CentOs7 安装社区版 docker-ce

作者：chengzime
原文链接：https://ld246.com/article/1565606600600
来源网站：链滴

1. 查看是否已存在 docker

```
docker version
```

2. 如果存在,删除旧版本 docker

```
yum remove docker \
          docker-client \
          docker-client-latest \
          docker-common \
          docker-latest \
          docker-latest-logrotate \
          docker-logrotate \
          docker-selinux \
          docker-engine-selinux \
          docker-engine
```

3. 检查内核版本，返回的值大于3.10即可

```
uname -r
```

4. 安装所需的软件包

yum-utils 提供的 yum-config-manager

device-mapper-persistent-data 和 lvm2

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

5. yum-config-manager 添加镜像源

```
yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.
epo
```

6. 可以启用edge和测试存储库，默认是关闭

```
# 开启
yum-config-manager --enable docker-ce-edge
yum-config-manager --enable docker-ce-test

# 关闭
yum-config-manager --disable docker-ce-edge
yum-config-manager --disable docker-ce-test
```

7. 安装 docker-ce

```
yum install docker-ce
```

8. 启动 docker

原文链接：CentOs7 安装社区版 docker-ce

```
systemctl start docker  # centos7.x
service docker start    # centos6.x
```

9. 查看 docker 状态

```
systemctl status docker
```

10. 测试运行 docker

```
docker run hello-world
```

11. 重新启动 docker

```
systemctl restart  docker  # centos7.x
service docker restart     # centos6.x
```

12. 关闭 docker

```
systemctl stop docker  # centos7.x
service docker stop    # centos6.x
```

13. 设置 docker 开机启动

```
systemctl enable docker
```

14. 查看版本号

```
docker version
```

15. 查看是否运行

```
ps -ef|grep docker
```

16. 其他命令

```
# 查看容器日志
docker logs 容器名称/id

# 设置容器自动重启,默认是 no
docker update --restart=always 容器名称/id

# 查看是否在运行
docker ps

# 查看所有容器包括停止运行的
docker ps -a

# 停止容器
docker stop 容器名称/id
```

```
# 删除容器
docker rm 容器名称/id

# 查看所有镜像
docker images

# 查询镜像库
docker search mysql (项目名称)

# 拉取镜像
docker pull mysql:5.7 (项目名,可加版本号,用 : 隔开)

# 重启容器
docker restart 容器名称/id

# 删除镜像
docker rmi 镜像id
```

17. docker 的默认安装路径

- docker 的默认安装路径在 /var/lib/docker

```
[root@bogon ~]# cd /var/lib/docker
[root@bogon docker]# ll
总用量 4
drwx------.  2 root root   24 8月   8 14:58 builder
drwx------.  4 root root   92 8月   8 14:58 buildkit
drwx------.  4 root root  150 8月  14 11:40 containers
drwx------.  3 root root   22 8月   8 14:58 image
drwxr-x---.  3 root root   19 8月   8 14:58 network
drwx------. 27 root root 4096 8月  14 11:40 overlay2
drwx------.  4 root root   32 8月   8 14:58 plugins
drwx------.  2 root root    6 8月  14 11:24 runtimes
drwx------.  2 root root    6 8月   8 14:58 swarm
drwx------.  2 root root    6 8月  14 11:24 tmp
drwx------.  2 root root    6 8月   8 14:58 trust
drwx------.  3 root root   97 8月   8 18:51 volumes
[root@bogon docker]#
```

- docker支持多种 graphDriver, 如: vfs、devicemapper、overlay、overlay2、aufs等, 我的使用存储驱动是 overlay2

使用不同的存储驱动, 会在 /var/lib/docker 目录下以驱动名称命名生成存储文件夹

```
drwx------. 27 root root 4096 8月  14 11:40 overlay2
```

- 镜像的位置在 /var/lib/docker/image 下所使用驱动名称命名的文件夹中, 我的存储驱动是 overla2, 所以目录是 /var/lib/docker/image/overlay2

```
root@bogon docker]# cd /var/lib/docker/image
[root@bogon image]# ll
总用量 0
```

```
drwx------. 5 root root 81 8月   8 18:50 overlay2
[root@bogon image]# cd overlay2/
[root@bogon overlay2]# ll
总用量 4
drwx------. 4 root root  58 8月   8 15:04 distribution
drwx------. 4 root root  37 8月   8 14:58 imagedb
drwx------. 5 root root  45 8月   8 18:08 layerdb
-rw-------. 1 root root 789 8月   8 18:50 repositories.json
[root@bogon overlay2]#
```

- 查看 repositories.json 文件

```
cat /var/lib/docker/image/overlay2
```

```
[root@bogon ~]# cat /var/lib/docker/image/overlay2/repositories.json
{"Repositories":{"b3log/solo":{"b3log/solo:latest":"sha256:3ad1297684d5fb4019421e747c821
ec339666846cdd1b41cb3c7ad32eb1da94","b3log/solo@sha256:4cf76d515a6fdb8352cab571
2620b1f3f3a56a9f83e48ea46557a3e4dfe355d":"sha256:3ad1297684d5fb4019421e747c8214e
339666846cdd1b41cb3c7ad32eb1da94"},"mysql":{"mysql:5.7":"sha256:f6509bac49801f48628
67728aba66d64533aaa7d384e03b75a8fe4e6c0f6599","mysql@sha256:540488d8f0e04c1077d
7934d1c1511fe417e2221dff508ce4621f5efe6131db":"sha256:f6509bac49801f48628167728ab
66d64533aaa7d384e03b75a8fe4e6c0f6599"},"nginx":{"nginx:latest":"sha256:e445ab08b2be8b
78655b714f89e5db9504f67defd5c7408a00bade679a50d44","nginx@sha256:eb3320e2f9ca40
b7c0aa71aea3cf7ce7d018f03a372564dbdb023646958770b":"sha256:e445ab08b2be8b17865
b714f89e5db9504f67defd5c7408a00bade679a50d44"}}}
```

- 这是一个 json 文件, 我们来格式化

```
cat /var/lib/docker/image/overlay2/repositories.json | python -m json.tool
```

```
[root@bogon ~]# cat /var/lib/docker/image/overlay2/repositories.json | python -m json.tool
{
    "Repositories": {
        "b3log/solo": {
            "b3log/solo:latest": "sha256:3ad1297684d5fb4019421e747c8214ec339666846cdd1b4
cb3c7ad32eb1da94",
            "b3log/solo@sha256:4cf76d515a6fdb8352cab57102620b1f3f3a56a9f83e48ea46557a3
4dfe355d": "sha256:3ad1297684d5fb4019421e747c8214ec339666846cdd1b41cb3c7ad32eb1
a94"
        },
        "mysql": {
            "mysql:5.7": "sha256:f6509bac49801f48628167728aba66d64533aaa7d384e03b75a8fe
e6c0f6599",
            "mysql@sha256:540488d8f0e04c1077d17934d1c1511fe417e2221dff508ce4621f5efe6
31db": "sha256:f6509bac49801f48628167728aba66d64533aaa7d384e03b75a8fe4e6c0f6599"
        },
        "nginx": {
            "nginx:latest": "sha256:e445ab08b2be8b178655b714f89e5db9504f67defd5c7408a00b
de679a50d44",
            "nginx@sha256:eb3320e2f9ca409b7c0aa71aea3cf7ce7d018f03a372564dbdb0236469
8770b": "sha256:e445ab08b2be8b178655b714f89e5db9504f67defd5c7408a00bade679a50d4
"
        }
    }
```

}

- 我们发现这里面跟我们使用 docker images 打印出来的容器信息一致

```
[root@bogon overlay2]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
b3log/solo          latest              3ad1297684d5        6 days ago          150MB
nginx               latest              e445ab08b2be        3 weeks ago         126MB
mysql               5.7                 f6509bac4980        3 weeks ago         373MB
[root@bogon overlay2]#
```

docker images 打印出来的 IMAGE ID 是完整 ID 的缩写