1. 创建和启动一个容器
2. docker run --detach --name web nginx:latest

Docker 将从 docker hub 上的Nginx 仓库下载, 安装 Nginx:latest 镜像, 然后运行该软件. --detach( 或者使用 -d 缩写形式 )选项, 在后台启用该程序

1. Docker run -d --name mailer dockerinaction/ch2\_mailer

2. 运行交互容器

1. Docker run --interactive --tty \

--link web:web \

--name web\_test

Busybox:latest /bin/sh

该命令使用run命令的两个标志: --interactive(或-i)和--tty(或-t).

1. --interactive 选项告诉docker 保持标准输入流(stdin, 标准输入)对容器开放， 即使容器没有终端连接。
2. --tty选项告诉docker 为容器分配一个虚拟终端， 这将允许你发信号给容器

2.验证是否运行： wget -O - <http://web:80/>

3. docker run -it \

--name agent \

--link web:insideweb \

--line mailer:insidemailer \

Dockerinaction/ch2\_agent

使用缩略标志在交互式容器中启动监控器

3.列举, 停止, 重新启动和查看容器输出

1. Docker ps

会显示每个运行的容器中的一下信息：

* 该容器id
* 使用的镜像
* 容器中执行的命令
* 容器运行的市场
* 容器暴露的网络端口
* 容器名

2.

docker logs web 查看web 服务器运行是否正常

Docker logs mailer 查看邮件服务器运行是否正常

Docker logs agent 查看监控器运行是否正常

Docker stop web 暂停web服务

4.

SQL\_CID = $(docker create -e MYSQL\_ROOT\_PASSWORD=ch2demo mysql:5)

docker start $SQL\_CID

MAILER\_CID = $(docker create dockerinaction/ch2\_mailer)

docker start $MAILER\_CID

WP\_CID = $(docker create --link $SQL\_CID:mysql -p 80 \

-v /run/lock/apache2/ -v /run/apache2/ \

--read-only wordpress:4)

docker start $WP\_CID

AGENT\_CID = $(docker create --link $WP\_CID:insideweb \

--link $MAILER\_CID:insidemailer \

dockerinaction/ch2\_mailer)

docker start $AGENT\_CID

5. 查看容器中那些进程在运行着

docker run -d -p 80:80 --name lamp-test tutum/lamp

docker top lamp-test

1. 在容器内杀死进程
2. 查看容器内的进程

Docker exec lamp-test ps

1. 杀死进程

Docker exec lamp-test kill <PID>