

# Docker.

Docker es una plataforma creada con el fin de desarrollar, implementar y ejecutar aplicaciones dentro de contenedores.

<https://docs.docker.com/reference/>

DESCANSO. Volvemos a las 10:30

# Docker. Arquitectura

- Cliente-servidor
- Cliente comunica con el “daemon”
- Daemon: build/run/deploy
- Cliente comunica con el daemon con REST API / UNIX Sockets.

## Docker Registry

- Almacena las imágenes (docker hub por defecto)
- Registro Privado

## Docker Imágenes

- BAsE: Read-only
- Imagen basada en otra imagen
- Dockerfile : crea tus propias imagenes

Docker Container: Instancia de una imagen con persistencia y red.

# Docker. Arquitectura

Especificación Docker:

OCI: Open Container Initiative

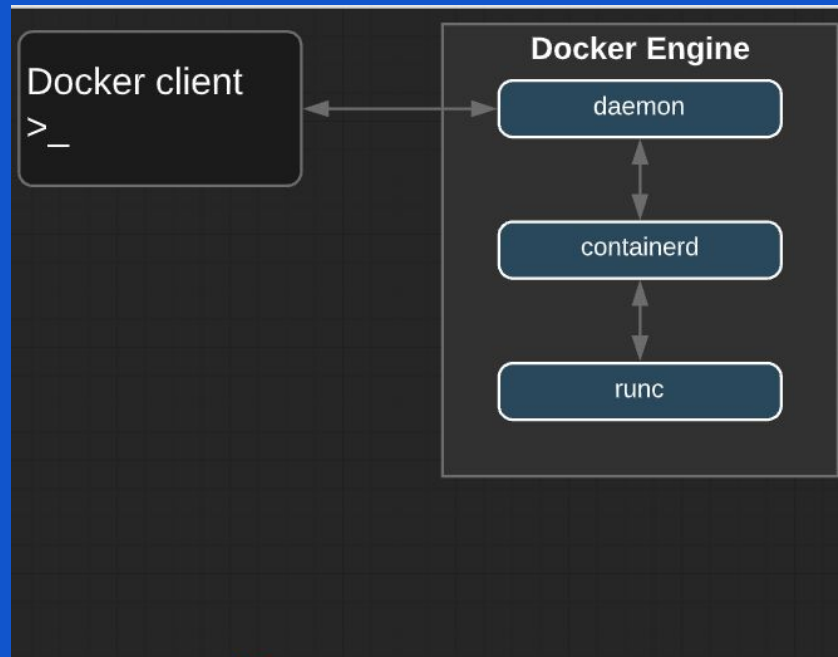
- Especificaciones de imagen
- Especificaciones de Runtime

runc

- OCI Runtime
- CLI
- Crea contenedores

Containerd:

- Ciclo de vida de contenedores (start/stop/pause/delete)
- Gestión de imágenes



# Docker. Network

Container Network Model

libnetwork (implementation of CNM): ingress, load balancing, etc

Drivers:

- bridge (software). Default
- host:
- overlay: multiple hosts
- macvlan: mac address to container

# Docker. Almacenamiento

## No Persistente

- Efímero
- Creado en cada contenedor, eliminado después
- Almacenamiento local: /var/lib/docker
- Drivers: overlay, aufs, etc

## Persistente:

- Volúmenes
- No acoplados a contenedores
- “Mount” en el contenedor.
- Eliminar el contenedor no elimina el volumen
- Local Driver, otros drivers: Blocks (HPC como Amazon EBS), NFS, Object Storage (Amazon S3)

# Docker. Dockerfile

Dockerfile: instrucciones para crear una imagen

- Capas (Layers)
- docker image build

Mejores prácticas:

- Contenedores son efímeros
- 12 Factor app

Evitar ficheros innecesarios (.dockerignore)

multi-stage builds

no instalar paquetes innecesarios

Desacoplar aplicaciones

# Docker. Dockerfile

Instrucciones:

FROM: Inicializa y crea una nueva “Imagen Base”

RUN: Ejecuta comandos en la nueva capa

CMD: Punto de entrada del contenedor. Solo uno por contenedor. Mejor usado para configurar.

LABEL: Etiquetas o metadata

EXPOSE: Puerto/s usados por el contenedor en ejecución

ENV: Variables de ambiente

ADD: Copia ficheros, directorios <src> <dest>

WORKDIR: Configura el directorio donde ejecutar RUN, CMD, COPY o ADD

ARG: Variables necesarios en “build-time” (ARG SRC\_DIR=/var/node)

ONBUILD:

USER: Cambiar usuario

HEALTHCHECK: Como hacer un test de un contenedor

SHELL: definir shell

VOLUME: Create un “mount point”: VOLUME [“/usr/share/gnix/html”]

ENTRYPOINT: >Define un contenedor con un comando ejecutable

`docker image build -t myorg/myapp:v1 --build-arg SRC_DIR=/var/node`



# Docker. Ejercicio

