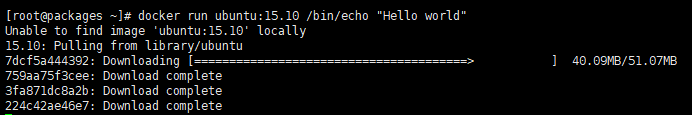
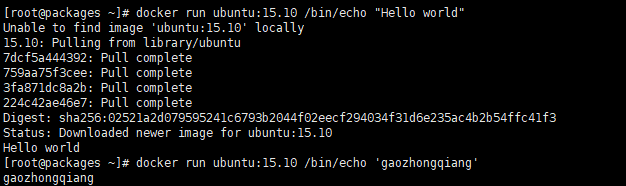
一、基础输出

1、输出hello world：docker run ubuntu:15.10 /bin/echo "Hello world" ------》当没有对应的环境容器会进行下载





参数解析：Docker 以 ubuntu15.10 镜像创建一个新容器，然后在容器里执行 bin/echo "Hello world"，然后输出结果。

a）docker：Docker 的二进制执行文件：

b）run：与前面的 docker 组合来运行一个容器

c）ubuntu:15.10：指定要运行的镜像

d）/bin/echo "Hello world"：在启动的容器里执行的命令

2、运行交互式容器：-i -t

runoob@runoob:~$ docker run -i -t ubuntu:15.10 /bin/bash

root@0123ce188bd8:/#

root@0123ce188bd8:/# cat /proc/version

Linux version 4.4.0-151-generic (buildd@lgw01-amd64-043) (gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.10) ) #178-Ubuntu SMP Tue Jun 11 08:30:22 UTC 2019

root@0123ce188bd8:/# ls

bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

root@0123ce188bd8:/#

root@0123ce188bd8:/# exit

root@runoob:~#

参数解析：

a）-t：在新容器内指定一个伪终端或终端

b）-i：允许你对容器内的标准输入 (STDIN) 进行交互

c）退出：exit、ctrl+D

3、启动容器（后台模式）

runoob@runoob:~$ docker run -d ubuntu:15.10 /bin/sh -c "while true; do echo hello world; sleep 1; done"

2b1b7a428627c51ab8810d541d759f072b4fc75487eed05812646b8534a2fe63 ----》容器id，唯一

1）查看运行的容器：docker ps



runoob@runoob:~$ docker ps

CONTAINER ID IMAGE COMMAND ...

5917eac21c36 ubuntu:15.10 "/bin/sh -c 'while t…" ...

参数解释：

**CONTAINER ID:** 容器 ID、**IMAGE:** 使用的镜像、**COMMAND:** 启动容器时运行的命令、**CREATED:** 容器的创建时间、**STATUS:** 容器状态、**PORTS:** 容器的端口信息和使用的连接类型（tcp\udp）、**NAMES:** 自动分配的容器名称。

status：created（已创建）、restarting（重启中）、running（运行中）、removing（迁移中）、paused（暂停）、exited（停止）、dead（死亡）

2）查看容器内标准输出（查看镜像内容）：docker logs

runoob@runoob:~$ docker logs 2b1b7a428627

3）停止容器：docker stop

二、容器使用

1、查看客户端所有命名：docker

2、详细信息：docker command –help

3、启动容器：docker run -it ubuntu /bin/bash ----》-i：交互式操作、-t：终端

4、退出终端容器：exit

5、启动已停止运行的容器

docker ps -a

docker start b750bbbcfd88 ---》镜像id

6、后台运行：docker run -itd --name ubuntu-test ubuntu /bin/bash ---》 -d：守护进程

7、停止一个容器：docker stop <容器 ID>

8、进入容器：docker attach <容器 ID> docker exec <容器 ID> -----》docker exec -it 243c32535da7 /bin/bash

9、导出容器：docker export <容器 ID> > 打包内容 ----》docker export 1e560fca3906 > ubuntu.tar

10、导入容器快照：

cat docker/ubuntu.tar | docker import - test/ubuntu:v1

指定url或目录导入：docker import http://example.com/exampleimage.tgz example/imagerepo

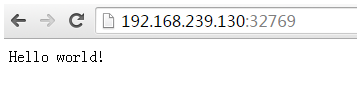
11、删除容器：docker rm <容器 ID> ---》 docker rm -f 1e560fca3906

12、清理所有终止的容器：docker container prune

案例：docker容器中运行一个 Python Flask 应用来运行一个web应用

docker pull training/webapp # 载入镜像

docker run -d -P training/webapp python app.py ----》 -P：将容器内部使用的网络端口映射到我们使用的主机上



设置其他端口：docker run -d -p 5000:5000 training/webapp python app.py

三、网络端口的快捷方式

1、docker port <容器 ID> ----》 可以查看指定 （ID 或者名字）容器的某个确定端口映射到宿主机的端口号

四、查看 WEB 应用程序日志

1、docker logs <容器 ID> ------》参数 -f：实时输出

runoob@runoob:~$ docker logs -f bf08b7f2cd89

\* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)

192.168.239.1 - - [09/May/2016 16:30:37] "GET / HTTP/1.1" 200 -

192.168.239.1 - - [09/May/2016 16:30:37] "GET /favicon.ico HTTP/1.1" 404 –

五、查看WEB应用程序容器的进程

1、docker top  <容器 ID> ----》 来查看容器内部运行的进程

runoob@runoob:~$ docker top wizardly\_chandrasekhar

UID PID PPID ... TIME CMD

root 23245 23228 ... 00:00:00 python app.py

六、检查 WEB 应用程序

1、docker inspect <容器 ID> ----》 它会返回一个 JSON 文件记录着 Docker 容器的配置和状态信息

runoob@runoob:~$ docker inspect wizardly\_chandrasekhar

七、停止\重启\移除web应用容器：docker stop\start\rm <容器 ID>