

# Infraestructura como código

# Índice

- 01 [¿Qué es?](#)
- 02 [Usos y beneficios](#)
- 03 [Principales herramientas](#)
- 04 [Repasando Terraform](#)



01

¿Qué es?

# IaC - Infraestructura como código

Como todos sabemos, el concepto de Infraestructura como código es cuando podemos definir a la Infraestructura mediante código en vez de crearla manualmente. Esto nos permite que sea más fácil reproducirla y replicarla, teniendo en cuenta que el código de la infraestructura ha evolucionado a lo largo de estos años en varios ámbitos. No solo podemos encontrarnos con IaC, sino también con “red **as a code**”, “configuration **as a code**”, por lo que ya no solo se trata a la infraestructura como un todo.

Por ejemplo, en vez de crear manualmente servidores y redes con su respectiva configuración, podemos definir su infraestructura, su plataforma y su configuración en un código como Terraform, Ansible u otro manifiesto.

02

# Usos y beneficios

# Beneficios

Aunque este proceso de despliegue de infraestructura se encuentre bien documentado, siempre vamos a encontrar obstáculos. La metodología de infraestructura como código nos ofrece una solución a todo esto. A continuación veamos algunos de los principales beneficios que nos brinda.

- Evitar el error humano: minimizamos el riesgo de equivocarnos cuando seguimos una serie de pasos.
- Tiempos y reducción de desperdicios: vamos a poder hacer un *deploy* en cuestión de minutos y sin necesidad de instalar algún componente extra.
- Control de versiones: nuestra infraestructura se va a encontrar definida en archivos, por lo que vamos a poder versionar.
- Reducción de costos: al automatizar procesos, podemos enfocarnos en otras tareas y mejorar lo que ya esté hecho.

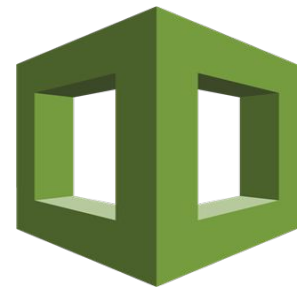
03

# Principales herramientas

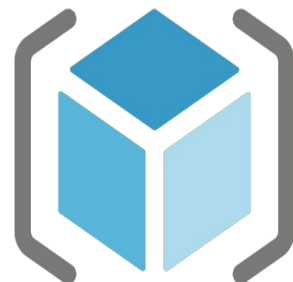
# Herramientas de IaC



[Cloud deployment Manager](#)



[Aws Cloudformation](#)



[Arm Templates](#)



HashiCorp

**Vagrant**

[Vagrant](#)



**Terraform**

[Terraform](#)



ANSIBLE

[Ansible](#)



04

# Repasemos Terraform

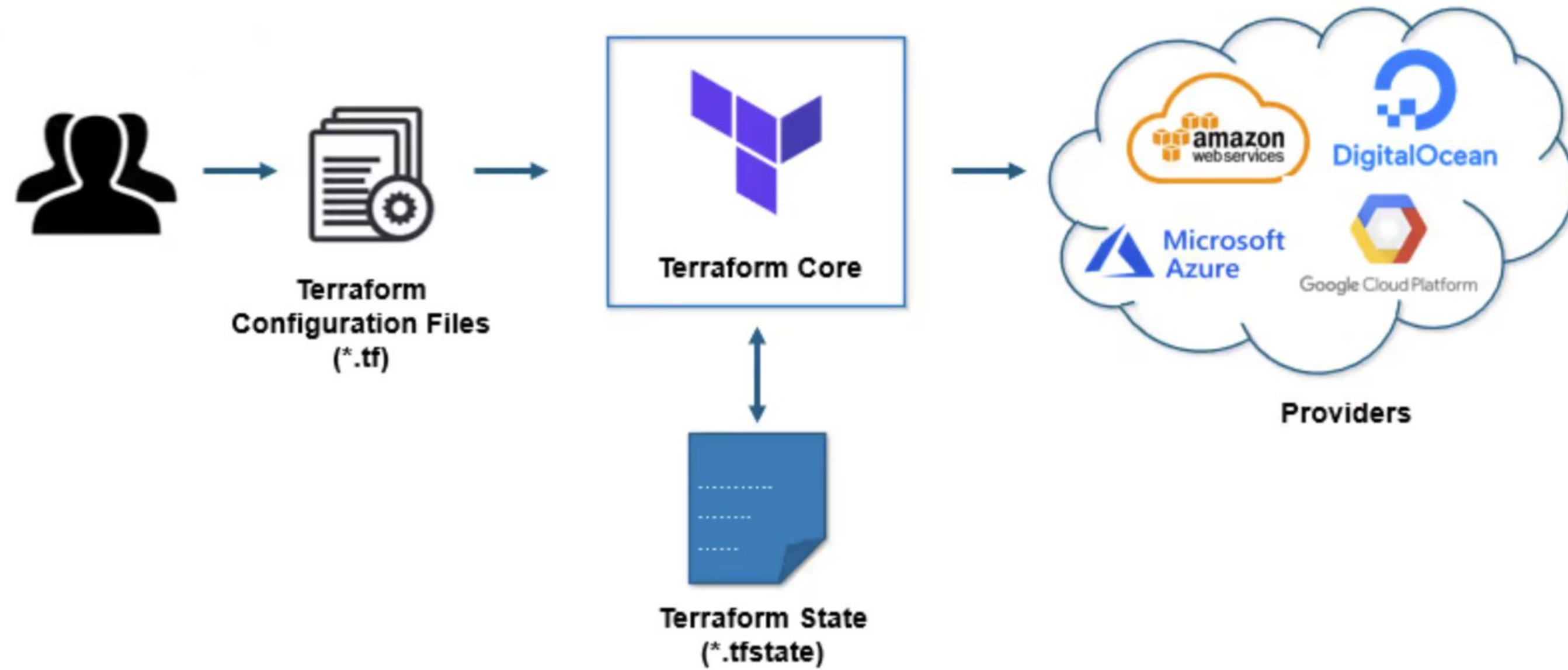
# Terraform

Es una herramienta de orquestación de código abierto desarrollada por [Hashicorp](#) que nos permite definir nuestra infraestructura como código. Esto quiere decir que es posible escribir en un fichero de texto la definición de nuestra infraestructura usando un lenguaje de programación declarativo y simple.

El ciclo de vida de Terraform consiste en: init, plan, aplicar y destruir.

- ➔ **Init** → Inicializa el directorio de trabajo que consta de todos los archivos de configuración.
- ➔ **Plan** → Se utiliza para crear un plan de ejecución para alcanzar el estado deseado de la infraestructura.
- ➔ **Apply** → Realiza los cambios en la infraestructura tal como se define en el plan y la infraestructura llega al estado deseado.
- ➔ **Destroy** → Se utiliza para eliminar todos los recursos de infraestructura antiguos.

# ¿Cómo funciona?



# Proveedores en Terraform

Los proveedores de Terraform se utilizan para crear, administrar y manipular recursos de la infraestructura. Los recursos de los proveedores son utilizados en máquinas físicas, máquinas virtuales, conmutadores de red, contenedores, etc.

Terraform tiene más de cien proveedores para diferentes tecnologías, y luego cada proveedor brinda acceso de usuario de Terraform a sus recursos. Por ejemplo, a través del proveedor de AWS, tiene acceso a cientos de recursos, como instancias EC2, S3, RDS, etc. A continuación detallamos algunos de los proveedores.

Aicloud	Archive	Arukas	AWS	Bitbucket	CenturyLinkCloud
Chef	Circonus	Cloudflare	Cobbler	CloudStack	Consul
Datadog	DNS	DigitalOcean	DNSMadeEasy	DNSimple	Docker
Dyn	Exteral	Fastly	Github	Google Cloud	Grafana
Heroku	HTTP	Icinga2	Ignition	influxDB	Kubernetes
Librato	Local	Logentries	Mailgun	New Relic	Nomad
NS1	Microsoft Azure	MySQL	Microsoft Azure(LegacyASM)	1&1	Oracle Public Cloud
Openstack	OpsGenie	Packet	PagerDuty	PostgreSQL	PowerDNS
ProfitBrcks	RabbitMQ	Rancher	Random	Rundeck	Scaleway
Softlayer	StatusCake	Spotinst	Template	Terraform	Terraform Enterprise
TLS	Triton	UltraDNS	Vault	VMwareCloud	VMware vSphere

¡Muchas gracias!