

容器服务

应用

产品文档



腾讯云

【版权声明】

©2013-2018 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【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。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。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模式的承诺或保证。

文档目录

应用

应用管理

应用管理概述

应用管理实践---不同环境下Wordpress应用部署

应用

应用概述

应用示例-nginx单服务应用

应用示例-guestbook应用

创建应用

删除应用

更新应用

查看应用

管理应用内服务

应用模板

应用模板概述

应用模板示例-nginx单服务应用

应用模板示例-guestbook应用

创建应用模板

删除应用模板

更新应用模板

查看应用模板

语法说明 - 基础语法

语法说明 - 扩展语法

语法说明 - 变量设置

语法说明 - 限制说明

应用模板内容操作指引

应用

应用管理

应用管理概述

最近更新时间：2018-04-18 17:14:22

随着微服务和Devops的兴起，需要在多个环境中对多个服务进行部署和管理。这大大增加了部署操作的复杂度。腾讯云容器服务支持通过应用的方式对服务进行分组管理，极大的简化了服务管理的复杂度。同时，通过应用模板来保存服务的部署信息，通过配置来管理服务在不同环境下的差异化信息，实现在不同环境下通过应用模板+配置快速的部署服务。

应用管理主要概念

应用管理包括三个主要的概念：

- **应用**：通过服务组的方式对多个服务进行统一管理。更多关于应用的说明可以参考[应用概述](#)
- **应用模板**：通过YAML形式的文本来描述服务的部署信息。方便用户将相同的应用部署到不同环境。更多关于应用模板的说明可以参考[应用模板概述](#)
- **配置管理**：在配置管理中可以将不同环境中差异化的部分通过配置项来区分，将应用中常变的值以变量的形式替代，方便修改和更新。更多关于应用中配置管理的内容可以参考[应用内使用配置项](#)

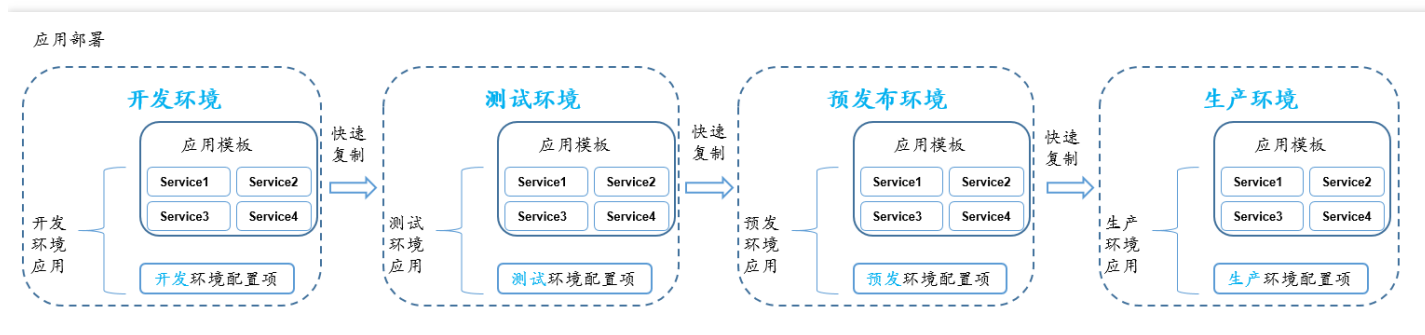
更多应用管理主要概念的说明可以参考[使用指引页面](#)

应用管理的使用场景说明

应用管理的使用场景主要包括两个：一个是多环境下应用的快速部署，另一个是基于服务组的Devops流程管理。

多环境下的应用快速部署

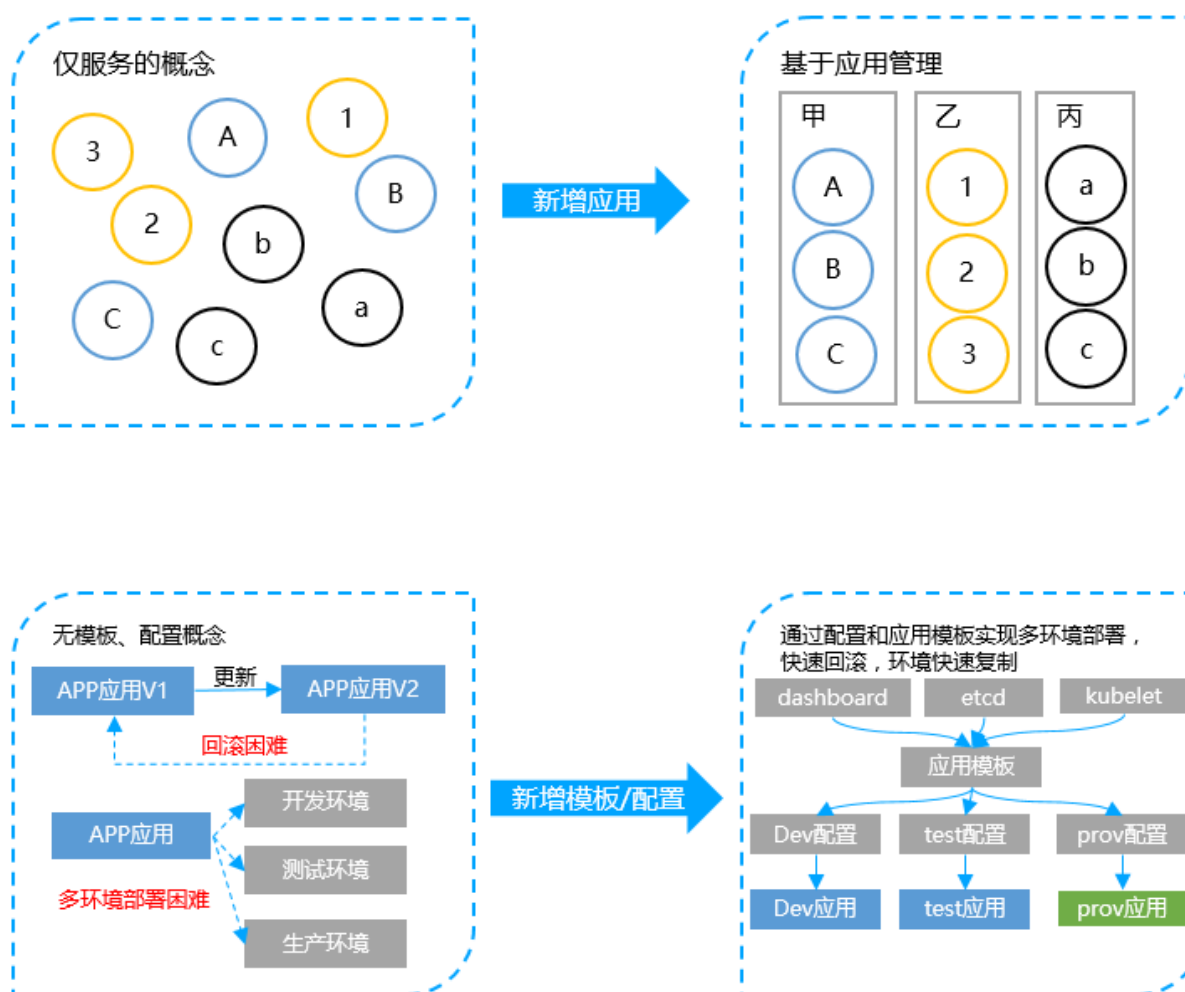
多环境下应用的快速部署，如下图所示，通过应用模板+不同环境下的配置，可以快速的在不同的环境下部署应用。在应用模板中描述多个服务部署的信息，在配置项中设置不同环境下差异化的选项，例如：实例副本数，数据库地址等。使用应用模板+配置实现快速的在多个环境中部署应用实例。更多关于多环境下部署应用可以参考[不同环境下Wordpress应用部署](#)。



基于服务组的Devops流程管理

在Devops的流程中，应用管理基于应用的概念对服务进行分组。基于服务组对关联性比较强的服务进行统一管理，降低管理多个服务的复杂度。

同时基于应用模板+配置实现在不同环境下管理的同步。修改服务的部署信息，可以通过应用模板来进行同步，从而实现多环境下信息的同步管理。



应用管理使用示例

我们用一个简单的应用 `nginxapp` 来进行说明，为了说明应用管理的使用方式。该应用下只有一个 `nginx` 服务。可以通过修改配置项中 `NAMESPACE` 变量的值，将应用快速的部署到不同的命名空间中。详细的内容可以参考示例 [Nginx应用](#)。

Guestbook是一个典型的web应用，包含前端fronted服务和后端的redis-master服务以及redis-slave服务。这些服务可以定义在一个应用模板内，通过描述性语言对服务的部署情况进行说明。将不同环境下差异的信息定义在配置文件中，从而通过应用模板和配置可以在不同环境中实现部署应用，并通过应用对这几个服务做统一的管理。详细的内容可以参考示例[Guestbook应用](#)。

应用管理实践---不同环境下Wordpress应用部署

最近更新时间：2018-09-13 09:40:34

在 Devops 流程中，往往包含多个环境，例如包含：开发(Dev)，测试(Test/beta)，预发布（pre-product），生产环境(product)。在不同环境中都需要对应用进行部署。

WordPress 是一个内容管理平台，是世界上建立博客和网站最流行的网络发布平台之一，可以让用户轻松地发布，管理和组织各种内容到目标的网站上。在本示例中我们将通过 wordpress 部署的例子，介绍如何通过 Namespace 命名空间对不同环境进行隔离，通过应用模板+配置实现不同环境下应用的部署。

步骤一：创建 wordpress 应用模板

在 Wordpress 中包含一个前端的 wordpress 服务和一个后端的 mariadb 服务(mariadb 是一个和 mysql 类似的数据库服务，可以参考[mariadb 介绍](#))。wordpress 应用模板创建过程包括以下几个步骤：

1. 新建 wordpress 应用模板
2. 导入 mariadb 服务
3. 导入 wordpress 服务
4. 参数转换为配置项

新建 wordpress 应用模板

在[应用模板列表](#)页面，单击 新建 按钮，新建一个应用模板。

腾讯云

总览

云产品

常用服务

容器服务

概览

应用中心

集群

应用

服务

Ingress

交付中心

镜像仓库

应用模板仓库

我的模板

模板市场

应用模板

+ 新建

模板ID/模板名称	服务数量
tp-el5uirgo singlenginx	1
tp-fdbnzixs nginx	1
tp-eh6ia6js php-nginx	1
tp-mev3xoy2 elk-devin	4
tp-51uxeyx8 d2048	1
tp-6aful3k8	1

在应用模板中，填写应用模板的名称 `wordpress`。

模板名称

wordpress

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

模板内容

模板可以通过从UI导入服务或新增空服务并手动编写来创建多个服务的YAML描述，详情可查看 [应用模板操作指引](#)

服务名	操作	内容
记录为空		

新增空服务

从UI导入服务

导入 mariadb 服务

在腾讯云容器服务中，支持两种方式导入服务的模板内容：(1) 通过控制台导入 (2) 通过 YAML 文件导入。两种方式可以按照需要任意选择其中一种。更多关于服务导入方式的说明可以参考[应用模板内容操作指引](#)。

导入方法 1：通过控制台之间导入

单击 导入服务 按钮，在控制台填写服务对应参数。

设置服务的基本信息：

1. 填写服务名称 mariadb
2. 填写服务描述 mariadb服务

设置服务的数据卷信息：

1. 数据卷选择使用云硬盘，云盘名称填写 vol

设置镜像参数：

1. 在设置容器运行参数中的镜像参数：
 - 容器名称设置为 mariadb
 - 镜像名称设置为 mariadb
 - 版本号选择为 latest

设置容器其他运行参数：

1. 设置容器环境变量：
 - MYSQL_ROOT_PASSWORD：root

设置数据卷的挂载点：

1. vol数据卷挂载点设置为：`/var/lib/mysql`
(更多关于数据挂载的说明，可以参考[数据卷概述](#))

设置服务的实例数：

1. 服务的实例数设置为 1

设置服务的访问方式：

1. 服务的访问方式设置为集群内访问
2. 服务的访问端口：容器端口和服务端口都设置成 3306
(更多关于服务访问方式的说明，可以参考[服务访问方式设置](#))

单击 确认 按钮，自动生成 YAML 形式的模板内容。

导入方法 2：通过 YAML 文件导入

在应用模板页面，单击 + 按钮，新增一个服务。服务名称设置为 mariadb。

模板名称

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

模板内容

模板可以通过从UI导入服务或新增空服务并手动编写来创建多个服务的YAML描述，详情可查看 [应用模板操作指引](#)

服务名	操作	内容
mariadb	保存 取消	

新增空服务

从UI导入服务

在模板内容编辑区域，将下面的 YAML 文本内容直接导入：

apiVersion: extensions/v1beta1

kind: Deployment

metadata:

annotations:

description: mariadb服务

creationTimestamp: null

name: mariadb

namespace: '{{.NAMESPACE}}'

spec:

replicas: 1

revisionHistoryLimit: 5

selector: {}

strategy: {}

template:

metadata:

creationTimestamp: null

spec:

containers:

- env:

- name: MYSQL_ROOT_PASSWORD

value: '{{.ROOT_PASSWORD_VALUE}}'

image: mariadb:latest

imagePullPolicy: Always

name: mariadb

resources:

requests:

cpu: 200m

securityContext:

privileged: false

volumeMounts:

- mountPath: /var/lib/mysql

name: vol

serviceAccountName: ""

volumes:

- name: vol

qcloudCbs:

cbsDiskId: '{{.ReleaseCBS_mariadb_vol}}'

fsType: ext4

status: {}

apiVersion: v1

kind: Service

metadata:

creationTimestamp: null

name: mariadb

```
namespace: '{{.NAMESPACE}}'  
spec:  
ports:  
- name: tcp-3306-3306-nh6kj  
nodePort: 0  
port: 3306  
protocol: TCP  
targetPort: 3306  
selector: {}  
type: ClusterIP  
status:  
loadBalancer: {}
```

这里自动提取了 `NAMESPACE` 和 `ReleaseCBS_mariadb_vol` 作为配置项。并填写 `NAMESPACE` 配置项的值为 `default`。`NAMESPACE` 用来表示服务部署到集群的哪个命名空间，更多关于命名空间的说明可以参考 [Namespace使用指引](#)。`ReleaseCBS_XXXX` 为容器服务为使用 CBS 云盘定义的变量，更多关于 `ReleaseCBS` 自定义变量的说明可以参考 [自定义变量--ReleaseCBS](#)。

模板内容

模板可以通过从UI导入服务或新增空服务并手动编写来创建多个服务的YAML描述，详情可查看 [应用模板操作指引](#)

服务名	操作	内容
mariadb	删除	<pre>37 cbsDiskId: '{{.ReleaseCBS_mariadb_vol}}' 38 fsType: ext4 39 status: {} 40 --- 41 apiVersion: v1 42 kind: Service 43 metadata: 44 creationTimestamp: null 45 name: mariadb 46 namespace: '{{.NAMESPACE}}' 47 spec: 48 ports: 49 - name: tcp-3306-3306-nh6kj 50 nodePort: 0 51 port: 3306 52 protocol: TCP 53 targetPort: 3306 54 selector: {} 55 type: ClusterIP</pre>

[新增空服务](#) [从UI导入服务](#)

默认配置项

以上服务配置中，若存在自定义变量，可以手动输入为变量赋值，也可使用已有的配置项来为变量赋值，更多参见 [变量设置说明](#)

注：变量替换类型的配置项用于字符串替换，文件挂载类型的配置项用于将指定内容挂载到容器

变量名	变量值	类型
NAMESPACE	default	变量替换
ROOT_PASSWORD_VALUE		变量替换
ReleaseCBS_mariadb_vol		变量替换

[使用已有配置项赋值](#)

导入 wordpress 服务

导入方法 1：通过控制台之间导入

单击 [导入服务](#) 按钮，在控制台填写服务对应参数

设置服务的基本信息：

- 1. 填写服务名称 `wordpress`
- 2. 填写服务描述 `wordpress服务`

设置服务的数据卷信息：

- 1. 数据卷选择使用云硬盘，云盘名称填写 `wordpress-persistent-storage`

设置镜像参数：

1. 在设置容器运行参数中的镜像参数：

容器名称设置为 `wordpress`

镜像名称设置为 `wordpress`

版本号选择为 `latest`

设置容器资源限制：

1. 设置容器的 CPU 分配资源为 0.1 核，限制最大的使用资源为 0.2 核

设置容器其他运行参数：

1. 设置容器环境变量：

`WORDPRESS_DB_HOST`： `mariadb`

`WORDPRESS_DB_PASSWORD`： `root`

设置数据卷的挂载点：

1. `vol` 数据卷挂载点设置为：`/var/www/html`

(更多关于数据挂载的说明，可以参考[数据卷概述](#))

设置服务的实例数：

1. 服务的实例数设置为 1

设置服务的访问方式：

1. 服务的访问方式设置为外网访问

2. 服务的访问端口：容器端口和服务端口都设置成 80

(更多关于服务访问方式的说明，可以参考[服务访问方式设置](#))

单击 `确认` 按钮，自动生成 YAML 形式的模板内容。

导入方法2：通过 YAML 文件导入

在应用模板页面，单击 `+` 按钮，新增一个服务。服务名称设置为 `wordpress`。

在模板内容编辑区域，将下面的 YAML 文本内容直接导入：

```
apiVersion: extensions/v1beta1
```

```
kind: Deployment
```

```
metadata:
```

```
  annotations:
```

```
  description: wordpress服务
```

```
  creationTimestamp: null
```

```
name: wordpress
namespace: '{{.NAMESPACE}}'
spec:
replicas: 1
revisionHistoryLimit: 5
selector: {}
strategy: {}
template:
  metadata:
    creationTimestamp: null
  spec:
    containers:
      - env:
          - name: WORDPRESS_DB_HOST
            value: mariadb
          - name: WORDPRESS_DB_PASSWORD
            value: root
        image: wordpress:latest
        imagePullPolicy: Always
        name: wordpress
      resources:
        limits:
          cpu: 200m
        requests:
          cpu: 100m
        securityContext:
          privileged: false
        volumeMounts:
          - mountPath: /var/www/html
            name: wordpress-persistent-storage
        serviceAccountName: ""
        volumes:
          - name: wordpress-persistent-storage
            qcloudCbs:
              cbsDiskId: '{{.ReleaseCBS_wordpress_wordpress_persistent_storage}}'
              fsType: ext4
    status: {}
---
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: null
  name: wordpress
  namespace: '{{.NAMESPACE}}'
spec:
```

```
ports:
- name: tcp-80-80-snfig
nodePort: 0
port: 80
protocol: TCP
targetPort: 80
selector: {}
type: LoadBalancer
status:
loadBalancer: {}
```

这里自动提取 ReleaseCBS_wordpress_wordpress_persistent_storage 作为配置项。ReleaseCBS_XXXX 为容器服务为使用 CBS 云盘定义的变量，更多关于ReleaseCBS 自定义变量的说明可以参考[自定义变量--ReleaseCBS](#)。

参数转换为配置项

在不同环境中部署，可能会存在不同环境下参数一致的情况。这里在模板内容区域将 CPU 使用资源设置为变量，在不同环境下设置成不同的值。

模板内容

模板可以通过从UI导入服务或新增空服务并手动编写来创建多个服务的YAML描述，详情可查看 [应用模板操作指引](#)

服务名	操作	内容
mariadb	删除	19 - env: 20 - name: WORDPRESS_DB_HOST 21 value: mariadb
wordpress	删除	22 - name: WORDPRESS_DB_PASSWORD 23 value: root 24 image: wordpress:latest 25 imagePullPolicy: Always 26 name: wordpress 27 resources: 28 limits: 29 cpu: {{.CPU_LIMITS}} 30 requests: 31 cpu: {{.CPU_REQUEST}} 32 securityContext: 33 privileged: false 34 volumeMounts: 35 - mountPath: /var/www/html 36 name: wordpress-persistent-storage 37 serviceAccountName: ""

[新增空服务](#) [从UI导入服务](#)

默认配置项

以上服务配置中，若存在自定义变量，可以手动输入为变量赋值，也可使用已有的配置项来为变量赋值，更多参见 [变量设置说明](#)

注：变量替换类型的配置项用于字符串替换，文件挂载类型的配置项用于将指定内容挂载到容器

变量名	变量值	类型
NAMESPACE	default	变量替换
ROOT_PASSWORD_VALUE		变量替换
ReleaseCBS_mariadb_vol		变量替换
CPU_LIMITS	200m	变量替换
CPU_REQUEST	100m	变量替换

在这个示例中我们将 CPU 分配资源量和最大限制使用资源量设置为变量。分别用变量 CPU_LIMITS 和 CPU_REQUEST 表示。变量的形式符合 {{.}} 这样的形式。更多关于模板中变量的使用可以参考 [变量设置](#)。

这里自动提取变量 CPU_LIMITS 和 CPU_REQUEST ，设置 CPU_LIMITS 的默认值为 200 m, CPU_REQUEST 为 100 m。(1 m=0.001 核)。

单击 完成后，保存模板信息。在[模板列表页](#)可以看到新创建的模板。

容器服务

概览

应用中心

集群

应用

服务

Ingress

交付中心

镜像仓库

应用模板仓库

我的模板

模板市场

应用模板

+ 新建

模板ID/模板名称	服务数量	创建时间
tp-7xq0950i wordpress	2	2018-02-06 12:18:12
tp-el5uirgo singlenginx	1	2018-01-31 11:22:29
tp-fdbnzixs nginx	1	2018-01-31 11:14:15
tp-eh6ia6js php-nginx	1	2017-12-23 22:00:21
tp-mev3xoy2 elk-devin	4	2017-12-20 20:49:54
tp-51uxeyx8 d2048	1	2017-12-20 16:44:15

步骤二：创建不同环境的命名空间

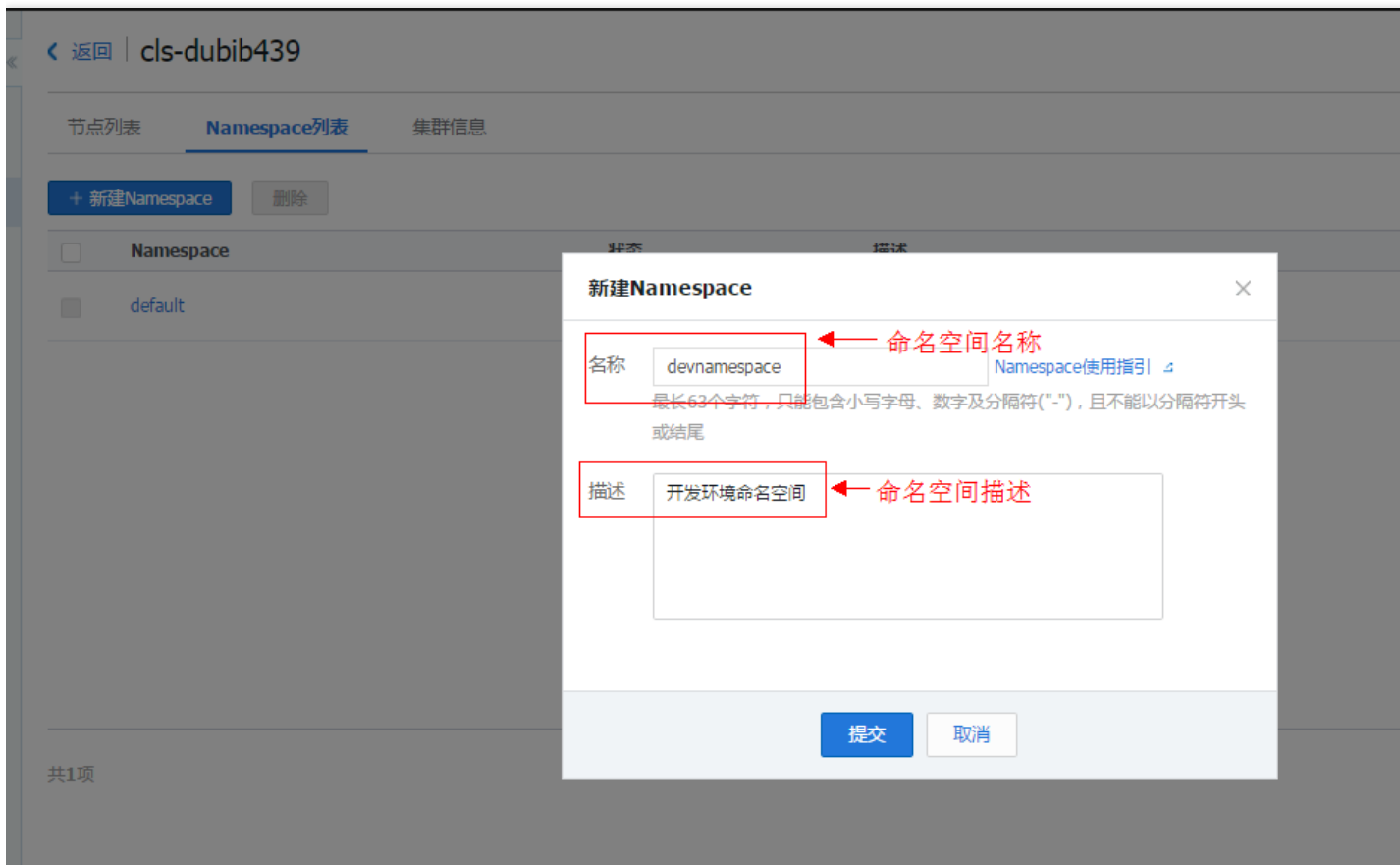
命名空间（Namespace）是对一组资源和对象的抽象集合，可以将不同环境的服务实例部署在不同命名空间中。

如果我们已经有了一个集群，可以直接在集群上开始创建命名空间。如果我们还没有创建好的集群，可以参考[集群基本操作](#)，创建一个集群。

创建集群 Namespace

1. 在集群列表页中选择某集群的 ID/名称。
2. 单击 Namespace 列表，单击【新建 Namespace】。

更多关于命名空间的操作，可以参考[Namespace 使用指引](#)。



在本示例中，我们依次创建 devnamespace、testnamespace、prenamespace、prodnamespace。



步骤三：创建不同环境的 CBS 盘

由于云硬盘页面，选择对应的区域单击新建按钮，创建新的云硬盘。

云硬盘 全部项目 ▾ 广州 深圳金融 上海 上海金融 北京 成都 香港 新加坡 多伦多 广州Open 硅谷 法兰克福 首尔

+ 新建 挂载 卸载 点击新建，创建新的CBS云硬盘

ID/名称	监控	状态 ▾	容量	可用区	可挂载/卸载 ▾	磁盘类型 ▾	磁盘属性 ▾	已挂载云主机	计费模式 ▾
disk-itsytwl 未命名		已挂载	8G	上海一区	不支持	云硬盘	系统盘	ins-m9pdl1h	包年包月
disk-ba8ndq1b 未命名		已挂载	10G	上海一区	不支持	云硬盘	数据盘	ins-m9pdl1h	包年包月
disk-30uk9k7d 未命名		已挂载	10G	上海一区	支持	云硬盘	数据盘	ins-m9pdl1h	包年包月
disk-m4hrs4h 未命名		已挂载	20G	上海一区	不支持	云硬盘	系统盘	ins-a7zse3qf	包年包月
disk-ccj4a8lh 未命名		已挂载	10G	上海一区	不支持	云硬盘	数据盘	ins-a7zse3qf	包年包月

填写对应的参数：

云服务器 云硬盘 全部项目 ▾ 广州 深圳金融 上海 上海金融 北京 成都 香港 新加坡 多伦多 广州Open 硅谷 法兰克福 首尔

+ 新建 挂载 卸载

新建数据盘

弹性云盘可以在同一可用区内的虚拟机之间自由挂载和卸载，不支持跨可用区挂载

命名 磁盘名称
你还可以输入7个字符

所属项目 默认项目 ▾ 项目和区域

地域 广州 深圳金融 上海 上海金融 北京 成都 香港 新加坡

可用区 上海一区 上海二区

计费模式 包年包月 计费类型和磁盘类型

云硬盘类型 普通云硬盘 SSD云硬盘 高性能云硬盘

容量 GB 磁盘容量
200GB 500GB 1000GB 4000GB

使用快照创建磁盘 ☐

购买数量 购买数量
单次最多可创建10块弹性云硬盘

购买时长 1个月 2个月 3个月 88折 83折 7折 5折
半年 1年 2年 3年

费用 6元

确定 取消 确认购买

设置的主要参数包括：

1. 磁盘名称，例如：wordpress-dev
2. 选择所属项目和所在地域。
3. 选择计费类型和磁盘类型，这里选择包年包月和普通云硬盘

- 选择容量大小，对于不同环境使用的磁盘，选择不同大小。在 dev , test , pre-product 使用 10 GB , product 环境使用 50 GB。
- 选择磁盘数量，由于应用中两个服务各使用了一块磁盘，所以每个环境需要购买两块磁盘。
- 单击 完成 按钮，确认购买

购买完成后，等待 2~3 分钟，可以看 CBS 盘的页面查看到对应的磁盘。

ID/名称	监控	状态	容量	可用区	可挂载/卸载	磁盘类型	磁盘属性	已挂载云主机	计费模式
disk-166wta4n wdp-dirty		待挂载	10G	上海一区	支持	云硬盘	数据盘	-	包年包月
disk-9pqc3m05 wdp_1		待挂载	10G	上海一区	支持	云硬盘	数据盘	-	包年包月
disk-m36ojmi5 wdp_0		待挂载	10G	上海一区	支持	云硬盘	数据盘	-	包年包月
disk-peivemz1 wordpress-dev_0		待挂载	10G	上海一区	支持	云硬盘	数据盘	-	包年包月
disk-aq0280h5 wordpress-dev_1		待挂载	10G	上海一区	支持	云硬盘	数据盘	-	包年包月
disk-fifv4ht wordpress_test...		待挂载	10G	上海一区	支持	云硬盘	数据盘	-	包年包月
disk-1ywc14gb wordpress_test...		待挂载	10G	上海一区	支持	云硬盘	数据盘	-	包年包月
disk-fyzh7vgp wordpress_pre...		待挂载	10G	上海一区	支持	云硬盘	数据盘	-	包年包月
disk-pxtmsa7b wordpress_pre...		待挂载	10G	上海一区	支持	云硬盘	数据盘	-	包年包月
disk-r68yb55t wordpress_pro...		待挂载	50G	上海一区	支持	云硬盘	数据盘	-	包年包月
disk-frs991e5 wordpress_pro...		待挂载	50G	上海一区	支持	云硬盘	数据盘	-	包年包月

步骤四：创建不同环境的配置项

在不同环境中，可以将不同环境的差异化信息通过配置项保存。在配置项页面，单击新建按钮，可以创建对应的配置文件。

配置ID/配置名称	版本数量	创建时间	修改时间	操作
ccscf-7z1oyw7j sasdasd	4	2017-10-23 10:31:41	2017-10-27 11:49:35	删除
ccscf-qualhswd tst	2	2017-10-21 18:57:46	2017-10-23 10:31:13	删除
ccscf-a6mpwzy9 ccxz	1	2017-10-20 20:05:23	2017-10-20 20:05:23	删除
ccscf-lheywa7b sdada	1	2017-10-20 20:03:12	2017-10-20 21:24:25	删除
ccscf-93lc12o9 sdasda	3	2017-10-20 19:59:33	2017-10-20 20:04:32	删除
ccscf-cdc6vz0p asdasd	3	2017-10-19 21:45:53	2017-10-19 21:46:36	删除

dev 环境配置项：

服务配置

新建配置项

配置项名称

devconfig

配置项名称必须输入，最长60个字符

版本号

v1-0

只能包含小写字母、数字及分隔符("-")，且必须以小写字母开头，小写字母或数字结尾

版本描述

dev环境配置项

编辑方式

☒ YAML

☐ 可视化

配置内容

```
1 NAMESPACE: devnamespace
2 ReleaseCBS_mariadb_vol: disk-peivemz1
3 ReleaseCBS_wordpress_wordpress_persistent_storage: disk-aq0280h5
4 CPU_LIMITS: 200m
5 CPU_REQUEST: 100m
6
```

```

NAMESPACE: devnamespace
ReleaseCBS_mariadb_vol: disk-peivemz1
ReleaseCBS_wordpress_wordpress_persistent_storage: disk-aq0280h5
CPU_LIMITS: 200m
CPU_REQUEST: 100m

```

test 环境配置项：

服务配置 | 新建配置项

配置项名称

testconfig

配置项名称必须输入，最长60个字符

版本号

v1-0

只能包含小写字母、数字及分隔符("-")，且必须以小写字母开头，小写字母或数字结尾

版本描述

测试环境配置项

编辑方式

☒ YAML ☐ 可视化

配置内容

```
1 NAMESPACE: testnamespace
2 ReleaseCBS_mariadb_vol: disk-fifiv4ht
3 ReleaseCBS_wordpress_wordpress_persistent_storage: disk-1ywc14gb
4 CPU_LIMITS: 200m
5 CPU_REQUEST: 100m
```

NAMESPACE: testnamespace
ReleaseCBS_mariadb_vol: disk-fifiv4ht
ReleaseCBS_wordpress_wordpress_persistent_storage: disk-1ywc14gb
CPU_LIMITS: 200m
CPU_REQUEST: 100m

pre-product 环境配置项：

[服务配置](#) | 新建配置项

配置项名称

preconfig

配置项名称必须输入，最长60个字符

版本号

v1-0

只能包含小写字母、数字及分隔符("-"), 且必须以小写字母开头, 小写字母或数字结尾

版本描述

预发布环境配置项

编辑方式

☒ YAML ☐ 可视化

配置内容

```
1 NAMESPACE: prenamespace
2 ReleaseCBS_mariadb_vol: disk-fyzh7vgp
3 ReleaseCBS_wordpress_wordpress_persistent_storage: disk-pxtmsa7b
4 CPU_LIMITS: 200m
5 CPU_REQUEST: 100m
```

NAMESPACE: prenamespace**ReleaseCBS_mariadb_vol: disk-fyzh7vgp****ReleaseCBS_wordpress_wordpress_persistent_storage: disk-pxtmsa7b****CPU_LIMITS: 200m****CPU_REQUEST: 100m****product 环境配置项：**

[服务配置](#) | 新建配置项

配置项名称

prodconfig

配置项名称必须输入，最长60个字符

版本号

v1-0

只能包含小写字母、数字及分隔符("-")，且必须以小写字母开头，小写字母或数字结尾

版本描述

生产环境配置项

编辑方式

☒ YAML ☐ 可视化

配置内容

```
1 NAMESPACE: prodnamespace
2 ReleaseCBS_mariadb_vol: disk-r68yb55t
3 ReleaseCBS_wordpress_wordpress_persistent_storage: disk-frs991e5
4 CPU_LIMITS: 800m
5 CPU_REQUEST: 400m
```

NAMESPACE: prodnamespace**ReleaseCBS_mariadb_vol: disk-r68yb55t****ReleaseCBS_wordpress_wordpress_persistent_storage: disk-frs991e5****CPU_LIMITS: 800m****CPU_REQUEST: 400m**

在 product 环境中，设置更大的可使用资源。将 CPU_LIMITS 和 CPU_REQUEST 分别设置成 800 m(0.8 核)和 400 m(0.4 核)。

步骤五：创建不同环境的应用

新建应用

在[应用列表](#)选择创建了命名空间的集群。单击 [新建](#) 按钮。

选择应用对应的模板和配置

在新建应用页面，选择对应的应用模板和配置项。

腾讯云

总览

云产品

常用服务

容器服务

概览

应用中心

集群

应用

服务

Ingress

交付中心

镜像仓库

应用模板仓库

配置项

运维中心

日志

< 返回 | 新建应用

1 应用配置

应用名称

wordpress

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

应用描述

开发环境应用

地域

广州

上海

上海金融

北京

新加坡

成都

香港

运行集群

cls-a4ed48t8 (免费体验集群)

应用名称： 设置为 wordpress-dev

应用描述: 开发环境应用

地域选择: 选择 步骤二 中的集群所在的地域

集群选择： 选择 步骤二 中的集群

模板选择: 选择 步骤一 中创建的应用模板

配置项选择: 选择 步骤四 中创建的开发环境的配置项

单击 下一步 ，对应用中的模板内容进行再次编辑。由于我们已经在应用模板和配置项中完成了对应的设置，所以这里直接单击 完成 按钮，完成应用内容的编辑。

[返回](#) | 新建应用

```
6   creationTimestamp: null
7   name: mariadb
8   namespace: '{{.NAMESPACE}}'
9 spec:
10  replicas: 1
11  revisionHistoryLimit: 5
12  selector: {}
13  strategy: {}
14  template:
15    metadata:
16      creationTimestamp: null
```

← 模板内容

[导入服务](#)

配置项	变量名	变量值	操作
	NAMESPACE	devnamespace	×
	ReleaseCBS_mariadb_vol	disk-peivemz1	×
	ReleaseCBS_wordpress_wordpress_pers	disk-aq0280h5	×
	CPU_LIMITS	200m	×
	CPU_REQUEST	100m	×

[新增变量](#) [从模板内容导入](#)

← 配置项

[上一步](#) [完成](#) ← 完成应用内容编辑

查看应用状态

在应用列表页面，可以查看到新创建的应用，只是此时应用还处于未部署状态。

应用列表

广州

上海

北京

新加坡

成都

所属集群

cls-dubib439 (tst)

+ 新建

请输入关键字

应用名称	状态	已部署 / 总服务个数	操作
wordpress-dev	未部署	0/2	更新应用 删除

应用详情页面部署应用中的服务

单击应用的名称，可以查看应用的详情。在应用的详情页面，可以对应用进行部署操作。

返回 | wordpress-dev 详情

5

基本信息

应用名称

wordpress-dev

运行集群

cls-dubib439

应用描述

开发环境应用

创建时间

2017-10-31 21:26:55

更新时间

2017-10-31 21:26:55

服务列表

服务名	部署状态	服务状态	操作
mariadb	未部署	-	部署 更新 删除 取消部署 查看Yaml
wordpress	未部署	-	部署 更新 删除 取消部署 查看Yaml

单击 **部署** 按钮完成应用中服务的部署。

查看服务的信息

在完成部署后，应用中服务的状态变为 **已部署**，服务的运行状态变为 **运行中**。

[返回](#) | wordpress-dev 详情

应用

基本信息

应用名称

wordpress-dev

运行集群

cls-dubib439

应用描述

开发环境应用

创建时间

2017-10-31 21:26:55

更新时间

2017-10-31 21:42:35

服务列表

服务名	部署状态	服务状态	操作
mariadb	已部署	运行中	部署 更新 删除 取消部署 查看Yaml
wordpress	已部署	运行中	部署 更新 删除 取消部署 查看Yaml

单击服务的名称，可以跳转到服务详细页面，可以查看更多服务的信息。

[返回](#) | wordpress

应用

基本信息

服务名称

wordpress

状态

运行中

运行集群

cls-dubib439

负载均衡ID

lb-o10k4d1h

实例数量

1

标签(label)

qcloud-app:wordpress

qcloud-application-label:wordpress-dev

修改

创建时间

2017-10-31 21:42:35

描述

wordpress服务

服务配置

实例数量

1

负载均衡IP

111.231.97.228

（外网访问：绑定域名或VIP + 负载均衡监听端口）

服务IP

192.168.255.203

（集群内访问：服务名或服务IP + 服务监听端口）

访问方式

公网访问（lb-o10k4d1h, 111.231.97.228）

Yaml文件

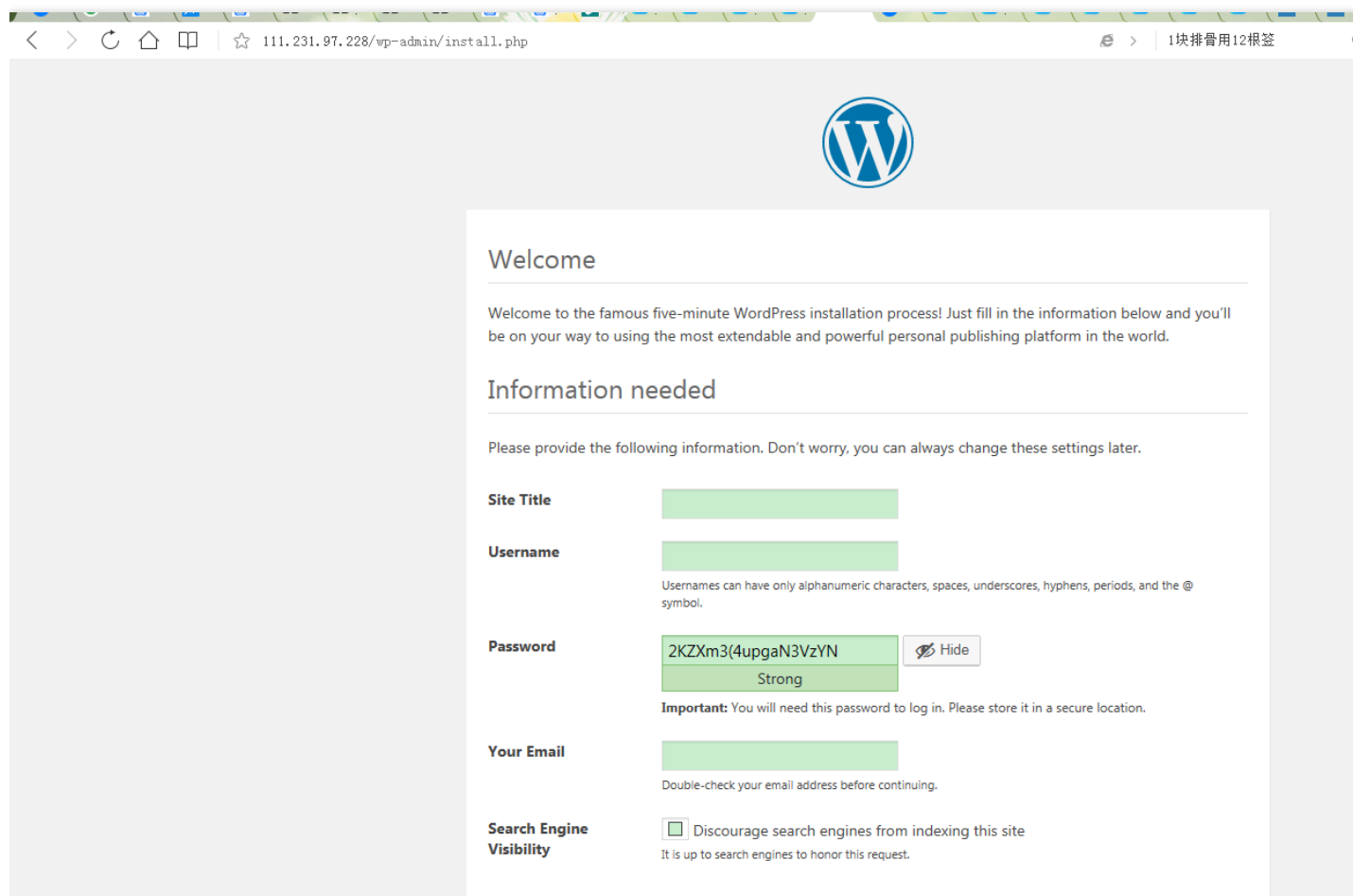
查看Yaml配置

端口映射

协议	服务端口（负载均衡监听端口）	主机端口	容器端
TCP	80	31583	80

访问服务测试

使用 wordpress 中服务访问的外网 IP 和服务端口，可以发起对服务的访问。



这样就完成了 dev(开发)环境的 wordpress 应用的部署。

应用部署到不同环境

执行同样的步骤，在应用配置时选择不同环境下对应的配置。可以将应用部署到不同的环境。部署完成后，可以在应用列表查看应用的信息。

应用列表 广州 上海 北京 新加坡 成都 所属集群 cls-dubib439 (tst) ▼				
+ 新建 请输入关键字				
应用名称	状态		已部署/总服务个数	操作
wordpress-prod	已部署	← 生产环境应用	2/2	更新应用 删除
wordpress-pre	已部署	← 预发布环境应用	2/2	更新应用 删除
wordpress-test	已部署	← 测试环境应用	2/2	更新应用 删除
wordpress-dev	已部署	← 开发环境应用	2/2	更新应用 删除

这样基于同一个应用模板和不同环境下的配置信息，就将应用部署到了不同的环境。

应用

应用概述

最近更新时间：2018-03-15 20:57:25

腾讯云容器服务支持通过应用的方式对服务进行分组。可以将关联性比较强的多个服务放在一个应用统一进行管理，从而简化了服务管理的复杂度。

应用支持的主要操作

在应用中支持通过应用模板+配置快速的创建和更新应用。支持单独对应用内的服务进行部署操作和更新操作。具体的操作说明可以参考：

- [创建应用](#)
- [删除应用](#)
- [更新应用](#)
- [查看应用](#)
- [管理应用内服务](#)

更多关于应用管理的内容可以参考[应用管理概述](#)。

应用示例-nginx单服务应用

最近更新时间：2018-09-13 10:11:59

在应用模板中我们使用nginx单服务应用作为应用模板的基础示例。在应用中我们同样使用nginx单服务应用作为示例。

步骤一：创建nginx单服务的应用的模板

如果不存在nginx单服务应用的应用模板，则需要创建对应的应用模板，具体的创建过程可以参考[应用模板示例-Nginx单服务应用](#)。

如果已经存在对应的应用模板，则跳过此步骤。

步骤二：新建应用

在[应用列表](#)中，单击新建按钮，开始基于应用模板创建应用。



步骤三：配置应用

在新建应用后，进入到应用配置页面，在该页面执行下面可以进行相应的配置。具体的内容包括：

1. 填写应用名称和应用描述
2. 选择应用需要部署到的地域和集群集群

腾讯云 总览 云产品 常用服务

容器服务 << < 返回 | 新建应用

1 应用配置 >

应用名称
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

应用描述

地域 广州 上海 上海金融 北京 新加坡 成都 香港

运行集群

单击下一步进入应用编辑页面，可以对应用中模板内容和配置项进行进一步的修改。

1. 选择应用模板，这里选择刚刚创建的 nginx 这个模板
2. 选择配置项。可以使用模板中默认的配置项也可以根据需求选择特定的配置项。这里我们选择使用模板中默认的配置项。

注意：

创建应用时会从应用模板拷贝一份副本。在应用中更新模板内容，修改的内容并不会同步回原来的应用模板。

步骤四：编辑应用

在应用编辑操作中，可以更加应用的需要对模板中的内容和配置文件的内容进行修改。因为我们在模板中已经将内容已经编辑好了，不再需要额外的修改，所以直接单击 **完成** 按钮，完成应用的编辑。更多关于应用编辑的操作，可以参考[应用模板内容操作指引](#)。

腾讯云

总览

云产品

常用服务

备案

Qcloud基础...

容器服务

概览

应用中心

集群

应用

服务

Ingress

交付中心

镜像仓库

应用模板仓库

配置项

运维中心

日志

体验&指引

免费实验室

使用指引

返回

新建应用

应用配置

应用列表

tp-61von2sq (nginx) 重新选择

选择应用模板后，将从模板内容复制到当前应用，继续编写或修改当前应用不会影响模板本身。

服务名	操作	内容
nginx	删除	<pre> 1 apiVersion: extensions/v1beta1 2 kind: Deployment 3 metadata: 4 annotations: 5 description: nginx 6 creationTimestamp: null 7 name: nginx 8 namespace: '{{.NAMESPACE}}' 9 spec: 10 replicas: 1 11 revisionHistoryLimit: 5 12 selector: {} 13 strategy: 14 rollingUpdate: 15 maxSurge: 1 16 maxUnavailable: 0 17 type: RollingUpdate 18 template: 19 metadata: </pre>

[新增空服务](#)
[从UI导入服务](#)

在应用编辑完成后，这个应用已经创建。在应用列表页面可以看到对应的应用。只是这时候应用还处于未部署状态。单击应用的名称，进入应用详情页面，可以对应用中的服务进行部署操作。

步骤五：部署应用中的服务

- 在应用详情页面，可以看到服务的状态为未部署，单击 **部署** 按钮，部署服务。

云产品 常用服务 备案 费用 工单 管理应用

< 返回 nginx 详情

应用名称 nginx

运行集群 cls-et0tztru

应用描述 nginx服务

创建时间 2018-02-06 16:15:04

更新时间 2018-02-06 16:15:04

服务列表

服务名	类型	部署状态	运行状态	IP地址	操作
nginx	Deployment+Service	未部署	-	-	部署 查看Yaml

1. 这样应用中的服务在集群中进行部署，查看应用中服务的状态已经变为了启动中。

云产品 常用服务 备案 Qcloud基础... 费用 工单 管理应用

< 返回 nginx 详情

应用名称 nginx

运行集群 cls-et0tztru

应用描述 nginx服务

创建时间 2018-02-06 16:15:04

更新时间 2018-02-06 16:16:05

服务列表

服务名	类型	部署状态	运行状态	IP地址	操作
nginx	Deployment+Service	已部署	启动中	192.168.255.24	取消部署 查看Yaml

重要：

更多关于应用内服务的操作可以参考[管理应用内服务](#)。

1. 单击应用中对应的服务，跳转到服务页面查看服务详情。

[< 返回](#) | nginx

实例列表

服务信息

实例信息

事件

日志

基本信息

服务名称 ?	nginx
状态	运行中
运行集群	cls-afnz40e3
负载均衡ID	lb-5duxxtc1
实例数量	2
标签(label)	qcloud-app:nginx qcloud-application-label:nginxapp 修改
创建时间	2017-11-09 16:30:46
描述	nginx服务

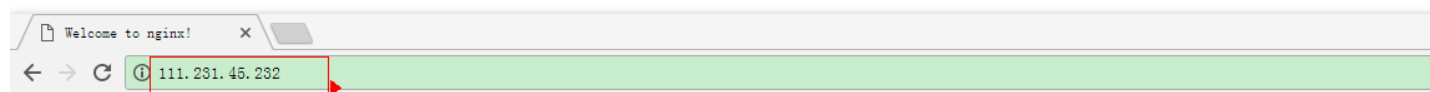
服务配置

实例数量	2
负载均衡IP	111.231.45.232 Ⓜ (外网访问：绑定域名或VIP + 负载均衡监听端口)
服务IP	172.23.255.26 Ⓜ (集群内访问：服务名或服务IP + 服务监听端口)
访问方式 ?	公网访问 (lb-5duxxtc1, 111.231.45.232)
Yaml文件	查看Yaml配置

外网访问地址

步骤六：访问部署的应用

在应用部署完成后，可以通过应用中服务暴露的外网访问地方进行访问。对 nginx 服务中，可以通过暴露的外网负载均衡器IP和端口访问服务。



通过负载均衡器IP，访问服务

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

应用示例-guestbook应用

最近更新时间：2018-09-14 09:42:22

Guestbook留言板是一个比较典型的web应用服务，由一个frontend前端服务和redis-master和redis-slave两个后端存储服务组成。用户通过web前端提交数据，写入到redis-master上，然后通过读取同步到redis-slave上的数据展示给用户。

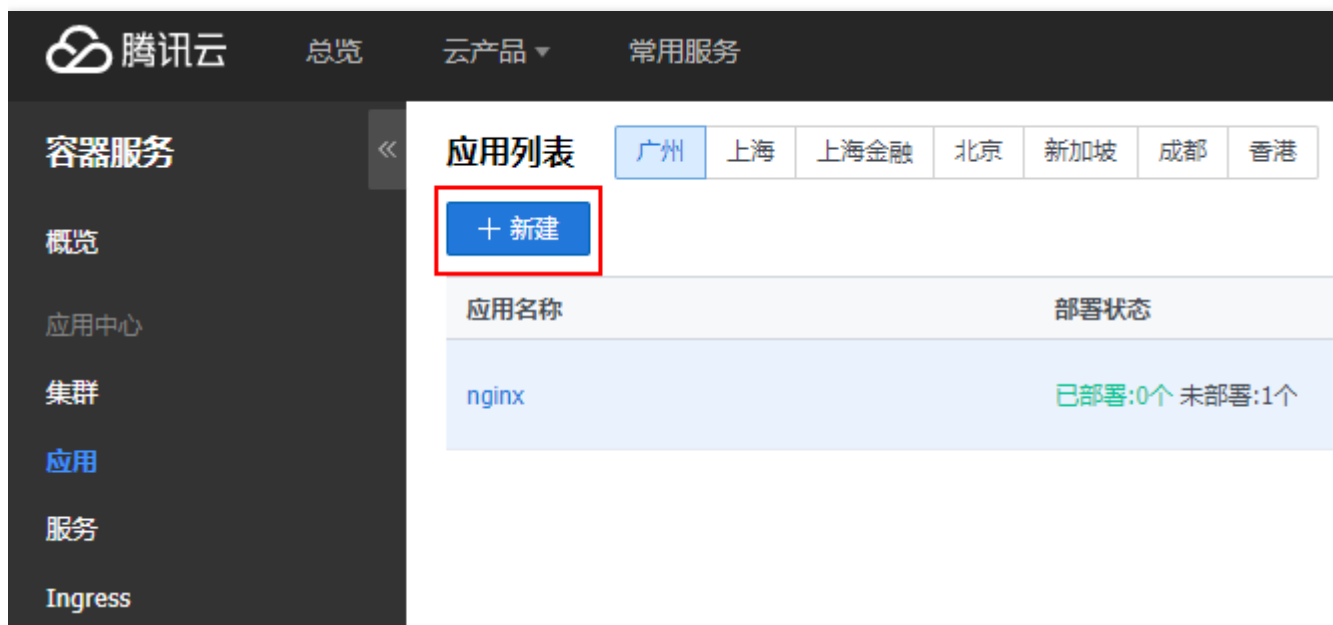
在[Guestbook应用模板](#)中已经介绍了如何将 Guestbook 的部署信息通过模板的方式保存起来。本示例将介绍如何使用创建好的模板来快速部署 Guestbook应用。

步骤一：创建Guestbook的应用的模板

创建Guestbook的应用模板的过程，可以参考[Guestbook应用模板](#)

步骤二：新建应用

在[应用列表](#)中，单击新建按钮。



步骤三：配置应用

在新建应用后，进入到应用配置页面，在该页面执行下面的操作

1. 填写应用名称和描述
2. 选择地域和应用需要部署到的集群

腾讯云

总览

云产品

常用服务

容器服务

概览

应用中心

集群

应用

服务

Ingress

交付中心

镜像仓库

应用模板仓库

配置项

运维中心

< 返回

新建应用

1 应用配置

应用名称

guestbook

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

应用描述

guestbook

地域

广州

上海

上海金融

北京

新加坡

成都

香港

运行集群

cls-et0tztru (cls0206)

单击下一步进入应用编辑页面，可以对应用中模板内容和配置项进行进一步的修改。

1. 选择应用模板，这里选择刚刚创建的 Guestbook 这个模板
2. 选择配置项，可以使用模板中默认的配置项也可以根据需要进行选择特定的配置项。这里我们选择使用模板中默认的配置项。



注意：

创建应用时会从应用模板拷贝一份副本。在应用中更新模板内容，修改的内容并不会同步回原来的应用模板。

步骤四：编辑应用

在应用编辑操作中，可以根据应用的需要对模板中的内容和配置文件的内容进行修改。因为我们在模板中已经将内容编辑好了，不再需要额外的修改，所以直接单击 **完成** 按钮，完成应用的编辑。更多关于应用编辑的操作，可以参考[应用模板内容操作指引](#)。

[返回](#) | 新建应用

应用配置

服务列表 tp-nwsmmlce (guestbook) [重新选择](#)

选择应用模板后，将从模板内容复制到当前应用，继续编写或修改当前应用不会影响模板本身。

服务名	操作	内容
frontend	删除	<pre>1 apiVersion: extensions/v1beta1 2 kind: Deployment 3 metadata: 4 name: frontend 5 namespace: {{.NAMESPACE}} 6 spec: 7 replicas: {{.FRONTEND_REPLICAS}} 8 template: 9 spec: 10 containers: 11 - env: 12 - name: GET_HOSTS_FROM 13 value: dns 14 image: ccr.ccs.tencentyun.com/library/gb-frontend:{{.FRONTEND_VERSION}} 15 imagePullPolicy: Always 16 name: php-redis 17 resources: 18 requests: 19 cpu: 100m</pre>
redis-master	删除	
redis-slave	删除	

[新增空服务](#) [从UI导入服务](#)

配置项 以上服务配置中，若存在自定义变量，可以手动输入为变量赋值，也可使用已有的配置项来为变量赋值，更多参见 [变量设置说明](#)
注: 变量替换类型的配置项用于字符串替换，文件挂载类型的配置项用于将指定内容挂载到容器

变量名	变量值	类型
NAMESPACE	default	变量替换
FRONTEND_REPLICAS	2	变量替换
FRONTEND_VERSION	v4	变量替换
REDIS_MASTER_VERSION	e2e	变量替换

在应用编辑完成后，这个应用已经创建。在应用列表页面可以看到对应的应用。只是这时候应用还处于未部署状态。单击应用的名称，进入应用详情页面，可以对应用中的服务进行部署操作。

步骤五：部署应用中的服务

在应用详情页面，可以看到服务的状态为未部署，单击部署按钮，部署服务。

[返回](#) | [guestbook 详情](#)

[管理应用指引](#)

基本信息

应用名称 [guestbook](#)
运行集群 [cls-et0tztru](#)
应用描述 [guestbook](#)
创建时间 2018-02-06 16:32:13
更新时间 2018-02-06 16:32:13

服务列表

服务名	类型	部署状态	运行状态	IP地址	操作
frontend	Deployment+Service	未部署	-	-	部署 查看Yaml
redis-master	Deployment+Service	未部署	-	-	部署 查看Yaml
redis-slave	Deployment+Service	未部署	-	-	部署 查看Yaml

查看应用中服务的状态已经变为了启动中。

[返回](#) | [guestbook 详情](#)

[管理应用指引](#)

基本信息

应用名称 [guestbook](#)
运行集群 [cls-et0tztru](#)
应用描述 [guestbook](#)
创建时间 2018-02-06 16:32:13
更新时间 2018-02-06 16:32:53

服务列表

服务名	类型	部署状态	运行状态	IP地址	操作
frontend	Deployment+Service	已部署	启动中	- 192.168.255.62	取消部署 查看Yaml
redis-master	Deployment+Service	已部署	启动中	- 192.168.255.179	取消部署 查看Yaml
redis-slave	Deployment+Service	已部署	启动中	- 192.168.255.191	取消部署 查看Yaml

单击应用中服务的名称，跳转到服务页面查看服务详情。

[< 返回](#) | frontend[实例列表](#)[服务信息](#)[实例信息](#)[事件](#)[日志](#)

基本信息

服务名称 	frontend		
状态	运行中		
运行集群	cls-afmxl2k9		
负载均衡ID	lb-01hfghtpn		
实例数量	2		
标签(label)	qcloud-app:frontend	qcloud-application-label:guestbook	修改
创建时间	2017-11-09 20:15:52		
描述	无		

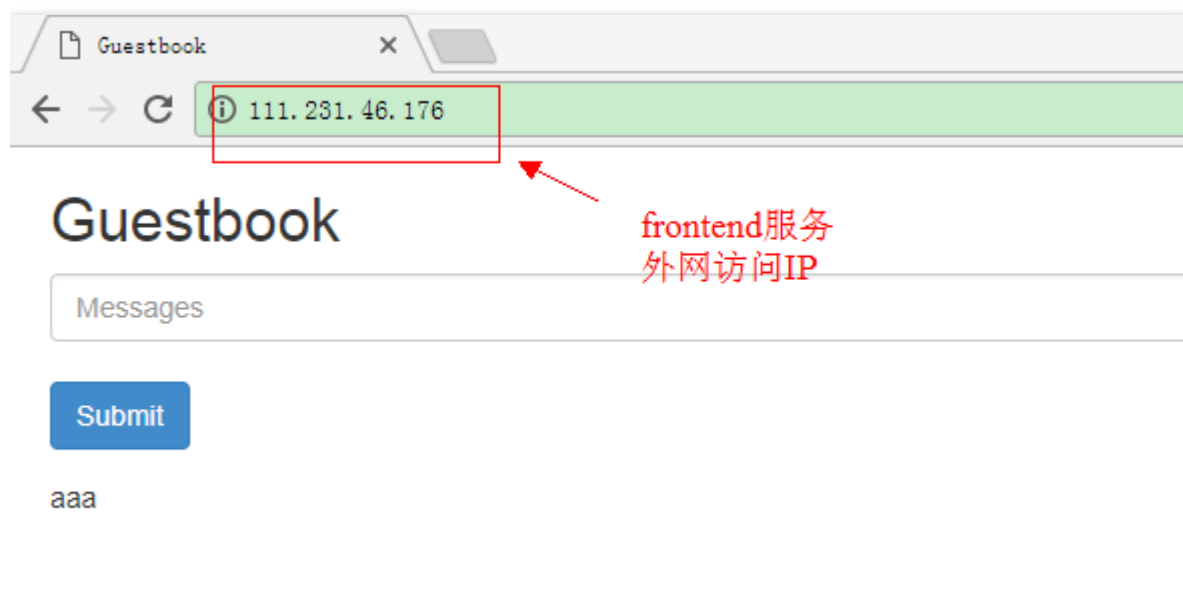
服务配置

实例数量	2
负载均衡IP	111.231.46.176  (外网访问：绑定域名或VIP + 负载均衡监听端口)
服务IP	172.23.255.136  (集群内访问：服务名或服务IP + 服务监听端口)
访问方式 	公网访问 (lb-01hfghtpn, 111.231.46.176)
Yaml文件	查看Yaml配置

服务外网
访问IP

步骤六：访问部署的应用

通过 frontend 服务的负载均衡器IP可以直接访问前端的服务。



创建应用

最近更新时间：2018-07-05 15:41:25

登录 [腾讯云容器服务控制台](#)。在应用管理->[应用页面](#)单击新建。



应用配置

在新建应用后，进入到应用配置页面，在该页面执行下面可以进行相应的配置。具体的内容包括：

1. 填写应用名称和应用描述
2. 选择应用需要部署到的地域和集群

腾讯云

总览

云产品

常用服务

容器服务

概览

应用中心

集群

应用

服务

Ingress

交付中心

镜像仓库

应用模板仓库

配置项

返回

新建应用

1应用配置

应用名称

nginx

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

应用描述

nginx服务

地域

广州

上海

上海金融

北京

新加坡

成都

香港

运行集群

cls-et0tztru (cls0206)

注意：

创建应用时会从应用模板拷贝一份副本。在应用中更新模板内容，修改的内容并不会同步回原来的应用模板。

应用编辑

配置应用后，进入应用编辑页面。在应用编辑页面可以对模板和配置的内容进行进行修改。修改后单击保存，完成应用内容的编辑。更多关于模板内容的操作，可以参考[应用模板内容操作指引](#)。

腾讯云

总览

云产品

常用服务

备案

Qcloud基础...

容器服务

概览

应用中心

集群

应用

服务

Ingress

交付中心

镜像仓库

应用模板仓库

配置项

运维中心

日志

体验&指引

免费实验室

使用指引

< 返回

新建应用

应用配置

服务列表

tp-61von2sq (nginx) 重新选择

选择应用模板后，将从模板内容复制到当前应用，继续编写或修改当前应用不会影响模板本身。

服务名	操作	内容
nginx	删除	<pre>1 apiVersion: extensions/v1beta1 2 kind: Deployment 3 metadata: 4 annotations: 5 description: nginx 6 creationTimestamp: null 7 name: nginx 8 namespace: '{{.NAMESPACE}}' 9 spec: 10 replicas: 1 11 revisionHistoryLimit: 5 12 selector: {} 13 strategy: 14 rollingUpdate: 15 maxSurge: 1 16 maxUnavailable: 0 17 type: RollingUpdate 18 template: 19 metadata:</pre>

[新增空服务](#) [从UI导入服务](#)

编辑完成后，单击 **完成** 按钮，完成应用创建。

部署应用中的服务

创建应用后，在[应用列表](#)可以看到新创建的应用。单击应用的名称，可以进入应用的详情页面，查看应用中各个服务的状态，以及对服务进行部署的操作。

[返回](#) | nginx 详情[管理应用](#)

基本信息

应用名称	nginx
运行集群	cls-et0tztru
应用描述	nginx服务
创建时间	2018-02-06 16:15:04
更新时间	2018-02-06 16:15:04

服务列表

服务名	类型	部署状态	运行状态	IP地址 ?	操作
nginx	Deployment+Service	未部署	-	-	部署 查看Yaml

进入应用详情后，可以看到应用中具体的服务信息。此时服务的状态还是未部署。单击 [部署](#) 按钮，执行服务的部署操作。

这样应用的服务就部署完成了，在详情页面可以看到服务的状态从 [未部署](#) 变为了 [已部署](#)。服务的运行状态变为了 [启动中](#)。

云产品 常用服务 备案 Qcloud基础... 费用 工单 9.2

< 返回 nginx 详情 管理应用

基本信息

应用名称	nginx
运行集群	cls-et0tztru
应用描述	nginx服务
创建时间	2018-02-06 16:15:04
更新时间	2018-02-06 16:16:05

服务列表

服务名	类型	部署状态	运行状态	IP地址	操作
nginx	Deployment+Service	已部署	启动中	192.168.255.24	取消部署 查看Ya

注意：

更多关于应用内服务的操作和状态说明可以参考[应用内服务管理](#)

访问部署的应用

在应用部署完成后，可以通过应用中服务暴露的外网访问地方进行访问。对 nginx 服务中，可以通过暴露的外网负载均衡器IP和端口访问服务。

Welcome to nginx! 111.231.46.207 服务外网访问IP

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

删除应用

最近更新时间：2018-05-30 15:16:43

删除应用

登录 [腾讯云容器服务控制台](#)。在应用管理->[应用列表](#)页面查看用户集群下的所有的应用。

在应用列表的右侧有一个应用有一个 [删除](#) 按钮，单击 [删除](#) 按钮即可删除应用。

应用列表 广州 上海 北京 新加坡 成都 所属集群 cls-mhihn5at (devinyan_勿删) 选择集群

+ 新建 请输入关键字

应用名称	状态	已部署 / 总服务个数	操作
tst	已部署	1/1	更新应用 删除
apache	已部署	2/2	更新应用 删除
pythonconfig	已部署	1/1	更新应用 删除
nginxhttp	已部署	1/1	更新应用 删除
nginxapp2	已部署	1/1	更新应用 删除
guestbook	已部署	3/3	更新应用 删除
simon55	已部署	1/1	更新应用 删除
simon2	未部署	0/1	更新应用 删除
privnginx	未部署	1/1	更新应用 删除
simon	未部署	0/1	更新应用 删除

注意

删除应用时会自动删除应用下部署的所有服务，请谨慎操作

更新应用

最近更新时间：2018-05-30 15:29:01

在应用列表页面更新应用

登录 [腾讯云容器服务控制台](#)。在应用管理->[应用页面](#)查看应用列表。单击右侧的 **更新应用** 按钮，更新指定的应用。



直接更新YAML或选择新模板更新

通过应用模板更新指，选择新的应用模板。将新的应用模板中的内容替换现有应用中服务的模板内容。在更新方式选择页面，选择通过模板更新。或者也可以直接编辑Yaml内容进行更新。

腾讯云

总览

云产品

常用服务

容器服务

概览

应用中心

集群

应用

服务

Ingress

交付中心

镜像仓库

应用模板仓库

配置项

运维中心

日志

体验&指引

免费实验室

使用指引

返回 | 更新应用

应用配置

编辑应用

预览变更

服务列表

选择模板导入

选择应用模板后，将从模板内容复制到当前应用，继续编写或修改当前应用不会影响模板本身。

服务名	操作	内容
frontend	删除	1 apiVersion: extensions/v1beta1 2 kind: Deployment 3 metadata: 4 name: frontend 5 namespace: {{.NAMESPACE}} 6 spec: 7 replicas: {{.FRONTEND_REPLICAS}} 8 template: 9 spec: 10 containers: 11 - env: 12 - name: GET_HOSTS_FROM 13 value: dns 14 image: ccr.ccs.tencentyun.com/library/gb-frontend:{{.FRONTEND_VERSION}} 15 imagePullPolicy: Always 16 name: php-redis 17 resources: 18 requests: 19 cpu: 100m
redis-master	删除	
redis-slave	删除	

新增空服务 从UI导入服务

配置项

以上服务配置中，若存在自定义变量，可以手动输入为变量赋值，也可使用已有的配置项来为变量赋值，更多参见 [变量设置说明](#)
注：变量替换类型的配置项用于字符串替换，文件挂载类型的配置项用于将指定内容挂载到容器

变量名	变量值	类型
NAMESPACE	default	变量替换
FRONTEND_REPLICAS	2	变量替换
FRONTEND_VERSION	v4	变量替换
REDIS_MASTER_VERSION	e2e	变量替换

查看应用变更内容

更新选择，单击 下一步 按钮。可以查看本次的变更内容。

[返回](#) | [更新应用](#)

[更新应用指引](#)

应用配置

编辑应用

3 预览变更

变更内容

服务	描述	
frontend	有变更	1 frontend变更详情
redis-master	无变更	2 apiVersion: extensions/v1beta1
redis-slave	无变更	3 kind: Deployment
		4 metadata:
		5 name: frontend
		6 namespace: default
		7 spec:
		8 replicas: 3
		9 replicas: 2
		10 template:

注意：

创建应用时会从应用模板拷贝一份副本。在更新直接应用内模板的内容，不会同步回原来的应用模板。

单击应用更新页面 完成 按钮，完成应用的更新。

执行应用的更新

在应用详情页面，可以查看到应用中服务的状态已经从 已部署 ，转变成了 待更新 。更多关于应用中服务状态的说明，可以参考[管理应用内服务](#)。

应用列表

广州

上海

上海金融

北京

新加坡

成都

香港

所属集群

cls-et0tztru (cls0206)

应用编排操作文档

+ 新建

请输入应用名称

应用名称	部署状态	服务状态	创建时间	操作
guestbook	已部署:2个 未部署:0个 待更新:1个	正常	2018-02-06 16:32:13	更新应用 部署 删除
nginx	已部署:0个 未部署:1个	-	2018-02-06 16:15:04	更新应用 部署 删除

< 返回

guestbook 详情

管理应用指引

基本信息

应用名称	guestbook
运行集群	cls-et0tztru
应用描述	guestbook
创建时间	2018-02-06 16:32:13
更新时间	2018-02-06 16:43:37

服务列表

服务名	类型	部署状态	运行状态	IP地址	操作
frontend	Deployment+Service	待更新	运行中	111.230.190.141 192.168.255.62	更新 取消部署 查看Yaml
redis-master	Deployment+Service	已部署	运行中	- 192.168.255.179	取消部署 查看Yaml
redis-slave	Deployment+Service	已部署	运行中	- 192.168.255.191	取消部署 查看Yaml

单击 **更新** 按钮，将使用新的模板内容和配置项对服务进更新。更新完成后，服务的状态重新变为 **已部署**。

注意：

只有在应用模板内容发生变化时，才能执行更新操作，更新应用中的服务。(如果应用中配置文件中变量的值发生了变化，而服务模板内容使用到了该变量，也视为服务的模板内容发生了变化。)

查看应用

最近更新时间：2018-05-30 15:26:01

查看应用列表

登录 [腾讯云容器服务控制台](#)。在应用管理->[应用页面](#)。选择地域的区域和集群，可以查看集群下所有的应用。

应用列表 广州 上海 北京 新加坡 所属集群 cls-mhihn5at (devinyan_勿删) ▼			
+ 新建		请输入关键字	
应用名称	状态	已部署/总服务个数	操作
tst	未部署	0/1	更新应用 删除
apache	已部署	2/2	更新应用 删除
pythonconfig	已部署	1/1	更新应用 删除
nginxhttp	已部署	1/1	更新应用 删除
nginxapp2	已部署	1/1	更新应用 删除
guestbook	已部署	3/3	更新应用 删除
simon55	已部署	1/1	更新应用 删除
simon2	未部署	0/1	更新应用 删除
privnginx	未部署	1/1	更新应用 删除
simon	未部署	0/1	更新应用 删除
test	未部署	0/1	更新应用 删除
cbs	未部署	0/1	更新应用 删除

应用过滤搜索

在应用列表搜索框中，输入名称可以过滤搜索出包含特定名称的应用。例如：输入 `nginx` 可以过滤出该集群下所有名称中包含nginx的应用。

应用列表

广州

上海

北京

新加坡

所属集群

cls-mhihn5at (devinyan_勿删)

▼

+ 新建

过滤指定名称

nginx

应用名称	状态	已部署/总服务个数	操作
搜索 "nginx" 找到 3 条结果, 返回原列表			
nginxhttp	已部署	1/1	更新应用 删除
nginxapp2	已部署	1/1	更新应用 删除
privnginx	未部署	1/1	更新应用 删除

查看应用详情

单击应用名称，可以查看应用详情。

< 返回

tst 详情

常见

基本信息

应用名称

tst

运行集群

cls-mhihn5at

应用描述

创建时间

2017-10-18 21:34:29

更新时间

2017-10-18 22:22:39

服务列表

应用中服务的部署状态

应用中服务的运行状态

应用中服务的操作

查看服务模板解析后的YAML

服务名	部署状态	服务状态	操作	
aaa	已部署	运行中	部署 更新 删除 取消部署	查看Yaml

应用详情页面，可以看到对应用中的服务进行相应操作。更过关于应用中服务的操作可以参考[管理应用内服务](#)。也可以查看应用中服务模板内容的YAML文件。模板内容变量会使用配置项中的对应值进行替换。

管理应用内服务

最近更新时间：2018-04-18 16:07:29

应用详情页面对服务进行操作

在应用详情页面，可以看到应用下所有服务的状态。服务的状态包括部署状态和运行状态。在对应服务的右侧，有服务操作的按钮，支持多种对服务的操作。

[返回](#) | [guestbook 详情](#)

基本信息

应用名称

guestbook

运行集群

cls-kiy3gmn5

应用描述

guestbook测试

创建时间

2017-10-20 10:15:32

更新时间

2017-10-20 10:18:55

服务列表

服务名	部署状态	服务状态	操作
frontend	部署待更新	运行中	部署 更新 删除 取消部署 查看Yaml
redis-master	部署待删除	运行中	部署 更新 删除 取消部署 查看Yaml
redis-slave	未部署	-	部署 更新 删除 取消部署 查看Yaml
nginx	未部署	-	部署 更新 删除 取消部署 查看Yaml

部署状态：服务内资源是否执行部署操作的描述

部署状态名称	部署状态说明
未部署	服务只是编辑了模板内容，未执行部署操作
已部署	服务只是编辑了模板内容后，执行部署操作完成
待更新	服务只执行部署操作后，再次编辑模板内容(或者配置项)
待删除	服务只执行部署操作后，在模板内容中删除该服务

在应用更新页面更新应用后，如果服务的模板内容有修改，则服务部署状态会改变。但服务的运行状态不会发生改变。

服务状态: 服务部署后，实际的运行状态

服务状态名称	服务状态说明
运行中	服务部署完成后，服务正常运行
启动中	服务部署后，服务正在启动
异常	服务部署后，服务实例运行异常
查询异常	服务部署后，查询服务信息失败
-	服务未部署，显示空的服务状态信息

服务不同部署状态下支持的操作说明

在应用中的服务支持：[部署](#)，[取消部署](#)，[更新](#)，[删除](#)，[查看YAML](#) 等操作。在服务不同状态下，支持的操作如下表所示：

状态名称	部署操作	取消部署操作	更新操作	删除部署操作	查看YAML操作
待部署	支持	不支持	不支持	不支持	支持
已部署	不支持	支持	不支持	不支持	支持
待更新	不支持	支持	支持	不支持	支持
待删除	不支持	不支持	不支持	支持	不支持

服务差异化对比状态说明

在应用更新页面，会自动对比选择将要修改的模板内容和应用中现有的模板内容直接的差异。对比后用 [新增](#)，[无变化](#)，[有更改](#)，[将删除](#) 这几个状态。具体的状态说明如下表所示：

状态名称	差异化状态说明
新增	应用中没有该服务，但在修改的内容中增加了该服务
无变化	将要修改的模板内容与应用中该服务现有的模板内容无差异
有更改	将要修改的模板内容与应用中该服务现有的模板内容有差异
删除	应用中有该服务，但在修改的内容中该服务不存在

应用模板

应用模板概述

最近更新时间：2018-03-15 21:18:10

在腾讯云容器服务中可以将多个服务的部署信息，通过YAML形式的描述文本保存到应用模板中。通过应用模板和不同环境下的配置，可以快速的在不同环境中部署应用中的服务。更多关于应用管理的内容可以参考[应用管理概述](#)。

应用模板主要操作

- [创建应用模板](#)
- [删除应用模板](#)
- [更新应用模板](#)
- [查看应用模板](#)

编排语法能力说明

应用模板中依赖于编排语法对服务的部署信息进行说明。编排语法本质对服务部署信息描述的一套规则，用户按照这个规则来描述服务的部署信息，编排引擎对描述信息进行解析后，按照解析得到的参数对服务进行部署。

腾讯云容器服务应用模板的编排语法底层支持kubernetes的原生编排语法，关于kubernetes的编排语法的详细说明可以参考[基础语法](#)。在kubernetes的编排语法的基础上支持变量替换，变量替换的详细说明可以参考[变量替换](#)。同时考虑到容器服务平台本身的能力，提供容器服务平台相关的扩展语法，具体的关于扩展语法的说明可以参考[扩展语法](#)。出于功能稳定闭环的考虑，容器平台对kubernetes的编排语法做了一定的限制。具体的限制说明可以参考[限制说明](#)。

应用模板示例-nginx单服务应用

最近更新时间：2018-05-30 15:30:12

nginx是最基础的一个服务，在应用中也算是最基础的一个应用，有点类似于"Hello World"示例。应用模板的示例，我们从"Hello World"的nginx应用开始。

步骤一: 新建应用模板

在[应用模板](#)列表中，单击新建按钮。

步骤二: 应用模板编辑

2.1 添加nginx服务**

应用模板中增加服务，是通过导入服务相对应的模板内容实现的。导入服务模板内容包括两种可选的方式: 1. 从控制台导入 2. YAML文件导入

注意：

可以根据场景选择使用其中的任意一种方式

更多关于服务导入的说明可以参考[\[应用模板内容操作指引\]](#)[4]。

导入方法1：控制台导入服务

单击UI导入服务按钮，会弹出导入服务的控制台，在导入服务控制台填写相应参数。

The screenshot shows the 'Import Service' (导入服务) page in the Tencent Cloud console. The left sidebar contains navigation links for 'Container Service' (容器服务), 'Overview' (概览), 'Application Center' (应用中心), 'Cluster' (集群), 'Application' (应用), 'Service' (服务), 'Ingress', 'Delivery Center' (交付中心), 'Image Repository' (镜像仓库), 'Application Template Repository' (应用模板仓库), 'My Templates' (我的模板), 'Template Market' (模板市场), 'Configuration Items' (配置项), 'Operations Center' (运维中心), 'Logs' (日志), 'Experience & Guide' (体验&指引), 'Free Lab' (免费实验室), and 'Usage Guide' (使用指引). The main content area is titled 'Import Service' (导入服务) and includes a 'Return' (返回) link. The form fields are as follows:

- Service Name (服务名称):** nginx. Below the input field, it states: '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'.
- Service Description (服务描述):** nginx. Below the input field, it states: '描述信息不超过1000个字符'.
- Data Volume (数据卷) (选填):** A link to 'Add Data Volume' (添加数据卷). Below it, it states: '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、配置项，还需挂载到容器的指定路径中。使用指引'.
- Run Container (运行容器):** A modal window with the following fields:
 - Name (名称):** nginx. Below the input field, it states: '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'.
 - Image (镜像):** nginx. To the right is a link 'Select Image' (选择镜像).
 - Version (Tag):** latest.At the bottom of the modal is a link 'Show Advanced Settings' (显示高级设置).

在 nginx 服务中设置的参数包括：

设置服务的基本信息：

1. 填写服务名称 nginx
2. 填写服务描述 nginx服务

设置服务的数据卷信息：

未使用数据盘，无。

设置镜像参数：

1. 在设置容器运行参数中的镜像参数：
 - 容器名称设置为 nginx
 - 镜像名称设置为 nginx
 - 版本号选择为 latest

设置服务的实例数：

1. 服务的实例数设置为1

设置服务的访问方式：

1. 服务的访问方式设置为集群内访问
2. 服务的访问端口：容器端口和服务端口都设置成80

更多关于参数设置内容可以参考 [服务创建][7]操作的相关文档。

填写参数后，单击导入服务的 **完成** 按钮，控制台自动导入服务的模板内容。

导入方法2：YAML文件导入

如果已经存在服务对应的模板内容的YAML文件，可以直接将模板内容导入到编辑框中。具体的步骤如下：

步骤一 创建对应的服务：

单击"新增空服务"号按钮，新增一个服务。服务名称设置为 **nginx**。

步骤二 导入模板内容：

可以将下面YAML文件中的内容，直接拷贝到编辑框中，导入服务的模板内容。

```
apiVersion: extensions/v1beta1
```

```
kind: Deployment
```

```
metadata:
```

```
  annotations:
```

```
  description: nginx服务
```

```
  creationTimestamp: null
```

```
  name: nginx
```

```
  namespace: '{{.NAMESPACE}}'
```

```
spec:
```

```
  replicas: 1
```

```
  revisionHistoryLimit: 5
```

```
  selector: {}
```

```
  strategy: {}
```

```
  template:
```

```
    metadata:
```

```
      creationTimestamp: null
```

```
    spec:
```

```
      containers:
```

```
        - image: nginx:latest
```

```
        imagePullPolicy: Always
```

```
        name: nginx
```

```
      resources:
```

```
        requests:
```

```
          cpu: 200m
```

```
      securityContext:
```

```
        privileged: false
```

```
      serviceAccountName: ""
```

```
volumes: null
status: {}
---
apiVersion: v1
kind: Service
metadata:
creationTimestamp: null
name: nginx
namespace: '{{.NAMESPACE}}'
spec:
ports:
- name: tcp-80-80-ogxxh
nodePort: 0
port: 80
protocol: TCP
targetPort: 80
selector: {}
type: LoadBalancer
status:
loadBalancer: {}
```

在示例中，使用了 `NAMESPACE` 将 `namespace` 填写完之后，如下图所示：

模板名称

nginx

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

模板内容

模板可以通过从UI导入服务或新增空服务并手动编写来创建多个服务的YAML描述，详情可查看 [应用模板操作指引](#)

服务名	操作	内容
nginx	删除	<pre>1 apiVersion: extensions/v1beta1 2 kind: Deployment 3 metadata: 4 annotations: 5 description: nginx 6 creationTimestamp: null 7 name: nginx 8 namespace: '{{.NAMESPACE}}' 9 spec: 10 replicas: 1 11 revisionHistoryLimit: 5 12 selector: {} 13 strategy: 14 rollingUpdate: 15 maxSurge: 1 16 maxUnavailable: 0 17 type: RollingUpdate 18 template: 19 metadata:</pre>

[新增空服务](#) [从UI导入服务](#)

默认配置项

以上服务配置中，若存在自定义变量，可以手动输入为变量赋值，也可使用已有的配置项来为变量赋值，更多参见 [变量设置说明](#)

注：变量替换类型的配置项用于字符串替换，文件挂载类型的配置项用于将指定内容挂载到容器

变量名	变量值	类型
NAMESPACE	default	变量替换

[使用已有配置项赋值](#)

步骤三: 完成应用模板编辑，并查看

在步骤二中，完成了应用模板的编辑。单击 完成 按钮，保存应用模板。这样应用模板就创建完成，可以在应用模板列表查看。

接下来可以使用创建的模板，进行应用服务部署。关于如何使用应用模板进行应用部署可以参考[创建应用](#)。关于 nginx 这个应用模板具体部署应用的过程可以参考[应用模板示例-Nginx单服务应用](#)。

应用模板示例-guestbook应用

最近更新时间：2018-05-30 15:28:44

Guestbook留言板是一个比较典型的web应用服务，由一个frontend前端服务和redis-master和redis-slave两个后端存储服务组成。用户通过web前端提交数据，写入到redis-master上，然后通过读取同步到redis-slave上的数据展示给用户。

为了实现快速的在不同集群或者集群的不同namespace中部署Guestbook应用，可以先将Gustbook部署相关的配置保存到应用模板中，然后使用应用模板快速部署对应的应用。

本示例将介绍如何将创建Gustbook的应用模板。

步骤一: 创建应用模板

在[应用模板](#)列表中，单击新建按钮。



步骤二: 编辑应用模板

2.1 填写应用模板名称

云产品 常用服务

返回 新建模板

模板名称

guestbook

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

2.2 创建frontend服务

(1) 单击图中 新增空服务 按钮，新增一个服务。服务名称设置为frontend。

模板内容

模板可以通过从UI导入服务或新增空服务并手动编写来创建多个服务的YAML描述，详情可查看 [应用模板操作指引](#)

服务名	操作	内容
frontend	保存 取消	

新增空服务 从UI导入服务

(2) 在模板内容的编辑框中，填写 frontend 服务的模板内容。可以直接拷贝下面的内容到编辑框中。

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: frontend
  namespace: {{.NAMESPACE}}
spec:
  replicas: {{.FRONTEND_REPLICAS}}
```

```
template:
spec:
containers:
- env:
- name: GET_HOSTS_FROM
value: dns
image: ccr.ccs.tencentyun.com/library/gb-frontend:{{.FRONTEND_VERSION}}
imagePullPolicy: Always
name: php-redis
resources:
requests:
cpu: 100m
securityContext:
privileged: false
dnsPolicy: ClusterFirst
restartPolicy: Always
---
apiVersion: v1
kind: Service
metadata:
name: frontend
namespace: {{.NAMESPACE}}
spec:
ports:
- name: tcp-80-80-fwdy6
port: 80
protocol: TCP
targetPort: 80
sessionAffinity: None
type: LoadBalancer
```

2.3 创建redis-master服务

(1) 和 frontend 服务创建过程类似，新建一个 redis-master 服务。

(2) 填写 redis-master 服务的模板内容。

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
name: redis-master
namespace: {{.NAMESPACE}}
spec:
template:
spec:
containers:
```

```
- image: ccr.ccs.tencentyun.com/library/redis:{{.REDIS_MASTER_VERSION}}
imagePullPolicy: Always
name: master
resources:
limits:
memory: 100Mi
requests:
cpu: 100m
memory: 100Mi
securityContext:
privileged: false
dnsPolicy: ClusterFirst
restartPolicy: Always
---
```

apiVersion: v1

kind: Service

metadata:

name: redis-master

namespace: {{.NAMESPACE}}

spec:

ports:

- name: tcp-6379-6379-6d3d9

port: 6379

protocol: TCP

targetPort: 6379

sessionAffinity: None

type: ClusterIP

2.4 创建redis-slave服务

(1) 和 frontend 服务创建过程类似，新建一个 redis-slave 服务。

(2) 填写 redis-slave 服务的模板内容。

apiVersion: extensions/v1beta1

kind: Deployment

metadata:

name: redis-slave

namespace: {{.NAMESPACE}}

spec:

template:

spec:

containers:

- env:

- name: GET_HOSTS_FROM

value: dns

```
image: ccr.ccs.tencentyun.com/library/gb-redisslave :{{.REDIS_SLAVE_VERSION}}
imagePullPolicy: Always
name: slave
resources:
limits:
memory: 100Mi
requests:
cpu: 100m
memory: 100Mi
securityContext:
privileged: false
dnsPolicy: ClusterFirst
restartPolicy: Always
---
```

apiVersion: v1

kind: Service

metadata:

name: redis-slave

namespace: {{.NAMESPACE}}

spec:

externalName: ""

ports:

- name: tcp-6379-6379-av5fd

port: 6379

protocol: TCP

targetPort: 6379

sessionAffinity: None

type: ClusterIP

创建后的服务如下图所示：

模板内容

模板可以通过从UI导入服务或新增空服务并手动编写来创建多个服务的YAML描述，详情可查看 [应用模板操作指引](#)

服务名	操作	内容
frontend	删除	7 template:
redis-master	删除	8 spec:
redis-slave	删除	9 containers:
		10 - env:
		11 - name: GET_HOSTS_FROM
		12 value: dns
		13 image: ccr.ccs.tencentyun.com/library/gb-redisslave:{{.REDIS_SLAVE_VERSION}}
		14 imagePullPolicy: Always
		15 name: slave
		16 resources:
		17 limits:
		18 memory: 100Mi
		19 requests:
		20 cpu: 100m
		21 memory: 100Mi
		22 securityContext:
		23 privileged: false
		24 dnsPolicy: ClusterFirst
		25 restartPolicy: Always

[新增空服务](#) [从UI导入服务](#)

2.5 导出配置项，并填写配置项内容

在应用模板汇总使用了多个变量，需要在配置项中为变量设置默认值。具体的做法如下。
填写配置中配置项的内容。在本示例中，配置项的默认值如下。（可以根据需要进行修改）

NAMESPACE: default
FRONTEND_REPLICAS: 2
FRONTEND_VERSION: v4
REDIS_MASTER_VERSION: e2e
REDIS_SLAVE_VERSION: v1

默认配置项 以上服务配置中，若存在自定义变量，可以手动输入为变量赋值，也可使用已有的配置项来为变量赋值，更多参见 [变量设置说明](#)

注：变量替换类型的配置项用于字符串替换，文件挂载类型的配置项用于将指定内容挂载到容器

变量名	变量值	类型
NAMESPACE	default	变量替换
FRONTEND_REPLICAS	2	变量替换
FRONTEND_VERSION	v4	变量替换
REDIS_MASTER_VERSION	e2e	变量替换
REDIS_SLAVE_VERSION	v1	变量替换

[使用已有配置项赋值](#)

步骤三: 完成应用模板编辑，并查看

在步骤二中，完成了应用模板的编辑。单击 **完成** 按钮，保存应用模板。

这样应用模板就创建完成，可以在应用模板列表查看。

腾讯云

总览

云产品

常用服务

容器服务

应用模板

+ 新建

模板ID/模板名称	服务数量
tp-nwsmmlce guestbook	3

接下来可以使用创建的模板，进行应用服务部署。关于如何使用应用模板进行应用部署可以参考[创建应用](#)。关于 **Guestbook** 这个应用模板具体部署应用的过程可以参考应用[模板示例-Guestbook应用](#)。

创建应用模板

最近更新时间：2018-05-30 15:26:59

新建应用模板

登录 [腾讯云容器服务控制台](#)。在应用管理->[应用模板](#)页面单击新建



应用模板增加服务

应用模板中增加服务，是通过导入服务相对应的模板内容实现的。导入服务模板内容包括两种可选的方式: 1. 从控制台导入 2. YAML文件导入

注意：

可以根据场景选择使用其中的任意一种方式

更多关于导入模板内容的说明，可以参考[应用模板内容操作指引](#)

导入方法1：控制台导入服务：

单击导入服务按钮，弹出服务导入的控制台

< 返回

新建模板

模板名称

wordpress

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

模板内容

模板可以通过从UI导入服务或新增空服务并手动编写来创建多个服务的YAML描述，详情可查看 [应用模板操作指引](#)

服务名	操作	内容
记录为空		

新增空服务

从UI导入服务

导入服务

服务名称

mariadb

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

服务描述

mariadb服务

描述信息不超过1000个字符

数据卷(选填)

使用云硬盘

vol

将生成变量

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、配置项，还需挂载到容器的指定路径中。[使用指引](#)

运行容器

名称

mariadb

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

镜像

mariadb

选择镜像

版权所有：腾讯云计算（北京）有限责任公司

第74 共105页

导入后结果：

模板内容

模板可以通过从UI导入服务或新增空服务并手动编写来创建多个服务的YAML描述，详情可查看 [应用模板操作指引](#)

服务名	操作	内容
mariadb	删除	<pre>37 cbsDiskId: '{{.ReleaseCBS_mariadb_vol}}' 38 fsType: ext4 39 status: {} 40 --- 41 apiVersion: v1 42 kind: Service 43 metadata: 44 creationTimestamp: null 45 name: mariadb 46 namespace: '{{.NAMESPACE}}' 47 spec: 48 ports: 49 - name: tcp-3306-3306-nh6kj 50 nodePort: 0 51 port: 3306 52 protocol: TCP 53 targetPort: 3306 54 selector: {} 55 type: ClusterIP</pre>

[新增空服务](#) [从UI导入服务](#)

默认配置项

以上服务配置中，若存在自定义变量，可以手动输入为变量赋值，也可使用已有的配置项来为变量赋值，更多参见 [变量设置说明](#)

注: 变量替换类型的配置项用于字符串替换，文件挂载类型的配置项用于将指定内容挂载到容器

变量名	变量值	类型
NAMESPACE	default	变量替换
ROOT_PASSWORD_VALUE		变量替换
ReleaseCBS_mariadb_vol		变量替换

[使用已有配置项赋值](#)

在导入服务控制台，填写对应的参数后，单击确认，则自动生成服务的模板内容。更多关于控制台导入模板内容的说明，可以参考[应用模板内容操作指引-控制台导入服务](#)。

导入方法2：YAML文件导入服务：

单击服务添加按钮，添加服务

模板名称

wordpress

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

模板内容

模板可以通过从UI导入服务或新增空服务并手动编写来创建多个服务的YAML描述，详情可查看 [应用模板操作指引](#)

服务名	操作	内容
mariadb	保存 取消	




[新增空服务](#) [从UI导入服务](#)

将在服务编辑中，编辑服务的内容。可以直接拷贝模板内容到服务编辑框。下面是一个nginx示例的模板内容。更多关于YAML文件导入模板内容的说明，可以参考[应用模板内容操作指引-YAML文件导入模板内容](#)。

apiVersion: extensions/v1beta1**kind: Deployment****metadata:**

annotations:

description: nginx服务

creationTimestamp: null

name: nginx

namespace: '{{.NAMESPACE}}'

spec:

replicas: 1

revisionHistoryLimit: 5

selector: {}

strategy: {}

template:

metadata:

creationTimestamp: null

spec:

containers:

- image: nginx:latest

```
imagePullPolicy: Always
name: nginx
resources:
requests:
cpu: 200m
securityContext:
privileged: false
serviceAccountName: ""
volumes: null
status: {}
---
apiVersion: v1
kind: Service
metadata:
creationTimestamp: null
name: nginx
namespace: '{{.NAMESPACE}}'
spec:
ports:
- name: tcp-80-80-ai6zd
nodePort: 0
port: 80
protocol: TCP
targetPort: 80
selector: {}
type: LoadBalancer
status:
loadBalancer: {}
```

应用模板内容编辑

在应用模板内容的编辑框中，可以编辑和修改对应服务的内容。例如：可以在模板内容中修改镜像的版本，从 "latest" 修改为 "stable" (镜像的版本从最新版修改为稳定版)。

模板名称

nginx

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

模板内容

模板可以通过从UI导入服务或新增空服务并手动编写来创建多个服务的YAML描述，详情可查看 [应用模板操作指引](#)

服务名	操作	内容
nginx	删除	<pre>14 rollingUpdate: 15 maxSurge: 1 16 maxUnavailable: 0 17 type: RollingUpdate 18 template: 19 metadata: 20 creationTimestamp: null 21 spec: 22 containers: 23 - image: nginx:stable ###nginx:latest 24 imagePullPolicy: Always 25 name: nginx 26 resources: 27 requests: 28 cpu: 200m 29 securityContext: 30 privileged: false 31 volumeMounts: 32 - mountPath: /data/config</pre>

[新增空服务](#) [从UI导入服务](#)

模板内容满足一定的编排语法规则，更多关于语法规则的说明可以参考[编排语法能力说明](#)。

应用模板配置项设置

在模板内容中，如果有些参数在不同环境下是不相同的，或者需要进行频繁的修改。可以将这部分参数转换成一个可配置的变量。例如在上面的示例中，可以将镜像版本设置为变量 `IMAGE_VERSION`，在配置项中设置 `IMAGE_VERSION` 的值为"stable"。

具体的操作步骤如下：

步骤一：编辑模板内容，将参数用变量替换

如下图所示将模板内容中镜像的版本这个参数使用 `IMAGE_VERSION` 变量替换。变量形式为"`{{.xxxx}}`"。更多关于变量设置的说明，可以参考[变量设置](#)。

模板名称

nginx

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

模板内容

模板可以通过从UI导入服务或新增空服务并手动编写来创建多个服务的YAML描述，详情可查看 [应用模板操作指引](#)

服务名	操作	内容
nginx	删除	<pre>14 rollingUpdate: 15 maxSurge: 1 16 maxUnavailable: 0 17 type: RollingUpdate 18 template: 19 metadata: 20 creationTimestamp: null 21 spec: 22 containers: 23 - image: nginx:{{.IMAGE_VERSION}} 24 imagePullPolicy: Always 25 name: nginx 26 resources: 27 requests: 28 cpu: 200m 29 securityContext: 30 privileged: false 31 volumeMounts: 32 - mountPath: /data/config</pre>

新增空服务 从UI导入服务

默认配置项

以上服务配置中，若存在自定义变量，可以手动输入为变量赋值，也可使用已有的配置项来为变量赋值，更多参见 [变量设置说明](#)

注：变量替换类型的配置项用于字符串替换，文件挂载类型的配置项用于将指定内容挂载到容器

变量名	变量值	类型
NAMESPACE	default	变量替换
IMAGE_VERSION	stable	变量替换
abac-authz-policy.jsonl	a=b	文件挂载

步骤二：设置配置项中变量的值

在配置项中设置 `IMAGE_VERSION` 的值为"stable"。这样在使用应用模板部署应用时，如果使用应用模板中的默认配置项，则会在部署时使用"stable"作为镜像的版本。当然也可以根据不同环境的需求，在配置项中将 `IMAGE_VERSION` 变量设置成不同的值，从而实现在不同环境下的使用不同的镜像版本进行部署。

应用模板保存

在写好应用模板的内容和默认配置项后，可以单击 完成 按钮，保存新创建的应用模板。

查看创建的模板

在[应用模板](#)列表页面，可查看到刚创建的应用模板。

容器服务	应用模板	
	+ 新建	
	模板ID / 模板名称	服务数量
	tp-7xq0950i wordpress	2
	tp-el5uirgo singlenginx	1
	tp-fdbnzixs nginx	1
	tp-eh6ia6js php-nginx	1
	tp-mev3xoy2 elk-devin	4
	tp-5luxeyx8 d2n48	1

删除应用模板

最近更新时间：2018-05-30 15:27:42

删除应用模板

登录 [腾讯云容器服务控制台](#)。在应用管理->[应用模板](#)页面查看应用模板列表。单击右侧的删除按钮，删除指定的应用模板。

注意：

删除应用模板并不会影响已经使用该应用模板部署的应用。

腾讯云

总览

云产品

常用服务

容器服务

应用中心

集群

应用

服务

Ingress

交付中心

镜像仓库

应用模板仓库

我的模板

模板市场

应用模板

+ 新建

请输入模板名称

模板ID/模板名称	服务数量	创建时间	更新时间	操作
tp-7xq0950i wordpress	2	2018-02-06 12:18:12	2018-02-06 12:18:12	更新模板 创建应用 删除
tp-el5uirgo singlenginx	1	2018-01-31 11:22:29	2018-01-31 19:58:32	更新模板 创建应用 删除
tp-fdbnzixs nginx	1	2018-01-31 11:14:15	2018-01-31 11:14:15	更新模板 创建应用 删除
tp-eh6ia6js php-nginx	1	2017-12-23 22:00:21	2017-12-27 15:38:44	更新模板 创建应用 删除
tp-mev3xoy2 elk-devin	4	2017-12-20 20:49:54	2017-12-20 21:09:59	更新模板 创建应用 删除
tp-51uxeyx8 d2048	1	2017-12-20 16:44:15	2017-12-20 16:44:15	更新模板 创建应用 删除

更新应用模板

最近更新时间：2018-05-30 15:30:57

应用模板列表

登录 [腾讯云容器服务控制台](#)。在应用管理->[应用模板](#)页面可以查看该用户下所有的应用模板。

应用模板					
+ 新建		请输入关键字			
模板ID/模板名称	服务数量	创建时间	更新时间	操作	
tp-5in7cs3z nginxapp	1	2017-10-18 17:59:30	2017-10-18 19:57:59	更新模板 删除	
tp-bjwapmlf nginxapp	1	2017-10-18 16:19:24	2017-10-18 16:19:24	更新模板 删除	
tp-nsf4s7vx test	1	2017-10-16 16:23:42	2017-10-16 16:23:42	更新模板 删除	
tp-2z4q6u8f apache	2	2017-10-16 12:05:34	2017-10-16 12:22:04	更新模板 删除	
tp-mer84m17 apachcbs	2	2017-10-15 17:25:22	2017-10-15 17:40:12	更新模板 删除	
tp-78yat52d pythonconfig	1	2017-10-15 17:14:59	2017-10-15 17:14:59	更新模板 删除	
tp-4j1bli3 test111	1	2017-10-15 16:26:21	2017-10-15 16:26:21	更新模板 删除	
tp-p3xcuah1 abc1	0	2017-10-13 11:05:57	2017-10-13 11:11:55	更新模板 删除	
tp-rzkwf82x tstapache	1	2017-10-12 13:15:00	2017-10-12 13:42:40	更新模板 删除	
tp-c35dl4dj testservice	1	2017-10-12 11:20:45	2017-10-12 11:20:45	更新模板 删除	
tp-2z3jqcsv frontend	1	2017-10-11 19:49:41	2017-10-11 19:49:41	更新模板 删除	

应用模板过滤搜索

在应用模板列表搜索框中，输入模板名称可以过滤搜索出包含特定名称的模板。例如：在搜索框中输入 `nginx`，过滤出名称中包含 `nginx` 的模板。

应用模板

+ 新建

过滤出特定的应用模板

nginx

模板ID/模板名称	服务数量	创建时间	更新时间	操作
搜索 "nginx" 找到 2 条结果, 返回原列表				
tp-5ln7cs3z nginxapp	1	2017-10-18 17:59:30	2017-10-18 19:57:59	更新模板 删除
tp-bjwapmlf nginxapp	1	2017-10-18 16:19:24	2017-10-18 16:19:24	更新模板 删除

应用模板详情

在应用模板页面单击应用模板ID，可以查看应用模板详情。

应用模板

+ 新建

点击模板Id，查看模板详情

请输入关键字

模板ID/模板名称	服务数量	创建时间	更新时间	操作
tp-5ln7cs3z nginxapp	1	2017-10-18 17:59:30	2017-10-18 19:57:59	更新模板 删除
tp-bjwapmlf nginxapp	1	2017-10-18 16:19:24	2017-10-18 16:19:24	更新模板 删除
tp-nsf4s7vx test	1	2017-10-16 16:23:42	2017-10-16 16:23:42	更新模板 删除
tp-2z4q6u8f apache	2	2017-10-16 12:05:34	2017-10-16 12:22:04	更新模板 删除
tp-mer84m17 apachcbs	2	2017-10-15 17:25:22	2017-10-15 17:40:12	更新模板 删除
tp-78yat52d pythonconfig	1	2017-10-15 17:14:59	2017-10-15 17:14:59	更新模板 删除
tp-4jl1lij3 test111	1	2017-10-15 16:26:21	2017-10-15 16:26:21	更新模板 删除
tp-p3xcuah1 abc1	0	2017-10-13 11:05:57	2017-10-13 11:11:55	更新模板 删除

应用模板详情中分为模板内容部分和配置项部分。其中模板内容包括每个服务的YAML格式的描述文本。配置项部分展示模板默认的配置项内容，也为YAML格式。

[返回](#) | tp-5in7cs3z 详情

模板名称 nginxapp

模板内容

nginx

```
1 apiVersion: extensions/v1beta1
2 kind: Deployment
3 metadata:
4   annotations:
5     description: 更新模板示例
6   creationTimestamp: null
7   name: nginx
8   namespace: '{{.NAMESPACE}}'
9 spec:
10  replicas: 2
11  revisionHistoryLimit: 5
12  selector: {}
13  strategy: {}
14  template:
15    metadata:
16      creationTimestamp: null
17    spec:
18      containers:
19        - image: nginx:latest
20          imagePullPolicy: Always
21          name: nginx
```

配置项

```
1 NAMESPACE: default
2
```

查看应用模板

最近更新时间：2018-05-30 15:31:47

应用模板列表

登录 [腾讯云容器服务控制台](#)。在应用管理->[应用模板](#)页面可以该用户下所有的应用模板。

应用模板					
+ 新建		请输入关键字			
模板ID/模板名称	服务数量	创建时间	更新时间	操作	
tp-5in7cs3z nginxapp	1	2017-10-18 17:59:30	2017-10-18 19:57:59	更新模板 删除	
tp-bjwapmlf nginxapp	1	2017-10-18 16:19:24	2017-10-18 16:19:24	更新模板 删除	
tp-nsf4s7vx test	1	2017-10-16 16:23:42	2017-10-16 16:23:42	更新模板 删除	
tp-2z4q6u8f apache	2	2017-10-16 12:05:34	2017-10-16 12:22:04	更新模板 删除	
tp-mer84m17 apachcbs	2	2017-10-15 17:25:22	2017-10-15 17:40:12	更新模板 删除	
tp-78yat52d pythonconfig	1	2017-10-15 17:14:59	2017-10-15 17:14:59	更新模板 删除	
tp-4j1bli3 test111	1	2017-10-15 16:26:21	2017-10-15 16:26:21	更新模板 删除	
tp-p3xcuah1 abc1	0	2017-10-13 11:05:57	2017-10-13 11:11:55	更新模板 删除	
tp-rzkwf82x tstapache	1	2017-10-12 13:15:00	2017-10-12 13:42:40	更新模板 删除	
tp-c35dl4dj testservice	1	2017-10-12 11:20:45	2017-10-12 11:20:45	更新模板 删除	
tp-2z3jqcsv frontend	1	2017-10-11 19:49:41	2017-10-11 19:49:41	更新模板 删除	

应用模板过滤搜索

在应用模板列表搜索框中，输入模板名称可以过滤搜索出包含特定名称的模板。

应用模板

+ 新建

过滤出特定的应用模板

nginx

模板ID/模板名称	服务数量	创建时间	更新时间	操作
搜索 "nginx" 找到 2 条结果, 返回原列表				
tp-5ln7cs3z nginxapp	1	2017-10-18 17:59:30	2017-10-18 19:57:59	更新模板 删除
tp-bjwapmlf nginxapp	1	2017-10-18 16:19:24	2017-10-18 16:19:24	更新模板 删除

应用模板详情

在应用模板页面单击应用模板ID，可以查看应用模板详情。

应用模板

+ 新建

请输入关键字

点击模板Id，查看模板详情

模板ID/模板名称	服务数量	创建时间	更新时间	操作
tp-5ln7cs3z nginxapp	1	2017-10-18 17:59:30	2017-10-18 19:57:59	更新模板 删除
tp-bjwapmlf nginxapp	1	2017-10-18 16:19:24	2017-10-18 16:19:24	更新模板 删除
tp-nsf4s7vx test	1	2017-10-16 16:23:42	2017-10-16 16:23:42	更新模板 删除
tp-2z4q6u8f apache	2	2017-10-16 12:05:34	2017-10-16 12:22:04	更新模板 删除
tp-mer84m17 apachcbs	2	2017-10-15 17:25:22	2017-10-15 17:40:12	更新模板 删除
tp-78yat52d pythonconfig	1	2017-10-15 17:14:59	2017-10-15 17:14:59	更新模板 删除
tp-4j1blij3 test111	1	2017-10-15 16:26:21	2017-10-15 16:26:21	更新模板 删除
tp-p3xcuah1 abc1	0	2017-10-13 11:05:57	2017-10-13 11:11:55	更新模板 删除

应用模板详情展示如下，分为模板内容部分和配置项部分。

[返回](#) | tp-5in7cs3z 详情

模板名称 nginxapp

模板内容

nginx

```
1 apiVersion: extensions/v1beta1
2 kind: Deployment
3 metadata:
4   annotations:
5     description: 更新模板示例
6   creationTimestamp: null
7   name: nginx
8   namespace: '{{.NAMESPACE}}'
9 spec:
10  replicas: 2
11  revisionHistoryLimit: 5
12  selector: {}
13  strategy: {}
14  template:
15    metadata:
16      creationTimestamp: null
17    spec:
18      containers:
19        - image: nginx:latest
20          imagePullPolicy: Always
21          name: nginx
```

配置项

```
1 NAMESPACE: default
2
```

语法说明 - 基础语法

最近更新时间：2018-10-11 16:10:53

腾讯云容器服务底层基于Kubernetes编排引擎。在容器服务的应用模板中原生的支持Kubernetes的语法。本文将对Kubernetes的常用的语法做介绍。

CPU/Memory限制设置

Kubernetes使用Limit和Request对容器使用的资源进行限制。更多详细内容可以参数[官方文档](#)。

对于CPU和内存的设置，示例如下：

```
resources:
  limits:
    memory: 128Mi
    cpu: 0
  requests:
    cpu: 200m
    memory: 128Mi
```

在上面的示例中，通过设置resources中limits和requests参数中CPU和Memory的值，来限制容器CPU和memory的使用。上面的示例中，CPU在调度时被分配的CPU为0.2核，CPU限制使用的最大量无上限。内存在调度时被分配的128M，内存最大使用量被限制为128M。

在CPU资源的限制中，可以使用单位 m, 1m=0.001核。在内存的限制中，可以使用M, Mi, G, Gi。

1M=1000 1000Byte, 1Mi=1024 1024Byte, 1G=1000 1000 1000Byte, 1Gi=1024 1024 1024Byte。

命名空间设置

Kubernetes使用namespace参数来设置服务所在的命名空间。例如设置命名空间为wordpress

```
namespace: wordpress
```

备注：

- 1、在同一个服务不同资源下，命名空间必须设置成一样。同一个应用中不同服务，也必须设置成一样。
- 2、如果namespace参数不填写，会默认使用default命名空间。

关于更多关于namespace的作用可以参考[Namespace使用指引](#)

实例数设置

Kubernetes使用replicas参数来设置实例的数量，如下所示:

```
spec:  
replicas: 2
```

通过设置replicas参数，把容器实例数设置为2。

镜像参数设置

Kubernetes使用image参数来设置容器使用的镜像。指定使用容器镜像busybox，版本号为latest示例如下：

```
image: busybox:latest
```

备注：如果不指定镜像的版本，会默认使用latest作为镜像的版本

服务类型设置

Kubernetes支持多种不同的服务访问类型，常用的服务访问类型包括: 负载均衡访问，集群内访问，节点端口访问，不设置访问方式。对于负载均衡访问，在容器服务中又分为外网负载均衡访问和VPC内负载均衡访问。Kubernetes通过Service资源中的 `type` 参数来指定服务的访问类型。

外网负载均衡访问

```
apiVersion: v1  
kind: Service  
metadata:  
labels:  
qcloud-app: nginx  
name: nginx  
spec:  
ports:  
- name: tcp-80-80-pfbp1  
port: 80  
protocol: TCP  
targetPort: 80  
selector:  
qcloud-app: nginx  
type: LoadBalancer
```

如上面的例子所示，通过设置 `type` 为 `LoadBalancer` 将服务的访问类型指定为外网负载均衡访问。

VPC内负载均衡访问

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    service.kubernetes.io/qcloud-loadbalancer-internal-subnetid: 'subnet-xxxxxx'
  labels:
    qcloud-app: nginx
  name: nginx
spec:
  ports:
    - name: tcp-80-80-pfbp1
      port: 80
      protocol: TCP
      targetPort: 80
  selector:
    qcloud-app: nginx
  type: LoadBalancer
```

VPC内访问的负载均衡，除了需要将通过设置 `type` 为 `LoadBalancer`，还需要设置一个 `service.kubernetes.io/qcloud-loadbalancer-internal-subnetid` 的 `annotations`，其值设置成负载均衡IP所在的子网(该子网必须在容器集群节点所在vpc内)。

集群内访问

```
apiVersion: v1
kind: Service
metadata:
  labels:
    qcloud-app: nginx
  name: nginx
spec:
  ports:
    - name: tcp-80-80-tlap9
      nodePort: 0
      port: 80
      protocol: TCP
      targetPort: 80
  selector:
    qcloud-app: nginx
  type: ClusterIP
```

如上面的例子所示，通过设置 `type` 为ClusterIP将服务的访问类型指定为集群内通过ClusterIP访问。对于外网负载均衡和内网负载均衡也会默认分配ClusterIP，也可以通过ClusterIP在集群内访问。

不设定访问方式

如果服务中没有 `Service` 资源，则不设定服务的访问方式。

备注：容器服务中暂时未开放节点端口的这种访问方式，在外网负载均衡访问，VPC内负载均衡访问和集群内访问三种访问方式中都会默认为服务分配节点端口。在Ingress中也会通过注册节点端口，来注册服务访问，如果Ingress中配置了对该服务的访问，将服务访问方式设定为"不设定访问方式"会到时候服务不可用。

更多关于服务访问方式的说明可以参考[kubernetes 官方文档](#)，也可以参考[服务访问方式设置](#)

磁盘挂载设置

Kubernetes支持挂载不类型的目录或者文件到容器指定的目录。其中涉及到两个步骤：挂载磁盘设置和挂载点设置。

例如挂载一个主机上的/home目录到容器中：

(1) 设置挂载的磁盘/home目录

```
volumes:
- hostPath:
  path: /home
  name: vol
```

(2) 将磁盘挂载到mnt目录

```
volumeMounts:
- mountPath: /mnt
  name: vol
```

详细的关于磁盘挂载的说明，可以参考[Kubernetes官方文档-volumes](#)

容器平台现还支持挂载CBS盘，NFS，和配置文件。对于不同类型磁盘的挂载，可以参考。

[扩展语法--CBS盘的使用](#)

[自定义变量--ReleaseConfig](#)

多容器服务

Kubernetes中支持在一个实例中包含多个容器，例如下面服务中一个实例包含一个nginx容器和一个busybox容器。

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    qcloud-app: nn
    name: nn
  namespace: default
spec:
  replicas: 1
  revisionHistoryLimit: 5
  selector:
    matchLabels:
      qcloud-app: nn
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        qcloud-app: nn
    spec:
      containers:
        - image: nginx:latest
          imagePullPolicy: Always
          name: nginx
          resources:
            requests:
              cpu: 200m
          securityContext:
            privileged: false
        - args:
            - "36000"
          command:
            - sleep
          image: busybox:latest
          imagePullPolicy: Always
          name: busybox
          resources:
            requests:
              cpu: 200m
          securityContext:
            privileged: false
```

```
serviceAccountName: ""  
volumes: null  
status: {}
```

如上面的例子所示，在 `containers` 参数中设置不同的容器信息，可以使服务实例包含多个容器。

Kubernetes基础编排语法

Kubernetes通过定义API结构体中的每一个参数，来定义编排的描述语言。关于Kubernetes API参数的详细说明可以参考：

更多关于Kubernetes编排描述语言的说明，可以参考[API Overview](#)

语法说明 - 扩展语法

最近更新时间：2018-03-15 21:15:00

为了更好的使用Kubernetes编排引擎，并使得Kubernetes编排引擎和腾讯云IaaS层的基础服务更好的结合。腾讯云容器服务在Kubernetes原生的编排语法的基础上，对编排的语法进行了扩展。下面将对扩展的语法进行介绍。

扩展语法---cbs盘使用

容器服务支持直接在服务实例上挂载CBS盘。具体的做法如下面的示例所示：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    qcloud-app: cbs
  name: cbs
  namespace: default
spec:
  replicas: 1
  revisionHistoryLimit: 5
  selector:
    matchLabels:
      qcloud-app: cbs
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
    labels:
      qcloud-app: cbs
    spec:
      containers:
        - args:
            - "360000"
          command:
            - sleep
          image: busybox:latest
          imagePullPolicy: Always
          name: busybox
          resources:
            requests:
```

```
cpu: 200m
securityContext:
privileged: true
volumeMounts:
- mountPath: /mnt
name: vol
serviceAccountName: ""
volumes:
- name: vol
qcloudCbs:
cbsDiskId: 'disk_xxxxxxx'
fsType: ext4
```

在volume盘指定时，指定类型为qcloudCbs，并设置对应的cbsDiskId为需要挂载盘的instanceId。(Cbs盘的instanceId可以在[云硬盘页面](#)查看)。挂载点位置设定与其他类型的磁盘设置一样，如上例所示，将cbs盘挂载到容器的/mnt目录。

备注：由于一块cbs盘只能同时被挂载一次，所以如果在服务中使用了cbs盘则容器实例不能大于1，且不支持滚动更新。

扩展语法---VPC内负载均衡访问

在基础语法中，通过设置 Service 中的 type 字段，能够设置服务的访问方式。在容器服务中定义 annotations 中 service.kubernetes.io/qcloud-loadbalancer-internal-subnetid 来区分是VPC内网访问负载均衡还是外网访问服务均衡。

如果设置了 service.kubernetes.io/qcloud-loadbalancer-internal-subnetid 这个 annotations 则认为是创建VPC内访问的负载均衡器，否则则认为是创建外网的负载均衡器。

具体示例如下所示：

```
apiVersion: v1
kind: Service
metadata:
annotations:
service.kubernetes.io/qcloud-loadbalancer-internal-subnetid: 'subnet-xxxxxx'
labels:
qcloud-app: nginx
name: nginx
spec:
ports:
- name: tcp-80-80-pfbp1
port: 80
```

```
protocol: TCP
targetPort: 80
selector:
qcloud-app: nginx
type: LoadBalancer
```

其中 `subnet-xxxxxx` 为集群节点所在VPC内的一个子网。

语法说明 - 变量设置

最近更新时间：2018-03-15 21:15:22

变量替换说明

应用模板中支持变量替换，变量的结构满足 `{{.}}`，在 `.` 后面加上对应的变量名。需要被替换的变量在模板解析时会被配置文件中配置项的值替换。例如在模板中定义 `FRONTEND_REPLICAS`：

```
spec:
  replicas: {{.FRONTEND_REPLICAS}}
```

在配置项中对变量 `FRONTEND_REPLICAS` 设定值：

```
FRONTEND_REPLICAS: 2
```

这样在模板解析时，模板文件中的 `{{.FRONTEND_REPLICAS}}` 会被替换成配置文件中的值 "2"

备注：变量名称满足正则表达式 `"[A-Za-z][A-Za-z0-9]*"`，最大长度为64个字符

自定义变量--ReleaseCBS

在容器服务中，如果需要挂载 cbs 盘，指定的描述语言如下所示：

```
volumes:
- name: vol
qcloudCbs:
  cbsDiskId: 'disk-pr47vtvt'
  fsType: ext4

volumeMounts:
- mountPath: /mnt
  name: vol
```

(上面这段描述语言表示：将 CBS 盘 `disk-pr47vtvt` 作为 `vol` 盘挂载，具体的挂载到容器的 `/mnt` 目录。)

由于 cbs 盘是不能重复挂载的，通过应用模板在不同环境部署应用时，需要指定不同的 cbs。所以容器服务提供 `ReleaseCBS` 变量来表示 CBS 盘。在应用部署时，会选择 **应用可使用的** 特定的 CBS 盘进行部署。具体示例如下

```
volumes:
- name: vol
qcloudCbs:
```

```
cbsDiskId: '{{ReleaseCBS_pr47vtvt}}'  
fsType: ext4
```

[补充cbs应用部署时cbs盘选择截图]

自定义变量--ReleaseSubnetId

在容器服务中，如果服务的访问方式选择为VPC内访问，则需要在描述文件中指定Lb对应的SubnetId，指定方式为，在service的描述文件中定义如下结果，其中subnet-s1jz1ycx为设定的SubnetId。

```
apiVersion: v1  
kind: Service  
metadata:  
  annotations:  
    service.kubernetes.io/qcloud-loadbalancer-internal-subnetid: 'subnet-s1jz1ycx'
```

由于SubnetId需要在集群所在的VPC(私有网络)下，所以SubnetId在不同集群中很有可能是不相同的。为了方便在应用部署时设置SubnetId，我们在模板中通过ReleaseSubNetId变量对需要设置的subnetId进行标识。下面是一个具体的示例：

```
apiVersion: v1  
kind: Service  
metadata:  
  annotations:  
    service.kubernetes.io/qcloud-loadbalancer-internal-subnetid: '{{ReleaseSubnetId_XXXX}}'
```

在应用部署时，将在配置文件中为ReleaseSubnetId_XXXX变量指定对应的subnetId。

备注：由于应用中的多个服务可能会设置不同的subnetId，所有在ReleaseSubnetId名称后通过"_"(下划线)和服务名称来区分不同服务的subnetId。

自定义变量--ReleaseConfig

配置文件是程序运行很重要的一个部分，很多程序往往需要从磁盘的某个位置读取对应的配置文件，于是Kubernetes支持将通过confimap中的某个key挂载到容器指定的目录。更多关于Kubernetes configmap的说明可以参考[Kubernetes官方文档](#)。

应用中包含一个属于应用的配置文件，通过自定义变量ReleaseConfig可以将应用的配置文件中key挂载到容器指定目录。

具体的使用示例如下：

```
volumes:
- configMap:
  Name: '{{ReleaseConfig}}'
  items:
  - key: NAMESPACE
  mode: 511
  path: NAMESPACE
  - key: FRONTEND_REPLICAS
  mode: 511
  path: FRONTEND_REPLICAS
  name: data
```

挂载点的设置为：

```
volumeMounts:
- mountPath: /mnt
  name: data
```

应用的配置信息为：

```
NAMESPACE: default
FRONTEND_REPLICAS: 2
```

这样，配置文件中的key：NAMESPACE和FRONTEND_REPLICAS会被挂载到容器的/mnt目录，挂载的文件名称分别为NAMESPACE和FRONTEND_REPLICAS。文件的内容为配置文件中相应的key所对应的内容

语法说明 - 限制说明

最近更新时间：2018-03-15 21:15:50

出于功能稳定闭环的考虑，容器平台对kubernetes的编排语法做了一定的限制。本文档将介绍这些限制的内容和原因。

支持的资源类型限制

Kubernetes包含多种不同资源类型。容器服务根据用户的需求，在UI界面上暂时只开放了 Deployment 和 Service 这两种最常用资源。（在创建 Deployment 会自动创建 Pod 资源和 replicaset ）。

所有在容器服务的应用编排中，暂时只支持 Deployment 和 Service 这两种资源。

资源名称限制

应用模板的每个服务内， Deployment 和 Service 资源名称必须和服务名称保持一致。

命名空间限制

- (1) 应用模板中所有资源必须在一个命名空间内。
- (2) kube-system 命名空间暂时不支持创建服务，所以在应用模板中暂时不支持命名空间为 kube-system 。

Label标签限制

- (1) 在应用模板中，可以使用Label标签对服务中的资源进行标记。在容器服务中， Service 通过 Select Label 标签去寻找对应的 Pod ,从而实现与 Deployment 的关联。 Deployment 关联 Pod 使用的是 Deployment 自身的 Select Label 。所以为了实现 Deployment 和 Service 的管理，增加了 Deployment 和 Service 的 Select Label 必须一致的限制。
- (2) 在应用模板中默认为每一个服务提供 qcloud-app 进行标识，提供 qcloud-application-label 标签标记属于哪个应用。这两个标签暂时不支持修改。

Cbs盘使用限制

-
- (1) 由于cbs只能同时被一个容器实例挂载，所有在使用了cbs盘的服务，实例数最大为1
 - (2) 使用了cbs盘的服务，更新操作只能使用重新创建更新，暂时不支持滚动更新。

应用模板内容操作指引

最近更新时间：2018-05-30 15:35:06

在应用模板(或者应用)中，通过YAML格式的描述性语言将服务的部署信息保存成模板内容，从而实现多次复用和多环境下部署不同的应用。可以在模板内容区域导入和编辑服务的模板内容。

模板内容支持两种方式导入：1. UI导入服务 2. 新增空服务

注意：

可以根据场景选择使用其中的任意一种方式

UI导入服务可以通过在控制台填写服务的相关参数，自动转换成模板内容的YAML描述文件，适用于YAML描述文件不存在的情况。**新增空服务**可以直接拷贝已经存在的YAML描述文件内容到模板内容区域，适用于YAML描述文件已经存在的情况。

UI导入服务的模板内容

在新建模板(或者新建应用)页面，单击 **UI导入服务** 按钮，可以在控制台填写对应的参数，自动转换成服务的模板内容。

服务导入页面参数设置，与服务创建页面参数设置保持一致。

主要参数包括：服务的基本信息，数据卷设置，运行容器参数设置，实例数量设置，服务访问类型和端口等。

例如在示例中部署一个nginx服务，填写的参数如下：

设置服务的基本信息：

1. 填写服务名称 **nginx**
2. 填写服务描述 **nginx服务**

设置服务的数据卷信息：

未使用数据盘，无。

设置镜像参数：

1. 在设置容器运行参数中的镜像参数：
容器名称设置为 **nginx**
镜像名称设置为 **nginx**
版本号选择为 **latest**

设置服务的实例数：

1. 服务的实例数设置为1

设置服务的访问方式：

1. 服务的访问方式设置为集群内访问
2. 服务的访问端口：容器端口和服务端口都设置成80

更多关于参数设置内容可以参考 [服务创建](#)操作的相关文档。

填写参数后，单击导入服务的 **完成** 按钮，控制台自动导入服务的模板内容。

新增空服务导入模板内容

如果已经存在服务对应的模板内容的YAML文件，可以直接将模板内容导入到编辑框中。具体的步骤如下：

步骤一 创建对应的服务：

单击"新增空服务"按钮，新增一个服务。

步骤二 导入模板内容：

可以将下面YAML文件中的内容，直接拷贝到编辑框中，导入服务的模板内容。

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  annotations:
  description: nginx服务
  creationTimestamp: null
  name: nginx
  namespace: default
spec:
  replicas: 1
  revisionHistoryLimit: 5
  selector: {}
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
    spec:
      containers:
      - image: nginx:latest
        imagePullPolicy: Always
```

```
name: nginx
resources:
  requests:
    cpu: 200m
  securityContext:
    privileged: false
  serviceAccountName: ""
  volumes: null
status: {}
```

```
---
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: null
  name: nginx
  namespace: default
spec:
  ports:
    - name: tcp-80-80-xoq5o
      nodePort: 0
      port: 80
      protocol: TCP
      targetPort: 80
  selector: {}
  type: LoadBalancer
status:
  loadBalancer: {}
```

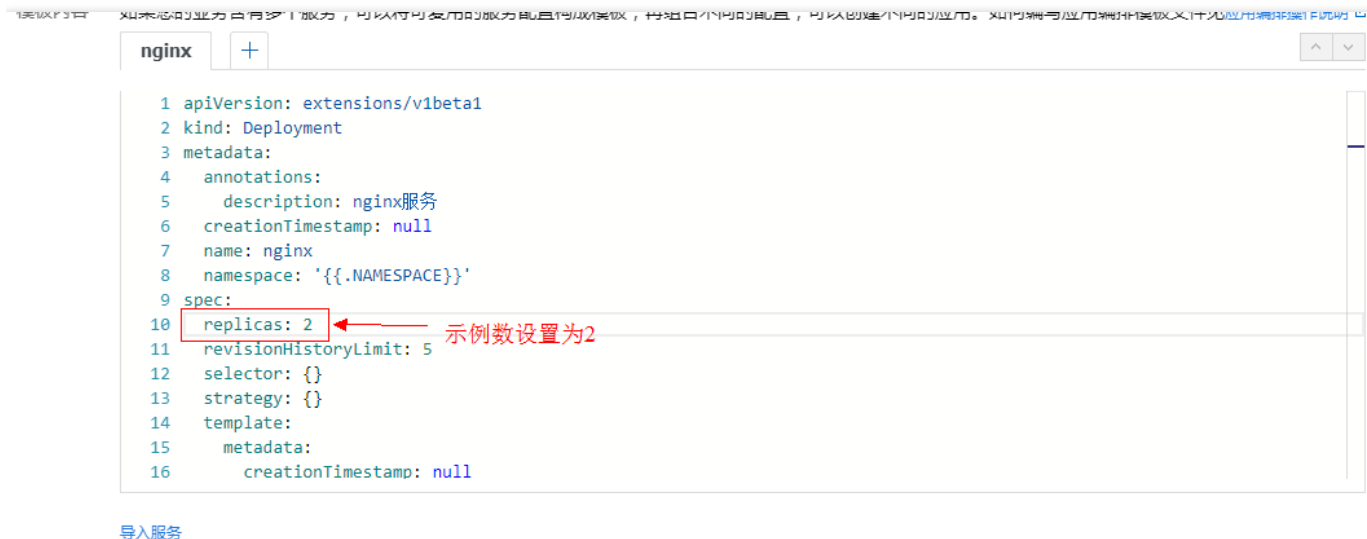
步骤三 提取模板内容中的变量：

在示例中，使用了 `NAMESPACE` 将 `namespace` 参数作为变量进行替换，更多关于变量替换的说明可以参考[变量替换](#)。

系统自动提取模板内容中的变量作为配置项。然后在配置项中填写变量的默认值，这里设置成 `default`。

编辑模板内容

在模板内容区域，可以直接对模板内容进行编辑。例如在模板内容中将服务实例的副本数修改为 `2`。



模板内容的语法规则，满足Kubernetes定义的语法，更多关于模板内容规则的说明可以参考[编排语法能力说明](#)。

模板内容中参数配置化

在模板内容中，如果有些参数在不同环境下是不相同的，或者需要进行频繁的修改。可以将这部分参数转换成一个可配置的变量。例如，上面的服务实例副本数，可能在不同环境下是不相同的，我们可以将它转换成变量 `NGINX_REPLICAS`。关于模板中变量的使用说明，可以参考[变量设置](#)。

系统提取模板内容中的变量作为配置项。然后在配置项中填写变量的默认值，这里设置成 2。

在不同的环境部署时，可以根据需要，选择不同的配置项，从而调整服务的实例数量。