

~~HENRY~~



BIG DATA



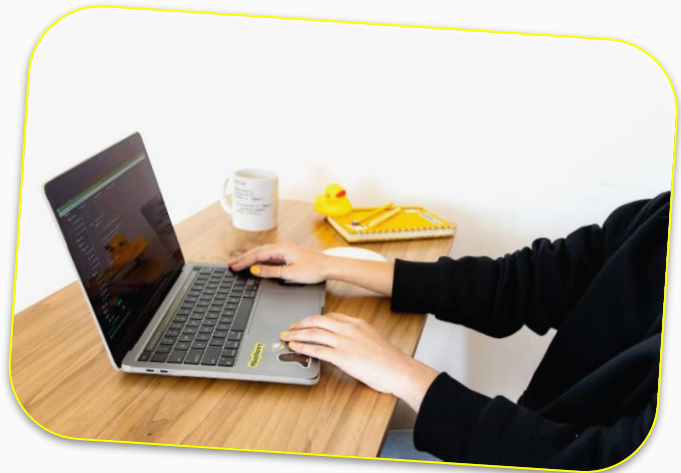


OBJETIVOS DE CLASE

- Comprender** el concepto Big Data y lo qué implican las 3V's de Big Data
- Analizar** algunos Casos de Uso
- Reconocer** las diferencias entre un Data Lake y un Data Warehouse
- Comprender** el concepto de Máquinas Virtuales y la Virtualización
- Explorar** qué es Docker y la Containerización



AGENDA



- Big Data
- La era de los Datos
- 3V's de Big Data
- Casos de Uso
- Data Lake
- Estrategias de Procesamiento
- Arquitectura Data Lake
- Máquinas Virtuales
- Docker
- ¿Qué es una imagen?
- ¿Qué es un contenedor?
- Docker Compose

La era de los datos





¿Qué es la era de los datos?

Los datos se han convertido en uno de los recursos más **valiosos y poderosos** de nuestra sociedad. Atrás quedaron los días en que la información se obtenía de fuentes limitadas y se procesaban manualmente. Ahora, vivimos en un mundo en el que los datos se generan a una velocidad vertiginosa y se almacenan en cantidades masivas.

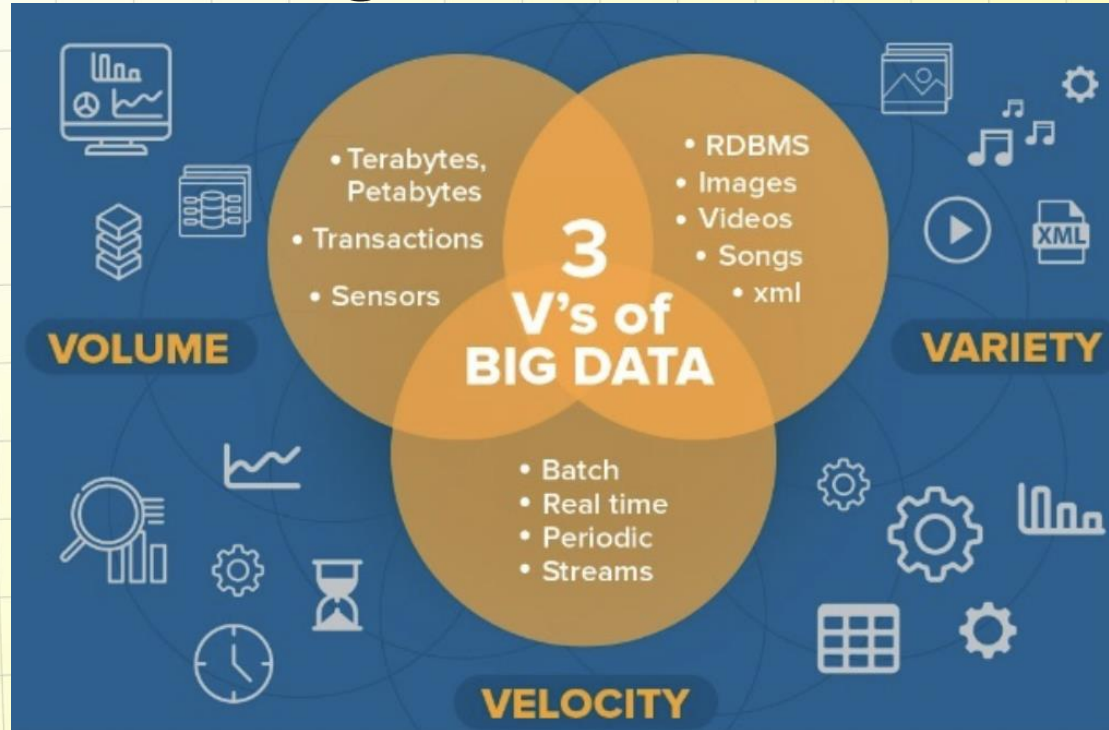




Las 3's V del Big Data



Las 3's V del Big Data



La era de los datos se caracteriza por tres factores principales: **volumen**, **velocidad** y **variedad**.



Casos de uso



**FRAUD
DETECTION**



**CLV
PREDICTION**
(CUSTOMER LIFE VALUE)



**RECOMMENDATION
ENGINE**



**MARKET BASKET
ANALYSIS**



**WARRANTY
ANALYTICS**



**INVENTORY
MANAGEMENT**



**CUSTOMER
SENTIMENT ANALYSIS**



**PRICE
OPTIMIZATION**



MERCHANDISING



Data Lake



Data Lake

¿Qué es?

Arquitectura de almacenamiento que permite almacenar **grandes volúmenes de datos** en su forma original, sin la necesidad de estructurarlos o transformarlos previamente

Características

Almacenamiento escalable

Integración de datos:

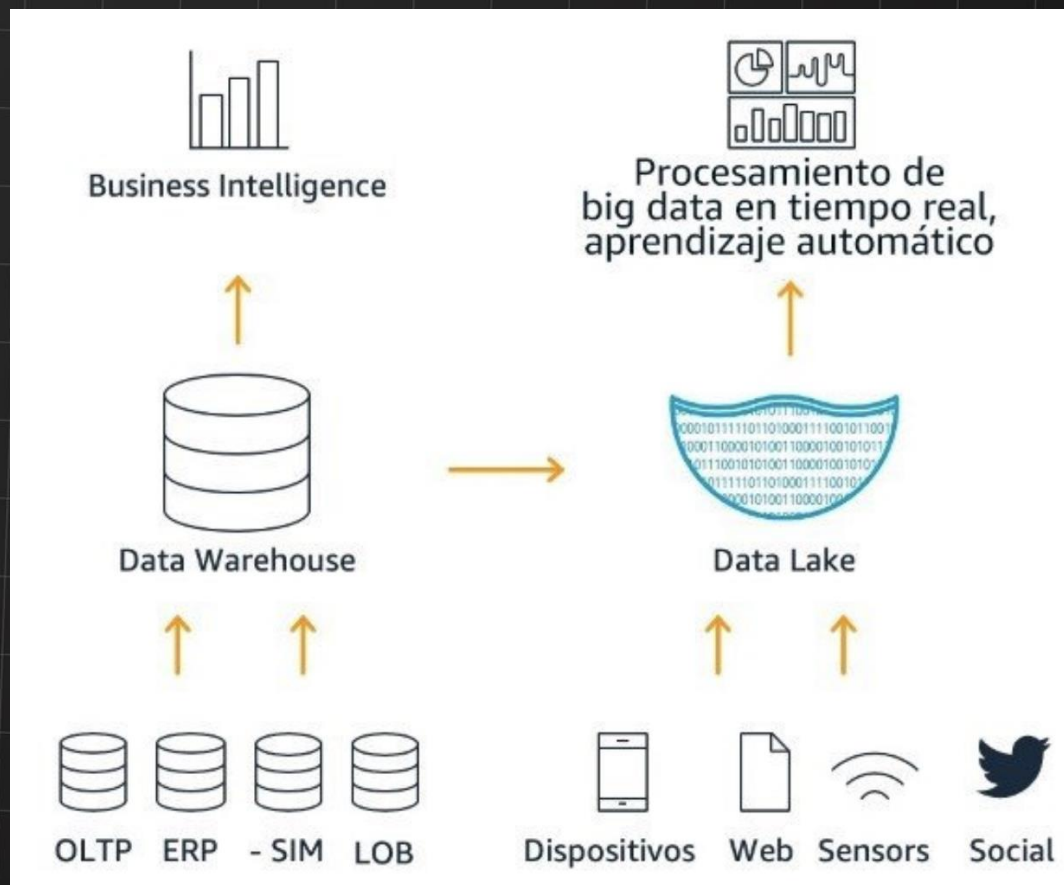
Datos en bruto

Metadatos

Seguridad y acceso controlado

¿Para qué sirve?

Los usuarios pueden acceder a los datos en bruto y aplicar diferentes técnicas de análisis, como **minería de datos**, aprendizaje automático o procesamiento de lenguaje natural, para obtener información valiosa y extraer conocimientos significativos.



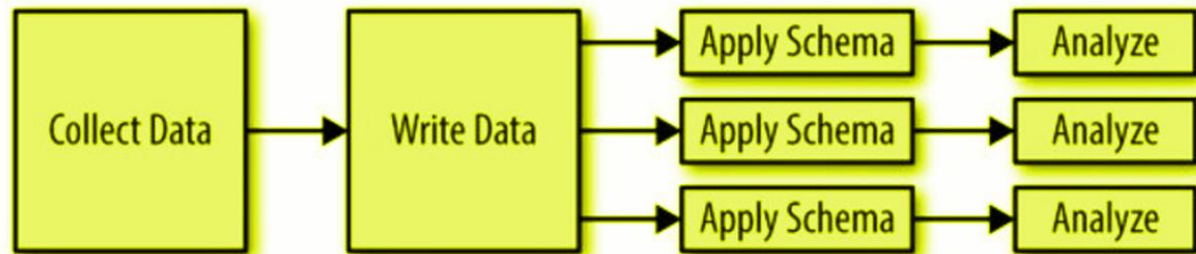


Estrategias de procesamiento o



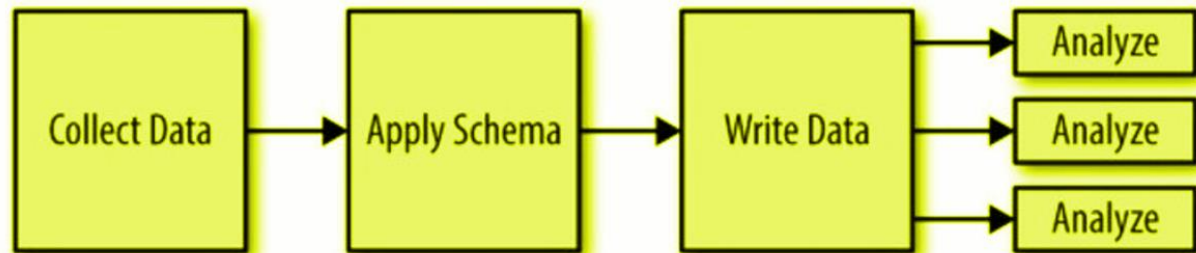
Data Lake

Schema-on-Read



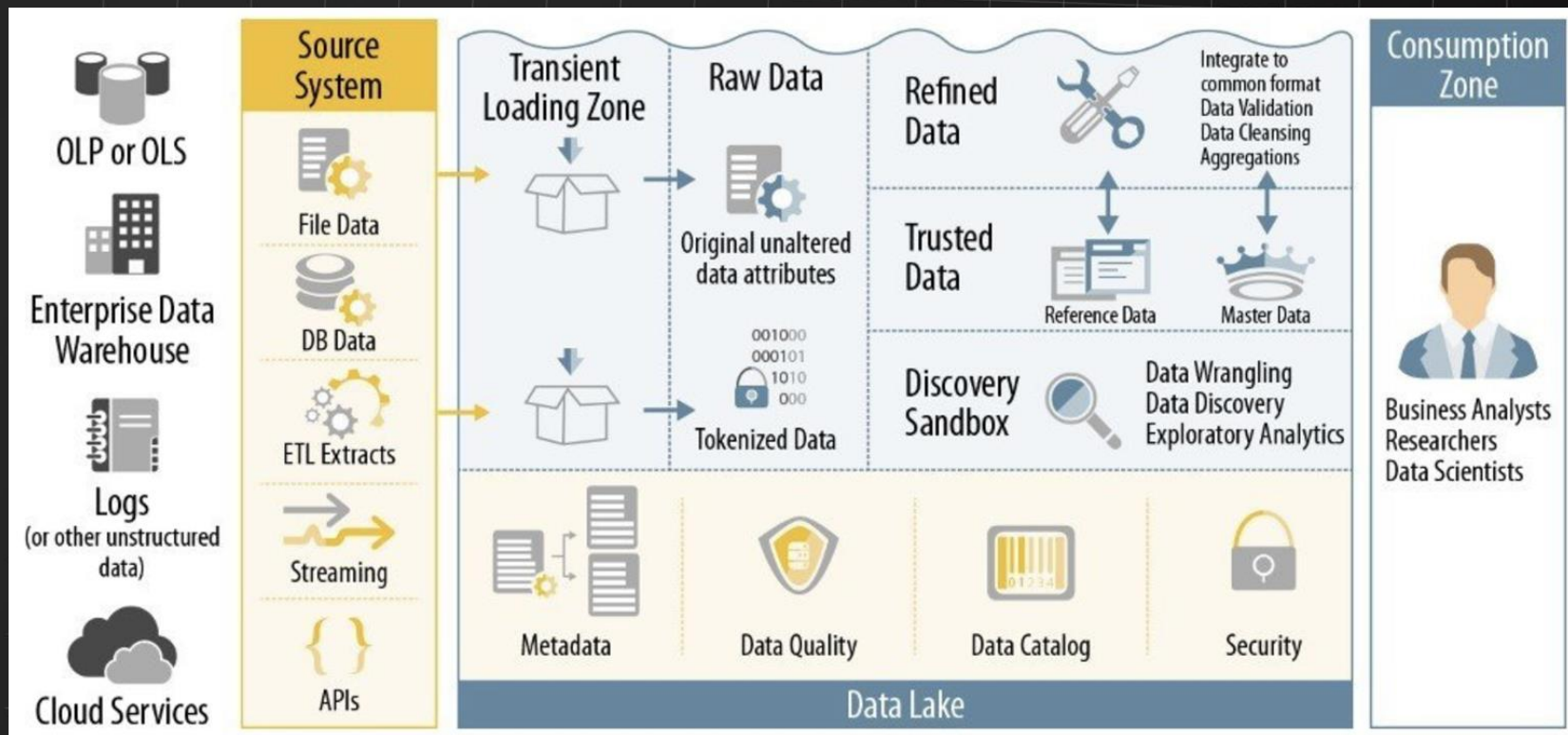
Data Warehouse

Schema-on-Write



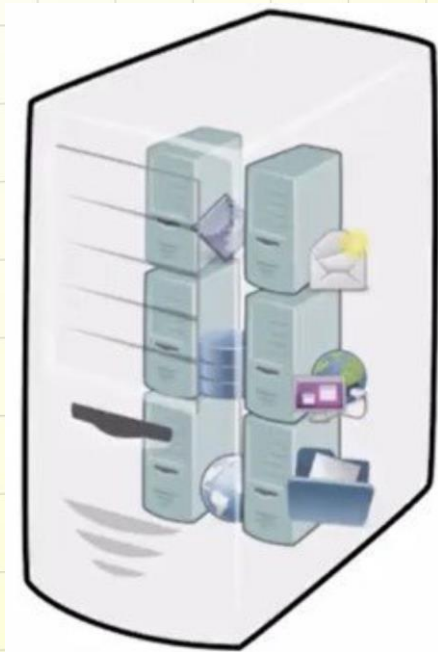


Arquitectura del Data Lake





virtualización



¿Qué es?

Permite a los científicos de datos crear entornos aislados y reproducibles para desarrollar y probar algoritmos, modelos y pipelines de datos sin afectar el entorno de producción.

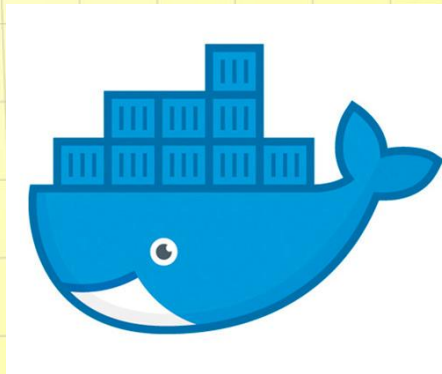


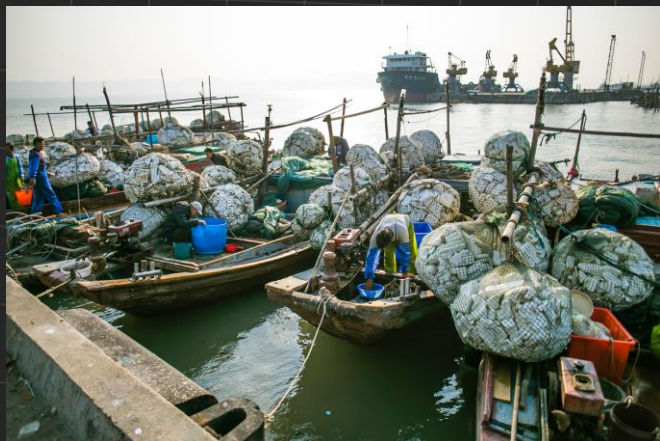
Docker y la containerización

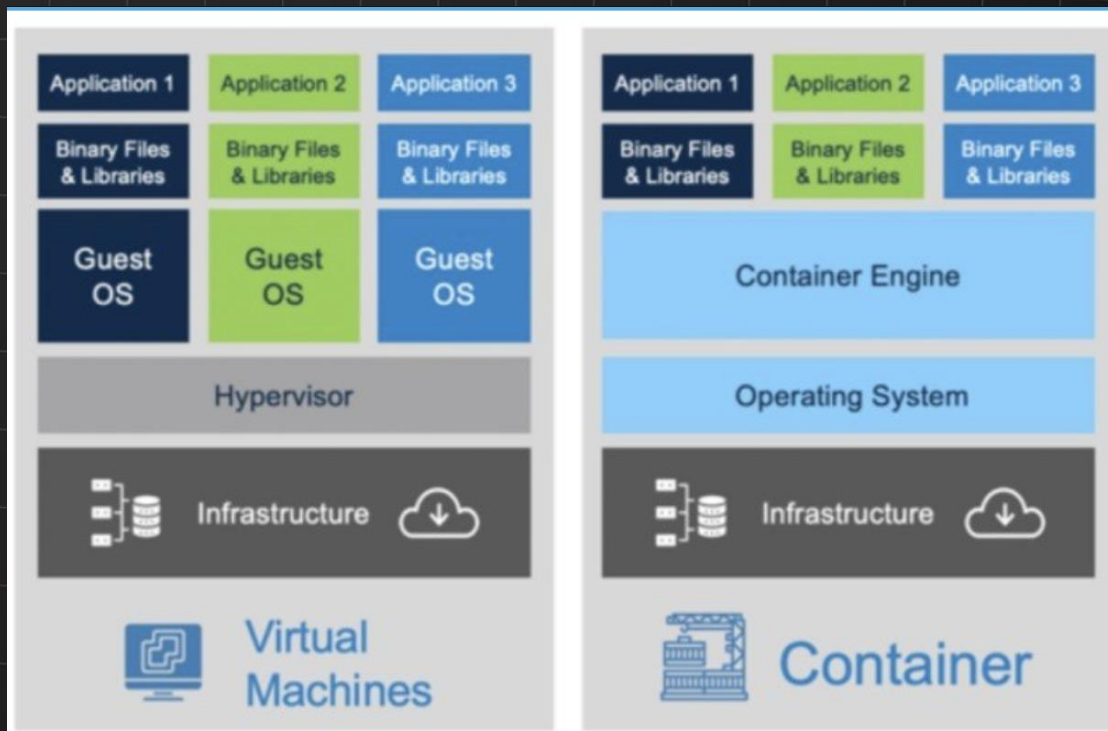


¿Qué es?

Docker es una **plataforma de virtualización** basada en contenedores que facilita la creación, gestión y despliegue de aplicaciones en entornos aislados y portables. Los contenedores Docker permiten **encapsular** aplicaciones junto con sus dependencias y configuraciones, lo que simplifica el desarrollo, la colaboración y el despliegue de aplicaciones en **diferentes entornos**.









Componentes del Docker Engine



Componentes

Docker daemon:

Es el centro de docker, por medio del cual, es posible la comunicación con los servicios de docker.

REST API:

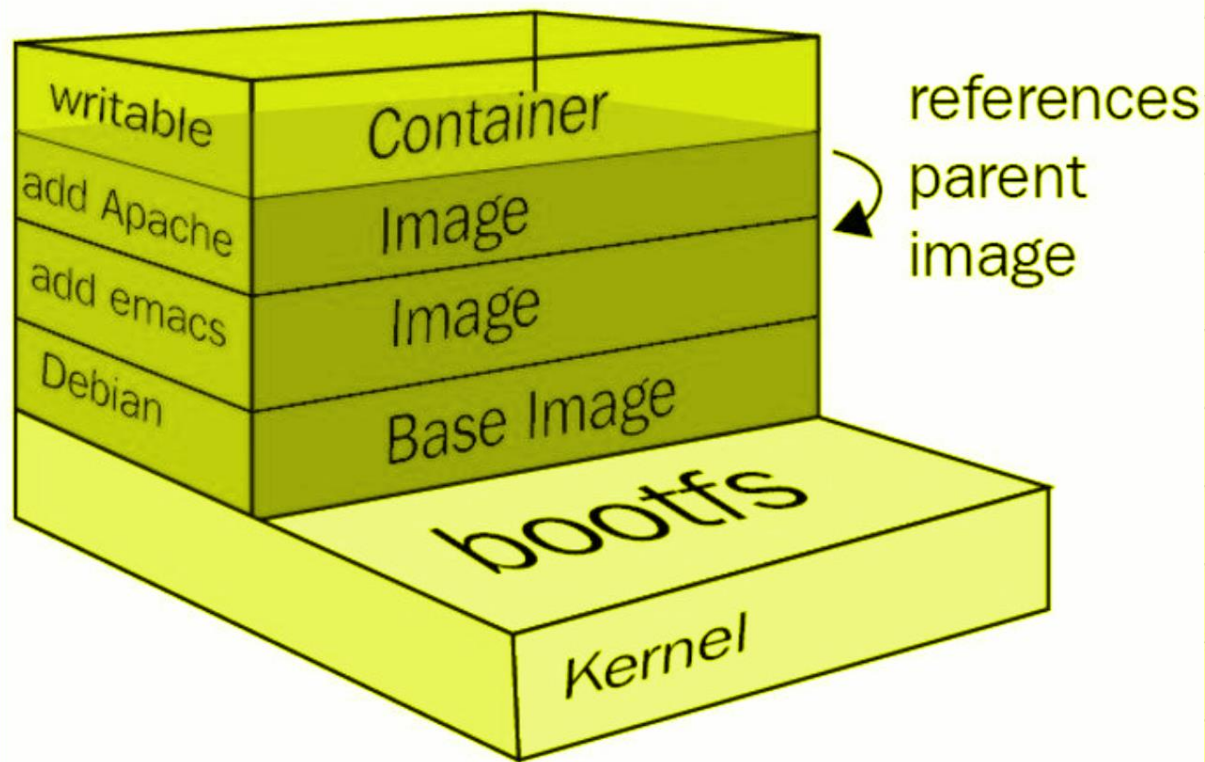
Como cualquier otra API, es la que nos permite visualizar docker de forma "gráfica".

Cliente de docker

Permite la comunicación con el centro de docker (Docker Daemon) que por defecto es la línea de comandos



**¿Qué es una
imagen?**





contenedores



¿Qué es un contenedor?



Entidad lógica, no tiene el límite estricto de las máquinas virtuales.



Ejecuta sus procesos de forma nativa.



Los procesos que se ejecutan dentro de los **contenedores** ven su universo como el contenedor lo define, no pueden ver más allá del contenedor, a pesar de estar corriendo en una máquina más grande.



No tienen forma de **consumir más recursos** que los que se les permite.



Sector del disco: Cuando un contenedor es ejecutado, el **daemon de docker** establece a qué parte puede acceder.



Docker hace que los procesos dentro de un contenedor estén aislados del resto del sistema, no le permite ver más allá.



Cada contenedor tiene un **ID único**, también tiene un nombre.



Docker compose



Para qué sirve

Simplifica el uso de **Docker** a partir de archivos YAML, con los que es más sencillo crear contenedores, conectarlos, habilitar puertos, volúmenes, etc.

Crea diferentes contenedores y al mismo tiempo, en cada contenedor, diferentes servicios, unirlos a un volumen común, iniciarlos y apagarlos, etc.

Componente fundamental para poder construir aplicaciones y microservicios.

Permite poder instruir al Docker Engine a realizar tareas, programáticamente siendo ésta la clave: **La facilidad para dar una serie de instrucciones, y luego repetirlas en diferentes ambientes**

Describe de **forma declarativa** la arquitectura de servicios necesaria en un archivo donde se declara lo que debe suceder.



RESUMEN DE LA CLASE

- ✓ Las 3V de los datos
- ✓ Docker
- ✓ Imagen
- ✓ Contenedores
- ✓ Docker Compose



¿PREGUNTAS?



**¡Muchas
gracias!**