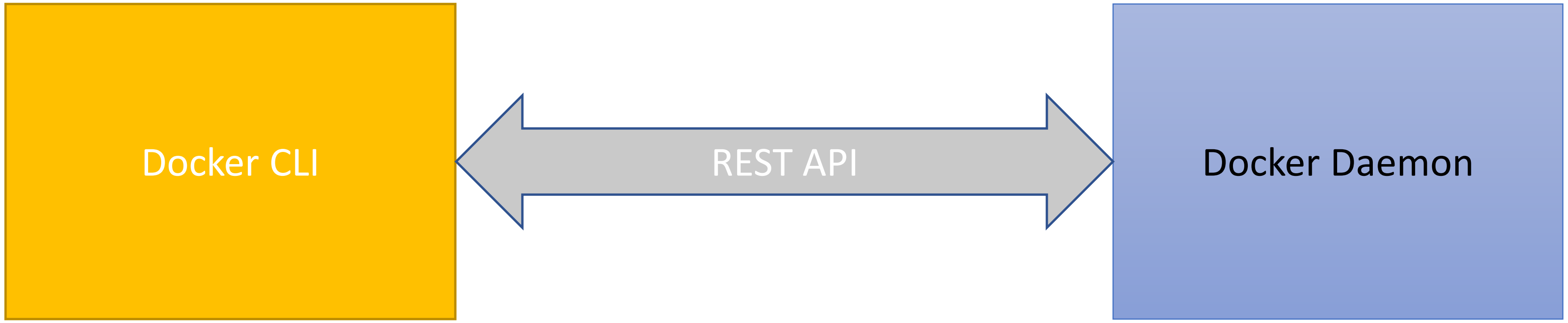
Docker Architecture

Docker Architecture



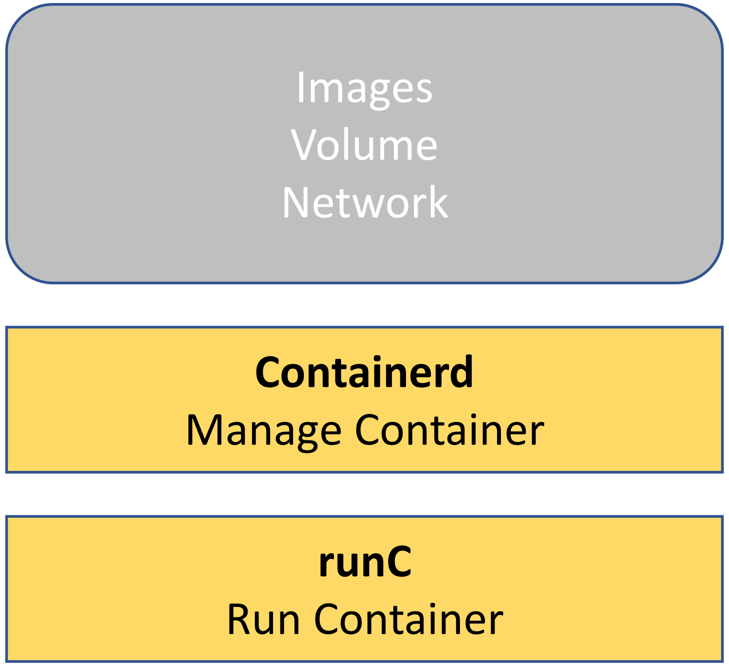
Docker CLI

REST API

Docker Daemon

Docker Daemon

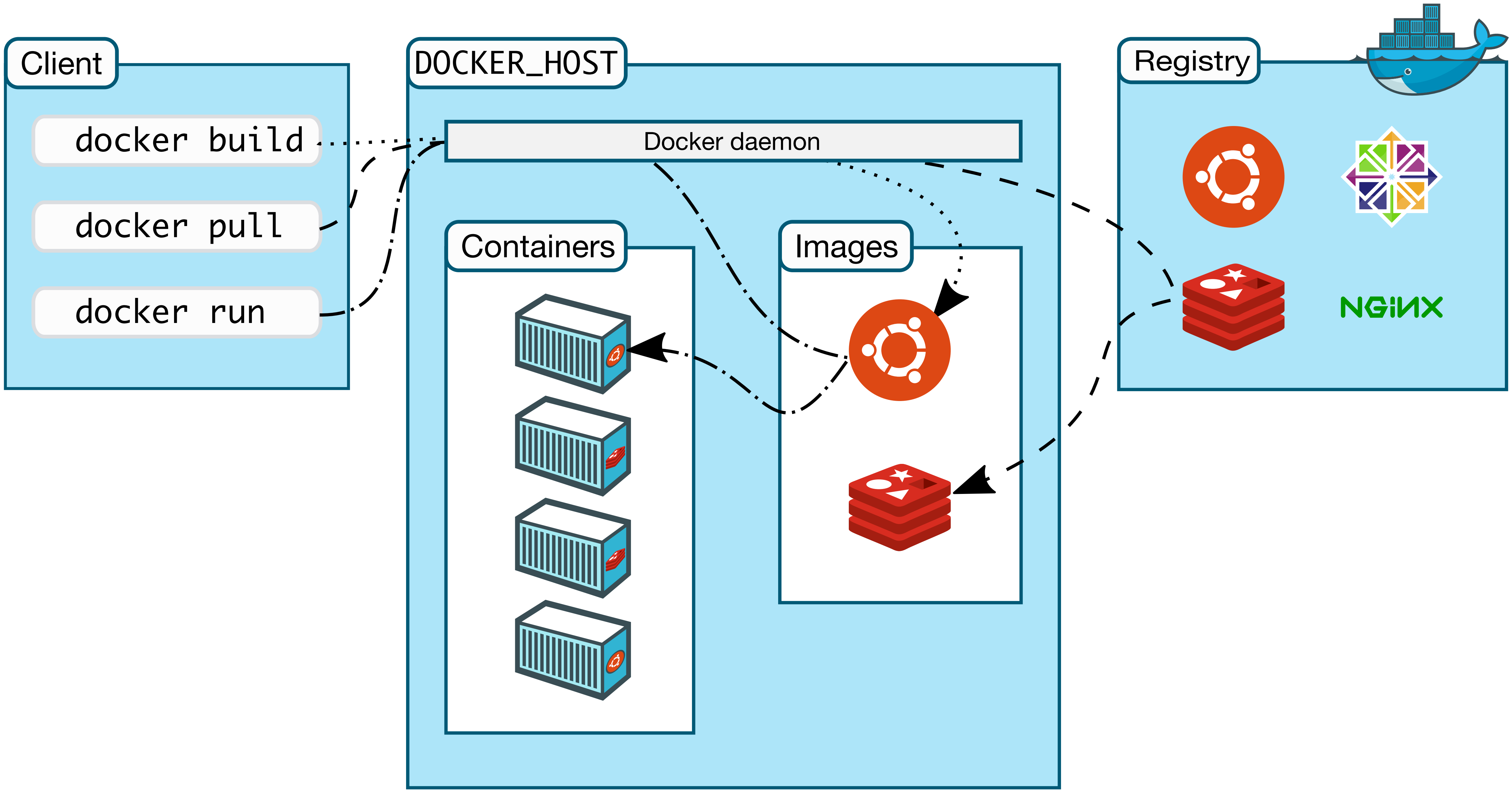
- Docker Object
  - Images
  - Volumes
  - Networks
  - Containers
    - Manage
    - Run



Images
Volume
Network

Containerd
Manage Container

runC
Run Container


Client
docker build
docker pull
docker run
DOCKER_HOST
Docker daemon
Containers
Images
Registry
NGINX

Docker Installation

- https://docs.docker.com/engine/install/
- Docker version
- Docker –v
- Docker system info
- Docker run –p 80:80 nginx

Docker Command

- docker <docker-object> <sub-command> [options] <Arguments/Commands>

- docker image ls
- docker volume ls
- docker network ls
- docker container ls

Create a New Container

- Create a new container

  docker container create nginx

- Check the container:
  - cd /var/lib/docker/containers
  - ls –l
- List the details for container
  - **docker** container ls
  - **docker** container ls -aq

Start A Container

- Start a container which is created earlier
  - **docker** container start 104be62d7603
  - **docker** container ls

Create and Start a container (run)

- Run a container
  - **docker** container run ubuntu
  - **docker** container ls -a
  - **docker** container run -it ubuntu (interactive)
  - **docker** container run -it --name=webapp ubuntu
- Rename
    - **docker** container rename webapp mywebapp
- Run in a **detached** mode
  - **docker** container run –d --name=webapp ubuntu
  - **docker** container **attach** 19cb

Executing Commands

- **docker** container exec -it 56rt /bin/bash
- Attach the container
    - **docker** container attach 56rt

- Demo:
    - run container without attach
    - With attach using **–d** option
    - Attach existing running container using **attach** option

Container Inspect

- **docker** container inspect webapp
  - Display detailed information on one or more containers

Container Stats

- **docker** container stats
  - Display a live stream of container(s) resource usage statistics
  - docker stats [OPTIONS] [CONTAINER...]
- docker stats **webservice 67b2525d8ad1**
- docker stats --format "{{.Container}}: {{.CPUPerc}}“
- docker stats --format "table {{.Container}}\t{{.CPUPerc}}\t{{.MemUsage}}"

Container Top

- **docker** container top webapp
  - Display the running processes of a container

Container Logs

- **docker** container logs webapp
- **docker** container logs -f webapp

Stop, Pause, delete

- **docker** container pause webapps
- **docker** container unpause webapps
- **docker** container stop webapps
- **docker** container rm webapps
- Stop all containers
  - **docker** container stop $(docker container ls -q)
- Remove all containers
  - **docker** container rm $(docker container ls -aq)
  - docker container prune

Container host name set

- **docker** container run –itd –name=mynginx hostname=nginxserver –rm nginx
    - Container **name** always unique. But multiple container have same host name

Container restart policy

- To configure the restart policy for a container, use the --restart flag when using the docker run command.

Flag	Description
no	Do not automatically restart the container. (the default)
on-failure	Restart the container if it exits due to an error
always	Always restart the container if it stops.
unless-stopped	Similar to always, except that when the container is stopped (manually or otherwise), it is not restarted even after Docker daemon restarts.

Container **restart policy** example

- docker container run -itd --name=nginx1 --restart=no nginx
  - If you stop the container, it will not start automatically. It's the default option
- docker container run -itd --name=nginx2 --restart=on-failure nginx
  - The container will start again if it fails due to an error
- docker container run -itd --name=nginx3 --restart=always nginx
  - Always restart the container if it stops
- docker container run -itd --name=nginx4 --restart=unless-stopped nginx
  - The container will not start again if it is stops manually. Not even container engine restarts
  - docker container stop nginx4
  - systemctl restart docker

Copy files in docker (**docker cp**)

- Copy files/folders between a container and the local filesystem
- Copy a local file into container
  - **docker** cp ./some_file CONTAINER:/work
- Copy files from container to local path
  - **docker** cp CONTAINER:/var/logs/ /tmp/app_logs
- Ex:
  - docker container cp index.html myhttpd:/var/www/http

Docker port mapping

- By default, when you create or run a container using docker create or docker run, it does not publish any of its ports to the outside world.
- To make a port available to services outside of Docker
  - use the **--publish** or **-p** flag.

Flag value	**Description**
-p 8080:80	Map TCP port 80 in the container to port 8080 on the Docker host.
-p 192.168.1.100:8080:80	Map TCP port 80 in the container to port 8080 on the Docker host for connections to host IP 192.168.1.100.
-p 8080:80/udp	Map UDP port 80 in the container to port 8080 on the Docker host.
-p 8080:80/tcp -p 8080:80/udp	Map TCP port 80 in the container to TCP port 8080 on the Docker host, and map UDP port 80 in the container to UDP port 8080 on the Docker host.

Docker port mapping

- docker container run -itd --name=mynginx --p 8080:80 nginx
  - Host port: 8080
  - Container port: 80

Free Disk Space on Host

- df –h
- `docker container prune`
- `docker image prune`