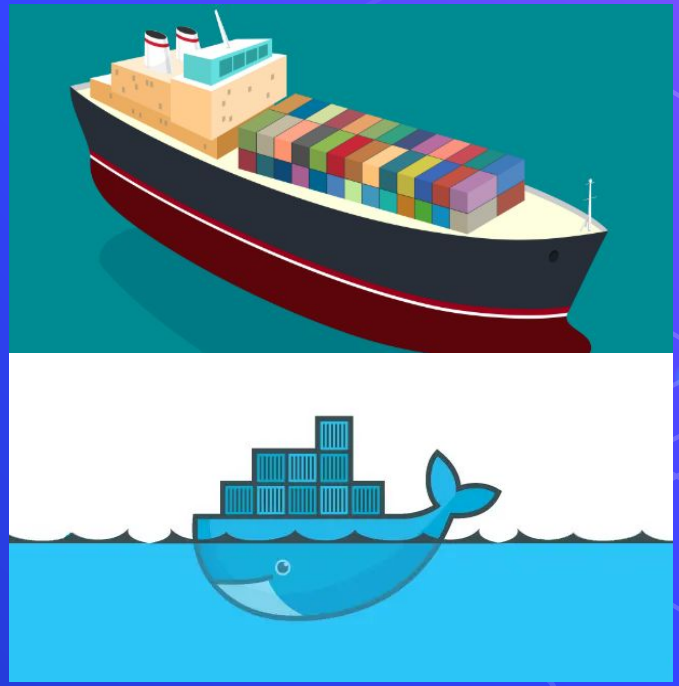
Contenedores

¿Qué son los contenedores?

Los contenedores que vemos normalmente son aquellos que llevan un producto interno, y son transportados en un barco de carga de un lugar a otro.



Los contenedores de software son algo parecidos, dentro de ellos podemos alojar todas las dependencias que nuestra aplicación necesite para ser ejecutada: empezando por el propio código, las librerías del sistema, el entorno de ejecución o cualquier tipo de configuración.

https://www.xataka.com/otros/docker-a-kubernetes-entendiendo-que-contenedores-que-mayores-revoluciones-industria-desarrollo
https://www.netapp.com/es/devops-solutions/what-are-containers/

¿Por qué usarlo?

Menos sobrecarga

Los contenedores requieren menos recursos del sistema que los entornos de máquinas virtuales tradicionales o de hardware porque no incluyen imágenes del sistema operativo.

Mayor portabilidad

Las aplicaciones que se ejecutan en contenedores se pueden poner en marcha fácilmente en sistemas operativos y plataformas de hardware diferentes.

Funcionamiento más constante

Los equipos de DevOps saben que las aplicaciones en contenedores van a ejecutarse igual, independientemente de dónde se pongan en marcha.

Mayor eficiencia

Los contenedores permiten poner en marcha, aplicar parches o escalar las aplicaciones con mayor rapidez.

Mejor desarrollo de aplicaciones

Los contenedores respaldan los esfuerzos ágiles y de DevOps para acelerar los ciclos de desarrollo, prueba y producción.

Proceso de Creación

Creación Dockerfile

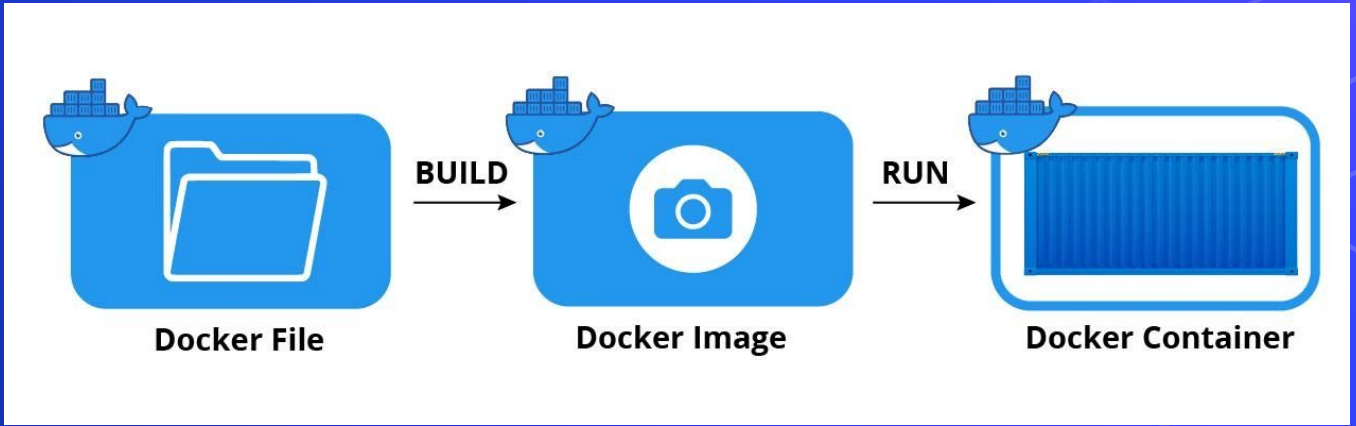
Para este paso se debe crear un documento donde se especifican los steps para generar una imagen de la app.

Creación de Imagen

Luego de tener el Dockerfile, se debe generar una imagen, la cual va a correr el paso a paso declarado en el archivo.

Correr Imagen

Luego de tener la imagen creada, se corre para poner en funcionamiento nuestra App.



LAB 1

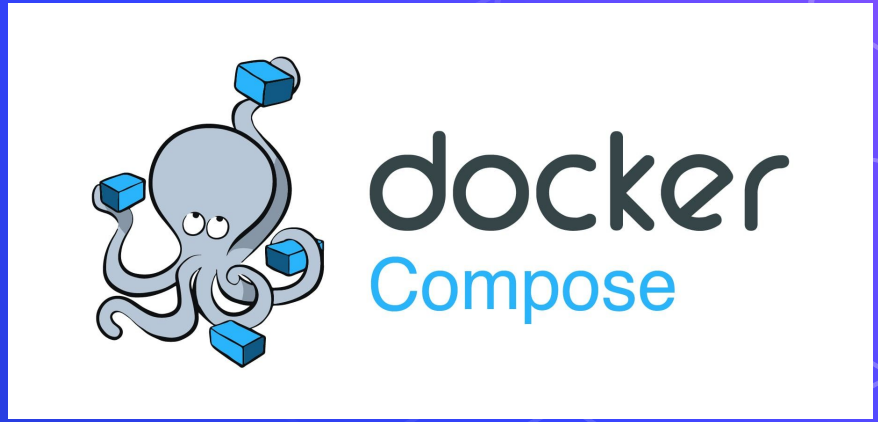
Objetivo: Crear un Dockerfile de una app, construir la imagen, correr la imagen y probarla localmente.

Docker Compose

¿Qué es Docker Compose?

Docker Compose es una herramienta para definir y ejecutar aplicaciones Docker multicontenedor. Permite definir los servicios, las redes y los volúmenes de la aplicación en un único archivo, y poner en marcha la aplicación con un único comando.

Docker Compose funciona en todos los entornos: producción, puesta en escena, desarrollo, pruebas, así como flujos de trabajo de CI.



https://learn.microsoft.com/es-es/azure/cognitive-services/containers/docker-compose-recipe
https://jsgiraldoh.io/Blog/Qu%C3%A9-es-docker-compose
https://www.dongee.com/tutoriales/que-es-docker-compose/#:~:text=Tutoriales%20DongeeMauricio-,%C2%BFQu%C3%A9%20diferencia%20hay%20entre%20Docker%20y%20Docker%20Compose%3F,y%20ejecutar%20aplicaciones%20docker%20multicontenedor

¿Por qué usarlo?

Simplifica los comandos de Docker

Ejecutando docker-compose up se iniciarán todos los contenedores definidos en tu archivo docker-compose en lugar de iniciar manualmente cada contenedor con el comando docker run..

Definir y conectar fácilmente múltiples contenedores

docker-compose te permite definir y conectar fácilmente múltiples contenedores para tu aplicación, sin necesidad de exponer puertos o enlazar contenedores.

Gestionar fácilmente los recursos

docker-compose te permite gestionar fácilmente los recursos para tu aplicación, como la red y los volúmenes, en un único archivo.

Proceso de Creación

1. Creación Dockerfile(s)

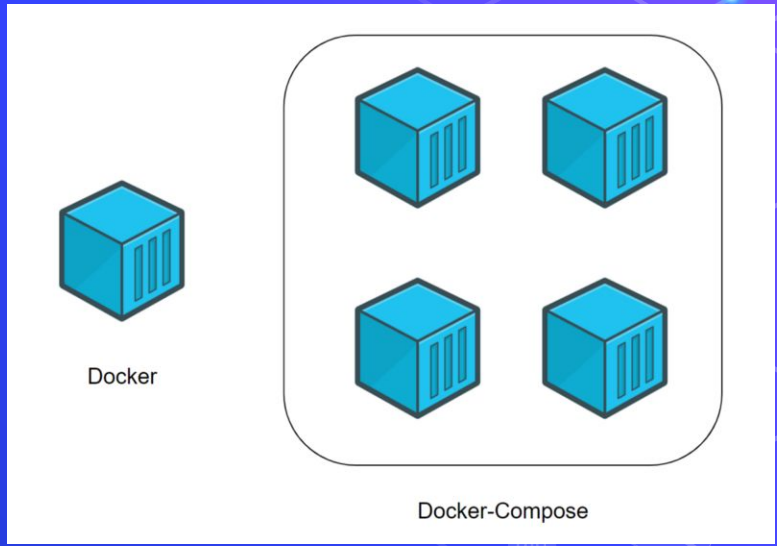
Definir el entorno de la aplicación con un Dockerfile para que pueda reproducirse en cualquier lugar.

2. Creación de docker-compose

Defina los servicios que componen su aplicación docker-compose.yml para que puedan ejecutarse juntos en un entorno aislado.

3. Correr docker-compose

Ejecutar *docker-compose up* y se iniciará y ejecutará toda su aplicación.



LAB 2

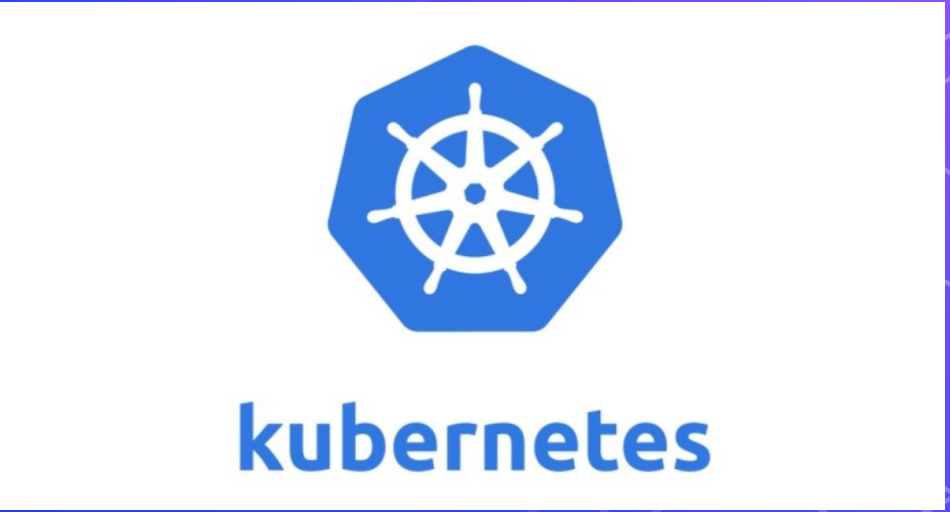
Objetivo: Crear dos servicios (API y BD), construir el archivo docker-compose, correrlo, probarlo y eliminarlo.

Kubernetes
K8s

¿Qué son Kubernetes?

Kubernetes es una plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios. Kubernetes facilita la automatización y la configuración declarativa. Tiene un ecosistema grande y en rápido crecimiento.

Es una herramienta de orquestación de contenedores, permite alta disponibilidad (replicas), escalabilidad, disaster recovery y baja downtime.



https://kubernetes.io/es/docs/concepts/overview/what-is-kubernetes/
https://cloud.google.com/learn/what-is-kubernetes?hl=es-419
https://www.cncf.io/blog/2019/08/19/how-kubernetes-works/

¿Por qué usarlo?

Operaciones automatizadas

Kubernetes cuenta con comandos integrados para manejar gran parte del trabajo pesado que forma parte de la administración de aplicaciones, lo que te permite automatizar las operaciones diarias. Puedes asegurarte de que las aplicaciones siempre se ejecuten de la manera que planeaste.

Abstracción de la infraestructura

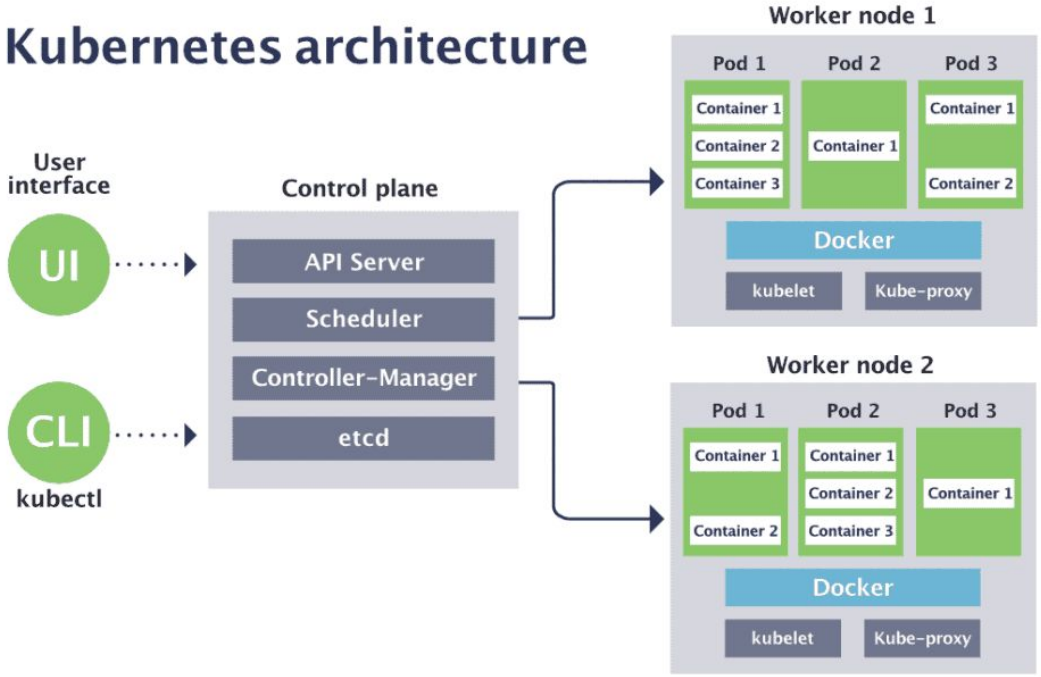
Cuando instalas Kubernetes, se encarga del procesamiento, las herramientas de redes y el almacenamiento en nombre de tus cargas de trabajo. Esto les permite a los desarrolladores enfocarse en las aplicaciones sin preocuparse por el entorno subyacente.

Supervisión del estado de los servicios

Kubernetes ejecuta verificaciones de estado de manera continua en los servicios, reinicia los contenedores con errores o que se detuvieron y solo pone los servicios a disposición de los usuarios una vez confirma que se están ejecutando.

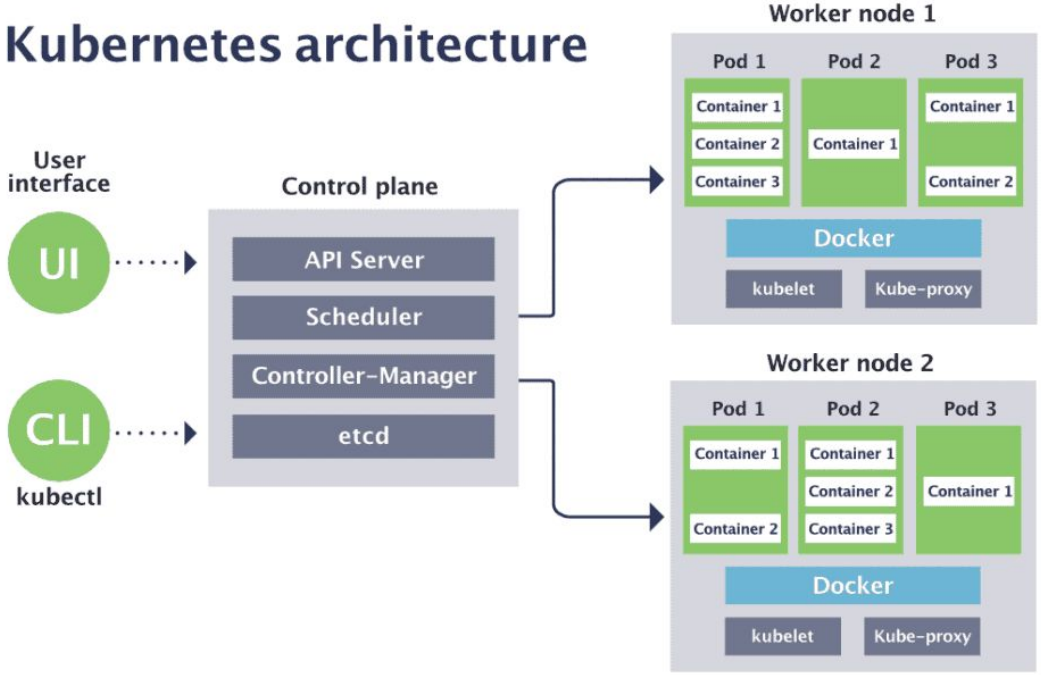
Arquitectura K8s

Arquitectura K8s



API Server: Interfaz para conectarse con el mundo exterior
Scheduler: Decide en donde va a correr los contenedores (en qué nodos)
Controller-Manager: se encargan del despliegue de la app, la forma de escalar, definen como pasar entre versiones (v1 a V2) (Deployment, DaemonSets, StatefulSets, Jobs, Cron)
etcd: guarda el state del cluster, y las configuraciones del cluster, funciona como un tipo de bd centralizada.

Arquitectura K8s



kubelet: es aquel que se comunica con el API Server y se encarga de correr el runtime (Docker)
Docker: es el elemento que utiliza Kubernetes para la gestión de los contenedores.
kube-proxy: Administra las comunicaciones de red dentro y fuera del cluster.
Pod: Representa una instancia única de un proceso en ejecución en un cluster.

Video: https://www.youtube.com/watch?v=cvziZliY-AI

LAB 3

Objetivo: Hacer el despliegue de una app en kubernetes local con microkube y mirar funcionamiento.