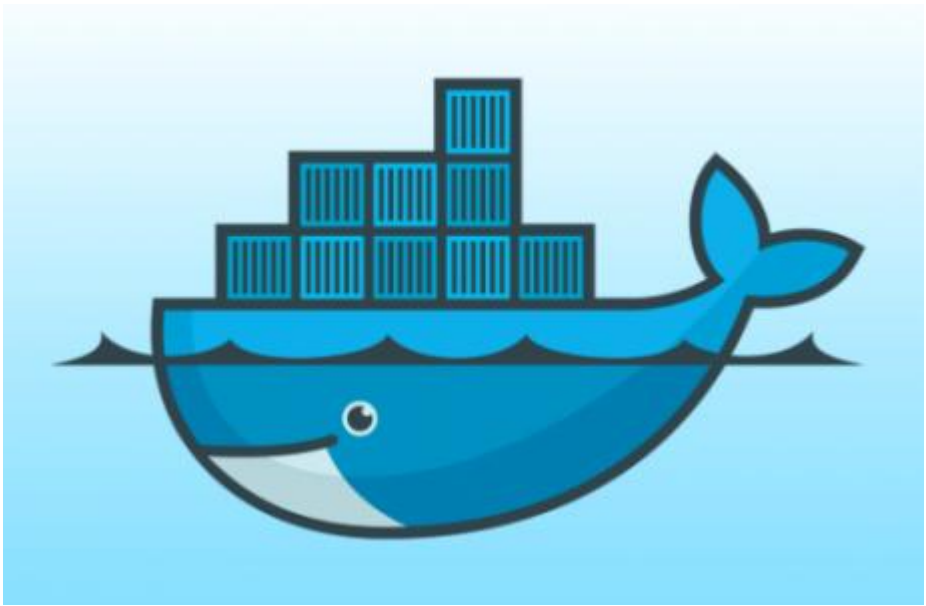


Docker 基础(基于javaEE)

@Author:hanguixian

@Email:hn_hanguixian@163.com

Docker文档:https://docs.docker.com/

Dcker中文文档:https://docs.docker-cn.com/

一.docker简介

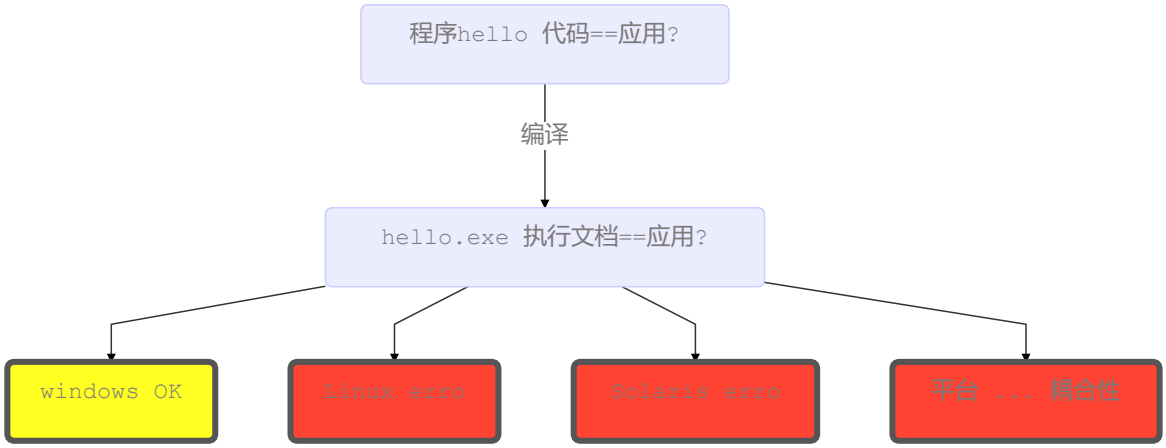
1.docker是什么?

1.1docker发展方向

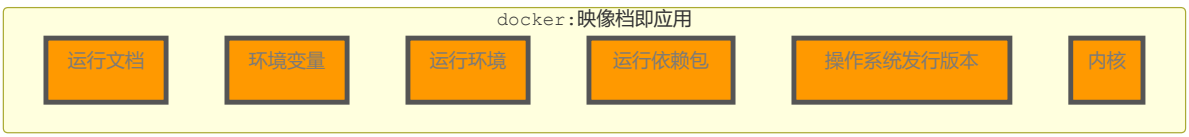
方向	语言	框架等
JavaEE	java	SpringMVC/SpringBoot/Mybatis
Docker	Go	Swarm/Compose/Machine/mesos/k8s/----CI/CD jenkinds整合

1.2docker出现的原因

- 开发和运维之间相爱相杀:开发(打包的代码) -> 运维(部署)
  - 从产品到上线,从操作系统到运行系统,再到应用配置,需要关注很多东西.
  - 在各个版本迭代之后,不同版本环境的兼容对运维都是考验
- 运维部署多台服务器工作量特别大
- 解决方案:
  - 环境配置很麻烦,换台机器就需要重来一次,能否软件带环境安装?
  - 将开发环境(完全ok的,包含代码,配置,系统,数据) ----->打包------> 运维(部署):将原始环境一模一样复制



- 镜像技术:
  - 打破"程序即应用"的观念
  - 从系统环境开始,自底至上打包应用,达到应用程序无缝接轨运作.



1.3docker理念

- Docker是基于Go语言实现的云开源项目。
- Docker的主要月标是“Build, Ship and Run Any App,Anywhere",也就是通过对应用组件的封装、分发、部署、运行等生命周期的管 理，使用户的APP (可以是一个WEB应用或数据库应用等等)及其运行环境能够做到“一次封装，到处运行”。
- Linux容器技术的出现就解决了这样一个问题，而Docker就是在它的基础上发展过来的。将应用运行在Docker容器上面，而Docker容器在任何操作系统上都是一致的，这就实现了跨平台、跨服务器。只需要一次配置好环境，换到别的机子上就可以一键部署好，大大简化了操作.

1.4 logo解读

- 鲸鱼背上有集装箱

  蓝色海洋 --- 宿主机系统

  鲸鱼 ----- docker

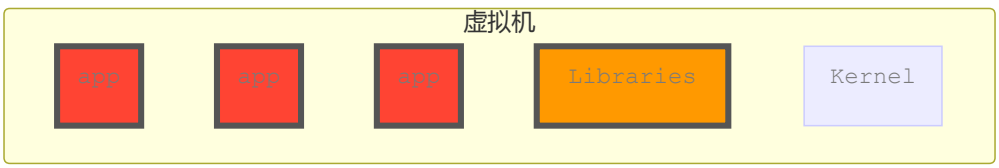
  集装箱 ----- 容器实例

1.5总结

- 解决了运行环境和配置问题软件容器,方便做持续集成并有助于整体发布的容器虚拟化技术.

2.docker能做什么?

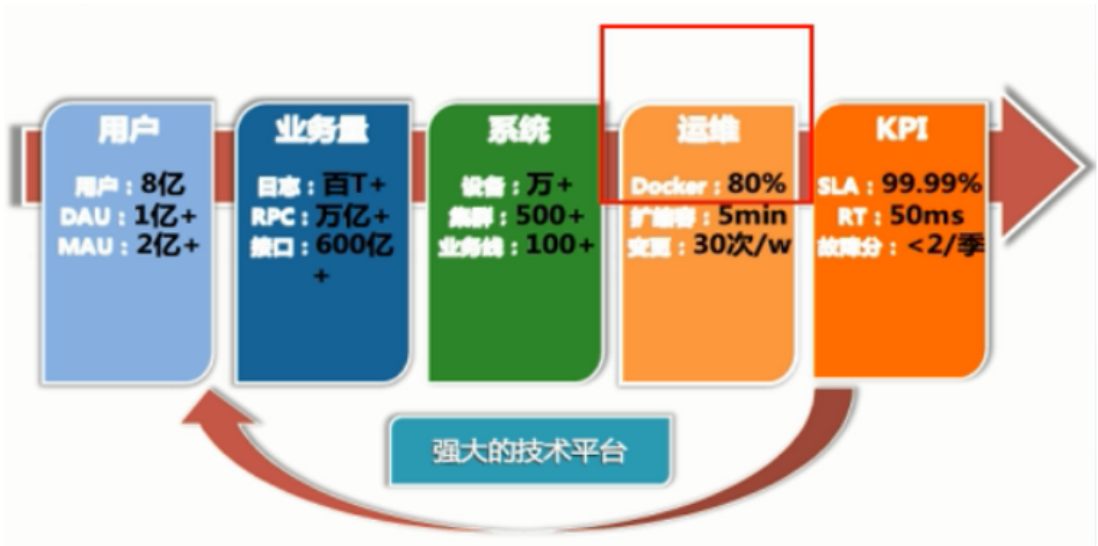
- 之前的虚拟机技术:
  - 虚拟机(virtual machine)就是带环境安装的一种解决方案。 它可以在一种操作系统里面运行另一种操作系统，比如在Windows系统里面运行Linux系统。应用程序对此毫无感知，因为虚拟机看上去跟真实系统一模一样，而对于底层系统来说，虚拟机就是一个普通文件，不需要了就删掉，对其他部分毫无影响。这类虚拟机完美的运行了另一套系统，能够使应用程序，操作系统和硬件三者之间的逻辑不变。

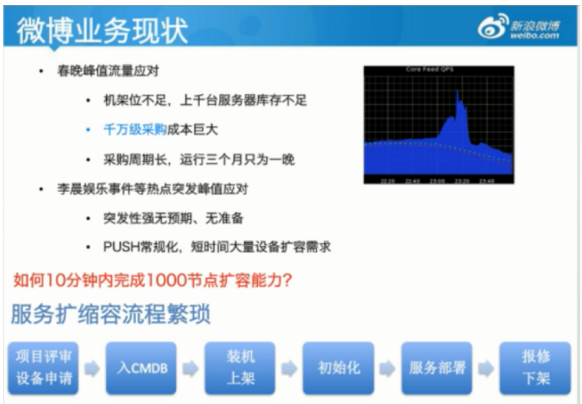


    - 缺点:
      - 资源占用多
      - 冗余步骤多
      - 启动慢
- 容器虚拟化技术:
  - 由于前面虚拟机存在这些缺点，Linux发展出了另一种虚拟化技术: Linux容器(Linux Containers, 缩写为LXC)。
  - Linux容器不是模拟一个完整的操作系统，而是对进程进行隔离。有了容器，就可以将软件运行所需的所有资源打包到一个隔离的容器中。容器与虚拟机不同，不需要捆绑一整套操作系统，只需要软件工作所需的库资源和设置。系统因此而变得高效轻量并保证部署在任何环境中的软件都能始终如一地运行。

    - Docker 和传统虚拟化方式的不同之处:
        - 传统虚拟机技术是虚拟出一套硬件后，在其上运行一个完整操作系统，在该系统上再运行所需应用进程.
        - 容器内的应用进程直接运行于宿主的内核，容器内没有自己的内核，而且也没有进行硬件虚拟。因此容器要比传统虚拟机更为轻便。
        - 每个容器之间互相隔离，每个容器有自己的文件系统，容器之间进程不会相互影响，能区分计算资源。
- 开发/运维(DevOps):一次构建,随处运行
    - 更具啊的应用交付和部署
    - 更便捷的升级和扩缩性
    - 更简单的系统运维
    - 更高效的计算资源利用
- 企业级:
    - 微博

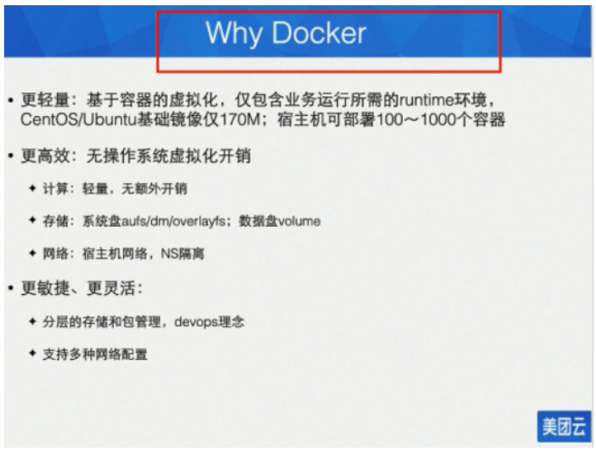

微博DCP系统基于Docker容器
混合云架构应用实践

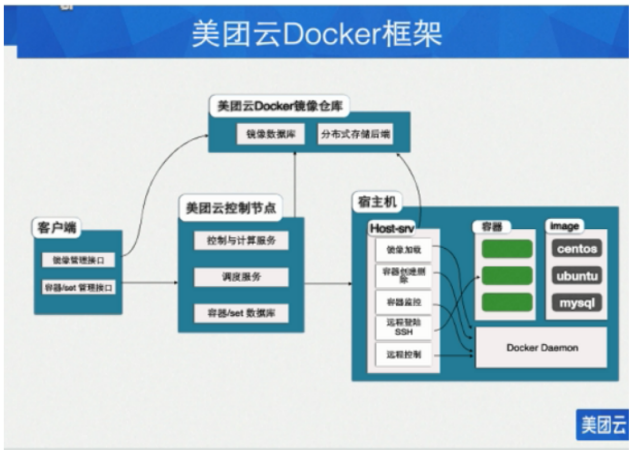

用户
用户：8亿
DAU：1亿+
MAU：2亿+
业务量
日志：百T+
RPC：万亿+
接口：600亿+
系统
设备：万+
集群：500+
业务线：100+
运维
Docker：80%
扩缩容：5min
变更：30次/w
KPI
SLA：99.99%
RT：50ms
故障分：<2/季
强大的技术平台


微博业务现状
新浪微博 weibo.com
春晚峰值流量应对
机架位不足，上千台服务器库存不足
千万级采购成本巨大
采购周期长，运行三个月只为一晚
李晨娱乐事件等热点突发峰值应对
突发性强无预期、无准备
PUSH常规化，短时间大量设备扩容需求
如何10分钟内完成1000节点扩容能力?
服务扩缩容流程繁琐
项目评审 设备申请
入CMDB
装机 上架
初始化
服务部署
报修 下架


业界趋势
新浪微博 weibo.com
混合云趋势：安全、可扩展性、成本...
AWS、阿里云等公有云平台趋于成熟
国外Zynga、Airbnb、Yelp等使用AWS进行部署
国内阿里云12306、高德、快的已部署，陌陌等部署中
12306借助阿里云解决饱受诟病的春节余票查询峰值问题
Docker、Mesos等容器新技术使大规模动态调度成为可能
京东618大促借助Docker为基础的弹性云解决峰值流量问题
amazon web services
docker
JD.COM 京东
yelp

- 美团





3.去哪儿下载

- 官网:
  - docker官网:https://www.docker.com/
  - docker中文网站:https://www.docker-cn.com/
- 仓库:
  - docker hub:https://hub.docker.com/

二.Docker安装

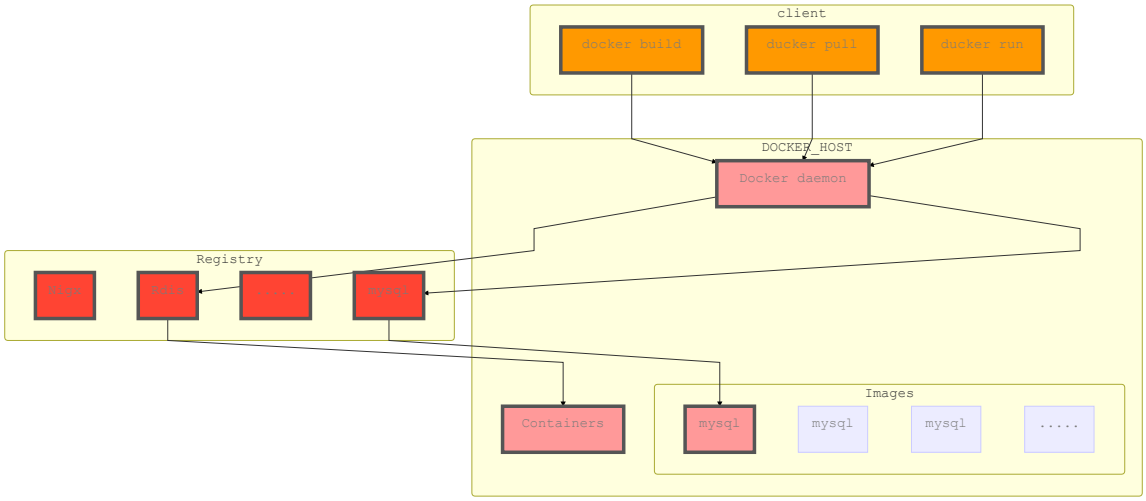
1.前提条件

- Docker.支持以下的CentOS版本:CentOS 7 (64-bit)
- CentOS 6.5 (64-bit) 或更高的版本
- 前提条件
  - 目前，CentOS仅发行版本中的内核支持Docker。
  - Docker运行在CentOS 7上，要求系统为64位、系统内核版本为3.10以上。
  - Docker运行在CentOS-6.5或更高的版本的CentOS上，要求系统为64位、系统内核版本为2.6.32-431或者更高版本。
- 查看自己的内核
  - uname命令用于打印当前系统相关信息(内核版本号、硬件架构、主机名称和操作系统类型等)。

```
uname -r
cat /etc/redhat-release
```

2.docker的基本组成

- docker的架构图



  - 镜像
    - Docker镜像(lmage) 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
    - 镜像和容器的关系类似类与对象的关系:

Docker	面向对象
容器	对象
镜像	类

  - 容器
    - Docker利用容器(Container) 独立运行的一个或一组应用。容器是用镜像创建的运行实例。
    - 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。
    - 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。
    - 容器的定义和镜像几乎一模一样，也是一堆层的统一视角， 唯一区别在于容器的最上面那一层是可读可写的。
  - 仓库
    - 仓库(Repository) 是集中存放镜像文件的场所。
    - 仓库(Repository)和仓库注册服务器(Registry) 是有区别的。|仓库注册服务器上往往存放着多个仓库，每个仓库中又包含了多个镜像，每个镜像有不同的标签(tag)。
    - 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

- 总结
  - 仓储/镜像/容器的概念:
  - 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似乎image镜像文件。只有通过这个镜像文件才能生成Docker容器。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。
  - image文件生成的容器实例，本身也是一个文件，称为镜像文件。
  - 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
  - 至于仓储，就是放了一堆镜像的地方，我们可以把镜像发布到仓储中，需要的时候从仓储中拉下来就可以了。

3.安装步骤

- CentOS6
  - yum install -y epel-release （Docker使用EPEL发布, RHEL系的OS首先要确保持有EPEL仓库,否则先检查OS的版本,然后安装相应的EPEL包。）
  - yum install -y docker-io (安装)
  - 安装后的配置文件: /etc/sysconfig/docker ()
  - 启动Docker后台服务: service docker start
  - docker version 验证
- CentOS7
  - 参考:https://docs.docker.com/install/linux/docker-ce/centos/
  - 1.安装所需的包。`yum-utils`提供了`yum-config-manager` 效用，并`device-mapper-persistent-data`和`lvm2`由需要`devicemapper`存储驱动程序。

```
sudo yum install -y yum-utils device-mapper-persistent-data lvm2
```

  - 2.使用以下命令设置**稳定**存储库。即使您还想从**边缘**或**测试**存储库安装构建，您始终需要**稳定的**存储 库。

```
 sudo yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
```

  - 3.安装*最新版本*的Docker CE

```
sudo yum install docker-ce
```

  - 启动 docker

```
 systemctl start docker
```

  - `docker`通过运行`hello-world` 映像验证是否已正确安装。

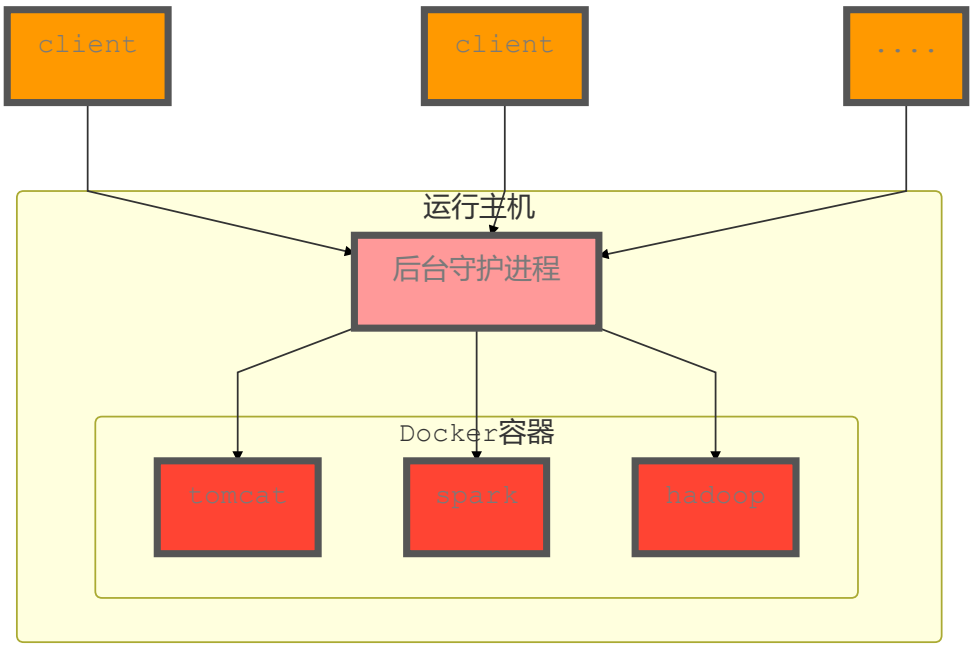
```
sudo docker run hello-world
```

4.使用和优化

- run: docker run xxx
- 阿里云加速:
- 网易云加速:

5.底层原理

- Docker如何运行
    - Docker是一个Client-Server结构的系统，Docker守护 进程运行在主机上，然 后通过Socket连接从客户端访问，守护进程从客户端接受命令并管理运行在主机上的容器。容器，是一个运行时环境，就是我们前面说到的集装箱。



- Docker比VM块
    - (1)docker有着比虚拟机更少的抽象层。由于docker不需要Hypervisor实现硬件资源虚拟化,运行在docker容器上的程序直接使用的都是实际物理机的硬件资源。因此在CPU、内存利用率上docker将会在效率上有明显优势。
    - (2)docker利用的是宿主机的内核,而不需要Guest OS。因此，当新建一个容器时,docker不需要和虚拟机一样重新加载一个操作系统内核。仍而避免引寻、加载操作系统内核返个比较费时费资源的过程，当新建一个虚拟机时,虚拟机软件需要加载GuestOS,返个新建过程是分钟级别的。而docker由于直接利用宿主机的操作系统，则省略了返个过程，因此新建一个docker容器只需要几秒钟。

	Docker容器	虚拟机(VM)
操作系统	与宿主机共享OS	宿主机OS上运行虚拟机OS
存储大小	镜像小，便于存储与传输	镜像庞大(vmdk、vdi等)
运行性能	几乎无额外性能损失	操作系统额外的CPU、内存消耗
移植性	轻便、灵活，适应于Linux	笨重，与虚拟化技术耦合度高
硬件亲和性	面向软件开发者	面向硬件运维者

三.Docker常用命令

1.帮助命令

- docker version
- docker info
- docker --help

2.镜像命令:

- docker images [列出主机上的镜像]
  - 各个选项说明:
  - REPOSITORY: 表示镜像的仓库源
  - TAG: 镜像的标签
  - IMAGE ID:镜像ID
  - CREATED:镜像创建时间
  - SIZE: 镜像大小
  - 同一仓库源可以有多个TAG，代表这个仓库源的不同个版本，我们使用REPOSITORY:TAG来定义不同的镜像。如果你不指定一个镜像的版本标签，例如你只使用ubuntu, docker将默认使用ubuntu:latest镜像
  - options说明:
    - -a :列出本地所有的镜像(含中间映像层)
    - -q:只显示镜像ID
    - --digests:显示镜像的摘要船息
    - --no-trunc :显示完整的镜像信息
- docker search xxx镜像的名字 [搜索镜像]
  - 网站 https://hub.docker.com
  - 命令:

        - docker search [OPTIONS] 镜像名字
        - OPTIONS说明:
            - --no-trunc : 显示完整的镜像描述 命令
            - -s:列出收藏数不小于指定值的镜像。
            - -- automated:只列出automated build类型的镜像;
- docker pull xxx [下载镜像]
    - docker pull 镜像名字[:TAG]
- docker rmi xxx镜像名字ID [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:TAG 镜像名2:TAG [例子: docker rmi -f hello-world nginx]
    - 删除全部 docker rmi -f $(docker images -qa)

3.容器命令:

- 新建并启动容器: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
    - 例如: docker run -it asdaswew
    - OPTIONS说明(常用) :有些是一个减号，有些是两个减号
    - --name="容器新名字":为容器指定一一个名称;
    - -d:后台运行容器，并返回容器ID，也即启动守护式容器;
    - -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;
    - -P:随机端口映射;
    - -p:指定端口映射，有以下四种格式 ip:hostPort:containerPortip.:containerPort hostPort:containerPortcontainerPort
- 列出当前所有正在运行的容器:docker ps [OPTIONS]
    - OPTIONS说明(常用)
    - -a :列出当前所有正在运行的容器+历史上运行过的
    - -l:显示最近创建的容器。
    - -n: 显示最近n个创建的容器。
    - -q :静默模式，只显示容器编号。
    - --no-trunc :不截断输出
- 退出容器
    - exit:容器停止退出
    - ctrl+P+Q:容器不停止退出
- 启动
    - docker start 容器id 或 容器名
- 重启
    - docker restart 容器id
- 停止
    - docker stop 容器id 或 容器名
- 强制停止
    - docker kill 容器id 或 容器名
- 删除已停止的

    - docker rm 容器id
    - 一次性删除多个容器
        - docker rm -f $(docker ps -a -q)
        - docker ps -a -q | xargs docker rm
- 重点
    - 启动守护式容器 docker run -d 容器名
    - 使用镜像centos:latest以后台模式启动一个容器 docker run -d centos 问题:然后docker ps -a进行查看，会发现容器已经退出
    - 很重要的要说明的一点: Docker容器后台运行,就必须有一个前台进程. 容器运行的命令如果不是那些一直挂起的命令 (比如运行top, tail) ，就是会自动退出的。
    - 这个是docker的机制问题,比如你的web容器，我们以nginx为例，正常情况下，我们配置启动服务只需要启动响应的service即可。例如service nginx start 但是,这样故,nginx为后台进程模式运行,就导致docker前台没有运行的应用，这样的容器后台启动后，会立即自杀因为他觉得他没事可做了.所以，最佳的解决方案是，将你要运行的程序以前台进程的形式运行
    - 查看容器日志
    - 查看容器内运行的进程
    - 查看容器内部细节
    - 进入正在运行的容器并以命令行交互
    - 从容器内拷贝文件到主机上

四.Docker镜像

1.是什么?

- 镜像是一种轻量级、可执行的独立软件包，用来打包软件运行环境和基于运行环境开发的软件，它包含运行某个软件所需的所有内容,包括代码、运行时、库、环境变量和配置文件。
- UnionFS (联合文件系统)
    - UnionFS (联合文件系统) : Union文件系统(UnionFS) 是一- 种分层、轻量级并且高性能的文件系统，它支持对文件系统的修改作为一次提交来一一层层的叠加，同时可以将不同目录挂载到同- - 个虚拟文件系统下(unite several directories into a single virtualfilesystem)。Union 文件系统是Docker镜像的基础。镜像可以通过分层来进行继承，基于基础镜像(没有父镜像)，可以制作各种具体的应用镜像。
    - 特性:一次同时加载多个文件系统，但从外面看起来，只能看到-一个文件系统，联合加载会把各层文件系统叠加起来，这样最终的文件系统会包含所有底层的文件和目录
- Docker镜像加载原理:
    - docker的镜像实际上由一~层- -层的文件系统组成，这种层级的文件系统UnionFS。
    - bootfs(boot file system)主要包含bootloader和kernel, bootloader主要是引导加载kernel, Linux刚启动时会加载bofs文件系统，在Docker镜像的最底层是bootfs。 这一-层与我们典型的Linux/Unix系统是一一样的，包含boot加载器和内核。当boot加载完成之后整个内核就都在内存中了，此时内存的使用权已由bootfs转交给内核， 此时系统也会卸载bootfs。
    - rootfs (root file system)，在bootfs之上。包含的就是典型Linux系统中的/dev, /proc, /bin, /etc等标准目录和文件。rootfs就是各种不同的操作系统发行版，比如Ubuntu, Centos等等。