

# 9、Docker

2020年6月10日 19:03

- 1、简介

**Docker**是一个开源的应用容器引擎；是一个轻量级容器技术；

Docker支持将软件编译成一个镜像；然后在镜像中各种软件做好配置，将镜像发布出去，其他使用者可以直接使用这个镜像；

运行中的这个镜像称为容器，容器启动是非常快速的。

- 2、核心概念

docker主机(Host): 安装了Docker程序的机器 (Docker直接安装在操作系统之上) ；

docker客户端(Client): 连接docker主机进行操作；

docker仓库(Registry): 用来保存各种打包好的软件镜像；

docker镜像(Images): 软件打包好的镜像；放在docker仓库中；

docker容器(Container): 镜像启动后的实例称为一个容器；容器是独立运行的一个或一组应用

使用Docker的步骤：

- 1)、安装Docker
- 2)、去Docker仓库找到这个软件对应的镜像；
- 3)、使用Docker运行这个镜像，这个镜像就会生成一个Docker容器；
- 4)、对容器的启动停止就是对软件的启动停止；

- 3、安装Docker

- 在linux虚拟机上安装docker

步骤：

- 1、检查内核版本，必须是3.10及以上  
`uname -r`
- 2、安装docker  
`yum install docker`
- 3、输入y确认安装
- 4、启动docker  
`[root@localhost ~]# systemctl start docker`  
`[root@localhost ~]# docker -v`  
Docker version 1.12.6, build 3e8e77d/1.12.6
- 5、开机启动docker  
`[root@localhost ~]# systemctl enable docker`  
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
- 6、停止docker  
`systemctl stop docker`

- 4、docker常用命令&操作

- 1) 镜像操作

操作	命令	说明
检索	<code>docker search 关键字 eg:</code> <code>docker search redis</code>	我们经常去docker hub上检索镜像的详细信息，如镜像的TAG。

拉取	docker pull 镜像名:tag	:tag是可选的, tag表示标签, 多为软件 的版本, 默认是latest
列表	docker images	查看所有本地镜像
删除	docker rmi image-id	删除指定的本地镜像

## ○ 2) 容器操作

软件镜像 (QQ安装程序) ----运行镜像----产生一个容器 (正在运行的软件, 运行的QQ) ;

步骤:

- 1、搜索镜像  
[root@localhost ~]# **docker search tomcat**
- 2、拉取镜像  
[root@localhost ~]# **docker pull tomcat**
- 3、根据镜像启动容器  
**docker run --name mytomcat -d tomcat:latest**
- 4、**docker ps**  
查看运行中的容器
- 5、停止运行中的容器  
**docker stop 容器的id**
- 6、查看所有的容器  
**docker ps -a**
- 7、启动容器  
**docker start 容器id**
- 8、删除一个容器  
**docker rm 容器id**
- 9、启动一个做了端口映射的tomcat  
[root@localhost ~]# **docker run -d -p 8888:8080 tomcat**  
-d: 后台运行  
-p: 将主机的端口映射到容器的一个端口 主机端口:容器内部的端口
- 10、为了演示简单关闭了linux的防火墙  
**service firewalld status**: 查看防火墙状态  
**service firewalld stop**: 关闭防火墙
- 11、查看容器的日志  
**docker logs container-name/container-id**

更多命令参看

<https://docs.docker.com/engine/reference/commandline/docker/>  
可以参考每一个镜像的文档

## ○ 3) 安装MySQL

**docker pull mysql**

[root@localhost ~]# **docker run --name mysql01 -e  
MYSQL\_ROOT\_PASSWORD=root -d mysql**

b874c56bec49fb43024b3805ab51e9097da779f2f572c22c695305dedd684c5f

[root@localhost ~]# **docker ps**

CONTAINER ID	IMAGE	COMMAND NAMES	CREATED
b874c56bec49	mysql	"docker-entrypoint.sh"	4 seconds ago
Up 3 seconds	3306/tcp	mysql01	

做了端口映射

```
[root@localhost ~]# docker run -p 3306:3306 --name mysql02 -e
MYSQL_ROOT_PASSWORD=root -d mysql
ad10e4bc5c6a0f61cbad43898de71d366117d120e39db651844c0e73863b9
434
```

```
[root@localhost ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	
ad10e4bc5c6a	mysql	"docker-entrypoint.sh"	4 seconds ago
Up 2 seconds	0.0.0.0:3306->3306/tcp	mysql02	

几个其他的高级操作

```
docker run --name mysql03 -v /conf/mysql:/etc/mysql/conf.d -e
MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:tag
```

把主机的/conf/mysql文件夹挂载到 mysqldocker容器的/etc/mysql/conf.d文件夹里面

改mysql的配置文件就只需要把mysql配置文件放在自定义的文件夹下 (/conf/mysql)

```
docker run --name some-mysql -e MYSQL_ROOT_PASSWORD=my-secret-
pw -d mysql:tag --character-set-server=utf8mb4 --collation-
server=utf8mb4_unicode_ci
```

指定mysql的一些配置参数

4) 启动安装好的MySQL

```
[root@localhost ~]# docker restart f712a2f0c808
f712a2f0c808
```