

Docker

一，什么是Docker技术

1. Docker 是一个开源项目，诞生于 2013 年初，最初是 dotCloud 公司内部的一个业余项目，诞生后异军突起，很受欢迎。它基于 Google 公司推出的 Go 语言实现。项目后来加入了 **Linux 基金会**，遵从了 **Apache 2.0 协议**，项目代码在 GitHub 上进行维护。
2. Docker 项目的**目标是实现轻量级的操作系统虚拟化解决方案**。Docker 的基础是 **Linux 容器 (LXC Linux container 是一种虚拟化的解决方案，这种是内核级的虚拟化)** 等技术。在 LXC 的基础上 Docker 进行了进一步的封装，屏蔽了底层的一些命令，让用户不需要去关心容器的管理，使得操作更为简便。用户操作 Docker 的容器就像操作一个快速轻量级的虚拟机一样简单。
3. 区别：Docker 和传统虚拟化的方式最大的不同点是，**Docker 是在操作系统层面上实现的虚拟化，直接复用本地主机的操作系统，而传统的方式则是在硬件层面实现的。**

二、为什么要使用Docker？

1. 与传统的虚拟化方式相比具有众多的**优势**。**a. Docker 容器的启动在秒级** **b. Docker 对系统资源利用率高**，一台主机上可以同时运行数千个 Docker 容器，容器之间是相互隔离的。**c. Docker 基本不消耗系统资源，使得运行在 Docker 里面的应用的性能很高。**
2. 相比于传统的虚拟化技术，Docker 在以下几个方面具有较大的优势。**a. 更快速的交付和部署**：开发者可以使用一个标准的镜像来构建一套开发容器，开发完成之后，运维人员可以直接使用这个容器来部署代码。**b. 更高效的虚拟化**：Docker 容器的运行不需要额外的 hypervisor（又称虚拟机监视器（英语：virtual machine monitor，缩写为 VMM），是用来建立与执行虚拟机）的软件、固件或硬件。）支持，它是内核级的虚拟化，因此可以实现更高的性能和效率。**c. 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。**d. 更简单的管理**：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。
3. 与传统虚拟机的对比：

特性	容器	虚拟机
启动	秒级	分钟级
硬盘使用	一般为 MB	一般为 GB
性能	接近原生	弱于
系统支持量	单机支持上千个容器	一般几十个

三、Docker 中的基本概念

1. Docker 中包括三个基本概念：**镜像 (Image)** **容器 (Container)** **仓库 (Repository)** 理解好这三个概念，就可以了解 Docker 的整个生命周期。
2. **镜像**：是一个只读的模板，例如一个完整的 Centos 操作系统镜像可以用来创建 Docker 容器，Docker 中提供了一个很简单的方式来创建镜像和更新镜像，甚至可以从其他地方直接拷贝已经做好的镜像直接使用。镜像有点类似于编程中的 Class 类，在运行的时候生成对象。
3. **容器**：是从镜像创建并运行的实例，就像一个启动好了的播放器程序，它可以被开始，停止，启动和删除。每个容器都是相互隔离的，绝对保证安全。你可以把容器看成一个启动了的 Linux 简化版系统，里面包括 root

用户权限，进程空间，用户空间和网络空间，还包括运行在里面的应用程序。

4. **仓库**：是集中存放镜像文件的地方。还有一种服务叫做仓库注册服务器（可以理解为GitHub这样的托管服务器），里面存放着多个仓库，每个仓库中又包含多个镜像，每个镜像又有不同的标签。仓库的概念有点类似于Git,也分为公有仓库和私有仓库，全世界最大的Docker仓库是Docker Hub,国内最大的Docker仓库是Docker Pool.用户可以在本地网络或者服务器上创建一个私有仓库，当用户创建了一个自己的镜像之后，使用push命令把镜像上传到自己的仓库中，下次在另外一台机器上使用这个镜像的时候，只需要从仓库中pull下来就可以直接使用了。

四、Docker安装

1. Docker的安装很简单，在linux系统上通过简单的几行命令就可以完成Docker的安装。
2. 我们介绍Centos7上Docker的安装，Centos7系统的CentOS-Extras库中已经带了Docker，可以直接使用yum命令安装

```
yum install docker
```

```
[root@bigdata-4 ~]# yum install docker
Loaded plugins: fastestmirror, langpacks
Determining fastest mirrors
epel/x86_64/metalink | 4.7 kB 00:00:00
* base: mirrors.163.com
* epel: mirrors.njupt.edu.cn
* extras: ap.stykers.moe
* updates: mirrors.163.com
base | 3.6 kB 00:00:00
epel | 4.7 kB 00:00:00
extras | 3.4 kB 00:00:00
updates | 3.4 kB 00:00:00
(1/4): extras/7/x86_64/primary_db | 200 kB 00:00:00
(2/4): epel/x86_64/updateinfo | 1.0 MB 00:00:01
(3/4): updates/7/x86_64/primary_db | 5.0 MB 00:00:03
(4/4): epel/x86_64/primary_db | 6.7 MB 00:00:05
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 2:1.13.1-91.git07f3374.e17.centos will be updated
--> Package docker.x86_64 2:1.13.1-96.gitb2f74b2.e17.centos will be an update
```

```
Updating      : 2:docker-common-1.13.1-96.gitb2f74b2.e17.centos.x86_64 1/6
Updating      : 2:docker-client-1.13.1-96.gitb2f74b2.e17.centos.x86_64 2/6
Updating      : 2:docker-1.13.1-96.gitb2f74b2.e17.centos.x86_64 3/6
Cleanup       : 2:docker-1.13.1-91.git07f3374.e17.centos.x86_64 4/6
Cleanup       : 2:docker-client-1.13.1-91.git07f3374.e17.centos.x86_64 5/6
Cleanup       : 2:docker-common-1.13.1-91.git07f3374.e17.centos.x86_64 6/6
Verifying     : 2:docker-1.13.1-96.gitb2f74b2.e17.centos.x86_64 1/6
Verifying     : 2:docker-client-1.13.1-96.gitb2f74b2.e17.centos.x86_64 2/6
Verifying     : 2:docker-common-1.13.1-96.gitb2f74b2.e17.centos.x86_64 3/6
Verifying     : 2:docker-common-1.13.1-91.git07f3374.e17.centos.x86_64 4/6
Verifying     : 2:docker-1.13.1-91.git07f3374.e17.centos.x86_64 5/6
Verifying     : 2:docker-client-1.13.1-91.git07f3374.e17.centos.x86_64 6/6

Updated:
  docker.x86_64 2:1.13.1-96.gitb2f74b2.e17.centos

Dependency Updated:
  docker-client.x86_64 2:1.13.1-96.gitb2f74b2.e17.centos
  docker-common.x86_64 2:1.13.1-96.gitb2f74b2.e17.centos

Complete!
```

3. 安装之后启动 Docker 服务，并让它随系统启动自动加载。

```
systemctl start docker.service
systemctl enable docker.service
```

4. 在命令行输入docker,如下图，说明docker安装成功

```
docker
```

```
[root@bigdata-4 ~]# docker
```

```
Usage:  docker COMMAND
```

```
A self-sufficient runtime for containers
```

```
Options:
```

```
  --config string      Location of client config files (default "/root/.docker")
  -D, --debug           Enable debug mode
  --help               Print usage
  -H, --host list       Daemon socket(s) to connect to (default [])
  -l, --log-level string Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string   Trust certs signed only by this CA (default "/root/.docker/ca.pem")
  --tlscert string      Path to TLS certificate file (default "/root/.docker/cert.pem")
  --tlskey string       Path to TLS key file (default "/root/.docker/key.pem")
  --tlsverify           Use TLS and verify the remote
  -v, --version         Print version information and quit
```

```
Management Commands:
```

```
  container    Manage containers
  image        Manage images
```

#默认是从官方仓库docker hub 查找镜像

```
docker search mysql
```

#过滤star大于100的镜像

```
docker search mysql -s 100
```

```
docker search mysql --filter stars=100
```