

RabbitMQ Operator 资产文档

一、资产基本介绍

• 资产简介

RabbitMQ 是一个由Erlang 语言开发的AMQP 的开源实现。它是应用层协议的一个开放标准，为面向消息的中间件设计，基于此协议的客户端与消息中间件可传递消息，并不受产品、开发语言等条件的限制。

RabbitMQ 较其他消息组件有以下优点：

- 由于erlang语言的特性，mq 性能较好，高并发。
- 吞吐量到万级，MQ功能比较完备。
- 健壮、稳定、易用、跨平台、支持多种语言、文档齐全
- 开源提供的管理界面非常棒，用起来很好用，社区活跃度高。


RabbitMQ Operator 可以在 Kubernetes 平台上一键部署一个高可用的 RabbitMQ 集群，同时集成时速云公有云 PaaS 平台的运维功能，实现对 RabbitMQ 集群的自动化运维。

• 核心能力

- 隐藏了高可用部署的复杂性，提供镜像模式的集群部署，保证数据的可靠性，适合生产环境使用。
- 通过自动化 Opeartor 生命周期管理简化运维工作：通过内置的 OLM (Operator Lifecycle Management) 框架，实现 Operator 的全生命周期自动化管理，大大简化了 Operator 运维工作。
- 无缝集成时速云公有云 PaaS 平台的存储供给与出口代理能力：创建 RabbitMQ 集群前服务无需提前准备集群存储，可以直接使用云原生应用平台提供的动态存储供给能力；集群创建完成后，可以轻松通过PaaS 平台提供的出口代理功能将集群暴露给外部访问。
- 常见运维操作完全自动化：支持集群的扩容。
- 企业级安全支持：所有镜像经过安全加固，通过镜像安全扫描。
- 完善的集群监控支持：支持查看集群的监控、日志、事件、审计信息，同时可以对集群设置告警策略，大大缩短了集群问题的发现、排查时间。

• 资产镜像安全扫描结果

- RabbitMQ Registry 镜像：dev-registry.tenxcloud.com/system_containers/daas-registry-rabbitmq:v1 镜像ID: c44036f45716
- RabbitMQ Bundle 镜像：dev-registry.tenxcloud.com/system_containers/rabbitmq-cluster-operator-bundle:v1 镜像ID: 1de534821a84
- RabbitMQ指标监控数据采集: dev-registry.tenxcloud.com/system_containers/rabbitmq_exporter:sha-8977cd6 镜像ID: 1d31500952de
- RabbitMQ Agent: dev-registry.tenxcloud.com/system_containers/rabbitmq-cluster-agent:v5.3.0 镜像ID: ee60da19dfd2
- RabbitMQ Operator 镜像：dev-registry.tenxcloud.com/system_containers/rabbitmq-cluster-operator:v5.3.0 镜像ID: 31a5100697ac


system_containers/rabbitmq
×

漏洞扫描

下载镜像 `docker pull 192.168.2.110/system_containers/rabbitmq`
📄
🔗 23

基本信息
版本及接口
属性
镜像分层
漏洞扫描
已创建工作负载

版本: 3.7
缺陷组件数: ✔ 13个
扫描时间: 2021-07-13 18:27:

漏洞扫描
刷新
导出

共计 201 条 < 1 / 21 >

缺陷码	严重度 ▾	组件	当前版本	修复版本
+ CVE-2016-2781	较低	coreutils	8.26-3	-
+ CVE-2019-9705	较低	cron	3.0pl1-128+deb9u1	-
+ CVE-2019-9704	较低	cron	3.0pl1-128+deb9u1	-
+ CVE-2020-6096	较低	glibc	2.24-11+deb9u3	-

- RabbitMQ 镜像: dev-registry.tenxcloud.com/system_containers/rabbitmq:3.7 镜像ID: e8261c2af9fe


system_containers/rabbitmq-cluster-operator
×

漏洞扫描

下载镜像 `docker pull 192.168.2.110/system_containers/rabbitm...`
📄
🔗 6

基本信息
版本及接口
属性
镜像分层
漏洞扫描
已创建工作负载

版本: v0.0.1-fp2
缺陷组件数: 🕒 13个 🕒 16个 ✔ 49个
扫描时间: 2021-07-13 18:27:

漏洞扫描
刷新
导出



未知漏洞

二、应用场景

- 异步处理

将业务中属于非核心或不重要的流程部分，使用消息异步通知的方式发给目标系统，这样主业务流程无需同步等待其他系统的处理结果，从而达到系统快速响应的目的。

- 应用解耦

以电商秒杀、抢购等流量短时间内暴增场景为例，传统做法是，用户下单后，订单系统发送查询请求到库存系统，等待库存系统返回请求结果给订单系统。如果库存系统发生故障，订单系统获取不到数据，订单失败。这种情况下，订单系统和库存系统两个子系统高耦合。

- 错峰流控与流量削峰

在电子商务系统或大型网站中，上下游系统处理能力存在差异，处理能力高的上游系统的突发流量可能会对处理能力低的某些下游系统造成冲击，需要提高系统的可用性的同时降低系统实现的复杂性。电商大促销等流量洪流突然来袭时，可以通过队列服务堆积缓存订单等信息，在下游系统有能力处理消息的时候再处理，避免下游订阅系统因突发流量崩溃。消息队列提供亿级消息堆积能力，3天的默认保留时长，消息消费系统可以错峰进行消息处理。

三、资产购买流程（补充注册、充值）

- 在“云市场”中查询需要购买的资产



- 查看资产详情



- 购买资产：在资产详情中点击“购买”



- 阅读《云原生应用市场用户协议》，同意后勾选“我已阅读并同意...”确认
- 点击“支付”

- 查看已购资产：购买资产后会自动跳转到“已购资产”页面显示被购买的资产



四、资产部署指南

- 查询购买(或自建)的 RabbitMQ Operator

- Kubernetes Master节点中执行如下操作，在指定的项目中添加私有Operator

```
* catalog.yaml
```

```
yaml
```

```
apiVersion: operators.coreos.com/v1alpha1
```

```
kind: CatalogSource
```

```
metadata:
```

```
  name: daas-registry-rabbitmq
```

```
spec:
```

```
  sourceType: grpc
```

```
  image: dev-registry.tenxcloud.com/system_containers/daas-registry-rabbitmq:v1
```

```
  displayName: tenxcloud registry
```

```
  publisher: tenxcloud
```

- * 在指定的项目中创建catalog

```
shell
```

```
[root@press2 olm]# kubectl -n gaozh create -f catalog.yaml
```

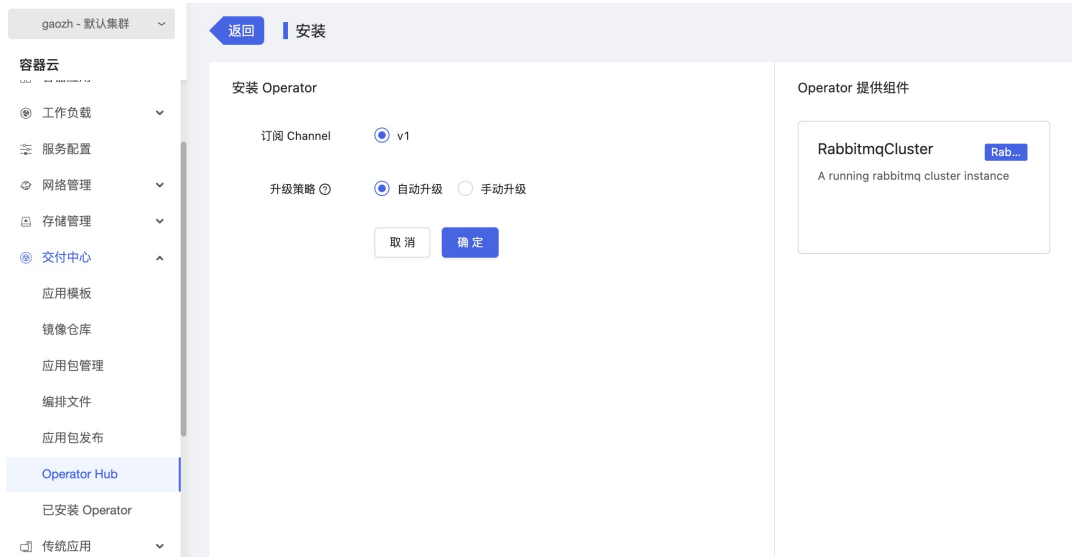
- 在容器云--交付中心--Opeartor Hub 里搜索 “RabbitMQ Cluser Operator”

```

```

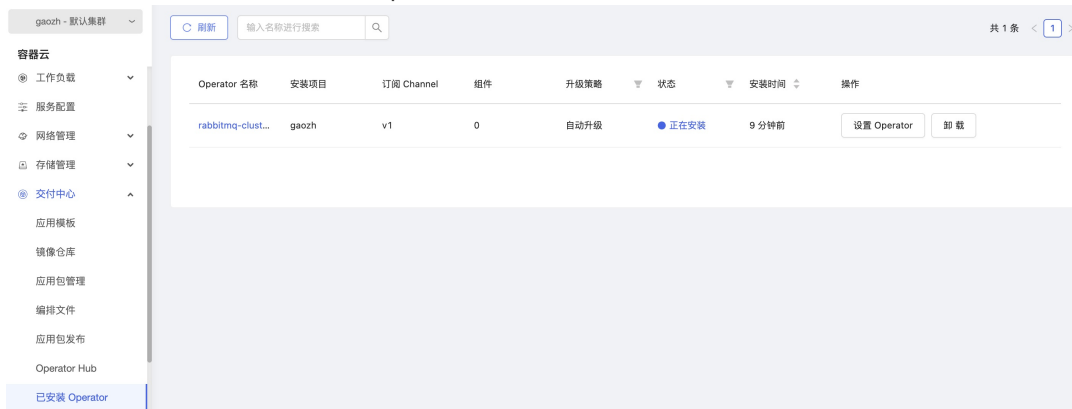
- 安装 RabbitMQ Operator

o 点击“安装”

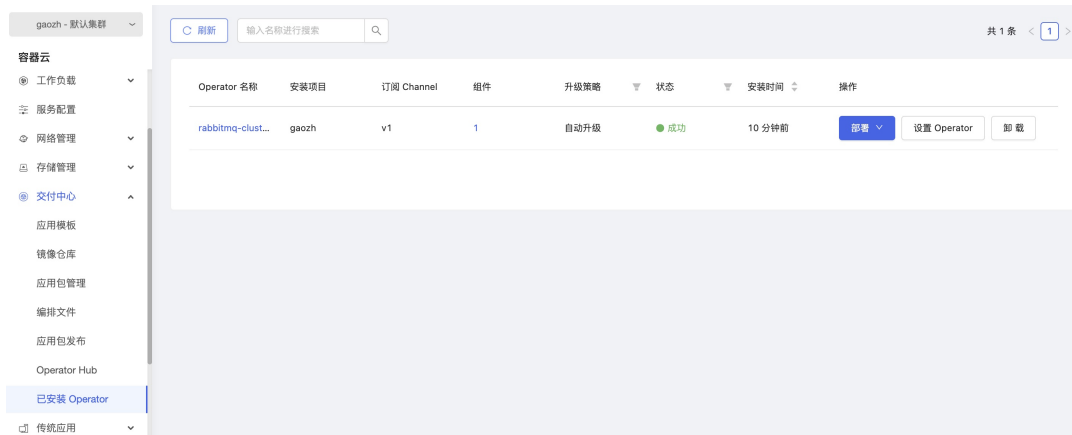


- 订阅 Channel: 一个 Channel 中可以包含一个或多个 Operator 版本
- 升级策略: 后续资产提供商有新 Operator 版本推送到该 Channel 后, 有两种升级策略:
 - 自动升级: 自动升级到 Channel 中新的版本
 - 手动升级: 需要手动确认后, 才能升级到 Channel 中新的版本

o 点击“确定”: 自动跳转到已安装 Operator 菜单



o 等待一段时间, 安装状态自动变为“成功”



o 点击 Operator 名称, 进入 Operator 详情

gaozh - 默认集群

返回 | Operator 详情

容器云

- 工作负载
- 服务配置
- 网络管理
- 存储管理
- 交付中心

应用模板

镜像仓库

应用包管理

编排文件

应用包发布

Operator Hub

已安装 Operator

传统应用

堆栈

rabbitmq-cluster-operator

状态: 成功

安装时间: 10 分钟前

刷新 卸载

详情 订阅 事件 Operator 应用

组件

rabbitmqclusters.daas.te...

RabbitmqCluster

创建 Operator 应用

描述

RabbitMQ Operator

o 查看 Operator 详情

详情 订阅 事件 Operator 应用

组件

rabbitmqclusters.daas.te...

RabbitmqCluster

创建 Operator 应用

描述

RabbitMQ Operator

ClusterServiceVersion Details

rabbitmq-cluster-operator.v0.0.1

详情 订阅 事件 Operator 应用

修改

Channel	v1
升级策略	自动升级
升级状态	最新 已安装2个

✓	RequirementsUnknown	requirements not yet checked	13 分钟前
✓	RequirementsNotMet	one or more requirements couldn't be found	13 分钟前
✓	AllRequirementsMet	all requirements found, attempting install	13 分钟前
✓	InstallSucceeded	waiting for install components to report healthy	13 分钟前
✓	InstallWaiting	installing: waiting for deployment rabbitmq-cluster-operator to become ready: Waiting for rollout to finish: 0 of 1 updated replicas are available...	13 分钟前
✓	InstallSucceeded	install strategy completed with no errors	12 分钟前

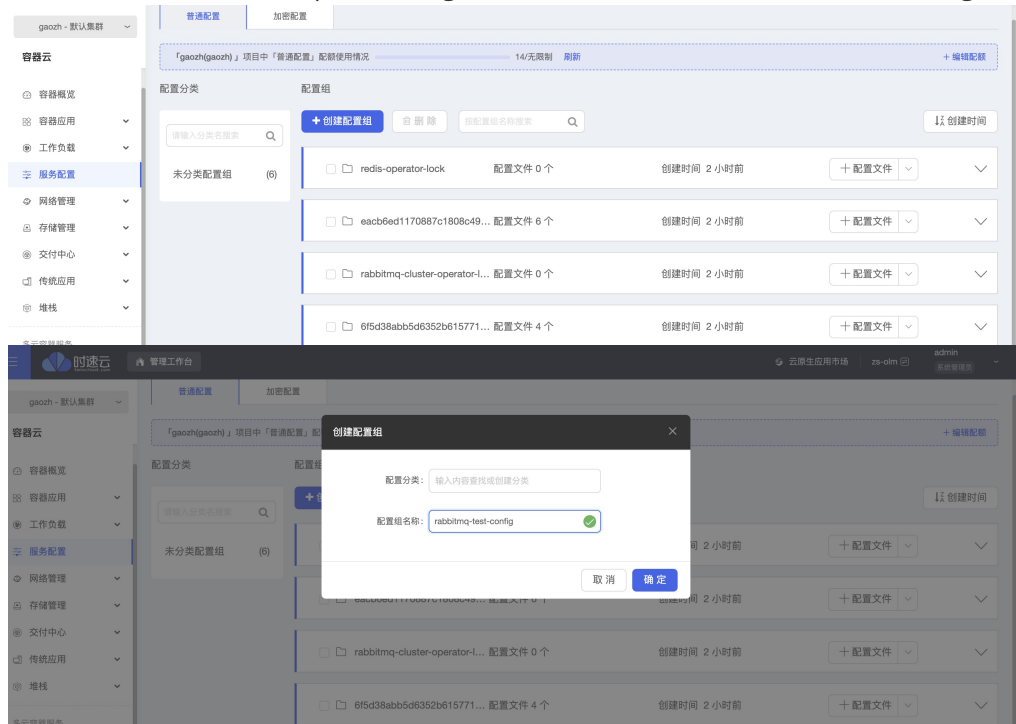
部署 刷新 输入 Operator 应用进行搜索 共 0 条 < 0 >

Operator 应用	状态	Operator 组件类型	安装时间	操作
暂无数据				

• 创建 RabbitMQ 集群 (补充界面部署方式)

◦ RabbitMQ Operator 部署完成后, 需要提前创建 RabbitMQ 集群配置

■ 创建配置名称如:"rabbitmq-test-config", 名称规范: "<RabbitMQ集群名称>-config"



■ 添加名称为"enabled_plugins"的配置文件, 内容为:

```
[rabbitmq_management,rabbitmq_peer_discovery_k8s].
```



- 添加名称为"rabbitmq.conf"的配置文件, 内容为:

```
cluster_formation.peer_discovery_backend =
rabbit_peer_discovery_k8s
cluster_formation.k8s.host = kubernetes.default
cluster_formation.k8s.address_type = hostname
cluster_formation.k8s.hostname_suffix = .rabbitmq-test.gaozh
cluster_formation.k8s.service_name = rabbitmq-test
cluster_formation.node_cleanup.interval = 30
cluster_formation.node_cleanup.only_log_warning = true
cluster_partition_handling = pause_minority

cluster_partition_handling = pause_minority
queue_master_locator=min-masters
loopback_users.guest = false
```

创建配置文件

即将保存一个配置文件，您可以在创建应用 → 添加服务时，关联使用该配置

名称：

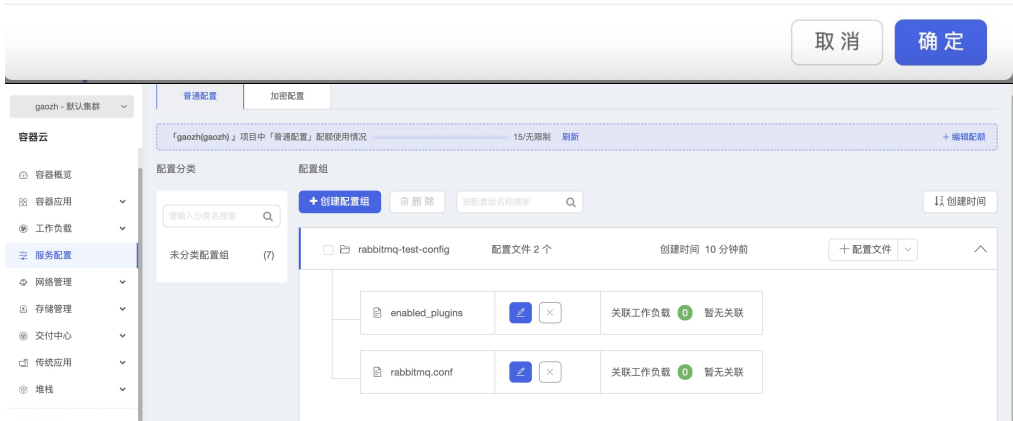
内容：导入或直接输入配置文件

本地文件导入 Git仓库导入

UTF-8

请上传文件或直接输入内容

```
配置文件内容 (读写)
1 cluster_formation.peer_discovery_backend = rabbit_peer_discovery_k8s
2 cluster_formation.k8s.host = kubernetes.default
3 cluster_formation.k8s.address_type = hostname
4 cluster_formation.k8s.hostname_suffix = .rabbitmq-test.gaozh
5 cluster_formation.k8s.service_name = rabbitmq-test
6 cluster_formation.node_cleanup.interval = 30
7 cluster_formation.node_cleanup.only_log_warning = true
8 cluster_partition_handling = pause_minority
9
10 cluster_partition_handling = pause_minority
11 queue_master_locator=min-masters
12 loopback_users.guest = false
13
```



配置说明:

* cluster_formation.k8s.hostname_suffix: 赋值规范如 ".<RabbitMQ集群名称>.<项目namespace>"

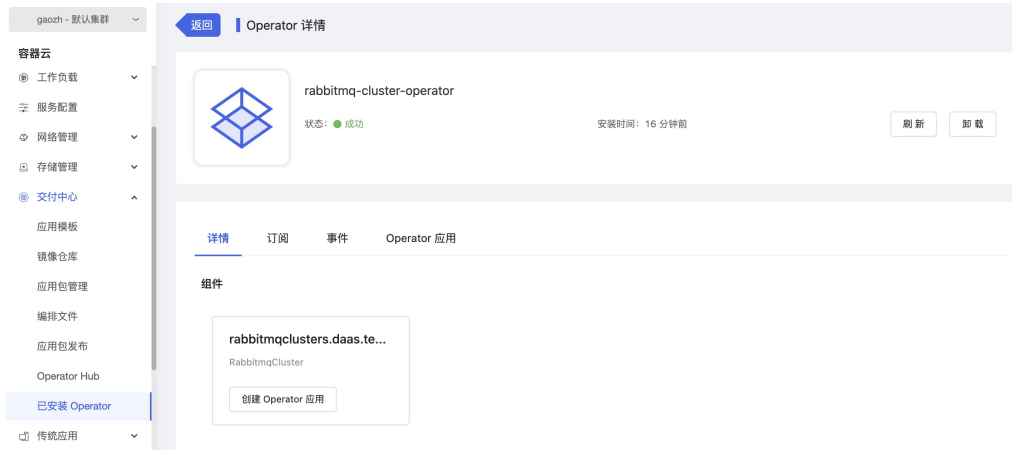
* cluster_formation.k8s.service_name: 赋值规范如 "<RabbitMQ集群名称>"

o RabbitMQ Operator 部署完成及配置创建完成后，可以从三个地方创建 RabbitMQ 集群:

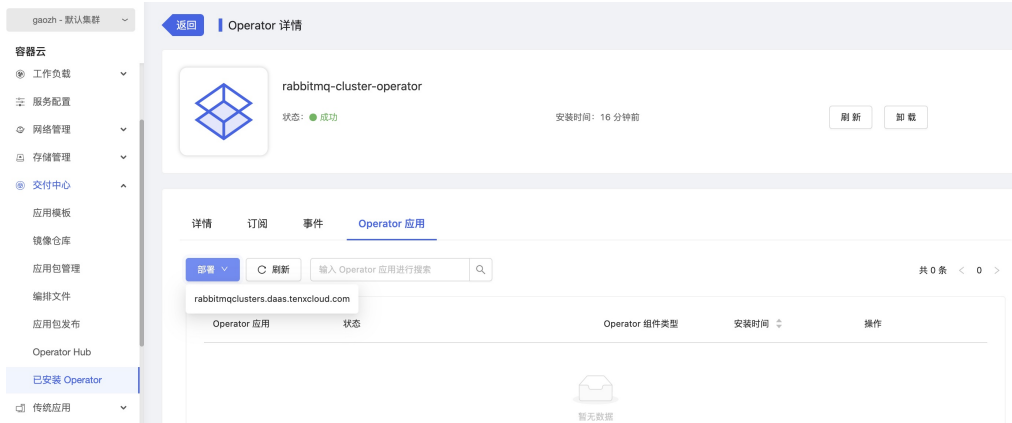
从 RabbitMQ Operator 列表



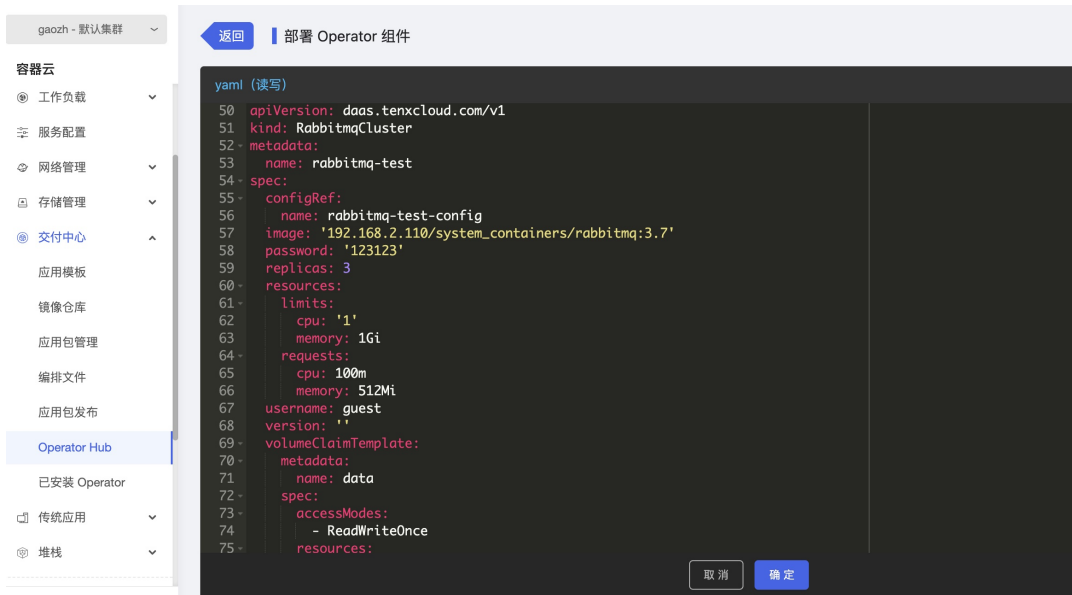
■ 从 RabbitMQ Operator 详情



■ 从 RabbitMQ Operator 详情里的 Operator 应用



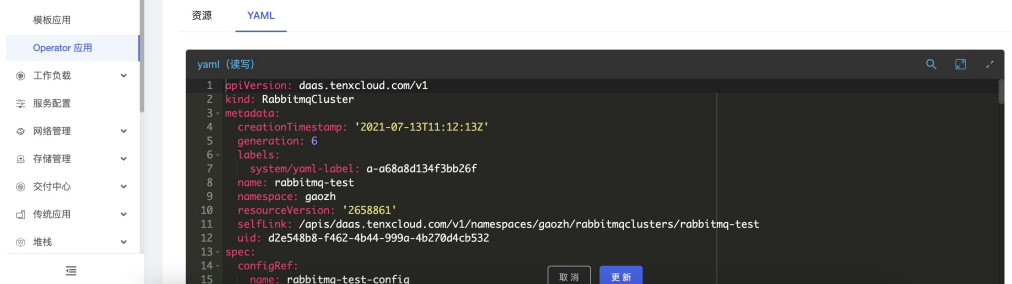
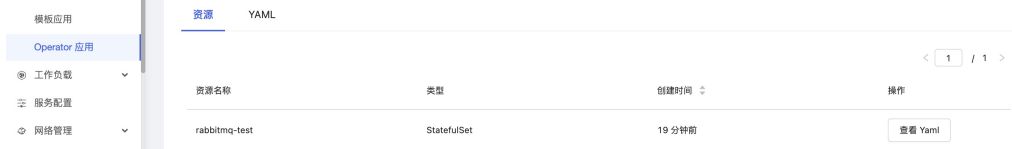
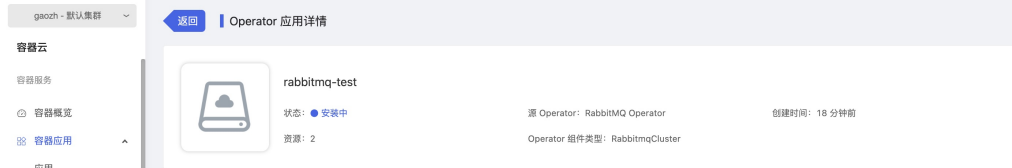
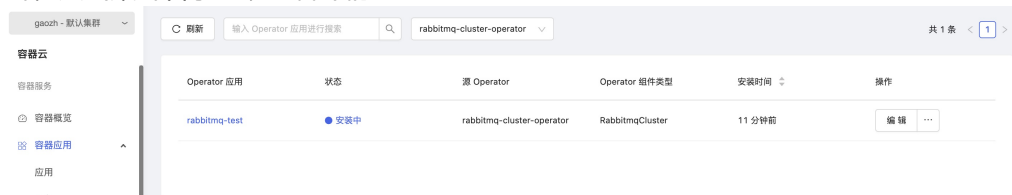
○ 创建 RabbitMQ 集群



- metadata.name: 集群名称。
- spec.resources.requests: 每个集群 Pod 请求的 CPU、内存资源大小，推荐使用 2C/4G 配置。
- spec.resources.limits: 每个集群 Pod 最大的 CPU、内存资源大小限制，推荐使用 2C/4G 配置。
- spec.password: 集群管理门户密码，需要修改。
- spec.configRef.name: 集群需要挂载的 RabbitMQ 配置，需要提前创建好对应的 configmap 配置
- spec.replicas: 集群副本数，推荐使用 3 节点、5 节点、7 节点，不支持其他节点数量的集群。
- spec.volumeClaimTemplate: 集群使用的存储类名称，从“容器云--存储管理--存储卷--创建存储卷--存储类下拉列表”中可以查看到可以使用的存储类

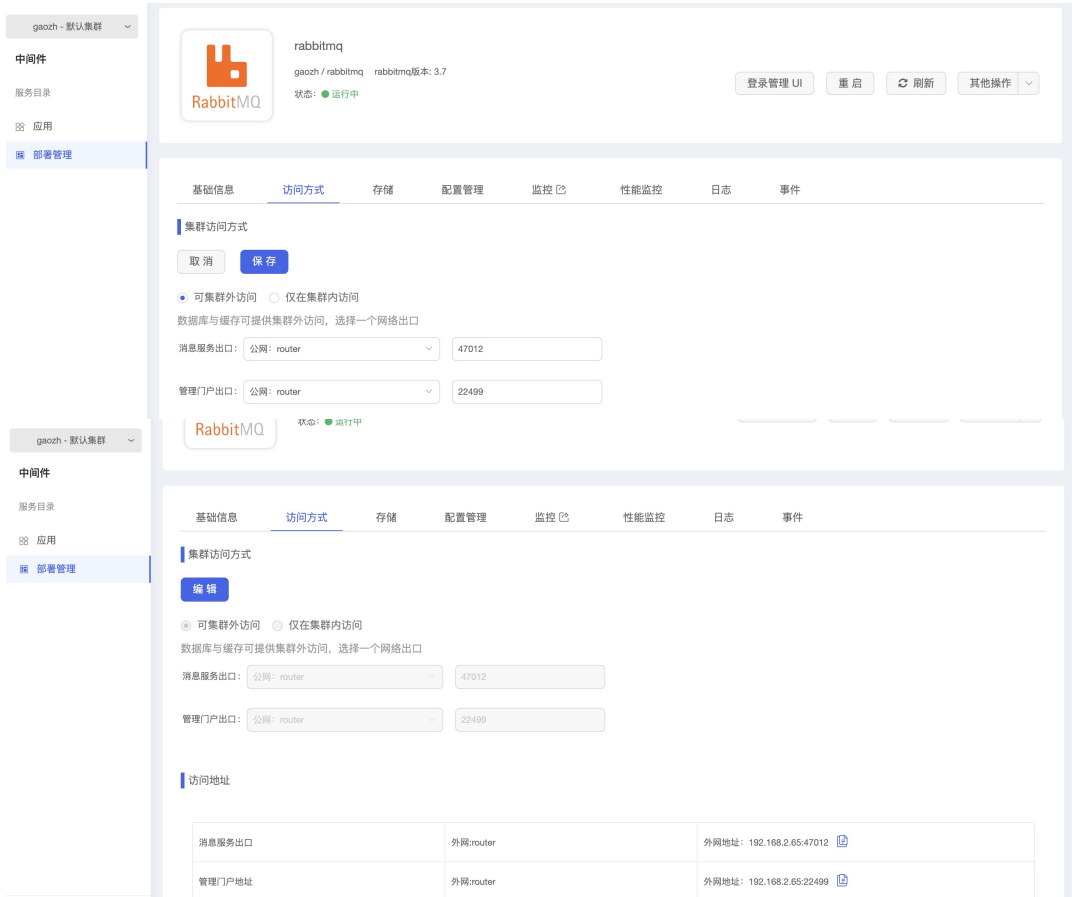


- spec.volumeClaimTemplate: 集群每个 Pod 使用的存储大小，根据业务数据量大小来设置
- 创建完成后，在“容器云--容器应用--Operator 应用”中可以查看创建的 RabbitMQ 集群，点击集群名字可以查看详情



- 配置 RabbitMQ 集群访问

○ 方案一(推荐), 在服务目录中设置访问方式



○ 方案二(不推荐), 通过添加应用路由暴露集群服务

* 在“容器云--网络管理--应用路由”页面, 点击“添加路由规则”



* 在“添加路由规则”页面, 选择一个服务出口代理 RabbitMQ 集群的admin服务



* 规则名称: 这条路由规则的名称

* 选择出口：选择一个平台的服务访问出口

* 选择服务：

* 需要访问RabbitMQ集群的管理页面，选择admin服务，名称格式为“<RabbitMQ集群名称>-admin-service”

* 需要访问RabbitMQ集群的通信接口，选择amqp服务，名称格式为“<RabbitMQ集群名称>-amqp-service”

* 点击“创建”后，可以看到生成的路由规则



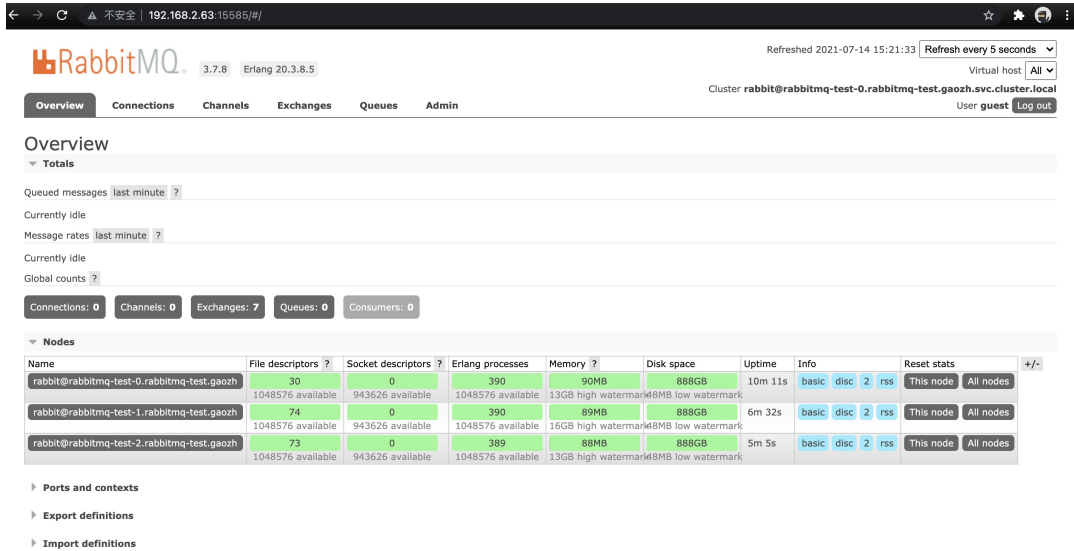
• 验证 RabbitMQ 集群访问

- 在“容器云--容器应用--容器服务”列表中，找到被代理的 RabbitMQ 集群的 Master 服务（<RabbitMQ集群名称>-master-service），点击“查看地址”，点击地址旁边的拷贝图标保存地址信息，用于后面访问验证。

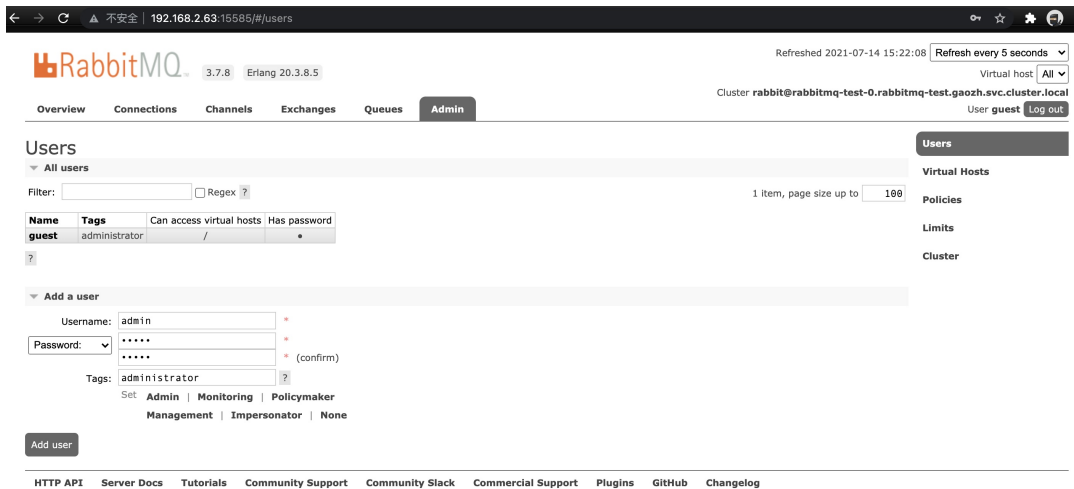


- 集群内：在 Kubernetes 集群内访问 RabbitMQ 服务，使用这个地址
- 服务代理-TCP：在 Kubernetes 集群外访问 RabbitMQ 服务，使用这个地址

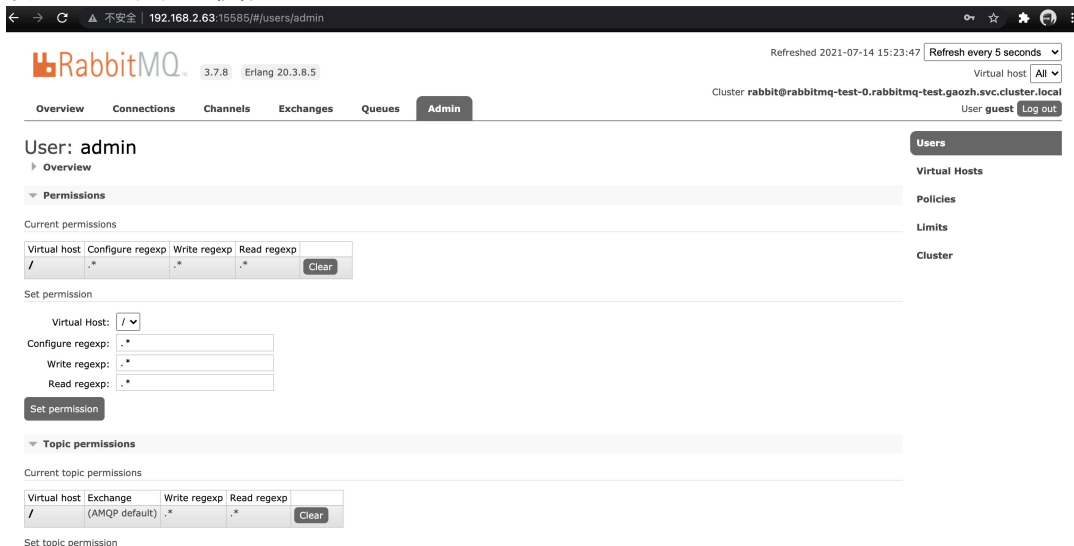
- 打开浏览器输入集群外访问地址，如"192.168.2.63:15585"，用户名：guest，密码：自定义



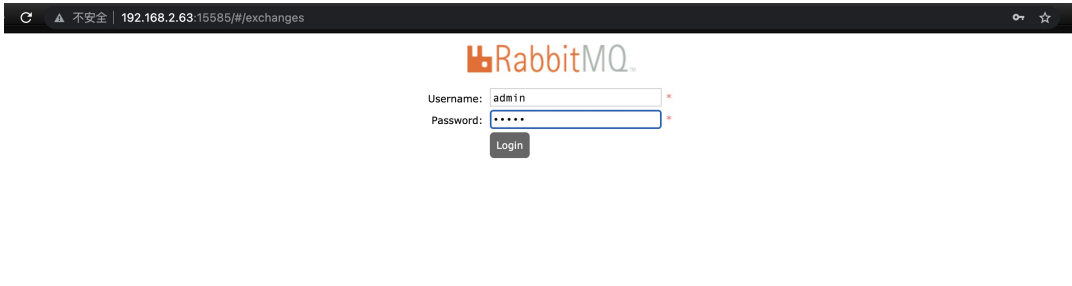
- 添加admin管理用户



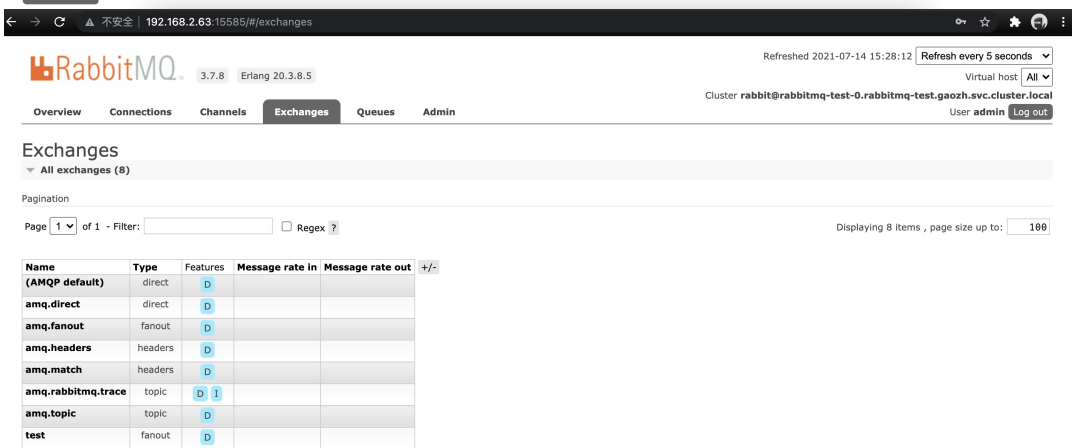
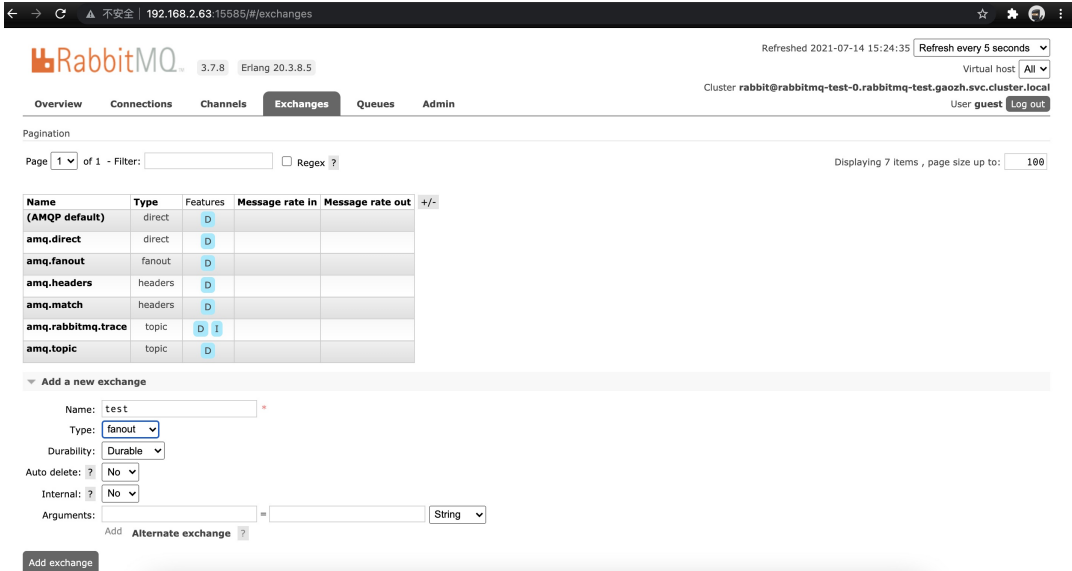
- 设置admin用户的权限



- 退出guest用户，用admin用户登录管理页面



- 添加交换，如: "test"



o 添加队列, 如: "test"

The screenshot shows the RabbitMQ web interface at 192.168.2.63:15585/#/queues. The 'Queues' tab is active, and the 'Add a new queue' form is displayed. The form fields are: Name: test; Durability: Durable; Node: rabbit@rabbitmq-test-0.rabbitmq-test-gaozh; Auto delete: No; Arguments: (empty). Below the form are links for Message TTL, Auto expire, Max length, Max length bytes, Overflow behaviour, Dead letter exchange, Dead letter routing key, Maximum priority, Lazy mode, and Master locator. An 'Add queue' button is at the bottom.

The screenshot shows the RabbitMQ web interface at 192.168.2.63:15585/#/queues. The 'test' queue is now listed in the 'All queues (1)' section. A table shows the queue details:

Overview	Node	Features	State	Ready	Unacked	Total	Message rates		
Name	Node						incoming	deliver	get ack
test	rabbit@rabbitmq-test-0.rabbitmq-test-gaozh	D		NaN	NaN	NaN			

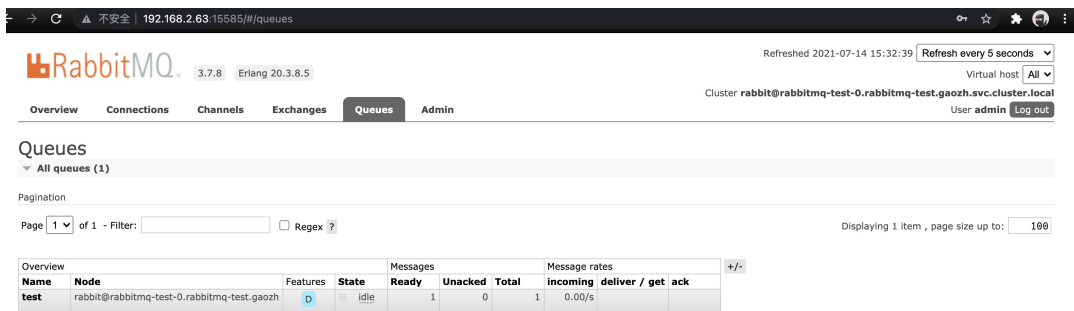
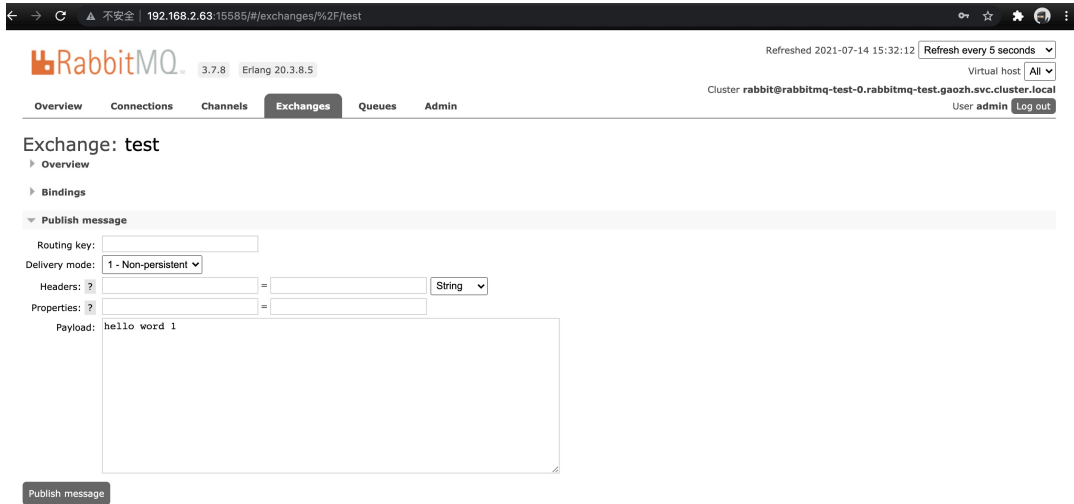
o 在"test交换"中绑定"test队列"

The screenshot shows the RabbitMQ web interface at 192.168.2.63:15585/#/exchanges/%2F/test. The 'Exchange: test' page is shown, and the 'Add binding from this exchange' form is displayed. The form fields are: To queue: test; Routing key: (empty); Arguments: (empty). A 'Bind' button is at the bottom.

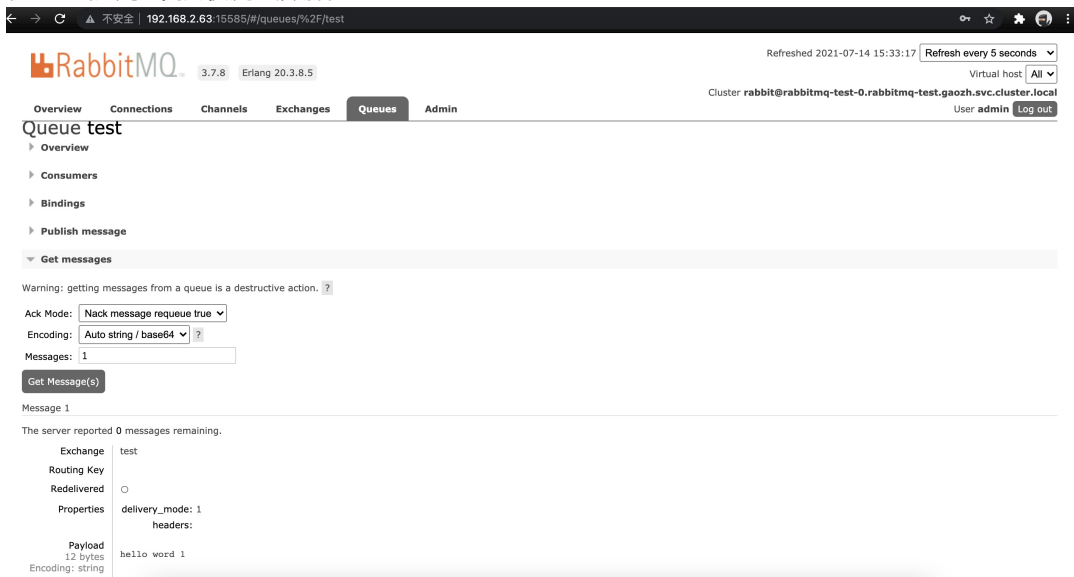
The screenshot shows the RabbitMQ web interface at 192.168.2.63:15585/#/exchanges/%2F/test. The 'Exchange: test' page is shown, and the 'Bindings' section displays a table with one binding:

To	Routing key	Arguments
test		Unbind

- 在"test交换"中发送测试数据"hello word 1"



- 在"test队列"中接收测试数据"hello word 1"



五、应用运维指南；（补充界面部署方式）

- **监控信息查看：**在“容器云--工作负载--有状态副本集”，点击进入 RabbitMQ 有状态副本集
- **日志信息查看：**在“容器云--工作负载--有状态副本集”，点击进入 RabbitMQ 有状态副本集
- **事件信息查看：**在“容器云--工作负载--有状态副本集”，点击进入 RabbitMQ 有状态副本集
- **审计信息查看：**在“安全和运维--平台运维--操作审计--审计记录”，选择“容器云/工作负载/容器组”、相应租户、项目后点击“立即查询”



