**Docker安装部署与CaaS实践**

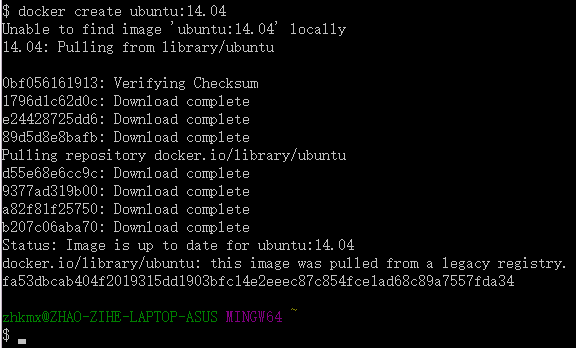
# 1、Linux(Ubuntu)下Docker安装

# 2、Docker容器

## 2.1 创建容器

**$ docker create ubuntu:14.04** 创建不运行

**$ docker run ubuntu:14.04** 创建并运行



两个命令执行之后：



## 2.2创造交互式容器

## 2.3 查看容器

**$ docker ps** 列出运行的容器

**$ docker ps –a**  列出包括不运行的容器

# 3、Docker适应非root用户

步骤：

1. 建立Docker用户组

**$ sudo groupadd docker**

可能提示：groupadd：“docker”组已存在

1. 加入Docker用户组

**$ sudo gpasswd -a zhkmxx930 docker**

1. 重启Docker服务

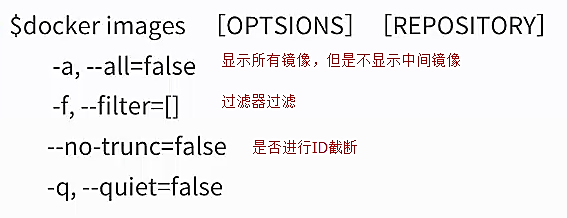
**$ sudo service docker restart**

1. 重新启动系统使之生效

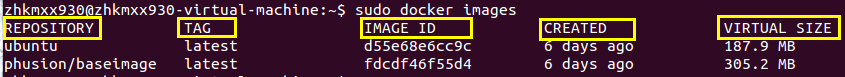
# 4、Docker镜像与仓库

## 4.1列出镜像

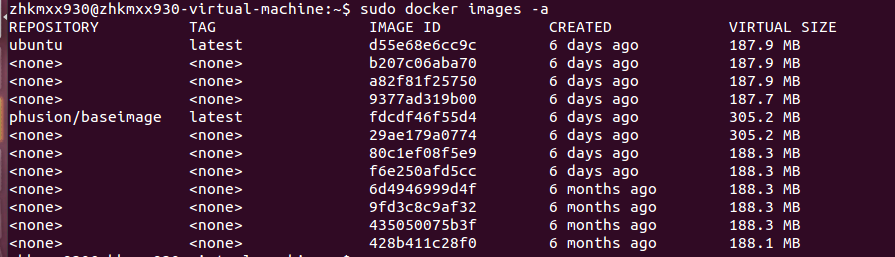
**$ sudo docker images**

-q 只显示唯一ID

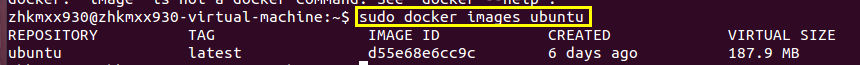
无任何条件：



显示所有镜像，未命名为中间层镜像：



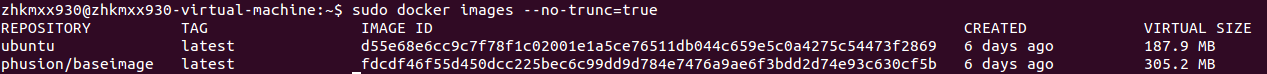
加入Filter：



截断ID：



未截断ID：



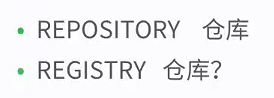
只显示唯一ID：



只显示未截断的唯一ID：



## 4.2关于仓库



REGISTRY包含许多REPOSITORY

REPOSITORY包含许多独立的镜像

## 4.3关于Tag标签



Docker中会出现同样的ID但名称不同的情况，也就是说可以将同一个镜像命名不同的名字。

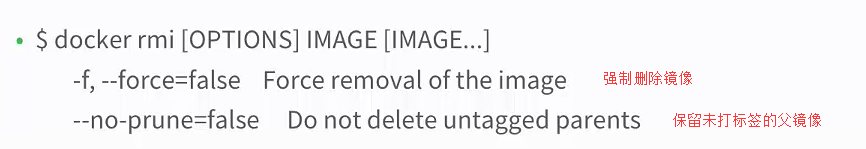
## 4.4查看镜像/容器

**$ sudo docker inspect [ID/镜像名:Tag]**



返回Json

## 4.5删除镜像



如果多个名称对应一个镜像，通过删除名称的方式需要删除所有ID对应的名称，才可以将其删除。而一对一则直接删除。

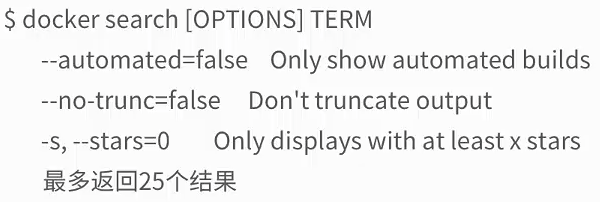
直接删除ID，即全删除

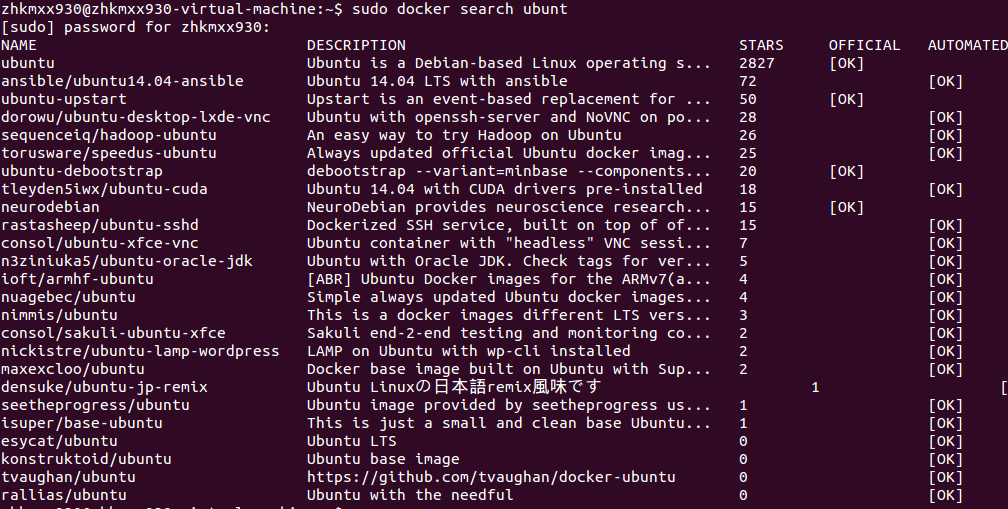
## 4.6获取镜像

方法一：Docker Hub官方网站

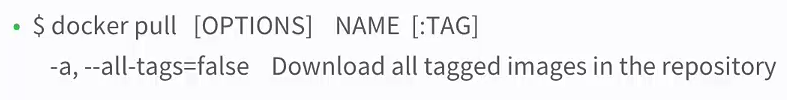
<http://registry.hub.docker.com>

方法二：



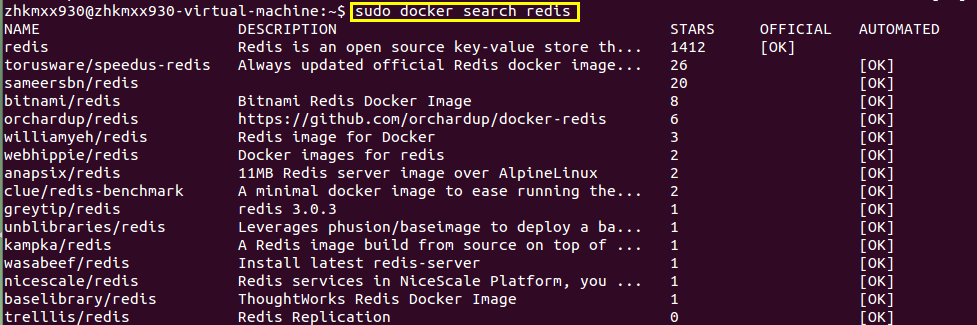


## 4.7Pull镜像(下载)

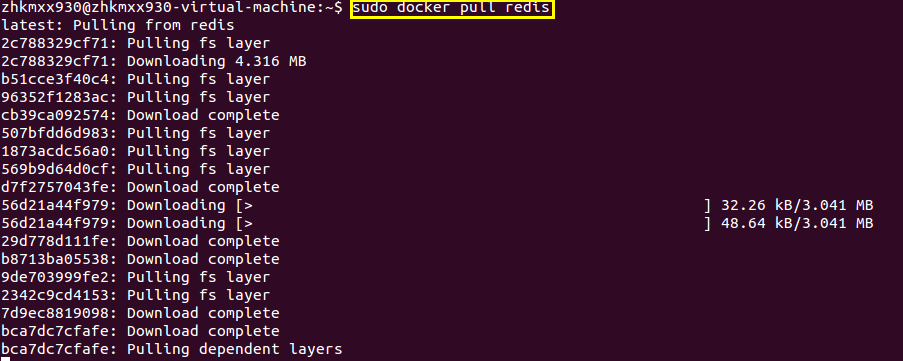


E.g.

首先search镜像，这里玩一下redis

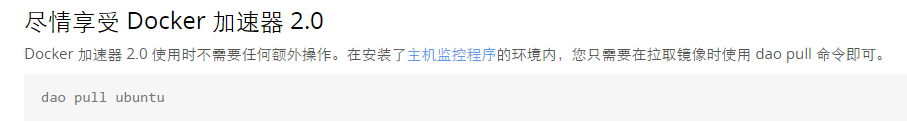


然后pull镜像



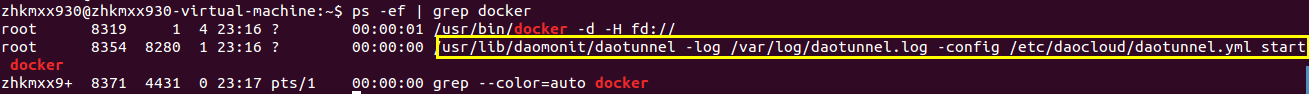
### 4.7.1如何使用Daocloud加速pull

到这里注册 [https://www.daocloud.io/](%20https:/www.daocloud.io/) 然后选择加速器，按照指示进行OK

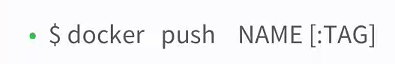


过程中可能会有下载失败的情况发生，这里只有多试几次。

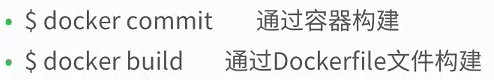
完成后，**$ sudo service docker restart**重启docker,然后**$ ps -ef | grep docker** 查看进程，得到如图所示结果。



## 4.8Push镜像

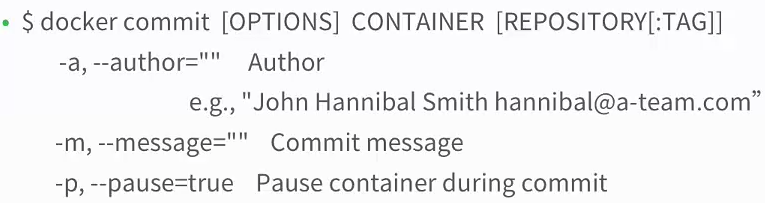


## 4.9构建镜像



### 4.9.1Commit构建

///////////////////////////////////////////////////////////////////////////



1. 构建交互式容器

**$ sudo docker run -i -t -p 80 --name commit\_test ubuntu /bin/bash**

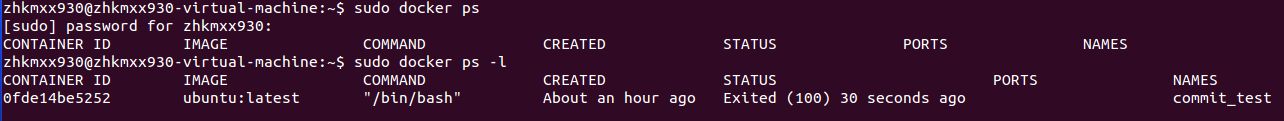


1. 退出容器



1. 查看所有容器

**$ sudo docker ps**



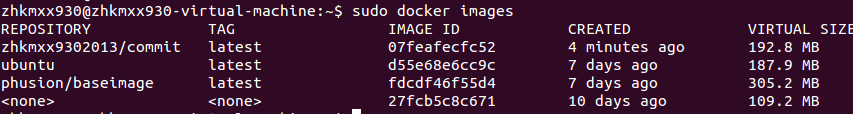
1. Commit镜像

**$ sudo docker commit -a 'zhkmxx930' -m 'nginx' commit\_test zhkmxx9302013/commit**

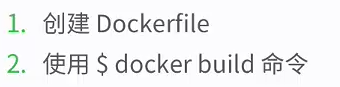


1. 查看镜像

**$ sudo docker images**



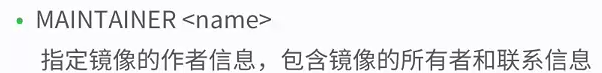
### DockerFile构建



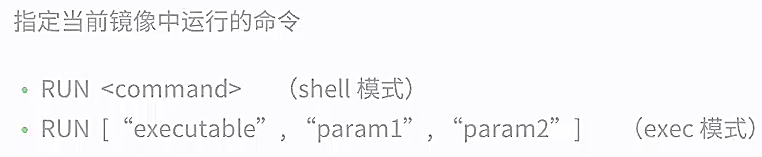
#### FROM指令



#### 4.9.2.2 MAINTAINER指令



#### RUN指令



#### 4.9.2.4 EXPOSE指令

