

Kafka Operator 资产文档

一、资产基本介绍

• 资产简介

Kafka 是最初由 LinkedIn 公司开发，是一个分布式、分区的、多副本的、多订阅者，基于 zookeeper 协调的分布式日志系统（也可以当做 MQ 系统），常见可以用于 web/nginx 日志、访问日志，消息服务等等，LinkedIn 于 2010 年贡献给了 Apache 基金会并成为顶级开源项目。

主要应用场景是：日志收集系统和消息系统。

Kafka 主要设计目标如下：

- 以时间复杂度为 $O(1)$ 的方式提供消息持久化能力，即使对 TB 级以上数据也能保证常数时间的访问性能。
- 高吞吐率。即使在非常廉价的商用机器上也能做到单机支持每秒 100K 条消息的传输。
- 支持 Kafka Server 间的消息分区，及分布式消费，同时保证每个 partition 内的消息顺序传输。
- 同时支持离线数据处理和实时数据处理。
- Scale out: 支持在线水平扩展。

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

• 核心能力

- 隐藏了高可用部署的复杂性：提供高可用模式，适合生产环境使用。
- 通过自动化 Operator 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 Kafka 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：支持集群的自动化备份（手动、定时）、恢复、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。
- 支持跨区高可用。
- 支持 Kafka 自身性能监控。

• 资产镜像安全扫描结果

- o CatalogSource 镜像: dev-registry.tenxcloud.com/dev-release-5_3_0/daas-kafka-registry:1.0.0 | b0b46feaec15



- o Olm-bundle 镜像: dev-registry.tenxcloud.com/dev-release-5_3_0/kafka-operator-bundle:1.0.0 | 49ea13f05e4e
- o Operator 镜像: dev-registry.tenxcloud.com/dev-release-5_3_0/kafka-cluster-operator:v5.3.0 | 6924692301fd
- o kafka 相关镜像:
 - dev-registry.tenxcloud.com/dev-release-5_3_0/kafka-exporter:v1.3.1 | 9dba221cd37c
 - dev-registry.tenxcloud.com/dev-release-5_3_0/kafka:2.13-2.6.0 | 308086154e92
 - dev-registry.tenxcloud.com/dev-release-5_3_0/kafka-manager:1.3.1.8 | be40e0189d08
 - dev-registry.tenxcloud.com/dev-release-5_3_0/jmx-javaagent:0.12.0 | ba844f123521

二、应用场景

• 监控

主机通过 Kafka 发送与系统和应用程序健康相关的指标，然后这些信息会被收集和处理从而创建监控仪表盘并发送警告。

• 消息队列

应用程序使用 Kafka 作为传统的消息系统实现标准的队列和消息的发布—订阅，例如搜索和内容提要 (Content Feed)。比起大多数的消息系统来说，Kafka 有更好的吞吐量，内置的分区，冗余及容错性，这让 Kafka 成为了一个很好的大规模消息处理应用的解决方案。消息系统一般吞吐量相对较低，但是需要更小的端到端延时，并尝尝依赖于 Kafka 提供的强大的持久性保障。在这个领域，Kafka 足以媲美传统消息系统，如 ActiveMR 或 RabbitMQ。

• 站点的用户活动追踪

为了更好地理解用户行为，改善用户体验，将用户查看了哪个页面、点击了哪些内容等信息发送到每个数据中心的 Kafka 集群上，并通过 Hadoop 进行分析、生成日常报告。

• 流处理

保存收集流数据，以提供之后对接的 Storm 或其他流式计算框架进行处理。很多用户会将那些从原始 topic 来的数据进行阶段性处理，汇总，扩充或者以其他方式转换到新的 topic 下再继续后面的处理。例如一个文章推荐的处理流程，可能是先从 RSS 数据源中抓取文章的内容，然后将其丢入一个叫做“文章”的 topic 中；后续操作可能是需要对这个内容进行清理，比如回复正常数据或者删除重复数据，最后再将内容匹配的结果返回给用户。这就在一个独立的 topic 之外，产生了一系列的实时数据处理的流程。

- **日志聚合**

使用 Kafka 代替日志聚合 (log aggregation)。日志聚合一般来说是从服务器上收集日志文件，然后放到一个集中的位置 (文件服务器或 HDFS) 进行处理。然而 Kafka 忽略掉文件的细节，将其更清晰地抽象成一个个日志或事件的消息流。这就让 Kafka 处理过程延迟更低，更容易支持多数数据源和分布式数据处理。比起以日志为中心的系统比如 Scribe 或者 Flume 来说，Kafka 提供同样高效的性能和因为复制导致的更高的耐用性保证，以及更低的端到端延迟。

- **持久性日志**

Kafka 可以为一种外部的持久性日志的分布式系统提供服务。这种日志可以在节点间备份数据，并为故障节点数据回复提供一种重新同步的机制。Kafka 中日志压缩功能为这种用法提供了条件。在这种用法中，Kafka 类似于 Apache BookKeeper 项目。

三、资产购买流程 (补充注册、充值)

- **添加 Kafka CatalogSource**

- 目前需要管理员进入后台手动创建 CatalogSource，样例 catsrc.yaml 其中 namespace 按需修改：

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: daas-registry-kafka
  namespace: one
spec:
  displayName: tenxcoud registry
  image: dev-registry.tenxcloud.com/system_containers/daas-kafka-registry:1.0.0
  publisher: TenxCloud
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 45m
```

```
## 安装 Kafka CatalogSource
[root@press2 ~]# kubectl apply -f catsrc.yaml

## 查看创建好的 CatalogSource
[root@press2 ~]# kubectl -n one get catsrc | grep kafka
daas-registry-kafka          tenxcoud registry   grpc   TenxCloud   47h

## 创建好的 CatalogSource 会自动创建对应的 pod
[root@press2 ~]# kubectl -n one get po | grep kafka
daas-registry-kafka-jd6wq    1/1
Running                      0           48m
```

在“云市场”中查询需要购买的资产

服务全局 - 不要求项目选择

云原生应用市场

我是用户

云市场

已购资产

订单管理

我是服务商

资产管理

出售管理

云市场

供资产交易与交付平台，实现应用、API、模型等多种交付介质的在平台上开放与交易。开放自有应用建设

kafka-opl

商品分类 **全部** 基础应用 网站建设 企业应用 数据库中间件集群

交付方式 **全部** 容器镜像 Operator Helm Chart 应用 API

价格 **全部** 免费 1 - 98 99 - 998 999 - 9998 9999 以上

默认 价格 更新时间 浏览量

应用 已认证

kafka

kafka-operator

74

kafka-operator

100 T 购买

查看资产详情

时速云 | 云市场

请输入名称关键字 搜索 控制台 admin

容器镜像 Operator Helm Chart 应用 API

kafka-operator Operator 已认证 立即购买

8 分钟前更新 2

产品介绍 价格 版本 服务支持

kafka-operator

版本 beta

版本描述 The Kafka operator is a Kubernetes operator to automate provisioning, management, autoscaling and operations of Apache Kafka clusters deployed to K8s.

Overview

Apache Kafka is an open-source distributed streaming platform, and some of the main features of the **Kafka-operator** are:

- the provisioning of secure and production ready Kafka clusters
- fine grained** broker configuration support
- advanced and highly configurable External Access via LoadBalancers using **Envoy**
- graceful Kafka cluster **scaling and rebalancing**
- monitoring via **Prometheus**
- encrypted communication using SSL
- automatic reaction and self healing based on alerts (plugin system, with meaningful default alert plugins) using **Cruise Control**
- graceful rolling upgrade
- advanced topic and user management via CRD

Prerequisites

The operator requires Cert Manager so you need to first install one if you don't already have one.

- **购买资产：**在资产详情中点击“购买”

我的订单

 请于 1 天 内完成付款，超时订单将自动取消

购买商品	kafka-operator
规格	¥ 100/天
有效时长	30 日
应付金额	100 T
* 付款项目	<input type="text" value="api资产-01"/> <input type="text" value="api"/>
	<input type="button" value="账户余额"/> (余额: 7000 T)
	<input checked="" type="checkbox"/> 我已阅读并同意《云原生应用市场用户协议》，并确保合法使用此资产或数据，不用于一切违法行为！
	<input type="button" value="支付"/>

- 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”

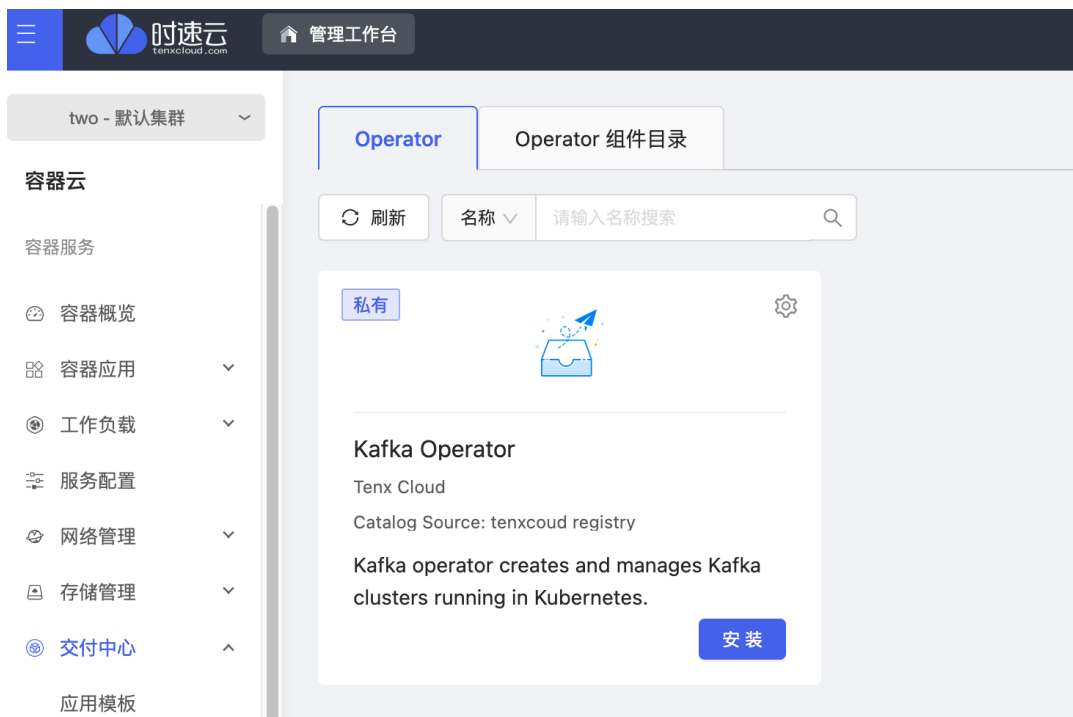
- **查看已购资产：**购买资产后会自动跳转到“已购资产”页面显示被购买的资产

资产名称	交付类型	初次购买时间	到期时间	使用情况	操作
kafka-operator	Operator	2021-07-14 18:39:02	2021-08-13 18:39:41	-	<input type="button" value="资产详情"/> ...

四、资产部署指南

- **查询购买的 Kafka Operator**

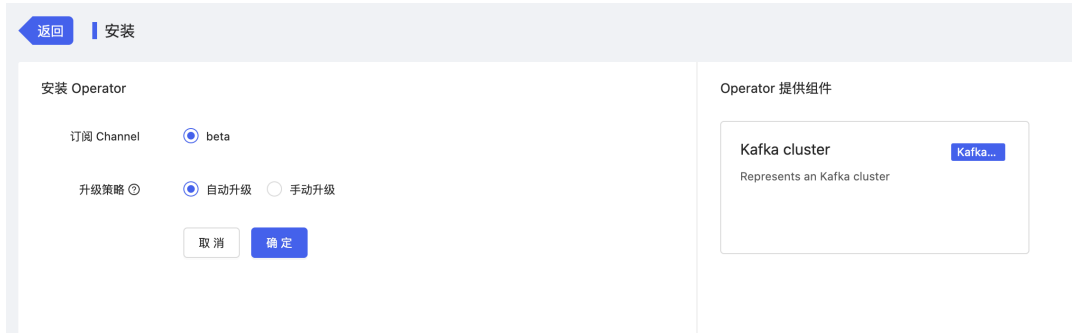
- 在容器云--交付中心--Opeartor Hub 里搜索 “Kafka Cluser Operator”



The screenshot shows the Tenx Cloud management console. The top navigation bar includes the Tenx Cloud logo and a '管理工作台' (Management Console) button. The left sidebar shows a navigation menu with '容器云' (Container Cloud) selected, and sub-items like '容器服务' (Container Service), '容器概览' (Container Overview), '容器应用' (Container Applications), '工作负载' (Workloads), '服务配置' (Service Configuration), '网络管理' (Network Management), '存储管理' (Storage Management), and '交付中心' (Delivery Center). The main content area displays the 'Operator' component catalog. A search bar is present with the text '名称 请输入名称搜索'. Below the search bar, there is a card for 'Kafka Operator' with a '私有' (Private) label and an '安装' (Install) button. The card text reads: 'Kafka Operator', 'Tenx Cloud', 'Catalog Source: tenxcoud registry', and 'Kafka operator creates and manages Kafka clusters running in Kubernetes.'

- **安装 Kafka Operator**

o 点击“安装”

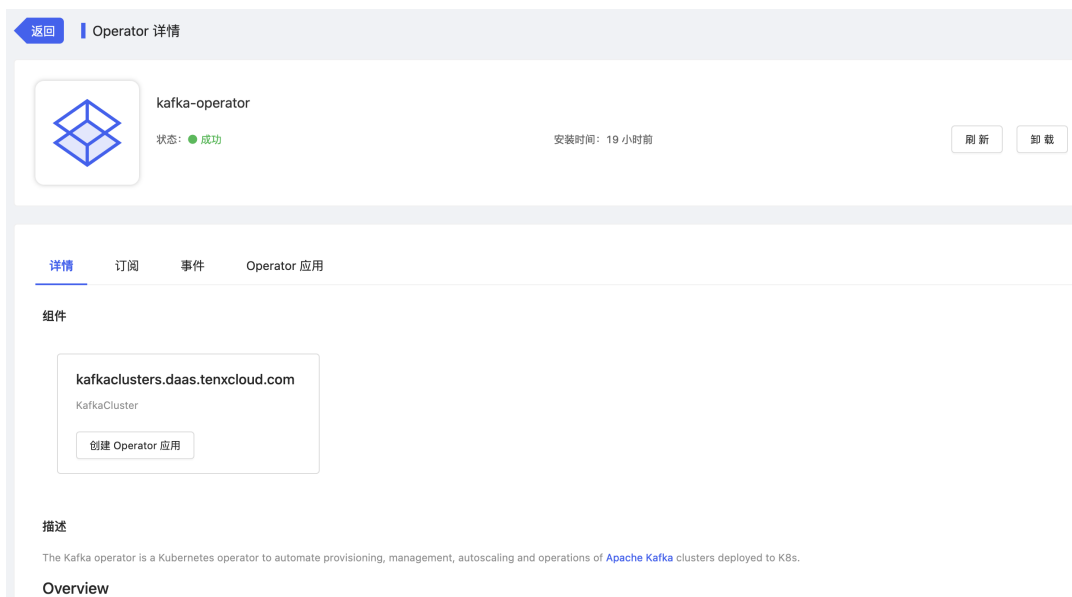


- 订阅 Channel: 一个 Channel 中可以包含一个或多个 Operator 版本
- 升级策略: 后续资产提供商有新 Operator 版本推送到该 Channel 后, 有两种升级策略:
 - 自动升级: 自动升级到 Channel 中新的版本
 - 手动升级: 需要手动确认后, 才能升级到 Channel 中新的版本

o 点击“确定”: 自动跳转到已安装 Operator 菜单。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



o 查看 Operator 详情

描述

The Kafka operator is a Kubernetes operator to automate provisioning, management, autoscaling and operations of [Apache Kafka](#) clusters deployed to K8s.

Overview

Apache Kafka is an open-source distributed streaming platform, and some of the main features of the **Kafka-operator** are:

- the provisioning of secure and production ready Kafka clusters
- **fine grained** broker configuration support
- advanced and highly configurable External Access via LoadBalancers using **Envoy**
- graceful Kafka cluster **scaling and rebalancing**
- monitoring via **Prometheus**
- encrypted communication using SSL
- automatic reaction and self healing based on alerts (plugin system, with meaningful default alert plugins) using **Cruise Control**
- graceful rolling upgrade
- advanced topic and user management via CRD

Prerequisites

The operator requires Cert Manager so you need to first install one if you don't already have one.

Kafka requires Zookeeper so you need to first have a Zookeeper cluster if you don't already have one.

Install Cert Manager

```
# pre-create cert-manager namespace and CRDs per their installation instructions
kubectl apply -f https://raw.githubusercontent.com/jetstack/cert-manager/v0.10.1/deploy/manifests/01-namespace.yaml
# Install the CustomResourceDefinitions and cert-manager itself
kubectl apply -f https://github.com/jetstack/cert-manager/releases/download/v0.10.1/cert-manager.yaml
```

Install Zookeeper

To install Zookeeper we recommend using the [Pravega's Zookeeper Operator](#).

ClusterServiceVersion Details

kafka-operator.0.6.1

• 创建 Kafka 集群

- Kafka Operator 部署完成后，可以从三个地方创建 Kafka 集群：

■ 从 Kafka Operator 列表

Operator 名称	安装项目	订阅 Channel	组件	升级策略	状态	安装时间	操作
kafka-operator	one	beta	1	自动升级	成功	19 小时前	部署 设置 Operator 卸载

■ 从 Kafka Operator 详情

返回 | Operator 详情

kafka-operator
状态: 成功
安装时间: 19 小时前
刷新 | 卸载

详情 | 订阅 | 事件 | Operator 应用

组件

kafkaclusters.daas.tenxcloud.com
KafkaCluster
创建 Operator 应用

描述

The Kafka operator is a Kubernetes operator to automate provisioning, management, autoscaling and operations of [Apache Kafka](#) clusters deployed to K8s.

Overview

■ 从 Kafka Operator 详情里的 Operator 应用

返回 | Operator 详情

kafka-operator
状态: ● 成功
安装时间: 19 小时前
刷新 卸载

详情 订阅 事件 **Operator 应用**

部署: ▼ C 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
-------------	----	---------------	------	----

○ 准备 ZK 集群

由于 Kafka 集群依赖于 ZK 集群，故需要先准备一个可用的 ZK 集群，推荐在“服务目录”-“应用”中创建 ZK 集群

服务目录: 一个中间件与大数据的完整交付平台, 包含云化的中间件、大数据应用的全生命周期管理。

分类: 全部 **数据库** 缓存 中间件 任务调度 其他

MySQL 集群
MySQL 集群是一种分布式设计, 目标是达到没有任... 部署

MongoDB 副本集
MongoDB 副本集完全兼容 MongoDB 协议, 提供稳... 部署

ElasticSearch 集群
ElasticSearch 支持分布式部署, 通过集群部署可以提... 部署

ZooKeeper 集群
ZooKeeper 是一个开源的分布式协调服务, 设计目标... 部署

one - 默认集群

中间件

服务目录

应用

部署管理

olm-zk
one / olm-zk zookeeper版本: 3.5.5
集群模式: 集群模式 ⓘ
状态: ● 运行中

基础信息 配置管理 访问方式 容器监控 日志 事件

参数

用户名: super
密码: ***** 显示

zookeeper版本 编辑
3.5.5

资源配置 编辑

2X 自定义
512 MB 内存
0.2-1 核 CPU

1024 ~ 2048 MB 内存
1 ~ 2 核 CPU

节点调度 编辑
Tips: 修改节点调度规则, 集群将会重启, 重启过程将无法提供服务, 请谨慎操作!
 使用系统默认调度 定义服务与节点亲和性

实例副本3个 更改实例数

o 创建 Kafka 集群

三 时速云 tenxcloud.com 管理工作台

one - 默认集群

返回 | 部署 Operator 组件

容器云

- 容器服务
 - 容器概览
 - 容器应用
 - 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
- 多云容器服务
 - 联邦应用
 - 联邦服务
 - 联邦工作负载
 - 联邦应用路由
 - 联邦配置
 - 联邦存储

```
yaml (读写)
1 - apiVersion: daas.tenxcloud.com/v1
2   kind: KafkaCluster
3   metadata:
4     annotations:
5       kafka-manager: kafka-olm-manager-svc
6       kafka-manager-port: '80'
7       manager.system/schemaPortname: kafka-olm-manager-svc/TCP/56672
8       system/daas-reboot: disable
9       system/lbgroup: group-xrfod
10      system/schemaPortname: 'kafka-olm-all-svc/TCP/26793,16861,29637'
11    name: kafka-olm
12    namespace: one
13  spec:
14    brokerConfigs:
15      - config: |
16          auto_create_topics_enable=true
17          socket.request.max.bytes=1195725956
18          zookeeper.connection.timeout.ms=60000
19          zookeeper.session.timeout.ms=6000
20          externalPort: 26793
21          id: 0
22          image: '192.168.2.38/system_containers/kafka:2.13-2.6.0'
23          kafkaHeapOpts: '-Xmx4096m -Xms4096m'
24          resourceReqs:
25            limits:
26              cpu: '2'
27              memory: 8Gi
28            requests:
29              cpu: '0.5'
30              memory: 256Mi
31          storageConfigs:
32            - mountPath: /kafka-logs
33              storageClassName: 2-38
34          zkPassword: 'root'
35          zkUsername: zookeeper_admin
36      - config: |
37          auto_create_topics_enable=true
38          socket.request.max.bytes=1195725956
39          zookeeper.connection.timeout.ms=60000
40          zookeeper.session.timeout.ms=6000
41          externalPort: 16861
42          id: 1
43          image: '192.168.2.38/system_containers/kafka:2.13-2.6.0'
44          kafkaHeapOpts: '-Xmx4096m -Xms4096m'
45          resourceReqs:
46            limits:
47              cpu: '2'
48              memory: 8Gi
49            requests:
50              cpu: '0.5'
51              memory: 256Mi
52          storageConfigs:
53            - mountPath: /kafka-logs
54              storageClassName: 2-38
55          zkPassword: '123'
56          zkUsername: zookeeper_admin
57      - config: |
58          auto_create_topics_enable=true
59          socket.request.max.bytes=1195725956
60          zookeeper.connection.timeout.ms=60000
61          zookeeper.session.timeout.ms=6000
62          externalPort: 29637
```

三 时速云 tenxcloud.com 管理工作台

one - 默认集群

返回 | 部署 Operator 组件

容器云

- 容器服务
 - 容器概览
 - 容器应用
 - 工作负载
 - 服务配置
 - 网络管理
 - 存储管理

```
yaml (读写)
41 externalPort: 16861
42 id: 1
43 image: '192.168.2.38/system_containers/kafka:2.13-2.6.0'
44 kafkaHeapOpts: '-Xmx4096m -Xms4096m'
45 resourceReqs:
46   limits:
47     cpu: '2'
48     memory: 8Gi
49   requests:
50     cpu: '0.5'
51     memory: 256Mi
52 storageConfigs:
53   - mountPath: /kafka-logs
54     storageClassName: 2-38
```

```

54 storageClassName: 2-38
55 zkPassword: '123'
56 zkUsername: zookeeper_admin
57 - config: |
58   auto_create_topics_enable=true
59   socket.request.max.bytes=1195725956
60   zookeeper.connection.timeout.ms=60000
61   zookeeper.session.timeout.ms=6000
62   externalPort: 29637
63   id: 2
64   image: '192.168.2.38/system_containers/kafka:2.13-2.6.0'
65   kafkaHeapOpts: '-Xmx4096m -Xms4096m'
66   resourceReqs:
67     limits:
68       cpu: '2'
69       memory: 8Gi
70     requests:
71       cpu: '0.5'
72       memory: 256Mi
73   storageConfigs:
74     - mountPath: /kafka-logs
75       storageClassName: 2-38
76   zkPassword: '123'
77   zkUsername: zookeeper_admin
78   cruiseControlConfig: {}
79   description: kafka-olm
80   envoyConfig:
81     image: '192.168.2.38/system_containers/envoy:0.1.0'
82   exporterImage: '192.168.1.52/system_containers/kafka-exporter:v1.3.1'
83   externalGroup: group-xrfod
84   externalIP: 192.168.3.116
85   headlessServiceEnabled: false
86   kafkaManger:
87     image: '192.168.2.38/system_containers/kafka-manager:1.3.1.8'
88     zkAddress: 'hxl-zk-service.aaaa:2181'
89   listenersConfig:
90     internallisteners:
91     - containerPort: 29092
92       name: plaintext
93       type: plaintext
94       usedForInnerBrokerCommunication: true
95   monitoringConfig:
96     jmxImage: '192.168.2.38/system_containers/jmx-javaagent:0.12.0'
97     pathToJar: ''
98   oneBrokerPerNode: false
99   replicas: 3
100   zkAddresses:
101   - 'hxl-zk-service.aaaa:2181/aaaa/kafka-olm'

```


- metadata.name: 集群名称
- metadata.namespace: 命名空间
- spec.brokerConfigs.image: broker 镜像地址
- spec.brokerConfigs.resourceReqs: 每个集群 broker Pod 请求的 CPU、内存资源大小，推荐使用 2C/8G 配置
- spec.brokerConfigs.storageConfigs.storageClassName: 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.brokerConfigs.zkPassword: ZK 集群密码，需要修改
- 注意：多个 Kafka 节点会生成多个 spec.brokerConfigs.config 节点，每个节点均需要手动修改
- spec.envoyConfig.image: 修改镜像仓库地址
- spec.exporterImage: 修改镜像仓库地址
- spec.kafkaManger.image: 修改镜像仓库地址

- spec.kafkaManger.zkAddress: 修改 zk 服务地址
- spec.monitoringConfig.jmxImage: 修改镜像仓库地址
- spec.zkAddress: 修改 zk 服务地址 (包括 namespace)
- 创建完成后, 在“容器云--容器应用--Operator 应用”中可以查看创建的 Kafka 集群, 点击集群名字可以查看详情

返回 | Operator 详情



kafka-operator
状态: ● 成功
安装时间: 21 小时前
刷新 卸载

详情 订阅 事件 **Operator 应用**

部署 刷新 olm

Operator 应用	状态	Operator 组件类型	安装时间	操作
kafka-olm	● 未知	KafkaCluster	21 小时前	编辑 ...


返回 | Operator 应用详情


kafka-olm
状态: ● 未知
源 Operator: Kafka Operator
创建时间: 21 小时前
资源: 0
Operator 组件类型: KafkaCluster

资源 YAML

资源名称	类型	创建时间	操作
暂无数据			

返回 | Operator 应用详情


kafka-olm
状态: ● 未知
源 Operator: Kafka Operator
创建时间: 21 小时前
资源: 0
Operator 组件类型: KafkaCluster

资源 **YAML**

```

yaml (读写)
1  apiVersion: daas.tenxcloud.com/v1
2  kind: KafkaCluster
3  metadata:
4  - annotations:
5    kafka-manager: kafka-olm-manager-svc
6    kafka-manager-port: '80'
7    manager.system/schemaPortName: kafka-olm-manager-svc/TCP/56672
8    system/daas-reboot: disable
9    system/lbgroup: group-xrfd
10   system/schemaPortName: 'kafka-olm-all-svc/TCP/26793,16861,29637'
11  creationTimestamp: '2021-07-12T11:32:25Z'
12  generation: 1
13  labels:
14   system/yaml-label: a-584de1329394482d
15  name: kafka-olm
16  namespace: one
  
```

取消 更新

• 验证 Kafka 集群访问

- 可以将 Kafka 管理服务暴露出来验证访问管理页面。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 Kafka 集群的 mg 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: Kafka 集群的 mg 服务, 名称格式为“-mg-svc”

返回 | 修改路由规则

* 规则名称: kafka-status-mg-svc

规则描述: 请输入规则描述

* 选择出口: 2.63

* 选择服务: kafka-status-mg-svc

* 端口协议: + 端口协议

服务端口协议	代理端口协议	操作
80 TCP	TCP 39369	✕

取消 修改

- 在“容器云--容器应用--容器服务”列表中，找到被代理的 Kafka 集群的管理 mg 服务 (<Kafka 集群名称>-mg-svc)，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 Kafka 服务，使用这个地址
 - 服务代理-TCP: 在 Kubernetes 集群外访问 Kafka 服务，使用这个地址

「one(one)」项目中「服务」配额使用情况 25/无限制 刷新

+ 创建 Service 启动 停止 刷新 删除 重新部署 stat

服务名称	状态	访问地址	创建时间
kafka-status-mg-svc	全部运行 1/1 个版本运行	查看地址	2 天前

192.168.2.63:39369

Kafka Manager Cluster

Clusters

Clusters

Active	Operations	Version
kafka-olm	<div style="border: 1px solid blue; padding: 2px; display: inline-block;">Modify</div> <div style="border: 1px solid orange; padding: 2px; display: inline-block;">Disable</div>	2.2.0

- 进入“容器云-工作负载-任务”，测试创建一个名为 hxl001 的 Topic

```

apiVersion: batch/v1
kind: Job
metadata:
  name: kafka-job
  namespace: one
spec:
  template:
    spec:
      restartPolicy: Never
      containers:
        - command: ["/opt/kafka/bin/kafka-topics.sh", "--create", "--bootstrap-server", "kafka-status-all-svc:29092", "--topic", "hx1001", "--partitions", "1", "--replication-factor", "1"]
          image: 192.168.2.110/system_containers/kafka:2.12-2.2.0
          name: jobs-kafka

```

- 修改以上 namespace 字段
- 修改 zk 服务地址
- 修改镜像地址

The screenshot shows the '时速云' management console. The main area displays a YAML configuration for a Job in the 'one' namespace. The configuration is as follows:

```

1 apiVersion: batch/v1
2 kind: Job
3 metadata:
4   name: kafka-job
5   namespace: one
6 spec:
7   template:
8     spec:
9       restartPolicy: Never
10      containers:
11        - command: ["/opt/kafka/bin/kafka-topics.sh", "--create", "--bootstrap-server", "kafka-status-all-svc:29092", "--topic", "hx1001", "--partitions", "1", "--replication-factor", "1"]
12          image: 192.168.2.110/system_containers/kafka:2.12-2.2.0
13          name: jobs-kafka

```

Below the configuration, there is a table showing the execution status of the Job:

名称	状态	CPU	内存	服务质量	镜像	所属节点	创建时间
kafka-job-9rwkt	执行完成 已重启 0 次	0.20 ~ 2.00C	0.50 ~ 2.00 Gi	有限超卖	查看镜像地址	192.168.2.63	11 分钟前

- Topic 创建成功后, 进入"容器云-工作负载-部署", 模拟 Producer 发送消息

The screenshot shows the '时速云' management console. The main area displays a YAML configuration for a Deployment in the 'one' namespace. The configuration is as follows:

```

1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: test-kafka-producer
5   namespace: one
6 spec:
7   replicas: 1
8   selector:
9     matchLabels:
10      app: test-kafka-producer
11   template:
12     metadata:
13       labels:
14         app: test-kafka-producer
15     spec:
16       containers:
17         - command:
18           - sleep
19           - "3600"
20           name: kafka
21           image: 192.168.2.110/system_containers/kafka:2.12-2.2.0

```

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: test-kafka-producer
  namespace: one
spec:

```

```

replicas: 1
selector:
  matchLabels:
    app: test-kafka-producer
template:
  metadata:
    labels:
      app: test-kafka-producer
  spec:
    containers:
      - command:
        - sleep
        - "3600"
        name: kafka
        image: 192.168.2.110/system_containers/kafka:2.12-2.2.0

```

- 修改以上 namespace 字段
- 修改镜像地址

在 kafka 容器组详情中，点击登陆终端里的 kafka 容器

The screenshot shows a Kubernetes dashboard interface. At the top, there's a '部署详情' (Deployment Details) section for 'test-kafka-producer'. It shows the deployment is running with 1/1 replicas. Below this is a table of container groups. One container group is highlighted: 'test-kafka-producer-75b4fb6f65-psg5t', which is in a '运行中' (Running) state. Below the table, there's a '容器组详情' (Container Group Details) section for the selected container group. It shows the pod name, status, and a '选择容器' (Select Container) dropdown menu with 'kafka' selected. At the bottom, there's a '资源现状' (Resource Status) section showing Pod IP, node name, and restart policy.

- 登陆终端里的 kafka 容器后，模拟 Producer 发送消息

```

bash-4.4# /opt/kafka/bin/kafka-console-producer.sh --topic hx1001 --
broker-list kafka-status-all-svc:29092
>hello kafka
>

```

[返回](#)

容器组详情



test-kafka-producer-75b4fb6f65-psg5t

状态: ● 运行中

上级资源

标签: app: test-kafka-producer

pod-template-hash: 75b4fb6f65

创建时间

节点选择器: --

[概览](#)[配置](#)[监控](#)[实时日志](#)[事件](#)[进程](#)

资源现状

Pod IP	172.31.20.79
所属节点	192.168.2.63
重启策略	Always

test-kafka-producer-7... X

```
bash-4.4# /opt/kafka/bin/kafka-console-producer.sh --topic hxl001 --broker-list kafka-status-all-svc:29092
>hello kafka
>[]
```

- 进入“容器云-工作负载-部署”，验证 Consumer 能够接收到消息

The screenshot shows the '部署' (Deployment) section in the container management console. The central area displays a YAML configuration for a Deployment named 'test-kafka-consumer' in the 'one' namespace. The configuration includes 1 replica, a selector matching 'app: test-kafka-consumer', and a container named 'kafka' with the command 'sleep 3600' and image '192.168.2.110/system_containers/kafka:2.12-2.2.0'. The right sidebar shows a 'Yaml 辅助工具' (YAML Assistant) with a 'Deployment x' dropdown and a list of instructions for creating and configuring the deployment.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test-kafka-consumer
  namespace: one
spec:
  replicas: 1
  selector:
    matchLabels:
      app: test-kafka-consumer
  template:
    metadata:
      labels:
        app: test-kafka-consumer
    spec:
      containers:
        - command:
            - sleep
            - "3600"
          name: kafka
          image: 192.168.2.110/system_containers/kafka:2.12-2.2.0
```



```
kafka-console-consumer.sh --bootstrap-server kafka-status-all-svc:29092
--topic hxl001 --from-beginning
>hello kafka
```

- 修改以上 namespace 字段
- 修改 kafka 服务地址
- 修改镜像地址

The screenshot shows a container management interface for a pod named 'test-kafka-consumer-795df59cc5-jr7hr'. The pod is in a 'Running' state. The interface includes a '选择容器' (Select Container) dropdown menu with 'kafka' selected and a '登录终端' (Login Terminal) button. Below the pod details, there are tabs for '概览' (Overview), '配置' (Configuration), '监控' (Monitoring), '实时日志' (Real-time Logs), '事件' (Events), and '进程' (Processes). The '资源现状' (Resource Status) section shows the following information:

Pod IP	172.31.20.125
所属节点	192.168.2.63
重启策略	Always

登陆终端里的 kafka 容器后，模拟 Consumer 消费消息

The screenshot shows the same container management interface as above, but with the terminal output visible. The terminal shows the following commands and output:

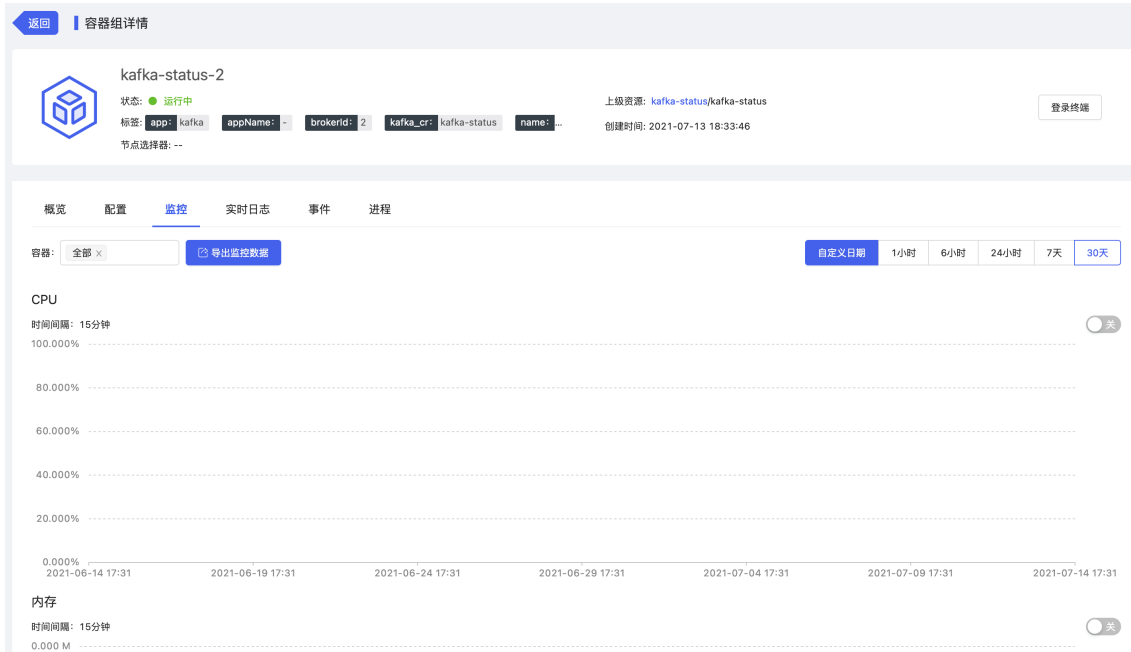
```
bash-4.4# kafka-console-consumer.sh --bootstrap-server kafka-status-all-svc:29092 --topic hxl001 --from-beginning
hahaha hxl
hahaha hxl

test123

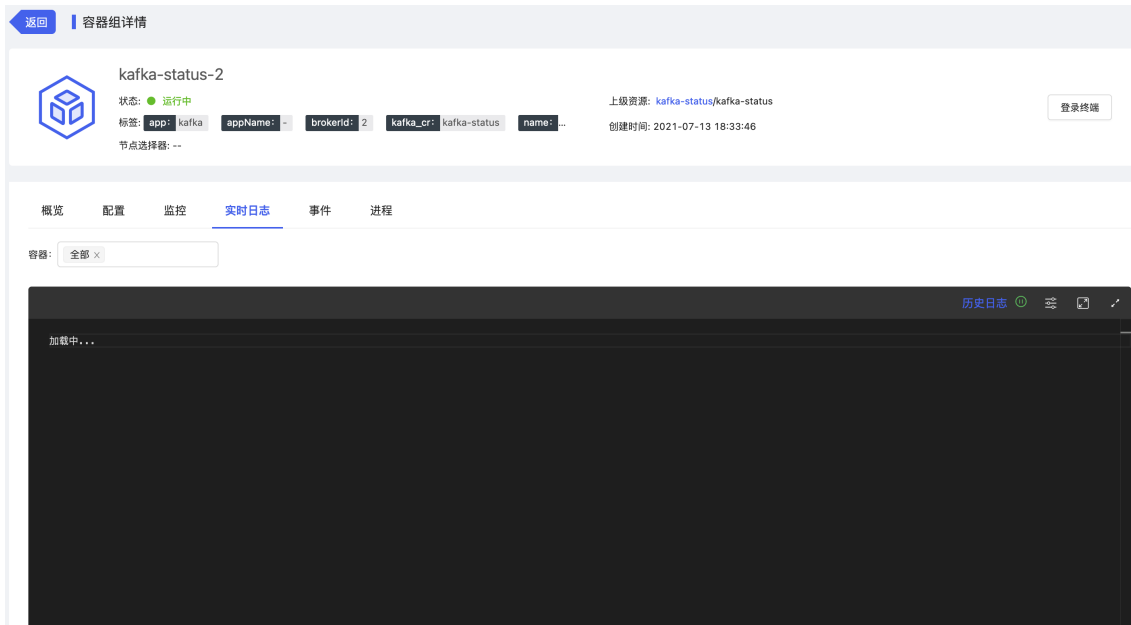
hello kafka
hiii
hello kafka
exit
hello kafka
```

五、应用运维指南；（补充界面部署方式）

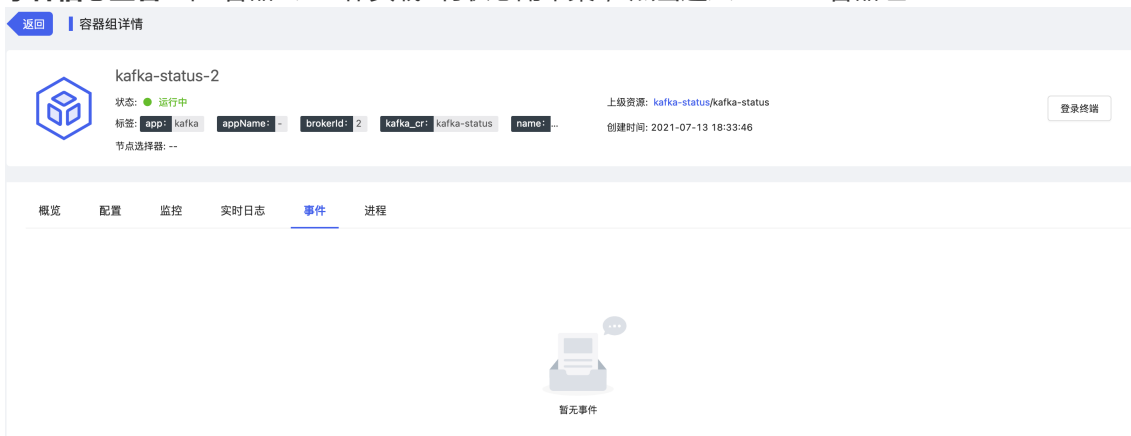
- **监控信息查看：**在“工作负载--容器组”，点击进入 Kafka 容器组



- **日志信息查看：**在“容器云--工作负载--有状态副本集”，点击进入 Kafka 容器组



- **事件信息查看：**在“容器云--工作负载--有状态副本集”，点击进入 Kafka 容器组



- **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”

时透云 管理工作台 云原生应用市场 admin 系统管理

服务全局 - 不要项目选择 容器云 / 工作负载 / ... zs-olm one (one) 选择操作类型 选择状态 开始日期 结束日期 立即查询 刷新

安全 and 运维

- 审计记录
- 审计分析
- 平台数据备份
- 资源报表
- 应用监控

共 0 条 < 1 / 0 >

时间	持续时间	操作类型	对象及类型	项目	集群名	状态	发起者	源 IP
暂无数据								