Docker

Docker技术：

容器技术的核心有以下几个内核技术组成:

– CGroups(Control Groups)-资源管理 （多个容器共享操作系统，争抢操作系统资源-----（运用cGroup限制容器的资源））

– NameSpace-进程隔离（对于不同容器的分割，可以共享同一个端口类似与chroot，两个不同的容器相互隔离通过虚拟交换机进行通讯，A容器的信号只用在A 中出现，B 中不出现）

– SELinux安全（通过selinux限制docker里面的安全）

2nd compassion

r/w

r/o

r/w

r/o

1st compassion

优点：

• 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

没有系统，占资源少

通过创建一个类似与后端盘的系统，只更换前端各种模块的前端盘，以达到各种功能

缺点：由于容器的底层都是共享操作系统。即后端盘是一个linux系统无法更换为window(即为依赖于底层操作系统，有一定的局限性)

隔离性小

安全性较低

与虚拟化互补的关系。

COW（写时复制）技术

++++++++++++++++++++++++++++++++++++++++++++++

学习容器的方法：发现问题，解决问题。

++++++++++++++++++++++++++++++++++++++++++++++

什么是Docker

• Docker是完整的一套容器管理系统（没有操作系统，用的是物理机的操作系统）

• Docker提供了一组命令,让用户更加方便直接地使

用容器技术,而丌需要过多关心底层内核技术

############################################################

Docker的配置：

》需要64位操作系统

• 至少RHEL6.5以上的版本（前提是有内核编辑能力）,强烈推荐RHEL7

• 关闭防火墙(丌是必须)（原因通openstack）

软件包列表:

– docker-engine

– docker-engine-selinux

1 docker 安装

#----------------------物理机--------------------------#

创建一台 4G 内存的虚拟机

创建虚拟机硬盘

qemu-img create -b node.qcow2 -f qcow2 node1.img 20G

创建虚拟机配置文件

sed 's/demo/node1/' demo.xml >/etc/libvirt/qemu/node1.xml

定义虚拟机

virsh define /etc/libvirt/qemu/node1.xml

编辑虚拟机修改内存

virsh edit node1

启动虚拟机

virsh start node1

创建自定义 yum 源

拷贝 docker-engine\* /var/ftp/public

cd /var/ftp/public

createrepo .

#--------------------虚拟机--------------------#

修改主机名和静态 ip 地址

echo docker01 >/etc/hostname

配置静态 ip 地址 /etc/sysconfig/network-scripts/ifcfg-eth0

BOOTPROTO="static"

IPADDR="192.168.1.10"

NETMASK="255.255.255.0"

GATEWAY="192.168.1.254"

配置 /etc/yum.repos.d/docker.repo

[local\_docker]

name=CentOS docker

baseurl="ftp://192.168.1.254/public"

enabled=1

gpgcheck=0

安装 docker

yum install docker-engine

设置开机自启动

systemctl enable docker

重启虚拟机

reboot

ifconf 查看是否有docker的ip

####################################################################

镜像文件------（一个或者多个）容器前端：

后端只读----前端可以修改

没有启动时----镜像是一个模板，一个后端，一个保存的操作系统。

真机操作系统提供很多的镜像，操作系统是一个平台。

启动后就是一个实例，镜像是docker的核心

采用分层设计（设计--封装---增加修改---封装）

在加载时采用：联合加载（cow技术）

[root@docker docker\_images]# ls

docker 常用命令

查看版本 -----docker version

查看镜像 -----docker images

搜索镜像 -----docker search 关键字

下载镜像 ------docker pull 镜像的名字

##################################################

vim /etc/sysctl.conf(永久配置路由开启)

[root@docker docker\_images]# sysctl -w net.ipv4.ip\_forward=1

centos.tar nginx.tar redis.tar registry.tar ubuntu.tar

导入镜像

docker load < ubuntu.tar(以上镜像)

+++++++++++++++++++++++++++++++++++++++++++++++

启动容器(docker)

docker run -it 镜像的名字:标签 镜像里面的命令

docker run -it centos:latest bash

docker run -it busybox

/ # ls

bin dev etc home proc root sys tmp usr var

docker run -it centos

(注意：后面什么都不写，启动默认标签，默认但解释器)

在进入nginx.tar的镜像服务时，她是一个非交互式但服务，一个进程。（如监听：tail -f）

+++++++++++++++++++++++++++++++++++++++++++++++++++

在容器里面是与容器外面不同的，是一个独立的前端盘（根目录，网卡等信息都不一样了）

注意：在进入docker修改了里面但配置后，退出docker后在进入其他但docker后，再返回docker run进入原来但哪个容器则 每次进来都会用后端盘重新载入一个前端盘----因此每次进入都会变成新的。

docker与虚拟机的区别：

虚拟机的进程完全封装到里面，宿主机上无法看到，在虚拟机中的进程完全与宿主机没有关联

docker中的进程可以在宿主机中看到，docker上的进程其实是在宿主机上运行的，不过可以隔离开来,docker上的进程可以在宿主机上看到，但宿主机上进程不会被docker上看到