



链滴

Docker 入门的一些常用命令

作者: [seanlee](#)

原文链接: <https://ld246.com/article/1512983208494>

来源网站: 链滴

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

<h3 id="Docker镜像命令">Docker 镜像命令</h3>

<h4 id="简易安装模式">简易安装模式</h4>

<p>CentOS 安装 命令: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight cl">sudo yum update sudo yum install docker</span></span></code></pre>
```

<p>Ubuntu 安装命令: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight cl">sudo apt- get update sudo apt- get docker. io</span></span></code></pre>
```

<h4 id="docker-镜像检索及下载">docker 镜像检索及下载</h4>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight cl">docker search 镜像名</span></span></code></pre>
```

<p>如检索 mongodb 时 <code>docker search mongodb</code> </p>

<p>镜像下载</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight cl">docker pull 镜像名</span></span></code></pre>
```

<p>如下载 mongodb 镜像 <code>docker pull mongo</code> </p>

<h4 id="修改默认的镜像仓库">修改默认的镜像仓库</h4>

<p>参考: Docker 中国官方镜像加速

由于 docker 官方的镜像仓库再国外, 国内访问及下载速度会很慢, 可以修改默认的镜像仓库地址以取更快的下载速度。通过 Docker 官方镜像加速, 中国区用户能够快速访问最流行的 Docker 镜像。镜像托管于中国大陆, 本地用户现在将会享受到更快的下载速度和更强的稳定性, 从而能够更敏捷地发和交付 Docker 化应用。 </p>

<h4 id="方式一-使用命令时指定地址-">方式一: 使用命令时指定地址: </h4>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight cl">$ docker pull registry.docker-cn.com/myname/myrepo:mytag</span></span></code></pre>
```

<p>例如: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight cl">$ docker pull registry.docker-cn.com/library/ubuntu:16.04</span></span></code></pre>
```

<blockquote>

<p>注: 除非修改了 Docker 守护进程的 <code>--registry-mirror</code> 参数 (见下文), 否则需完整地指定官方镜像的名称。例如, library/ubuntu、library/redis、library/nginx。 </p>

</blockquote>

<h4 id="方式二-使用---registry-mirror-配置-Docker-守护进程">方式二: 使用 --registry-mirror 配置 Docker 守护进程</h4>

<p>配置 Docker 守护进程默认使用 Docker 官方镜像加速。这样可以默认通过官方镜像加速拉取镜, 而无需在每次拉取时指定 registry.docker-cn.com。 </p>

<p>一、 Docker 守护进程启动时传入 --registry-mirror 参数: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight cl">
```

```
cl">$ docker --registry-mirror=https://registry.docker-cn.com daemon
```

```
</span></span></code></pre>
```

二、为了永久性保留更改，您可以修改 /etc/docker/daemon.json 文件并添加上 registry-mirrors 键值。

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> "registry-mirrors"
```

```
["https://registry.docker-cn.com"]
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">}
```

```
</span></span></code></pre>
```

修改保存后重启 Docker 以使配置生效。

查看镜像列表及删除

[查看镜像列表及删除](#)

查看本地镜像列表

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">docker images
```

```
</span></span></code></pre>
```

删除特定镜像

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">docker rmi image-id
```

```
</span></span></code></pre>
```

删除所有镜像

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">docker rmi $(docker images -q)
```

```
</span></span></code></pre>
```

Dockers 容器命令

[Dockers 容器命令](#)

容器启动

[容器启动](#)

简单的启动容器的命令

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">docker run --name container-name -d image-name
```

```
</span></span></code></pre>
```

container-name 是你为容器起的名字，image-name 是你下载下来镜像的名字

例如：

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">docker run --name test-mongodb -d mongo
```

```
</span></span></code></pre>
```

查看容器列表

[查看容器列表](#)

通过命令查看运行中的容器列表

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">docker ps
```

```
</span></span></code></pre>
```

查看运行和停止状态的容器列表

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">docker ps -a
```

```
</span></span></code></pre>
```

停止和启动容器

[停止和启动容器](#)

启动容器

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">docker start container-name/container-id
```

```
</span> </span> </code> </pre>
```

<p>停止容器</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> docker stop container-name/container-id
```

```
</span> </span> </code> </pre>
```

<h4 id="端口映射"> 口映射</h4>

<p>因为应用程序是跑在容器当中的，应用再容器中使用的端口并不是服务器主机本身的端口，因此要将容器的端口映射为服务器本身的端口。

docker 的端口映射是使用-p 参数完成的，命令如下： </p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> docker run -d -p 27017:27017 --name port-mongodb mongo
```

```
</span> </span> </code> </pre>
```

<h4 id="删除容器"> 除容器</h4>

<p>删除单个容器</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> docker rm container-id
```

```
</span> </span> </code> </pre>
```

<p>删除所有容器</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> docker rm $(docker ps -a -q)
```

```
</span> </span> </code> </pre>
```

<h4 id="查看容器日志"> 查看容器日志</h4>

<p>查看当前容器的日志</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> docker logs container-name/ container-id
```

```
</span> </span> </code> </pre>
```

<h4 id="登陆容器"> 登 容器</h4>

<p>运行中的容器其实是一个功能完备的 Linux 操作系统，所以我们可以像 常规的系统一样登录并 问容器。 </p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> docker exec -it container-id/container-name bash
```

```
</span> </span> </code> </pre>
```