

Ramón Millán

Master Principal Sales Consultant en Oracle.

La importancia de ‘cloud native’ en las operadoras

El nuevo paradigma ‘cloud native’ se ha puesto de moda en las telecomunicaciones con la llegada del 5G. Se trata de un paso más allá de NFV (*Network Functions Virtualization*), para crear una infraestructura de red más ágil y escalable.

Estas tecnologías nativas en la nube llevan ya varios años siendo utilizadas exitosamente por las OTT (Facebook, Netflix, Twitter, etc.) y compañías ofreciendo aplicaciones SaaS en la nube pública (Amazon, Google, Microsoft, Oracle, etc.). Su grado de despliegue en las redes de las operadoras es aún muy pequeño, pero se trata de un proceso imparable debido a la mayor eficiencia que ofrece en términos de CAPEX y OPEX, así como la mayor rapidez para ofrecer nuevos servicios, algo imprescindible en un sector que requiere de grandes inversiones y con dificultades para crecer en ingresos.

“Cloud native” se basa en cuatro componentes clave interrelacionados: contenedores (*containers*), microservicios (*microservices*), DevOps (*Development and Operations*) y CI/CD (*Continuous Integration and Continuous Delivery*). La CNCF (*Cloud Native Computing Foundation*) busca impulsar la adopción de “cloud native” mediante un ecosistema de herramientas de código abierto e independientes de vendedores específicos, lo cual no sólo supone ventajas económicas, también facilita la interoperabilidad, sin impedir la posibilidad de realizar adaptaciones.

El **contenedor** es la herramienta más importante para poder ofrecer aplicaciones en la nube moderna. Un contenedor es, básicamente, una forma de empaquetar el *software*, que abarca todo lo necesario para que una aplicación se ejecute encapsulada dentro de una sola imagen. El contenedor no incluye el sistema operativo, pero sí todos los ficheros de configuración, librerías y código de las que depende la aplicación, siendo así completamente independiente del servidor o dispositivo donde se ejecute. El contenedor será luego ejecutado en un sistema operativo que puede ser compartido por otros contenedores.

Predecible, repetible e inmutable

De este modo, el contenedor puede ser visto en cuanto a portabilidad como una máquina virtual, pero sin la sobrecarga de tener que enlazar también el sistema operativo completo. Esto hace que sean más ligeros y ágiles que las máquinas virtuales, consumiendo menos recursos *hardware* y con unos tiempos de inicialización sensiblemente menores. Cuando se ejecuta un contenedor, se sabe exactamente como lo hará, es decir, es predecible, repetible e inmutable. No aparecen errores inesperados cuando se mueve a una nueva máquina o un nuevo entorno.

Docker y *Kubernetes* son dos de las tecnologías de código abierto más conocidas para permitir la ejecución en contenedores. *Docker* permite construir, transferir, desplegar y ejecutar aplicaciones *software* dentro de contenedores de una manera muy sencilla y confiable. Independientemente de dónde se esté ejecutando la imagen, se comportará del mismo modo. *Kubernetes* interacciona con el motor de *Docker*, encargándose de la orquestación automática de contenedores, permitiendo que un gran número de contenedores trabajen conjuntamente en armonía, reduciendo la carga operativa.

Se encarga de funciones tales como el re-arranque automático, auto-replicación, auto-escalado, etc., de la imagen del contenedor en múltiples máquinas, algo totalmente imprescindible en el caso de tener microservicios. *Kubernetes* se encarga de gestionar automáticamente todo el grupo de servidores, distribuyendo los contenedores según los recursos disponibles, además de crear, ejecutar, vigilar, medir, destruir y relanzar los contenedores, debe mantener y controlar en todo momento cada aspecto relevante de los contenedores y su estado.

‘Cloud native’ son un conjunto de tecnologías que permiten crear, desplegar y ejecutar aplicaciones, explotando las ventajas de la computación en la nube

Los microservicios

Los contenedores se integran bien con las arquitecturas basadas en **microservicios**. El *software* en las operadoras de telecomunicaciones, incluso ya virtualizado, se ha basado tradicionalmente en aplicaciones monolíticas. Aunque lo ideal es desarrollar las aplicaciones desde un inicio basadas en microservicios, también es posible migrar las aplicaciones antiguas en distintas fases, algo que ya ha sido exitosamente realizado por las OTT.

Los microservicios buscan fragmentar las aplicaciones monolíticas, de gran tamaño y complejidad, por una colección de servicios más pequeños, con su propia lógica, base de datos, etc. Así, la aplicación es desarrollada como un conjunto de pequeños servicios, cada uno ejecutando sus propios procesos sobre distintos contenedores y que se comunican entre sí utilizando API (*Application Programming Interfaces*). Las aplicaciones que utilizan microservicios son más escalables, robustas y eficientes, más rápidas de desarrollar y desplegar y más fáciles de entender y mantener.

Por otro lado, **DevOps** hace referencia a la continua y estrecha comunicación, colaboración e integración, entre los desarrolladores de *software* y los grupos de operaciones, para crear aplicaciones de alta calidad que cumplan las necesidades de los clientes. Se trata de un conjunto de principios y herramientas que siguen el nuevo paradigma de desarrollo y gestión Agile, facilitando la creación de una cultura y entorno, donde el desarrollo, prueba y actualización de aplicaciones se haga de forma rápida, frecuente y sin errores.

Finalmente, **CI/CD** está estrechamente relacionado con DevOps. La integración continua trata de asegurar que todos los programadores utilizan la última versión

del código disponible para el desarrollo de nueva funcionalidad, así como que los fallos son corregidos rápidamente. De este modo, se mantiene siempre la versión de software más reciente estable y lista para ser desplegada en cualquier momento. El despliegue continuo trata

de automatizar el proceso de entrega de *software* mediante un conjunto de pruebas automatizadas, que una vez pasadas sin errores, permitirán la instalación en producción. Entre las herramientas CI/CD más populares están GitLab, Jenkins y Drone. ■

Docker y Kubernetes son dos tecnologías de código abierto que, utilizadas de forma conjunta, ayudan a que las aplicaciones sean ejecutadas en contenedores

