

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Sistemas de Bases de Datos 1

## MANUAL DE TECNICO

Nombre: Juan Daniel Enrique Roman Barrientos    Registro Academico: 201801364

Instructor: Luis Espino

Sección de Laboratorio: B


Auxiliar: Jonnathan Castillo

Fecha de Entrega: 03-05-2021


# MANUAL TECNICO

## Creación Máquina Virtual:


1. Utilizaremos GCP, ya que esta pagina nos proporciona 300 credits gratis.
2. Utiizamos las siguientes configuraciones para crear la maquina virtual

**Nombre** 

El nombre es permanente


**Etiquetas**  (Opcional)

+ Agregar etiqueta

**Región** 

La región es permanente

us-central1 (Iowa)

**Zona** 

La zona es permanente

us-central1-a

**Configuración de la máquina**

**Familia de máquinas**

Uso general

Optimizada para procesamiento

Memoria optimizada

GPU

Tipos de máquinas para cargas de trabajo comunes, optimizados en función del costo y la flexibilidad


**Series**

E2

Selección de la plataforma de CPU según la disponibilidad

**Tipo de máquina**

e2-medium (2 CPU virtuales, 4 GB de memoria)



vCPU


1 núcleo compartido


Memoria

4 GB


GPU

-

 Plataforma de CPU y GPU


**Servicio de VM confidencial** 


☐ Habilita el servicio de procesamiento confidencial en esta instancia de VM.

**Contenedor** 

☐ Implementa una imagen de contenedor en esta instancia de VM. [Más](#)

Disco de arranque



Nuevo disco persistente equilibrado de 50 GB  
Imagen  
 Windows Server 2019 Datacenter

Cambiar

Si usas Windows y planeas ejecutar un software de Microsoft adicional, llena el [formulario de verificación de licencias](#).

[Obtén más información](#) sobre los requisitos de movilidad de licencias de Microsoft.

Identidad y acceso a la API

Cuenta de servicio

Compute Engine default service account

Permiso de acceso

☐ Permitir el acceso predeterminado  
☒ Permitir el acceso total a todas las API de Cloud  
☐ Configurar el acceso para cada API

Firewall

Agregar etiquetas y reglas de firewall para permitir un tráfico de red determinado desde Internet

☒ Permitir tráfico HTTP  
☒ Permitir tráfico HTTPS

Administración, seguridad, discos, redes, usuario único

Se aplicarán cargos por esta instancia. [Precios de Compute Engine](#)

Crear

Cancelar

[REST](#) o [línea de comandos](#) equivalente

- Una vez creada la maquina nos saldrá la ip con la que podremos acceder mediante conexión remota

<input type="checkbox"/>	<input checked="" type="checkbox"/>	proyecto2-bd1-1s	us-east1-d	10.142.0.2 (nic0)	34.74.1.140	RDP	
--------------------------	-------------------------------------	------------------	------------	-------------------	-------------	-----	--

## Configuración Backend:

1. Instalamos nodejs
2. Inicializamos nuestro proyecto con el siguiente comando  
npm init --yes
3. Instalamos las siguientes dependencias  
npm install body-parser cors express morgan mysql
4. Realizamos las siguientes configuraciones para que escuche a la ip publica que nos proporciona la máquina virtual.

```
You, 7 hours ago | 1 author (You)
const express = require("express");
const morgan = require("morgan");
const cors = require("cors");
const bodyParser = require("body-parser");
const mysql = require("mysql");
let app = express();

✓ let conexion = mysql.createConnection({
  host: "0.0.0.0",
  user: "root",
  password: "JuanAnonymo2000@",
  database: "proyecto2",
});

app.set("port", process.env.PORT || 3000);
app.use(bodyParser.json());
app.use(cors());
app.use(morgan("dev"));

✓ app.use(
✓   express.urlencoded({
     extended: true,
   })
);

/*****
✓ app.listen(app.get("port"), "0.0.0.0", () => {
  console.log("Listening on", app.get("port"));
});
```

## Deploy App:

1. Creamos otra maquina virtual solo que ahora con sistema operativo Linux
2. Instalamos Docker con el siguiente comando  
`sudo apt-get install docker.io`
3. Preparamos un dockerfile para la configuración de la app y tiene la siguiente estructura.

```
client > dockerfile
1 FROM node:14.9 as build
2 WORKDIR /fronted
3 ENV PATH /fronted/node_modules/.bin:$PATH
4 COPY package*.json ./
5 RUN npm install --silent
6 RUN npm install -g @angular/cli@8.1.3 ionic@4.1.0 cordova --silent
7
8 COPY . .
9
10 RUN npm run build
11
12 FROM nginx:stable-alpine
13 COPY --from=build /fronted/www /usr/share/nginx/html
14 EXPOSE 80
15 CMD ["nginx", "-g", "daemon off;"]
16
```

4. Creamos una imagen de este dockerfile con el siguiente comando  
`sudo docker build -t appweb:v1`
5. Una vez creada la imagen, la preparamos para ponerla en la nube, para esto es necesario publicar el contenedor en la nube y se hace de la siguiente manera  
`gcloud auth configure-docker`  
`docker tag appweb:v1 gcr.io/halogen-ethos-310222/appweb:v1`  
`docker push gcr.io/halogen-ethos-310222/appweb`
6. Verificamos que nuestro contenedor se encuentre en la nube

Nombre ^	Nombre de host	Visibilidad ?
 appweb	gcr.io	Público

## 7. Configuramos Google Cloud Run de la siguiente manera

### 1 Configuración del servicio

Cada servicio expone un extremo único y ajusta de forma automática la escala de la infraestructura subyacente para controlar las solicitudes entrantes. El nombre de la plataforma de implementación y el servicio no se pueden cambiar.

Nombre del servicio \*

app-web

Plataforma de implementación ?

☒ Cloud Run (completamente administrado)

Región \*

us-central1 (Iowa)

[¿Cómo se selecciona la región?](#)

☐ Cloud Run for Anthos

SIGUIENTE

### 2 Configurar la primera revisión del servicio

Un servicio puede tener varias revisiones. La configuración de cada revisión es inmutable.

☒ Implementar una revisión desde una imagen de contenedor

URL de la imagen del contenedor \*

gcr.io/halogen-ethos-310222/appweb@sha256:46975e

SELECCIONAR

P. ej., us-docker.pkg.dev/cloudrun/container/hello  
debe detectar solicitudes HTTP en el puerto \$PORT y no depender de un estado local. [¿Cómo se compila un contenedor?](#)

☐ Implementar continuamente revisiones nuevas desde el repositorio de código fuente

Configuración avanzada



SIGUIENTE

## Configuración avanzada



### CONTENEDOR

### VARIABLES

### CONEXIONES

#### General

Puerto de contenedor

80

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.

Cuenta de servicio

Compute Engine default service account

Identidad que debe usar la revisión creada.

#### Capacity

Memoria asignada

256 MiB

Memoria para asignar a cada instancia de contenedor.

Asignación de 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?](#)

#### Ajuste de escala automático ?

Número mínimo de instancias \*

0

Número máximo de instancias

100

SIGUIENTE

✓ Configuración del servicio

✓ Configurar la primera revisión del servicio

3 Configura la forma en que se activa el servicio

Un servicio se puede invocar directamente o a través de eventos. Haz clic en "Agregar activador de Eventarc" para crear un nuevo activador basado en eventos.

[Más información](#)

**Entrada ?**

- ☒ Permitir todo el tráfico
- ☐ Permitir el tráfico interno y el proveniente de Cloud Load Balancing
- ☐ Permitir solo el tráfico interno

**Autenticación \* ?**

- ☒ Permitir invocaciones sin autenticar  
Marca esta opción si estás creando una API pública o un sitio web.
- ☐ Autenticación obligatoria  
Administrar los usuarios autorizados con Cloud IAM

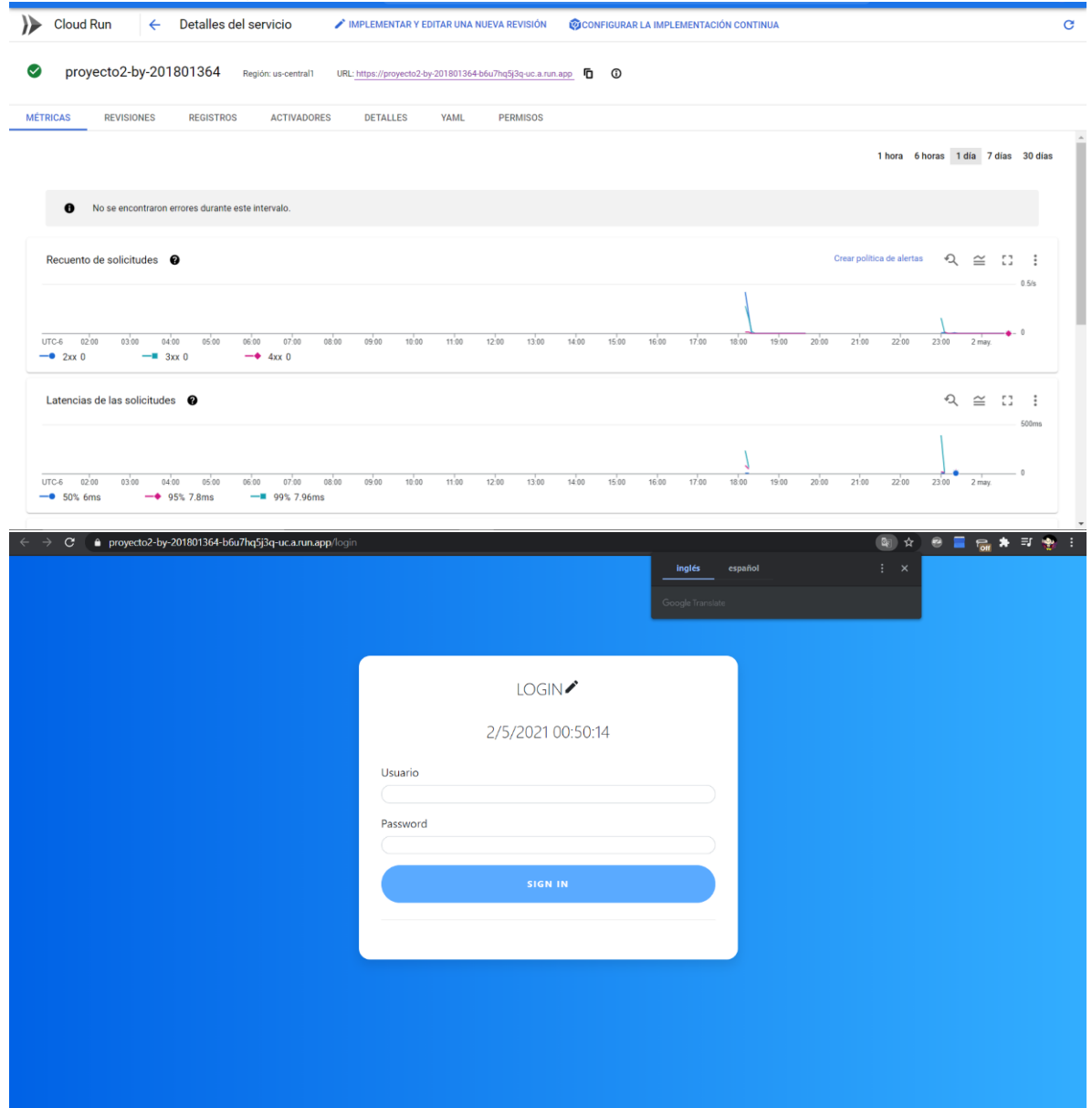
[+ AGREGAR ACTIVADOR DE EVENTARC](#)

CREAR

CANCELAR



8. Una configurado le damos en crear y nos muestra el url de la pagina quedando de la siguiente manera



**LINK PAGINAS:**

**GCP:**

<https://proyecto2-by-201801364-b6u7hq5i3q-uc.a.run.app/>

**AWS:**

<http://ace2-practica1-prenda-1s2021.s3-website.us-east-2.amazonaws.com/>