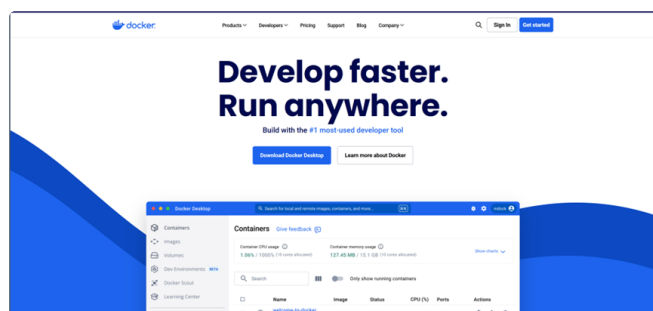


# Docker入门手册

## 1 什么是Docker

Docker是基于Go语言实现的开源容器项目。它诞生于2013年年初，最初发起者是dotCloud 公司。Docker自开源后受到业界广泛的关注和参与，目前已有80多个相关开源组件项目（包括Containerd、Moby、Swarm等），逐渐形成了围绕Docker容器的完整的生态体系。dotCloud公司也随之快速发展壮大，在2013年年底直接改名为Docker Inc，并专注于Docker相关技术和产品的开发，目前已经成为全球最大的Docker容器服务提供商。官方网站为<https://www.docker.com/>，如下图所示。



现在主流的操作系统包括Linux各大发行版、macOS、Windows等都已经支持Docker。

## 2 Docker安装配置

### 2.1 安装Docker引擎

Docker 引擎是使用Docker容器的核心组件，可以在主流的操作系统和云平台上使用，包括Linux操作系统（如Ubuntu、Debian、CentOS、Redhat等）， macOS和Windows操作系统，以及IBM、亚马逊、微软等知名云平台。用户可以访问Docker官网的[Get Docker](#) 页面，查看获取Docker的方式，以及Docker支持的平台类型。

### 2.2 Ubuntu 环境下安装Docker

Ubuntu操作系统对Docker 的支持十分成熟，可以支持包括x86\_64、armhf、s390x(IBMZ)、ppc64le等系统架构，只要是64 位即可。

Docker 目前支持的最低Ubuntu 版本为14.04 LTS,但实际上从稳定性上考虑，推荐使用16.04.LTS或18.0.4 LTS 版本，并且系统内核越新越好，以支持Docker最新的特性。具体步骤如下：

- Ubuntu系统版本：20.04LTS
- Docker安装版本：26.0.0

参考链接：

- ubuntu安装Docker-CSDN博客
- 【Docker】在 Ubuntu 上安装 Docker 的详细指南\_ubuntu安装docker-CSDN博客

注意：本平台在Ubuntu系统下安装Docker，如无特殊情况，建议使用Ubuntu系统安装，否则可能无法复现。

## 2.3 CentOS 环境下安装Docker

Docker目前支持CentOS 7及以后的版本。系统的要求跟Ubuntu情况类似，64 位操作系统，内核版本至少为3.10。

首先，为了方便添加软件源，以及支持devicemapper存储类型，安装如下软件包：

```
sudo yum update
sudo yum install -y yum-utils device-mapper-persistent-data lvm2
```

添加Docker稳定版本的yum软件源：

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

之后更新yum软件源缓存，并安装Docker：

```
sudo yum update
sudo yum install -y docker-ce
```

最后，确认Docker 服务启动正常：

```
sudo systemctl start docker
```

## 2.4 Windows环境下安装Docker

目前Docker可以通过虚拟机方式来支持Windows 7.1和Windows 8，只要平台CPU支持硬件虚拟化特性即可。如果无法确定自己计算机的CPU是否支持该特性也无须担心，实际上，目前市面上主流的CPU都早已支持硬件虚拟化特性。

对Windows 10的用户来说， Docker 官方为64位Windows 10 Pro环境（需支持Hyper-V）提供了原生虚拟化应用Docker for Windows。 Windows 环境下Docker CE同样支持两个版本：稳定版和测试版。这两个版本的异同可以参见上节。下载地址可见官网文档中心的Get Docker-Docker CE-Windows 。安装步骤可参考以下链接：

- Windows安装使用Docker，方便你的开发和部署(DockerDesktop篇)\_windows安装docker-CSDN博客

## 3 使用Docker镜像和容器

镜像是Docker三大核心概念中最重要的，自Docker诞生之日起镜像就是相关社区最为热门的关键词。Docker运行容器前需要本地存在对应的镜像，如果镜像不存在，Docker会尝试先从默认镜像仓库下载（默认使用Docker Hub公共注册服务器中的仓库），用户也可以通过配置，使用自定义的镜像仓库。

容器是Docker的另一个核心概念。简单来说，容器是镜像的一个运行实例。所不同的是，镜像是静态的只读文件，而容器带有运行时需要的可写文件层，同时，容器中的应用进程处于运行状态。

### 3.1 获取镜像

镜像是运行容器的前提，官方的Docker Hub网站已经提供了数十万个镜像供大家开放下载。（需要一些手段才能访问得到）

如果Docker Hub访问不到的话，可以访问国内的一些镜像网站，具体可参考以下链接：

1. 🔥 截止目前，国内仍然可用docker镜像加速器汇总（2024年12月） - Kelen
  2. Docker/DockerHub 国内镜像源/加速列表（12月3日更新-长期维护） - 轩源的网络日志
- 可以使用 `docker [image] pull` 命令直接从Docker Hub镜像源来下载镜像。该命令的格式为 `docker [image] pull NAME[:TAG]`。
- 其中，`NAME` 是镜像仓库名称（用来区分镜像），`TAG` 是镜像的标签（往往用来表示版本信息）。通常情况下，描述一个镜像需要包括“名称+标签”信息。
- 例如，获取一个Ubuntu 18.04系统的基础镜像可以使用如下的命令：

```
docker pull ubuntu:18.04
# 下为输出
# 18.04: Pulling from library/ubuntu
# ...
# Digest:sha256:e27e9d7f7f2Bd67aa9e2d7540bdc2b33254b452ee8e60f388875e5b7d9b2b696
# Status: Downloaded newer image for ubuntu:18.04
```

如果从非官方的仓库下载，则需要在仓库名称前指定完整的仓库地址。例如从网易蜂巢的镜像源来下载 `ubuntu:18.04` 镜像，可以使用如下命令，此时下载的镜像名称为

`hub.c.163.com/public/ubuntu:18.04`：

```
docker pull hub.c.163.com/public/ubuntu:18.04
```

### 3.2 docker镜像和容器的基本命令

Docker镜像和容器的命令众多，网上也有丰富的资料可以查询，具体可参考以下链接：

1. Docker篇——Docker常用命令大全！史上最全面的Docker命令，涵盖常规命令、镜像基本命令、容器基本命令以及常用其他命令，一篇文章速通！\_有docker status的命令有哪些-CSDN博客
2. Docker-容器常用命令\_docker 容器的相关命令-CSDN博客

## 4 访问Docker仓库

仓库（Repository）是集中存放镜像的地方，又分公共仓库和私有仓库。有时候容易把仓库与注册服务器（Registry）混淆。实际上注册服务器是存放仓库的具体服务器，一个注册服务器上可以有多个仓库，而每个仓库下面可以有多个镜像。从这方面来说，仓库可以被认为是一个具体的项目或目录。例如对于仓库地址 `private-docker.com/ubuntu` 来说，`private-docker.com` 是注册服务器地址，`ubuntu` 是仓库名。

### 4.1 Docker Hub 公共镜像市场

Docker Hub是 Docker官方提供的最大的公共镜像仓库，目前包括了超过100 000的镜像，地址为 <https://hub.docker.com>。

#### 4.1.1 登录

可以通过命令行执行 `docker login` 命令来输入用户名、密码和邮箱来完成注册和登录。注册成功后，本地用户目录下会自动创建 `.docker/config.json` 文件，保存用户的认证信息。登录成功的用户可以上传个人制作的镜像到 Docker Hub。

#### 4.1.2 基本操作

用户无须登录即可通过 `docker search` 命令来查找官方仓库中的镜像，并利用 `docker [image] pull` 命令来将它下载到本地。根据是否为官方提供，可将这些镜像资源分为两类：

1. 一种是类似于 `centos` 这样的基础镜像，也称为根镜像。这些镜像是由 Docker 公司创建、验证、支持、提供，这样的镜像往往使用单个单词作为名字；
2. 另一种类型的镜像，比如 `ansible/centos7-ansible` 镜像，是由Docker用户 `ansible` 创建并维护的，带有用户名称为前缀，表明是某用户下的某仓库。

用户也可以在登录后通过 `docker push` 命令来将本地镜像推送到 Docker Hub。

#### 4.1.3 4.1.3.自动创建

自动创建（Automated Builds）是 Docker Hub提供的自动化服务，这一功能可以自动跟随项目代码的变更而重新构建镜像。要配置自动创建，包括如下的步骤：

1. 创建并登录 Docker Hub，以及目标网站如 Github;

2. 在目标网站中允许 Docker Hub 访问服务；
3. 在 Docker Hub 中配置一个“自动创建”类型的项目；
4. 选取一个目标网站中的项目（需要含 Dockerfile ）和分支；
5. 指定 Dockerfile 的位置，并提交创建。

之后，可以在 Docker Hub 的“自动创建”页面中跟踪每次创建的状态。

## 4.2 第三方镜像市场

### 4.2.1 查看镜像

访问 <https://hub.tenxcloud.com>，即可看到已存在的仓库和存储的镜像，包括 Ubuntu、Java、Mongo、MySQL、Nginx 等热门仓库和镜像。时速云官方仓库中的镜像会保持与 DockerHub 中官方镜像的同步。

### 4.2.2 下载镜像

下载镜像也是使用 `docker pull` 命令，但是要在镜像名称前添加注册服务器的具体地址。格式为 `index.tenxcloud.com/<namespace>/<repository>:<tag>`。

例如，要下载 Docker 官方仓库中的 `node:latest` 镜像，可以使用如下命令：

```
docker pull index.tenxcloud.com/docker_library/node:latest
```

## 4.3 搭建本地私有仓库

### 4.3.1 使用 registry 镜像创建私有仓库

安装 Docker 后，可以通过官方提供的 registry 镜像来简单搭建一套本地私有仓库环境：

```
docker run -d -p 5000:5000 registry:2
```

这将自动下载并启动一个 `registry` 容器，创建本地的私有仓库服务。默认情况下，仓库会被创建在容器的 `/var/lib/registry` 目录下。可以通过 `-v` 参数来将镜像文件存放在本地的指定路径。例如下面的例子将上传的镜像放到 `/opt/data/registry` 目录：

```
docker run -d -p 5000:5000 -v /opt/data/registry:/var/lib/registry registry:2
```

此时，在本地将启动一个私有仓库服务，监听端口为 5000。

仓库是集中维护容器镜像的地方，为 Docker 镜像文件的分发和管理提供了便捷的途径。

## 5 五、相关资源和教程

### 书籍

- [Docker技术入门与实践（第三版） 前言 · Docker -- 从入门到实践](#)
- [深入浅出Docker 百度网盘链接](#)

### 官方用户手册

- [手册 -- Docker官方文档|Docker中文文档|Docker中文文档|Docker官方教程](#)

### 教程

- [Docker 教程 | 菜鸟教程](#)
- [Docker最新超详细版教程通俗易懂-Docker最新超详细版 通俗易懂，快速上手的Docker基础+进阶（零基础入门必备）从入门到精通\\_哔哩哔哩\\_bilibili](#)
- [只要一小时，零基础入门Docker - 知乎](#)

### 进阶教程

- [超详细docker高级教程\(全\)\\_docker教程-CSDN博客](#)
- [Docker网络模式及配置（高级网络功能）\\_docker 主机网络 resolve-CSDN博客](#)
- [docker——libnetwork插件网络功能 - 明王不动心 - 博客园](#)
- [Docker官方三剑客：Compose、Machine和Swarm\\_docker compose、docker swarm是什么-CSDN博客](#)