

## ## TelegramBotGCP-UI

```
`TelegramBotGCP-UI es una plataforma chat brindada a operadores para
gestionar conversaciones con los clientes de Moviestar`
```

La arquitectura de la aplicación está montada en Google Cloud el cual brindará escalabilidad, microservicios sin servidor, funcionalidades desacopladas y administración por eventos.

Todo esto nos permite generar un proyecto que a la larga se convertirá en un proyecto fácil de mantener y abaratado por el paradigma cloud pay as you go.

### ### DEMO DEPLOYED

## BACKEND

<https://bot-service-36gz5wrlea-ue.a.run.app>

## FRONTEND

`https://ui-service-36gz5wrlea-ue.a.run.app`

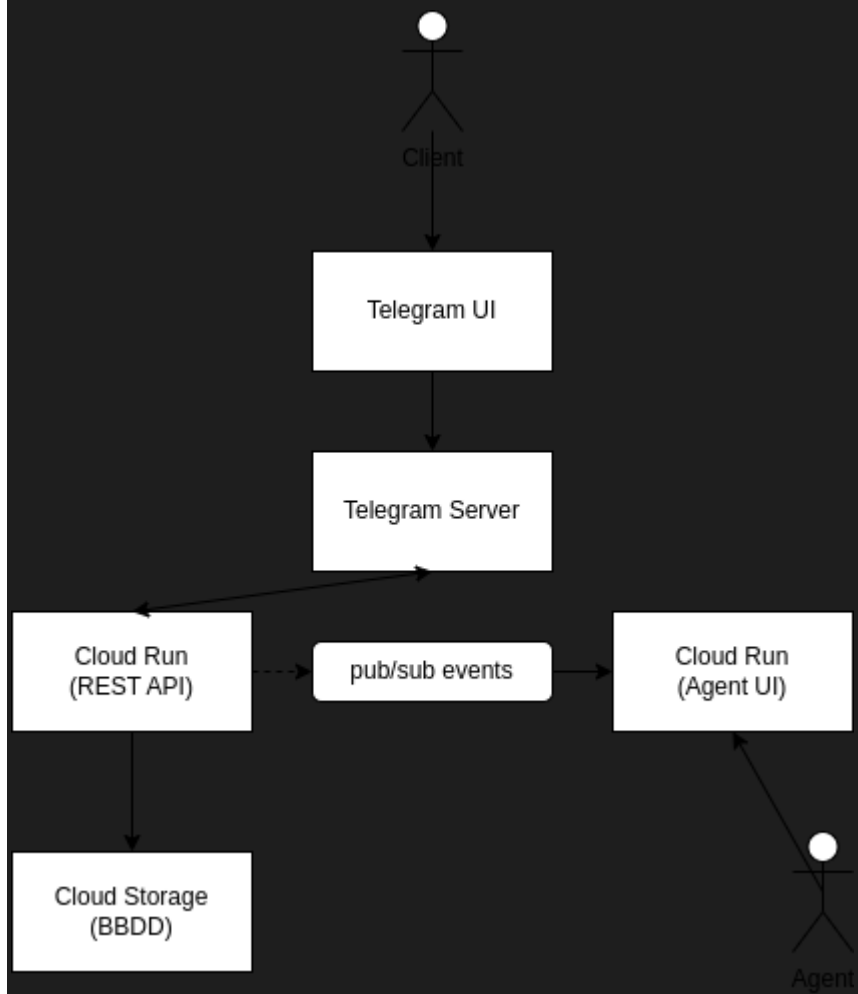
### ### HOW TO RUN

En la terminal de google cloud dentro de un proyecto ya creado, ejecute el script que desplegará todos los servicios.

```
git clone https://github.com/marc459/TelegramBotGCP-UI && cd TelegramBotGCP-UI && sh deploy-all.sh
```

<input type="checkbox"/>		Name	Req/sec	Region	Authentication	Ingress
<input type="checkbox"/>		<a href="#">bot-service</a>	0	us-east1	Allow unauthenticated	All
<input type="checkbox"/>		<a href="#">ui-service</a>	0.01	us-east1	Allow unauthenticated	All

### ### DIAGRAMA



### ## SWAGGERS

```
openapi: 3.0.0
info:
  title: WebSocket API
  version: 1.0.0
servers:
  - url: ws://ui-service-36gz5wrlea-ue.a.run.app
    description: Development server
paths:
  /ws:
    post:
      summary: Establece una conexión WebSocket
      operationId: connectWebSocket
      responses:
```

```
        '200':
          description: OK
    /ws/{id}:
      parameters:
        - name: id
          in: path
          required: true
          description: ID de la conexión WebSocket
          schema:
            type: string
      delete:
        summary: Cierra una conexión WebSocket
        operationId: closeWebSocket
      responses:
        '200':
          description: OK
```

### ### BACKEND

El backend esta conformado por un cloud run que mantiene una conexion con la api de telegram, el cual proporcionara recibir y enviar mensajes con los clientes.

En efecto es una API REST que procesará los mensajes los guardara en Cloud Storage de google cloud y los expondra en un tema pub/sub para notificar al resto de servicios.

### ### FRONTEND

El frontend esta conformado por un cloud tun, y se encarga de visualizar los mensajes de los clientes que solicitan informacion a través de telegram.

Se trata de establecer una conexion a traves de web sockets con el backend.

### ### BIBLIOGRAFÍA

<https://www.cloudskillsboost.google/focuses/8389?parent=catalog>

<https://cloud.google.com/run/docs/tutorials/pubsub?hl=es-419#command-line>

<https://medium.com/@iacomini.riccardo/build-a-serverless-telegram-bot-on-gcp-with-cloud-run-8d4ec9080b0f>

<https://github.com/meteatamel/knative-tutorial/blob/master/docs/deploycloudrun.md>