Docker的基本使用

# 1.docker的安装

# 2.docker的基本命令

Usage: docker [OPTIONS] COMMAND [arg...]

A self-sufficient runtime for linux containers.

Options:

--api-cors-header= Set CORS headers in the remote API

-b, --bridge= Attach containers to a network bridge

--bip= Specify network bridge IP

-D, --debug=false Enable debug mode

-d, --daemon=false Enable daemon mode

--default-gateway= Container default gateway IPv4 address

--default-gateway-v6= Container default gateway IPv6 address

--default-ulimit=[] Set default ulimits for containers

--dns=[] DNS server to use

--dns-search=[] DNS search domains to use

-e, --exec-driver=native Exec driver to use

--exec-opt=[] Set exec driver options

--exec-root=/var/run/docker Root of the Docker execdriver

--fixed-cidr= IPv4 subnet for fixed IPs

--fixed-cidr-v6= IPv6 subnet for fixed IPs

-G, --group=docker Group for the unix socket

-g, --graph=/var/lib/docker Root of the Docker runtime

-H, --host=[] Daemon socket(s) to connect to

-h, --help=false Print usage

--icc=true Enable inter-container communication

--insecure-registry=[] Enable insecure registry communication

--ip=0.0.0.0 Default IP when binding container ports

--ip-forward=true Enable net.ipv4.ip\_forward

--ip-masq=true Enable IP masquerading

--iptables=true Enable addition of iptables rules

--ipv6=false Enable IPv6 networking

-l, --log-level=info Set the logging level

--label=[] Set key=value labels to the daemon

--log-driver=json-file Default driver for container logs

--log-opt=map[] Set log driver options

--mtu=0 Set the containers network MTU

-p, --pidfile=/var/run/docker.pid Path to use for daemon PID file

--registry-mirror=[] Preferred Docker registry mirror

-s, --storage-driver= Storage driver to use

--selinux-enabled=false Enable selinux support

--storage-opt=[] Set storage driver options

--tls=false Use TLS; implied by --tlsverify

--tlscacert=~/.docker/ca.pem Trust certs signed only by this CA

--tlscert=~/.docker/cert.pem Path to TLS certificate file

--tlskey=~/.docker/key.pem Path to TLS key file

--tlsverify=false Use TLS and verify the remote

--userland-proxy=true Use userland proxy for loopback traffic

-v, --version=false Print version information and quit

Commands:

attach Attach to a running container

build Build an image from a Dockerfile

commit Create a new image from a container's changes

cp Copy files/folders from a container's filesystem to the host path

create Create a new container

diff Inspect changes on a container's filesystem

events Get real time events from the server

exec Run a command in a running container

export Stream the contents of a container as a tar archive

history Show the history of an image

images List images

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

info Display system-wide information

inspect Return low-level information on a container or image

kill Kill a running container

load Load an image from a tar archive

login Register or log in to a Docker registry server

logout Log out from a Docker registry server

logs Fetch the logs of a container

pause Pause all processes within a container

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

ps List containers

pull Pull an image or a repository from a Docker registry server

push Push an image or a repository to a Docker registry server

rename Rename an existing container

restart Restart a running container

rm Remove one or more containers

rmi Remove one or more images

run Run a command in a new container

save Save an image to a tar archive

search Search for an image on the Docker Hub

start Start a stopped container

stats Display a stream of a containers' resource usage statistics

stop Stop a running container

tag Tag an image into a repository

top Lookup the running processes of a container

unpause Unpause a paused container

version Show the Docker version information

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

# 3.docker 网络

# 4.docker mount