docker容器

原创 Mr大表哥 2017-02-27 22:01:40 评论(0) 227人阅读

简单的说，容器是独立运行的一个或一组应用，以及它们的运行态环境。对应的，虚拟机可以理解为模拟运行的一整套操作系统（提供了运行态环境和其他系统环境）和跑在上面的应用。
下面介绍如何来管理一个容器，包括创建、启动和停止等。
启动容器有两种方式，一种是基于镜像新建一个容器并启动，另外一个是将在终止状态（stopped）的容器重新启动。
1、新建并启动
所需要的命令主要为docker run
下面的命令则启动一个 bash 终端，允许用户进行交互。

```
[root@localhost media]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
docker.io/centos    latest              50dae1ee8677        6 months ago        196.7 MB
docker.io/centos    centos6             cf2c3ece5e41        7 months ago        194.6 MB
[root@localhost ~]# docker run -it docker.io/centos:centos6 /bin/bash
[root@0eecd2f8a665 /]#
```

-t 选项让 Docker 分配一个伪终端（pseudo-tty）并绑定到容器的标准输入上；
-i则让容器的标准输入保持打开(即交互式)；
可以使用--name给容器起个形象的名称。
/bin/bash就是在/bin下启用bash,即开启命令解释器。（就是在容器内要执行的命令）
容器内无法使用ip a来查看容器的IP，但是可以用ifconfig。

在交互模式下，用户可以通过所创建的终端来输入命令，例如：
容器的核心为所执行的应用程序，所需要的资源都是应用程序运行所必需的。除此之外，并没有其它的资源。可以在伪终端中利用ps或 top 来查看进程信息。

```
[root@0eecd2f8a665 /]# pwd
/
[root@0eecd2f8a665 /]# ps -ef
UID         PID   PPID  C STIME TTY          TIME CMD
root          1      0  0 10:25 ?        00:00:00 /bin/bash
root         12      1  0 10:34 ?        00:00:00 ps -ef
[root@0eecd2f8a665 /]# ps
  PID TTY          TIME CMD
    1 ?        00:00:00 bash
   13 ?        00:00:00 ps
```

ps -ef：可以查看完整的所有进程信息

```
[root@0eecd2f8a665 /]# top

top - 10:36:34 up 29 min,  0 users,  load average: 0.41, 0.90, 0.79
Tasks:   2 total,   1 running,   1 sleeping,   0 stopped,   0 zombie
Cpu(s):  2.7%us,  0.3%sy,  0.0%ni, 97.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1868660k total,  1629728k used,   238932k free,        4k buffers
Swap:  4194300k total,   102100k used,  4092200k free,   833572k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
    1 root      20   0 11496 1184  832 S  0.0  0.1   0:00.06 bash
   14 root      20   0 14944 1120  908 R  0.0  0.1   0:00.01 top
```

由上可见，容器中仅运行了指定的 bash 应用。这种特点使得 Docker 对资源的利用率极高，是货真价实的轻量级虚拟化。

如果这个时候我们正常退出， exit 或者Ctrl+d（退出容器，容器处于终止状态），

docker ps -a （查看容器处于什么状态）查看容器处于 Exit 状态如果需要正常退出可以使用 CTRL–p + CTRL -q ----就像先按 CTRL -p 然后 CTRL -q 退出伪终端

```
[root@64f58c7359dd /]# [root@localhost ~]#
[root@localhost ~]#
```

先按住crtl+p，然后放开p键，但是不放开ctrl键，然后按q,最后回车即可。这样退出容器，容器依然在后台运行，保持up状态。

```
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                          COMMAND             CREATED             STATUS
      PORTS               NAMES
64f58c7359dd        docker.io/centos:centos6       "/bin/bash"         22 minutes ago      Up 22 minutes
                    sad_engelbart
0eecd2f8a665        docker.io/centos:centos6       "/bin/bash"         41 minutes ago      Exited (0) 22 minutes
ago                 fervent_kare
[root@localhost ~]#
[root@localhost ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
docker.io/centos    latest              50dae1ee8677        6 months ago        196.7 MB
docker.io/centos    centos6             cf2c3ece5e41        7 months ago        194.6 MB
[root@localhost ~]# docker run docker.io/centos:latest /bin/echo 'hello world'
hello world
[root@localhost ~]#
```

上面的命令输出一个 "hello world" ，之后终止容器。(即在容器内执行命令，命令执行完后，就终止容器)，这跟在本地直接执行 /bin/echo 'hello world' 几乎感觉不出任何区别。

```
[root@localhost ~]# docker ps
CONTAINER ID        IMAGE                          COMMAND             CREATED             STATUS              PO
RTS                 NAMES
64f58c7359dd        docker.io/centos:centos6       "/bin/bash"         2 hours ago         Up 2 hours
                    sad_engelbart
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                          COMMAND                  CREATED             STATUS
          PORTS               NAMES
0c73f8990ce9        docker.io/centos:latest        "/bin/echo 'hello wor"   48 minutes ago      Exited (0) 47 min
utes ago                        angry_payne
64f58c7359dd        docker.io/centos:centos6       "/bin/bash"              2 hours ago         Up 2 hours
                                sad_engelbart
0eecd2f8a665        docker.io/centos:centos6       "/bin/bash"              2 hours ago         Exited (0) 2 hour
s ago                           fervent_kare
[root@localhost ~]#
```

docker ps：后面不加参数，只显示up状态的容器；　　docker ps -a: 显示所有状态的容器

总结：当利用docker run 来创建容器时，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，不存在就从公有仓库下载

2）. 利用镜像创建并启动一个容器

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）. 从地址池配置一个ip地址给容器

6）. 执行用户指定的应用程序

7）. 执行完毕后容器被终止

2、启动exited状态的容器到up状态

```
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                         COMMAND                  CREATED             STATUS
          PORTS               NAMES
0c73f8990ce9        docker.io/centos:latest       "/bin/echo 'hello wor"   56 minutes ago      Exited (0) 56 min
utes ago                        angry_payne
64f58c7359dd        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Up 2 hours
                                sad_engelbart
0eecd2f8a665        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Exited (0) 2 hour
s ago                           fervent_kare
[root@localhost ~]# docker start 0eecd2f8a665
0eecd2f8a665
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                         COMMAND                  CREATED             STATUS
          PORTS               NAMES
0c73f8990ce9        docker.io/centos:latest       "/bin/echo 'hello wor"   56 minutes ago      Exited (0) 56 min
utes ago                        angry_payne
64f58c7359dd        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Up 2 hours
                                sad_engelbart
0eecd2f8a665        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Up 3 seconds
                                fervent_kare
[root@localhost ~]#
```

格式：docker start exited状态容器ID号

①停止正在运行的容器（当Docker容器中指定的应用终结时，容器也自动终止。例如对于前面所讲中启动了一个终端的容器，用户通过 exit 命令或Ctrl+d来退出终端时，所创建的容器立刻终止
终止状态的容器可以用docker ps -a 命令看到。）

```
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                         COMMAND                  CREATED             STATUS
          PORTS               NAMES
0c73f8990ce9        docker.io/centos:latest       "/bin/echo 'hello wor"   59 minutes ago      Exited (0) 59 min
utes ago                        angry_payne
64f58c7359dd        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Up 2 hours
                                sad_engelbart
0eecd2f8a665        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Up 2 minutes
                                fervent_kare
[root@localhost ~]# docker stop 64f58c7359dd
64f58c7359dd
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                         COMMAND                  CREATED             STATUS
          PORTS               NAMES
0c73f8990ce9        docker.io/centos:latest       "/bin/echo 'hello wor"   About an hour ago   Exited (0) About
an hour ago                     angry_payne
64f58c7359dd        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Exited (137) 2 se
conds ago                       sad_engelbart
0eecd2f8a665        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Up 3 minutes
                                fervent_kare
[root@localhost ~]#
```

格式：docker stop up状态的ID号
（这是停止正在运行的容器的第一种方法）

```
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                         COMMAND                  CREATED             STATUS
          PORTS               NAMES
0c73f8990ce9        docker.io/centos:latest       "/bin/echo 'hello wor"   About an hour ago   Exited (0) About
an hour ago                     angry_payne
64f58c7359dd        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Exited (137) 16 s
econds ago                      sad_engelbart
0eecd2f8a665        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Up 3 minutes
                                fervent_kare
[root@localhost ~]# docker kill 0eecd2f8a665
0eecd2f8a665
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                         COMMAND                  CREATED             STATUS
          PORTS               NAMES
0c73f8990ce9        docker.io/centos:latest       "/bin/echo 'hello wor"   About an hour ago   Exited (0) About
an hour ago                     angry_payne
64f58c7359dd        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Exited (137) 34 s
econds ago                      sad_engelbart
0eecd2f8a665        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Exited (137) 4 se
conds ago                       fervent_kare
[root@localhost ~]#
```

格式：docker kill up状态的容器的ID号
（停止正在运行的容器的第二种方法）

②重启容器

```
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                         COMMAND                  CREATED             STATUS
          PORTS               NAMES
0c73f8990ce9        docker.io/centos:latest       "/bin/echo 'hello wor"   About an hour ago   Exited (0) About
an hour ago                     angry_payne
64f58c7359dd        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Exited (137) 4 mi
nutes ago                       sad_engelbart
0eecd2f8a665        docker.io/centos:centos6      "/bin/bash"              3 hours ago         Exited (137) 4 mi
nutes ago                       fervent_kare
[root@localhost ~]# docker restart 0eecd2f8a665
0eecd2f8a665
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                         COMMAND                  CREATED             STATUS
          PORTS               NAMES
0c73f8990ce9        docker.io/centos:latest       "/bin/echo 'hello wor"   About an hour ago   Exited (0) About
an hour ago                     angry_payne
64f58c7359dd        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Exited (137) 5 mi
nutes ago                       sad_engelbart
0eecd2f8a665        docker.io/centos:centos6      "/bin/bash"              3 hours ago         Up 2 seconds
                                fervent_kare
[root@localhost ~]#
```

格式:docker restart 需要重启的容器的ID号

③守护态运行（守护态运行就是在后台运行）

```
[root@localhost ~]# docker run -d docker.io/centos:latest /bin/sh -c "while true ; do echo hello world ; sleep 1 ; done"
7d94a359158eca28190735769995b565be18eab14489882658e93ceb293bbd00f
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                         COMMAND                  CREATED             STATUS                      PORTS
          NAMES
7d94a359158e        docker.io/centos:latest       "/bin/sh -c 'while tr"   37 seconds ago      Up 34 seconds
          determined_cori
0c73f8990ce9        docker.io/centos:latest       "/bin/echo 'hello wor"   About an hour ago   Exited (0) About an hour ago
          angry_payne
64f58c7359dd        docker.io/centos:centos6      "/bin/bash"              2 hours ago         Exited (137) 21 minutes ago
          sad_engelbart
0eecd2f8a665        docker.io/centos:centos6      "/bin/bash"              3 hours ago         Up 16 minutes
          fervent_kare
[root@localhost ~]#
```

容器启动后会返回一个唯一的 id，也可以通过dockerps命令来查看容器信息。

1．docker run -d 运行一个新的容器，我们通过-d 命令让他作为一个后台运行

2．docker.io/centos:latest 是一个我们想要在内部运行命令的镜像

3．/bin/sh -c 是我们想要在容器内部运行的命令

4．while true; do echo hello weibo; sleep 1; done 这是一个简单的脚本，我们仅仅只是每秒打印一次 hello word 一直到我们结束它

或者

```
[root@localhost ~]# docker run -dit docker.io/centos:latest /bin/bash
a80320cdf908211899599218 53d7b336fdddd5e0af221312e023719f31308938
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                         COMMAND                  CREATED             STATUS                      PORTS
          NAMES
a80320cdf908        docker.io/centos:latest       "/bin/bash"              2 seconds ago       Up 2 seconds
          tiny_blackwell
7d94a359158e        docker.io/centos:latest       "/bin/sh -c 'while tr"   9 minutes ago       Up 9 minutes
          determined_cori
0c73f8990ce9        docker.io/centos:latest       "/bin/echo 'hello wor"   About an hour ago   Exited (0) About an hour ago
          angry_payne
64f58c7359dd        docker.io/centos:centos6      "/bin/bash"              3 hours ago         Exited (137) 30 minutes ago
          sad_engelbart
0eecd2f8a665        docker.io/centos:centos6      "/bin/bash"              3 hours ago         Up 25 minutes
          fervent_kare
[root@localhost ~]#
```

④查看容器详细信息

```
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                     COMMAND                  CREATED             STATUS                      PORTS
                    NAMES
a80320cdf908        docker.io/centos:latest   "/bin/bash"              7 minutes ago       Up 7 minutes
                    tiny_blackwell
7d94a359158e        docker.io/centos:latest   "/bin/sh -c 'while tr"   17 minutes ago      Up 17 minutes
                    determined_cori
0c73f8990ce9        docker.io/centos:latest   "/bin/echo 'hello wor"   About an hour ago   Exited (0) About an hour ago
                    angry_payne
64f58c7359dd        docker.io/centos:centos6  "/bin/bash"              3 hours ago         Exited (137) 38 minutes ago
                    sad_engelbart
0eecd2f8a665        docker.io/centos:centos6  "/bin/bash"              3 hours ago         Up 32 minutes
                    fervent_kare
[root@localhost ~]# docker inspect a80320cdf908      格式：docker  inspect  需要查看的容器的ID号
[
    {
        "Id": "a80320cdf90821189959921853d7b336fdddd5e0af221312e023719f31308938",
        "Created": "2017-02-06T13:52:09.394714265Z",
        "Path": "/bin/bash",
        "Args": [],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 56666,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2017-02-06T13:52:09.937083455Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
```

只查看指定容器的详细信息的某一个部分

```
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                     COMMAND                  CREATED             STATUS                      PORTS
                    NAMES
a80320cdf908        docker.io/centos:latest   "/bin/bash"              14 minutes ago      Up 4 seconds
                    tiny_blackwell
7d94a359158e        docker.io/centos:latest   "/bin/sh -c 'while tr"   24 minutes ago      Up 24 minutes
                    determined_cori
0c73f8990ce9        docker.io/centos:latest   "/bin/echo 'hello wor"   About an hour ago   Exited (0) About an hour ago
                    angry_payne
64f58c7359dd        docker.io/centos:centos6  "/bin/bash"              3 hours ago         Exited (137) 45 minutes ago
                    sad_engelbart
0eecd2f8a665        docker.io/centos:centos6  "/bin/bash"              3 hours ago         Up 39 minutes
                    fervent_kare
[root@localhost ~]# docker inspect -f '{{.NetworkSettings.IPAddress}}' a80320cdf908
172.17.0.4
[root@localhost ~]#      用docker inspect查看容器的ip地址
```

```
[root@localhost ~]# docker inspect -f '{{.Config.Cmd}}' a80320cdf908
{[/bin/bash]}
[root@localhost ~]#      用docker inspect查看容器内执行的程序
```

⑤进入容器

在使用 -d 参数时，容器启动后会进入后台。某些时候需要进入容器进行操作，有很多种方法，包括使用docker attach 命令或nsenter命令。

```
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                     COMMAND                  CREATED             STATUS                      PORTS
                    NAMES
a80320cdf908        docker.io/centos:latest   "/bin/bash"              22 minutes ago      Up 7 minutes
                    tiny_blackwell
7d94a359158e        docker.io/centos:latest   "/bin/sh -c 'while tr"   32 minutes ago      Up 32 minutes
                    determined_cori
0c73f8990ce9        docker.io/centos:latest   "/bin/echo 'hello wor"   About an hour ago   Exited (0) About an hour ago
                    angry_payne
64f58c7359dd        docker.io/centos:centos6  "/bin/bash"              3 hours ago         Exited (137) 53 minutes ago
                    sad_engelbart
0eecd2f8a665        docker.io/centos:centos6  "/bin/bash"              3 hours ago         Up 47 minutes
                    fervent_kare
[root@localhost ~]# docker attach a80320cdf908      格式：docker attach  需要进入的容器的ID号
[root@a80320cdf908 /]#
```

```
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                     COMMAND                  CREATED             STATUS                      PORT
S                   NAMES
a80320cdf908        docker.io/centos:latest   "/bin/bash"              29 minutes ago      Exited (0) 2 seconds ago
                    tiny_blackwell
7d94a359158e        docker.io/centos:latest   "/bin/sh -c 'while tr"   39 minutes ago      Up 39 minutes
                    determined_cori
0c73f8990ce9        docker.io/centos:latest   "/bin/echo 'hello wor"   2 hours ago         Exited (0) 2 hours ago
                    angry_payne
64f58c7359dd        docker.io/centos:centos6  "/bin/bash"              3 hours ago         Exited (137) About an hour ago
                    sad_engelbart
0eecd2f8a665        docker.io/centos:centos6  "/bin/bash"              3 hours ago         Up 54 minutes
                    fervent_kare
[root@localhost ~]# docker attach --sig-proxy=false 7d94a359158e
hello world
hello world
hello world
hello world
hello world
hello world
^C[root@localhost ~]# docker ps
CONTAINER ID        IMAGE                     COMMAND                  CREATED             STATUS              PORTS
   NAMES
7d94a359158e        docker.io/centos:latest   "/bin/sh -c 'while tr"   42 minutes ago      Up 42 minutes
   determined_cori
0eecd2f8a665        docker.io/centos:centos6  "/bin/bash"              3 hours ago         Up 58 minutes
   fervent_kare
[root@localhost ~]#
```

不使用容器转发信号，允许我们使用 ctrl +c 来退出，执行docker ps 查看在后台运行。
但是使用 attach 命令有时候并不方便。当多个窗口同时 attach 到同一个容器的时候，所有窗口都会同步显示。当某个窗口因命令阻塞时,其他窗口也无法执行操作了。

```
[root@localhost ~]# docker ps
CONTAINER ID        IMAGE                     COMMAND                  CREATED             STATUS              PORTS
   NAMES
7d94a359158e        docker.io/centos:latest   "/bin/sh -c 'while tr"   46 minutes ago      Up 46 minutes
   determined_cori
0eecd2f8a665        docker.io/centos:centos6  "/bin/bash"              4 hours ago         Up About an hour
   fervent_kare
[root@localhost ~]# docker exec -it 0eecd2f8a665 /bin/bash
[root@0eecd2f8a665 /]#
```

格式：docker exec -it 需要进入的容器ID或者容器名 /bin/bash

使用nsenter进入容器（该方法不常用）

安装nsenter工具在util-linux包2.23版本后包含。如果系统中util-linux包没有该命令，可以按照下面的方法从源码安装

```
#wget https://www.kernel.org/pub/linux/utils/util-linux/v2.24/util-linux-2.24.tar.gz
#tar util-linux-2.24.tar.gz
#cd util-linux-2.24
# ./configure --without-ncurses&& make nsenter
#cpnsenter /usr/local/bin
```

nsenter可以访问另一个进程的名字空间。nsenter要正常工作需要有 root 权限

庆幸的是centos7使用的是util-linux-2.23，所以就直接使用系统提供的util-linux包了。

```
[root@localhost ~]# docker exec -it furious_jepsen /bin/bash
```

为了连接到容器，你还需要找到容器的第一个进程的PID，可以通过下面的命令获取。

```
PID=$(docker inspect --format "` `.`State`.`Pid`" <container>)
```

通过这个PID，就可以连接到这个容器：

```
nsenter --target $PID --mount --uts --ipc --net --pid
```

下面给出一个完整的例子：

```
[root@localhost ~]# docker run -idt centos:centos6 /bin/bash
e8969bbc12e836767469134873 20ad121e3d72d70b5e8f3a2f4fadbf8091c535
[root@localhost ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
        NAMES
e8969bbc12e8        centos:centos6      "/bin/bash"         7 seconds ago       Up 4 seconds
        elegant_nobel
[root@localhost ~]# PID=$(docker inspect --format "{{.State.Pid}}" e8969bbc12e8)
[root@localhost ~]# echo $PID
5178
[root@localhost ~]# nsenter --target $PID --mount --uts --ipc --net --pid
```

附：更简单的，建议大家下载 .bashrc_docker，并将内容放到 .bashrc中。

```
#wget ~https://github.com/yeasy/docker_practice/raw/master/_local/.bashrc_docker
#echo "[ -f ~/.bashrc_docker ] && . ~/.bashrc_docker" >> ~/.bashrc
#source ~/.bashrc
```

这个文件中定义了很多方便使用 Docker 的命令，例如docker-pid可以获取某个容器的PID；而docker-enter 可以进入容器或直接在容器内执行命令。

```
echo $(docker-pid<container>)
docker-enter <container> ls
```

⑥容器的导入和导出

```
[root@localhost ~]# docker ps
CONTAINER ID        IMAGE                     COMMAND                  CREATED             STATUS              PORTS
   NAMES
7d94a359158e        docker.io/centos:latest   "/bin/sh -c 'while tr"   About an hour ago   Up About an hour
   determined_cori
0eecd2f8a665        docker.io/centos:centos6  "/bin/bash"              4 hours ago         Up About an hour
   fervent_kare
[root@localhost ~]# docker export 7d94a359158e > abc.tar
[root@localhost ~]# ls\
>
abc.tar  anaconda-ks.cfg  公共  模板  视频  图片  文档  下载  音乐  桌面
[root@localhost ~]# cat abc.tar | docker import - centos7:test
sha256:1d9e14220217c8d7d83a8b6d685b8c7de2fa8668341c262c517212f6d168fa1f
[root@localhost ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
centos7             test                1d9e14220217        5 seconds ago       196.7 MB
docker.io/centos    latest              50dae1ee8677        6 months ago        196.7 MB
docker.io/centos    centos6             cf2c3ece5e41        7 months ago        194.6 MB
[root@localhost ~]#
```

此外，也可以通过指定 URL 或者某个目录来导入，例如

```
#docker import http://example.com/exampleimage.tgzexample/imagerepo
```

*注：用户既可以使用docker load 来导入镜像存储文件到本地镜像库，也可以使用docker import 来导入一个容器快照到本地镜像库。这两者的区别在于容器快照文件将丢弃所有的历史记录和元数据信息（即仅保存容器当时的快照状态），而镜像存储文件将保存完整记录，体积也要大。此外，从容器快照文件导入时可以重新指定标签等元数据信息。

⑦删除容器

```
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                     COMMAND                  CREATED             STATUS                      PORT
S                   NAMES
a80320cdf908        docker.io/centos:latest   "/bin/bash"              59 minutes ago      Exited (0) 30 minutes ago
                    tiny_blackwell
7d94a359158e        docker.io/centos:latest   "/bin/sh -c 'while tr"   About an hour ago   Up About an hour
                    determined_cori
0c73f8990ce9        docker.io/centos:latest   "/bin/echo 'hello wor"   2 hours ago         Exited (0) 2 hours ago
                    angry_payne
64f58c7359dd        docker.io/centos:centos6  "/bin/bash"              4 hours ago         Exited (137) About an hour ago
                    sad_engelbart
0eecd2f8a665        docker.io/centos:centos6  "/bin/bash"              4 hours ago         Up About an hour
                    fervent_kare
[root@localhost ~]# docker rm a80320cdf908
a80320cdf908
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE                     COMMAND                  CREATED             STATUS                      PORT
S                   NAMES
7d94a359158e        docker.io/centos:latest   "/bin/sh -c 'while tr"   About an hour ago   Up About an hour
                    determined_cori
0c73f8990ce9        docker.io/centos:latest   "/bin/echo 'hello wor"   2 hours ago         Exited (0) 2 hours ago
                    angry_payne
64f58c7359dd        docker.io/centos:centos6  "/bin/bash"              4 hours ago         Exited (137) About an hour ago
                    sad_engelbart
0eecd2f8a665        docker.io/centos:centos6  "/bin/bash"              4 hours ago         Up About an hour
                    fervent_kare
[root@localhost ~]#
```

格式：docker rm [-f] 处于终止状态的容器的ID号（加了-f是删除一个运行中的容器）

批量删除多个容器

```
[root@localhost ~]# docker rm $(docker ps -a -q)
0c73f8990ce9
64f58c7359dd
Failed to remove container (7d94a359158e): Error response from daemon: Conflict, You cannot remove a running container. Stop the
container before attempting removal or use -f
Failed to remove container (0eecd2f8a665): Error response from daemon: Conflict, You cannot remove a running container. Stop the
container before attempting removal or use -f
[root@localhost ~]# docker rm -f $(docker ps -a -q)
7d94a359158e
0eecd2f8a665
[root@localhost ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
[root@localhost ~]#
```

显示素有容器的ID号：docker ps -aq

```
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:e7:13:d3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.6/24 brd 192.168.1.255 scope global dynamic eno16777736
       valid_lft 69533sec preferred_lft 69533sec
    inet6 fe80::20c:29ff:fee7:13d3/64 scope link
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN qlen 500
    link/ether 52:54:00:41:14:c8 brd ff:ff:ff:ff:ff:ff
5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 02:42:da:d9:f9:a1 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:daff:fed9:f9a1/64 scope link
       valid_lft forever preferred_lft forever
[root@localhost ~]#
```

docker 0:叫虚拟网桥，在主机上自动产生，是容器的网关。