

容器服务

配置项

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

配置项

配置项概述

配置导入到容器

配置和模板管理多环境

配置导入到环境变量

配置项

配置项概述

最近更新时间：2018-05-30 15:02:16

配置项概述

什么是配置项

配置用来规定一些程序在启动时读入设定，提供了一种修改程序设置的手段，针对不同的对象可以使用不同的配置。

配置项是多个配置的集合，配置项的值可以是字符串，也可以是文件。

配置项支持YAML格式和可视化编辑，单击查看[YAML语法](#)。

配置项只支持新增版本，不支持修改版本。

配置项有什么价值

1. 使用配置项功能可以帮您管理不同环境、不同业务的配置，支持多版本,支持Yaml格式
2. 方便您部署相同应用的不同环境，配置文件支持多版本，方便您进行更新和回滚应用
3. 方便您快速将您的配置以文件的形式导入到容器中

配置文件的操作和使用方法

配置文件的创建

1. 进入到配置文件列表页，单击新建
2. 填写基本信息和配置文件内容，支持YAML格式和可视化编辑

3. 第二步：在容器配置中设置该数据卷的挂载路径

配置文件

[+ 新建](#)

配置ID/配置名称	创建时间	修改时间	操作
[模糊]	2017-07-18 16:29:22	2017-07-18 16:29:22	删除
[模糊]	2017-07-18 16:27:34	2017-07-18 16:28:29	删除
[模糊]	2017-07-18 15:47:06	2017-07-18 16:24:32	删除
[模糊]	2017-07-18 15:46:58	2017-07-18 15:46:58	删除
[模糊]	2017-07-18 09:29:50	2017-07-18 10:06:31	删除
[模糊]	2017-07-17 16:50:34	2017-07-17 16:50:34	删除
[模糊]	2017-07-17 16:50:15	2017-07-17 16:50:15	删除

文件名称

文件名称必须输入，最长60个字符

版本号

版本号由小写字母、数字和 - 组成，且由小写字母开头，小写字母或数字结尾

版本描述

编辑方式 YAML 可视化

配置内容

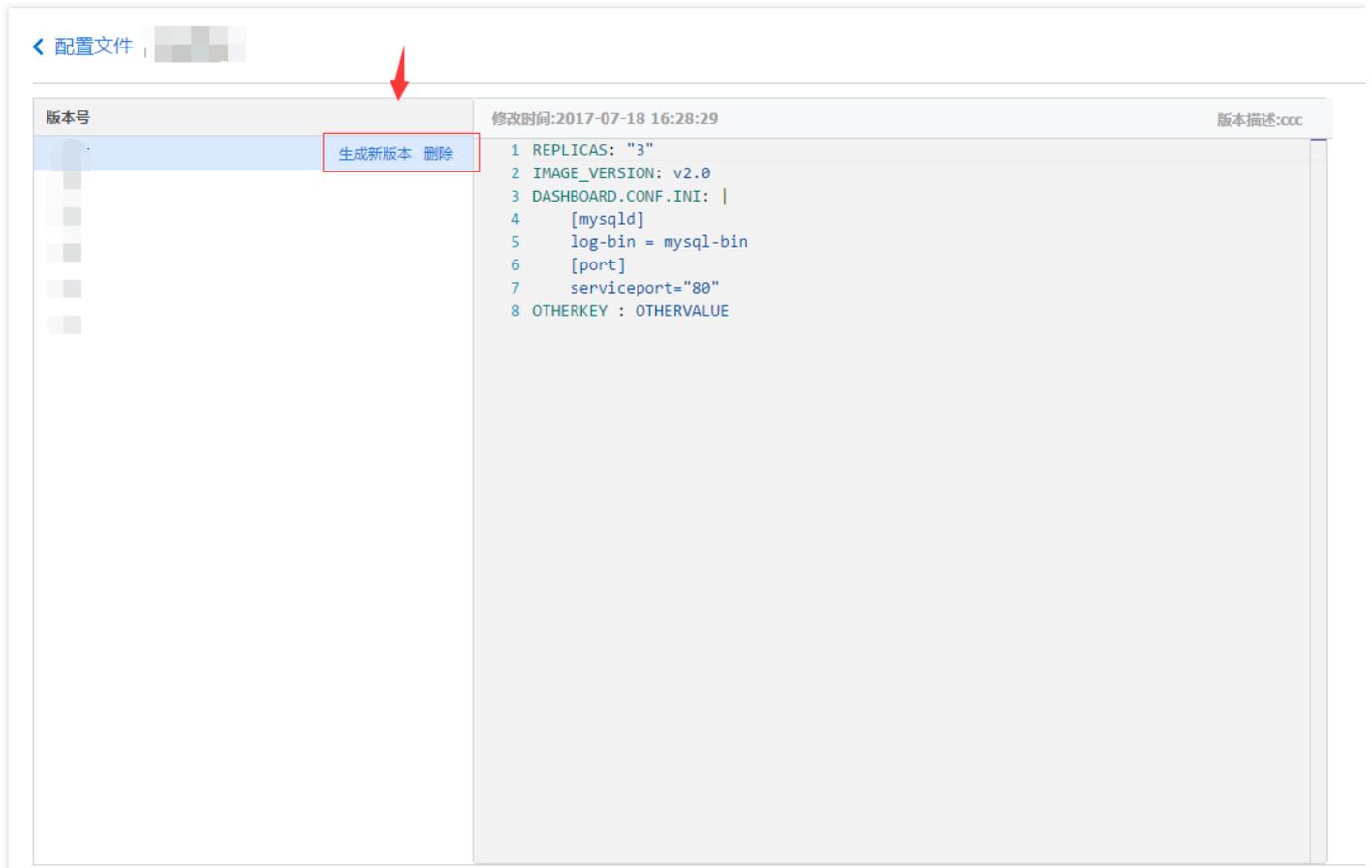
变量名		变量值	
REPLICAS	=	3	×
IMAGE_VERSION	=	v2.0	×
DASHBOARD.CONF.INI	=	[mysqld]	×
OTHERKEY	=	OTHERVALUE	×

[+ 添加变量](#)

配置文件的新版本修订

1. 单击配置文件ID，进入配置文件详情页

2. 选择任意一个版本，单击生成新版本，则是在该版本的基础上修改为一个新版本



配置文件的删除

1. 配置文件详情页可删除指定版本
2. 配置文件列表页可删除该配置文件，并删除配置文件下所有版本

配置文件的使用

使用方法一：将配置项中的配置以数据卷的形式挂载到容器中，[查看详情](#)

使用方法二：使用配置文件 + 应用模板来进行多环境部署，[查看详情](#)

使用方法三：创建服务时使用配置文件导入到环境变量，[查看详情](#)

配置导入到容器

最近更新时间：2018-05-30 15:12:42

配置导入到容器

第一步：创建配置项

1. 进入控制台，创建配置项，填写 `配置项名称`、`版本号`、`版本描述`
2. 支持按YAML语法编辑，支持按可视化编辑
3. YAML语法编辑，格式为 `key : value` 格式，value可以是字符串，也可以是文本。若Value是文本，则用 `"|"` 表示。YAML已缩进作为分割。
4. 支持可视化编辑，可视化的value支持字符串和文本。
如下图：

>> < 服务配置 | 新建配置项

配置项名称 1
配置项名称必须输入，最长60个字符

版本号 2
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 3

编辑方式 YAML 可视化 4

配置内容

```
1 testkey:testvalue
2 testfilekey: |
3   [mysqld]
4   log-bin = mysql-bin
5   [port] 5
6   serviceport="80"
```

创建完成后，创建服务即可使用该配置项。

第二步：通过配置项定义数据卷

1. 进入控制台，单击创建/更新服务
2. 添加数据卷，并指定配置文件类型，选择需要挂载到容器的配置文件的Key.

3. 选择 使用配置项 ， 填写 数据卷名称 用于后续指定目录挂载。
4. 这里测试填写名称 testname ，选择上上述创建的配置项中的 testfilekey



第三步：将数据卷挂载到指定目录

- 填写容器镜像等基本信息，单击高级设置，填写挂载点
- 设置指定的挂载目录，这里我们挂载到 /config 目录下。

- 单击完成，创建完成后即可在容器中使用该配置文件，导入到该路径后，配置文件的key将作为文件名存在。

>
<
返回
|
更新服务

新增request限制

request 用于集群分配资源，当集群中的节点没有request所要求的资源数量时，容器会创建失败。

limit 用于设置容器使用资源的最大上限，避免异常情况下节点资源消耗过多。

内存限制 ①

MIB

内存限制默认request = limit，更多资源限制见[详情](#)。

工作目录

指定容器运行后的工作目录，[查看详情](#)

运行命令

控制容器运行的输入命令，[查看详情](#)

运行参数

传递给容器运行命令的输入参数，[查看详情](#)

环境变量 ①

新增变量

变量名只能包含大小写字母、数字及下划线，并且不能以数字开头

挂载点 ①

▼

▼

×

添加挂载点

容器健康检查 ①

存活检查 检查容器是否正常，不正常则重启实例

就绪检查 检查容器是否就绪，不就绪则停止转发流量到当前实例

[查看健康检查和就绪检查详细介绍](#)

特权级容器

×

容器开启特权级，将拥有宿主机的root权限

隐藏高级设置

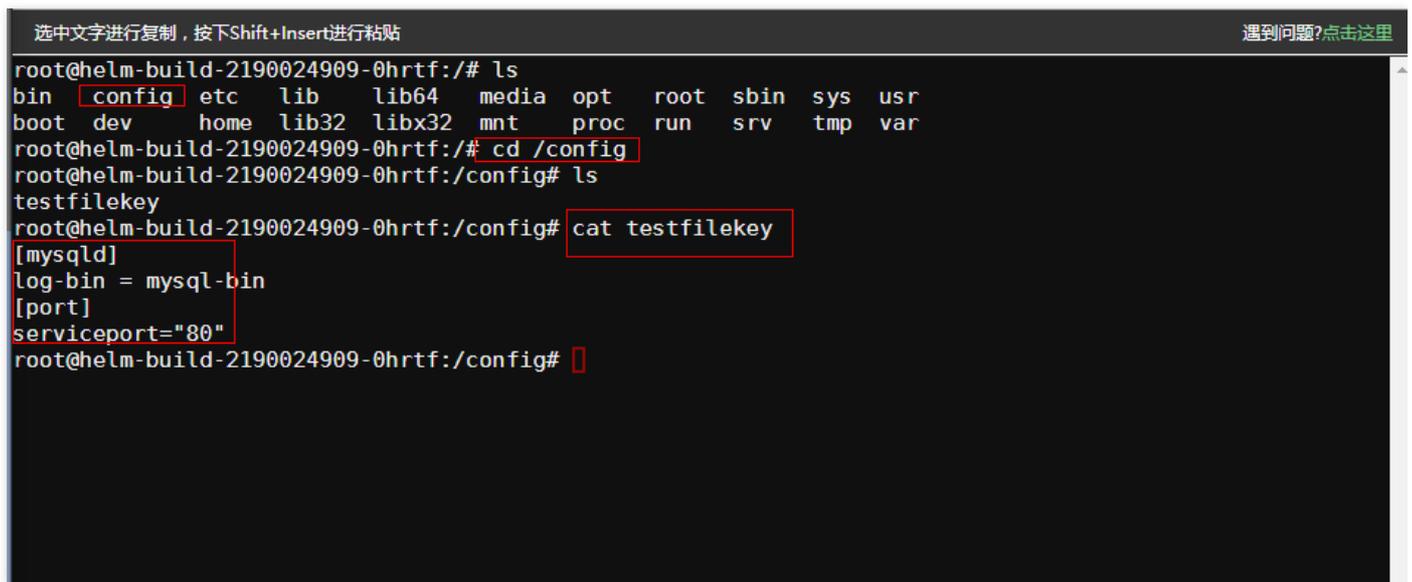
开始更新

取消

第四步：登录容器验证

- 进入控制台，进入服务实例列表页，单击远程终端，登录到容器

- 查看容器内配置的挂载情况



第五步：设置应用程序读取配置

从第四步的验证结果中可以看到，配置项已经在容器启动时以数据卷的形式挂载到容器中了，对于的业务代码或应用程序的容器镜像只需要在启动服务是按上述步骤，就可将配置挂载到容器中，供应用程序初始化等操作。

配置和模板管理多环境

最近更新时间：2018-03-27 13:12:15

配置和模板管理多环境

第一步：创建配置项

1. 进入控制台，创建配置项，填写 `配置项名称`、`版本号`、`版本描述`
2. 支持按YAML语法编辑，支持按可视化编辑
3. YAML语法编辑，格式为 `key : value` 格式，value可以是字符串，也可以是文本。若Value是文本，则用 `"|"` 表示。YAML已缩进作为分割。
4. 支持可视化编辑，可视化的value支持字符串和文本。
如下图：

>> < 服务配置 | 新建配置项

配置项名称 1
配置项名称必须输入，最长60个字符

版本号 2
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 3

编辑方式 YAML 可视化 4

配置内容

```
1 testkey:testvalue
2 testfilekey: |
3   [mysqld]
4   log-bin = mysql-bin
5   [port] 5
6   serviceport="80"
```

创建完成后，创建服务即可使用该配置项。

第二步：创建或更新应用模板使用配置项进行变量替换

1. 进入创建或更新应用模板页面
2. 将常变的参数修改成变量的形式
3. 选择已有配置项为变量赋值

← 新建模板

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
nginx	删除	<pre> 19 spec: 20 containers: 21 - image: nginx 22 imagePullPolicy: Always 23 name: {{.testkey}} 24 resources: 25 limits: 26 cpu: 500m 27 memory: 1Gi 28 requests: 29 cpu: 250m 30 memory: 256Mi 31 securityContext: 32 privileged: false 33 serviceAccountName: "" 34 volumes: null 35 status: {} 36 --- 37 </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	<input type="text" value="default"/>	变量替换
testkey	<input type="text"/>	变量替换

[使用已有配置项赋值](#)

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

服务名

nginx

新增空服务 从

选择配置项 ×

选择的变量只有在模板中存在才会进行赋值操作，导入模板不存在的变量不会进行任何赋值操作。

配置项： 版本：

<input type="checkbox"/>	变量名	变量值	模板中是否存在
<input type="checkbox"/>	testkey	testvalue	存在
<input checked="" type="checkbox"/>	testfilekey	[mysql] log-bin = mysql-bin [port] serviceport="80"	不存在



配置导入到环境变量

最近更新时间：2018-05-30 15:13:24

配置导入到环境变量

第一步：创建配置项

1. 进入控制台，创建配置项，填写 **配置项名称**、**版本号**、**版本描述**
2. 支持按YAML语法编辑，支持按可视化编辑
3. YAML语法编辑，格式为 **key : value** 格式，value可以是字符串，也可以是文本。若Value是文本，则用 **"|"** 表示。YAML已缩进作为分割。
4. 支持可视化编辑，可视化的value支持字符串和文本。
如下图：

<
服务配置
|
新建配置项

配置项名称 1

配置项名称必须输入，最长60个字符

版本号 2

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

3

编辑方式 YAML 可视化 4

配置内容

1 testkey:testvalue
 2 testfilekey: |
 3 [mysqld]
 4 log-bin = mysql-bin
 5 [port] 5
 6 serviceport="80"

完成

取消

创建完成后，创建服务即可使用该配置项。

第二步：创建服务通过配置项导入环境变量

- 1.进入控制台，单击创建/更新服务
- 2.环境变量参数选择导入环境变量

部署设置(Deployment)

数据卷(选填) ⓘ

[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
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 [选择镜像](#)

镜像版本 (Tag)

资源限制

CPU限制				内存限制							
request	0.25	-	limit	0.5	核	request	256	-	limit	1024	MiB

Request用于预分配资源,当集群中的节点没有request所要求的资源数量时,容器会创建失败。
Limit用于设置容器使用资源的最大上限,避免异常情况下节点资源消耗过多。

环境变量 ⓘ [新增变量](#) [从配置项导入](#)
变量名只能包含大小写字母、数字及下划线，并且不能以数字开头

[显示高级设置](#)

注意：服务创建完成后，容器的配置信息可以通过更新服务的方式进行修改

部署设置(Deployment)
选择配置项 ✕

配置文件:
版本:

<input type="checkbox"/> 变量名	变量值	命名规范
<input checked="" type="checkbox"/> testkey	testvalue	符合
<input type="checkbox"/> testfilekey	[mysqld] log-bin = mysql-bin [port] serviceport="80"	符合

提示: 选择的变量将追加到环境变量中, 不符合环境变量命名规范的变量将不会被选中

注意: 服务创建完成后, 容器的配置信息可以通过更新服务的方式进行修改

3.完成创建