Fundamentos de Docker_

Contenedores

Emerald

Emerald Digital S.C.
Aplicaciones web, ciencia de datos y automatización

Juan Manuel Ruiz Aranda
jmanuel.rar@emeraldigital.com
@jmruizab

Objetivo_

Aprender como utilizar docker en nuestros proyectos.

Emerald

Emerald Digital S.C.
Aplicaciones web, ciencia de datos y automatización

Emerald Digital S.C.
Aplicaciones web, ciencia de datos y automatización

Agenda_

1. Fundamentos de Docker
2. Docker compose
3. Introducción a Docker swarm
4. Consejos y buenas prácticas

https://github.com/emeraldigital/curso-docker

En mi compu si funciona..._

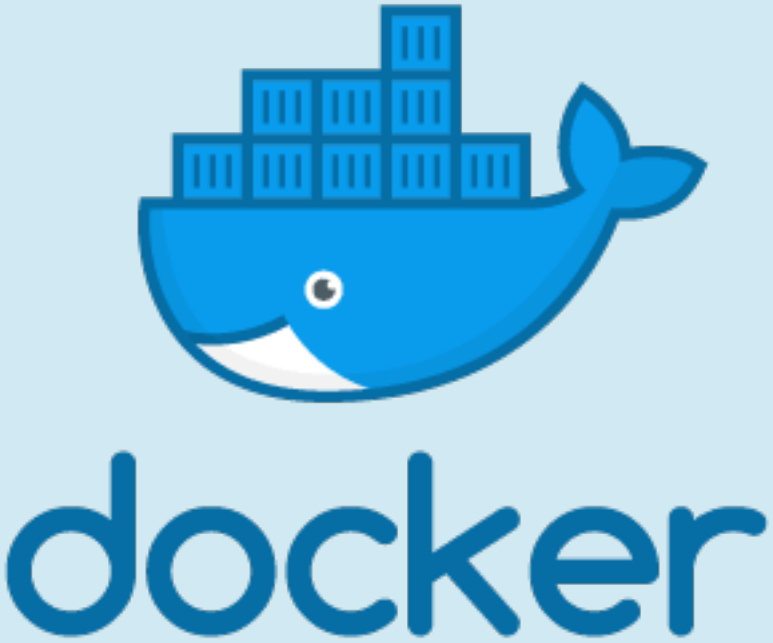
Emerald

Emerald Digital S.C.
Aplicaciones web, ciencia de datos y automatización

¿Qué es docker?_

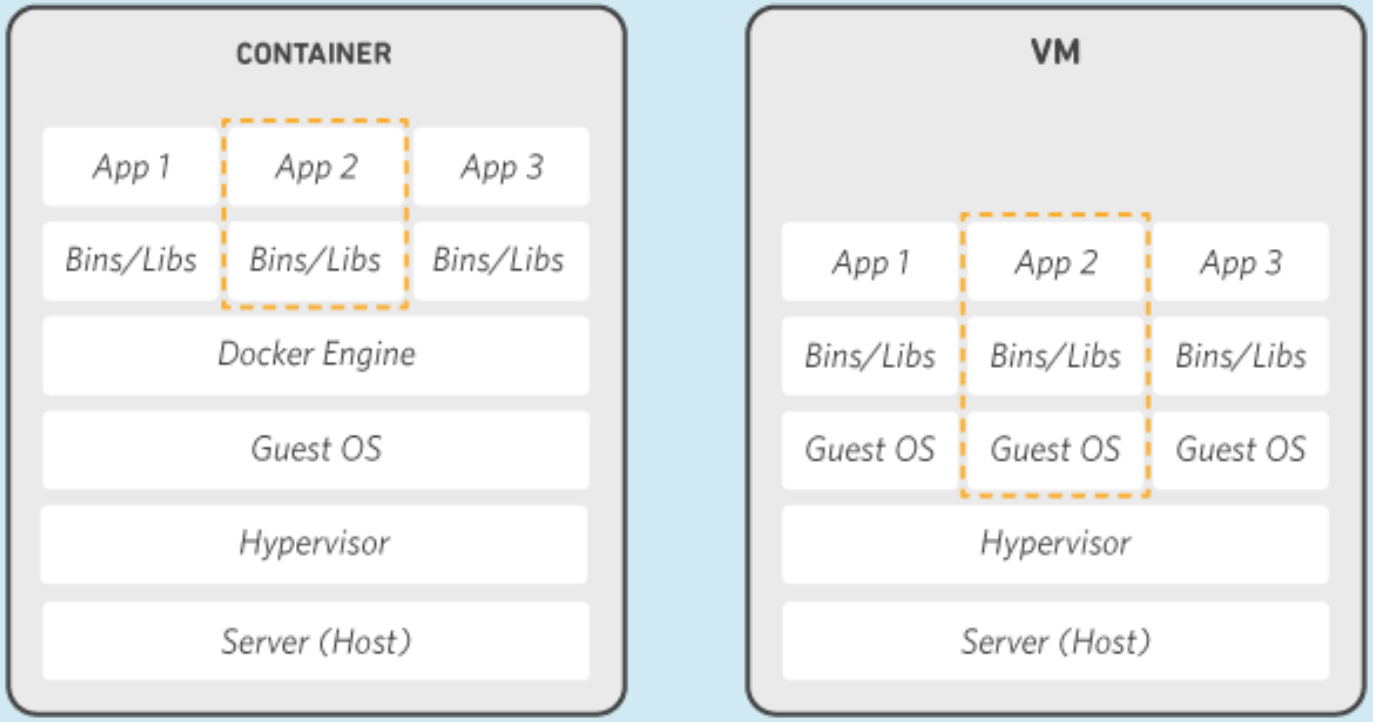
Docker es una plataforma de software que permite estandarizar los entornos de ejecución de nuestras aplicaciones.

Encapsulando nuestro código fuente y sus dependencias dentro de un entorno controlado (contenedor).



¿Contenedores?_

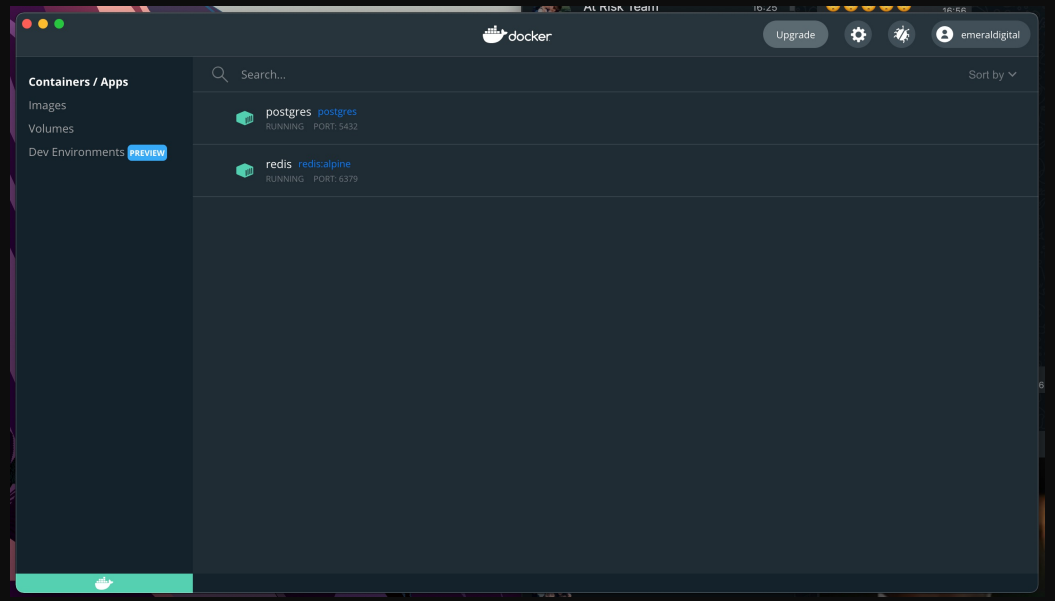
Docker virtualiza el software, las maquinas virtuales el hardware.



Instalando docker_

Docker es multi plataforma, el instalador se puede descargar desde [su página oficial.]

Proporciona un conjunto sencillo de comandos de consola y un cliente con UI para crear, iniciar o detener contenedores.



Emerald

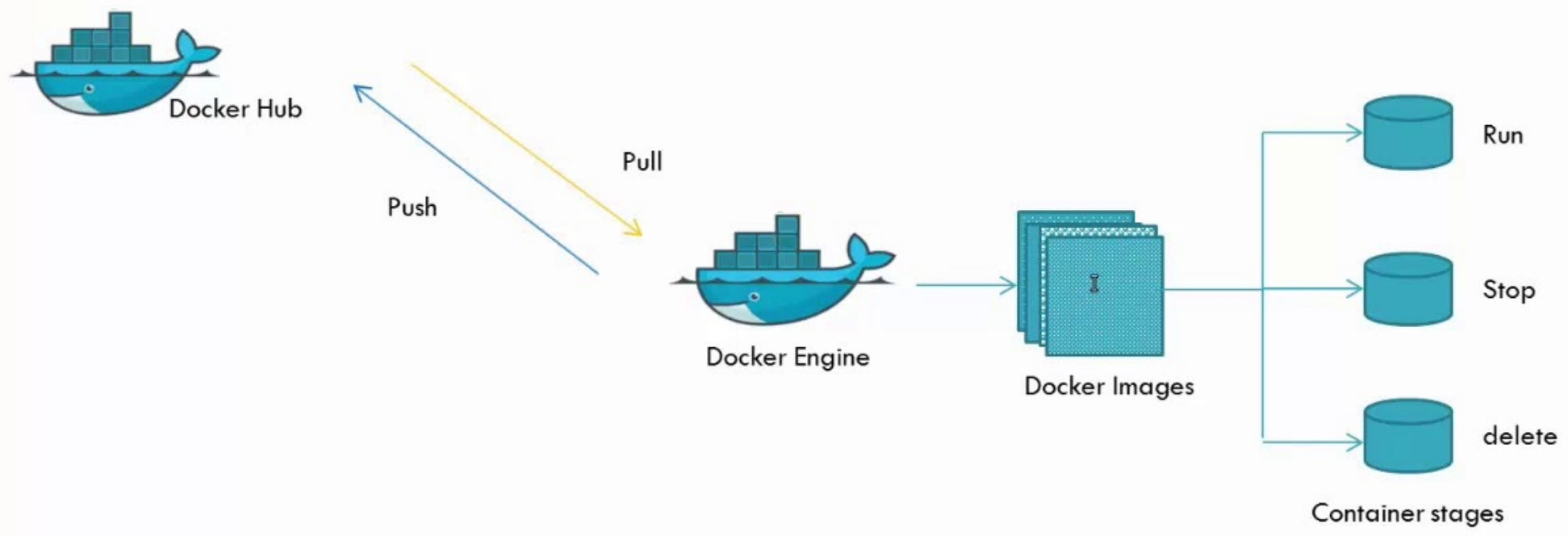
Emerald Digital S.C.
Aplicaciones web, ciencia de datos y automatización

¿Docker Hub?_

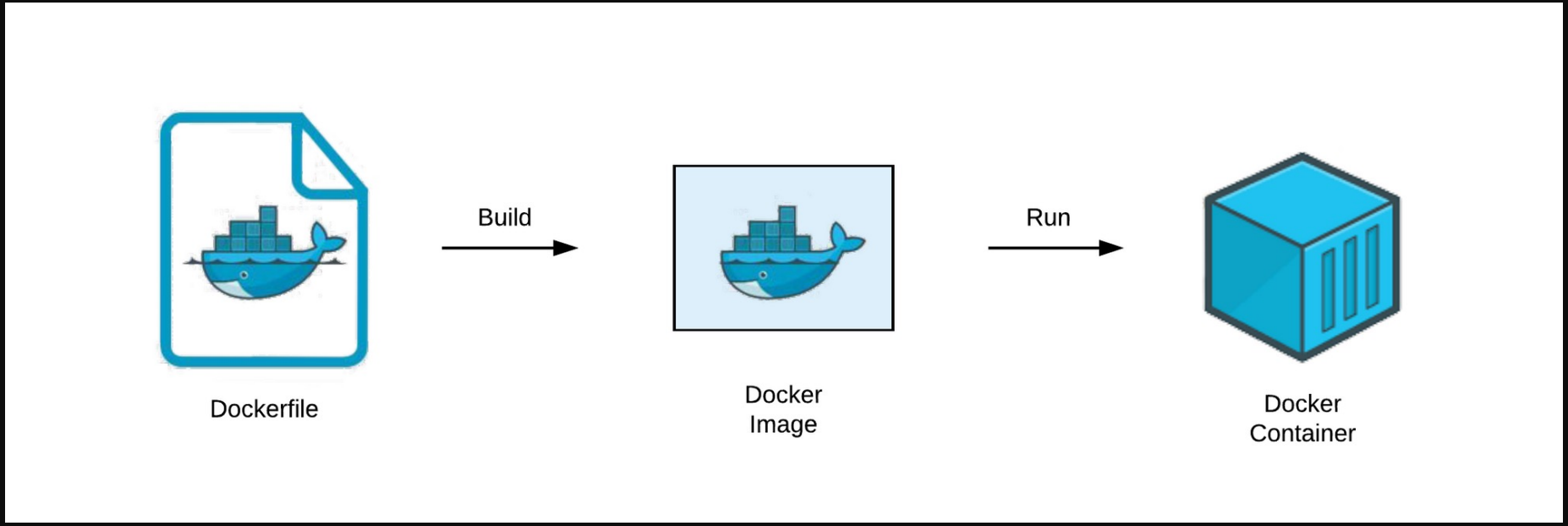
Plataforma colaborativa, similar a github.

→ Buscar y publicar imágenes de docker.
→ Integración con API's y webhooks.

Flujo de trabajo_



Funcionamiento de docker_



Dockerfile_

```
FROM ubuntu:16.04
# MAINTAINER someuser@somedomain.com
RUN apt-get update
RUN apt-get install -y mysql
CMD echo "My first image created."
```

Dockerfile reference

Es un archivo de texto plano que define las instrucciones necesarias para que nuestra aplicación se ejecute.

El Dockerfile es transformado por el engine de docker en una imagen (`docker build`).

Dockerfile_

```
FROM ubuntu:16.04
# MAINTAINER someuser@somedomain.com
RUN apt-get update
RUN apt-get install -y mysql
CMD echo "My first image created."
```

Dockerfile reference

Estructura del Dockerfile

FROM:	Indica la imagen:version
RUN:	Ejecuta comandos (bash) dentro de nuestra imagen
CMD:	Comando a ejecutar cuando corramos el contendor nuestra aplicación
EXPOSE:	Expone puertos de nuestro contenedor
COPY:	Copia los archivos desde el contexto de ejecución a la imagen

Docker Image_

```
# Imágenes
docker pull <image>
docker build -t <image> .
docker images
docker rmi <image>
```

Una imagen de Docker es una plantilla de solo lectura que define un contenedor.

Al montar una imagen con el comando `docker run` se creará un contenedor.

[Docker's images reference]

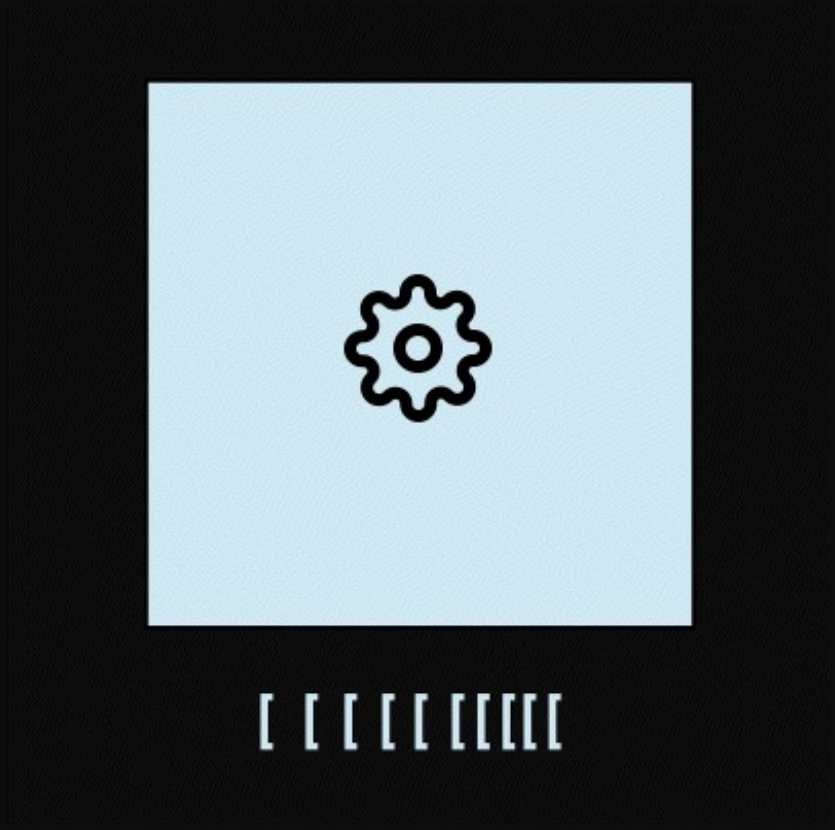
Docker Container_

```
# Contenedores
docker ps <-a>
docker run
docker rm
docker start
docker stop
docker restart
docker logs
docker exec
```

Un contenedor de Docker es una imagen de Docker instanciada (en ejecución).

Docker's containers reference

Recapitulemos_



Emerald Digital S.C.
Aplicaciones web, ciencia de datos y automatización

Ejercicio 1_

Hello world!

https://github.com/emeraldigital/curso-docker

Fundamentos de Docker

Emerald Digital S.C.
Aplicaciones web, ciencia de datos y automatización

Ejercicio 2_

Dockerfile

https://github.com/emeraldigital/curso-docker

Fundamentos de Docker

Emerald Digital S.C.
Aplicaciones web, ciencia de datos y automatización

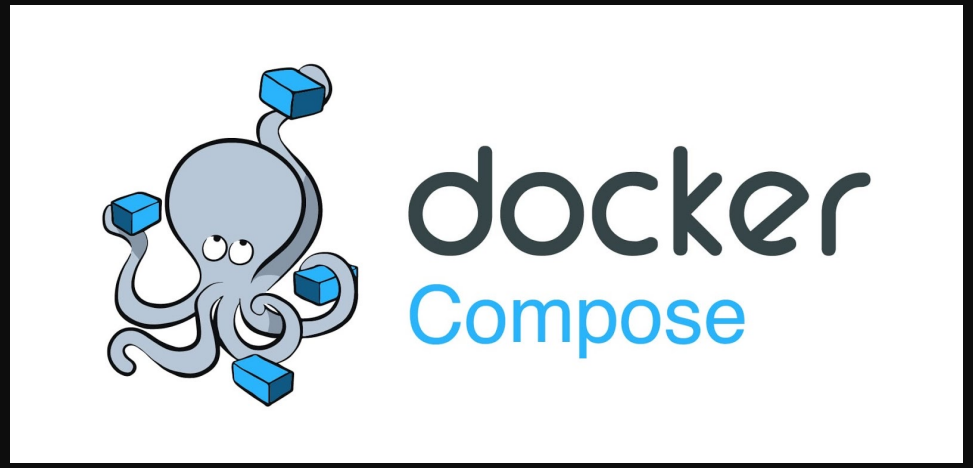
Ejercicio 2_

Docker pull

https://github.com/emeraldigital/curso-docker

Fundamentos de Docker

¿Docker Compose?_



Es una herramienta que nos permite usar archivos *.yaml (`docker-compose.yml`) para definir la ejecución de nuestras aplicaciones distribuidas en servicios.

[Docker's compose reference]

Emerald Digital S.C.
Aplicaciones web, ciencia de datos y automatización

Ejercicio 4_

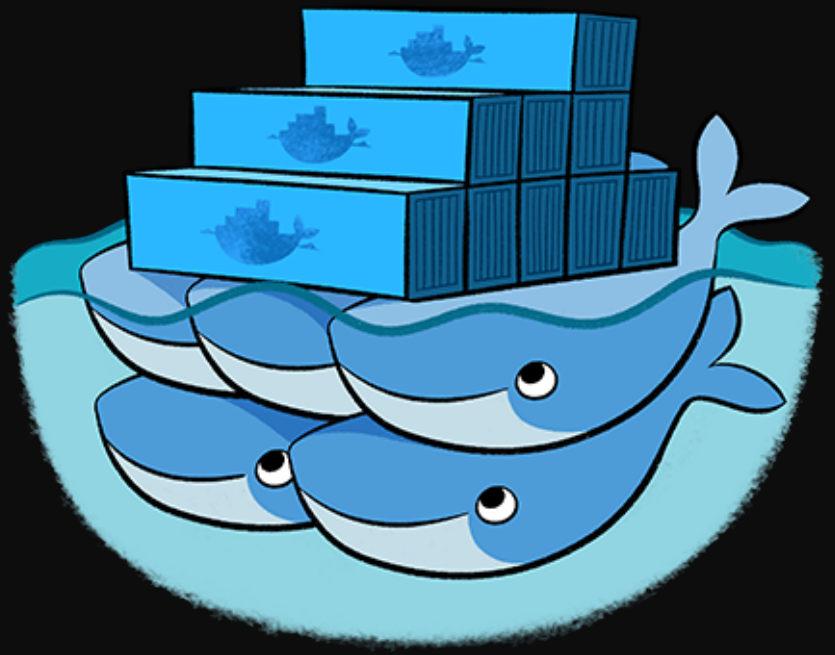
Docker compose

https://github.com/emeraldigital/curso-docker

Fundamentos de Docker

¿Docker swarm?_

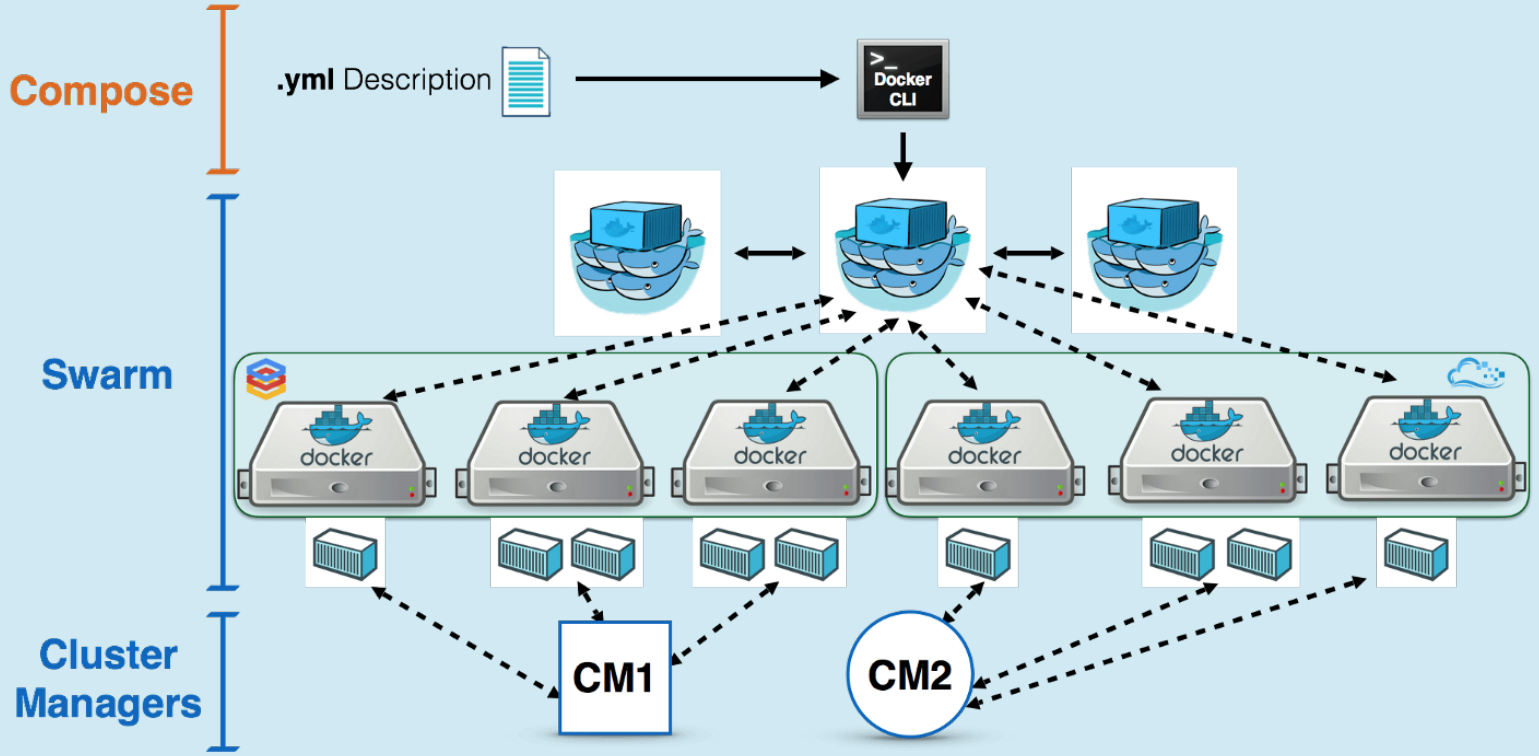
Es una herramienta para orquestar contenedores en clústers, permitiendo automatizar la gestión de nuestros servicios contenerizados.



Docker swarm reference

Clúster Swarm_

Un clúster de swarm consiste en Docker Engine implementado en múltiples nodos.

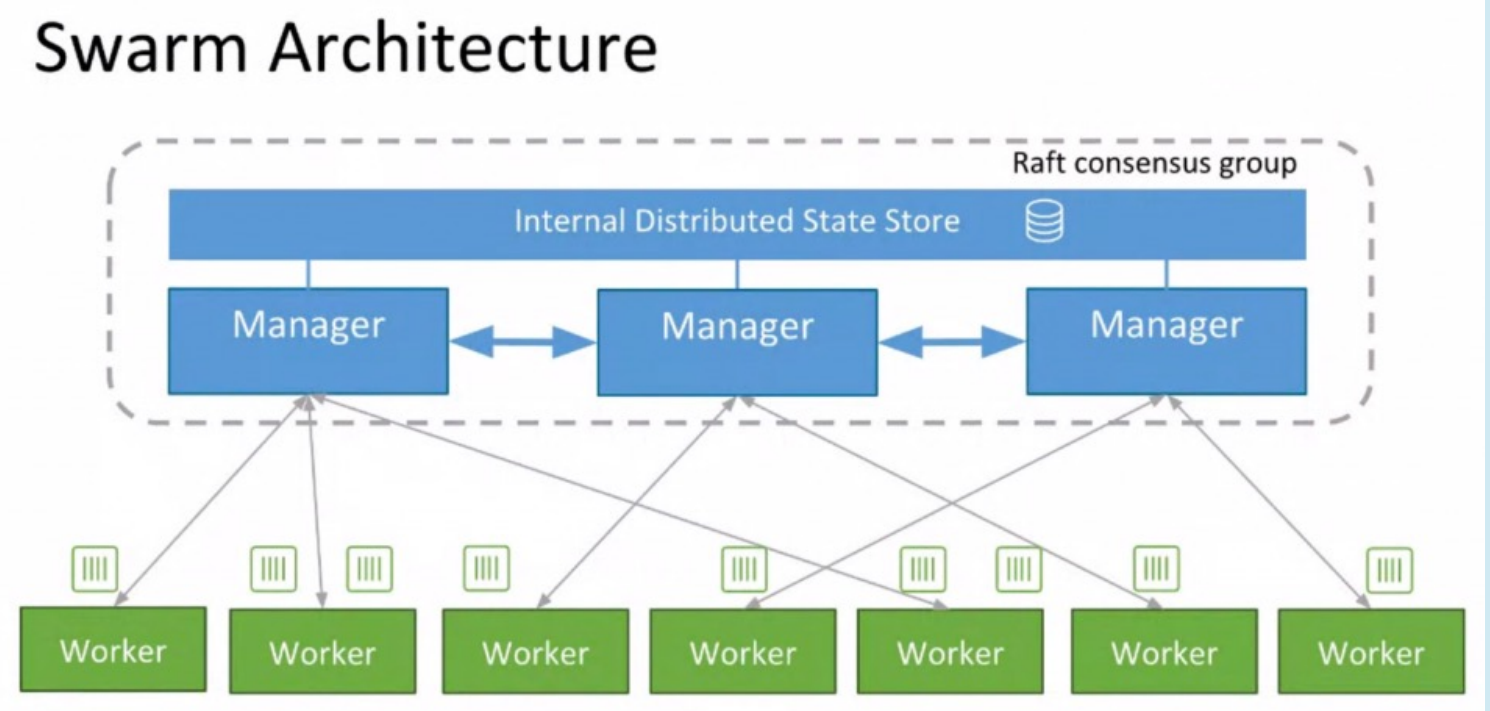


Clúster Swarm_

Managers → realizan la orquestación y la administración del clúster.

Workers → reciben y ejecutan tareas desde los nodos de administración.

[Docker swarm reference]



Emerald Digital S.C.
Aplicaciones web, ciencia de datos y automatización

Ejercicio 5_

Docker swarm

https://github.com/emeraldigital/curso-docker

Fundamentos de Docker

Consejos y buenas prácticas_

Emerald

Emerald Digital S.C.
Aplicaciones web, ciencia de datos y automatización

Mejores practicas_

- → Utilizar en lo posible imágenes oficiales
- → Generar contenedores efímeros
- → Entender el contexto de construcción (build)
- → Utilizar .dockerignore
- → Desacoplar aplicaciones (microservicios)
- → Usar multi-stage builds
- → Minimizar el número de capas (RUN, COPY, ADD)
  - → Instalar únicamente paquetes necesarios
  - → Optimizar comando RUN
  - → Utilizar argumentos multilínea ("\")
- → Evitar el uso del usuario root en comandos donde sea posible

Emerald

Emerald Digital S.C.
Aplicaciones web, ciencia de datos y automatización

¿Dudas?_

Por su atención, Gracias!_

Emerald

Emerald Digital S.C.
Aplicaciones web, ciencia de datos y automatización

CONTACTO
@jmruizab
jmanuel.rar@emeraldigital.com