

Pruebas para IaC

Aunque se utilice un sistema de control de versiones y automatización los cambios pueden fallar, las pruebas son esenciales para detectar errores antes de llegar a producción, verificar que el sistema funcione como lo esperado y documentar cómo debe comportarse la infraestructura.

Pruebas manuales Vs Pruebas automatizadas

Manuales: consumen mucho tiempo y no son escalables, a medida que la red crece es inviable verificar cada conexión, dejando margen para errores ocultos.

Automatizadas: Requieren un esfuerzo inicial de scripting, pero reducen el costo de mantenimiento y el tiempo de validación a largo plazo, permitiendo ejecuciones en paralelo.

Ciclo de Pruebas: Estático y Dinámico

Estático (Pre-despliegue):

- analiza el código sin crear recursos
- verifica sintaxis, normas de nombrado y dependencias
- rápido, seguro y facilita corrección temprana

Dinámico (Post-Despliegue):

- se ejecuta sobre recursos vivos
- verifica la funcionalidad real
- es una alerta temprana en el sistema activo

(conectividad de red y respuesta de puertos)

Entornos de Prueba: Para evitar riesgos en producción, se deben aislar cambios en entornos específicos.

Entornos de módulos: Aislados y efímeros, usados para crear y destruir recursos individuales rápidamente.

Entornos de staging (Persistentes): Replican la topología de producción casi idénticamente, sirve para validar cambios complejos y minimizar la diferencias entre entornos antes del lanzamiento final.

Dry-runs (Ejecuciones simuladas): se visualizan los cambios sin aplicarlos.

Pruebas de Contrato (Gestion de dependencias): asegura que diferentes módulos de infraestructura puedan hablar entre si correctamente sin romperse cuando uno de ellos cambia.

- El contrato es el acuerdo entre un Módulo Proveedor y un Módulo consumidor
- Nos permite actualizar modulos de bajo nivel sin miedo a romper los sistemas que dependen de ellos

Pruebas de integración (Análisis dinámico): prueban que la infraestructura realmente funciona en la nube, realiza:

1. Desplegar la infraestructura en un entorno de prueba
2. Verifica su estado real mediante una API del proveedor
3. Destruir recursos (limpieza)

Pruebas de extremo a extremo (E2E)

Una prueba de integración verifica que la API de la nube dice que el servidor esta running, pero una prueba E2E realiza una petición HTTP real a la URL para ver si la aplicación responde, su objetivo es validar el flujo completo del usuario y detectar fallos que la infraestructura por si sola no muestra como **problemas de permiso o errores de red.**

- Su implementación es costosa y lenta

Pruebas de humo (Smoke Test): es una variante rápida y crítica de E2E.

- "si al encenderlo sale humo, apágalo"
- se ejecutan antes de la batería completa de pruebas, si falla se detiene, ahorrando tiempo, valida funcionalidades como **se puede conectar a la db?**, **Responde el servidor web?**

Estrategia de la Pirámide de Pruebas

Base: Pruebas unitarias, gran cantidad, rápidas y baratas

Medio: Pruebas integración, validan que los modulos interactuen bien con la nube

Arriba: E2E, pocas, lentas, caras

NOTA: evitar solo pruebas manuales o la pirámide invertida



