**Docker使用基本教程**

# [Docker中文文档](http://www.widuu.com/docker/index.html)

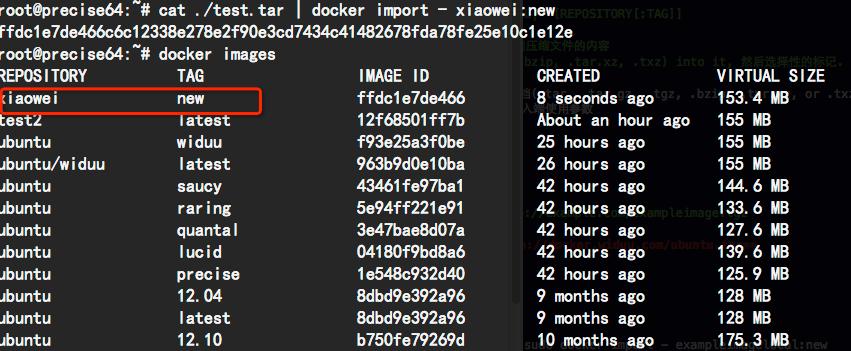
## 1.启动docker--有两种方式

1. 如果你是二进制安装的，你可以使用sudo <path to>/docker -d &来启动
2. 如果你是apt-get安装的你可以用service docker start开启

## 2.搜索安装镜像

1. 官方安装方式docker pull imagename从docker的索引中心下载，imagename是镜像名称，例如docker pull ubuntu就是下载base ubuntu并且tag是latest。
2. 因为官方速度比较慢你可以从我提供的国内镜像下载，我用的七牛云存储，
   * ubuntu下载地址：<http://docker.widuu.com/ubuntu.tar>
   * centos下载地址：<http://docker.widuu.com/centos.tar>
   * debian下载地址：<http://docker.widuu.com/debian.tar.gz>
   * opensuse下载地址: [http://docker.widuu.com/opensuse131.tar](http://docker.widuu.com/opensuse_13_1.tar)

#第一种方法url下载安装ubuntu docker import http://docker.widuu.com/ubuntu.tar #第二种方法，下载下来然后根据自己配置安装 wget http://docker.widuu.com/ubuntu.tar cat test.tar | sudo docker import - xiaowei:new



# 学习基本命令

### 启动docker

如果你使用快速安装并且添加到PATH中，docker就被安装到快捷启动了，当Ubuntu系统启动时就启动了！你只需要运行命令sudo run docker help获得输出.

如果你获得结果是"docker: command not found"或者像" /var/lib/docker/repositories: permission denied",你需要手动指定路径执行它.

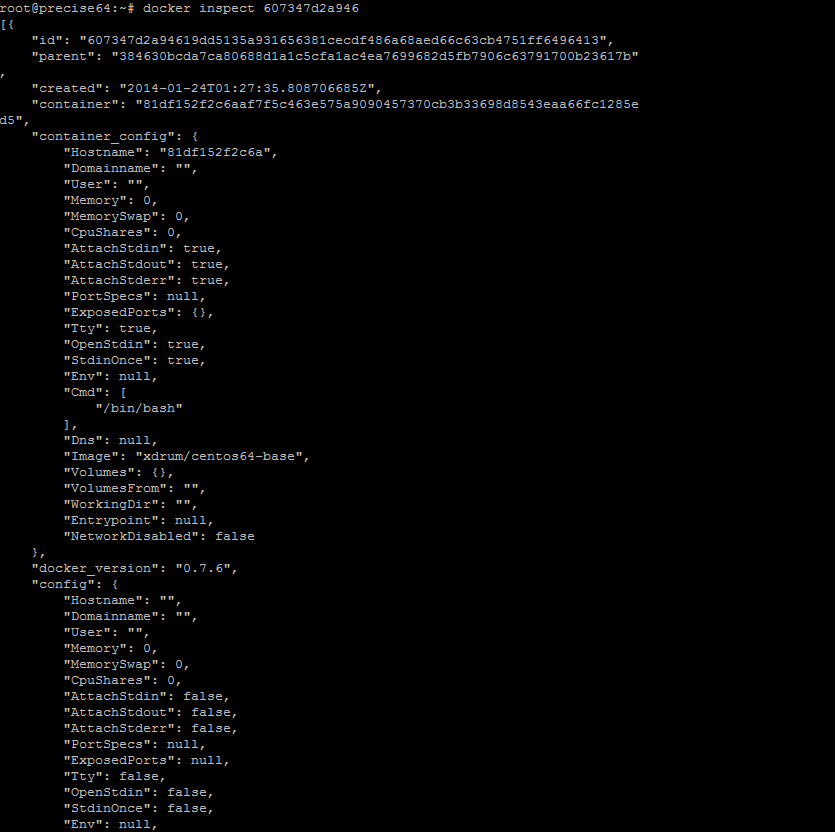
# Run docker in daemon mode sudo <path to>/docker -d &

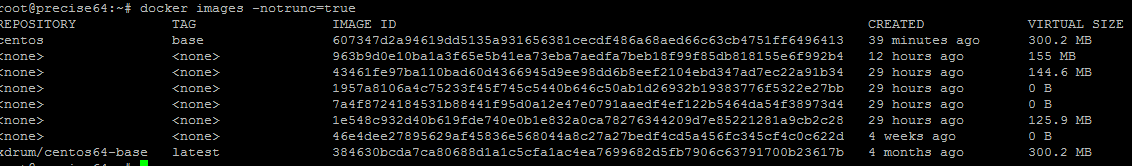
### 下载一个预建立的镜像

# 下载一个ubuntu镜像 sudo docker pull ubuntu

这个将从索引仓库中通过名字找到ubuntu镜像，并且从索引仓库顶级中心来下载到本地镜像存储

注：当镜像下载成功后，你可以看到12位的hash值像539c0211cd76，这是下载完整的镜像的精简ID，这些短的镜像ID是完整镜像ID的前12个字符--可以使用docker inspect 或者 docker images -notrunc=true来获得完整的镜像ID





#### 运行交互性的shell（就像运行一个单独的linux）

# 使用ubuntu运行一个交互性的shell, # 分配一个伪终端，附带stdin和stdout(输入/输出流) # 如果你想退出分离出来的伪终端, # 可以使用CTRL -p+CTRL -q --就像先按CTRL -p 然后CTRL -q sudo docker run -i -t ubuntu /bin/bash

#### 管理员命令和docker组

docker进程经常使用root用户运行，从docker的0.5.2版本开始，docker的进程绑定Unix Socket来代替一个TCP端口，在默认情况下Unix Socket属于root用户，所以在默认情况下，你需要赋予权限sudo

从0.5.3版本开始，如果你(或者你安装的docker)创建的时候用的docker或者添加用户的unix群组，当docker进程启动的时候，这个docker进程会把Unix socket的读写(read/writable)的所有权交给docker群组.docker进程一般必须root用户运行，但是你运行docker客户端的用户是docker组的，这个时候你就不需要加sudo就可以运行全部的客户端命令

警告：docker群组必须是和root等价的

#### Example:

# 如果不存在docker群组，添加一个用户群组 sudo groupadd docker # Add the connected user "${USER}" to the docker group. # Change the user name to match your preferred user. # You may have to logout and log back in again for # this to take effect. sudo gpasswd -a ${USER} docker # Restart the docker daemon. sudo service docker restart

#### 让Docker使用其它的host/port或者Unix socket

警告

改变默认的docker进程绑定的TCP端口或者Unix docker的用户组，将会通过允许非root用户获得root权限来修改主机，会增加你的安全风险！

使用 -H 他可以运行你改变docker进程监听指定的IP和端口。默认情况下，他会监听 unix:///var/run/docker.sock只允许本地的root用户连接，你可以设置他 0.0.0.0:4243 或者指定主机IP给任何用户，但是这不是我们所推荐的，因为那么低权限将会获得root正在运行的主机进程的访问权限！

同样，docker客户端可以使用-H 来指定连接的端口

-H 授权主机和端口的格式是这样的:tcp://[host][:port] or unix://path

举个例子

* tcp://host:4243 -> 使用tcp连接 host:4243
* unix://path/to/socket -> 使用socket位于 path/to/socket

-H 当空的时候 将会使用默认值就像没有 -H一样

-H 也可以用简短的方式授权TCP绑定： host[:port] or :port

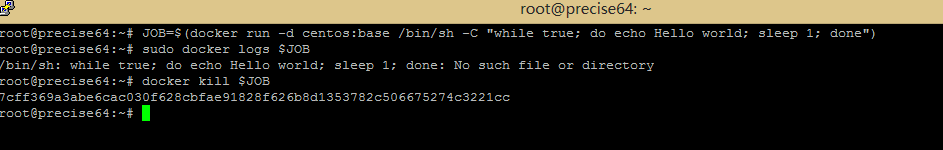
# Run docker in daemon mode sudo <path to>/docker -H 0.0.0.0:5555 -d & # Download an ubuntu image sudo docker -H :5555 pull ubuntu

你可以是用多个 -H 例如你想监听你所有的TCP和Unix socket

# Run docker in daemon mode sudo <path to>/docker -H tcp://127.0.0.1:4243 -H unix:///var/run/docker.sock -d & # Download an ubuntu image, use default Unix socket sudo docker pull ubuntu # OR use the TCP port sudo docker -H tcp://127.0.0.1:4243 pull ubuntu

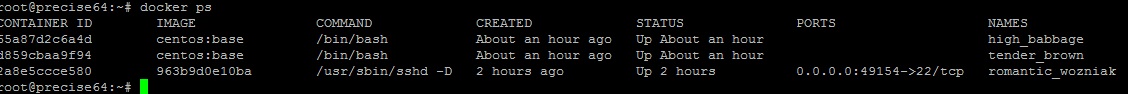
#### 开启一个长时间运行的工作进程

# 开启一个非常有用的长时间工作进程 JOB=$(sudo docker run -d ubuntu /bin/sh -c "while true; do echo Hello world; sleep 1; done") # 到目前为止的收集的输出 sudo docker logs $JOB # 杀死这个进程 sudo docker kill $JOB



#### 监听所有运行着的容器

sudo docker ps



#### 为服务绑定一个TCP端口

# 为容器绑定4444端口，并告诉网络监听4444 JOB=$(sudo docker run -d -p 4444 ubuntu:12.10 /bin/nc -l 4444) # 查看我的容器用的公共端口 PORT=$(sudo docker port $JOB 4444 | awk -F: '{ print $2 }') # 连接公共端口 echo hello world | nc 127.0.0.1 $PORT # 确认网络连接是否工作 echo "Daemon received: $(sudo docker logs $JOB)"

#### 提交（保存）容器的状态

保存你的容器状态作为一个容器镜像，所以容易状态可以重用

当你保存你的容器的时候，仅仅只是保存现在容器和创建容器时候的不同状态（as a diff）,用'docker images'命令查看哪些镜像是你正在所使用

#Commit your container to a new named image sudo docker commit <container\_id> <some\_name> # List your containers sudo docker images

现在你有一个新的镜像状态（其实就是一个新的镜像），从中你可以创建新的实例