

4.Docker容器的创建，启动和停止

1.容器是独立运行的一个或一组应用，及他们的运行环境。容器是Docker中的一个重要的概念。 2.docker容器的启动有两种方式:

- 基于镜像新建容器并启动 例如我们可以启动一个容器，打印出当前的日历表

```
docker run dockerlaowang/centos cal
```

```
docker: /bin/cal: not found
[root@bigdata-4 centos]# docker run dockerlaowang/centos cal
      August 2019
Su Mo Tu We Th Fr Sa
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

[root@bigdata-4 centos]#
```

我们还可以通过指定参数，启动一个bash交互终端。

```
docker run -t -i dockerlaowang/centos /bin/bash
```

```
[root@bigdata-4 centos]# docker run -t -i dockerlaowang/centos /bin/bash
[root@a8b966cbf488 /]# cal
      August 2019
Su Mo Tu We Th Fr Sa
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

[root@a8b966cbf488 /]#
```

参数-t让Docker分配一个伪终端并绑定在容器的标准输入上，-i让容器的标准输入保持打开。

使用docker run命令来启动容器，docker在后台运行的标准操作包括

1. 检查本地是否存在指定的镜像，不存在则从公有仓库下载
2. 使用镜像创建并启动容器
3. 分配一个文件系统，并在只读的镜像层外面挂载一层可读可写层
4. 从宿主主机配置的网桥接口中桥接一个虚拟接口到容器中去
5. 从地址池分配一个ip地址给容器
6. 执行用户指定的应用程序
7. 执行完毕之后容器被终止

- 将在终止状态的容器从新启动 可以使用docker start命令，直接将一个已经终止的容器启动运行起来。

```
docker stop a8b966cbf488
docker start a8b966cbf488
```

3.守护态运行 很多时候，我们希望容易在后台以守护态运行，此时可以添加-d参数来实现（d是daemon的首字母）例如我们启动centos后台容器，每隔一秒打印当天的日历。

```
docker run -d dockerlaowang/centos /bin/sh -c "while true;do echo hello docker;sleep 1;done"
```

启动之后，我们使用docker ps查看容器的信息

```
[root@bigdata-4 centos]# docker ps
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAME |
|---------------|----------------------|------------------------|----------------|---------------|-------|--------|
| d708387c5f74 | dockerlaowang/centos | "/bin/sh -c 'while..." | 30 seconds ago | Up 30 seconds | | practi |
| tical_morse | | | | | | |
| a8b966cbf488 | dockerlaowang/centos | "/bin/bash" | 12 minutes ago | Up 5 minutes | | practi |
| tical_shannon | | | | | | |

```
[root@bigdata-4 centos]#
```

要查看启动的centos容器中的输出，可以使用

```
docker logs cid
```

```
[root@bigdata-4 centos]# docker ps
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAME |
|--------------|----------------------|-------------------------|----------------|---------------|-------|-------------------|
| d708387c5f74 | dockerlaowang/centos | "/bin/sh -c 'while...'" | 30 seconds ago | Up 30 seconds | | practical_morse |
| a8b966cbf488 | dockerlaowang/centos | "/bin/bash" | 12 minutes ago | Up 5 minutes | | practical_shannon |

```
[root@bigdata-4 centos]# docker logs dockerlaowang/centos
Error: No such container: dockerlaowang/centos
[root@bigdata-4 centos]# docker logs d708387c5f74
hello docker
hello docker
hello docker
hello docker
hello docker
hello docker
```

[illegible]

4.终止容器 使用docker stop来终止一个运行中的容器。并且可以使用docker ps -a来看终止状态的容器。

```
docker stop cid
```

