



## PRÁCTICA

### Módulo de Migración a la nube - Google Cloud Platform

Yordany Fernández Padrón

En el presente documento se describen los pasos realizados para dar respuesta a la práctica del *Módulo de Migración a la nube*.

#### Primera parte

- Se creó el proyecto llamado “*Yordany KC Project*”
- Mi cuenta personal se quedó sin créditos por haberlos usado previamente por lo que tuve que utilizar otra cuenta. Se asignaron los permisos de propietario a mi cuenta personal y a la cuenta del profesor (o.martinezabella@gmail.com).
- Se crearon los siguientes avisos de facturación para el presupuesto mensual y trimestral respectivamente, para llevar un control del gasto del período de prueba:

<input type="checkbox"/>	Nombre de presupuesto ↑	Período del presupuesto	Tipo de presupuesto	Se aplica a	Activar alertas al alcanzar el
<input type="checkbox"/>	<a href="#">Presupuesto mensual</a>	Mensual	Importe especificado	Esta cuenta de facturación	50%, 75%, 90% y 100%
<input type="checkbox"/>	<a href="#">Presupuesto trimestral</a>	Trimestral	Importe especificado	Esta cuenta de facturación	50% y 90%

- Se realizó el diseño solicitado utilizando app.diagrams.net y se encuentra en la siguiente dirección:

<https://github.com/KeepCodingCloudDevops13/gcp>

#### Segunda parte

- Se creó la instancia de base de datos MySQL llamada *yordany-kc-project-mysql* con la siguiente configuración:



Región	us-central1 (Iowa)
Versión de la base de datos	MySQL 8.0.41
Tipo de máquina	db-perf-optimized-N-8
CPU virtuales	8 vCPU
RAM	64 GB
Caché de datos	Habilitada (375 GB)
Almacenamiento	100 GB SSD
Conexiones	IP pública
Copia de seguridad	Manual
Disponibilidad	Zona única

- Se realizó la configuración de las copias de seguridad automáticas a iniciar a las 12:00 pm, hora estándar de Montevideo, Uruguay (GMT-3) como se muestra en la siguiente imagen.

Nivel de copia de seguridad	Nivel de copia de seguridad estándar
Copias de seguridad automatizadas	Habilitada
Período de copias de seguridad	12:00 p.m. – 4:00 p.m. (GMT-3)
Se retuvieron las copias de seguridad automáticas	15
Recuperación de un momento determinado	Inhabilitada
Ubicación	Multirregión: predeterminada

- Se realizó la creación del nuevo usuario llamado alumno según lo indicado en la práctica y se realizó la conexión a ambas bases de datos usando dicho usuario.
- Se creó el bucket llamado *yordany\_kc\_bucket* y el directorio *Bkp\_MySQL* en el cual se almacenará los archivos en formato SQL exportados de las bases de datos.
- Se realizó la exportación de la base de datos “google” y “cloud” hacia el almacenamiento creado previamente, como se muestra en la siguiente imagen:



Navegador de carpetas

yordany\_kc\_bucket

Bkp\_MySQL/

Depósitos > yordany\_kc\_bucket > Bkp\_MySQL

Crear carpeta Subir Transferir los datos

Filtrar solo por prefijo de nombre Filtro Filtrar obje

Nombre	Tamaño
YFP_Cloud_SQL_Export_2026-01-1...	1.7 KB

- Se realizó la importación desde el archivo sql exportado previamente de forma satisfactoria.

Operaciones y registros			
Fecha/hora de creación	Hora de finalización	Tipo	Estado
11 ene 2026, 10:28:52 a.m.	11 ene 2026, 10:29:02 a.m.	Importar	Se completó correctamente la importación desde gs://yordany_kc_bucket/Bkp_MySQL/YFP_Cloud_SQL_Exp 01-11 (10:09:50).sql.
11 ene 2026, 10:10:53 a.m.	11 ene 2026, 10:11:04 a.m.	Exportar	Se completó correctamente la exportación a gs://yordany_kc_bucket/Bkp_MySQL/YFP_Cloud_SQL_Exp 01-11 (10:09:50).sql.

- Se realizaron los cambios de la instancia a la mínima configuración permitida de CPU y RAM. Para realizar dicha modificación hubo que habilitar el API de Compute Engine. La configuración quedó como se muestra a continuación:

Región	us-central1 (Iowa)
Versión de la base de datos	MySQL 8.0.41
Tipo de máquina	db-perf-optimized-N-2
CPU virtuales	2 vCPU
RAM	16 GB
Caché de datos	Habilitada (375 GB)
Almacenamiento	100 GB SSD
Conexiones	IP pública
Copia de seguridad	Automatizada
Disponibilidad	Zona única

### Tercera parte

- Se creó una instancia con SO Debian para la instalación de los paquetes necesarios con la que se crearía la imagen solicitada en la práctica. A continuación, los detalles de esta y la captura del nginx corriendo una vez instalado y configurado.



Nombre	yfp-instancia-base-nginx
ID de instancia	8303143363688664694
Descripción	Ninguna
Tipo	Instancia
Estado	✓ Activa
Fecha y hora de creación	ene 11, 2026, 11:15:55 a.m. UTC-03:00
Ubicación ?	us-central1-c
Imagen de origen del disco de arranque	<a href="#">debian-12-bookworm-v20251209</a>
Arquitectura del disco de arranque	X86_64

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*

- Posteriormente, se procedió a la creación de la imagen “yfp-imagen-nginx-v1” a partir del disco de la instancia previamente creada. Como buena práctica, la instancia se detuvo antes de realizar la imagen.

Estado	Nombre	Ubicación	Tamaño del archivo ?	Tamaño del disco
✓	<a href="#">yfp-imagen-nginx-v1</a>	us	1.01 GB	10 GB

- En este punto se creó una plantilla utilizando la imagen recién creada.

Nombre ↑	Tipo de máquina	Imagen	Tipo de disco	Ubicación ?
<a href="#">yfp-plantilla-nginx</a>	e2-micro	yfp-imagen-nginx-v1	Disco persistente balanceado	us-central1

- Se creó el grupo de instancias que usarán la plantilla creada.

Estado	Nombre ↑	Instancias	Plantilla	Tipo de grupo
✓	<a href="#">yfp-instance-group-nginx</a>	1	<a href="#">yfp-plantilla-nginx (Regional)</a>	Administrado



- Se creó un balanceador para permitir el acceso a único punto y que se realice el escalado correspondiente a partir de la configuración realizada para el auto esalado.

Nombre	Tipo de balanceador de cargas	Tipo de acceso	Protocolos
<a href="#">yfp-nginx-balanceador</a>	Aplicación	Externo	HTTP

- Se creó la instancia “*yfp-instance-autoescalado-test*” desde la cual se realizan las pruebas para verificar que se está realizando el escalado correctamente. Utilizando el siguiente script y subiendo la cantidad de workers (10, 20 y 100) logramos ver el correcto balanceo.

```
#!/bin/bash

# Dirección del balanceador
TARGET="http://35.209.0.17"

# Función para hacer peticiones en paralelo
make_requests() {
    while true; do
        curl -s -o /dev/null "$TARGET" &
        done
    }

# Número de procesos paralelos (ajusta según necesites más carga)
NUM_WORKERS=10

echo "Iniciando $NUM_WORKERS workers para generar carga..."

# Lanzar múltiples procesos en background
for i in $(seq 1 $NUM_WORKERS); do
    make_requests &
done

# Esperar indefinidamente
wait
```

Estado	Nombre ↑	Fecha/hora de creación	Plantilla	Zona
✓	<a href="#">yfp-instance-group-nginx-5crq</a>	ene 11, 2026, 2:56:06 p.m. UTC-03:00	<a href="#">yfp-plantilla-nginx (Regional)</a>	us-central1-f
✓	<a href="#">yfp-instance-group-nginx-gwhs</a>	ene 11, 2026, 11:58:59 a.m. UTC-03:00	<a href="#">yfp-plantilla-nginx (Regional)</a>	us-central1-c
✓	<a href="#">yfp-instance-group-nginx-t14b</a>	ene 11, 2026, 3:00:42 p.m. UTC-03:00	<a href="#">yfp-plantilla-nginx (Regional)</a>	us-central1-b

