

# 安装 Docker

官方网站上有各种环境下的安装指南，这里主要介绍下 Ubuntu、Debian 和 CentOS 系列的安装。

## 版本及系统要求

### Docker版本说明

2017/3/3，Docker官方发表了一篇博客，Docker版本从1.13.\*直接跳入17.03，该版本的意思是17年3月。同时，还声明了Docker以后会以CE（Community Edition）和EE（Enterprise Edition）的形式发布。其中，CE版本每个月发布一次，也就是说，随后的版本就是17.03、17.04、17.05等，而EE的版本是每三个月发布一次，对应的就是17.03、17.06等。对于发布的每个EE版本，Docker官网都会提供一年的技术支持。

### CE和EE对各OS发行版的支持情况(V17.03)

Platform	Docker EE	Docker CE
Ubuntu	yes	yes
Debian		yes
Red Hat Enterprise Linux yes	yes	
CentOS	yes	yes
Fedora		yes
Oracle Linux	yes	
SUSE Linux Enterprise Server	yes	
Microsoft Windows Server 2016	yes	
Microsoft Windows 10		yes
macOS		yes
Microsoft Azure	yes	yes
Amazon Web Services	yes	yes

## CentOS 操作系统安装 Docker

### 系统要求

Docker 最低支持 CentOS 7。

Docker 需要安装在 64 位的平台，并且内核版本不低于 3.10。CentOS 7 满足最低内核的要求，但由于内核版本比较低，部分功能（如 **overlay2** 存储层驱动）无法使用，并且部分功能可能不太稳定。

## 使用脚本自动安装

Docker 官方为了简化安装流程，提供了一套安装脚本，CentOS 系统上可以使用这套脚本安装：

```
curl -sSL https://get.docker.com/ | sh
```

执行这个命令后，脚本就会自动的将一切准备工作做好，并且把 Docker 安装在系统中。

不过，由于伟大的墙的原因，在国内使用这个脚本可能会出现某些下载出现错误的情况。国内的一些云服务商提供了这个脚本的修改版本，使其使用国内的 Docker 软件源镜像安装，这样就避免了墙的干扰。

## 阿里云的安装脚本

```
curl -sSL http://acs-public-mirror.oss-cn-hangzhou.aliyuncs.com/docker-engine/internet | sh -
```

## DaoCloud 的安装脚本

```
curl -sSL https://get.daocloud.io/docker | sh
```

## 手动安装

### 添加内核参数

默认配置下，在 CentOS 使用 Docker 可能会碰到下面的这些警告信息：

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

添加内核配置参数以启用这些功能。

```
$ sudo tee -a /etc/sysctl.conf <<-EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
```

然后重新加载 **sysctl.conf** 即可

```
$ sudo sysctl -p
```

## 添加 yum 源

虽然 CentOS 软件源 **Extras** 中有 Docker，名为 **docker**，但是不建议使用系统源中的这个版本，它的版本相对比较陈旧，而且并非 Docker 官方维护的版本。因此，我们需要使用 Docker 官方提供的 CentOS 软件源。

执行下面的命令添加 **yum** 软件源。

```
$ sudo tee /etc/yum.repos.d/docker.repo <<- 'EOF'
[dockerrepo]
name=Docker Repository
baseurl=https://yum.dockerproject.org/repo/main/centos/7/
enabled=1
gpgcheck=1
gpgkey=https://yum.dockerproject.org/gpg
EOF
```

## 安装 Docker

更新 **yum** 软件源缓存，并安装 **docker-engine**。

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

## 启动 Docker 引擎

```
$ sudo systemctl enable docker
$ sudo systemctl start docker
```

## 建立 docker 用户组

默认情况下，**docker** 命令会使用 **Unix socket** 与 Docker 引擎通讯。而只有 **root** 用户和 **docker** 组的用户才可以访问 Docker 引擎的 **Unix socket**。出于安全考虑，一般 Linux 系统上不会直接使用 **root** 用户。因此，更好地做法是将需要使用 **docker** 的用户加入 **docker** 用户组。

建立 **docker** 组：

```
$ sudo groupadd docker
```

将当前用户加入 **docker** 组：

```
$ sudo usermod -aG docker $USER
```

## 参考文档

参见 [Docker 官方 CentOS 安装文档](#)。

# Ubuntu、Debian 系列安装 Docker

## 系统要求

Docker 支持以下版本的 [Ubuntu](#) 和 [Debian](#) 操作系统：

- Ubuntu Xenial 16.04 (LTS)
- Ubuntu Trusty 14.04 (LTS)
- Ubuntu Precise 12.04 (LTS)
- Debian testing stretch (64-bit)
- Debian 8 Jessie (64-bit)
- Debian 7 Wheezy (64-bit)（*必须启用 backports*）

Ubuntu 发行版中，LTS（Long-Term-Support）长期支持版本，会获得 5 年的升级维护支持，这样的版本会更稳定，因此在生产环境中推荐使用 LTS 版本。

Docker 目前支持的 Ubuntu 版本最低为 12.04 LTS，但从稳定性上考虑，推荐使用 14.04 LTS 或更高的版本。

Docker 需要安装在 64 位的 x86 平台或 ARM 平台上（如[树莓派](#)），并且要求内核版本不低于 3.10。但实际上内核越新越好，过低的内核版本可能会出现部分功能无法使用，或者不稳定。

用户可以通过如下命令检查自己的内核版本详细信息：

```
$ uname -a
Linux device 4.4.0-45-generic #66~14.04.1-Ubuntu SMP Wed Oct 19 15:05:38 UTC
2016 x86_64 x86_64 x86_64 GNU/Linux
```

## 升级内核

如果内核版本过低，可以用下面的命令升级系统内核。

### Ubuntu 12.04 LTS

```
sudo apt-get install -y --install-recommends linux-generic-lts-trusty
```

### Ubuntu 14.04 LTS

```
sudo apt-get install -y --install-recommends linux-generic-lts-xenial
```

## Debian 7 Wheezy

Debian 7 的内核默认为 3.2，为了满足 Docker 的需求，应该安装 **backports** 的内核。

执行下面的命令添加 **backports** 源：

```
$ echo "deb http://http.debian.net/debian wheezy-backports main" | sudo tee  
/etc/apt/sources.list.d/backports.list
```

升级到 **backports** 内核：

```
$ sudo apt-get update  
$ sudo apt-get -t wheezy-backports install linux-image-amd64
```

## Debian 8 Jessie

Debian 8 的内核默认为 3.16，满足基本的 Docker 运行条件。但是如果打算使用 **overlay2** 存储层驱动，或某些功能不够稳定希望升级到较新版本的内核，可以添加 **backports** 源，升级到新版本的内核。

执行下面的命令添加 **backports** 源：

```
$ echo "deb http://http.debian.net/debian jessie-backports main" | sudo tee  
/etc/apt/sources.list.d/backports.list
```

升级到 **backports** 内核：

```
$ sudo apt-get update  
$ sudo apt-get -t jessie-backports install linux-image-amd64
```

需要注意的是，升级到 **backports** 的内核之后，会因为 **AUFS** 内核模块不可用，而使用默认的 **devicemapper** 驱动，并且配置为 **loop-lvm**，这是不推荐的。因此，不要忘记安装 Docker 后，配置 **overlay2** 存储层驱动。

## 配置 GRUB 引导参数

在 Docker 使用期间，或者在 **docker info** 信息中，可能会看到下面的警告信息：

```
WARNING: Your kernel does not support cgroup swap limit. WARNING: Your
```

```
kernel does not support swap limit capabilities. Limitation discarded.
```

或者

```
WARNING: No memory limit support  
WARNING: No swap limit support  
WARNING: No oom kill disable support
```

如果需要这些功能，就需要修改 GRUB 的配置文件 `/etc/default/grub`，在 `GRUB_CMDLINE_LINUX` 中添加内核引导参数 `cgroup_enable=memory swapaccount=1`。

然后不要忘了更新 GRUB：

```
$ sudo update-grub  
$ sudo reboot
```

使用脚本自动安装

Docker 官方为了简化安装流程，提供了一套安装脚本，Ubuntu 和 Debian 系统可以使用这套脚本安装：

```
curl -sSL https://get.docker.com/ | sh
```

执行这个命令后，脚本就会自动的将一切准备工作做好，并且把 Docker 安装在系统中。

不过，由于伟大的墙的原因，在国内使用这个脚本可能会出现某些下载出现错误的情况。国内的一些云服务商提供了这个脚本的修改版本，使其使用国内的 Docker 软件源镜像安装，这样就避免了墙的干扰。

阿里云的安装脚本

```
curl -sSL http://acs-public-mirror.oss-cn-hangzhou.aliyuncs.com/docker-engine/internet | sh -
```

DaoCloud 的安装脚本

```
curl -sSL https://get.daocloud.io/docker | sh
```

手动安装

安装所需的软件包

## 可选内核模块

从 Ubuntu 14.04 开始，一部分内核模块移到了可选内核模块包([linux-image-extra-\\*](#))，以减少内核软件包的体积。正常安装的系统应该会包含可选内核模块包，而一些裁剪后的系统可能会将其精简掉。[AUFS](#) 内核驱动属于可选内核模块的一部分，作为推荐的 [Docker](#) 存储层驱动，一般建议安装可选内核模块包以使用 [AUFS](#)。

如果系统没有安装可选内核模块的话，可以执行下面的命令来安装可选内核模块包：

```
$ sudo apt-get install linux-image-extra-$(uname -r) linux-image-extra-virtual
```

## 12.04 LTS 图形界面

在 Ubuntu 12.04 桌面环境下，需要一些额外的软件包，可以用下面的命令安装。

```
$ sudo apt-get install xserver-xorg-lts-trusty libgl1-mesa-glx-lts-trusty
```

## 添加 APT 镜像源

虽然 Ubuntu 系统软件源中有 Docker，名为 [docker.io](#)，但是不应该使用系统源中的这个版本，它的版本太旧。我们需要使用 Docker 官方提供的软件源，因此，我们需要添加 APT 软件源。

由于官方源使用 HTTPS 以确保软件下载过程中不被篡改。因此，我们首先需要添加使用 HTTPS 传输的软件包以及 CA 证书。

*国内的一些软件源镜像（比如[阿里云](#)）不是太在意系统安全上的细节，可能依旧使用不安全的 HTTP，对于这些源可以不执行这一步。*

```
$ sudo apt-get update
$ sudo apt-get install apt-transport-https ca-certificates
```

为了确认所下载软件包的合法性，需要添加 Docker 官方软件源的 GPG 密钥。

```
$ sudo apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-keys 58118E89F3A912897C070ADB76221572C52609D
```

然后，我们需要向 [source.list](#) 中添加 Docker 软件源，下表列出了不同的 Ubuntu 和 Debian 版本对应的 APT 源。

操作系统版本	REPO
Precise 12.04 (LTS)	<a href="https://apt.dockerproject.org/repo/ubuntu-precise main">deb https://apt.dockerproject.org/repo ubuntu-precise main</a>

Trusty 14.04 (LTS)	deb https://apt.dockerproject.org/repo ubuntu-trusty main
Xenial 16.04 (LTS)	deb https://apt.dockerproject.org/repo ubuntu-xenial main
Debian 7 Wheezy	deb https://apt.dockerproject.org/repo debian-wheezy main
Debian 8 Jessie	deb https://apt.dockerproject.org/repo debian-jessie main
Debian Stretch/Sid	deb https://apt.dockerproject.org/repo debian-stretch main

用下面的命令将 APT 源添加到 `source.list`（将其中的 `<REPO>` 替换为上表的值）：

```
$ echo "<REPO>" | sudo tee /etc/apt/sources.list.d/docker.list
```

添加成功后，更新 apt 软件包缓存。

```
$ sudo apt-get update
```

## 安装 Docker

在一切准备就绪后，就可以安装最新版本的 Docker 了，软件包名称为 `docker-engine`。

```
$ sudo apt-get install docker-engine
```

如果系统中存在旧版本的 Docker（`lxc-docker`, `docker.io`），会提示是否先删除，选择是即可。

## 启动 Docker 引擎

### Ubuntu 12.04/14.04、Debian 7 Wheezy

```
$ sudo service docker start
```

### Ubuntu 16.04、Debian 8 Jessie/Stretch

```
$ sudo systemctl enable docker
$ sudo systemctl start docker
```

## 建立 docker 用户组



默认情况下，**docker** 命令会使用 **Unix socket** 与 Docker 引擎通讯。而只有 **root** 用户和 **docker** 组的用户才可以访问 Docker 引擎的 **Unix socket**。出于安全考虑，一般 Linux 系统上不会直接使用 **root** 用户。因此，更好地做法是将需要使用 **docker** 的用户加入 **docker** 用户组。

建立 **docker** 组：

```
$ sudo groupadd docker
```

将当前用户加入 **docker** 组：

```
$ sudo usermod -aG docker $USER
```

参考文档

- [Docker 官方 Ubuntu 安装文档](#)
- [Docker 官方 Debian 安装文档](#)

## Docker的离线安装

受限网络无法在线安装，可采用官方文档中的如下方式进行离线安装：

### Install from a package

If you cannot use Docker's repository to install Docker, you can download the .rpm file for your release and install it manually. You will need to download a new file each time you want to upgrade Docker.

1. Go to [https://download.docker.com/linux/centos/7/x86\\_64/stable/Packages/](https://download.docker.com/linux/centos/7/x86_64/stable/Packages/) and download the .rpm file for the Docker version you want to install.
2. Install Docker CE, changing the path below to the path where you downloaded the Docker package.

```
$ sudo yum install /path/to/package.rpm
```

## Docker官方镜像加速

修改镜像源

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。该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，从而能够更敏捷地开发和交付 Docker 化应用。

Docker 中国官方镜像加速可通过 [registry.docker-cn.com](https://registry.docker-cn.com) 访问。该镜像库只包含流行的公有镜像。私有镜像仍需要从美国镜像库中拉取。

修改 `/etc/docker/daemon.json` 文件并添加上 `registry-mirrors` 键值，保存后重启docker以生效

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

## 内网代理设置

First, create a systemd drop-in directory for the docker service:

```
mkdir /etc/systemd/system/docker.service.d
```

Now create a file called `/etc/systemd/system/docker.service.d/http-proxy.conf` that adds the `HTTP_PROXY` environment variable:

```
[Service]  
Environment="HTTP_PROXY=http://proxy.example.com:80/"
```

If you have internal Docker registries that you need to contact without proxying you can specify them via the `NO_PROXY` environment variable:

```
Environment="HTTP_PROXY=http://proxy.example.com:80/"  
Environment="NO_PROXY=localhost,127.0.0.0/8,docker-  
registry.somecorporation.com"
```

Flush changes:

```
$ sudo systemctl daemon-reload
```

Verify that the configuration has been loaded:

```
$ sudo systemctl show docker --property Environment  
Environment=HTTP_PROXY=http://proxy.example.com:80/
```

Restart Docker:

```
$ sudo systemctl restart docker
```