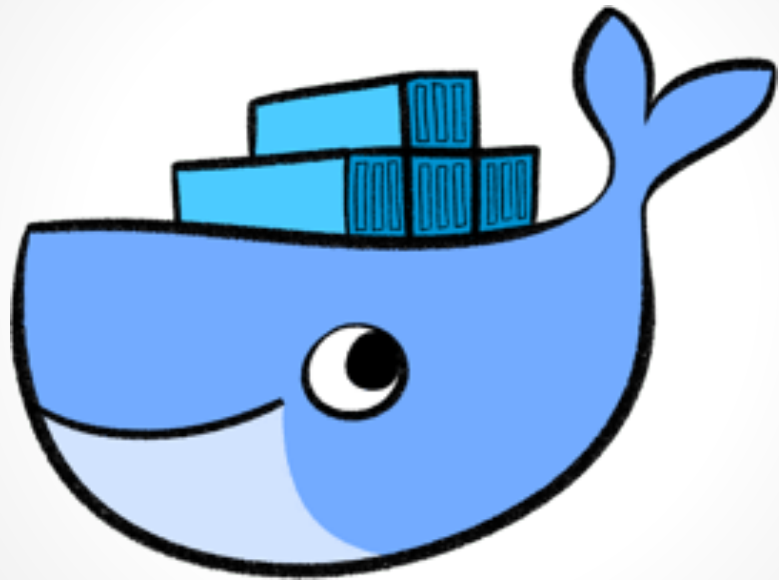


Curso de Docker



Comunidad de Castilla la Mancha

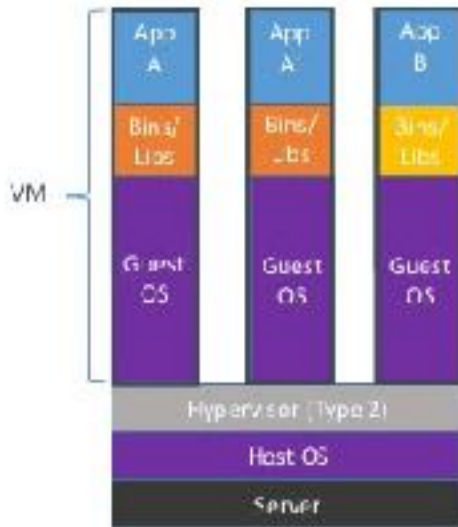
Repaso

Guía del Curso

- Introducción a Docker
- Arquitecturas de Microservicios
- Construcción de Imágenes
- Desarrollo con Contenedores
- Integración Continua
- Introducción a Kubernetes

Características Docker

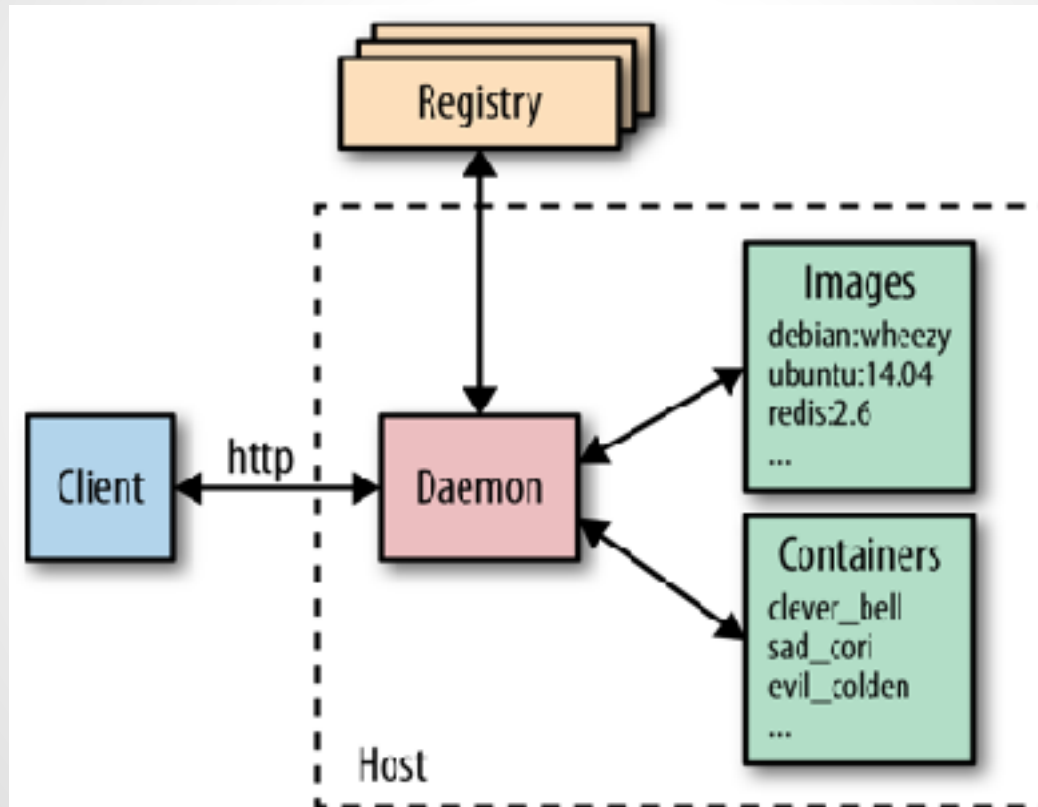
Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries



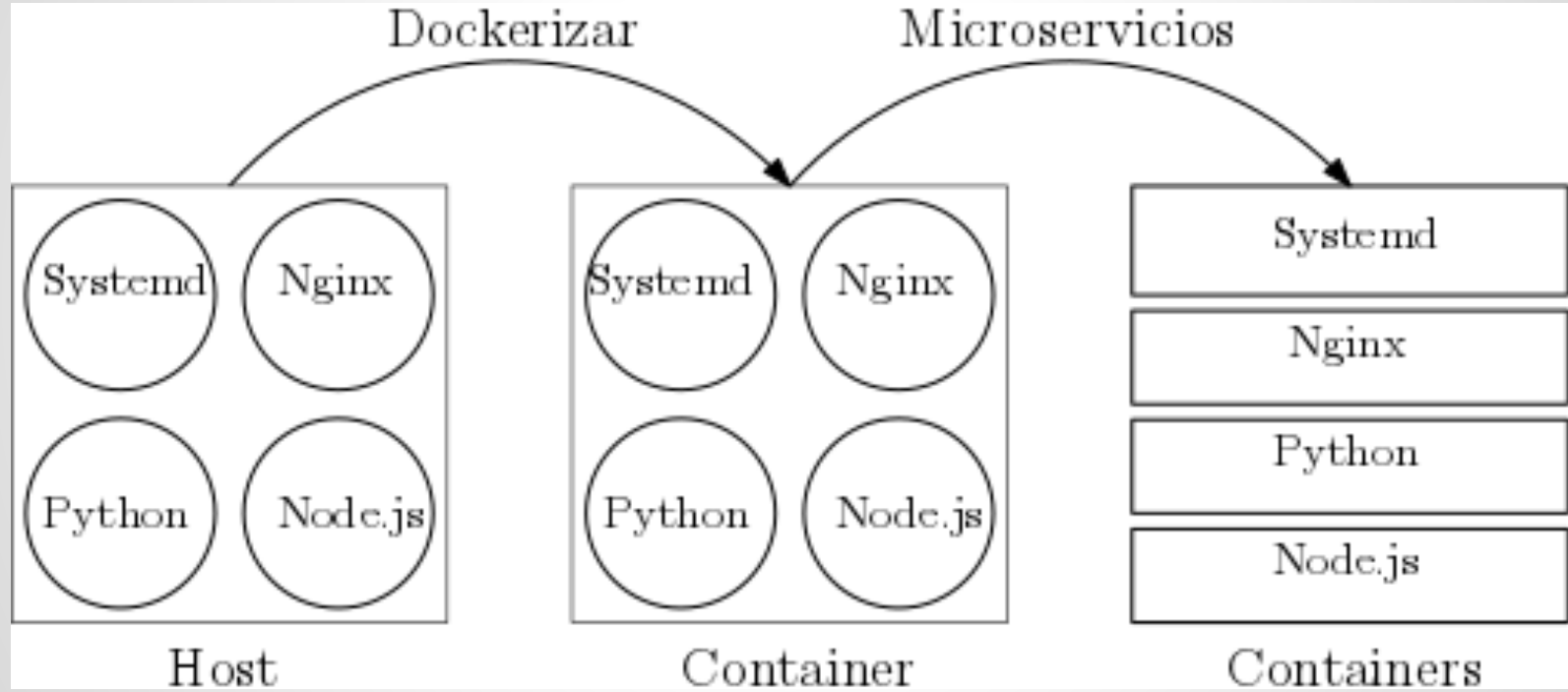
Componentes Docker



Comandos Comunes para Contenedores

- docker info
- docker version
- docker run (docker container run)
- docker ps (docker container list)
- docker inspect (docker container inspect)
- docker stop (docker container stop)
- docker start (docker container start)
- docker rm (docker container rm)
- docker cp (docker container cp)
- docker exec (docker container exec)
- docker logs (docker container logs)
- docker stats (docker container stats)
- docker system prune
- \$DOCKER_HOST

Relación entre Docker y Microservicios



Guía del Curso

- Introducción a Docker
- Arquitecturas de Microservicios
- Construcción de Imágenes
- Desarrollo con Contenedores
- Integración Continua
- Introducción a Kubernetes

Construcción de Imágenes

- Docker Build y Dockerfile
- La Caché de Docker
- Buenas Prácticas
- Comandos Comunes para Imágenes.

Docker Build y Dockerfile

Dockerfile

```
FROM ubuntu # Layer 0  
MAINTAINER Florian Lopes # Layer 1  
RUN mkdir -p /some/dir # Layer 2  
RUN apt-get install -y curl # Layer 3
```

FINAL IMAGE

Layer 3 (Image 3)

Layer 2 (Image 2)

Layer 1 (Image 1)

Layer 0 (Base image)

Docker Build y Dockerfile

- FROM (from my image)
- RUN
- CMD (["echo", "\$HOME"])
- EXNTRYPOINT
- LABEL key=value
- EXPOSE
- ENV key=value
- ADD src dest
- VOLUME /path
- USER user[:group]
- WORKDIR
- ARG name[=default]
(FROM ubuntu:\${tag})
- ONBUILD
- STOPSIGNAL
- HEALTHCHECK (see docs)

La Caché de Docker

```
FROM ubuntu:latest
RUN apt-get update -y
RUN apt-get install -y python-pip python-dev
WORKDIR /app
ENV DEBUG=True
EXPOSE 80
VOLUME /data
COPY . /app
RUN pip install -r requirements.txt
ENTRYPOINT ["python"]
CMD ["app.py"]
```

Comandos Comunes para Imágenes

- docker login
- docker build (context, tag, cache...)
- docker pull
- docker image ls (docker images)
- docker inspect
- docker image rm (docker rmi)

Docker Compose

<https://github.com/docker/compose/releases>

```
curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-  
`uname -s`-`uname -m` -o /usr/local/bin/docker-compose  
chmod +x /usr/local/bin/docker-compose
```

<https://docs.docker.com/compose/compose-file/#build>

Buenas Prácticas

- Usa .dockerignore
- Un proceso por contenedor
- Imágenes de tamaño reducido:
 - docker-castilla-la-mancha/flask-ubuntu vs docker-castilla-la-mancha/flask-alpine
- Minimiza en número de capas:

```
RUN apt-get update
RUN apt-get install -y bzip2
RUN apt-get install -y cvs
RUN apt-get install -y git
RUN apt-get install -y mercurial
```

VS

```
RUN apt-get update && \
    apt-get install -y \
    bzip2 \
    cvs \
    git \
    mercurial \
    apt-get clean
```

Buenas Prácticas

- Optimiza el uso de la caché:
 - `docker-castilla-la-mancha/flask-alpine` vs `docker-castilla-la-mancha/flask-build-cache`
- Parametriza usando argumentos

```
FROM ubuntu
ARG user=root
ARG password
RUN echo $user $password
docker build -t imagen --build-arg password=secret .
```

- Multi-stage builds
 - `docker-castilla-la-mancha/go-multi-stage`

Dockerfiles en Docker Hub

- python
- rabbitmq
- mysql
- ubuntu

Docker Java Build

<https://hackernoon.com/crafting-perfect-java-docker-build-flow-740f71638d63>