1.安装Docker

​​ 由于我们的环境都使用的是CentOS，因此这里我们将Docker安装到CentOS上。注意：这里建议安装在CentOS7.x以上的版本，在CentOS6.x的版本中，安装前需要安装其他很多的环境而且Docker很多补丁不支持更新。

（1）yum 包更新到最新

sudo yum update

（2）安装需要的软件包， yum-util 提供yum-config-manager功能，另外两个是devicemapper驱动依赖的

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

（3）设置yum源为阿里云

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

（4）安装docker

sudo yum install docker-ce

（5）安装后查看docker版本

docker -v

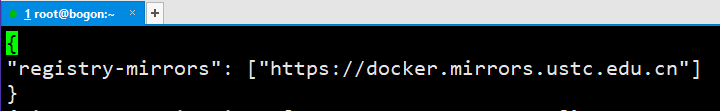
2.设置ustc的镜像

编辑该文件：

vi /etc/docker/daemon.json

在该文件中输入如下内容：

{ "registry-mirrors": ["https://docker.mirrors.ustc.edu.cn"] }



3.Docker的启动与停止

**systemctl**命令是系统服务管理器指令

启动docker：

systemctl start docker

停止docker：

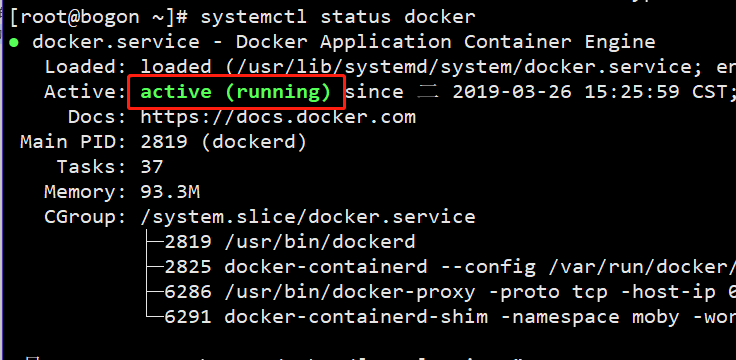
systemctl stop docker

重启docker：

systemctl restart docker

查看docker状态：

systemctl status docker



开机启动：

systemctl enable docker

查看docker概要信息

docker info

查看docker帮助文档

docker --help

4常用命令

4.1镜像相关命令

**4.1.1 查看镜像**

docker images

REPOSITORY：镜像名称

TAG：镜像标签

IMAGE ID：镜像ID

CREATED：镜像的创建日期（不是获取该镜像的日期）

SIZE：镜像大小

这些镜像都是存储在Docker宿主机的/var/lib/docker目录下

**4.1.2 搜索镜像**

如果你需要从网络中查找需要的镜像，可以通过以下命令搜索

docker search 镜像名称

NAME：仓库名称

DESCRIPTION：镜像描述

STARS：用户评价，反应一个镜像的受欢迎程度

OFFICIAL：是否官方

AUTOMATED：自动构建，表示该镜像由Docker Hub自动构建流程创建的

**4.1.3 拉取镜像**

拉取镜像就是从中央仓库中下载镜像到本地

docker pull 镜像名称

例如，我要下载centos7镜像

docker pull centos:7

**4.1.4 删除镜像**

按镜像ID删除镜像

docker rmi 镜像ID

删除所有镜像

docker rmi `docker images -q`

4.2容器相关命令

### 4.2.1 查看容器

查看正在运行的容器

docker ps

查看所有容器

docker ps –a

查看最后一次运行的容器

docker ps –l

查看停止的容器

docker ps -f status=exited

### 4.2.2 创建与启动容器

创建容器常用的参数说明：

创建容器命令：docker run

-i：表示运行容器

-t：表示容器启动后会进入其命令行。加入这两个参数后，容器创建就能登录进去。即分配一个伪终端。

--name :为创建的容器命名。

-v：表示目录映射关系（前者是宿主机目录，后者是映射到宿主机上的目录），可以使用多个－v做多个目录或文件映射。注意：最好做目录映射，在宿主机上做修改，然后共享到容器上。

-d：在run后面加上-d参数,则会创建一个守护式容器在后台运行（这样创建容器后不会自动登录容器，如果只加-i -t两个参数，创建后就会自动进去容器）。

-p：表示端口映射，前者是宿主机端口，后者是容器内的映射端口。可以使用多个-p做多个端口映射

（1）交互式方式创建容器

docker run -it --name=容器名称 镜像名称:标签 /bin/bash

这时我们通过ps命令查看，发现可以看到启动的容器，状态为启动状态

退出当前容器

exit

（2）守护式方式创建容器：

docker run -di --name=容器名称 镜像名称:标签

登录守护式容器方式：

docker exec -it 容器名称 (或者容器ID) /bin/bash

### 4.2.3 停止与启动容器

停止容器：

docker stop 容器名称（或者容器ID）

启动容器：

docker start 容器名称（或者容器ID）

### 4.2.4 文件拷贝

如果我们需要将文件拷贝到容器内可以使用cp命令

docker cp 需要拷贝的文件或目录 容器名称:容器目录

也可以将文件从容器内拷贝出来

docker cp 容器名称:容器目录 需要拷贝的文件或目录

### 4.2.5 目录挂载

我们可以在创建容器的时候，将宿主机的目录与容器内的目录进行映射，这样我们就可以通过修改宿主机某个目录的文件从而去影响容器。 创建容器 添加-v参数 后边为 宿主机目录:容器目录，例如：

docker run -di --name mysql -p33306:3306 -e MYSQL\_ROOT\_PASSWORD=123456 -v /usr/loacl/myhtml:/usr/local/tomcat/webapps --privileged=true mysql:5.6

如果你共享的是多级的目录，可能会出现权限不足的提示。

这是因为CentOS7中的安全模块selinux把权限禁掉了，我们需要添加参数 --privileged=true 来解决挂载的目录没有权限的问题

### 4.2.6 查看容器IP地址

我们可以通过以下命令查看容器运行的各种数据

docker inspect 容器名称（容器ID）

也可以直接执行下面的命令直接输出IP地址

docker inspect --format='{{.NetworkSettings.IPAddress}}' 容器名称（容器ID）

### 4.2.6 查看容器IP地址

我们可以通过以下命令查看容器运行的各种数据

docker inspect 容器名称（容器ID）

也可以直接执行下面的命令直接输出IP地址

docker inspect --format='{{.NetworkSettings.IPAddress}}' 容器名称（容器ID）

### 4.2.7 删除容器

删除指定的容器：

docker rm 容器名称（容器ID）

5.MYSQL5.6操作

设置好Docker,设置ustc镜像,启动Docker

安装MYSQL5.6

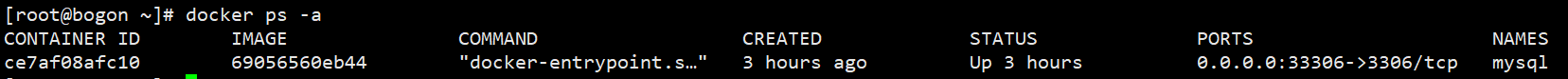
拉取mysql5.6镜像. 命令 docker pull mysql:5.6

拉取完成后查看下镜像 命令 docker images

创建mysql5.6容器,目录挂载模式

docker run -di --name mysql -p3306:3306 -e MYSQL\_ROOT\_PASSWORD=123456 -v /usr/loacl/myhtml:/usr/local/tomcat/webapps --privileged=true mysql:5.6

查看mysql5.6容器运行状 命令 docker ps -a



安装JDK1.8 将下载好的JKD1.8上传到宿主机

参考: <https://blog.csdn.net/remote_roamer/article/details/56011868>

复制gz文件到容器中的root的用户目录下 命令

docker cp root/jdk-8u121-linux-x64.tar.gz <容器ID>:/root

3.进入容器

docker exec -it <容器名称或者ID> bash

4. 在容器中解开gz文件

cd ~

tar -zxvf jdk-8u121-linux-x64.tar.gz

chown root:root ~/jdk1.8.0\_121 -R

5. 建立容器系统的JAVA\_HOME目录

mkdir /usr/lib/jvm

6.移动jdk目录到JAVA\_HOME目录

mv ~/jdk1.8.0\_121 /usr/lib/jvm

7.安装vim（可选）

apt-get update

apt-get install vim

8.设置环境变量

vim ~/.bashrc

#set oracle jdk environment

export JAVA\_HOME=/usr/lib/jvm/jdk1.8.0\_121 ## 这里要注意目录要换成自己解压的jdk 目录

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

export PATH=${JAVA\_HOME}/bin:$PATH

9.使得环境变量马上生效

source ~/.bashrc

10.验证JDK版本

java -version

迁移和备份

## 6.1 容器保存为镜像

我们可以通过以下命令将容器保存为镜像

docker commit mynginx mynginx\_i

## 6.2 镜像备份

我们可以通过以下命令将镜像保存为tar 文件

docker save -o mynginx.tar mynginx\_i

## 6.3 镜像恢复与迁移

首先我们先删除掉mynginx\_img镜像 然后执行此命令进行恢复

docker load -i mynginx.tar

-i 输入的文件

执行后再次查看镜像，可以看到镜像已经恢复