

(75条消息)Docker-tag_小工匠-CSDN博客

 blog.csdn.net/yangshangwei/article/details/52799675



概述

我们都知道git的tag功能是为了将代码的某个状态打上一个戳，通过tag我们可以很轻易的找到对应的提交。

docker的tag似乎更加灵活，docker将文件等信息的变动抽象为一次次的commit，每一次commit以后可能走向不同的分支，当我们完成dockerfile的构建后，会生成一串无规则的字符串代表此次生成的ID，此时，tag的作用就是为他创建一个友好的NAME，方便我们对镜像库的管理。

docker images

docker images命令 查询本地的镜像

```
[root@entel2 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
entel_..._images    latest             a55c335a5232       36 hours ago       7.393 GB
entel_..._images    latest             38eb69df8f14       39 hours ago       23.93 GB
entel_base_images   with_ora_tt_rpms   ee2a2c6588ab       10 weeks ago       3.603 GB
entel_base_image    withtt_oracle      f12ea669055f       3 months ago       3.357 GB
entel_base_image    withtt             42409dbfab32       3 months ago       3.357 GB
autumnw/centos6.5_x86-64  withssh           508b60032889       3 months ago       728.1 MB
autumnw/centos6.5_x86-64  pdns              dfb956ee2d4a       2 years ago        674.8 MB
```

关于tag，我们比较关注REPOSITORY(名称)，TAG(标签)，IMAGE ID(镜像ID)三列。

tag的生成

创建运行容器时指定tag

```
[root@localhost apache2]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              apache2            421e206ae6f9       43 minutes ago     588.4 MB
ubuntu              latest            a0caa2553854       10 months ago      574.5 MB
```

```
ubuntu:latest - 10 months ago
[root@localhost apache2]# docker run -it ubuntu:apache2 /bin/bash
root@e116beb2b5e2:/# ll
total 44
drwxr-xr-x. 21 root root 4096 Oct 12 09:43 ./
drwxr-xr-x. 21 root root 4096 Oct 12 09:43 ../
-rwxr-xr-x. 1 root root 0 Oct 12 09:43 .dockerenv*
drwxr-xr-x. 2 root root 4096 Nov 21 2015 bin/
drwxr-xr-x. 2 root root 6 Apr 10 2014 boot/
drwxr-xr-x. 5 root root 380 Oct 12 09:43 dev/
drwxr-xr-x. 88 root root 4096 Oct 12 09:43 etc/
drwxr-xr-x. 2 root root 6 Apr 10 2014 home/
drwxr-xr-x. 13 root root 4096 Nov 21 2015 lib/
drwxr-xr-x. 2 root root 33 Oct 28 2015 lib64/
drwxr-xr-x. 2 root root 6 Oct 28 2015 media/
drwxr-xr-x. 2 root root 6 Apr 10 2014 mnt/
drwxr-xr-x. 2 root root 6 Oct 28 2015 opt/
dr-xr-xr-x. 258 root root 0 Oct 12 09:43 proc/
drwx----- 2 root root 58 Oct 12 08:59 root/
drwxr-xr-x. 7 root root 4096 Oct 28 2015 run/
drwxr-xr-x. 2 root root 4096 Nov 21 2015 sbin/
drwxr-xr-x. 2 root root 6 Oct 28 2015 srv/
dr-xr-xr-x. 13 root root 0 Oct 12 06:53 sys/
drwxrwxrwt. 3 root root 28 Nov 21 2015 tmp/
drwxr-xr-x. 10 root root 97 Nov 21 2015 usr/
drwxr-xr-x. 12 root root 4096 Oct 12 08:59 var/
root@e116beb2b5e2:/#
```

手动指定docker tag

语法

```
docker tag old-image[:old-tag] new-image[:new-tag]
1
```

tag帮助

```
[root@entel1 ~]# docker tag --help
```

Usage: docker tag [OPTIONS] IMAGE[:TAG] [REGISTRYHOST/][USERNAME/]NAME[:TAG]

Tag an image into a repository

- f, --force=false Force
- help=false Print usage
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

-f 意思是强制覆盖

栗子

把entel_zmc_images :latest指定为entel_zmc_images:zmc_base

```
docker tag entel_zmc_images entel_zmc_images:zmc_base
```

1

ps.记住，tag中不能再包含：(冒号)了!

```
[root@entell ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
entel_zmc_images     latest             a55c335a5232       28 hours ago       7.393 GB
entel_zmc_images     latest             78ff7137c3a6       12 days ago        4.447 GB
entel_zmc_images     latest             5ed6a6c28ed3       12 days ago        4.794 GB
entel_zmc_images     latest             7c2b8e830743       13 days ago        6.576 GB
entel_zmc_images     zmc_base          7c2b8e830743       13 days ago        6.576 GB
```

`docker tag entel_zmc_images entel_zmc_images:zmc_base`，系统又帮我们加上了latest，也就是说，当我们不指定tag的时候，系统会自动帮我们补上latest的tag，然后去匹配，如果命中，就用对应的容器ID去创建新的tag。

当我们执行刚才命令的时候，其实还是存在一个entel_zmc_images:zmc_base的镜像，按理来说会发生重名，但是系统还是优先使用了ID为7c2b8e830743的镜像。

latest

这个latest其实在使用中不是最新的意思，而是默认值(defalut)的意思。

也就是说，如果在tag为可选的命令中，我们没有写上tag，如 `docker pull entel_zmc_images:zmc_base vs docker pull entel_zmc_images`，前者有确定的tag，而后者没有，这时系统会自动添加一个:latest标签，然后去匹配。

这时如果latest对应的镜像不存在就会报错！

运行镜像

理解了latest，那么下面两种方式就很简单了。

```
# 运行默认镜像，默认会找tag
docker run entel_zmc_images
```

```
# 运行指定tag的镜像
docker run entel_zmc_images:zmc_base
```

- 1
- 2
- 3
- 4
- 5

删除镜像

同理，我们直接删除某个name，系统只会删除latest标签的那一条记录，如

```
docker rmi entel_zmc_images
```

1

这个删除操作不会影响相同ID的其他镜像。

docker镜像的版本控制

如果需要升级某个docker镜像，我们可以这样做。

- 1.给每个新生成的镜像都打上相应版本的tag。此时可能存在image:latest、image:v1、image:v2等。
- 2.我们要从v1升级到v2，首先我们将导入的v2镜像强制重命名为image:latest，命令为 `docker tag -f image:v2 image:latest`
- 3.docker stop之前正在运行的容器
- 4.启用docker run image，此时image的等价镜像image:latest就是最新的V2镜像。

总结下步骤：load/tag/stop/run

-  点赞 5
-  评论
-  分享
-  收藏 4
-  手机看
-  打赏
- ...
- [订阅博主](#)
- [关注](#)

[qxqsunshine的博客](#)

 4011

docker 标记(Tag),推送(push),拉取(pull)你自己的镜像

链接:<https://blog.csdn.net/jpiverson/article/details/50731568>里面的有很详细的步骤和见解.1 输入docker images命令来查看当前的镜像列表:2 找到镜像的id3 使用IMAGE ID和docker tag命令来标记docker-whale镜像.命令格式:eg : docker tag 7d9495d...

future-wy

2736

docker镜像构建、镜像打tag、镜像推送

docker镜像构建、镜像打tag、镜像推送 镜像构建：将本地镜像打包 docker buile -t isoName:tag -f /Dockerfile isoName:镜像名称tag :版本号Dockerfile:dockerfile文件 镜像打tag：镜像添加版本号docker tag [ImageId] registry.cn-...



抢沙发

pointomine的博客

1万+

Docker命令之：tag命令

作用：Create a tag TARGET IMAGE that refers to SOURCE IMAGE使用镜像源创建一个指定镜像的标签语法：docker tag SOURCE IMAGE[:TAG] TARGET IMAGE[:TAG] 实例：以hyperledger/fabric-zookeeper:x86_64-1.0.1为镜像源创建一个tag为hype... CodyGuo的博客

1万+

docker-tags 命令行获取docker远程仓库上指定镜像的tag列表

docker-tags命令行获取docker远程仓库上指定镜像的tag列表说明用于命令行获取docker远程仓库上指定镜像的所有tag列表，支持版本号模糊搜索，可与dockersearch配合搜索。1.脚本

```
#!/bin/bashAPI="https://registry.hub.docker.com/v1/repositories"DEFAULT_NAME="nginx"DEF.....
```

Docker-tag_花米徐xl_lx的专栏-CSDN博客_docker tag registryhost/

8-26

Docker-tag 概述 docker images tag的生成 创建运行容器时指定tag 手动指定docker tag 语法 栗子 latest 运行镜像 删除镜像 docker镜像的版本控制 概述我们都知道git...

docker镜像build, tag, push_Yu的博客-CSDN博客

8-26

```
#!/bin/bashversion=$1docker build -f dockerfile -t 镜像名:${version} .docker tag 镜像名:${version} 镜像仓库/镜像名:${version} docker push 镜像...
```

2万+

Docker实战-编写Dockerfile

一、编译镜像1. 编译镜像 Dockerfile类似于Makfile，用户使用docker build就可以编译镜像，使用该命令可以设置编译镜像时使用的CPU数量、内存大小、文件路径等 语法：
docker build [OPTIONS] PATH| URL| - 常见选项： -t 设置镜像的名称和TAG，格式为name:tag -...
菜刚RyuGou的专栏

1万+

Dockerfile 指令详解

Docker创建镜像的方式有两种：一种通过commit的方式：把做了一系列操作的容器关闭，然后利用docker的commit指令：dockercommit 容器ID 镜像名：tag。然后dockerpush到镜像仓库。别人pull下来的再次启动的时候，就是你当前的操作的形态。另一种是通过Dockerfile构建的方式：把操作的步骤通过脚本的形式写下来，然后构建的时候，Docker会按照你写.....

Docker tag_衣舞晨风-CSDN博客

8-26

语法:docker tag old-image-repository[:old-tag] new-image-repository[:new-tag]//或者
docker tag old-image-id new-image-repository[:new-tag]小注: 我们...

Docker-tag - To be a better man~

12-16

概述我们都知道git的tag功能是为了将代码的某个状态打上一个戳,通过tag我们可以很轻易的找到对应的提交。 docker的tag似乎更加灵活,docker将文件等信息的变动抽象为...
极客on之路

6384

docker 删除某个 tag

我怎么删除 image id 一样的某些 tag 或者 REPOSITORY，只保留一个第 1 条附言：
2017-03-31 10:59:40 +08:00docker rmi image_name docker
rmihub.doge.net/ubuntu:latest...
木灵的专栏

7万+

Docker 重命名镜像 TAG

使用dockerimages时,可能会出现REPOSITORY和TAG均为none的镜像,如下图重命名镜像dockertagIMAGEID(镜像id)REPOSITORY:TAG (仓库:标签) #例子
dockertagca1b6b825289registry.cn-hangzhou.aliyuncs.com/xxxxxxx:v1.0

Docker 重命名镜像 TAG_木灵的专栏-CSDN博客


8-27

使用dockerimages时,可能会出现REPOSITORY和TAG均为none的镜像,如下图重命名镜像dockertagIMAGEID(镜像id)REPOSITORY:TAG(仓库:标签)#例子
dockertagca1b6b825289...

Docker命令之: tag命令_point0mine的博客-CSDN博客

8-26

作用:Create a tag TARGET IMAGE that refers to SOURCE IMAGE使用镜像源创建一个指定镜像的标签语法:docker tag SOURCE IMAGE[:TAG] TARGET IMAGE[:TAG]实例:以...
妙音

 1万+

docker search怎么查详情

刚用dockersearch的人,看到查出一堆结果,此时会有两个疑问1.如何查详情?答案是没
有,docker没有查看详情的命令。2.结果太多,哪个是最好的,该选哪个?可以简单认为
官方是最好的,热度高的是最好的。*显示官方dockersearch[包名]--filter"is-
official=true"*过滤热度dockersearch[包名]--stars=33.....
成长的足迹

 5万+

docker build命令详解

dockerbuild命令用于根据给定的Dockerfile和上下文以构建Docker镜像。dockerbuild命令的使用格式:dockerbuild[OPTIONS]<PATH|URL|->1.常用OPTIONS选项说明--
build-arg,设置构建时的环境变量--no-cache,默认false。设置该选项,将不使用
BuildCache构建镜像.....

docker常用命令-docker tag_OneZeroTwoFour-CSDN博客_docker tag...

8-23

docker tag语义标记本地镜像到某一仓库语法docker tag SOURCE IMAGE[:TAG]
TARGET IMAGE[:TAG]实例将my:v1归到tomcat:v2docker tag my:v1
tomcat:v2 docker ...

docker镜像构建、镜像打tag、镜像推送_future-wy-CSDN博客

[docker镜像构建、镜像打tag、镜像推送](#) [镜像构建:将本地镜像打包 docker buile -t isoName:tag -f /DockerfileisoName:镜像名称tag :版本号Dockerfile:dockerfile...](#)

©2020 CSDN 皮肤主题: Age of Ai 设计师: meimeiellie [返回首页](#)


[公安备案号 11010502030143](#) [京ICP备19004658号](#) [京网文〔2020〕1039-165号](#) [经营性网站备案信息](#) ©1999-2020 北京创新乐知网络技术有限公司

分类专栏

- 

[【小工匠聊架构】](#)


付费

2篇
- 


[【APM】](#)

2篇
- 


[【MyBatis源码解析】](#)

5篇
- 


[【LeetCode】](#)

1篇
- 


[【Tomcat架构&源码&调优】](#)

11篇
- 


[【Data Structures & Algorithms】](#)

13篇
- 


[【MQ-Apache RocketMQ】](#)

11篇
- 


[【MQ-Apache Kafka】](#)

4篇
- 

[【MQ-RabbitMQ】](#)

2篇
- 

[【ZK-Apache ZooKeeper】](#)
















77篇
- 










[【ES-Elasticsearch】](#)






20篇
- 

[【Spring Cloud Finchley】](#)

20篇

-  【实战-Spring Cloud Finchley实战】 7篇
-  【Spring Boot2.X】 16篇
-  【实战-SSM In Action】 48篇
-  【MyBatis】 31篇
-  【Redis-入门到精通】 25篇
-  【Redis-进阶实战】 29篇
-  【Java - Java 8】 7篇
-  【Java - Java Base】 27篇
-  【Java并发编程】 50篇
-  【Java设计模式】 6篇
-  【JVM高级特性】 37篇
-  【Java工具类】 2篇
-  【Spring-IOC】 38篇
-  【Spring-AOP】 25篇
-  【Spring-AOP进阶】 23篇

-  [【Spring-JDBC】](#) 20篇
-  [【Spring-Cache】](#) 9篇
-  [【Spring-MVC】](#) 19篇
-  [【Spring-OXM】](#) 5篇
-  [【Quartz任务调度】](#) 20篇
-  [【MySQL深度剖析】](#) 22篇
-  [【MySQL基础篇】](#) 23篇
-  [【Oracle基础】](#) 63篇
-  [【Oracle通用优化】](#) 24篇
-  [【Git】](#) 1篇
-  [【Maven】](#) 3篇
-  [【Dubbo】](#) 1篇
-  [【系统运维-Linux】](#) 56篇
-  [【系统运维-Shell】](#) 29篇
-  [【容器技术-Docker】](#) 8篇

• 	<u>【Nginx】</u>	1篇
• 	<u>【Hadoop】</u>	5篇
• 	<u>【Python】</u>	18篇
• 	<u>【前端开发-Vue2.x基础篇】</u>	5篇
• 	<u>【前端开发-JavaScript基础篇】</u>	4篇
• 	<u>【移动开发-Android基础篇】</u>	29篇
• 	<u>【移动开发-Android网络编程】</u>	3篇
• 	<u>【移动开发-AndroidMD】</u>	12篇
• 	<u>【移动开发-Android常见UI】</u>	3篇
• 	<u>【移动开发-Android-图片处理】</u>	8篇
• 	<u>【移动开发-Android杂货箱】</u>	17篇
• 	<u>【移动开发-Android中级篇】</u>	11篇
• 	<u>【移动开发-AndroidStudio】</u>	18篇
• 	<u>【万花筒】</u>	9篇