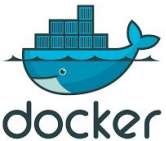
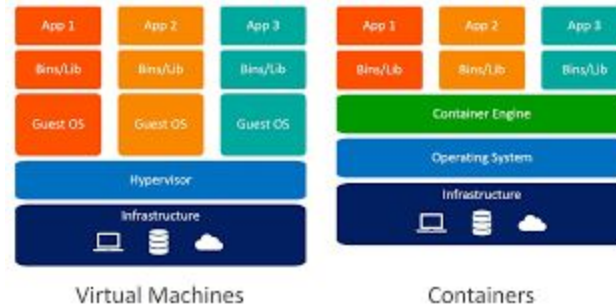




docker

Que es Docker?

Docker es una plataforma de código abierto diseñada para facilitar la creación, implementación y ejecución de aplicaciones en contenedores. Los contenedores son entornos ligeros y portátiles que encapsulan todo lo necesario para ejecutar una aplicación, incluidas las bibliotecas, las dependencias y el propio código, en un único paquete. Esto proporciona una consistencia y reproducibilidad en distintos entornos, desde el desarrollo hasta la producción.



Docker – Términos

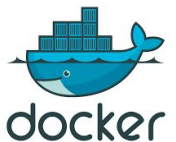
Imágen: Es un paquete ejecutable que incluye todo lo necesario para ejecutar una aplicación.

Contenedor: Es una instancia en ejecución de una imagen

Dockerfile: Archivo de configuración que define cómo se construye una imagen de Docker.

Docker Hub: Repositorio público de imágenes de Docker

Docker Compose: Es una herramienta que permite definir y gestionar aplicaciones multicontenedor.



Docker – comandos útiles I

Mostrar imágenes de Docker en el host

```
$ docker images
```

Descargar una imagen

```
$ docker pull
```

```
$ docker pull mongodb/mongodb-atlas-local:latest
```

Eliminar una imagen

```
$ docker rmi [IMAGE ID]
```



Docker – comandos útiles II

Muestra la lista de todos los contenedores en ejecución

```
$ docker ps  
$ docker ps -as
```

Eliminar un contenedor

```
$ docker rm [CONTAINER]
```

Ver los logs de un contenedor

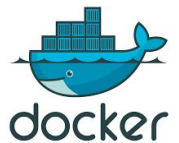
```
$ docker logs
```



Docker – comandos útiles III

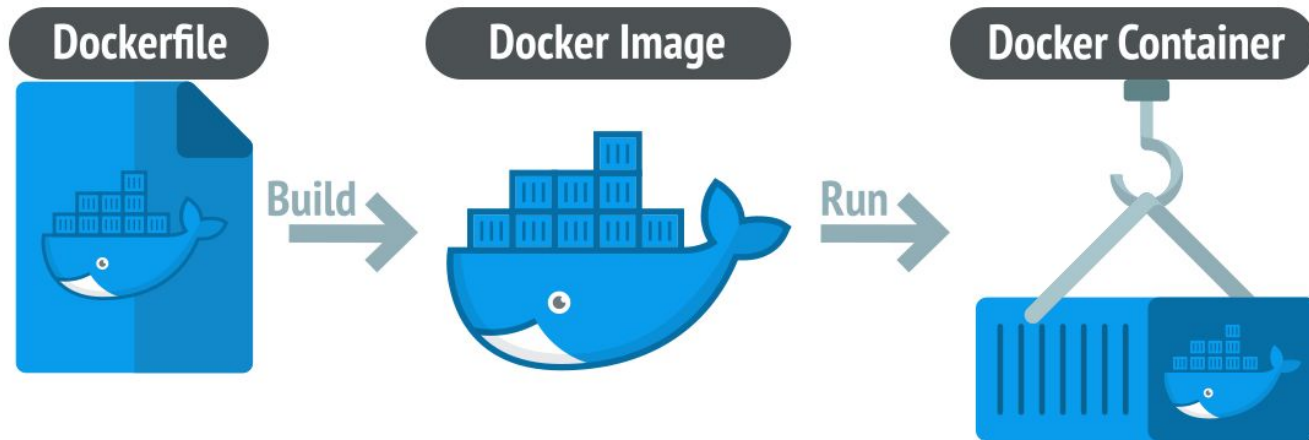
- # Docker container. Permite gestionar los contenedores
- # create: crear un contenedor
- # exec: ejecutar comandos dentro de un contenedor en ejecución
- # rm: eliminar un contenedor
- # run: crea un contenedor a partir de una imagen y lo ejecuta
- # ls: listar los contenedores en ejecución. Con -a lista todos

```
$docker run -p 27017:27017 -name nombre mongodb/mongodb-atlas-local  
$docker start nombre  
$docker stop nombre  
$docker exec -ti nombre sh
```



Dockerfile - I

Docker puede crear imágenes automáticamente leyendo las instrucciones de un Dockerfile. Un Dockerfile es un documento de texto que contiene todos los comandos que un usuario puede llamar en la línea de comando para ensamblar una imagen. Esta página describe los comandos que puede usar en un Dockerfile.



Dockerfile - II

```
FROM python:3.12
```

```
# Creamos un directorio para la  
aplicacion
```

```
WORKDIR /app
```

```
# Copiamos la aplicacion..
```

```
COPY . .
```

```
# Instalamos las dependencias
```

```
RUN pip install -r requirements.txt
```

```
# Exponemos un puerto
```

```
EXPOSE 2020
```

```
# Iniciamos la aplicacion
```

```
CMD ["python", "<python_app>"]
```

Creamos la imagen a partir de un dockerfile

```
docker build -t image_name .
```

Creamos el contenedor y lo ejecutamos

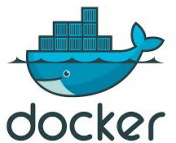
```
docker run -p 2020:2020 --name <cont_name>  
image_name
```



docker-compose - I

Docker Compose es una herramienta para definir y ejecutar aplicaciones de múltiples contenedores. Es la clave para desbloquear una experiencia de desarrollo e implementación optimizada y eficiente.

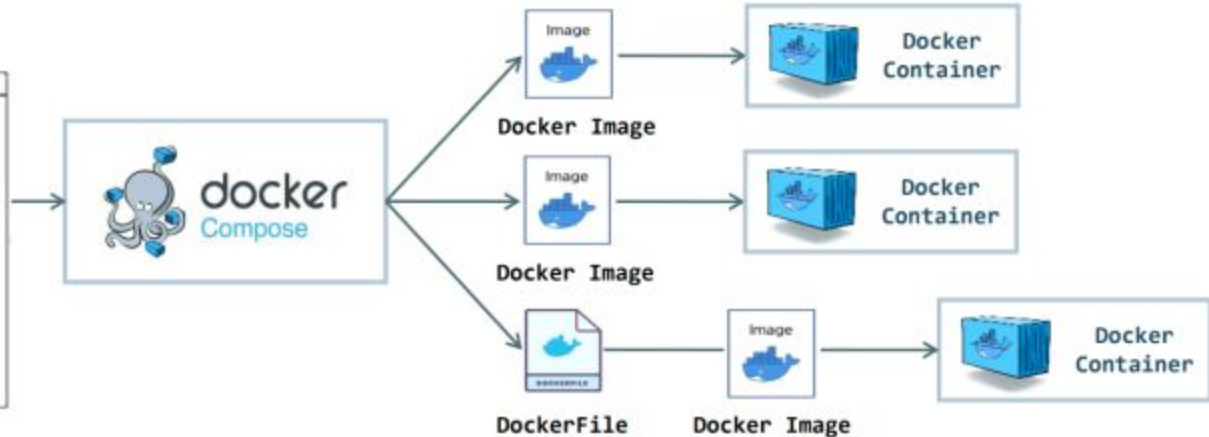
Compose simplifica el control de toda su pila de aplicaciones, lo que facilita la administración de servicios, redes y volúmenes en un archivo de configuración YAML único y comprensible. Luego, con un solo comando, crea e inicia todos los servicios desde su archivo de configuración.



docker-compose - II

docker-compose.yml

```
*** docker-compose.yml ***
version: "3.7"
services:
  db:
    image: mysql:5.0.19
    restart: always
    environment:
      - MYSQL_DATABASE=example
      - MYSQL_ROOT_PASSWORD=password
  app:
    build: app
    restart: always
  web:
    build: web
    restart: always
    ports:
      - 80:80
```



iniciar: docker-compose up -d
detener: docker-compose down

