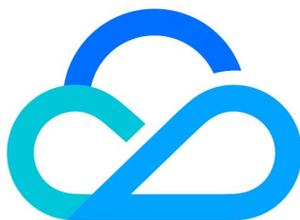


云原生分布式云中心 实践教学



腾讯云

【 版权声明 】

©2013–2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分內容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

实践教程

多集群应用和高可用容灾实践

实践教学

多集群应用和高可用容灾实践

最近更新时间：2024-10-22 15:15:01

概述

本文将通过配置和部署一个典型 Dubbo 项目 Q 云书城（Q Cloud Book Mall，QCBM）为例，详细介绍和展示如何在多云多集群场景下管理一个典型的微服务项目，通过简洁的操作实现多集群应用分发管理，调度和迁移，灰度更新等能力，最终搭建起具备多集群容灾能力的完整微服务业务。

前期准备

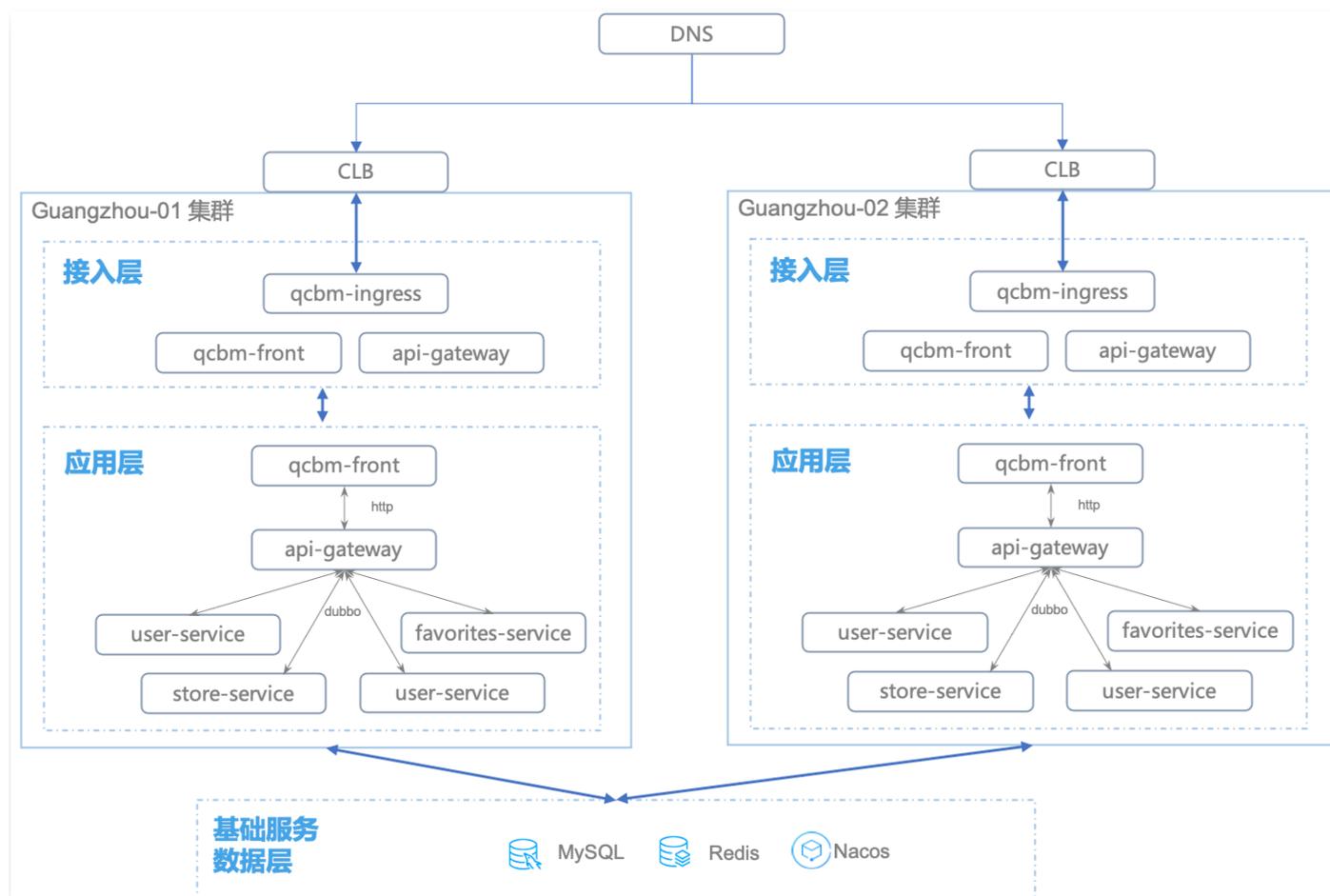
在开始部署服务前，请确认以下步骤已完成：

- [注册腾讯云账号](#)，并完成 [实名认证](#)。
- 阅读 QCBM 项目，了解 Dubbo 微服务架构，详情可参见 [QCBM 项目](#)。
- 掌握容器服务基础知识，详情见 [容器服务 TKE 介绍](#)。

技术架构

QCBM 是采用微服务架构，并使用 dubbo 框架开发的一个网上书城 Demo 项目，详情可参见 [QCBM 项目](#)。本文将 QCBM 扩展至多集群上统一管理，在整体架构上主要划分为接入层，应用层和数据层，配合相应管控和监控组件，实现多集群管理和多活容灾能力。

整体架构图如下：



- 接入层使用腾讯云 CLB 或其他云厂商 LB 产品结合 K8s ingress 和 service 实现流量接入和转发，也支持配置 Nginx 实现。本文以 [腾讯云负载均衡 CLB](#) 为例。
- 数据层使用腾讯云数据库 MYSQL 和 Redis，或使用其他云厂商的数据库产品，同时也支持对接自建的 MySQL 和 Redis。本文以 [腾讯云数据库 MYSQL](#) 和 [腾讯云数据库 Redis](#) 为例说明。
- 应用层由以下微服务应用组成，通过 [应用治理](#) 向多集群快速部署发布，并支持根据集群差异化配置进行灰度更新。
 - QCBM-Front：使用 react 开发的前端项目，基于 Nginx 官方提供的 1.19.8 docker 镜像构建和部署。
 - QCBM-Gateway：API 网关，接受前端的 http 请求，并转化为后台的 dubbo 请求。
 - User-Service：基于 dubbo 的微服务，提供用户注册、登录、鉴权等功能。
 - Favorites-Service：基于 dubbo 的微服务，提供用户图书收藏功能。
 - Order-Service：基于 dubbo 的微服务，提供用户订单生成和查询等功能。
 - Store-Service：基于 dubbo 的微服务，提供图书信息的存储等功能。

操作步骤

基础环境

1. 分别在平台部署所需的基础服务，包括：Mysql、Redis、CLB、Nacos、TSW 等。
2. 准备 QCBM 服务所运行的集群，本文以 TKE 集群为例进行演示，参考 [创建集群](#) 在两个不同地域创建 TKE 集群。

注意

在实际生产环境中，多个集群可位于不同地域、可用区、甚至不同云服务商，实现可控的基础设施故障域。

在 TDCC 注册集群

参考 [创建注册集群](#)，将集群注册至分布式云中心。例如，注册 "guangzhou-01" 和 "guangzhou-02" 集群，完成后查看集群状态正常。

ID/名称	kubernetes版本	类型/状态	节点数
cls-7m0m0c000 guangzhou-02	1.20.6	注册集群(运行中)	1台
cls-7m0m0c000 guangzhou-01	1.20.6	注册集群(运行中)	3台

注意

首次进入分布式云中心控制台，需要开通云原生分布式云中心 TDCC 服务，详情见 [云原生分布式云中心快速入门](#)。

部署多集群应用

参考 [应用管理](#) 创建和管理多集群的应用，快速的将 QCBM 应用部署到多集群上。

创建分发策略

1. 登录 [云原生分布式云中心控制台](#)，进入 [分布式应用管理 > 分发策略](#) 页面。
2. 单击 **新建**，创建分发策略 "qcbm-subscription"，在弹出的页面选择想要发布的目标集群，此处选择 "guangzhou-01" 集群。如下图所示：

名称

最长63个字符，只能包含小写字母、数字及分隔符("-")，且必须以小写字母开头，数字或小写字母结尾

分发集群 (subscribers) 选择已有集群 自定义Label

有以下可用集群 (共4项 已加载 4 项) 已选择 1 项

多个过滤标签用回车键分隔

ID/集群名称	状态
<input type="checkbox"/> cls- guangzhou-02	运行中
<input type="checkbox"/> cls- shanghai-01	运行中
<input type="checkbox"/> cls- dylan-dtcc-nj-minikube	运行中
<input checked="" type="checkbox"/> cls- guangzhou-01	运行中

支持按住shift键进行多选

关联资源 (feeds) 添加

请选择想要分发的K8s资源，分发策略将把上述关联的资源分发至指定的集群。如需新建更多K8s资源[应用管理](#) [创建资源](#)

创建分发策略 取消

3. 单击**创建**完成创建，查看 qcbm-subscription 分发策略创建成功。

创建 Namespace

参考 [QCBM 项目](#)，创建 qcbm 命名空间，根据分发策略发布至指定集群。

⚠ 注意

创建流程与单集群下创建的过程一致，仅需在配置页面最后选择指定的分发策略，例如：qcbm-subscription。

方式 1：使用控制台

1. 登录 [云原生分布式云中心控制台](#)，进入[分布式应用管理](#) > [应用管理](#)页面。

- 单击**命名空间** > **新建**，创建名称为 qcbm 的命名空间 Namespace，分发策略选择 qcbm-subscription。

← 新建Namespace

名称

最长63个字符，只能包含小写字母、数字及分隔符("-")，且必须以小写字母开头，数字或小写字母结尾

描述

分发策略

选择分发策略 不关联分发策略 新建分发策略 已有分发策略

[添加](#)

如您有业务需要新的分发策略，您可以[新建分发策略](#)

- 单击**创建完成创建**，命名空间 qcbm 成功创建并分发至指定集群上。

方式 2：使用 YAML 部署

- 登录 [云原生分布式云中心控制台](#)，进入**分布式应用管理** > **应用管理**页面。
- 单击 **YAML 创建资源**，输入以下 YAML 配置。
- 单击**创建完成创建**，修改分发策略选择 qcbm-subscription。

```
apiVersion: v1
kind: Namespace
metadata:
  name: qcbm
spec:
  finalizers:
  - kubernetes
```

创建 ConfigMap

参考 [QCBM 项目](#)，创建名称为 qcbm-env 的 ConfigMap 用于存放相关配置，根据分发策略发布至指定集群。

注意

创建流程与单集群下创建的过程一致，仅需在配置页面最后选择指定的分发策略，例如：qcbm-subscription。

方式 1：使用控制台

1. 登录 [云原生分布式云中心控制台](#)，进入 [分布式应用管理 > 应用管理](#) 页面。
2. 单击 [配置管理 > ConfigMap > 新建](#)，创建名称为 qcbm-env 的 ConfigMap，分发策略选择 qcbm-subscription。

名称

最长63个字符，只能包含小写字母、数字及分隔符("-")，且必须以小写字母开头，数字或小写字母结尾

命名空间

变量名		变量值	
<input type="text" value="NACOS_HOST"/>	=	<input type="text" value=""/>	✕
<input type="text" value="MYSQL_HOST"/>	=	<input type="text" value=""/>	✕
<input type="text" value="REDIS_HOST"/>	=	<input type="text" value=""/>	✕
<input type="text" value="SW_AGENT_COLLECTOR_BACKEND_SERVICES"/>	=	<input type="text" value=""/>	✕
<input type="text" value="NACOS_PORT"/>	=	<input type="text" value=""/>	✕
<input type="text" value="MYSQL_PORT"/>	=	<input type="text" value=""/>	✕
<input type="text" value="REDIS_PORT"/>	=	<input type="text" value=""/>	✕

变量名为空时，在变量名称中粘贴一行或多行key=value或key: value的键值对可以实现快速批量输入

[添加变量](#)

分发策略

选择分发策略 不关联分发策略 新建分发策略 已有分发策略

↻ ✕

[添加](#)

如您有业务需要新的分发策略，您可以[新建分发策略](#)

创建ConfigMap
取消

3. 单击**创建完成创建**，查看配置成功的创建并分发至指定集群上。

方式 2：使用 YAML 部署

1. 登录 [云原生分布式云中心控制台](#)，进入**分布式应用管理 > 应用管理**页面。
2. 单击 **YAML 创建资源**，输入以下 YAML 配置。
3. 单击**创建完成创建**，修改分发策略选择 `qcbm-subscription`。

```
apiVersion: v1
data:
  MYSQL_HOST: ## 根据实际配置填写
  MYSQL_PORT: ## 根据实际配置填写
  NACOS_HOST: ## 根据实际配置填写
  NACOS_PORT: ## 根据实际配置填写
  REDIS_HOST: ## 根据实际配置填写
  REDIS_PORT: ## 根据实际配置填写
  SW_AGENT_COLLECTOR_BACKEND_SERVICES: ## 根据实际配置填写
kind: ConfigMap
metadata:
  name: qcbm-env
  namespace: qcbm
```

创建 Secret

参考 [QCBM 项目](#)，创建名称为 `qcbm-keys` 的 Secret 用于存放相关配置，根据分发策略发布至指定集群。

⚠ 注意

创建流程与单集群下创建的过程一致，仅需在配置页面最后选择指定的分发策略，例如：`qcbm-subscription`。

方式 1：使用控制台

1. 登录 [云原生分布式云中心控制台](#)，进入**分布式应用管理 > 应用管理**页面。
2. 单击**配置管理 > Secret > 新建**，创建名称为 `qcbm-keys` 的 Secret，分发策略选择 `qcbm-subscription`。
3. 单击**创建完成创建**，查看配置成功的创建并分发至指定集群上。

方式 2：使用 YAML 部署

1. 登录 [云原生分布式云中心控制台](#)，进入**分布式应用管理 > 应用管理**页面。
2. 单击 **YAML 创建资源**，输入以下 YAML 配置。

3. 单击创建完成创建，修改分发策略选择 qcbm-subscription。

```
apiVersion: v1
kind: Secret
metadata:
name: qcbm-keys
namespace: qcbm
labels:
  qcloud-app: qcbm-keys
data:
# echo -n xxx | base64 , 加 -n 去掉 echo 默认加的换行
MYSQL_ACCOUNT: ## 根据实际配置填写
MYSQL_PASSWORD: ## 根据实际配置填写
REDIS_PASSWORD: ## 根据实际配置填写
SW_AGENT_AUTHENTICATION: ## 根据实际配置填写
type: Opaque
```

创建 Deployment

参考 [QCBM 项目](#)，创建 user-service、store-service、qcbm-gateway、qcbm-front、order-service、favorites-service 工作负载，根据分发策略发布至指定集群。

⚠ 注意

创建流程与单集群下创建的过程一致，仅需在配置页面最后选择指定的分发策略，例如：qcbm-subscription。

方式 1：使用控制台

1. 登录 [云原生分布式云中心控制台](#)，进入 [分布式应用管理 > 应用管理](#) 页面。
2. 单击 [工作负载 > Deployment > 新建](#)，创建名称为 user-service、store-service、qcbm-gateway、qcbm-front、order-service、favorites-service deployment 工作负载，分发策略选择 qcbm-

subscription.

工作负载名

最长40个字符，只能包含小写字母、数字及分隔符("-"), 且必须以小写字母开头，数字或小写字母结尾

描述

Label k8s-app = user-service 删除

[新增](#)

标签名称只能包含字母、数字及分隔符("-","_",".","/"), 且必须以字母、数字开头和结尾
标签值只能包含字母、数字及分隔符("-","_","."), 且必须以字母、数字开头和结尾

命名空间 qcbm

类型

- Deployment (可扩展的部署Pod)
- DaemonSet (在每个主机上运行Pod)
- StatefulSet (有状态集的运行Pod)
- CronJob (按照Cron的计划定时运行)
- Job (单次任务)

数据卷 (选填)

使用主机路径	<input type="text" value="dumpath"/>	主机路径设置 重新设置	✕
使用主机路径	<input type="text" value="logpath"/>	主机路径设置 重新设置	✕

[添加数据卷](#)

为容器提供存储，目前支持临时路径、主机路径、云硬盘数据卷、文件存储NFS、配置文件、PVC，还需挂载到容器的指定路径中。[使用指引](#)

实例内容器 user-service + 添加容器

名称	<input type="text" value="user-service"/>			
	<small>最长63个字符，只能包含小写字母、数字及分隔符("-"), 且不能以分隔符开头或结尾</small>			
镜像	<input type="text" value="registry.cn-hangzhou.aliyuncs.com/qcbm/user-service"/>	选择镜像		
镜像版本 (Tag)	<input type="text" value="fd3edd"/>			
环境变量	ConfigMap	<input type="text" value="NACOS_HOST"/>	qcbm-env	NACOS_HOST ✕
	ConfigMap	<input type="text" value="MYSQL_HOST"/>	qcbm-env	MYSQL_HOST ✕
	ConfigMap	<input type="text" value="MYSQL_PORT"/>	qcbm-env	MYSQL_PORT ✕
	ConfigMap	<input type="text" value="REDIS_HOST"/>	qcbm-env	REDIS_HOST ✕

创建工作负载
取消

3. 单击**创建**完成创建，查看配置成功的创建并分发至指定集群上。

方式 2: 使用 YAML 部署

1. 登录 [云原生分布式云中心控制台](#)，进入**分布式应用管理 > 应用管理**页面。
2. 单击 **YAML 创建资源**，输入以下 YAML 配置。

3. 单击创建完成创建，修改分发策略选择 qcbm-subscription。

```
apiVersion: apps/v1
kind: Deployment
metadata:
name: user-service
namespace: qcbm
labels:
  app: user-service
  version: v1
spec:
replicas: 1
selector:
  matchLabels:
    app: user-service
    version: v1
template:
  metadata:
    labels:
      app: user-service
      version: v1
  spec:
    containers:
      - name: user-service
        image: ccr.ccs.tencentyun.com/qcbm/user-service:fdd3edd
        imagePullPolicy: Always
        env:
          - name: NACOS_HOST # dubbo 服务注册中心 nacos 的 IP 地址
            valueFrom:
              configMapKeyRef:
                key: NACOS_HOST
                name: qcbm-env
                optional: false
          - name: MYSQL_HOST # Mysql 地址
            valueFrom:
              configMapKeyRef:
                key: MYSQL_HOST
                name: qcbm-env
                optional: false
          - name: MYSQL_PORT
            valueFrom:
              configMapKeyRef:
                key: MYSQL_PORT
                name: qcbm-env
                optional: false
          - name: REDIS_HOST # Redis 的 IP 地址
```

```
valueFrom:
  configMapKeyRef:
    key: REDIS_HOST
    name: qcbm-env
    optional: false
- name: REDIS_PORT
valueFrom:
  configMapKeyRef:
    key: REDIS_PORT
    name: qcbm-env
    optional: false
- name: MYSQL_ACCOUNT
valueFrom:
  secretKeyRef:
    key: MYSQL_ACCOUNT
    name: qcbm-keys
    optional: false
- name: MYSQL_PASSWORD
valueFrom:
  secretKeyRef:
    key: MYSQL_PASSWORD
    name: qcbm-keys
    optional: false
- name: REDIS_PASSWORD
valueFrom:
  secretKeyRef:
    key: REDIS_PASSWORD
    name: qcbm-keys
    optional: false
- name: SW_AGENT_COLLECTOR_BACKEND_SERVICES
valueFrom:
  configMapKeyRef:
    key: SW_AGENT_COLLECTOR_BACKEND_SERVICES
    name: qcbm-env
    optional: false
- name: SW_AGENT_AUTHENTICATION
valueFrom:
  secretKeyRef:
    key: SW_AGENT_AUTHENTICATION
    name: qcbm-keys
    optional: false
ports:
- containerPort: 20880 # dubbo 端口号
  protocol: TCP
volumeMounts:
- name: dumpath
  mountPath: /app/dumps
```

```
- name: logpath
  mountPath: /app/logs
volumes:
- name: dumpath
  hostPath:
    path: /data/dumps/user-service
    type: DirectoryOrCreate
- name: logpath
  hostPath:
    path: /data/logs/user-service
    type: DirectoryOrCreate
```

创建 Service

参考 [QCBM 项目](#)，创建名称为 qcbm-front 和 api-gateway 的 Service，根据分发策略发布至指定集群。

⚠ 注意

创建流程与单集群下创建的过程一致，仅需在配置页面最后选择指定的分发策略，例如：qcbm-subscription。

方式 1：使用控制台

1. 登录 [云原生分布式云中心控制台](#)，进入 [分布式应用管理 > 应用管理](#) 页面。
2. 单击 [服务与路由 > Service > 新建](#)，创建名称为 qcbm-front 和 api-gateway 的 Service，分发策略选择 qcbm-subscription。
3. 单击 [创建完成创建](#)，查看配置成功的创建并分发至指定集群上。

方式 2：使用 YAML 部署

1. 登录 [云原生分布式云中心控制台](#)，进入 [分布式应用管理 > 应用管理](#) 页面。
2. 单击 [YAML 创建资源](#)，输入以下 YAML 配置。
3. 单击 [创建完成创建](#)，修改分发策略选择 qcbm-subscription。

```
apiVersion: v1
kind: Service
metadata:
  name: qcbm-front
  namespace: qcbm
spec:
  type: NodePort
  selector:
    app: qcbm-front
```

```
version: v1
ports:
  - name: http
    port: 80
    targetPort: 80
    nodePort: 30080
---
apiVersion: v1
kind: Service
metadata:
  name: api-gateway
  labels:
    app: api-gateway
  namespace: qcbm
  annotations:
    service.kubernetes.io/qcloud-loadbalancer-internal-subnetid: ## 根据实际配置填写
spec:
  externalTrafficPolicy: Cluster
  ports:
    - name: http
      port: 8080
      targetPort: 8080
      protocol: TCP
      nodePort: 32500
  selector: # 将后端服务 qcbm-gateway 和该 Service 进行映射
    app: qcbm-gateway
    version: v1
  type: LoadBalancer
```

创建 Ingress

参考 [QCBM 项目](#)，创建名称为 qcbm-ingress 的 Ingress，根据分发策略发布至指定集群。

⚠ 注意

创建流程与单集群下创建的过程一致，仅需在配置页面最后选择指定的分发策略，例如：qcbm-subscription。

方式 1：使用控制台

1. 登录 [云原生分布式云中心控制台](#)，进入 [分布式应用管理 > 应用管理](#) 页面。

2. 单击**服务与路由** > **Ingress** > **新建**，创建名称为 qcbm-ingress 的 Ingress，分发策略选择 qcbm-subscription。
3. 单击**创建完成创建**，查看配置成功的创建并分发至指定集群上。

方式 2：使用 YAML 部署

1. 登录 [云原生分布式云中心控制台](#)，进入**分布式应用管理** > **应用管理**页面。
2. 单击**YAML 创建资源**，输入以下 YAML 配置。
3. 单击**创建完成创建**，修改分发策略选择 qcbm-subscription。

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: qcbm-ingress
  namespace: qcbm
  annotations:
    ingress.cloud.tencent.com/direct-access: "false"
    kubernetes.io/ingress.class: qcloud
    kubernetes.io/ingress.extensiveParameters:
      '{"AddressIPVersion":"IPV4"}'
    kubernetes.io/ingress.http-rules:
      '[{"host":"qcbm.com","path":"/","backend":{"serviceName":"qcbm-
front","servicePort":"80"}}]'
  spec:
    rules:
      - host: qcbm.com
        http:
          paths:
            - path: /
              backend:
                serviceName: qcbm-front
                servicePort: 80
```

查看部署结果

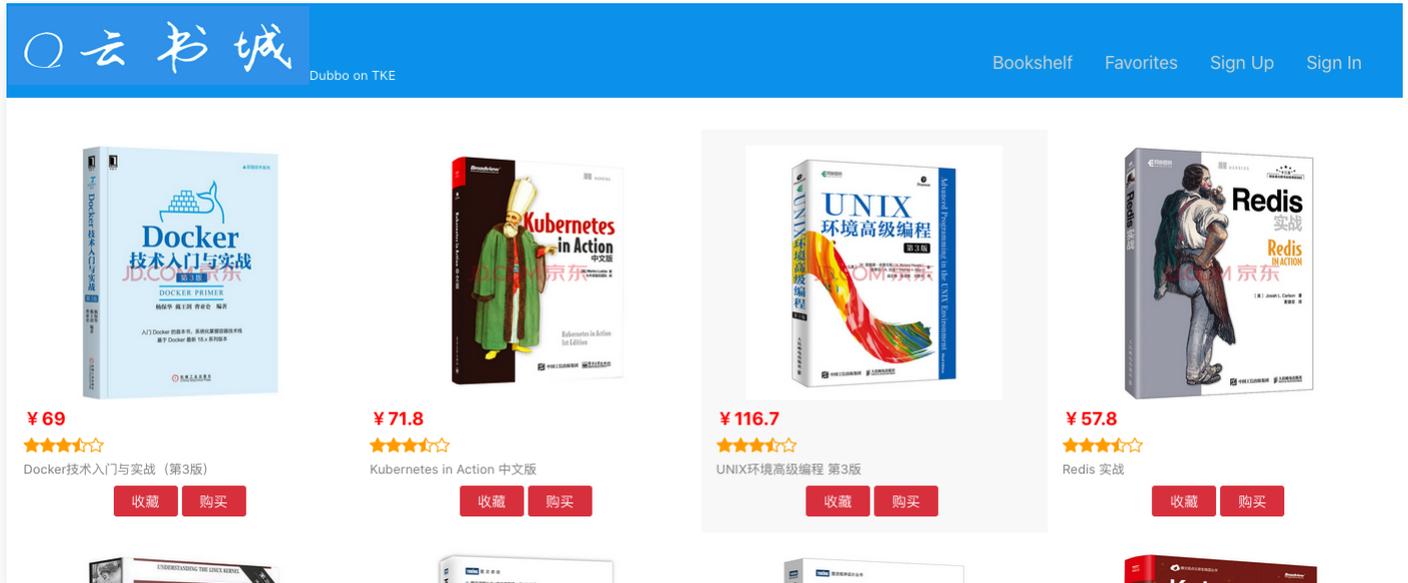
至此，您已完成 QCBM 在 "guangzhou-01" 集群上的部署，可通过以下步骤查看部署结果：

1. 登录 [云原生分布式云中心控制台](#)，进入**分布式应用管理** > **应用管理**页面。

2. 单击服务与路由 > Ingress 进入 Ingress 页面，可查看到创建的 qcbm-ingress。单击进入详情 > 实例管理。

名称	类型 T	VIP	后端服务	创建时间	操作
qcbm-ingress 后	lb- 公网LB	(IPV4)后	http://qcbm.com/-->qcbm-front:80	2021-11-24 19:00:27	更新转发配置 编辑YAML 删除

3. 通过 Ingress 的 VIP 即可访问 Q 云书城页面。



多集群调度管理

新增集群发布

根据业务发展和容灾保障的需求，需要将 QCBM 应用部署至另外一个集群，可以通过编辑分发策略快速实现。

1. 登录 [云原生分布式云中心控制台](#)，进入分布式应用管理 > 分发策略页面。
2. 选择分发策略 qcbm-subscription，单击关联集群，添加 "guangzhou-02" 集群。
3. 单击确定，分布式云中心将自动将所有 QCBM 应用分发至两个集群上。单击名称进入详情页面可查看详情信息和拓扑图。

分发策略/Subscription/(qcbm-subscription)

详情 实例列表 YAML

基本信息

名称: qcbm-subscription

subscribers: clusters.clusternet.io/cluster-name In ["cls-...", "cls-..."]

关联资源:

- 资源类型: Namespace/资源名称: qcbm
- 资源类型: ConfigMap/命名空间: qcbm/资源名称: qcbm-env
- 资源类型: Secret/命名空间: qcbm/资源名称: qcbm-keys
- 资源类型: Deployment/命名空间: qcbm/资源名称: user-service
- 资源类型: Deployment/命名空间: qcbm/资源名称: store-service
- 资源类型: Deployment/命名空间: qcbm/资源名称: qcbm-gateway
- 资源类型: Deployment/命名空间: qcbm/资源名称: qcbm-front
- 资源类型: Deployment/命名空间: qcbm/资源名称: order-service
- 资源类型: Deployment/命名空间: qcbm/资源名称: favorites-service
- 资源类型: Service/命名空间: qcbm/资源名称: qcbm-front
- 资源类型: Ingress/命名空间: qcbm/资源名称: qcbm-ingress
- 资源类型: Service/命名空间: qcbm/资源名称: api-gateway

创建时间: 2021-11-24 14:14:57

拓扑图

差异化配置集群下实例

QCBM 应用已部署在两个集群上，由于不同集群间环境的差异，QCBM 应用需要在不同集群上有差异化的配置，可以通过差异化策略实现。

例如 `api-gateway` Service 需要为 "guangzhou-02" 集群指定 `subnet-id`，可通过以下步骤实现：

1. 登录 [云原生分布式云中心控制台](#)，进入 [分布式应用管理 > 应用管理](#) 页面。
2. 进入 [服务与路由 > Service](#)，单击 `api-gateway` 名称进入 [详情 > 实例管理](#) 页面，选择 "guangzhou-02" 集群下的实例，新建差异化策略。

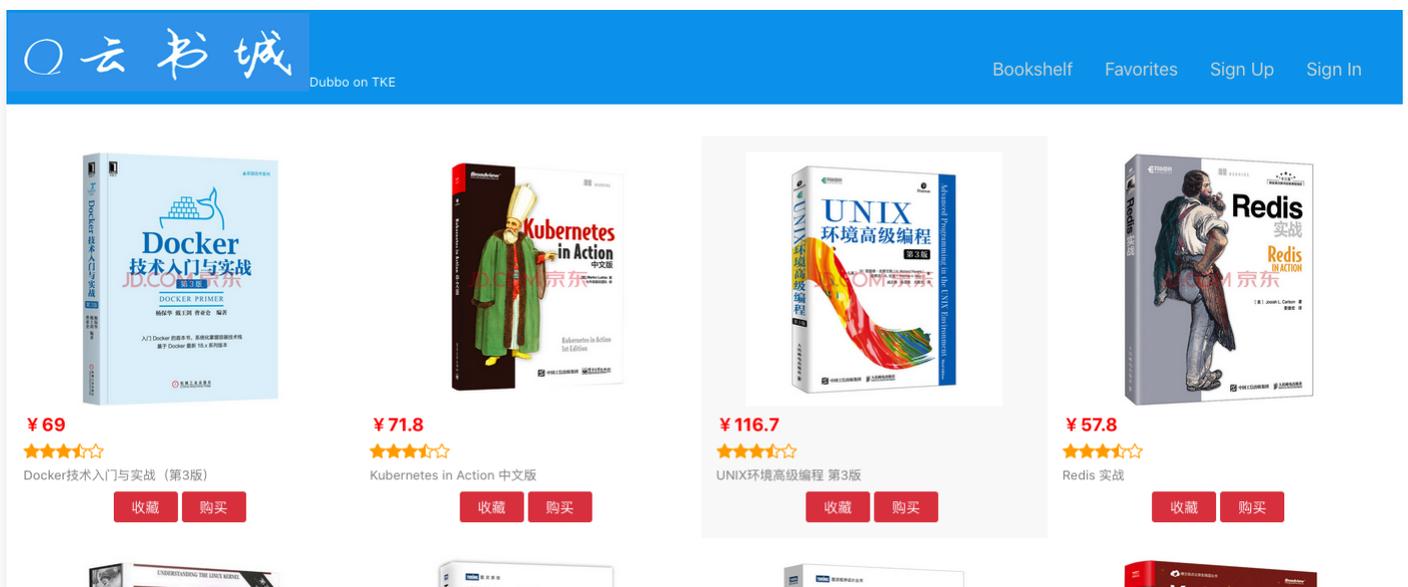
```

apiVersion: apps.clusternet.io/v1alpha1
kind: Localization
metadata:
name: qcbm-svc-gateway-override
namespace: ## 根据实际配置填写
spec:
priority: 300
feed:
  apiVersion: v1
  kind: Service
  name: api-gateway
  namespace: qcbm
    
```

```
overrides:
- name: svc-override
  type: JSONPatch
  value: |-
    [
      {
        "op": "replace",
        "path": "/metadata/annotations/service.kubernetes.io~1qcloud-
loadbalancer-internal-subnetid",
        "value": "" ## 根据实际配置填写
      }
    ]
```

3. 单击确定保存，分布式云中心将根据该差异化策略自动调整对应集群下的配置。

4. 进入 "guangzhou-02" 集群的 Ingress 页面，找到 qcbm-ingress 地址，通过该地址即可访问 "guangzhou-02" 集群下的 Q 云书城页面。



多活容灾能力验证

按照以上云原生分布式云中心 TDCC 多集群应用管理的操作，QCBM 业务应用分别部署在 "guangzhou-01" 和 "guangzhou-02" 集群上，对外暴露 ingress 地址提供 Q 云书城的服务。

此时可以使用已有或 [新申请域名](#)，添加 ingress 地址解析，实现基本的多活容灾能力。例如参考以下步骤验证多集群高可用：

1. 配置本地 hosts 域名 "tke-demo.cn"，默认访问 "guangzhou-01" 集群的 ingress 地址，页面能够正常访问，工作正常。
2. 手动驱逐该集群的节点或停止应用，模拟环境故障，此时访问 "tke-demo.cn" 返回错误。
3. 通过修改 hosts 配置来模拟 DNS 切换。

```
# /etc/hosts
#1.14.x.x tke-demo.cn # guangzhou-01 ingress address
```

```
129.226.x.x tke-demo.cn # guangzhou-02 ingress address
```

4. 重新打开浏览器访问 "tke-demo.cn", 页面能够正常访问, 工作正常。

按照以上步骤可以验证, 通过 TDCC 分布式云中心部署的多活容灾 QCBM 业务, 实现了多集群高可用容灾, 保证了业务对灾难级别事件的保护的能力。更进一步按照同样的流程, 可以将 QCBM 业务部署至本地集群、第三方云服务商集群、边缘集群上, 实现跨云服务和容灾的能力。