# 先决条件

## Linux 64位

## Linux Kernel 版本 >= 3.10

检查版本，输入：

*$ uname –r*

# 更新你的apt源

## 以root或sudo方式登陆系统

## 打开终端

## 更新包信息，确保apt以https方式可以工作，且CA证书已安装

*$ sudo apt-get update*

*$ sudo apt-get install apt-transport-https ca-certificates*

## 添加新的GPG key。使用以下命令从服务器上下载并添加到adv keychain

*$ sudo apt-key adv \ --keyserver hkp://ha.pool.sks-keyservers.net:80 \ --recv-keys 58118E89F3A912897C070ADBF76221572C52609D*

## 根据版本，找到属于自己的版本库

| **Ubuntu version** | **Repository** |
| --- | --- |
| Precise 12.04 (LTS) | deb https://apt.dockerproject.org/repo ubuntu-precise main |
| Trusty 14.04 (LTS) | deb https://apt.dockerproject.org/repo ubuntu-trusty main |
| Wily 15.10 | deb https://apt.dockerproject.org/repo ubuntu-wily main |
| Xenial 16.04 (LTS) | deb https://apt.dockerproject.org/repo ubuntu-xenial main |

## 使用以下命令，新增apt包地址

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

其中，以Ubuntu 14为例，<REPO>替换成deb https://apt.dockerproject.org/repo ubuntu-trusty main，使用以下命令

*$ echo "deb https://apt.dockerproject.org/repo ubuntu-trusty main" | sudo tee /etc/apt/sources.list.d/docker.list*

## 更新APT目录

*$ sudo apt-get update*

## 确认APT获取了正确的仓库

运行以下命令，会返回所有可安装的Docker版本信息的条目。每条都应该有这个URL” <https://apt.dockerproject.org/repo/>”。当前已安装版本会被”\*\*\*”标记。以下输出省略了开头。

$ apt-cache policy docker-engine

docker-engine:

Installed: 1.12.2-0~trusty

Candidate: 1.12.2-0~trusty

Version table:

\*\*\* 1.12.2-0~trusty 0

500 https://apt.dockerproject.org/repo/ ubuntu-trusty/main amd64 Packages

100 /var/lib/dpkg/status

1.12.1-0~trusty 0

500 https://apt.dockerproject.org/repo/ ubuntu-trusty/main amd64 Packages

1.12.0-0~trusty 0

500 https://apt.dockerproject.org/repo/ ubuntu-trusty/main amd64 Packages

# 安装最新版本

确保你满足了所有先决条件，且更新了你的apt源

## 安装Docker

$ sudo apt-get install docker-engine

## 开启守护进程

$ sudo service docker start

## 测试安装结果，运行hello-world镜像

$ sudo docker run hello-world

# 我的小结

以上翻译官网教程，我的过程是，先以普通用户模式下载并安装了docker（参照官网另一个教程），然后使用以下命令：

$ sudo usermod –aG docker yolo

完成的以普通用户(yolo)管理docker，证明安装版docker会自动创建docker组，只需要加进去就可以了。

完整的以普通用户管理docker的教程如下：

## Manage Docker as a non-root user

The docker daemon binds to a Unix socket instead of a TCP port. By default that Unix socket is owned by the user root and other users can only access it using sudo. The docker daemon always runs as the root user.

If you don’t want to use sudo when you use the docker command, create a Unix group called docker and add users to it. When the docker daemon starts, it makes the ownership of the Unix socket read/writable by the docker group.

**Warning**: The docker group is equivalent to the root user. For details on how this impacts security in your system, see [Docker Daemon Attack Surface](https://docs.docker.com/engine/security/security/#docker-daemon-attack-surface) for details.

To create the docker group and add your user:

1. Log into Ubuntu as a user with sudo privileges.
2. Create the docker group.
3. $ sudo groupadd docker
4. Add your user to the docker group.
5. $ sudo usermod -aG docker $USER
6. Log out and log back in so that your group membership is re-evaluated.
7. Verify that you can docker commands without sudo.
8. $ docker **run** hello-world

If this fails, you will see an error:

Cannot connect to the Docker daemon. Is 'docker daemon' running on this host?

Check whether the DOCKER\_HOST environment variable is set for your shell.

$ **env** | grep DOCKER\_HOST

If it is set, the above command will return a result. If so, unset it.

$ **unset** DOCKER\_HOST

You may need to edit your environment in files such as ~/.bashrc or~/.profile to prevent the DOCKER\_HOST variable from being set erroneously.