

Todo lo que debes saber

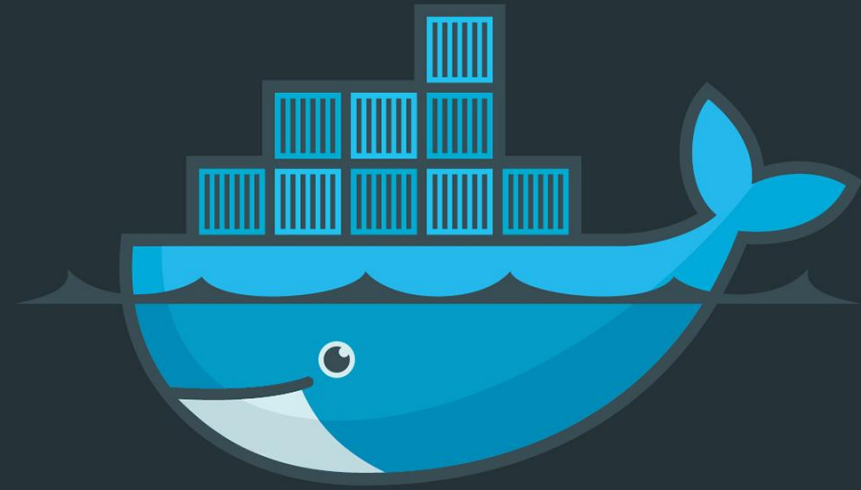
para iniciar con Docker



Fernando Sonego
Solution Architect en Algeiba Dev



CATCHIT



docker

1979: Unix V7

2000: FreeBSD Jails

2001: Linux Vserver

2004: Oracle Solaris Containers

2005: Open VZ (Open Virtuozzo)

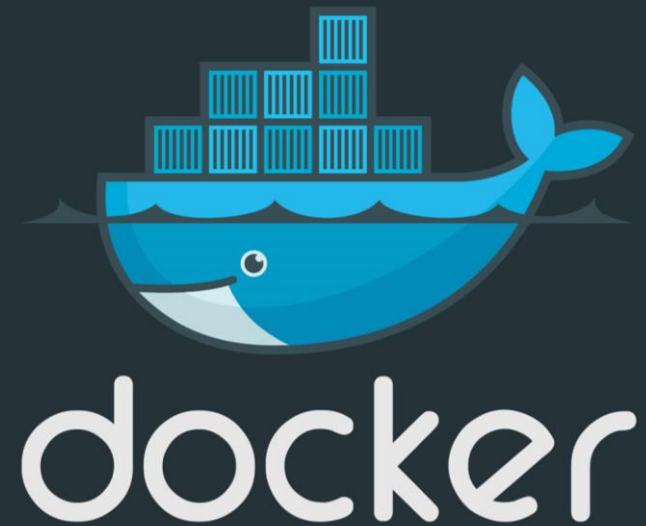
2006: Process Containers

2008: LXC

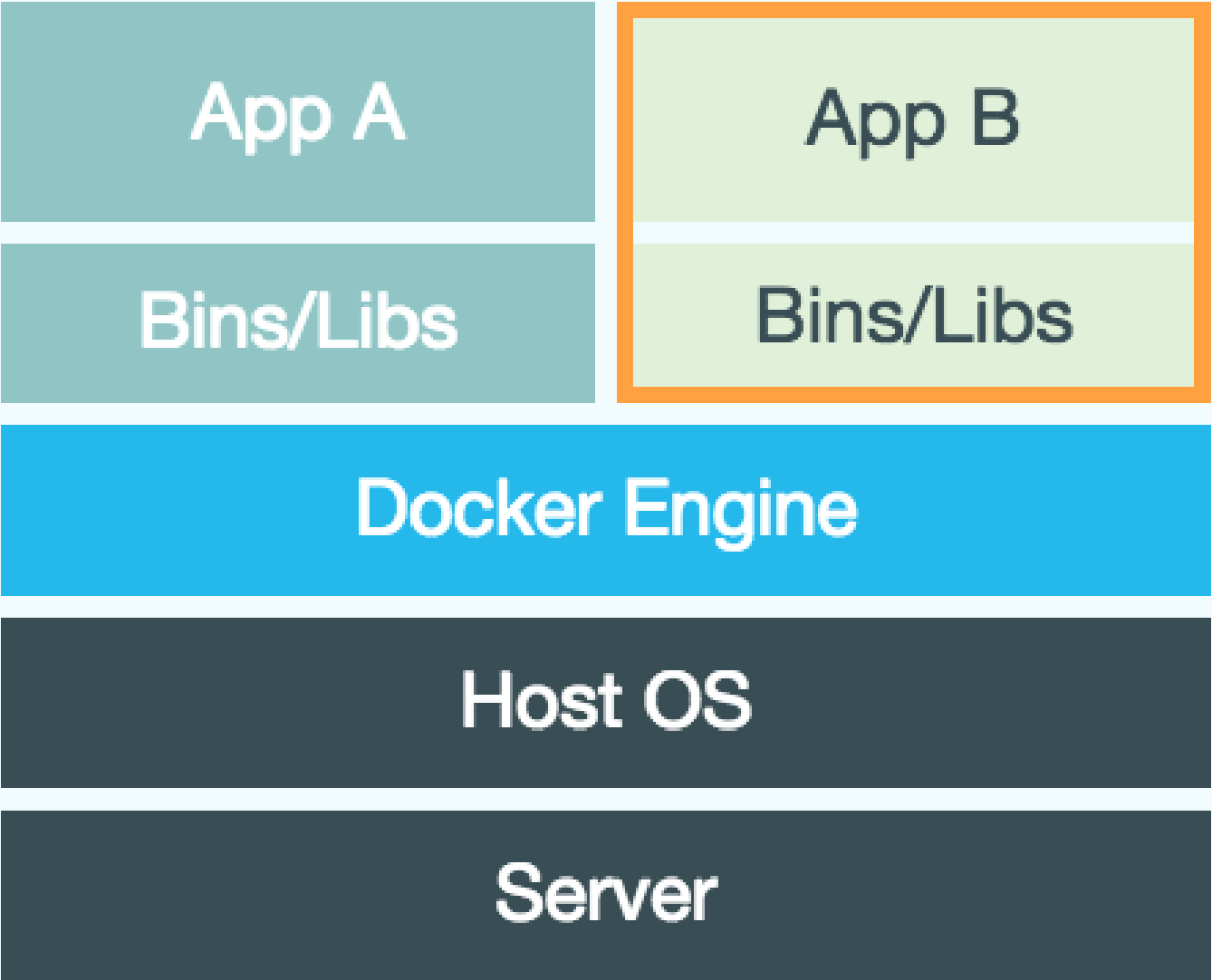
2011: Warden

2013: LMCTFY

2013: Docker and the Future





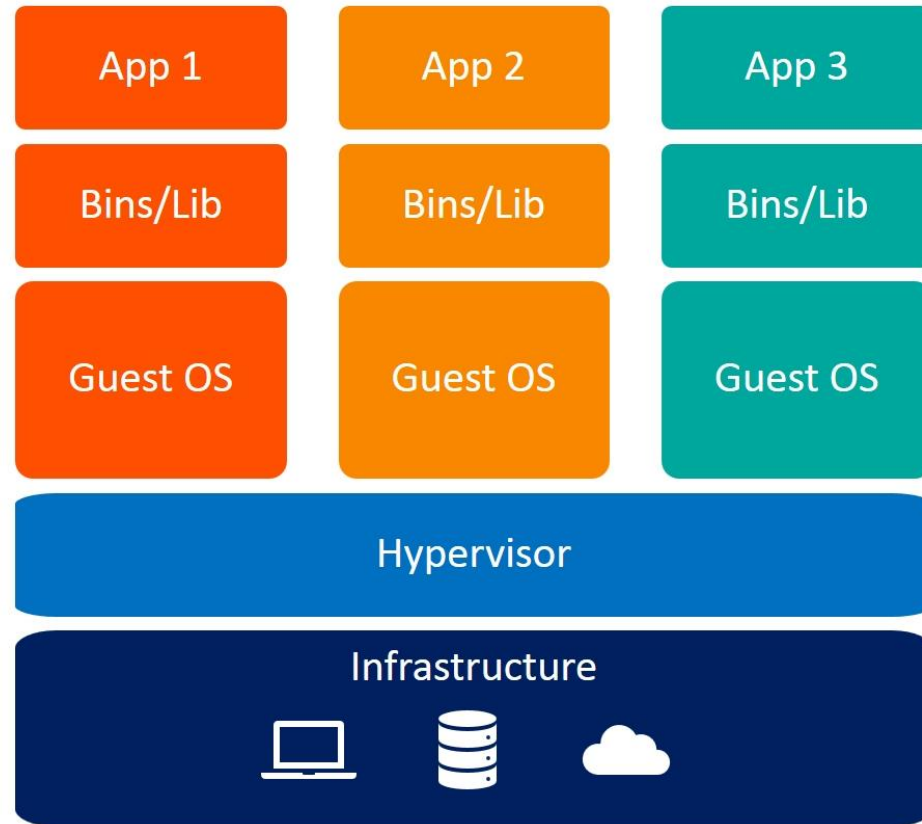


Características de Docker

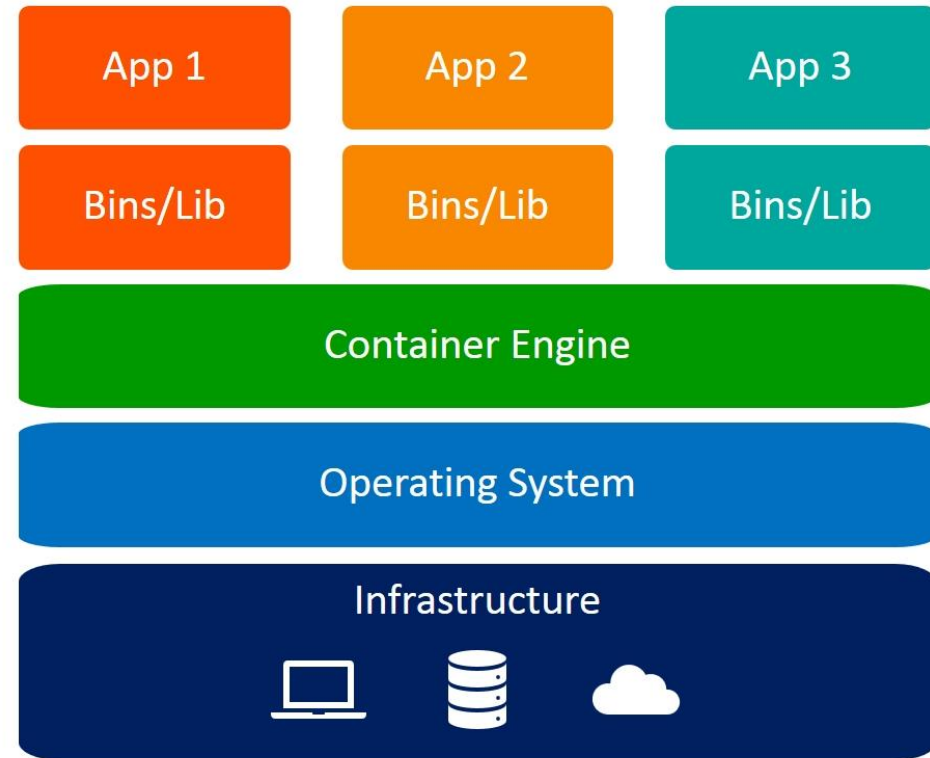
- Contenedores autogestionados
- Fiabilidad
- Aplicaciones libres de dependencias de su anfitrión.
- Posibilidad de desplegar una gran cantidad de contenedores en un mismo equipo
- Servicios disponibles en segundos
- Contenedores livianos: facilitan su almacenamiento, transporte y despliegue
- Capacidad de ejecutar casi cualquier aplicación
- Compatibilidad multi plataforma
- Compartir nuestros contenedores con el mundo.



Virtual Machine Vs Containers



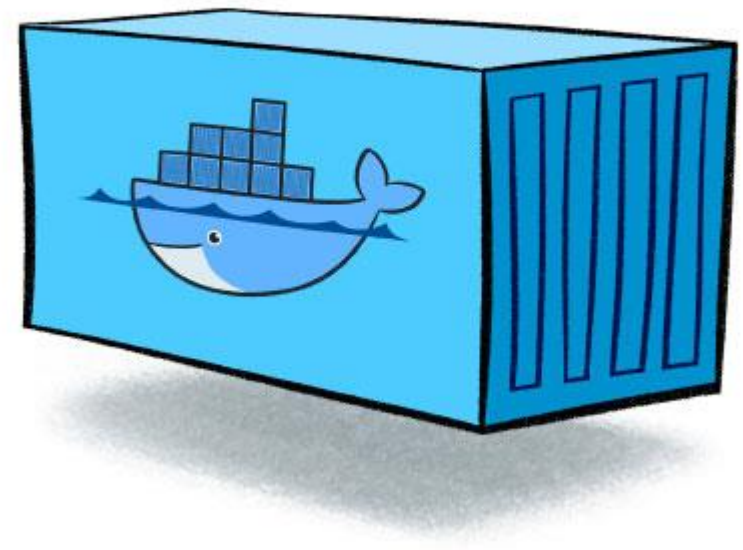
Virtual Machines



Containers

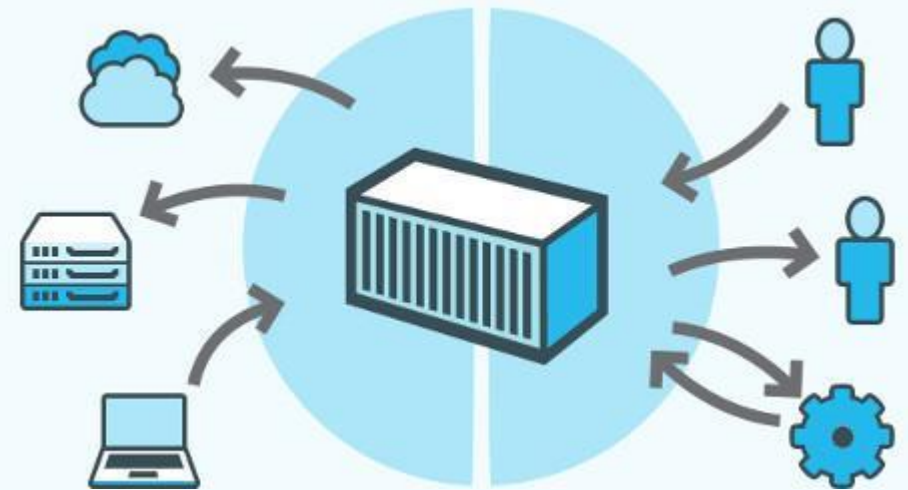
Resumiendo

Que entendimos hasta este momento.



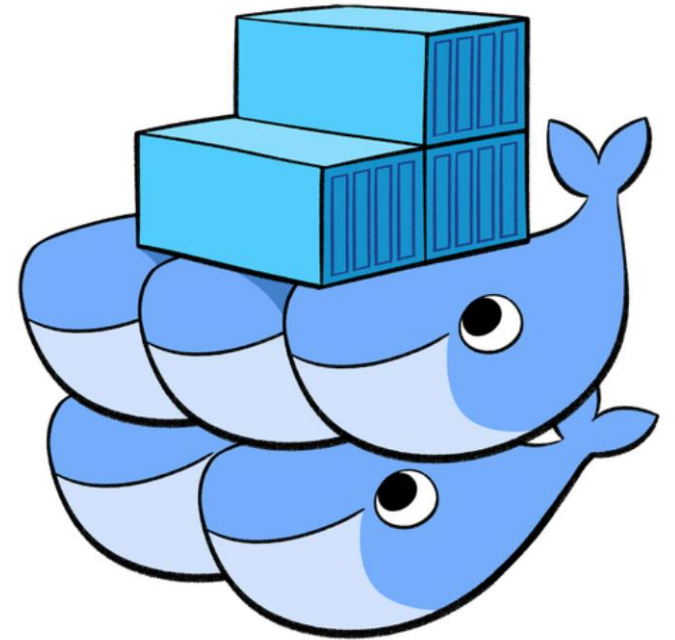
Instalación

Que necesitamos para instalarlo en Windows



Contenedores e Imágenes

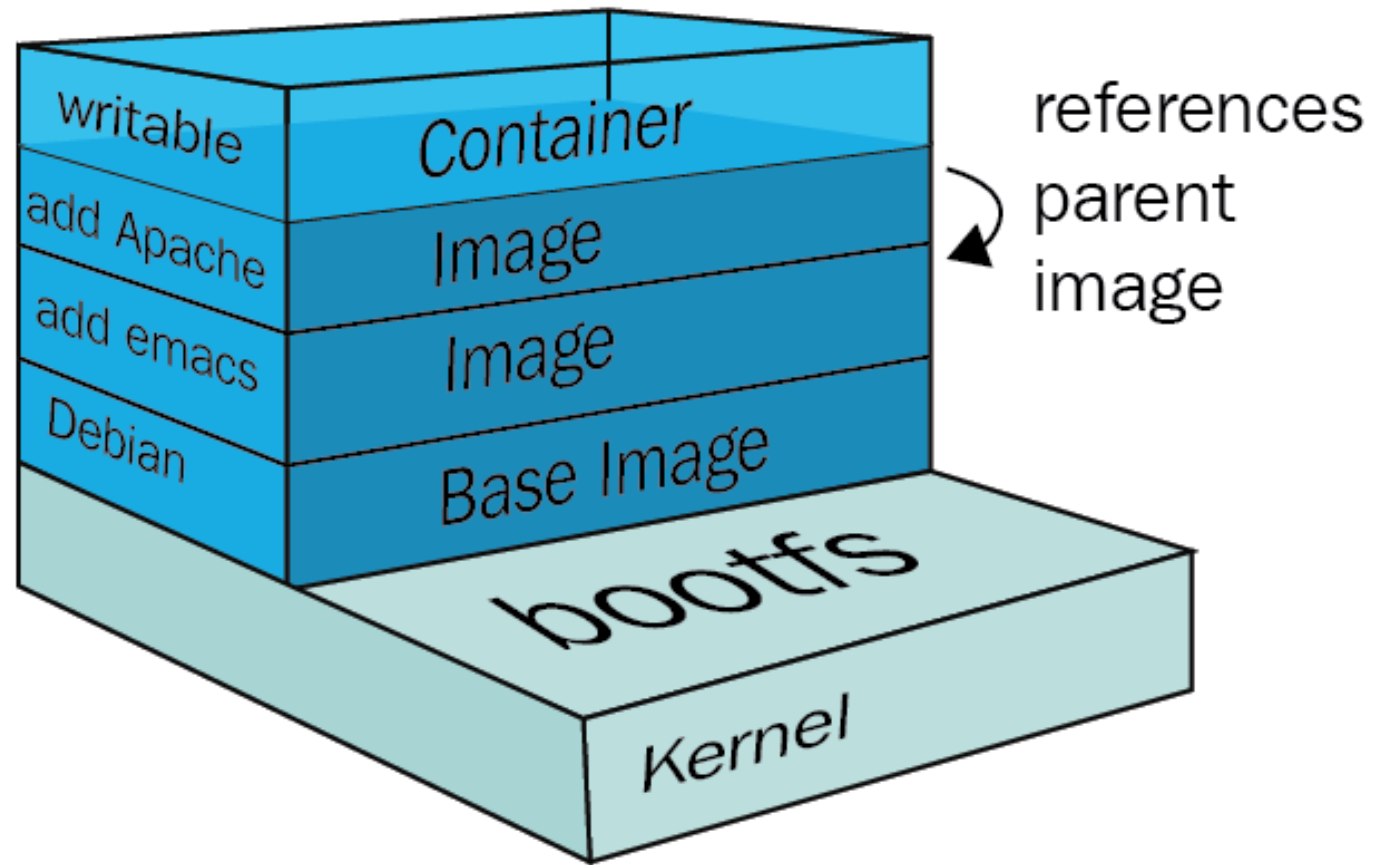
¿Cuál es la diferencia?



Contenedores e Imágenes

- **Cómo define Docker las imágenes**
 - Es una plantilla vacía o con una aplicación preinstalada para la creación de uno o más contenedores.
 - Podemos crearlas nosotros o usar creadas por la comunidad desde Docker Hub.
- **Contenedor**
 - Contiene lo necesario para ejecutar la aplicación
 - Se basa en una imagen o varias de ellas.

Imagen



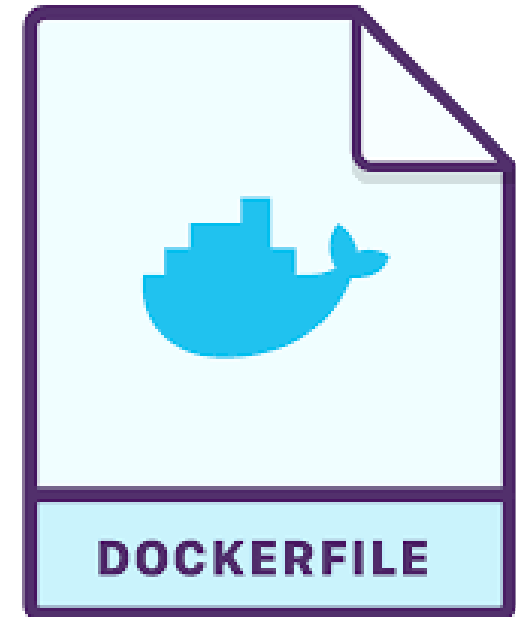


Docker Hub y Docker Search

Herramientas para imagenes

Dockerfile

La simplicidad en un solo archivo txt



¿Que es un dockerfile?

Dockerfile es un documento de texto que contiene los comandos necesarios que un usuario puede llamar por la línea de comandos para montar o armar una imagen.

Como se ve un Dockerfile

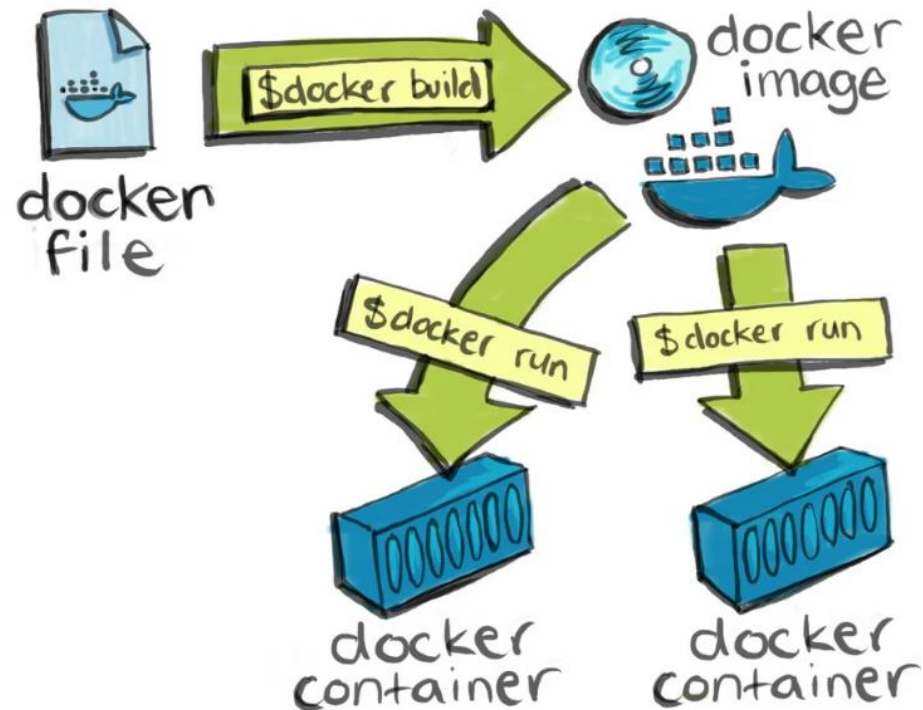
```
FROM mcr.microsoft.com/dotnet/core/aspnet:5.0-nanoserver-1809 AS base
WORKDIR /app
EXPOSE 80
EXPOSE 443
```

```
FROM mcr.microsoft.com/dotnet/core/sdk:5.0-nanoserver-1809 AS build
WORKDIR /src
COPY ["DockerApi/DockerApi.csproj", "DockerApi/"]
RUN dotnet restore "DockerApi/DockerApi.csproj"
COPY . .
WORKDIR "/src/DockerApi"
RUN dotnet build "DockerApi.csproj" -c Release -o /app/build
```

```
FROM build AS publish
RUN dotnet publish "DockerApi.csproj" -c Release -o /app/publish
```

```
FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "DockerApi.dll"]
```


Partes del Dockerfile



From: Define una imagen base para crear una nueva imagen con Dockerfile.

```
FROM mcr.microsoft.com/dotnet/core/aspnet:5.0-nanoserver-1809 AS base
```

Maintainer: hace referencia al creador del archivo

Run: Permite ejecutar comandos en la imagen antes de ser creada.

```
RUN dotnet publish "DockerApi.csproj" -c Release -o /app/publish
```

Add/Copy: Permite agregar o copiar archivos desde el equipo local a la imagen.

```
COPY . .
```

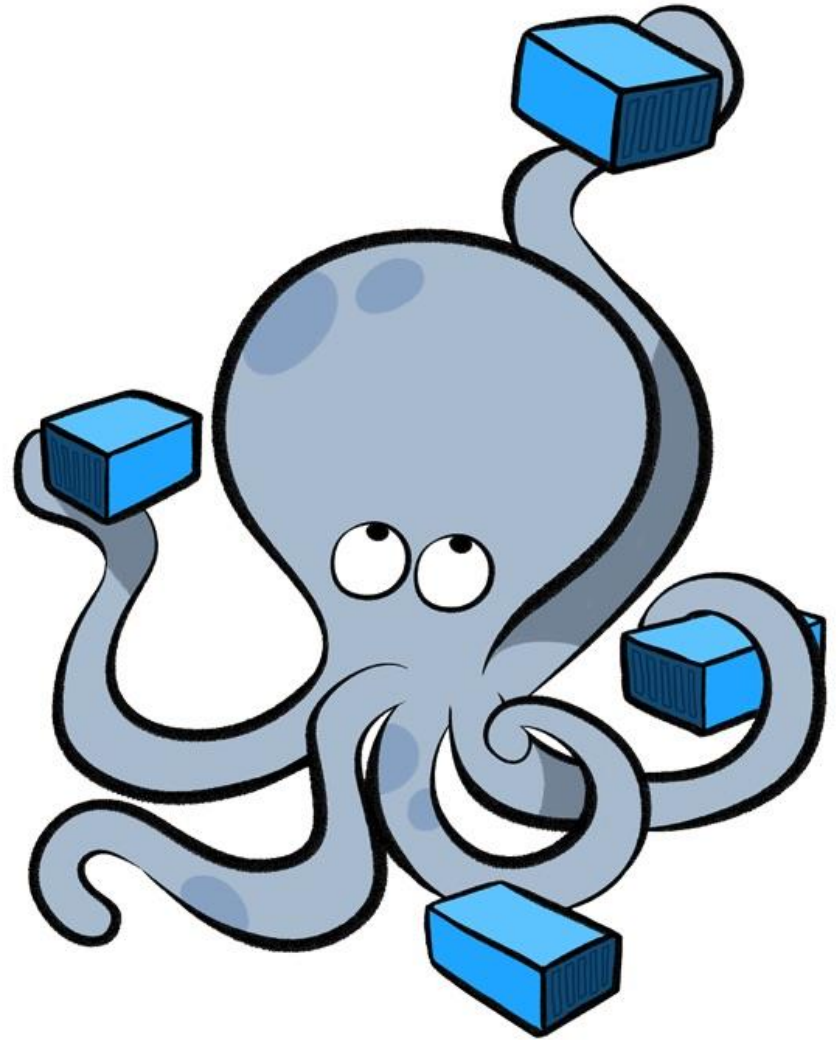
Expose: Permite exponer por defecto un puerto para el contenedor.

```
EXPOSE 80
```

CMD: Permite ejecutar acciones al crear el contenedor.

Docke Compose

Desplegando sin limites

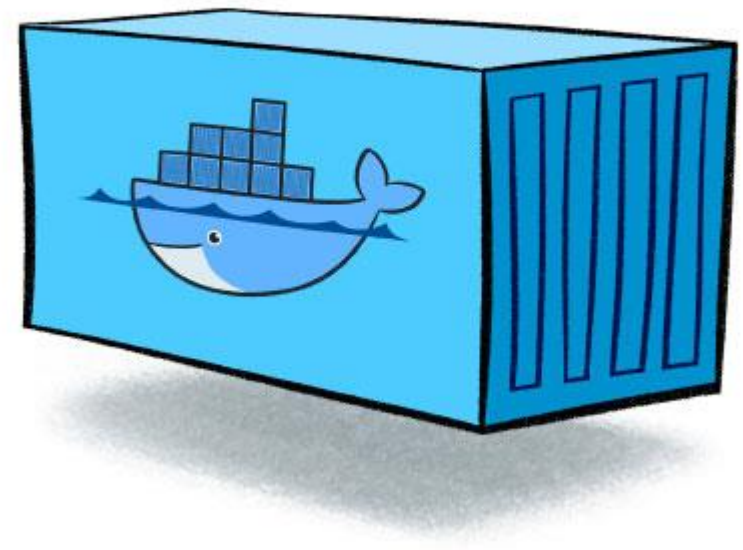


¿Que es un Docker-Compose?

Docker-Compose es una herramienta que permite simplificar el uso de **Docker**, generando scripts que facilitan el diseño y la construcción de servicios.

Demo

Wordpress y MySql en segundos



Conclusiones y Preguntas



Todo lo que debes saber para iniciar con Docker

viernes, 28 de mayo de 2020



Fernando Sonego

Solution Architect

✉ fernando.sonego@gmail.com

🐦 [@FernandoSonego](https://twitter.com/FernandoSonego)

🌐 www.withoutdebugger.com



CATCHIT algeibba 

