

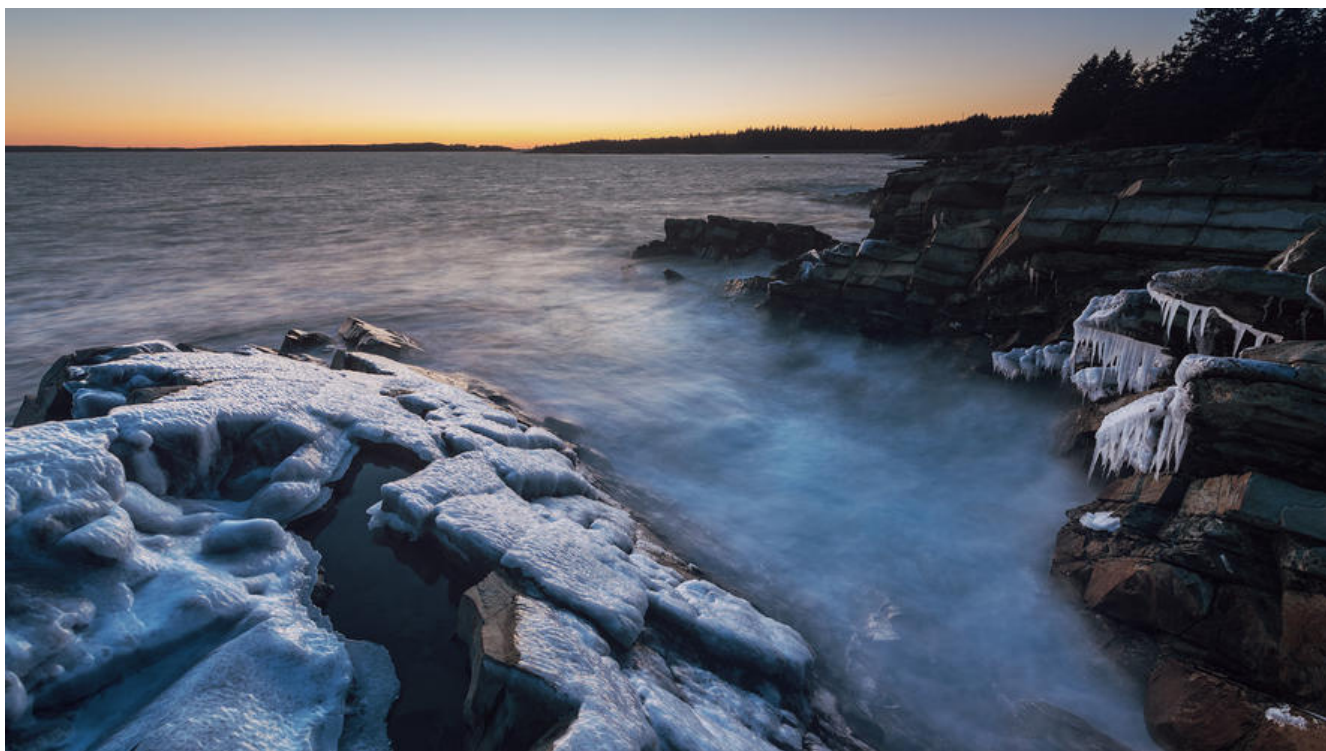
Docker 学习（一）入门篇

作者: [wky181](#)

原文链接: <https://ld246.com/article/1569816031155>

来源网站: [链滴](#)

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



什么是Docker

Docker 是基于Go语言实现的开源容器项目。利用操作系统本身已有的机制和特性，可以实现远超传统虚拟机的轻量级虚拟化。它是内核级的虚拟化。期望达到使项目运行环境“一次封装，到处运行的目的”。

利用docker创建的运行环境叫做docker容器，容器是通过docker镜像创建的，docker镜像文件可以在私有仓库中也可以放在共有仓库中。最大的公有仓库是官方[Docker Hub](#)。

可以将Docker容器理解为一种轻量级的沙盒，每个容器内运行着一个应用，不同容器之间相互隔离容器之间也可以通过网络互相通信。容器的创建和停止十分迅速，几乎和原生应用一样。

为什么要用Docker

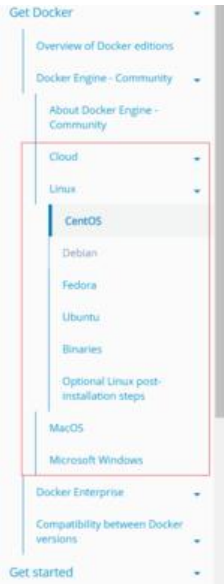
在云时代，开发者创建的应用必须要能很方便地在网络上部署，如果能够不受服务器的操作系统的影和服务器是否安装应用所需环境，让应用在“任何时间任何地点”都是可快速部署的，这正是 Docker 所能够提供的最大优势。

举个简单的例子，假设用户试图基于最常见的 LAMP (Linux+Apache+MySQL+PHP) 组合来构建网站。按照传统的做法，首先需要安装 Apache、MySQL 和 PHP 以及它们各自运行所依赖的环境之后分别对它们进行配置（包括创建合适的用户、配置参数等）；一旦需要服务器迁移（例如从阿里迁移到其他云），往往需要对每个应用都进行重新部署和调试。极大地降低了用户的工作效率。究其源，是这些应用直接运行在底层操作系统上，无法保证同一份应用在不同的环境中行为一致。而 Docker 提供了一种更为聪明的方式，通过容器来打包应用、解耦应用和运行平台。这意味着迁移的时候，需要在新的服务器上启动需要的容器就可以了，无论新旧服务器是否是同一类型的平台。

Docker的安装

[docker官方文档](#)

这个是docker官方的安装文档，这个连接是centos版本的，可以在侧边栏选择其他的版本。然后跟文档一步一步来就行了，这里以centos为例。



Get Docker Engine - Community for CentOS

Estimated reading time: 10 minutes

To get started with Docker Engine - Community on CentOS, make sure you meet the prerequisites, then install Docker.

Prerequisites

Docker EE customers

To install Docker Enterprise Edition (Docker EE), go to [Get Docker EE for CentOS instead of this topic](#).

To learn more about Docker EE, see [Docker Enterprise Edition](#).

OS requirements

To install Docker Engine - Community, you need a maintained version of CentOS 7. Archived versions aren't supported or tested.

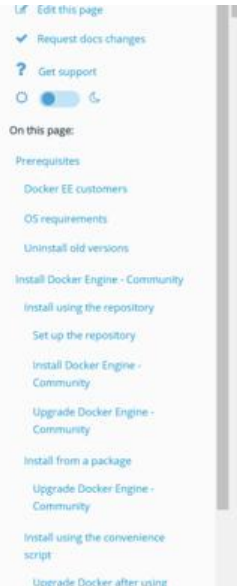
The `centos-extras` repository must be enabled. This repository is enabled by default, but if you have disabled it, you need to re-enable it.

The `overlay2` storage driver is recommended.

Uninstall old versions

Older versions of Docker were called `docker` or `docker-engine`. If these are installed, uninstall them, along with associated dependencies.

```
$ sudo yum remove docker \
    docker-client \
    docker-client-latest \
```



1. 先卸载旧版本

```
$ sudo yum remove docker \
    docker-client \
    docker-client-latest \
    docker-common \
    docker-latest \
    docker-latest-logrotate \
    docker-logrotate \
    docker-engine
```

2. 添加repo

```
$ sudo yum-config-manager \
    --add-repo \
    https://download.docker.com/linux/centos/docker-ce.repo
```

中间的optional (可选的) 直接跳过。

3. 安装docker-ce

```
$ sudo yum install docker-ce docker-ce-cli containerd.io
```

这个步骤，我在安装的时候总是超时，安装不上，最后在网上找到的解决方法。根据下面这个链接把像站换成国内就行啦。<https://mirror.tuna.tsinghua.edu.cn/help/docker-ce/>

4. 启动docker

```
$ sudo systemctl start docker
```

5. 验证docker是否安装成功

```
$ sudo docker -v
```

Hello World

Docker安装成功后，按照以下代码运行，就会打印出 Hello from Docker!，具体是怎么一回事，我以后会讲。

```
docker pull hello-world
docker run hello-world
```

```
[root@localhost ~]# docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:92c7f9c92844bbbb5d0a101b22f7c2a7949e40f8ea90c8b3bc396879d95e899a
Status: Image is up to date for hello-world:latest
[root@localhost ~]# docker run hello-world
WARNING: IPv4 forwarding is disabled. Networking will not work.

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.
    ( amd64)
 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

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

ps: 我在执行docker pull hello-world命令时，竟然报错了！

```
[root@localhost docker]# docker pull hello-world
Using default tag: latest
Error response from daemon: Get https://registry-1.docker.io/v2/library/hello-world/manifests/latest: Get https://auth.docker.io/token?scope=repository%3Alibrary%2Fhello-world%3Apull&service=registry.docker.io: net/http: TLS handshake timeout
```

好像还是因为镜像问题。按照这篇[博客](#)解决了问题。具体就是查看 /etc/docker/ 下有没有 daemon.json，若没有就新建一个，然后#vim/etc/docker/daemon.json，添加下面的内容。

```
{
  "registry-mirrors": ["https://registry.docker-cn.com","http://hub-mirror.c.163.com"]
}
```

重启docker服务

```
restart docker.service
```