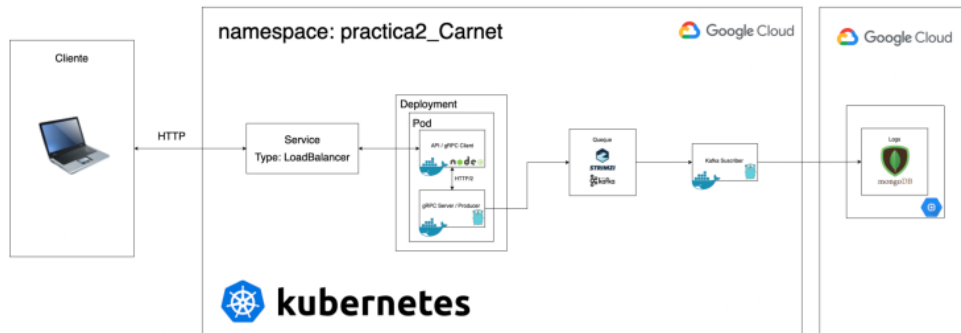


MANUAL TÉCNICO - 201901557

Arquitectura



Imágenes utilizadas

- jeffmendezdk/kafka_subs_201901557:latest
- jeffmendezdk/server_grpc_201901557:latest
- jeffmendezdk/api_client_grpc_201901557:latest

Deployment gRPC

Para la creación del deployment se utilizaron las dos imágenes subidas a docker hub del cliente y servidor de gRPC con una réplica.

```
# Deployment
apiVersion: apps/v1
kind: Deployment
metadata:
  name: grpc-deployment
  namespace: practica2-201901557
  labels:
    app: grpc-deployment
spec:
  selector:
    matchLabels:
      app: grpc-deployment
  replicas: 1
  template:
    metadata:
      labels:
        app: grpc-deployment
    spec:
      # Pod
      hostname: grpc-pod
      containers:
        - name: grpc-server
          image: jeffmendezdk/server_grpc_201901557:latest
          ports:
            - containerPort: 5505
        - name: grpc-client
          image: jeffmendezdk/api_client_grpc_201901557:latest
          ports:
            - containerPort: 4505
```

Servicio LoadBalancer

El servicio apunta al puerto 4505 del pod del cliente de gRPC.

```
# Service
apiVersion: v1
kind: Service
metadata:
  name: service-grpc
  namespace: practica2-201901557
  labels:
    app: grpc-deployment
spec:
  selector:
    app: grpc-deployment
  ports:
    - port: 4505
      targetPort: 4505
      protocol: TCP
  type: LoadBalancer
```

Instalación kafka

- kubectl create -f 'https://strimzi.io/install/latest?namespace=practica2-201901557' -n practica2-201901557
- kubectl apply -f https://strimzi.io/examples/latest/kafka/kafka-persistent-single.yaml -n practica2-201901557
- kubectl wait kafka/my-cluster --for=condition=Ready --timeout=300s -n kafka
- Creación topic juegos

```
apiVersion: kafka.strimzi.io/v1beta2
kind: KafkaTopic
metadata:
  name: juegos
  namespace: practica2-201901557
  labels:
    strimzi.io/cluster: my-cluster
spec:
  partitions: 1
  replicas: 1
```

Deployment subscriber

Para el subscriber de kafka no es necesario crear un servicio, el pod asignado es el 6505.

```
# Deployment
apiVersion: apps/v1
kind: Deployment
metadata:
  name: kafka-sub-deployment
  namespace: practica2-201901557
  labels:
    app: kafka-sub-deployment
spec:
  selector:
    matchLabels:
      app: kafka-sub-deployment
  replicas: 1
  template:
    metadata:
      labels:
        app: kafka-sub-deployment
    spec:
      # Pod
      hostname: subscriber-pod
      containers:
        - name: kafka-subscriber
          image: jeffmendezdk/kafka_subs_201901557:latest
          ports:
            - containerPort: 6505
```

Base de datos - MongoDB

Se creó una instancia, donde se le instaló docker y se ejecutó el siguiente docker-compose.

```
version: "3.9"
services:
  db:
    image: mongo
    container_name: dbmongo
    environment:
      MONGO_INITDB_ROOT_USERNAME: mongoadmin
      MONGO_INITDB_ROOT_PASSWORD: Solpass1S_2022
    ports:
      - "27017:27017"
    restart: always
    volumes:
      - mongodata:/data/db
    networks:
      - p2-201901557

volumes:
  mongodata:

networks:
  p2-201901557:
    name: "p2-201901557"
    driver: bridge
```