

2021.2.22 Docker 的基本常识

•Docker 的使用场景

Docker 的容器技术可以在一台主机上轻松为任何应用创建一个轻量级的、可移植的、自给自足的容器。通过这种容器打包应用程序，意味着简化了重新部署、调试这些琐碎的重复工作，极大的提高了工作效率。例如：[项目从腾讯云迁移阿里云](#)，如果采用了 Docker 容

器技术，迁移只需要在新的云服务器上启动项目需要的容器即可。↵

二 . Docker 的优点

1.快速的交付和部署

对（DevOps）人员来说，最希望的就是项目环境一次创建或配置，可以在任意地方正常运行。开发者可以使用一个标准的镜像来构建一套项目开发容器，代码开发完成之后，运维人员可以直接此容器打包迁移。Docker 可以快速创建容器，快速迭代应用程序，并让整个过程全程可见，使团队中的其他成员更容易理解应用程序是如何创建和工作的。Docker 将会大量地节约开发、测试、部署的时间。↵

2. 更高效的虚拟化

Docker 容器的运行不需要额外的管理系统支持，它是内核级的虚拟化，并且容器调用的是宿主机的 bin 和 lib。因此可以实现更高的性能和效率。↵

3. 更轻松的迁移和扩展

Docker 容器几乎可以在任意的平台上运行，包括物理机、虚拟机、公有云、私有云、个人电脑、服务器等。这种兼容性可以让用户把一个应用程序从一个平台直接迁移到另外一个平台。↵

4. 简单的管理

使用 Docker 只需要小小的修改，就可以替代以往大量的更新工作。所有的修改都以增量的方式被分发和更新，从而实现自动化并且高效的管理。↵

三 . Docker 的核心概念

1 镜像 (Image)

Docker 镜像的含义：一个镜像包含了一个应用程序

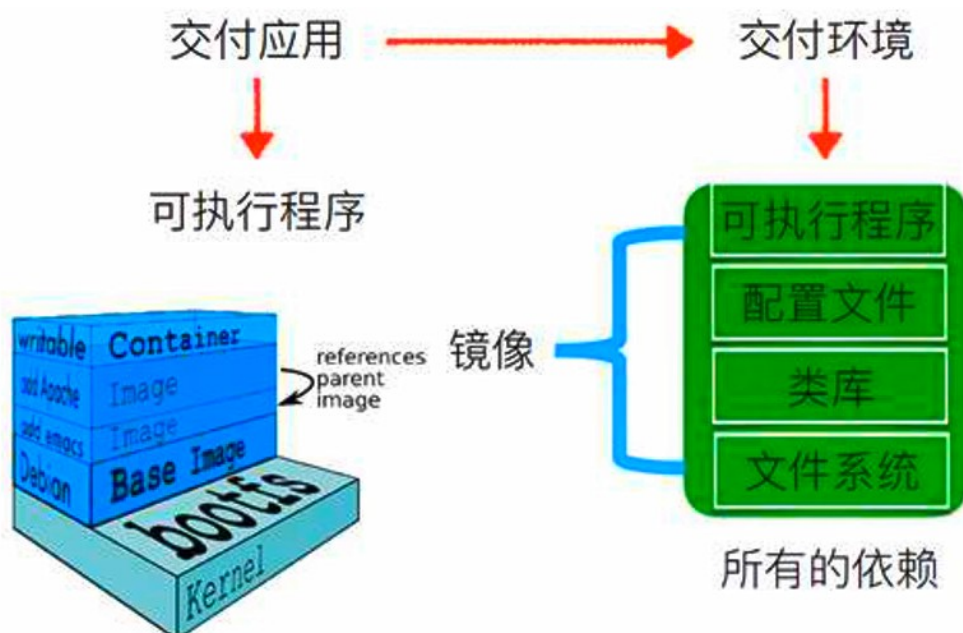
镜像是创建容器的基础，镜像类似传统虚拟机的快照，可以理解为是一个面向 Docker 容器引擎的只读模板。比如：一个镜像可以包含一个完整的 CentOS 操作系统环境，里面仅安装了 Apache 的应用程序（称之为 Apache 镜像）或用户需要的其它应用程序。↵

Docker 提供了一个很简单的机制来创建镜像或者更新现有的镜像（堆叠），用户甚至可以直接从其他人那里获取一个已经做好的镜像来直接使用。↵

2 容器 (Container)

注意：容器相互之间是没有任何关联的

容器是从镜像创建的运行实例。它可以被启动、开始、停止、删除。多个容器之间都是相互隔离的、保证了应用平台的安全性。可以把容器看做是一个正在运行的简易版的 Linux 环境（包括 root 用户权限、进程空间、用户空间和网络空间等）和运行在其中的一个或一组应用程序。Docker 容器用来运行和隔离应用地方。↵

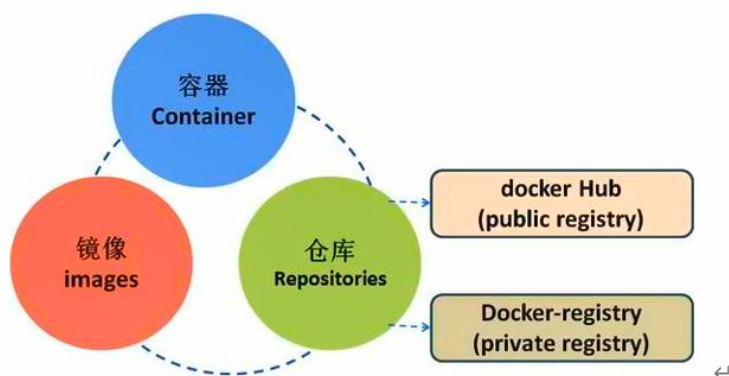


注意：镜像是只读的，容器在启动的时候创建一层可写层作为最上层。↵

3. 仓库 (Repository)

仓库是集中存放镜像的地方。通常会把仓库和仓库注册服务器（Registry）混为一谈，并不严格区分。实际上，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/>），存放了数量庞大的官方镜像供用户下载。国内的公开仓库有 Docker Pool 等，可以提供大陆用户更稳定快速的访问。当然，用户也可以在本地网络内创建一个私有仓库。当用户创建了自己的镜像之后就可以使用 **push** 命令将它上传到公有或者私有仓库，这样下次在另外一台机器上使用这个镜像时候，只需要从仓库上 **pull** 下来就可以了。↵



注意：Docker 仓库的概念跟 Git 类似，注册服务器可以理解为 GitHub 这样的托管服务。↵

四 . Docker 的安装

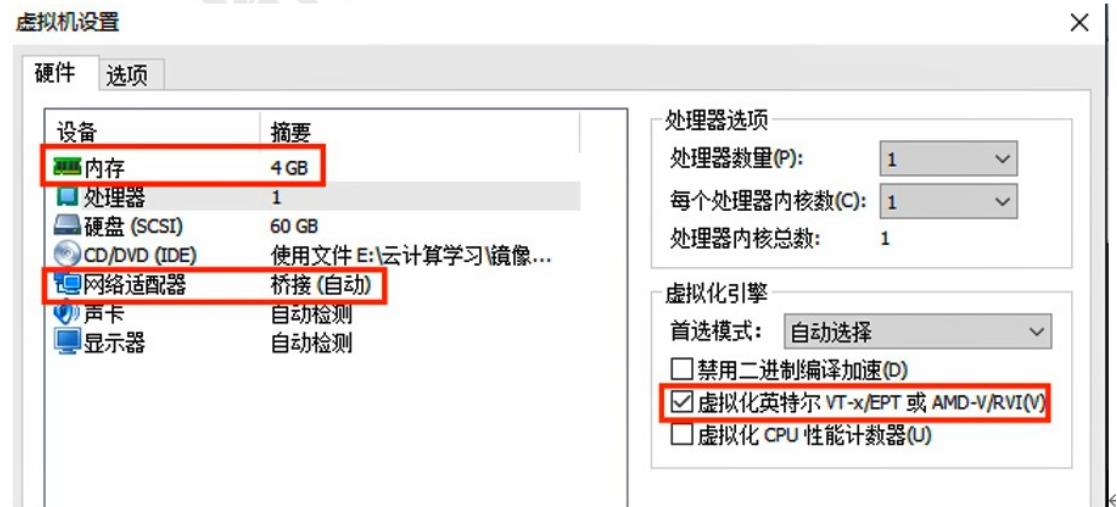
Docker 支持在主流的操作系统平台上使用，包括 Windows 系统、Linux 系统及 MacOS 系统等。目前最新的 RedHat 、CentOS、Ubuntu 等操作系统官方软件源中都已经默认自带了 Docker 包，可以直接安装使用，也可以用 Docker 自己的 YUM 源进行配置。↵

Docker 发行版本：↵

- Docker-EE 企业版↵
- Docker-CE 社区版↵

1. windows 安装 Docker (作为了解)

本次选用 Windows7 系统安装，虚拟硬件如下：↵



运行 DockerToolbox-1.11.2↵

一路下一步↵



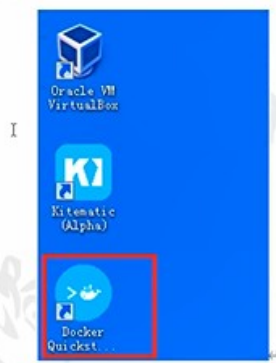
所有都安装↵



点击完成↵

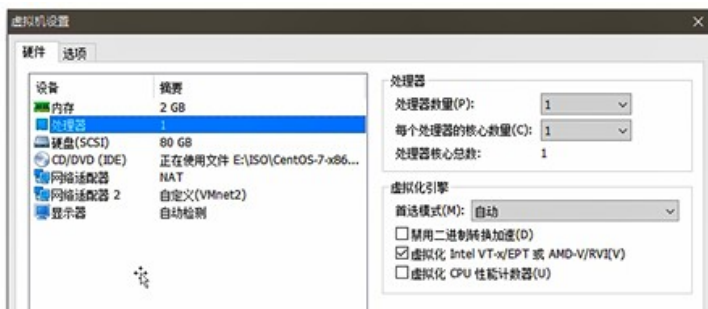


运行 Docker Quickstart Terminal



2. linux 安装 Docker

CentOS 系统下安装 Docker 可以有两种方式：一种是使用 curl 获得 Docker 的安装脚本进行安装，另一种是使用 YUM 仓库来安装 Docker。主要注意的是目前 Docker 只能支持 64 位系统。



1) 安装前准备，关闭所有的防火墙

```
[root@localhost ~]# iptables -F
[root@localhost ~]# systemctl stop firewalld
[root@localhost ~]# setenforce 0
```

2) 保证可以访问互联网

```
[root@localhost ~]# ping www.baidu.com
PING www.a.shifen.com (112.80.248.75) 56(84) bytes of data.
64 bytes from 112.80.248.75 (112.80.248.75): icmp_seq=1 ttl=128 time=30.6 ms
64 bytes from 112.80.248.75 (112.80.248.75): icmp_seq=2 ttl=128 time=38.2 ms
64 bytes from 112.80.248.75 (112.80.248.75): icmp_seq=3 ttl=128 time=30.1 ms
64 bytes from 112.80.248.75 (112.80.248.75): icmp_seq=4 ttl=128 time=32.0 ms
```

```
[root@localhost ~]# ls /etc/yum.repos.d/
backup CentOS-Base.repo CentOS-Media.repo
[root@localhost yum.repos.d]# yum clean all && yum makecache fast
```

3) 默认安装

```
[root@localhost ~]# yum -y install docker
```

4) 开启 Docker 并设置开启自启动

```
[root@localhost ~]# systemctl start docker
[root@localhost ~]# systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker to
/usr/lib/systemd/system/docker.
[root@localhost ~]# docker version
Client:
Version:      1.13.1
API version:  1.26
```

·安装 Docker-ce (最新的版本)

1) 安装前准备，关闭所有的防火墙

```
[root@localhost ~]# iptables -F
[root@localhost ~]# systemctl stop firewalld
[root@localhost ~]# setenforce 0
```

2) 保证可以访问互联网

```
[root@localhost ~]# ping www.baidu.com
PING www.a.shifen.com (112.80.248.75) 56(84) bytes of data.
64 bytes from 112.80.248.75 (112.80.248.75): icmp_seq=1 ttl=128 time=30.6 ms
64 bytes from 112.80.248.75 (112.80.248.75): icmp_seq=2 ttl=128 time=38.2 ms
64 bytes from 112.80.248.75 (112.80.248.75): icmp_seq=3 ttl=128 time=30.1 ms
64 bytes from 112.80.248.75 (112.80.248.75): icmp_seq=4 ttl=128 time=32.0 ms
```

3) 配置本地的 yum 仓库

```
[root@localhost ~]# ls /etc/yum.repos.d/
```

4) 下载阿里云的仓库

```
[root@localhost ~]# wget http://mirrors.aliyun.com/repo/Centos-7.repo
```

5) 解决 Docker 的依赖关系

```
[root@localhost ~]# wget http://mirrors.aliyun.com/repo/Centos-7.repo
[root@localhost ~]# wget http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
[root@localhost ~]# yum -y install docker-ce
```

6) 启动并设置开机自启动

```
[root@localhost ~]# systemctl start docker
[root@localhost ~]# systemctl enable docker
[root@localhost ~]# docker version
[root@localhost ~]# docker info
```

·配置阿里云的镜像加速

 <https://cr.console.aliyun.com/> (主要推荐)

1) 修改配置文件

```
[root@localhost ~]# vim /etc/docker/daemon.json ( 文件可能没有 )
{
"registry-mirrors":["https://nyakyfun.mirror.aliyuncs.com"]
}
[root@localhost ~]#systemctl daemon-reload
[root@localhost ~]# systemctl restart docker
```

2) <https://www.daocloud.io/mirror#taccelerator-doc>

执 行 : curl -sSL https://get.daocloud.io/daotools/set_mirror.sh|sh -s
<http://f1361db2.m.daocloud.io>

五 . Docker 的基础操作

1. 查看 docker 版本信息

```
[root@localhost ~]# docker version
[root@localhost ~]# docker info
```

·docker 镜像操作

Docker 创建容器前需要本地存在对应的镜像, 如果本地加载不到相关镜像, Docker 默认就会尝试从镜像仓库 <https://hub.docker.com> 下载, 这是由 Docker 官方维护的一个公共仓库, 可以满足用户的绝大部分需求, 用户也可以通过配置来使用自定义的镜像仓库。↵

↵

1、docker search //从 Docker Hub 搜索镜像↵

命令格式: docker search 关键字↵

参数: ↵

--automated=false 仅显示自动创建的镜像↵

--no-trunc=false 输出信息不截断显示↵

-s, --stars=0 指定仅显示评价为指定星级以上的镜像↵

```
[root@localhost ~]# docker search nginx↵
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
nginx	Official build of Nginx.	13174	[OK]	
jwilder/nginx-proxy	Automated Nginx reverse proxy for docker con...	1802		[OK]
richarvey/nginx-php-fpm	Container running Nginx + PHP-FPM capable of...	773		[OK]
linuxserver/nginx	An Nginx container, brought to you by LinuxS...	112		
bitnami/nginx	Bitnami nginx Docker Image	83		[OK]
tiangolo/nginx-rtmp	Docker image with Nginx using the nginx-rtmp...	72		[OK]
jc21/nginx-proxy-manager	Docker container for managing Nginx proxy ho...	60		
alfg/nginx-rtmp	NGINX, nginx-rtmp-module and FFmpeg from sou...	59		[OK]
nginxdemos/hello	NGINX webserver that serves a simple page co...	48		[OK]
jlesage/nginx-proxy-manager	Docker container for Nginx Proxy Manager	42		[OK]
nginx/nginx-ingress	NGINX Ingress Controller for Kubernetes	30		
privatebin/nginx-fpm-alpine	PrivateBin running on an Nginx, php-fpm & AL...	24		[OK]
schmunk42/nginx-redirect	A very simple container to redirect HTTP tra...	18		[OK]
nginxinc/nginx-unprivileged	Unprivileged NGINX Dockerfiles	15		
centos/nginx-112-centos7	Platform for running nginx 1.12 or building ...	13		
blacklabelops/nginx	Dockerized Nginx Reverse Proxy Server.	13		[OK]
centos/nginx-18-centos7	Platform for running nginx 1.8 or building n...	13		
nginx/nginx-prometheus-exporter	NGINX Prometheus Exporter	12		
raulr/nginx-wordpress	Nginx front-end for the official wordpress:f...	12		[OK]
sophos/nginx-vts-exporter	Simple server that scrapes Nginx vts stats a...	7		[OK]
mailu/nginx	Mailu nginx frontend	6		[OK]
bitnami/nginx-ingress-controller	Bitnami Docker Image for NGINX Ingress Contr...	5		[OK]
wodby/nginx	Generic nginx	1		[OK]
ansibleplaybookbundle/nginx-apb	An APB to deploy NGINX	1		[OK]

1) 查找镜像

```
[root@localhost ~]# docker search nginx #查找镜像
```

NAME	DESCRIPTION	STARS
nginx	Official build of Nginx.	14447
jwilder/nginx-proxy	Automated Nginx reverse proxy for docker con...	1965
richarvey/nginx-php-fpm	Container running Nginx + PHP-FPM capable of...	807
jc21/nginx-proxy-manager	Docker container for managing Nginx proxy ho...	147
linuxserver/nginx	An Nginx container, brought to you by LinuxS...	141
tiangolo/nginx-rtmp	Docker image with Nginx using the nginx-rtmp...	113
bitnami/nginx	Bitnami nginx Docker Image	94
jlesage/nginx-proxy-manager	Docker container for Nginx Proxy Manager	94

浏览器查找镜像

hub.docker.com/

1. 下载 nginx 镜像

```
[root@localhost ~]# docker pull nginx #下载 nginx 镜像
```

```
[root@localhost ~]# docker run--name k8s2-nginx -d -p 8000:80 nginx #定义开启 nginx
```

```
[root@localhost ~]# docker images #查看已存在镜像
```

访问 linux 的真机 IP 地址

192.168.200.5:8000

3) 查看镜像信息

使用 docker images #查看系统已有镜像

格式 : docker images 仓库名称[:标签]

示例 :

```
[root@server02 ~]# docker images nginx
```

显示信息 :

REPOSITORY : 镜像名

TAG : 标签

IMAGE ID : ID 号

CREATED : 创建的时间天数

SIZE : 镜像的内存的大小

4) 查看镜像的详细信息

命令格式：docker inspect 镜像名或 ID

示例：

```
[root@server02 ~]# docker inspect nginx
```

5) 为镜像添加新的标签 (类似于改名)

命令格式：docker tag 名称[:标签] 新名称[:新标签]

示例：为 nginx 改名为 nginx-1

```
[root@server02 ~]# docker tag nginx nginx-1
REPOSITORY    TAG       IMAGE ID        CREATED         SIZE
nginx-1        latest    f6d0b4767a6c    7 weeks ago    133MB
nginx          latest    f6d0b4767a6c    7 weeks ago    133MB
```

注意：两个镜像都一样，删除一个也无所谓，除了名字不一样其余都一样

6) 删除镜像 (rmi)

删除镜像的方式有两种：使用镜像标签删除或者 ID 号

命令格式：docker rmi 镜像名称

： docker rmi 镜像 ID

示例：删除 nginx 镜像

```
[root@server02 ~]# docker rmi nginx-1
Untagged: nginx-1:latest
```

7) 存出镜像和载入镜像

当需要把一台机器上的镜像迁移到另一台机器上的时候，需要把镜像保存成本地文件，这个过程叫存出镜像，可以使用 docker save 命令进行存储操作

命令格式：docker save -o 存储镜像为本地文件 镜像名

示例：导出镜像

```
[root@server02 ~]# docker save -o nginx nginx #导出镜像
```

```
[root@server02 ~]# ls -l nginx    #查看存出的镜像文件
```

```
-rw----- 1 root root 137152000 3月  8 11:07 nginx
```

导出的文件可以发送给任何一台主机

示例：导入镜像

命令格式：docker load < 导出的本地镜像文件

或者使用：docker load --input 本地镜像名

```
[root@server02 ~]# docker load < nginx    #导入镜像文件
```

```
Loaded image: nginx:latest
```

8) 上传镜像

本地存储的镜像越来越多，就需要指定一个专门存放这些镜像的地方也就是仓库了。目前比较方便的就是公共仓库，默认上传到 Docker Hub 官方仓库，需要注册使用公共仓库的账号，可以使用 docker login 命令来输入用户名、密码和邮箱来完成注册和登录。在上传镜像之前，需要对本地镜像添加新的标签，然后再使用 docker push 命令进行上传。

命令格式: docker push 仓库名称:标签

例如在公共仓库已经注册成功了一个账号为 crushlinux 和对应的密码，新增镜像的标签为 192.168.200.12:5000/nginx:test

```
[root@server02 ~]# docker login    #执行登录的命令
```

```
Username: crushlinux                #输入仓库账号
```

```
Password: xxxxxx                    #输入仓库账号密码
```

```
Login succeeded
```

成功登入后就可以上传的

```
[root@server02 ~]# docker tag nginx
192.168.200.12:5000/nginx:test #给仓库改名
[root@server02 ~]# docker push 192.168.200.12:5000/nginx:test
#上传镜像
```

·docker 的容器操作