

# GSP313 - Implementación de Balanceo de Cargas (L4 y L7)

---

En este proyecto, completé el Lab de Desafío GSP313, que requería una comprensión sólida de los servicios de red de Google Cloud. El desafío consistía en implementar no uno, sino **dos** tipos de balanceadores de carga (NLB y ALB) en un mismo proyecto, cada uno con sus propias VMs y reglas.

Este documento sirve como mi registro personal de la arquitectura que implementé, los comandos exactos que utilicé, y la lección clave que aprendí para superar el error más común de este laboratorio.

## 1. El Desafío: Dos Balanceadores, Un Proyecto

---

El objetivo era doble:

1. **Balanceador de Cargas de Red (NLB L4):** Crear un balanceador regional (de paso) que distribuyera tráfico TCP entre tres instancias de VM creadas manualmente ( `web1` , `web2` , `web3` ).
2. **Balanceador de Cargas de Aplicaciones (ALB L7):** Crear un balanceador global (HTTP) que distribuyera tráfico entre un Grupo de Instancias Administrado (MIG).

## 2. Tarea 1: Creación de Servidores Base (para el NLB)

---

Primero, preparé el terreno para el Balanceador de Red (NLB). Esto implicó crear tres instancias de VM idénticas, cada una con un servidor Apache que mostraba su propio nombre ( `web1` , `web2` , `web3` ).

También creé la regla de firewall `www-firewall-network-lb` para permitir el tráfico HTTP (puerto 80) a cualquier instancia con la etiqueta `network-lb-tag` .

```
# Configurar mi región y zona
gcloud config set compute/region us-west4
gcloud config set compute/zone us-west4-c
ZONE="us-west4-c"

# 1. Crear instancia web1
gcloud compute instances create web1 \
  --zone=$ZONE \
  --tags=network-lb-tag \
  --machine-type=e2-small \
  --image-family=debian-12 \
  --image-project=debian-cloud \
  --metadata=startup-script='#!/bin/bash
  apt-get update
  apt-get install apache2 -y
  service apache2 restart
  echo "<h3>Web Server: web1</h3>" | tee /var/www/html/index.html'


# 2. Crear instancia web2
gcloud compute instances create web2 \
  --zone=$ZONE \
  --tags=network-lb-tag \
  --machine-type=e2-small \
  --image-family=debian-12 \
  --image-project=debian-cloud \
  --metadata=startup-script='#!/binS/bash
  apt-get update
  apt-get install apache2 -y
  service apache2 restart
  echo "<h3>Web Server: web2</h3>" | tee /var/www/html/index.html'

# 3. Crear instancia web3
gcloud compute instances create web3 \
  --zone=$ZONE \
  --tags=network-lb-tag \
  --machine-type=e2-small \
  --image-family=debian-12 \
  --image-project=debian-cloud \
  --metadata=startup-script='#!/bin/bash
  apt-get update
  apt-get install apache2 -y
  service apache2 restart
  echo "<h3>Web Server: web3</h3>" | tee /var/www/html/index.html'

# 4. Crear la regla de firewall para el NLB
gcloud compute firewall-rules create www-firewall-network-lb \
  --network=default \
  --action=allow \
  --direction=INGRESS \
```

```
--rules=tcp:80 \  
--target-tags=network-lb-tag
```

Después de crear las instancias, verifiqué que cada una respondía correctamente usando `curl` a sus IPs externas.

 2025-11-10 12.50.28 console.cloud.google.com 6eab1a7f593f.png

### 3. Tarea 2: Configuración del Balanceador de Cargas de Red (L4)

---

Con las VMs listas, construí el Balanceador de Cargas de Red (L4). Este tipo de balanceador es **Regional** y utiliza un "Target Pool".

**Lección Clave:** El *health check* para este balanceador ( `http-basic-check` ) también es **Regional** por defecto. Esto es crucial, como veremos en la Tarea 3.

```

REGION="us-west4"
ZONE="us-west4-c"

# 1. Crear una IP estática REGIONAL para el NLB
gcloud compute addresses create network-lb-ip-1 \
  --region=$REGION

# 2. Crear una Verificación de Estado (Health Check) REGIONAL
# NOTA: Este es el `http-basic-check` regional.
gcloud compute http-health-checks create http-basic-check \
  --port 80 \
  --region=$REGION

# NOTA: Si el lab espera un health check 'http' y no 'regional-http', usa:
# gcloud compute http-health-checks create http-basic-check --port 80

# 3. Crear el Grupo de Destino (Target Pool)
gcloud compute target-pools create www-pool \
  --region=$REGION \
  --http-health-check=http-basic-check

# 4. Añadir mis 3 instancias al Target Pool
gcloud compute target-pools add-instances www-pool \
  --instances=web1,web2,web3 \
  --instances-zone=$ZONE

# 5. Crear la Regla de Reenvío (Forwarding Rule)
gcloud compute forwarding-rules create www-rule \
  --region=$REGION \
  --ports=80 \
  --address=network-lb-ip-1 \
  --target-pool www-pool

```

Este bloque de comandos se ejecutó con éxito, creando el primer balanceador.

 2025-11-10 12.55.27 console.cloud.google.com 1f044e4ba1e7.png

## 4. Tarea 3: Creación del Balanceador de Cargas de Aplicaciones (L7)

---

Esta fue la parte más compleja y donde ocurrió el aprendizaje clave. Este balanceador es **Global** y utiliza un "Backend Service" que apunta a un Grupo de Instancias Administrado (MIG).

### 4.1. El Detalle Técnico: Global vs. Regional (¡La Clave del Éxito!)

El error más común de este lab es intentar reutilizar el *health check* `http-basic-check` de la Tarea 2.

- **El Problema:** El `backend-service` (Paso 5) es un recurso **Global**. Por lo tanto, *requiere* un *health check* que también sea **Global**. El `http-basic-check` que creé en la Tarea 2 era **Regional** (o `http` simple, que también es global pero da conflicto de nombre).
- **La Solución:** La forma correcta de evitar la cascada de errores es crear un **NUEVO** *health check* con un nombre diferente (ej: `http-global-check` ) y especificar que es `--global` .

## 4.2. Los Comandos Correctos (La Pila L7 Completa)

Este es el bloque de comandos completo y correcto para la Tarea 3, que evita todos los conflictos.

```
REGION="us-west4"
```

```
ZONE="us-west4-c"
```

```
# PASO 1: Crear la Plantilla de Instancias (es GLOBAL por defecto)
```

```
gcloud compute instance-templates create lb-backend-template \
  --tags=allow-health-check \
  --machine-type=e2-medium \
  --image-family=debian-12 \
  --image-project=debian-cloud \
  --metadata=startup-script='#!/bin/bash
    apt-get update
    apt-get install apache2 -y
    service apache2 restart
    echo "<p>Page served from: $HOSTNAME</p>" | tee /var/www/html/index.html'
```

```
# PASO 2: Crear el Grupo de Instancias Administrado (MIG)
```

```
gcloud compute instance-groups managed create lb-backend-group \
  --template=lb-backend-template \
  --size=2 \
  --zone=$ZONE
```

```
# PASO 3: Crear la regla de firewall fw-allow-health-check
```

```
gcloud compute firewall-rules create fw-allow-health-check \
  --network=default \
  --action=allow \
  --direction=INGRESS \
  --source-ranges=130.211.0.0/22,35.191.0.0/16 \
  --target-tags=allow-health-check \
  --rules=tcp:80
```

```
# PASO 4: Crear la IP estática GLOBAL para el ALB
```

```
gcloud compute addresses create lb-ipv4-1 \
  --ip-version=IPV4 \
  --global
```

```
# PASO 4.5 (¡LA CLAVE!): Crear un NUEVO health check GLOBAL
```

```
gcloud compute health-checks create http http-global-check \
  --port 80 \
  --global
```

```
# PASO 5 (Corregido): Crear el Servicio de Backend
```


```
# Le decimos que use nuestro NUEVO check global.
```

```
gcloud compute backend-services create web-backend-service \
  --protocol=HTTP \
  --health-checks=http-global-check \
  --global
```

```
# PASO 6: Añadir mi MIG al Servicio de Backend
```


```
gcloud compute backend-services add-backend web-backend-service \
  --instance-group=lb-backend-group \
```

```
--instance-group-zone=$ZONE \  
--global  
  
# PASO 7: Crear el Mapa de URLs  
gcloud compute url-maps create web-map-http \  
--default-service web-backend-service  
  
# PASO 8: Crear el Proxy HTTP  
gcloud compute target-http-proxies create http-lb-proxy \  
--url-map web-map-http  
  
# PASO 9: Crear la Regla de Reenvío GLOBAL  
gcloud compute forwarding-rules create http-content-rule \  
--address=lb-ipv4-1 \  
--global \  
--target-http-proxy=http-lb-proxy \  
--ports=80
```

 2025-11-10 13.21.06 console.cloud.google.com 51e77ae4e4e3.png

La ejecución de este bloque completo fue exitosa, creando toda la pila del ALB L7.

Aquí está la prueba de mi nuevo *health check* global:

 2025-11-10 13.22.59 console.cloud.google.com 733a5a078f05.png

 2025-11-10 13.23.18 console.cloud.google.com c5182c9f3916.png

## 5. Verificación Final y Conclusión

---

El último paso fue el más importante: **esperar**.

El balanceador de cargas de aplicaciones (L7) tarda entre 5 y 10 minutos en propagarse y marcar las instancias del MIG como "HEALTHY". Durante este tiempo, la calificación del lab fallará.

Usé el siguiente comando para monitorear el estado hasta que ambas instancias estuvieran "HEALTHY":


```
gcloud compute backend-services get-health web-backend-service --global
```

Una vez que el estado fue "HEALTHY", la consola de Google Cloud mostró ambos balanceadores y sus backends correctamente.

**Vista de Balanceadores:** Ambos balanceadores (L4 y L7) activos.

 2025-11-10 13.22.09 console.cloud.google.com 36080da557e5.png

**Vista de Backends:** Ambos backends (el Target Pool Regional y el Backend Service Global) activos.

 2025-11-10 13.22.02 console.cloud.google.com 2038fc069426.png

Finalmente, probé la IP pública ( `lb-ipv4-1` ) del balanceador L7 en un navegador. Al recargar, el nombre de host cambiaba, confirmando que el balanceo de carga funcionaba.

 2025-11-10 13.25.16 34.107.205.207 11836c97ab36.png

## Conclusión

Este desafío fue un ejercicio excelente sobre las diferencias críticas entre los recursos Regionales y Globales en Google Cloud. La clave del éxito no fue solo ejecutar los comandos, sino entender por qué ciertos componentes (como un `target-pool` regional y un `backend-service` global) no pueden compartir los mismos recursos con el mismo nombre y alcance.