ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

ECYS | DTT
DESARROLLO DE TRANSFERENCIA TECNOLÓGICA

Día, Fecha:	Jueves, 24/10/2024
Hora de inicio:	10:40 - 12:20

Introducción a la Programación y Computación 2 [P]

Denilson Florentín de León Aguilar

Recordatorio Grabar Clase

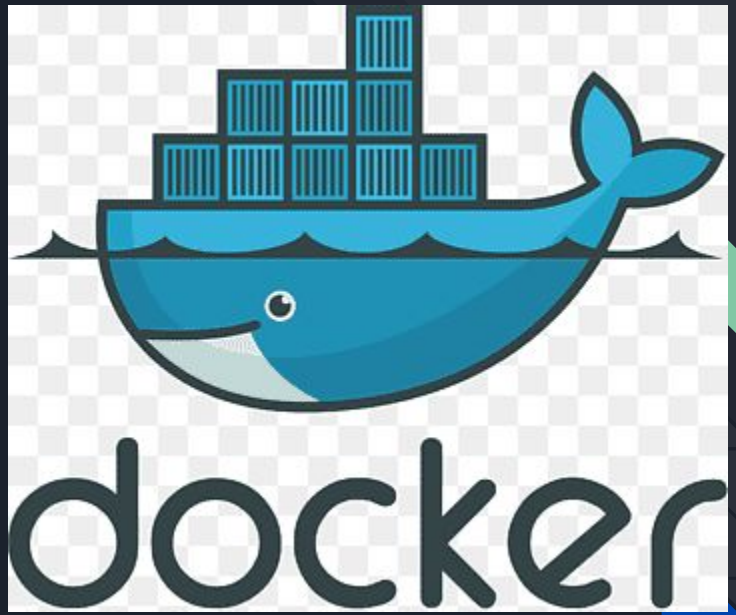
Contenido clase 13

- Notas finales
- Docker

Notas Finales

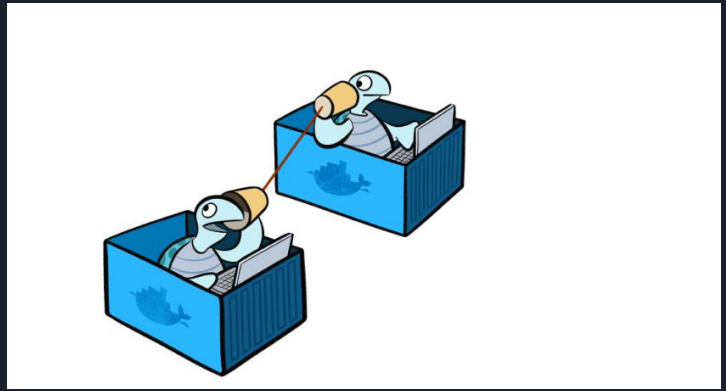
Felicidades

DOCKER



¿Qué es Docker?

Docker es una plataforma de código abierto diseñada para simplificar la creación, implementación y ejecución de aplicaciones mediante contenedores. Los contenedores permiten empaquetar una aplicación y todas sus dependencias en una unidad estándar para el desarrollo y ejecución, garantizando que la aplicación se ejecute de la misma manera en cualquier entorno.

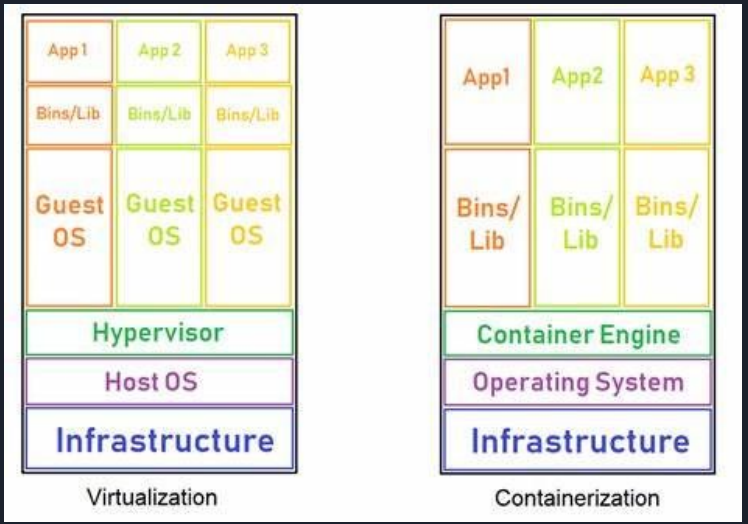


Denilson de León

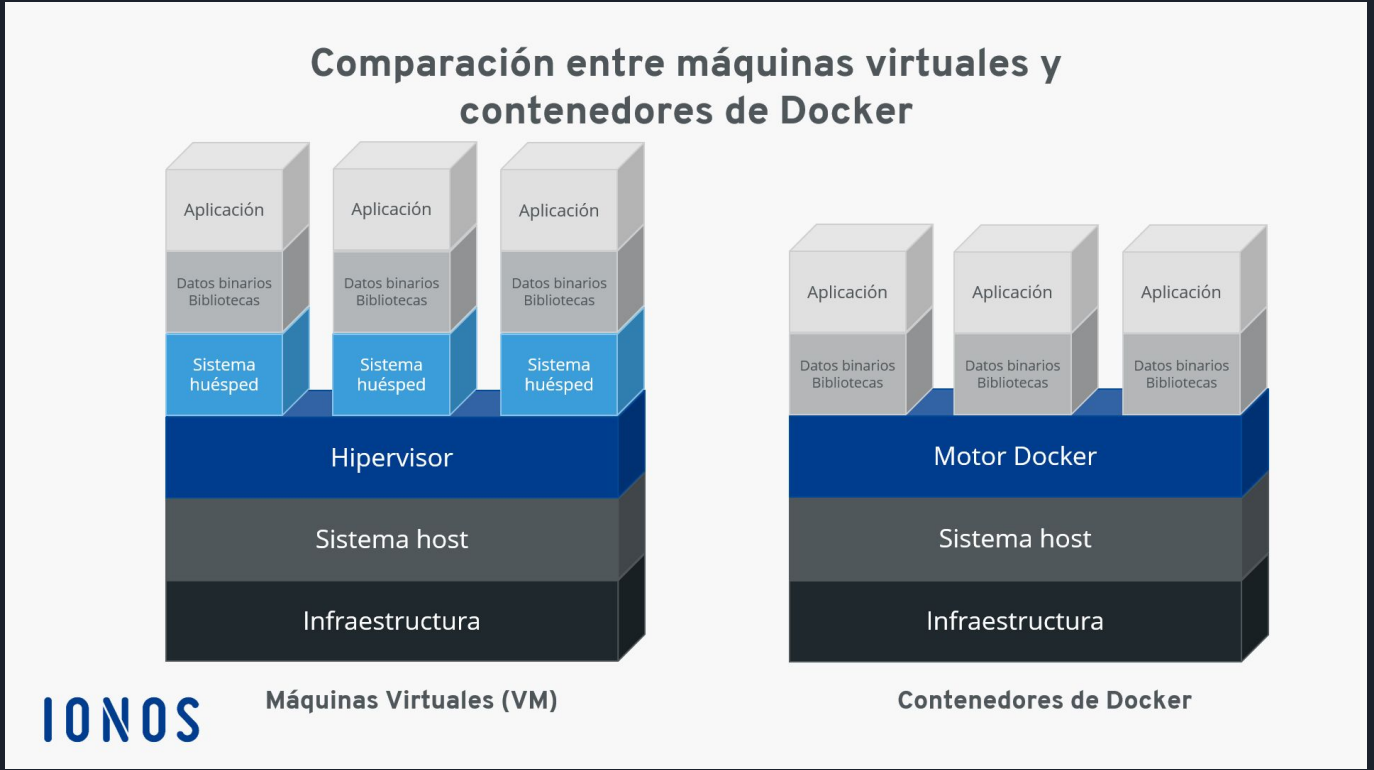
Diferencia entre Docker y Virtualización (VM)

Docker: Utiliza la virtualización a nivel de sistema operativo para ejecutar contenedores. Los contenedores comparten el mismo kernel del sistema operativo anfitrión, lo que los hace más livianos y eficientes en términos de recursos.

Virtualización (VM): Utiliza la virtualización a nivel de hardware para ejecutar máquinas virtuales. Cada máquina virtual incluye su propio sistema operativo completo, lo que puede resultar en un uso más intensivo de recursos y una mayor complejidad en la gestión.



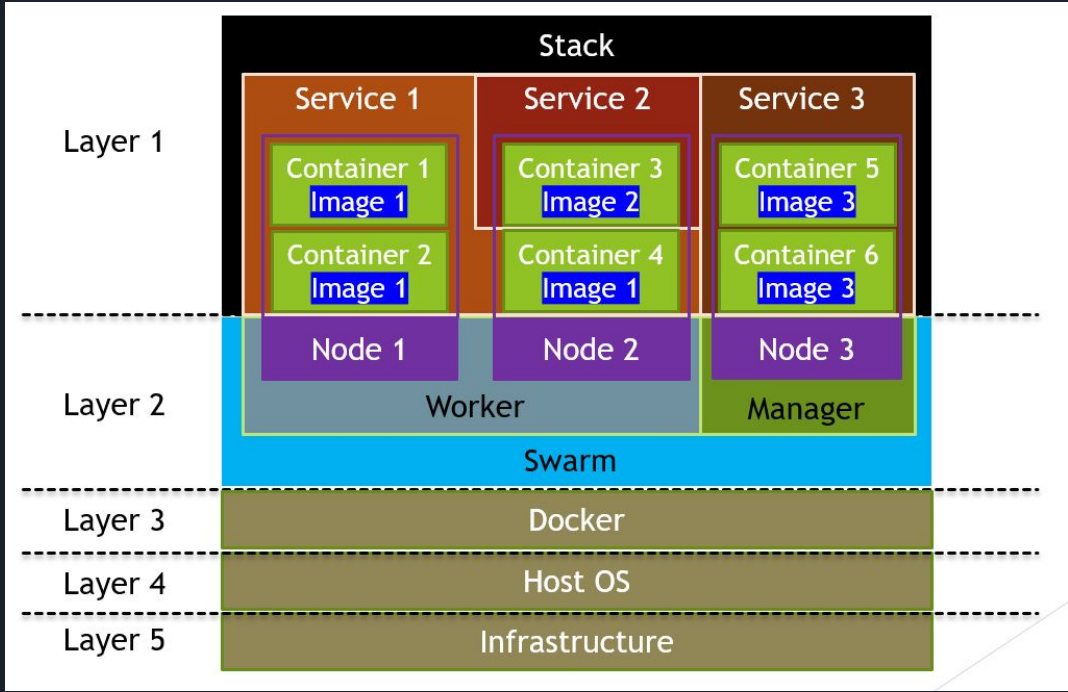
Diferencia entre Docker y Virtualización (VM)



¿Para qué sirve Docker?

Desarrollo de Aplicaciones: Docker simplifica el proceso de desarrollo de aplicaciones al permitir a los desarrolladores crear, probar y distribuir aplicaciones de manera consistente y reproducible en cualquier entorno.

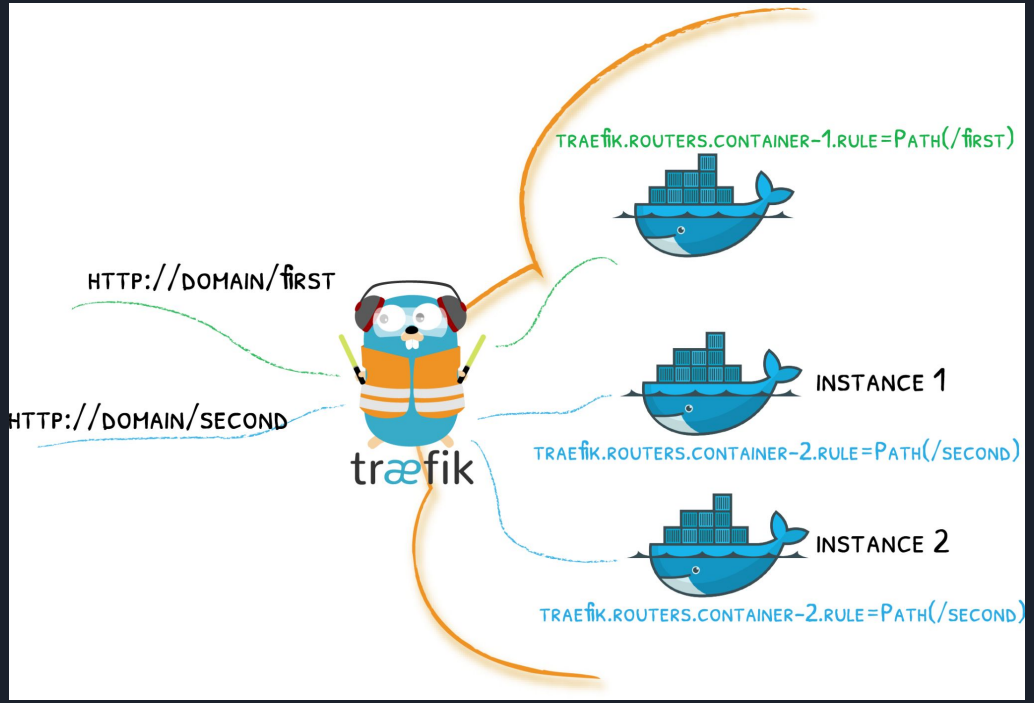
Implementación de Aplicaciones: Docker facilita la implementación de aplicaciones al proporcionar un entorno aislado y portátil para ejecutar aplicaciones en cualquier infraestructura, ya sea en la nube, en servidores locales o en máquinas virtuales.

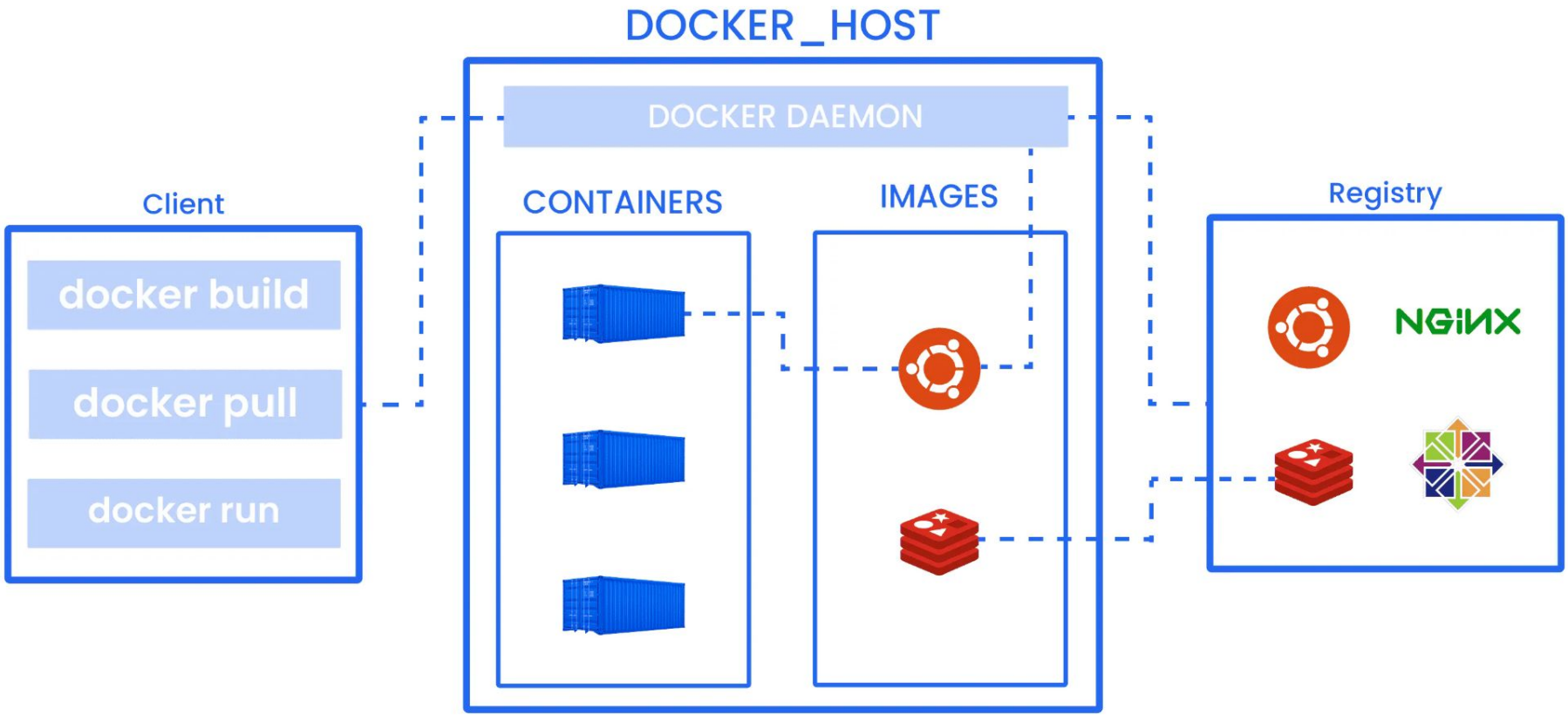


¿Para qué sirve Docker?

Escalabilidad: Docker facilita la escalabilidad horizontal de aplicaciones al permitir la creación rápida de múltiples instancias de contenedores para satisfacer la demanda variable de recursos.

Gestión de Infraestructura: Docker simplifica la gestión de infraestructura al permitir la automatización de la creación, configuración y despliegue de aplicaciones mediante herramientas de orquestación como Docker Compose y Kubernetes.




DOCKER_HOST
DOCKER DAEMON
Client
docker build
docker pull
docker run
CONTAINERS
IMAGES
Registry
NGINX

¿Cómo usar Docker?

Escalabilidad: Docker facilita la escalabilidad horizontal de aplicaciones al permitir la creación rápida de múltiples instancias de contenedores para satisfacer la demanda variable de recursos.

Gestión de Infraestructura: Docker simplifica la gestión de infraestructura al permitir la automatización de la creación, configuración y despliegue de aplicaciones mediante herramientas de orquestación como Docker Compose y Kubernetes.

1. Instalación de Docker

Antes de comenzar, asegúrate de tener Docker instalado en tu sistema. Puedes descargar e instalar Docker desde el sitio web oficial de Docker según el sistema operativo que estés utilizando.

Los pasos para ubuntu son:

- Agregar repositorios.
- Instalar con apt-get install
- Comprobar

```
# Add Docker's official GPG key:

sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc



# Add the repository to Apt sources:

echo \

 "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \

 $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \

 sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update
```

1. Instalación de Docker

Instalamos una vez agregados los repository

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

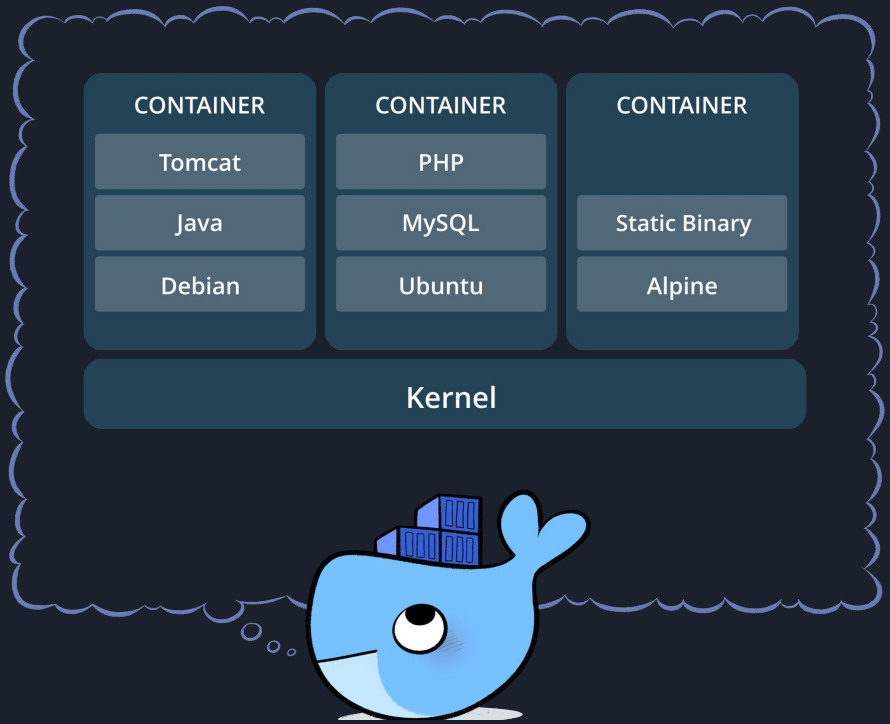
Comprobamos estado

```
sudo docker run hello-world
```

2. Creación de un Dockerfile

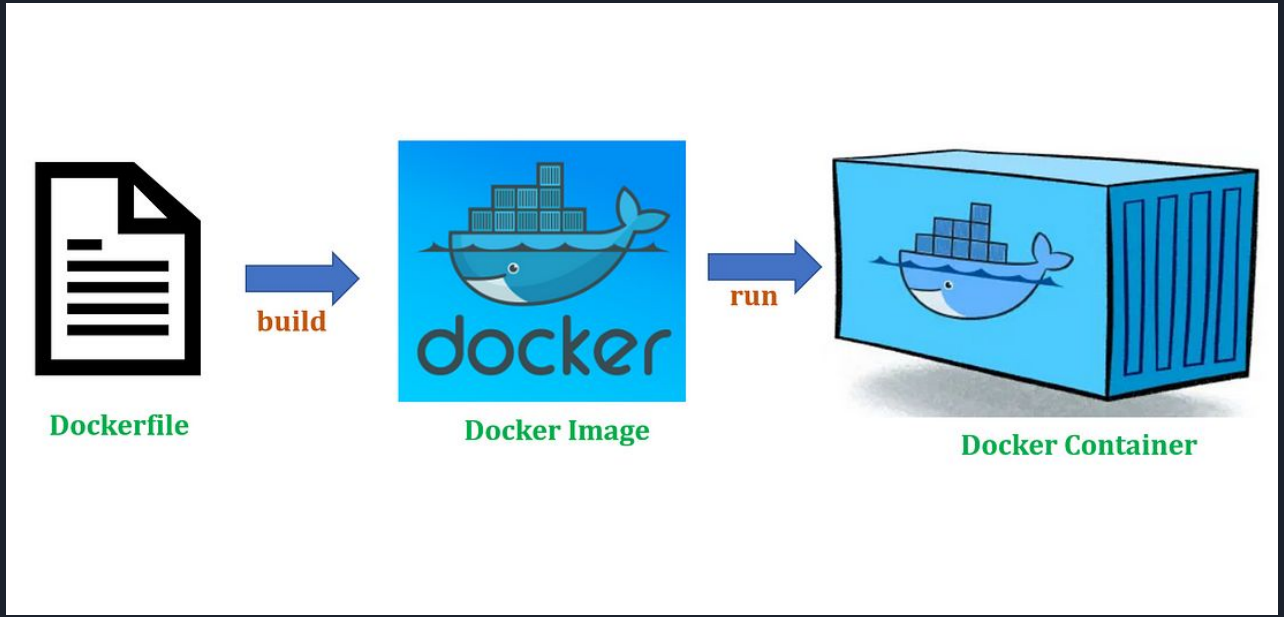
Un Dockerfile es un archivo de texto que contiene instrucciones para construir una imagen Docker. Para crear un Dockerfile, sigue estos pasos:

- Crea un nuevo archivo llamado Dockerfile en el directorio raíz de tu proyecto.
- Dentro del archivo, especifica la imagen base que deseas utilizar, las dependencias necesarias y los comandos de ejecución de tu aplicación.



2. Creación de un Dockerfile

Una vez teniendo el Dockerfile en el directorio a empaquetar, se ejecuta Build y se crea la imagen

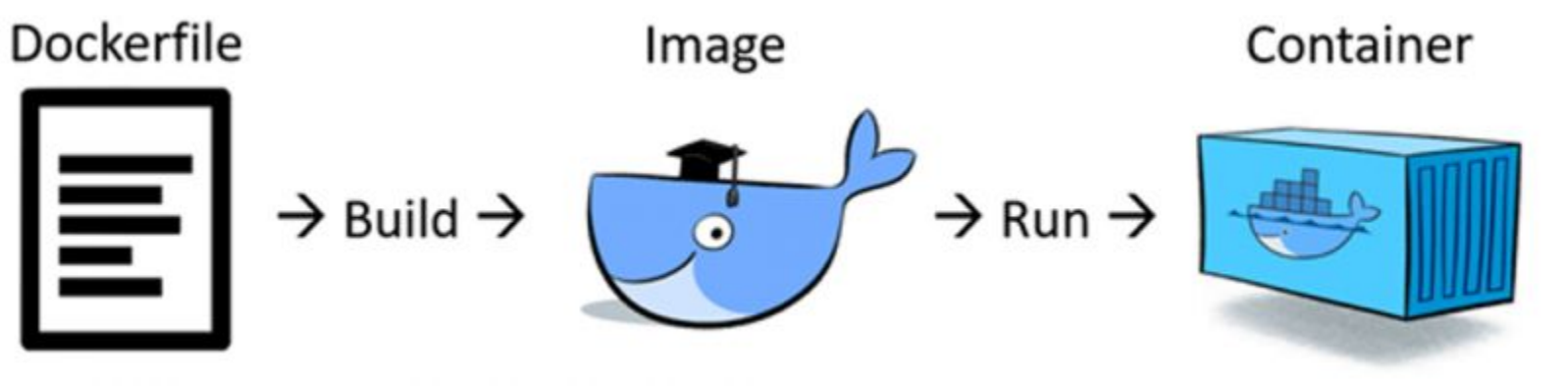


3. Construcción de la imagen Docker

Una vez que hayas creado el Dockerfile, puedes construir la imagen Docker ejecutando el siguiente comando en tu terminal:

```
docker build -t nombre_de_la_imagen .
```

Este comando construirá una nueva imagen Docker utilizando las instrucciones definidas en el Dockerfile y la etiquetará con el nombre especificado.

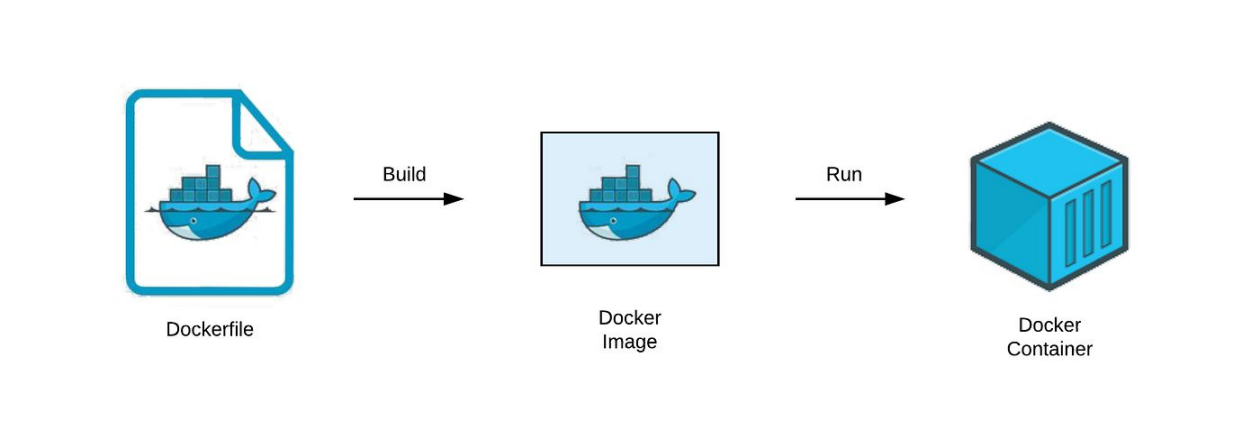


4. Ejecución de un contenedor Docker

Una vez que hayas construido la imagen Docker, puedes ejecutar un contenedor basado en esa imagen utilizando el siguiente comando:

```
docker run nombre_de_la_imagen
```

Este comando iniciará un nuevo contenedor basado en la imagen especificada y ejecutará la aplicación dentro del contenedor.



5. Gestión de contenedores Docker

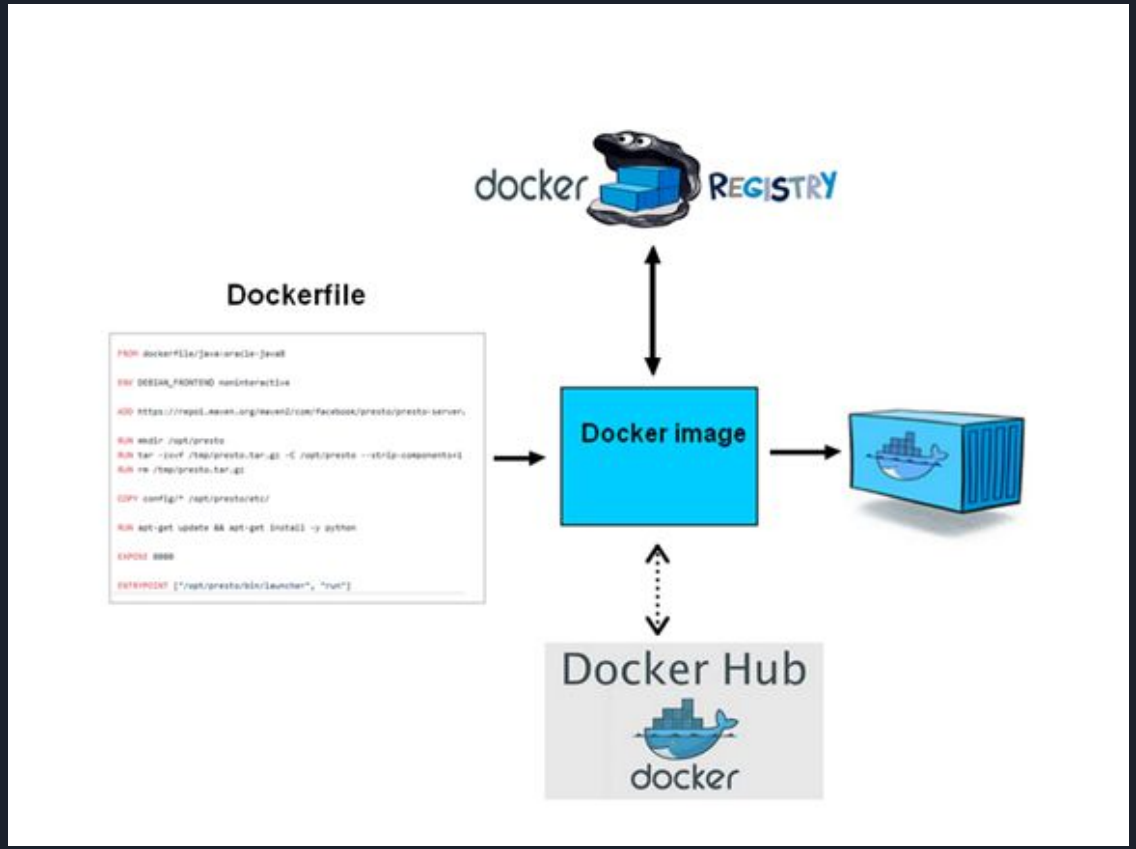
Puedes administrar contenedores Docker utilizando una variedad de comandos. Algunos de los comandos más comunes incluyen:

- **docker ps:** Muestra una lista de contenedores en ejecución.
- **docker stop <ID_del_contenedor>:** Detiene un contenedor en ejecución.
- **docker rm <ID_del_contenedor>:** Elimina un contenedor.
- **docker logs <ID_del_contenedor>:** Muestra los registros de salida de un contenedor.

```
docker ps
docker stop <ID_del_contenedor>
docker rm <ID_del_contenedor>
docker logs <ID_del_contenedor>
```

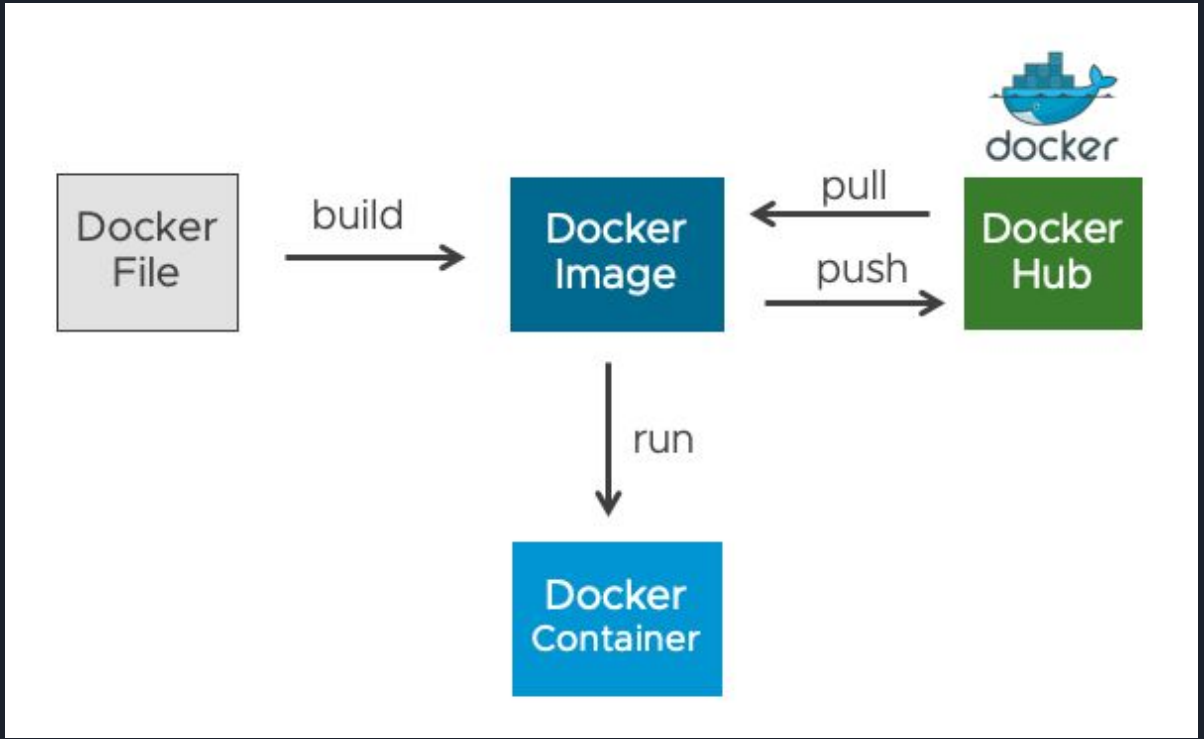
6. Distribución de imágenes Docker

Una vez que hayas construido una imagen Docker, puedes distribuirla a otros usuarios compartiéndola en un registro de Docker público o privado, como Docker Hub.



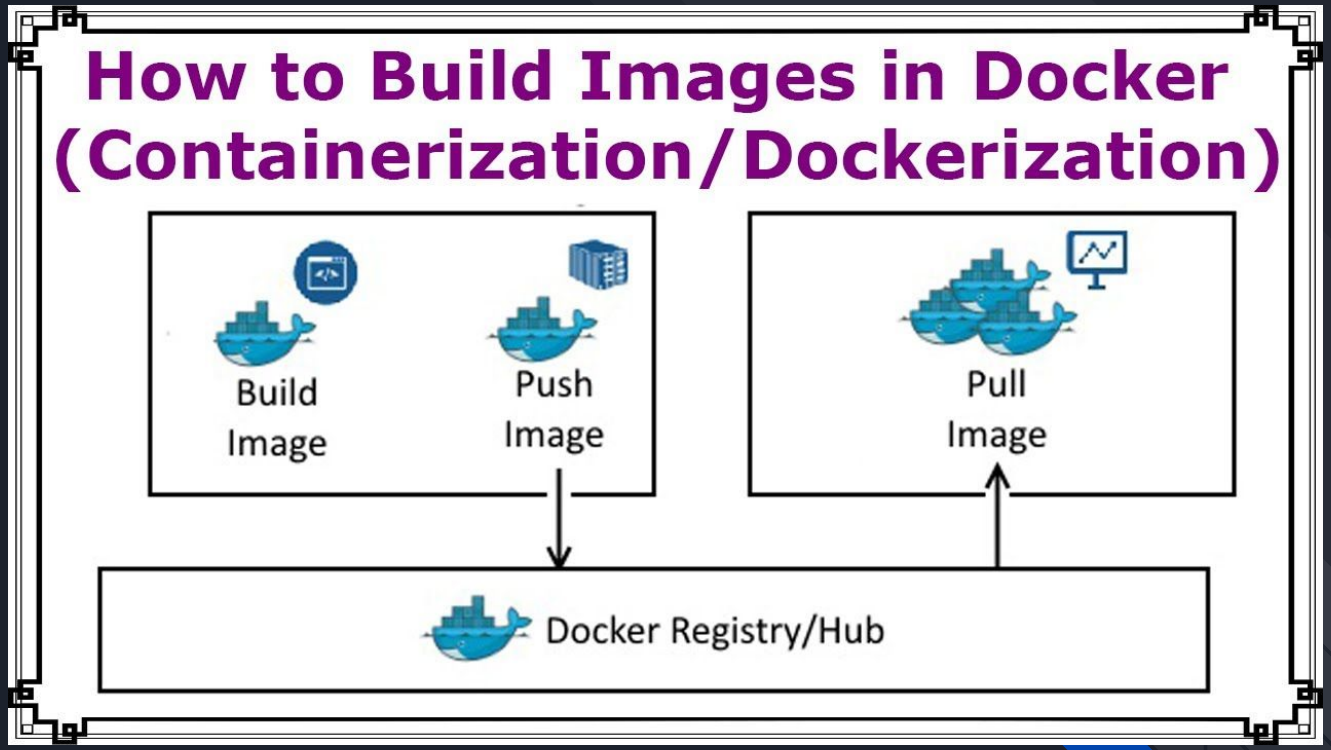
7. Monitoreo y depuración

Docker proporciona herramientas para monitorear y depurar contenedores en ejecución. Puedes usar comandos como docker stats para monitorear el uso de recursos de tus contenedores y docker exec para ejecutar comandos dentro de un contenedor en ejecución.

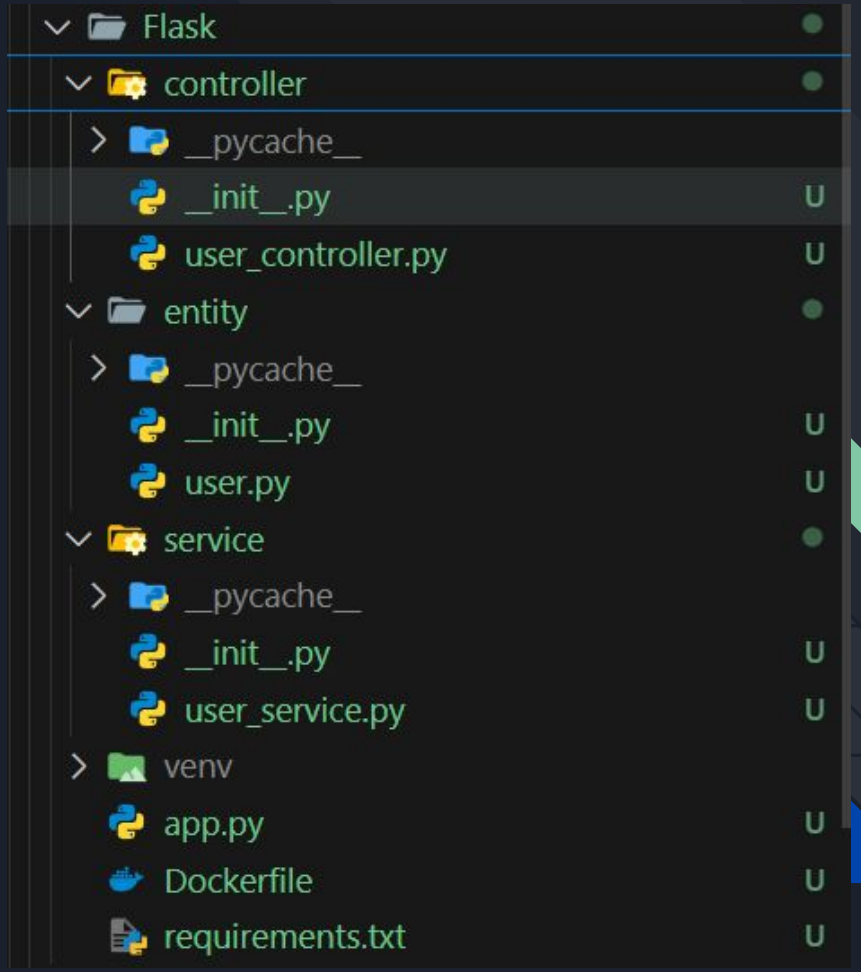


Recordatorio Captura de pantalla

Ejemplo


How to Build Images in Docker
(Containerization/Dockerization)
Build
Image
Push
Image
Pull
Image
Docker Registry/Hub

Ejemplo


Flask
controller
__pycache__
__init__.py U
user_controller.py U
entity
__pycache__
__init__.py U
user.py U
service
__pycache__
__init__.py U
user_service.py U
venv
app.py U
Dockerfile U
requirements.txt U

El ejemplo es un servidor de Flask. Se trata de un registro de usuarios, se recomendó una buena organización de módulos y carpetas para un desarrollo ordenado. Además, se adjuntó el archivo Dockerfile.