



链滴

# Docker 安装与简单使用介绍

作者: [GeekBoyDqz](#)

原文链接: <https://ld246.com/article/1567341955067>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## 一、Docker安装

### 1、镜像源配置

- 由于centos系统源自带的docker版本比较老不建议使用，这里安装新版docker
- docker依赖环境：
  - 1、内核版本：3.10+ (由于namespace中user需运行在内核3.10版本上)
  - 2、64bit cpu

#### 1.1：下载docker-ce镜像

进入清华大学镜像网站首页，找到docke-ce镜像

|              |                  |
|--------------|------------------|
| artixlinux   | 2019-02-23 14:49 |
| bananian     | 2019-02-23 10:29 |
| bioconductor | 2019-02-23 14:55 |
| bjlx         | 2019-02-23 13:13 |
| blackarch    | 2019-02-23 15:30 |
| centos       | 2019-02-23 13:38 |
| centos-alt   | 2019-02-23 13:28 |

#### 1.2：选择Linux版本

## Index of /docker-ce/

File Name ↓

Parent directory/

linux/

mac/

win/

### 1.3: 选择CentOS

## Index of /docker-ce/linux/

File Name ↓

Parent directory/

centos/

debian/

fedora/

raspbian/

static/

ubuntu/

### 1.4: 复制链接地址

## Index of /docker-ce/linux/centos/

File Name ↓

Parent directory/

7/

docker-ce.repo

SPG

在新标签页中打开链接(T)

在新窗口中打开链接(W)

在隐身窗口中打开链接(G)

链接另存为(K)...

复制链接地址(E)

检查(N)

Ctrl+Shift+I

### 1.5: 下载镜像源

登入Linux系统使用wget进行下载到系统中

```
[root@docker ~]# cd /etc/yum.repos.d/  
[root@docker yum.repos.d]# wget https://mirrors.tuna.tsinghua.edu.cn/docker-ce/linux/centos/docker-ce.repo
```

注意：下载下来的yum源中默认的地址为docker官方站点，可以手动更改yum源指定成国内的docker镜像网站

## 1.6: 更改镜像源地址

可将下面的清华镜像网站docker的下载路径复制更改docker.repo镜像源的默认官方源

<https://mirrors.tuna.tsinghua.edu.cn/docker-ce/>

```
[root@docker yum.repos.d]# vim docker-ce.repo
```

```
name=Docker CE Stable - $basearch  
baseurl=https://download.docker.com/linux/centos/7/$basearch/stable  
enabled=1  
gpgcheck=1  
gpgkey=https://download.docker.com/linux/centos/gpg  
  
[docker-ce-stable-debuginfo]  
name=Docker CE Stable - Debuginfo $basearch  
baseurl=https://download.docker.com/linux/centos/7/debug-$basearch/stable  
enabled=0  
gpgcheck=1  
gpgkey=https://download.docker.com/linux/centos/gpg  
  
[docker-ce-stable-source]  
name=Docker CE Stable - Sources  
baseurl=https://download.docker.com/linux/centos/7/source/stable  
enabled=0  
gpgcheck=1  
gpgkey=https://download.docker.com/linux/centos/gpg  
  
[docker-ce-edge]  
name=Docker CE Edge - $basearch  
:s@https://download.docker.com/@https://mirrors.tuna.tsinghua.edu.cn/docker-ce/
```

- 替换后为下图:

```
[docker-ce-nightly]  
name=Docker CE Nightly - $basearch  
baseurl=https://mirrors.tuna.tsinghua.edu.cn/docker-ce/linux/centos/7/$basearch/  
nightly  
enabled=0  
gpgcheck=1  
gpgkey=https://mirrors.tuna.tsinghua.edu.cn/docker-ce/linux/centos/gpg  
  
[docker-ce-nightly-debuginfo]  
name=Docker CE Nightly - Debuginfo $basearch  
baseurl=https://mirrors.tuna.tsinghua.edu.cn/docker-ce/linux/centos/7/debug-$bas  
earch/nightly  
enabled=0  
gpgcheck=1  
gpgkey=https://mirrors.tuna.tsinghua.edu.cn/docker-ce/linux/centos/gpg  
  
[docker-ce-nightly-source]  
name=Docker CE Nightly - Sources  
baseurl=https://mirrors.tuna.tsinghua.edu.cn/docker-ce/linux/centos/7/source/nig  
htly  
enabled=0  
gpgcheck=1  
gpgkey=https://mirrors.tuna.tsinghua.edu.cn/docker-ce/linux/centos/gpg  
83,1 底端
```

## 2、安装Dokcer

#检查yum源

```
[root@docker ~]# yum repolist
```

## #安装docker

```
[root@docker ~]# yum -y install docker-ce
```

### ● 报错:

如果之前安装了旧版本的docker则安装新版的时候会出现以下报错

```
Transaction check error:
  file /usr/bin/docker from install of docker-ce-cli-1:19.03.1-3.el7.x86_64 conflicts
with file from package docker-common-2:1.13.1-102.git7f2769b.el7.centos.x86_64
  file /usr/bin/dockerd from install of docker-ce-3:19.03.1-3.el7.x86_64 conflicts wi
th file from package docker-common-2:1.13.1-102.git7f2769b.el7.centos.x86_64
```

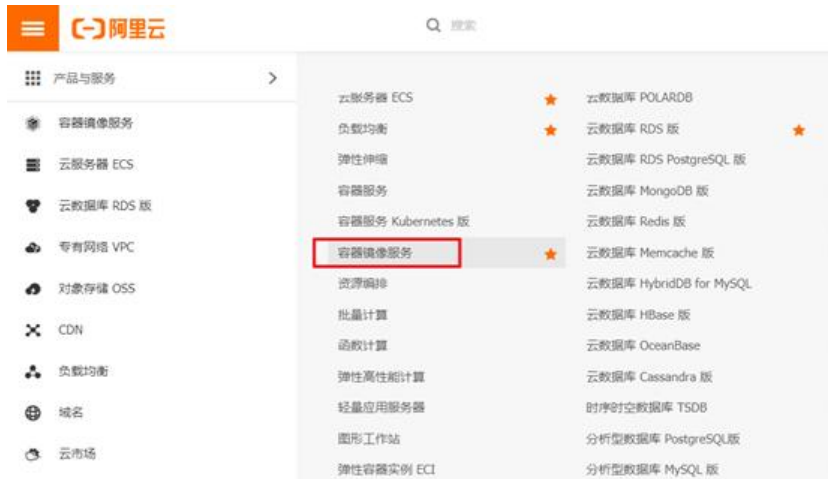
### ● 解决方法:

卸载掉旧版的软件，然后安装新版docke

```
[root@docker ~]# yum erase docker-common-2:1.13.1-102.git7f2769b.el7.centos.x86_64
```

## 3、镜像加速配置

#启动docker之前，配置镜像加速；镜像加速器(推荐阿里云)



容器镜像服务

---

▼ 默认实例

镜像仓库

命名空间

授权管理

代码源

访问凭证

▶ 企业版实例

▼ 镜像中心

镜像搜索

我的收藏

镜像加速器

镜像加速器

---

加速器

使用加速器可以提升获取Docker官方镜像的速度

加速器地址

https://[redacted].com
复制

操作文档

Ubuntu CentOS Mac Windows

1. 安装 / 升级Docker客户端

推荐安装 1.10.0 以上版本的Docker客户端, 参考文档 [docker-ce](#)

2. 配置镜像加速器

针对Docker客户端版本大于 1.10.0 的用户

您可以通过修改daemon配置文件 `/etc/docker/daemon.json` 来使用加速器

```
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<-' EOF'
{
  "registry-mirrors": ["https://[redacted].com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

```
[root@docker ~]# mkdir /etc/docker/
```

```
[root@docker ~]# vim /etc/docker/daemon.json
{
  "registry-mirrors": ["https://xxxxxxxxx.aliyuncs.com"]
}
```

## 4、启动Docker

```
[root@docker ~]# systemctl enable docker.service;systemctl start docker.service
```

```
[root@docker ~]# docker version
```

#查看Docker详细信息

```
[root@docker ~]# docker info
```

```
Containers: 0      #当前系统中的容器数量
Running: 0        #其中运行状态的容器数量
Paused: 0        #其中暂停状态的容器数量
Stopped: 0       #其中停止状态的容器数量
Images: 0        #镜像数量
Server Version: 18.09.2  #服务器版本
Storage Driver: overlay2  #存储驱动后端, 专用的文件驱动
```

```
Backing Filesystem: xfs
Supports d_type: true
Native Overlay Diff: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins: #插件
Volume: local #存储插件
Network: bridge host macvlan null overlay #网络插件
Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog #日志插件
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
containerd version: 9754871865f7fe2f4e74d43e2fc7ccd237edcbce
runc version: 09c8266bf2fc9519a651b04ae54c967b9ab86ec
init version: fec3683
Security Options:
seccomp #安全
Profile: default
Kernel Version: 3.10.0-957.el7.x86_64
Operating System: CentOS Linux 7 (Core)
OSType: linux
Architecture: x86_64
CPUs: 4
Total Memory: 3.84GiB
Name: docker
ID: 35ZE:Y2HG:DYNW:4ZPK:OIZS:BDF6:6E5B:YOEL:7ZC2:LT5G:EE4D:FC2W
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): false
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
127.0.0.0/8
Registry Mirrors:
https://registry.docker-cn.com/ #刚才手动配置的加速镜像, 表明加速正常
Live Restore Enabled: false
Product License: Community Engine
```

- 以上就表明docker安装完成并且可以正常启动, 正常加速镜像
- 解决以下问题:

```
WARNING: bridge-nf-call-iptables is disabled
```

```
WARNING: bridge-nf-call-ip6tables is disabled
```

```
[root@docker ~]# vim /etc/sysctl.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
```

```
[root@docker ~]# sysctl -p
```

## 5、使用DaoCloud监控容器

官方地址: <https://dashboard.daocloud.io>

## 5.1: 执行脚本

```
[root@docker ~]# curl -sSL https://get.daocloud.io/daomonit/install.sh | sh -s 46d6b11a819f119cdda1242a672b1003340e468
```



1 请选择您的场景

DaoCloud 可以帮助您轻松接入遍布各地的计算资源, 无论主机的操作系统和网络环境。您要接入的主机属于下面哪一种情况?

已有一台新主机  去买一台新主机  免费试用胶囊主机

2 请选择主机上的操作类型

Ubuntu  CentOS  Windows  Mac  其它 Linux 系统

3 安装主机监控程序

安装好 Docker 后, 运行主机安装命令, [查看如何安装 Docker](#) **执行以下脚本**

```
curl -sSL https://get.daocloud.io/daomonit/install.sh | sh -s 46d6b11a819f119cdda1242a672b1003340e468
```

4 等待连接...

```
* Start Daomonit...
Starting daomonit (via systemctl): [ 确定 ]

You can view daomonit log at /var/log/daomonit.log
And You can Start or Stop daomonit with: service daomonit start/stop/restart/statu

*****
*****
***
*** Installed and Started Daomonit 0.1.70
***
*** NOTICE:
*** You can pull image very Fast by dao, For Example:
***   dao pull ubuntu
***
*****
*****
```



### 3 安装主机监控程序

安装好 Docker 后，运行主机安装命令。查看如何安装 Docker

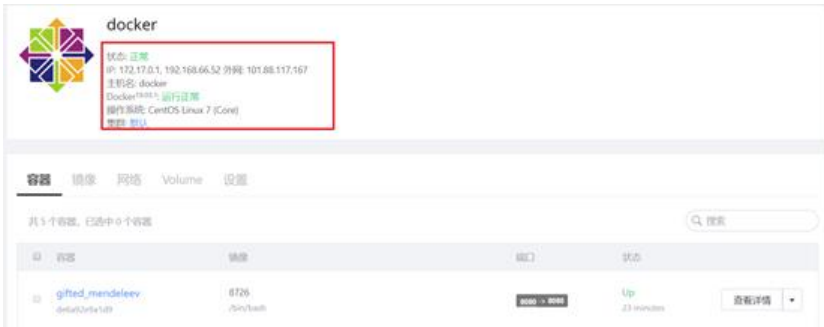
```
curl -sSL https://get.daocloud.io/daomoni/install.sh | sh -s 46d6b11a819f1119cdda1242a672b1003340e468
```

恭喜您接入成功!

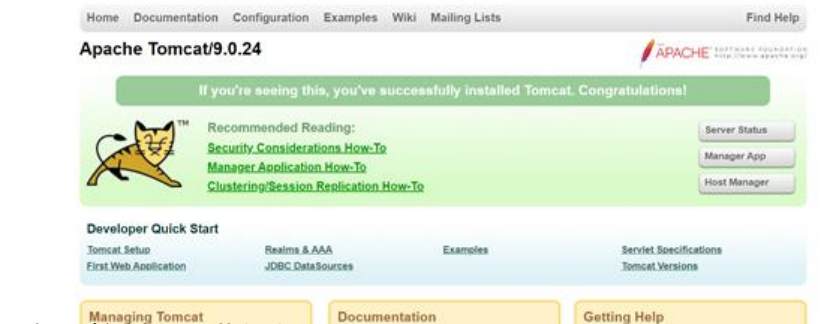
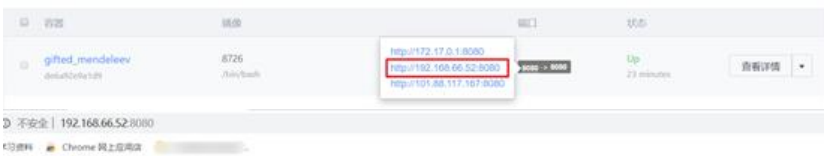
查看主机

继续接入

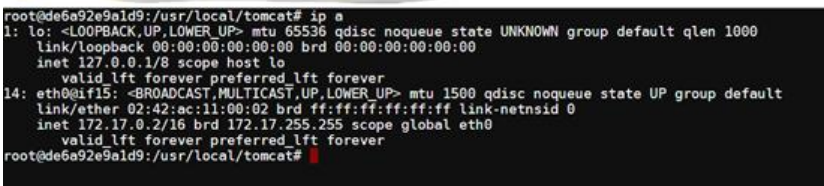
## 5.2: 查看主机



### ● 访问其中一个容器



### ● 打开控制台，进入容器



## 5.3: 删除主机

删除主机

删除操作仅适用于已经断开连接的主机。您可以通过下述命令加载主机监控程序，再进行删除。

```
dpkg -r daemontools # For Ubuntu/Debian  
rpm -e daemontools # For CentOS/Fedora
```

```
[root@docker ~]# rpm -e daemontools
```

## 二、Docker使用

### 1、搜索镜像

```
[root@docker ~]# docker search tomcat #search后面跟镜像名
```

| NAME | DESCRIPTION | STARS | OFFICIAL | AUTOMATED |
|------|-------------|-------|----------|-----------|
| 仓库名称 | 镜像名称        | 描述信息  |          | 镜像状态      |

### 2、拉取镜像

#默认拉取的是docker hub上的镜像

- 方式1:

```
[root@docker ~]# docker pull tomcat
```

- 方式2:

```
[root@docker ~]# docker image pull 镜像名
```

#拉取私有仓库镜像

```
[root@docker ~]# docker pull 私有地址:port/[namespace] 镜像名:tag
```

- 例如：下载第三方镜像 <https://quay.io/>

```
[root@docker ~]# docker pull quay.io/coreos/flannel:v0.10.0-amd64
```

### 3、查看本地镜像

```
[root@docker ~]# docker image list
```

```
[root@docker ~]# docker images
```

### 4、删除镜像

```
[root@docker ~]# docker rmi name:[tag]
```

注意：删除之前需要先删除正在使用该镜像的容器;不加tag则默认删除最后一个

删除一个或多个镜像之间用空格隔开

#### 4.1：强制删除

```
[root@docker ~]# docker rmi tomcat:9.0 --force
```

强制删除之后使用该镜像的容器也随之无法使用

## 4.2: 只查看镜像ID

```
[root@docker ~]# docker images -q
96c4e536d0eb
62a9f311b99c
```

## 5、查看容器状态

### 5.1: 查看正在运行的容器

```
[root@docker ~]# docker ps
```

### 5.2: 查看所有容器

```
[root@docker ~]# docker ps --all
```

## 6、停止容器

```
[root@docker ~]# docker container stop id(或名称)
```

## 7、重启容器

```
[root@docker ~]# docker container restart id(或名称)
```

## 8、删除容器

```
[root@docker ~]# docker container rm id(或名称)
```

- 强制删除

```
[root@docker ~]# docker container rm -f id(或名称)
```

## 9、只查看容器ID

- 查看所有

```
[root@docker ~]# docker ps -a -q
```

## 10、运行一个容器

```
docker run -it -d -p 8070:8080 --name web -v /root/software:/software --privileged=true 镜  
名 /bin/bash
```

-it 创建一个交互式的容器

-d 作为一个守护进程在后台运行

- p 映射端口8070 本机的端口 映射的容器的端口
- name 自定义容器名称
- v 挂载目录/root/software(本地目录)到/software(容器目录), 在创建前容器是没有software目录的 docker 容器会自己创建
- privileged=true 关闭安全权限, 否则你容器操作文件夹没有权限

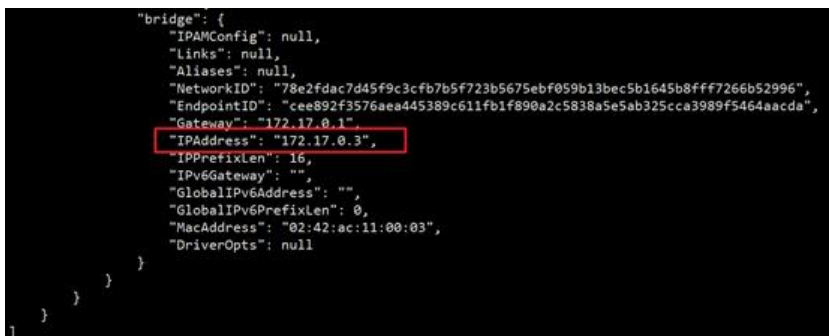
## 11、查看容器IP

- 方式1: 登入容器中查看

```
root@3842ef70fd14:/# cat /etc/hosts
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.3 3842ef70fd14
```

- 方式2: 外部使用命令

```
[root@docker ~]# docker inspect <container id> #容器ID或容器名
```



```
"bridge": {
  "IPAMConfig": null,
  "Links": null,
  "Aliases": null,
  "NetworkID": "78e2fdac7d45f9c3cfb7b5f723b5675ebf059b13bec5b1645b8fff7266b52996",
  "EndpointID": "cee892f3576aea445389c611fb1f890a2c5838a5e5ab325cca3989f5464aacda",
  "Gateway": "172.17.0.1",
  "IPAddress": "172.17.0.3",
  "IPPrefixLen": 16,
  "IPv6Gateway": "",
  "GlobalIPv6Address": "",
  "GlobalIPv6PrefixLen": 0,
  "MacAddress": "02:42:ac:11:00:03",
  "DriverOpts": null
}
```

- 方式3: 获取所有容器IP和名称

```
[root@docker ~]# docker inspect -f '{{.Name}} - {{.NetworkSettings.IPAddress }}' $(docker ps -q)
```

```
[root@docker ~]# docker inspect -f '{{.Name}} - {{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' $(docker ps -aq)
```

- 方式4: 显示所有容器IP地址

```
[root@docker ~]# docker inspect --format='{{.Name}} - {{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' $(docker ps -aq)
```

## 12: 登入容器

```
docker exec -it <容器名或ID> /bin/bash
```

```
[root@docker ~]# docker exec -it web /bin/bash
root@55b8603a6e54:/usr/local/tomcat#
```

## 13: 文件拷贝

```
[root@docker ~]# docker cp 拷贝的文件名称 文件在宿主机路径:拷贝至容器中的路径 容器名称或D
```

## 14: 安装vim

```
root@55b8603a6e54:~# apt-get update  
root@55b8603a6e54:~# apt-get install vim
```

- 命令说明:

命令的作用是: 同步 /etc/apt/sources.list 和 /etc/apt/sources.list.d 中列出的源的索引, 这样才能取到最新的软件包。

## 15: 重命名容器名

```
[root@docker ~]# docker container <原容器名> <新容器名>
```

#查看是否修改成功

```
[root@docker ~]# docker container ps
```

## 16: 查看容器占用空间大小

```
[root@docker ~]# docker container ps -as
```

#配置grep、awk进行过滤也可以

# 三、镜像制作

- Create a new image from a container's changes
- Usage
  - docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]

| Name, shorthand | Default | Description  |
|-----------------|---------|--|
| --author, -a    |         | Author (e.g., "John Hannibal Smith hannibal@a-team.com") |
| --change, -c    |         | Apply Dockerfile instruction to the created image        |
| --message, -m   |         | Commit message   |
| --pause, -p     | true    | Pause container during commit                            |

## 1、基于dockerfile

使用docker build命令制作

## 2、基于容器

- 进入到容器中修改配置, 使其符合当前业务, 然后再基于此运行的容器制作作为镜像
- 例如在tomcat中添加一个索引文件, 然后基于该更改后的容器进行制作镜像

注意：制作镜像时，添加-p选项暂停容器，防止制作镜像时文件不完整

```
[root@docker ~]# docker commit -h  
Flag shorthand -h has been deprecated, please use --help
```

Usage: docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]

Create a new image from a container's changes

Options:

- a, --author string Author (e.g., "John Hannibal Smith <hannibal@a-team.com>")
- c, --change list Apply Dockerfile instruction to the created image
- m, --message string Commit message
- p, --pause Pause container during commit (default true)

### ● 制作镜像

```
[root@docker ~]# docker commit -p web #web为修改后的容器名
```

```
[root@docker ~]# docker commit -p web  
sha256:b7f4719708427d4e3f10476574c6ea70135d5e3332ddeccce5f9e1d3bbfeb7db7  
[root@docker ~]# docker images  
REPOSITORY          TAG                IMAGE ID           CREATED            SIZE  
<none>              <none>            b7f471970842     9 seconds ago    554MB  
tomcat              latest            96c4e536d0eb     10 days ago      506MB  
mysql              5.7              e1e1680ac726     2 weeks ago      373MB  
mysql              latest           62a9f311b99c     2 weeks ago      445MB
```

### ● 添加tag

```
[root@docker ~]# docker tag b7f471970842 dqzboy/tomcat:v0.1  
[root@docker ~]# docker images
```

```
[root@docker ~]# docker tag b7f471970842 dqzboy/tomcat:v0.1  
[root@docker ~]# docker images  
REPOSITORY          TAG                IMAGE ID           CREATED            SIZE  
dqzboy/tomcat       v0.1              b7f471970842     6 minutes ago    554MB  
tomcat              latest            96c4e536d0eb     10 days ago      506MB  
mysql              5.7              e1e1680ac726     2 weeks ago      373MB  
mysql              latest           62a9f311b99c     2 weeks ago      445MB
```

### ● 添加新tag

一个镜像可以有多个tag，但是一个tag只能在一个镜像上

```
[root@docker ~]# docker tag dqzboy/tomcat:v0.1 dqzboy/tomcat:latest  
[root@docker ~]# docker images
```

```
[root@docker ~]# docker tag dqzboy/tomcat:v0.1 dqzboy/tomcat:latest  
[root@docker ~]# docker images  
REPOSITORY          TAG                IMAGE ID           CREATED            SIZE  
dqzboy/tomcat       latest            b7f471970842     11 minutes ago    554MB  
dqzboy/tomcat       v0.1             b7f471970842     11 minutes ago    554MB  
tomcat              latest            96c4e536d0eb     10 days ago      506MB  
mysql              5.7              e1e1680ac726     2 weeks ago      373MB  
mysql              latest           62a9f311b99c     2 weeks ago      445MB
```