# DOCKER Y KUBERNETES INTERMEDIO

Proyecto Final del Curso

Integrantes

Quintana Cubas Lenin Alexander
Ponte Arica Anthony Rosemberg
Ramirez Vasquez Jonathan
Silva Ramos Juan Armando

# Contenido

# Proyecto Curso Docker y Kubernetes

## Entregables del Proyecto Final: Gestión de Eventos y Participantes

Los entregables se organizan en categorías que abarcan desde el código fuente hasta la documentación y los artefactos de despliegue.

Se creó el siguiente repositorio Git para todos los entregables del presente proyecto: https://github.com/jonathan0284/Curso_Docker_K8S.git

### 1. Código Fuente

### 1.1. Repositorios de Microservicios:

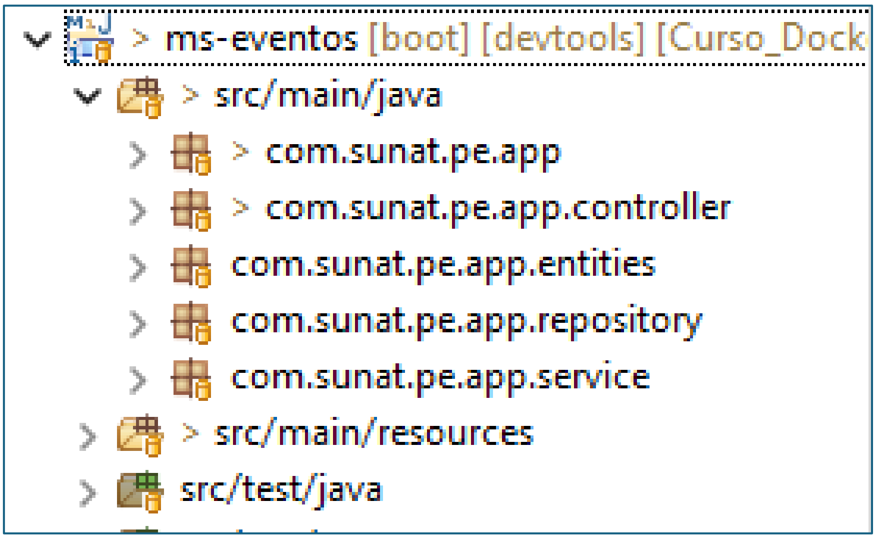
- Código fuente de los dos Micro servicios: gestión de eventos y gestión de inscripciones (participantes).

  ms-eventos:
  https://github.com/jonathan0284/Curso_Docker_K8S/tree/main/Microservicios/ms-eventos

  ms-participantes:
  https://github.com/jonathan0284/Curso_Docker_K8S/tree/main/Microservicios/ms-participantes

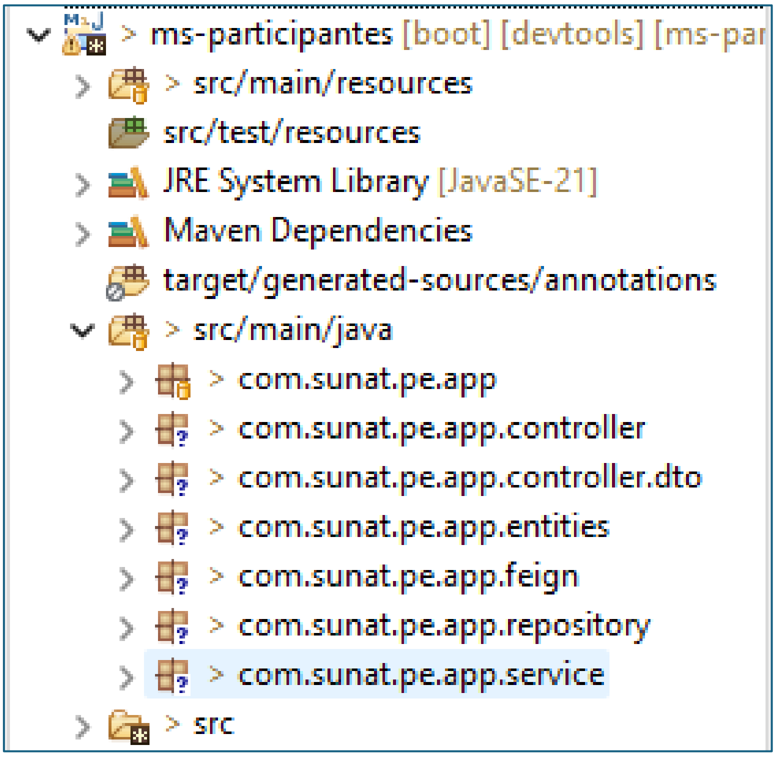
- Estructura del proyecto conforme a las mejores prácticas (separación en capas: controlador, servicio, repositorio, entidades, etc.).

**ms-eventos**



```
ms-eventos [boot] [devtools] [Curso_Dock
  src/main/java
    com.sunat.pe.app
    com.sunat.pe.app.controller
    com.sunat.pe.app.entities
    com.sunat.pe.app.repository
    com.sunat.pe.app.service
  src/main/resources
  src/test/java
```

**ms-participantes**



```
ms-participantes [boot] [devtools] [ms-par
  src/main/resources
  src/test/resources
  JRE System Library [JavaSE-21]
  Maven Dependencies
  target/generated-sources/annotations
  src/main/java
    com.sunat.pe.app
    com.sunat.pe.app.controller
    com.sunat.pe.app.controller.dto
    com.sunat.pe.app.entities
    com.sunat.pe.app.feign
    com.sunat.pe.app.repository
    com.sunat.pe.app.service
  src
```

○ Uso de control de versiones (Git) con un historial claro de commits: https://github.com/jonathan0284/Curso_Docker_K8S/activity

## 2. Artefactos Contenerizados

### 2.1. Imágenes Docker:

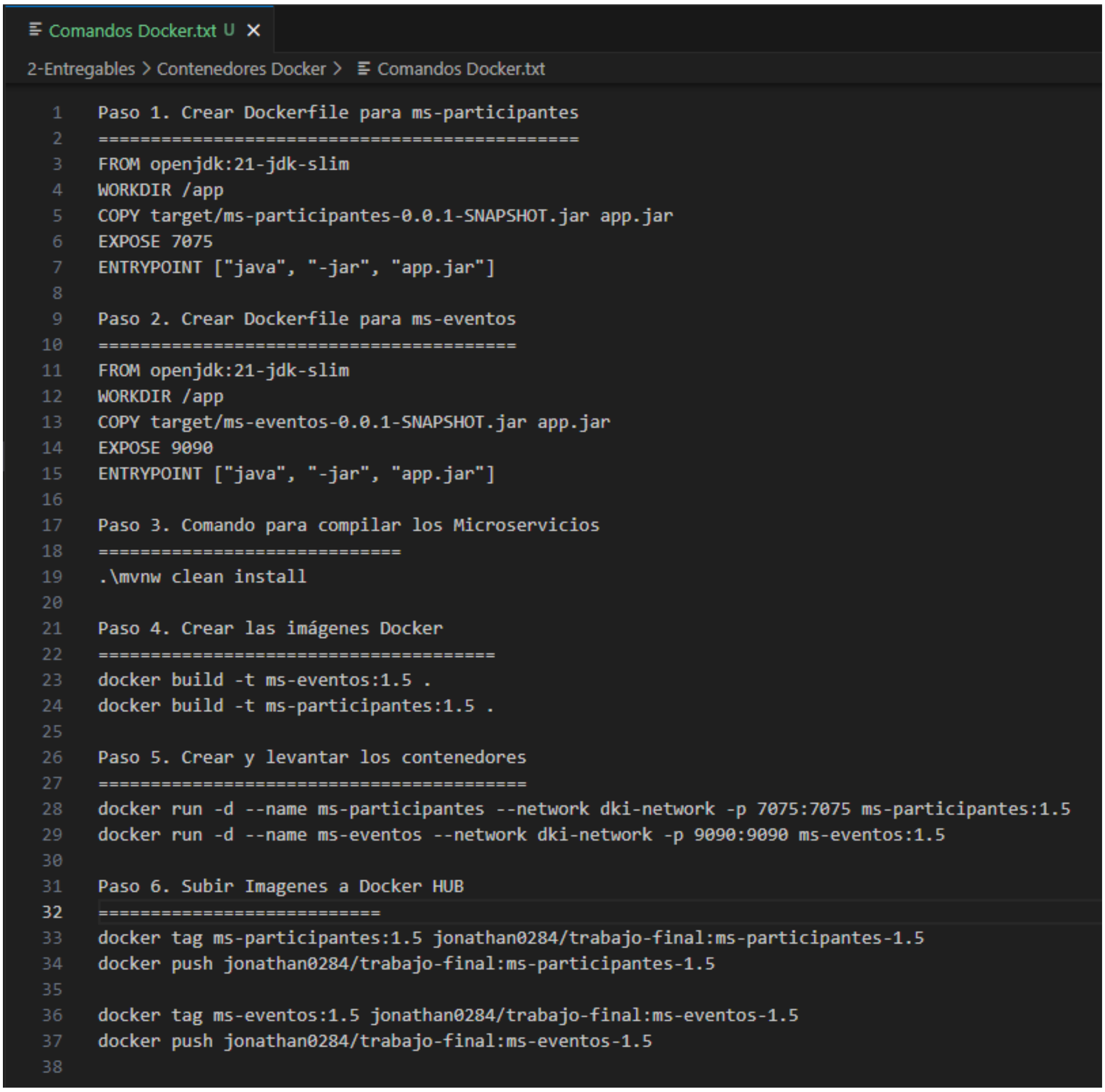
- Imágenes Docker de ambos microservicios, construidas y publicadas en un repositorio como Docker Hub.

Comandos Docker.txt

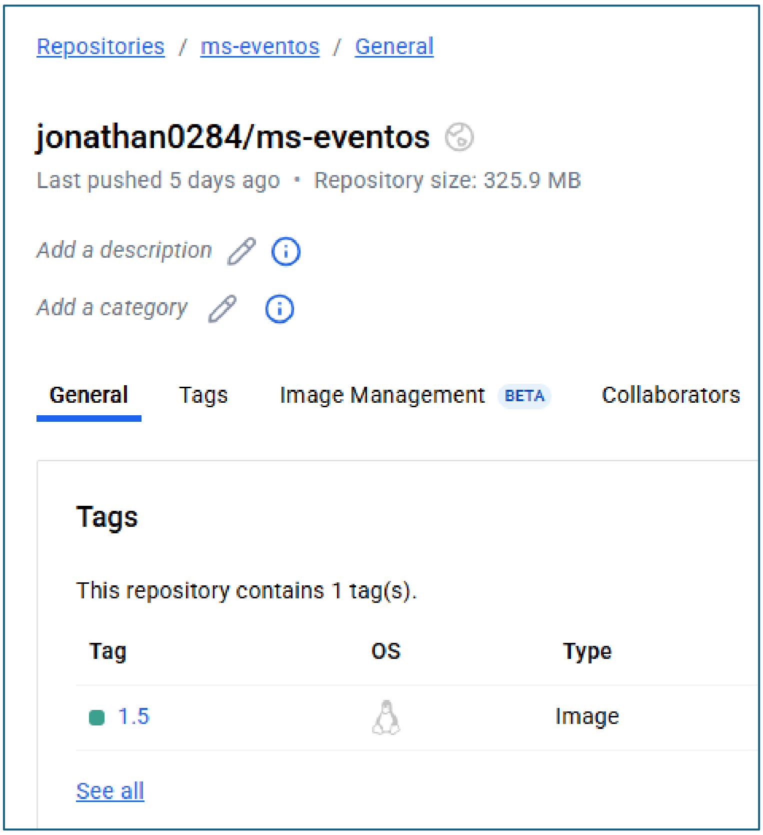
https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/contenedores%20docker/Comandos%20Docker.txt

**ms-eventos**
https://hub.docker.com/repository/docker/jonathan0284/ms-eventos

**ms-participantes**
https://hub.docker.com/repository/docker/jonathan0284/ms-participantes/general



```
Paso 1. Crear Dockerfile para ms-participantes
==============================================
FROM openjdk:21-jdk-slim
WORKDIR /app
COPY target/ms-participantes-0.0.1-SNAPSHOT.jar app.jar
EXPOSE 7075
ENTRYPOINT ["java", "-jar", "app.jar"]

Paso 2. Crear Dockerfile para ms-eventos
========================================
FROM openjdk:21-jdk-slim
WORKDIR /app
COPY target/ms-eventos-0.0.1-SNAPSHOT.jar app.jar
EXPOSE 9090
ENTRYPOINT ["java", "-jar", "app.jar"]

Paso 3. Comando para compilar los Microservicios
=============================
.\mvnw clean install

Paso 4. Crear las imágenes Docker
======================================
docker build -t ms-eventos:1.5 .
docker build -t ms-participantes:1.5 .

Paso 5. Crear y levantar los contenedores
=========================================
docker run -d --name ms-participantes --network dki-network -p 7075:7075 ms-participantes:1.5
docker run -d --name ms-eventos --network dki-network -p 9090:9090 ms-eventos:1.5

Paso 6. Subir Imagenes a Docker HUB
===========================
docker tag ms-participantes:1.5 jonathan0284/trabajo-final:ms-participantes-1.5
docker push jonathan0284/trabajo-final:ms-participantes-1.5

docker tag ms-eventos:1.5 jonathan0284/trabajo-final:ms-eventos-1.5
docker push jonathan0284/trabajo-final:ms-eventos-1.5
```



Repositories / ms-eventos / General

jonathan0284/ms-eventos

Last pushed 5 days ago • Repository size: 325.9 MB

*Add a description*

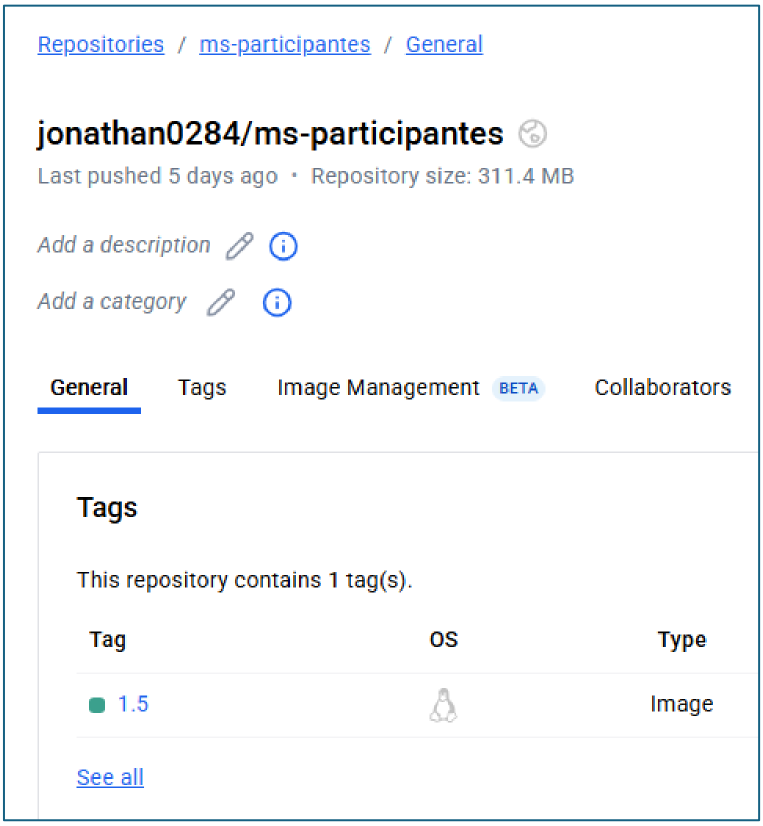
*Add a category*

Tags

This repository contains 1 tag(s).

| Tag | OS | Type |
|---|---|---|
| 🟩 1.5 | 🐧 | Image |

See all



Repositories / ms-participantes / General

jonathan0284/ms-participantes

Last pushed 5 days ago • Repository size: 311.4 MB

*Add a description*

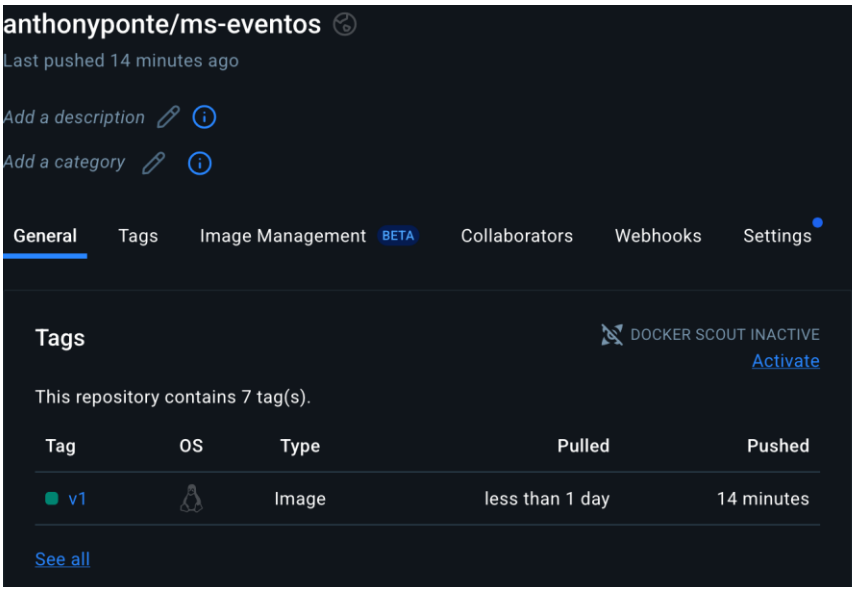
*Add a category*

Tags

This repository contains 1 tag(s).

| Tag | OS | Type |
|---|---|---|
| 🟩 1.5 | 🐧 | Image |

See all



anthonyponte/ms-eventos

Last pushed 14 minutes ago

Tags

DOCKER SCOUT INACTIVE
Activate

This repository contains 7 tag(s).

| Tag | OS | Type | Pulled | Pushed |
|---|---|---|---|---|
| v1 | | Image | less than 1 day | 14 minutes |

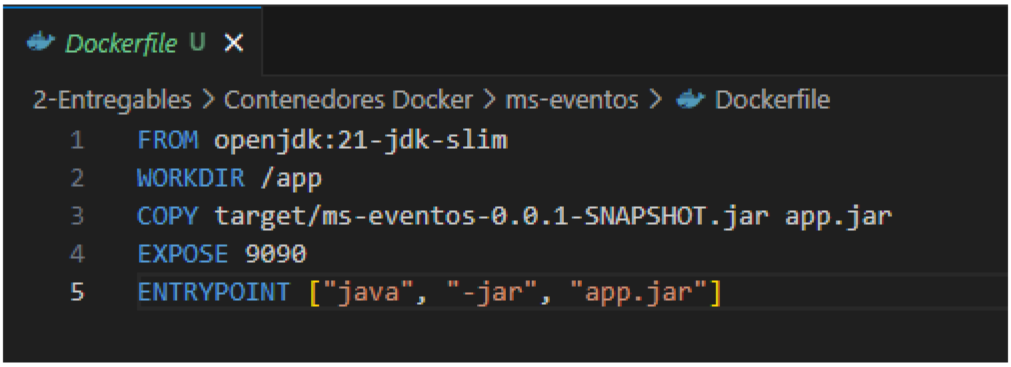
See all

- Imágenes optimizadas según las mejores prácticas (Dockerfile eficiente).

**ms-eventos**

Dockerfile

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/contenedores%20docker/ms-eventos/Dockerfile



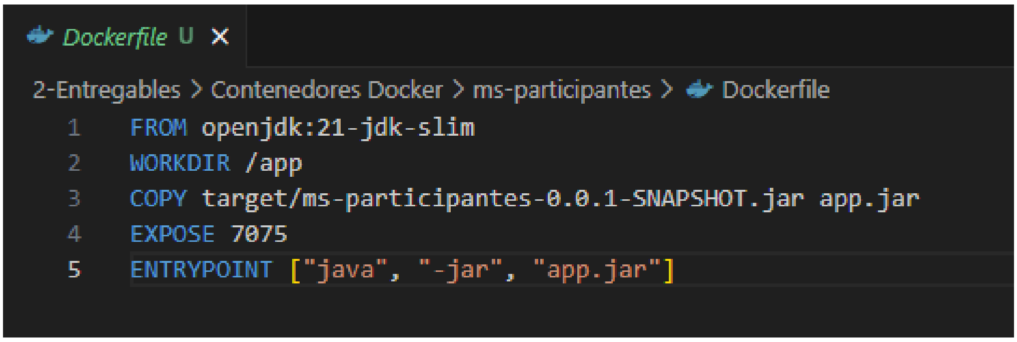
Dockerfile U ×
2-Entregables > Contenedores Docker > ms-eventos > Dockerfile

```
FROM openjdk:21-jdk-slim
WORKDIR /app
COPY target/ms-eventos-0.0.1-SNAPSHOT.jar app.jar
EXPOSE 9090
ENTRYPOINT ["java", "-jar", "app.jar"]
```

**ms-participantes**

Dockerfile

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/contenedores%20docker/ms-participantes/Dockerfile


Dockerfile U ×
2-Entregables > Contenedores Docker > ms-participantes > Dockerfile

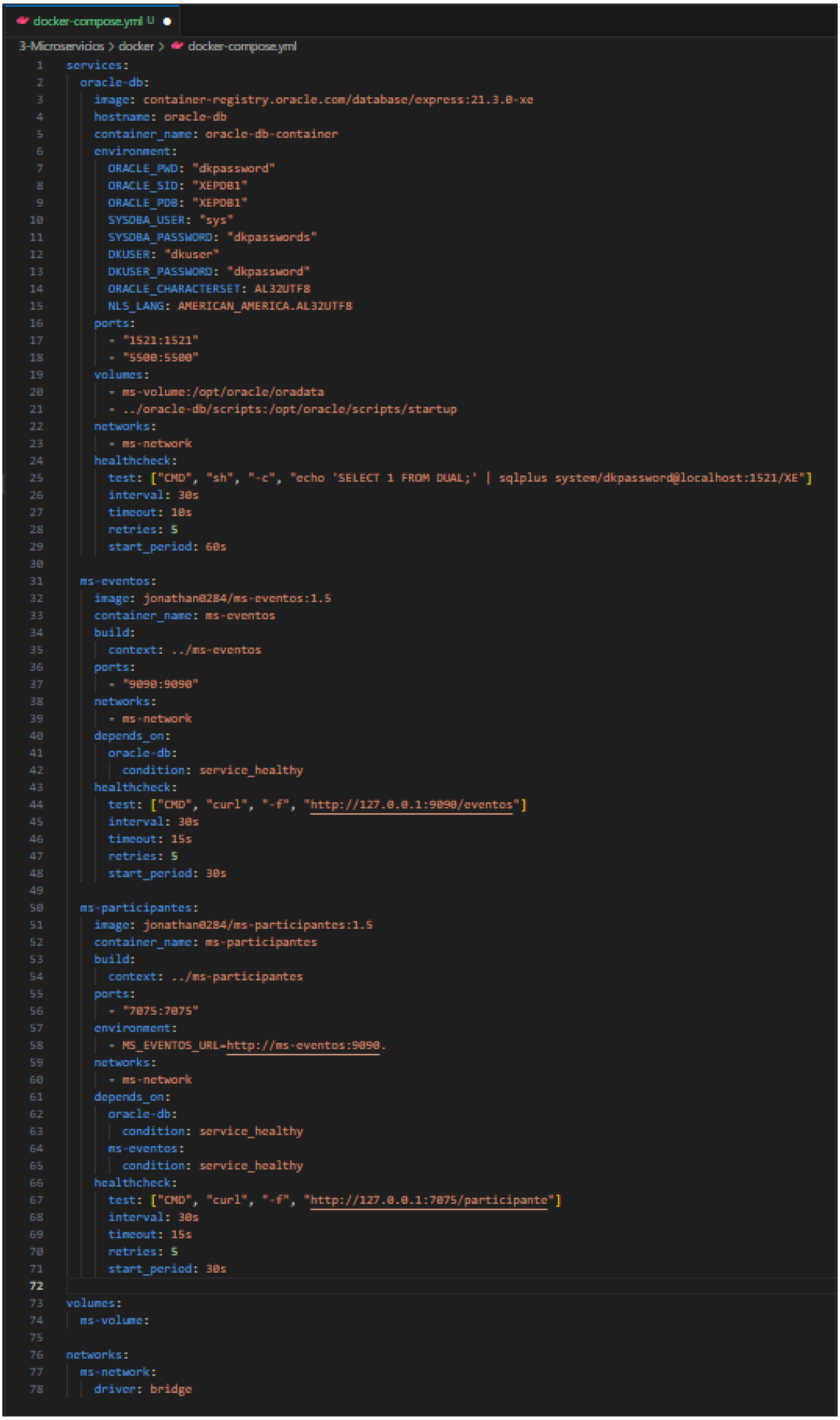
```
FROM openjdk:21-jdk-slim
WORKDIR /app
COPY target/ms-participantes-0.0.1-SNAPSHOT.jar app.jar
EXPOSE 7075
ENTRYPOINT ["java", "-jar", "app.jar"]
```

## 2.2. Docker Compose:

- Archivo docker-compose.yml para levantar ambos microservicios junto con sus bases de datos en un entorno de desarrollo local.

docker-compose.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/docker-compose/docker-compose.yml



```yaml
services:
  oracle-db:
    image: container-registry.oracle.com/database/express:21.3.0-xe
    hostname: oracle-db
    container_name: oracle-db-container
    environment:
      ORACLE_PWD: "dkpassword"
      ORACLE_SID: "XEPDB1"
      ORACLE_PDB: "XEPDB1"
      SYSDBA_USER: "sys"
      SYSDBA_PASSWORD: "dkpasswords"
      DKUSER: "dkuser"
      DKUSER_PASSWORD: "dkpassword"
      ORACLE_CHARACTERSET: AL32UTF8
      NLS_LANG: AMERICAN_AMERICA.AL32UTF8
    ports:
      - "1521:1521"
      - "5500:5500"
    volumes:
      - ms-volume:/opt/oracle/oradata
      - ../oracle-db/scripts:/opt/oracle/scripts/startup
    networks:
      - ms-network
    healthcheck:
      test: ["CMD", "sh", "-c", "echo 'SELECT 1 FROM DUAL;' | sqlplus system/dkpassword@localhost:1521/XE"]
      interval: 30s
      timeout: 10s
      retries: 5
      start_period: 60s

  ms-eventos:
    image: jonathan0284/ms-eventos:1.5
    container_name: ms-eventos
    build:
      context: ../ms-eventos
    ports:
      - "9090:9090"
    networks:
      - ms-network
    depends_on:
      oracle-db:
        condition: service_healthy
    healthcheck:
      test: ["CMD", "curl", "-f", "http://127.0.0.1:9090/eventos"]
      interval: 30s
      timeout: 15s
      retries: 5
      start_period: 30s

  ms-participantes:
    image: jonathan0284/ms-participantes:1.5
    container_name: ms-participantes
    build:
      context: ../ms-participantes
    ports:
      - "7075:7075"
    environment:
      - MS_EVENTOS_URL=http://ms-eventos:9090.
    networks:
      - ms-network
    depends_on:
      oracle-db:
        condition: service_healthy
      ms-eventos:
        condition: service_healthy
    healthcheck:
      test: ["CMD", "curl", "-f", "http://127.0.0.1:7075/participante"]
      interval: 30s
      timeout: 15s
      retries: 5
      start_period: 30s

volumes:
  ms-volume:

networks:
  ms-network:
    driver: bridge
```

# 3. Configuración para Kubernetes

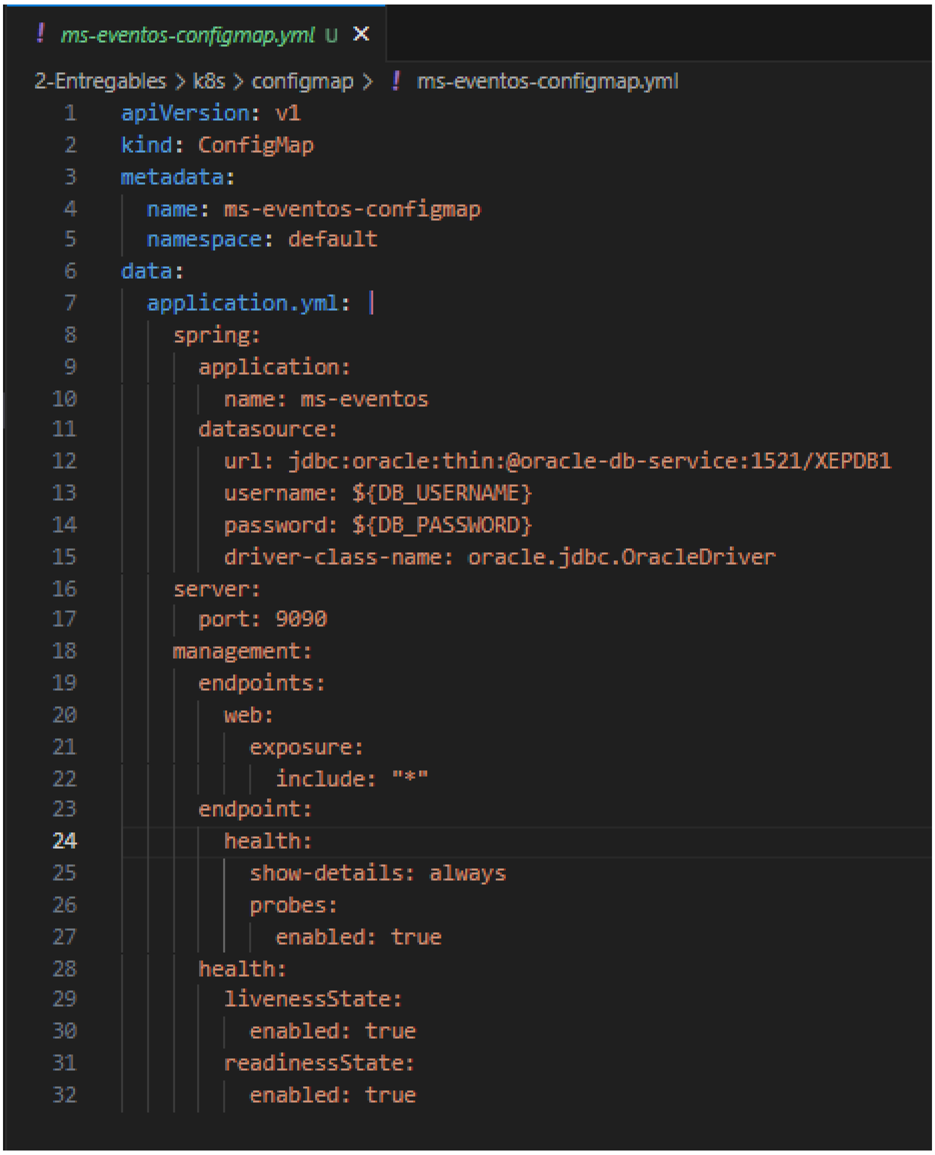
## 3.1. Archivos YAML:

- **Deployments:** Configuración de pods y réplicas para ambos microservicios.

  **ms-eventos-configmap.yml**

  ms-eventos-configmap.yml

  https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/configmap/ms-eventos-configmap.yml

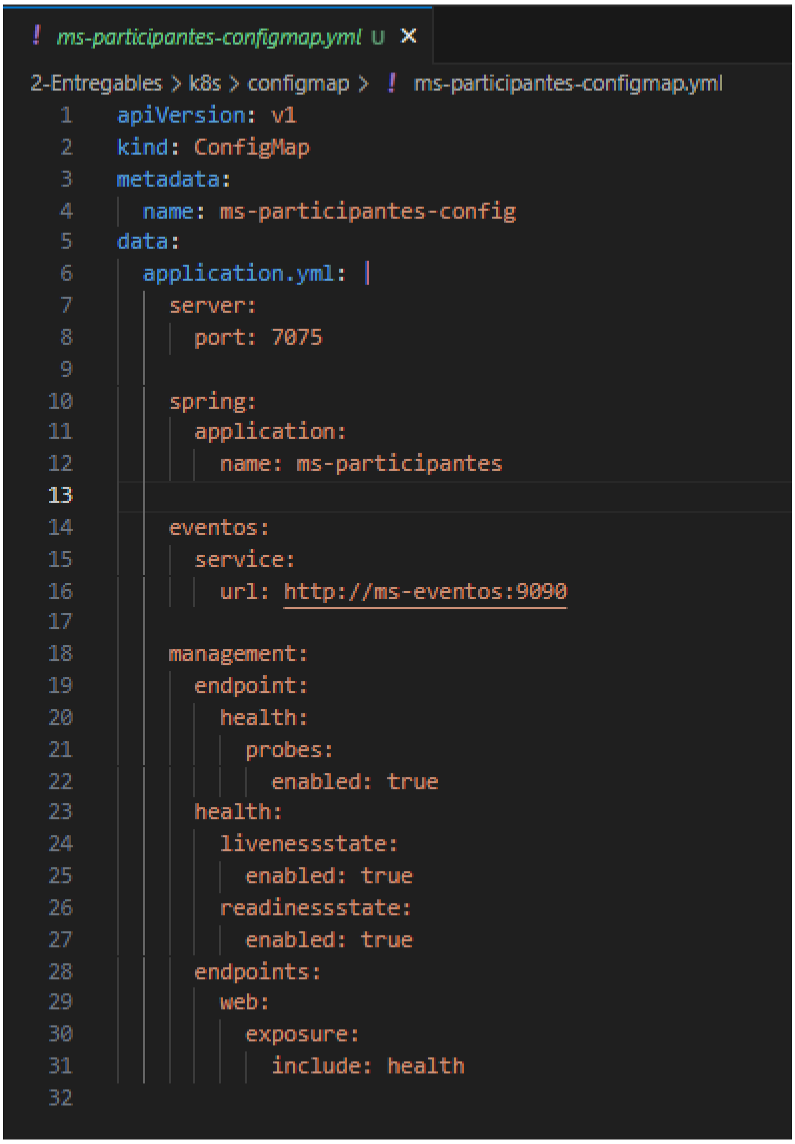

! ms-eventos-configmap.yml U ×
2-Entregables > k8s > configmap > ! ms-eventos-configmap.yml

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: ms-eventos-configmap
  namespace: default
data:
  application.yml: |
    spring:
      application:
        name: ms-eventos
      datasource:
        url: jdbc:oracle:thin:@oracle-db-service:1521/XEPDB1
        username: ${DB_USERNAME}
        password: ${DB_PASSWORD}
        driver-class-name: oracle.jdbc.OracleDriver
    server:
      port: 9090
    management:
      endpoints:
        web:
          exposure:
            include: "*"
      endpoint:
        health:
          show-details: always
          probes:
            enabled: true
      health:
        livenessState:
          enabled: true
        readinessState:
          enabled: true
```

## ms-participantes-configmap.yaml

ms-participantes-configmap.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/configmap/ms-participantes-configmap.yml



! ms-participantes-configmap.yml U ×

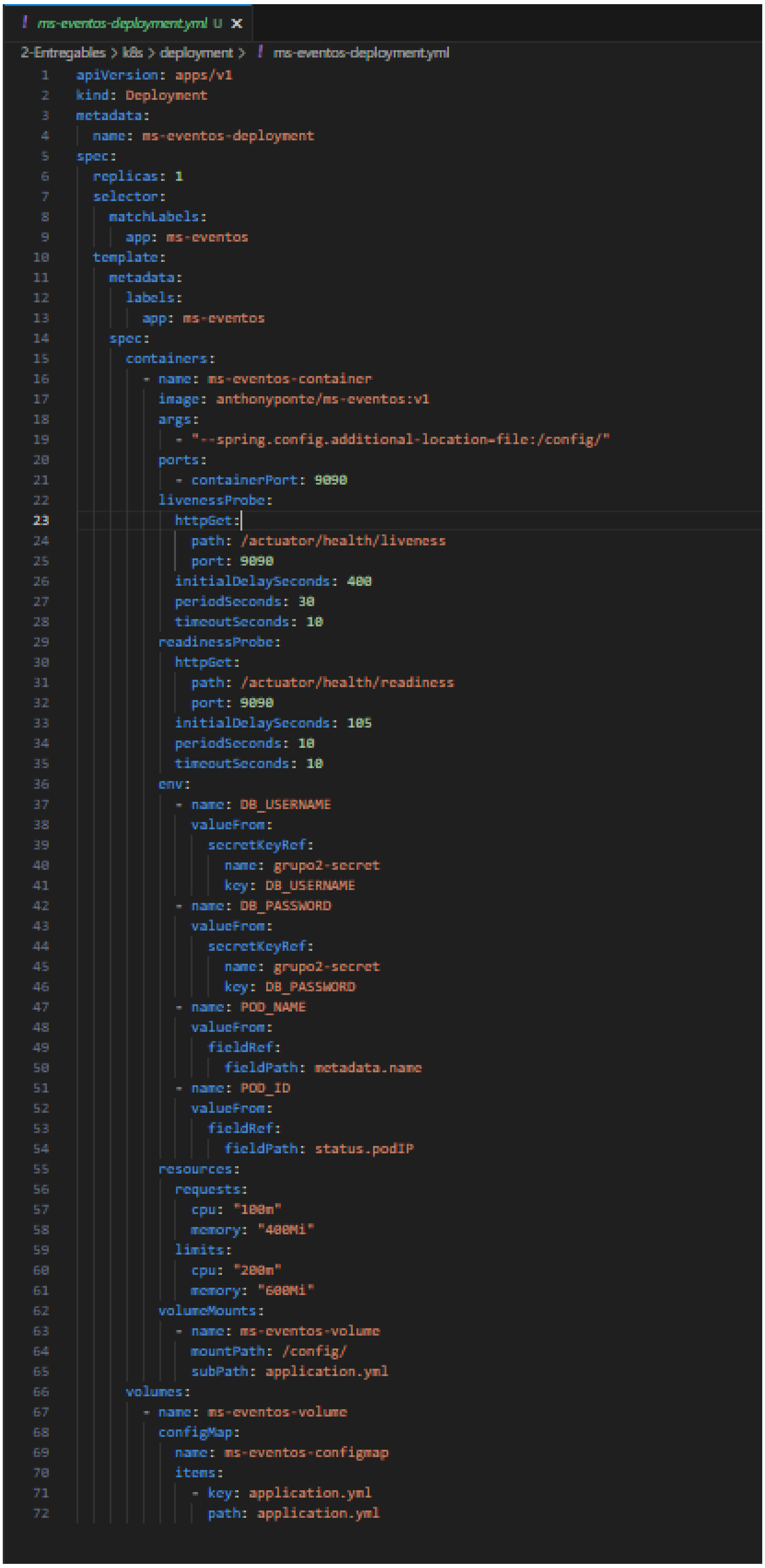
2-Entregables > k8s > configmap > ! ms-participantes-configmap.yml

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: ms-participantes-config
data:
  application.yml: |
    server:
      port: 7075

    spring:
      application:
        name: ms-participantes

    eventos:
      service:
        url: http://ms-eventos:9090

    management:
      endpoint:
        health:
          probes:
            enabled: true
      health:
        livenessstate:
          enabled: true
        readinessstate:
          enabled: true
      endpoints:
        web:
          exposure:
            include: health

```

## ms-eventos-deployment.yml

ms-eventos-deployment.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/deployment/ms-eventos-deployment.yml

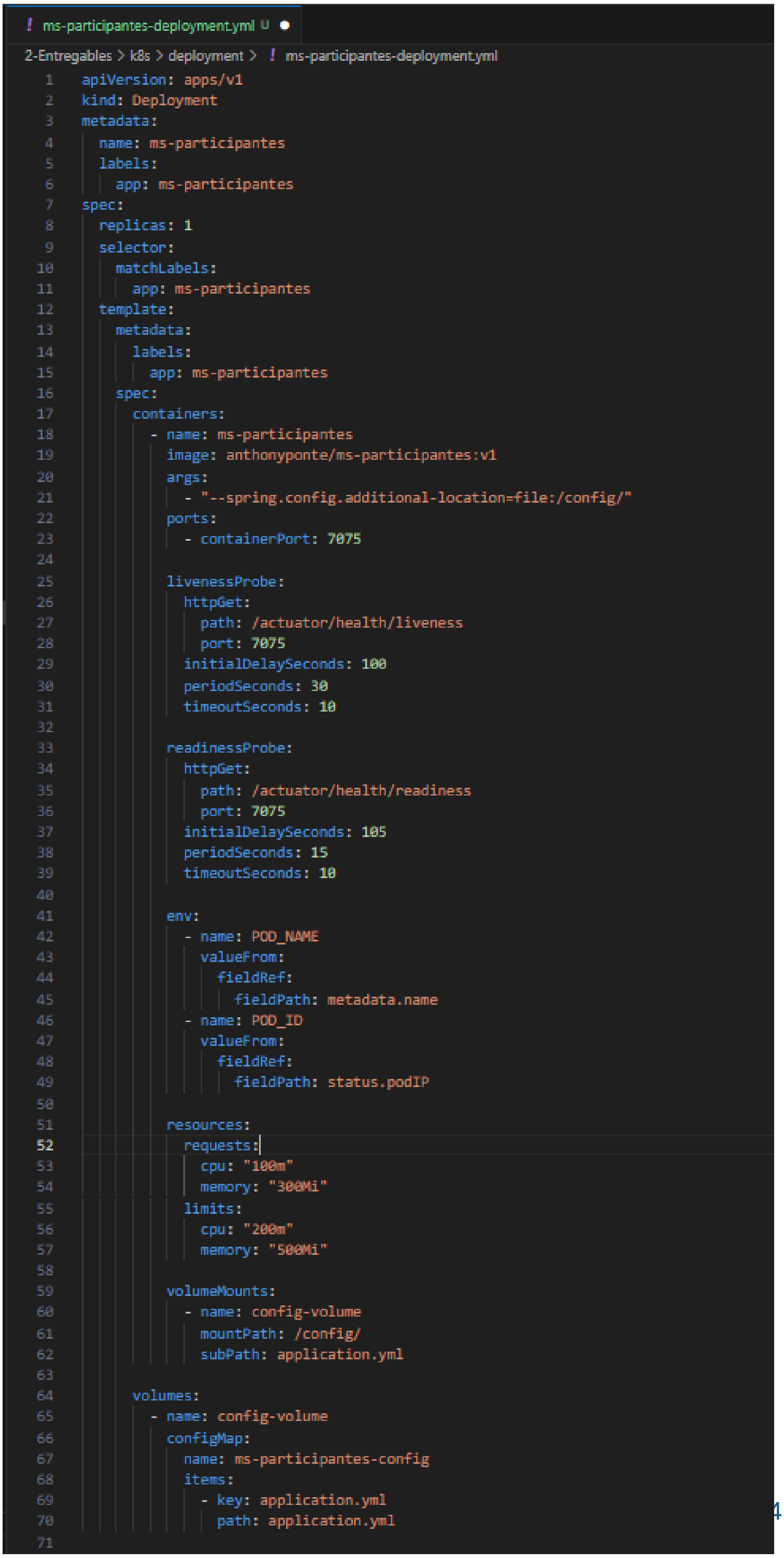

ms-eventos-deployment.yml U ✕
2-Entregables > k8s > deployment > ms-eventos-deployment.yml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ms-eventos-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ms-eventos
  template:
    metadata:
      labels:
        app: ms-eventos
    spec:
      containers:
        - name: ms-eventos-container
          image: anthonyponte/ms-eventos:v1
          args:
            - "--spring.config.additional-location=file:/config/"
          ports:
            - containerPort: 9090
          livenessProbe:
            httpGet:
              path: /actuator/health/liveness
              port: 9090
            initialDelaySeconds: 400
            periodSeconds: 30
            timeoutSeconds: 10
          readinessProbe:
            httpGet:
              path: /actuator/health/readiness
              port: 9090
            initialDelaySeconds: 105
            periodSeconds: 10
            timeoutSeconds: 10
          env:
            - name: DB_USERNAME
              valueFrom:
                secretKeyRef:
                  name: grupo2-secret
                  key: DB_USERNAME
            - name: DB_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: grupo2-secret
                  key: DB_PASSWORD
            - name: POD_NAME
              valueFrom:
                fieldRef:
                  fieldPath: metadata.name
            - name: POD_ID
              valueFrom:
                fieldRef:
                  fieldPath: status.podIP
          resources:
            requests:
              cpu: "100m"
              memory: "400Mi"
            limits:
              cpu: "200m"
              memory: "600Mi"
          volumeMounts:
            - name: ms-eventos-volume
              mountPath: /config/
              subPath: application.yml
      volumes:
        - name: ms-eventos-volume
          configMap:
            name: ms-eventos-configmap
            items:
              - key: application.yml
                path: application.yml
```

**ms-participantes-deployment.yml**

ms-participantes-deployment.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/deployment/ms-participantes-deployment.yml


ms-participantes-deployment.yml U ●
2-Entregables > k8s > deployment > ms-participantes-deployment.yml

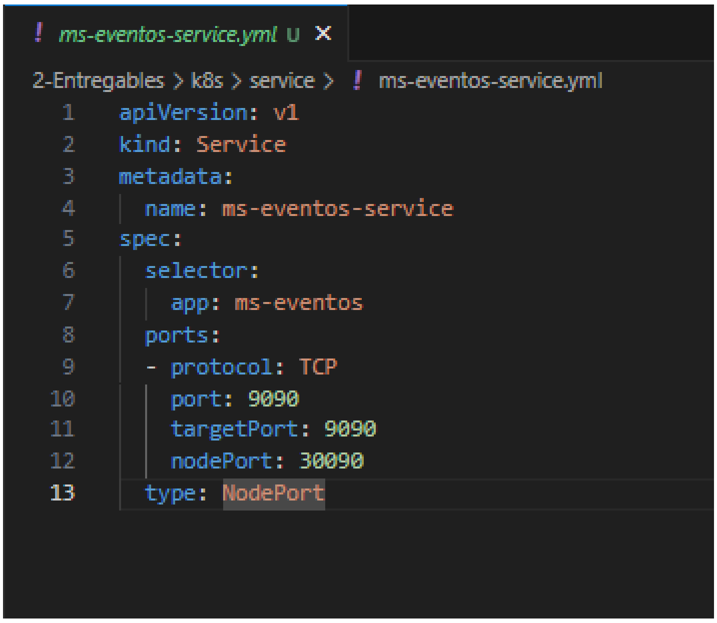
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ms-participantes
  labels:
    app: ms-participantes
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ms-participantes
  template:
    metadata:
      labels:
        app: ms-participantes
    spec:
      containers:
        - name: ms-participantes
          image: anthonyponte/ms-participantes:v1
          args:
            - "--spring.config.additional-location=file:/config/"
          ports:
            - containerPort: 7075

          livenessProbe:
            httpGet:
              path: /actuator/health/liveness
              port: 7075
            initialDelaySeconds: 100
            periodSeconds: 30
            timeoutSeconds: 10

          readinessProbe:
            httpGet:
              path: /actuator/health/readiness
              port: 7075
            initialDelaySeconds: 105
            periodSeconds: 15
            timeoutSeconds: 10

          env:
            - name: POD_NAME
              valueFrom:
                fieldRef:
                  fieldPath: metadata.name
            - name: POD_ID
              valueFrom:
                fieldRef:
                  fieldPath: status.podIP

          resources:
            requests:
              cpu: "100m"
              memory: "300Mi"
            limits:
              cpu: "200m"
              memory: "500Mi"

          volumeMounts:
            - name: config-volume
              mountPath: /config/
              subPath: application.yml

      volumes:
        - name: config-volume
          configMap:
            name: ms-participantes-config
            items:
              - key: application.yml
                path: application.yml

```

- **Services:** Definición de servicios ClusterIP o NodePort para los

**ms-eventos-service.yml**

ms-eventos-service.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/service/ms-eventos-service.yml

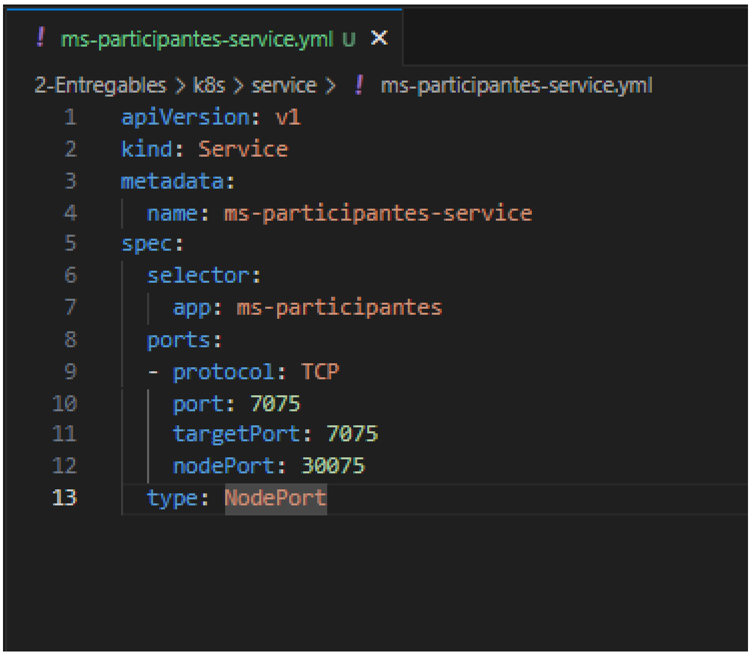

! ms-eventos-service.yml U ×
2-Entregables > k8s > service > ! ms-eventos-service.yml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: ms-eventos-service
spec:
  selector:
    app: ms-eventos
  ports:
  - protocol: TCP
    port: 9090
    targetPort: 9090
    nodePort: 30090
  type: NodePort
```

**ms-participantes-service.yml**

ms-participantes-service.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/service/ms-participantes-service.yml



! ms-participantes-service.yml U ×
2-Entregables > k8s > service > ! ms-participantes-service.yml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: ms-participantes-service
spec:
  selector:
    app: ms-participantes
  ports:
  - protocol: TCP
    port: 7075
    targetPort: 7075
    nodePort: 30075
  type: NodePort
```

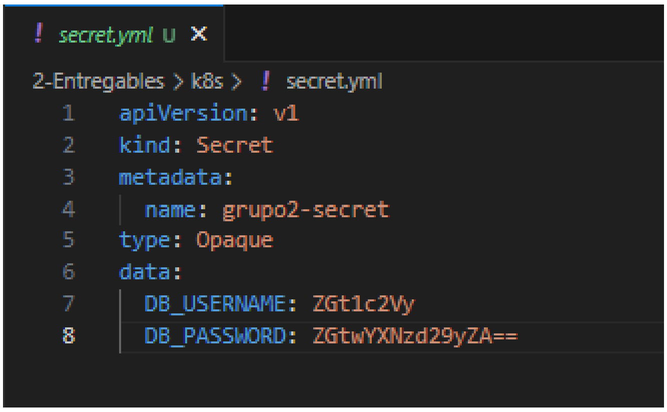
## 4. Microservicios.

- **Ingress:** Configuración de rutas basadas en contexto para exponer las APIs externamente (Opcional, este tema se cubrió en en nivel Básico)
- **ConfigMaps y Secrets:** Manejo de variables de entorno y datos sensibles.

**secret.yml**

secret.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/secret.yml



secret.yml U ✕

2-Entregables > k8s > secret.yml

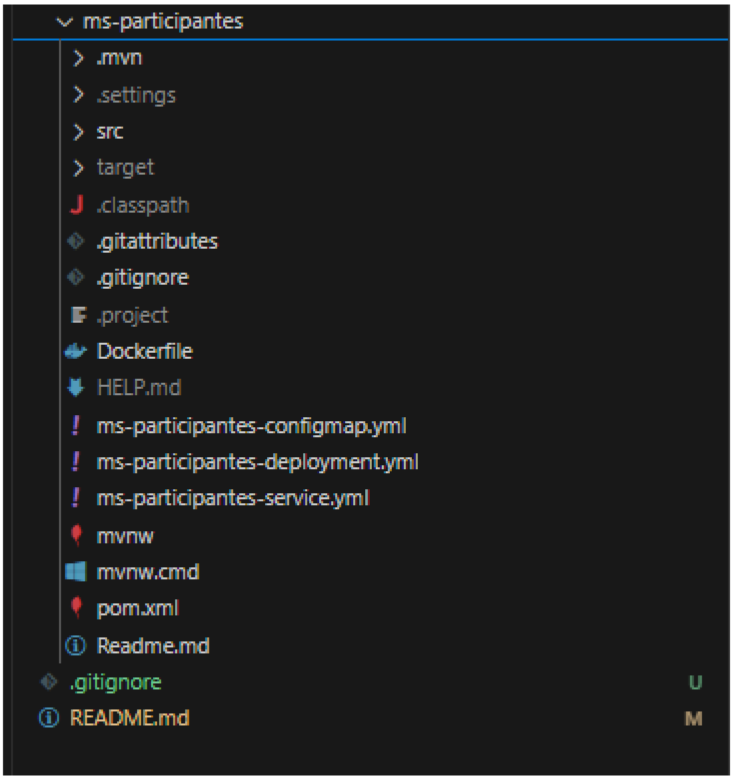
```yaml
apiVersion: v1
kind: Secret
metadata:
  name: grupo2-secret
type: Opaque
data:
  DB_USERNAME: ZGt1c2Vy
  DB_PASSWORD: ZGtwYXNzd29yZA==
```

- **Probes:** Liveness y readiness probes configuradas.

## 4.1. Manifiestos completos:

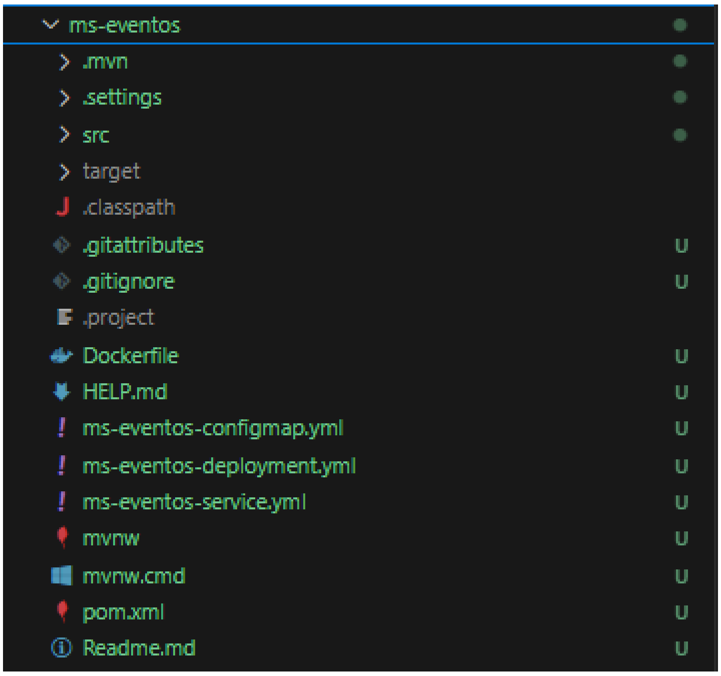
- Conjunto de manifiestos YAML organizados y documentados en carpetas para cada microservicio.

**ms-participantes**



```
ms-participantes
  > .mvn
  > .settings
  > src
  > target
  .classpath
  .gitattributes
  .gitignore
  .project
  Dockerfile
  HELP.md
  ms-participantes-configmap.yml
  ms-participantes-deployment.yml
  ms-participantes-service.yml
  mvnw
  mvnw.cmd
  pom.xml
  Readme.md
.gitignore                                   U
README.md                                    M
```

**ms-eventos**



```
⌄ ms-eventos
  > .mvn
  > .settings
  > src
  > target
    .classpath
    .gitattributes                  U
    .gitignore                      U
    .project
    Dockerfile                      U
    HELP.md                         U
    ms-eventos-configmap.yml        U
    ms-eventos-deployment.yml       U
    ms-eventos-service.yml          U
    mvnw                            U
    mvnw.cmd                        U
    pom.xml                         U
    Readme.md                       U
```

## 5. Bases de Datos

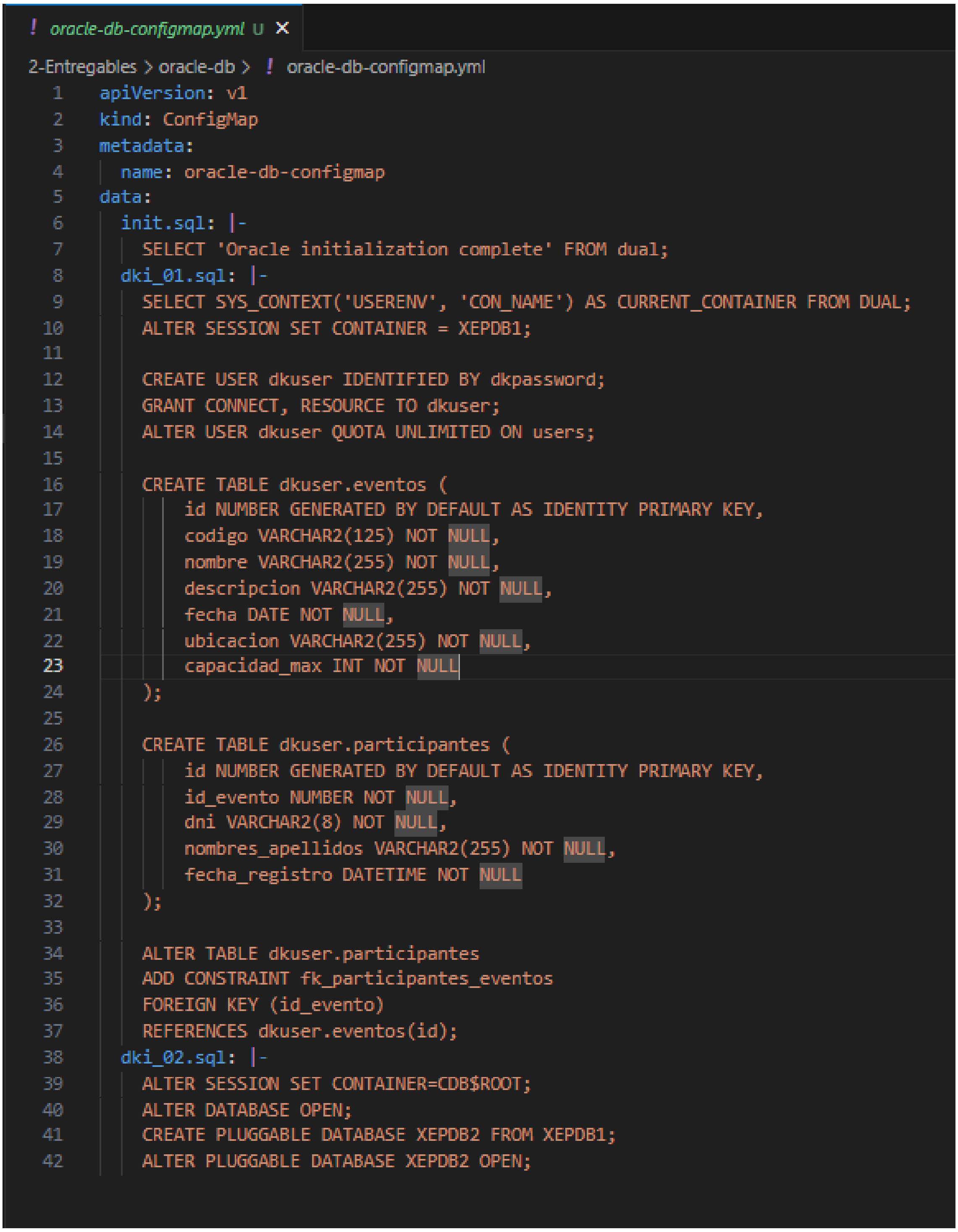
### 5.1. Scripts de Inicialización:

- Scripts SQL para inicializar las bases de datos con tablas necesarias y datos de prueba.

**oracle-db-configmap.yml**

oracle-db-configmap.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/oracle-db-configmap.yml



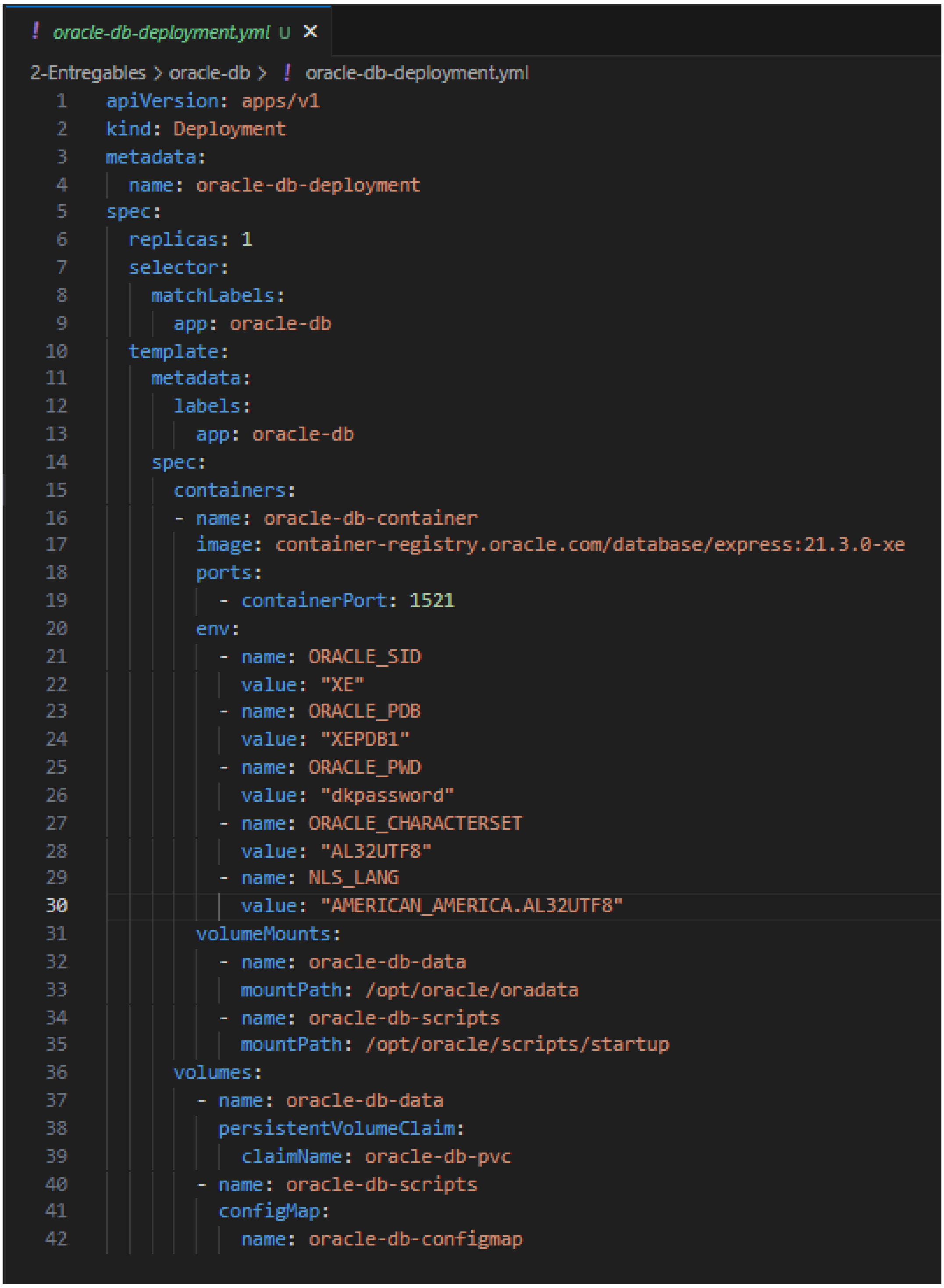
! oracle-db-configmap.yml U ×
2-Entregables > oracle-db > ! oracle-db-configmap.yml

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: oracle-db-configmap
data:
  init.sql: |-
    SELECT 'Oracle initialization complete' FROM dual;
  dki_01.sql: |-
    SELECT SYS_CONTEXT('USERENV', 'CON_NAME') AS CURRENT_CONTAINER FROM DUAL;
    ALTER SESSION SET CONTAINER = XEPDB1;

    CREATE USER dkuser IDENTIFIED BY dkpassword;
    GRANT CONNECT, RESOURCE TO dkuser;
    ALTER USER dkuser QUOTA UNLIMITED ON users;

    CREATE TABLE dkuser.eventos (
        id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
        codigo VARCHAR2(125) NOT NULL,
        nombre VARCHAR2(255) NOT NULL,
        descripcion VARCHAR2(255) NOT NULL,
        fecha DATE NOT NULL,
        ubicacion VARCHAR2(255) NOT NULL,
        capacidad_max INT NOT NULL
    );

    CREATE TABLE dkuser.participantes (
        id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
        id_evento NUMBER NOT NULL,
        dni VARCHAR2(8) NOT NULL,
        nombres_apellidos VARCHAR2(255) NOT NULL,
        fecha_registro DATETIME NOT NULL
    );

    ALTER TABLE dkuser.participantes
    ADD CONSTRAINT fk_participantes_eventos
    FOREIGN KEY (id_evento)
    REFERENCES dkuser.eventos(id);
  dki_02.sql: |-
    ALTER SESSION SET CONTAINER=CDB$ROOT;
    ALTER DATABASE OPEN;
    CREATE PLUGGABLE DATABASE XEPDB2 FROM XEPDB1;
    ALTER PLUGGABLE DATABASE XEPDB2 OPEN;
```

**oracle-db-deployment.yml**

oracle-db-deployment.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/oracle-db-deployment.yml

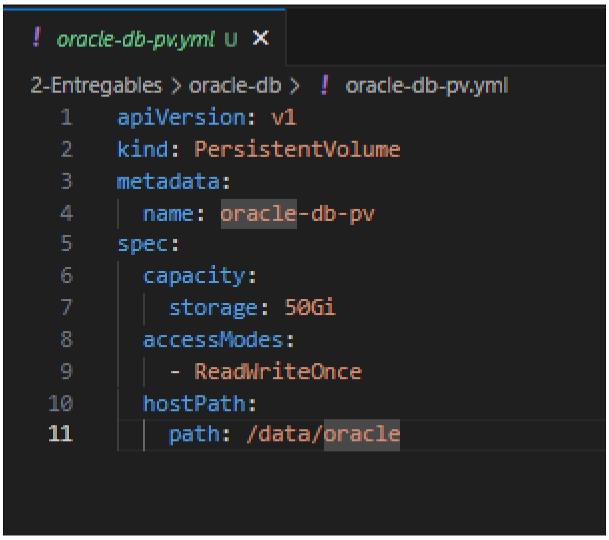

! oracle-db-deployment.yml U ×
2-Entregables > oracle-db > ! oracle-db-deployment.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: oracle-db-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: oracle-db
  template:
    metadata:
      labels:
        app: oracle-db
    spec:
      containers:
      - name: oracle-db-container
        image: container-registry.oracle.com/database/express:21.3.0-xe
        ports:
          - containerPort: 1521
        env:
          - name: ORACLE_SID
            value: "XE"
          - name: ORACLE_PDB
            value: "XEPDB1"
          - name: ORACLE_PWD
            value: "dkpassword"
          - name: ORACLE_CHARACTERSET
            value: "AL32UTF8"
          - name: NLS_LANG
            value: "AMERICAN_AMERICA.AL32UTF8"
        volumeMounts:
          - name: oracle-db-data
            mountPath: /opt/oracle/oradata
          - name: oracle-db-scripts
            mountPath: /opt/oracle/scripts/startup
      volumes:
        - name: oracle-db-data
          persistentVolumeClaim:
            claimName: oracle-db-pvc
        - name: oracle-db-scripts
          configMap:
            name: oracle-db-configmap
```

**oracle-db-pv.yml**

oracle-db-pv.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/oracle-db-pv.yml

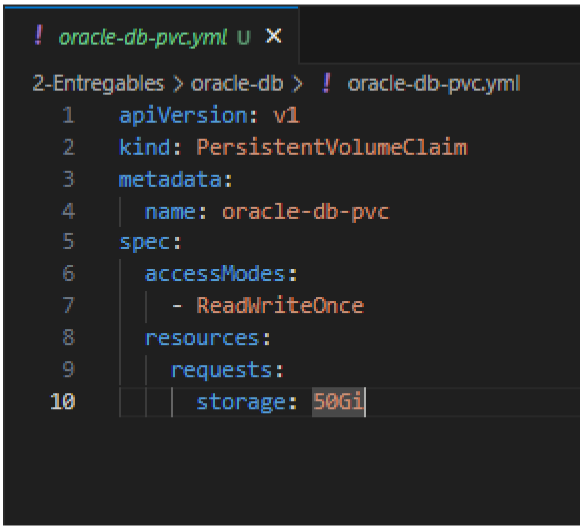


```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: oracle-db-pv
spec:
  capacity:
    storage: 50Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: /data/oracle
```

**oracle-db-pvc.yml**

oracle-db-pvc.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/oracle-db-pvc.yml

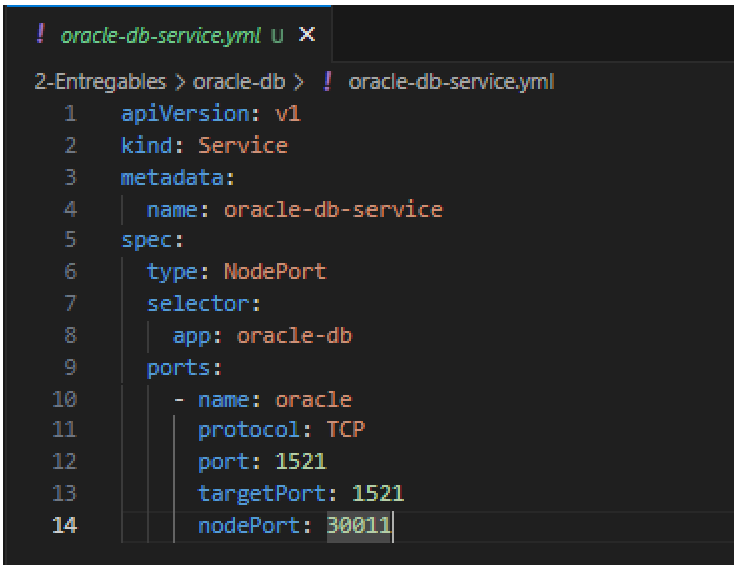


```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: oracle-db-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 50Gi
```

**oracle-db-service.yml**

oracle-db-service.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/oracle-db-service.yml



! oracle-db-service.yml U ×

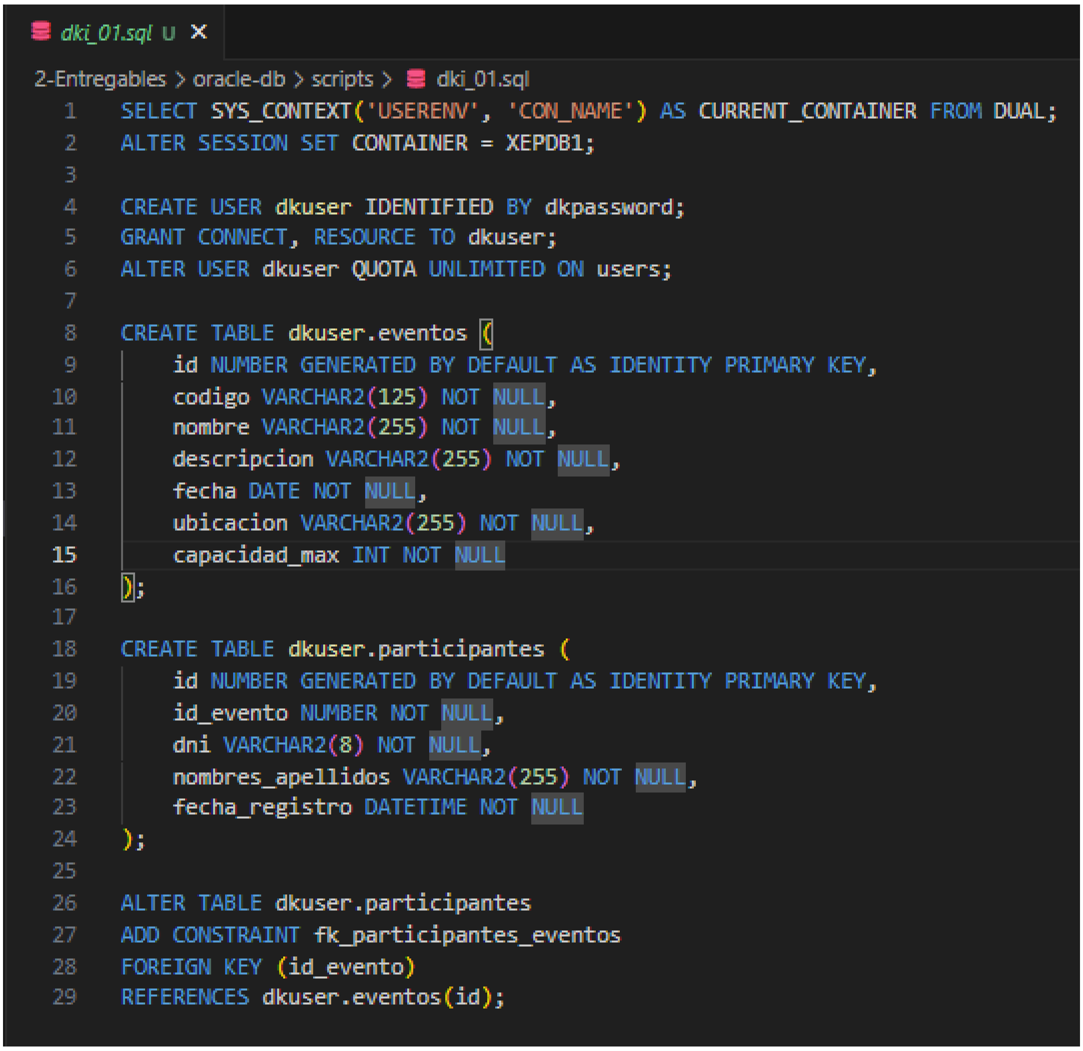
2-Entregables > oracle-db > ! oracle-db-service.yml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: oracle-db-service
spec:
  type: NodePort
  selector:
    app: oracle-db
  ports:
    - name: oracle
      protocol: TCP
      port: 1521
      targetPort: 1521
      nodePort: 30011
```

**dki_01.sql**

dki_01.sql

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/scripts/dki_01.sql

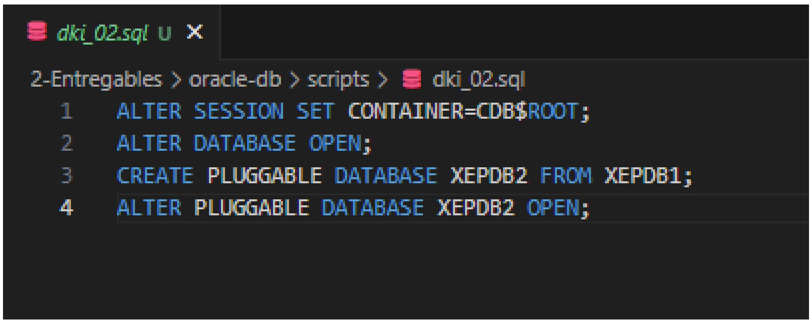


```sql
SELECT SYS_CONTEXT('USERENV', 'CON_NAME') AS CURRENT_CONTAINER FROM DUAL;
ALTER SESSION SET CONTAINER = XEPDB1;

CREATE USER dkuser IDENTIFIED BY dkpassword;
GRANT CONNECT, RESOURCE TO dkuser;
ALTER USER dkuser QUOTA UNLIMITED ON users;

CREATE TABLE dkuser.eventos (
    id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    codigo VARCHAR2(125) NOT NULL,
    nombre VARCHAR2(255) NOT NULL,
    descripcion VARCHAR2(255) NOT NULL,
    fecha DATE NOT NULL,
    ubicacion VARCHAR2(255) NOT NULL,
    capacidad_max INT NOT NULL
);

CREATE TABLE dkuser.participantes (
    id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    id_evento NUMBER NOT NULL,
    dni VARCHAR2(8) NOT NULL,
    nombres_apellidos VARCHAR2(255) NOT NULL,
    fecha_registro DATETIME NOT NULL
);

ALTER TABLE dkuser.participantes
ADD CONSTRAINT fk_participantes_eventos
FOREIGN KEY (id_evento)
REFERENCES dkuser.eventos(id);
```

**dki_02.sql**

dki_02.sql

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/scripts/dki_02.sql



```sql
ALTER SESSION SET CONTAINER=CDB$ROOT;
ALTER DATABASE OPEN;
CREATE PLUGGABLE DATABASE XEPDB2 FROM XEPDB1;
ALTER PLUGGABLE DATABASE XEPDB2 OPEN;
```

### 5.2. Volúmenes:

- Configuración de volúmenes en Kubernetes para garantizar la persistencia de los datos.

## 6. Documentación

### 6.1. Manual de Despliegue:

- Pasos detallados para:
  - Contenerizar y construir las imágenes:

    *Verificar la existencia de la imagen dki-volume:*

    *docker volume ls*

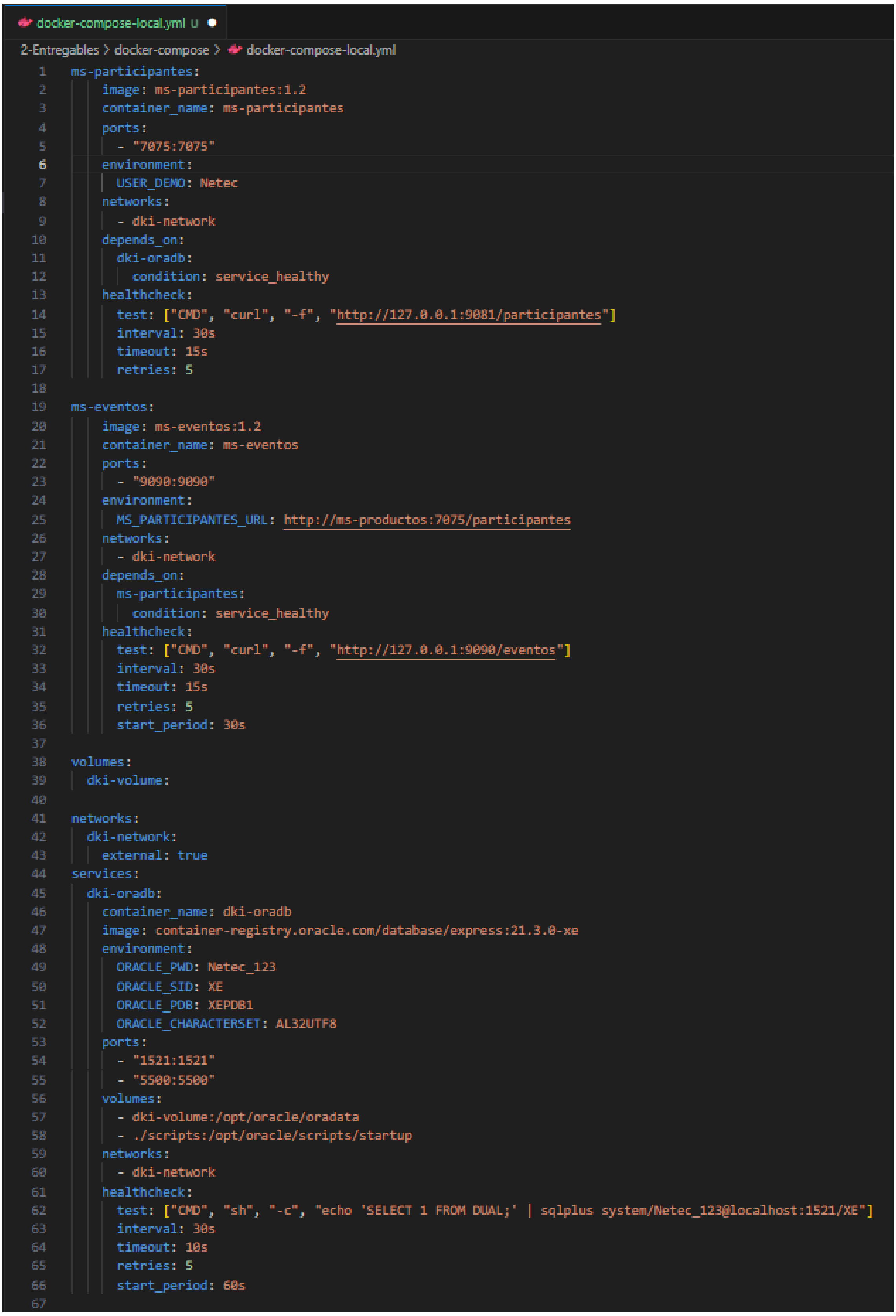
    *Creación de la imagen dki-volume:*

    *docker volume create dki-volume*

  - Desplegar en Kubernetes utilizando los manifiestos YAML.
  - Configurar Docker Compose para desarrollo local.

docker-compose-local.yml

[https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/docker-compose/docker-compose-local.yml](https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/docker-compose/docker-compose-local.yml)



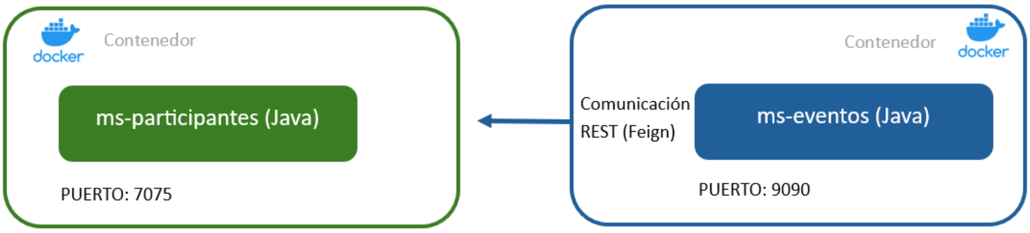
docker-compose-local.yml U ●

2-Entregables > docker-compose > docker-compose-local.yml

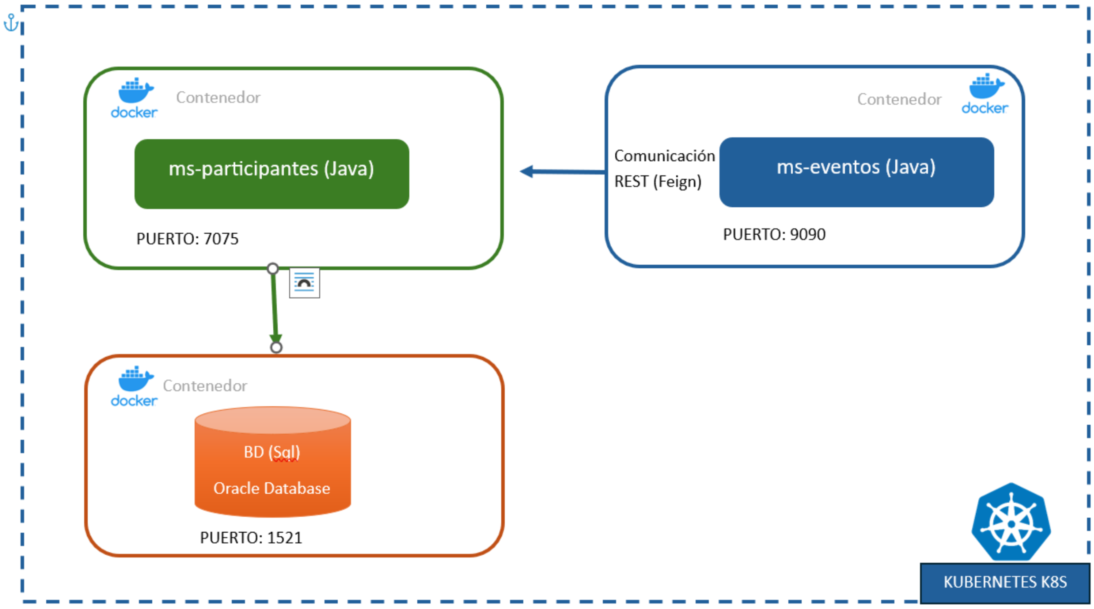
```yaml
ms-participantes:
    image: ms-participantes:1.2
    container_name: ms-participantes
    ports:
      - "7075:7075"
    environment:
      USER_DEMO: Netec
    networks:
      - dki-network
    depends_on:
      dki-oradb:
        condition: service_healthy
    healthcheck:
      test: ["CMD", "curl", "-f", "http://127.0.0.1:9081/participantes"]
      interval: 30s
      timeout: 15s
      retries: 5

ms-eventos:
    image: ms-eventos:1.2
    container_name: ms-eventos
    ports:
      - "9090:9090"
    environment:
      MS_PARTICIPANTES_URL: http://ms-productos:7075/participantes
    networks:
      - dki-network
    depends_on:
      ms-participantes:
        condition: service_healthy
    healthcheck:
      test: ["CMD", "curl", "-f", "http://127.0.0.1:9090/eventos"]
      interval: 30s
      timeout: 15s
      retries: 5
      start_period: 30s

volumes:
  dki-volume:

networks:
  dki-network:
    external: true
services:
  dki-oradb:
    container_name: dki-oradb
    image: container-registry.oracle.com/database/express:21.3.0-xe
    environment:
      ORACLE_PWD: Netec_123
      ORACLE_SID: XE
      ORACLE_PDB: XEPDB1
      ORACLE_CHARACTERSET: AL32UTF8
    ports:
      - "1521:1521"
      - "5500:5500"
    volumes:
      - dki-volume:/opt/oracle/oradata
      - ./scripts:/opt/oracle/scripts/startup
    networks:
      - dki-network
    healthcheck:
      test: ["CMD", "sh", "-c", "echo 'SELECT 1 FROM DUAL;' | sqlplus system/Netec_123@localhost:1521/XE"]
      interval: 30s
      timeout: 10s
      retries: 5
      start_period: 60s
```

### 6.2. Diagrama de Arquitectura:

- Representación gráfica de la solución, incluyendo:
  - Estructura de microservicios:



  - Comunicación entre ellos.
  - Despliegue en Kubernetes:

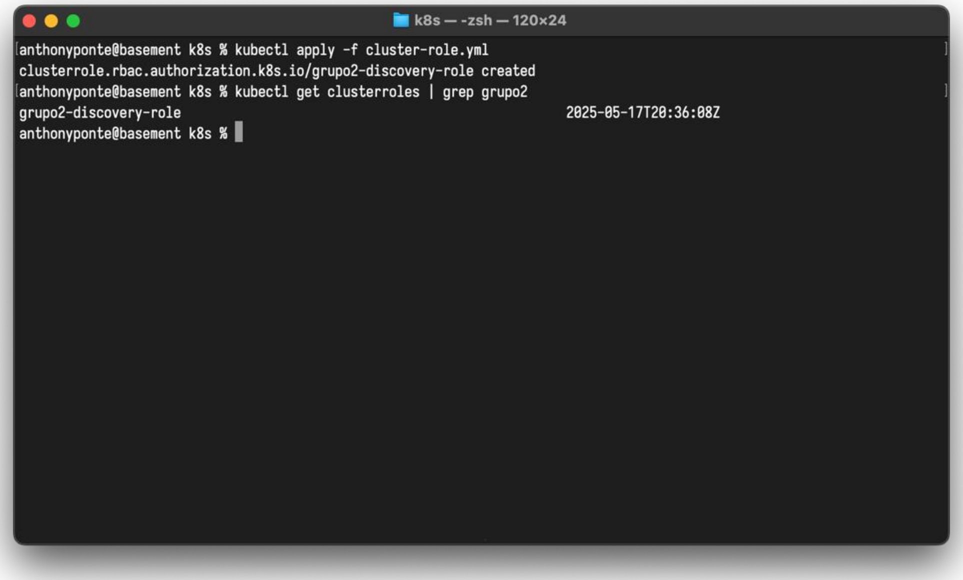


## 7. Resultados

### 7.1. Validación del Despliegue:

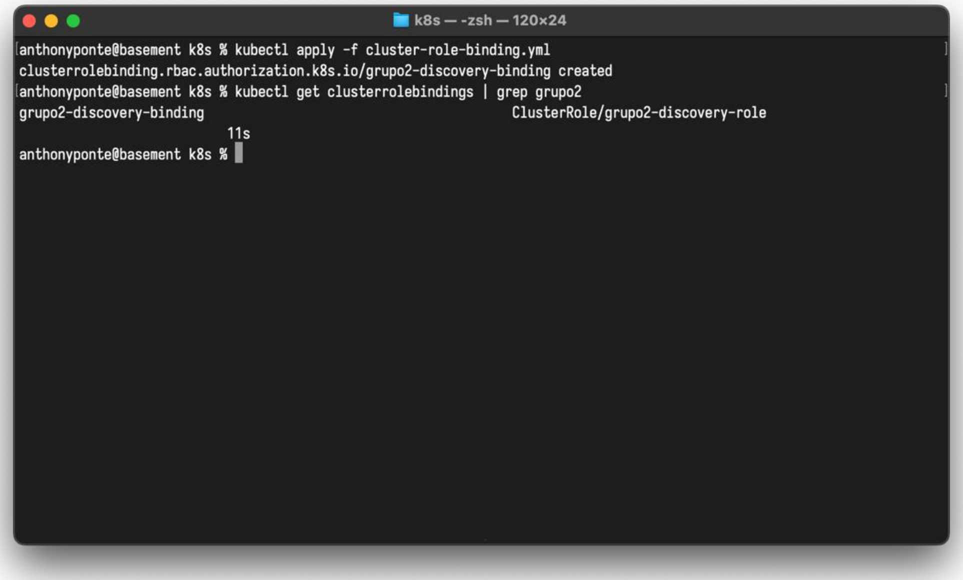
- Evidencia del despliegue exitoso en Kubernetes (por ejemplo, capturas de pantalla o logs que muestren los pods corriendo).

```
kubectl apply -f cluster-role.yml

kubectl get clusterroles | grep grupo2
```



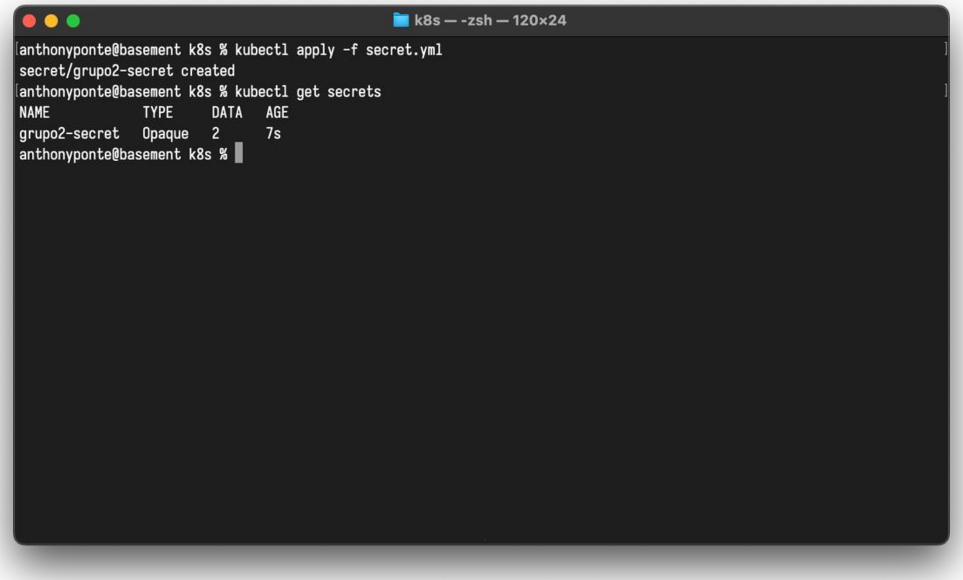
```
anthonyponte@basement k8s % kubectl apply -f cluster-role.yml
clusterrole.rbac.authorization.k8s.io/grupo2-discovery-role created
anthonyponte@basement k8s % kubectl get clusterroles | grep grupo2
grupo2-discovery-role                                                  2025-05-17T20:36:08Z
anthonyponte@basement k8s % 
```

```
kubectl apply -f cluster-role-binding.yml

kubectl get clusterrolebindings | grep grupo2
```



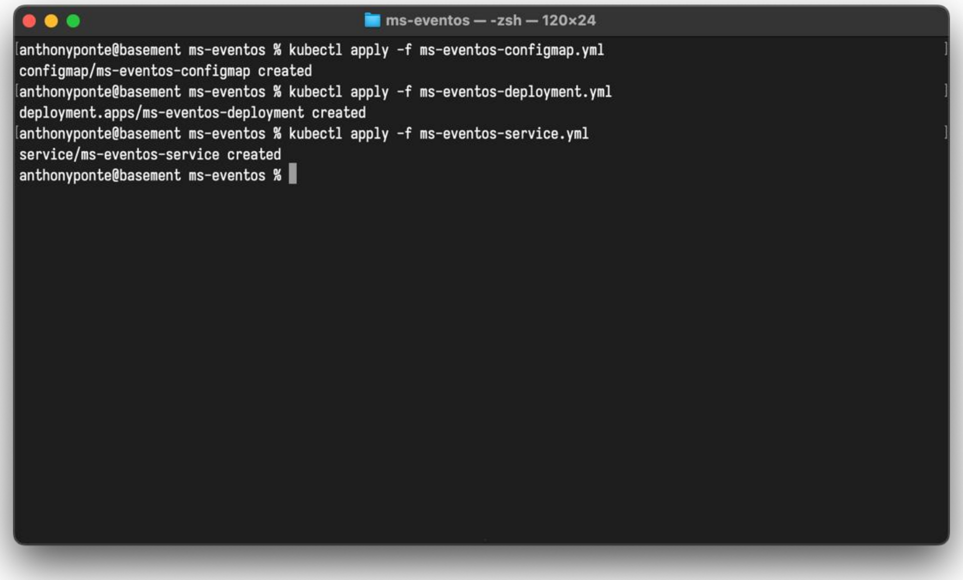
```
anthonyponte@basement k8s % kubectl apply -f cluster-role-binding.yml
clusterrolebinding.rbac.authorization.k8s.io/grupo2-discovery-binding created
anthonyponte@basement k8s % kubectl get clusterrolebindings | grep grupo2
grupo2-discovery-binding                                        ClusterRole/grupo2-discovery-role
                           11s
anthonyponte@basement k8s % 
```

```
kubectl apply -f secret.yml

kubectl get secrets
```



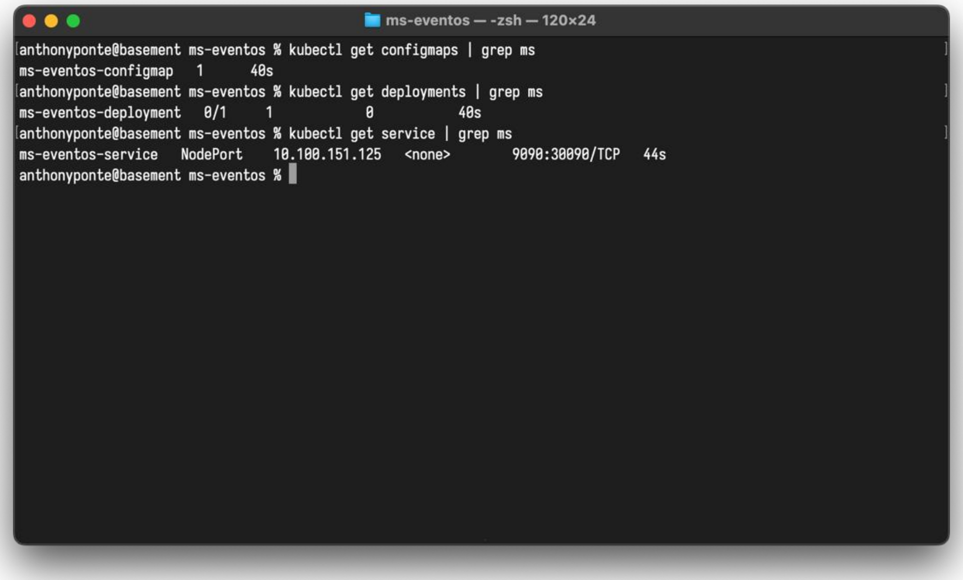
```
anthonyponte@basement k8s % kubectl apply -f secret.yml
secret/grupo2-secret created
anthonyponte@basement k8s % kubectl get secrets
NAME            TYPE     DATA   AGE
grupo2-secret   Opaque   2      7s
anthonyponte@basement k8s %
```

```
kubectl apply -f ms-eventos-configmap.yml

kubectl apply -f ms-eventos-deployment.yml

kubectl apply -f ms-eventos-service.yml
```



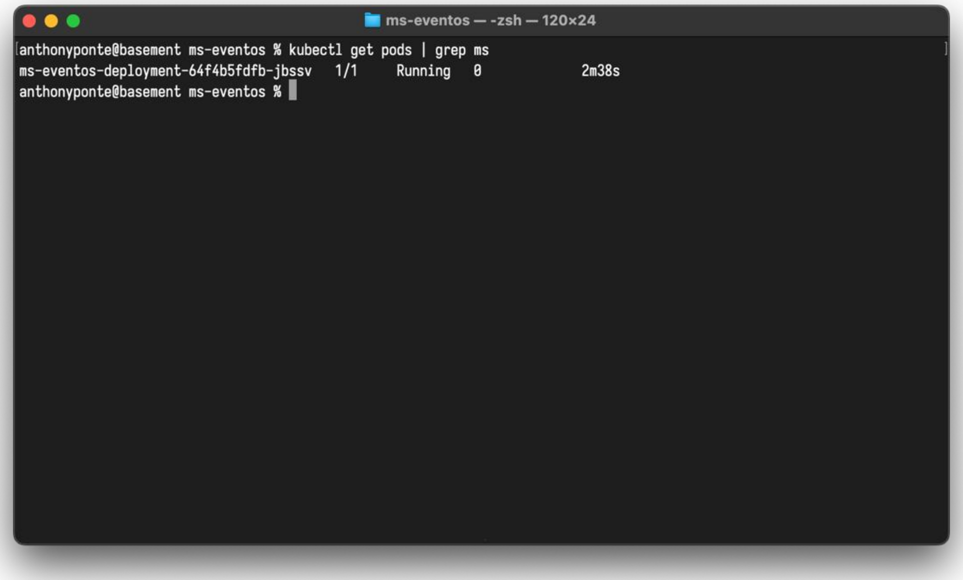
```
anthonyponte@basement ms-eventos % kubectl apply -f ms-eventos-configmap.yml
configmap/ms-eventos-configmap created
anthonyponte@basement ms-eventos % kubectl apply -f ms-eventos-deployment.yml
deployment.apps/ms-eventos-deployment created
anthonyponte@basement ms-eventos % kubectl apply -f ms-eventos-service.yml
service/ms-eventos-service created
anthonyponte@basement ms-eventos %
```

```
kubectl get configmaps | grep ms

kubectl get deployments | grep ms

kubectl get service | grep ms
```

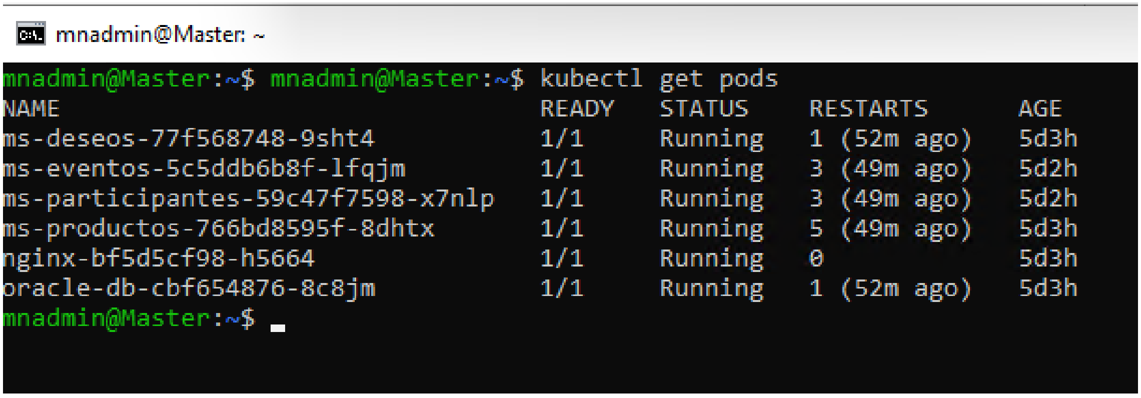


```
anthonyponte@basement ms-eventos % kubectl get configmaps | grep ms
ms-eventos-configmap   1      40s
anthonyponte@basement ms-eventos % kubectl get deployments | grep ms
ms-eventos-deployment   0/1     1            0           40s
anthonyponte@basement ms-eventos % kubectl get service | grep ms
ms-eventos-service   NodePort    10.100.151.125   <none>        9090:30090/TCP   44s
anthonyponte@basement ms-eventos % 
```

```
kubectl get pods | grep ms
```

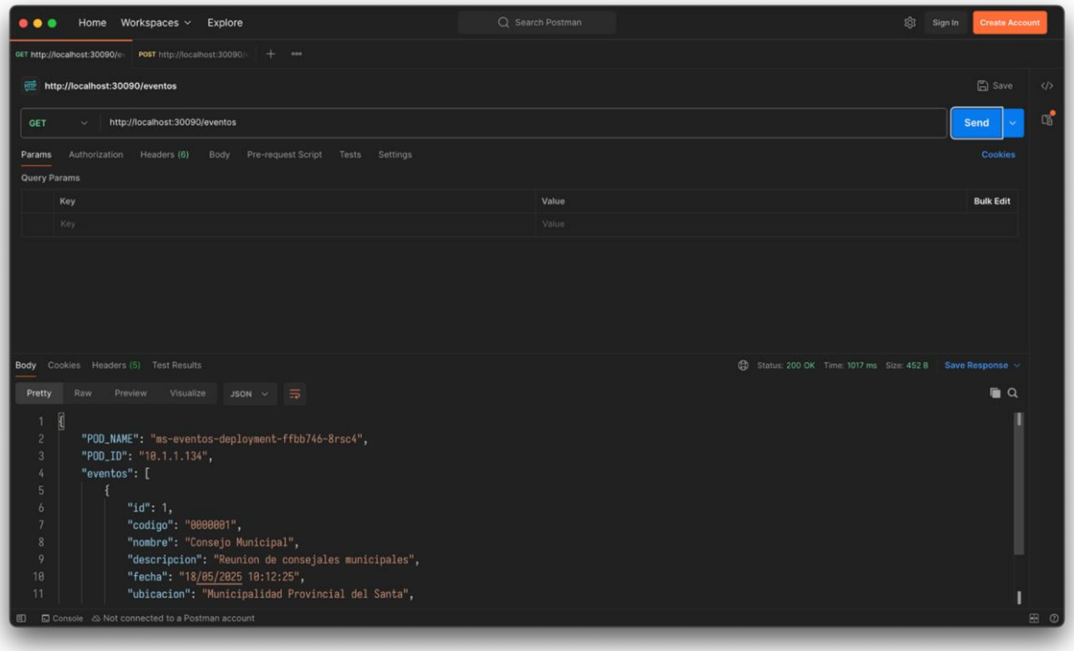


```
anthonyponte@basement ms-eventos % kubectl get pods | grep ms
ms-eventos-deployment-64f4b5fdfb-jbssv   1/1     Running   0             2m38s
anthonyponte@basement ms-eventos % 
```



```
mnadmin@Master:~$ mnadmin@Master:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS      AGE
ms-deseos-77f568748-9sht4           1/1     Running   1 (52m ago)   5d3h
ms-eventos-5c5ddb6b8f-lfqjm         1/1     Running   3 (49m ago)   5d2h
ms-participantes-59c47f7598-x7nlp   1/1     Running   3 (49m ago)   5d2h
ms-productos-766bd8595f-8dhtx       1/1     Running   5 (49m ago)   5d3h
nginx-bf5d5cf98-h5664               1/1     Running   0             5d3h
oracle-db-cbf654876-8c8jm           1/1     Running   1 (52m ago)   5d3h
mnadmin@Master:~$ 
```

- Pruebas realizadas en los servicios mediante Postman o CURL.

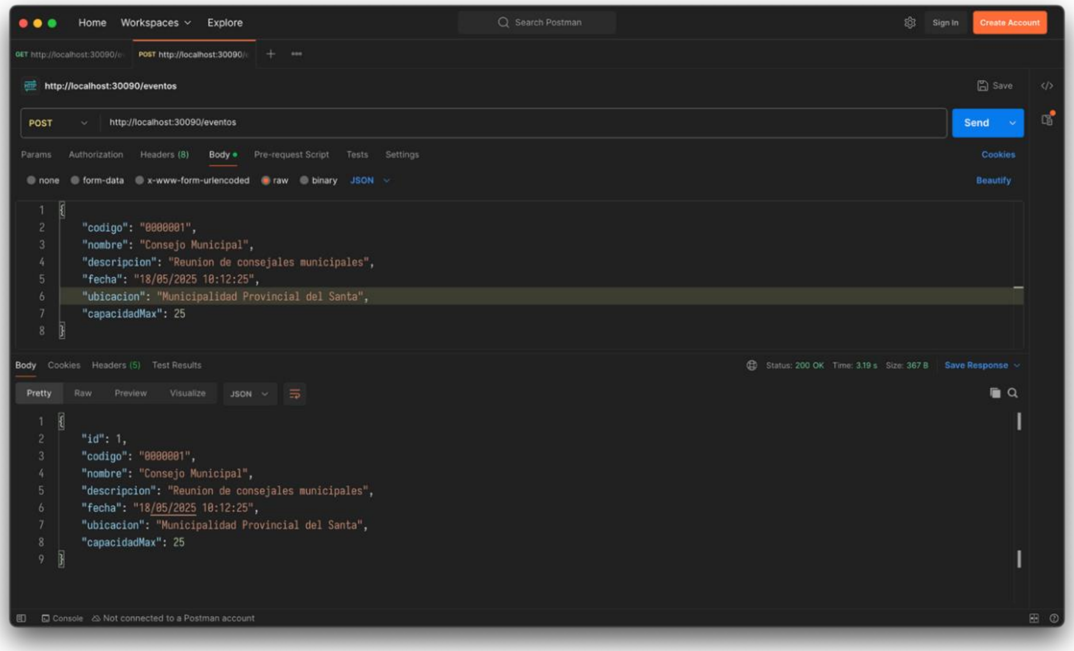


http://localhost:30090/eventos

GET http://localhost:30090/eventos

Status: 200 OK Time: 1017 ms Size: 452 B

```
{
    "POD_NAME": "ms-eventos-deployment-ffbb746-8rsc4",
    "POD_ID": "10.1.1.134",
    "eventos": [
        {
            "id": 1,
            "codigo": "0000001",
            "nombre": "Consejo Municipal",
            "descripcion": "Reunion de consejales municipales",
            "fecha": "18/05/2025 10:12:25",
            "ubicacion": "Municipalidad Provincial del Santa",
```



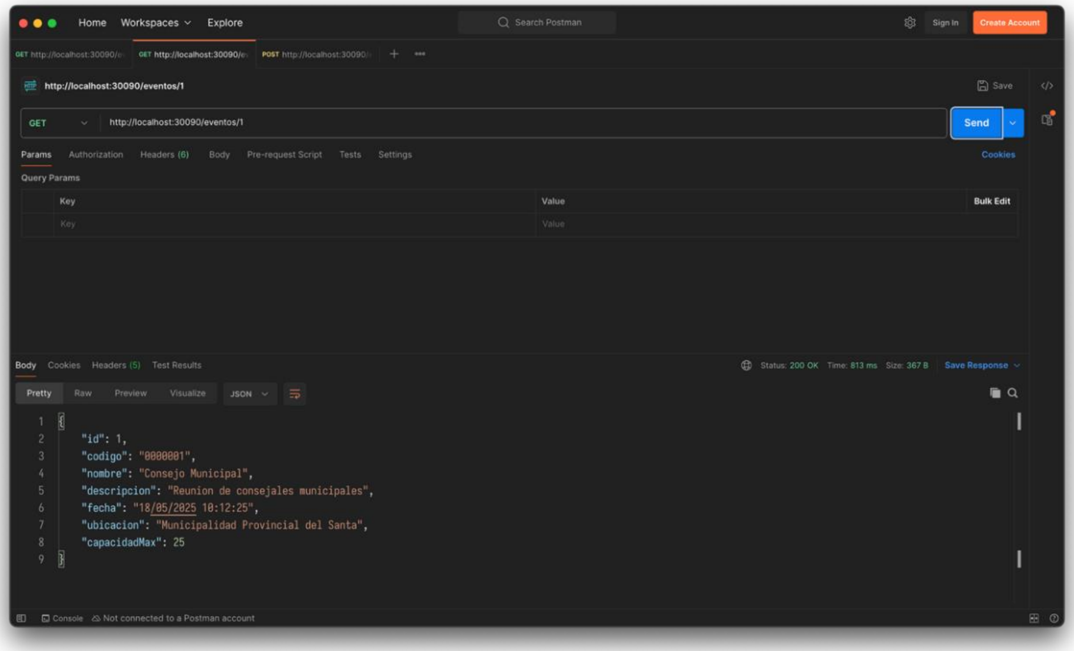
http://localhost:30090/eventos

POST http://localhost:30090/eventos

```
{
    "codigo": "0000001",
    "nombre": "Consejo Municipal",
    "descripcion": "Reunion de consejales municipales",
    "fecha": "18/05/2025 10:12:25",
    "ubicacion": "Municipalidad Provincial del Santa",
    "capacidadMax": 25
}
```

Status: 200 OK Time: 3.19 s Size: 367 B

```
{
    "id": 1,
    "codigo": "0000001",
    "nombre": "Consejo Municipal",
    "descripcion": "Reunion de consejales municipales",
    "fecha": "18/05/2025 10:12:25",
    "ubicacion": "Municipalidad Provincial del Santa",
    "capacidadMax": 25
}
```

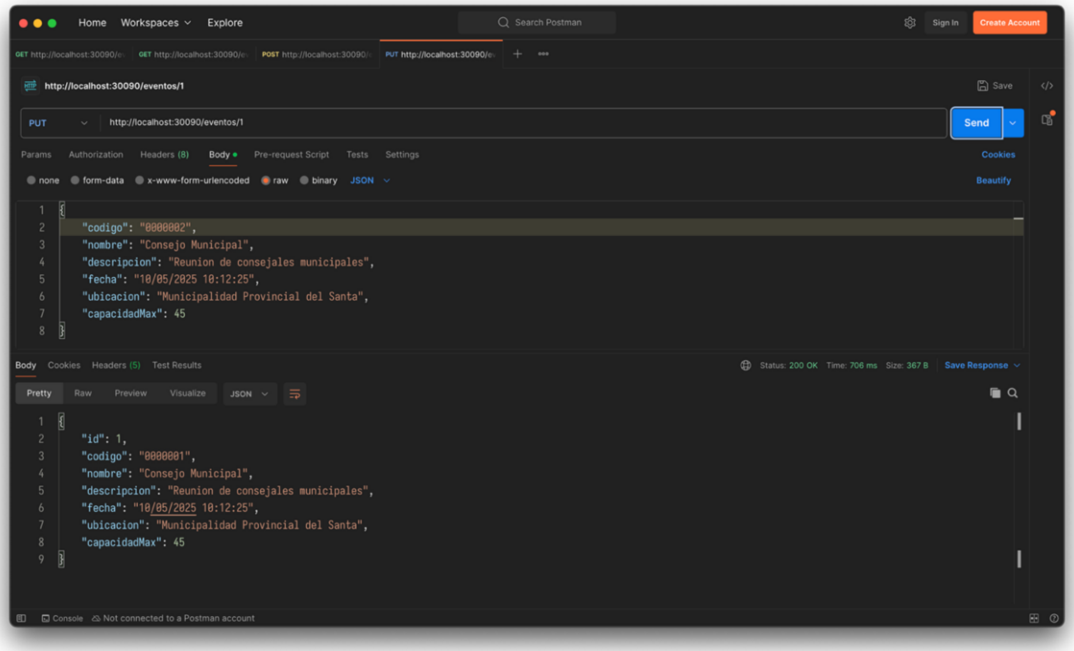


http://localhost:30090/eventos/1

GET http://localhost:30090/eventos/1

Status: 200 OK Time: 813 ms Size: 367 B

```
{
    "id": 1,
    "codigo": "0000001",
    "nombre": "Consejo Municipal",
    "descripcion": "Reunion de consejales municipales",
    "fecha": "18/05/2025 10:12:25",
    "ubicacion": "Municipalidad Provincial del Santa",
    "capacidadMax": 25
}
```



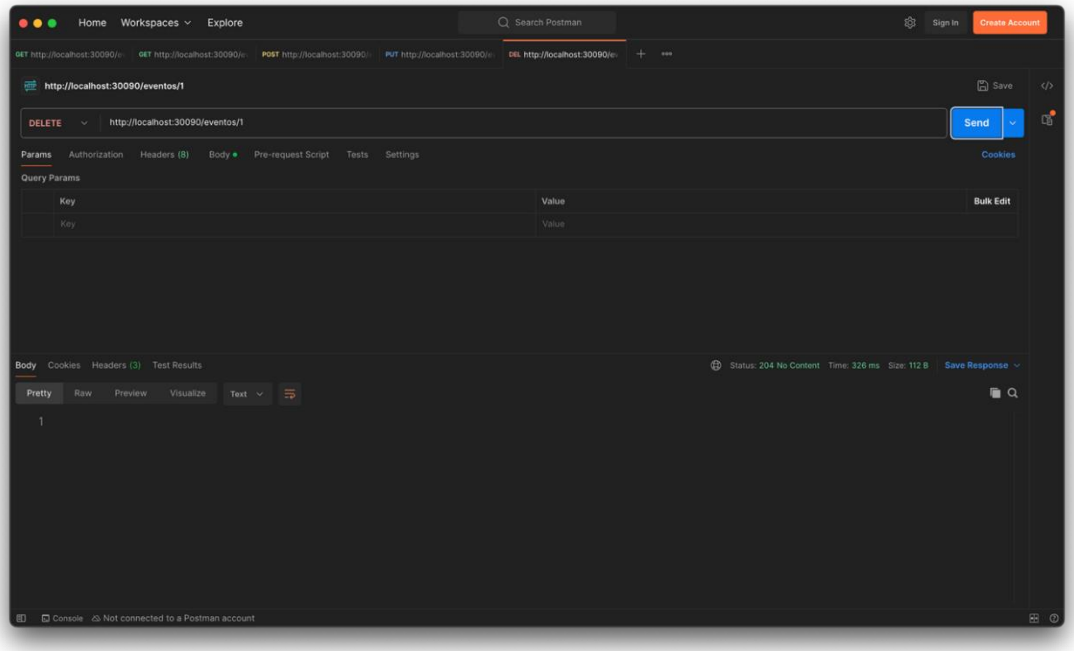
http://localhost:30090/eventos/1

PUT http://localhost:30090/eventos/1

```
{
  "codigo": "0000002",
  "nombre": "Consejo Municipal",
  "descripcion": "Reunion de consejales municipales",
  "fecha": "10/05/2025 10:12:25",
  "ubicacion": "Municipalidad Provincial del Santa",
  "capacidadMax": 45
}
```

Status: 200 OK Time: 706 ms Size: 367 B

```
{
  "id": 1,
  "codigo": "0000001",
  "nombre": "Consejo Municipal",
  "descripcion": "Reunion de consejales municipales",
  "fecha": "10/05/2025 10:12:25",
  "ubicacion": "Municipalidad Provincial del Santa",
  "capacidadMax": 45
}
```



http://localhost:30090/eventos/1

DELETE http://localhost:30090/eventos/1

Query Params

| Key | Value |
|---|---|
| Key | Value |

Status: 204 No Content Time: 326 ms Size: 112 B

## 7.2. Registro de Imágenes:

- URL de las imágenes Docker publicadas (Docker Hub).

**ms-eventos:**
https://hub.docker.com/repository/docker/jonathan0284/ms-eventos/general

**ms-participantes:**
https://hub.docker.com/repository/docker/jonathan0284/ms-participantes/general