Terminologies:

Term	Description	Similar Java term (just for relating/understanding)
Dockerfile (no extension)	Infrastructure as code	.java
Build	Creating an image snapshot from the Dockerfile	compile / package
Image	VM Snapshot	.class / .jar
Tag	Version of image / release	jar version
Container	Light weight VM Created from a specific image version We can create multiple container from same Image	instances / Objects
DockerHub	Image Repository	Maven Repository

Docker Commands:

command	Description
docker pull *image*	pulls an image from DockerHub
docker images	shows all the images in our machine
docker ps	shows all the running containers
docker ps -a	shows all the containers including stopped containers
docker run *image*	creates a container from an image
docker stop *container id/container name*	stops a running container
docker system prune -f	To remove all the stopped containers volumes etc (docker do not touch the running containers) (-f forcefully remove)
docker system prune -a	same as above + unused images

Image Name Format:

*docker pull **[repository-host:port] / [owner] / imageName [:tag]***

- Optional
  - repository-host:port
  - owner
  - tag
- Mandatory
  - imageName

Docker Run Options:

Option	**Description**
-d	To run the container in the backgound
-i	To attach the standard input to the container
-t	To attach the standard output to the container
--name *somename*	To assign given name to our container
--entrypoint*=/bin/bash*	Entrypoint is the command to be executed once a container is created. An image might have an entrypoint already. This option overrides the container's entrypoint
-p 8080:8080	To map a host port with a container port -p **host-port:container-port**
-v /a/b/c:/user/share	To map a host directory with a container directory -p host-directory:container-directory

Dockerfile Commands:

command	Description
FROM *image*	The image which needs to be used as base image. Very first command in the Dockerfile
ADD *host-files* *container-files*	Adds files from your host into your image
RUN *command*	Runs any given command - useful to install any software, create directories etc
ENV *envVar* *envValue*	Sets an environment variable in the image
WORKDIR *dir*	Sets a default working directory. [if we ignore / is the working directory]
EXPOSE *port-#*	Exposes a port for your application
ENTRYPOINT *command-to-be-executed*	Command to be executed once a container is created from this image
#	Comments the line

Docker Network Usage:

```
docker network create name
docker run --network=name nginx
docker run --network=name ubuntu
```