

ENTORNOS DE DESARROLLO

UD3. Docker como entorno de
desarrollo

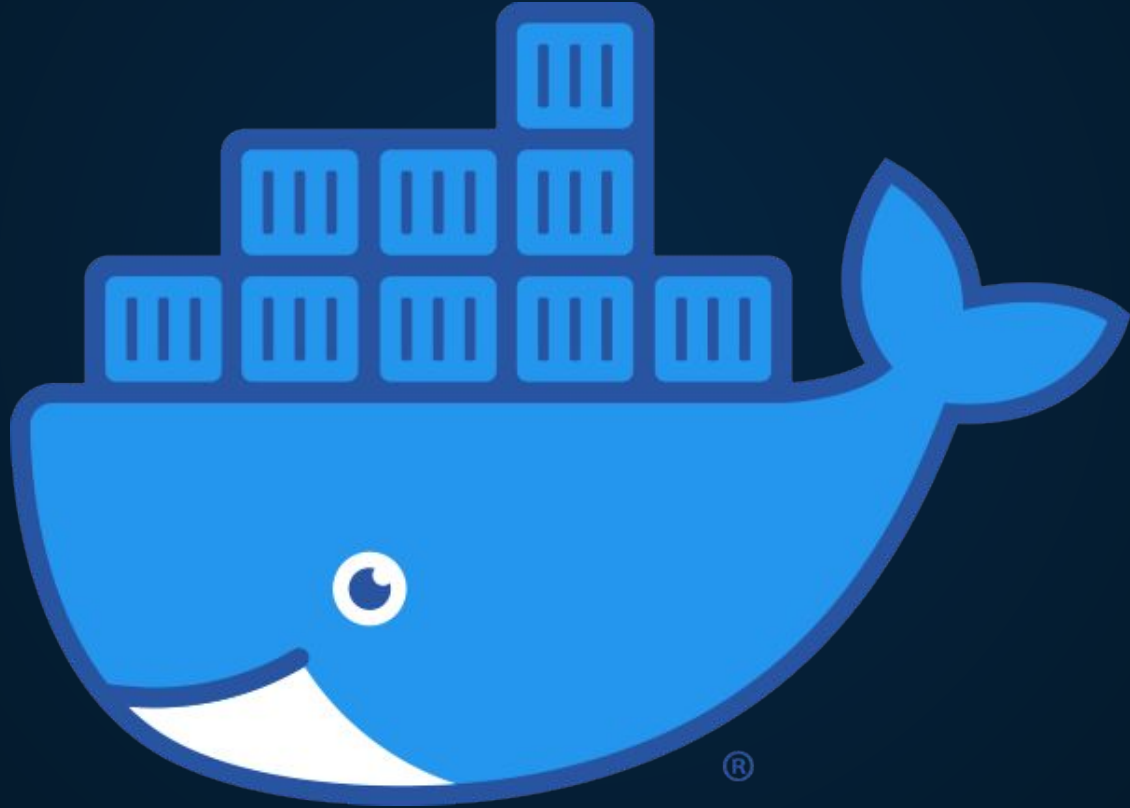
Objetivos

- ❖ **Instalar, configurar y administrar el software de base y de aplicación del sistema.**
- ❖ **Comprender el concepto de contenedor.**
- ❖ **Conocer elementos básicos de Docker**
- ❖ **Ser capaz de crear imágenes personalizadas**
- ❖ **Ser capaz de gestionar volúmenes en Docker**
- ❖ **Ser capaz de gestionar la red en Docker.**
- ❖ **Ser capaz de utilizar Docker Compose.**

Contenidos

- ◆ ¿Qué es Docker y cómo funciona?
- ◆ Introducción a los contenedores y a Docker
- ◆ Instalación de Docker
- ◆ Primeros pasos con Docker
- ◆ Creación de imágenes personalizadas con Docker.
- ◆ Redes y volúmenes en Docker.
- ◆ Docker compose.
- ◆ Utilidades para gestionar Docker fácilmente.

Be Docker my friend



¿Qué es Docker?

Tecnología que permite ejecutar un proceso totalmente aislado del resto de los procesos de la máquina.

Este proceso definirá todas las dependencias que necesita para ser ejecutado.

Ejemplo: “Llegamos a un proyecto nuevo del que se necesita instalar y configurar una serie de dependencias para su funcionamiento o que solo funciona sobre una distribución concreta de Linux. Con docker se consigue empaquetar en una imagen de forma que las dependencias están disponibles a la hora de ejecutar ese proceso, herramienta o app”

Es una virtualización a nivel de sistema operativo en Linux y también en Windows y MacOS

Build, Ship and Run, any app, anywhere

Un programa, herramienta o proceso será:

- Portable: capaz de ejecutarse en distintos tipos de SO's que permitan la ejecución de docker haciendo que esta se ejecute exactamente igual en todas ellas.
- Immutable: en todos los SO's que se ejecute esa imagen con todas las dependencias no cambia.
- Ligero

Imagen

Una imagen se configura a través de un fichero llamado Dockerfile. En él, se definen los siguientes puntos:

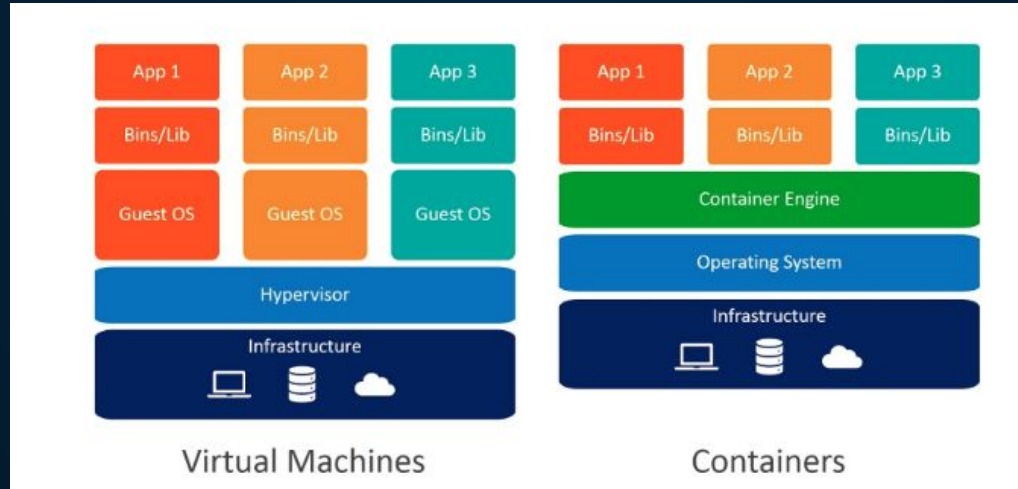
- OS sobre el que se va a ejecutar la aplicación (microlinux (ubuntu, xenial, etc...))
- software: listado de librerías a instalar para que nuestra aplicación pueda funcionar (apache, mysql, php o python, módulos del lenguaje)
- aplicación

Cuando el Dockerfile se construye, se crea una imagen.

Esta imagen se usará para correr los contenedores.

Contenedores

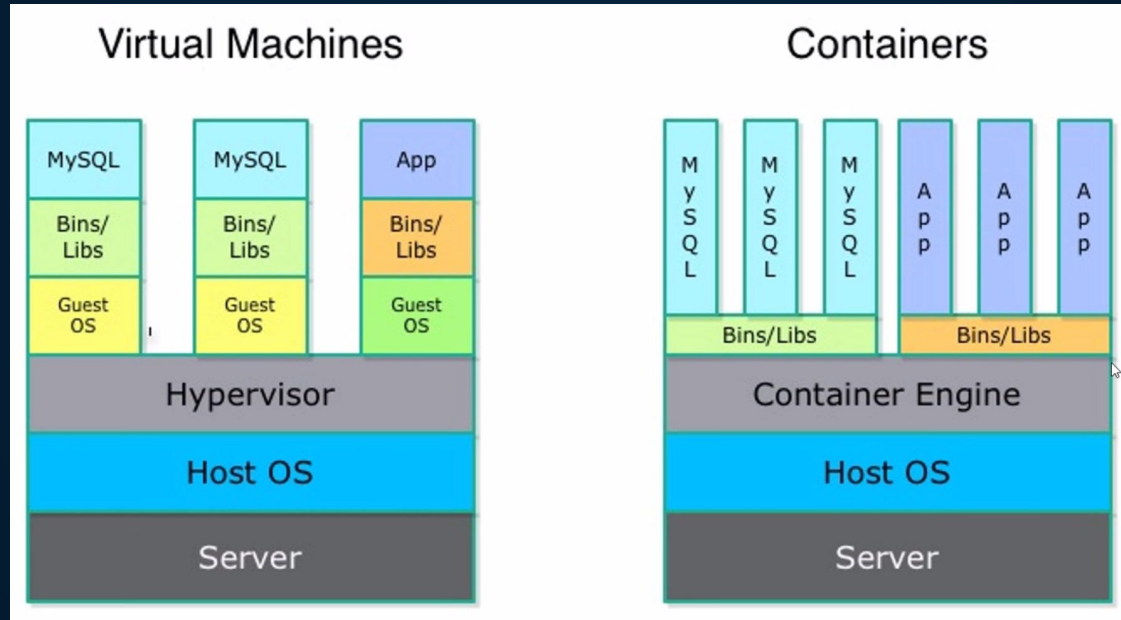
- Permite empaquetar todo tipo de aplicaciones y portarlas a otros sistemas con muy poco peso y con todas sus dependencias. Crean como una imagen de una aplicación independiente del lenguaje de programación.



Características de los contenedores

- Autogestión
- Fiabilidad
- Aplicaciones libres de dependencias instaladas en el sistema base
- Múltiple despliegue de contenedores en un mismo equipo físico
- Rápida puesta en marcha
- Capacidad para ejecutar amplia gama de app's
- Compatibilidad multisistema
- Se puede compartir los contenedores (DockerHub)

VM's vs Containers



Instalación

- Autogestión
- Fiabilidad
- Aplicaciones libres de dependencias instaladas en el sistema base
- Multiple despliegue de contenedores en un mismo equipo físico
- Rápida puesta en marcha
- Capacidad para ejecutar amplia gama de app's
- Compatibilidad multisistema
- Se puede compartir los contenedores (DockerHub)

Instalación

- <https://www.docker.com/products/docker-desktop> (MacOS o Windows)
- <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-22-04> (Linux)
- Para Windows, seguir guía de instalación adjunta en Aules.