

INFRAESTRUCTURA COMO CÓDIGO

¿Qué es?

La infraestructura como código (IaC), es una práctica que utiliza principios de desarrollo de software para automatizar los procesos de crear la infraestructura. La Infraestructura como Código automatiza el aprovisionamiento y la administración de la infraestructura.

Principios

- **Los sistemas se pueden reproducir fácilmente:** Al definir la infraestructura de un proceso, la segunda vez que se quiera realizar ese mismo proceso ya está definido en el código y es más rápido de realizar.
- **Los sistemas son desechables:** Si un servidor falla, se puede desechar y volver a construir porque ya está definida cuál es la estructura que debería tener ese servidor.
- **Los sistemas son consistentes:** Se cuenta con un archivo donde encuentras que debería de tener el servidor y lo expuesto en ese archivo debería de estar reflejado en el propio servidor.
- **Los sistemas son repetibles:** Se puede escribir una definición de código una única vez y utilizarla N veces.
- **El diseño siempre está cambiando:** Es más fácil cuando se tiene el diseño plasmado en un archivo de definición porque ya se saben los cambios que hay que hacer.

Por qué es importante

La capacidad de tratar la infraestructura como código aporta los beneficios de la automatización, la visibilidad, la eficiencia y la escalabilidad a la gestión de la infraestructura, las aplicaciones pueden implementarse de forma más rápida y fiable, con menos esfuerzo manual y riesgo de errores humanos o vulnerabilidades de seguridad.

Prácticas generales

- Utilizar archivos de definición, todas las herramientas de infraestructura como código tienen un formato propio para definir infraestructura.
- Sistemas y procesos autodocumentados, la IaC nos permite reutilizarlo.
- Hacer versiones del código, lo que nos permite guardar todos los cambios que hacemos, si tenemos un fallo en un punto, podemos volver atrás en el tiempo para solucionarlo.
- Preferir cambios pequeños para que el cambio no impacte tanto.
- Mantener los servicios continuamente disponibles

Beneficios/ventajas

- **Velocidad:** Al crear un archivo donde almacenamos todas las configuraciones se vuelve repetible, lo que nos deja la infraestructura preparada para otros trabajos.
- **Menor coste:** La creación de infraestructura conlleva una reducción de gastos, no hace falta trabajadores que configuren manualmente y mantengan la infraestructura.
- **Homogeneidad:** Al tener los archivos ya predefinidos la infraestructura como código puede mejorar la homogeneidad y reducir los errores que a menudo se producen durante la configuración manual, dando lugar a la consistencia
- **Trazabilidad:** La infraestructura como código promueve el control de versiones, de modo que tienen una trazabilidad completa de los cambios de cada configuración.

- **Fácil testeo:** La infraestructura como código permite a los equipos probar las aplicaciones en entornos de producción al principio del ciclo de desarrollo.

Métodos y herramientas

Las **herramientas y marcos de trabajo especiales de IaC** ofrecen sus propios lenguajes de configuración, eliminan la necesidad de tener que aprender las diferentes interfaces de cada API. Hay distintos tipos de herramientas para la implementación de infraestructura

- **Herramientas para la definición de infraestructura:** son herramientas que nos permiten generar código para la IaC. Terraform, AWS CloudFormation.
- **Herramientas para la configuración de servidores:** nos permiten configurar los servidores con el estado que deseamos. Chef, Puppet.

Diferencias con otros métodos anteriores

La principal diferencia con otros métodos es que, en el entorno clásico no virtualizado, todos los recursos están siempre conectados al hardware físico, lo que conlleva una infraestructura poco flexible. Con la virtualización del hardware la situación cambia, gracias a esto se evita que los clientes tengan que disponer de un hardware especial para ello, por un lado, garantiza una mayor fiabilidad y por el otro también es más fácil añadir nuevos servicios. Los entornos definidos por software, se caracterizan por el hecho de que el sistema de control se abstrae de los componentes hardware y se implementa en un software de control central. Proveedores y clientes se benefician de un mayor rendimiento de hardware ya que este no procesa datos. La infraestructura utiliza todas estas técnicas mencionadas para automatizar la administración por software y maximizar el potencial de la nube.

Infraestructura como código mutable vs. inmutable

La infraestructura como código se convierte en el nuevo estándar para las buenas prácticas, y con ello las infraestructuras pasan de los modos de funcionamiento tradicionales a otros más inmutables porque se quiere alcanzar la entrega continua, integrando la publicación de versiones y las pruebas automatizadas.

- **Infraestructura como código mutable:** Debe actualizarse continuamente para poder seguir respondiendo a la evolución de las necesidades empresariales. Los profesionales informáticos deben abordar cada servidor y conmutador individualmente, lo que requiere largas horas dedicadas a identificar problemas y generar soluciones.
- **Infraestructura como código inmutable:** Representa un futuro simplificado, que requiere que cada componente siga unas especificaciones exactas sin variaciones. Cuando se introduce un cambio necesario, la infraestructura como código antigua se sustituye por una nueva que se adapta a los nuevos requisitos. Gracias a ello, se pueden diseñar e implementar aplicaciones de forma mucho más rápida y estable.

Enfoques declarativo e imperativo

La infraestructura como código tiene dos enfoques:

- **Enfoque declarativo:** Define el estado deseado del sistema, guarda en una lista el estado actual de los objetos de su sistema facilitando el desmontaje de la infraestructura y se prepara la infraestructura deseada de manera automática. Si hay cambios en ese estado, la herramienta declarativa los aplicará automáticamente.
- **Enfoque imperativo:** Define los comandos específicos en un orden concreto para lograr la configuración deseada. Hay que resolver de manera manual cómo se debe aplicar.

Bibliografía

[Infrastructure as code - automation is not enough](#), Kief Morris

[Meet Infrastructure as Code](#), Chris Riley

[Infrastructure as Software](#), Kris Növa

[Curso de Infraestructura Como Código con Terraform](#), Yolanda López

[¿Qué es la infraestructura como código?](#), f5 glossary

[¿Qué es la infraestructura como código?](#), hpe.com

[¿Qué es la infraestructura como código \(IaC\)?](#), redhat.com

[Infraestructura Como Código: Beneficios Y Herramientas](#), Ekaterina Novoseltseva

[Infraestructura como Código: Qué es y herramientas](#), Frankier Flores

[nfrastructure as code \(IaC\): gestionar la infraestructura de TI como código](#), ionos.es