一，Docker简介

百科说：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，容器是完全使用沙箱机制，相互之间不会有任何接口。

看起来有点雾，用过虚拟机的应该对虚拟化技术有点印象，不知道也没关系，就把它当成轻量级的虚拟机吧（虽然一个是完全虚拟化，一个是操作系统层虚拟化），这个解释到位：https://www.cnblogs.com/kex1n/p/6933039.html

百科又说：Docker 使用客户端-服务器 (C/S) 架构模式 使用远程API来管理和创建Docker容器。Docker 容器（Container）通过 Docker 镜像（Image）来创建，二者之间的关系类似于面向对象编程中的对象与类

那Docker由什么组成呢， 包括三个基本概念:

仓库（Repository）

镜像（Image）

容器(Container）

打个比方：你如果想玩英雄联盟中骚气的亚索，你首先得有这个英雄（Docker的镜像），然后你得花金币去英雄商店（Docker的仓库）买，接着进游戏就会看到一个半蹲的发型飘逸的剑客（Docker的容器），所以：

1，其中Registry是Docker用于存放镜像文件的仓库，Docker 仓库的概念跟Git 类似（就像商店存放所有的英雄，只是更改英雄的权限在某些非程序员手里）。

2，所谓镜像就是构建容器的源代码，是一个只读的模板，由一层一层的文件系统组成的，类似于虚拟机的镜像（英雄也是只读的，有自己的技能被动，你也不能进行操作）。

3，那么容器就是由Docker镜像创建的运行实例，类似于虚拟机，容器之间是相互隔离的，包含特定的应用及其所需的依赖文件（好比每个英雄都是隔离的，都有自己的皮肤，技能以及走的路线)。

   注：Docker Hub是Docker公司提供的一个注册服务器（Register）来保存多个仓库，每个仓库又可以包含多个具备不同tag的镜像。

二，安装Docker

我是虚拟机装的Centos7，linux 3.10 内核，docker官方说至少3.8以上，建议3.10以上（ubuntu下要linux内核3.8以上， RHEL/Centos 的内核修补过， centos6.5的版本就可以——这个可以试试）

1，root账户登录，查看内核版本如下

[root@localhost ~]# uname -a

Linux localhost.localdomain 3.10.0-957.el7.x86\_64 #1 SMP Thu Nov 8 23:39:32 UTC 2018

x86\_64 x86\_64 x86\_64 GNU/Linux

2，把yum包更新到最新

[root@localhost ~]# yum update

已加载插件：fastestmirror

Loading mirror speeds from cached hostfile

\* base: centos.ustc.edu.cn

\* extras: mirrors.aliyun.com

\* updates: mirrors.cn99.com

base | 3.6 kB 00:00:00

extras | 3.4 kB 00:00:00

updates | 3.4 kB 00:00:00

正在解决依赖关系

--> 正在检查事务

---> 软件包 NetworkManager.x86\_64.1.1.12.0-6.el7 将被 升级

---> 软件包 NetworkManager.x86\_64.1.1.12.0-10.el7\_6 将被 更新

（期间要选择确认，输入 y 即可）

3，安装需要的软件包， yum-util 提供yum-config-manager功能，另外两个是devicemapper驱动依赖的

[root@localhost ~]# yum install -y yum-utils device-mapper-persistent-data lvm2

已加载插件：fastestmirror

Loading mirror speeds from cached hostfile

\* base: centos.ustc.edu.cn

\* extras: mirrors.aliyun.com

\* updates: mirrors.cn99.com

软件包 device-mapper-persistent-data-0.7.3-3.el7.x86\_64 已安装并且是最新版本

软件包 7:lvm2-2.02.180-10.el7\_6.8.x86\_64 已安装并且是最新版本

正在解决依赖关系

--> 正在检查事务

---> 软件包 yum-utils.noarch.0.1.1.31-50.el7 将被 安装

--> 正在处理依赖关系 python-kitchen，它被软件包 yum-utils-1.1.31-50.el7.noarch 需要

...

...

4，设置yum源（选择其中一个）

yum-config-manager --add-repo http://download.docker.com/linux/centos/docker-ce.repo（中央仓库）

yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo（阿里仓库）

[root@localhost ~]# yum-config-manager --add-repo

https://download.docker.com/linux/centos/docker-ce.repo

已加载插件：fastestmirror

adding repo from: https://download.docker.com/linux/centos/docker-ce.repo

grabbing file https://download.docker.com/linux/centos/docker-ce.repo to

/etc/yum.repos.d/docker-ce.repo

repo saved to /etc/yum.repos.d/docker-ce.repo

5，可以查看所有仓库中所有docker版本，并选择特定版本安装

[root@localhost ~]# yum list docker-ce --showduplicates | sort -r

已加载插件：fastestmirror

可安装的软件包

\* updates: mirrors.cn99.com

Loading mirror speeds from cached hostfile

\* extras: mirrors.aliyun.com

docker-ce.x86\_64 3:19.03.2-3.el7 docker-ce-stable

docker-ce.x86\_64 3:19.03.1-3.el7 docker-ce-stable

docker-ce.x86\_64 3:19.03.0-3.el7 docker-ce-stable

docker-ce.x86\_64 3:18.09.8-3.el7 docker-ce-stable

docker-ce.x86\_64 3:18.09.7-3.el7 docker-ce-stable

docker-ce.x86\_64 3:18.09.6-3.el7 docker-ce-stable

docker-ce.x86\_64 3:18.09.5-3.el7 docker-ce-stable

docker-ce.x86\_64 3:18.09.4-3.el7 docker-ce-stable

docker-ce.x86\_64 3:18.09.3-3.el7 docker-ce-stable

docker-ce.x86\_64 3:18.09.2-3.el7 docker-ce-stable

docker-ce.x86\_64 3:18.09.1-3.el7 docker-ce-stable

docker-ce.x86\_64 3:18.09.0-3.el7 docker-ce-stable

docker-ce.x86\_64 18.06.3.ce-3.el7 docker-ce-stable

docker-ce.x86\_64 18.06.2.ce-3.el7 docker-ce-stable

docker-ce.x86\_64 18.06.1.ce-3.el7 docker-ce-stable

docker-ce.x86\_64 18.06.0.ce-3.el7 docker-ce-stable

docker-ce.x86\_64 18.03.1.ce-1.el7.centos docker-ce-stable

docker-ce.x86\_64 18.03.0.ce-1.el7.centos docker-ce-stable

docker-ce.x86\_64 17.12.1.ce-1.el7.centos docker-ce-stable

6，安装Docker，命令：yum install docker-ce-版本号，我选的是docker-ce-18.03.1.ce，如下

[root@localhost ~]# yum install docker-ce-18.03.1.ce

已加载插件：fastestmirror

Loading mirror speeds from cached hostfile

\* base: centos.ustc.edu.cn

\* extras: mirrors.aliyun.com

\* updates: mirrors.cn99.com

正在解决依赖关系

--> 正在检查事务

---> 软件包 docker-ce.x86\_64.0.18.03.1.ce-1.el7.centos 将被 安装

（期间要选择确认，输入 y 即可）

7， 启动Docker，命令：systemctl start docker，然后加入开机启动，如下

[root@localhost ~]# systemctl start docker

[root@localhost ~]# systemctl enable docker

Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.

[root@localhost ~]# docker version

Client:

Version: 18.03.1-ce

API version: 1.37

Go version: go1.9.5

Git commit: 9ee9f40

Built: Thu Apr 26 07:20:16 2018

OS/Arch: linux/amd64

Experimental: false

Orchestrator: swarm

Server:

Engine:

Version: 18.03.1-ce

API version: 1.37 (minimum version 1.12)

Go version: go1.9.5

Git commit: 9ee9f40

Built: Thu Apr 26 07:23:58 2018

OS/Arch: linux/amd64

Experimental: false

附一些常用Docker命令，更多命令详解，请访问：http://www.docker.org.cn/dockerppt/106.html:

 -----------------     docker ps 查看当前正在运行的容器

 -----------------    docker ps -a 查看所有容器的状态

 -----------------    docker start/stop id/name 启动/停止某个容器

 -----------------    docker attach id 进入某个容器(使用exit退出后容器也跟着停止运行)

 -----------------    docker exec -ti id 启动一个伪终端以交互式的方式进入某个容器（使用exit退出后容器不停止运行）

 -----------------    docker images 查看本地镜像

 -----------------    docker rm id/name 删除某个容器

-----------------     docker rmi id/name 删除某个镜像

 -----------------    docker run --name test -ti ubuntu /bin/bash  复制ubuntu容器并且重命名为test且运行，然后以伪终端交互式方式进入容器，运行bash

-----------------     docker build -t soar/centos:7.1 .  通过当前目录下的Dockerfile创建一个名为soar/centos:7.1的镜像

 -----------------    docker run -d -p 2222:22 --name test soar/centos:7.1  以镜像soar/centos:7.1创建名为test的容器，并以后台模式运行，并做端口映射到宿主机2222端口，P参数重启容器宿主机端口会发生改变

————————————————

版权声明：本文为CSDN博主「秋刀鱼亦会过期」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/u014069688/article/details/100532774>

docker官网地址  https://www.docker.com/

docker启动命令,docker重启命令,docker关闭命令

启动

systemctl start docker

守护进程重启

systemctl daemon-reload

重启docker服务

systemctl restart docker / service docker restart

关闭

docker service docker stop / docker systemctl stop docker

————————————————

版权声明：本文为CSDN博主「OOEM」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/qq\_23315711/article/details/82224282