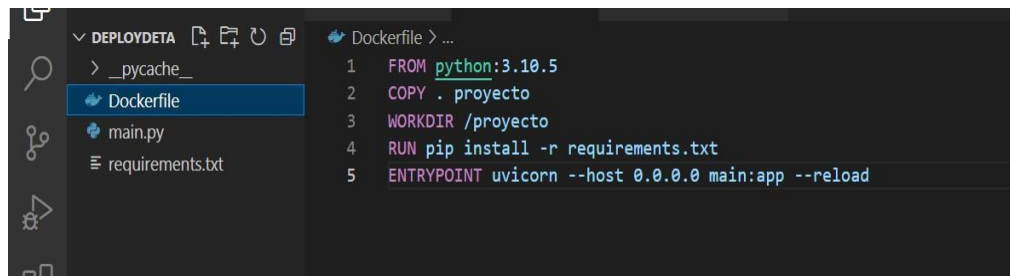


Pasos para crear y desplegar un contenedor de Docker con FastAPI

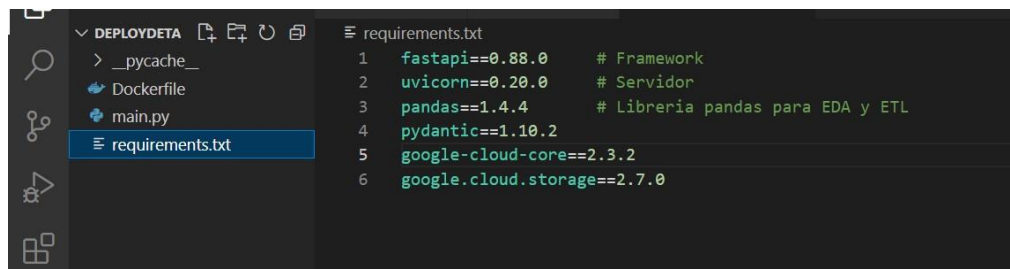
Paso 1:

- **Crear api en un entorno local con Python y FastAPI**
- **Crear archivo Dockerfile y requirements.txt con las siguientes características:**



A screenshot of a code editor showing the content of a Dockerfile. The file explorer on the left shows a project structure with files: __pycache__, Dockerfile, main.py, and requirements.txt. The Dockerfile content is as follows:

```
1 FROM python:3.10.5
2 COPY . proyecto
3 WORKDIR /proyecto
4 RUN pip install -r requirements.txt
5 ENTRYPOINT uvicorn --host 0.0.0.0 main:app --reload
```



A screenshot of a code editor showing the content of a requirements.txt file. The file explorer on the left shows a project structure with files: __pycache__, Dockerfile, main.py, and requirements.txt. The requirements.txt content is as follows:

```
1 fastapi==0.88.0 # Framework
2 uvicorn==0.20.0 # Servidor
3 pandas==1.4.4 # Libreria pandas para EDA y ETL
4 pydantic==1.10.2
5 google-cloud-core==2.3.2
6 google.cloud.storage==2.7.0
```

Paso 2:

- **Con Docker Desktop prendido y conectado a nuestra cuenta correr los siguientes comandos por terminal:**
- `docker build -t "nombre de la imagen del contenedor"`
- `docker run -it -p 8000:8000 -v cd/:"nombre de la carpeta donde se encuentra la imagen (proyecto)" "nombre de la imagen"`
- **De esta manera se inicializa el contenedor y se prueba la api, CTRL+C para detener**
- **Renombramos los datos de la imagen:** `docker tag "nombre de la imagen" "nombre de usuario Docker"/"nombre de la imagen"`
- **Enviamos la imagen a Docker Hub:** `docker push "nombre de usuario Docker"/"nombre de la imagen"`

Paso 3:

- **Desde la terminal de Google Cloud hacemos lo siguiente:**
- **Nos logueamos en Docker:** `docker login`
- **Ponemos nuestro nombre de usuario y contraseña de Docker**
- `docker pull "nombre de usuario Docker"/"nombre de la imagen"`
- **Renombramos la imagen para poder traerla y depositarla en GCP:**
- `docker tag "nombre de usuario Docker"/"nombre de la imagen" gcr.io/"id del proyecto"/"nombre nuevo de la imagen: versión"`
- `Docker push gcr.io/"id del proyecto"/"nombre nuevo de la imagen: versión"`
- **Tener habilitado Google Container Registry API**
- **Puede pedirnos autorizar a enviar en un cartel**

Paso 4:

- Con la nueva imagen ya en GCP, desde Cloud Run vamos a “Crear Servicio” y completamos con la siguiente configuración:



Cloud Run



Crear servicio

Cada servicio expone un extremo único y ajusta de forma automática la escala de la infraestructura subyacente para controlar las solicitudes entrantes. No se puede cambiar el nombre del servicio ni la región más adelante.

- ☒ Implementar una revisión desde una imagen de contenedor

URL de la imagen del contenedor

gcr.io/proyecto-olist/datalake@sha256:d585681c1b1be0a916 [SELECCIONAR](#)

REALIZAR PRUEBAS CON UN CONTENEDOR DE MUESTRA

Debe detectar las solicitudes HTTP en \$PORT y no depender del estado local. [¿Cómo se compila un contenedor?](#)

- ☐ Implementar de forma continua revisiones nuevas desde un repositorio de código fuente

Nombre del servicio *

datalake

Región *

us-central1 (Iowa)

[¿Cómo se selecciona la región?](#)

Precios y asignación de CPU ?

- ☒ La CPU solo se asigna durante el procesamiento de la solicitud
Se te cobra por solicitud y solo cuando la instancia de contenedor procesa una solicitud.
- ☐ La CPU siempre está asignada
Se te cobra por todo el ciclo de vida de la instancia de contenedor.

Ajuste de escala automático ?

Número mínimo de instancias *

0

Número máximo de instancias

10



Cloud Run



Crear servicio

Precios y asignación de CPU ?

- ☒ La CPU solo se asigna durante el procesamiento de la solicitud
Se te cobra por solicitud y solo cuando la instancia de contenedor procesa una solicitud.
- ☐ La CPU siempre está asignada
Se te cobra por todo el ciclo de vida de la instancia de contenedor.

Ajuste de escala automático ?

Número mínimo de instancias *

0

Número máximo de instancias

10

Configúralo en 1 para reducir los inicios en frío. [Más información](#)

- ☐ Interna
Permite el tráfico de VPC y ciertos servicios de Google Cloud en el proyecto, la VPC compartida, el balanceador de cargas HTTP(S) interno y el tráfico que permiten los Controles del servicio de VPC. [Más información](#)
- ☒ Todos
Permite el acceso directo a tu servicio desde Internet.

Autenticación * ?

- ☒ Permitir invocaciones sin autenticar
Marca esta opción cuando crees una API pública o un sitio web.
- ☐ Exigir autenticación
Administra los usuarios autorizados con Cloud IAM.

Contenedores, Herramientas de redes, seguridad



CONTENEDOR

HERRAMIENTAS DE REDES

SEGURIDAD



Cloud Run



Crear servicio

Contenedores, Herramientas de redes, seguridad



CONTENEDOR

HERRAMIENTAS DE REDES

SEGURIDAD

General

Puerto de contenedor

8000

Las solicitudes se enviarán al contenedor de este puerto. Recomendamos detectar en \$PORT, en lugar de en este número específico.

Comando de contenedor

Deja el campo en blanco para usar el comando de punto de entrada definido en la imagen de contenedor.

Argumentos de contenedor

Argumentos pasados al comando del punto de entrada.

- ☐ Aumento de CPU de inicio **VISTA PREVIA**
Asigna más capacidad de CPU durante el tiempo de inicio para iniciar los contenedores más rápido. [Más información](#)



Las verificaciones de estado se pueden configurar mediante YAML

[LEARN MORE](#)

Capacidad

Memoria

2 GiB

Memoria para asignar a cada instancia de contenedor.

CPU

1

Cantidad de CPU virtuales asignadas a cada instancia de contenedor.



Cloud Run



Crear servicio

Capacidad

Memoria

2 GiB

Memoria para asignar a cada instancia de contenedor.

CPU

1

Cantidad de CPU virtuales asignadas a cada instancia del contenedor.

Tiempo de espera de la solicitud

300

seconds

Tiempo en el que se debe mostrar una respuesta (máximo de 3600 segundos).

Cantidad máxima de solicitudes por contenedor

80

La cantidad máxima de solicitudes simultáneas que pueden llegar a cada instancia de contenedor. [¿Qué es la simultaneidad?](#)

Entorno de ejecución

El entorno de ejecución en el que se ejecuta tu contenedor. [Más información](#)



Predeterminada

Cloud Run seleccionará un entorno de ejecución adecuado para ti.



Primera generación

Inicios en frío más rápidos.



Segunda generación

Compatibilidad con el sistema de archivos de red, compatibilidad total con Linux, CPU y rendimiento de red más rápidos.

Variables del entorno

+ AGREGAR VARIABLE

Secretos ?

CREA UNA REFERENCIA A UN SECRET

- **Luego click a Crear**

Paso 5:

- Una vez creado ya podremos ver en Cloud Run el servicio activo
- Si entramos en el mismo se puede ver la URL del mismo a la que solo falta ingresar y agregarle “/docs” en la barra de direcciones para acceder a la API
- Con esta configuración permitimos 10 conexiones simultáneas y que se prenda el servicio únicamente cuando se accede, luego se apaga automáticamente hasta la siguiente conexión, evitando costos.