

Reporte de Laboratorio: Interactuando con Módulos de Terraform (GSP751)

Autor: Christhian Rodriguez

Fecha: 12 de Diciembre, 2025

Tema: Modularización de Infraestructura como Código (IaC)

1. Introducción

El objetivo de este laboratorio fue transicionar de escribir configuraciones de Terraform monolíticas (un solo archivo gigante) a una arquitectura **modular**. Aprendimos a consumir módulos públicos del Registry de Terraform y, más importante aún, a construir nuestros propios módulos locales para encapsular la lógica, reutilizar código y mejorar la seguridad.

2. Tarea 1: Uso de Módulos del Registry

En la primera parte, no escribimos el código desde cero, sino que "alquilamos" una receta ya hecha.

- **Acción:** Clonamos el repositorio `terraform-google-network`.
- **Concepto:** Usamos el bloque `module` apuntando a una fuente externa (`source`) y definimos variables específicas para nuestro entorno (Red y Subredes).
- **Resultado:** Se desplegó una VPC completa sin tener que definir cada recurso de red manualmente.

3. Tarea 2: Construcción de un Módulo Personalizado (Bucket Estático)

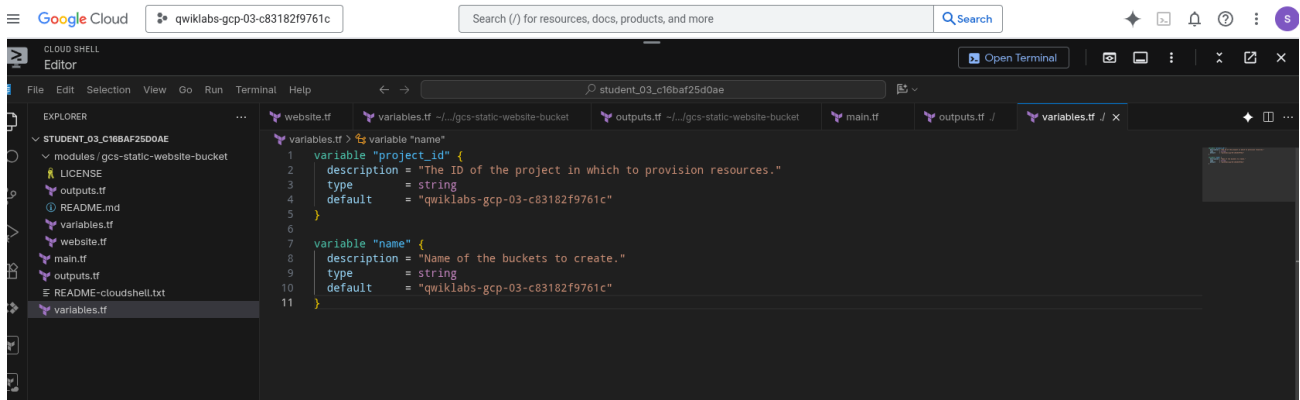
Esta fue la sección principal del laboratorio. Creamos una estructura de archivos que separa la lógica del recurso (el módulo) de la configuración del usuario (la raíz).

3.1 Estructura del Proyecto

Organizamos el directorio para separar "La Cocina" (el módulo) del "Mostrador" (la raíz).

- **Raíz (~/): `main.tf` , `variables.tf` , `outputs.tf` .**

- **Módulo (~/modules/gcs...):** `website.tf` , `variables.tf` , `outputs.tf` , `LICENSE` .



Evidencia de Estructura: En la siguiente imagen se aprecia el árbol de directorios a la izquierda, mostrando la carpeta `modules` y los archivos raíz.

3.2 Definición del Módulo ("La Cocina")

Dentro de la carpeta `modules/gcs-static-website-bucket` , creamos el archivo `website.tf` . Este archivo contiene el recurso `google_storage_bucket` y utiliza variables internas para configurar sus propiedades dinámicamente.



Código del Recurso: Aquí vemos la definición del recurso con bloques dinámicos para el ciclo de vida y la retención.

3.3 Configuración Raíz ("El Mostrador")

En el directorio raíz, editamos el `main.tf` para **llamar** al módulo que acabamos de crear. Pasamos los valores específicos (como el `project_id`) desde la raíz hacia el módulo.

```
main.tf | module "gcs-static-website-bucket"
1 module "gcs-static-website-bucket" {
2   source = "../modules/gcs-static-website-bucket"
3
4   name       = var.name
5   project_id = var.project_id
6   location   = "us-central1"
7
8   lifecycle_rules = [{
9     action = {
10      type = "Delete"
11    }
12    condition = {
13      age       = 365
14      with_state = "ANY"
15    }
16  }]
17 }
```

Llamada al Módulo: El bloque `module` en la raíz apuntando a `../modules/gcs-static-website-bucket`.

También definimos un `output` en la raíz para que Terraform nos muestre el nombre del bucket al finalizar. Esto conecta la salida del módulo con la consola del usuario.

```
modules > gcs-static-website-bucket > website.tf | resource "google_storage_bucket" "bucket"
1 resource "google_storage_bucket" "bucket" {
2   name       = var.name
3   project    = var.project_id
4   location   = var.location
5   storage_class = var.storage_class
6   labels     = var.labels
7   force_destroy = var.force_destroy
8   uniform_bucket_level_access = true
9
10  versioning {
11    enabled = var.versioning
12  }
13
14  dynamic "retention_policy" {
15    for_each = var.retention_policy == null ? [] : [var.retention_policy]
16    content {
17      is_locked       = var.retention_policy.is_locked
18      retention_period = var.retention_policy.retention_period
19    }
20  }
21
22  dynamic "encryption" {
23    for_each = var.encryption == null ? [] : [var.encryption]
24    content {
25      default_kms_key_name = var.encryption.default_kms_key_name
26    }
27  }
28
29  dynamic "lifecycle_rule" {
30    for_each = var.lifecycle_rules
31    content {
32      action {
33        type = lifecycle_rule.value.action.type
34        storage_class = lookup(lifecycle_rule.value.action, "storage_class", null)
35      }
36      condition {
37        age = lookup(lifecycle_rule.value.condition, "age", null)
38        created_before = lookup(lifecycle_rule.value.condition, "created_before", null)
39        with_state = lookup(lifecycle_rule.value.condition, "with_state", null)
40        matches_storage_class = lookup(lifecycle_rule.value.condition, "matches_storage_class", null)
41        num_newer_versions = lookup(lifecycle_rule.value.condition, "num_newer_versions", null)
42      }
43    }
44  }
45 }
```

Output Raíz:

3.4 Despliegue (Terraform Apply)

Ejecutamos `terraform init` para registrar el módulo local y `terraform apply` para crear la infraestructura.

```
student_03_c16baf25d0ae@cloudshell:~ (qwiklabs-gcp-03-c83182f9761c)$ terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# module.gcs-static-website-bucket.google_storage_bucket.bucket will be created
+ resource "google_storage_bucket" "bucket" {
+   effective_labels = {
+     "goog-terraform-provisioned" = "true"
+   }
+   force_destroy    = true
+   id               = (known after apply)
+   location         = "US-CENTRAL1"
+   name             = "qwiklabs-gcp-03-c83182f9761c"
+   project          = "qwiklabs-gcp-03-c83182f9761c"
+   project_number   = (known after apply)
+   public_access_prevention = (known after apply)
+   rpo              = (known after apply)
+   self_link        = (known after apply)
+   storage_class     = "STANDARD"
+   terraform_labels = {
+     "goog-terraform-provisioned" = "true"
+   }
+   time_created     = (known after apply)
+   uniform_bucket_level_access = true
+   updated          = (known after apply)
+   url              = (known after apply)

+   lifecycle_rule {
+     action {
+       type = "Delete"
+     }
+     condition {
+       age                = 365
+       matches_prefix     = []
+       matches_storage_class = []
+       matches_suffix     = []
+       with_state         = "ANY"
+     }
+   }
+   versioning {
+     enabled = true
+   }
+ }

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ bucket-name = "module.gcs-static-website-bucket.bucket"

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: [
```

El Plan de Ejecución: Terraform detecta que debe crear un recurso `google_storage_bucket` dentro del módulo.

```

    versioning {
      + enabled = true
    }
  }
}

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + bucket-name = "module.gcs-static-website-bucket.bucket"

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

module.gcs-static-website-bucket.google_storage_bucket.bucket: Creating...
module.gcs-static-website-bucket.google_storage_bucket.bucket: Creation complete after 1s [id=qwiklabs-gcp-03-c83182f9761c]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:
bucket-name = "module.gcs-static-website-bucket.bucket"
student_03_c16baf25d0ae@cloudshell:~ (qwiklabs-gcp-03-c83182f9761c) $ cd ~
curl https://raw.githubusercontent.com/hashicorp/learn-terraform-modules/master/modules/aws-s3-static-website-bucket/www/index.html > index.html
curl https://raw.githubusercontent.com/hashicorp/learn-terraform-modules/blob/master/modules/aws-s3-static-website-bucket/www/error.html > error.html

```

Confirmación y Creación: Confirmamos con "yes" y obtenemos el nombre del bucket en los Outputs verdes.

3.5 Carga de Contenido y Verificación

Descargamos archivos HTML de ejemplo y los subimos al bucket usando `gsutil`.

```

bucket-name = "module.gcs-static-website-bucket.bucket"
student_03_c16baf25d0ae@cloudshell:~ (qwiklabs-gcp-03-c83182f9761c) $ cd ~
curl https://raw.githubusercontent.com/hashicorp/learn-terraform-modules/master/modules/aws-s3-static-website-bucket/www/index.html > index.html
curl https://raw.githubusercontent.com/hashicorp/learn-terraform-modules/blob/master/modules/aws-s3-static-website-bucket/www/error.html > error.html
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
100    257    100    257    0     0    1167    0 --:--:-- --:--:-- --:--:--   1168
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
100     14    100     14    0     0     93    0 --:--:-- --:--:-- --:--:--    93
student_03_c16baf25d0ae@cloudshell:~ (qwiklabs-gcp-03-c83182f9761c) $ gsutil cp *.html gs://qwiklabs-gcp-03-c83182f9761c
Copying file://error.html [Content-Type=text/html]...
Copying file://index.html [Content-Type=text/html]...
/ [2 files][ 271.0 B/ 271.0 B]
Operation completed over 2 objects/271.0 B.
student_03_c16baf25d0ae@cloudshell:~ (qwiklabs-gcp-03-c83182f9761c) $

```

Subida de Archivos: Usando `gsutil cp` para mover el `index.html` a la nube.

Verificamos en la Consola de Google Cloud que el bucket existe y contiene los archivos.

The screenshot shows the Google Cloud Console interface for a bucket named 'qwiklabs-gcp-03-c83182f9761c'. The bucket details are as follows:

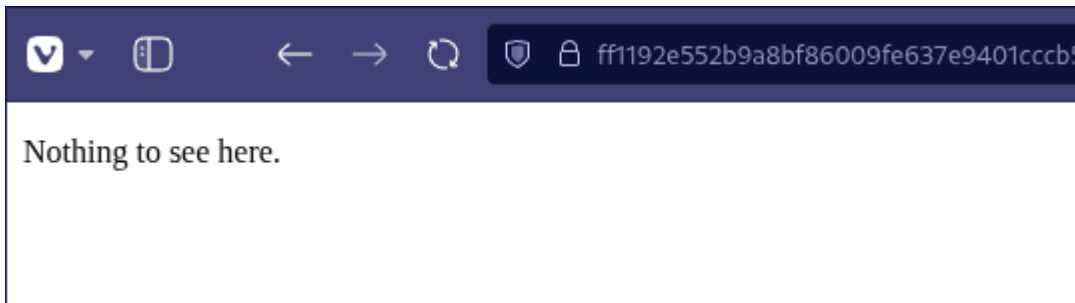
Location	Storage class	Public access	Protection
us-central1 (Iowa)	Standard	Not public	Soft Delete, Object versioning

The 'Objects' tab is selected, showing a folder browser view. The bucket contains two objects:

Name	Size	Type	Created	Storage class	Last modified
error.html	14 B	text/html	Dec 11, 2025, 8:31:41 PM	Standard	Dec 11, 2025, 8:31:41 PM
index.html	257 B	text/html	Dec 11, 2025, 8:31:41 PM	Standard	Dec 11, 2025, 8:31:41 PM

Verificación en GUI: El bucket `qwiklabs-gcp...` con los archivos cargados.

Finalmente, accedemos a la URL pública para ver el sitio estático funcionando.



Sitio Web en Vivo:

3.6 Limpieza

Como buena práctica, destruimos la infraestructura al terminar.

Terraform Destroy:

```
Plan: 0 to add, 0 to change, 1 to destroy.

Changes to Outputs:
- bucket-name = "module.gcs-static-website-bucket.bucket" -> null

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

module.gcs-static-website-bucket.google_storage_bucket.bucket: Destroying... [id=qwiklabs-gcp-03-c83182f9761c]
module.gcs-static-website-bucket.google_storage_bucket.bucket: Destruction complete after 1s

Destroy complete! Resources: 1 destroyed.
student_03_c16baf25d0ae@cloudshell:~ (qwiklabs-gcp-03-c83182f9761c)$
```

4. Zona de Aprendizaje: Análisis y Conceptos Clave

Esta sección resume las discusiones técnicas mantenidas durante el laboratorio para consolidar el conocimiento.

4.1 La Analogía de la Panadería (Modularización)

Entendimos que usar módulos es como operar una franquicia de panaderías:

- **El Módulo (La Cocina):** Es donde están las reglas, las recetas (`resources`) y los procesos internos. Es código genérico y reutilizable.
- **La Raíz (El Mostrador):** Es donde se hacen los pedidos. No se cocina aquí, solo se pasan parámetros (`variables`) a la cocina.
- **Beneficio:** Si necesitas 10 buckets, no construyes 10 cocinas. Haces 10 pedidos al mismo módulo.

4.2 El Flujo de Datos (La Tubería)

Aprendimos por qué parece que "duplicamos" variables. Terraform no conecta mágicamente la raíz con el módulo.

- **Input:** Usuario -> `variables.tf` (Raíz) -> `main.tf` (Llamada) -> `variables.tf` (Módulo) -> Recurso.
- **Output:** Recurso -> `outputs.tf` (Módulo) -> `outputs.tf` (Raíz) -> Consola del Usuario.

4.3 Seguridad: Lo que se comparte y lo que se oculta

Discutimos la importancia crítica de no exponer datos sensibles al subir código a repositorios como GitHub (Git).

Archivo / Carpeta	¿Subir al Repo?	Razón
<code>main.tf</code> , <code>variables.tf</code> , <code>outputs.tf</code>	SÍ	Es la estructura y lógica del código (la receta).
<code>LICENSE</code> , <code>README.md</code>	SÍ	Documentación y permisos legales para usuarios.
<code>terraform.tfstate</code>	NO	Peligro Crítico. Contiene contraseñas y datos reales en texto plano.
<code>*.tfvars</code>	NO	Contiene tus valores secretos (API keys, passwords).
<code>.terraform/</code>	NO	Son binarios y plugins pesados específicos de tu PC.

Conclusión: Un buen ingeniero de DevOps siempre incluye un archivo `.gitignore` configurado para bloquear el estado y las variables sensibles antes de hacer un commit.