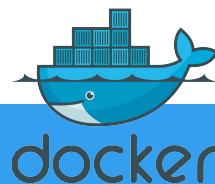


docker

Contenido

- Conceptos Generales
- Imágenes
- Contenedores
- Volúmenes
- Redes
- Docker Compose

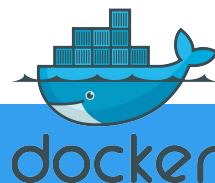


Conceptos Generales

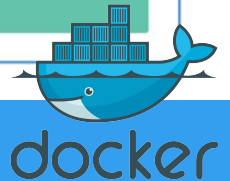
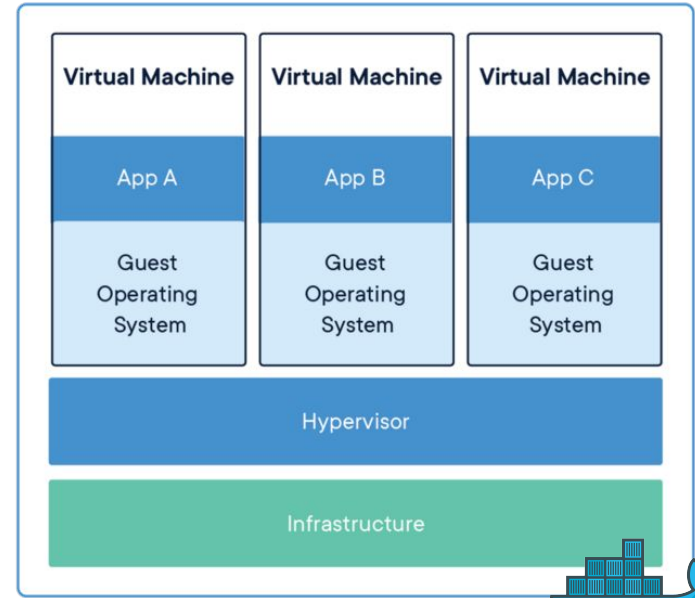
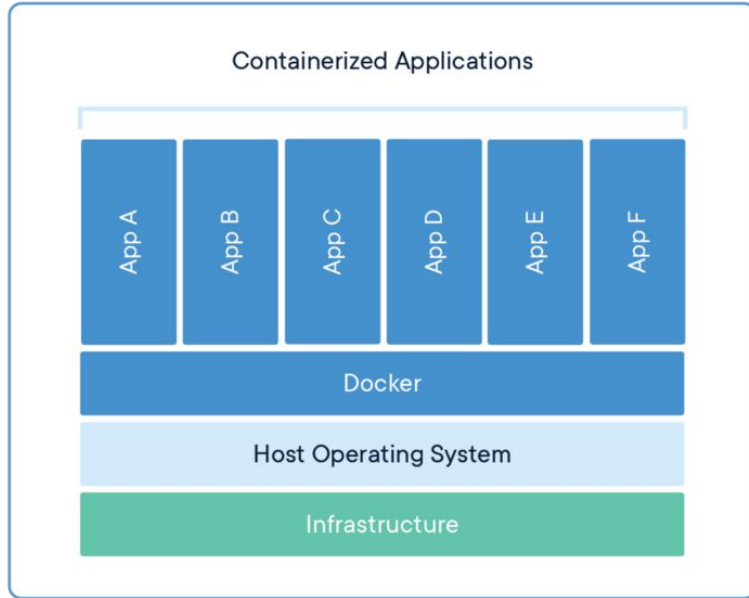
Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

Web: <https://www.docker.com/>

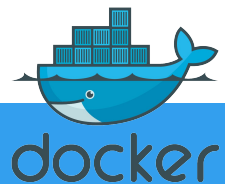
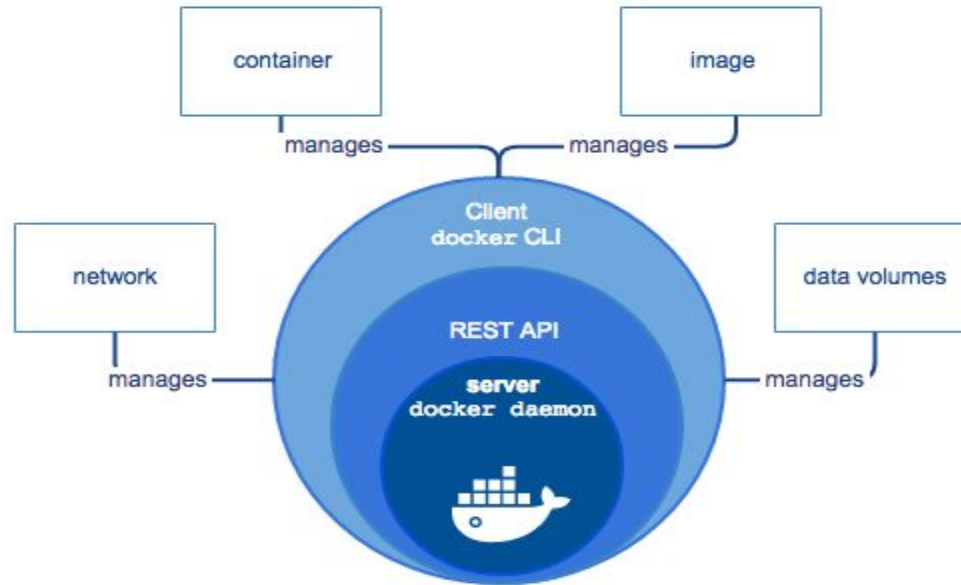
Doc: <https://docs.docker.com/>



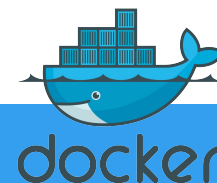
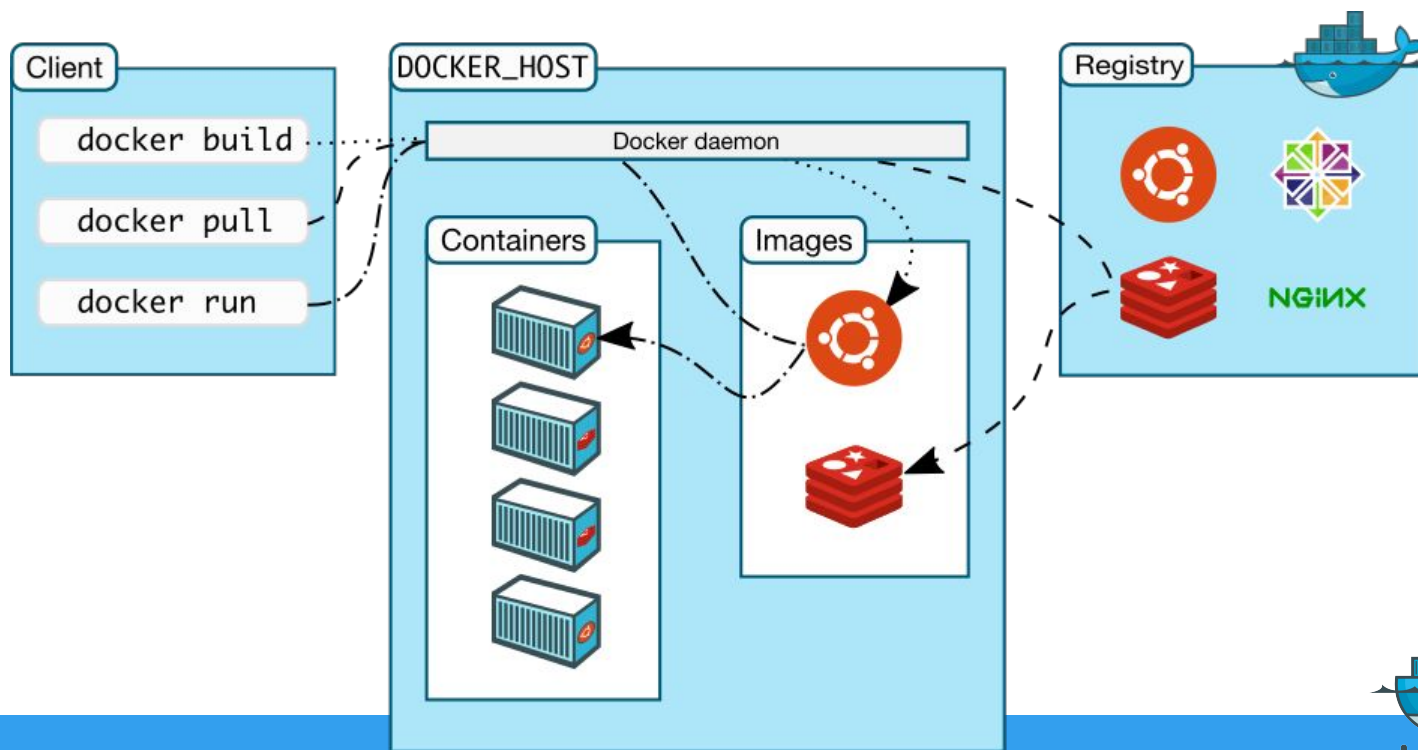
Conceptos Generales - VM Vs Docker



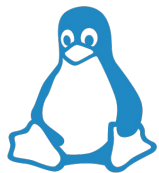
Conceptos Generales - Componentes



Conceptos Generales - Componentes



Conceptos Generales - Instalar Docker



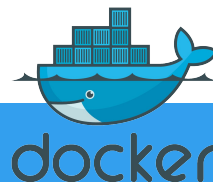
Docker for Linux



Docker Desktop for Mac

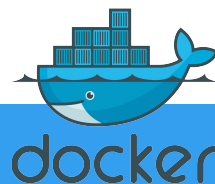


Docker Desktop for Windows



Imágenes

Una imagen es una representación estática de la aplicación o el servicio y de su configuración y las dependencias. Para ejecutar la aplicación o el servicio, se crea una instancia de la imagen de la aplicación para crear un contenedor, que se ejecutará en el host de Docker.



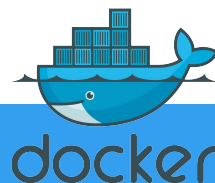
Imágenes - Dockerfile



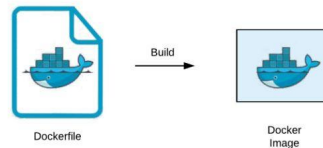
Es un archivo de texto plano que contiene una serie de instrucciones necesarias para crear una imagen que, posteriormente, se convertirá en una sola aplicación utilizada para un determinado propósito.

Instrucciones populares: FROM, RUN, CMD, LABEL, EXPOSE, ENV, ADD, COPY, ENTRYPOINT, VOLUME

```
FROM ubuntu
RUN mkdir /myvol
RUN echo "hello world" > /myvol/greeting
VOLUME /myvol
```



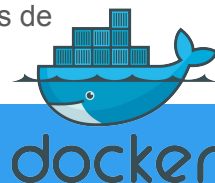
Imágenes - docker build



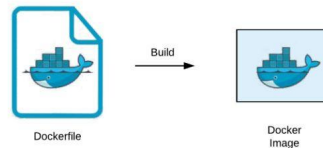
Uso: `docker build [OPTIONS] PATH | URL | -`

El comando `docker build` construye imágenes Docker a partir de un Dockerfile y un "contexto". El contexto de una compilación es el conjunto de archivos ubicados en el `PATH` o `URL` especificada. El proceso de compilación puede hacer referencia a cualquiera de los archivos en el contexto. Por ejemplo, su compilación puede usar una instrucción `COPY` para hacer referencia a un archivo en el contexto.

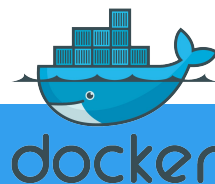
El parámetro `URL` puede referirse a tres tipos de recursos: repositorios Git, contextos tarball preenvasados y archivos de texto sin formato.



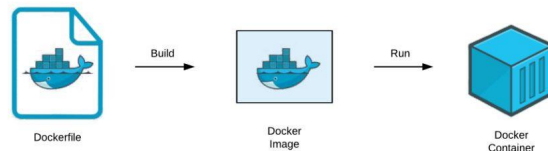
Imágenes - comandos útiles



Command	Descripción
<code>docker image build</code>	Build an image from a Dockerfile
<code>docker image history</code>	Show the history of an image
<code>docker image import</code>	Import the contents from a tarball to create a filesystem image
<code>docker image inspect</code>	Display detailed information on one or more images
<code>docker image load</code>	Load an image from a tar archive or STDIN
<code>docker image ls</code>	List images
<code>docker image prune</code>	Remove unused images
<code>docker image pull</code>	Pull an image or a repository from a registry
<code>docker image push</code>	Push an image or a repository to a registry
<code>docker image rm</code>	Remove one or more images
<code>docker image save</code>	Save one or more images to a tar archive (streamed to STDOUT by default)
<code>docker image tag</code>	Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE



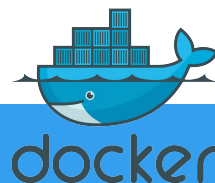
Contenedores



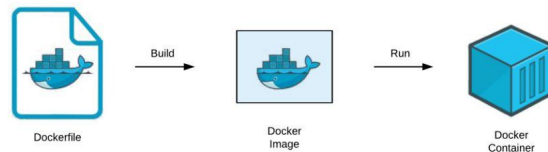
Un contenedor es una instancia de tiempo de ejecución de una imagen acoplable.
Un contenedor Docker consta de:

- Una imagen de Docker
- Un entorno de ejecución.
- Un conjunto estándar de instrucciones

El concepto se toma prestado de los contenedores de envío, que definen un estándar para enviar mercancías a nivel mundial. Docker define un estándar para enviar software.



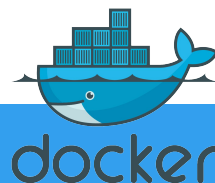
Contenedores - docker run



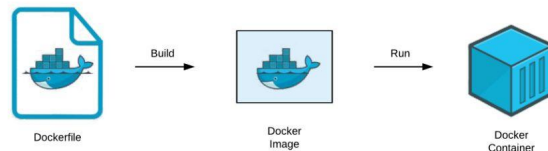
Uso: `docker run [OPTIONS] IMAGE[:TAG|@DIGEST] [COMMAND] [ARG...]`

El comando `docker run` debe especificar una IMAGEN para generar el contenedor.

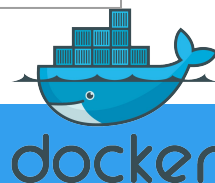
Ejemplo: `docker run -d -p 80:80 my_image service nginx start`



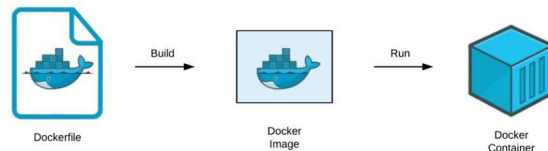
Contenedores - comandos útiles



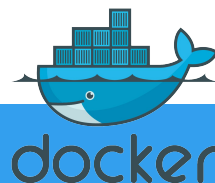
Command	Description
<code>docker container commit</code>	Create a new image from a container's changes
<code>docker container cp</code>	Copy files/folders between a container and the local filesystem
<code>docker container diff</code>	Inspect changes to files or directories on a container's filesystem
<code>docker container exec</code>	Run a command in a running container
<code>docker container export</code>	Export a container's filesystem as a tar archive
<code>docker container inspect</code>	Display detailed information on one or more containers
<code>docker container kill</code>	Kill one or more running containers
<code>docker container logs</code>	Fetch the logs of a container
<code>docker container ls</code>	List containers
<code>docker container pause</code>	Pause all processes within one or more containers
<code>docker container port</code>	List port mappings or a specific mapping for the container
<code>docker container prune</code>	Remove all stopped containers



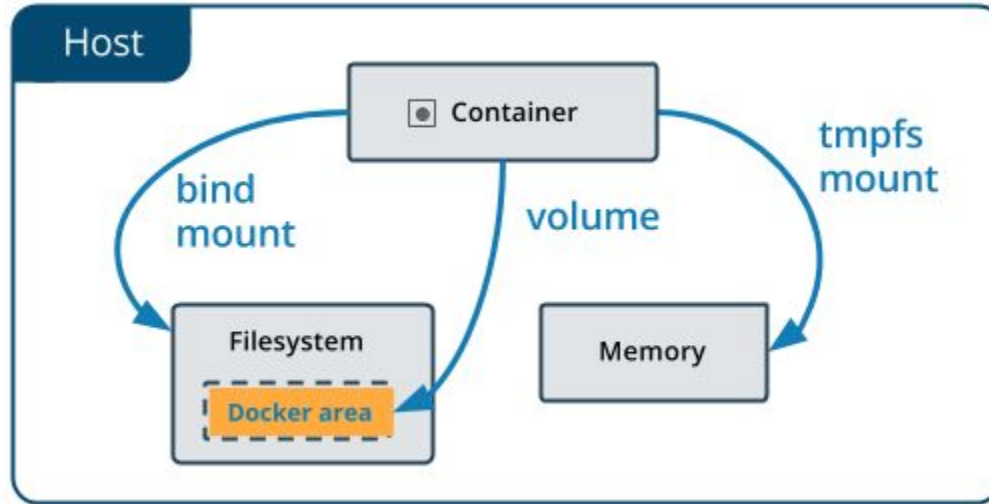
Contenedores - comandos útiles



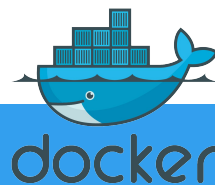
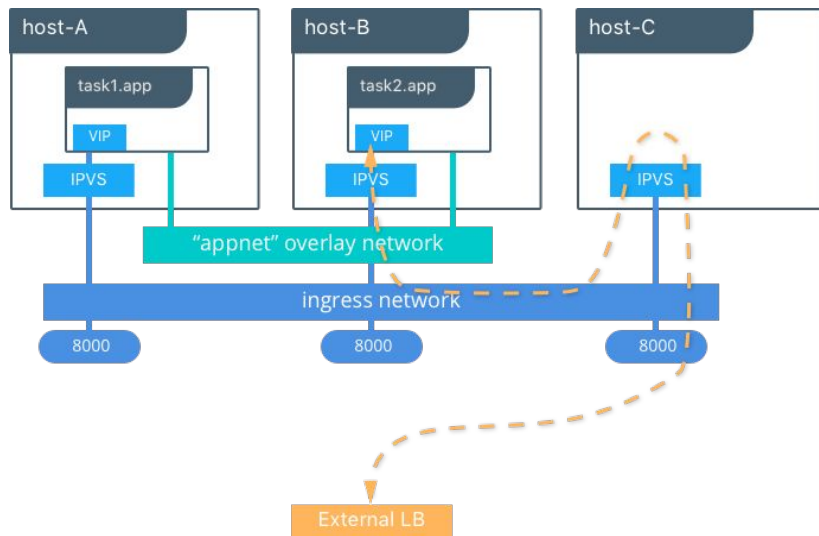
Command	Description
<code>docker container rename</code>	Rename a container
<code>docker container restart</code>	Restart one or more containers
<code>docker container rm</code>	Remove one or more containers
<code>docker container run</code>	Run a command in a new container
<code>docker container start</code>	Start one or more stopped containers
<code>docker container stats</code>	Display a live stream of container(s) resource usage statistics
<code>docker container stop</code>	Stop one or more running containers
<code>docker container top</code>	Display the running processes of a container
<code>docker container unpause</code>	Unpause all processes within one or more containers
<code>docker container update</code>	Update configuration of one or more containers



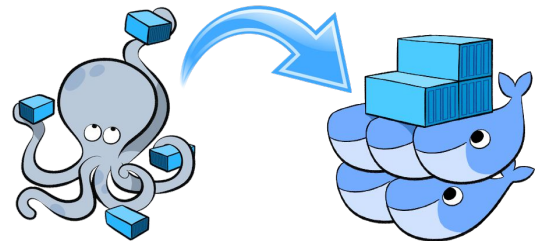
Volúmenes



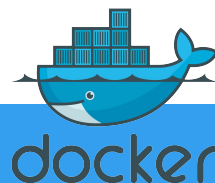
Redes



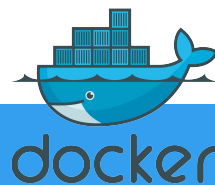
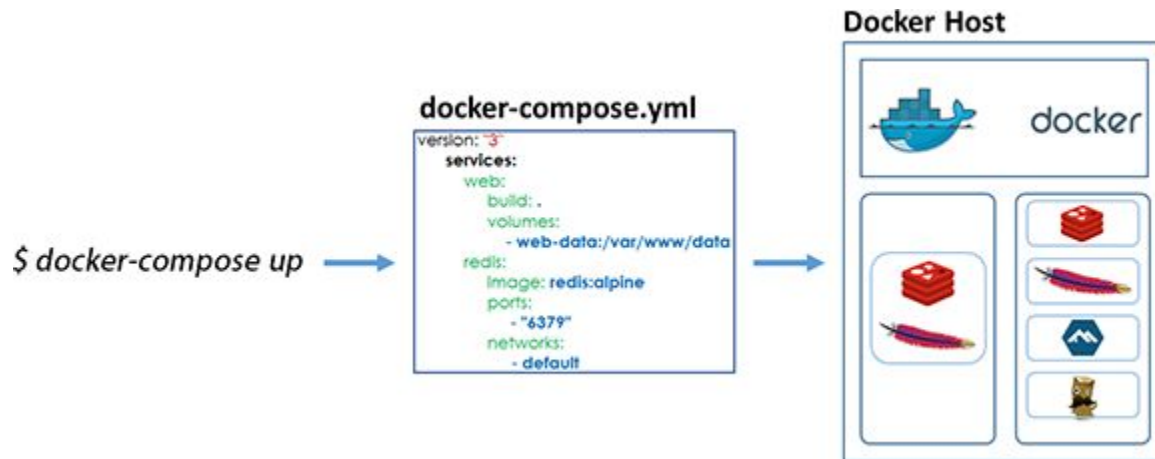
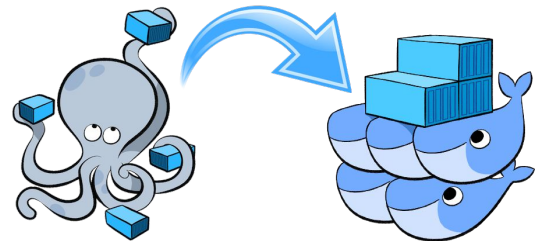
- Docker Compose



Compose es una herramienta para definir y ejecutar aplicaciones Docker de contenedores múltiples. Compose utiliza un archivo YAML para configurar los servicios de su aplicación. Luego con un solo comando crea e inicia todos los servicios desde su configuración.



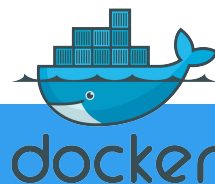
- Docker Compose



Listo mucha teoria...

Ahora vamos a jugar nosotros....

PD: esto es el santo grial.... -> <https://hub.docker.com/>





docker