# docker列出镜像用法

* 查看镜像、容器、数据卷所占用的空间

docker system df

## 虚悬镜像

上面的镜像列表中，还可以看到一个特殊的镜像，这个镜像既没有仓库名，也没有标签，均为

这个镜像原本是有镜像名和标签的，原来为 mongo:3.2 ，随着官方镜像维护，发布了新版本后，重新 docker pull mongo:3.2 时， mongo:3.2 这个镜像名被转移到了新下载的镜像身上，而旧的镜像上的这个名称则被取消，从而成为了 。除了 docker pull 可能导致这种情况， docker build 也同样可以导致这种现象。由于新旧镜像同名，旧镜像名称被取消，从而出现仓库名、标签均为 的镜像。这类无标签镜像也被称为 虚悬镜像(dangling image) ，

可以用下面的命令专门显示这类镜像

$ docker images -f dangling=true

REPOSITORY TAG IMAGE ID CREATED SIZE

<none> <none> 00285df0df87 5 days ago 342 MB

一般来说，虚悬镜像已经失去了存在的价值，是可以随意删除的，可以用下面的命令删除。

$ docker rmi $(docker images -q -f dangling=true)

## 列出部分镜像

根据仓库名列出镜像

$ docker images ubuntu

列出特定的某个镜像，也就是说指定仓库名和标签

$ docker images ubuntu:16.04

docker images 还支持强大的过滤器参数 –filter ，或者简写 -f 。之前我们已经看到了使用过滤器来列出虚悬镜像的用法，它还有更多的用法。比如，我们希望看到在mongo:3.2 之后建立的镜像，可以用下面的命令

$ docker images -f since=mongo:3.2

想查看某个位置之前的镜像也可以，只需要把 since 换成 before 即可。此外，如果镜像构建时，定义了 LABEL ，还可以通过 LABEL 来过滤

$ docker images -f label=com.example.version=0.1

也可以使用reference

$ docker images --filter=reference='busy\*:\*libc'

REPOSITORY TAG IMAGE ID CREATED SIZE

busybox uclibc e02e811dd08f 5 weeks ago 1.09 MB

busybox glibc 21c16b6787c6 5 weeks ago 4.19 MB

## 以特定格式显示

默认情况下， docker images 会输出一个完整的表格，但是我们并非所有时候都会需要这些内容。比如，刚才删除虚悬镜像的时候，我们需要利用 docker images 把所有的虚悬镜像的ID 列出来，然后才可以交给 docker rmi 命令作为参数来删除指定的这些镜像，这个时候就用到了 -q 参数。

$ docker images -q

5f515359c7f8

05a60462f8ba

fe9198c04d62

00285df0df87

f753707788c5

f753707788c5

1e0c3dd64ccd

–filter 配合 -q 产生出指定范围的 ID 列表，然后送给另一个 docker 命令作为参数，从而针对这组实体成批的进行某种操作的做法在 Docker 命令行使用过程中非常常见，不仅仅是镜像，将来我们会在各个命令中看到这类搭配以完成很强大的功能。因此每次在文档看到过滤器后，可以多注意一下它们的用法

### Format the output

The formatting option (--format) will pretty print container output using a Go template.

Valid placeholders for the Go template are listed below:

| **Placeholder** | **Description** |
| --- | --- |
| .ID | Image ID |
| .Repository | Image repository |
| .Tag | Image tag |
| .Digest | Image digest |
| .CreatedSince | Elapsed time since the image was created |
| .CreatedAt | Time when the image was created |
| .Size | Image disk size |

When using the --format option, the image command will either output the data exactly as the template declares or, when using the table directive, will include column headers as well.

The following example uses a template without headers and outputs the ID and Repositoryentries separated by a colon for all images:

$ docker images --format "{{.ID}}: {{.Repository}}"

77af4d6b9913: <none>

b6fa739cedf5: committ

78a85c484f71: <none>

30557a29d5ab: docker

5ed6274db6ce: <none>

746b819f315e: postgres

746b819f315e: postgres

746b819f315e: postgres

746b819f315e: postgres

To list all images with their repository and tag in a table format you can use:

$ docker images --format "table {{.ID}}\t{{.Repository}}\t{{.Tag}}"

IMAGE ID REPOSITORY TAG

77af4d6b9913 <none> <none>

b6fa739cedf5 committ latest

78a85c484f71 <none> <none>

30557a29d5ab docker latest

5ed6274db6ce <none> <none>

746b819f315e postgres 9

746b819f315e postgres 9.3

746b819f315e postgres 9.3.5

746b819f315e postgres latest

参考：https://blog.csdn.net/wos1002/article/details/78795459?utm\_source=copy