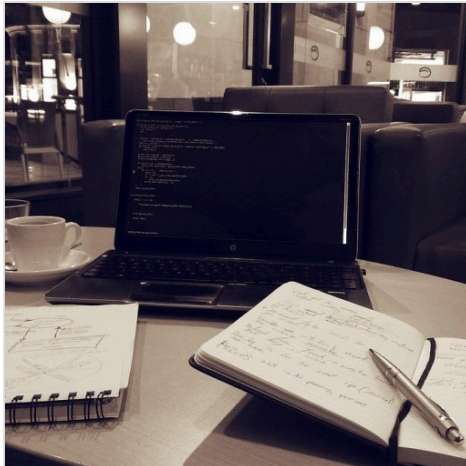


L U M B R E



Infraestructura como código

Grado: 8

Materia: DevOps para Datos · Ingeniería de Datos

11/10/2026

Antes de empezar

La infraestructura configurada a mano se pierde y no se reproduce. Tratarla como código —archivos versionados que declaran servidores, redes y bases de datos— permite levantar, duplicar y deshacer entornos enteros con un comando.

QUÉ APRENDERÁS

- Gestionar servidores, redes y recursos en la nube mediante definiciones declarativas en código, reproducibles y versionadas.

Infraestructura como código

Infraestructura que se lee y se versiona

Tratar la infraestructura como código es tratarla como software: se escribe, se revisa, se versiona y se aplica. Un entorno ya no es una máquina que alguien tocó una vez, sino un archivo reproducible.

VIDEO

¿Qué es IaC? Infraestructura como código explicado

Cómo declarar servidores, redes y recursos en archivos versionados en lugar de configurarlos a mano.



Escanea para verlo en Lumbre

https://www.youtube.com/watch?v=HtgNRu_TySw

Qué es IaC

La gestión de servidores, redes y recursos mediante archivos de configuración legibles por máquina, en lugar de paneles manuales. El código describe la infraestructura deseada.

El problema del clic

- Configurar a mano en una consola no es repetible.
- Nadie sabe muy bien qué se tocó ni en qué orden.
- Un entorno de producción diverge del de pruebas.

Declarativo frente a imperativo

Declarativo: describes qué quieres (una red, dos máquinas). Imperativo: detallas cada paso para lograrlo. El declarativo deja al motor resolver el cómo.

Idempotencia

Aplicar la misma definición muchas veces produce el mismo resultado, sin duplicar recursos. Es la propiedad que hace seguro reejecutar el código.

Herramientas

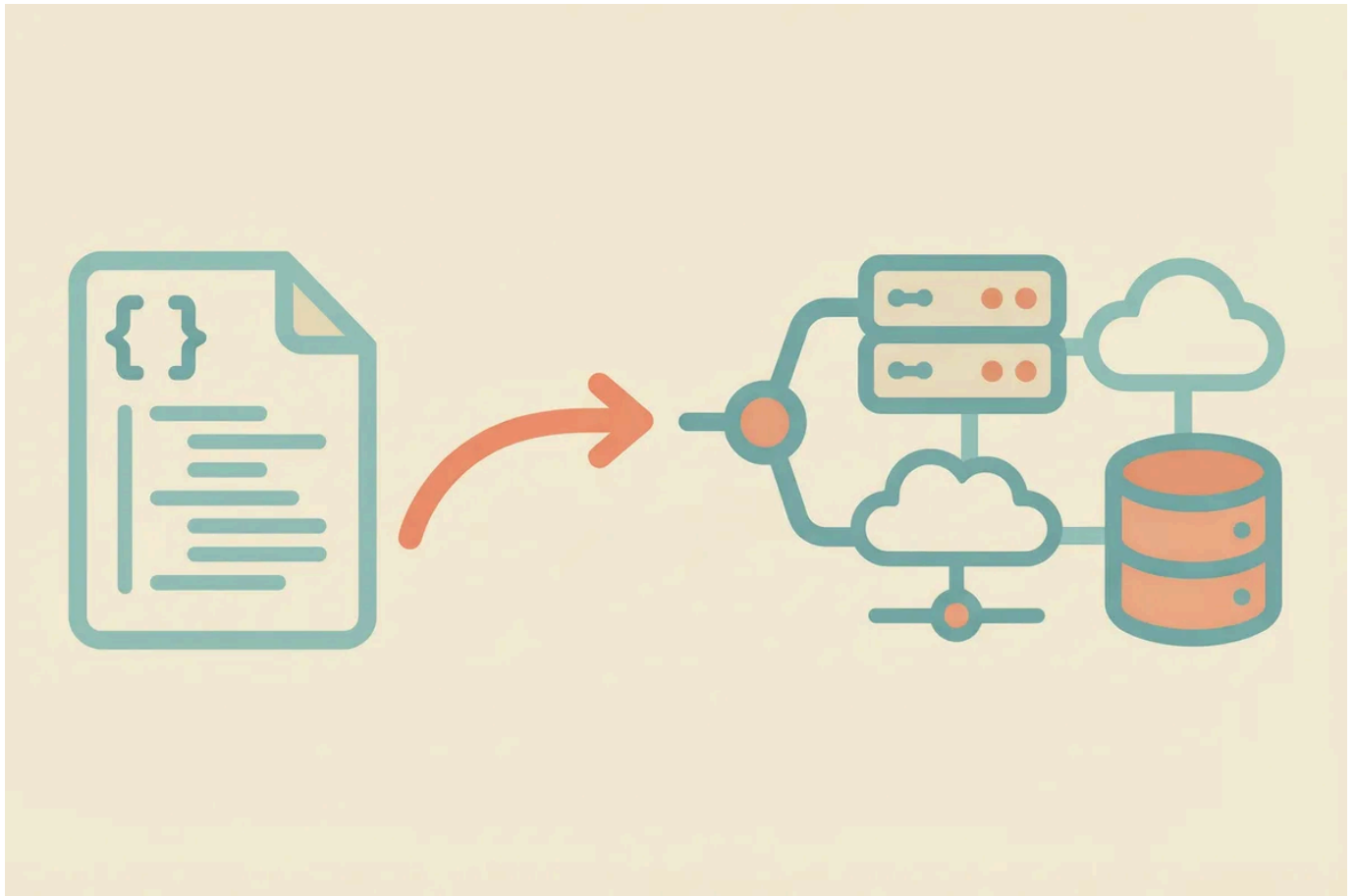
- Terraform y su lenguaje declarativo para aprovisionar nube.
- Ansible y similares para configurar y configurar máquinas.
- Plantillas nativas de cada proveedor en la nube.

Versionado y revisión

Al vivir en un repositorio, los cambios de infraestructura se revisan como cualquier código, con historial y aprobaciones antes de aplicarse.

Ambientes reproducibles

- Levantar un entorno nuevo ejecutando el mismo código.
- Duplicar producción para pruebas sin copiar a mano.
- Destruir lo que ya no hace falta.



La infraestructura se escribe como código versionado: el archivo declarativo levanta, duplica y destruye entornos reales.

Gestión del estado

Las herramientas rastrean el estado real creado y lo comparan con el código. Ese archivo de estado es delicado: debe compartirse y protegerse.

GitOps

Tratar un repositorio git como única fuente de verdad: lo que está desplegado debe coincidir con el código, y un cambio se aprueba con una solicitud de cambio.

Seguridad y secretos

- No meter credenciales en el código.
- Usar almacenes de secretos cifrados.
- El código describe la forma, no las llaves.

Deriva de configuración

Si alguien cambia la consola sin tocar el código, la realidad se aleja de lo declarado. Detectar y reconciliar ese deriva es clave.

Errores frecuentes

- Configurar a mano en la consola y alejar la realidad de lo declarado en el código.
- Meter credenciales y secretos dentro de los archivos de infraestructura.
- No compartir ni proteger el archivo de estado, y corromperlo entre varios.

Ejemplo resuelto: duplicar producción para pruebas

Un equipo necesita un entorno idéntico a producción para unas pruebas. Como toda la infraestructura vive en archivos declarativos versionados, basta clonar la definición, cambiar un parámetro de nombre y aplicar: se levantan la misma red, las mismas máquinas y la misma base, y al terminar se destruyen con un comando.

1. escribir la definición declarativa de la infraestructura
2. revisar y aprobar el cambio en el repositorio
3. planificar para simular qué creará o modificará
4. aplicar para alcanzar el estado descrito
5. destruir el entorno cuando ya no hace falta

¿Para qué sirve en la realidad?

Equipos que gestionan decenas de entornos en la nube necesitan que levantar, duplicar o deshacer infraestructura sea un archivo reproducible y revisable, no una máquina que alguien tocó una vez a mano.

1. La idempotencia en la infraestructura como código garantiza que:

- ☐ el código se ejecute más rápido
- ☐ aplicar la misma definición repetidas veces no duplica ni altera recursos
- ☐ no se necesite versionar
- ☐ la infraestructura crezca en cada ejecución

2. Con infraestructura como código, un entorno puede destruirse y volver a levantarse idéntico ejecutando el mismo código.

- ☐ Verdadero
- ☐ Falso

ACTIVIDAD INTERACTIVA DISPONIBLE EN LUMBRE

3. IaC



4. Ordena el flujo de trabajo de IaC:

Elementos:

- editar la definición declarativa
- revisar y aprobar el cambio
- simular qué creará o modificará
- aplicar para alcanzar el estado descrito

Escribir	· · ·
Revisar	· · ·
Plan	· · ·
Aplicar	· · ·

Infraestructura como código https://en.wikipedia.org/wiki/Infrastructure_as_code Gestión de infraestructura mediante definiciones en código.

5. Une cada término de IaC con su definición:

5.	declarativo	-----
5.	idempotencia	-----
5.	GitOps	-----
5.	deriva de configuración	-----

Opciones: describir el qué sin detallar el cómo · mismo resultado al aplicar varias veces · repositorio como única fuente de verdad · la realidad se aleja del código

IaC

Infraestructura como código

Enfoque

declarativo

idempotente

reproducible

Herramientas

Terraform

Ansible

plantillas de nube

Prácticas

versionar

revisar

GitOps

Cuidados

secretos cifrados

estado compartido

deriva de configuración

El mapa mental interactivo es interactivo en la versión online.



GitOps <https://en.wikipedia.org/wiki/GitOps> Usar un repositorio git como fuente de verdad del despliegue.

6. Alguien amplió un disco desde la consola de nube y nadie lo sabe. ¿Qué práctica de IaC impediría ese deriva y cómo lo detectarías?

Referencia curricular

Grado:	8
Materia:	DevOps para Datos · Ingeniería de Datos

Versión interactiva completa (con actividades, videos y autoevaluación):



<https://univ.lumbre.lat/object/cdia-dvops-004-infraestructura-como-codigo>

Material educativo generado desde Lumbre · lumbre.lat