

MICROSERVICIOS

Vas a investigar qué otras compañías, además de Netflix, utilizan hoy en día una arquitectura basada en microservicios.

POR: CARRION HUACANI GEAN CARLO

Amazon

No soporta tantos dispositivos como Netflix, pero tampoco es que sea fundamental para cubrir su sector. Migró hace tres años a la arquitectura de microservicios siendo una de las primeras grandes compañías que la implementaban en producción. No hay cifra aproximada de la cantidad de solicitudes que pueden recibir a diario, pero no son pocas. Entre éstas encontramos multitud de aplicaciones, las API del servicio web que ofrecen o la propia web de Amazon, cuyos ingenieros reconocen que habría sido imposible sobre la arquitectura monolítica con la que trabajaban previamente.



Ebay

Una de las empresas con mayor visión de futuro, siendo pionera en la adopción de tecnologías como Docker o ésta que nos ocupa. Su aplicación principal comprende varios servicios autónomos, y cada uno ejecutará la lógica propia de cada área funcional que se ofrece a los clientes.

eBay es otro buen ejemplo de eficiencia de microservicios. La aplicación principal de la compañía utiliza varios microservicios que deben poder procesar millones de solicitudes por día.

Según el CTO Steven Fisher, eBay se basa en 1000 servicios que deben ejecutarse dentro del sistema. Mediante el uso de microservicios, eBay puede agregar nuevas funcionalidades más fácilmente, asignar nuevos trabajos de desarrollo y eliminar las dependencias innecesarias.



PayPal

PayPal ahora está construyendo una puerta de enlace de gráficos única y uniforme utilizando Apollo Federation. Netflix también implementó previamente una solución similar. Como parte de este esfuerzo, algunos equipos de desarrollo de backend en PayPal decidieron utilizar GraphQL para exponer su API debido a las ventajas de productividad percibidas. Netflix también comparte esta tendencia, donde los ingenieros adoptaron los microservicios GraphQL para el desarrollo rápido de aplicaciones.



Características convierten a un servicio en un Microservicios

- **Componentización a través de Servicios:** Las arquitecturas de microservicios utilizarán bibliotecas, pero su forma principal de dividir en componentes su propio software es dividirlos en servicios
- **Organizado en torno a las capacidades comerciales:** Cuando se busca dividir una aplicación grande en partes, a menudo la administración se enfoca en la capa de tecnología, lo que genera equipos de interfaz de usuario, equipos de lógica del lado del servidor y equipos de base de datos.
- **Productos no Proyectos:** La mayoría de los esfuerzos de desarrollo de aplicaciones que vemos utilizan un modelo de proyecto: donde el objetivo es entregar alguna pieza de software que luego se considera completa. Una vez finalizado, el software se entrega a una organización de mantenimiento y el equipo del proyecto que lo creó se disuelve.
- **Terminales inteligentes y canalizaciones tontas:** Los equipos de microservicios utilizan los principios y protocolos sobre los que se basa la red mundial (y, en gran medida, Unix). Los recursos de uso frecuente se pueden almacenar en caché con muy poco esfuerzo por parte de los desarrolladores o la gente de operaciones.

Características convierten a un servicio en un Microservicios

- **Gobernanza descentralizada:** Al dividir los componentes del monolito en servicios, tenemos una opción al construir cada uno de ellos
- **Gestión de datos descentralizada:** La descentralización de la gestión de datos se presenta de varias maneras diferentes. En el nivel más abstracto, significa que el modelo conceptual del mundo diferirá entre sistemas. Este es un problema común cuando se integra en una gran empresa, la vista de ventas de un cliente diferirá de la vista de soporte
- **Automatización de Infraestructura:** Muchos de los productos o sistemas que se construyen con microservicios están siendo desarrollados por equipos con amplia experiencia en Entrega continua y su precursor, Integración continua
- **Diseño para el fracaso:** Cualquier llamada de servicio podría fallar debido a la falta de disponibilidad del proveedor, el cliente debe responder a esto de la mejor manera posible
- **Diseño evolutivo:** Los profesionales de microservicios, por lo general, provienen de un fondo de diseño evolutivo y ven la descomposición del servicio como una herramienta adicional para permitir que los desarrolladores de aplicaciones controlen los cambios en su aplicación sin ralentizar el cambio