## 文档控制

|  |  |  |  |
| --- | --- | --- | --- |
| **文档名称** | Docker安装与私有仓库搭建 | | |
| **说明** | docker的在环境下的安装与私有仓库的配置 | | |
| **修订历史** | | | |
| **版本** | **作者** | **日期** | **修改说明** |
| V1.0 | 任新林 | 2018.10.10 | 创建文档 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 一、Ubuntu 安装 Docker CE

1. 系统要求

* Docker CE 支持以下版本的 Ubuntu 操作系统：
* Artful 17.10 (Docker CE 17.11 Edge)
* Zesty 17.04
* Xenial 16.04 (LTS)
* Trusty 14.04 (LTS)

Docker CE 可以安装在 64 位的 x86 平台或 ARM 平台上。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，这样版本会更稳定，因此在生产环境中推荐使用 LTS 版本,当前最新的 LTS 版本为 Ubuntu 16.04。

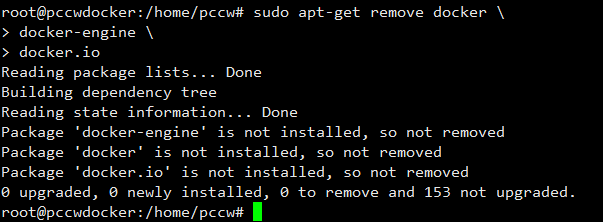
1. 卸载旧版本

旧版本的 Docker 称为 docker 或者 docker-engine ，使用以下命令卸载旧版本：

$ sudo apt-get remove docker \

docker-engine \

docker.io



如上图表示此服务器未安装docker.

1. Ubuntu 14.04 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：

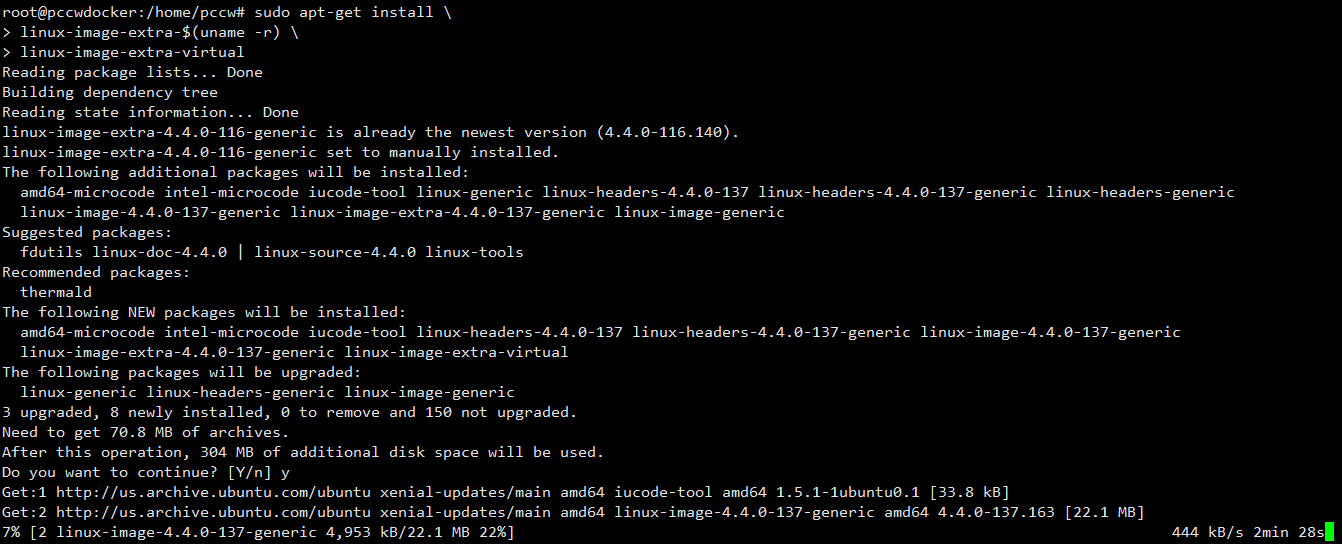
$ sudo apt-get update



$ sudo apt-get install \

linux-image-extra-$(uname -r) \

linux-image-extra-virtual

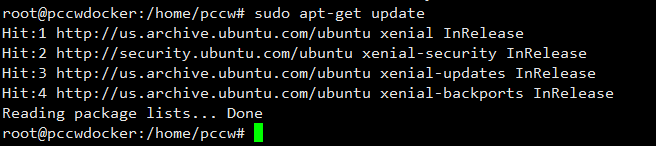


等待安装完成。

1. 使用APT镜像源安装

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。

$ sudo apt-get update



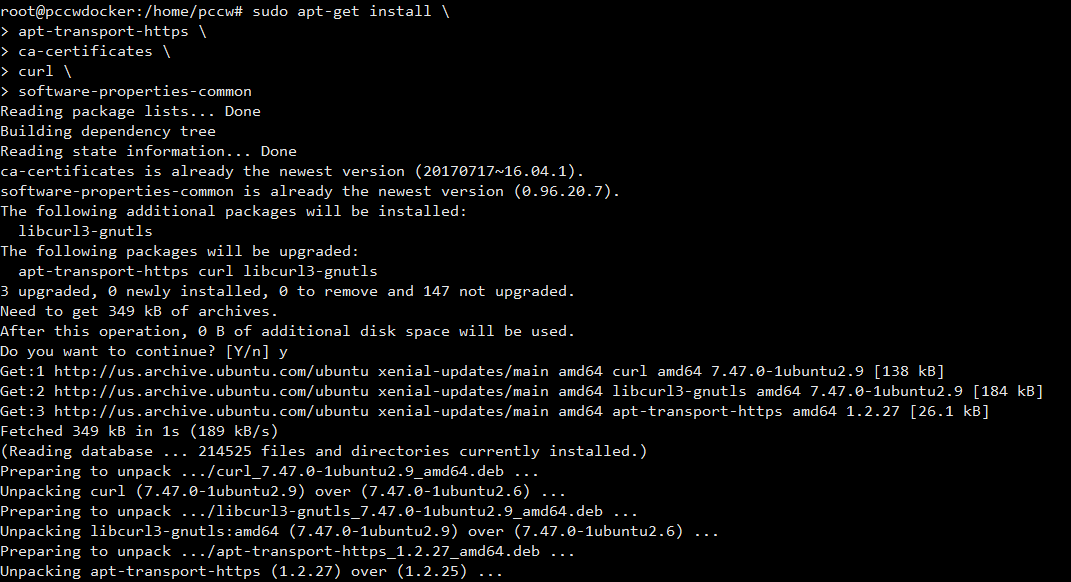
$ sudo apt-get install \

apt-transport-https \

ca-certificates \

curl \

software-properties-common

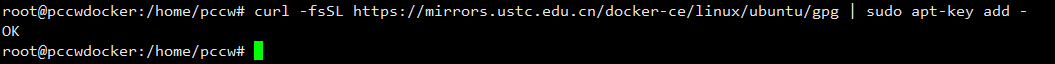


等待安装完成。

1. 鉴于国内网络问题，强烈建议使用国内源，官方源请在注释中查看。

为了确认所下载软件包的合法性，需要添加软件源的 GPG 密钥。

$ curl -fsSL https://mirrors.ustc.edu.cn/docker-ce/linux/ubuntu/gpg | sudo apt-key add –



# 官方源

# $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –

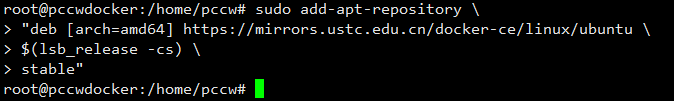
然后，我们需要向 source.list 中添加 Docker 软件源；

$ sudo add-apt-repository \

"deb [arch=amd64] https://mirrors.ustc.edu.cn/docker-ce/linux/ubuntu \

$(lsb\_release -cs) \

stable"



# 官方源

# $ sudo add-apt-repository \

# "deb [arch=amd64] https://download.docker.com/linux/ubuntu \

# $(lsb\_release -cs) \

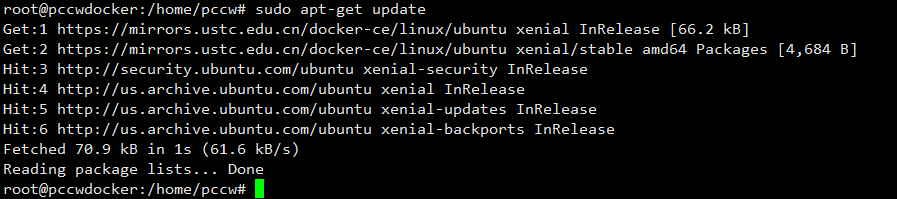
# stable"

以上命令会添加稳定版本的 Docker CE APT 镜像源，如果需要最新或者测试版本的Docker CE 请将 stable 改为 edge 或者 test。从 Docker 17.06 开始，edge test 版本的APT 镜像源也会包含稳定版本的 Docker。

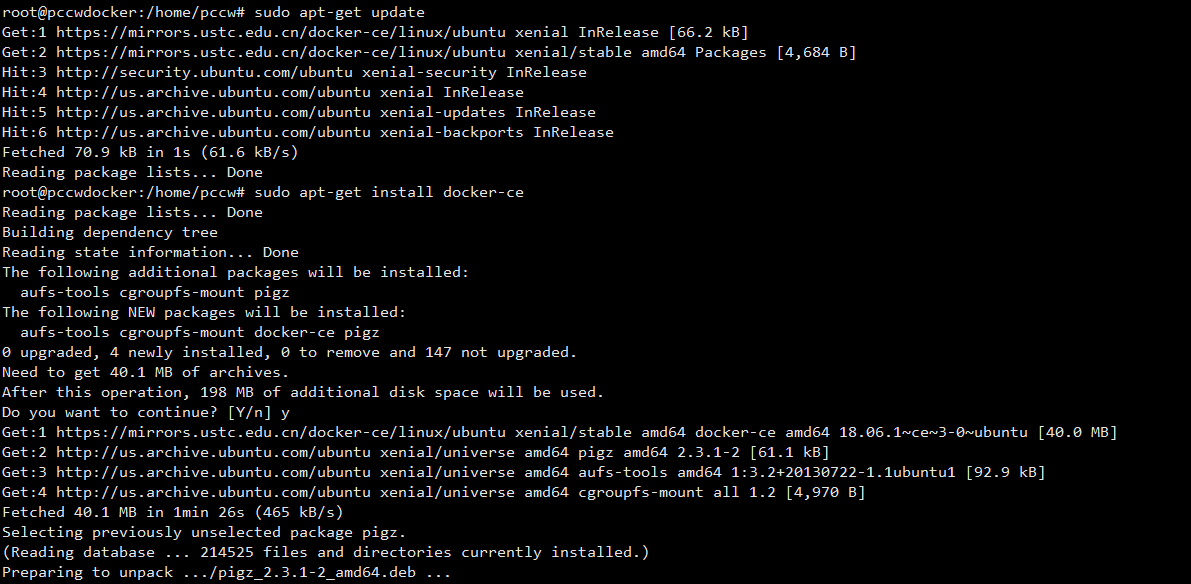
1. 安装 Docker CE

更新 apt 软件包缓存，并安装 docker-ce ：

$ sudo apt-get update



$ sudo apt-get install docker-ce



等待安装完成。

*##使用脚本自动安装*

*在测试或开发环境中 Docker 官方为了简化安装流程，提供了一套便捷的安装脚本，Ubuntu系统上可以使用这套脚本安装：*

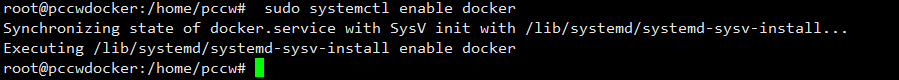
*$ curl -fsSL get.docker.com -o get-docker.sh*

*$ sudo sh get-docker.sh --mirror Aliyun*

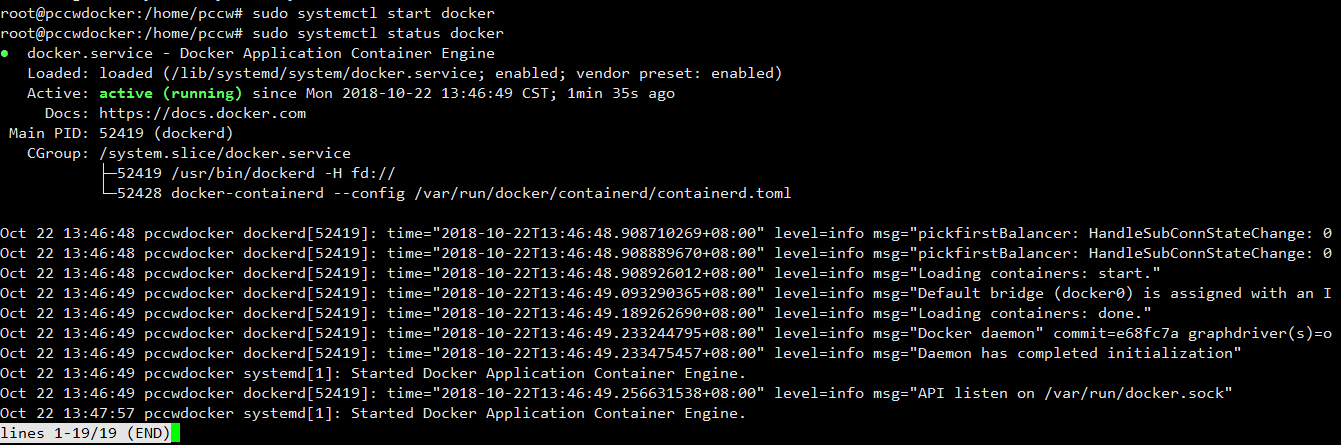
*执行这个命令后，脚本就会自动的将一切准备工作做好，并且把 Docker CE 的 Edge 版本安装在系统中。*

1. 启动 Docker CE

$ sudo systemctl enable docker



$ sudo systemctl start docker



1. 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



表明该组已经存在，不需要创建；

将当前用户加入 docker 组：

$ sudo usermod -aG docker $USER，比如有一个pccw用户；



退出当前终端并重新登录，进行如下测试。

1. 测试 Docker 是否安装正确

$ docker run hello-world

Unable to find image 'hello-world:latest' locally

latest: Pulling from library/hello-world

ca4f61b1923c: Pull complete

Digest: sha256:be0cd392e45be79ffeffa6b05338b98ebb16c87b255f48e297ec7f98e123905c

Status: Downloaded newer image for hello-world:latest

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:

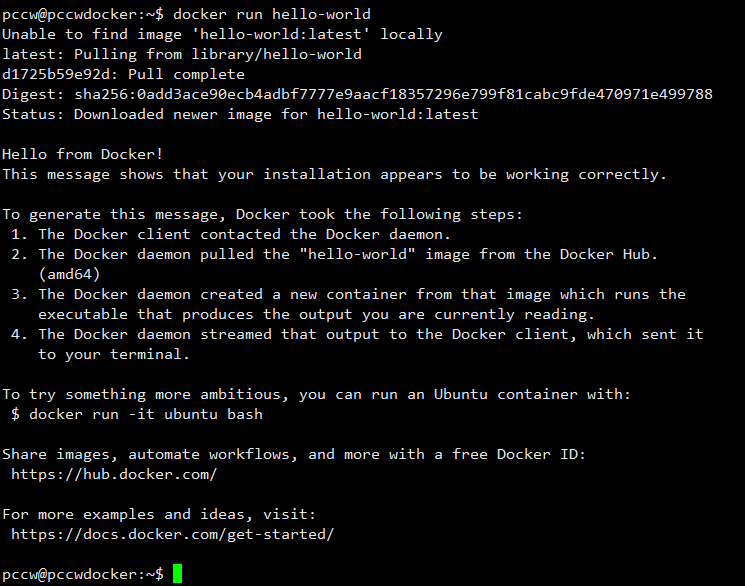
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:

https://cloud.docker.com/

For more examples and ideas, visit:

<https://docs.docker.com/engine/userguide/>



1. 若能正常输出以上信息，则说明安装成功。

# 二、搭建docker registry 私有仓库

1. 搭建 docker 私有仓库

docker 官方提供了 registry 的镜像，可以使用它来建私有仓库。

1. 搭建过程

目标：

* 在ubuntu 14 的服务器上，安装最新版本的 docker-ce，在 docker hub 获取 registry 镜像，并启动仓库容器。
* 在 docker 客户端机器，可以从服务器的仓库获取镜像文件。

1. 首先确保服务端已安装最新版docker;
2. 从 docker hub 获取官方 registry 镜像，并启动 registry 容器
3. 获取 registry 镜像：

docker pull registry # 默认就是最新的 latest 版本

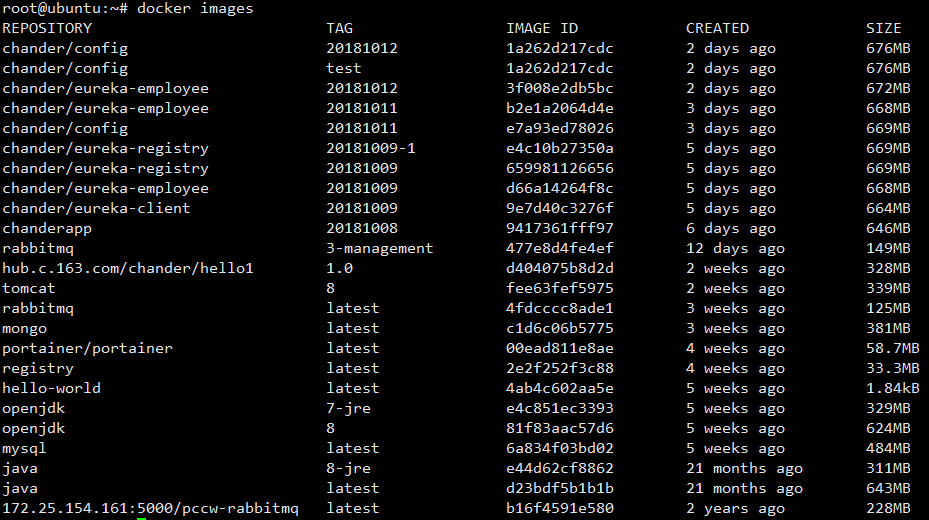
2. 用registry 镜像启动一个仓库容器：

docker run –d --name=pccw-docker-registry --restart=always -p 5000:5000 -v /opt/data/registry:/tmp/registry registry

说明：启动一个名字为 pccw-docker-registry 的容器，端口映射到宿主机的5000，挂载宿主机目录 /opt/data/registry 到容器的 /tmp/registry ，用于存储 push 进去的镜像文件。

5、在宿主机本地测试仓库

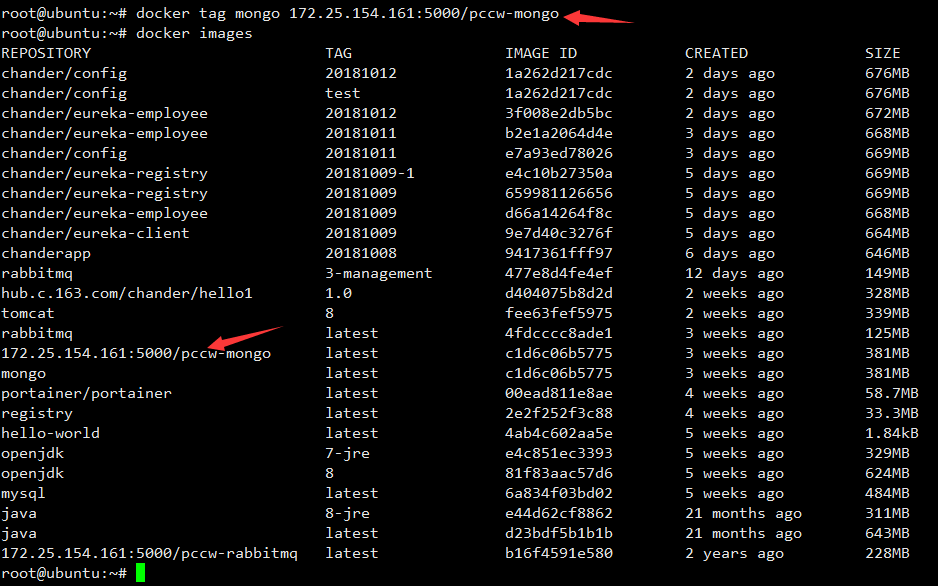
* 首先查看宿主机本地的image，挑选一个image作为测试对象；



我们就以mongo镜像为测试对象；

* 将mongo镜像重新打 tag ：

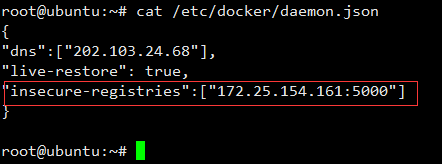
docker tag mongo 172.25.154.161:5000/pccw-mongo

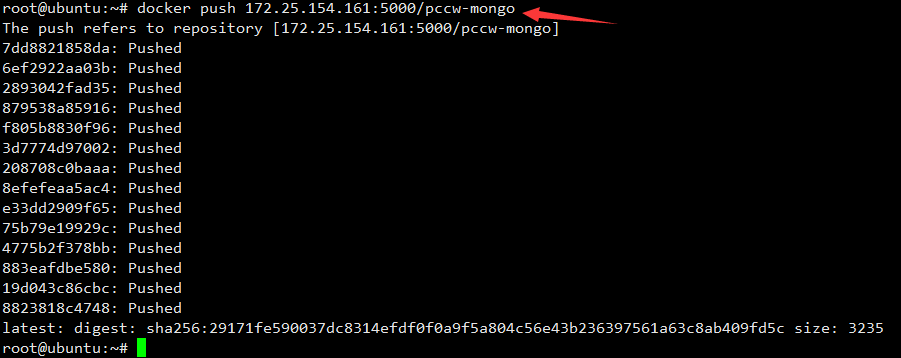


* 将新打标签的镜像push 到本地仓库，docker 命令会从“. 和 : ”的格式识别是本地路径的仓库：

docker push 172.25.154.161:5000/pccw-mongo

*注意：执行上面这条命令的时候，首先需要修改/etc/docker/daemon.json文件，添加如下内容，否则在进行push的时候会报http: server gave HTTP response to HTTPS client的错误。*





* 查看仓库中是否已经上传镜像成功：



6、从其他客户端获取镜像

* 客户端下载私有仓库的镜像，首先也需要修改/etc/docker/daemon.json文件，添加如下配置：

"insecure-registries":["172.25.154.161:5000"]

添加完成后，重新启动docker服务即可。

* 然后在客户端docker用户下执行以下命令，即可获取远程私有仓库的镜像；

docker pull 172.25.154.161:5000/pccw-mongo