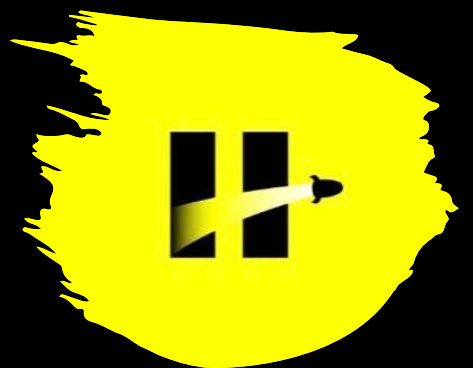
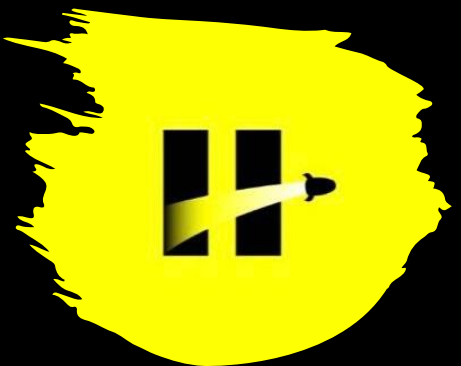
M4
BIG DATA

Clase 2



Objetivos de la clase

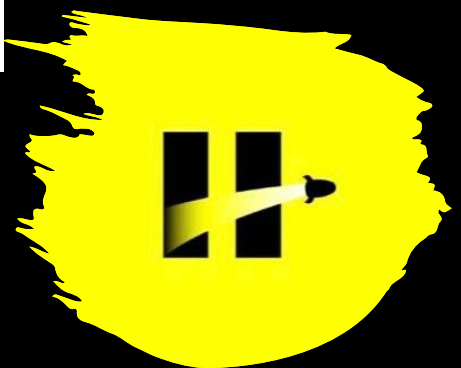
- Seguir Conociendo el ecosistema de Hadoop
- Identificar Vendors de Hadoop
- Virtualización
  - Que es Virtualizar
  - Ventajas de virtualización
- Docker
  - Que es Docker?
  - Componentes de Docker
- Practica



Frameworks Hadoop

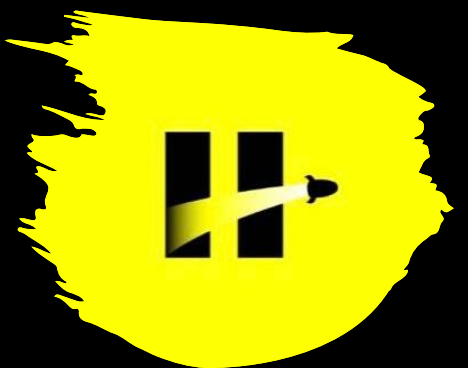
Apache Hive	Consultas SQL sobre Hadoop
Apache Sqoop	Transferencia de datos entre bases relacionales y Hadoop
Apache Spark	Procesamiento en memoria de ETL's, Streaming, Machine Learning y Grafos
Apache Kafka	Sistema de colas de mensajería que utiliza el patrón productor/consumidor
Apache HBase	Base de datos NoSQL de tipo columnar que se ejecuta sobre HDFS
Apache Ranger	Administración de políticas de seguridad sobre componentes de Hadoop
Apache Atlas	Herramienta que provee funcionalidades de Data Governance sobre Hadoop
Apache Nifi	Orquestación de flujos de datos desde y hacia Hadoop

Un Framework es una estructura base utilizada como punto de partida para elaborar un proyecto con objetivos específicos.



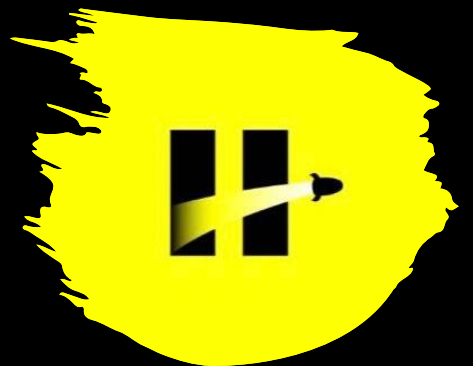
Apache Hive

Hive es una tecnología distribuida diseñada y construida sobre **Hadoop**. Que nos permite hacer consultas y analizar grandes cantidades de datos almacenados en **HDFS**. Tiene un lenguaje de consulta llamado **HiveQL** o HQL que internamente transforma las consultas SQL en trabajos MapReduce que ejecutan en Hadoop. El lenguaje de consulta HQL es un dialecto de SQL, que no sigue el estándar ANSI SQL, sin embargo es muy similar.



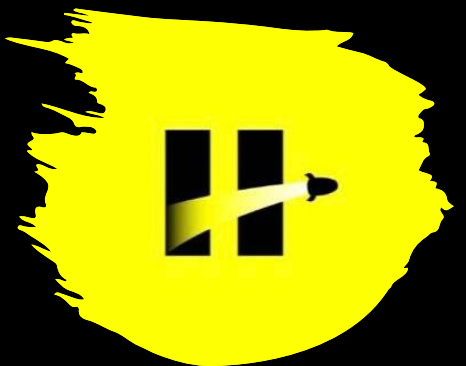
Apache Sqoop

Apache Sqoop es una **herramienta de línea de comandos** desarrollada para transferir grandes volúmenes de datos de bases de datos relacionarles a Hadoop, de ahí su nombre que viene de la fusión de SQL y Hadoop. Concretamente transforma datos relacionarles en Hive o HBase en una dirección y en la otra de HDFS a datos relacionares como **MySQL, Oracle, Postgress** o a un **data warehouse**.



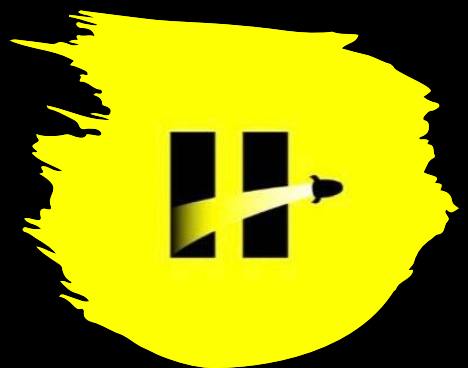
Apache Kafka

Apache Kafka es un sistema de mensajería y una plataforma completa de streaming y de procesamiento de datos en tiempo real. Nos proporciona la capacidad de publicar y procesar flujos de eventos de forma escalable y tolerante a fallos.



Apache HBase

HBase es un sistema de gestión de bases de datos no relacionales orientado a columnas que se ejecuta sobre Hadoop Distributed File System (HDFS). HBase proporciona una forma tolerante a fallas de almacenar conjuntos de datos escasos, que son comunes en muchos casos de uso de big data. Es muy adecuado para el procesamiento de datos en tiempo real o el acceso aleatorio de lectura / escritura a grandes volúmenes de datos.



Apache Ranger

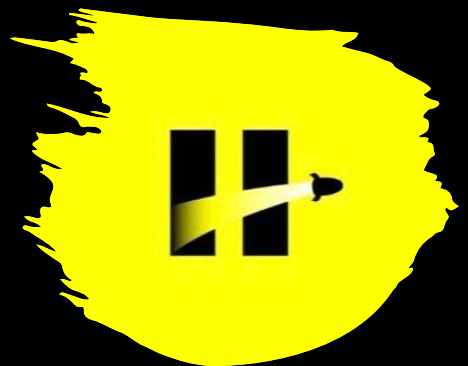
Apache Ranger ofrece un enfoque integral de la seguridad de un clúster de Hadoop. Proporciona una plataforma centralizada para definir, administrar y gestionar las políticas de seguridad de manera uniforme en todos los componentes Hadoop.

Apache Atlas

Apache Atlas es una herramienta *open-source*, con licencia Apache 2.0, para la gobernanza del dato la cual permite la integración con todo el ecosistema de datos de las empresas.

Apache Nifi

Apache NiFi es un **sistema distribuido dedicado a extraer, transformar y cargar datos (ETL)**. Es **Open Source** y está desarrollado y mantenido por la Apache Software Foundation.



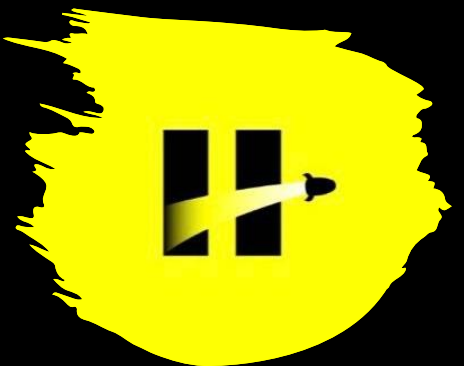
Otras Tecnologías Big Data

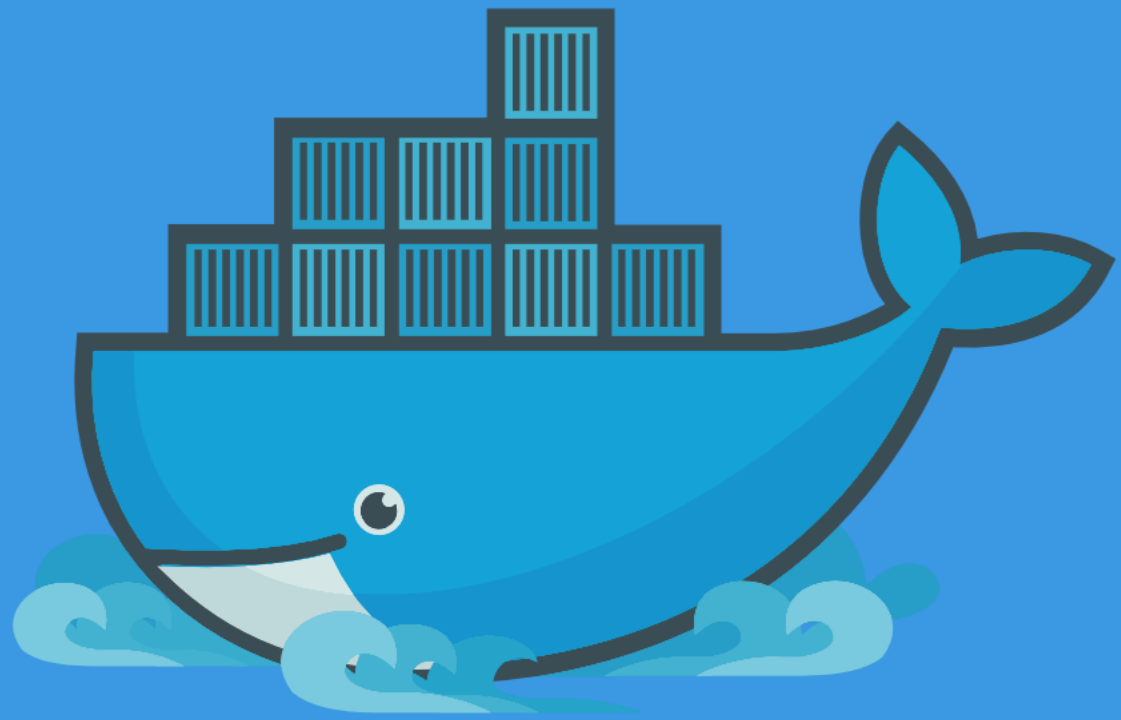
Vendors

Un Vendor es una empresa o persona física que proporciona bienes o servicios a otras personas o empresas.

Algunos Vendors

- Cloudera https://www.cloudera.com/
- Amazon EMR https://aws.amazon.com/es/emr/
- Azure HDInsight https://azure.microsoft.com/es-es/services/hdinsight/
- IBM Analytics Engine https://cloud.ibm.com/catalog/services/analytics-engine
- Google Dataproc https://cloud.google.com/dataproc
- MapR https://mapr.com/

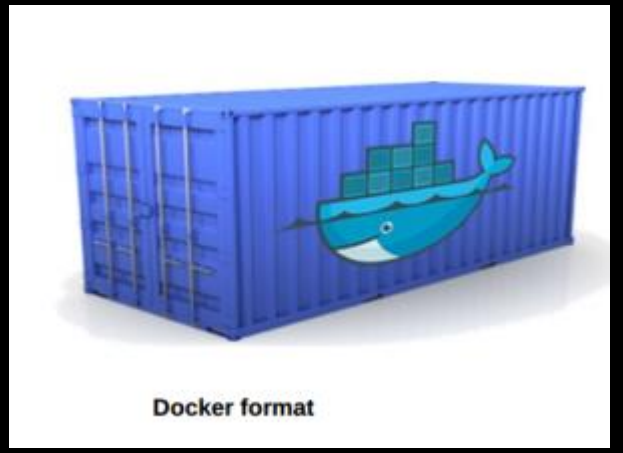




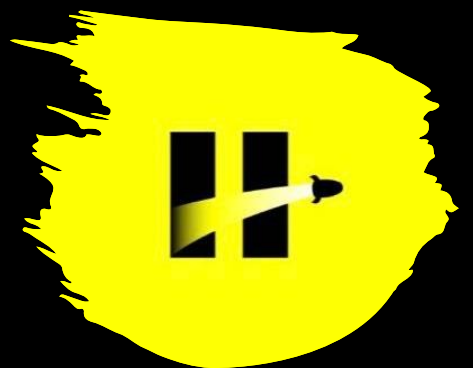
Que Es Docker?

La idea detrás de Docker es crear contenedores **ligeros y portables** para las aplicaciones software que puedan ejecutarse en cualquier máquina con Docker instalado, **independientemente del sistema operativo que la máquina tenga por debajo**, facilitando así también los despliegues.

Pero….. ¿Qué es un contenedor?

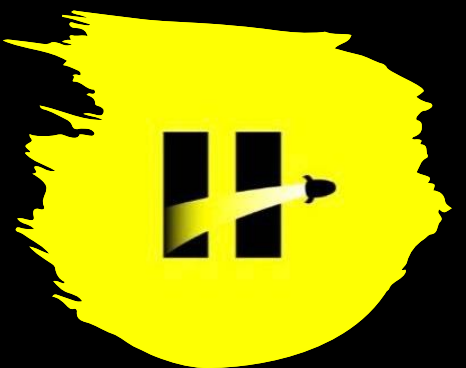


Este concepto ya es antiguo, y viene de Linux, pero por hacerte un símil con el mundo real, imagina en tu cabeza un contenedor de esos que suelen llevar los barcos de mercancías, que contiene distintos productos. Es algo auto contenido en sí, que se puede llevar de un lado a otro de forma independiente, es portable.



Como funcionan las Aplicaciones?

- Ahora, volviendo al software, para que podamos acceder como usuarios normales a una aplicación, dicha aplicación software necesita estar ejecutándose en una máquina, en un ordenador. Pero además, dependiendo del tipo de aplicación, dicho ordenador también necesita tener instaladas una serie de cosas para que la aplicación se ejecute correctamente: cierta versión de Java instalado, un servidor de aplicaciones (p.e tomcat, que es el software que realmente estará ejecutando mi aplicación y haciendo que pueda interactuar con ella).

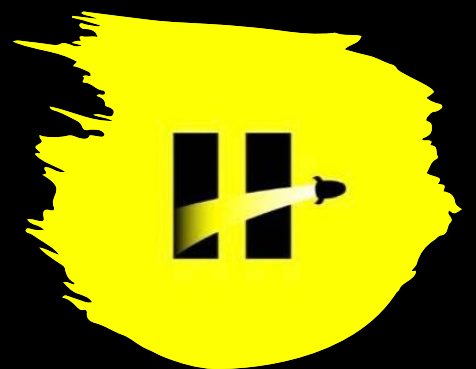


La Magia de Docker

Docker, me **permite meter en un contenedor** (“una caja”, algo auto contenido, cerrado**) todas aquellas cosas que mi aplicación necesita** para ser ejecutada (java, Maven, tomcat…) y la propia aplicación. Así yo me puedo llevar ese contenedor a cualquier máquina que tenga instalado Docker y ejecutar la aplicación sin tener que hacer nada más, ni preocuparme de qué versiones de software tiene instalada esa máquina, de si tiene los elementos necesarios para que funcione mi aplicación , de si son compatibles…

Yo ejecutaré mi aplicación software desde el contenedor de Docker, y dentro de él estarán todas las librerías y cosas que necesita

dicha aplicación para funcionar correctamente.

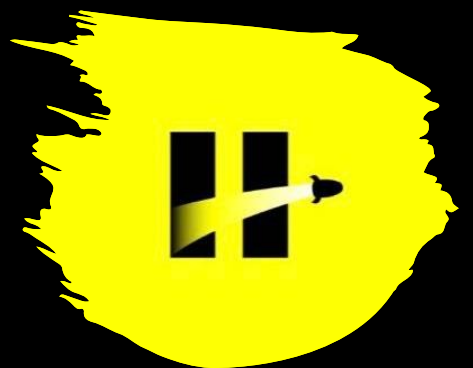


Pero entonces es una maquina virtual?

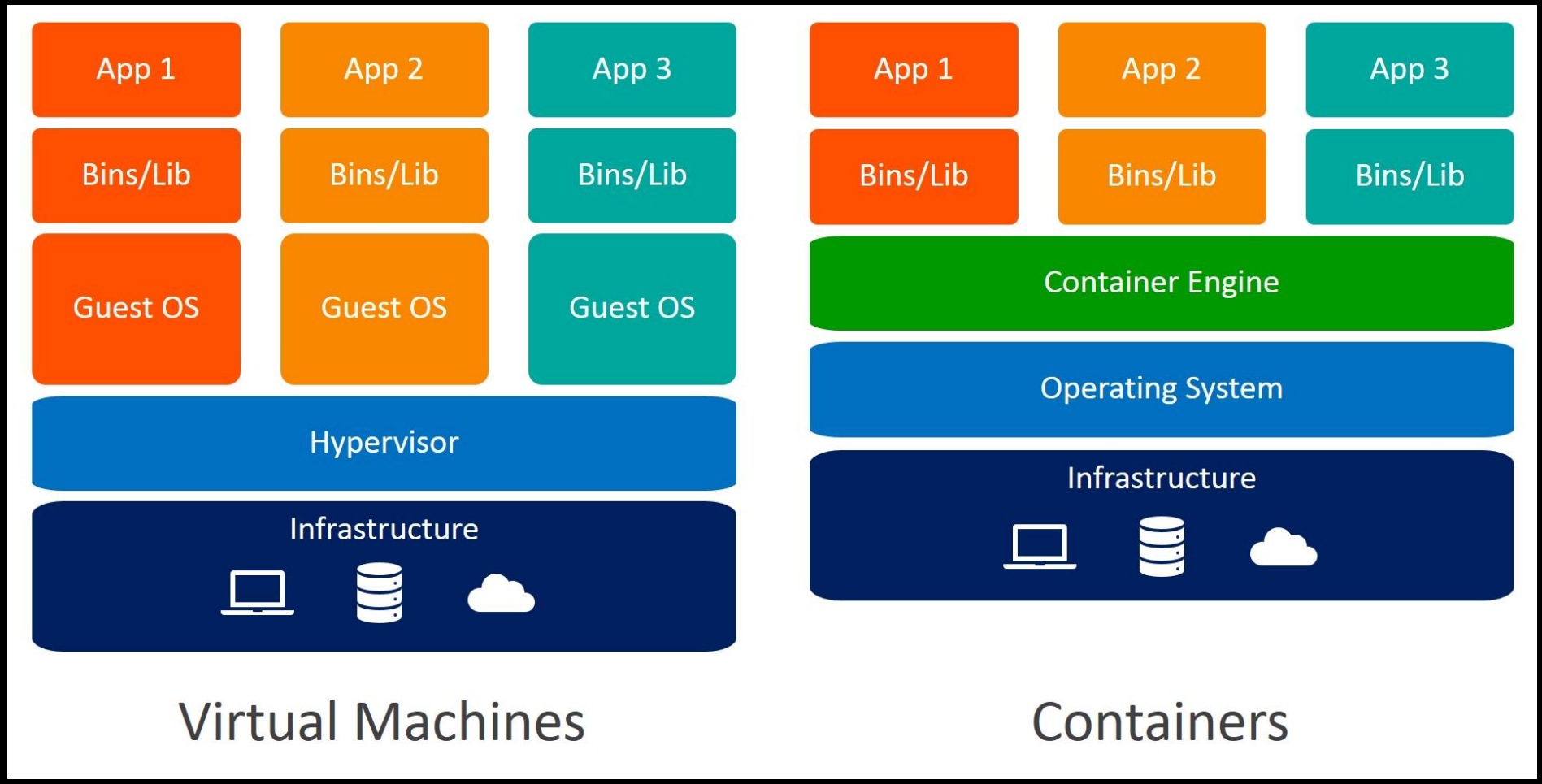
Puede que cuando hablamos de que en un mismo ordenador podemos tener varios contenedores Docker te hayas preguntado: ¿y esto no es lo mismo que una máquina virtual?
Realmente el concepto es algo similar, pero un contenedor no es lo mismo que una máquina virtual.
Un contenedor **es más ligero**, ya que mientras que a una máquina virtual necesitas instalarle un sistema operativo para funcionar, un contenedor de Docker funciona utilizando el sistema operativo que tiene la máquina en la que se ejecuta el contenedor.

Digamos que el contenedor de **Docker toma los recursos más básicos**, que no cambian de un ordenador a otro del sistema operativo de la máquina en la que se ejecuta. Y los aspectos más específicos del sistema que pueden dar más problemas a la hora de llevar el software de un lado a otro, se meten en el interior del contenedor.

Que un contenedor Docker tome los aspectos básicos de funcionamiento del sistema operativo de la máquina en la que se ejecuta lo vuelve más ligero que una máquina virtual.



Docker Vs Virtual Machine



Vamos A GitHub