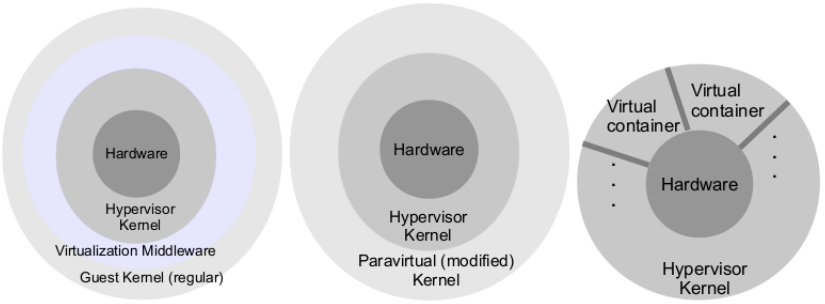
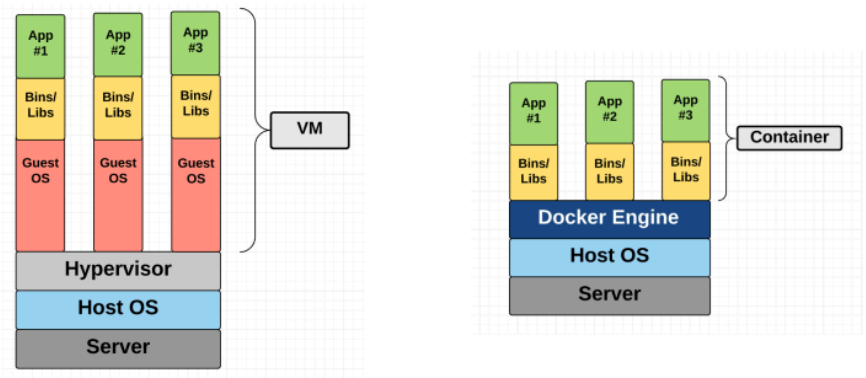
RESUM DOCKERS:

Kernel. Kernel manages hardware and essential operations with the hardware.





Docker history <img> : llista les capes que te la imatge i la seva mida.

```
joel@joelot:~$ sudo docker history netcat4
IMAGE          CREATED       CREATED BY                                      SIZE      COMMENT
f00b82ffa03c   3 days ago    EXPOSE map[12345/tcp:{}]                        0B        buildkit.dockerfile.v0
<missing>      3 days ago    ENTRYPOINT ["/bin/netcat" "-lkp" "12345" "-e…   0B        buildkit.dockerfile.v0
<missing>      3 days ago    RUN /bin/sh -c apt update && apt install -y …   43.7MB    buildkit.dockerfile.v0
<missing>      6 weeks ago   /bin/sh -c #(nop)  CMD ["/bin/bash"]            0B
<missing>      6 weeks ago   /bin/sh -c #(nop) ADD file:a3272496fda5a8d02…   78.1MB
<missing>      6 weeks ago   /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      6 weeks ago   /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      6 weeks ago   /bin/sh -c #(nop)  ARG LAUNCHPAD_BUILD_ARCH     0B
<missing>      6 weeks ago   /bin/sh -c #(nop)  ARG RELEASE                  0B
```

Example (Bad):

```
FROM ubuntu

COPY hello.c /
RUN apt update
RUN apt install -y gcc
RUN gcc /hello.c -o /hello
RUN apt remove -y gcc
```

Example (Good):

```
1  FROM ubuntu
2
3  COPY hello.c /
4  RUN apt update \
5  && apt install -y gcc \
6  && gcc /hello.c -o /hello \
7  && apt remove -y gcc
```

Docker diff : mostra tots els canvis que s'han fet respecte la img original.
Docker run -it <container> /bin/bash

Si fem docker inspect <container> podem veure diferents paràmetres del contenidor, entre ells els paràmetres de red i la seva IP. Per defecte es connecta a bridge amb 172.0.0.X

Pots borrar tots els contenidors amb **docker container prune**

Pots executar el container en segon pla dettacher amb **docker run -dit <container>**
Pots fer volums amb **docker run -it -v /dir/local:/dir/docker <container>**

Per crear imatges amb dockerfile: **docker build -t "name" .**

```
FROM ubuntu
RUN apt update && apt install -y netcat-traditional
ENTRYPOINT ["/bin/netcat", "-lkp", "12345", "-e", "/bin/cat"]
EXPOSE 12345
```

Llavors fas **docker run -dit -p12345:12345 "nom_img"**

Expose sols informa que el contenidor escolta a aquest port pero no mapeja ports !!!

DOCKER NETWORK:

- Al crear un contenidor li dona una ip 172.17.0.X i el posa al bridge (es pot veure amb tots els altres aqui).
- Al crear una network amb **docker network create --drive brige my-net** se li asigna una net 172.X.0.0 on pots afegir contenidors a ella especificant al **docker run --network "my-net" <img>**
- Pots entrar també amb la id del container o amb el nom si li has especificat al run amb **--name "cont-name"**

DOCKER COMPOSE:

```
version: '3'
services:
  nginx:
    image: nginx
    networks:
      - my-net2
  lynx:
    build: .
    image: lynx2
    entrypoint: /bin/bash
    stdin_open: true
    networks:
      - my-net2
networks:
  my-net2:
    driver: bridge
```

Podem crear networks i adjuntar-hi els contenidors. També podem crear contenidors a partir de Dockerfiles.

Especificar en un .env les variables i al compose:
Enviroment:

- **NGINX_PORT =${port}**

1. Run the app (in background):
   `docker-compose up -d`
2. Verify status and debugging:
   `docker-compose logs`
   `docker-compose ps`
3. After changes in the image/code:
   `docker-compose build`
4. Use down+up to run the app again:
   `docker-compose down`
   `docker-compose up -d`
5. To start/stop the same containers:
   `docker-compose stop`
   `docker-compose start`
6. To remove the containers:
   `docker-compose rm`

Command	Description
docker create IMG	Creates (not start) a container from an image IMG
docker create -it debian /bin/bash	Creates an interactive container from debian
docker start ID/name	Starts a previous created container
docker run IMG	Runs (creates and starts) a container from an image
docker run -dit debian /bin/bash	Runs an interactive container in background
docker stop ID/name	Stops a running container
docker ps	Lists running containers
docker ps -a	Lists running and stopped containers
docker exec -it mycont /bin/bash	Executes a bash in the container and attaches it locally
docker attach mycont	Attaches locally the streams of a running container
docker rm mycont	Removes mycont
docker container prune	Removes all stopped containers
docker inspect ID/name	Provides information about the container
docker diff ID/name	Lists the filesystem changes regarding the image
docker logs ID/name	Lists the commands executed