- **Goal**: Uniformed software packing and execution
  - ↳ standard packaging
  - ↓ isolate application
  - ↳ light-weight
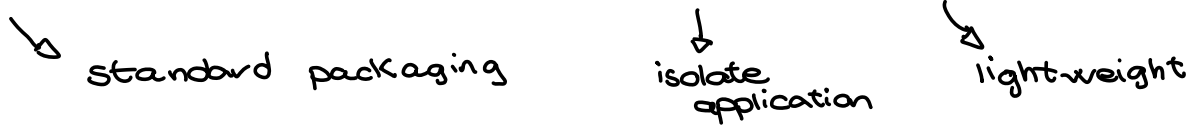
- **CoW, Copy-on-write strategy**
  - ↓ quando si avvia un container un piccolo container layer viene aggiunto
  - ↳ Small as possible
  - ↳ creazione di nuove layer
  - ↳ se un file é cambiato durante la creazione di una immagine
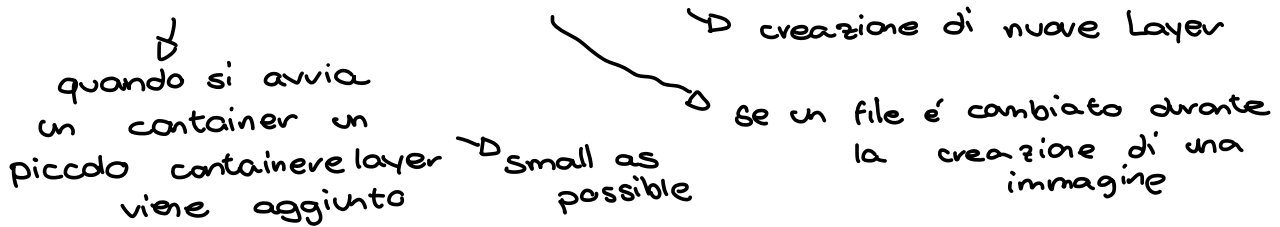
- **Exec** → to execute a command with in a running container

- **Hard limit** → container to use no more than a given amount of the resource

- **Soft limit** → container to use as much resource as it need unless certain condition are met

- **Volumes** → directory mounted into the container
  - ↳ data on remote host or cloud providers

- **Bind mount** → referenced by its path

- **ONBUILD** → command as an instruction the parent Docker file gives to child Dockerfile

- **VOLUME** → create a mount point with name and marks it as an holding externally mounted volume

- **EXPOSE** → doesn't actually publish the port, inform

- **CNM** → Sandbox , endpoint , network , cluster

un astrazione che descrive networke life-cycle

container's network stack configuration

veth, only one networke one sandbox

group of endpoint

group of Docker host

→ **Network naspace** → isolated network stack

- **host** → direct and full access to the host namespace

- **overlay** → networking that can span multiple host, L3 encapsulation

- **Service** → are containers in production

↳ a set of replicas of container

Scaling
change num of replicas

only service that have no bounded port can be scaled using compose

- **Docker Swarm** → a group of machines, running Docker in a cluster

security , networking, state e initialization

the desired state → controll loop