Manual de Provisionamiento con Terraform

SRE-GCP Team

25 de junio de 2025

Contents

1 Introducción

1.1 Propósito del Manual

Este manual documenta el uso de Terraform para provisionar la infraestructura del proyecto SRE-GCP, incluyendo Load Balancer, Google Kubernetes Engine (GKE), Cloud SQL, entre otros. Se enfoca en optimización y escalabilidad, proporcionando a los SRE guías prácticas para gestionar recursos de manera eficiente y escalable en Google Cloud Platform (GCP).

1.2 Audiencia Objetivo

Dirigido a profesionales SRE con experiencia en infraestructura como código (IaC), interesados en optimizar y mantener entornos cloud-native.

2 Estructura de Módulos

2.1 Descripción de Módulos

La infraestructura se organiza en módulos dentro del directorio `terraform/modules/`:

- `load_balancer`: Configura un balanceador de carga HTTP(S) global con `web-backend-service`.
- `gke`: Define un clúster GKE con pool de nodos escalable.
- `cloud_sql`: Provisiona una instancia PostgreSQL en Cloud SQL.
- `cloud_dns`: Gestiona registros DNS para el dominio.
- `bastion`: Crea un host bastion para acceso seguro.
- `artifact_registry`: Configura Container Registry para imágenes Docker.

Cada módulo incluye archivos `main.tf`, `variables.tf`, y `outputs.tf`.

2.2 Relación entre Módulos

Los módulos están interconectados: el `load_balancer` depende de `gke` para los backends, y `gke` utiliza imágenes de `artifact_registry`. Esto asegura una provisionación coherente.

3 Configuración de Variables y Outputs

3.1 Definición de Variables

Las variables se definen en `variables.tf` y se configuran en `terraform.tfvars`. Ejemplo:

```
project_id        = "rugged-silo-463917-i2"
region            = "us-central1"
zone              = "us-central1-a"
domain            = "example.com"
database_name     = "appdb"
database_user     = "appuser"
database_password = "secure_password"
```

3.2 Uso de Outputs

Los outputs en `outputs.tf` exponen datos clave, como:

```
output "load_balancer_ip" {
  value = google_compute_global_forwarding_rule.default.
     ip_address
}
```

Esto permite recuperar la IP del balanceador con `terraform output load`$_b alancer_i p$.

4 Mejores Prácticas para Gestión del Estado

4.1 Gestión de `terraform.tfstate`

El archivo terraform.tfstate almacena el estado de la infraestructura. Nunca lo modifiques manualmente; úsalo con terraform apply o terraform destroy.

4.2 Almacenamiento Remoto

Configura un backend remoto (por ejemplo, Google Cloud Storage) para evitar conflictos:

```
terraform {
  backend "gcs" {
    bucket = "tf-state-rugged-silo-463917-i2"
    prefix = "terraform/state"
  }
}
```

Inicializa con terraform init.

4.3 Recuperación de Estados Corruptos

Si terraform.tfstate se corrompe, usa un backup (terraform.tfstate.backup) o el comando:

```
terraform import <resource_type>.<resource_name> <resource_id>
```

Ejemplo: terraform import google$_c ompute_i nstance.bastionbastion-host-$20250624.

5 Optimización y Escalabilidad

5.1 Configuración de Recursos Escalables

- Ajusta el autoscaling de GKE en gke/main.tf:

```
node_pool {
  autoscaling {
    min_node_count = 1
    max_node_count = 5
  }
}
```

- Escala Cloud SQL verticalmente con tier = "db-custom-2-4096" para 2 vCPUs y 4 GB RAM.

5.2 Validación de Infraestructura

Valida la configuración con terraform validate y revisa planes con terraform plan -var-file="terraform.tfvars" antes de aplicar.

6 Ejemplo Práctico

Para provisionar la infraestructura: 1. Crea terraform.tfvars con:

```
project_id        = "rugged-silo-463917-i2"
region            = "us-central1"
zone              = "us-central1-a"
domain            = "example.com"
database_name     = "appdb"
database_user     = "appuser"
database_password = "secure_password"
```

2. Inicializa y aplica:

```
cd terraform
terraform init
terraform apply -var-file="terraform.tfvars"
```

Confirma con yes para provisionar los recursos.

7 Escenarios de Uso

- **Actualización de la Infraestructura por un SRE Senior**: Un SRE senior puede modificar variables.tf para aumentar el max_node_count a 10 y ejecutar terraform apply para escalar GKE. - **Recuperación de un Estado Corrupto de Terraform**: Si terraform.tfstate falla, un SRE puede restaurar desde terraform.tfst
y reimportar recursos con terraform import.

8 Referencias y Recursos Adicionales

- [Terraform Documentation](https://www.terraform.io/docs) - [GCP Terraform Provider](https://registry.terraform.io/providers/hashicorp/google/latest/docs)