

Docker Useful Commands – Part I

Pull image to local repo and Update latest

Two purpose

- ❑ If image is not in local repository, download image to local repository
- ❑ If found in local repository, update to latest version.

`docker pull <image_name>`

Run Container

`docker run -d --name <container> <image>`

-d = run container in detached mode
 --name = preferred container name. If not specified, docker automatically assigns name.

Build image using Dockerfile

- ❑ Your docker file must be named as "Dockerfile"

`docker build -t <custom_image_name>:<tag> .`

-t = image name with optional tag
 dot represents docker file in current path.

List Images and Container

//show images
`docker images`

//show running container
`docker ps`

//show running and exited container
`docker ps -a`

Start/Stop Existing Container

`docker stop <container_id_or_name>`
`docker start <container_id_or_name>`
`docker restart <container_id_or_name>`
`docker pause <container_id_or_name>`
`docker unpause <container_id_or_name>`

IP address of Container

`docker inspect <container_name>`

In the output, look for IP address.

★ Docker uses default 172.17.0.0/16 subnet for container networking.

Container Network

- ❑ Three types of container network are possible:

1. Bridge = default network.
2. None = no IP address. Isolated container

`docker run -d --net=none <image_name>`

3. Host = Host IP address is shared.

`docker run -d --net=host <image_name>`

★ Docker container runs as long as it takes for the command issued in container to complete.

★ By default, any data created inside container is only available from within the container and while the container is running.

Mount External Volumes

`docker run -d -name <container> -v <host_vol>:<container_vol> image`

-v = sets up bindmount volume that links a directory from inside container to directory on host machine

E.g.
 -v ~/nginxlogs:/var/log/nginx mounts host directory ~/nginxlogs to container directory /var/log/nginx

-v nginxlogs:/var/log/nginx is not mounting, but creating volume name.

Map Port to Container

`docker run -d -name <container> -p <host_port>:<container_port> image`

-p = map host port to container port

- ❑ If host port is not specified then a random port is chosen from host

`docker run -d -name <container> -p container_port> image`

Attach to running container (Terminal access)

Option 1:

`docker exec -it <container_name> <cmd>`

-i = interactive, -t is terminal
 <cmd> can be any acceptable command on the container.
 E.g.
 'bash' to access container terminal or
 'cat /etc./hosts' to access content of specific file.

Option 2: as if viewing container screen
`docker attach <container_name>`

Type Ctrl+p then Ctrl+q to detach from container. Do not type Ctrl+c as this key sequence sends SIGKILL to container.

★ Docker installation creates file in /etc/apt/source.list.d/ named docker.list that pulls latest Docker version when you try apt-get update.

Output of Container

`docker logs -f <container_name>`

-f = follow upcoming log messages.
 When you are done, hit Ctrl+C

Build image from running container

`docker commit <container_name> <new_image>`

-f = follow upcoming log messages.
 When you are done, hit Ctrl+C

Delete Container and Image

//remove container
`docker rm <container_id_or_name>`

- ❑ If container is running, above command doesn't work so use -f to remove forcefully.
`docker rm -f <container_id_or_name>`

//remove image
`docker rmi <image_name>`

Resource Usage

//running container's resource usage statistics
`docker stats`

//list all containers even if not running
`docker status --all`

Running Processes

//shows running processes in a container

`docker top <container>`

★ Users who need to run docker without root access. Add such user to Docker group

//docker group exists, so there is no need to add but kept for reference.
`$ sudo groupadd docker`

//specify your username
`$ sudo gpasswd -a <username> docker`

`$sudo service docker restart`