链滴

# Docker 小白入门

作者：jchain

原文链接：https://ld246.com/article/1593746317676

来源网站：链滴

# 1 docker概述

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的镜像，然后发布到任何流行的 Linux或Windows 机器上，也可以实现虚拟化。容器是完全使用沙箱机制相互之间不会有任何接口。

# 2 docker安装

```
# 1、yum 包更新到最新
yum update -y

# 2、安装所需软件包。yum-util 提供yum-config-manager功能，另外两个是devicemapper驱动赖
yum install -y yum-utils device-mapper-persistent-data lvm2

# 3、设置yum 源 （只需要执行一种）
yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
yum-config-manager --add-repo https://mirrors.ustc.edu.cn/docker-ce/linux/centos/docker-e.repo
yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.epo

# 4、安装docker,出现输入的提示都按 y
yum install -y docker-ce

# 5、查看docker版本
docker -v
```

# 3 docker简单概念(架构)

主要分为3块：

- image镜像(类)
- container容器（对象、实例）
- repo仓库

使用过程和Git操作类似。

1. 首先有一个远程仓库（就比如 github,gitee或者你自己搭建的私服。docker的远程仓库就叫做DockerHub）,远程仓库中放的都是一个个的镜像（就比如你github中的每个代码仓库）

2. 然后从仓库中拉取某个镜像到本地（镜像名称+版本）

3. 最后就是创建一个镜像的实例，比如有一个tomcat的镜像，你可以根据镜像创建多个容器（指定同的端口就行了）

# 4 配置镜像加速器

默认是从 DockerHub 上拉取镜像，国外速度比较慢，所以换个镜像源。镜像源有很多，我这里使用阿里云的镜像加速器

```
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": ["https://3k7haef0.mirror.aliyuncs.com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

# 5 docker命令

## 5.1 docker服务相关命令

也就是计算机本身对 docker 操作

```
# 启动 docker 服务
systemctl start docker

# 停止 docker 服务
systemctl stop docker

# 重启 docker 服务
systemctl restart docker

# 查看 docker 服务状态
systemctl status docker

# 设置 docker 开机启动
systemctl enable docker
```

## 5.2 docker镜像相关命令

在docker中 对镜像的操作，主要包括 远程镜像到本地镜像的拉取，删除操作。当然也包括push到远镜像仓库(就像push代码到github上一样）这里暂时没有push操作

```
# 搜索远程镜像,xxx标识搜索的内容，比如redis,mysql
docker search xxx

# 查看本地镜像
docker images

# 查看所有镜像 id
docker images -q

# 拉取镜像,:后面是版本,如果不写版本号，则是最新的 latest
docker pull xxx
docker pull xxx:version

# 删除镜像, remove image
docker rmi 镜像id(Image_Id)
docker rmi 名称:版本

# 删除所有镜像
docker rmi `docker images -q`
```

## 5.3 docker 容器相关命令

也就是我在远程仓库上拉取了一个镜像到本地，比如是tomcat镜像，此时我需要从这个镜像创建一实例

（相当于有个Tomcat的类，我需要创建一个tomcat对象的过程 new Tomcat() ）

```
# 查看容器
docker ps
docker ps -a  # 查看所有的

# 创建容器
# -i:保持容器运行。
# -t: 分配一个终端，通常与 -i一起使用
# -d: 后台运行创建容器（创建容器后不会立即进入容器），以守护模式运行容器，可以使用docker exec进入容器，并退出后容器不会退出
# -it 创建的容器一般为交互式容器，-id 创建的是守护式容器
# --name 为容器起个名字
docker run -i -t --name=xxx1 redis:5.0 /bin/bash
docker run -i -d --name=xxx2 redis:5.0 /bin/bash

# 进入容器
docker exec -it xxx2 /bin/bash

# 跳出容器
exit

# 启动容器 xxx为容器名称
```

```
docker start xxx

# 停止容器 xxx 为 容器名称
docker stop xxx

# 删除容器 （开启的容器不能删除）
docker rm xxx/id
docker rm `docker ps -aq` # 删除所有的容器

# 查看容器信息 xxx为容器的名称
docker inspect xxx
```

# 6 docker容器的数据卷

## 6.1 数据卷的概念和作用

其实就是可以让容器内部的文件可以直接与宿主机上的某个目录互通（数据一致）

简单来说就是这样的一个关系。本身docker是Linux上的一个软件，但是在这个docker软件中呢我们去拉取很多镜像并且创建很多个容器，创建的每个容器之间是隔离的（沙箱机制），那么对于每个容的配置我们每次都进入到（docker exec）容器内部去修改才行,比如有一个tomcat容器，我要去修改口号，那么可以进入到容器内部，然后找到tomcat 修改 server.xml 中的port内容，然后再重启容器这样其实是比较麻烦，那么可以有一种方式就是把 这个容器内部的server.xml文件直接映射到宿主机的某个文件，这样每次只需要修改宿主机上的这个文件就可以了。

**数据卷**：就是是宿主机上的某个目录或者文件，当容器内目录和宿主机目录绑定后，对方的修改会立同步。

一个数据卷可以被多个容器挂载，一个容器也可以挂载多个数据卷

**作用**：外部容器和容器的数据打通，容器之间数据打通（通过同时挂载相同的数据卷）

## 6.2 如何配置数据卷?

- 创建容器时，使用 -v 参数设置数据卷

```
docker run ... -v 宿主机目录(文件):容器内目录(文件) ...
```

注意：

1. 目录必须是绝对路径 (/开头的)
2. 如果目录不存在会自动创建
3. 可以挂载多个数据卷 （多个 -v）

## 6.3 配置数据卷容器

1. 多容器数据交换（的方式）：
    1. 多个容器挂载同一个数据
    2. 数据卷容器

2. 配置

1. 创建一个容器 （数据卷容器c3），使用 -v 参数设置数据卷

```
docker run -it --name=c3 -v /volume centos:7 /bin/bash
```

2. 创建启动c1,c2容器，使用 --volumes-from 参数 设置数据卷

```
docker run -it --name=c1 --volumes-from c3 centos:7 /bin/bash
docker run -it --name=c2 --volumes-from c3 centos:7 /bin/bash
```

# 7 docker部署应用

操作步骤：

- 搜索xxx镜像
- 拉取xxx镜像
- 创建容器
- 操作容器中的xxx