# 什么是Docker?

简介：Docker是一个开源的引擎，可以轻松的为任何应用创建一个轻量级的、可移植的、自给自足的容器。开发者在笔记本上编译测试通过的容器可以批量地在生产环境中部署，包括VMs（虚拟机）、bare metal、OpenStack 集群和其他的基础应用平台。

Docker是一个开源的引擎，可以轻松的为任何应用创建一个轻量级的、可移植的、自给自足的容器。开发者在笔记本上编译测试通过的容器可以批量地在生产环境中部署，包括VMs（虚拟机）、bare metal、OpenStack 集群和其他的基础应用平台。

Docker通常用于如下场景：

- web应用的自动化打包和发布；

- 自动化测试和持续集成、发布；

- 在服务型环境中部署和调整数据库或其他的后台应用；

- 从头编译或者扩展现有的OpenShift或Cloud Foundry平台来搭建自己的PaaS环境。

进入docker的脚本：

* docker\_in.sh

|  |
| --- |
| [root@localhost docker]# cat docker\_in.sh  #!/bin/bash  # Using nsenter to access docker  docker\_in(){  NAME\_ID=$1  PID=$(docker inspect --format "{{ .State.Pid }}" $NAME\_ID)  nsenter -t $PID -m -u -i -n -p  }  docker\_in $1 |

# Chapter2

目前，最大的公开仓库是Docker Hub，存放了数据量庞大的镜像供用户下载。国内的公开仓库包括Docker Pool等，可以提供稳定的国内访问。

# Chapter3

## 3.1 获取镜像

🡪 docker pull命令从网络上下载镜像。

docker pull NAME[:TAG]。如果如果不显示的指定TAG，则默认会选择lastest标签，即下载最新版本的镜像。

sudo docker pull ubuntu

sudo docker pull centos

运行一个镜像

|  |
| --- |
| hadoop@hadoop:~$ sudo docker run -t -i ubuntu /bin/bash  [sudo] hadoop 的密码：  root@9d84d5d421fa:/# uname  Linux  root@9d84d5d421fa:/# uname -a  Linux 9d84d5d421fa 4.4.0-31-generic #50-Ubuntu SMP Wed Jul 13 00:07:12 UTC 2016  x86\_64 x86\_64 x86\_64 GNU/Linuxroot@9d84d5d421fa:/# cat /etc/issue  Ubuntu 16.04.1 LTS \n \l  root@9d84d5d421fa:/# |

## 3.2查看镜像信息

docker image

docker inspect [IMAGE ID]🡪获取镜像的详细信息

3.3搜寻镜像

docker search XXX

## 3.4删除镜像

docker rmi

使用镜像ID删除镜像，docker rmi 命令后跟上镜像的ID(也可以是ID能区分的部分前缀串)时，会先尝试删除所有指向该镜像的标签，然后删除该镜像本身。

**注意：当该镜像创建的容器存在时，镜像文件是无法被删除的，例如**

**先利用ubuntu镜像创建一个简单的容器，输出一句话“hello! I am here!”:**

**$sudo docker run ubuntu echo ‘helo!I am here!’**

**使用docker ps –a命令查看本机存在的所有容器：**

**docker ps –a**

|  |
| --- |
| root@ubuntu:/home/hadoop# docker ps -a  CONTAINER ID IMAGE COMMAND CREATED STATUS  PORTS NAMES5e7b728abcce ubuntu "echo 'hello,I am her" 12 seconds ago Exited (0)  11 seconds ago tiny\_kilby476fae0087be haskell "/bin/bash" 6 minutes ago Up 6 minut  es boring\_williams99281f24373c centos "/bin/bash" 9 minutes ago Up 9 minut  es big\_rosalindf4e22826a355 centos "/bin/bash" 9 minutes ago Up 9 minut  es silly\_meninsky605fb8a865c9 centos "/bin/bash" 11 minutes ago Exited (12  7) 10 minutes ago lonely\_heisenberg46d29c349689 centos "/bin/bash" 11 minutes ago Exited (12  7) 11 minutes ago clever\_crayroot@ubuntu:/home/hadoop# docker rmi ubuntu |

**删除依赖该镜像的所有容器**

**docker docker rm**

## 3.5创建镜像

**root@ubuntu:/home/hadoop/code# docker commit -m "add a new file" -a "Docker Newbee" d73a695e64aa test**

**返回：**

**sha256:db20f86ef738df6402ee6c71523aa670b8d85b3ff285becbb1043e934fc2b556**

### 🡪docker image

#### 基于已有镜像的容器创建

**root@ubuntu:/home/hadoop/code# docker images**

**REPOSITORY TAG IMAGE ID CREATED SIZE**

**test latest db20f86ef738 5 seconds ago 183 MB**

#### 基于本地模板导入

OpenVZ提供的模板来创建（http://openvz.org/Download/template/precreated）

下载一个debian-8.0-x86\_64-minimal.tar.gz的模板压缩包

|  |
| --- |
| **root@ubuntu:/home/hadoop/docker# cat debian-8.0-x86\_64-minimal.tar.gz | docker import - debain:8**  **.0sha256:da287d42e29502949f5758c33cd2d5f1402e120f9fa19dac1d51478a5d380a50**  **root@ubuntu:/home/hadoop/docker# docker images**  **REPOSITORY TAG IMAGE ID CREATED SIZE**  **debain 8.0 da287d42e295 8 seconds ago 301.7 MB** |

## 3.6存入和载入镜像

使用docker save和docker load命令来存入和载入镜像。

docker save

$sudo docker save –o ubuntu\_14.04.tar ubuntu14.04

载入镜像

docker load

$sudo docker load –input ubuntu\_14.04.tar

或者

$sudo docker load < ubuntu\_14.04.tar

## 3.7上传镜像

**root@ubuntu:/home/hadoop/docker# docker tag test:latest user/test:latest**

**root@ubuntu:/home/hadoop/docker# docker push user/test:latest**

# Chapter4

## 4.1创建容器

**root@ubuntu:/home/hadoop/docker# docker create -it ubuntu:latest**

**056d1faa59f3794324db42dc21cdec790d9b3a9fbe32107578aec920b8392e26**

**root@ubuntu:/home/hadoop/docker# docker ps -a**

**CONTAINER ID IMAGE COMMAND CREATED STATUS**

**PORTS NAMES056d1faa59f3 ubuntu:latest "/bin/bash" 11 seconds ago Created**

**distracted\_mclean**

**root@ubuntu:/home/hadoop/docker# docker start 05**

**05**

**root@ubuntu:/home/hadoop/docker# docker ps -a**

**CONTAINER ID IMAGE COMMAND CREATED STATUS**

**PORTS NAMES056d1faa59f3 ubuntu:latest "/bin/bash" About a minute ago Up 8 seconds**

**distracted\_mclean**

**下面的命令则启动一个bash终端，允许用户进行交互**

**$sudo docker run –t –I ubuntu:14.04 /bin/bash**

**-t 让Docker分配一个伪终端（pseudo-tty）并绑定到容器的标准输入上**

**-i 则让容器的标准输入保持打开**

**使用Ctrl+d或者exit退出容器**

## 4.2 终止容器

**root@ubuntu:/home/hadoop/docker# docker stop 05 47 99 f4**

**05**

**47**

**99**

**f4**

**//可以使用docker ps –a –q 命令看到处于终止状态的容器的ID信息。**

**root@ubuntu:/home/hadoop/docker# docker ps -a -q**

**056d1faa59f3**

**476fae0087be**

**99281f24373c**

**处于终止状态的容器可以通过docker start命令来重新启动：**

**$sudo docker start 05**

**docker restart命令将一个运行状态的容器终止，然后在重新启动它：**

**$sudo docker restart 05**

## 4.3 进入容器

**在使用-d参数时，容器启动后会进入后台，用户无法看到容器的信息。某些时候如果需要进入容器进行操作，有多种方法。**

**docker attach**

**docker exec**

**nsenter**

**attach命令**

**docker attach是Docker自带的命令。**

**$sudo docker run –idt ubuntu**

**exec命令**

**$sudo docker exec –ti ID /bin/bash**

**nsenter工具**

**为了使nsenter连接到容器，还需要找到容器的进程PID,可以通过以下命令获取：**

**PID=$(docker inspect –format “{{ .State.Pid }} <container>)**

**$nsenter –target $PID –mount –uts –ipc –net –pid**

## 4.4 删除容器

$sudo docker rm

$sudo docker stop ID

$sudo docker rmi 镜像ID

## 4.5导入和导出容器

# Chapter5 仓库

Docker Pool(<http://dockerpool.com>)是国内专业的Docker技术社区，目前也提供官方的镜像下载管理服务。

# Chapter 8使用Docker创建镜像

Dockerfile例子：

|  |
| --- |
| #This dockerfile uses the ubuntu image  #command format:Instruction [arguments/command] ..  FROM ubuntu  #  MAINTAINER docker\_user  #  RUN echo "deb http://mirrors.163.com/ubuntu/ xenial main restricted universe multiverse \  deb http://mirrors.163.com/ubuntu/ xenial-security main restricted universe multiverse \  deb http://mirrors.163.com/ubuntu/ xenial-updates main restricted universe multiverse \  deb http://mirrors.163.com/ubuntu/ xenial-proposed main restricted universe multiverse \  deb http://mirrors.163.com/ubuntu/ xenial-backports main restricted universe multiverse \  deb-src http://mirrors.163.com/ubuntu/ xenial main restricted universe multiverse \  deb-src http://mirrors.163.com/ubuntu/ xenial-security main restricted universe multiverse \  deb-src http://mirrors.163.com/ubuntu/ xenial-updates main restricted universe multiverse \  deb-src http://mirrors.163.com/ubuntu/ xenial-proposed main restricted universe multiverse \  deb-src http://mirrors.163.com/ubuntu/ xenial-backports main restricted universe multiverse" > /  etc/apt/sources.list  RUN apt-get update && apt-get install -y nginx  RUN echo "\ndaemon off;" >> /etc/nginx/nginx.conf  CMD /usr/sbin/nginx |

# Chapter9 操作系统

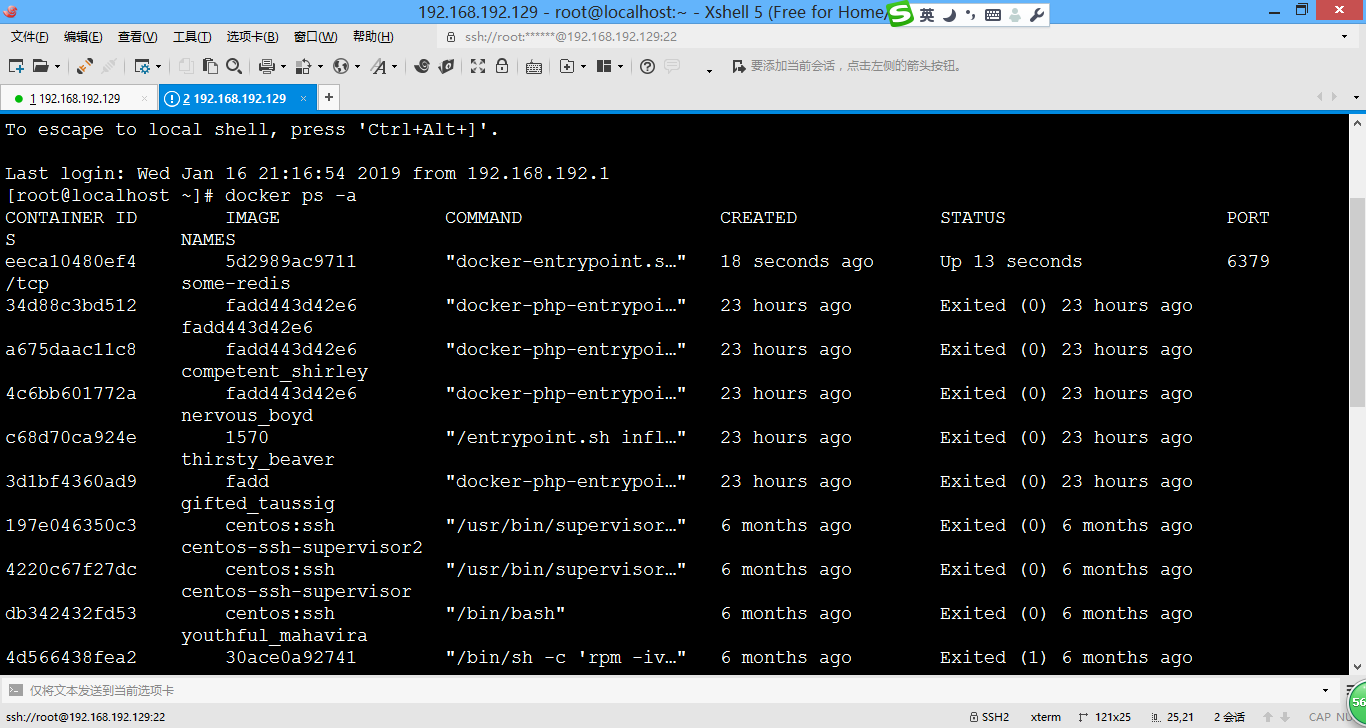
<http://alpha.release.core-os.net/amd64-usr/current/coreos_production_vmware_insecure.zip>

## 阿里云Hub

<https://dev.aliyun.com/detail.html?spm=5176.1972343.2.273.uQOyab&repoId=2038>

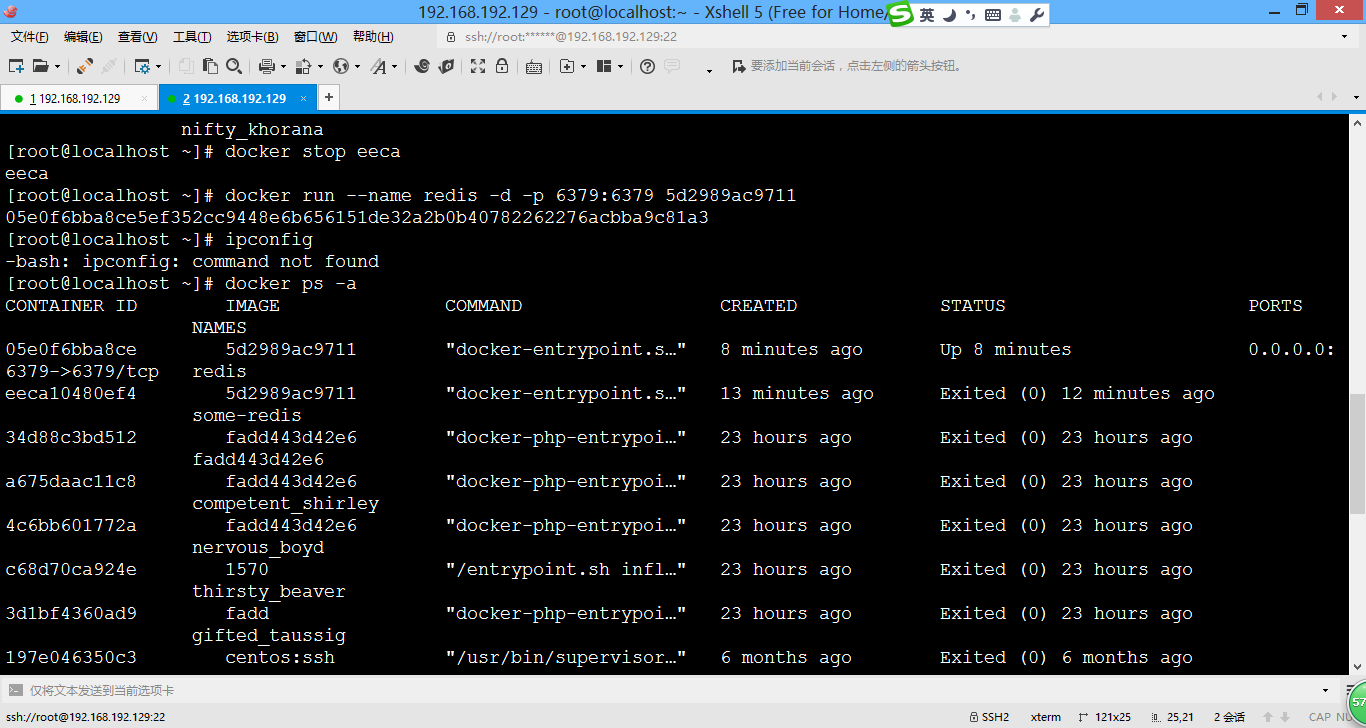
docker pull registry.cn-hangzhou.aliyuncs.com/21epub/ubuntu-python-env

Redis

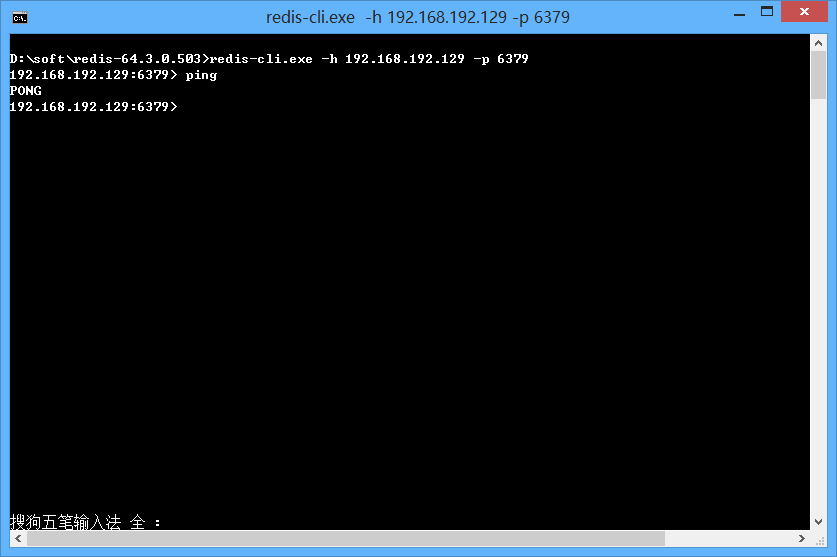


[root@localhost ~]# docker run --name redis -d -p 6379:6379 5d2989ac9711

05e0f6bba8ce5ef352cc9448e6b656151de32a2b0b40782262276acbba9c81a3

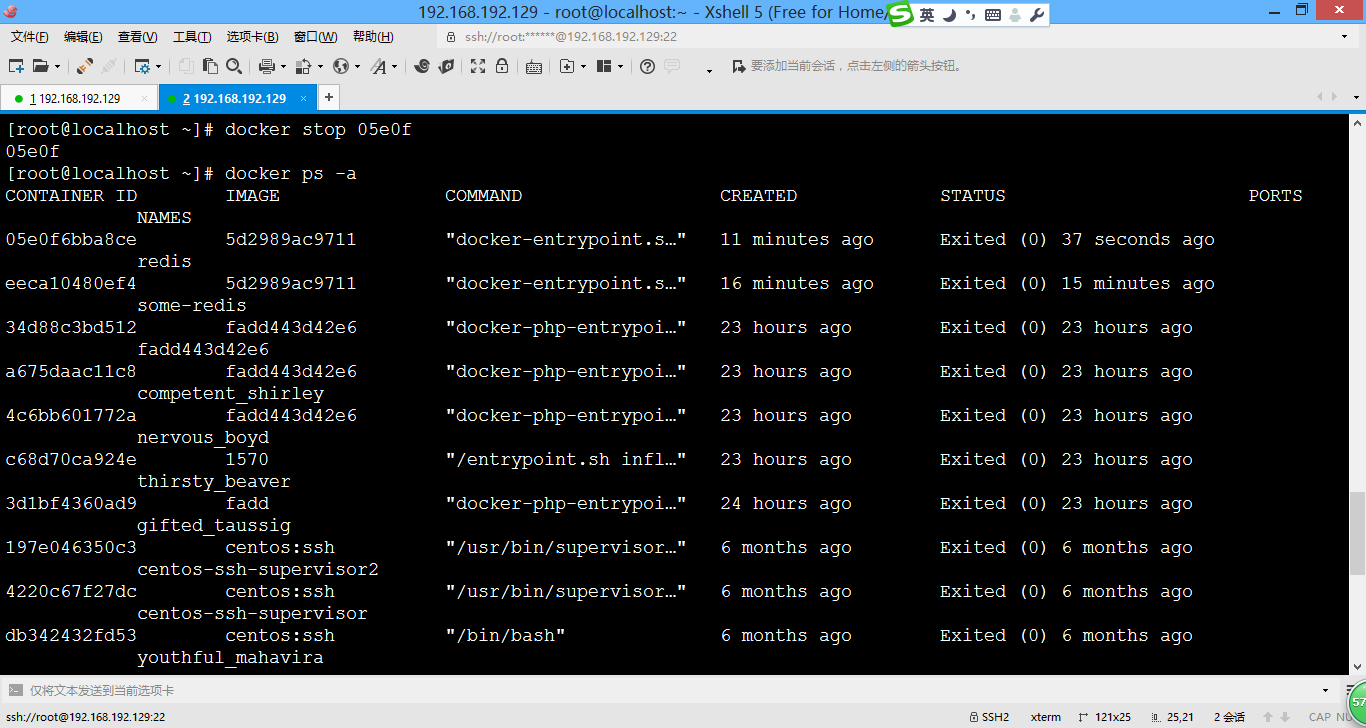


使用redis-cli连接



停止redis实例

docker stop 05e0f



# Influxdb使用

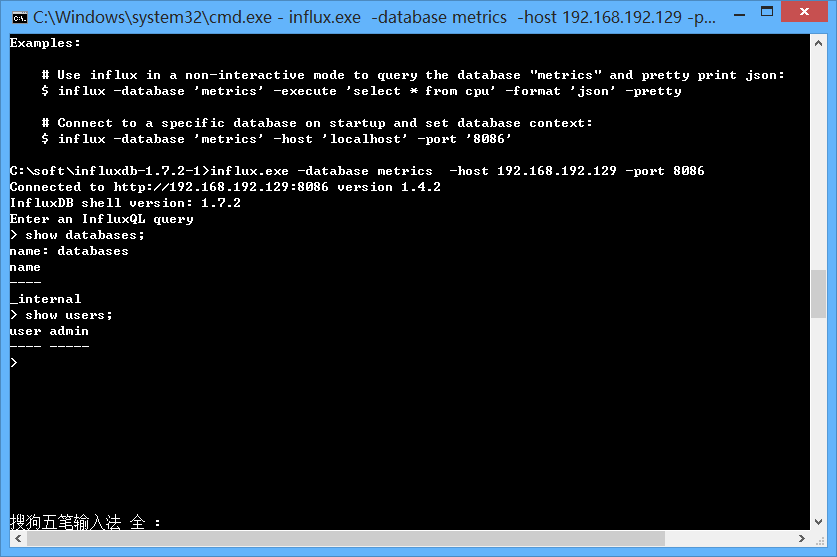
## 先启支influxdb

[root@localhost docker]# docker run --name influxdb -d -p 8086:8086 1570d5fc4d8f

795389cb9ccdec2a215978a4fd3c3ae01f75c44bc7ab69beead0012d9bbda594

## 在windows上面使用influx.exe命令连接上去

influx.exe -database metrics -host 192.168.192.129 -port 8086



# Docker删除容器与镜像

2018年02月27日 14:11:56 [灬点点](https://me.csdn.net/qq_32447301) 阅读数：25943

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/qq\_32447301/article/details/79387649

列出所有容器ID

docker ps -aq

* 1

查看所有运行或者不运行容器

docker ps -a

* 1

停止所有的container（容器），这样才能够删除其中的images：

docker stop $(docker ps -a -q) 或者 docker stop $(docker ps -aq)

* 1

如果想要删除所有container（容器）的话再加一个指令：

docker rm $(docker ps -a -q) 或者 docker rm $(docker ps -aq)

* 1

查看当前有些什么images

docker images

* 1

删除images（镜像），通过image的id来指定删除谁

docker rmi <image id>

* 1

想要删除untagged images，也就是那些id为的image的话可以用

docker rmi $(docker images | grep "^<none>" | awk "{print $3}")

* 1

要删除全部image（镜像）的话

docker rmi $(docker images -q)

* 1

强制删除全部image的话

docker rmi -f $(docker images -q)

* 1

从容器到宿主机复制

docker cp tomcat：/webapps/js/text.js /home/admin

docker cp 容器名: 容器路径 宿主机路径

* 1
* 2

从宿主机到容器复制

docker cp /home/admin/text.js tomcat：/webapps/js

docker cp 宿主路径中文件 容器名 容器路径

* 1
* 2

删除所有停止的容器

docker container prune

* 1

删除所有不使用的镜像

docker image prune --force --all或者docker image prune -f -a

* 1

停止、启动、杀死、重启一个容器

docker stop Name或者ID

docker start Name或者ID

docker kill Name或者ID

docker restart name或者ID

* 1
* 2
* 3
* 4

docker进入容器，查看配置文件

docker exec ：在运行的容器中执行命令

-d :分离模式: 在后台运行

-i :即使没有附加也保持STDIN（标准输入） 打开,以交互模式运行容器，通常与 -t 同时使用；

-t: 为容器重新分配一个伪输入终端，通常与 -i 同时使用；

docker exec -it f94d2c317477 /bin/bash

* 1
* 2
* 3
* 4
* 5

出现root@f94d2c317477:/usr/share/elasticsearch/config# vi elasticsearch.yml  
bash: vi: command not found

apt-get update && apt-get install vim -y

* 1

修改配置、退出容器

1、如果要正常退出不关闭容器，请按Ctrl+P+Q进行退出容器

2、如果使用exit退出，那么在退出之后会关闭容器，可以使用下面的流程进行恢复

使用docker restart命令重启容器

使用docker attach命令进入容器

容器

是从镜像创建并运行实例，就像启动了一个播放程序，它可以被开始、启动、停止、启动和删除。

Docker Hub是Docker官方维护的一个仓库

# 附录

## Docker的安装（以CentOS7 为例）

启动服务 systemctl start docker.service

设置开机启动 systemctl enable docker.service

[root@localhost ~]# docker

## 获取镜像

切换国内镜像 curl -sSL https://get.daocloud.io/daotools/set\_mirror.sh | sh -s <http://ef017c13.m.daocloud.io>

然后再重启 systemctl restart docker

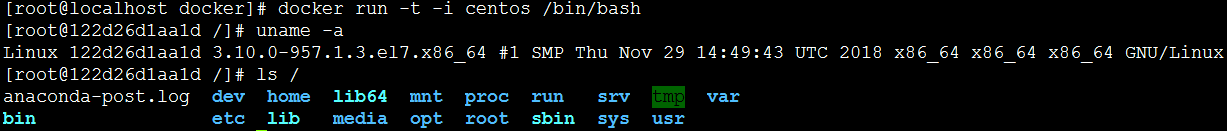
查找start大于100的镜像docker search mariadb –s 100

拉取镜像 docker pull mariadb

## 创建镜像

进入镜像

[root@localhost docker]# docker run -t -i centos /bin/bash



docker commit –m 'my new image' -a 'admin@qq.com' containerId my/sinatra:v2.0.1

## 安装工具箱busybox

Busybox是一个集成了100多个linux命令的工具箱，他在单一可执行文件中提供了精简的UNIX工具箱。该工具箱中包含了常见的简单实用工具如cat、echo、find、grep、find\mount\telnet等，Busybox是Linux系统的瑞士军刀。

[root@localhost ~]# docker pull busybox

[root@localhost ~]# docker run --name budy\_box\_1 busybox cal

February 2019

Su Mo Tu We Th Fr Sa

1 2

3 4 5 6 7 8 9

10 11 12 13 14 15 16

17 18 19 20 21 22 23

24 25 26 27 28

[root@localhost ~]# docker run -it busybox

/ # ping