Runnnig and Managing Docker Containers

Concepts

- To start a container and keep it running in the background you need to explicitly tell Docker to do so
- To keep it running you have to 'detach' it
- using -d to daemonize
- `docker run -dit debian`
- t & i works together for interact with the container os
    - i for interactivity
    - t for TTY

Naming

Similar to IP addresses and DNS, you acn give a conatiner name and refer to it this way, instead of its container ID.

- `docker run -dit --name=web debian`
- `docker ps`

Removing

1. STOP: `docker stop web`
    - yoy can also start a stopped container using `docker start`
2. REMOVE: `docker rm web`

Always Restart Policy

- If a container crashes for some reason, you might want it to restart on its own instead of waiting for a human to manually restart it
- `docker run -dit --restart=always name=container_name debian`
-
- OUTPUT: Name: always, MaximunRetryCount: 0;

if a container stops for any reason and no restart policy was specified, the conatiner will remain stopped since the default restart policy is 'no'.

Without supply a restart policy, if the container stops due to an error, docker engine restarts, or host machine is rebooted, the container will remains stopped.

Policies

- default
- on-failure -> restart due to an error

- unless-stopped: it will remain stopped if before a system reboot or engine restart the user manually stopped
- always

Stop

- docker stop container_name
- docker kill container_name

Kill it skips graceful shutdown. Kill might put the container in such a state that it will not be able to restart.

Sopped Containers

- DELETE STOPPED CONTAINERS -> `docker system prune`
- this will remove all stopped containers

Auto Deting Stopped Container

- `docker run --rm hello-world`