

Práctica de Migración a la Nube

Google Cloud Platform

Módulo: Migración a la Nube

KeepCoding - Bootcamp DevOps

Autor: Antonio Prado

Fecha: Enero 2026

Índice

1. Primera Parte: Configuración Inicial del Proyecto
2. Segunda Parte: Cloud SQL y Cloud Storage
3. Tercera Parte: Compute Engine y Autoescalado
4. Cuarta Parte: App Engine
5. Diagrama de Arquitectura

1. Primera Parte: Configuración Inicial

1.1 Creación del Proyecto

Se ha creado un nuevo proyecto en Google Cloud Platform con el identificador **silver-fragment-481408-c9**.

Select a project  **New project**

Search projects and folders

Recent Starred All












Name	Type	ID
✓  anzeni-cloud-lab 	Project	silver-fragment-481408-c9 

1.2 Acceso al Profesor

Se ha otorgado acceso completo al proyecto al profesor (o.martinezabella@gmail.com) mediante el rol de Owner en IAM.






Pasos realizados:

1. Acceder a IAM & Admin → IAM
2. Clic en Grant Access
3. Añadir el email del profesor como Principal
4. Asignar el rol correspondiente

<input type="checkbox"/> Type	Principal ↑	Name	Role	Security Insights ⓘ	
<input type="checkbox"/> 	519247385290-compute@developer.gserviceaccount.com	Compute Engine default service account	Editor	11032/11034 excess permissions	
<input type="checkbox"/> 	o.martinezabella@gmail.com		Owner 		
<input type="checkbox"/> 	silver-fragment-481408-c9@appspot.gserviceaccount.com	App Engine default service account	Cloud SQL Client		
<input type="checkbox"/> 	terraform-admin@silver-fragment-481408-c9.iam.gserviceaccount.com	terraform-admin	Editor	11028/11038 excess permissions	
<input type="checkbox"/> 	tonipl@gmail.com	tonipl	Owner	12143/12527 excess permissions	

1.3 Avisos de Facturación

Se han configurado alertas de presupuesto para controlar el gasto del proyecto. Las alertas se han establecido en los siguientes porcentajes:

Budget name ↑	Budget period	Budget type	Applies to	Trigger alerts at	Spend and budget amount
Alerta de seguridad	Monthly	Specified amount	This billing account	50%, 90%, and 100%	 -€0.00 / €5.00 Includes -€12.75 savings
App Engine	Monthly	Specified amount	Service "App Engine"	30%, 60%, and 100%	 €0.00 / €20.00 No credits used
Cloud Storage	Monthly	Specified amount	Service "Cloud Storage"	10%, 50%, and 100%	 €0.00 / €20.00 No credits used
CloudSQL	Monthly	Specified amount	Service "Cloud SQL"	5%, 65%, and 100%	 -€0.00 / €20.00 Includes -€8.09 savings
General	Monthly	Specified amount	This billing account	50%, 90%, and 100%	 -€0.00 / €100.00 Includes -€12.75 savings

2. Segunda Parte: Cloud SQL y Cloud Storage

2.1 Creación de la Instancia Cloud SQL

Se ha creado una instancia de Cloud SQL con MySQL para alojar las bases de datos del proyecto.

Configuración de la instancia:

- Nombre: **anzeni-db**
- Tipo: MySQL
- Región: europe-southwest1 (Madrid)
- Connection name:
silver-fragment-481408-c9:europe-southwest1:anzeni-db

2.2 Creación de Usuario y Bases de Datos


Se ha creado un usuario específico para la aplicación y dos bases de datos.

- Usuario: **alumno**
- Base de datos 1: **google**
- Base de datos 2: **cloud**


2.3 Cloud Storage - Bucket para Exportaciones



Se ha creado un bucket en Cloud Storage para almacenar las exportaciones de las bases de datos SQL.

- Nombre del bucket: **backup-sql-anzeni**
- Uso: Almacenar exports/imports de Cloud SQL

Buckets > backup-sql-anzeni 

Create folder Upload ▾ Transfer data ▾ Other services ▾

Filter by name prefix only ▾  Filter Filter objects and folders

<input type="checkbox"/>	Name	Size	Type	Created 
<input type="checkbox"/>	 Cloud_SQL_Export_2025-12-16 (1...	1.7 KB	application/x-sql	Dec 16, 2025, 11:15:18 AM

3. Tercera Parte: Compute Engine y Autoescalado

Esta parte consiste en crear una infraestructura de autoescalado con Nginx utilizando Managed Instance Groups.

3.1 Creación de la VM Base con Nginx

Se ha creado una máquina virtual base para servir como plantilla para el grupo de instancias.

Configuración:

1. Acceder a Compute Engine → VM instances
2. Crear una nueva instancia con nombre: **vm-nginx-base**
3. Machine type: **e2-micro** (configuración mínima)
4. Boot disk: Debian/Ubuntu
5. Firewall: Allow HTTP traffic

Instalación de Nginx:

Una vez creada la VM, se conectó por SSH y se ejecutaron los siguientes comandos:

```
sudo apt update && sudo apt install nginx -y && sudo systemctl enable nginx
```

3.2 Creación de la Custom Image

Se ha creado una imagen personalizada a partir del disco de la VM base para poder replicarla en el grupo de instancias.

Pasos realizados:

5. Detener la VM base (para consistencia del disco)
6. Ir a Compute Engine → Storage → Images
7. Crear imagen con nombre: **imagen-nginx**
8. Source: disco de la VM base

← Images Edit Delete Create instance Export	
✓ imagen-nginx	
Source disk	vm-nginx
Location	eu (European Union)
Architecture	x86/64
Labels	None
Tags ?	— ✎
Creation time	Dec 22, 2025, 1:49:29 PM UTC+01:00
Encryption type	Google-managed

[Equivalent REST](#)

3.3 Creación del Instance Template

Se ha creado una plantilla de instancia que utiliza la imagen personalizada.

Configuración:

- Nombre: **template-nginx**
- Machine type: **e2-micro** (mínima CPU y RAM)
- Boot disk: Custom image → imagen-nginx
- Firewall: Allow HTTP traffic

Basic information

Name	template-nginx
Type	Instance Template
Creation time	Dec 22, 2025, 1:56:33 PM UTC+01:00
In use by	mig-nginx
Location	europe-west1
Reservations	Automatically choose
Labels	None
Tags [?]	—
Placement policy	No policy [?]
Confidential VM service [?]	Disabled

Machine configuration

Machine type	e2-micro
Minimum CPU platform	None
Architecture	—
vCPUs to core ratio	—
Custom visible cores [?]	—
All-core turbo-only mode [?]	—
Display device	Disabled
GPUs	None

3.4 Creación del Managed Instance Group (MIG)

Se ha creado un grupo de instancias gestionado con política de autoescalado basado en consumo de CPU.

Configuración del MIG:

- Nombre: **mig-nginx**
- Instance template: template-nginx
- Location: Single zone

Configuración del Autoescalado:

- Minimum instances: **1**
- Maximum instances: **4**
- Target CPU utilization: **10%** (muy bajo para forzar escalado rápido)
- Initialization period (cooldown): **60 segundos**

Justificación del tiempo de enfriamiento:

El cooldown de 60 segundos es el tiempo que el autoescalador espera antes de considerar las métricas de una instancia recién creada. Este valor es suficiente para que Nginx arranque y las métricas de CPU se estabilicen, evitando decisiones erróneas basadas en picos temporales durante el arranque.

3.5 VM Independiente para Test de Autoescalado

Se ha creado una máquina virtual independiente con un script para probar el autoescalado del MIG.

Configuración de la VM:

- Nombre: **vm-atacante**
- Machine type: e2-micro
- Misma zona que el MIG

Script de test (test-autoescalado.sh):

```
#!/bin/bash
IP_MIG="[IP_DE_INSTANCIA_MIG]"
while true; do
  for i in {1..100}; do curl -s http://$IP_MIG > /dev/null & done
  sleep 0.5
done
```

3.6 Verificación del Autoescalado

Al ejecutar el script de carga, el MIG detectó el aumento de CPU y escaló automáticamente de 1 a 4 instancias (el máximo configurado).

Resultado: El autoescalado funcionó correctamente, pasando de 1 instancia a 4 instancias bajo carga.

← mig-nginx
Edit
Restart/replace VMs
Delete Group

Overview
Details
Monitoring
Errors

Instances by status
4 instances ⓘ
✓ 4

Instance by health ⓘ
Not configured
Autohealing off. [Configure](#)

Status	✓ Ready
Creation Time	Dec 22, 2025, 2:01:25 PM UTC+01:00
Description	
Target running size	4
Template	template-nginx (Regional)
Location	europa-west1-c
Resize requests	None ✎

VM instances
|| Suspend
■ Stop
▶ Start / Resume
⊖ Remove from group

Filter Enter property name or value

<input type="checkbox"/>	Status	Name ↑	Creation Time	Template
<input type="checkbox"/>	✓	mig-nginx-3s44	Jan 4, 2026, 8:29:07 PM UTC+01:00	template-nginx (Regional)
<input type="checkbox"/>	✓	mig-nginx-5bh3	Jan 4, 2026, 11:18:45 PM UTC+01:00	template-nginx (Regional)
<input type="checkbox"/>	✓	mig-nginx-87wf	Jan 4, 2026, 9:01:00 PM UTC+01:00	template-nginx (Regional)
<input type="checkbox"/>	✓	mig-nginx-ssxs	Jan 4, 2026, 11:50:30 PM UTC+01:00	template-nginx (Regional)

4. Cuarta Parte: App Engine

Esta parte consiste en desplegar una aplicación Python en App Engine conectada a Cloud SQL.

4.1 Preparación del Código

El código original del repositorio de Google estaba en Python 2.7 (ya no soportado), por lo que fue necesario adaptarlo a Python 3.

Cambios realizados:

- Migración de webapp2 a Flask
- Cambio de MySQLdb a PyMySQL
- Runtime actualizado a python312

Archivo app.yaml:

```
runtime: python312
env_variables:
  CLOUDSQL_CONNECTION_NAME:
"silver-fragment-481408-c9:europa-southwest1:anzeni-db"
  CLOUDSQL_USER: "alumno"
  CLOUDSQL_PASSWORD: "[password]"
  CLOUDSQL_DATABASE: "google"
```

4.2 Troubleshooting: Errores Encontrados y Soluciones

Durante el despliegue se encontraron varios errores que fueron resueltos:

Error 1: Runtime python39 no soportado

- **Error:** "Runtime python39 is end of support and no longer allowed"
- **Solución:** Cambiar runtime a python312 en app.yaml

Error 2: Bucket de staging sin permisos

- **Error:** "service account does not have access to the bucket staging.xxx.appspot.com"
- **Solución:**
 1. Ir a Cloud Storage → Buckets
 2. Seleccionar el bucket staging.silver-fragment-481408-c9.appspot.com
 3. En Permissions → Grant Access
 4. Añadir la service account con rol Storage Object Admin

Error 3: Connection Refused a Cloud SQL

- **Error:** "Can't connect to MySQL server - Connection refused"
- **Causa:** La service account de App Engine no tenía permisos para acceder a Cloud SQL
- **Solución:**
 5. Habilitar Cloud SQL Admin API: gcloud services enable sqladmin.googleapis.com
 6. Ir a IAM & Admin → IAM
 7. Editar la service account de App Engine
 8. Añadir el rol **Cloud SQL Client**

4.3 Deploy del Servicio Default

Una vez resueltos los errores, se desplegó la aplicación en el servicio default.

```
gcloud app deploy
```

URL: <https://silver-fragment-481408-c9.ew.r.appspot.com>

4.4 Deploy del Servicio "practica" con Versiones

Se creó un nuevo servicio llamado "practica" con dos versiones diferentes.

Modificación del app.yaml:

```
runtime: python312
service: practica
```

Deploy de versión 1.0.0:

```
gcloud app deploy --version=version-1-0-0
```

Deploy de versión 2.0.0:

```
gcloud app deploy --version=version-2-0-0
```

4.5 Configuración del Split de Tráfico

Se configuró la distribución del tráfico al 50% entre las dos versiones de forma aleatoria.

Configuración:

9. Ir a App Engine → Services → practica
10. Seleccionar ambas versiones
11. Clic en Split traffic
12. Split by: Random
13. version-1-0-0: 50%
14. version-2-0-0: 50%

Verificación:

```
$ gcloud app services describe practica
split:
  allocations:
    version-1-0-0: 0.5
    version-2-0-0: 0.5
  shardBy: RANDOM
```

Service practica					
Filter Filter versions					
<input type="checkbox"/>	Version	Status	Traffic Allocation	Instances [?]	Runtime
<input type="checkbox"/>	version-2-0-0	Serving	<div><div></div></div> 50%	0	python312
<input type="checkbox"/>	version-1-0-0	Serving	<div><div></div></div> 50%	0	python312

5. Diagrama de Arquitectura

A continuación se presenta el diagrama de arquitectura completo del proyecto, mostrando todos los componentes y sus relaciones. Está incluida una sección con la arquitectura del BONUS de práctica de despliegue en Terraform y sus componentes.

