## 使用 yum 安装（CentOS 7下）

Docker 要求 CentOS 系统的内核版本高于 3.10 ，查看本页面的前提条件来验证你的CentOS 版本是否支持 Docker 。

通过 uname -r 命令查看你当前的内核版本

安装一些必要的系统工具：

sudo yum install -y yum-utils device-mapper-persistent-data lvm2

添加软件源信息：

sudo yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo

更新 yum 缓存：

sudo yum makecache fast

安装 Docker-ce：

sudo yum -y install docker-ce

启动 Docker 后台服务

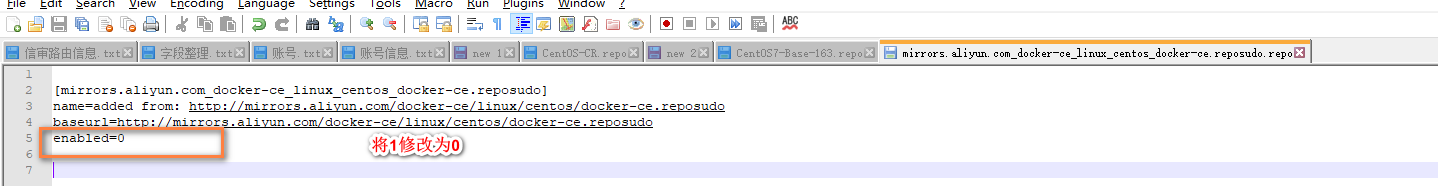
sudo systemctl start docker

>[root@zabbix yum.repos.d]# yum clean all  　　　　--清空yum缓存的软件包和header  
　　>[root@zabbix yum.repos.d]# yum makecache  　　  --把yum源中的包信息进行缓存

Yum update

yum -y install wget

这是执行sudo yum makecache fast命令失败提示404



## 删除 Docker CE

执行以下命令来删除 Docker CE：

$ sudo yum remove docker-ce

$ sudo rm -rf /var/lib/docker

## 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。

各个选项说明:

* **REPOSITORY：**表示镜像的仓库源
* **TAG：**镜像的标签
* **IMAGE ID：**镜像ID
* **CREATED：**镜像创建时间
* **SIZE：**镜像大小
* 同一仓库源可以有多个 TAG，代表这个仓库源的不同个版本，如ubuntu仓库源里，有15.10、14.04等多个不同的版本，我们使用 REPOSITORY:TAG 来定义不同的镜像。
* 所以，我们如果要使用版本为15.10的ubuntu系统镜像来运行容器时，命令如下：
* runoob@runoob:~$ docker run -t -i ubuntu:15.10 /bin/bash
* root@d77ccb2e5cca:/#

## 获取一个新的镜像

当我们在本地主机上使用一个不存在的镜像时 Docker 就会自动下载这个镜像。如果我们想预先下载这个镜像，我们可以使用 docker pull 命令来下载它。

## 查找镜像

我们可以从 Docker Hub 网站来搜索镜像，Docker Hub 网址为： https://hub.docker.com/

我们也可以使用 docker search 命令来搜索镜像。比如我们需要一个httpd的镜像来作为我们的web服务。我们可以通过 docker search 命令搜索 httpd 来寻找适合我们的镜像。

runoob@runoob:~$ docker search httpd

**NAME:**镜像仓库源的名称

**DESCRIPTION:**镜像的描述

**OFFICIAL:**是否docker官方发布

下载完成后，我们就可以使用这个镜像了。

runoob@runoob:~$ docker run httpd

**创建镜像**

当我们从docker镜像仓库中下载的镜像不能满足我们的需求时，我们可以通过以下两种方式对镜像进行更改。

* 1.从已经创建的容器中更新镜像，并且提交这个镜像
* 2.使用 Dockerfile 指令来创建一个新的镜像

**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

我们可以使用 **docker images** 命令来查看我们的新镜像 **runoob/ubuntu:v2**：

runoob@runoob:~$ docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

runoob/ubuntu v2 70bf1840fd7c 15 seconds ago 158.5 MB

ubuntu 14.04 90d5884b1ee0 5 days ago 188 MB

php 5.6 f40e9e0f10c8 9 days ago 444.8 MB

nginx latest 6f8d099c3adc 12 days ago 182.7 MB

mysql 5.6 f2e8d6c772c0 3 weeks ago 324.6 MB

httpd latest 02ef73cf1bc0 3 weeks ago 194.4 MB

ubuntu 15.10 4e3b13c8a266 4 weeks ago 136.3 MB

hello-world latest 690ed74de00f 6 months ago 960 B

training/webapp latest 6fae60ef3446 12 months ago 348.8 MB

使用我们的新镜像 **runoob/ubuntu** 来启动一个容器

runoob@runoob:~$ docker run -t -i runoob/ubuntu:v2 /bin/bash

root@1a9fbdeb5da3:/#





docker run -di --name ldymysql -p 33306:3306 -e MYSQL\_ROOT\_PASSWORD=123456 mysql

docker exec -it ldymysql /bin/bash

mysql -u root -p

GRANT ALL ON \*.\* TO 'root'@'%';

ALTER USER 'root'@'localhost' IDENTIFIED BY 'password' PASSWORD EXPIRE NEVER;

ALTER USER 'root'@'%' IDENTIFIED WITH mysql\_native\_password BY '123456';

flush privileges;

## docker ps：列出所有正在运行的容器

## docker image ls：列出所有的镜像

## docker stop <name>：暂停容器

## docker rm <name>：删除容器

### docker ps -as 查看全部container，包括exited的容器，并列出容器的大小

## docker命令

1.docker images 列出本地主机上的镜像。

2.docker pull 从Docker Hub中拉取或者更新指定镜像。

3.docker run

-d 标识是让 docker 容器在后台运行。

-p 标识通知 Docker 将容器内部使用的网络端口映射到我们使用的主机上。

第一个是外部,第二个是内部

–name 定义一个容器的名字，如果在执行docker run时没有指定Name，那么deamon会自动生成一个随机数字符串当做UUID。

-e 设置环境变量，或者覆盖已存在的环境变量。

例如：docker run -di --name ldymysql -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=123456 mysql

含义：容器的名字为mysql，将容器的3306端口映射到本机的3306端口，mysql数据库的密码为password

，运行的镜像为mysql/mysql-server:latest

4.docker ps

-a 查看已经创建的容器

-s 查看已经启动的容器

5.docker start con\_name 启动容器名为con\_name的容器。

6.docker stop con\_name 停止容器名为con\_name的容器。

7.docker rm con\_name 删除容器名为con\_name的容器。

8.docker rmi img\_name 删除镜像名为img\_name的镜像。

9.docker rename old\_name new\_name 重命名一个容器。

---------------------

Docker start id/name 都能启动容器

mysql -u root -p

## [Linux查看某个端口是否启动](https://www.cnblogs.com/gouge/p/7089997.html)

#### 查看命令

netstat -an | grep 16010

查看端口号是否开启

lsof -i :3306

docker pull tomcat

docker run --name ldytomcat -p 8080:8080 -v $PWD/test:/usr/local/tomcat/webapps/test -d tomcat

docker pull zookeeper

docker pull double

## docker 操作命令

启动        systemctl start docker

守护进程重启   sudo systemctl daemon-reload

重启docker服务   systemctl restart  docker

重启docker服务  sudo service docker restart

关闭docker service docker stop

关闭docker systemctl stop docker

## Nginx

docker pull nginx

docker run --name ldyngix -p 80:80 -d nginx

##### 创建挂载目录

我们这里需要挂载可手动修改配置文件的nginx，而不是全部封装在docker容器中运行的。

mkdir -p /nginx/{conf,conf.d,html,logs}

##### 编写nginx.conf配置文件

随便找一份就行

### 执行挂载命令

docker run --name ldynginx -d -p 80:80 -v /nginx/conf/nginx.conf:/etc/nginx/nginx.conf -v /nginx/logs:/var/log/nginx -d docker.io/nginx

## fastdfs

1.查找Docker Hub上的redis镜像

docker search fastdfs



mkdir /usr/local/fastdfs/etc/

mkdir /usr/local/fastdfs/data/storage\_data

mkdir /usr/local/fastdfs/data/store\_path

mkdir /usr/local/fastdfs/data/tracker\_data

## 获取配置文件

启动一个fastdfs的docker容器，查看容器id，从容器中下载配置文件并且下载到上面创建的 /usr/local/fastdfs/etc/ 目录中.

docker run -ti --name ldyfdfs --net=host delron/fastdfs sh

docker ps -a

docker cp -a a50dc32fd594:/fdfs\_conf/. /usr/local/fastdfs/etc