sudo systemctl start docker#启动服务

docker run ubuntu:15.10 /bin/echo "Hello world"

* **docker:** Docker 的二进制执行文件。
* **run:** 与前面的 docker 组合来运行一个容器。
* **ubuntu:15.10** 指定要运行的镜像，Docker 首先从本地主机上查找镜像是否存在，如果不存在，Docker 就会从镜像仓库 Docker Hub 下载公共镜像。
* **/bin/echo "Hello world":**在启动的容器里执行的命令
* runoob@runoob:~$ docker run -i -t ubuntu:15.10 /bin/bash #退出是exit
* root@0123ce188bd8:/#
* **-t:** 在新容器内指定一个伪终端或终端。
* **-i:** 允许你对容器内的标准输入 (STDIN) 进行交互。

**-**d 后台运行-----想要进入容器需要使用指令

docker exec -it 243c32535da7 /bin/bash

docker attach 1e560fca3906（**注意：** 如果从这个容器退出，会导致容器的停止）

后台模式

docker run -d ubuntu:15.10 /bin/sh -c "while true; do echo hello world; sleep 1; done"

docker ps#查询后台列表(默认活动的 docker ps –a 所有)

在宿主主机内使用 **docker logs** 命令，查看容器内的标准输出：

docker logs 2b1b7a428627

停止

docker stop 4ac789d7df75

拉取镜像

docker pull ubuntu

启动某容器

docker start 2026a83f5f5e

镜像没有vi编辑器

1. apt update
2. apt-get install -y vim

导出镜像

docker export 2026a83f5f5e >ubuntu.tar

**导入容器快照**

可以使用 docker import 从容器快照文件中再导入为镜像，以下实例将快照文件 ubuntu.tar 导入到镜像 test/ubuntu:v1:

cat docker/ubuntu.tar | docker import - test/ubuntu:v1

**删除容器**

docker rm -f 1e560fca3906

下面的命令可以清理掉所有处于终止状态的容器。

$ docker container prune

## 运行一个 web 应用

docker pull training/webapp # 载入镜像

docker run -d -p 5000:5000 training/webapp python app.py

docker port bf08b7f2cd89

docker logs -f bf08b7f2cd89

--检查进程

docker top 22c6768f2735

使用 **docker inspect** 来查看 Docker 的底层信息。它会返回一个 JSON 文件记录着 Docker 容器的配置和状态信息

## 列出镜像列表

我们可以使用 **docker images** 来列出本地主机上的镜像。

### 更新镜像

更新镜像之前，我们需要使用镜像来创建一个容器。

runoob@runoob:~$ docker run -t -i ubuntu:15.10 /bin/bash

root@e218edb10161:/#

在运行的容器内使用 **apt-get update** 命令进行更新。

在完成操作之后，输入 exit 命令来退出这个容器。

此时 ID 为 e218edb10161 的容器，是按我们的需求更改的容器。我们可以通过命令 docker commit 来提交容器副本。

runoob@runoob:~$ docker commit -m="has update" -a="runoob" e218edb10161 runoob/ubuntu:v2

sha256:70bf1840fd7c0d2d8ef0a42a817eb29f854c1af8f7c59fc03ac7bdee9545aff8

各个参数说明：

* **-m:** 提交的描述信息
* **-a:** 指定镜像作者
* **e218edb10161：**容器 ID
* **runoob/ubuntu:v2:** 指定要创建的目标镜像名
* 删除镜像
* docker rmi hello-world

docker run --privileged -it -d centos:centos7 /usr/sbin/init #生成一个新的容器

docker start mysqlbase

docker exec -it mysqlbase bash

mysql -h localhost -u root -p

mysql -h 127.0.0.1 -u root -p

mysql -h 127.0.0.1 -P 33061 -u root -p

我是boy2

CREATE USER 'lllogin'@'%' IDENTIFIED BY '123456';

GRANT ALL ON yaya.\* TO 'lllogin'@'%';

查找mysql配置

find -name my.cnf

无密码进入

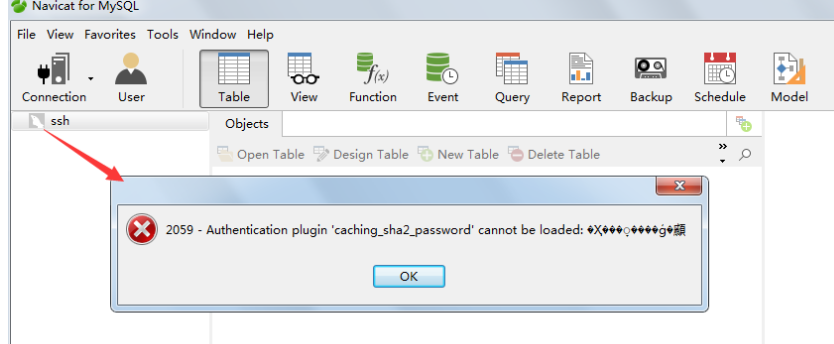
找到[mysqld],在它的后面任意下一行添加如下代码

skip-grant-tables

docker network connect eureka\_net mysqlbase --alias mysqlIP

utf8 -- UTF-8 Unicode

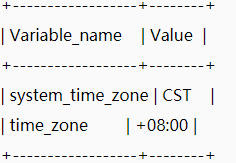
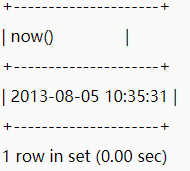
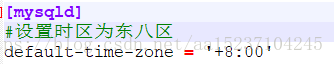
utf8\_general\_ci



alter user 'root'@'%' identified with mysql\_native\_password by 'Admin@123';

set password for 'root'@'localhost' = password('boy2');

根据网友Marksmanbat评论，如果在执行该步骤的时候出现ERROR 1290 (HY000): The MySQL server is running with the --skip-grant-tables option so it cannot execute this statement 错误。则执行下 flush privileges 命令，再执行该命令即可。

1. 设置为北京时间：
   1. mysql> set time\_zone='+8:00';
   2. mysql> Query OK, 0 rows affected (0.00 sec)
2. 再次检查mysql系统时区：
   1. mysql>show variables like '%time\_zone%';
   2. 
3. 再通过select now()来验证时区：
   1. mysql> select now();
   2. 
4. 找到my.ini, 在mysqld 下增加  default-time-zone = '+8:00'
5. 必须放到mysqld 下,放入其它位置无效
6. 

show global variables like 'port';

更改现有容器的接口

1．/var/lib/docker/containers/[hash\_of\_the\_container]/hostconfig.json



2．/var/lib/docker/containers/[hash\_of\_the\_container]/ config.v2.json



# 查看IP

# 1.安装net-tools 命令：yum install net-tools -y 2.查看 docker0 inet addr后面的即为docker容器ip 命令：ifconfig

**Docker-Centos 安装JDK**

yum search java|grep jdk

yum install java-1.8.0-openjdk

# [Docker] docker解决mysql连接慢的问题

##### 配置文件路径:

/etc/mysql/mysql.conf.d

skip-name-resolve

//innodb\_flush\_log\_at\_trx\_commit=0

//sync\_binlog=0

### 运行容器，并且可以通过 exec 命令进入 CentOS 容器。

$ docker run -itd --name centos-test centos:centos7

docker exec -it centosbase /bin/bash

步骤1：先确认容器文件的默认路径

默认情况下为/var/lib/docker/containers

步骤2：进入容器目录

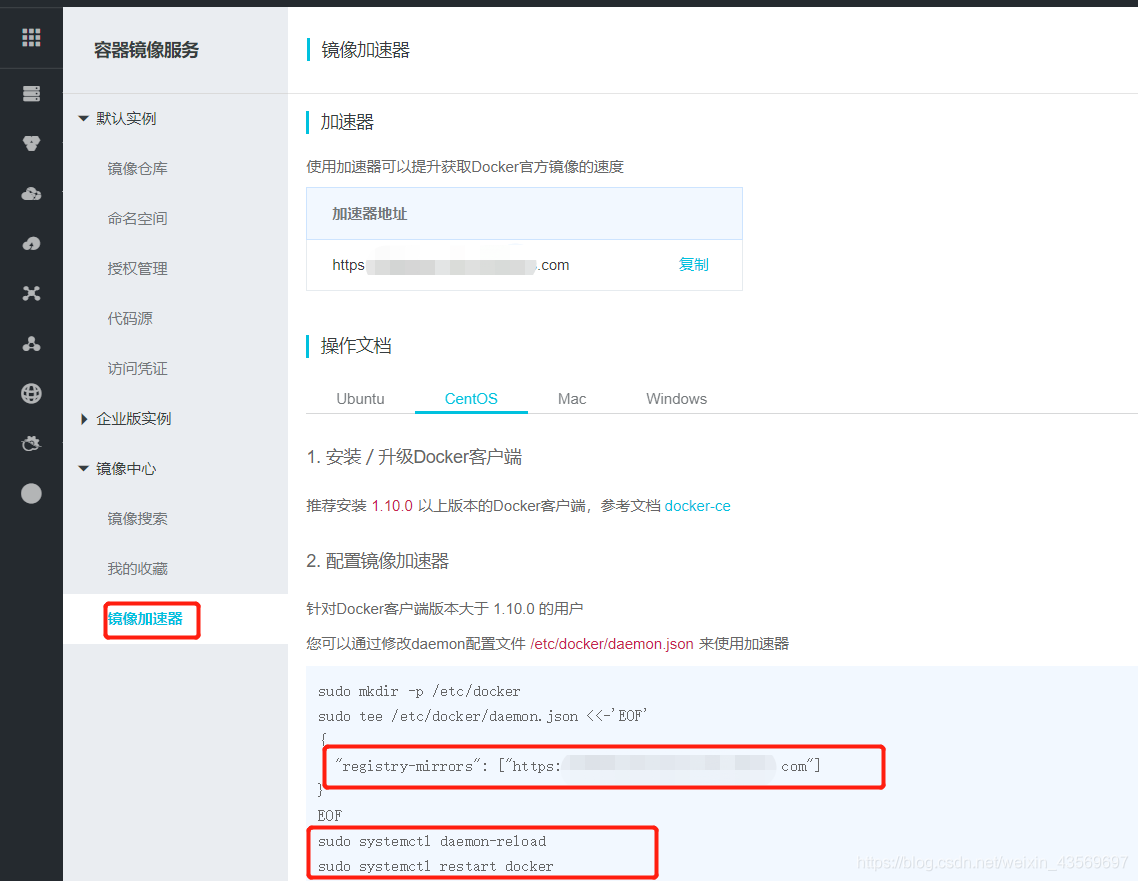
cd  /var/lib/docker/containers

步骤3：查看所有容器的大小  
du -sh \*

步骤3：查看某个容器的大小  
du -sh <容器id>

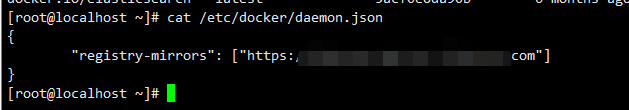
docker cp 宿主机文件路径  镜像名称:镜像中文件存放路径

## docker下载镜像卡死或太慢

找了网上很多方法，使用镜像中国也是下载卡死。  
最后一种成功了，下载速度很满意  
1.先再阿里云注册账号，不需要绑定什么，直接注册就可以了，有账号的直接登录就好。  
地址：https://cr.console.aliyun.com [点击跳转](https://cr.console.aliyun.com/)  
搜索 **容器镜像服务**  
  
2.创建镜像仓库  
  
3.创建好之后点到这个选项卡  
  
在终端输入

vim /etc/docker/daemon.json

* 1

将"registry-mirrors": ["https://......com"] （对应自己的加速地址）复制到文件中  
这是我的格式  
  
最后是重新加载文件和重启docker（不要忘记这两步）

sudo systemctl daemon-reload//重新加载

sudo systemctl restart docker//重启

* 1
* 2

最后就可以幸福的下载镜像了  
docker install mysql(还是这种命令，不需要有任何的修改)

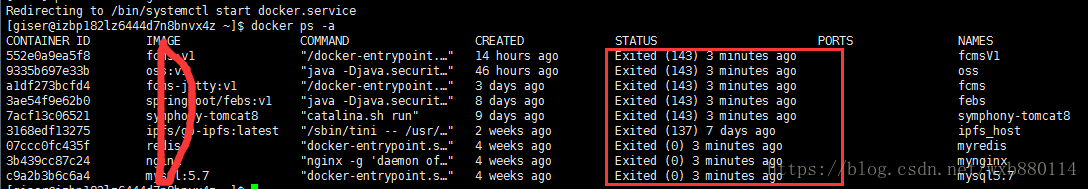
1、docker服务自动重启设置

systemctl enable docker.service

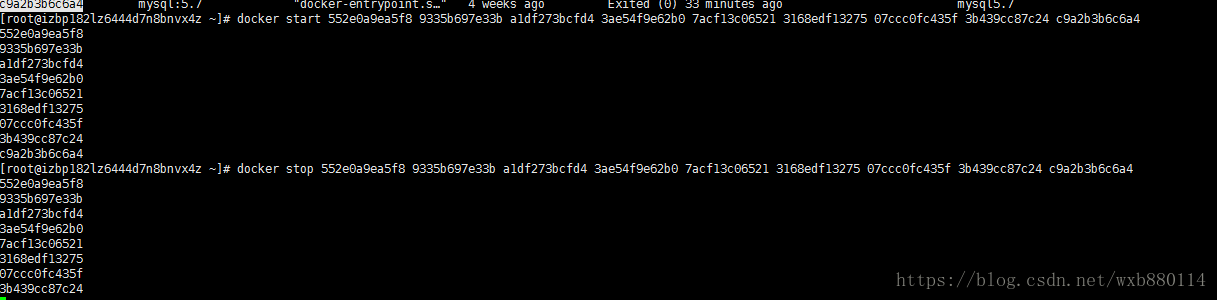
https://img-blog.csdn.net/20180930084945717?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3d4Yjg4MDExNA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

2、docker容器自动启动设置

docker ps -a



使用一下命令对镜像从起： docker restart imageid



在运行docker容器时可以加如下参数来保证每次docker服务重启后容器也自动重启：

docker run *--restart=always*

如果已经启动了则可以使用如下命令：

docker update *--restart=always <CONTAINER ID>*

日志

$ docker inspect --format='{{.LogPath}}' <容器ID>

/var/lib/docker/containers/545e06a75cc0ac8f8c1e6f7217455660187124a3eed031b5eb2f6f0edeb426cb/545e06a75cc0ac8f8c1e6f7217455660187124a3eed031b5eb2f6f0edeb426cb-json.log

echo > 日志路径

$ echo "" > $(docker inspect --format='{{.LogPath}}' <容器ID>)

echo "" > $(docker inspect --format='{{.LogPath}}' 00ffe61bf50f) #master

echo "" > $(docker inspect --format='{{.LogPath}}' 0ad8e8d298b9) #master

docker logs --tail=100 <容器ID>