1.安装环境说明

Docker官方建议在Ubuntu中安装，因为Docker是基于Ubuntu发布的，而且一般Docker出现的问题Ubuntu是最先更新或者打补丁的。在很多版本的CentOS中是不支持更新最新的一些补丁包的。

由于我们学习的环境都使用的是CentOS，因此这里我们将Docker安装到CentOS上。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。

> 注意：这里建议安装在CentOS7.x以上的版本，在CentOS6.x的版本中，安装前需要安装其他很多的环境而且Docker很多补丁不支持更新。

- 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）。

```
[root@localhost ~]# uname -r
3.10.0-957.el7.x86_64
```

```
[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.安装Docker

使用yum命令在线安装

```
[root@localhost ~]# yum install docker
已加载插件：fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.163.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.163.com
base                                                    | 3.6 kB  00:00:00
extras                                                  | 3.4 kB  00:00:00
updates                                                 | 3.4 kB  00:00:00
updates/7/x86_64/primary_db                             | 6.5 MB  00:00:01
```

```
正在解决依赖关系
--> 正在检查事务
---> 软件包 docker.x86_64.2.1.13.1-94.gitb2f74b2.el7.centos 将被 升级
---> 软件包 docker.x86_64.2.1.13.1-96.gitb2f74b2.el7.centos 将被 更新
--> 正在处理依赖关系 docker-common = 2:1.13.1-96.gitb2f74b2.el7.centos，它被软件包 2:docker-1.13.1-96.gitb2f74b2.el7.centos.x86_64 需要
--> 正在处理依赖关系 docker-client = 2:1.13.1-96.gitb2f74b2.el7.centos，它被软件包 2:docker-1.13.1-96.gitb2f74b2.el7.centos.x86_64 需要
--> 正在检查事务
---> 软件包 docker-client.x86_64.2.1.13.1-94.gitb2f74b2.el7.centos 将被 升级
---> 软件包 docker-client.x86_64.2.1.13.1-96.gitb2f74b2.el7.centos 将被 更新
---> 软件包 docker-common.x86_64.2.1.13.1-94.gitb2f74b2.el7.centos 将被 升级
---> 软件包 docker-common.x86_64.2.1.13.1-96.gitb2f74b2.el7.centos 将被 更新
--> 解决依赖关系完成

依赖关系解决

================================================================================
================================================================================
 Package                                架构
     版本
源                                   大小
================================================================================
================================================================================
正在更新:
 docker                                 x86_64
       2:1.13.1-96.gitb2f74b2.el7.centos
   extras                                18 M
为依赖而更新:
 docker-client                          x86_64
       2:1.13.1-96.gitb2f74b2.el7.centos
   extras                               3.9 M
 docker-common                          x86_64
       2:1.13.1-96.gitb2f74b2.el7.centos
   extras                                96 k

事务概要
```

```
================================================================================
================================================================================
========================================
升级  1 软件包 (+2 依赖软件包)

总下载量：22 M
Is this ok [y/d/N]: y
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
(1/3): docker-common-1.13.1-96.gitb2f74b2.el7.centos.x86_64.rpm

                |  96 kB  00:00:00
(2/3): docker-client-1.13.1-96.gitb2f74b2.el7.centos.x86_64.rpm

                | 3.9 MB  00:00:01
(3/3): docker-1.13.1-96.gitb2f74b2.el7.centos.x86_64.rpm

                 |  18 MB  00:00:06
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
----------------------------------------
总计

      3.1 MB/s |  22 MB  00:00:07
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  正在更新     : 2:docker-common-1.13.1-
96.gitb2f74b2.el7.centos.x86_64
                                                                      1/6
  正在更新     : 2:docker-client-1.13.1-
96.gitb2f74b2.el7.centos.x86_64
                                                                      2/6
  正在更新     : 2:docker-1.13.1-96.gitb2f74b2.el7.centos.x86_64

                                    3/6
  清理         : 2:docker-1.13.1-94.gitb2f74b2.el7.centos.x86_64

                                      4/6
  清理         : 2:docker-client-1.13.1-
94.gitb2f74b2.el7.centos.x86_64
                                                                      5/6
  清理         : 2:docker-common-1.13.1-
94.gitb2f74b2.el7.centos.x86_64
                                                                      6/6
```

```
  验证中       : 2:docker-1.13.1-96.gitb2f74b2.el7.centos.x86_64
                                                              1/6
  验证中       : 2:docker-client-1.13.1-
96.gitb2f74b2.el7.centos.x86_64
                                                              2/6
  验证中       : 2:docker-common-1.13.1-
96.gitb2f74b2.el7.centos.x86_64
                                                              3/6
  验证中       : 2:docker-client-1.13.1-
94.gitb2f74b2.el7.centos.x86_64
                                                              4/6
  验证中       : 2:docker-1.13.1-94.gitb2f74b2.el7.centos.x86_64
                                                              5/6
  验证中       : 2:docker-common-1.13.1-
94.gitb2f74b2.el7.centos.x86_64
                                                              6/6

更新完毕:
  docker.x86_64 2:1.13.1-96.gitb2f74b2.el7.centos



作为依赖被升级:
  docker-client.x86_64 2:1.13.1-96.gitb2f74b2.el7.centos
                       docker-common.x86_64 2:1.13.1-
96.gitb2f74b2.el7.centos

完毕!
```

3.安装后查看Docker版本

```
[root@localhost ~]# docker -v
Docker version 1.13.1, build b2f74b2/1.13.1
```

4.启动与停止Docker

启动docker：service docker start 停止docker：service docker stop 重启docker：service docker restart 查看docker状态：service docker status 开机启动：chkconfig docker on 查看docker概要信息：docker info 查看docker帮助文档：docker --help

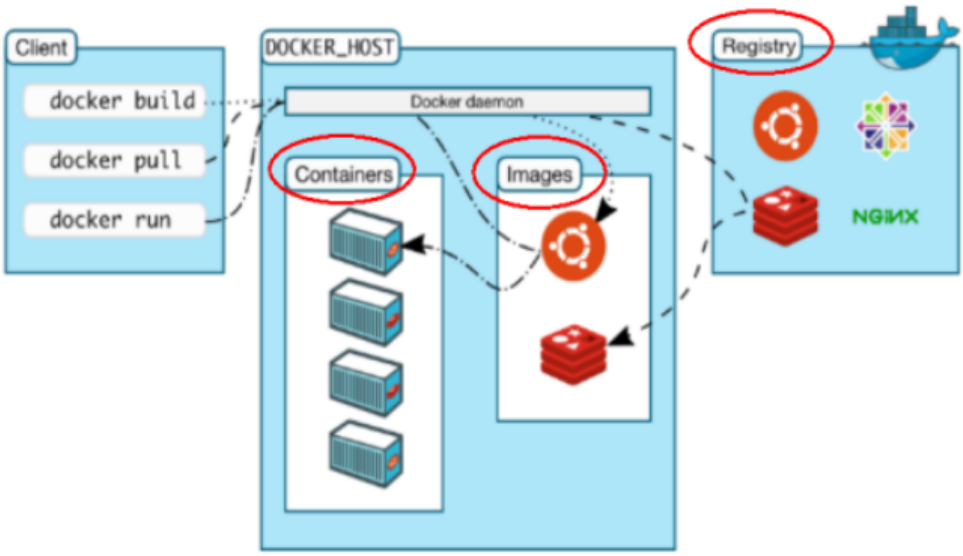
```
[root@localhost ~]# service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
```

```
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since 一 2019-06-17 15:17:51 CST; 48s ago
     Docs: http://docs.docker.com
 Main PID: 15257 (dockerd-current)
   CGroup: /system.slice/docker.service
           ├─15257 /usr/bin/dockerd-current --add-runtime docker-runc=/usr/libexec/docke...
           └─15262 /usr/bin/docker-containerd-current -l unix:///var/run/docker/libconta...

6月 17 15:17:49 localhost.localdomain dockerd-current[15257]: time="2019-06-17T15:17:49...
6月 17 15:17:50 localhost.localdomain dockerd-current[15257]: time="2019-06-17T15:17:50...
6月 17 15:17:50 localhost.localdomain dockerd-current[15257]: time="2019-06-17T15:17:50...
6月 17 15:17:50 localhost.localdomain dockerd-current[15257]: time="2019-06-17T15:17:50...
6月 17 15:17:50 localhost.localdomain dockerd-current[15257]: time="2019-06-17T15:17:50...
6月 17 15:17:50 localhost.localdomain dockerd-current[15257]: time="2019-06-17T15:17:50...
6月 17 15:17:51 localhost.localdomain dockerd-current[15257]: time="2019-06-17T15:17:50...
6月 17 15:17:51 localhost.localdomain dockerd-current[15257]: time="2019-06-17T15:17:50...
6月 17 15:17:51 localhost.localdomain systemd[1]: Started Docker Application Container....
6月 17 15:17:51 localhost.localdomain dockerd-current[15257]: time="2019-06-17T15:17:51...
Hint: Some lines were ellipsized, use -l to show in full.
```

4.Docker的基本组成

Docker的架构图



4.1.镜像（image）

Docker 镜像（Image）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。

容器与镜像的关系类似于面向对象编程中的对象与类。

Docker	面向对象
容器	对象
镜像	类

4.2.容器（container）

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。

4.3.仓库（repository）

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.

```
从仓库中拉取镜像
使用镜像启动容器
```

4.4.总结

需要正确的理解仓储/镜像/容器这几个概念:

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。