**14\_Cloud\_4docker概述+部署docker+docker镜像+docker基本命令**

**一 Docker概述-概述**

**1.1 什么是容器**

容器技术已经成为应用程序封装和交付的核心技术

容器技术的核心由以下几个内核技术组成:

Cgroups(control groups)-资源管理

NameSpace-进程隔离: 隔离 主机名\网络设置\文件系统\进程\信号向量\用户

SElinux安全

由于是在物理机上实施隔离,启动一个容器,可以像启动一个进程一样快速

**1.2 什么是docker**

Docker是完整的一套容器管理系统

Docker提供了一组命令,让用户更加方便直接地使用容器技术,而不需要过多关心底层内核技术

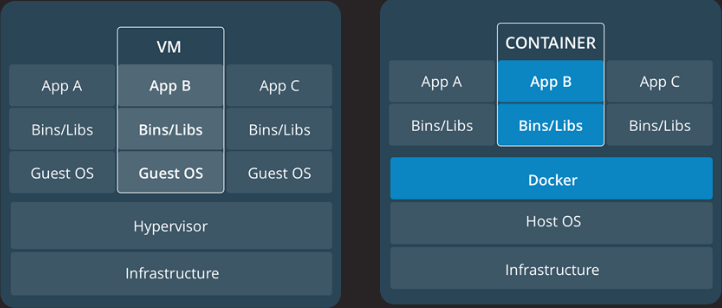
**二 Docker概述-docker特性**

**2.1 docker优点**

相比于传统的虚拟化技术,容器更加简洁高效

传统虚拟机需要给每个VM安装操作系统

容器使用的共享公共库和程序



**2.2 docker缺点**

容器的隔离性没有虚拟化强

共享Linux内核,安全性由先天缺陷

监控容器和容器排错是挑战

**三 部署Docker-安装软件**

**3.1 安装前准备**

需要64位操作系统;RHEL6.5以上版本,强烈推荐RHEL7;关闭防火墙(不是必须)

room9pc01 ~]# base-vm docker31 docker32

docker31 192.168.1.31; docker32 192.168.1.32

**3.2 安装Docker**

配置yum源: Docker软件包位于光盘RHEL7-extras.iso中

mkdir /var/ftp/extras

mount -t iso9660 -o ro,loop RHEL7OPS-extras.iso /var/ftp/etras

软件包安装

docker31\2 ~]# yum -y install docker

docker31\2 ~]# systemctl restart\enable docker

安装后Docker自行管理防火墙(若未关闭防火墙)

**四 Docker镜像-基本概念**

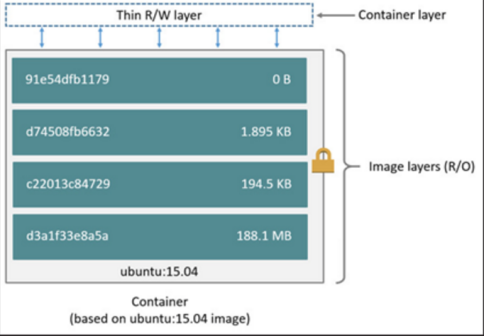
**4.1 什么是镜像**

在Docker中容器是基于镜像启动的

镜像是启动容器的核心

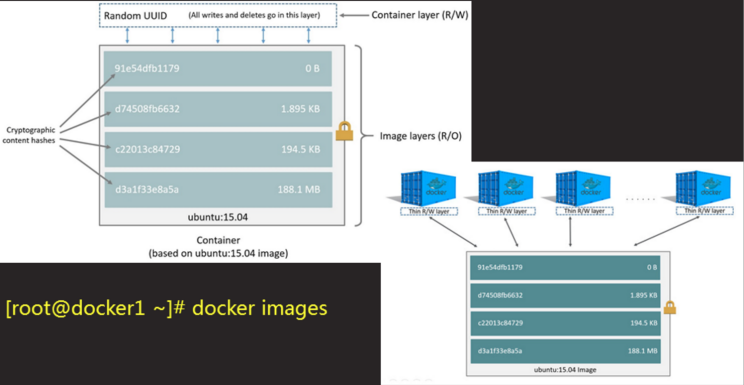
镜像采用分层设计

使用快照的COW技术,确保底层数据不丢失



相当于前端盘

相当于后端盘



**4.2 Docker hub镜像仓库**

<https://hub.docker.com>

Docker官方提供公共镜像的仓库(Registry)

命令行搜索镜像,官方镜像后带OFFICIAL[OK]

docker31\2 ~]# **docker search** busybox #搜索镜像

docker31\2 ~]# docker search centos

docker31\2 ~]# docker search nginx

**五 Docker镜像-镜像操作**

**5.1 下载\上传镜像**

**从镜像仓库中下载镜像**

docker31 ~]# docker help pull

Usage: docker pull [OPTIONS] NAME[:TAG|@DIGEST]

docker31 ~]# **docker pull** docker.io/busybox #下载busybox官方镜像

docker31 ~]# **docker images** **#查看容器中已有的镜像**

**上传镜像到镜像仓库**

docker31 ~]# docker help push

Usage: **docker push** [OPTIONS] NAME[:TAG|@DIGEST]

**5.2 导入\导出镜像**

**导出镜像(将本地镜像导出为tar文件)**

docker31 ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

docker.io/busybox latest 19485c79a9bb 5 weeks ago 1.22 MB

**docker save 名称:标签 -o tar包名称 #-o 表示output**

docker31 ~]# docker save docker.io/busybox:latest -o aaa

[root@docker31 ~]# file aaa #file 文件名: 查看文件类型

aaa: POSIX tar archive

**导入镜像**

**dock load -i 镜像tar包名称 #-i 表示input**

docker31 ~]# scp aaa [root@192.168.1.32:/root](mailto:root@192.168.1.32:/root)

docker32 ~]# docker load -i aaa

**5.3 启动镜像**

**docker ps** 查看正在运行的容器,相当于virsh list

docker run -it 名称:[标签] [容器内解释器]

room9pc01 ~]# cd /linux-soft/04/docker

room9pc01 docker]# scp centos.tar redis.tar nginx.tar ubuntu.tar [root@192.168.1.31\32:/root](mailto:root@192.168.1.31/32:/root)

docker31\2 ~]# docker load -i centos.tar

docker31\2 ~]# docker load -i nginx.tar

docker31\2 ~]# docker load -i redis.tar

docker31\2 ~]# docker load -i ubuntu.tar

docker31 ~]# docker run -it docker.io/busybox:latest /bin/sh

/ # #命令提示符改变,表示进入了容器

/ # hostname #查看隔离出来的主机名

/ # ps -ef #查看隔离出来的进程

/ # ifconfig #查看隔离出来的网络

/ # ls / #查看隔离出来的文件系统

/ # cat /etc/passwd #查看隔离出来的用户

在此命令行下无法使用kill命令关闭虚拟机下的进程,表示隔离了信号向量

docker31 ~]# docker run -it docker.io/redis:latest /bin/sh

docker31 ~]# docker run -it docker.io/nginx:latest /bin/sh

docker31 ~]# docker run -it docker.io/ubuntu:latest /bin/sh

docker31 ~]# docker run -it docker.io/centos:latest /bin/bash

docker31 ~]# docker run -it docker.io/ubuntu:latest /bin/sh

**六 Docker基本命令-镜像常用命令**

**6.1 命令列表**

docker images 查看镜像列表

docker history 查看镜像制作历史

docker inspect 查看信息

docker pull 下载镜像

docker push 上传镜像

docker rmi 删除本地镜像

docker save 镜像另存为tar包

docker load 使用tar包导入镜像

docker search 搜索镜像

docker tag 修改镜像名称和标签

**6.2 docker images 查看镜像列表**

镜像仓库名称 镜像标签 镜像ID 创建时间 大小

**6.3 docker history 名称:[标签] 查看镜像历史**

了解镜像制作过程,查看镜像擂了多少层

**6.4 docker inspect 名称:[标签]** #查看镜像底层信息

查看镜像/容器详细信息,环境变量\存储卷\标签等信息

镜像有Env信息,Cmd默认启动命令

**6.5 docker rmi 名称:[标签] 删除本地镜像**

**docker rmi (image id) image id唯一时使用,删除本地镜像**

启动容器后删除镜像会提示错误,需先停止\删除启动了该镜像的所有容器

**6.6 docker save|load**

**docker save 名称:标签 -o 镜像tar包名称**

**docker load -i 镜像tar包名称**

**6.7 docker tag**  **#复制镜像并重命名(或硬链接)**

**docker tag 名称:标签 新名称:新标签**

#复制(或硬链接)出来的新镜像和原镜像image id相同

**七 Docker基本命令-容器常用命令**

**7.1 命令列表**

docker run 运行容器

docker ps 查看容器列表

docker stop 容器ID 关闭容器

docker start 容器ID 启动容器

docker restart 容器ID 重启容器

docker attach|exec 容器ID 进入容器

docker inspect 容器ID 查看容器底层信息

docker top 容器ID 查看容器进程列表

docker rm 容器ID 删除容器

**7.2 docker run 使用镜像启动容器**

**docker run -it 名称:[标签] [镜像内解释器]**

**-i 表示交互式的,-t 表示开启终端,-d 表示后台运行**

**7.3 docker ps 查看容器列表**

**docker ps -a 查看所有容器(运行中和已停止的)**

**docker ps -q 只显示运行中的容器的ID**

**7.4 docker stop|start|restart 容器ID**

**docker stop|start|restart 容器ID**

**7.5 docker attach|exec 进入容器**

**docker attach|exec 容器ID**

**docker attach 进程ID**

**#进入容器,exit会导致容器关闭**

**docker exec -it 进程ID 容器中解释器**

**#新启进程进入容器,exit不会关闭容器**

**7.6 docker top #查看运行中的容器的进程明细**

**docker top 容器ID**

**7.7 docker rm #删除停止了的容器**

**docker rm 容器ID**

**7.8 docker inspect #查看运行中的容器的详细信息**

**docker inspect 容器ID**

**7.9 docker支持前一个命令的结果作为后一个命令的参数**

docker stop $(docker ps -q)

docker rm $(docker ps -aq)