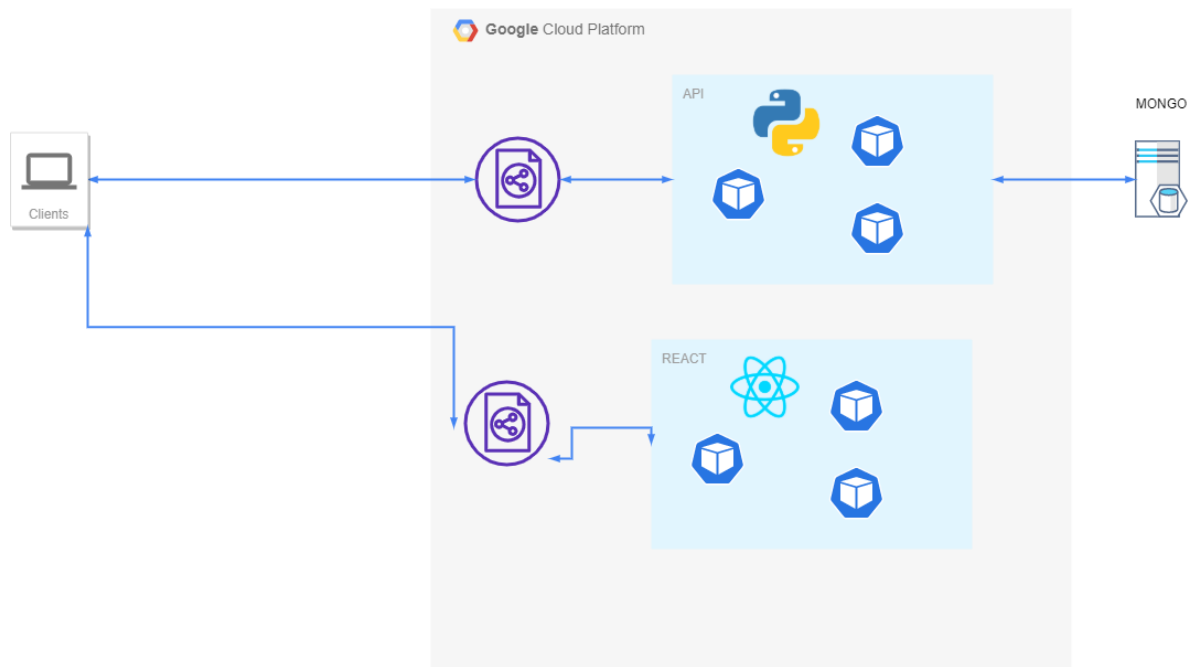


EXAMEN FINAL

ANDHY LIZANDRO SOLIS OSORIO 201700886

Arquitectura de la implementacion del tercer parcial



La arquitectura se desarrolló utilizando 2 nubes, la primera nube es Google cloud que se encarga de ejecutar las aplicaciones a través de contenedores utilizando Kubernetes, la otra nube es AWS que se encarga de ejecutar en una instancia EC2 la base de datos de mongo.

Del lado de Google cloud se implemento:

- 2 balanceadores de cargas para exponer los pods hacia el internet, estos balanceadores de carga se encargan de distribuir las tareas entre cada pod, para así poder tener una alta Disponibilidad
- Cada pod contenía solamente un contenedor, los pods del servidor web tenían una imagen de React para poder visualizar la imagen, en el caso de la Api se colocaron contenedores que ejecutaban Python con el servidor de Flask
- La base de datos se encontraba en la nube utilizando el servicio de ec2 para ejecutar mongo, utilizando un contenedor y Docker

La Aplicación

Servidor Web 1

Username

Username

Crear Usuario

Password

Password

Loggearse

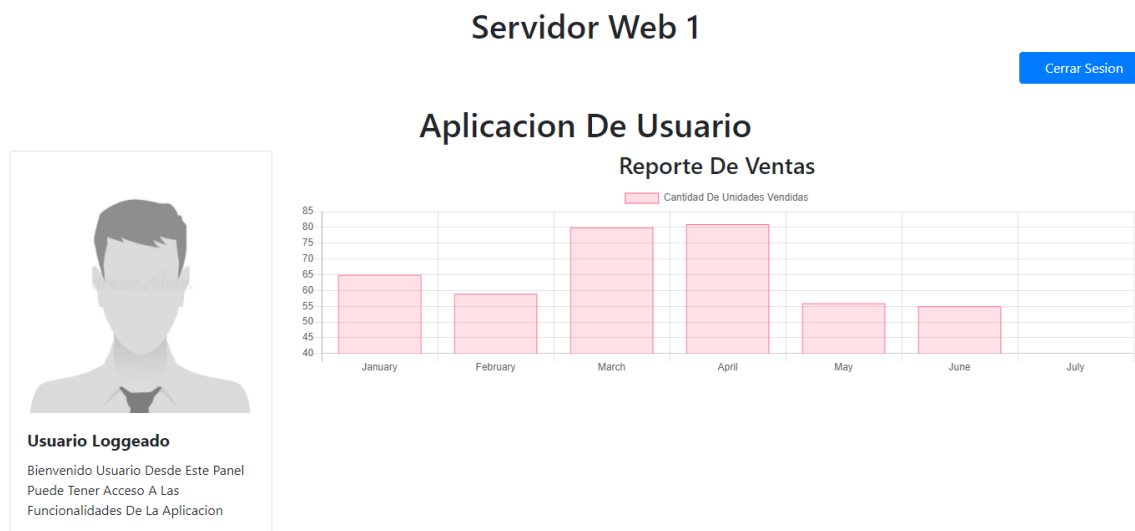
No Se Ha Encontrado Un Usuario

No se ha loggeado en la pagina

La aplicación web corre en el puerto 80 para funcionar con http, el sistema de logeo esta administrado por amazon incognito



Se utilizo Amazon cognito ya que permite almacenar toda la información sensible del usuario como lo es las contraseñas, manejarlas por nosotros utilizando la capa gratuita se pudo manejar las sesiones de usuario, para conectar Amazon cognito con React se utilizo una librería que la misma Amazon creo para que las aplicaciones de JavaScript tengan acceso a esa funcionalidad.



El menú principal luego de haber pasado la autenticación de Amazon cognito

Para comunicarse con la API se utilizo Postman, utilizando peticiones get y post

Postman interface showing a POST request to 104.155.139.111:5000/mongo?mensaje=Soy Un Mensaje&usuario=Juanito. The response is "Se Cargo Dato Exitosamente".

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> mensaje	Soy Un Mensaje	
<input checked="" type="checkbox"/> usuario	Juanito	
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 330 ms Size: 174 B Save Response

Pretty Raw Preview Visualize JSON

```
1 "Se Cargo Dato Exitosamente"
```

Postman con la petición post para cargar los datos

Postman interface showing a GET request to 104.155.139.111:5000/mongo?mensaje=Soy Un Mensaje&usuario=Juanito. The response is a JSON object containing the stored data.

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> mensaje	Soy Un Mensaje	
<input checked="" type="checkbox"/> usuario	Juanito	
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 320 ms Size: 232 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "1": {
3     "usuario": "Juanito",
4     "mensaje": "Soy Un Mensaje"
5   }
6 }
```

Petición get para obtener los datos almacenados.