Docker ppt学习

# Docker学习笔记：Docker 基础用法和命令帮助

参考：<http://www.docker.org.cn/dockerppt/106.html>

**一、Docker的基础用法**

Docker镜像首页，包括官方镜像和其它公开镜像

因为国情的原因，国内下载 Docker HUB 官方的相关镜像比较慢，可以使用 [docker.cn](http://opskumu.github.io/docker.cn) 镜像，镜像保持和官方一致，关键是速度块，推荐使用。

### 3.1 Search images

$ sudo docker search ubuntu

### 3.2 Pull images

$ sudo docker pull ubuntu # 获取 ubuntu 官方镜像

$ sudo docker images # 查看当前镜像列表

### 3.3 Running an interactive shell

$ sudo docker run -i -t ubuntu:14.04 /bin/bash

* docker run - 运行一个容器
* -t - 分配一个（伪）tty (link is external)
* -i - 交互模式 (so we can interact with it)
* ubuntu:14.04 - 使用 ubuntu 基础镜像 14.04
* **/bin/bash - 运行命令 bash shell**

注: ubuntu 会有多个版本，通过指定 tag 来启动特定的版本 [image]:[tag]

$ sudo docker ps # 查看当前运行的容器,

ps -a #列出当前系统所有的容器

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

6c9129e9df10 ubuntu:14.04 /bin/bash 6 minutes ago Up 6 minutes cranky\_babbage

### 3.4 相关快捷键

* 退出：Ctrl-Dorexit
* detach：Ctrl-P + Ctrl-Q
* attach:docker attach CONTAINER-ID

## ****二、Docker 命令帮助****

### 4.1 docker help

#### docker command

$ sudo docker # docker 命令帮助

Commands:

attach Attach to a running container # 当前 shell 下 attach 连接指定运行镜像

build Build an image from a Dockerfile # 通过 Dockerfile 定制镜像

commit Create a new image from a container's changes # 提交当前容器为新的镜像

cp Copy files/folders from the containers filesystem to the host path

# 从容器中拷贝指定文件或者目录到宿主机中

create Create a new container # 创建一个新的容器，同 run，但不启动容器

diff Inspect changes on a container's filesystem # 查看 docker 容器变化

[root@master testtmp]# docker ps -l

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

1f4d00854721 centos:latest "/bin/bash" About an hour ago Exited (127) About an hour ago jovial\_meitner

[root@master testtmp]# docker diff 1f4d0085

C /root

A /root/.bash\_history

[root@master testtmp]#

events Get real time events from the server # 从 docker 服务获取容器实时事件

exec Run a command in an existing container # 在已存在的容器上运行命令

export Stream the contents of a container as a tar archive

# 导出容器的内容流作为一个 tar 归档文件[对应 import ]

history Show the history of an image # 展示一个镜像形成历史

root@master testtmp]# docker history centos

IMAGE CREATED CREATED BY SIZE COMMENT

97cad5e16cb6 2 weeks ago /bin/sh -c #(nop) CMD ["/bin/bash"] 0 B

05fe84bf6d3f 2 weeks ago /bin/sh -c #(nop) LABEL name=CentOS Base Ima 0 B

af0819ed1fac 2 weeks ago /bin/sh -c #(nop) ADD file:54df3580ac9fb66389 196.5 MB

3690474eb5b4 11 weeks ago /bin/sh -c #(nop) MAINTAINER https://github.

images List images # 列出系统当前镜像

import Create a new filesystem image from the contents of a tarball

# 从tar包中的内容创建一个新的文件系统映像[对应 export]

info Display system-wide information # 显示系统相关信息

[root@master testtmp]# docker info

Containers: 6

Images: 6

Storage Driver: devicemapper

Pool Name: docker-8:2-413022-pool

Pool Blocksize: 65.54 kB

Backing Filesystem: extfs

Data file: /dev/loop0

Metadata file: /dev/loop1

. . .

inspect Return low-level information on a container # 查看容器详细信息

[root@master testtmp]# docker inspect 1f

Error: No such image or container: 1f

[]

[root@master testtmp]# docker inspect 1f4d

[

{

"Id": "1f4d00854721b28b87698f85e1201492edae6a93665e6ed3592e0f1f1c320201",

"Created": "2016-11-19T16:55:54.450886779Z",

"Path": "/bin/bash",

"Args": [],

"State": {

. . .

kill Kill a running container # kill 指定 docker 容器

load Load an image from a tar archive # 从一个 tar 包中加载一个镜像[对应 save]

login Register or Login to the docker registry server

# 注册或者登陆一个 docker 源服务器

logout Log out from a Docker registry server # 从当前 Docker registry 退出

logs Fetch the logs of a container # 输出当前容器日志信息

port Lookup the public-facing port which is NAT-ed to PRIVATE\_PORT

# 查看映射端口对应的容器内部源端口

pause Pause all processes within a container # 暂停容器

ps List containers # 列出容器列表

[root@master Desktop]# docker attach 1f4

You cannot attach to a stopped container, start it first

[root@master Desktop]# docker start 1f4

1f4

[root@master Desktop]# docker attach 1f4

[root@1f4d00854721 /]# ls

pull Pull an image or a repository from the docker registry server

# 从docker镜像源服务器拉取指定镜像或者库镜像

push Push an image or a repository to the docker registry server

# 推送指定镜像或者库镜像至docker源服务器

restart Restart a running container # 重启运行的容器

rm Remove one or more containers # 移除一个或者多个容器

rmi Remove one or more images

# 移除一个或多个镜像[无容器使用该镜像才可删除，否则需删除相关容器才可继续或 -f 强制删除]

run Run a command in a new container

# 创建一个新的容器并运行一个命令

save Save an image to a tar archive # 保存一个镜像为一个 tar 包[对应 load]

search Search for an image on the Docker Hub # 在 docker hub 中搜索镜像

start Start a stopped containers # 启动容器

stop Stop a running containers # 停止容器

tag Tag an image into a repository # 给源中镜像打标签

top Lookup the running processes of a container # 查看容器中运行的进程信息

[root@master Desktop]# docker top 1f4

Error response from daemon: Container 1f4 is not running

[root@master Desktop]# docker start 1f4

1f4

[root@master Desktop]# docker top 1f4

UID PID PPID C STIME TTY TIME CMD

root 3246 2222 0 12:04 pts/2 00:00:00 /bin/bash

unpause Unpause a paused container # 取消暂停容器

version Show the docker version information # 查看 docker 版本号

wait Block until a container stops, then print its exit code

# 截取容器停止时的退出状态值

Run 'docker COMMAND --help' for more information on a command.

#### docker option

Usage of docker:

--api-enable-cors=false Enable CORS headers in the remote API # 远程 API 中开启 CORS 头

-b, --bridge="" Attach containers to a pre-existing network bridge # 桥接网络

use 'none' to disable container networking

--bip="" Use this CIDR notation address for the network bridge's IP, not compatible with -b

# 和 -b 选项不兼容，具体没有测试过

-d, --daemon=false Enable daemon mode # daemon 模式

-D, --debug=false Enable debug mode # debug 模式

--dns=[] Force docker to use specific DNS servers # 强制 docker 使用指定 dns 服务器

--dns-search=[] Force Docker to use specific DNS search domains # 强制 docker 使用指定 dns 搜索域

-e, --exec-driver="native" Force the docker runtime to use a specific exec driver # 强制 docker 运行时使用指定执行驱动器

--fixed-cidr="" IPv4 subnet for fixed IPs (ex: 10.20.0.0/16)

this subnet must be nested in the bridge subnet (which is defined by -b or --bip)

-G, --group="docker" Group to assign the unix socket specified by -H when running in daemon mode

use '' (the empty string) to disable setting of a group

-g, --graph="/var/lib/docker" Path to use as the root of the docker runtime # 容器运行的根目录路径

-H, --host=[] The socket(s) to bind to in daemon mode # daemon 模式下 docker 指定绑定方式[tcp or 本地 socket]

specified using one or more tcp://host:port, unix:///path/to/socket, fd://\* or fd://socketfd.

--icc=true Enable inter-container communication # 跨容器通信

--insecure-registry=[] Enable insecure communication with specified registries (no certificate verification for HTTPS and enable HTTP fallback) (e.g., localhost:5000 or 10.20.0.0/16)

--ip="0.0.0.0" Default IP address to use when binding container ports # 指定监听地址，默认所有 ip

--ip-forward=true Enable net.ipv4.ip\_forward # 开启转发

--ip-masq=true Enable IP masquerading for bridge's IP range

--iptables=true Enable Docker's addition of iptables rules # 添加对应 iptables 规则

--mtu=0 Set the containers network MTU # 设置网络 mtu

if no value is provided: default to the default route MTU or 1500 if no default route is available

-p, --pidfile="/var/run/docker.pid" Path to use for daemon PID file # 指定 pid 文件位置

--registry-mirror=[] Specify a preferred Docker registry mirror

-s, --storage-driver="" Force the docker runtime to use a specific storage driver # 强制 docker 运行时使用指定存储驱动

--selinux-enabled=false Enable selinux support # 开启 selinux 支持

--storage-opt=[] Set storage driver options # 设置存储驱动选项

--tls=false Use TLS; implied by tls-verify flags # 开启 tls

--tlscacert="/root/.docker/ca.pem" Trust only remotes providing a certificate signed by the CA given here

--tlscert="/root/.docker/cert.pem" Path to TLS certificate file # tls 证书文件位置

--tlskey="/root/.docker/key.pem" Path to TLS key file # tls key 文件位置

--tlsverify=false Use TLS and verify the remote (daemon: verify client, client: verify daemon) # 使用 tls 并确认远程控制主机

-v, --version=false Print version information and quit # 输出 docker 版本信息

### 4.2 docker search

$ sudo docker search --help

[root@master Desktop]# docker search --help

Usage: docker search [OPTIONS] TERM

Search the Docker Hub for images

--automated=false Only show automated builds

--help=false Print usage

--no-trunc=false Don't truncate output

-s, --stars=0 Only displays with at least x stars

[root@master Desktop]#

Usage: docker search TERM

Search the Docker Hub for images # 从 Docker Hub 搜索镜像

--automated=false Only show automated builds

--no-trunc=false Don't truncate output

-s, --stars=0 Only displays with at least xxx stars

示例：

$ sudo docker search -s 100 ubuntu # 查找 star 数至少为 100 的镜像，找出只有官方镜像 start 数超过 100，默认不加 s 选项找出所有相关 ubuntu 镜像 NAME DESCRIPTION STARS OFFICIAL AUTOMATED

ubuntu Official Ubuntu base image 425 [OK]

### 4.3 docker info

$ sudo docker info

Containers: 1 # 容器个数 Images: 22 # 镜像个数 Storage Driver: devicemapper # 存储驱动 Pool Name: docker-8:17-3221225728-pool

Pool Blocksize: 65.54 kB

Data file: /data/docker/devicemapper/devicemapper/data

Metadata file: /data/docker/devicemapper/devicemapper/metadata

Data Space Used: 1.83 GB

Data Space Total: 107.4 GB

Metadata Space Used: 2.191 MB

Metadata Space Total: 2.147 GB

Library Version: 1.02.84-RHEL7 (2014-03-26) Execution Driver: native-0.2 # 存储驱动 Kernel Version: 3.10.0-123.el7.x86\_64

Operating System: CentOS Linux 7 (Core)

### 4.4 docker pull && docker push

$ sudo docker pull --help # pull 拉取镜像 Usage: docker pull [OPTIONS] NAME[:TAG] Pull an image or a repository from the registry

-a, --all-tags=false Download all tagged images in the repository $ sudo docker push # push 推送指定镜像 Usage: docker push NAME[:TAG] Push an image or a repository to the registry

示例：

$ sudo docker pull ubuntu # 下载官方 ubuntu docker 镜像，默认下载所有 ubuntu 官方库镜像 $ sudo docker pull ubuntu:14.04 # 下载指定版本 ubuntu 官方镜像

$ sudo docker push 192.168.0.100:5000/ubuntu # 推送镜像库到私有源[可注册 docker 官方账户，推送到官方自有账户] $ sudo docker push 192.168.0.100:5000/ubuntu:14.04 # 推送指定镜像到私有源

### 4.5 docker images

列出当前系统镜像

$ sudo docker images --help

Usage: docker images [OPTIONS] [NAME] List images

-a, --all=false Show all images (by default filter out the intermediate image layers) # -a 显示当前系统的所有镜像，包括过渡层镜像，默认 docker images 显示最终镜像，不包括过渡层镜像

-f, --filter=[] Provide filter values (i.e. 'dangling=true')

--no-trunc=false Don't truncate output

-q, --quiet=false Only show numeric IDs

--digests=false Show digests

示例：

$ sudo docker images # 显示当前系统镜像，不包括过渡层镜像

$ sudo docker images -a # 显示当前系统所有镜像，包括过渡层镜像

$ sudo docker images ubuntu # 显示当前系统 docker ubuntu 库中的所有镜像 REPOSITORY TAG IMAGE ID CREATED VIRTUAL SIZE

ubuntu 12.04 ebe4be4dd427 4 weeks ago 210.6 MB

ubuntu 14.04 e54ca5efa2e9 4 weeks ago 276.5 MB

ubuntu 14.04-ssh 6334d3ac099a 7 weeks ago 383.2 MB

### 4.6 docker rmi

删除一个或者多个镜像

$ sudo docker rmi --help

Usage: docker rmi IMAGE [IMAGE...] Remove one or more images

-f, --force=false Force removal of the image # 强制移除镜像不管是否有容器使用该镜像

--no-prune=false Do not delete untagged parents # 不要删除未标记的父镜像

### 4.7 docker run

$ sudo docker run --help

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...] Run a command in a new container

-a, --attach=[] Attach to stdin, stdout or stderr.

-c, --cpu-shares=0 CPU shares (relative weight) # 设置 cpu 使用权重

--cap-add=[] Add Linux capabilities

--cap-drop=[] Drop Linux capabilities

--cidfile="" Write the container ID to the file # 把容器 id 写入到指定文件

--cpuset="" CPUs in which to allow execution (0-3, 0,1) # cpu 绑定 -d,

--detach=false Detached mode: Run container in the background, print new container id # 后台运行容器

--device=[] Add a host device to the container (e.g. --device=/dev/sdc:/dev/xvdc)

--dns=[] Set custom dns servers # 设置

dns --dns-search=[] Set custom dns search domains # 设置 dns 域搜索

-e, --env=[] Set environment variables # 定义环境变量

--entrypoint="" Overwrite the default entrypoint of the image # ？ -

-env-file=[] Read in a line delimited file of ENV variables # 从指定文

件读取变量值

--expose=[] Expose a port from the container without publishing it to your host # 指定对外提供服务端口

-h, --hostname="" Container host name # 设置容器主机名

-i, --interactive=false Keep stdin open even if not attached # 保持标准输出开启即使没有 attached

--link=[] Add link to another container (name:alias) # 添加链接到另外一个容器

--lxc-conf=[] (lxc exec-driver only) Add custom lxc options --lxc-conf="lxc.cgroup.cpuset.cpus = 0,1"

-m, --memory="" Memory limit (format: <number><optional unit>, where unit = b, k, m or g) # 内存限制

--name="" Assign a name to the container # 设置容器名

--net="bridge" Set the Network mode for the container # 设置容器网络模式 'bridge': creates a new network stack for the container on the docker bridge 'none': no networking for this container 'container:<name|id>': reuses another container network stack 'host': use the host network stack inside the container. Note: the host mode gives the container full access to local system services such as D-bus and is therefore considered insecure.

-P, --publish-all=false Publish all exposed ports to the host interfaces # 自动映射容器对外提供服务的端口 -p, --publish=[] Publish a container's port to the host # 指定端口映射 format: ip:hostPort:containerPort | ip::containerPort | hostPort:containerPort (use 'docker port' to see the actual mapping) --privileged=false Give extended privileges to this container # 提供更多的权限给容器 --restart="" Restart policy to apply when a container exits (no, on-failure[:max-retry], always) --rm=false Automatically remove the container when it exits (incompatible with -d) # 如果容器退出自动移除和 -d 选项冲突 --security-opt=[] Security Options

--sig-proxy=true Proxify received signals to the process (even in non-tty mode). SIGCHLD is not proxied.

-t, --tty=false Allocate a pseudo-tty # 分配伪终端 -u, --user="" Username or UID # 指定运行容器的用户 uid 或者用户名 -v, --volume=[] Bind mount a volume (e.g., from the host: -v /host:/container, from docker: -v /container) # 挂载卷 --volumes-from=[] Mount volumes from the specified container(s) # 从指定容器挂载卷 -w, --workdir="" Working directory inside the container # 指定容器工作目录

示例：

$ sudo docker images ubuntu

REPOSITORY TAG IMAGE ID CREATED VIRTUAL SIZE

ubuntu 14.04 e54ca5efa2e9 4 weeks ago 276.5 MB

... ... $ sudo docker run -t -i -c 100 -m 512MB -h test1 -d --name="docker\_test1" ubuntu /bin/bash # 创建一个 cpu 优先级为 100，内存限制 512MB，主机名为 test1，名为 docker\_test1 后台运行 bash 的容器 a424ca613c9f2247cd3ede95adfbaf8d28400cbcb1d5f9b69a7b56f97b2b52e5 $ sudo docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

a424ca613c9f ubuntu:14.04 /bin/bash 6 seconds ago Up 5 seconds docker\_test1 $ sudo docker attach docker\_test1

root@test1:/# pwd /

root@test1:/# exit exit

关于cpu优先级:

By default all groups have 1024 shares. A group with 100 shares will get a ~10% portion of the CPU time -[archlinux cgroups](https://wiki.archlinux.org/index.php/cgroups)

### 4.8 docker start|stop|kill... ...

Dockerstart|stop|kill|restart|pause|unpause|rm|commit|inspect|logs

docker start CONTAINER [CONTAINER...]# 运行一个或多个停止的容器

docker stop CONTAINER [CONTAINER...]# 停掉一个或多个运行的容器-t选项可指定超时时间

docker kill [OPTIONS] CONTAINER [CONTAINER...]# 默认 kill 发送 SIGKILL 信号-s可以指定发送 kill 信号类型

docker restart [OPTIONS] CONTAINER [CONTAINER...]# 重启一个或多个运行的容器-t选项可指定超时时间

docker pause CONTAINER# 暂停一个容器，方便 commit

docker unpause CONTAINER# 继续暂停的容器

docker rm [OPTIONS] CONTAINER [CONTAINER...]# 移除一个或多个容器

-f, --force=false Force removal of running container #强制删除一个正在运行中的容器

-l, --link=false Remove the specified link and not the underlying container

-v, --volumes=false Remove the volumes associated with the container

docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]# 提交指定容器为镜像

-a, --author="" Author (e.g., "John Hannibal Smith [hannibal@a-team.com](mailto:hannibal@a-team.com)")

-m, --message="" Commit message

-p, --pause=true Pause container during commit

# 默认 commit 是暂停状态

docker inspect CONTAINER|IMAGE [CONTAINER|IMAGE...]# 查看容器或者镜像的详细信息

docker logs CONTAINER# 输出指定容器日志信息

-f, --follow=false Follow log output

# 类似 tail -f

-t, --timestamps=false Show timestamps

--tail="all" Output the specified number of lines at the end of logs (defaults to all logs)

参考文档：[Docker Run Reference](https://docs.docker.com/reference/run/)