

# Workshop GCP DESARROLLO

## Prerrequisitos (Instalar)

- 1.1. Python 3.10
- 1.2. Git local
- 1.3. Docker
- 1.4. Visual Code
- 1.5. Extension Cloud Code
- 1.6. SDK Cloud (<https://cloud.google.com/sdk/docs/install-sdk#windows>)

Primero comenzaremos por conocer kubernetes, desplegar una aplicación, luego pasaremos por spanner para conectar la aplicación django desplegada en kubernetes y por último gestionar los registros generados con las herramientas anteriores en cloud loggin.

## 1. Kurnet

### **Descripción**

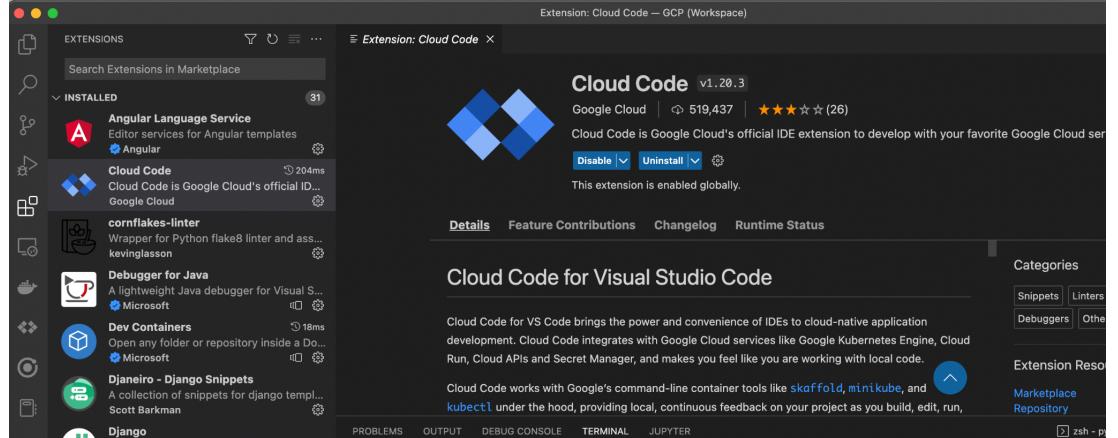
Google Kubernetes Engine (GKE) proporciona un entorno administrado para implementar, administrar y escalar las aplicaciones en contenedores mediante la infraestructura de Google. El entorno de GKE consta de varias máquinas (en particular, instancias de Compute Engine) que se agrupan para formar un clúster.

### **Características claves**

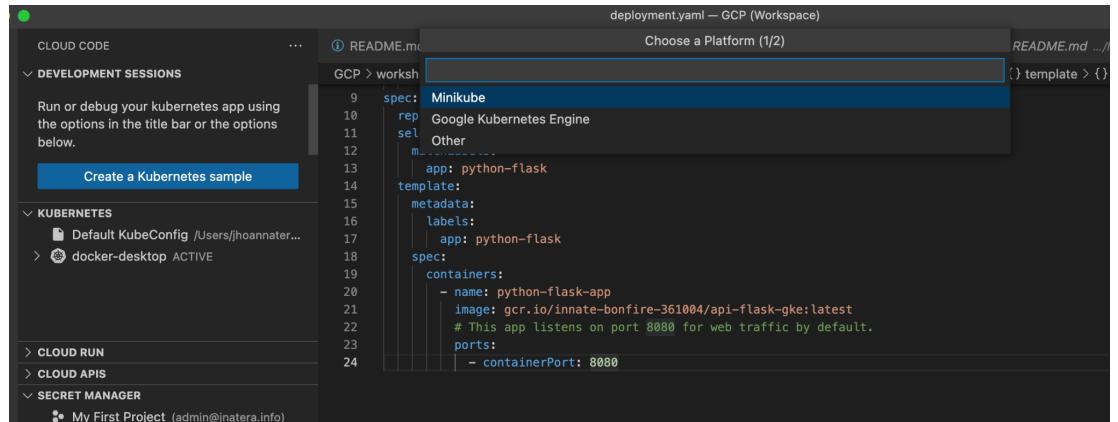
1. Ajuste de escala automático de Pods y clústeres
2. Aplicaciones y plantillas predefinidas de Kubernetes
3. Seguridad y redes nativas de contenedores
4. Migrar cargas de trabajo tradicionales a contenedores de GKE con facilidad
5. Administración de identidades y accesos
6. Supervisión y registros integrados.
7. Opciones de clústeres.
8. Ajuste de escala automático.
9. Aislamiento del contenedor.
10. Compatibilidad con imágenes de Docker.
11. Portabilidad de las cargas de trabajo locales y en la nube.
12. Balanceo de cargas global.
13. Canales de versiones.
14. Seguridad de la cadena de suministro del software
15. Facturación por segundo

## 1.1. Instalar de Cloud Code y Crear cluster en GCP

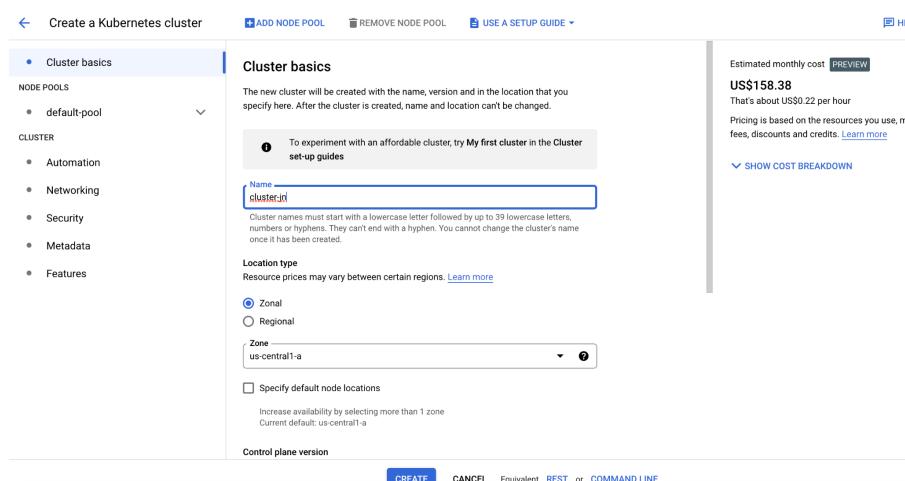
- Vamos a loguearnos en google desde la consola con gcloud auth login
- Instalamos la extensión Cloud Code, para administrar Kubernetes desde nuestro IDE Visual Code, para ello vamos al área de extensiones para proceder a instalar.



- Luego vamos a abrir esta extensión y presionamos botón de CREATE NEW CLUSTER y nos saldrá la siguiente imagen y luego seleccionamos Google Kubernetes Engine y abrirá la página para crear cluster en la nube.



- A continuación ingresamos el nombre del cluster y mantenemos el resto de información por defecto.



[Create a Kubernetes cluster](#)

[ADD NODE POOL](#) [REMOVE NODE POOL](#) [USE A SETUP GUIDE ▾](#)

- Cluster basics
- NODE POOLS
- default-pool

CLUSTER

- Automation
- Networking
- Security
- Metadata
- Features

Regional  
Zone — us-central1-a ?

Specify default node locations  
Increase availability by selecting more than 1 zone  
Current default: us-central1-a

Static version  
Manually manage the version upgrades. GKE will only upgrade the control plane and nodes if it's necessary to maintain security and compatibility, as described in the release schedule.  
[Learn more.](#)

Release channel  
Let GKE automatically manage the cluster's control plane version. [Learn more.](#)

Release channel — Regular channel (default)

Version — 1.22.12-gke.2300 (default)

These versions have passed internal validation and are considered production quality but don't have enough historical data to guarantee their stability. Known issues generally have known workarounds. [Release notes](#)

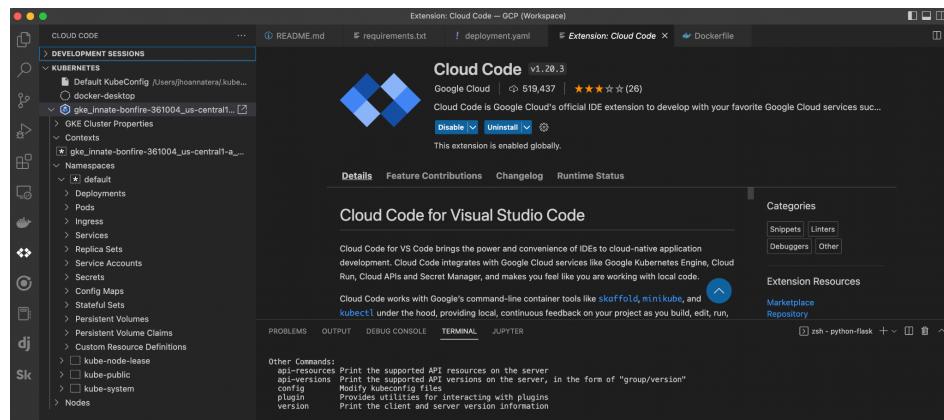
[CREATE](#) [CANCEL](#) Equivalent [REST](#) or [COMMAND LINE](#)

- Luego esperamos unos minutos a que se cree el nuevo cluster creado.

The screenshot shows the Google Cloud Platform interface for managing Kubernetes clusters. In the top navigation bar, 'My First Project' is selected. Below it, there are tabs for 'OVERVIEW', 'OBSERVABILITY', and 'COST OPTIMISATION'. The main table displays a single cluster entry:

Status	Name	Location	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
Up	cluster-jp	us-central1-a	3	0	0 GB	—	—

- Luego de terminar el proceso de creación del cluster vía web, le damos refresh en IDE para que aparezca el nuevo cluster creado, al desplegar podemos visualizar todas las opciones necesarias para administrar kubernetes y otros servicios relacionados.



- A continuación descargamos el repositorio para los códigos de prueba.  
`git clone https://github.com/jnatera/workshop-izzi-dev.git`

- Luego nos ubicamos en la carpeta “Kubernetes/python-flask”
- Vamos a comprobar que la aplicación de prueba funciona corriendo en docker local

```
docker build -t kube-app .

docker run -e PORT=8080 -p 8080:8080 kube-app
```

- Compilamos la imagen en la nube

```
gcloud builds submit --tag
gcr.io/$GOOGLE_CLOUD_PROJECT/python-flask-nombre
```

- Mantener seleccionado el cluster de gcp en vez del que se encuentra por defecto en la extensión cloud code “Desktop”

## 1.2. Creación de un Deployment

- Primero adaptamos el nombre del proyecto en el archivo yaml para que se registre la imagen gke.
- Nos ubicamos en la carpeta python-flask
- Compilamos la imagen con el siguiente comando

```
gcloud builds submit --tag
gcr.io/$GOOGLE_CLOUD_PROJECT/python-flask-nombre-gke .
```

- Cambiamos el nombre del proyecto en el archivo deployment.yaml para poderlo aplicar.

```
image: gcr.io/cambiar-aqui/python-flask-nombre-gke:latest
```

- Ejecutamos el siguiente comando para aplicar los cambios en un deployment.

```
kubectl apply -f gke/dev/1-deployment.yaml
```

- Con el siguiente comando podemos verificar si el deployment se encuentra ready. El parámetro -w permite esperar que exista cambios en el cluster para refrescar la consola.

```
kubectl get deployments -w
```

## 1.3. Creación de Services

- Luego que se encuentre el deployment generado, procedemos a crear el services para que pueda exponerse el servicio hacia el exterior con una ip pública en este caso.
- Ejecutamos el siguiente comando para aplicar los cambios.

```
kubectl apply -f gke/dev/2-service.yaml
```

- Ahora confirmamos que se encuentra desplegado el servicio y con una ip pública generada.

```
kubectl get services
```

## 1.4. Creación de Ingress

- Ejecutamos el comando para aplicar cambios de ingress

```
kubectl apply -f gke/dev/3-ingress.yaml
```

## 1.5. Uso de Skaffold

Skaffold es una herramienta de línea de comandos que permite el desarrollo continuo para aplicaciones nativas de Kubernetes. Puedes usar Skaffold a fin de configurar un lugar de trabajo de desarrollo local para usarlo con las canalizaciones de entrega continua de Google Cloud Deploy.

Estas son algunas de las ventajas que proporciona Skaffold cuando se usa con Google Cloud Deploy:

- **Integración sencilla**

Comienza con un bucle de desarrollo local. Puedes compartir el archivo skaffold.yaml con tu equipo, lo que puede ayudar a que los miembros nuevos del equipo sean coherentes.

- **Control coherente** sobre la renderización en diversos destinos de implementación

Puedes usar perfiles de Skaffold, con implementación y renderización diferentes para distintos destinos.

- **Elección de herramientas de renderización** sin comprometer cómo se definen las canalizaciones de entrega

El uso de Skaffold permite que Google Cloud Deploy separe la definición de la canalización de entrega de los detalles de procesamiento. Esta separación te permite experimentar con tus manifiestos sin interrumpir la canalización de entrega.

- **Proceso de renderización coherente**

La fuente de renderización y los contenedores se usan para generar manifiestos renderizados.

- **Verificaciones de estado de la implementación**

Skaffold las realiza y Google Cloud Deploy las usa.

- Nos vamos a la carpeta hello-work
- Cambiamos el nombre del proyecto en los archivos yaml que corresponda.
- Ejecutamos el siguiente comando para que realice todo el trabajo configurado incluyendo compilar la imagen y de esta forma nos evitamos el conjunto de pasos

```
skaffold run -f skaffold.yaml
```

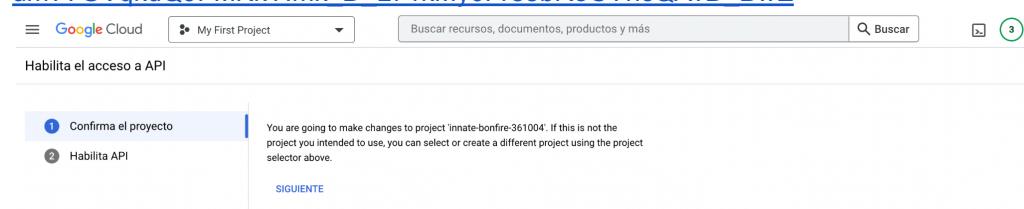
## 2. Cloud SQL Spanner

Cloud Spanner es un servicio de bases de datos relacionales esencial y completamente administrado que ofrece coherencia en las transacciones a escala global, replicación síncrona y automática con alta disponibilidad y compatibilidad con dos dialectos de SQL: SQL estándar de Google (ANSI 2011 con extensiones) y PostgreSQL.

En este apartado, se muestra cómo realizar operaciones básicas en Cloud Spanner con Google Cloud Console. A continuación presentaremos los siguientes puntos:

- Crear una instancia, una base de datos y una tabla de Spanner
  - Agregar un esquema
  - Escribir y modificar datos
  - Ejecutar una consulta
- 
- Mantener la facturación habilitada y un proyecto seleccionado para continuar.
  - Habilitamos la API de Cloud Spanner

[https://console.cloud.google.com/flows/enableapi?apiid=spanner.googleapis.com&hl=es&\\_ga=2.258030858.1691381217.1665388043-1415742975.1661315684&\\_gac=1.93462895.1662053724.CjwKCAjwsMGYBhAEEiwAGUXJaVU0J5u\\_WgqHUO8DWy2umvYGVqkdQ9FMKkYImk\\_D\\_L71kMy0Fre8bRoCYhcQAvD\\_BwE](https://console.cloud.google.com/flows/enableapi?apiid=spanner.googleapis.com&hl=es&_ga=2.258030858.1691381217.1665388043-1415742975.1661315684&_gac=1.93462895.1662053724.CjwKCAjwsMGYBhAEEiwAGUXJaVU0J5u_WgqHUO8DWy2umvYGVqkdQ9FMKkYImk_D_L71kMy0Fre8bRoCYhcQAvD_BwE)



Google Cloud My First Project Buscar recursos, documentos, productos y más Buscar

Habilita el acceso a API

1 Confirma el proyecto Estás a punto de habilitar "Cloud Spanner API".

2 Habilita API HABILITAR

- Debemos crear una instancia, que es una asignación de recursos que usan las bases de datos de Spanner en esa instancia.  
[https://console.cloud.google.com/spanner/instances?hl=es&\\_ga=2.47929767.1691381217.1665388043-1415742975.1661315684&\\_gac=1.156451657.1662053724.CjwKCAjwsMGYBhAEEiwAGUXJaVU0J5u\\_WgqHUO8DWy2umvYGVqkdQ9FMKkYImkD\\_L71kMy0Fre8bRoCYhcQAvD\\_BwE](https://console.cloud.google.com/spanner/instances?hl=es&_ga=2.47929767.1691381217.1665388043-1415742975.1661315684&_gac=1.156451657.1662053724.CjwKCAjwsMGYBhAEEiwAGUXJaVU0J5u_WgqHUO8DWy2umvYGVqkdQ9FMKkYImkD_L71kMy0Fre8bRoCYhcQAvD_BwE)
- Haga clic en **Crear instancia**.
- En el nombre de la instancia, ingresa test-mi-nombre.
- En el ID de la instancia, ingresa test-mi-nombr.
- Usa una configuración **Regional**.
- Elige cualquier configuración regional en el menú desplegable. La configuración de tu instancia determina la ubicación geográfica donde tus instancias se almacenan y se replican.
- Usa la capacidad de procesamiento predeterminada de 1,000 unidades de procesamiento.
- Haga clic en **Crear**.

Google Cloud My First Project Buscar recursos, documentos, productos

## Te damos la bienvenida a Spanner

Spanner es una base de datos relacional completamente administrada que ofrece coherencia en las transacciones y alta disponibilidad a escala global. Para usar Spanner, crea una instancia dentro de tu proyecto y, luego, configura el entorno de desarrollo a fin de acceder a Spanner para agregar datos. [Más información](#)

INICIAR PRUEBA GRATUITA

CREAR UNA INSTANCIA APROVISIONADA

Pon Spanner en funcionamiento

1 Crear una instancia

Obt

Google Cloud My First Project Personalizar y controlar Google Chrome

[Crear una instancia](#)

**Ingresar un nombre para la instancia**  
Las instancias tienen un nombre y un ID. El nombre se usa solo con fines de visualización. El ID es un identificador permanente y único.

Nombre de la instancia \* test-mi-nombre  
El nombre debe tener entre 4 y 30 caracteres.

ID de instancia \* test-mi-nombre  
Se permiten letras minúsculas, números y guiones.

**Elige una configuración**  
Determina la ubicación de los nodos y datos. Afecta el costo, el rendimiento y la replicación. Una configuración multirregional seleccionará la región líder predeterminada para las réplicas líderes. Puedes cambiar la región líder en cualquier momento con una declaración DDL. [Más información](#)

[COMPARAR CONFIGURACIONES DE REGIONES](#)

Regional  
 Multirregional

us-central1 (Iowa)

**Asignar capacidad de procesamiento**  
La capacidad de procesamiento determina la cantidad de capacidad de procesamiento de datos, las consultas por segundo (QPS) y los límites de almacenamiento en tu instancia. Un nodo equivale a 1,000 unidades de procesamiento. Esta opción modifica la facturación.

Unidad \* Unidades de procesamiento... Cantidad \* 1000  
Solo números enteros. Ingresa incrementos de 100 hasta 1,000, seguidos de incrementos de 1,000.

✓ ORIENTACIÓN SOBRE LA CAPACIDAD DE PROCESAMIENTO

[CREAR](#) [CANCELAR](#)

Google Cloud My First Project Buscar recursos, documentos, productos y más

Spanner Todas las instancias > INSTANCIA test-mi-nombre: Descripción general EDITAR INSTANCIA BORRAR INSTANCIA LEARN MOSTRAR PANEL DE INFORMACIÓN

**Descripción general**

Nombre	test-mi-nombre	Operaciones	Rendimiento	Almacenamiento total de la base de datos
ID	test-mi-nombre	Operaciones de lectura: -/s	Operaciones de lectura: -/s	0 B / 4 TB
Configuración	us-central1 (Iowa)	Operaciones de escritura: -/s	Operaciones de escritura: -/s	
Capacidad de procesamiento	Unidades de procesamiento: 1,000	Nodos: 1		
Uso de CPU (media)	-			

Bases de datos [CREAR BASE DE DATOS](#) ACTUALIZAR

Filtro Filtrar bases de datos

Nombre	Dialecto	Uso de CPU	Tamaño	Período de retención de la versión

- **Crear una base de datos**
- Haz clic en la instancia **Instancia de prueba**.
- Haga clic en **Crear base de datos**.
- En el nombre de la base de datos, ingresa example-db-mi-nombre.
- Selecciona un dialecto de base de datos.

- Selecciona SQL estándar de Google, deja el campo de texto **Define tu esquema** (Define your schema) por ahora. Definirás tu esquema en la siguiente sección de esta guía de inicio rápido.
- Haga clic en **Crear**.  
Google Cloud Console muestra la página **Descripción general** de la base de datos que creaste.

**Asigna un nombre a la base de datos**  
Ingresa un nombre permanente para tu base de datos. Debe tener al menos dos caracteres y empezar con una letra.

Nombre de la base de datos \*  
example-db-mi-nombre

Se permiten letras minúsculas, números, guiones y guiones bajos

**Selecciona el dialecto de la base de datos**  
Elige entre los dialectos SQL estándar de Google y PostgreSQL para tu base de datos de Spanner.

SQL estándar de Google  
 PostgreSQL

**Define tu esquema (opcional)**  
A continuación, agrega las instrucciones de SQL del lenguaje de definición de datos para Spanner. Separa las sentencias con un punto y coma. [Más información](#)

PLANTILLAS DE DDL ▾      ACCESOS DIRECTOS

Presiona Alt + F1 para ver las opciones de accesibilidad.

1

▼ MOSTRAR OPCIONES DE ENcriptación

**CREAR**      CANCELAR

## Cree un esquema para su base de datos

- Haz clic en **Crear tabla** al comienzo de la lista de tablas.  
Google Cloud Console muestra la página Escribir declaraciones DDL.
- Ingrresa:

```
CREATE TABLE Singers (
    SingerId    INT64 NOT NULL,
    FirstName   STRING(1024),
    LastName    STRING(1024),
```

```

SingerInfo BYTES (MAX) ,
BirthDate DATE,
) PRIMARY KEY (SingerId);

```

← Escribe declaraciones DDL

### Define el esquema example-db-mi-nombre

Usa el lenguaje de definición de datos (DDL) de Cloud Spanner para definir el esquema de tu instancia. Las declaraciones DDL te permiten modificar una base de datos; crear, modificar o descartar tablas en una base de datos, y crear o descartar índices en una base de datos. Para escribir varias declaraciones, sepáralas con punto y coma. [Más información](#)

Dialecto de la base de datos SQL estándar de Google

PLANTILLAS DE DDL ▾ ACCESOS DIRECTOS

Presiona Alt + F1 para ver las opciones de accesibilidad.

```

1 CREATE TABLE Singers (
2   SingerId INT64 NOT NULL,
3   FirstName STRING(1024),
4   LastName STRING(1024),
5   SingerInfo BYTES(MAX),
6   BirthDate DATE,
7 ) PRIMARY KEY (SingerId);

```

**ENVIAR** CANCELAR

Todas las instancias > INSTANCIA test-mi-nombre: Descripción general > BASE DE DATOS DE SQL ESTÁNDAR DE GOOGLE example-db-mi-nombre: Descripción general

**ESCRIBIR DDL**



Nombre de la base de datos example-db-mi-nombre

Dialecto de la base de datos SQL estándar de Google

Tipo de encriptación Administrada por Google

Actualizaciones del esquema  Actualización completada  
Para ver todas las actualizaciones, visita [Cloud Logging](#).

Uso de CPU (media)

2.52 %

Operaciones

Operaciones de lectura: 0.00/s  
Operaciones de escritura: -/s

Rendimiento

Operaciones de lectura: -/s  
Operaciones de escritura: -/s

Abrir

0 B

**TABLAS**

VISTAS

Las tablas están estructuradas con filas, columnas y valores, y contienen índices y claves primarias. La relación superior-secundario entre las tablas se puede definir mediante la intercalación de tablas o las claves externas. [Más información](#)

+ CREAR TABLA

Filtro Filtrar tablas

Nombre ↑

Índices

Intercalado en

Supervisada por

Singers

-

-

## Insertar datos

- En la página de la base de datos, haz clic en la tabla Singers en la lista de tablas.

Google Cloud Console muestra la página **Esquema (Table)** de la tabla Singers.

- En el menú de navegación, haz clic en **Datos** para mostrar la página **Datos** de la tabla Singers.
- Haga clic en **Insert**.

Google Cloud Console muestra la página **Consulta** de la tabla Singers con una pestaña de consulta nueva que contiene las declaraciones **INSERT** y **SELECT** de la plantilla que editas para insertar una fila en la tabla Singers y ver el resultado de esa inserción:

### 1 fila insertadas

[Ver todos los datos en la tabla](#)

Esta declaración insertadas 1 fila y no mostró ninguna fila.

■ Haga clic en **Ejecutar**.

Todas las instancias > INSTANCIA test-mi-nombre: Descripción general > BASE DE DATOS DE SQL ESTÁ example-db-mi-nombre: Descripción general

**Datos**

**INSERTAR** EDITAR BORRAR

Filtro Filtra datos

	SingerId	FirstName	LastName	SingerInfo	BirthDate
<input type="checkbox"/>	1	Marc	Richards		

```
-- Add new values in the VALUES clause in order of the column list.
-- Each value must be type compatible with its associated column.
INSERT INTO
  Singers (SingerId,
            BirthDate,
            FirstName,
            LastName,
            SingerInfo)
VALUES
  (3, -- type: INT64
   NULL, -- type: DATE
   'Kena', -- type: STRING(1024)
   '', -- type: STRING(1024)
   NULL -- type: BYTES(MAX)
  );
-- Change values in the WHERE condition to match the inserted row.
SELECT
  *
FROM
  Singers
WHERE
  SingerId=3;
```

er	Todas las instancias > INSTANCIA test-mi-nombre: Descripción general > BASE DE DATOS DE SQL ESTÁNDAR DE GOOGLE example-db-mi-nombre: Descripción general >																		
	<p><b>Datos</b></p> <p><b>INSERTAR</b>   <b>EDITAR</b>   <b>BORRAR</b></p> <p><b>Filtro</b> Filtra datos</p> <table border="1"> <thead> <tr> <th><input type="checkbox"/></th> <th>SingerId</th> <th>FirstName</th> <th>LastName</th> <th>SingerInfo</th> <th>BirthDate</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>1</td> <td>Marc</td> <td>Richards</td> <td></td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td>2</td> <td>Catalina</td> <td>Smith</td> <td></td> <td></td> </tr> </tbody> </table>	<input type="checkbox"/>	SingerId	FirstName	LastName	SingerInfo	BirthDate	<input type="checkbox"/>	1	Marc	Richards			<input type="checkbox"/>	2	Catalina	Smith		
<input type="checkbox"/>	SingerId	FirstName	LastName	SingerInfo	BirthDate														
<input type="checkbox"/>	1	Marc	Richards																
<input type="checkbox"/>	2	Catalina	Smith																

## ■ EDITAR DATOS

```
-- Change values in the SET clause to update the row where the WHERE
condition is true.

UPDATE
    Singers
SET
    BirthDate='',
    FirstName='Kena',
    LastName='',
    SingerInfo=''
WHERE
    SingerId=3;
SELECT
    *
FROM
    Singers
WHERE
    SingerId=3;
```

## ■ Actualizar datos

```
-- Change values in the SET clause to update the row where the WHERE
condition is true.

UPDATE
    Singers
SET
    BirthDate='1961-04-01'
WHERE
    SingerId=3;
SELECT
    *
FROM
```

```

Singers
WHERE
SingerId=3;

```

- Ejecutar consulta

```
SELECT * FROM Singers;
```

- Creamos una tabla intercalada

```

CREATE TABLE Albums (
    SingerId      INT64 NOT NULL,
    AlbumId       INT64 NOT NULL,
    AlbumTitle    STRING(MAX),
) PRIMARY KEY (SingerId, AlbumId),
INTERLEAVE IN PARENT Singers ON DELETE CASCADE;

```

```

CREATE TABLE Songs (
    SingerId      INT64 NOT NULL,
    AlbumId       INT64 NOT NULL,
    TrackId       INT64 NOT NULL,
    SongName      STRING(MAX),
) PRIMARY KEY (SingerId, AlbumId, TrackId),
INTERLEAVE IN PARENT Albums ON DELETE CASCADE;

```

Singers(1)	"Marc"	"Richards"	<Bytes>		
Albums(1, 1)				"Total Junk"	
Albums(1, 2)				"Go, Go, Go"	
Songs(1, 2, 1)					"42"
Songs(1, 2, 2)					"Nothing Is The Same"
Singers(2)	"Catalina"	"Smith"	<Bytes>		
Albums(2, 1)				"Green"	
Songs(2, 1, 1)					"Let's Get Back Together"
Songs(2, 1, 2)					"Starting Again"
Songs(2, 1, 3)					"I Knew You Were Magic"
Albums(2, 2)				"Forever Hold Your Peace"	
Albums(2, 3)				"Terrified"	
Songs(2, 3, 1)					"Fight Story"

- Proceder a borrar instancia para limpieza
- Eliminar instancias.

## 3. Spanner con Kubernetes

A continuación vamos a configurar

## 4. Cloud Logging

### 4.1. ¿Qué es?

Cloud Logging es un servicio completamente administrado que te permite almacenar, buscar, analizar, supervisar y generar alertas sobre los datos y eventos de registro desde Google Cloud y Amazon Web Services. Puedes recopilar datos de registro de más de 150 componentes de aplicación comunes, sistemas locales y sistemas de nube híbrida.

Logging incluye almacenamiento para registros a través de [buckets de registro](#), una interfaz de usuario llamada [Explorador de registros](#) y [una API](#) para administrar los registros de manera programática. Logging te permite leer y escribir entradas de registro, consultar tus registros y controlar cómo se enrutan y usan los registros.

### Pruebas manuales de logs.

- Escribamos un registro no estructurado en los logs para probar logging.

```
gcloud logging write my-test-log "A simple entry."
```

- Guardaremos un registro estructurado

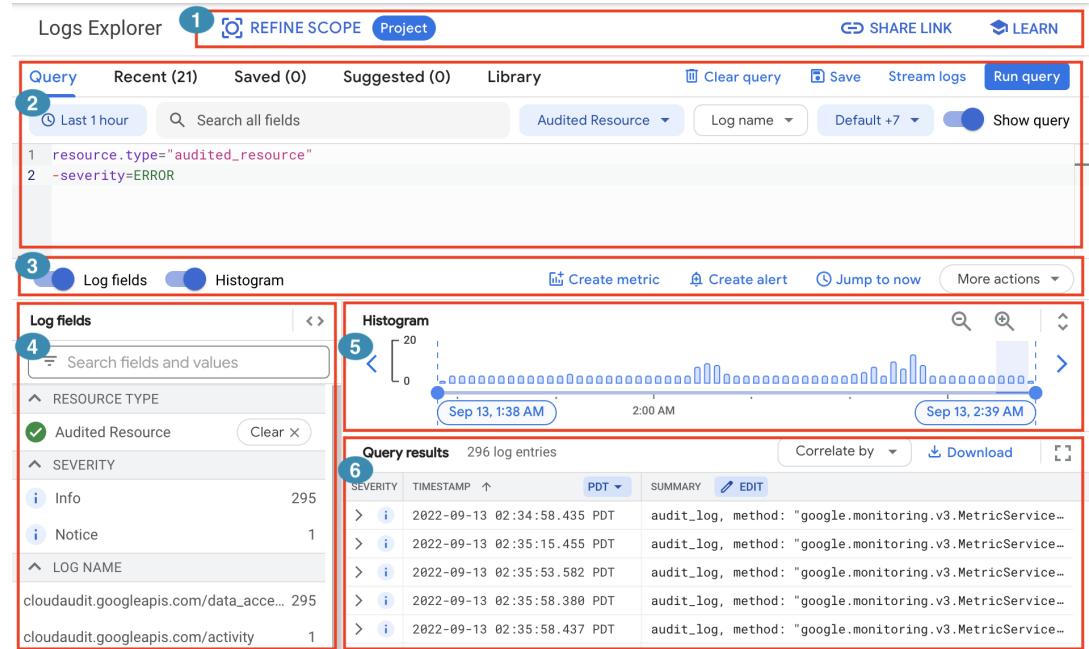
```
gcloud logging write --payload-type=json my-test-log '{ "message": "My second entry", "weather": "partly cloudy"}'
```

- Revisamos desde consola los registros guardados

```
gcloud logging read "resource.type=global"
```

### 4.2. Explorar registros

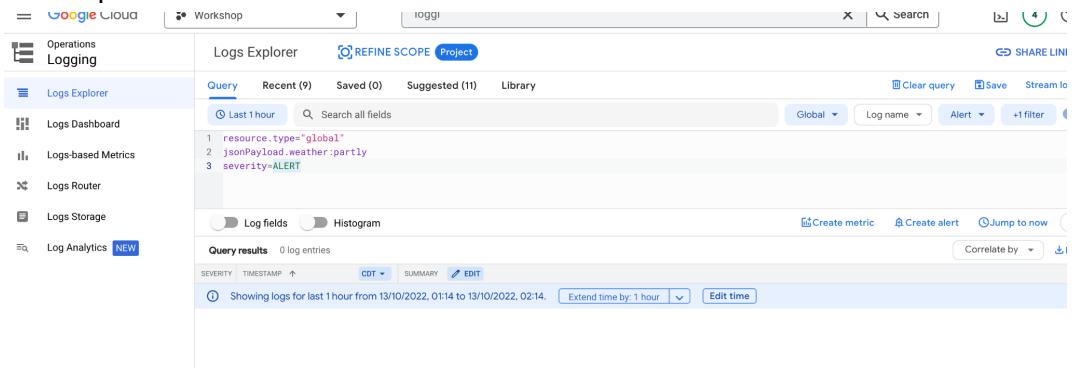
El explorador se divide en las siguientes partes a la vista.



- Para mostrar todas las entradas de registro con datos estructurados que tienen una clave de weather en la que el campo value contiene partly, haz lo siguiente:

```
jsonPayload.weather:partly
```

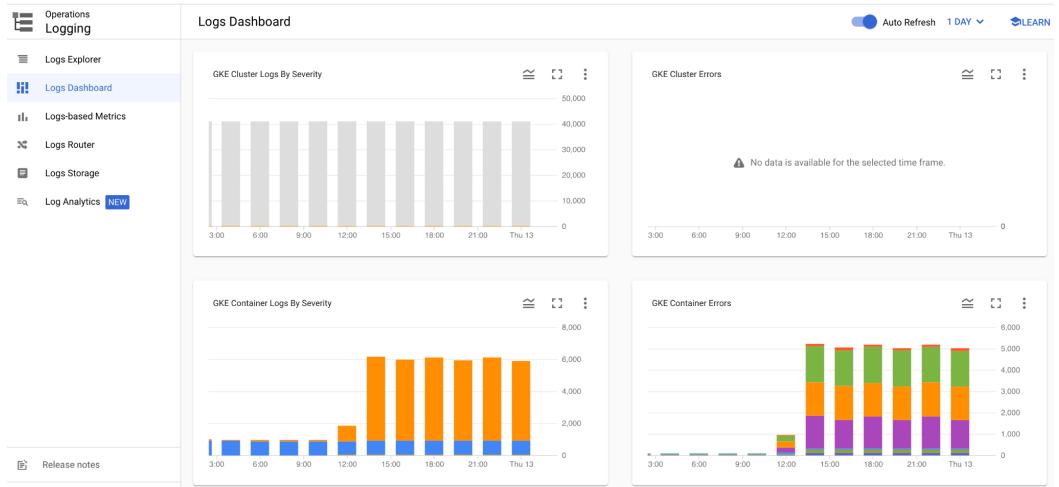
- En la interfaz se vería de la siguiente manera incluyendo varios parámetros de búsqueda.



- Si deseamos eliminar un registro creado previamente utilizaremos el siguiente comando

```
gcloud logging logs delete my-test-log
```

- A continuación veremos un ejemplo de como se ve el tablero por defecto que contiene el explorador de registros logging, donde podemos ver contadores generales de tipo de alertas.



- 
- 

## 4.3. Enrutador de registros.

Enabled	Type	Name	Description	Destination	Created	Last updated
<input checked="" type="checkbox"/>	Cloud Logging bucket	_Default		logging.googleapis.com/projects/inname-bonfire-361004/locations/global/buckets/_Default		
<input checked="" type="checkbox"/>	Cloud Logging bucket	_Required		logging.googleapis.com/projects/inname-bonfire-361004/locations/global/buckets/_Required		
<input checked="" type="checkbox"/>	Cloud Logging bucket	refwr	rr	logging.googleapis.com/projects/inname-bonfire-361004/locations/global/buckets/_Default	2022-10-13 02:08:39.003 CDT	2022-10-13 02:09:44