Win10运行docker

基本命令 docker run

/\*

\*docker：Docker的二进制执行文件

\*run：与前面的doker组合来运行一个容器

\*ubuntu:15.10指定要运行的镜像

\*/bin/echo “Hello World”在启动容器里执行的命令

\*/

PS C:\Users\Administrator> docker run ubuntu:15.10 /bin/echo "Hello World"

Unable to find image 'ubuntu:15.10' locally

15.10: Pulling from library/ubuntu

7dcf5a444392: Pull complete 759aa75f3cee: Pull complete 3fa871dc8a2b: Pull complete 224c42ae46e7: Pull complete Digest: sha256:02521a2d079595241c6793b2044f02eecf294034f31d6e235ac4b2b54ffc41f3

Status: Downloaded newer image for ubuntu:15.10

Hello World

/\*

\*-t：在新容器内指定一个伪终端或终端

\*-i：允许你对容器内的标准输入进行交互

\*exit：用来退出容器

\*/

PS C:\Users\Administrator> docker run -i -t ubuntu:15.10 /bin/bash

root@02dac64361a4:/# cat /proc/version

Linux version 4.19.128-microsoft-standard (oe-user@oe-host) (gcc version 8.2.0 (GCC)) #1 SMP Tue Jun 23 12:58:10 UTC 2020

root@02dac64361a4:/# ls

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

root@02dac64361a4:/# exit

exit

PS C:\Users\Administrator> docker run -d ubuntu:15.10 /bin/sh -c "while true;do echo hello world;sleep 1;done"//创建一个以进程方式运行的容器

44e4074132ddd6cb9d9f7483598f5a446043dd04460afb73d8c3e168ff705673//输出容器id

PS C:\Users\Administrator> docker ps//确认容器运行，可以用docker ps查看

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

44e4074132dd ubuntu:15.10 "/bin/sh -c 'while t…" 23 seconds ago Up 22 seconds boring\_dirac

//输出详情介绍

**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。

状态有7种：

* created（已创建）
* restarting（重启中）
* running 或 Up（运行中）
* removing（迁移中）
* paused（暂停）
* exited（停止）
* dead（死亡）
* **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。

PS C:\Users\Administrator> docker logs 44e4074132ddd6cb9d9f7483598f5a446043dd04460afb73d8c3e168ff705673//查看容器内标准输出

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

PS C:\Users\Administrator> docker logs boring\_dirac //可以用容器名代替容器id

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

hello world

docker stop 容器名或容器id来停止容器

PS C:\Users\Administrator> docker ps -a //查看全部容器包括没有运行的

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

44e4074132dd ubuntu:15.10 "/bin/sh -c 'while t…" 14 minutes ago Exited (137) 2 minutes ago boring\_dirac

453330fe45eb ubuntu:15.10 "/bin/bash" 27 minutes ago Exited (0) 15 minutes ago silly\_hoover

02dac64361a4 ubuntu:15.10 "/bin/bash" 31 minutes ago Exited (0) 28 minutes ago gracious\_pike

69f807bcba07 ubuntu:15.10 "/bin/echo 'Hello Wo…" 36 minutes ago Exited (0) 36 minutes ago dazzling\_herschel

caba66036427 hello-world "/hello" 47 minutes ago Exited (0) 47 minutes ago condescending\_agnesi

a1315ff9545f hello-world:latest "/hello" 6 days ago Exited (0) 6 days ago nice\_kare

c82c3665a4a6 hello-world "/hello" 8 days ago Exited (0) 8 days ago compassionate\_golick

c122e3f2e338 hello-world "/hello" 8 days ago Exited (0) 8 days ago amazing\_zhukovsky

PS C:\Users\Administrator> docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

89aeb1f39f05 ubuntu "/bin/bash" 11 minutes ago Up 6 minutes ubuntu-test

f9f4b8304f3c ubuntu "/bin/bash" 14 minutes ago Exited (0) 11 minutes ago beautiful\_booth

44e4074132dd ubuntu:15.10 "/bin/sh -c 'while t…" 31 minutes ago Exited (137) 20 minutes ago boring\_dirac

453330fe45eb ubuntu:15.10 "/bin/bash" 45 minutes ago Exited (0) 33 minutes ago silly\_hoover

02dac64361a4 ubuntu:15.10 "/bin/bash" 49 minutes ago Exited (0) 46 minutes ago gracious\_pike

69f807bcba07 ubuntu:15.10 "/bin/echo 'Hello Wo…" 53 minutes ago Exited (0) 16 minutes ago dazzling\_herschel

caba66036427 hello-world "/hello" About an hour ago Exited (0) About an hour ago condescending\_agnesi

a1315ff9545f hello-world:latest "/hello" 6 days ago Exited (0) 6 days ago nice\_kare

c82c3665a4a6 hello-world "/hello" 8 days ago Exited (0) 8 days ago compassionate\_golick

c122e3f2e338 hello-world "/hello" 8 days ago Exited (0) 8 days ago amazing\_zhukovsky

PS C:\Users\Administrator> docker rm 453330fe45eb//移除一个容器

453330fe45eb

PS C:\Users\Administrator> docker start dazzling\_herschel//启动容器

dazzling\_herschel

PS C:\Users\Administrator> docker stop dazzling\_herschel//停止容器

dazzling\_herschel

PS C:\Users\Administrator> docker run -itd --name ubuntu-test ubuntu /bin/bash//终端后台运行方式启动一个名字是 ubuntu-test 的容器

89aeb1f39f0586c0a8191484ba9565f7f58eaeac7ff4f4b7a8e175e7cea82c5b

PS C:\Users\Administrator> docker restart 89aeb1f39f05重启一个容器

89aeb1f39f05

PS C:\Users\Administrator> docker exec -it 89aeb1f39f05 /bin/bash //使用后台容器的终端标准输入，使用exec之后会进入容器 exit退出不会停止容器，会在后台运行

root@89aeb1f39f05:/# exit

exit

PS C:\Users\Administrator> docker export 89aeb1f39f05 > ubuntu.tar

//导出一个容器到tar文件

PS C:\Users\Administrator> docker pull training/webapp //拉取一个镜像

Using default tag: latest

latest: Pulling from training/webapp

Image docker.io/training/webapp:latest uses outdated schema1 manifest format. Please upgrade to a schema2 image for better future compatibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/

e190868d63f8: Already exists 909cd34c6fd7: Already exists 0b9bfabab7c1: Already exists a3ed95caeb02: Already exists 10bbbc0fc0ff: Already exists fca59b508e9f: Already exists e7ae2541b15b: Already exists 9dd97ef58ce9: Already exists a4c1b0cb7af7: Already exists Digest: sha256:06e9c1983bd6d5db5fba376ccd63bfa529e8d02f23d5079b8f74a616308fb11d

Status: Image is up to date for training/webapp:latest

docker.io/training/webapp:latest

PS C:\Users\Administrator> docker images //查看当前镜像

REPOSITORY TAG IMAGE ID CREATED SIZE

ubuntu latest f63181f19b2f 7 days ago 72.9MB

nginx latest f6d0b4767a6c 2 weeks ago 133MB

hello-world latest bf756fb1ae65 13 months ago 13.3kB

ubuntu 15.10 9b9cb95443b5 4 years ago 137MB

training/webapp latest 6fae60ef3446 5 years ago 349MB

PS C:\Users\Administrator> docker run -d -P training/webapp python app.py //运行该拉取镜像使用-d后台运行 -P 端口随机映射 python app.py是运行命令

495f5faba832c9a43db38f2f7414e8d355012da9ea763b109d75ffa03b08a176

PS C:\Users\Administrator> docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

495f5faba832 training/webapp "python app.py" 8 seconds ago Up 8 seconds 0.0.0.0:49153->5000/tcp blissful\_aryabhata

89aeb1f39f05 ubuntu "/bin/bash" 38 minutes ago Up 33 minutes ubuntu-test

之后再浏览器上输入127.0.0.1:49153就可以访问该页面

PS C:\Users\Administrator> docker run -d -p 5000:5000 training/webapp python app.py

//运行镜像training/webapp 运行命令是python app.py –p 远程端口5000映射到本地端口5000 -d后台方式运行

946006efc0661d71bcbceda075f4b919b555139d3761ed59d8394c77df35b28e

PS C:\Users\Administrator> docker ps //查看运行容器

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

946006efc066 training/webapp "python app.py" 7 seconds ago Up 7 seconds 0.0.0.0:5000->5000/tcp affectionate\_solomon

495f5faba832 training/webapp "python app.py" 3 minutes ago Up 3 minutes 0.0.0.0:49153->5000/tcp blissful\_aryabhata

89aeb1f39f05 ubuntu "/bin/bash" 41 minutes ago Up 37 minutes ubuntu-test

之后再浏览器上输入127.0.0.1:5000就可以访问该页面

PS C:\Users\Administrator> docker port 495f5faba832 //可以专门用来查询映射端口的命令，使用容器id

5000/tcp -> 0.0.0.0:49153

PS C:\Users\Administrator> docker port blissful\_aryabhata //可以专门用来查询映射端口的命令，使用容器名

5000/tcp -> 0.0.0.0:49153

PS C:\Users\Administrator> docker top blissful\_aryabhata //查看web应用程序容器的进程

UID PID PPID C STIME TTY TIME CMD

root 2866 2844 0 02:41 ? 00:00:00 python app.py

PS C:\Users\Administrator> docker inspect blissful\_aryabhata //查看docker底层信息，返回json文件记录着docker容器的配置和状态信息

[

{

"Id": "495f5faba832c9a43db38f2f7414e8d355012da9ea763b109d75ffa03b08a176",

"Created": "2021-01-29T02:41:59.1216683Z",

"Path": "python",

"Args": [

"app.py"

],

"State": {

"Status": "running",

"Running": true,

"Paused": false,

"Restarting": false,

"OOMKilled": false,

"Dead": false,

"Pid": 2866,

"ExitCode": 0,

"Error": "",

"StartedAt": "2021-01-29T02:41:59.4539449Z",

"FinishedAt": "0001-01-01T00:00:00Z"

},

"Image": "sha256:6fae60ef344644649a39240b94d73b8ba9c67f898ede85cf8e947a887b3e6557",

"ResolvConfPath": "/var/lib/docker/containers/495f5faba832c9a43db38f2f7414e8d355012da9ea763b109d75ffa03b08a176/resolv.conf",

"HostnamePath": "/var/lib/docker/containers/495f5faba832c9a43db38f2f7414e8d355012da9ea763b109d75ffa03b08a176/hostname",

"HostsPath": "/var/lib/docker/containers/495f5faba832c9a43db38f2f7414e8d355012da9ea763b109d75ffa03b08a176/hosts",

"LogPath": "/var/lib/docker/containers/495f5faba832c9a43db38f2f7414e8d355012da9ea763b109d75ffa03b08a176/495f5faba832c9a43db38f2f7414e8d355012da9ea763b109d75ffa03b08a176-json.log",

"Name": "/blissful\_aryabhata",

"RestartCount": 0,

"Driver": "overlay2",

"Platform": "linux",

"MountLabel": "",

"ProcessLabel": "",

"AppArmorProfile": "",

"ExecIDs": null,

"HostConfig": {

"Binds": null,

"ContainerIDFile": "",

"LogConfig": {

"Type": "json-file",

"Config": {}

},

"NetworkMode": "default",

"PortBindings": {},

"RestartPolicy": {

"Name": "no",

"MaximumRetryCount": 0

},

"AutoRemove": false,

"VolumeDriver": "",

"VolumesFrom": null,

"CapAdd": null,

"CapDrop": null,

"CgroupnsMode": "host",

"Dns": [],

"DnsOptions": [],

"DnsSearch": [],

"ExtraHosts": null,

"GroupAdd": null,

"IpcMode": "private",

"Cgroup": "",

"Links": null,

"OomScoreAdj": 0,

"PidMode": "",

"Privileged": false,

"PublishAllPorts": true,

"ReadonlyRootfs": false,

"SecurityOpt": null,

"UTSMode": "",

"UsernsMode": "",

"ShmSize": 67108864,

"Runtime": "runc",

"ConsoleSize": [

50,

120

],

"Isolation": "",

"CpuShares": 0,

"Memory": 0,

"NanoCpus": 0,

"CgroupParent": "",

"BlkioWeight": 0,

"BlkioWeightDevice": [],

"BlkioDeviceReadBps": null,

"BlkioDeviceWriteBps": null,

"BlkioDeviceReadIOps": null,

"BlkioDeviceWriteIOps": null,

"CpuPeriod": 0,

"CpuQuota": 0,

"CpuRealtimePeriod": 0,

"CpuRealtimeRuntime": 0,

"CpusetCpus": "",

"CpusetMems": "",

"Devices": [],

"DeviceCgroupRules": null,

"DeviceRequests": null,

"KernelMemory": 0,

"KernelMemoryTCP": 0,

"MemoryReservation": 0,

"MemorySwap": 0,

"MemorySwappiness": null,

"OomKillDisable": false,

"PidsLimit": null,

"Ulimits": null,

"CpuCount": 0,

"CpuPercent": 0,

"IOMaximumIOps": 0,

"IOMaximumBandwidth": 0,

"MaskedPaths": [

"/proc/asound",

"/proc/acpi",

"/proc/kcore",

"/proc/keys",

"/proc/latency\_stats",

"/proc/timer\_list",

"/proc/timer\_stats",

"/proc/sched\_debug",

"/proc/scsi",

"/sys/firmware"

],

"ReadonlyPaths": [

"/proc/bus",

"/proc/fs",

"/proc/irq",

"/proc/sys",

"/proc/sysrq-trigger"

]

},

"GraphDriver": {

"Data": {

"LowerDir": "/var/lib/docker/overlay2/92673703e8d4fb4dfab621d6c35a7d64d5003fddc9639a18f021dbcf960bb53a-init/diff:/var/lib/docker/overlay2/1b6592f4ab1e7d312041ca8f0ccf08010b2cc950997ab91490b6419a5ffcb85e/diff:/var/lib/docker/overlay2/5b1231e14cda67d90a92236e403a45c72b2df0b94bdcc42108edccae67d7c69d/diff:/var/lib/docker/overlay2/2a9fe85681609842ec06a00a3fbf98100ea0dd40da9744723333d1a4802c219a/diff:/var/lib/docker/overlay2/2e2a660e74794ce53d8090715b79b93270b2cf3dfd6db2f2d01540ba11f7a5fb/diff:/var/lib/docker/overlay2/a40f00739af9477fe423709aaf6c22bbd905a2aee6955d8e09d4ddb4c0e6cd1d/diff:/var/lib/docker/overlay2/12d59951cb8dd32678f558385db94c0945f48ac03fd3494f14ef48367f8f0a56/diff:/var/lib/docker/overlay2/6764cc349c36cc5edc6afeeed7ae6d462b591d444e941e49601e07d582803d69/diff:/var/lib/docker/overlay2/e250c1866eab432e84d1ef37d22c2e71fffed34ba6130520ad4eab93f24d0fd0/diff:/var/lib/docker/overlay2/81bfa66767a13c6818f0f3d6729dfbe3ecc5c0cace945aaaf4d26d8a89ce0cae/diff:/var/lib/docker/overlay2/2bf2b5d84a43382338527afaffd4d29b47a7305007601e7093881ab87cec8d4e/diff:/var/lib/docker/overlay2/f8e990b88efc347185d9e3f435629323e25598118b4a6d6c2e29ed86c482136f/diff:/var/lib/docker/overlay2/2203d2c328b999498400d0af5971e22b7894ce219c00faa69debe2c3b567964b/diff:/var/lib/docker/overlay2/86162ab7b6a4b3bb1f1b531ac27a50a8b97bb7b64189ab0008569642862ade57/diff",

"MergedDir": "/var/lib/docker/overlay2/92673703e8d4fb4dfab621d6c35a7d64d5003fddc9639a18f021dbcf960bb53a/merged",

"UpperDir": "/var/lib/docker/overlay2/92673703e8d4fb4dfab621d6c35a7d64d5003fddc9639a18f021dbcf960bb53a/diff",

"WorkDir": "/var/lib/docker/overlay2/92673703e8d4fb4dfab621d6c35a7d64d5003fddc9639a18f021dbcf960bb53a/work"

},

"Name": "overlay2"

},

"Mounts": [],

"Config": {

"Hostname": "495f5faba832",

"Domainname": "",

"User": "",

"AttachStdin": false,

"AttachStdout": false,

"AttachStderr": false,

"ExposedPorts": {

"5000/tcp": {}

},

"Tty": false,

"OpenStdin": false,

"StdinOnce": false,

"Env": [

"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

],

"Cmd": [

"python",

"app.py"

],

"Image": "training/webapp",

"Volumes": null,

"WorkingDir": "/opt/webapp",

"Entrypoint": null,

"OnBuild": null,

"Labels": {}

},

"NetworkSettings": {

"Bridge": "",

"SandboxID": "5ca51a183a21a3b5fe0c12fb6384ba61fb4b73bec1d96a3e7a644e199785e1ee",

"HairpinMode": false,

"LinkLocalIPv6Address": "",

"LinkLocalIPv6PrefixLen": 0,

"Ports": {

"5000/tcp": [

{

"HostIp": "0.0.0.0",

"HostPort": "49153"

}

]

},

"SandboxKey": "/var/run/docker/netns/5ca51a183a21",

"SecondaryIPAddresses": null,

"SecondaryIPv6Addresses": null,

"EndpointID": "1f18e05e948b1192111e752ac670e64aa4c1dcd921109642751a94f781303e36",

"Gateway": "172.17.0.1",

"GlobalIPv6Address": "",

"GlobalIPv6PrefixLen": 0,

"IPAddress": "172.17.0.3",

"IPPrefixLen": 16,

"IPv6Gateway": "",

"MacAddress": "02:42:ac:11:00:03",

"Networks": {

"bridge": {

"IPAMConfig": null,

"Links": null,

"Aliases": null,

"NetworkID": "cdd13d6bb8a1991e40d91c910ec77a260b8194c3e7f31c9b1780bf943f7f6931",

"EndpointID": "1f18e05e948b1192111e752ac670e64aa4c1dcd921109642751a94f781303e36",

"Gateway": "172.17.0.1",

"IPAddress": "172.17.0.3",

"IPPrefixLen": 16,

"IPv6Gateway": "",

"GlobalIPv6Address": "",

"GlobalIPv6PrefixLen": 0,

"MacAddress": "02:42:ac:11:00:03",

"DriverOpts": null

}

}

}

}

]

PS C:\Users\Administrator> docker ps //查看运行容器

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

946006efc066 training/webapp "python app.py" 16 minutes ago Up 16 minutes 0.0.0.0:5000->5000/tcp affectionate\_solomon

495f5faba832 training/webapp "python app.py" 19 minutes ago Up 19 minutes 0.0.0.0:49153->5000/tcp blissful\_aryabhata

89aeb1f39f05 ubuntu "/bin/bash" 57 minutes ago Up 53 minutes ubuntu-test

PS C:\Users\Administrator> docker stop blissful\_aryabhata //停止容器运行

blissful\_aryabhata

PS C:\Users\Administrator> docker start blissful\_aryabhata /启动容器运行

blissful\_aryabhata

PS C:\Users\Administrator> docker ps -l //查看最后一次创建的容器

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

946006efc066 training/webapp "python app.py" 17 minutes ago Up 17 minutes 0.0.0.0:5000->5000/tcp affectionate\_solomon

PS C:\Users\Administrator> docker ps //查看运行的容器

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

946006efc066 training/webapp "python app.py" 20 minutes ago Up 20 minutes 0.0.0.0:5000->5000/tcp affectionate\_solomon

495f5faba832 training/webapp "python app.py" 23 minutes ago Up 40 seconds 0.0.0.0:49155->5000/tcp blissful\_aryabhata

89aeb1f39f05 ubuntu "/bin/bash" About an hour ago Up 57 minutes ubuntu-test

PS C:\Users\Administrator> docker stop blissful\_aryabhata //要移除容器要先停止，这里用容器名来停止

blissful\_aryabhata

PS C:\Users\Administrator> docker rm blissful\_aryabhata //可以移除了

blissful\_aryabhata

PS C:\Users\Administrator> docker rm affectionate\_solomon //如果直接移除正在运行的容器会报如下错误

Error response from daemon: You cannot remove a running container 946006efc0661d71bcbceda075f4b919b555139d3761ed59d8394c77df35b28e. Stop the container before attempting removal or force remove

PS C:\Users\Administrator> docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

/\*镜像的仓库源 镜像版本 镜像id 镜像创建时间 镜像大小\*/

ubuntu latest f63181f19b2f 8 days ago 72.9MB

nginx latest f6d0b4767a6c 2 weeks ago 133MB

hello-world latest bf756fb1ae65 13 months ago 13.3kB

ubuntu 15.10 9b9cb95443b5 4 years ago 137MB

training/webapp latest 6fae60ef3446 5 years ago 349MB

PS C:\Users\Administrator> docker run -t -i ubuntu:15.10 /bin/bash

//如果要使用版本为15.10的ubuntu系统镜像来运行容器时，使用本命令 -I 标准输入输出 -t终端，如果不制定版本标签，那么将默认使用ubuntu:latest

root@dd5da157eeac:/#

PS C:\Users\Administrator> docker pull ubuntu:13.10 ；

//拉取一个版本号为13.10的ubuntu镜像

13.10: Pulling from library/ubuntu

Image docker.io/library/ubuntu:13.10 uses outdated schema1 manifest format. Please upgrade to a schema2 image for better future compatibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/

a3ed95caeb02: Pull complete 0d8710fc57fd: Pull complete 5037c5cd623d: Pull complete 83b53423b49f: Pull complete e9e8bd3b94ab: Pull complete 7db00e6b6e5e: Pull complete Digest: sha256:403105e61e2d540187da20d837b6a6e92efc3eb4337da9c04c191fb5e28c44dc

Status: Downloaded newer image for ubuntu:13.10

docker.io/library/ubuntu:13.10

PS C:\Users\Administrator> docker search httpd //查找镜像

NAME DESCRIPTION STARS OFFICIAL AUTOMATED

//镜像仓库源的名字 镜像仓库的描述

Start点赞数 是否docker官方发布 自动构建

httpd The Apache HTTP Server Project 3333 [OK]

centos/httpd-24-centos7 Platform for running Apache httpd 2.4 or bui… 36

centos/httpd 33 [OK]

arm32v7/httpd The Apache HTTP Server Project 9

polinux/httpd-php Apache with PHP in Docker (Supervisor, CentO… 4 [OK]

salim1983hoop/httpd24 Dockerfile running apache config 2 [OK]

clearlinux/httpd httpd HyperText Transfer Protocol (HTTP) ser… 1

publici/httpd httpd:latest 1 [OK]

solsson/httpd-openidc mod\_auth\_openidc on official httpd image, ve… 1 [OK]

hypoport/httpd-cgi httpd-cgi 1 [OK]

inanimate/httpd-ssl A play container with httpd, ssl enabled, an… 1 [OK]

dariko/httpd-rproxy-ldap Apache httpd reverse proxy with LDAP authent… 1 [OK]

jonathanheilmann/httpd-alpine-rewrite httpd:alpine with enabled mod\_rewrite 1 [OK]

lead4good/httpd-fpm httpd server which connects via fcgi proxy h… 1 [OK]

interlutions/httpd httpd docker image with debian-based config … 0 [OK]

appertly/httpd Customized Apache HTTPD that uses a PHP-FPM … 0 [OK]

amd64/httpd The Apache HTTP Server Project 0

manasip/httpd 0

manageiq/httpd Container with httpd, built on CentOS for Ma… 0 [OK]

trollin/httpd 0

e2eteam/httpd 0

manageiq/httpd\_configmap\_generator Httpd Configmap Generator 0 [OK]

ysli/httpd Httpd for DeepWeb 0 [OK]

itsziget/httpd24 Extended HTTPD Docker image based on the off… 0 [OK]

dockerpinata/httpd 0

PS C:\Users\Administrator> docker pull httpd //拉取镜像

Using default tag: latest

latest: Pulling from library/httpd

a076a628af6f: Already exists e444656f7792: Pull complete 0ec35e191b09: Pull complete 4aad5d8db1a6: Pull complete eb1da3ea630f: Pull complete Digest: sha256:2fab99fb3b1c7ddfa99d7dc55de8dad0a62dbe3e7c605d78ecbdf2c6c49fd636

Status: Downloaded newer image for httpd:latest

docker.io/library/httpd:latest

PS C:\Users\Administrator> docker run httpd //使用镜像

AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message

AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message

[Fri Jan 29 03:27:59.088096 2021] [mpm\_event:notice] [pid 1:tid 140392719611008] AH00489: Apache/2.4.46 (Unix) configured -- resuming normal operations

[Fri Jan 29 03:27:59.094144 2021] [core:notice] [pid 1:tid 140392719611008] AH00094: Command line: 'httpd -D FOREGROUND'

PS C:\Users\Administrator> docker rmi hello-world

//移除镜像，在移除镜像之前需要把改镜像创建的容器移除

Error response from daemon: conflict: unable to remove repository reference "hello-world" (must force) - container c82c3665a4a6 is using its referenced image bf756fb1ae65

PS C:\Users\Administrator> docker rm c82c3665a4a6 //移除容器c82c3665a4a6

c82c3665a4a6

PS C:\Users\Administrator> docker rm c122e3f2e338 //移除容器c122e3f2e338

c122e3f2e338

PS C:\Users\Administrator> docker rmi hello-world

//移除镜像，提示还用容器caba66036427没移除

Error response from daemon: conflict: unable to remove repository reference "hello-world" (must force) - container caba66036427 is using its referenced image bf756fb1ae65

PS C:\Users\Administrator> docker rm caba66036427 //移除容器caba66036427

caba66036427

PS C:\Users\Administrator> docker rmi hello-world //移除镜像成功

Untagged: hello-world:latest

Untagged: hello-world@sha256:31b9c7d48790f0d8c50ab433d9c3b7e17666d6993084c002c2ff1ca09b96391d

Deleted: sha256:bf756fb1ae65adf866bd8c456593cd24beb6a0a061dedf42b26a993176745f6b

Deleted: sha256:9c27e219663c25e0f28493790cc0b88bc973ba3b1686355f221c38a36978ac63

**创建镜像**

当我们从 docker 镜像仓库中下载的镜像不能满足我们的需求时，我们可以通过以下两种方式对镜像进行更改。

* 1、从已经创建的容器中更新镜像，并且提交这个镜像
* 2、使用 Dockerfile 指令来创建一个新的镜像

PS C:\Users\Administrator> docker run -t -i ubuntu:15.10 /bin/bash //运行一个容器-t终端 -I 交互 镜像 ubuntu:15.10

root@4d8c93539a6a:/# apt-get update //在容器内部进行更新

Ign http://archive.ubuntu.com wily InRelease

Ign http://archive.ubuntu.com wily-updates InRelease

Ign http://archive.ubuntu.com wily-security InRelease

Ign http://archive.ubuntu.com wily Release.gpg

Ign http://archive.ubuntu.com wily-updates Release.gpg

Ign http://archive.ubuntu.com wily-security Release.gpg

Ign http://archive.ubuntu.com wily Release

Ign http://archive.ubuntu.com wily-updates Release

Ign http://archive.ubuntu.com wily-security Release

Err http://archive.ubuntu.com wily/main Sources

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily/restricted Sources

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily/universe Sources

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily/main amd64 Packages

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily/restricted amd64 Packages

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily/universe amd64 Packages

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily-updates/main Sources

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily-updates/restricted Sources

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily-updates/universe Sources

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily-updates/main amd64 Packages

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily-updates/restricted amd64 Packages

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily-updates/universe amd64 Packages

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily-security/main Sources

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily-security/restricted Sources

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily-security/universe Sources

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily-security/main amd64 Packages

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily-security/restricted amd64 Packages

404 Not Found [IP: 91.189.88.142 80]

Err http://archive.ubuntu.com wily-security/universe amd64 Packages

404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily/main/source/Sources 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily/restricted/source/Sources 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily/universe/source/Sources 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily/main/binary-amd64/Packages 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily/restricted/binary-amd64/Packages 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily/universe/binary-amd64/Packages 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily-updates/main/source/Sources 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily-updates/restricted/source/Sources 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily-updates/universe/source/Sources 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily-updates/main/binary-amd64/Packages 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily-updates/restricted/binary-amd64/Packages 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily-updates/universe/binary-amd64/Packages 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily-security/main/source/Sources 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily-security/restricted/source/Sources 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily-security/universe/source/Sources 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily-security/main/binary-amd64/Packages 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily-security/restricted/binary-amd64/Packages 404 Not Found [IP: 91.189.88.142 80]

W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/wily-security/universe/binary-amd64/Packages 404 Not Found [IP: 91.189.88.142 80]

E: Some index files failed to download. They have been ignored, or old ones used instead.

root@4d8c93539a6a:/# exit //退出容器

exit

PS C:\Users\Administrator> docker ps //查看运行的容器

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

946006efc066 training/webapp "python app.py" 54 minutes ago Up 54 minutes 0.0.0.0:5000->5000/tcp affectionate\_solomon

89aeb1f39f05 ubuntu "/bin/bash" 2 hours ago Up 2 hours ubuntu-test

PS C:\Users\Administrator> docker commit -m="has update" -a="shuhan" 4d8c93539a6a shuhan/ubuntu:v2 //将刚刚更新的容器上传成新的镜像，commit是提交 -m是描述信息 -a是作者 然后是容器id 最后是容器名字 冒号后边是版本号

sha256:a3f3f32fd39562e3d603aae6d4696a1a7595b2ead0fca78c6ed60b08d8c416fe

PS C:\Users\Administrator> docker images //查看镜像

REPOSITORY TAG IMAGE ID CREATED SIZE

shuhan/ubuntu v2 a3f3f32fd395 18 seconds ago 137MB

ubuntu latest f63181f19b2f 8 days ago 72.9MB

nginx latest f6d0b4767a6c 2 weeks ago 133MB

httpd latest 683a7aad17d3 2 weeks ago 138MB

ubuntu 15.10 9b9cb95443b5 4 years ago 137MB

training/webapp latest 6fae60ef3446 5 years ago 349MB

ubuntu 13.10 7f020f7bf345 6 years ago 185MB

PS C:\Users\Administrator> docker run -t -i shuhan/ubuntu:v2 /bin/bash //从刚刚的新镜像创建一个容器，终端交互运行

root@5d457f10bc41:/#

### 构建镜像

我们使用命令 **docker build** ， 从零开始来创建一个新的镜像。为此，我们需要创建一个 Dockerfile 文件，其中包含一组指令来告诉 Docker 如何构建我们的镜像。

runoob@runoob:~$ cat Dockerfile

FROM centos:6.7

MAINTAINER Fisher "fisher@sudops.com"

RUN /bin/echo 'root:123456' |chpasswd

RUN useradd runoob

RUN /bin/echo 'runoob:123456' |chpasswd

RUN /bin/echo -e "LANG=\"en\_US.UTF-8\"" >/etc/default/local

EXPOSE 22

EXPOSE 80

CMD /usr/sbin/sshd -D

每一个指令都会在镜像上创建一个新的层，每一个指令的前缀都必须是大写的。

第一条FROM，指定使用哪个镜像源

RUN 指令告诉docker 在镜像内执行命令，安装了什么。。。

然后，我们使用 Dockerfile 文件，通过 docker build 命令来构建一个镜像。

runoob@runoob:~$ docker build -t runoob/centos:6.7 .

Sending build context to Docker daemon 17.92 kB

Step 1 : FROM centos:6.7

---&gt; d95b5ca17cc3

Step 2 : MAINTAINER Fisher "fisher@sudops.com"

---&gt; Using cache

---&gt; 0c92299c6f03

Step 3 : RUN /bin/echo 'root:123456' |chpasswd

---&gt; Using cache

---&gt; 0397ce2fbd0a

Step 4 : RUN useradd runoob

......

参数说明：

* **-t** ：指定要创建的目标镜像名
* **.** ：Dockerfile 文件所在目录，可以指定Dockerfile 的绝对路径

使用docker images 查看创建的镜像已经在列表中存在,镜像ID为860c279d2fec

runoob@runoob:~$ docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

runoob/centos 6.7 860c279d2fec About a minute ago 190.6 MB

runoob/ubuntu v2 70bf1840fd7c 17 hours ago 158.5 MB

ubuntu 14.04 90d5884b1ee0 6 days ago 188 MB

php 5.6 f40e9e0f10c8 10 days ago 444.8 MB

nginx latest 6f8d099c3adc 12 days ago 182.7 MB

mysql 5.6 f2e8d6c772c0 3 weeks ago 324.6 MB

httpd latest 02ef73cf1bc0 3 weeks ago 194.4 MB

ubuntu 15.10 4e3b13c8a266 5 weeks ago 136.3 MB

hello-world latest 690ed74de00f 6 months ago 960 B

centos 6.7 d95b5ca17cc3 6 months ago 190.6 MB

training/webapp latest 6fae60ef3446 12 months ago 348.8 MB

我们可以使用新的镜像来创建容器

runoob@runoob:~$ docker run -t -i runoob/centos:6.7 /bin/bash

[root@41c28d18b5fb /]# id runoob

uid=500(runoob) gid=500(runoob) groups=500(runoob)

从上面看到新镜像已经包含我们创建的用户 runoob。

PS C:\Users\Administrator> docker tag a3f3f32fd395 shuhan/ubuntu:v3

//设置镜像标签，把id是a3f3f32fd395 的镜像标签设置为v3

PS C:\Users\Administrator> docker images //查看镜像

REPOSITORY TAG IMAGE ID CREATED SIZE

shuhan/ubuntu v2 a3f3f32fd395 13 minutes ago 137MB

shuhan/ubuntu v3 a3f3f32fd395 13 minutes ago 137MB

ubuntu latest f63181f19b2f 8 days ago 72.9MB

nginx latest f6d0b4767a6c 2 weeks ago 133MB

httpd latest 683a7aad17d3 2 weeks ago 138MB

ubuntu 15.10 9b9cb95443b5 4 years ago 137MB

training/webapp latest 6fae60ef3446 5 years ago 349MB

ubuntu 13.10 7f020f7bf345 6 years ago 185MB

PS C:\Users\Administrator> docker run -d -P training/webapp python app.py

//启动一个容器 后台启动-d 随机映射一个主机端口-P 容器执行命令 python app.py

31bfb63fa61f3a64652ae68a5cdfb563fc9fa18f736595927e31eb7f639ee7ff

PS C:\Users\Administrator> docker ps //查看运行容器

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

31bfb63fa61f training/webapp "python app.py" 6 seconds ago Up 5 seconds 0.0.0.0:49156->5000/tcp mystifying\_johnson

89aeb1f39f05 ubuntu "/bin/bash" 4 hours ago Up 4 hours ubuntu-test

PS C:\Users\Administrator> docker run -d -p 5000:5000 training/webapp python app.py

//启动容器 -d后台运行 -p 指定映射端口 镜像training/webapp 容器运行命令python app.py

6e63f8f2c987406a1b0f349e7c56e683804d64b4077d662e2a27334bc75b599d

PS C:\Users\Administrator> docker ps //查看容器运行

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

6e63f8f2c987 training/webapp "python app.py" 6 seconds ago Up 5 seconds 0.0.0.0:5000->5000/tcp reverent\_nightingale

31bfb63fa61f training/webapp "python app.py" About a minute ago Up About a minute 0.0.0.0:49156->5000/tcp mystifying\_johnson

89aeb1f39f05 ubuntu "/bin/bash" 4 hours ago Up 4 hours ubuntu-test

PS C:\Users\Administrator> docker run -d -p 127.0.0.1:5001:5000 training/webapp python app.py //容器启动 -d后台运行 –p指定映射的ip端口从远程端口5000，镜像training/webapp 命令python app.py 默认绑定tcp端口

6b81ab0f557ed4799ef576af7be2d382f2f1de2e929ac0f80e56fc39b54b6694

PS C:\Users\Administrator> docker ps //查看运行容器

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

6b81ab0f557e training/webapp "python app.py" 7 seconds ago Up 7 seconds 127.0.0.1:5001->5000/tcp trusting\_mendeleev

6e63f8f2c987 training/webapp "python app.py" 4 minutes ago Up 4 minutes 0.0.0.0:5000->5000/tcp reverent\_nightingale

31bfb63fa61f training/webapp "python app.py" 6 minutes ago Up 6 minutes 0.0.0.0:49156->5000/tcp mystifying\_johnson

89aeb1f39f05 ubuntu "/bin/bash" 4 hours ago Up 4 hours ubuntu-test

PS C:\Users\Administrator> docker run -d -p 127.0.0.1:5000:5000/udp training/webapp python app.py //启动容器 -d后台运行 -p指定端口 /udp绑定udp端口 training/webapp镜像 python app..py 容器命令

c13a427a36f30bbe9319552251ecb03dea89d729a6e19eb53374b4f87cf240f5

PS C:\Users\Administrator> docker ps //查看运行容器

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

c13a427a36f3 training/webapp "python app.py" 3 seconds ago Up 2 seconds 5000/tcp, 127.0.0.1:5000->5000/udp strange\_hawking

6b81ab0f557e training/webapp "python app.py" 3 minutes ago Up 3 minutes 127.0.0.1:5001->5000/tcp trusting\_mendeleev

6e63f8f2c987 training/webapp "python app.py" 7 minutes ago Up 7 minutes 0.0.0.0:5000->5000/tcp reverent\_nightingale

31bfb63fa61f training/webapp "python app.py" 9 minutes ago Up 9 minutes 0.0.0.0:49156->5000/tcp mystifying\_johnson

89aeb1f39f05 ubuntu "/bin/bash" 4 hours ago Up 4 hours ubuntu-test

PS C:\Users\Administrator> docker ps //查看运行容器

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

c13a427a36f3 training/webapp "python app.py" 3 seconds ago Up 2 seconds 5000/tcp, 127.0.0.1:5000->5000/udp strange\_hawking

6b81ab0f557e training/webapp "python app.py" 3 minutes ago Up 3 minutes 127.0.0.1:5001->5000/tcp trusting\_mendeleev

6e63f8f2c987 training/webapp "python app.py" 7 minutes ago Up 7 minutes 0.0.0.0:5000->5000/tcp reverent\_nightingale

31bfb63fa61f training/webapp "python app.py" 9 minutes ago Up 9 minutes 0.0.0.0:49156->5000/tcp mystifying\_johnson

89aeb1f39f05 ubuntu "/bin/bash" 4 hours ago Up 4 hours ubuntu-test

PS C:\Users\Administrator> docker port c13a427a36f3 查看容器端口

5000/udp -> 127.0.0.1:5000

PS C:\Users\Administrator> docker port 6b81ab0f557e 5000 查看容器端口

127.0.0.1:5001

PS C:\Users\Administrator> docker run -d -P --name shuhan training/webapp python app.py

//运行容器 -d后台运行 -P随机映射主机端口 –name为容器指定名字 training/webapp镜像 python app.py容器内运行命令

bde3faffb2e0dffc25ce079f2b5c8369eb4f80ac67804f5cf0e2caaffe56120d

PS C:\Users\Administrator> docker ps -l //查看最新启动的容器

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

bde3faffb2e0 training/webapp "python app.py" 9 seconds ago Up 8 seconds 0.0.0.0:49157->5000/tcp shuhan

PS C:\Users\Administrator> docker network create -d bridge test-net

//创建一个新的docker网络 create创建 -d后台 bridge模式 test-net是名字

6d27271587e9cab0c087df3aa1d8db301845b47de1bcf6c984dd98057ce547e4

PS C:\Users\Administrator> docker network ls

//查看当前的network

NETWORK ID NAME DRIVER SCOPE

cdd13d6bb8a1 bridge bridge local

b710b103e552 host host local

634c3161ec3f none null local

6d27271587e9 test-net bridge local

创建一个带ping功能的镜像，利用这个镜像创建2个容器链接到刚刚新建的network，用ping测试链接

PS C:\Users\Administrator> docker run -it ubuntu /bin/bash

//以终端运行ubuntu 命令/bin/bash

root@4e83dd1a8d6b:/# apt-get update

//更新

root@4e83dd1a8d6b:/# apt install iputils-ping

//安装ping

PS C:\Users\Administrator> docker commit -m="has installed ping" -a="shuhan" 4e83dd1a8d6b shuhan/ubuntu:ping

//提交这个安装了ping的容器到镜像中 -m描述 -a作者 容器id 镜像仓库名：标签

PS C:\Users\Administrator> docker run -itd --name test1 --network test-net shuhan/ubuntu:ping /bin/bash

//用刚刚安装了ping的镜像启动一个容器 后台终端运行 名字是test1 网络是 test-net

镜像仓库名：标签 运行命令/bin/bash

9ceed673ee34989e4dcef9f71f30da4375d73bd67edc905cf21991ca78b1285c

PS C:\Users\Administrator> docker run -itd --name test2 --network test-net shuhan/ubuntu:ping /bin/bash

//用刚刚安装了ping的镜像启动一个容器 后台终端运行 名字是test2 网络是 test-net

镜像仓库名：标签 运行命令/bin/bash

4cf02ea31e284809d1d8e28adbdcbe5e64b5d254c5d656e13129d153a03828ba

PS C:\Users\Administrator> docker exec -it test1 /bin/bash

//进入容器运行

root@9ceed673ee34:/# ping test2

//查看链接ping值

PING test2 (172.18.0.3) 56(84) bytes of data.

64 bytes from test2.test-net (172.18.0.3): icmp\_seq=1 ttl=64 time=0.242 ms

64 bytes from test2.test-net (172.18.0.3): icmp\_seq=2 ttl=64 time=0.120 ms

64 bytes from test2.test-net (172.18.0.3): icmp\_seq=3 ttl=64 time=0.040 ms

64 bytes from test2.test-net (172.18.0.3): icmp\_seq=4 ttl=64 time=0.110 ms

64 bytes from test2.test-net (172.18.0.3): icmp\_seq=5 ttl=64 time=0.058 ms

PS C:\Users\Administrator> docker exec -it test2 /bin/bash

//进入容器运行

root@4cf02ea31e28:/# ping test1

//查看链接ping值

PING test1 (172.18.0.2) 56(84) bytes of data.

64 bytes from test1.test-net (172.18.0.2): icmp\_seq=1 ttl=64 time=0.166 ms

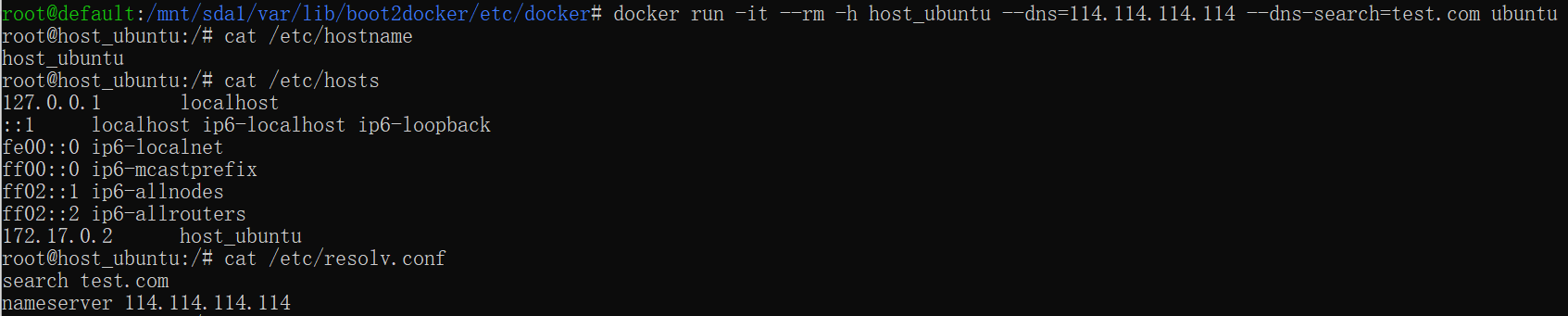
64 bytes from test1.test-net (172.18.0.2): icmp\_seq=2 ttl=64 time=0.038 ms

64 bytes from test1.test-net (172.18.0.2): icmp\_seq=3 ttl=64 time=0.116 ms

64 bytes from test1.test-net (172.18.0.2): icmp\_seq=4 ttl=64 time=0.093 ms

64 bytes from test1.test-net (172.18.0.2): icmp\_seq=5 ttl=64 time=0.039 ms

**配置 DNS**

[](https://www.runoob.com/wp-content/uploads/2016/05/docker-net6.png)

先进入 /etc/docker/daemon.json文件里增加

{

"dns" : [

"114.114.114.114",

"8.8.8.8"

]

}

root@VM-4-6-ubuntu:/etc# docker pull ubuntu

Using default tag: latest

latest: Pulling from library/ubuntu

83ee3a23efb7: Pull complete

db98fc6f11f0: Pull complete

f611acd52c6c: Pull complete

Digest: sha256:703218c0465075f4425e58fac086e09e1de5c340b12976ab9eb8ad26615c3715

Status: Downloaded newer image for ubuntu:latest

docker.io/library/ubuntu:latest

root@VM-4-6-ubuntu:/etc# docker run -it --rm ubuntu cat etc/resolv.conf

**--rm**：容器退出时自动清理容器内部的文件系统。

**-h HOSTNAME 或者 --hostname=HOSTNAME**： 设定容器的主机名，它会被写到容器内的 /etc/hostname 和 /etc/hosts。

**--dns=IP\_ADDRESS**： 添加 DNS 服务器到容器的 /etc/resolv.conf 中，让容器用这个服务器来解析所有不在 /etc/hosts 中的主机名。

**--dns-search=DOMAIN**： 设定容器的搜索域，当设定搜索域为 .example.com 时，在搜索一个名为 host 的主机时，DNS 不仅搜索 host，还会搜索 host.example.com。

PS C:\Users\Administrator> docker login登录dockerhub

Authenticating with existing credentials...

Login Succeeded

PS C:\Users\Administrator> docker logout登出

Removing login credentials for <https://index.docker.io/v1/>

PS C:\Users\Administrator> docker search ubuntu//查找官方仓库镜像

NAME DESCRIPTION STARS OFFICIAL AUTOMATED

ubuntu Ubuntu is a Debian-based Linux operating sys… 11773 [OK]

dorowu/ubuntu-desktop-lxde-vnc Docker image to provide HTML5 VNC interface … 495 [OK]

websphere-liberty WebSphere Liberty multi-architecture images … 267 [OK]

rastasheep/ubuntu-sshd Dockerized SSH service, built on top of offi… 250 [OK]

consol/ubuntu-xfce-vnc Ubuntu container with "headless" VNC session… 231 [OK]

ubuntu-upstart Upstart is an event-based replacement for th… 110 [OK]

neurodebian NeuroDebian provides neuroscience research s… 78 [OK]

1and1internet/ubuntu-16-nginx-php-phpmyadmin-mysql-5 ubuntu-16-nginx-php-phpmyadmin-mysql-5 50 [OK]

ubuntu-debootstrap debootstrap --variant=minbase --components=m… 44 [OK]

open-liberty Open Liberty multi-architecture images based… 42 [OK]

i386/ubuntu Ubuntu is a Debian-based Linux operating sys… 24

nuagebec/ubuntu Simple always updated Ubuntu docker images w… 24 [OK]

1and1internet/ubuntu-16-apache-php-5.6 ubuntu-16-apache-php-5.6 14 [OK]

1and1internet/ubuntu-16-apache-php-7.0 ubuntu-16-apache-php-7.0 13 [OK]

1and1internet/ubuntu-16-nginx-php-phpmyadmin-mariadb-10 ubuntu-16-nginx-php-phpmyadmin-mariadb-10 11 [OK]

1and1internet/ubuntu-16-nginx-php-5.6-wordpress-4 ubuntu-16-nginx-php-5.6-wordpress-4 8 [OK]

1and1internet/ubuntu-16-nginx-php-5.6 ubuntu-16-nginx-php-5.6 8 [OK]

1and1internet/ubuntu-16-apache-php-7.1 ubuntu-16-apache-php-7.1 6 [OK]

1and1internet/ubuntu-16-nginx-php-7.0 ubuntu-16-nginx-php-7.0 4 [OK]

pivotaldata/ubuntu A quick freshening-up of the base Ubuntu doc… 4

pivotaldata/ubuntu16.04-build Ubuntu 16.04 image for GPDB compilation 2

1and1internet/ubuntu-16-php-7.1 ubuntu-16-php-7.1 1 [OK]

smartentry/ubuntu ubuntu with smartentry 1 [OK]

pivotaldata/ubuntu-gpdb-dev Ubuntu images for GPDB development 1

pivotaldata/ubuntu16.04-test Ubuntu 16.04 image for GPDB testing 0

PS C:\Users\Administrator> docker pull ubuntu

//将官方ubuntu镜像下载到本地

Using default tag: latest

latest: Pulling from library/ubuntu

Digest: sha256:703218c0465075f4425e58fac086e09e1de5c340b12976ab9eb8ad26615c3715

Status: Image is up to date for ubuntu:latest

docker.io/library/ubuntu:latest

PS C:\Users\Administrator> docker tag ubuntu:13.10 chenshuhan/ubuntu:13.10

//将自己的镜像推送到docker hub

PS C:\Users\Administrator> docker image ls

//查看镜像

REPOSITORY TAG IMAGE ID CREATED SIZE

shuhan/ubuntu ping ef2847a0db95 30 minutes ago 101MB

shuhan/ubuntu v2 a3f3f32fd395 3 hours ago 137MB

shuhan/ubuntu v3 a3f3f32fd395 3 hours ago 137MB

ubuntu latest f63181f19b2f 8 days ago 72.9MB

nginx latest f6d0b4767a6c 2 weeks ago 133MB

httpd latest 683a7aad17d3 2 weeks ago 138MB

ubuntu 15.10 9b9cb95443b5 4 years ago 137MB

training/webapp latest 6fae60ef3446 5 years ago 349MB

chenshuhan/ubuntu 13.10 7f020f7bf345 6 years ago 185MB

ubuntu 13.10 7f020f7bf345 6 years ago 185MB

shuhan@PWA6KQOQJ0WFD51:~/Dockerfile$ vim Dockerfile

//文件内容如下

FROM nginx

RUN echo '这是一个本地构建的nginx镜像' > /usr/share/nginx/html/index.html

shuhan@PWA6KQOQJ0WFD51:~/Dockerfile$ ls

Dockerfile

shuhan@PWA6KQOQJ0WFD51:~/Dockerfile$ docker build -t nginx:v3

//这里少了个 .

"docker build" requires exactly 1 argument.

See 'docker build --help'.

Usage: docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile

shuhan@PWA6KQOQJ0WFD51:~/Dockerfile$ docker build -t nginx:v3 .

//在dockerfile文件的存放目录下执行构建动作

// 最后有一个点号 它代表上下文路径

//上下文路径，是指 docker 在构建镜像，有时候想要使用到本机的文件（比如复制），docker build 命令得知这个路径后，会将路径下的所有内容打包。

[+] Building 0.4s (4/5)

[+] Building 0.6s (4/5)

[+] Building 0.7s (6/6) FINISHED

=> [internal] load build definition from Dockerfile 0.1s

=> => transferring dockerfile: 140B 0.0s

=> [internal] load .dockerignore 0.1s

=> => transferring context: 2B 0.0s

=> [internal] load metadata for docker.io/library/nginx:latest 0.0s

=> [1/2] FROM docker.io/library/nginx 0.2s

=> => resolve docker.io/library/nginx:latest 0.0s

=> [2/2] RUN echo '这是一个本地构建的nginx镜像' > /usr/share/nginx/html/index.html 0.3s

=> exporting to image 0.1s

=> => exporting layers 0.0s

=> => writing image sha256:1ce1c7efc9baa15b3560c64cd99b34045e93e37e30c942d730acf4ebe1a3c1b9 0.0s

=> => naming to docker.io/library/nginx:v3 0.0s

shuhan@PWA6KQOQJ0WFD51:~/Dockerfile$ docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

nginx v3 1ce1c7efc9ba 28 seconds ago 133MB

shuhan/ubuntu ping ef2847a0db95 58 minutes ago 101MB

shuhan/ubuntu v2 a3f3f32fd395 3 hours ago 137MB

shuhan/ubuntu v3 a3f3f32fd395 3 hours ago 137MB

ubuntu latest f63181f19b2f 8 days ago 72.9MB

nginx latest f6d0b4767a6c 2 weeks ago 133MB

httpd latest 683a7aad17d3 2 weeks ago 138MB

ubuntu 15.10 9b9cb95443b5 4 years ago 137MB

training/webapp latest 6fae60ef3446 5 years ago 349MB

chenshuhan/ubuntu 13.10 7f020f7bf345 6 years ago 185MB

ubuntu 13.10 7f020f7bf345 6 years ago 185MB

## 指令详解

具体参考：<https://www.cnblogs.com/panwenbin-logs/p/8007348.html>

**小例子**

# This my first nginx Dockerfile

# Version 1.0

# Base images 基础镜像

FROM centos

#MAINTAINER 维护者信息

MAINTAINER tianfeiyu

#ENV 设置环境变量

ENV PATH /usr/local/nginx/sbin:$PATH

#ADD 文件放在当前目录下，拷过去会自动解压

ADD nginx-1.8.0.tar.gz /usr/local/

ADD epel-release-latest-7.noarch.rpm /usr/local/

#RUN 执行以下命令

RUN rpm -ivh /usr/local/epel-release-latest-7.noarch.rpm

RUN yum install -y wget lftp gcc gcc-c++ make openssl-devel pcre-devel pcre && yum clean all

RUN useradd -s /sbin/nologin -M www

#WORKDIR 相当于cd

WORKDIR /usr/local/nginx-1.8.0

RUN ./configure --prefix=/usr/local/nginx --user=www --group=www --with-http\_ssl\_module --with-pcre && make && make install

RUN echo "daemon off;" >> /etc/nginx.conf

#EXPOSE 映射端口

EXPOSE 80

#CMD 运行以下命令

CMD ["nginx"]

