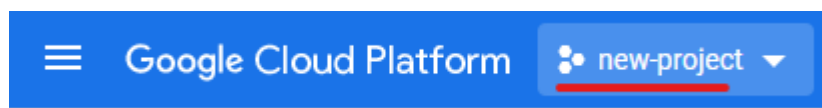


Guía de Configuración y Despliegue de NodeJS en App Engine

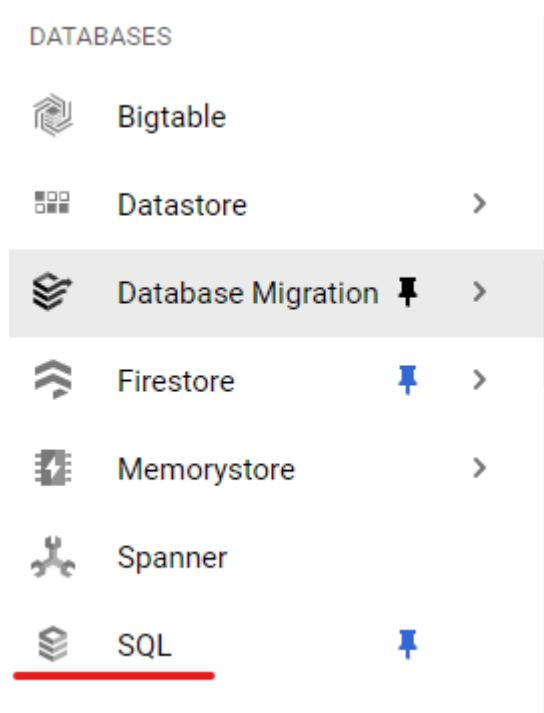
Configuración de CloudSQL

CloudSQL es una herramienta de Google Cloud Platform que provee servidores de bases de datos administrados. Te dan la facilidad de gestionar tus bases de datos sin necesidad de configurar un servidor, y se puede encargar de operaciones como backups, escalamiento automático, creación de usuarios y bases de datos, etc.

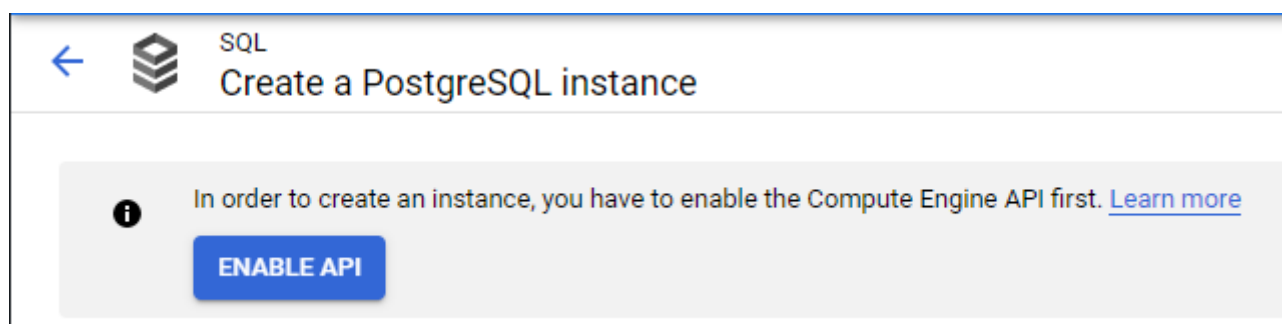
Navegar a [Google Cloud Platform](#). Seleccionar el proyecto a utilizar:



En la barra lateral, navegar a Cloud SQL



Habilitar Compute Engine API



Crear una instancia. Se puede utilizar MySQL, Postgres, o SQLServer, la configuración es similar en todas. Para la información básica de la instancia:

- Ingresar un nombre descriptivo de la instancia
- Auto-generar o crear una contraseña segura. Esta va a ser la contraseña del usuario *root* de la DB.
- Seleccionar una region. Esta es la ubicación física del servidor donde se va a alojar la DB. Hay que procurar que sea en una región de US, y que sea la misma o una cercana a la región de App Engine, para disminuir la latencia.
- Seleccionar "Single Zone". Para producción, se recomienda "Multiple Zones", ya que replica tu base de datos en otra zona si es que llega a haber problemas en la zona original. En este caso, por el tipo de proyecto, y para reducir costos, elegimos "Single Zone".

← Create a PostgreSQL instance

Instance info

Instance ID *

project-management

Use lowercase letters, numbers, and hyphens. Start with a letter.

Password *

.....



GENERATE

Database version *

PostgreSQL 12



Choose region and zonal availability

For better performance, keep your data close to the services that need it. Region is permanent, while zone can be changed any time.

Region

us-central1 (Iowa)



☒ Single zone

In case of outage, no failover. Not recommended for production.

☐ Multiple zones (Highly available)

Automatic failover to another zone within your selected region. Recommended for production instances. Increases cost.

Configurar los recursos de cómputo que se le van a dar a la instancia. Para este proyecto, 1 core de CPU y 3.75GB de RAM (La configuración mínima en GCP), va a ser suficiente, y nos ayuda a mantener los costos bajos.

Machine type



Machine Type

Choose a preset or customize your own. For better performance, choose a machine type with enough memory to hold your largest table.

Lightweight

☒ 1 vCPU, 3.75 GB

☐ 2 vCPU, 3.75 GB

☐ 4 vCPU, 3.75 GB

☐ Custom

Configurar el almacenamiento asignado a la instancia. Se recomienda un SSD, con sólo 10GB de almacenamiento, ya que el volumen estimado de registros en la base de datos es bajo.

Storage



Storage type

Choice is permanent. Storage type affects performance.

☒ SSD (Recommended)
Most popular choice. Lower latency than HDD with higher QPS and data throughput.

☐ HDD
Lower performance than SSD with lower storage rates.

Storage capacity

10 - 30,720 GB. Higher capacity improves performance, up to the limits set by the machine type. Capacity can't be decreased later.

☒ 10 GB

☐ 20 GB

☐ 100 GB

☐ 200 GB

☐ Custom

☐ Enable automatic storage increases
If enabled, whenever you are nearing capacity, storage will be incrementally (and permanently) increased. [Learn more](#)

Configurar la conectividad de la instancia. Permitir que se conecte por IP pública, y autorizar la red '0.0.0.0/0'.

Nota: Esta configuración permite que se pueda conectar a la DB desde cualquier dispositivo en cualquier red, si se tienen las credenciales correctas. No es recomendado para producción, pero con motivos de simplificar la conexión durante el proyecto, lo podemos dejar así.

Connections

Choose a network path for connecting to this instance. For extra security, consider using the Cloud SQL proxy. [Learn more](#)



Private IP

Requires additional APIs and permissions, which may require your system admin. Can't be disabled once enabled. [Learn more](#)



Public IP

Authorize a network or use [Cloud SQL Proxy](#) to connect to this instance. [Learn more](#)



You have added 0.0.0.0/0 as an allowed network. This prefix will allow any IPv4 client to pass the network firewall and make login attempts to your instance, including clients you did not intend to allow. Clients still need valid credentials to successfully log in to your instance.

Authorized networks

default (0.0.0.0/0)

(Not saved)



[ADD NETWORK](#)

El resumen de la instancia debería de verse similar a este:

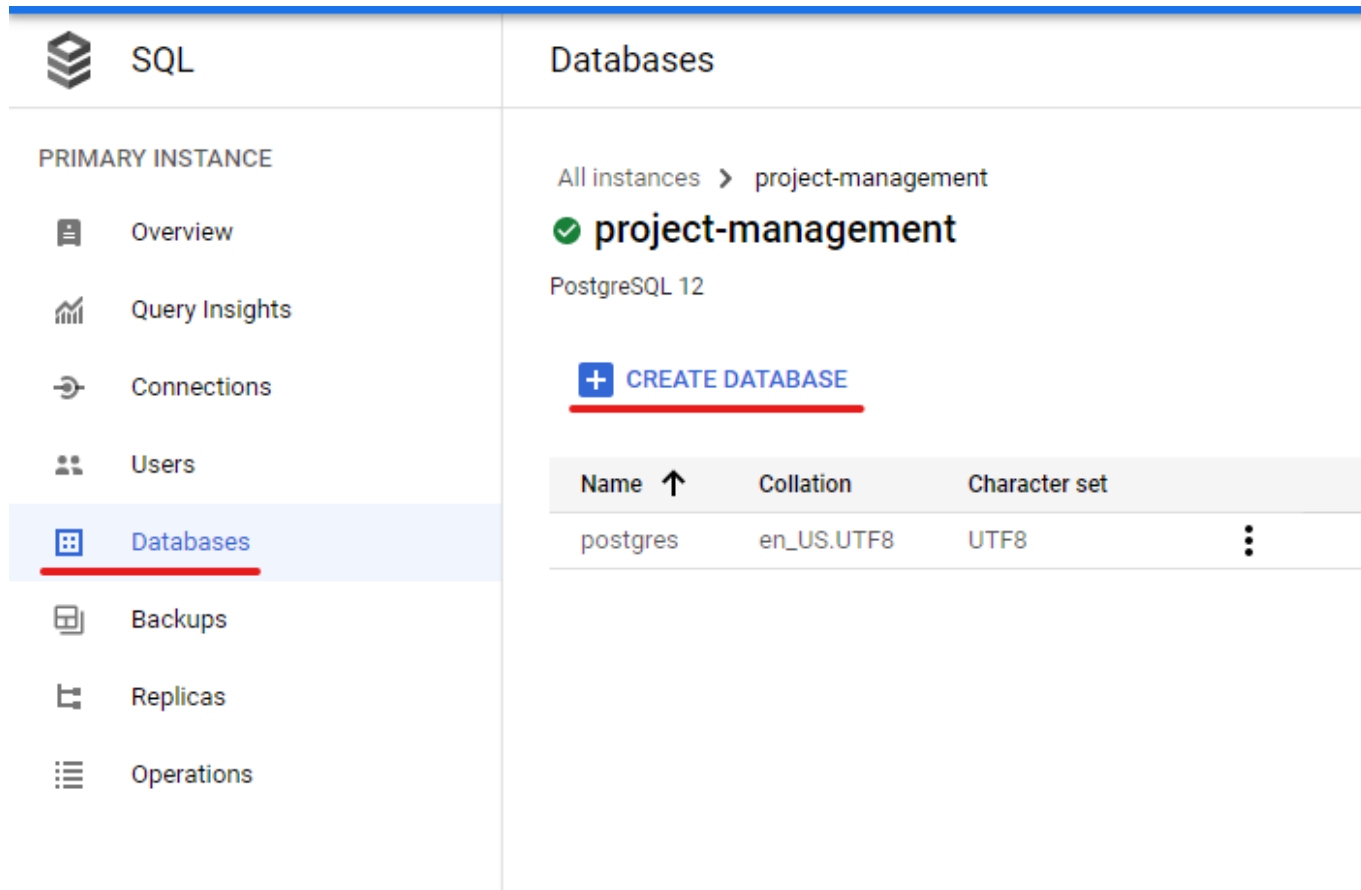
Summary

| | |
|---|---------------------|
| Region | us-central1 (Iowa) |
| DB Version | PostgreSQL 12 |
| vCPUs | 1 vCPU |
| Memory | 3.75 GB |
| Storage | 10 GB |
| Network throughput (MB/s)  | 250 of 2,000 |
| Disk throughput (MB/s)  | Read: 4.8 of 240.0 |
| | Write: 4.8 of 72.0 |
| IOPS  | Read: 300 of 15,000 |
| | Write: 300 of 4,500 |
| Connections | Public IP |
| Backup | Manual |
| Availability | Single zone |
| Point-in-time recovery | Disabled |

Con la configuración lista, seleccionar el botón "Create Instance". La creación de la instancia puede tardar varios minutos.

Con esto, tenemos una instancia, es decir, un servidor de base de datos, en el que podemos crear la base de datos que se va a utilizar en el proyecto.

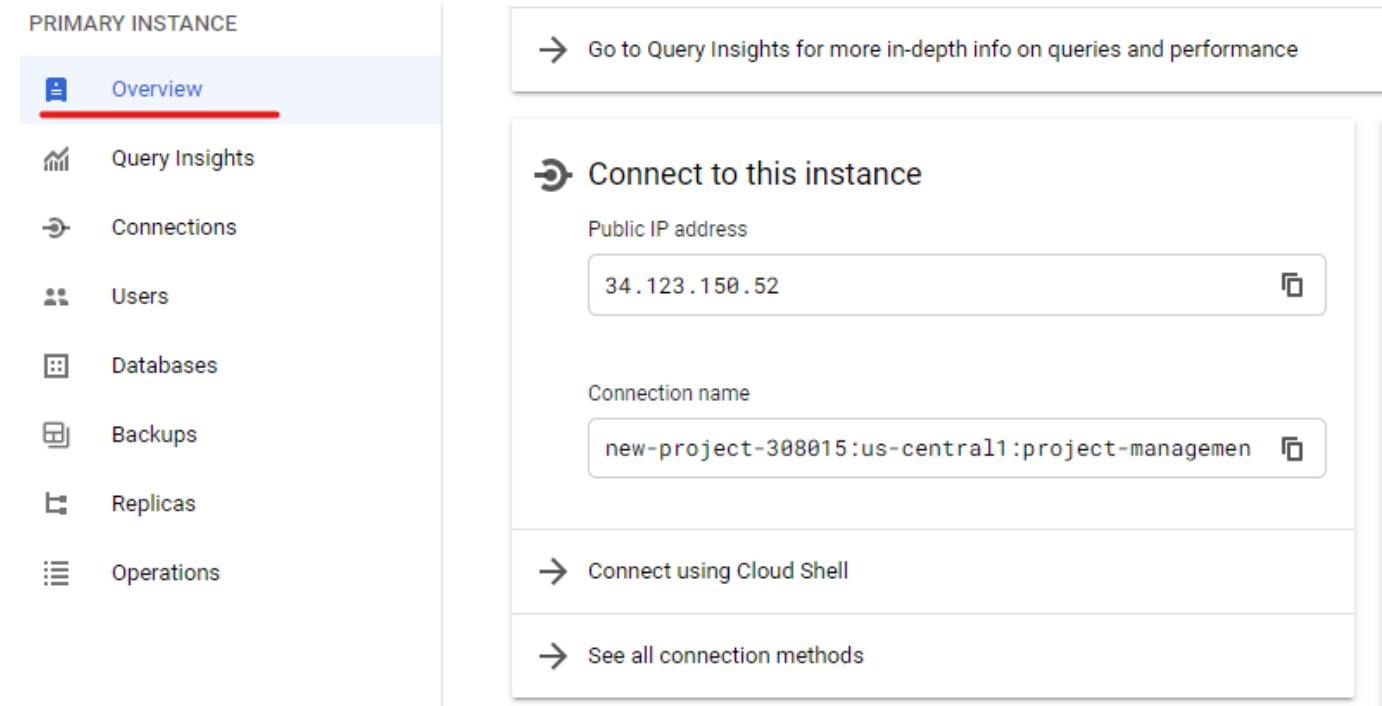
Crear una base de datos.



The screenshot shows the Google Cloud SQL console interface. On the left, a sidebar under the 'SQL' header lists navigation options: Overview, Query Insights, Connections, Users, **Databases** (highlighted with a red underline), Backups, Replicas, and Operations. The main panel is titled 'Databases' and shows the breadcrumb 'All instances > project-management'. Below this, a green checkmark icon precedes the instance name 'project-management' and the version 'PostgreSQL 12'. A blue button with a plus icon and the text 'CREATE DATABASE' is prominently displayed. Below the button is a table with the following data:

| Name ↑ | Collation | Character set | |
|----------|------------|---------------|---|
| postgres | en_US.UTF8 | UTF8 | ⋮ |

Con la base de datos creada, la podemos utilizar para almacenar los datos del proyecto. Para conectarse a la DB por medio de un cliente de DB como [DBeaver](#), o desde la aplicación de NodeJS, se puede consultar la información en la sección de "Overview". La IP pública es el host de la base de datos, el puerto es el default del gestor (MySQL: 3306, PostgreSQL: 5432), el usuario es el nombre de usuario por default del gestor (MySQL: *root*, PostgreSQL: *postgres*), la contraseña es la que se ingresó en la configuración básica de la instancia, y el nombre de la DB es el nombre que se seleccionó en el paso anterior.



Links de referencia

- [MySQL CloudSQL Quickstart](#)
- [What is CloudSQL?](#)

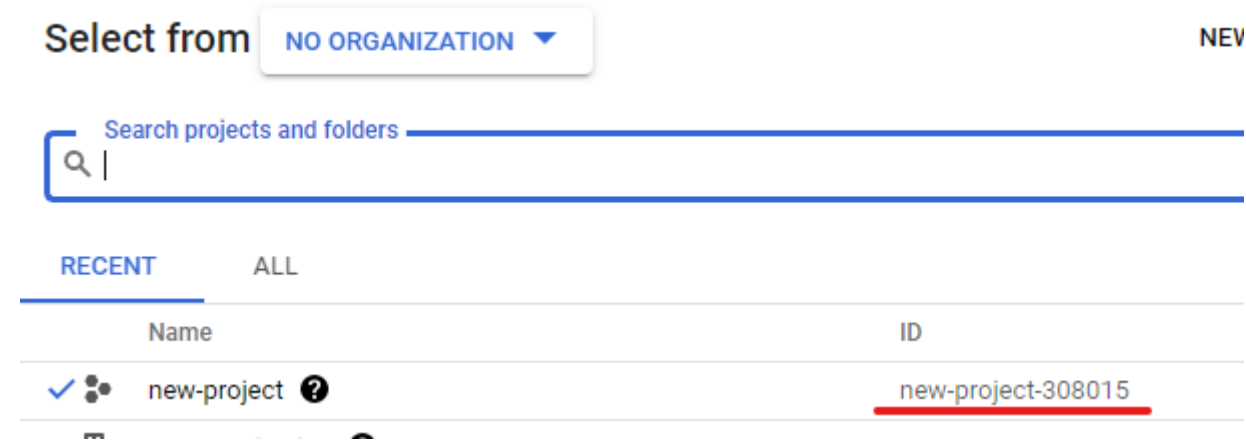
App Engine

App Engine es un ambiente *serverless* en google cloud, en el que con configuración mínima, puedes ejecutar aplicaciones web, cómo las de NodeJS. Algunas de las ventajas que brinda esto sobre utilizar un servidor virtual es que evita la configuración extensa de proxys, SSL, etc, y tiene capacidad de auto-escalamiento, replicando más instancias de la aplicación si llegara a ser necesario.

Para crear una aplicación en app engine, así cómo para desplegar nuevas versiones, se puede utilizar *gcloud cli*. Es una herramienta de terminal que permite administrar varios recursos de google cloud platform.

Instalar y configurar *gcloud cli* siguiendo la [guía oficial de instalación](#) y la [guía de configuración](#).

Obtén el ID del proyecto de google cloud al que quieres desplegar, desde la consola de GCP.



Desde una terminal, ejecutar el comando: `gcloud app create --project=[PROJECT_ID]`, y seleccionar la misma región que se utilizó para la instancia de CloudSQL. Tener los dos recursos en la misma región reduce la latencia en la comunicación entre la aplicación y la DB.

Para desplegar la aplicación, es necesario agregar un archivo [app.yaml](#).

Ejemplo de app.yaml para aplicación Backend de NodeJS:

```
runtime: nodejs12

automatic_scaling:
  min_idle_instances: automatic
  max_idle_instances: 1
  min_pending_latency: 3000ms
  max_pending_latency: automatic

handlers:
- url: /*
  secure: always
  redirect_http_response_code: 301
  script: auto
```

Este archivo debe de existir en la carpeta raíz del proyecto. Con el app.yaml creado, desde la terminal, navega a la carpeta raíz del proyecto y ejecuta el comando: `gcloud app deploy --project=[PROJECT_ID]`. Al terminar la ejecución, en la terminal se va a mostrar una URL donde se encuentra la aplicación.

Links de referencia

- [Hosting a static website on App Engine](#)
- [NodeJS App Engine Quickstart](#)
- [app.yaml Configuration File Reference](#)