



链滴

学学Docker

作者: [Hassan](#)

原文链接: <https://ld246.com/article/1471922153538>

来源网站: [链滴](#)

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



环境

Vmware11 安装 Centos 7 系统, 版本如下。

```
$ uname -s -r  
Linux 3.10.0-229.el7.x86_64
```

[安装](https://github.com/HassanChiang/Blog/blob/master/source/_posts/201602/glance-docker.md#安装)

1.使用root账户或具有root权限的用户登录。

2.更新安装包。

```
sudo yum update
```

3.给yum添加docker repository。

```
$ sudo tee /etc/yum.repos.d/docker.repo &&-'EOF'  
[dockerrepo]  
name=Docker Repository  
baseurl=https://yum.dockerproject.org/repo/main/centos/$releasever/  
enabled=1  
gpgcheck=1  
gpgkey=https://yum.dockerproject.org/gpg  
EOF
```

4.安装docker引擎

```
$ sudo yum install docker-engine
```

5.启动docker守护进程

```
sudo service docker start
```

6.验证docker安装和启动是否正确

```
$ sudo docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from hello-world  
a8219747be10: Pull complete  
91c95931e552: Already exists  
hello-world:latest: The image you are pulling has been verified. Important: image verification is a tech preview feature and should not be relied on to provide security.  
Digest: sha256:aa03e5d0d5553b4c3473e89c8619cf79df368babd1.7.1cf5daeb82aab55838d  
Status: Downloaded newer image for hello-world:latest  
Hello from Docker.  
This message shows that your installation appears to be working correctly.
```

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(Assuming it was not already locally available.)
3. The Docker daemon created a new container from that image which runs the

executable that produces the output you are currently reading.

4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

For more examples and ideas, visit:

<http://docs.docker.com/userguide/>

</div>

<p>ok, 看见以上信息, 就说明docker已经正确安装并启动。 </p>

<h3>在系统里创建一个docker用户组</h3>

<p>从0.5.2开始docker的守护进程总是以root用户来运行。docker守护进程绑定的是Unix的socket不是一个TCP端口。Unix的socket默认属于root用户, 所以, 使用docker时必须加上sudo。 </p>

<p>从0.5.3开始, 创建一个名为docker组, 然后将用户加入这个组内。当docker守护进程启动时, 会把Unix的读写权限赋予docker组。这样, 当你作为docker组内用户使用docker客户端时, 你就无使用sudo了。 </p>

<p>需要注意的是, docker 用户组是等价于root用户的, 进一步了docker用户组对系统安全的影响可以看看这个: 走你。 </p>

<p>执行: </p>

```
<pre class="brush: java">sudo usermod -aG docker your_username</pre>
```

<p>执行完后, 退出shell, 再重新登录, 以确保用户获得正确的运行权限。 </p>

<p>重新登录系统后, 执行: </p>

```
<pre class="brush: bash">docker run hello-world</pre>
```

<p>执行结果应该同“安装”部分第六步的执行结果一样。 </p>

<h3>设置docker守护进程开机启动</h3>

```
<pre class="brush: bash">$ sudo chkconfig docker on</pre>
```

<h3>运行一下官方教程提供的镜像</h3>

<p>执行: </p>

```
<pre class="brush: bash">$ docker run docker/whalesay cowsay boo
```

Unable to find image 'docker/whalesay:latest' locally

latest: Pulling from docker/whalesay

e9e06b06e14c: Pull complete

a82efea989f9: Pull complete

37bea4ee0c81: Pull complete

07f8e8c5e660: Pull complete

676c4a1897e6: Pull complete

5b74edbcaa5b: Pull complete

1722f41ddcb5: Pull complete

99da72cfe067: Pull complete

5d5bd9951e26: Pull complete

fb434121fc77: Already exists

Digest: sha256:d6ee73f978a366cf97974115abe9c4099ed59c6f75c23d03c64446bb9cd49163

Status: Downloaded newer image for docker/whalesay:latest

< boo >

是依靠容器ID识别的，由于容器ID的字符太长，我们通常只需键入容器ID的前4个字符即可。当然，们还可以使用容器名，但显然用4字符的容器ID更为简便。

查看当前运行的容器可以使用命令：

```
$ docker ps
CONTAINER ID   IMAGE                                COMMAND                                CREATED        STATUS
PORTS         NAMES
a45a2d3b95e7   sample-spring-boot-app             "/bin/sh -c 'java -DC" 3 seconds ago  Up 2
econds        0.0.0.0:8080->8080/tcp              silly_morse
0b2494881e6c   sample-spring-boot-app             "/bin/sh -c 'java -DC" 3 minutes ago  Up 3
minutes      0.0.0.0:32771->8080/tcp              compassionate_austin
```

简单的理解，可以把“镜像-容器”理解为“类-实例”。容器是一个例，这个实例是根据镜像创建的。