

Workshop IZZI

Jueves a Martes de 9 a 11 am

ÍNDICE	

1. Prerrequisitos (Instalar)

- 1.1. Python 3.10
- 1.2. Git local
- 1.3. Docker
- 1.4. Visual Code
- 1.5. SDK Cloud
- 1.6. Manejador de base de datos DBeaver o Workbench.
- 1.7. Lista de permisos requeridos por usuario
 - Acceso a repositorios. Github
 - Cloud run admin
 - Invoker
 - Cloud build viewer
 - Compute engine admin.
 - ServiceAccountUser

2. Técnicas de aprendizaje (opcional)

- 2.1. Aprendizaje por frecuencia espaciada.
- 2.2. Aprendizaje por recompensa.
- 2.3. Aprendizaje por asociación.
- 2.4. Efecto Dunning Krager.

3. Práctica básica (Flask y Cloud run)

Cloud Run

Cloud Run es una plataforma de procesamiento administrada que le permite ejecutar contenedores sin estado que se pueden invocar a través de solicitudes HTTP. Cloud Run es una plataforma sin servidores que quita la complejidad de la administración de la infraestructura, para que pueda enfocarse en lo que más importa: compilar aplicaciones extraordinarias.

Cloud Run se basa en Knative, por lo que puede ejecutar sus contenedores completamente administrados con Cloud Run o en el clúster de Google Kubernetes Engine con Cloud Run en GKE.

El objetivo de esta práctica lab es que compile una imagen de aplicación sencilla, alojarla en contenedores y la implemente en Cloud Run.

- Implementa un contenedor de muestra que responda a solicitudes web entrantes con esta [Guía de inicio rápido](#).
- ¿Quieres desarrollar desde la fuente? Implementa una aplicación de muestra en Cloud Run desde la fuente con esta [guía](#).
- Ejecuta migraciones de bases de datos, informes nocturnos o transformaciones de datos por lotes con [los trabajos de Cloud Run](#).

Características

- Admite cualquier lenguaje, objeto binario o biblioteca

Usa el lenguaje de programación que deseas, las bibliotecas de cualquier lenguaje o sistema operativo, o incluso tus propios objetos binarios.

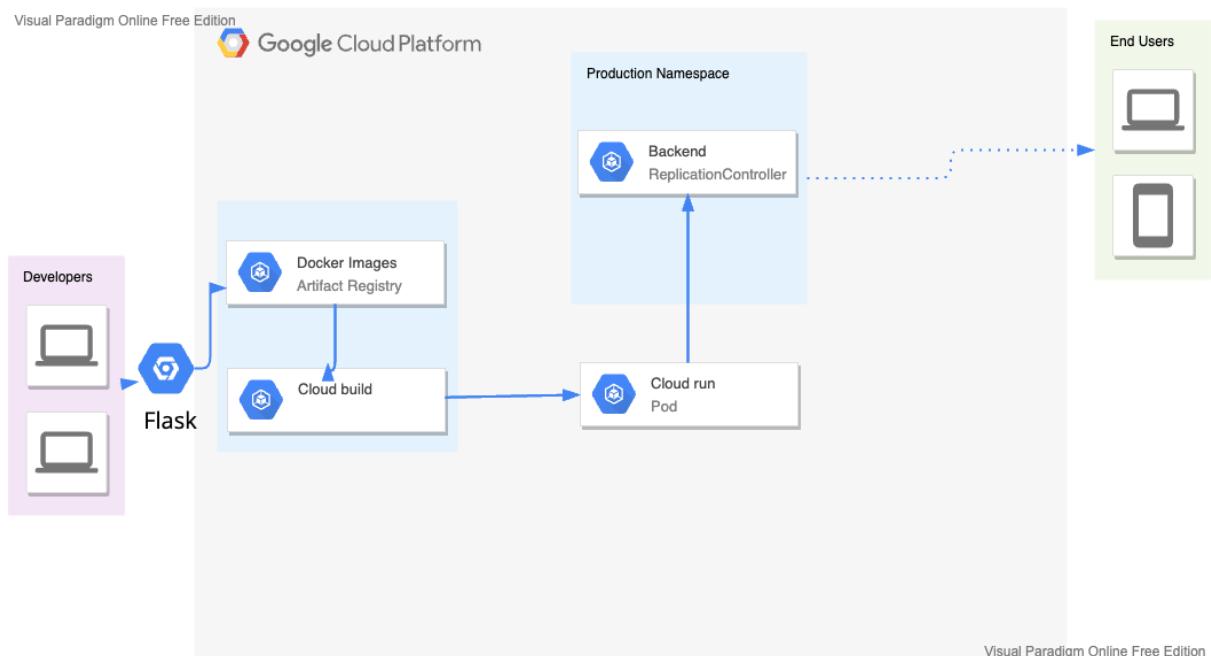
- Aprovecha los estándares y los flujos de trabajo de los contenedores

Los contenedores se convirtieron en un estándar para empaquetar e implementar el código y sus dependencias. Cloud Run funciona muy bien con el ecosistema de contenedores: [Cloud Build](#), [Cloud Code](#), [Artifact Registry](#) y [Docker](#).

- Paga por lo que usas

Paga solo por el tiempo de ejecución de tu código, con una facturación que se redondea a la centena de milisegundos más cercana.

3.1. Imagen de arquitectura



3.2. Pre requisitos de configuración inicial

- Iniciar terminal en la máquina personal o iniciar en la consola de GCP.
- Verificar la cuenta en la consola
 - `gcloud auth list`
- Autenticar con la cuenta a utilizar
 - `gcloud auth login`
- Configurar región en la que vamos a trabajar
 - `gcloud config set compute/region us-central1`

- Si se desea activar la api cloud run desde consola, ejecutar siguiente comando

gcloud services enable run.googleapis.com

- Setear una variable de entorno para fijar la región y proyecto en la que vamos a trabajar más adelante

LOCATION="us-central1" ó export LOCATION="us-central1"

GOOGLE_CLOUD_PROJECT="nombre-proyecto"

- Fijar proyecto en el que vamos a trabajar

gcloud config set project \${GOOGLE_CLOUD_PROJECT}

- Listamos los proyectos para tomar el número de proyecto.

\$ gcloud projects list

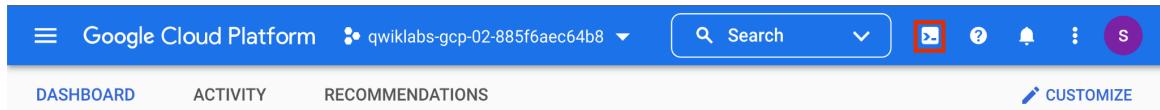
- Configuramos la variable del número de proyecto (PROJECT_NUMBER)

GOOGLE_CLOUD_PROJECT_NUMBER="project-example-33"

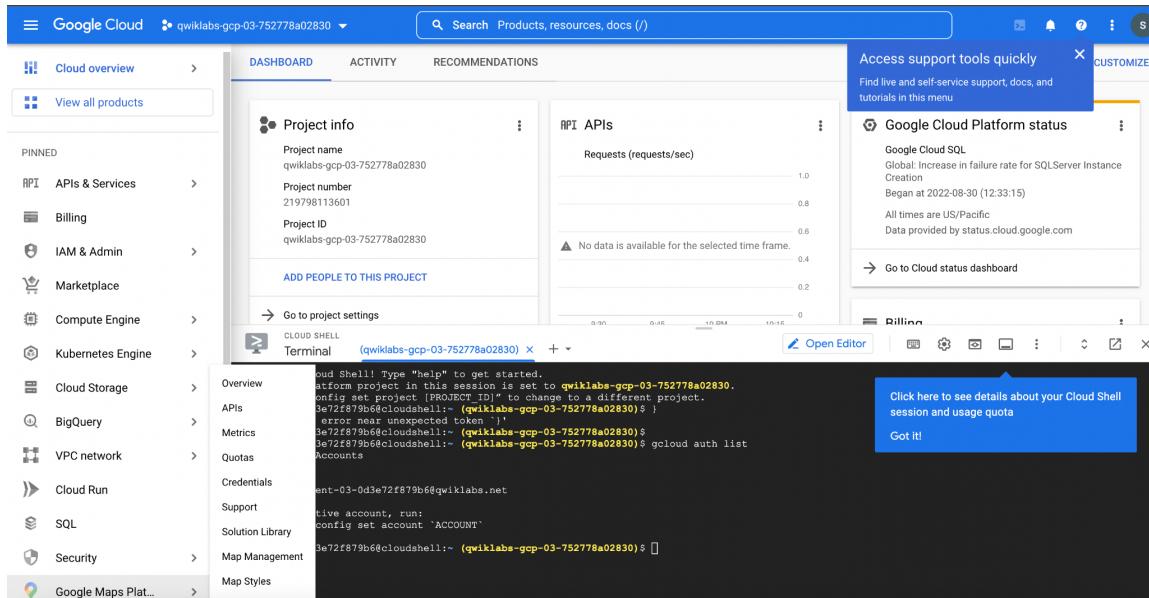
- Confirmar la configuración

gcloud config list

Botón para abrir consola desde la consola



Luego de abrir la consola, se verá de esta forma



3.3. Correr aplicacion en docker local para probar funcionalidad

- docker build -t helloworld .
- docker run -e PORT=8080 -p 8080:80 helloworld

3.4. Desplegar una aplicación python desde código fuente.

- Abrir código python xxx
- Ejecutar comando
\$ gcloud run deploy
\$ (Si deseas apuntar la carpeta actual) gcloud run deploy --source .
\$ (Especificar el nombre de la imagen y de la app) gcloud run deploy
hello-nombre --image
us-central1-docker.pkg.dev/\${GOOGLE_CLOUD_PROJECT}/cloud-run-source-deploy/hello-img
 - Preguntará por el nombre de la app, elegir un nombre.
 - Solicitará activar apis necesarias, responder Si a todo
 - run.googleapis.com
 - artifactregistry.googleapis.com
 - Preguntará en qué región se va desplegar, escribir [26] us-central1
 - Al final solicitará confirmar el despliegue si todos los pasos fueron exitosos. Responder sí para continuar.
 - Preguntará si desea activar el despliegue sin autenticación del servicio. Si

Debería finalizar con una respuesta de la siguiente manera

```
Operation "operations/acat.pz-284/69178757-495f8237-cfe0-4fa8-954c-f352365cc76a" finished successfully.
Deploying from source requires an Artifact Registry Docker repository to store built containers. A repository named
[cloud-run-source-deploy] in region [us-central1] will be created.

Do you want to continue (Y/n)? y
This command is equivalent to running `gcloud builds submit --tag [IMAGE] /Users/jhoannatera/Desktop/PROYECTOS/GCP/work_shop_izzi/practic_01/helloworld_from_code` and
`gcloud run deploy helloworldfromcode --image [IMAGE]`.

Allow unauthenticated invocations to [helloworldfromcode] (y/N)? y
Building using Dockerfile and deploying container to Cloud Run service [helloworldfromcode] in project [innate-bonfire-361004] region [us-central1]
OK Building and deploying new service... Done.
OK Creating Container Repository...
OK Uploading sources...
OK Building Container... Logs are available at [https://console.cloud.google.com/cloud-build/builds/1c396db7-0931-4a86-b125-d229c
3dfb2e97?project=284769178757].
OK Creating Revision...
OK Routing traffic...
OK Setting IAM Policy...
Done.
Service [helloworldfromcode] revision [helloworldfromcode-00001-zaw] has been deployed and is serving 100 percent of traffic.
Service URL: https://helloworldfromcode-te7xy2moeq-uc.a.run.app
```

- Al final de la respuesta se encontrará la url del servicio desplegado, a continuación abrirlo en el navegador.

Cloud Build

Compila, prueba e implementa en nuestra plataforma de CI/CD sin servidores.

- Compila software rápidamente en todos los lenguajes de programación, incluidos Java, Go, Node.js y muchos más.
- Elige entre [15 tipos de máquinas](#) y ejecuta cientos de compilaciones simultáneas por grupo.
- Se implementa en varios entornos, como en las VM, Kubernetes, Firebase o sin servidores.
- Accede a flujos de trabajo de CI/CD alojados en la nube y completamente administrados dentro de [tu red privada](#).
- Mantén tus datos en reposo dentro de una región geográfica o una ubicación específica con residencia de datos

Compilaciones extremadamente rápidas

Accede a las máquinas conectadas a través de la red global de Google con el fin de reducir de forma significativa el tiempo de compilación. Ejecuta compilaciones en VMs con alta capacidad de CPU o almacena en caché el código fuente, las imágenes y otras dependencias para incrementar aún más la velocidad de compilación.

- Automatiza tus implementaciones

Crea canalizaciones como parte de los pasos de compilación a fin de automatizar las implementaciones. Implementa con integraciones incorporadas de [Google Kubernetes Engine](#), [App Engine](#), [Cloud Functions](#) y [Firebase](#). Usa Spinnaker con Cloud Build para crear y ejecutar canalizaciones complejas.

- Compatibilidad con múltiples nubes

Implementa en múltiples nubes como parte de tu canalización de CI/CD. Cloud Build incluye imágenes de compilador que cuentan con lenguajes y herramientas instalados. Asimismo, las tareas alojadas en contenedores de Cloud Build son completamente portátiles en diferentes nubes.

- Confirma la implementación en minutos

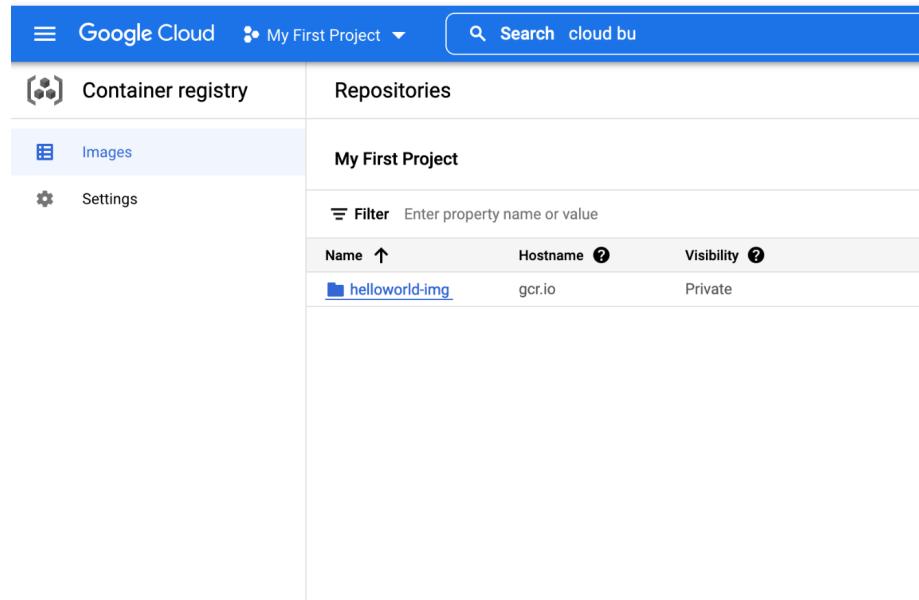
Pasar de la solicitud de extracción a la compilación, la prueba y la implementación es muy sencillo. Configura activadores para compilar, probar o implementar automáticamente el código fuente cuando envías cambios a GitHub, [Cloud Source Repositories](#) o un repositorio de Bitbucket.

- Privacidad inigualable

Ejecuta compilaciones en una infraestructura protegida con la seguridad de Google Cloud. Activa flujos de trabajo de CI/CD completamente administrados desde repositorios privados de código fuente alojados en redes privadas, incluido GitHub Enterprise.

3.5. Aloje su aplicación en contenedores y súbelo a Artifact Registry

- Compilar la imagen en gcp en la ruta central
`gcloud builds submit --tag gcr.io/${GOOGLE_CLOUD_PROJECT}/helloworld-img`
- Listar las imágenes en container registry desde consola
`$ gcloud container images list`
- Configurar docker para correr imágenes que se encuentran en container registry.
`$ gcloud auth configure-docker`
- Para ejecutar y probar la aplicación de manera local desde Cloud Shell, inicie con el siguiente comando estándar de docker
`$ docker run -d -p 8080:8080 gcr.io/${GOOGLE_CLOUD_PROJECT}/helloworld-img`



- Desplegar desde la imagen compilada anteriormente
`$ gcloud run deploy --image
gcr.io/$GOOGLE_CLOUD_PROJECT/helloworld-img --allow-unauthenticated
--region=$LOCATION`

3.6. Desplegar servicio desde la interfaz cloud run gcp de una imagen existente..

- Seleccionar la imagen
- Mantener parámetros por defecto y continuar.

Seleccione la imagen, coloque el nombre del servicio y al final marcar no autenticado.

Cloud Run

Create service

A service exposes a unique endpoint and automatically scales the underlying infrastructure to handle incoming requests. Service name and region cannot be changed later.

- Deploy one revision from an existing container image

Container image URL —

gcr.io/innate-bonfire-361004/helloworld-img@sha256:d2032166ff55 · SELECT

TEST WITH A SAMPLE CONTAINER

Should listen for HTTP requests on \$PORT and not rely on local state.[How to build a container](#)

- Continuously deploy new revisions from a source repository

Service name * —

helloworld-img

Region * —

us-central1 (Iowa)

[How to pick a region?](#)

CPU allocation and pricing

- CPU is only allocated during request processing

You are charged per request and only when the container instance processes a request.

- CPU is always allocated

You are charged for the entire lifecycle of the container instance.

Now view

- CPU is always allocated
You are charged for the entire lifecycle of the container instance.

Auto-scaling ?

Minimum number of instances * <input type="text" value="0"/>	Maximum number of instances <input type="text" value="100"/>
--	--

Set to one to reduce cold starts. [Learn more](#)

Ingress ?

- Allow all traffic
 Allow internal traffic and traffic from Cloud Load Balancing
 Allow internal traffic only

Authentication * ?

- Allow unauthenticated invocations
Tick this if you are creating a public API or website.
 Require authentication
Manage authorised users with Cloud IAM.

Container, connections, security

CREATE

CANCEL

3.7. Compilar y enviar una imagen mediante Docker a GCP.

- Compilamos la imagen con docker.
- Vamos a la carpeta donde se encuentra el código python y ejecutamos el siguiente comando.

```
$ gcloud builds submit --tag  
us-central1-docker.pkg.dev/${GOOGLE_CLOUD_PROJECT}/cloud-run-sourc  
e-deploy/hello-docker
```

- Si aún no configuras Docker para Google Cloud CLI para autenticar solicitudes Container Registry, hazlo con el siguiente comando.
- \$ gcloud auth configure-docker
- A continuación enviamos la imagen del contenedor a Container Registry. Para ello ejecutamos el siguiente comando.

- gcloud builds submit --tag us-central1-docker.pkg.dev/\${GOOGLE_CLOUD_PROJECT}/cloud-run-source-deploy/hello-docker*
- Ir al servicio Artifact Registry y entrar a cloud-run-source-deploy/ y confirmar que se encuentre la imagen.

The screenshot shows the Google Cloud Artifact Registry interface. The left sidebar has 'Artifact Registry' selected. The main area shows a breadcrumb path: 'us-central1-docker.pkg.dev > innate-bonfire-361004 > cloud-run-source-deploy'. Below the path is a search bar and a 'DELETE' button. A 'SETUP INSTRUCTIONS' link is also present. A 'Filter' input field is available. A table lists the image 'hello-docker' with columns 'Name', 'Created', and 'Updated'. The 'Name' column shows a small Docker icon followed by 'hello-docker'. The 'Created' and 'Updated' columns both show '5 minutes ago'.

- Ir a cloud run y crear un servicio por medio de la interfaz de cloud run (click a Create service).
- Seleccione la imagen creada por docker y enviada a artifact registry y seleccione hello-docker version latest.

The screenshot shows the 'Create service' dialog in Google Cloud Cloud Run. It includes fields for 'Container image URL' and 'Service name'. Below these are sections for 'TEST WITH A SAMPLE CONTAINER' and 'Continuously deploy new revisions from a source repository'. On the right, there's a 'Cloud Run pricing' section with 'Free tier' options. At the bottom right are 'SELECT' and 'CANCEL' buttons. A separate window titled 'ARTIFACT REGISTRY' is open, showing the 'innate-bonfire-361004' project with a list of images under 'hello-docker'. One image, '8bad1cbed9', is highlighted as 'latest' and was updated '6 minutes ago'. There are also other entries like 'helloworldfromcode' and 'nuevo2'.

3.8. Desplegar en cloud run por trabajo (jobs)

A diferencia de un servicio de Cloud Run, que escucha y entrega solicitudes, un trabajo de Cloud Run solo ejecuta sus tareas y se cierra cuando finaliza. Un trabajo no escucha ni entrega solicitudes, y no puede aceptar parámetros arbitrarios en la ejecución.

Puedes estructurar un trabajo como una sola tarea o como varias tareas independientes (hasta 10,000 tareas) que se pueden ejecutar en paralelo. Cada tarea ejecuta una instancia de contenedor y se puede configurar para que se reintente en caso de falla. Cada tarea conoce su índice, que se almacena en la variable de entorno CLOUD_RUN_TASK_INDEX. Si procesas datos en paralelo, el

código es responsable de determinar qué tarea controla qué subconjunto de los datos.

Puedes establecer tiempos de espera para las tareas y especificar la cantidad de reintentos en caso de falla de la tarea. Si alguna tarea excede la cantidad máxima de reintentos, esa tarea se marca como *con errores* y el trabajo se marca como *con errores* una vez que se ejecuten todas las tareas.

De forma predeterminada, cada tarea se ejecuta durante 10 minutos como máximo: puedes cambiar esto a un tiempo más corto o a un máximo de hasta 1 hora, si [cambias la configuración del tiempo de espera de la tarea](#). No hay un tiempo de espera explícito para la ejecución de un trabajo: después de que se completaron todas las tareas, la ejecución del trabajo está completa.

Los trabajos usan el [entorno de ejecución de segunda generación](#).

Aquí aprenderemos a crear un trabajo de Cloud Run simple, empaquetarlo en una imagen de contenedor, subir esta imagen de contenedor a Container Registry y luego implementar Cloud Run.

- Compilamos el código en una imagen y enviamos al registro.
- Usamos BuildPack para compilar el contenedor (Buildpack es un compilador de código fuente)

```
$ gcloud builds submit --pack image=gcr.io/${PROJECT_ID}/logger-job
```

- A continuación creamos el trabajo con el contenedor que acabas de compilar, con el siguiente comando.

```
$ gcloud beta run jobs create job-quickstart \
  --image gcr.io/${PROJECT_ID}/logger-job \
  --tasks 50 \
  --set-env-vars SLEEP_MS=10000 \
  --set-env-vars FAIL_RATE=0.5 \
  --max-retries 5 \
  --region ${LOCATION}
```

- Ahora ejecutamos el trabajo en Cloud Run.
- Para ejecutar el trabajo que acabas de crear, ejecuta el siguiente paso y al solicitar la región selecciona us-central1(26) (puedes elegir otra si gustas)

```
$ gcloud beta run jobs execute job-quickstart
```

- Para ver detalles de la ejecución puedes correr el siguiente comando.

```
$ gcloud beta run jobs executions describe job-quickstart-66pm5
```

- Puedes ejecutar el trabajo de forma local para comprobar funcionamiento.

```
docker run --rm -e FAIL_RATE=0.9 -e SLEEP_MS=1000  
gcr.io/PROJECT_ID/logger-job
```

3.9. Compilación y desplegar con Cloud Build desde una configuración yaml.

En esta sección, se explica un ejemplo de archivo de configuración de compilación para una app de Python. Tiene pasos de compilación para instalar requisitos, agregar pruebas de unidades y, luego de pasar las pruebas, compilar e implementar la app.

- Vamos a trabajar con la carpeta /practic_01/python-example-flask/
- De ser necesario aplicamos permisos necesarios a nuestra cuenta
 - # Grant the Cloud Run Admin role to the Cloud Build service account
 - gcloud projects add-iam-policy-binding
 - \$GOOGLE_CLOUD_PROJECT \
--member
 - "serviceAccount:\$GOOGLE_CLOUD_PROJECT_NUMBER@cloudbui
ld.gserviceaccount.com" \
--role roles/run.admin
- # Grant the IAM Service Account User role to the Cloud Build service account on the Cloud Run runtime service account
 - gcloud iam service-accounts add-iam-policy-binding \
\$GOOGLE_CLOUD_PROJECT_NUMBER-compute@developer.gser
viceaccount.com \
--member="serviceAccount:\$GOOGLE_CLOUD_PROJECT_NUMBE
R@cloudbuild.gserviceaccount.com" \
--role="roles/iam.serviceAccountUser"
- Para realizar todas las tareas configuradas en el archivo de cloudbuild es necesario correr el siguiente comando.
 - \$ gcloud builds submit --region=us-central1 --config cloudbuild.yaml
 - substitutions=_ARTIFACT_REGISTRY_REPO="cloud-run-source-de
ploy",_BUCKET_NAME="\${GOOGLE_CLOUD_PROJECT}_cloudbuil
d",_PROJECT_ID="\${GOOGLE_CLOUD_PROJECT}",_SHORT_SHA
="mi-nombre" .

- A continuación se podrá ver el servicio desplegado en cloud run y ver el log de la ejecución en cloud build.

The screenshot shows the Google Cloud Build interface. On the left, there's a sidebar with 'Cloud Build' selected, showing 'Dashboard' and 'History'. The main area displays a successful build with ID 86774d39, started on 1 Sept 2022, 01:55:34. It lists 5 steps: 0: python pip install -r requirements.txt --user, 1: python python -m pytest --junitxml=jhoan_test..., 2: gcr.io/cloud-builders/docker build -t us-central1-docker.pkg.dev/innate... 3: gcr.io/cloud-builders/docker push us-central1-docker.pkg.dev/innate... 4: google/cloud-sdk gcloud run deploy helloworld-jhoan --image. The duration of the build was 00:02:15. Below the steps is a detailed 'BUILD LOG' tab showing the command-line output of the build process, including fetching storage objects, copying files, and building Docker images.

The screenshot shows the Google Cloud Run interface. At the top, it says 'Cloud Run' and 'Services'. Below that, there are tabs for 'SERVICES', 'JOBS', and 'PREVIEW', with 'SERVICES' currently selected. There is a 'Filter' button and a 'Filter services' input field. The table below lists two services: 'helloworld-jhoan' and 'helloworld-latest'. Both services have 0 req/sec, are in the us-central1 region, use unauthenticated authentication, and were last deployed 4 and 15 minutes ago respectively, both by Cloud Build.

3.10. Trabajo de limpieza

- Eliminar imagen compilada


```
$ gcloud container images delete
gcr.io/$GOOGLE_CLOUD_PROJECT/helloworld-img
```
- Para borrar el servicio de Cloud Run, use el siguiente comando


```
$ gcloud beta run services delete helloworld-img-ser
```

3.11. Tarea individual.

- Realizar cada tarea arriba mencionada

3.12. Tarea para la casa.

- Desplegar un servicio desde yaml con una aplicación diferente.
Ejemplo en nodejs, python, php, front o back.

4. Practica (Django, Sql cloud, Cloud run)

4.1. Imagen de arquitectura

4.2. SQL Cloud (Mysql)

- Crear instancia sql cloud.
- Conectarse por proxy a la instancia de la base de datos.
- Conectarse por ip pública desde el manejador.

4.3. Conectar aplicación python a mysql

4.4. Cloud run trigger

4.5. Desplegar aplicación en cloud run

4.6. Tarea individual.

4.7. Tarea para la casa.

5. Práctica (Django, Sql cloud, Cloud Run, jenkins)

5.1. Imagen de arquitectura

5.2. Instalar y configurar jenkins desde marketplace.

- Permisos
- Instalar y activar plugin gcloud sdk.
- Crear archivo jenkins en la aplicación.
- Creamos pipeline en jenkins

5.3. Desplegar a Jenkins desde el repositorio.

5.4. Manejo de secrets y variables de entorno

5.5. Tarea individual.

5.6. Tarea para la casa.

6. Practica (Django, Sql cloud, Cloud Run, jenkins, api gateway)

6.1. Imagen de arquitectura

6.2. Crear envoltura con api gateway.

6.3. Activar api gateway

6.4. Generar el archivo openapi desde django y adaptarlo para api gateway

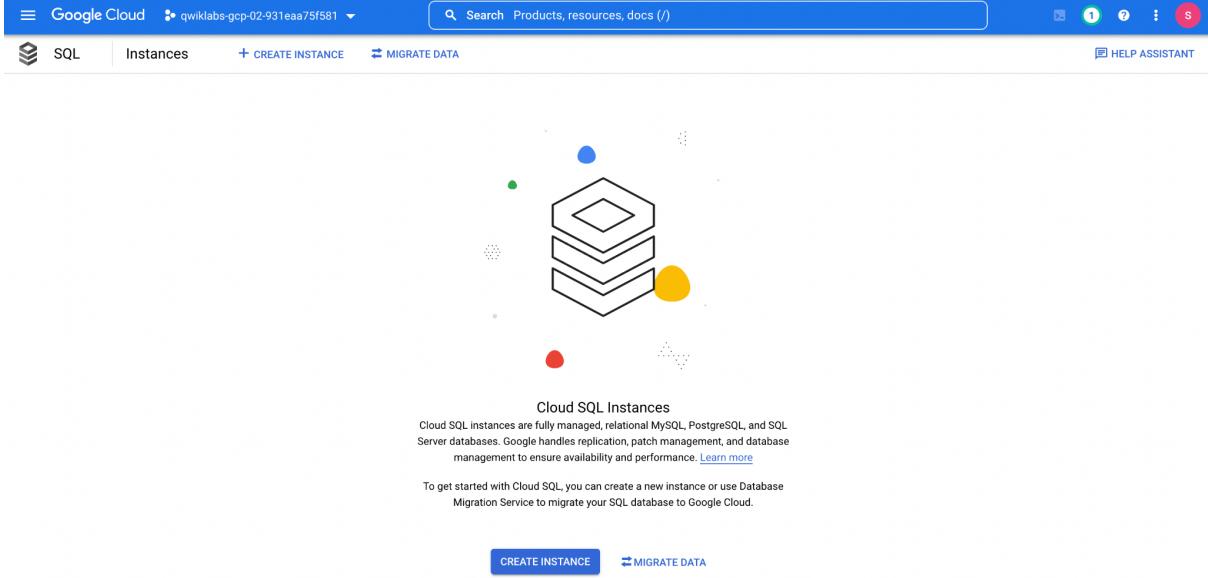
6.5. Configurar puerta de enlace.

6.6. Desplegamos y probamos

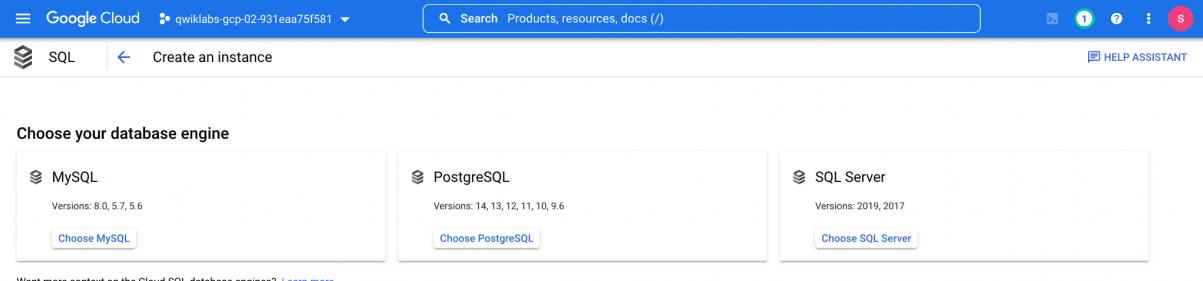
6.7. Tarea individual.

6.8. Tarea para la casa.

sql



The screenshot shows the Google Cloud SQL Instances page. At the top, there's a navigation bar with the Google Cloud logo, a search bar, and various icons. Below the navigation is a header with 'SQL' and 'Instances' buttons, a '+ CREATE INSTANCE' button, and a 'MIGRATE DATA' button. The main area features a large graphic of three stacked hexagons representing a database. Below the graphic, the heading 'Cloud SQL Instances' is displayed, followed by a brief description: 'Cloud SQL instances are fully managed, relational MySQL, PostgreSQL, and SQL Server databases. Google handles replication, patch management, and database management to ensure availability and performance.' A 'Learn more' link is provided. Further down, instructions for getting started are given: 'To get started with Cloud SQL, you can create a new instance or use Database Migration Service to migrate your SQL database to Google Cloud.' At the bottom of the page are two buttons: 'CREATE INSTANCE' and 'MIGRATE DATA'.



The screenshot shows the 'Create an instance' page for Cloud SQL. The top navigation bar and header are identical to the previous page. The main content is titled 'Choose your database engine'. It lists three options: 'MySQL', 'PostgreSQL', and 'SQL Server'. Each option has a small icon, a list of supported versions, and a 'Choose [engine]' button. Below the engine choices, a note reads: 'Want more context on the Cloud SQL database engines? [Learn more](#)'.

≡ Google Cloud • qwiklabs-gcp-02-931eaa75f581 ▾

Search

← Create a MySQL instance

Instance info

Instance ID *

blog-db

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

Password *

 GENERATE

Set a password for the root user. [Learn more](#)

No password

▼ PASSWORD POLICY

Database version *

MySQL 8.0

▼

Choose a configuration to start with

These suggested configurations will pre-fill this form as a starting point for creating an instance. You can customize as needed later.



You've made changes that override the selected configuration. [RESET](#)

Production

Optimized for the most critical workloads. Highly available, performant, and durable.

Development

Performant but not highly available, while reducing cost by provisioning less compute and storage.

Google Cloud • qwiklabs-gcp-02-931eaa75f581 ▾

Search

← Create a MySQL instance

▼ CONFIGURATION DETAILS

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.

You have the location org policy in effect that is restricting your regions.

Region

us-central1 (Iowa)

Zonal availability

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.

▼ SPECIFY ZONES

Customize your instance

You can also customize instance configurations later

▼ SHOW CONFIGURATION OPTIONS

CREATE INSTANCE CANCEL

Google Cloud • qwiklabs-gcp-02-931eaa75f581 ▾

Search Products, resources, docs (/)

SQL Overview EDIT IMPORT EXPORT RESTART STOP DELETE CLONE

PRIMARY INSTANCE

All instances > blog-db

blog-db

MySQL 8.0

Instance is being created. This may take a few minutes. While this operation is running, you may continue to view information about the instance.

1 hour 6 hours ✓ 1 day 7

Chart CPU utilization

No data is available for the selected time frame.

UTC-5 6:00 PM 8:00 PM 10:00 PM Aug 30 2:00 AM 4:00 AM 6:00 AM 8:00 AM 10:00 AM 12:00 PM

Connect to this instance

Connection name

qwiklabs-gcp-02-931eaa75f581:us-central1:blog-db

Need help connecting?

Configuration

vCPUs 4 Memory 26 GB

Click here to see details Creating blog-db

Release Notes

CLOUD SHELL Terminal (qwiklabs-gcp-02-931eaa75f581) + -

☰ Google Cloud • qwiklabs-gcp-02-931eaa75f581 ▾

Search Products, resources, docs (/)

 SQL	Users
PRIMARY INSTANCE	
Overview	All instances > blog-db
Connections	 blog-db
Users	MySQL 8.0
Databases	User accounts enable users and applications to connect to your instance. Learn more
Backups	+ ADD USER ACCOUNT
Replicas	
Operations	

Search Products, resources, docs (/)

Add a user account to instance blog-db

Choose how to authenticate

You can manage access to this instance using Cloud IAM or MySQL built-in authentication. [Learn more](#)

Built-in authentication
Creates a new username and password specific to this instance. User account will have cloudsqlsuperuser root access, but you can customize that later as needed. [Learn more](#)

User name *

Password (Optional) [GENERATE](#)

[▼ PASSWORD POLICY](#)

Host name ?
 Allow any host (%)
 Restrict host by IP address or address range

Users created with built-in authentication have the same privileges as the root user. [Learn more](#)

Cloud IAM
Associates an existing IAM principal with this user account. Must have a role providing instance-level access assigned to connect.

[ADD](#) [CANCEL](#)

 SQL

PRIMARY INSTANCE

- [Overview](#)
- [Connections](#)
- [Users](#)
- [Databases](#)
- [Backups](#)
- [Replicas](#)
- [Operations](#)

Users

All instances > blog-db

 **blog-db**

MySQL 8.0

User accounts enable users and applications to connect to your instance. [Learn more](#)

[+ ADD USER ACCOUNT](#)

	User name	Host name	Authentication	Password status	⋮
	blogdbuser	% (any host)	Built-in	N/A	⋮
	root	% (any host)	Built-in	N/A	⋮

≡ Google Cloud ⚙ qwiklabs-gcp-02-931eaa75f581 ▾

 SQL

PRIMARY INSTANCE

- [Overview](#)
- [Connections](#)
- [Users](#)
- [Databases](#)
- [Backups](#)
- [Replicas](#)
- [Operations](#)

Connections

All instances > blog-db

 **blog-db**

MySQL 8.0

[NETWORKING](#) [SECURITY](#) [CONNECTIVITY TESTS](#)

Choose how you want your source to connect to this instance, then define which networks are authorized to connect. [Learn more](#)

You can use the Cloud SQL Proxy for extra security with either option. [Learn more](#)

Instance IP assignment

Private IP
Assigns an internal, Google-hosted VPC IP address. Requires additional APIs and permissions. Can't be disabled once enabled. [Learn more](#)

Public IP
Assigns an external, internet-accessible IP address. Requires using an authorized network or the Cloud SQL Proxy to connect to this instance. [Learn more](#)

Authorized networks

You can specify CIDR ranges to allow IP addresses in those ranges to access your instance. [Learn more](#)

 You have not authorized any external networks to connect to your Cloud SQL instance. External applications can still connect to the instance through the Cloud SQL Proxy. [Learn more](#)

Connections

Choose how you want your source to connect to this instance, then define which networks are authorized to connect. [Learn more](#)

You can use the Cloud SQL Proxy for extra security with either option. [Learn more](#)

Instance IP assignment

Private IP

Assigns an internal, Google-hosted VPC IP address. Requires additional APIs and permissions. Can't be disabled once enabled. [Learn more](#)

Public IP

Assigns an external, internet-accessible IP address. Requires using an authorized network or the Cloud SQL Proxy to connect to this instance. [Learn more](#)

Authorized networks

You can specify CIDR ranges to allow IP addresses in those ranges to access your instance. [Learn more](#)



You have not authorized any external networks to connect to your Cloud SQL instance. External applications can still connect to the instance through the Cloud SQL Proxy. [Learn more](#)

[ADD NETWORK](#)

App Engine authorization

All apps in this project are authorized by default. You can use [Cloud IAM](#) to authorize apps in other projects. [Learn more](#)

[SAVE](#)

[DISCARD CHANGES](#)



SQL

PRIMARY INSTANCE

[Overview](#)[Connections](#)[Users](#)[Databases](#)[Backups](#)[Replicas](#)[Operations](#)

Connections

Assigns an external, internet-accessible IP address. Requires using an authorized network or the Cloud SQL Proxy to connect to this instance. [Learn more](#)

Authorized networks

You can specify CIDR ranges to allow IP addresses in those ranges to access your instance. [Learn more](#)

New network

Name

Use [CIDR notation](#)

Network *

Example: 199.27.25.0/24

[CANCEL](#) [DONE](#)

[ADD NETWORK](#)

App Engine authorization

All apps in this project are authorized by default. You can use [Cloud IAM](#) to authorize apps in other projects. [Learn more](#)

[Release Notes](#)[SAVE](#)[DISCARD CHANGES](#)

