



¿Qué es 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. El soporte, las herramientas y los servicios para Kubernetes están ampliamente disponibles.

Google liberó el proyecto Kubernetes en el año 2014. Kubernetes se basa en la experiencia de Google corriendo aplicaciones en producción a gran escala por década y media, junto a las mejores ideas y prácticas de la comunidad.

Kubernetes ofrece un entorno de administración centrado en contenedores. Kubernetes orquesta la infraestructura de cómputo, redes y almacenamiento para que las cargas de trabajo de los usuarios no tengan que hacerlo. Esto ofrece la simplicidad de las Plataformas como Servicio (PaaS) con la flexibilidad de la Infraestructura como Servicio (IaaS) y permite la portabilidad entre proveedores de infraestructura.

¿Qué hace de Kubernetes una plataforma?

A pesar de que Kubernetes ya ofrece muchas funcionalidades, siempre hay nuevos escenarios que se benefician de nuevas características. Los flujos de trabajo de las aplicaciones pueden optimizarse para acelerar el tiempo de desarrollo. Una solución de orquestación propia puede ser suficiente al principio, pero suele requerir una automatización robusta cuando necesita escalar. Es por ello que Kubernetes fue diseñada como una plataforma: para poder construir un ecosistema de componentes y herramientas que hacen más fácil el desplegar, escalar y administrar aplicaciones.

Las etiquetas, o Labels, le permiten a los usuarios organizar sus recursos como deseen. Las anotaciones, o Annotations, les permiten asignar información arbitraria a un recurso para facilitar sus flujos de trabajo y hacer más fácil a las herramientas administrativas inspeccionar el estado.

Además, el Plano de Control de Kubernetes usa las mismas APIs que usan los desarrolladores y usuarios finales.

Los usuarios pueden escribir sus propios controladores, como por ejemplo un planificador o scheduler, usando sus propias APIs desde una herramienta de línea de comandos.

Este diseño ha permitido que otros sistemas sean construidos sobre Kubernetes.

Lo que Kubernetes no es

Kubernetes no es una Plataforma como Servicio (PaaS) convencional. Ya que Kubernetes opera a nivel del contenedor y no a nivel del hardware, ofrece algunas características que las PaaS también ofrecen, como deployments, escalado, balanceo de carga, registros y monitoreo. Dicho esto, Kubernetes no es monolítico y las soluciones que se ofrecen de forma predeterminada son opcionales e intercambiables.

Kubernetes ofrece los elementos esenciales para construir una plataforma para desarrolladores, preservando la elección del usuario y la flexibilidad en las partes más importantes.

Entonces, podemos decir que Kubernetes:

No limita el tipo de aplicaciones que soporta. Kubernetes busca dar soporte a un número diverso de cargas de trabajo, que incluyen aplicaciones con y sin estado así como aplicaciones que procesan datos. Si la aplicación puede correr en un contenedor, debería correr bien en Kubernetes.

No hace deployment de código fuente ni compila tu aplicación. Los flujos de integración, entrega y deployment continuo (CI/CD) vienen determinados por la cultura y preferencia organizacional y sus requerimientos técnicos.

No provee servicios en capa de aplicación como middleware (por ejemplo, buses de mensaje), frameworks de procesamiento de datos (como Spark), bases de datos (como MySQL), caches o sistemas de almacenamiento (como Ceph). Es posible correr estas aplicaciones en Kubernetes, o acceder a ellos desde una aplicación usando un mecanismo portable como el Open Service Broker.

No dictamina las soluciones de registros, monitoreo o alerta que se deben usar. Hay algunas integraciones que se ofrecen como prueba de concepto, y existen mecanismos para recolectar y exportar métricas.

No provee ni obliga a usar un sistema o lenguaje de configuración (como jsonnet) sino que ofrece una API declarativa que puede ser usada con cualquier forma de especificación declarativa

No provee ni adopta un sistema exhaustivo de mantenimiento, administración o corrección automática de errores.

Además, Kubernetes no es un mero sistema de orquestación. De hecho, Kubernetes elimina la necesidad de orquestar. Orquestación se define como la ejecución de un flujo de trabajo definido: haz A, luego B y entonces C. Kubernetes está compuesto de un conjunto de procesos de control independientes y combinables entre si que llevan el estado actual hacia el estado deseado. No debería importar demasiado como llegar de A a C. No se requiere control centralizado y, como resultado, el sistema es más fácil de usar, más poderoso, robusto, resiliente y extensible.

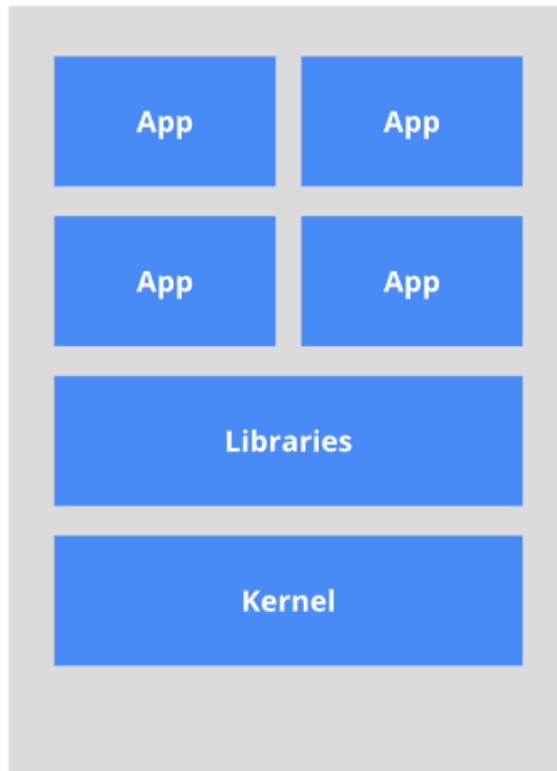
¿Por qué usar contenedores?

La **Manera Antigua** de desplegar aplicaciones era instalarlas en un servidor usando el administrador de paquetes del sistema operativo. La desventaja era que los ejecutables, la configuración, las librerías y el ciclo de vida de todos estos componentes se entretrejan unos a otros. Podíamos construir imágenes de máquina virtual inmutables para tener rollouts y rollbacks predecibles, pero las máquinas virtuales son pesadas y poco portables.

La **Manera Nueva** es desplegar contenedores basados en virtualización a nivel del sistema operativo, en vez del hardware. Estos contenedores están aislados entre ellos y con el servidor anfitrión: tienen sus propios sistemas de archivos, no ven los procesos de los demás y el uso de recursos puede ser limitado. Son más fáciles de construir que una máquina virtual, y porque no están acoplados a la infraestructura y sistema de archivos del anfitrión, pueden llevarse entre nubes y distribuciones de sistema operativo.

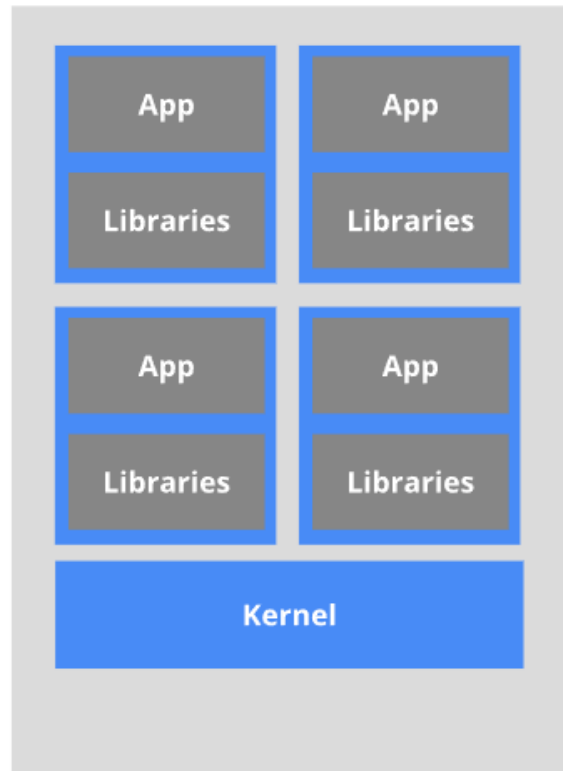
Ya que los contenedores son pequeños y rápidos, una aplicación puede ser empaquetada en una imagen de contenedor. Esta relación uno a uno entre aplicación e imagen nos abre un abanico de beneficios para usar contenedores. Con contenedores, podemos crear imágenes inmutables al momento de la compilación en vez del despliegue ya que las aplicaciones no necesitan componerse junto al resto del stack ni atarse al entorno de infraestructura de producción. Generar una imagen de contenedor al momento de la compilación permite tener un entorno consistente que va desde desarrollo hasta producción. De igual forma, los contenedores son más transparentes que las máquinas virtuales y eso hace que el monitoreo y la administración sean más fáciles. Esto se aprecia más cuando los ciclos de vida de los contenedores son administrados por la infraestructura en vez de un proceso supervisor escondido en el contenedor. Por último, ya que solo hay una aplicación por contenedor, administrar el despliegue de la aplicación se reduce a administrar el contenedor.

The old way: Applications on host



*Heavyweight, non-portable
Relies on OS package manager*

The new way: Deploy containers



*Small and fast, portable
Uses OS-level virtualization*

En resumen, los beneficios de usar contenedores incluyen:

Ágil creación y despliegue de aplicaciones: Mayor facilidad y eficiencia al crear imágenes de contenedor en vez de máquinas virtuales

Desarrollo, integración y despliegue continuo: Permite que la imagen de contenedor se construya y despliegue de forma frecuente y confiable, facilitando los rollbacks pues la imagen es inmutable

Separación de tareas entre Dev y Ops: Puedes crear imágenes de contenedor al momento de compilar y no al desplegar, desacoplando la aplicación de la infraestructura

Observabilidad No solamente se presenta la información y métricas del sistema operativo, sino la salud de la aplicación y otras señales

Consistencia entre los entornos de desarrollo, pruebas y producción: La aplicación funciona igual en un laptop y en la nube.

Portabilidad entre nubes y distribuciones: Funciona en Ubuntu, RHEL, CoreOS, tu datacenter físico, Google Kubernetes Engine y todo lo demás
Administración centrada en la aplicación: Eleva el nivel de abstracción del sistema operativo y el hardware virtualizado a la aplicación que funciona en un sistema con recursos lógicos.

Microservicios distribuidos, elásticos, liberados y débilmente

acoplados: Las aplicaciones se separan en piezas pequeñas e independientes que pueden ser desplegadas y administradas de forma dinámica, y no como una aplicación monolítica que opera en una sola máquina de gran capacidad.

Aislamiento de recursos: Hace el rendimiento de la aplicación más predecible.

Utilización de recursos: Permite mayor eficiencia y densidad.

¿Qué significa Kubernetes? ¿Qué significa K8S?

El nombre Kubernetes proviene del griego y significa timonel o piloto. Es la raíz de gobernador y de cibernética. K8s es una abreviación que se obtiene al reemplazar las ocho letras "ubernete" con el número 8.

Fuente: Kubernetes ,Última modificación Jul 17, 2022

Sitio Web: <https://kubernetes.io/es>



ISPC : instituto Superior Politécnico Córdoba.

Materia : Proyecto Integrador I "Capa de almacenamiento".

Grupo: Plata.

Profesor: Gonzalo Vera.