



# Migración a la nube

Módulo 9 - DevOps y automatización de infraestructura



# Cloud Source Repositories

Repositorios GIT privados.

# ¿Por qué es GIT?

- Se trata del sistema de control de versiones más utilizado del mundo.
- El control de versiones, permite mantener una **gestión centralizada del código de una aplicación**, gestionar privilegios de los usuarios y poder administrar los cambios entre versiones.
- Compatible con cualquier lenguaje y prácticamente cualquier formato de fichero.
- Se trata de una solución de código abierto.



GitLab



# Comandos más usados de GIT

- Inicialización de una carpeta como GIT: **git init**
- Configuración inicial: **git config --global user.name "nombre"** y **git config --global user.email "email"**
- Clonar un repositorio: **git clone [dirección del repositorio HTTP o SSH]**
- Subir nuevos cambios:
  - Añadir archivos: **git add [nombre del archivo o todos con "."]**
  - Etiquetar los cambios: **git commit -m "descripción"**
  - Subir los cambios locales al repositorio: **git push -u origin master**
- Bajar nuevos cambios:
  - Comprueba si hay cambios en el repositorio remoto: **git fetch**
  - Comprueba y baja los cambios del repositorio remoto: **git pull**
- Listar las distintas ramas y cambiar: **git branch** y **git checkout [nombre de la rama]**



# ¿Por qué Source Repositories?

- Diseña, desarrolla y administra tu código con seguridad.
- 100% git compatible.
- Sincronización automática con repositorios externos en GitHub o Bitbucket.
- Repositorios distribuidos geográficamente en varios centros de datos que se ejecutan en la infraestructura de alta disponibilidad.
- Capa gratuita para menos de 5 usuarios por proyecto, 50Gb de almacenamiento y 50Gb de salida al mes.



# Cloud Build

Compila, prueba e implementa, CI/CD sin servidores.

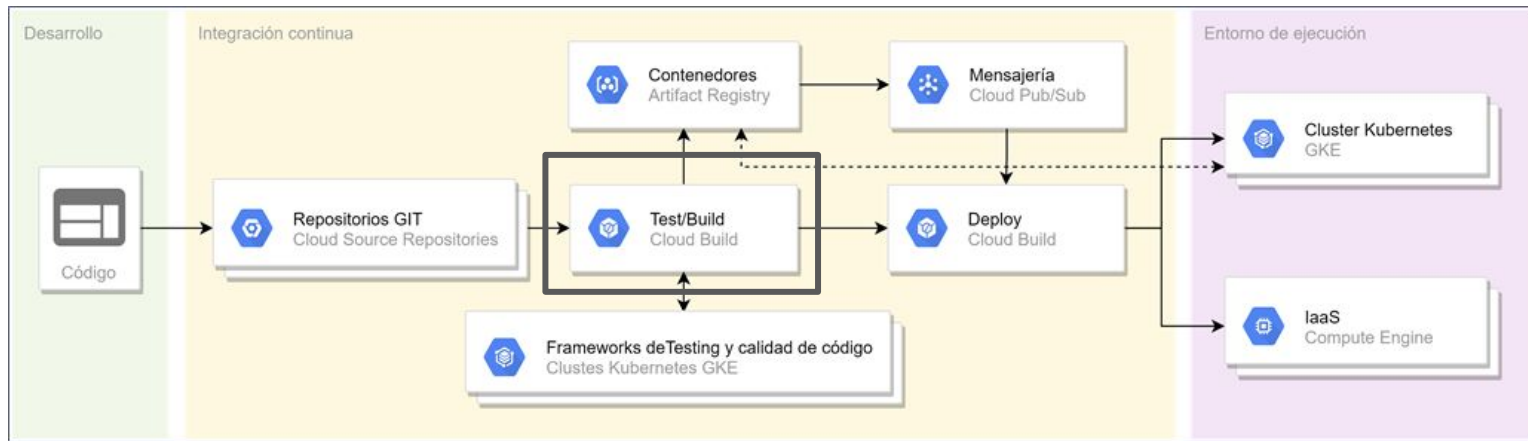


# ¿Por qué Cloud Build?

- Plataforma completamente sin servidores de CI/CD.
- Automatización de implementaciones fácilmente sobre GKE, App Engine, Cloud Functions y Firebase.
- Seguridad, bloqueando automáticamente la implementación de imágenes vulnerables.
- Compatibles con repositorios internos y externos.
- Garantiza el mantener los datos dentro de una región.
- Se factura por minutos de compilación, no se cobra por el tiempo que una tarea está encolada.
- Capa gratuita, los primeros 120 minutos de compilación al día son gratis, luego 0,003\$/minuto según cpu/ram.



# ¿Qué rol suele jugar Cloud Build?





# Empezando con Terraform

The background of the slide is a solid blue color. It features several abstract, geometric shapes in a lighter shade of blue. These shapes are composed of various polygons, including triangles and quadrilaterals, which are arranged in a way that creates a sense of depth and movement. The shapes are primarily located on the right side and bottom of the slide, with some extending towards the center.



# ¿Qué es Terraform?

- Herramienta para orquestación de infraestructura como código (IaC)
- Código abierto
- Desarrollado por Hashicorp
- Sintaxis propia (**HCL**) basada en JSON
- Posibilidad de gestionar varios proveedores



# Ventajas del IaC

- **Reducción de tiempos** de provisión.
- Se **evitan fallos** manuales.
- **Reducción de costes** basados en tiempos y esfuerzos.
- Poder replicar y **escalar** procesos.
- Llevar un **control** pormenorizado de los cambios.
- Despliegues híbridos (cloud y on-premise) o multicloud





# Instalación

Terraform se distribuye como un único archivo ejecutable, se debe descomprimir y registrar en el path del sistema para poder utilizarlo:

- **Windows**

Editar Variables del Entorno del Sistema > Variables de entorno > Path

- **Linux o Mac**

```
sudo mv ~/Downloads/terraform /usr/local/bin/
```

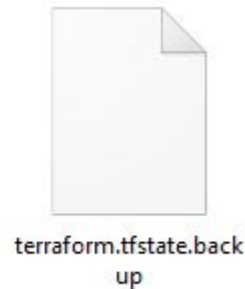
Link de descarga: <https://developer.hashicorp.com/terraform/downloads>



Creando infraestructura

# Estructura de directorio

- **Documento básico:** main.tf
- Archivos propios de Terraform.





# Obteniendo credenciales

- **Cuenta de servicio:** con privilegios de **editor** del proyecto.
- Descargar clave en JSON.

 **Cuentas de servicio** [+ CREAR CUENTA DE SERVICIO](#)

### Claves

Añade un par de claves nuevo o sube el certificado de clave pública de un par de claves que ya tengas. Recuerda que los certificados públicos deben tener el formato RSA\_X509\_PEM. [Más información sobre los formatos de claves de subida](#)

AÑADIR CLAVE ▾

Tipo	Estado	Clave	Fecha de creación de clave	Fecha de vencimiento de la clave
No hay filas que mostrar				

**Tipo de clave**

☒ JSON

Recomendado



# Comandos principales

- **init:** Inicializa el proyecto de terraform e instala dependencias.
- **apply:** Ejecuta toda la configuración y crea la infraestructura.
- **destroy:** Elimina toda la infraestructura en base a la configuración.
- **plan:** Revisa toda la configuración y la compara con la infra actual.
- **validate:** Valida que la sintaxis de la configuración es correcta.





# Configurando GCP

```
terraform {  
  required_providers {  
    google = {  
      source = "hashicorp/google"  
      version = "4.81.0"  
    }  
  }  
}
```

```
provider "google" {  
  
  credentials = file("cred.json")  
  
  project = "id_proyecto"  
  region  = "europe-west3"  
  zone    = "europe-west3-a"  
}
```



# Redes

```
resource "google_compute_network" "vpc_network" {  
  name = "keep-network"  
}
```

Direcciones IP

```
resource "google_compute_address" "vm_static_ip" {  
  name = "keep-static-ip"  
}
```



# Máquinas virtuales

```
resource "google_compute_instance" "vm_instance" {  
  name          = "keep-instance"  
  machine_type  = [maquina]  
  tags          = ["etiqueta",...]  
}
```

## Imágen de disco

```
boot_disk {  
  initialize_params {  
    image = [image]  
  }  
}
```

## Configuración de red

```
network_interface {  
  network = google_compute_network.vpc_network.name  
  access_config {  
    nat_ip = google_compute_address.vm_static_ip.address  
  }  
}
```



# Almacenamiento

```
resource "google_storage_bucket" "example_bucket" {  
  name      = "keep-almacenamiento12345"  
  location = "EUROPE-WEST3"
```

Hospedar web estática

```
  website {  
    main_page_suffix = "index.html"  
    not_found_page   = "404.html"  
  }  
}
```



# Funciones especiales

```
resource "random_string" "texto" {
```

```
  length = 9
```

```
  special = false
```

```
  upper = false
```

```
}
```

```
...
```

```
name = "nombre-concatenado-${random_string.texto.result}"
```

```
...
```

```
depends_on = [google_storage_bucket.nombre_del_bucket]
```

Muchas gracias

¿Preguntas?