

容器服务

持续集成与持续交付

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

持续集成与持续交付

基于TencentHub 的持续集成和持续部署

持续集成与持续交付

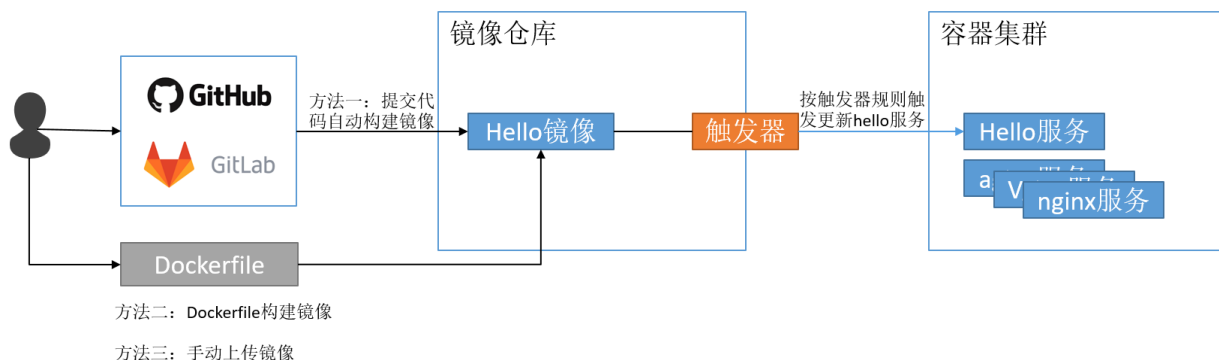
基于TencentHub 的持续集成和持续部署

最近更新时间：2018-04-23 15:11:31

整体流程

图示

集群生命周期



1. 通过容器服务镜像仓库设置镜像来源, 目前支持 源码构建、Dockerfile 构建、手动上传 三种方式。
2. 设置源码构建即可实现每次提交代码, 自动生成容器镜像功能。
3. 若需要设置镜像生成后自动触发更新服务, 则需要针对镜像设置触发器。
4. 设置源码构建和触发器完成后即可实现-提交代码后自动触发构建镜像, 镜像生产自动触发更新服务。

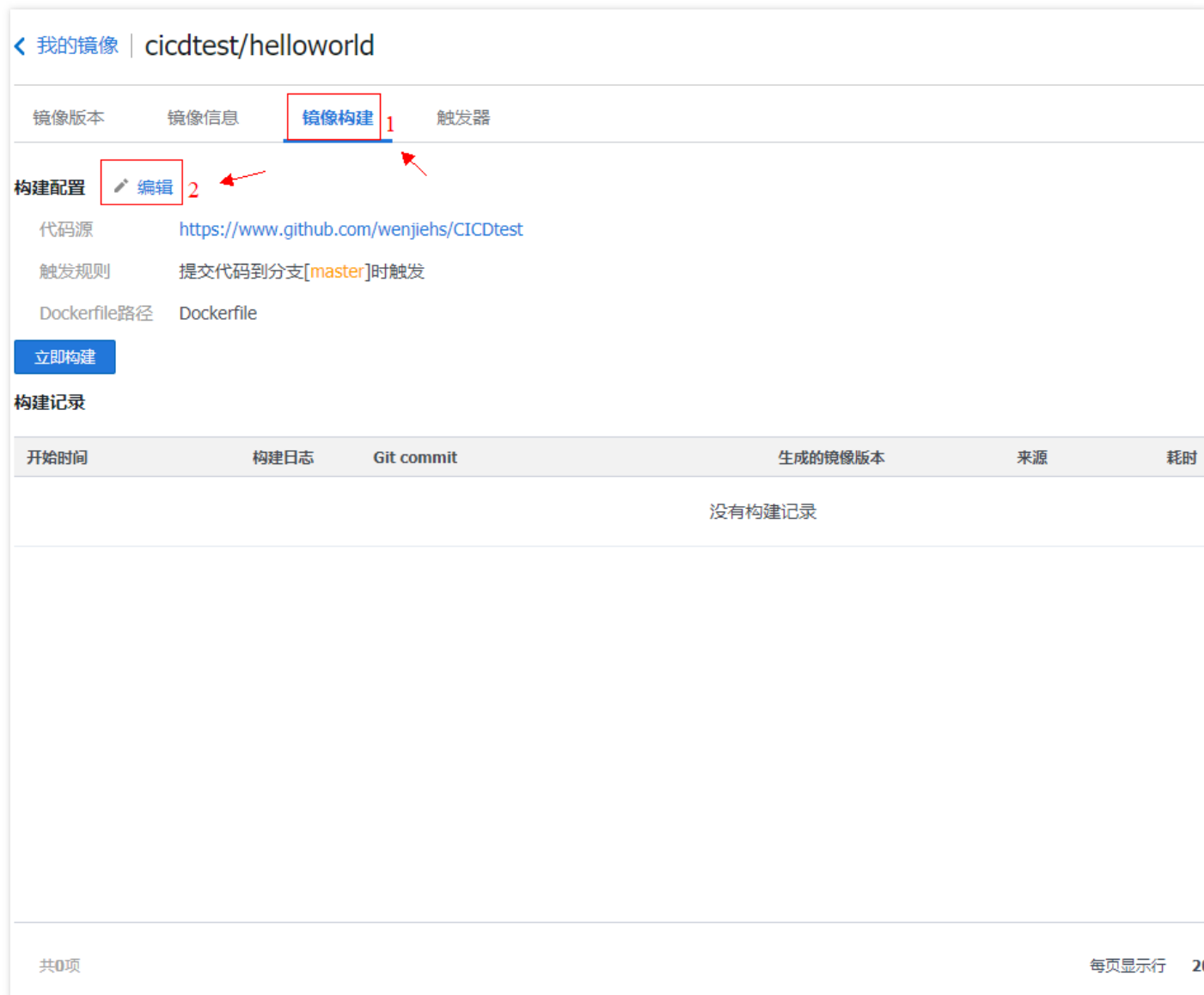
说明

- 源码构建和触发器设置均是独立模块, 可以只使用其中任意环节。
- 源码构建 Dockerfile 构建镜像可查看 [源码构建概述](#)。
- 触发器可支持多种触发动作, 详情查看 [镜像触发器概述](#)。

操作步骤

一、设置源码持续构建镜像

1. 新建需要测试的镜像仓库 `helloworld` , 进入 `helloworld` 详情页。



2. 设置镜像构建设置需要关注以下几点：

- 代码源：当前支持 GitHub 和 Gitlab
- 源码 Repository：需要包含 Dockerfile 文件，Dockerfile 文件是构建镜像的步骤描述；还需要包含所需代码，同时在 Dockerfile 中添加代码。详情可见 [如何编写 Dockerfile](#)。
- 下述提供了本次测试 Helloworld 的 Dockerfile 和 index 文件, 将 Dockerfile 和 index.html 放在同一个根目录下。

Dockerfile 文件：

```
FROM nginx
#file author
MAINTAINER tencentcloudccs
ADD ./ /usr/share/nginx/html
```

index.html 文件

```
hello world
```

- 触发方式：设置提交代码到分支触发。
- Dockerfile 路径：由于 Dockerfile 和 index.html 都根目录下，所以直接填写 Dockerfile 即可。更多见 [源码构建 Dockerfile 路径设置](#)。

[我的镜像](#) | 构建配置

镜像地址 `ccr.ccs.tencentyun.com/cienv/sshdirtest`

代码源  Github  Gitlab

Organization 

Repository ☒ starter 公有仓库

触发方式 满足以下任意条件即触发构建镜像

- ☒ 添加新Tag时触发
- ☒ 提交代码到分支时触发

☒ master

镜像版本命名规则 - ☐ 分支/标签 - ☒ 更新时间 - ☐ commit号

Dockerfile路径 `-f $REPO/qcloud/Dockerfile`

构建目录 `$REPO/qcloud`

构建参数

VERSION	=	1.0.0	×	<code>--build-arg=VERSION=1.0.0</code>
WELCOME_MSG	=	hello	×	<code>--build-arg=WELCOME_MSG=hello</code>

[新增变量](#)

完成

可以引用arg变量作为镜像Tag的一部分

`-f $REPO/qcloud/Dockerfile`

`$REPO/qcloud`

`--build-arg=VERSION=1.0.0`

`--build-arg=WELCOME_MSG=hello`

第一步设置完成后，即可实现每次提交代码，自动生成容器镜像功能。可提交代码进行测试，也可以手动执行立即构建。

镜像版本

镜像信息

镜像构建

触发器

构建配置

编辑

代码源

https://www.github.com/wenjiehs/CICDtest

触发规则

提交代码到分支[master]时触发

Dockerfile路径

Dockerfile

立即构建

构建记录

开始时间	构建日志	Git commit	生成的镜像版本	来源	耗时	状态	操作
2017-08-23 11:51:54		439027d1111e570de224dbdbd2208dceacffa...	hellotest-master-20170...	源码构建	N/A	构建中	取消构建

二、设置镜像仓库触发器持续部署服务

1. 首先需要有一个目标服务，用户持续部署，如无，则需创建一个。这里我们创建一个 [Nginx 服务](#)。
2. 进入上述设置好源码构建的 helloworld 镜像仓库详情页，进入触发器设置。

< 我的镜像 | cicdtest/helloworld

镜像版本

镜像信息

镜像构建

触发器

1

通过触发器，可以在每次生成新的Tag（镜像版本）时，自行执行动作，如：自动更新使用该镜像仓库的服务

添加触发器

2

触发记录

触发器名称	触发器条件	触发动作	触发结果	触发时间
没有记录				

3. 添加完成一个触发器，需要关注以下几点：

- 触发条件：根据生成的镜像的版本号(tag),判断是否执行触发动作，若触发条件匹配则执行触发动作，这里选择全部触发,即只要有新的镜像 tag 生成就会执行触发动作。
- 触发动作：当前支持更新服务指定容器。选择需要持续部署的服务和容器。

[< 我的镜像](#) | cicdtest/helloworld

镜像版本

镜像信息

镜像构建

触发器

通过触发器，可以在每次生成新的Tag（镜像版本）时，自行执行动作，如：自动更新使用该镜像仓库的服务

触发器名称

cicdtest

英文字母开头，2-64个字符以内，支持a-z、A-Z、0-9、-、_

触发条件

1

☒ 全部触发

镜像仓库内，有新的Tag生成，或Tag发生更新时，触发动作

☐ 指定Tag触发

镜像仓库内，有指定Tag生成或更新时，触发动作

☐ 正则触发

镜像仓库内，有符合正则表达式的Tag生成或更新时，触发动作

触发动作

更新容器的镜像

需指定更新的服务，以及该服务下的指定容器镜像

选择服务/镜像

2

1/cds-2nqy3s3i/default/nginx/nginx

v

保存

取消

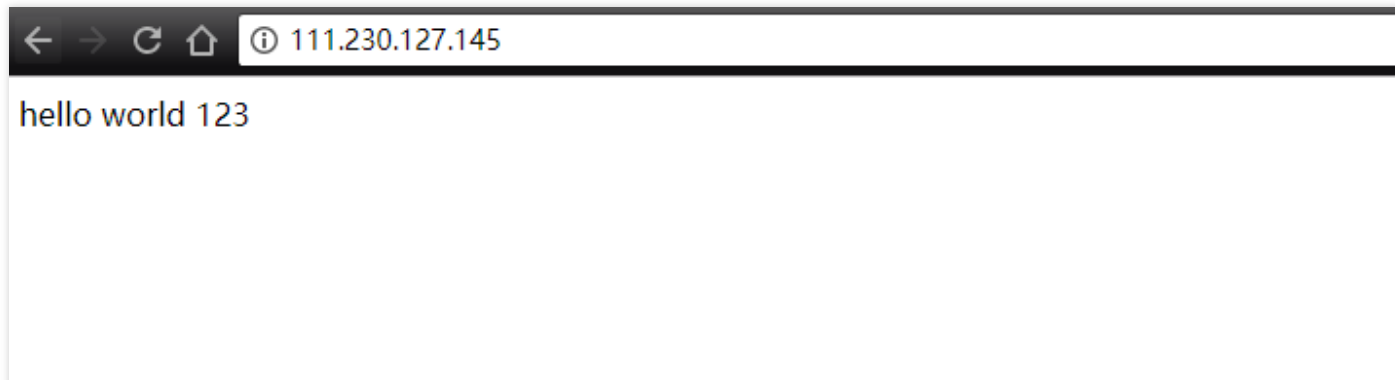
[添加触发器](#)

第二步设置完成后，即可手动上传镜像或手动构建镜像验证触发器。以下是手动构建验证测试。

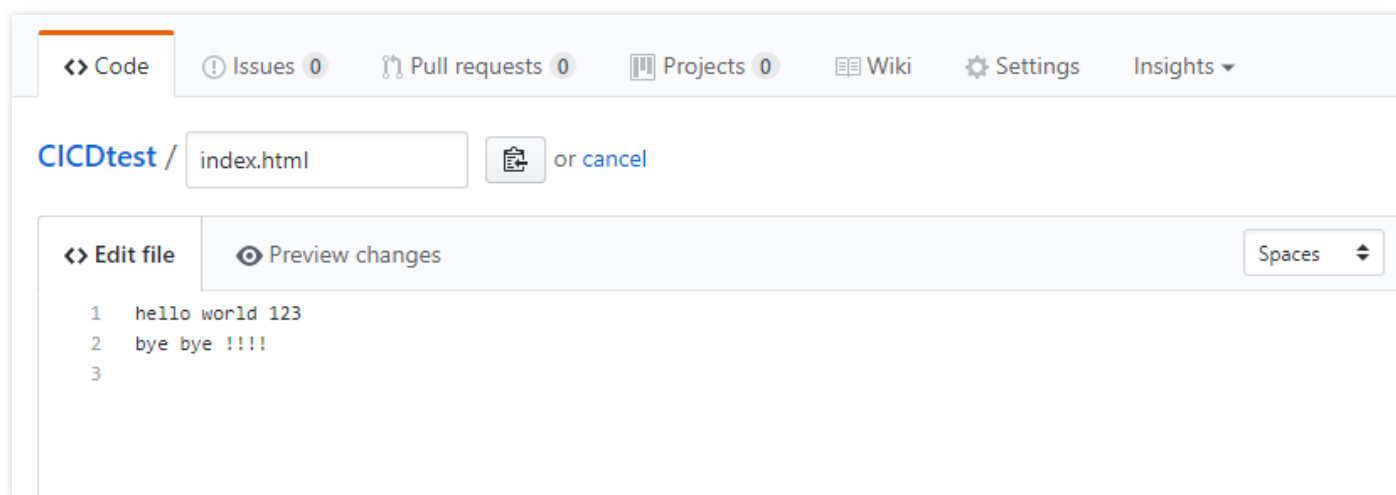
三、提交代码验证

通过上述第一步和第二步已经完成了持续集成镜像和持续交付服务，下述进行提交代码做整个流程的最终效果验证。

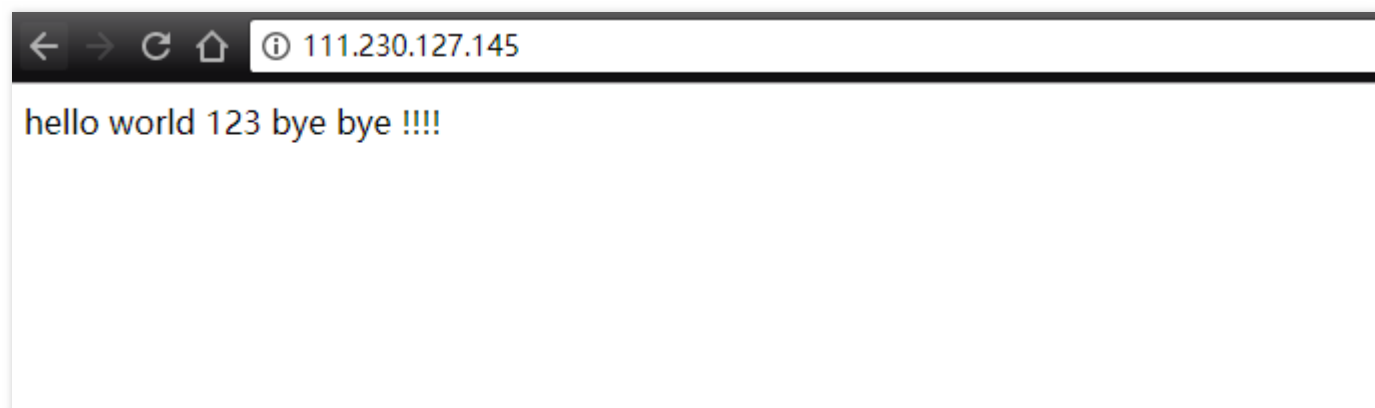
1. 通过公网访问该 Nginx 服务，当前效果如下。



2. 更新 index.html 文件并提交到 Github。可查看部署过程。



3. 再次访问该服务。



到此为止，已经完成基于 TencentHub 持续集成和持续交付的部署。