

Comandos utilizados

Para realizar el deployment de cada aplicación, se utilizó el comando **kubectl apply -f archivo.yaml**

Estructura de un archivo .yaml utilizado en el proyecto:

```
api-node > cat api-node.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: apinode
5    namespace: solp2
6    labels:
7      role: apinode
8  spec:
9    replicas: 2
10   selector:
11     matchLabels:
12       role: apinode
13   minReadySeconds: 20
14   strategy:
15     type: RollingUpdate
16     rollingUpdate:
17       maxUnavailable: 1
18       maxSurge: 1
19   template:
20     metadata:
21       labels:
22         role: apinode
23     spec:
24       containers:
25       - name: apinode
26         image: sergiol0rpr/apinodejs:latest
27         imagePullPolicy: Always
28         ports:
29         - containerPort: 3003
30   ---
31   apiVersion: v1
32   kind: Service
33   metadata:
34     name: apinode
35     namespace: solp2
36   spec:
37     type: LoadBalancer
38     ports:
39     - port: 3004
40       targetPort: 3003
41     selector:
42       role: apinode
```

Para listar los servicios en ejecución dentro del clúster se utilizó el comando **kubectl get services -n nombreNamespace**. Ejemplo:

En este caso el namespace que generamos se llama solp2.

```
mrrobot10@mrrobot10:~$ kubectl get services -n solp2
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
apinode       LoadBalancer  10.44.11.156    35.224.108.103   3004:30023/TCP   3h44m
apiredis      LoadBalancer  10.44.11.1      34.27.186.83     5001:32556/TCP   2d5h
clientgrpc    LoadBalancer  10.44.4.93      34.30.141.43     3002:31320/TCP   2d16h
ingresstraffic LoadBalancer  10.44.7.195     35.223.250.249   4001:31841/TCP   4h46m
servergrpc    LoadBalancer  10.44.6.242     34.121.200.86    3001:30669/TCP   2d17h
```

Para listar los pods: **kubectl get pods -n so1p2**:

```
mrrobot10@mrrobot10:~$ kubectl get pods -n solp2
NAME                                READY   STATUS    RESTARTS   AGE
apinode-879bcbd7d-5wmpm             1/1     Running   0           3h51m
apinode-879bcbd7d-m49qv             1/1     Running   0           3h51m
apiredis-84f5b8cb9b-pzx6c          1/1     Running   0           9h
clientgrpc-7c88cf9669-j5vw4        1/1     Running   0           2d16h
ingresstraffic-f94f444b6-b8pw6      1/1     Running   0           4h54m
servergrpc-5f4c9968bb-4h648        1/1     Running   0           2d17h
```

grpc

gRPC es un framework de código abierto desarrollado por Google que se utiliza para crear servicios y aplicaciones de comunicación de alta eficiencia en redes. A continuación, se destacan algunas de las ventajas y características clave de gRPC:

1. Eficiencia en la comunicación: gRPC utiliza el Protocol Buffers (protobuf) como su lenguaje de descripción de interfaces, lo que permite una serialización binaria eficiente de datos. Esto reduce el tamaño de los datos transmitidos a través de la red y mejora la velocidad de transferencia.
2. Multiplataforma: gRPC es compatible con una variedad de lenguajes de programación, lo que facilita la creación de servicios y aplicaciones interoperables en diferentes entornos.
3. Protocolo HTTP/2: gRPC se basa en HTTP/2, lo que permite la multiplexación de múltiples solicitudes y respuestas en una sola conexión, reduciendo la latencia y mejorando el rendimiento.

Archivo proto utilizado:

```
grpcClient > proto > user.proto > ...
1  syntax = "proto3";
2
3
4  message User {
5      string id = 1;
6      string carnet = 2;
7      string nombre = 3;
8      string curso = 4;
9      string nota = 5;
10     string semestre = 6;
11     string year = 7;
12 }
13
14
15 service UserService {
16     rpc GetUser (UserRequest) returns (User);
17     rpc CreateUser (User) returns (User);
18 }
19
20 message UserRequest {
21     string id = 1;
22 }
```

Comando utilizado para generar el archivo pb.go : ***protoc proto/user.proto --go_out=plugins=grpc:.***

Script de la base de datos

```
CREATE TABLE nota(
idNota int auto_increment,
carnet varchar(25),
nombre varchar(100),
curso varchar(25),
nota int,
semestre varchar(25),
year varchar(25),
primary key(idNota)
);
```

Captura de la instancia:



SQL

Instancias

 CREAR INSTANCIA

 MIGRAR DATOS

[MOSTRAR PANEL DE INFORMACIÓN](#)

 Filtro

Ingresar el nombre o el valor de la propiedad





<input type="checkbox"/>	ID de Instancia  ↑	Edición de Cloud SQL	Tipo	Dirección IP pública	Dirección IP privada	Nombre de la conexión con la instancia	Alta disponibilidad	Acciones
<input type="checkbox"/>	 dbproyecto	Enterprise	MySQL 8.0	34.31.192.14		sopes1-403505:us-central1:dbproyecto	HABILITAR	

Cloud Run:

Google Cloud

SOA2S

regist

Buscar

17

Cloud Run

Detalles del servicio

IMPLEMENTAR Y EDITAR UNA NUEVA REVISIÓN

CONFIGURAR LA IMPLEMENTACIÓN CONTINUA

front-so1

Región: us-central1

URL: https://front-so1-jjvx62uz3a-uc.a.run.app

MÉTRICAS

SLO

REGISTROS

REVISIONES

HERRAMIENTAS DE REDES

SEGURIDAD

ACTIVADORES

INTEGRACIONES

VISTA PREVIA

YAML

Panel

Predefinido

1 hora

6 horas

1 día

7 días

30 días

CUSTOMIZE DASHBOARD

Cargas de Trabajo:

Kubernetes Engine

Cargas de tra...

ACTUALIZAR

IMPLEMENTAR

CREAR TRABAJO

NUEVO

BORRAR

OPERATIONS

Clúster

Espacio de nombres

RESTABLECER

SAVE

Las cargas de trabajo son unidades de procesamiento implementables que se pueden crear y administrar en un clúster.

DESCRIPCIÓN GENERAL

OBSERVABILIDAD

OPTIMIZACIÓN DE COSTOS

Filtro

Es objeto de sistema : Falso

Filtrar cargas de trabajo

<input type="checkbox"/>	Nombre	Estado	Tipo	Pods	Espacio de nombres	Clúster
<input type="checkbox"/>	apinode	OK	Deployment	2/2	so1p2	cluster-sopes
<input type="checkbox"/>	apiredis	OK	Deployment	1/1	so1p2	cluster-sopes
<input type="checkbox"/>	clientgrpc	OK	Deployment	1/1	so1p2	cluster-sopes
<input type="checkbox"/>	ingress-nginx-admission-create	BackoffLimitExceeded	Job	0/1	ingress-nginx	cluster-sopes
<input type="checkbox"/>	ingress-nginx-admission-patch	BackoffLimitExceeded	Job	0/1	ingress-nginx	cluster-sopes
<input type="checkbox"/>	ingress-nginx-controller	OK	Deployment	1/1	ingress-nginx	cluster-sopes
<input type="checkbox"/>	ingresstraffic	OK	Deployment	1/1	so1p2	cluster-sopes
<input type="checkbox"/>	servergrpc	OK	Deployment	1/1	so1p2	cluster-sopes