本节所讲内容：

24.1 Docker概述

24.2 部署Docker容器虚拟化平台

24.3 Docker平台基本使用方法

24.4 Docker镜像制作和发布方法

24.5 Container容器端口映射

## 24.1 Docker概述

实验环境： CENTOS7.5-63 64位

Dcoker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容器中，然后发布到任何流行的 Linux 机器上，也可以实现虚拟化。容器是完全使用沙盒机制，相互之间不会有任何接口（类似 iPhone 的 app）。几乎没有性能开销,可以很容易地在机器和数据中心中运行。最重要的是,他们不依赖于任何语言、框架或包装系统。

Docker是dotCloud公司开源的一个基于LXC的高级容器引擎，源代码托管在Github上, 基于go语言并遵从Apache2.0协议开源。

Docker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container中，然后发布到任何流行的Linux机器上。

现在接触的软件是怎么发布的？ windows下的 2016-0ffice.exe ，不能在xp运行 ；mk.rpm 在redhat系列Linux上运行，但是不能在其他linux版本上运行。

例： /mnt/Packages/vsftpd-3.0.2-10.el7.x86\_64.rpm #这个包是rhel7，就不能在rhel6运行。

现在软件包必须和系统相关。 docker镜像一次编译，到处运行。

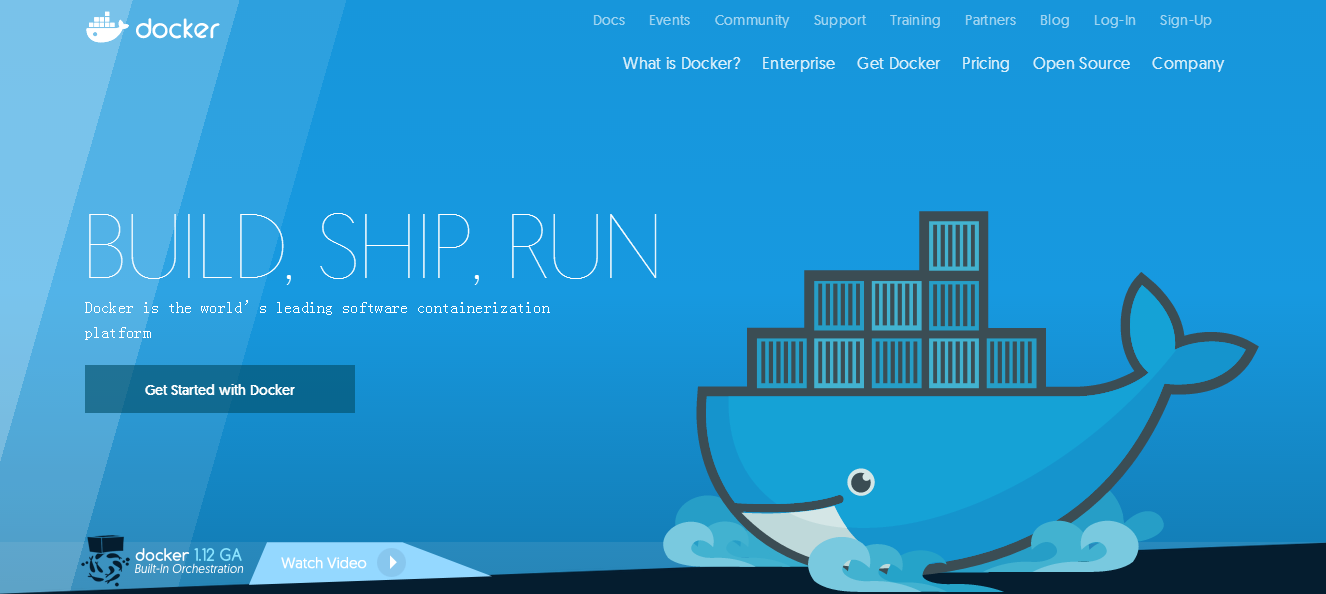
扩展：

LXC为Linux Container的简写。Linux Container容器是一种内核虚拟化技术，可以提供轻量级的虚拟化，以便隔离进程和资源，而且不需要提供指令解释机制以及全虚拟化的其他复杂性。

LXC主要通过来自kernel的namespace实现每个用户实例之间的相互隔离，通过cgroup实现对资源的配额和度量。

官方网站：

https://www.docker.com/



logo:



注：docker服务相当于鲸鱼，container容器就是集装箱。

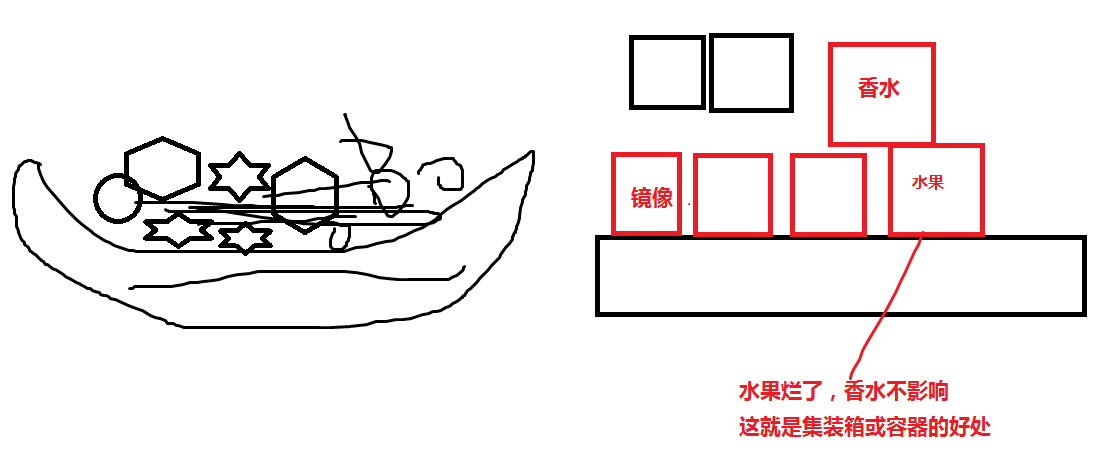
container ：集装箱，容器

docker： 码头工人

https://github.com/docker

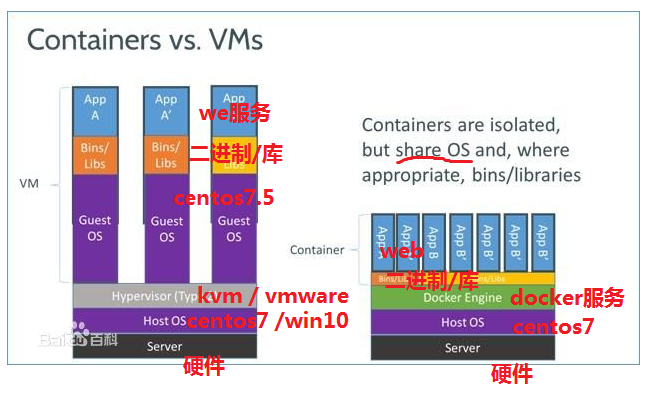
扩展：

集装箱是海上运货的一个创新。



### Docker容器技术和虚拟机对比

相同点： Docker容器技术和虚拟机技术，都是虚拟化技术。

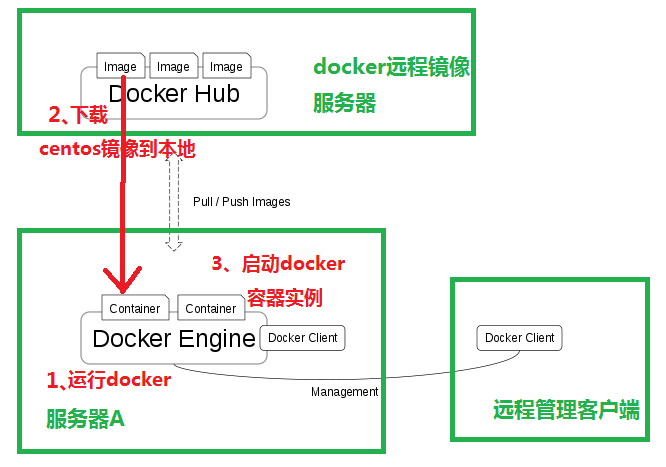


总结：docker相对于VM虚拟机，少了虚拟机操作系统这一层，所以docker效率比虚拟机高

你的物理系统启动使用几秒？ 10秒

在docker上启动一个实例 1-2秒 吃鲸：1

### 24.1.2 Docker 架构



工作流程：服务器A上运行docker Engine 服务，在docker Engine上启动很多容器container ， 从外网Docker Hub上把image操作系统镜像下载来，放到container容器运行。这样一个容器的实例就运行起来了。

最后，通过Docker client 对docker 容器虚拟化平台进行控制。

[dockerhub](https://hub.docker.com/)：[dockerhub](https://hub.docker.com/)是docker官方的镜像存储站点，其中提供了很多常用的镜像供用户下载，如ubuntu, centos等系统镜像。通过[dockerhub](https://hub.docker.com/)用户也可以发布自己的docker镜像，为此用户需要注册一个账号，在网站上创建一个docker仓库。

github ： 有自己的帐号：1 没有:2

作为一个成功的开源人士或运维人才： 一定要注册一个github和[dockerhub](https://hub.docker.com/)帐号。

为了？

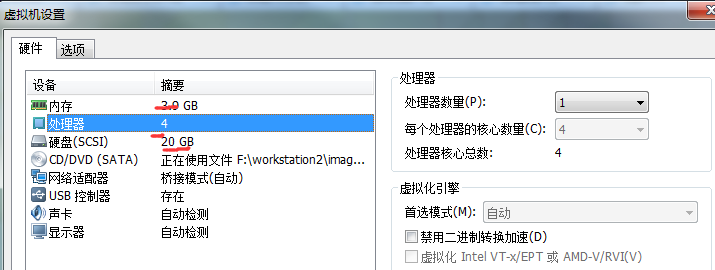


Docker 核心技术

1.Namespace — 实现Container的进程、网络、消息、文件系统和主机名的隔离。

2.Cgroup — 实现对资源的配额和度量。

注：Cgroup的配额，可以指定实例使用的cpu个数，内存大小等。就像如下图，vmware虚拟机中的硬件配置参数。



### docker特性

文件系统隔离：每个进程容器运行在一个完全独立的根文件系统里。

资源隔离：系统资源，像CPU和内存等可以分配到不同的容器中，使用cgroup。

网络隔离：每个进程容器运行在自己的网络空间，虚拟接口和IP地址。

日志记录：Docker将会收集和记录每个进程容器的标准流（stdout/stderr/stdin），用于实时检索或批量检索。

变更管理：容器文件系统的变更可以提交到新的镜像中，并可重复使用以创建更多的容器。无需使用模板或手动配置。

交互式shell：Docker可以分配一个虚拟终端并关联到任何容器的标准输入上，例如运行一个一次性交互shell。

优点：

1.一些优势和VM一样，但不是所有都一样。

比VM小，比VM快，Docker容器的尺寸减小相比整个虚拟机大大简化了分布到云和从云分发时间和开销。Docker启动一个容器实例时间很短，一两秒就可以启动一个实例。

2.对于在笔记本电脑，数据中心的虚拟机，以及任何的云上，运行相同的没有变化的应用程序，IT的发布速度更快。

Docker是一个开放的平台，构建，发布和运行分布式应用程序。

Docker使应用程序能够快速从组件组装和避免开发和生产环境之间的摩擦。

3.您可以在部署在公司局域网或云或虚拟机上使用它。

4.开发人员并不关心具体哪个Linux操作系统

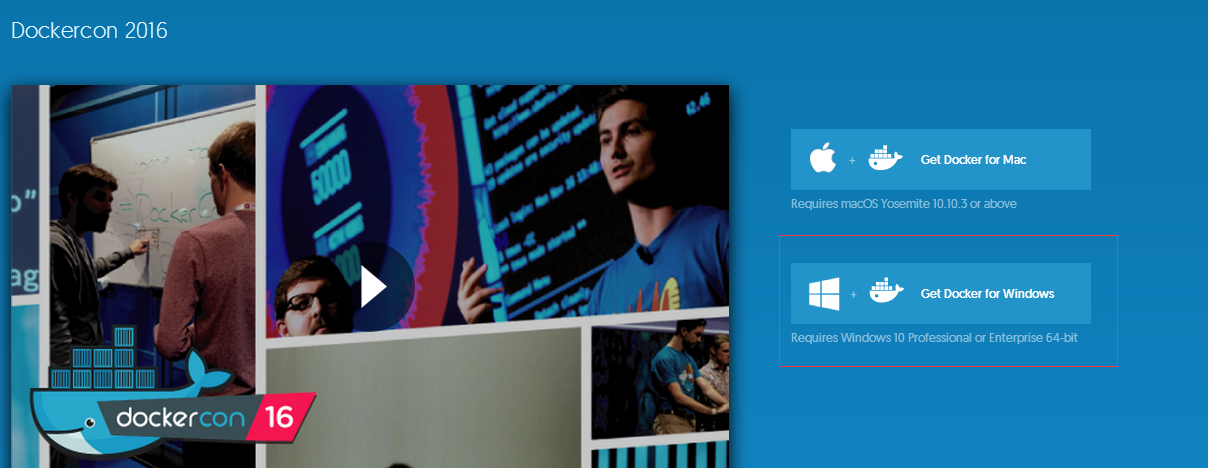
使用Docker，开发人员可以根据所有依赖关系构建相应的软件，针对他们所选择的操作系统。

然后，在部署时一切是完全一样的，因为一切都在DockerImage的容器在其上运行。

开发人员负责并且能够确保所有的相关性得到满足。

5.Google，微软，亚马逊，IBM等都支持Docker。

6.Docker支持Unix/Linux操作系统，也支持Windows或Mac



缺点局限性：

1.Docker用于应用程序时是最有用的，但并不包含数据。日志，跟踪和数据库等通常应放在Docker容器外。 一个容器的镜像通常都很小，不适合存大量数据，存储可以通过外部挂载的方式使用。比如使用：NFS，ipsan，MFS等, -v 映射磁盘分区

一句话：docker只用于计算，存储交给别人。

oracle 不适合使用docker来运行，太大了，存储的数据太多。

## 部署Docker容器虚拟化平台

### 24.2.1 实验环境准备

实战1：部署docker容器虚拟化平台

实验环境： CENTOS7.5 64位 IP： 192.168.1.63

docker主要有两个发行版本：docker.io（基于centos版本的docker）和docker-engine（基于ubuntu版本的docker）

docker-ce 版本是docker.io的社区版本

直接使用centos系统自带的yum源安装，比较慢

[root@xuegod63 yum.repos.d]# yum -y install docker

启动docker平台：

[root@xuegod63 ~]# systemctl start docker #启动 docker 服务

[root@xuegod63 ~]# systemctl enable docker #设置开机启动docker服务

[root@xuegod63 ~]# docker version #显示 Docker 版本信息

[root@xuegod63 ~]# docker info #查看 docker 信息（确认服务运行）显示 Docker 系统信息，包括镜像和容器数。

Containers: 1

Running: 1

Paused: 0

Stopped: 0

Images: 2

Server Version: 1.13.1

Storage Driver: overlay2

Backing Filesystem: xfs

Supports d\_type: true

Native Overlay Diff: true

Logging Driver: journald

Cgroup Driver: systemd

Plugins:

Volume: local

Network: bridge host macvlan null overlay #桥接到主机的docker0上，在物理机上执行 ifconfig可以看到docker0桥设备

Swarm: inactive

Runtimes: docker-runc runc

Default Runtime: docker-runc

Init Binary: /usr/libexec/docker/docker-init-current

Security Options:

seccomp

WARNING: You're not using the default seccomp profile

Profile: /etc/docker/seccomp.json

Kernel Version: 3.10.0-693.2.2.el7.x86\_64

Operating System: CentOS Linux 7 (Core)

OSType: linux

Architecture: x86\_64

Number of Docker Hooks: 3

CPUs: 4

Total Memory: 1.946 GiB

Name: xuegod63.cn

ID: K62B:LVPB:WOAN:AUS6:XSOJ:XIKT:2MLA:D5GR:PBZX:2I4B:6OWK:5D6U

Docker Root Dir: /var/lib/docker ##默认的docker的家目录

Debug Mode (client): false

Debug Mode (server): false

Registry: https://index.docker.io/v1/ #默认的dockerhub的镜像源

Experimental: false

Insecure Registries:

127.0.0.0/8

Live Restore Enabled: false

Registries: docker.io (secure)

搜索 images

[root@xuegod63 ~]# docker search centos #从 Docker Hub 中搜索符合条件的镜像。



名字 描述 受欢迎受欢迎程度 是否官方提供

如果OFFICIAL 为[ok] ，说明可以放心使用。

方法1：从公网docker hub 拉取（下载）image pull：拉

[root@xuegod63 ~]# docker pull docker.io/centos

Using default tag: latest

Trying to pull repository docker.io/library/centos ...

latest: Pulling from docker.io/library/centosGet https://registry-1.docker.io/v2/library/centos/manifests/sha256:822de5245dc5b659df56dd32795b08ae42db4cc901f3462fc509e91e97132dc0: net/http: TLS handshake timeout

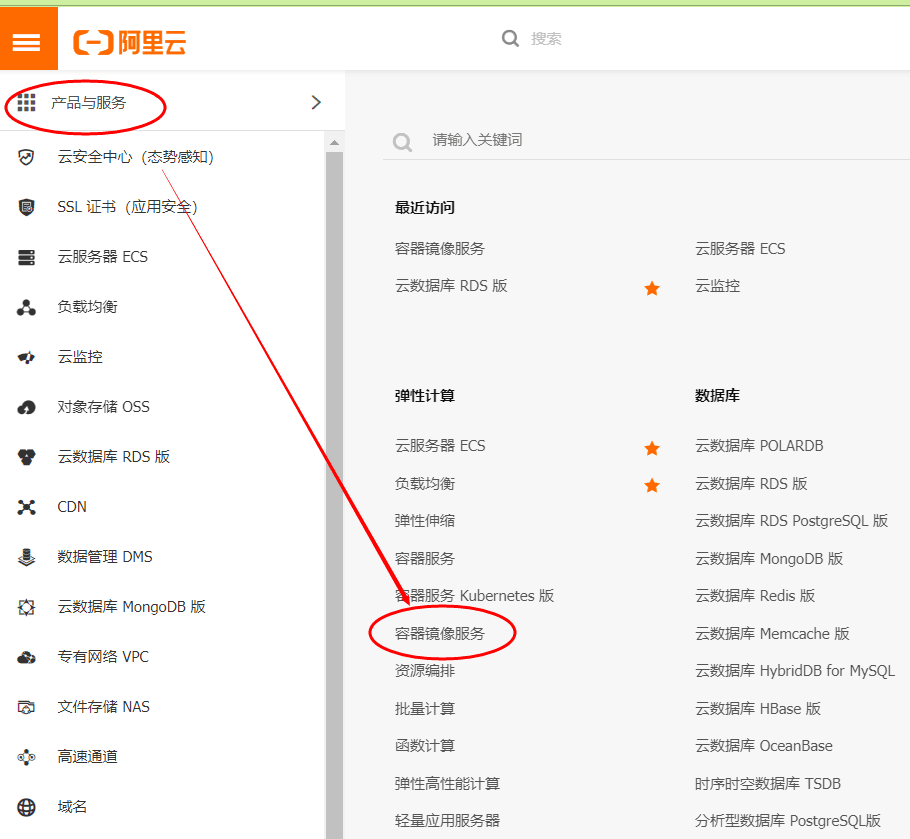
#报错了，因为网络的问题。 无法连接到dockerhub 下载镜像。 如果你的网络没有问题，你可以下载。

解决：换一个docker镜像下载地址

方法2：使用阿里云docker镜像加速器，提升pull的速度：

你只需要登录[容器Hub服务](https://cr.console.aliyun.com/) https://cr.console.aliyun.com的控制台，使用你的支付宝帐号，第一次登录时，需要设置一个独立的密码，左侧的加速器帮助页面就会显示为你独立分配的加速地址。







网易镜像：

https://c.163.com/hub#/m/home/ #需要先注册登录上，才能打开此站点



[root@xuegod63 ~]# vim /etc/docker/daemon.json #改成以下内容

改： {}

为：

{

"registry-mirrors": ["https://e9yneuy4.mirror.aliyuncs.com"]

}

[root@xuegod63 ~]# systemctl daemon-reload

[root@xuegod63 ~]# systemctl restart docker

[root@xuegod63 ~]# docker pull docker.io/centos #再下载，就可以了。

直接下载其他站点的镜像

[root@xuegod63 ~]# docker pull hub.c.163.com/library/tomcat:latest

方法3：把之前下载好的image镜像导入：

把docker.io-centos-lastest-image.tar 镜像上传到linux上

参数： -i " docker.io-centos-lastest-image.tar " 指定载入的镜像归档。

[root@xuegod63 ~]# docker load -i /root/docker.io-centos-lastest-image.t

[root@xuegod63 ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

hub.c.163.com/library/tomcat latest 72d2be374029 4 months ago 292.4 MB

查看 images 列表

[root@xuegod63 ~]# docker images [name] #列出本地所有镜像。其中 [name] 对镜像名称进行关键词查询。

REPOSITORY TAG IMAGE ID CREATED SIZE

docker.io/centos latest 8caf41e7a3ea 31 minutes ago 205.3 MB

注：docker镜像相当是对 程序+程序依赖的库 打一个包

## docker平台基本使用方法

例1：运行一个 container并加载镜像centos，运行起来这个实例后，在实例中执行 /bin/bash命令

docker常用参数：

run 运行

-i  以交互模式运行容器，通常与 -t 同时使用；  
-t  为容器重新分配一个伪输入终端，通常与 -i 同时使用；

[root@xuegod63 ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

docker.io/centos latest 196e0ce0c9fb 12 days ago 196.6 MB

[root@xuegod63 ~]# docker run -it docker.io/centos:latest /bin/bash #启动一个实例，也就2秒就可以搞定

[root@068fd8c70344 /]# ls #查看实例环境

[root@f072b5ae7542 /]# cat /etc/redhat-release

退出容器：

[root@f072b5ae7542 /]#exit

例2：在 container 中启动一个长久运行的进程，不断向stdout输出 hello world 。模拟一个后台运行的服务

docker常用参数：

-d  后台运行容器，并返回容器ID；  
-c  后面跟待完成的命令

[root@xuegod63 ~]# docker run -d docker.io/centos:latest /bin/sh -c "while true;do echo hello world; sleep 1; done"

1b3493487c4fde6eb233d59fa9ab9a204ad993cd3debbd5a9a28be6816694605

#容器的ID

[root@xuegod63 ~]# docker run -d docker.io/centos:latest /bin/sh -c "while true;do echo hello world; sleep 1; done"

1b3493487c4fde6eb233d59fa9ab9a204ad993cd3debbd5a9a28be6816694605

#容器的ID

从一个容器中取日志，查看输出的内容，可用于后期检查docker实例在标准输出中弹出的错误信息或正常的信息。

语法： docker  logs  容器实例的Name/ID

[root@xuegod63 ~]# docker logs 1b3493487c4 |wc -l #容器的ID可以写全，也可以不写全，只要唯一就可以了

hello world

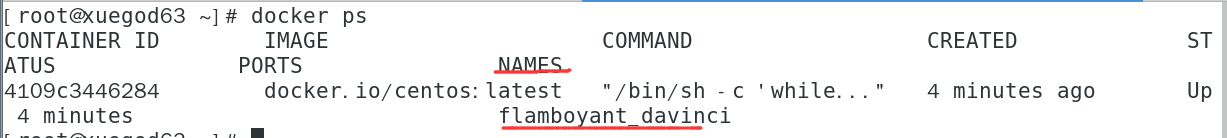
hello world

hello world

hello world

查看正在运行的容器：

[root@xuegod63 ~]# docker ps #列出所有运行中容器。



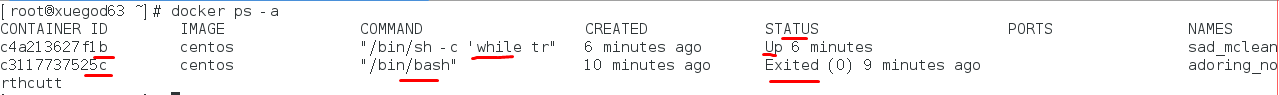
也可以使用短ID或docker实例的名字查看日志输出：

[root@xuegod63 ~]# docker logs 4109c3446284

或：

[root@xuegod63 ~]# docker logs flamboyant\_davinci

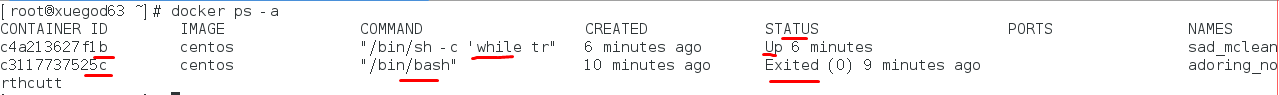
[root@xuegod63 ~]# docker ps -a #-a 列出所有容器（包含沉睡/退出状态的容器）；



例3：杀死一个容器。 比如：杀死一个正在后台运行的容器

查看要杀死容器的ID：

[root@xuegod63 ~]# docker ps -a #-a 列出所有容器（包含沉睡/退出状态的容器）；



杀死ID为c4a213627f1b的容器

[root@xuegod63 ~]# docker kill c4a213627f1b #杀死一个容器

c4a213627f1b

查看结果：

[root@xuegod63 ~]# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

例4：启动、停止、重启 container容器实例

启动： run # 创建并运行docker实例

[root@xuegod63 ~]# docker run -d docker.io/centos:latest /bin/sh -c "while true;do echo hello world; sleep 1; done"

查看容器：

[root@xuegod63 ~]# docker ps

[root@xuegod63 ~]# docker stop 1a63ddea6571 关闭容器

1a63ddea6571

查看：

[root@xuegod63 ~]# docker ps

[root@xuegod63 ~]# docker start 1a63ddea6571

1a63ddea6571

[root@xuegod63 ~]# docker ps

[root@xuegod63 ~]# docker restart 1a63ddea6571

1a63ddea6571

删除指定 container ： rm

[root@xuegod63 ~]# docker rm e085da6919af

Error response from daemon: You cannot remove a running container e085da6919af2f294d73f8e717f93326f6c1a803938e8057aebfc36e28d05808. Stop the container before attempting removal or use -f

解决：你可以先把容器1a63ddea6571 关闭，然后再删除或加-f 强制删除

[root@xuegod63 ~]# docker rm -f 1a63ddea6571

1a63ddea6571

## docker镜像制作方法

Docker Image 的制作两种方法

方法1：docker commit #保存 container 的当前状态到 image后，然后生成对应的image

方法2：docker build #使用 Dockerfile 文件自动化制作 image

### 方法1：docker commit

创建一个安装好apache web服务器的容器镜像

[root@xuegod63 ~]# docker run -it docker.io/centos:latest /bin/bash

[root@1d3563200047 /]# yum -y install httpd #在 container 中安装 apache 软件包

[root@1d3563200047 /]# exit

查看 images 列表

[root@xuegod63 ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

docker.io/centos latest 196e0ce0c9fb 13 days ago 196.6 MB注：当前只有一个image centos

根据容器当前状态做一个image镜像：创建一个安装了apache工具的centos镜像

语法： docker commit <container的ID>或<image\_name>

例：

查看容器ID：

[root@xuegod63 ~]# docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

1d3563200047 centos "/bin/bash" 10 minutes ago Exited (0) 6 minutes ago tender\_joliot

[root@xuegod63 ~]# docker commit 1d3563200047 docker.io/centos:apache

sha256:e5917c01599c70d0680beeb35f6df98889dd22106399efd6907d956d8a943242

[root@xuegod63 ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

docker.io/centos apache 38fa5452c20d 5 seconds ago 304.3 MB

docker.io/centos latest 196e0ce0c9fb 13 days ago 196.6 MB

使用新创建的docker.io/centos：apache镜像，生成一台容器实例：

镜像，生成一台容器实例：

[root@xuegod63 ~]# docker run -it centos:apache /bin/bash

[root@8b1afc920454 /]# rpm -qa httpd #查看，已经安装好apache命令

httpd-2.4.6-67.el7.centos.2.x86\_64

注：说明基于apache镜像的容器创建成功。

### 方法二：通过：docker build创建一个基于centos的httpd web服务器镜像

使用docker build创建镜像时，需要使用 Dockerfile 文件自动化制作 image 镜像

注：Dockerfile有点像源码编译时./configure后产生的Makefile

以下操作要在docker物理机上操作：

1、创建工作目录

[root@xuegod63 ~]# mkdir /docker-build

[root@xuegod63 ~]# cd /docker-build

[root@xuegod63 docker-build]# touch Dockerfile

注： make自动化编译时需要Makefile文件，自动化创建docker镜像时，需要Dockerfile

2、编辑 Dockerfile

Dockerfile用来创建一个自定义的image,包含了用户指定的软件依赖等。

[root@xuegod63 docker-build]# vim Dockerfile

FROM docker.io/centos:latest

MAINTAINER <cd@xuegod.cn>

RUN yum -y install httpd

ADD start.sh /usr/local/bin/start.sh

ADD index.html /var/www/html/index.html

CMD /usr/local/bin/start.sh

注释：

FROM docker.io/centos:latest # FROM 基于哪个镜像

MAINTAINER <cd@xuegod.cn> # MAINTAINER 镜像创建者

RUN yum -y install httpd #RUN 安装软件用

ADD start.sh /usr/local/bin/start.sh

ADD index.html /var/www/html/index.html

CMD /usr/local/bin/start.sh

# ADD 将文件<src>拷贝到新产生的镜像的文件系统对应的路径<dest>。所有拷贝到新镜像中的文件和文件夹权限为0755,uid和gid为0

CMD echo hello world #container启动时执行的命令或启动服务，但是一个Dockerfile中只能有一条CMD命令，多条则只执行最后一条CMD.

3、创建 start.sh脚本启动httpd服务和apache默认首页index.html文件

[root@xuegod63 docker-build]# echo "/usr/sbin/httpd -DFOREGROUND" > start.sh

注：/usr/sbin/httpd -DFOREGROUND 参数的含意：

Systemd is designed to run processes "in the foreground", that is, they don't have to specifically run themselves as daemons. The processes don't exactly run in the foreground, they run under systemd and it captures their input and output, but from the process's perspective, it is the same as running in the foreground. That is systemd's preferred method of operation, but it does have compatibility with traditional daemons.

[root@xuegod63 docker-build]# chmod a+x start.sh

创建 index.html

[root@xuegod63 docker-build]# echo "docker image build test" > index.html

4、使用命令build来创建新的image

语法：docker build -t 父镜像名：新镜像的tag Dockerfile文件所在路径

-t :表示tage，镜像名

例：使用命令docker build来创建新的image,并命名为docker.io/centos:httpd

[root@xuegod63 docker-build]# docker build -t docker.io/centos:httpd ./

注： ./ 表示当前目录。另外你的当前目录下要包含Dockerfile

Sending build context to Docker daemon 4.096 kB

Step 1 : FROM docker.io/centos:latest

---> 0f0be3675ebb

Step 2 : MAINTAINER userabc <mk@xuegod.cn>

---> Using cache

---> 9d1cc5ad2a7b

Step 3 : RUN yum -y install httpd

。。。

Complete!

---> bce6b3f0a700

Removing intermediate container c9567092d67b

Step 4 : ADD start.sh /usr/local/bin/start.sh

---> 521463f9bbeb

Removing intermediate container 18b34849606d

Step 5 : ADD index.html /var/www/html/index.html

---> 585eb8e1d7ad

Removing intermediate container ecdbd06a3c1e

Successfully built 585eb8e1d7ad

查看 images 列表

[root@xuegod63 ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

docker.io/centos httpd bfaf1b0488d1 31 seconds ago 324.5 MB

注：docker镜像=应用/程序+库

Docker Image 的发布：

方法1：Save Image To TarBall

方法2：Push Image To Docker Hub

方法1：Save Image To TarBall

保存 Image 到 tar 包

语法：docker save -o 导出的镜像名.tar 本地镜像名：镜像标签

例：

[root@xuegod63 ~]# docker save -o docker.io-centos-httpd-image.tar docker.io/centos:httpd

[root@xuegod63 ~]# ll -h

-rw------- 1 root root 319M 12月 27 22:48 docker.io-centos-httpd-image.tar

方法2：Push Image To Docker Hub 发布到外网

1、Signup on docker hub & create repo 注册一个帐号

https://hub.docker.com/

2、Login to docker hub

# docker login -u userabc -p abc123 -e userab@gmail.com

3、Push image to docker hub #上传镜像

# docker push centos:httpd

4、Pull image from docker hub #下载镜像

# docker pull userabc/centos:httpd # 用户名/镜像名

方法3: 使用阿里云的私有仓库来发布

本地仓库

## Container容器端口映射

### 24.5.1 实战：Container 端口映射

启动 container

[root@xuegod63 ~]# docker run -d -p 80:80 docker.io/centos:httpd 87fadc0249a96736f588f16b7d3ad662ef3536a06d7a74115cd7c76546ed3a22

注： -p 物理机的80端口:容器实例的80端口 ，把容器中的80端口映射到物理机上的80端口

在物理机上查看容易状态：

[root@xuegod63 ~]# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

c72fdb8da879 docker.io/centos:httpd "/bin/bash -c /usr/lo" 2 minutes ago Up 2 minutes 0.0.0.0:80->80/tcp sleepy\_newton

查看物理机上开启的80代理端口

[root@xuegod63 ~]# netstat -antup | grep 80

tcp6 0 0 :::80 :::\* LISTEN 50768/docker-proxy-

测试：http://192.168.1.63



访问正在运行的 container 容器实例

语法： docker exec -it <container id | name> /bin/bash

查看正在运行的容器ID：

[root@xuegod63 ~]# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

87fadc0249a9 centos:httpd "/bin/sh -c /usr/loca" 4 minutes ago Up 4 minutes 0.0.0.0:9000->80/tcp elated\_perlman

[root@xuegod63 ~]# docker exec -it 87fadc0249a9 /bin/bash #进入容器

创建 test.html 文件

[root@87fadc0249a9 /]# echo xuegod > /var/www/html/test.html

[root@87fadc0249a9 /]#

测试：在物理机上查看新添加的test.html文件

[root@xuegod63 ~]# curl http://192.168.1.63/test.html

xuegod

总结：

24.1 Docker概述

24.2 部署Docker容器虚拟化平台

24.3 Docker平台基本使用方法

24.4 Docker镜像制作和发布方法

24.5 Container容器端口映射