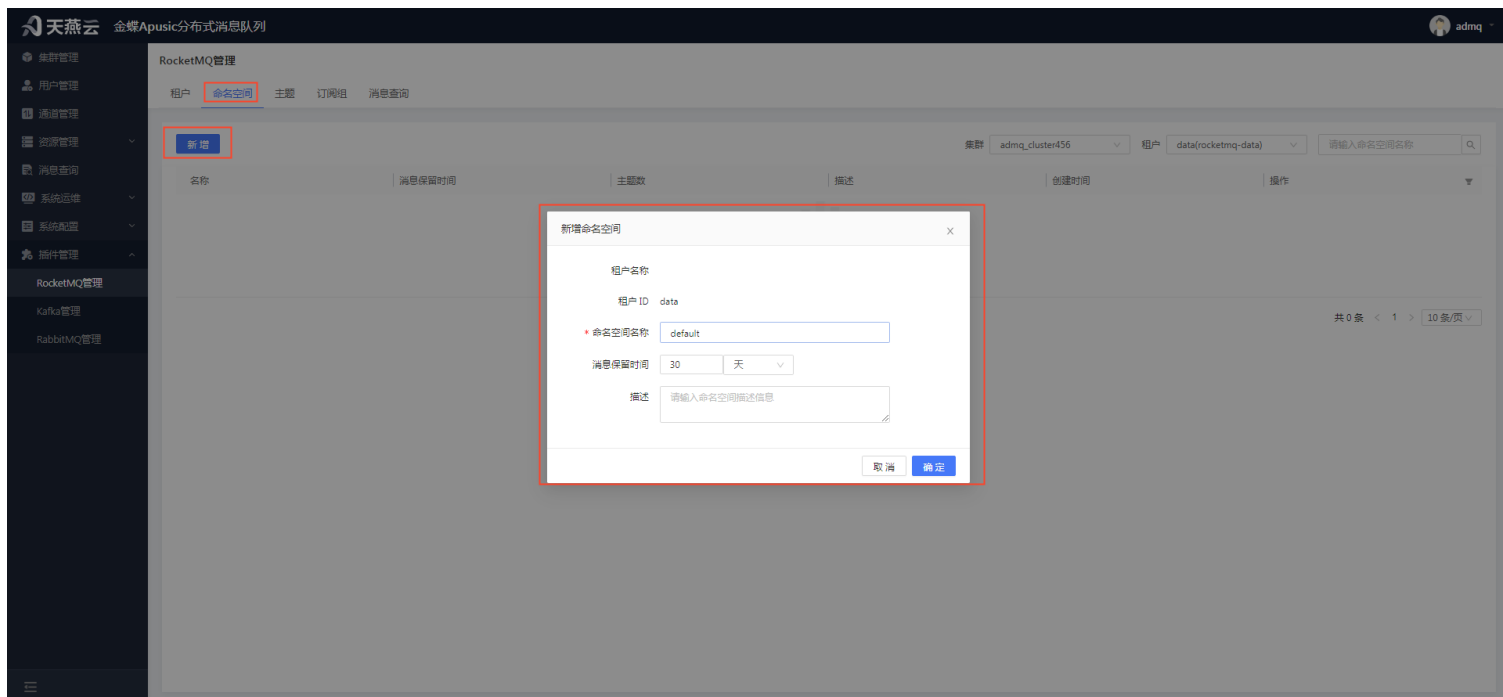
3.5.3.3.2 RocketMQ 插件

在插件管理 - RocketMQ管理 - 租户页面创建租户: data, 在该租户下创建命名空间: default。

- 创建租户:



- 创建命名空间:



备注: 客户端使用的租户名是租户ID, 租户ID是以rocketmq-开头, 创建租户后租户ID会自动生成且以“rocketmq-”开头。

之后可以使用RocketMQ客户端连接到 IP1:9876;IP2:9876;IP3:9876服务地址 (单机服务地址: IP:9876) 接收和发送消息 (设置NamesrvAddr), 示例如下:

```
public void testSendAndReceive() throws Exception {
    int maxReConsumeTimes = 3;
    int consumeTimeout = 5;

    String groupName = "vv-group";
    String topic = "vv-topic";
```

```

DefaultMQProducer producer = new DefaultMQProducer(groupName);
producer.setNamesrvAddr(nameServer);
producer.start();
for (int i = 0; i < 2; i++) {
    Message msg = new Message(topic, "vv", ("hello world - " + i).getBytes());
    SendResult sendResult = producer.send(msg);
    System.out.println(time() + " send result: " + sendResult);
}

subscribe(groupName, nameServer, maxReConsumeTimes, consumeTimeout, topic);
while (true);
}

void subscribe(String groupName, String nameServer, int maxReConsumeTimes, int consumeTimeout,
String topic) throws Exception {
    DefaultMQPushConsumer consumer = new DefaultMQPushConsumer(groupName, false);
    consumer.setNamesrvAddr(nameServer);
    consumer.setMaxReconsumeTimes(maxReConsumeTimes);
    consumer.setConsumeTimeout(consumeTimeout);
    consumer.setPullBatchSize(1024);
    consumer.setPullInterval(100);
    consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_FIRST_OFFSET);
    consumer.subscribe(topic, "*");

    consumer.registerMessageListener((MessageListenerConcurrently) (msgList, context) -> {

        msgList.forEach(action -> {
            System.out.println(time() + " recv topic: " + topic + " - " + action.getTopic()
                + ", msgId: " + action.getMsgId()
                + ", data: " + new String(action.getBody()));
        });

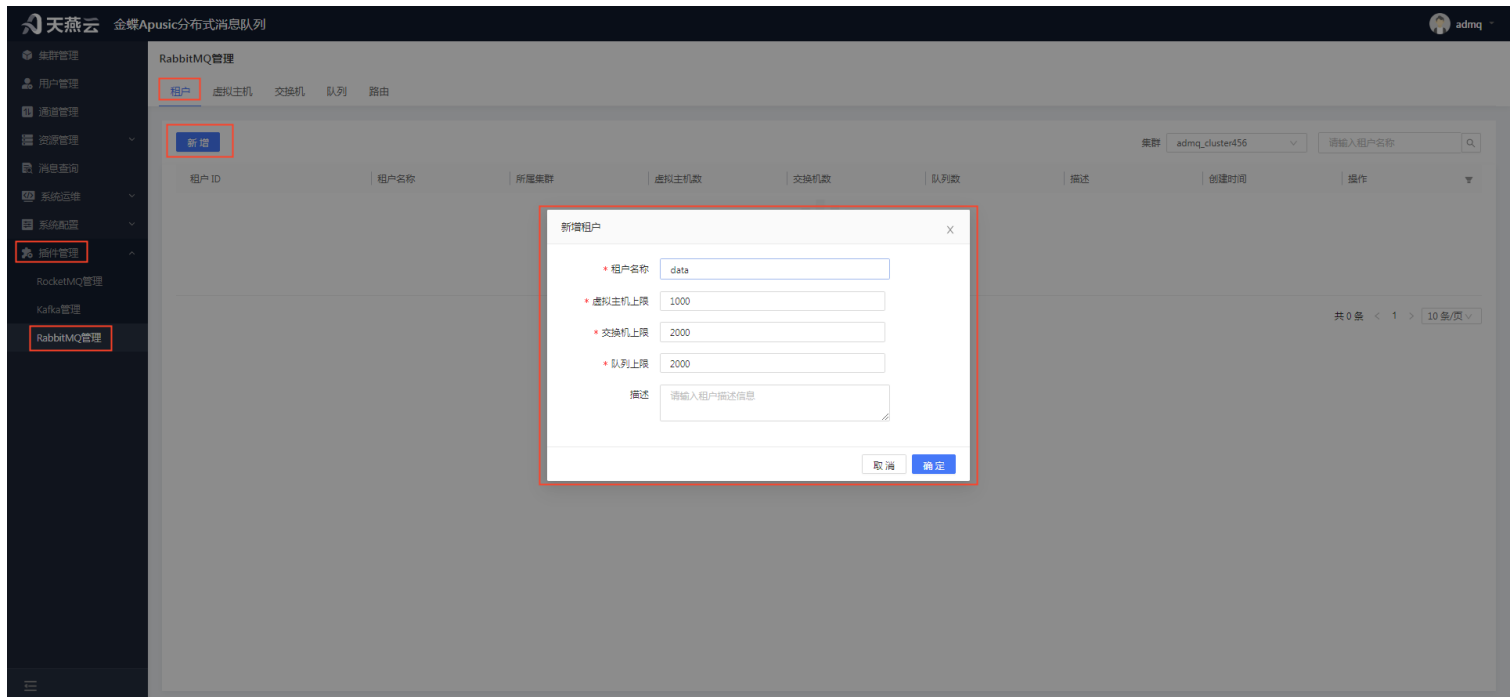
        return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
    });
    consumer.start();
}

```

3.5.3.3.3 RabbitMQ 插件

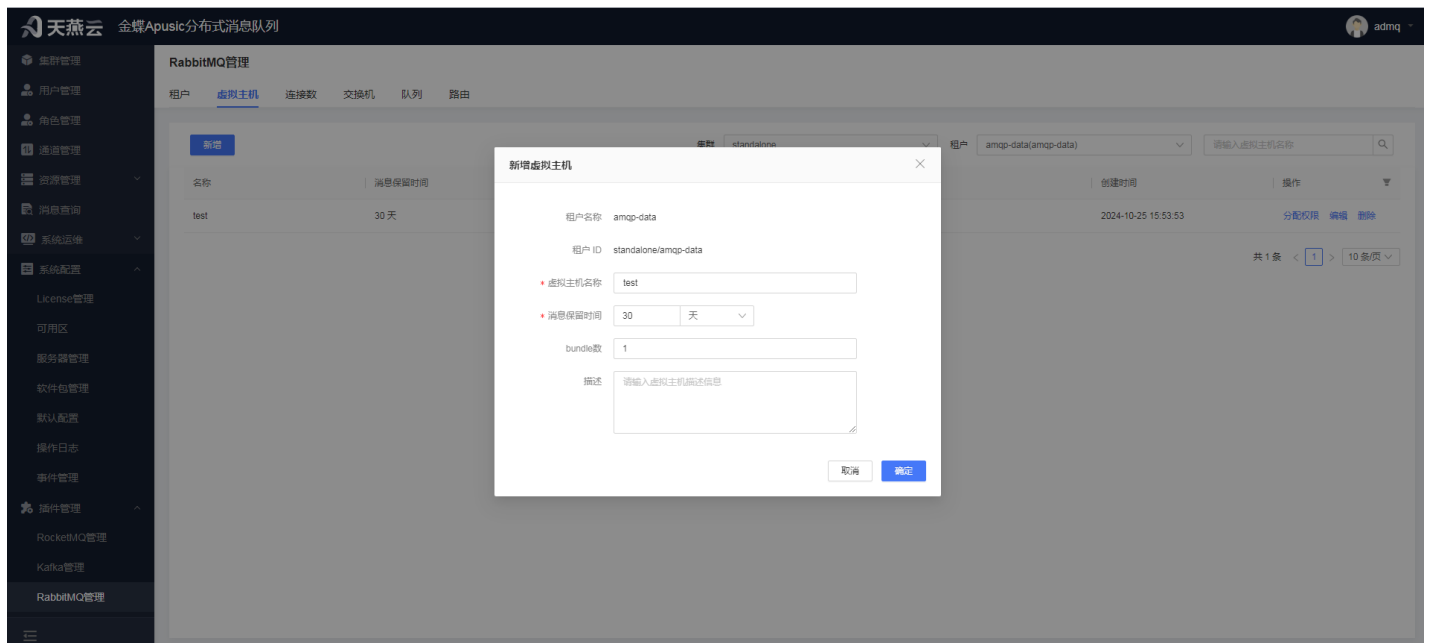
在插件管理 - RabbitMQ管理 - 租户页面创建租户：data，在该租户下创建虚拟主机、交换机、队列、路由资源。

- 创建租户：

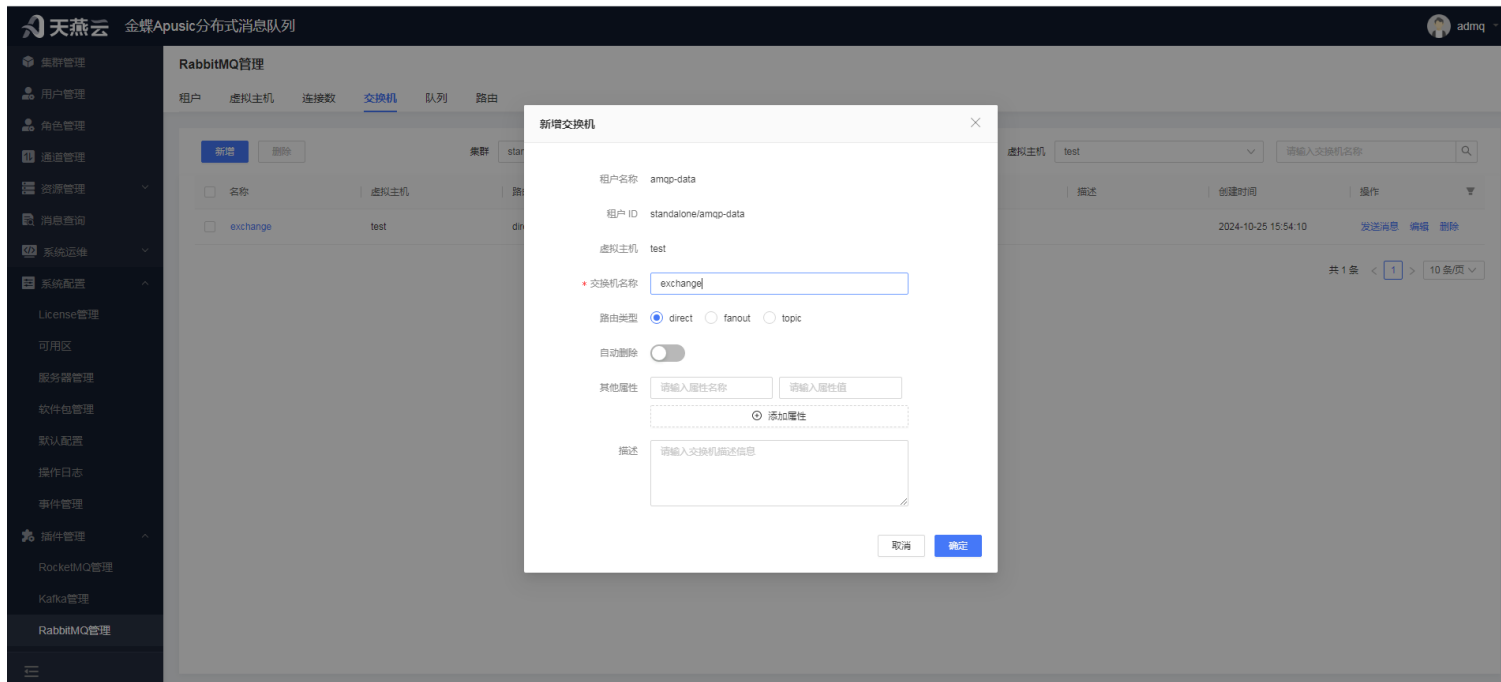


备注：RabbitMQ插件的租户ID需和配置文件中租户名称保持一致。

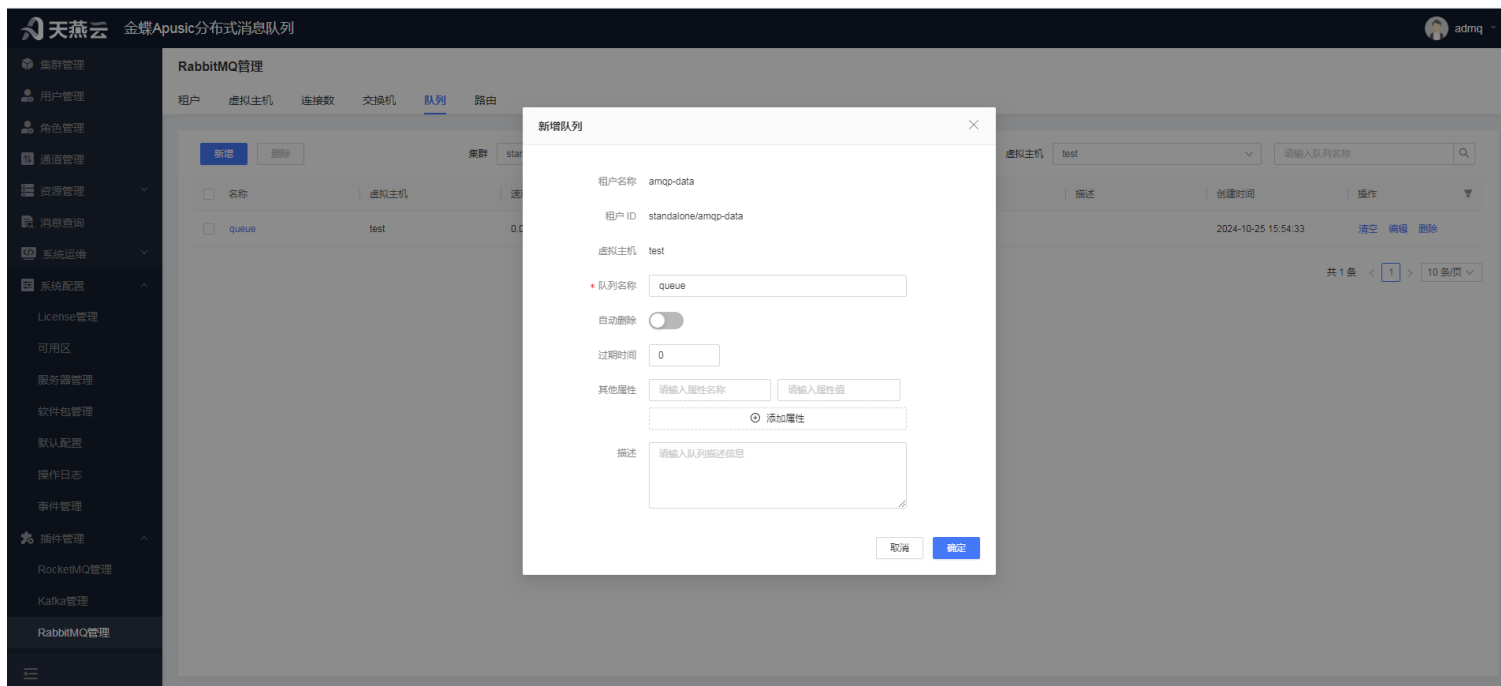
- 创建虚拟主机：



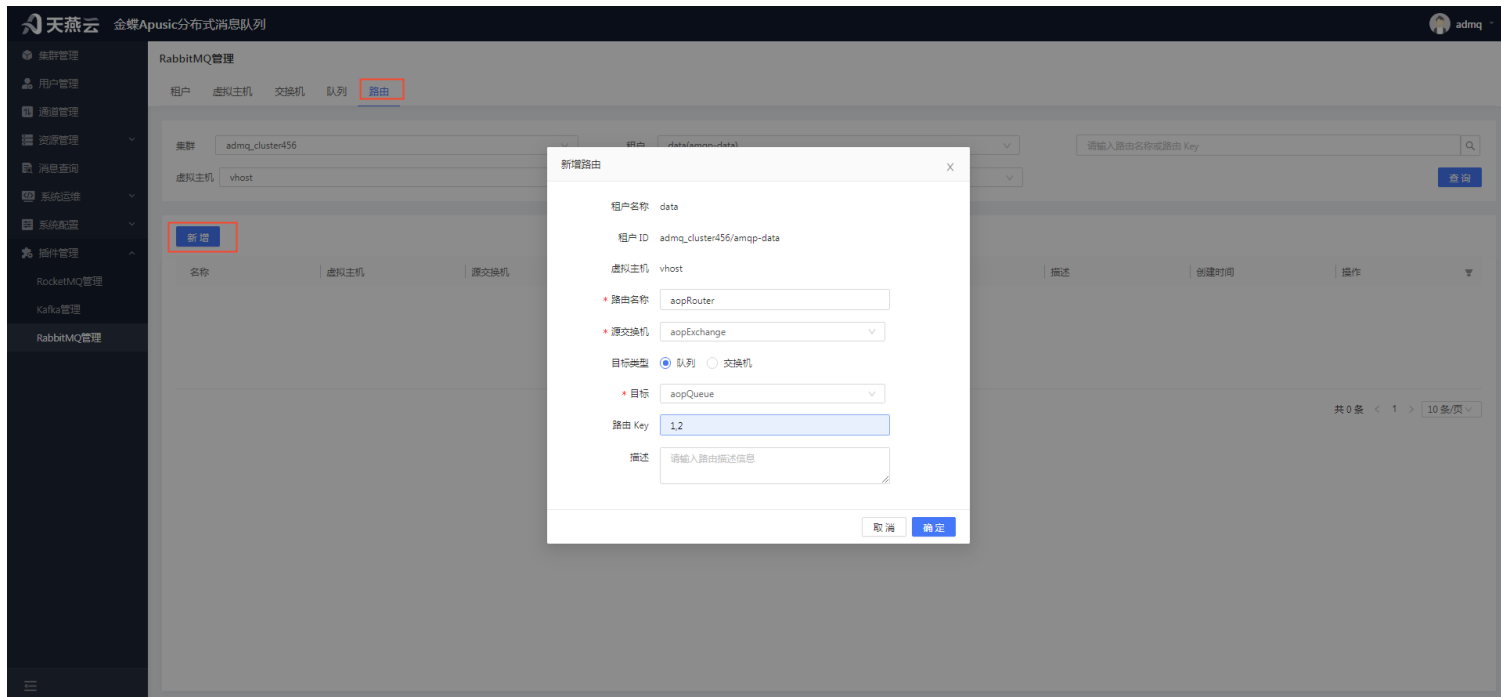
- 创建交换机：



• 创建队列:



• 创建路由:



- 在租户页创建租户：amqp-data；点击租户名称，创建命名空间：vhost1（和客户端使用的vhost名称保持一致）。

之后可以使用RabbitMQ客户端连接到服务地址接收和发送消息（虚拟主机名称需和客户端使用的vhost名称保持一致），示例如下：

```
private String exchange = "aopExchange";
private String server = "172.20.140.140";
private int port = 5672;
private String vhost = "vhost";

private Connection connection;
private Channel channel;

// 发送消息
public void send(String message) throws Exception {
    for (int i = 0; i < 100; i++) {
        String messageValue = message + "_" + i;
        channel.basicPublish(exchange, "", null, messageValue.getBytes());
        log.info("send message: {}", messageValue);
    }
}

// 接收消息
public void receive() throws Exception {
    channel.basicConsume(queue, true, new DefaultConsumer(channel) {
        @Override
        public void handleDelivery(String consumerTag, Envelope envelope, AMQP.BasicProperties
properties,
                                byte[] body) throws IOException {
            log.info("receive message: {}", new String(body));
        }
    });
}
```

```

});

while (true);
}

// 建立连接
public void createConnectionAndChannel() throws Exception {
    ConnectionFactory connectionFactory = new ConnectionFactory();
    connectionFactory.setVirtualHost(vhost);
    connectionFactory.setHost(server);
    connectionFactory.setPort(port);
    connection = connectionFactory.newConnection();
    channel = connection.createChannel();

    // exchange declare
    channel.exchangeDeclare(exchange, BuiltinExchangeType.FANOUT, true, false, false, null);

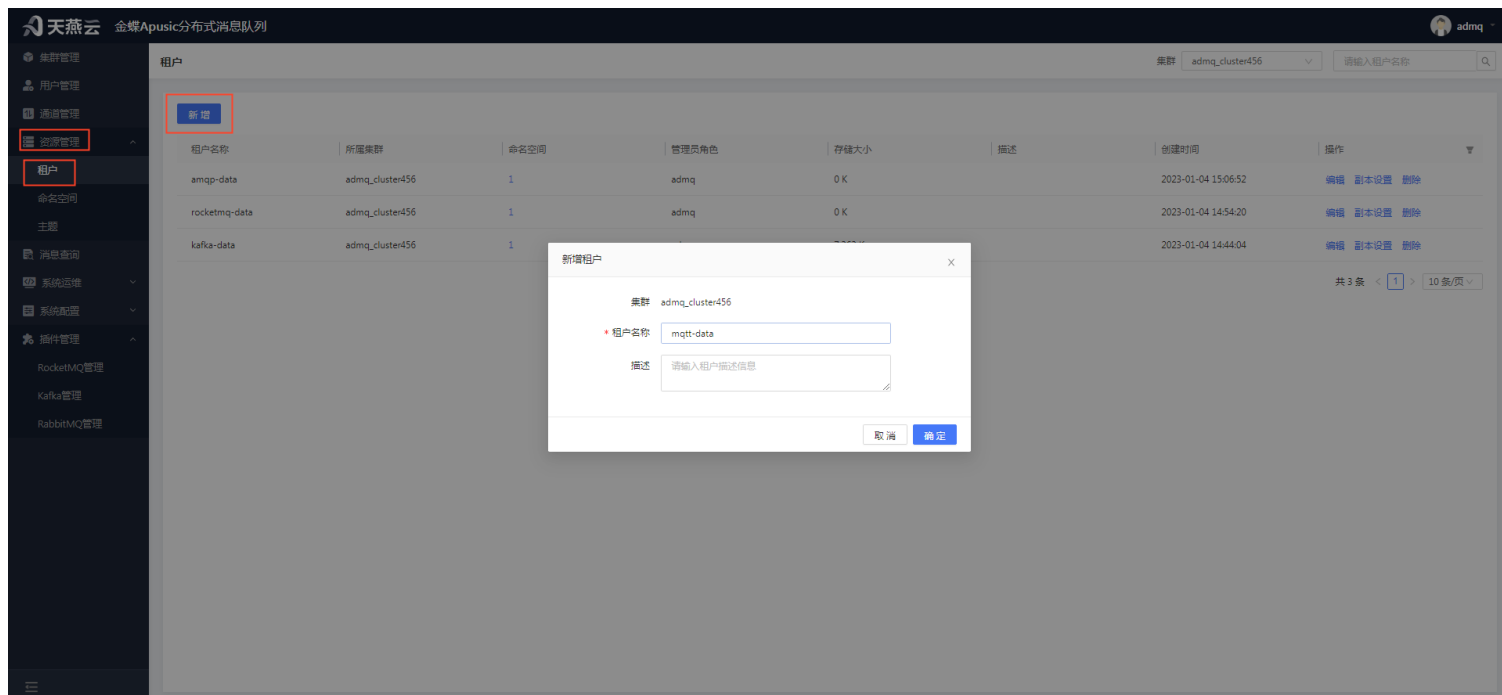
    // queue declare and bind
    channel.queueDeclare(queue, true, false, false, null);
    channel.queueBind(queue, exchange, "");
}

```

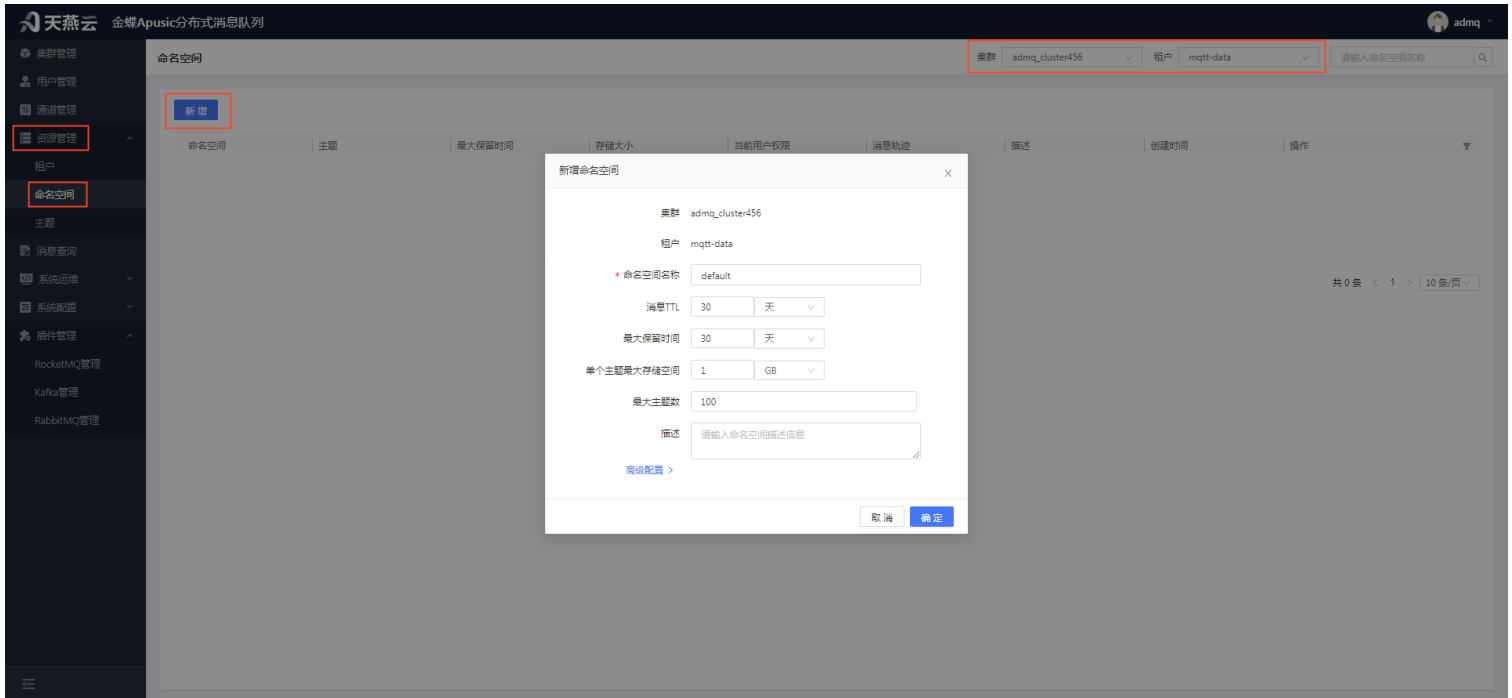
3.5.3.3.4 MQTT 插件

在资源管理 - 租户页面创建租户以"mqtt-"开头的租户如: mqtt-data, 在该租户下命名空间default、主题mqtt-topic。

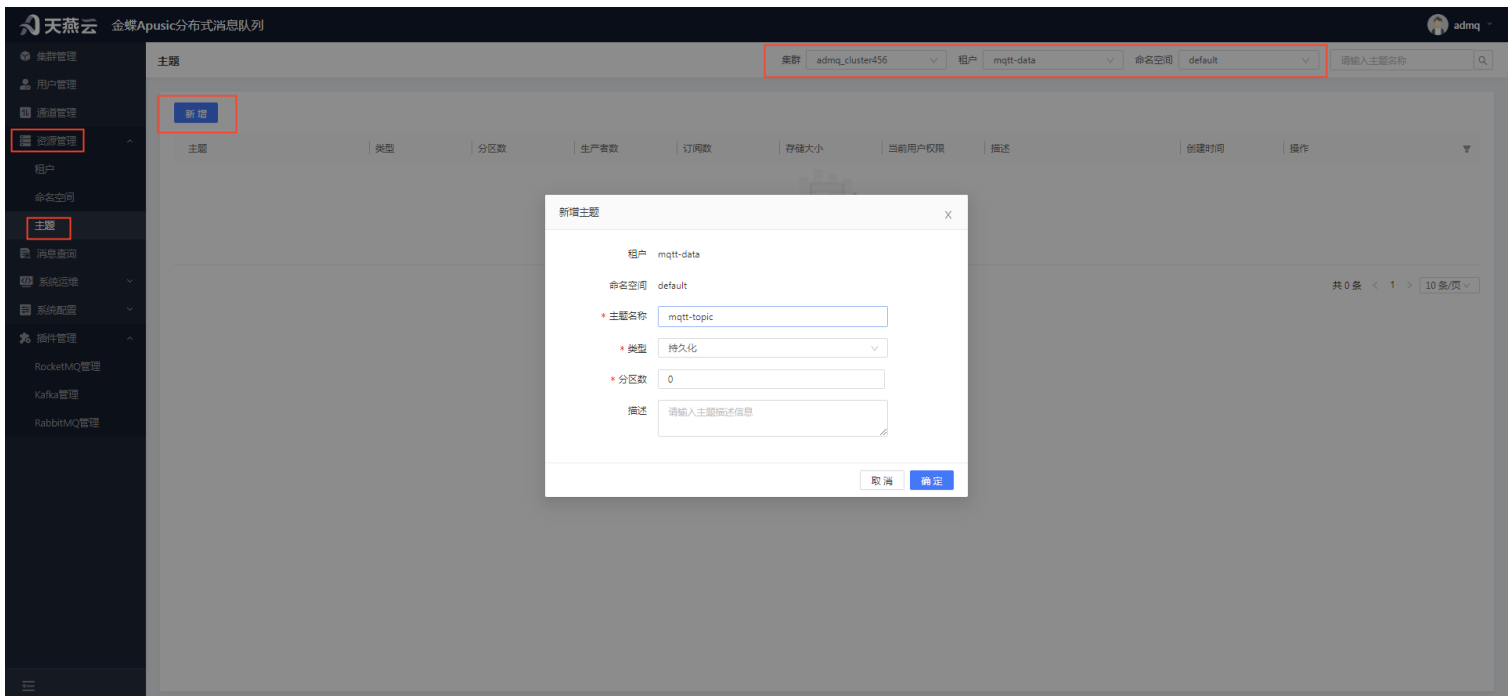
创建租户:



创建命名空间:



创建主题：



之后可以使用RabbitMQ客户端连接到服务地址接收和发送消息，示例如下：

```
private String server = "IP"
private int port = 1883;
private String topic = "mqtt-topic";

public void receiveMessage() throws Exception {
    MQTT mqtt = new MQTT();
    mqtt.setHost(server, port);
}
```

```

BlockingConnection connection = mqtt.blockingConnection();
connection.connect();
Topic[] topics = { new Topic(topic, QoS.AT_LEAST_ONCE) };
connection.subscribe(topics);

while (true) {
    Message message = connection.receive(120, TimeUnit.SECONDS);
    if (message == null) {
        break;
    }
    message.ack();
    log.info("receive message: {}", new String(message.getPayload()));
}
connection.disconnect();
}

public void sendMessage() throws Exception {
    long time = System.currentTimeMillis();
    int limit = speed < 1 ? Integer.MAX_VALUE : speed;
    LongAdder curCount = new LongAdder();

    MQTT mqtt = new MQTT();
    mqtt.setHost(server, port);
    BlockingConnection connection = mqtt.blockingConnection();
    connection.connect();
    int i = 0;
    while (repeat == -1 || i < repeat) {
        String messageValue = message + "_" + i;
        connection.publish(topic, messageValue.getBytes(), QoS.AT_LEAST_ONCE, false);
        log.info("send message: {}", messageValue);
        if (interval > 0) {
            Thread.sleep(interval);
        }
        time = speedLimit(curCount, limit, time);
        i++;
    }
    connection.disconnect();
}
}

```

3.6 通过命令行安装ADMQ

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。

3.6.1 节点授权

节点授权目前支持两种授权模式：KBC模式、授权中心模式，可选择其中一种授权模式进行授权。

1.KBC模式

节点授权模式适用于没有安装ADMQ管控台,直接用命令行的方式安装ADMQ节点

获取特征码之前需要手动执行下启动命令

```
./bin/admq standalone
```

然后通过如下两种方式, 可以获取特征码

- 通过日志获取

```
cat logs/standalone-linux-4-195.log | grep kbc
```

```
[root@openstack-140-170 admq-V2.4.4]# cat logs/standalone-openstack-140-170.novaLocal.log | grep kbc
15:02:50.852 [main] INFO com.apusic.admq.license.handler.ManagerEndLicenseHandler - kbc授权码: SZTY-807951870
[root@openstack-140-170 admq-V2.4.4]#
```

- 通过特征码文件获取

```
cat licenses/LICENSE-kbc_authorize_code.txt
```

```
[root@openstack-140-170 admq-V2.4.4]# cat licenses/LICENSE-kbc_authorize_code.txt
SZTY-807951870[root@openstack-140-170 admq-V2.4.4]#
```

通过特征码激活授权, 然后将申请到许可文件更名为license.xml后缀上传到admq-license文件夹中

```
[root@openstack-140-170 admq-V2.4.4]# cd admq-licenses/
[root@openstack-140-170 admq-licenses]# ls
license.xml
[root@openstack-140-170 admq-licenses]#
```

2、授权中心模式

单机模式需要修改 config/standalone.conf文件

集群模式需要修改 config/broker.conf和config/storage.conf两个文件

```
## 监控信息不采集为
ropRestServerPort=9888
ropAclEnable=false
# 定时任务分区数
rmqScheduleTopicPartitionNum=1
# 偏移量管理分区数
offsetsTopicNumPartitions=3

### --- 日志配置 --- ###
logLevel=info
logSize=250M
logKeepTime=30d
logKeepSize=10G

apusic_acls_enable=true
apusic_acls_authUrls=1
apusic_acls_tenant=public
apusic_acls_ns=深
```

```
apusic_acls_enable=true
apusic_acls_authUrls=ip:port
apusic_acls_tenant=public
apusic_acls_ns=xxxx有限公司
```

3.6.2 单机部署

解压软件包并进入安装目录

```
tar zxvf admq-V2.4.5-all.tar.gz
cd admq-V2.4.5
```

修改配置文件

进入conf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
advertisedAddress	本机IP	127.0.0.1
clusterName	集群名称	apusic-mq-standalone

放置License文件

把license文件拷贝到admq-licenses目录。

启动

```
bin/admq-daemon start standalone
```

检查状态

```
bin/admqctl admin clusters list
```

如果有结果则表示正常。

停止

```
bin/admq-daemon stop standalone
```

3.6.3 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部署三种类型的组件。

3.6.3.1 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/v2.4.5/zk
cp admq-V2.4.5-all.tar.gz /admq/v2.4.5/zk
```

解压软件包并进入目录

```
tar zxvf admq-V2.4.5-all.tar.gz
cd admq-V2.4.5
```

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。

```

Welcome to ZooKeeper!
Jline support is enabled
18:45:35.071 [main-SendThread(172.20.140.161:2181)] INFO org.apache.zookeeper.ClientCnxn - Opening socket connection to server 172.20.140.161/172.20.140.161:2181.
18:45:35.071 [main-SendThread(172.20.140.161:2181)] INFO org.apache.zookeeper.ClientCnxn - SASL config status: Will not attempt to authenticate using SASL (unknown error)
18:45:35.081 [main-SendThread(172.20.140.161:2181)] INFO org.apache.zookeeper.ClientCnxn - Socket connection established, initiating session, client: /18.0.0.222:54844, sess
18:45:35.101 [main-SendThread(172.20.140.161:2181)] INFO org.apache.zookeeper.ClientCnxn - Session establishment complete on server 172.20.140.161/172.20.140.161:2181, sess
WATCHER::
[zk: 172.20.140.161:2181(CONNECTED) 0] ls /
[zk: 172.20.140.161:2181(CONNECTED) 1]

```

3.6.3.2 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

```

18:51:17.688 [Curator-Framework-0] INFO org.apache.curator.framework.imps.CuratorFrameworkImpl - backgroundOperationsLoop exiting
18:51:17.688 [main-EventThread] INFO org.apache.zookeeper.ClientCnxn - EventThread shut down for session: 0x1018c43f84e0003
18:51:17.688 [main] INFO org.apache.zookeeper.ZooKeeper - Session: 0x2018c43f1f0003 closed
18:51:18.472 [main-SendThread(172.20.140.161:2181)] WARN org.apache.zookeeper.ClientCnxn - An exception was thrown while closing send thread for session 0x1018c43f84e0002.
org.apache.zookeeper.ClientCnxn$IOException: Unable to read additional data from server sessionid 0x1018c43f84e0002, likely server has closed socket
    at org.apache.zookeeper.ClientCnxn$SendThread.run(ClientCnxn.java:77) ~[org.apache.zookeeper-zookeeper-3.6.3.jar:3.6.3]
    at org.apache.zookeeper.ClientCnxn$SocketNIO.doTransport(ClientCnxn$SocketNIO.java:350) ~[org.apache.zookeeper-zookeeper-3.6.3.jar:3.6.3]
    at org.apache.zookeeper.ClientCnxn$SendThread.run(ClientCnxn.java:1290) ~[org.apache.zookeeper-zookeeper-3.6.3.jar:3.6.3]
18:51:18.574 [main] INFO org.apache.zookeeper.ZooKeeper - Session: 0x1018c43f84e0002 closed
18:51:18.574 [metadata-store-3-1] INFO org.apache.pulsar.metadata.impl.ZKSessionWatcher - Got ZK session watch event: WatchedEvent state:Closed type:None path:null
18:51:18.575 [main-EventThread] INFO org.apache.zookeeper.ClientCnxn - EventThread shut down for session: 0x1018c43f84e0002
18:51:18.585 [main-SendThread(172.20.140.161:2181)] WARN org.apache.zookeeper.ClientCnxn - An exception was thrown while closing send thread for session 0x1018c43f84e0003.
org.apache.zookeeper.ClientCnxn$IOException: Unable to read additional data from server sessionid 0x1018c43f84e0003, likely server has closed socket
    at org.apache.zookeeper.ClientCnxn$SendThread.run(ClientCnxn.java:77) ~[org.apache.zookeeper-zookeeper-3.6.3.jar:3.6.3]
    at org.apache.zookeeper.ClientCnxn$SocketNIO.doTransport(ClientCnxn$SocketNIO.java:350) ~[org.apache.zookeeper-zookeeper-3.6.3.jar:3.6.3]
    at org.apache.zookeeper.ClientCnxn$SendThread.run(ClientCnxn.java:1290) ~[org.apache.zookeeper-zookeeper-3.6.3.jar:3.6.3]
18:51:18.688 [main-EventThread] INFO org.apache.zookeeper.ZooKeeper - Session: 0x1018c43f84e0003 closed
18:51:18.688 [main] INFO org.apache.zookeeper.ZooKeeper - Session: 0x1018c43f84e0003 closed
18:51:18.687 [metadata-store-3-1] INFO org.apache.pulsar.metadata.impl.ZKSessionWatcher - Got ZK session watch event: WatchedEvent state:Closed type:None path:null
18:51:18.688 [main] INFO org.apache.pulsar.PulsarClusterMetadataSetup - Cluster metadata for "apusic-mq" setup correctly

```

3.6.3.3 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/v2.4.5/storage
cp admq-V2.4.5-all.tar.gz /admq/v2.4.5/storage

```

解压软件包并进入目录

```

tar zxvf admq-V2.4.5-all.tar.gz
cd admq-V2.4.5

```

放置License文件

把license文件拷贝到admq-licenses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
advertisedAddress	本机IP	127.0.0.1
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/ledgers
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
```

```
19:07:36.362 [main] INFO org.apache.bookkeeper.tools.cli.commands.BookiesListBookiesCommand - All Bookies:
19:07:48.381 [main] INFO org.apache.bookkeeper.tools.cli.commands.BookiesListBookiesCommand - Bookie0:172.20.140.163:3181, IP:172.20.140.163, Port:3181, Hostname:172.20.140.163
19:07:48.382 [main] INFO org.apache.bookkeeper.tools.cli.commands.BookiesListBookiesCommand - Bookie1:172.20.140.162:3181, IP:172.20.140.162, Port:3181, Hostname:openstack-140-162
19:07:48.383 [main] INFO org.apache.bookkeeper.tools.cli.commands.BookiesListBookiesCommand - Bookie2:172.20.140.161:3181, IP:172.20.140.161, Port:3181, Hostname:openstack-140-161
19:07:48.392 [main-SendThread(172.20.140.161:2181)] WARN org.apache.bookkeeper.ClientUtils - An exception was thrown while listing send threads for session 8d1d6c13f8e6bb.
org.apache.bookkeeper.ClientUtils$SendThreadException: Unable to read additional data from server sessionid 8d1d6c13f8e6bb, likely server has closed socket
at org.apache.bookkeeper.client.ClientUtils$SendThread.run(ClientUtils.java:110) [19:07:48.392:WARN:org.apache.bookkeeper:ClientUtils:110]
```

可以查询到三个storage节点

3.6.3.4 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/v2.4.5/broker
cp admq-V2.4.5-all.tar.gz /admq/v2.4.5/broker
```

解压软件包并进入目录

```
tar zxvf admq-V2.4.5-all.tar.gz
```

```
cd admq-V2.4.5
```

放置License文件

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
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
advertisedAddress	本机IP	127.0.0.1
clusterName	集群名称	apusic-mq-standalone

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]#
```

4 快速入门

4.1 客户端使用

以JAVA客户端为例:

- 创建ADMQ客户端连接

```
public class Global {
    public static String clusterUrl = "pulsar://ip:6650";
    public static String defaultTopic = "persistent://租户/命名空间/主题";
}
```

```
public class GetPulsarClient {
    public static PulsarClient get(){
        try{
            PulsarClient pulsarClient;
            pulsarClient=PulsarClient
                .builder()
                .serviceUrl(Global.clusterUrl)
                .authentication(AuthenticationFactory.token("token值"))
                .build();
            return pulsarClient;
        }catch(Exception e){
            System.out.println(e);
            return null;
        }
    }
}
```

- 创建ADMQ客户端消费者

```
public class simpleConsumer {
    public static void main(String[] args) {
        try{
            receiveSync();
        }catch (Exception e){
            System.out.println(e);
        }
    }
    public static void receiveSync() throws PulsarClientException {
        Consumer<byte[]> simpleConsumer= GetPulsarClient.get().newConsumer()
            .topic(Global.defaultTopic)
            .subscriptionName("admqSub3")
            .subscribe();
        while(true){
            //System.out.println("simpleConsumerSync received: ");
        }
    }
}
```

```

Message msg = simpleConsumer.receive();
//System.out.println("simpleConsumerSync received: "+new String(msg.getData()));
try{
    System.out.println("simpleConsumerSync received: "+new String(msg.getData()));
    simpleConsumer.acknowledge(msg);
    System.out.println("this msg is acked!!!");
}catch(Exception e){
    System.out.println(e);
    simpleConsumer.negativeAcknowledge(msg);
}
}
}
}

```

- 创建ADMQ客户端生产者

```

public class simpleProducer {
    public static void main(String[] args){
        try{
            sendSimpleMessageSync();
        }catch(Exception e){
            e.printStackTrace();
            System.err.println(e);
        }
    }

    public static void sendSimpleMessageSync() throws PulsarClientException,
    InterruptedException {
        PulsarClient pulsarClient= GetPulsarClient.get();
        Producer<byte[]> simpleProducer=pulsarClient.newProducer()
            .topic(Global.defaultTopic)
            .producerName("simplePulsarProducer")
            .compressionType(CompressionType.SNAPPY)
            .sendTimeout(0, TimeUnit.SECONDS)
            .create();
        for(int i=0;i<100;i++){
            String msg="Hello simpleProducerSync!"+i;
            simpleProducer.send(msg.getBytes());
            System.out.println(msg);
            //Thread.sleep(100);
        }
    }
}

```

5 管控台使用

5.1 登录管理控制台

金蝶Apusic分布式消息队列管理控制台是一个基于Web的GUI管理和监控工具，用于管理用户、权限、租户、命名空间、主题、订阅、broker、部署集群。

1、在浏览器中键入URL：

<https://ip:12306>或者<http://ip:12305>

2、在登录界面输入用户名 (admq) ,密码(11111111)进行登录。



3、首次登录需要修改密码。



4、点击【点击前往】按钮，进入首次登录修改密码页面。



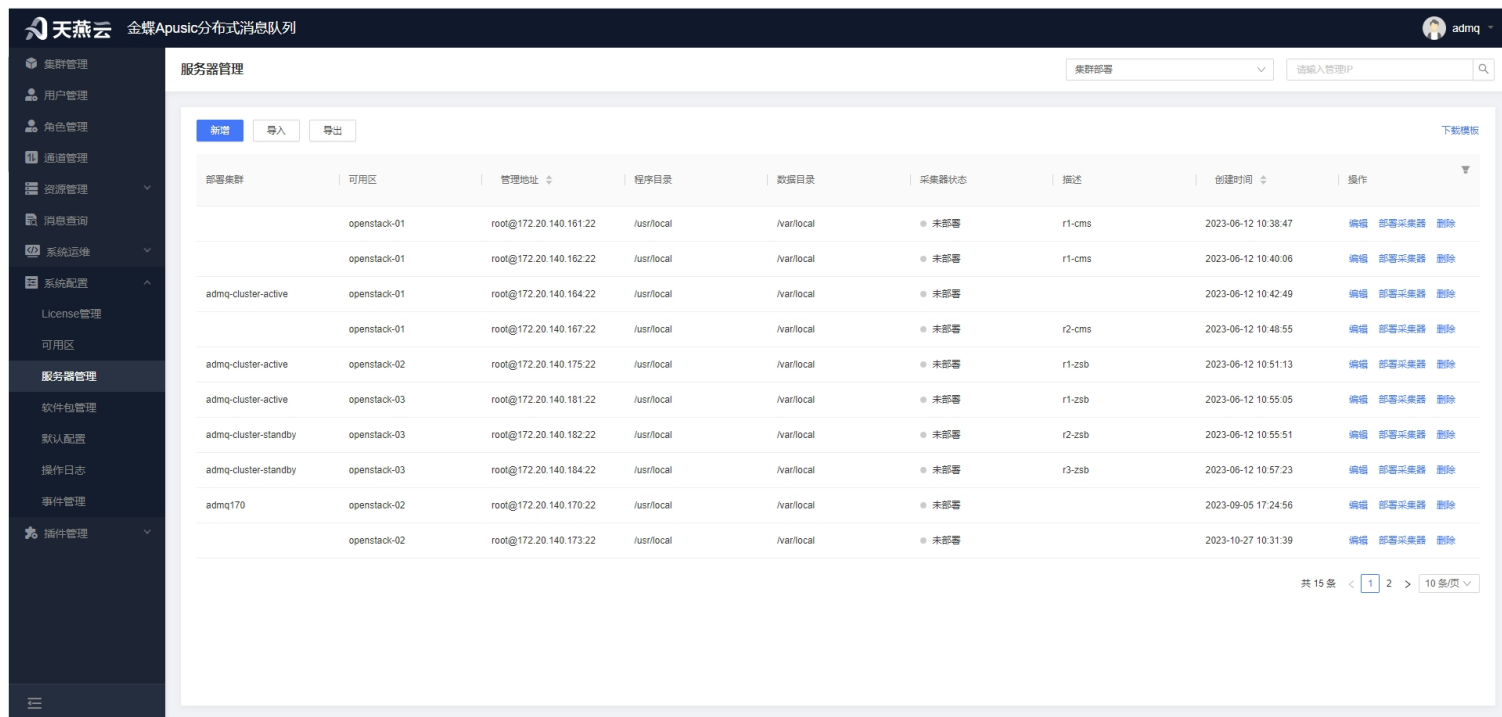
- 当前密码：输入当前密码
- 新密码：输入需要设置的新密码
- 确认密码：再次输入需要设置的新密码（两次密码需统一）

备注：ADMQ管理控制台集成了AUC和AMP，若想对集群节点进行监控，需通过AUC登录后注册服务跳转到ADMQ管控台界面后进行集群部署和监控运维。具体详见第10章监控运维。

5.2 系统配置

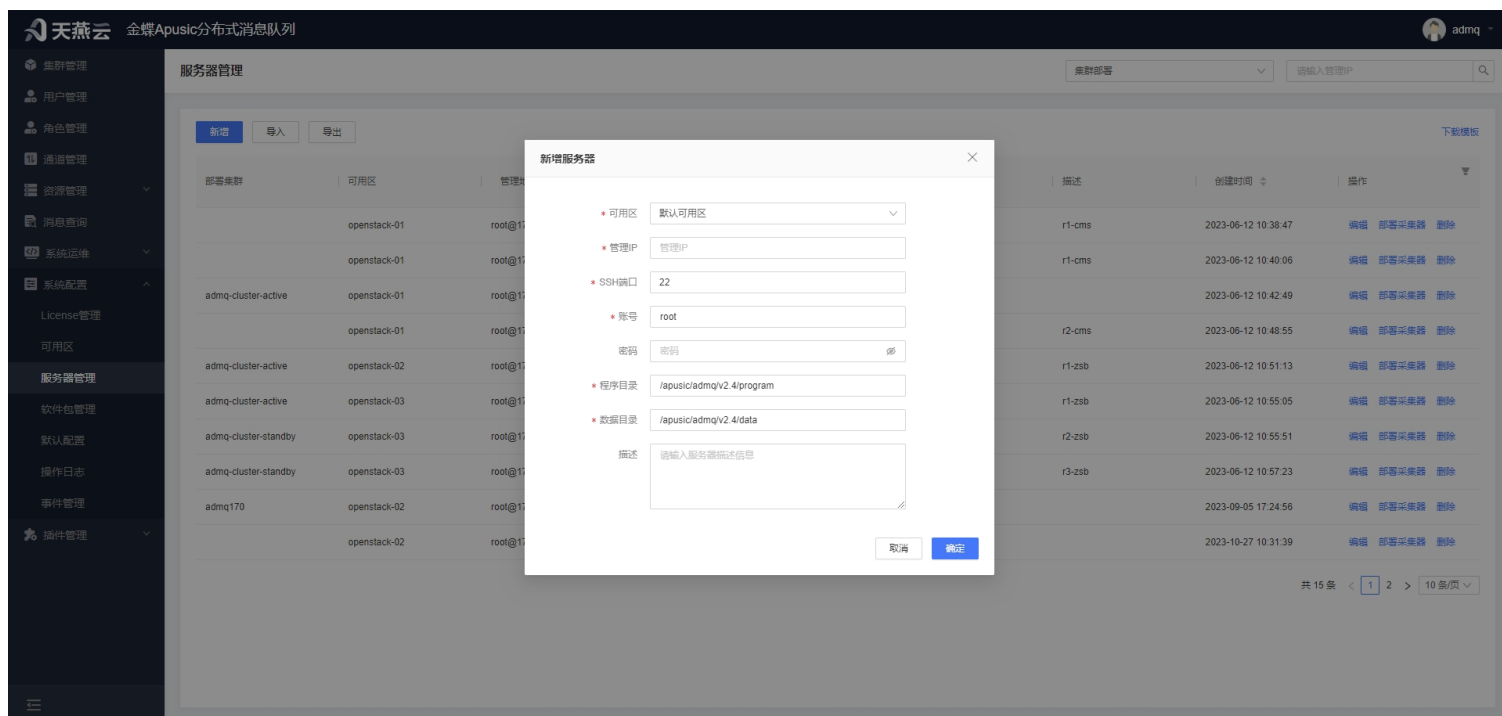
5.2.1 服务器管理

服务器管理是用于新增、编辑、删除、查询服务器信息以及部署采集器，作为创建集群信息时需被引用的基础信息。



5.2.1.1 新增服务器

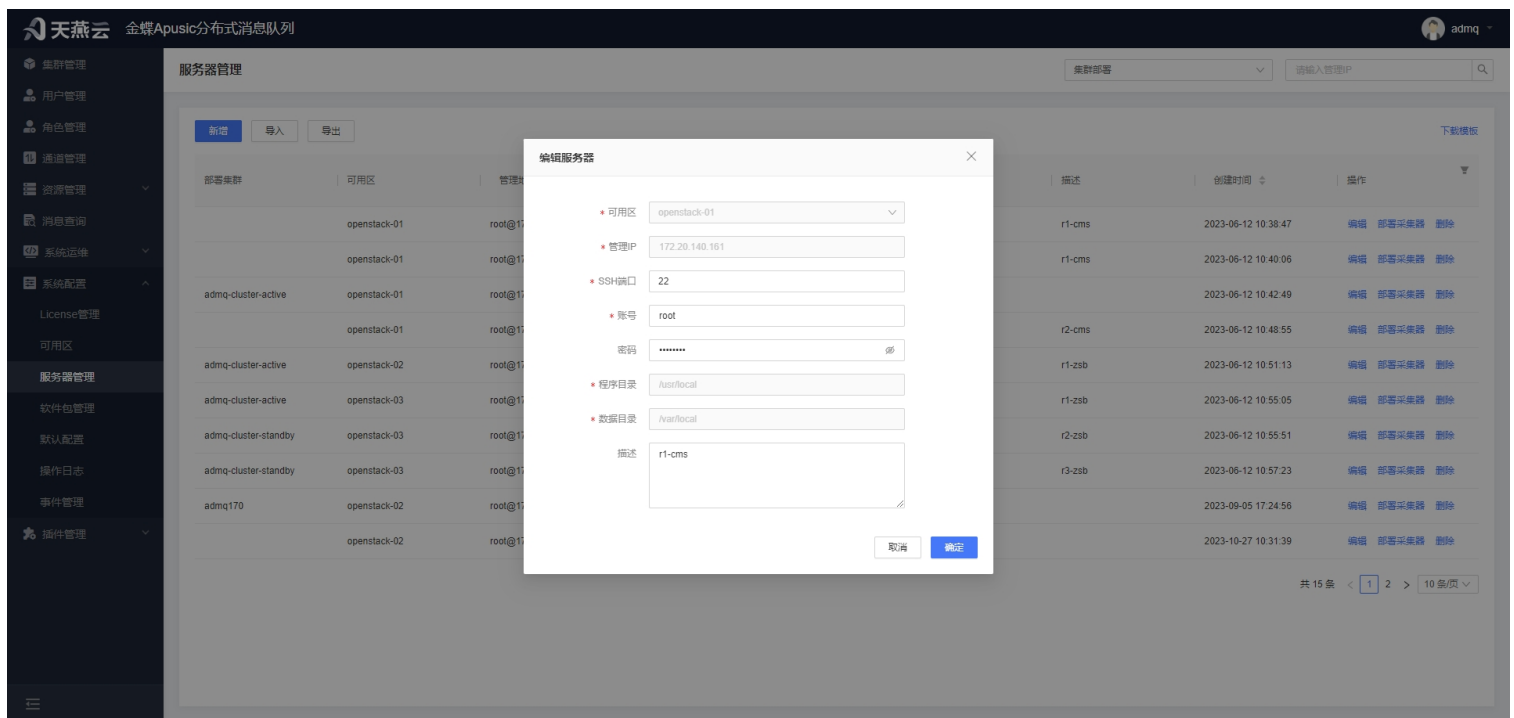
进入【系统配置】>【服务器管理】，在服务器管理页面点击左上角【新增】按钮，新增服务器信息。



- 机房：服务器所在的机房名称，自定义
- 管理IP：服务器IP地址
- ssh端口：服务器的端口，如22
- 账号：服务器的用户名，如root
- 密码：服务器的密码（如果不填则需要用户设置免密登录）
- 程序目录：服务器的集群节点信息存储目录
- 数据目录：服务器的集群节点信息数据存储目录
- 描述：服务器的详细信息

5.2.1.2 编辑服务器

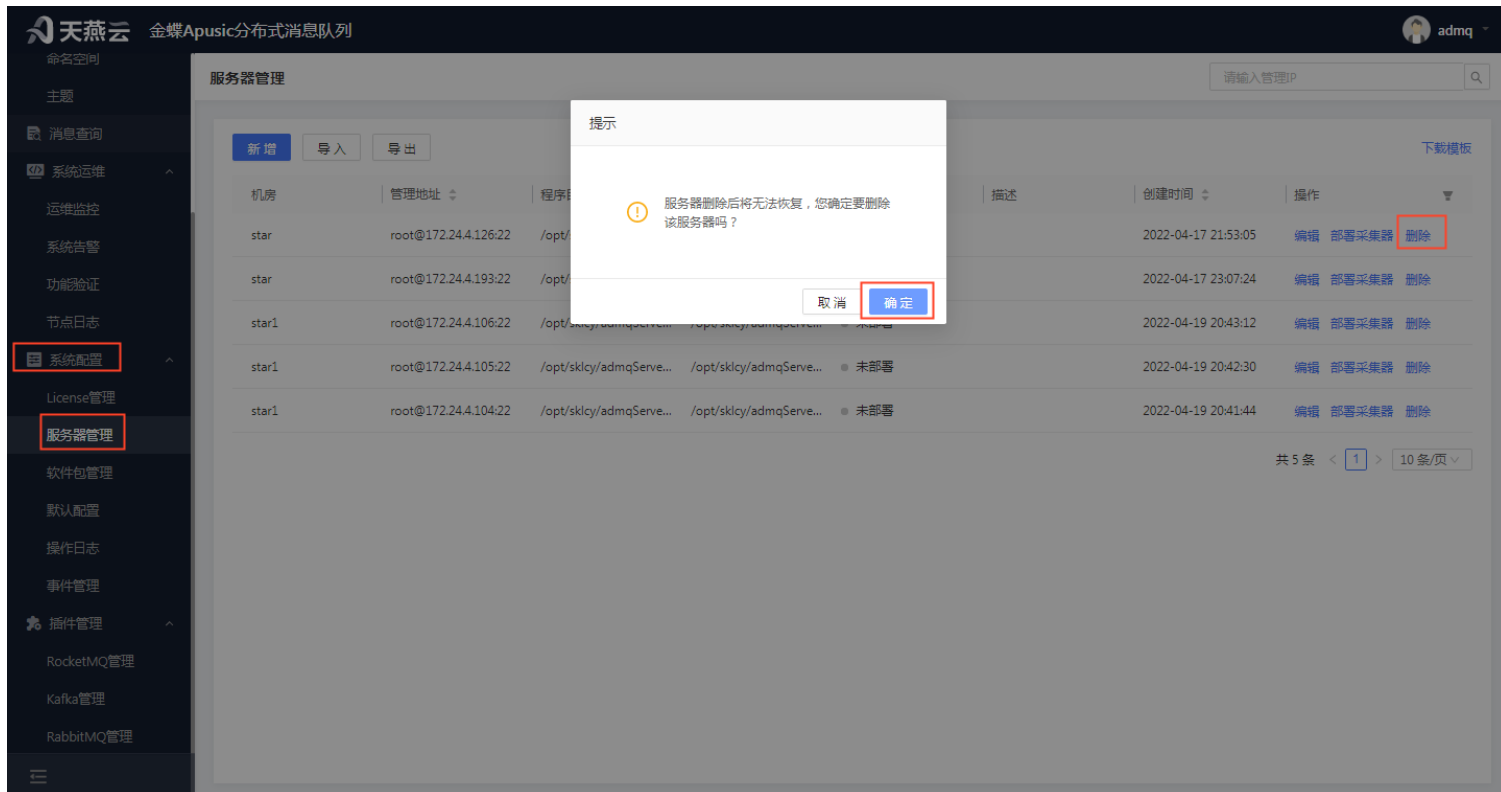
进入【系统配置】>【服务器管理】，在服务器管理页面选择需要编辑的服务器，点击【编辑】按钮，编辑服务器信息



- 机房：服务器所在的机房名称，不可修改
- 管理IP：服务器IP地址，不可修改
- ssh端口：服务器的端口，如22，可修改
- 账号：服务器的用户名，如root，可修改
- 密码：服务器的密码，可修改
- 程序目录：服务器的集群节点信息存储目录，不可修改
- 数据目录：服务器的集群节点信息数据存储目录，不可修改
- 描述：服务器的详细信息，可修改

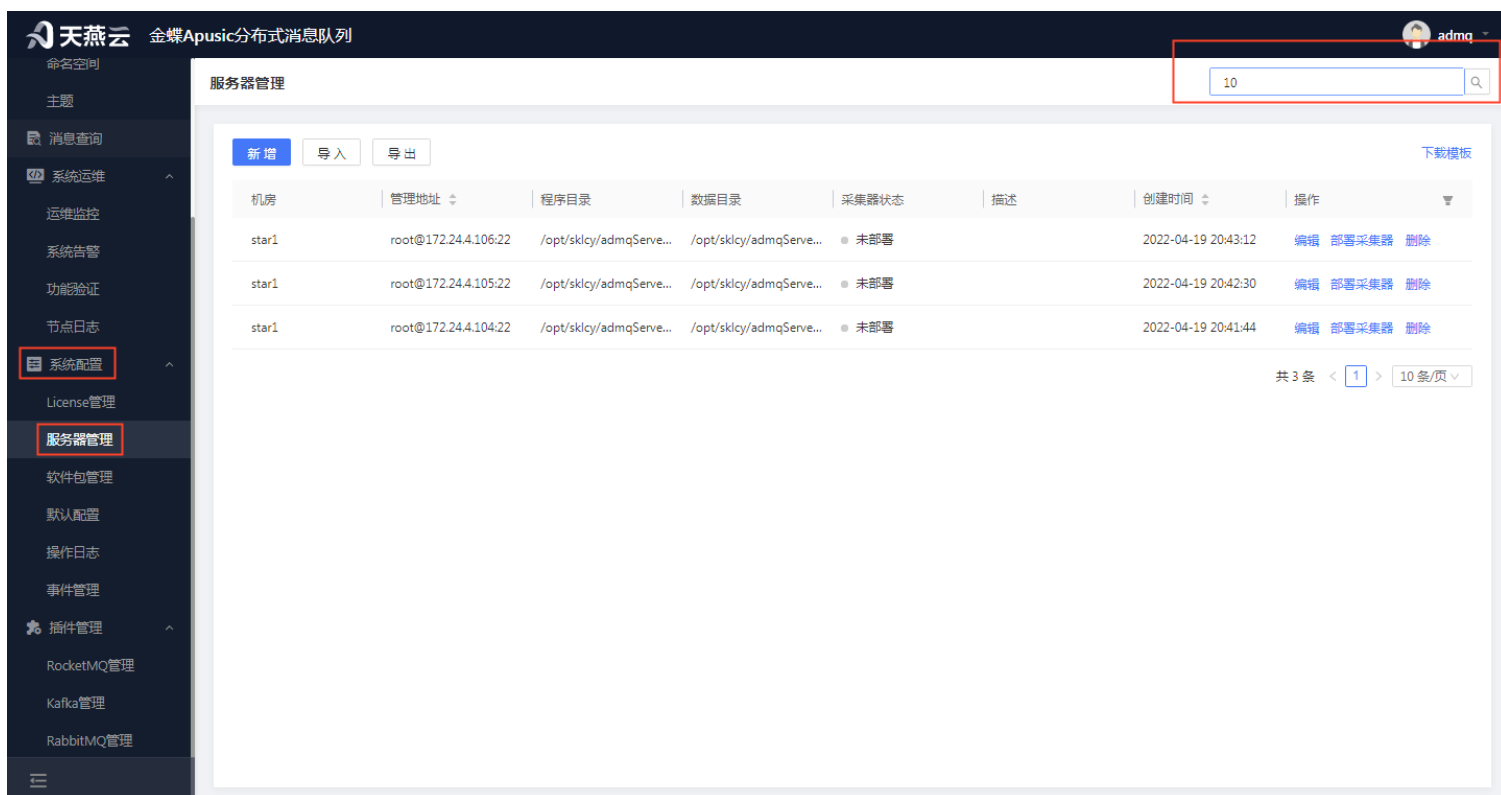
5.2.1.3 删除服务器

进入【系统配置】>【服务器管理】，进入服务器管理页面，选择需要删除的服务器信息，点击【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，删除服务器信息。注意：只有未被集群引用的服务器信息可以被删除，已被集群引用的服务器信息不可以被删除。



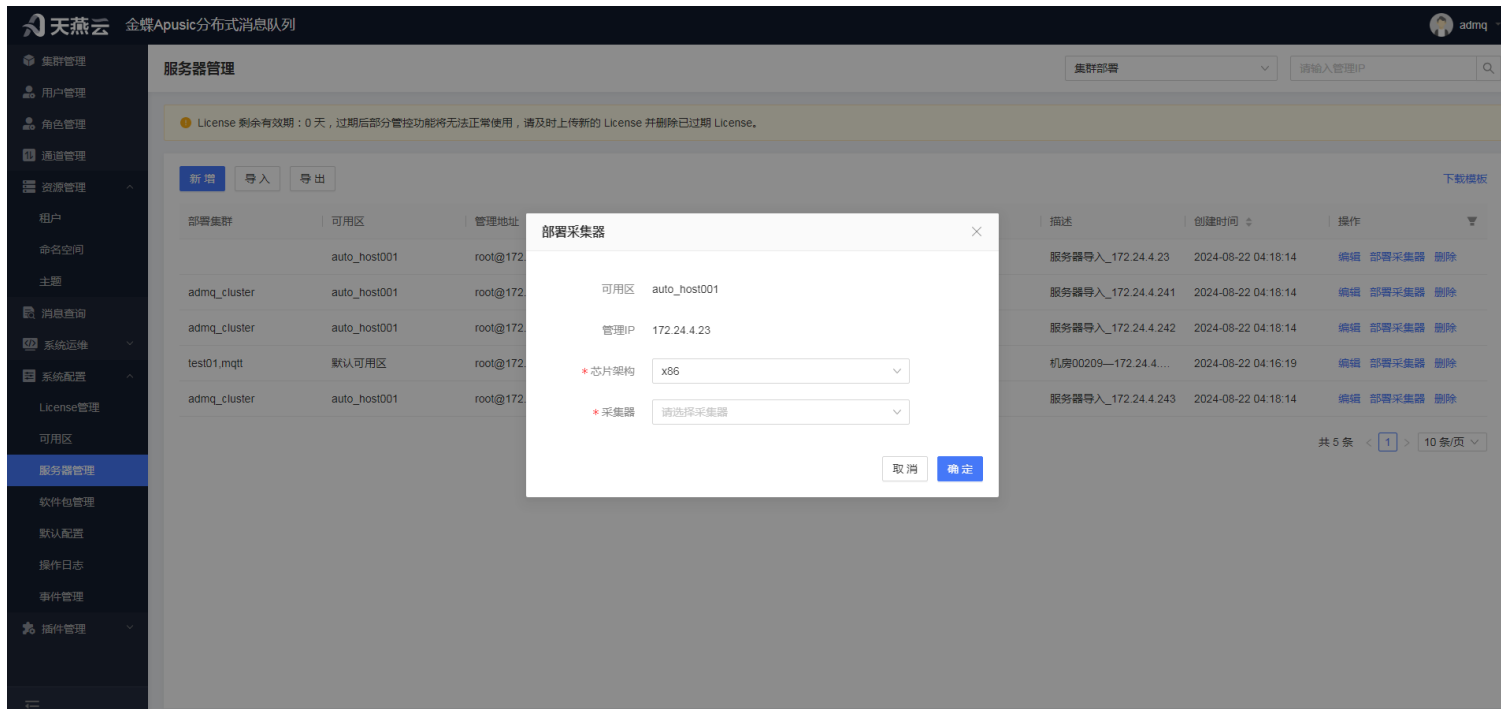
5.2.1.4 查询服务器

进入【系统配置】>【服务器管理】，进入服务器管理页面，模糊或精确输入管理IP，可查询服务器信息。



5.2.1.5 部署采集器

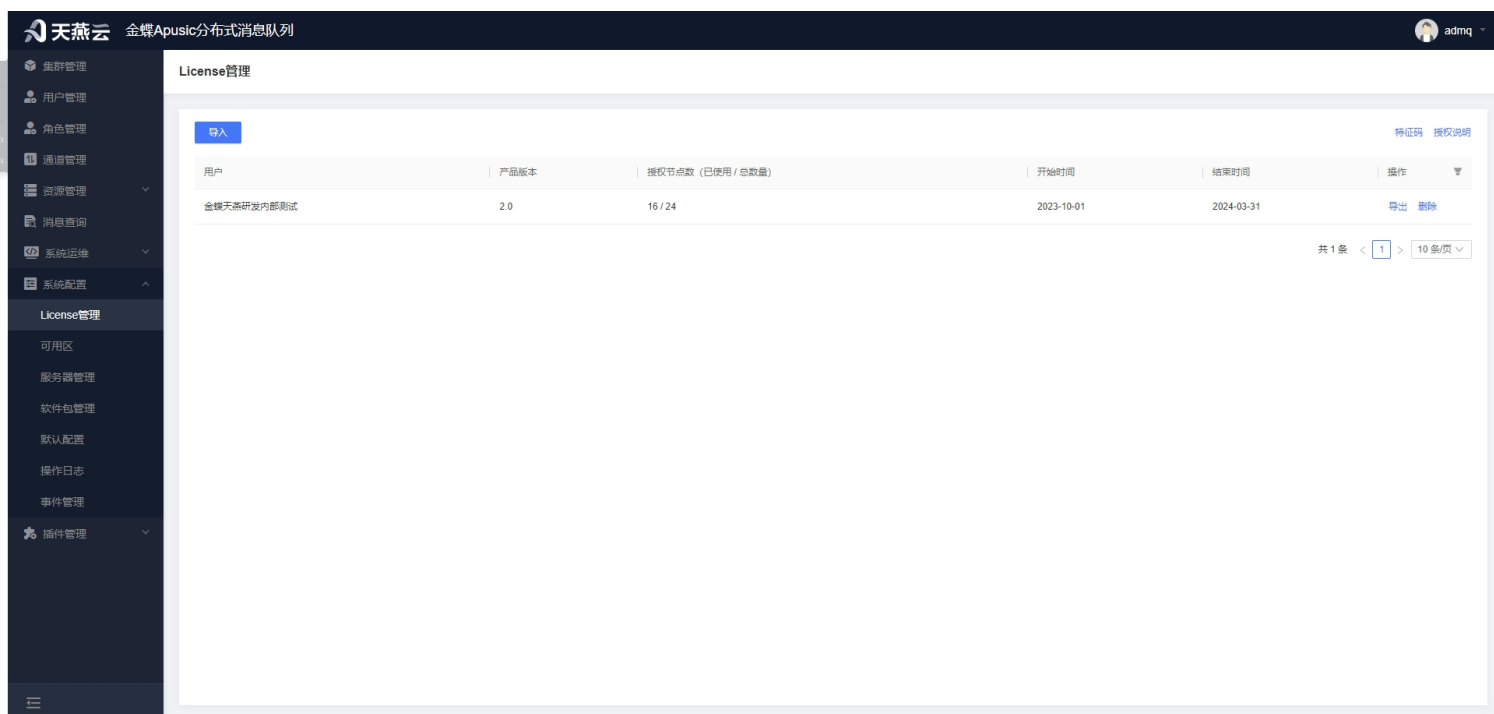
进入【系统配置】>【服务器管理】，进入服务器管理页面，选择需要部署采集器的服务器信息，点击【部署采集器】按钮，会弹出部署采集器弹框



- 可用区：所选服务器所在的可用区，不可变
- 管理IP：所选服务器的管理IP，不可变
- 芯片架构：下拉框，四种选项：arm、x86、risc-v、mips
- 采集器：下拉框，上传的软件包
- 备注：部署采集器前需要先上传采集器软件包，上传过程详见2.3.1。

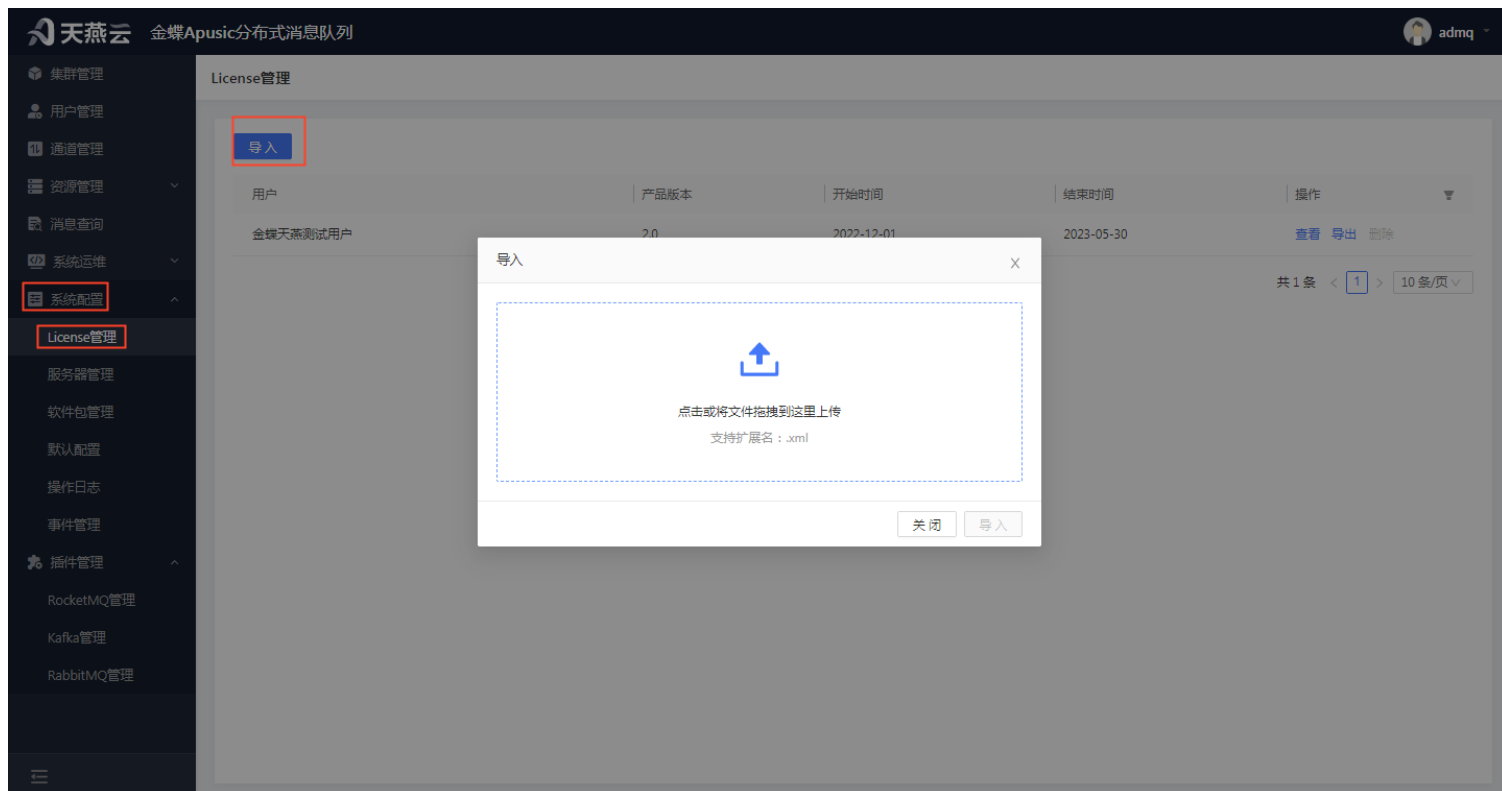
5.2.2 License管理

License管理是用于导入、导出、删除License信息，作为创建集群添加节点时需被引用的基础信息。



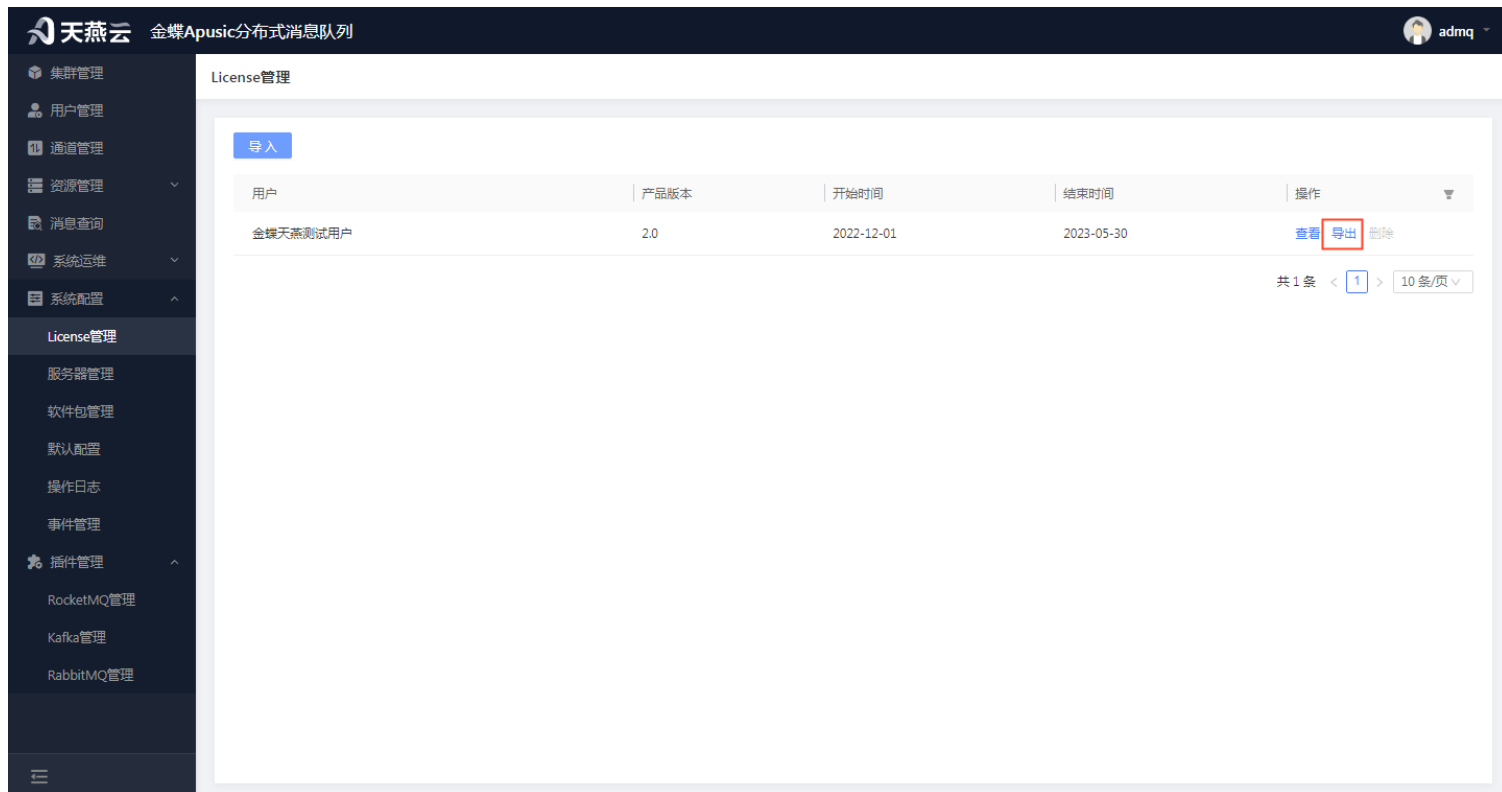
5.2.2.1 导入License

进入【系统配置】>【License管理】，点击左上角的【导入】按钮，点击【文件上传】按钮或拖动license文件进来，可导入License信息。注意：只能导入xml文件类型的license,导入license信息，最大计算节点数和最大存储节点数将会进行累加。



5.2.2.2 导出License

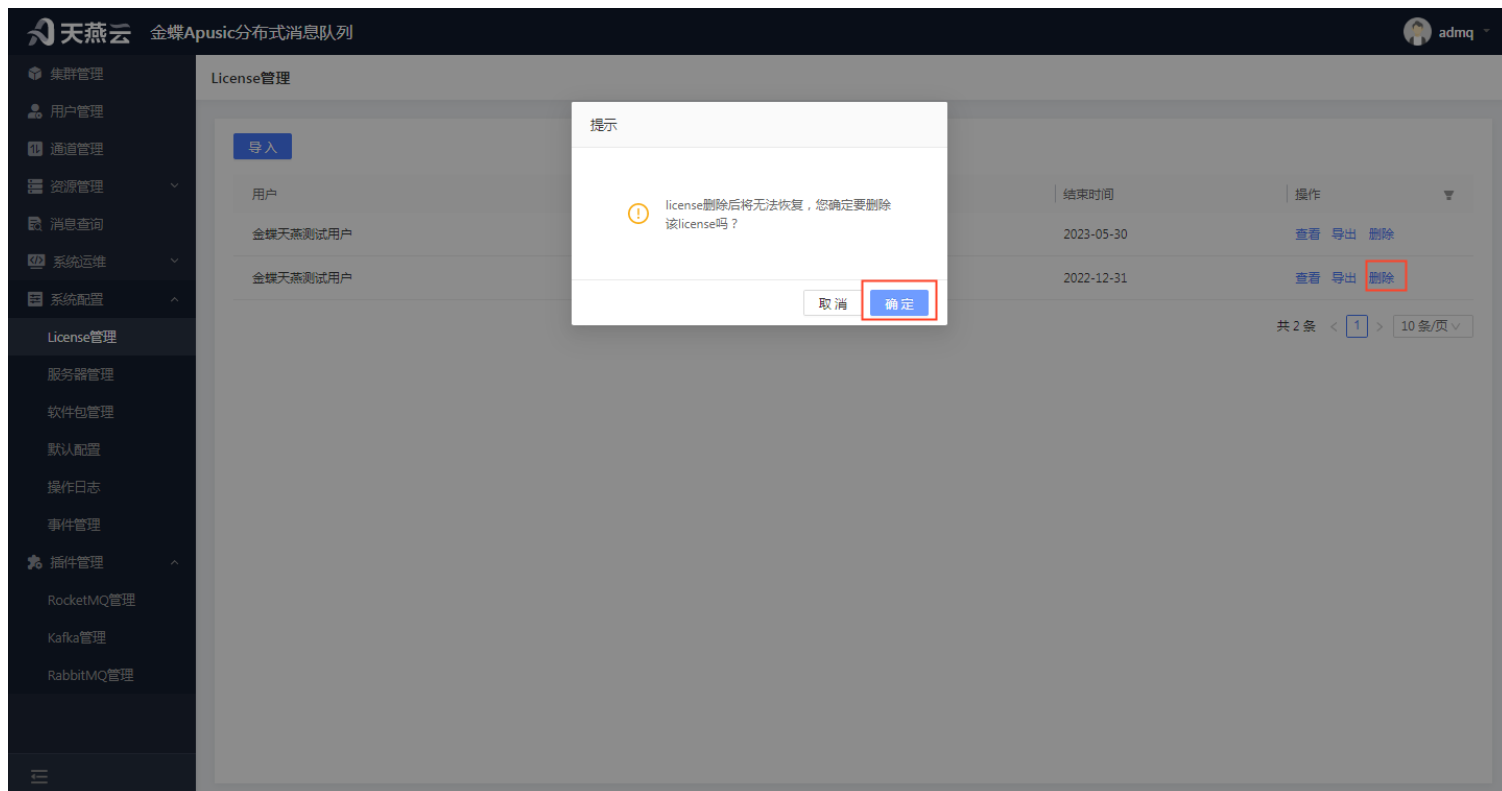
进入【系统配置】>【License管理】，选择需要导出的license点击【导出】按钮，可导出License信息。



备注：导出的License名称为导入时的license名称。

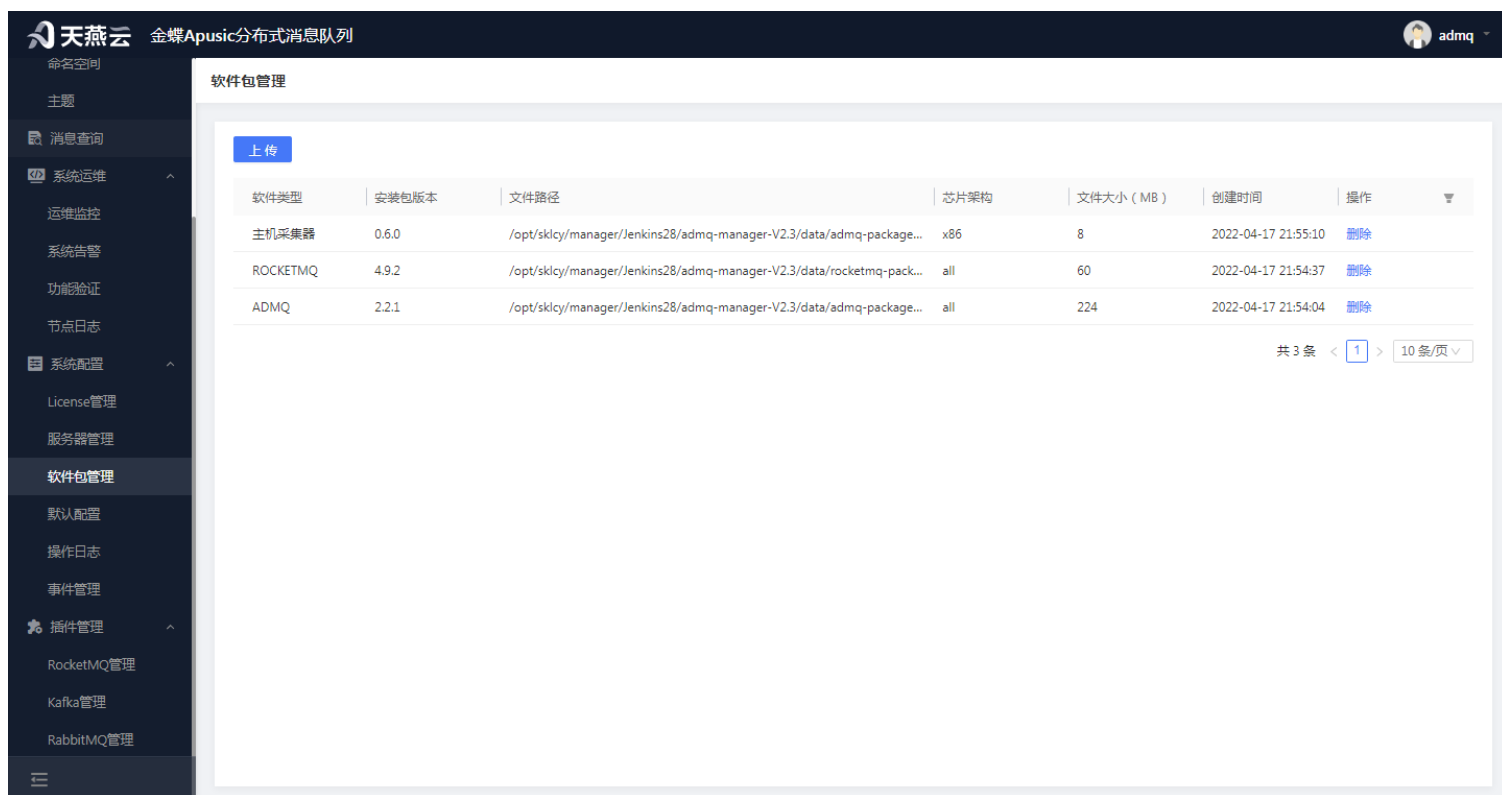
5.2.2.3 删除License

进入【系统配置】>【License管理】，选择需要删除的license，点击【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除License信息。注意：删除license后，对应可启动的broker数和可启动的bookkeeper数会对应减少数量。



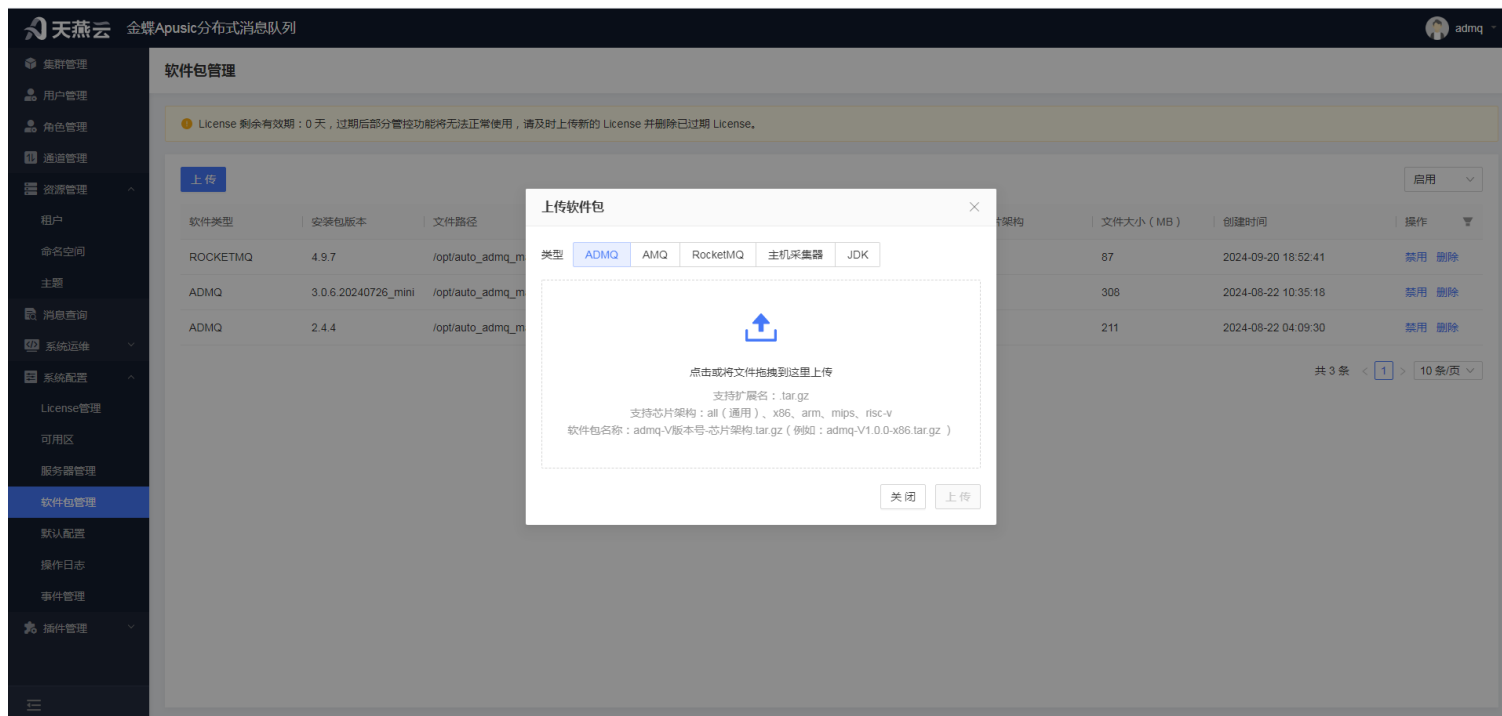
5.2.3 软件包管理

软件包管理是用于上传、删除软件包信息，作为新建集群、部署采集器时需被引用的基础信息。



5.2.3.1 上传软件包

进入【系统配置】>【软件包管理】，点击左上角的【上传】按钮，点击【上传】按钮或拖动软件包文件进来，可上传软件包信息。

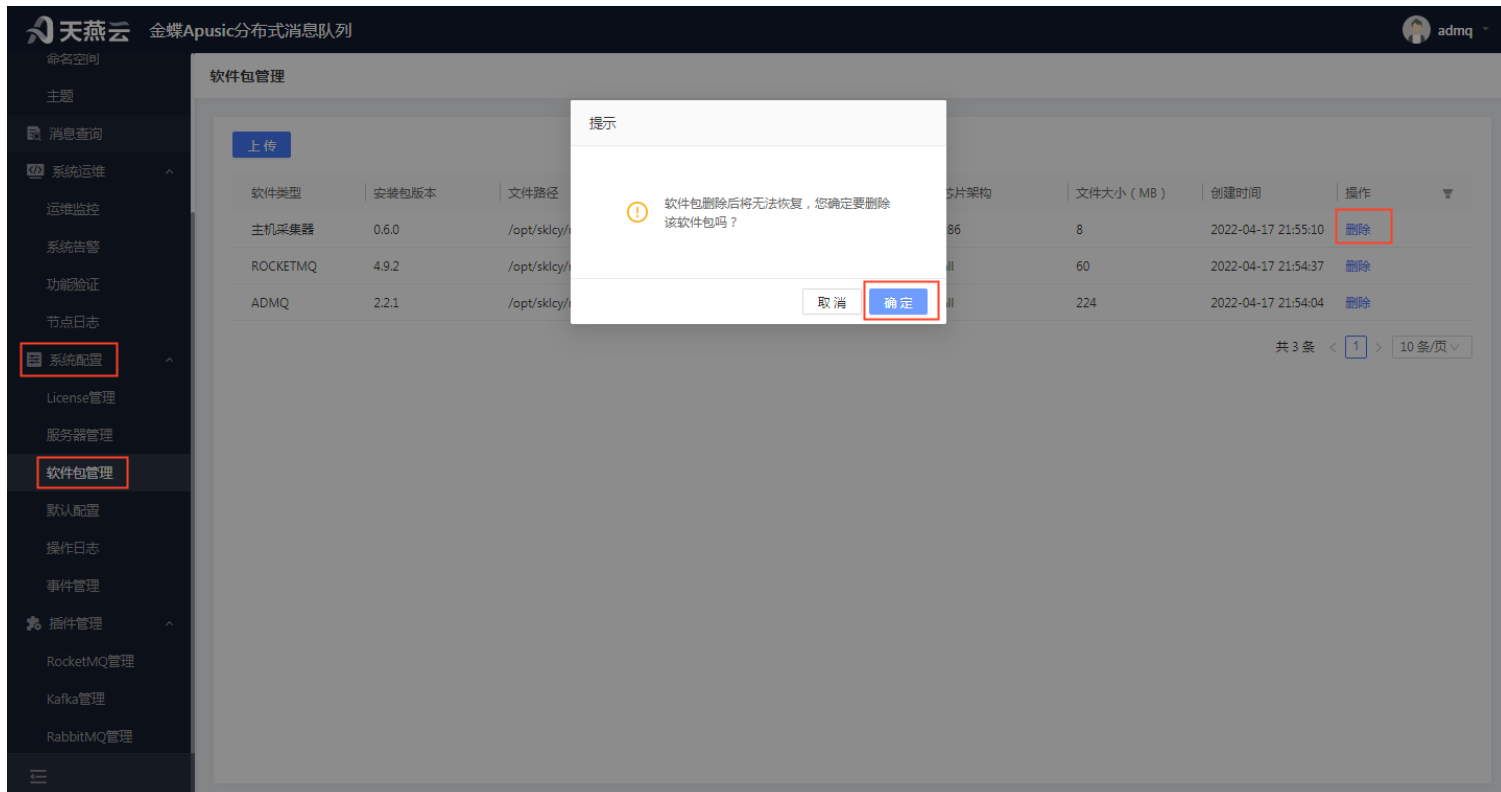


类型：ADMQ、AMQ、RocketMQ、主机采集器、JDK，单选

备注：可上传.tar.gz或者.zip的文件，软件包名称格式：xxx-V版本号(数字和小数点)-芯片架构.tar.gz（例如：admq-V1.0.0-x86.tar.gz）。

5.2.3.2 删除软件包

进入【系统配置】>【软件包管理】，选择需要删除的软件包，点击【删除】按钮，在弹出的确认删除弹框中点击【确认】按钮，可删除软件包信息。注意：当有集群信息引用此软件包或者服务器信息引用此软件包时，软件包不可进行删除，若要进行删除，需要先删除引用此软件包的集群信息或者服务器信息。

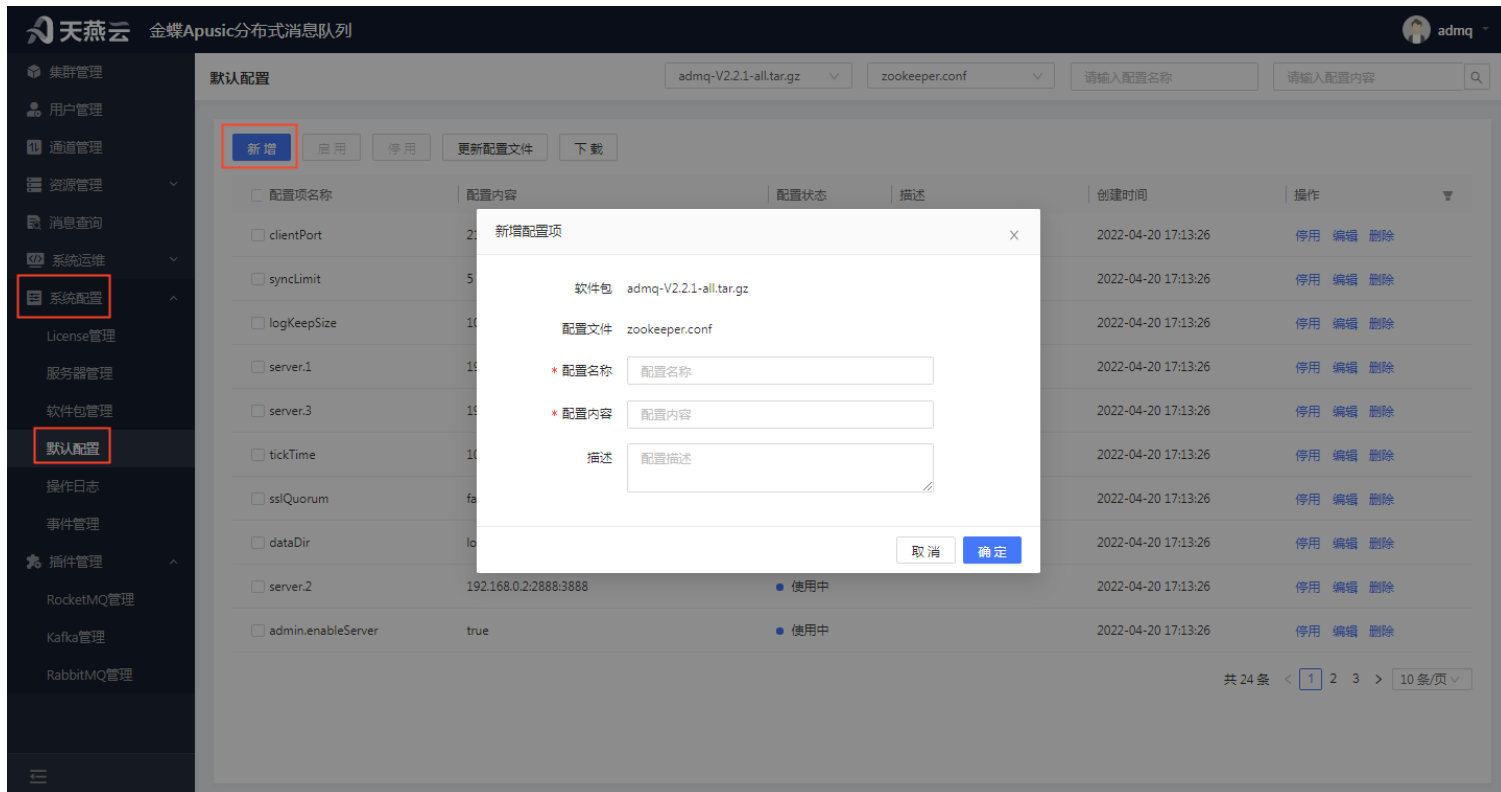


5.2.4 默认配置

默认配置信息是为了更方便的管理集群的默认配置文件，可新增默认配置信息、编辑默认配置信息、删除默认配置信息、更新默认配置信息、启用默认配置信息、停用默认配置信息、下载默认配置文件、查询配置信息。

5.2.4.1 新增配置信息

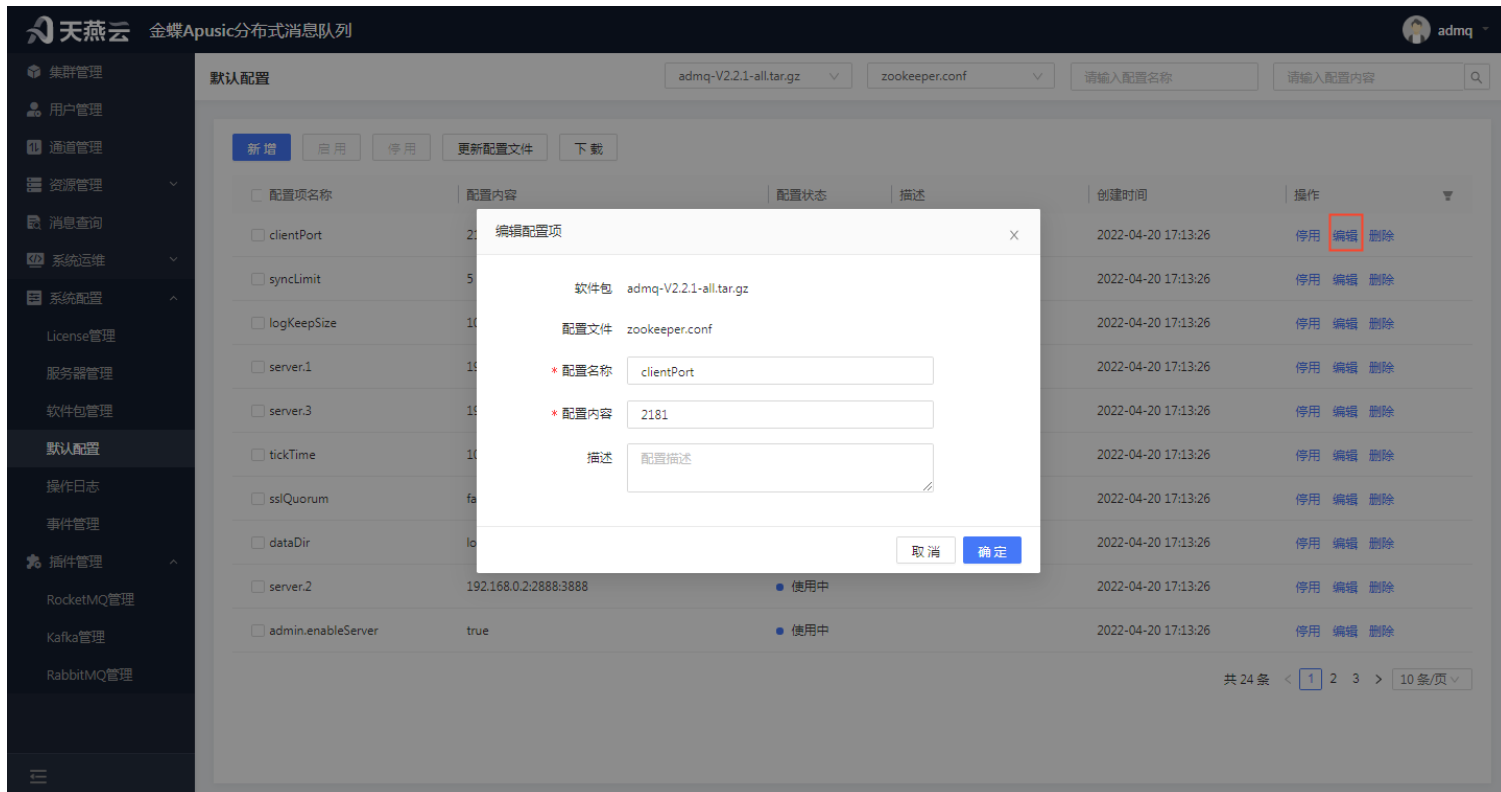
进入【系统配置】->【默认配置】，在默认配置页面，点击左上角【新增】按钮，可新增默认配置信息。注意：新增默认配置信息需要更新配置文件才会更新到物理配置文件，部署集群的时候节点配置才能复制参数。



- 软件包：软件包的名称，根据选择显示，不可修改，
- 配置文件：配置项所在配置文件名称，根据选择显示，不可修改
- 配置名称：配置项的名称，自定义，唯一标识
- 配置内容：配置项的值，自定义
- 描述：配置项的描述信息，自定义

5.2.4.2 修改配置信息

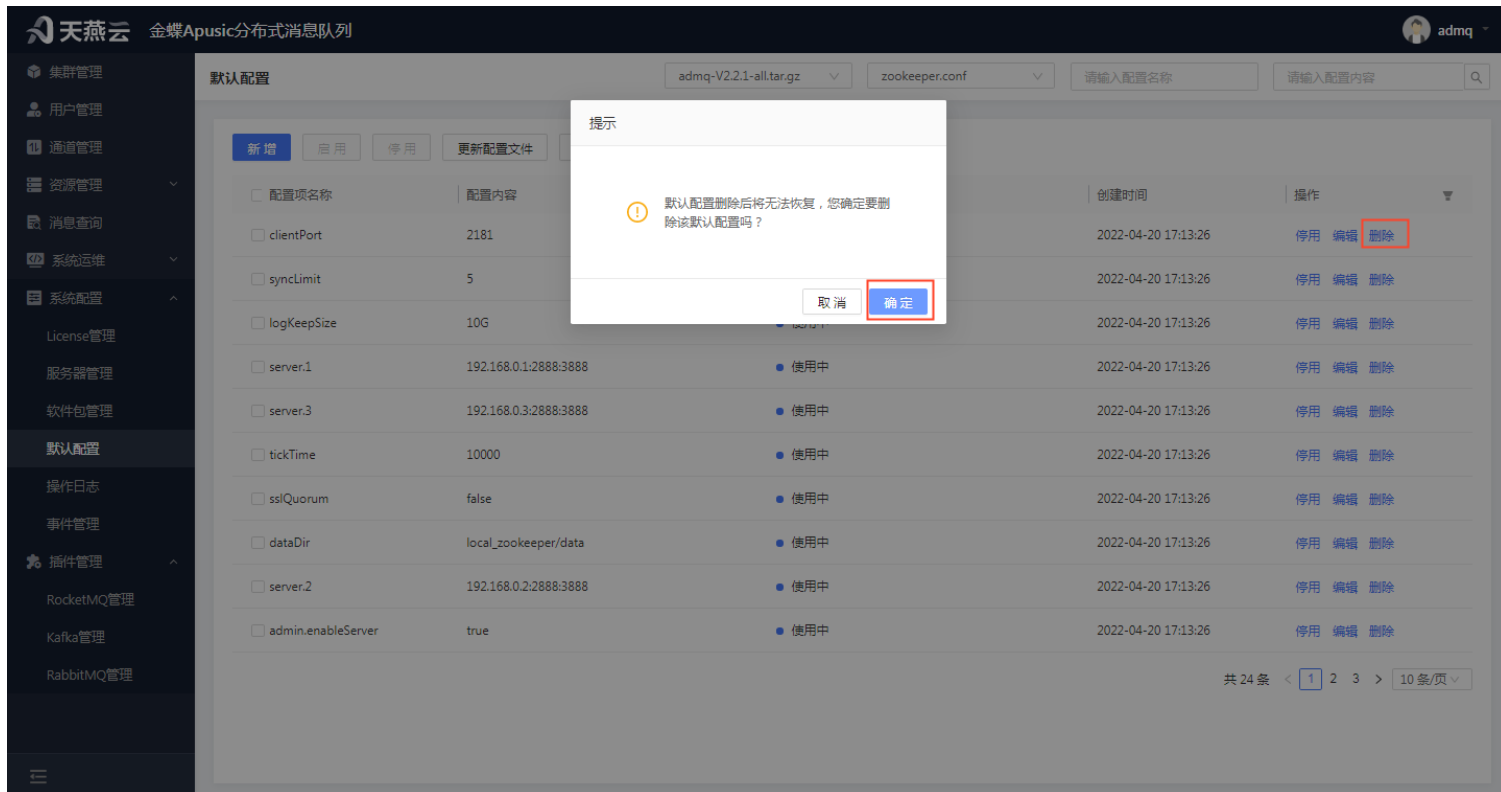
进入【系统配置】->【默认配置】，在默认配置页面，选择需要修改的配置，点击【编辑】按钮，可修改默认配置信息。注意：修改默认配置信息需要更新配置文件才会更新到物理配置文件，部署集群的时候节点配置才能复制参数。



- 软件包：软件包的名称，不可修改
- 配置文件：配置项所在配置文件名称，不可修改
- 配置名称：配置项的名称，可修改
- 配置内容：配置项的值，可修改
- 描述：配置项的描述信息，可修改

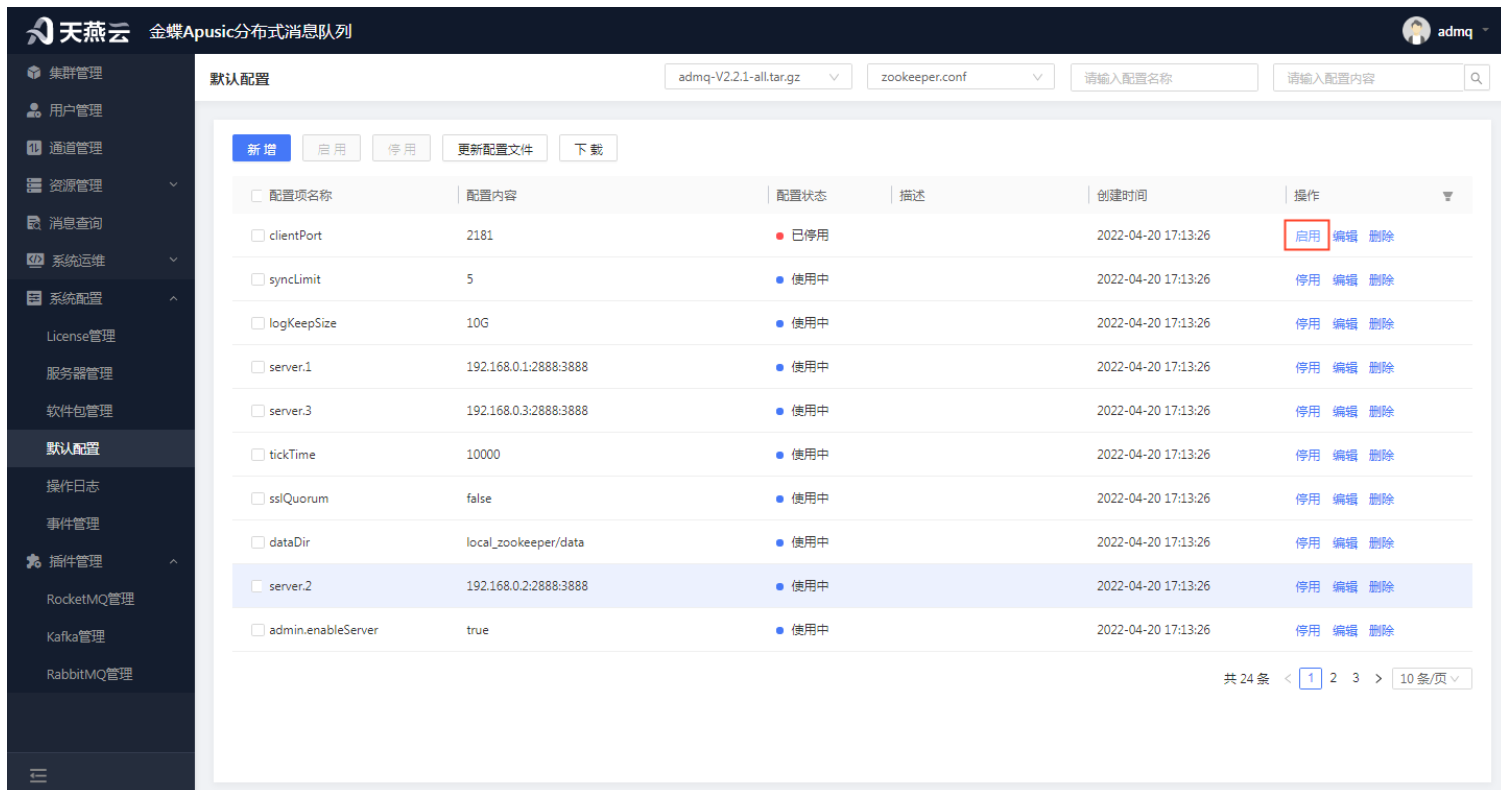
5.2.4.3 删除配置信息

进入【系统配置】->【默认配置】，在默认配置页面，选择需要删除的配置，点击【删除】按钮，在确认删除弹框中，点击【确定】按钮，可删除默认配置信息。注意：删除默认配置信息需要更新配置文件才会更新到物理配置文件，部署集群的时候节点配置才能生效；删除配置项需谨慎，避免因删除了重要的配置项而出现集群服务起不来或服务不可用。



5.2.4.4 启用配置参数

进入【系统配置】->【默认配置】，在默认配置页面，选择需要启用的配置（配置已停用），点击【启用】按钮，可启用默认配置信息。注意：启用默认配置信息需要更新配置文件才会更新到物理配置文件，部署集群的时候节点配置才能生效。



5.2.4.5 停用配置信息

进入【系统配置】->【默认配置】，在默认配置页面，选择需要停用的配置（配置已启用），点击【停用】按钮，可停用默认配置信息。注意：停用默认配置信息需要更新配置文件才会更新到物理配置文件，部署集群的时候节点配置才能生效（停用某些默认配置并且已更新配置文件，就表示配置文件中已删除这些配置）；停用配置项需谨慎，避免因停用了重要的配置项而出现集群服务起不来或服务不可用。

天燕云 金蝶Apusic分布式消息队列

默认配置

adm-q-V2.2.1-all.tar.gz | zookeeper.conf | 请输入配置名称 | 请输入配置内容

新增 启用 停用 更新配置文件 下载

配置项名称	配置内容	配置状态	描述	创建时间	操作
clientPort	2181	使用中		2022-04-20 17:13:26	停用 编辑 删除
syncLimit	5	使用中		2022-04-20 17:13:26	停用 编辑 删除
logKeepSize	10G	使用中		2022-04-20 17:13:26	停用 编辑 删除
server.1	192.168.0.1:2888:3888	使用中		2022-04-20 17:13:26	停用 编辑 删除
server.3	192.168.0.3:2888:3888	使用中		2022-04-20 17:13:26	停用 编辑 删除
tickTime	10000	使用中		2022-04-20 17:13:26	停用 编辑 删除
sslQuorum	false	使用中		2022-04-20 17:13:26	停用 编辑 删除
dataDir	local_zookeeper/data	使用中		2022-04-20 17:13:26	停用 编辑 删除
server.2	192.168.0.2:2888:3888	使用中		2022-04-20 17:13:26	停用 编辑 删除
admin.enableServer	true	使用中		2022-04-20 17:13:26	停用 编辑 删除

共 24 条 < 1 2 3 > 10 条/页

5.2.4.6 下载配置文件

进入【系统配置】->【默认配置】，在默认配置页面，右上角选择需要下载的配置项名称，点击【下载】按钮，可下载默认配置文件。

天燕云 金蝶Apusic分布式消息队列

默认配置

adm-q-V2.2.1-all.tar.gz | zookeeper.conf | 请输入配置名称 | 请输入配置内容

新增 启用 停用 更新配置文件 下载

配置项名称	配置内容	配置状态	描述	创建时间	操作
clientPort	2181	使用中		2022-04-20 17:13:26	停用 编辑 删除
syncLimit	5	使用中		2022-04-20 17:13:26	停用 编辑 删除
logKeepSize	10G	使用中		2022-04-20 17:13:26	停用 编辑 删除
server.1	192.168.0.1:2888:3888	使用中		2022-04-20 17:13:26	停用 编辑 删除
server.3	192.168.0.3:2888:3888	使用中		2022-04-20 17:13:26	停用 编辑 删除
tickTime	10000	使用中		2022-04-20 17:13:26	停用 编辑 删除
sslQuorum	false	使用中		2022-04-20 17:13:26	停用 编辑 删除
dataDir	local_zookeeper/data	使用中		2022-04-20 17:13:26	停用 编辑 删除
server.2	192.168.0.2:2888:3888	使用中		2022-04-20 17:13:26	停用 编辑 删除
admin.enableServer	true	使用中		2022-04-20 17:13:26	停用 编辑 删除

共 24 条 < 1 2 3 > 10 条/页

5.2.4.7 查询配置项信息

进入【系统配置】->【默认配置】，在默认配置页面，右上角选择软件包名称、配置文件名称，精确或模糊输入配置名称、配置内容，可查询默认配置信息。备注：配置名称、配置内容两者选其一也可进行查询。

天燕云 金蝶Apusic分布式消息队列 admq

默认配置

adm-q-V2.2.1-all.tar.gz zookeeper.conf cl 请输入配置内容

新增 启用 停用 更新配置文件 下载

配置项名称	配置内容	配置状态	描述	创建时间	操作
clientPort	2181	使用中		2022-04-20 17:13:26	停用 编辑 删除
syncLimit	5	使用中		2022-04-20 17:13:26	停用 编辑 删除
metricsProvider.className	org.apache.zookeeper.metrics.prometheus.Prometh...	使用中		2022-04-20 17:13:26	停用 编辑 删除
maxClientCnxns	300	使用中		2022-04-20 17:13:26	停用 编辑 删除

共 4 条 < 1 > 10 条/页

5.2.5 操作日志

操作日志用于查询用户的操作纪录信息。

天燕云 金蝶Apusic分布式消息队列

操作日志

请选择模块 | 请输入用户名称 | 2022-12-22 → 2022-12-23 | 刷新

用户	操作IP	模块	事件	操作结果	操作时间
admq	172.24.6.60	系统配置-配置文件管理	将配置项clientPort的状态修改为已生效,所属软件包:admq-V2.2.1-all.tar.gz,节点类型:admq协调器	成功	2022-12-23 23:34:23
admq	172.24.6.60	系统配置-配置文件管理	将配置项clientPort的状态修改为未生效,所属软件包:admq-V2.2.1-all.tar.gz,节点类型:admq协调器	成功	2022-12-23 23:32:04
admq	172.24.6.60	系统配置-LICENSE管理	删除证书文件	成功	2022-12-23 23:27:08
admq	172.24.6.60	系统配置-LICENSE管理	上传证书	成功	2022-12-23 23:25:19
admq	172.24.6.60	用户管理-登录登出	登录	成功	2022-12-23 22:27:48
admq	172.24.6.60	系统配置-服务器管理	对集群admq_cluster126在服务器172.24.4.126上的单机程序发送了命令stop	成功	2022-12-23 19:30:13
admq	172.24.6.60	集群管理-集群管理	将集群admq_cluster126的单机程序配置文件刷新到远程服务器172.24.4.126	成功	2022-12-23 19:30:09
admq	172.24.6.60	集群管理-集群管理	修改了配置项:messagingProtocols, 值:rocketmq,所属集群:admq_cluster126,所在服务器:172.24.4.126,节点类型:单机程序	成功	2022-12-23 19:30:07
admq	172.24.6.60	用户管理-登录登出	登录	成功	2022-12-23 19:28:18
admq	172.24.6.60	系统配置-服务器管理	对集群admq_cluster126在服务器172.24.4.126上的单机程序发送了命令start	成功	2022-12-23 18:04:20

共 30 条 < 1 2 3 > 10 条/页

5.2.5.1 操作日志查询

进入【系统配置】>【操作日志】，进入操作日志页面，下拉选择模块、输入用户名称，选择查询开始日期和结束日期，可查询操作日志信息。

天燕云 金蝶Apusic分布式消息队列

操作日志

集群管理-集群管理 | admq | 2022-12-22 → 2022-12-23 | 刷新

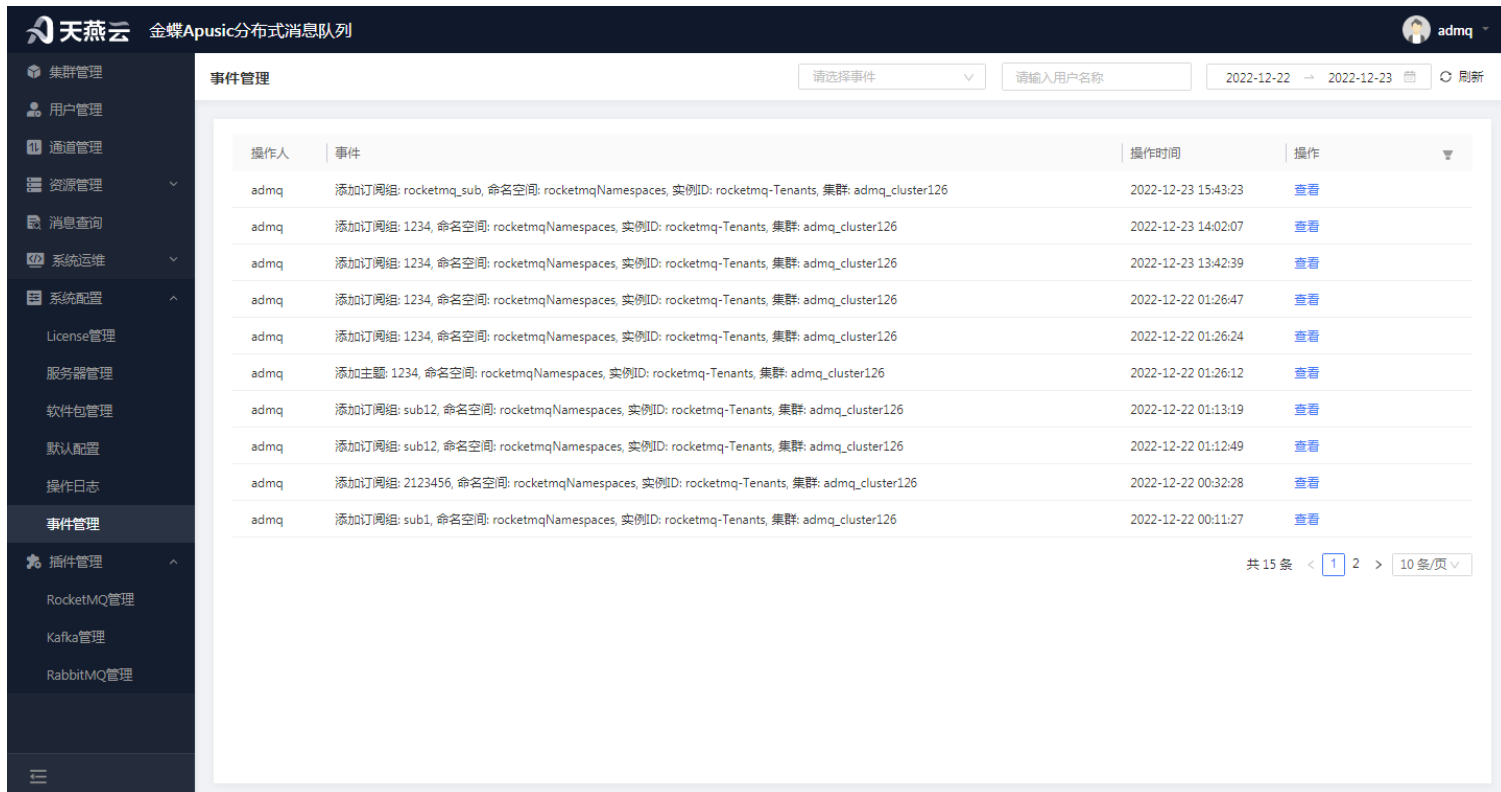
用户	操作IP	模块	事件	操作结果	操作时间
admq	172.24.6.60	集群管理-集群管理	将集群admq_cluster126的单机程序配置文件刷新到远程服务器172.24.4.126	成功	2022-12-23 19:30:09
admq	172.24.6.60	集群管理-集群管理	修改了配置项:messagingProtocols, 值:rocketmq,所属集群:admq_cluster126,所在服务器:172.24.4.126,节点类型:单机程序	成功	2022-12-23 19:30:07
admq	172.24.6.60	集群管理-集群管理	将集群admq_cluster126的单机程序配置文件刷新到远程服务器172.24.4.126	成功	2022-12-23 18:02:46
admq	172.24.6.60	集群管理-集群管理	新增了配置项:mq.rocket.monitor.enabled, 值:true,所属集群:admq_cluster126,所在服务器:172.24.4.126,节点类型:单机程序	成功	2022-12-23 18:02:34

共 4 条 < 1 > 10 条/页

备注：超级管理员用户可查询到所有用户的操作日志信息，普通用户只能查询自己的操作日志信息。

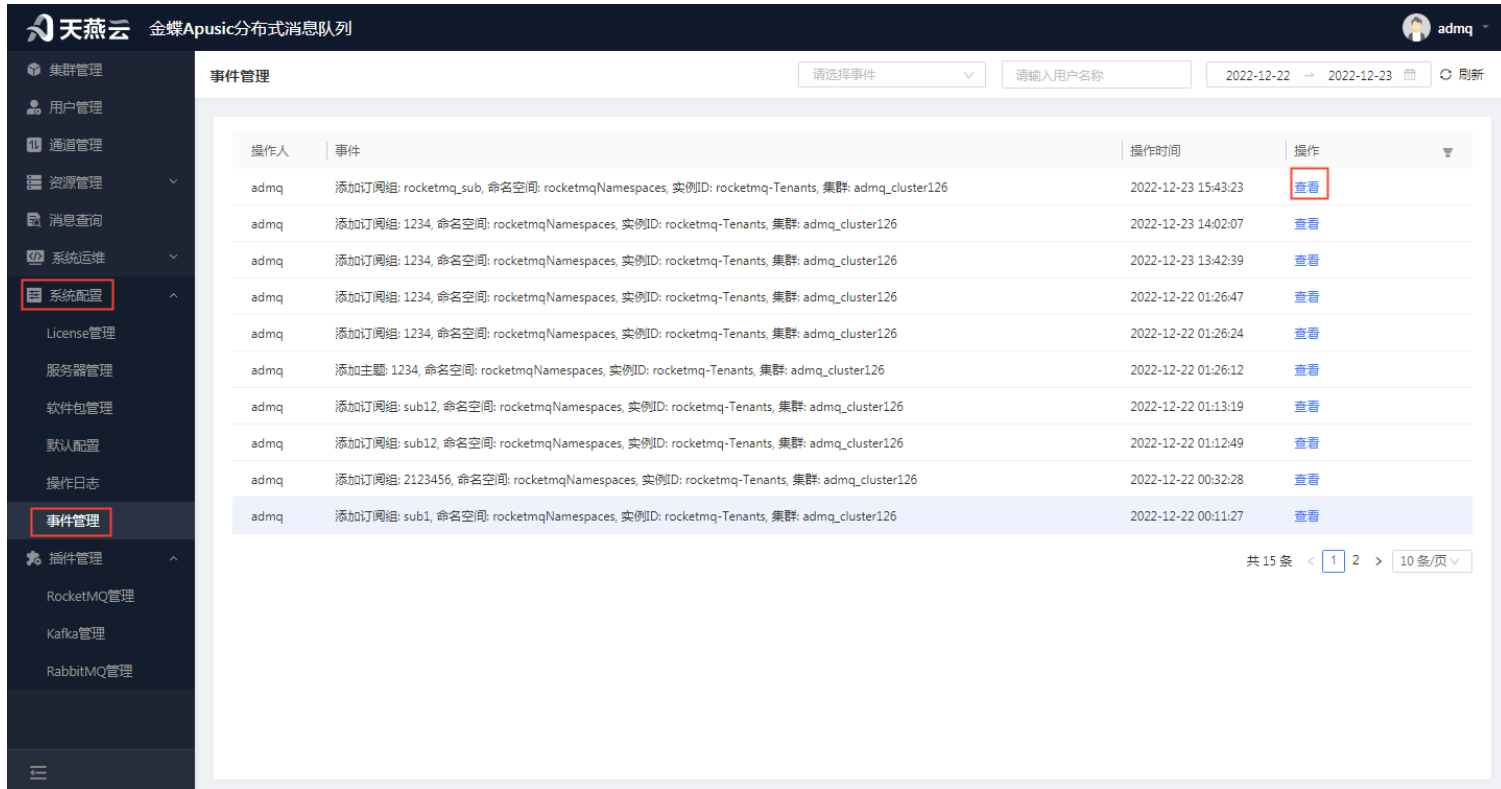
5.2.6 事件管理

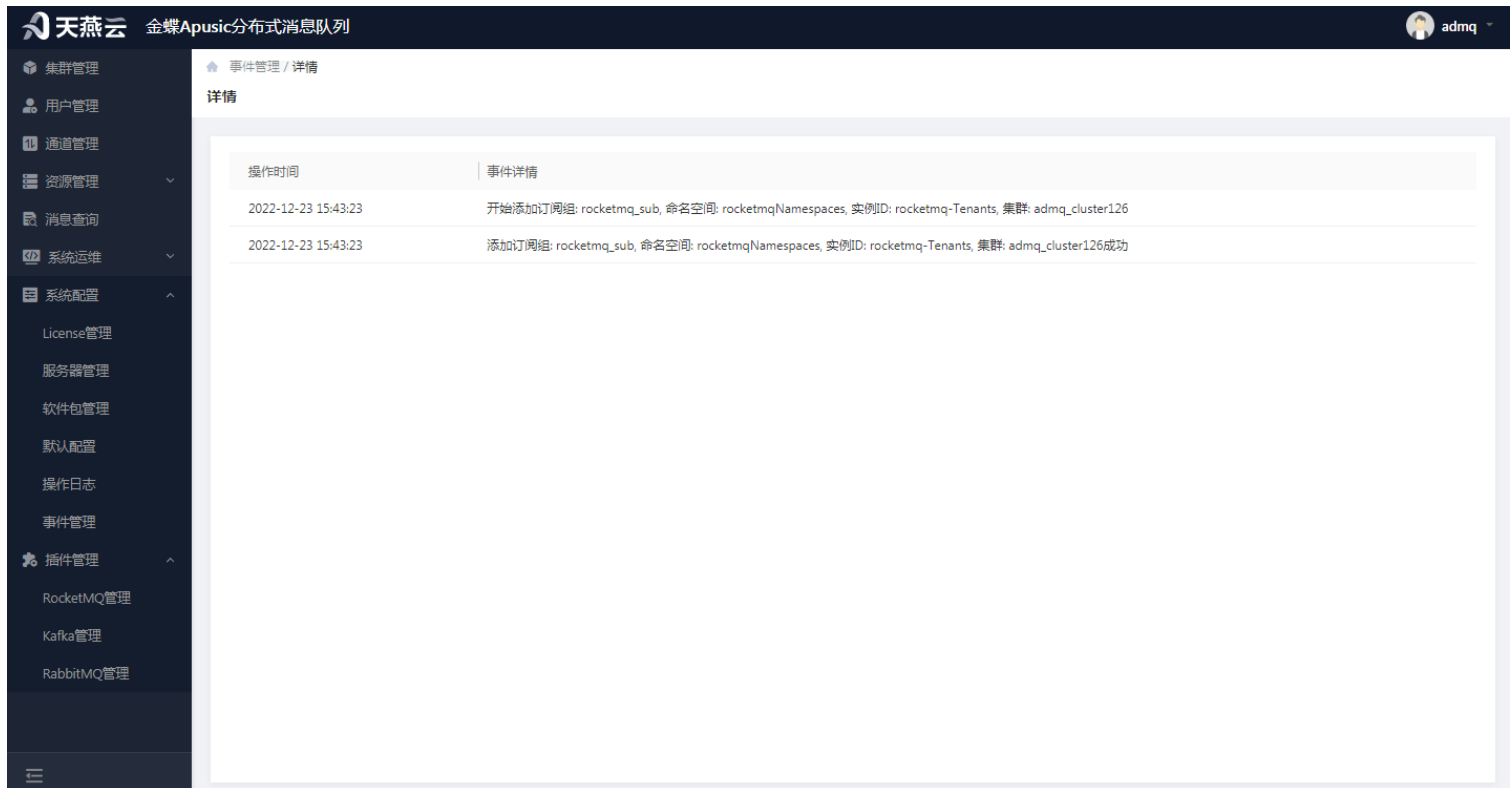
管理控制台支持对事件进行管理，可跟踪资源等系统事件情况。事件管理可查看事件详情信息、可查询事件信息。



5.2.6.1 查看事件详细信息

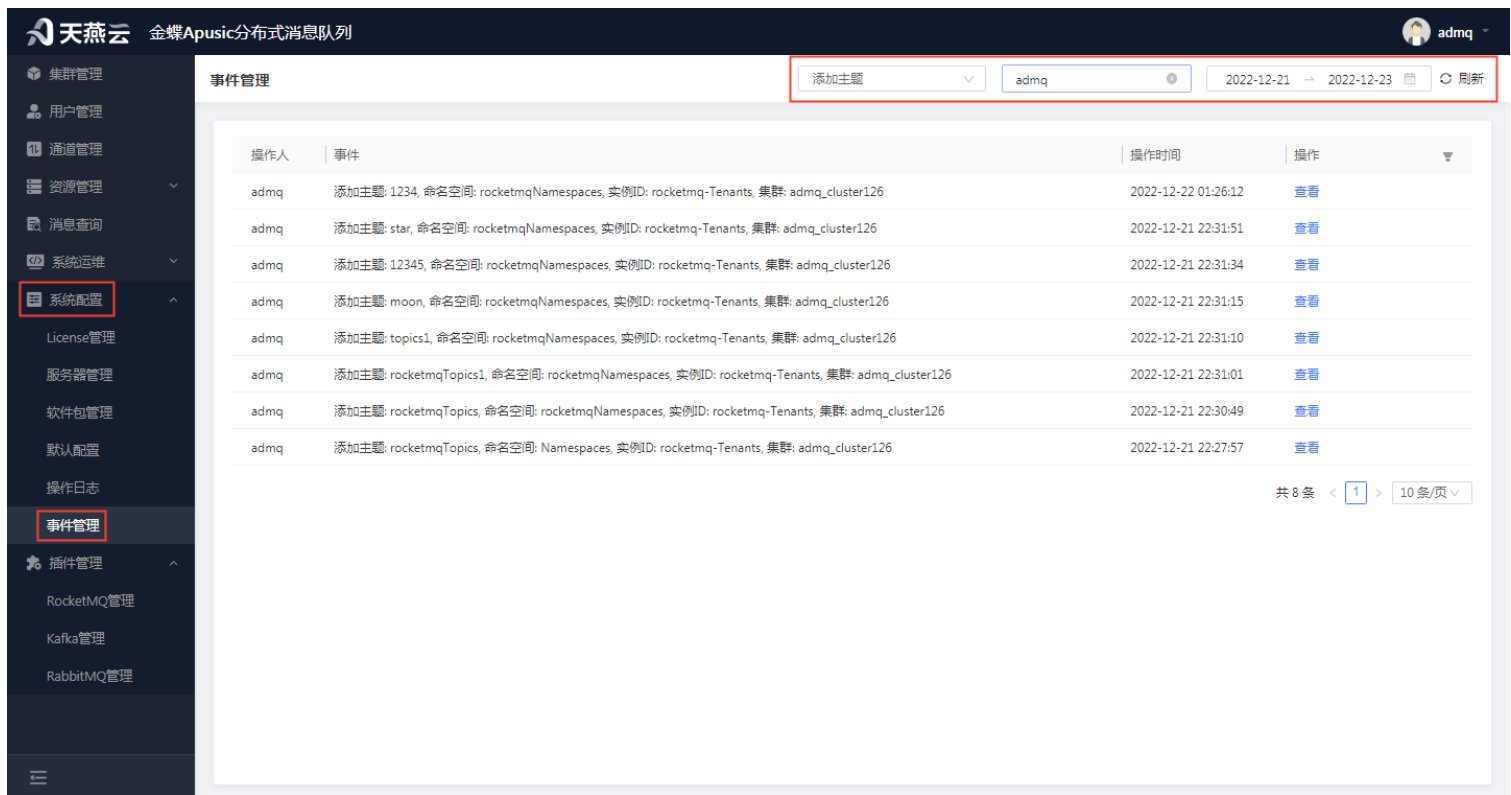
进入【系统配置】>【事件管理】，进入事件管理页面，选择要查看详情的事件，点击【查看】按钮，可查看事件详情信息。





5.2.6.2 查询事件信息

进入【系统配置】>【事件管理】，进入事件管理页面，下拉选择事件、输入用户名称，选择查询开始日期和结束日期，可查询事件信息。

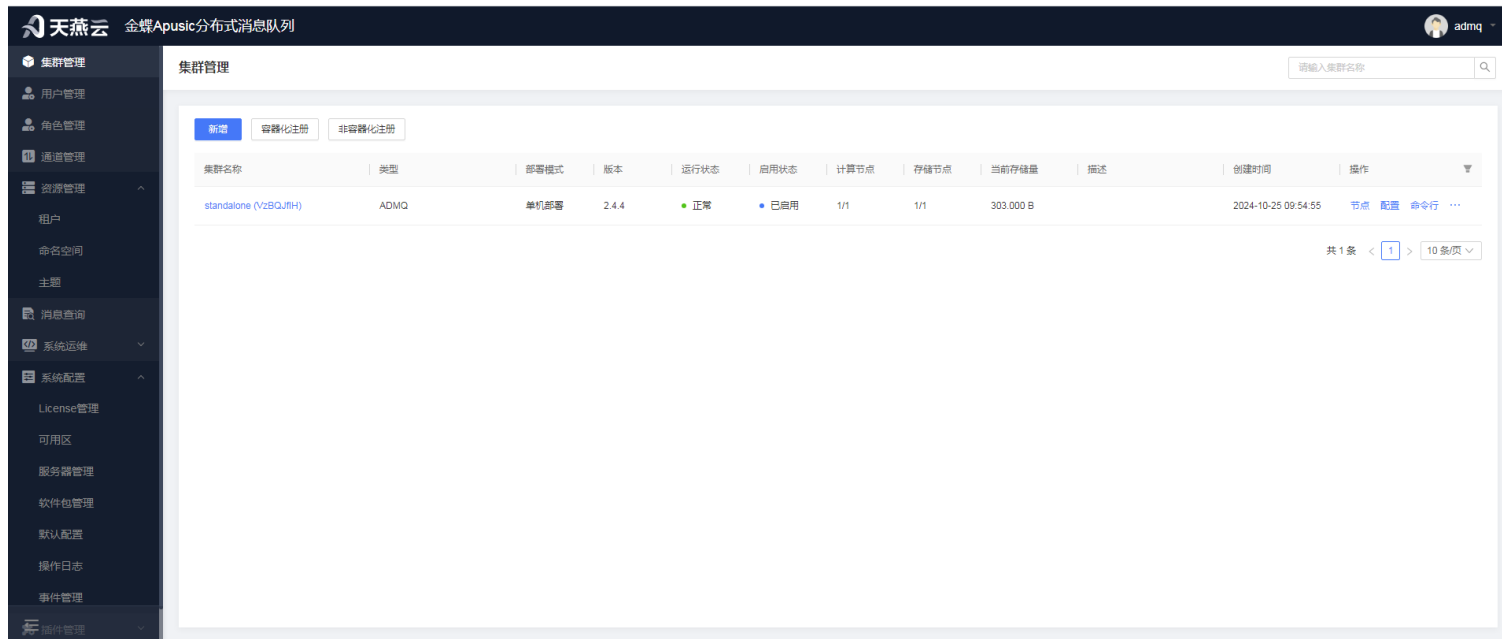


备注：超级管理员用户可查询到所有用户的操作日志信息，普通用户只能查询自己的操作日志信息。

5.3 集群

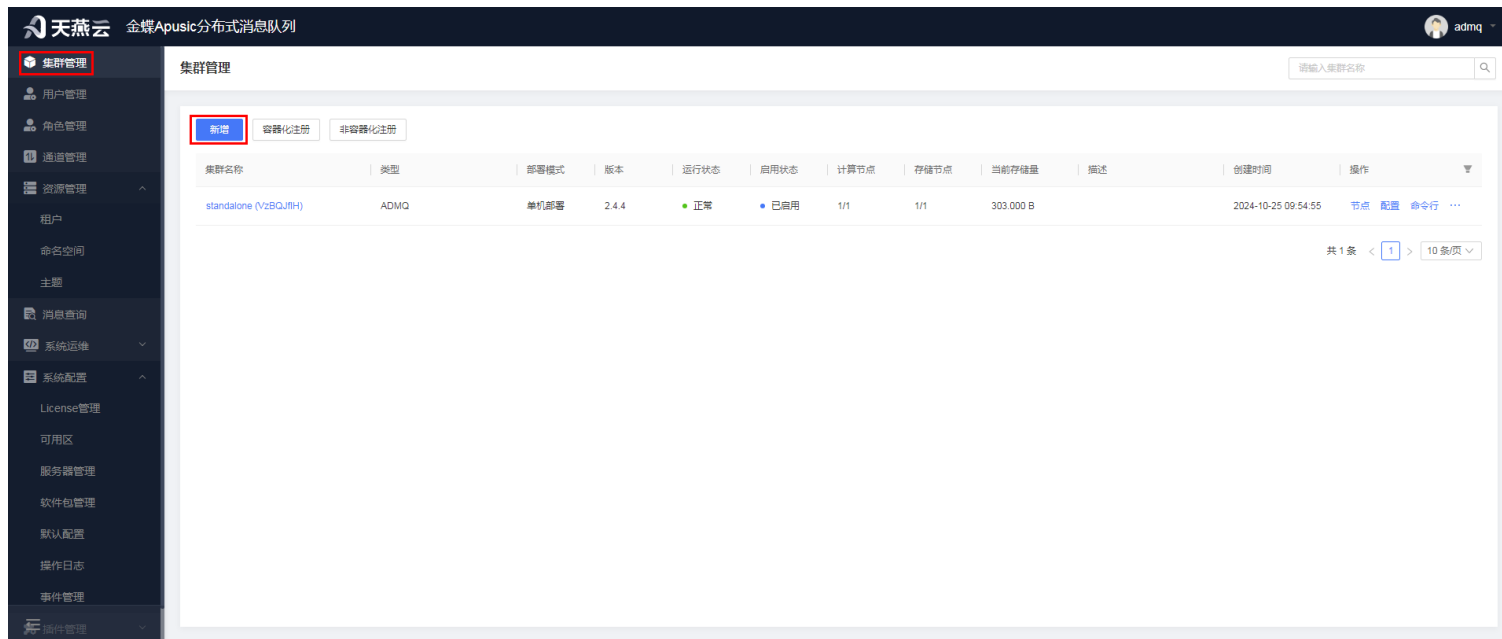
5.3.1 集群管理

集群管理用于部署管理集群，有新增集群信息、配置集群信息、删除集群信息、查询集群详细信息、查看令牌、证书管理、依赖管理、插件管理；部署节点、启动节点、重启节点、停止节点、配置节点参数、删除节点、编辑节点、节点资源监控；一键部署、一键启动、一键停止。



5.3.1.1 新增集群

进入【集群管理】，在集群管理页面左上角点击【新增】按钮，新增集群信息。

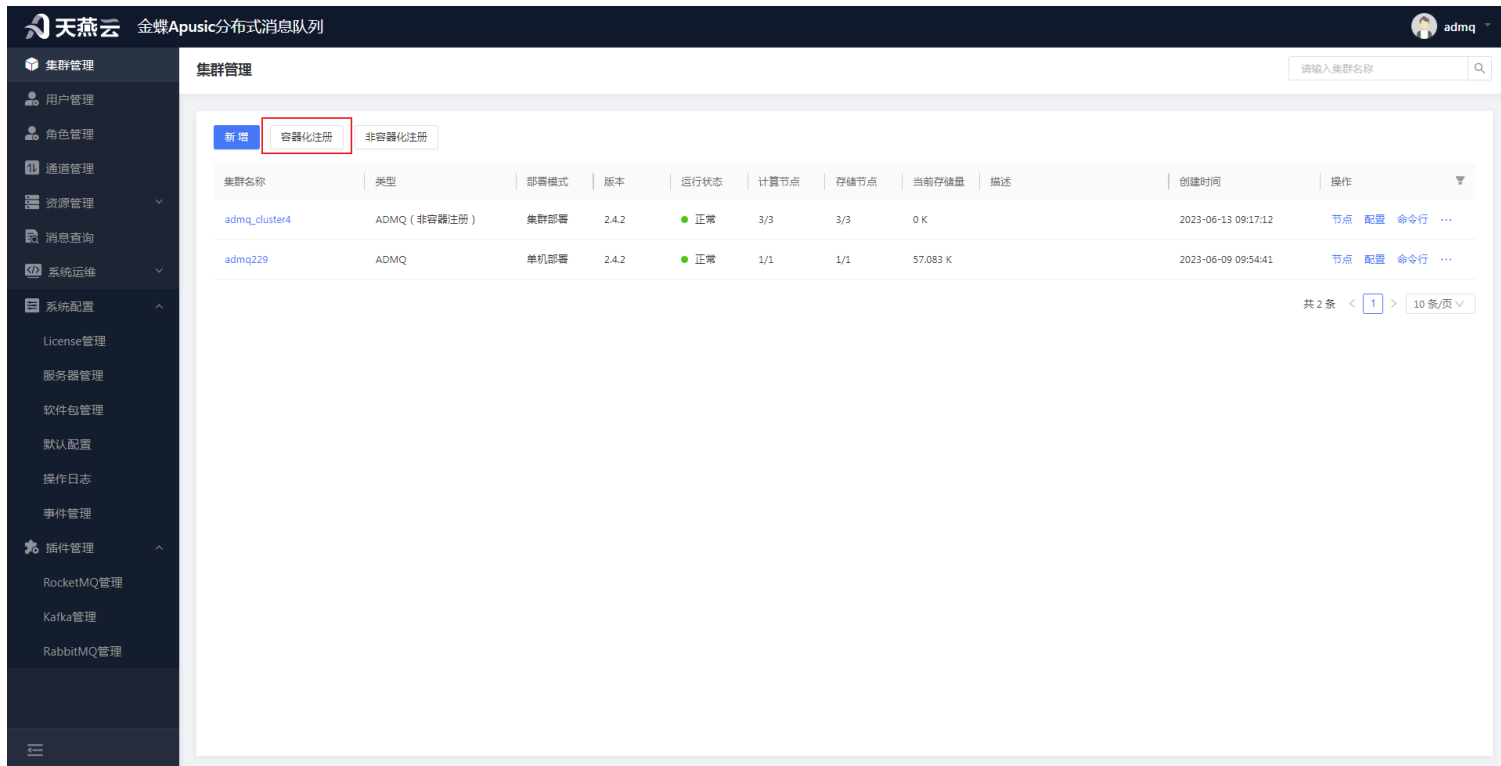


- 集群名称：自定义，唯一标识
- 集群类型：单项选项，ADMQ、RocketMQ，可部署ADMQ集群或者RocketMQ集群
- 部署模式：单项选项，集群部署、单机部署
- 可用区：部署服务器的所在数据中心
- 版本：部署ADMQ的版本号
- 开启插件：选择需要部署的开源协议插件
- 本地协调器服务器：集群的本地协调器地址，下拉选择已经新增的服务器（详见2.1服务器管理）
- 存储程序服务器：集群的存储程序地址，下拉选择已经新增的服务器（详见2.1服务器管理）；若选择部署模式为单机，则默认为本地协调器选择的地址，不可更改
- 计算程序服务器：集群的计算程序地址，下拉选择已经新增的服务器（详见2.1服务器管理）；若选择部署模式为单机，则默认为本地协调器选择的地址，不可更改
- 密钥文件：生成用户token的文件，与集群进行绑定。有默认密钥和动态生成密钥，默认密钥则是软件包默认带的密钥文件，动态生成密钥则是随机生成token.key，每个集群的token.key都不一样。默认动态生成密钥文件。
- 备注：新增集群信息前，需要先创建一些基础信息，如上传软件包、新增服务器信息等。

5.3.1.2 容器化注册 (docker)

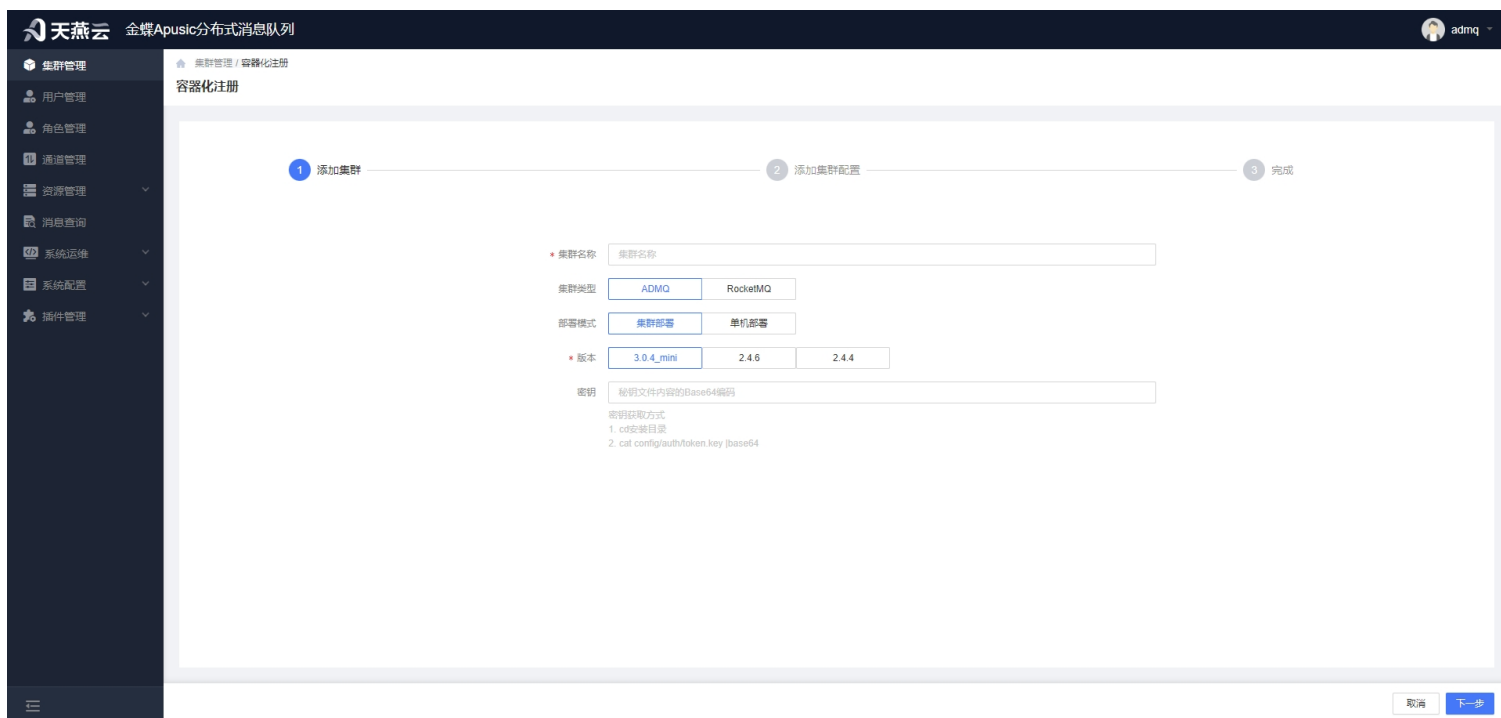
容器化注册是指将云上部署的ADMQ集群注册到现有的管控台上进行统一管控。

进入【集群管理】，在集群管理页面左上角点击【容器化注册】按钮，进入到容器化注册页面：（预备条件：事先已在云上部署了ADMQ集群，本次注册的非docker部署，但是流程类似）

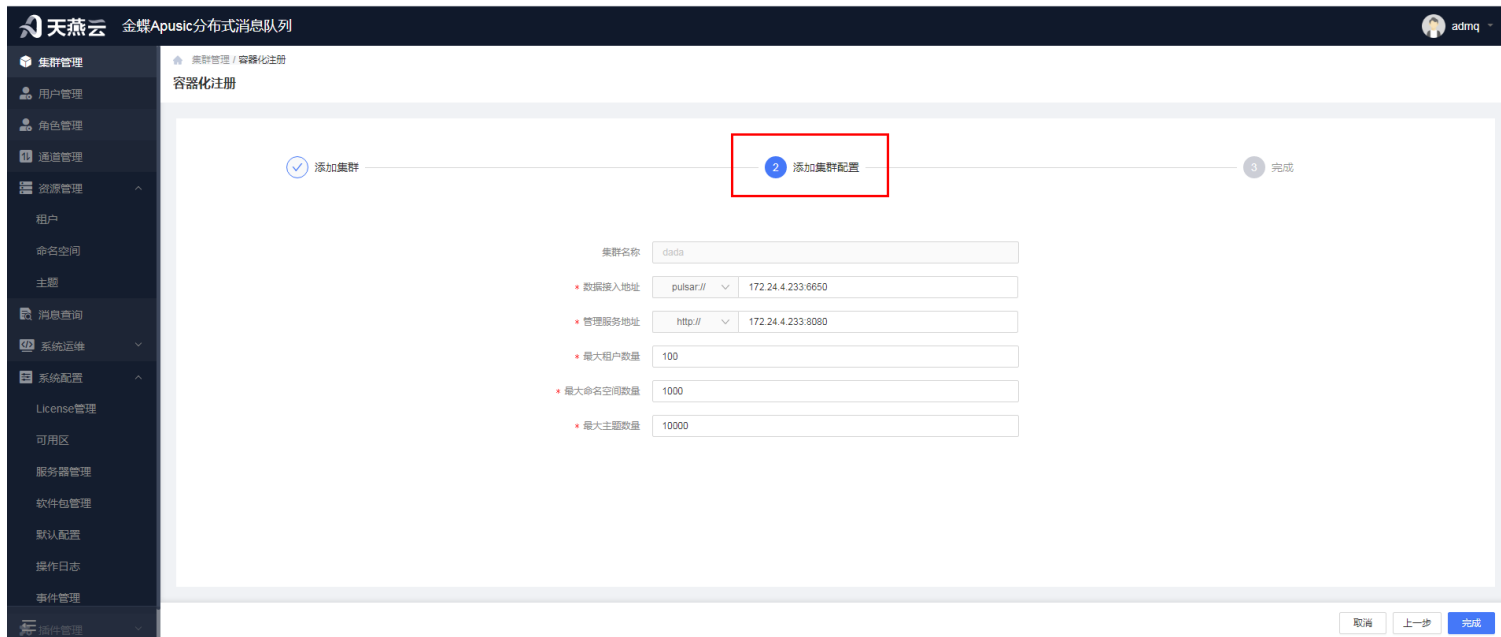


具体步骤如下：

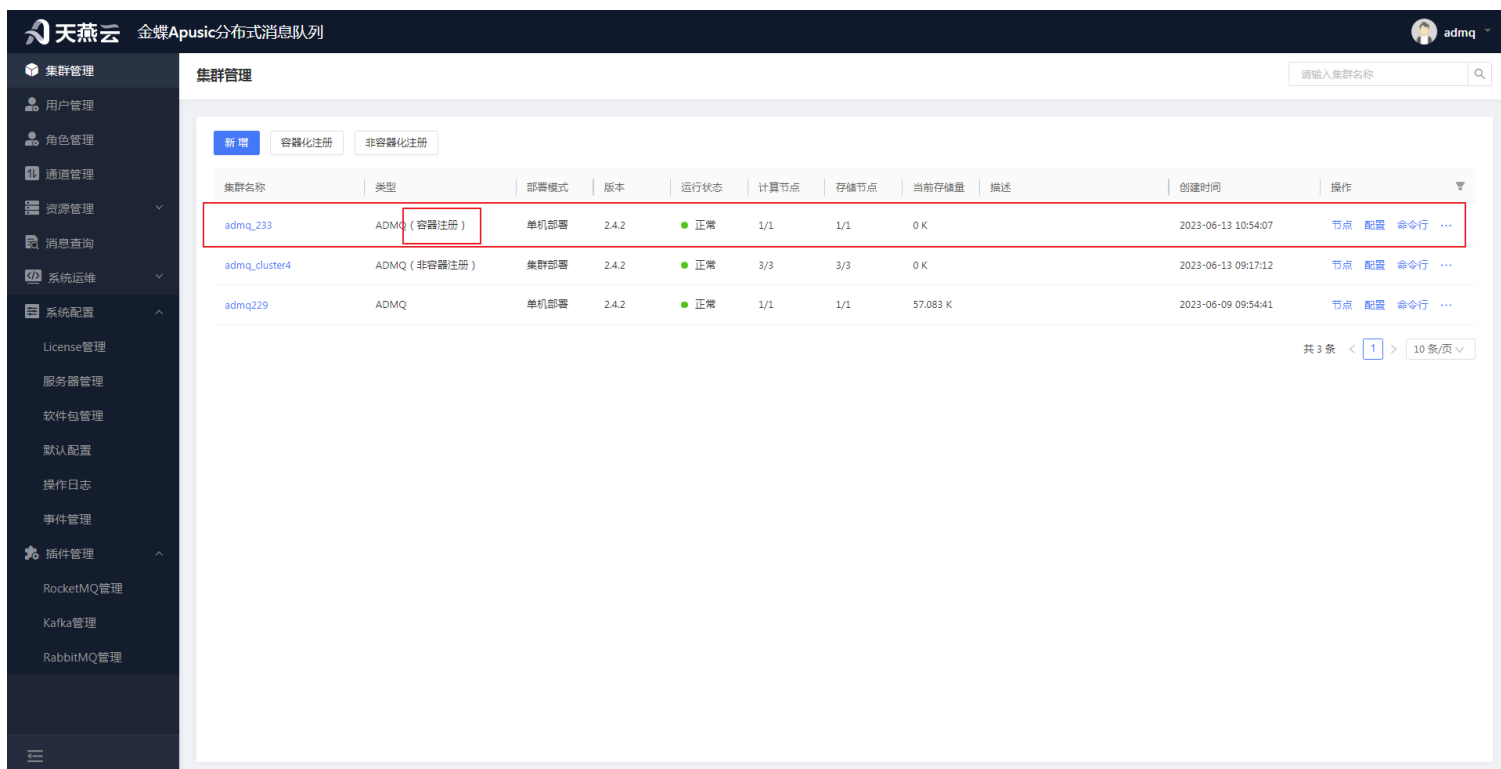
- (1) 添加集群：获取集群base64编码的密钥，具体获取方式参照非容器化注册中的获取方式，输入相关信息，点击下一步



- (2) 添加集群配置：输入相关信息，点击【完成】按钮



(3) 完成: 点击【完成】按钮会自动跳转到集群列表页面:



5.3.1.3 容器化注册 (k8s)

1、例如接入集群 `cc` 集群, `cc`为当前集群的实例名称

```
$ kubectl get po -n wx -o wide |grep cc
```

```
cc-admq-broker-0          1/1      Running    0          3h33m    172.20.7.243
172.24.4.238             <none>   <none>
cc-admq-broker-1          1/1      Running    0          3h32m    172.20.1.41
```

```

172.24.4.147 <none> <none>
cc-admq-broker-2 1/1 Running 0 3h32m 172.20.6.150
172.24.4.237 <none> <none>
cc-admq-storage-0 1/1 Running 1 3h33m 172.20.8.200
172.24.4.239 <none> <none>
cc-admq-storage-1 1/1 Running 0 3h32m 172.20.7.244
172.24.4.238 <none> <none>
cc-admq-storage-2 1/1 Running 0 3h32m 172.20.1.42
172.24.4.147 <none> <none>
cc-admq-zk-0 2/2 Running 0 3h33m 172.20.6.149
172.24.4.237 <none> <none>
cc-admq-zk-1 2/2 Running 0 3h33m 172.20.1.40
172.24.4.147 <none> <none>
cc-admq-zk-2 2/2 Running 0 3h33m 172.20.8.201
172.24.4.239 <none> <none>

```

2、打开集群接入页面

选择容器化注册



容器接入

1 添加集群
2 添加集群配置
3 完成

* 集群名称

集群类型 ADMQ RocketMQ

部署模式 集群部署 单机部署

* 版本 3.0.4_mini 2.4.6 2.4.4

密钥

密钥获取方式
1. cd安装目录
2. cat config/auth/token.key |base64

取消
下一步

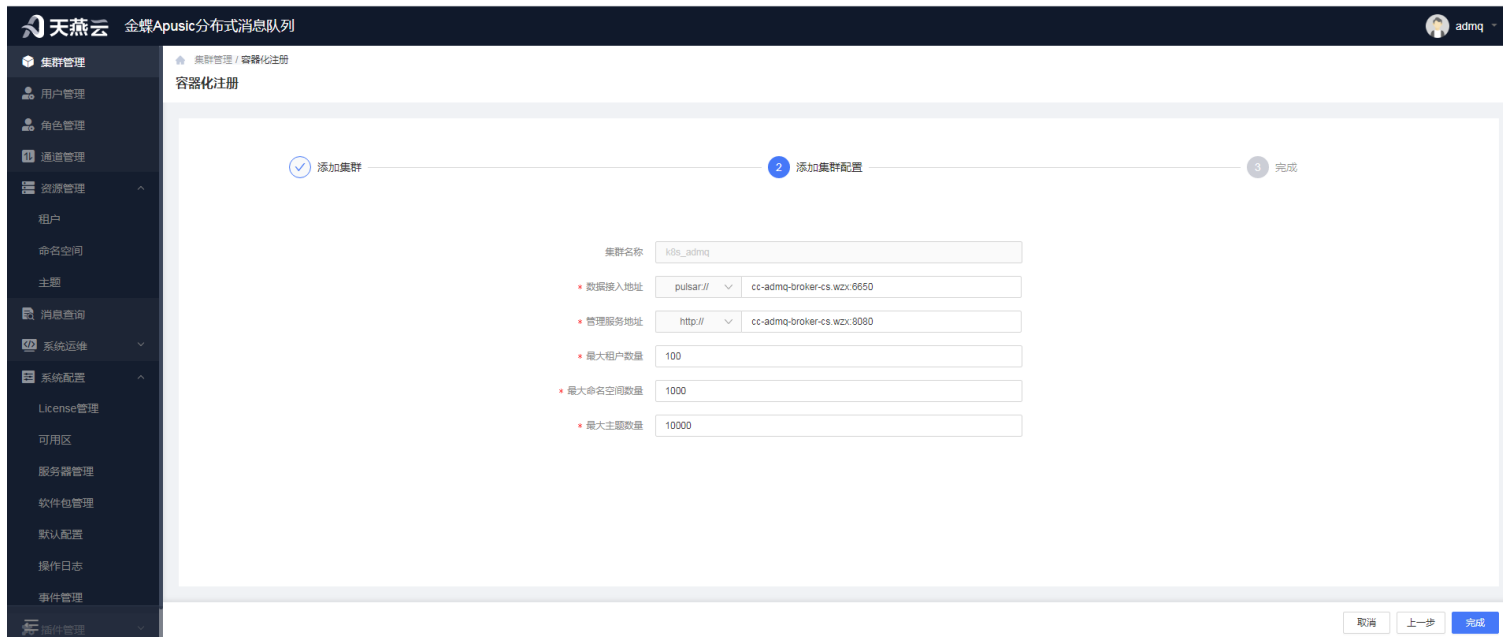
添加集群配置

必填项

```
$ kubectl get svc
NAME                                TYPE                CLUSTER-IP          EXTERNAL-IP          PORT(S)
AGE
cc-admq-broker-cs                   ClusterIP           10.68.142.147      <none>
8080/TCP,15673/TCP,6650/TCP        3h42m
cc-admq-broker-hs                   ClusterIP           None                <none>
6650/TCP,5672/TCP,6672/TCP        3h42m
cc-admq-storage-cs                  ClusterIP           10.68.168.35       <none>
8000/TCP
3h42m
cc-admq-storage-hs                  ClusterIP           None                <none>
3181/TCP
3h42m
cc-admq-zk-cs                       ClusterIP           10.68.61.214       <none>
2181/TCP,9141/TCP
3h42m
cc-admq-zk-hs                       ClusterIP           None                <none>
2888/TCP,2181/TCP,3888/TCP        3h42m
```

数据接入地址: pulsar://cc-admq-broker-cs.wzx:6650

管理服务地址: http://cc-admq-broker-cs.wzx:8080

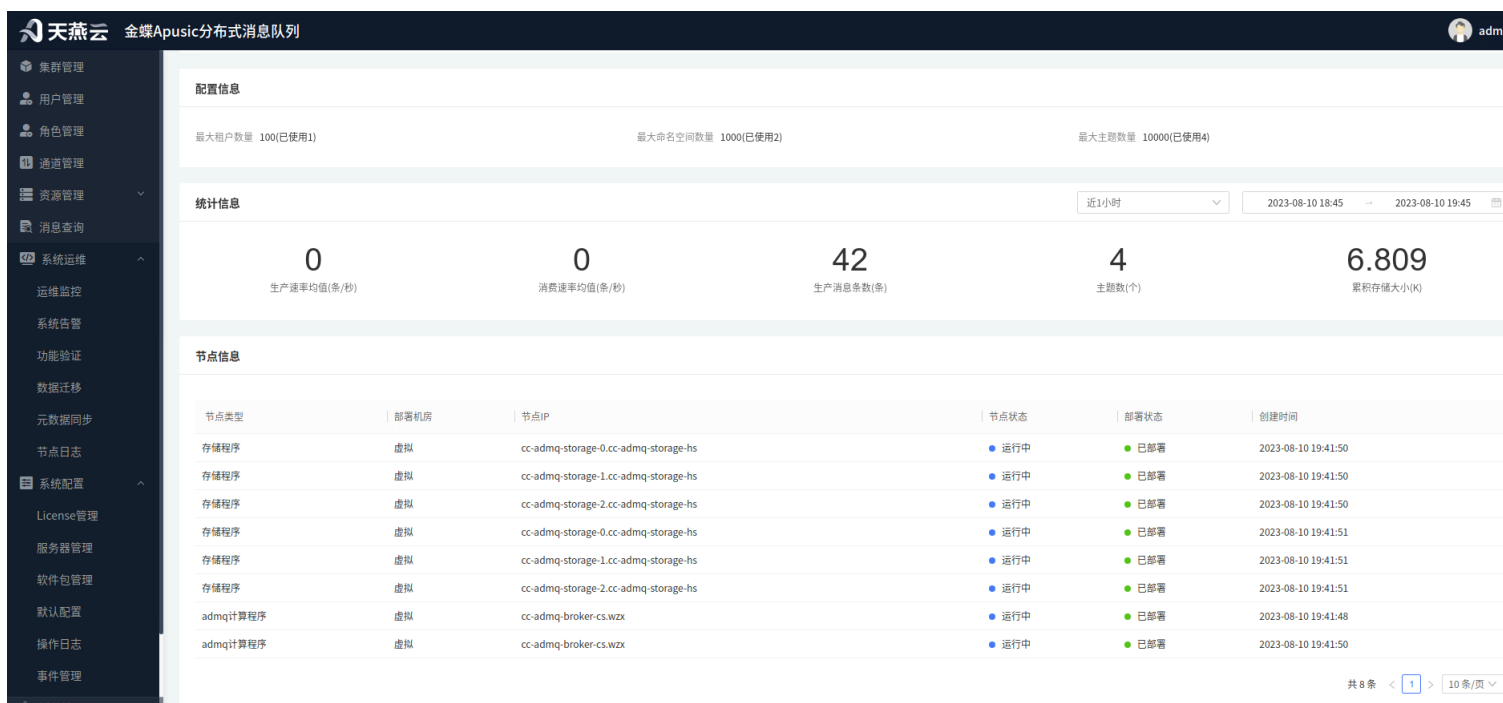


3、查看集群状态及配置

查看集群状态



集群详情



4、RabbitMQ配置

新增租户

RabbitMQ管理

租户 虚拟机 交换机 队列 路由

新增

集群 cc 请输入租户名称

租户 ID	租户名称	所属集群	虚拟主机数	交换机数	队列数	描述	创建时间	操作
amqp-data	amqp-data	cc	2	1	1		2023-08-10 19:43:03	编辑 删除

共 1 条 < 1 > 10 条/页

新增虚拟机

RabbitMQ管理

租户 虚拟机 交换机 队列 路由

新增

集群 cc 租户 data(amqp-data) 请输入虚拟机名称

名称	消息保留时间	交换机数	队列数	描述	创建时间	操作
vhost01	null undefined	0	0	系统自动创建	2023-08-10 19:45:00	分配权限 编辑 删除
vhost02	30 天	1	1		2023-08-10 19:43:19	分配权限 编辑 删除

共 2 条 < 1 > 10 条/页

新增交换机

RabbitMQ管理

租户 虚拟机 交换机 队列 路由

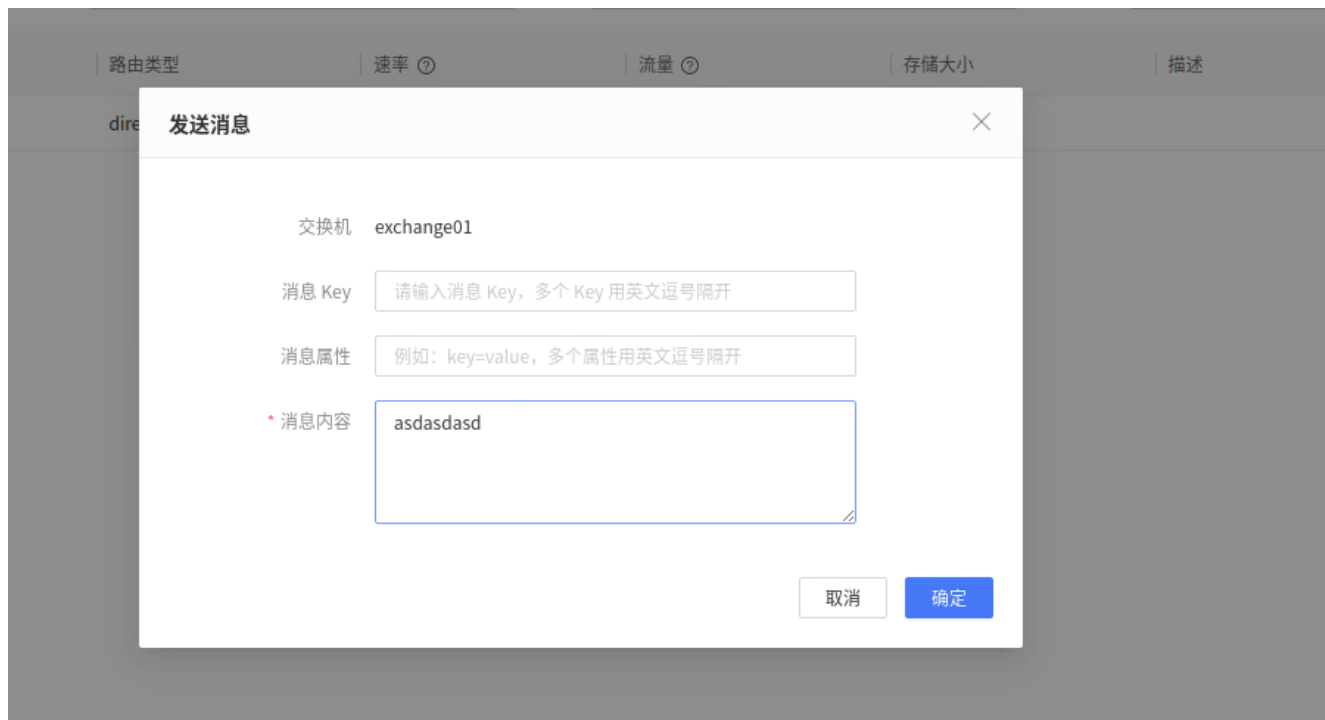
新增 删除

集群 cc 租户 data(amqp-data) 虚拟机 vhost02 请输入交换机名称

<input type="checkbox"/> 名称	虚拟机	路由类型	速率	流量	存储大小	描述	创建时间	操作
<input type="checkbox"/> exchange01	vhost02	direct	0.0/0.0	0 K/0 K	236.000 B		2023-08-10 19:43:37	发送消息 编辑 删除

共 1 条 < 1 > 10 条/页

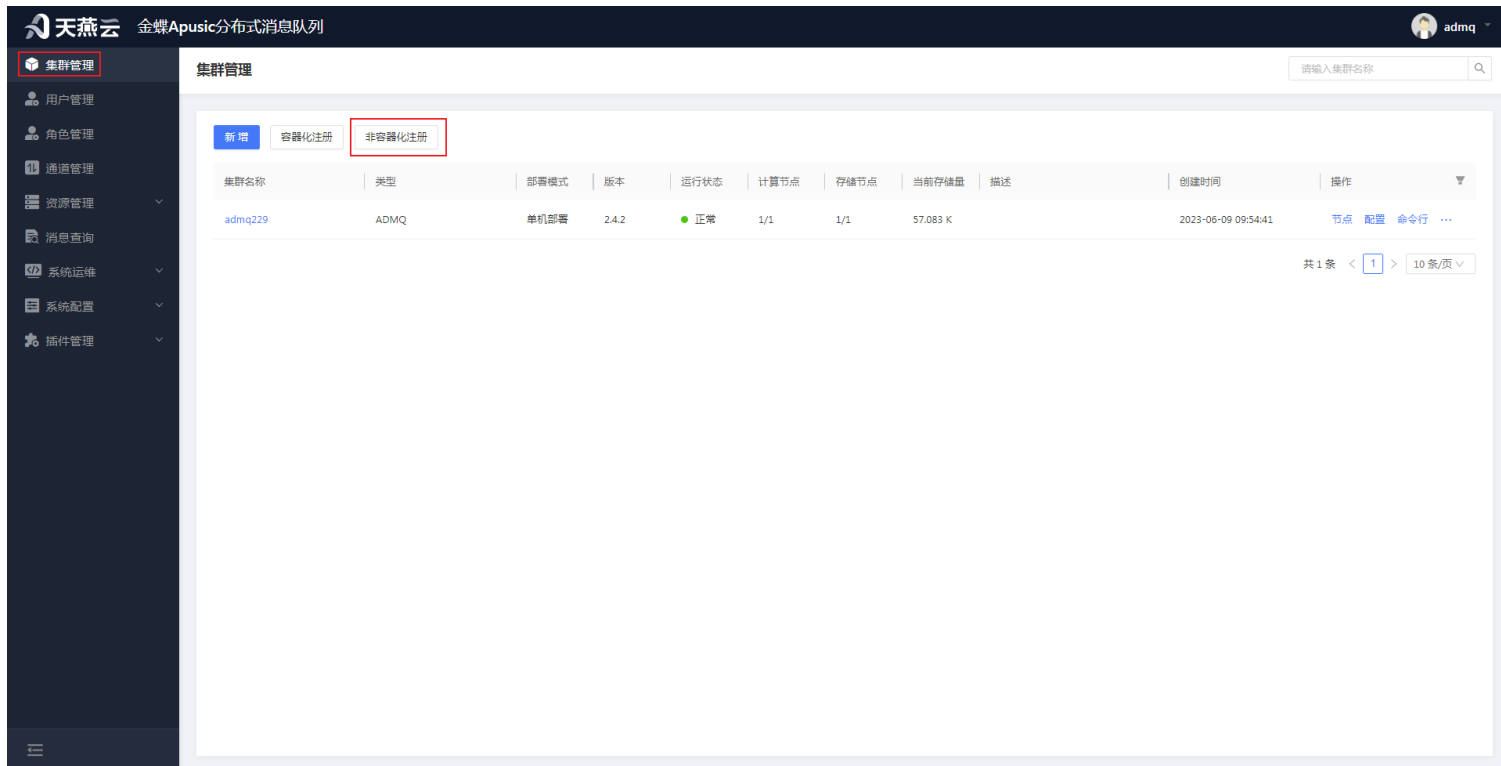
发送消息



5.3.1.4 非容器化注册

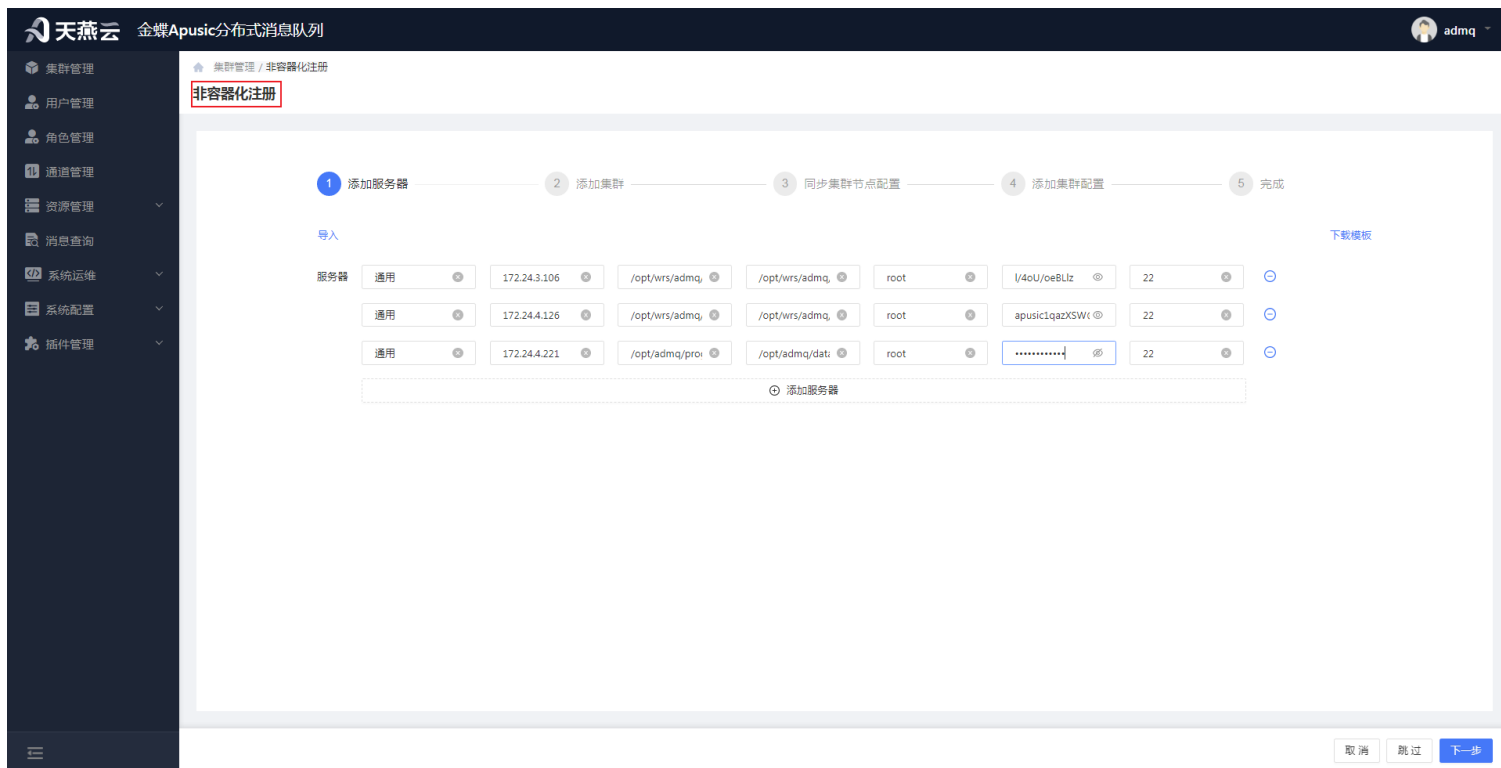
非容器化注册是指将后台部署的ADMQ集群或者将其它管控台部署的ADMQ集群注册到现有的管控台上进行统一管控。（也就是将非现有管控台部署的ADMQ集群注册到现有管控台进行统一管控运维。）

进入【集群管理】，在集群管理页面左上角点击【非容器化注册】按钮，进入到非容器化注册页面：（预备条件：事先已在其它地方部署了ADMQ集群）

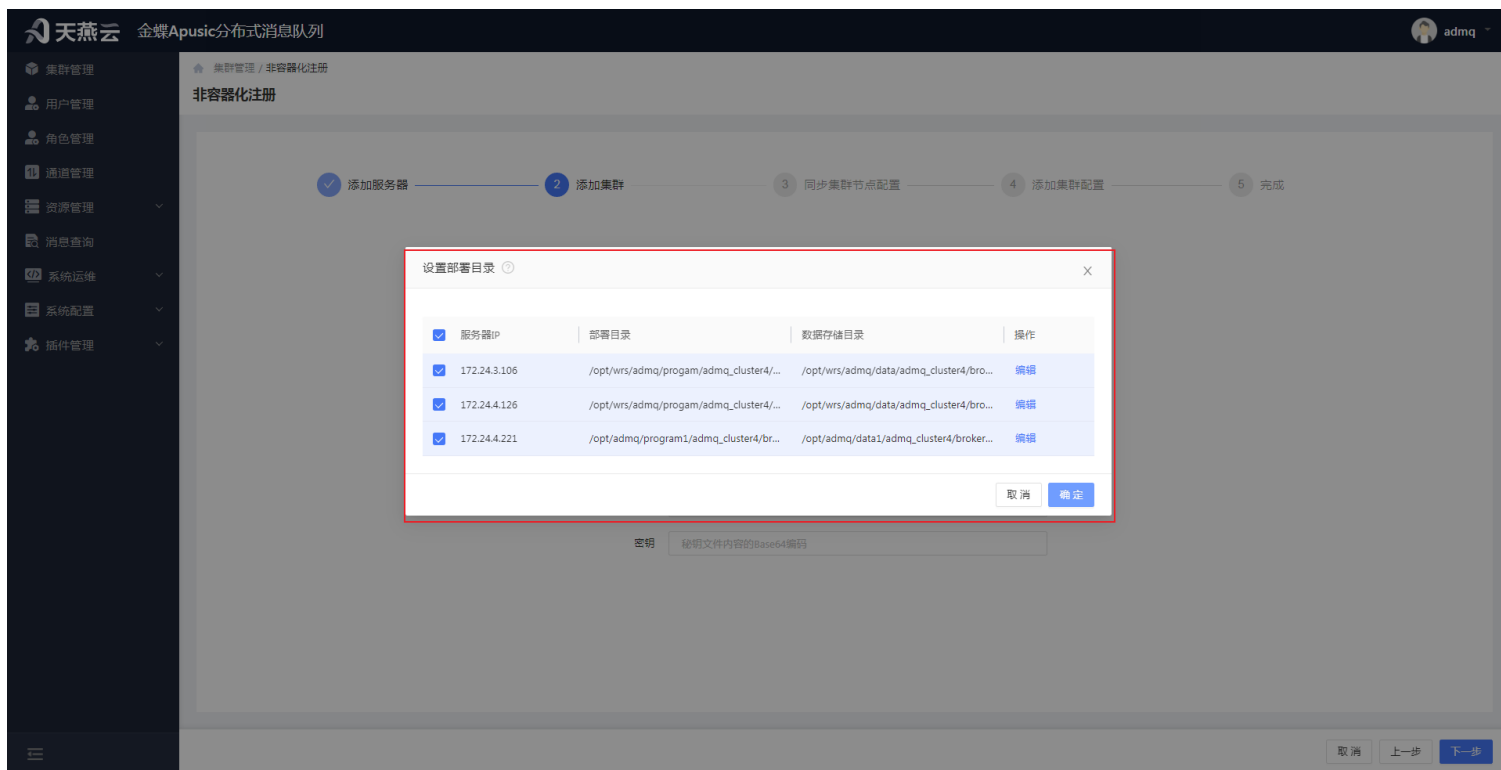
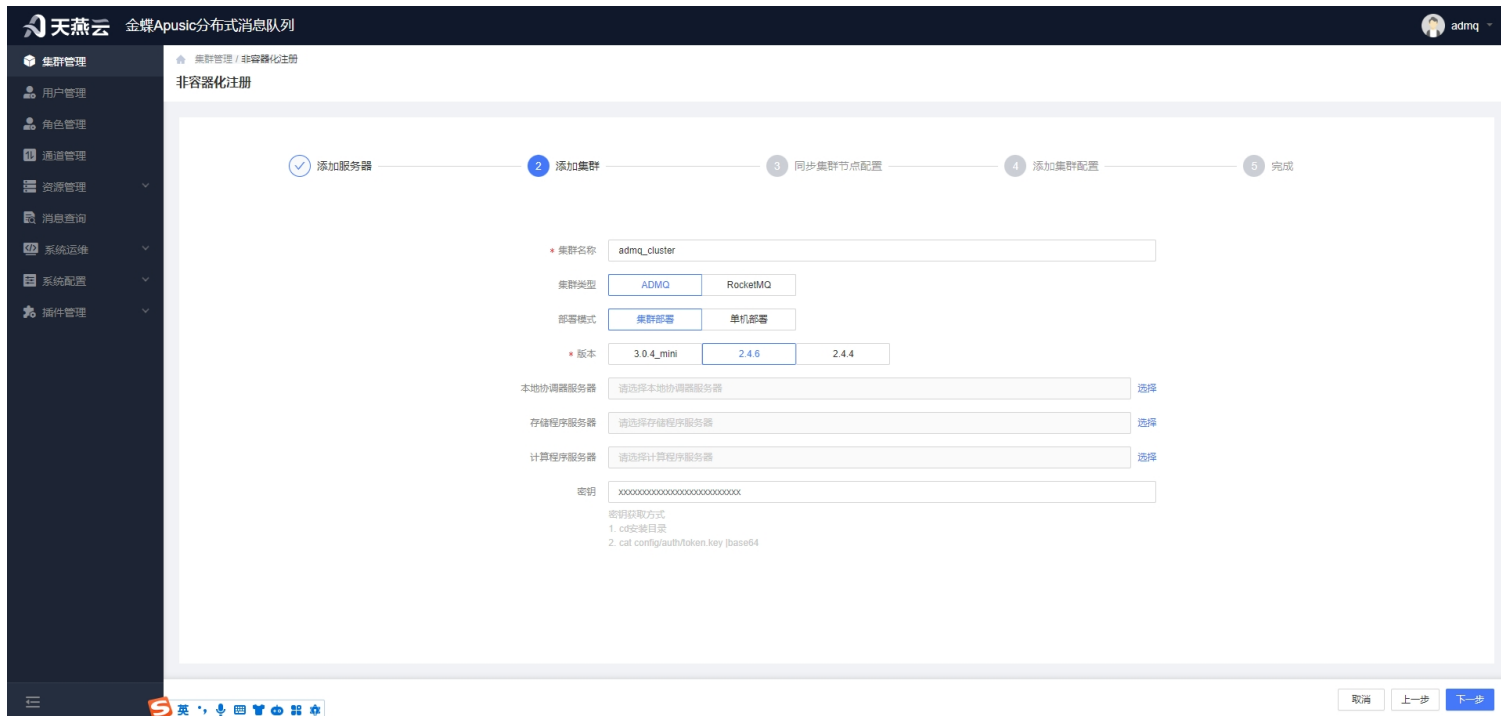


具体步骤如下：

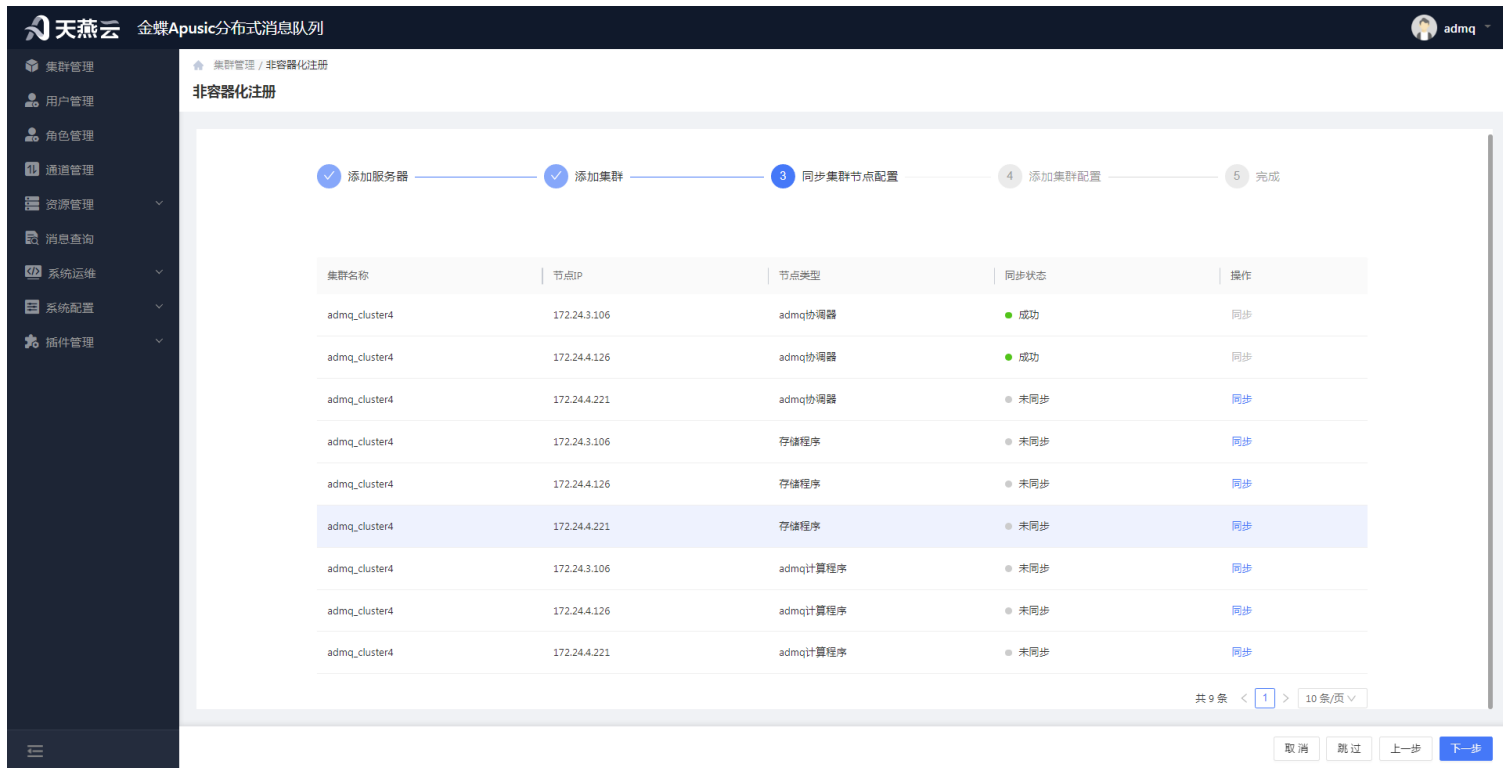
(1) 添加服务器：输入相关信息，点击下一步



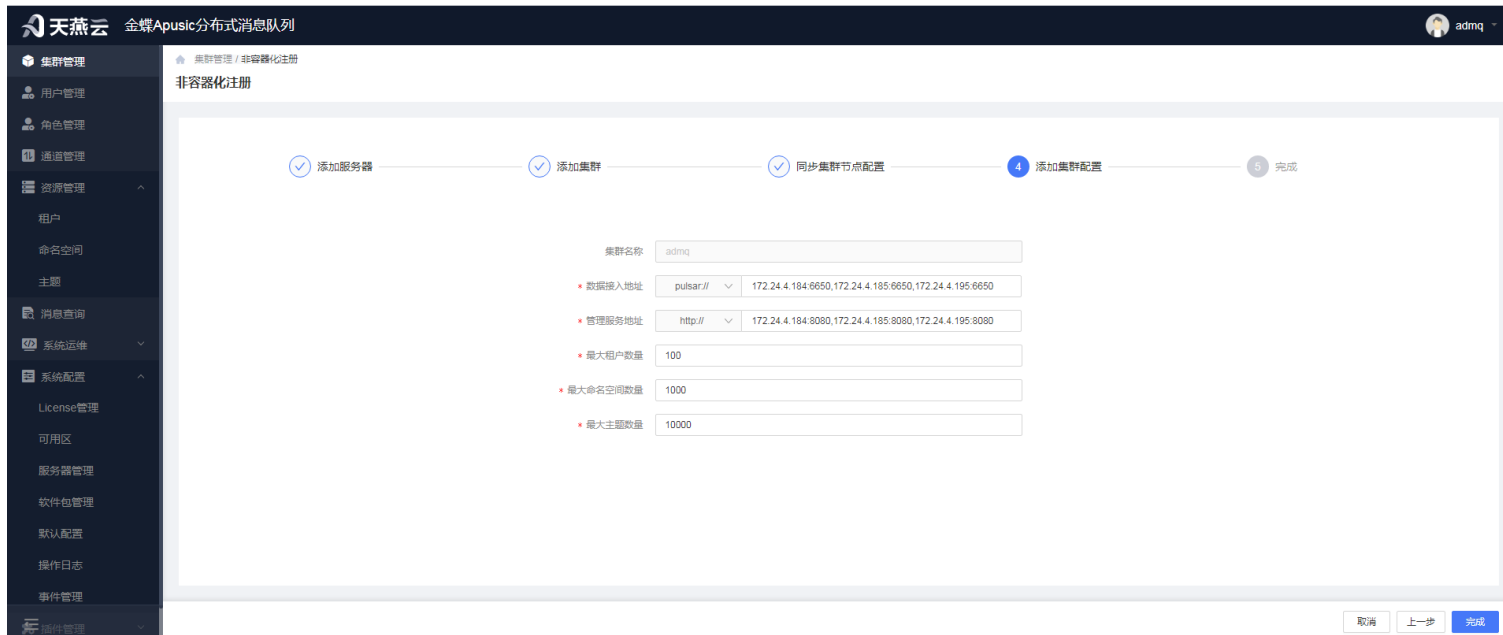
(2) 添加集群：在集群的任意一个broker的部署目录（如/opt/wrs/admq/progam/admq_cluster4/broker/wfVULpN7/admq-V2.4.2/）下，使用命令 `cat config/auth/token.key |base64`，获取集群base64编码的密钥，然后输入相关信息，点击下一步



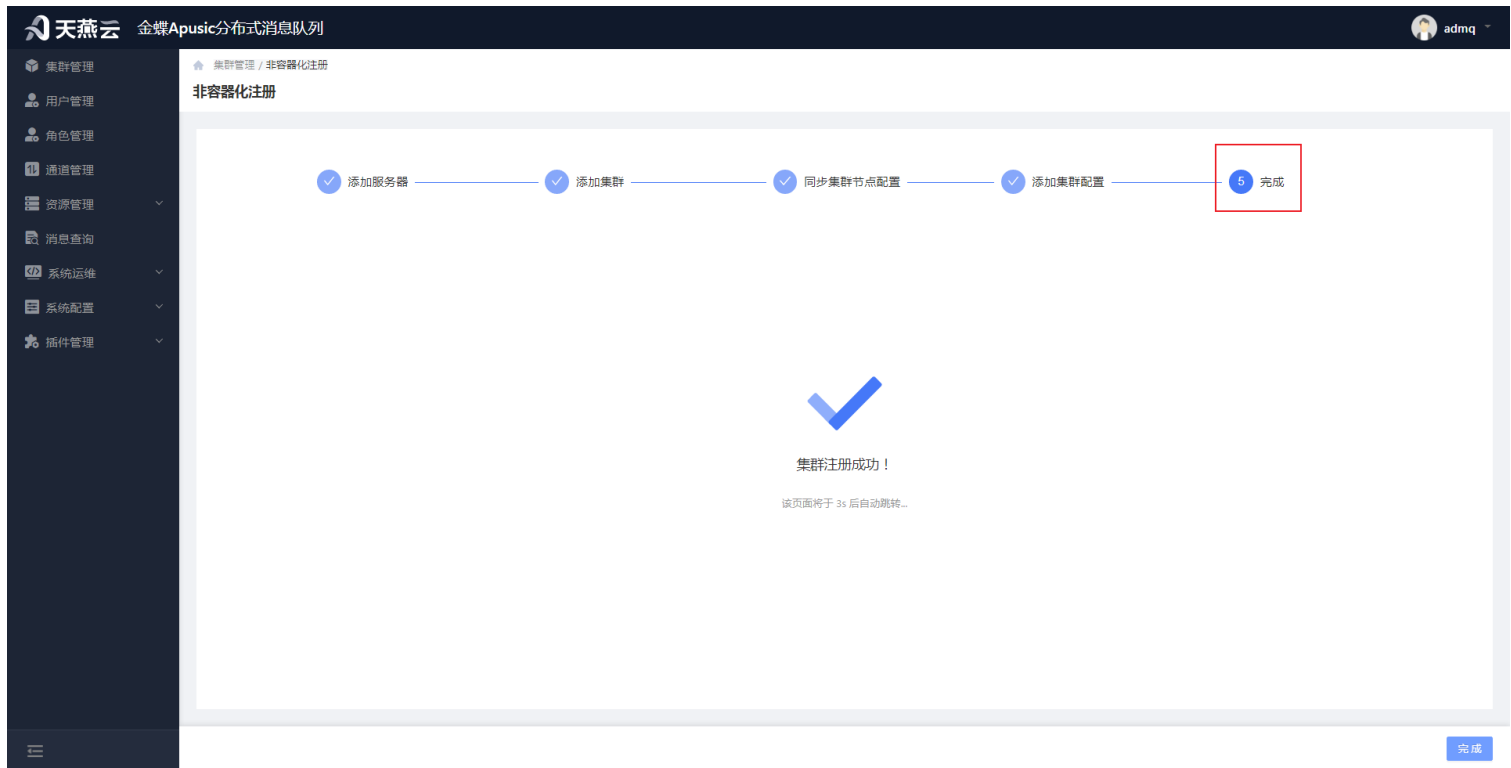
(3) 同步集群节点配置：点击每个节点的【同步】按钮进行同步，当同步状态显示成功，表示同步成功。（备注：若同步不成功，在操作一栏会出现错误原因，可能是文件不存在，这时需要点击【上一步】，重新编辑部署目录后再次同步即可。）



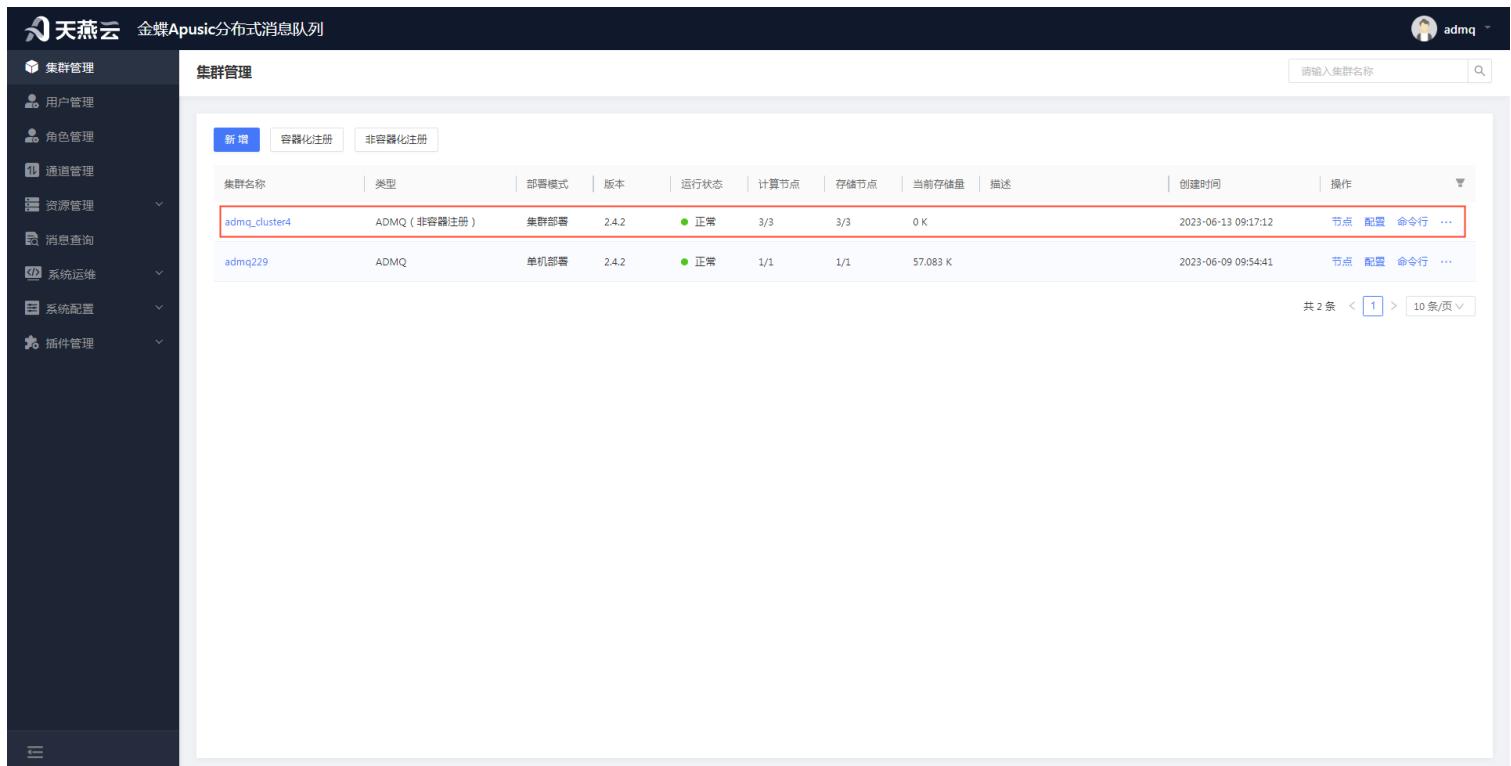
(4) 添加集群配置:



(5) 完成:



集群注册完成后点击【完成】按钮会跳转到集群列表页面，或者等待几秒后也会自动跳转到集群列表页面，然后可对该集群节点进行启动、停止、配置更新等多种操作：



5.3.1.5 配置集群信息

进入【集群管理】，在集群管理页面选择需要配置的集群信息，点击【配置】按钮，配置集群信息。

天燕云 金蝶Apusic分布式消息队列 admq

集群管理

新增

集群名称	类型	部署模式	版本	运行状态	计算节点	存储节点	当前存储量	描述	创建时间	操作
admq_single126	ADMQ	单机部署	2.2.1	● 正常	1/1	1/1	0 K		2022-04-18 00:11:04	令牌 节点 配置 更多
admq_cluster456	ADMQ	集群部署	2.2.1	● 正常	0/3	0/3	0 K		2022-04-19 20:43:54	令牌 节点 配置 更多
rocketmq_cluster4...	ROCKETMQ	集群部署	4.9.2	● 正常	0/3	--			2022-04-19 20:48:41	令牌 节点 配置 更多
rocketmq_M5	ROCKETMQ	主从部署	4.9.2	● 正常	0/4	--			2022-04-19 20:48:03	令牌 节点 配置 更多
dddasd	ROCKETMQ	单机部署	4.9.2	● 正常	1/1	--			2022-04-18 00:01:42	令牌 节点 配置 更多

共 5 条 < 1 > 10 条/页

5.3.1.5.1 ADMQ集群配置

ADMQ集群的配置页面，包括：基础配置、资源配置、安全配置、其它配置

天燕云 金蝶Apusic分布式消息队列 admq

集群管理 / 集群配置

集群配置

基础配置 编辑

集群名称	admq_single126	数据接入地址	pulsar://172.24.4.126:6650
管理服务地址	http://172.24.4.126:8080	KAFKA服务地址	
MQTT服务地址		RABBITMQ服务地址	
ROCKETMQ服务地址		描述	

资源配置 编辑

* 最大租户数量	100(已使用0)	自动创建主题	允许
* 最大命名空间数量	1000(已使用0)	自动删除无效主题	不允许
* 最大主题数量	10000(已使用0)		

安全配置 编辑

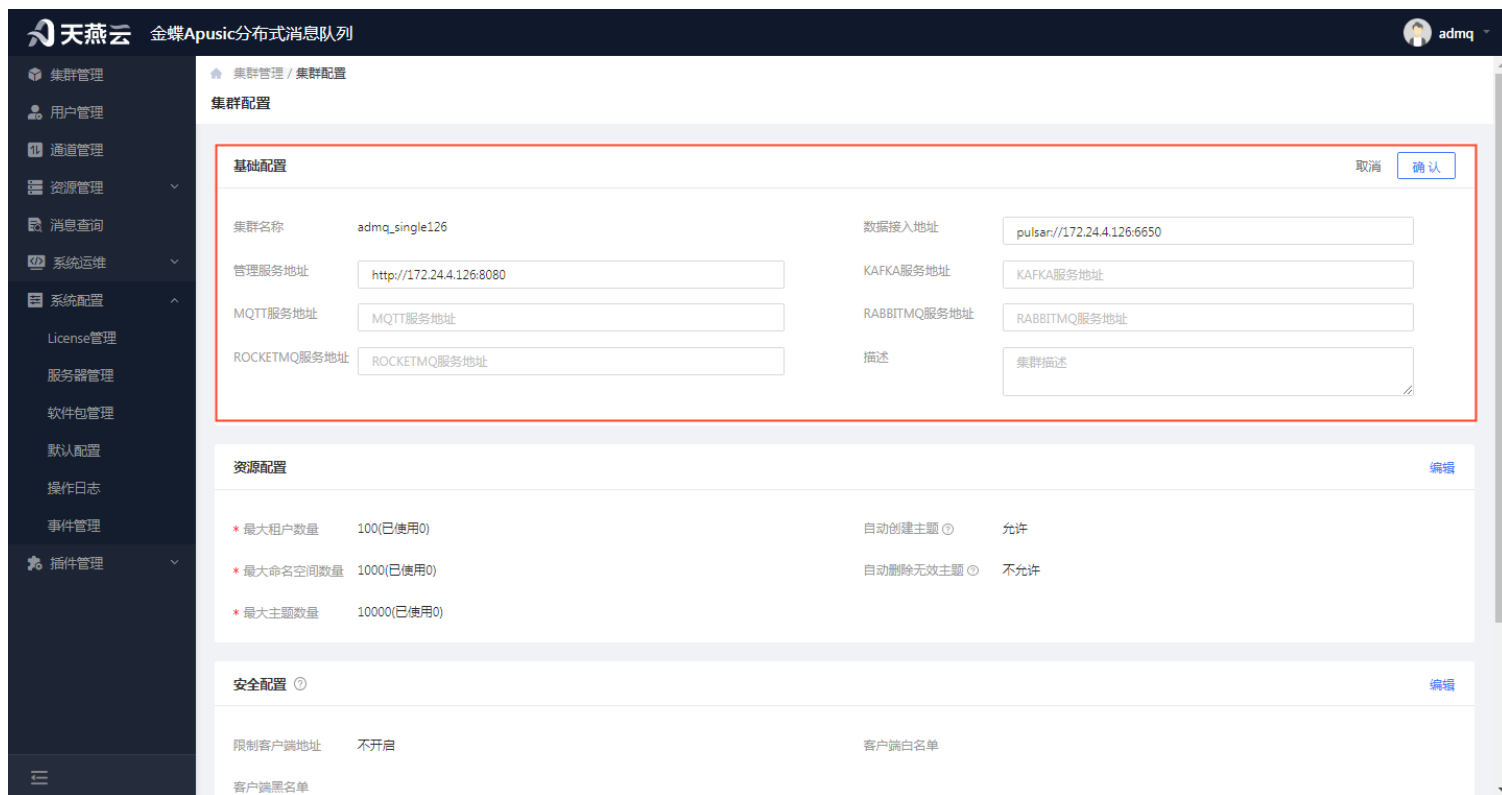
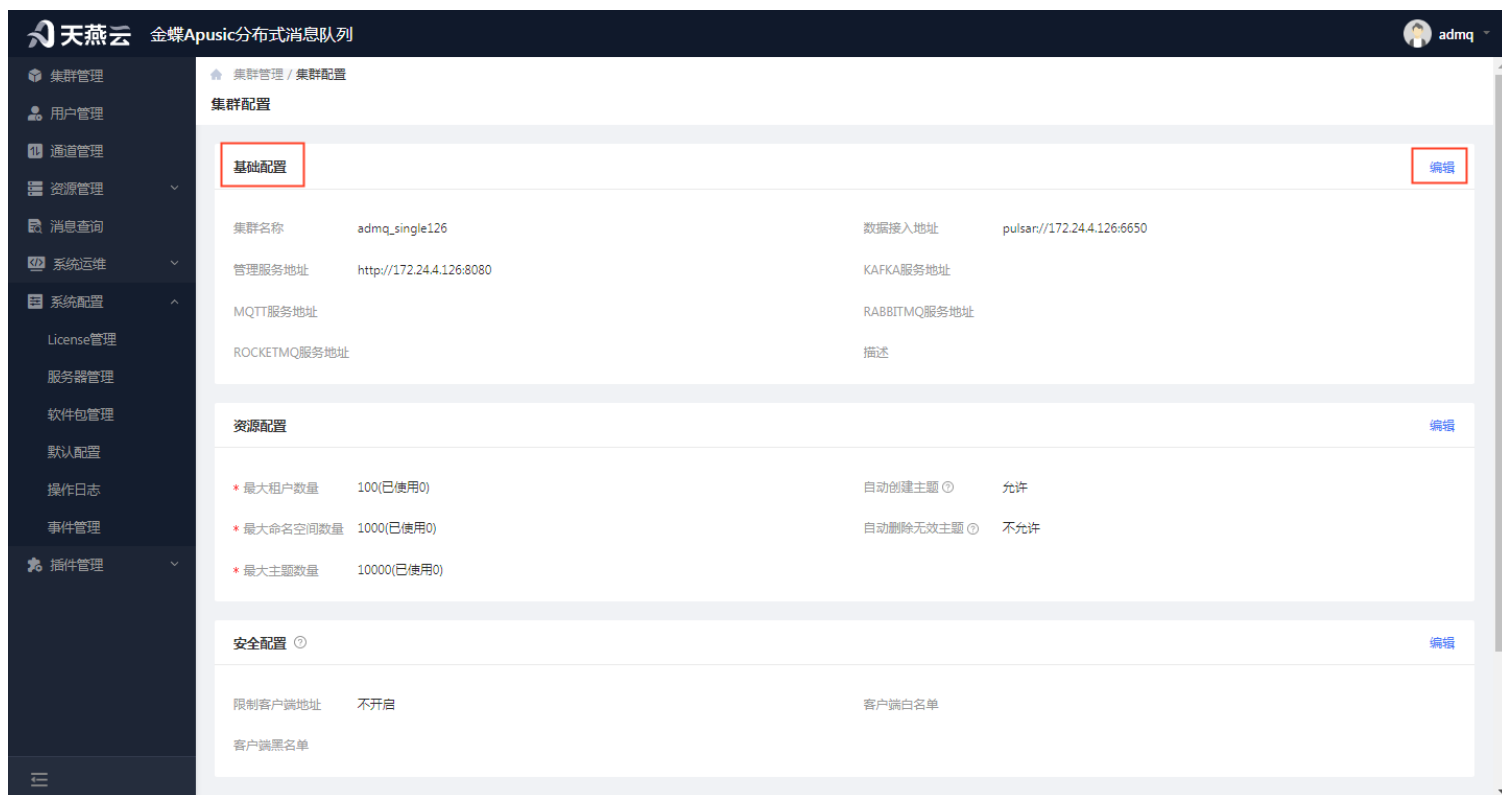
限制客户端地址	不开启	客户端白名单	
客户端黑名单			

其他配置 编辑

事务功能	不开启	* Journal日志存储目录	
* 数据存储目录			

5.3.1.5.1.1 基础配置

在集群配置页面的基础配置部分，点击右边的编辑按钮，可配置基础信息



- 集群名称：ADMQ集群的唯一标识，不可变
- 数据接入地址：集群计算程序服务地址，系统将会根据选择的计算程序地址默认填写，如 pulsar://172.24.4.104:6650,172.24.4.105:6650,172.24.4.106:6650，也可手动更改

- 管理服务地址：admq集群的web地址，系统将会根据选择的计算程序地址默认填写，如<http://172.24.1.116:8080,172.24.1.118:8080,172.24.4.130:8080>，也可手动更改（但若填写错误，则无法跟broker进行交互）
- KAFKA服务地址：kafka插件的服务地址，不显示，若加载插件后该部分会自动去除
- MQTT服务地址：mqtt插件的服务地址，不显示，若加载插件后该部分会自动去除
- RABBITMQ服务地址：rabbitmq插件的服务地址，不显示，若加载插件后该部分会自动去除
- ROCKETMQ服务地址：rocketmq插件的服务地址，不显示，若加载插件后该部分会自动去除
- 描述：集群的详细信息介绍

5.3.1.5.1.2 资源配置

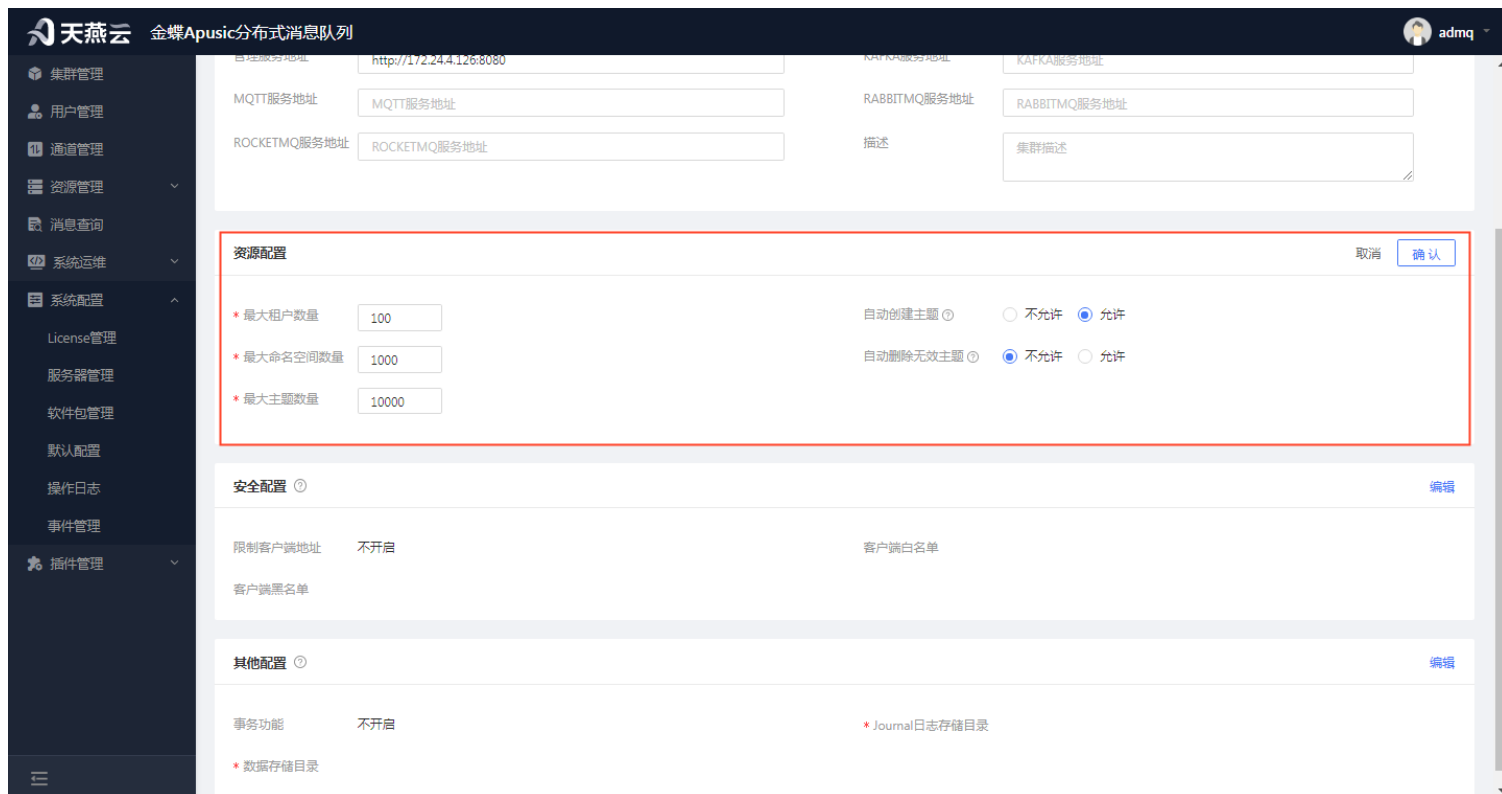
在集群配置页面的资源配置部分，点击右边的编辑按钮，可对集群的资源进行配置

The screenshot displays the '资源配置' (Resource Configuration) page in the management console. The page is divided into several sections, each with a table of configuration items and an '编辑' (Edit) button.

资源配置			
* 最大租户数量	100(已使用0)	自动创建主题	允许
* 最大命名空间数量	1000(已使用0)	自动删除无效主题	不允许
* 最大主题数量	10000(已使用0)		

安全配置			
限制客户端地址	不开启	客户端白名单	
客户端黑名单			

其他配置			
事务功能	不开启	* Journal日志存储目录	
* 数据存储目录			

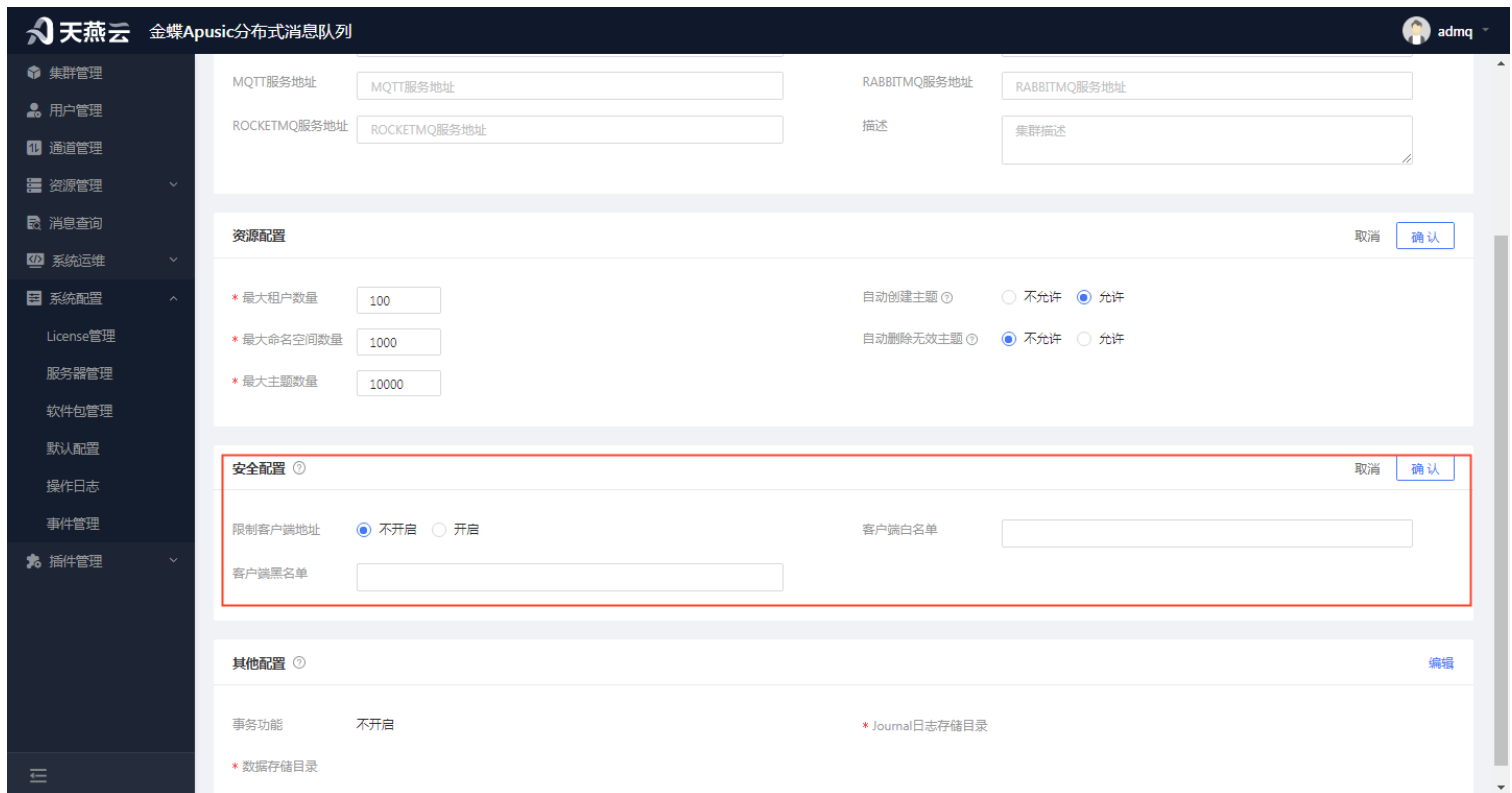
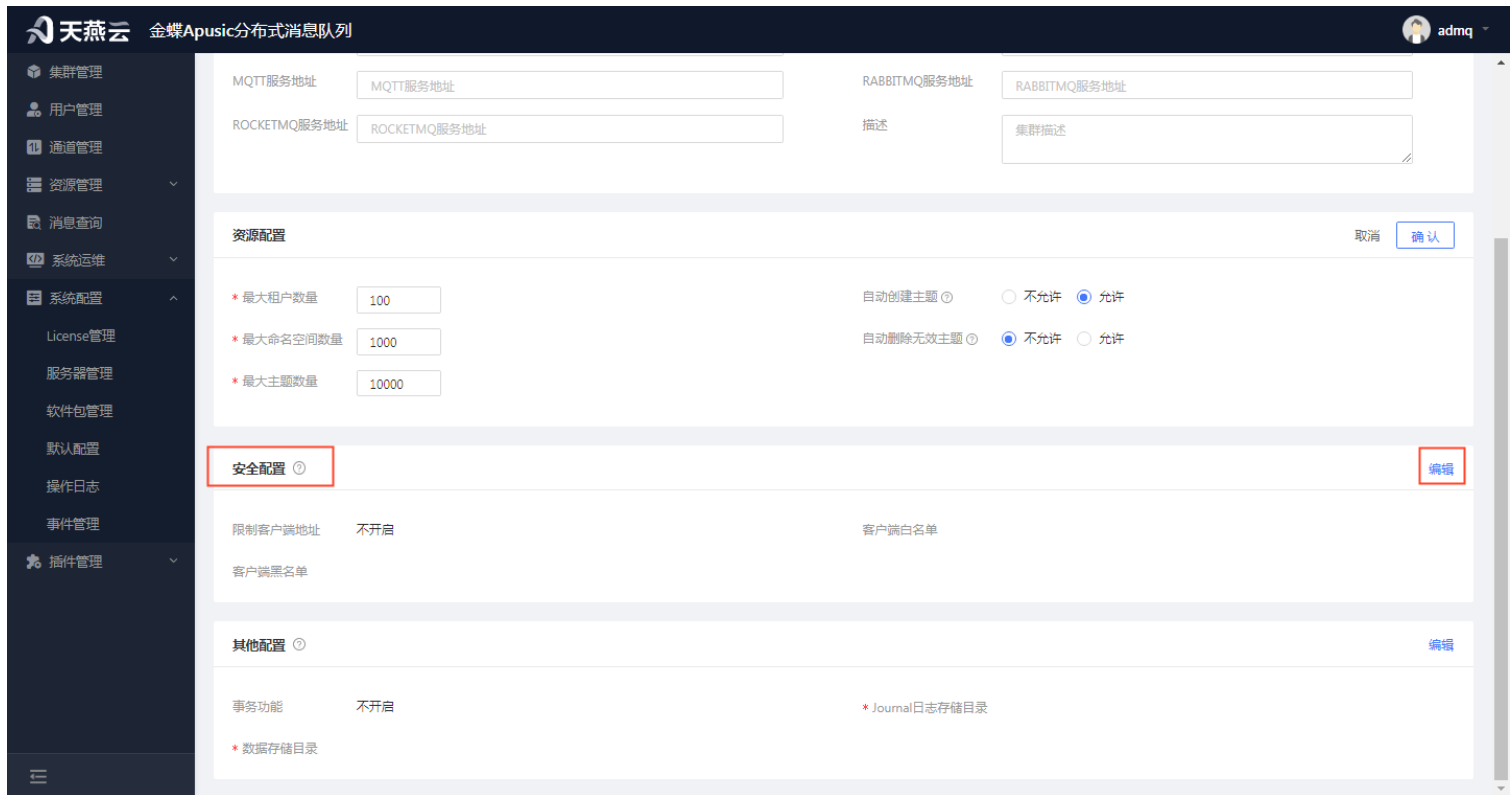


- 最大租户数量：集群下可创建的最大租户数量
- 最大命名空间数量：集群下可创建的最大命名空间数量
- 最大主题数量：集群下可创建的最大主题数量
- 自动创建主题：单选框，默认允许
- 自动删除无效主题：单选框，默认不允许

5.3.1.5.1.3 安全配置

通过添加客户端黑、白名单地址，来限制客户端使用情况，该功能可选择开启或不开启。

在集群配置页面的安全配置部分，点击右边的编辑按钮，可对集群进行安全配置。



- 限制客户端地址: 单选框, 不开启、开启, 默认不开启, 开启后只有客户端白名单才可成功访问ADMQ集群服务, 黑名单不可访问。
- 客户端白名单: 可访问集群服务的客户端IP地址, IP格式: 单个IP (如: 192.168.1.1), 多个IP之间以逗号隔开 (如: 192,168.1.1,192.168.1.2或者192.168.1.[1-255],192.168.1.1)
- 客户端黑名单: 不可访问集群服务的客户端IP地址, IP格式如客户端白名单的IP格式

5.3.1.5.1.4 其他配置

在集群配置页面的其他配置部分，点击右边的编辑按钮，可对集群的其他信息进行配置

天燕云 金蝶Apusic分布式消息队列

MQTT服务地址 RABBITMQ服务地址

ROCKETMQ服务地址 描述

资源配置 取消 确认

* 最大租户数量 自动创建主题 不允许 允许

* 最大命名空间数量 自动删除无效主题 不允许 允许

* 最大主题数量

安全配置 取消 确认

限制客户端地址 不开启 开启 客户端白名单

客户端黑名单

其他配置 取消 编辑

事务功能 不开启 开启 * Journal日志存储目录

* 数据存储目录

天燕云 金蝶Apusic分布式消息队列

MQTT服务地址 RABBITMQ服务地址

ROCKETMQ服务地址 描述

资源配置 取消 确认

* 最大租户数量 自动创建主题 不允许 允许

* 最大命名空间数量 自动删除无效主题 不允许 允许

* 最大主题数量

安全配置 取消 确认

限制客户端地址 不开启 开启 客户端白名单

客户端黑名单

其他配置 取消 确认

事务功能 不开启 开启 * Journal日志存储目录

* 数据存储目录

- 事务功能：单选框，开启、不开启，默认不开启
- Journal日志存储目录：日志的存储目录，可修改
- 数据存储目录：数据的存储目录，可修改

5.3.1.5.2 RocketMQ的配置页面

RocketMQ集群的配置页面，包括：基础配置、安全配置、存储设置

天燕云 金蝶Apusic分布式消息队列

集群管理 / 集群配置

集群配置

基础配置				编辑
集群名称	dddasd	数据接入地址		
描述				

安全配置				编辑
自动创建主题	不允许	自动创建订阅组	不允许	
开启认证	开启	客户端白名单	172.24.4.126	

存储设置				编辑
* 文件删除时间	1	* 磁盘使用上限 (%)	85	
* 文件保留时间 (小时)	72			

5.3.1.5.2.1 基础配置

在集群配置页面的基础配置部分，点击右边的编辑按钮，可配置基础信息

天燕云 金蝶Apusic分布式消息队列

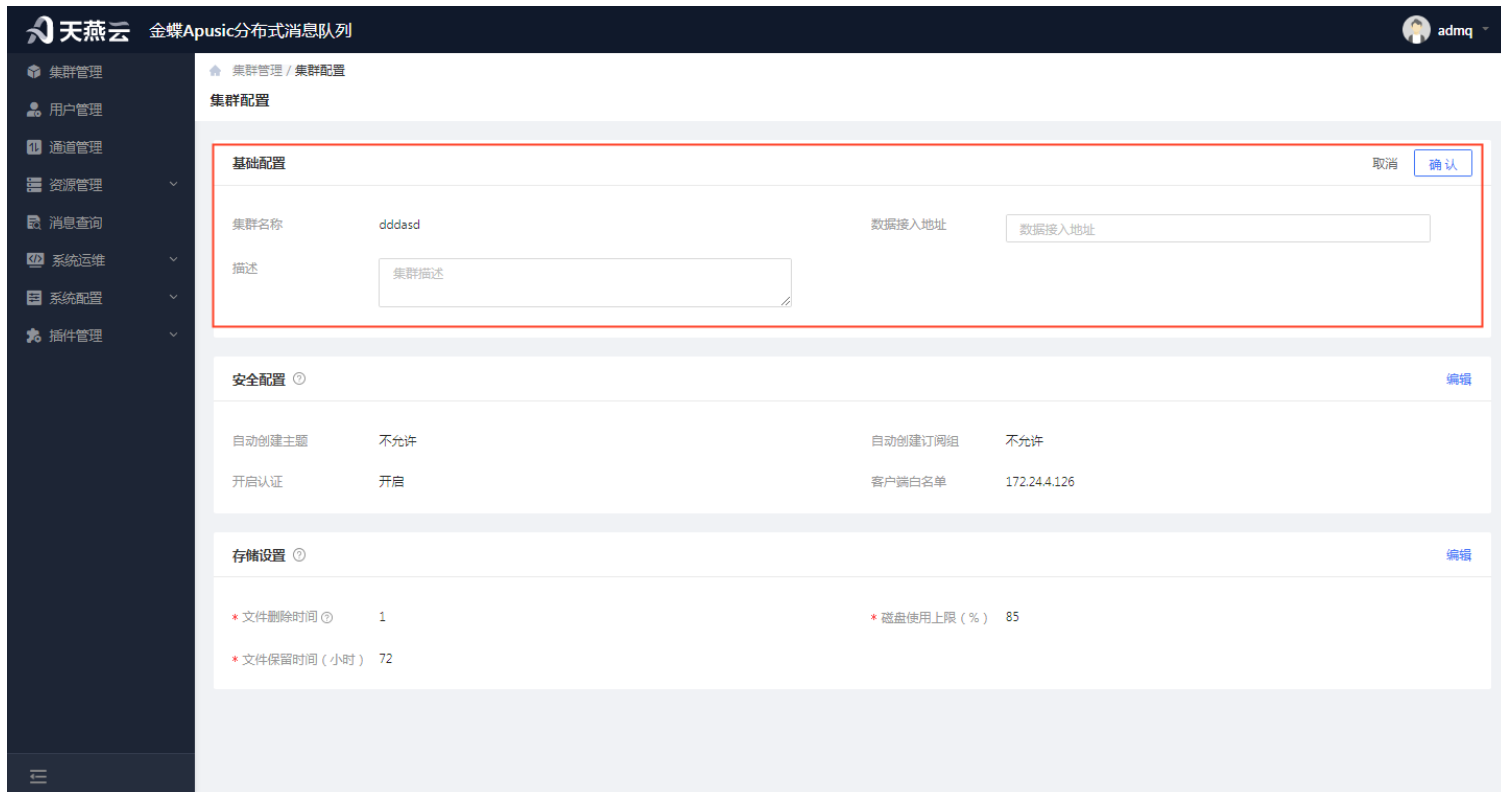
集群管理 / 集群配置

集群配置

基础配置				编辑
集群名称	dddasd	数据接入地址		
描述				

安全配置				编辑
自动创建主题	不允许	自动创建订阅组	不允许	
开启认证	开启	客户端白名单	172.24.4.126	

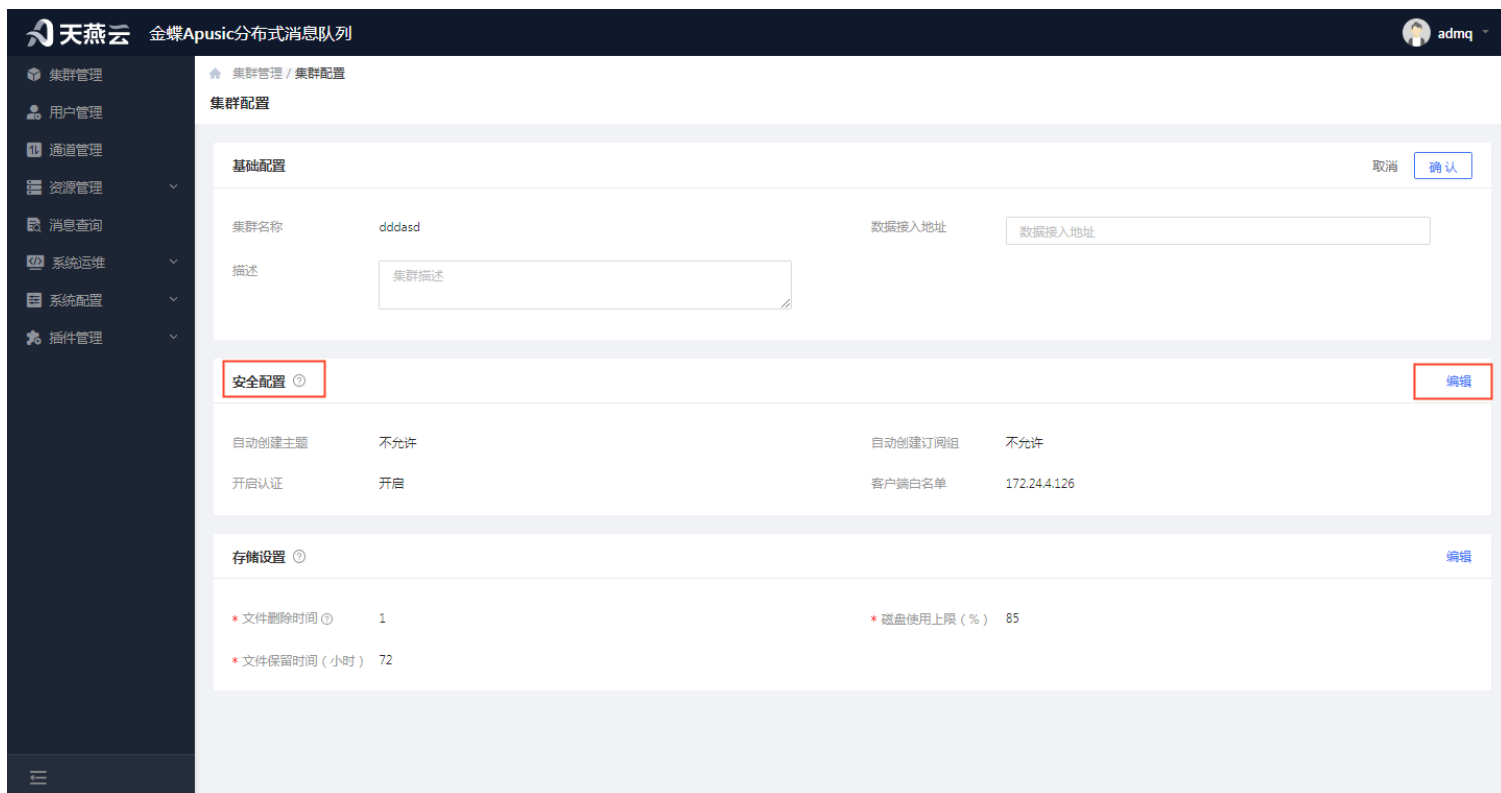
存储设置				编辑
* 文件删除时间	1	* 磁盘使用上限 (%)	85	
* 文件保留时间 (小时)	72			

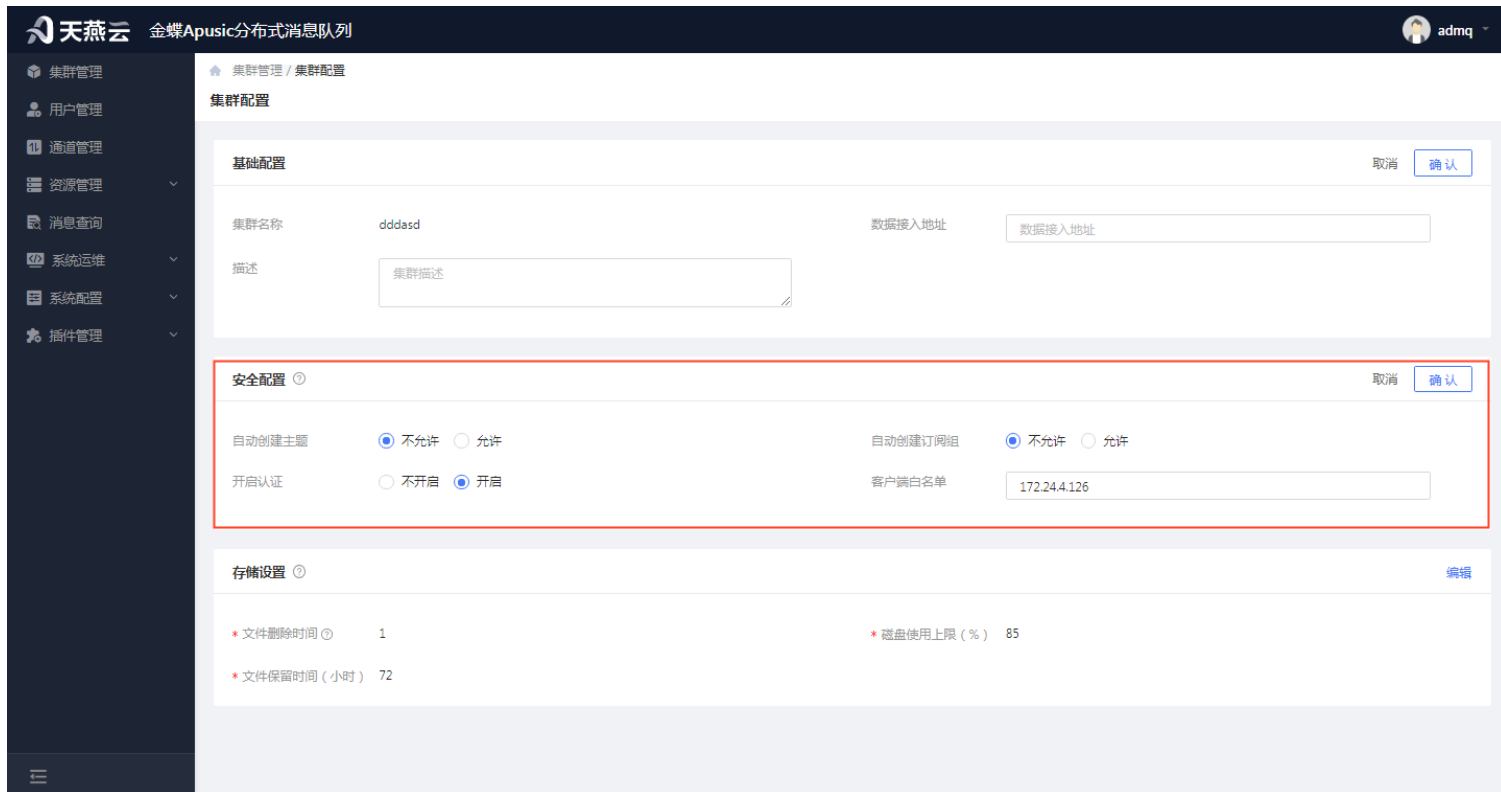


- 集群名称：ROcketMQ集群的唯一标识，不可变
- 数据接入地址：集群计算程序服务地址，可编辑
- 描述：集群详细信息介绍，可编辑

5.3.1.5.2.2 安全配置

在集群配置页面的安全配置部分，点击右边的编辑按钮，可配置安全信息

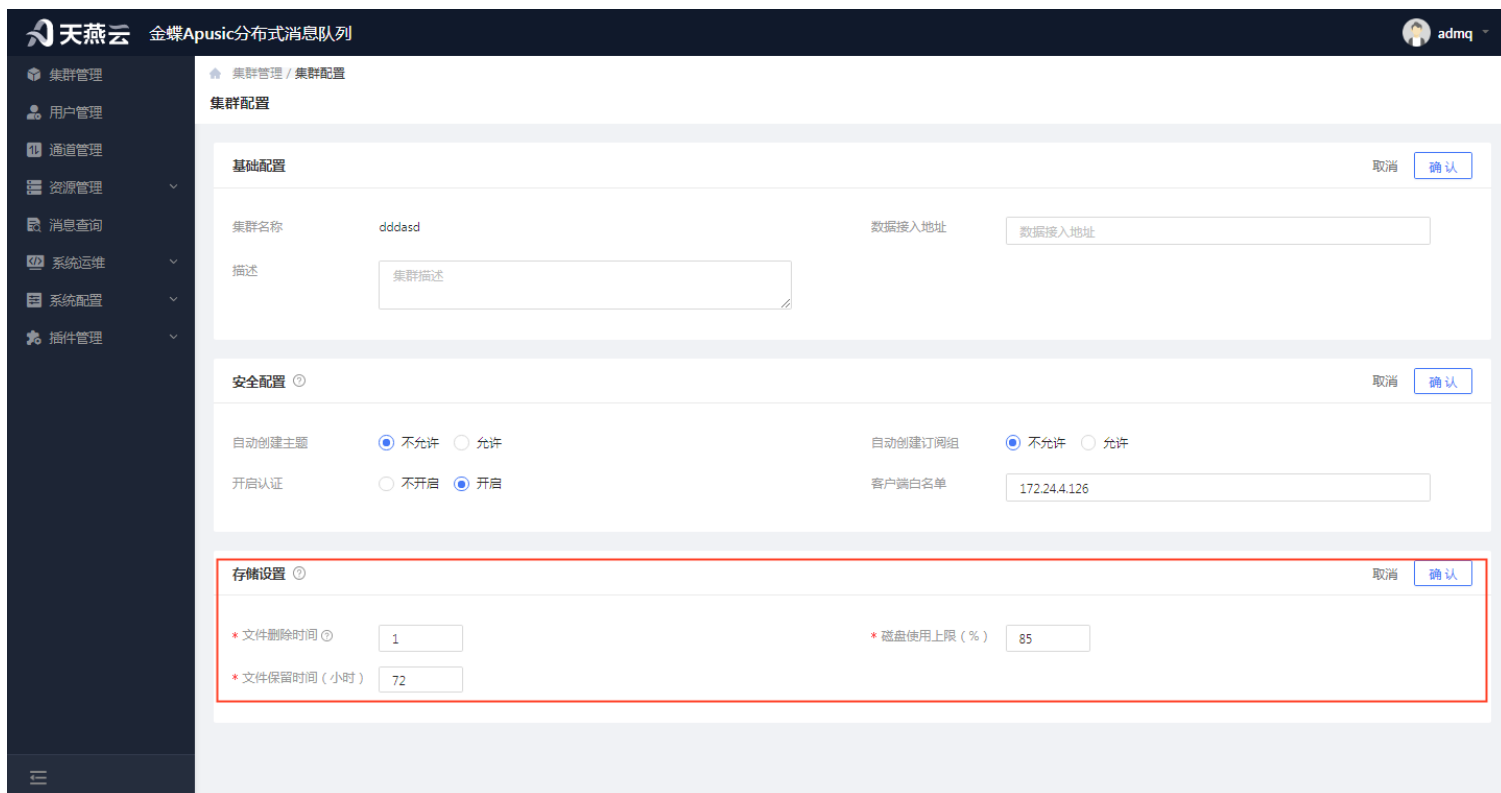
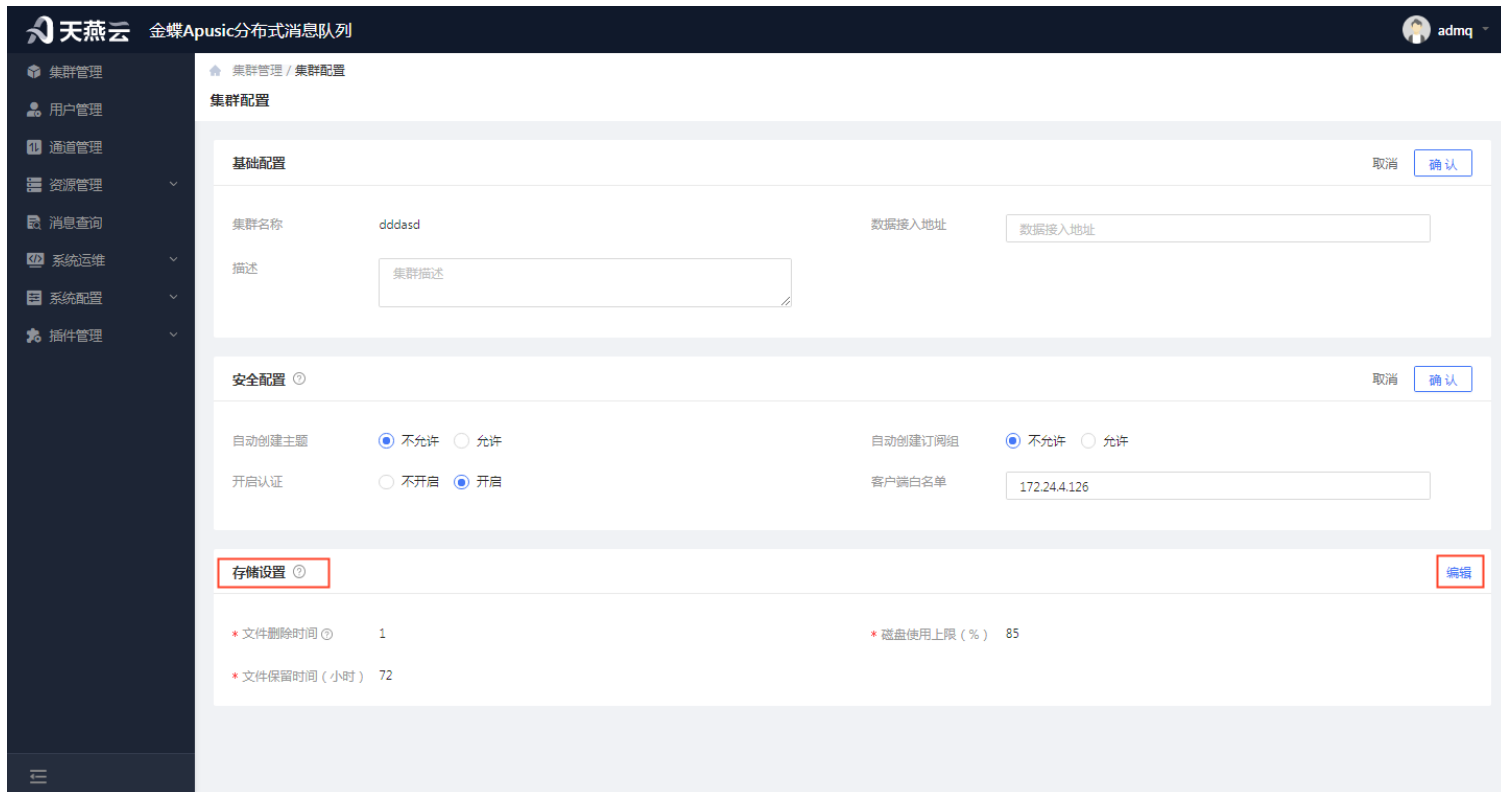




- 自动创建主题：单选框，允许、不允许，默认不允许
- 开启认证：单选框，开启、不开启，默认开启
- 自动创建订阅组：单选框，允许、不允许，默认不允许
- 客户端白名单：客户端白名单IP，可编辑

5.3.1.5.2.3 存储设置

在集群配置页面的存储设置部分，点击右边的编辑按钮，可配置存储信息。

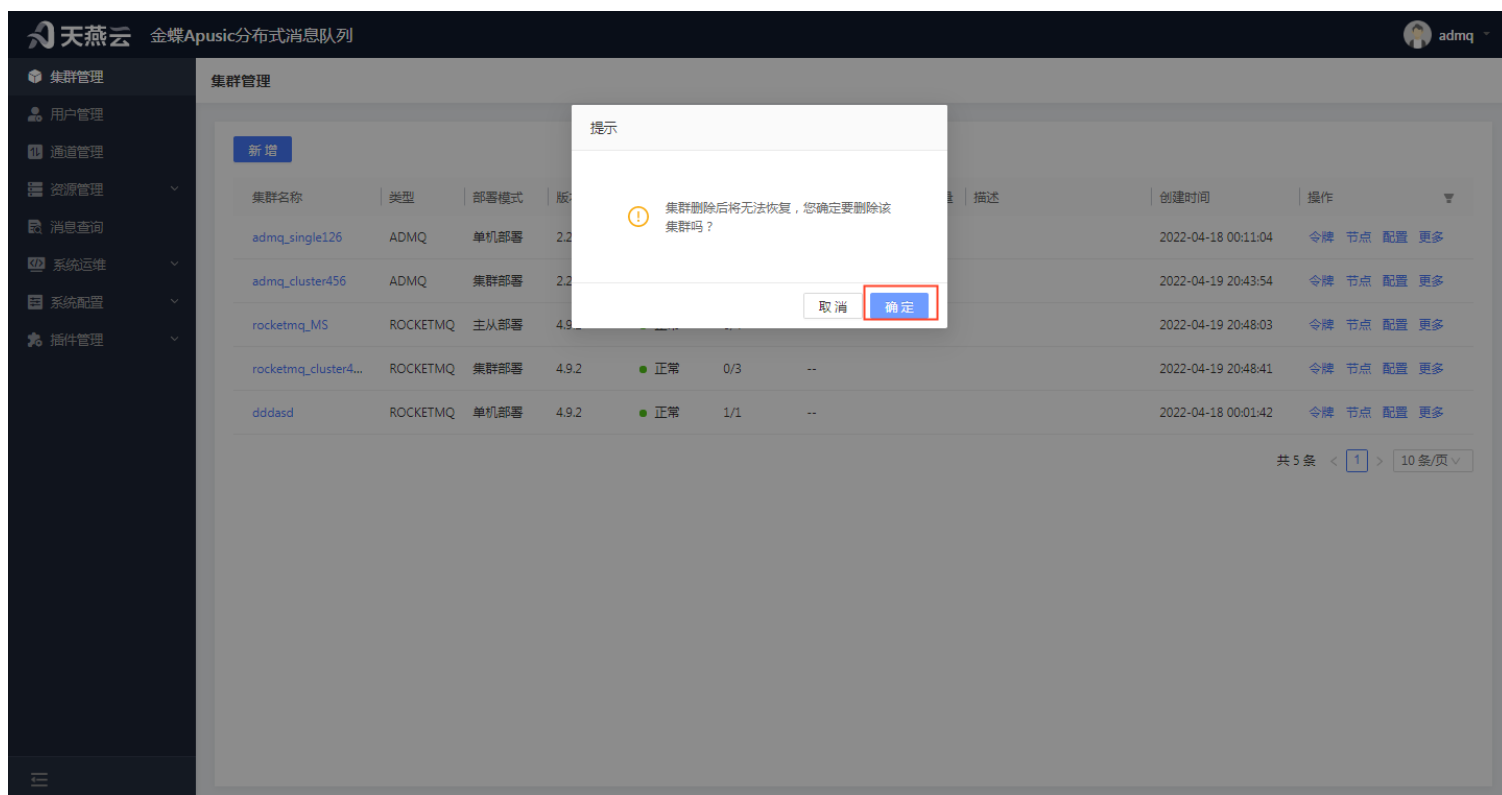
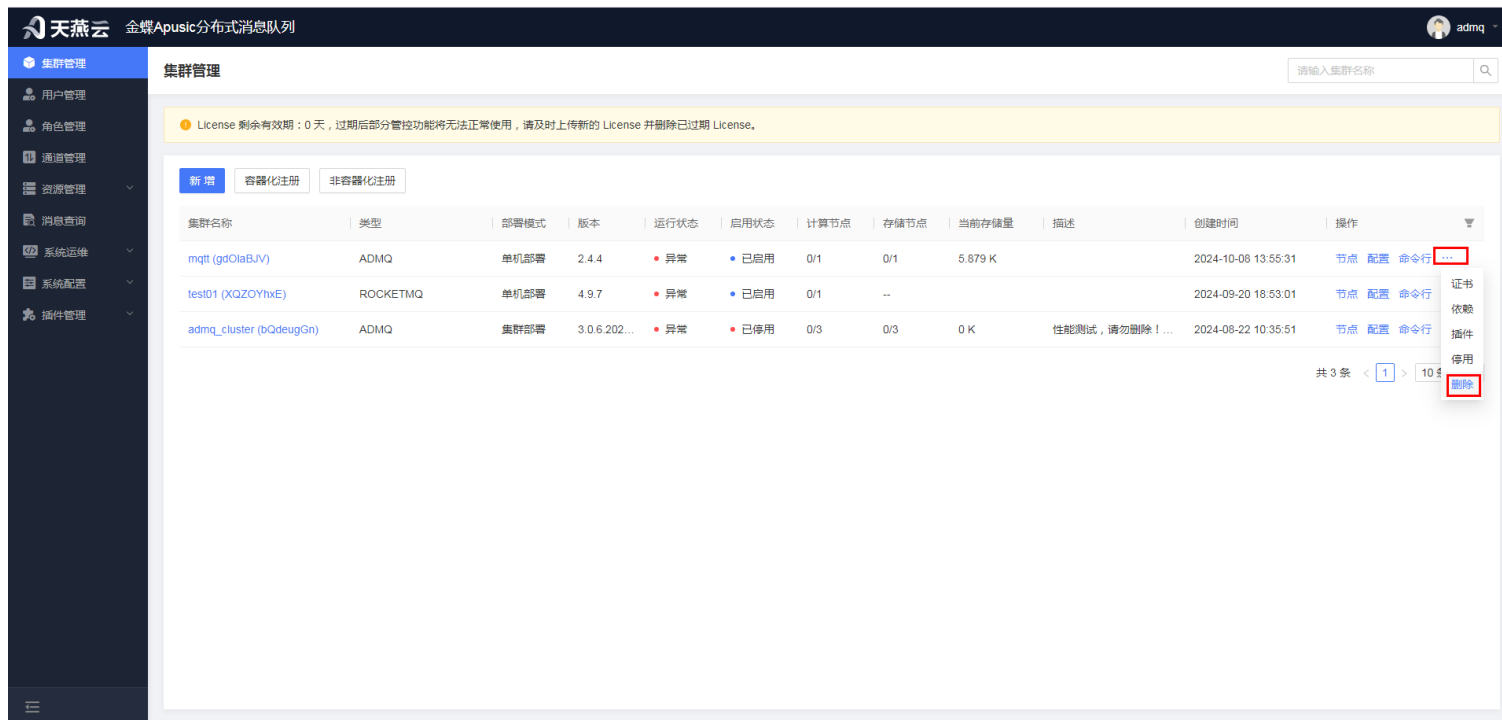


- 文件删除时间：每天固定整点删除文件，默认是1，可修改
- 磁盘使用上限：文件存储占总磁盘的百分比，默认85，可修改
- 文件保留时间（小时）：文件保留在磁盘的时间，默认72小时，可修改

5.3.1.6 删除集群

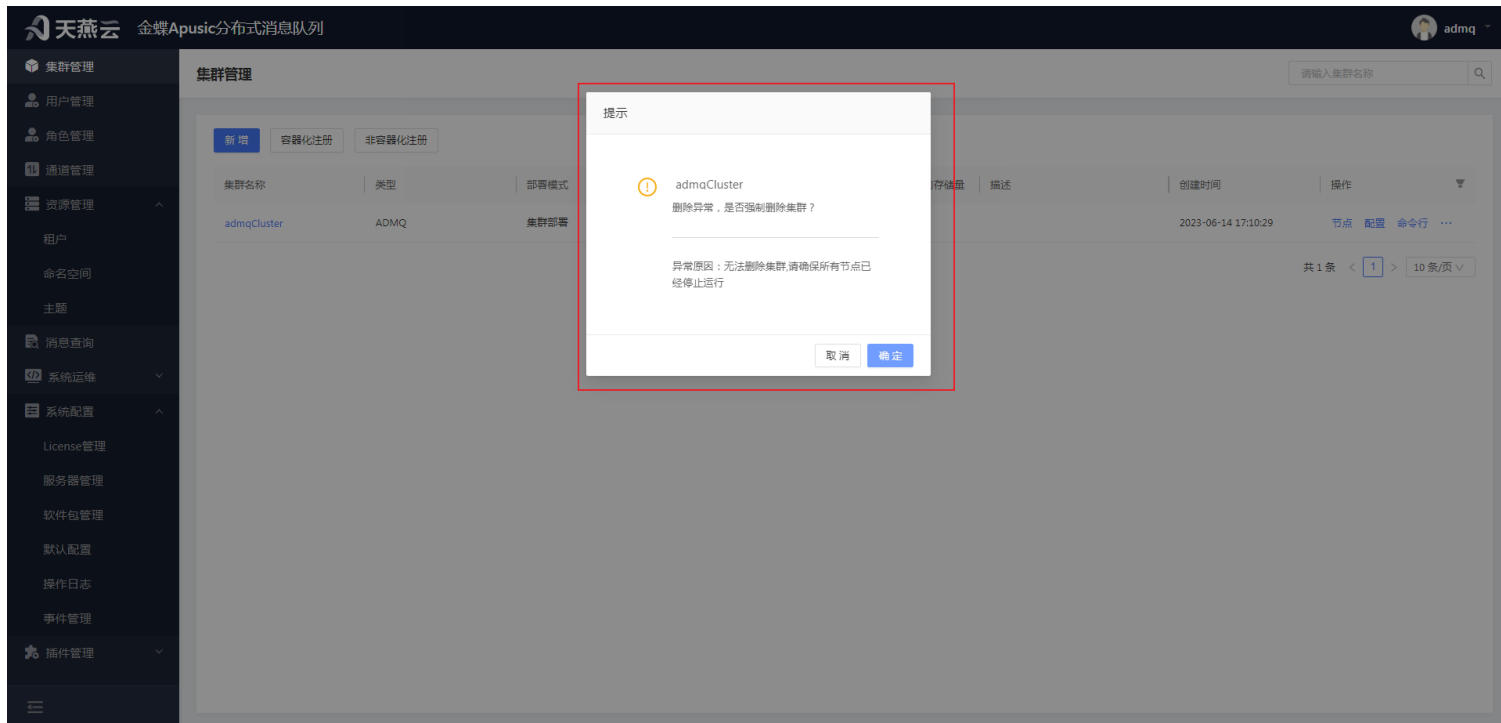
(1) 正常删除集群

进入【集群管理】，在集群管理页面选择需要删除的集群信息，点击【...】-【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，删除集群信息。注意：集群有节点启动不可删除，需停止所有节点，才可删除；集群删除后，集群下所有的资源信息（租户、命名空间、主题）也被删除。



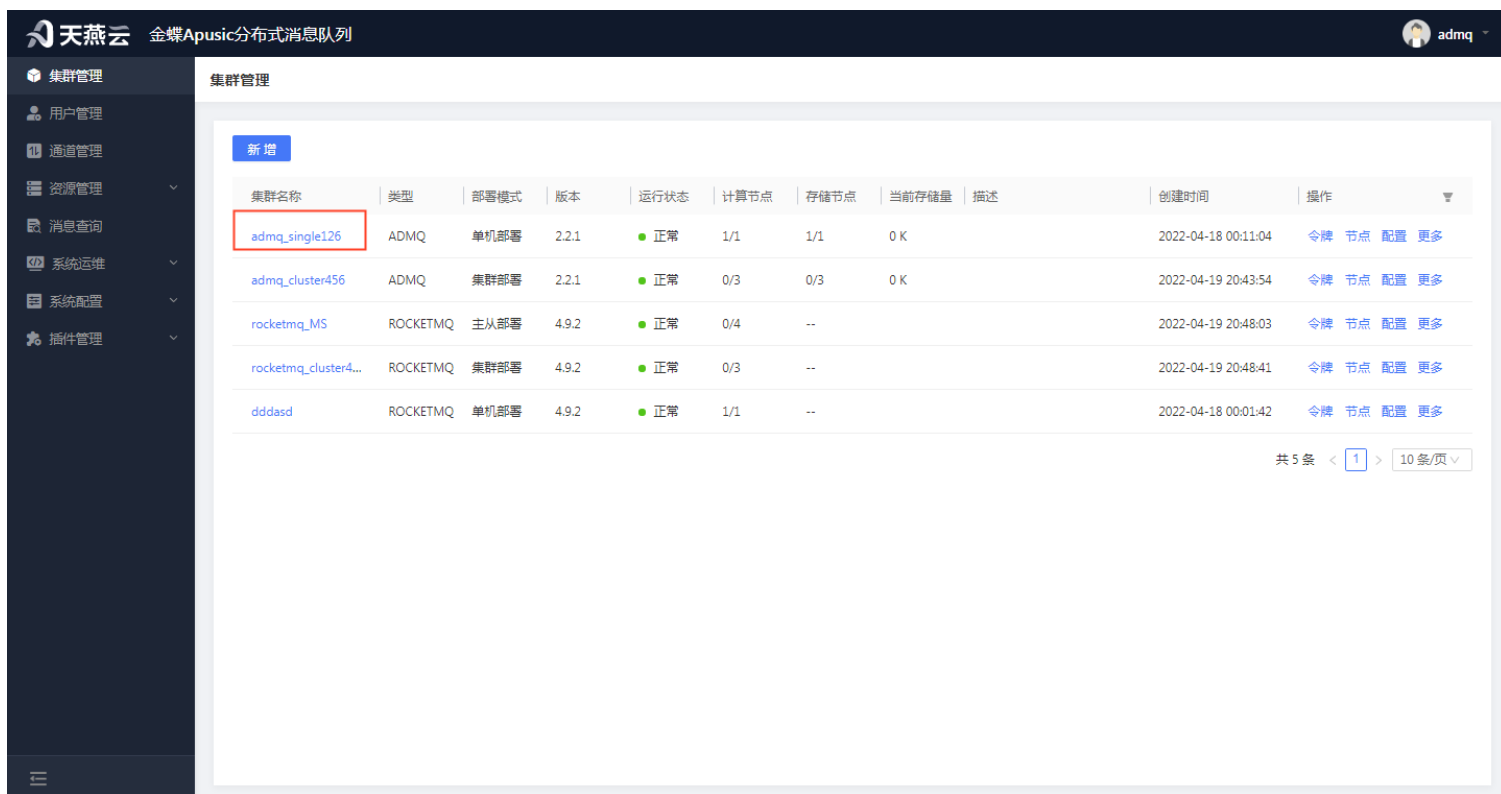
(2) 强制删除集群

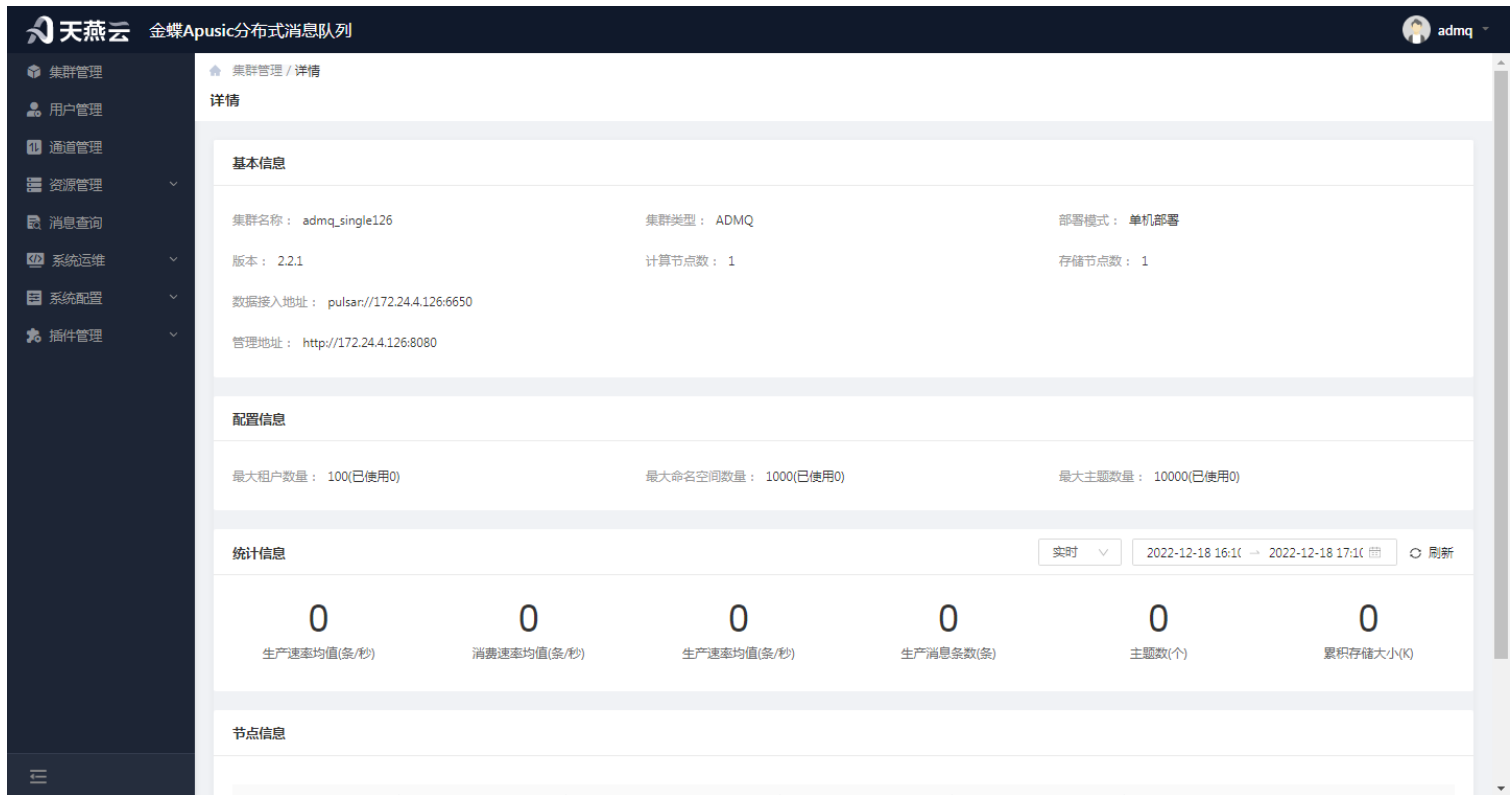
当远程服务器出现问题后，该ADMQ集群就没有任何作用，这时可以使用强制删除集群功能，将该集群从管控台数据库中删除。具体操作过程：正常删除集群弹窗界面点击【确定】按钮后会出现强制删除集群的弹窗界面，点击【确定】按钮可强制删除集群。



5.3.1.7 查询集群详细信息

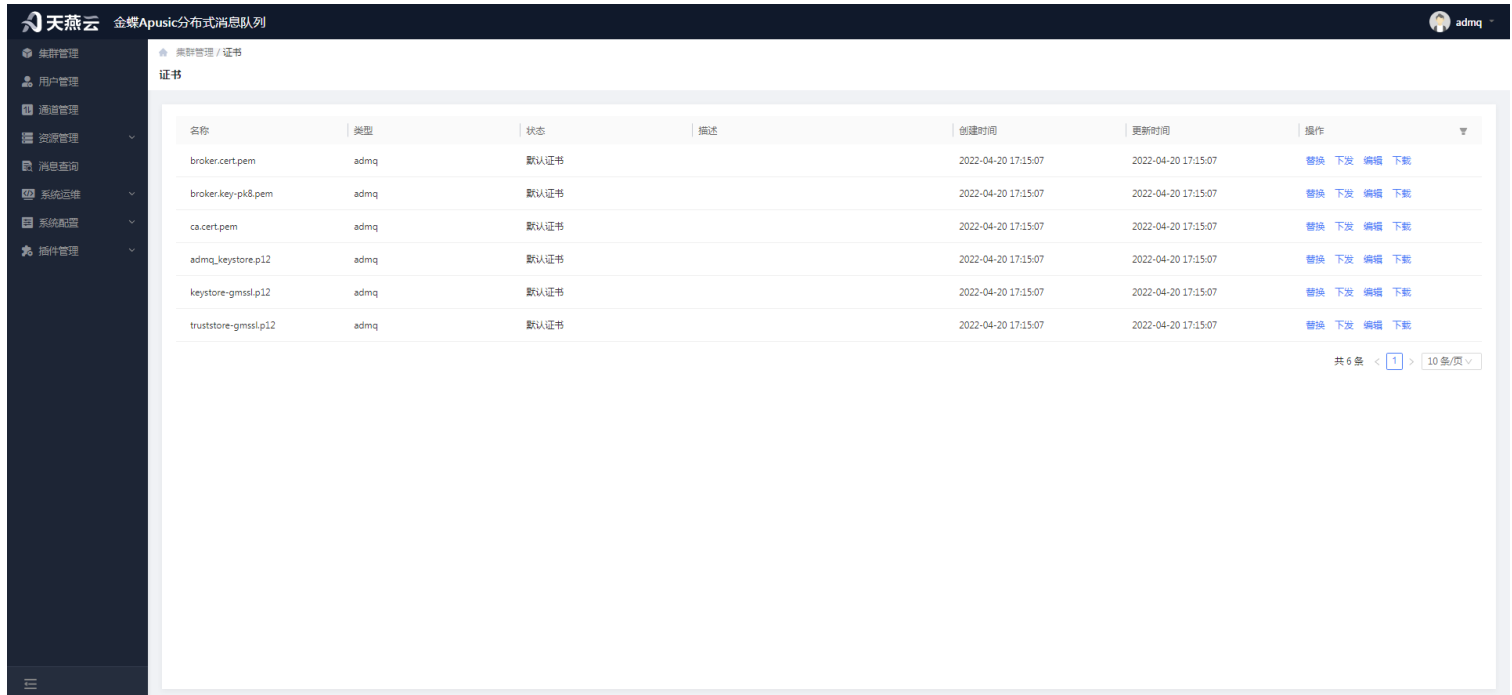
集群详情用户监控集群的一些信息，如基本信息、配置信息、统计信息、节点信息等，进入【集群管理】，在集群管理页面，选择需要查看集群详细信息的集群，双击集群名称，进入集群管理详情页面。





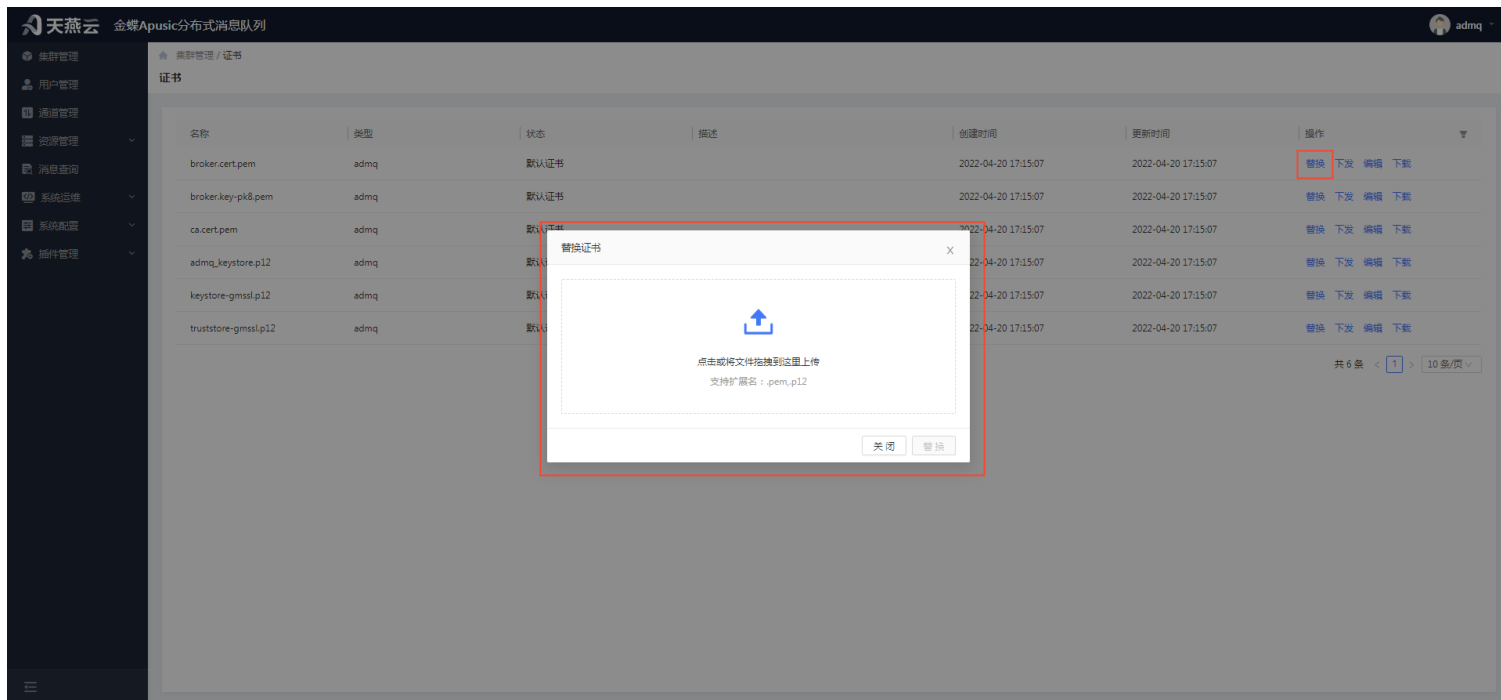
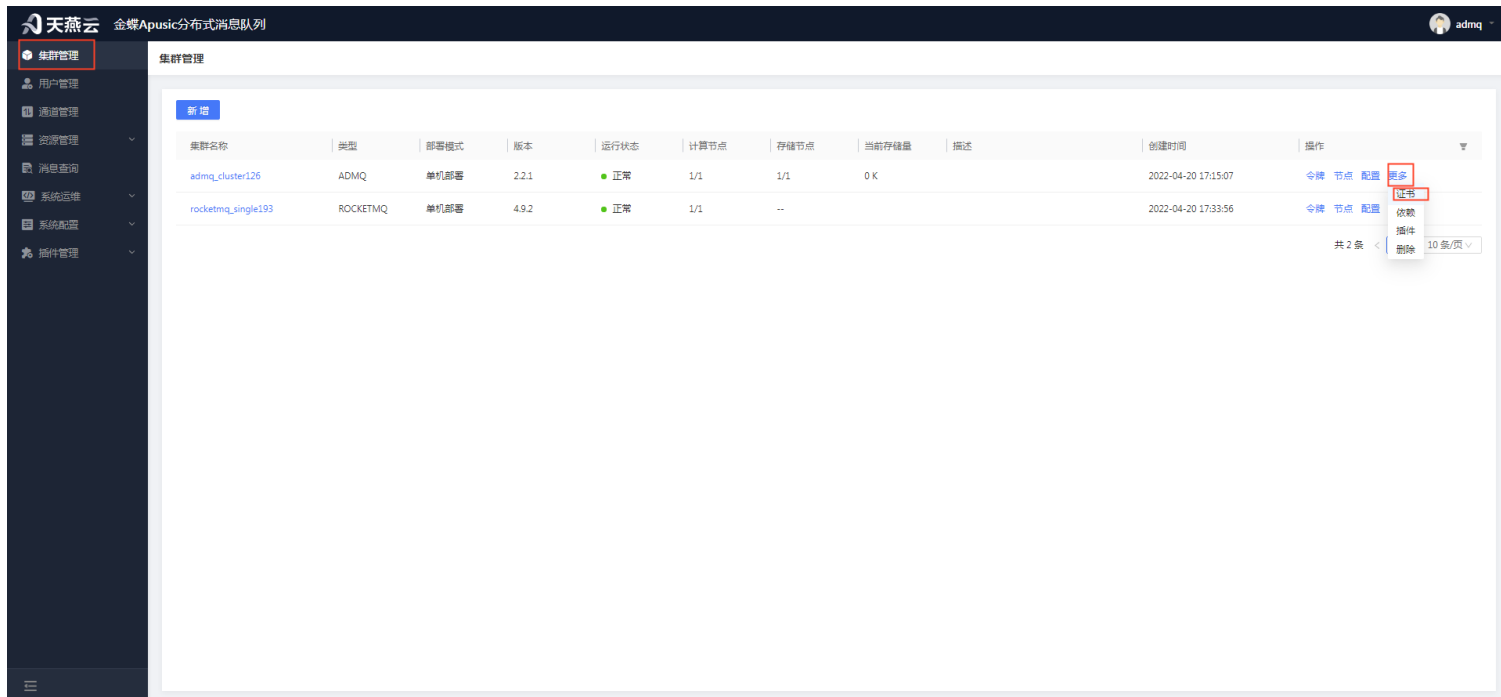
5.3.1.8 证书管理

管理控制台支持对证书进行管理，默认证书为admq软件包下的证书，证书管理可替换证书、下发证书、编辑证书、下载证书。



5.3.1.8.1 替换证书

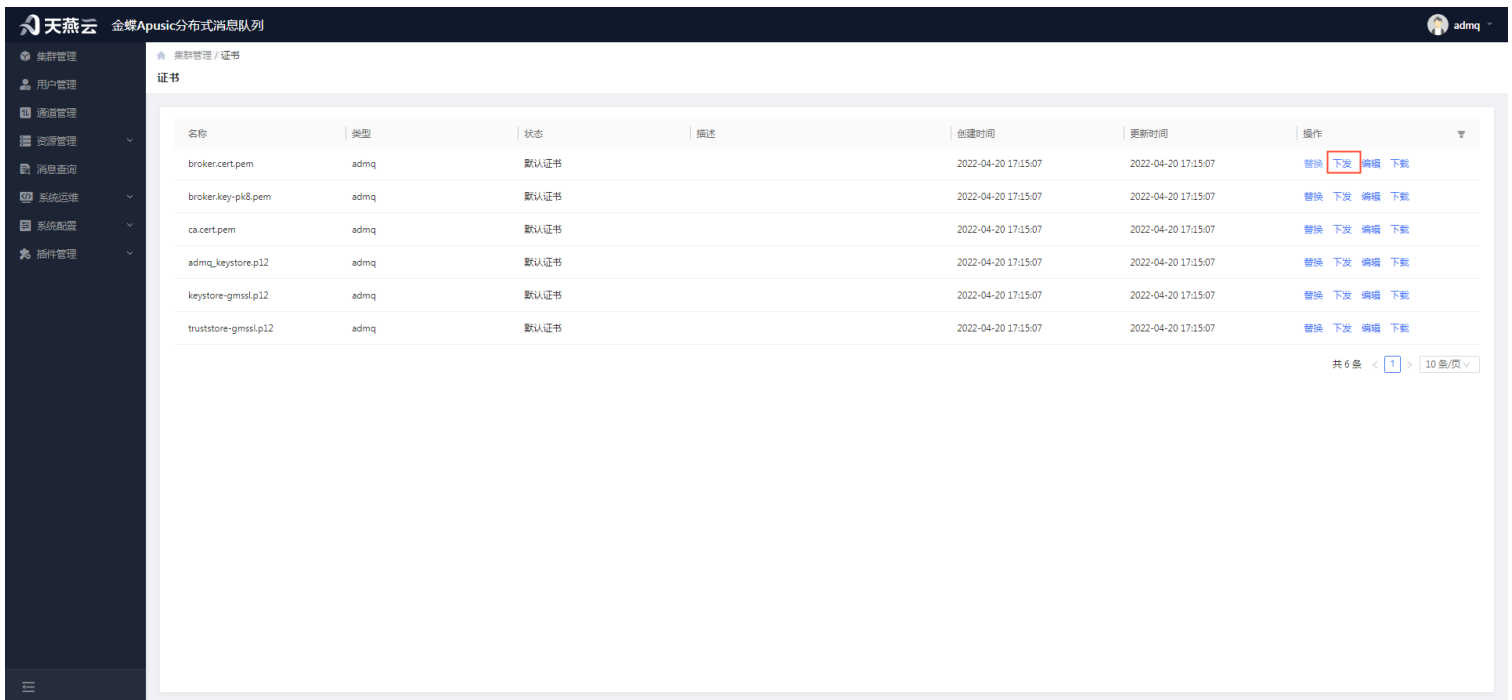
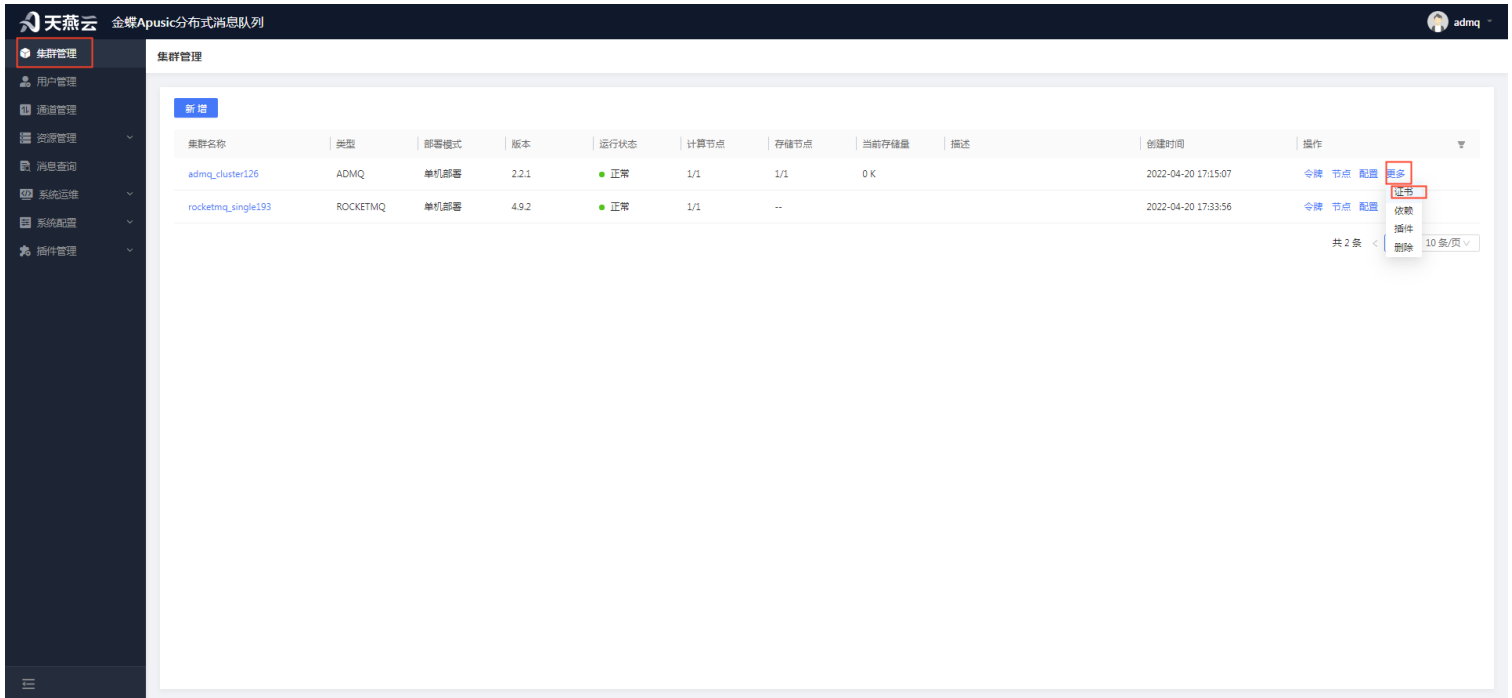
进入集群管理，选择需要管理证书的集群，点击【更多】按钮->【证书】按钮，在证书页面，选择需替换的证书，点击【替换】按钮后上传或拖曳证书，可以替换证书。注意：替换的证书需要下发才能下发到集群各个节点下，且下发后应重启集群节点。



- 证书类型：只能上传.pem和p12文件的证书
- 备注：替换的证书，只是内容被替换，证书名称不会被修改。

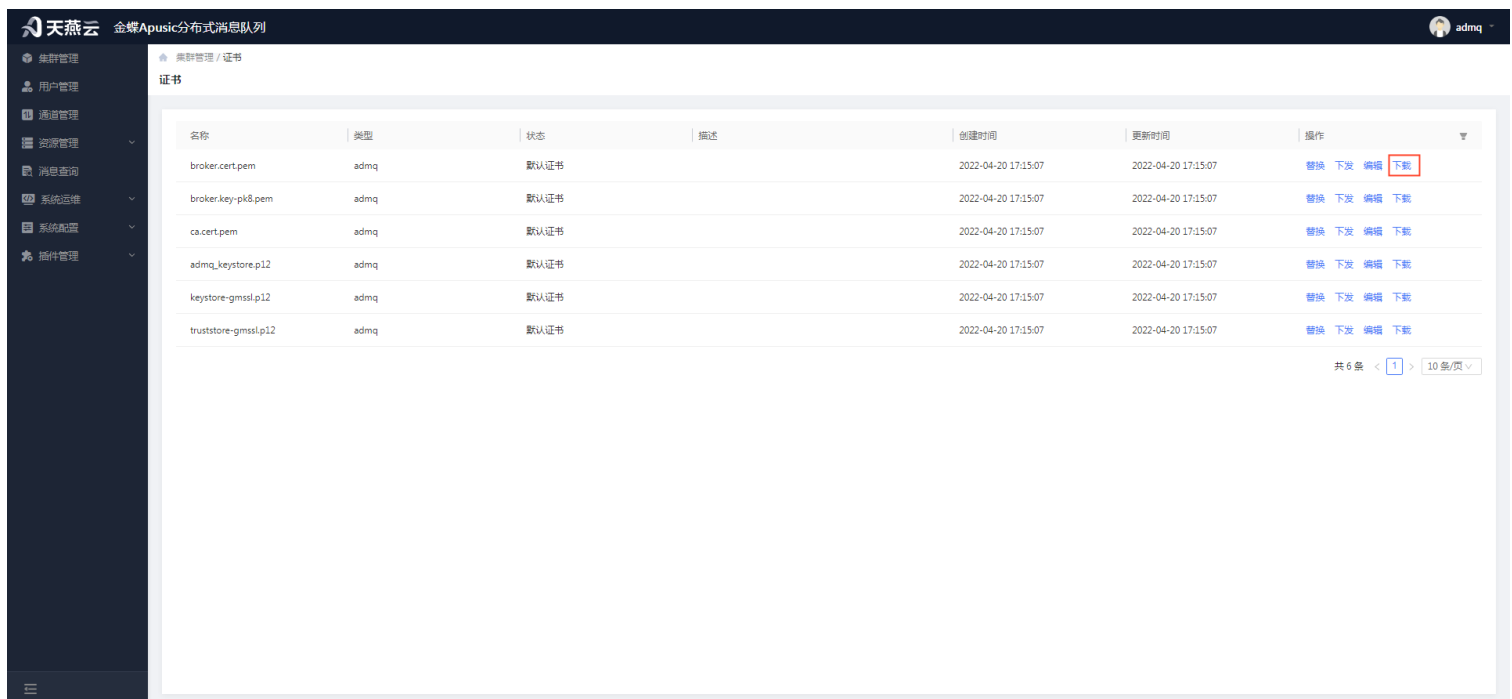
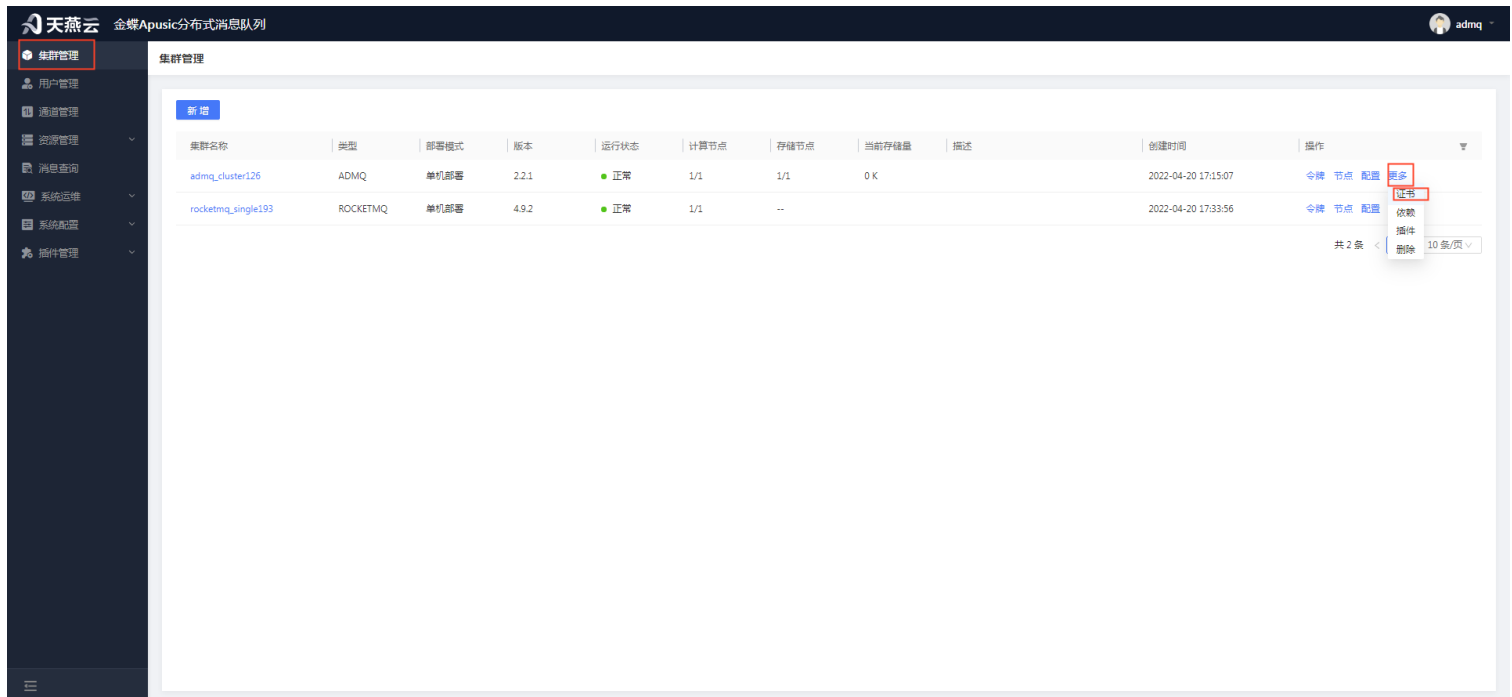
5.3.1.8.2 下发证书

进入集群管理，选择需要管理证书的集群，点击【更多】按钮->【证书】按钮，在证书页面，选择需下发的证书，点击【下发】按钮，可下发证书到已部署节点服务器上。注意：下发证书应成功部署集群后才能下发，下发后需生效要重启集群节点。



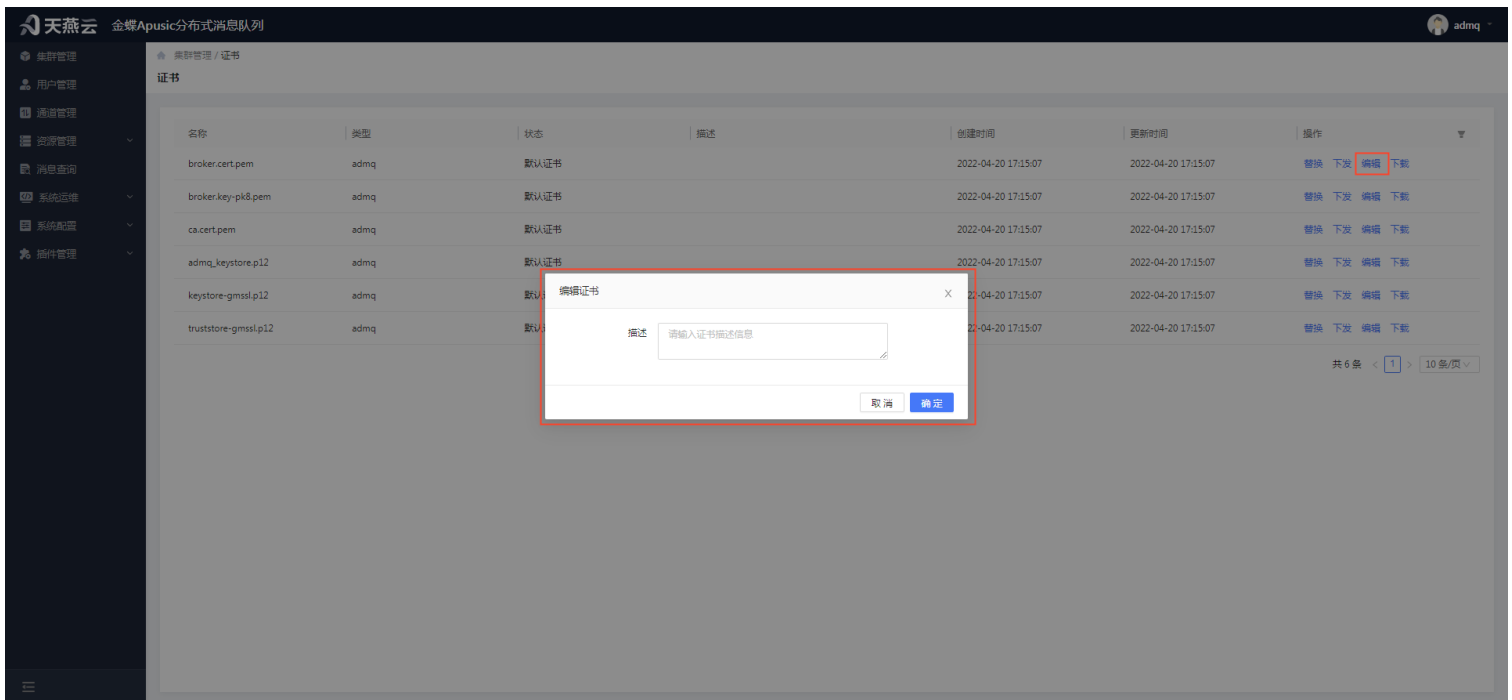
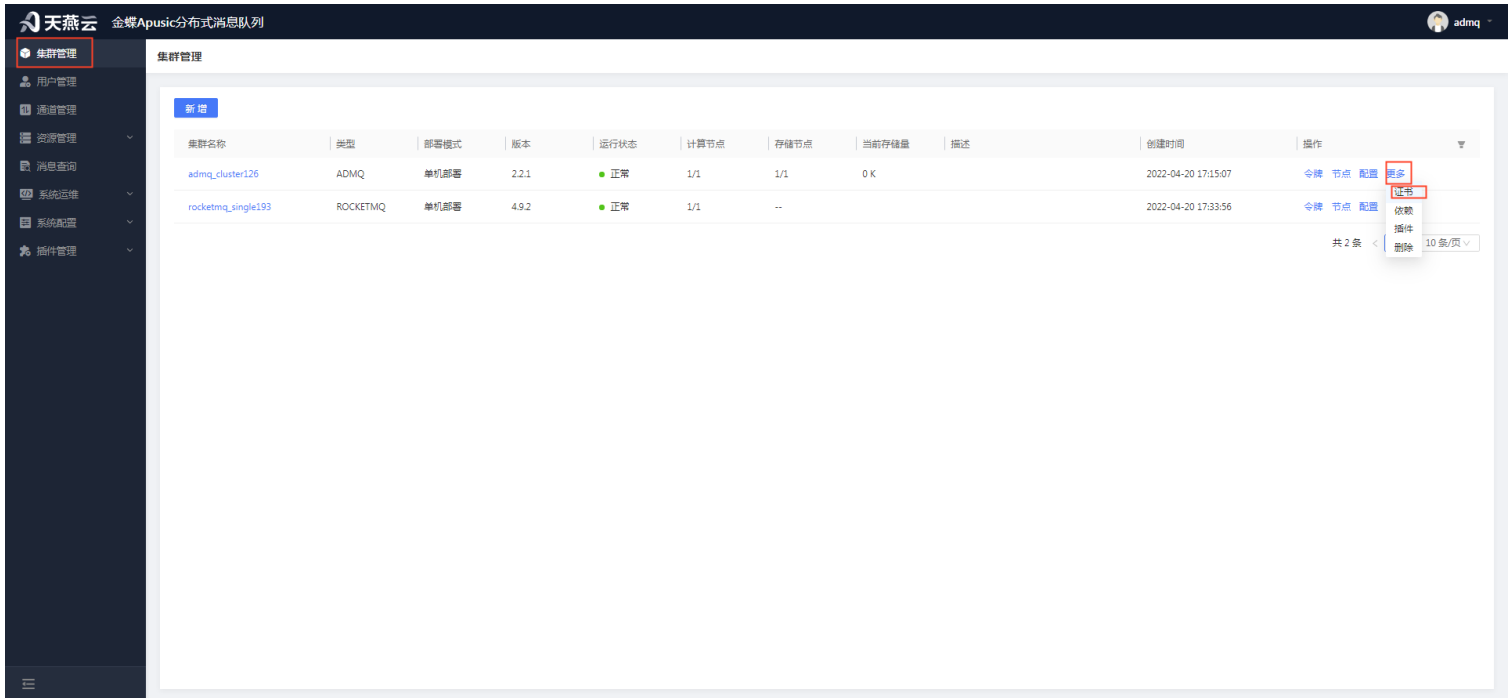
5.3.1.8.3 下载证书

进入集群管理，选择需要管理证书的集群，点击【更多】按钮->【证书】按钮，在证书页面，选择需下载的证书，点击【下载】按钮，可下载证书到已部署节点服务器上。注意：下载证书是下载管理控制台上的证书（非节点服务器上的证书）



5.3.1.8.4 编辑证书

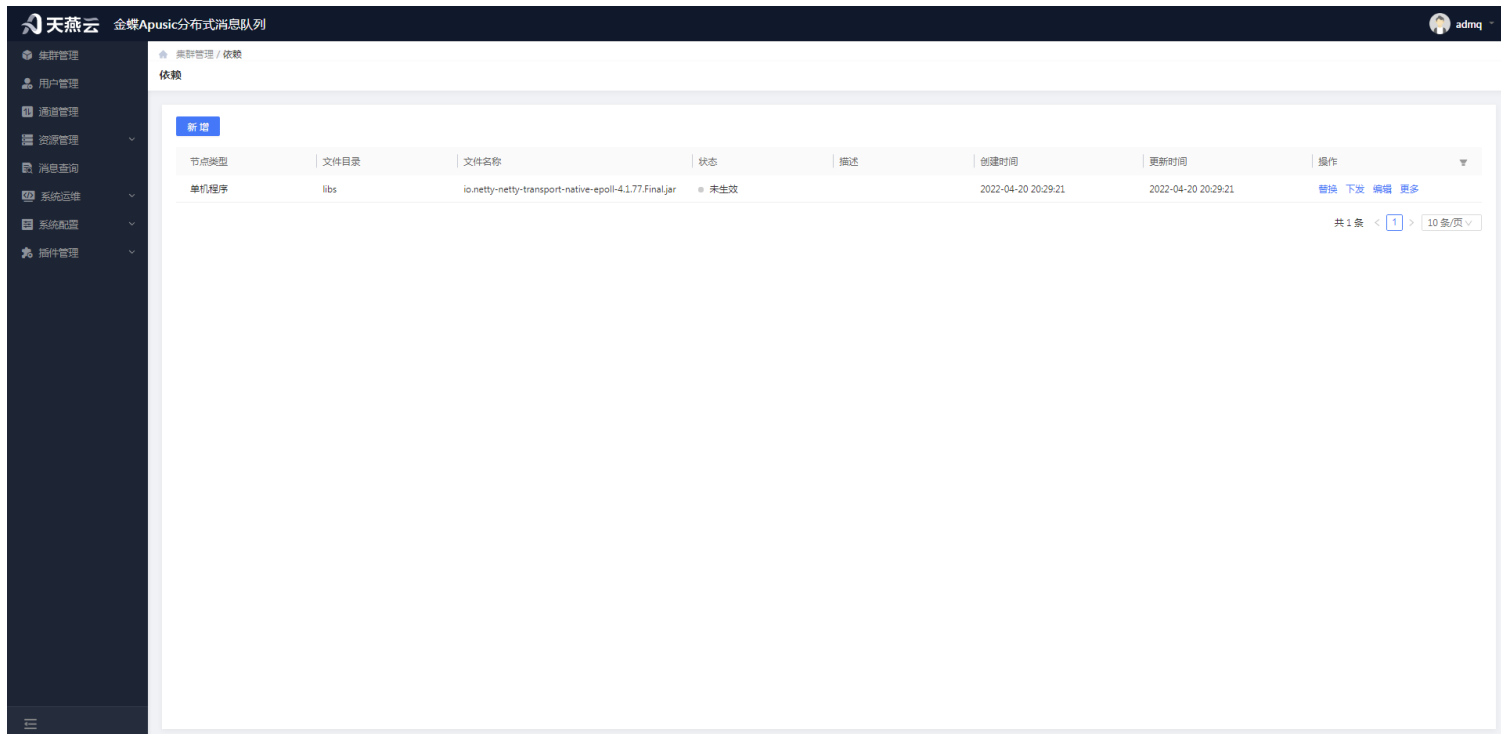
进入集群管理，选择需要管理证书的集群，点击【更多】按钮->【证书】按钮，在证书页面，选择需编辑的证书，点击【编辑】按钮，可编辑证书信息。



- 描述：证书描述信息

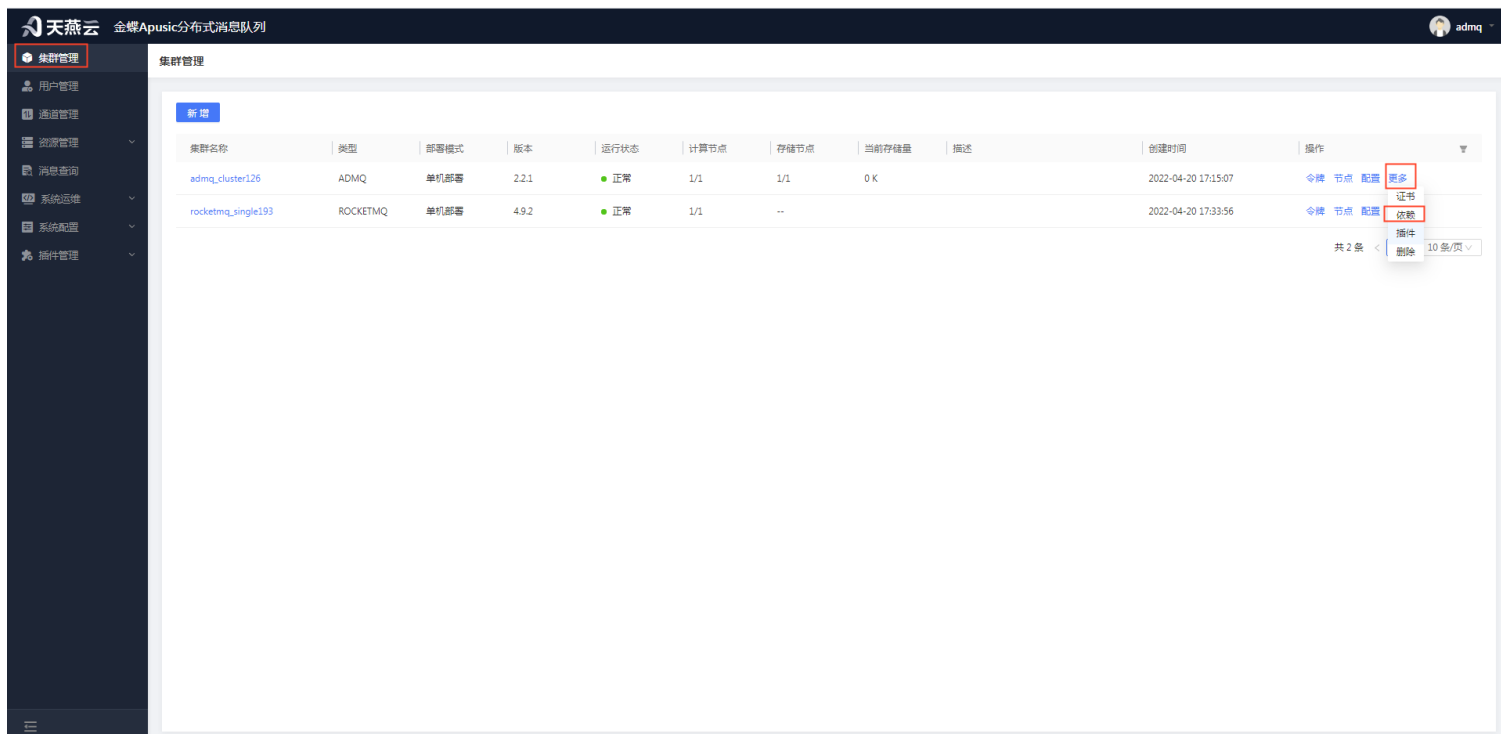
5.3.1.9 依赖管理

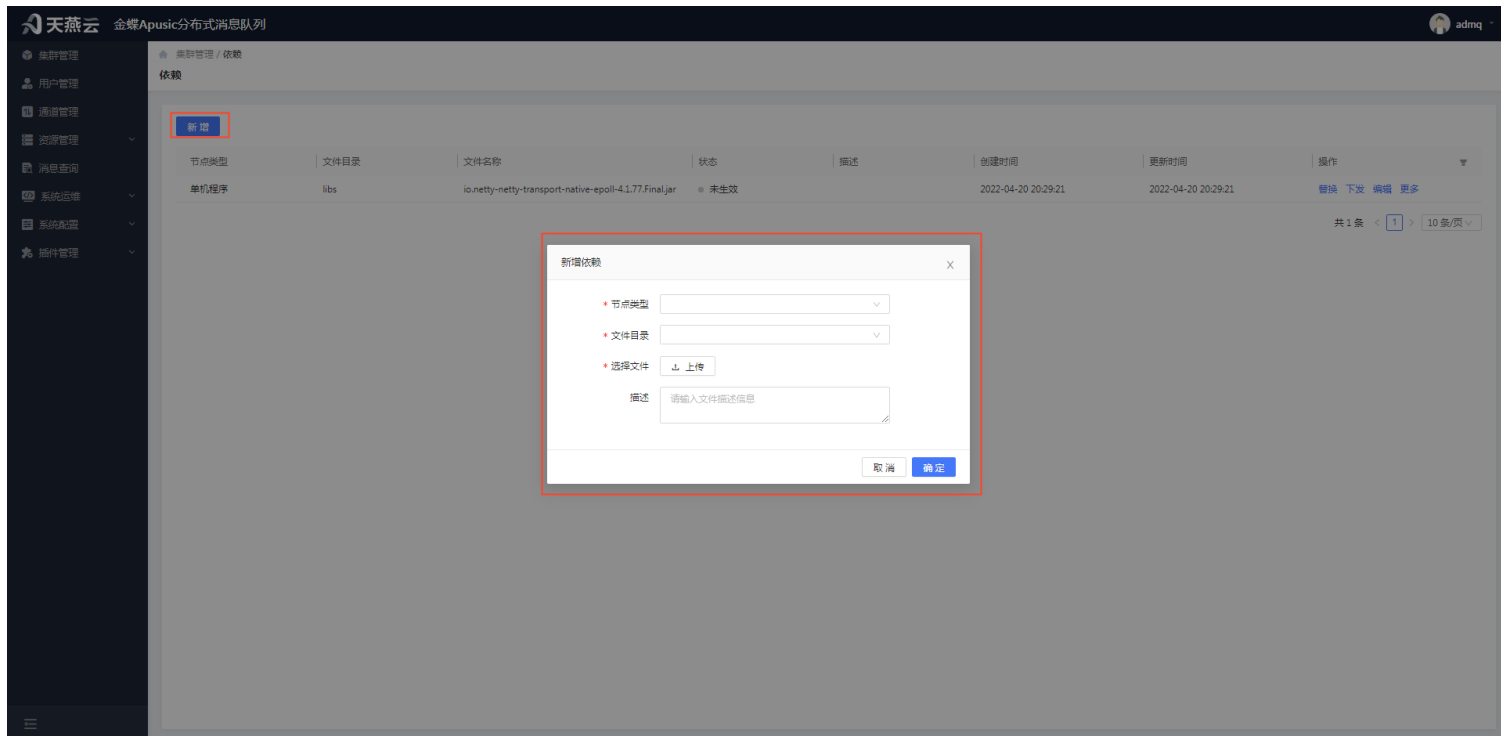
为方便对已部署的集群进行升级，管理控制台支持对依赖进行管理，应用场景主要用于核心引擎服务（软件包）升级。依赖管理可新增依赖、下发依赖、替换依赖、编辑依赖、下载依赖、删除依赖。



5.3.1.9.1 新增依赖

进入集群管理，选择需要管理依赖的集群，点击【更多】按钮->【依赖】按钮，在依赖页面，点击左上角【新增】按钮，可新增依赖信息。注意：新增的依赖需要下发才能更新到集群各个节点下，且下发后应重启集群节点，才会生效。

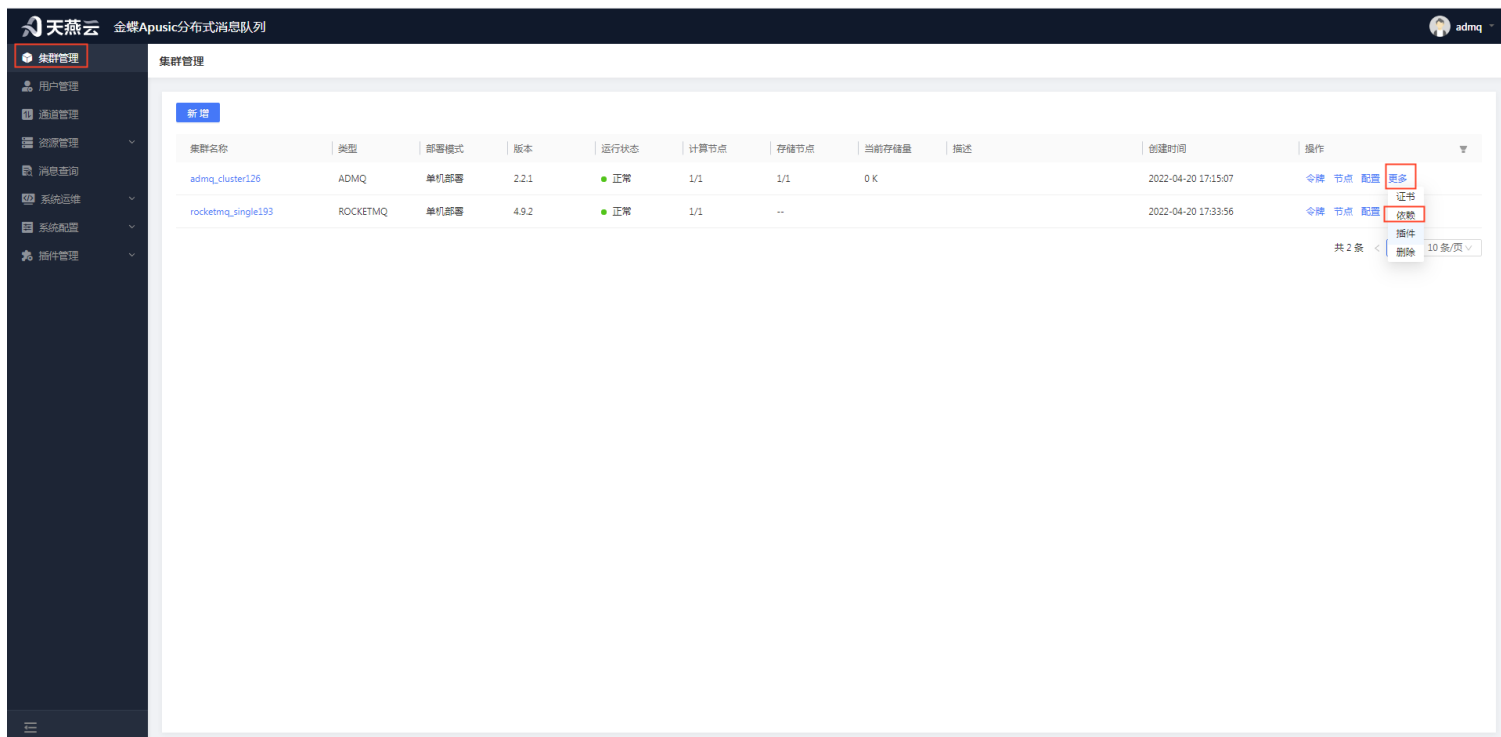


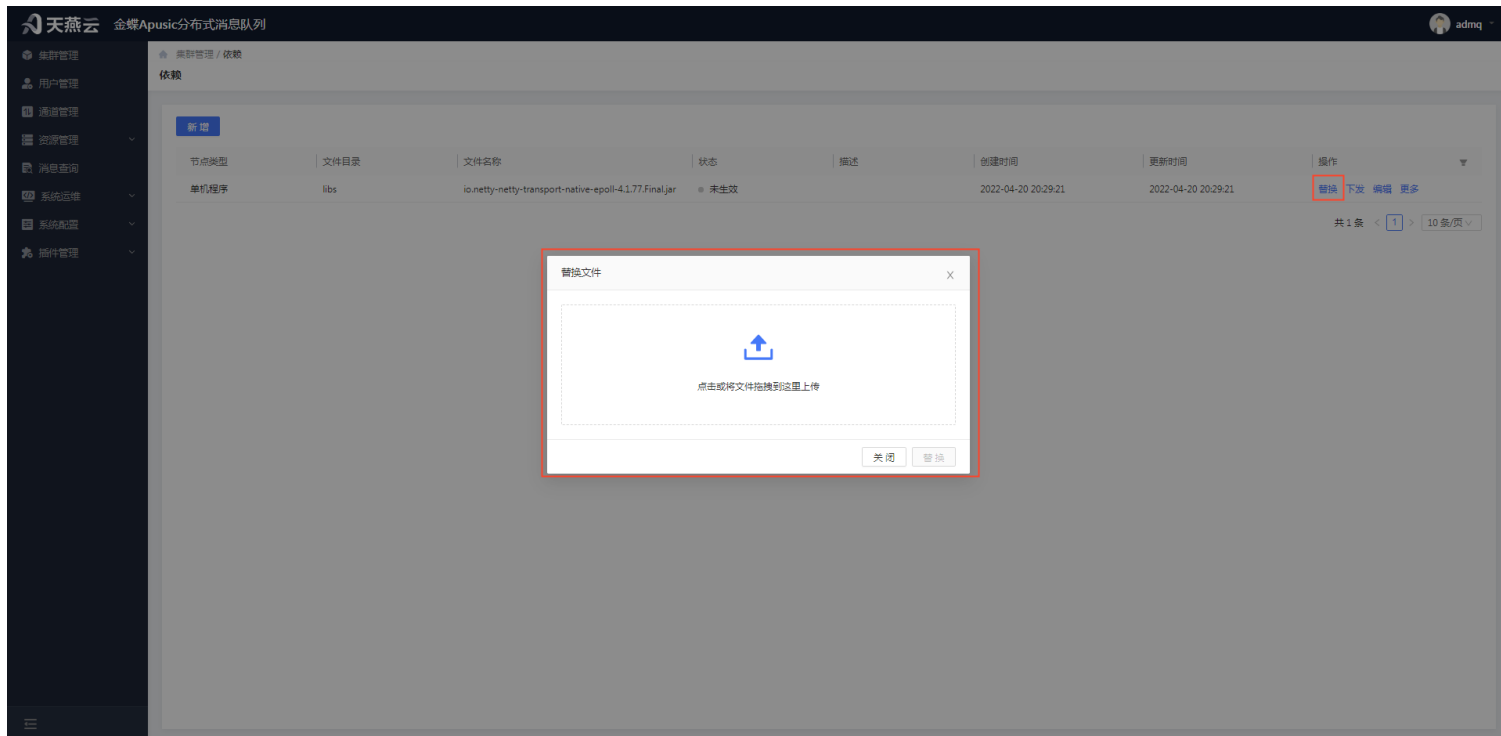


- 节点类型：节点的类型，下拉选项。若是单机，有单机程序；若是集群，有协调器、全局协调器、存储程序、计算程序
- 文件目录：依赖的上级目录，默认为软件包下的目录，有bin、protocols、config、libs、licenses
- 选择文件：要上传的依赖，可拖拽
- 描述：依赖文件的描述信息

5.3.1.9.2 替换依赖

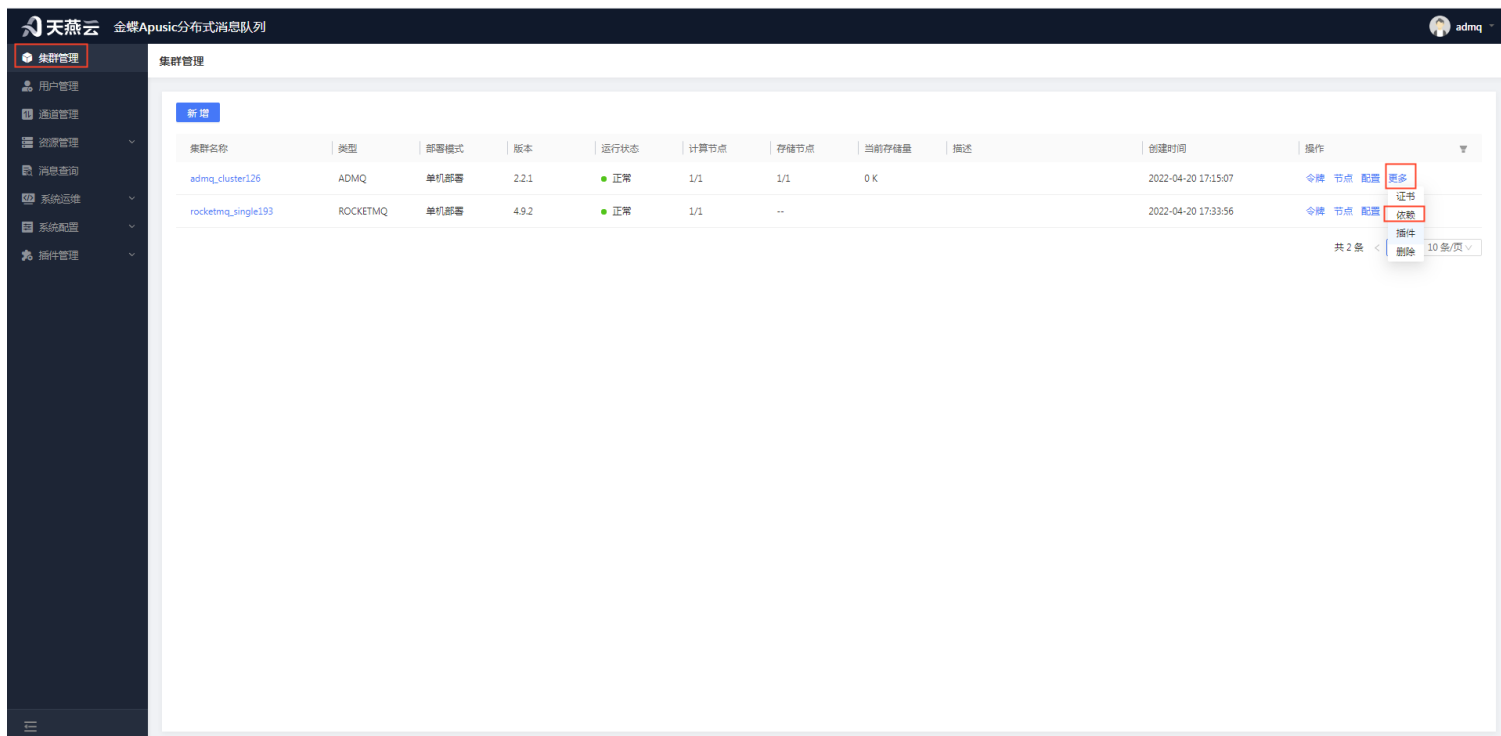
进入集群管理，选择需要管理依赖的集群，点击【更多】按钮->【依赖】按钮，在依赖页面，选择需替换的依赖，点击【替换】按钮后上传或拖拽依赖，可以替换依赖。注意：替换的依赖需要下发才能更新到集群各个节点下，且下发后应重启集群节点，才会生效。

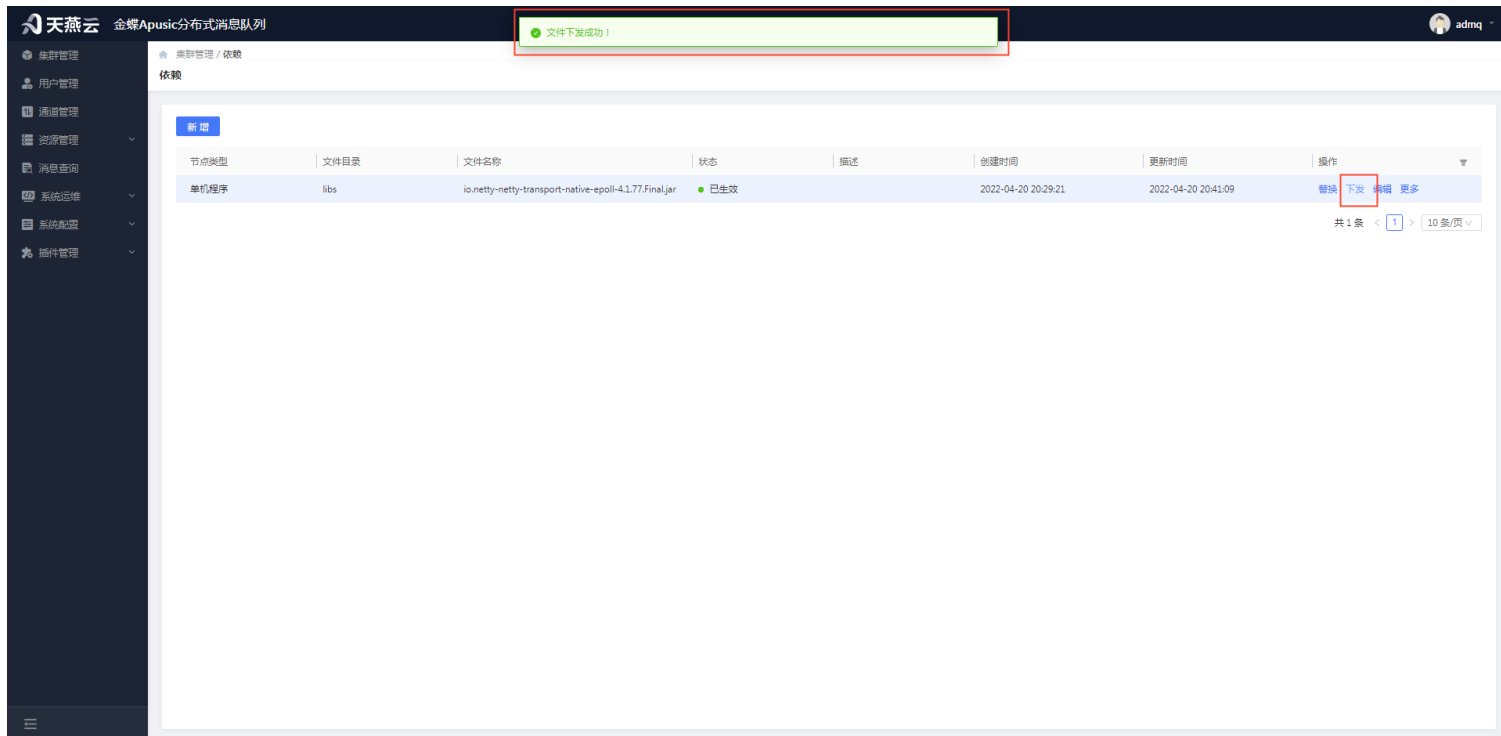




5.3.1.9.3 下发依赖

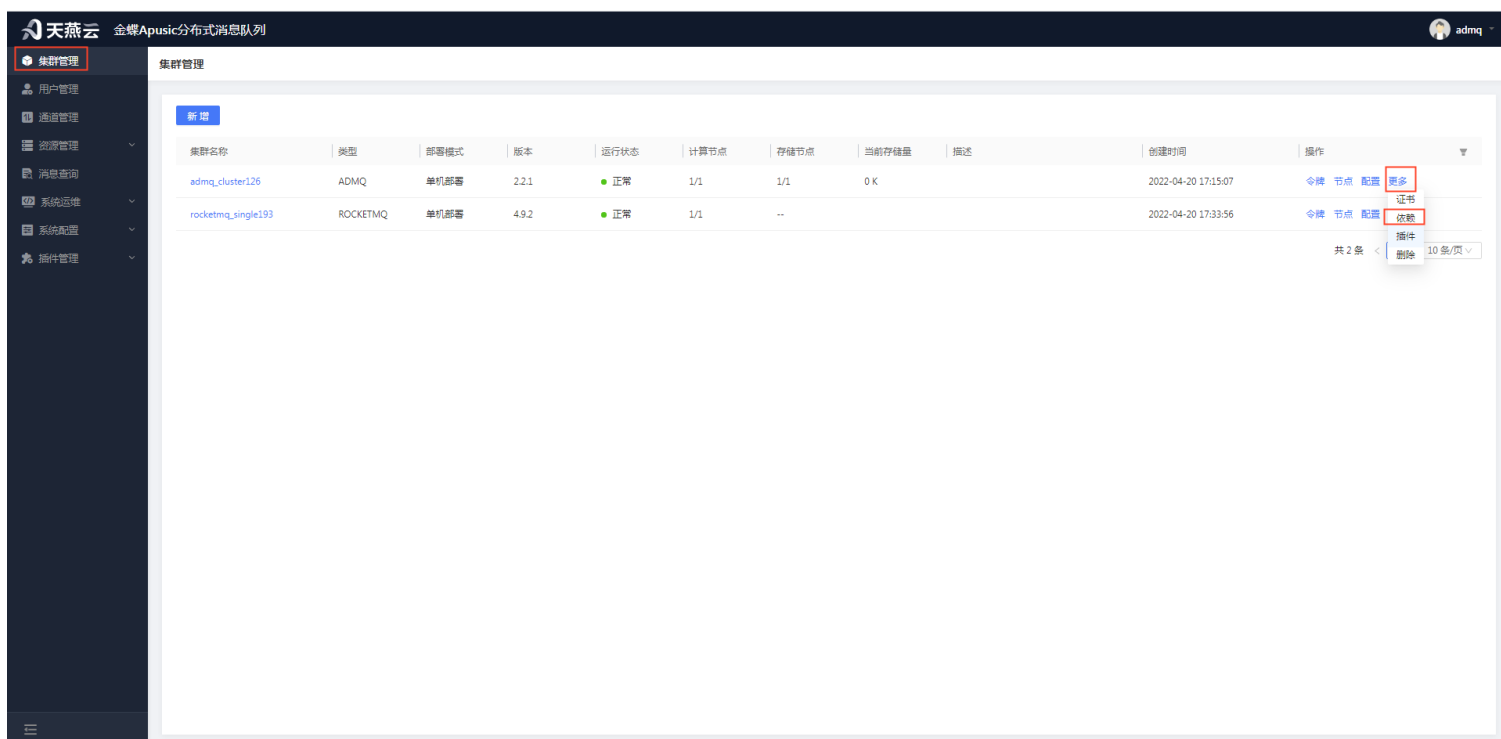
进入集群管理，选择需要管理依赖的集群，点击【更多】按钮->【依赖】按钮，在依赖页面，选择需下发的依赖，点击【下发】按钮，可下发依赖到已部署节点服务器上，下发成功后，状态会从未生效变成已生效。注意：成功部署集群后才能下发依赖，下发后应重启集群节点，才会生效。

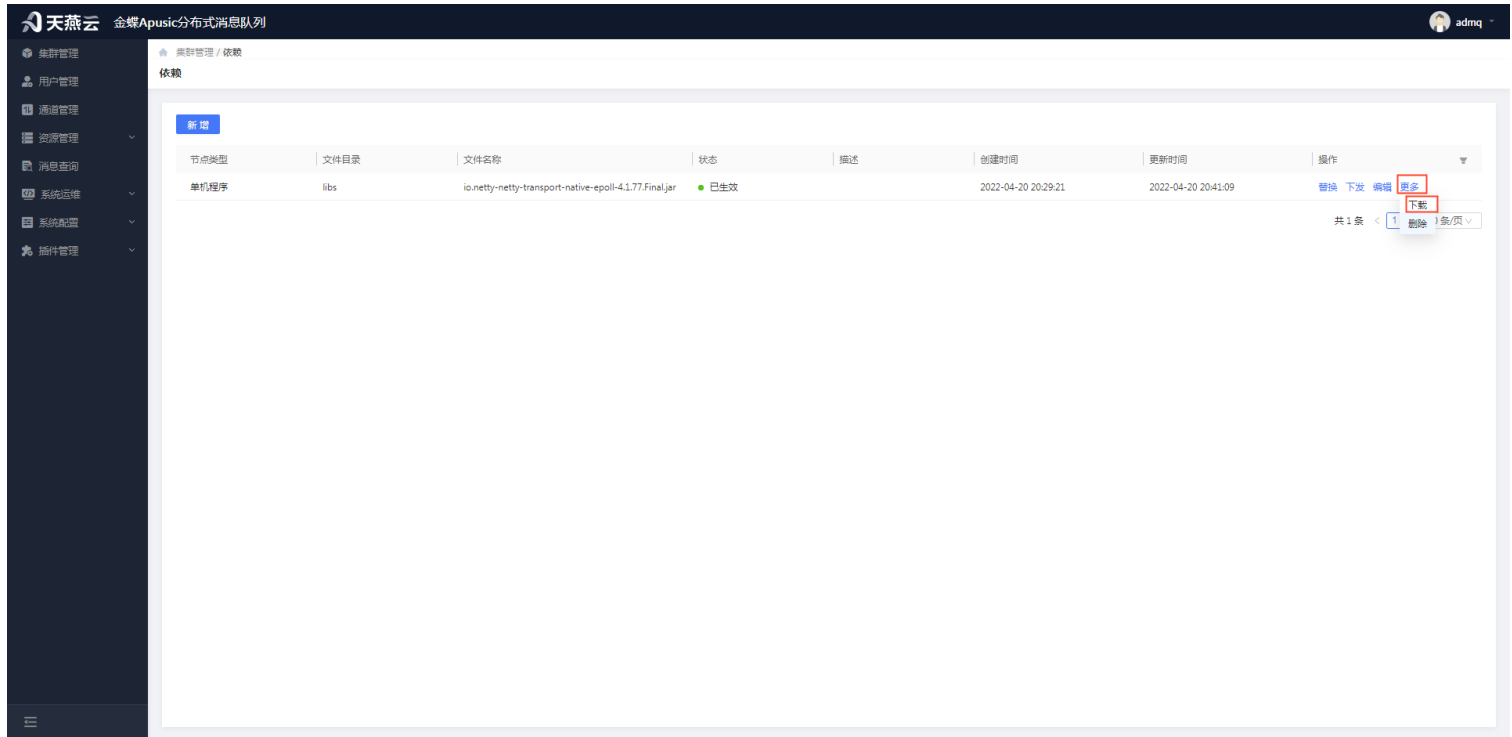




5.3.1.9.4 下载依赖

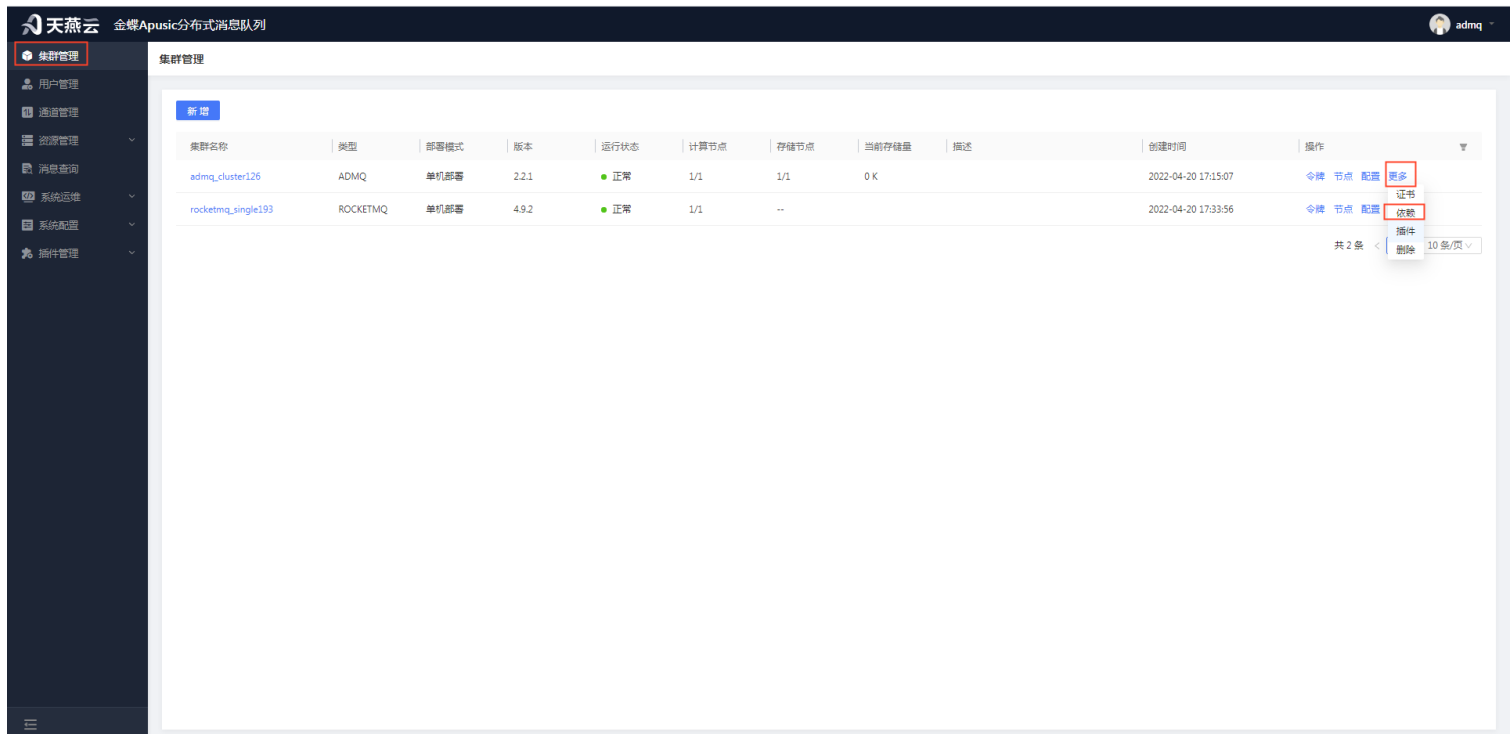
进入集群管理，选择需要管理依赖的集群，点击【更多】按钮->【依赖】按钮，在依赖页面，选择需下载的依赖，点击【更多】按钮->【下载】按钮，可下载依赖。注意：下载依赖是下载管理控制台上的依赖（非节点服务器上的依赖）

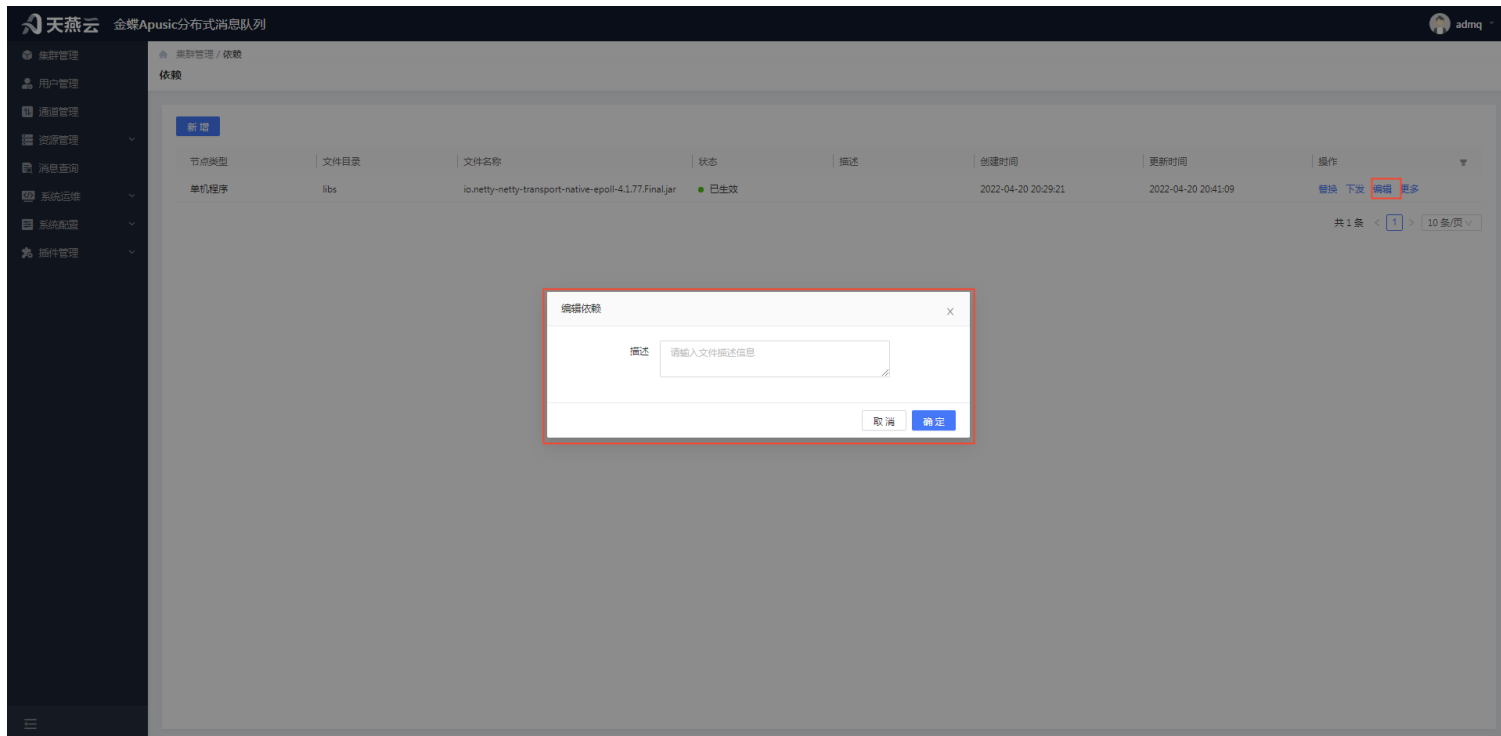




5.3.1.9.5 编辑依赖

进入集群管理，选择需要管理依赖的集群，点击【更多】按钮->【依赖】按钮，在依赖页面，选择需编辑的依赖，点击【编辑】按钮，可编辑依赖信息。

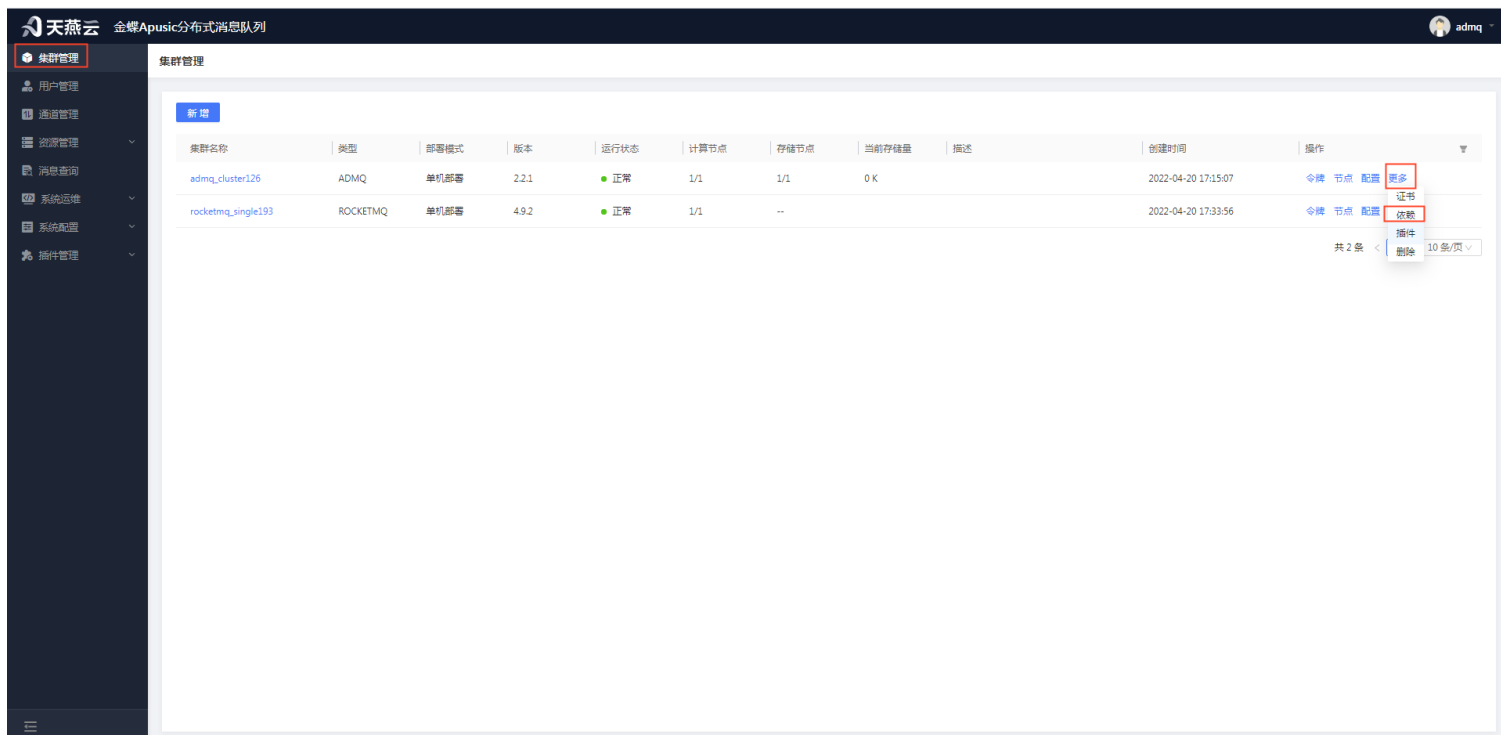


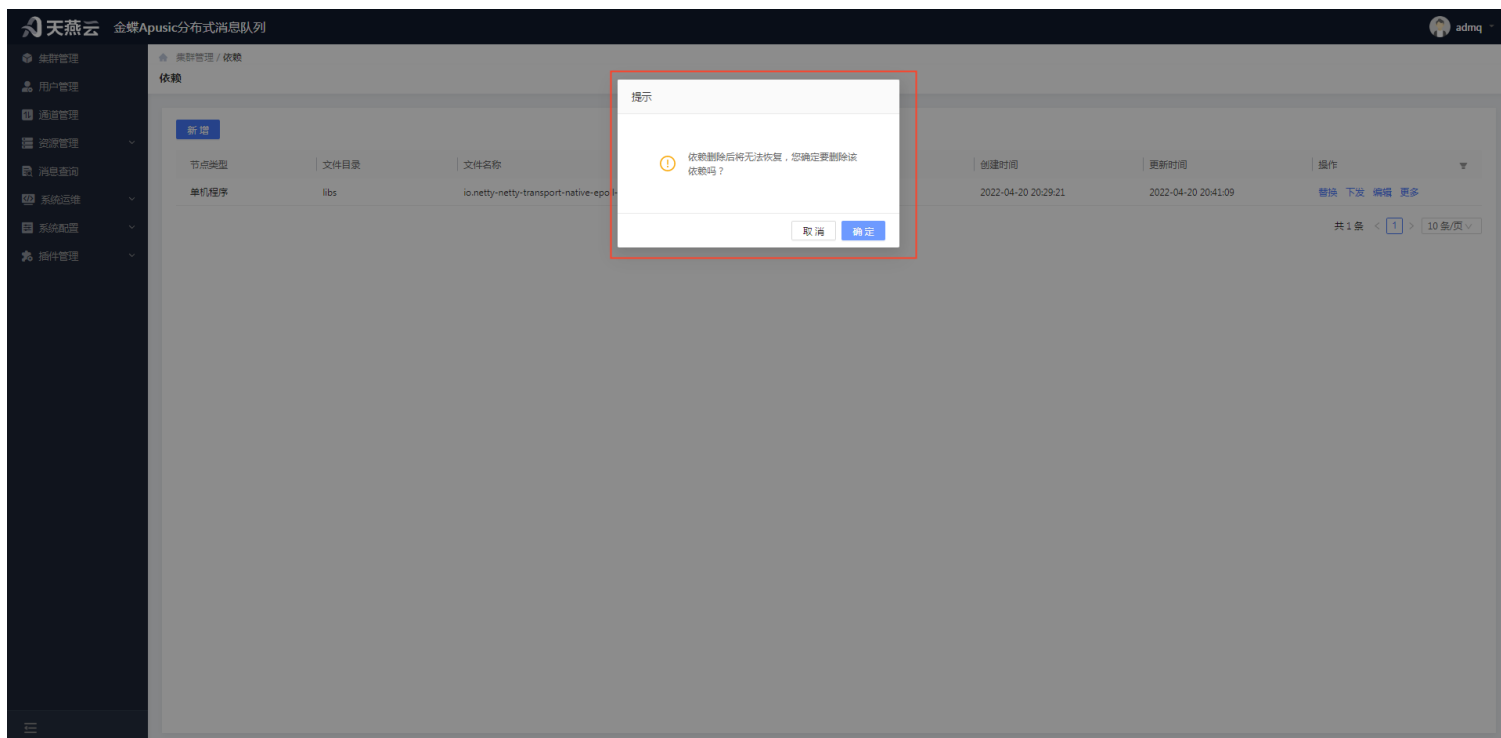
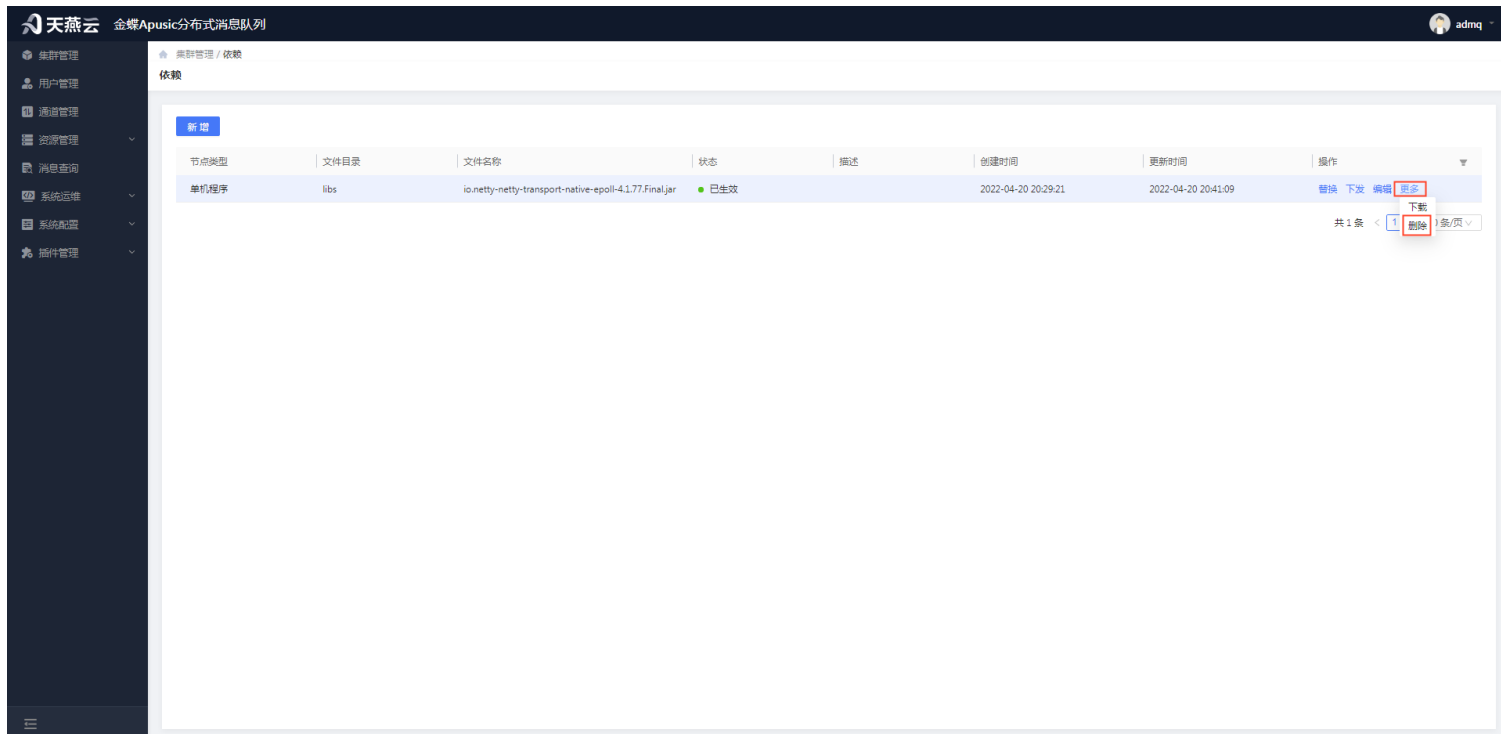


描述：依赖文件的描述信息

5.3.1.9.6 删除依赖

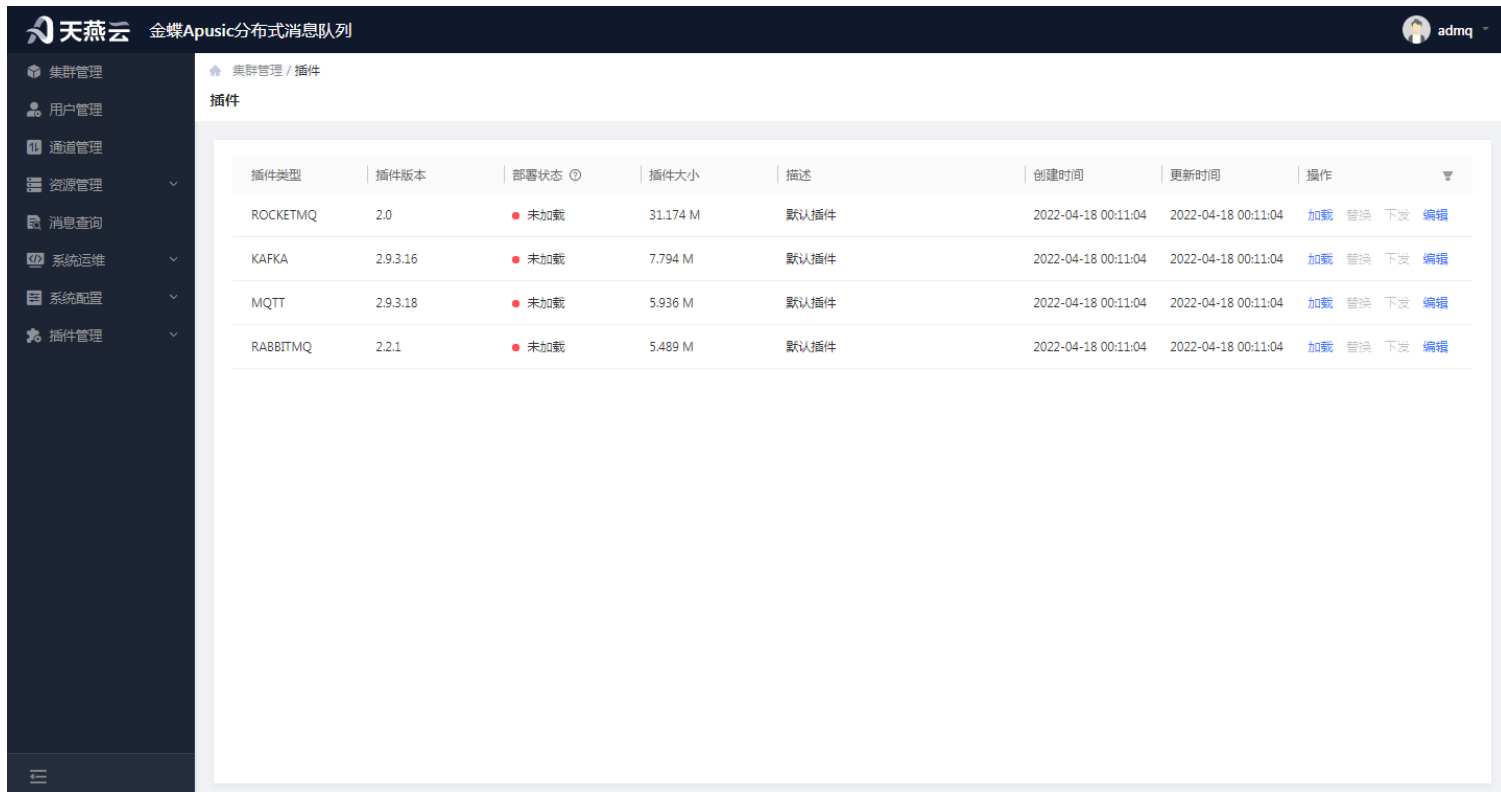
进入集群管理，选择需要管理依赖的集群，点击【更多】按钮->【依赖】按钮，在依赖页面，选择需编辑的依赖，点击【更多】按钮->【删除】按钮，可删除依赖信息。注意：删除依赖，只是删除管理控制台的依赖文件，不能删除节点服务器上的文件。





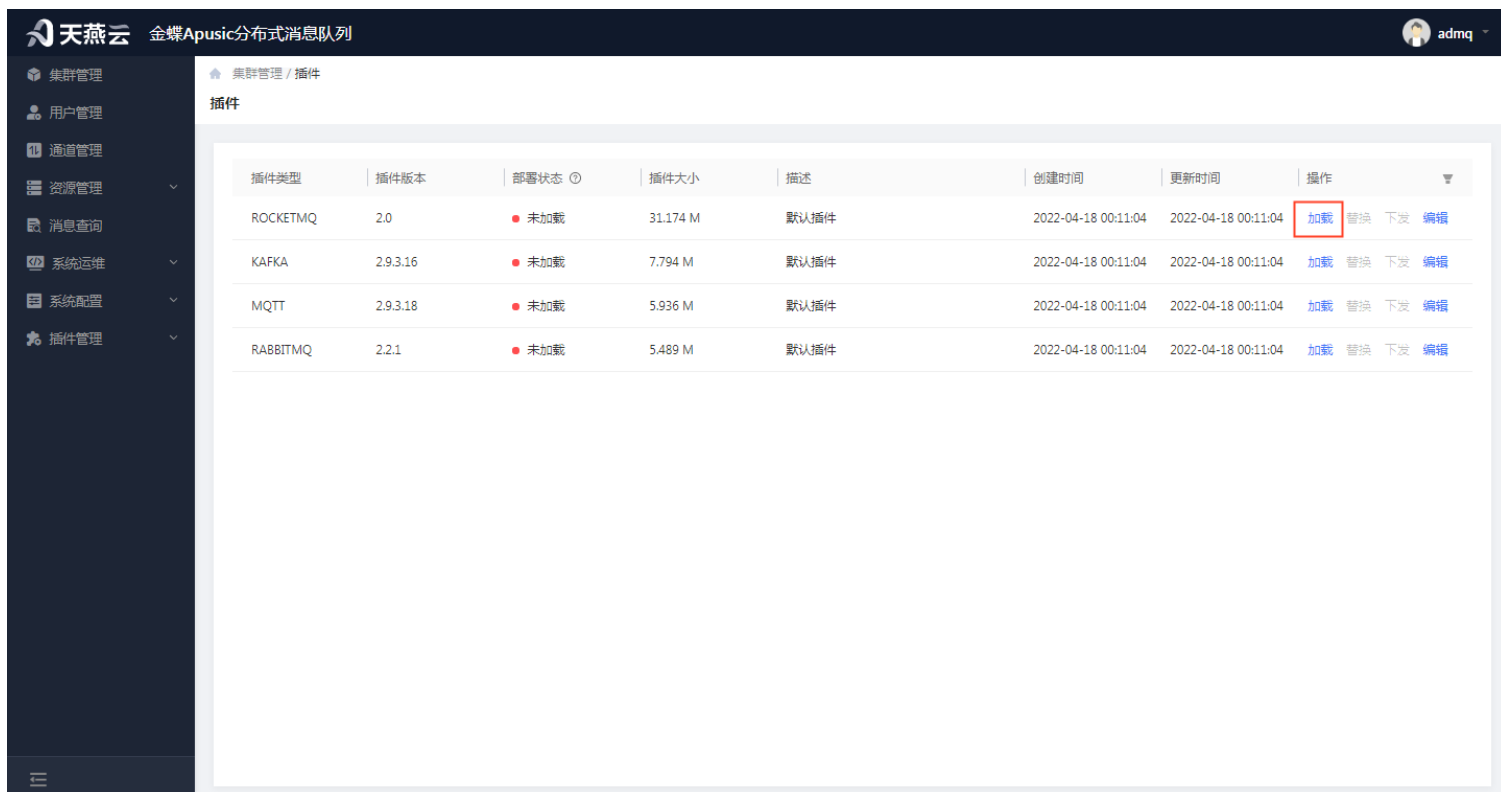
5.3.1.10 插件管理

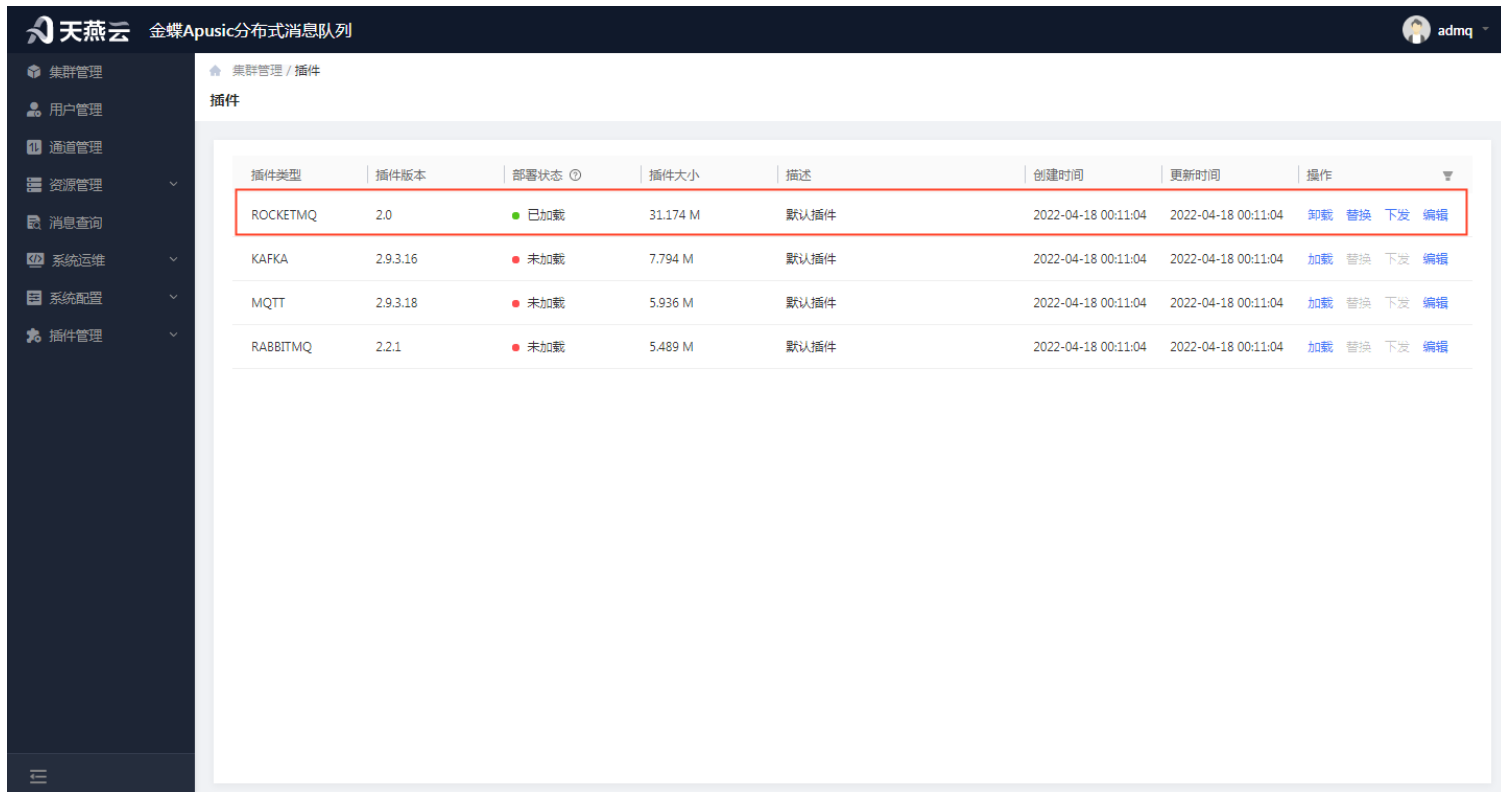
为方便对已部署的集群进行插件配置，管理控制台支持对ADMQ内置的插件进行管理，应用场景主要用于启动ROCKETMQ服务、KAFKA服务、MQTT服务、RABBITMQ服务。插件管理包括：加载插件、替换插件、下发插件、编辑插件。



5.3.1.10.1 加载插件

进入集群管理，选择需要加载插件服务的集群，点击【更多】按钮->【插件】按钮，在插件页面，选择需加载的插件，点击【加载】按钮后即可加载插件。注意：加载插件后应重启集群节点后才会生效。

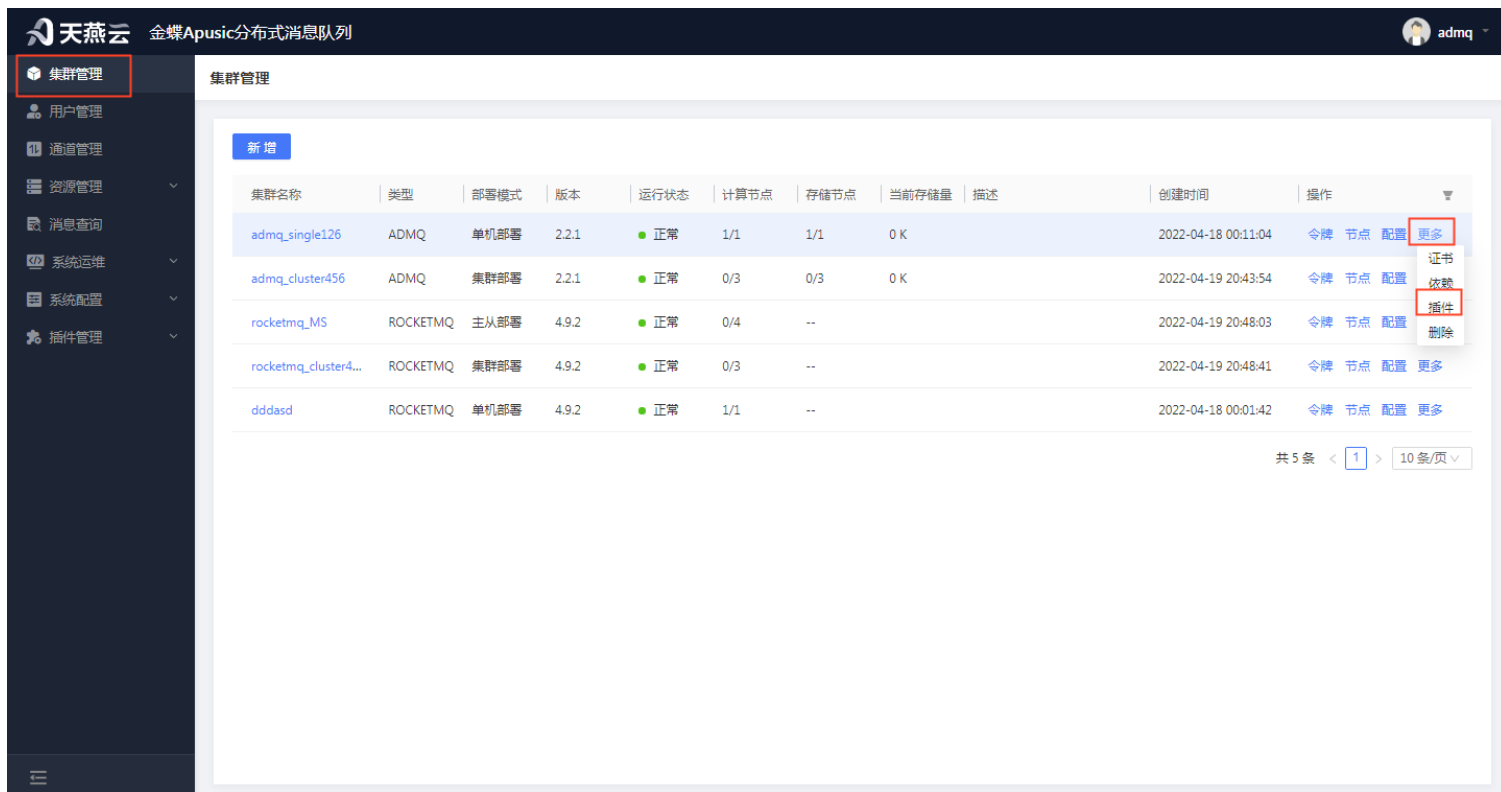


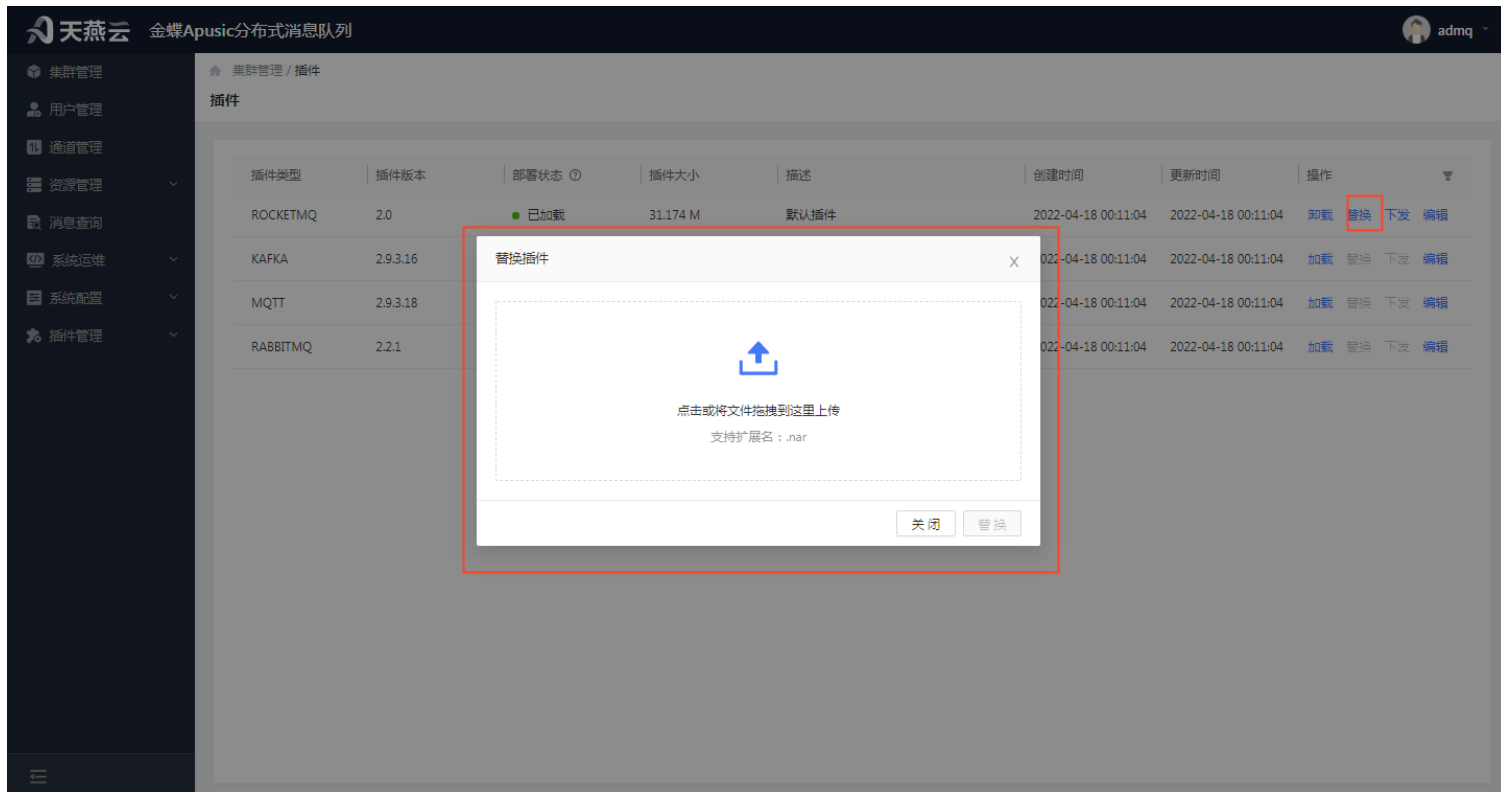


备注：插件只有被加载后才可进行替换和下发，四种插件可同时被加载。

5.3.1.10.2 替换插件

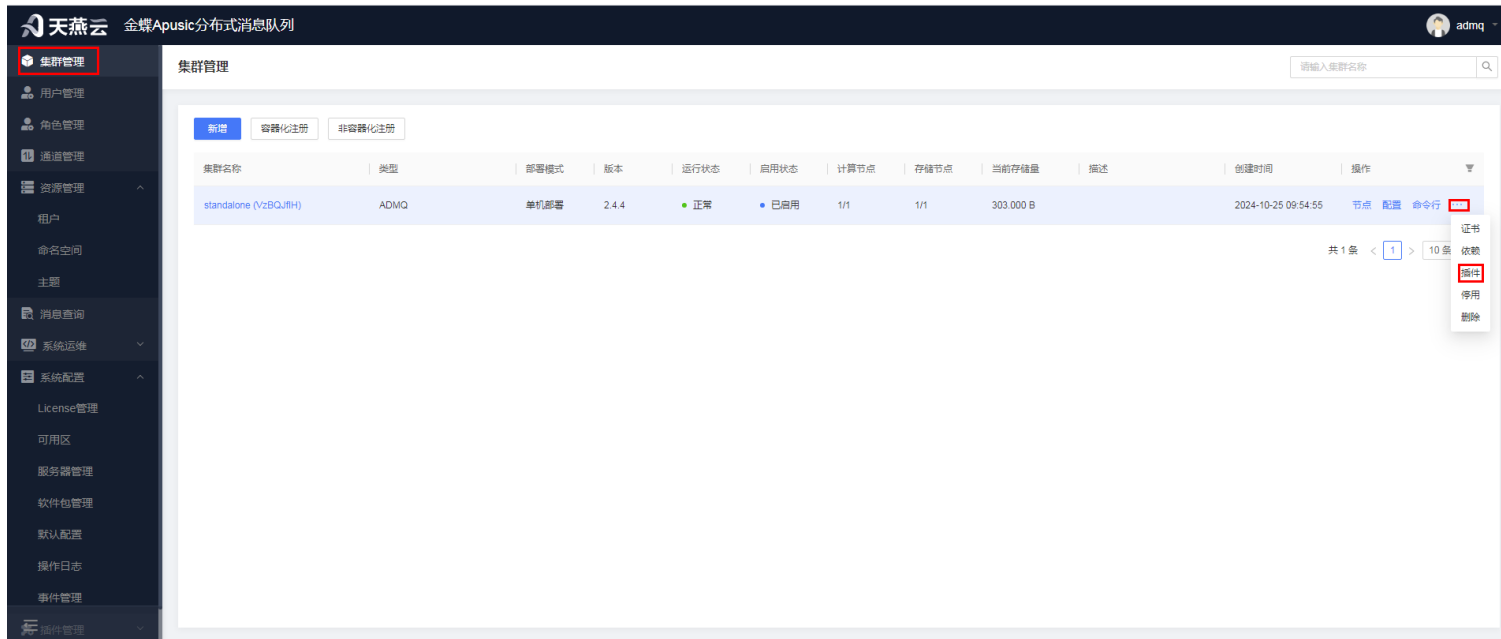
进入集群管理，选择需要替换插件的集群，点击【更多】按钮->【插件】按钮，在插件页面，选择需替换的插件，点击【更多】按钮->【替换】按钮后上传或拖拽插件，可以替换插件。注意：替换的插件需要下发才能更新到集群各个节点下，且下发后应重启集群节点，才会生效。

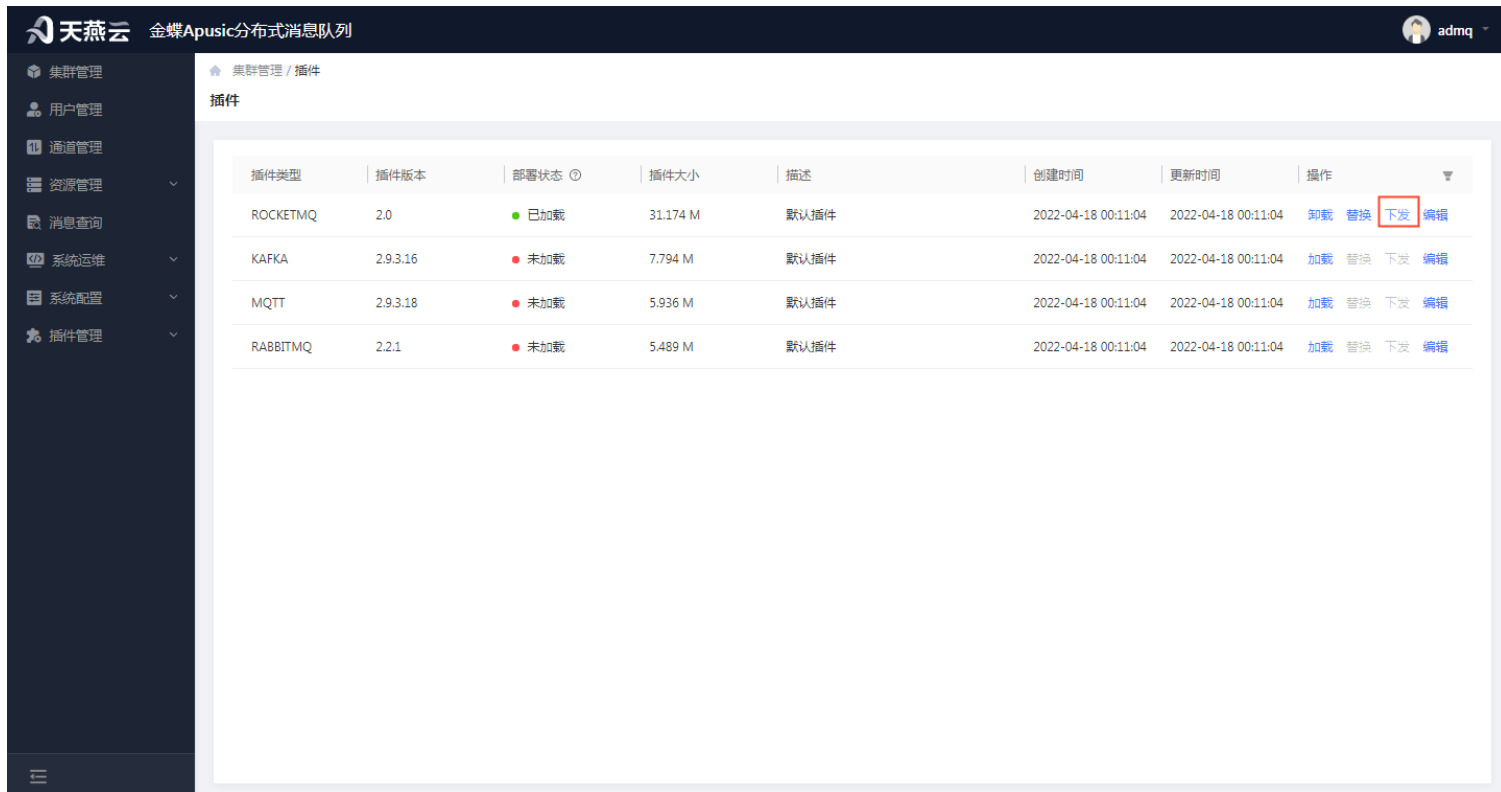




5.3.1.10.3 下发插件

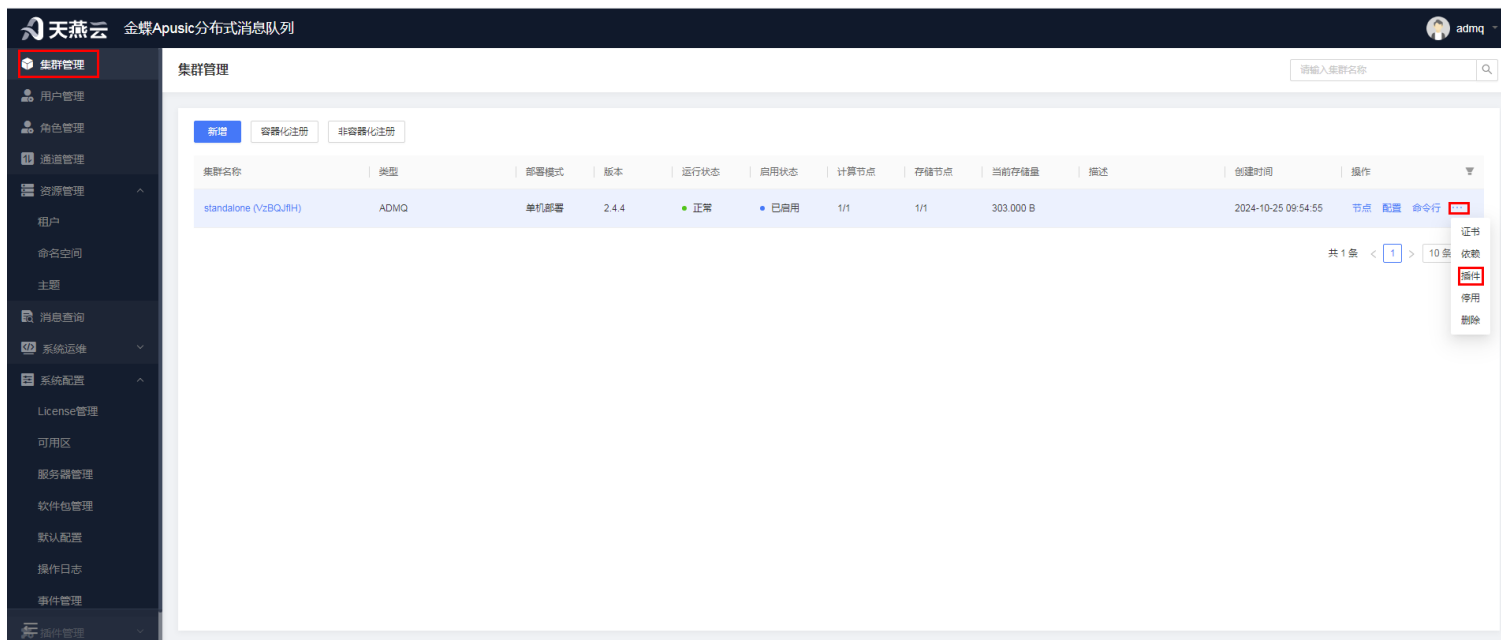
进入集群管理，选择需要下发插件的集群，点击【更多】按钮->【插件】按钮，在插件页面，选择需下发的插件，点击【下发】按钮，可下发插件到已部署节点服务器上。注意：成功部署集群后才能下发插件，下发后应重启集群节点，才会生效。

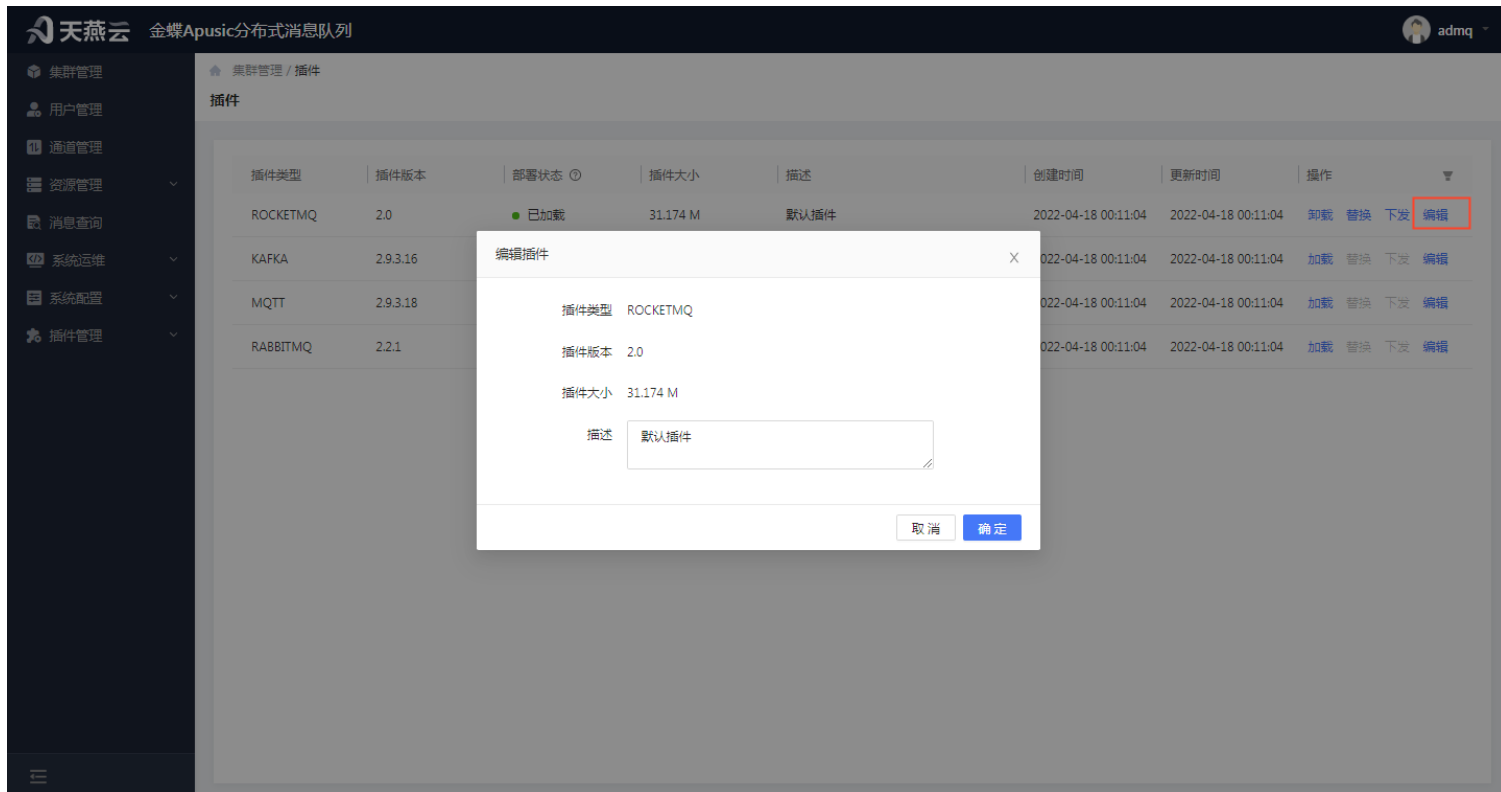




5.3.1.10.4 编辑插件

进入集群管理，选择需要编辑插件的集群，点击【更多】按钮->【插件】按钮，在插件页面，选择需编辑的插件，点击【编辑】按钮，可编辑插件信息。





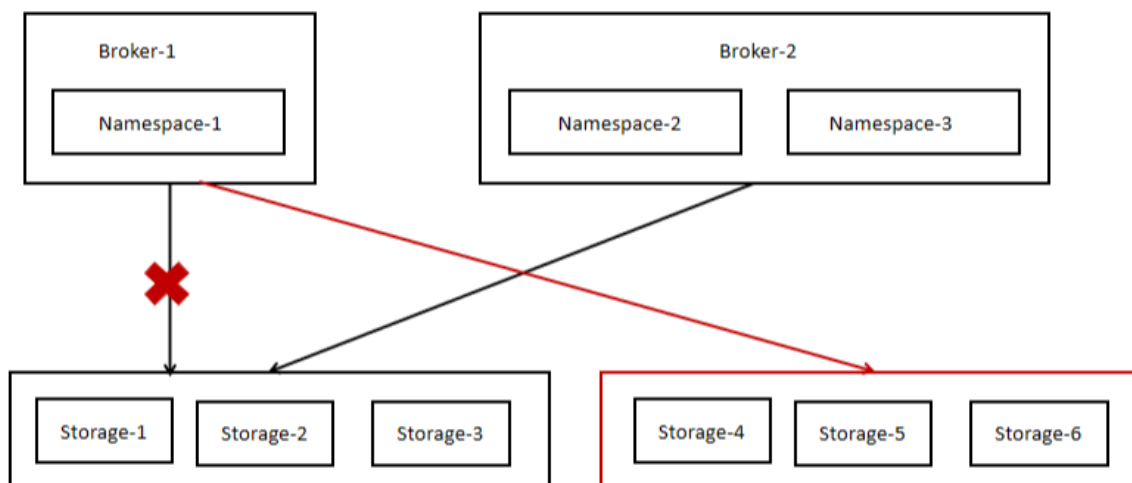
描述: 插件描述信息

5.3.1.11 资源隔离

客户业务上经常会出现这种情况，就是集群虽然扩容了，但总有那么几个用户的流量比较大，头部效应很明显，其他用户的Topic只要和这几个用户在一个Broker上，必然被他们忽高忽低的流量所影响，对于这种大流量客户，最好的方式就是让他们迁移到独立的集群上，避免影响其他小用户，然而，这些头部用户刚开始的业务量可能并不大，因此使用了共享集群，当这些用户后续的业务量快速上升时，再把他们迁移到新的集群上，就可能涉及到业务层面的改动，影响客户业务的正常运行。

既要做到业务层面无感知，又要快速隔离大流量用户，这是现在很多消息队列都无法做到的，或者需要很复杂的方案才能做到，而ADMQ的架构天然支持特定Namespace的隔离，限制用户的某些Namespace迁移到指定的Broker上。

资源隔离分Broker层的资源隔离和Storage层的资源隔离。



Broker层的资源隔离：上图中的Namespace-1被单独隔离到Broker-1中，其它Namespace继续在Broker-2中共享机器资源，如果线上某些用户出现了突发流量，name可以先新增一个Broker节点专门用于突发流量的用户隔离。利用ADMQ的Namespace隔离能力，可以让指定的Namespace下所有Bundle只落在指定的

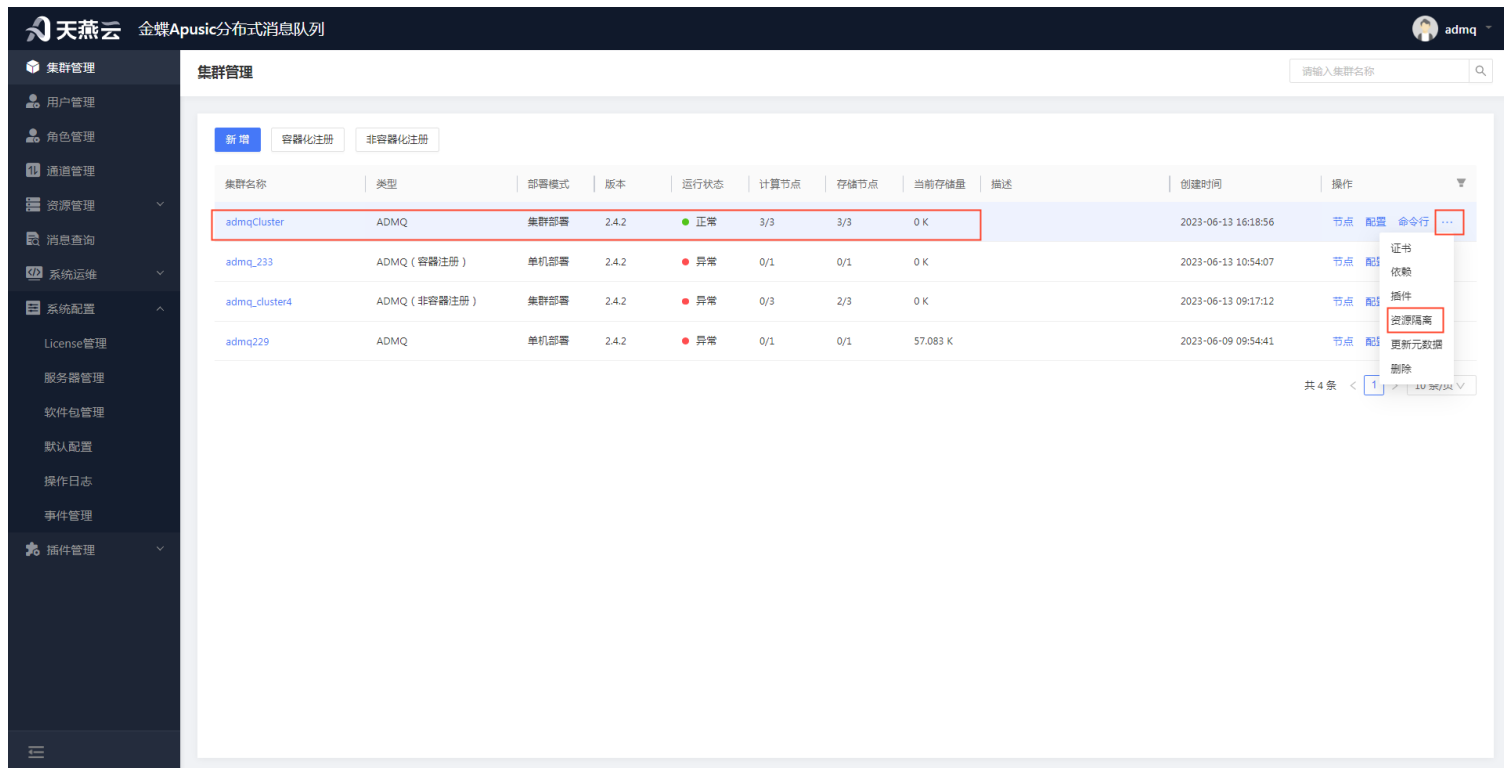
Broker节点上，避免影响其他用户。

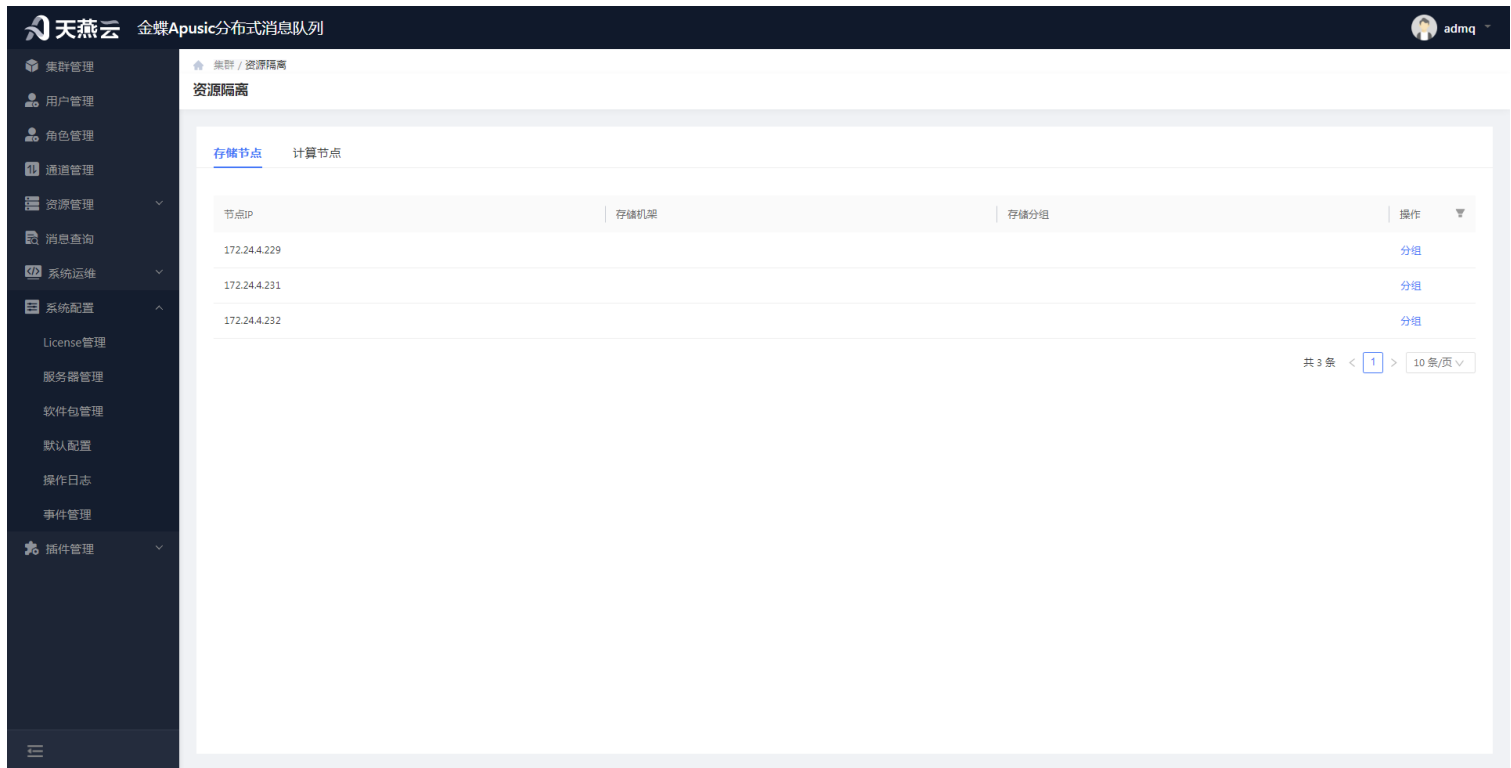
Storage层的资源隔离：上图中Broker-1、Broker-2共享了Storage-1、Storage-2、Storage-3（图中两个黑色箭头的指示方向），如果此时来自Broker-1中的Namespace-1的流量比较大，那么Storage-1、Storage-2、Storage-3节点的I/O会被占用，导致Broker-2上其他Namespace的读写延迟增加。对于Broker-1这种大流量节点，我们也可以实现快速隔离，把Broker-1中的Namespace-1的流量切换到新增节点Storage-4、Storage-5、Storage-6上（图中红色方框里的存储节点以及红色箭头的指示方向），Broker-2继续使用Storage-1、Storage-2、Storage-3。

管控台设置资源隔离的具体操作方式：

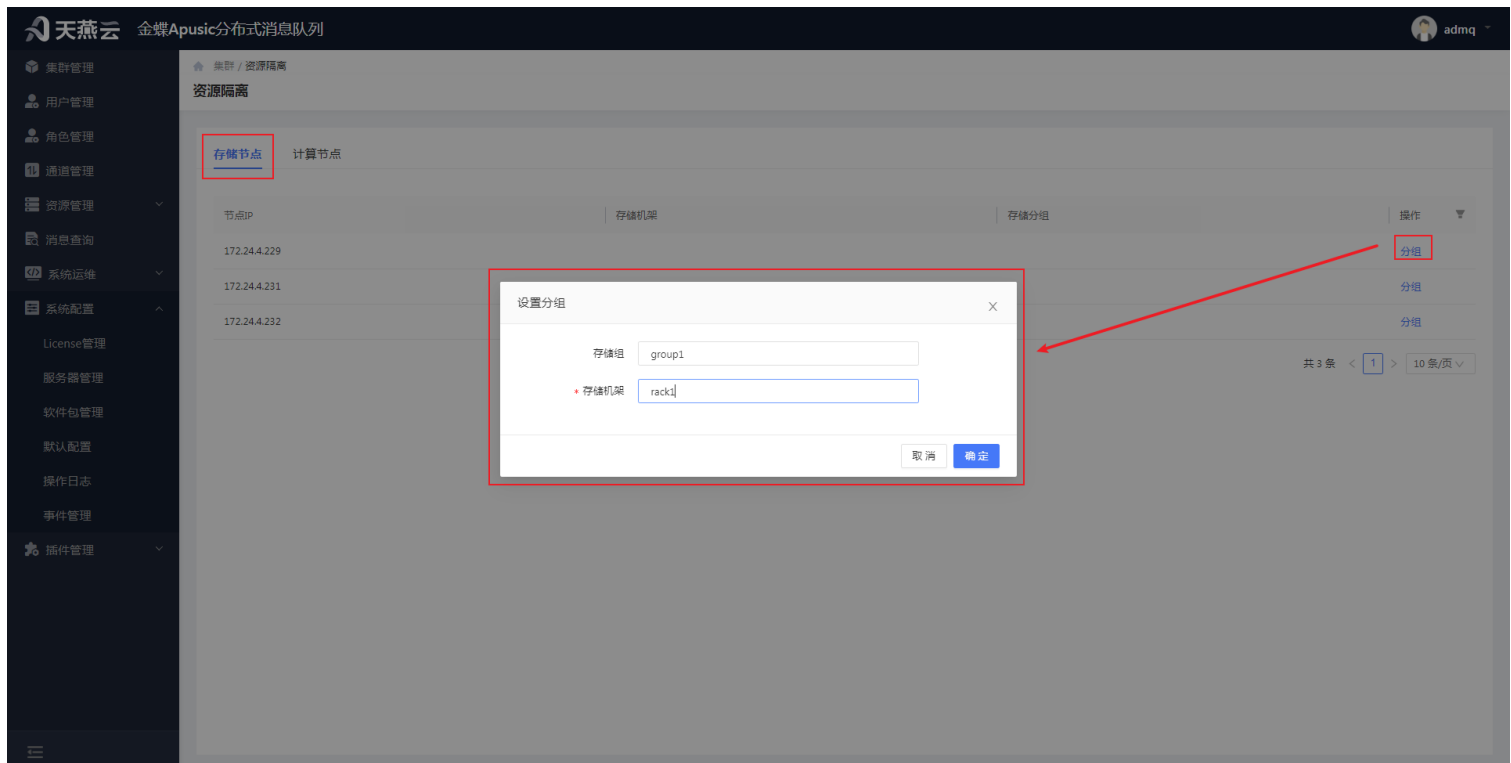
(1) 存储、计算 资源隔离设置

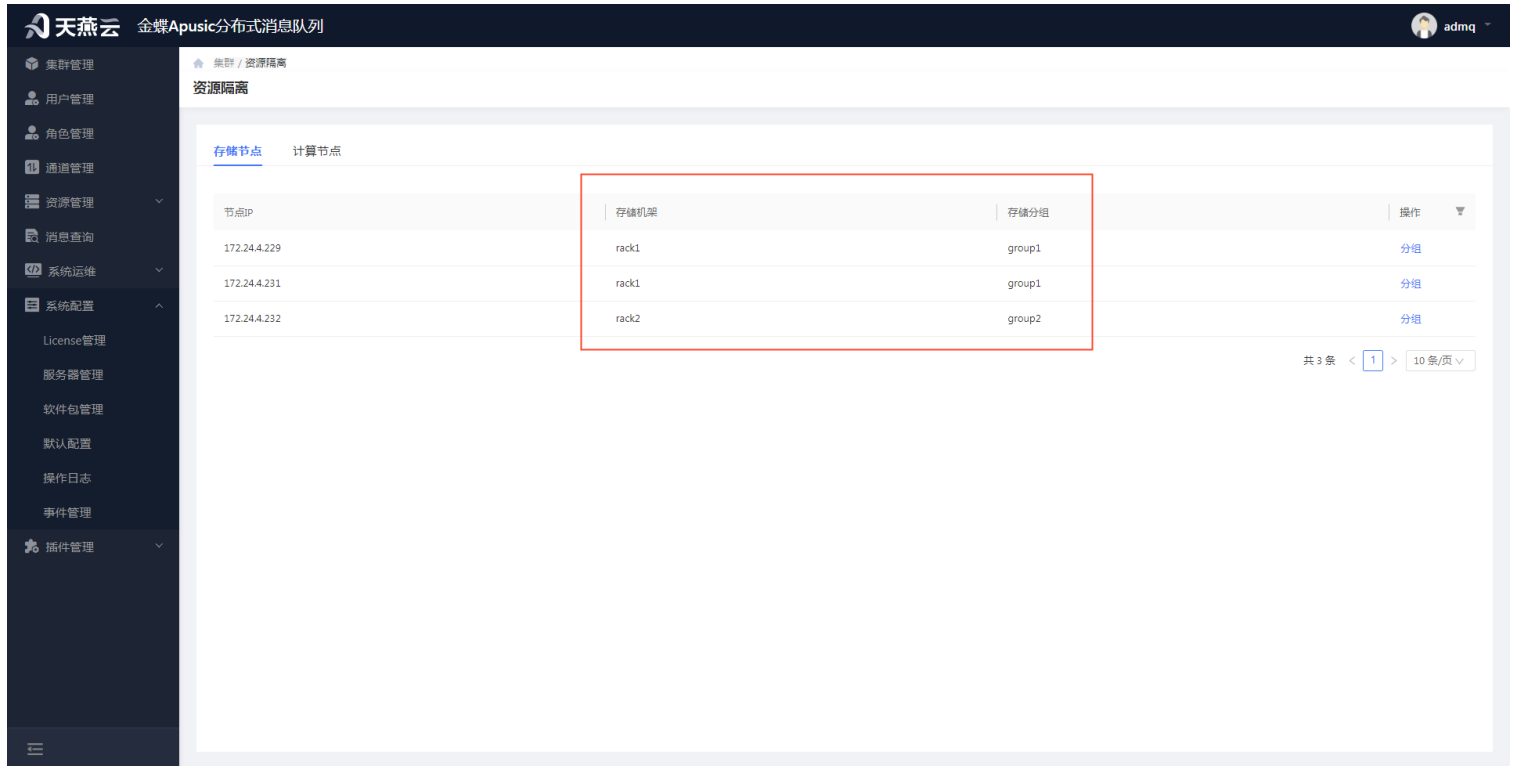
进入集群管理，在集群列表页面选择需要设置资源隔离的ADMQ集群，【更多】—》【资源隔离】，进入到具体设置页面：



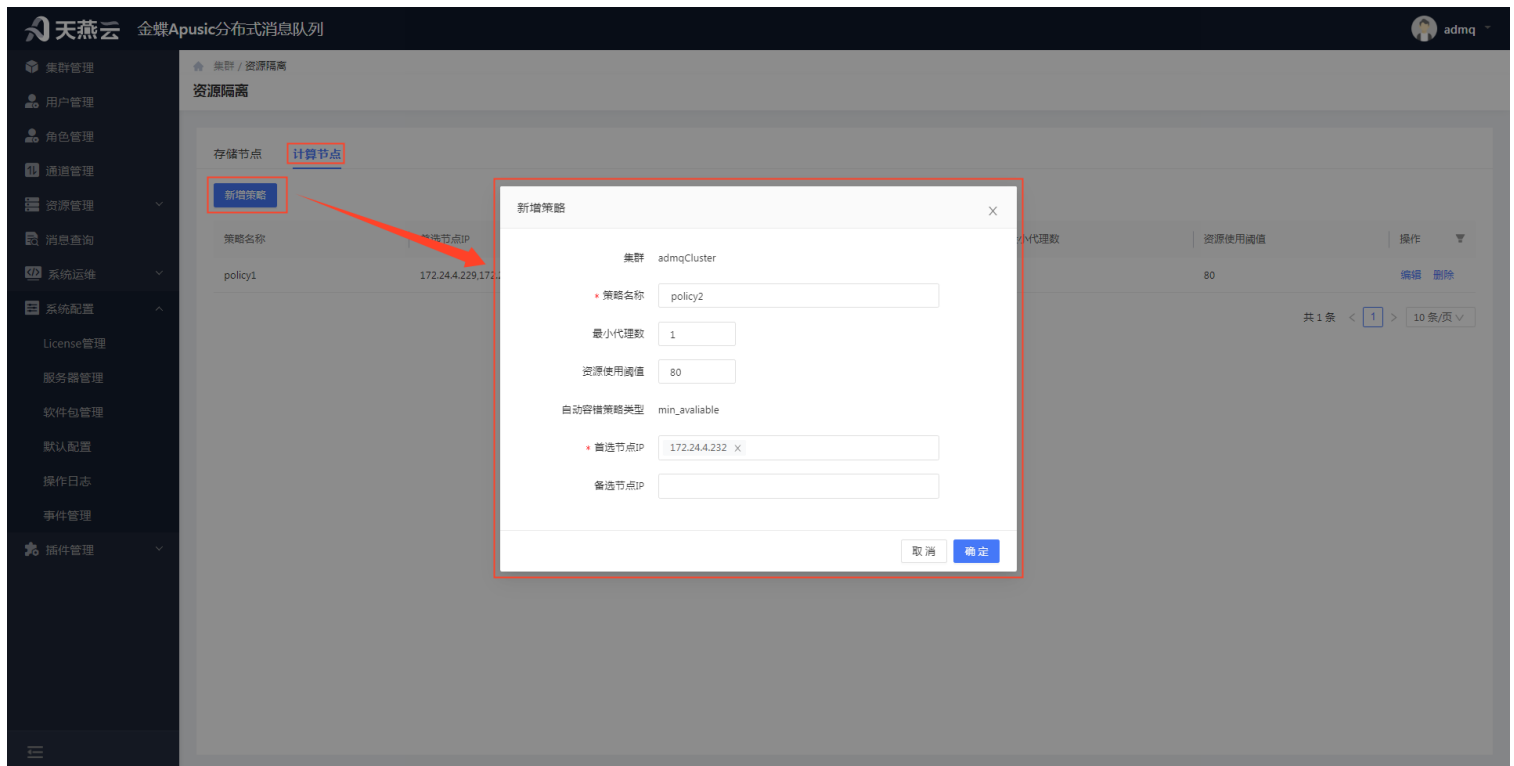


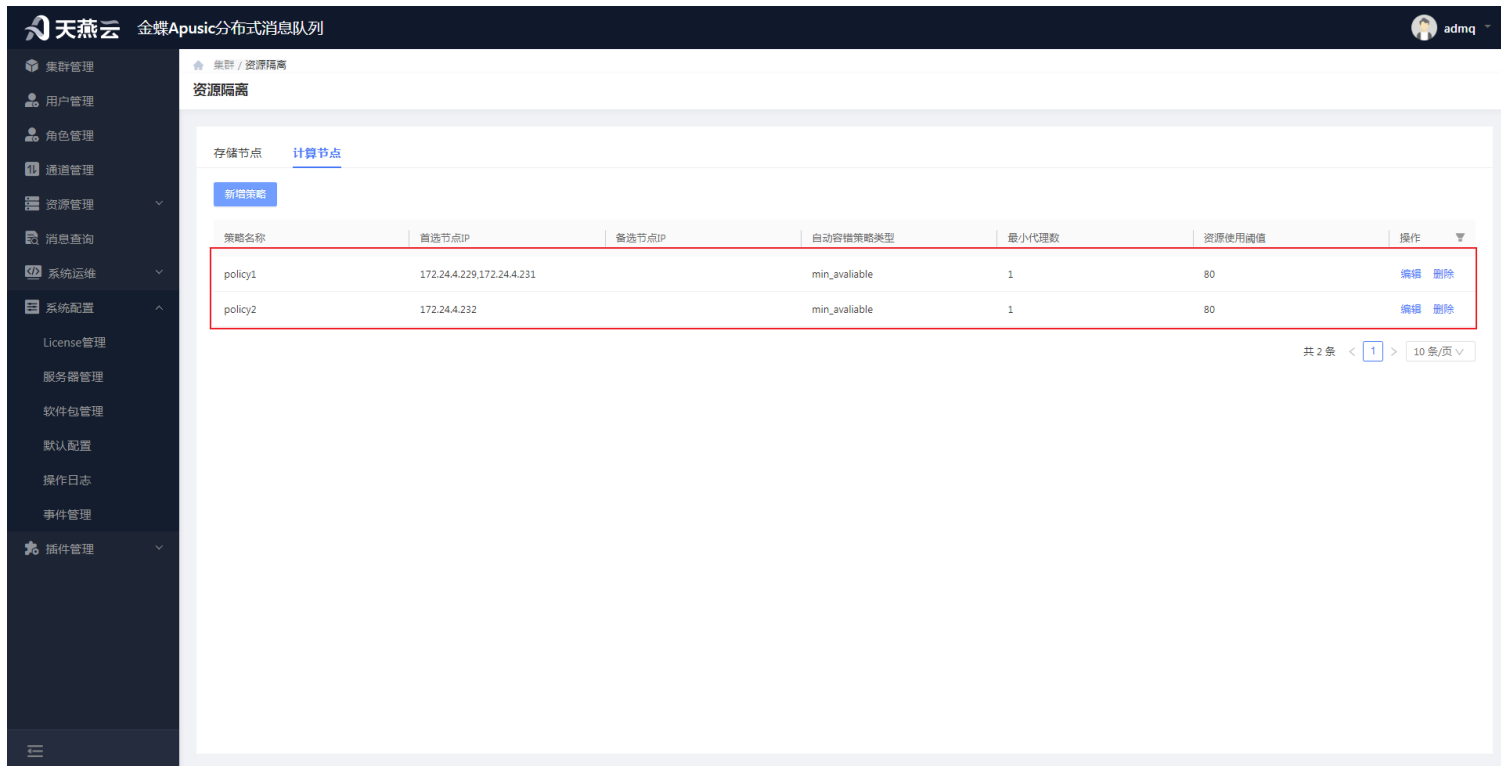
在“存储节点”页签的节点列表，点击【分组】，会出现 设置分组 弹窗，为每个存储节点设置存储机架和存储分组，设置好后会在节点列表的存储机架一栏和存储分组一栏展示出来：





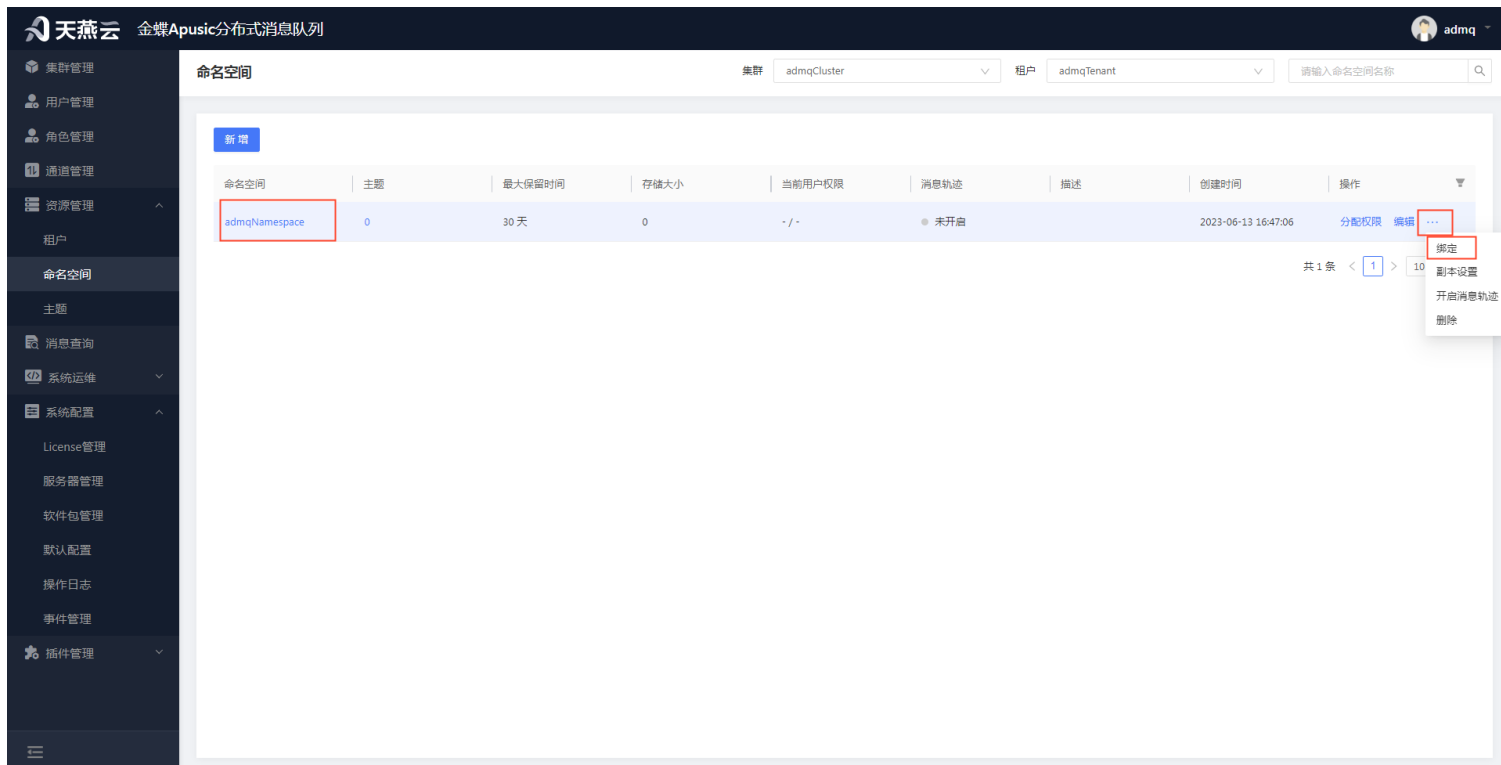
在“计算节点”页签，点击左上角【新增策略】，出现新增策略弹窗，输入相关信息，点击【确定】，新增的策略会展示在“计算节点”页签的列表页面：

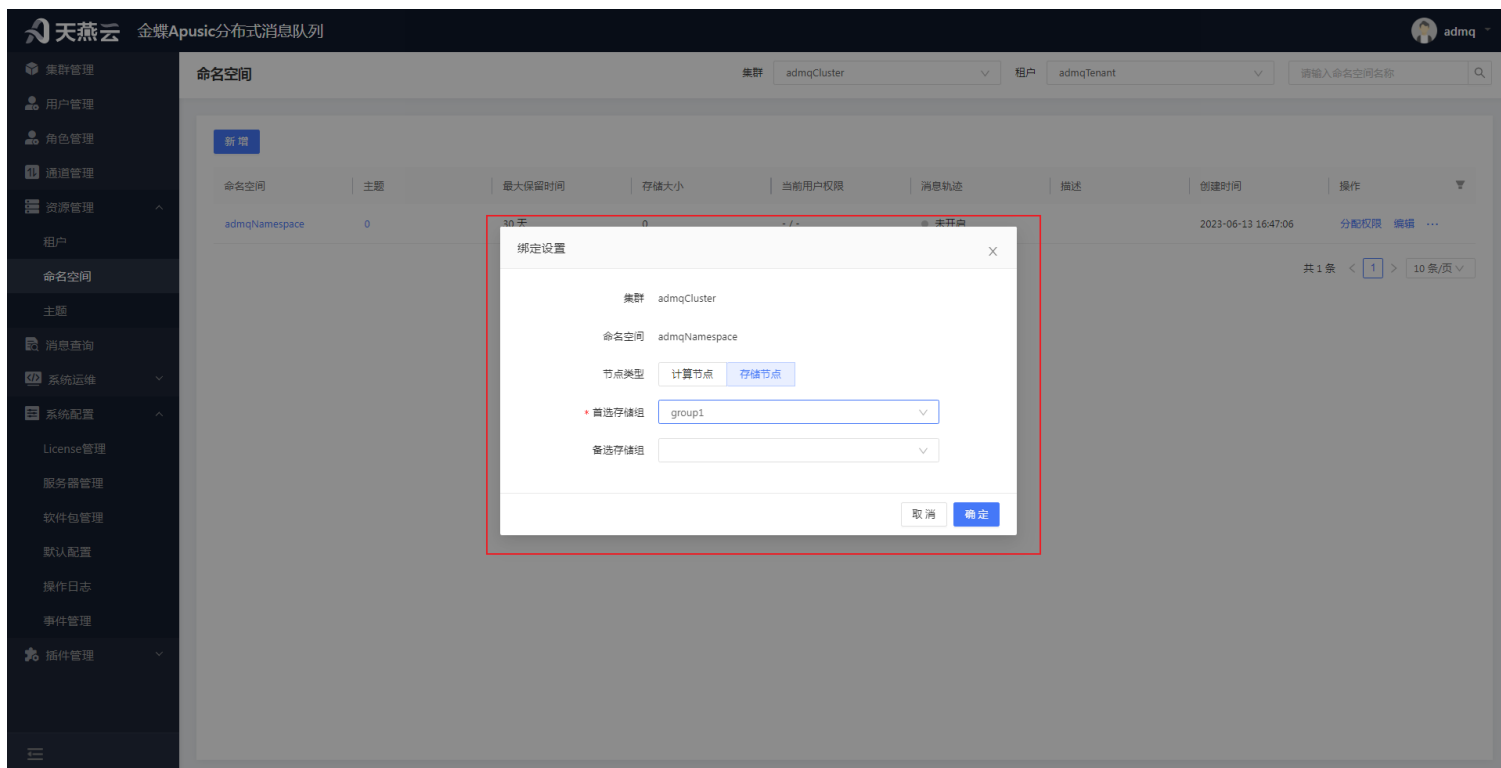
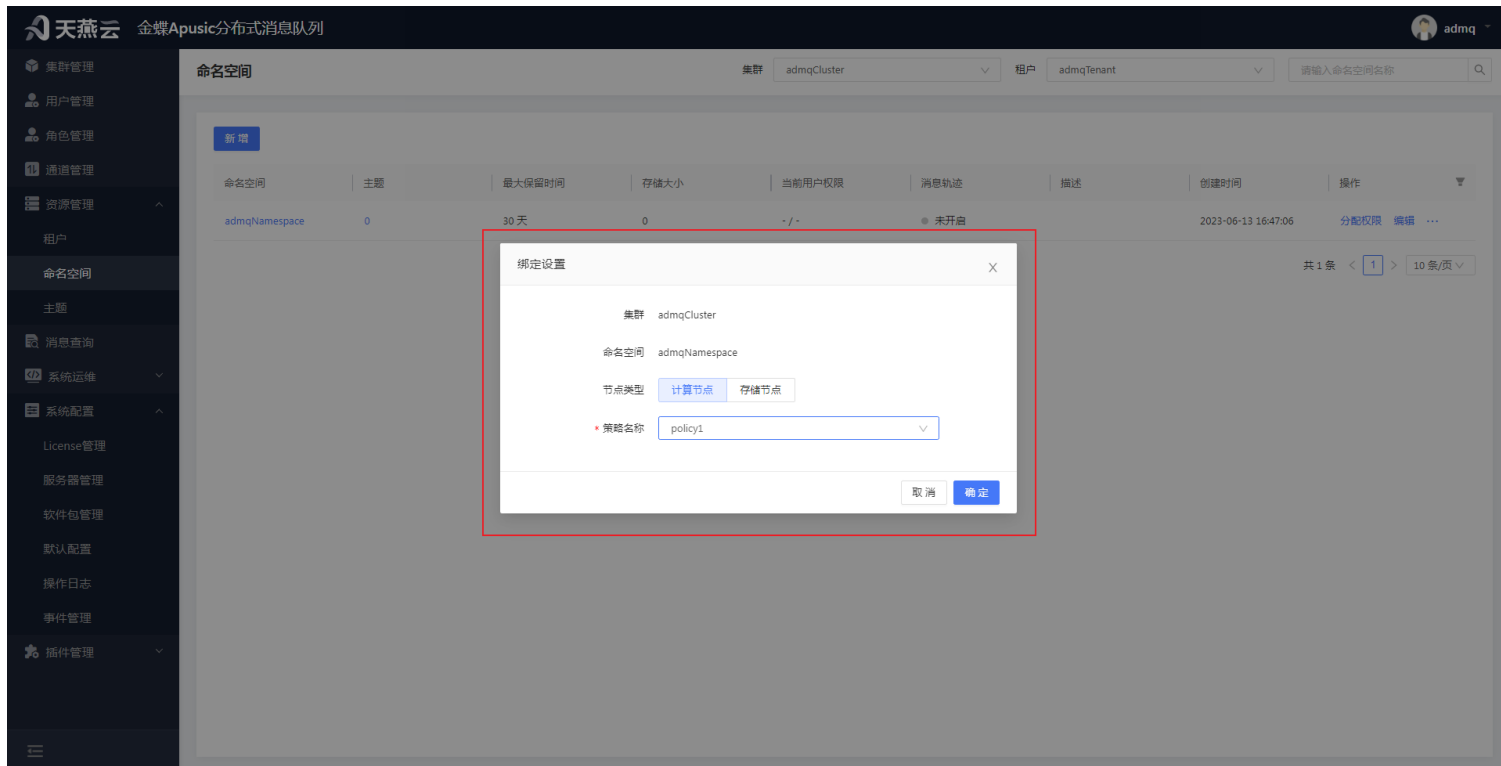




(2) 命名空间绑定

进入【资源管理】—>【命名空间】页面，选择需要绑定的命名空间，点击【...】—>【绑定】按钮，出现 绑定 弹窗，输入相关信息，点击【确定】按钮，进行绑定。备注：计算节点和存储节点需要分开绑定，例如：为计算节点选择策略后，需要先点击确定，然后再为存储节点选择存储组。



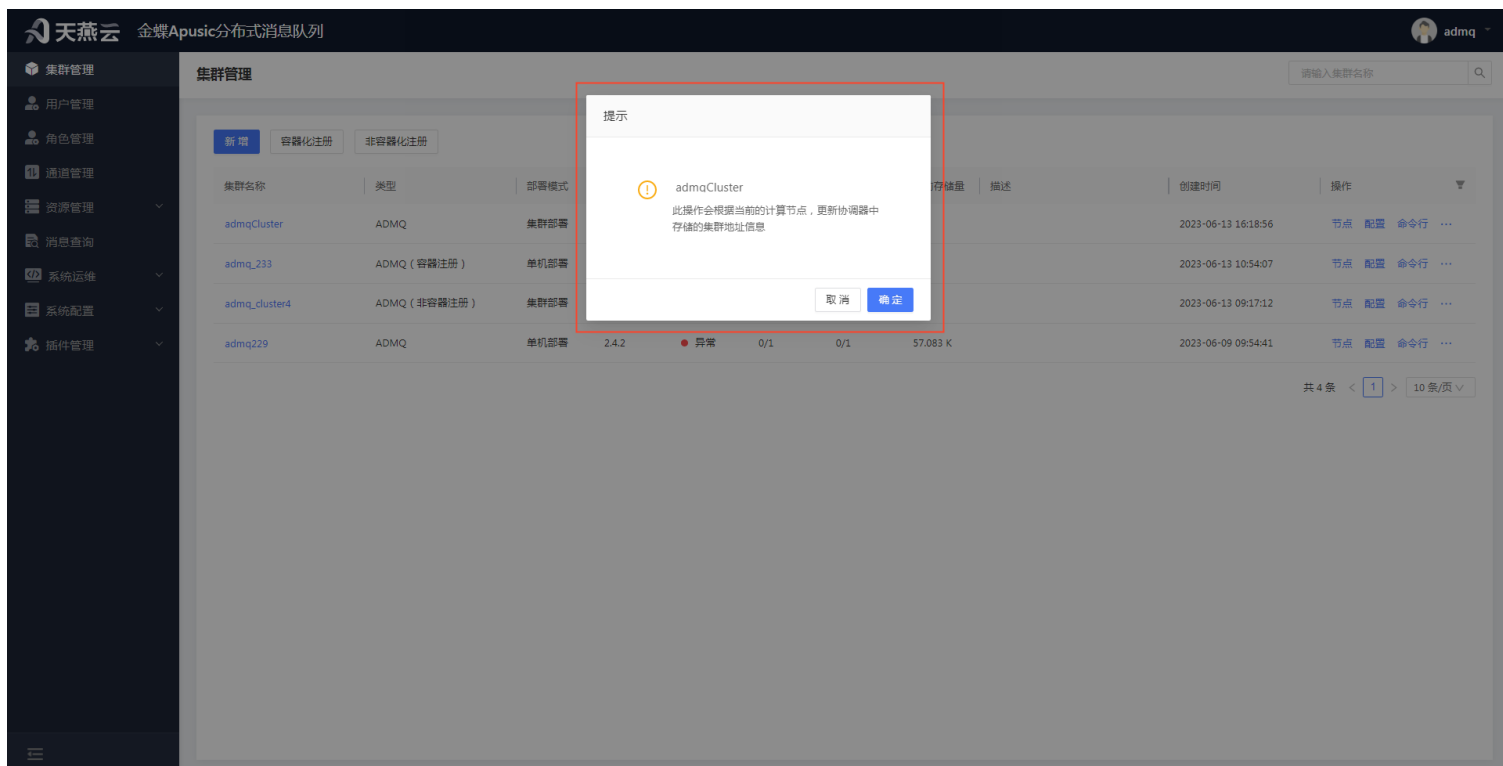
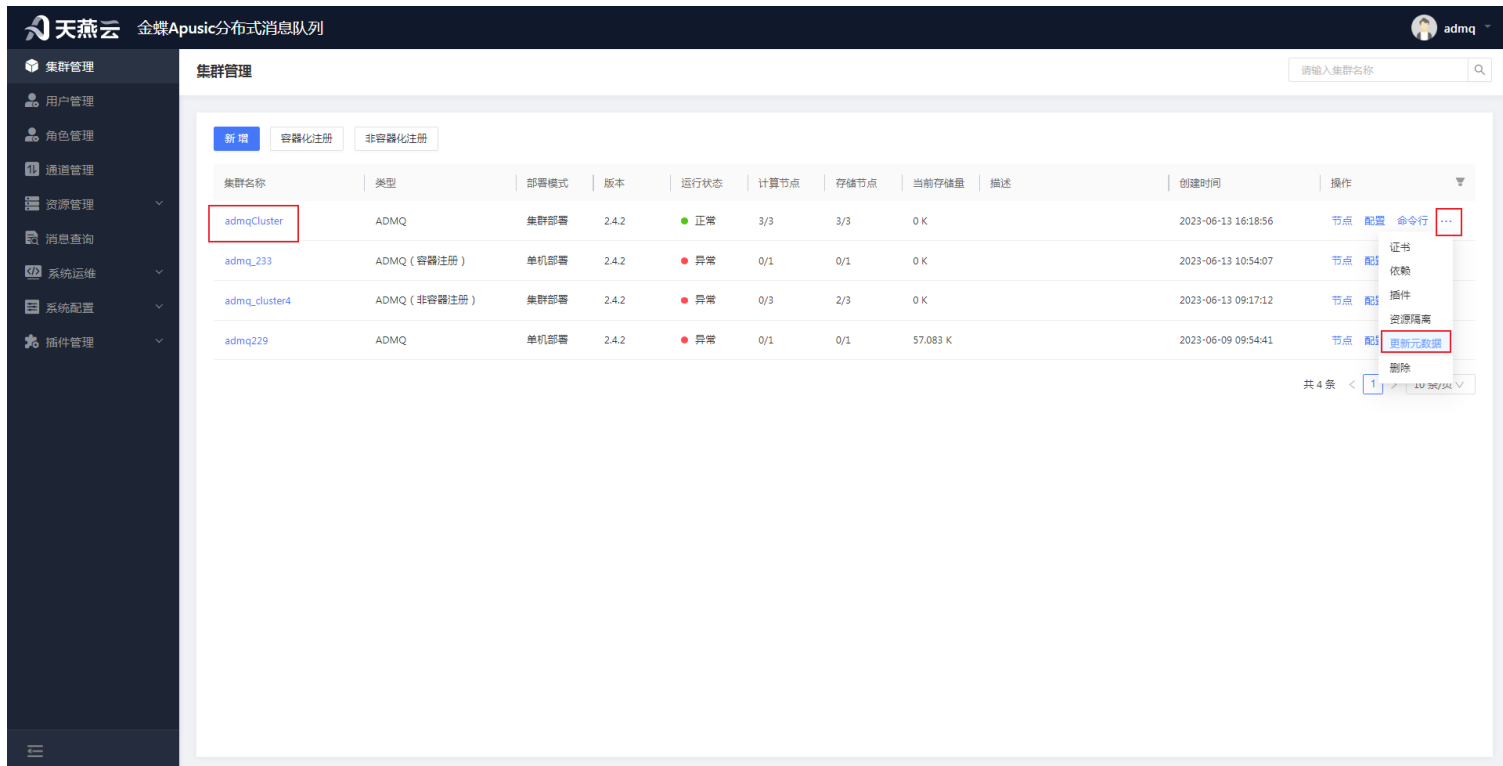


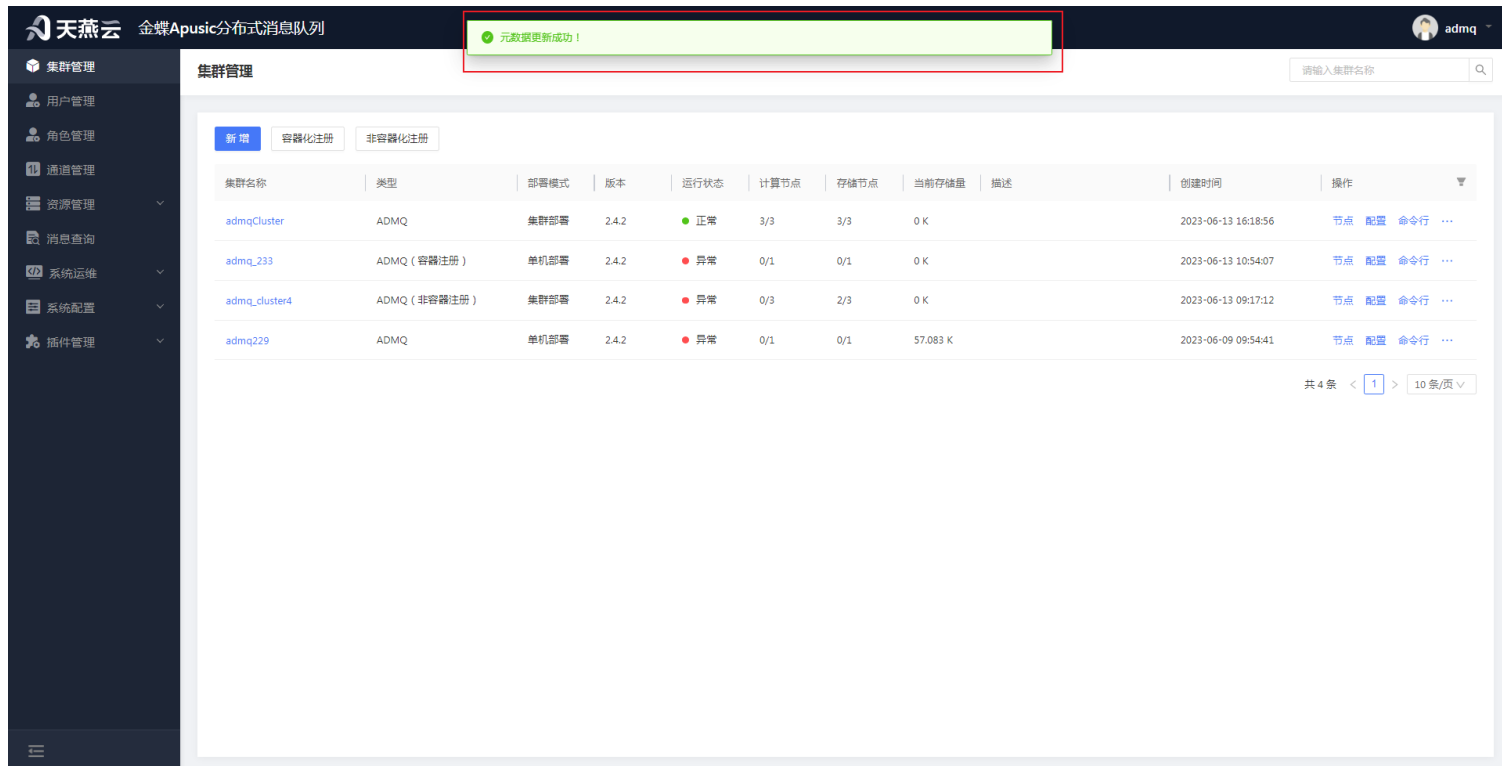
5.3.1.12 元数据初始化

当集群信息被修改后，可以使用该功能更新集群相关数据，确保ADMQ集群服务可以被正常访问。

备注：（1）使用该功能前确保集群服务正常启动。（2）若首次部署集群初始化失败，可使用该功能进行重新初始化。

进入集群管理页面，在集群列表页面选择需要更新元数据的集群，【更多】里点击【更新元数据】按钮，出现提示弹窗，点击【确定】按钮，进行更新元数据。

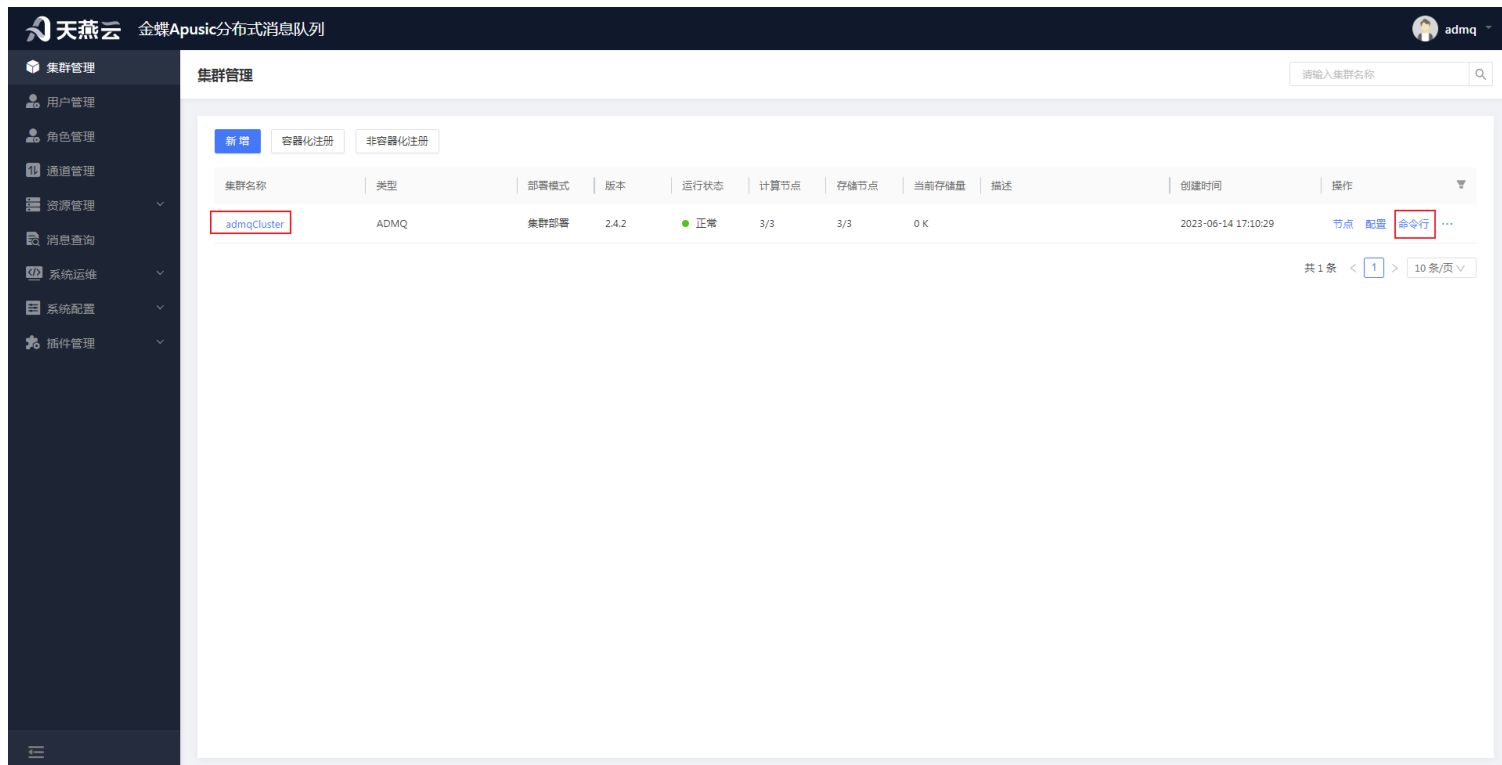


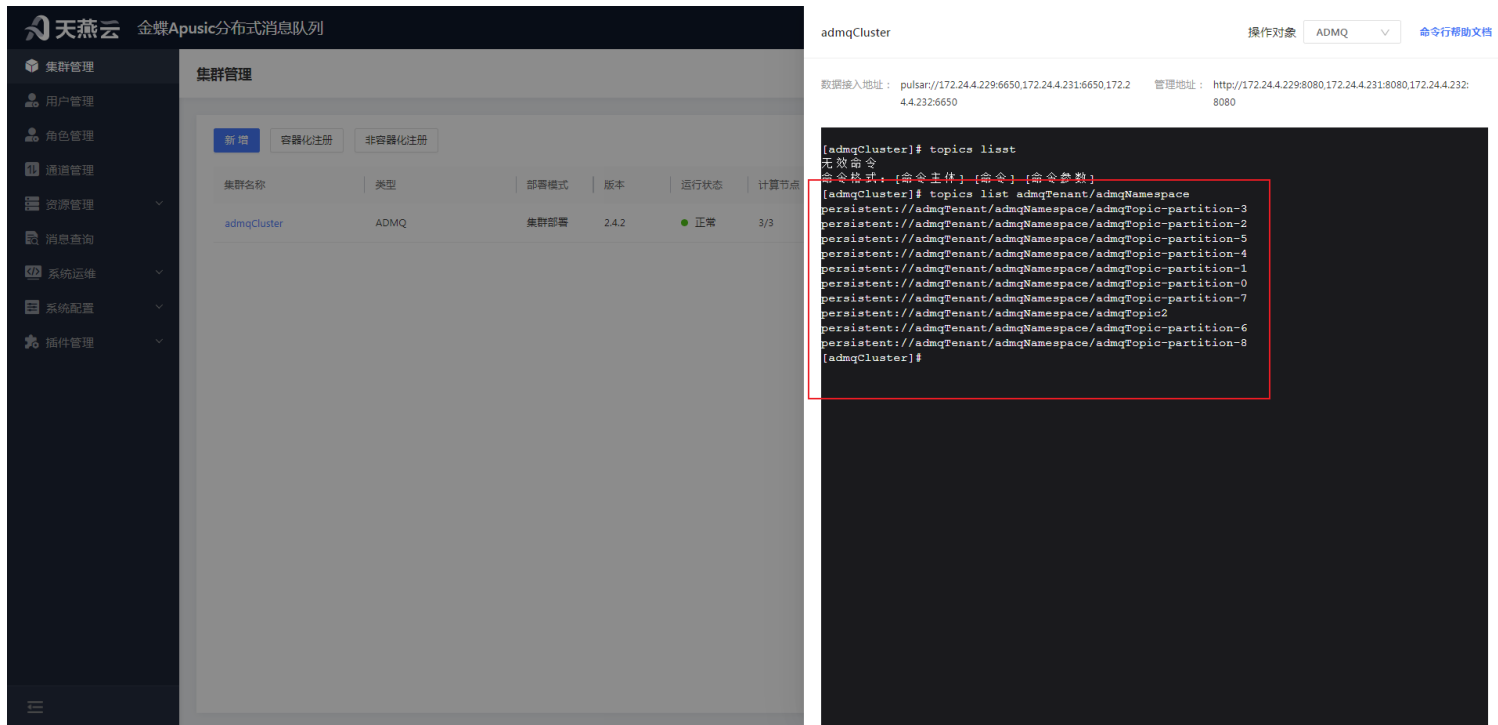


5.3.1.13 命令行操作

ADMQ管控台提供了命令行界面工具，通过该图形化界面，可执行各种管理任务，例如创建和删除主题、订阅组、命名空间等操作，为运维人员和开发人员提供了一种快速而强大的方式来管理和监控ADMQ实例。

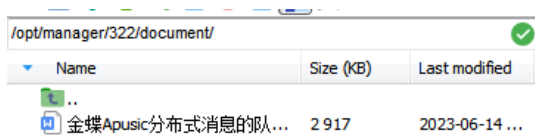
具体使用步骤：进入集群管理页面，在集群列表页面选择集群，点击【命令行】按钮，页面右侧会出现弹窗界面，在此界面可输入相关命令行：



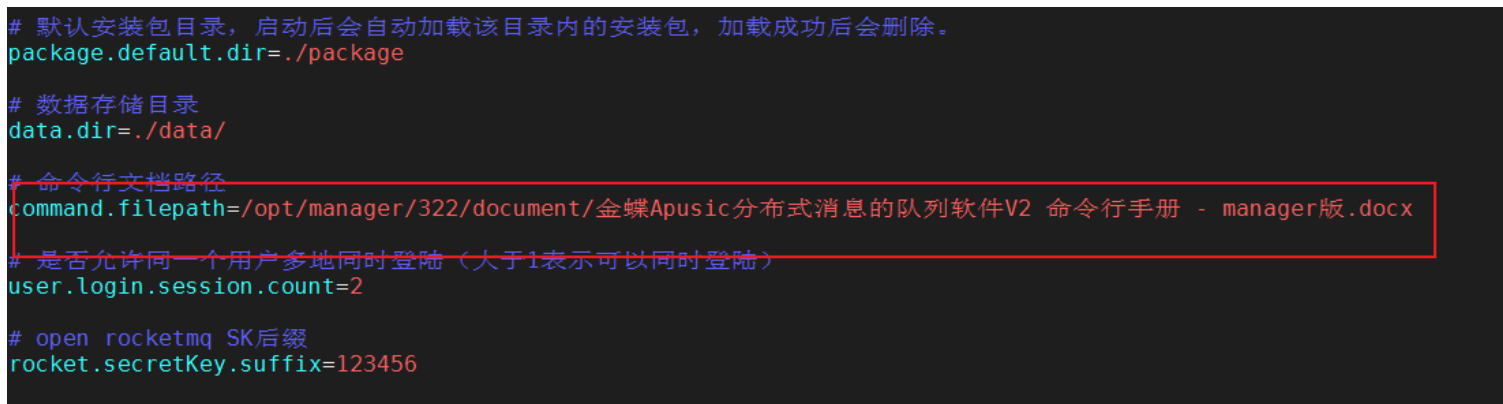


备注：使用命令行操作界面右上角【命令行帮助文档】进行下载文件之前需要先将相关命令行操作文档上传至服务器，然后在管控台的配置文件中配置相关参数，重启管控台后方可进行下载，具体操作过程如下：

(1) 上传命令行文件



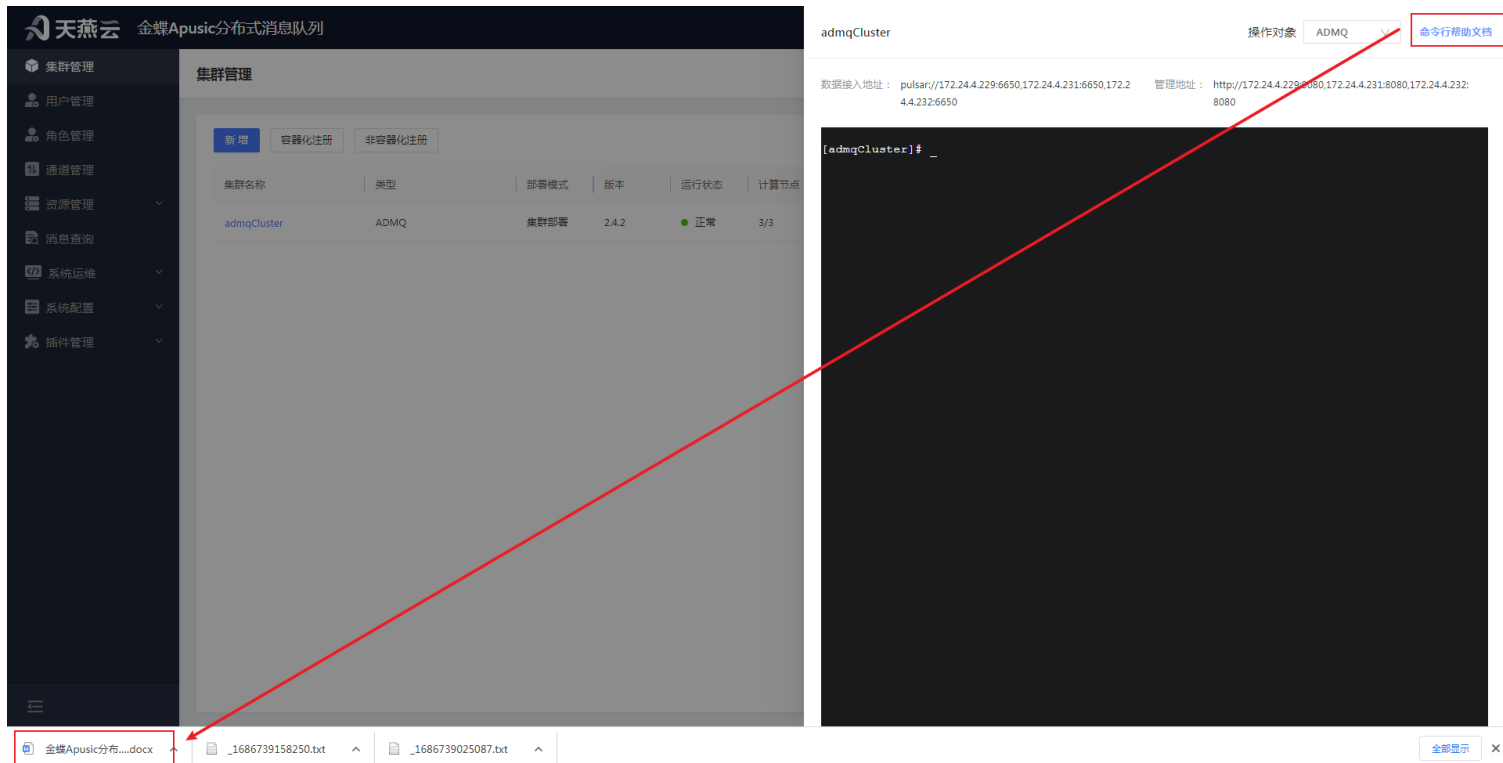
(2) 修改管控台配置文件参数



(3) 重启管控台

- 停止操作：bin/admq-manager stop
- 启动操作：bin/admq-manager start

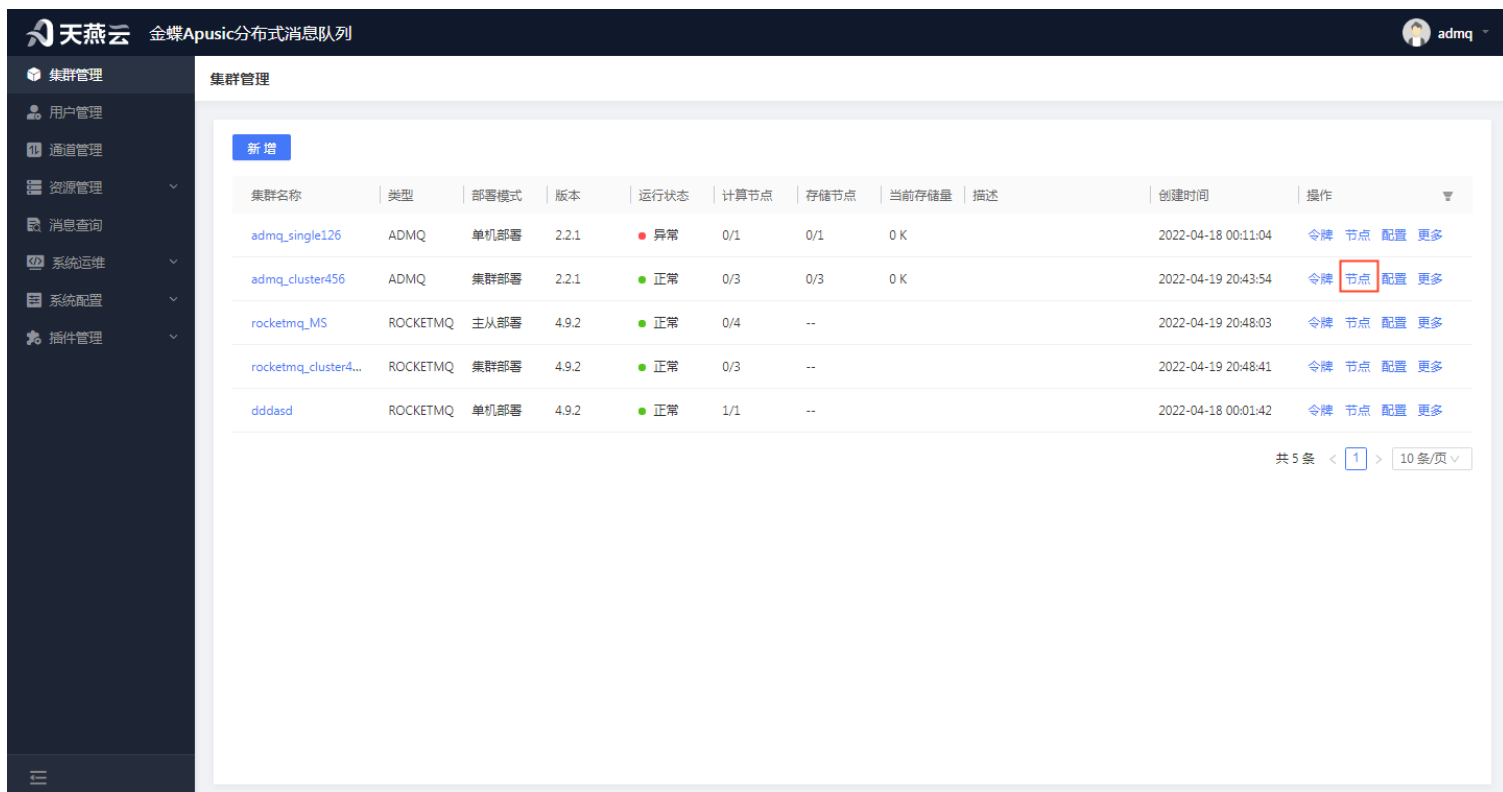
(4) 在命令行操作界面，点击右上角【命令行帮助文档】，成功下载上传至服务器的文档：



5.3.1.14 节点相关操作

5.3.1.14.1 部署节点

进入【集群管理】，在集群管理页面，选择需要部署节点的集群信息，点击【节点】按钮，进入部署节点页面；选择需要部署的节点，点击【部署】按钮，可部署节点。



节点

集群名称	节点类型	部署机房	节点IP	节点状态	部署状态	描述	创建时间	操作
adm_q_cluster456	admq协调器	star1	172.24.4.104	未运行	未部署		2022-04-19 20:43:54	监控 部署 更多
adm_q_cluster456	admq协调器	star1	172.24.4.105	未运行	未部署		2022-04-19 20:43:54	监控 部署 更多
adm_q_cluster456	admq协调器	star1	172.24.4.106	未运行	未部署		2022-04-19 20:43:54	监控 部署 更多
adm_q_cluster456	存储程序	star1	172.24.4.104	未运行	未部署		2022-04-19 20:43:54	监控 部署 更多
adm_q_cluster456	存储程序	star1	172.24.4.105	未运行	未部署		2022-04-19 20:43:54	监控 部署 更多
adm_q_cluster456	存储程序	star1	172.24.4.106	未运行	未部署		2022-04-19 20:43:54	监控 部署 更多
adm_q_cluster456	admq计算程序	star1	172.24.4.104	未运行	未部署		2022-04-19 20:43:54	监控 部署 更多
adm_q_cluster456	admq计算程序	star1	172.24.4.105	未运行	未部署		2022-04-19 20:43:54	监控 部署 更多
adm_q_cluster456	admq计算程序	star1	172.24.4.106	未运行	未部署		2022-04-19 20:43:54	监控 部署 更多

共 9 条 < 1 > 10 条/页

- 备注：若是集群部署，则集群的每个节点都需要点击【部署】按钮，部署节点前，需先新建集群信息。

5.3.1.14.2 重新部署节点

当节点部署时，可能由于服务器网络等原因，部署状态显示异常，需要重新进行部署；进入【集群管理】，在集群管理页面，选择需要重新部署节点的集群信息，点击【节点】按钮，进入部署节点页面；选择部署状态异常的节点，点击【重新部署】按钮，可重新部署节点。

集群管理

集群名称	类型	部署模式	版本	运行状态	计算节点	存储节点	当前存储量	描述	创建时间	操作
adm_q_single126	ADMQ	单机部署	2.2.1	异常	0/1	0/1	0 K		2022-04-18 00:11:04	令牌 节点 配置 更多
adm_q_cluster456	ADMQ	集群部署	2.2.1	正常	0/3	0/3	0 K		2022-04-19 20:43:54	令牌 节点 配置 更多
rocketmq_MS	ROCKETMQ	主从部署	4.9.2	正常	0/4	--			2022-04-19 20:48:03	令牌 节点 配置 更多
rocketmq_cluster4...	ROCKETMQ	集群部署	4.9.2	正常	0/3	--			2022-04-19 20:48:41	令牌 节点 配置 更多
dddasd	ROCKETMQ	单机部署	4.9.2	正常	1/1	--			2022-04-18 00:01:42	令牌 节点 配置 更多

共 5 条 < 1 > 10 条/页

节点

新增 部署 启动 停止

集群名称	节点类型	部署机房	节点IP	节点状态	部署状态	描述	创建时间	操作
admq_cluster456	admq协调器	star1	172.24.4.104	未运行	异常		2022-04-19 20:43:54	监控 重新部署 更多
admq_cluster456	admq协调器	star1	172.24.4.105	未运行	异常		2022-04-19 20:43:54	监控 重新部署 更多
admq_cluster456	admq协调器	star1	172.24.4.106	未运行	部署中		2022-04-19 20:43:54	监控 重新部署 更多
admq_cluster456	存储程序	star1	172.24.4.104	未运行	部署中		2022-04-19 20:43:54	监控 重新部署 更多
admq_cluster456	存储程序	star1	172.24.4.105	未运行	部署中		2022-04-19 20:43:54	监控 重新部署 更多
admq_cluster456	存储程序	star1	172.24.4.106	未运行	部署中		2022-04-19 20:43:54	监控 重新部署 更多
admq_cluster456	admq计算程序	star1	172.24.4.104	未运行	部署中		2022-04-19 20:43:54	监控 重新部署 更多
admq_cluster456	admq计算程序	star1	172.24.4.105	未运行	部署中		2022-04-19 20:43:54	监控 重新部署 更多
admq_cluster456	admq计算程序	star1	172.24.4.106	未运行	部署中		2022-04-19 20:43:54	监控 重新部署 更多

共 9 条 < 1 > 10 条/页

- 备注：重新部署节点，需先部署状态为异常的节点才可重新部署。

5.3.1.14.3 启动节点

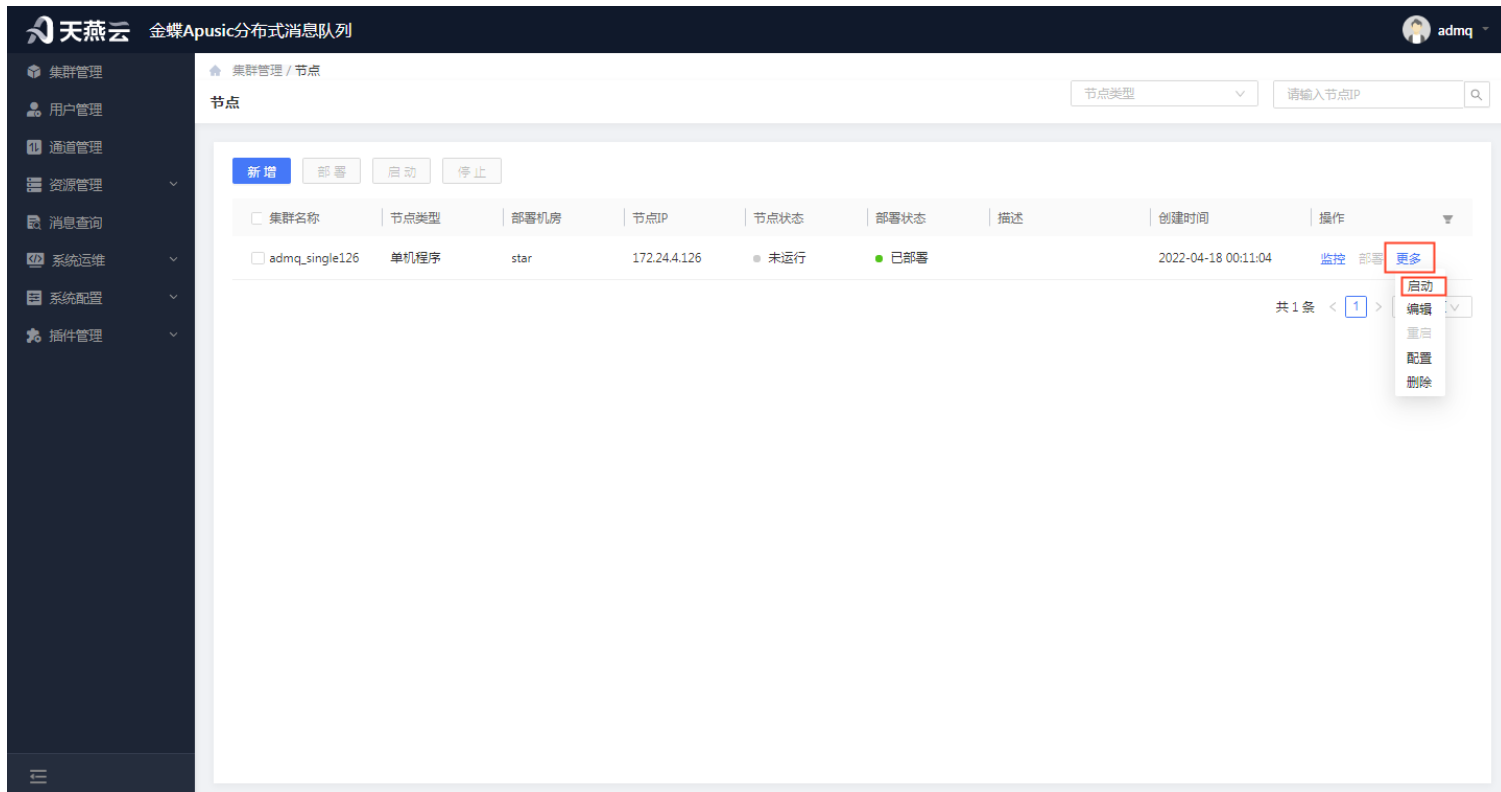
进入【集群管理】，在集群管理页面，选择需要启动节点的集群信息，点击【节点】按钮，进入部署节点页面，点击【更多】按钮—>【启动】按钮，可启动节点。

集群管理

新增

集群名称	类型	部署模式	版本	运行状态	计算节点	存储节点	当前存储量	描述	创建时间	操作
admq_cluster456	ADMQ	集群部署	2.2.1	正常	0/3	0/3	0 K		2022-04-19 20:43:54	令牌 节点 配置 更多
admq_single126	ADMQ	单机部署	2.2.1	异常	0/1	0/1	0 K		2022-04-18 00:11:04	令牌 节点 配置 更多
rocketmq_MS	ROCKETMQ	主从部署	4.9.2	正常	0/4	--			2022-04-19 20:48:03	令牌 节点 配置 更多
rocketmq_cluster4...	ROCKETMQ	集群部署	4.9.2	正常	0/3	--			2022-04-19 20:48:41	令牌 节点 配置 更多
dddasd	ROCKETMQ	单机部署	4.9.2	正常	1/1	--			2022-04-18 00:01:42	令牌 节点 配置 更多

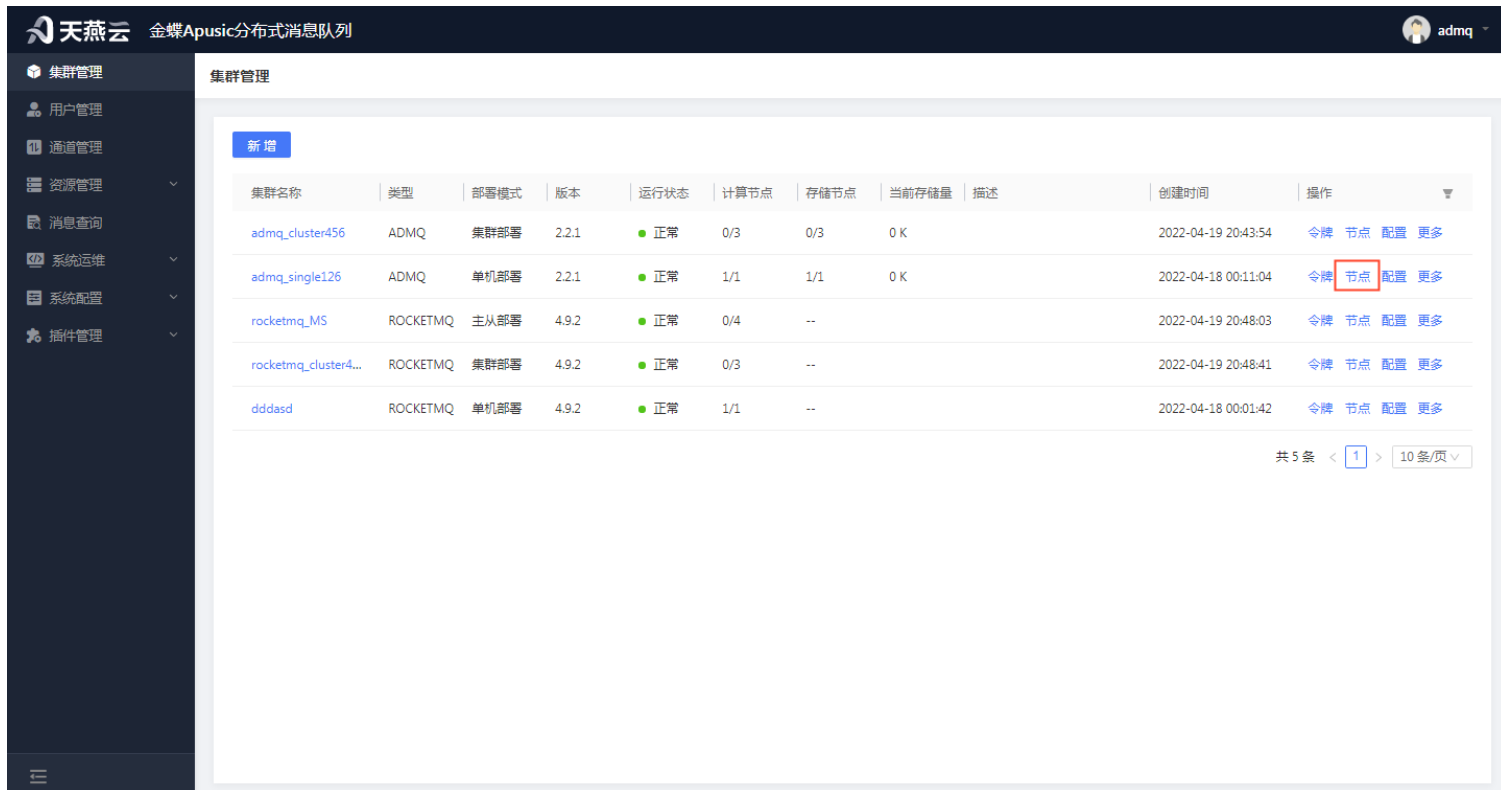
共 5 条 < 1 > 10 条/页

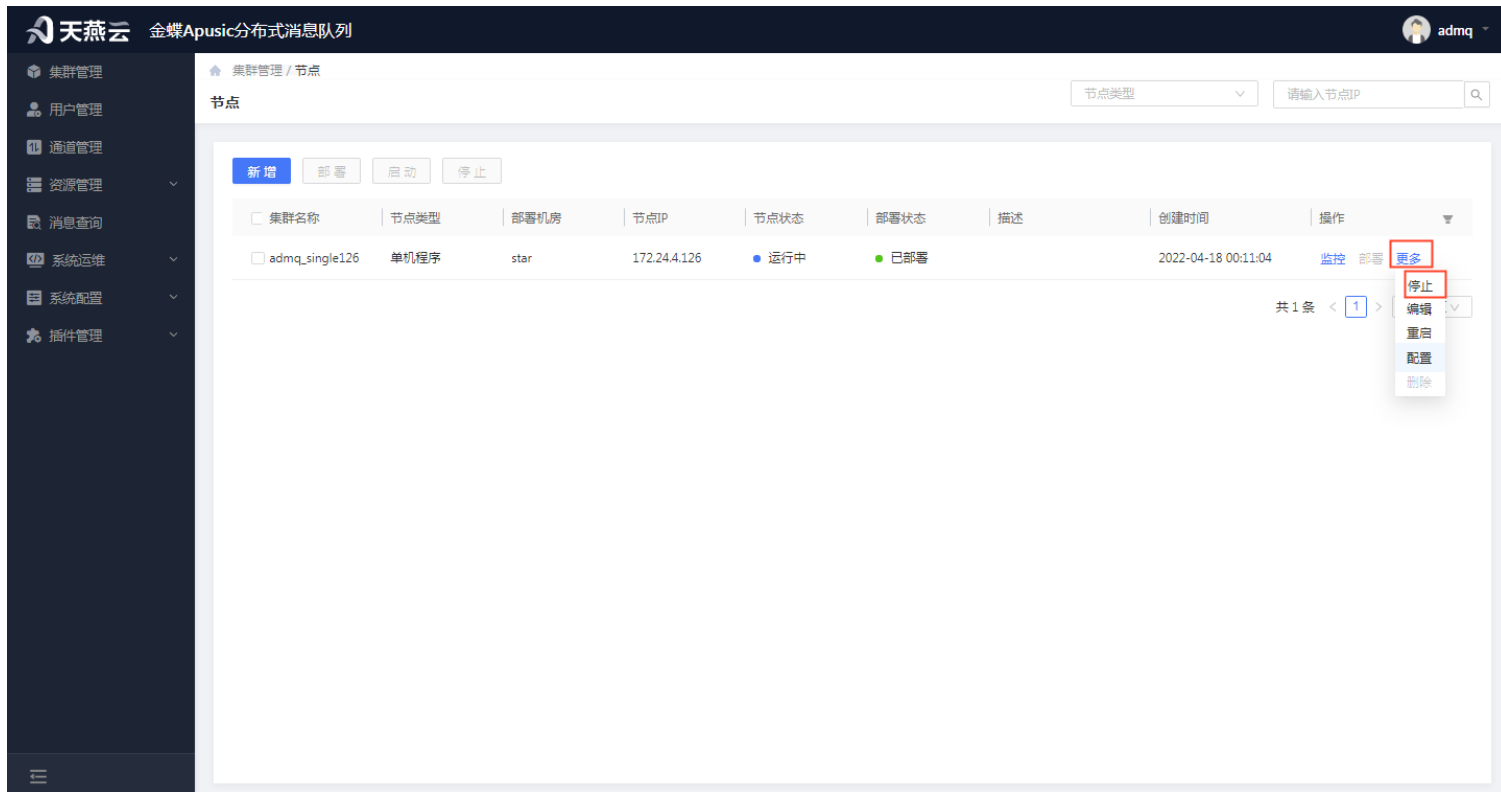


- 备注：未部署的节点不可启动，已启动的节点也不可再进行启动，若是单机，则只需要启动一个单机程序，若是集群，可选中左上角“集群名称”左边的方框，实现一键启动多个节点的效果。

5.3.1.14.4 停止节点

进入【集群管理】，在集群管理页面，选择需要启动节点的集群信息，点击【节点】按钮，进入部署节点页面，点击【更多】按钮—>【停止】按钮，可停止节点。

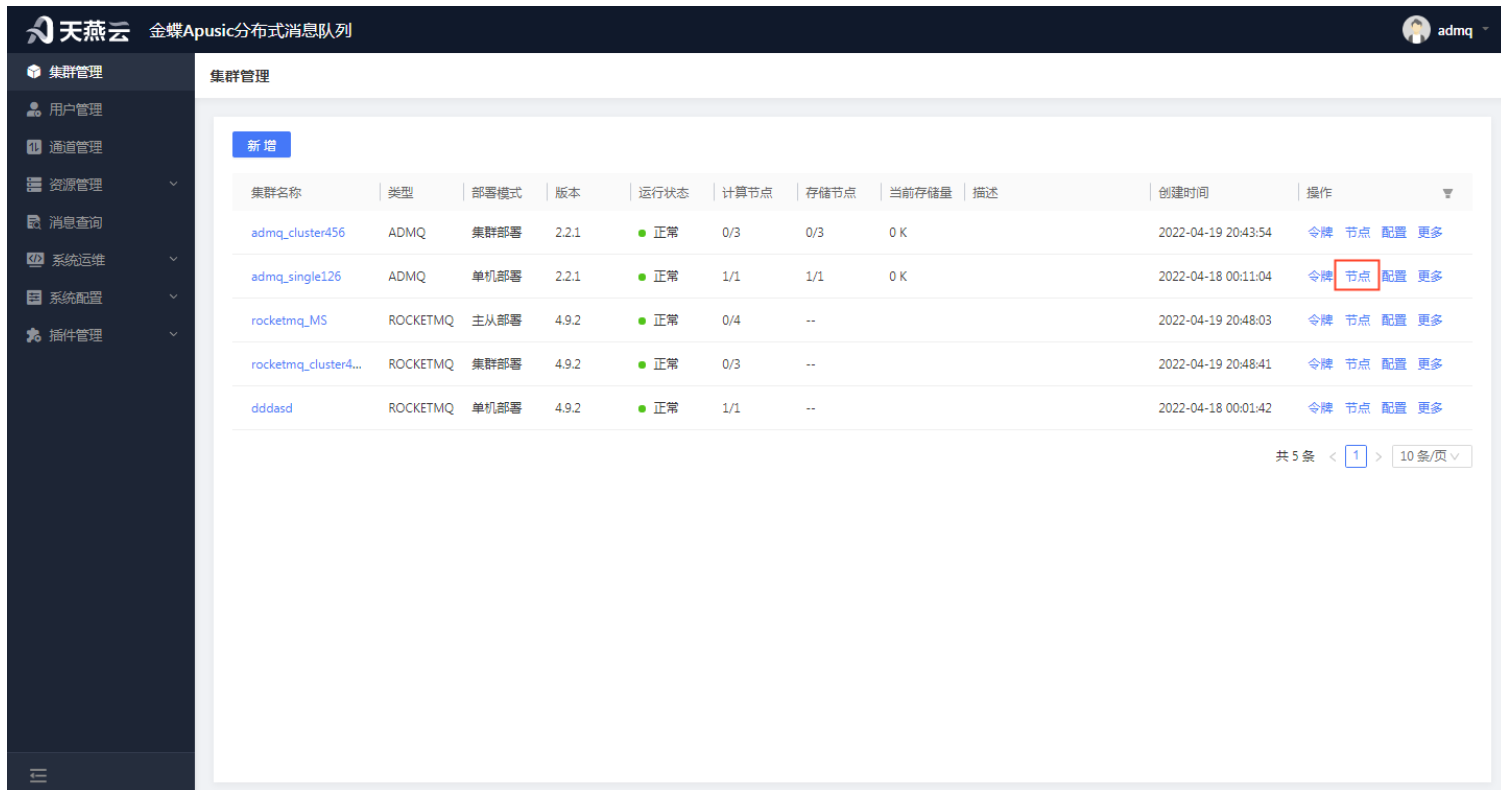


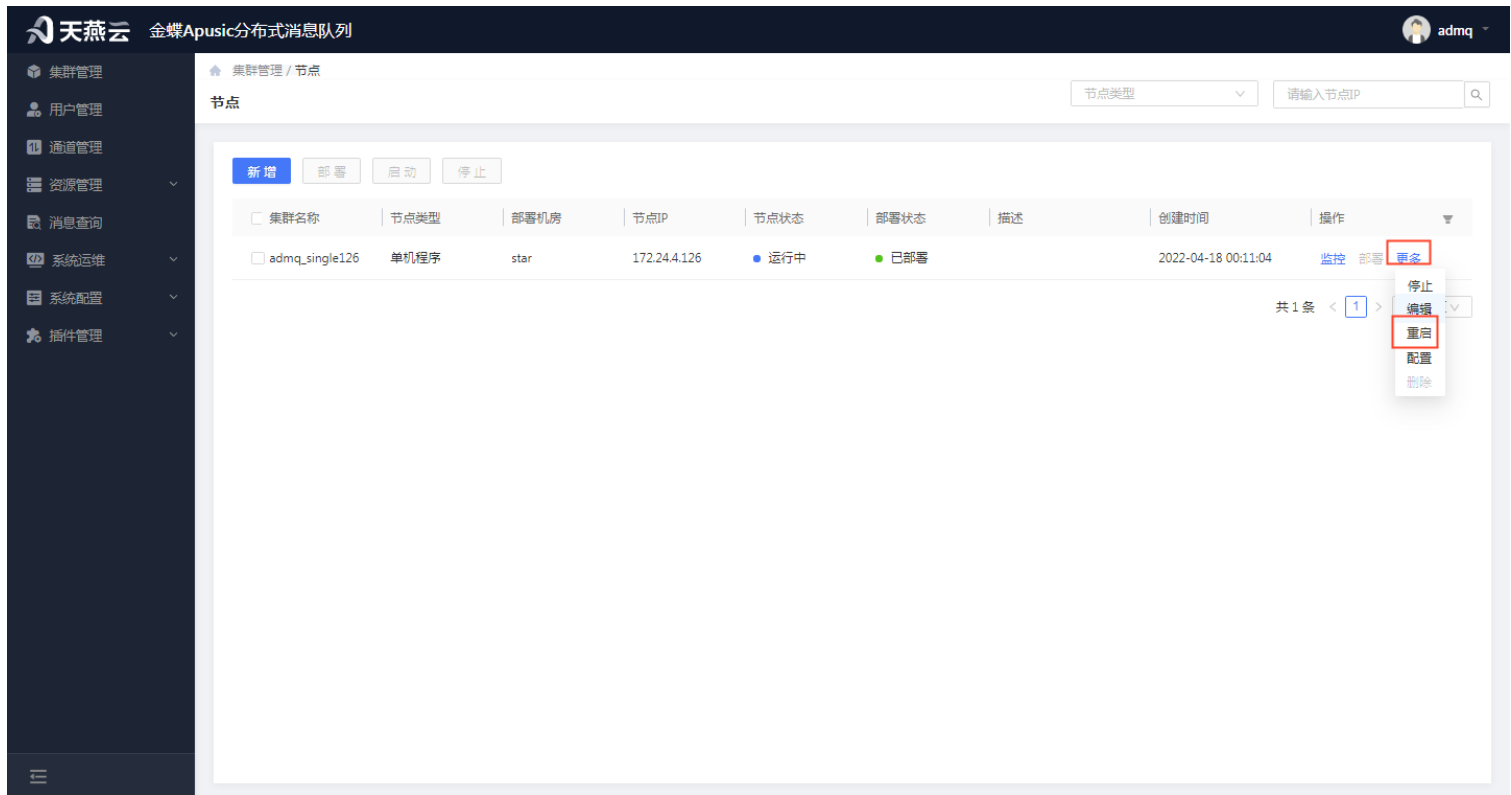


- 备注：只有已启动的节点可以停止，未启动的节点和未部署的节点不可进行停止。若是集群，可选中左上角“集群名称”左边的方框，实现一键停止多个节点的效果。

5.3.1.14.5 重启节点

进入【集群管理】，在集群管理页面，选择需要启动节点的集群信息，点击【节点】按钮，进入部署节点页面，点击【更多】按钮—>【重启】按钮，可停止节点。

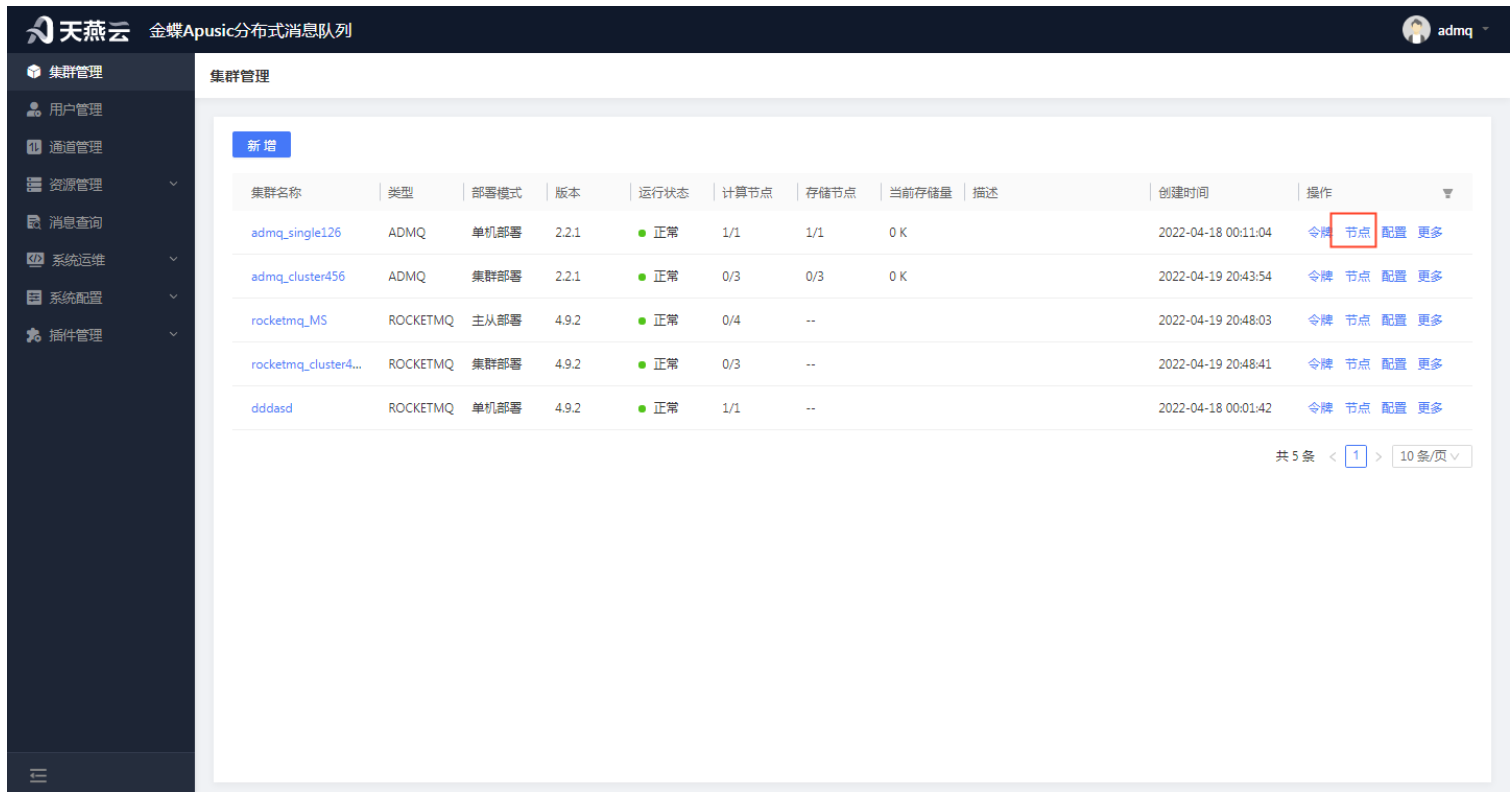


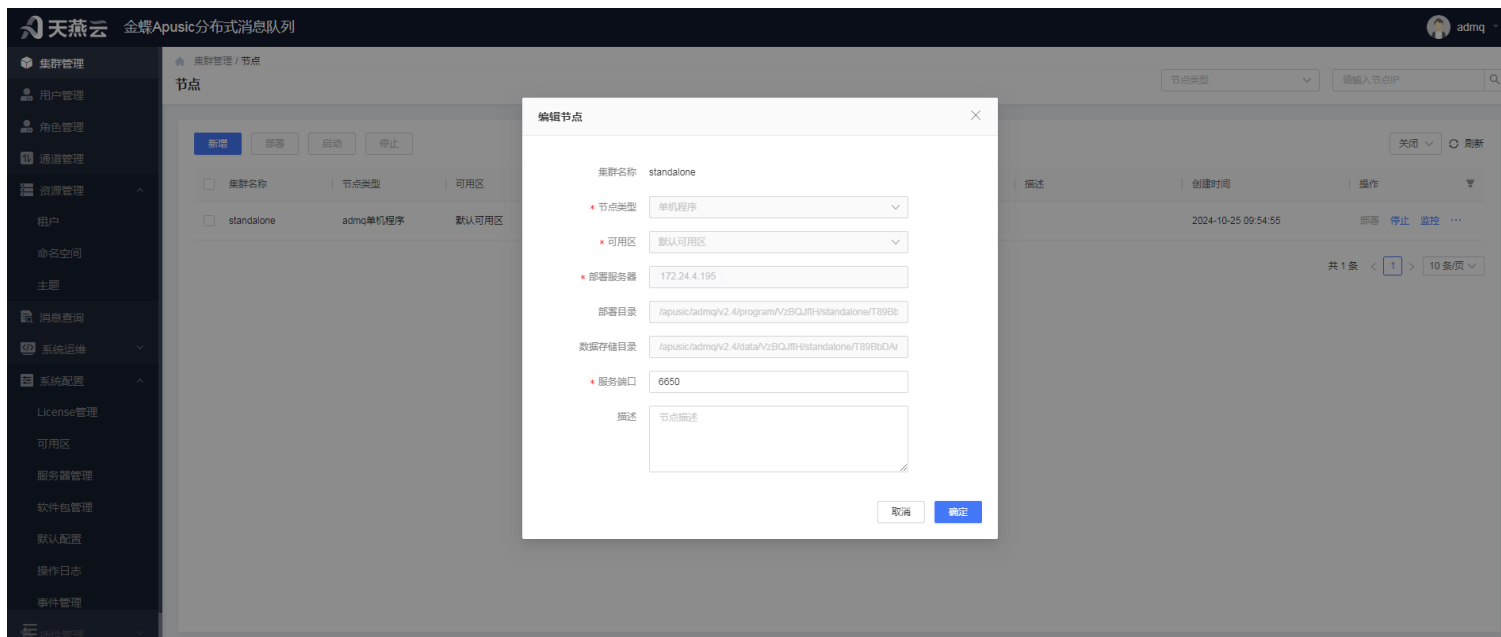
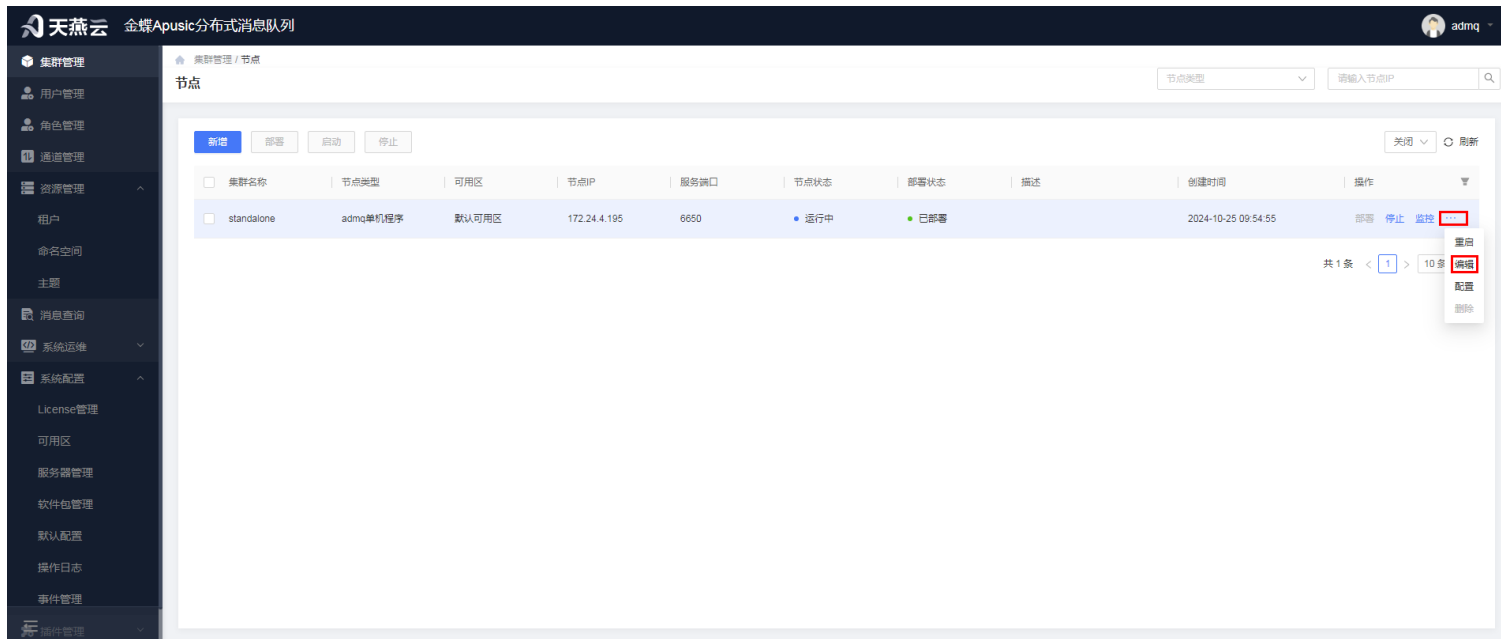


- 备注：只有已启动的节点可以停止，未启动的节点和未部署的节点不可进行停止。

5.3.1.14.6 编辑节点

进入【集群管理】，在集群管理页面，选择需要编辑节点的集群信息，点击【节点】按钮，进入部署节点页面，点击【更多】按钮->【编辑】按钮，在弹出的编辑节点弹框中可对节点信息进行编辑。





只可对描述信息进行重新编辑。

5.3.1.15 集群扩缩容

5.3.1.15.1 删除节点

进入【集群管理】，在集群管理页面，选择需要删除节点的集群信息，点击【节点】按钮，进入部署节点页面，点击【更多】按钮->【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除节点。

天燕云 金蝶Apusic分布式消息队列
admq

- 集群管理
- 用户管理
- 通道管理
- 资源管理
- 消息查询
- 系统运维
- 系统配置
- 插件管理

集群管理

新增

集群名称	类型	部署模式	版本	运行状态	计算节点	存储节点	当前存储量	描述	创建时间	操作
admq_single126	ADMQ	单机部署	2.2.1	● 正常	1/1	1/1	0 K		2022-04-18 00:11:04	令牌 节点 配置 更多
admq_cluster456	ADMQ	集群部署	2.2.1	● 正常	0/3	0/3	0 K		2022-04-19 20:43:54	令牌 节点 配置 更多
rocketmq_M5	ROCKETMQ	主从部署	4.9.2	● 正常	0/4	--			2022-04-19 20:48:03	令牌 节点 配置 更多
rocketmq_cluster4...	ROCKETMQ	集群部署	4.9.2	● 正常	0/3	--			2022-04-19 20:48:41	令牌 节点 配置 更多
dddasd	ROCKETMQ	单机部署	4.9.2	● 正常	1/1	--			2022-04-18 00:01:42	令牌 节点 配置 更多

共 5 条 < 1 > 10 条/页

天燕云 金蝶Apusic分布式消息队列
admq

- 集群管理
- 用户管理
- 通道管理
- 资源管理
- 消息查询
- 系统运维
- 系统配置
- 插件管理

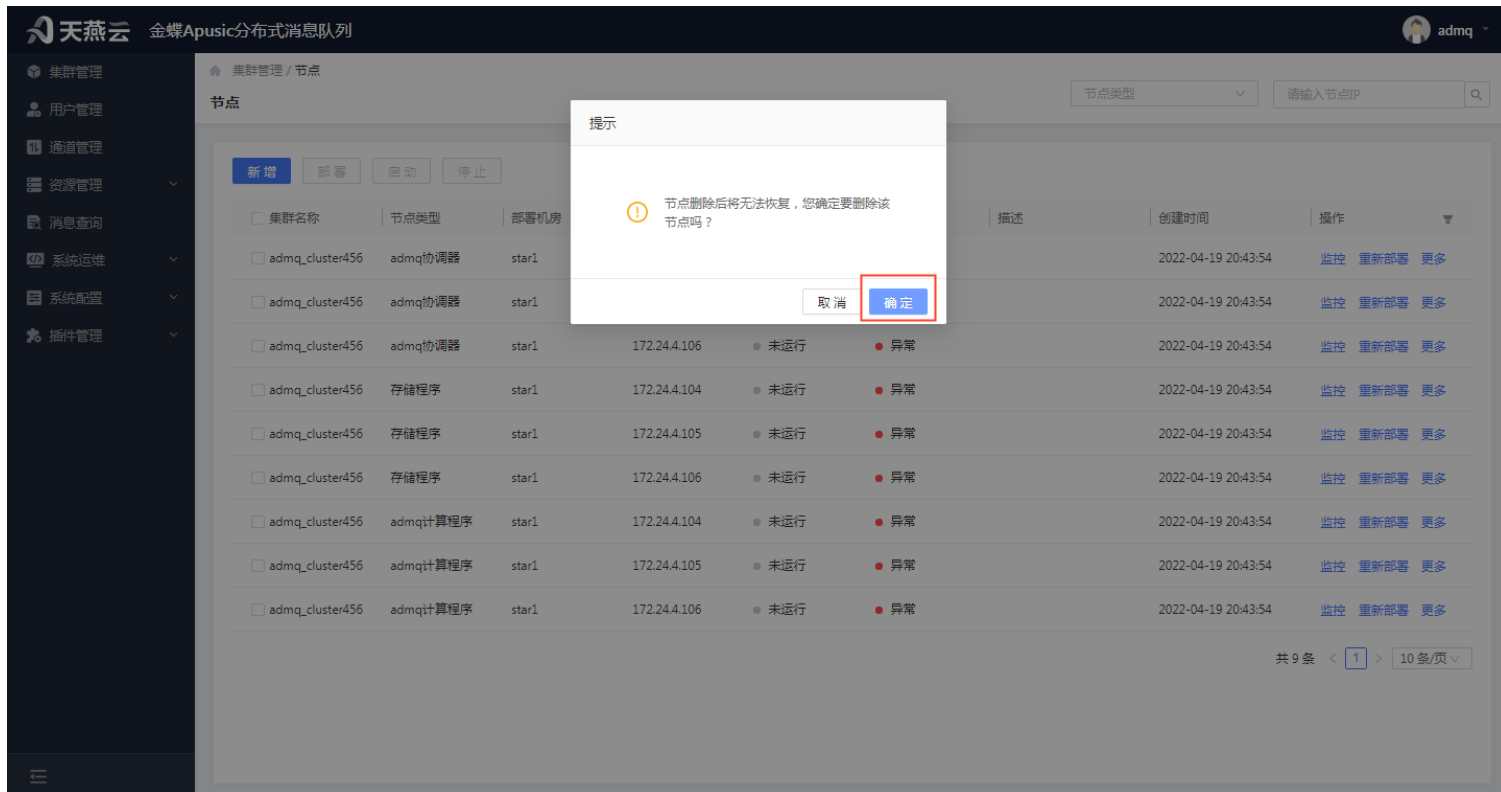
节点

新增
部署
启动
停止

节点类型
请输入节点IP

集群名称	节点类型	部署机房	节点IP	节点状态	部署状态	描述	创建时间	操作
admq_cluster456	admq协调器	star1	172.24.4.104	● 未运行	● 异常		2022-04-19 20:43:54	监控 重新部署 更多
admq_cluster456	admq协调器	star1	172.24.4.105	● 未运行	● 异常		2022-04-19 20:43:54	监控 重新部署
admq_cluster456	admq协调器	star1	172.24.4.106	● 未运行	● 异常		2022-04-19 20:43:54	监控 重新部署
admq_cluster456	存储程序	star1	172.24.4.104	● 未运行	● 异常		2022-04-19 20:43:54	监控 重新部署 删除
admq_cluster456	存储程序	star1	172.24.4.105	● 未运行	● 异常		2022-04-19 20:43:54	监控 重新部署 更多
admq_cluster456	存储程序	star1	172.24.4.106	● 未运行	● 异常		2022-04-19 20:43:54	监控 重新部署 更多
admq_cluster456	admq计算程序	star1	172.24.4.104	● 未运行	● 异常		2022-04-19 20:43:54	监控 重新部署 更多
admq_cluster456	admq计算程序	star1	172.24.4.105	● 未运行	● 异常		2022-04-19 20:43:54	监控 重新部署 更多
admq_cluster456	admq计算程序	star1	172.24.4.106	● 未运行	● 异常		2022-04-19 20:43:54	监控 重新部署 更多

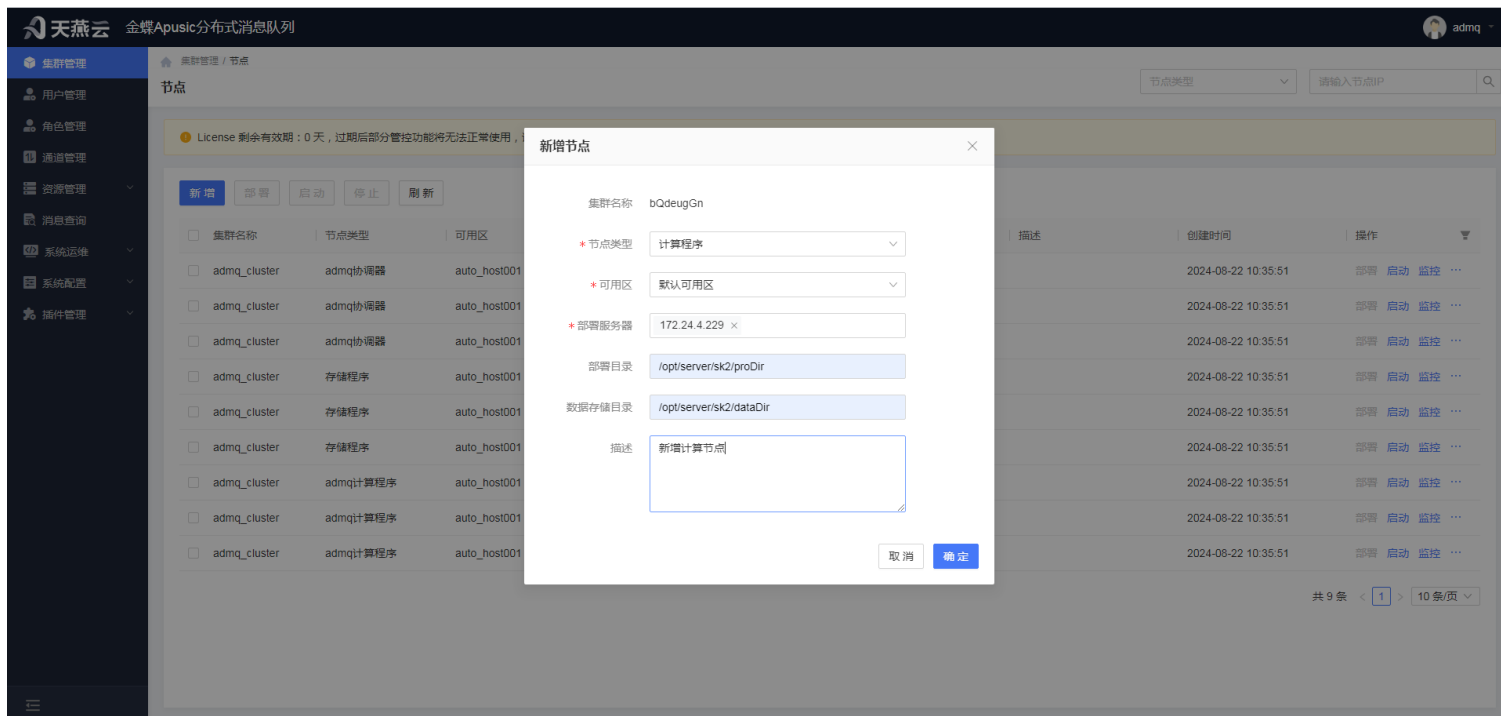
共 9 条 < 1 > 10 条/页



- 备注：只有未启动的节点才可以删除，单机节点不可以进行删除，直接删除集群即可。

5.3.1.15.2 新增节点

进入【集群管理】，在集群管理页面，点击左上角【新增】按钮，出现新增节点弹窗，输入相关信息，点击【确定】按钮，然后在节点列表页面对新增的节点进行部署操作和启动操作（备注：新增节点之前需要将节点服务器上传到服务器管理模块中）：



- 集群名称：新增节点所在的集群名称，不可修改
- 节点类型：新增节点的类型，下拉框，三种选项：协调器、存储程序、计算程序
- 可用区：节点服务器所属的可用区查看权限

- 部署服务器：新增节点服务器的IP值
- 部署目录：软件的部署目录，不是必选项，不填是默认使用服务器管理中的默认软件部署目录，可自己重新输入覆盖默认值。
- 数据存储目录：数据的存储目录，不是必选项，不填是默认使用服务器管理中的默认数据部署目录，可自己重新输入覆盖默认值。
- 存储服务端口：存储节点的服务端口
- 描述：新增节点的描述信息

节点正在部署...

集群名称	节点类型	部署机房	节点IP	节点状态	部署状态	描述	创建时间	操作
admqCluster	admq协调器	深圳	172.24.4.229	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	admq协调器	深圳	172.24.4.231	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	admq协调器	深圳	172.24.4.232	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	存储程序	深圳	172.24.4.229	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	存储程序	深圳	172.24.4.231	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	存储程序	深圳	172.24.4.232	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	admq计算程序	深圳	172.24.4.229	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	admq计算程序	深圳	172.24.4.231	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	admq计算程序	深圳	172.24.4.232	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	admq计算程序	深圳	172.24.4.230	未运行	部署中	新增计算节点	2023-06-14 15:31:49	重新部署 启动 监控 ...

共 10 条 < 1 > 10 条/页

节点启动命令发送成功!

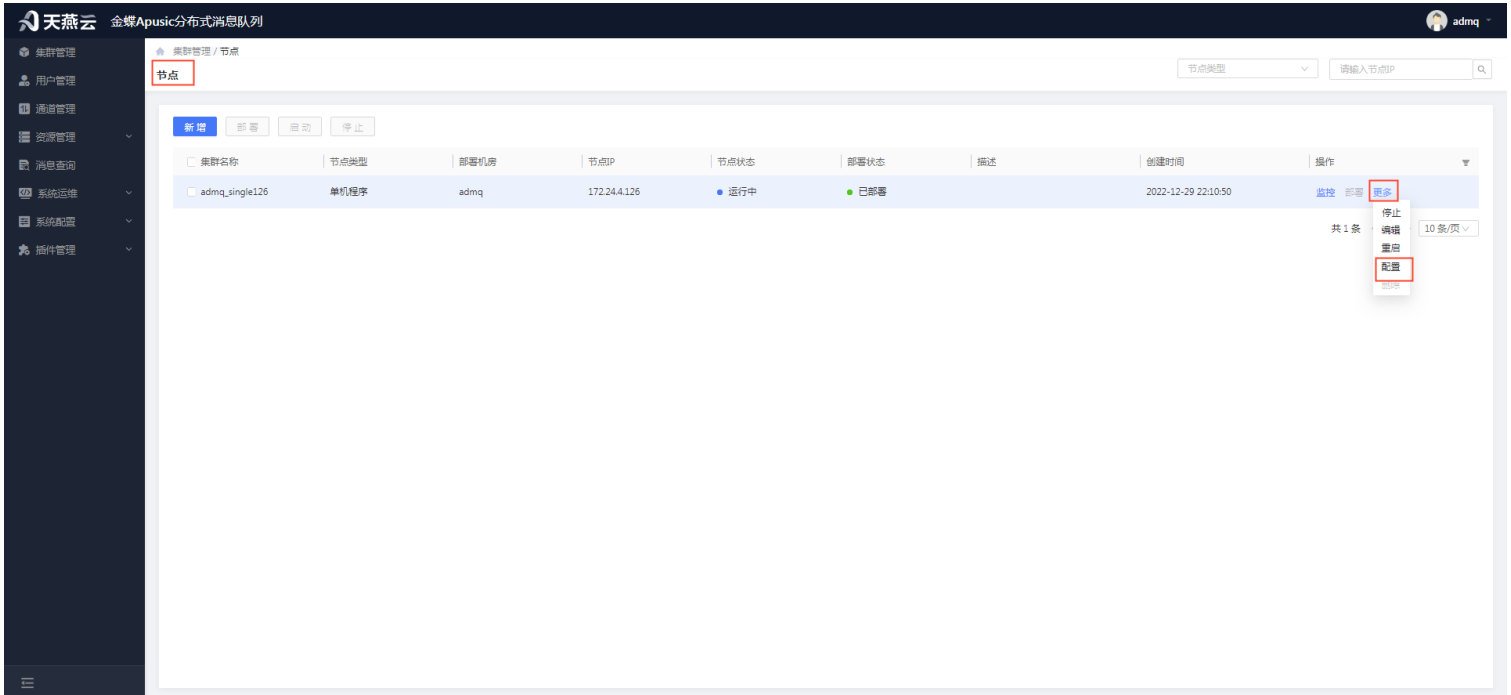
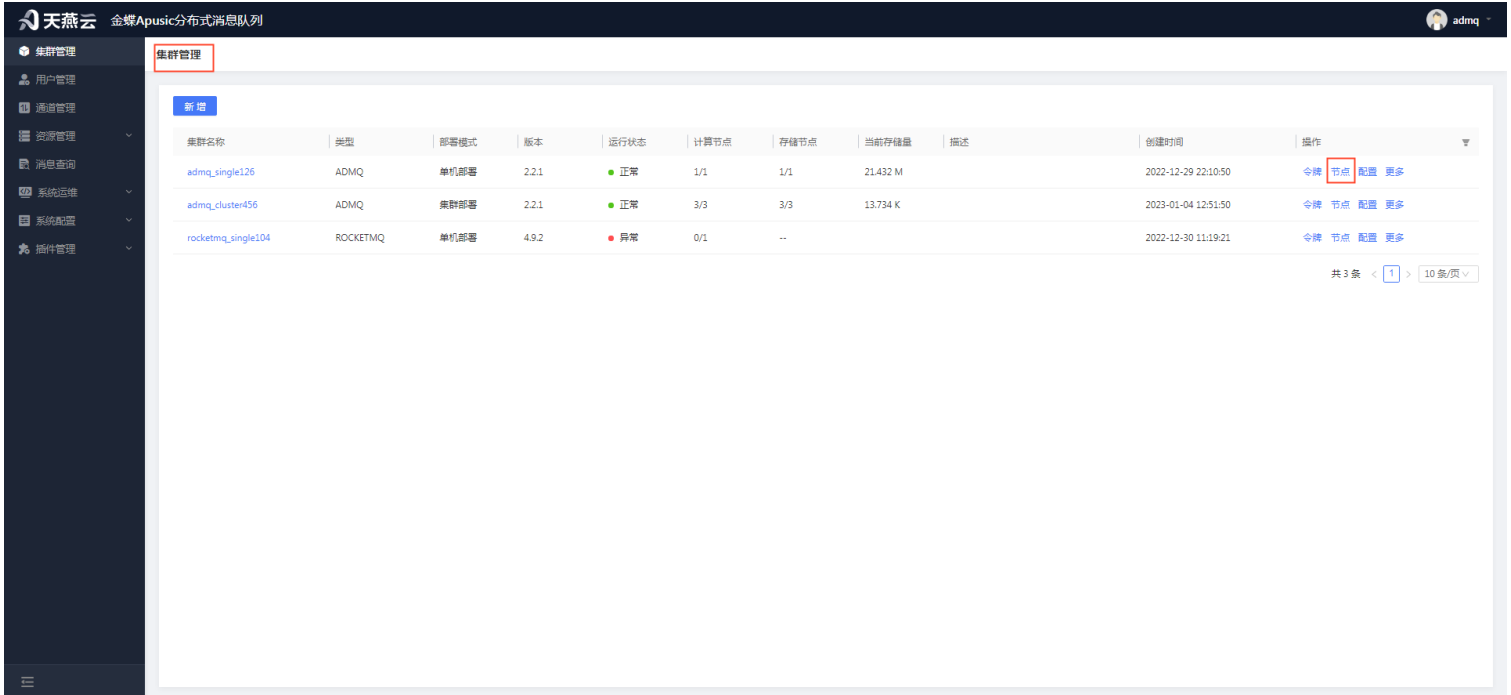
集群名称	节点类型	部署机房	节点IP	节点状态	部署状态	描述	创建时间	操作
admqCluster	admq协调器	深圳	172.24.4.229	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	admq协调器	深圳	172.24.4.231	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	admq协调器	深圳	172.24.4.232	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	存储程序	深圳	172.24.4.229	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	存储程序	深圳	172.24.4.231	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	存储程序	深圳	172.24.4.232	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	admq计算程序	深圳	172.24.4.229	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	admq计算程序	深圳	172.24.4.231	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	admq计算程序	深圳	172.24.4.232	运行中	已部署		2023-06-14 10:39:41	部署 停止 监控 ...
admqCluster	admq计算程序	深圳	172.24.4.230	未运行	已部署	新增计算节点	2023-06-14 15:31:49	部署 启动 监控 ...

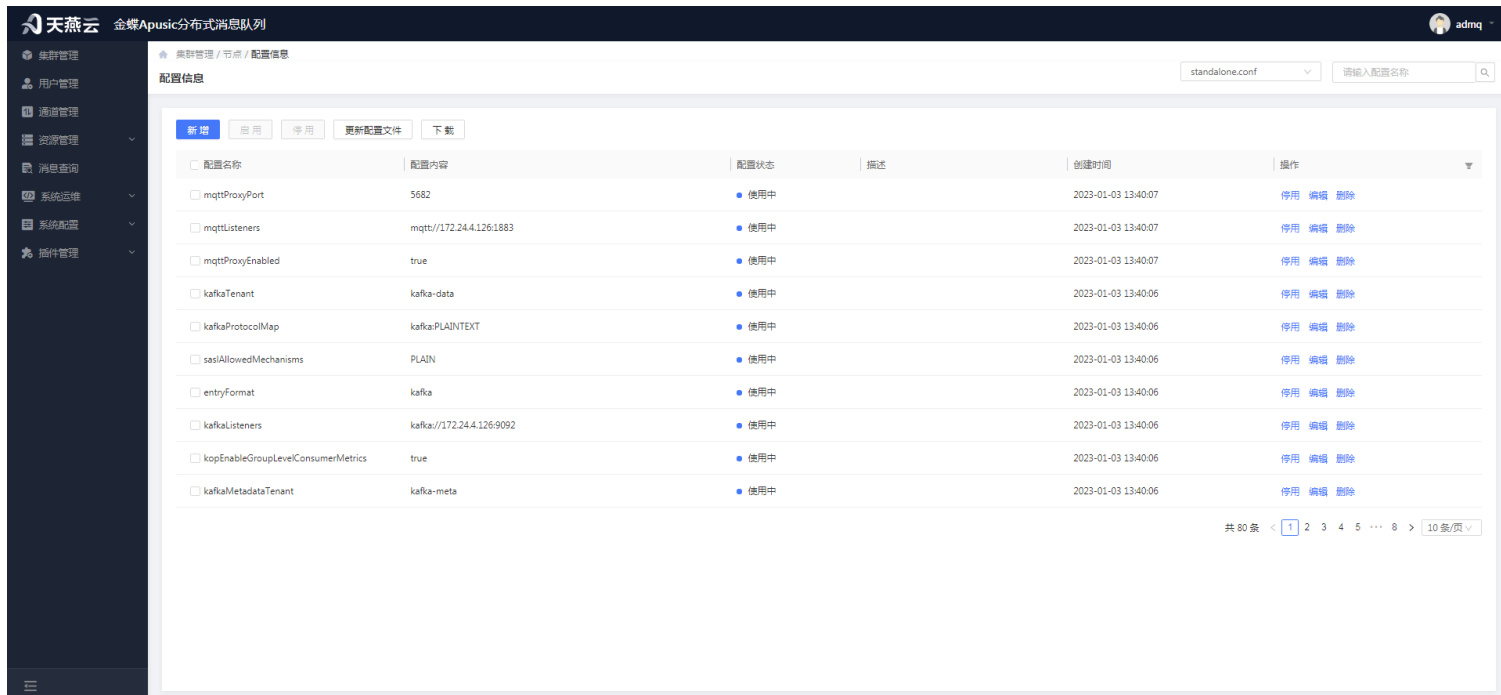
共 10 条 < 1 > 10 条/页

5.3.1.16 配置节点参数

节点配置信息是为了更方便的管理节点配置文件，可新增配置参数、编辑配置参数、删除配置参数、更新配置参数、启用配置参数、停用配置参数、下载配置文件、查询配置参数。

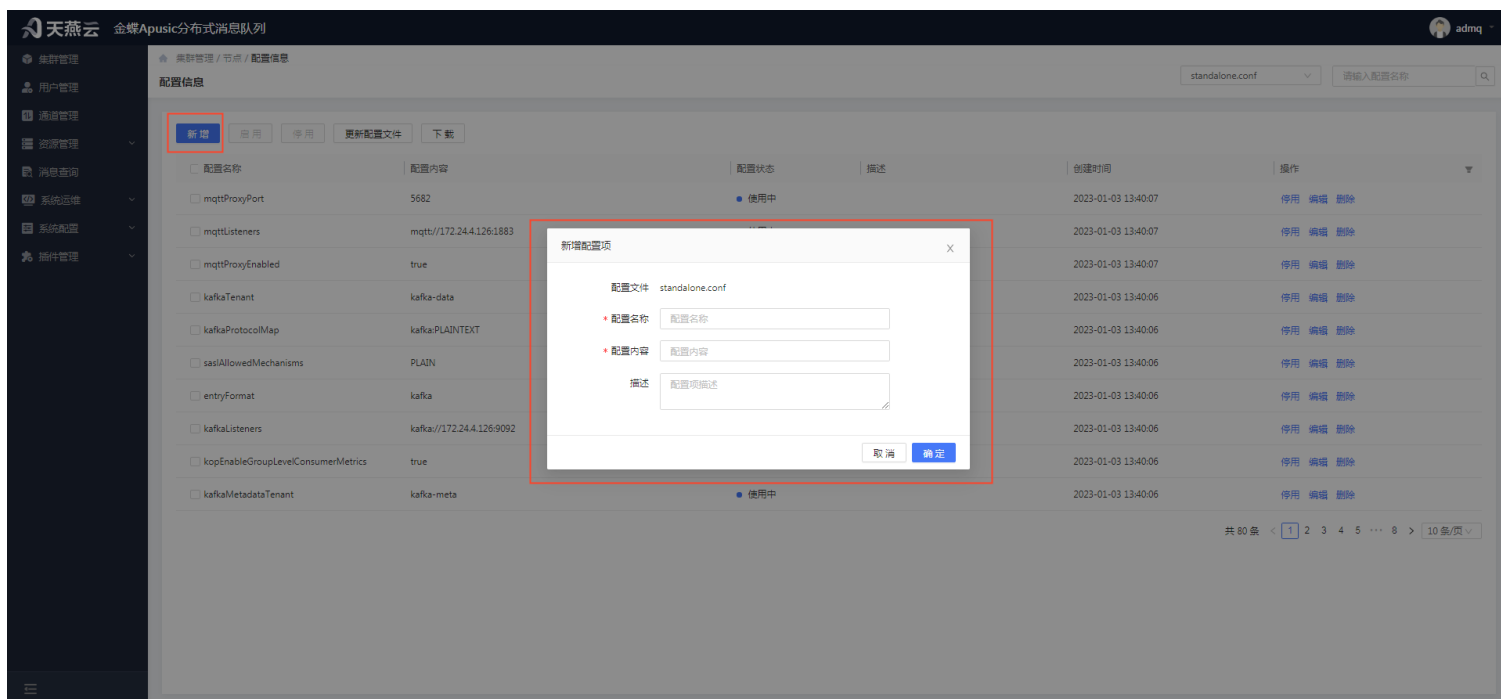
进入节点“配置信息”页面步骤如下：





5.3.1.16.1 新增配置参数

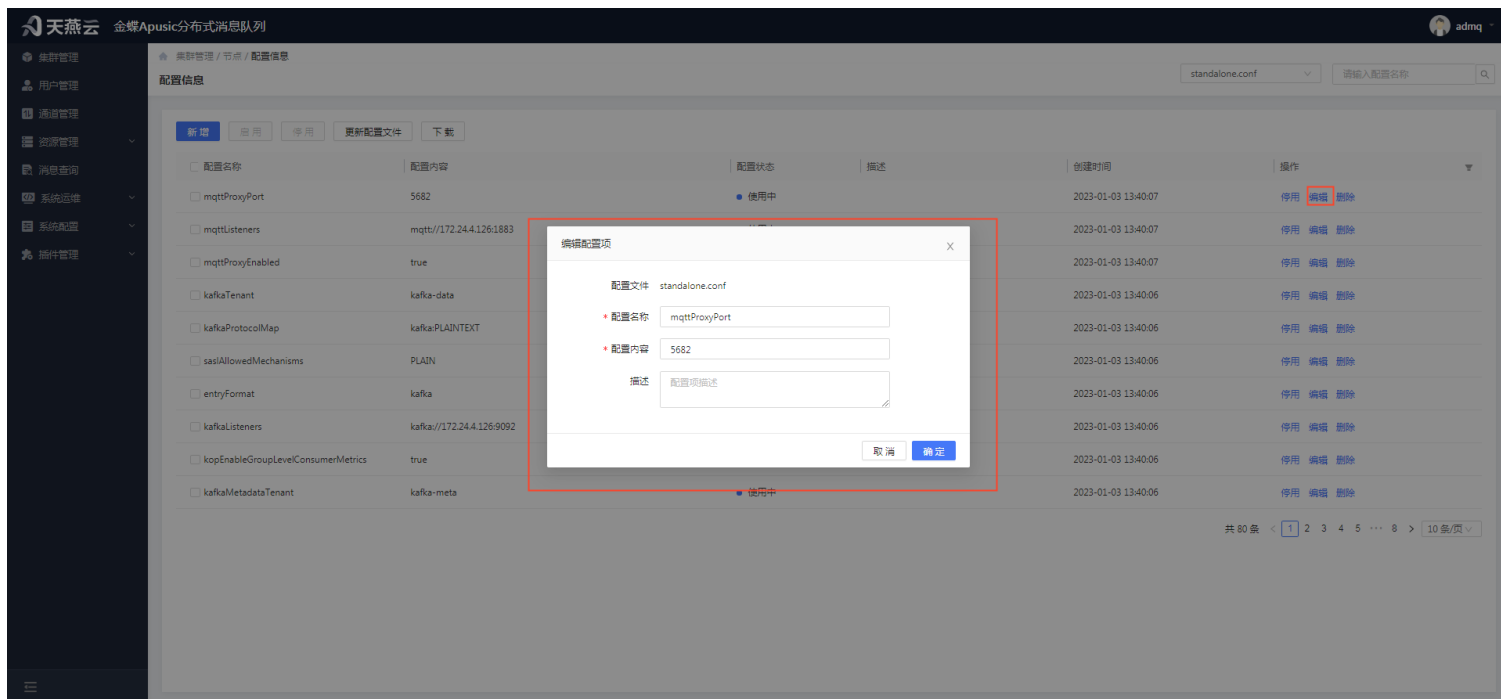
在节点“配置信息”页面，点击【新增】按钮，可新增配置信息。



- 配置文件：配置项的配置文件名称，根据选择的节点进行显示，不可修改
- 配置名称：配置项的名称，自定义，唯一标识
- 配置内容：配置项的值，自定义
- 描述：配置项的描述信息，自定义
- 备注：新增配置信息需要更新配置文件才会更新到节点物理文件中，且新增的配置信息，需要重启节点，参数才会生效。

5.3.1.16.2 修改配置参数

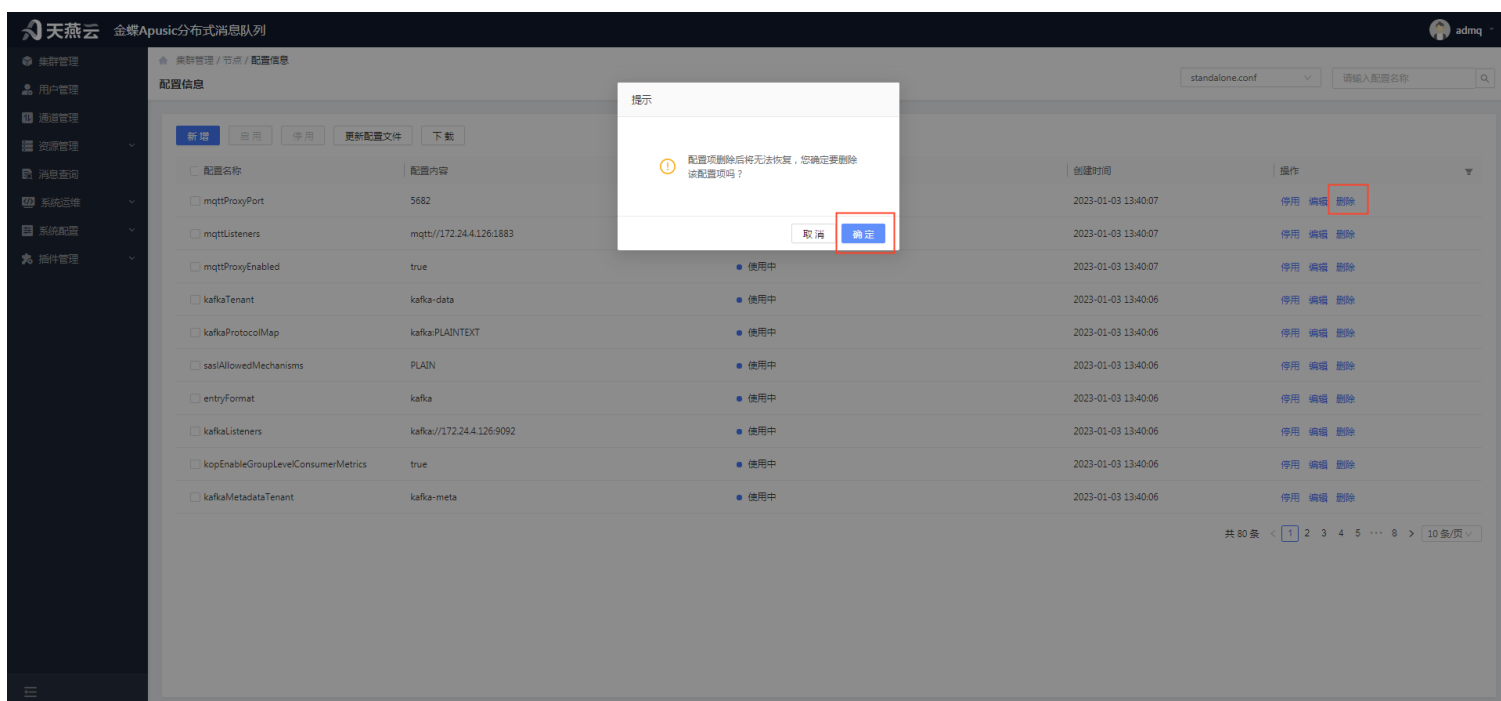
在配置信息页面，选择需要修改的配置项，点击【编辑】按钮，可修改配置信息。



- 配置文件：配置项的配置文件名称，根据选择的节点进行显示，不可修改
- 配置名称：配置项的名称，自定义，唯一标识
- 配置内容：配置项的值，自定义
- 描述：配置项的描述信息，自定义
- 备注：修改配置信息需要更新配置文件才会更新到节点物理文件中，且修改的配置信息，需要重启节点，参数才会生效

5.3.1.16.3 删除配置参数

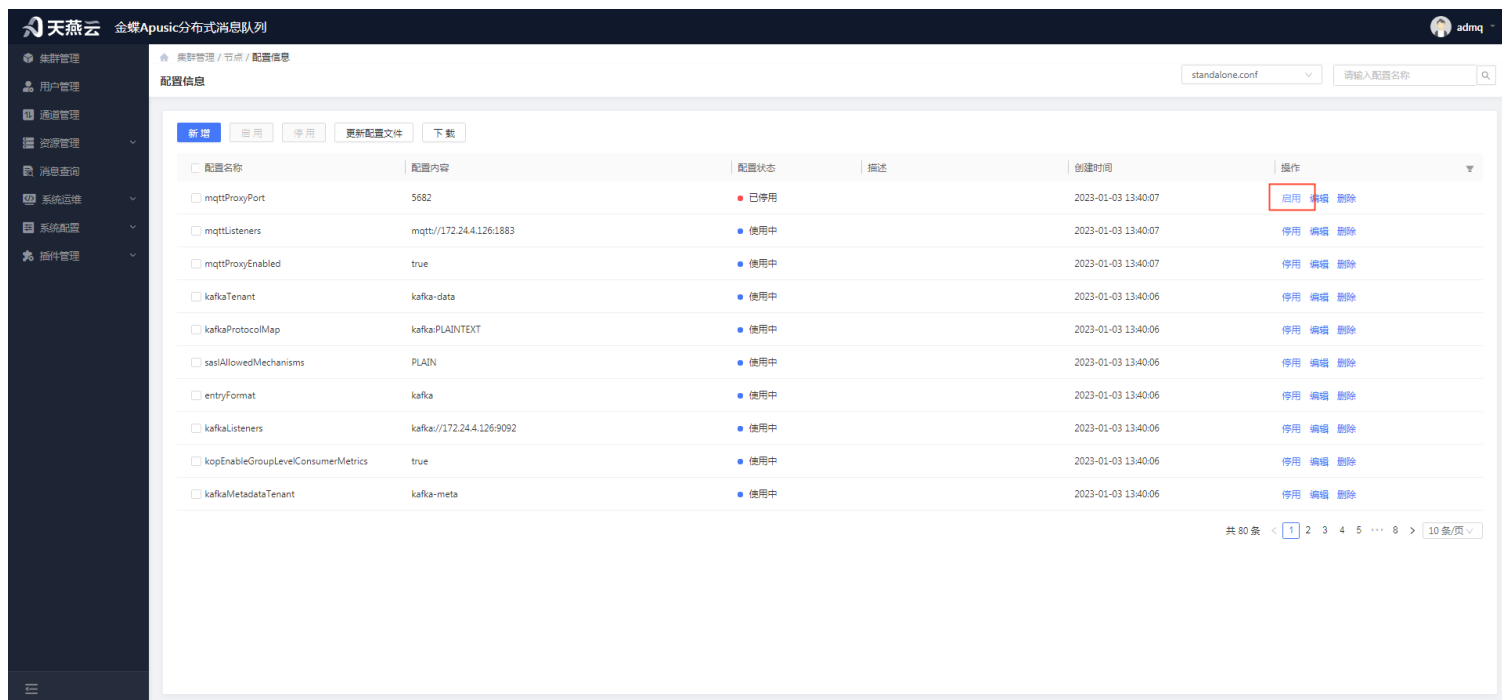
在配置信息页面，选择需要删除的配置项，点击【删除】按钮，在弹出的确认删除弹框中，点击【确定】按钮，可删除配置信息。



- 备注：删除配置信息需要更新配置文件才会更新到节点物理文件中，且删除的配置信息，需要重启节点，参数才会生效；删除配置项需谨慎，避免因删除了重要的配置项而出现集群服务起不来或服务不可用。

5.3.1.16.4 启用配置参数

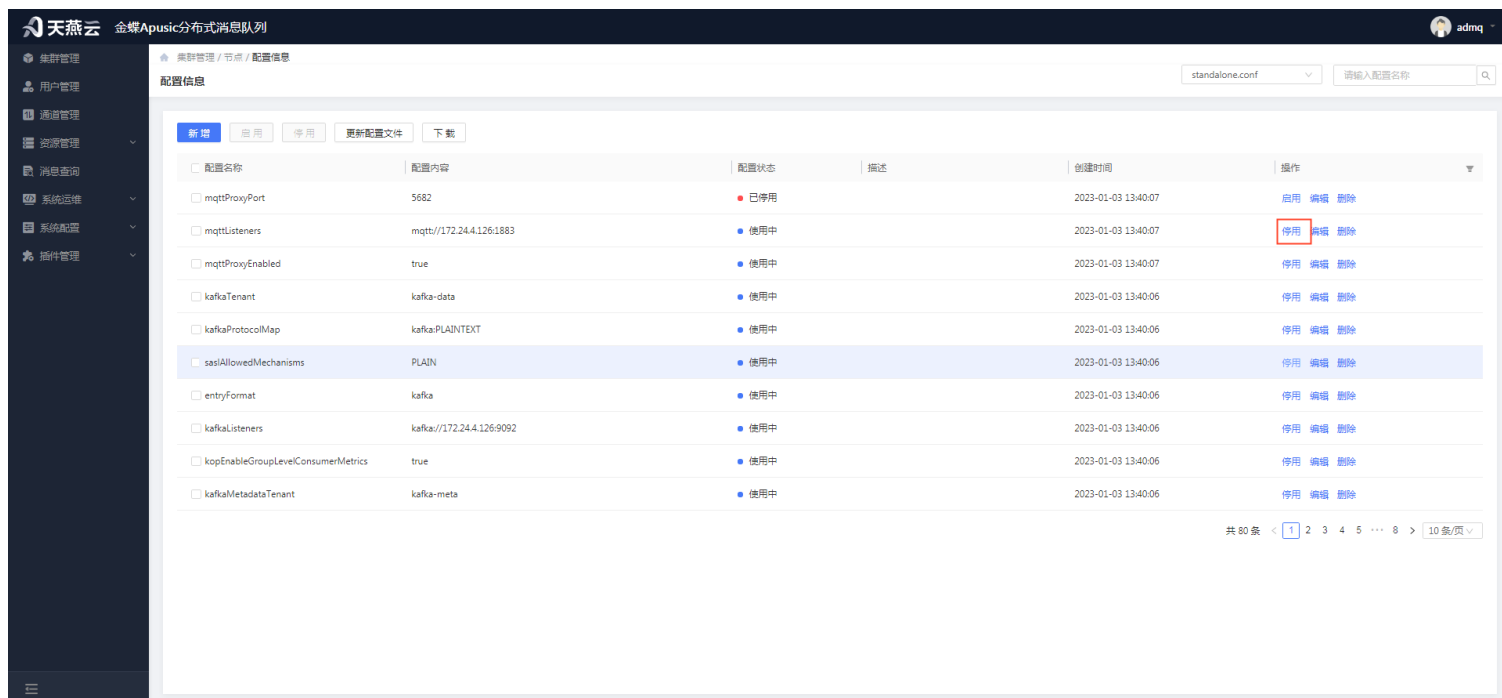
在配置信息页面，选择需要启用的配置项，点击【启动】按钮或勾选配置项后点击上方的【启用】按钮，可启用配置信息。



- 备注：启用配置信息需要更新配置文件才会更新到节点物理文件中，且启用的配置信息，需要重启节点，参数才会生效。

5.3.1.16.5 停用配置参数

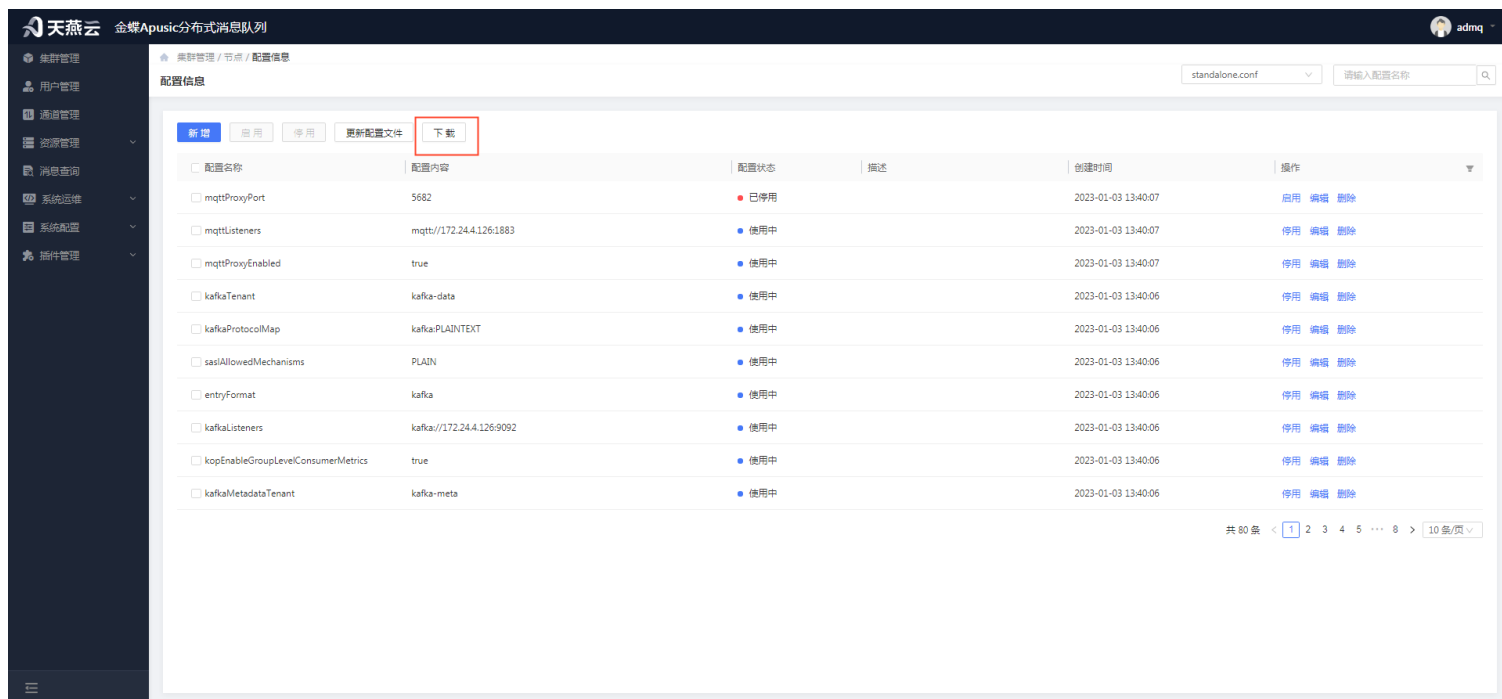
在配置信息页面，选择需要停用的配置项，点击【停用】按钮或勾选配置项后点击上方的【停用】按钮，可停用配置信息。



- 备注：停用配置信息需要更新配置文件才会更新到节点物理文件中，且停用的配置信息，需要重启节点，参数才会生效；停用配置项需谨慎，避免因停用了重要的配置项而出现集群服务起不来或服务不可用。

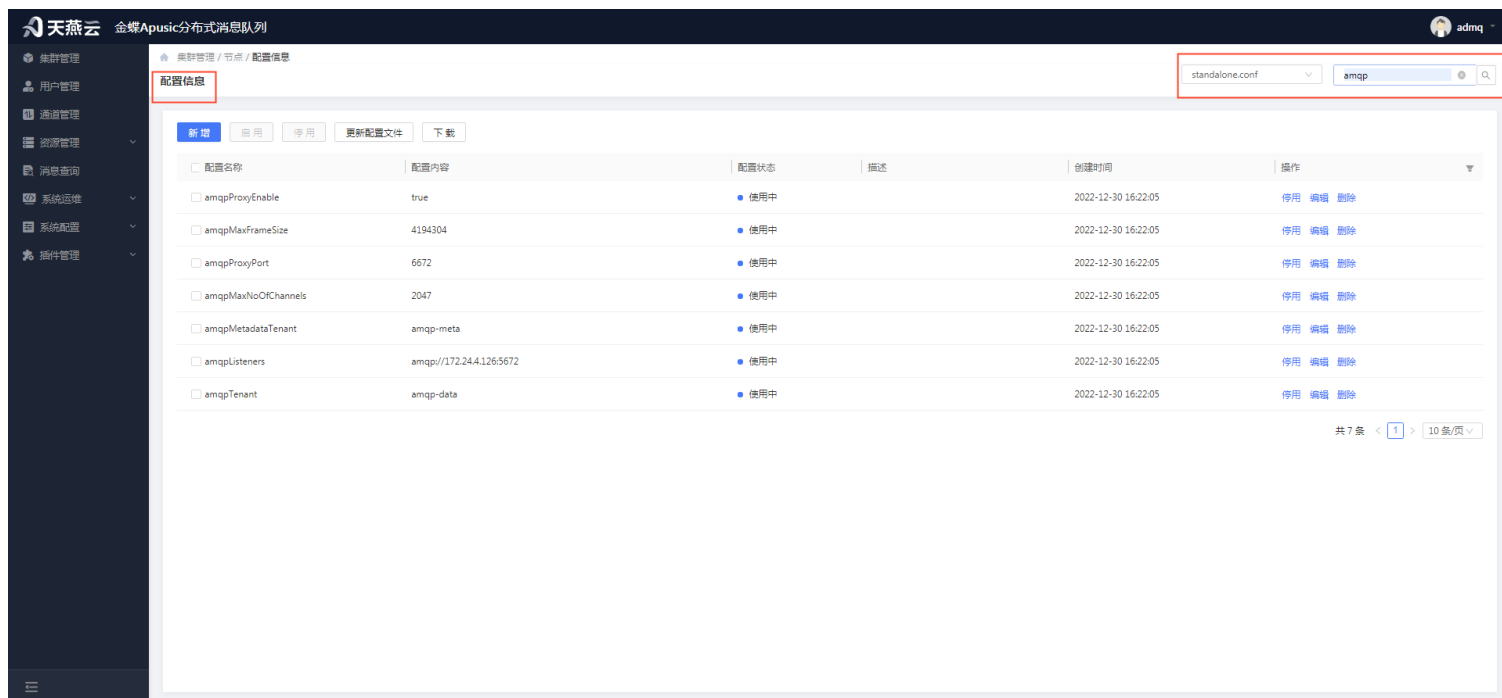
5.3.1.16.6 下载配置文件

在配置信息页面，点击【下载】按钮，可下载选择节点的配置文件。



5.3.1.16.7 查询配置项

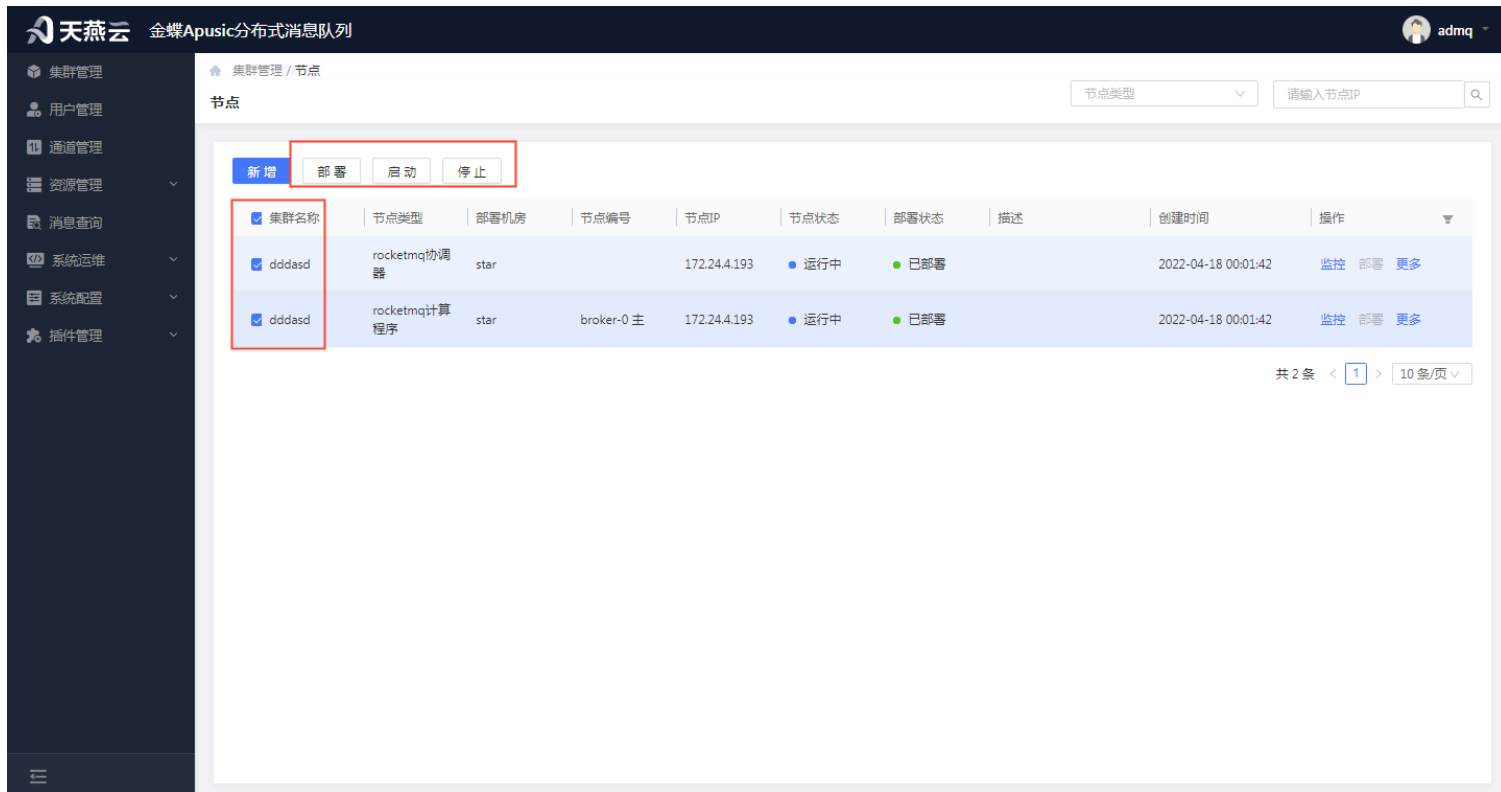
在配置信息页面，模糊或精确输入配置名称，即可查询到所需要的配置信息。



- 备注：查询的配置文件是在集群管理中选择的节点对应的配置文件，不可修改配置文件。

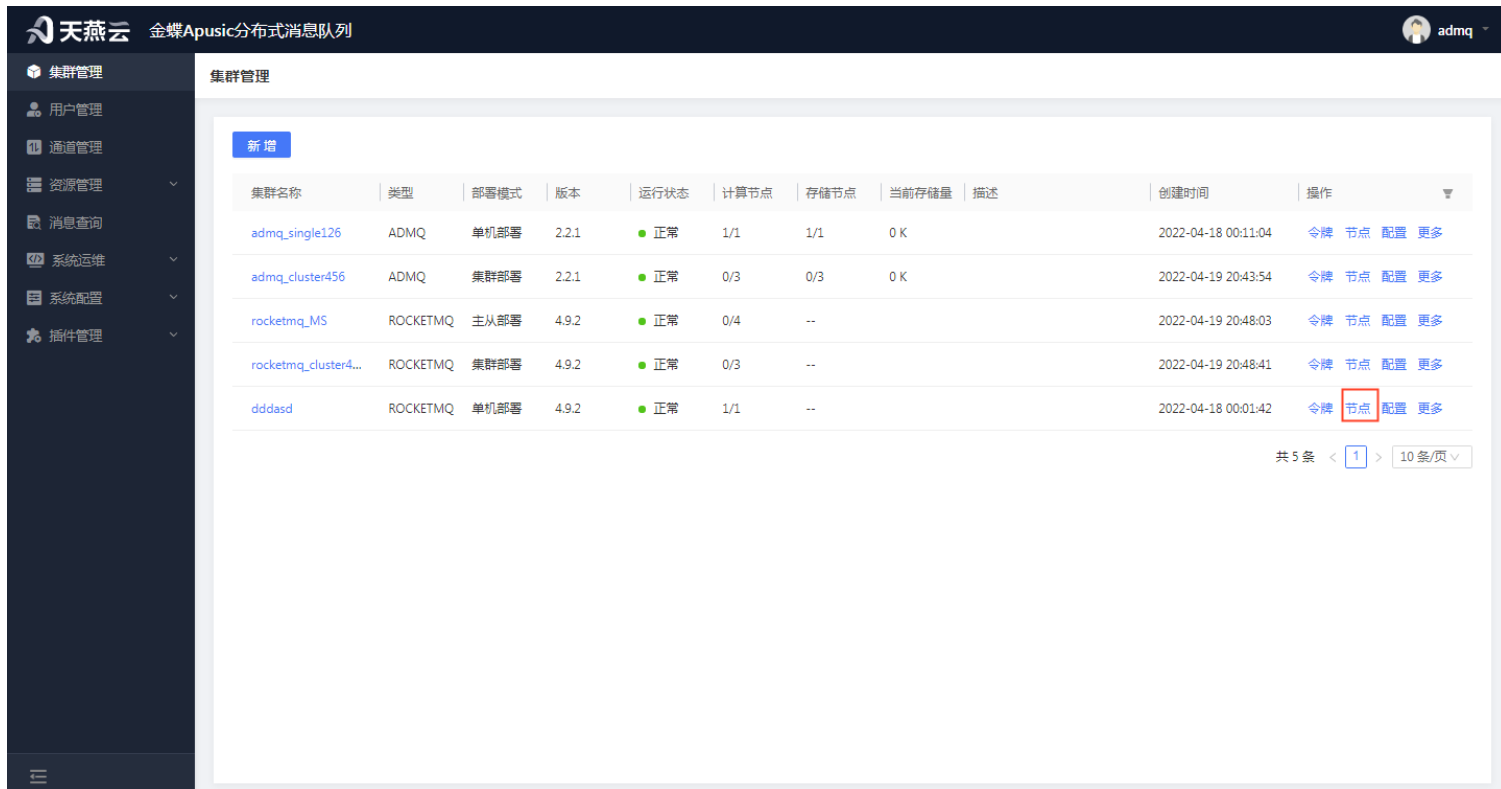
5.3.1.17 一键操作

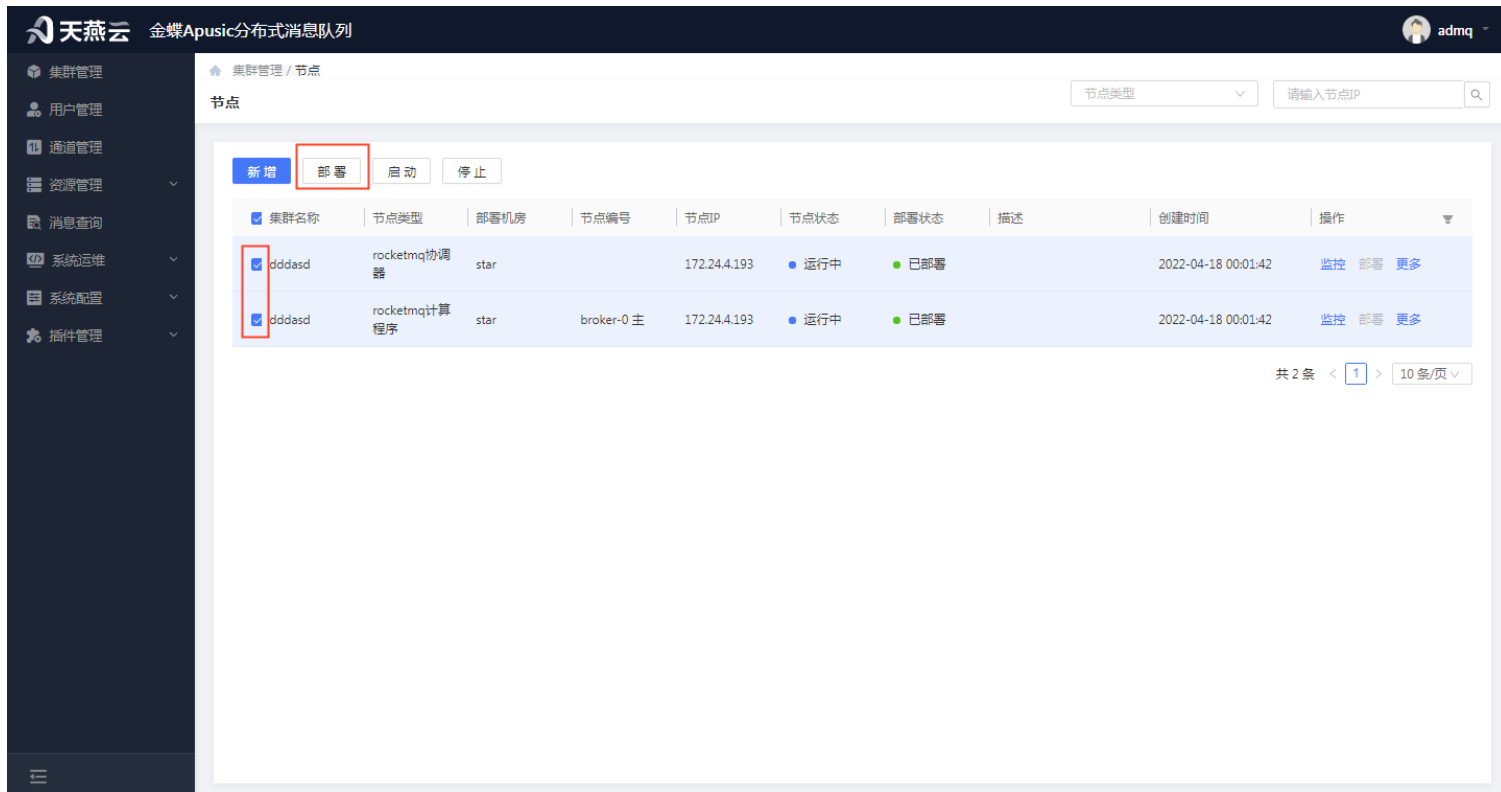
对于有多个节点的集群，可同时对多个节点进行操作，包括：一键部署、一键启动、一键停止。



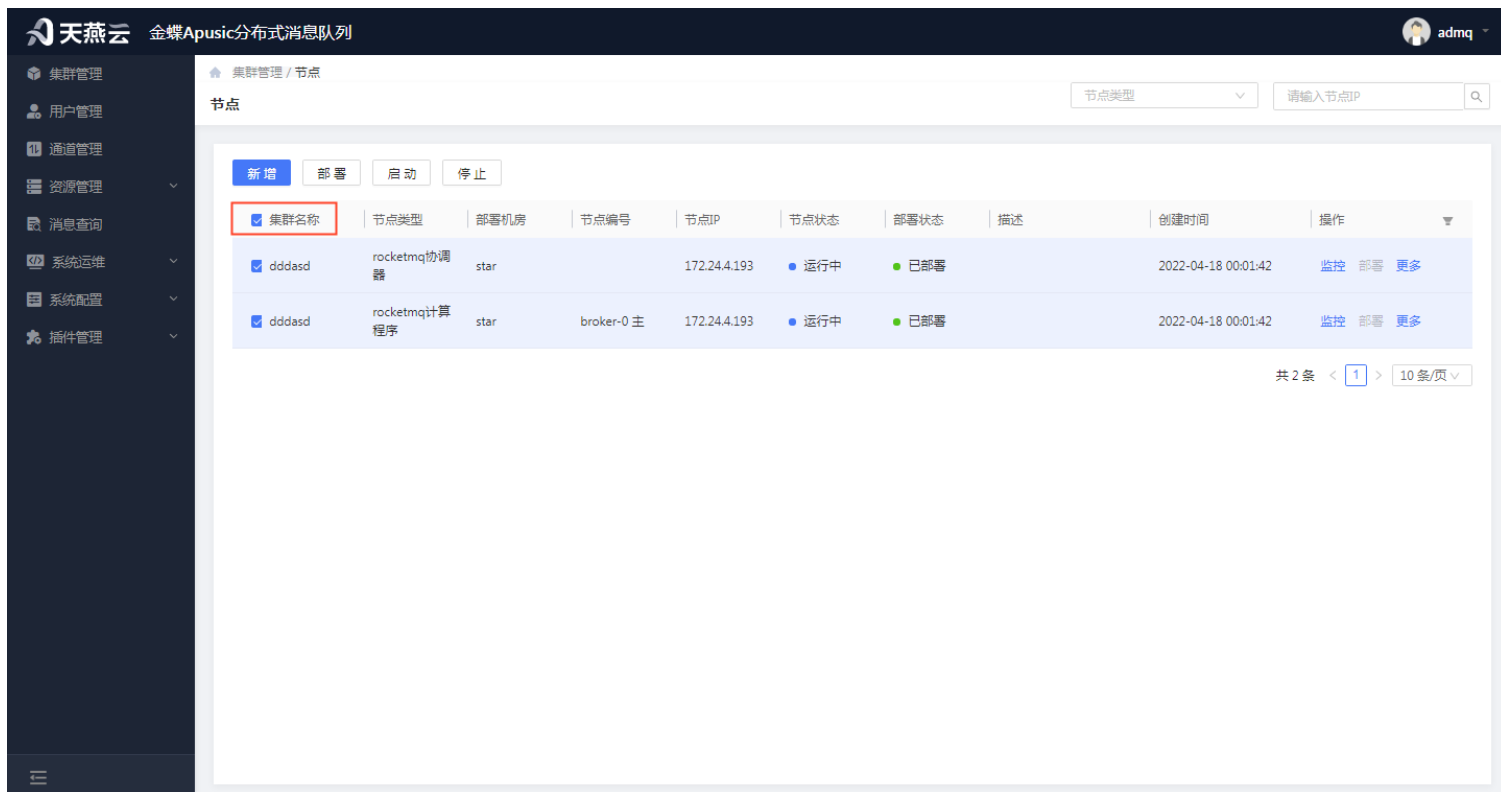
5.3.1.17.1 一键部署

进入【集群管理】，在集群管理页面，选择需要部署节点的集群信息，点击【节点】按钮，进入部署节点页面，选中需要操作的节点前的白色方框，点击左上角的【部署】按钮，即可一键对多个节点进行部署。



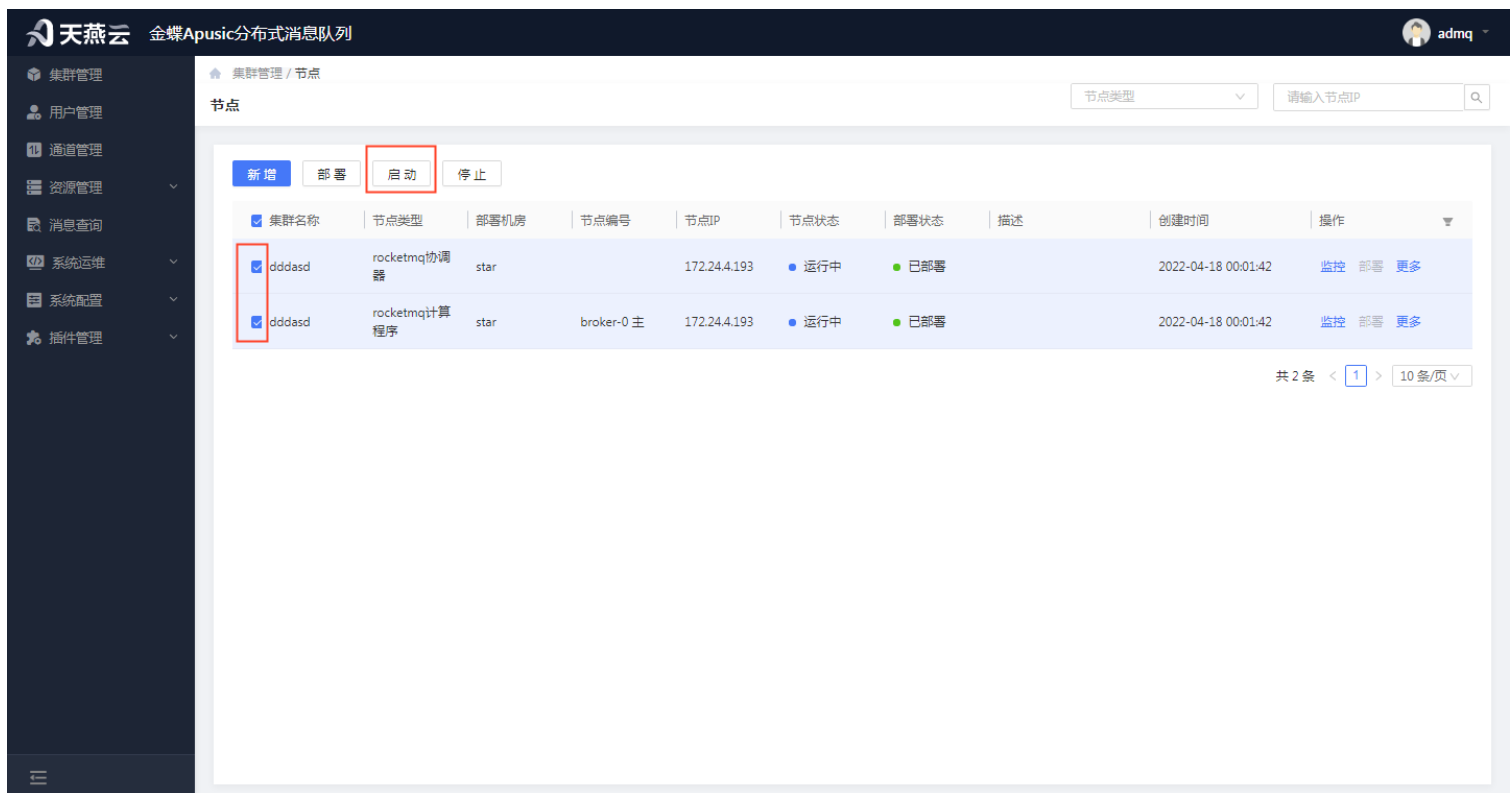
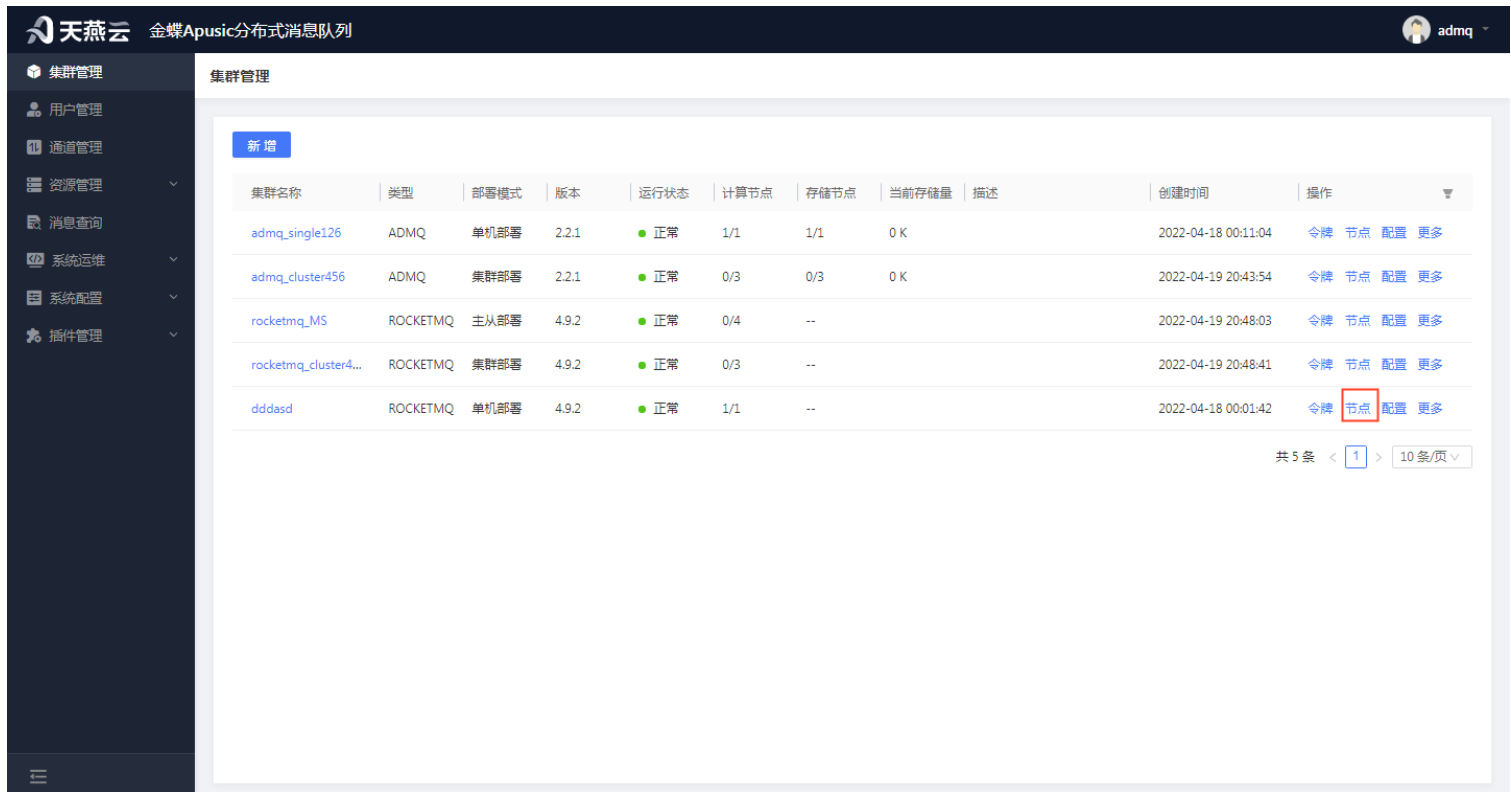


- 备注：若想一次性选择集群的所有节点，可直接勾选集群名称前的白色方框



5.3.1.17.2 一键启动

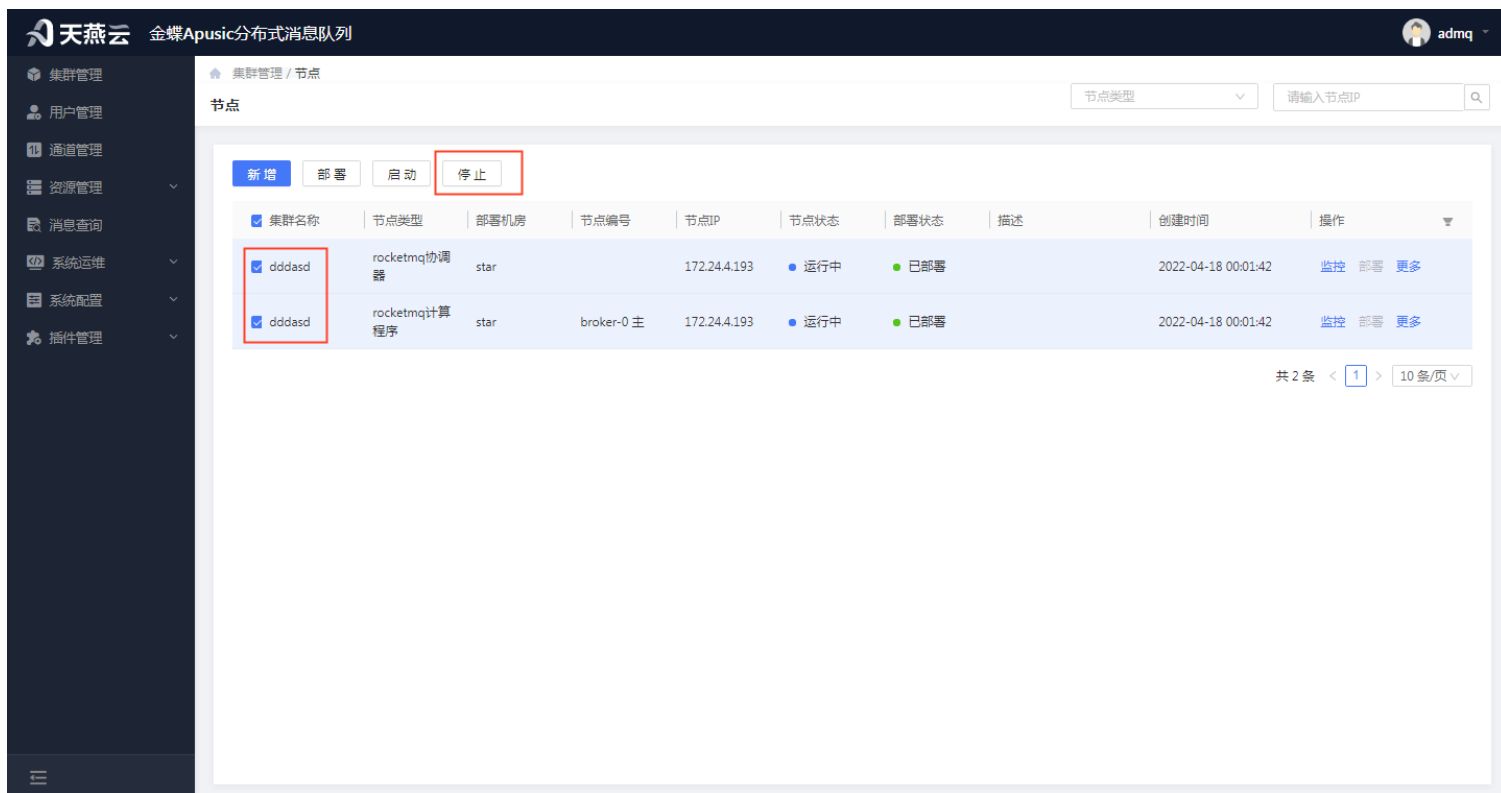
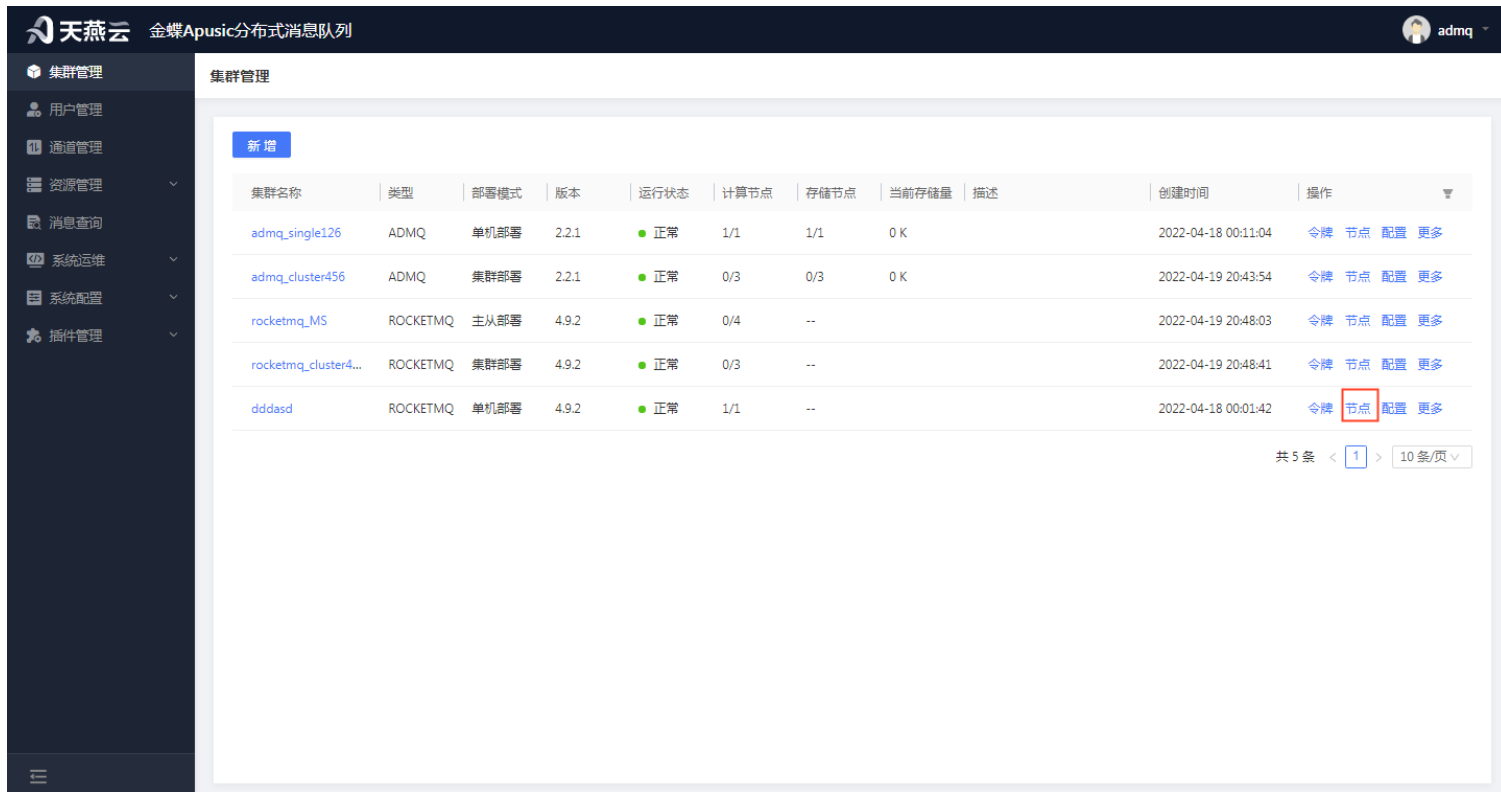
进入【集群管理】，在集群管理页面，选择需要启动节点的集群信息，点击【节点】按钮，进入部署节点页面，选中需要操作的节点前的白色方框，点击左上角的【启动】按钮，即可一键启动多个节点。



- 备注：若想一次性选择集群的所有节点，可直接勾选集群名称前的白色方框。

5.3.1.17.3 一键停止

进入【集群管理】，在集群管理页面，选择需要停止节点的集群信息，点击【节点】按钮，进入部署节点页面，选中需要操作的节点前的白色方框，点击左上角的【停止】按钮，即可一键停止多个节点。



- 备注：若想一次性选择集群的所有节点，可直接勾选集群名称前的白色方框。

5.3.1.18 节点资源监控

- 备注：
 - (1) 目前监控支持原生RocketMQ、原生ADMQ、kafka插件，rabbitmq插件、rocketmq插件暂不支持。
 - (2) 若想使用节点资源监控，必须使用AUC登录界面跳转到ADMQ管理控制台后再创建集群。

(3) 节点监控需结合AUC和AMP使用，且只能通过AUC登录跳转到ADMQ界面，具体过程如下：

在admq-manager的config文件夹下的application.properties配置文件中添加配置项：（如果使用AUC登录，则需要把配置项security.same-site.enabled设置成false）

amp.url=<http://172.24.4.170:9002>

auc.url=<http://172.24.4.170:9000>

重新启动admq-manager：

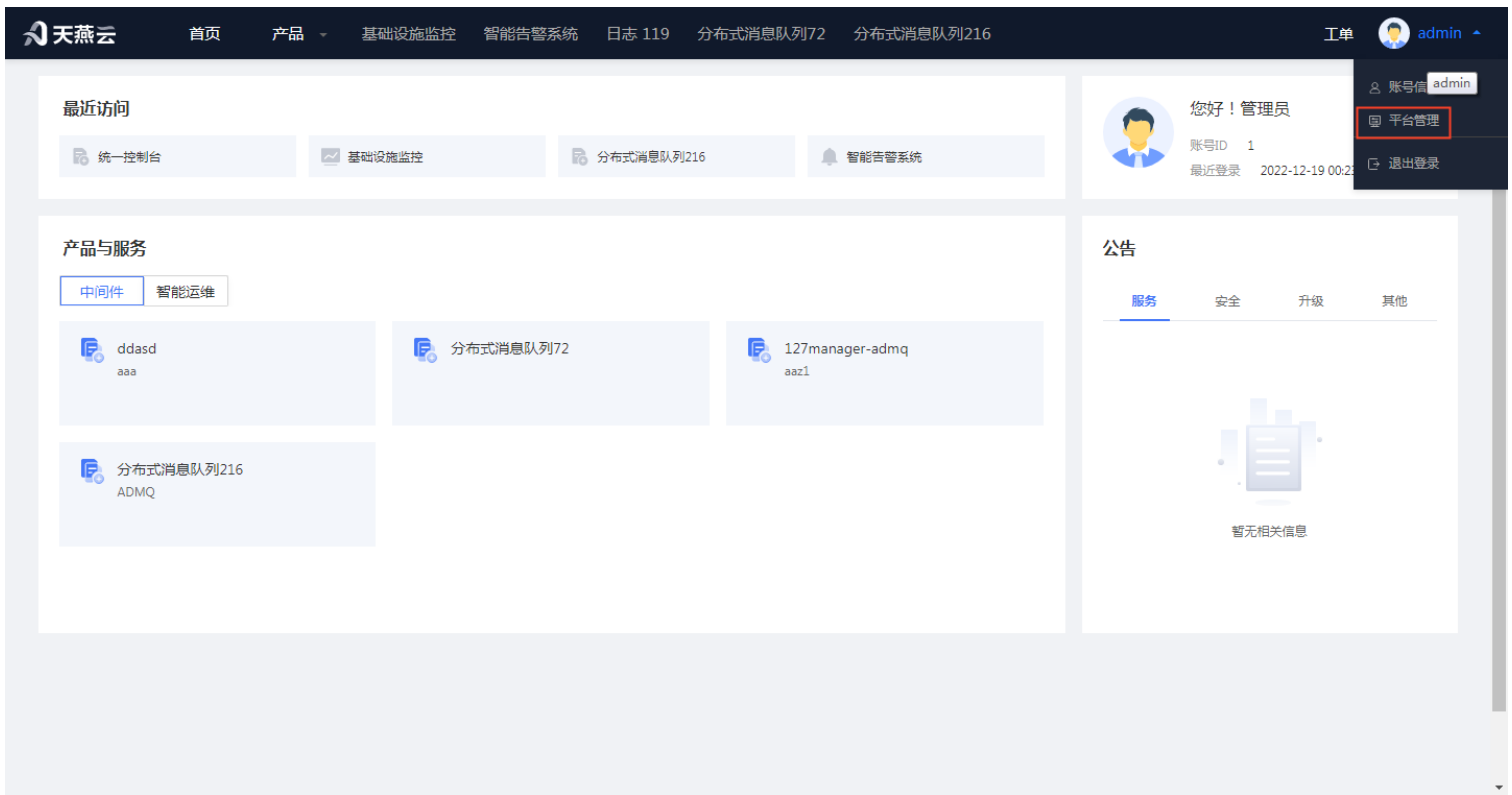
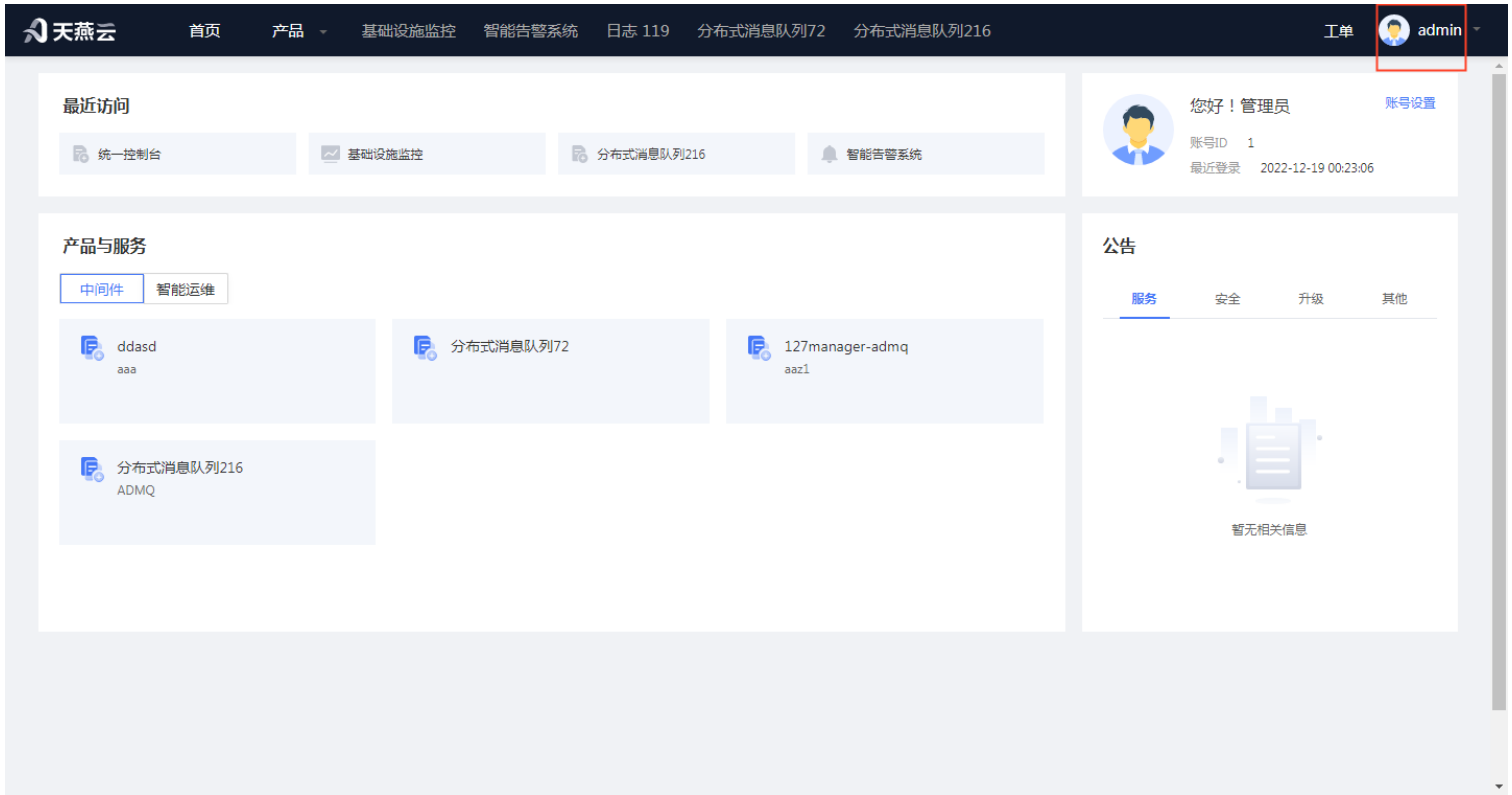
bin/admq-manager stop

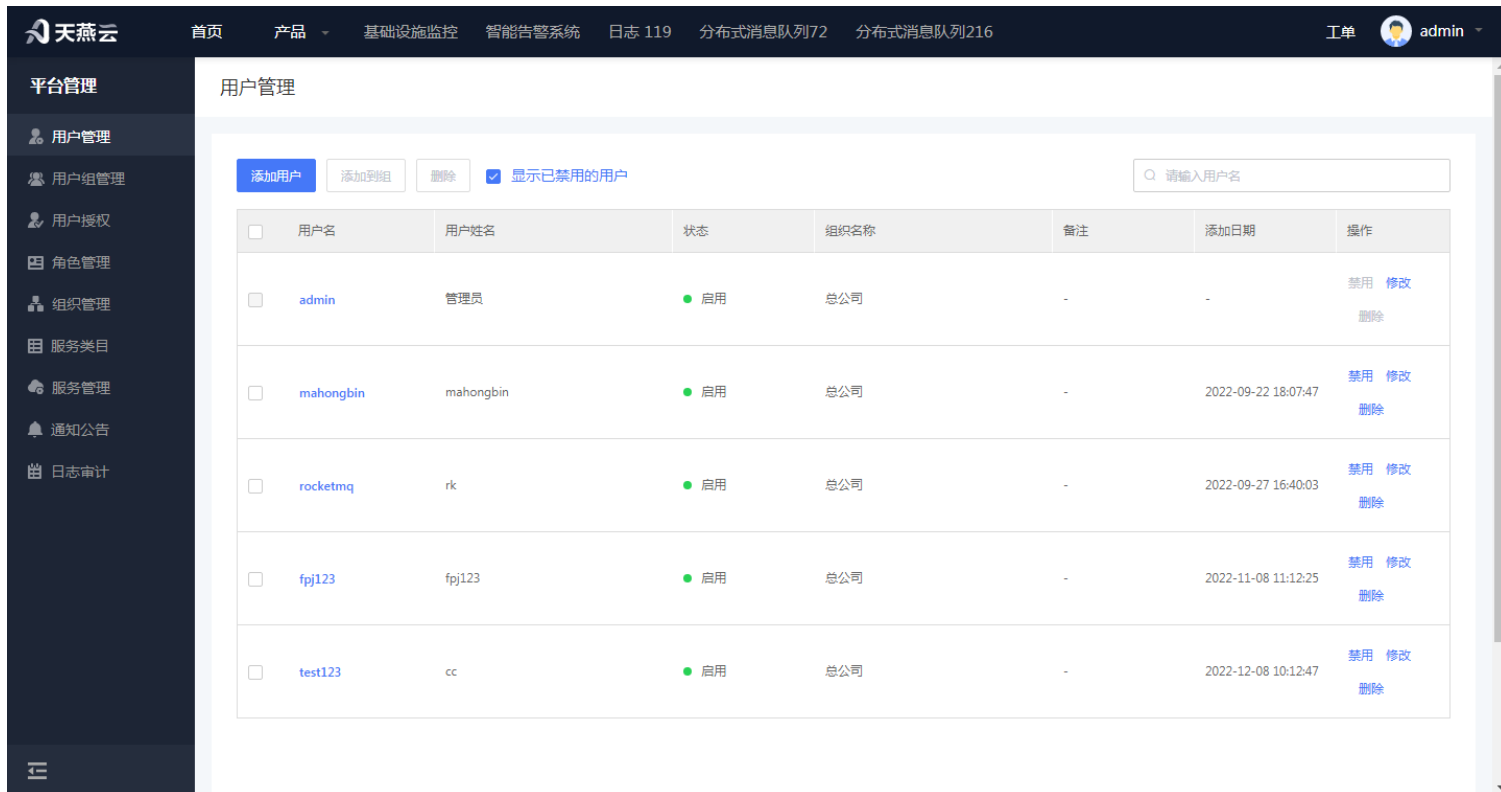
bin/admq-manager start

访问AUC界面：<http://172.24.4.170:9000>

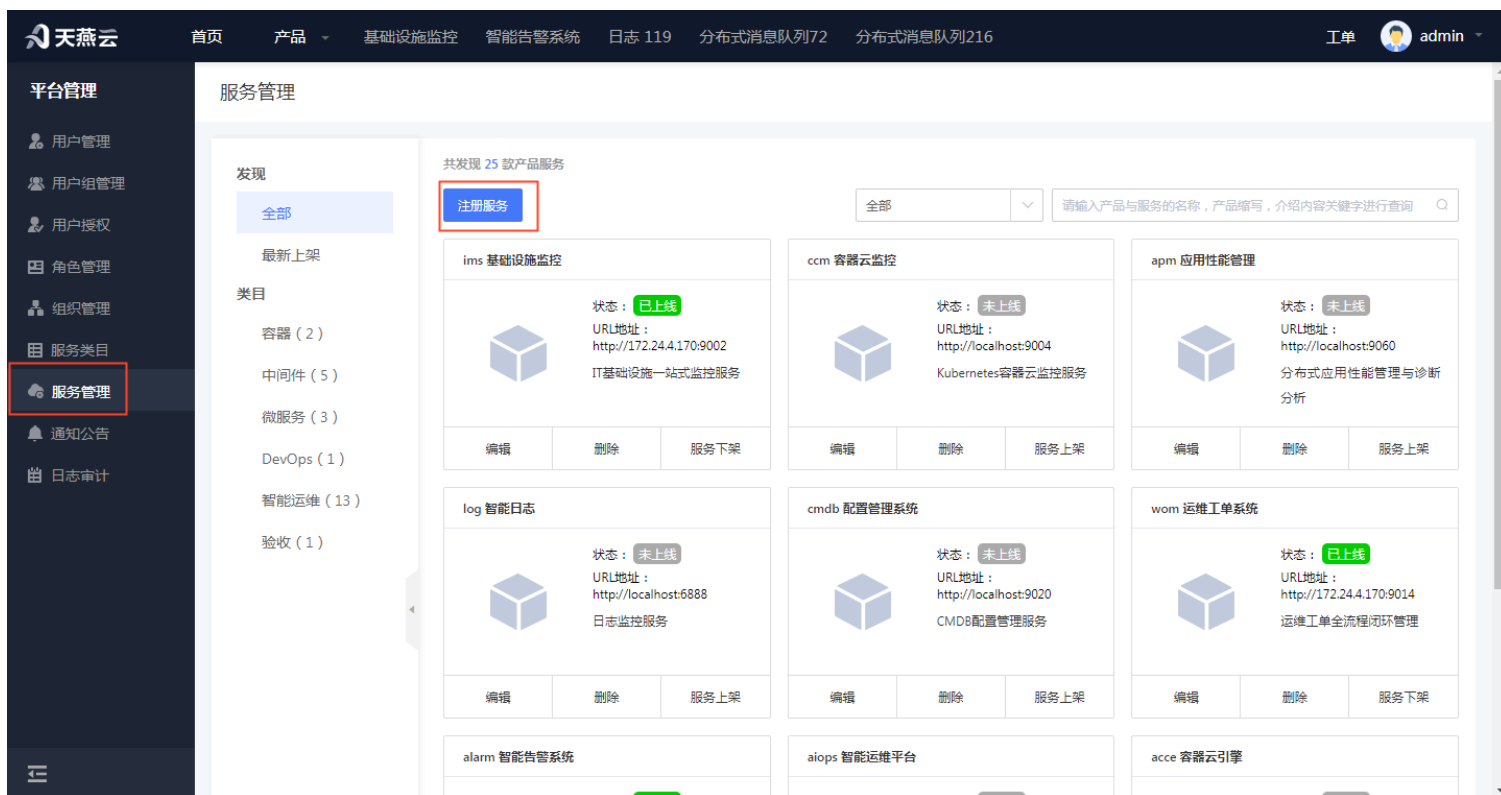


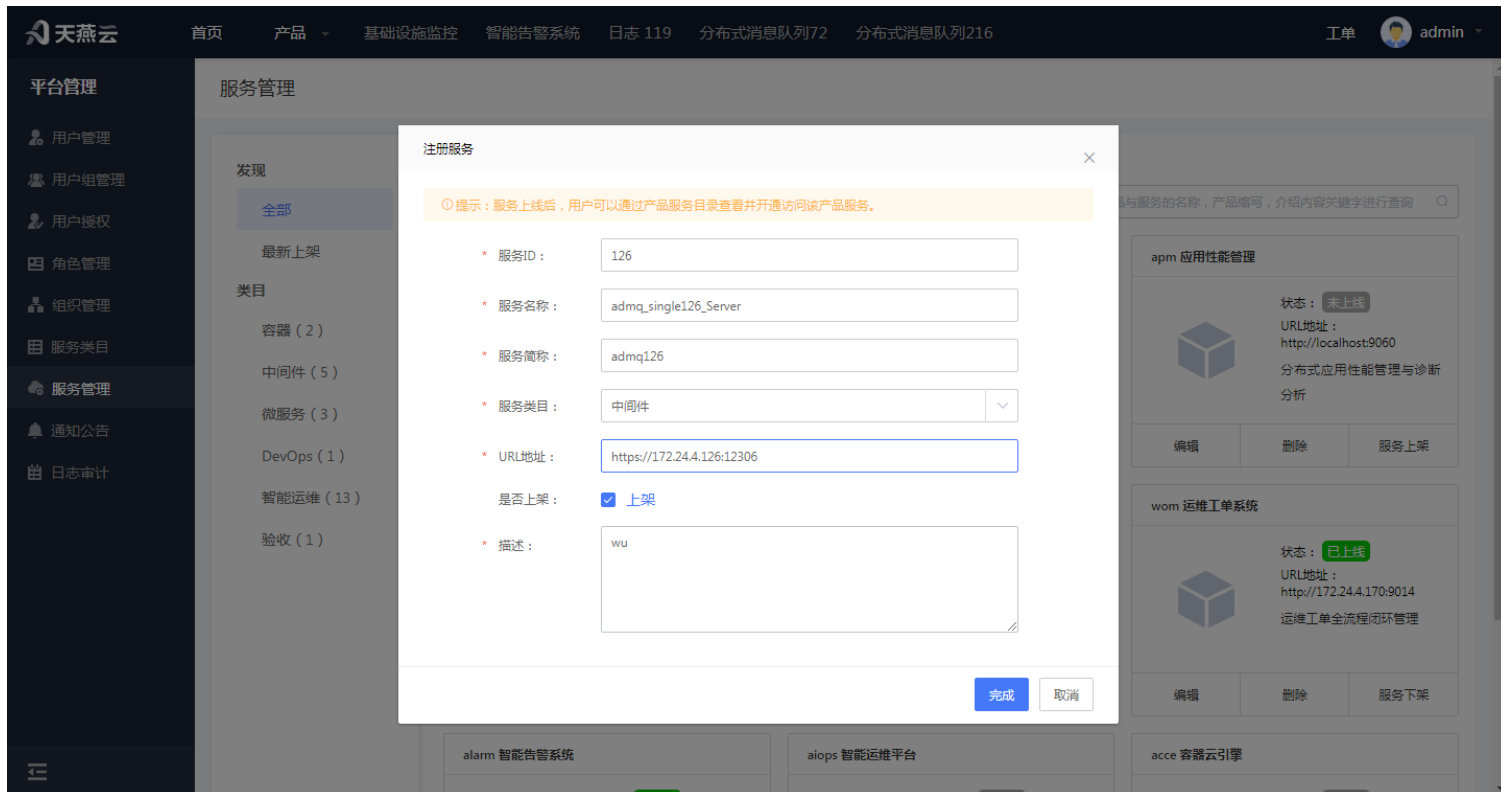
选中AUC首页右上角“admin”，会弹出弹窗，选中弹窗中的【平台管理】按钮，跳转到平台管理页面





进入【服务管理】，在服务管理页面点击【注册服务】按钮，进行注册服务（该步骤是可选的，如果已注册分布式消息系统，则直接修改注册服务的URL地址即可）：





- 服务ID：产品服务ID的数值，自定义
- 服务名称：产品服务名称，自定义
- 服务简称：产品服务的简称代码
- 服务类目：选择中间件
- URL地址：服务的URL访问的地址
- 是否上架：上架（若选择不上架，服务注册后需上架才能使用）
- 描述：服务的详细信息介绍

选中产品，会出现弹窗，在弹窗中点击已注册的服务，会跳转到ADMQ管控台首页。

天燕云 首页 **产品** 基础设施监控 智能告警系统 日志 119 分布式消息队列72 分布式消息队列216 工单 admin

平台管理

- 用户管理
- 用户组管理
- 用户授权
- 角色管理
- 组织管理
- 服务类目
- 服务管理
- 通知公告
- 日志审计

全部 注册服务 全部 请输入产品与服务的名称，产品缩写，介绍内容关键字进行查询

最新上架

类目

- 容器 (2)
- 中间件 (6)**
- 微服务 (3)
- DevOps (1)
- 智能运维 (13)
- 验收 (1)

<p>aump 中间件管理平台</p> <p>状态: 未上线</p> <p>URL地址: http://localhost:9032</p> <p>支持开源中间件和Apusic信创中间件的全生命周期管理</p> <p>编辑 删除 服务上架</p>	<p>admqmanager ddasd</p> <p>状态: 已上线</p> <p>URL地址: http://172.24.6.42:3000</p> <p>aaa</p> <p>编辑 删除 服务下架</p>	<p>admq72 分布式消息队列72</p> <p>状态: 已上线</p> <p>URL地址: http://172.24.5.72:3000</p> <p>编辑 删除 服务下架</p>
<p>aaa 127manager-admq</p> <p>状态: 已上线</p> <p>URL地址: http://172.24.4.127:12305</p> <p>aaa1</p> <p>编辑 删除 服务下架</p>	<p>ADMQ 分布式消息队列216</p> <p>状态: 已上线</p> <p>URL地址: https://172.24.4.216:12306</p> <p>ADMQ</p> <p>编辑 删除 服务下架</p>	<p>adm126 admq_single126_Server</p> <p>状态: 已上线</p> <p>URL地址: https://172.24.4.126:12306</p> <p>wu</p> <p>编辑 删除 服务下架</p>

1 1 前往 1 页

天燕云 首页 产品 基础设施监控 智能告警系统 日志 119 分布式消息队列72 分布式消息队列216 工单 admin

中间件

智能运维

中间件
提供云原生信创中间件，支撑企业级分布式应用程序运行。

ddasd aaa	分布式消息队列72 ★	127manager-admq aaa1
分布式消息队列216 ★ ADMQ	admq_single126_Server ☆ wu	

验收 (1)

<p>状态: 已上线</p> <p>URL地址: http://172.24.4.127:12305</p> <p>aaa1</p> <p>编辑 删除 服务下架</p>	<p>状态: 已上线</p> <p>URL地址: https://172.24.4.216:12306</p> <p>ADMQ</p> <p>编辑 删除 服务下架</p>	<p>状态: 已上线</p> <p>URL地址: https://172.24.4.126:12306</p> <p>wu</p> <p>编辑 删除 服务下架</p>
---	--	--

1 1 前往 1 页

天燕云 首页 产品 基础设施监控 智能告警系统 日志 119 分布式消息队列72 分布式消息队列216 工单 admin

分布式消息队列

集群管理

新增

集群名称	类型	部署模式	版本	运行状态	计算节点	存储节点	当前存储量	描述	创建时间	操作
admq_cluster456	ADMQ	集群部署	2.2.1	● 正常	0/3	0/3	0 K		2022-04-19 20:43:54	令牌 节点 配置 更多
rocketmq_MS	ROCKETMQ	主从部署	4.9.2	● 正常	0/4	--			2022-04-19 20:48:03	令牌 节点 配置 更多
rocketmq_cluste...	ROCKETMQ	集群部署	4.9.2	● 正常	0/3	--			2022-04-19 20:48:41	令牌 节点 配置 更多
dddasd	ROCKETMQ	单机部署	4.9.2	● 异常	0/1	--			2022-04-18 00:01:42	令牌 节点 配置 更多

共 4 条 < 1 > 10 条/页

进入【集群管理】，在集群管理页面，新增ADMQ集群和RocketMQ集群并启动，详情见3.1集群管理。选择需要监控节点的集群信息，点击【节点】按钮，进入部署节点页面，点击【监控】按钮，可弹出节点监控页面

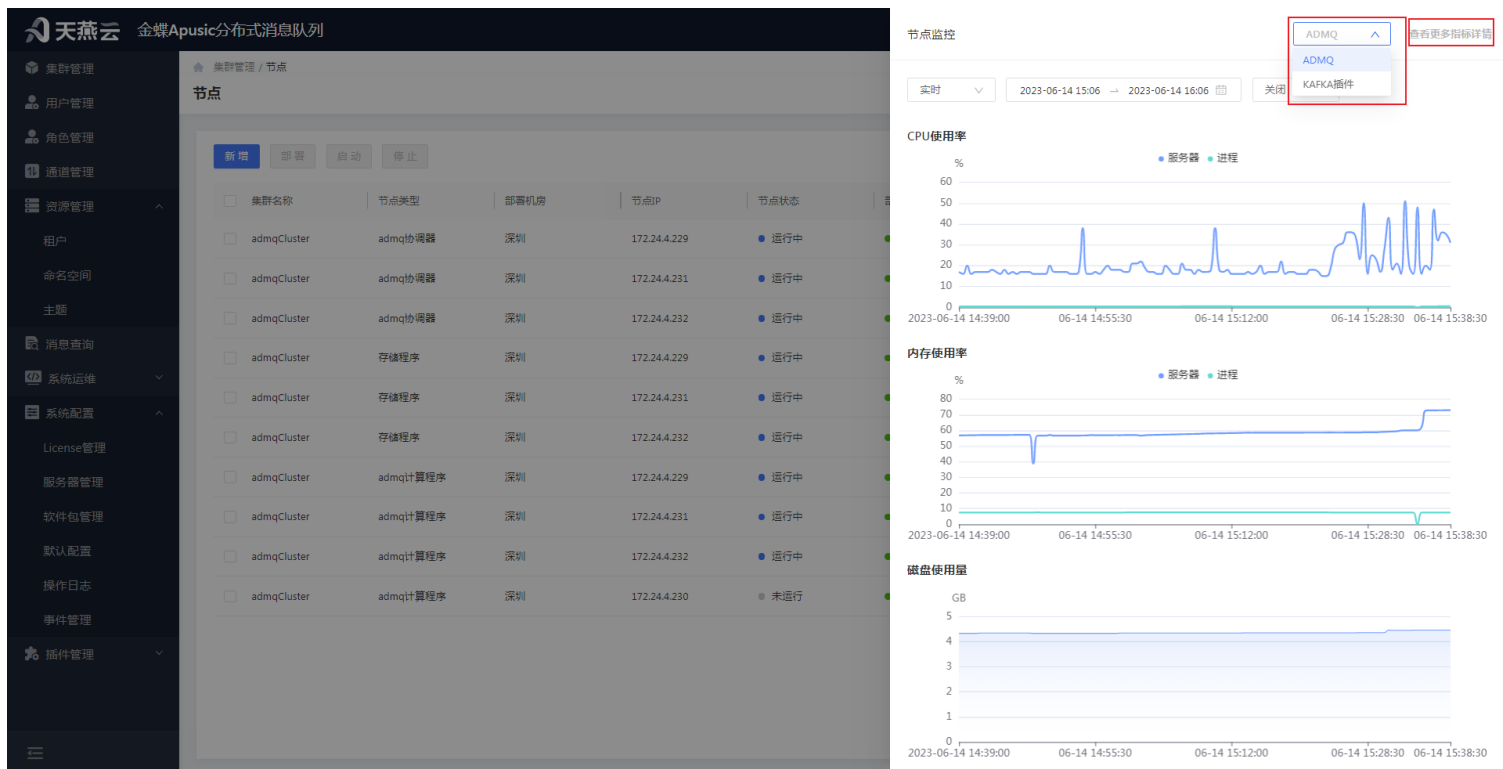
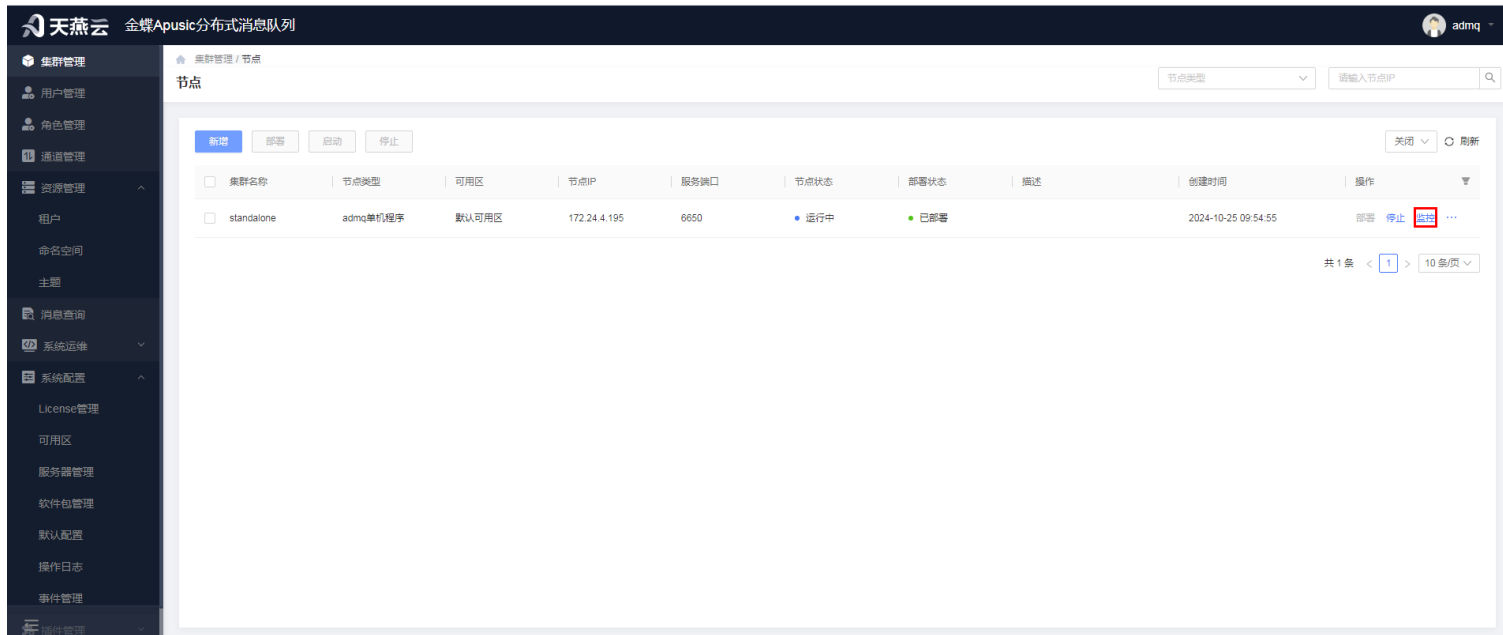
天燕云 金蝶Apusic分布式消息队列 admin

集群管理

新增

集群名称	类型	部署模式	版本	运行状态	计算节点	存储节点	当前存储量	描述	创建时间	操作
admq_single126	ADMQ	单机部署	2.2.1	● 正常	1/1	1/1	0 K		2022-04-18 00:11:04	令牌 节点 配置 更多
admq_cluster456	ADMQ	集群部署	2.2.1	● 正常	0/3	0/3	0 K		2022-04-19 20:43:54	令牌 节点 配置 更多
rocketmq_MS	ROCKETMQ	主从部署	4.9.2	● 正常	0/4	--			2022-04-19 20:48:03	令牌 节点 配置 更多
rocketmq_cluster4...	ROCKETMQ	集群部署	4.9.2	● 正常	0/3	--			2022-04-19 20:48:41	令牌 节点 配置 更多
dddasd	ROCKETMQ	单机部署	4.9.2	● 正常	1/1	--			2022-04-18 00:01:42	令牌 节点 配置 更多

共 5 条 < 1 > 10 条/页

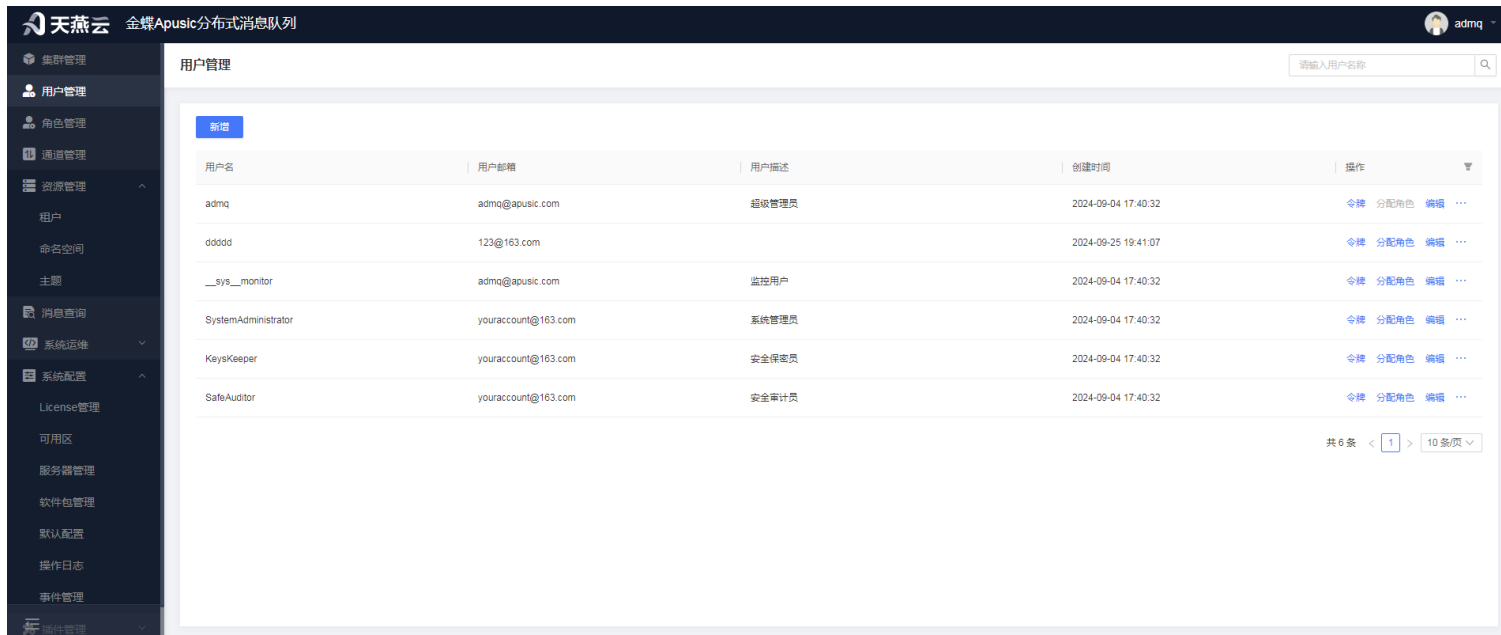


• 备注：在节点监控页面中点击 右上角“查看更多指标详情”，可跳转到amp监控页面，该部分需要结合AUC和AMP进行使用，具体详见第10章运维监控。

5.4 用户

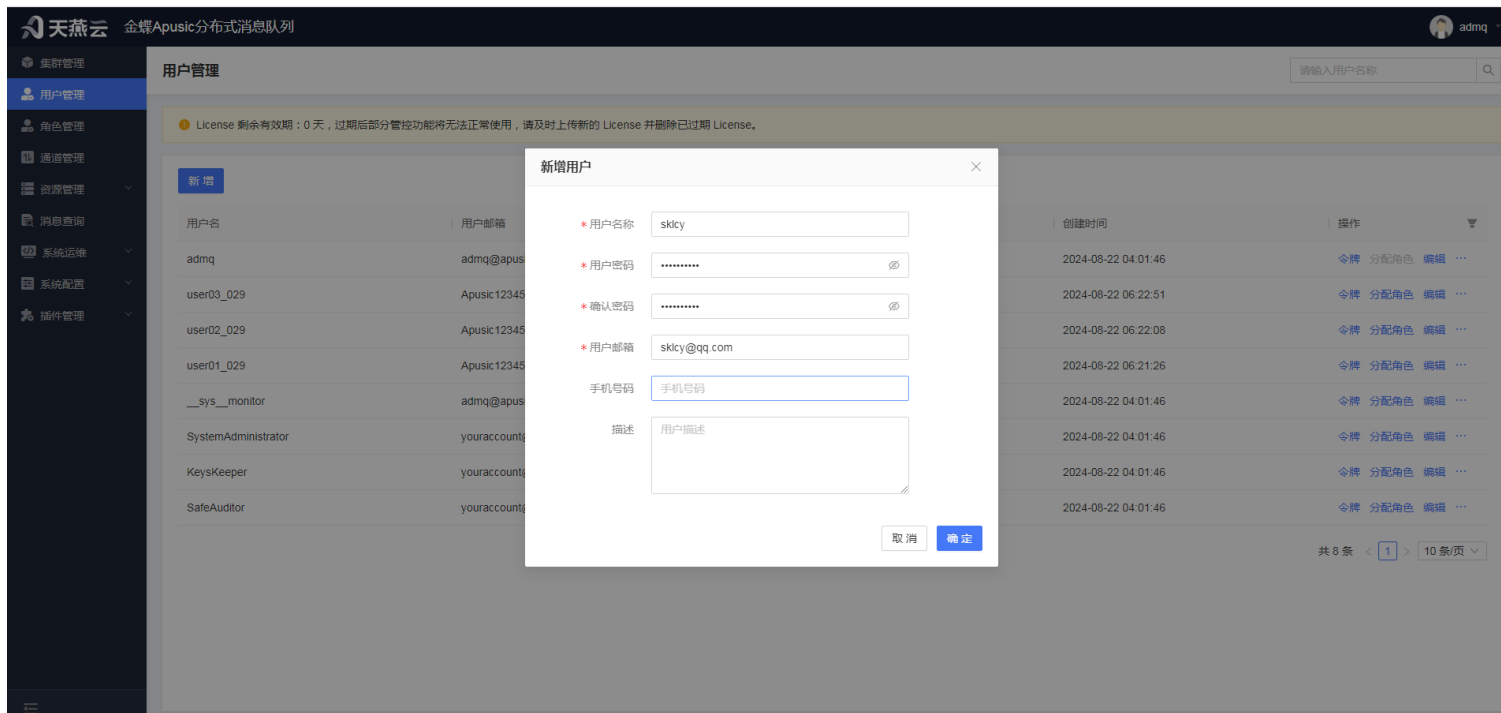
5.4.1 用户管理

用户管理用于新增用户信息、编辑用户信息、删除用户信息、修改用户密码、查看令牌



5.4.1.1 新增用户

进入【用户管理】，在用户管理页面点击左上角【新增】按钮，可新增用户信息。

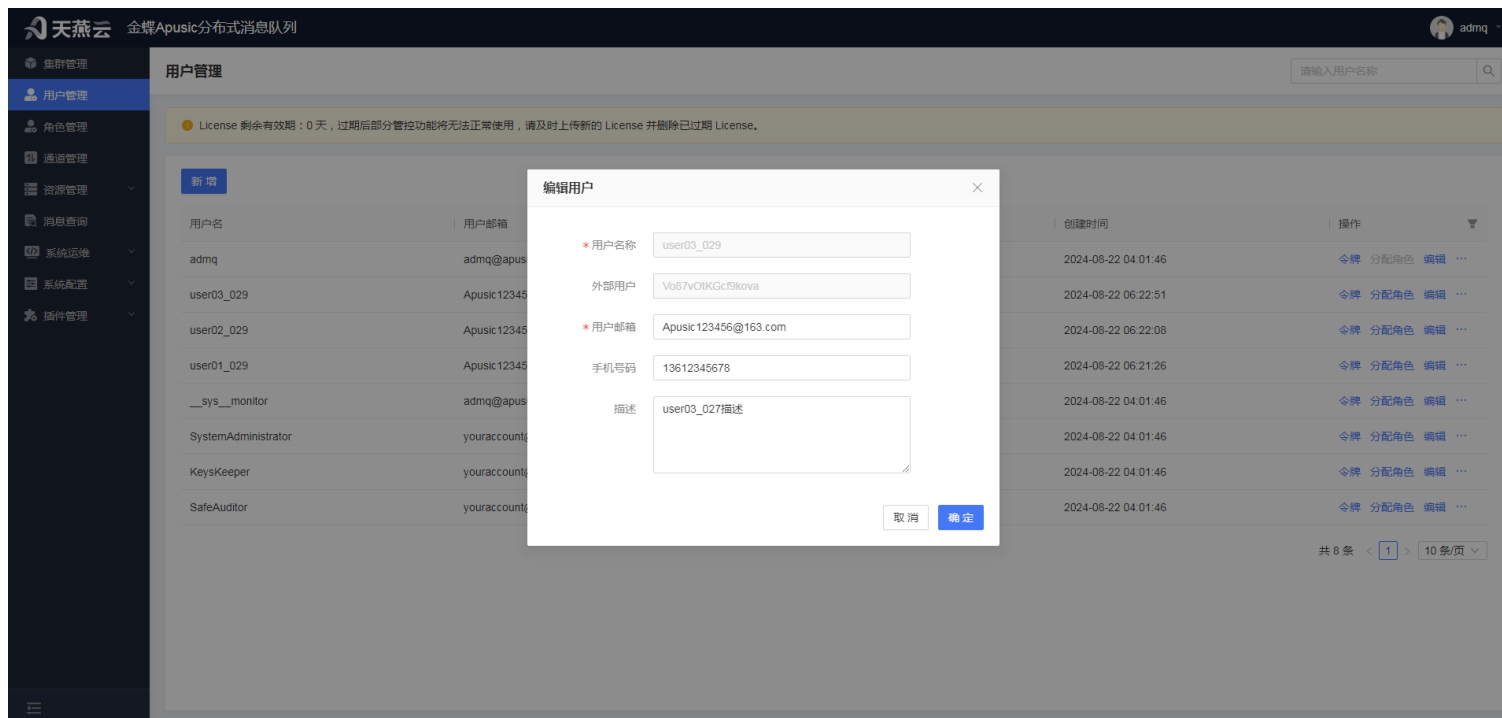


- 用户名称: 自定义, 唯一标识
- 用户密码: 用户的登录密码
- 确认密码: 用户的确认密码
- 用户邮箱: 用户的邮箱
- 手机号码: 用户的手机号码
- 描述: 用户的描述信息

- 备注：新增用户信息，需先建集群信息，且只有超级管理可以新增用户信息，普通用户不可新增用户信息。

5.4.1.2 编辑用户

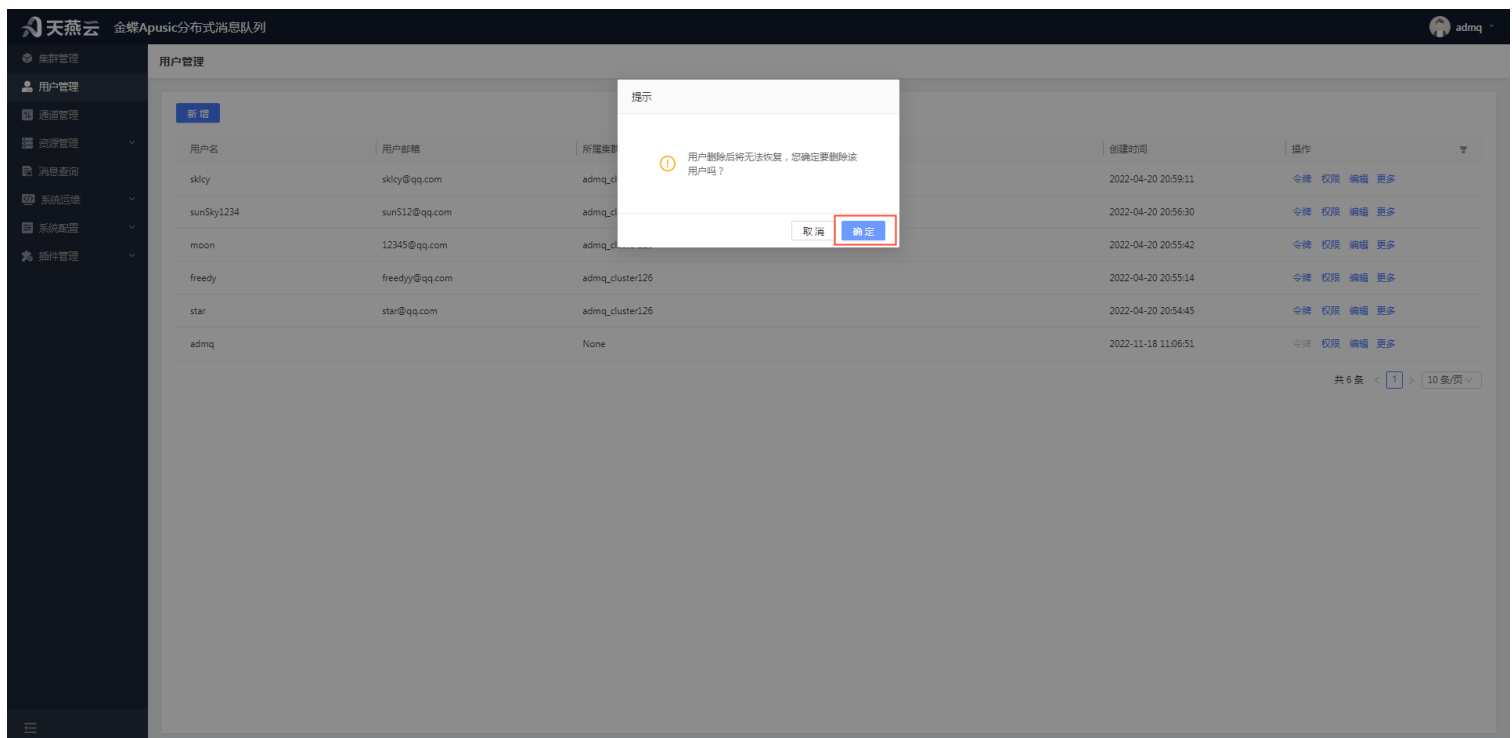
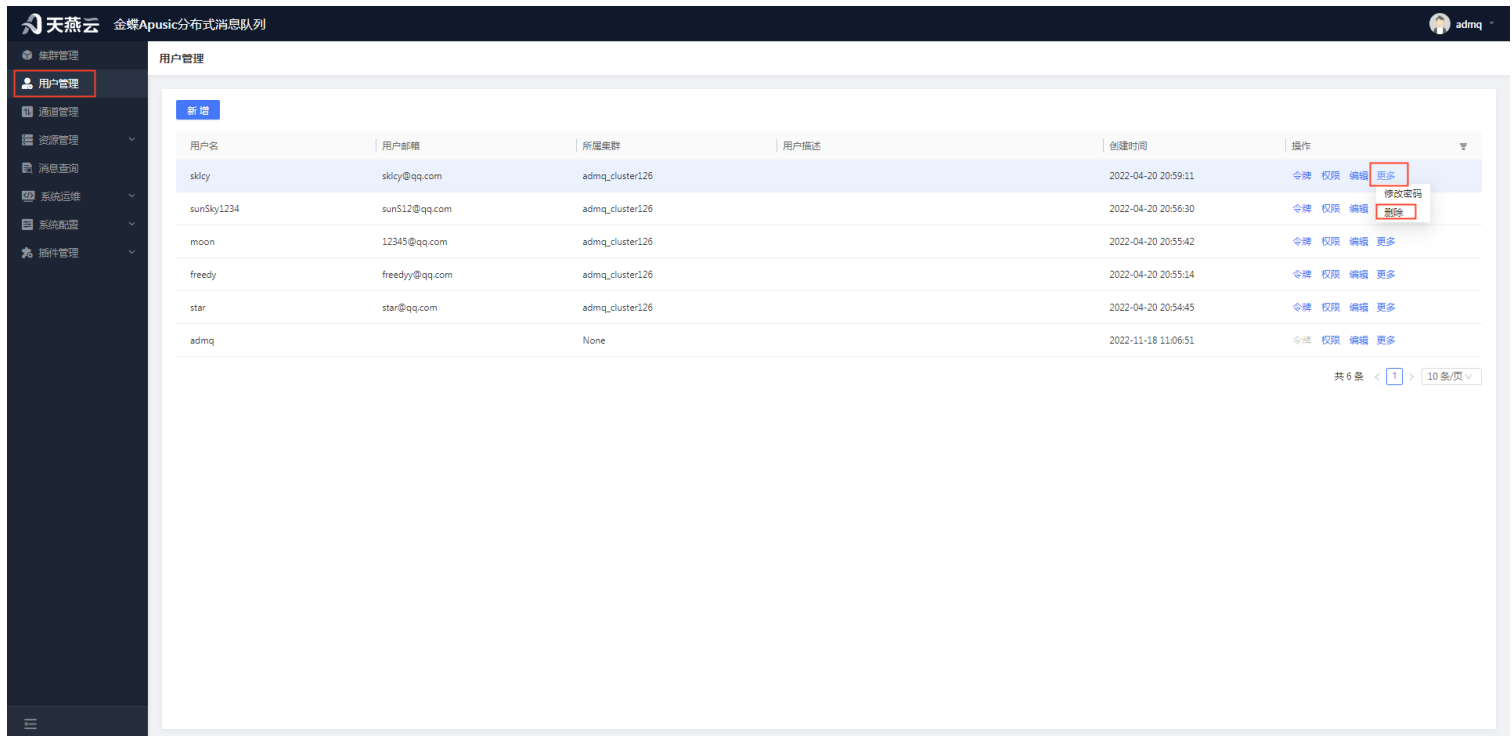
进入【用户管理】，进入用户管理页面，选择需要编辑的用户信息，点击【编辑】按钮，可修改用户信息。



- 用户名称：置灰显示，不可修改
- 外部用户：置灰显示，不可修改
- 用户邮箱：用户的邮箱，可修改
- 手机号码：用户的手机号码，可修改
- 描述：用户的描述信息，可修改

5.4.1.3 删除用户

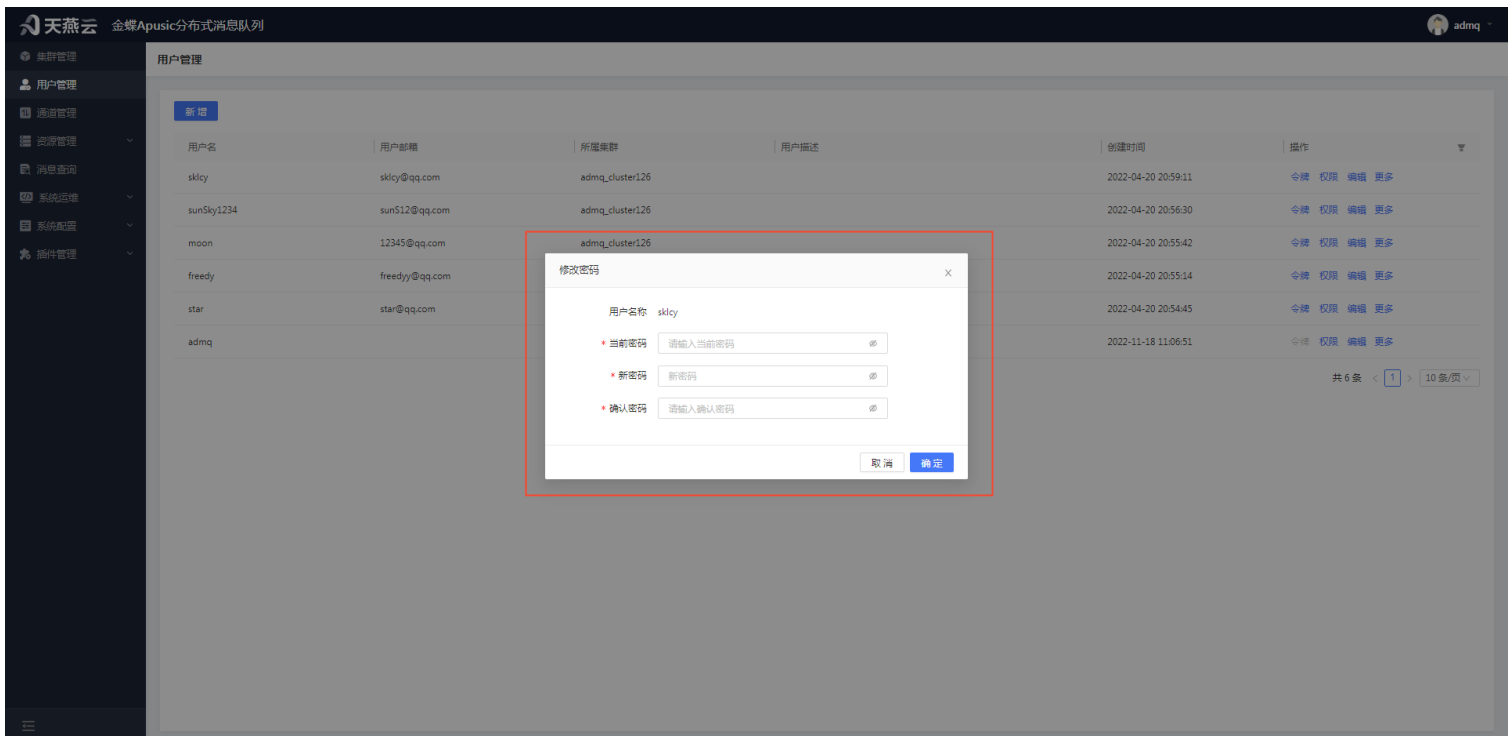
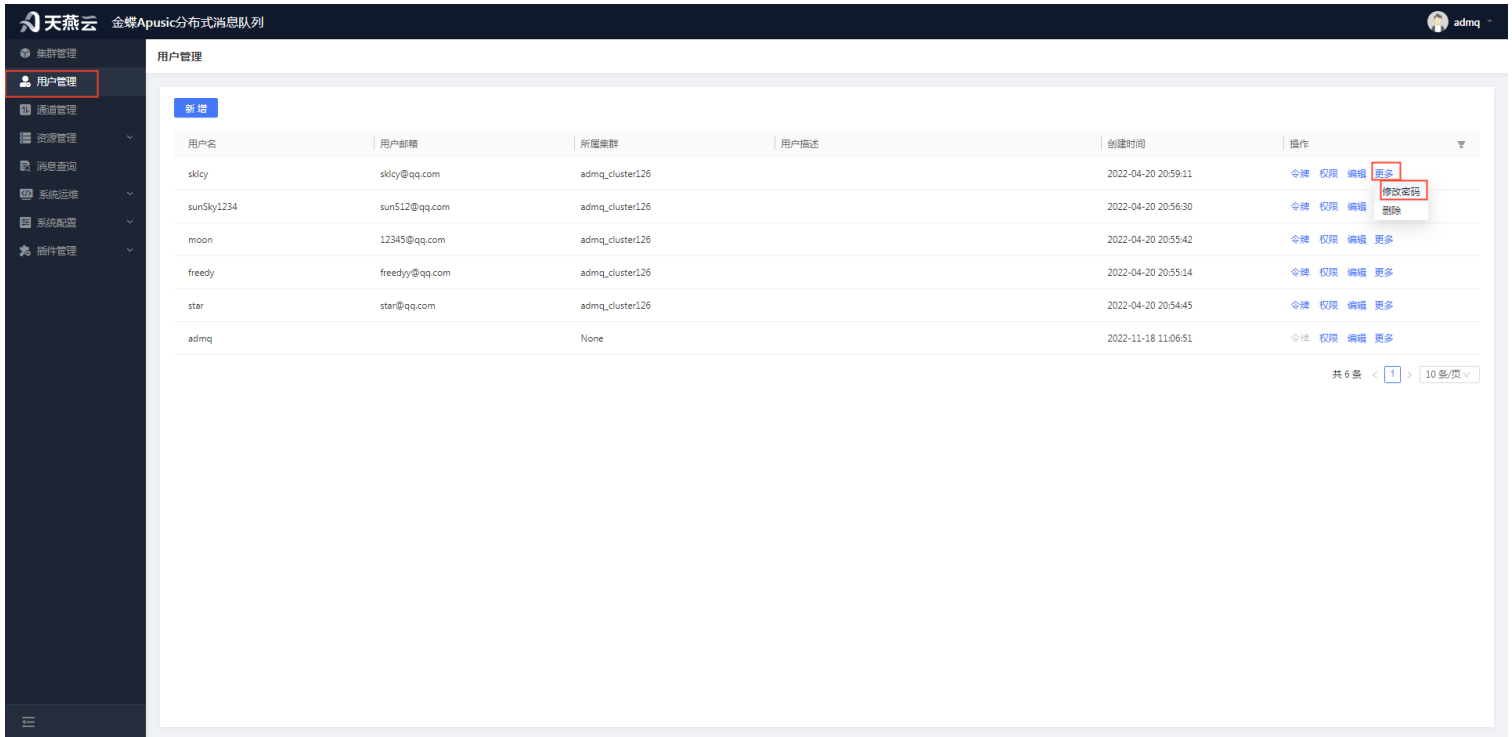
进入【用户管理】，进入用户管理页面，选择需要删除的用户信息，点击【更多】按钮-【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除用户信息。



- 备注：只有超级管理员用户可以删除用户信息，普通用户不可删除用户信息。

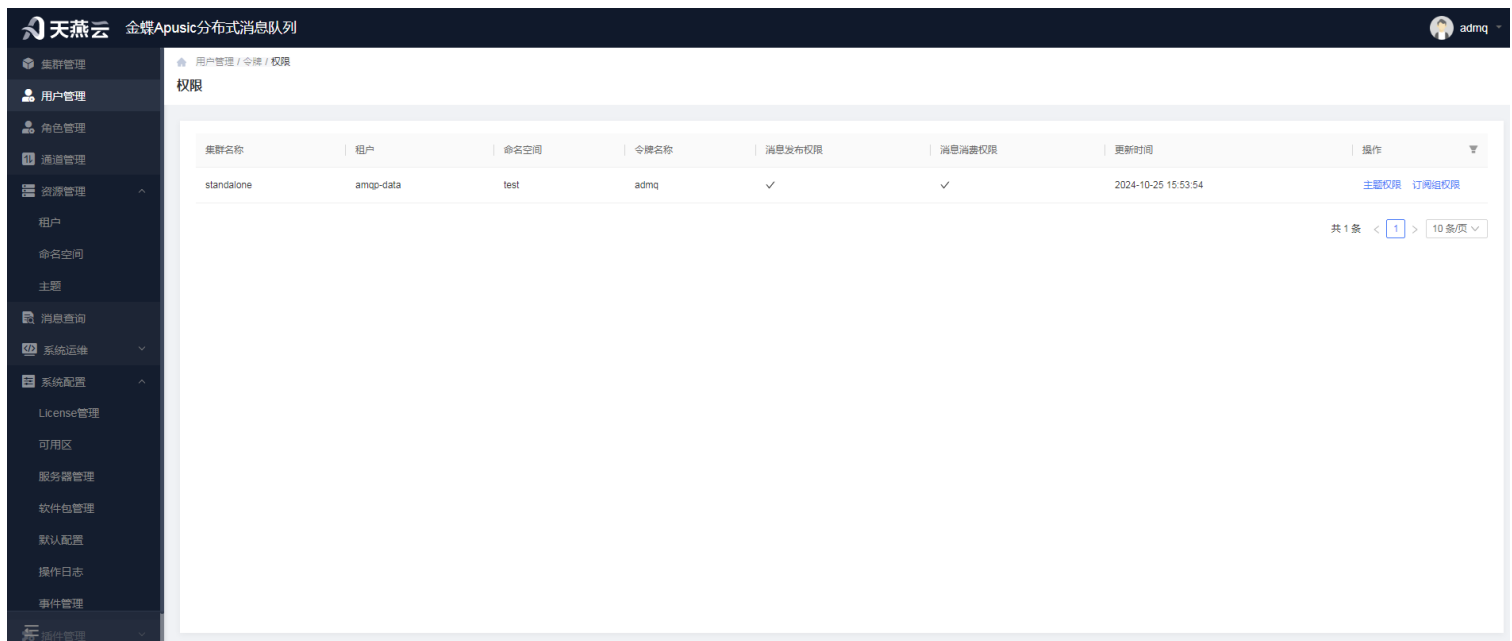
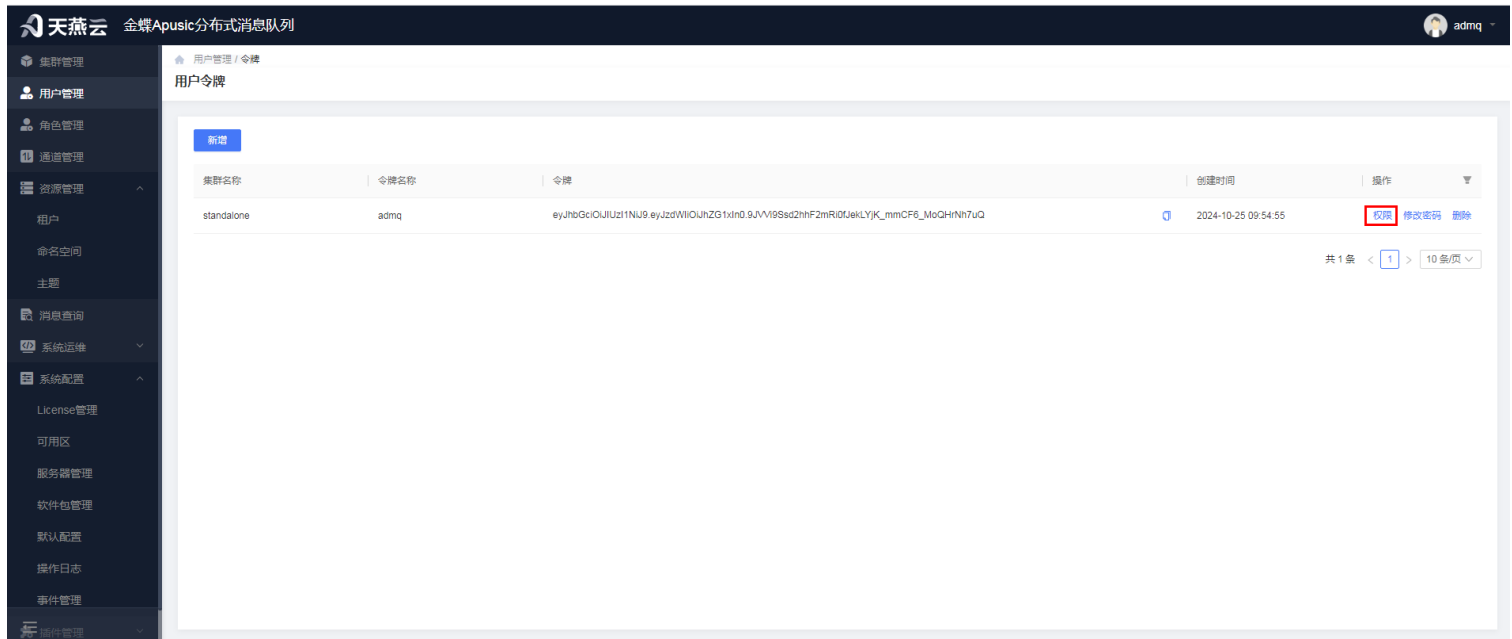
5.4.1.4 修改用户密码

进入【用户管理】，进入用户管理页面，选择需要修改密码的用户信息，点击【更多】按钮-【修改密码】按钮，可修改用户密码信息。



- 用户名: 用户的名称, 置灰显示, 不可修改
- 当前密码: 用户当前密码
- 新密码: 用户的新密码
- 确认密码: 再次输入用户的新密码
- 备注: 当前密码输入10次错误, 默认将会30分钟内禁止当前用户登录

5.4.1.5 查看用户令牌



5.4.1.7 普通用户新增令牌

5.4.1.7.1 普通用户新增令牌流程

新增普通用户、新增角色、角色授权（数据权限）、普通用户分配权限、新增令牌。

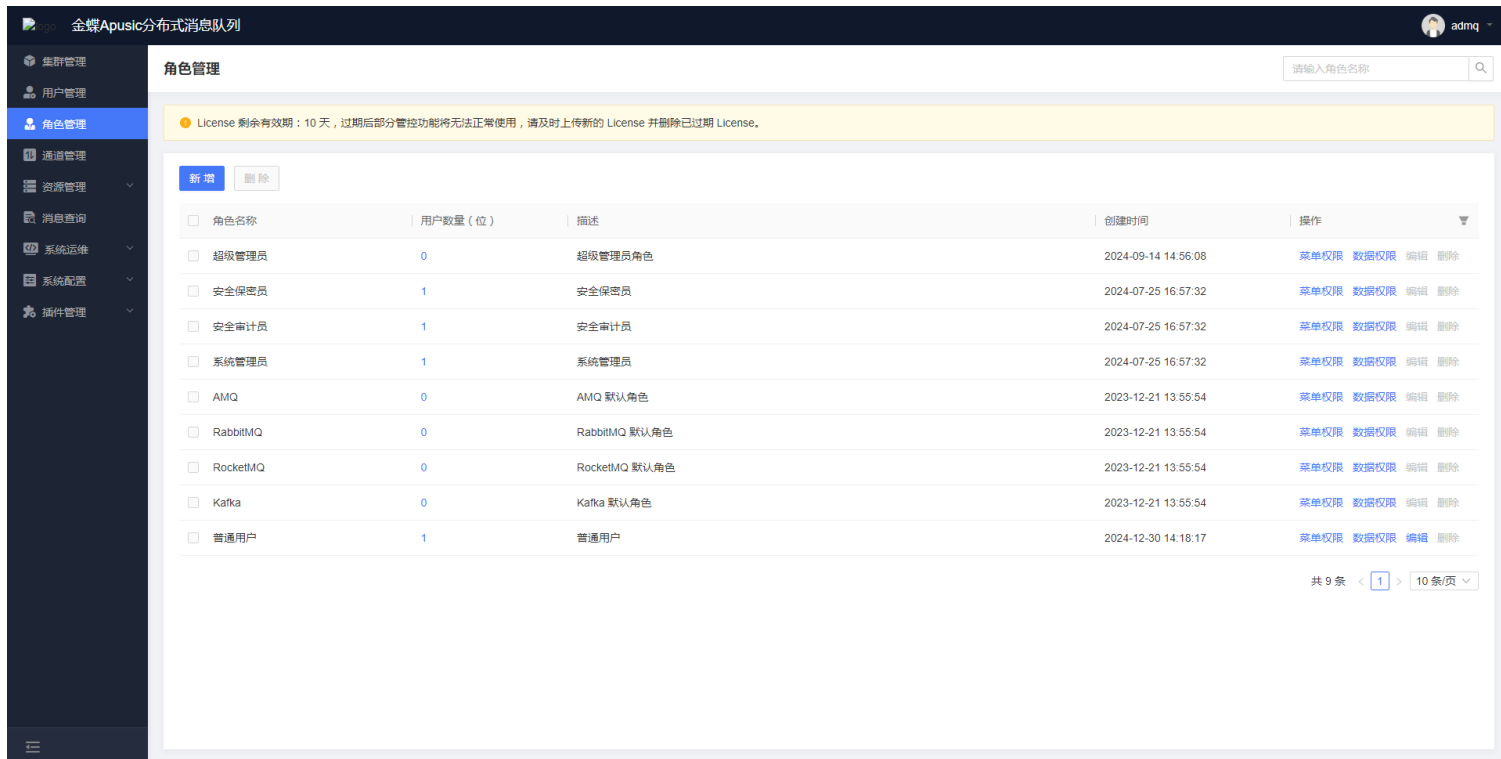
5.4.1.7.2 新增普通用户

进入【用户管理】，在用户管理页面点击左上角【新增】按钮，新增用户信息。

5.4.1.7.3 新增角色

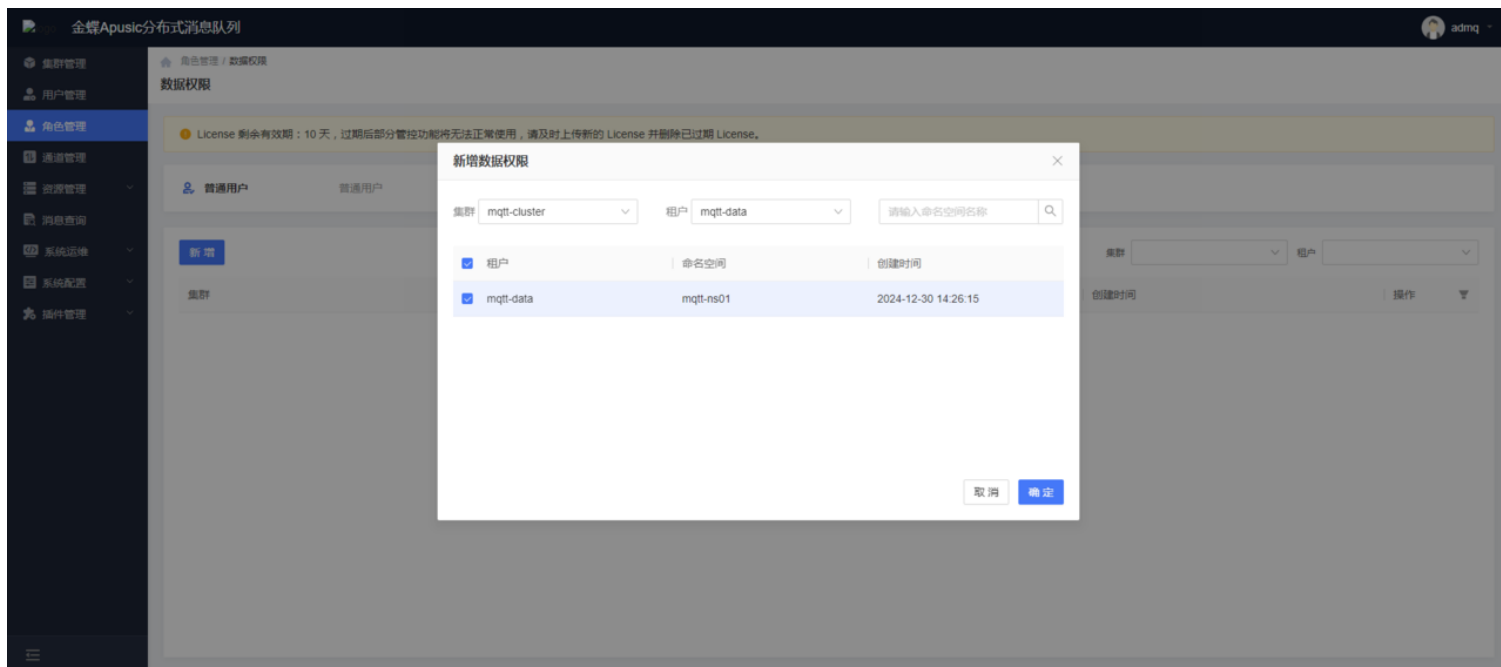
如果没有“普通用户”角色，需在【角色管理】创建“普通用户”角色。

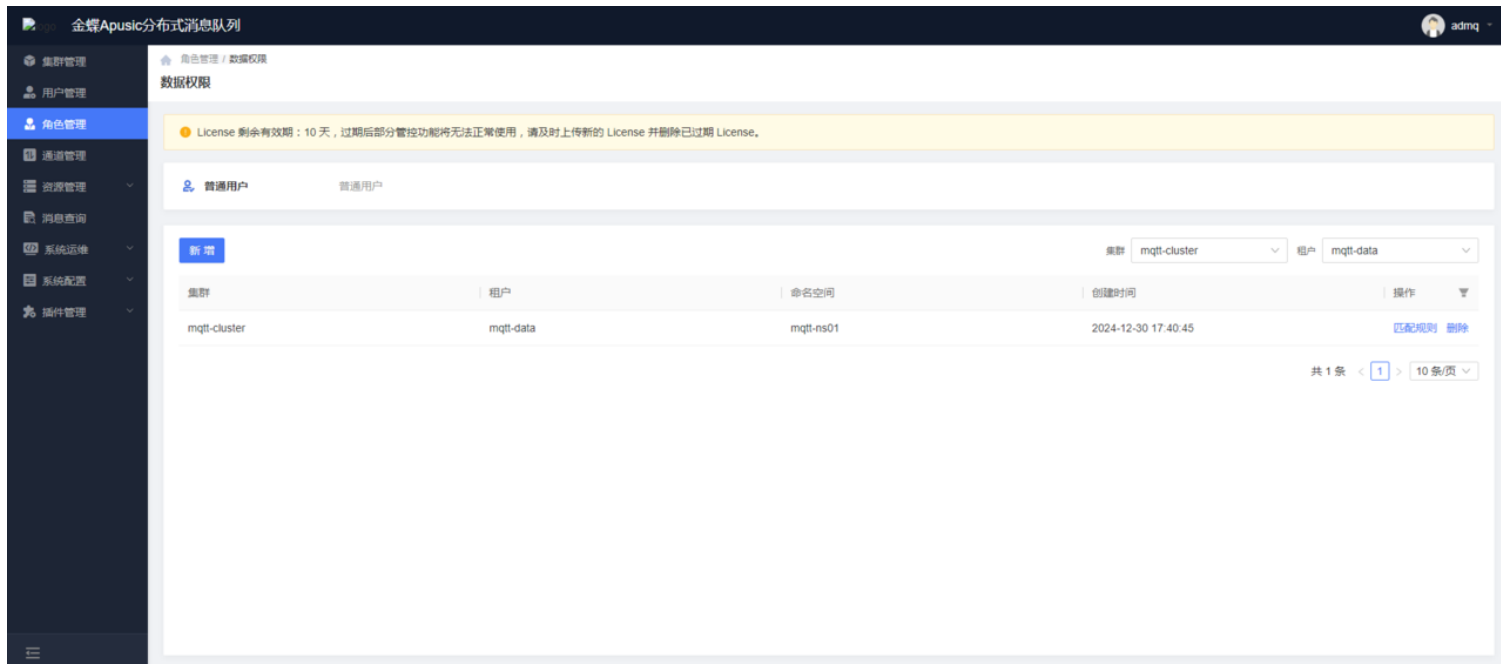
进入【角色管理】，进入角色管理页面，点击【新增】按钮，创建“普通用户”角色。



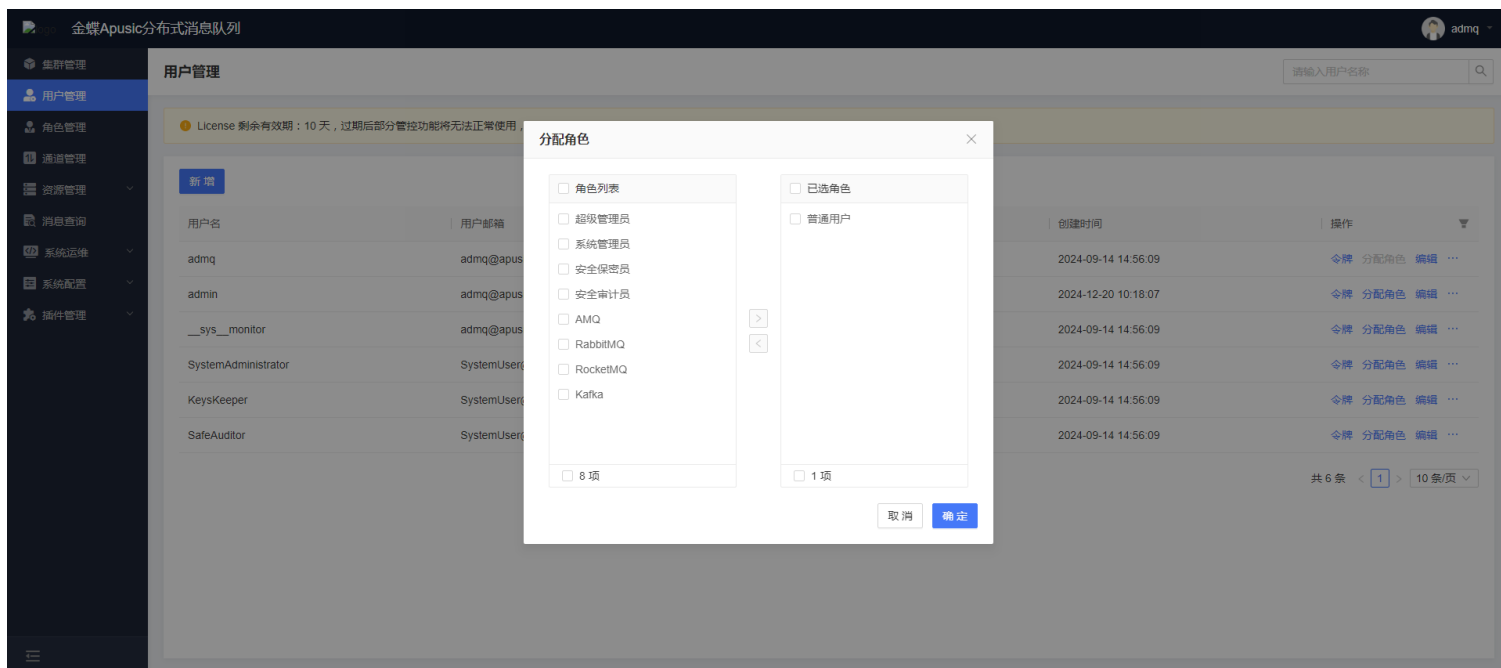
5.4.1.7.4 角色授权

点击【数据权限】，点击【新增】按钮，添加对应集群、租户、命名空间的权限。

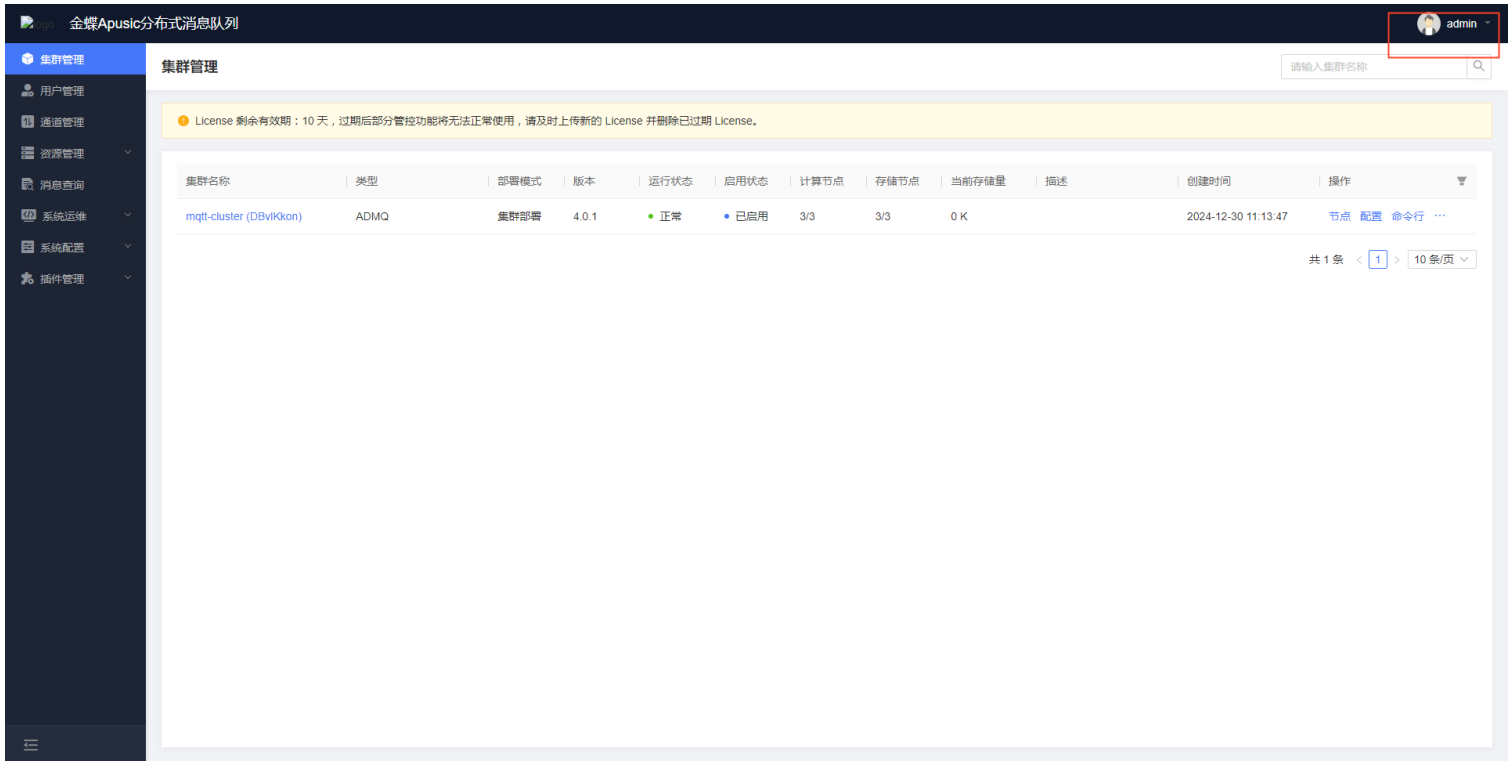




5.4.1.7.5 普通用户分配角色权限

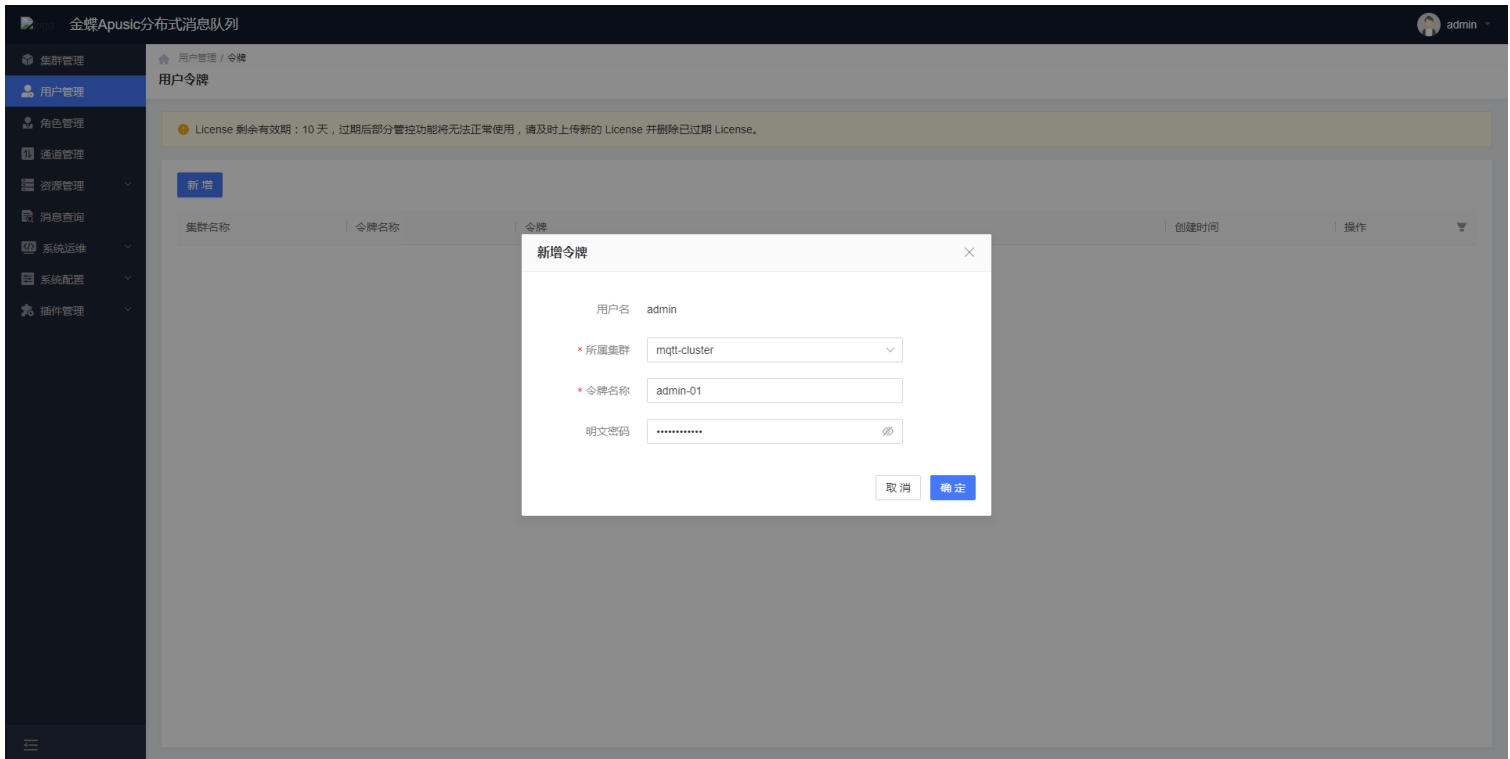


重新登录，使用新增用户登录管控台。

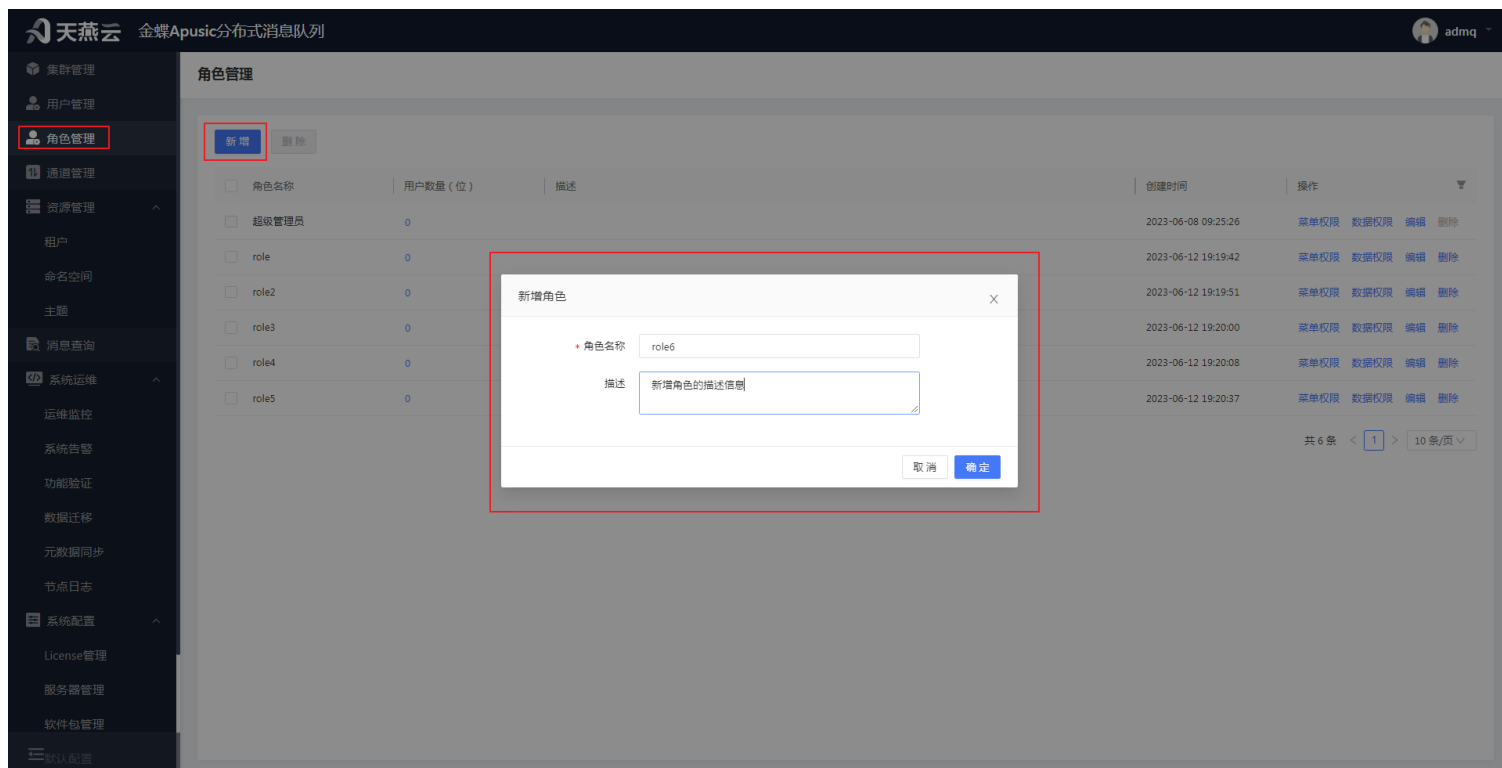


5.4.1.7.6 新增令牌

进入【用户管理】，进入用户管理页面，然后再点击【令牌】按钮，进入用户令牌页面，点击【新增】，设置所属集群，令牌，密码，创建对应令牌。



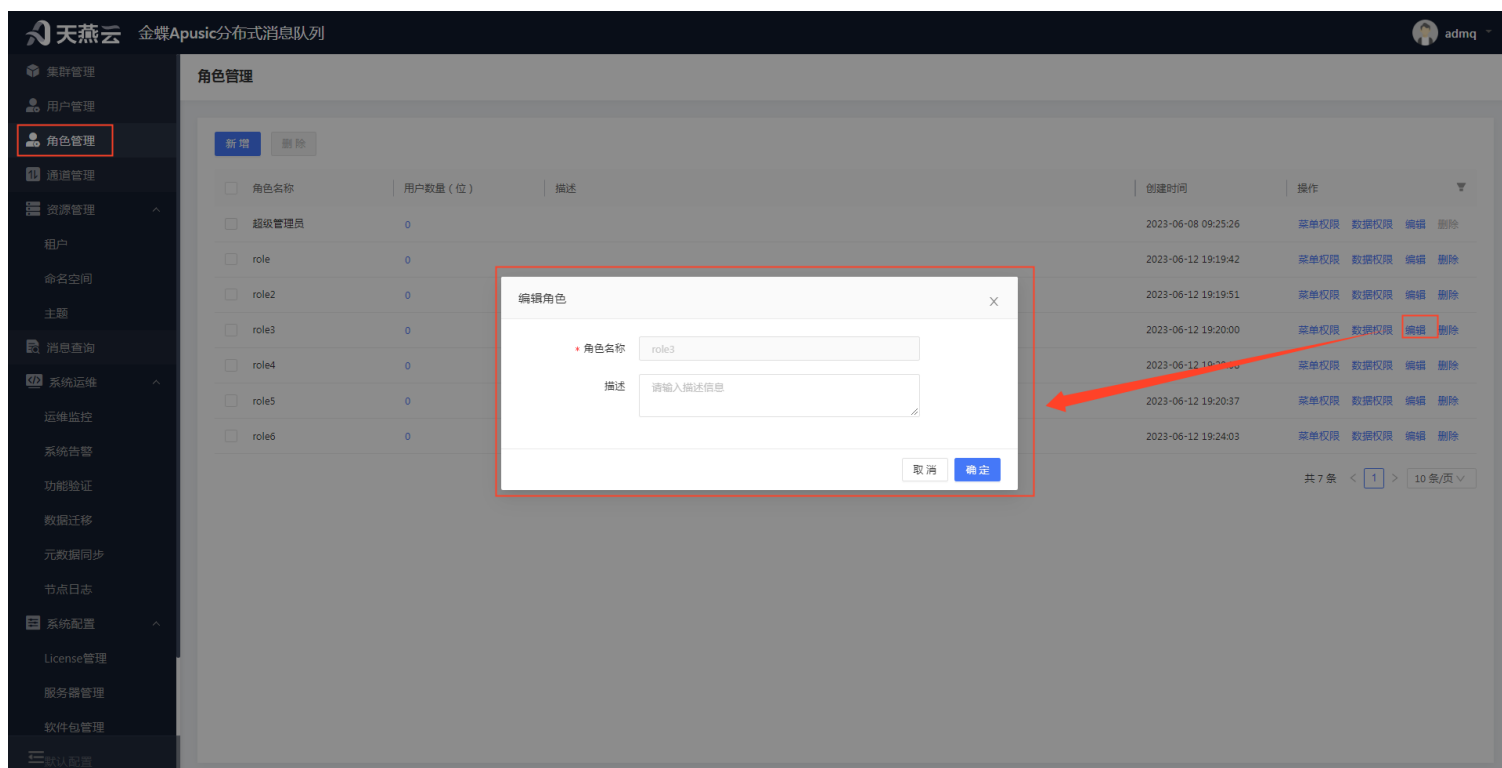
进入【角色管理】，在角色管理页面点击左上角【新增】按钮，可新增角色信息。



- 角色名称：自定义，唯一标识
- 描述：角色的描述信息

5.5.1.2 编辑角色

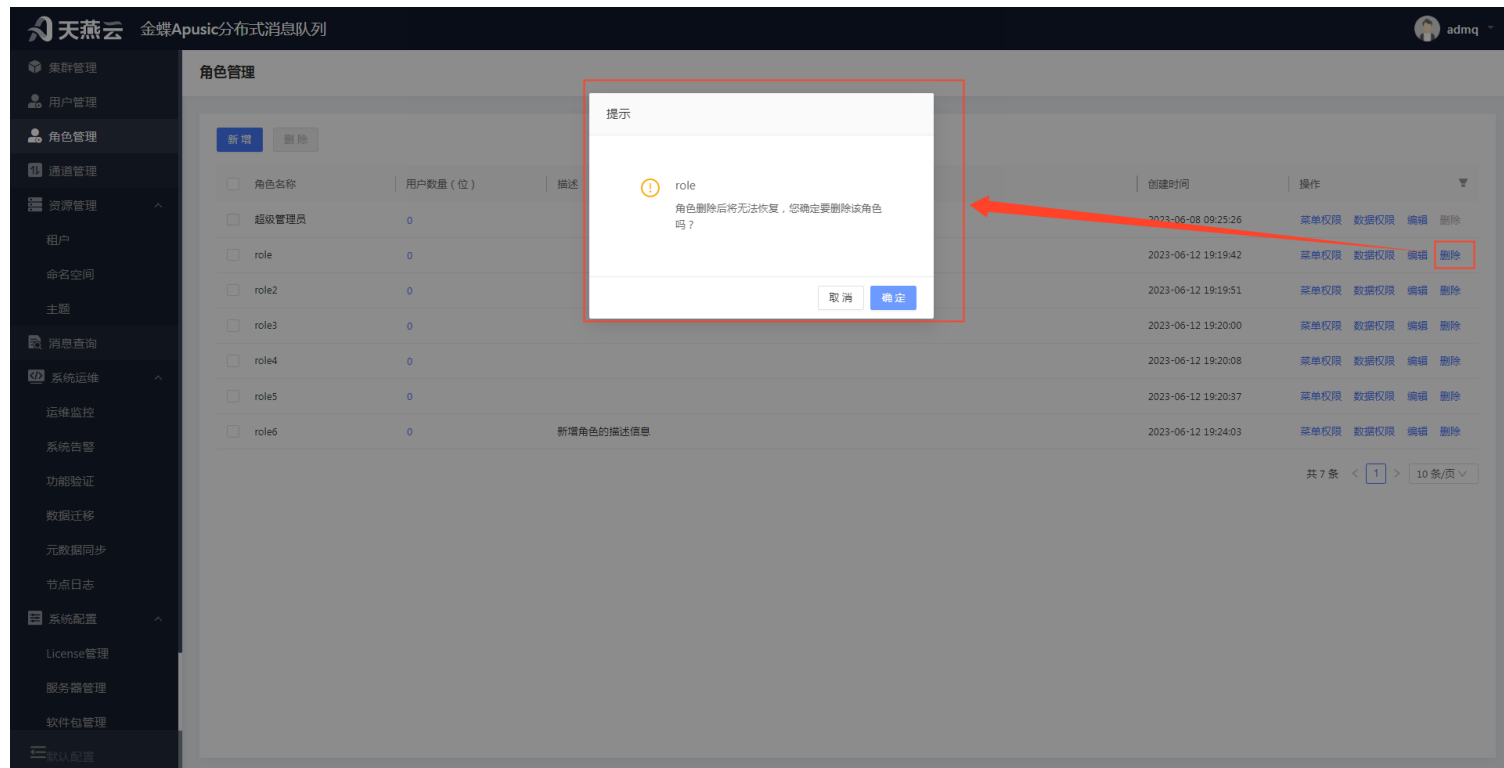
进入【角色管理】，在角色管理列表页面，选择需要编辑的角色信息，点击【编辑】按钮，可修改角色信息。



- 角色名称：置灰显示，不可修改
- 描述：角色的描述信息，可修改

5.5.1.3 删除角色

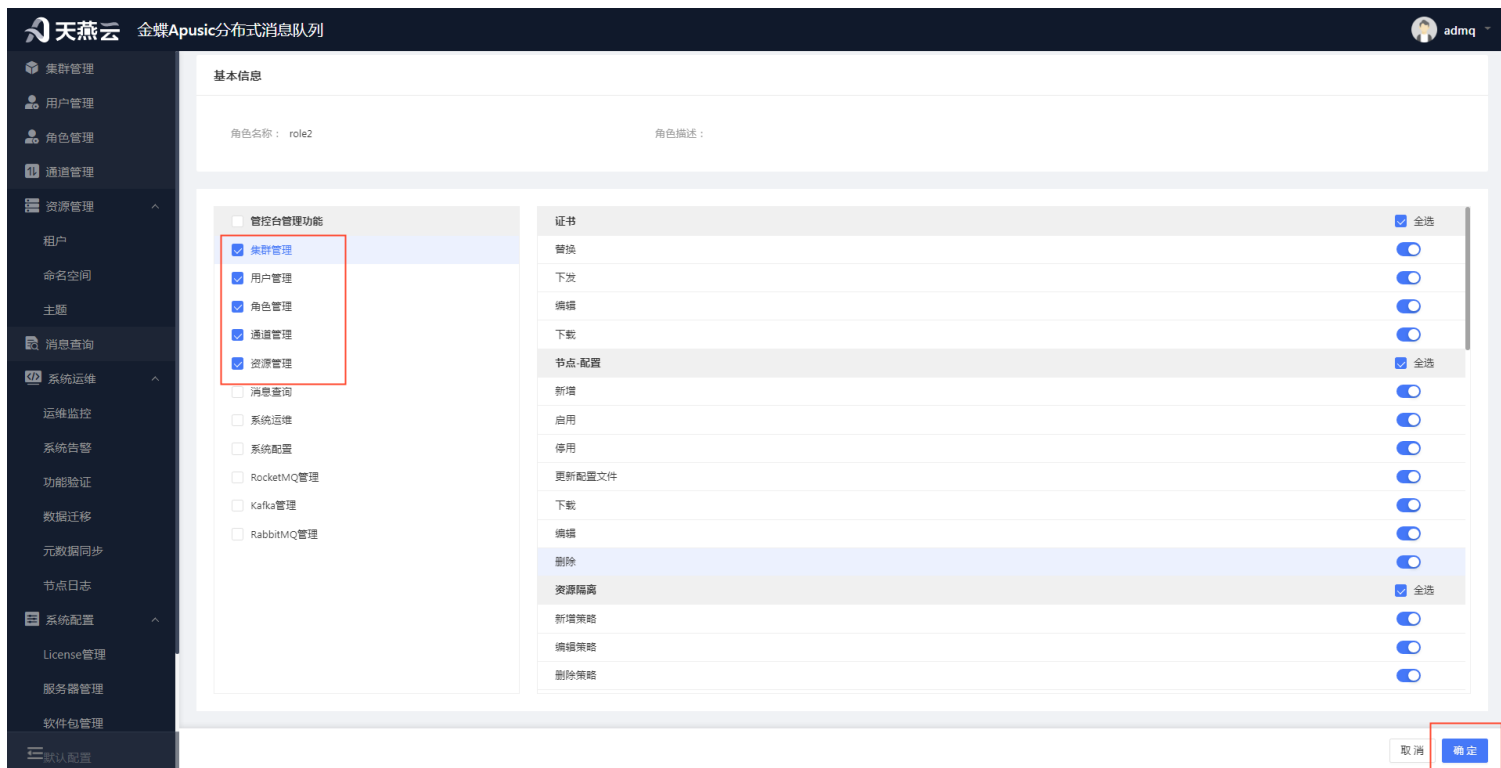
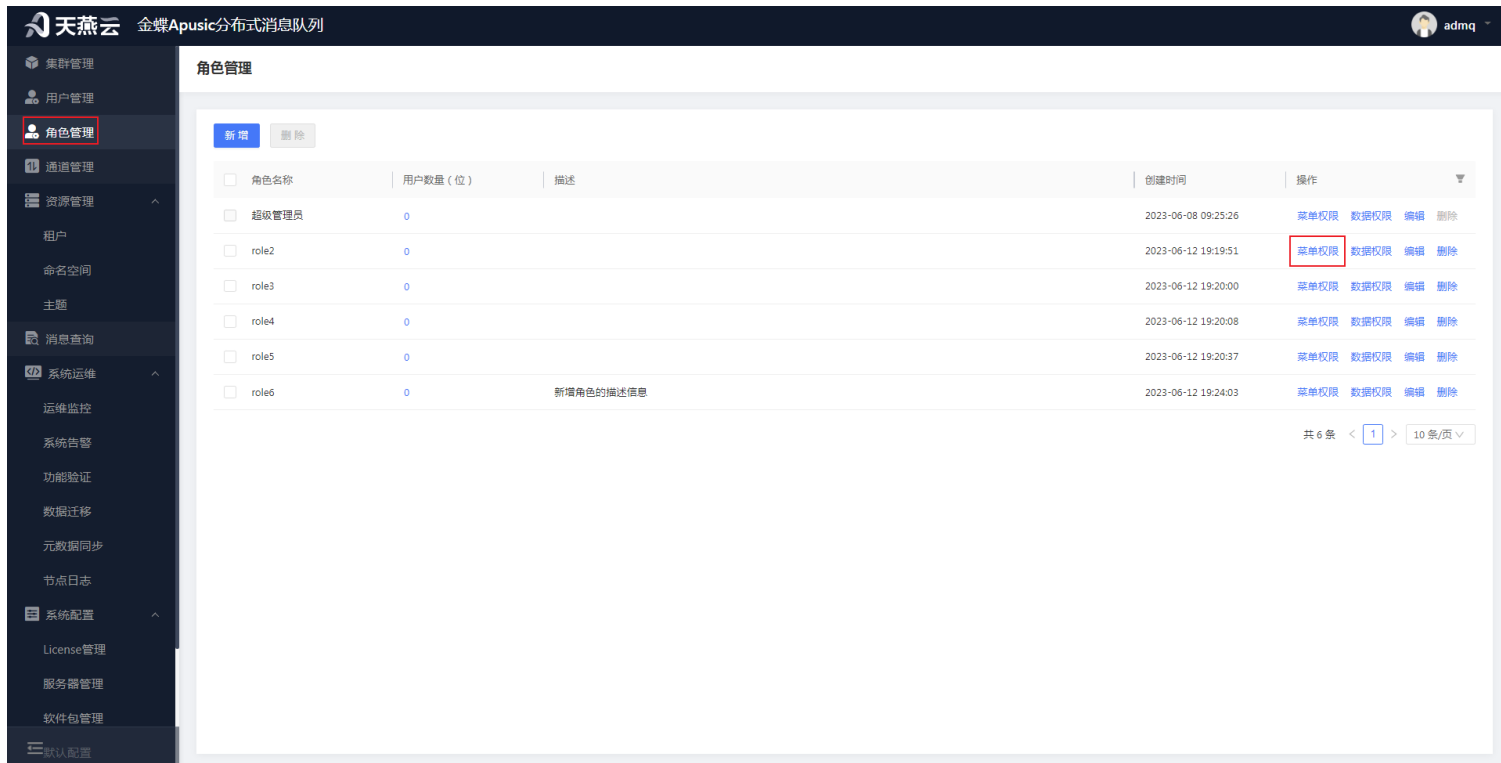
进入【角色管理】，在角色管理页面，选择需要删除的角色，点击【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除角色。



5.5.1.4 角色授权

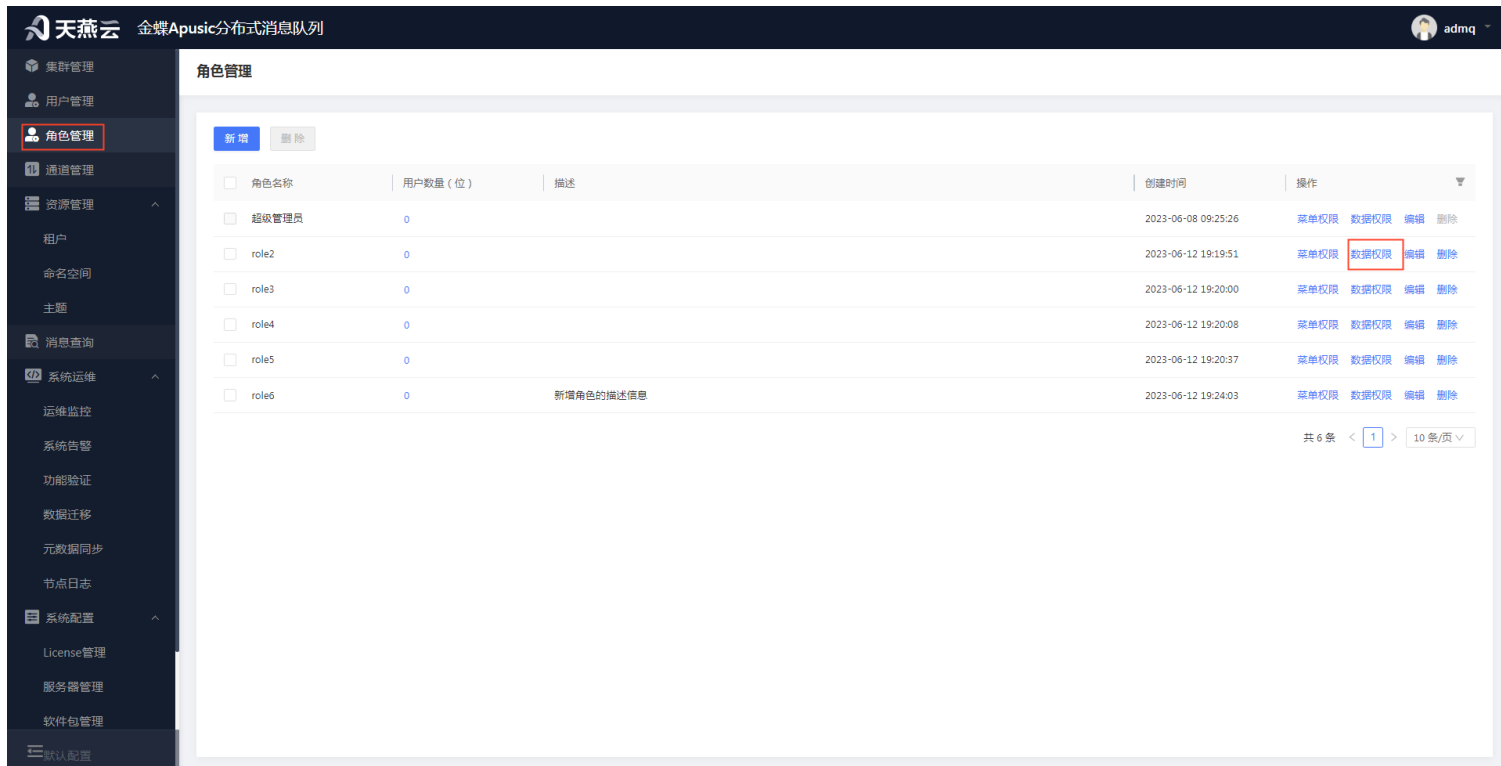
5.5.1.4.1 分配菜单权限

进入【角色管理】，在角色管理列表页面，选择某个角色进行分配菜单权限，点击【菜单权限】按钮，进入到菜单权限界面，然后选中相关管控台功能，点击确定按钮，完成菜单权限分配：

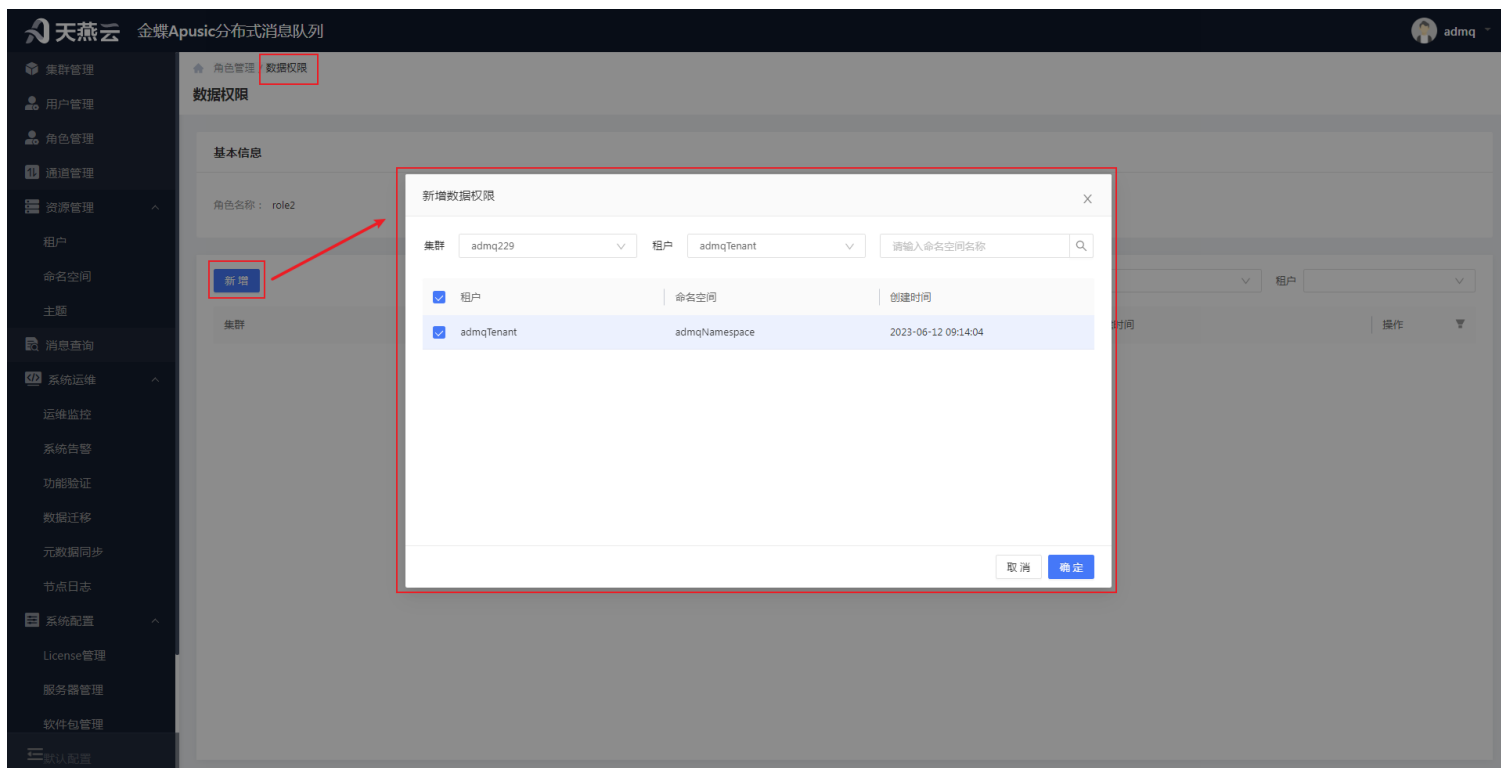


5.5.1.4.2 分配数据权限

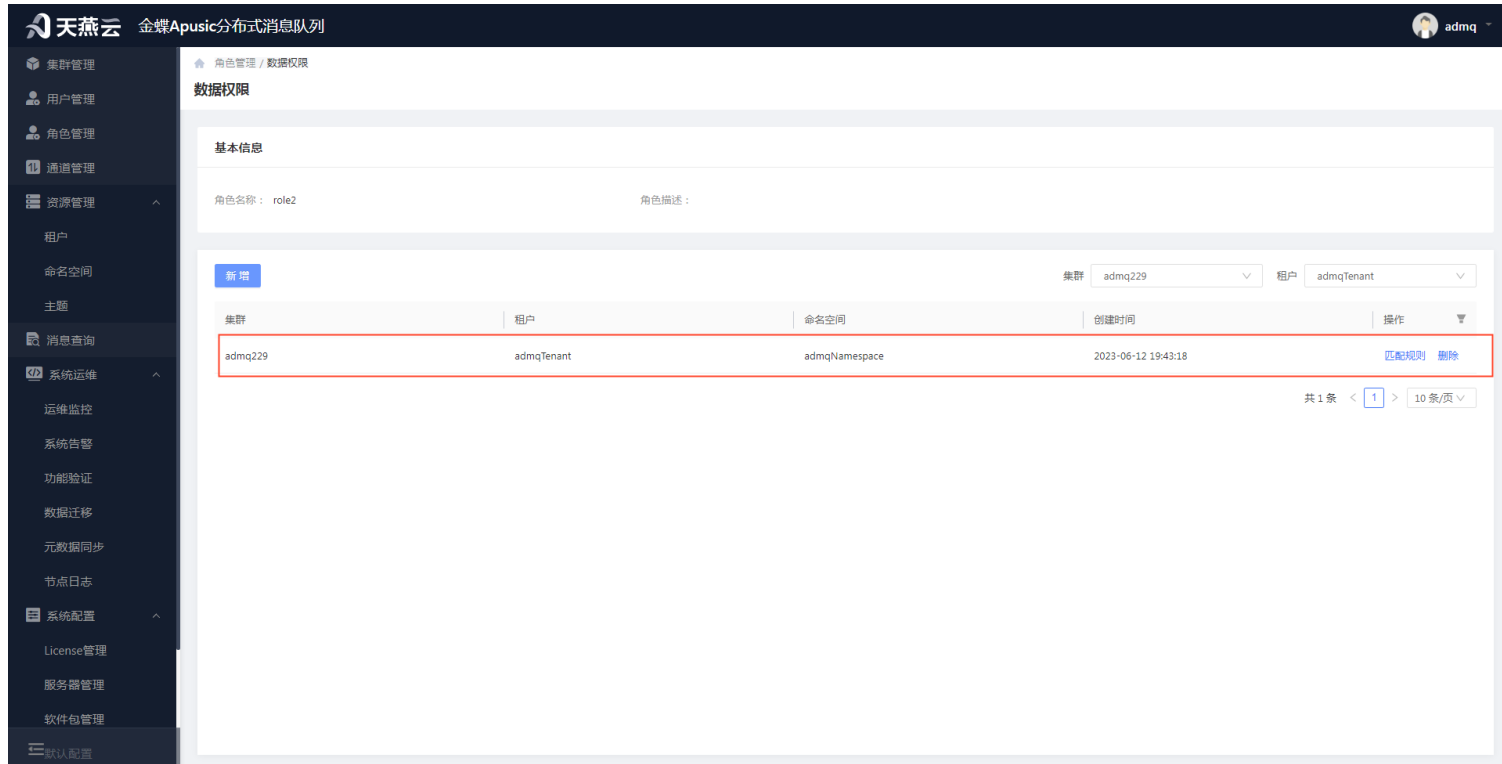
进入【角色管理】，在角色管理列表页面，选择某个角色进行分配数据权限，点击【数据权限】按钮，进入到数据权限界面：



在数据权限界面，点击【新增】按钮，出现新增数据权限弹窗，输入相关信息，点击确定按钮，可为某个角色添加数据权限：



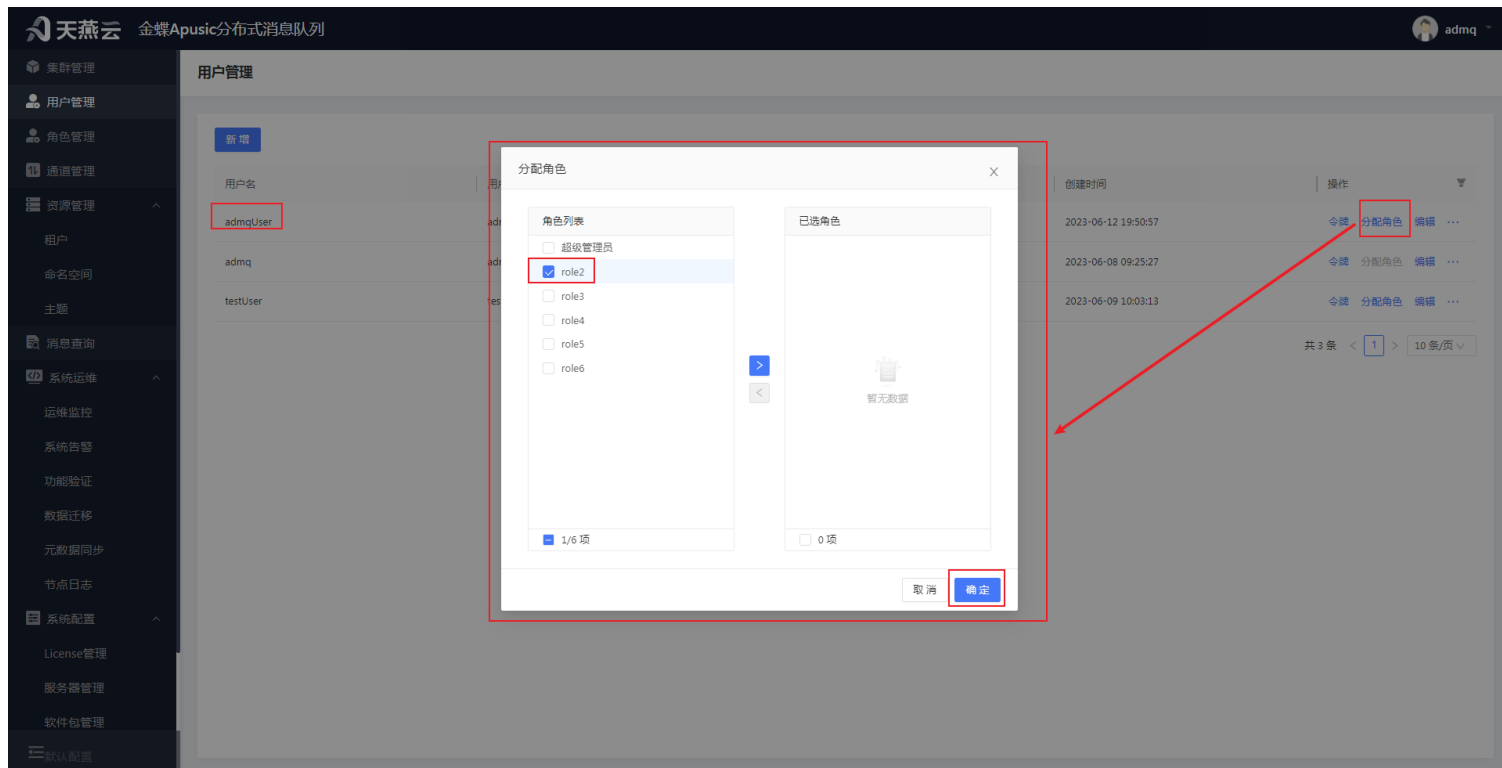
为角色添加数据权限成功后，会在数据权限列表页面展示出来：



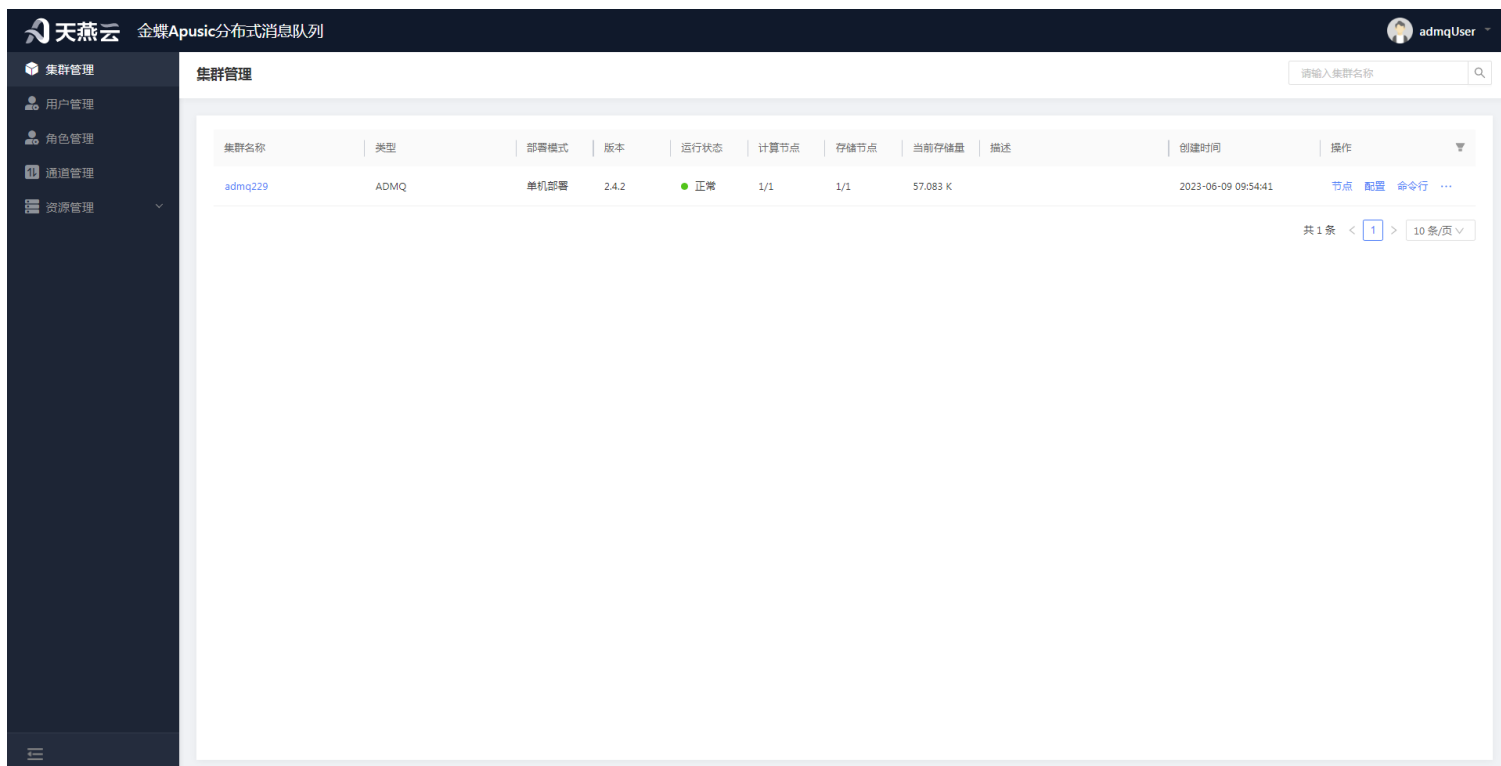
5.5.1.5 角色分配

角色分配是指将授予了不同权限的角色分配给不同的普通用户，以控制普通用户的访问权限和操作能力。备注：首先先创建普通用户，用户名为admq为超级用户，默认拥有所有菜单权限和数据权限。不需要再为其分配权限。

进入【用户管理】，在用户管理列表页面，选择需要分配角色的普通用户，点击【分配权限】按钮，会出现分配角色弹窗，选中相关角色，点击确定按钮，可为该用户分配相关角色：



使用普通用户重新登录管控台：

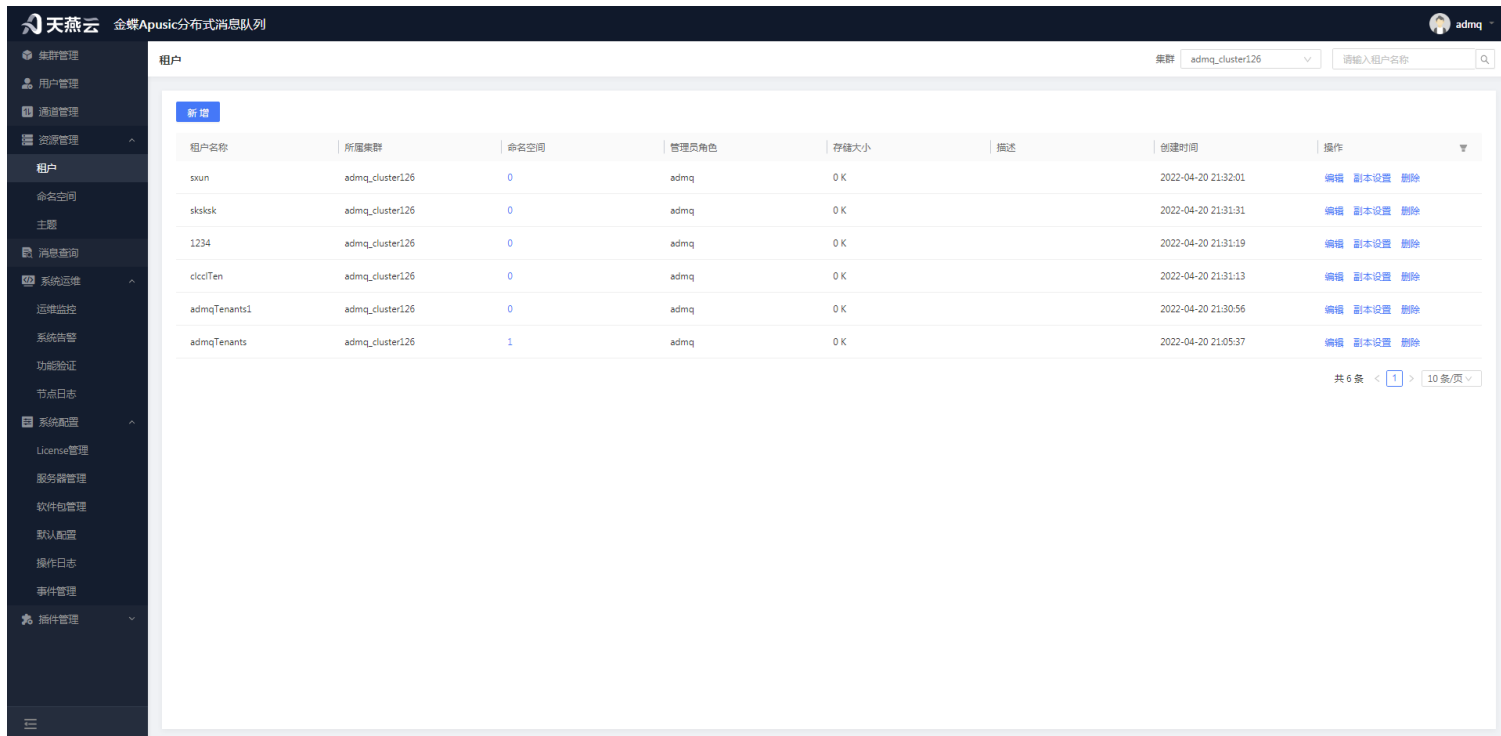


5.6 资源管理

资源管理用于管理监控租户、命名空间、主题，为消息中间件提供基础资源建立。

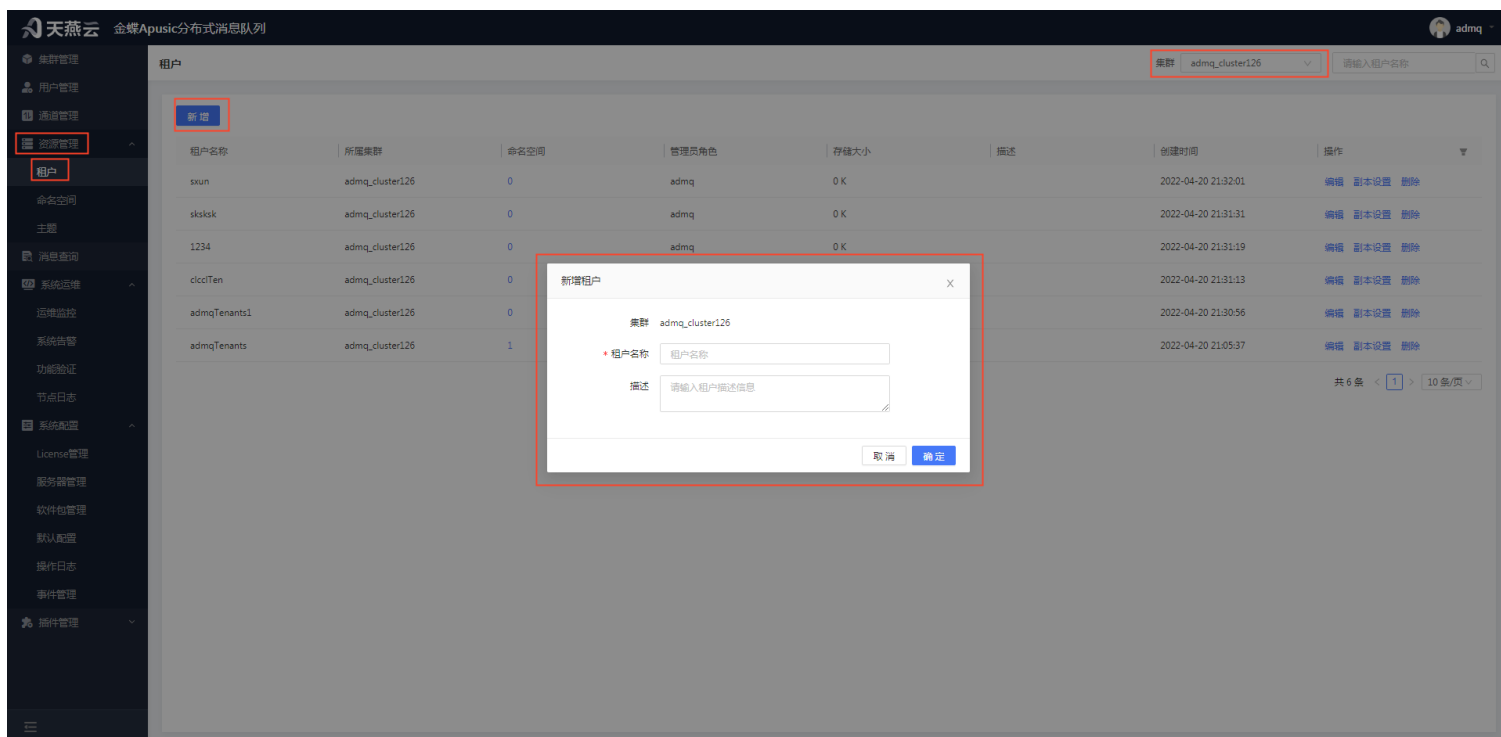
5.6.1 租户

租户用于新增租户信息、编辑租户信息、删除租户信息、查询租户信息、设置副本。注意：新增、编辑、删除租户之前，需要保证集群服务正常，否则操作将会失败。



5.6.1.1 新增租户

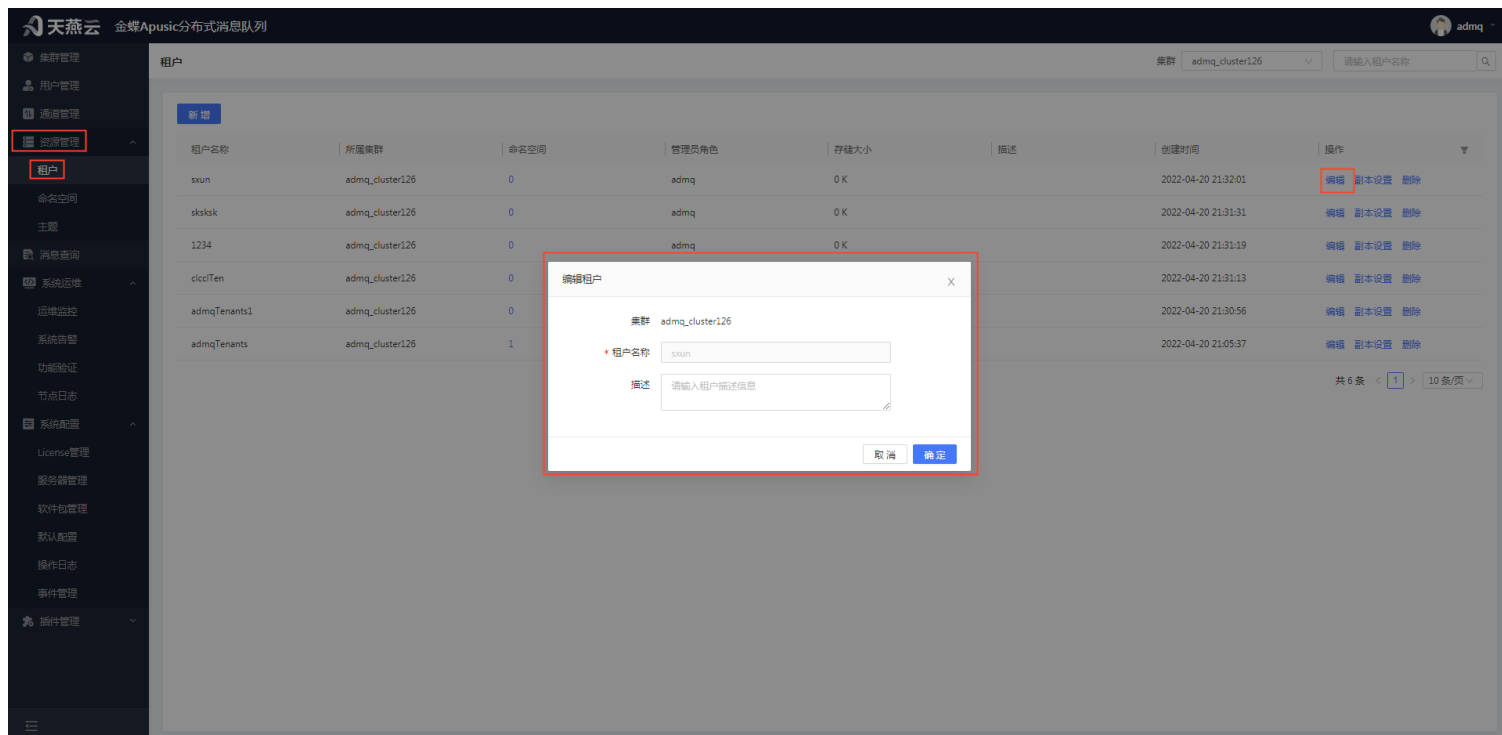
进入【资源管理】->【租户】，在租户页面左上角点击【新增】按钮，可新增租户信息。注意：点击新增前，需在右上角下拉选择需要新增租户的集群，点击新增按钮时，将会自动带入选择的集群名称。



- 集群：租户所属的集群，用户新增前，自行选择
- 租户名称：租户的名称，自定义，唯一性标识（不同集群下可重复）
- 描述：租户描述信息

5.6.1.2 编辑租户

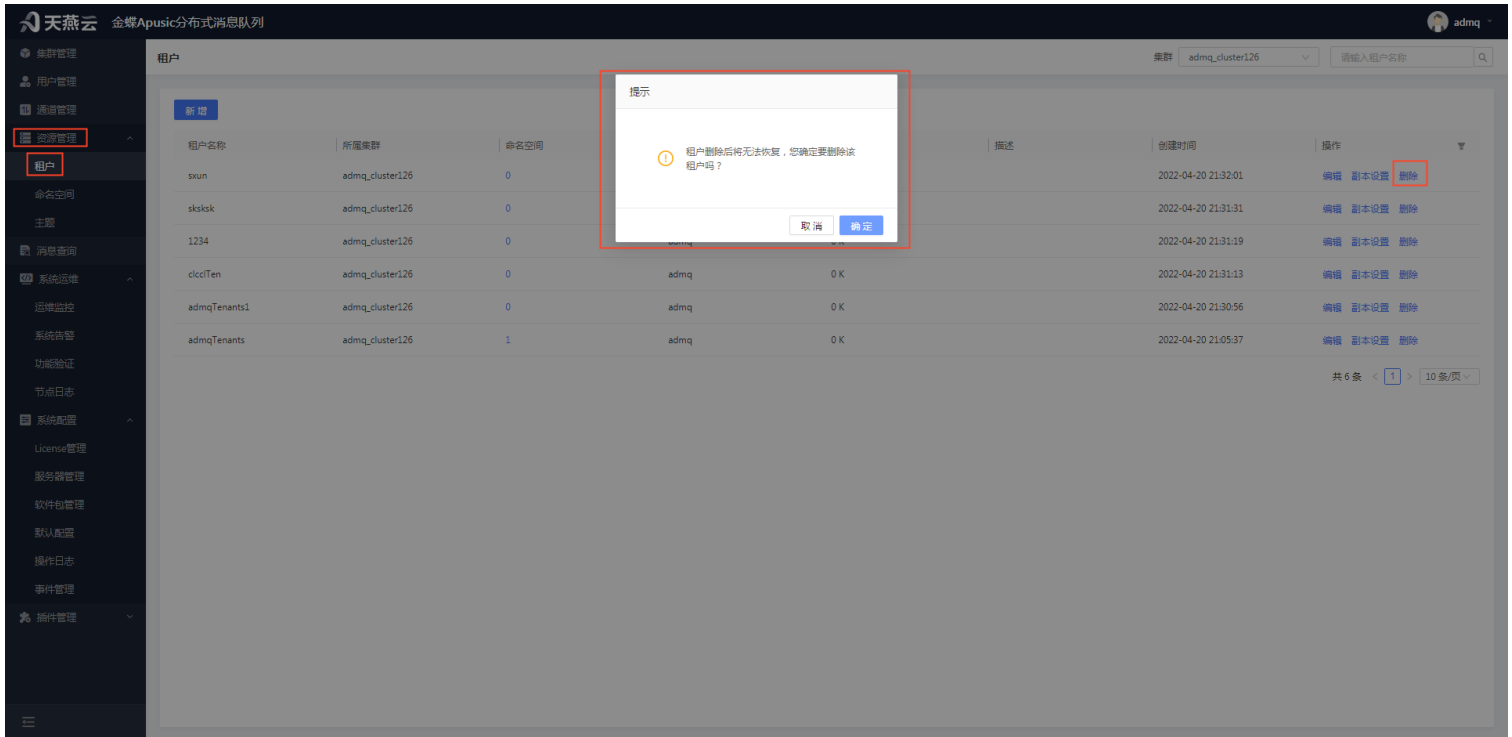
进入【资源管理】->【租户】，在租户页面选择需要修改的租户，点击【编辑】按钮，可编辑租户信息。注意：在修改租户信息前，默认只显示一个集群的租户信息，可在右上角下拉选择集群。



- 集群：租户所属的集群，不可修改
- 租户名称：租户名称，置灰显示，不可修改
- 描述：租户描述信息

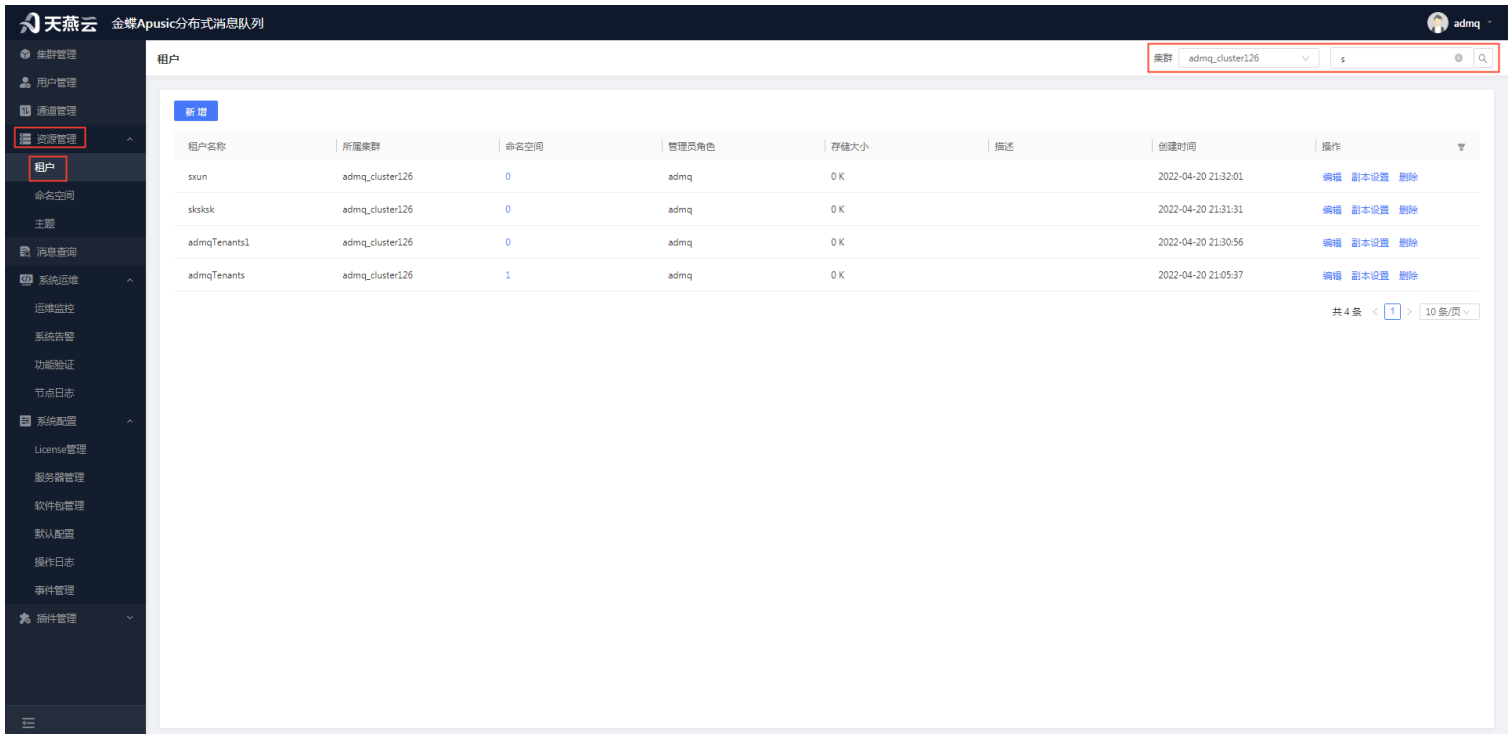
5.6.1.3 删除租户

进入【资源管理】->【租户】，在租户页面选择需要删除的租户，点击【删除】按钮，可删除租户信息。注意：当租户下存在命名空间时，不可删除租户信息，要删除租户，需要先删除命名空间。



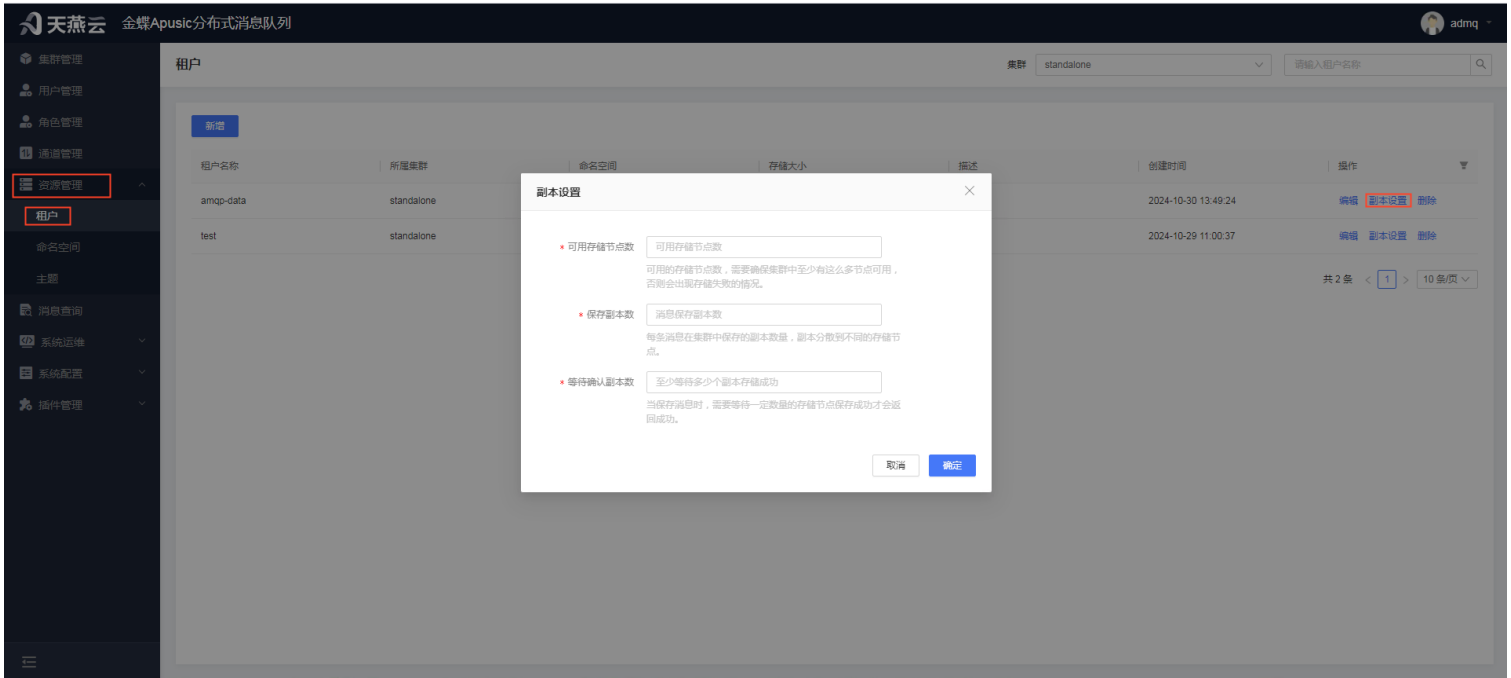
5.6.1.4 查询租户

进入【资源管理】->【租户】，在租户页面选择集群信息，精确或者模糊输入租户名称，可查询租户信息。



5.6.1.5 副本设置

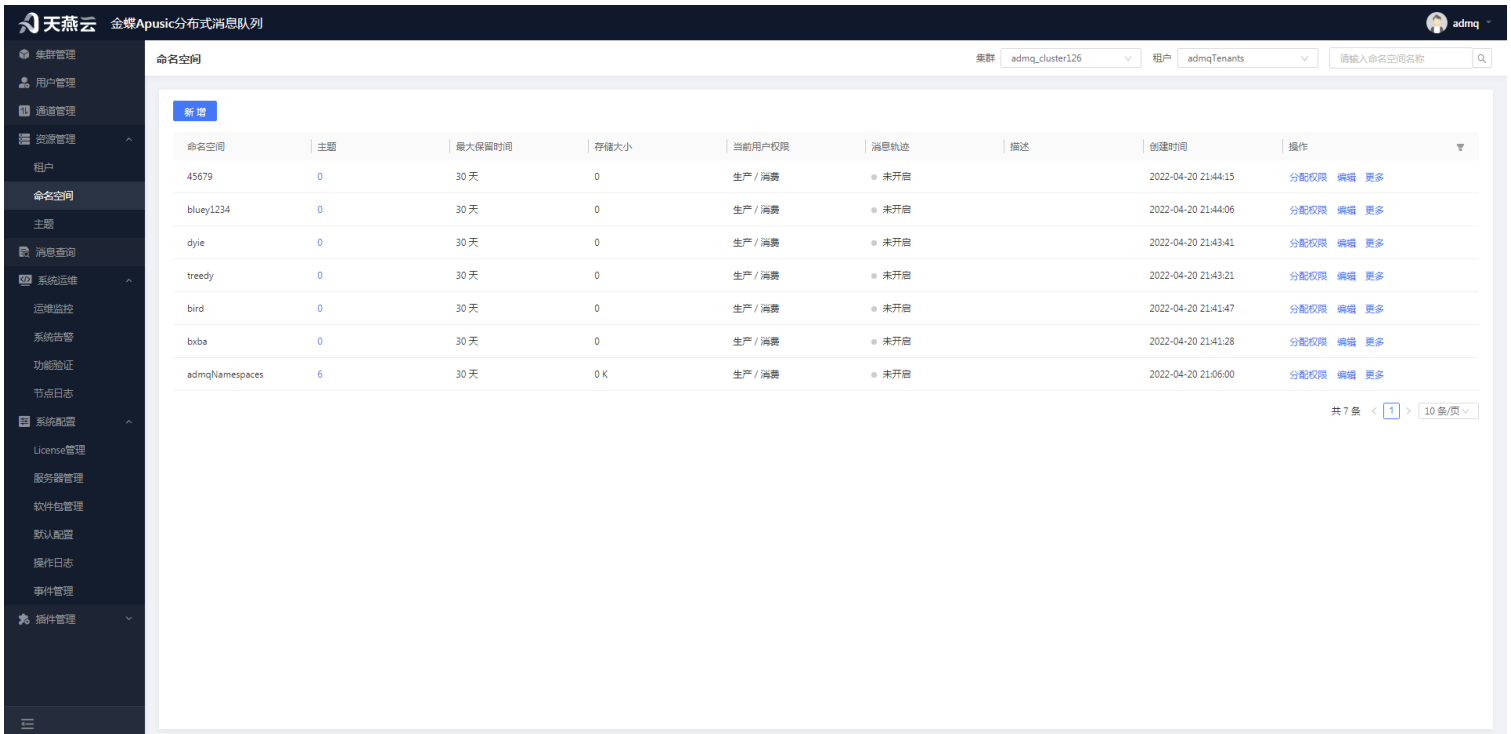
进入【资源管理】->【租户】，在租户页面选择要设置副本的租户，点击【副本设置】，可设置副本。



- 可用存储节点数：可用的存储节点数，自定义，不能大于租户所属集群的可用存储节点数
- 保存副本数：消息保存副本数，自定义，不能大于可用存储节点个数，且不小于等待确认副本数
- 等待确认副本数：至少等到多少个副本存储成功，自定义，不可大于可用存储节点个数和保存副本数

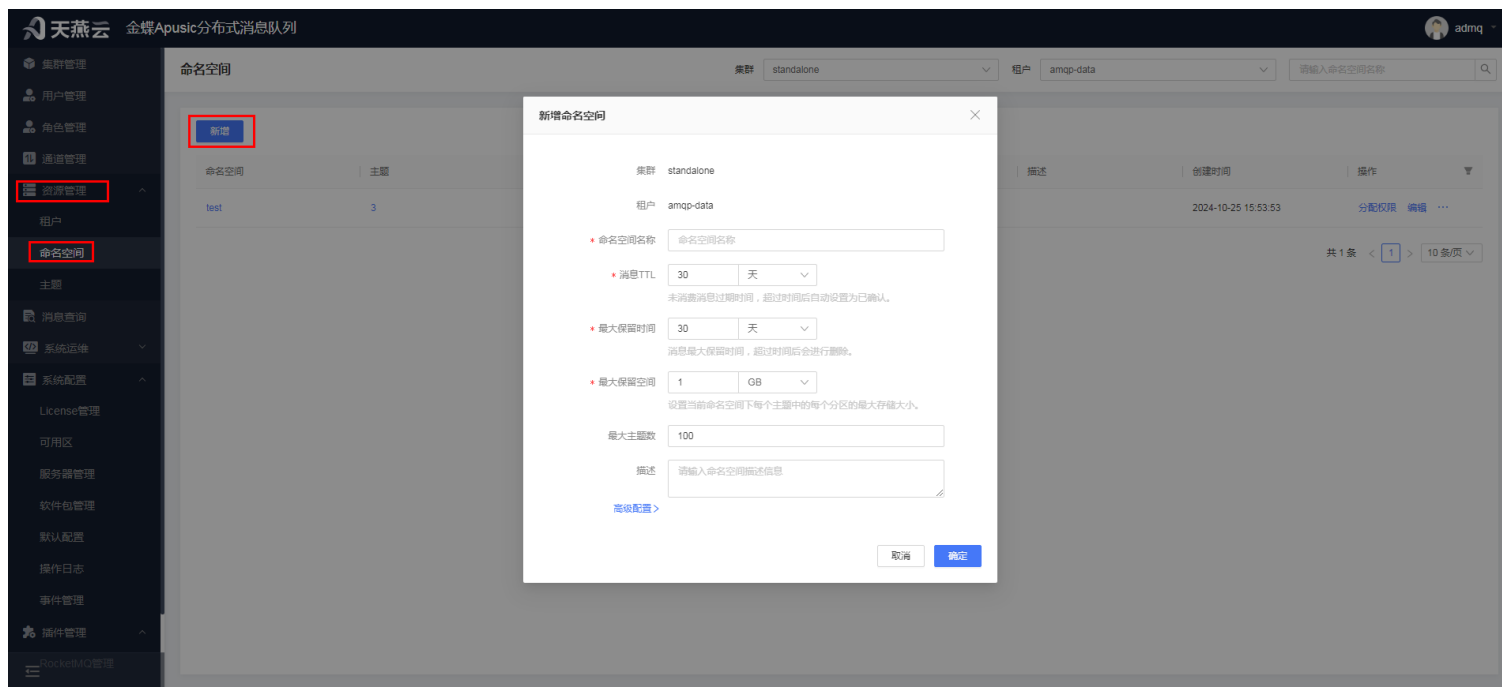
5.6.2 命名空间

命名空间属于租户下，用于新增命名空间信息、编辑命名空间信息、删除命名空间信息、查询命名空间信息、命名空间授权。注意：新增、编辑、删除命名空间之前，需要保证集群服务正常，否则操作将会失败。

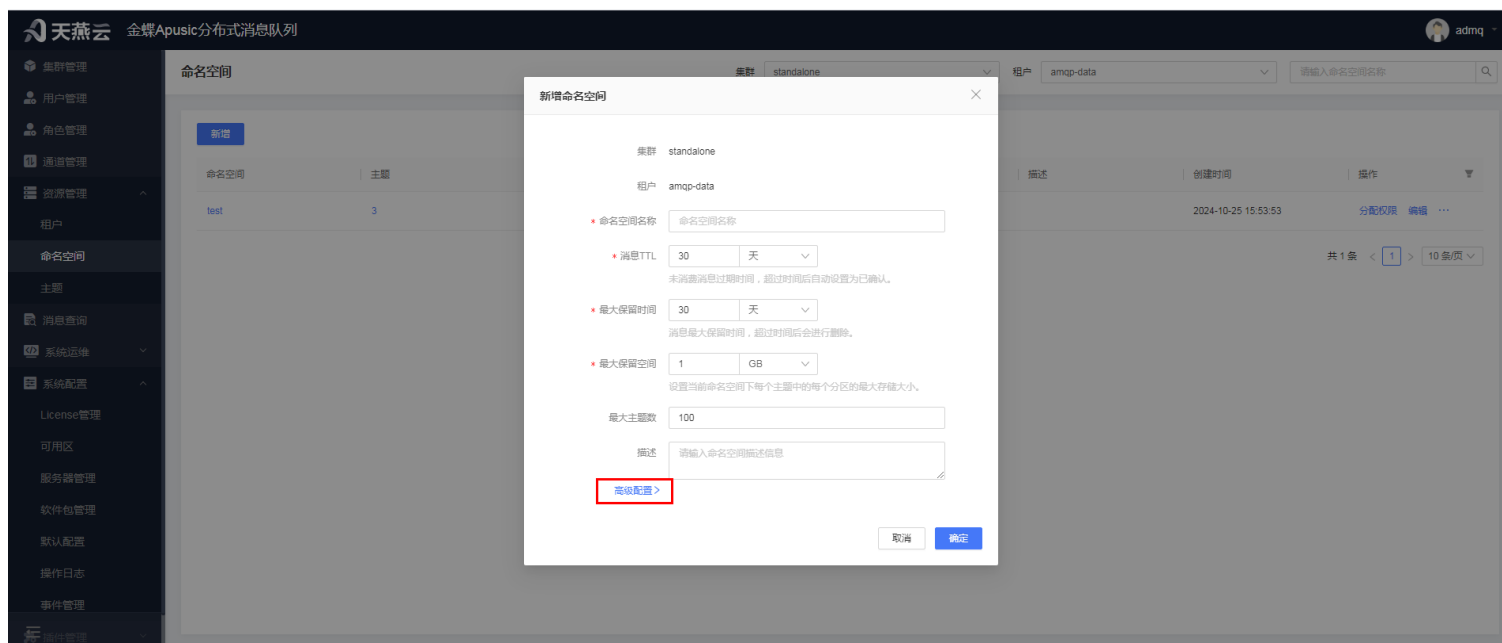


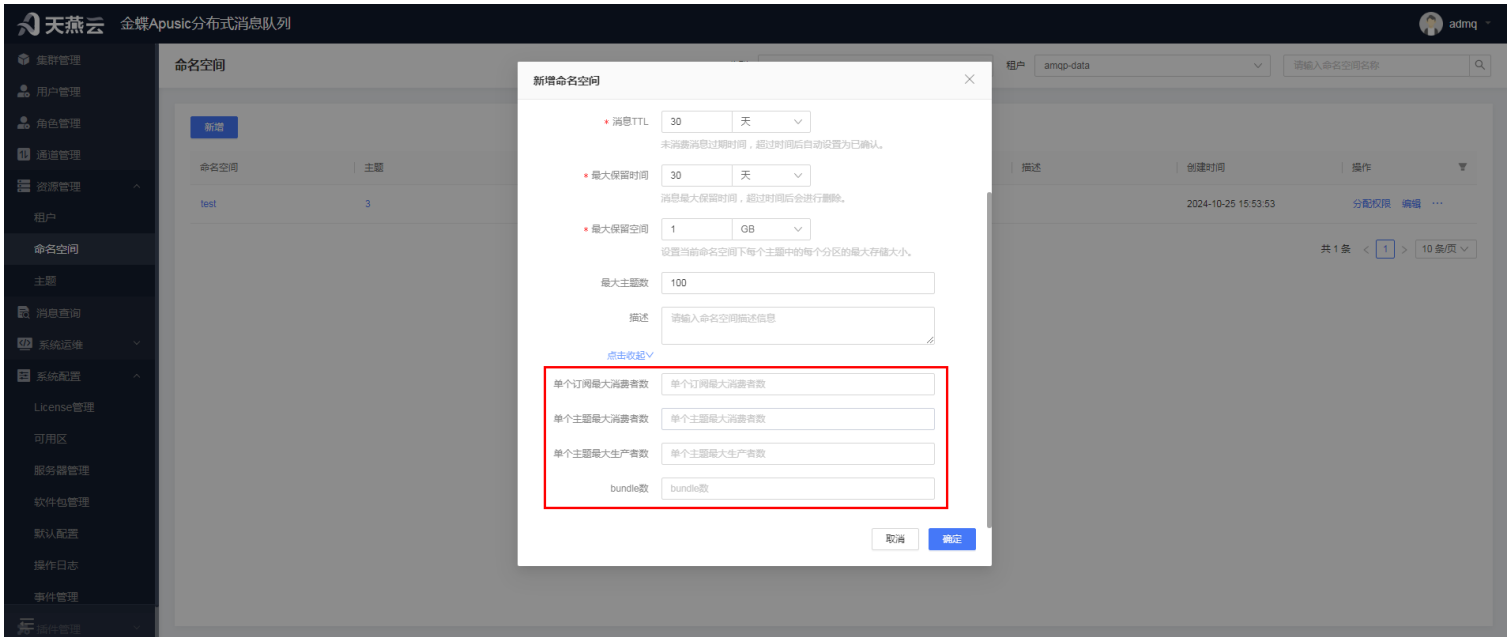
5.6.2.1 新增命名空间

进入【资源管理】->【命名空间】，在命名空间页面左上角点击【新增】按钮，可新增命名空间信息。注意：点击新增前，需在右上角下拉选择需要新增命名空间的集群、租户，点击新增按钮时，将会自动带入选择的集群名称和租户名称。



在弹窗内点击【高级配置】按钮，会出现高级配置选项：





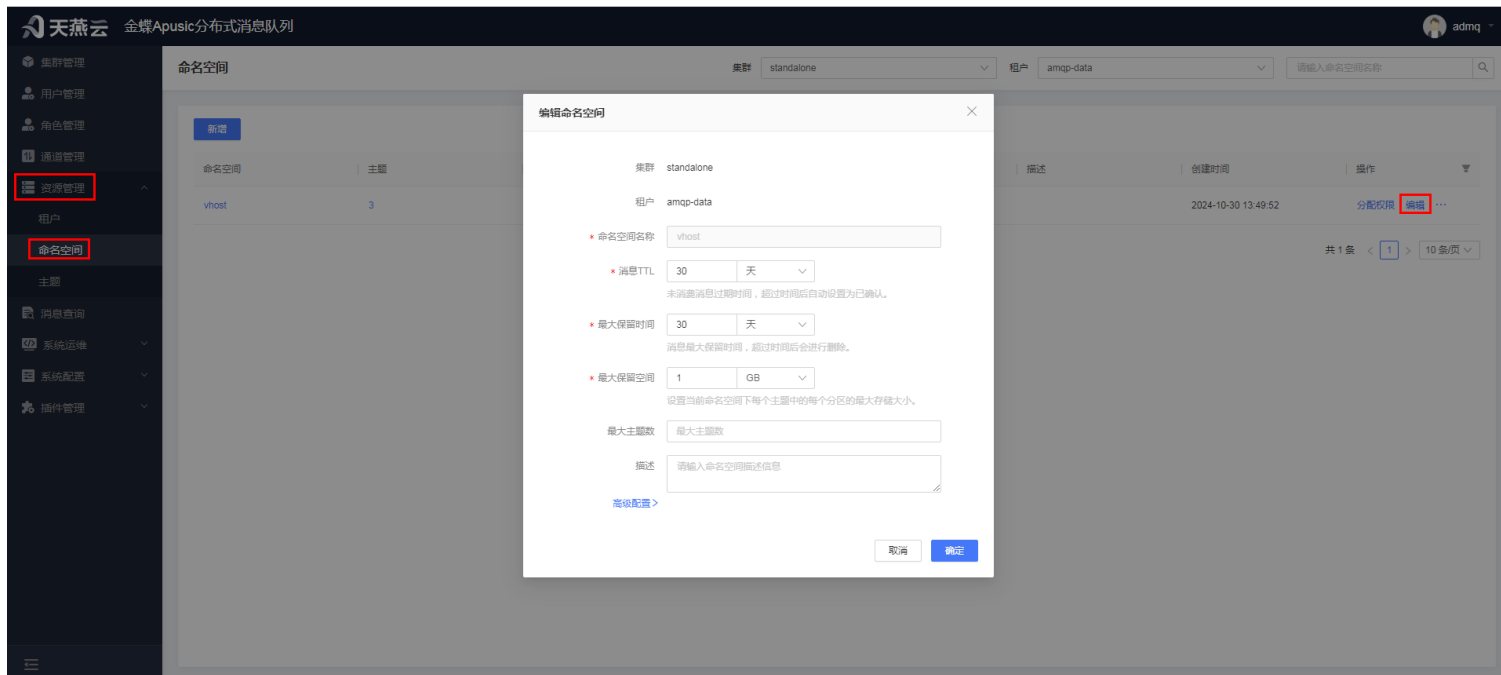
- 集群：租户所属的集群，用户新增前，自行选择，不可修改
- 租户：命名空间所属的租户，用户新增前，自行选择，不可修改
- 命名空间：自定义，唯一性标识
- 消息TTL：未确认的消息多长时间后被标记为已确认，单位：天、小时、分钟，默认30天
- 最大保留时间：已确认的消息多长时间后被删除，单位：天、小时、分钟，默认30天
- 最大保留空间：已确认的消息多少大小后被删除，单位：MB、GB、TB，默认1GB
- 最大主题数：命名空间下最大主题数，超过此大小不可新增主题，默认为100
- 描述：命名空间描述信息

高级配置：

- 单个订阅最大消费者数：命名空间下的主题，1个订阅号只能有多少个消费者接收消息，默认不限制
- 单个主题最大消费者数：命名空间下的主题，1个主题只能有多少个消费者接收消息，默认不限制
- 单个主题最大生产者数：命名空间下的主题，1个主题只能有多少个生产者发送消息，默认不限制
- bundle数：bundle数，不填，默认保存为4个

5.6.2.2 编辑命名空间

进入【资源管理】->【命名空间】，在命名空间页面选择需要修改的命名空间，点击【编辑】按钮，可编辑命名空间信息。注意：在修改租户信息前，默认只显示一个集群一个租户下的命名空间信息，可在右上角下拉选择集群、下拉选择租户。



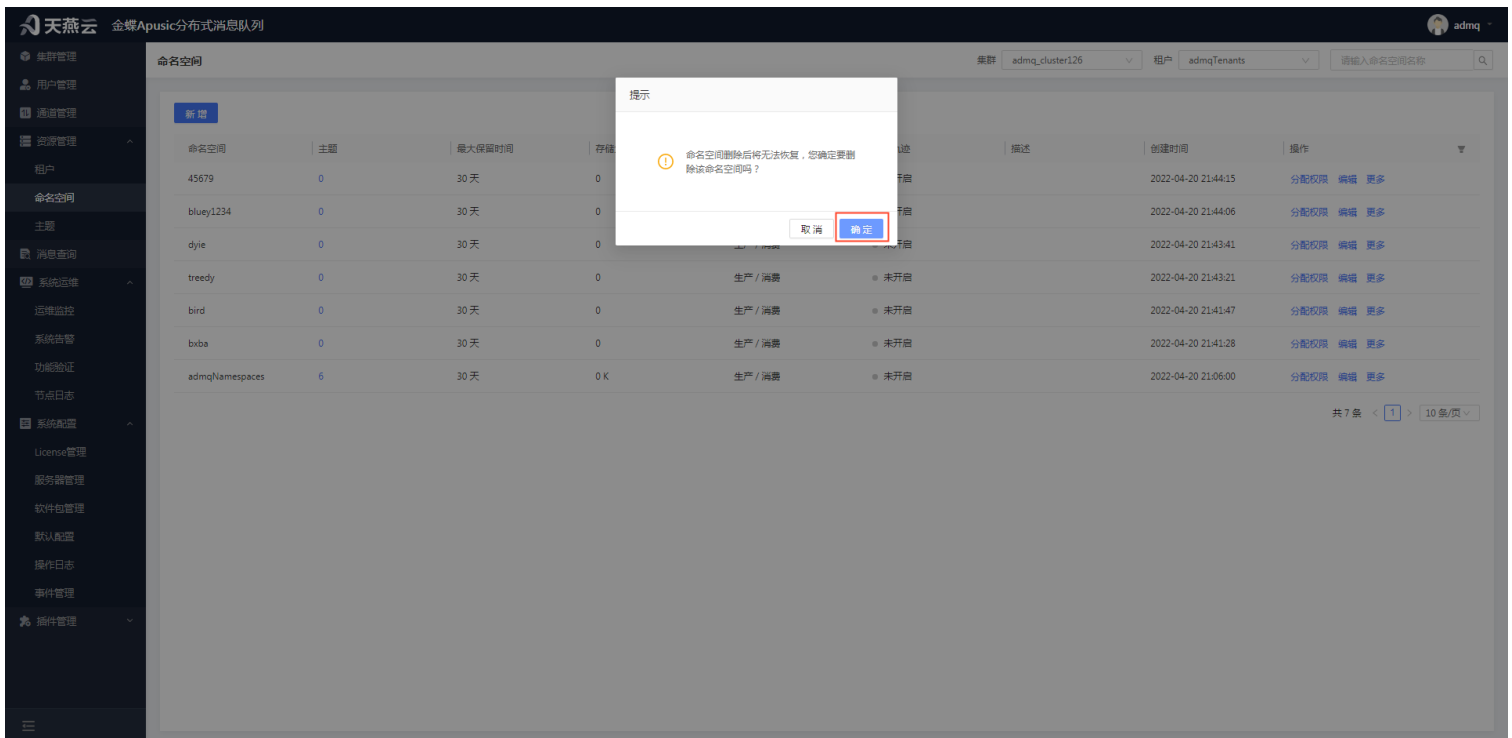
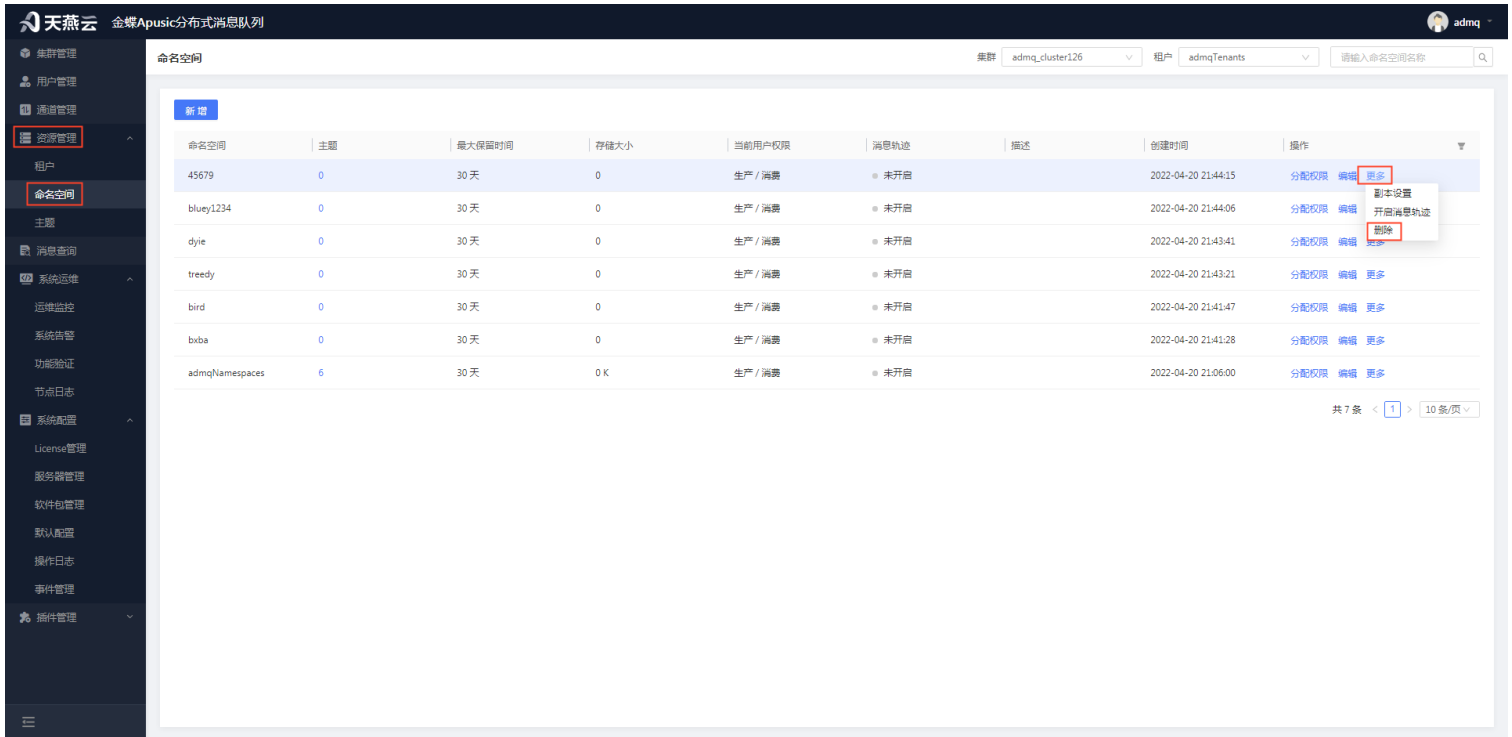
- 集群：不可修改，租户所属的集群
- 租户：不可修改，命名空间所属的租户
- 命名空间名称：新增命名空间时的名称，不可修改，置灰显示
- 消息TTL：未确认的消息多长时间后被标记为已确认，单位：天、小时、分钟，默认30天，可修改
- 最大保留时间：已确认的消息多长时间后被删除，单位：天、小时、分钟，默认30天，可修改
- 最大保留空间：已确认的消息多少大小后被删除，单位：MB、GB、TB，默认1GB，可修改
- 最大主题数：命名空间下最大主题数，超过此大小不可新增主题，默认为100，可修改
- 描述：命名空间描述信息，可修改

高级配置：

- 单个订阅最大消费者数：命名空间下的主题，1个订阅号只能有多少个消费者接收消息，默认不限制，可修改
- 单个主题最大消费者数：命名空间下的主题，1个主题只能有多少个消费者接收消息，默认不限制，可修改
- 单个主题最大生产者数：命名空间下的主题，1个主题只能有多少个生产者发送消息，默认不限制，可修改
- bundle数：bundle数，不填时默认保存为4个，不可修改

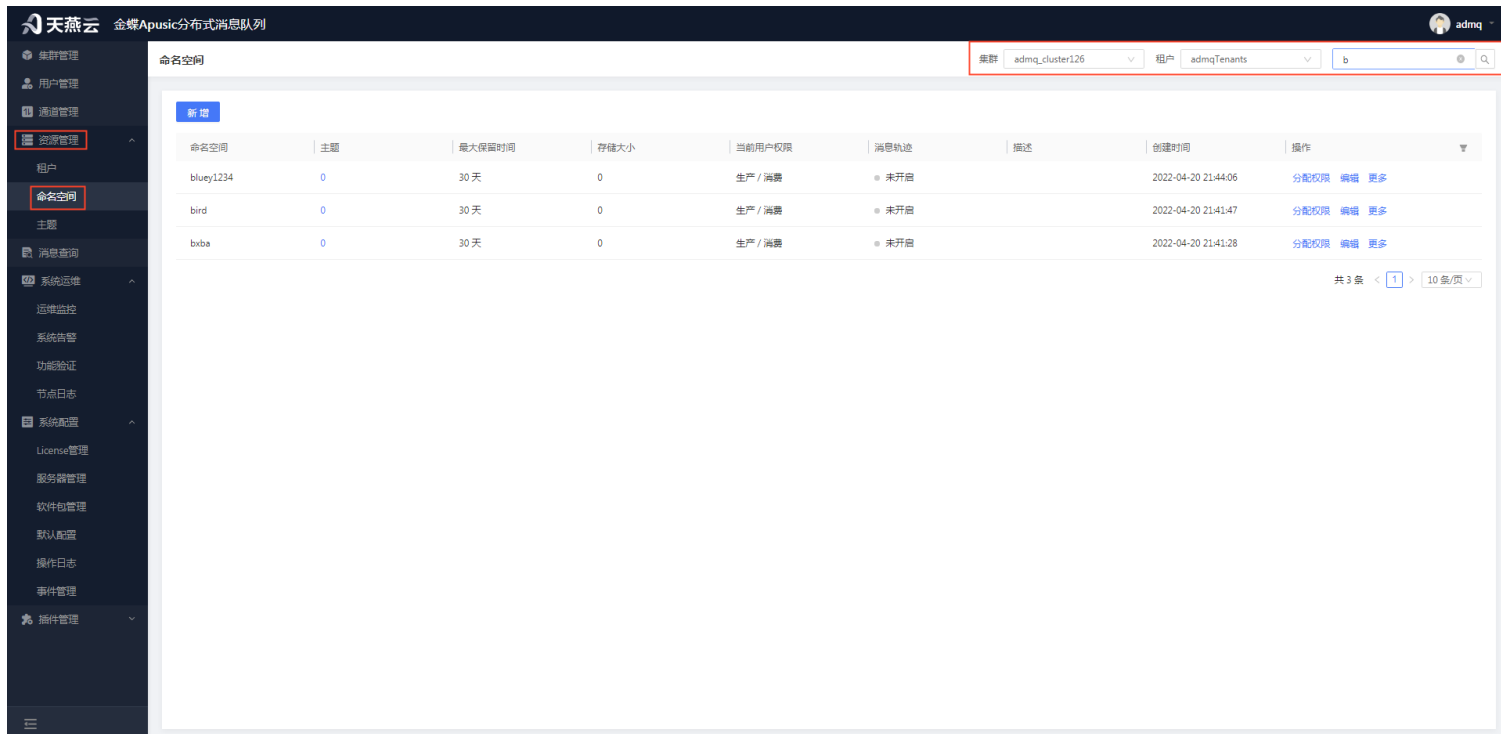
5.6.2.3 删除命名空间

进入【资源管理】->【命名空间】，在命名空间页面选择需要删除的命名空间，点击【更多】按钮->【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除命名空间信息。注意：当删除命名空间信息时，命名空间下的主题资源将会被删除。



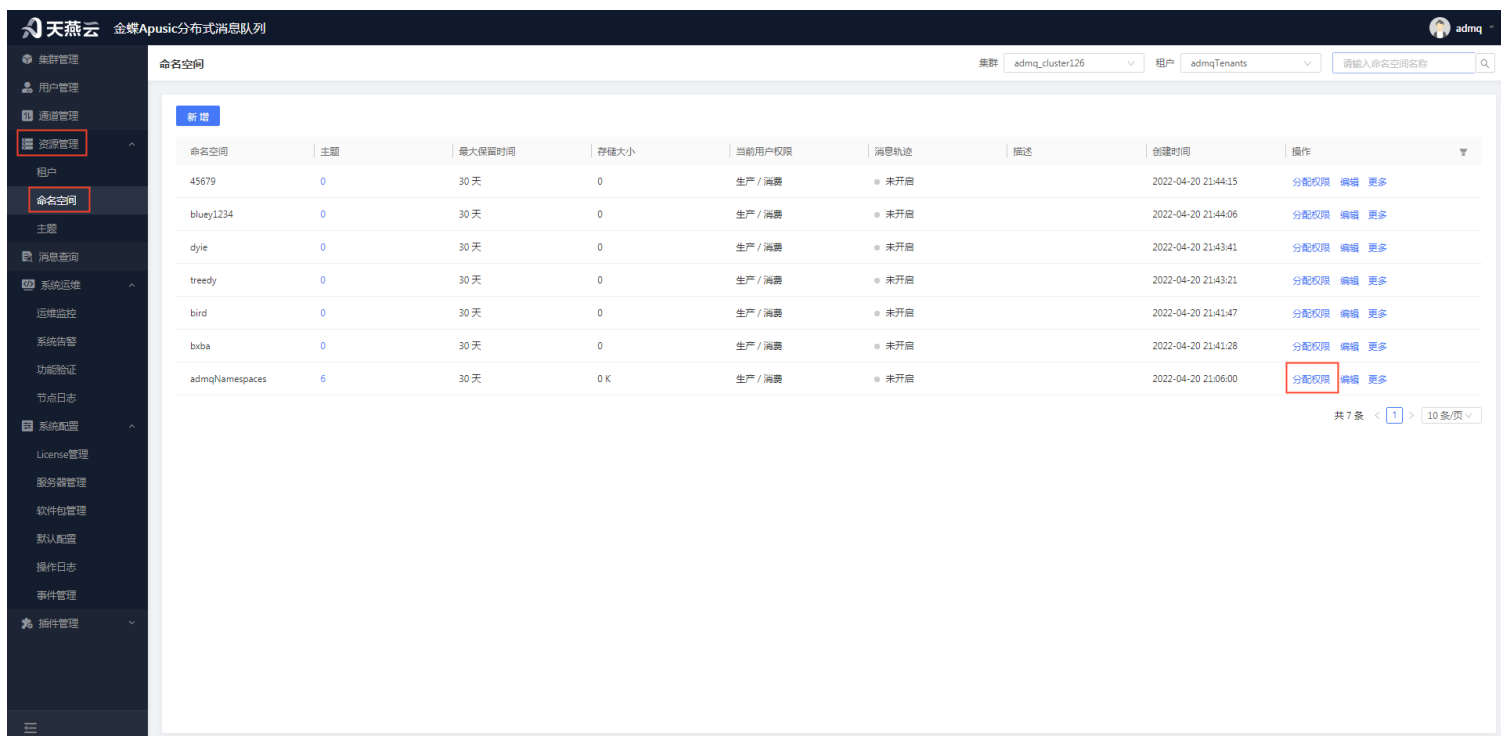
5.6.2.4 查询命名空间

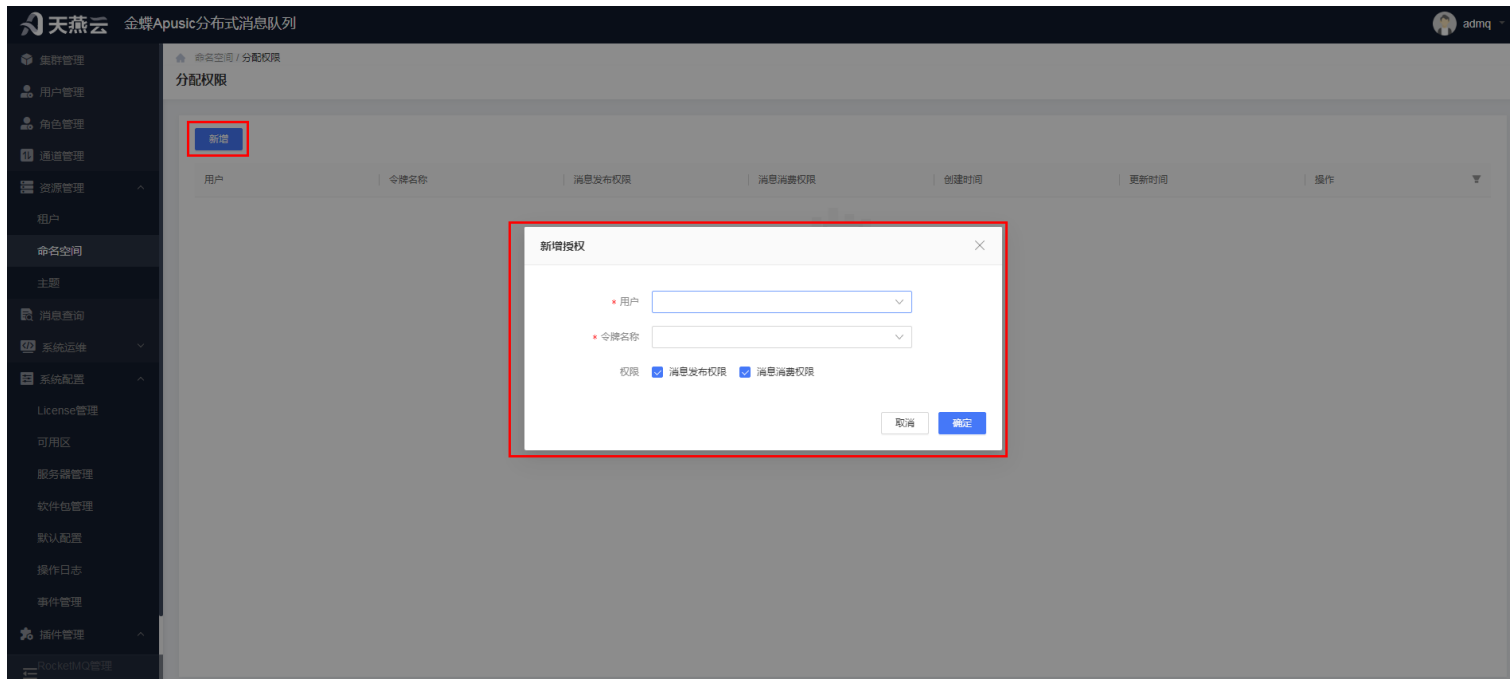
进入【资源管理】->【命名空间】，在命名空间页面选择集群、选择租户，精确或者模糊输入命名空间名，点击【查询】按钮，可查询命名空间信息。



5.6.2.5 新增命名空间权限

进入【资源管理】->【命名空间】，在命名空间页面选择需要分配权限的命名空间，点击【分配权限】按钮，进入分配权限页面；点击【新增】按钮，下拉选择用户，勾选权限前的复选框，可新增命名空间下用户的权限信息。注意：授权主要是给普通用户提供授权，默认超级管理员拥有所有命名空间的消息发布权限和消息消费权限。

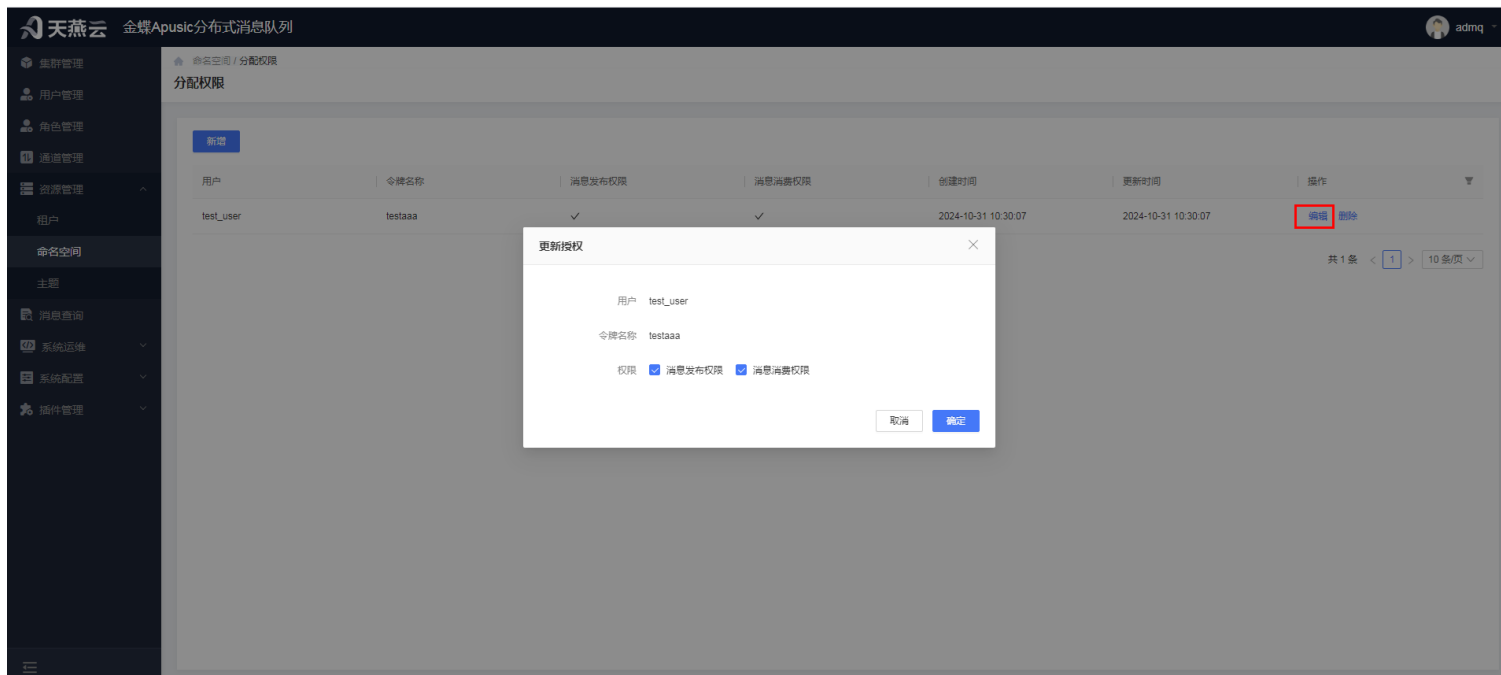
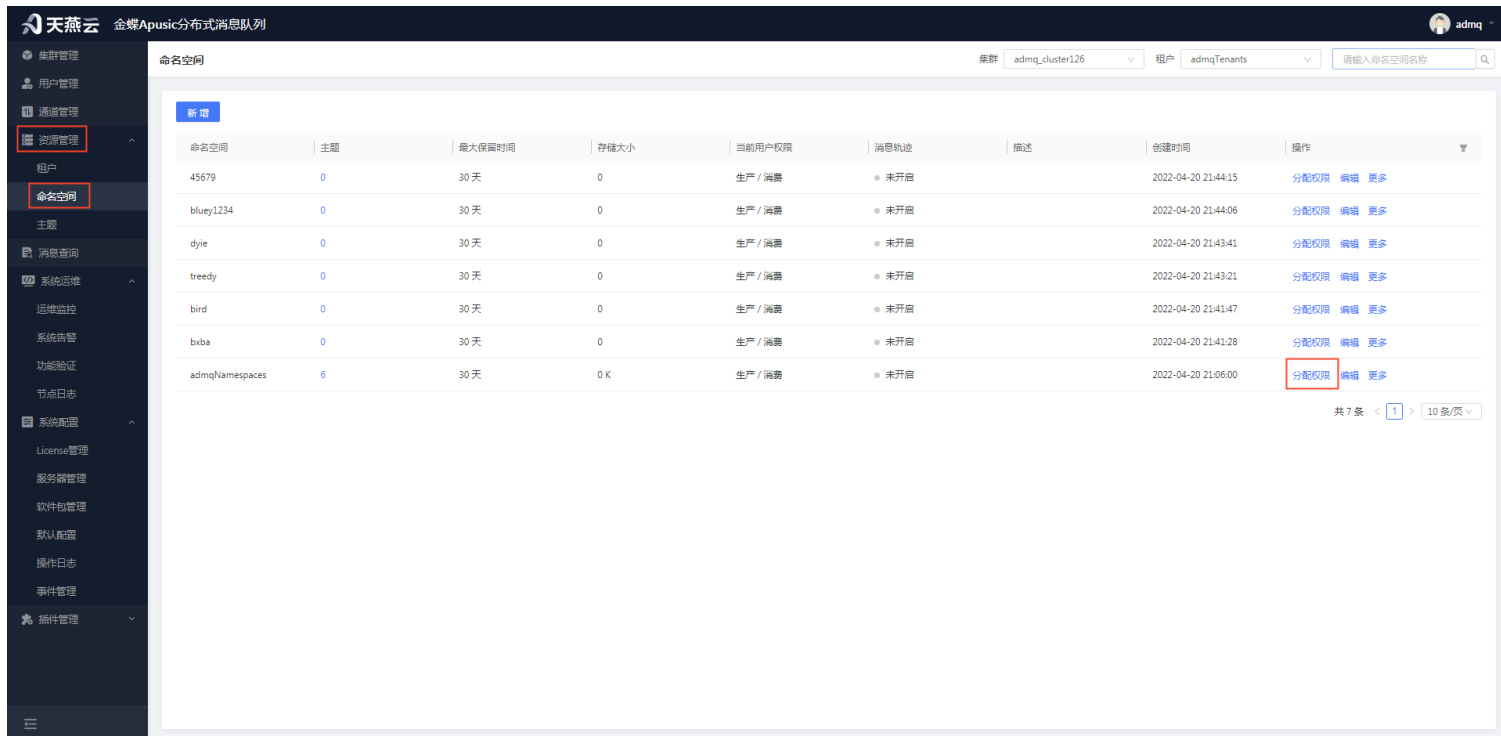




- 用户：下拉选择，获取的用户管理（4.1）的用户信息
- 权限：复选框，消息发布权限和消息消费权限

5.6.2.6 编辑命名空间权限

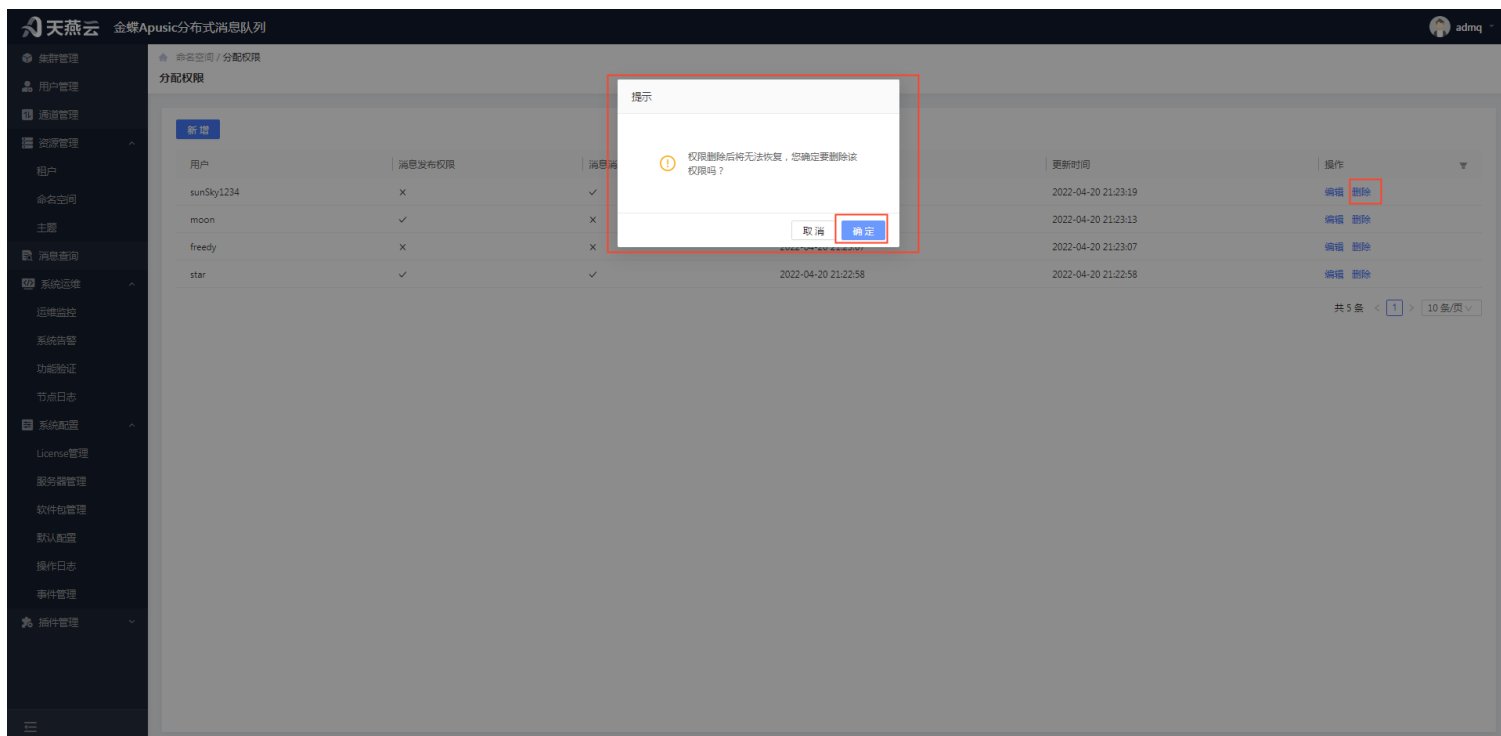
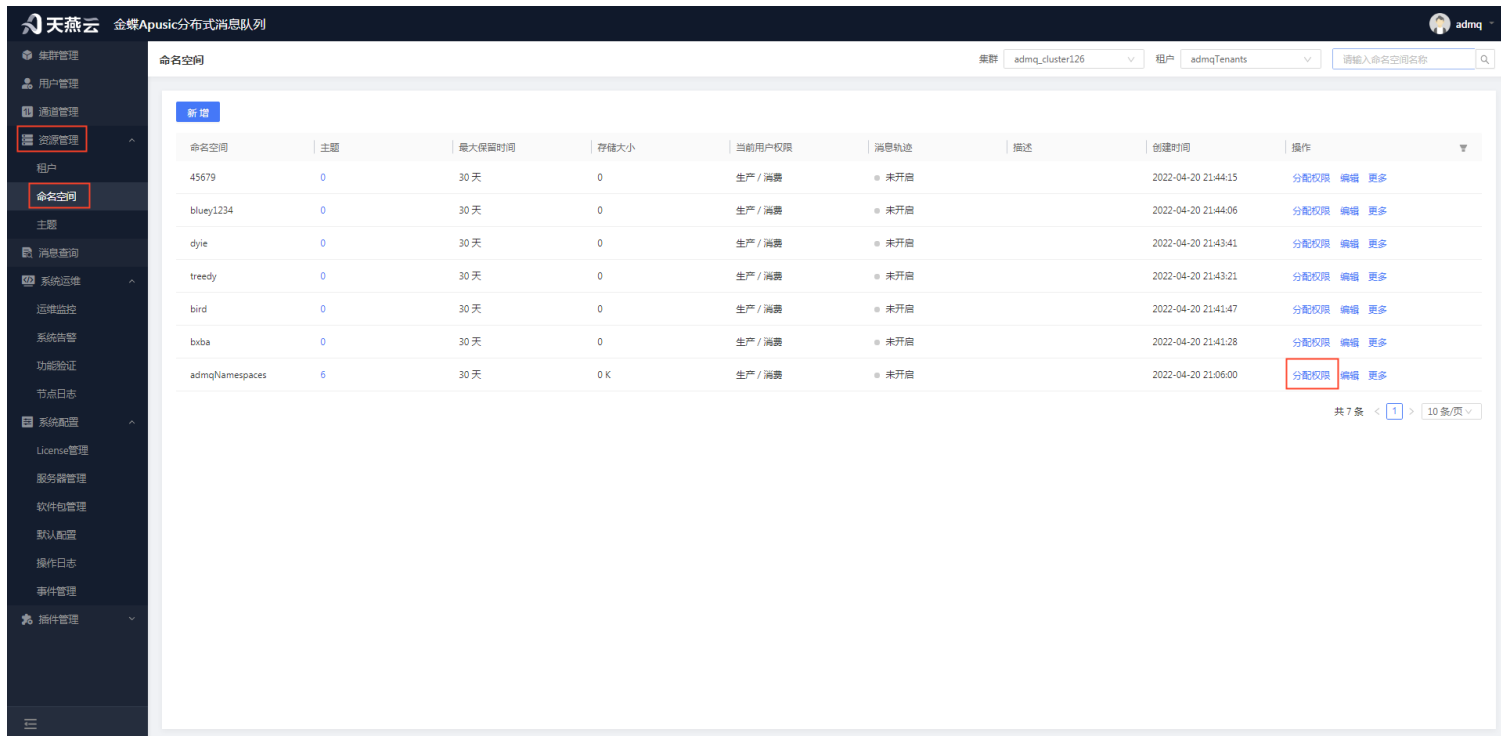
进入【资源管理】->【命名空间】，在命名空间页面选择需要分配权限的命名空间，点击【分配权限】按钮，进入分配权限页面；选择需要更新授权的用户，点击【编辑】按钮，可更新用户在命名空间下的权限。注意：授权主要是给普通用户提供授权，默认超级管理员拥有所有命名空间的消息发布权限和消息消费权限。



- 用户：用户名称，不可修改
- 令牌名称：令牌名称，不可修改
- 权限：复选框，消息发布权限和消息消费权限

5.6.2.7 删除命名空间权限

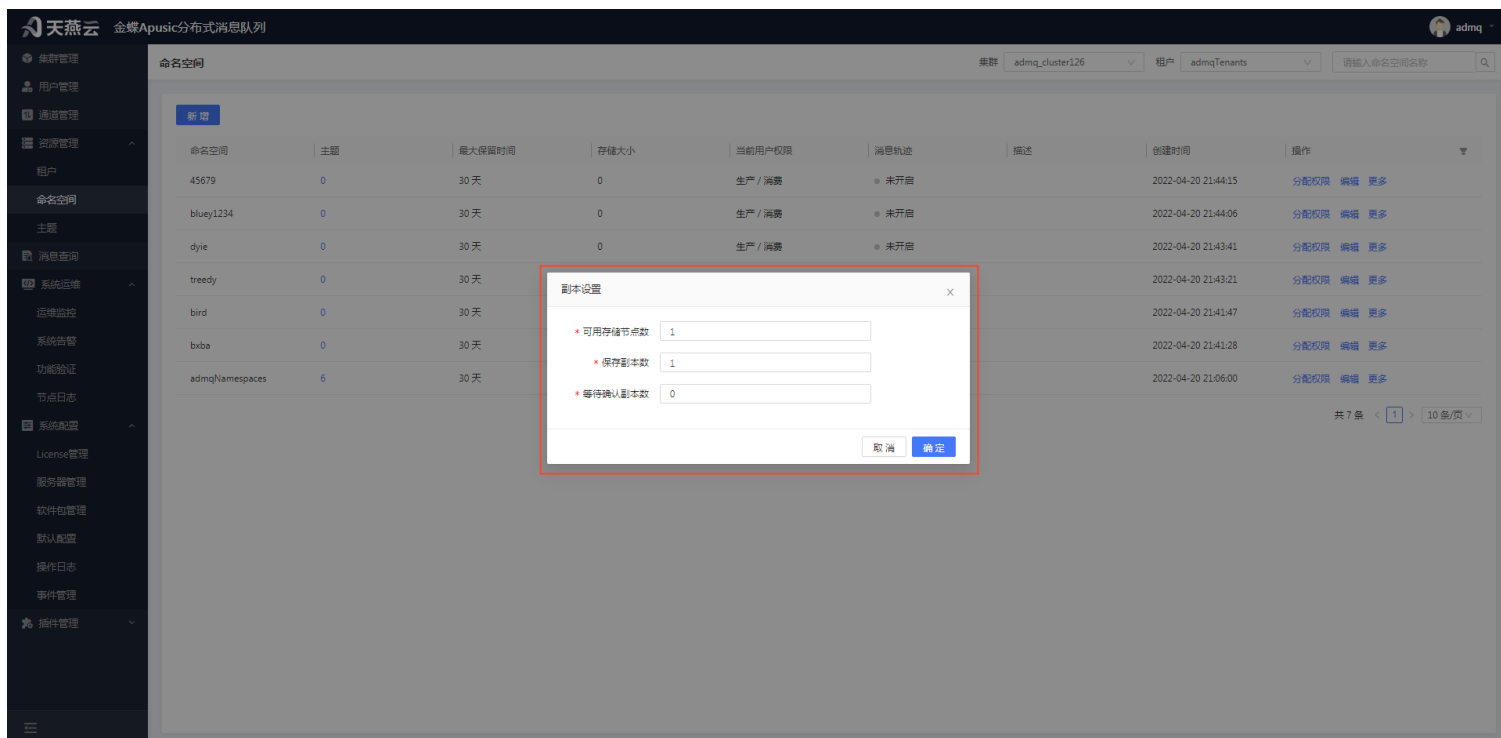
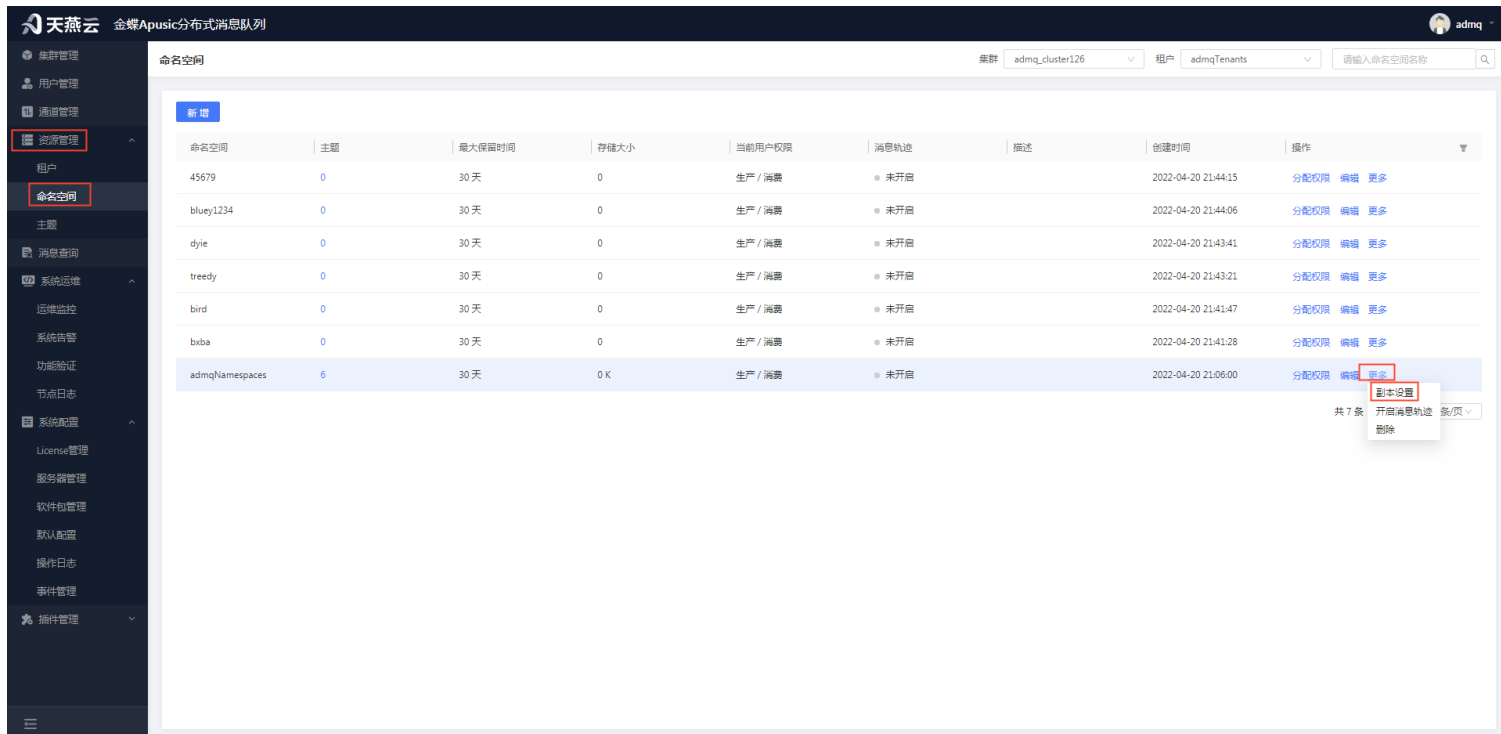
进入【资源管理】->【命名空间】，在命名空间页面选择需要分配权限的命名空间，点击【分配权限】按钮，进入分配权限页面；选择需要删除授权的用户，点击【删除】按钮，在弹出的确认删除弹框点击【确定】按钮，可删除用户在命名空间下的权限。注意：授权主要是给普通用户提供授权，默认超级管理员拥有所有命名空间的消息发布权限和消息消费权限。



- 备注：删除用户后，使用用户的token发送接收消息时，会提示没有权限。

5.6.2.8 副本设置

进入【资源管理】->【命名空间】，在命名空间页面选择要设置副本的命名空间，点击【更多】按钮->【副本设置】按钮，可设置副本。注意：设置副本前，需确保集群服务正常。

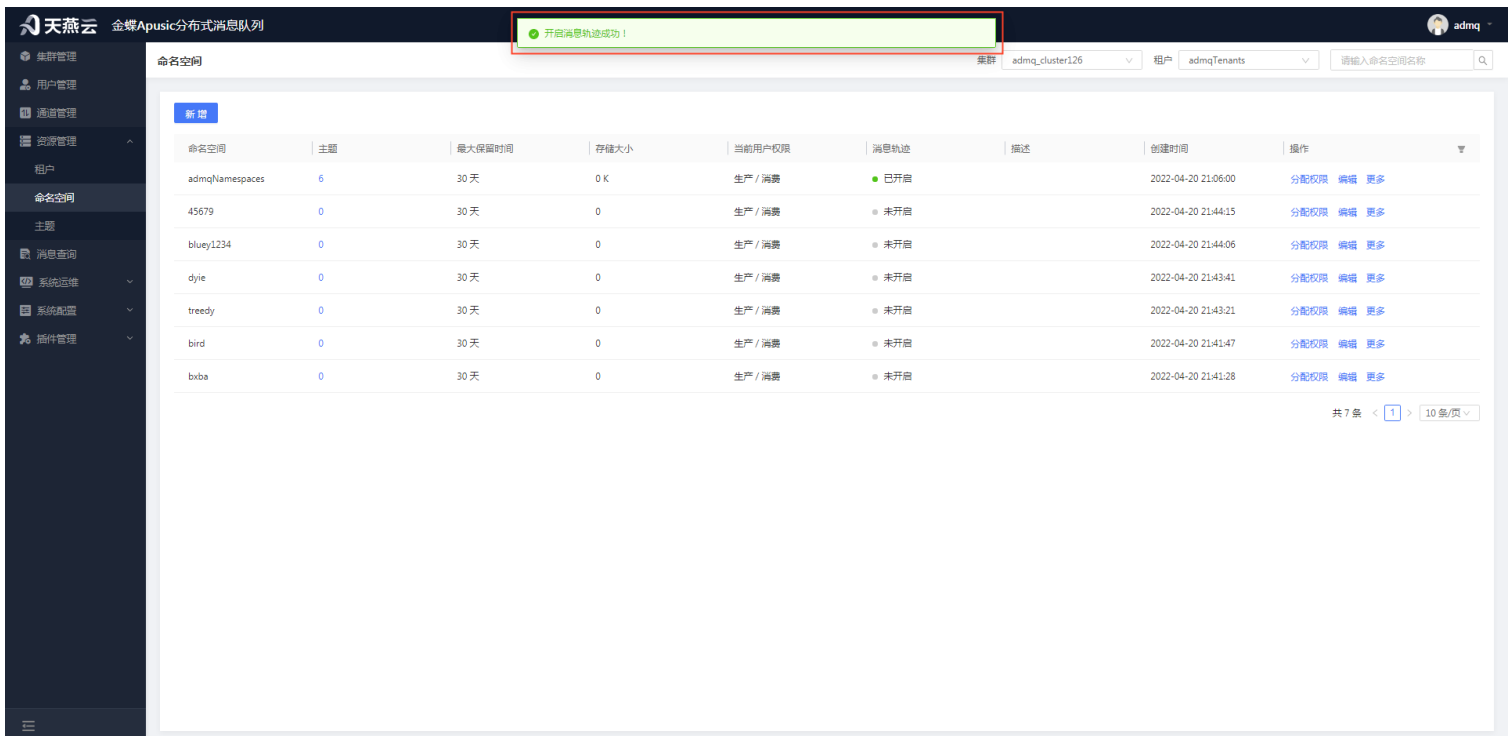
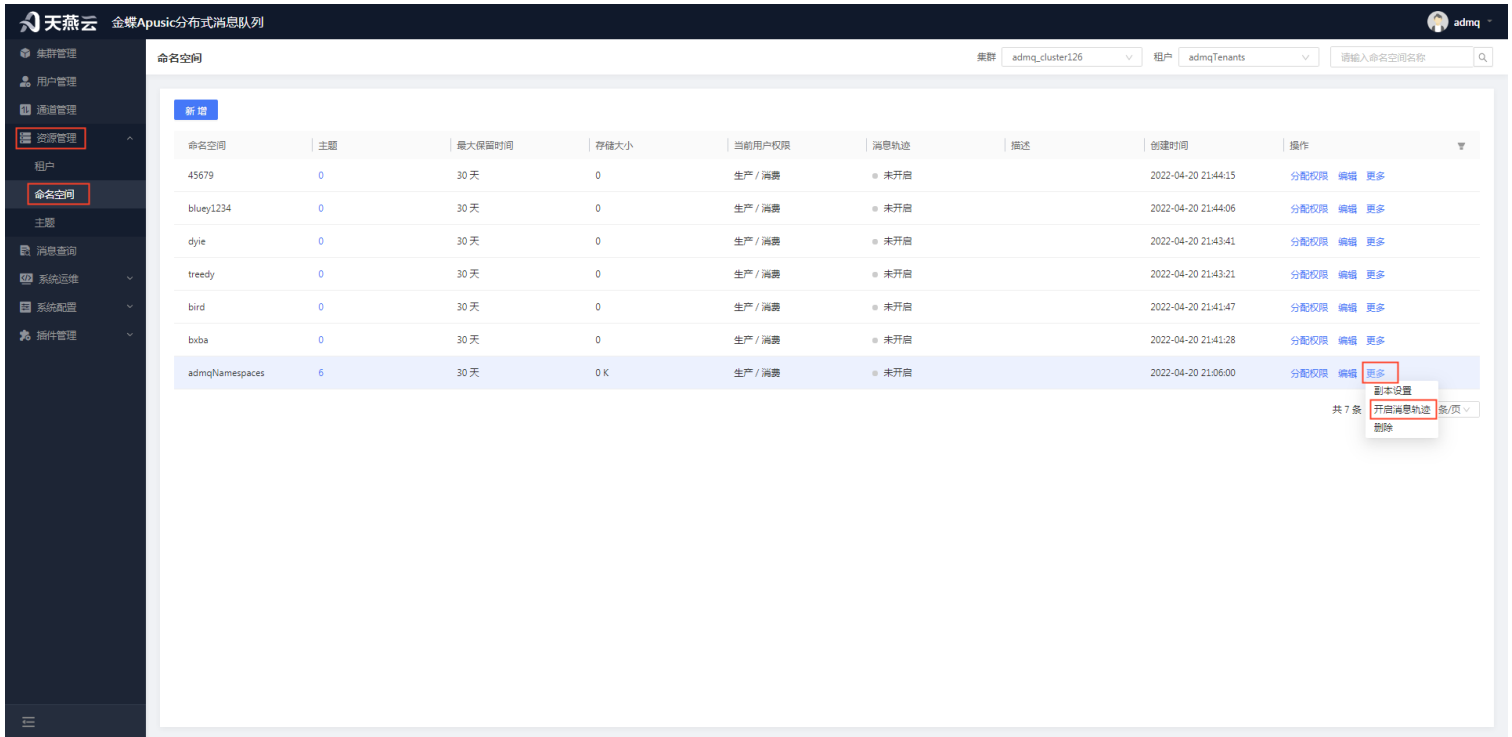


可用存储节点数：可用的存储节点数，自定义，不能大于租户所属集群的可用存储节点数

- 保存副本数：消息保存副本数，自定义，不能大于可用存储节点个数，且不小于等待确认副本数
- 等待确认副本数：至少等到多少个副本存储成功，自定义，不可大于可用存储节点个数和保存副本数

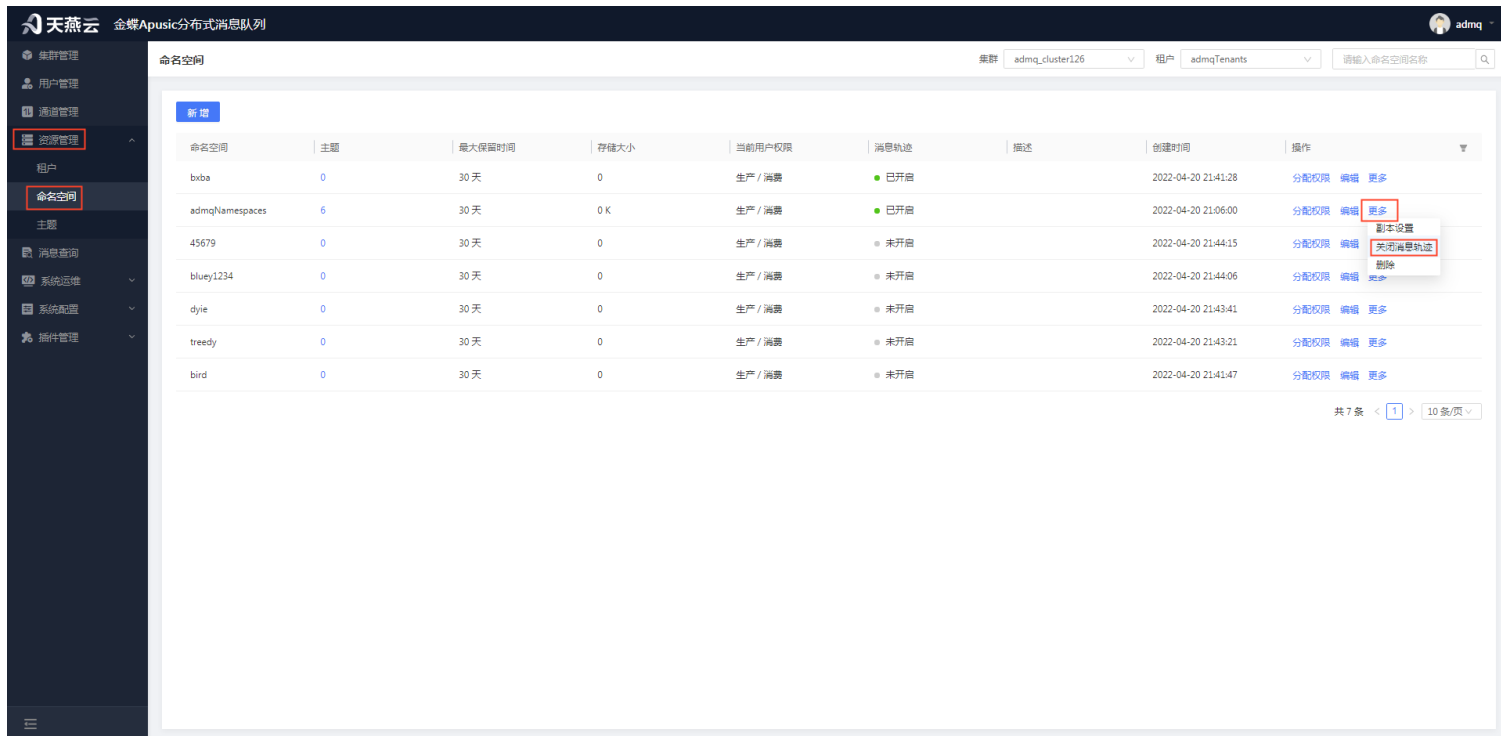
5.6.2.9 开启消息轨迹

主题的消息轨迹默认不会开启，若消息查询需查询消息轨迹，则需要为主题所属命名空间开启消息轨迹。进入【资源管理】->【命名空间】，在命名空间页面选择要开启轨迹的命名空间，点击【更多】按钮->【开启消息轨迹】按钮，可开启消息轨迹，开启成功后会弹出“开启消息轨迹成功！”提示。注意：开启消息轨迹后再创建的主题，才能生效。



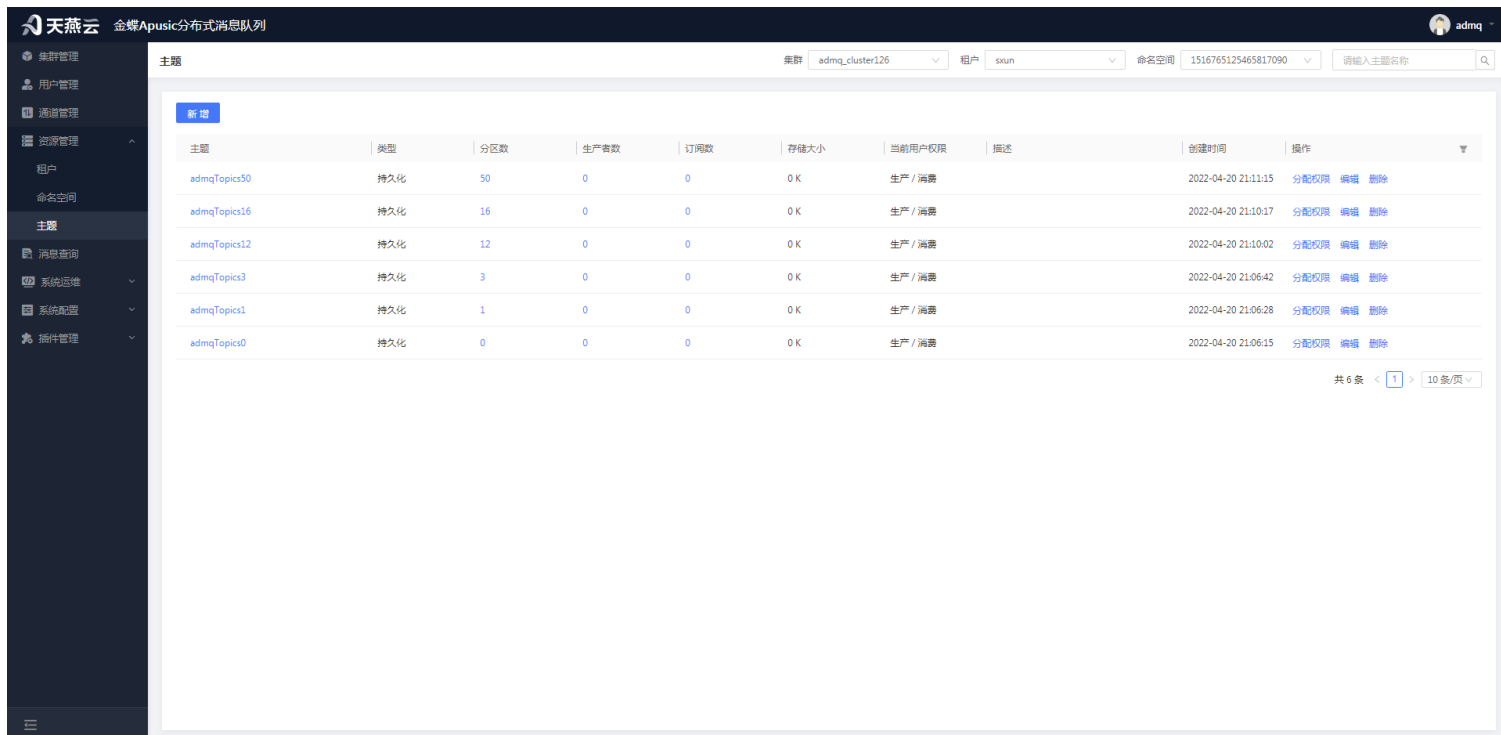
5.6.2.10 关闭消息轨迹

开启的消息轨迹，若需要关闭消息轨迹，则需要为主题所属命名空间关闭消息轨迹。进入【资源管理】->【命名空间】，在命名空间页面选择要关闭轨迹的命名空间，点击【更多】按钮->【关闭消息轨迹】按钮，可关闭消息轨迹。注意：关闭消息轨迹后再创建的主题，才能生效。



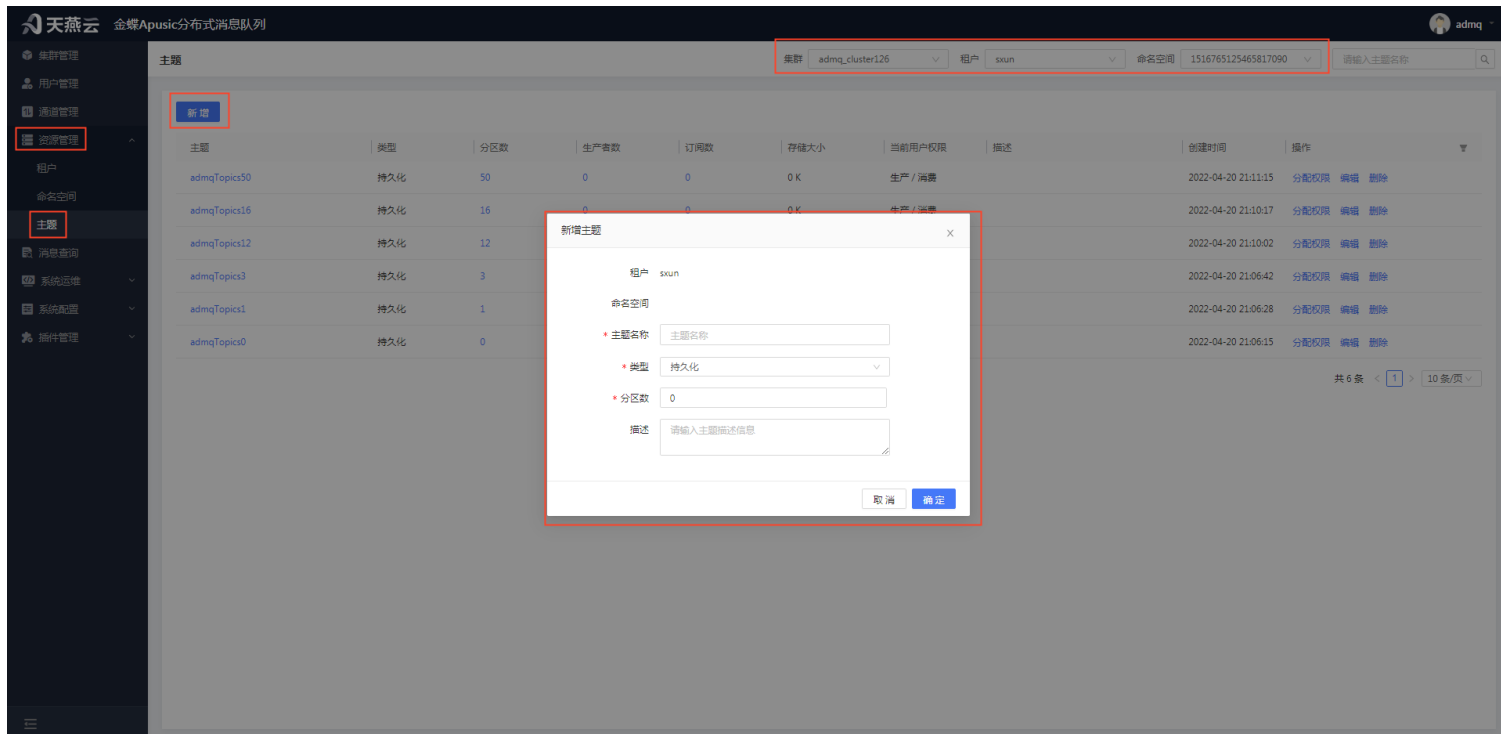
5.6.3 主题

主题属于租户下，用于新增主题信息、编辑主题信息、删除主题信息、查询主题信息。注意：新增、编辑、删除主题时，需要保证集群服务正常，否则操作将会失败。



5.6.3.1 新增主题

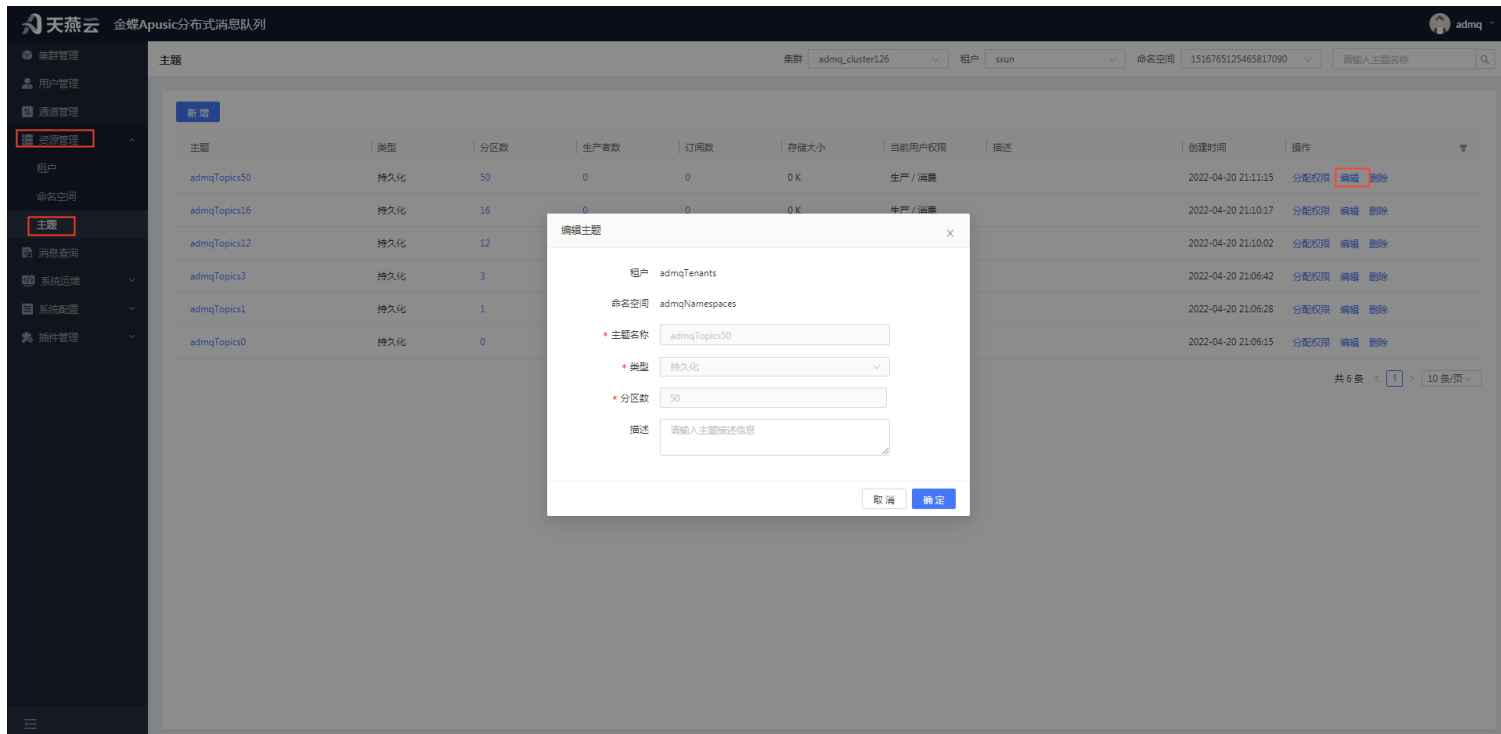
进入【资源管理】->【主题】，在主题页面左上角点击【新增】按钮，可新增主题信息。注意：点击新增前，需在右上角下拉选择需要新增命名空间的集群、租户、命名空间，点击新增按钮时，将会自动带入选择的集群名称和租户名称、命名空间名称。



- 租户：主题所属租户名称，不可修改，
- 命名空间：主题所属命名空间名称，不可修改
- 主题名称：自定义，唯一性标识，不同租户下的主题可重复
- 类型：主题的类型，下拉选择，持久化和非持久化，默认持久化
- 分区数：主题的分区数量，默认为0
- 描述：主题描述信息

5.6.3.2 编辑主题

进入【资源管理】->【主题】，在主题页面选择需要修改的主题名称，点击【编辑】按钮，可编辑主题信息。注意：在修改主题信息前，默认只显示一个集群一个租户下一个命名空间下的的主题信息，可在右上角下拉选择集群、下拉选择租户、下拉选择命名空间。

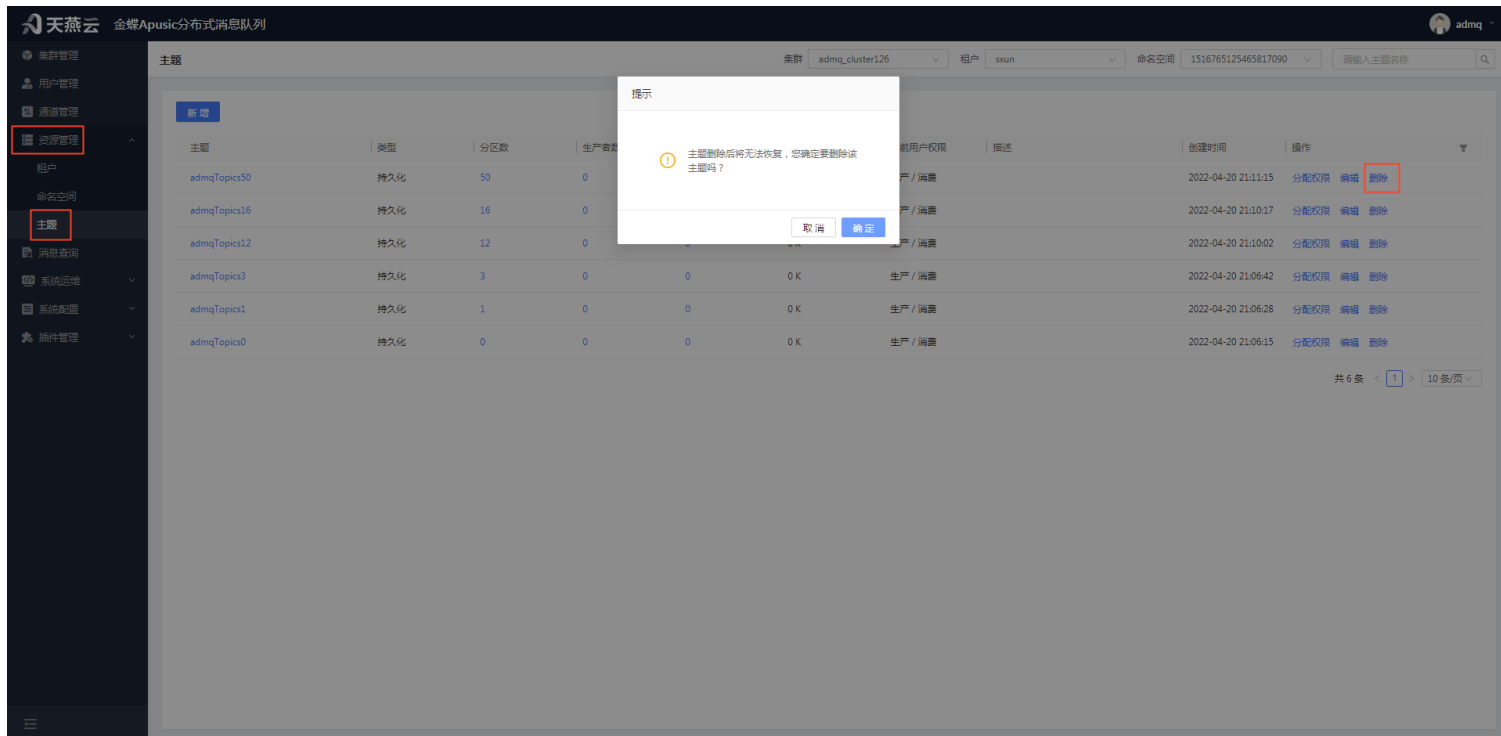


- 租户：主题所属租户名称，不可修改
- 命名空间：主题所属命名空间名称，不可修改
- 主题名称：主题的名称，不可修改，
- 类型：主题的类型，置灰显示，不可修改，
- 分区数：主题的分区数量，置灰显示，不可修改，
- 描述：主题描述信息，可修改

5.6.3.3 删除主题

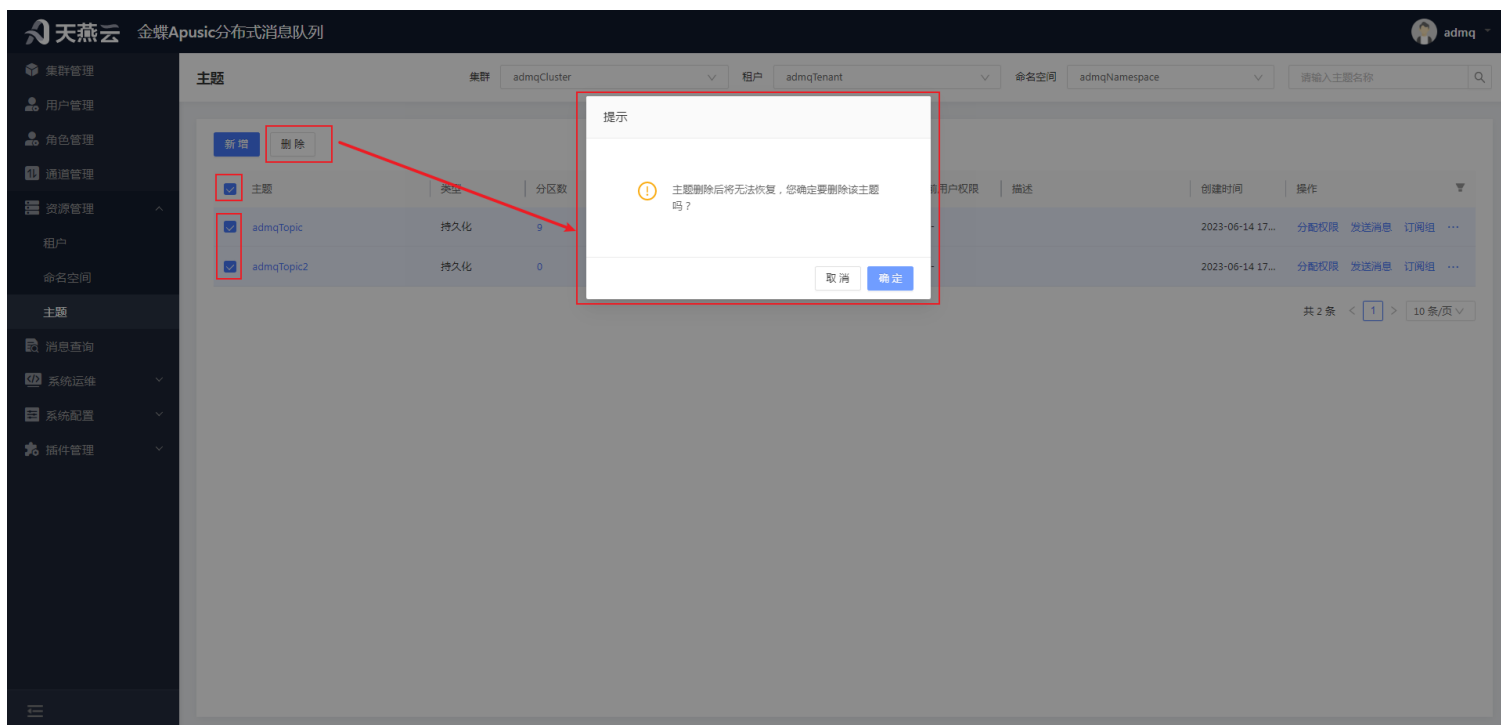
(1) 单个主题删除

进入【资源管理】->【主题】，在主题页面选择需要删除的主题，点击【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除命名主题信息。



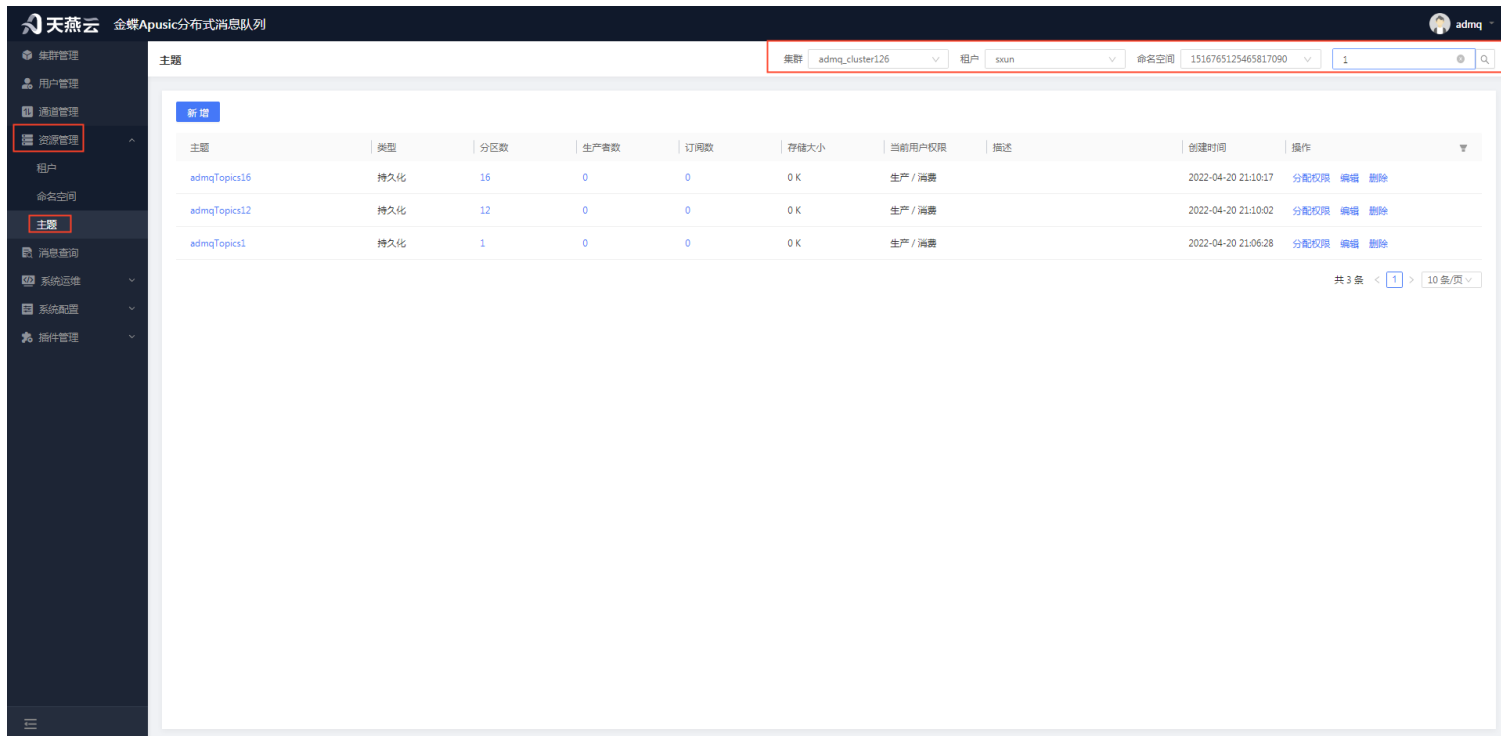
(2) 批量主题删除

进入【资源管理】->【主题】，在主题页面选择需要删除的主题，选中左侧的单选框，然后点击右上角的【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可批量删除主题。备注：最左侧主题左边的单选框为全选。



5.6.3.4 查询主题

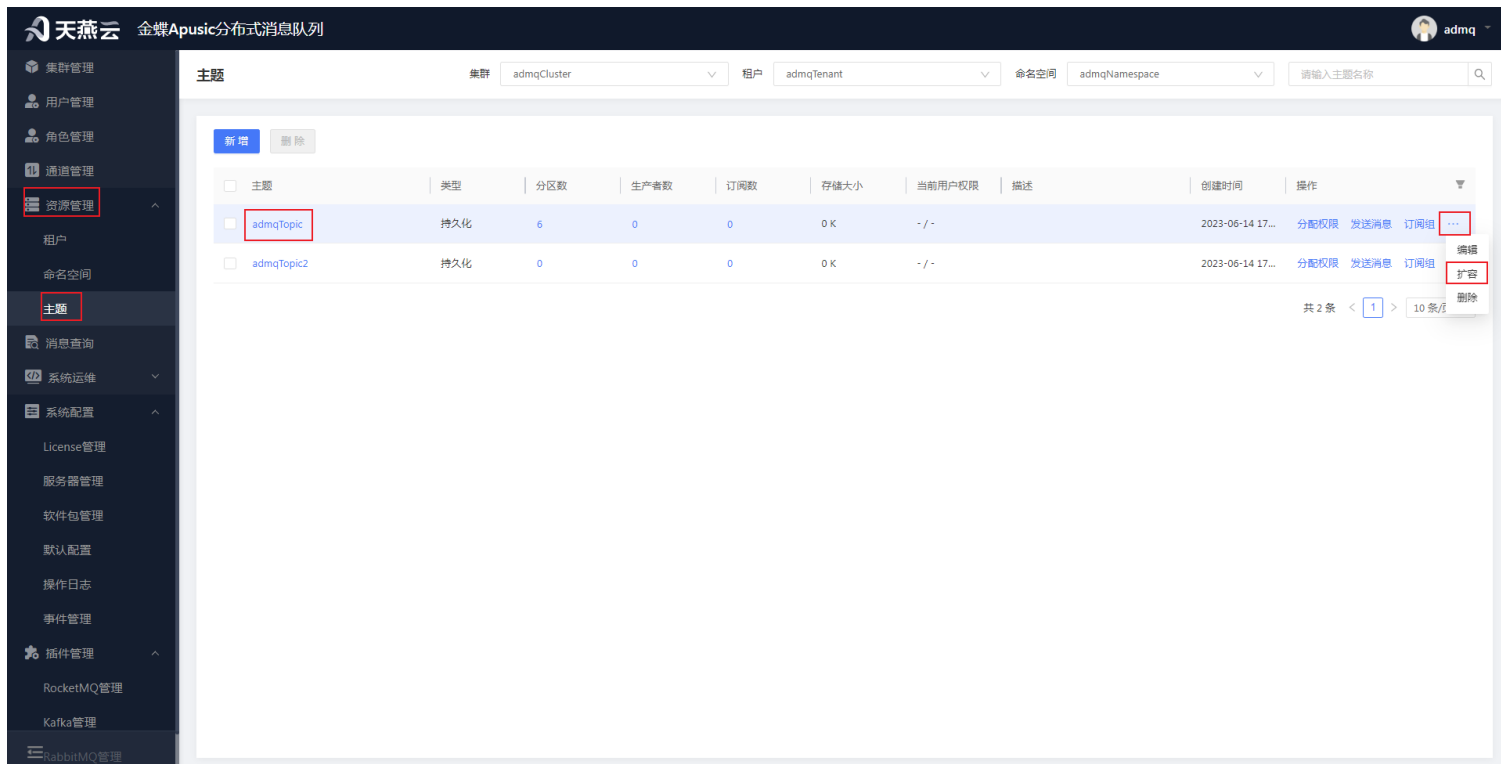
进入【资源管理】->【主题】，在命名空间页面右上角选择集群、选择租户、选择命名空间、精确或者模糊输入主题名称，点击【查询】按钮，可查询主题。

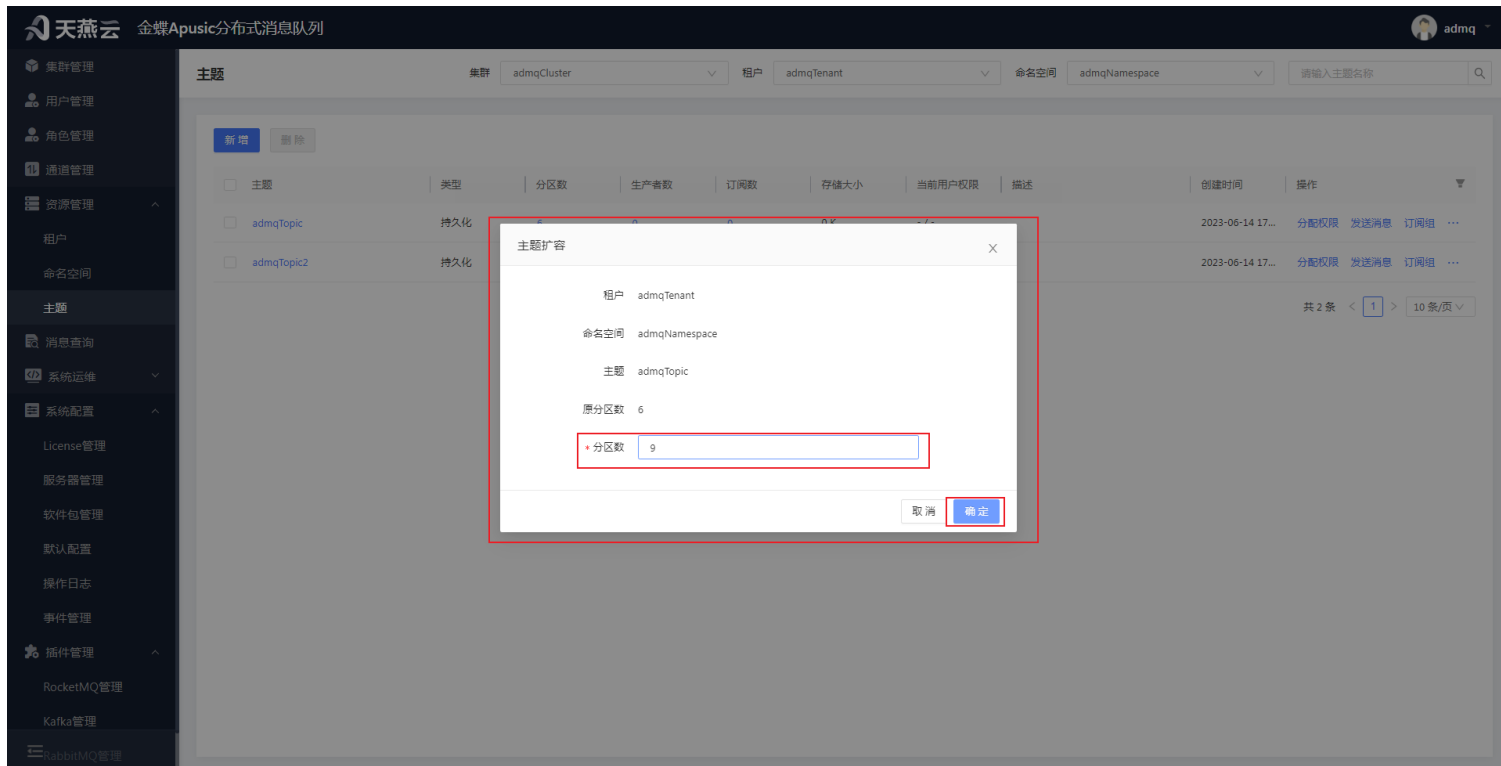


5.6.3.5 主题分区扩容

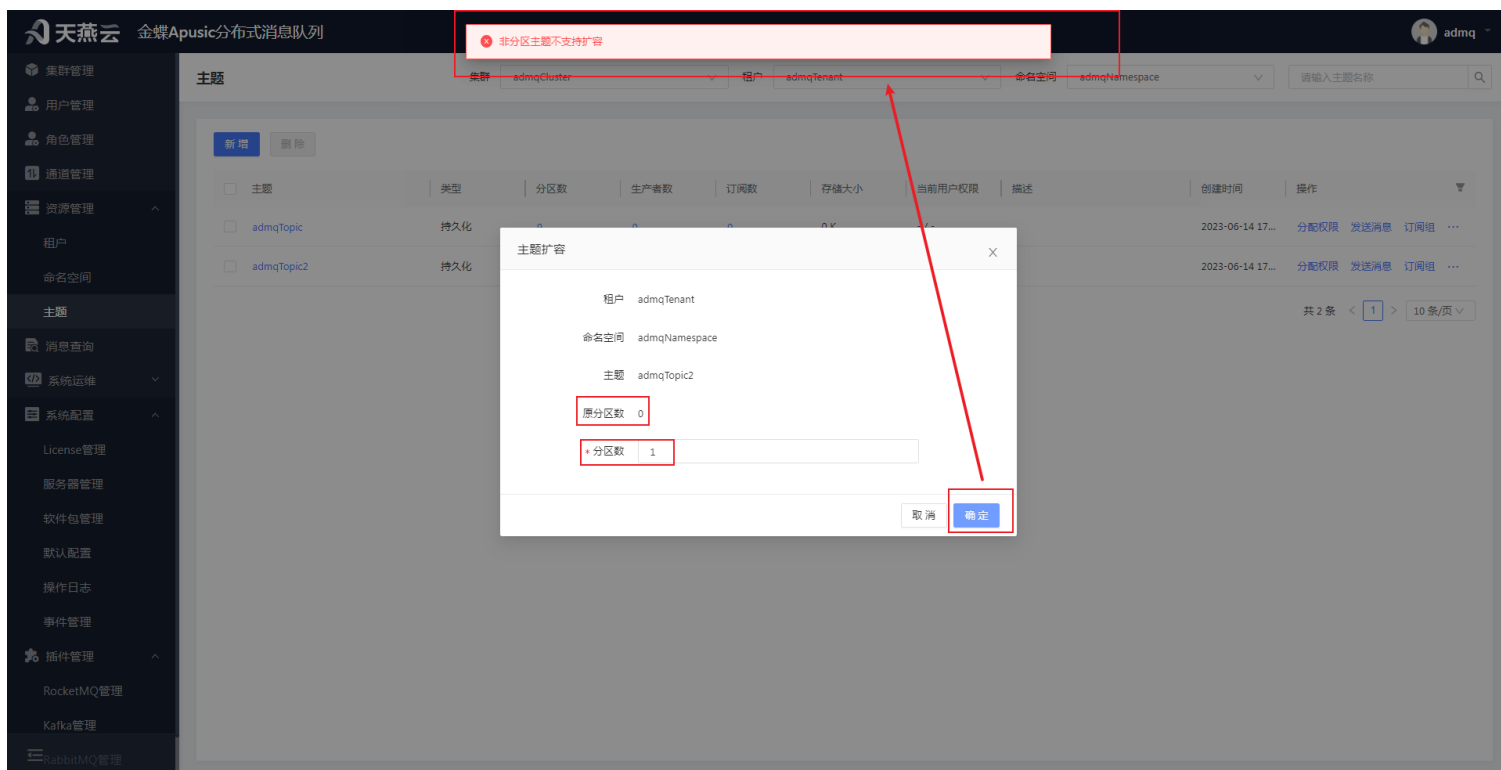
在ADMQ中，主题的分区数是在创建主题时指定的，通过扩容功能可增加已创建主题的分区数，以满足不断变化的负载需求。具体扩容步骤：

通过【资源管理】—【主题】，进入到主题列表页面，选择需要扩容的主题，点击【...】—【扩容】按钮，出现主题扩容弹窗，输入新的分区数，点击【确定】按钮，进行主题分区扩容



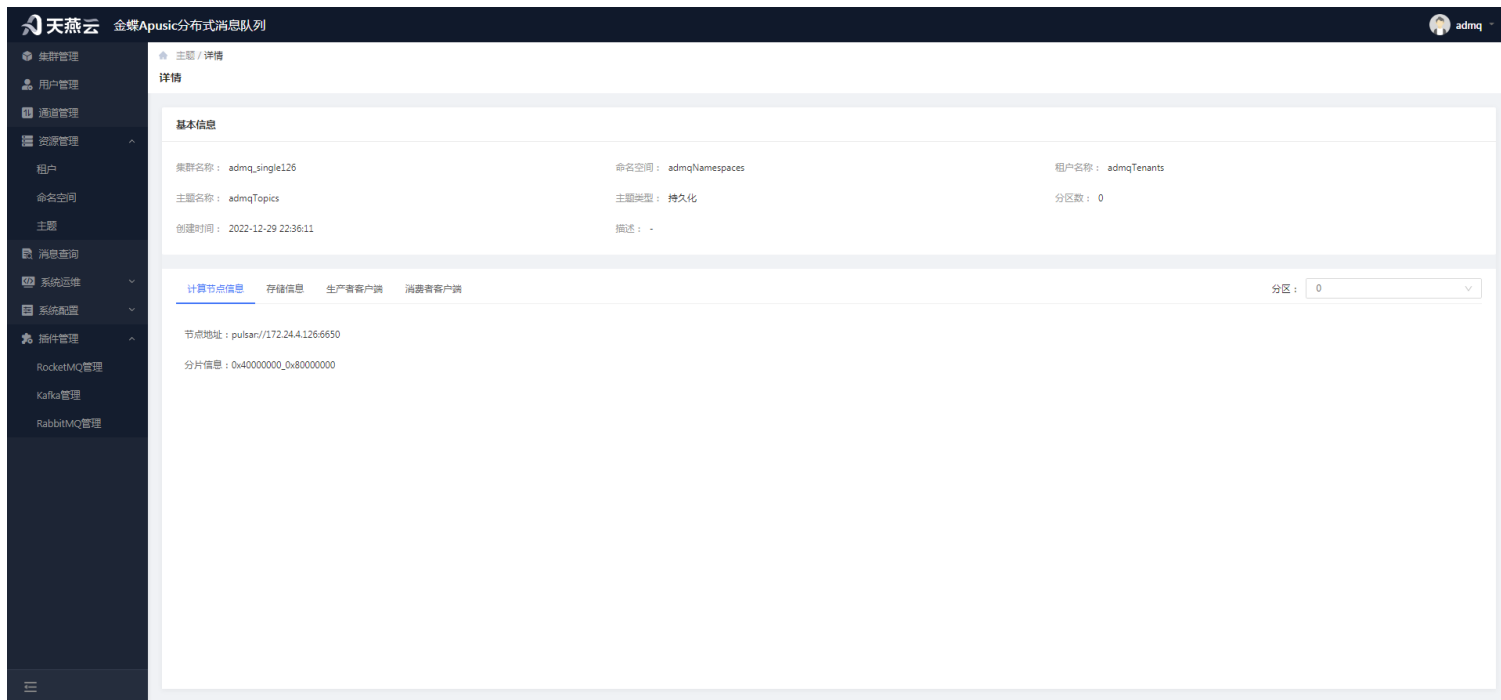
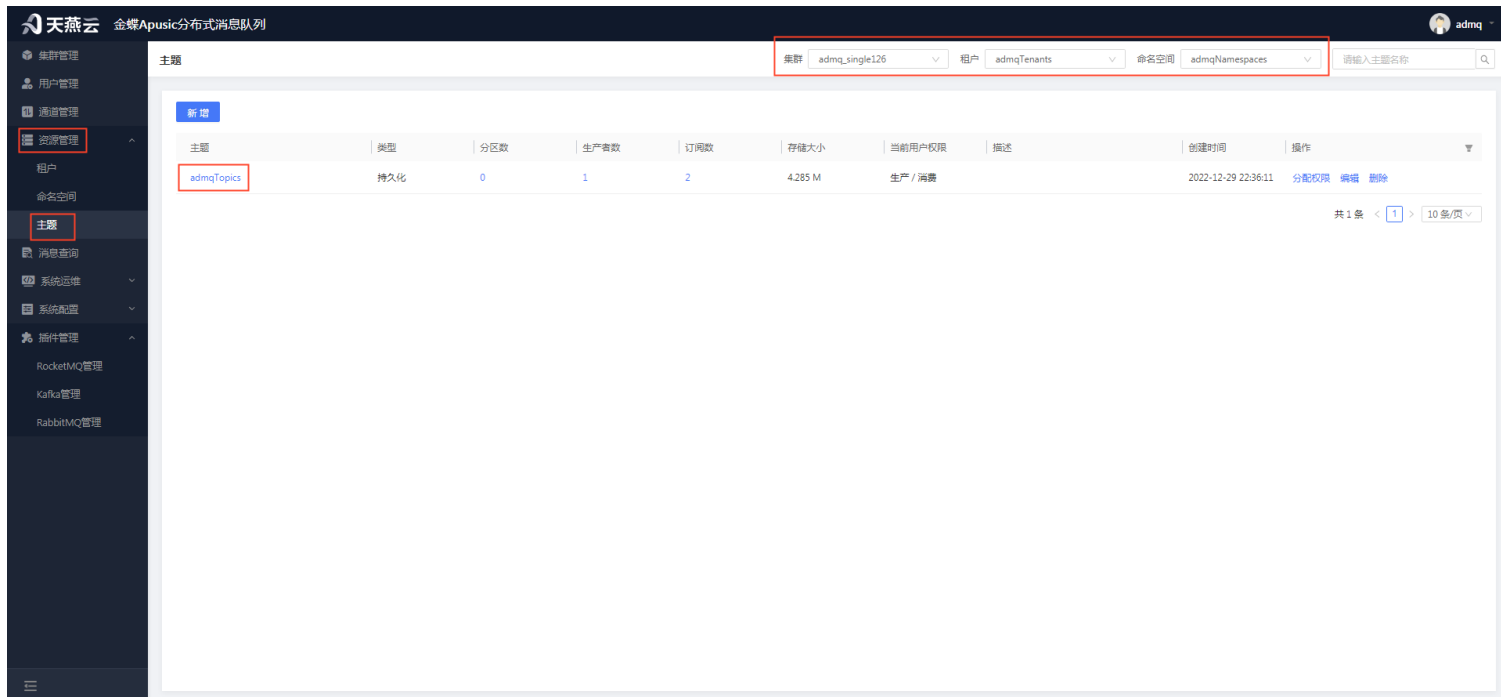


- 备注：分区为0的主题不可进行扩容：



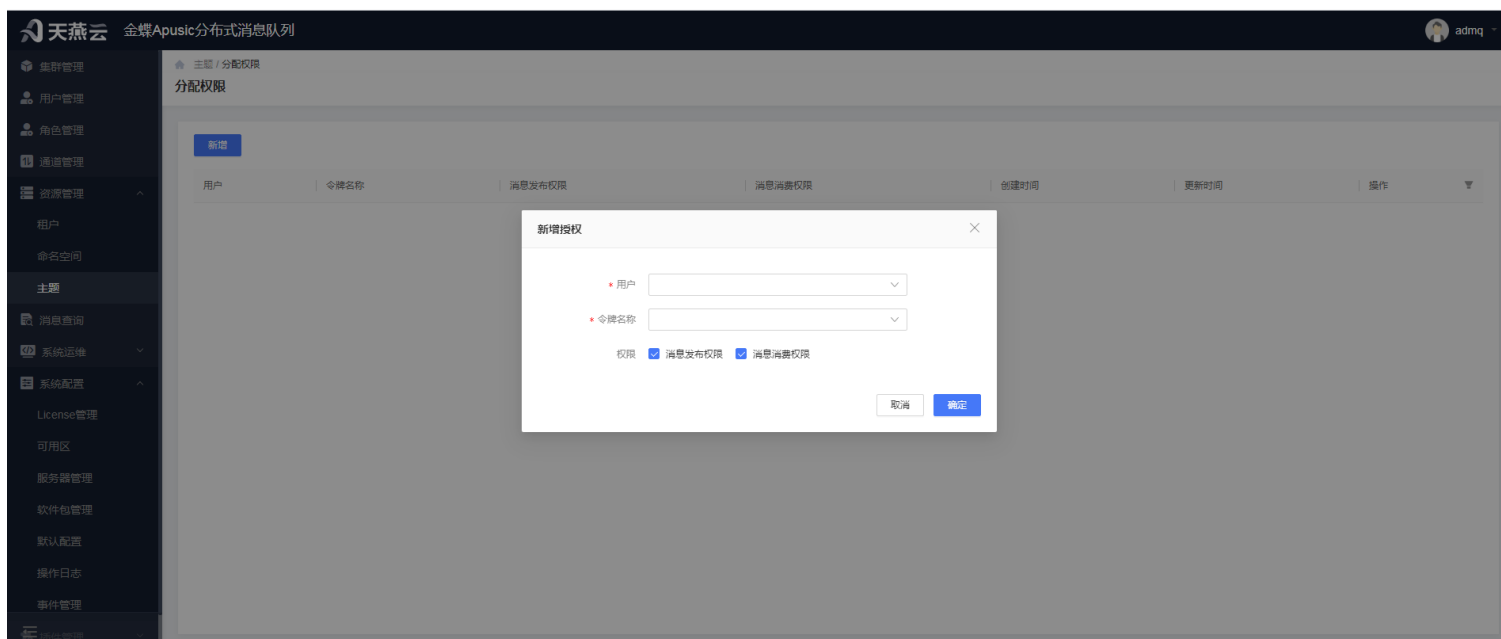
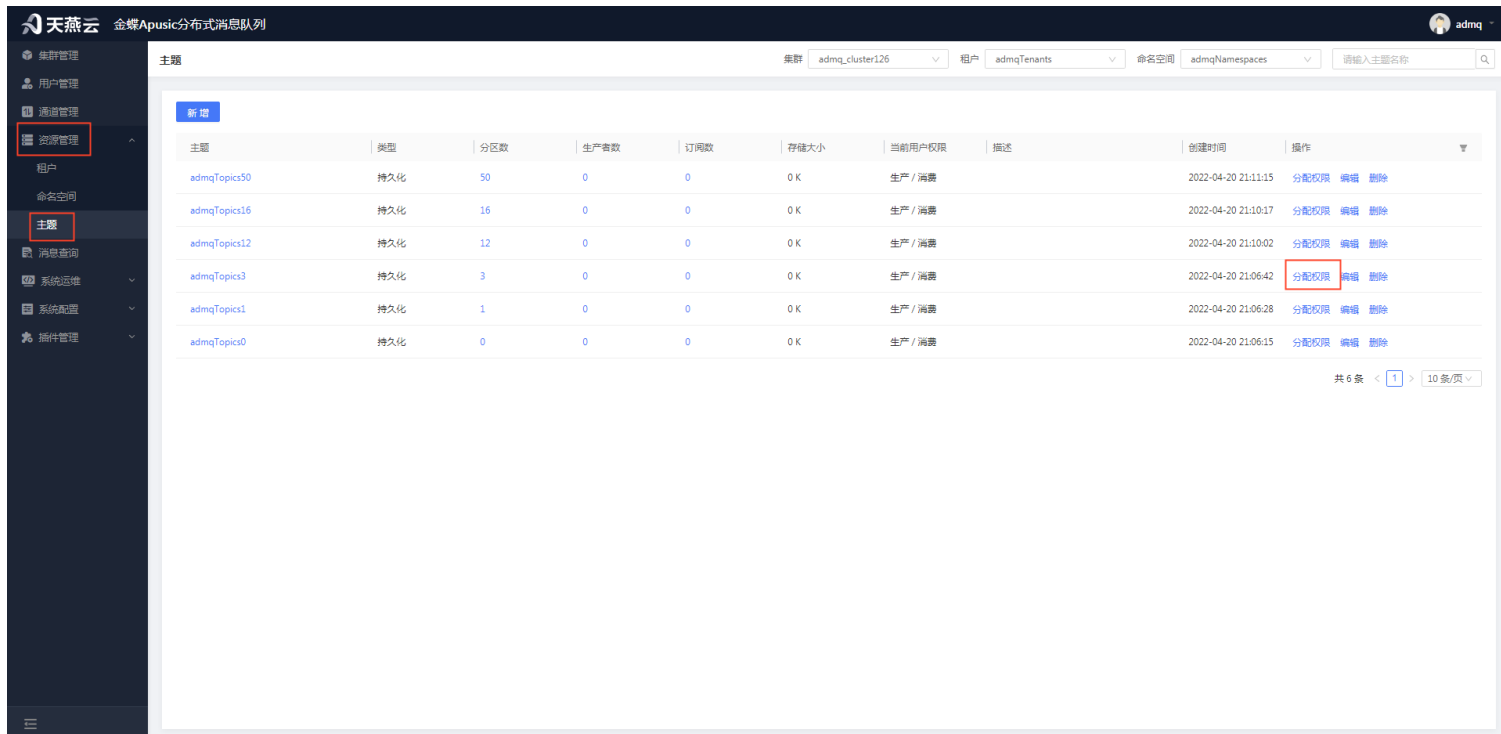
5.6.3.6 查看主题详情

进入【资源管理】->【主题】，在命名空间页面右上角选择集群、选择租户、选择命名空间，点击具体主题名称，可进行查看主题具体信息。



5.6.3.7 新增主题权限

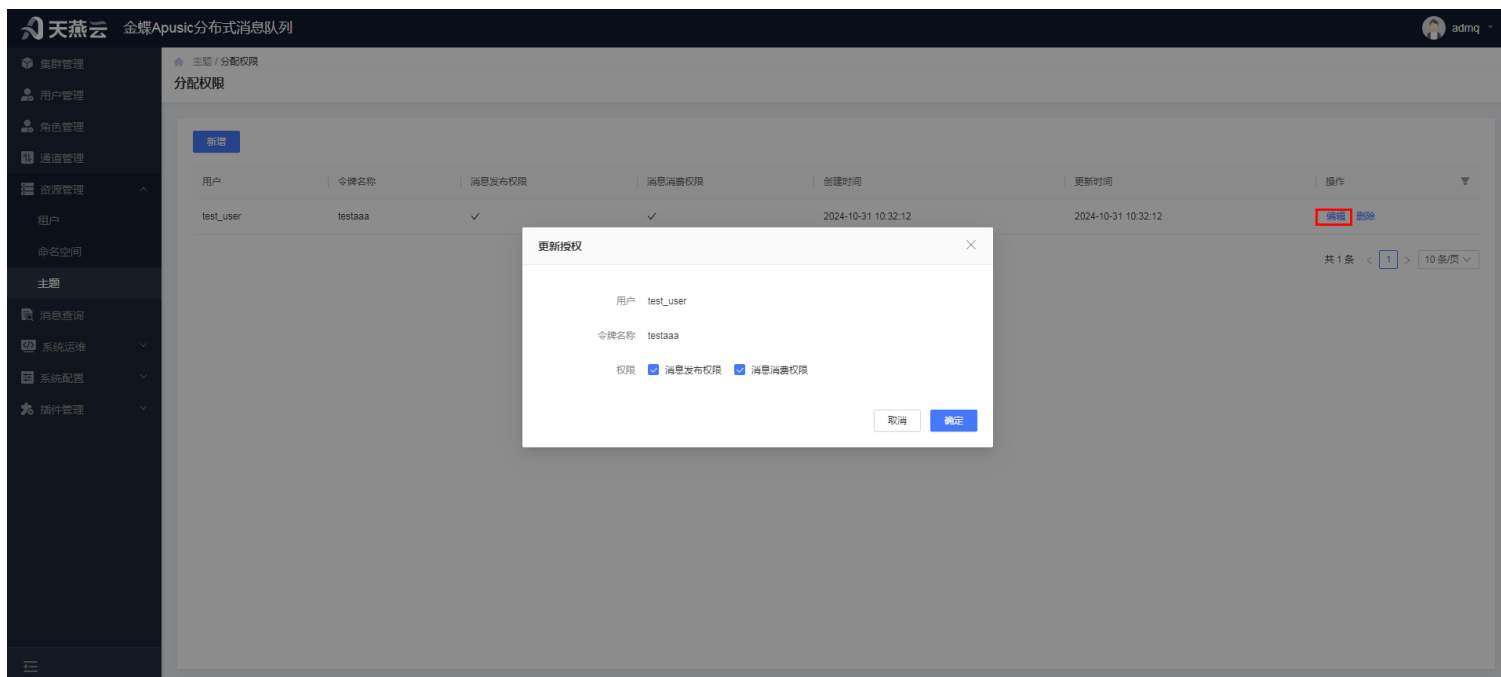
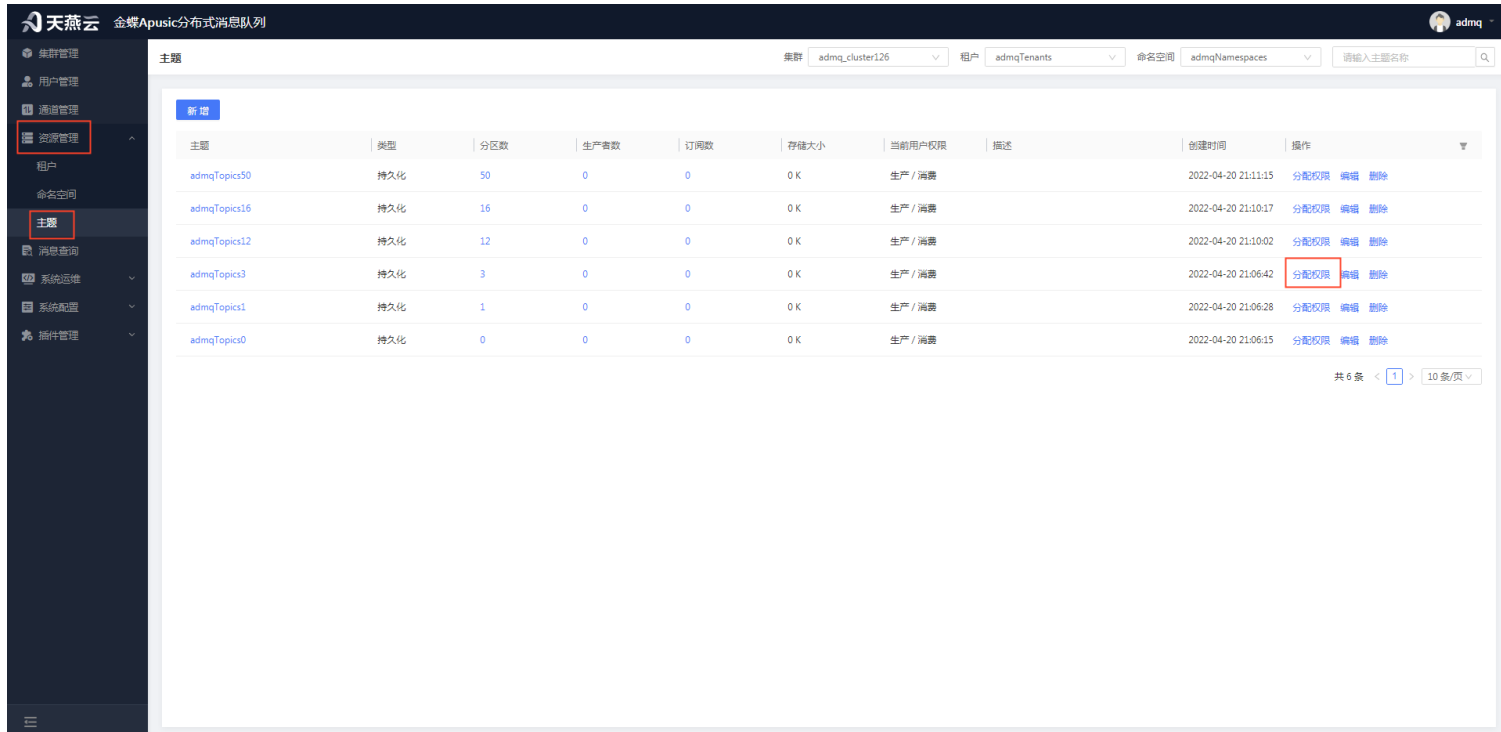
进入【资源管理】->【主题】，在主题页面选择需要分配权限的主题，点击【分配权限】按钮，进入分配权限页面；点击左上角【新增】按钮，可新增主题下用户的权限信息。注意：新增权限前，需满足集群服务可用；授权主要是给普通用户提供授权，默认超级管理员拥有所有主题的消息发布权限和消息消费权限；若主题所属的命名空间也有权限，则按最小粒度，主题设置的权限有效；若用户在命名空间无权限则需要先对命名空间进行授权。



- 用户：下拉选择，获取的用户管理（4.1）的用户信息
- 令牌名称：下拉选择，获取该用户下的令牌信息
- 权限：复选框，消息发布权限和消息消费权限

5.6.3.8 编辑主题权限

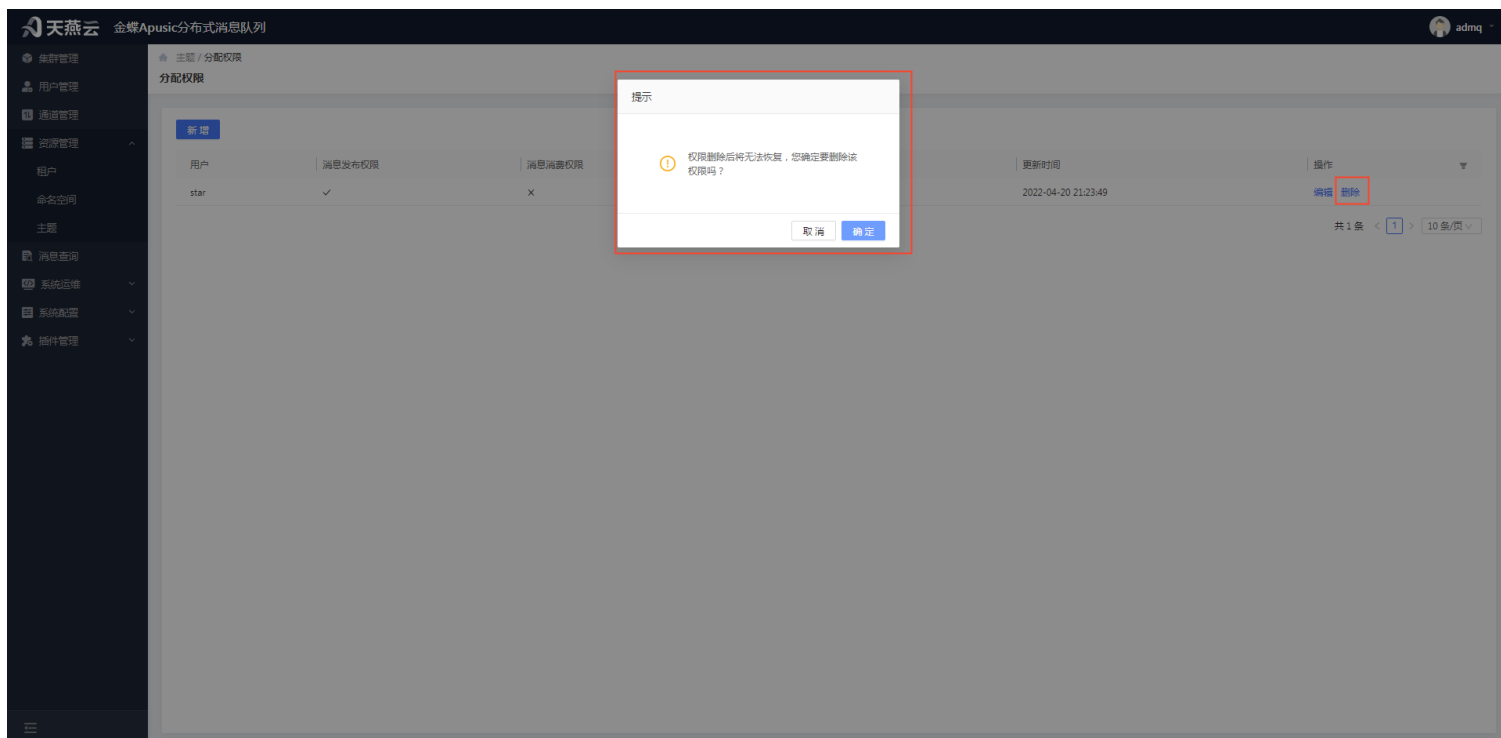
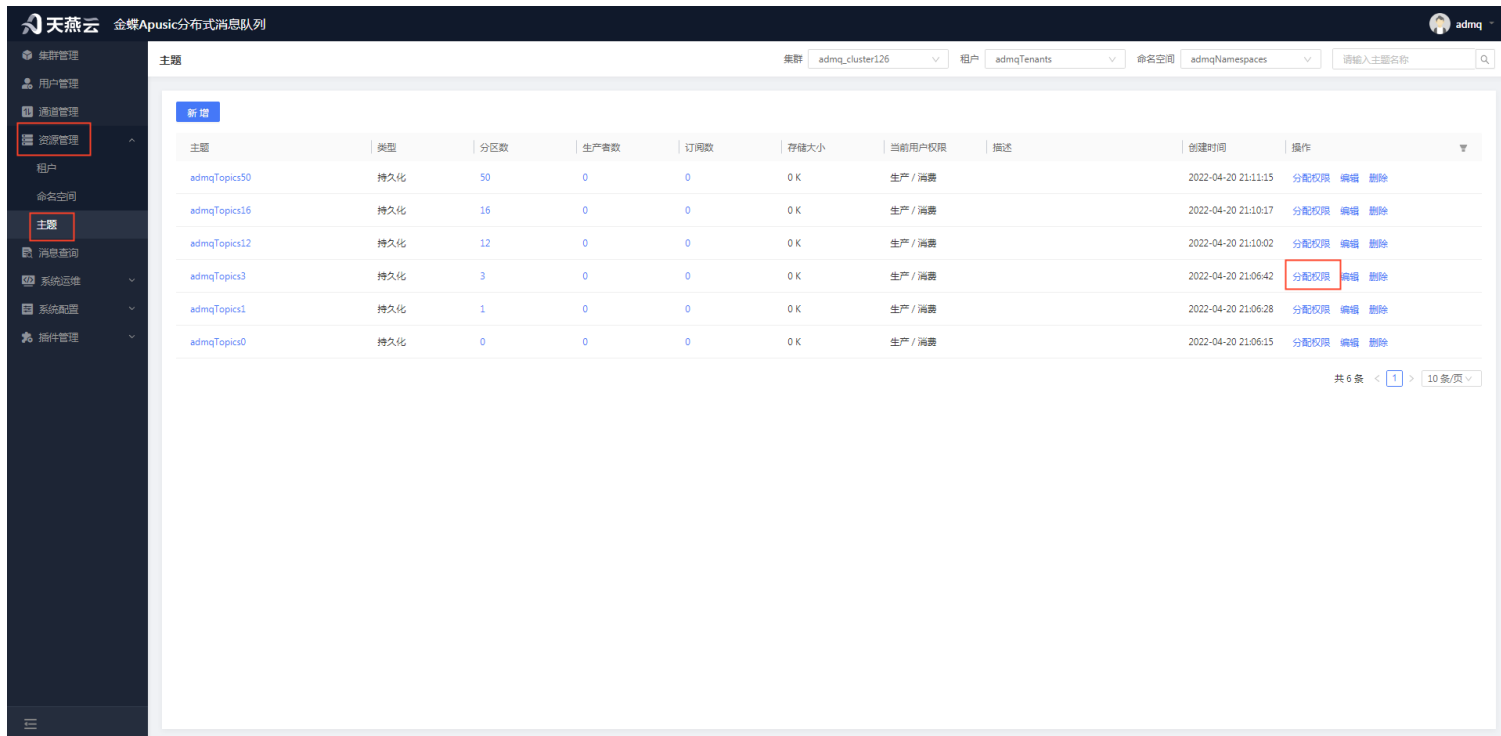
进入【资源管理】->【主题】，在主题页面选择需要分配权限的主题，点击【分配权限】按钮，进入分配权限页面；选择需要更新权限的用户，点击【编辑】按钮，可更新用户在主题下的权限。注意：更新权限前，需满足集群服务可用；若主题所属命名空间也有权限，则按最小粒度，主题设置的权限有效。



- 用户：用户名称，不可修改
- 令牌名称：令牌名称，不可修改
- 权限：复选框，消息发布权限和消息消费权限，可修改

5.6.3.9 删除主题权限

进入【资源管理】->【主题】，在主题页面选择需要分配权限的主题，点击【分配权限】按钮，进入分配权限页面；选择需要删除权限的用户，点击【删除】按钮，在弹出的确认删除弹框点击【确定】按钮，可删除用户在主题下的权限。注意：删除权限前，需满足集群服务可用；若删除权限的用户的命名空间下有权限，则命名空间权限生效。

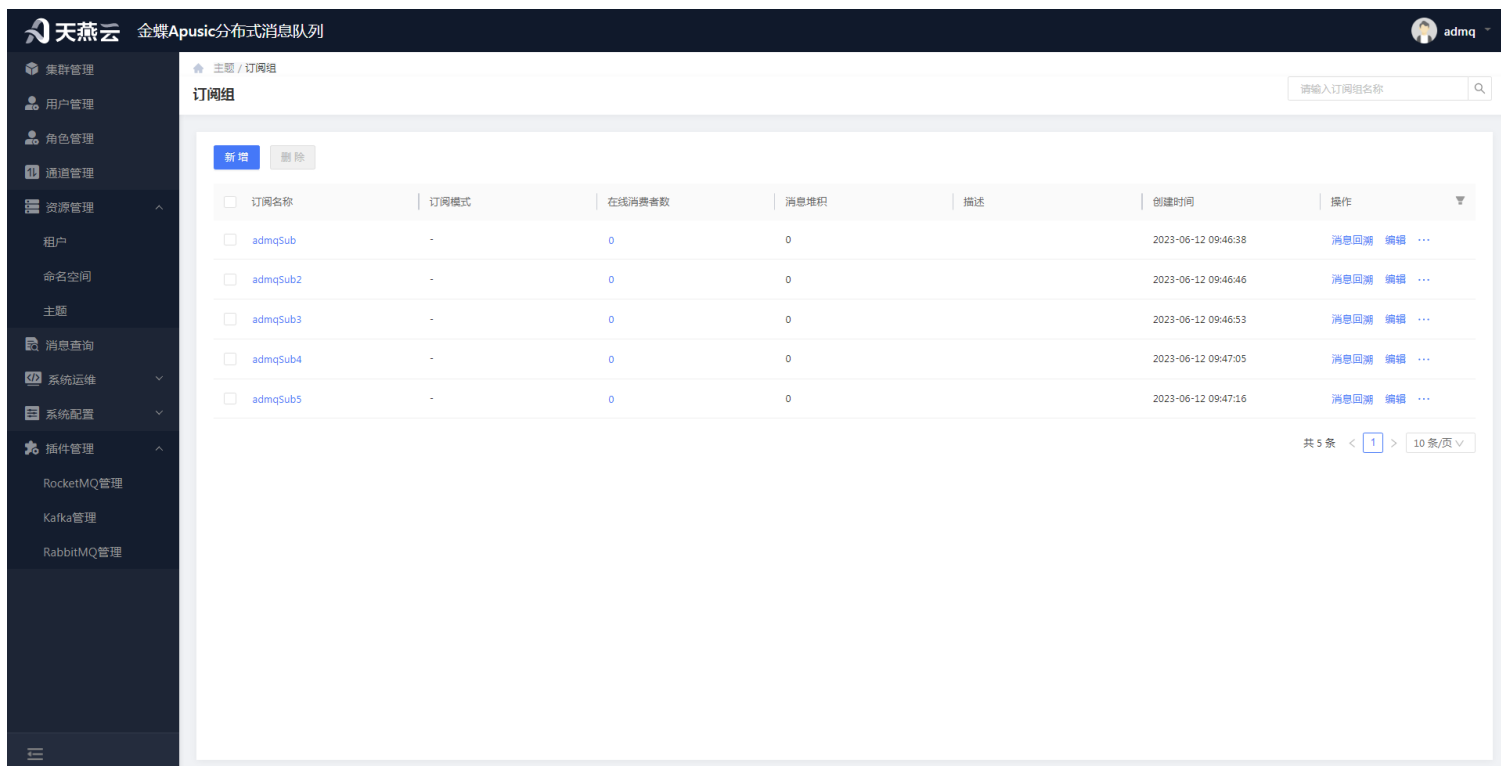
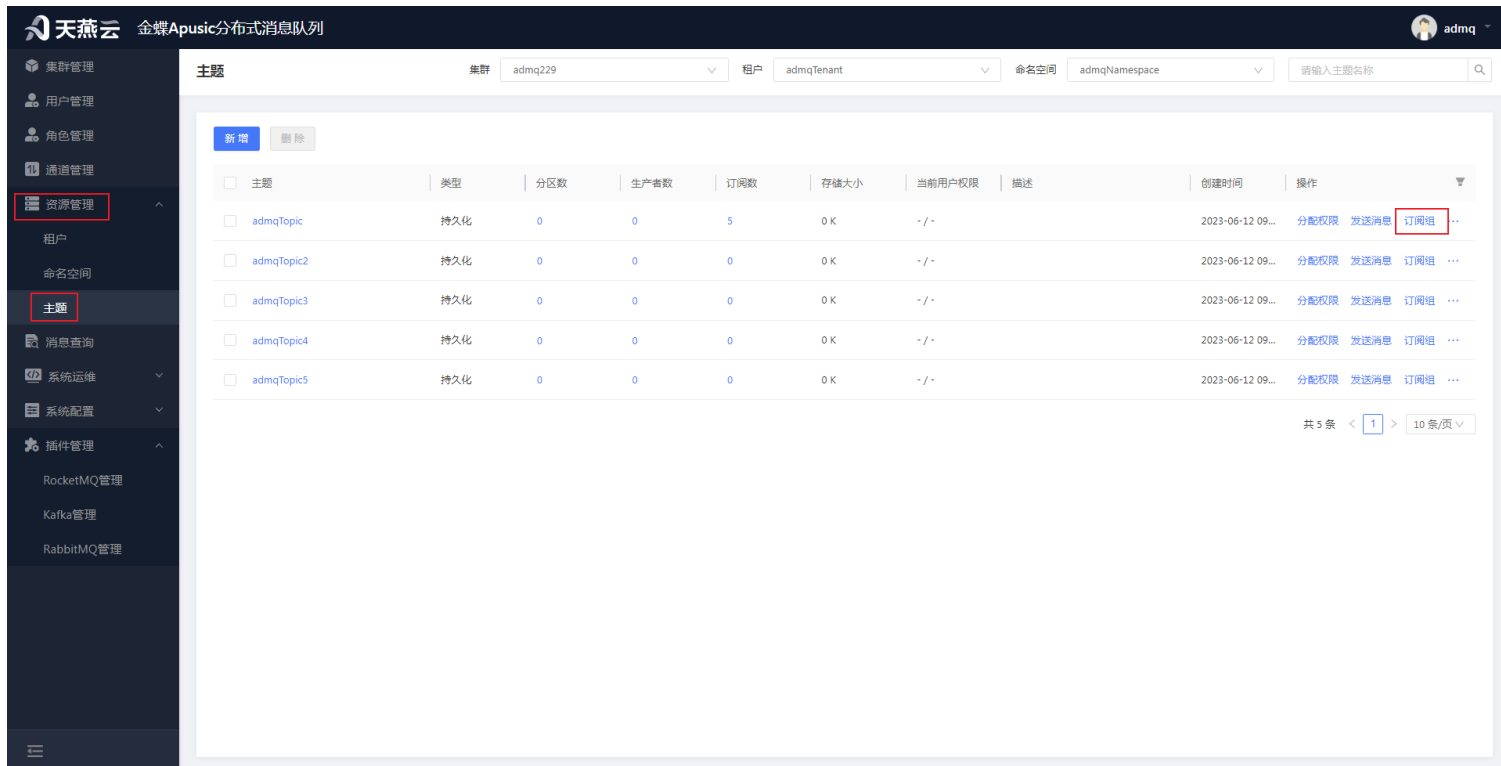


- 备注：删除用户后，使用用户的token向该主题发送接收消息时，会提示没有权限。

5.6.4 订阅组

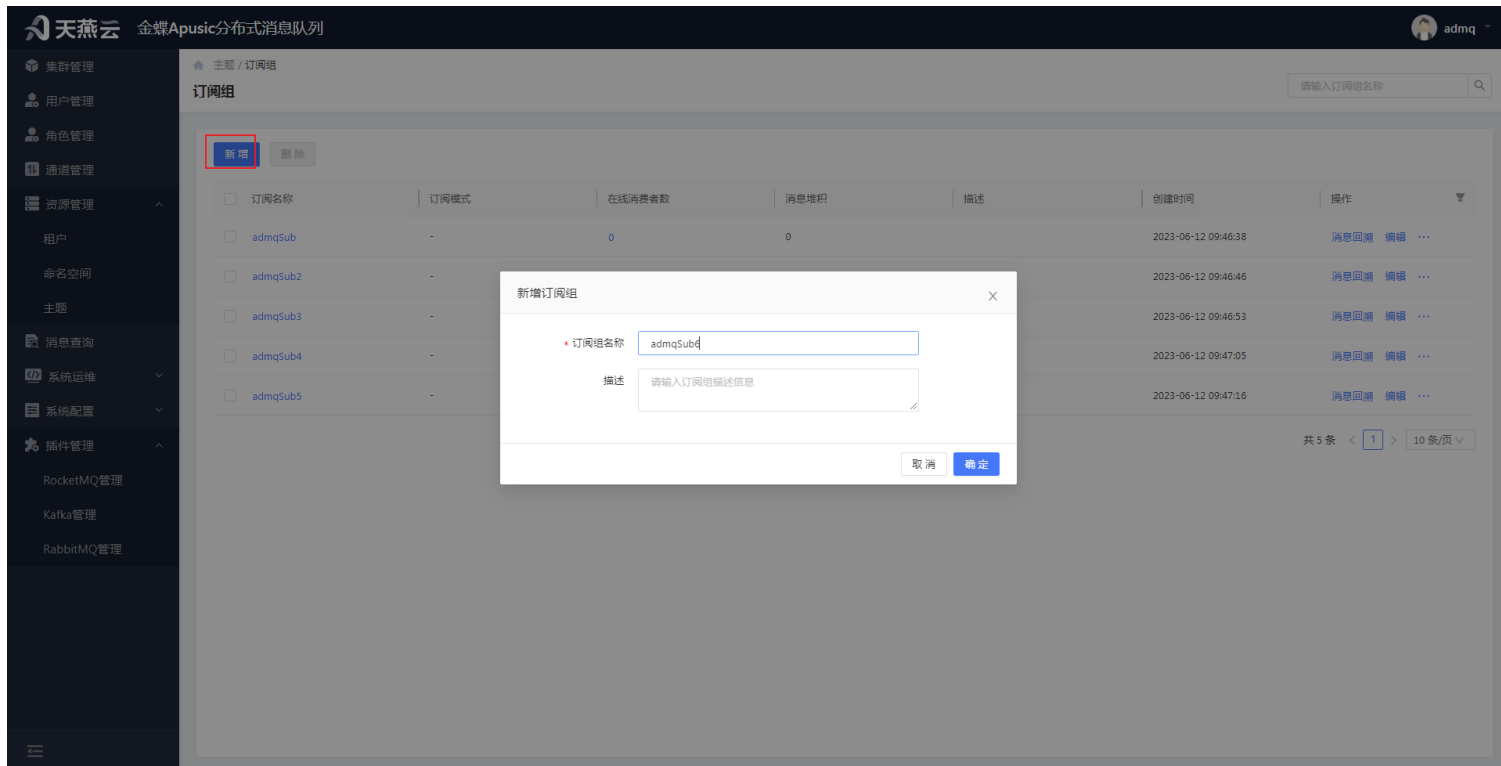
发送消息和接收消息，都需要绑定订阅组，订阅组就是作用于管理订阅组。订阅组可进行的操作有：新增订阅组、编辑订阅组、删除订阅组、查询订阅组、消息回溯、下载未消费、下载未确认。

资源管理—》主题—》在主题列表选择某一主题在操作列表点击“订阅组”按钮，进入到订阅组列表页面：



5.6.4.1 新增订阅组

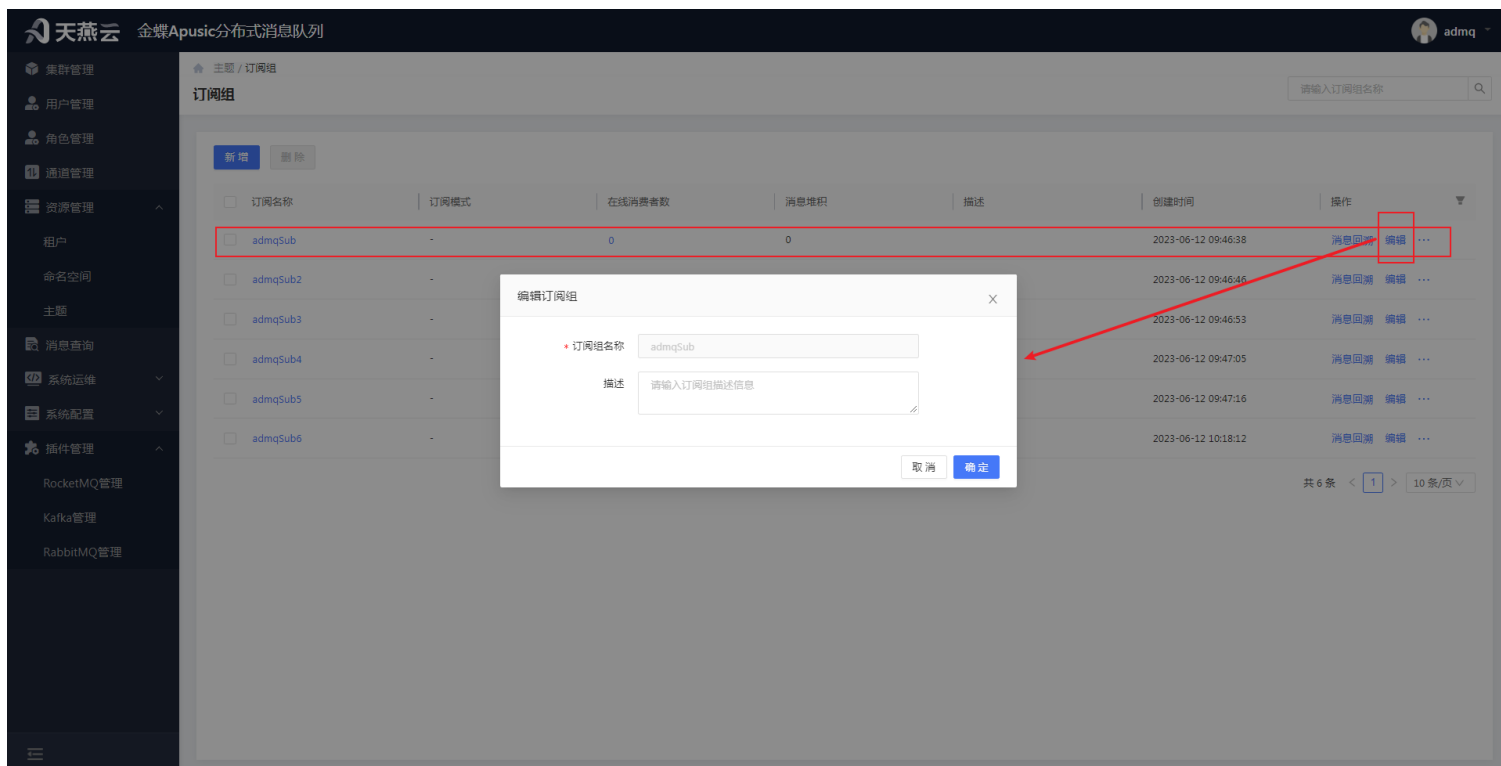
资源管理—》主题—》在主题列表选择某一主题在操作列表点击“订阅组”按钮，进入到订阅组列表页面，在订阅组列表页面，点击左上角【新增】按钮，可新增订阅组信息。注意：新增订阅组前，需满足集群服务可用。



- 订阅组名称：订阅组的名称，自定义，唯一标识
- 描述：订阅组的描述信息

5.6.4.2 编辑订阅组

资源管理—》主题—》在主题列表选择某一主题在操作列表点击“订阅组”按钮，进入到订阅组列表页面，选择要编辑的订阅组名称后的【编辑】按钮，可编辑订阅组信息。注意：编辑订阅组前，需满足集群服务可用。

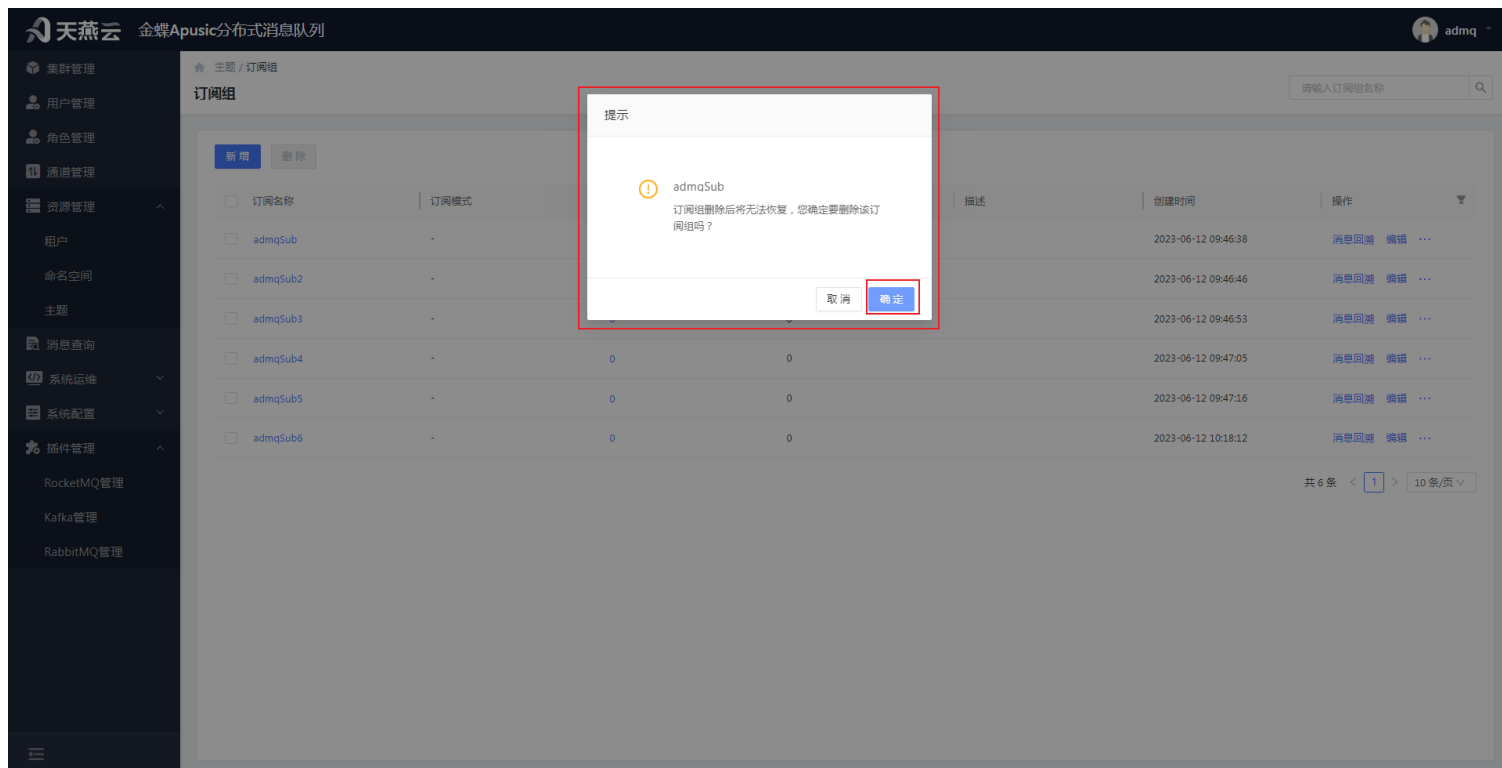
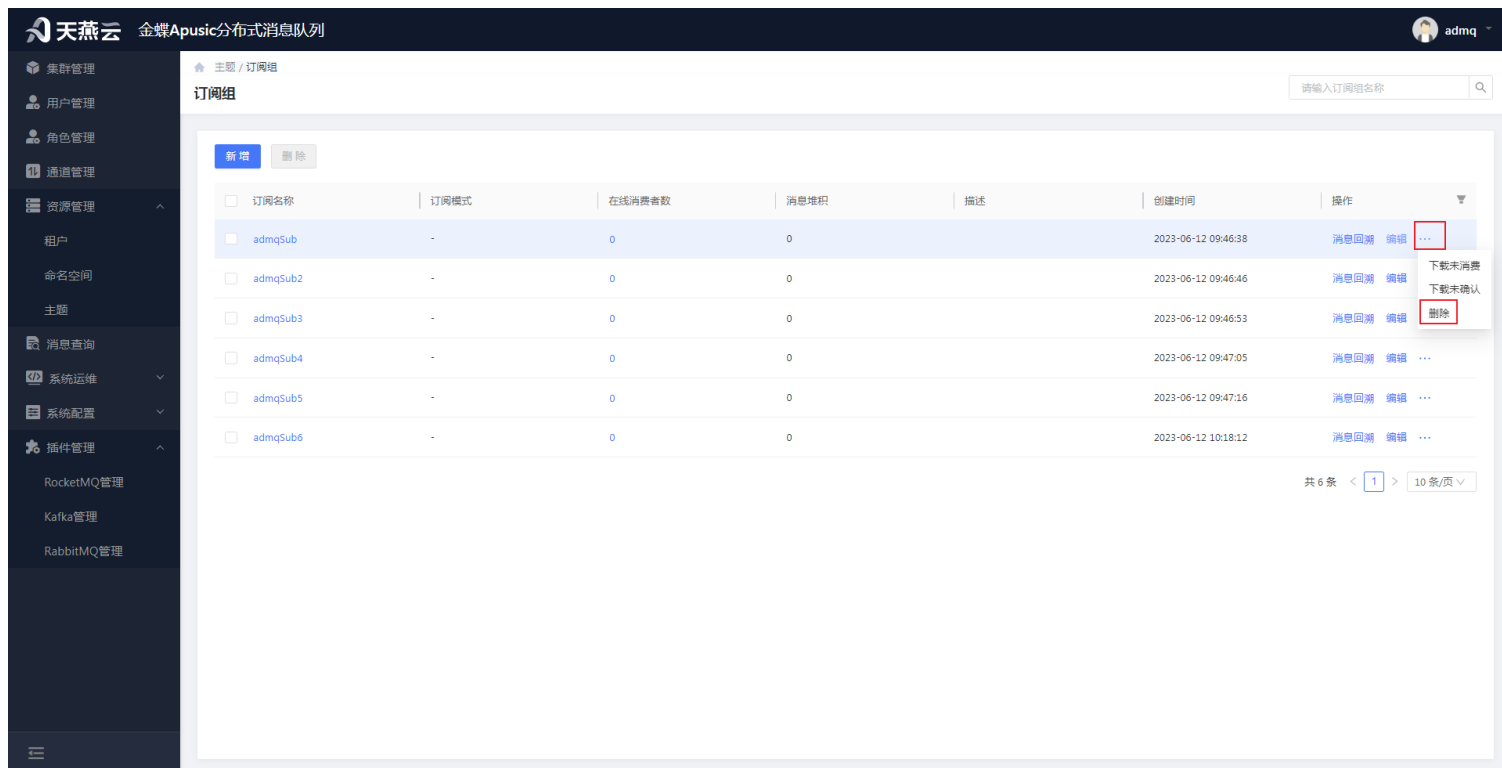


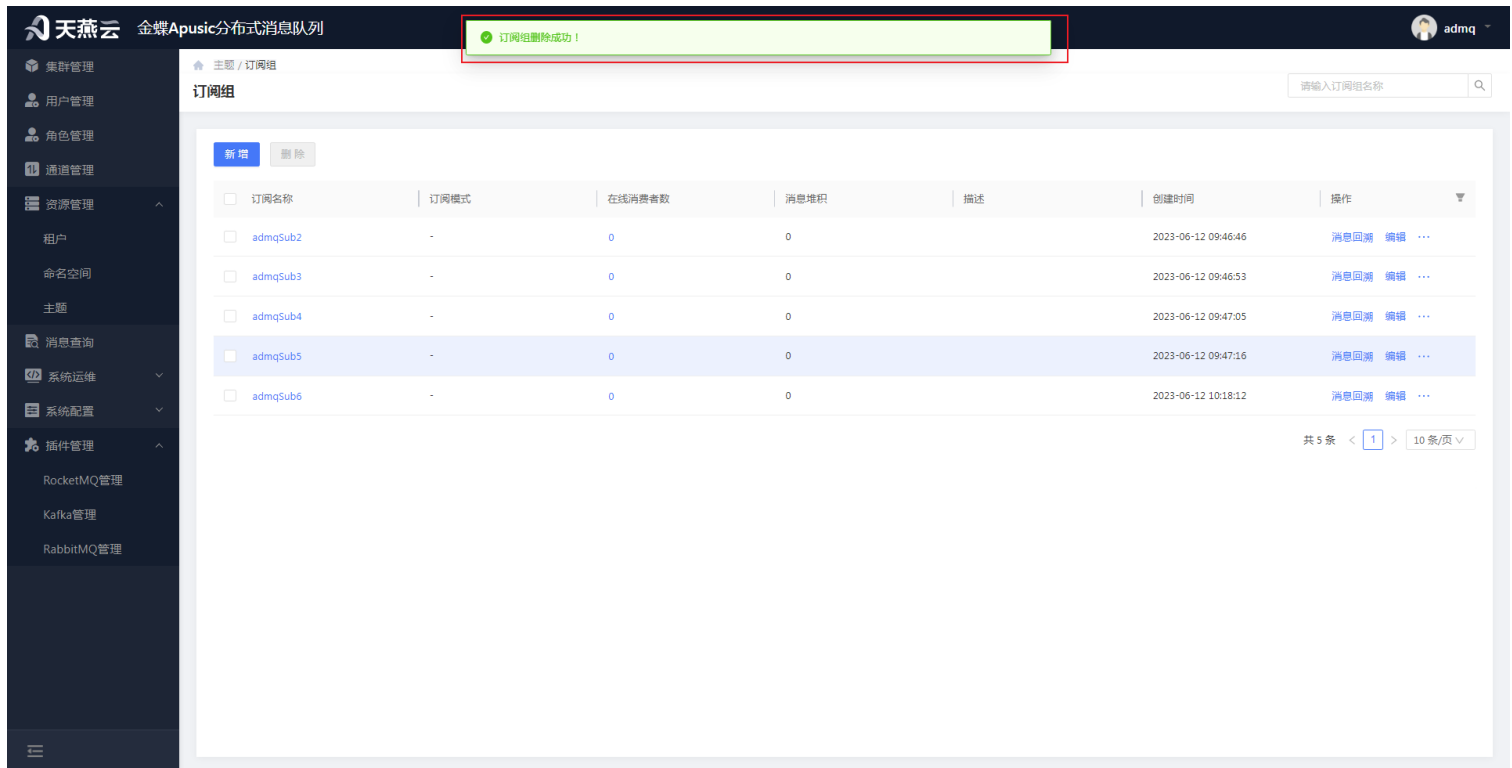
- 订阅组名称：订阅组的名称，置灰显示，不可修改

- 描述：订阅组的描述信息，可修改

5.6.4.3 删除订阅组

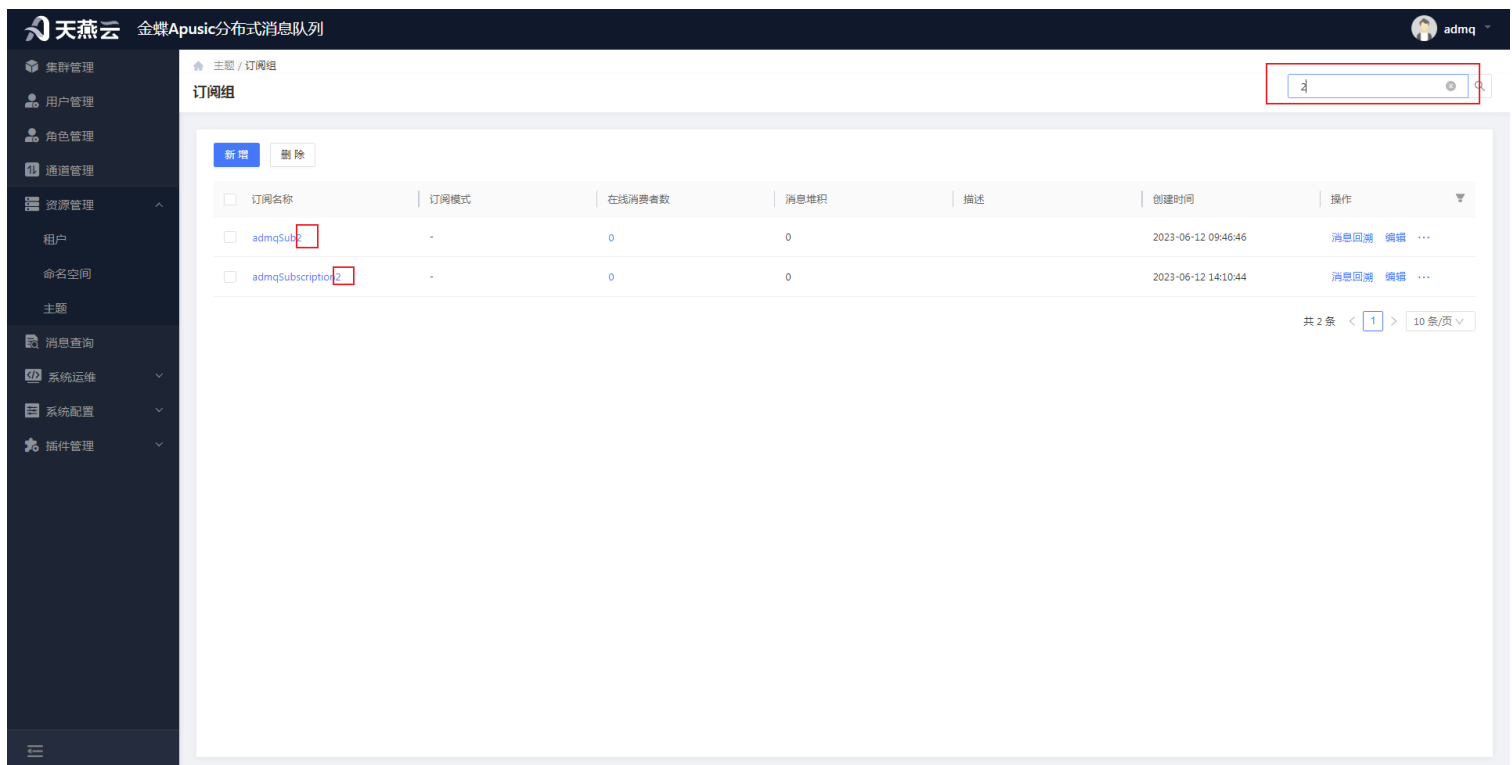
资源管理—》主题—》在主题列表选择某一主题在操作列表点击“订阅组”按钮，进入到订阅组列表页面，选择要删除的订阅组名称后的【更多】->【删除】按钮，在弹出的弹窗内点击【确定】按钮，可删除订阅组信息。注意：删除订阅组前，需满足集群服务可用。





5.6.4.4 查询订阅组

资源管理—》主题—》在主题列表选择某一主题在操作列表点击“订阅组”按钮，进入到订阅组列表页面，在右上角的查询框中精确或模糊输入订阅组名称，可查询订阅组信息。注意：订阅组名称不区分字母大小写。



5.6.4.5 查看订阅组详情

资源管理—》主题—》在主题列表选择某一主题在操作列表点击“订阅组”按钮，进入到订阅组列表页面，选择需要查看详细信息的订阅组，点击订阅组名称，可查看订阅组详细信息，如订阅组的基本信息、订阅组消费统计情况、主题消费统计情况。

天燕云 金蝶Apusic分布式消息队列 admq

集群管理 用户管理 角色管理 通道管理 资源管理 消息查询 系统运维 系统配置 插件管理

主题 / 订阅组

请输入订阅组名称

新增 删除

订阅名称	订阅模式	在线消费者数	消息堆积	描述	创建时间	操作
admSub2	-	0	169		2023-06-12 09:46:46	消息回溯 编辑 ...
admSub3	-	0	169		2023-06-12 09:46:53	消息回溯 编辑 ...
admSub4	-	0	169		2023-06-12 09:47:05	消息回溯 编辑 ...
admSub5	-	0	169		2023-06-12 09:47:16	消息回溯 编辑 ...
admSub	独占	0	40		2023-06-12 10:42:20	消息回溯 编辑 ...
admSubscription	-	0	169		2023-06-12 14:10:29	消息回溯 编辑 ...
admSubscription2	-	0	169		2023-06-12 14:10:44	消息回溯 编辑 ...

共 7 条 < 1 > 10 条/页

天燕云 金蝶Apusic分布式消息队列 admq

集群管理 用户管理 角色管理 通道管理 资源管理 消息查询 系统运维 系统配置 插件管理

主题 / 订阅组 / 订阅组详情

订阅组详情

基本信息

租户名称: admqTenant 命名空间: admqNamespace 订阅组名称: admqTopic
消费模式: 独占 创建时间: 2023-06-12 10:42:20 描述: -

订阅组消费统计 主题消费统计 实时 2023-06-12 14:05 2023-06-12 15:05 关闭 刷新

消息消费速率

条/秒

1

0

2023-06-12 14:06 06-12 14:20 06-12 14:34 06-12 14:48 06-12 15:05

消息堆积数量

条

120

100

80

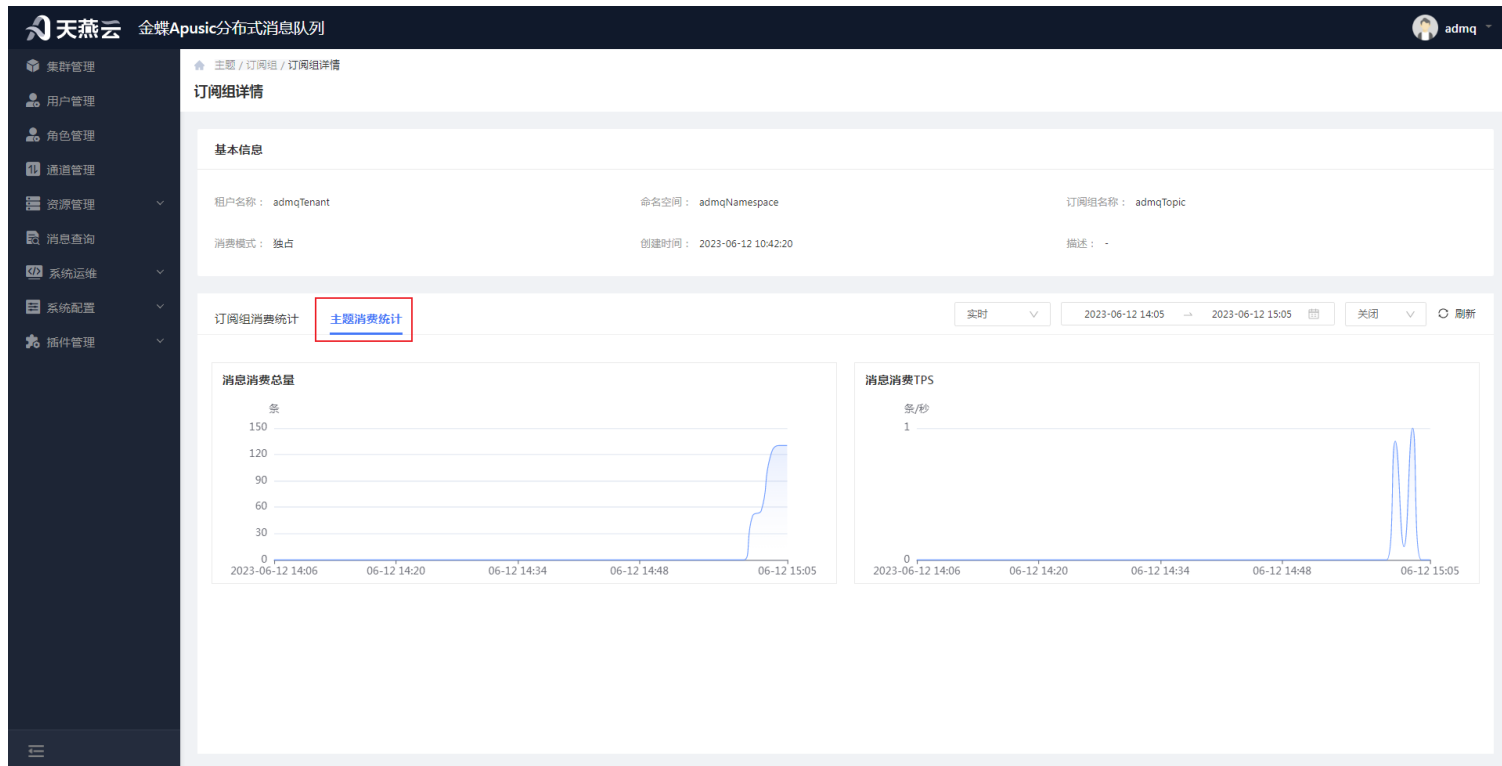
60

40

20

0

2023-06-12 14:06 06-12 14:20 06-12 14:34 06-12 14:48 06-12 15:05



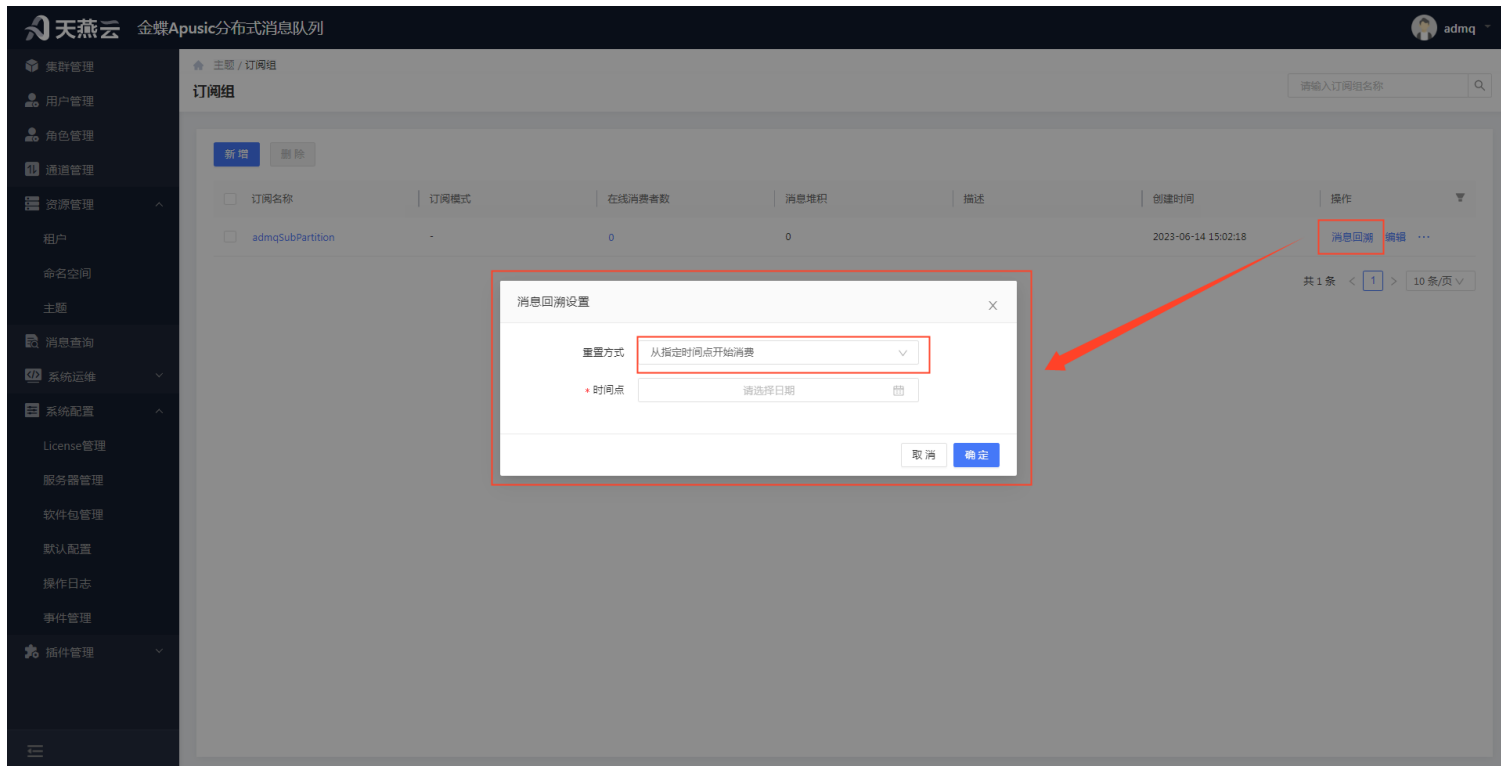
5.6.4.6 消息回溯

消息回溯是指可以根据需要重新消费已经被消费过的消息，具体来说，当一个消费者订阅了一个主题时，ADMQ会为该消费者维护一个消费进度，记录该消费者已经消费过的最后一条消息的位置，如果需要进行消息回溯，则可以将消费进度重置到之前的某个位置，从该位置开始重新消费消息。

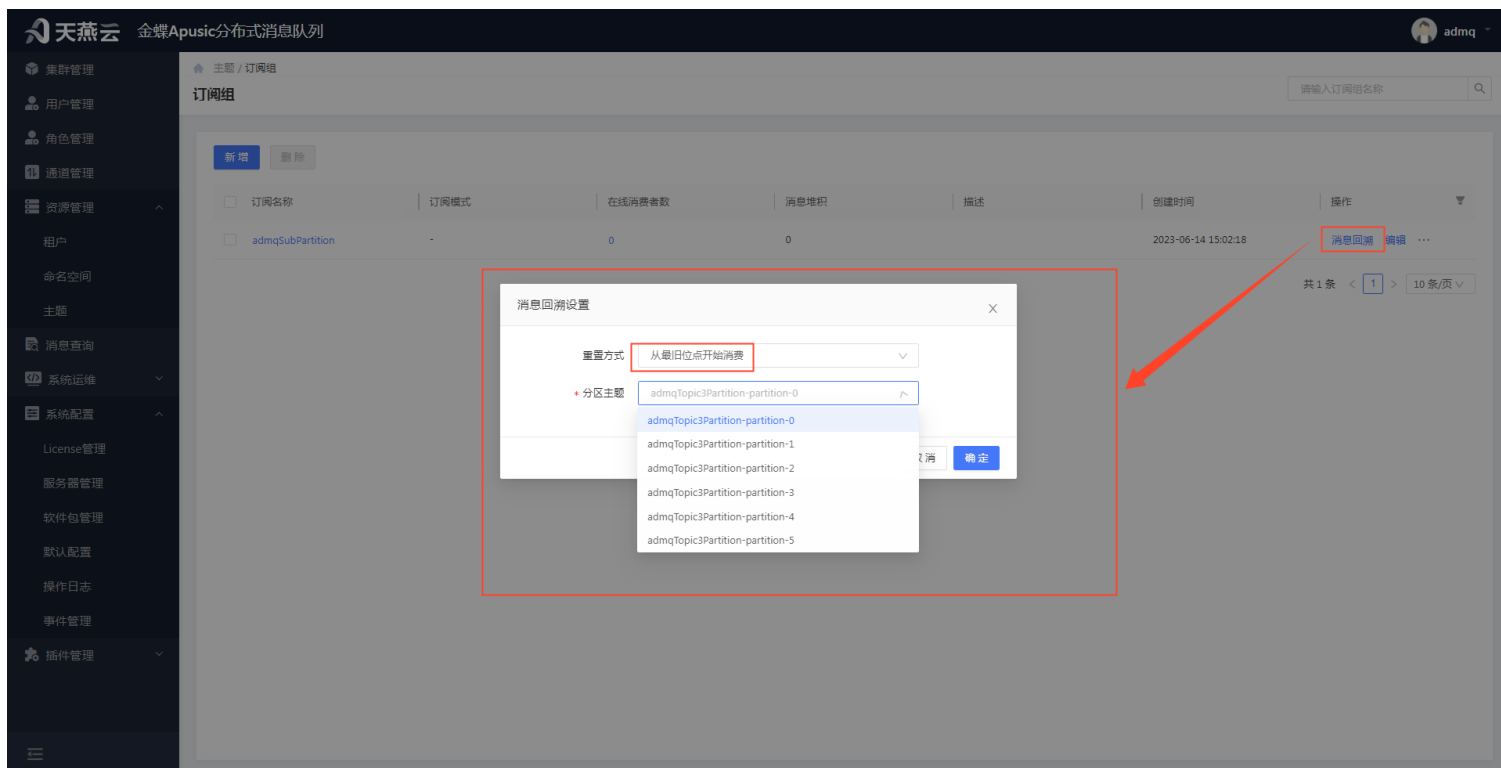
消息回溯可以帮助用户在一些特殊场景下重新消费消息，例如：数据处理出错、数据质量问题、数据流追溯等。同时，ADMQ还提供了不同的重置方式，可以根据不同的需求设置不同的回溯策略。管控支持的重置消费的方式共有四种，分别是：从指定时间点开始消费、从最旧位点开始消费、从最新位点开始消费、从指定位置开始消费。

具体操作过程：资源管理—主题—在主题列表选择某一主题在操作列表点击“订阅组”按钮，进入到订阅组列表页面，选择需要重置消费的订阅组，点击【消息回溯】按钮，出现消息回溯弹窗，可进行重置消费。（备注：若创建的分区数是0的主题，在“消息回溯设置”弹窗，分区主题就是主题名称；若创建的分区数是大于0的主题，在“消息回溯设置”弹窗，分区主题就是主题分区名称。）

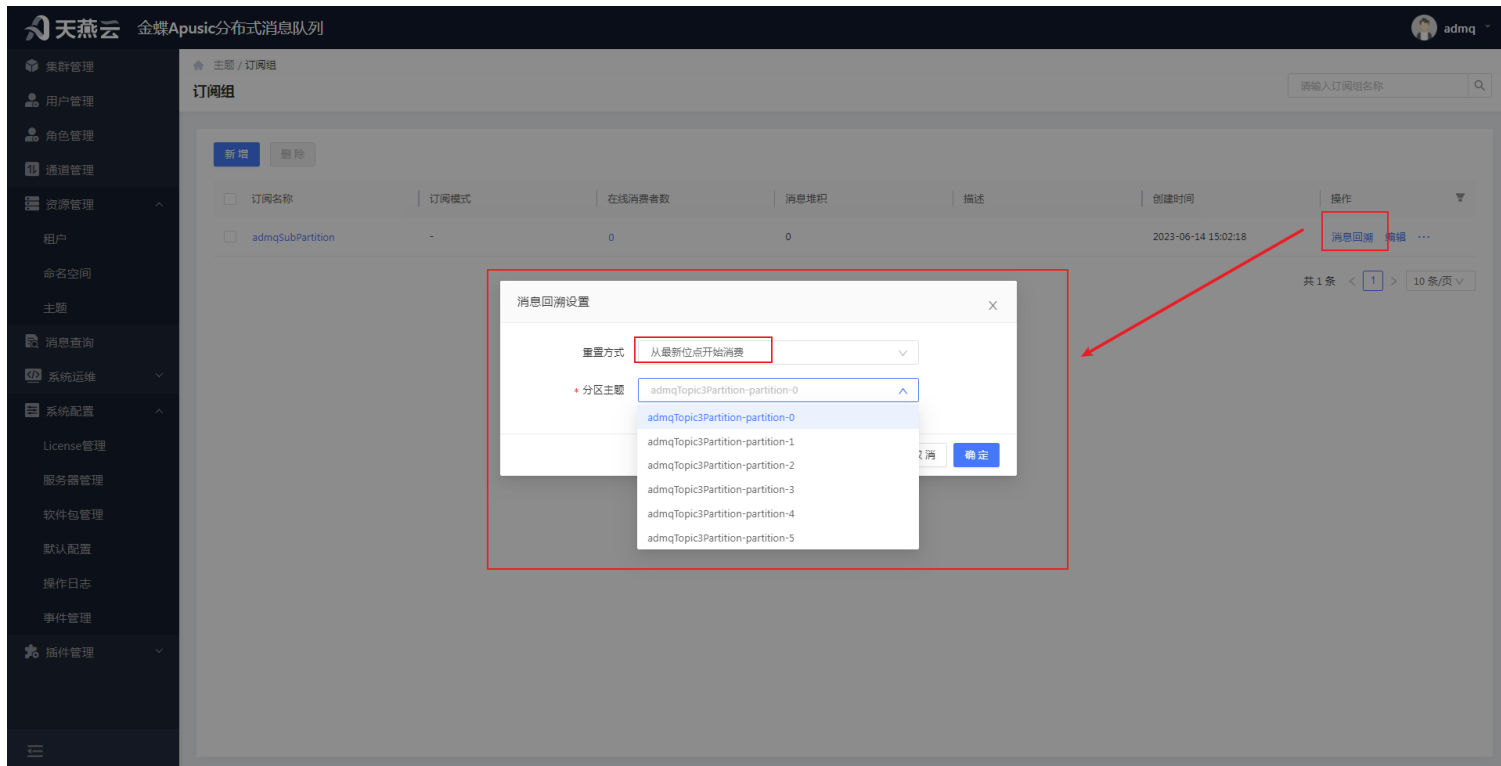
(1) 从指定时间点开始消费



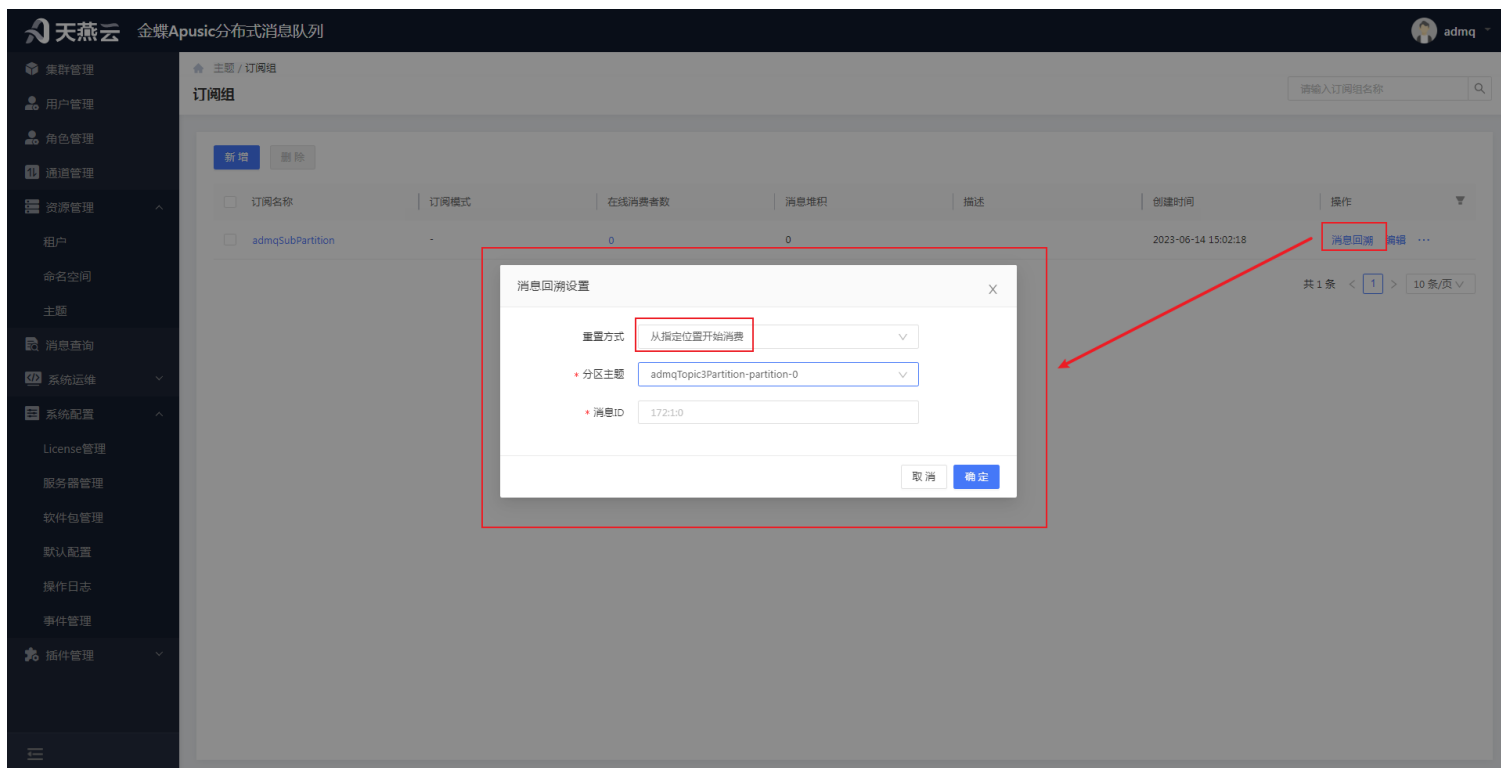
(2) 从最旧位点开始消费



(3) 从最新位点开始消费



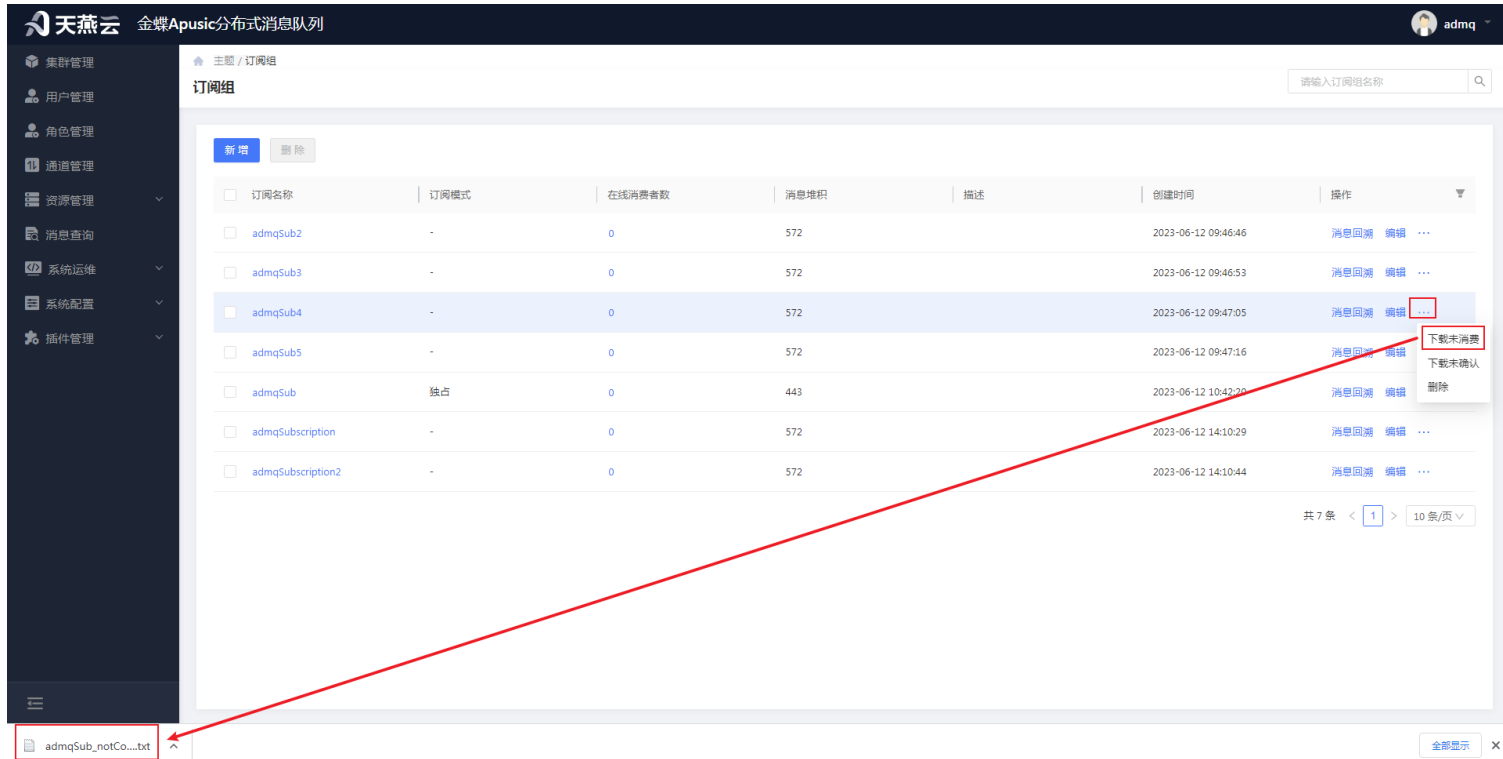
(4) 从指定位置开始消费



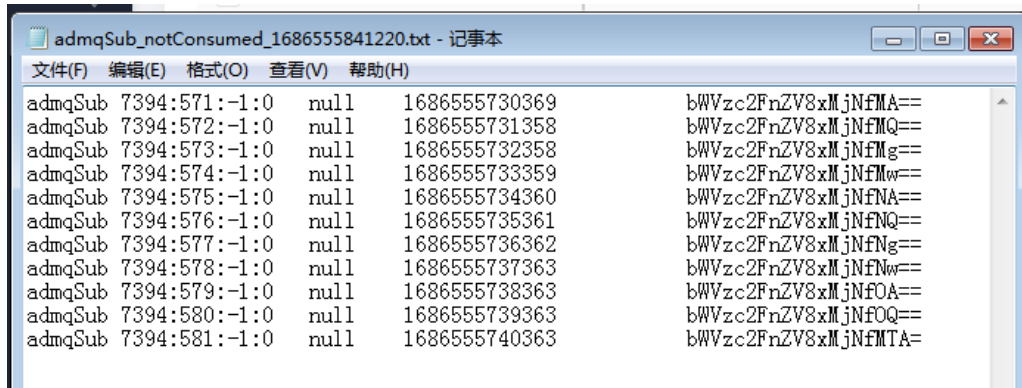
5.6.4.7 下载未消费

下载未消费功能可以让消费者将某个主题中尚未被消费的消息先下载到本地存储起来，以便后续的离线处理。这个功能对于一些需要离线处理大量数据的场景非常有用，例如：批量数据分析、数据挖掘等。

资源管理—》主题—》在主题列表选择某一主题在操作列表点击“订阅组”按钮，进入到订阅组列表页面，选择要删除的订阅组名称后的【更多】->【下载未消费】按钮，进行下载未被消费的消息到具体文件中。



打开下载的admSub_notConsumed_1686555841220.txt文件（事先向订阅组所在的主题发送了11条消息并且未被消费）：

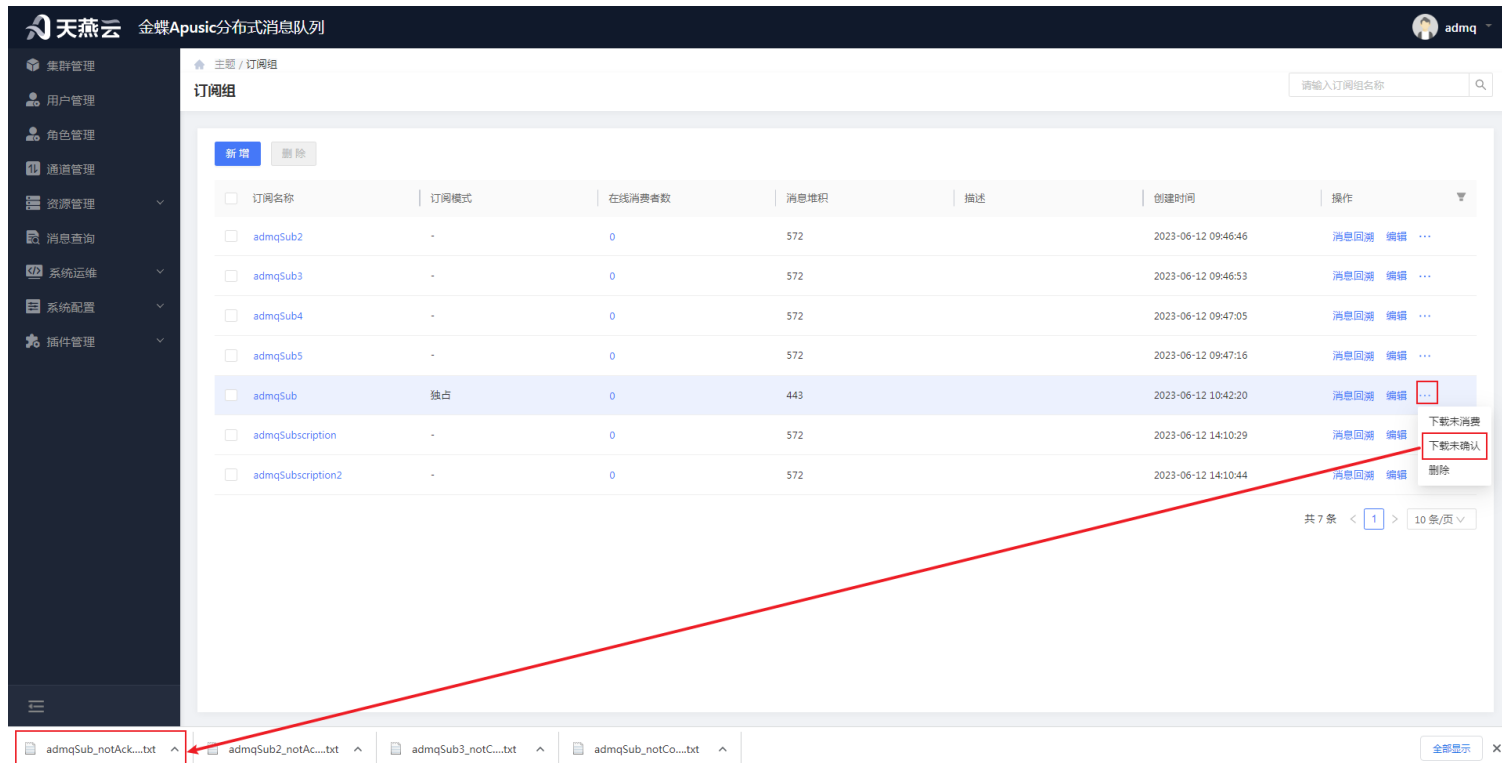


从左至右含义依次是：订阅组名称、偏移量、key值、时间戳、消息内容。

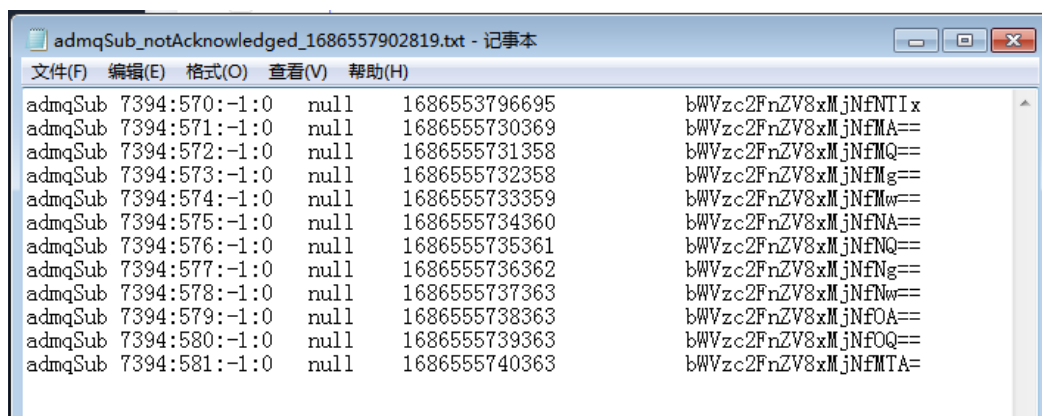
5.6.4.8 下载未确认

下载未消费功能可以让消费者将已经被消费但还未被确认的消息先下载到本地进行存储起来，以便在网络不稳定或消费者出现故障时进行消息重传。对需要保证消息可靠性的场景来说，这个功能非常有用。

资源管理—》主题—》在主题列表选择某一主题在操作列表点击“订阅组”按钮，进入到订阅组列表页面，选择要删除的订阅组名称后的【更多】->【下载未确认】按钮，进行下载已消费但未被确认的消息到具体文件中。



打开下载的admqsSub_notAcknowledged_1686557902819.txt文件（事先向订阅组所在的主题发送了11条消息并且未被消费）：



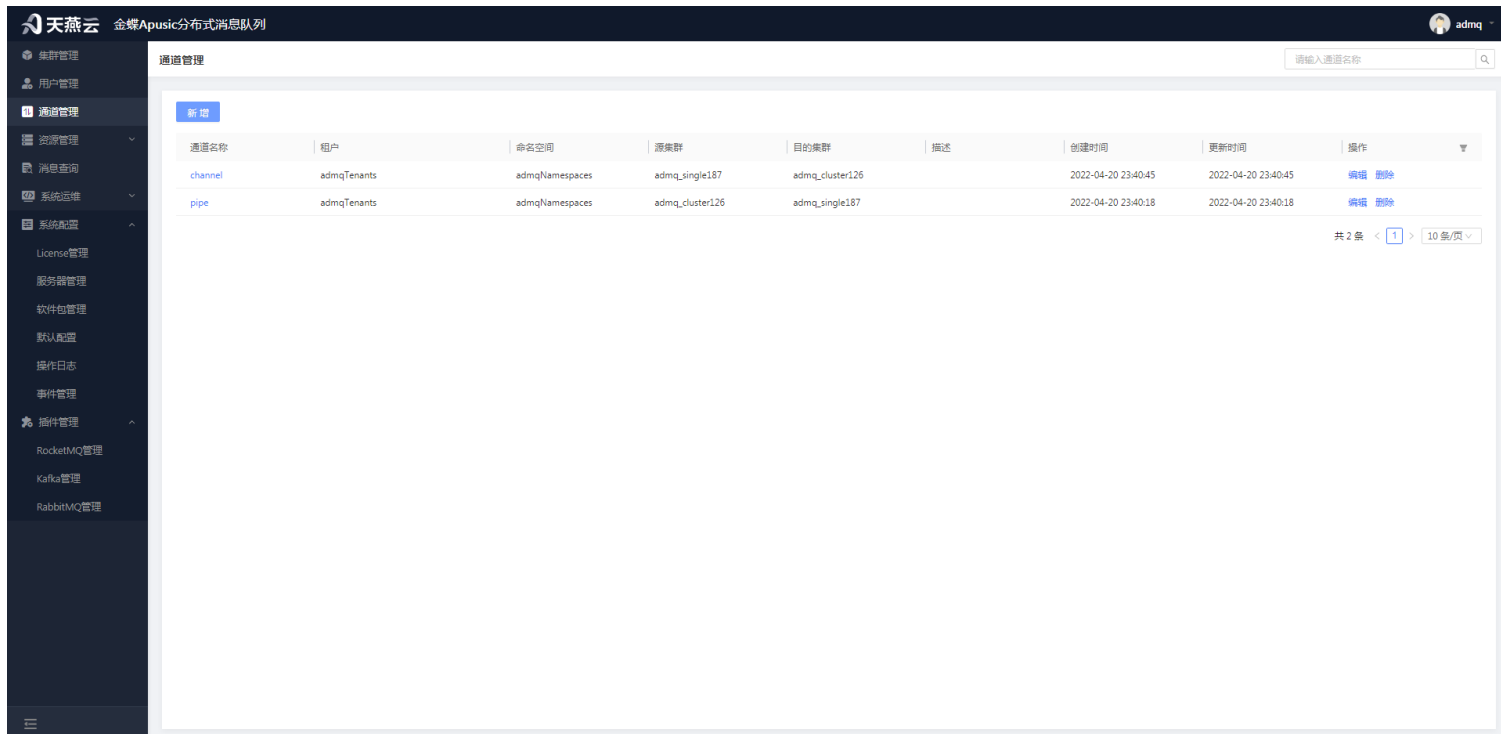
从左至右含义依次是：订阅组名称、偏移量、key值、时间戳、消息内容。

5.7 通道

5.7.1 通道管理

通道管理作用于跨集群间的消息复制功能，生产者发送到某一集群后，消费者能通过其他集群消费消息。通道管理用于新增通道信息、编辑通道信息、删除通道信息。注意：新增、编辑、删除通道之前，需要保证源集群和目的集群服务正常，否则操作将会失败。

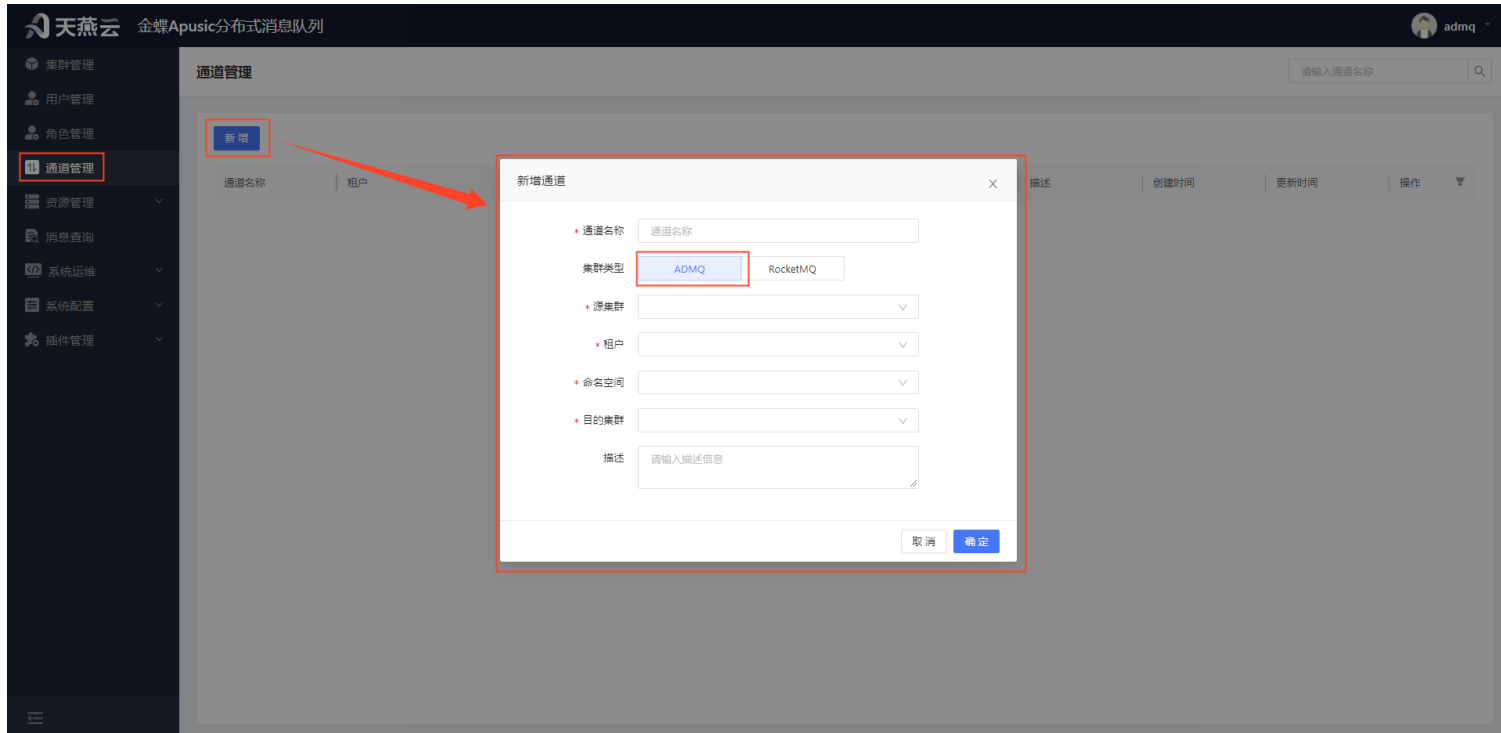
备注：通道目前支持原生ADMQ、原生RocketMQ。



5.7.1.1 新增通道

5.7.1.1.1 原生ADMQ通道

进入【通道管理】，在通道管理页面点击左上角【新增】按钮，可新增通道信息。注意：新增通道信息前，需要2个集群且2个集群中拥有相同名称的租户、命名空间，需集群服务器应正常。



- 通道名称：自定义，唯一标识
- 集群类型：选择ADMQ
- 源集群：被复制消息的集群，下拉选择

- 租户：被复制消息的租户，下拉选择
- 命名空间：被复制消息的命名空间，下拉选择
- 目的集群：需复制消息的集群，下拉选择
- 描述：通道描述信息

5.7.1.1.2 原生RocketMQ通道

在对数据安全要求比较高的业务场景下，可能需要部署多个集群，集群之间通过连接器方式进行复制，也就是发送到某一个集群的消息会实时同步到其他集群。

前置条件：需要2个集群且2个集群中拥有相同名称的租户、命名空间、主题，需集群服务器应正常。

具体实现步骤：

(1) 部署RocketMQ Connector

RocketMQ集群消息同步依赖于RocketMQ Connector，通过RocketMQ Replicator Connector组件实时读取源集群的消息并发送到目标集群，从而实现集群消息同步功能。

1. 拷贝rocketmq-connect.tar.gz到服务器并解压：tar zxvf rocketmq-connect.tar.gz
2. 修改配置：vi conf/connect-distributed.conf

```
# 集群部署时，修改后边的数字
workerId=DEFAULT_WORKER_1
# 数据保存路径
storePathRootDir=/home/connect/mq-connect/storeRoot

## 管理服务端口
httpPort=8082

# RocketMQ 集群名称和连接到地址；一般选择一个集群保存任务及进度。
clusterName="rocketmq_dest"
namesrvAddr=172.24.4.217:19876
# 认证配置
aclEnable=true
accessKey=adm123456
secretKey=eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1xIn0.14FEG3SGV7jNRfTE3pkTnAb4GSXhiddHec_8BazbY

autoCreateGroupEnable=false

# 不需要配置
pluginPaths=
```

3. 启动程序：./start.sh

4. 添加进程检测任务

使用linux crontab监控程序运行情况，发现未运行则重新启动程序。

```
crontab -e
```

添加以下内容（5分钟检测一次进程是否存在，不存在则启动）：

```
* /5 * * * * cd /admq/rocketmq-connect/bin && check.sh rocketmq-connect
```

其中check.sh脚本内容如下，可以根据实际情况进行改动

```
#!/usr/bin/env bash

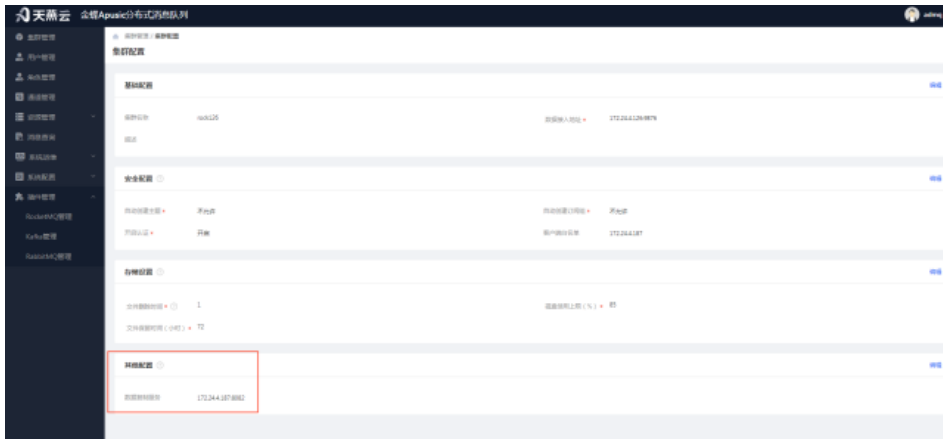
date=`date "+%Y-%m-%d %H:%M:%S"`
function check() {
name=$1
pid=`ps -ef |grep $name |grep java |grep -v grep |awk '{print $2}'`
if [ "$pid" != "" ]; then
echo "$date $name is running ..."
else
su - admq -c "cd `pwd` && bin/start.sh"
echo "$date start $name ..." >> program.status
fi
}

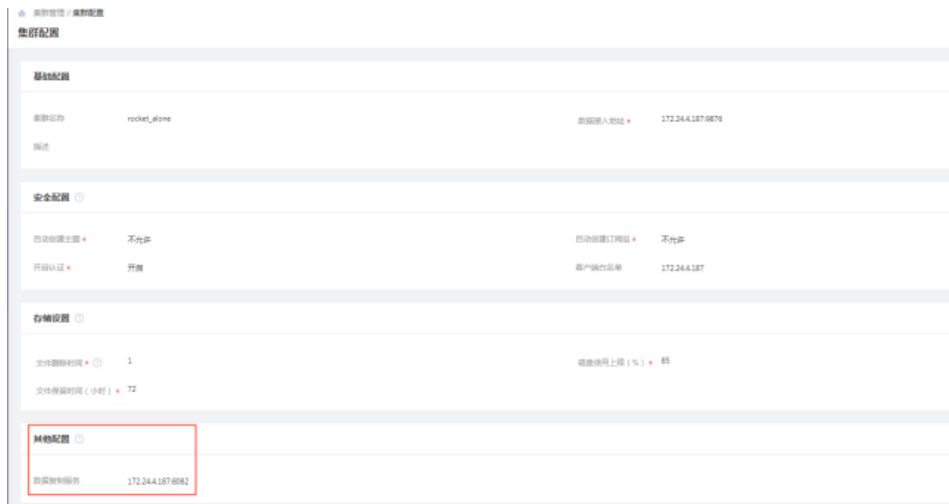
check $1
```

其中admq为启动该程序的系统用户，如果root用户则不需要切换到该用户

(2) 分别对两个RocketMQ集群设置RocketMQ Connector服务地址

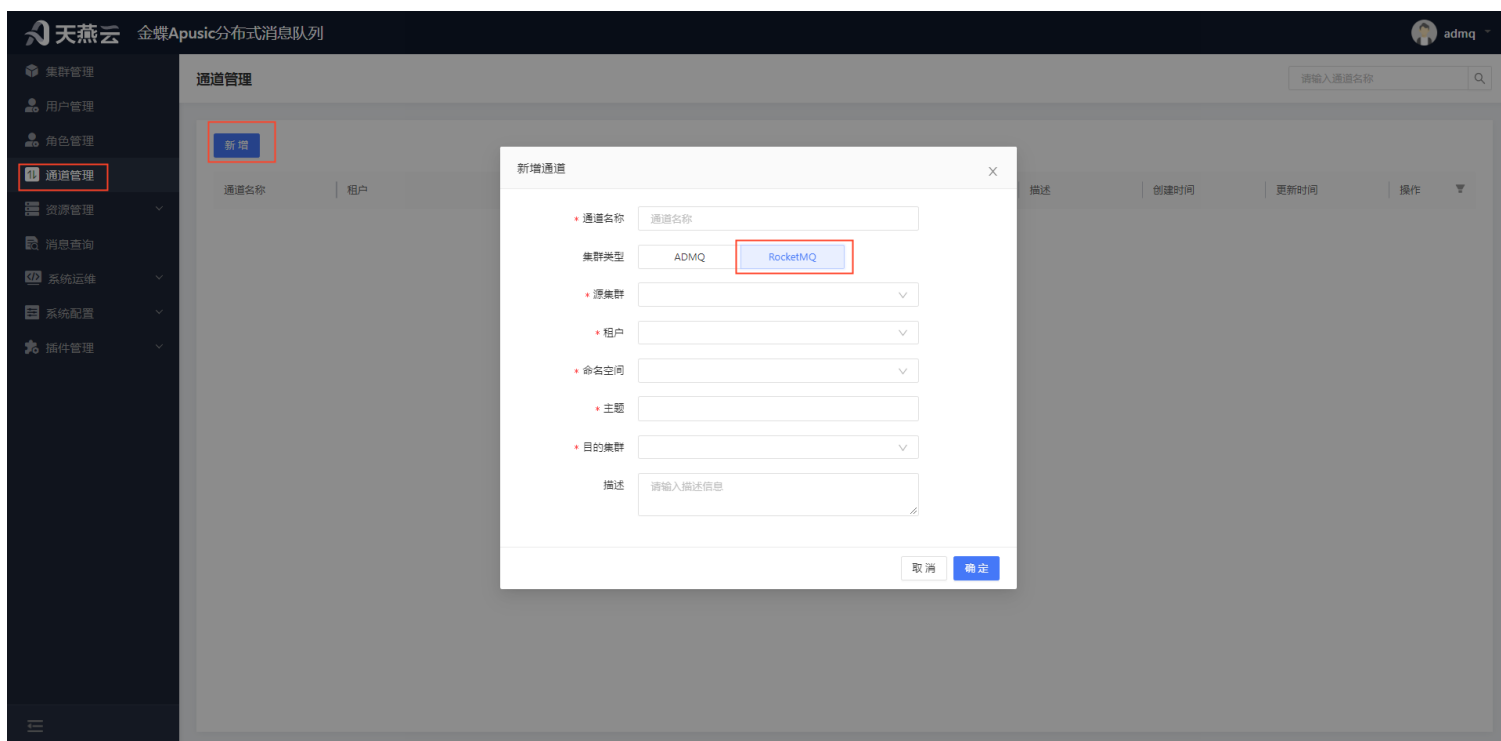
具体流程：在集群列表界面，分别对两个集群，点击更多操作里面的【配置】按钮，进入到集群配置页面，在其他配置模块，设置RocketMQ Connector服务地址：





(3) 创建同步通道

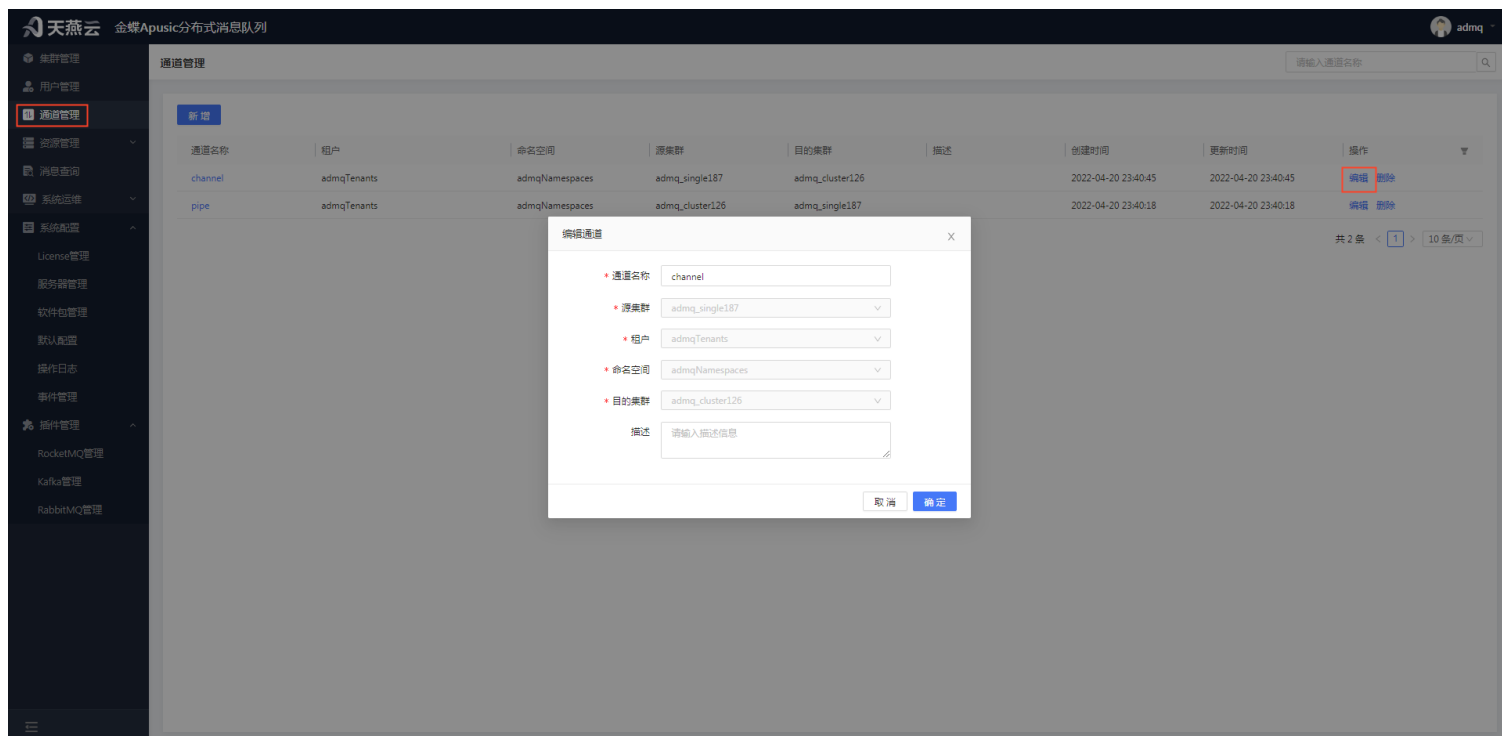
进入【通道管理】，在通道管理页面点击左上角【新增】按钮，可新增通道信息。



- 通道名称：自定义，唯一标识
- 集群类型：选择RocketMQ
- 源集群：被复制消息的集群，下拉选择
- 租户：被复制消息的租户，下拉选择
- 命名空间：被复制消息的命名空间，下拉选择
- 主题：被复制消息的主题，下拉选择
- 目的集群：需复制消息的集群，下拉选择
- 描述：通道描述信息

5.7.1.2 编辑通道

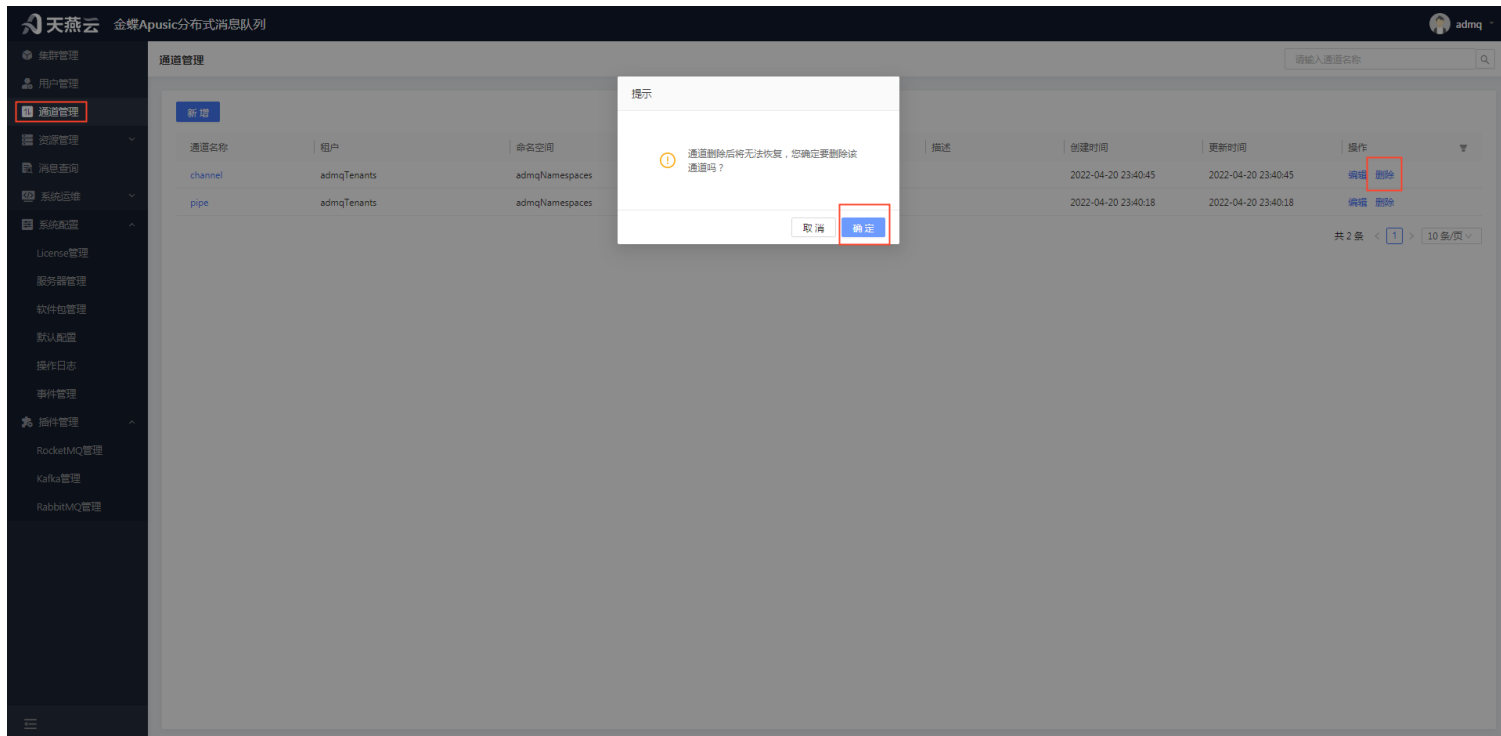
进入【通道管理】，进入通道管理页面，选择需要编辑的通道信息，点击【编辑】按钮，可修改通道信息。



- 通道名称：自定义，唯一标识，可修改
- 源集群：被复制消息的集群,不可修改，置灰显示
- 租户：被复制消息的租户，不可修改，置灰显示
- 命名空间：被复制消息的命名空间,不可修改，置灰显示
- 目的集群：需复制消息的集群,不可修改，置灰显示
- 描述：通道描述信息，可修改

5.7.1.3 删除通道

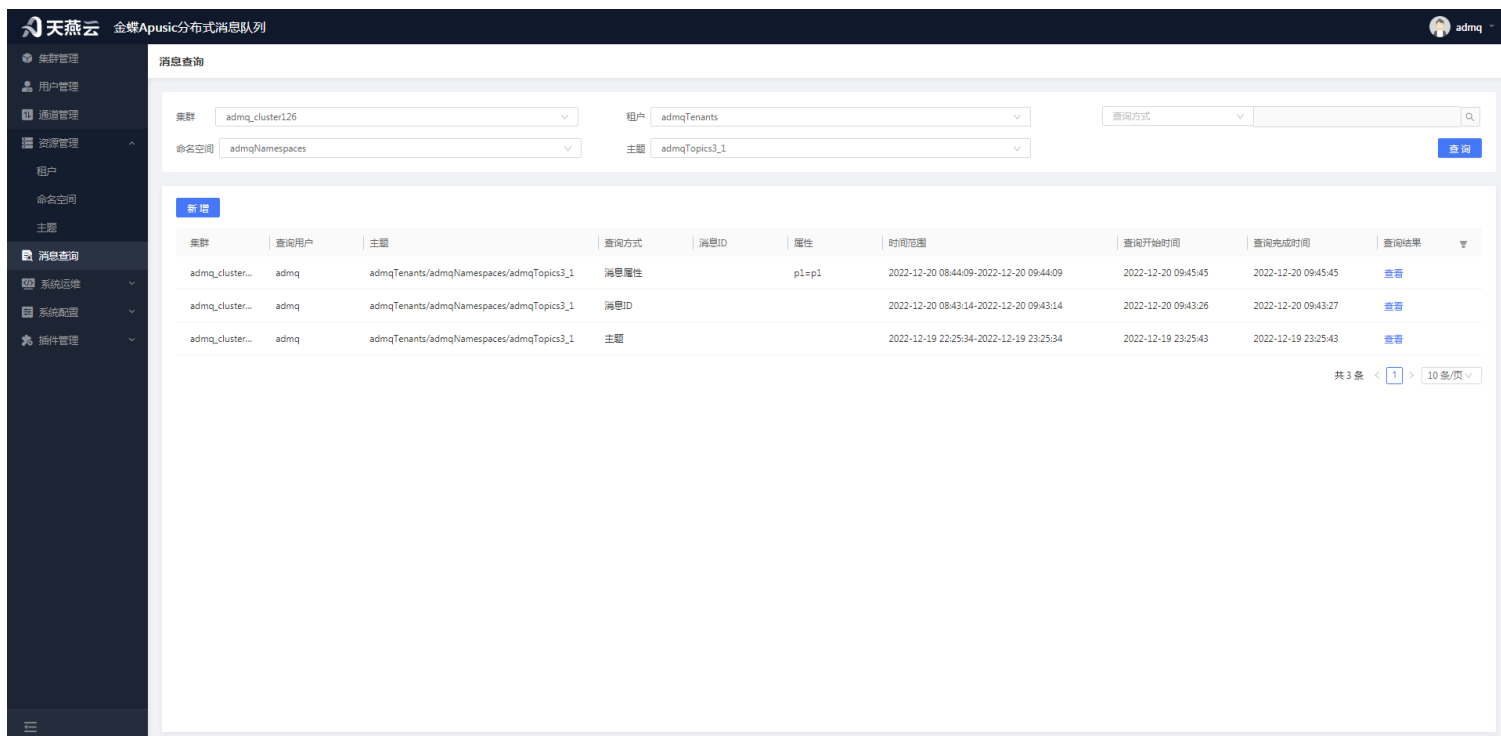
进入【通道管理】，进入通道管理页面，选择需要删除的通道信息，点击【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，删除通道信息。注意：通道信息被删除后，源集群就不能复制消息到目的集群了



5.8 消息查询

5.8.1 消息查询

消息作用于监控查询消息的轨迹，当生产者发送消息时，消费者接收时，可在消息查询中查询到消息轨迹。消息查询用于新增消息查询信息，查询消息信息、查看消息轨迹，其中查询方式可按主题查询、按消息ID查询、按消息属性查询。



注意：

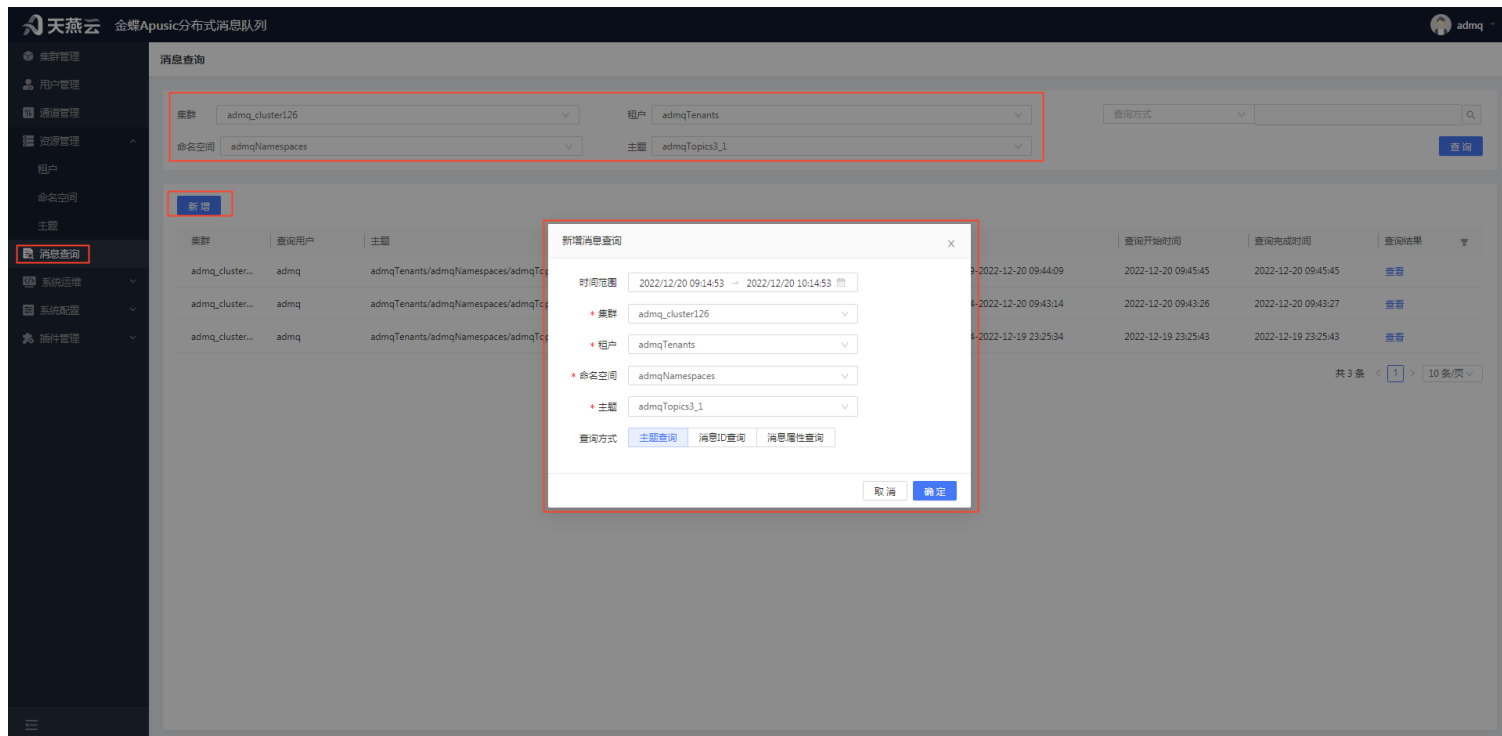
消息查询包含两部分：1. 生产者发送消息到ADMQ 2. ADMQ存储消息、消费者接收消息、消费者接收消息后返回确认。

生产者发送消息到ADMQ，命名空间不需要开启消息轨迹就可以查到。

ADMQ存储消息、消费者接收消息、消费者接收消息后返回确认，需要对主题所在的命名空间开启消息轨迹（参考章节 5.2.9），且需要对集群节点新增配置参数：messageTraceEnabled=true并重启集群服务（参考章节 3.1.16），之后才能查到；除此之外，只有当命名空间开启消息轨迹后创建的主题才会记录消息轨迹。

5.8.1.1 新增消息查询

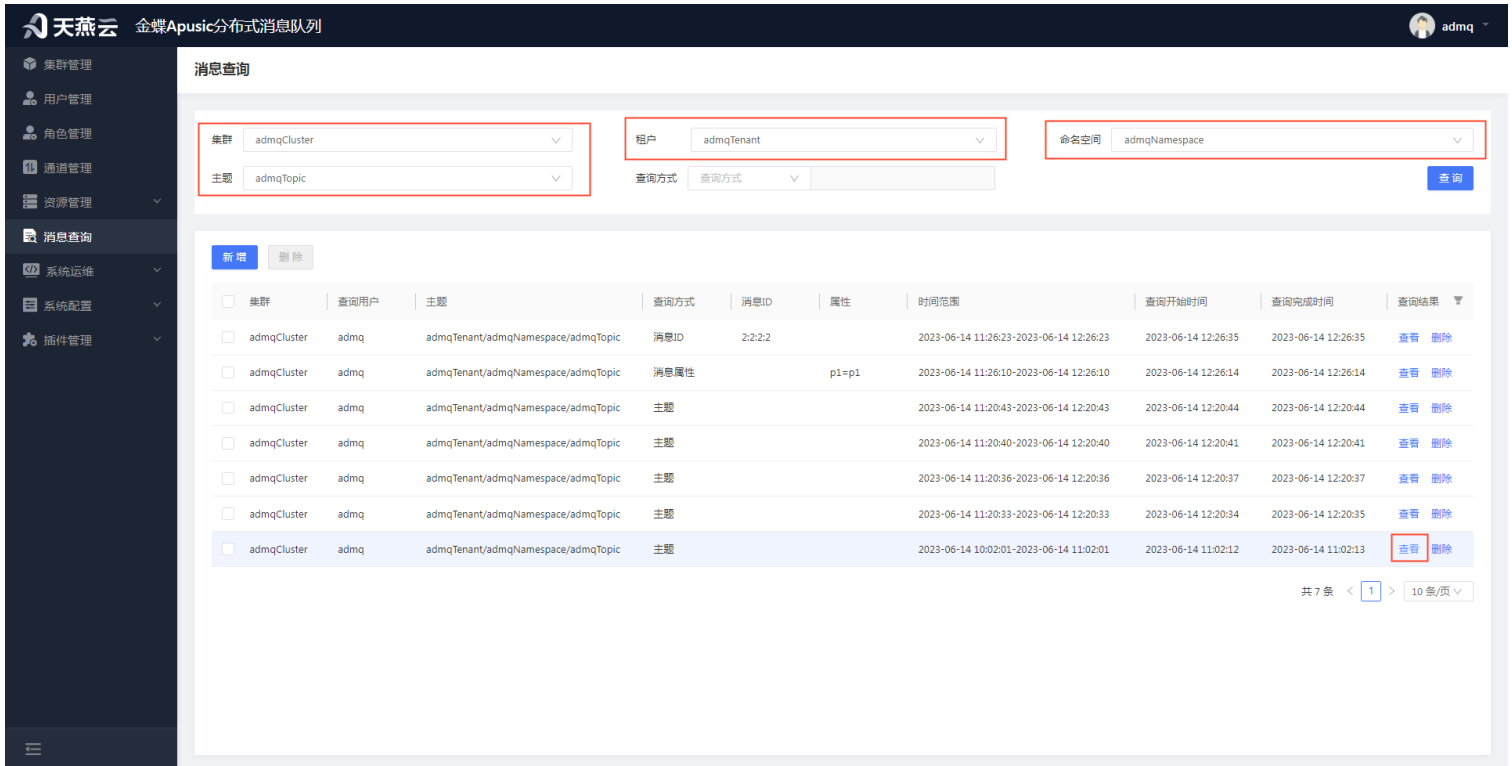
进入【消息查询】，在页面顶端选择需要查询消息所在的集群、租户、命名空间，主题，然后在消息查询页面点击左上角【新增】按钮，可新增消息查询信息。



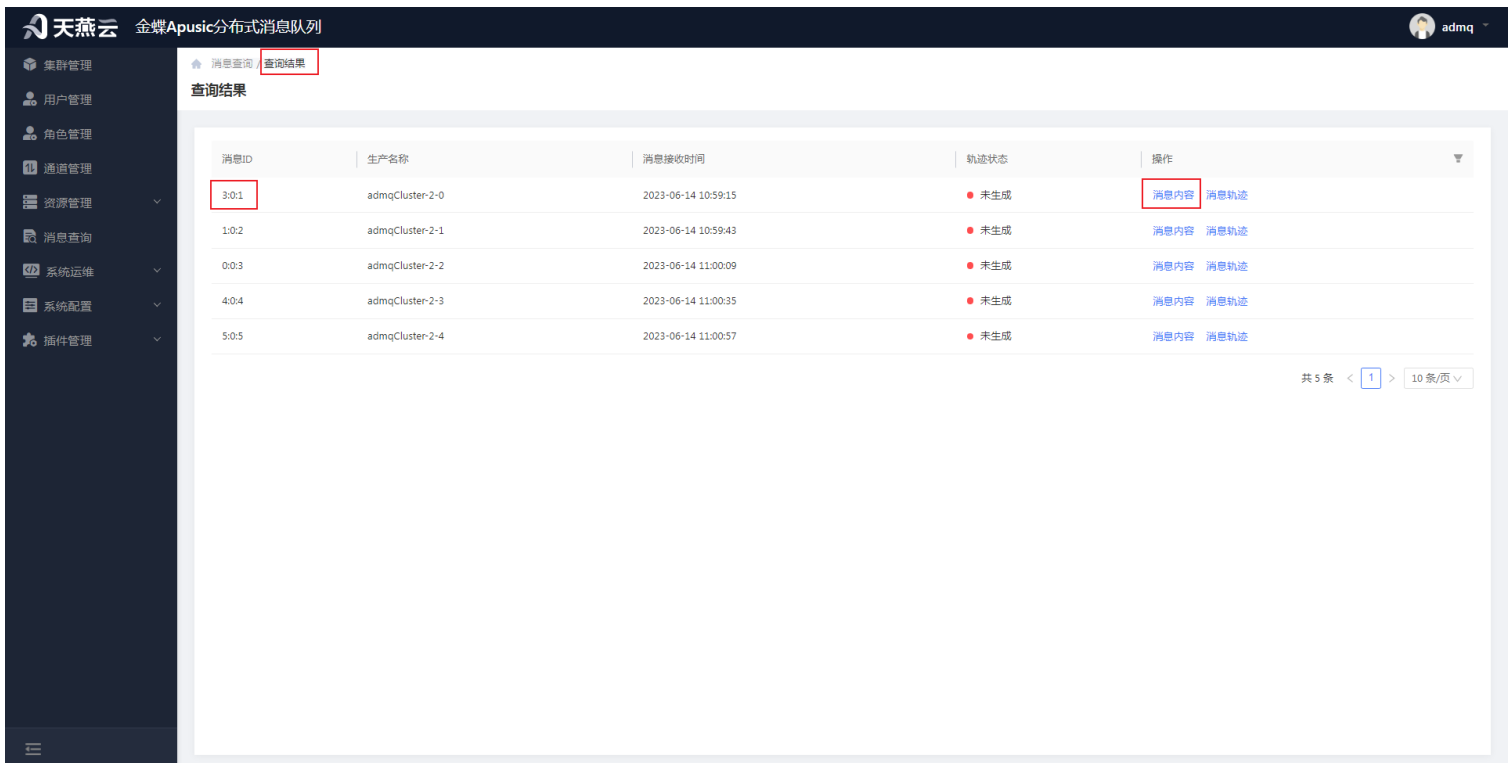
- 时间范围：默认为近1小时的时间范围
- 集群：需查询消息的集群，下拉选择
- 租户：需查询消息的租户，下拉选择
- 命名空间：需查询消息的命名空间，下拉选择
- 主题：需查询消息的主题，下拉选择（当主题较多时，也可模糊输入主题名，可自动带出符合条件的主题）
- 查询方式：单选，包括：主题查询、消息ID查询、消息属性查询。选择主题查询时，直接点击【确定】按钮即可；选择消息ID查询时，可输入消息ID，选填项，用来过滤消息ID，不输入则默认查询主题下所有的消息ID；选择消息属性查询时，需输入属性值，必填项，用来过滤消息属性，格式为name=value，多个属性之间用逗号隔开。

5.8.1.2 查看消息内容

进入【消息查询】，在消息查询页面选择需要查看消息内容的消息查询，点击【查看】按钮：



进入查询结果页面，选择需要查看消息内容的消息ID，点击【消息内容】按钮，可查询对应消息ID的消息内容如：基本信息、消息内容、其他属性值。



The screenshot shows the '消息内容' (Message Content) page in the Apusic management console. The page is divided into several sections:

- 基本信息 (Basic Information):**
 - 主题: persistent://admqTenant/admqNamespace/admqTopic
 - 消息ID: 3:0:1
 - 生产者: admqCluster-2-0
 - 消息接收时间: 2023-06-14 10:59:403
- 消息内容 (Message Content):**
 - 十六进制数据 (Hexadecimal Data):** 6d65737361676556616c756531
 - 解码后数据 (Decoded Data):** messageValue1
- 其他属性 (Other Attributes):**

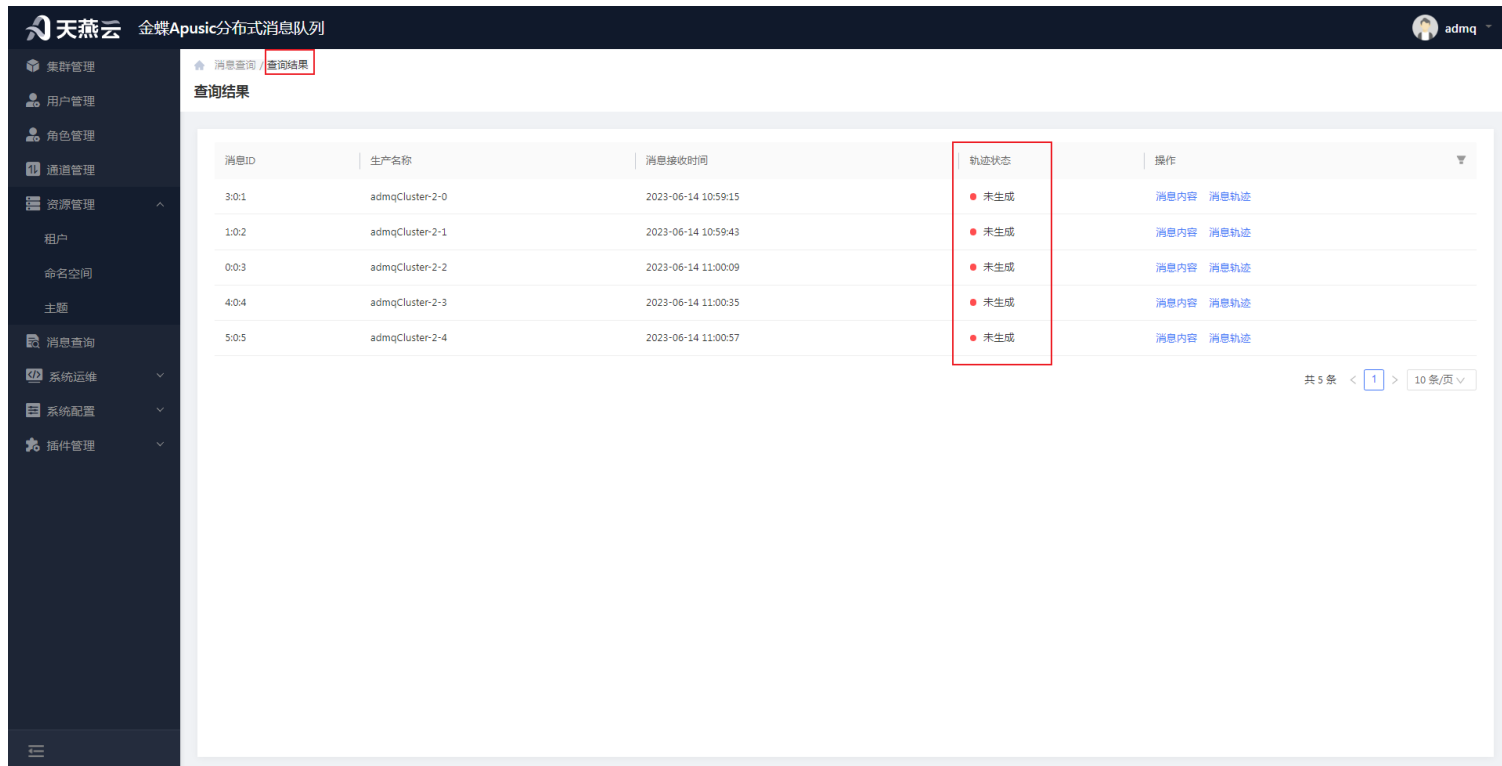
属性值	属性内容
X-Pulsar-batch-size	44
X-Pulsar-num-batch-message	1

5.8.1.3 查看消息轨迹

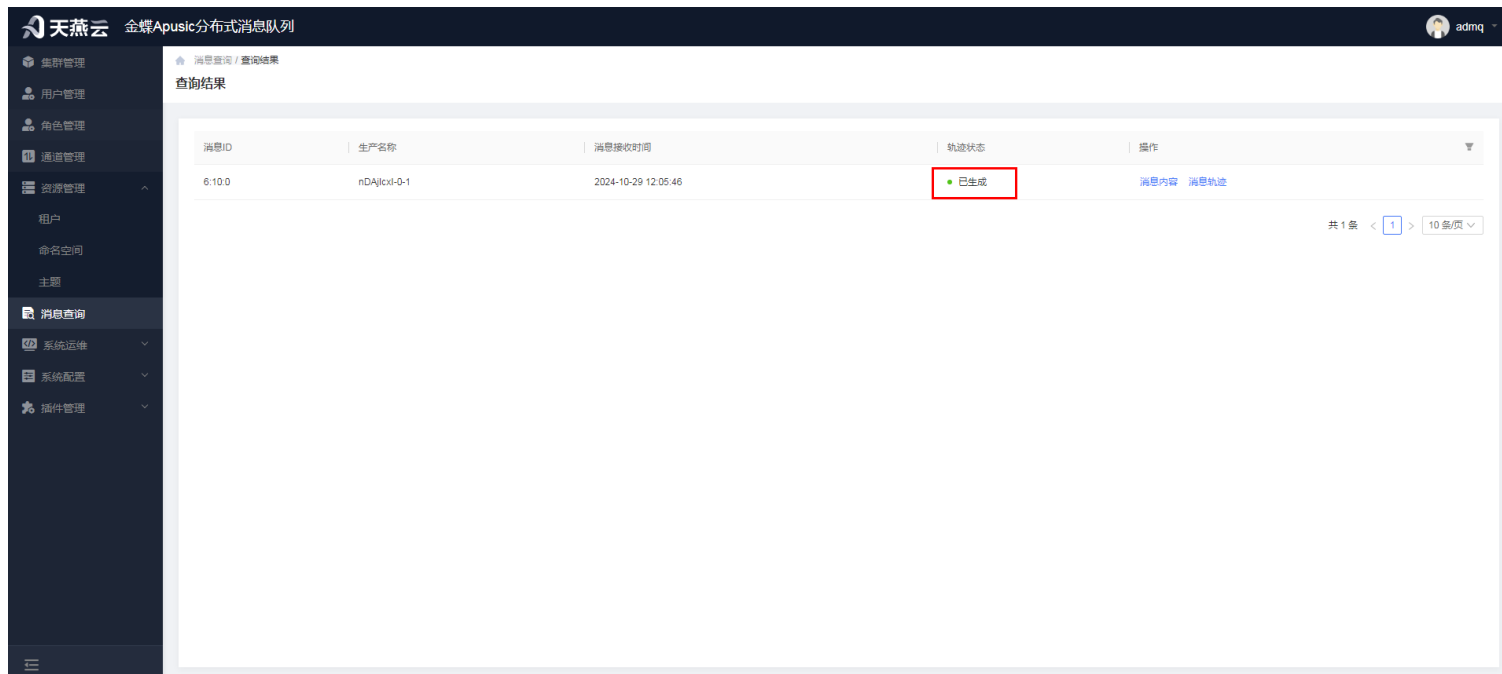
新创建的消息查询会记录消息轨迹的前提条件:

- (1) 需要对集群计算节点新增配置参数: messageTraceEnabled=true并重启集群服务 (参考章节 3.1.16)
- (2) 需要对命名空间开启消息轨迹然后再在该命名空间下创建主题并对其发送消息 (参考章节 5.2.9) ;
- (3) 生产者发送消息且消费者消费消息。

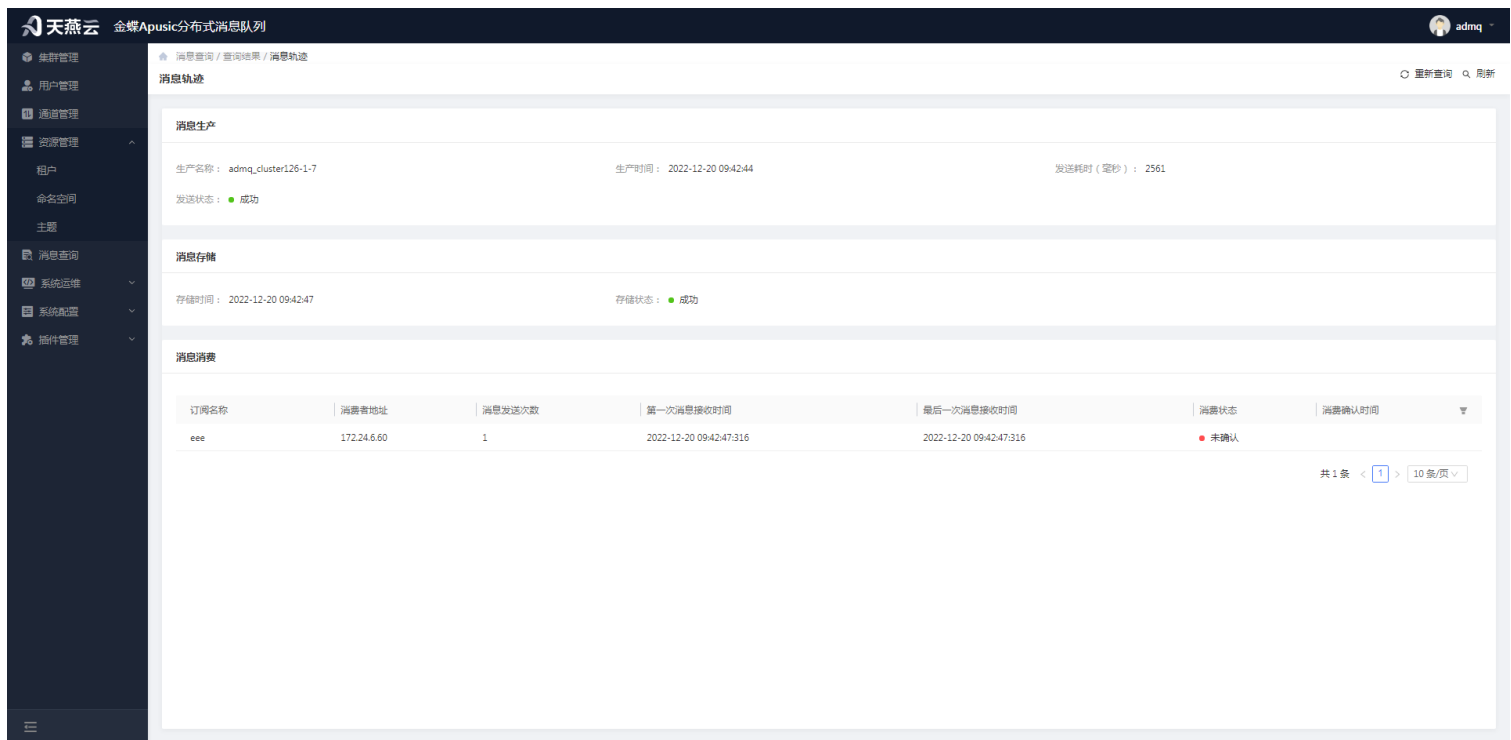
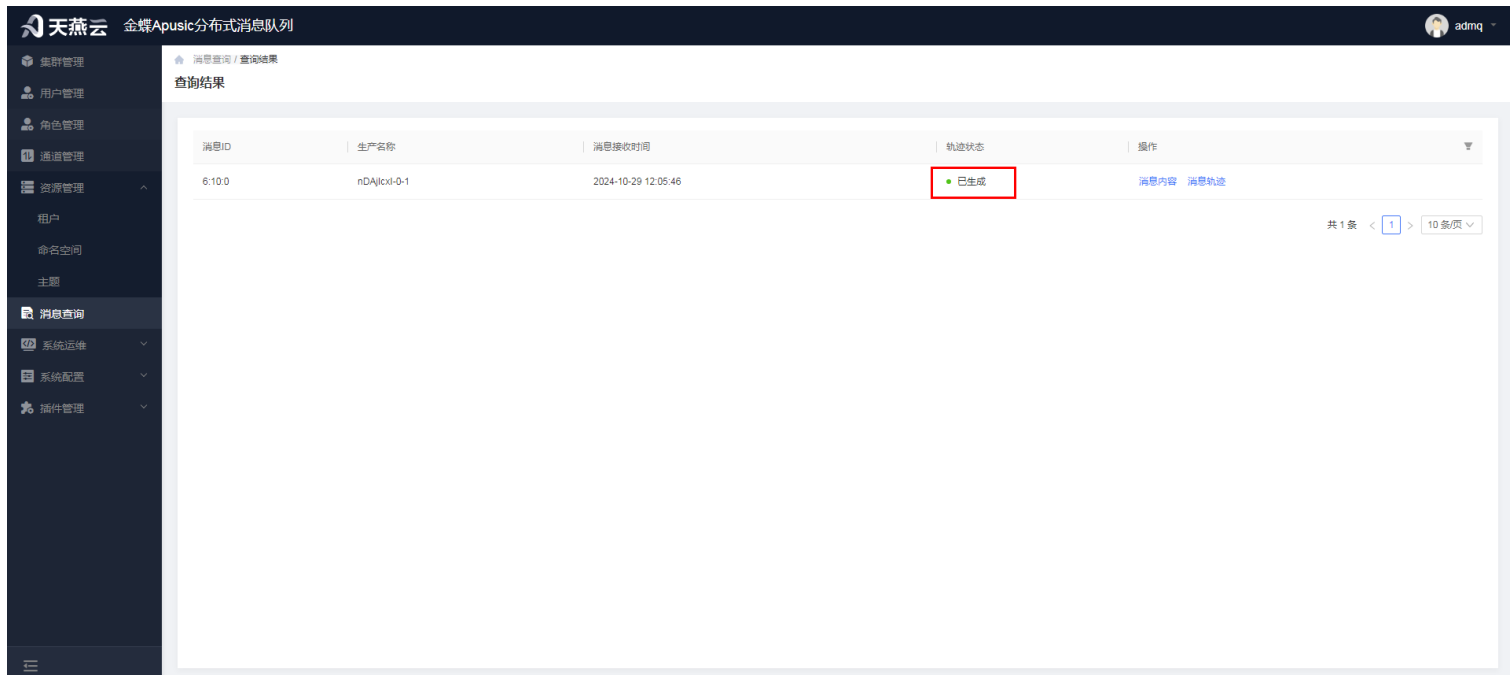
备注: 若未设置前提条件, 在消息查询列表页面, 选择某一条消息查询, 点击【查看】按钮进入到查询结果页面, 轨迹状态一栏显示“未生成”:



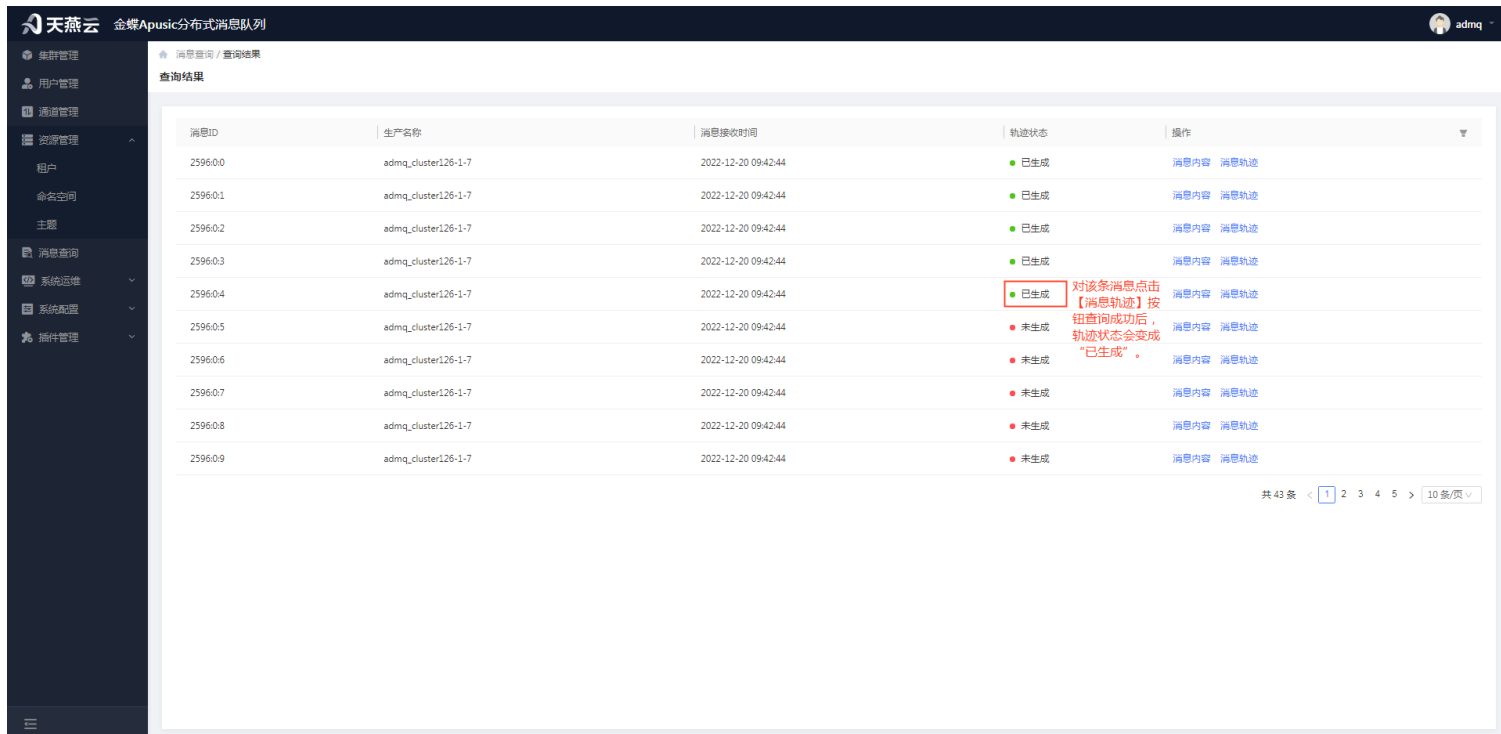
前提条件设置后，新增消息查询后，点击【查看】按钮进入到查询结果页面，轨迹状态一栏显示“已生成”：



进入查询结果页面，选择需要查看轨迹的消息ID，点击【消息轨迹】按钮，可查询对应消息ID的消息轨迹信息，轨迹信息包含消息生产、消息存储、消息消费的信息。

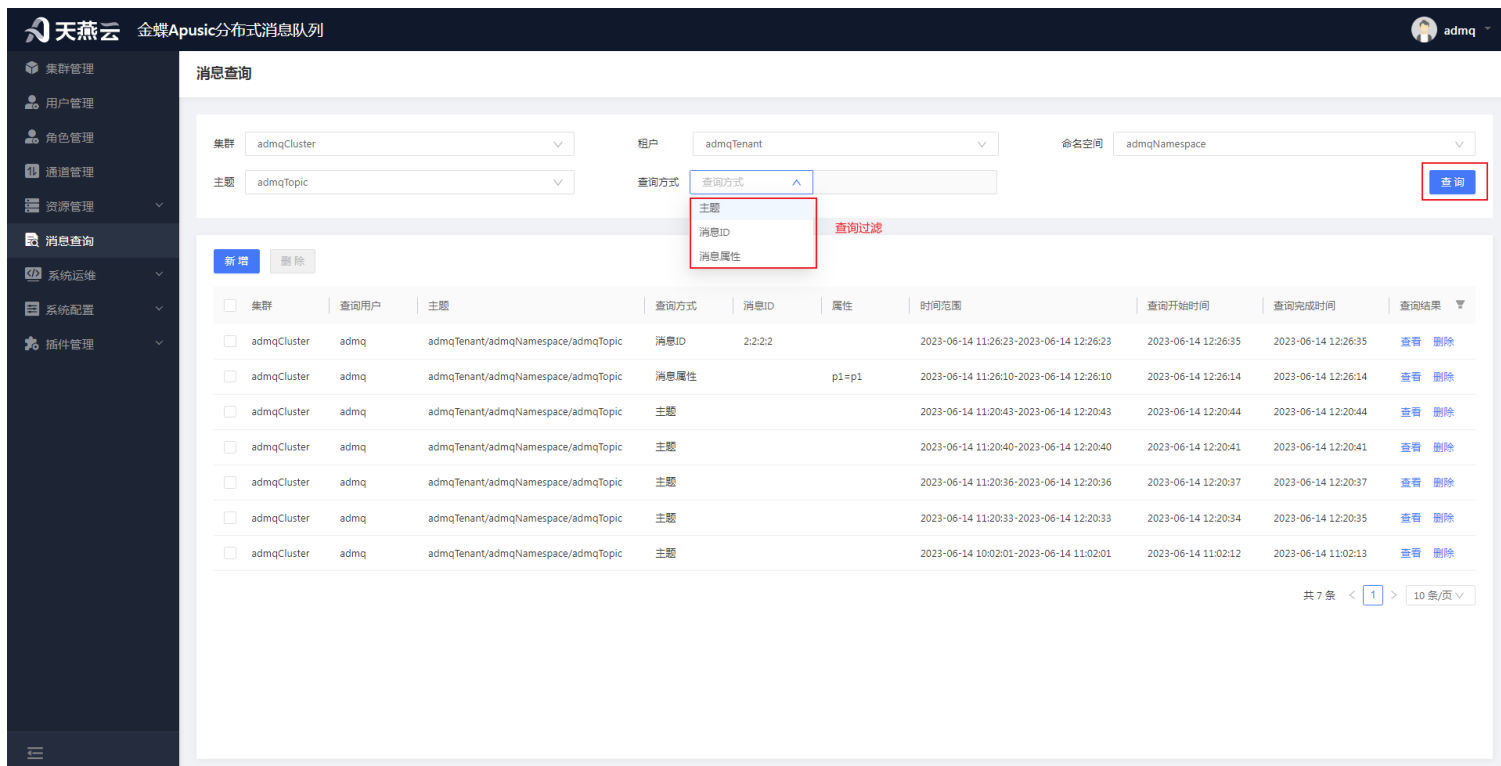


注意，查询消息轨迹后，在查询结果页面，消息的轨迹状态会从“未生成”变成“已生成”，如下：



5.8.1.4 查询已创建的消息查询

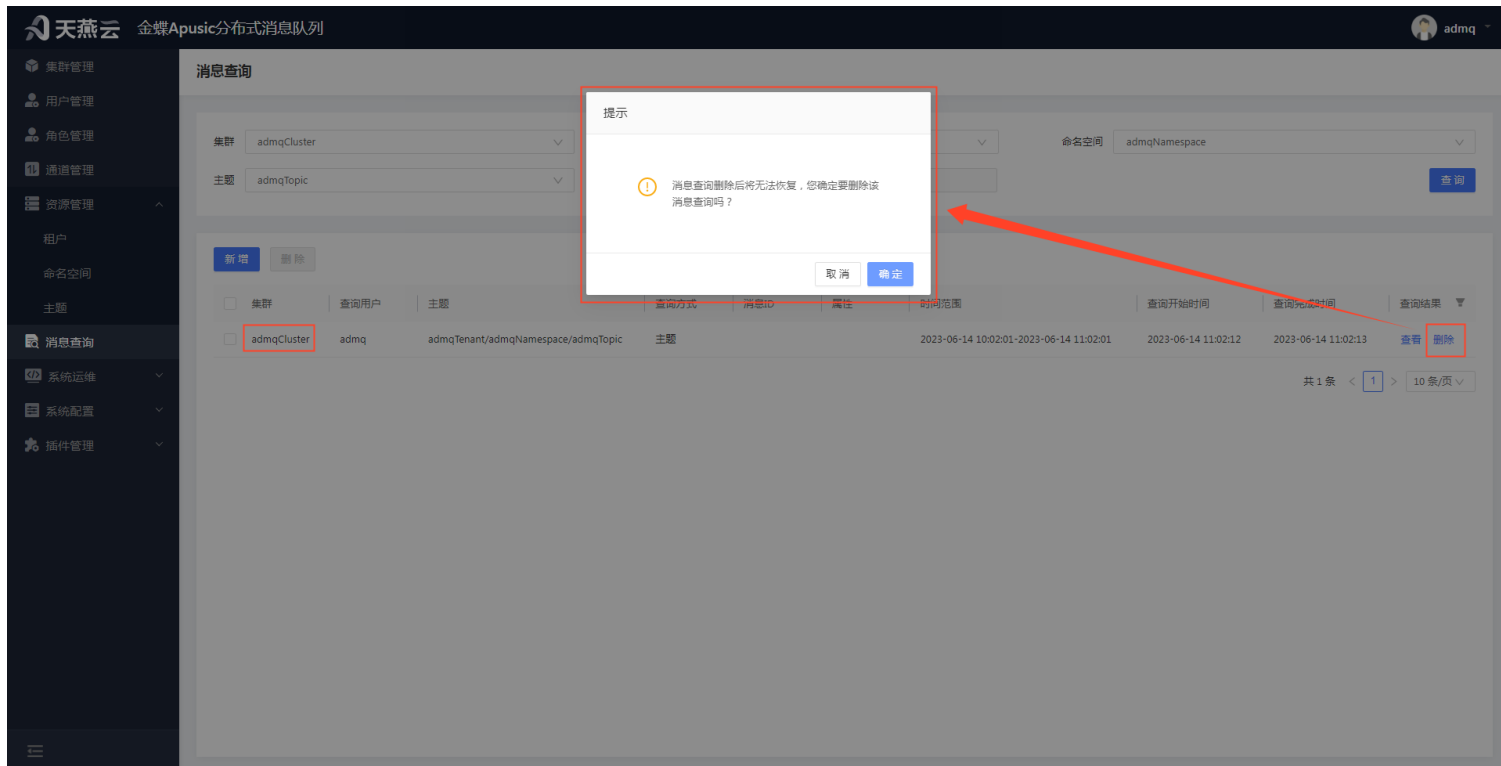
进入【消息查询】，在消息查询页面右上角选择集群、租户、命名空间、主题名称后，可通过查询方式进行过滤（也可不过滤，默认全部），点击【查询】按钮，可查询消息查询信息。



5.8.1.5 删除已创建的消息查询

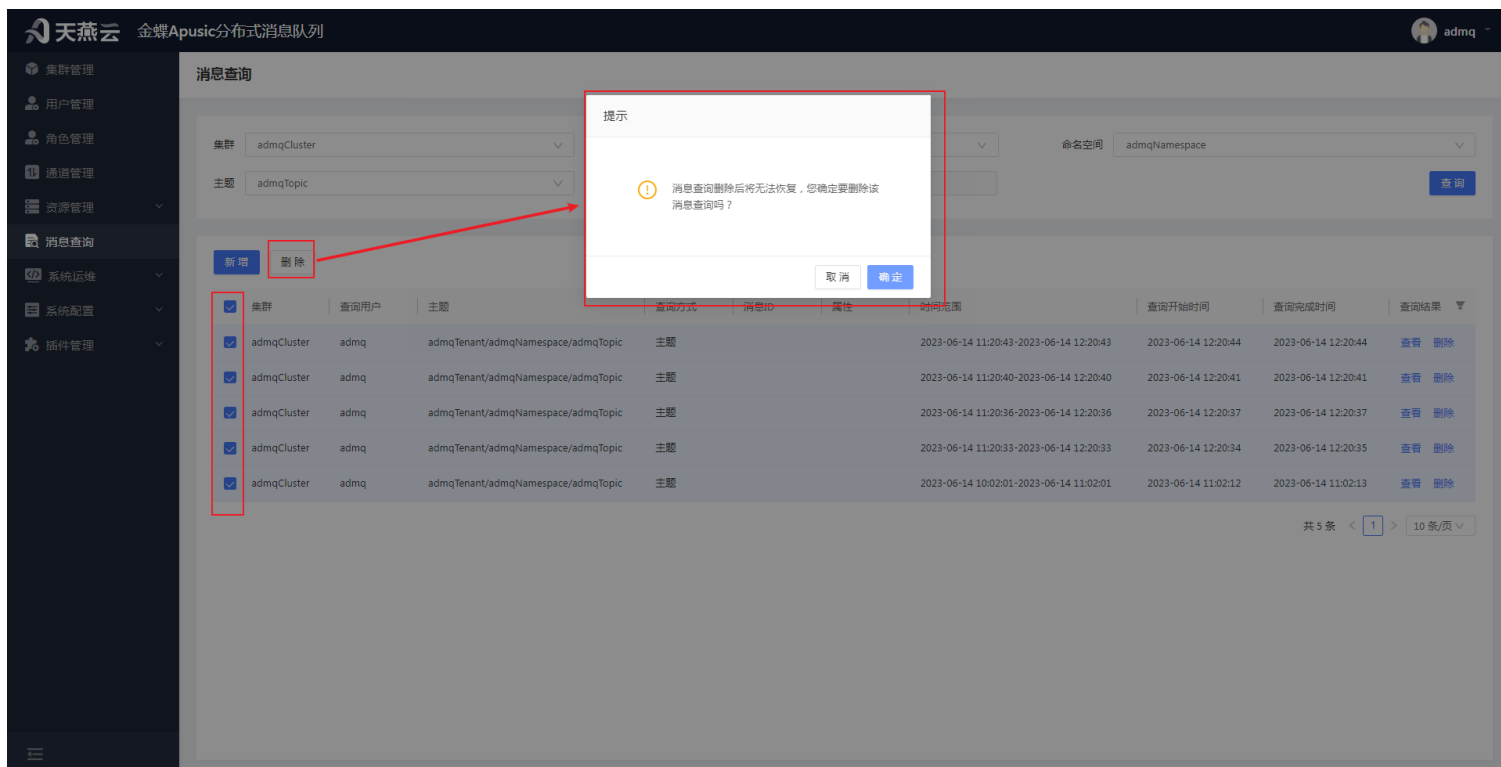
(1) 单个删除

进入【消息查询】，在消息查询页面的上方模块中选择集群、租户、命名空间、主题后，然后在下方模块的消息查询列表页面选择需要删除的消息查询，点击【删除】，出现提示弹窗，点击【确定】按钮，进行删除操作。



(2) 批量删除

进入【消息查询】，在消息查询页面的上方模块中选择集群、租户、命名空间、主题后，然后在下方模块的消息查询列表页面选择需要删除的消息查询，选中左侧的单选框，然后点击右上方的【删除】按钮，出现提示弹窗，进行批量删除操作。

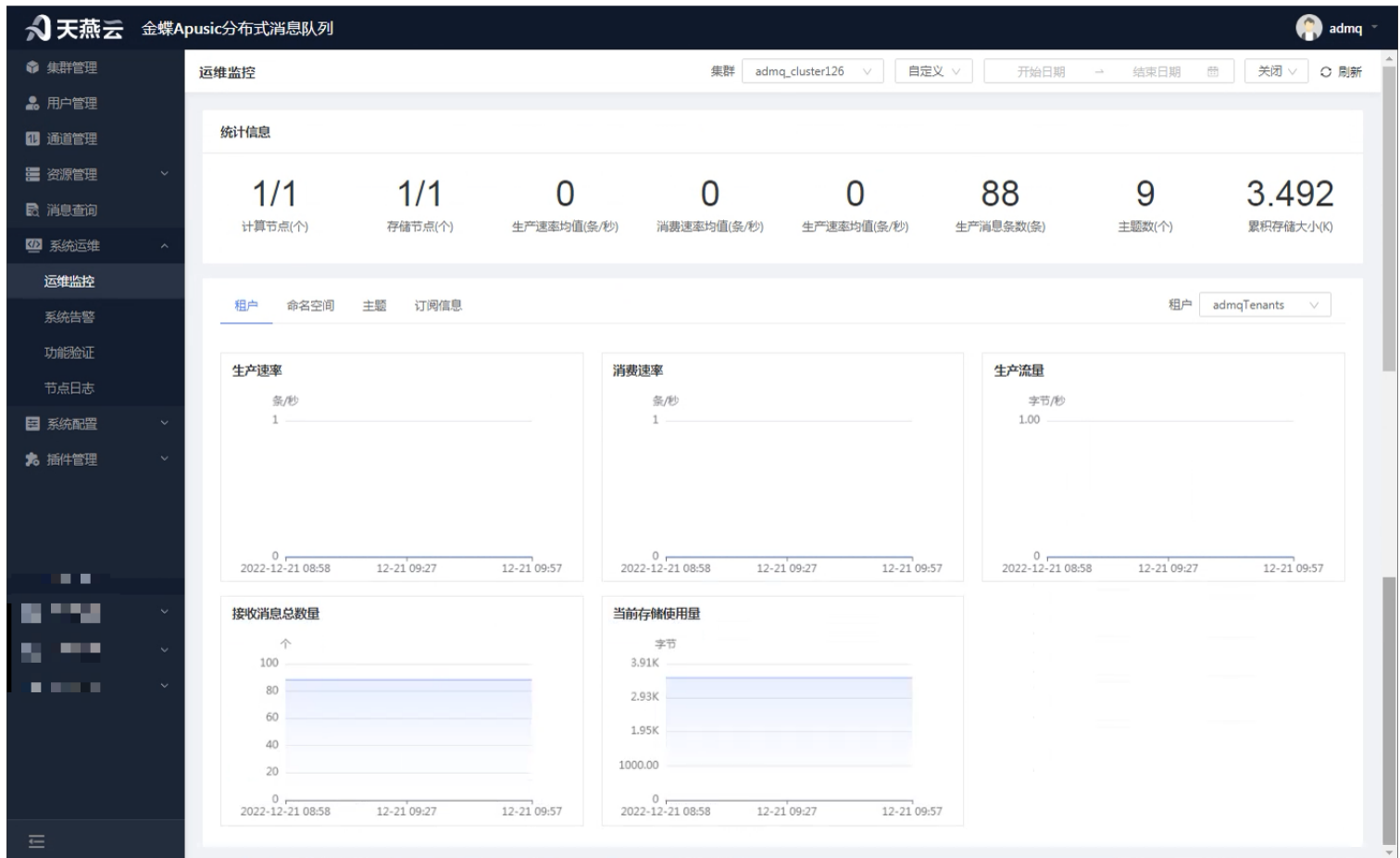


5.9 系统运维

系统运维用于监控管理系统，如节点服务器信息、消息信息等。有运维监控、系统告警、功能验证、节点日志。

5.9.1 运维监控

进入【系统运维】->【运维监控】，可查询到运维监控信息。运维监控用于监控管理集群信息，可以监控近1小时内消息的统计信息、近24小时的消息统计信息、近3天的消息统计信息、近7天的消息统计信息、也可自定义查询，还能根据租户、命名空间、主题、订阅消息，查询消息的生产速率、消费速率、生产流量、消费流量、生产者数量、主题数量、接收消息总数量、当前存储使用量等，并以图表的形式直观显示。



5.9.2 系统告警

(1) ADMQ管控台本身的告警功能

进入【系统运维】->【系统监控】，可监控到集群以及集群各节点的状态，并以告警事件的形式通知客户；可以统计近1小时内的告警数量、近24小时的告警数量、近3天的告警数量、近7天的告警数量、也可自定义查询近1个月的告警数量；也可根据告警级别、告警主体、告警状态、告警类别进行查询告警事件。

天燕云 金蝶Apusic分布式消息队列 admq

系统告警 集群 admq_cluster126 自定义 2022-12-19 11:11 2022-12-21 11:11 关闭 刷新

统计信息

1	0	1	0	0
总告警数(条)	一级告警(条)	二级告警(条)	三级告警(条)	四级告警(条)

告警事件

告警级别 告警主体 告警状态 未处理 告警类别

告警级别	告警服务器	告警主体	告警类别	告警次数	告警状态	最近一次告警时间	操作
二级告警	172.24.4.126	集群	集群管理服务连接失败	1	● 未处理	2022-12-19 23:15:30	查看 解决

共 1 条 < 1 > 10 条/页

可点击【查看】按钮查看具体的告警信息：

天燕云 金蝶Apusic分布式消息队列 admq

系统告警 集群 admq_cluster126 自定义 2022-12-19 11:11 2022-12-21 11:11 关闭 刷新

统计信息

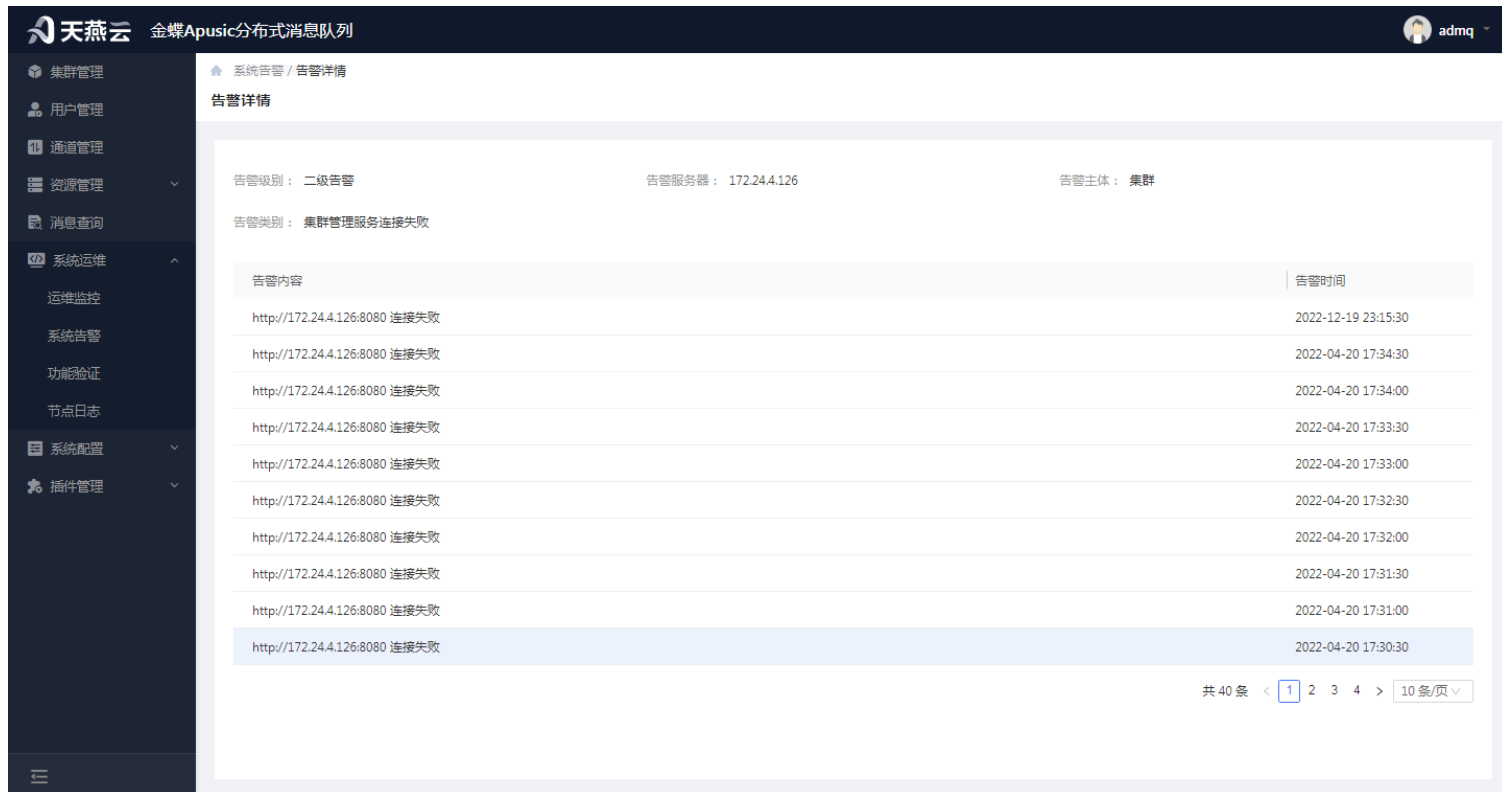
1	0	1	0	0
总告警数(条)	一级告警(条)	二级告警(条)	三级告警(条)	四级告警(条)

告警事件

告警级别 告警主体 告警状态 未处理 告警类别

告警级别	告警服务器	告警主体	告警类别	告警次数	告警状态	最近一次告警时间	操作
二级告警	172.24.4.126	集群	集群管理服务连接失败	1	● 未处理	2022-12-19 23:15:30	查看 解决

共 1 条 < 1 > 10 条/页



(2) 与AMP智能告警系统集成的主动告警功能

可通过AMP告警接口主动发送系统告警信息，具体操作：

(1) 修改管控台配置文件并重启管控台：

```
amp.monitor.push.enable=true
```

```
amp.warn.url=http://172.24.4.159:9016
```

```
amp.key=6c9be976-4ab5-44fd-a174-d3752fd2e37d
```

(2) 访问AMP的智能告警系统，会出现相关的告警信息。

天燕云 首页 产品 基础设施监控 应用性能管理 运维工单系统 智能告警系统 工单 admin

智能告警

告警事件

事件 / 告警事件

智能告警ADMQ230 x 处理中 x 待处理 x 事件级别 开始日期 结束日期 输入事件内容/事件ID查询 更多

事件认领 关闭

事件ID	事件内容	监控应用	生成方式	关联对象	事件级别	创建时间	状态	工单跟踪	操作
202305111654-306302	http://172.24.4.230:8080 连接失败	智能告警ADM...	规则检测	集群	严重	2023-05-11 16:54:30	待处理		认领 关闭
202305111649-308747	http://172.24.4.233:8080 连接失败	智能告警ADM...	规则检测	集群	严重	2023-05-11 16:49:30	待处理		认领 关闭
202305111649-159744	http://172.24.4.232:8080 连接失败	智能告警ADM...	规则检测	集群	严重	2023-05-11 16:49:15	待处理		认领 关闭
202305111641-007798	进程不存在	智能告警ADM...	规则检测	admj计算程序	严重	2023-05-11 16:41:01	待处理		认领 关闭
202305111616-007241	http://172.24.4.231:8080 连接失败	智能告警ADM...	规则检测	集群	严重	2023-05-11 16:16:00	待处理		认领 关闭
202305091130-149136	节点admj计算程序(172.24.4.231)无法连接节...	智能告警ADM...	规则检测	集群	严重	2023-05-09 11:30:15	待处理		认领 关闭
202305091130-151254	节点admj计算程序(172.24.4.232)无法连接节...	智能告警ADM...	规则检测	集群	严重	2023-05-09 11:30:15	待处理		认领 关闭
202305091130-156096	节点admj计算程序(172.24.4.233)无法连接节...	智能告警ADM...	规则检测	集群	严重	2023-05-09 11:30:15	待处理		认领 关闭
202305091127-291021	http://172.24.4.229:8080 连接失败	智能告警ADM...	规则检测	集群	严重	2023-05-09 11:27:30	待处理		认领 关闭
202305081723-459626	http://172.24.4.231:8080 连接失败	智能告警ADM...	规则检测	集群	严重	2023-05-08 17:23:45	待处理		认领 关闭

共3页 < 1 2 3 > 10条/页

5.9.3 功能验证

功能验证作用于验证admj功能是否可正常使用，应用场景主要用于产品升级后。功能验证用于新增功能验证、编辑功能验证、执行功能验证、查看历史执行功能验证信息、删除功能验证。

天燕云 金蝶Apusic分布式消息队列 admin

功能验证

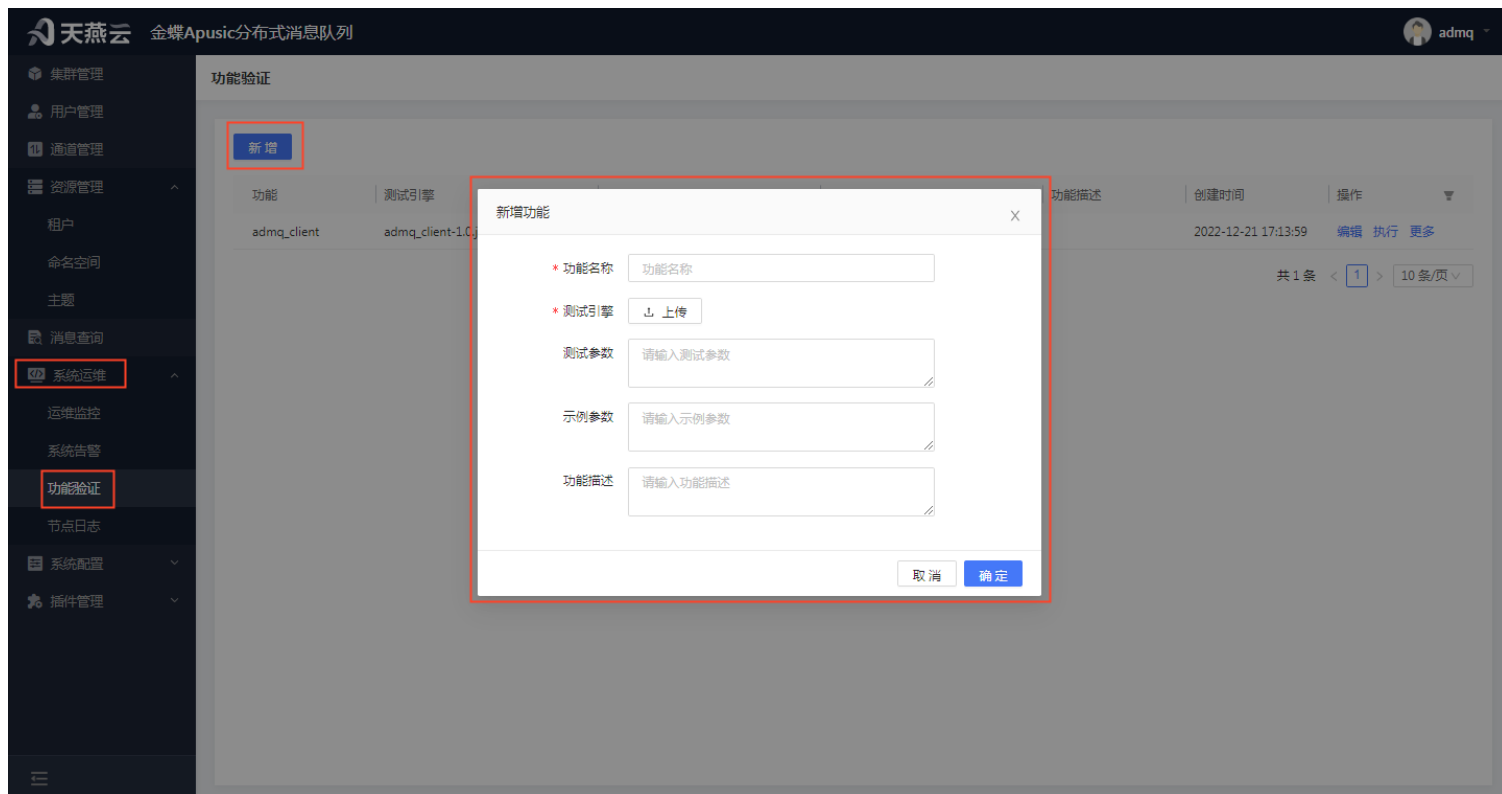
新增

功能	测试引擎	测试参数	示例参数	功能描述	创建时间	操作
admj_client	admj_client-1.0.jar				2022-12-21 17:13:59	编辑 执行 更多

共1条 < 1 > 10条/页

5.9.3.1 新增功能验证

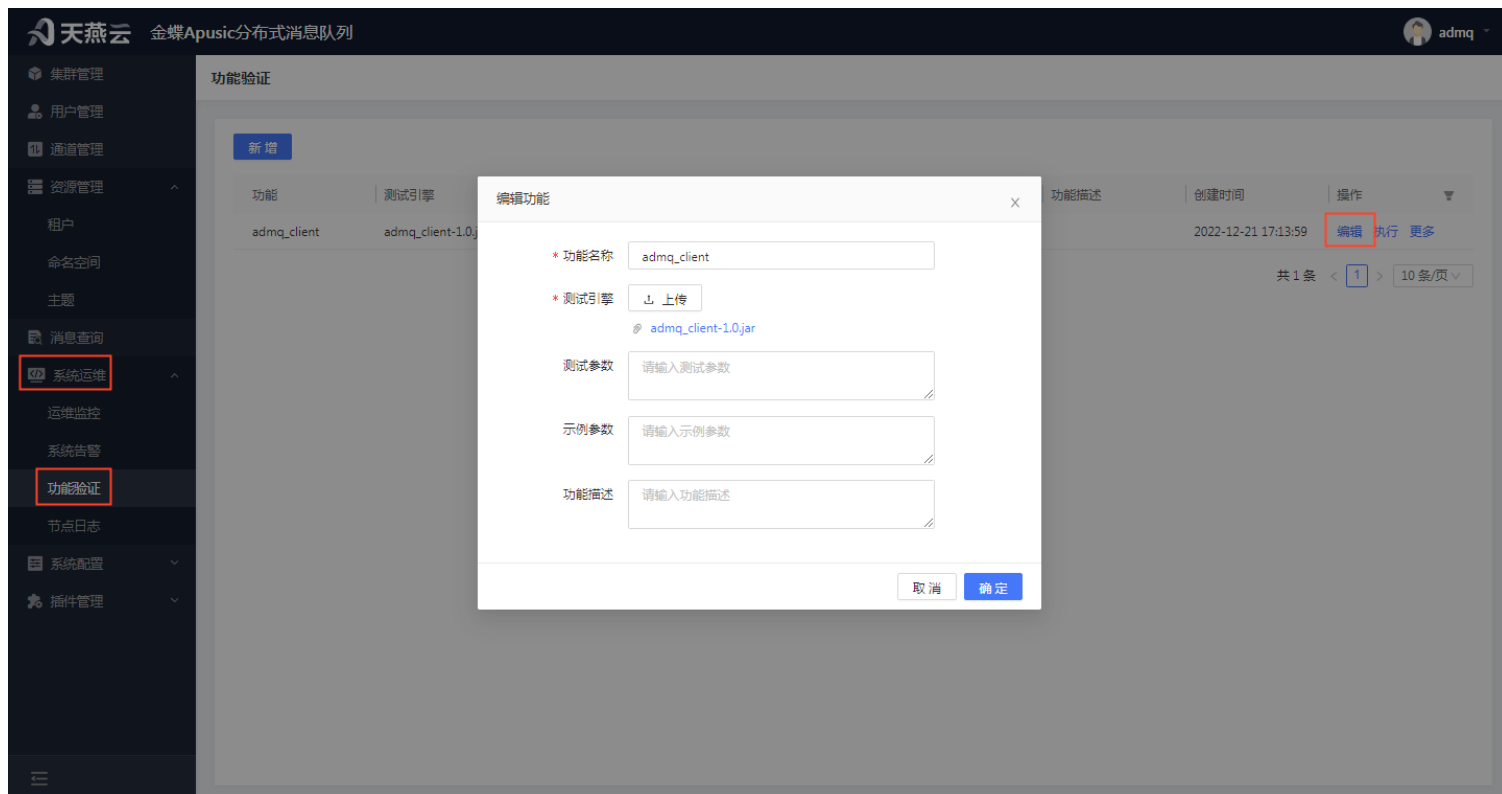
进入【系统运维】->【功能验证】，在功能验证页面点击左上角【新增】按钮，可新增功能验证信息。



- 功能: 功能名称, 自定义
- 测试引擎: 上传的测试引擎文件
- 测试参数: 功能的参数, 自定义, 不可随便写入, 写入错误, 可能执行功能验证会失败
- 示例参数: 功能的示例参数, 自定义
- 功能描述: 功能的描述信息

5.9.3.2 编辑功能验证

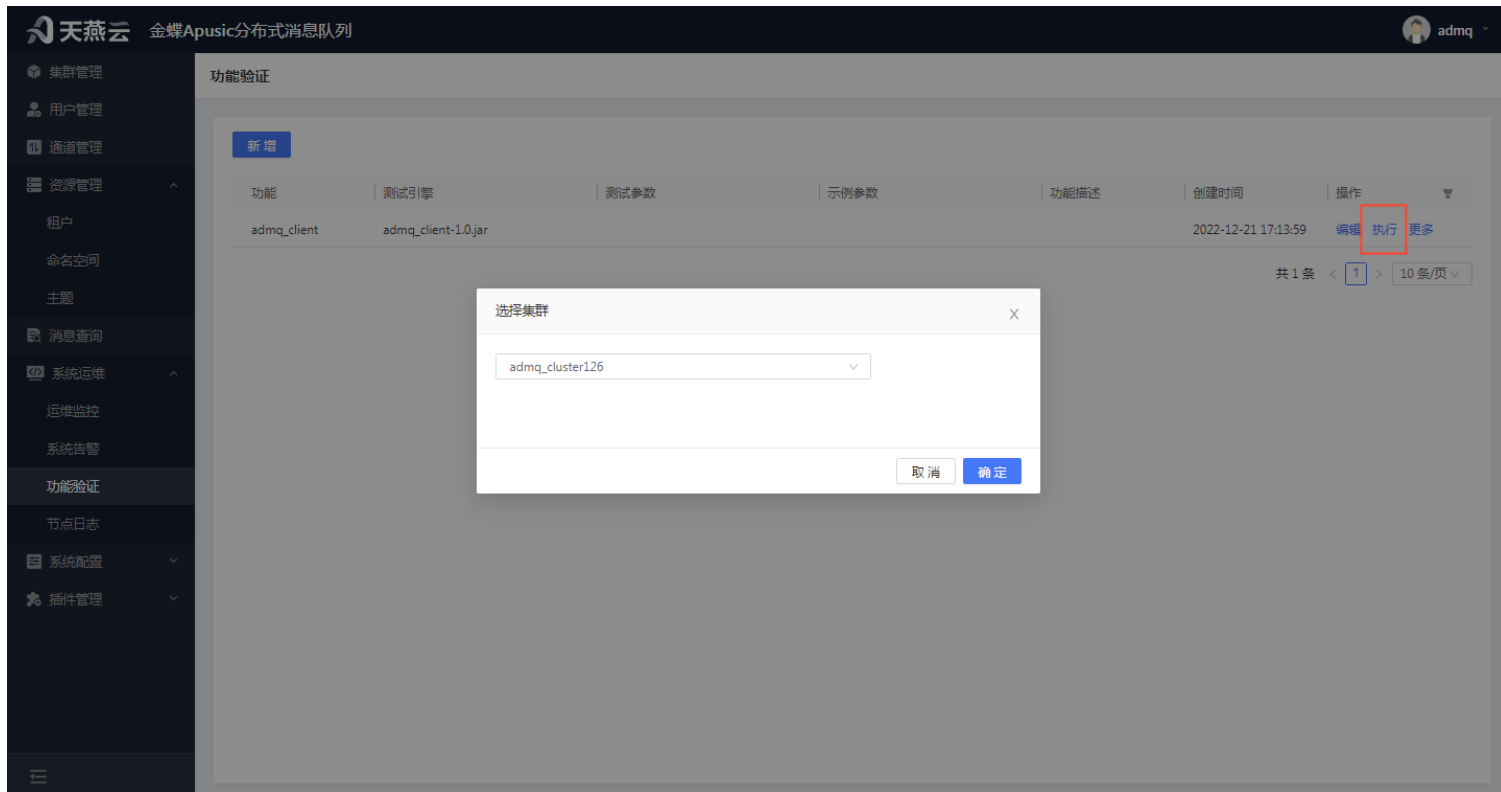
进入【系统运维】->【功能验证】，在功能验证页面选择需要修改的功能，点击【编辑】按钮，可编辑功能验证信息。



- 功能：功能名称，自定义，可修改
- 测试引擎：上传的测试引擎文件,可重新上传引擎文件覆盖原来上传的
- 测试参数：功能的参数，自定义，不可随便写入，写入错误，可能执行功能验证会失败，可修改
- 示例参数：功能的示例参数，自定义，可修改
- 功能描述：功能的描述信息，自定义，可修改

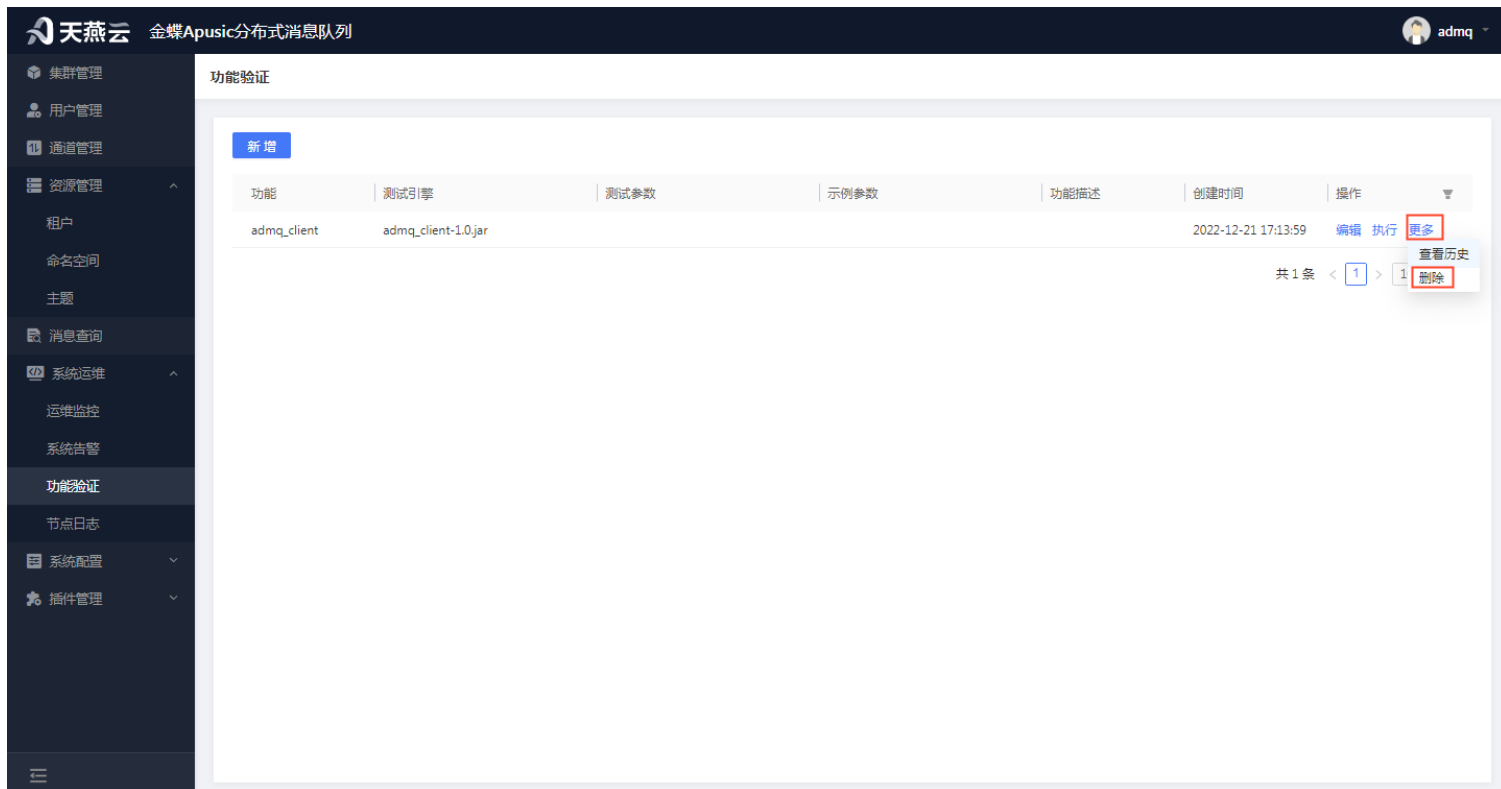
5.9.3.3 执行功能验证

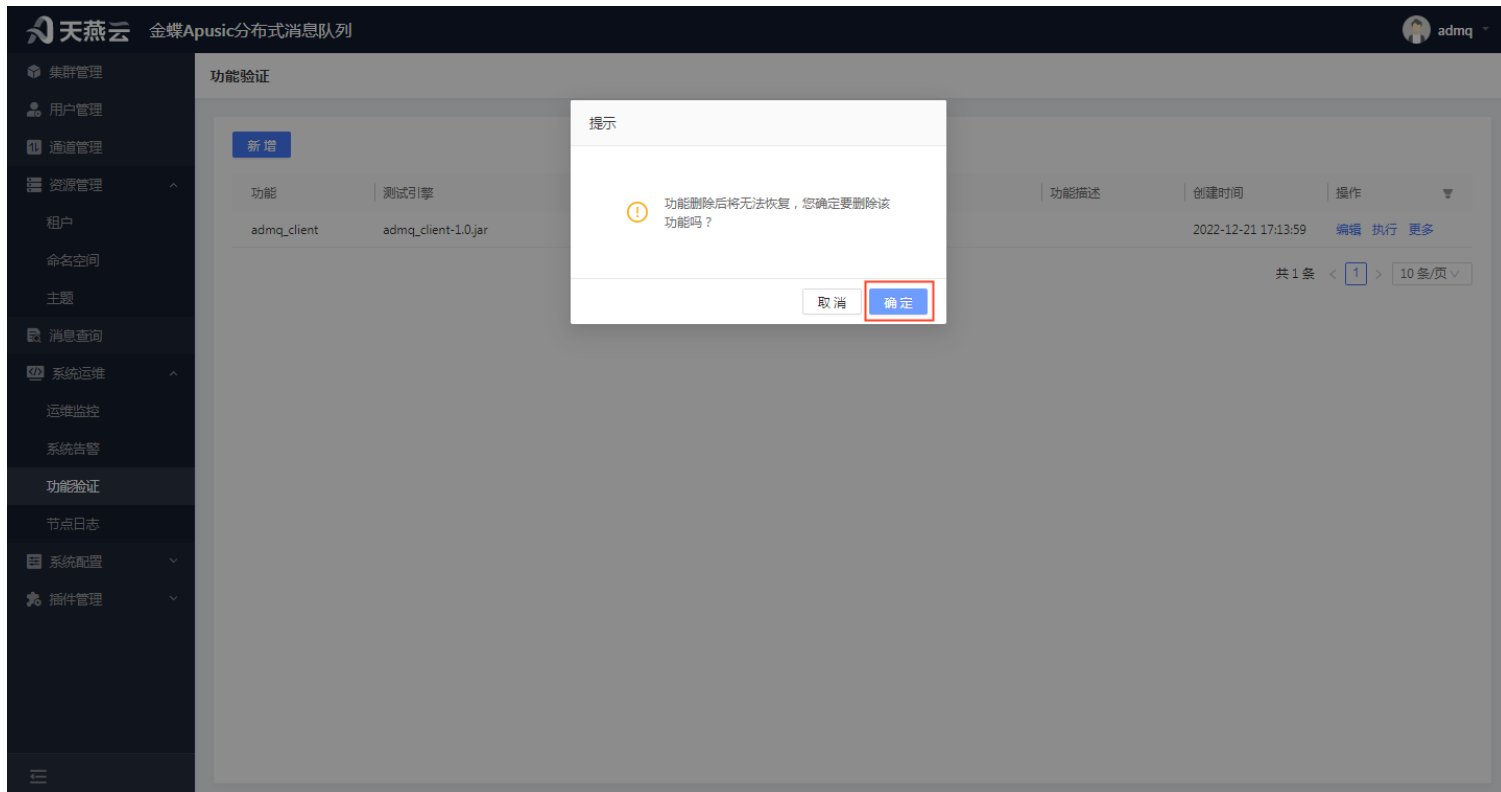
进入【系统运维】->【功能验证】，在功能页面选择需要执行的功能验证信息，点击【更多】按钮->【执行】按钮，在弹出的选择集群弹框中，下拉选择集群执行功能验证信息，可执行功能验证信息。



5.9.3.4 删除功能验证

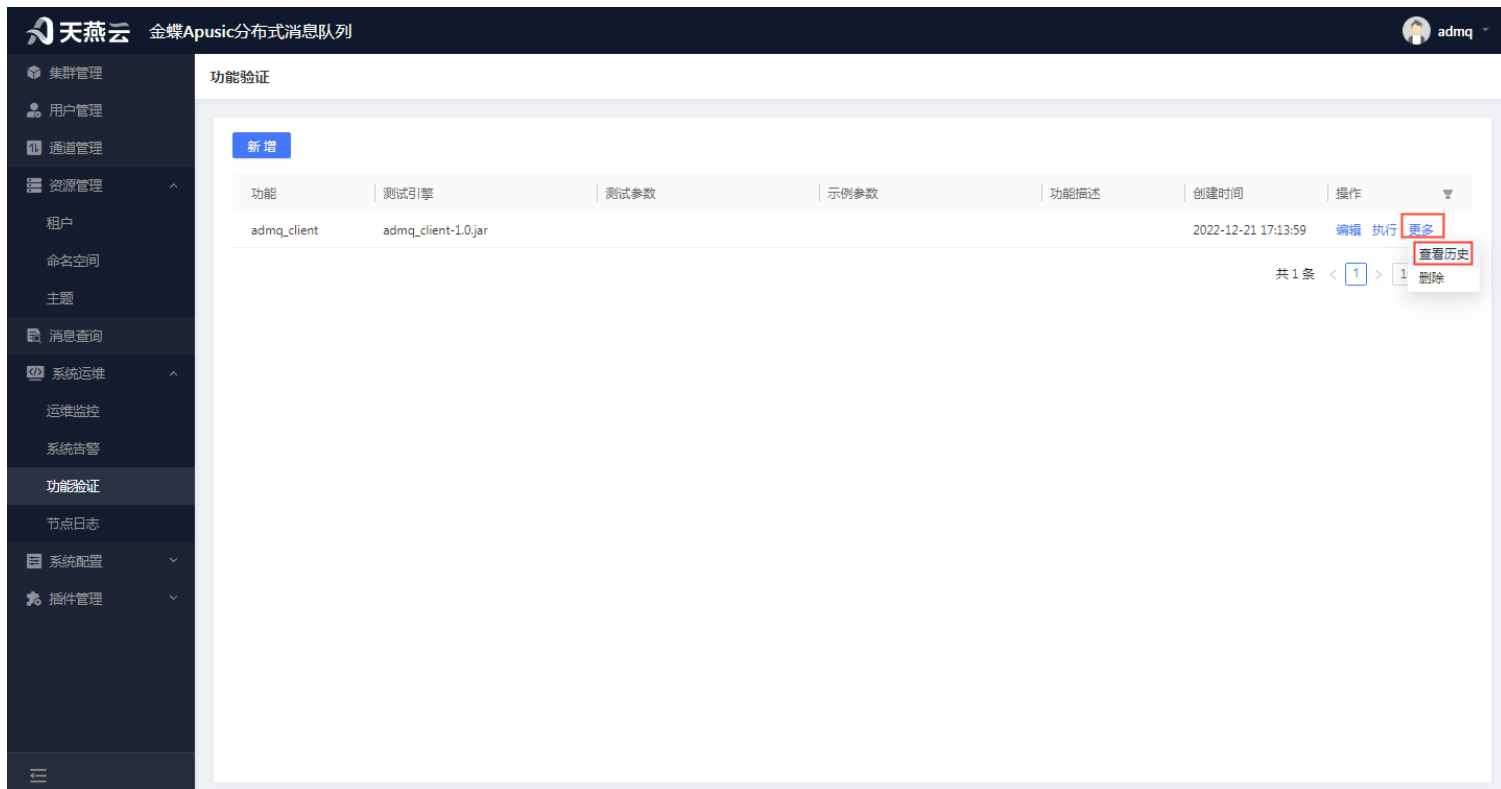
进入【系统运维】->【功能验证】，在功能页面选择需要删除的功能验证信息，点击【更多】按钮->【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除节点。

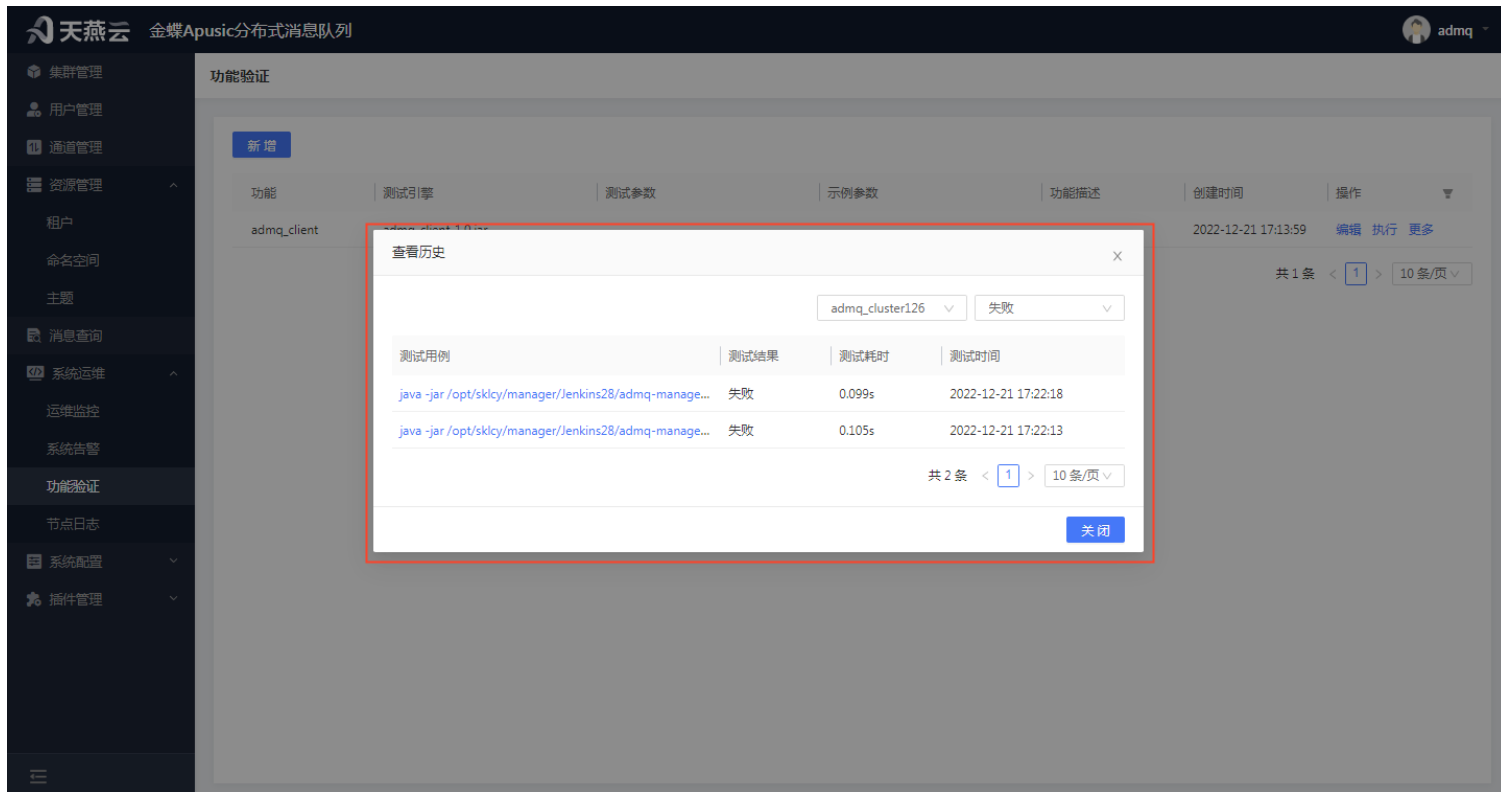




5.9.3.5 查看历史执行功能验证

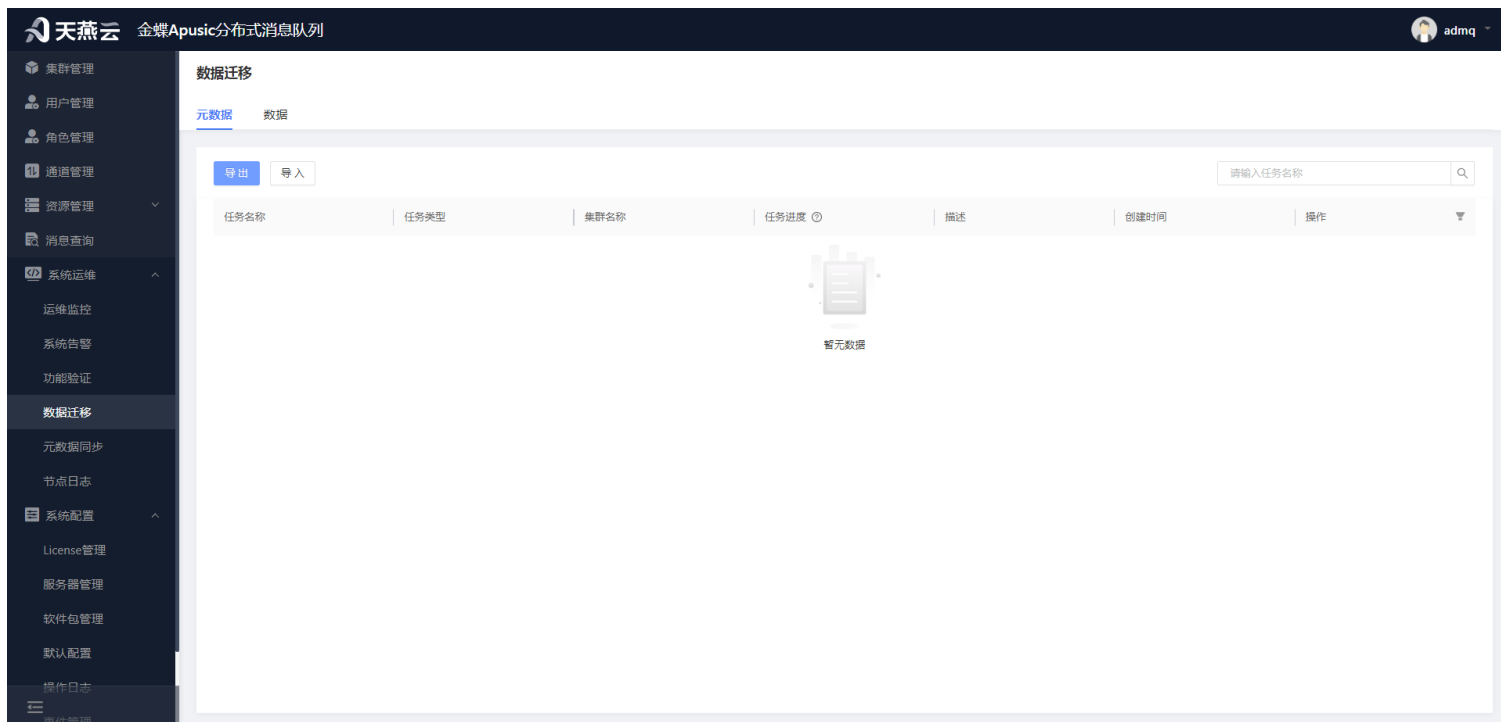
进入【系统运维】->【功能验证】，在功能页面选择需要查看历史执行的功能验证信息，点击【更多】按钮->【查看历史】按钮，弹出查看历史弹框，在弹框中可查看到历史功能测试用例、测试结果、测试耗时、测试时间等信息。





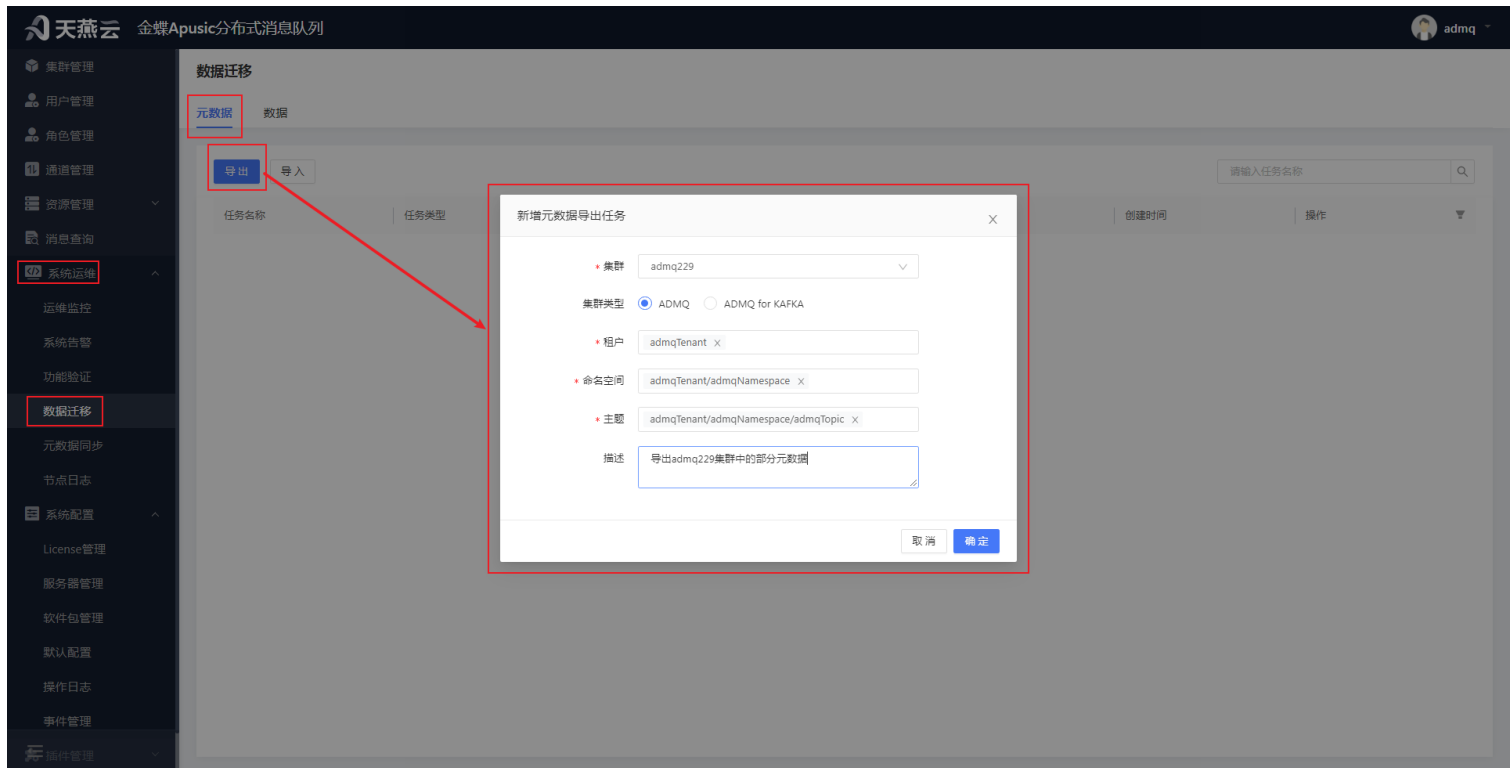
5.9.4 数据迁移

ADMQ的数据迁移可以将现有的ADMQ集群中的数据迁移到另一个ADMQ集群中，包括元数据迁移、数据迁移，这个功能对于需要扩容、备份或迁移ADMQ集群的场景非常有用。目前主要支持原生ADMQ和Kafka插件，RocketMQ插件和RabbitMQ插件暂不支持。



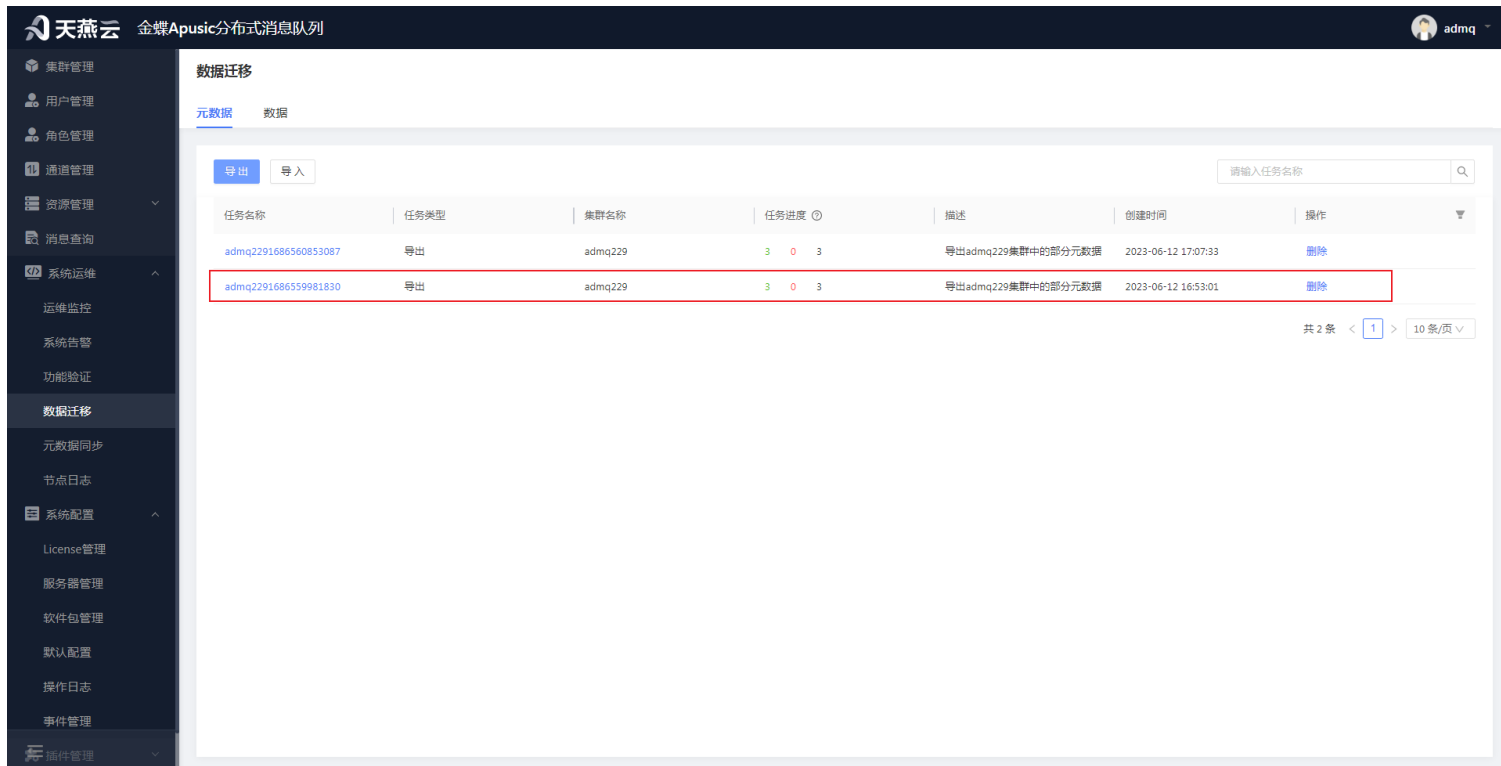
5.9.4.1 元数据迁移_导出

进入【系统运维】->【数据迁移】，选择元数据页签，在元数据页面点击左上角【导出】按钮，出现“新增元数据导出任务”弹窗页面，输入相关信息。点击“确定”按钮，可将相关集群中的元数据导出到具体文件中。

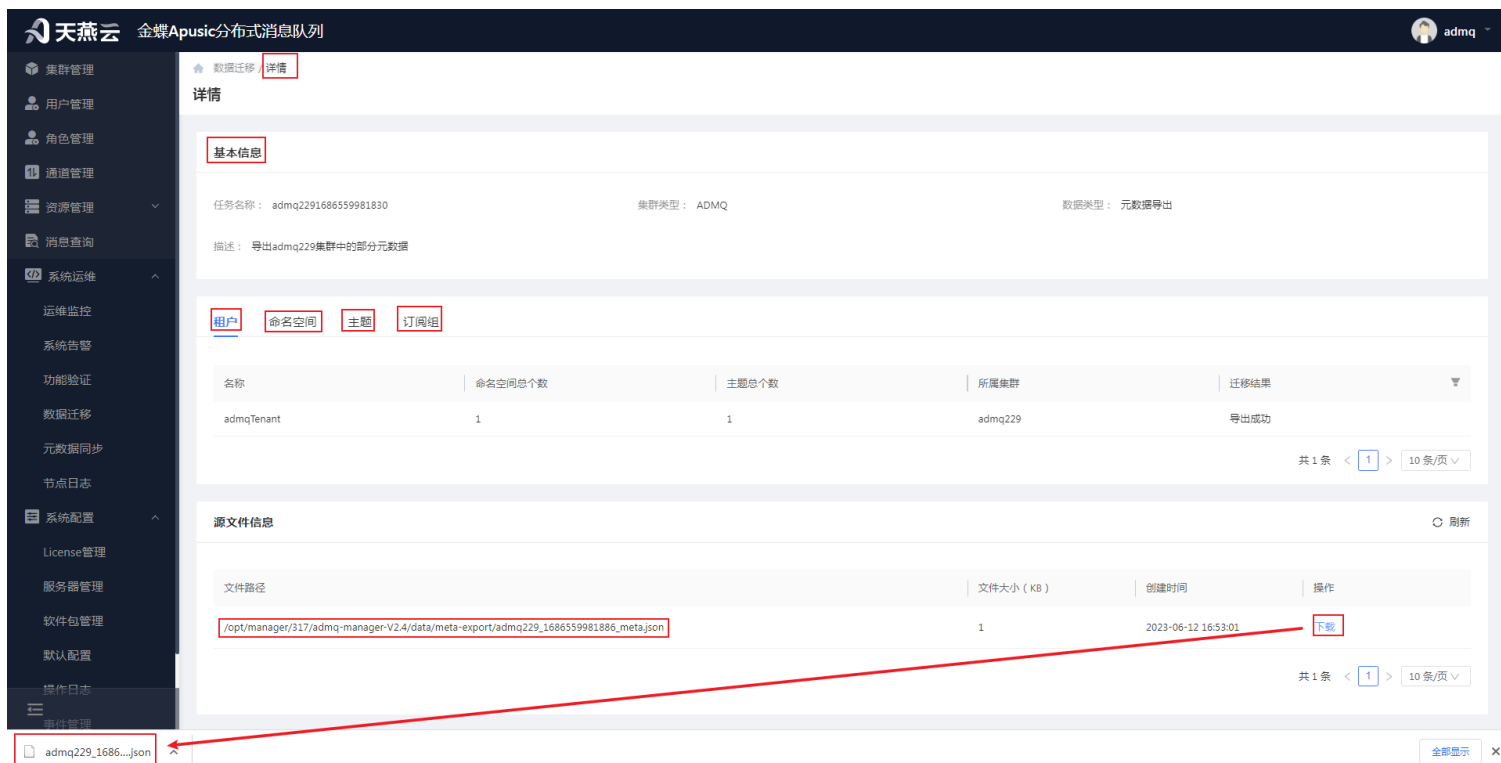


- 集群：下拉框，选择将哪个集群的元数据导出到文件中进行本地存储。
- 集群类型：单选框，可选ADMQ和ADMQ for KAFKA（Kafka插件）。
- 租户：选择器，全选、选中某个或多个租户名称
- 命名空间：选择器，全选、选中某个或多个命名空间名称
- 主题：选择器，全选、选中某个或多个主题名称
- 描述：元数据导出任务的描述信息

导出成功后，会在元数据页面的展示列表展示出来：

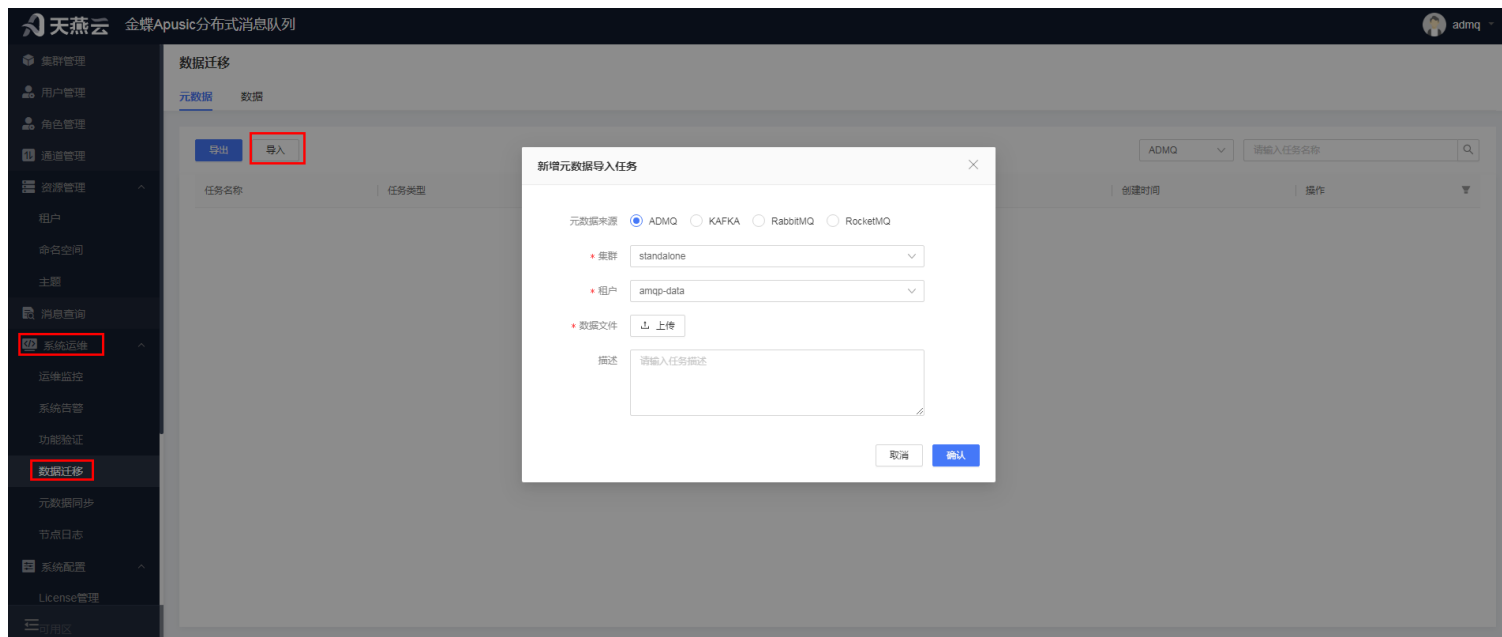


点击任务名称，可进入到任务详情页面，在详情页面的源文件信息模块，选择某个文件进行下载：



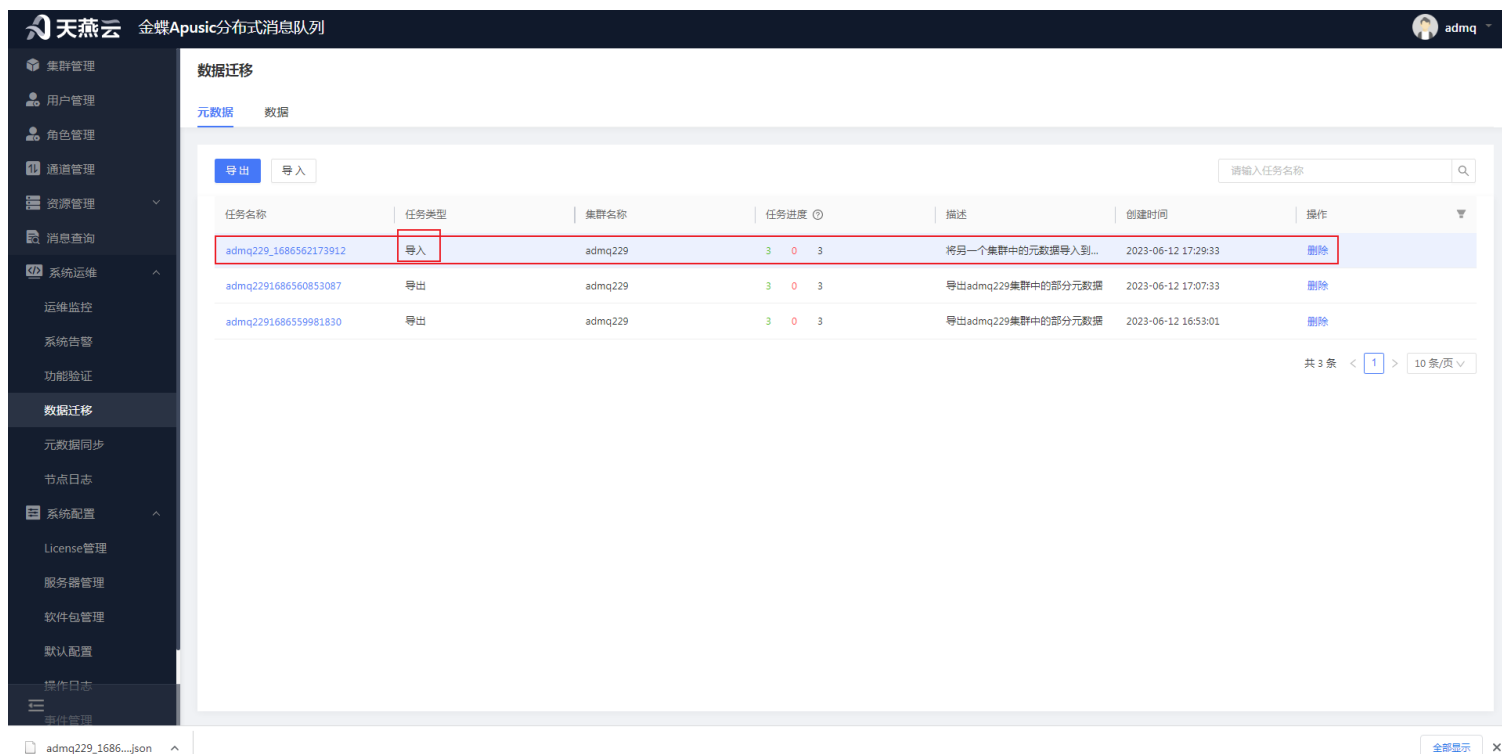
5.9.4.2 元数据迁移_导入

进入【系统运维】->【数据迁移】，选择元数据页签，在元数据页面点击左上角【导入】按钮，出现“新增元数据导入任务”弹窗页面，输入相关信息并上传相关json文件，点击“确认”按钮，可将相关集群中的元数据导入到另一个集群中。

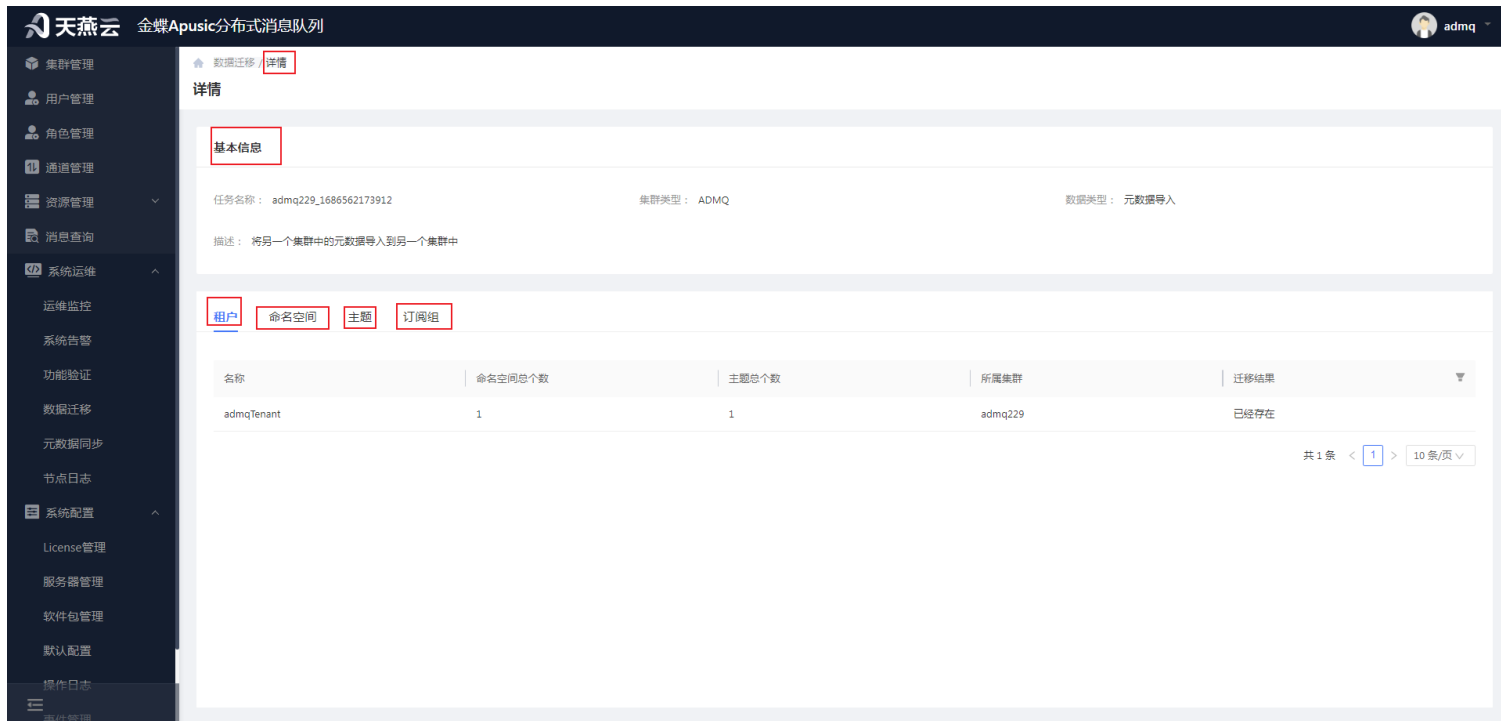


- 云数据来源：单选框，选择元数据来源
- 集群：下拉框，选择将本地文件中的元数据导入到哪个集群中。
- 租户：下拉框，选择将本地文件中的元数据导入到哪个租户中。
- 数据文件：点击上传，选择已下载的json文件进行上传。
- 描述：元数据导入任务的描述信息

导入成功后，会在元数据页面的展示列表展示出来：

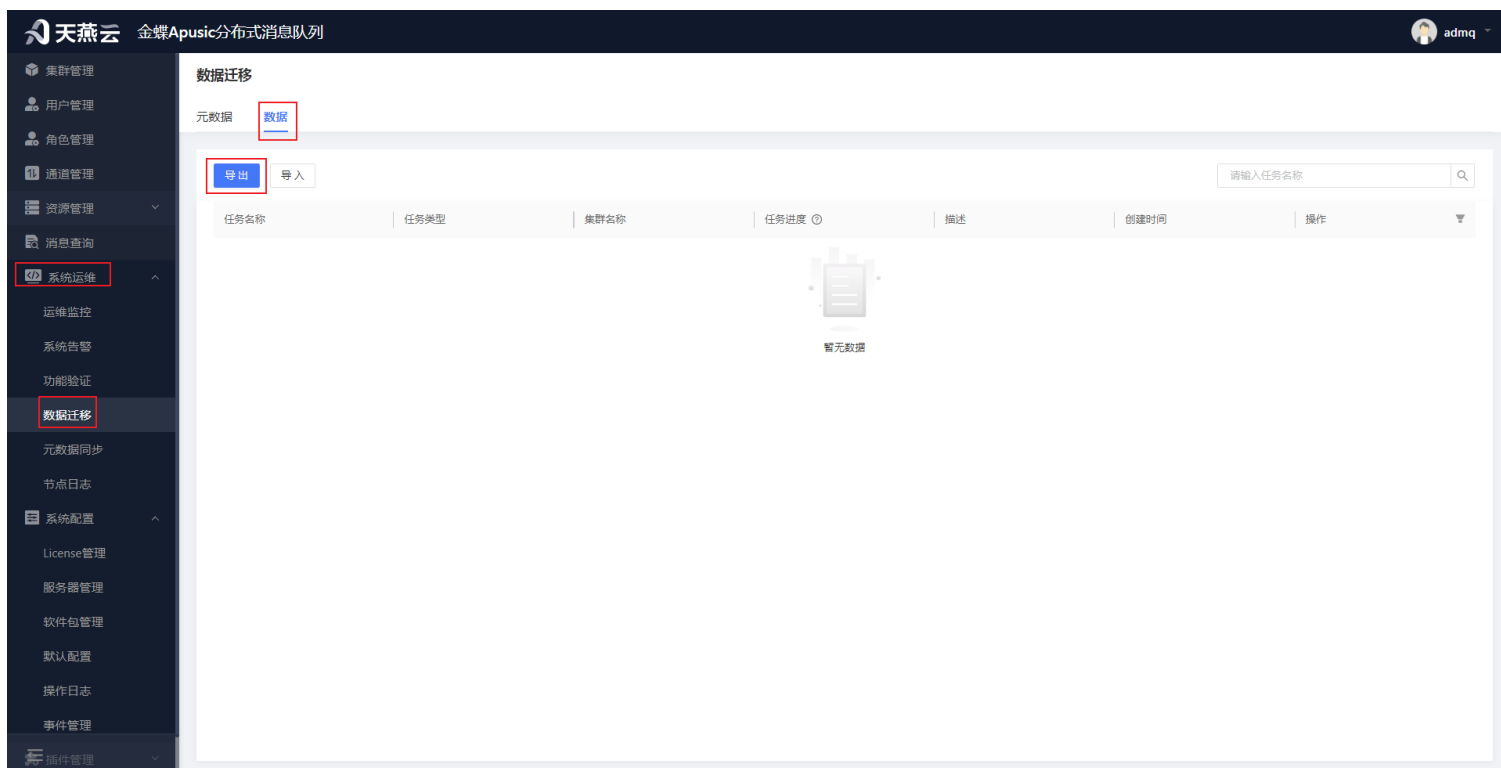


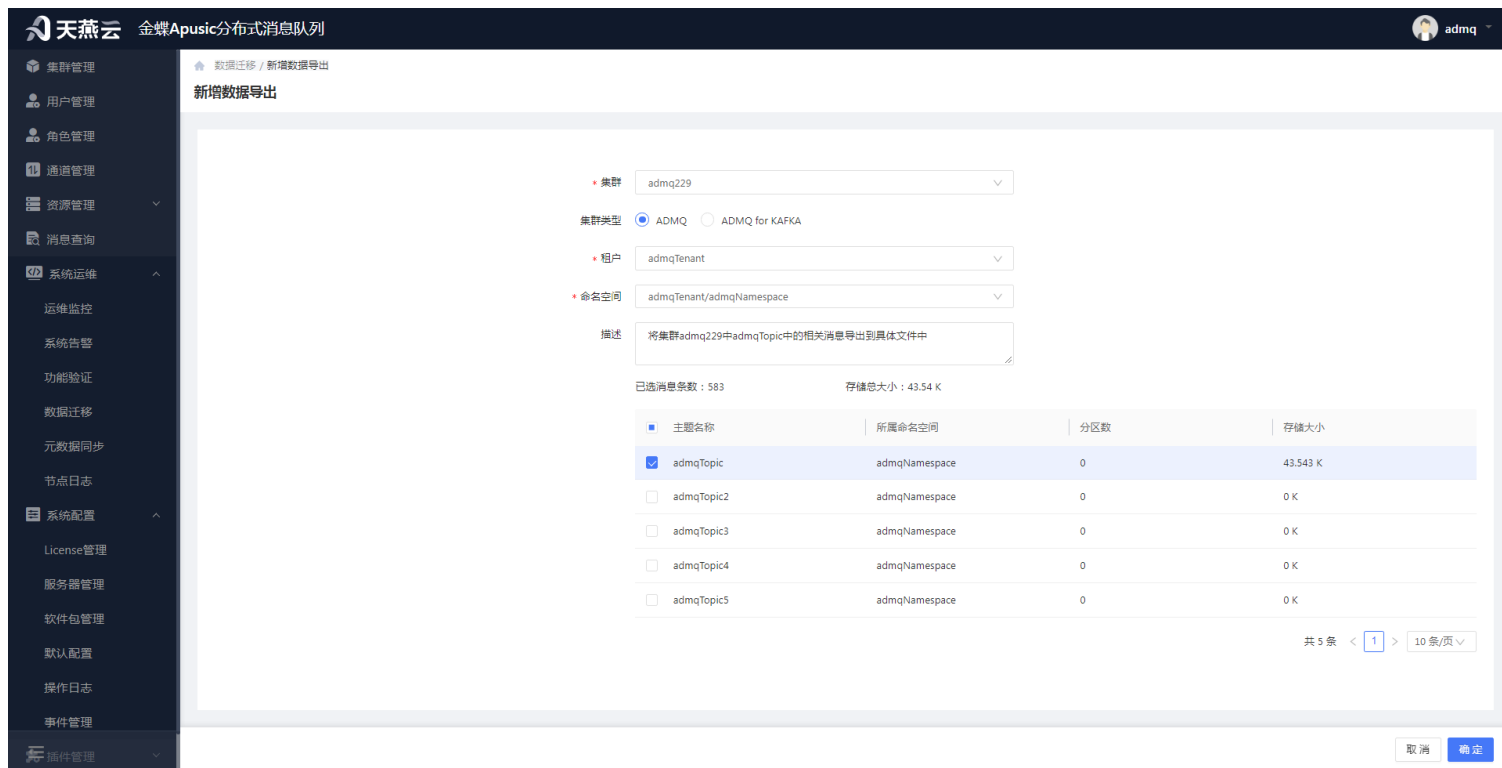
点击任务名称，可进入到任务详情页面：



5.9.4.3 数据迁移_导出

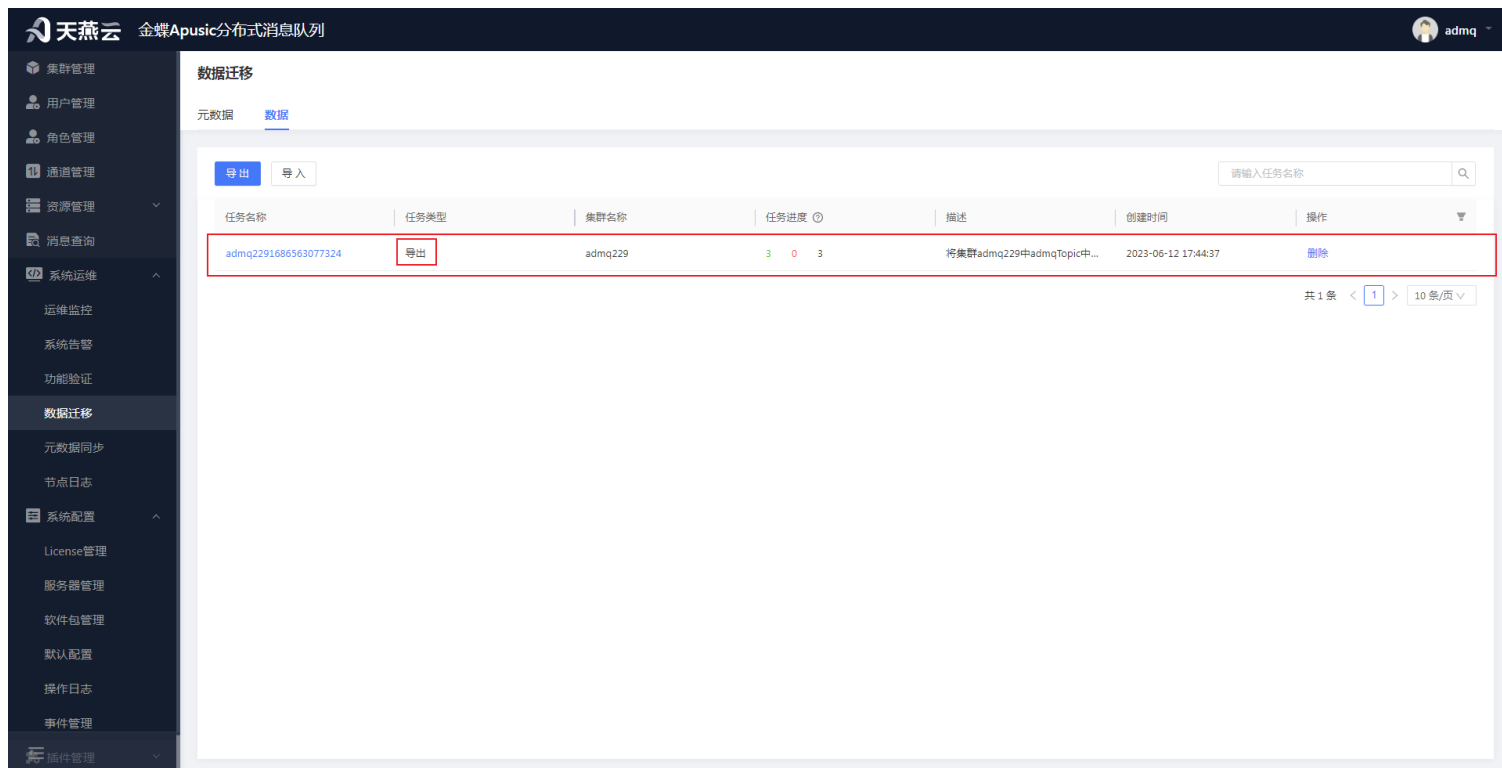
进入【系统运维】->【数据迁移】，选择数据页签，在数据页面点击左上角【导出】按钮，会进入到“新增数据导出”页面，输入相关信息。点击“确定”按钮，可将相关集群中的数据导出到具体文件中。



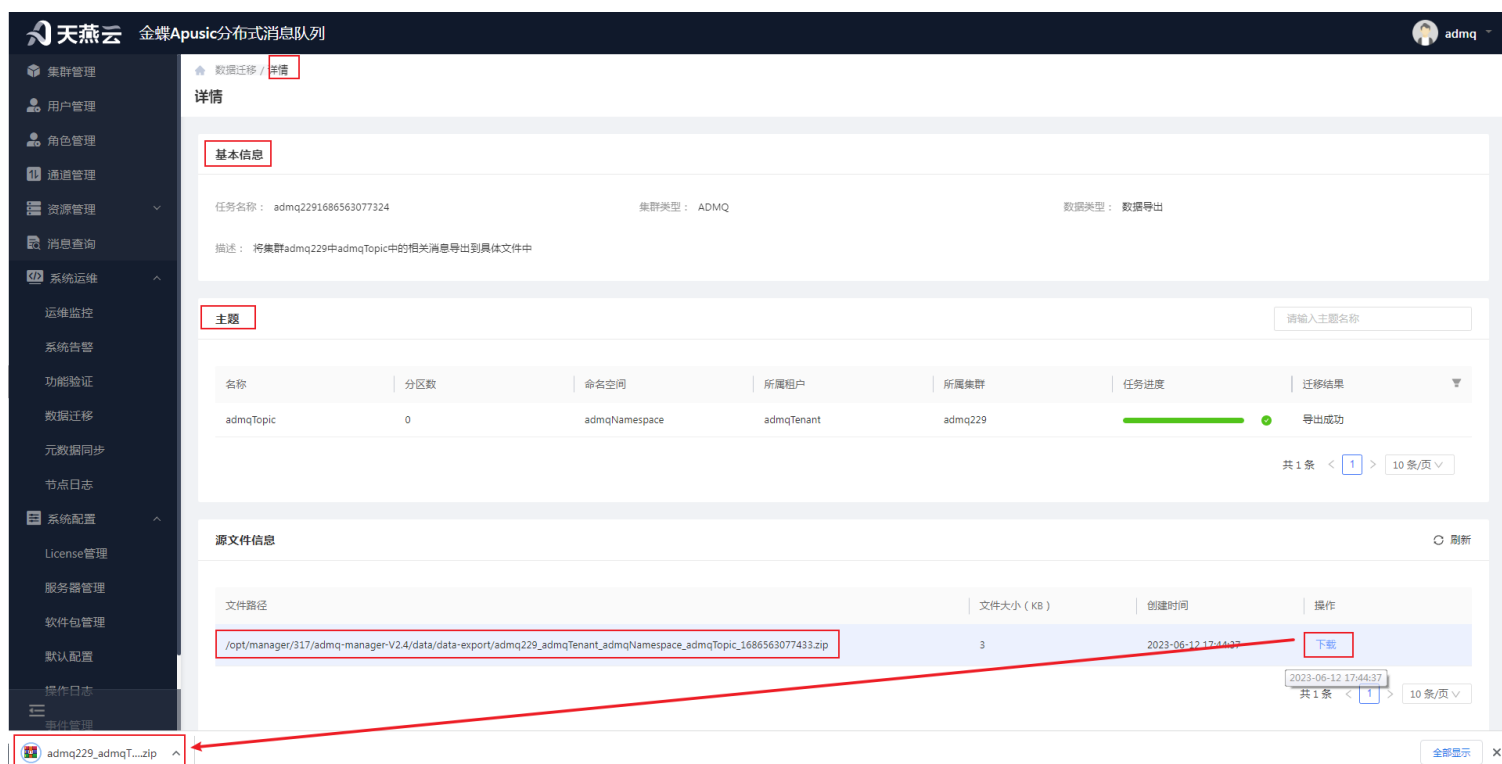


- 集群：下拉框，选择将哪个集群的数据导出到文件中进行本地存储。
- 集群类型：单选框，可选ADMQ和ADMQ for KAFKA（Kafka插件）。
- 租户：下拉框，需要导出的数据所在的租户名称，一次只可选择一个
- 命名空间：下拉框，需要导出的数据所在的命名空间名称，一次只可选择一个
- 描述：数据导出任务的描述信息
- 主题：多选框，一次可选中多个主题下的消息

导出成功后，会在元数据页面的展示列表展示出来：

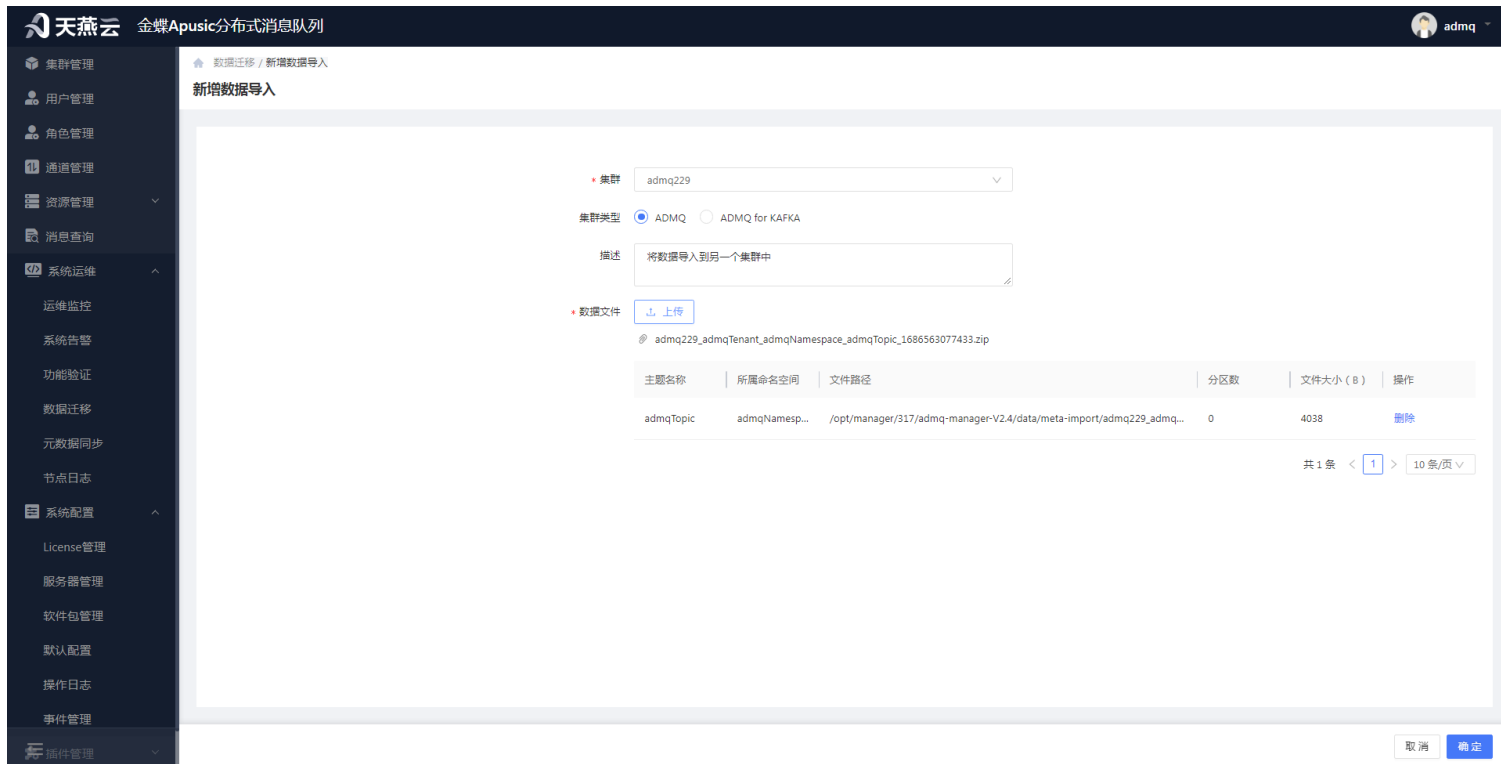
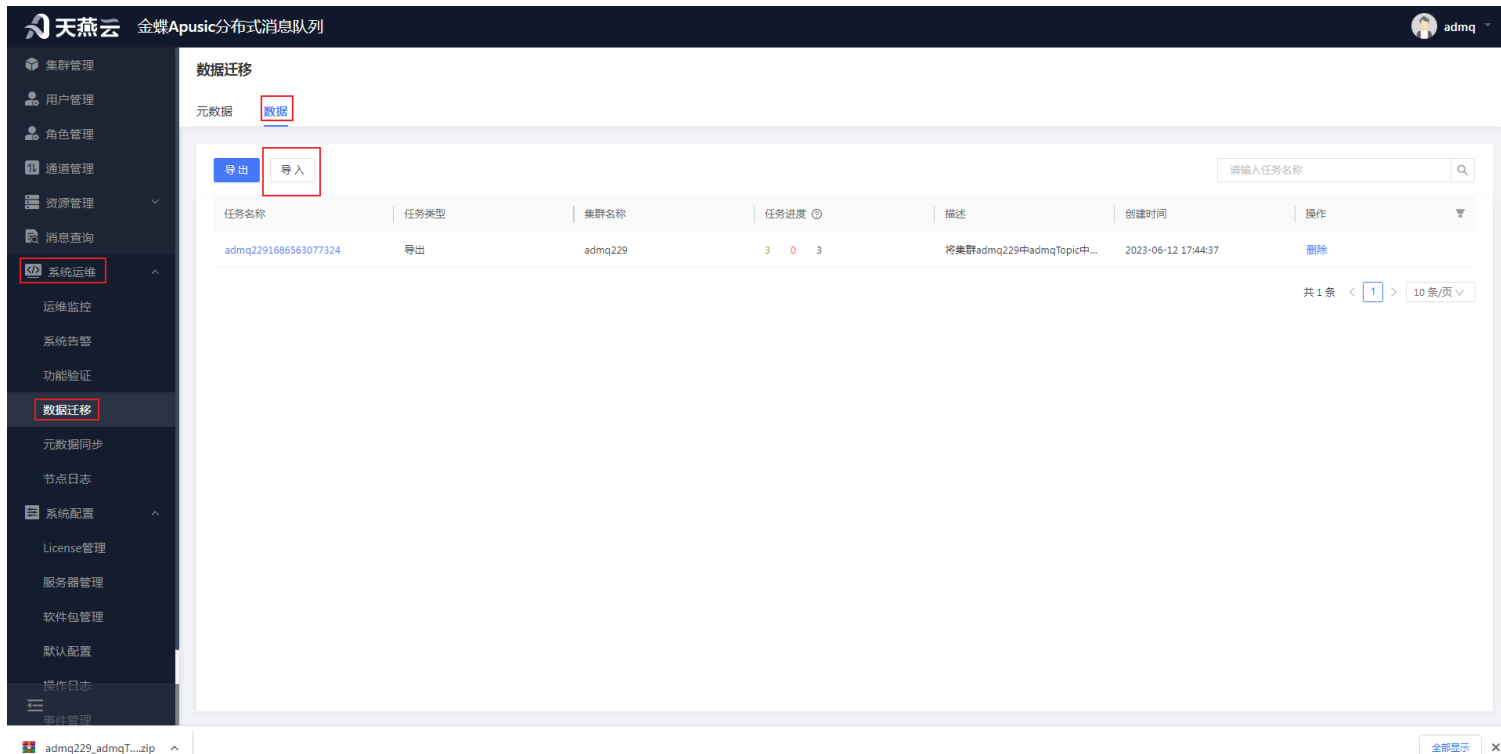


点击任务名称，可进入到任务详情页面，在详情页面的源文件信息模块，选择某个文件进行下载：



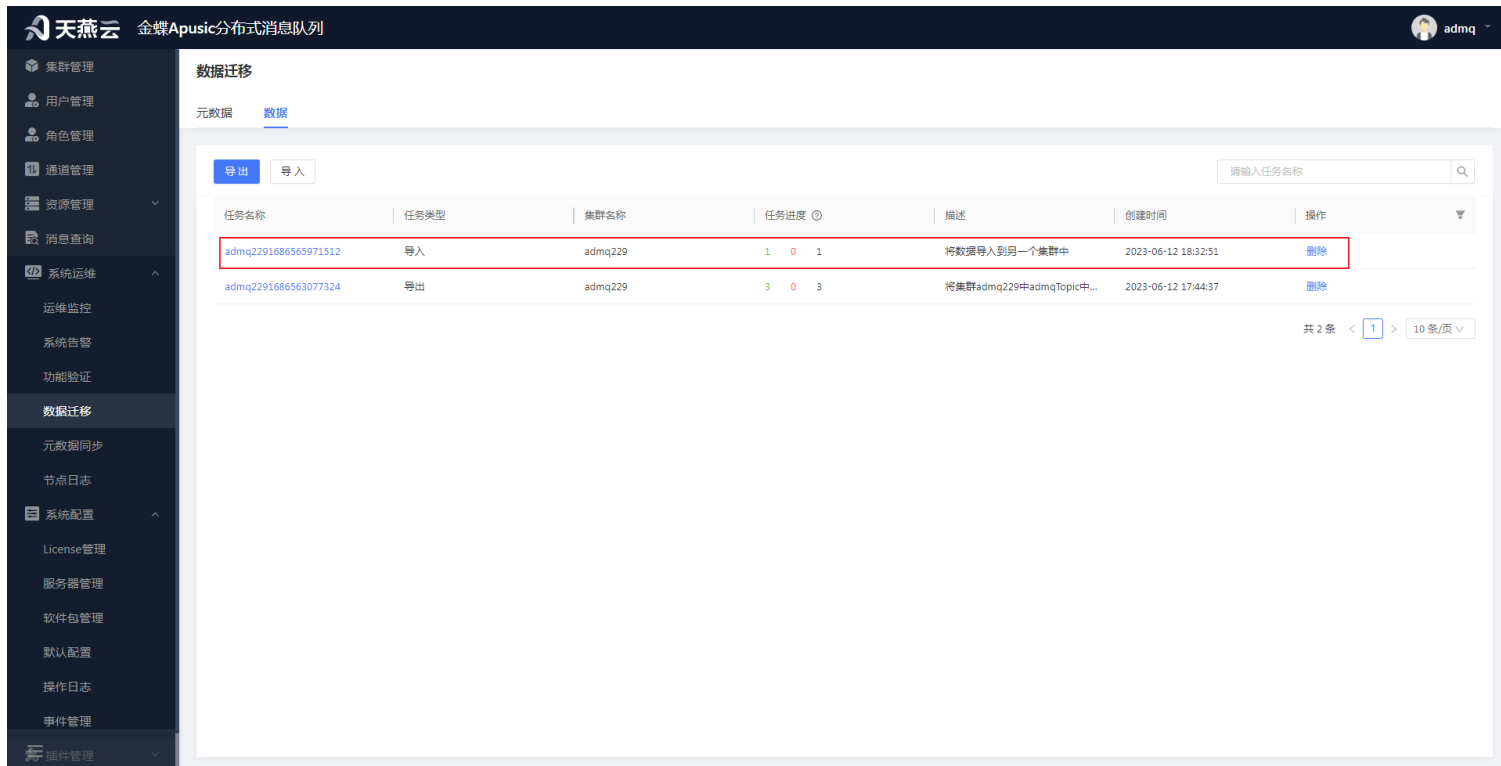
5.9.4.4 数据迁移_导入

进入【系统运维】->【数据迁移】，选择数据页签，在数据页面点击左上角【导入】按钮，进入到“新增数据导入”页面，输入相关信息并上传相关数据文件，点击“确认”按钮，可将相关集群中的数据导入到另一个集群中。

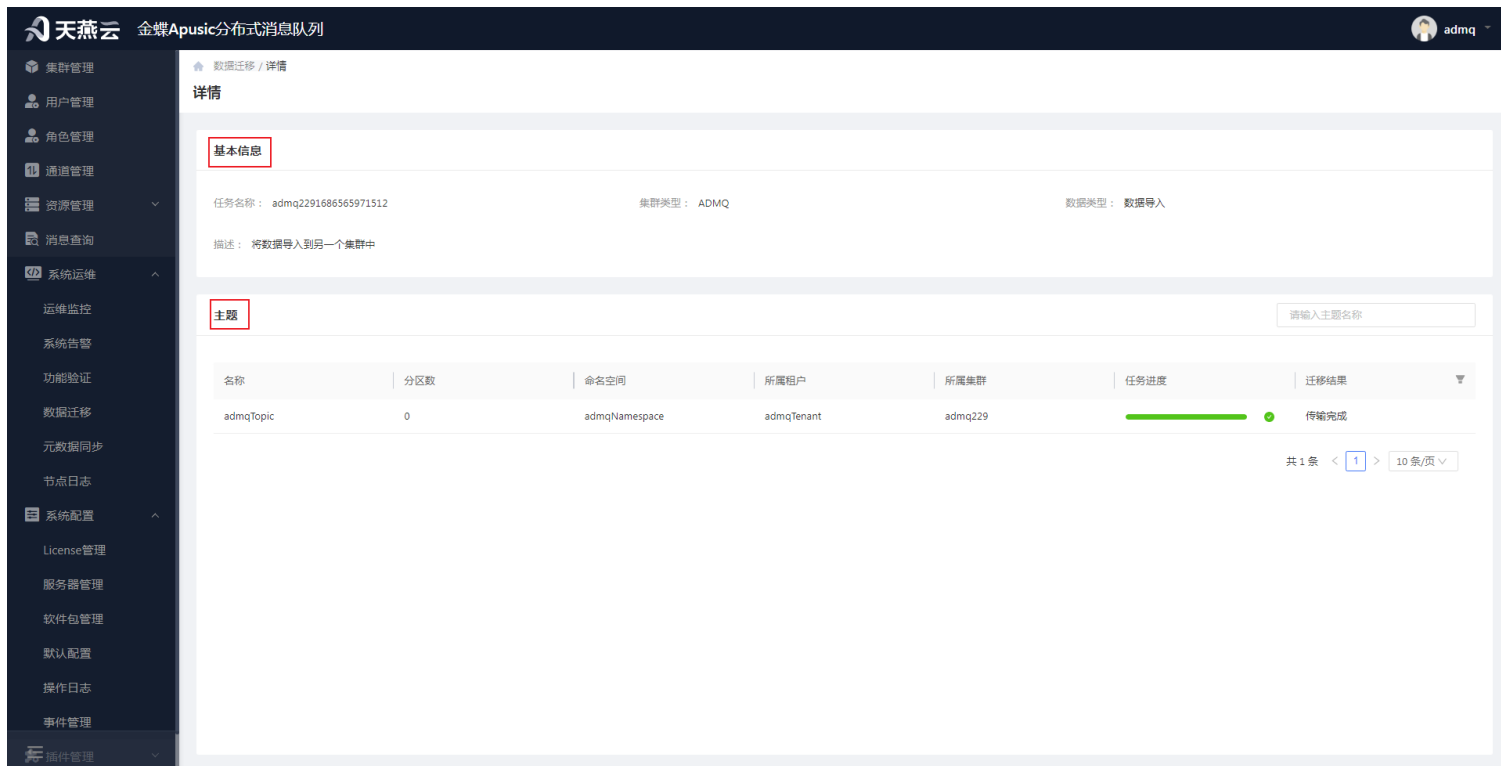


- 集群：下拉框，选择将数据文件导入到哪个集群中。
- 集群类型：单选框，ADMQ、ADMQ for KAFKA (Kafka插件)。
- 描述：数据导入任务的描述信息
- 数据文件：点击上传，选择已下载的压缩文件进行上传。

导入成功后，会在元数据页面的展示列表展示出来：



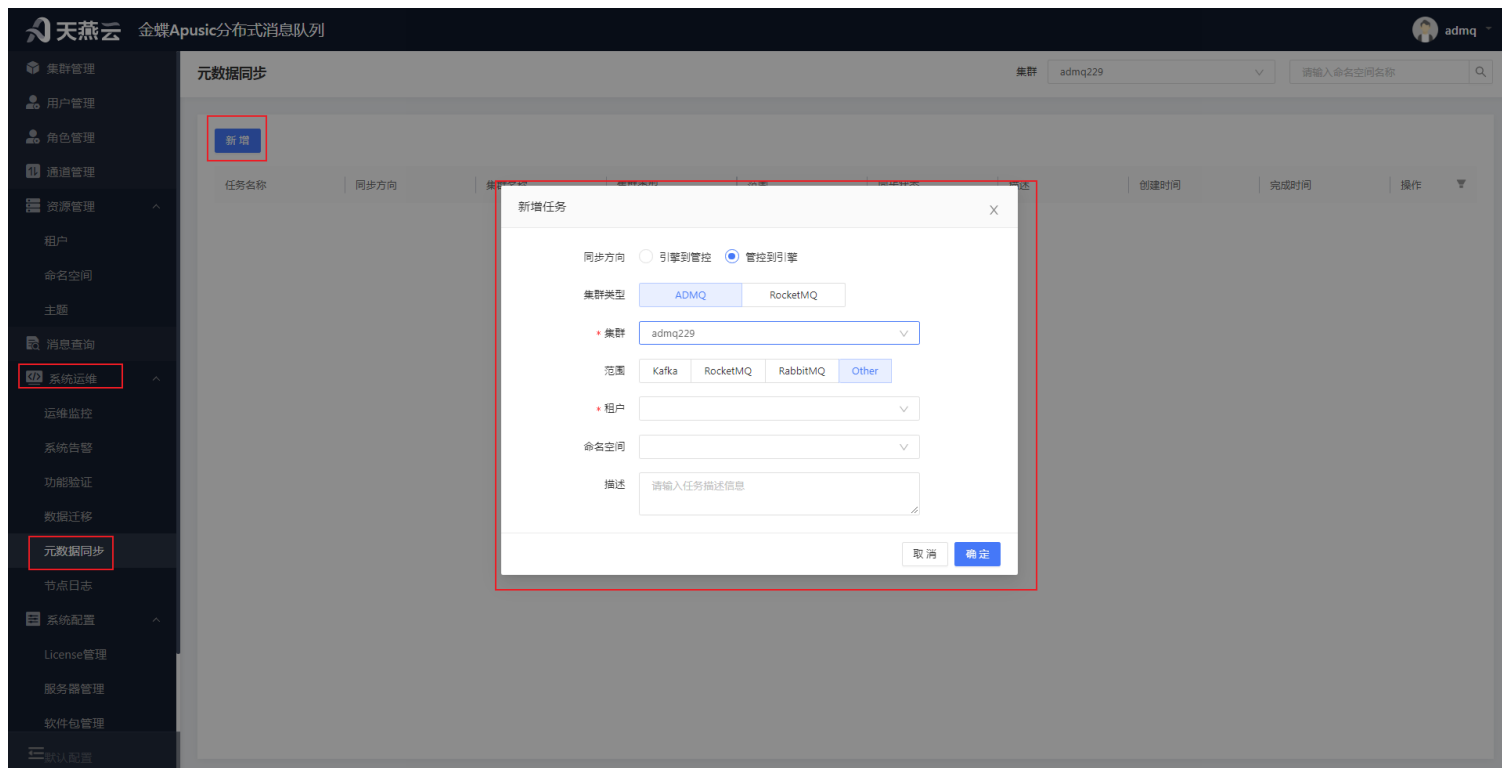
点击任务名称，可进入到任务详情页面：



5.9.5 元数据同步

将命令行或者客户端创建的元数据同步到管控台或者将管控台创建的元数据同步到引擎上。

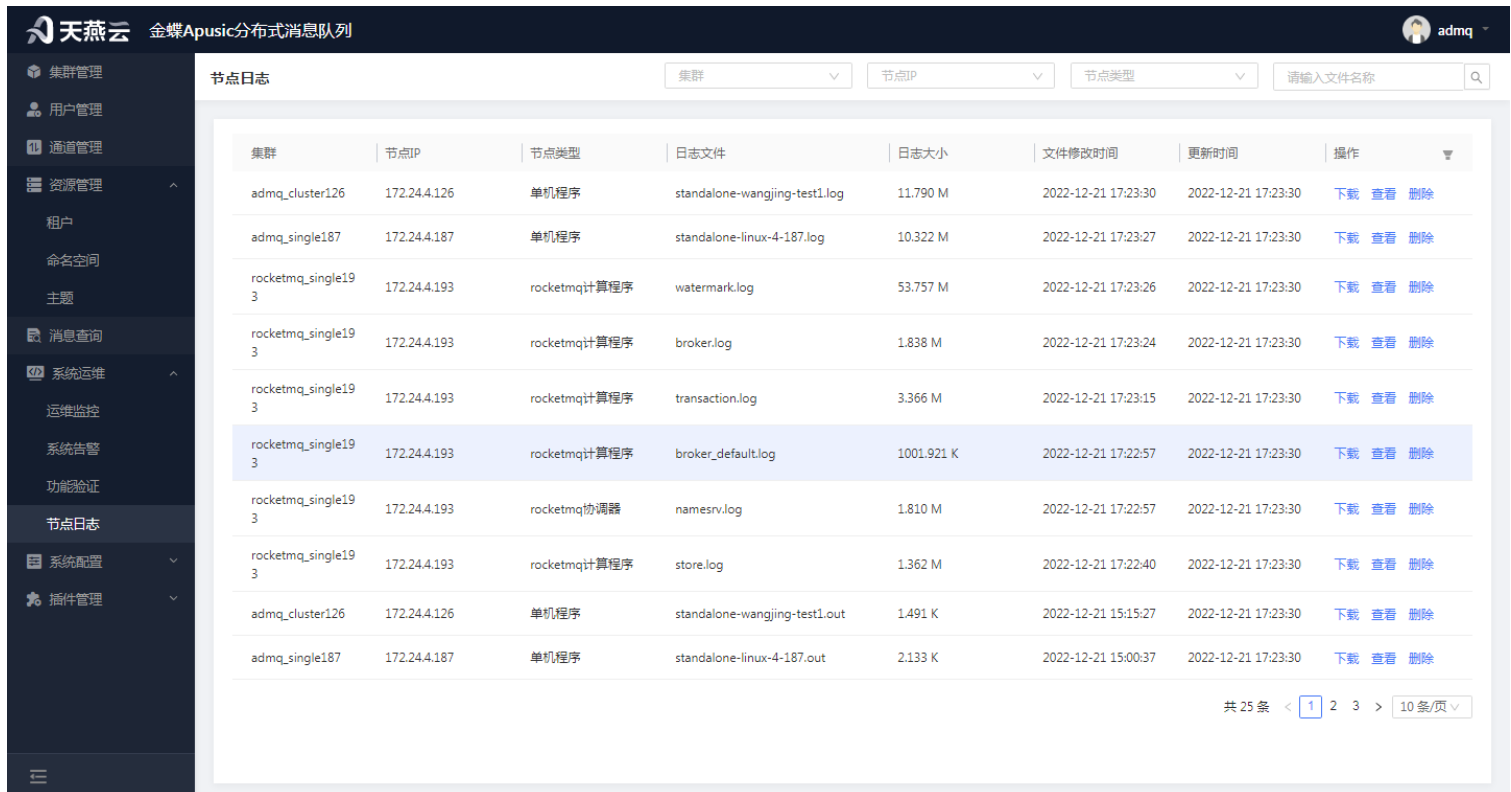
进入【系统运维】->【元数据同步】，选择数据页签，在元数据同步页面点击左上角【新增】按钮，出现新增任务弹窗，输入相关信息，点击“确定”按钮，即可创建一个同步任务。



- 同步方向：单选框，可选：引擎到管控、管控到引擎
- 集群类型：单选，包括：ADMQ、RocketMQ
- 集群：下拉框，需要同步的元数据所在的集群名称
- 范围：单选，可选：Kafka、RocketMQ、RabbitMQ、Other
- 租户：数据导出任务的描述信息
- 命名空间：多选框，一次可选中多个主题下的消息
- 描述：同步任务的描述信息

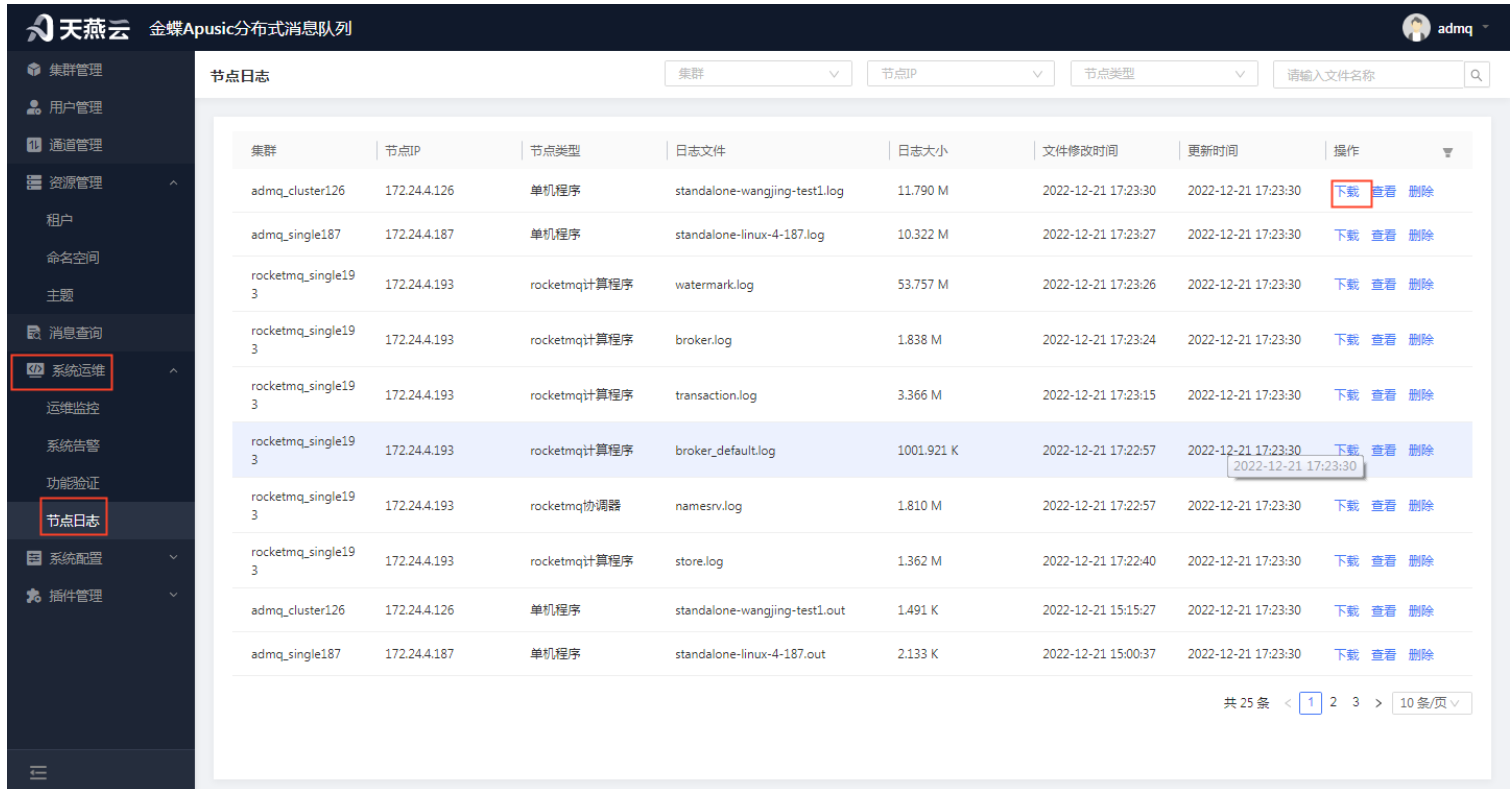
5.9.6 节点日志

管理控制台支持对集群的计算节点、存储节点、协调器的日志进行管理，可下载日志、查看实时日志、删除日志、查询日志信息等。



5.9.6.1 下载节点日志

进入【系统运维】>【节点日志】，在节点日志页面，选择需要下载日志的节点，点击【下载】按钮，可下载节点日志信息。



5.9.6.2 查看节点日志

进入【系统运维】>【节点日志】，在节点日志页面，选择需要查看实时日志的节点，点击【查看】按钮，可查看节点实时日志信息。注意：当日志信息是压缩文件的形式，不可直接查看，可下载日志到本地解压后进行查看。

节点日志

集群	节点IP	节点类型	日志文件	日志大小	文件修改时间	更新时间	操作
adm_cluster126	172.24.4.126	单机程序	standalone-wangjing-test1.log	11.794 M	2022-12-21 17:25:30	2022-12-21 17:25:30	下载 查看 删除
admq_single187	172.24.4.187	单机程序	standalone-linux-4-187.log	10.325 M	2022-12-21 17:25:27	2022-12-21 17:25:30	下载 查看 删除
rocketmq_single193	172.24.4.193	rocketmq计算程序	watermark.log	53.790 M	2022-12-21 17:25:26	2022-12-21 17:25:30	下载 查看 删除
rocketmq_single193	172.24.4.193	rocketmq计算程序	broker.log	1.839 M	2022-12-21 17:25:24	2022-12-21 17:25:30	下载 查看 删除
rocketmq_single193	172.24.4.193	rocketmq计算程序	transaction.log	3.368 M	2022-12-21 17:25:15	2022-12-21 17:25:30	下载 查看 删除
rocketmq_single193	172.24.4.193	rocketmq计算程序	broker_default.log	1002.535 K	2022-12-21 17:24:57	2022-12-21 17:25:30	下载 查看 删除
rocketmq_single193	172.24.4.193	rocketmq协调器	namesrv.log	1.811 M	2022-12-21 17:24:57	2022-12-21 17:25:30	下载 查看 删除
rocketmq_single193	172.24.4.193	rocketmq计算程序	store.log	1.363 M	2022-12-21 17:24:40	2022-12-21 17:25:30	下载 查看 删除
adm_cluster126	172.24.4.126	单机程序	standalone-wangjing-test1.out	1.491 K	2022-12-21 15:15:27	2022-12-21 17:25:30	下载 查看 删除
admq_single187	172.24.4.187	单机程序	standalone-linux-4-187.out	2.133 K	2022-12-21 15:00:37	2022-12-21 17:25:30	下载 查看 删除

共 25 条 < 1 2 3 > 10 条/页

节点日志 / 详情

详情

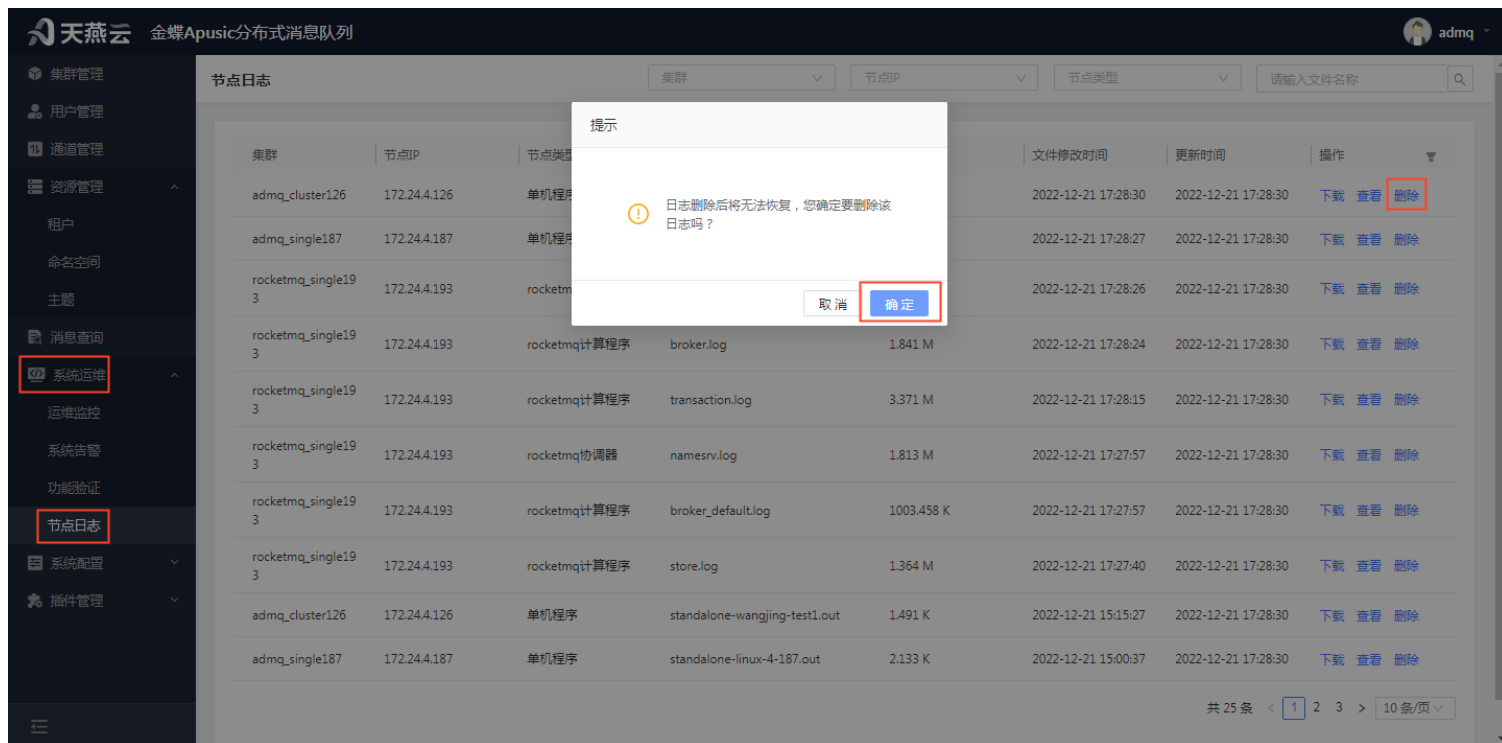
```

17:25:47.887 [metadata-store-27-1] WARN com.apusic.admq.security.ClientAuthService - [/admin/v2/brokers/version HTTP/1.1] 200
17:25:47.887 [metadata-store-27-1] WARN com.apusic.admq.security.ClientAuthService - [/admin/v2/brokers/version HTTP/1.1] 200
17:25:56.255 [pool-38-thread-1] INFO com.apusic.admq.license.HeartbeatTask - heartbeat ...
17:26:00.003 [pulsar-web-56-7] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:26:00 +0800] "GET /admin/v2/brokers/version HTTP/1.1" 200
17:26:00.024 [pulsar-web-56-5] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:26:00 +0800] "GET /admin/v2/broker-stats/topics HTTP/1.1" 200
17:26:00.029 [pulsar-web-56-11] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:26:00 +0800] "GET /metrics HTTP/1.1" 302 0 "-" "Java/1.8.0_291" 0
17:26:00.034 [pulsar-web-56-2] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:26:00 +0800] "GET /metrics HTTP/1.1" 302 0 "-" "Java/1.8.0_291" 0
17:26:00.052 [prometheus-stats-61-1] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:26:00 +0800] "GET /metrics/ HTTP/1.1" 200 1145354 "-" "Java/1.8.0_291" 20
17:26:00.071 [prometheus-stats-61-1] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:26:00 +0800] "GET /metrics/ HTTP/1.1" 200 1145355 "-" "Java/1.8.0_291" 35
17:26:30.003 [pulsar-web-56-9] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:26:30 +0800] "GET /admin/v2/brokers/version HTTP/1.1" 200
17:26:47.887 [metadata-store-27-1] WARN com.apusic.admq.security.ClientAuthService - [/admin/v2/brokers/version HTTP/1.1] 200
17:26:47.887 [metadata-store-27-1] WARN com.apusic.admq.security.ClientAuthService - [/admin/v2/brokers/version HTTP/1.1] 200
17:26:47.888 [metadata-store-27-1] WARN com.apusic.admq.security.ClientAuthService - [/admin/v2/brokers/version HTTP/1.1] 200
17:27:00.003 [pulsar-web-56-12] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:27:00 +0800] "GET /admin/v2/brokers/version HTTP/1.1" 200
17:27:00.031 [pulsar-web-56-8] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:27:00 +0800] "GET /admin/v2/broker-stats/topics HTTP/1.1" 200
17:27:00.035 [pulsar-web-56-5] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:27:00 +0800] "GET /metrics HTTP/1.1" 302 0 "-" "Java/1.8.0_291" 1
17:27:00.036 [pulsar-web-56-5] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:27:00 +0800] "GET /metrics HTTP/1.1" 302 0 "-" "Java/1.8.0_291" 0
17:27:00.056 [prometheus-stats-61-1] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:27:00 +0800] "GET /metrics/ HTTP/1.1" 200 1145415 "-" "Java/1.8.0_291" 20
17:27:00.077 [prometheus-stats-61-1] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:27:00 +0800] "GET /metrics/ HTTP/1.1" 200 1145416 "-" "Java/1.8.0_291" 39
17:27:30.003 [pulsar-web-56-2] INFO org.eclipse.jetty.server.RequestLog - 172.24.4.126 - - [21/十二月/2022:17:27:30 +0800] "GET /admin/v2/brokers/version HTTP/1.1" 200
17:27:47.888 [metadata-store-27-1] WARN com.apusic.admq.security.ClientAuthService - [/admin/v2/brokers/version HTTP/1.1] 200
17:27:47.888 [metadata-store-27-1] WARN com.apusic.admq.security.ClientAuthService - [/admin/v2/brokers/version HTTP/1.1] 200

```

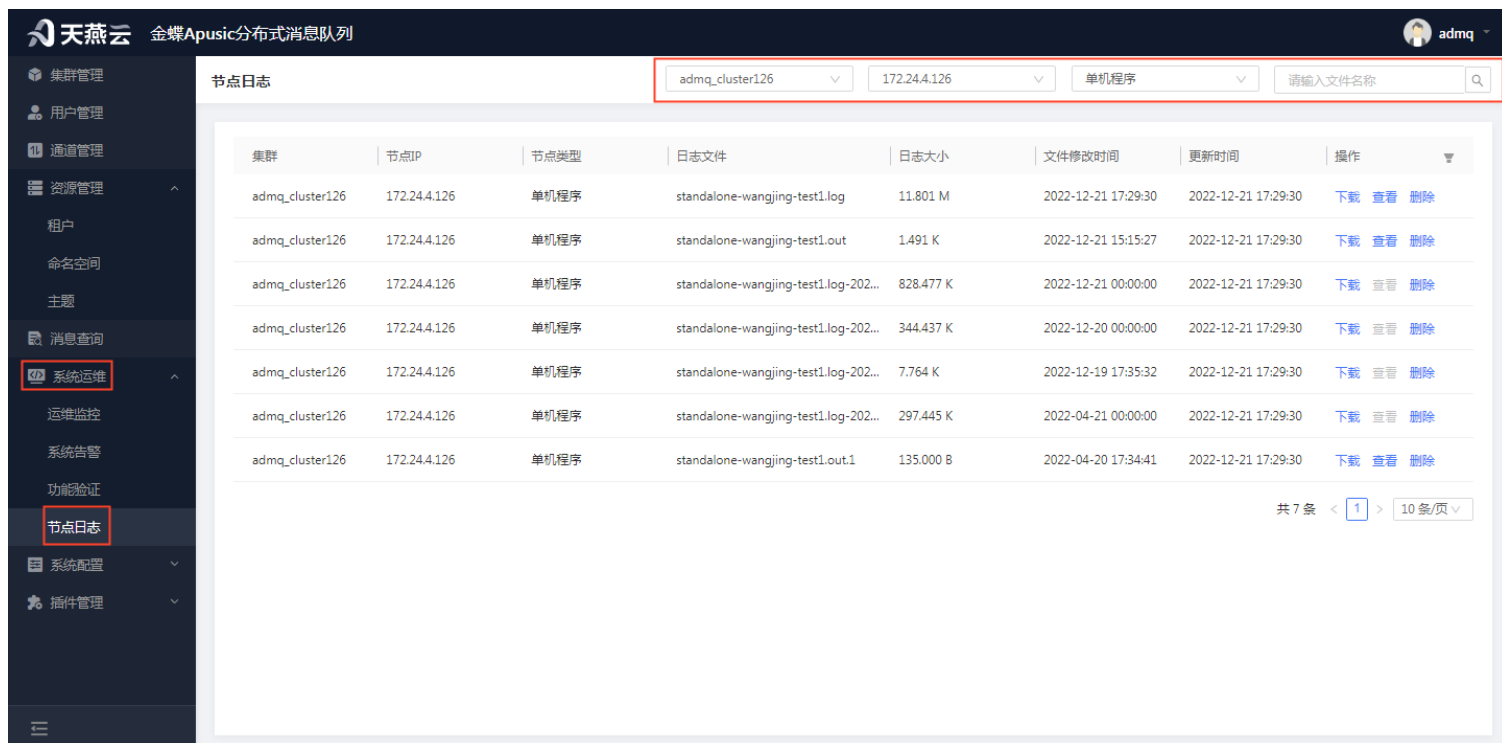
5.9.6.3 删除节点日志

进入【系统运维】>【节点日志】，在节点日志页面，选择需要删除日志的节点，点击【删除】按钮，在弹出的确认删除弹框点击【确定】按钮，可删除日志信息。注意：删除日志会同步删除节点服务器的日志文件，需谨慎删除。



5.9.6.4 查询节点日志

进入【系统运维】>【节点日志】，在节点日志页面，选择集群名称、选择IP、选择节点类型、输入文件名称，可查询节点日志信息。



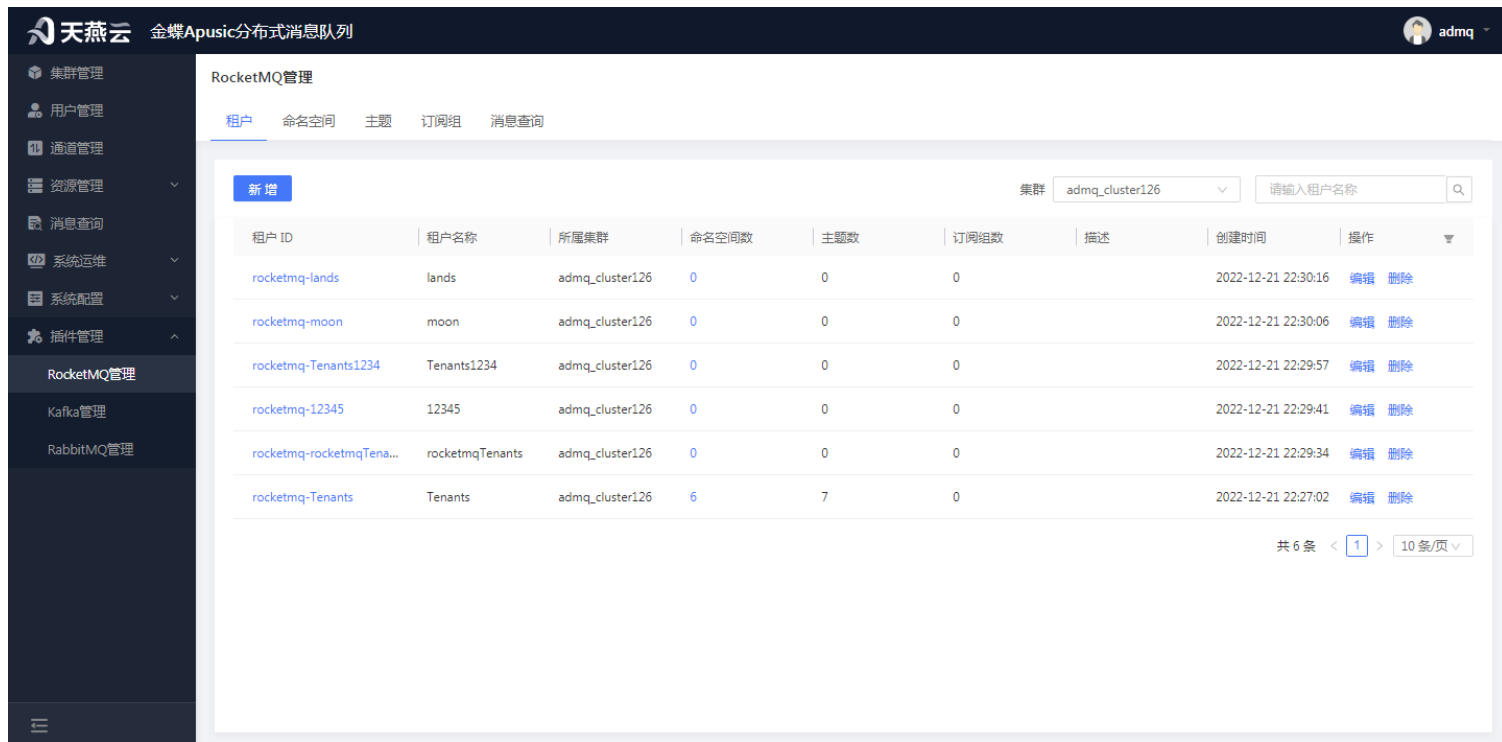
5.10 插件管理

插件管理用于管理rocketmq插件、kafka插件、rabbitmq插件的资源，如消息实例、命名空间、主题、订阅组、消息查询、死信消息、虚拟主机、交换机、队列、路由等。

5.10.1 RocketMQ管理

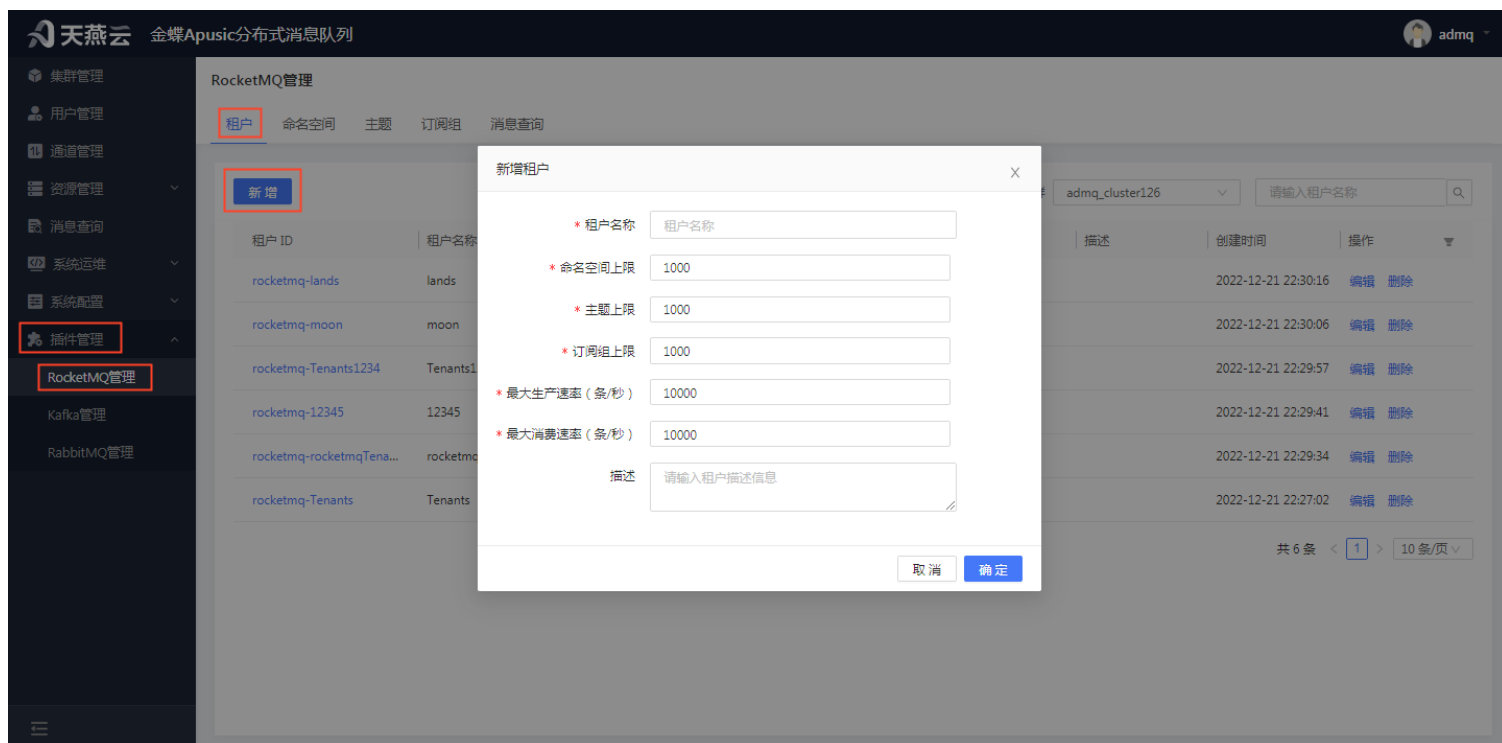
5.10.1.1 租户

租户是管理存储配额、隔离策略的基本单元，使用租户的目的是为了在多用户环境下，使用同一套程序，且保证用户间数据隔离。租户用于新增租户、编辑租户、删除租户、查询租户。



5.10.1.1.1 新增租户

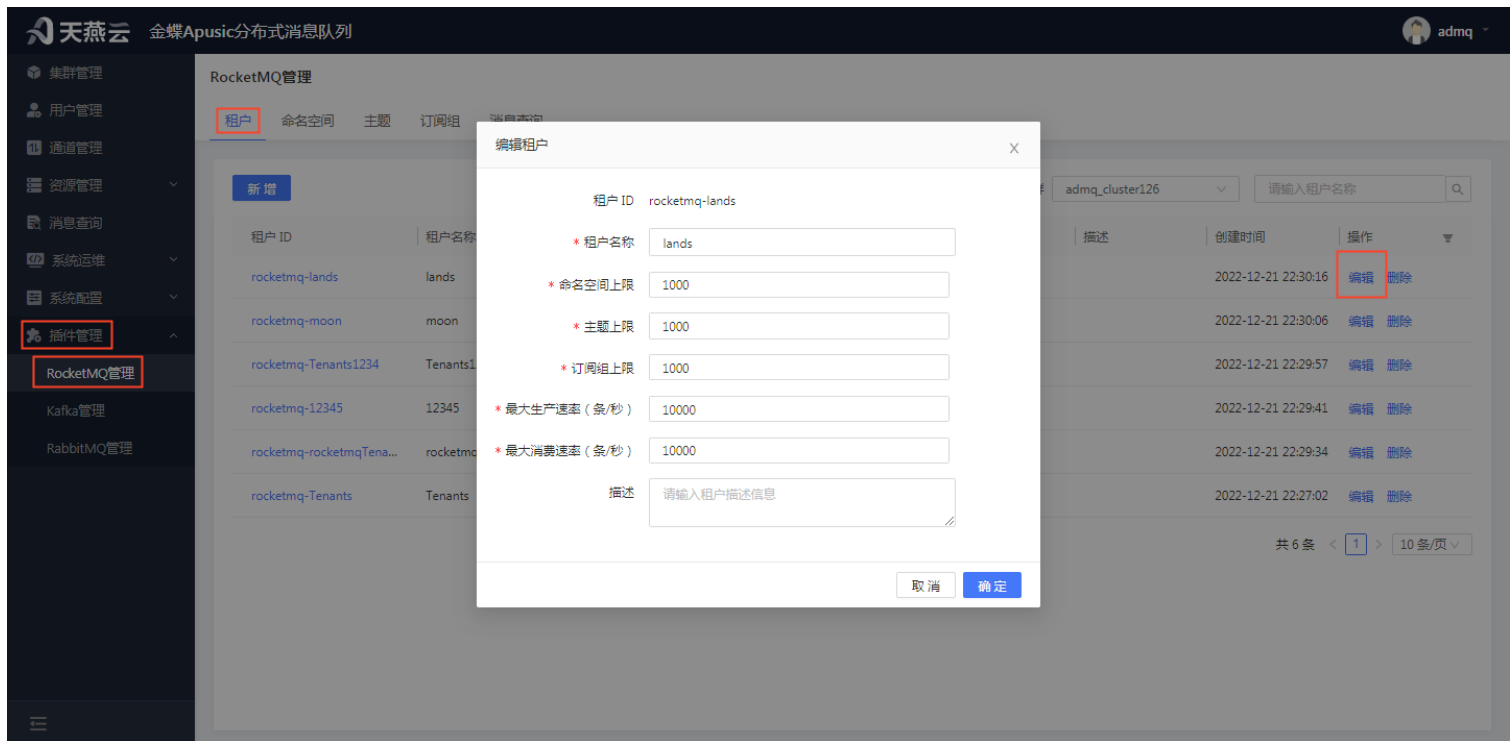
进入【插件管理】->【RocketMQ管理】，选择“租户”页签，点击左上角【新增】按钮，可新增租户。注意：新增租户前，需满足集群服务可用。



- 租户名称：租户的名称，自定义，唯一标识
- 命名空间上限：租户下可新增的命名空间数量，默认为1000，可更改
- 主题上限：租户下可新增的主题数量，默认为1000，可更改
- 订阅组上限：租户下可新增的订阅组数量，默认为1000，可更改
- 最大生产速率（条/秒）：生产者的最大发送速率，默认为10000，可自定义
- 最大消费速率（条/秒）：消费者的最大消费速率，默认为10000，可自定义
- 描述：租户描述信息
- 注意：租户名称只是为了方便记，真正的租户是租户ID。

5.10.1.1.2 编辑租户

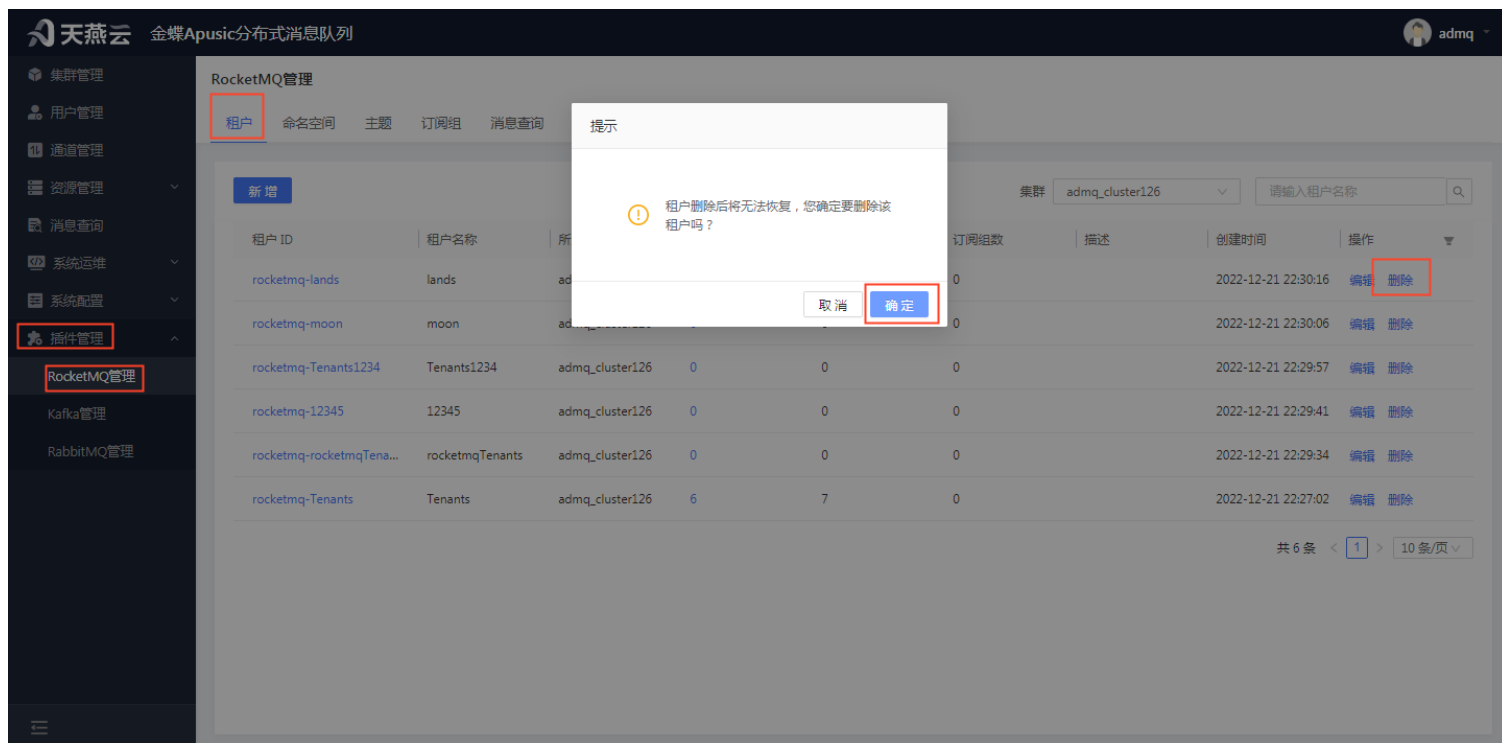
进入【插件管理】->【RocketMQ管理】，选择“租户”页签，选择租户列表中某一个租户名称后的【编辑】按钮，可编辑租户信息。注意：编辑租户前，需满足集群服务可用。



- 租户ID：租户的ID，不可修改，唯一标识
- 租户名称：租户的名称，可修改
- 命名空间上限：租户下可新增的命名空间数量，可修改
- 主题上限：租户下可新增的主题数量，可修改
- 订阅组上限：租户下可新增的订阅组数量，可修改
- 最大生产速率（条/秒）：生产者的最大发送速率，可修改
- 最大消费速率（条/秒）：消费者的最大消费速率，可修改
- 描述：租户描述信息，可修改
- 注意：租户名称只是为了方便记，真正的租户是租户ID。

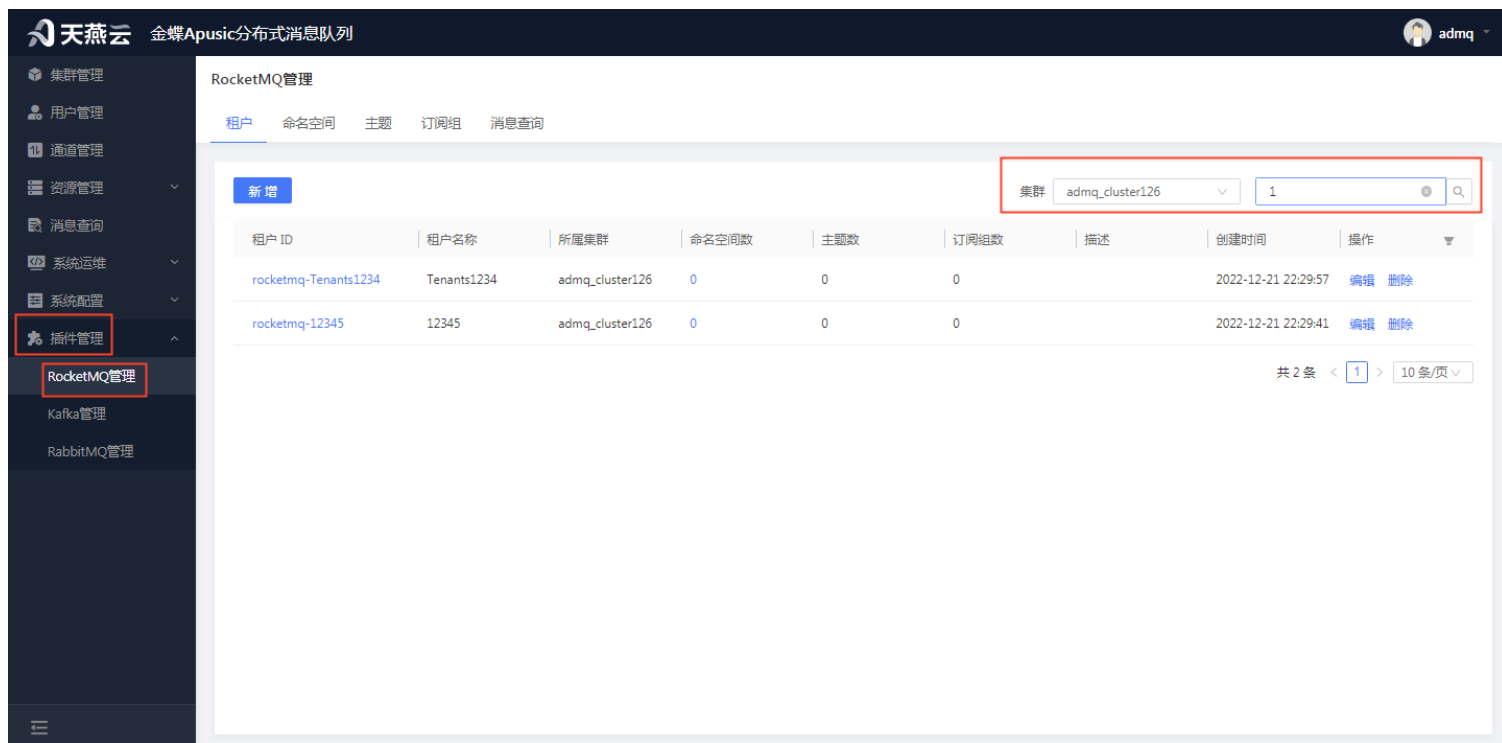
5.10.1.1.3 删除租户

进入【插件管理】->【RocketMQ管理】，选择“租户”页签，选择租户列表中某一个租户名称后的【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除租户信息。注意：删除租户前，需满足集群服务可用，且租户下无命名空间，否则不可删除。



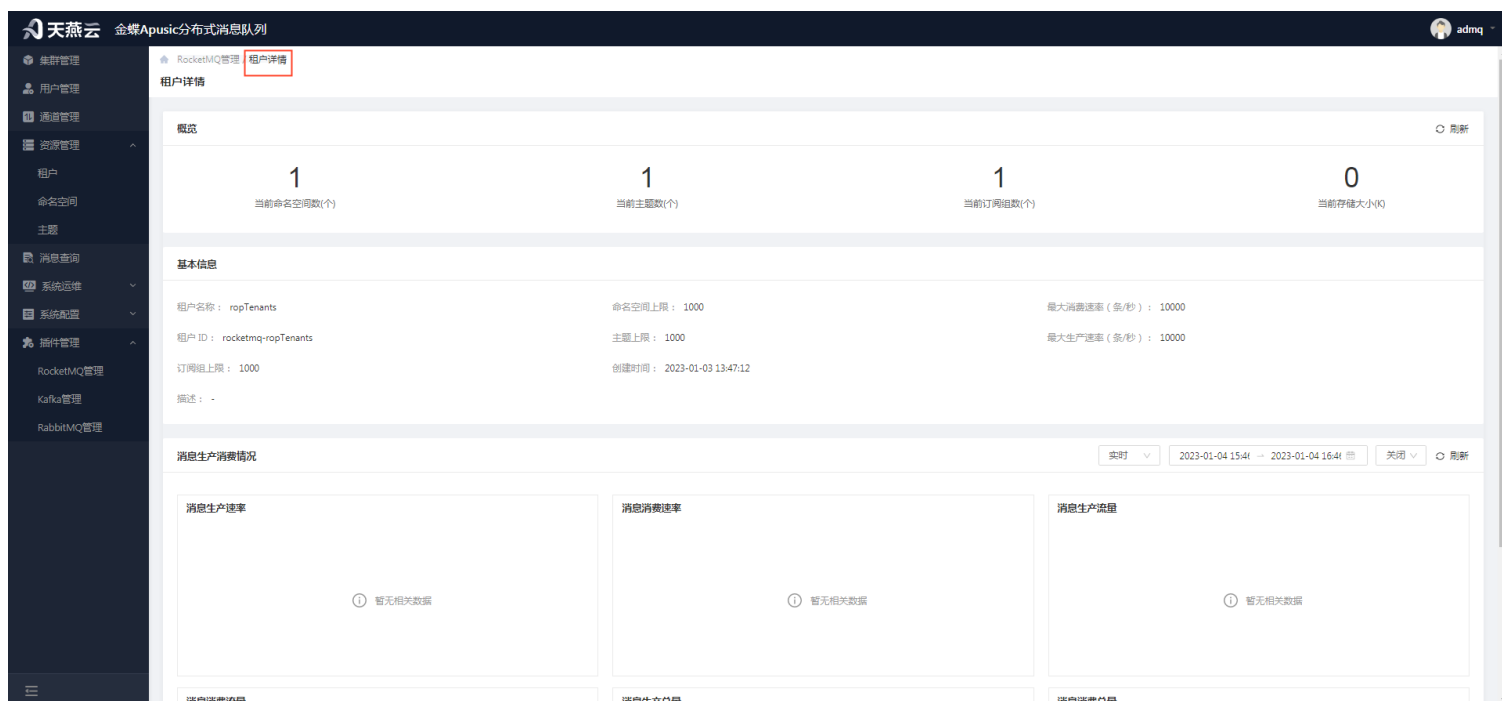
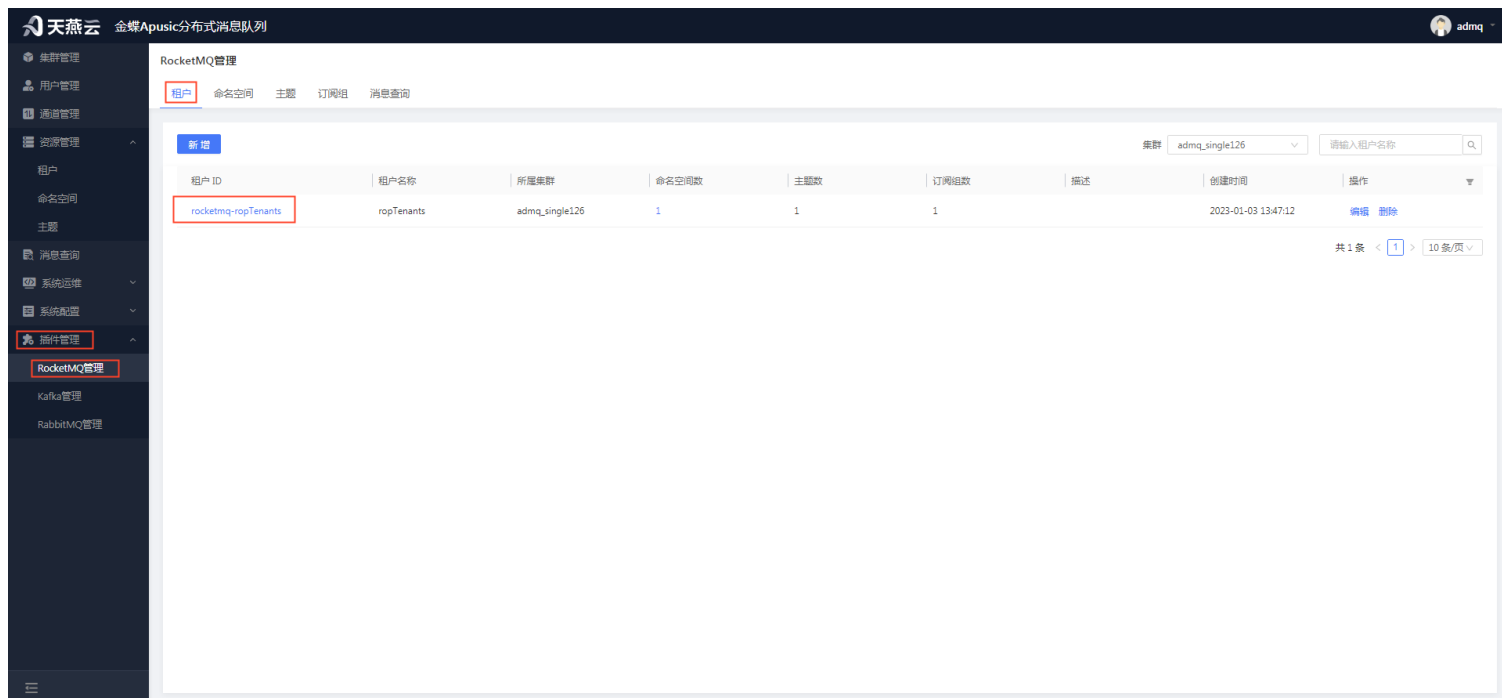
5.10.1.1.4 查询租户

进入【插件管理】->【RocketMQ管理】，选择“租户”页签，右上角选择集群名称，精确或模糊输入租户名称、可查询租户信息。



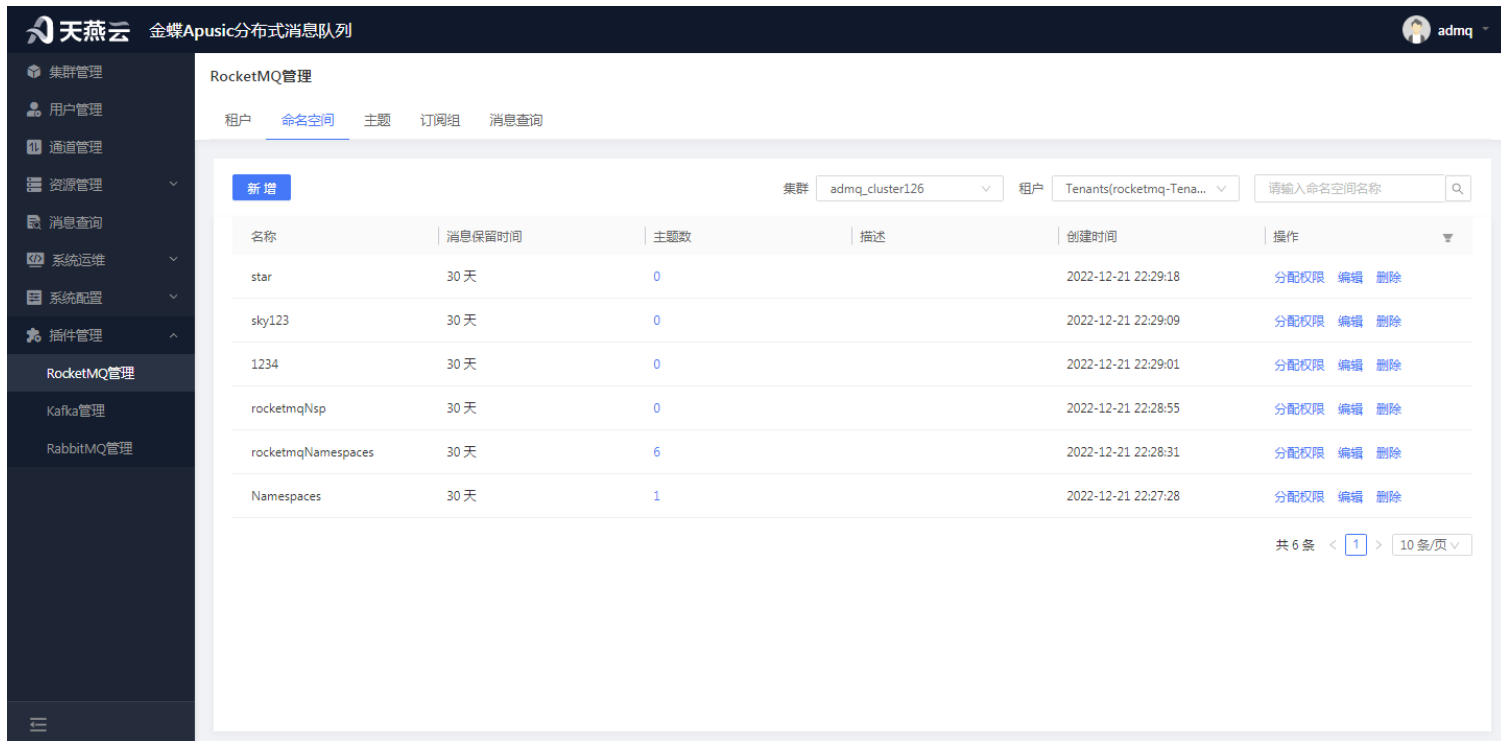
5.10.1.1.5 查看租户详情

进入【插件管理】->【RocketMQ管理】，选择“租户”页签，点击某个租户的租户ID，可查看租户详情。



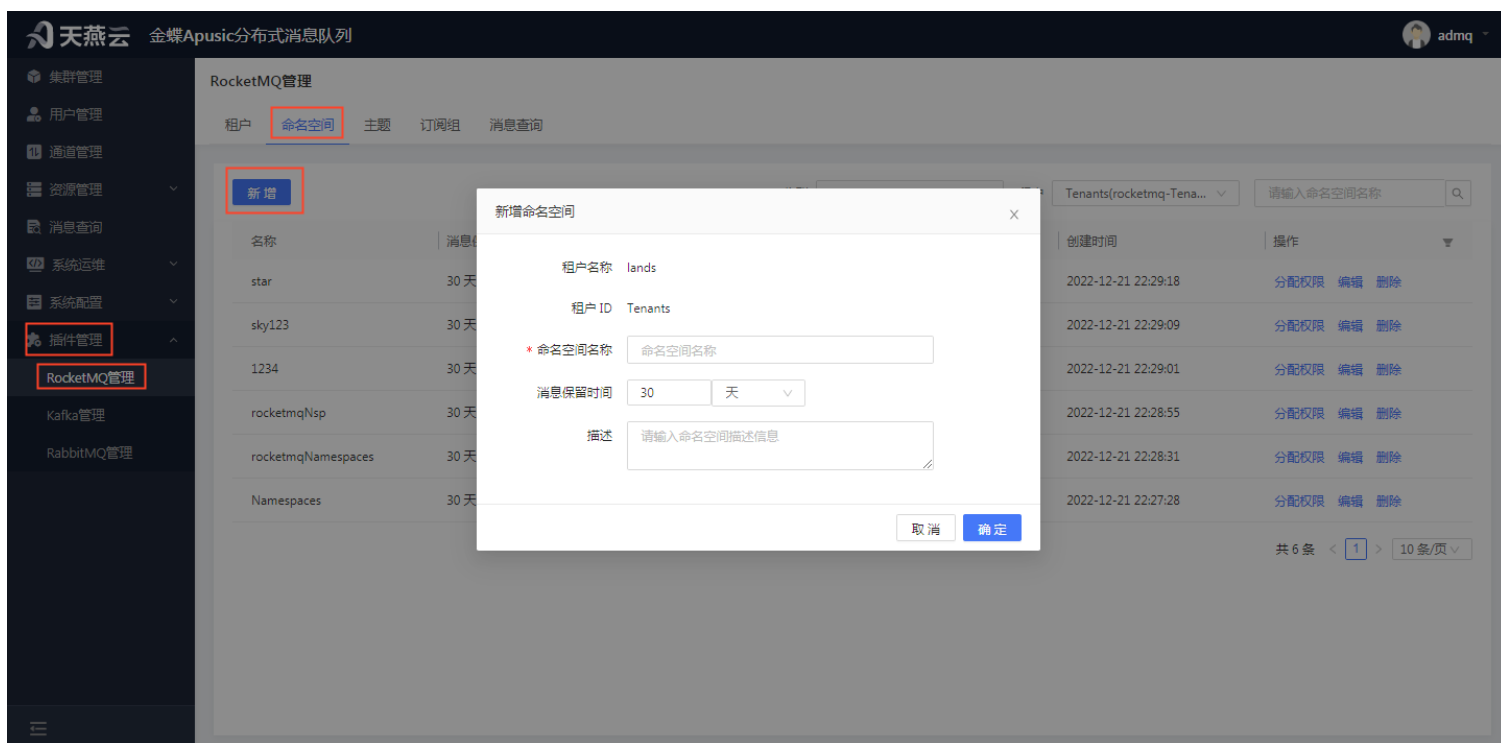
5.10.1.2 命名空间

命名空间作用于管理命名空间资源、命名空间用户权限。命名空间用于新增命名空间、编辑命名空间、删除命名空间、查询命名空间、新增命名空间权限、编辑命名空间权限、删除命名空间权限。



5.10.1.2.1 新增命名空间

进入【插件管理】->【RocketMQ管理】，选择“命名空间”页签，点击左上角【新增】按钮，可新增命名空间信息。注意：新增命名空间前，需满足集群服务可用。

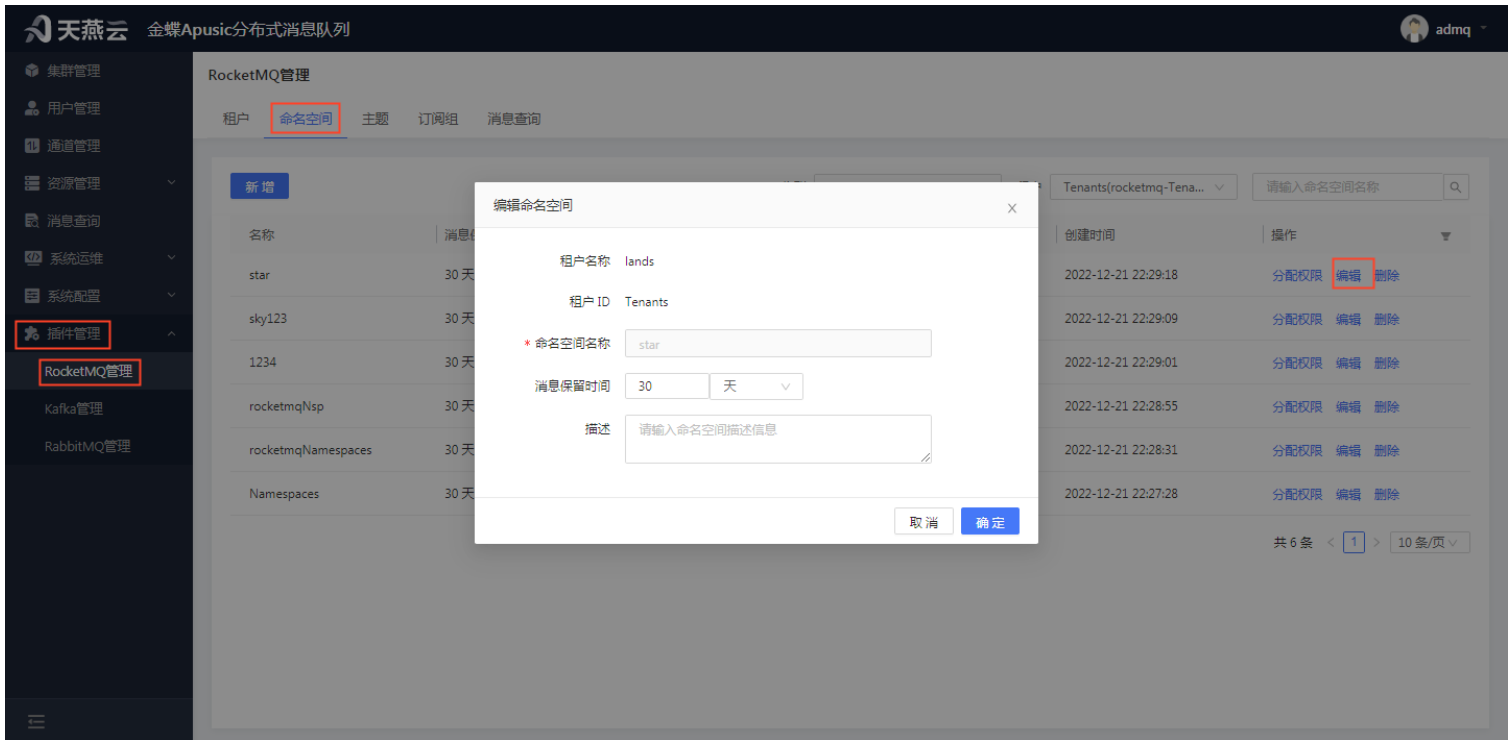


- 租户名称：命名空间所属的租户名称，根据选择的租户自动带出，不可修改
- 租户ID：命名空间所属租户的租户ID，根据选择的租户自动带出，不可修改
- 命名空间名称：命名空间的名称，自定义，唯一标识，可自定义
- 消息保留时间：消息的最大保留时间，超过则不保存消息，单位（分钟）、（小时）、（天），默认30天（最大值）

- 描述：命名空间的描述信息，可自定义

5.10.1.2.2 编辑命名空间

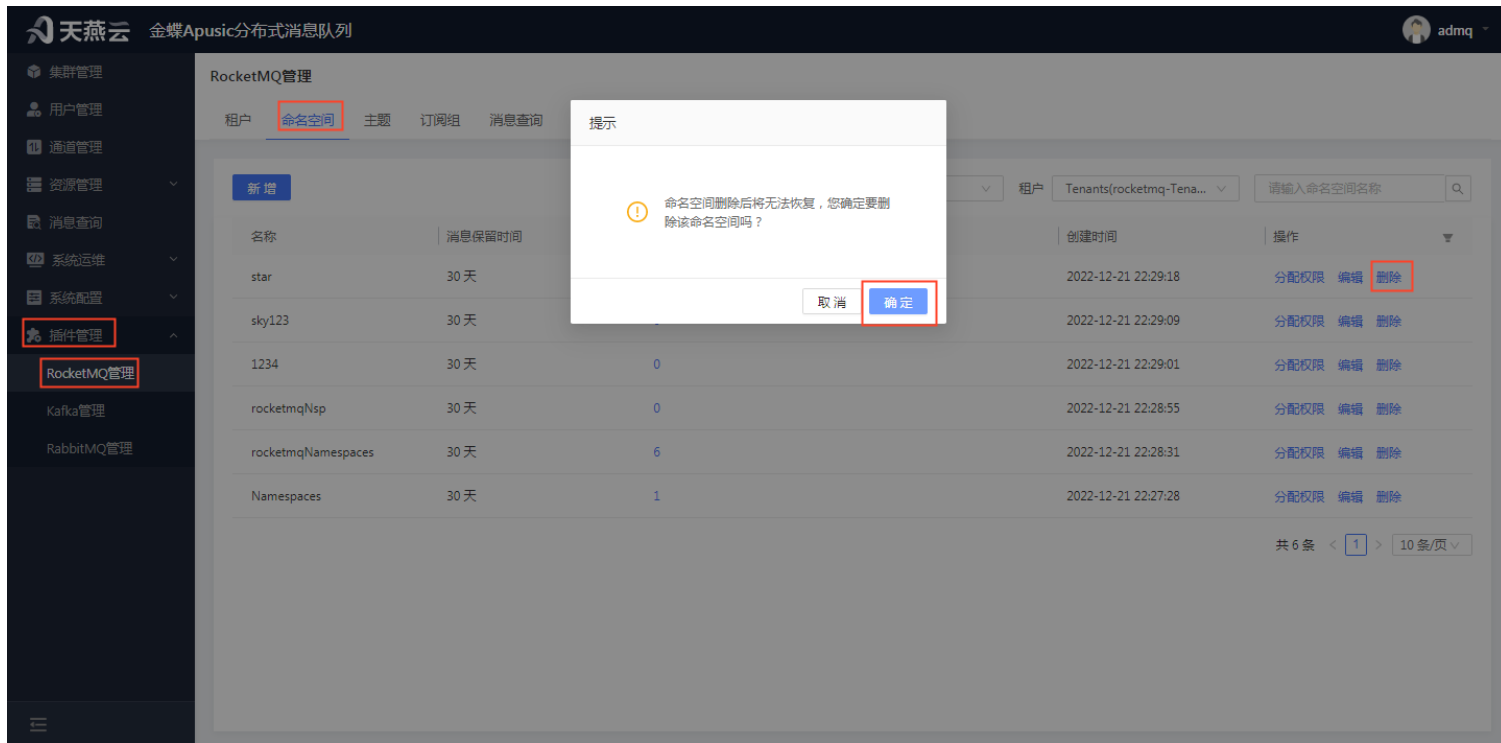
进入【插件管理】->【RocketMQ管理】，选择“命名空间”页签，选择要编辑的命名空间名称后的【编辑】按钮，可编辑命名空间信息。注意：编辑命名空间前，需满足集群服务可用。



- 租户名称：命名空间所属的租户名称，根据选择的租户自动带出，不可修改
- 租户ID：命名空间所属租户的租户ID，根据选择的租户自动带出，不可修改
- 命名空间名称：命名空间的名称，置灰显示，不可修改
- 消息保留时间：消息的最大保留时间，超过则不保存消息，单位（分钟）、（小时）、（天），可修改
- 描述：命名空间的描述信息，可修改

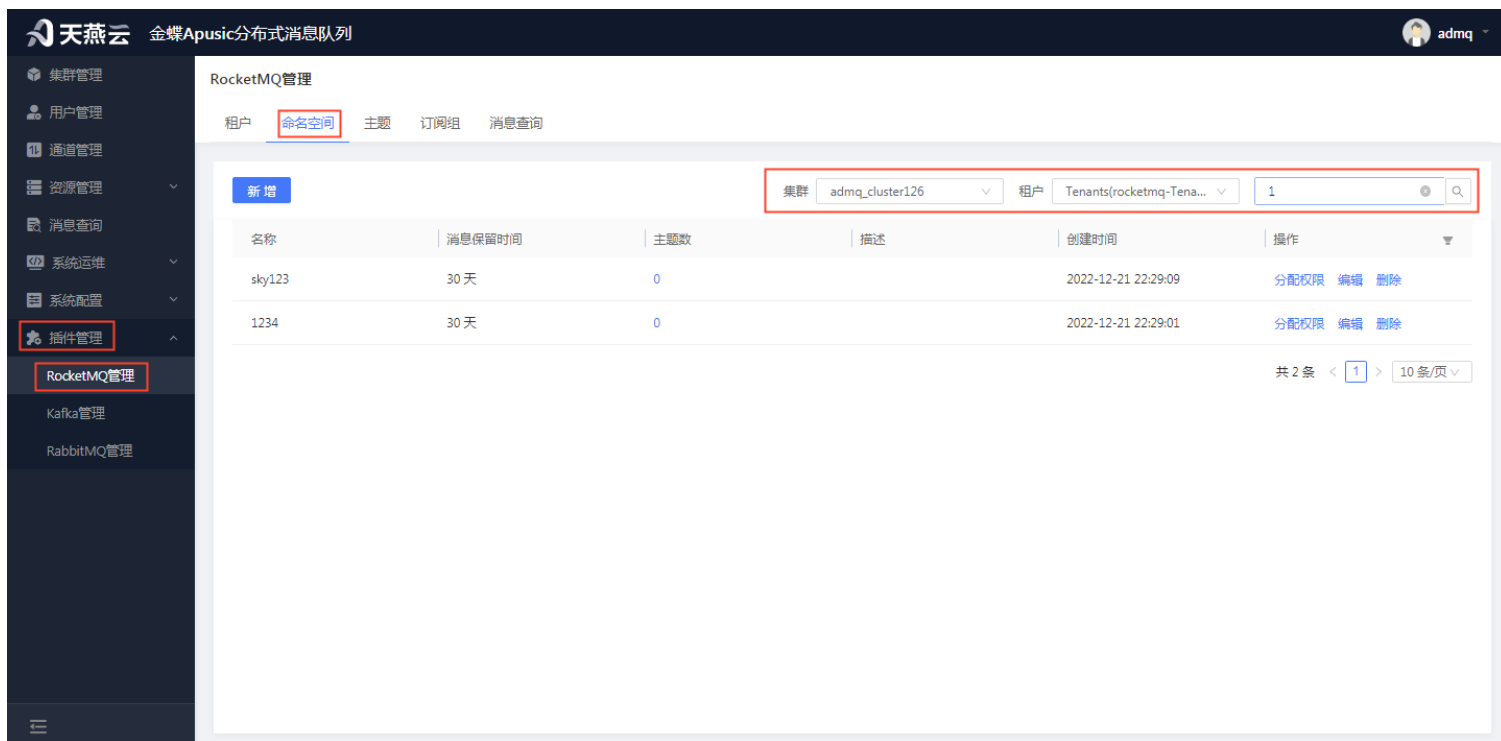
5.10.1.2.3 删除命名空间

进入【插件管理】->【Kafka管理】，选择“命名空间”页签，选择要删除的命名空间名称后的【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除命名空间信息。注意：删除命名空间前，需满足集群服务可用，且命名空间下无订阅组，删除命名空间后，命名空间下的主题信息也被删除。



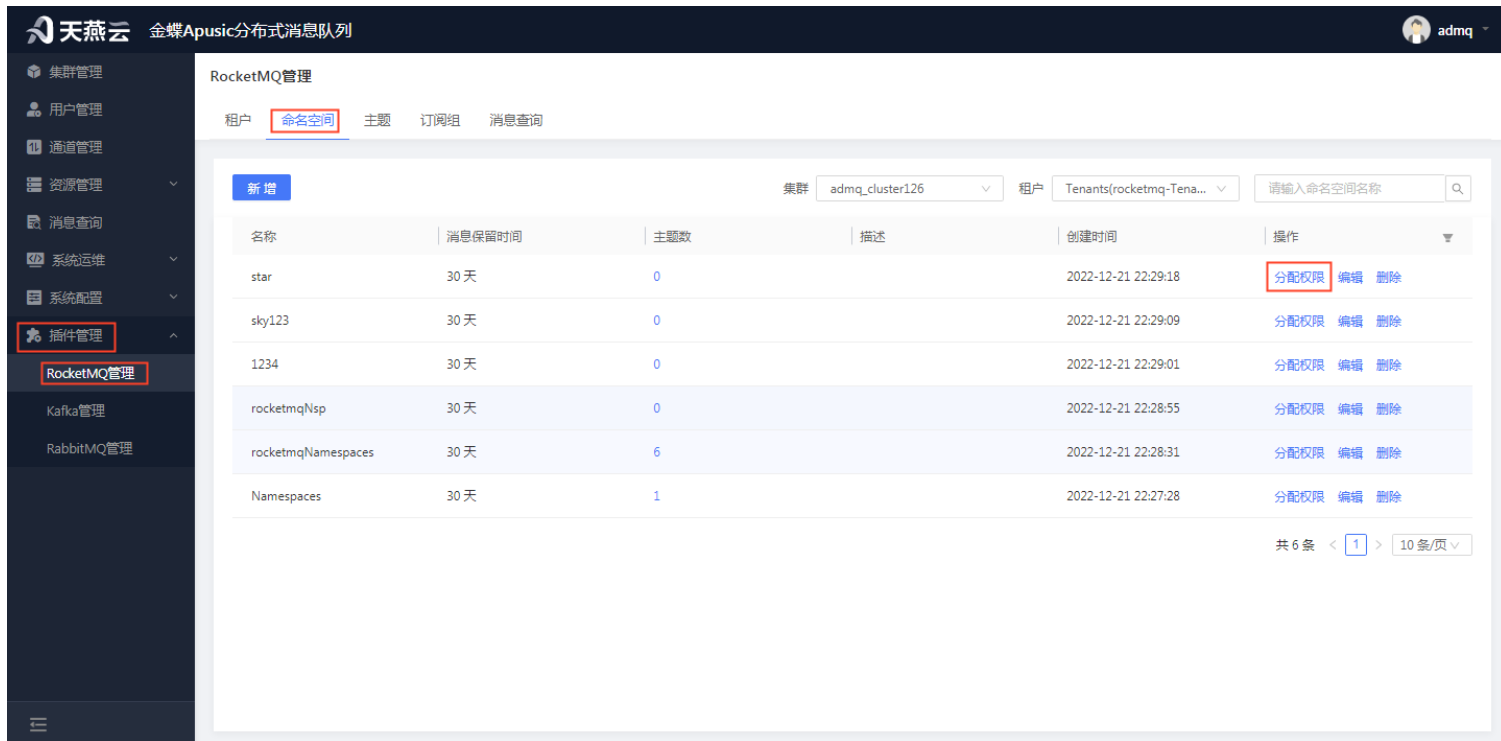
5.10.1.2.4 查询命名空间

进入【插件管理】->【RocketMQ管理】，选择“命名空间”页签，右上角选择集群名称、租户名称、精确或模糊输入命名空间名称，可查询命名空间信息。（注意：包含的也有）

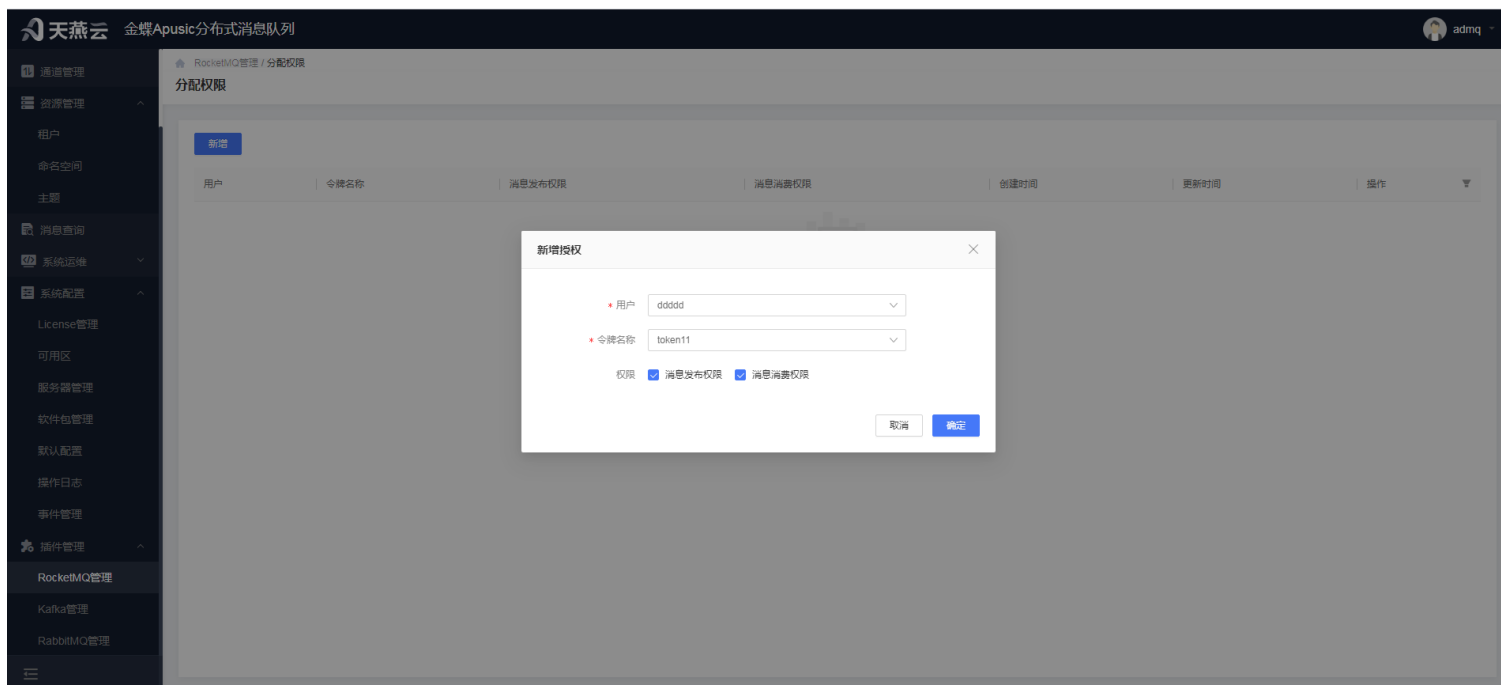


5.10.1.2.5 新增命名空间权限

进入【插件管理】->【RocketMQ管理】，选择“命名空间”页签，选择要新增授权的命名空间名称后的【分配权限】按钮后，会进入到分配权限页面。



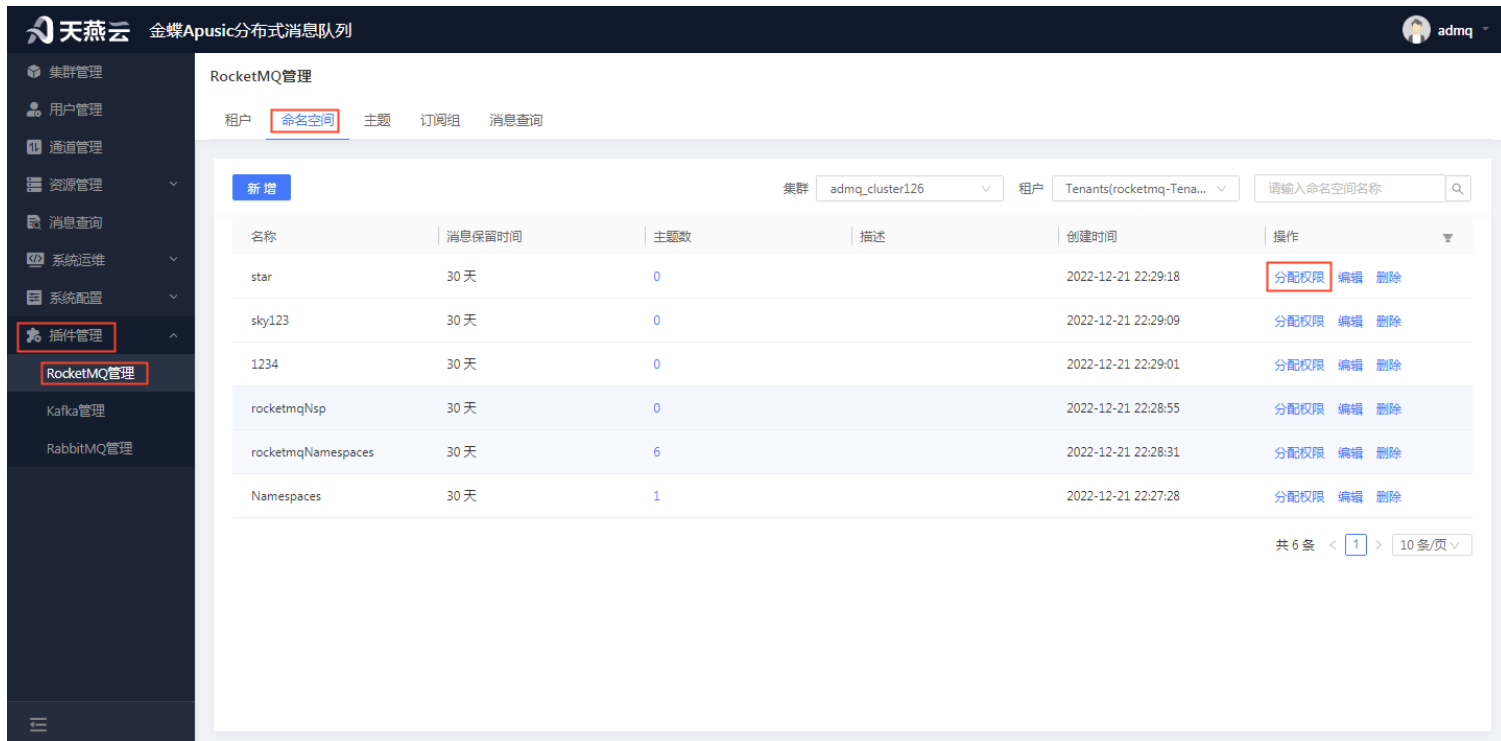
在分配权限页面，点击左上角的【新增】按钮，可新增命名空间下用户的权限信息。注意：新增权限前，需满足集群服务可用；授权主要是给普通用户提供授权，默认超级管理员拥有所有命名空间的消息发布权限和消息消费权限。



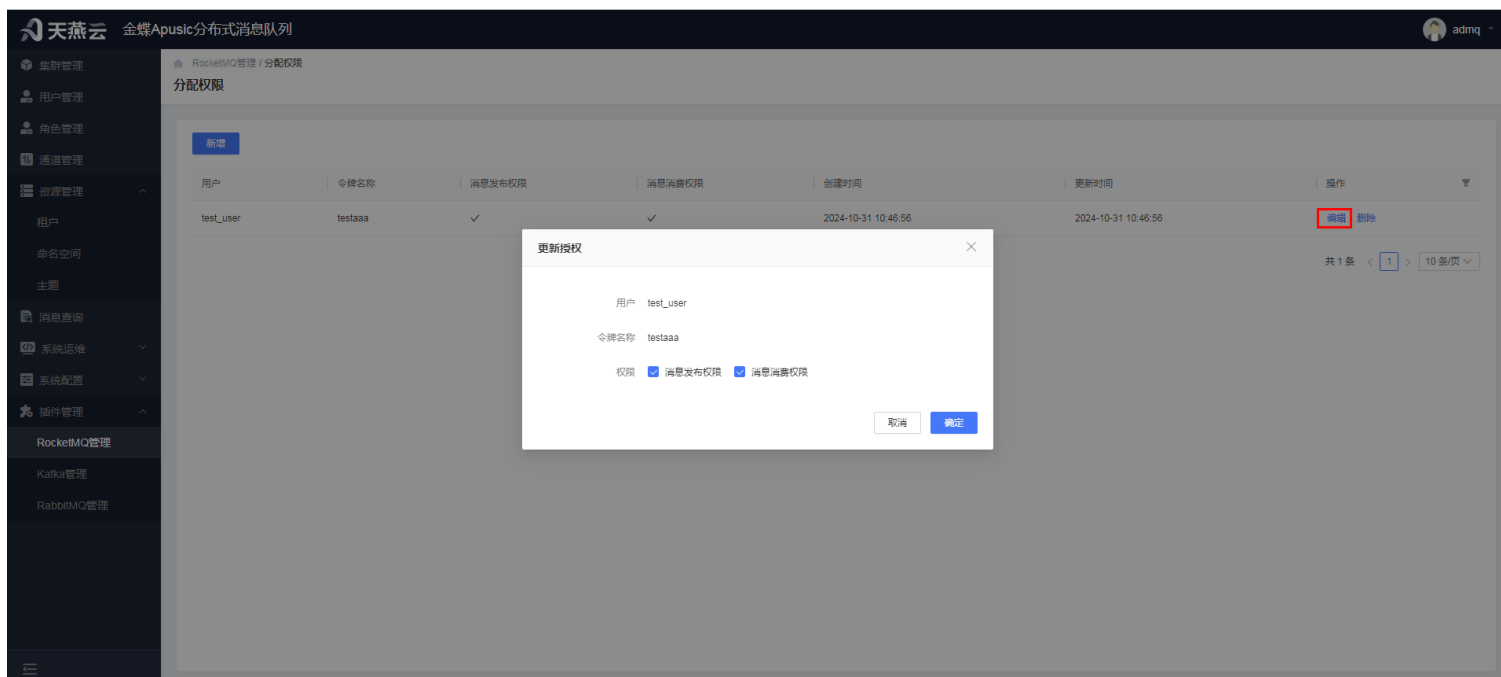
- 用户：下拉选择，获取的用户管理（4.1）的用户信息
- 令牌名称：下拉选择，选择用户下的令牌
- 权限：复选框，消息发布权限和消息消费权限

5.10.1.2.6 编辑命名空间权限

进入【插件管理】->【RocketMQ管理】，选择“命名空间”页签，选择要更新授权的命名空间名称后的【分配权限】按钮，跳转到分配权限页面。



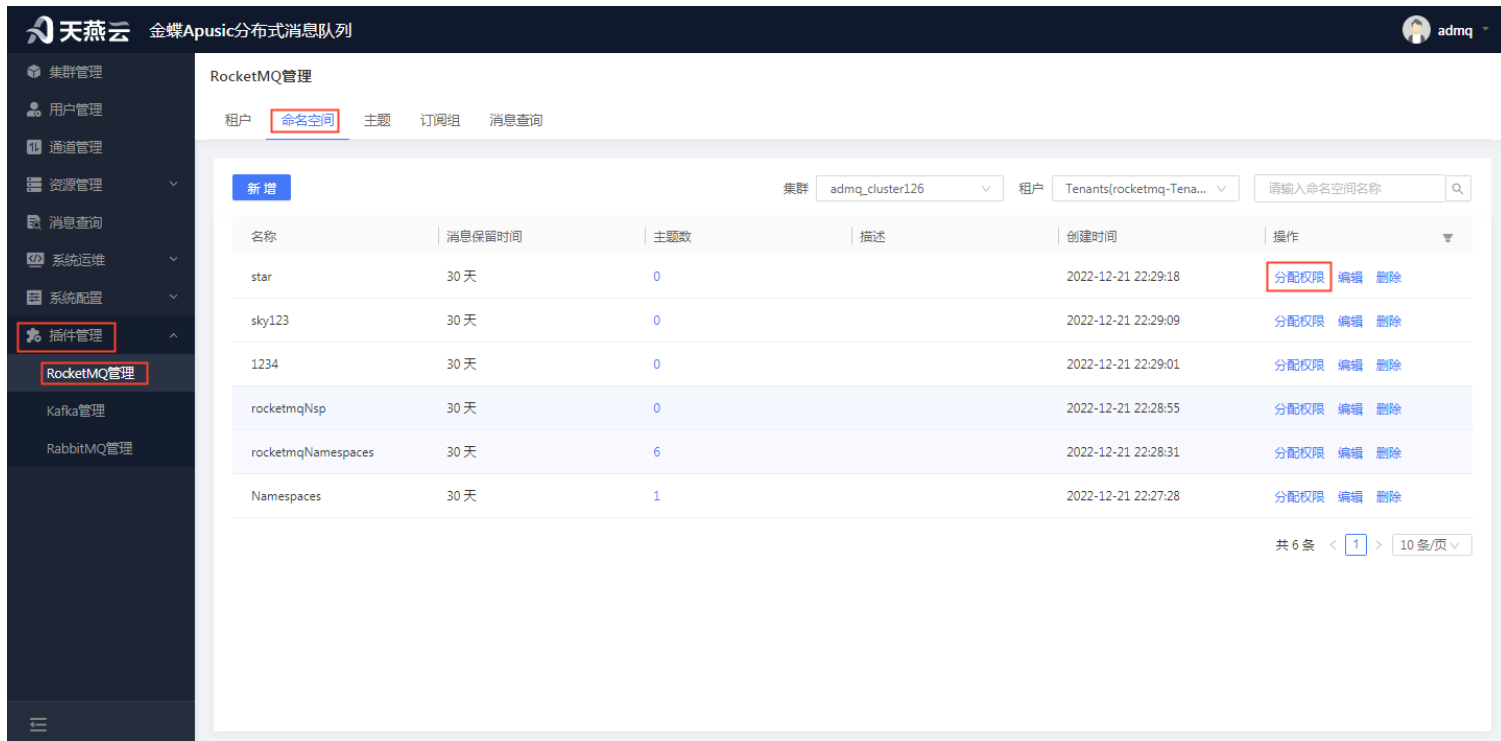
在分配权限页面，选择需要更新授权的用户，点击【编辑】按钮，可更新用户在命名空间下的权限。注意：更新权限前，需满足集群服务可用。



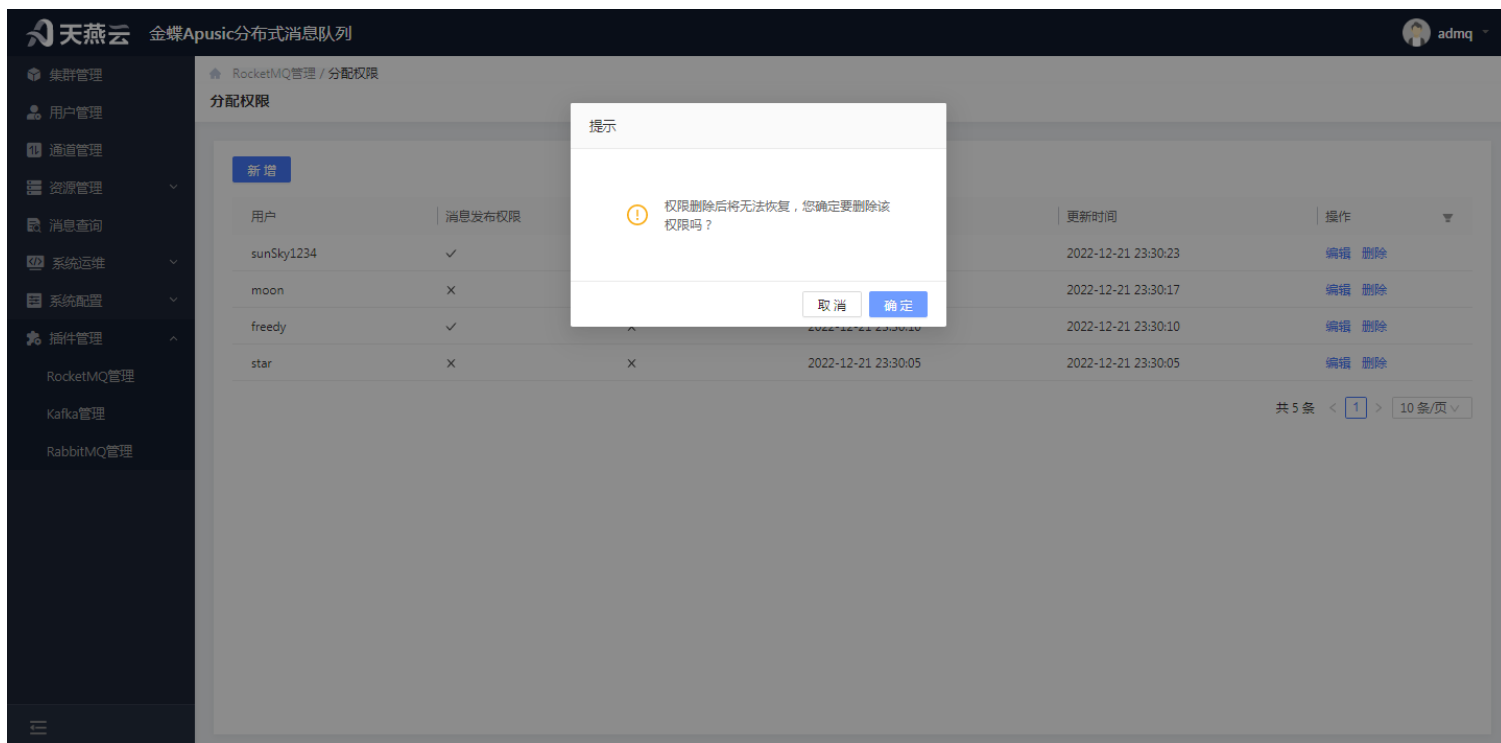
- 用户：用户名称，不可修改
- 令牌名称：令牌名称，不可修改
- 权限：复选框，消息发布权限和消息消费权限，可修改

5.10.1.2.7 删除命名空间权限

进入【插件管理】->【Kafka管理】，选择“命名空间”页签，选择要删除授权的命名空间名称后的【分配权限】按钮，进入到分配权限页面。

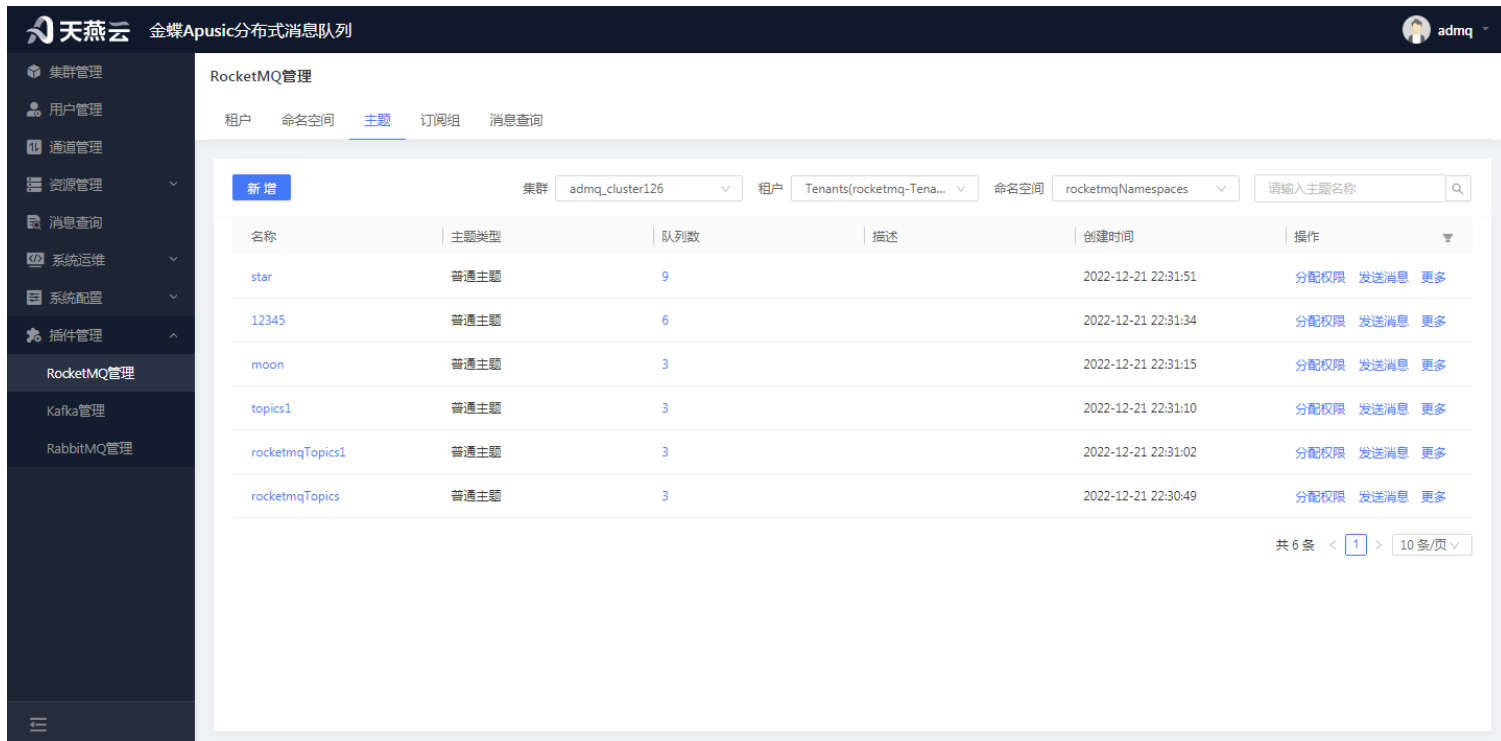


在分配权限页面，选择需要删除授权的用户，点击【删除】按钮，在弹出的确认删除弹框点击【确定】按钮，可删除用户在命名空间下的权限。注意：删除权限前，需满足集群服务可用。



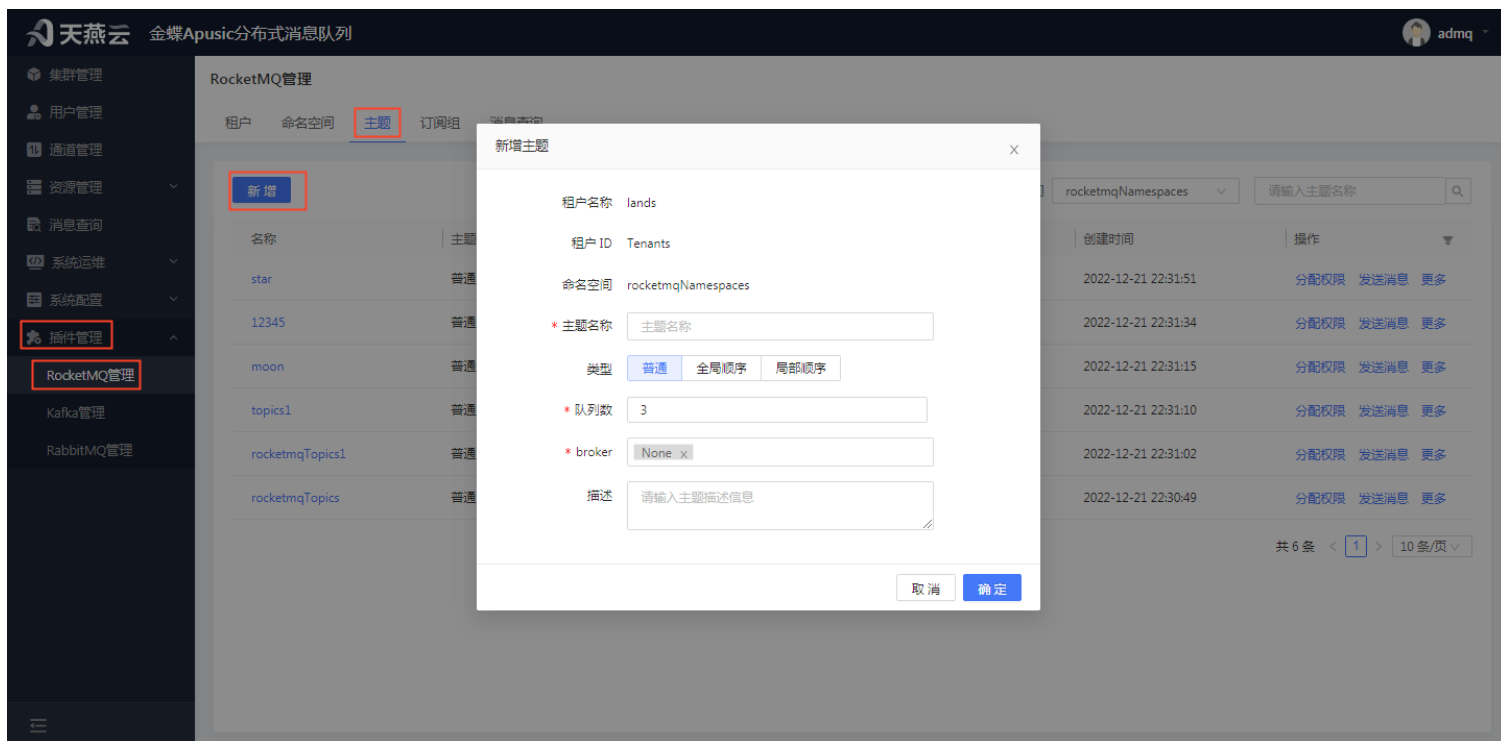
5.10.1.3 主题

主题作用于管理主题资源、主题的用户权限。主题用于新增主题、编辑主题、删除主题、查询主题、新增主题权限、编辑主题权限、删除主题权限、发送消息。



5.10.1.3.1 新增主题

进入【插件管理】->【RocketMQ管理】，选择“主题”页签，点击左上角【新增】按钮，可新增主题信息。注意：新增主题前，需满足集群服务可用。

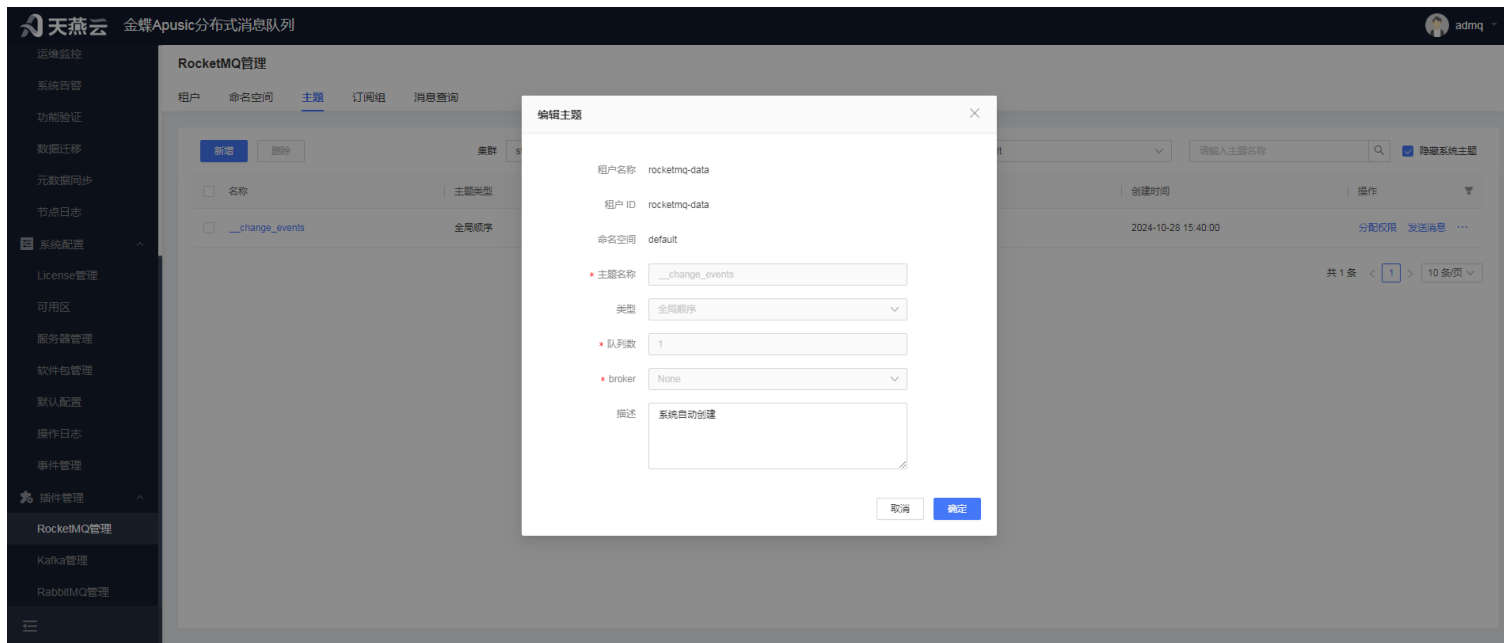
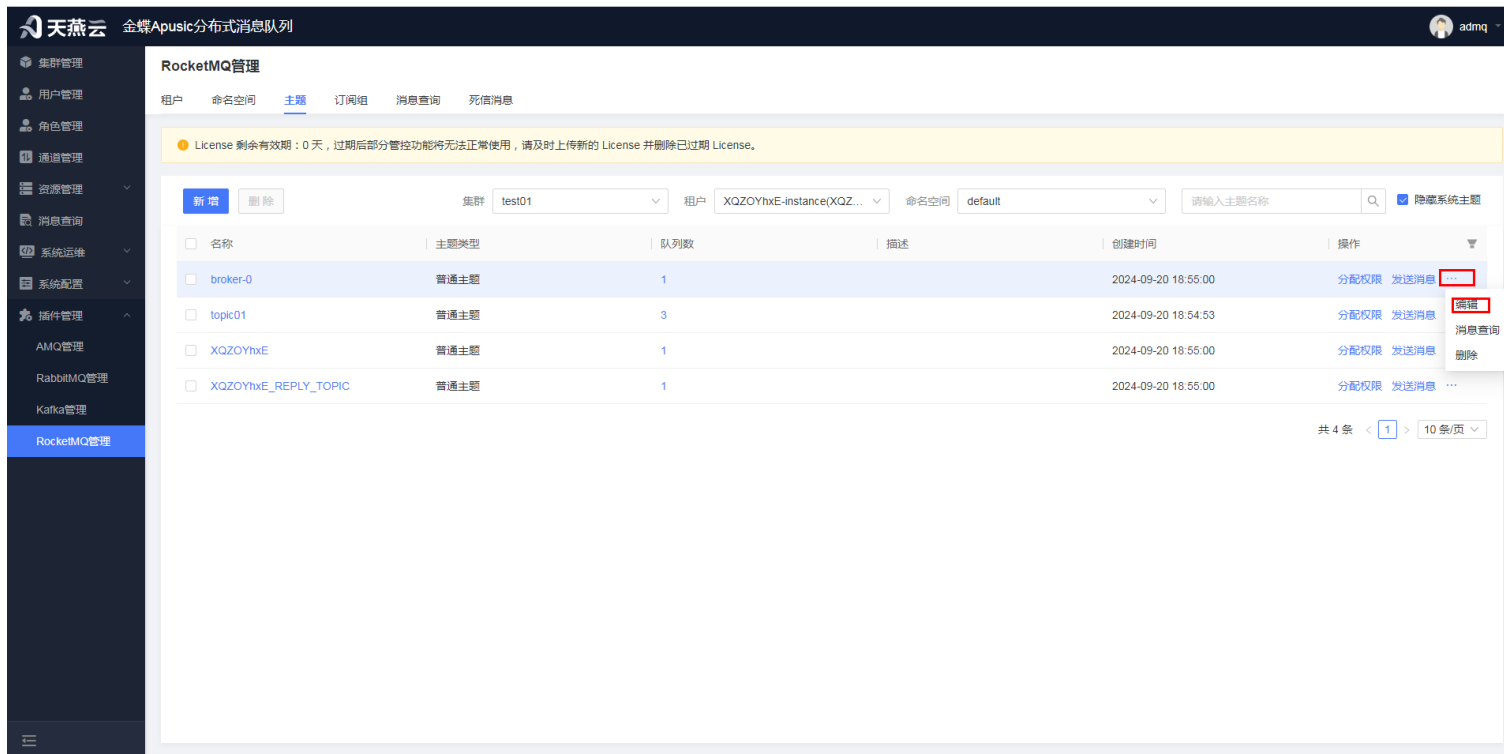


- 租户名称: 主题所属的租户的名称，根据选择的租户自动带出，不可修改
- 租户ID: 主题所属租户的租户ID，根据选择的租户自动带出，不可修改
- 命名空间: 主题所属命名空间名称，根据选择的命名空间自动带出，不可修改
- 主题名称: 主题的名称，自定义，唯一标识

- 类型：主题类型，单选项，包括：普通、全局顺序、局部顺序
- 队列数：主题下的队列数量，默认且最小是3，可自定义
- broker：主题所在集群的broker，若是插件加载的，则是None
- 描述：主题的描述信息

5.10.1.3.2 编辑主题

进入【插件管理】->【RocketMQ管理】，选择"主题"页签，选择要编辑的主题名称后的【...】按钮->【编辑】按钮，可编辑主题信息。注意：编辑主题前，需满足集群服务可用。（加图片）

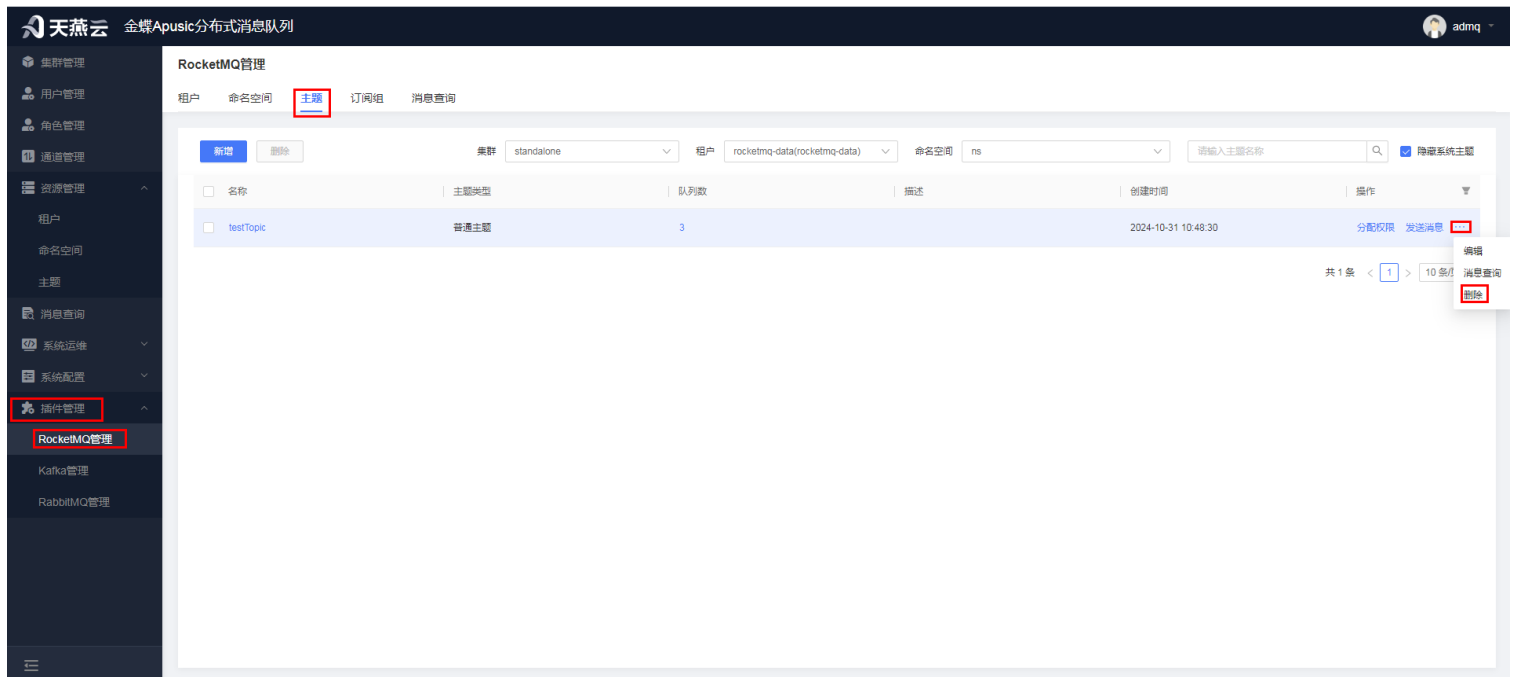


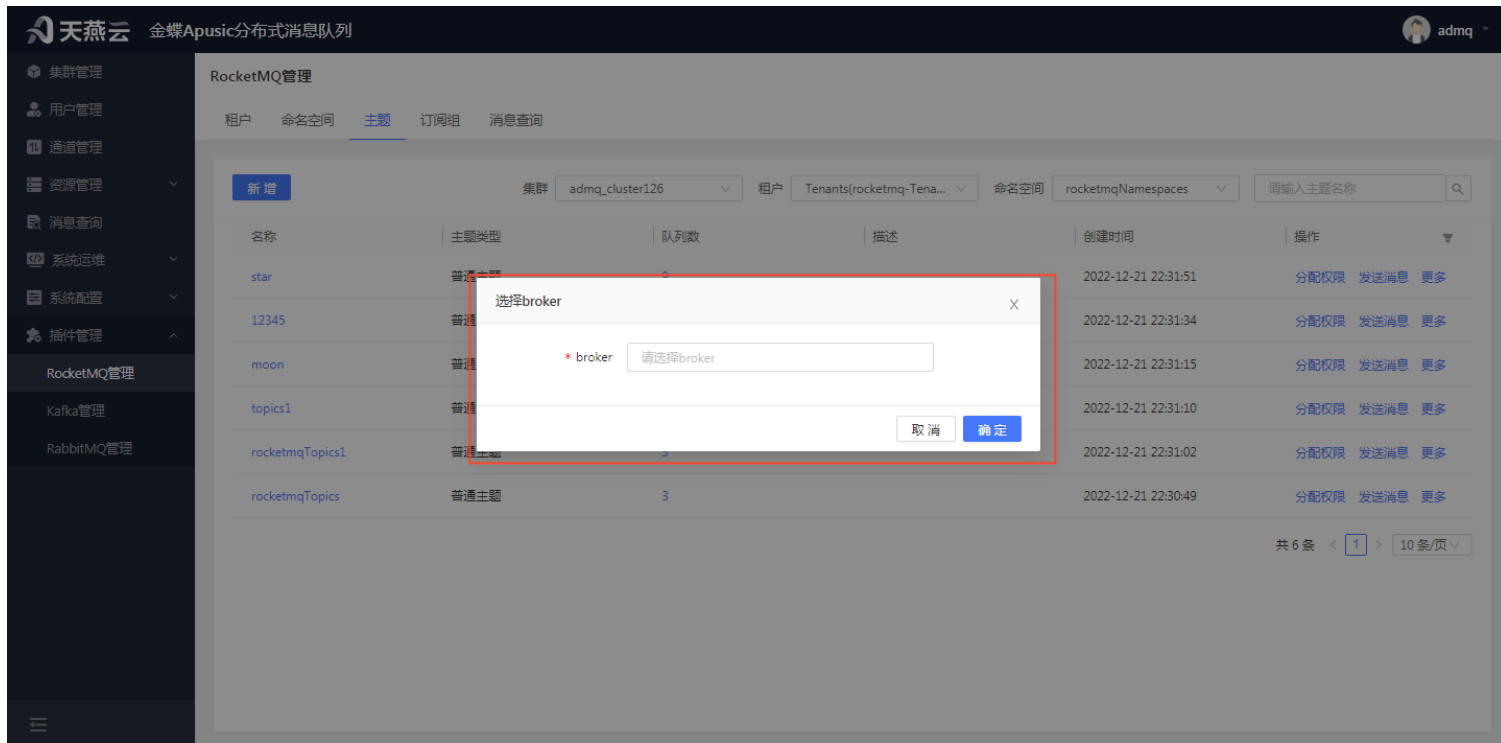
- 租户名称：主题所属租户名称，根据选择的租户自动带出，不可修改

- 租户ID：主题所属租户的租户ID，根据选择的租户自动带出，不可修改
- 命名空间：主题所属命名空间名称，根据选择的命名空间自动带出，不可修改
- 主题名称：主题的名称，置灰显示，不可修改
- 类型：主题类型，置灰显示，不可修改
- 队列数：主题下的队列数量，可修改
- broker：主题所在集群的broker，不可修改
- 描述：主题的描述信息

5.10.1.3.3 删除主题

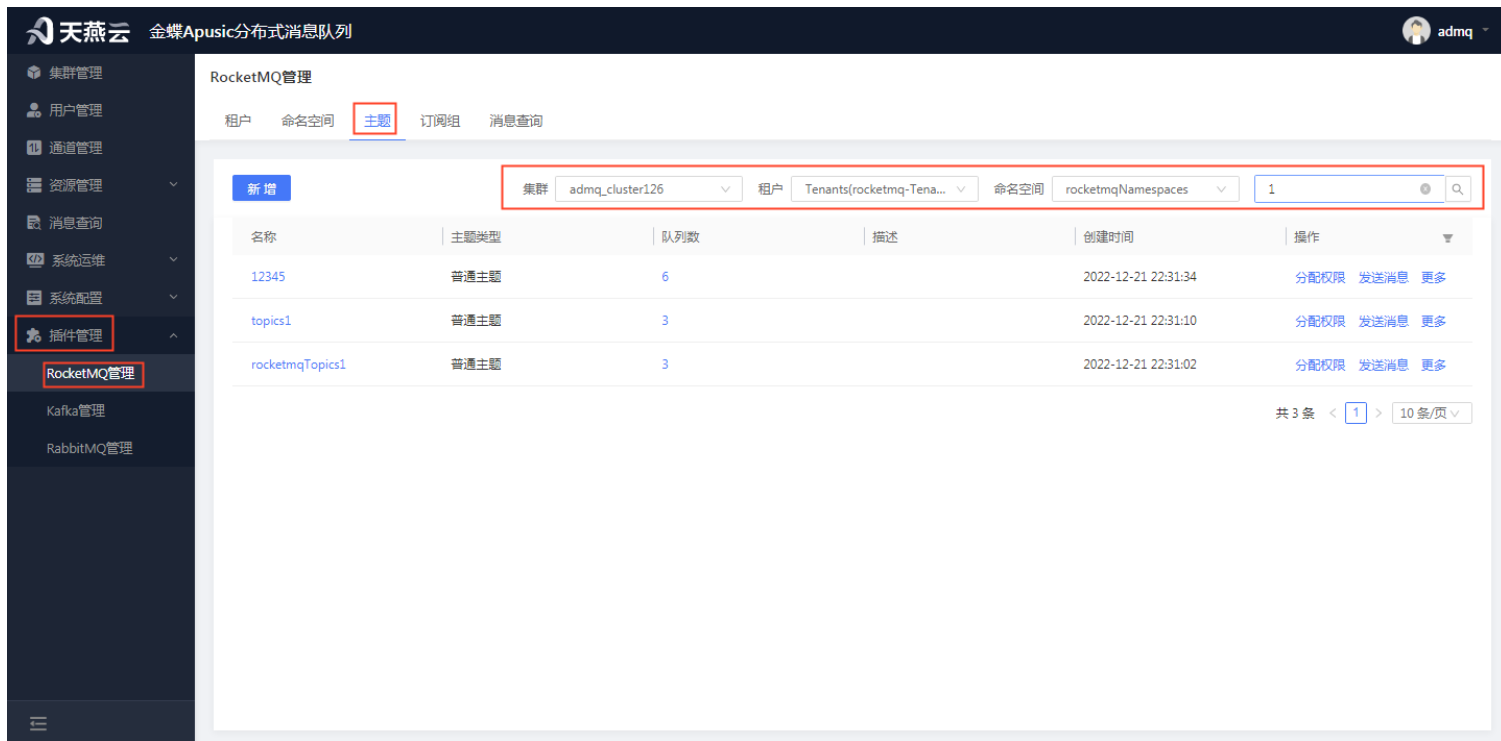
进入【插件管理】->【RocketMQ管理】，选择"主题"页签，选择要删除的主题名称后的【...】-【删除】按钮，在弹出的弹窗，先选择主题所在的broker，然后点击【确定】按钮，可删除主题信息。注意：删除主题前，需满足集群服务可用。





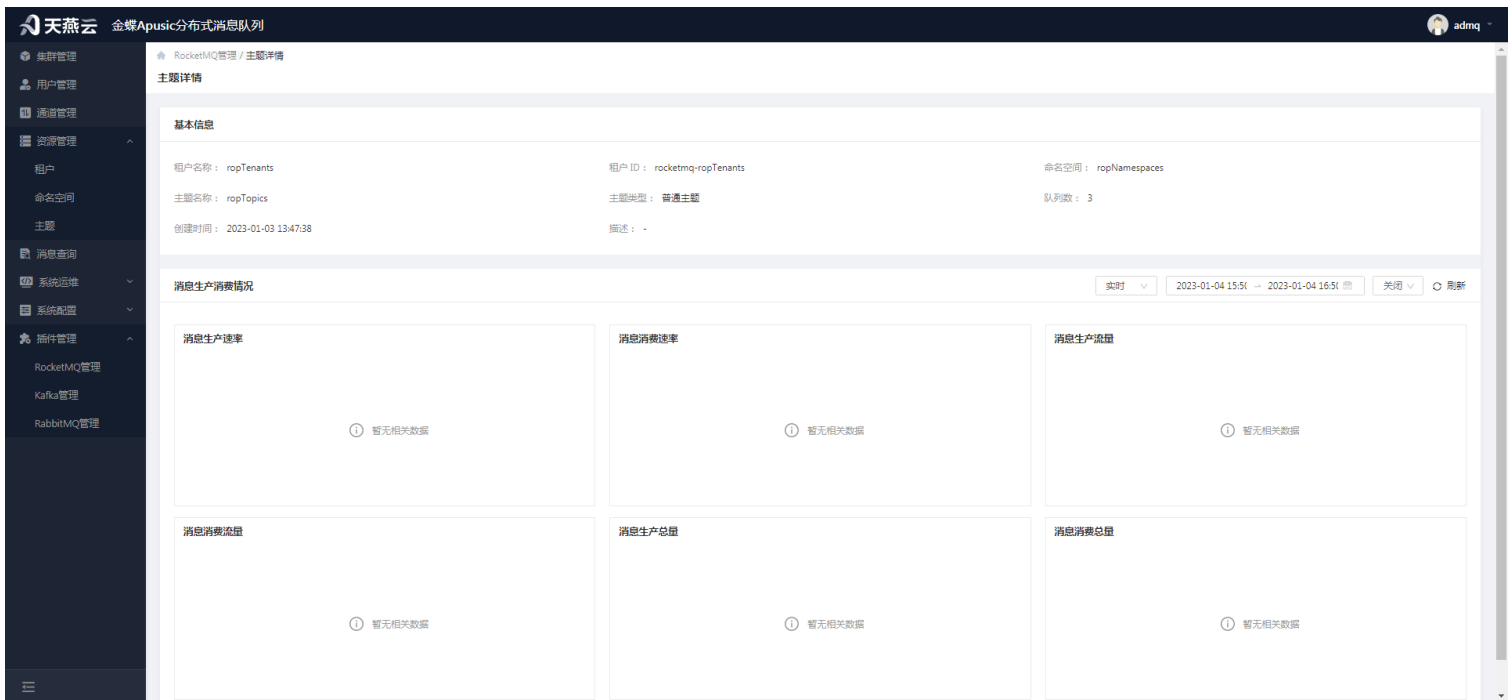
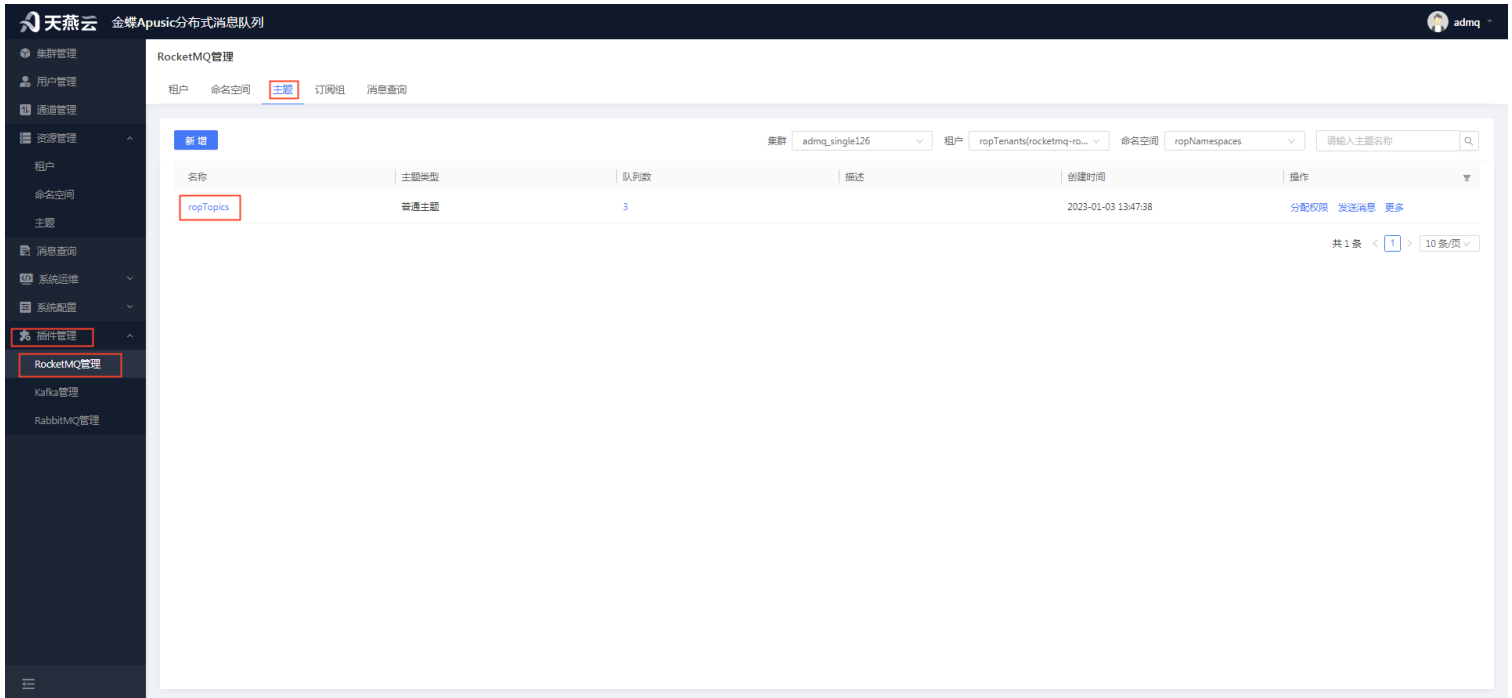
5.10.1.3.4 查询主题

进入【插件管理】->【RocketMQ管理】，选择"主题"页签，右上角选择集群名称、租户名称、命名空间名称、输入主题名称，可查询主题信息。



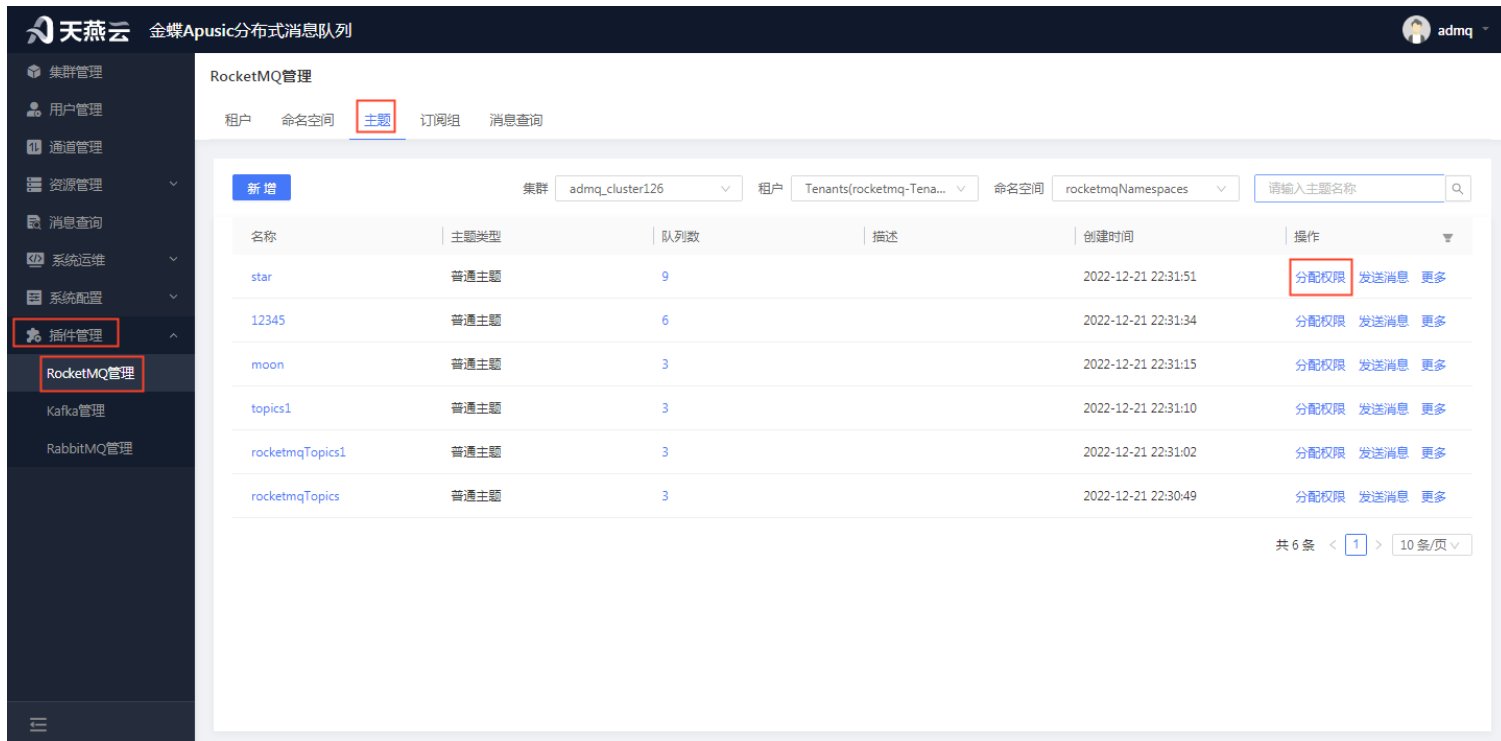
5.10.1.3.5 查看主题详情

进入【插件管理】->【RocketMQ管理】，选择"主题"页签，点击某个主题的名称，可查看主题详情。

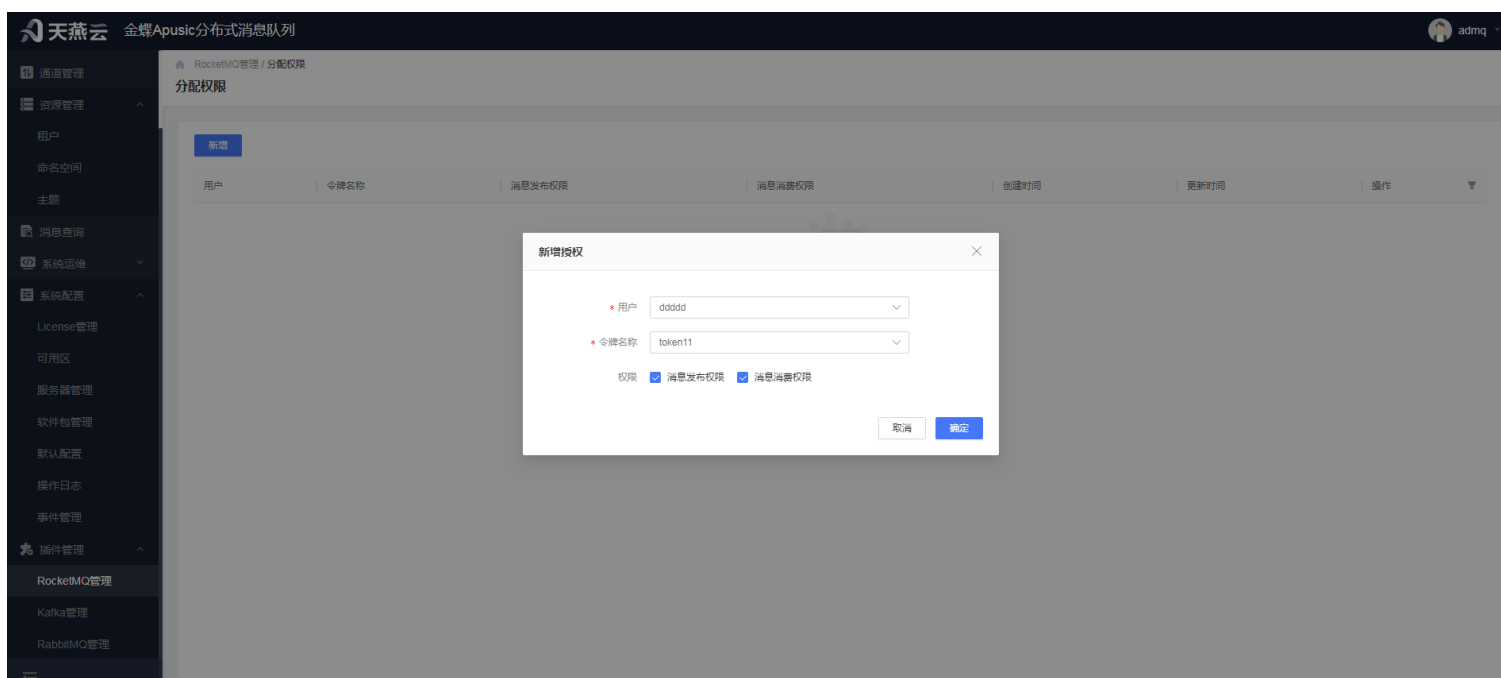


5.10.1.3.6 新增主题权限

进入【插件管理】->【RocketMQ管理】，选择“主题”页签，选择要新增授权的主题名称后的【分配权限】按钮，跳转到分配权限页面。



在分配权限页面，点击左上角的【新增】按钮，可新增主题下用户的权限信息。注意：新增权限前，需满足集群服务可用；授权主要是给普通用户提供授权，默认超级管理员拥有所有主题的消息发布权限和消息消费权限；若主题所属命名空间也有权限，则按最小粒度，主题设置的权限有效；若用户在命名空间无权限则需要先对命名空间进行授权。

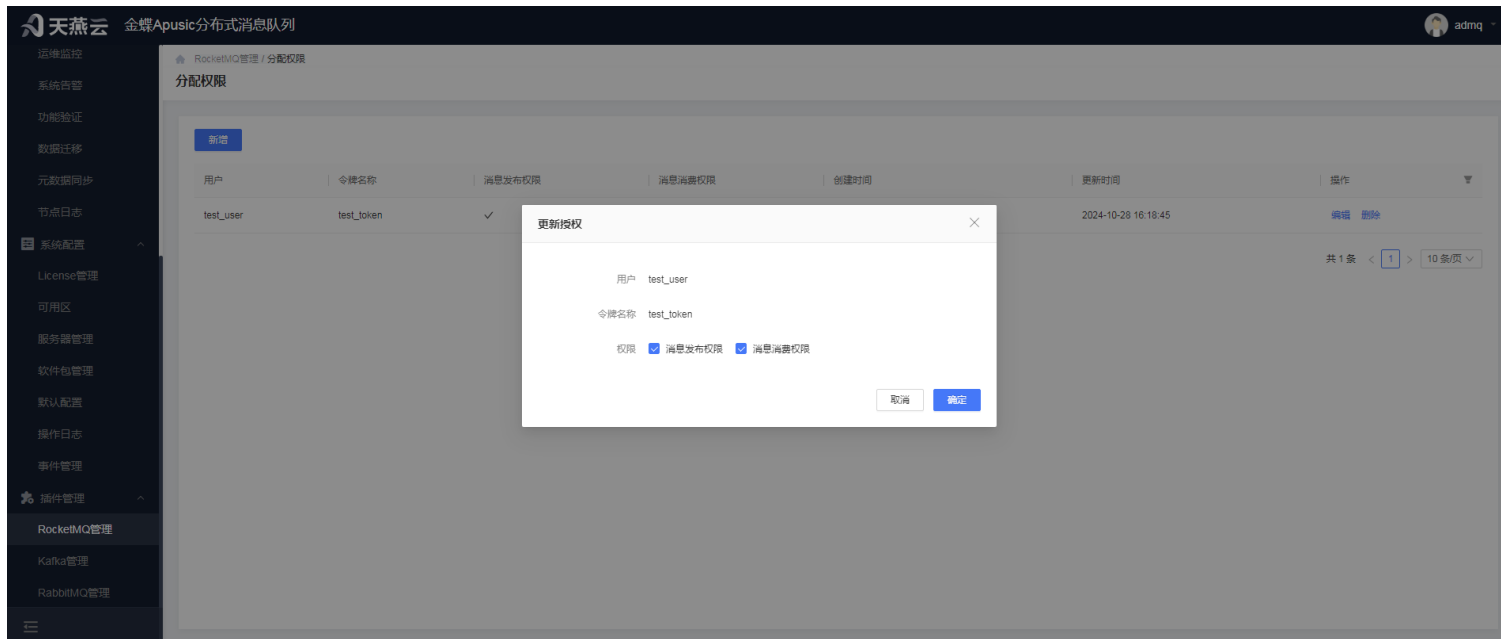


- 用户：下拉选择，获取的用户管理（4.1）的用户信息
- 权限：复选框，消息发布权限和消息消费权限

5.10.1.3.7 编辑主题权限

进入【插件管理】->【RocketMQ管理】，选择"主题"页签，选择要更新授权的主题名称后的【更多】按钮->【分配权限】按钮，进入到分配权限页面。

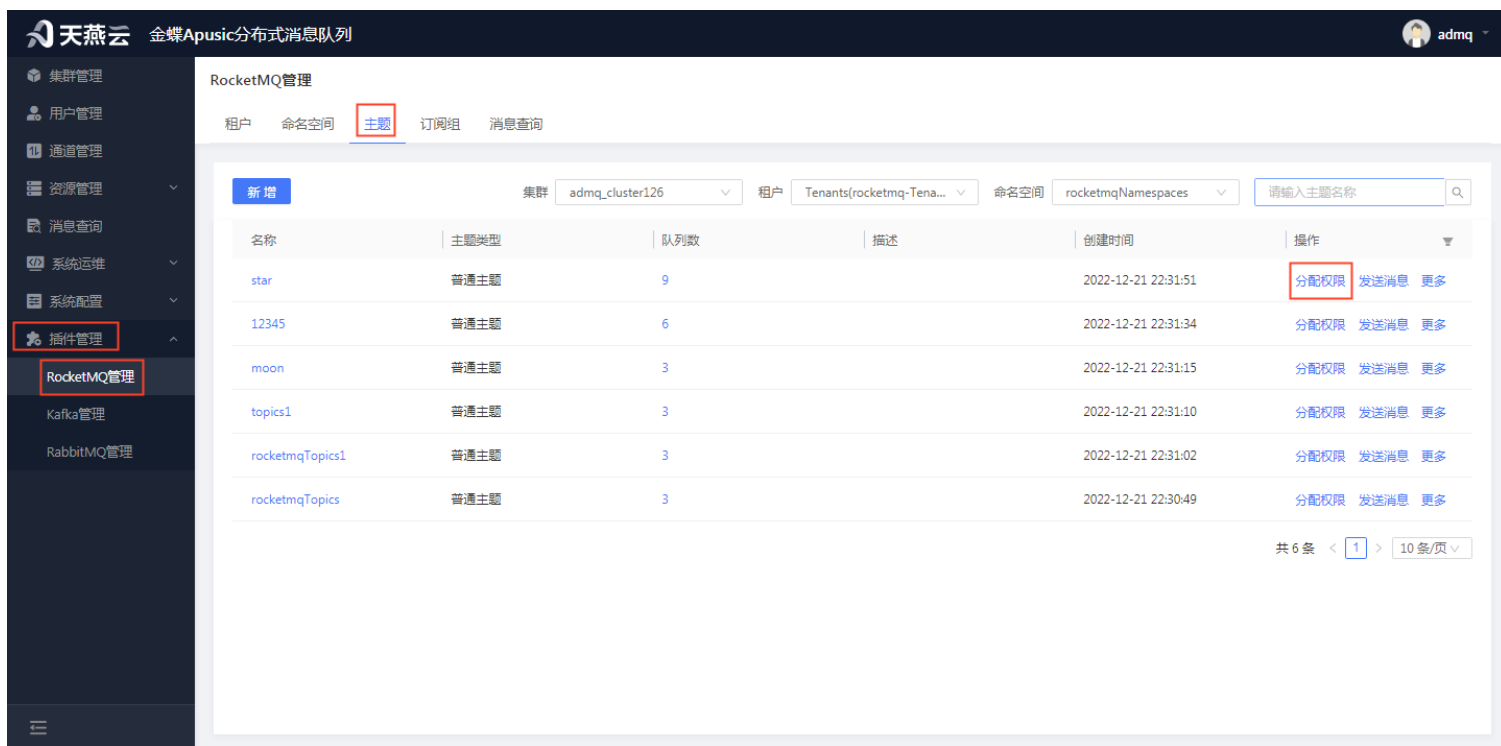
在分配权限页面，选择需要更新授权的用户，点击【编辑】按钮，可更新用户在主题下的权限。注意：更新权限前，需满足集群服务可用；若主题所属命名空间也有权限，则按最小粒度，主题设置的权限有效。



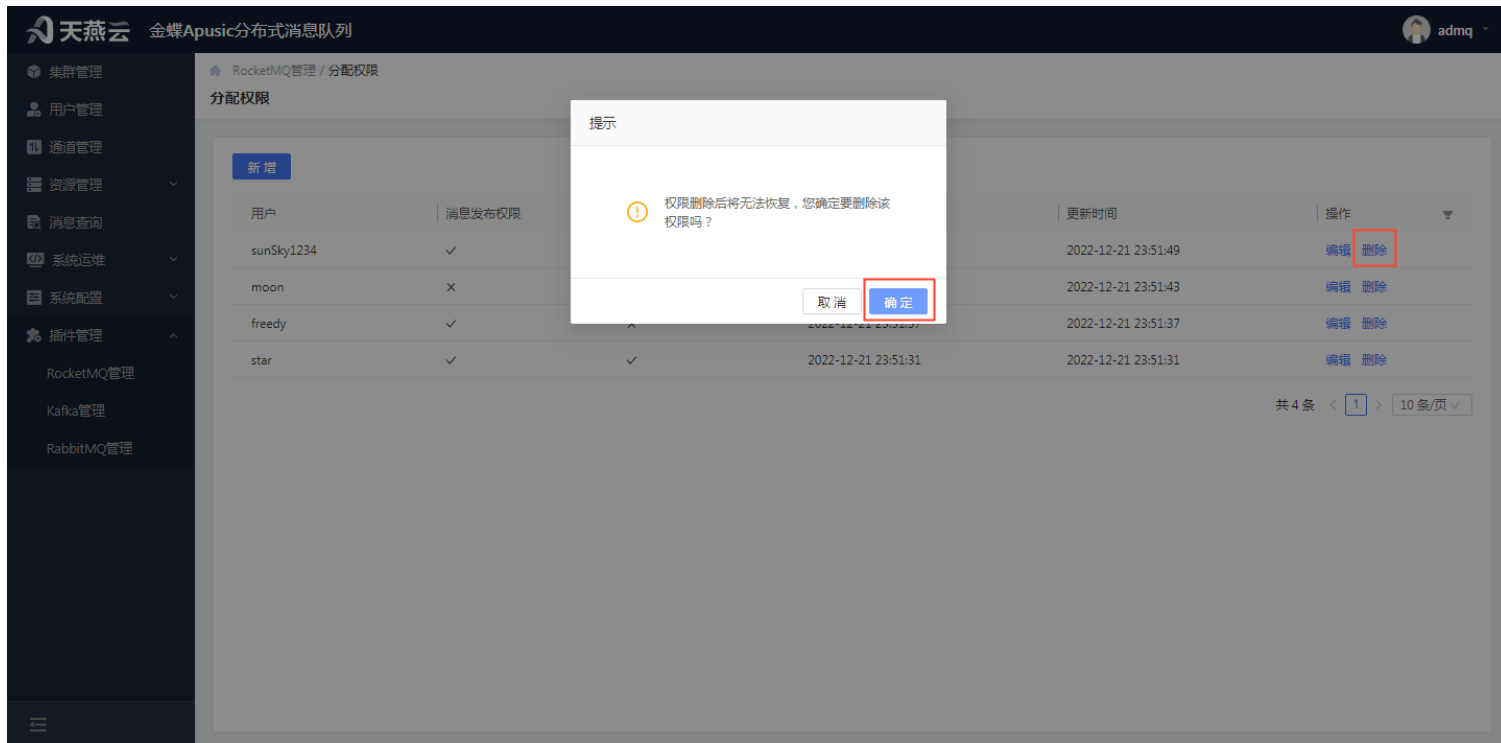
- 用户：用户名称，不可修改
- 令牌名称：令牌名称，不可修改
- 权限：复选框，消息发布权限和消息消费权限

5.10.1.3.8 删除主题权限

进入【插件管理】->【Kafka管理】，选择"主题"页签，选择要删除授权的主题名称后的【更多】按钮->【分配权限】按钮，进入到分配权限页面。

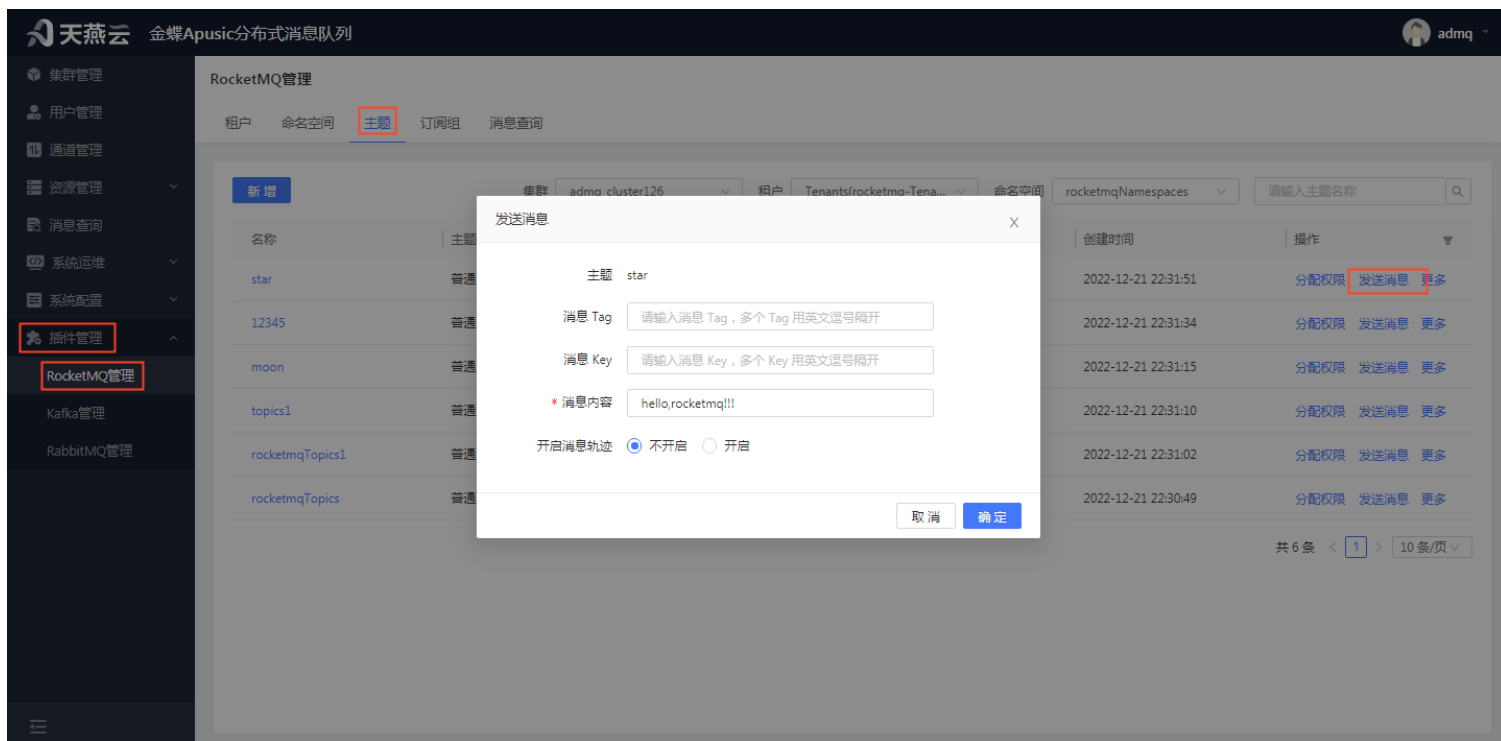


在分配权限页面，选择需要删除授权的用户，点击【删除】按钮，在弹出的确认删除弹框点击【确定】按钮，可删除用户在主题下的权限。注意：删除权限前，需满足集群服务可用；若主题所属命名空间也有权限，则按最小粒度，主题设置的权限有效，若删除权限的用户的命名空间下有权限，则命名空间权限生效。



5.10.1.3.9 发送消息

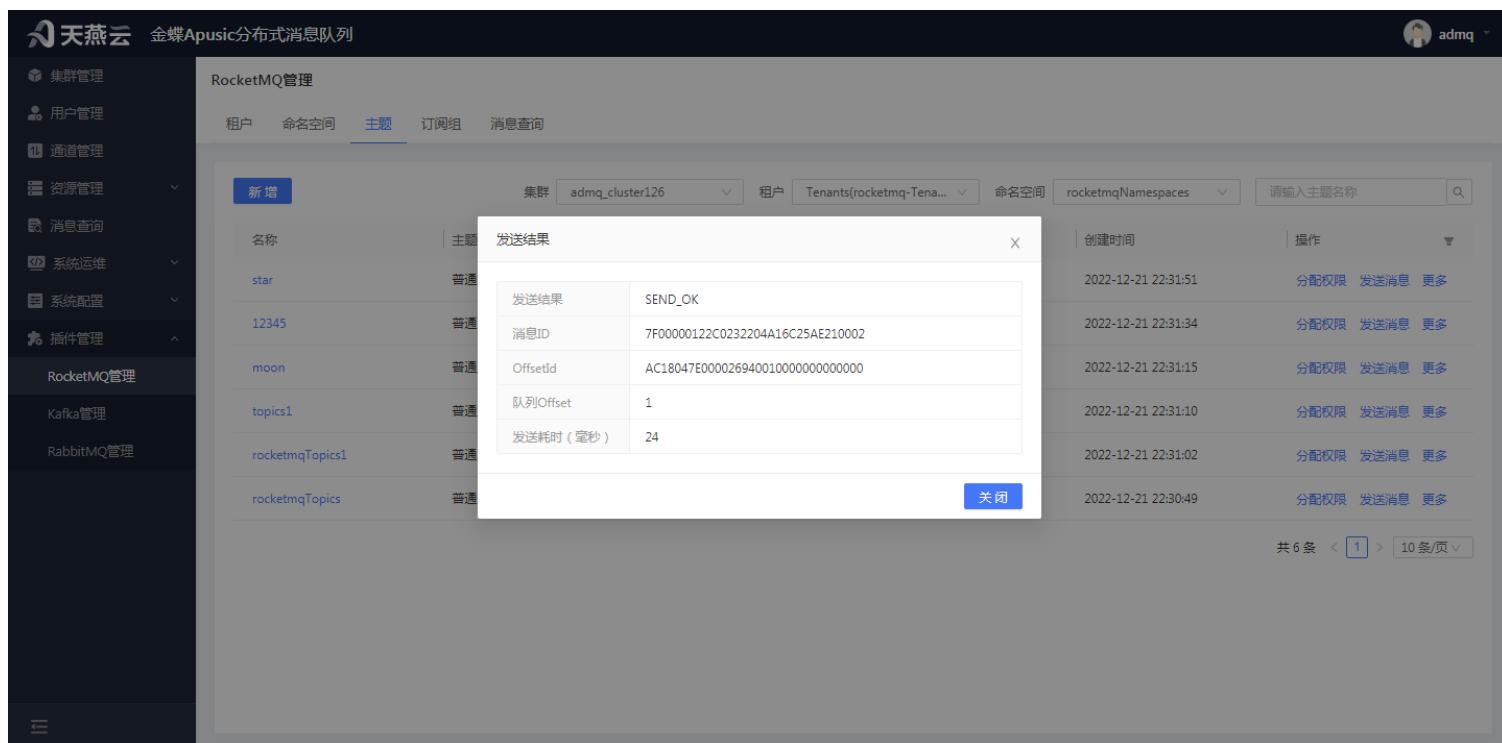
进入【插件管理】->【RocketMQ管理】，选择"主题"页签，选择要发送消息的主题名称后的【发送消息】按钮，可发送消息，发送的消息，可在消息查询中查到。



- 主题：发送消息的主题名称，根据选择的主题名称自动带出，不可修改
- 消息Tag：消息的Tag，多个Tag用英文逗号隔开，可自定义
- 消息Key：消息的key，多个Key用英文逗号隔开，可自定义
- 消息内容：发送的消息内容，自定义

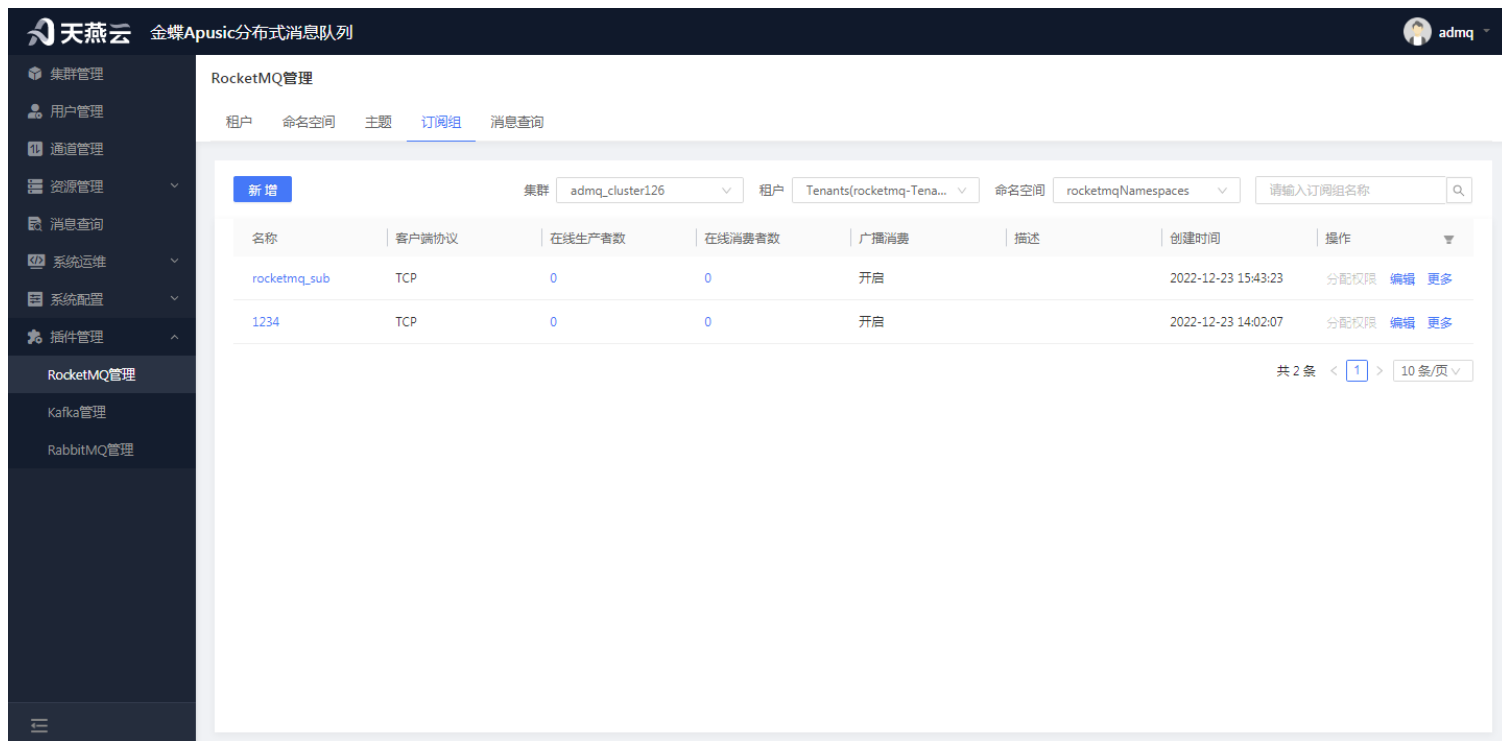
- 开启消息轨迹：开启、不开启，单选题，默认不开启

在新建界面点击【确定】按钮后，会出现成功发送消息的弹窗，如下：



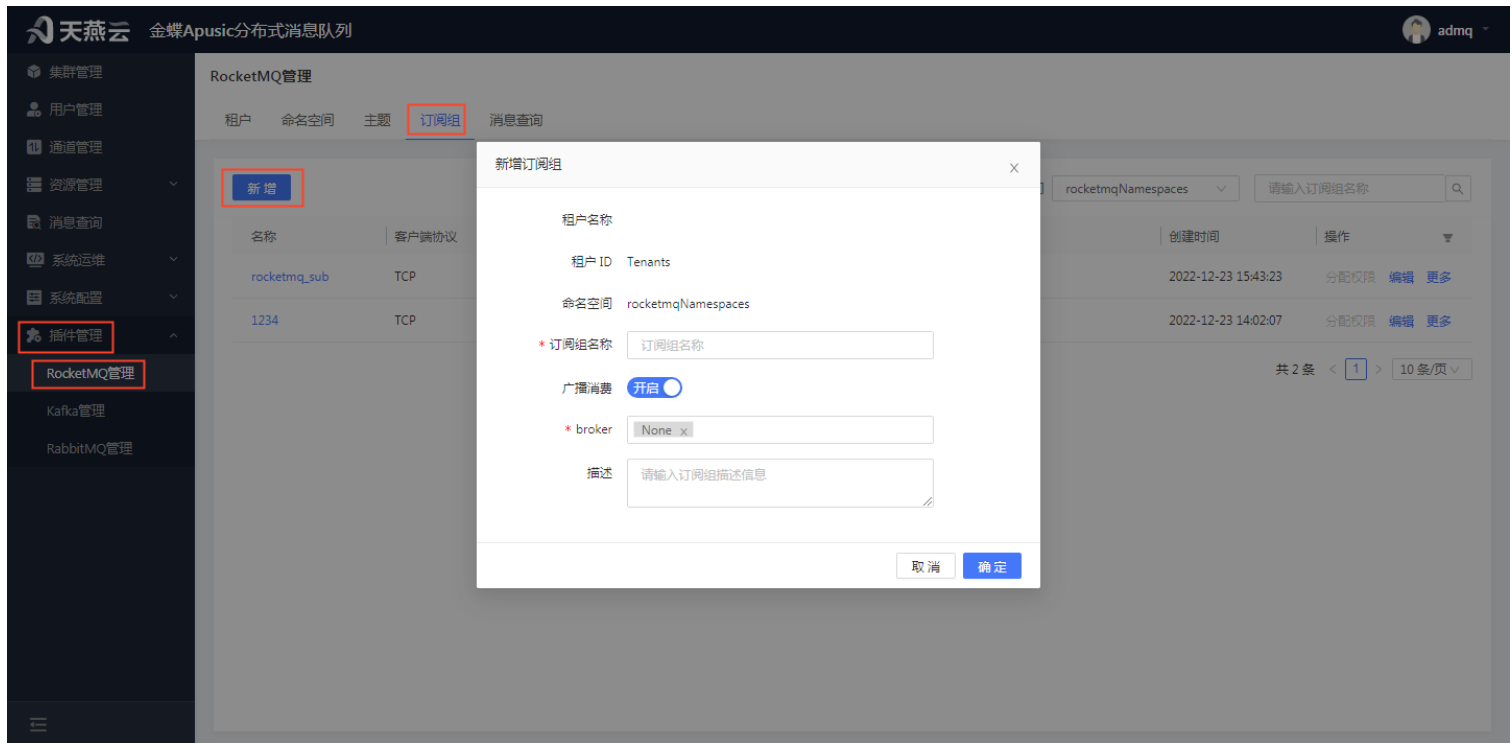
5.10.1.4 订阅组

发送消息和接收消息，都需要绑定订阅组，订阅组就是作用于管理订阅组。订阅组用于新增订阅组、编辑订阅组、删除订阅组、查询订阅组、重置消费位点（回溯消息）



5.10.1.4.1 新增订阅组

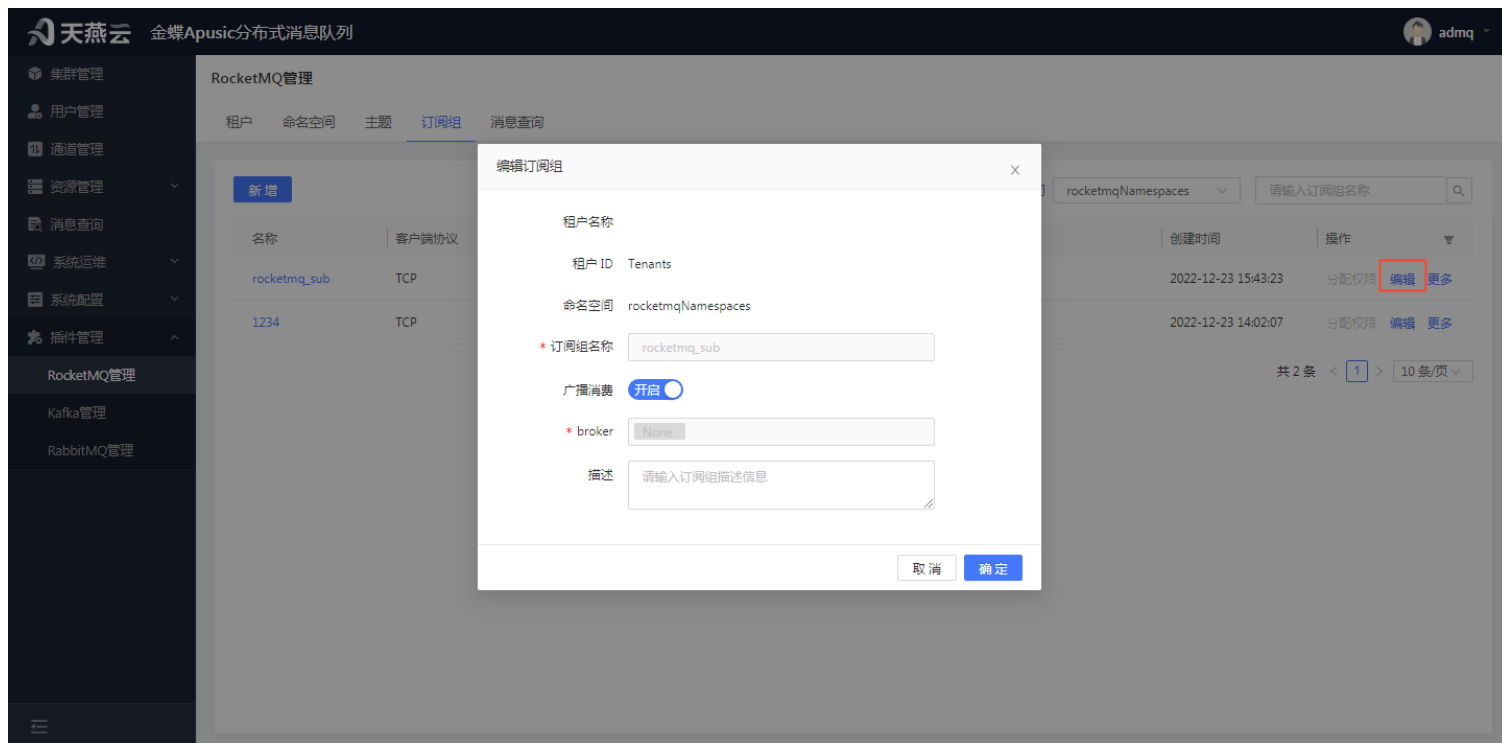
进入【插件管理】->【RocketMQ管理】，选择订阅组页签，点击左上角【新增】按钮，可新增订阅组信息。注意：新增订阅组前，需满足集群服务可用。



- 租户名称：订阅组所属租户的名称，根据选择的租户自动带出，不可修改
- 租户ID：订阅组所属租户的租户ID，根据选择的租户自动带出，不可修改
- 命名空间：订阅组所属命名空间名称，根据选择的命名空间自动带出，不可修改
- 订阅组名称：订阅组的名称，自定义，唯一标识
- 广播消费：消费模式是否为广播模式，开启、不开启，默认开启
- broker：所属broker，默认None
- 描述：订阅组的描述信息

5.10.1.4.2 编辑订阅组

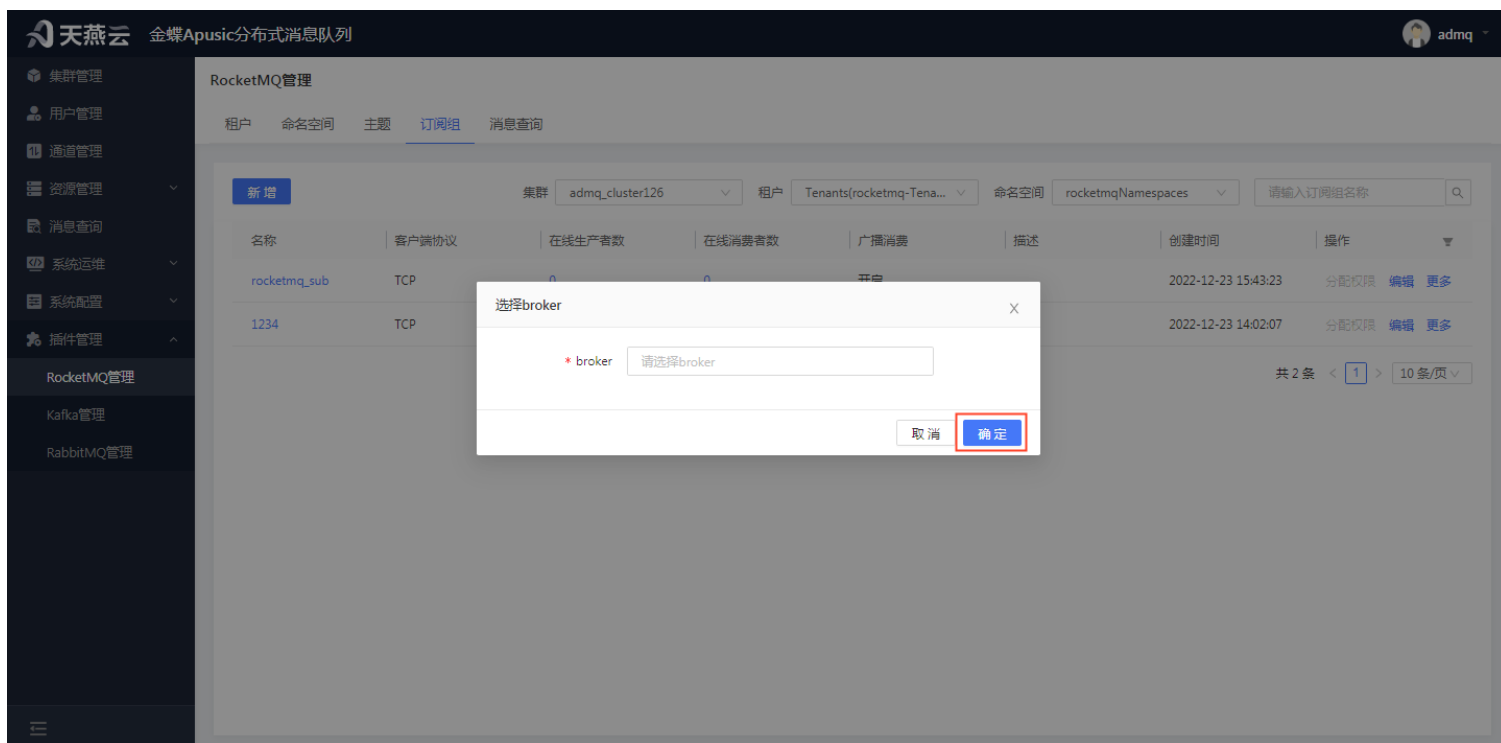
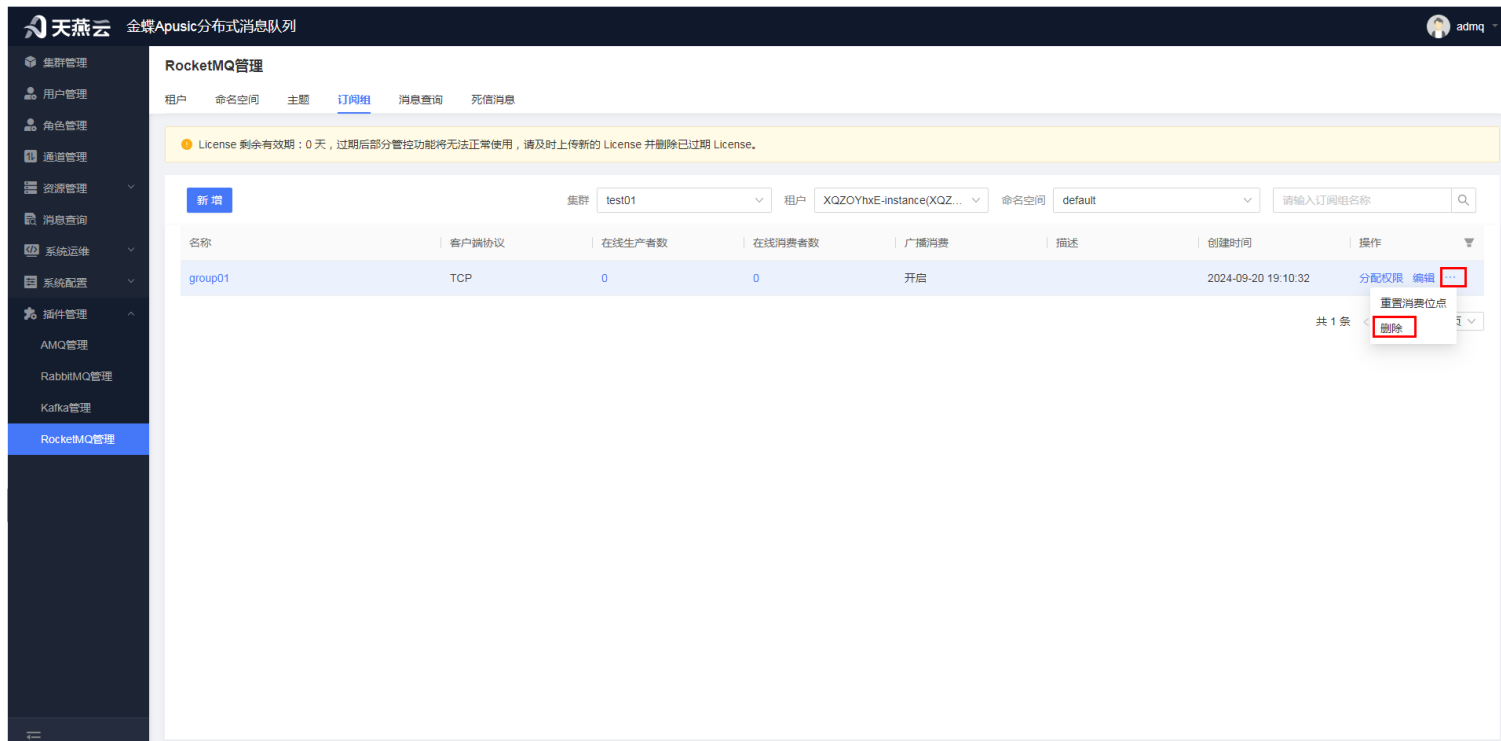
进入【插件管理】->【RocketMQ管理】，选择订阅组页签，选择要编辑的订阅组名称后的【编辑】按钮，可编辑订阅组信息。注意：编辑订阅组前，需满足集群服务可用。



- 租户名称：订阅组所属租户的名称，不可修改
- 租户ID：订阅组所属租户的租户ID，不可修改
- 命名空间：订阅组所属命名空间名称，不可修改
- 订阅组名称：订阅组的名称，置灰显示，不可修改
- 广播消费：默认开启，可修改
- broker：所属broker，置灰显示，不可修改
- 描述：订阅组的描述信息，可修改

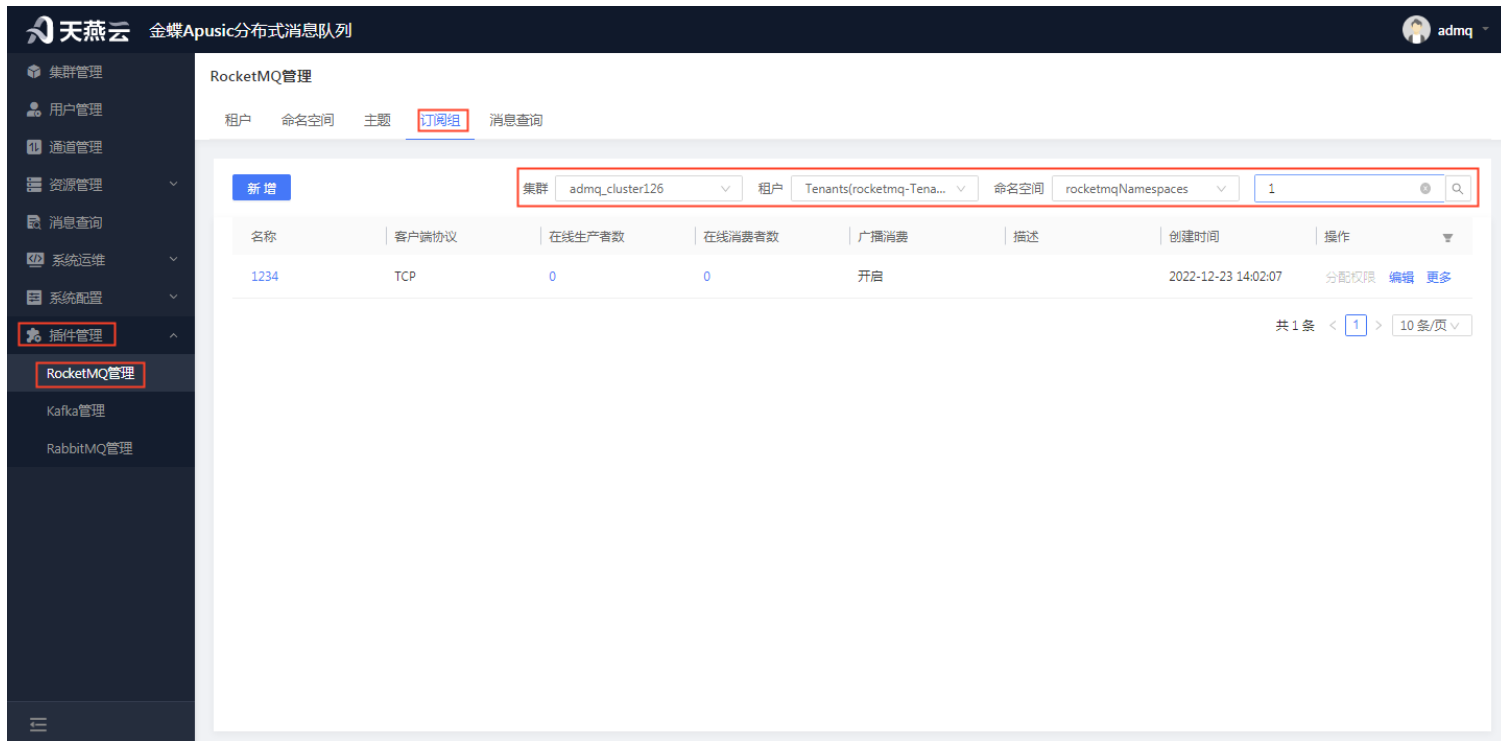
5.10.1.4.3 删除订阅组

进入【插件管理】->【RocketMQ管理】，选择订阅组页签，选择【...】->【删除】按钮，在弹出的弹窗内选择broker后，点击【确定】按钮，可删除订阅组信息。注意：删除订阅组前，需满足集群服务可用。



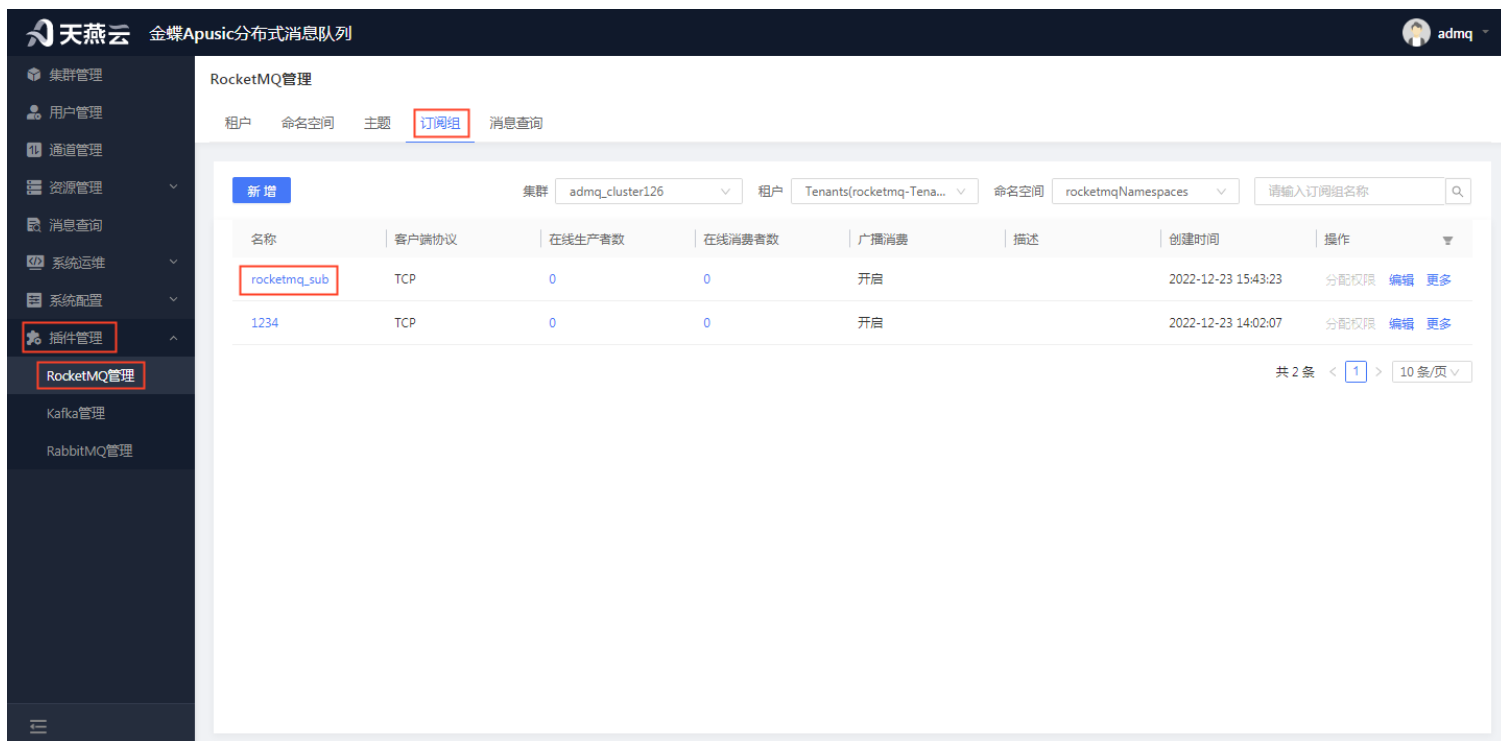
5.10.1.4.4 查询订阅组

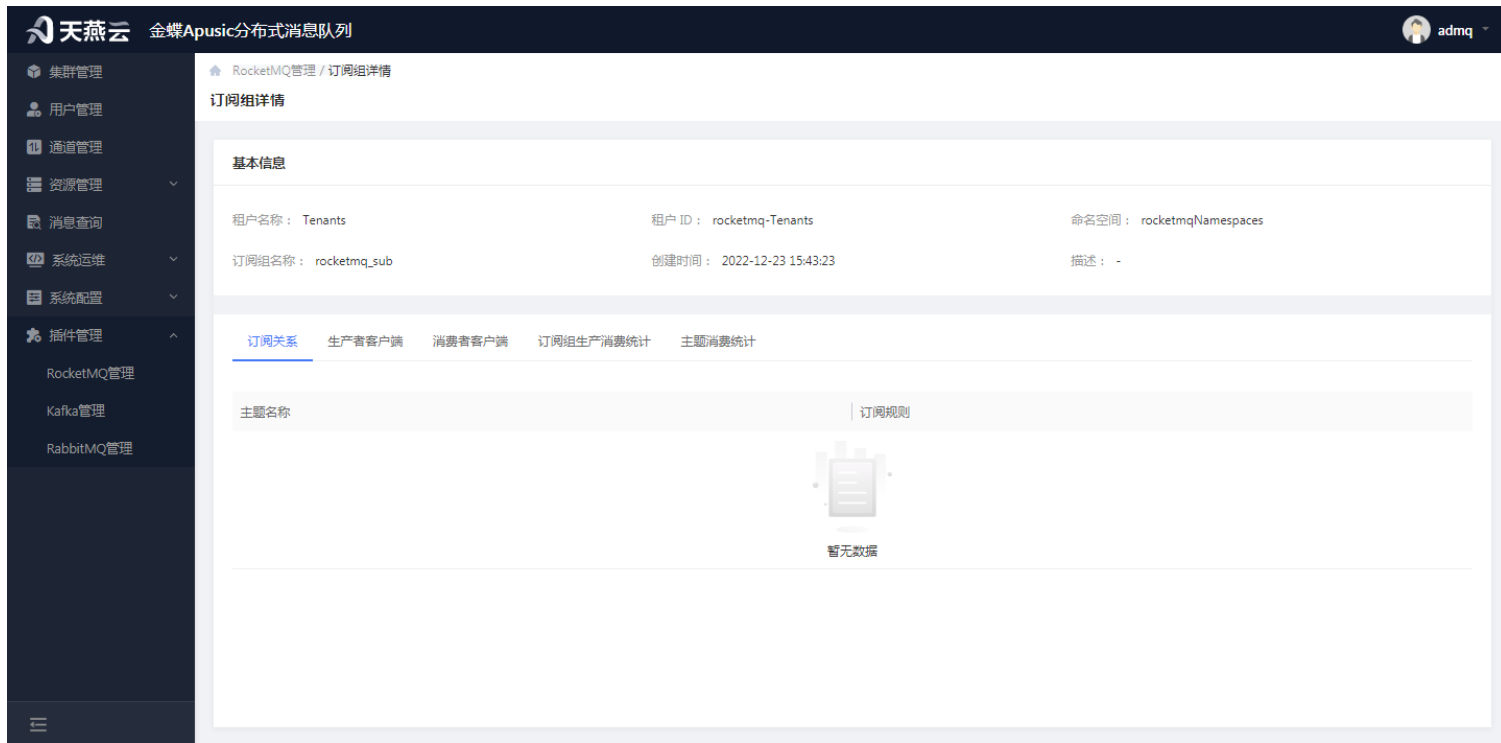
进入【插件管理】->【RocketMQ管理】，选择订阅组页签，右上角选择集群名称、实例名称、命名空间名称，精确或模糊输入订阅组名称，可查询订阅组信息。



5.10.1.4.5 查看订阅组详情

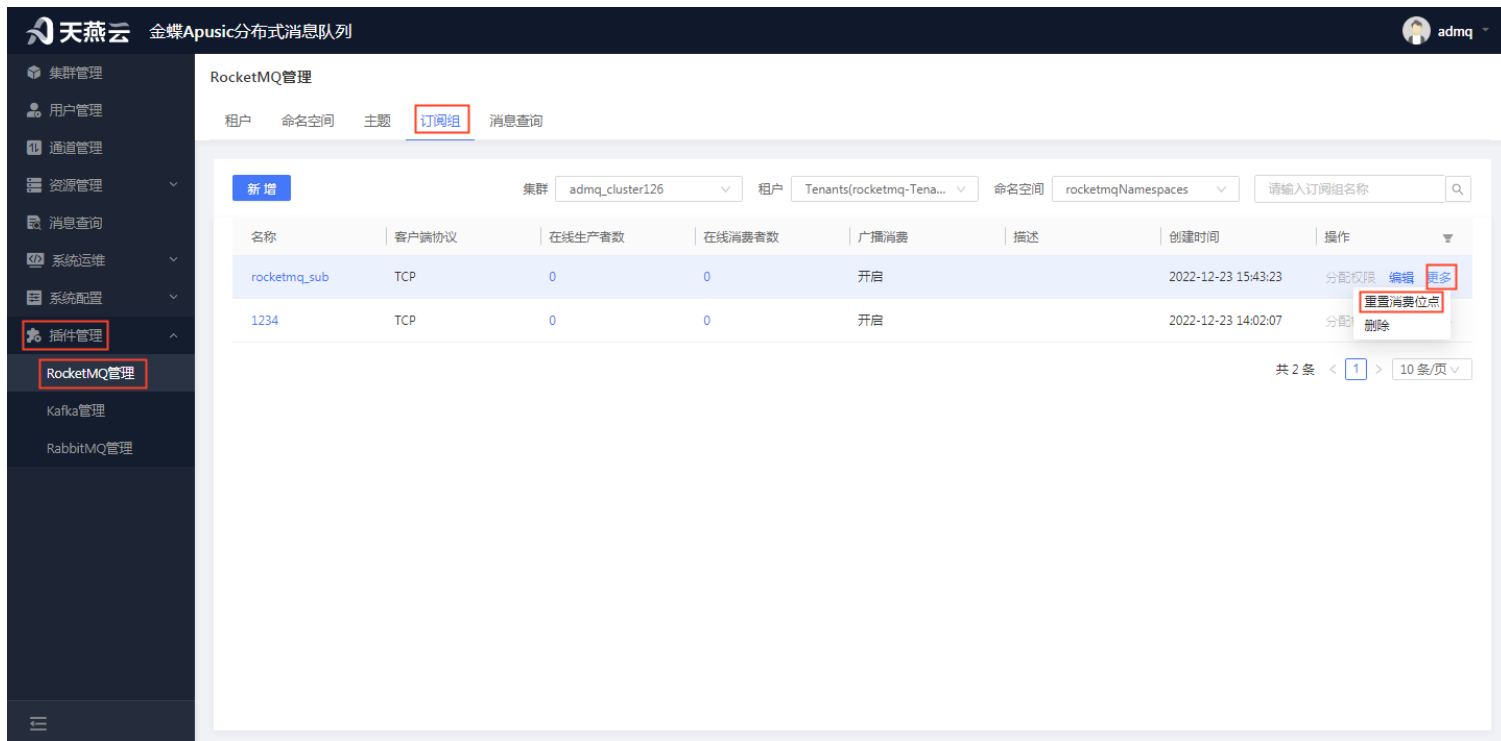
进入【插件管理】->【RocketMQ管理】，选择订阅组页签，选择需要查看详细信息的订阅组，点击订阅组名称，可查看订阅组详细信息，如基本信息，订阅关系、生产者客户端、消费者客户端、订阅组生产消费统计、主题消费统计。

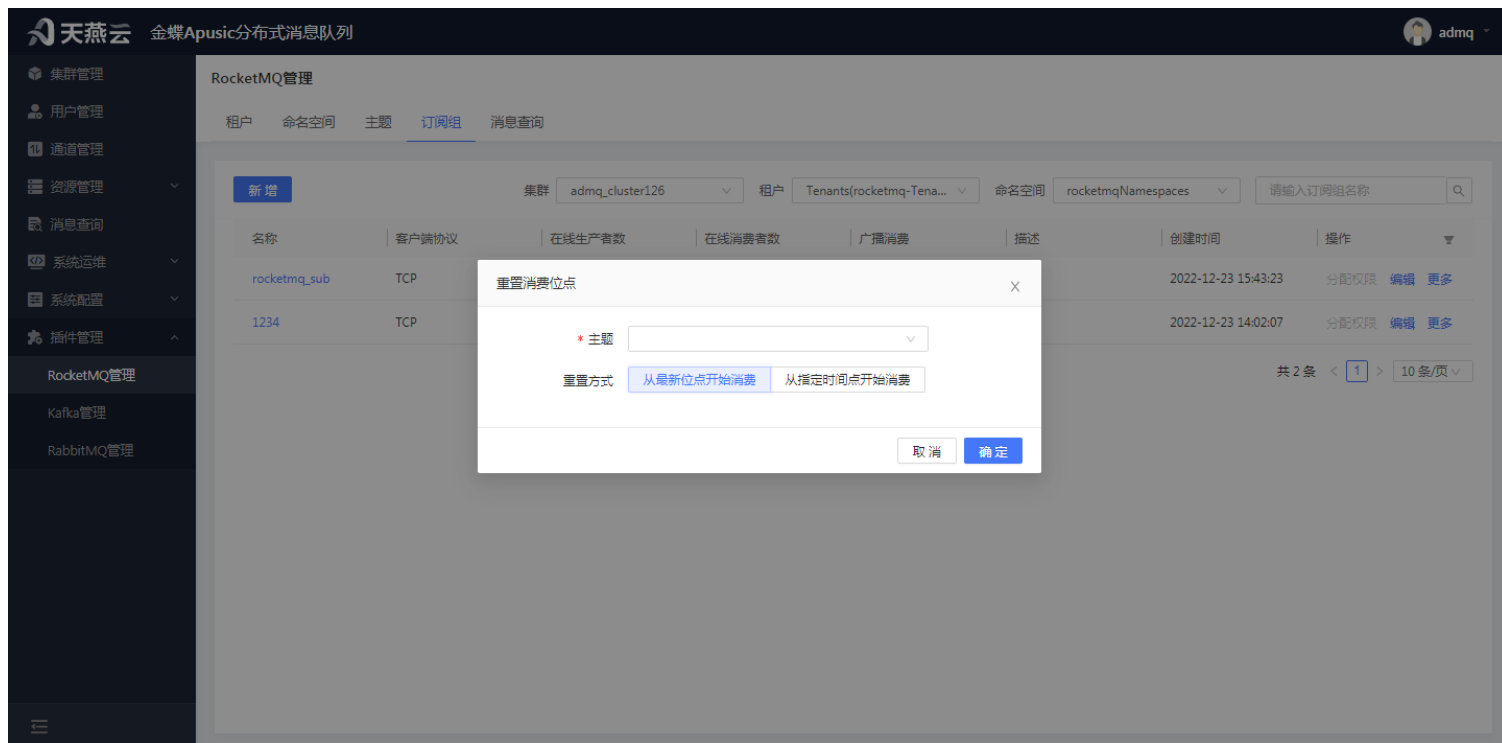




5.10.1.4.6 重置消费位点

进入【插件管理】->【RocketMQ管理】，选择订阅组页签，选择需要重置消费的订阅组，点击【更多】->【重置消费位点】按钮，可重置消费。

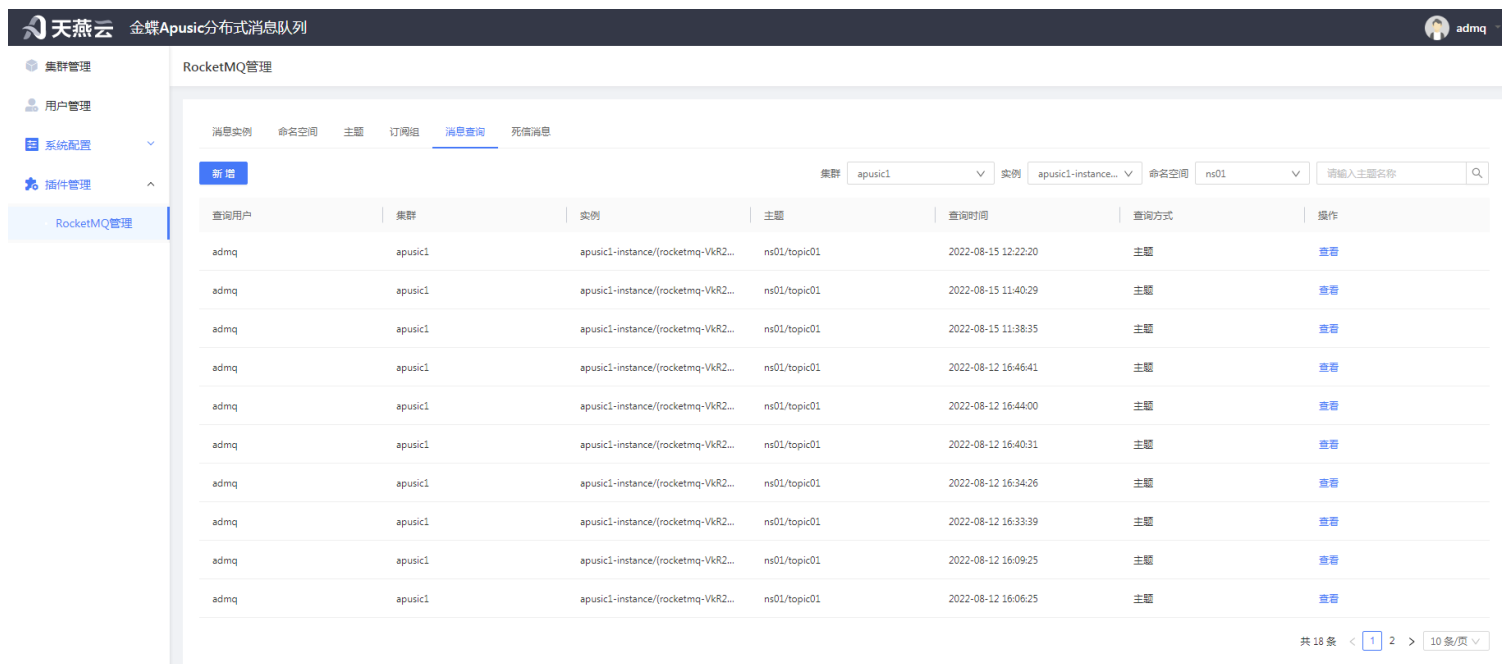




- 主题：重置消费的主题名，下拉选择（主题需发送和接收消息后才能被选择）
- 重置方式：重置消费的方式，单选按钮，从最新开始位点开始消费、从指定时间点开始消费

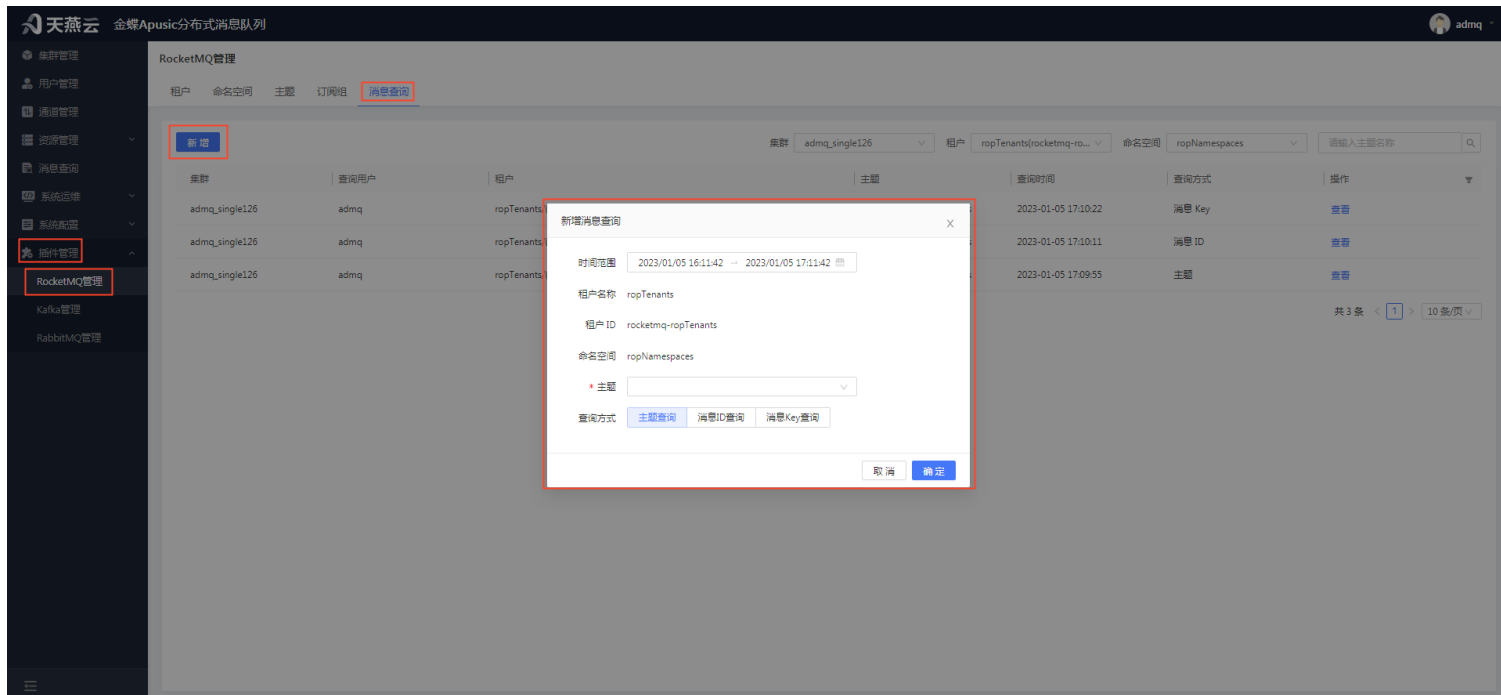
5.10.1.5 消息查询

消息查询用于查询消息具体信息，如消息生产信息、消息消费信息等。消息查询用于新增消息查询、查询已创建的消息查询、查看查询消息详情信息。



5.10.1.5.1 新增消息查询

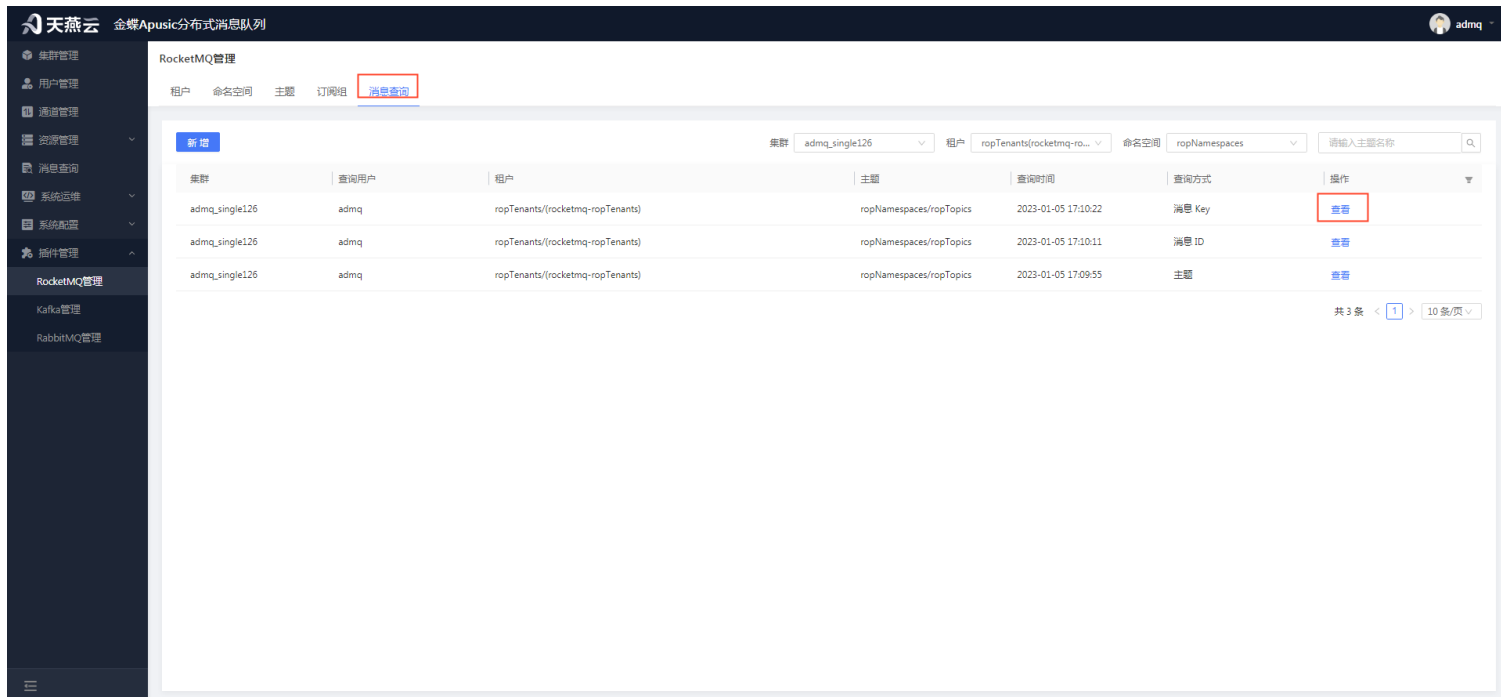
进入【插件管理】->【RocketMQ管理】，选择消息查询页签，点击左上角【新增】按钮，可新增消息查询信息。注意：新增消息查询前，需满足集群服务可用。



- 时间范围：要查询的消息时间范围，默认为1小时，可手动选择
- 租户名称：租户的名称，根据选择的租户自动带出，不可修改
- 租户ID：租户的ID，根据选择的租户自动带出，不可修改
- 命名空间：命名空间名称，根据选择的命名空间自动带出，不可修改
- 主题：主题名称，根据选择的命名空间下拉选择
- 查询方式：消息查询的查询方式（根据查询方式进行消息查询过滤），主题查询、消息ID查询、消息Key查询

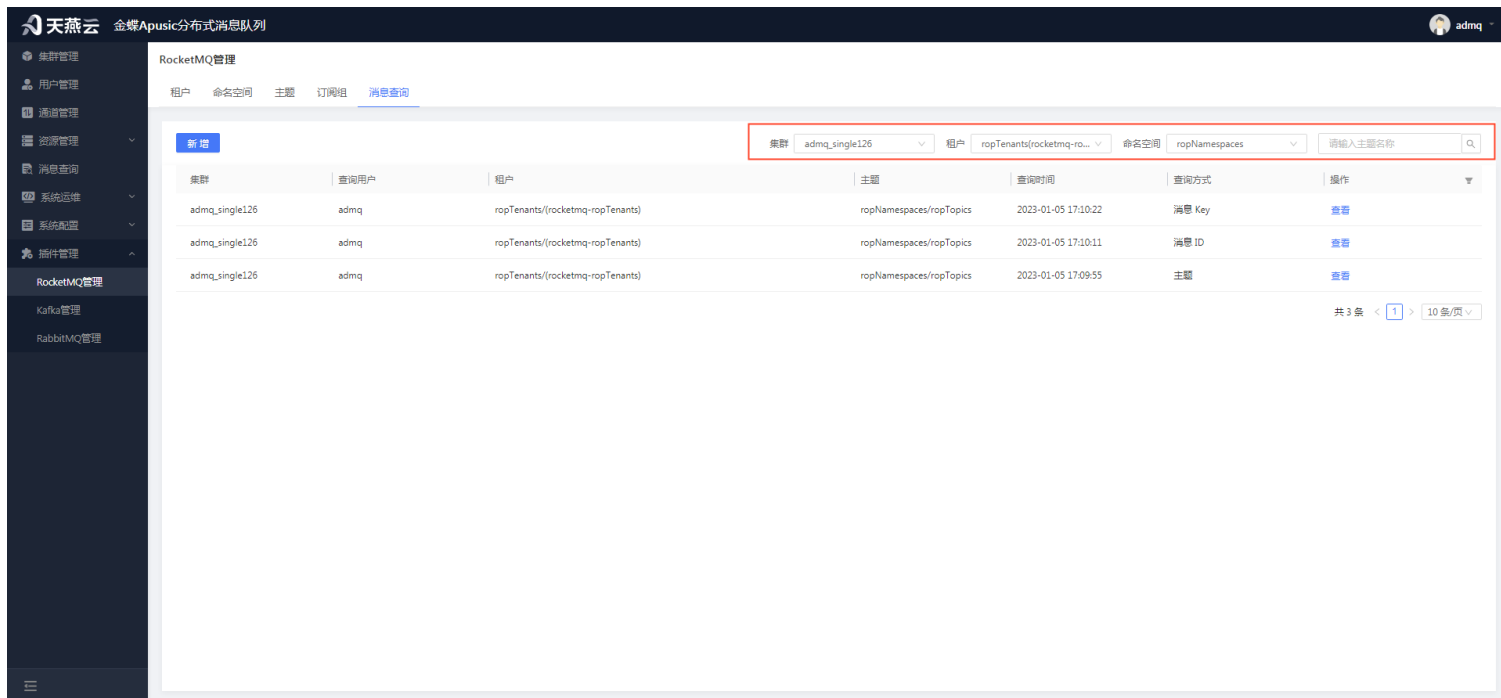
5.10.1.5.2 查看消息详情信息

进入【插件管理】->【RocketMQ管理】，选择消息查询页签，选择新增的消息查询，点击【查看】按钮，可查看查询结果，选择某条消息ID，点击【查看】操作，可查看具体某条消息的详细信息（查询条件信息、消息生产信息、消息消费信息）。



5.10.1.5.3 查询已创建的消息查询

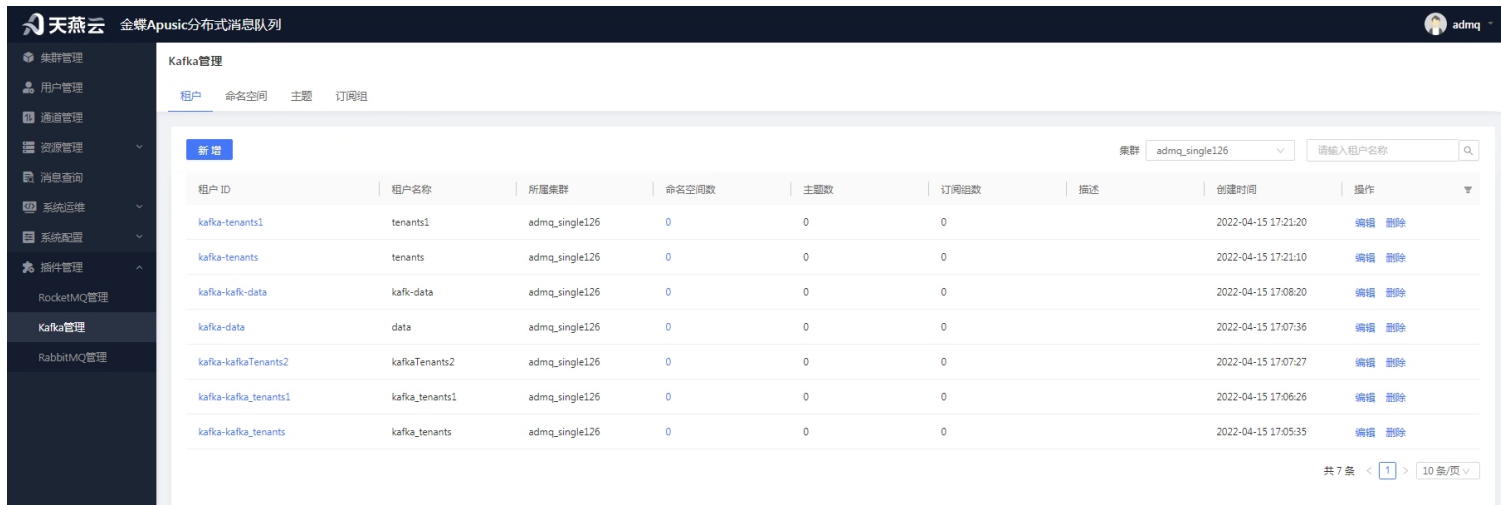
进入【插件管理】->【RocketMQ管理】，选择消息查询页签，选择集群名称、租户名称、命名空间、输入主题名称，可查询到满足条件的消息查询。



5.10.2 Kafka管理

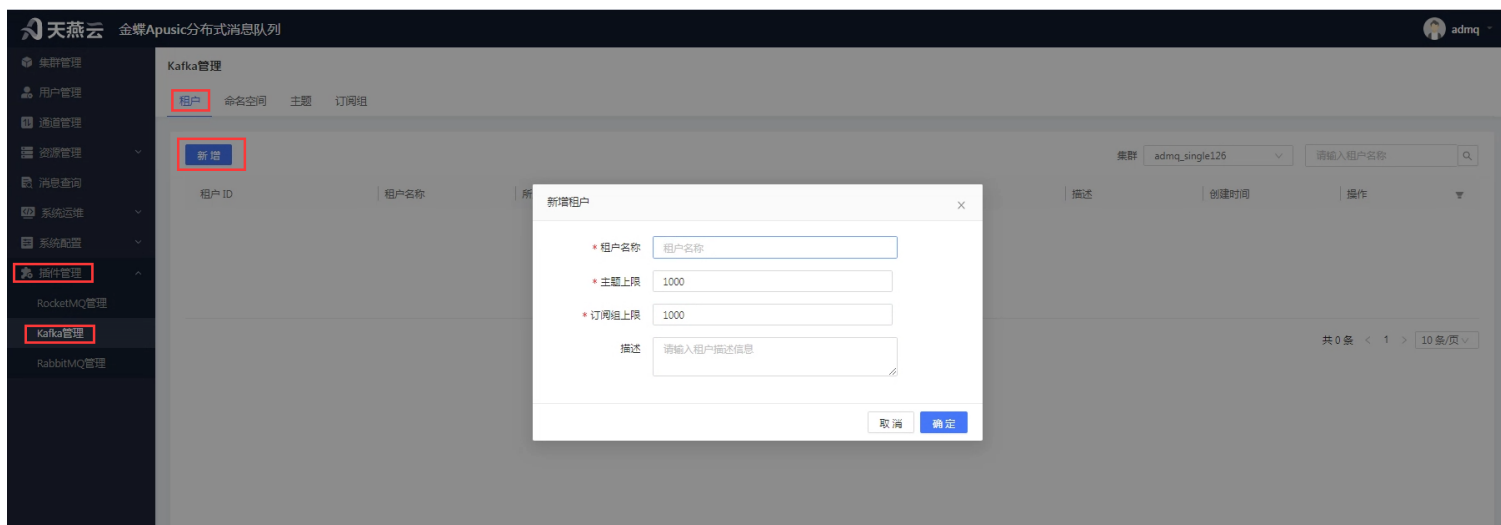
5.10.2.1 租户

租户是管理存储配额、隔离策略的基本单元，使用租户的目的是为了在多用户环境下，使用同一套程序，且保证用户间数据隔离。租户用于新增租户、编辑租户、删除租户、查询租户。



5.10.2.1.1 新增租户

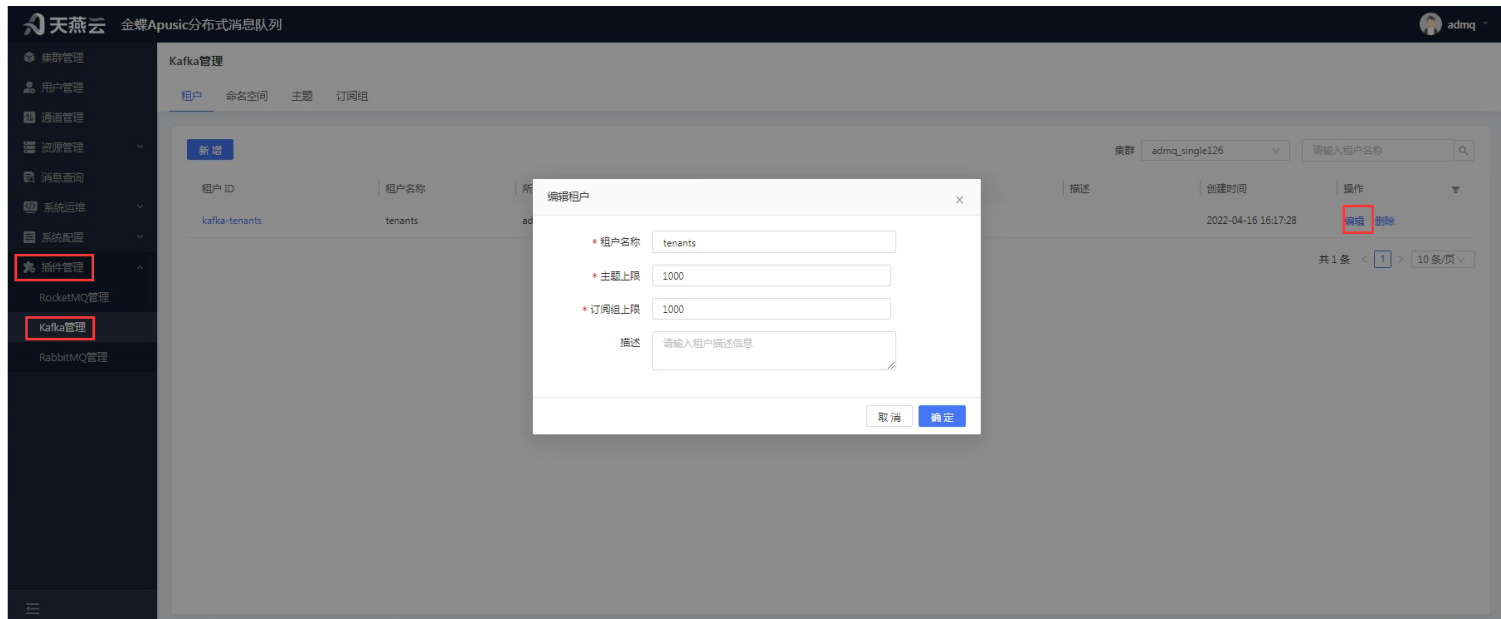
进入【插件管理】->【Kafka管理】，选择“租户”页签，点击左上角【新增】按钮，可新增租户。注意：新增租户前，需满足集群服务可用。



- 租户名称：租户的名称，自定义，唯一标识
- 主题上限：租户下可新增的主题数量，默认为1000，可更改
- 订阅组上限：租户下可新增的订阅组数量，默认为1000，可更改
- 描述：租户描述信息
- 注意：租户名称只是为了方便记，真正的租户是租户ID，客户端使用的是租户ID。

5.10.2.1.2 编辑租户

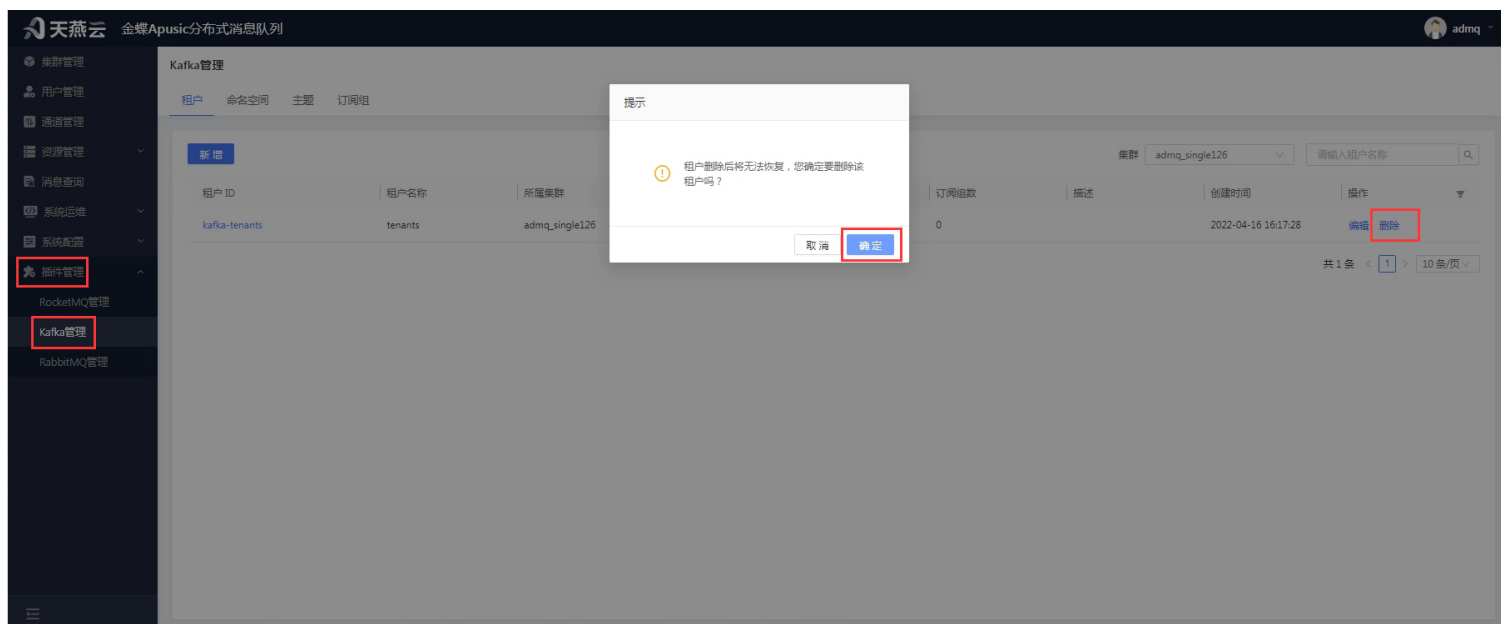
进入【插件管理】->【Kafka管理】，选择“租户”页签，选择租户列表中某一个租户名称后的【编辑】按钮，可编辑租户信息。注意：编辑租户前，需满足集群服务可用。



- 租户名称：租户的名称，可修改
- 主题上限：租户下可新增的主题数量
- 订阅组上限：租户下可新增的订阅组数量
- 描述：租户描述信息

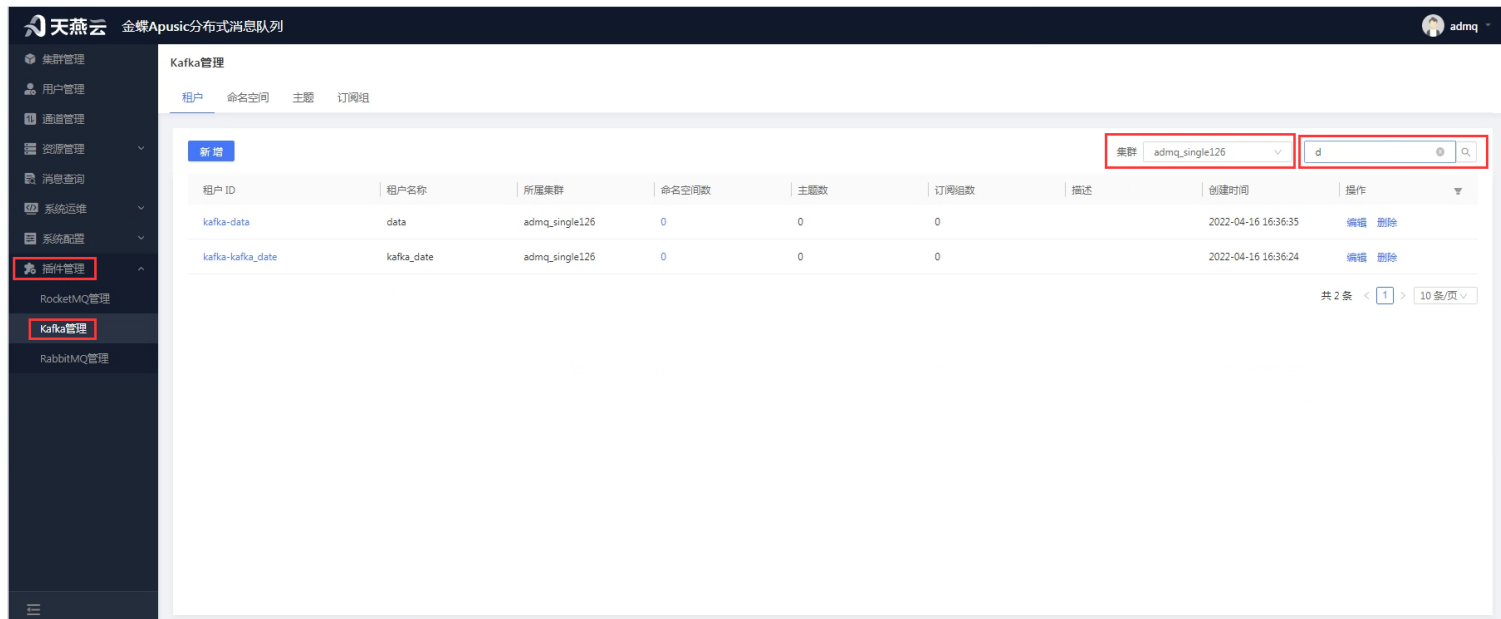
5.10.2.1.3 删除租户

进入【插件管理】->【Kafka管理】，选择“租户”页签，选择租户列表中某一个租户名称后的【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除租户信息。注意：删除租户前，需满足集群服务可用，且租户下无命名空间，否则不可删除。



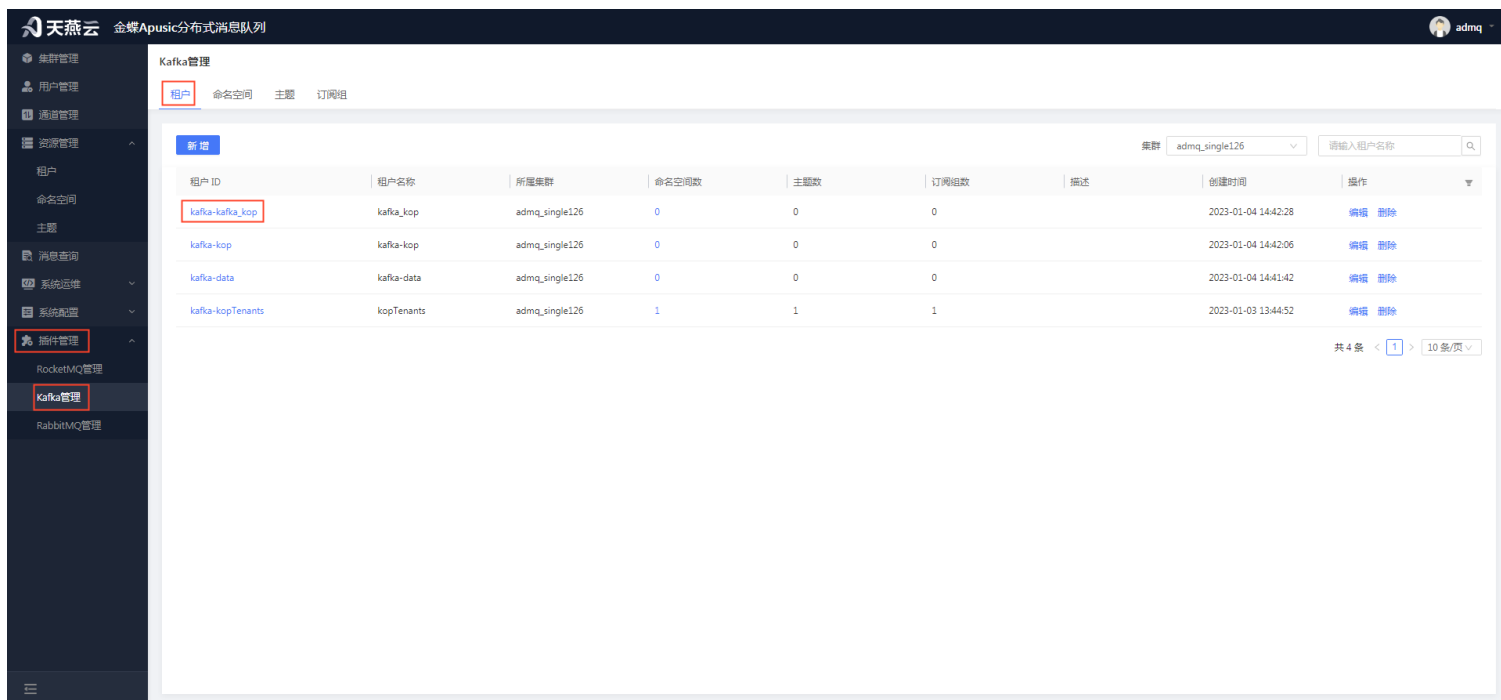
5.10.2.1.4 查询租户

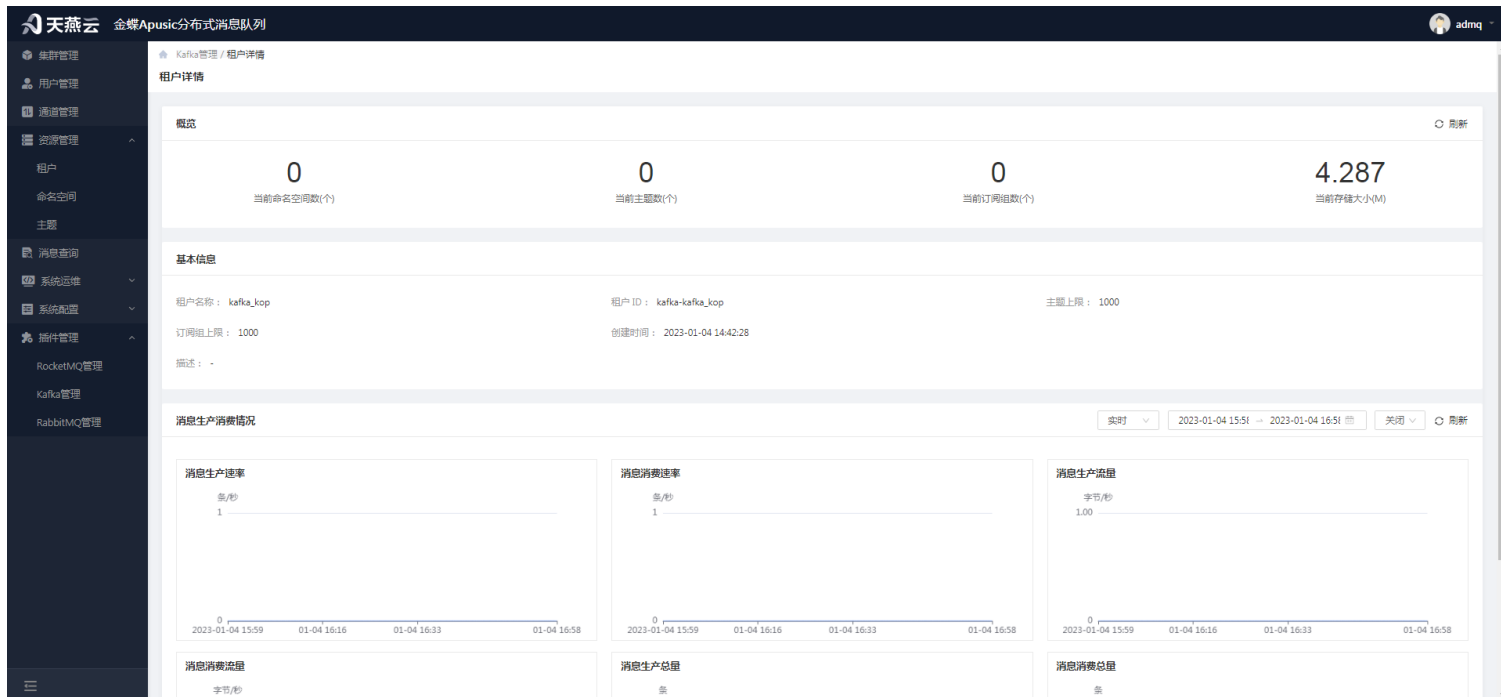
进入【插件管理】->【Kafka管理】，选择“租户”页签，右上角选择集群名称、输入租户名称、可查询租户信息。



5.10.2.1.5 查看租户详情

进入【插件管理】->【Kafka管理】，选择“租户”页签，择需要查看详细信息的租户，点击租户ID，可查看租户详细信息。





5.10.2.2 命名空间

命名空间作用于管理命名空间资源、命名空间用户权限。命名空间用于新增命名空间、编辑命名空间、删除命名空间、查询命名空间、新增命名空间权限、编辑命名空间权限、删除命名空间权限。

Kafka管理

租户 | 命名空间 | 主题 | 订阅组

新增

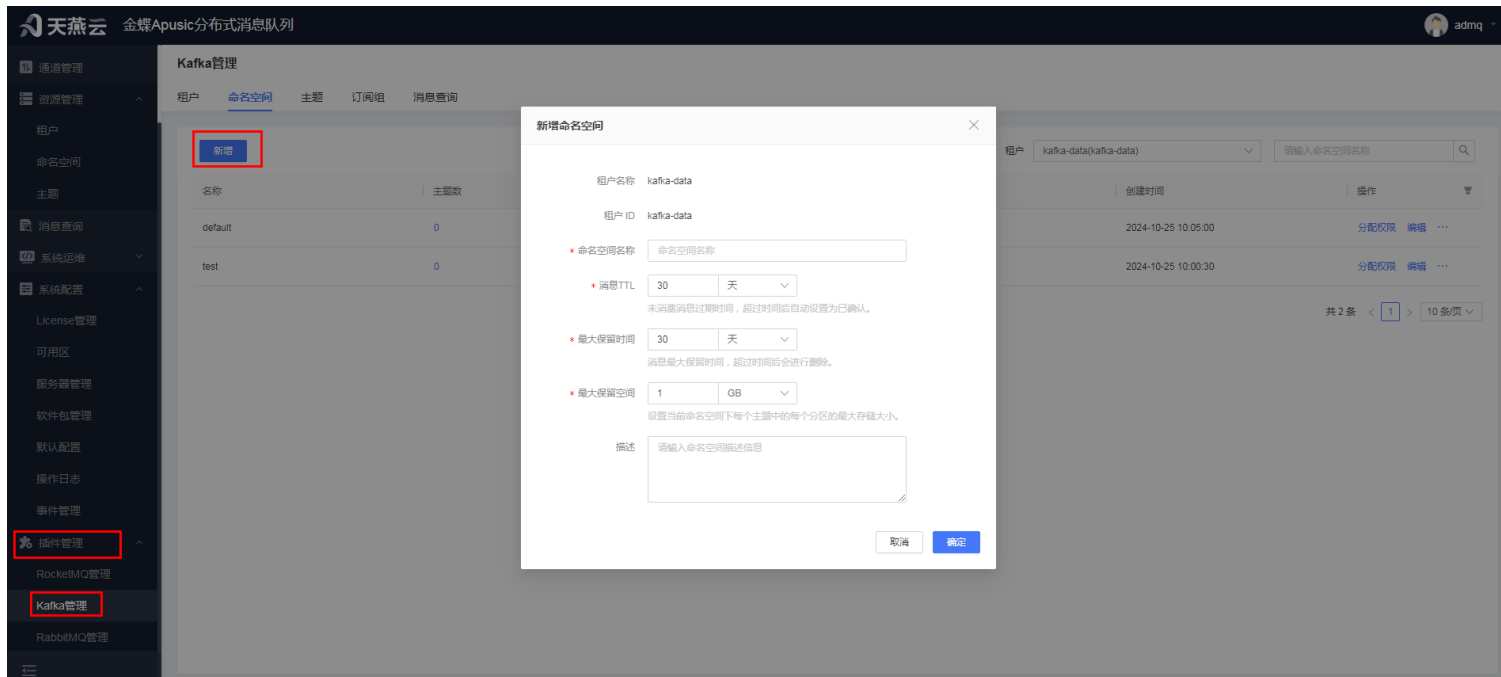
集群: admq_single126 | 租户: 1234(kafka-1234) | 请输入命名空间名称

名称	主题数	描述	创建时间	操作
34567	0		2022-04-16 16:47:01	分配权限 编辑 删除
lakes	0		2022-04-16 16:46:54	分配权限 编辑 删除
skysky	0		2022-04-16 16:46:38	分配权限 编辑 删除
mmon12345	0		2022-04-16 16:46:29	分配权限 编辑 删除
kafka_namespaces	0		2022-04-16 16:45:57	分配权限 编辑 删除

共 5 条 < 1 > 10 条/页

5.10.2.2.1 新增命名空间

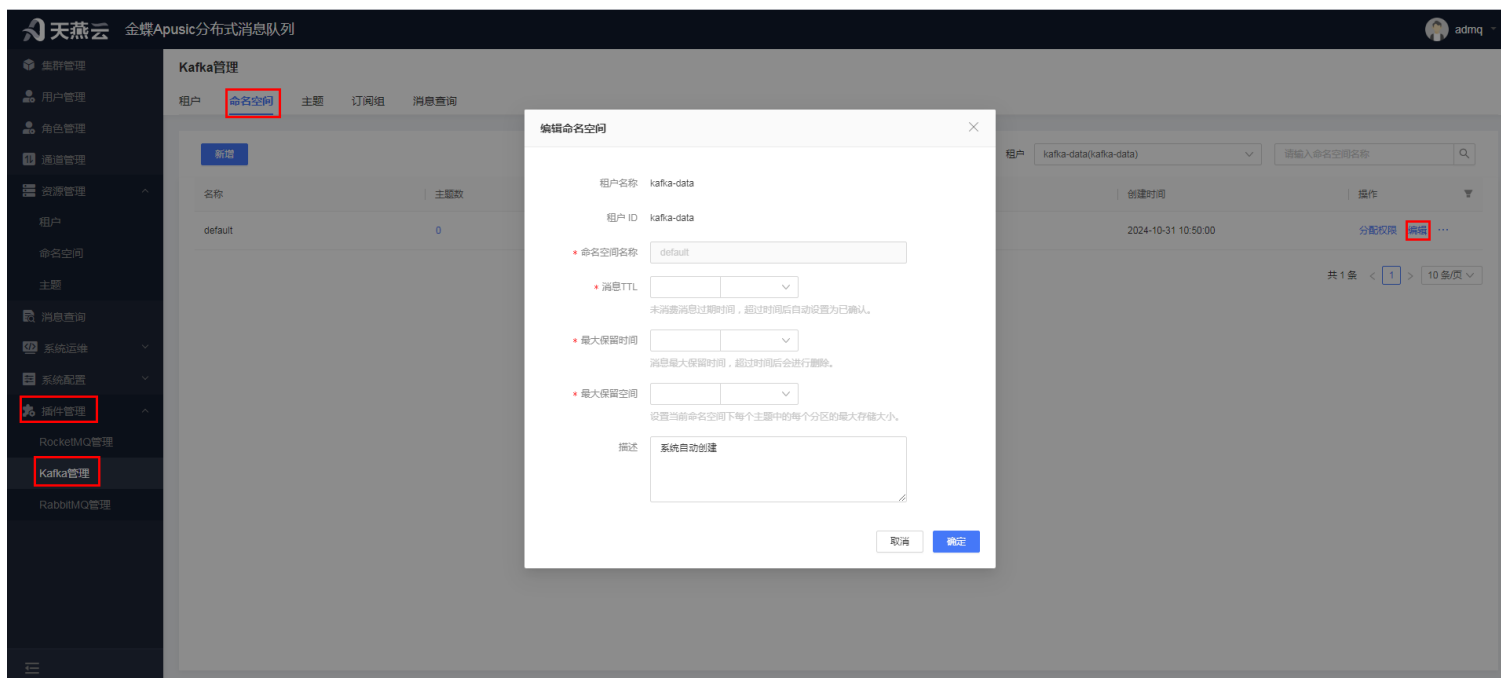
进入【插件管理】->【Kafka管理】，选择“命名空间”页签，点击左上角【新增】按钮，可新增命名空间信息。注意：新增命名空间前，需满足集群服务可用。



- 租户名称：命名空间所属的租户名称，根据选择的租户自动带出，不可修改
- 租户ID：命名空间所属租户的租户ID，根据选择的租户自动带出，不可修改
- 命名空间名称：命名空间的名称，自定义，唯一标识
- 消息TTL：未消费消息过期时间，超过时间后自动设置为已确认。
- 最大保留时间：消息最大保留时间，超过时间后会进行删除。
- 最大保留空间：设置当前命名空间下每个主题中的每个分区的最大存储大小。
- 描述：命名空间的描述信息

5.10.2.2.2 编辑命名空间

进入【插件管理】->【Kafka管理】，选择“命名空间”页签，选择要编辑的命名空间名称后的【编辑】按钮，可编辑命名空间信息。注意：编辑命名空间前，需满足集群服务可用。

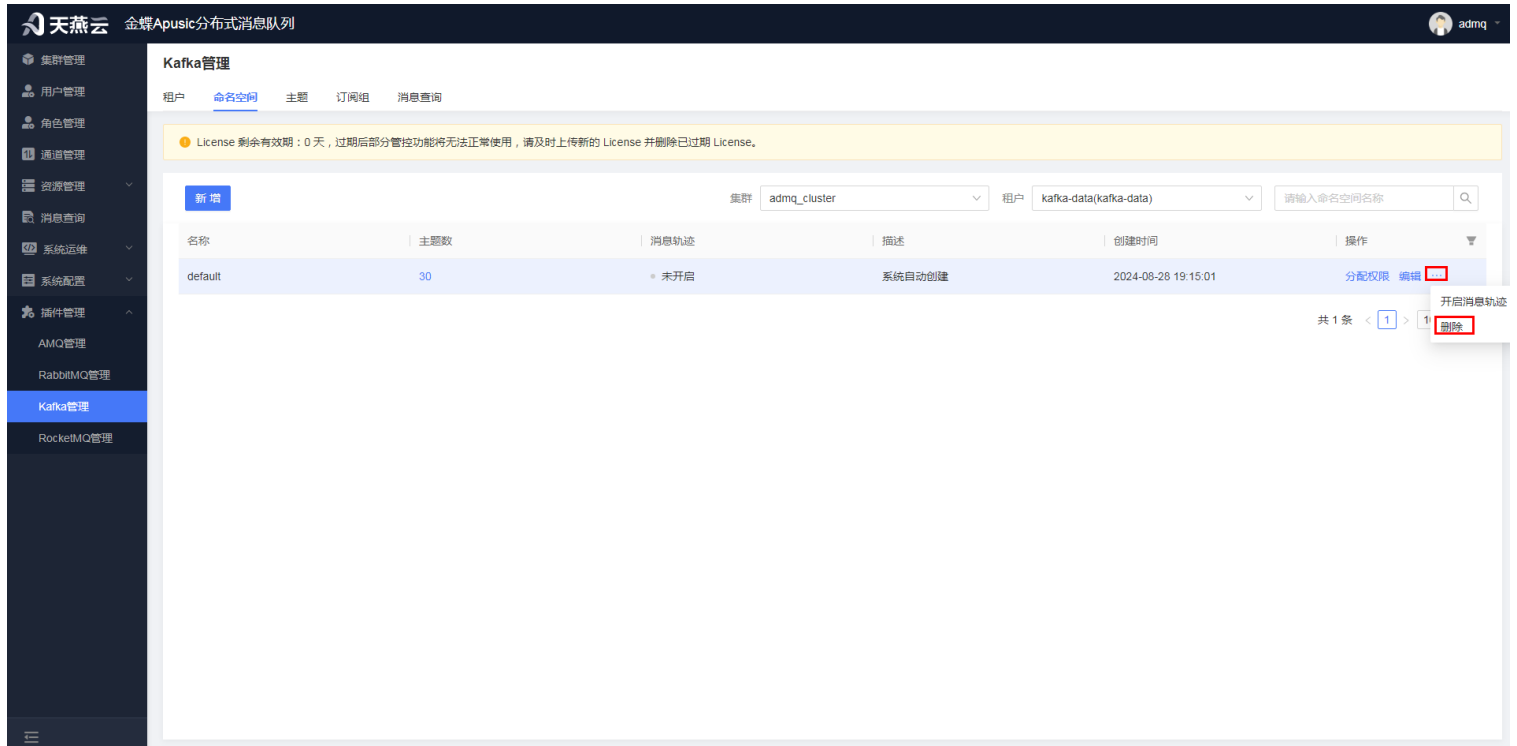


- 租户名称：命名空间所属的租户名称，根据选择的租户自动带出，不可修改

- 租户ID：命名空间所属租户的租户ID，根据选择的租户自动带出，不可修改
- 命名空间名称：命名空间的名称，置灰显示，不可修改
- 消息保留时间：消息的最大保留时间，超过则不保存消息，单位（分钟）、（小时）、（天），可修改
- 描述：命名空间的描述信息，可修改

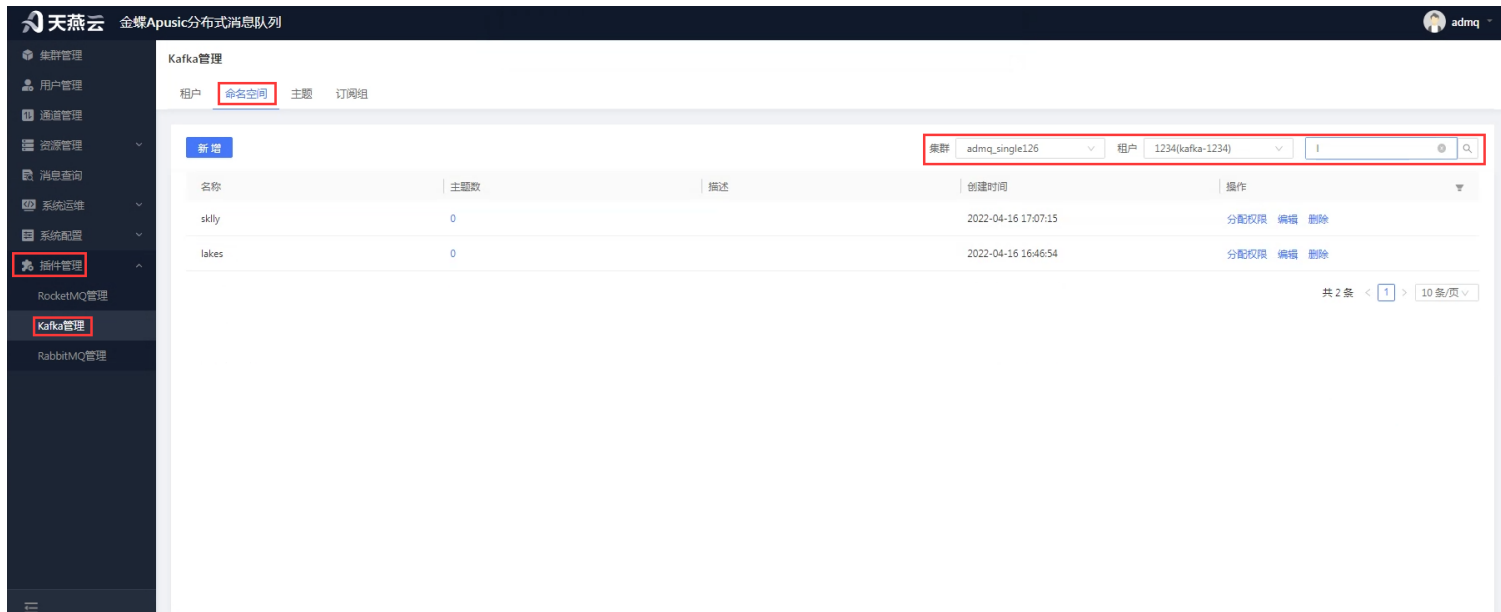
5.10.2.2.3 删除命名空间

进入【插件管理】->【Kafka管理】，选择“命名空间”页签，选择要删除的命名空间名称后的【...】-【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除命名空间信息。注意：删除命名空间前，需满足集群服务可用，且命名空间下无订阅组，删除命名空间后，命名空间下的主题信息也被删除。



5.10.2.2.4 查询命名空间

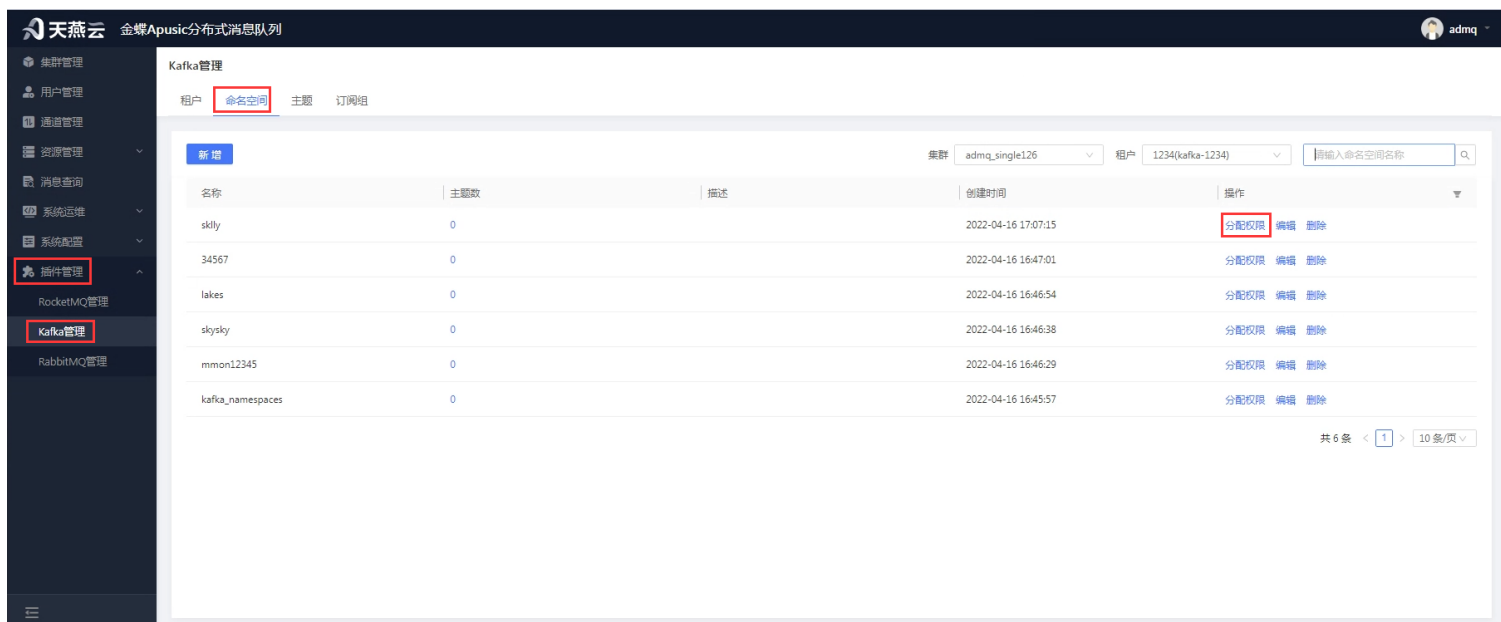
进入【插件管理】->【Kafka管理】，选择“命名空间”页签，右上角选择集群名称、租户名称、输入命名空间名称，可查询命名空间信息。（注意：包含的也有）



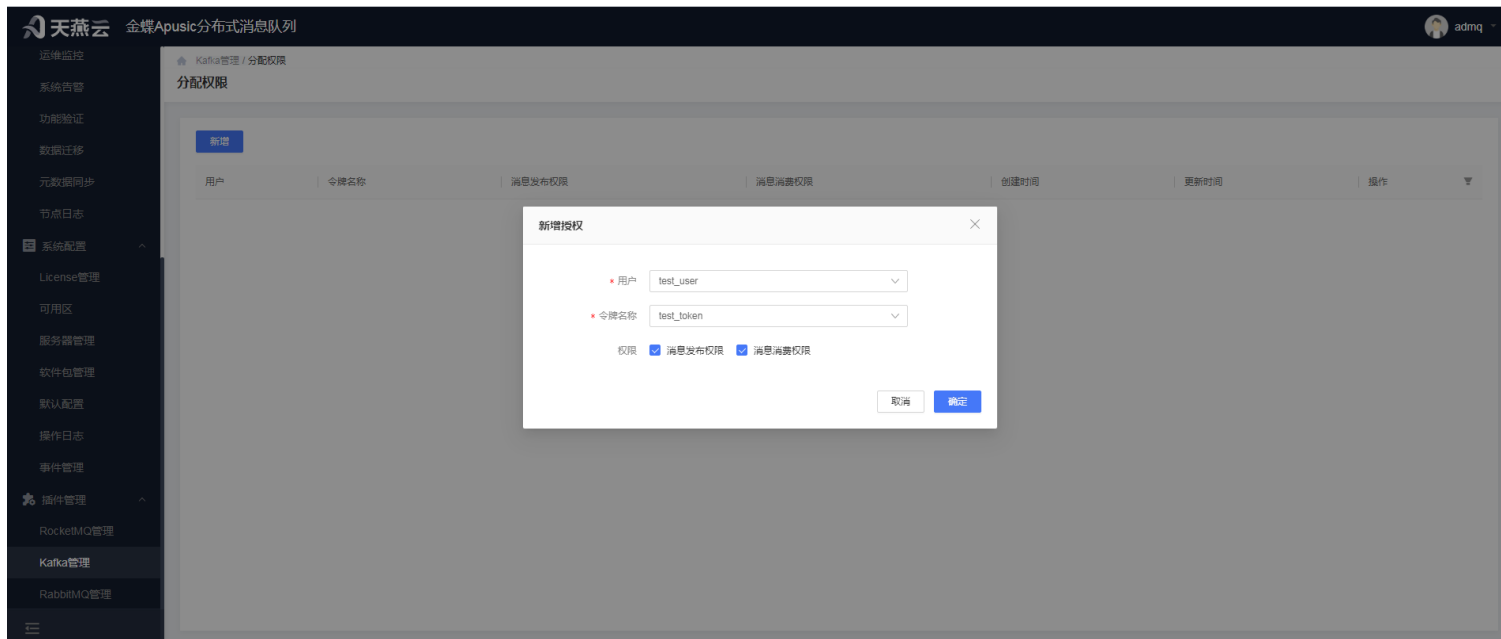
5.10.2.2.5 发布消费权限

5.10.2.2.5.1 新增权限

进入【插件管理】->【Kafka管理】，选择“命名空间”页签，选择要新增授权的命名空间名称后的【分配权限】按钮后，会进入到分配权限页面。



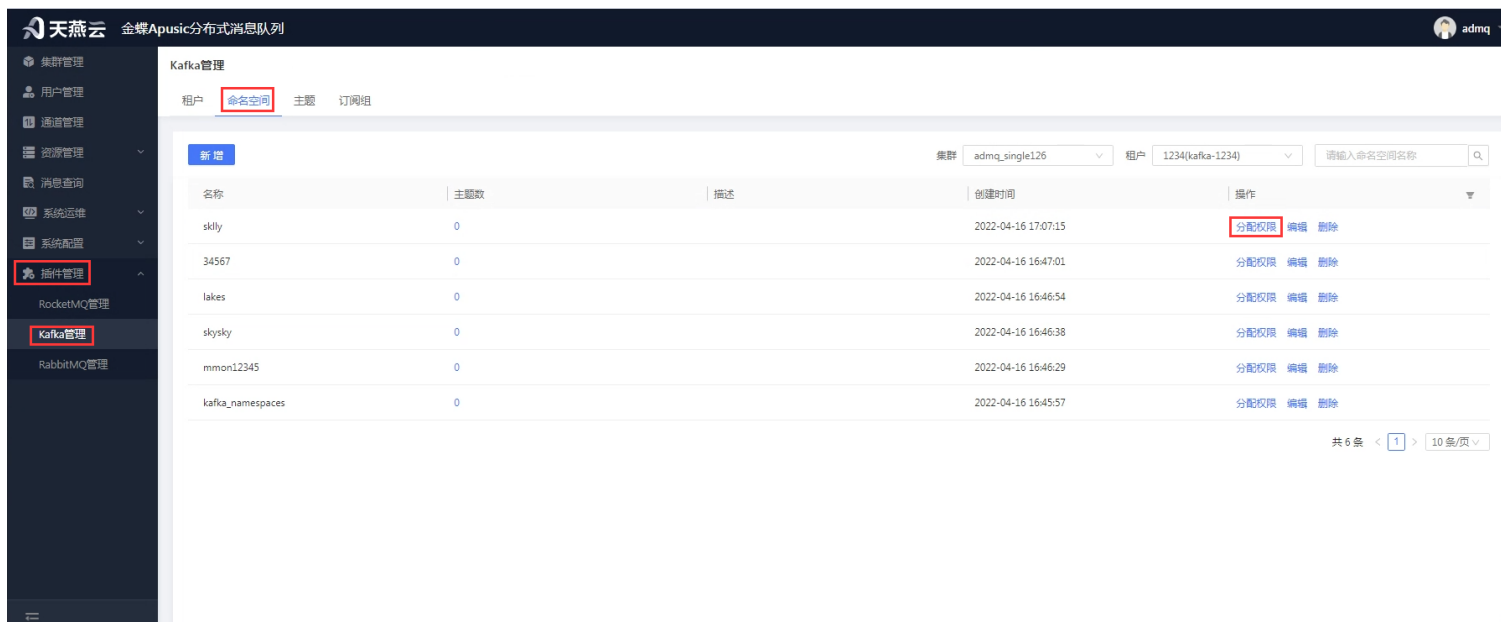
在分配权限页面，点击左上角的【新增】按钮，可新增命名空间下用户的权限信息。注意：新增权限前，需满足集群服务可用；授权主要是给普通用户提供授权，默认超级管理员拥有所有命名空间的消息发布权限和消息消费权限。



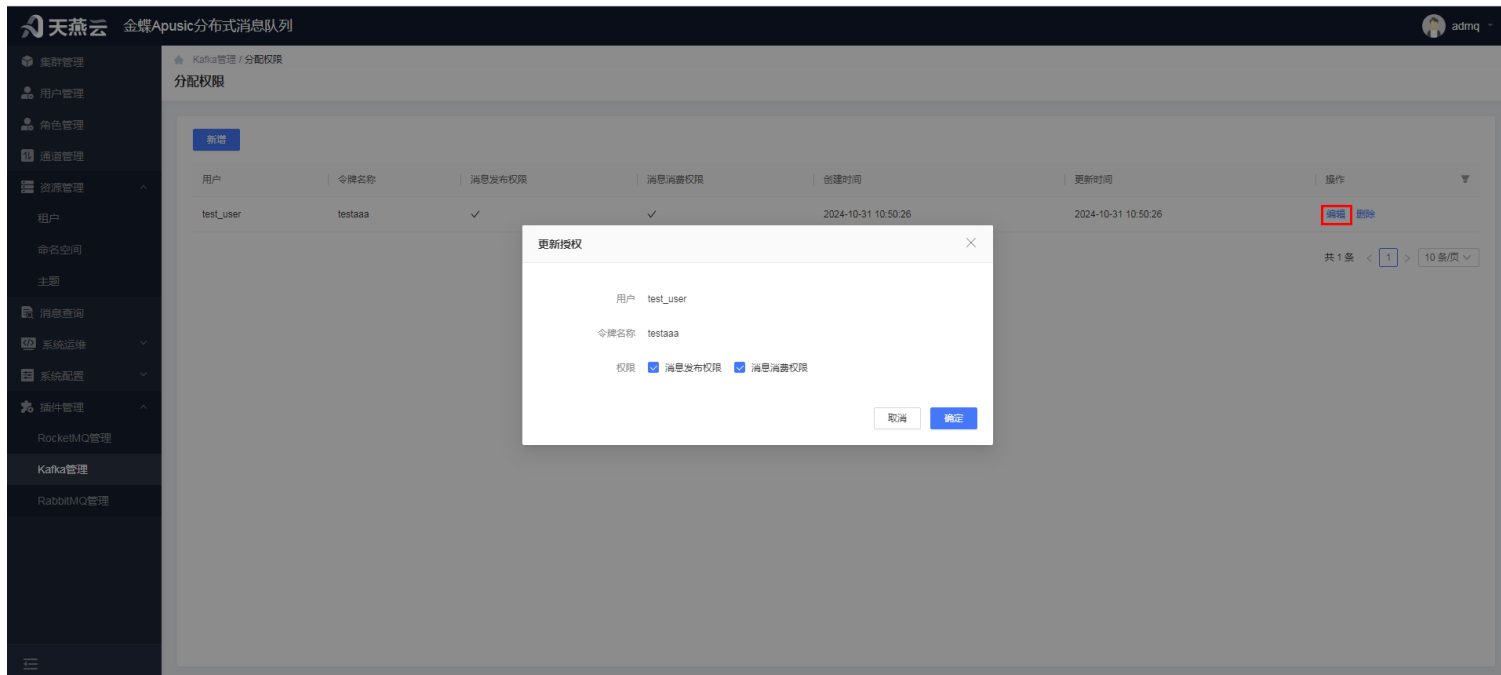
- 用户：下拉选择，获取的用户管理（4.1）的用户信息
- 令牌名称：下拉选择，获取选择用户的令牌。
- 权限：复选框，消息发布权限和消息消费权限

5.10.2.2.5.2 编辑权限

进入【插件管理】->【Kafka管理】，选择“命名空间”页签，选择要更新授权的命名空间名称后的【分配权限】按钮，跳转到分配权限页面。



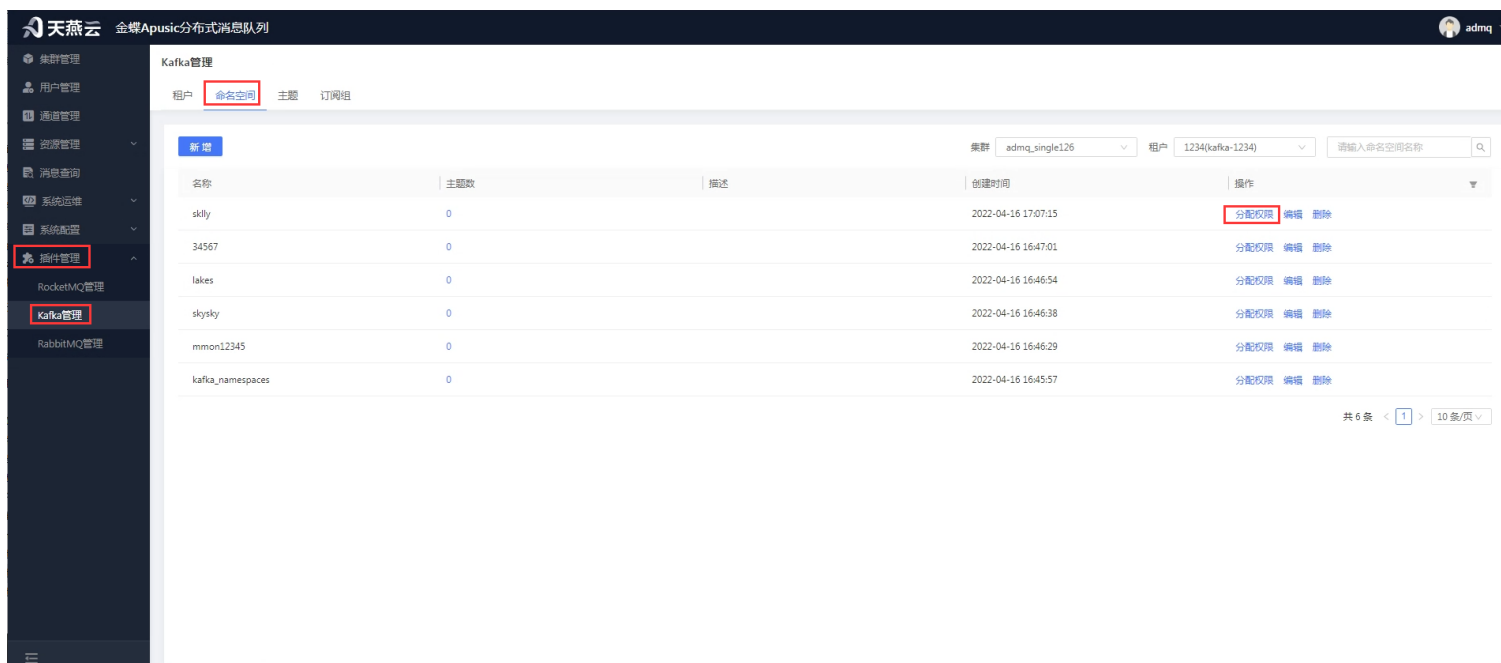
在分配权限页面，选择需要更新授权的用户，点击【编辑】按钮，可更新用户在命名空间下的权限。注意：更新权限前，需满足集群服务可用。



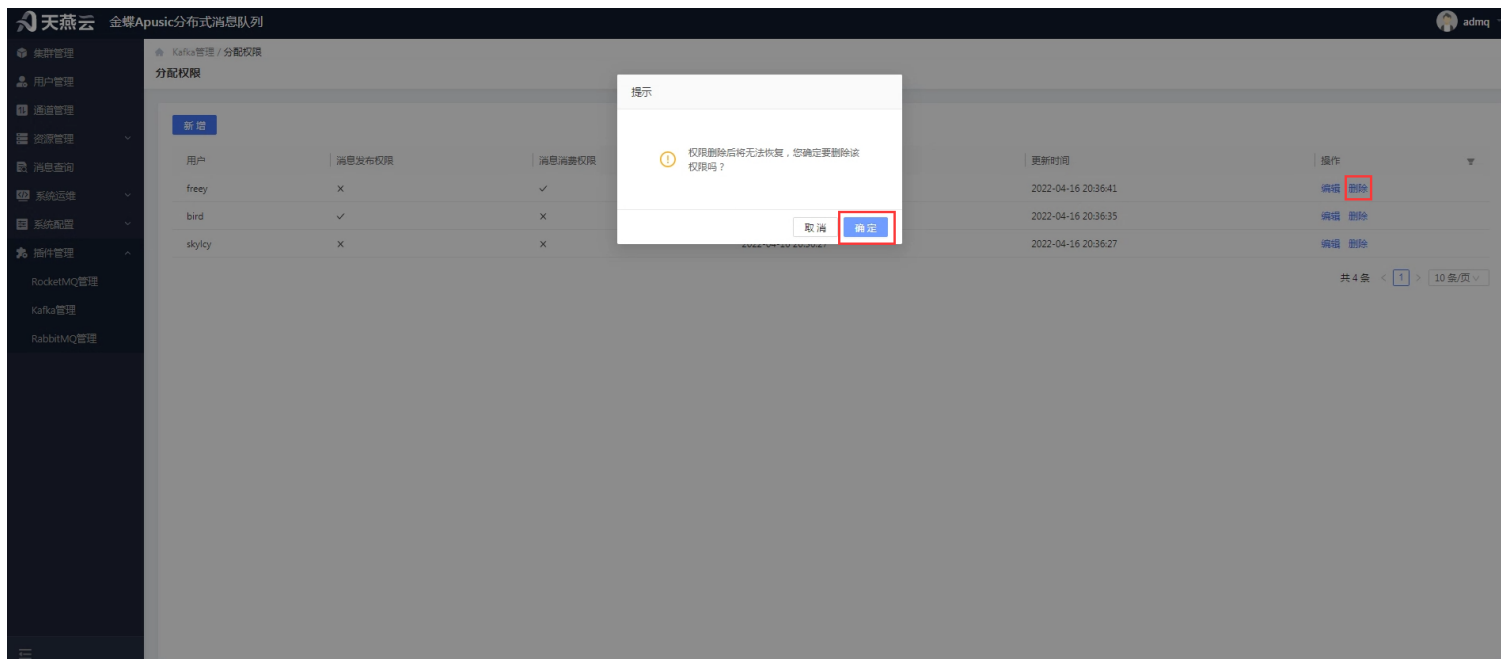
- 用户：用户名称，不可修改
- 令牌名称：令牌名称，不可修改
- 权限：复选框，消息发布权限和消息消费权限，可修改

5.10.2.2.5.3 删除权限

进入【插件管理】->【Kafka管理】，选择“命名空间”页签，选择要删除授权的命名空间名称后的【分配权限】按钮，进入到分配权限页面。

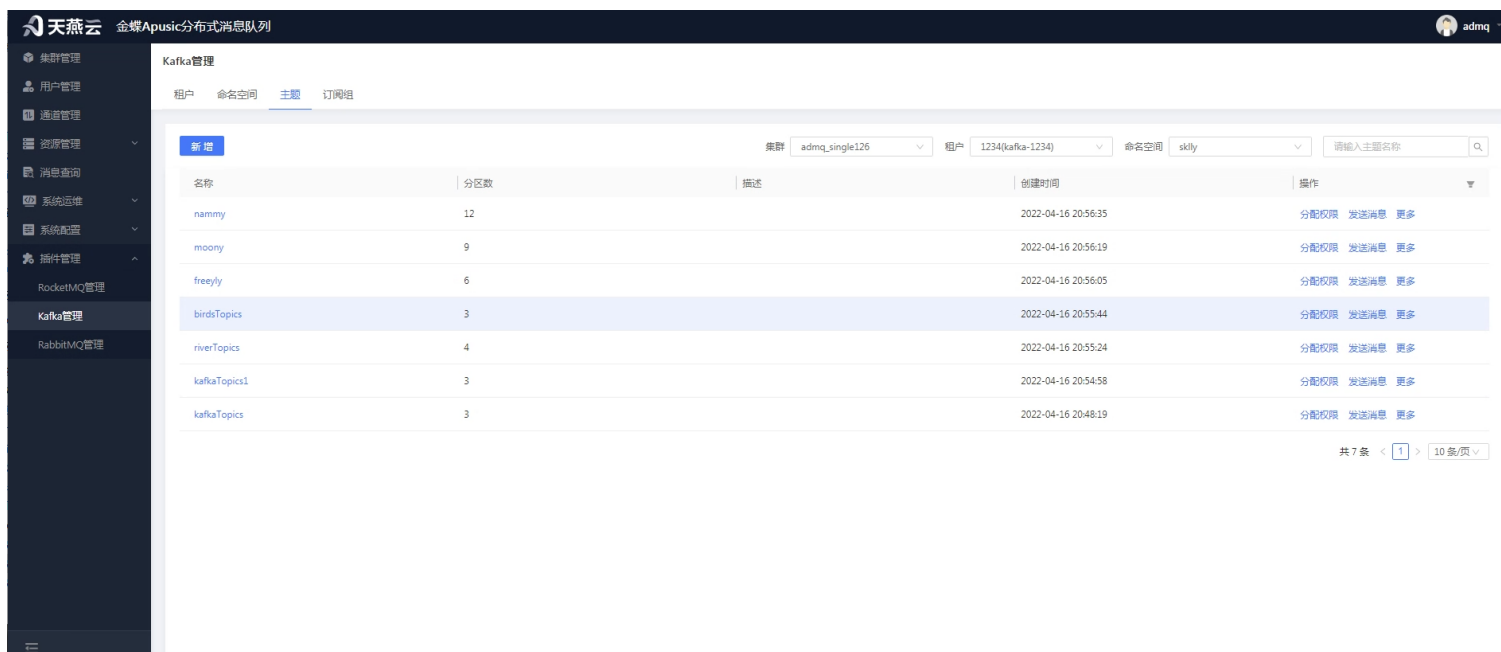


在分配权限页面，选择需要删除授权的用户，点击【删除】按钮，在弹出的确认删除弹框点击【确定】按钮，可删除用户在命名空间下的权限。注意：删除权限前，需满足集群服务可用。



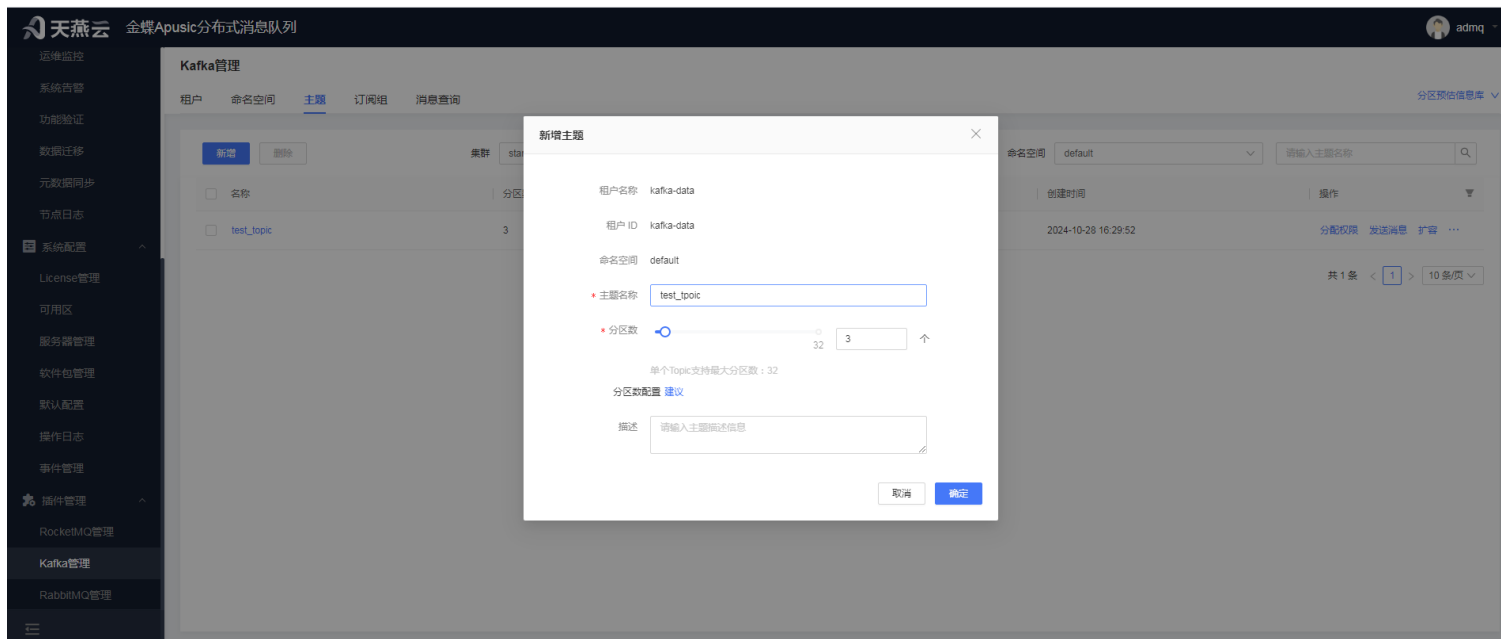
5.10.2.3 主题

主题作用于管理主题资源、主题的用户权限。主题用于新增主题、编辑主题、删除主题、查询主题、新增主题权限、编辑主题权限、删除主题权限、发送消息。



5.10.2.3.1 新增主题

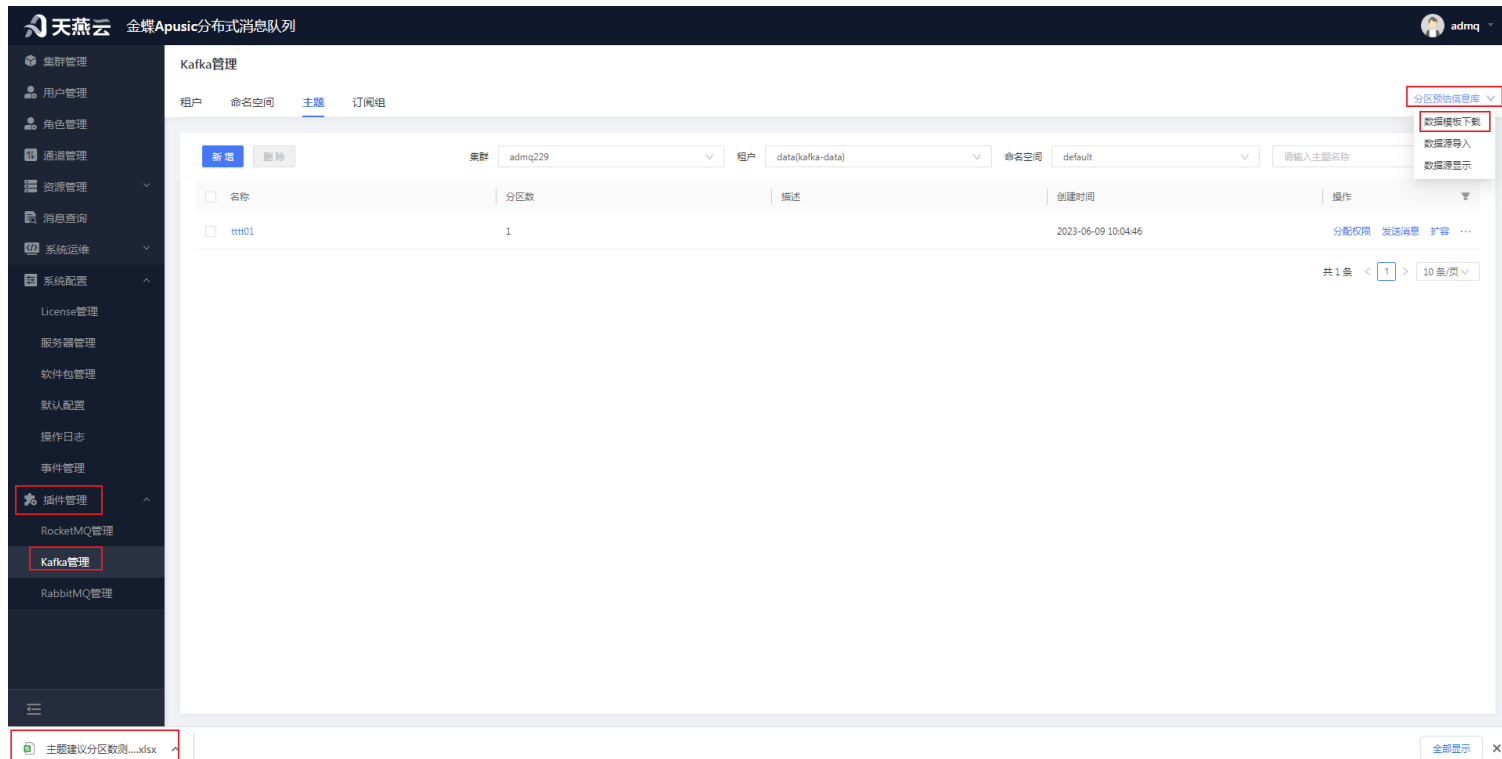
进入【插件管理】->【Kafka管理】，选择"主题"页签，点击左上角【新增】按钮，可新增主题信息。注意：新增主题前，需满足集群服务可用。



- 租户名称：主题所属的租户的名称，根据选择的租户自动带出，不可修改
- 租户ID：主题所属租户的租户ID，根据选择的租户自动带出，不可修改
- 命名空间：主题所属命名空间名称，根据选择的命名空间自动带出，不可修改
- 主题名称：主题的名称，自定义，唯一标识
- 分区数：主题的分区数量，默认分区数为3，可修改
- 描述：主题的描述信息

备注：创建主题时可以使用“预估分区”工具进行匹配合适的分区，预置条件：先下载数据模板，填写事先收集到的性能测试结果数据（这个数据是事先收集的），保存文件然后进行数据源导入操作，导入成功后会在“数据源显示”展示出来，具体操作步骤如下：

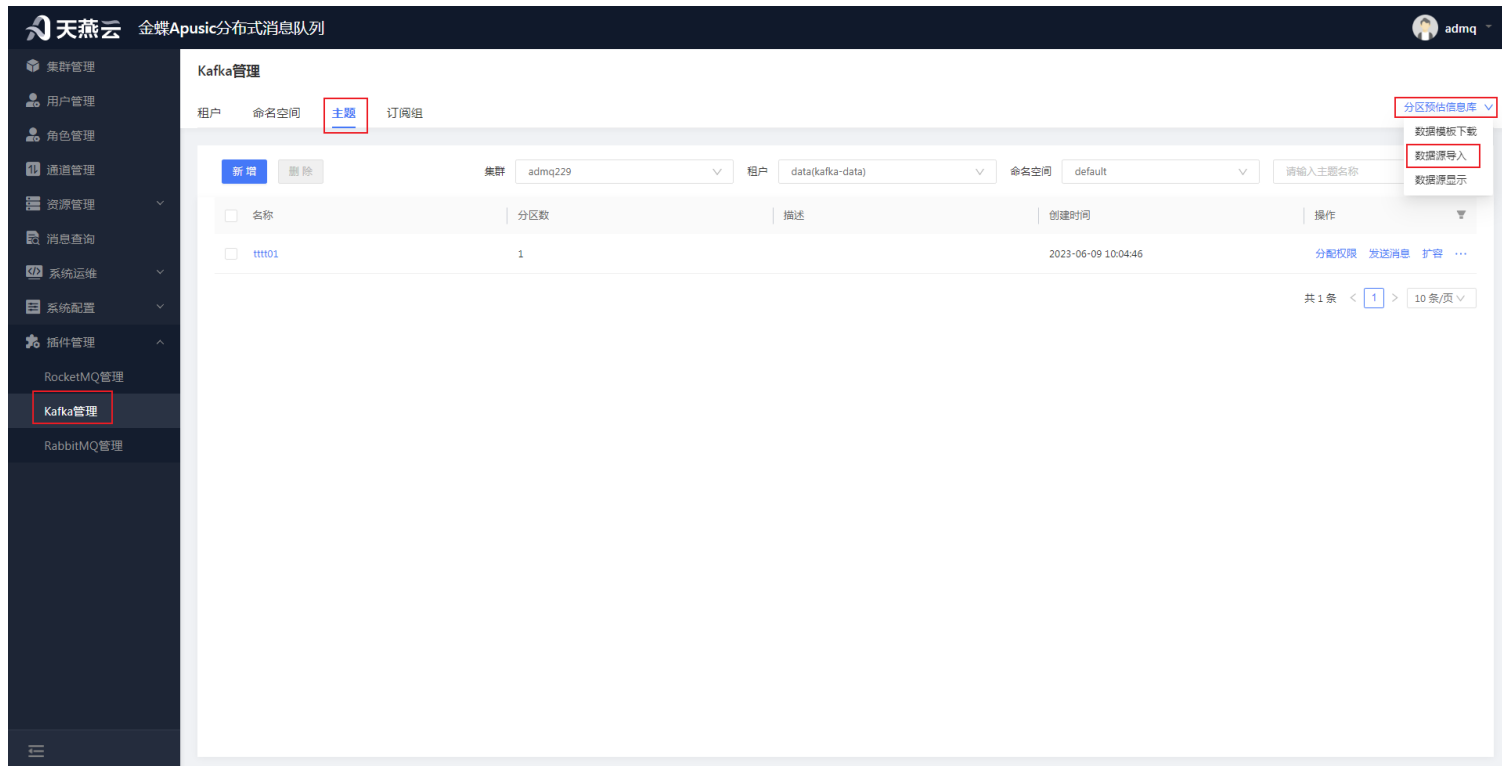
1、插件管理—》Kafka管理—》主题，右上角点击“分区预估信息库”，会出现下拉框，点击“数据模板下载”，下载文件 主题建议分区数测试数据.xlsx



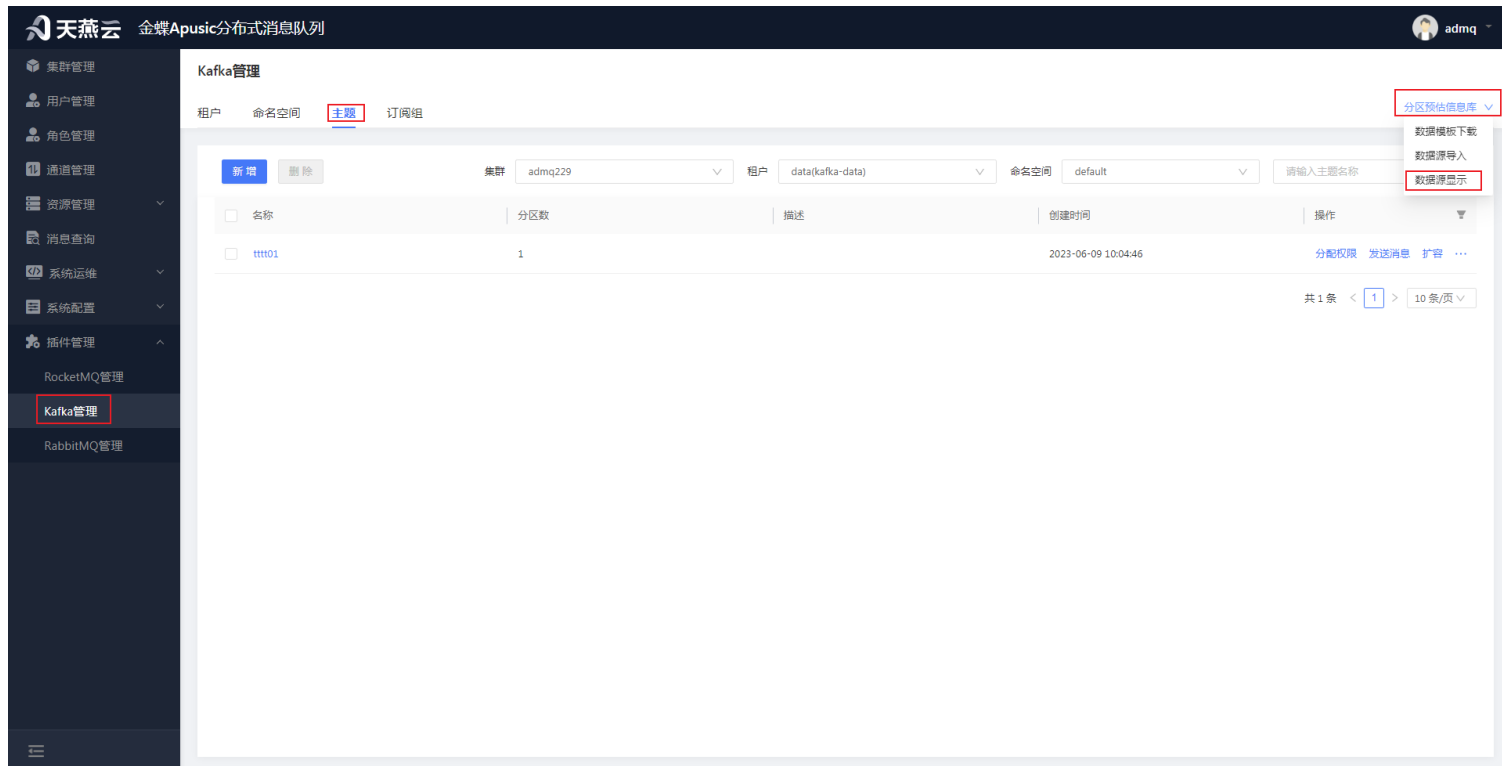
2、填写性能测试数据至模板文件中（注意：这里的数据是虚构的，不是真实的测试数据，真实的性能测试数据需要不断收集）：

	A	B	C	D	E	F	G	H	I
	存储节点数(必填)	计算节点数(必填)	发送速率(条/s, 必填)	接收速率(条/s, 必填)	生产者数量(必填)	消费者数量(必填)	消息大小(kb/msg, 必填)	建议最小分区(必填)	建议最大分区(必填)
1	3	3	100000	100000	2	2	1	6	12
2	3	6	200000	200000	4	2	1	3	6
3	10	4	50000	20000	1	1	3	2	7
4	5	5	20000	10000	1	1	1	2	4
5	1	1	30000	30000	1	1	1	3	6

3、将文件导入到信息库，Kafka管理—》主题—》分区预估信息库—》点击“数据源导入”，出现导入弹窗，将文件拖入固定区域，点击导入按钮：



成功导入后，点击“数据源显示”，会将数据显示出来（展示数据的目的是为了用户可以随时查看之前的性能测试数据）：

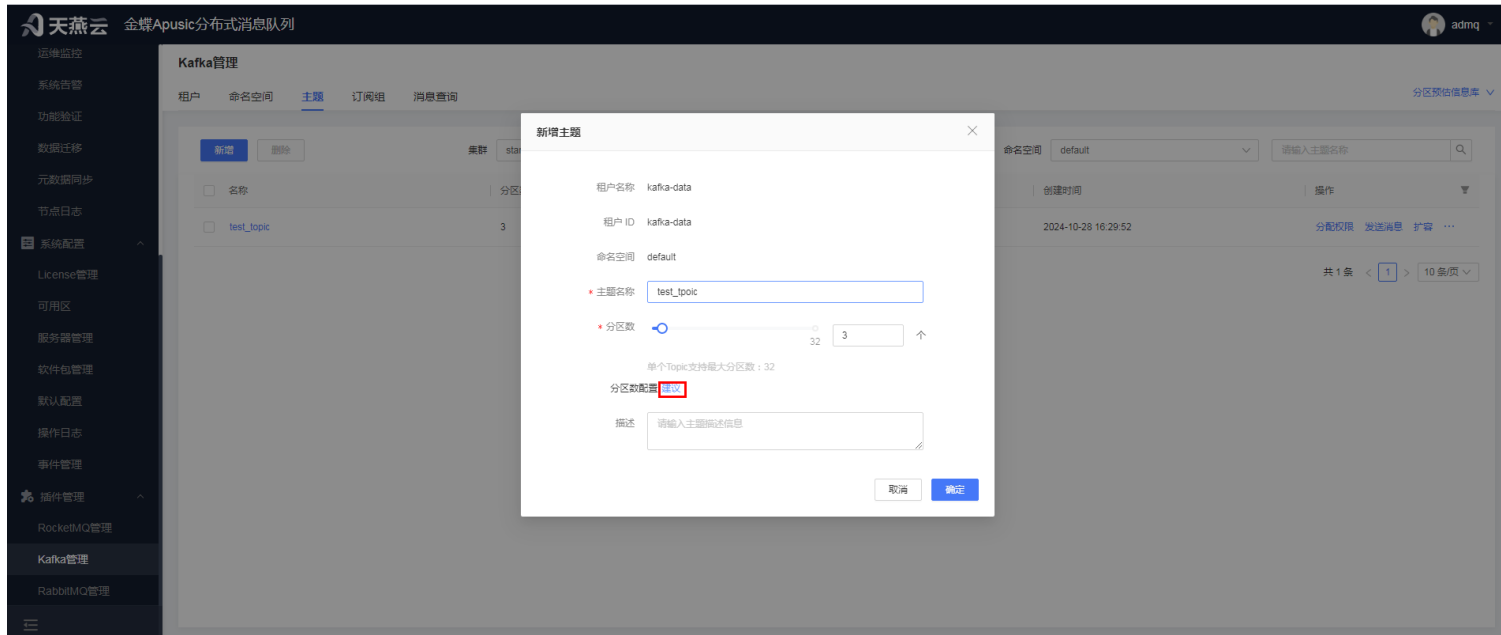


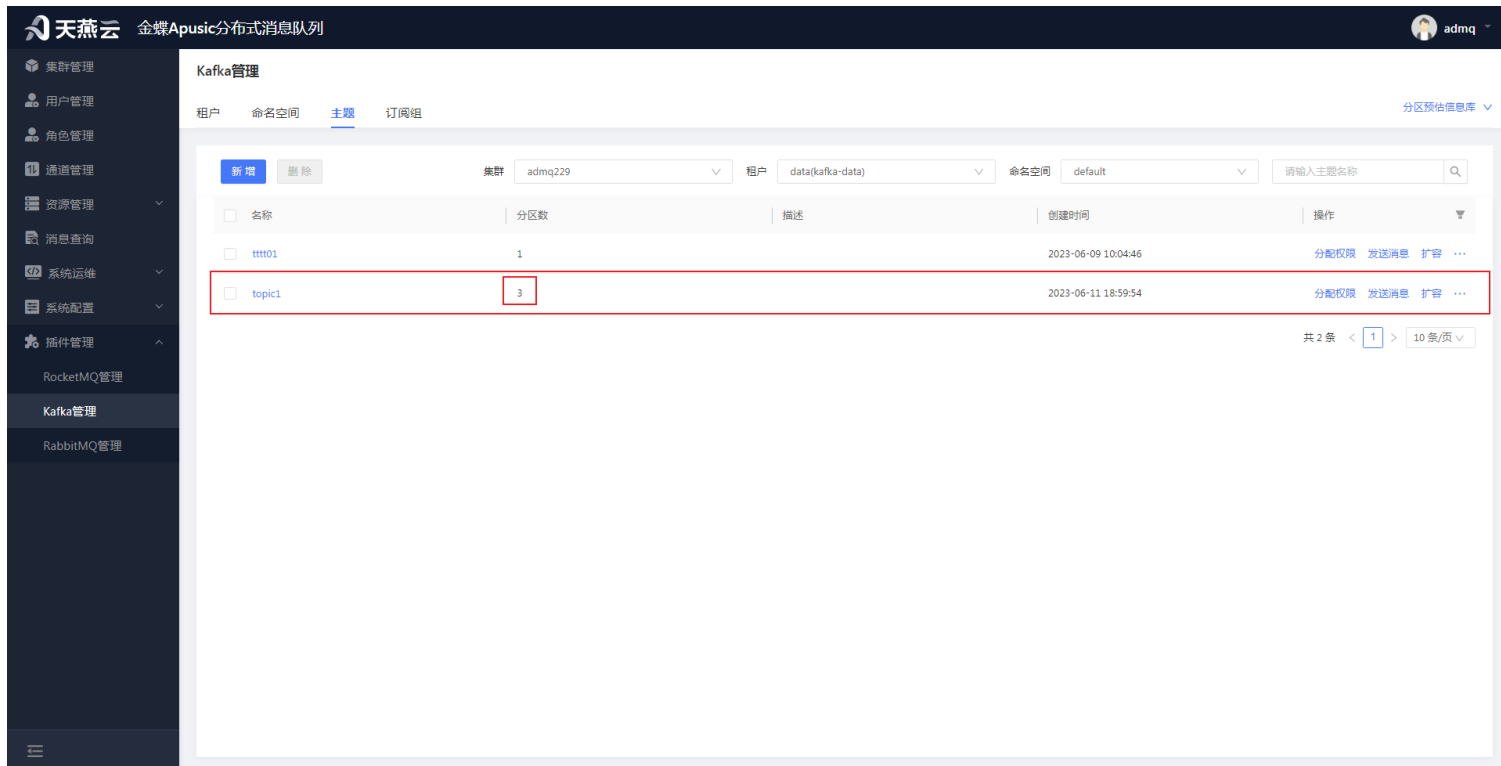
分区预估数据源								
存储节点数	计算节点数	发送速率...	接收速率...	生产者数量	消费者数量	消息大小...	建议最小...	建议最大...
3	3	100000	100000	2	2	1	6	12
3	6	200000	200000	4	2	1	3	6
10	4	50000	20000	1	1	3	2	7
5	5	20000	10000	1	1	1	2	4
1	1	30000	30000	1	1	1	3	6

共 5 条 < 1 > 10 条/页

关闭

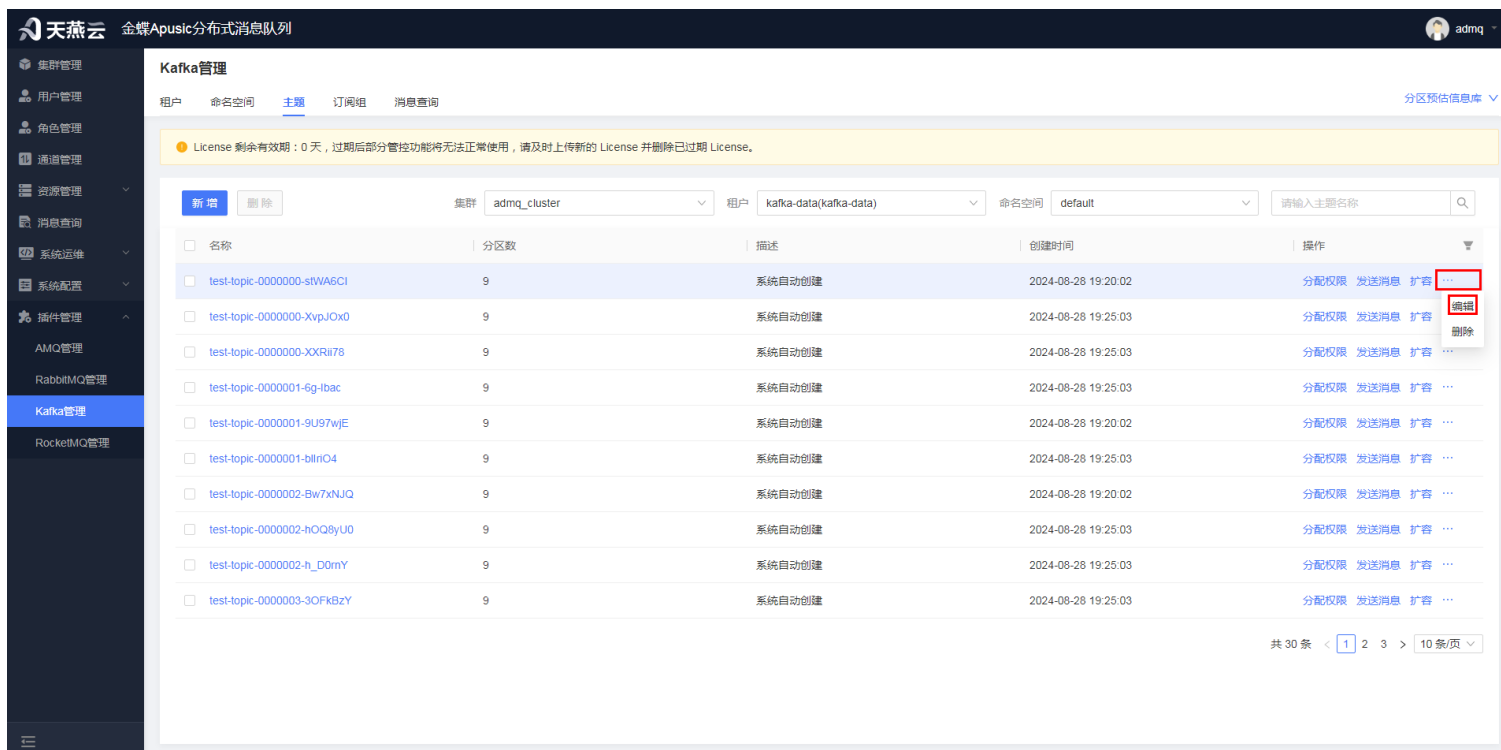
3、创建主题时，在弹窗页面点击“建议”，出现分区数配置建议弹窗页面，输入相关信息后，点击“确定”按钮，

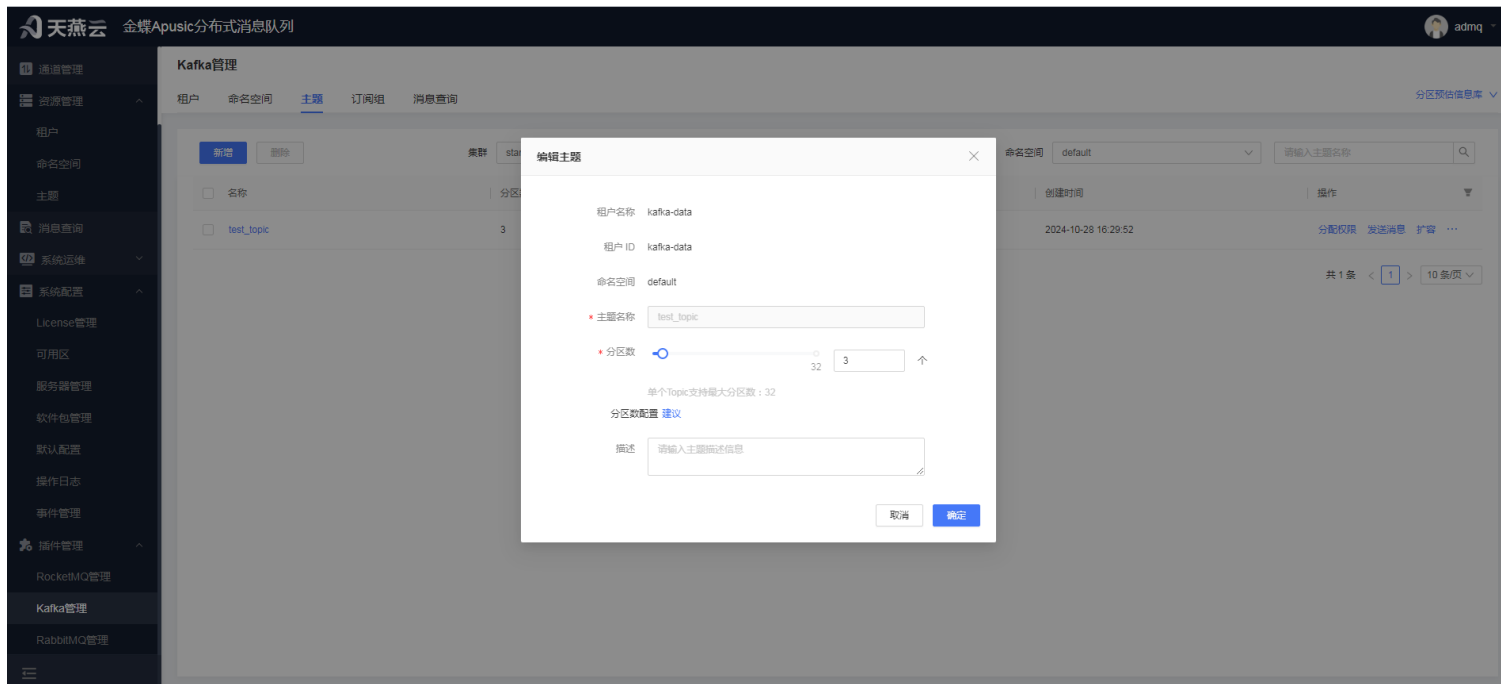




5.10.2.3.2 编辑主题

进入【插件管理】->【Kafka管理】，选择"主题"页签，选择要编辑的主题名称后的【...】按钮->【编辑】按钮，可编辑主题信息。注意：编辑主题前，需满足集群服务可用。

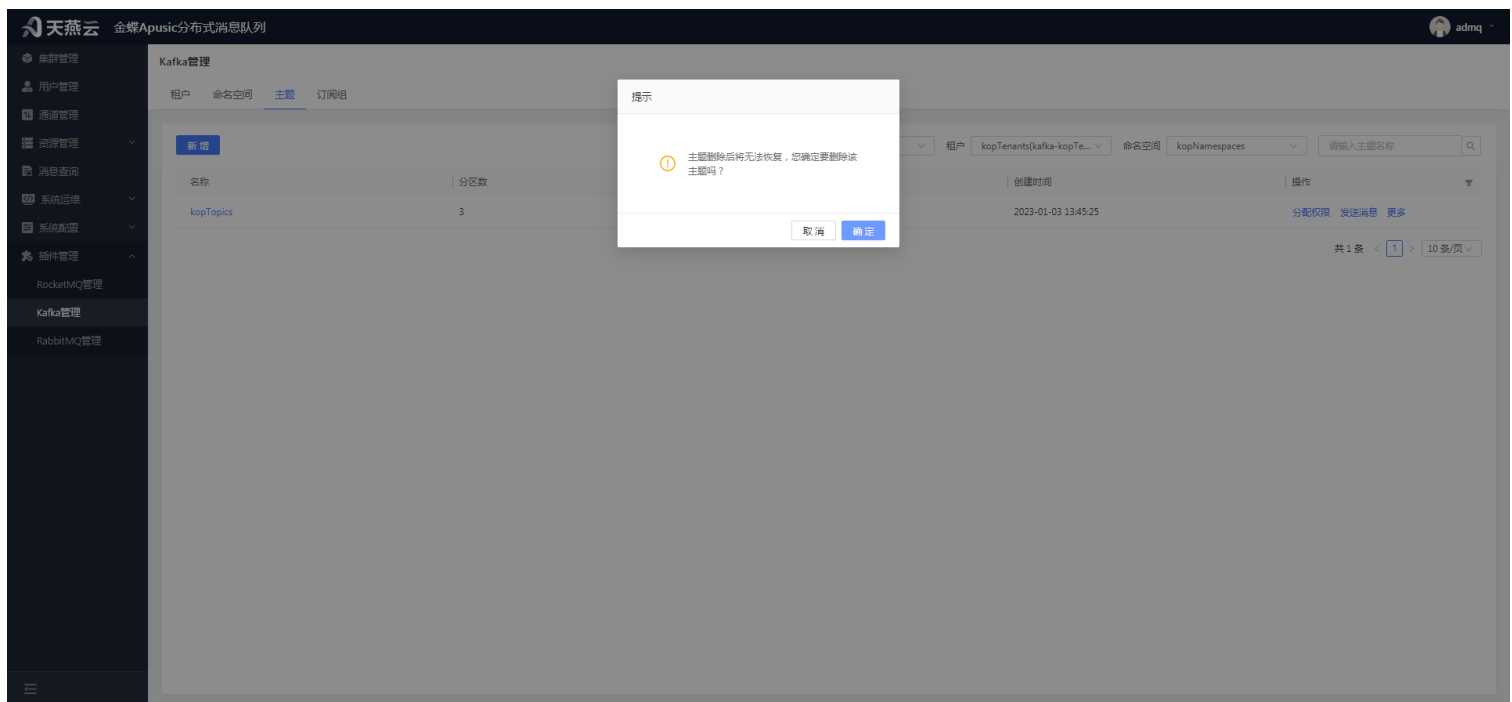
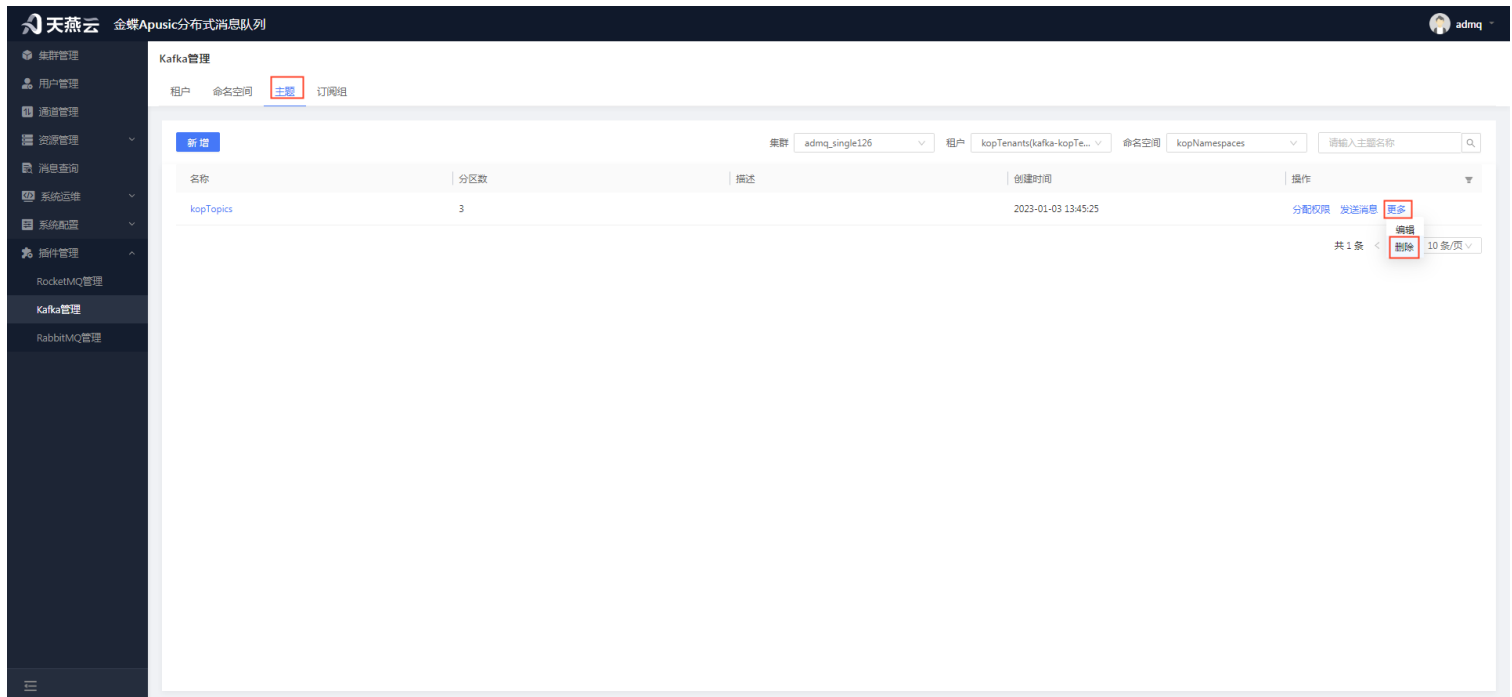




- 租户名称：主题所属租户名称，根据选择的租户自动带出，不可修改
- 租户ID：主题所属租户的租户ID，根据选择的租户自动带出，不可修改
- 命名空间：主题所属命名空间名称，根据选择的命名空间自动带出，不可修改
- 主题名称：主题的名称，置灰显示，不可修改
- 分区数：主题的分区数量，置灰显示，不可修改
- 描述：主题的描述信息

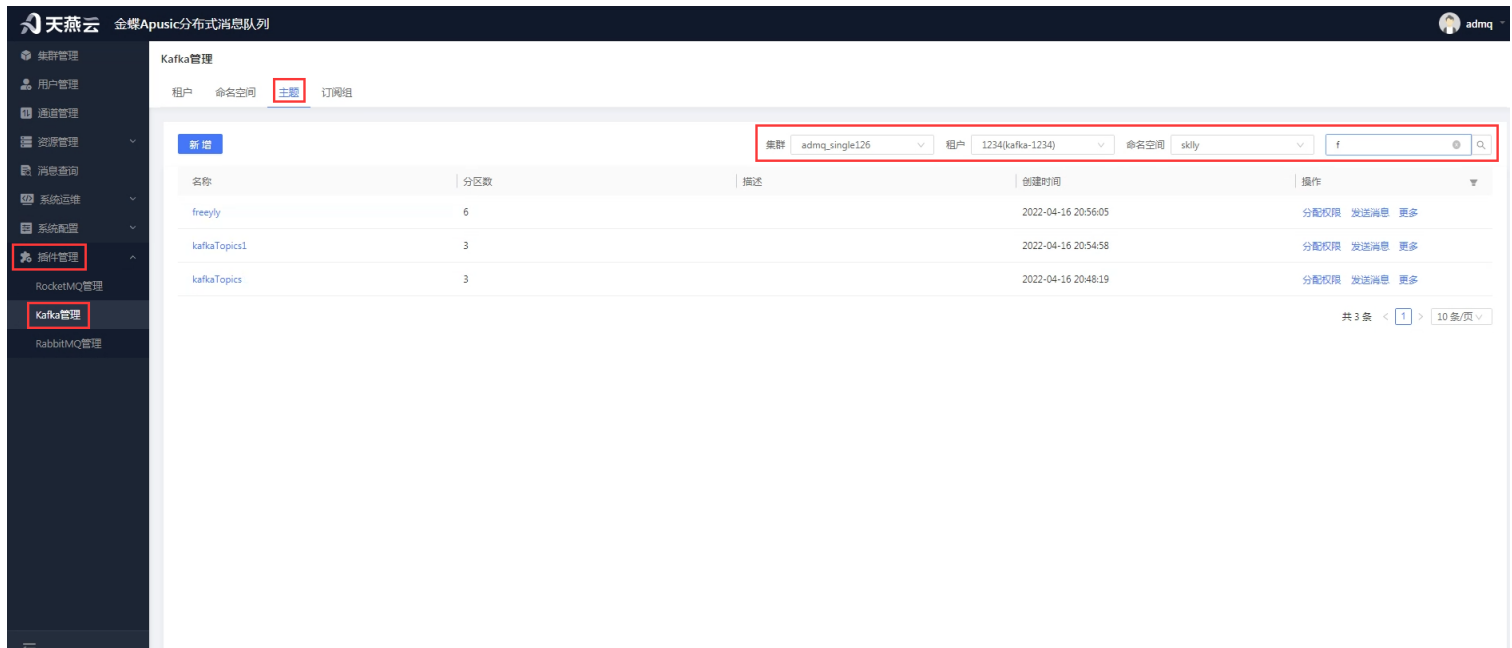
5.10.2.3.3 删除主题

进入【插件管理】->【Kafka管理】，选择"主题"页签，选择要删除的主题名称后的【更多】按钮->【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除主题信息。注意：删除主题前，需满足集群服务可用。



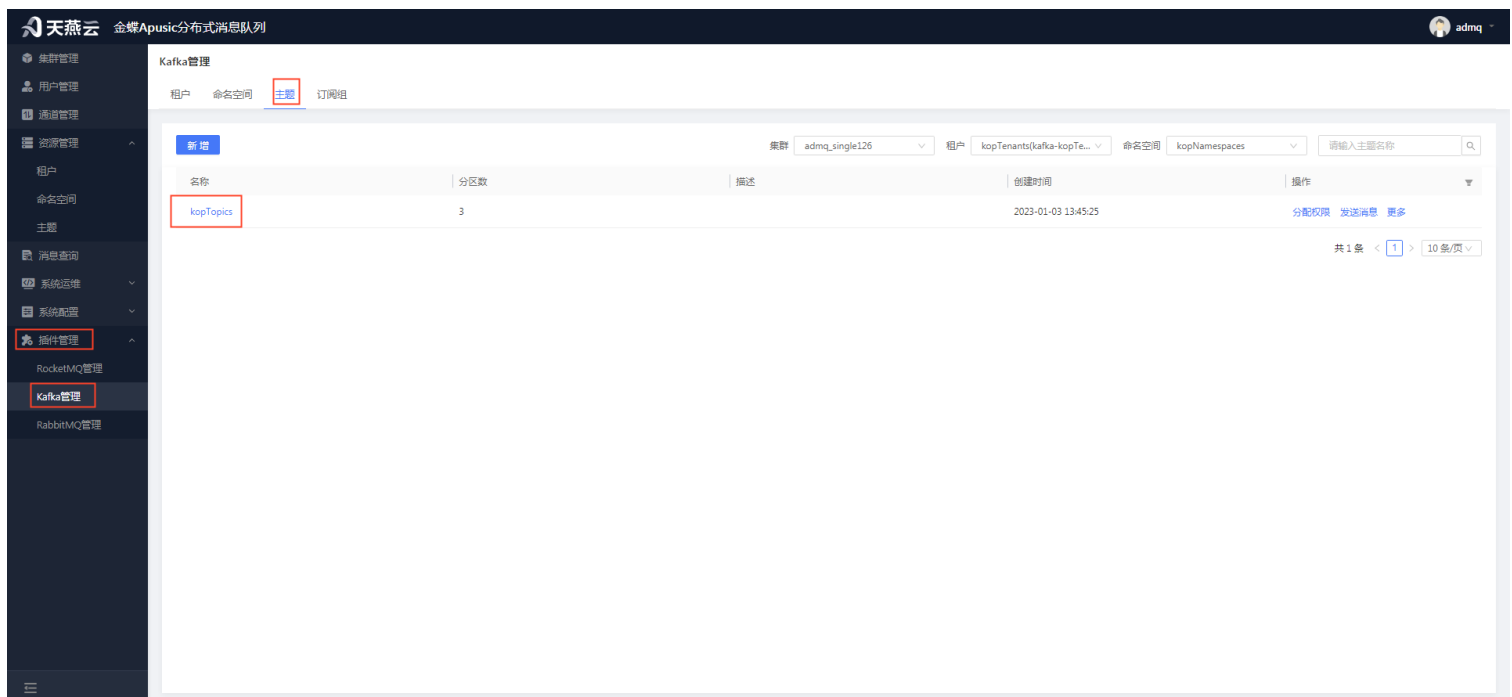
5.10.2.3.4 查询主题

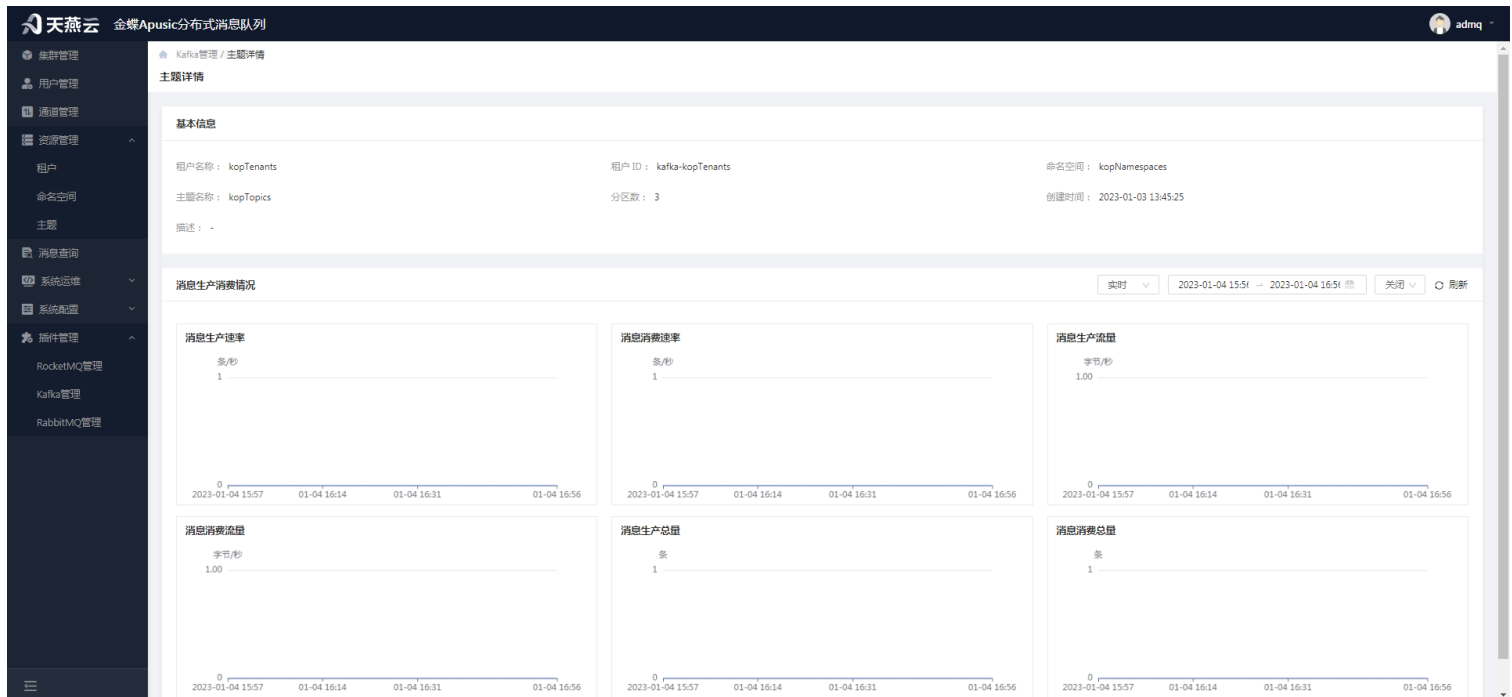
进入【插件管理】->【Kafka管理】，选择"主题"页签，右上角选择集群名称、租户名称、命名空间名称、输入主题名称，可查询主题信息。



5.10.2.3.5 查看主题详情

进入【插件管理】->【Kafka管理】，选择“主题”页签，点击某个主题的名称，可查看主题详情。

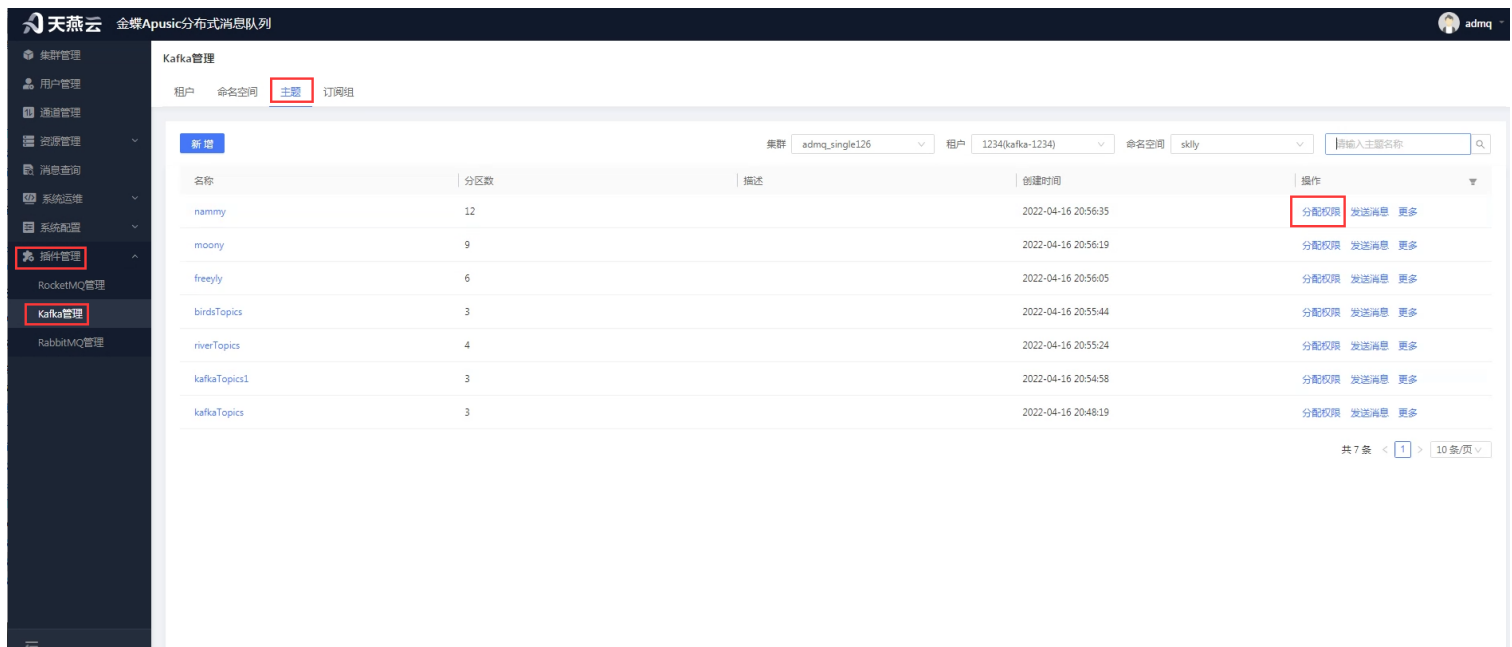




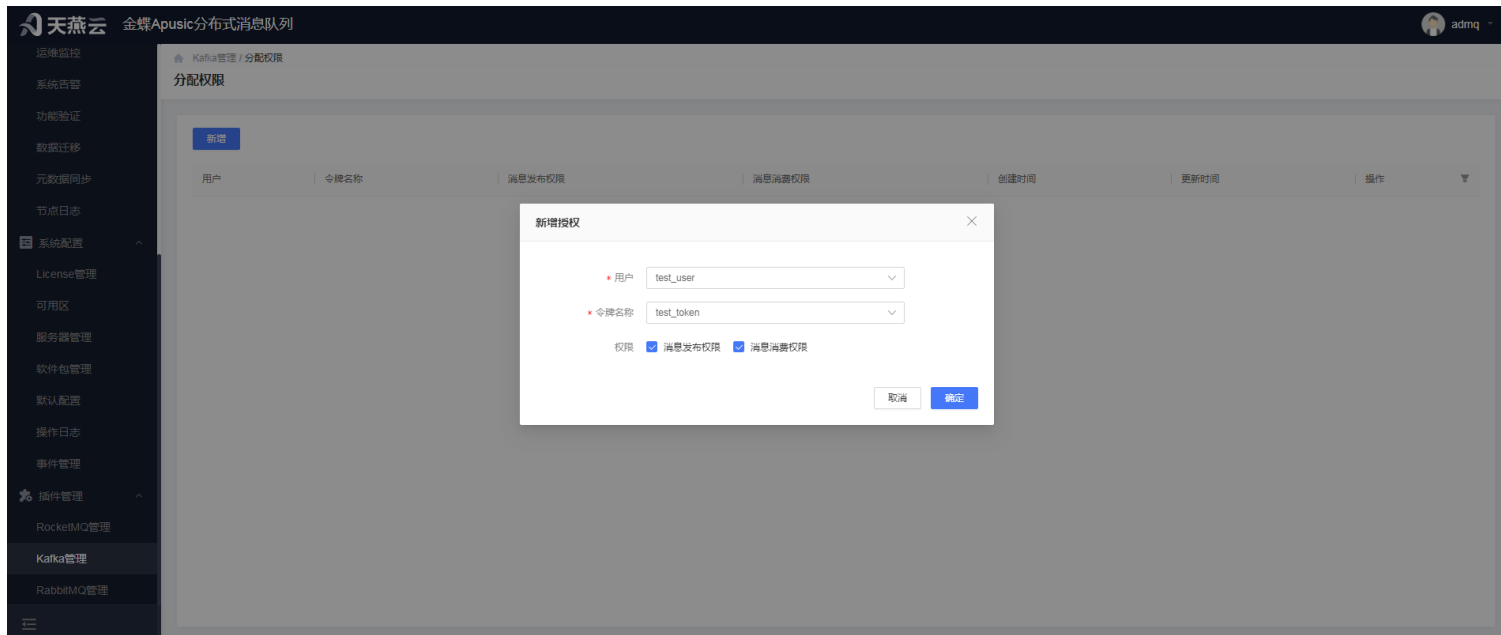
5.10.2.3.6 发布消费权限

5.10.2.3.6.1 新增权限

进入【插件管理】->【Kafka管理】，选择"主题"页签，选择要新增授权的主题名称后的【分配权限】按钮，跳转到分配权限页面。



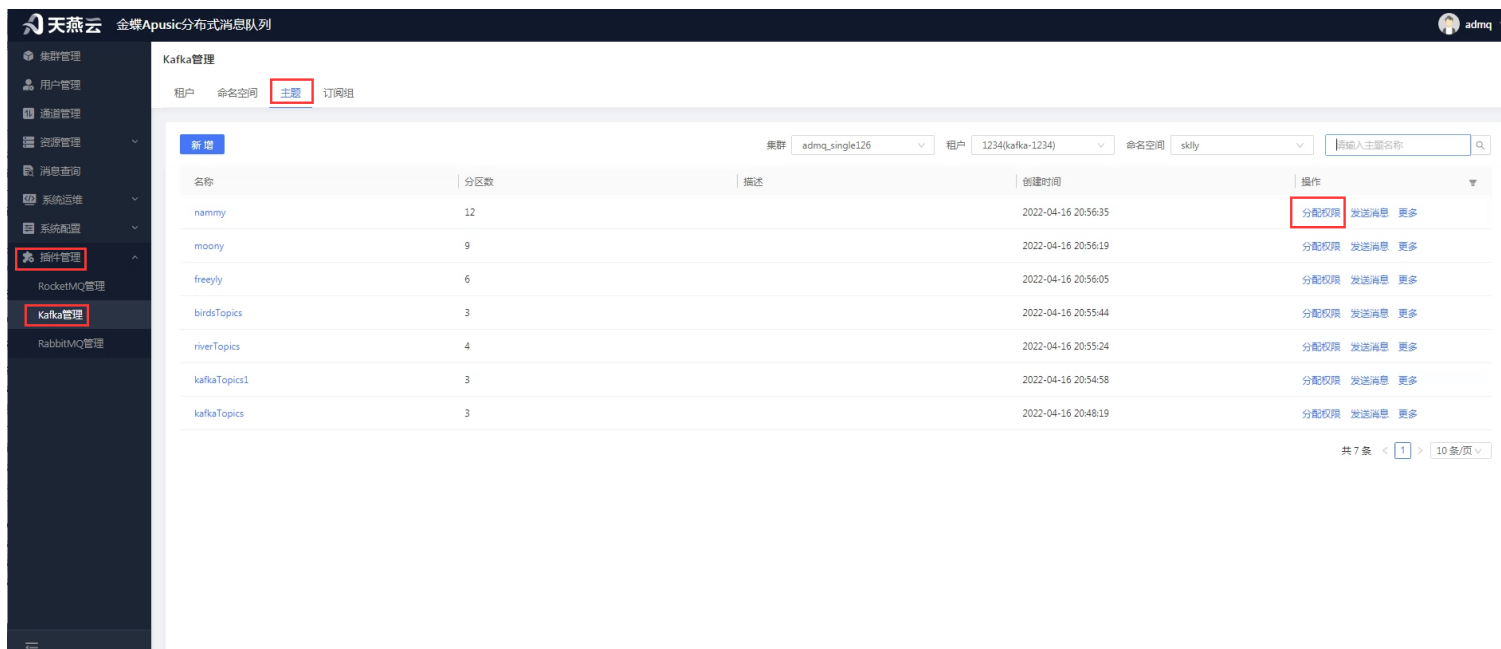
在分配权限页面，点击左上角的【新增】按钮，可新增主题下用户的权限信息。注意：新增权限前，需满足集群服务可用；授权主要是给普通用户提供授权，默认超级管理员拥有所有主题的消息发布权限和消息消费权限；若主题所属命名空间也有权限，则按最小粒度，主题设置的权限有效；若用户在命名空间无权限则需要先对命名空间进行授权。



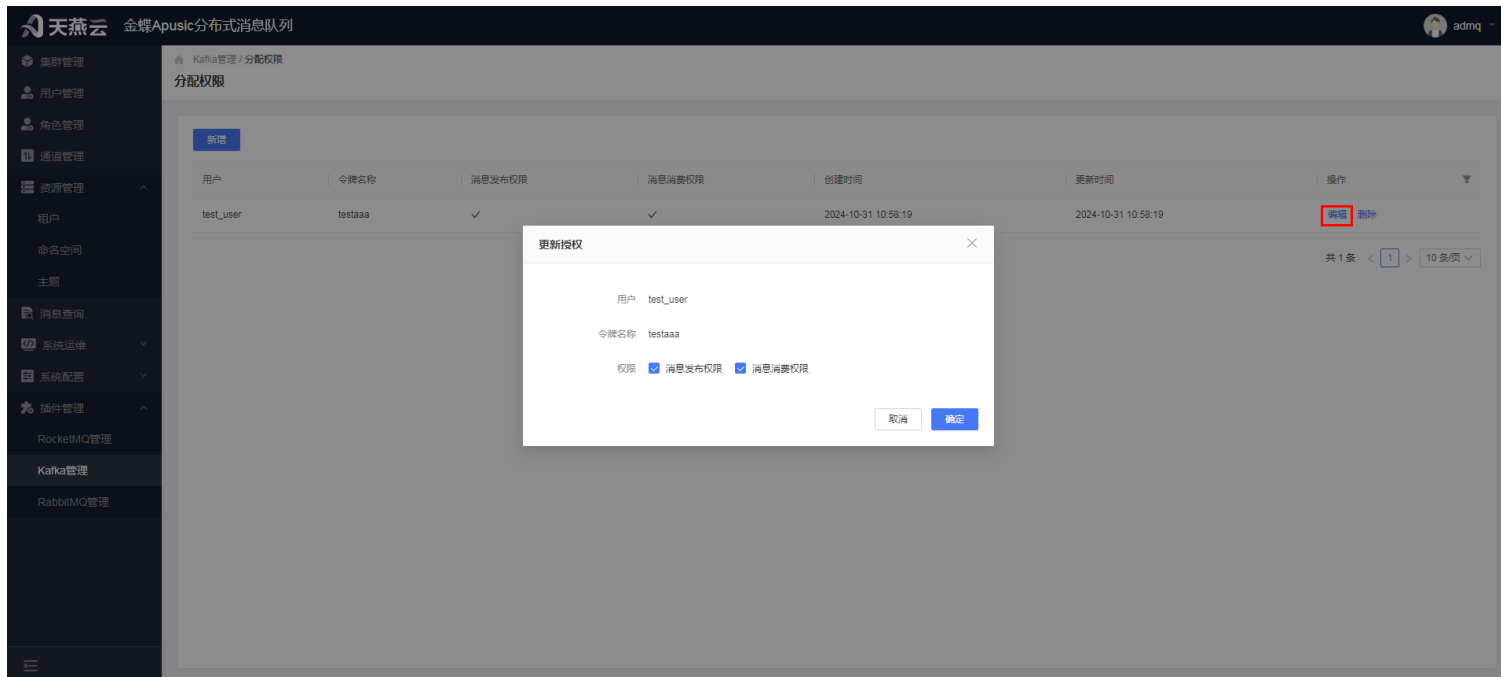
- 用户：下拉选择，获取的用户管理（4.1）的用户信息
- 令牌名称：下拉选择，获取选择用户的令牌。
- 权限：复选框，消息发布权限和消息消费权限

5.10.2.3.6.2 编辑权限

进入【插件管理】->【Kafka管理】，选择"主题"页签，选择要更新授权的主题名称后的【更多】按钮->【分配权限】按钮，进入到分配权限页面。



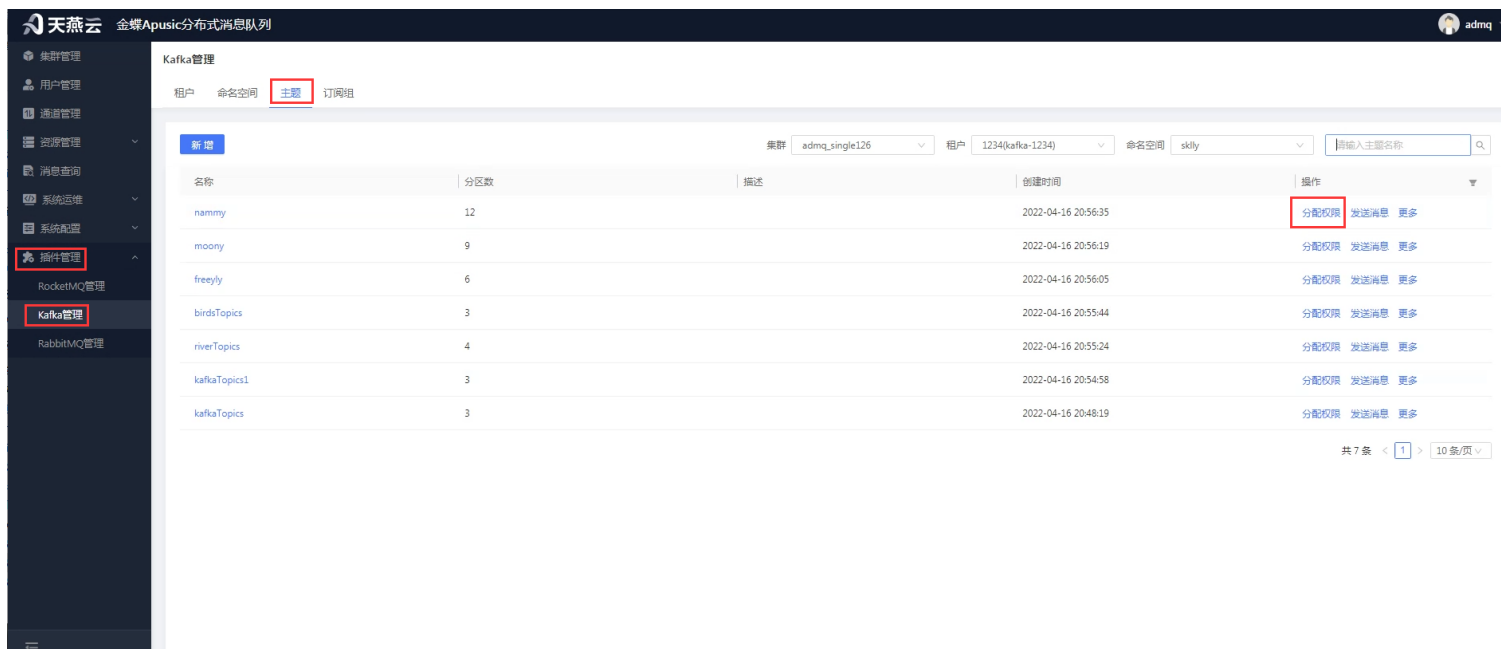
在分配权限页面，选择需要更新授权的用户，点击【编辑】按钮，可更新用户在主题下的权限。注意：更新权限前，需满足集群服务可用；若主题所属命名空间也有权限，则按最小粒度，主题设置的权限有效。



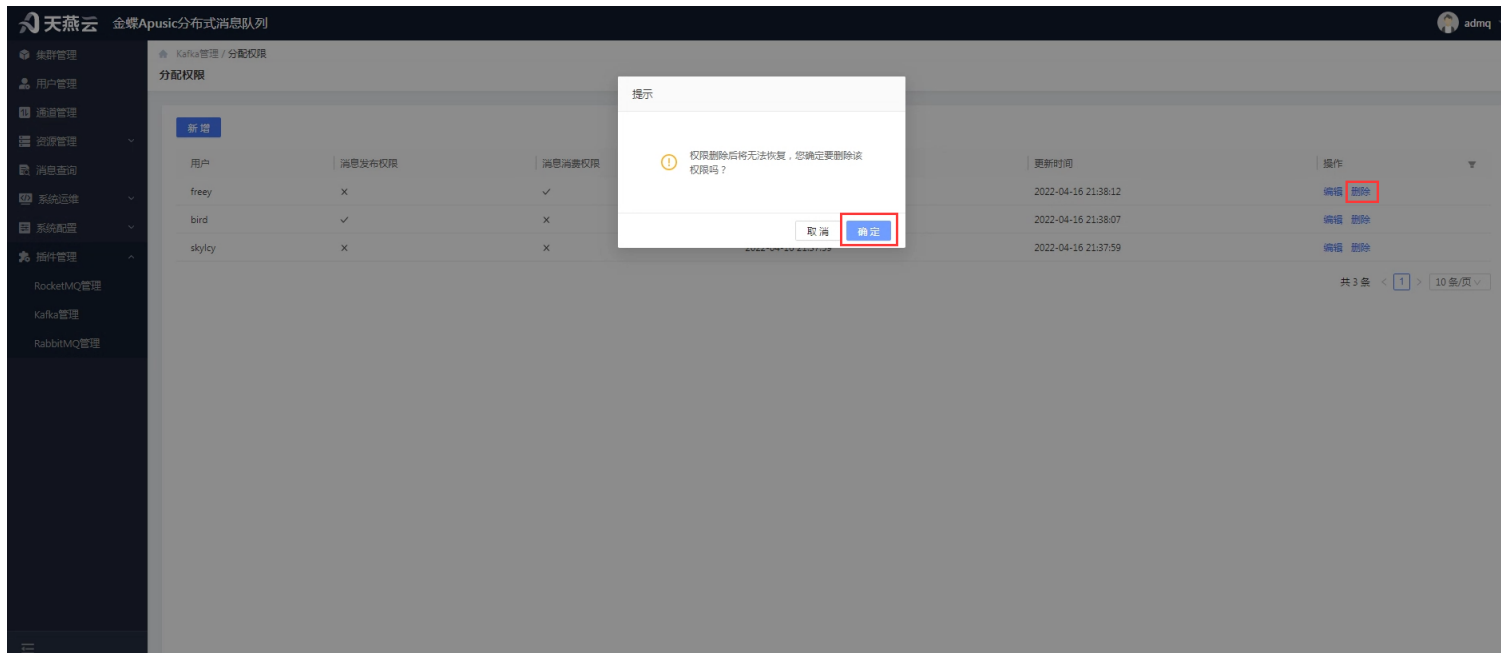
- 用户: 用户名称, 不可修改
- 令牌名称: 令牌名称, 不可修改
- 权限: 复选框, 消息发布权限和消息消费权限

5.10.2.3.6.3 删除权限

进入【插件管理】->【Kafka管理】，选择"主题"页签，选择要删除授权的主题名称后的【更多】按钮->【分配权限】按钮，进入到分配权限页面。

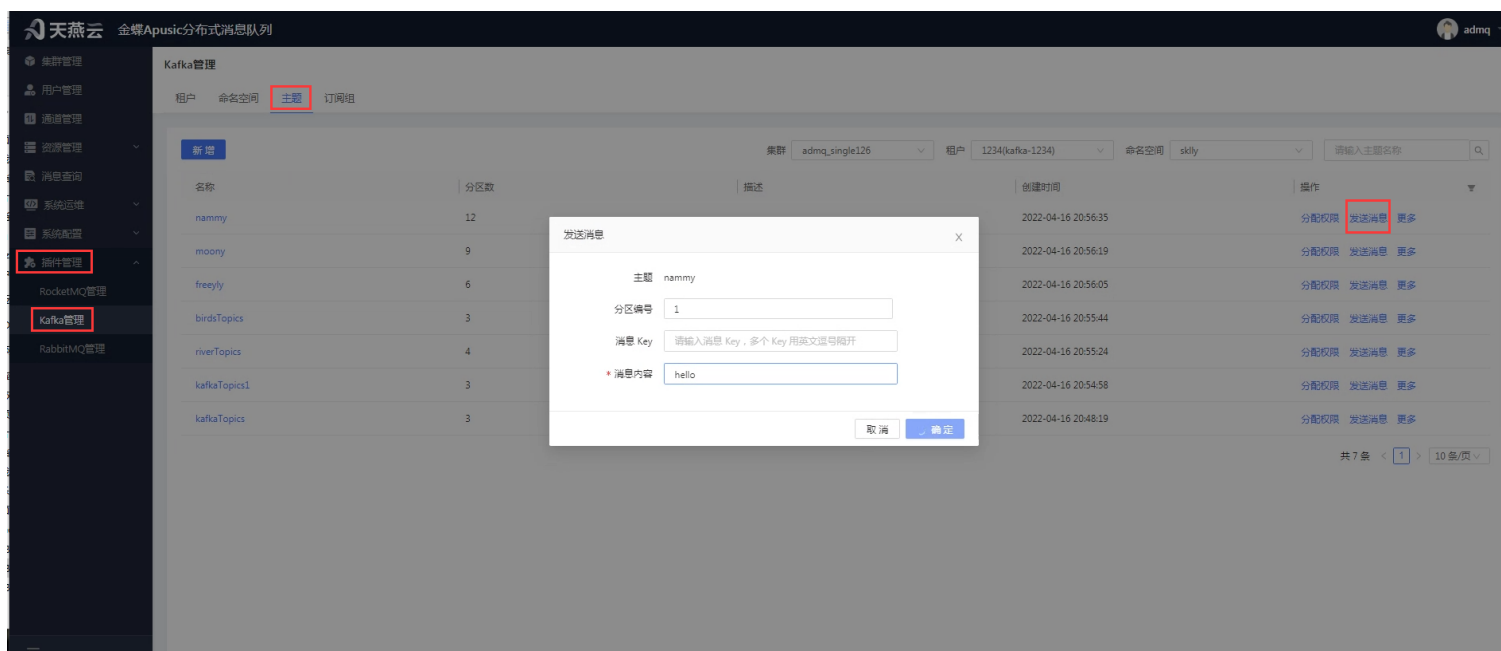


在分配权限页面，选择需要删除授权的用户，点击【删除】按钮，在弹出的确认删除弹框点击【确定】按钮，可删除用户在主题下的权限。注意：删除权限前，需满足集群服务可用；若主题所属命名空间也有权限，则按最小粒度，主题设置的权限有效，若删除权限的用户的命名空间下有权限，则命名空间权限生效。



5.10.2.3.7 发送消息

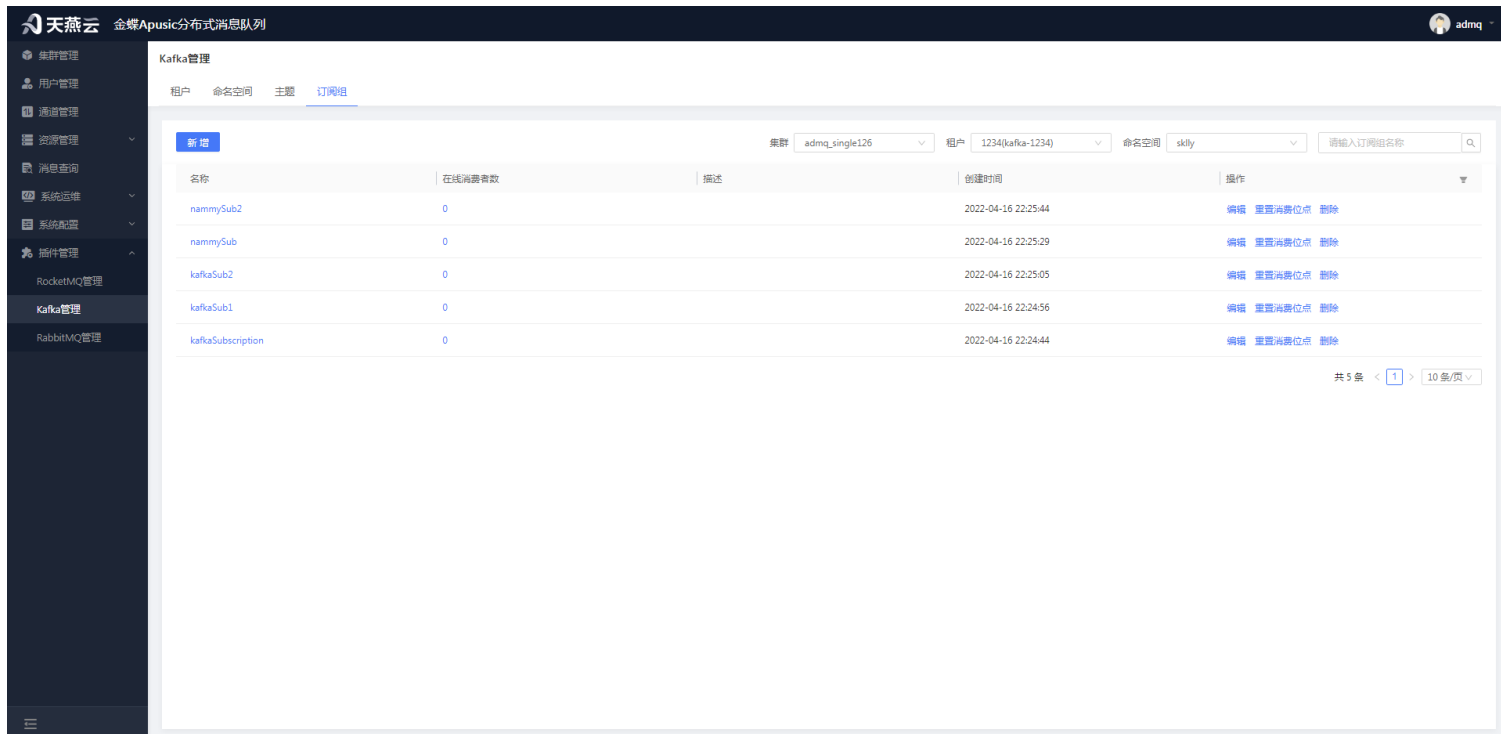
进入【插件管理】->【Kafka管理】，选择"主题"页签，选择要发送消息的主题名称后的【发送消息】按钮，可发送消息，发送的消息，可在消息查询中查到。
(这里功能有问题，消息发不出去)



- 主题：发送消息的主题名称，根据选择的主题名称自动带出，不可修改
- 分区编号：主题的分区编号，自定义，用来指定将消息发送到主题的某一个确定的分区
- 消息Key：消息key，自定义
- 消息内容：发送的消息内容，自定义

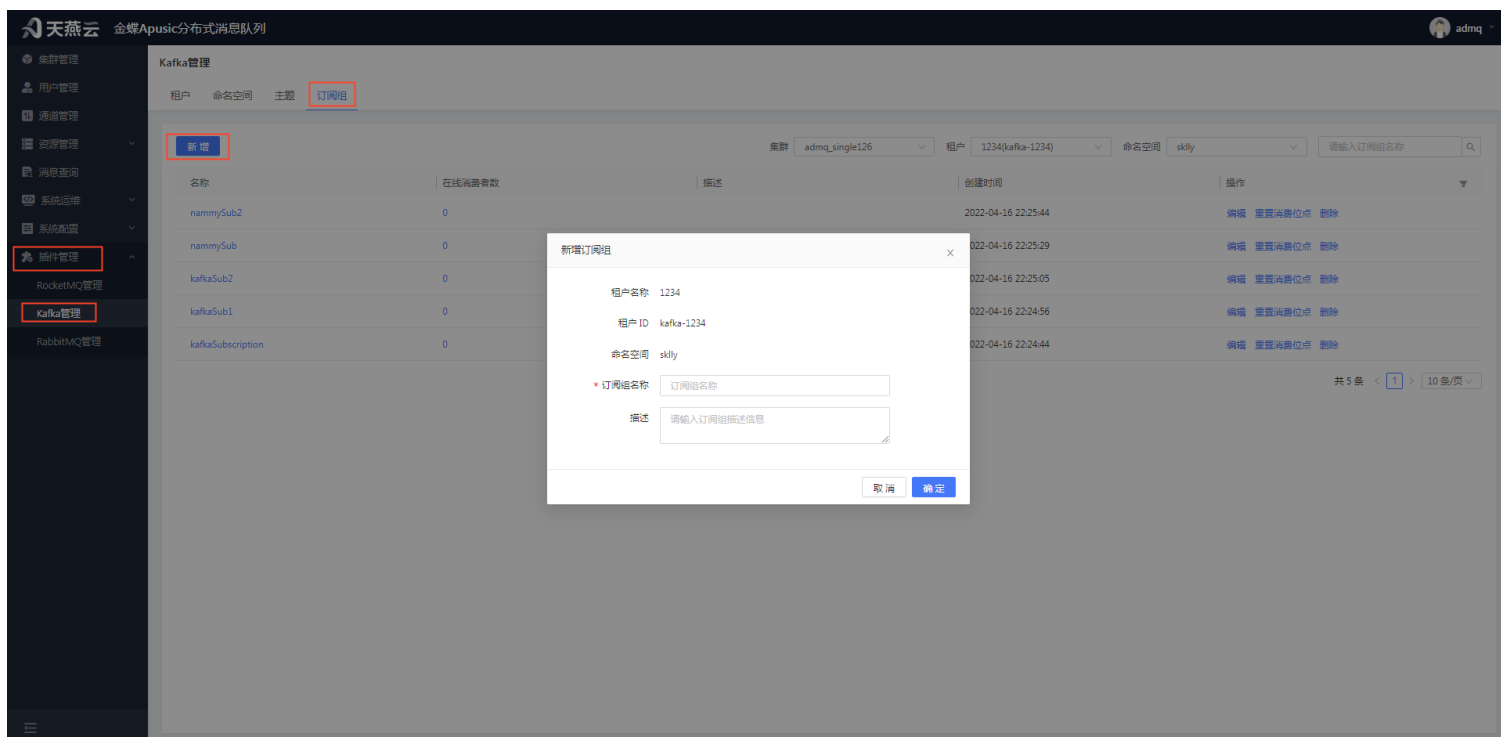
5.10.2.4 订阅组

发送消息和接收消息，都需要绑定订阅组，订阅组就是作用于管理订阅组。订阅组用于新增订阅组、编辑订阅组、删除订阅组、查询订阅组、重置消费位点（回溯消息）



5.10.2.4.1 新增订阅组

进入【插件管理】->【Kafka管理】，选择“订阅组”页签，点击左上角【新增】按钮，可新增订阅组信息。注意：新增订阅组前，需满足集群服务可用。

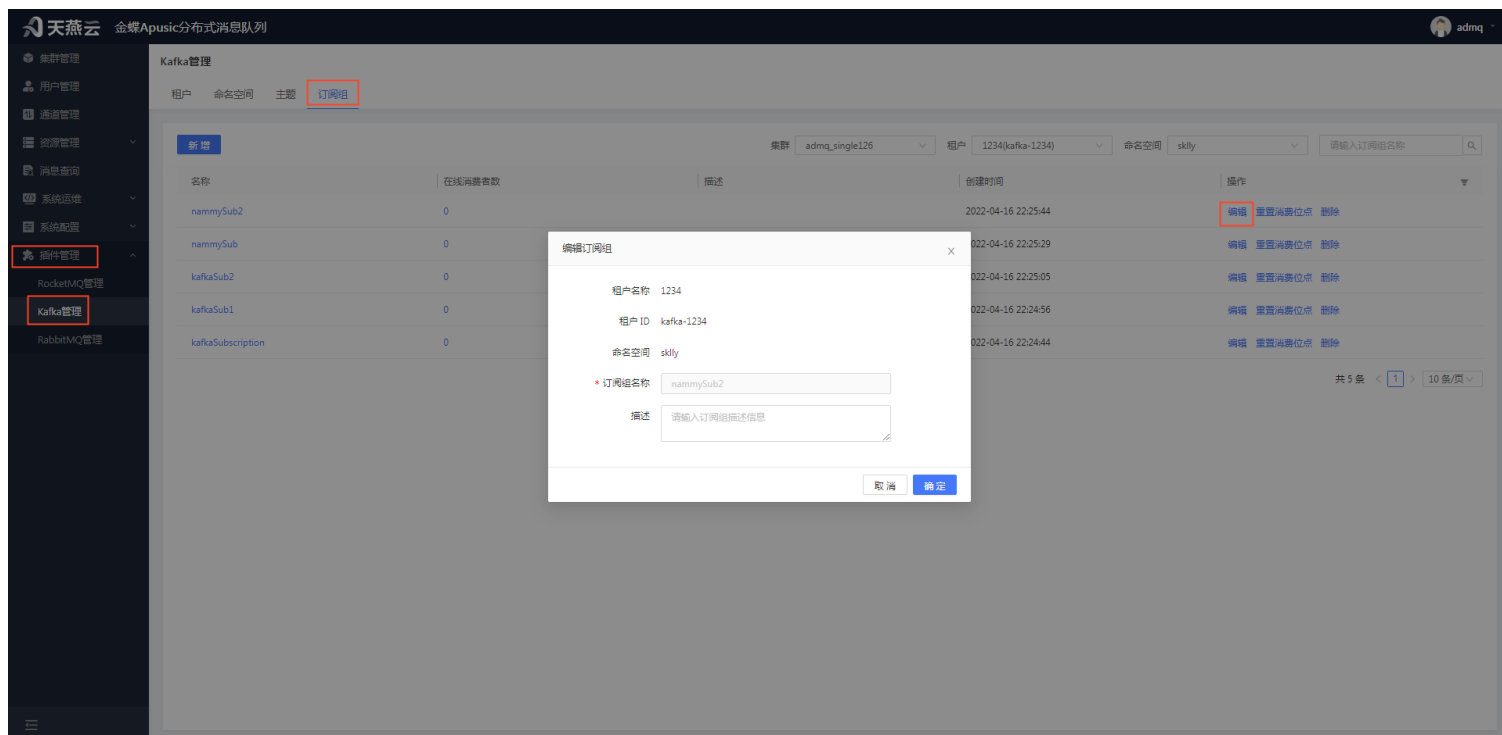


- 租户名称：订阅组所属的租户名称，根据选择的租户自动带出，不可修改
- 租户ID：订阅组所属租户的租户ID，根据选择的租户自动带出，不可修改
- 命名空间：订阅组所属命名空间名称，根据选择的命名空间自动带出，不可修改
- 订阅组名称：订阅组的名称，自定义，唯一标识

- 描述：订阅组的描述信息

5.10.2.4.2 编辑订阅组

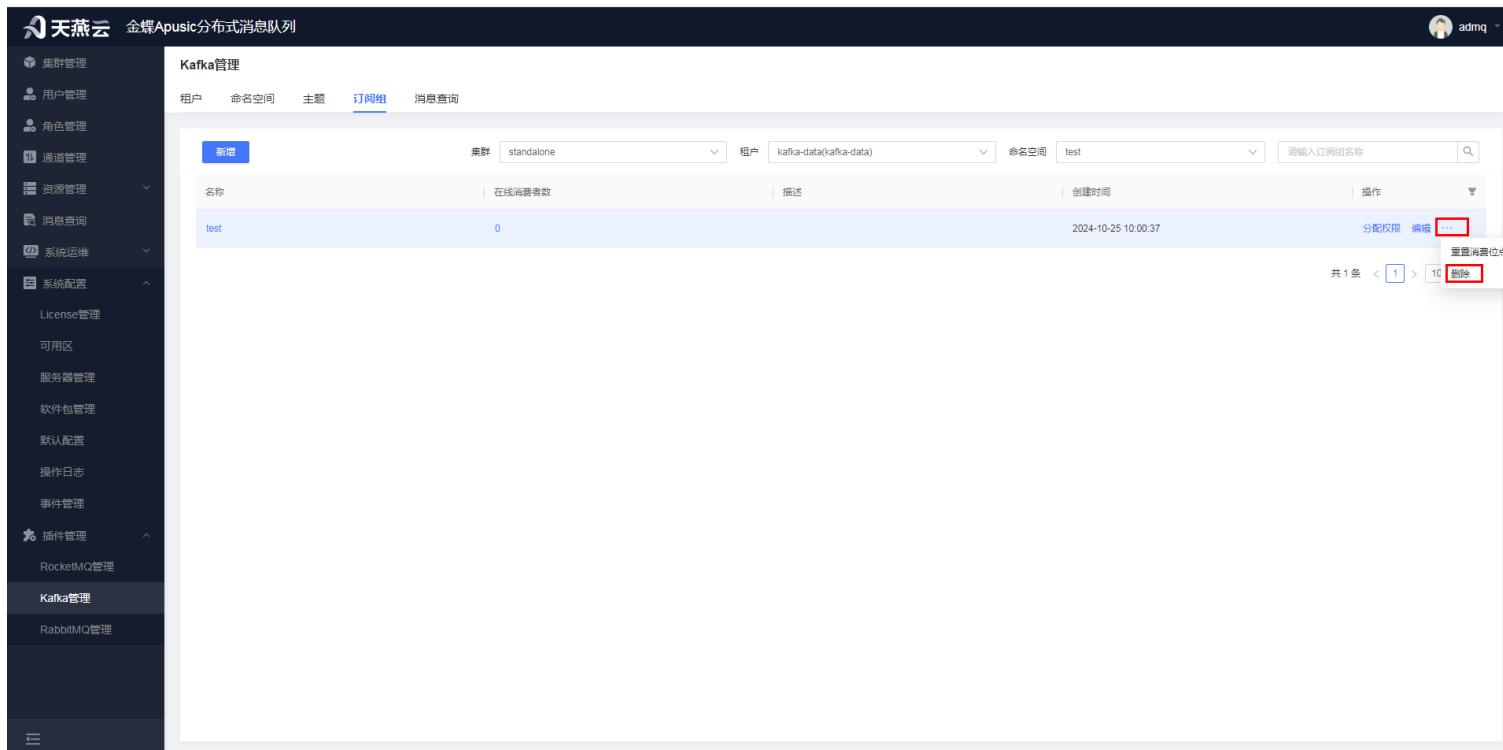
进入【插件管理】->【Kafka管理】，选择“订阅组”页签，选择要编辑的订阅组名称后的【编辑】按钮，可编辑订阅组信息。注意：编辑订阅组前，需满足集群服务可用。



- 租户名称：订阅组所属的租户名称，根据选择的租户自动带出，不可修改
- 租户ID：订阅组所属租户的租户ID，根据选择的租户自动带出，不可修改
- 命名空间：订阅组所属命名空间名称，根据选择的命名空间自动带出，不可修改
- 订阅组名称：订阅组的名称，置灰显示，不可修改
- 描述：订阅组的描述信息

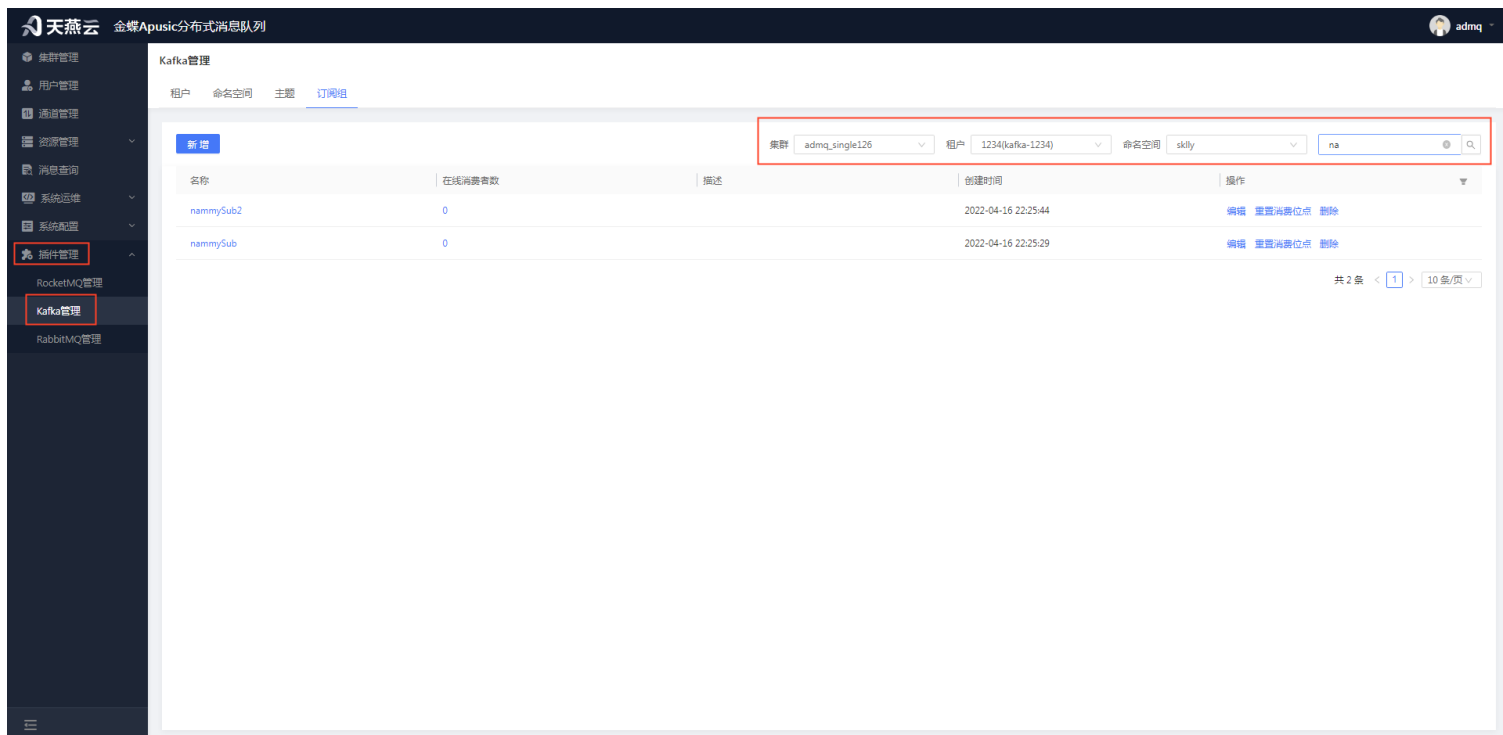
5.10.2.4.3 删除订阅组

进入【插件管理】->【Kafka管理】，选择“订阅组”页签，选择要删除的订阅组名称后的【...】-【删除】按钮，可删除订阅组信息。注意：删除订阅组前，需满足集群服务可用。



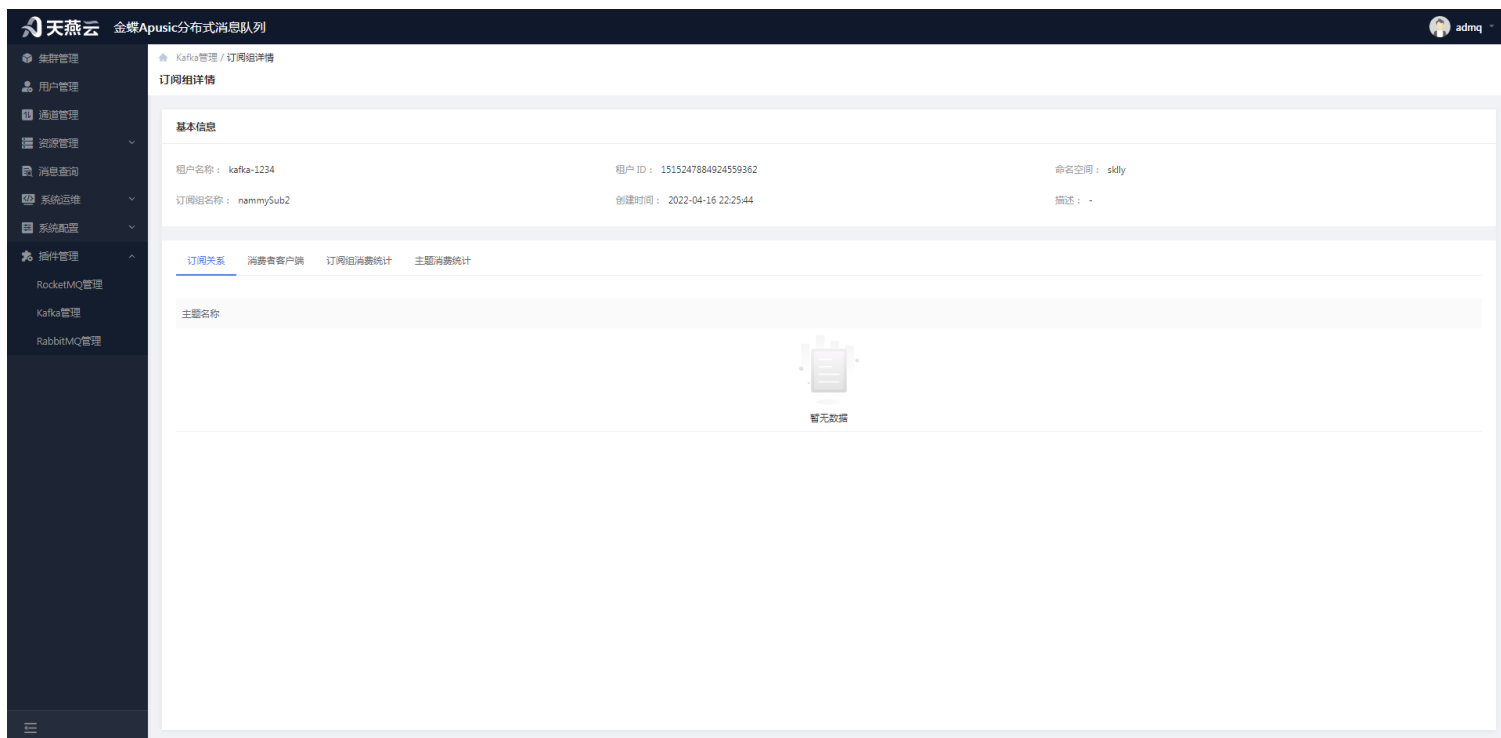
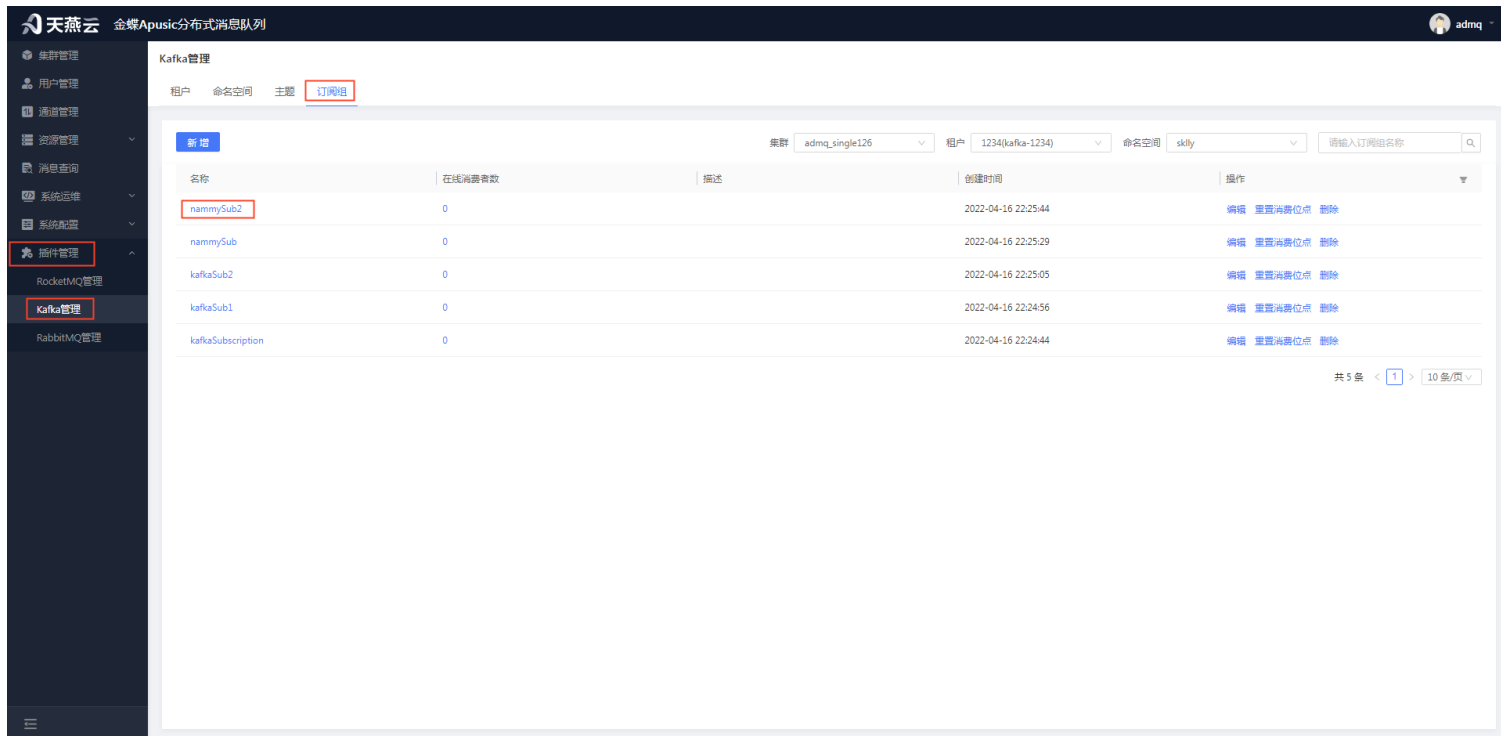
5.10.2.4.4 查询订阅组

进入【插件管理】->【Kafka管理】，选择“订阅组”页签，右上角选择集群名称、实例名称、命名空间名称、输入订阅组名称，可查询订阅组信息。



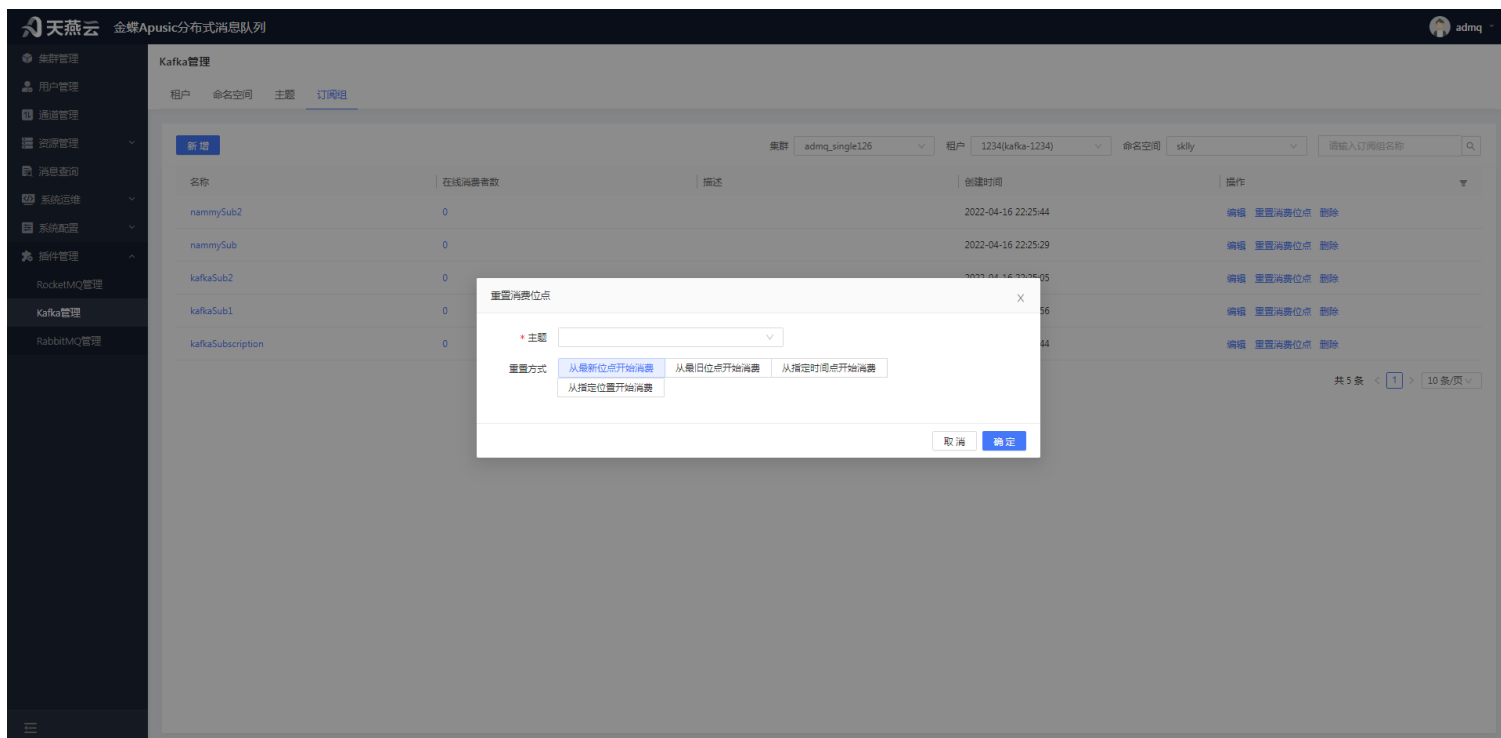
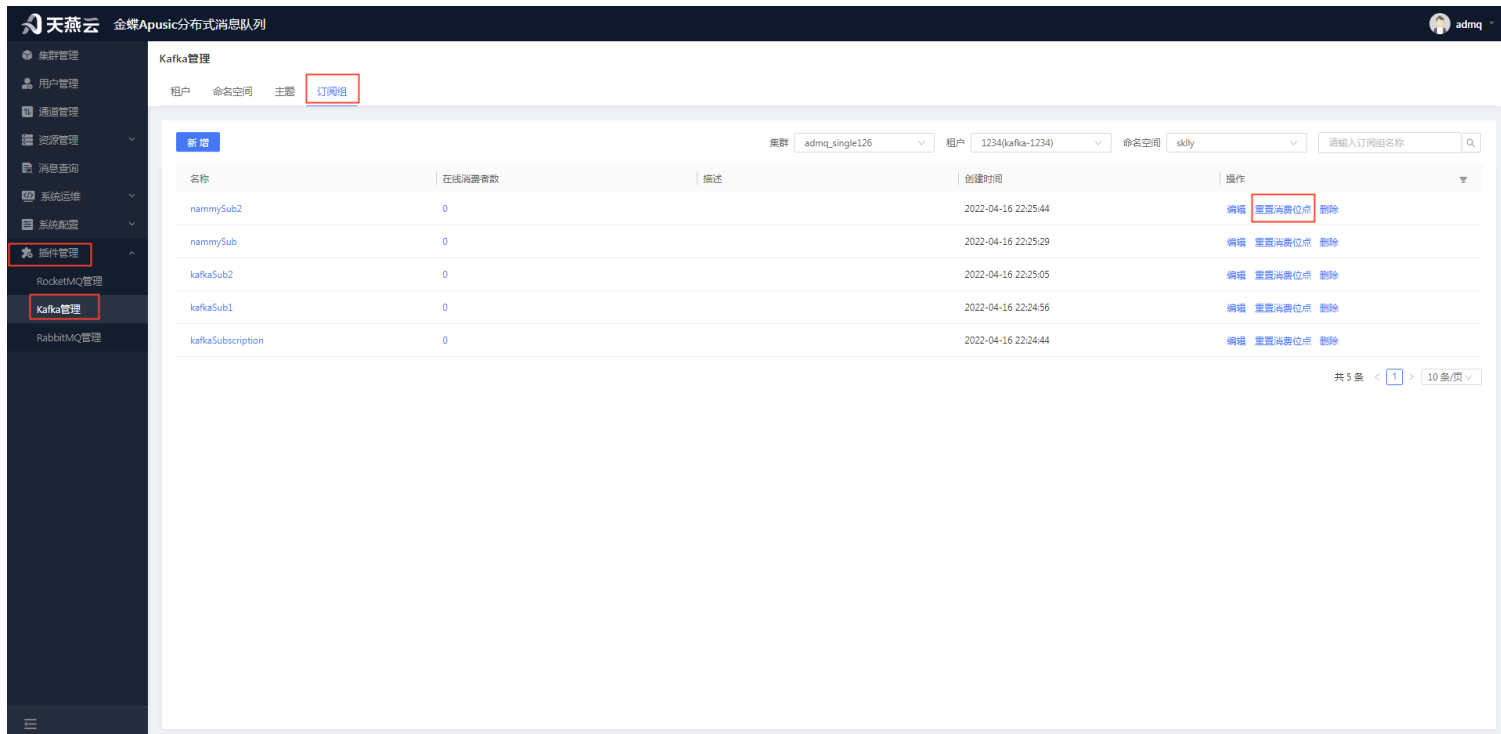
5.10.2.4.5 查看订阅组详情

进入【插件管理】->【Kafka管理】，选择“订阅组”页签，选择需要查看详细信息的订阅组，点击订阅组名称，可查看订阅组详细信息，如基本信息，订阅关系、生产者客户端、消费者客户端、订阅组生产消费统计、主题消费统计。



5.10.2.4.6 重置消费位点

进入【插件管理】->【Kafka管理】，选择“订阅组”页签，选择需要重置消费的订阅组，点击【重置消费位点】按钮，可重置消费。



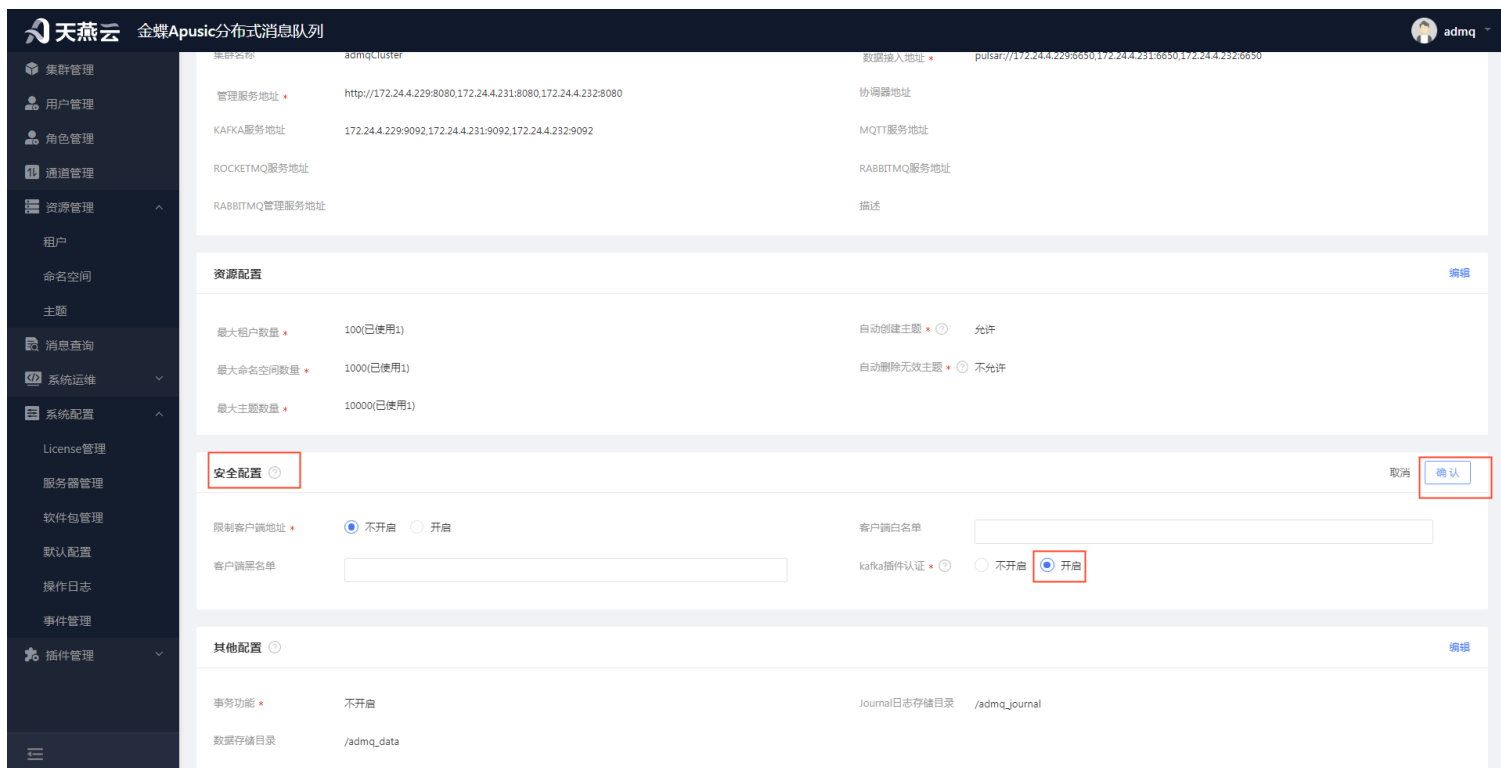
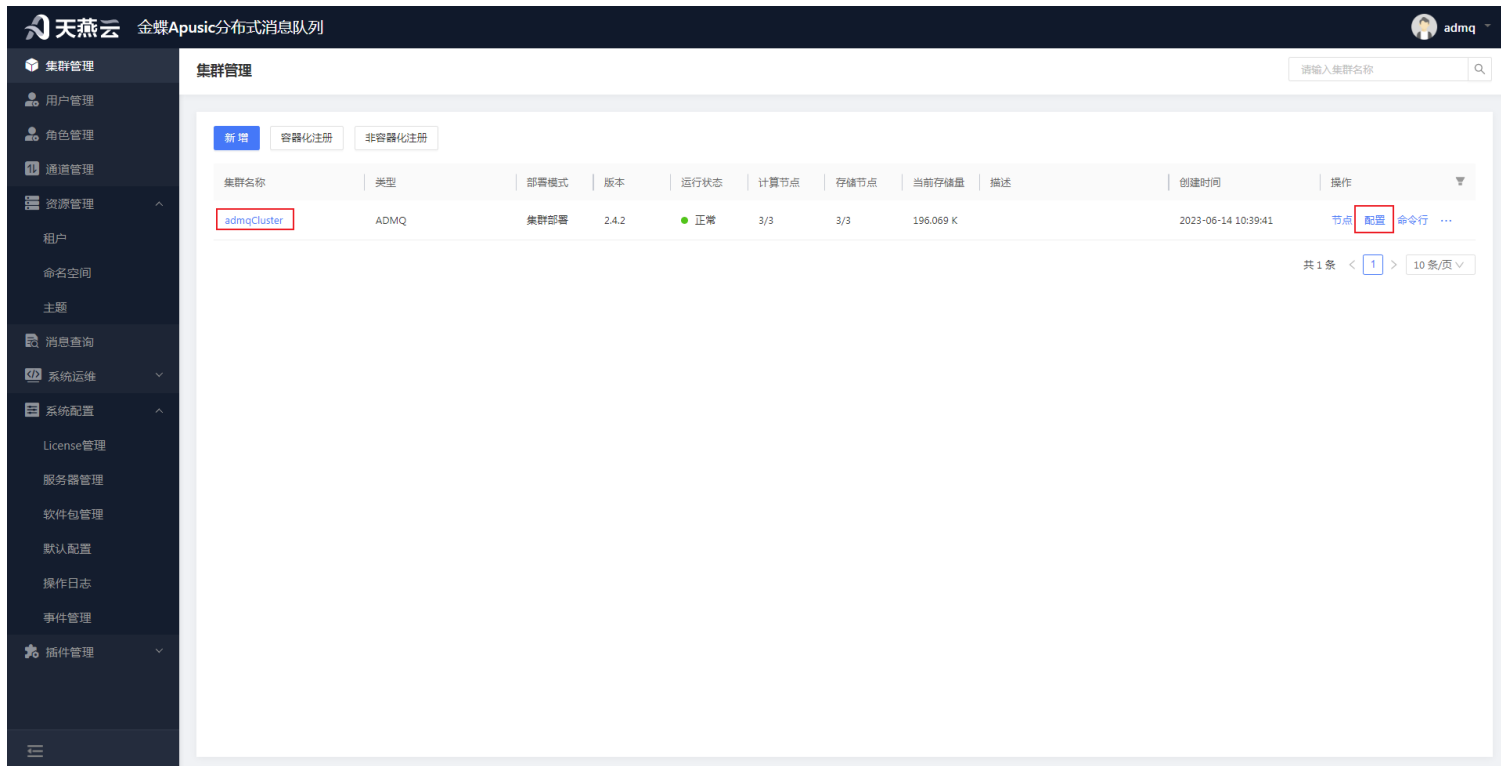
- 主题：重置消费的主题名，下拉选择（主题需发送和接收消息后才能被选择）
- 重置方式：重置消费的方式，单选按钮，从最新位点开始消费、从最旧位点开始消费、从指定时间点开始消费、从指定位置开始消费

5.10.2.5 Kafka认证

5.10.2.5.1 认证设置

加载Kafka插件之后，kafka插件认证默认不开启，开启Kafka认证的方式如下：

进入集群管理页面，在集群列表页面选择某一集群，在操作一栏中，点击【配置】按钮，进入到集群配置页面，在安全配置模块点击右上角【编辑】按钮，Kafka插件认证选择开启，然后【确认】，重启集群计算节点即可



5.10.2.5.2 SASL/SCRAM机制

ADMQ作为一款优秀的消息中间件，安全必然是很重要的一个特性，很多应用场景下我们需要控制生产、消费端、乃至IP进行相关的控制，操作粒度达到主题级别的消息的操作，ADMQ的Kop插件目前支持Kafka的SASL/SCRAM认证机制，是一种用于解决 PLAIN 机制安全问题的新机制。

具体操作过程：

- (1) 启动Kafka插件服务后，在集群的计算节点设置如下参数并重启计算节点：

```
saslAllowedMechanisms=SCRAM-SHA-256
// SCRAM-SHA-256, SCRAM-SHA-512
```

(2) 使用原生Kafka工具kafka-configs.sh来对用户进行相关操作:

创建用户:

```
bin/kafka-configs.sh --zookeeper 172.24.4.231:2181,172.24.4.232:2181,172.24.4.233:2181 --alter -
--add-config 'SCRAM-SHA-256=[password=producer-sec],SCRAM-SHA-512=[password=producer-sec]' --
entity-type users --entity-name producer
```

删除用户:

```
bin/kafka-configs.sh --zookeeper 172.24.4.231:2181,172.24.4.232:2181,172.24.4.233:2181 --alter -
--delete-config 'SCRAM-SHA-256,SCRAM-SHA-512' --entity-type users --entity-name producer
```

(3) 新增令牌并授权

在管控用户管理中新创建一个普通用户，为该普通用户创建一个令牌（该令牌名称需要与使用kafka-config.sh工具创建的用户名称一致），然后对相关资源（例如命名空间、主题）分配权限的时候使用该令牌对其授予权限。

(4) 使用Kafka客户端发送消息

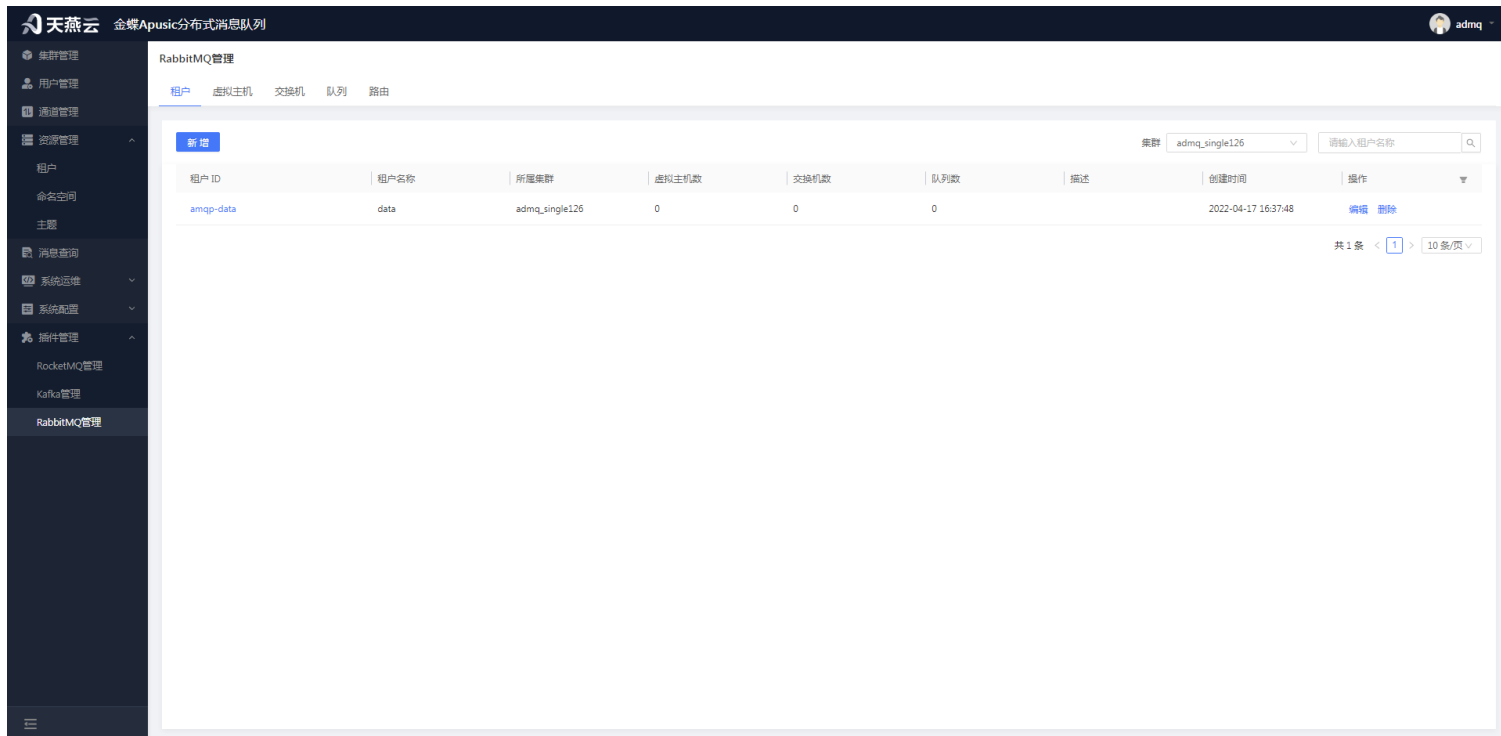
客户端代码中需要加入如下设置:

```
security.protocol=SASL_PLAINTEXT
sasl.mechanism=SCRAM-SHA-256 // 可以设置成SCRAM-SHA-256 、 SCRAM-SHA-512
sasl.jaas.config=org.apache.kafka.common.security.scram.ScramLoginModule required
username="producer" password="producer-sec";
```

5.10.3 RabbitMQ管理

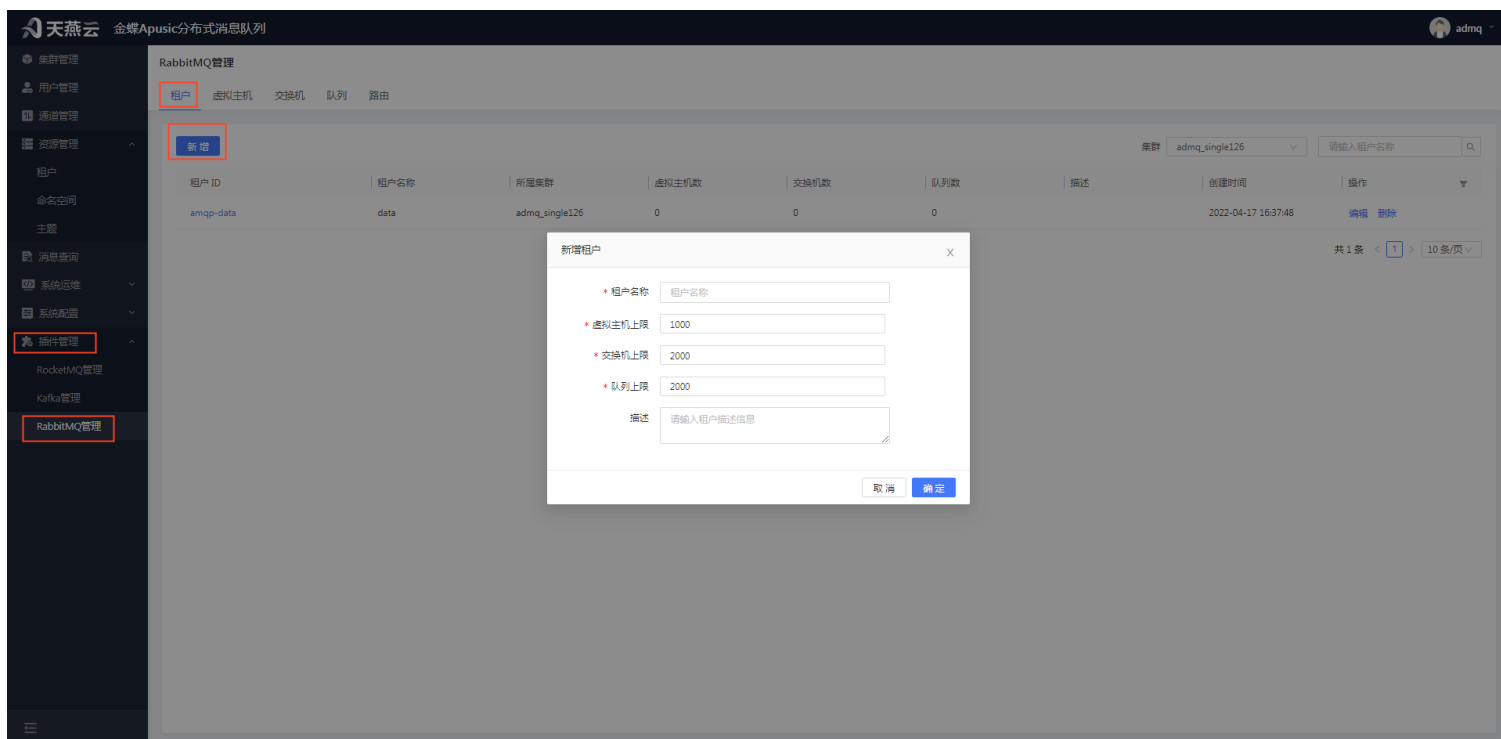
5.10.3.1 租户

租户是管理存储配额、隔离策略的基本单元，使用租户的目的是为了在多用户环境下，使用同一套程序，且保证用户间数据隔离。租户用于新增租户、编辑租户、删除租户、查询租户。



5.10.3.1.1 新增租户

进入【插件管理】->【RabbitMQ管理】，选择“租户”页签，点击左上角【新增】按钮，可新增租户。注意：新增租户前，需满足集群服务可用；RabbitMQ的租户ID需要和standalone.conf配置文件中amqpTenant的值保持一致。

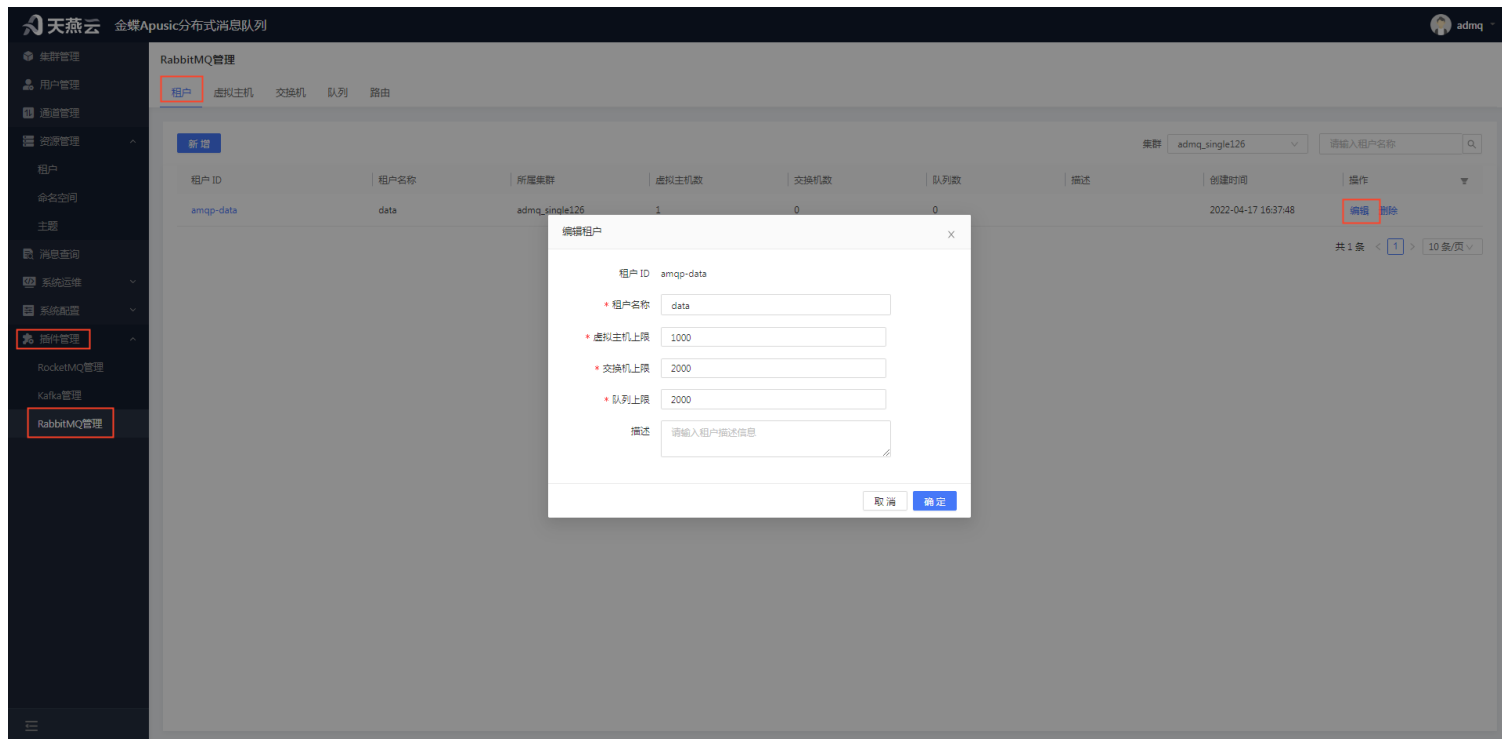


- 租户名称：租户的名称，自定义，唯一标识
- 虚拟主机上限：租户下可新增的虚拟主机上限，默认为1000，可更改
- 交换机上限：虚拟主机下可新增的交换机数量，默认为2000，可更改
- 队列上限：虚拟主机下可新增的队列数量，默认为2000，可更改

- 描述：租户描述信息
- 注意：租户名称只是为了方便记，真正的租户是租户ID。

5.10.3.1.2 编辑租户

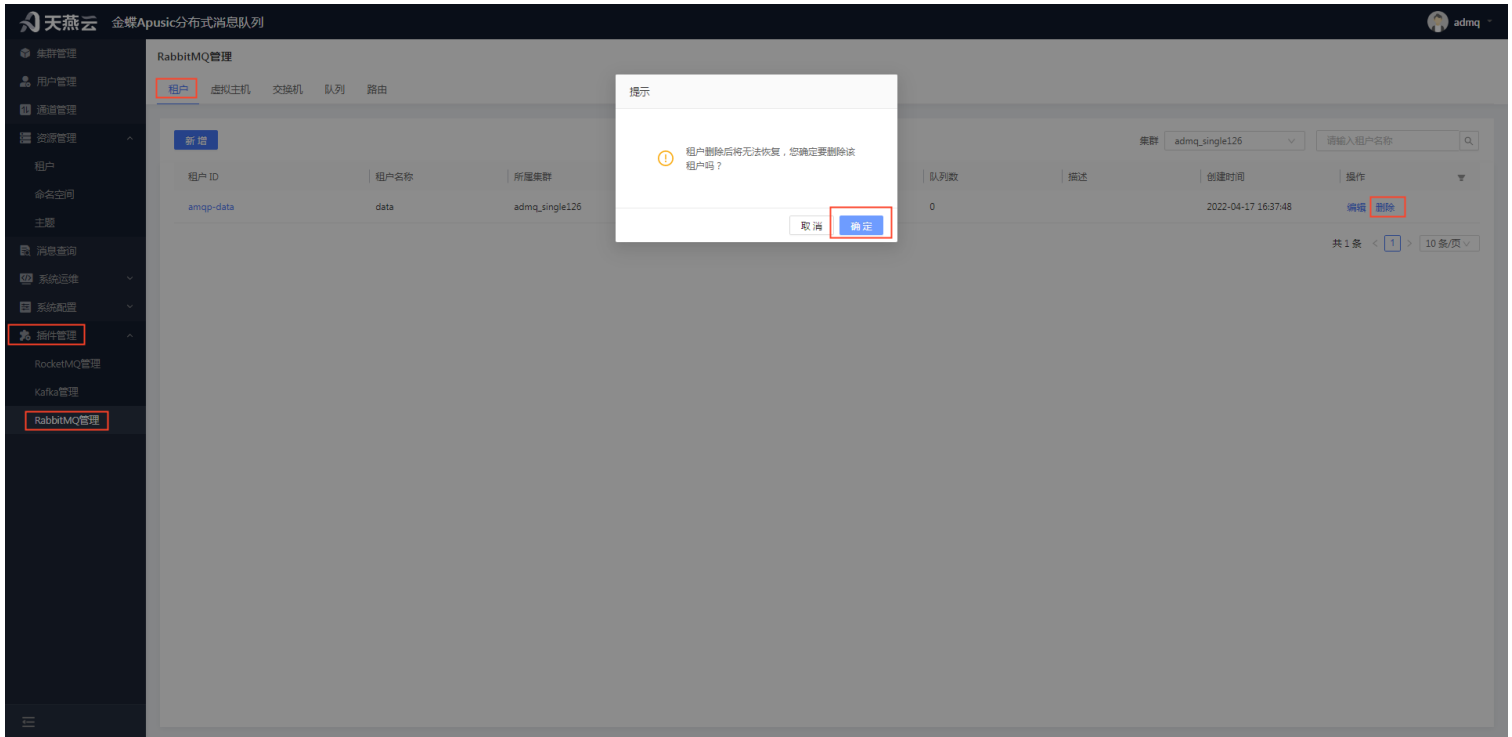
进入【插件管理】->【RabbitMQ管理】，选择“租户”页签，选择租户列表中某一个租户名称后的【编辑】按钮，可编辑租户信息。注意：编辑租户前，需满足集群服务可用。



- 租户ID：租户的ID，根据租户名称自动生成的，不可修改。
- 租户名称：租户的名称，可修改
- 虚拟主机上限：租户下可新增的虚拟主机上限，可更改
- 交换机上限：虚拟主机下可新增的交换机数量，可更改
- 队列上限：虚拟主机下可新增的队列数量，可更改
- 描述：租户描述信息，可更改

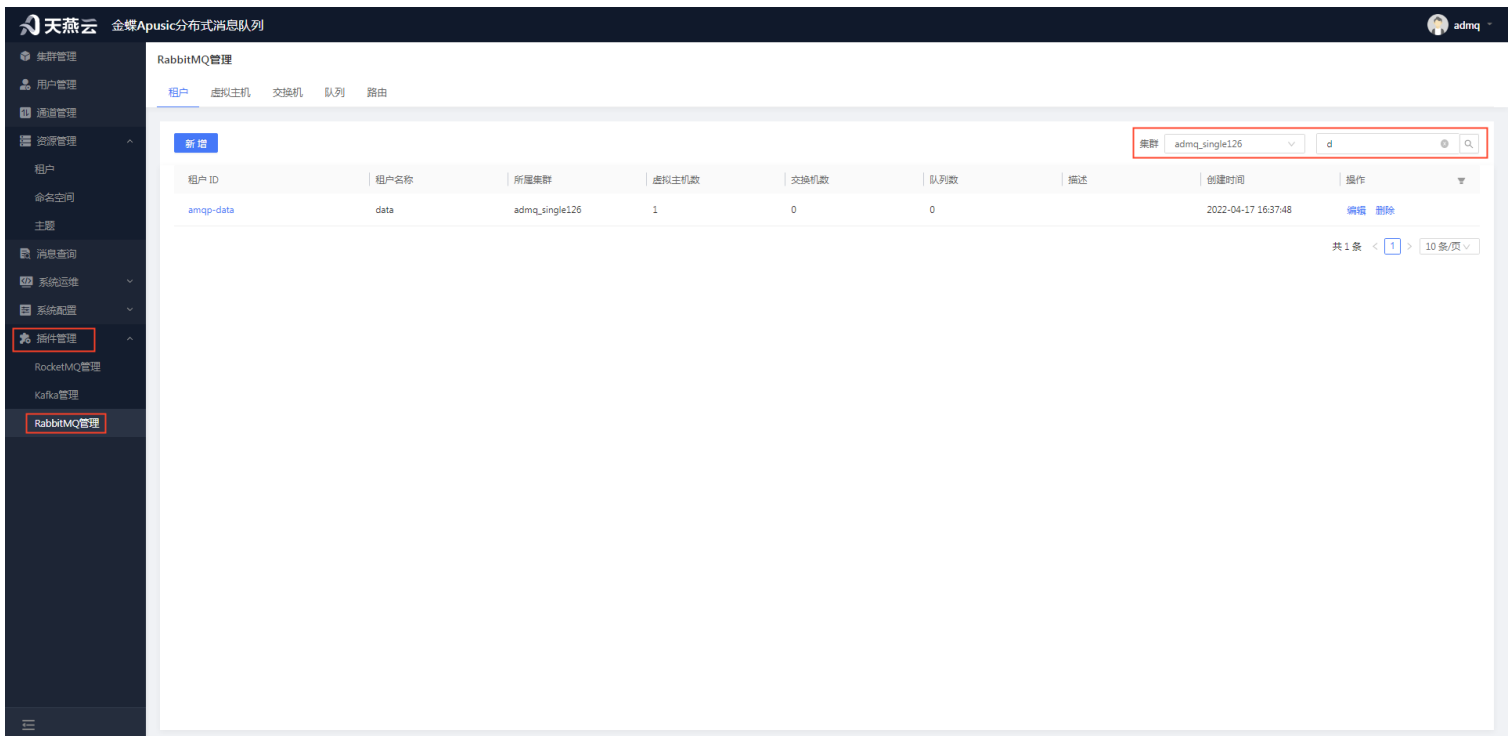
5.10.3.1.3 删除租户

进入【插件管理】->【RabbitMQ管理】，选择“租户”页签，选择租户列表中某一个租户名称后的【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除租户信息。注意：删除租户前，需满足集群服务可用，且租户下无虚拟主机，否则不可删除。



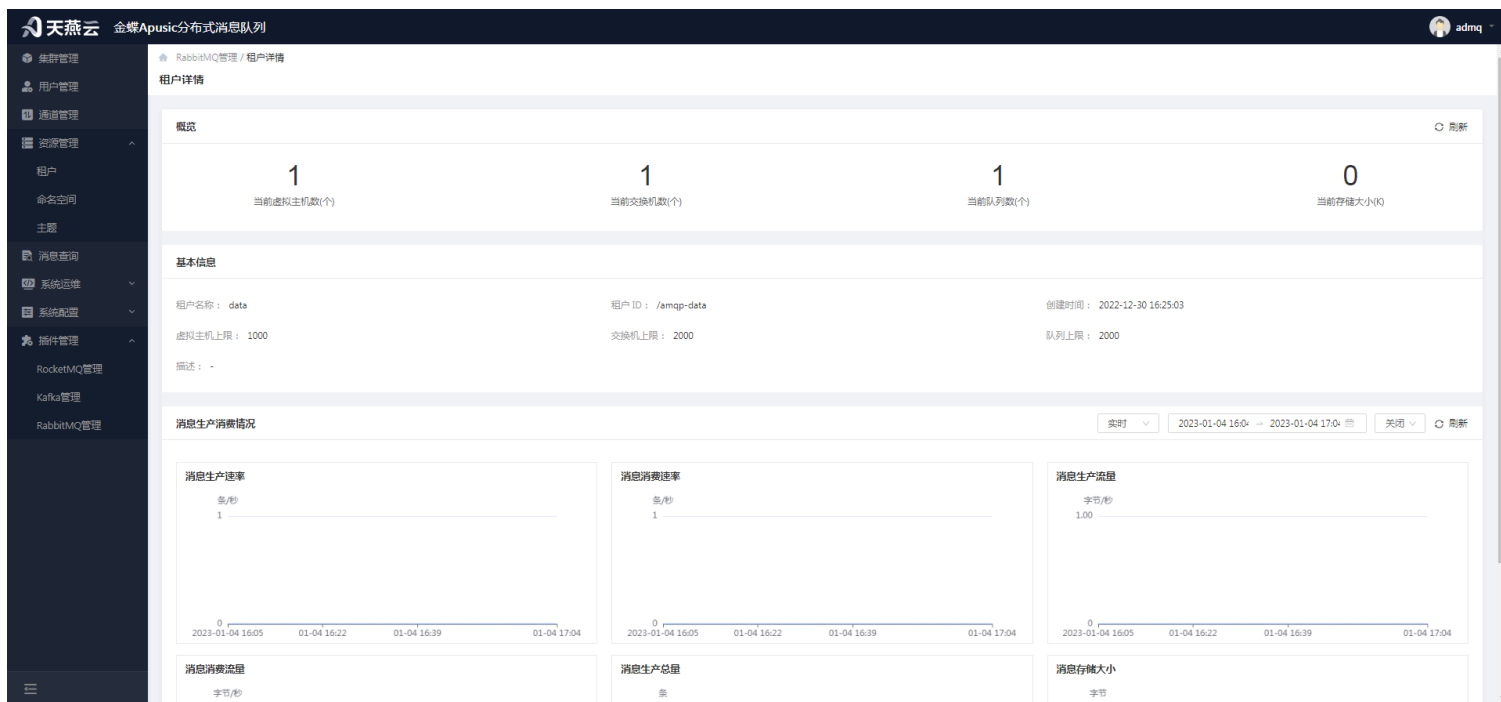
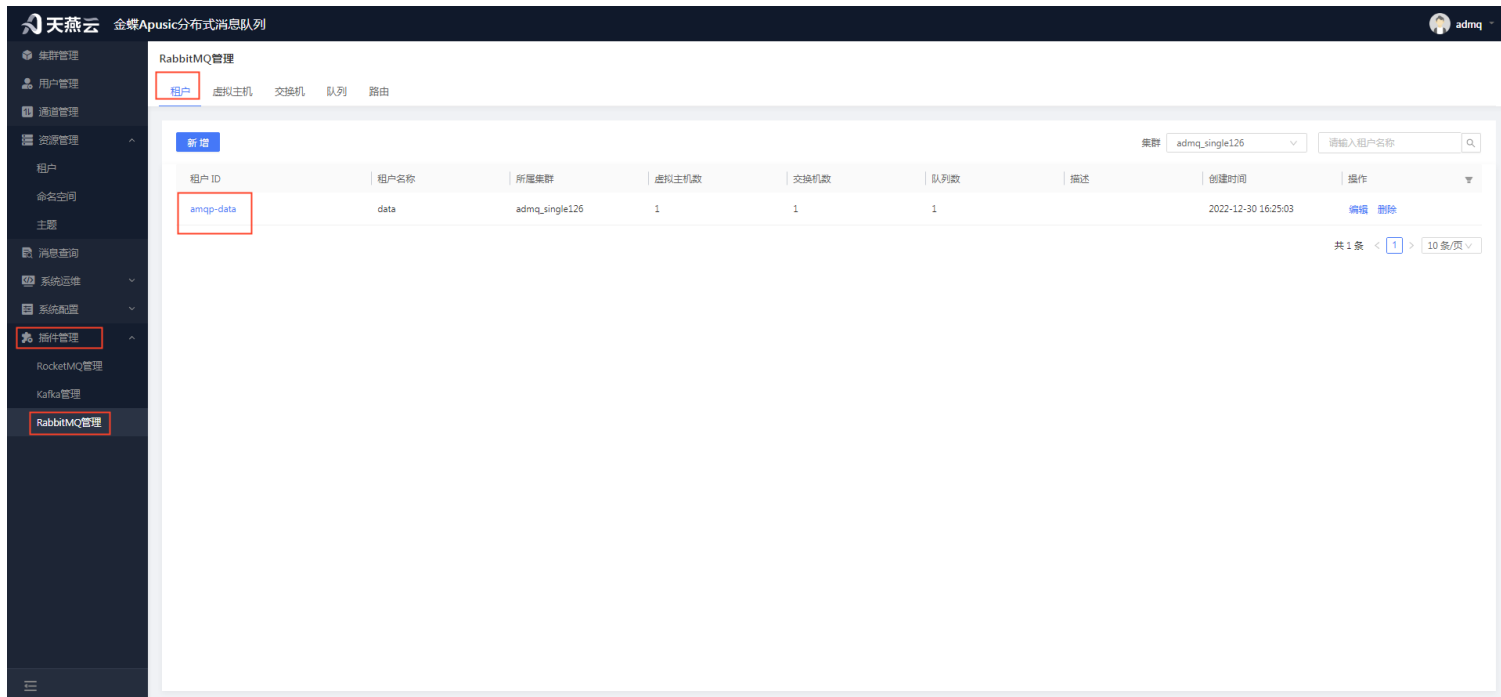
5.10.3.1.4 查询租户

进入【插件管理】->【RabbitMQ管理】，选择“租户”页签，右上角选择集群名称、输入租户名称、可查询租户信息。



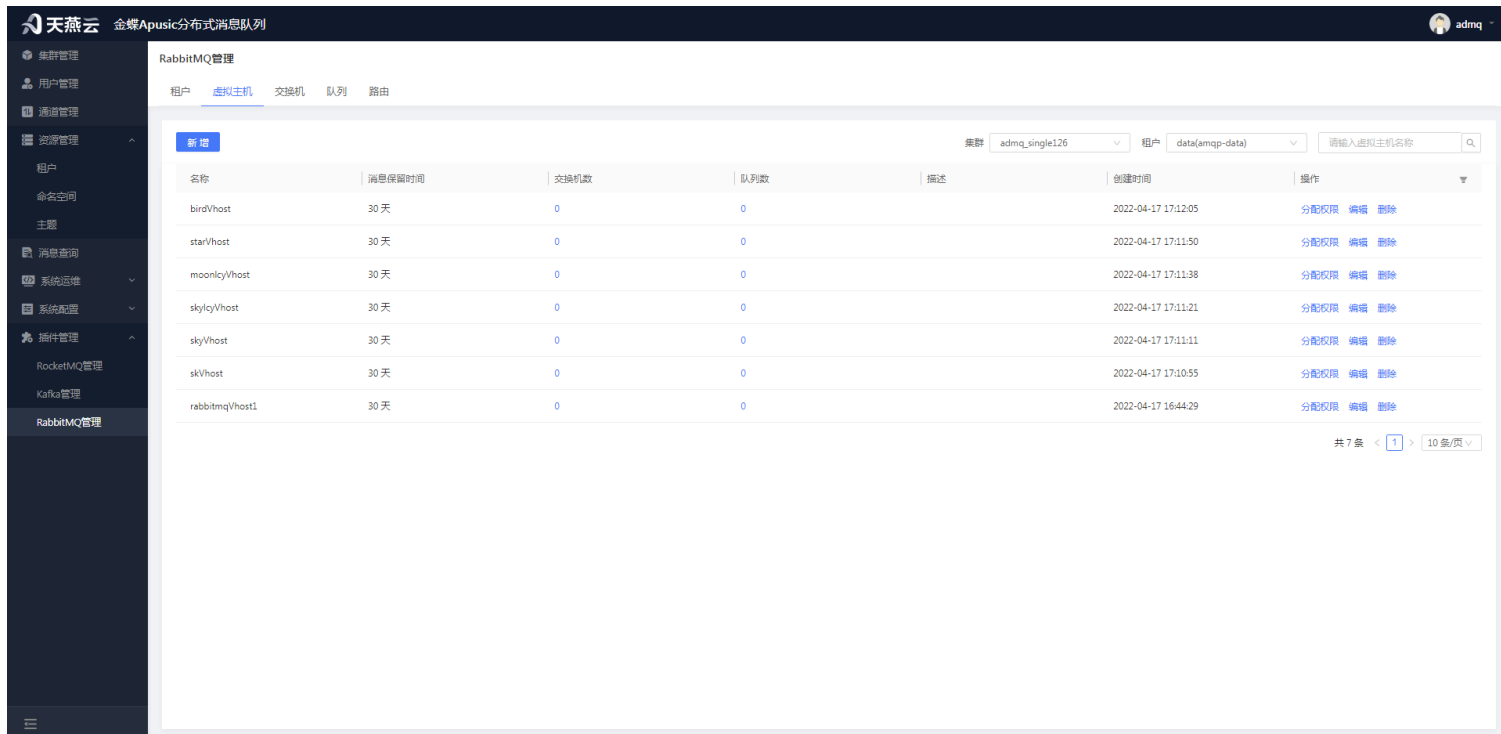
5.10.3.1.5 查看租户详情

进入【插件管理】->【RabbitMQ管理】，选择“租户”页签，选择需要查看详细信息的租户，点击租户ID，可查看租户详细信息。



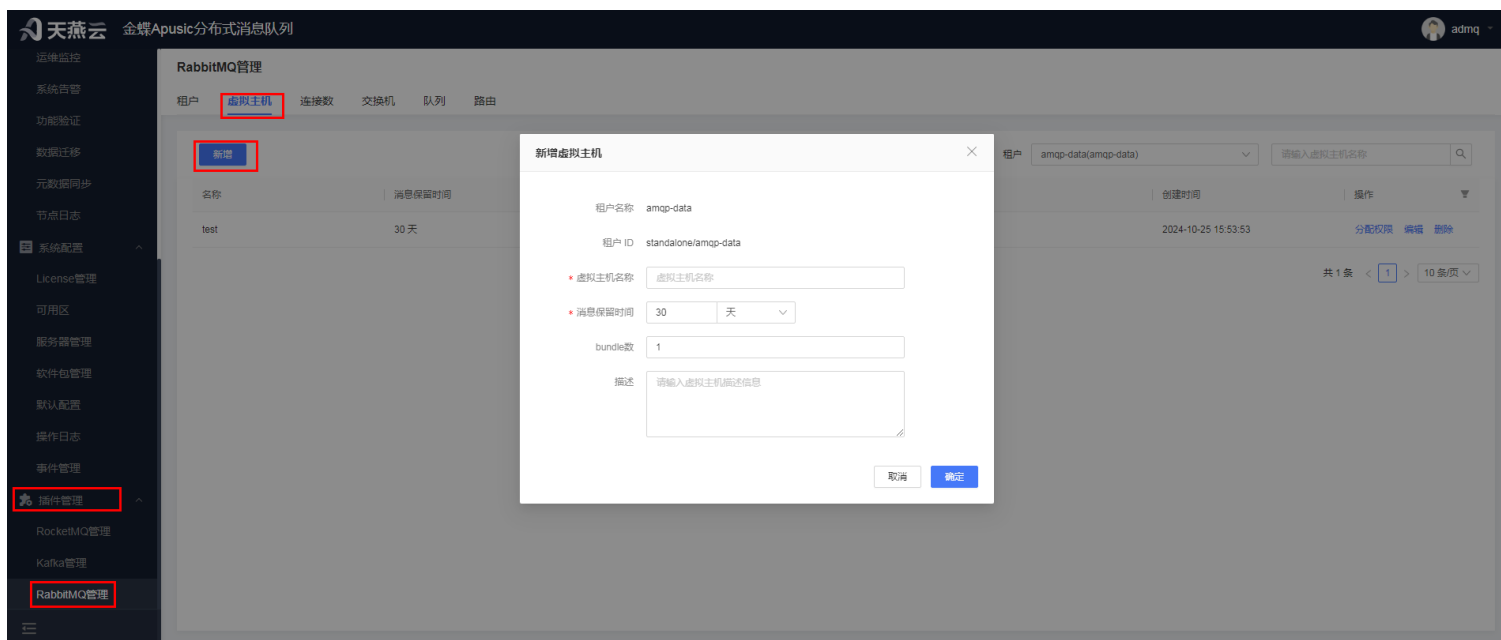
5.10.3.2 虚拟主机

虚拟主机是共享相同的身份认证和加密环境的独立服务器域，拥有自己的交换机、队列，绑定和权限机制。虚拟主机用于新增虚拟主机、编辑虚拟主机、删除虚拟主机、查询虚拟主机、新增虚拟主机权限、编辑虚拟主机权限、删除虚拟主机权限。



5.10.3.2.1 新增虚拟主机

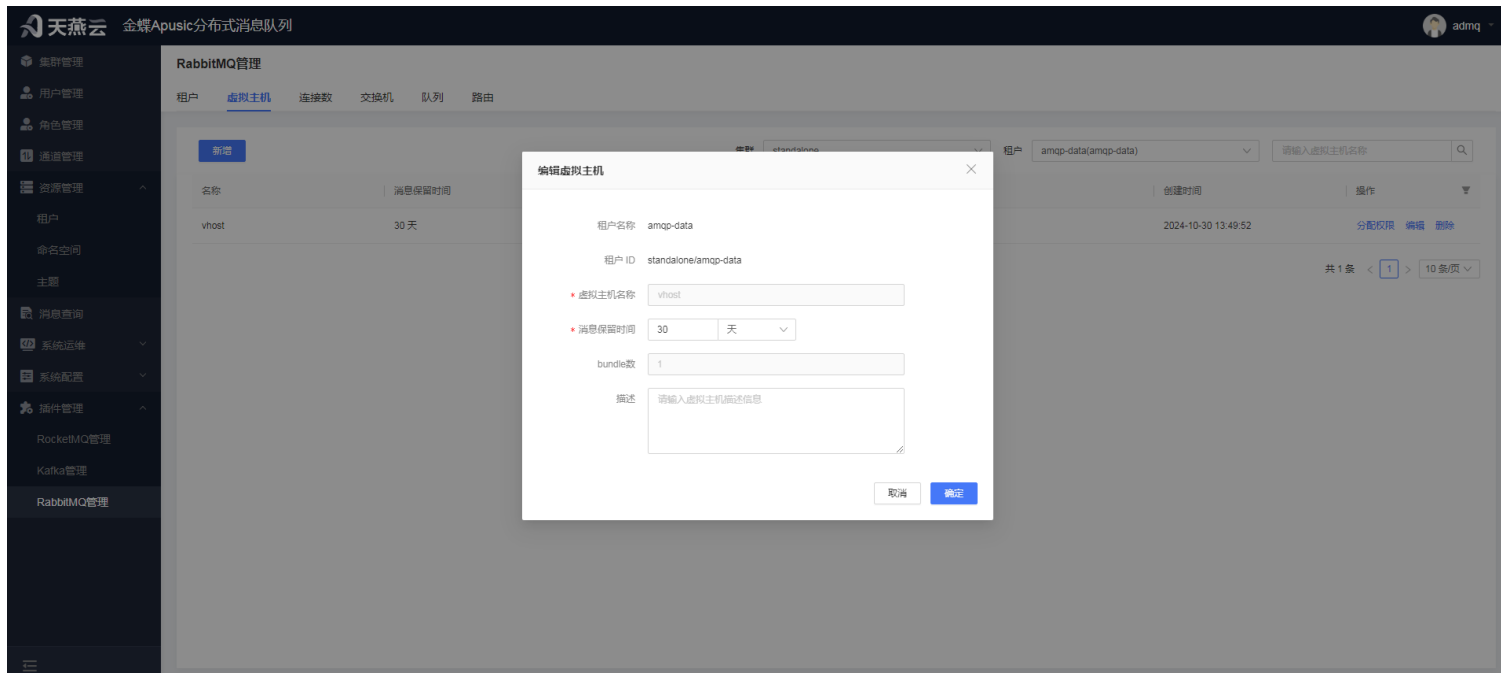
进入【插件管理】->【RabbitMQ管理】，选择“虚拟主机”页签，点击左上角【新增】按钮，可新增虚拟主机。注意：新增虚拟主机前，需满足集群服务可用。



- 租户名称：虚拟主机所属的租户名称，根据选择的租户自动带出，不可修改
- 租户ID：虚拟主机所属租户的租户ID，根据选择的租户自动带出，不可修改
- 虚拟主机名称：虚拟主机的名称，自定义，唯一标识
- 消息保留时间：消息的最大保留时间，超过则不保存消息，单位（分钟）、（小时）、（天），默认30天（最大值）
- bundle数：bundle数量
- 描述：虚拟主机的描述信息

5.10.3.2.2 编辑虚拟主机

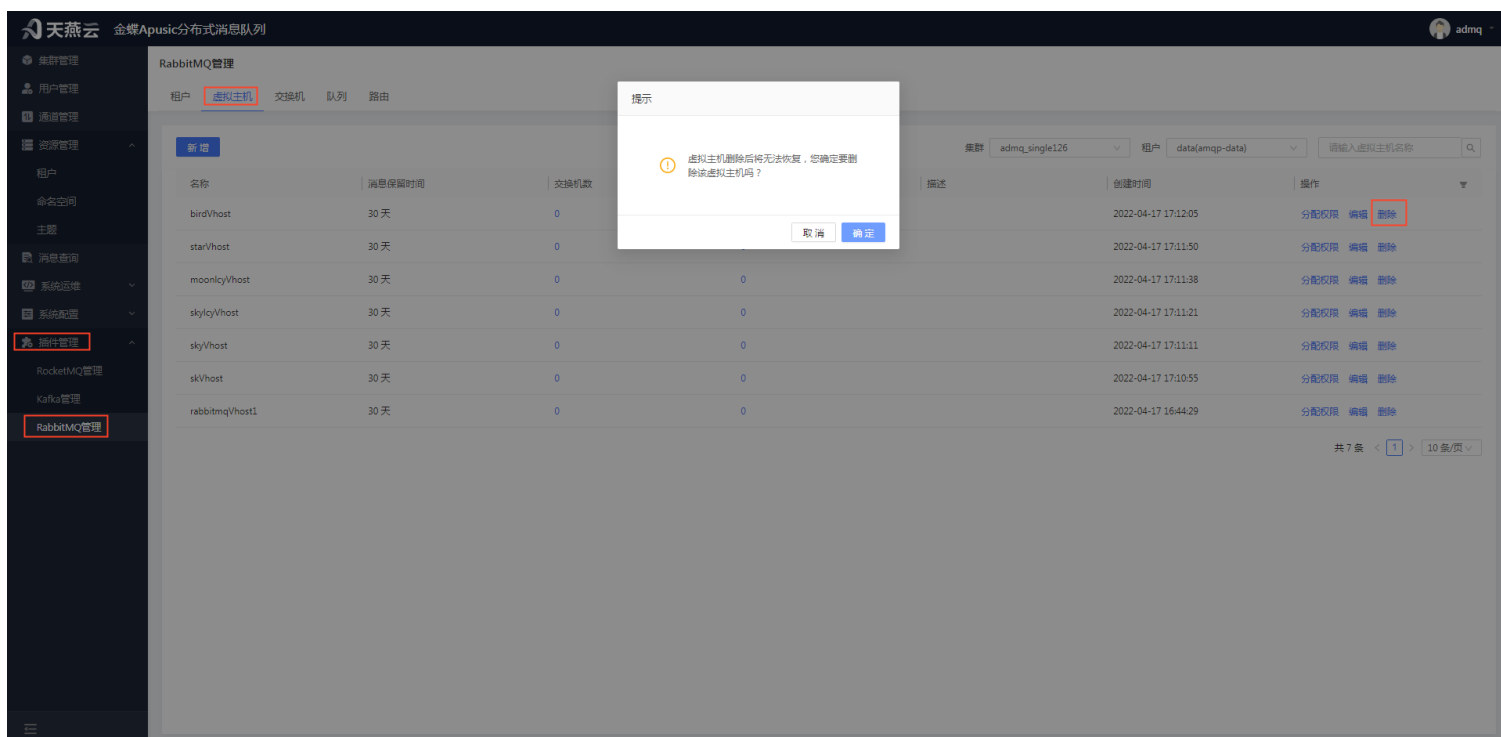
进入【插件管理】->【RabbitMQ管理】，选择“虚拟主机”页签，选择要编辑的虚拟主机名称后的【编辑】按钮，可编辑虚拟主机信息。注意：编辑虚拟主机前，需满足集群服务可用。



- 租户名称：虚拟主机所属的租户名称，根据选择的租户自动带出，不可修改
- 租户ID：虚拟主机所属租户的租户ID，根据选择的租户自动带出，不可修改
- 虚拟主机名称：虚拟主机的名称，置灰显示，不可修改
- 消息保留时间：消息的最大保留时间，超过则不保存消息，单位（分钟）、（小时）、（天），可修改
- bundle数：bundle数量，不可修改
- 描述：命名空间的描述信息，可修改

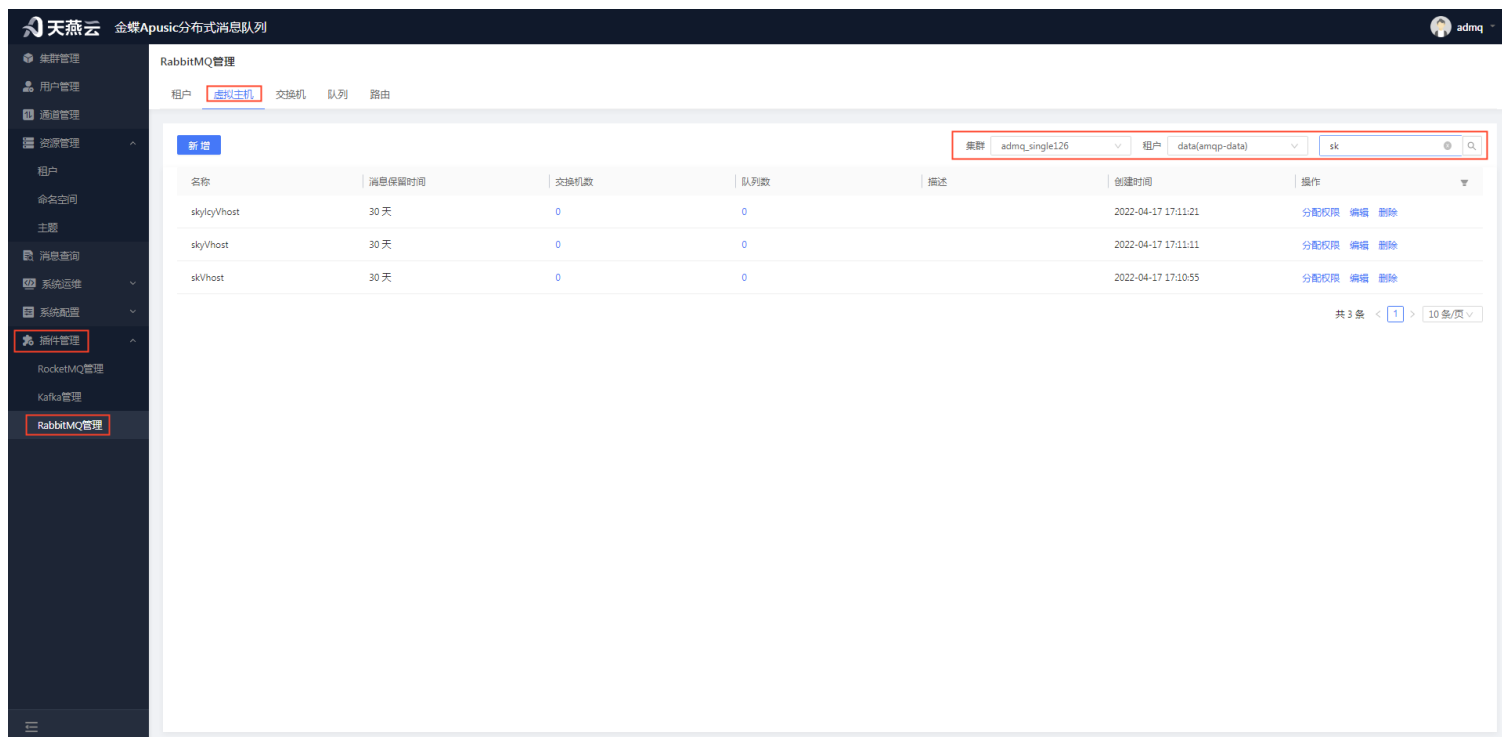
5.10.3.2.3 删除虚拟主机

进入【插件管理】->【RabbitMQ管理】，选择“虚拟主机”页签，选择要编辑的虚拟主机名称后的【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除虚拟主机信息。注意：删除虚拟主机前，需满足集群服务可用，且虚拟主机下无交换机、队列，删除虚拟主机后，虚拟主机下的交换机、队列信息也被删除。



5.10.3.2.4 查询虚拟主机

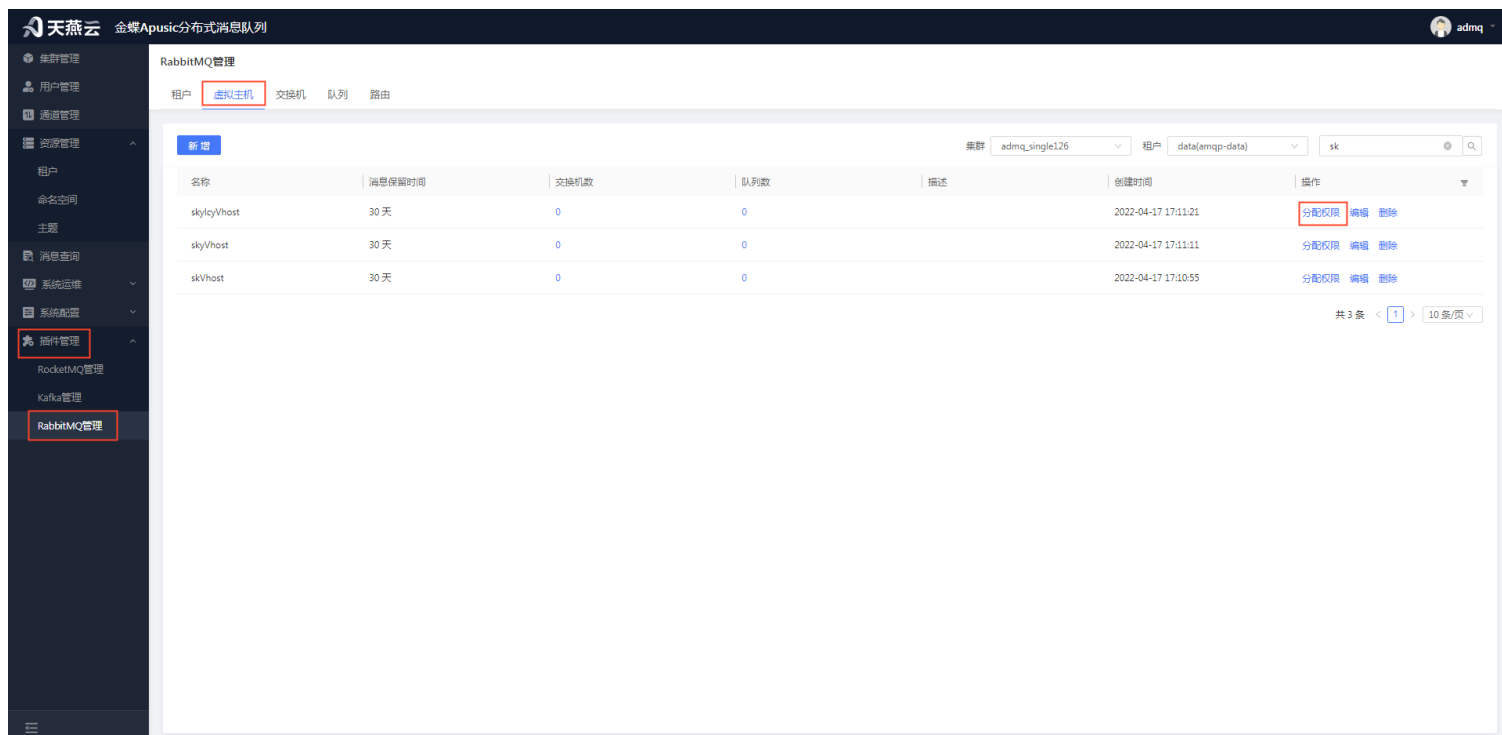
进入【插件管理】->【RabbitMQ管理】，选择“虚拟主机”页签，右上角选择集群名称、租户名称、输入虚拟主机名称，可查询虚拟主机信息。（注意：包含的也有）



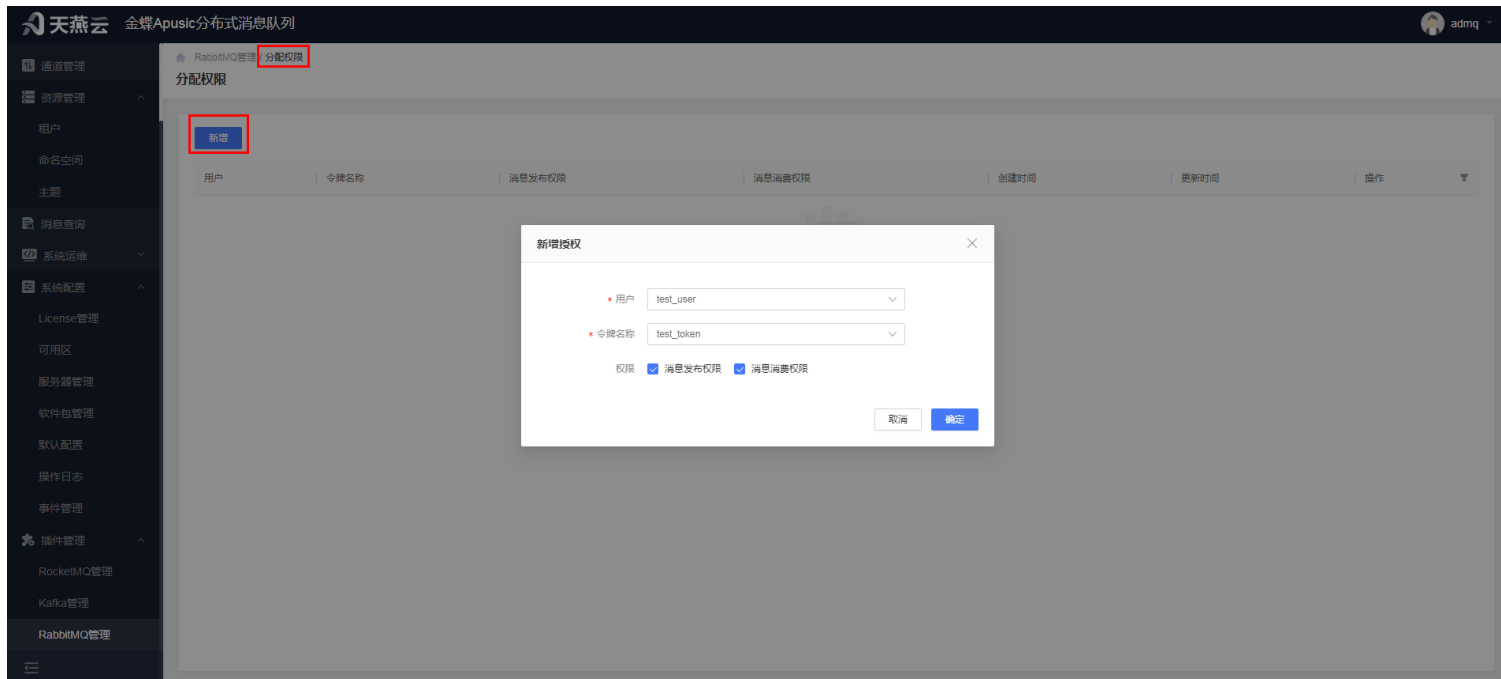
5.10.3.2.5 发布消费权限

5.10.3.2.5.1 新增权限

进入【插件管理】->【RabbitMQ管理】，选择“虚拟主机”页签，选择要新增授权的虚拟主机名称后的【分配权限】按钮后，会进入到分配权限页面。



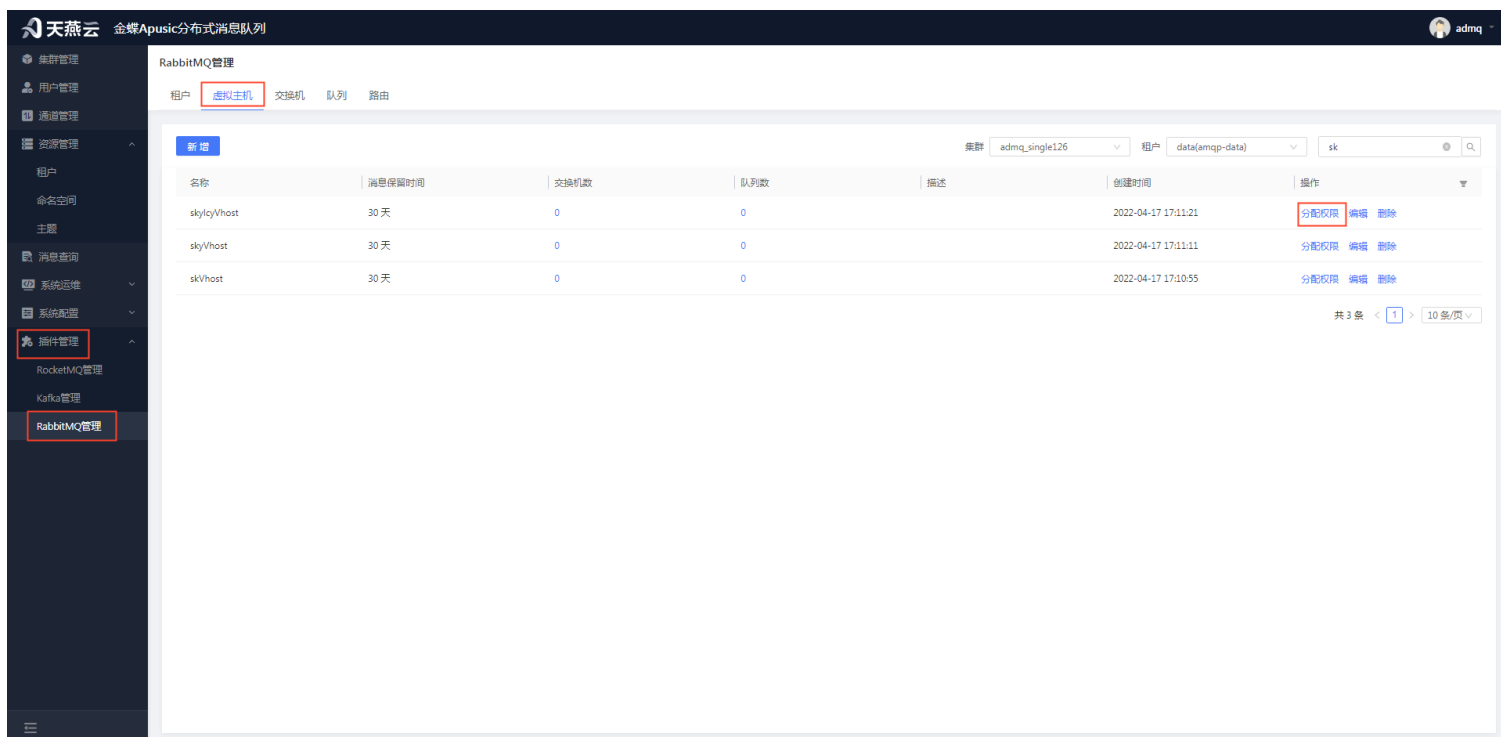
在分配权限页面，点击左上角的【新增】按钮，可新增命名空间下用户的权限信息。注意：新增权限前，需满足集群服务可用；授权主要是给普通用户提供授权，默认超级管理员拥有所有命名空间的消息发布权限和消息消费权限。



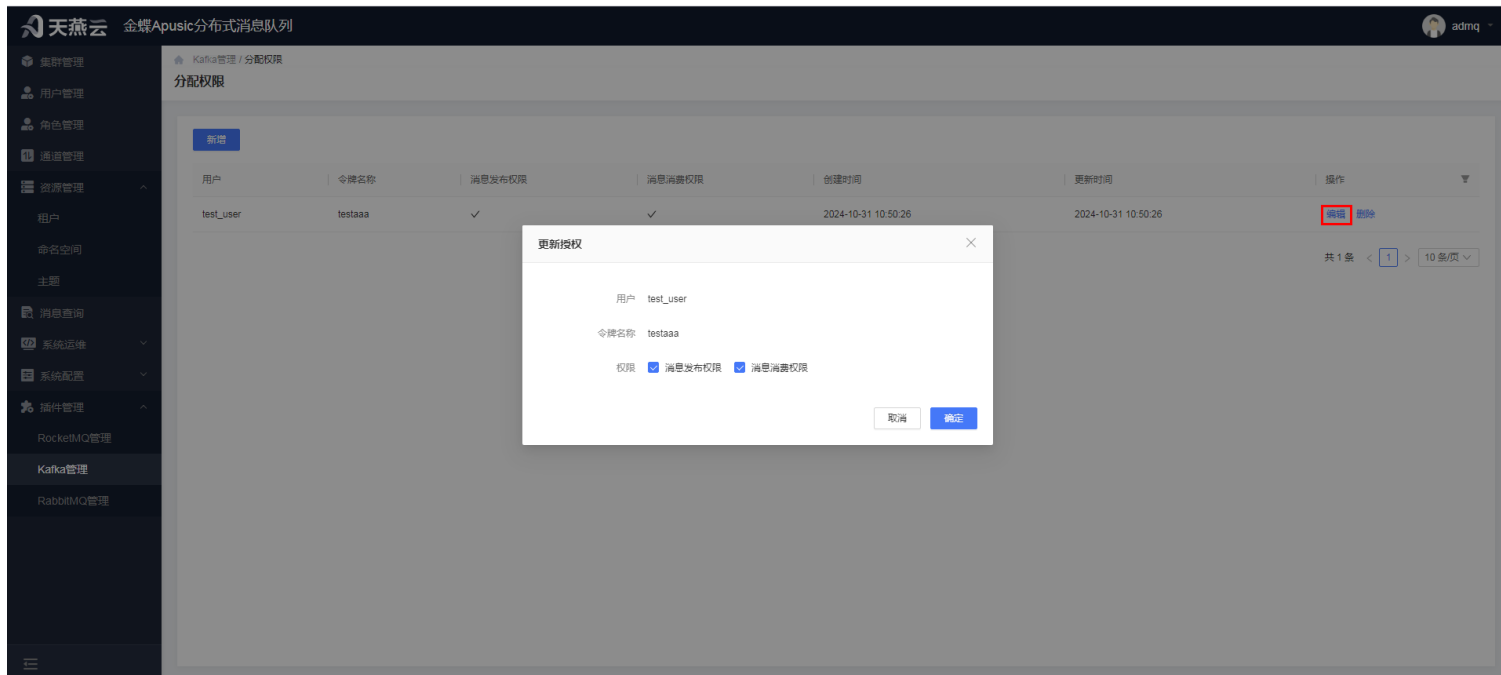
- 用户：下拉选择，获取的用户管理（4.1）的用户信息
- 令牌名称：下拉选择，获取选择用户的令牌
- 权限：复选框，消息发布权限和消息消费权限

5.10.3.2.5.2 编辑权限

进入【插件管理】->【RabbitMQ管理】，选择“虚拟主机”页签，选择要更新授权的虚拟主机名称后的【分配权限】按钮，跳转到分配权限页面。



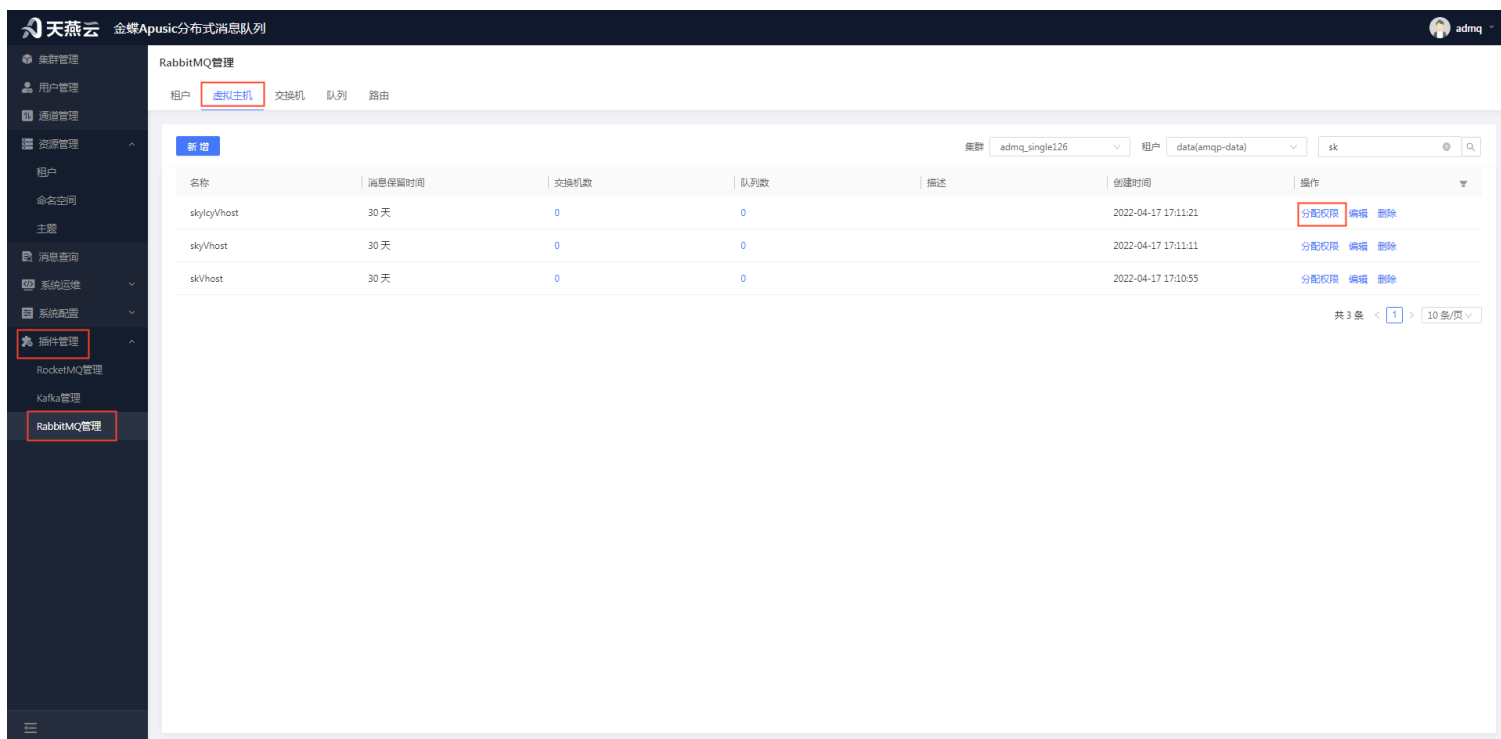
在分配权限页面，选择需要更新授权的用户，点击【编辑】按钮，可更新用户在命名空间下的权限。注意：更新权限前，需满足集群服务可用。



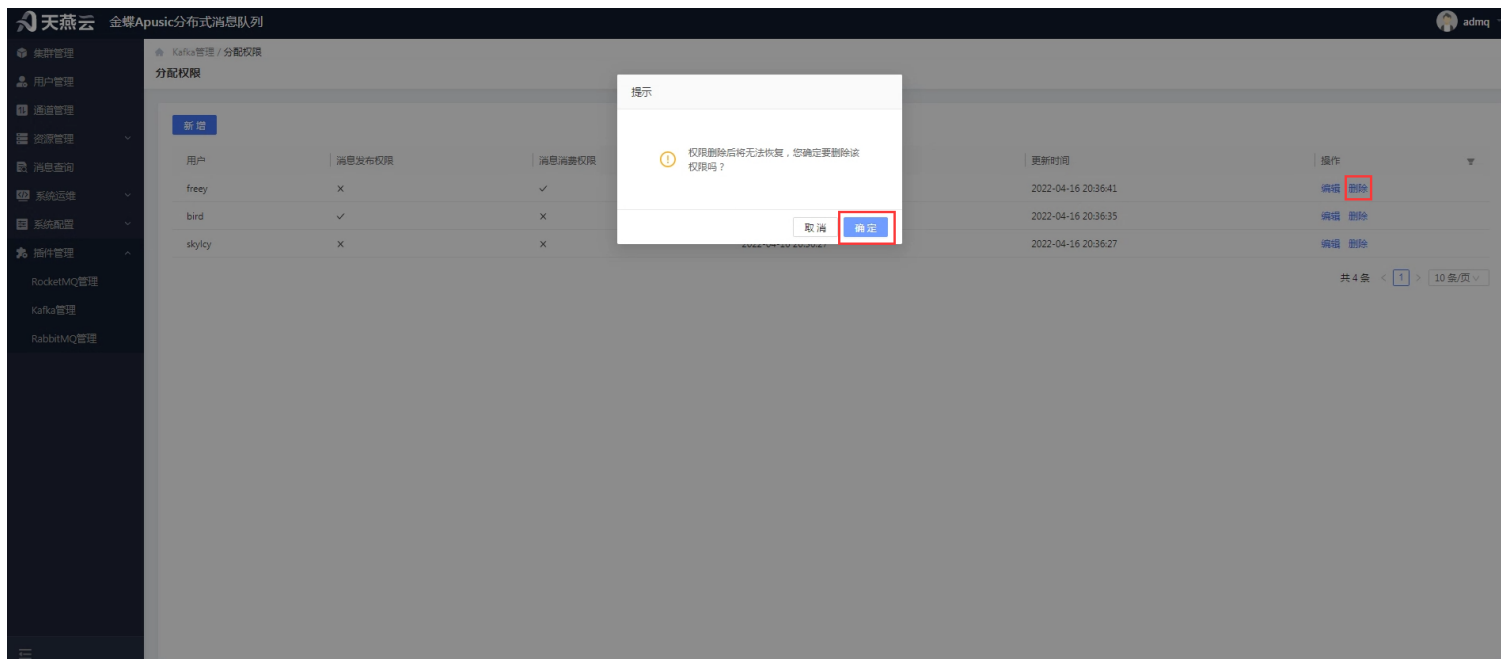
- 用户：用户名称，不可修改
- 令牌名称：令牌名称，不可修改
- 权限：复选框，消息发布权限和消息消费权限，可修改

5.10.3.2.5.3 删除权限

进入【插件管理】->【RabbitMQ管理】，选择“虚拟主机”页签，选择要删除授权的虚拟主机名称后的【分配权限】按钮，进入到分配权限页面。

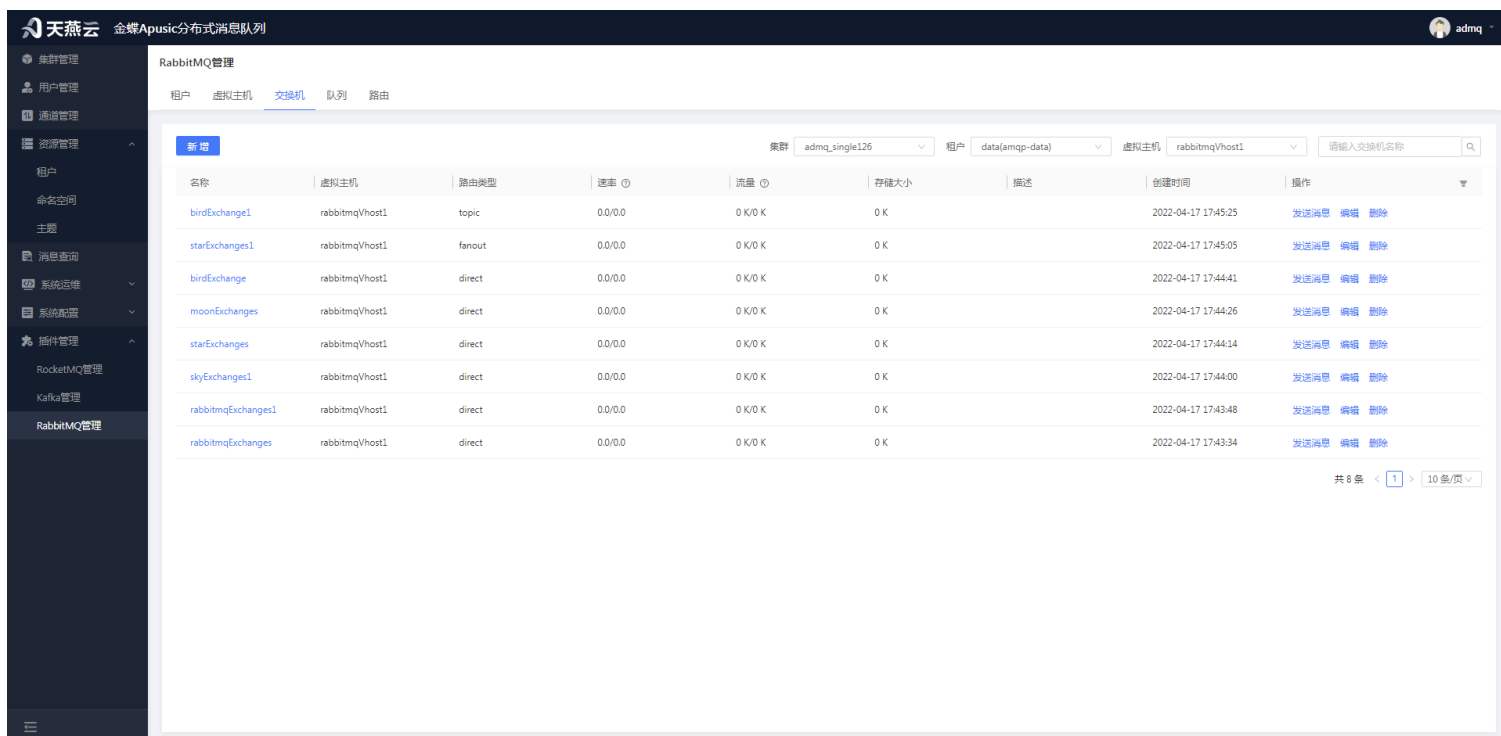


在分配权限页面，选择需要删除授权的用户，点击【删除】按钮，在弹出的确认删除弹框点击【确定】按钮，可删除用户在虚拟主机下的权限。注意：删除权限前，需满足集群服务可用。



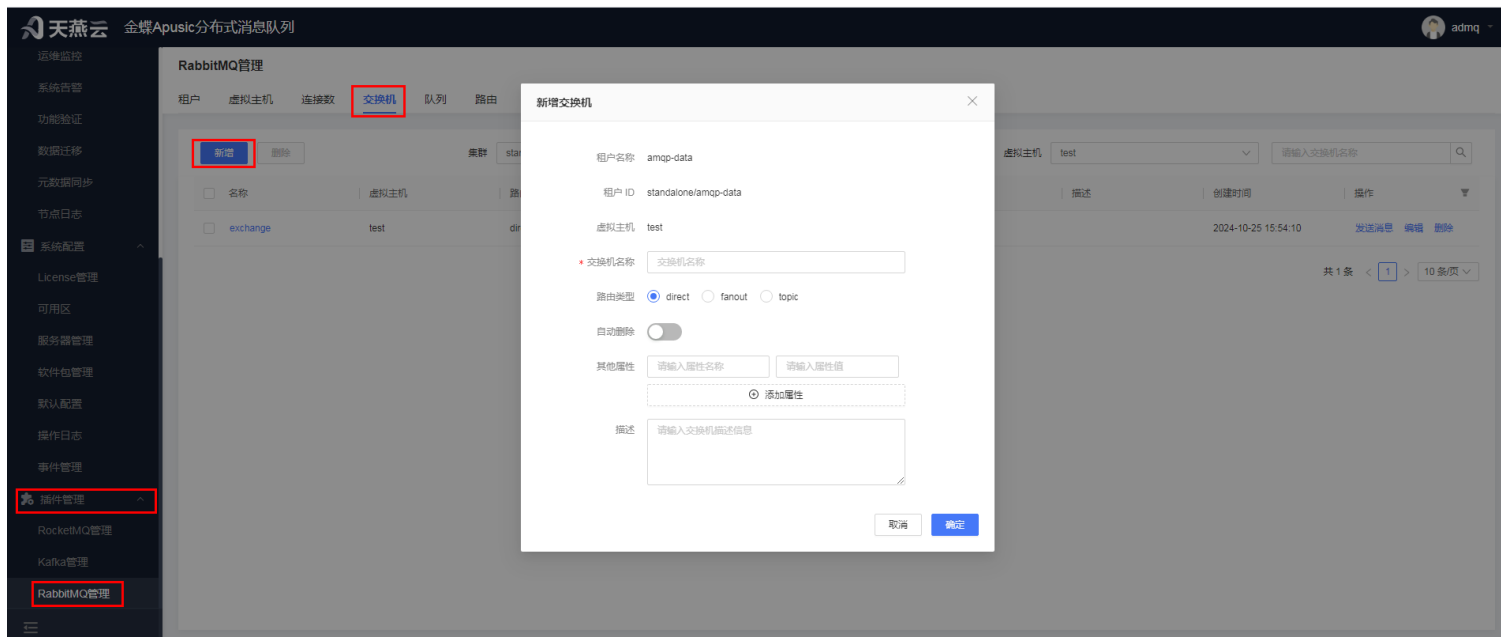
5.10.3.3 交换机

交换机作用于管理交换机资源，用来接收生产者发送的消息并将这些消息路由给服务器中的队列。交换机用于新增交换机、编辑交换机、删除交换机、查询交换机、发送消息。



5.10.3.3.1 新增交换机

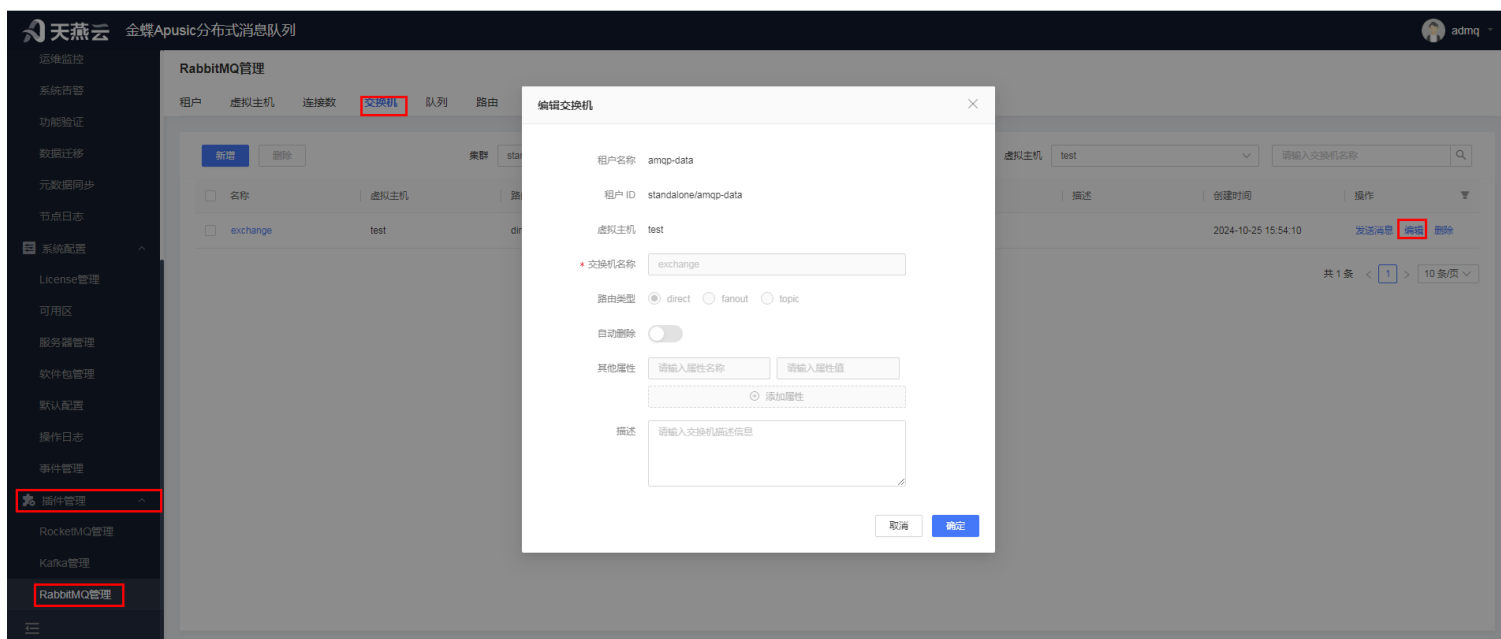
进入【插件管理】->【RabbitMQ管理】，选择“交换机”页签，点击左上角【新增】按钮，可新增交换机信息。注意：新增交换机前，需满足集群服务可用。



- 租户名称：虚拟主机所属的租户的名称，根据选择的租户自动带出，不可修改
- 租户ID：虚拟主机所属租户的租户ID，根据选择的租户自动带出，不可修改
- 虚拟主机：交换机所属虚拟主机名称，根据选择的虚拟主机自动带出，不可修改
- 交换机名称：交换机的名称，自定义，唯一标识
- 路由类型：单选框，交换机分发消息时的不同分发策略（匹配规则），目前共三种类型：direct、fanout、topic，可修改
- 自动删除：是否自动删除，对应autoDelete参数
- 其他属性：设置交换机的其他属性
- 描述：交换机的描述信息

5.10.3.3.2 编辑交换机

进入【插件管理】->【RabbitMQ管理】，选择"交换机"页签，选择要编辑的交换机名称后的【编辑】按钮，可编辑交换机信息。注意：编辑交换机前，需满足集群服务可用。

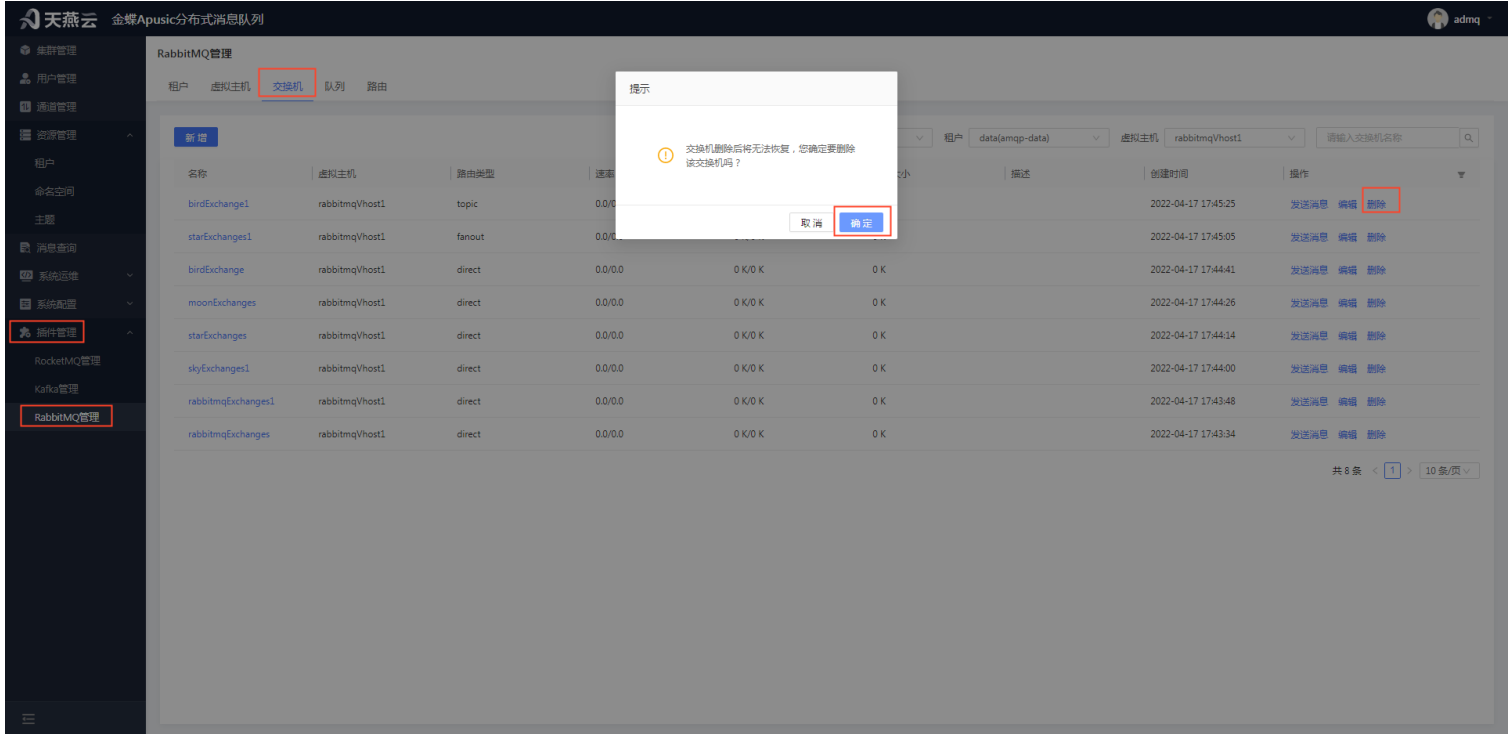


- 租户名称：虚拟主机所属租户名称，根据选择的租户自动带出，不可修改
- 租户ID：虚拟主机所属租户的租户ID，根据选择的租户自动带出，不可修改
- 虚拟主机：交换机所属虚拟主机名称，根据选择的虚拟主机自动带出，不可修改

- 交换机名称：交换机的名称，置灰显示，不可修改
- 路由类型：新建交换机的时候选择的路由类型，置灰显示，不可修改
- 自动删除：是否自动删除，对应autoDelete参数，置灰显示，不可修改
- 其他属性：设置交换机的其他属性
- 描述：交换机的描述信息

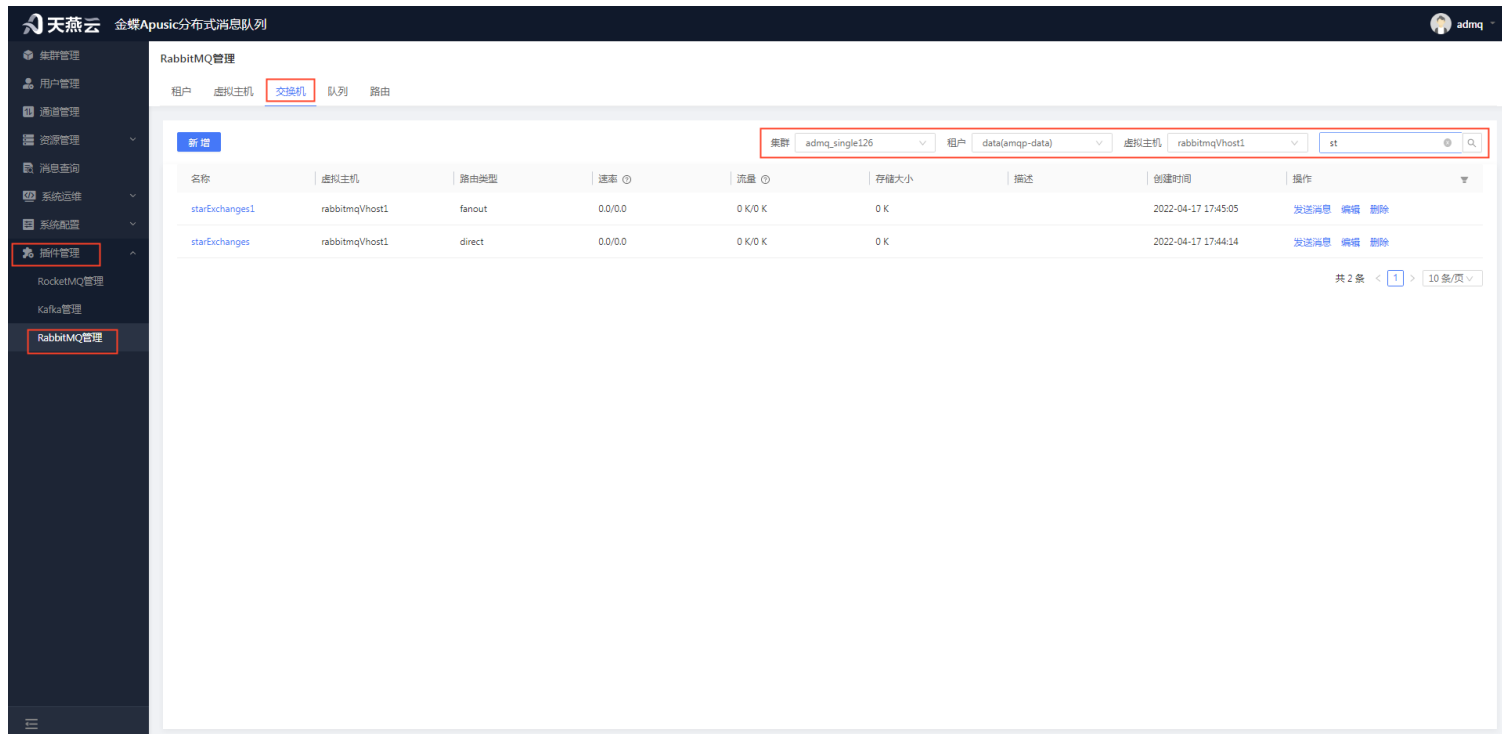
5.10.3.3.3 删除交换机

进入【插件管理】->【RabbitMQ管理】，选择"交换机"页签，选择要删除的交换机名称后的【删除】按钮，在弹出的确认删除弹框中点击【确定】按钮，可删除交换机信息。注意：删除交换机前，需满足集群服务可用。



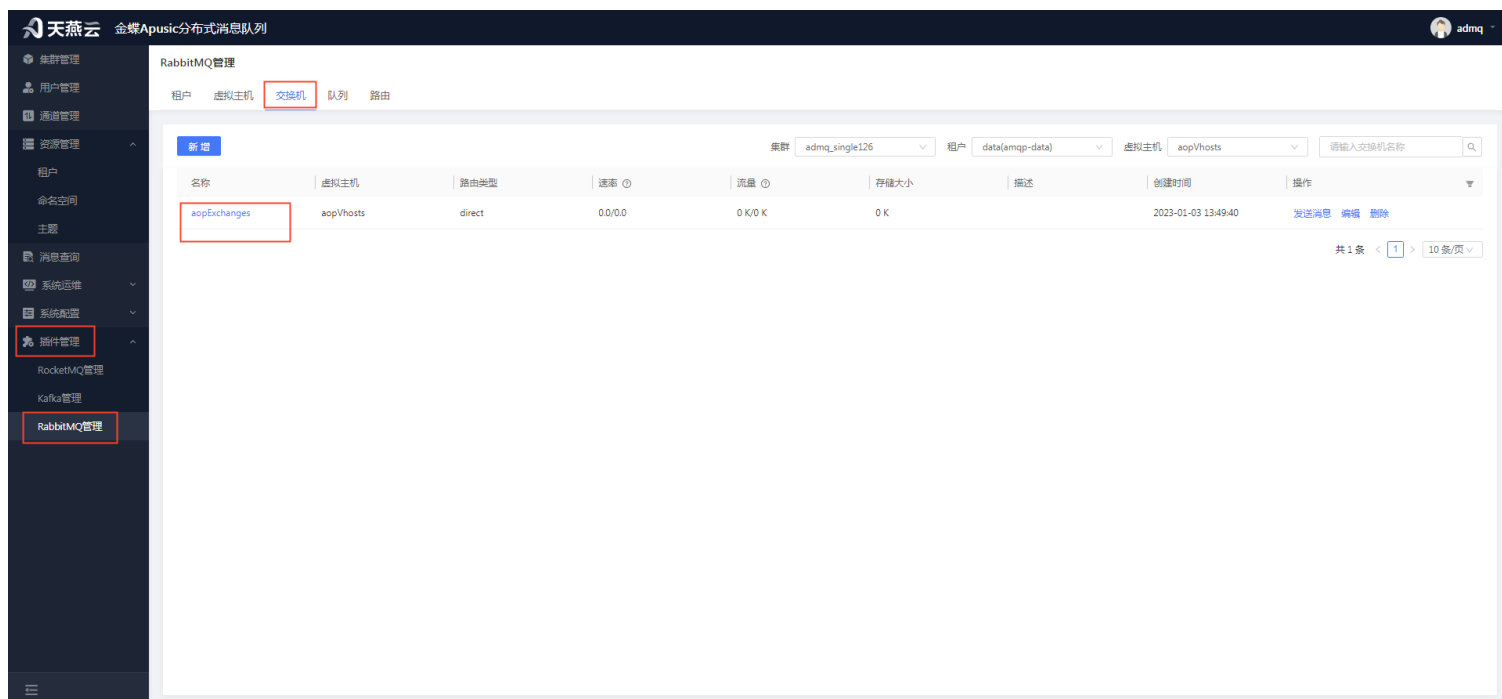
5.10.3.3.4 查询交换机

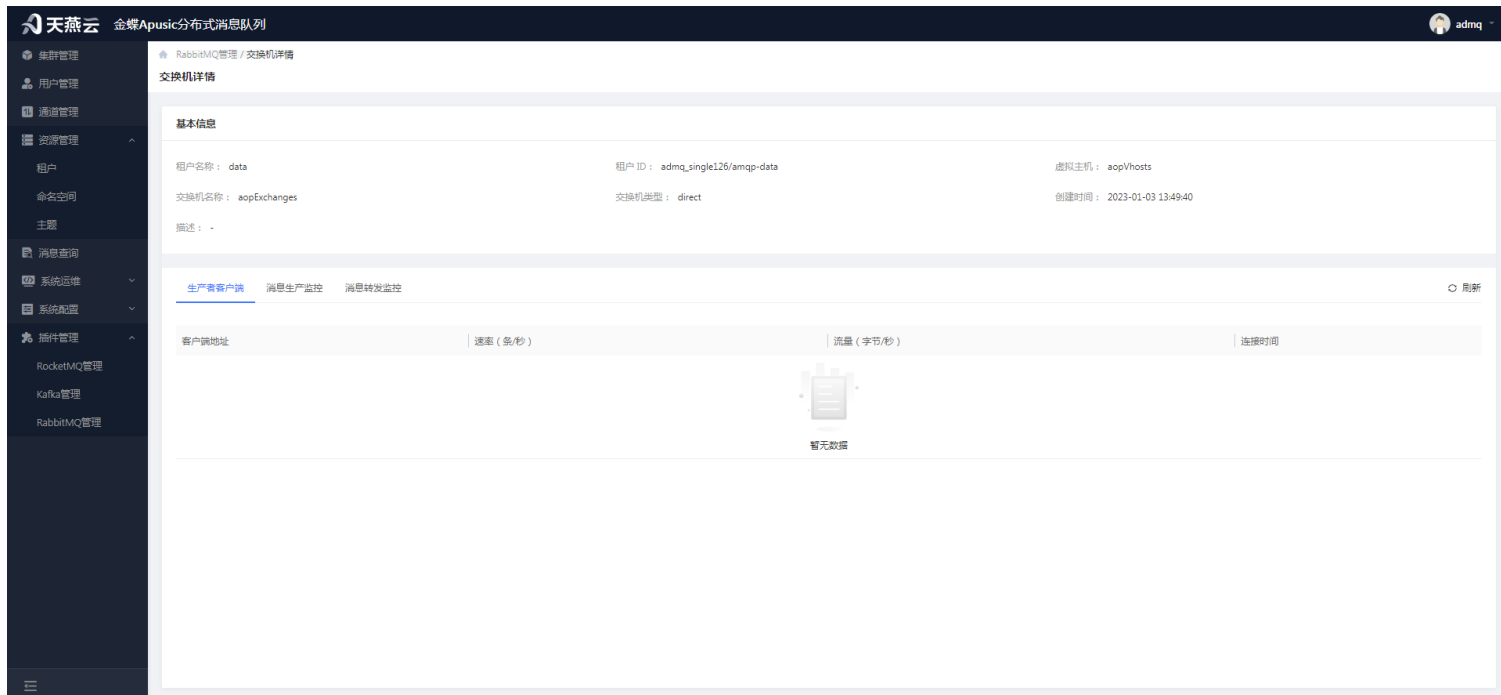
进入【插件管理】->【RabbitMQ管理】，选择"交换机"页签，右上角选择集群名称、租户名称、虚拟主机名称、输入交换机名称，可查询交换机信息。



5.10.3.3.5 查看交换机详情

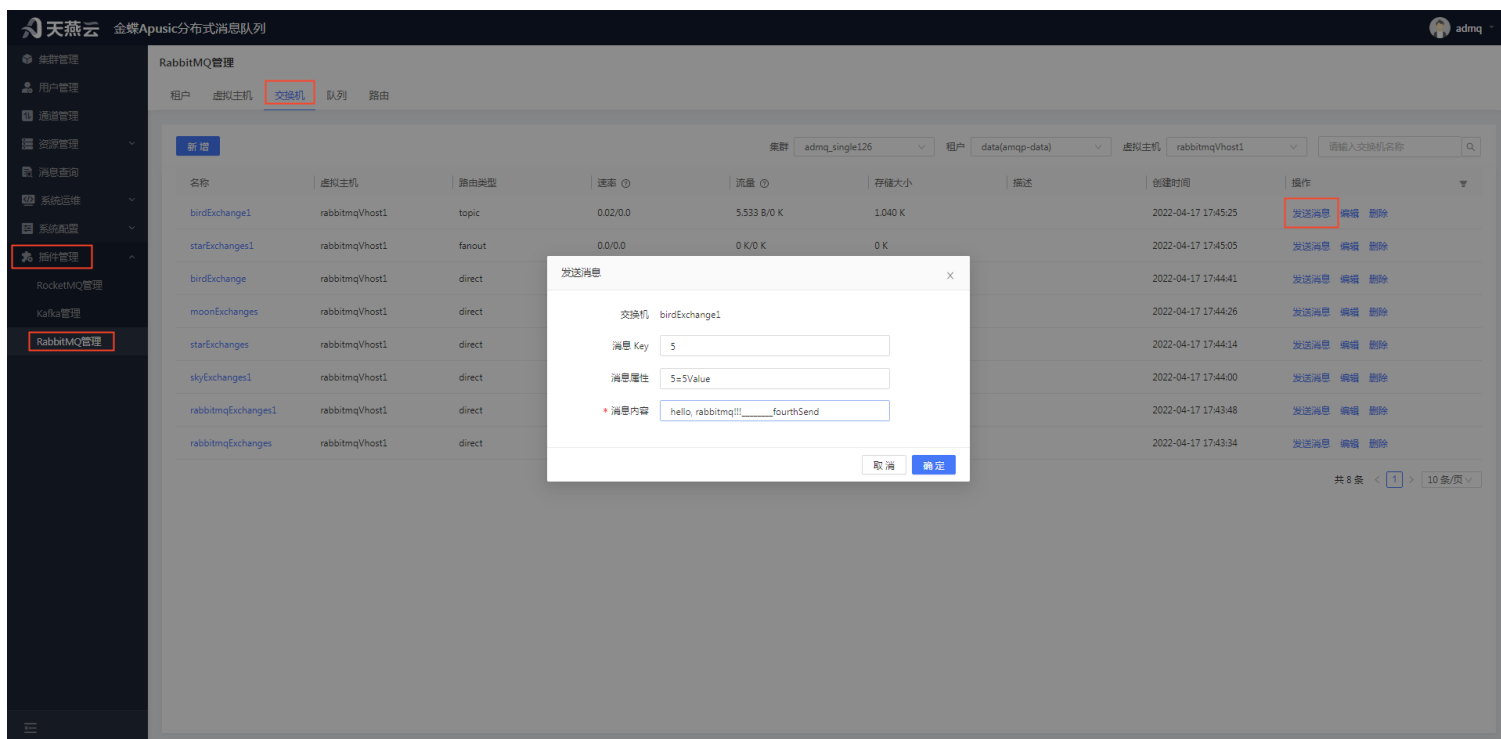
进入【插件管理】->【RabbitMQ管理】，选择“交换机”页签，选择需要查看详细信息的交换机，点击交换机，可查看交换机详细信息。





5.10.3.3.6 发送消息

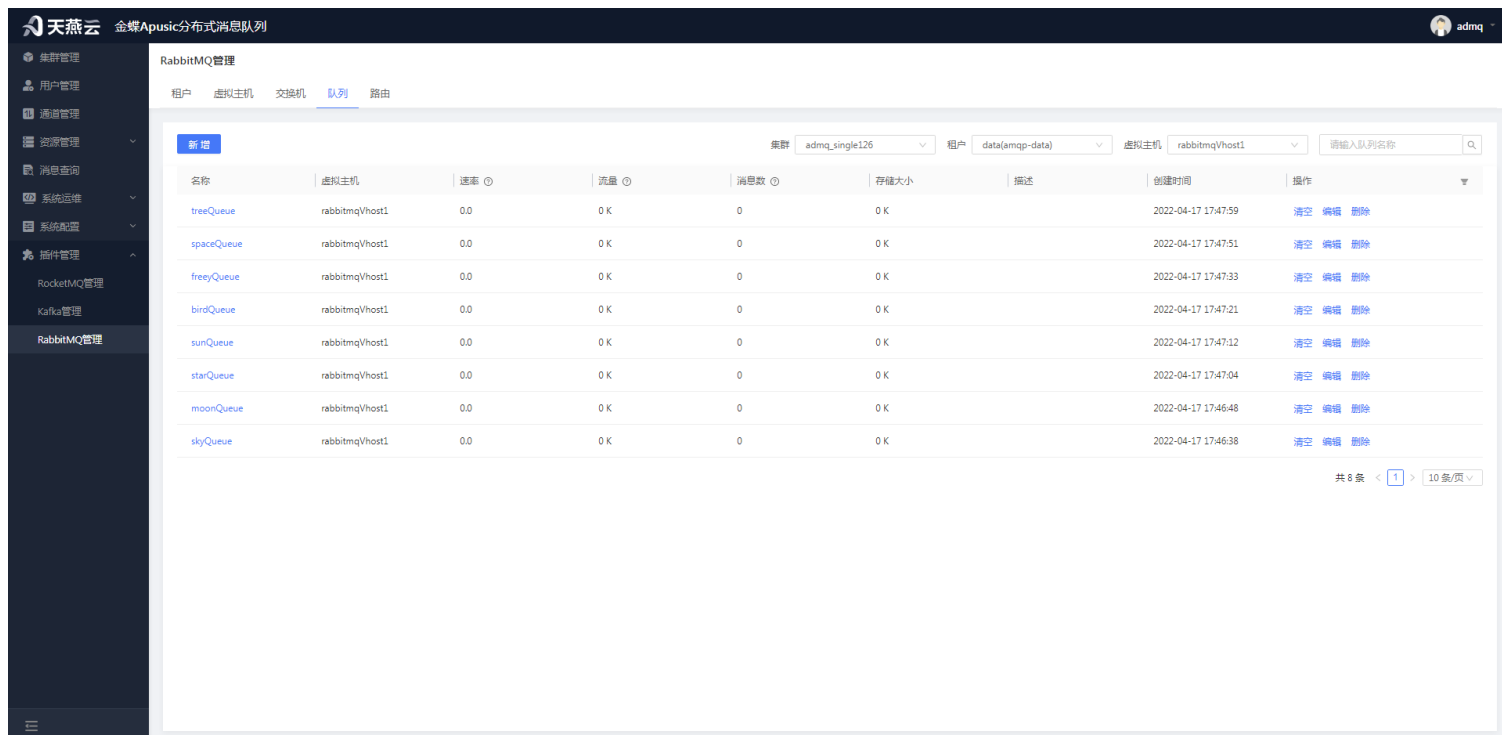
进入【插件管理】->【RabbitMQ管理】，选择“交换机”页签，选择要发送消息的交换机名称后的【发送消息】按钮，可发送消息，发送的消息，可在消息查询中查到。



- 交换机：发送消息的交换机名称，根据选择的交换机名称自动带出，不可修改
- 消息Key：消息的key，自定义
- 消息属性：消息的属性，例如：key=value，多个属性用英文逗号隔开
- 消息内容：发送的消息内容，自定义

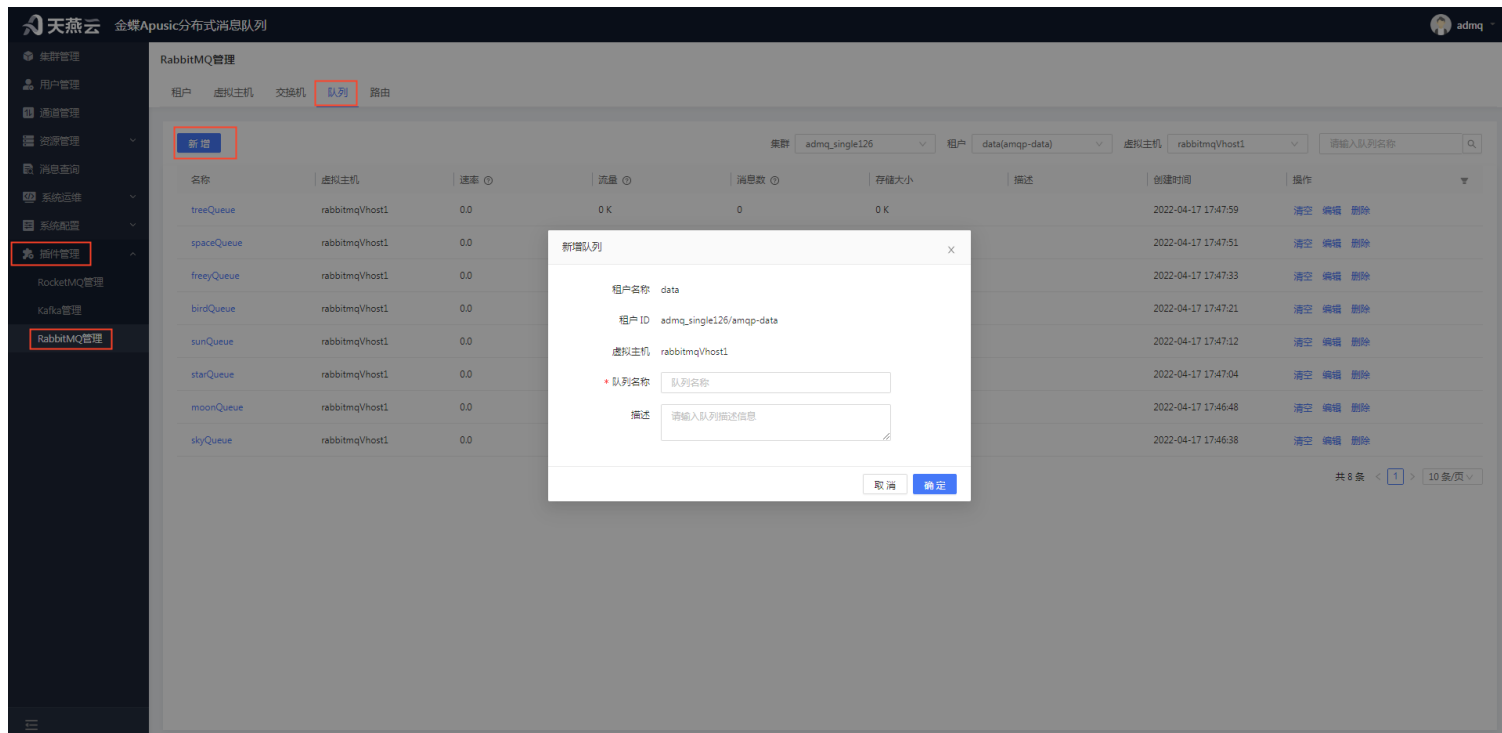
5.10.3.4 队列

队列用来存储生产者发送的消息，生产者并不会直接将消息发送到队列上，而是将消息发送到交换机，由交换机按照一定规则转发给指定队列。队列用于新增队列、编辑队列、删除队列、查询队列、查看队列详细信息、清空队列



5.10.3.4.1 新增队列

进入【插件管理】->【RabbitMQ管理】，选择“队列”页签，点击左上角【新增】按钮，可新增队列信息。注意：新增队列前，需满足集群服务可用。

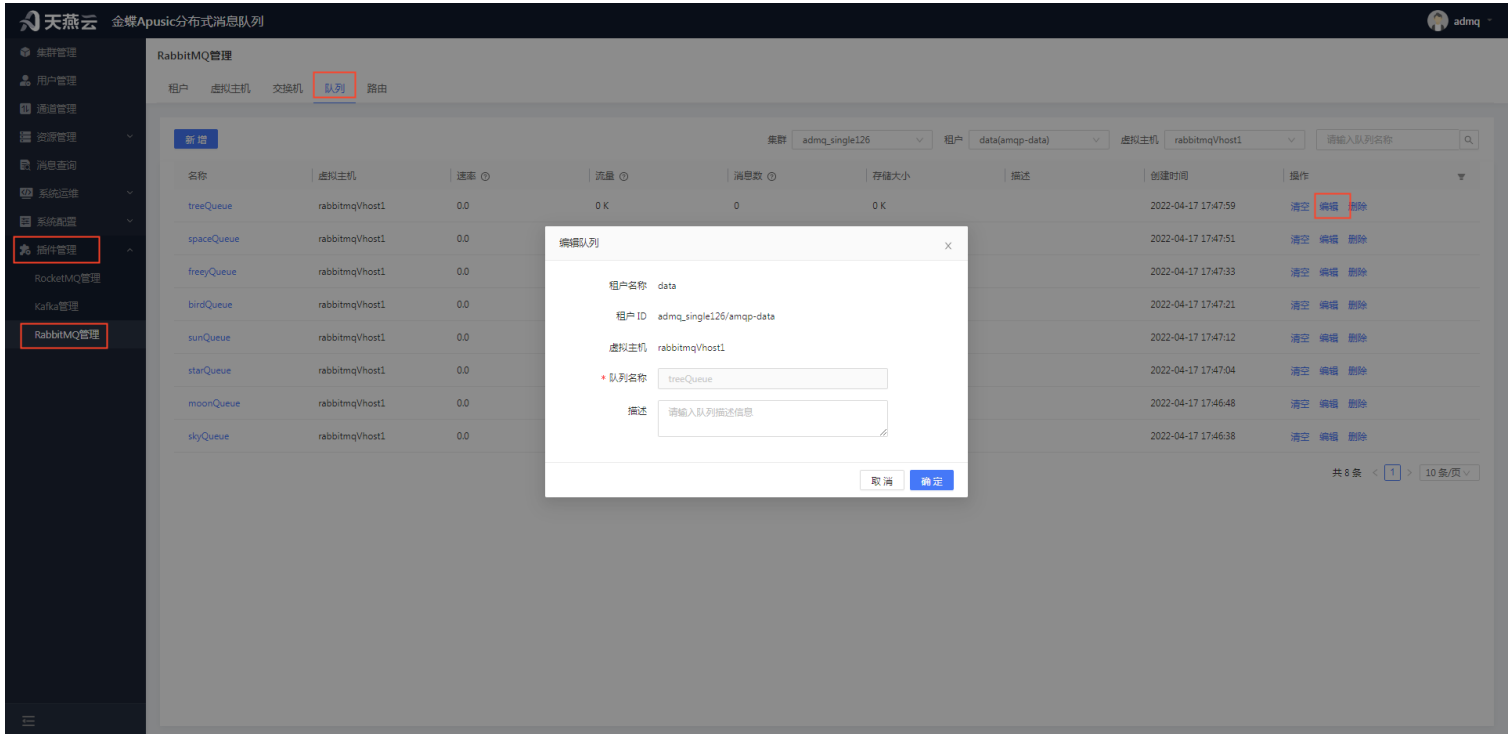


- 租户名称：虚拟主机所属的租户名称，根据选择的租户自动带出，不可修改
- 租户ID：虚拟机所属租户的租户ID，根据选择的租户自动带出，不可修改

- 虚拟主机：队列所属虚拟主机名称，根据选择的虚拟主机自动带出，不可修改
- 队列名称：队列的名称，自定义，唯一标识
- 描述：队列的描述信息

5.10.3.4.2 编辑队列

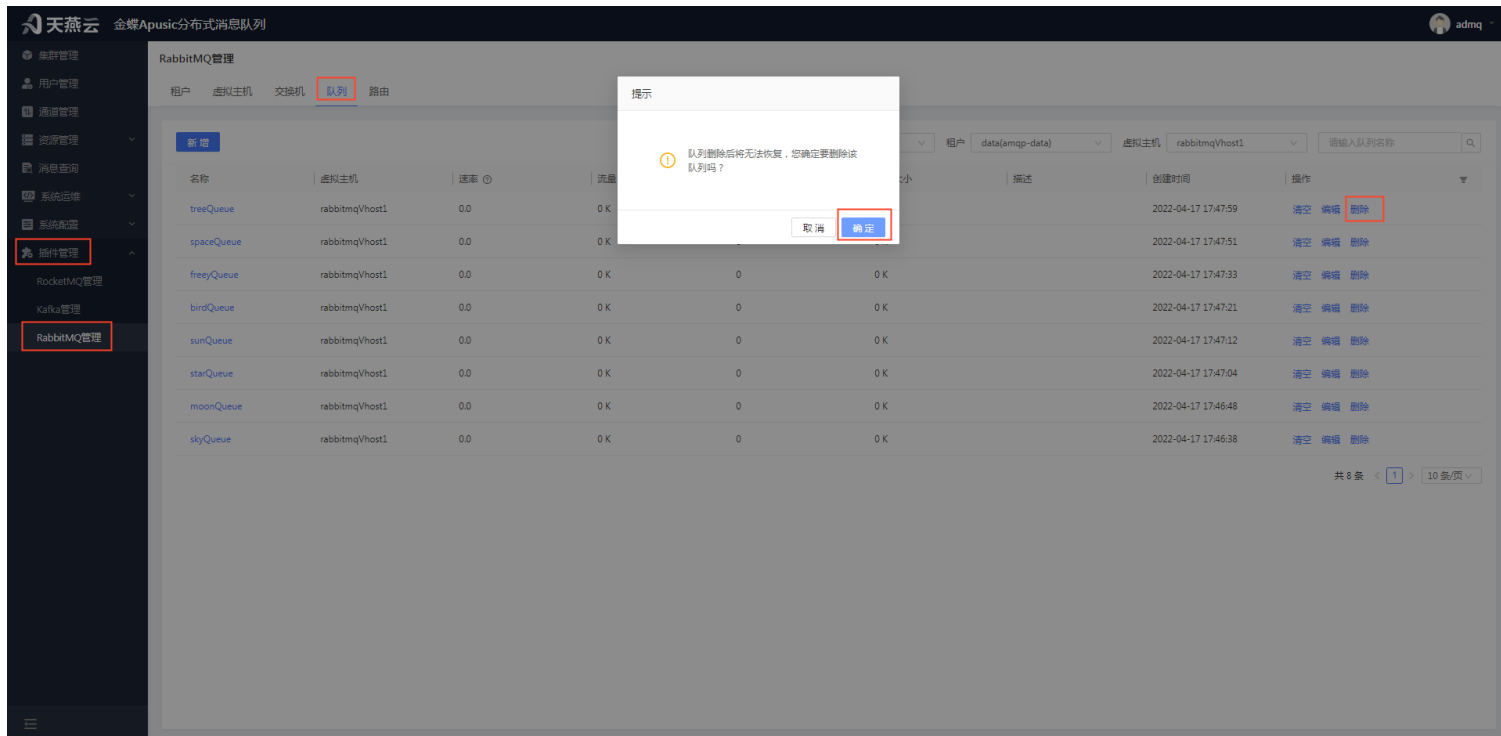
进入【插件管理】->【RabbitMQ管理】，选择“队列”页签，选择要编辑的队列名称后的【编辑】按钮，可编辑队列信息。注意：编辑队列前，需满足集群服务可用。



- 租户名称：虚拟主机所属的租户名称，根据选择的租户自动带出，不可修改
- 租户ID：虚拟主机所属租户的租户ID，根据选择的租户自动带出，不可修改
- 虚拟主机：队列所属虚拟主机名称，根据选择的虚拟主机自动带出，不可修改
- 队列名称：队列的名称，置灰显示，不可修改
- 描述：队列的描述信息

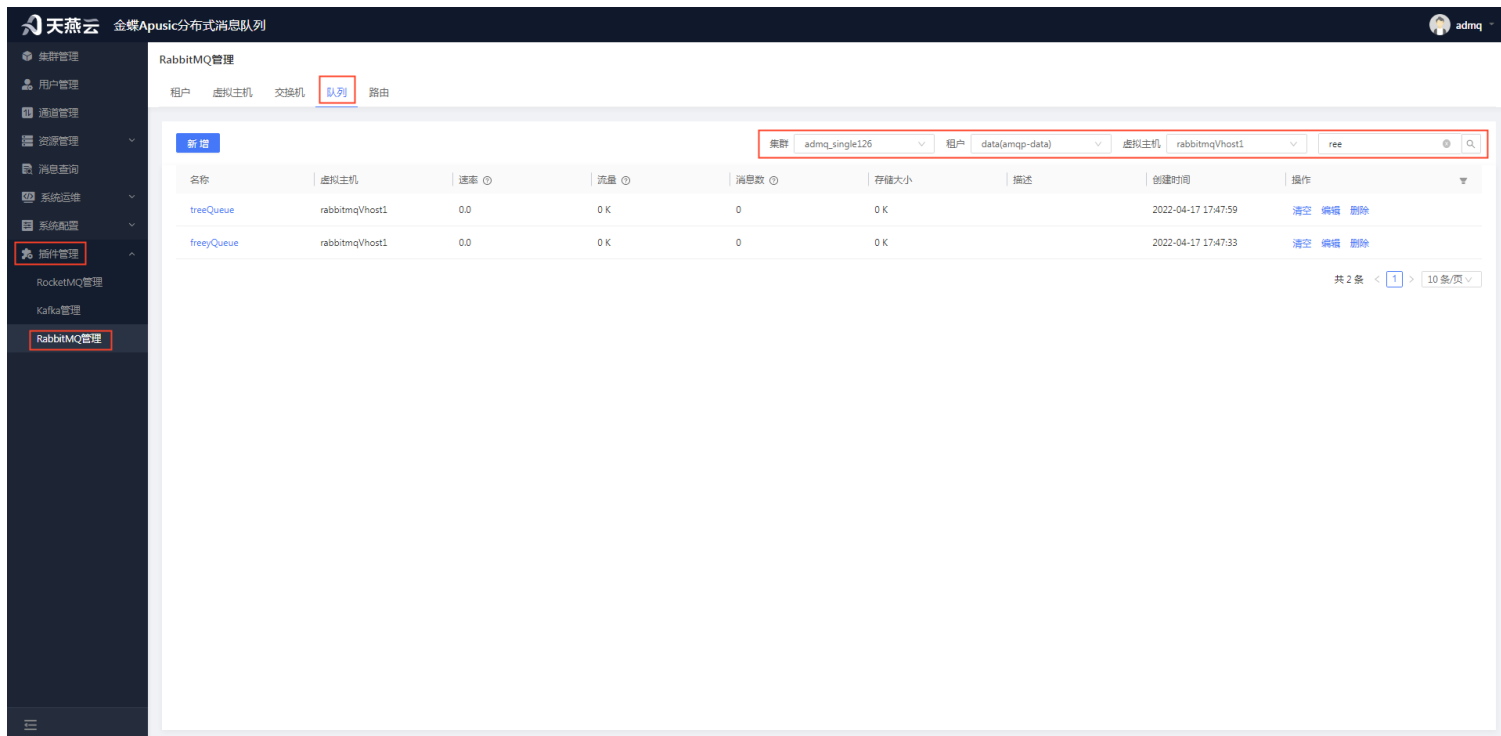
5.10.3.4.3 删除队列

进入【插件管理】->【RabbitMQ管理】，选择“队列”页签，选择要删除的队列名称后的【删除】按钮，可删除队列信息。注意：删除队列前，需满足集群服务可用。



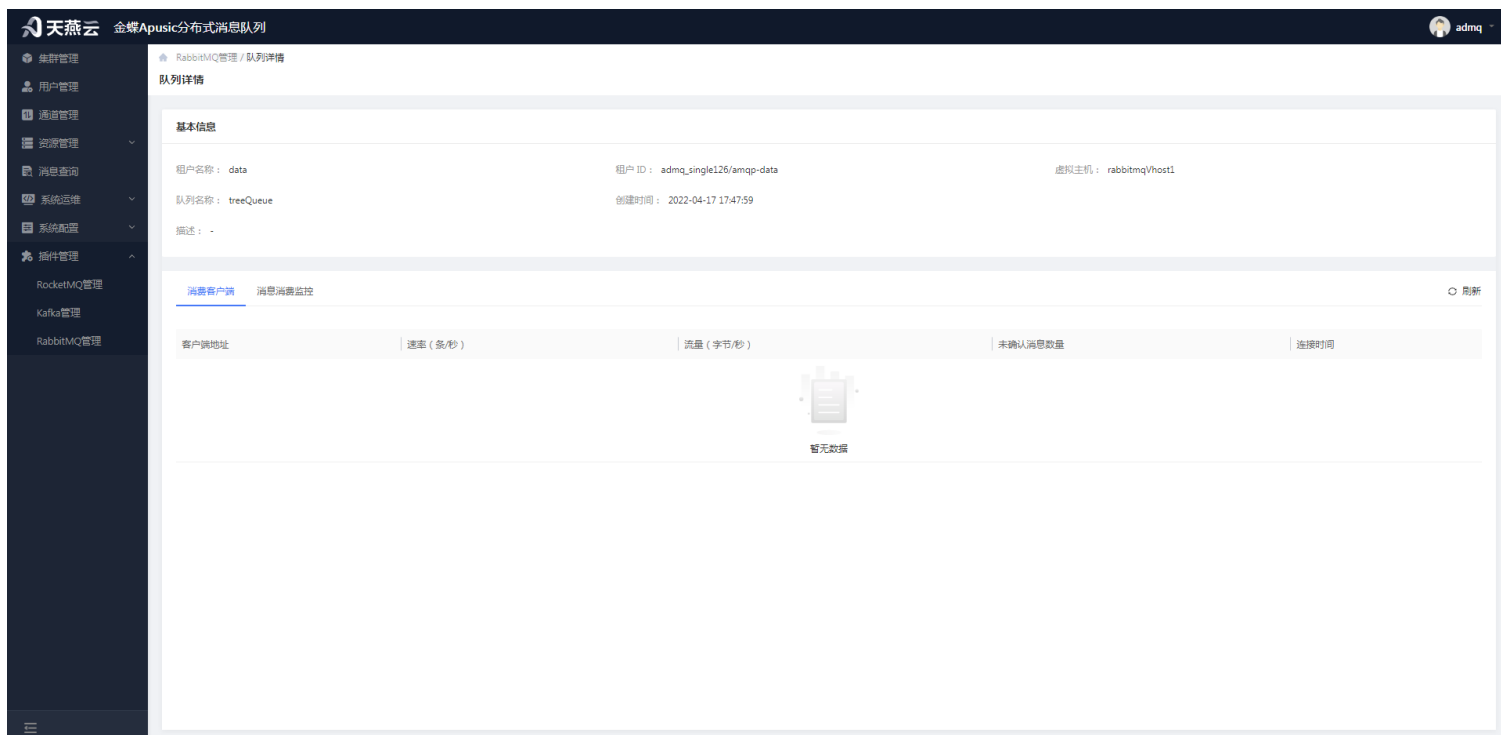
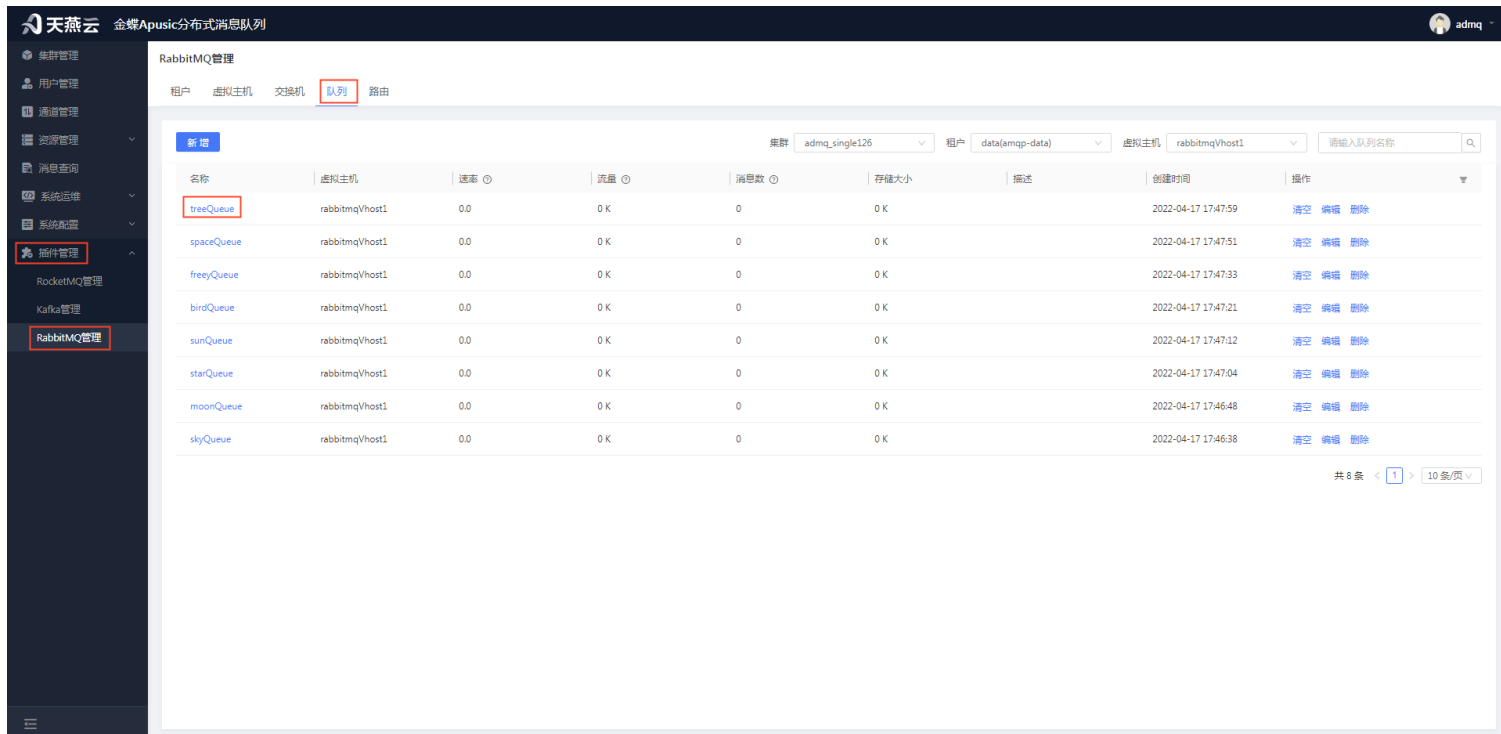
5.10.3.4.4 查询队列

进入【插件管理】->【RabbitMQ管理】，选择“队列”页签，右上角选择集群名称、租户名称、虚拟主机名称、输入队列名称，可查询队列信息。



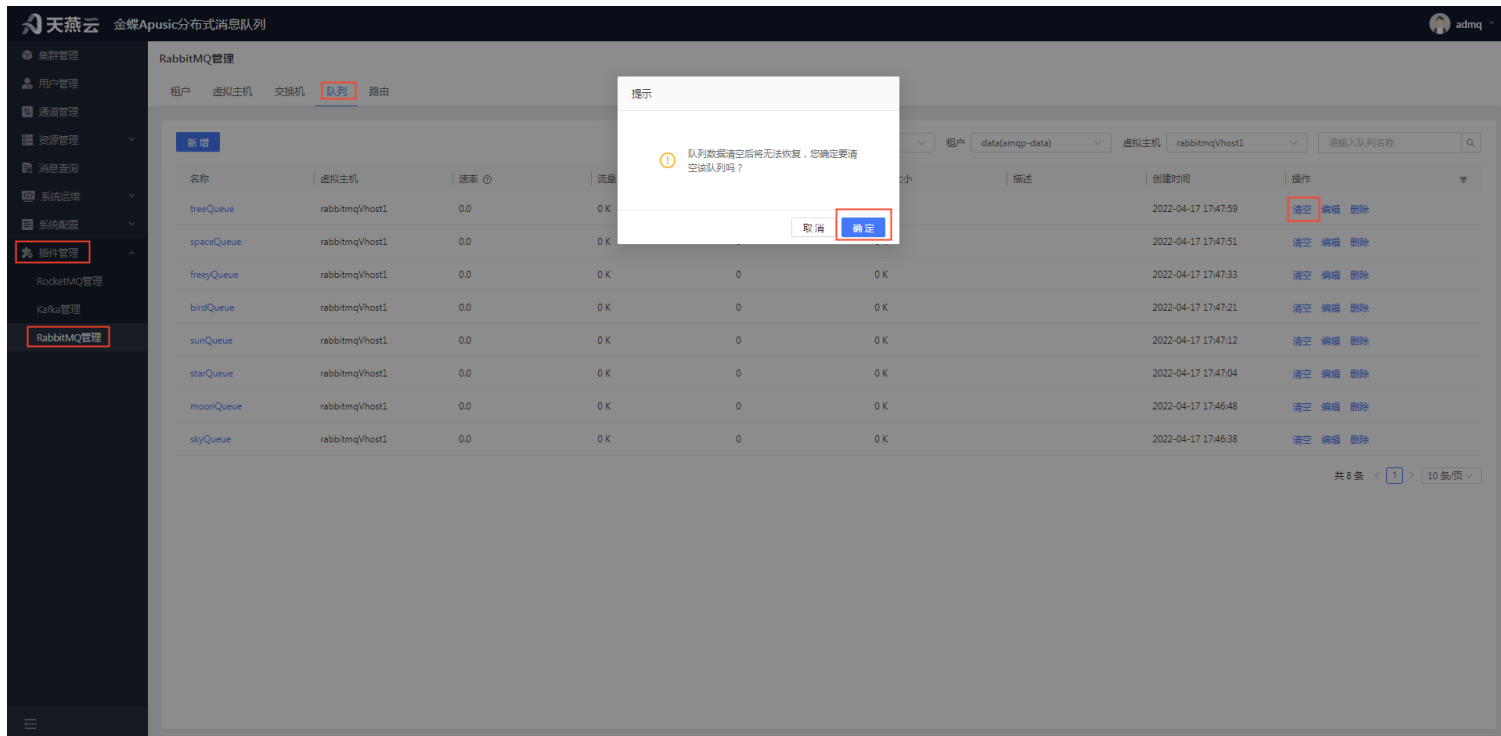
5.10.3.4.5 查看队列详情

进入【插件管理】->【RabbitMQ管理】，选择“队列”页签，选择需要查看详细信息的队列，点击队列名称，可查看队列详细信息，如基本信息，消费客户端、消息消费监控



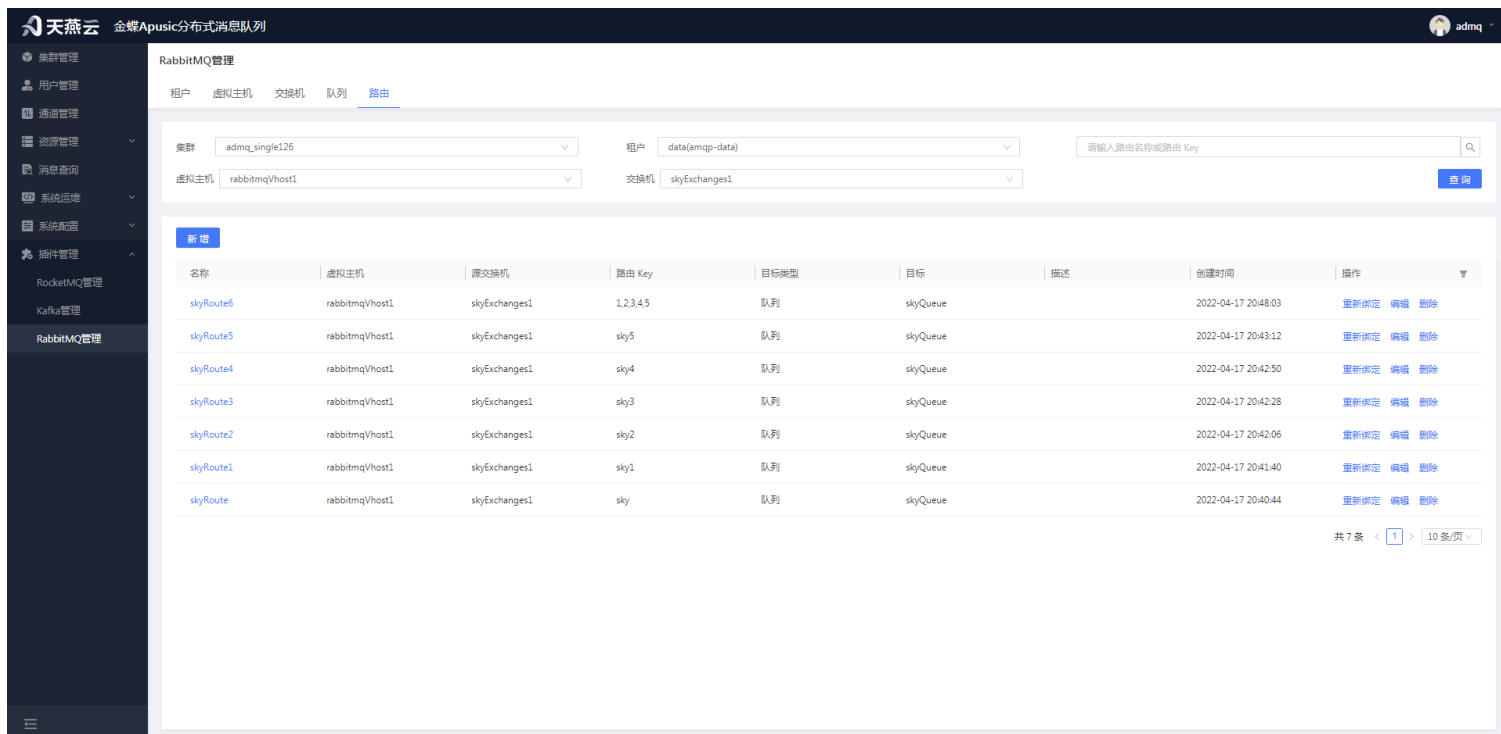
5.10.3.4.6 清空队列信息

进入【插件管理】->【RabbitMQ管理】，选择“队列”页签，选择要清空数据的队列名称后的【清空】按钮，在弹出的确认清空弹框点击【确定】按钮，可清空存储在队列上的消息。注意：清空队列数据前，需满足集群服务可用。



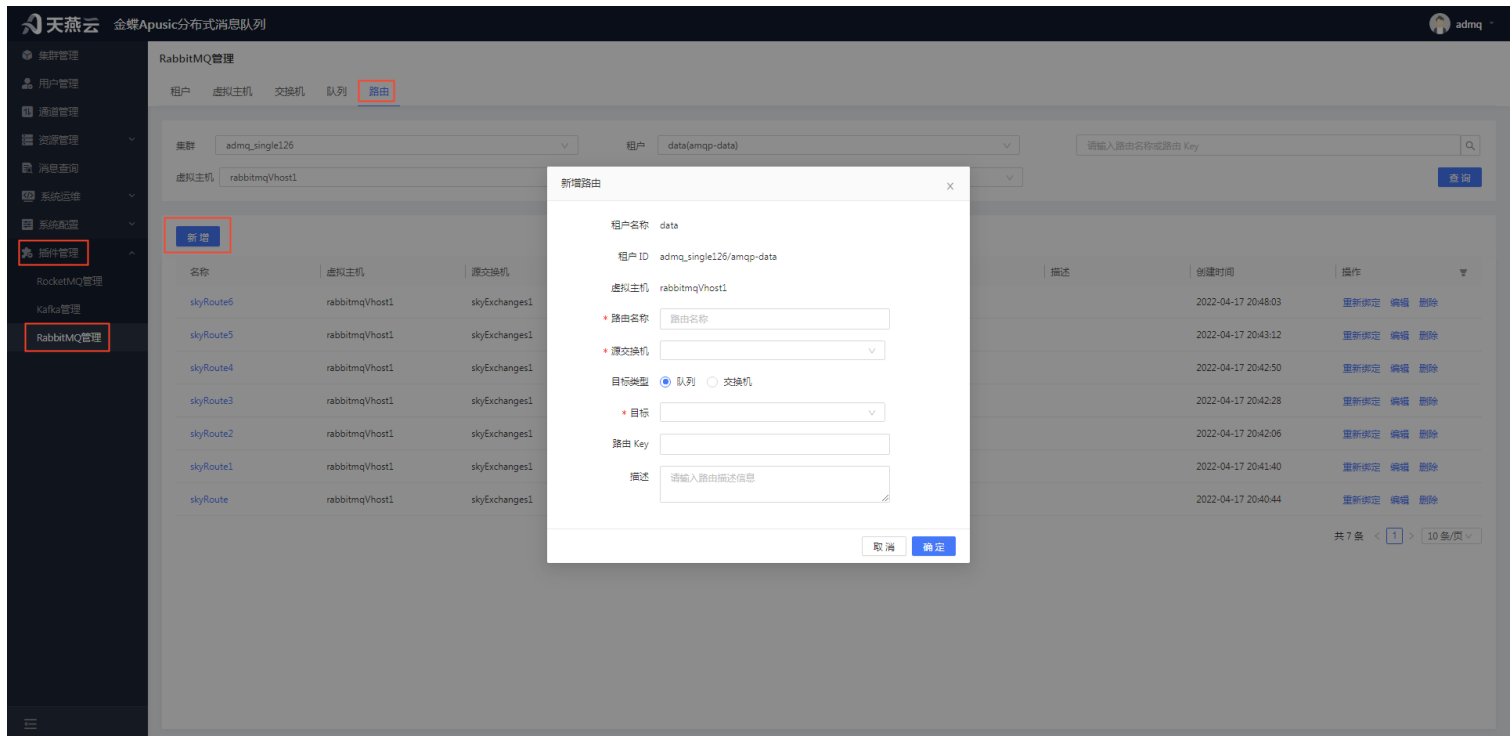
5.10.3.5 路由

路由表示交换机和队列之间的绑定关系，每一个绑定关系会存在一个或者多个路由key，新增一个路由相当于在交换机中建立了一个路由关系表，生产者发送消息的时候，需要声明一个路由key，交换机拿到路由key之后，和路由中的路由key进行匹配，而匹配的规则是通过交换机类型来决定的。路由用来新增路由信息、编辑路由信息、删除路由信息、查询路由信息、查看路由详细信息、重新绑定



5.10.3.5.1 新增路由信息

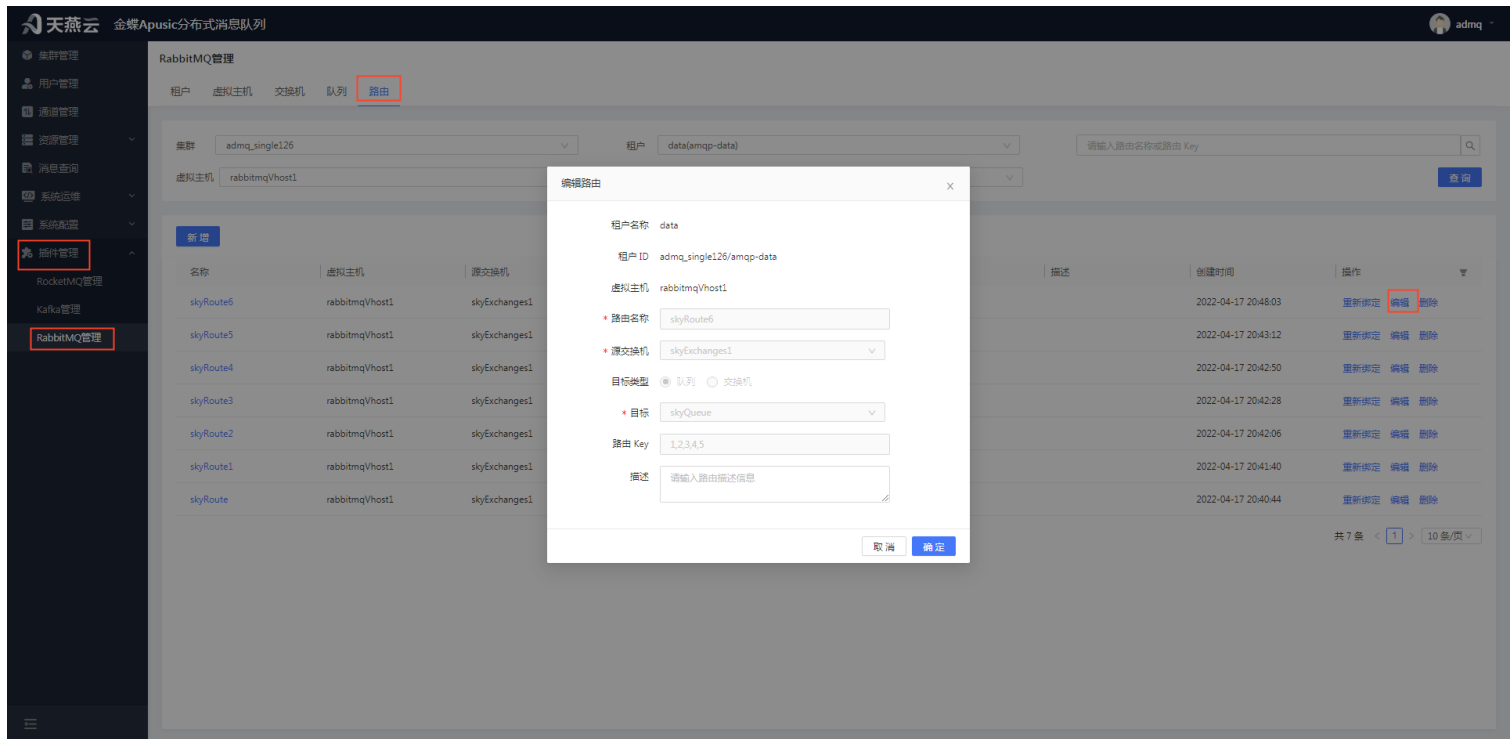
进入【插件管理】->【RabbitMQ管理】，选择“路由”页签，点击左上角【新增】按钮，可新增路由信息。注意：新增路由前，需满足集群服务可用。



- 租户名称: 虚拟主机所属的租户名称, 根据选择的租户自动带出, 不可修改
- 租户ID: 虚拟主机所属租户的租户ID, 根据选择的租户自动带出, 不可修改
- 虚拟主机: 队列所属虚拟主机名称, 根据选择的虚拟主机自动带出, 不可修改
- 路由名称: 路由的名称, 自定义, 唯一标识
- 源交换机: 路由绑定的源交换机, 下拉框, 可选择
- 目标类型: 路由绑定的目标类型, 单选框, 可选队列或者交换机
- 目标: 路由绑定的目标, 下拉框, 可选择
- 路由key: 路由的key, 自定义, 可一个或多个, 多个之间以逗号隔开
- 描述: 路由的描述信息

5.10.3.5.2 编辑路由信息

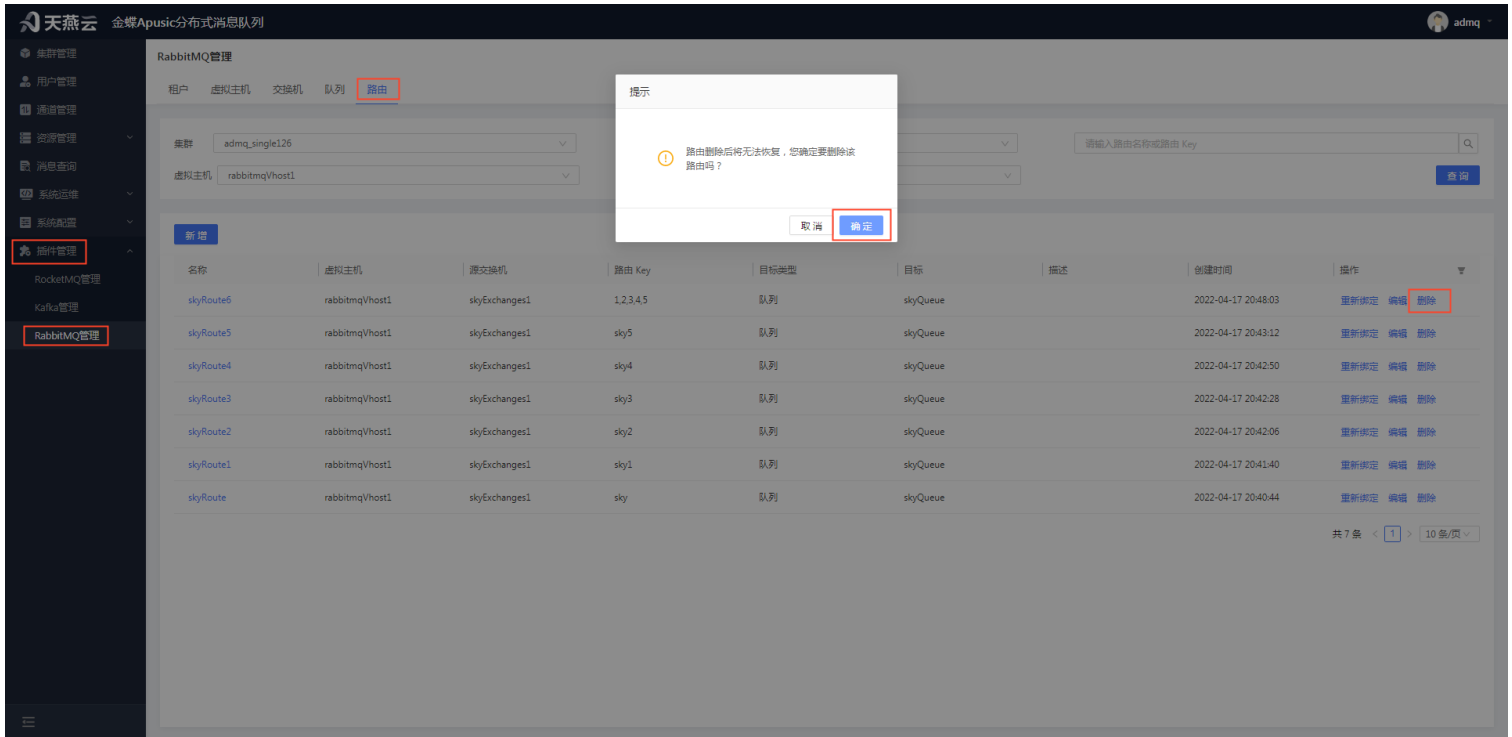
进入【插件管理】->【RabbitMQ管理】, 选择“路由”页签, 选择要编辑的路由名称后的【编辑】按钮, 可编辑路由信息。注意: 编辑路由前, 需满足集群服务可用。



- 租户名称：虚拟主机所属的租户名称，根据选择的租户自动带出，不可修改
- 租户ID：虚拟主机所属租户的租户ID，根据选择的租户自动带出，不可修改
- 虚拟主机：队列所属虚拟主机名称，根据选择的虚拟主机自动带出，不可修改
- 路由名称：路由的名称，置灰显示，不可修改
- 源交换机：路由绑定的源交换机，置灰显示，不可修改
- 目标类型：路由绑定的目标类型，置灰显示，不可修改
- 目标：路由绑定的目标，置灰显示，不可修改
- 路由key：路由的key，置灰显示，不可修改
- 描述：路由的描述信息，可修改

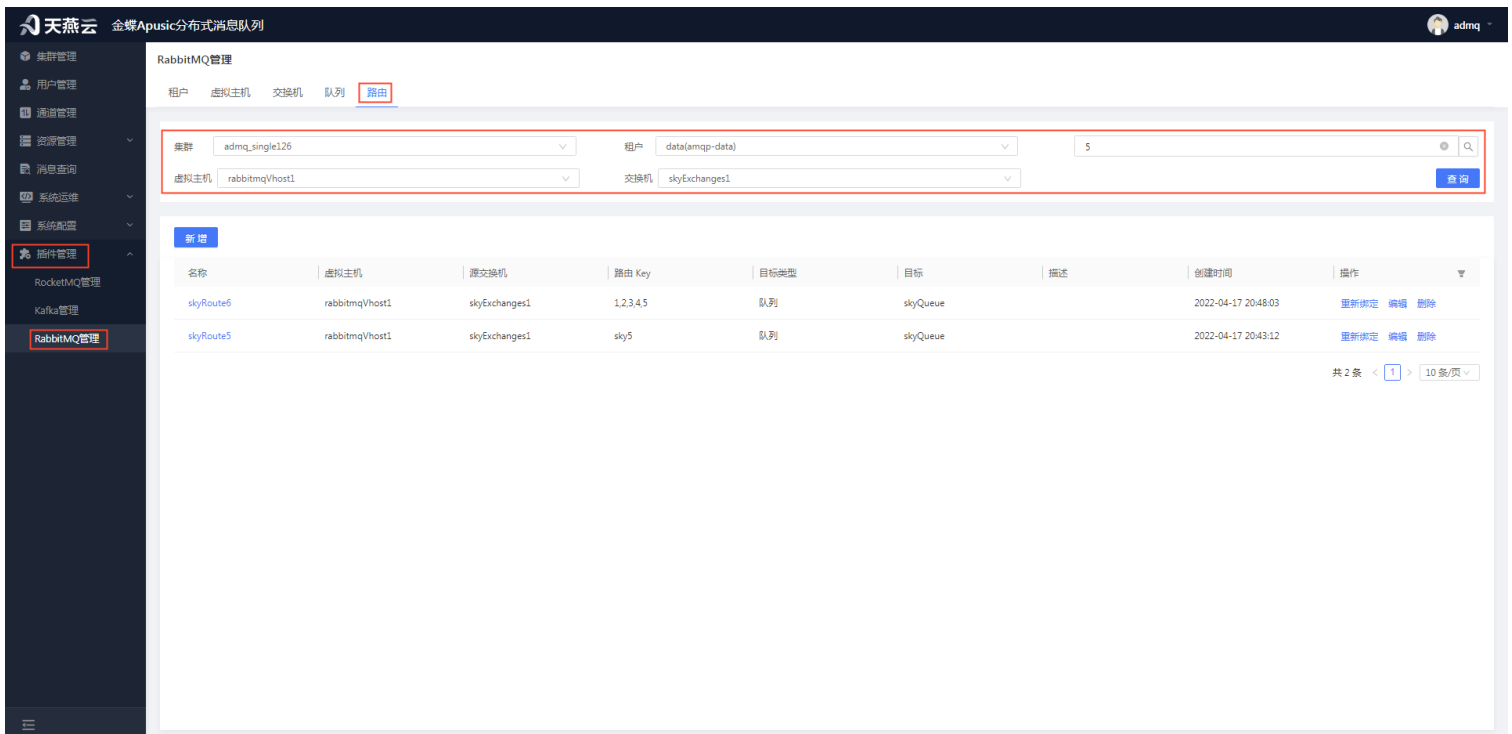
5.10.3.5.3 删除路由信息

进入【插件管理】->【RabbitMQ管理】，选择“路由”页签，选择要删除的路由名称后的【删除】按钮，可删除路由信息。注意：删除路由前，需满足集群服务可用。



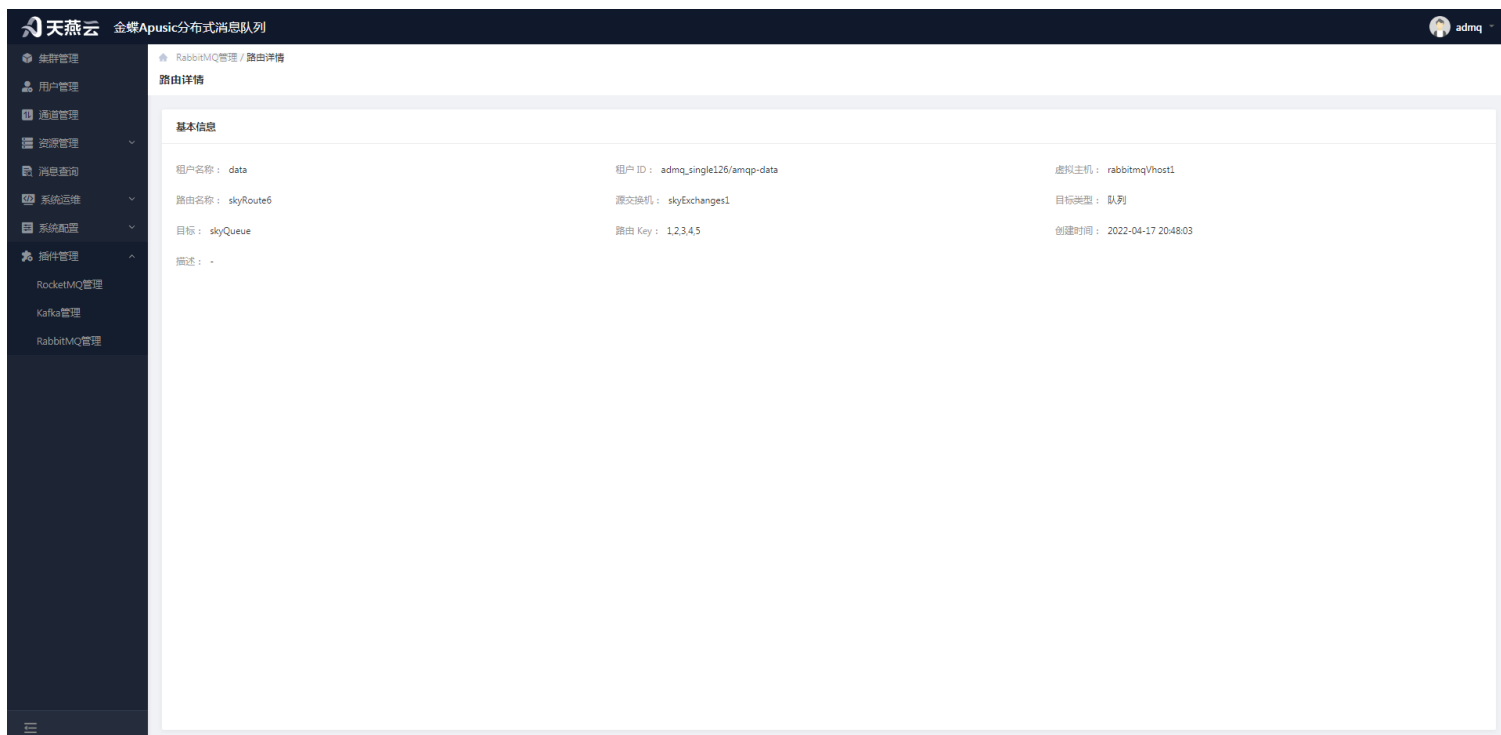
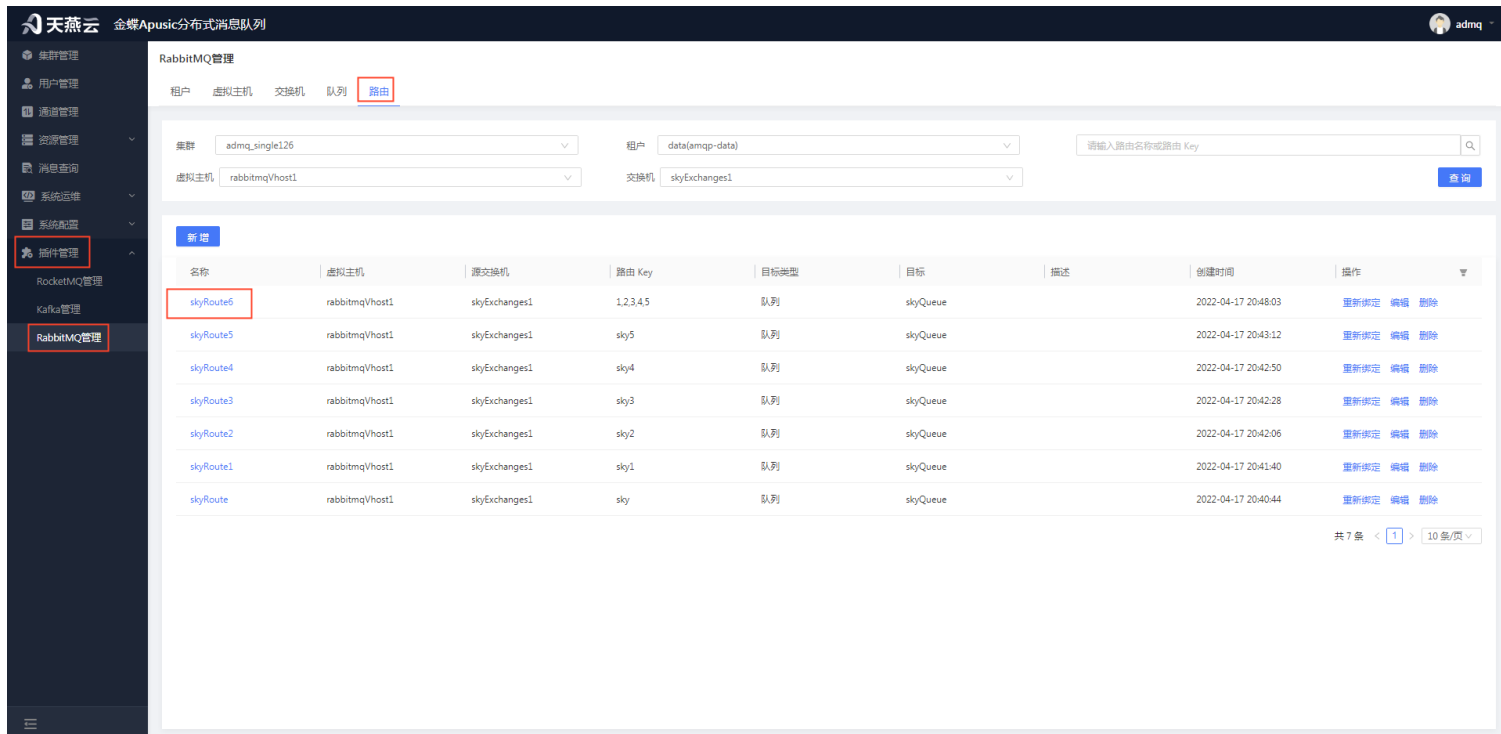
5.10.3.5.4 查询路由

进入【插件管理】->【RabbitMQ管理】，选择“路由”页签，右上角选择集群名称、租户名称、虚拟主机名称、交换机名称，输入路由名称或路由key，可查询路由信息。



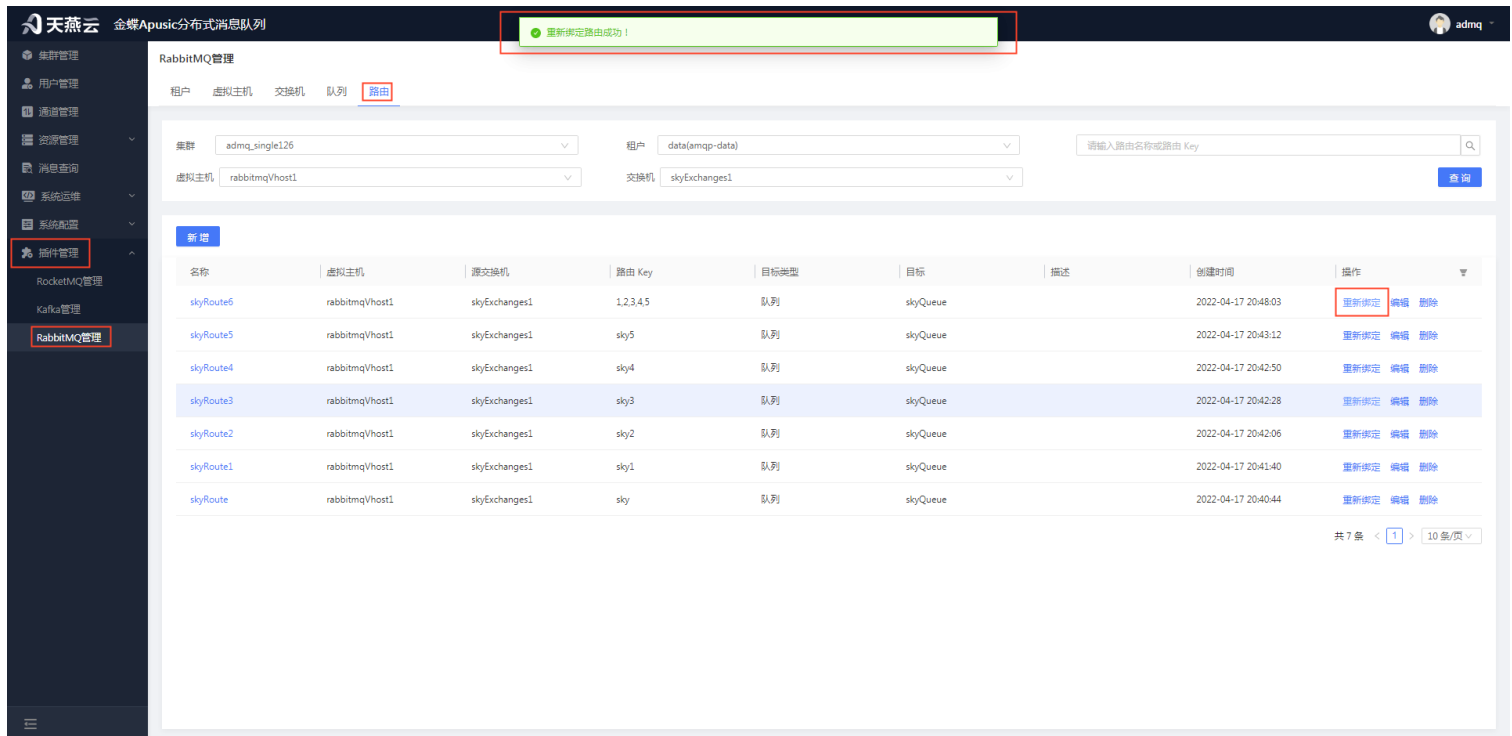
5.10.3.5.5 查看路由详情

进入【插件管理】->【RabbitMQ管理】，选择“路由”页签，选择需要查看详细信息的路由，点击路由名称，可查看路由详细信息，如基本信息。



5.10.3.5.6 重新绑定

进入【插件管理】->【RabbitMQ管理】，选择“路由”页签，选择要重新绑定的路由名称后的【重新绑定】按钮后，页面弹出“重新绑定路由成功！”的提示，表示重新绑定成功。注意：重新绑定路由前，需满足集群服务可用。



5.10.3.6 RabbitMQ插件认证授权

ADMQ的RabbitMQ插件支持认证授权和SSL连接。

5.10.3.6.1 认证

默认情况下，RabbitMQ插件没有加密、认证授权，可通过开启该功能来提高用户数据安全性，目前，该特性仅在Java中可用。

RabbitMQ插件认证机制采用PLAIN机制，采用基于ADMQ令牌的认证机制实现认证。因此，如果你想为RabbitMQ插件启用身份验证特性，你需要为以下组件启用身份验证：(a) ADMQ brokers (b) RabbitMQ插件(RabbitMQ插件的一些配置依赖于原生ADMQ的配置) (c) RabbitMQ客户端

目前，PLAIN机制基于JWT身份验证，因此必须使用AuthenticationProviderToken配置authenticationProviders。

PLAIN

目前，为了提高安全性，我们使用Kafka插件中的SASL(Simple Authentication security Layer) PLAIN机制来实现RabbitMQ插件中的PLAIN机制，这是一种更好的授权方式。如果您希望使用PLAIN机制为RabbitMQ插件启用身份验证特性，请遵循以下步骤：

- (1) 在ADMQ的计算节点上开启认证

对于PLAIN机制，Rabbitmq的身份验证被转化成ADMQ的JWT身份验证，所以需要配置JWT身份验证，并在conf/broker.conf或conf/standalone.conf文件中设置以下属性：

- 启用ADMQ Broker的认证

```
authenticationEnabled=true
authenticationProviders=org.apache.pulsar.broker.authentication.AuthenticationProviderToken
```

- 启用ADMQ Broker和RabbitMQ插件之间的认证

```
brokerClientAuthenticationPlugin=org.apache.pulsar.client.impl.auth.AuthenticationToken
brokerClientAuthenticationParameters=token:<token-of-super-user-role>
superUserRoles=<super-user-roles>
```

- 指定密钥

-如果使用密钥，请按如下方式设置属性：

```
tokenSecretKey=file:///path/to/secret.key
```

```
tokenSecretKey=data:;base64,FLFyW0oLJ2Fi22KKcm21J18mbAdztfSHN/lAT5ucEKU=
```

-如果使用公钥/私钥，设置属性如下：

```
tokenPublicKey=file:///path/to/public.key
```

(2) Rabbitmq客户端启用认证

属性	描述	举例
password	password必须是来自ADMQ的令牌身份验证参数。	token:xxx

```
connectionFactory.setPort(5672);
connectionFactory.setPassword("token:xxxx");
Connection connection = connectionFactory.newConnection();
```

5.10.3.6.2 授权

目前，RabbitMQ插件授权只支持 虚拟主机/命名空间 级别的生产 and 消费权限。要在AoP中启用授权，请确保已启用认证。

(1) 为ADMQ Broker开启授权并分配超级用户

```
authorizationEnabled=true
```

(2) 生成JWT令牌

令牌是与用户关联的凭证，可以在用户管理中查看相关的令牌值。

(3) 授权

在管理控制台上为虚拟主机授予produce权限或者consume权限，具体详见虚拟主机发布消费权限一节。

5.10.3.6.3 TLS/SSL连接

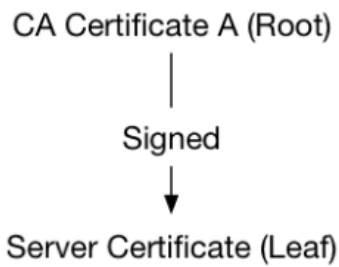
目前RabbitMQ插件中的TLS/SSL仅为代理端口设计，代理端口保证了集群代理的正确寻址。

下面的例子展示了如何通过TLS/SSL连接RabbitMQ插件服务，对于用户来说，客户端就像原生RabbitMQ一样。

(1) 使用tls-gen的基本配置文件

```
git clone https://github.com/rabbitmq/tls-gen tls-gen
cd tls-gen/basic
# private key password
make PASSWORD=123456
make verify
make info
ls -l ./result
```

基本tls-gen配置文件生成的证书链是这样的：



使用以上步骤生成的文件已经可以保证单向认证的正常运行，但是为了保证双向认证，还需要生成以下两个文件。Java使用TrustStore工具来信任和统一管理相关文件。下面，我们创建两个信任文件，一个包含CA的信任链信息，另一个包含客户端使用的服务器证书：

```
keytool -import -alias server1 -file /path/to/ca_certificate.pem -keystore
/path/to/rabbitstore_ca
keytool -import -alias server1 -file /path/to/server_certificate.pem -keystore
/path/to/rabbitstore
```

(2) ADMQ的broker节点上配置相关参数

Configuration Key	Description
amqpTlsEnabled	Aop Tls Enable, true or false
amqpSslProtocol	The protocol server sending to the client, default TLSv1.2
amqpSslKeystoreType	The certification type Server Keystore used, default PKCS12
amqpSslKeystoreLocation	The Server Keystore p12 file location, example /usr/local/server.p12
amqpSslKeystorePassword	The Server Keystore p12 file password
amqpSslTruststoreType	The certification type Truststore(CA) used, default JKS
amqpSslTruststoreLocation	The Truststore(CA) file location, example /usr/local/rabbitstore_ca
amqpSslTruststorePassword	The Truststore(CA) file password
amqpSslKeymanagerAlgorithm	The Keymanager algorithm, default SunX509
amqpSslTrustmanagerAlgorithm	The Trustmanager algorithm, default SunX509
amqpSslClientAuth	When set to true, TLS connection will be peer

(2) RabbitMQ客户端配置

```

• amqpSslClientAuth = false
    connectionFactory.setPort(6672);
    connectionFactory.useSslProtocol("TLSv1.2");
    Connection connection = connectionFactory.newConnection();
• amqpSslClientAuth = true
    connectionFactory.setPort(6672);
    char[] keyPassphrase = "Keystore file password".toCharArray();
    KeyStore ks = KeyStore.getInstance("PKCS12");
    ks.load(new FileInputStream("The client p12 Keystore file location"), keyPassphrase);
    KeyManagerFactory kmf = KeyManagerFactory.getInstance("SunX509");
    kmf.init(ks, keyPassphrase);
  
```

```

char[] trustPassphrase = "Truststore file password".toCharArray();
KeyStore tks = KeyStore.getInstance("JKS");
tks.load(new FileInputStream("The Truststore p12 file location"), trustPassphrase);
TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509");
tmf.init(tks);
SSLContext c = SSLContext.getInstance("TLSv1.2");
c.init(kmf.getKeyManagers(), tmf.getTrustManagers(), null);
connectionFactory.useSslProtocol(c);
Connection connection = connectionFactory.newConnection();

```

5.11 监控运维

- 备注：若想使用AMP监控运维，必须使用AUC登录界面跳转到ADMQ管理控制台后再创建集群，创建集群具体步骤见第3章。

ADMQ集成AUC（金蝶Apsic统一管控平台）和AMP（金蝶天燕智能运维系统），通过AUC统一登录，使用AMP作为ADMQ的监控运维工具。

ADMQ的核心组件Broker、Bookkeeper和zookeeper都内置了Prometheus 接口，提供了Topic使用相关的指标，也暴露出了集群中各个组件的整体健康状况。这些相关指标可以直接使用 HTTP 方式获取，并支持接入到AMP中，进行指标的统计汇总，以及针对指标的监控和报警。

5.11.1 管控台集成AUC和AMP

5.11.1.1 AUC登录

在管控台config目录下application.properties配置文件中添加参数：

```

# AUC服务地址；如果使用AUC登录，则需要把配置项security.same-site.enabled设置成false
auc.url=http://172.24.4.170:9000
# AMP服务地址
amp.url=http://172.24.4.170:9002

```

重新启动admq-manager服务：

```

bin/admq-manager stop
bin/admq-manager start

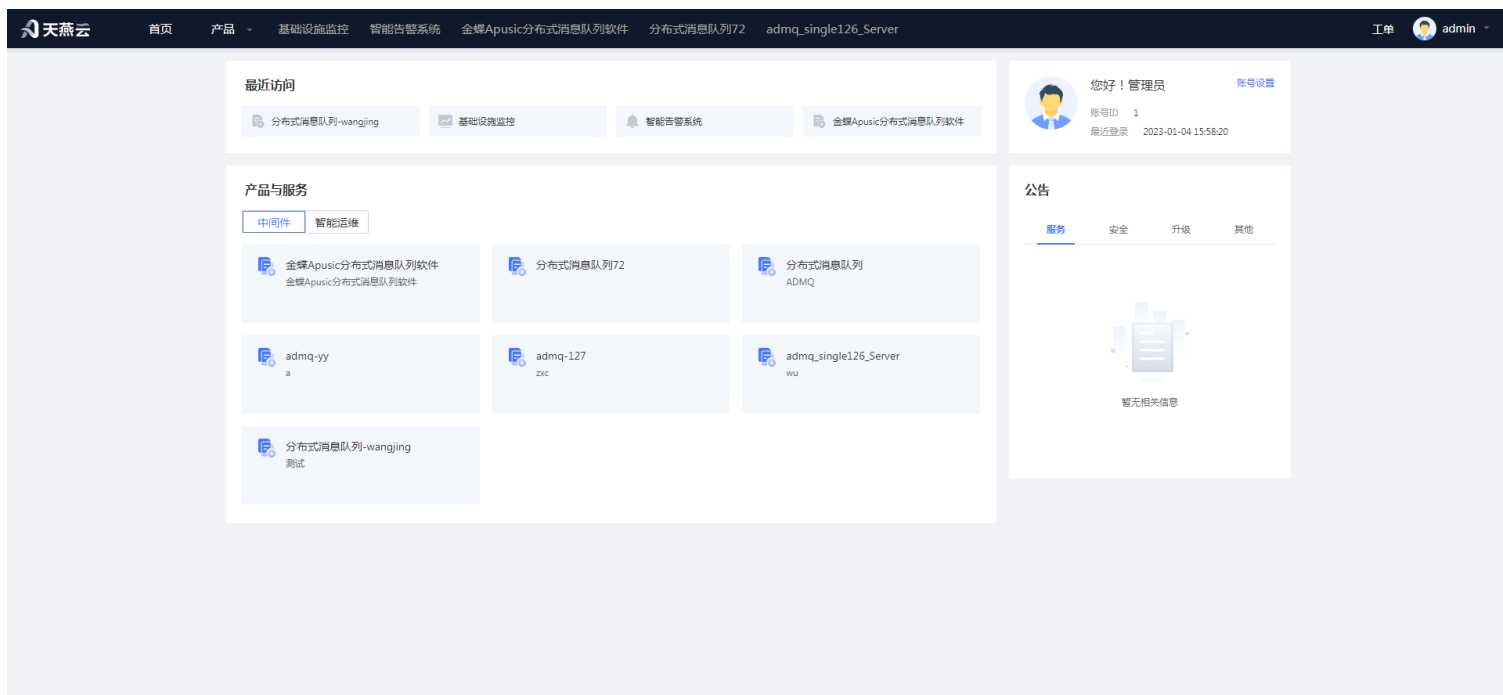
```

浏览器输入URL：<http://172.24.4.170:9000/>，进入AUC登录界面。



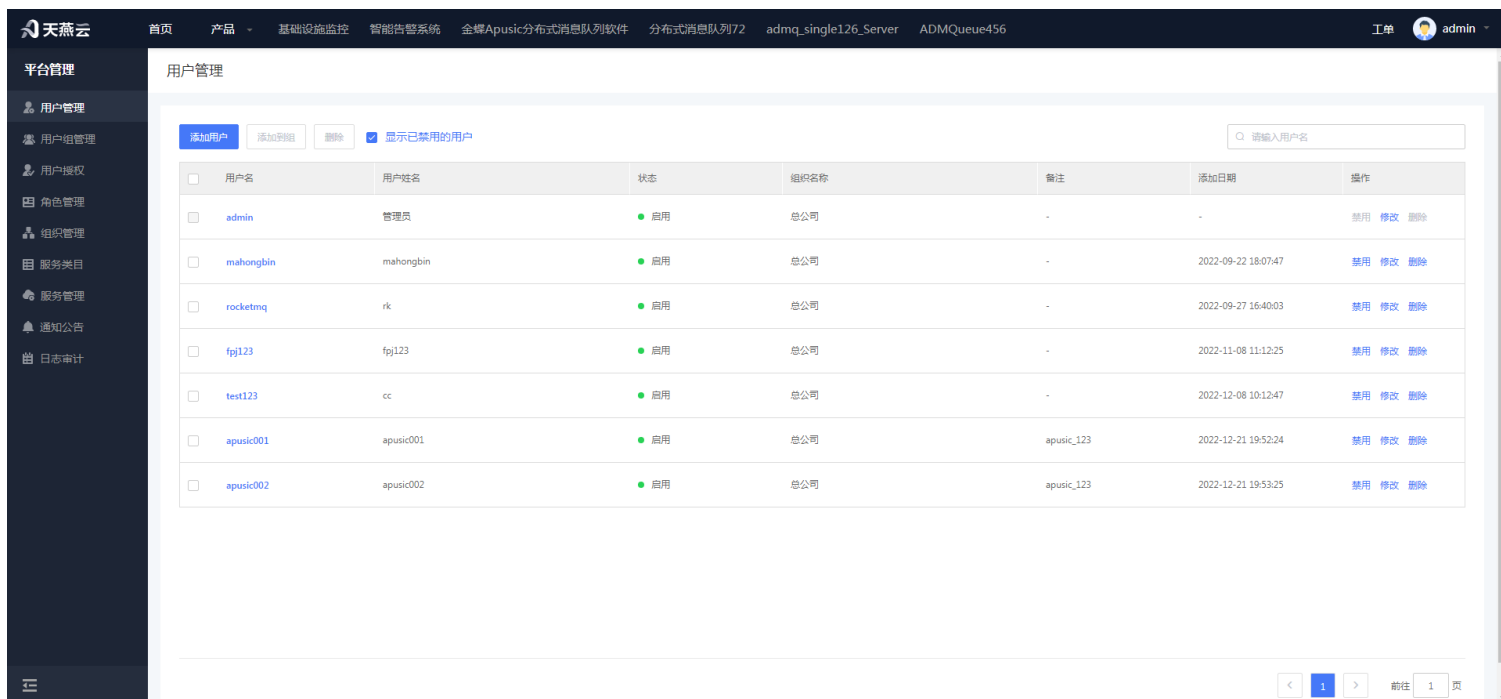
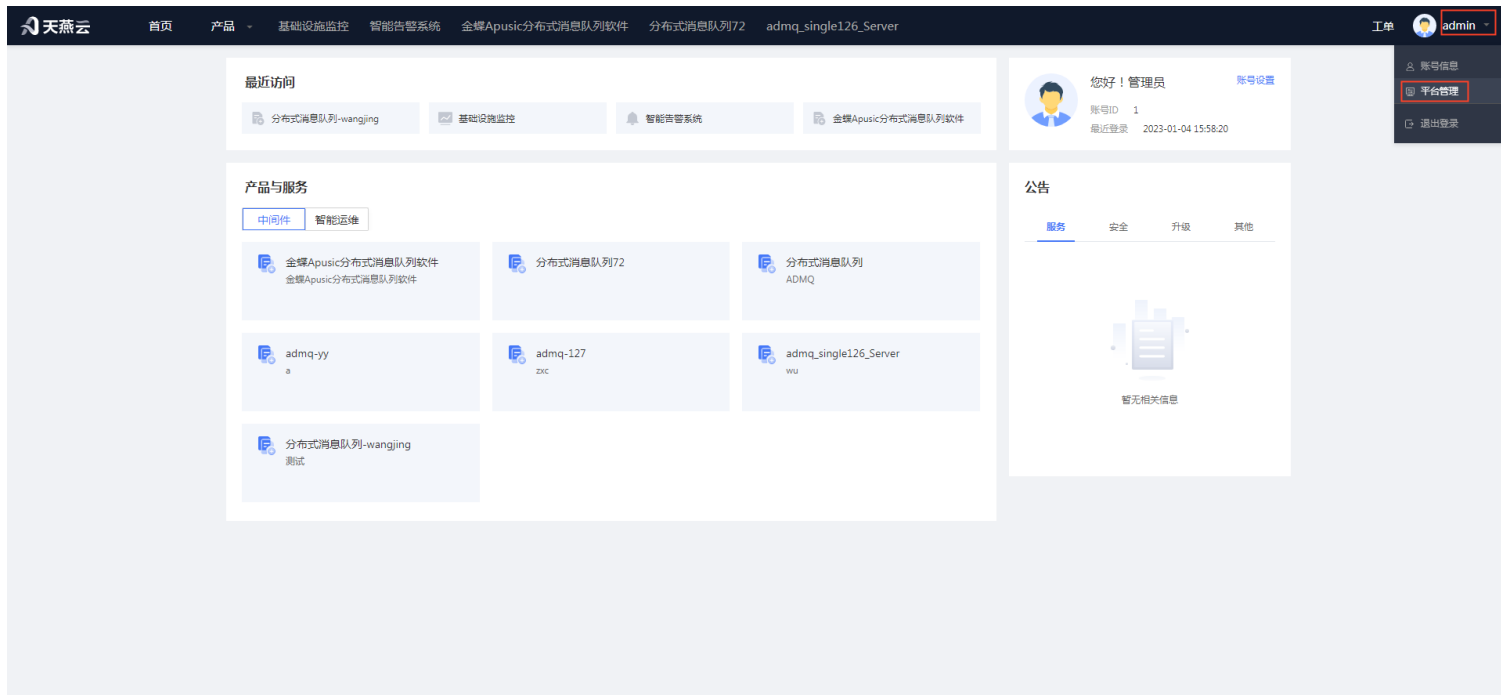
© 2000-2022 金蝶天燕云计算股份有限公司版权所有

输入用户名、密码、验证码，进入AUC管理界面：

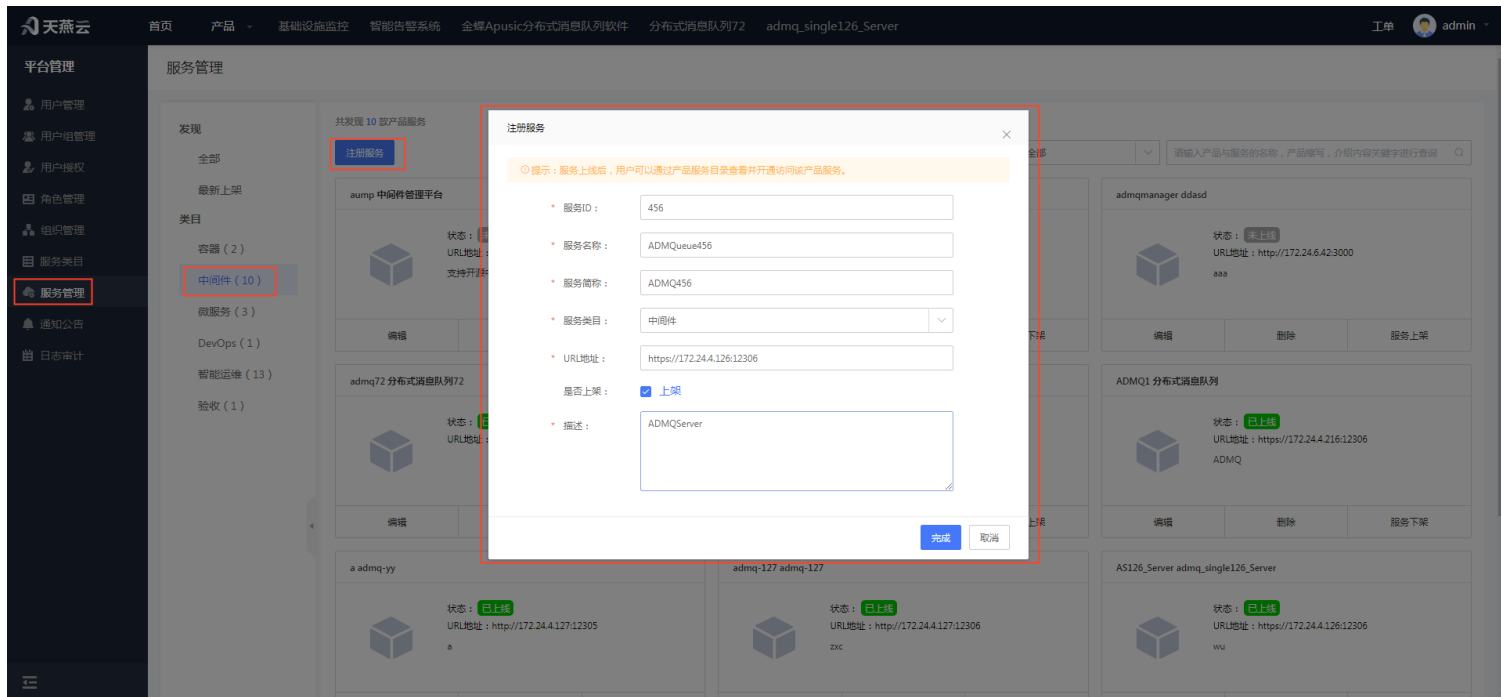


5.11.1.2 注册ADMQ服务并跳转到ADMQ界面

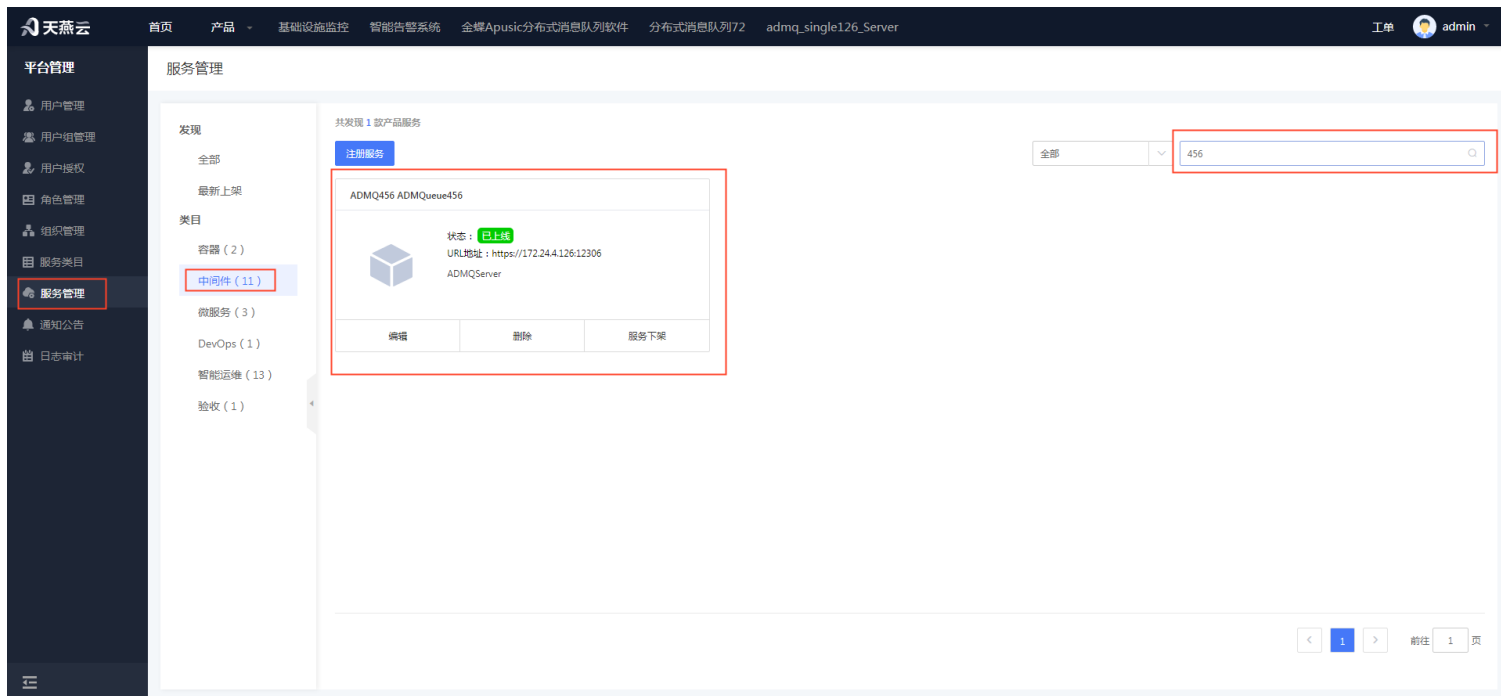
选择【admin】->【平台管理】，会跳转到平台管理页面。



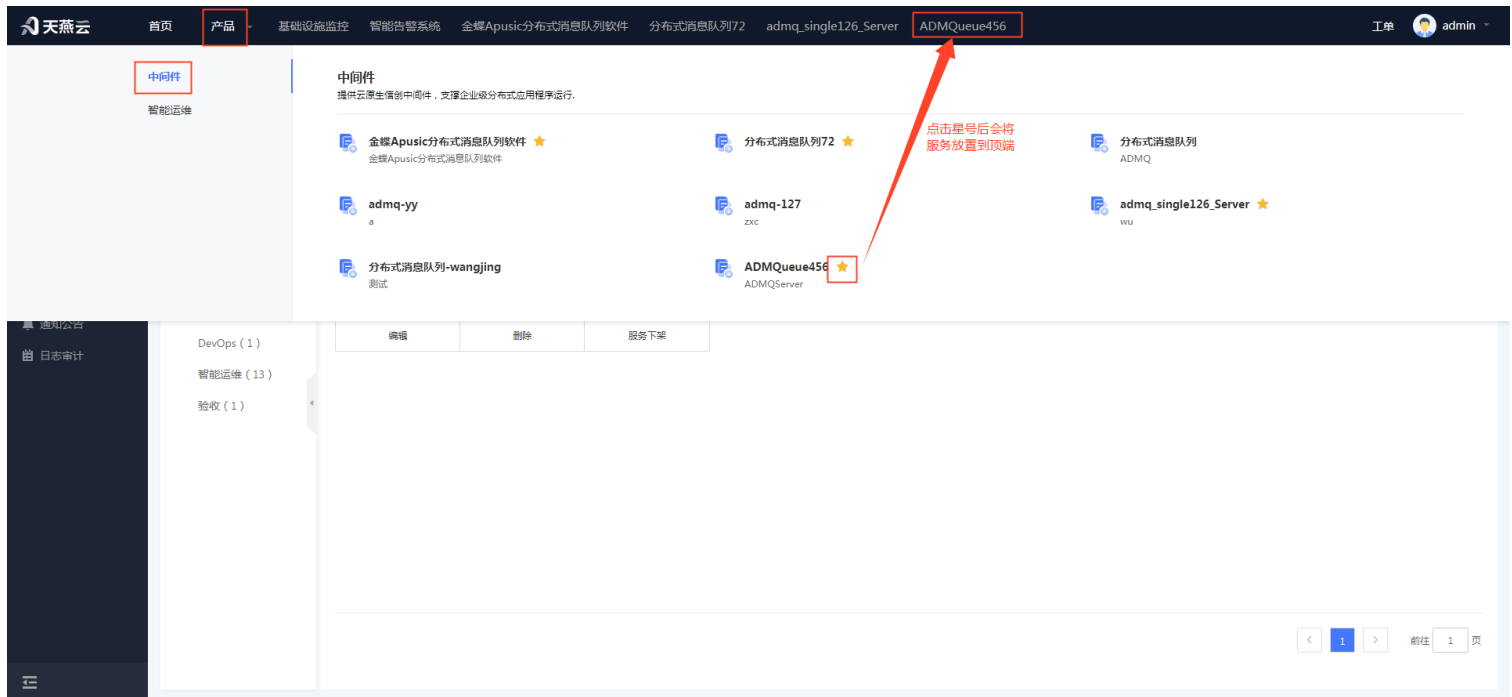
在平台管理页面，选择【服务管理】->类目选择【中间件】，点击“注册服务”按钮，弹出注册页面，填入配置项值，点击完成按钮，进行注册ADMQ服务。



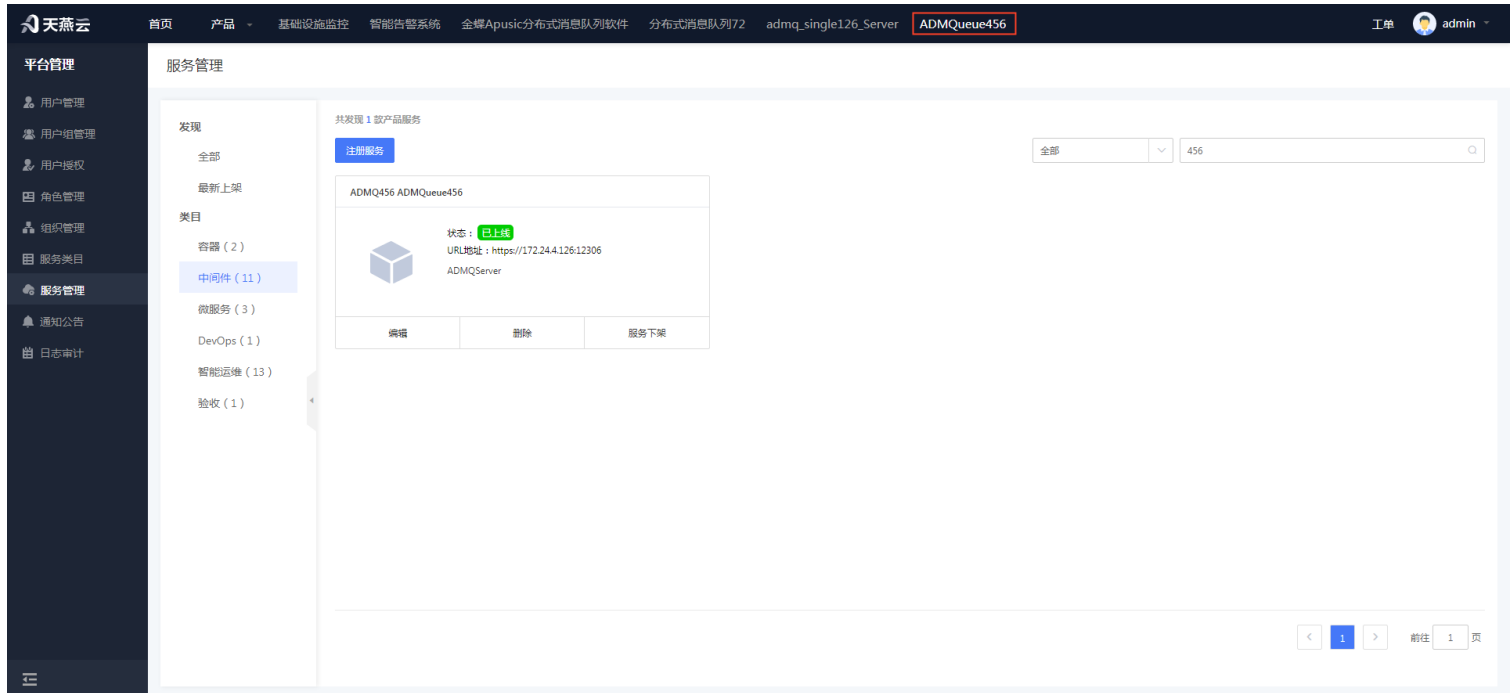
在平台管理页面，选择【服务管理】->类目选择【中间件】，右上角输入注册的服务名称，可查到已注册的具体服务：

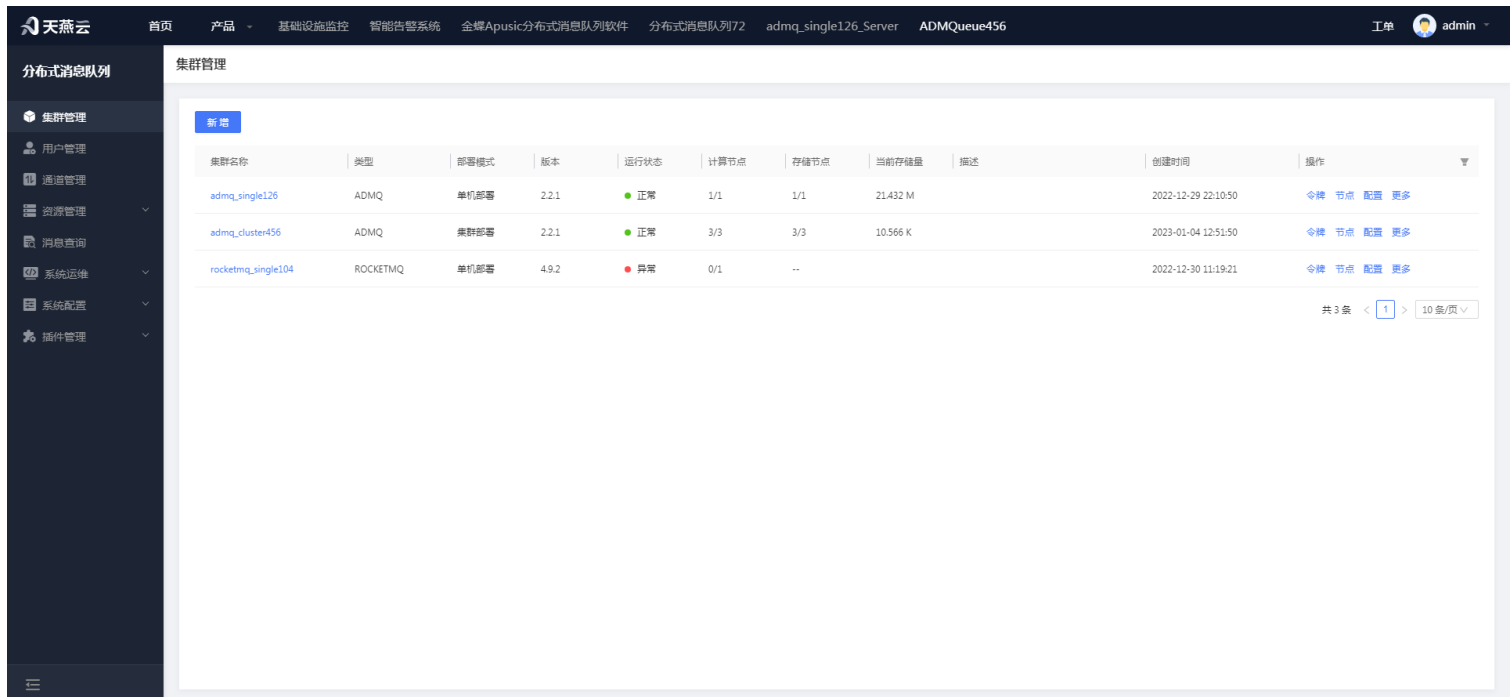


在平台管理页面，选择【产品】->【中间件】，找到注册的服务，放置到页面顶栏：

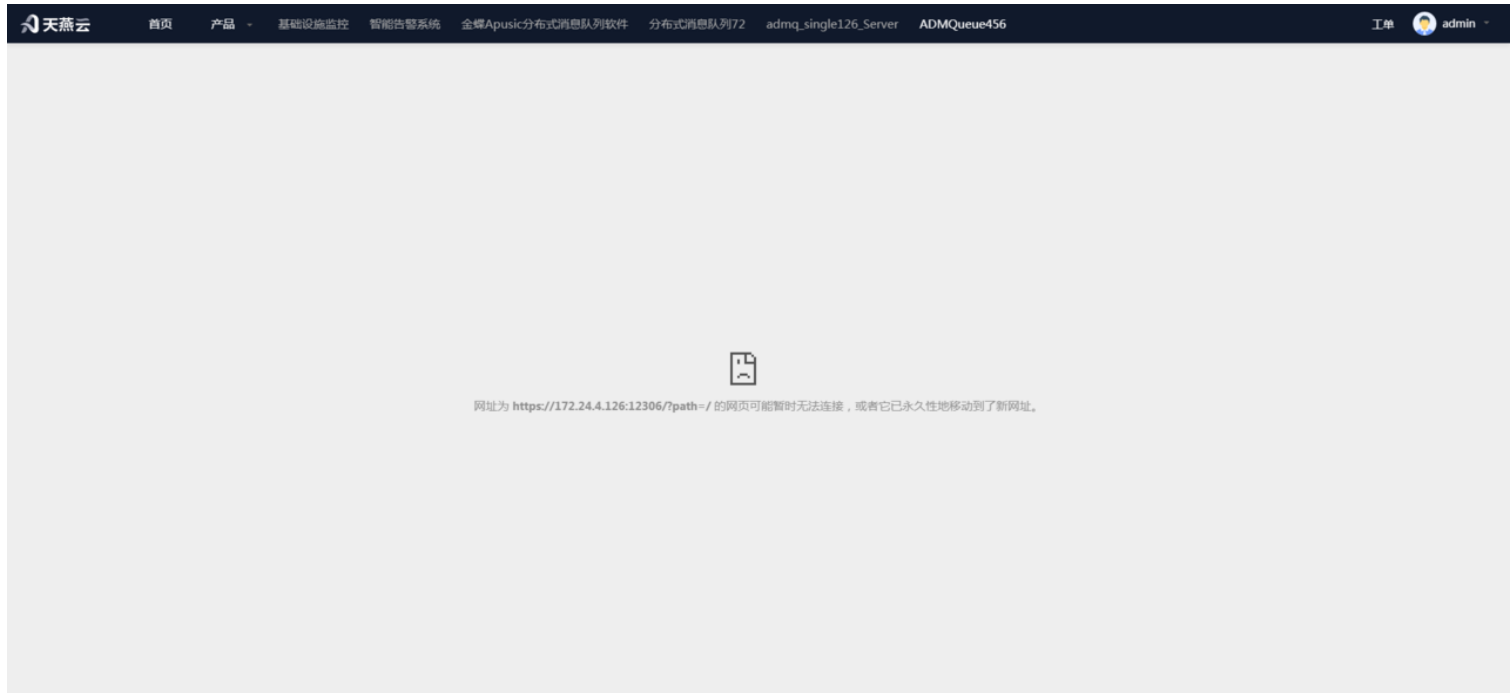


在顶栏点击服务名称，可跳转到ADMQ管控台界面：





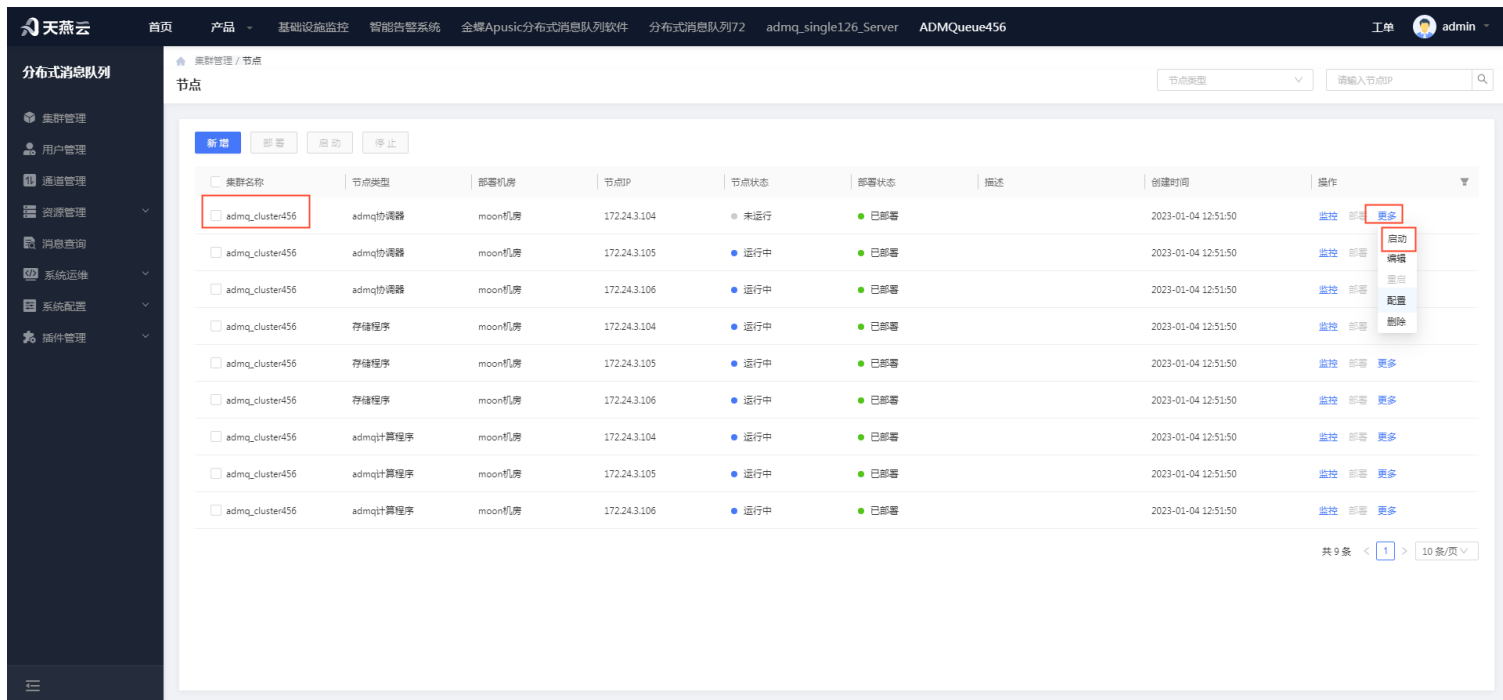
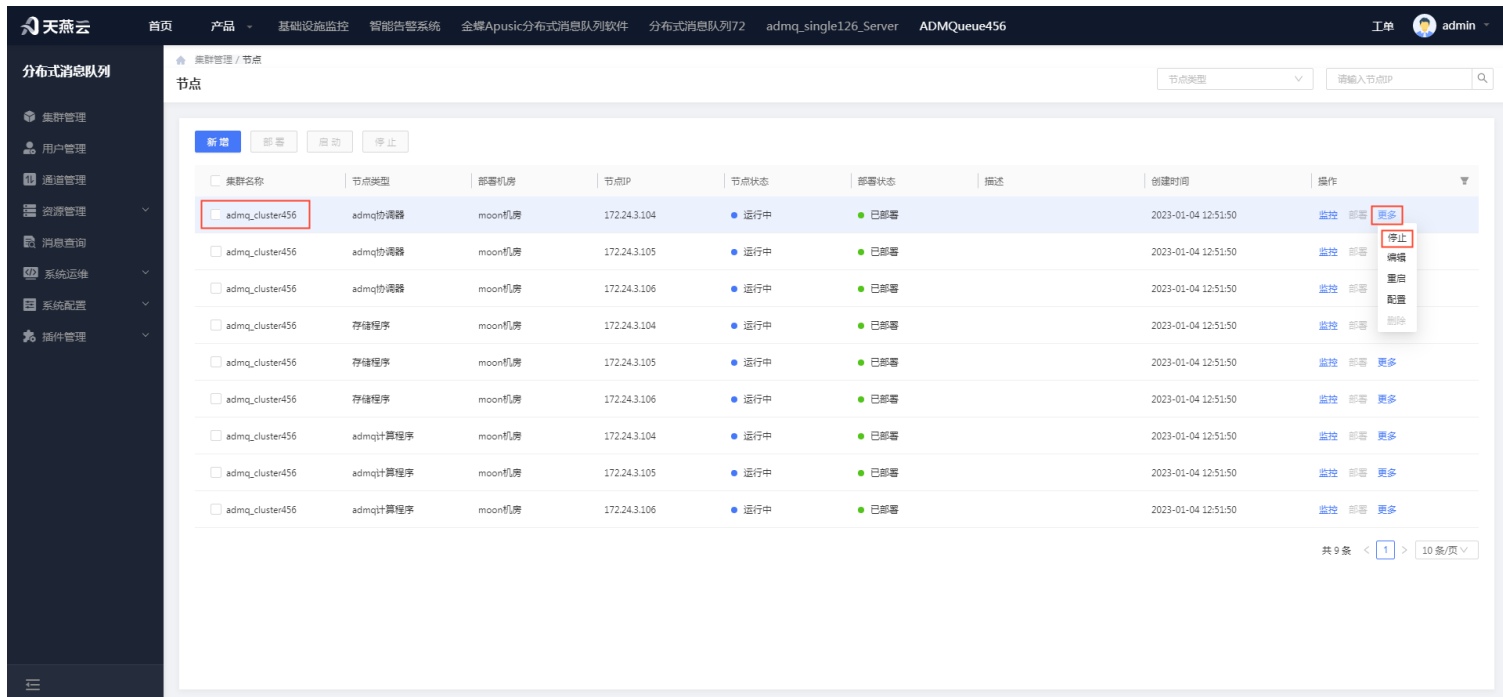
备注：若点击服务名称，跳转时出现如下界面：



需在浏览器上访问管控台URL：<https://172.24.4.126:12306/>，进入管控台登录界面（不用输入用户名、密码），然后重新点击服务名称，即可成功跳转到ADMQ管控管理界面。

5.11.1.3 节点跳转到监控页面

通过AUC跳转到ADMQ管控管理界面后，在集群列表选择需要监控的集群，点击节点按钮，进入到节点页面，选择需要监控的节点，【更多】里停止节点，然后启动节点：



然后对停止、启动后的节点，点击监控按钮，出现“节点监控”弹窗，点击“查看更多指标详情”按钮，即可跳转到AMP的节点监控界面，实时查看节点资源使用情况。

天燕云 首页 产品 基础设施监控 智能告警系统 金蝶Apusic分布式消息队列软件 分布式消息队列72 admq_single126_Server ADMQueue456 工单 admin

分布式消息队列

集群管理 / 节点

节点类型 请输入节点IP

新增 部署 启动 停止

集群名称	节点类型	部署机房	节点IP	节点状态	部署状态	描述	创建时间	操作
<input type="checkbox"/> admq_cluster456	admq协调器	moon机房	172.24.3.104	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	admq协调器	moon机房	172.24.3.105	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	admq协调器	moon机房	172.24.3.106	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	存储程序	moon机房	172.24.3.104	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	存储程序	moon机房	172.24.3.105	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	存储程序	moon机房	172.24.3.106	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	admq计算程序	moon机房	172.24.3.104	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	admq计算程序	moon机房	172.24.3.105	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	admq计算程序	moon机房	172.24.3.106	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多

共 9 条 < 1 > 10 条/页

天燕云 首页 产品 基础设施监控 智能告警系统 金蝶Apusic分布式消息队列软件 分布式消息队列72 admq_single126_Server ADMQueue456 工单 admin

分布式消息队列

集群管理 / 节点

节点类型 请输入节点IP

新增 部署 启动 停止

集群名称	节点类型	部署机房	节点IP	节点状态	部署状态	描述	创建时间	操作
<input type="checkbox"/> admq_cluster456	admq协调器	moon机房	172.24.3.104	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	admq协调器	moon机房	172.24.3.105	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	admq协调器	moon机房	172.24.3.106	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	存储程序	moon机房	172.24.3.104	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	存储程序	moon机房	172.24.3.105	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	存储程序	moon机房	172.24.3.106	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	admq计算程序	moon机房	172.24.3.104	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	admq计算程序	moon机房	172.24.3.105	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多
<input type="checkbox"/> admq_cluster456	admq计算程序	moon机房	172.24.3.106	运行中	已部署		2023-01-04 12:51:50	监控 部署 更多

节点监控

查看更多指标详情

实时 2023-01-04 18:01 → 2023-01-04 19:01 关闭 刷新

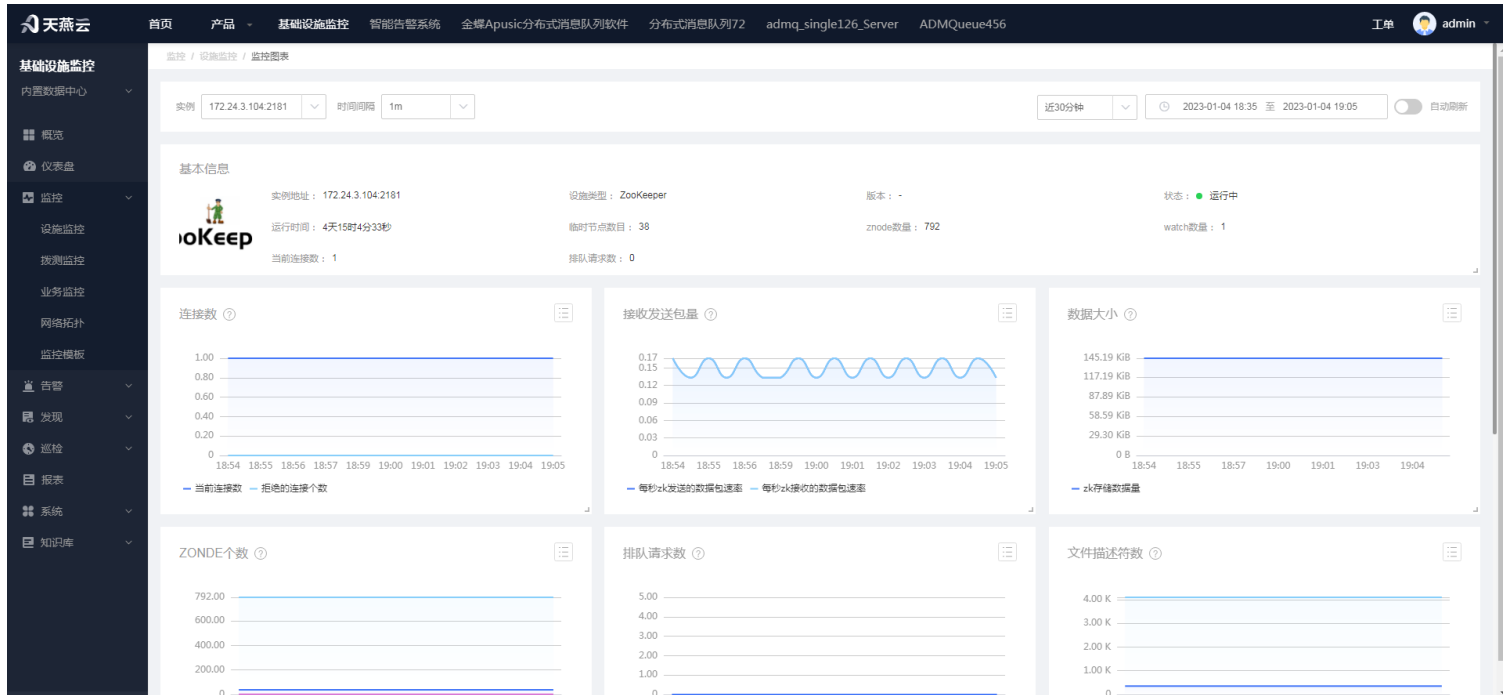
CPU使用率

暂无数据

内存使用率

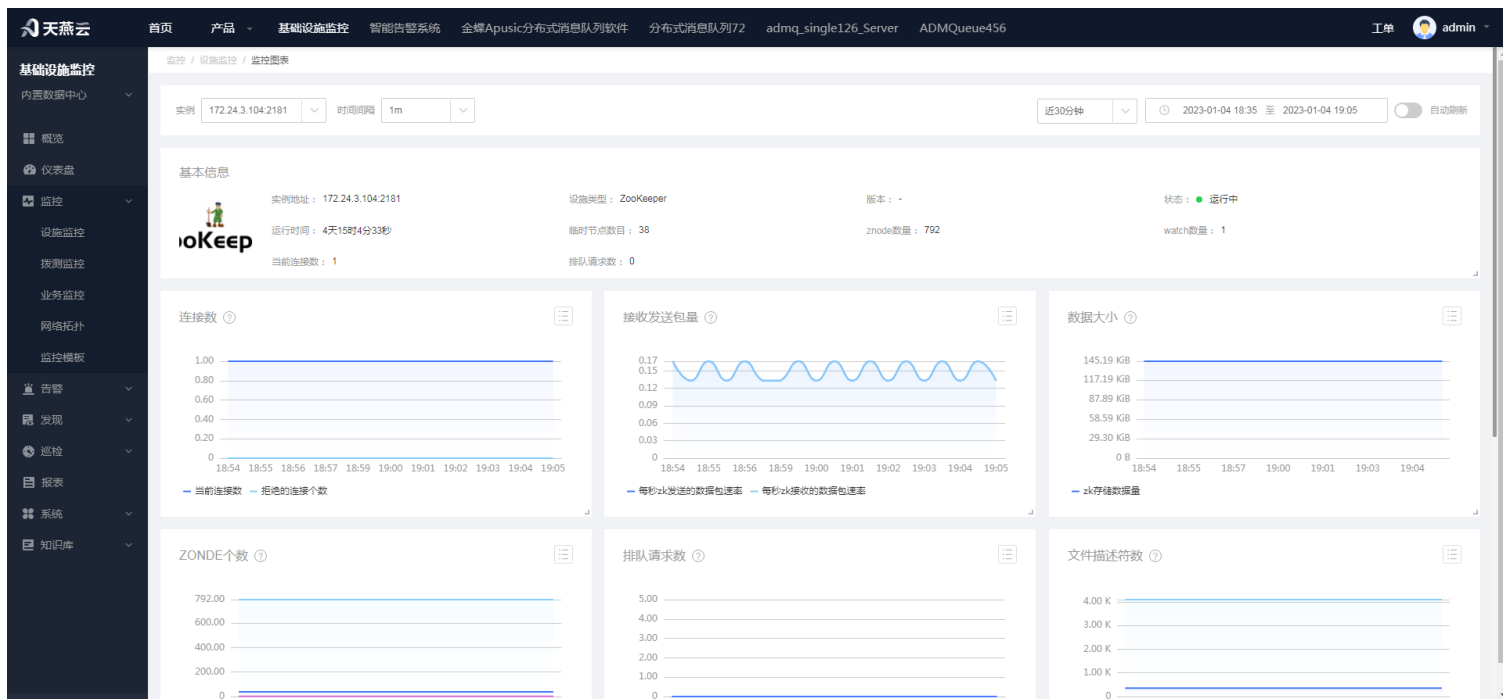
暂无数据

磁盘使用量

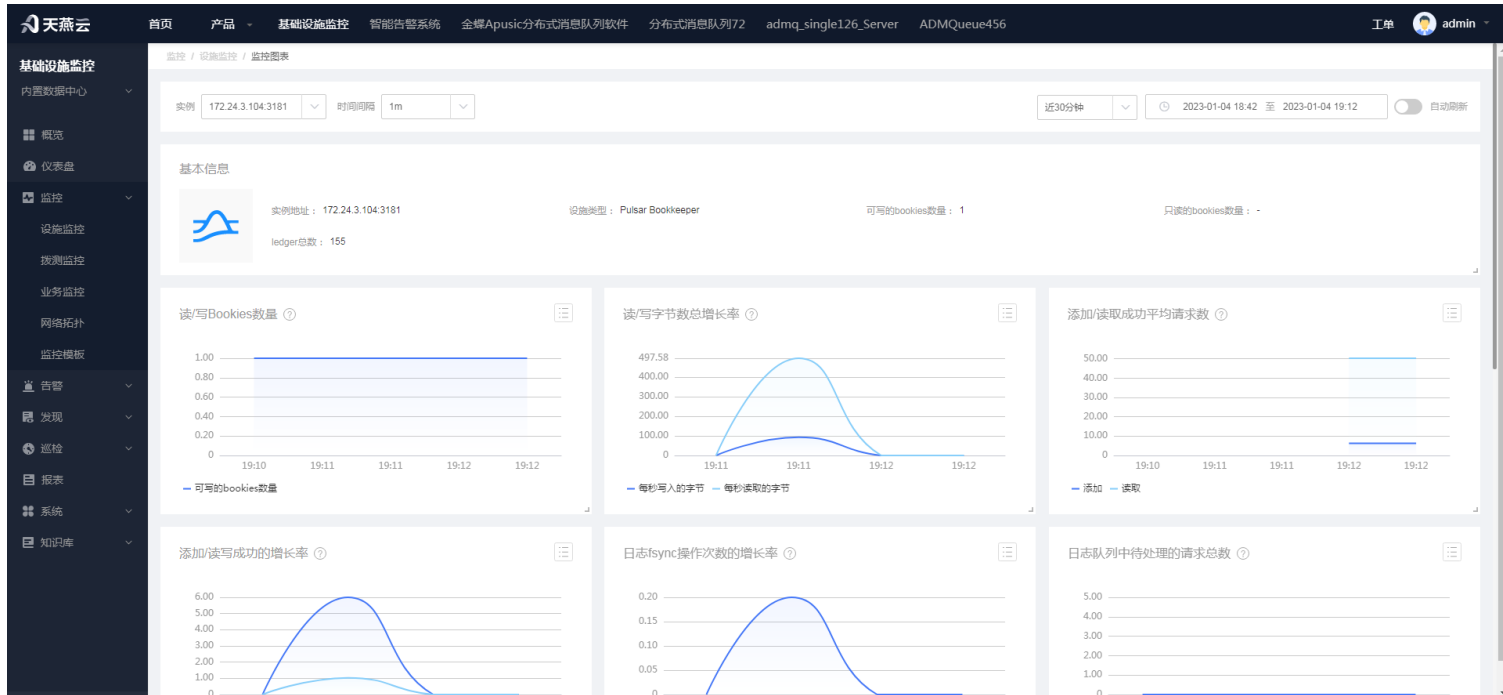


5.11.1.4 各个组件监控界面

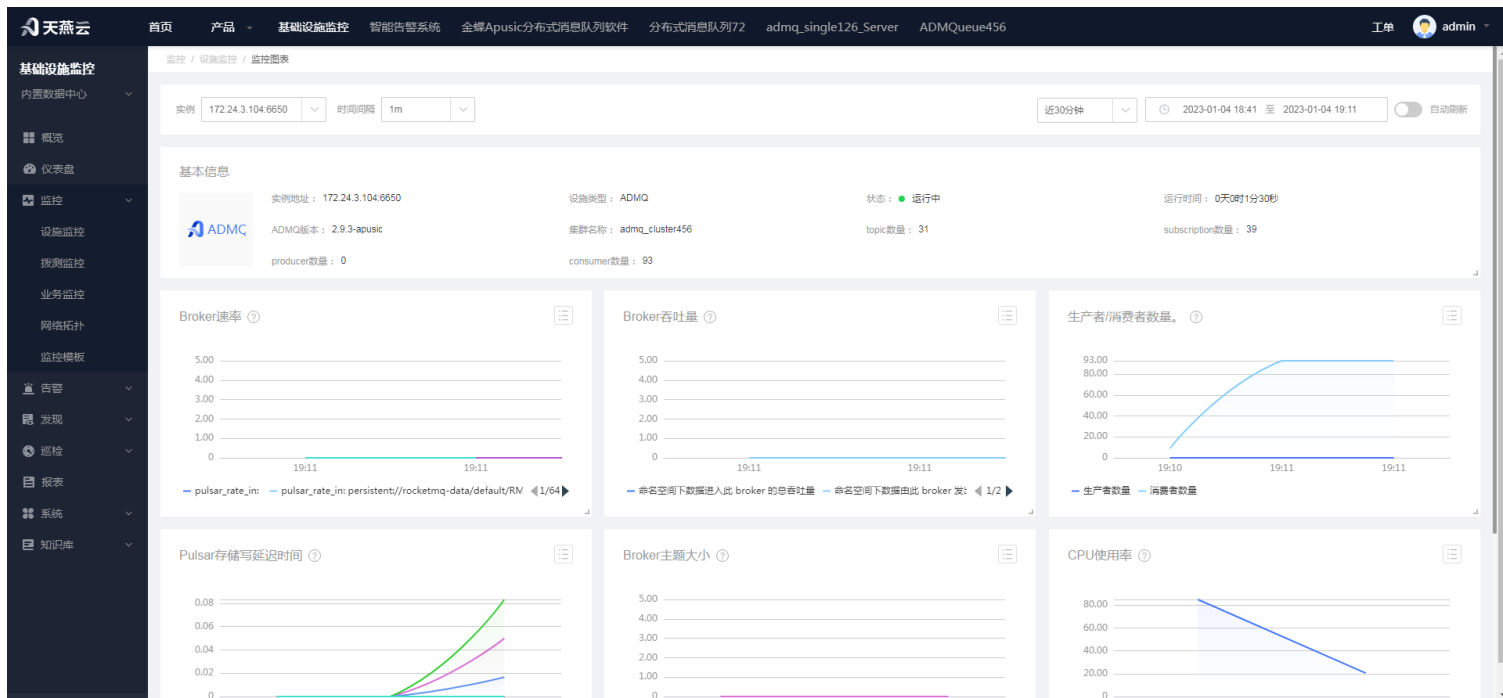
zookeeper节点监控界面:



storage节点监控界面:



broker节点监控界面:



5.11.2 监控指标

5.11.2.1 zookeeper监控指标

zookeeper 通过 Prometheus 接口提供了多个统计指标，AMP内置的模版对这些指标进行了汇总计算：

指标表达式	指标描述
zk_up	设施状态（运行：1，停止：0）
zk_server_leader	是否是leader（1是0否）

zk_ephemerals_count	zk的临时节点数目 (个)
zk_znode_count	zk的znode数量 (个)
zk_watch_count	zk的watch数目 (个)
zk_packets_sent	zk 发送的数据包速率 (个/s)
zk_packets_received	zk 接收的数据包速率 (个/s)
zk_num_alive_connections	当前连接数 (个)
zk_open_file_descriptor_count	打开文件描述符数 (个)
zk_max_file_descriptor_count	最大文件描述符数 (个)
zk_avg_latency	zk 处理平均延迟 (ms)
zk_min_latency	zk 处理最小时延 (ms)
zk_max_latency	zk 处理最大时延 (ms)
zk_outstanding_requests	排队请求数 (个)
zk_approximate_data_size	zk存储数据量 (bytes)

5.11.2.2 storage监控指标

storage通过 Prometheus 接口提供了多个统计指标，AMP内置的模版对这些指标进行了汇总计算：

指标表达式	指标描述
count(bookie_SERVER_STATUS == 1)	可写入Bookies数量 (个)
count(bookie_SERVER_STATUS == 0)	只读的Bookies数量 (个)
sum(irate(bookie_READ_BYTES[30s]))	读取字节数增长率 (MB/s)
sum(irate(bookie_WRITE_BYTES[30s]))	写入字节数增长率 (MB/s)
avg(bookkeeper_server_ADD_ENTRY_REQUEST{success="true", quantile="0.99"})	添加成功平均请求数 (个)
avg(bookkeeper_server_READ_ENTRY_REQUEST{success="true", quantile="0.99"})	读取成功平均请求数 (个)
sum(irate(bookkeeper_server_ADD_ENTRY_count{success="true"}[30s]))	添加entry成功增长率 (个/s)
sum(irate(bookkeeper_server_READ_ENTRY_count{success="true"}[30s]))	读取entry成功增长率 (个/s)
sum(irate(bookie_journal_JOURNAL_SYNC_count{success="true"}[30s]))	日志sync操作次数的总增长率 (次/s)
sum(bookie_journal_JOURNAL_QUEUE_SIZE)	日志队列中待处理的请求总数 (个)
sum(bookie_journal_JOURNAL_FORCE_WRITE_QUEUE_SIZE)	强制写入队列中等待的请求总数 (个)
sum(bookie_journal_JOURNAL_CB_QUEUE_SIZE)	回调队列中待处理的回调总数 (个)
sum(bookie_ledgers_count)	bookie中存储的ledger总数 (个)
sum(bookie_entries_count)	bookie中存储的条目总数 (个)
sum(bookie_read_cache_size)	bookie读取缓存大小 (byte)
sum(bookie_write_cache_size)	bookie写入缓存大小 (byte)
sum(bookie_DELETED_LEDGER_COUNT)	bookie启动以来被删除的ledger总数 (个)
sum(bookie_ledger_writable_dirs)	bookie中可写入目录的数量 (个)

5.11.2.3 broker监控指标

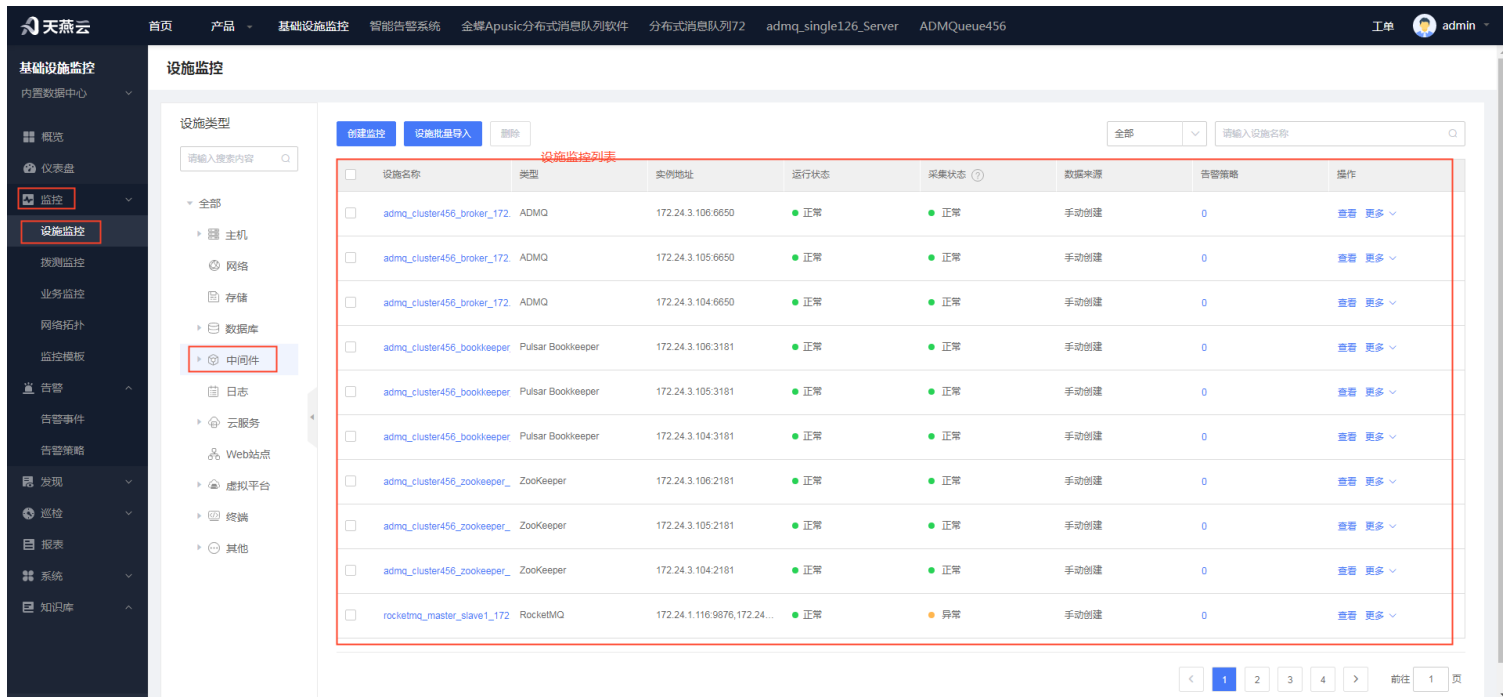
Broker通过 Prometheus 接口提供了多个统计指标，AMP内置的模版对这些指标进行了汇总计算：

指标表达式	指标描述
amp_pulsar_server_status	设施状态
amp_pulsar_namespace_count	命名空间数 (个)
amp_pulsar_topics_count	主题数 (个)
amp_pulsar_io_inRate	消息接收频率 (消息/秒)
amp_pulsar_io_outRate	消息发出频率 (消息/秒)
amp_pulsar_throughput_inSize	流入吞吐量 (字节/秒)
amp_pulsar_throughput_outSize	流出吞吐量 (字节/秒)
amp_pulsar_storage_size	所有消息占用的总存储空间大小
amp_pulsar_entrySizeLeOverflow_Count	大小超1M的条目数
amp_pulsar_storage_writeRate	存储写频率 (条目/秒)
amp_pulsar_storage_readRate	存储读频率 (条目/秒)
amp_pulsar_cpu_usage	cpu利用率 (%)
amp_pulsar_jvm_memoryCommittedSize	jvm已提交内存大小 (MB)
amp_pulsar_jvm_memoryUsedSize	jvm已使用内存 (MB)
amp_pulsar_jvm_memoryMaxSize	jvm最大内存 (MB)
amp_pulsar_jvm_memoryUsage	jvm内存使用率 (%)

5.11.3 AMP配置告警策略

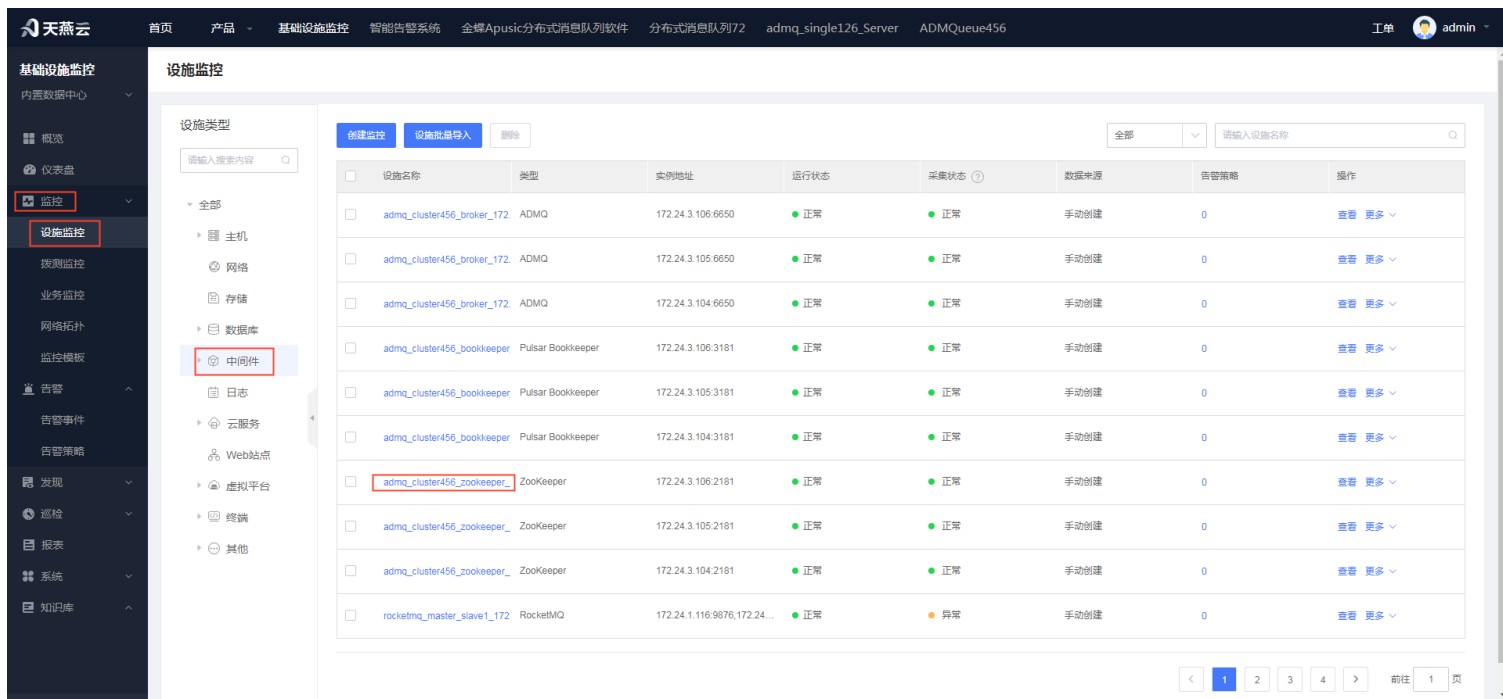
在设施监控界面点击具体的设施名称或者“查看”进入具体的一个设施，切换到“告警策略”页签，可以对该设施进行添加管理告警策略，可以查看该设施的目前创建的告警策略，该位置维护该设施的报警策略，也可以在“告警”模块下的“告警策略”下创建管理告警策略。

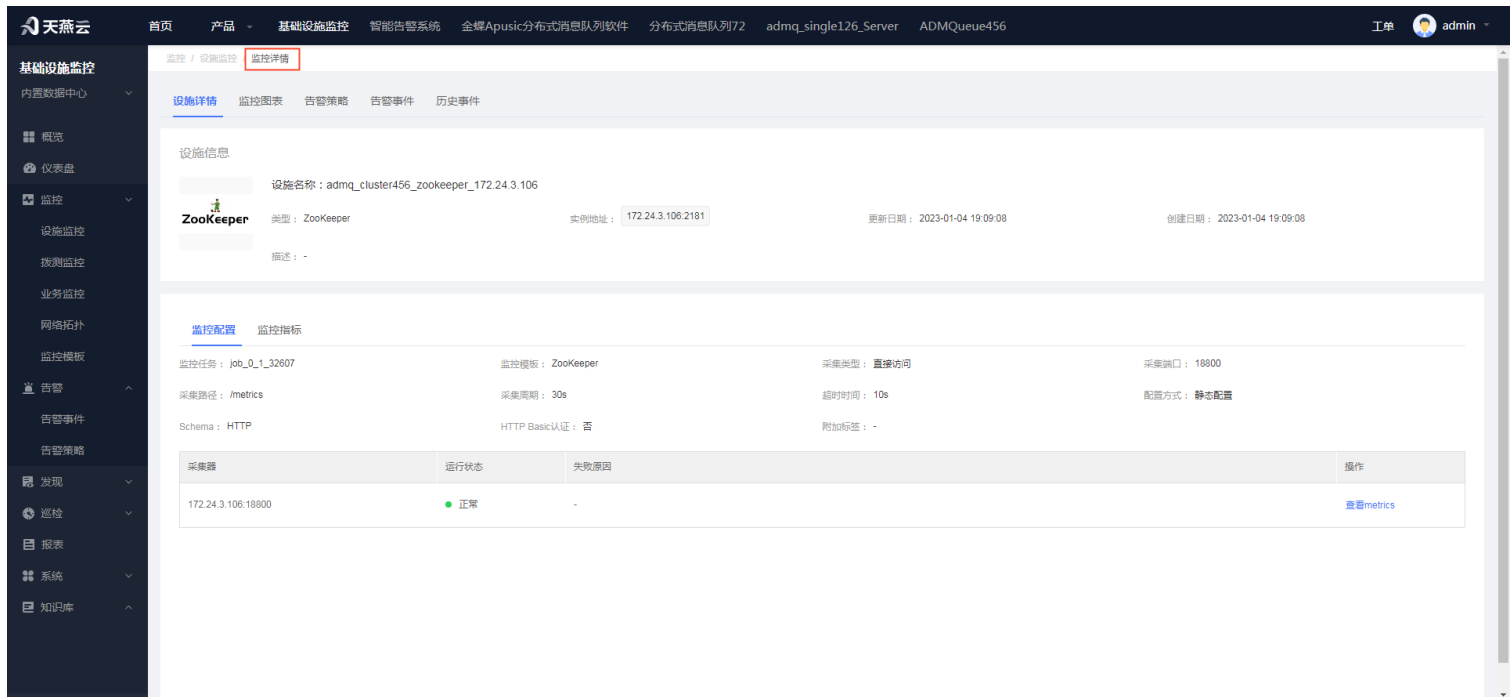
【监控】->【设施监控】->【中间件】，进入到设施监控列表：



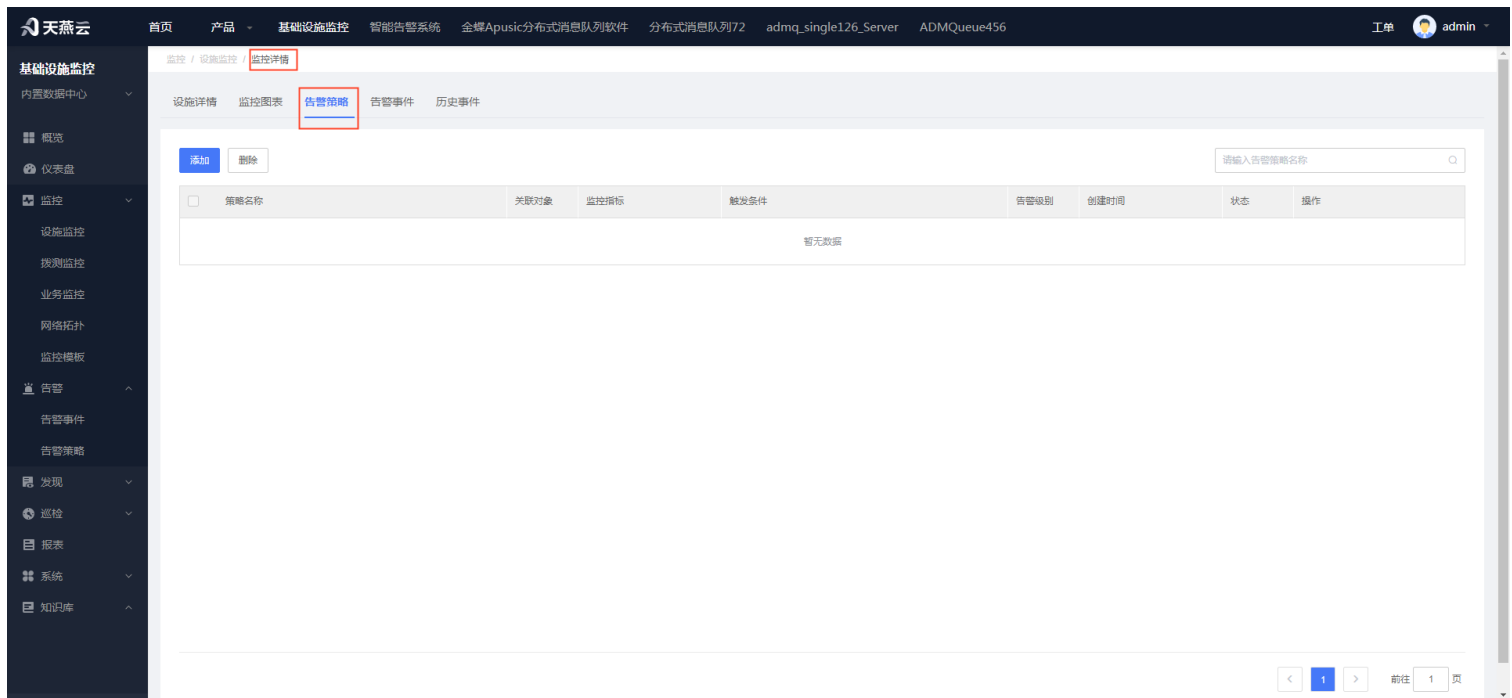
5.11.3.1 配置zookeeper告警策略

在设施监控列表点击集群某个zookeeper节点名称，进入到zookeeper的监控详情页面：

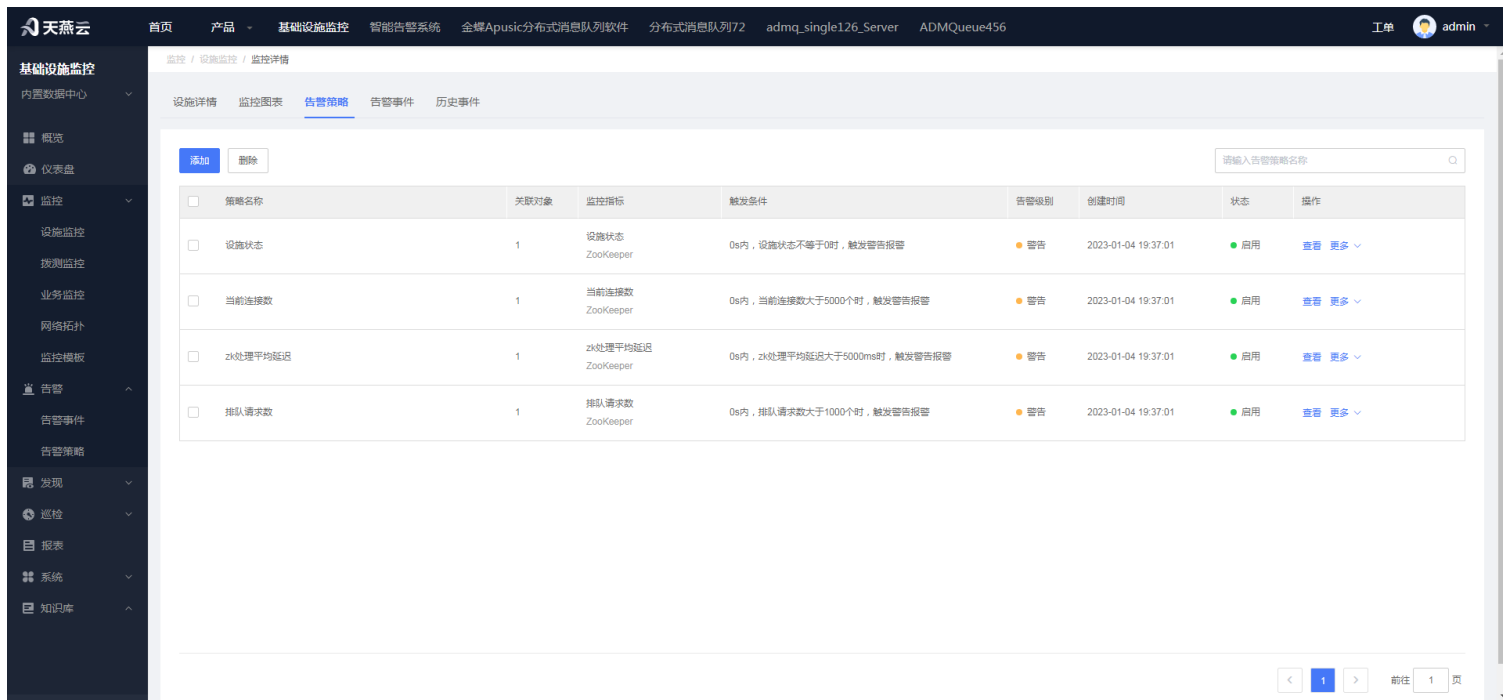
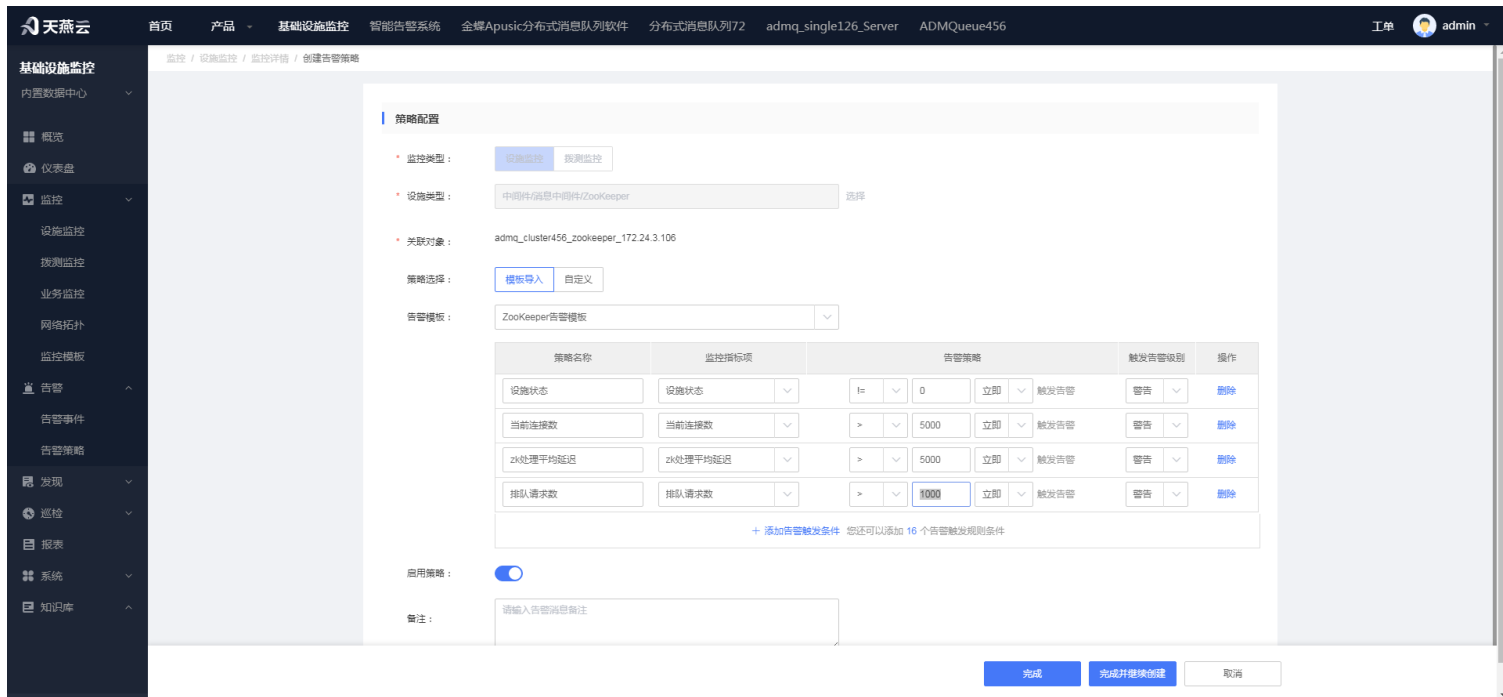




在监控详情页面，切换到“告警策略”页签：



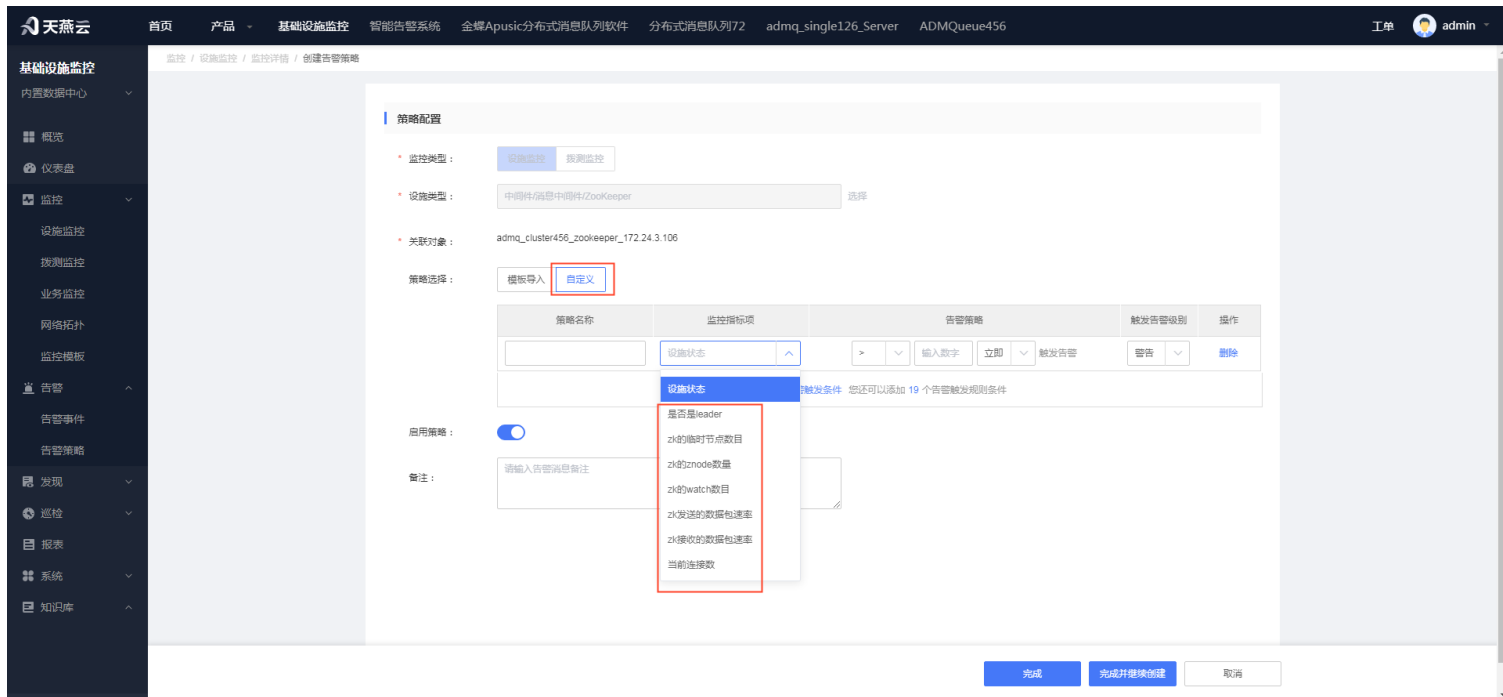
点击“添加”按钮后创建zookeeper告警策略：



AMP内置的Zookeeper告警模版已经包含了4条告警策略:

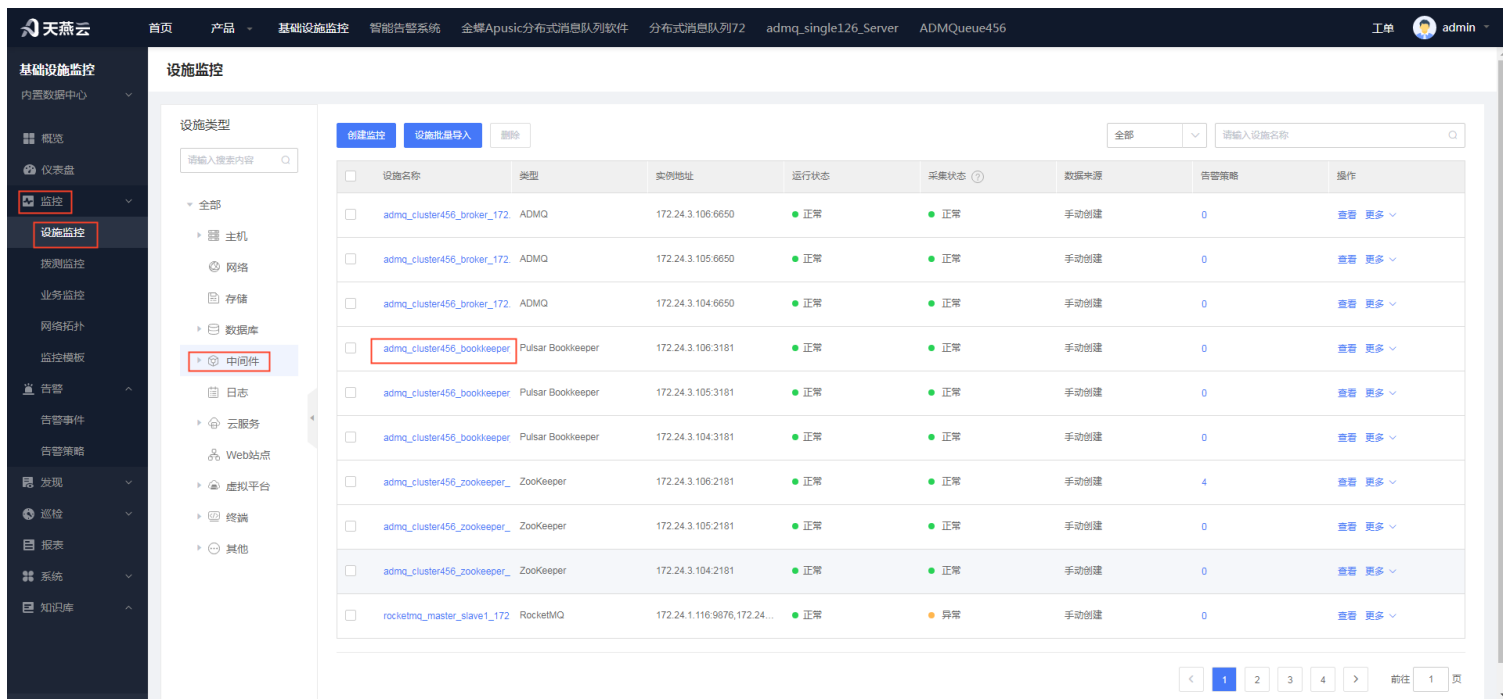
策略名称	监控项指标	告警触发条件
设施状态	设施状态	0s内, 设施状态不等于0时, 触发警告报警
当前连接数	当前连接数	0s内, 当前连接数大于5000个时, 触发警告报警
zk处理平均延迟	zk处理平均延迟	0s内, zk处理平均延迟大于5000ms时, 触发警告报警
排队请求数	排队请求数	0s内, 排队请求数大于1000个时, 触发警告报警

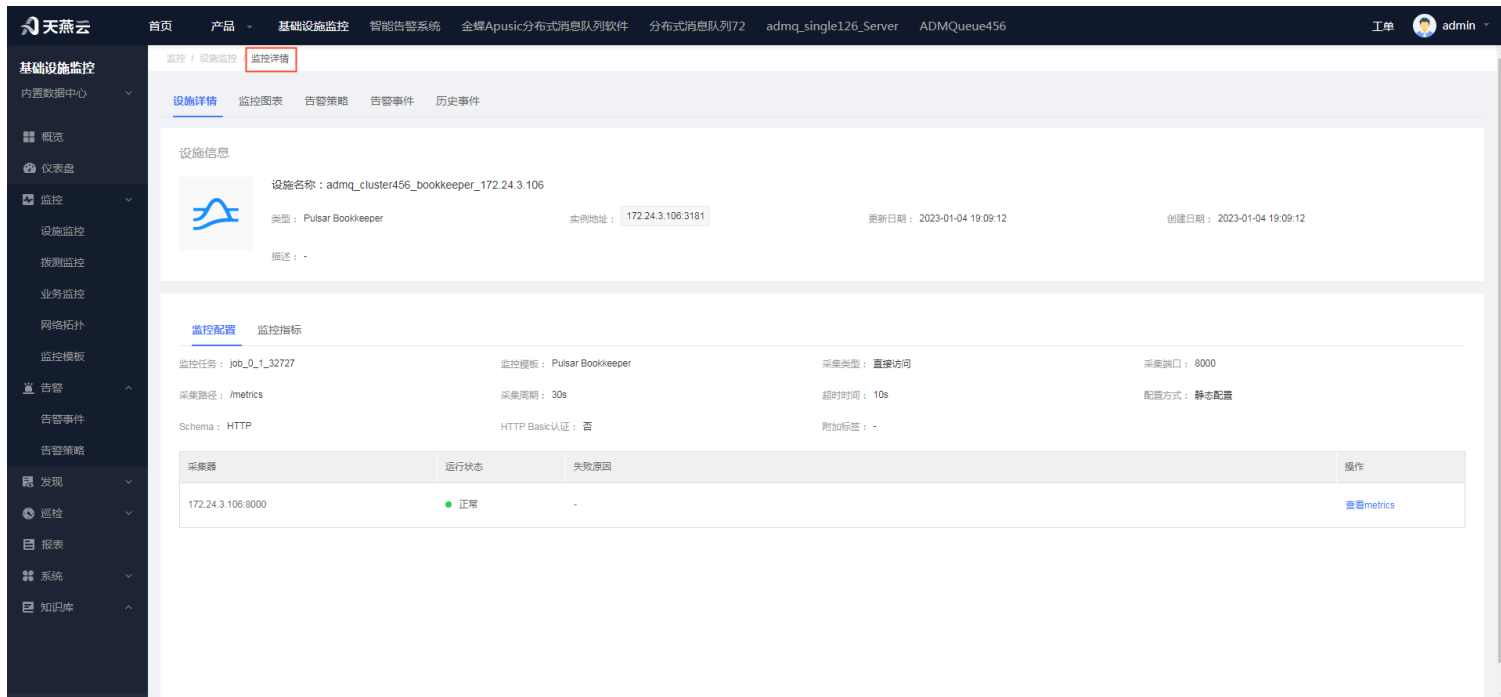
用户也可以根据zookeeper的监控指标项添加自定义告警策略:



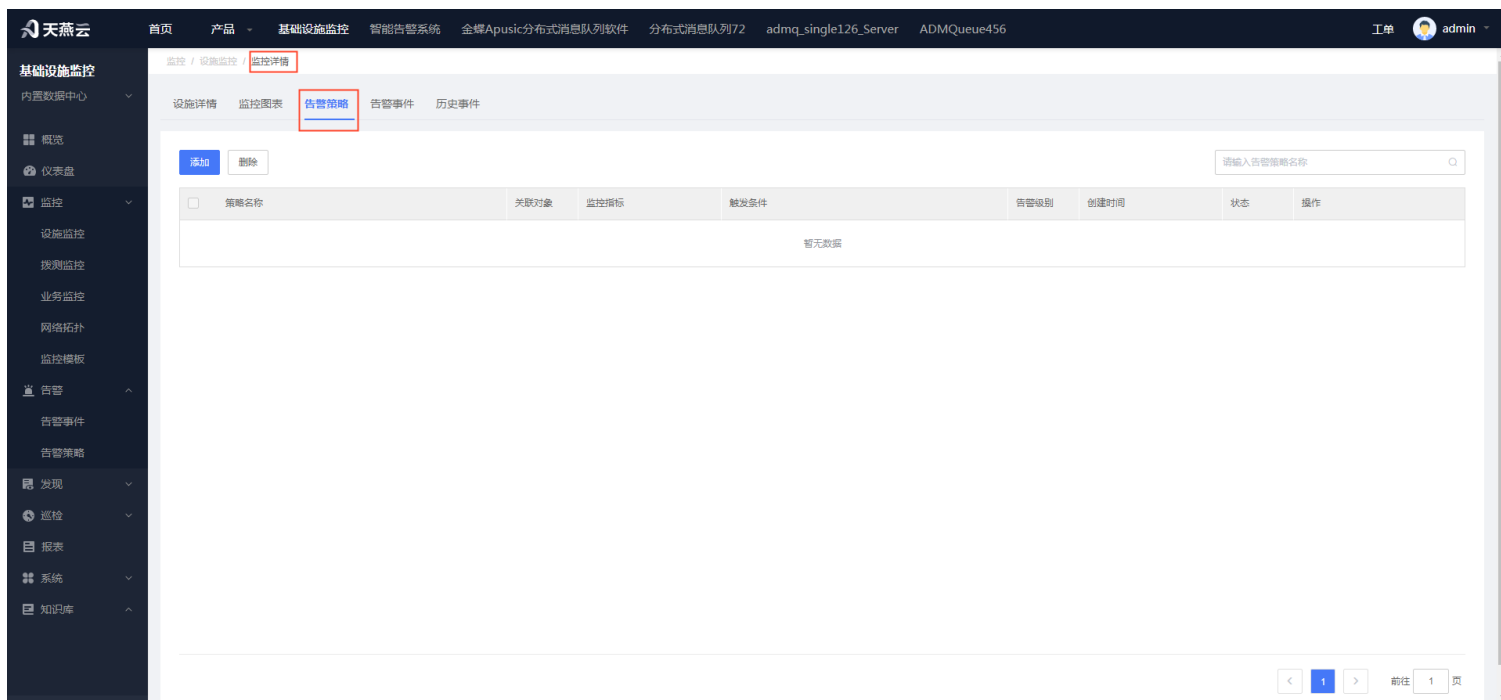
5.11.3.2 配置storage告警策略

在设施监控列表点击集群某个zookeeper节点名称，进入到zookeeper的监控详情页面：

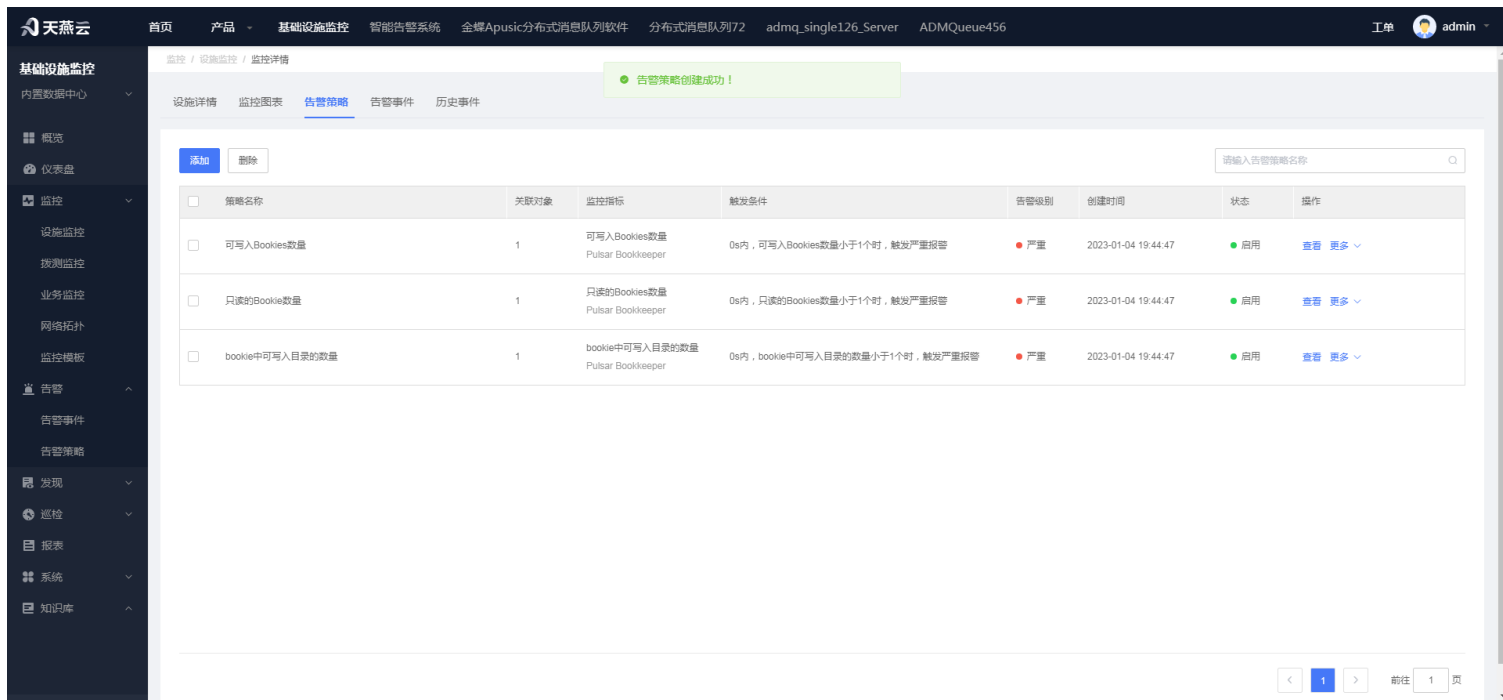
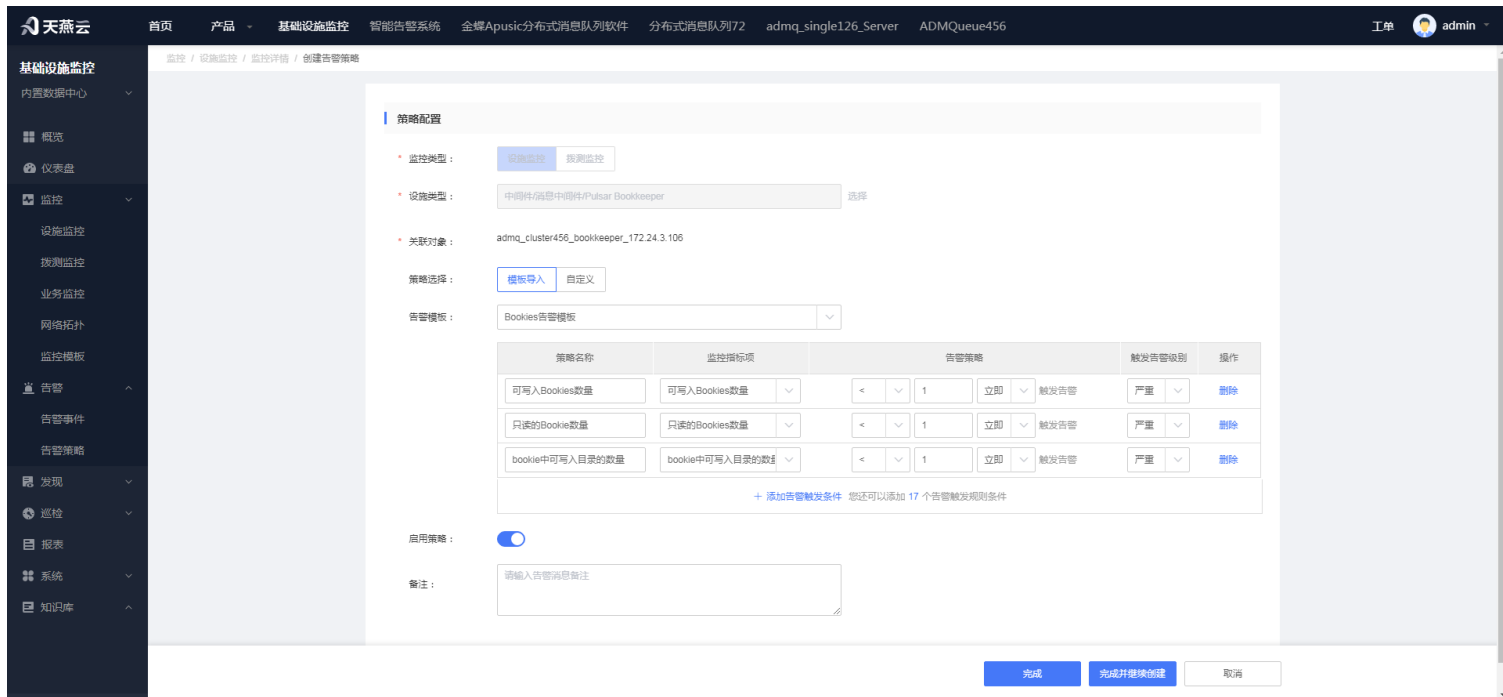




在监控详情页面，切换到“告警策略”页签：



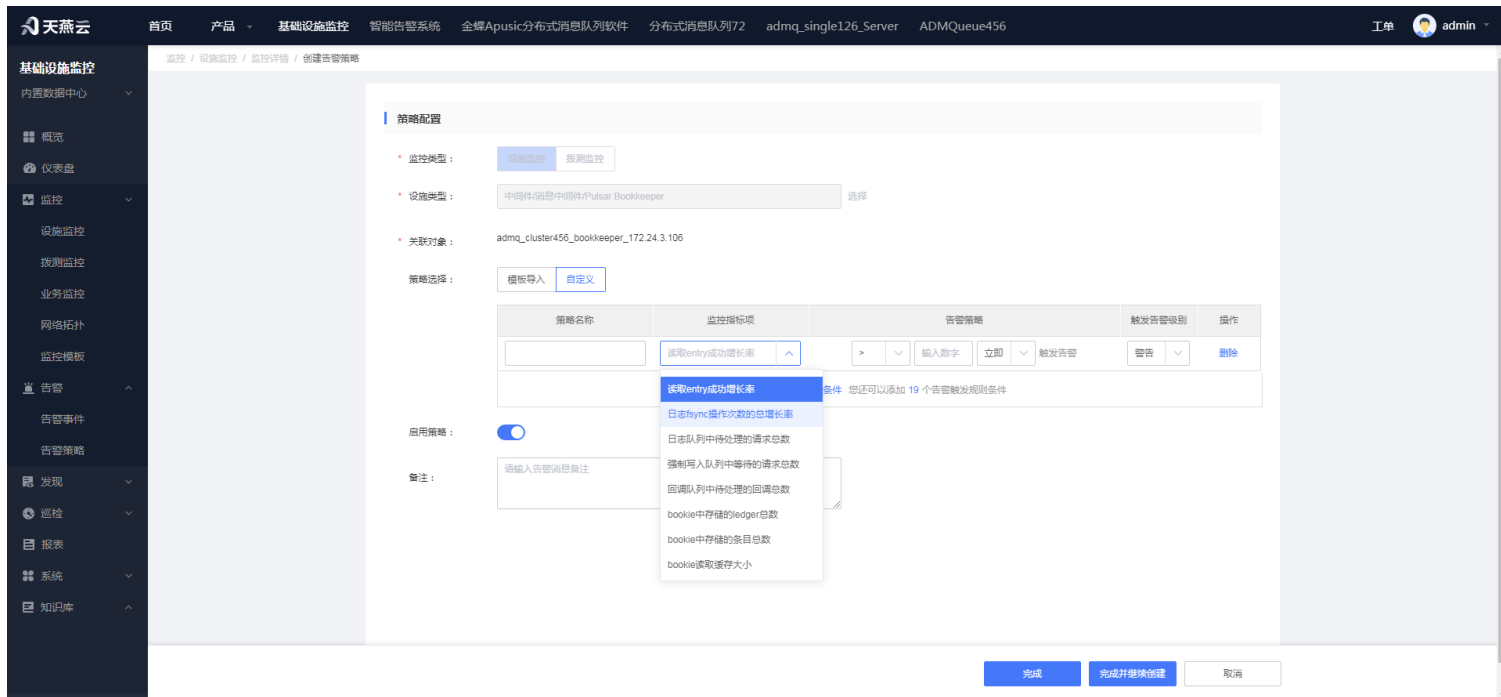
点击“添加”按钮后创建storage告警策略：



AMP内置的bookkeeper告警模版已经包含了3条告警策略：

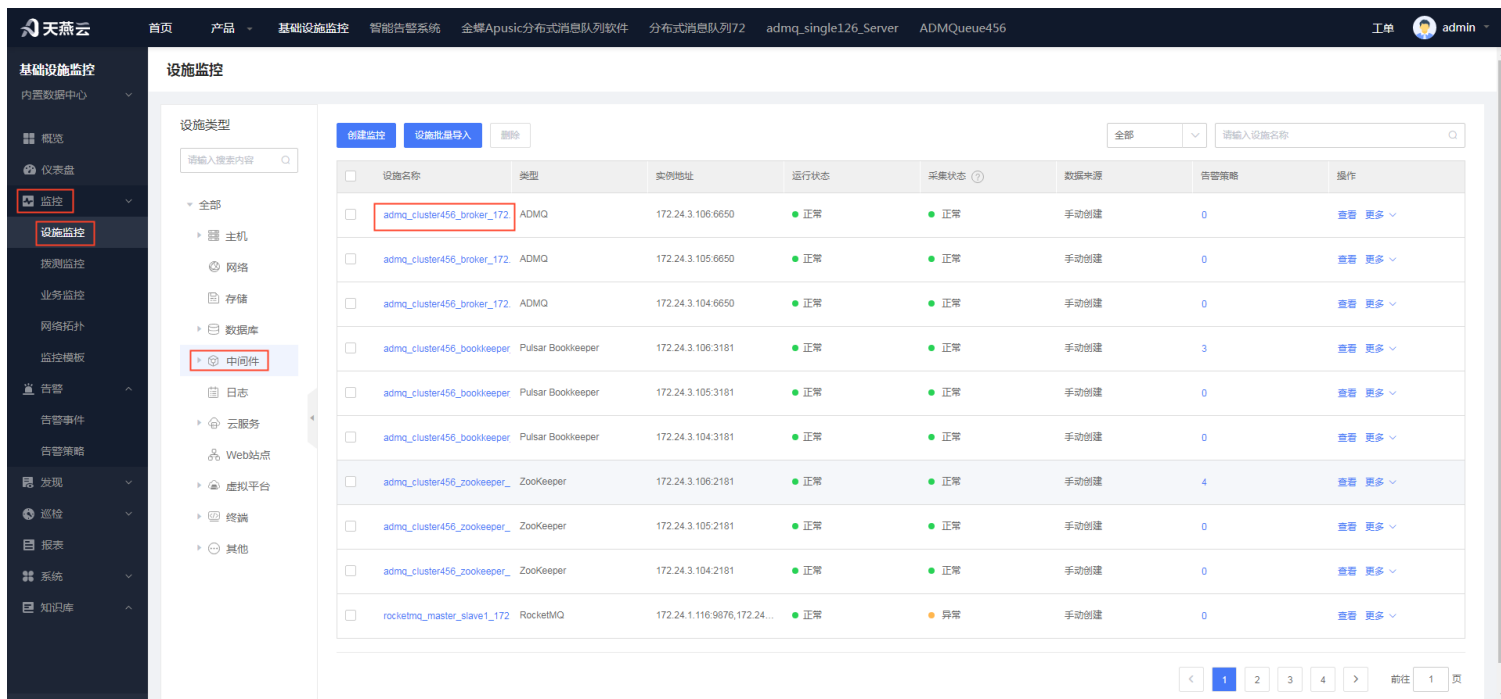
策略名称	监控项指标	告警触发条件
可写入Bookies数量	可写入Bookies数量	0s内，可写入Bookies数量小于1个时，触发严重报警
只读的Bookie数量	只读的Bookie数量	0s内，只读的Bookies数量小于1个时，触发严重报警
bookie中可写入目录的数量	bookie中可写入目录的数量	0s内，bookie中可写入目录的数量小于1个时，触发严重报警

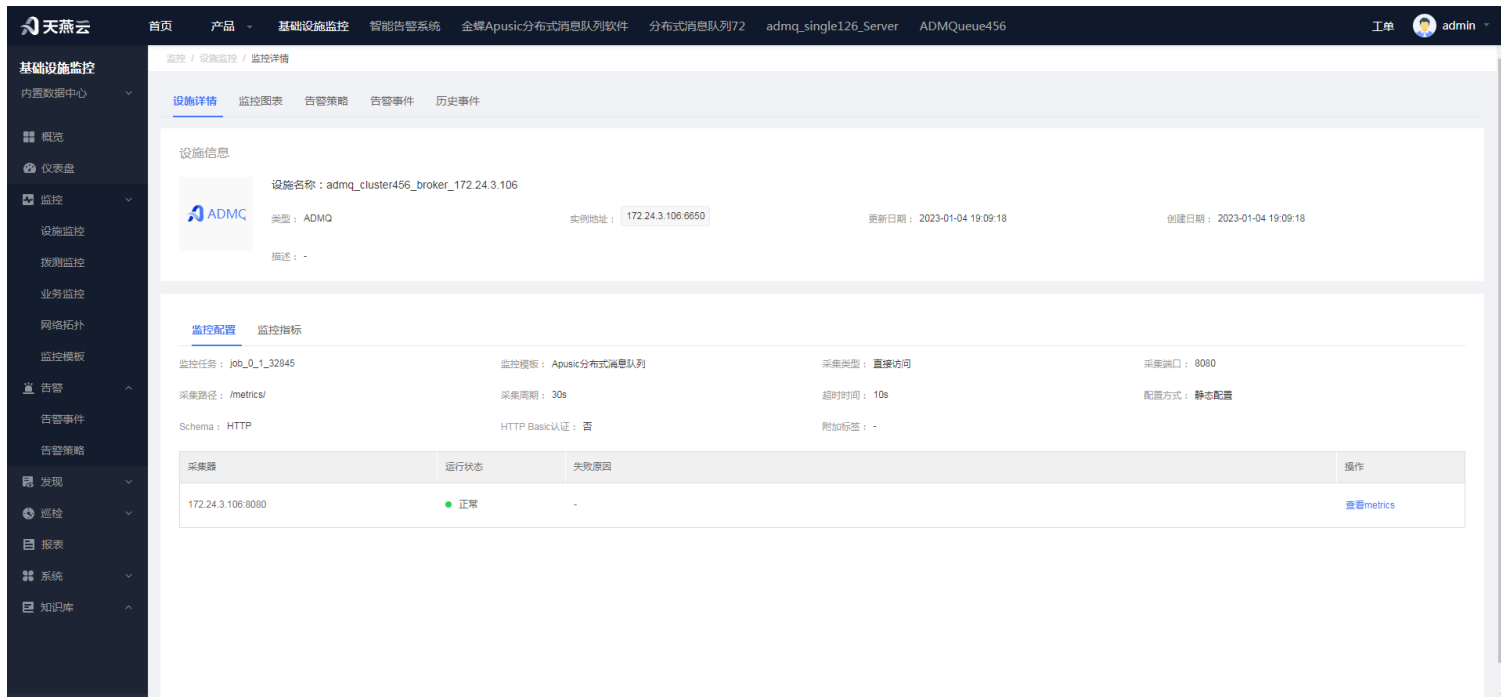
用户也可以根据bookkeeper的监控指标项添加自定义告警策略：



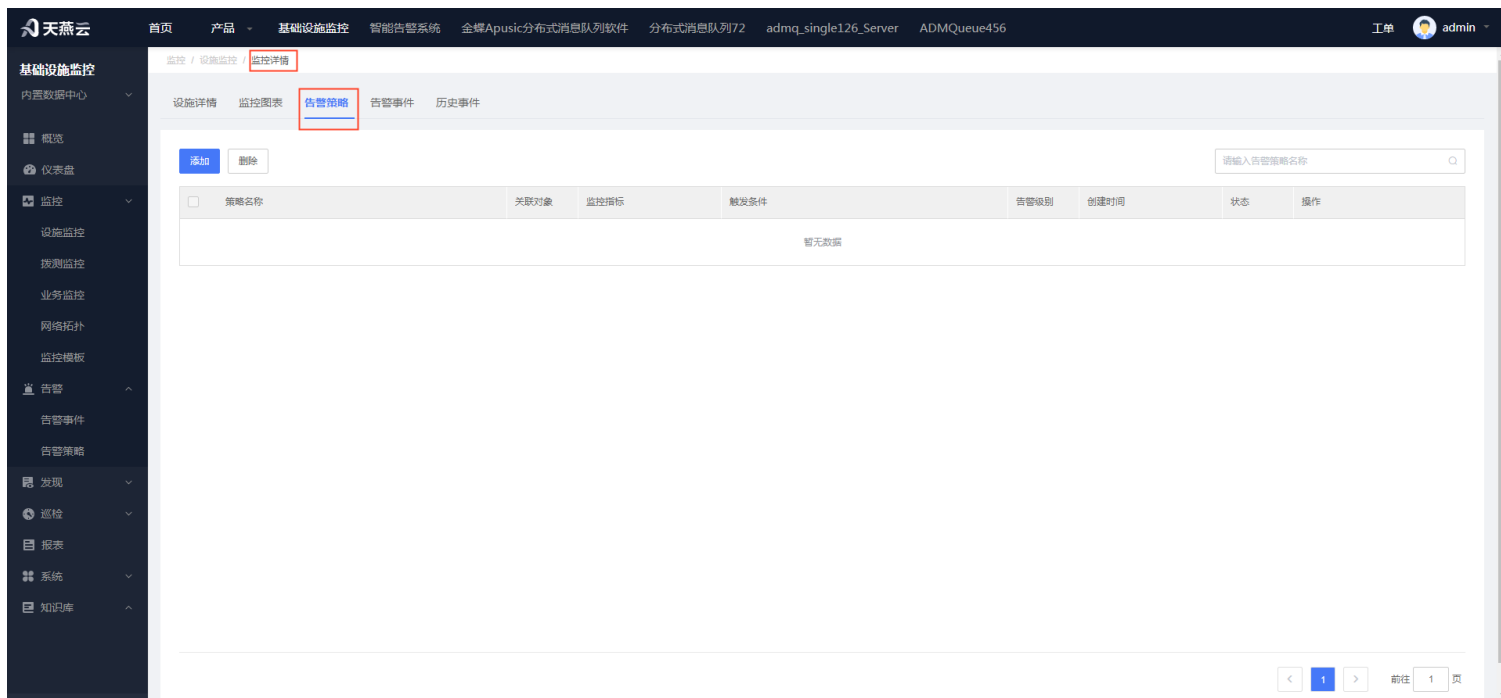
5.11.3.3 配置broker告警策略

在设施监控列表点击集群某个zookeeper节点名称，进入到zookeeper的监控详情页面：

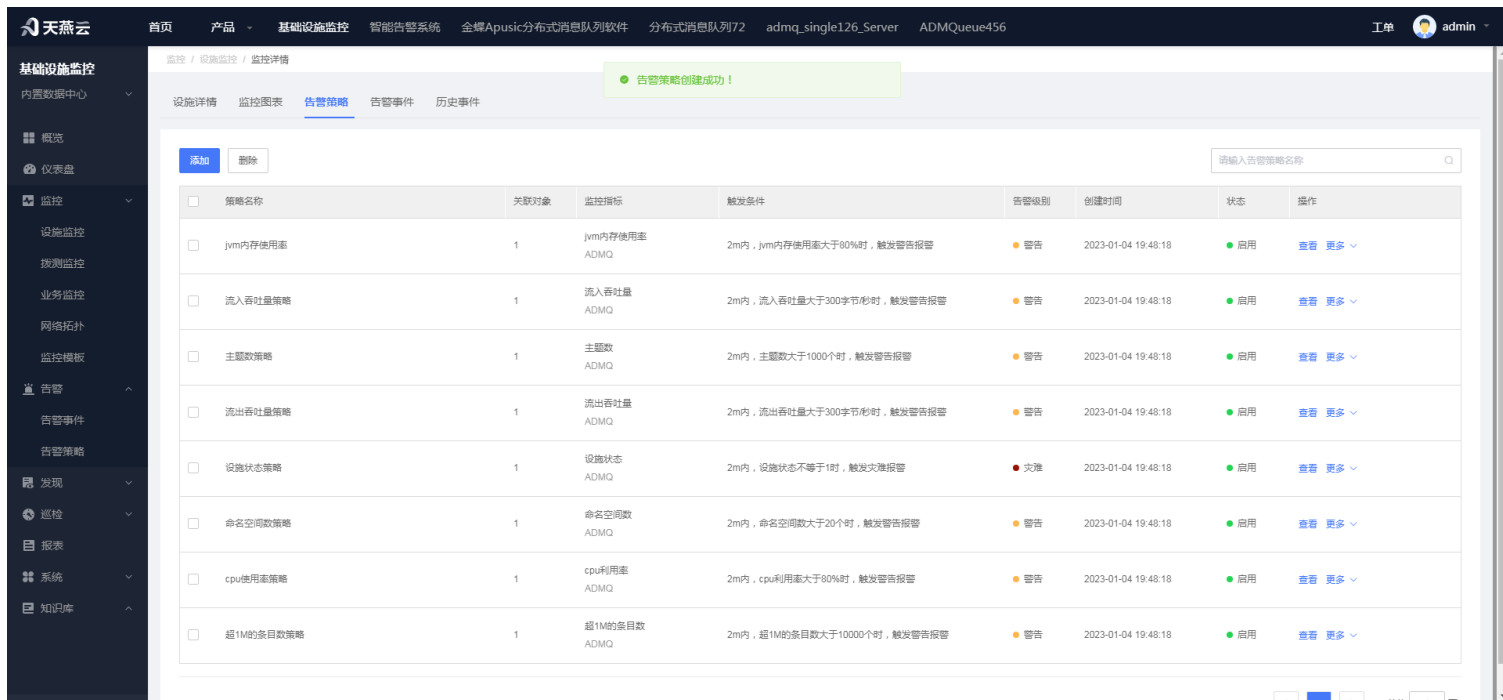
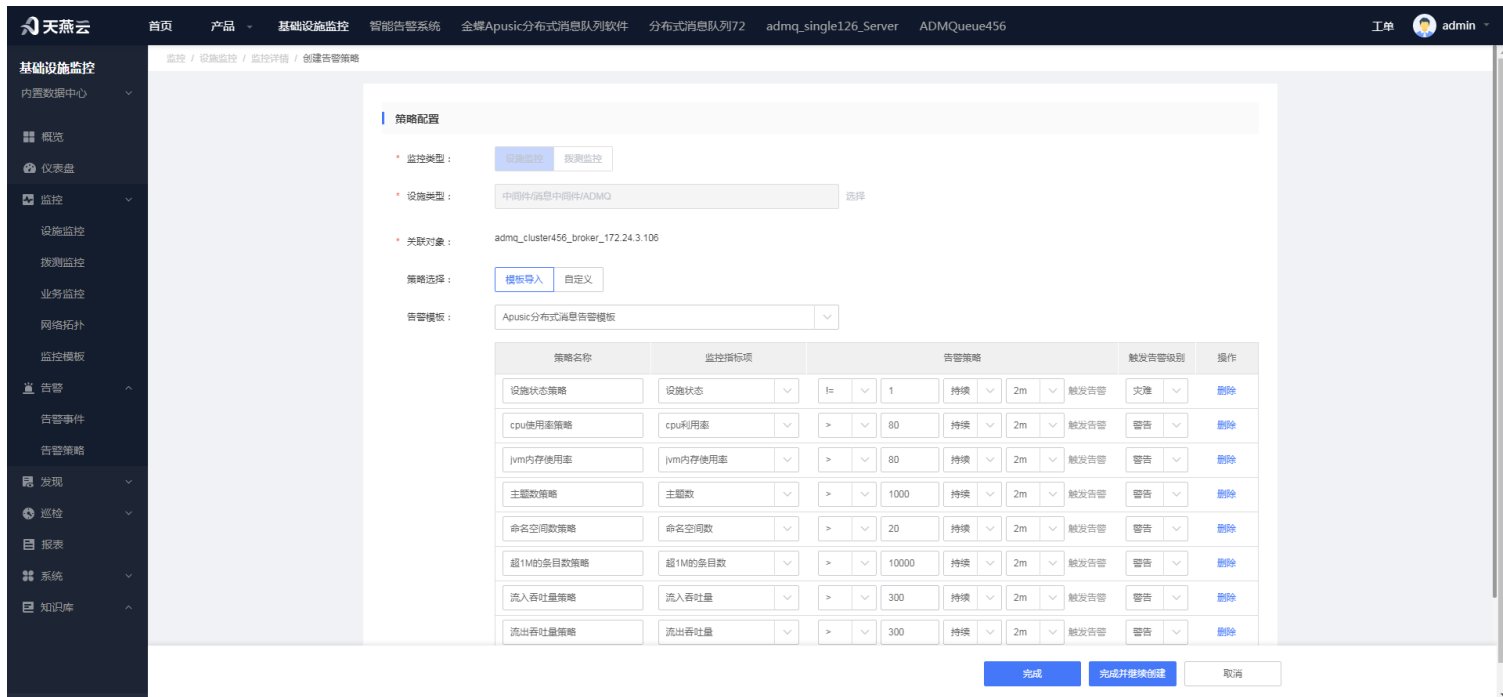




在监控详情页面，切换到“告警策略”页签：



点击“添加”按钮后创建broker告警策略：

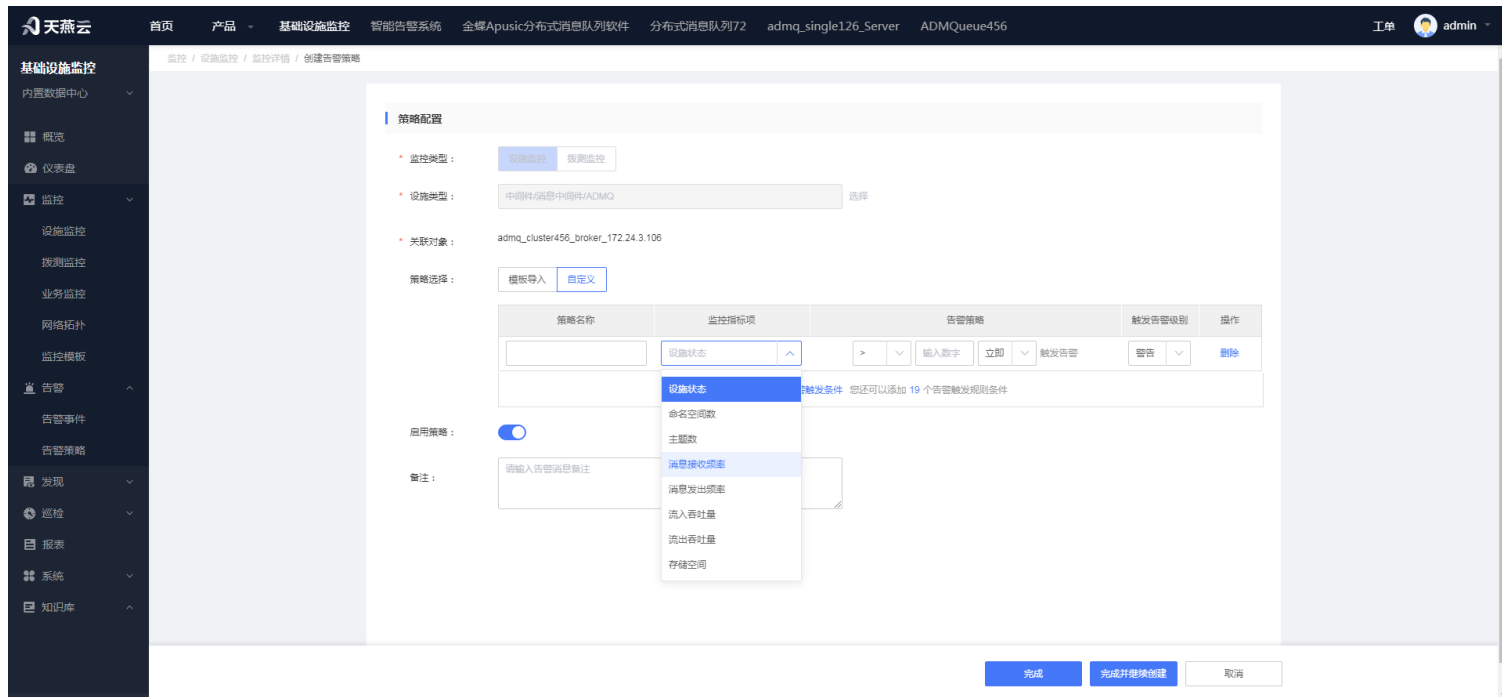


AMP内置的broker告警模版已经包含了8条告警策略:

策略名称	监控项指标	告警触发条件
设施状态策略	设施状态	2m内, 设施状态不等于1时, 触发灾难报警
cpu使用率策略	cpu利用率	2m内, cpu利用率大于80%时, 触发警告报警
jvm内存使用率	jvm内存使用率	2m内, jvm内存使用率大于80%时, 触发警告报警
主题数策略	主题数	2m内, 主题数大于1000个时, 触发警告报警
命名空间数策略	命名空间数	2m内, 命名空间数大于20个时, 触发警告报警
超1M的条目数策略	超1M的条目数	2m内, 超1M的条目数大于10000个时, 触发警告报警

流入吞吐量策略	流入吞吐量	2m内, 流入吞吐量大于300字节/秒时, 触发警告报警
流出吞吐量策略	流出吞吐量	2m内, 流出吞吐量大于300字节/秒时, 触发警告报警

用户也可以根据broker的监控指标项添加自定义告警策略:

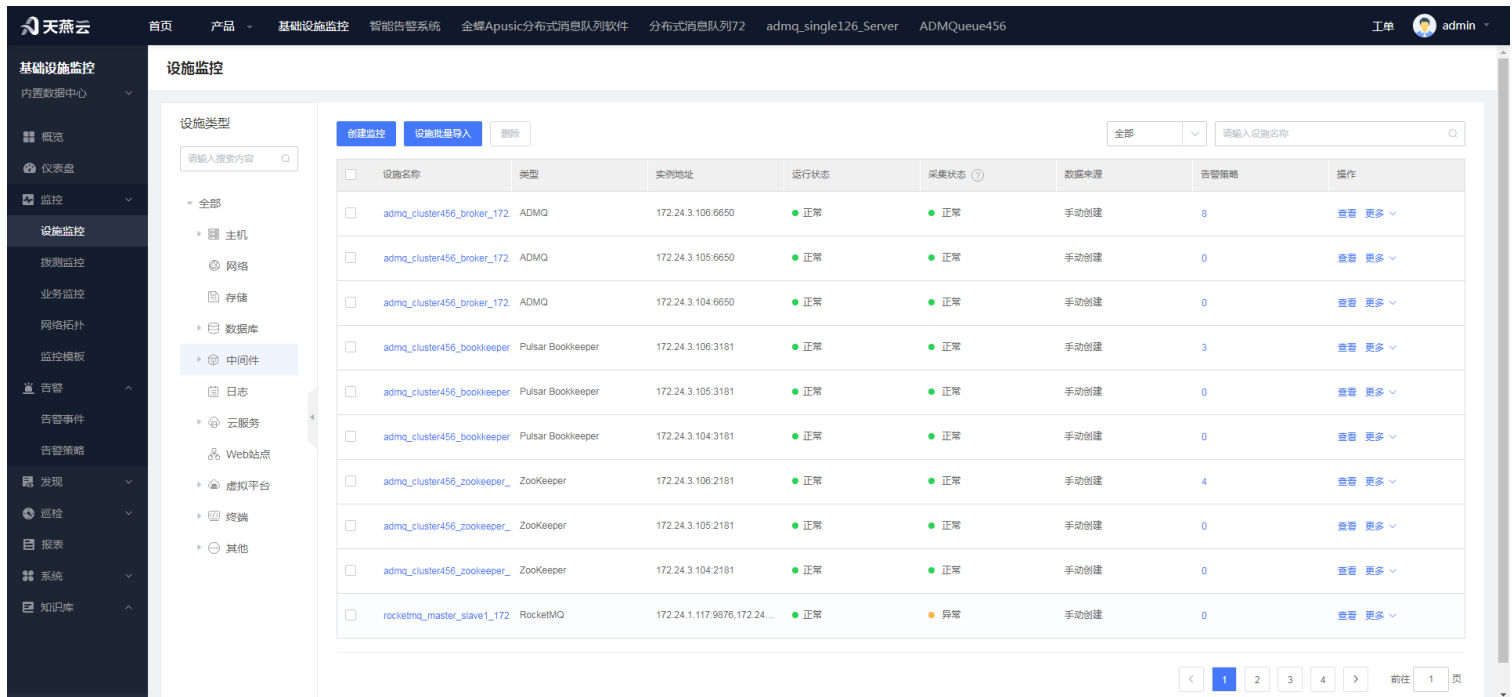


5.11.4 AMP监控指标展示

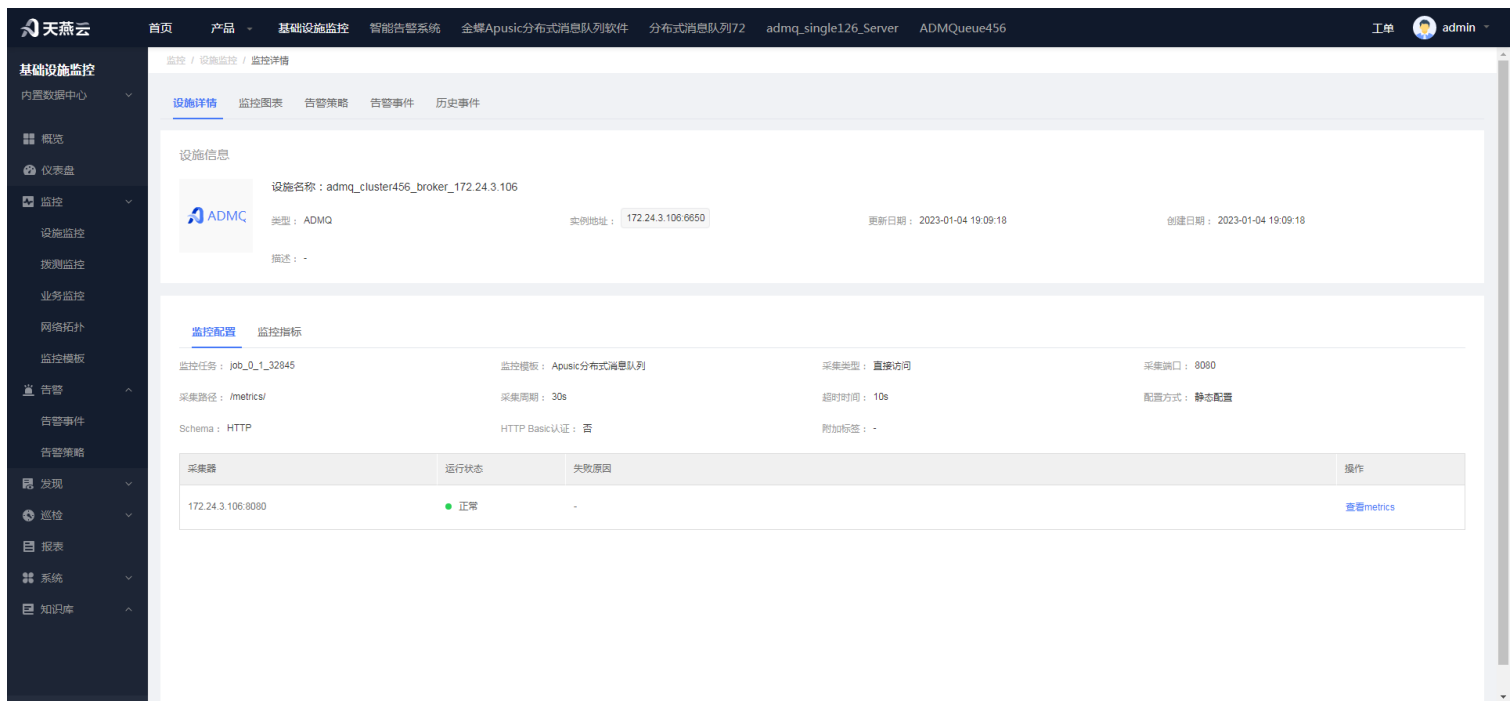
5.11.4.1 按核心组件维度展示

进入设施监控界面, 可以看到设施监控的列表信息, 包括设施名称, 设施类型, 实例地址, 监控模板, 运行状态, 采集器状态, 可以进行信息编辑、监控配置、查看、删除的操作。

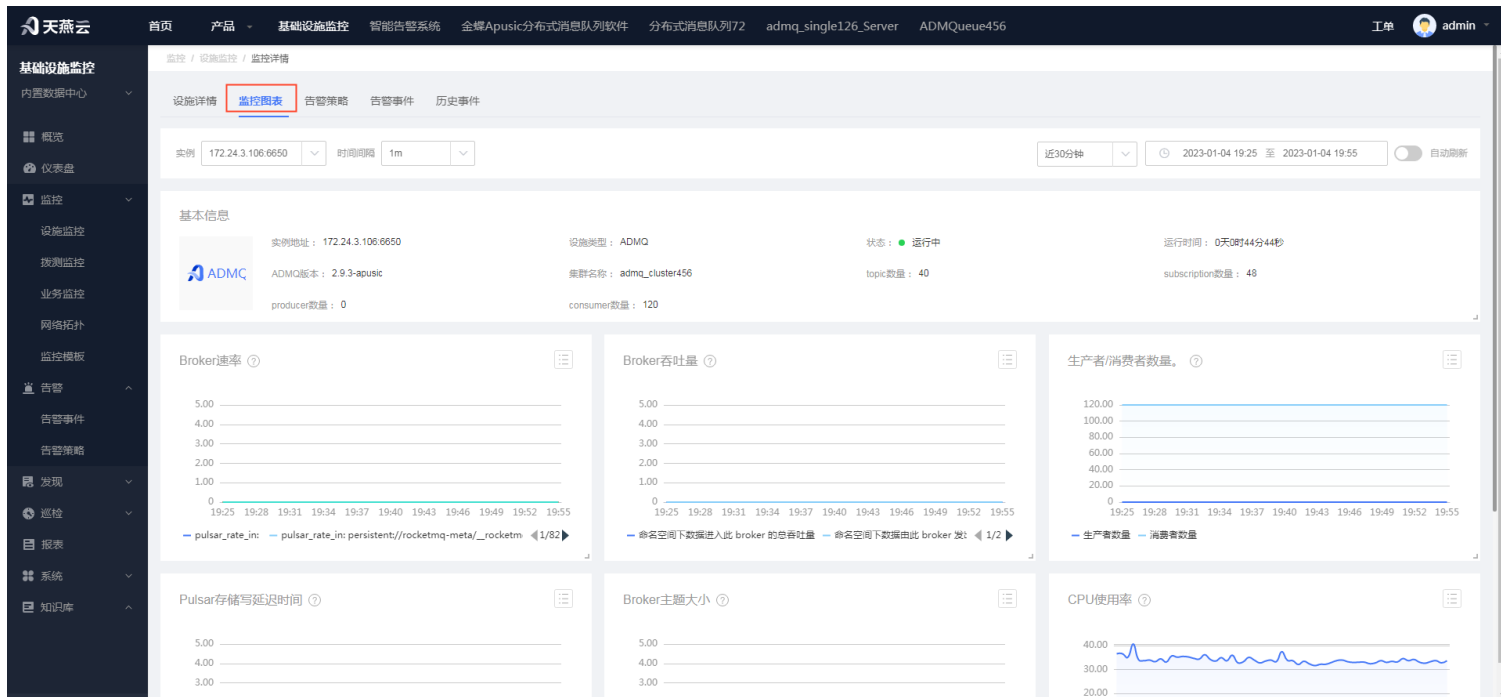
- 设施名称: 创建基础设施输入的设施名称。
- 设施类型: 创建基础设施选择的设施类型。
- 实例地址: 监控对象的ip地址。
- 监控模板: 设施配置任务时选择的监控模板。
- 运行状态: 设施实例运行的状态, 全部运行正常显示正常, 存在运行异常的显示异常, 没有配置监控, 状态显示未知。
- 采集器状态: 采集器运行的状态, 全部运行正常显示正常, 存在运行异常的显示异常, 没有配置监控, 状态显示提示未配置。
- 告警策略: 设施创建的报警策略的数量。
- 信息编辑: 进入修改基础设施界面, 修改设施。
- 监控配置: 进入修改监控配置界面, 修改监控配置的设置。
- 删除: 点击删除标签并确认, 可以删除当前的基础设施和监控任务。



在设施监控的列表界面可以看到ADMQ核心组件的监控项：Linux主机、zookeeper、bookkeeper和broker。点击具体的设施名称或者“查看”进入具体的一个设施，选择“设施详情”页签，可以查看设施信息，监控任务信息，监控项信息。



切换到“监控图表”页签，可以查看该设施的基本信息和监控静态数据和动态图表数据，监控基本信息信息包括ip地址，设施类型，状态等信息。监控图表是对监控指标的可视化展示：

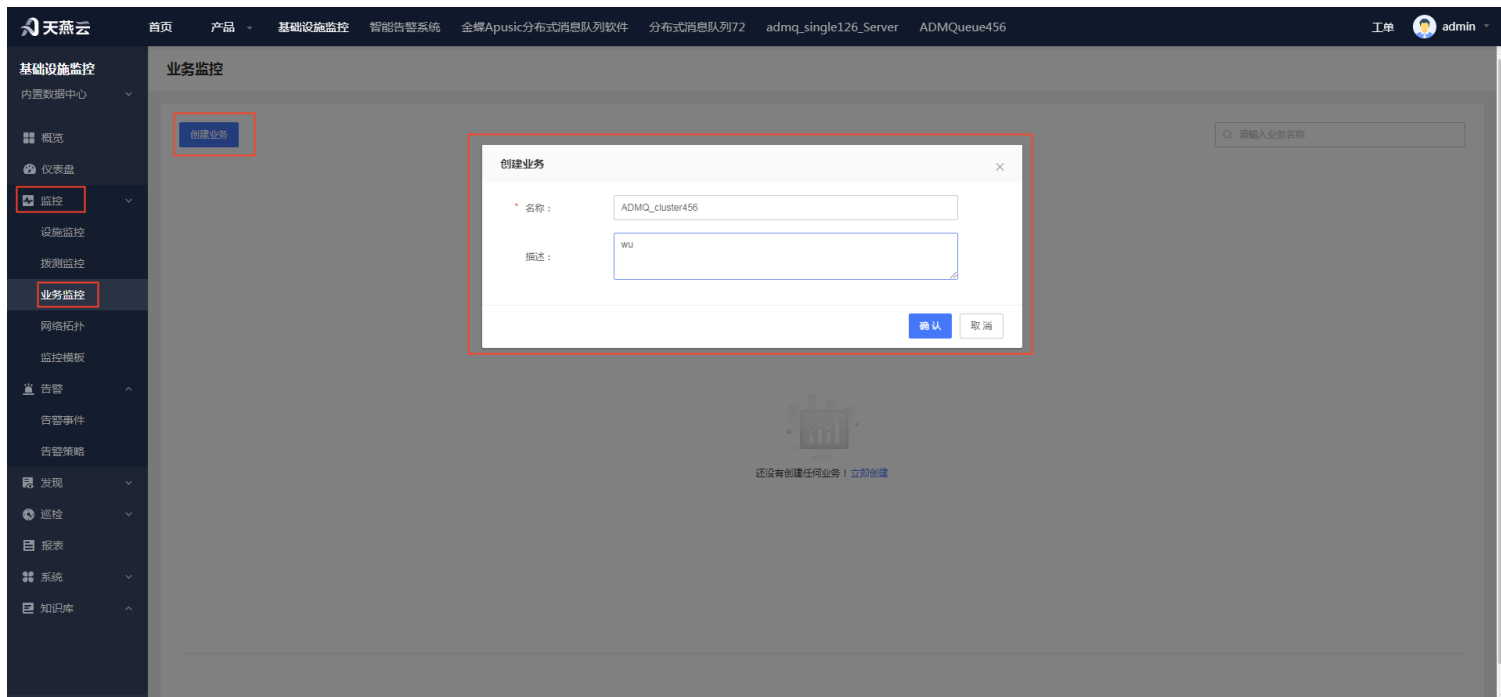


5.11.4.2 按业务视图维度展示

业务监控是为用户提供的可以对用户所使用的平台，平台下的所有应用的管理及监控功能，帮助用户全面了解自己应用平台下的应用关系，应用所关联的设施状态等信息，有助于用户快速了解业务及应用的监控状态，在发生告警信息后，能快速进行处理。

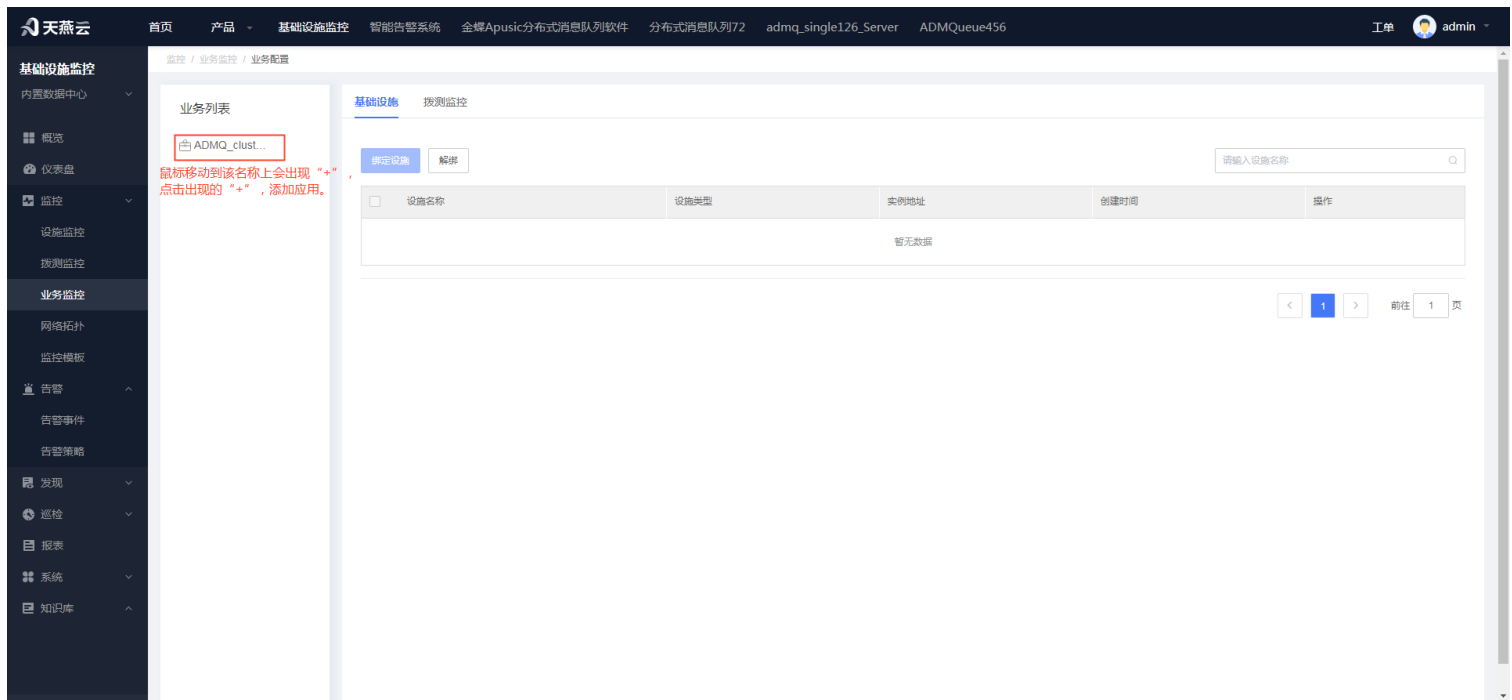
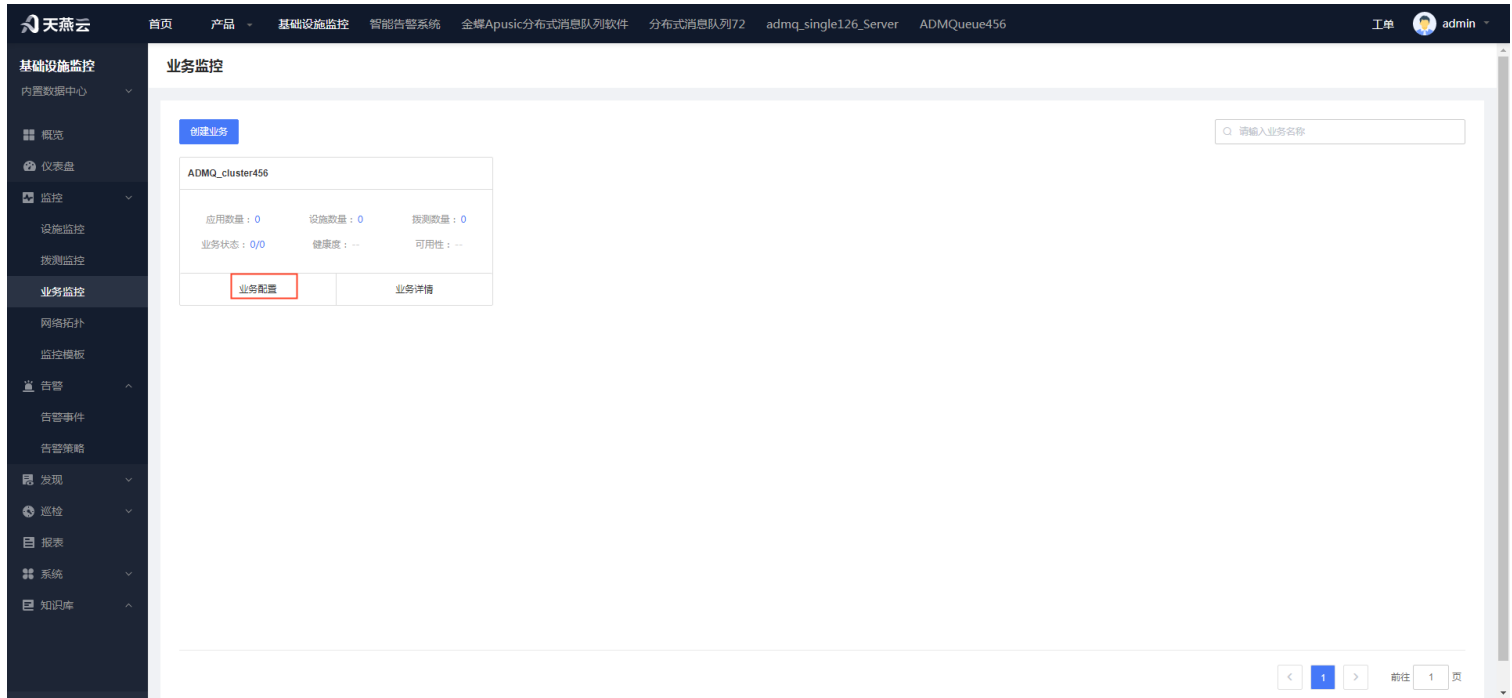
5.11.4.2.1 创建ADMQ监控业务

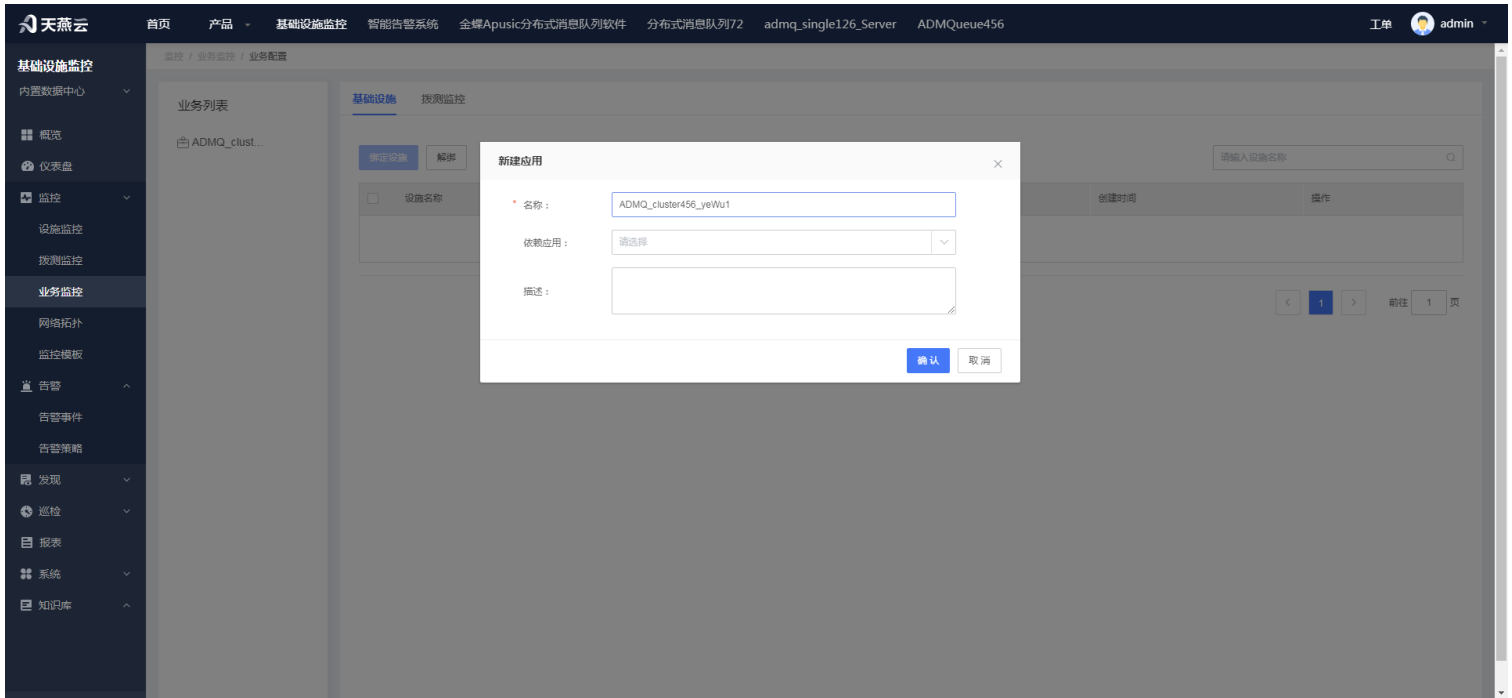
进入到【监控】下的【业务监控】，点击“创建业务”按钮，弹出“创建业务”窗口，输入业务的名称和业务内容，点击完成按钮，完成业务创建操作。



5.11.4.2.2 创建ADMQ应用

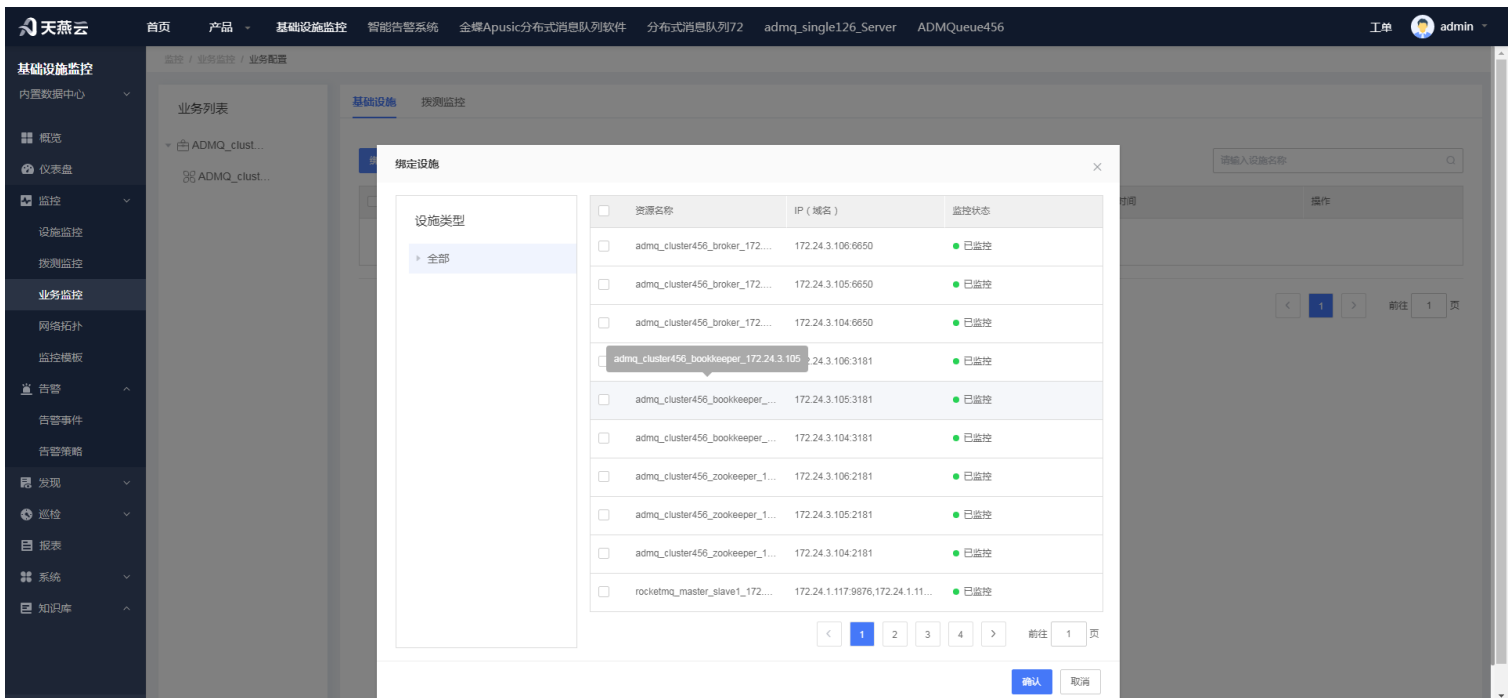
在业务监控的概览页面，找到我们刚刚创建的ADMQ，点击“业务配置”进入业务配置页面。鼠标移动到业务列表下的ADMQ，点击出现的“+”，添加应用。

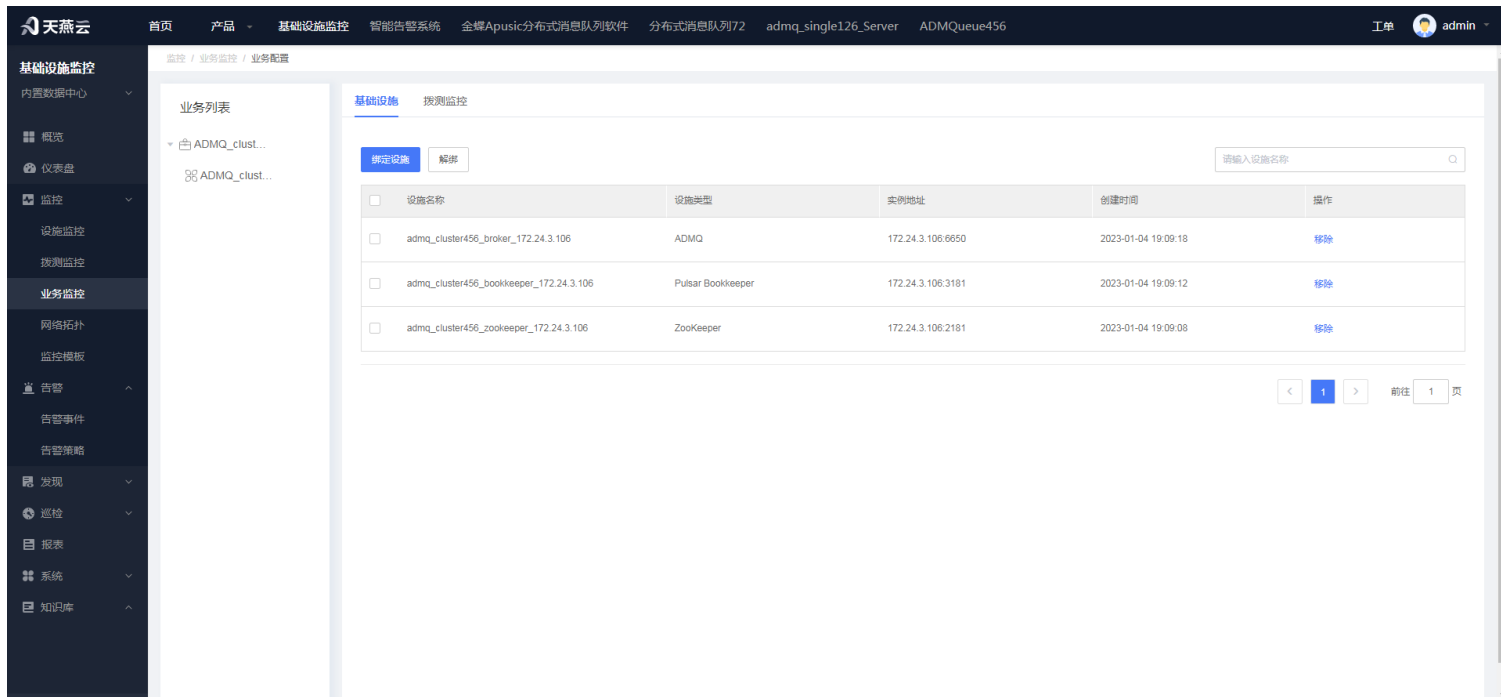




5.11.4.2.3 为ADMQ应用绑定设施

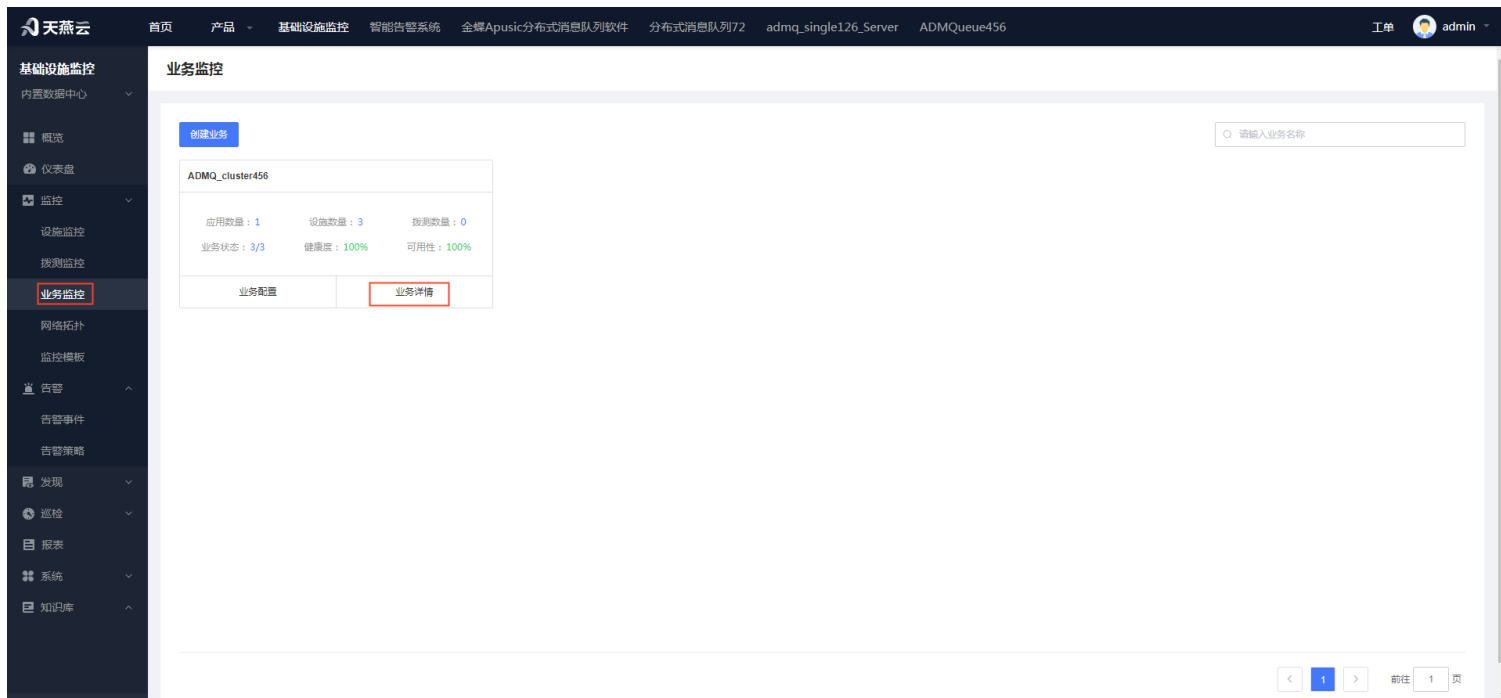
选择刚刚创建的ADMQ应用，点击“绑定设施”按钮，选择ADMQ对应的Linux节点和zookeeper、bookkeeper和broker组件：

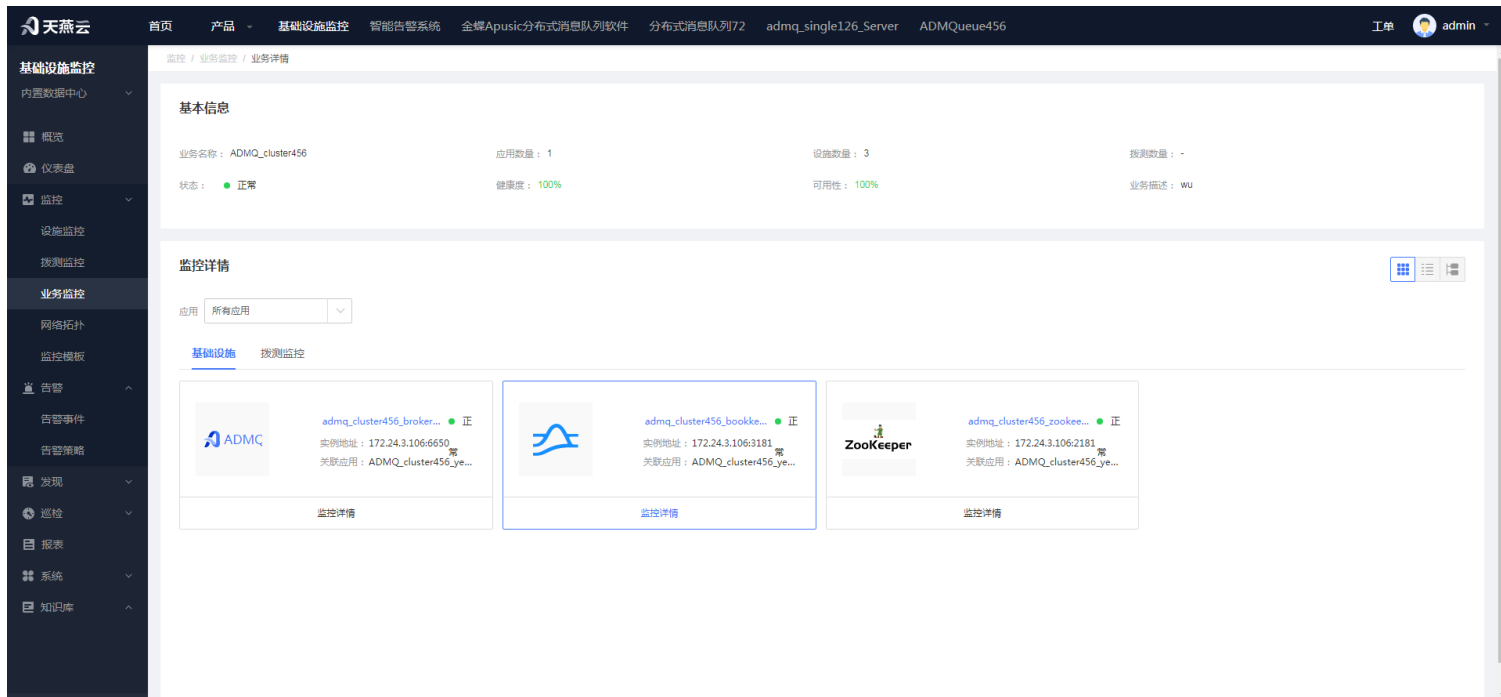




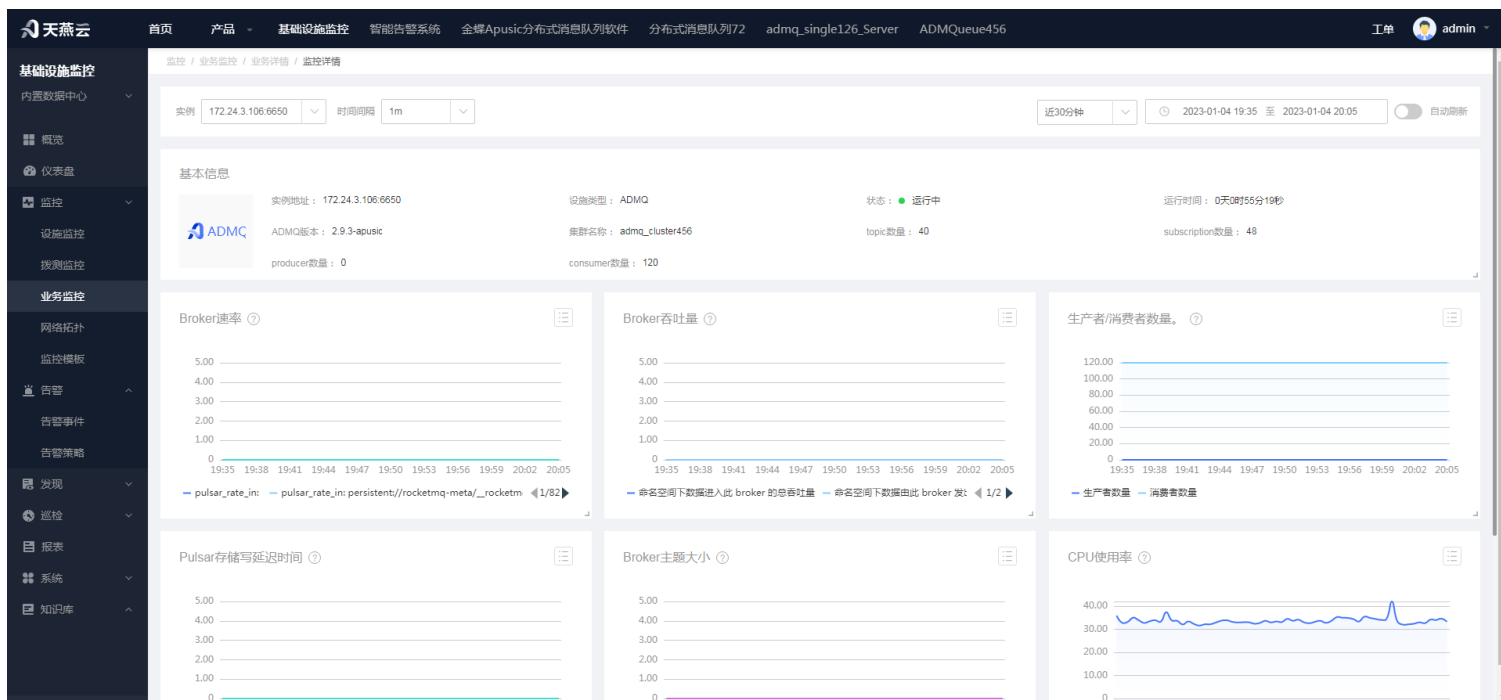
5.11.4.2.4 查看ADMQ监控业务

完成ADMQ应用的设施绑定后，回到业务监控概览页面，点击ADMQ的“业务详情”，进入详情页面，在这里可以查看ADMQ集群的整体健康状况：





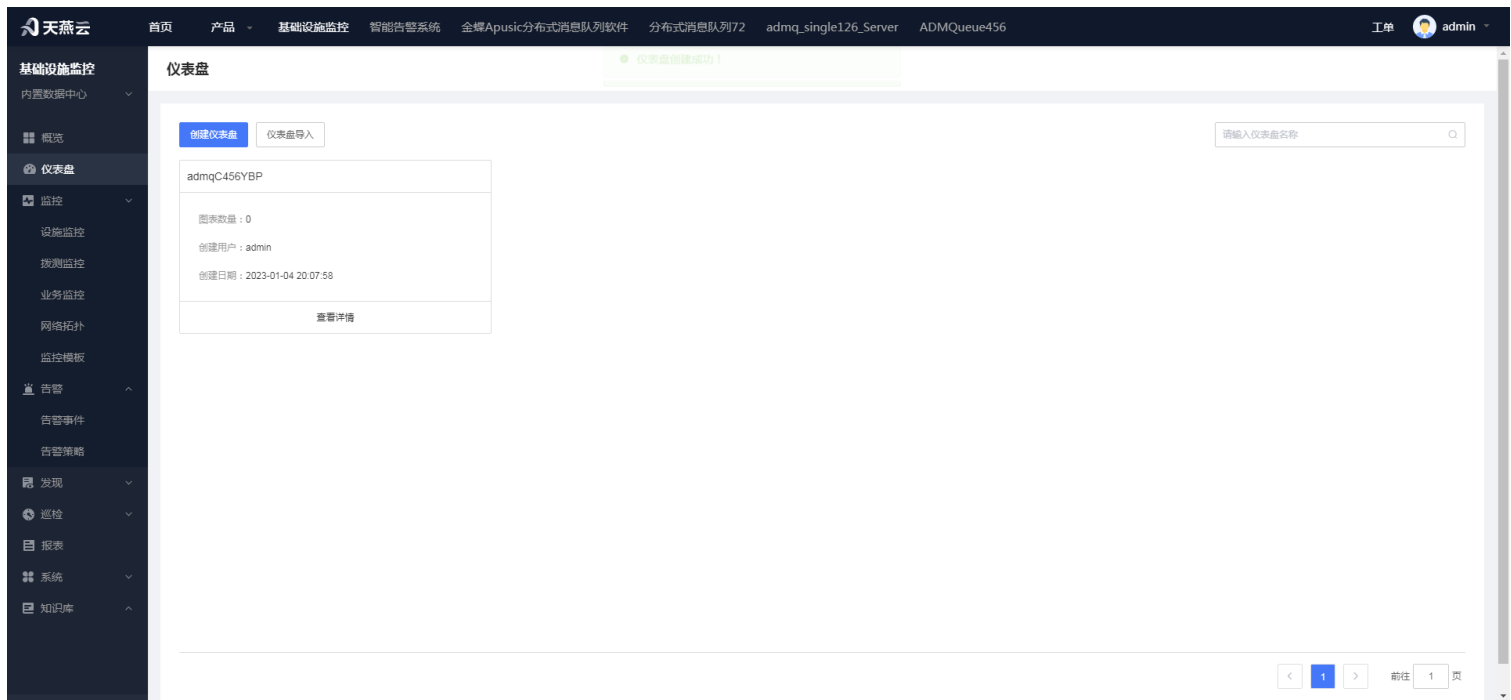
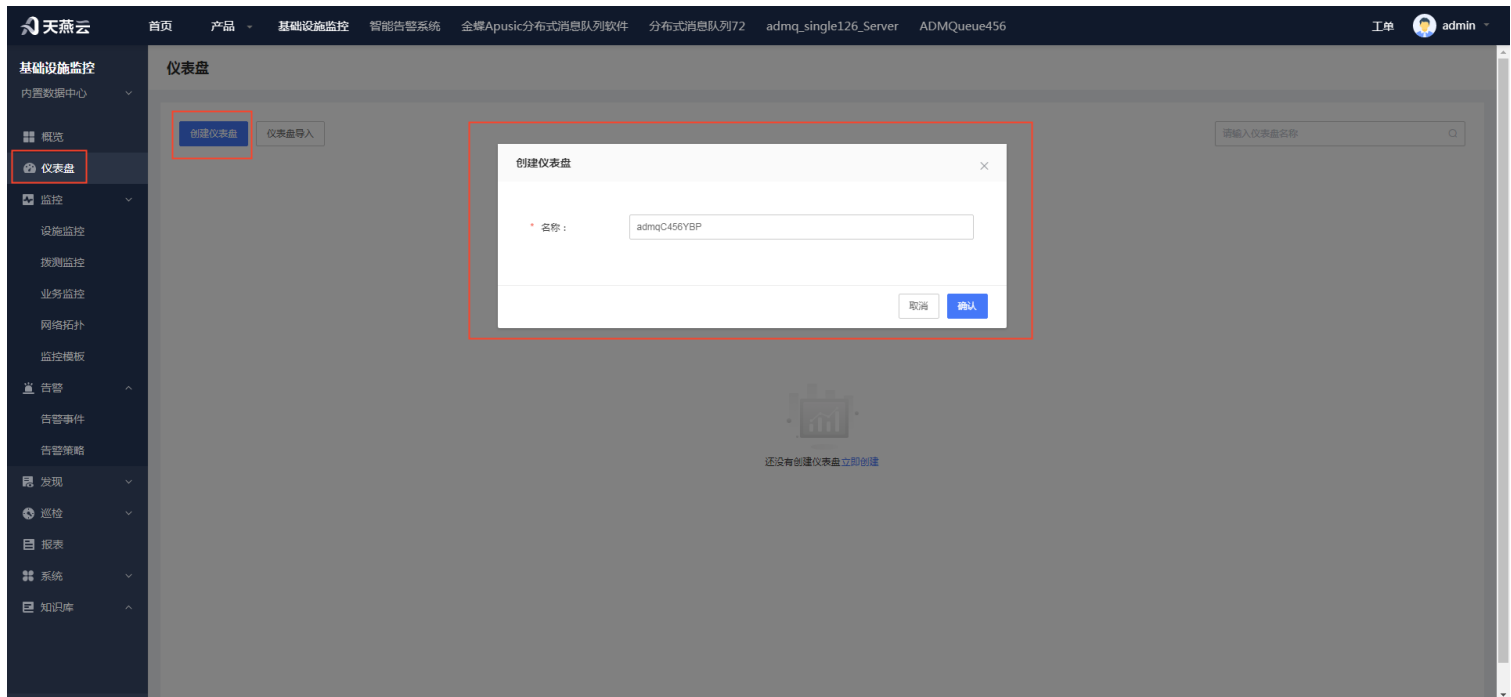
选择其中一个组件，可以查看该组件的“监控详情”：



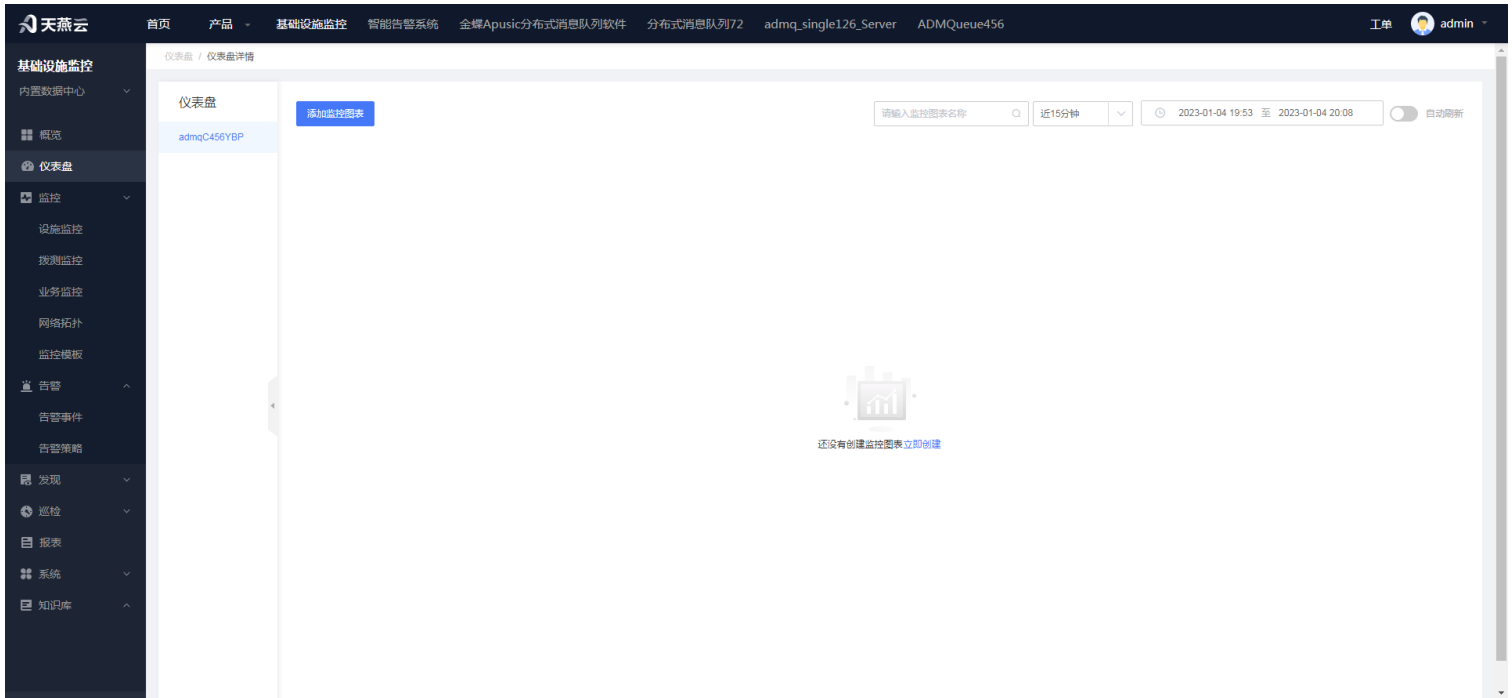
5.11.4.3 自定义仪表盘

AMP提供了自定义仪表盘的功能，可以将多个监控设施的多个指标放在一个仪表盘中，让用户方便查看，所有的关键指标一目了然，可以作为监控大屏使用。

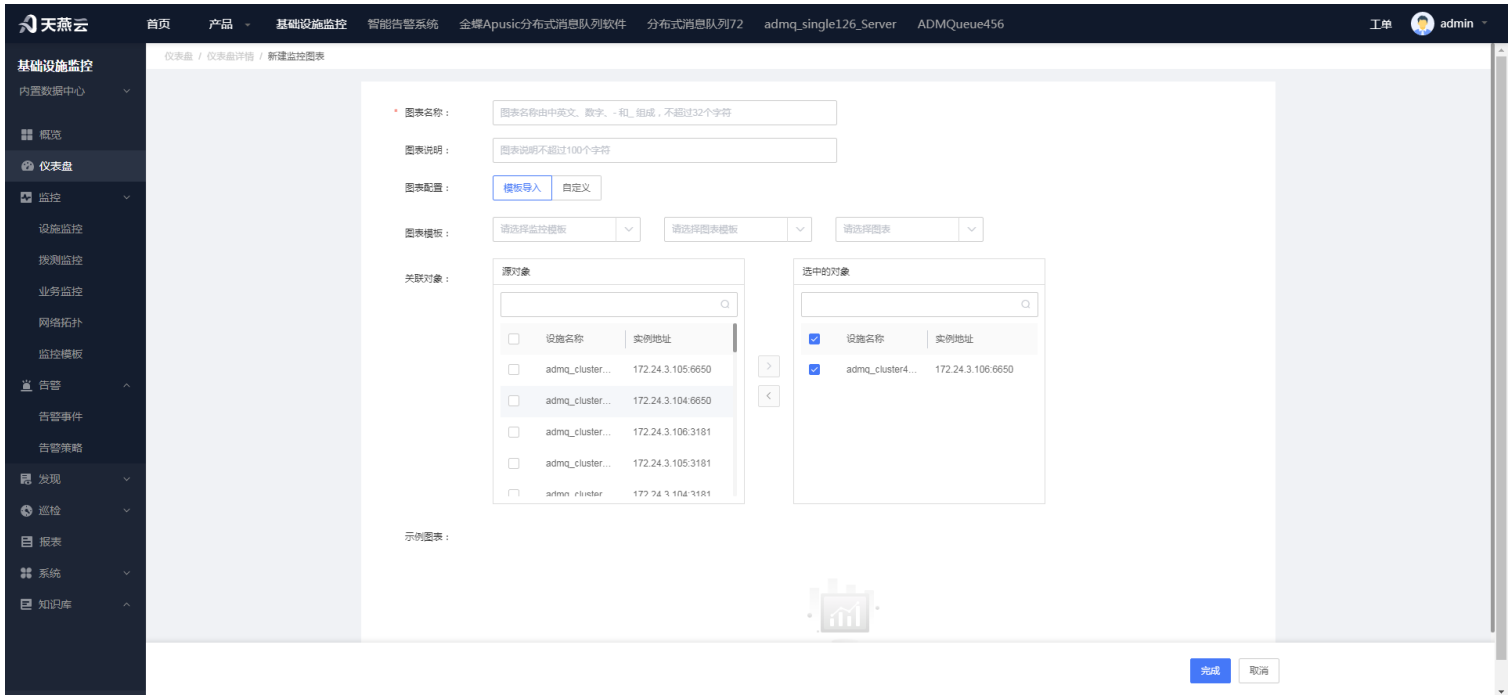
在仪表盘概览页面，点击“创建仪表盘”按钮，输入仪表盘名称后完成创建。

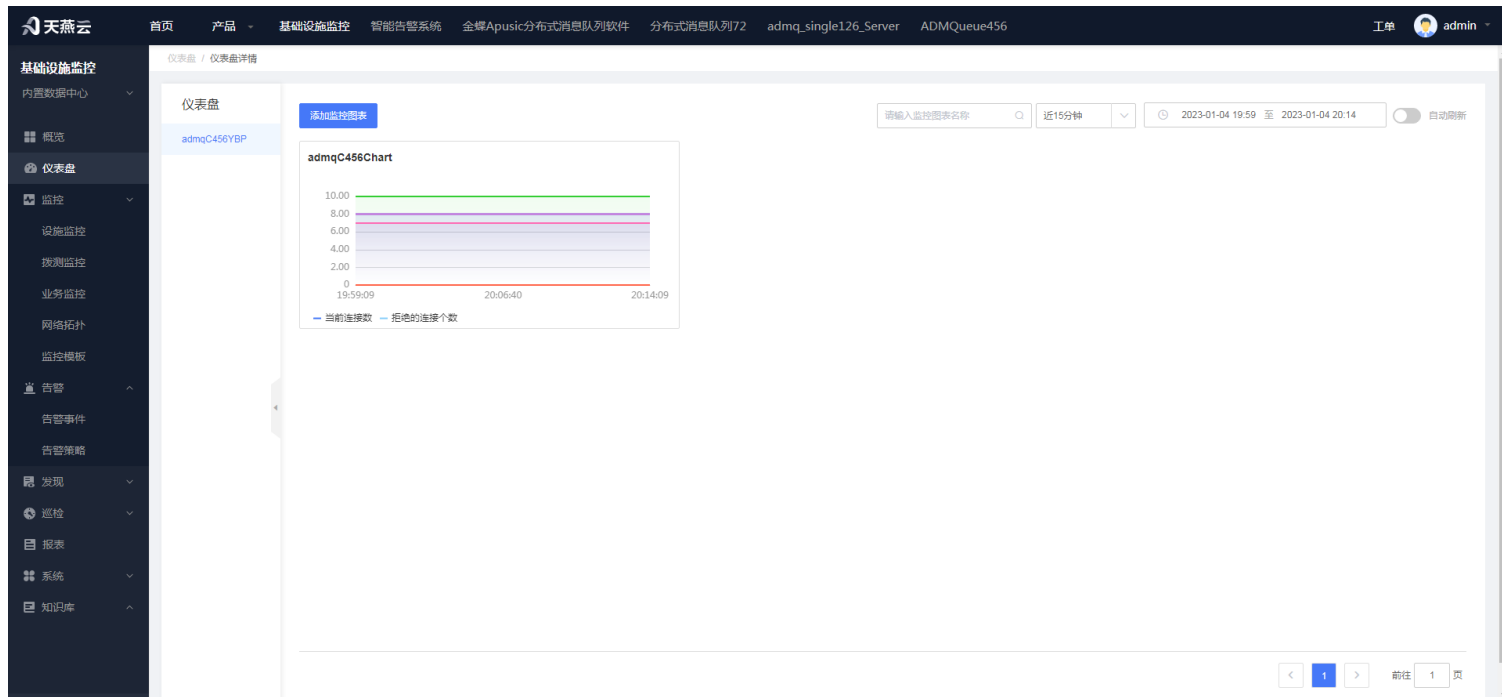


点击仪表盘的“查看详情”按钮，进入到仪表盘详情页面：

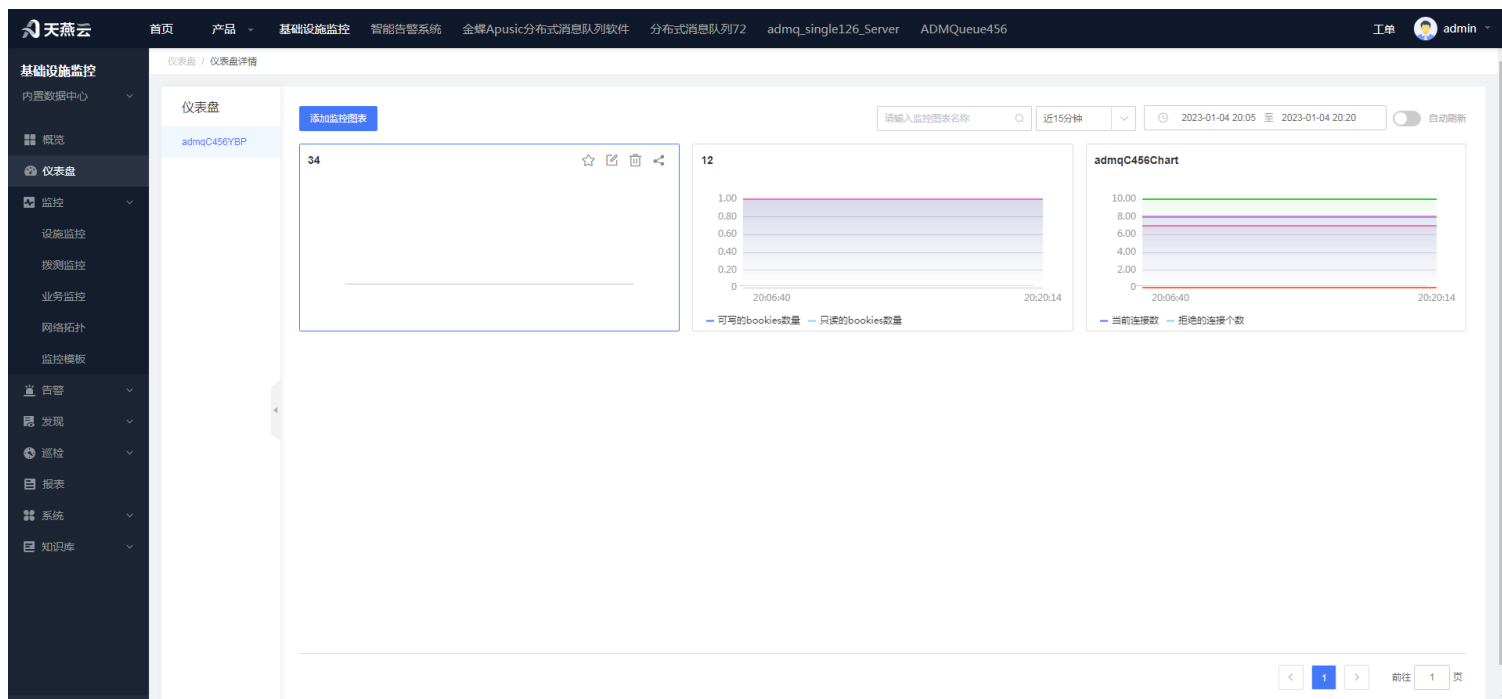


点击“添加监控图表”，设置要添加的图表。





在图表模版中，可以选择ADMQ的组件zookeeper、Bookkeeper和Pulsar，然后选择关注的指标。例如对于zookeeper，我们关注连接数、处理平均延迟和排队请求数，可以分别添加这几个指标的图表。对于核心组件要在仪表盘展示的指标，一般都是告警策略关注的指标。完成添加后，这些关键指标将在一个仪表盘中统一展示。



6 性能测试

测试环境

服务器	硬件配置	软件配置	备注
172.100.140.162	CPU: Intel Xeon 2.93G 8核, 内存: 32G, 硬盘: 500G	ADMQ v2.4 manager, JDK11.0.16.1	ADMQ 管理控制台
172.100.140.186	CPU: Intel Xeon 2.93G 8核, 内存: 32G, 硬盘: 500G	ADMQ v2.4, JDK11.0.16.1	ADMQ 单机部署服务器
172.100.140.183	CPU: Intel Xeon 2.93G 8核, 内存: 32G, 硬盘: 500G	ADMQ v2.4, JDK11.0.16.1	ADMQ 集群部署服务器
172.100.140.184	CPU: Intel Xeon 2.93G 8核, 内存: 32G, 硬盘: 500G	ADMQ v2.4, JDK11.0.16.1	ADMQ 集群部署服务器
172.100.140.185	CPU: Intel Xeon 2.93G 8核, 内存: 32G, 硬盘: 500G	ADMQ v2.4, JDK11.0.16.1	ADMQ 集群部署服务器

性能测试指标

在上述测试环境中, ADMQ v2.0.5 性能指标如下:

- **单机部署** 形态下递增Topic数量的吞吐量测试, 消费吞吐量达到 350MB/s 左右, TPS为 300000 ;
- **集群部署** 形态下递增Topic数量的吞吐量测试, 集群消费吞吐量稳定在 280MB/s 左右, TPS为 260000 ;

7 信创适配

金蝶Apusic分布式消息队列软件 V2.0.5 支持主流信创国产CPU、操作系统、数据库，出厂适配兼容良好，支持国产环境的安装部署，稳定可靠运行，性能良好。

- 国产CPU

表1- 支持的CPU

厂商	CPU架构	适配支持
鲲鹏	arm64(AArch64)	√
飞腾	arm64(AArch64)	√
海光	amd64	√
兆芯	x86_64	√
龙芯	loongarch64	√
申威	SW-64	√

- 操作系统

表2- 支持的操作系统

备注：以下操作系统均搭配上上述国产CPU架构组合

厂商/社区	操作系统	适配支持
麒麟	银河麒麟高级服务器操作系统 V10	√
麒麟	中标麒麟高级服务器操作系统 V7.0	√
统信	统信服务器操作系统 V20	√
openEuler	openEuler V22.03/V23/03	√
普华	普华服务器操作系统 V4.0	√
中科方德	中科方德服务器操作系统 V4.0	√

- 数据库

表3- 支持的数据库

厂商	数据库	适配支持
达梦	Dameng v7/V8	√
人大金仓	kingbaseES v7/v8	√
神舟通用	ShenTong v7.0	√
海量	Vastbase G100	√

8 常见问题

8.1 概念相关

8.1.1 是否支持国产数据库

支持。ADMQ支持mysql、sqlite、herddb、postgresql、达梦、人大金仓等数据库。

8.1.2 是否支持通过其他消息中间件的客户端发送消息

目前支持通过Kafka协议、MQTT协议、JMS协议和AMQP协议的客户端发送消息，对应的消息中间件有RabbitMQ、Kafka。

对于使用其他消息中间件的客户端，ADMQ并不支持该客户端的全部功能，只包含收发消息等基本功能。

8.1.3 能够支持其他消息中间件的版本

Kafka V1.0及以上、RabbitMQ V1.0及以上、RocketMQ V4.X, MQTT V5.0

8.1.4 是否能够同时支持多种消息协议

ADMQ可以同时开启KOP、AOP、AMQP插件，以支撑多种消息协议。但长久运行后，系统的存储和负载会提升，从而造成相互间的影响。建议生产环境中不要同时开启所有插件。

8.1.5 是否有权限管理功能

权限包含两类

- 管控台权限：管理员可以管理集群和资源（包括：租户、命名空间、主题），并且可以分配资源权限给用户。
- 消息收发权限：生产者和消费者连接到ADMQ时需要携带token，token会控制消费者是否有主题的消息发送和消息读取权限。

8.1.6 消费者的四种订阅模式区别

独占模式 (Exclusive)

只能有一个消费者读取消息，后续的消费者不能获取到消息，而且会报错（如成功开启一个独占接收后，再开一个客户端时会提示错误，直接断开，无法连接）。

灾备模式 (Failover)

有且仅有一个消费者在消费消息。和独有模式的区别是，这种模式下消费者会保持和broker之间的连接，一旦正在消费的消费者和broker的连接挂了，则可以任意选择一个正常的消费者消费消息（如成功开启一个灾备模式接收后，可以再开一个客户端处于连接中，但是无法接收消息，只有第一个客户端挂了后，这个客户端才能正常接收消息）。

独占和灾备模式的区别是，独占情况下只能有一个客户端能连到admq，第二个会直接退出。灾备模式是可以有多个客户端连接到admq，但是只能有一个发送消息，最先连接上的那个发送。

共享模式 (Shared)

每个连接的消费者获取一部分数据。如果其中一个消费者挂了，则之前发送给这个消费者的且没有得到确认的消息会重新发送给其他消费者。该模式就是平均发送给每一个消费者。但是有一种可能是不平均的，就是有的消费者处理的慢，那它收到的消息就比其他的少。

主键负载均衡 (key_shared)

按照key做hash，每一个key的消息只能被唯一的一个消费者接收。

这种模式下，不能保证每个消费者都能收到消息。

8.1.7 消息持久化和非持久化区别

在topic中指定协议，persistent表示持久化，non-persistent表示非持久化，非持久化的消息不会被存储，发完就没有了，不能读取历史消息，持久化消息则可以存储起来，对于那些已经被消费者读取并确认的消息，能够再次读取，用于系统问题分析等场景。

8.1.8 死信队列

正常主题的消息发送到消费者，消费者处理失败的消息会自动转到死信队列里，然后消费者就可以从死信队列里收到那些处理失败的消息。

8.1.9 批量消息发送

支持批量消息和消息压缩，当消息大小（batch_bytes）、数量（batch_messages）、时间（batch_delay），满足任意一个后才会发送消息。

8.1.10 消息加密

消息在生产者和消费者之间的传输过程中需要加密，采用AES加密算法。生产者和消费者保留私钥，ADMQ不解析消息内容，只提供转发功能。

但是对于同一个主题不能加密和不加密混合使用，如：对主题TopicTest先发过加密的消息。后面再发送不加密的消息，接收时就会提示错误。

8.1.11 消息保留策略

支持设置未确认的消息多长时间后被标记已确认，已确认的消息多长时间后被删除。在对应的命名空间中设置清理策略，TTL和保留时间和大小。

8.1.12 支持发送哪些类型的消息

目前支持byte数组、string、json、avro格式的消息。

8.1.13 租户、命名空间、主题、用户的作用和说明

- 租户、命名空间、主题 ADMQ是发布订阅模型，生产者发送的每一条消息都需要指定一个主题，消费者订阅的时候也需要告诉admq接收哪一个或者哪几个主题的消息。租户、命名空间和主题都是为了隔离数据的，一个租户下面可以有多个命名空间，一个命名空间下可以有多个主题。
- 用户 用户是权限判断用的。生产者接收消息和消费者订阅消息都需要携带用户信息（每个用户对应一个唯一的token），ADMQ会根据用户信息判断生产者是不是能往一个主题发送消息，消费者能不能订阅一个主题的消息。

8.1.14 消息查询

ADMQ会记录消息生产、存储和消费状态，方便用户排查消息发送和接收情况。

默认不会开启消息轨迹功能，需要在“资源管理”-“命名空间”-“...”中手动开启轨迹，不开启轨迹只能查询消息生产情况和消息内容，不能查到消息消费情况。

以下情况查不到信息：

- 客户端使用系统提供的AES加密发送消息。
- 客户端批量发送，同时开启了逐条确认功能。
- 客户端采用累积确认。

8.2 部署相关

ADMQ支持单节点和集群两种部署方式，单节点部署时只提供部分简单功能，用于验证消息的接收和发送是否正常，集群部署提供完整的功能。

集群部署时至少需要3台服务器，需要分别部署计算组件、协调器和存储组件。

8.2.1 单机启动失败

单机程序启动后不能直接kill进程，需要在管控台-集群管理-节点中点击“停止”，或者在部署服务器上执行bin/admq-daemon stop standalone。

8.2.2 节点启动失败

首先查看管控台是否有错误日志，确认添加服务器时使用的用户是否有部署目录的读写和执行权限。

然后登录启动失败的节点服务器，手动启动节点，查看是否有错误日志，启动命令如下：

- 计算节点：bin/admq broker
- 存储节点：bin/admq storage
- 协调器：bin/admq zookeeper

如出现“VM option 'UseG1GC' is experimental and must be enabled via --XX:+UnlockExperimentalVMOptions”，则修改config/broker_env.sh和config/storage_env.sh，在UseG1GC前添加参数：--XX:+UnlockExperimentalVMOptions。

8.2.3 没有java执行权限

解压安装包，修改bin/admq文件，添加命令：

export JAVA_HOME=有权限的jdk目录

重新压缩安装包，上传到管控台上进行部署。

8.2.4 管控台启动节点报JAVA_HOME not set

```
Method)
java:672)
hConnect(SocketChannelImpl.java:946)
ransport(ClientCnxnSocketNIO.java:344)
un(ClientCnxn.java:1290)
.i.ProgramServiceImpl [ProgramServiceImpl.java : 585] 执行命令完成，命令：cd /apusic/admq/v2.4/program/pQfrlpEi/storage/rIpcIp
orage/rIpcIpUA/admq-V2.4/bin/admq daemon start storage, 结果：[doing start storage :., starting storage, logging to /apusic
4/logs, which: no java in (/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin), Error: JAVA_HOME not set, and no java
r/sbin:/usr/bin:/root/bin.]
min [MQAdmin.java : 87] [http://172.22.241.30:8080] admq admin 无法连接
min [MQAdmin.java : 87] [http://172.22.241.29:8080] admq admin 无法连接
```

解决方案：在/root/.bashrc文件里增加export JAVA_HOME=xxxx即可

```
Last login: Thu Dec 21 16:08:12 2023 from 172.21.33.101
[root@openstack-140-160 ~]# cat /root/.bashrc
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
rootpasswd=FyWU+Q0ghldh
export rootpasswd
export JAVA_HOME=/opt/tool/jdk-17.0.8
[root@openstack-140-160 ~]#
```

8.2.5 协调器连接超时或失败

```

... 8 more
20:12:49.292 [main] ERROR org.apache.pulsar.PulsarBrokerStarter - Failed to start pulsar service.
org.apache.pulsar.broker.PulsarServerException: org.apache.pulsar.metadata.api.MetadataStoreException: org.apache.zookeeper KeeperException$ConnectionLossException: Keeper
errorCode = ConnectionLoss
    at org.apache.pulsar.broker.PulsarService.start(PulsarService.java:876) ~[org.apache.pulsar-pulsar-broker-2.10.3-apusic.jar:2.10.3-apusic]
    at org.apache.pulsar.PulsarBrokerStarter$BrokerStarter.start(PulsarBrokerStarter.java:275) ~[org.apache.pulsar-pulsar-broker-2.10.3-apusic.jar:2.10.3-apusic]
    at org.apache.pulsar.PulsarBrokerStarter.run(PulsarBrokerStarter.java:356) ~[org.apache.pulsar-pulsar-broker-2.10.3-apusic.jar:2.10.3-apusic]
    at com.apusic.admq.MQBroker.main(MQBroker.java:34) ~[org.apache.pulsar-pulsar-broker-2.10.3-apusic.jar:2.10.3-apusic]
Caused by: org.apache.pulsar.metadata.api.MetadataStoreException: org.apache.zookeeper KeeperException$ConnectionLossException: KeeperErrorCode = ConnectionLoss
    at org.apache.pulsar.metadata.impl.ZKMetadataStore.<init>(ZKMetadataStore.java:112) ~[org.apache.pulsar-pulsar-metadata-2.10.3-apusic.jar:2.10.3-apusic]
    at org.apache.pulsar.metadata.impl.MetadataStoreFactoryImpl.newInstance(MetadataStoreFactoryImpl.java:59) ~[org.apache.pulsar-pulsar-metadata-2.10.3-apusic.jar:2.10.3-apusic]
    at org.apache.pulsar.metadata.impl.MetadataStoreFactoryImpl.createExtended(MetadataStoreFactoryImpl.java:36) ~[org.apache.pulsar-pulsar-metadata-2.10.3-apusic.jar:2.10.3-apusic]
    at org.apache.pulsar.metadata.api.extended.MetadataStoreExtended.create(MetadataStoreExtended.java:40) ~[org.apache.pulsar-pulsar-metadata-2.10.3-apusic.jar:2.10.3-apusic]
    at org.apache.pulsar.broker.PulsarService.createLocalMetadataStore(PulsarService.java:1035) ~[org.apache.pulsar-pulsar-broker-2.10.3-apusic.jar:2.10.3-apusic]
    at org.apache.pulsar.broker.PulsarService.start(PulsarService.java:679) ~[org.apache.pulsar-pulsar-broker-2.10.3-apusic.jar:2.10.3-apusic]
    ... 3 more
Caused by: org.apache.zookeeper KeeperException$ConnectionLossException: KeeperErrorCode = ConnectionLoss
    at org.apache.zookeeper KeeperException.create(KeeperException.java:102) ~[org.apache.zookeeper-zookeeper-3.6.3.jar:3.6.3]
    at org.apache.bookkeeper.zookeeper.ZooKeeperWatcherBase.waitForConnection(ZooKeeperWatcherBase.java:159) ~[org.apache.bookkeeper-bookkeeper-server-4.14.7-apusic.jar:4.14.7-apusic]
    at org.apache.pulsar.metadata.impl.PulsarZooKeeperClient$Builder.build(PulsarZooKeeperClient.java:259) ~[org.apache.pulsar-pulsar-metadata-2.10.3-apusic.jar:2.10.3-apusic]
    at org.apache.pulsar.metadata.impl.ZKMetadataStore.<init>(ZKMetadataStore.java:104) ~[org.apache.pulsar-pulsar-metadata-2.10.3-apusic.jar:2.10.3-apusic]
    at org.apache.pulsar.metadata.impl.MetadataStoreFactoryImpl.newInstance(MetadataStoreFactoryImpl.java:59) ~[org.apache.pulsar-pulsar-metadata-2.10.3-apusic.jar:2.10.3-apusic]
    at org.apache.pulsar.metadata.impl.MetadataStoreFactoryImpl.createExtended(MetadataStoreFactoryImpl.java:36) ~[org.apache.pulsar-pulsar-metadata-2.10.3-apusic.jar:2.10.3-apusic]
    at org.apache.pulsar.metadata.api.extended.MetadataStoreExtended.create(MetadataStoreExtended.java:40) ~[org.apache.pulsar-pulsar-metadata-2.10.3-apusic.jar:2.10.3-apusic]
    at org.apache.pulsar.broker.PulsarService.createLocalMetadataStore(PulsarService.java:1035) ~[org.apache.pulsar-pulsar-broker-2.10.3-apusic.jar:2.10.3-apusic]
    at org.apache.pulsar.broker.PulsarService.start(PulsarService.java:679) ~[org.apache.pulsar-pulsar-broker-2.10.3-apusic.jar:2.10.3-apusic]
    ... 3 more

```

解决方案:

遇到这种阻塞的办法, 可以尝试把 hostname 配置成服务器IP, 然后在 hosts 文件里配置上所有的节点 IP。

ifconfig 查看服务器 IP 地址

```

linux-4-231
[root@linux-4-231 ~]# ifconfig
ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.24.4.231 netmask 255.255.255.0 broadcast 172.24.4.255
    inet6 fe80::250:56ff:fe99:6ad prefixlen 64 scopeid 0x20<link>
    ether 00:50:56:99:06:ad txqueuelen 1000 (Ethernet)
    RX packets 396967026 bytes 685402662112 (638.3 GiB)
    RX errors 0 dropped 26338 overruns 0 frame 0
    TX packets 314672969 bytes 813067961643 (757.2 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 158713760 bytes 1016184742373 (946.3 GiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 158713760 bytes 1016184742373 (946.3 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

执行 hostname 查看配置的地址。

```

Last login: Tue Nov 28 09:07:02 2023 from 172.21.32.7
[root@openstack-140-171 ~]# hostname
openstack-140-171.novalocal

```

如果是域名, 则执行 hostname ip 手动改成服务器 IP

```

[root@openstack-140-171 ~]# hostname 172.20.140.171

```

vi /etc/hosts

8.3.2 查看历史消息

支持指定消息起始位置查看消息，history 根据这个消息id 读取后续的消息(start_index, 这个需要填消息ID, 不包含当前id)。

8.3.3 license文件

license文件夹里无合法的license的文件，管控台服务无法启动成功。如果这个license在使用过程中过期了，如果这时候连一个合法的license都没有，管控台会通知broker和bookkeeper关掉，管控台服务也会关掉。

8.3.4 权限

租户和命名空间需要在管控台创建好，否则后台发送时会失败，主题和订阅名称可在后台自动创建成功。后台发送和接收消息时，需要带上认证信息token。token为管控系统中的用户的令牌。

8.3.5 消息确认

消费者那边在接收到消息后是会发送确认给admq的，如果不发送确认，下次启动还会收到消息。

8.3.6 怎么设置不同的租户

topic上可以设置。比如persistent://tenant_name/namespace_name/topic_name，前面的persistent表示需要保存消息到文件里，non-persistent表示不保存。发送消息只输入主题默认就放到默认租户和默认命名空间了（public/default）。

8.3.7 客户端启动后没有收到消息

如果客户端是第一次启动，则默认情况下是不能收到该客户端启动之前生产者发送的消息的。

可以在程度里设置从最早的消息开始读取。

8.3.8 同一个订阅名称是否能设置不同的订阅模式

不能。每个 <主题, 订阅> 只能有一种模式。

8.3.9 使用事务时程序报错

默认情况下，ADMQ集群是没有开启事务支持的，需要手动修改配置文件开启事务并重启ADMQ的broker节点。

8.3.10 使用java客户端收发消息样例

AOP插件:

```
//发送消息
java -jar admq_client-2.4.1.jar amqp --service 172.24.4.187:6672 --action pub --vhost ns001
--exchange exchange001 --routing_key key --message 789 --print --repeat 50 --amqp_password
token:eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ0ZXN0MDEtMTg3In0.dy_zusN06H5x3aqHogKqgEx_ogl5aKESLLgQBzWMB_

//接收消息
java -jar admq_client-2.4.1.jar amqp --service 172.24.4.187:6672 --action sub --vhost
ns001 --queue queue0001 --print --amqp_password
token:eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ0ZXN0MDEtMTg3In0.dy_zusN06H5x3aqHogKqgEx_ogl5aKESLLgQBzWMB_
```

KOP插件:

//发送消息

```
java -jar admq_client-2.4.1.jar kafka --action pub --service 172.24.3.106:9092 --message 123
--topic tenant01/ns002/topic01 --user tenant01/ns002 --token
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJkZWZhdWx0In0.NyXw1WmMpV_8pyNE1bdE__mgwvKtPhojcfmYGePsDg8
```

//接收消息

```
java -jar admq_client-2.4.1.jar kafka --action sub --service 172.24.3.106:9092 --message 123
--topic tenant01/ns002/topic01 --user tenant01/ns002 --token
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJkZWZhdWx0In0.NyXw1WmMpV_8pyNE1bdE__mgwvKtPhojcfmYGePsDg8
```

MOTT插件:

//发送消息

```
java -jar admq_client-2.4.1.jar mqtt --service 172.24.3.105:5682 --action pub --topic
persistent://mqtt-tenant/mqtt-ns01/topic01 --message 消息hello789 --repeat 1000
```

//接收消息

```
java -jar admq_client-2.4.1.jar mqtt --service 172.24.3.105:5682 --action sub --topic
persistent://mqtt-tenant/mqtt-ns01/topic01
```

8.4 测试相关

8.4.1 有哪些第三方测试认证

- 2021/12/13 完成中测软评的产品测试。 [ZCRP_YZ_20210206评测用例及执行记录](#)
[ZCRP_IR_20210206-评测报告](#)

8.5 其他材料

- 产品彩页: [ADMO产品彩页](#)

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

