

docker学习笔记

docker的基本组成

1. 镜像

Docker 镜像（Image）就是一个只读的模板。镜像可以用来创建 Docker 容器，一个镜像可以创建很多容器。它也相当于是一个root文件系统。比如官方镜像 centos:7 就包含了完整的一套 centos:7 最小系统的 root 文件系统。相当于容器的“源代码”，docker镜像文件类似于Java的类模板，而docker容器实例类似于java中新出来的实例对象。

容器与镜像的关系类似于面向对象编程中的对象与类。

Docker	面向对象
容器	对象
镜像	类

2. 容器

1 从面向对象角度

Docker 利用容器（Container）独立运行的一个或一组应用，应用程序或服务运行在容器里面，容器就类似于一个虚拟化的运行环境，容器是用镜像创建的运行实例。就像是Java中的类和实例对象一样，镜像是静态的定义，容器是镜像运行时的实体。容器为镜像提供了一个标准的和隔离的运行环境，它可以被启动、开始、停止、删除。每个容器都是相互隔离的、保证安全的平台

2 从镜像容器角度

可以把容器看做是一个简易版的Linux 环境（包括root用户权限、进程空间、用户空间和网络空间等）和运行在其中的应用程序。

3. 仓库

仓库（Repository）是集中存放镜像文件的场所。

类似于

Maven仓库，存放各种jar包的地方；

github仓库，存放各种git项目的地方；

Docker公司提供的官方registry被称为Docker Hub，存放各种镜像模板的地方。

仓库分为公开仓库（Public）和私有仓库（Private）两种形式。

最大的公开仓库是 Docker Hub(<https://hub.docker.com/>)，

存放了数量庞大的镜像供用户下载。国内的公开仓库包括阿里云、网易云等

小总结

需要正确的理解仓库/镜像/容器这几个概念:

Docker 本身是一个容器运行载体或称之为管理引擎。我们把应用程序和配置依赖打包好形成一个可交付的运行环境，这个打包好的运行环境就是image镜像文件。只有通过这个镜像文件才能生成Docker容器实例(类似Java中新出来一个对象)。

image文件可以看作是容器的模板。Docker 根据 image 文件生成容器的实例。同一个 image 文件，可以生成多个同时运行的容器实例。

镜像文件

* image 文件生成的容器实例，本身也是一个文件，称为镜像文件。

容器实例

* 一个容器运行一种服务，当我们需要的时候，就可以通过docker客户端创建一个对应的运行实例，也就是我们的容器

仓库

* 就是放一堆镜像的地方，我们可以把镜像发布到仓库中，需要的时候再从仓库中拉下来就可以了。

docker的安装

1. 确定是 CentOS7 及以上 的版本

1.1 查看CentOS系统版本的命令

1.1.1 此命令将显示您正在运行的 CentOS 版本的信息

```
cat /etc/redhat-release
```

```
[root@localhost ~]# cat /etc/redhat-release  
CentOS Linux release 7.9.2009 (Core)
```

1.1.2 此命令将显示您正在运行的 CentOS 版本、内核版本和其他相关信息

```
cat /etc/os-release
```

```
[root@localhost ~]# cat /etc/os-release  
NAME="CentOS Linux"  
VERSION="7 (Core)"  
ID="centos"  
ID_LIKE="rhel fedora"  
VERSION_ID="7"  
PRETTY_NAME="CentOS Linux 7 (Core)"  
ANSI_COLOR="0;31"  
CPE_NAME="cpe:/o:centos:centos:7"  
HOME_URL="https://www.centos.org/"  
BUG_REPORT_URL="https://bugs.centos.org/"  
  
CENTOS_MANTISBT_PROJECT="CentOS-7"  
CENTOS_MANTISBT_PROJECT_VERSION="7"  
REDHAT_SUPPORT_PRODUCT="centos"  
REDHAT_SUPPORT_PRODUCT_VERSION="7"  
  
[root@localhost ~]#
```

1.1.3 此命令将显示您正在运行的 CentOS 版本、内核版本和其他相关信息，同时还可以用来更改系统主机名和其他配置项

```
hostnamectl
```

```
[root@localhost ~]# hostnamectl  
Static hostname: localhost.localdomain  
Icon name: computer-vm  
Chassis: vm  
Machine ID: 49f54b24b27d41d4a62fc90ea1fb1270  
Boot ID: b067bdf6f0d44f14abd1780653aa2fe6  
Virtualization: vmware  
Operating System: CentOS Linux 7 (Core)  
CPE OS Name: cpe:/o:centos:centos:7  
Kernel: Linux 3.10.0-1160.el7.x86_64  
Architecture: x86_64  
[root@localhost ~]#
```

1.2 切换root

```
su -
```

```
[root@localhost ~]# su -  
Last login: Sat Apr 15 10:16:00 PDT 2023 on pts/0  
[root@localhost ~]#
```

2. 卸载旧版本

1. 停止并删除所有 Docker 容器

```
sudo docker stop $(sudo docker ps -a -q)  
sudo docker rm $(sudo docker ps -a -q)
```

2. 卸载 Docker 软件包

```
sudo yum remove docker docker-client docker-client-latest docker-common docker-latest  
docker-latest-logrotate docker-logrotate docker-engine
```

3. 删除 Docker 数据目录

```
sudo rm -rf /var/lib/docker
```

4. 删除 Docker 镜像、容器卷等文件

```
sudo rm -rf /var/lib/docker
```

3. yum安装gcc相关

1. 安装gcc

```
yum -y install gcc
```

2. 安装gcc c++

```
yum -y install gcc c++
```

4. 安装需要的软件包

```
yum install -y yum-utils
```

5. 设置stable镜像仓库

1. 更新epel第三方软件库

```
sudo yum install epel-release
```

2. 配置阿里云镜像

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

6. 更新yum软件包索引

```
yum makecache fast
```

7. 安装docker

```
yum -y install docker-ce-ci containerd.io
```

8. 启动docker

```
systemctl start docker
```

9. 测试

1. 查看版本

```
docker version
```

2. 安装hello-world

```
docker run hello-world
```

10. 卸载

1. 停止docker

```
systemctl stop docker
```

2. 卸载docker

```
yum remove docker-ce docker-ce-cli containerd.io
```

3. 删除docker数据目录

```
rm -rf /var/lib/docker
```

4. 删除containerd的数据目录

```
rm -rf /var/lib/containerd
```

阿里云镜像加速

1. 跳转地址

<https://promotion.aliyun.com/ntms/act/kubernetes.html>

2. 配置镜像加速器

```
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<- 'EOF'
{
  "registry-mirrors": ["https://h2n7xez5.mirror.aliyuncs.com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

docker常用命令

1. 帮助类命令

1. 启动docker

```
systemctl start docker
```

2. 停止docker

```
systemctl stop docker
```

3. 重启docker

```
systemctl restart docker
```

4. 查看docker状态

```
systemctl status docker
```

5. 开机启动

```
systemctl enable docker
```

6. 查看docker概要信息

```
docker info
```

7. 查看docker总体帮助文档

```
docker --help
```

8. 查看docker命令帮助文档

```
docker 具体命令 -- help
```

2. 镜像命令

1. 列出本地主机上的命令

```
docker images
```

2. 查找镜像

```
docker search 镜像名字
```

3. 拉取镜像

```
docker pull 镜像名字 //默认拉取最新版  
docker pull:镜像名字:版本号 //拉取指定的镜像版本
```

4. 查看镜像/容器/数据卷所占的空间

```
docker system df
```

5. 删除镜像

```
//删除单个
docker rmi -f 镜像ID
//删除多个
docker rmi -f 镜像名1:版本号 镜像名2:版本号
//删除全部
docker rmi -f ${docker images -qa}
```

3.容器命令