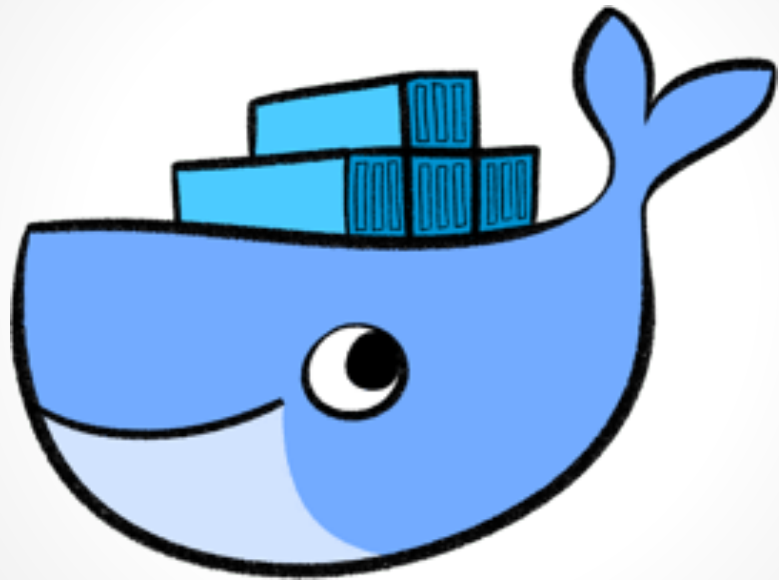


Curso de Docker



Comunidad de Castilla la Mancha

About Me

Pablo Chico de Guzmán

@chico_de_guzman

pchico83@gmail.com



About Me

Pablo Chico de Guzmán

@chico_de_guzman

pablo@okteto.com



www.okteto.com

Enfoque del Curso

- Ejemplos y buenas prácticas:
 - Dockerfile
 - docker-compose.yml
 - github.com/pchico83/docker-castilla-la-mancha
- No sobre tecnologías de bajo nivel
- No sobre despliegues en producción

Enfoque del Curso

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

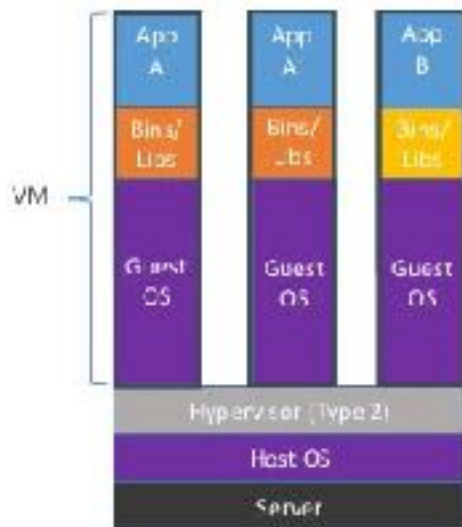
Introducción a Docker

- Características de Docker
- Componentes de Docker
- Ciclo de Desarrollo con Docker
- Instalando Docker
- Comandos Comunes para Contenedores
- Conclusiones

Características Docker

Virtualización a nivel del Sistema Operativo

Containers vs. VMs



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



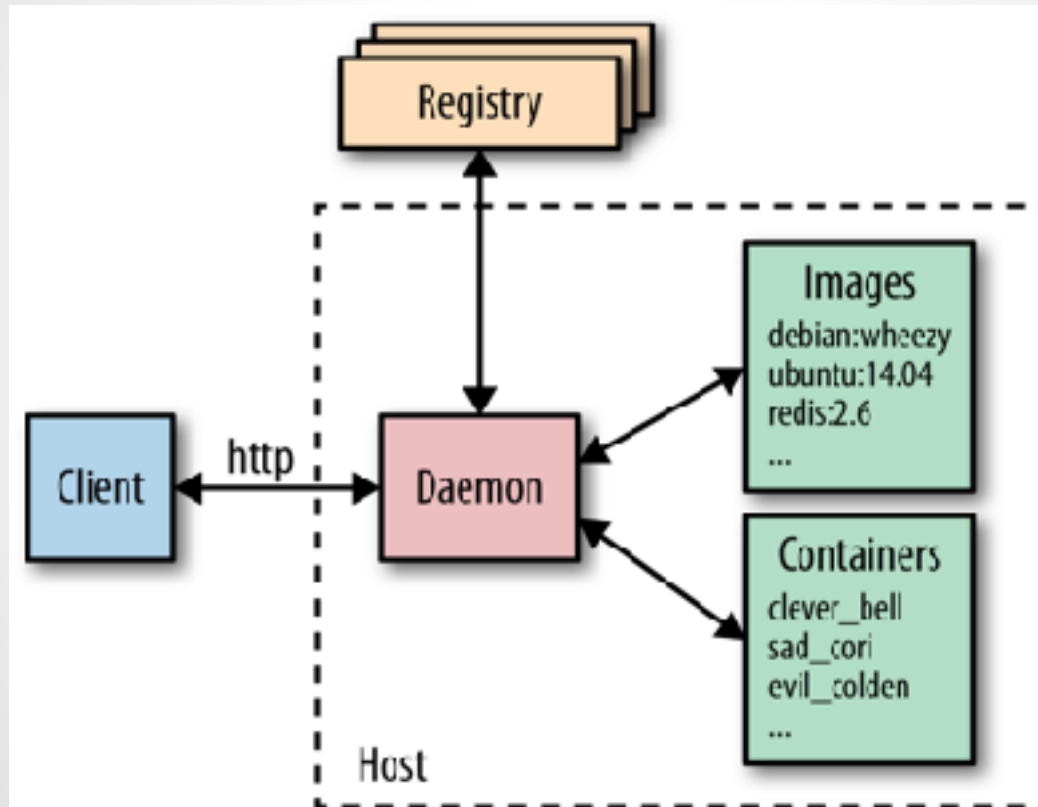
Características Docker

- Docker es open source
- Docker es una “friendly” API a contenedores:
un proceso aislado del resto de los procesos de la máquina gracias a correr sobre su propio sistema de ficheros, su propio espacio de usuarios y procesos, sus propias interfaces de red... es por ello que a veces se dice que un contenedor es una máquina virtual ligera

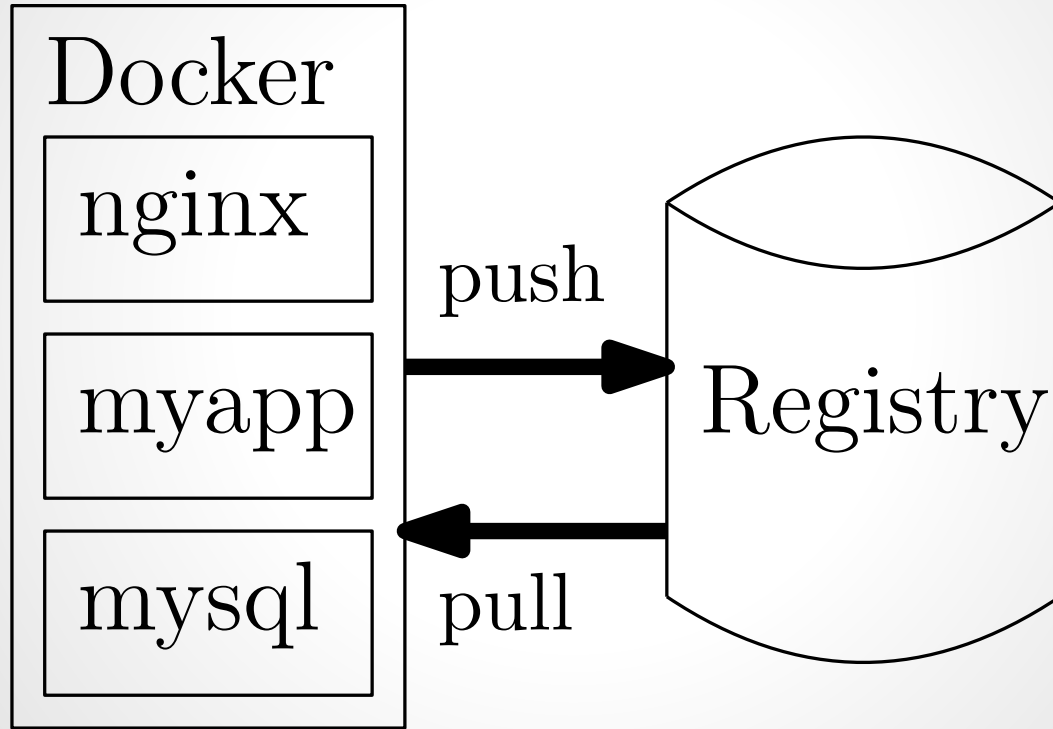
Características Docker

- “Build, Ship and Run, any app, anywhere”:
 - Portabilidad: independiente de OS/Hardware
 - Inmutabilidad: contiene todas las dependencias
 - Aislado: ligero pero menos seguro que VMs

Componentes Docker



Ciclo de Desarrollo



Instalando Docker

- Docker for Mac:

<https://docs.docker.com/docker-for-mac/install/>

- Docker for Windows:

<https://docs.docker.com/docker-for-windows/install/>

- Docker for ubuntu:

<https://docs.docker.com/engine/installation/linux/docker-ce/ubuntu/>

```
curl -fsSL get.docker.com -o get-docker.sh  
sh get-docker.sh
```

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

Comandos Comunes para Contenedores

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

Conclusiones

- Docker es una tecnología disruptiva que todo desarrollador debe conocer
- Ignorar Docker será una desventaja competitiva para cualquier compañía

Arquitecturas de Microservicios

- Arquitecturas SOA
- Microservicios: definición
- Relación entre Docker y Microservicios

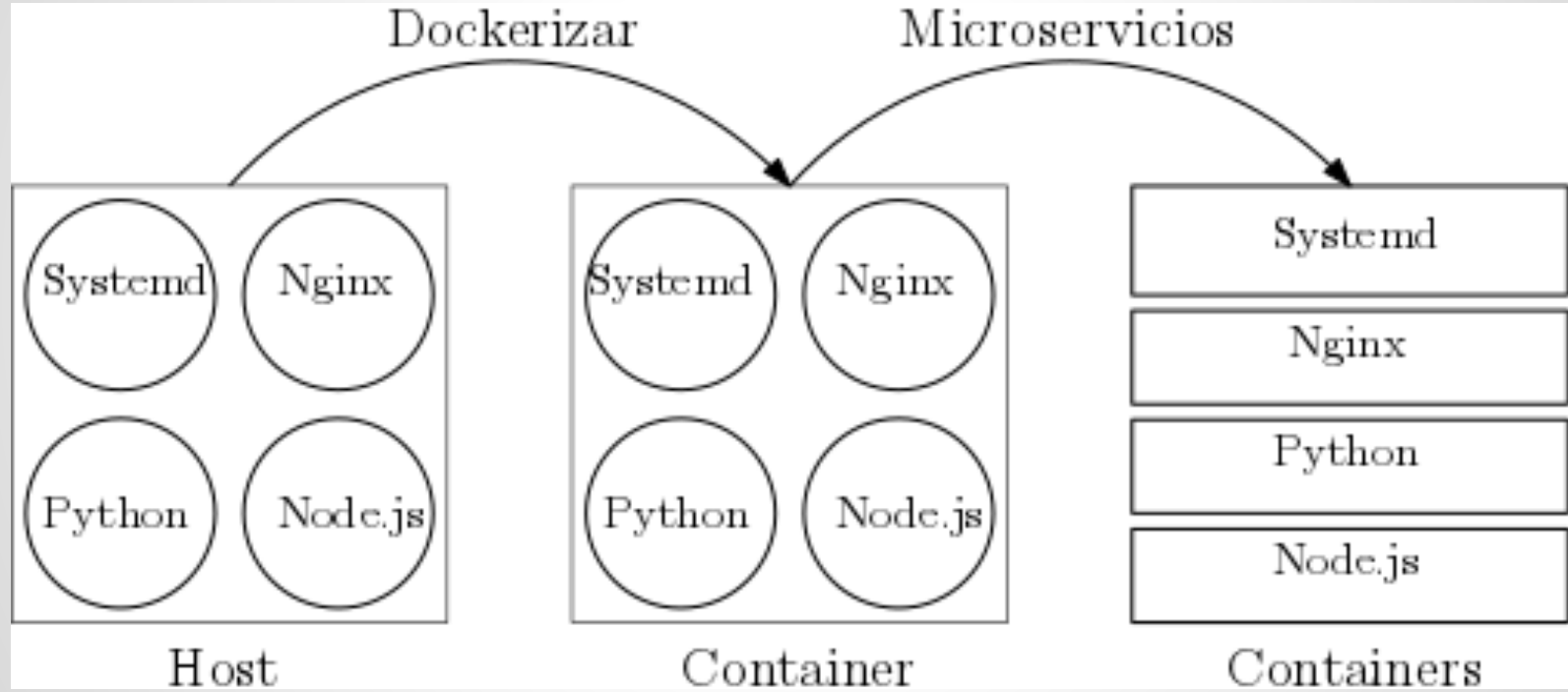
Arquitecturas SOA

- Ofrecer un contrato, una API REST, versionada
- Minimizar dependencias, optimizar su “scope”
- Abstraer su implementación
- Maximizar su reutilización
- Ciclo de vida independiente
- Abstraer localización física
- Evitar mantener estado en lo posible
- Mantener la calidad, bien testados

Microservicios: definición

- Evolución de arquitectura SOA
- Tamaño reducido
- Desarrollable en unas dos semanas
- Abordable por equipo pequeño
- Independientes
- Despliegue más sencillo
- Altamente reutilizables
- Fáciles de externalizar

Relación entre Docker y Microservicios



Docker Hub

- Registro de Imágenes de Docker
- Hay otros: AWS, Google, quay.io
- Se puede instalar registro privado
- docker login
- Imágenes Públicas
- Imágenes Privadas y Equipos
- Autobuild de imágenes