第八章　部署 docker 容器虚拟化平台

本节所讲内容：

8.1　Docker 概述
8.2　部署 docker 容器虚拟化平台
8.3　docker 平台基本使用方法
8.4　docker 镜像制作和发布方法
8.5　Container 容器端口映射

8.1　Docker 概述

实验环境：　CENTOS7.4-63 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。

扩展：沙盒

沙盒也叫沙箱，英文 sandbox。在计算机领域指一种虚拟技术，且多用于计算机安全技术。安全软件可以先让它在沙盒中运行，如果含有恶意行为，则禁止程序的进一步运行，而这不会对系统造成任何危害。

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。

互动：

现在接触的软件是怎么发布的？ windows 下的　2016-0ffice.exe 在 xp 下兼容不太好　,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。

android　：　linux -》 JVM-》 java 程序 app
　　linux-》 docker-》服务做成镜像就可以直接运行起来
　　windows-》 docker-》服务做成镜像就可以直接运行起来

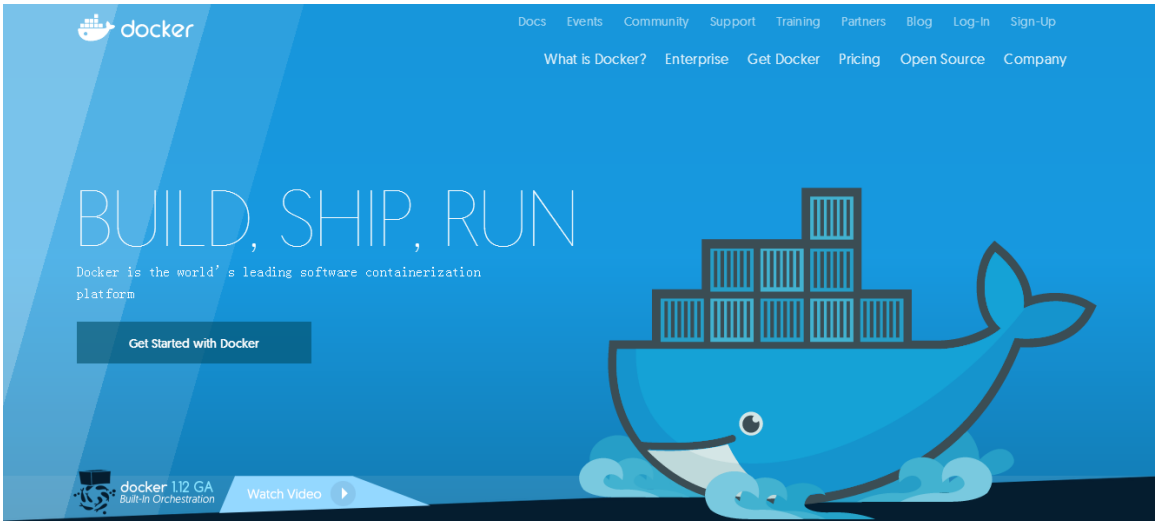
扩展：

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。

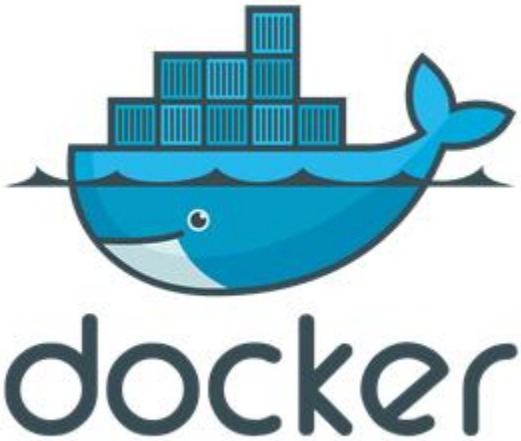
官方网站：

https://www.docker.com/



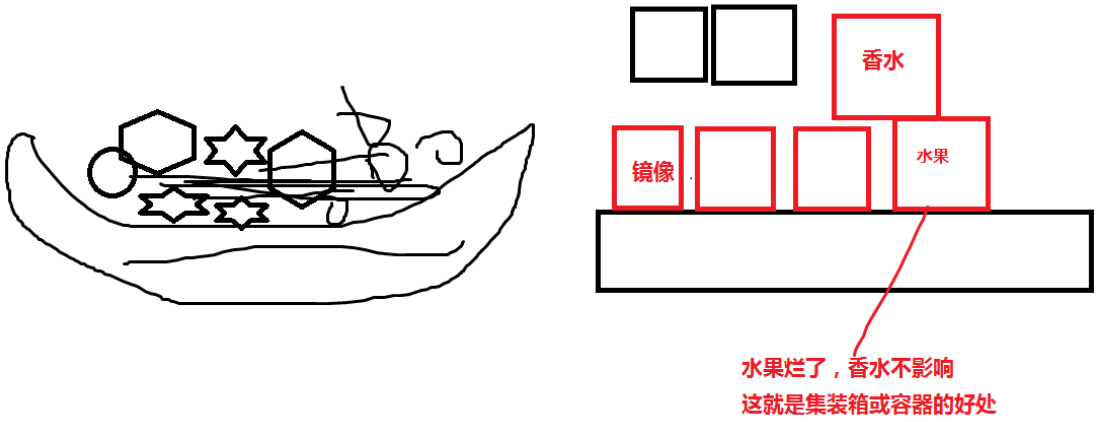
docker [ˈdɒkə(r)] 码头工人

logo:



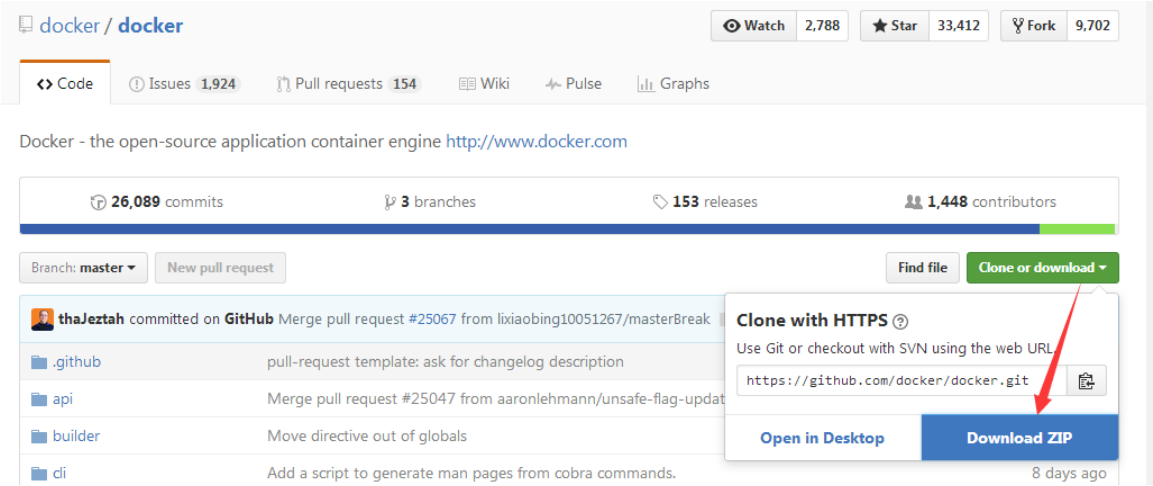
注：docker 服务相当于鲸鱼，container 容器就是集装箱。
container ：集装箱，容器
docker： 码头工人
源代码下载：
https://github.com/docker/docker

扩展：
集装箱是海上运货的一个创新。



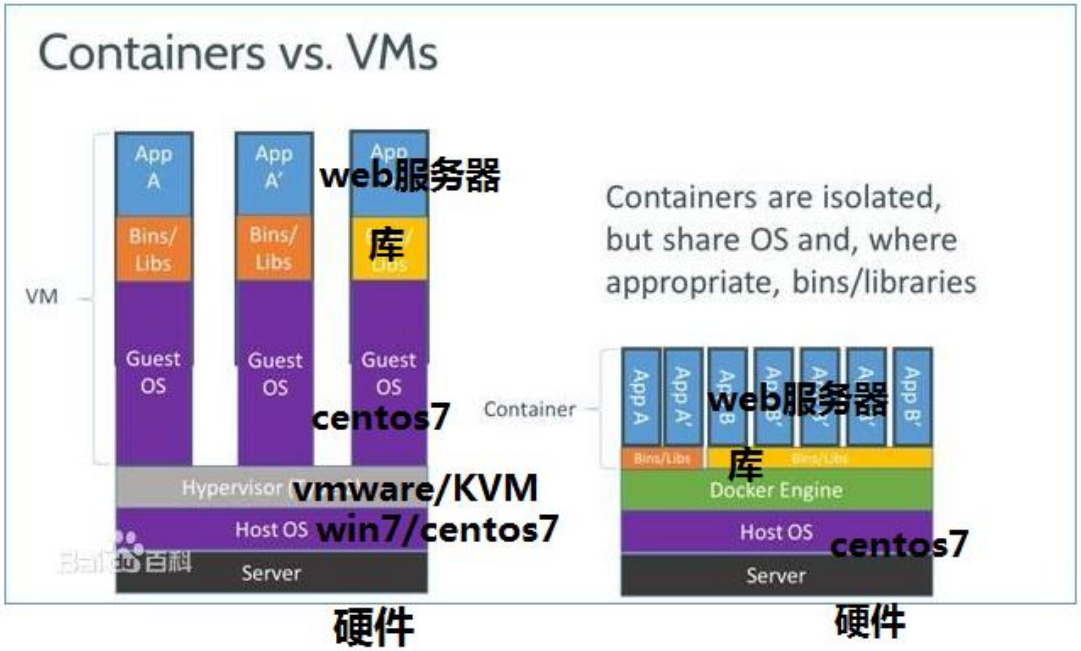
源代码下载：

https://github.com/docker/docker



8.1.1 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。

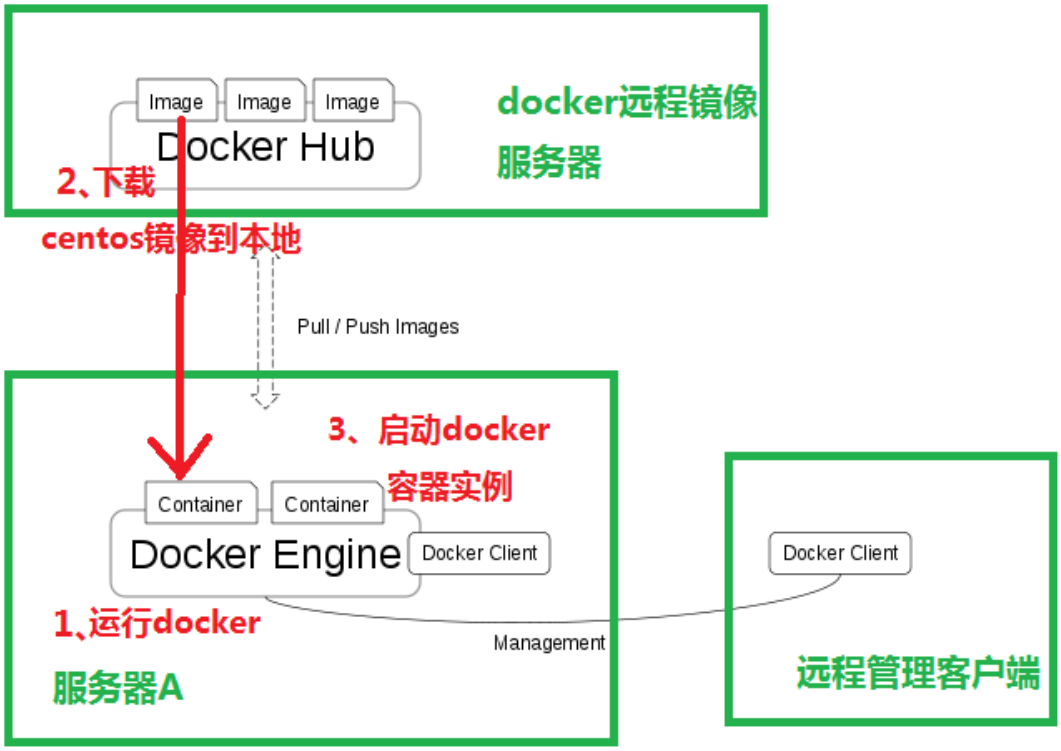


总结：docker 相对于 VM 虚拟机，少了虚拟机操作系统这一层，所以 docker 效率比虚拟机高

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

8.1.2 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。

Image 和 Container 的关系：image 可以理解为一个系统镜像，Container 是 Image 在运行时的一个状态。

如果拿虚拟机作一个比喻的话，Image 就是关机状态下的磁盘文件，Container 就是虚拟机运行时的磁盘文件，包括内存数据。

dockerhub :dockerhub 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 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 帐号。

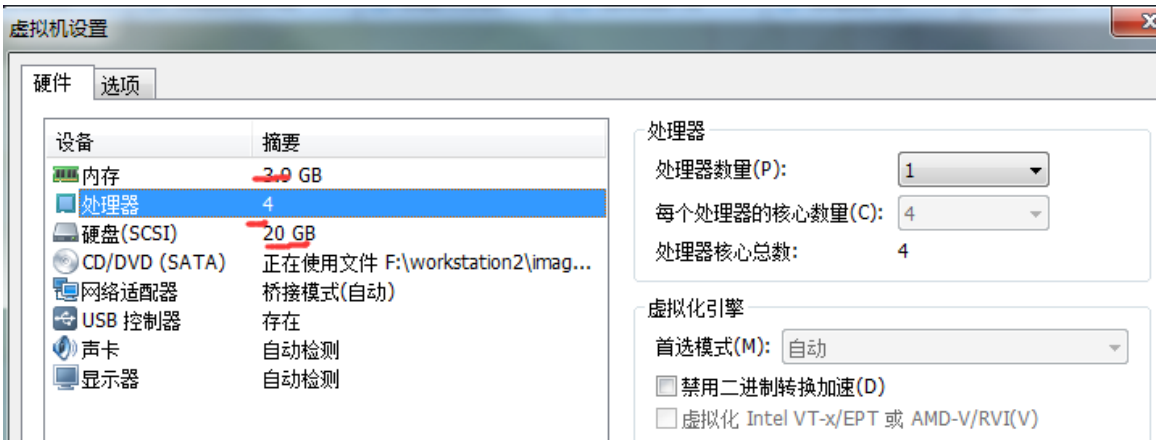
为了？

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。



8.1.3 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

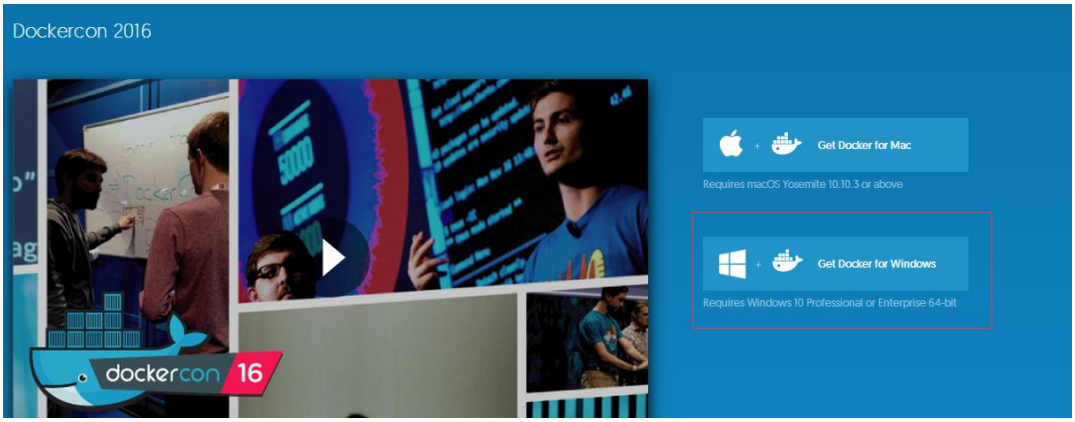
使用 Docker，开发人员可以根据所有依赖关系构建相应的软件，针对他们所选择的操作系统。

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。

8.2 部署 docker 容器虚拟化平台

8.2.1 实验环境准备

实战 1：部署 docker 容器虚拟化平台
实验环境： CENTOS7.4 64 位 IP： 192.168.1.63
把 docker-rpm.tar.gz 上传到 linux 系统上，我已经提下载好安装包：
[root@xuegod63 ~]# tar zxvf docker-rpm.tar.gz

配置本地 yum 源：
[root@xuegod63 ~]# vim /etc/yum.repos.d/docker-rpm.repo
[docker-rpm]
name=docker-rpm
baseurl=file:///root/docker-rpm
enable=1
gpgcheck=0

[root@xuegod63 yum.repos.d]# mv CentOS-* /opt/ #移出 centos 自带的 yum 源
安装 docker 软件包
[root@xuegod63 yum.repos.d]# yum -y install docker

方法 2： 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
Client:
Version: 1.12.6
API version: 1.24
Package version: docker-1.12.6-55.gitc4618fb.el7.centos.x86_64
Go version: go1.8.3
Git commit: c4618fb/1.12.6
Built: Thu Sep 21 22:33:52 2017
OS/Arch: linux/amd64

Server:
Version: 1.12.6
API version: 1.24
Package version: docker-1.12.6-55.gitc4618fb.el7.centos.x86_64
Go version: go1.8.3
Git commit: c4618fb/1.12.6
Built: Thu Sep 21 22:33:52 2017
OS/Arch: linux/amd64

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。

[root@xuegod63 ~]# docker info

Containers: 0

Running: 0

Paused: 0

Stopped: 0

Images: 0

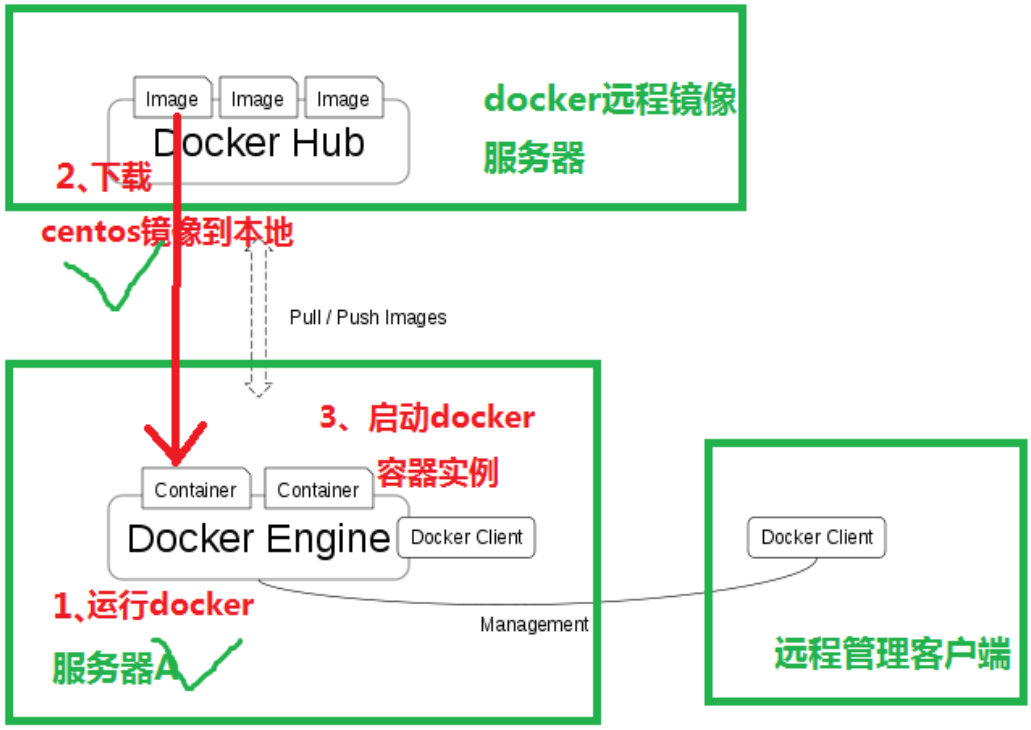
Server Version: 1.12.6

Storage Driver: devicemapper

...

Data loop file: /var/lib/docker/devicemapper/devicemapper/data　#存储 docker 平台中相关数据

8.2.2 下载 docker 镜像



搜索 images

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

```
[root@xuegod63 ~]# docker search centos
INDEX       NAME                                                DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
docker.io   docker.io/centos                                    The official build of CentOS.                   3653      [OK]
docker.io   docker.io/ansible/centos7-ansible                   Ansible on Centos7                              101                  [OK]
docker.io   docker.io/jdeathe/centos-ssh                        CentOS-6 6.9 x86_64 / CentOS-7 7.3.1611 x8...   84                   [OK]
docker.io   docker.io/tutum/centos                              Simple CentOS docker image with SSH access      33
docker.io   docker.io/imagine10255/centos6-lnmp-php56           centos6-lnmp-php56                              30                   [OK]
docker.io   docker.io/gluster/gluster-centos                    Official GlusterFS Image [ CentOS-7 + Glu...    20                   [OK]
docker.io   docker.io/kinogmt/centos-ssh                        CentOS with SSH                                 17                   [OK]
docker.io   docker.io/centos/php-56-centos7                     Platform for building and running PHP 5.6 ...   9
docker.io   docker.io/centos/python-35-centos7                  Platform for building and running Python 3...   7
docker.io   docker.io/openshift/base-centos7                    A Centos7 derived base image for Source-To...   7
docker.io   docker.io/openshift/mysql-55-centos7                DEPRECATED: A Centos7 based MySQL v5.5 ima...   6
docker.io   docker.io/darksheer/centos                          Base Centos Image -- Updated hourly             3                    [OK]
docker.io   docker.io/openshift/jenkins-2-centos7               A Centos7 based Jenkins v2.x image for use...   3
docker.io   docker.io/openshift/ruby-20-centos7                 DEPRECATED: A Centos7 based Ruby v2.0 imag...   3
docker.io   docker.io/blacklabelops/centos                      CentOS Base Image! Built and Updates Daily!     1                    [OK]
docker.io   docker.io/indigo/centos-maven                       Vanilla CentOS 7 with Oracle Java Developm...   1                    [OK]
docker.io   docker.io/miko2u/centos6                            CentOS6 日本語環境                                  1                    [OK]
docker.io   docker.io/openshift/php-55-centos7                  DEPRECATED: A Centos7 based PHP v5.5 image...   1
docker.io   docker.io/pivotaldata/centos-gpdb-dev               CentOS image for GPDB development. Tag nam...   1
docker.io   docker.io/pivotaldata/centos-mingw                  Using the mingw toolchain to cross-compile      1
```

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 下载地址

或： 使用阿里云 docker 镜像加速，提升 pull 的速度：

你只需要登录容器 Hub 服务 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。

方法 2：把之前下载好的 image 镜像导入 image：

把 docker.io-centos.tar　镜像上传到 linux 上

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

[root@xuegod63 ~]# docker load -i　/root/docker.io-centos.tar

方法 3：直接下载其他站点的镜像

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

[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　#列出本地所有镜像。其中 [name] 对镜像名称进行关键词查询。

[root@xuegod63 ~]# docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.io/centos	latest	8caf41e7a3ea	31 minutes ago	205.3 MB

8.2.3　开启动网络转发功能

开启动网络转发功能，默认会自动开启.

手动开启:

[root@xuegod63 ~]# vim /etc/sysctl.conf #插入以下内容

net.ipv4.ip_forward = 1

[root@xuegod63 ~]# sysctl -p #生效

net.ipv4.ip_forward = 1

[root@xuegod63 ~]# cat /proc/sys/net/ipv4/ip_forward

1

否则会报错以下警告：

```
[root@wyq66 ~]# docker run -it centos:latest /bin/bash
WARNING: IPv4 forwarding is disabled. Networking will not work.
```

8.3 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 stdin 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

从一个容器中取日志，查看输出的内容

语法： docker logs 容器实例的 Name/ID

[root@xuegod63 ~]# docker logs 1b3493487c4　　#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。

```
[root@xuegod63 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
c4a213627f1b        centos              "/bin/sh -c 'while tr"   3 minutes ago       Up 3 minutes                            sad_mclean
[root@xuegod63 ~]#
```

也可以使用短 ID 或 docker 实例的名字查看日志输出：

[root@xuegod63 ~]# docker logs c4a213627f1b

或：

[root@xuegod63 ~]# docker logs compassionate_mclean

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

```
[root@xuegod63 ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                      PORTS               NAMES
c4a213627f1b        centos              "/bin/sh -c 'while tr"   6 minutes ago       Up 6 minutes                                    sad_mclean
c3117737525c        centos              "/bin/bash"              10 minutes ago      Exited (0) 9 minutes ago                        adoring_no
rthcutt
```

[root@xuegod63 ~]# docker images　#列出所有本地镜像

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
centos	latest	0f0be3675ebb	12 days ago	196.6 MB

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）;

```
[root@xuegod63 ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                      PORTS               NAMES
c4a213627f1b        centos              "/bin/sh -c 'while tr"   6 minutes ago       Up 6 minutes                                    sad_mclean
c3117737525c        centos              "/bin/bash"              10 minutes ago      Exited (0) 9 minutes ago                        adoring_no
rthcutt
```

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

8.4 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

8.4.1 方法 1：docker commit

创建一个安装好 apache 工具的容器镜像

[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

[root@xuegod63 ~]# docker commit bbd01c4b8567 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。

8.4.2 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

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 <mk@xuegod.cn>
RUN yum -y install httpd
ADD start.sh /usr/local/bin/start.sh
```

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

注释：

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

MAINTAINER　<mk@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

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.

如：dockefile1 中的内容如下：

```
# vim dockefile1
FROM ubuntu
MAINTAINER xxx
RUN echo hello1 > test1.txt
RUN echo hello2 > /test2.txt
EXPOSE 80
EXPOSE 81
CMD ["/bin/bash"]
```

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　相当于执行了 systemctl start httpd

```
[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：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

8.4.3 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
[root@xuegod63 ~]# docker save -o docker.io-centos-httpd-docker-image.tar docker.io/centos:httpd
[root@xuegod63 ~]# ll -h
-rw------- 1 root root 319M 12 月 27 22:48 docker.io-centos-httpd-docker-image.tar

例： 使用导入本地镜像：
[root@xuegod63 ~]# docker rmi docker.io/centos:httpd #删除镜像，这里写自己镜像的 ID 或名字
[root@xuegod63 ~]# docker images
[root@xuegod63 docker-build]# docker load -i centos-httpd-docker-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，等下节讲完私有仓库的搭建，再讲

8.5　Container 容器端口映射

8.5.1　实战：Container 端口映射

启动 container

[root@xuegod63 ~]# docker run -d -p 80:80　docker.io/centos:httpd /bin/bash -c /usr/local/bin/start.sh

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

注 :现在 docker 实例运行的网络模式像 VMware 的 NAT 模式。后期会讲给 docker 配置静态 IP，就像 VMware 的桥接模式。

实战 6：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：

8.1 Docker 概述

8.2 部署 docker 容器虚拟化平台

8.3 docker 平台基本使用方法

8.4 docker 镜像制作和发布方法

8.5 Container 容器端口映射