#stop and Remove ALL containers

docker stop $(docker ps -aq); docker rm $(docker ps -aq)

#stop ALL containers

docker stop $(docker ps -a -q)

do

# remove ALL containers

docker rm -f $(docker ps -a -q)

Do

For Containers :

docker stop $(docker ps -qa)

docker rm $(docker ps -qa)

For Images :

dockemi $(docker images -qa)

docker images

docker rmi -f b00ea124ed62 529165268aa2 0c45f7936948

docker images

Volume Example usage:

docker run -v c:\ContainerData:c:\data:RO for read-only access

docker run -v c:\ContainerData:c:\data:RW for read-write access

docker run -v c:\ContainerData:c:\data for read-write access (default)

docker run -itd -p 8030:80 -m 1GB --name nginx1 -v c:/html:/usr/share/nginx/html nginx

docker run -itd -p 8040:80 -m 1GB --name nginx2 -v c:/html:/usr/share/nginx/html:ro nginx:v2

Docker run :

--privileged

$ docker run -t -i --rm ubuntu bash

root@bc338942ef20:/# mount -t tmpfs none /mnt

mount: permission denied

$ docker run -t -i --privileged ubuntu bash

root@50e3f57e16e6:/# mount -t tmpfs none /mnt

root@50e3f57e16e6:/# df -h

Filesystem Size Used Avail Use% Mounted on

none 1.9G 0 1.9G 0% /mnt

-w

$ docker run -w /path/to/dir/ -i -t ubuntu pwd

The -w lets the command being executed inside directory given, here /path/to/dir/.

Note : If the path does not exist it is created inside the container.

docker run -itd -p 8050:80 -m 1GB --name nginx3 -w //usr//share//nginx//html -v c:/html:/usr/share/nginx/html nginx

-e, --env, --env-file

$ docker run -e MYVAR1 --env MYVAR2=foo --env-file ./env.list ubuntu bash

$ docker run --env VAR1=value1 --env VAR2=value2 ubuntu env | grep VAR

VAR1=value1

VAR2=value2

Limiting Memory

$ docker run -d -p 8081:80 --memory=20m --memory-swap=20m nginx

$ docker container run -d --memory-reservation=250m --name mymem1 alpine:3.8 sleep 3600

Limiting CPU

--cpus

Docker 1.13 and higher:

$ docker run -it --cpus=".5" ubuntu /bin/bash

Docker 1.12 and lower:

$ docker run -it --cpu-period=100000 --cpu-quota=50000 ubuntu /bin/bash

$ docker run -it --cpus-shares="512" ubuntu /bin/bash

docker stats :

$ docker stop $(docker ps -aq); docker rm $(docker ps -aq)

$ docker run -itd -p 8030:80 --name nginx7 -v c:/html:/usr/share/nginx/html:ro nginx:v2

$ docker stats

CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS

779eb8148aa7 nginx7 0.00% 1.914MiB / 8.75GiB 0.02% 906B / 0B 0B / 4.1kB 2

Create and start a container

$ docker create -t -i fedora bash

6d8af538ec541dd581ebc2a24153a28329acb5268abe5ef868c1f1a261221752

$ docker start -a -i 6d8af538ec5

bash-4.2#

Copy :

Copy a file from host to container:

docker cp Dockerfile 779eb8148aa7:/tmp/Dockerfile

docker exec -it 779eb8148aa7 //bin/bash

docker cp Dockerfile 779eb8148aa7:/tmp/Dockerfile123

docker exec -it 779eb8148aa7 //bin/bash

Copy a file from Docker container to host:

docker cp 779eb8148aa7:/tmp/Dockerfile123 Dockerfile\_Delete

Copy a Folder from host to container:

docker cp /home/captain/my\_dir ubu\_container:/home

docker cp ubu\_container:/home/my\_dir /home/captain

Logs :

$ docker logs 779eb8148aa7 --follow