

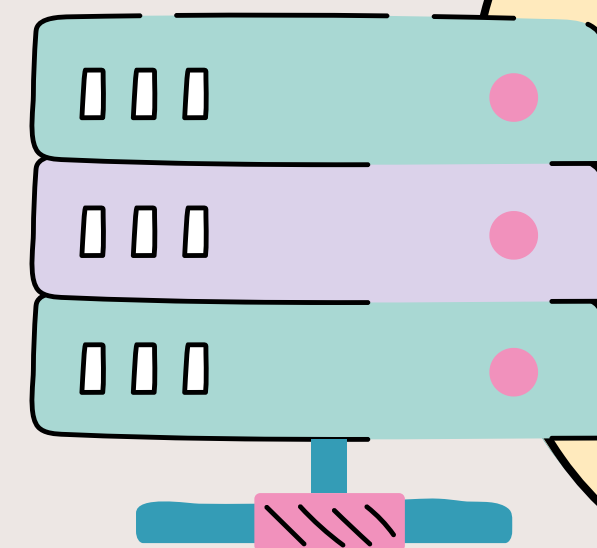


TERRAFORM

Automatiza tu infraestructura como código



Por Juan Duran



¿Qué es Terraform?

Terraform es una herramienta de **infraestructura como código** (IaC) que permite definir, configurar y gestionar recursos de infraestructura mediante **archivos declarativos**. Esto facilita la creación y mantenimiento de **entornos en la nube** y en **infraestructuras propias**, haciendo que el proceso sea repetible y menos propenso a errores humanos.

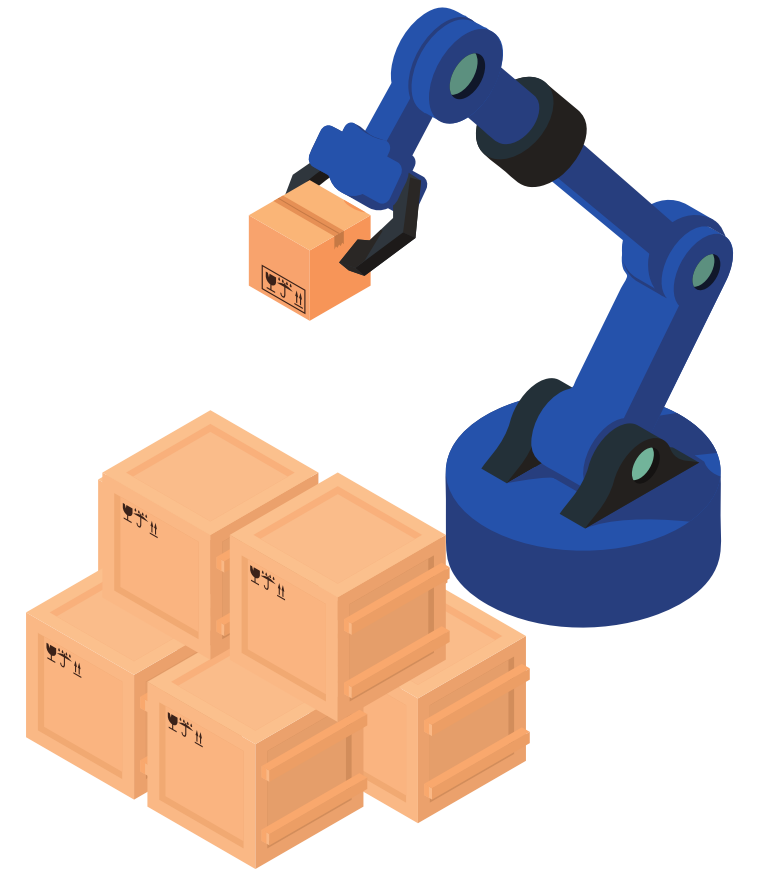
- Define **infraestructura** con archivos de configuración
- Soporta múltiples **proveedores** y **plataformas**
- Facilita **automatización** y **colaboración**



¿Por qué usar Terraform?

La **principal ventaja** de Terraform es la **automatización** completa del **ciclo de vida de la infraestructura**. Evita tareas manuales tediosas y reduce el riesgo de errores al tener todo versionado y controlado con código. Además, su soporte para múltiples proveedores permite gestionar distintos entornos con una única herramienta.

- **Automatiza** despliegues y cambios
- **Reduce** errores manuales
- Trabaja con muchas **plataformas** cloud y on-premise



Puntos Clave de Terraform

Terraform trabaja de **manera declarativa**, lo que significa que tú defines el **estado** deseado y Terraform se encarga de hacer los cambios necesarios para lograrlo. Antes de aplicar cualquier cambio, **genera** un **plan** para que puedas ver exactamente qué se va a modificar. También gestiona la **creación, actualización** y **eliminación** de recursos, manteniendo todo sincronizado.

- **Configuración declarativa**
- **Planificación** previa de cambios
- **Gestión completa** del ciclo de vida



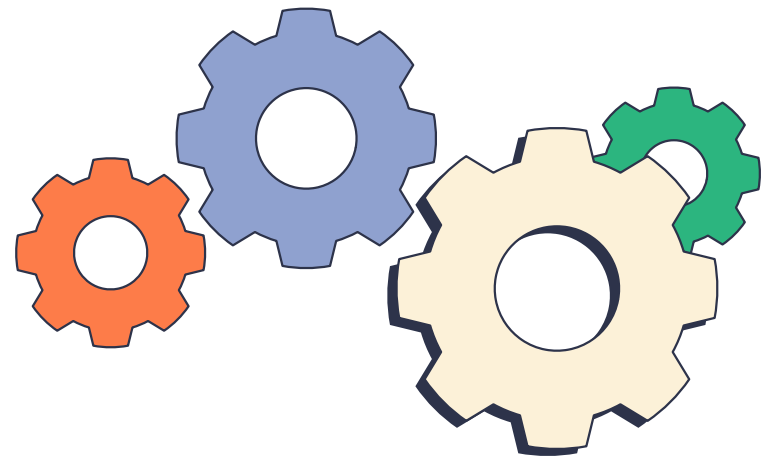
¿Cómo funciona Terraform?

El proceso **empieza** escribiendo **archivos de configuración** donde se describen los recursos que quieres. **Terraform compara** el estado actual de la infraestructura con el estado deseado que definiste, **calcula** las diferencias y **aplica** los cambios necesarios. Para ello, mantiene un archivo de estado que refleja el entorno actual.

- **Escribes** la **configuración**
- **Terraform compara** estados
- **Aplica cambios** automáticamente
- **Mantiene** un archivo de estado



Proveedores y recursos



Terraform es **compatible** con muchos **proveedores** como **AWS**, **Azure**, **Google Cloud**, **Kubernetes**, **bases de datos** y más. Esto permite gestionar toda tu infraestructura desde un solo lugar, facilitando entornos híbridos o multi-cloud sin complicaciones.

- Soporta **proveedores cloud** y plataformas diversas
- **Gestiona recursos** variados y complejos
- Ideal para **entornos híbridos y multi-cloud**

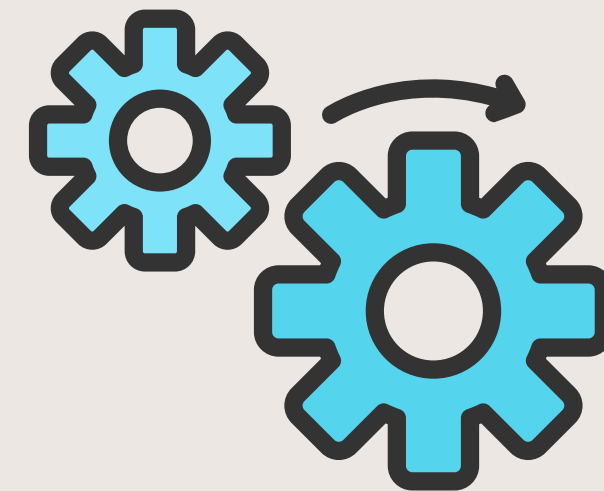


Lenguaje de configuración



El **lenguaje** de configuración de **Terraform** es **simple** y **legible**, permitiendo definir recursos y sus propiedades. Además, ofrece funcionalidades como variables, condicionales y módulos, que permiten **adaptar** las **configuraciones** a proyectos de cualquier tamaño y complejidad.

- **Fácil de leer y escribir**
- Uso de **variables** y **condicionales**
- **Modular y adaptable**



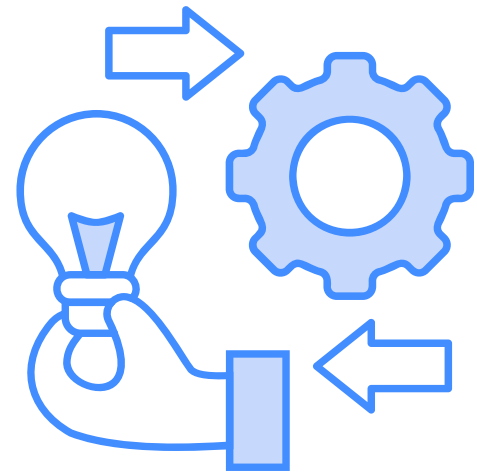
Estado y su importancia

El **archivo de estado es el núcleo de Terraform**, ya que registra el estado actual de los recursos gestionados. Esto permite **detectar cambios** y **coordinar actualizaciones**, especialmente en equipos donde varias personas trabajan sobre la misma infraestructura. Sin embargo, debe manejarse con cuidado por motivos de seguridad.

- **Guarda** el estado actual de recursos
- **Facilita** sincronización en equipos
- **Fundamental** para evitar conflictos
- **Requiere** manejo seguro



Módulos en Terraform



Los **módulos** son **bloques reutilizables** de configuración que ayudan a **organizar el código**, fomentan la **estandarización** y facilitan el **mantenimiento**. Usar módulos es una buena práctica que mejora la colaboración y permite escalar proyectos sin perder orden ni control.

- **Paquetes reutilizables** de configuración
- Mejoran **organización** y mantenimiento
- **Facilitan colaboración** en equipos



Casos de uso reales



Terraform es usado para **despliegues rápidos**, para gestionar miles de recursos, y por equipos **DevOps** para **automatizar pipelines y migraciones**. Es especialmente valioso para gestionar **clusters** de **Kubernetes** y entornos complejos con muchos servicios.

- **Despliegues ágiles** en startups
- **Gestión a gran escala** en corporaciones
- **Automatización DevOps** y pipelines
- **Gestión de Kubernetes** y migraciones

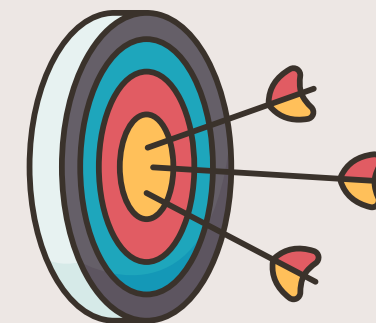


Retos comunes y buenas prácticas



Aunque **Terraform simplifica** la **infraestructura**, es importante **gestionar** correctamente el archivo de **estado** y **modularizar** el código para mantener la calidad y evitar errores. También es fundamental **documentar** y **versionar** las configuraciones, y automatizar despliegues con pipelines CI/CD para mayor eficiencia.

- **Proteger** y **manejar** el archivo de estado
- **Modularizar** configuraciones
- **Documentar** y versionar código
- **Automatizar** con CI/CD





Conclusiones



Automatización

Terraform permite gestionar toda la infraestructura con código, reduciendo tareas manuales y errores.

Infraestructura

Define el estado deseado y Terraform se encarga de aplicarlo, simplificando la gestión.

Multi-proveedor

Soporta múltiples plataformas y servicios, ideal para entornos híbridos y multi-cloud.

Colaboración

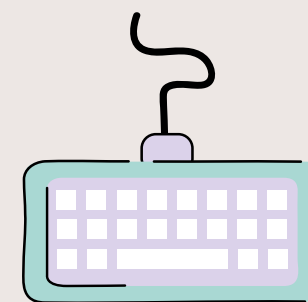
El archivo de estado y la planificación previa facilitan el trabajo en equipo y el control de cambios.

Modularidad

Los módulos fomentan configuraciones limpias, organizadas y fáciles de mantener.

Buenas prácticas

Gestionar bien el estado, documentar y automatizar despliegues es fundamental para el éxito.



Gracias



Por Juan Duran

“Coding, Gaming and Leveling Up”

