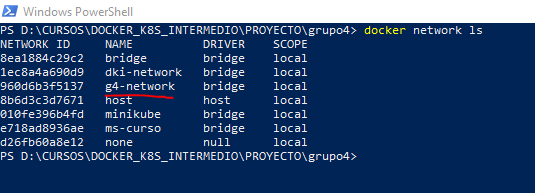
Pasos a seguir:

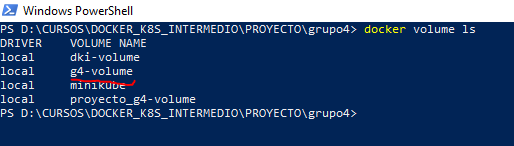
1. Crear network

docker network create --driver bridge g4-network



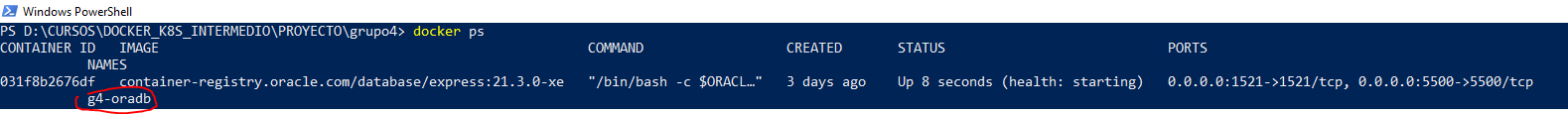
1. Crear volumen

docker volume create g4-volume



1. Crear el contenedor Oracle

docker run -d --name g4-oradb -e ORACLE\_PWD=Netec\_123 --network g4-network --volume g4-volume:/opt/oracle/oradata -p 1521:1521 -p 5500:5500 container-registry.oracle.com/database/express:21.3.0-xe



**Datos de la conexión**

Usuario: sys

Rol: sysdba

Contraseña: Netec\_123

Nombre del Host: localhost

Puerto: 1521

SID: xe

1. Crear usuario de BD

Ejecutar lo siguiente:

**ALTER** **SESSION** **SET** CONTAINER = XEPDB1;

**CREATE** **USER** g4user **IDENTIFIED** **BY** g4password

**DEFAULT** **TABLESPACE** users

**TEMPORARY** **TABLESPACE** temp;

**GRANT** **CONNECT**, RESOURCE **TO** g4user;

**ALTER** **USER** g4user QUOTA UNLIMITED **ON** users;

**ALTER** **USER** g4user QUOTA UNLIMITED **ON** users;

1. Creación de tablas:

Este punto no es estrictamente necesario, ya que la creación de objetos se realiza al levantar los microservicio.

-- G4USER.EVENTO definition

**CREATE** **TABLE** "G4USER"."EVENTO"

( "ID" **NUMBER**(19,0) GENERATED **BY** **DEFAULT** **AS** **IDENTITY** **MINVALUE** 1 **MAXVALUE** 9999999999999999999999999999 **INCREMENT** **BY** 1 **START** **WITH** 1 CACHE 20 NOORDER NOCYCLE NOKEEP NOSCALE **NOT** **NULL** ENABLE,

**"CAPACIDAD"** **NUMBER**(10,0) **NOT** **NULL** ENABLE,

**"DESCRIPCION"** **VARCHAR2**(255 **CHAR**),

**"FECHA"** **TIMESTAMP** (6) **NOT** **NULL** ENABLE,

**"NOMBRE"** **VARCHAR2**(255 **CHAR**) **NOT** **NULL** ENABLE,

**"UBICACION"** **VARCHAR2**(255 **CHAR**),

**PRIMARY** **KEY** (**"ID"**)

**USING** **INDEX** **PCTFREE** 10 INITRANS 2 MAXTRANS 255 **COMPUTE** **STATISTICS**

STORAGE(**INITIAL** 65536 **NEXT** 1048576 MINEXTENTS 1 **MAXEXTENTS** 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL **DEFAULT** FLASH\_CACHE **DEFAULT** CELL\_FLASH\_CACHE **DEFAULT**)

**TABLESPACE** **"USERS"** ENABLE,

**CONSTRAINT** **"UK3GNB3B0HCJYISTT3JEW62CX7X"** **UNIQUE** (**"NOMBRE"**)

**USING** **INDEX** **PCTFREE** 10 INITRANS 2 MAXTRANS 255 **COMPUTE** **STATISTICS**

STORAGE(**INITIAL** 65536 **NEXT** 1048576 MINEXTENTS 1 **MAXEXTENTS** 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL **DEFAULT** FLASH\_CACHE **DEFAULT** CELL\_FLASH\_CACHE **DEFAULT**)

**TABLESPACE** **"USERS"** ENABLE

) SEGMENT CREATION **IMMEDIATE**

**PCTFREE** 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

**NOCOMPRESS** LOGGING

STORAGE(**INITIAL** 65536 **NEXT** 1048576 MINEXTENTS 1 **MAXEXTENTS** 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL **DEFAULT** FLASH\_CACHE **DEFAULT** CELL\_FLASH\_CACHE **DEFAULT**)

**TABLESPACE** **"USERS"**;

-- G4USER.PARTICIPANTE definition

**CREATE** **TABLE** "G4USER"."PARTICIPANTE"

( "ID" **NUMBER**(19,0) GENERATED **BY** **DEFAULT** **AS** **IDENTITY** **MINVALUE** 1 **MAXVALUE** 9999999999999999999999999999 **INCREMENT** **BY** 1 **START** **WITH** 1 CACHE 20 NOORDER NOCYCLE NOKEEP NOSCALE **NOT** **NULL** ENABLE,

**"ID\_EVENTO"** **NUMBER**(19,0) **NOT** **NULL** ENABLE,

**"ID\_PARTICIPANTE"** **NUMBER**(19,0) **NOT** **NULL** ENABLE,

**PRIMARY** **KEY** (**"ID"**)

**USING** **INDEX** **PCTFREE** 10 INITRANS 2 MAXTRANS 255 **COMPUTE** **STATISTICS**

STORAGE(**INITIAL** 65536 **NEXT** 1048576 MINEXTENTS 1 **MAXEXTENTS** 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL **DEFAULT** FLASH\_CACHE **DEFAULT** CELL\_FLASH\_CACHE **DEFAULT**)

**TABLESPACE** **"USERS"** ENABLE

) SEGMENT CREATION **IMMEDIATE**

**PCTFREE** 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

**NOCOMPRESS** LOGGING

STORAGE(**INITIAL** 65536 **NEXT** 1048576 MINEXTENTS 1 **MAXEXTENTS** 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL **DEFAULT** FLASH\_CACHE **DEFAULT** CELL\_FLASH\_CACHE **DEFAULT**)

**TABLESPACE** **"USERS"**;

1. Descargar proyectos y repositorios:

**Proyectos principales:**

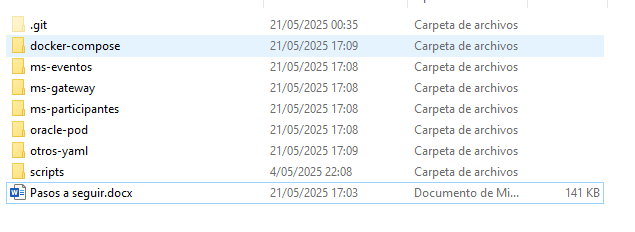
<https://github.com/jbalboar/ms-eventos>

<https://github.com/jbalboar/ms-participantes>

**Proyecto para Spring Cloud Gateway:**

<https://github.com/jbalboar/ms-gateway>

**Proyecto donde se encuentra los distintos yamls y scripts de BD:**



1. Compilar y generar jar

Ubicarse en la ruta de cada proyecto y ejecutar:

.\mvnw clean install -Dmaven.test.skip=true

1. Crear la imagen de ambos proyectos

Ubicarse en la ruta de cada proyecto y ejecutar:

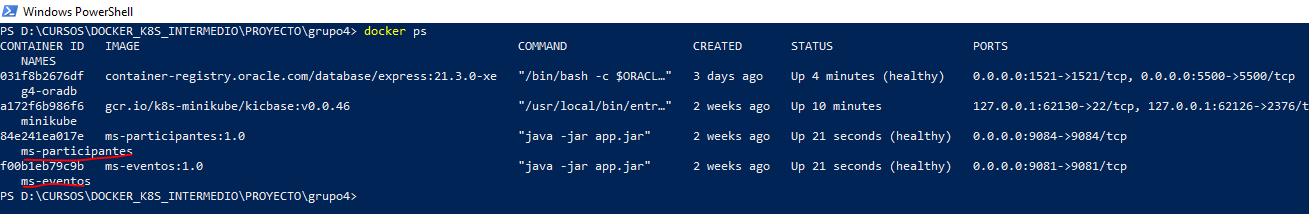
docker build -t ms-eventos:1.0 .

docker build -t ms-participantes:1.0 .

1. Crear y levantar los contenedores

docker run -d --name ms-eventos --network g4-network -p 9081:9081 ms-eventos:1.0

docker run -d --name ms-participantes --network g4-network -p 9081:9081 ms-participantes:1.0



1. Para docker-compose, asegurarse de lo siguiente:

* Detener y eliminar los contenedores:

ms-participantes

ms-eventos

g4-oradb

1. Crear el archivo docker-compose-yml con el siguiente contenido:

services:

  g4-oradb:

    container\_name: g4-oradb

    image: container-registry.oracle.com/database/express:21.3.0-xe

    environment:

      ORACLE\_PWD: g4password

      ORACLE\_SID: XE

      ORACLE\_PDB: XEPDB1

      ORACLE\_CHARACTERSET: AL32UTF8

    ports:

      - "1521:1521"

      - "5500:5500"

    volumes:

      - g4-volume:/opt/oracle/oradata

      - ./scripts:/opt/oracle/scripts/startup

    networks:

      - g4-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

  ms-eventos:

    image: ms-eventos:1.0

    container\_name: ms-eventos

    ports:

      - "9081:9081"

    environment:

      USER\_DEMO: Netec

    networks:

      - g4-network

    depends\_on:

      g4-oradb:

        condition: service\_healthy

    healthcheck:

      test: ["CMD", "curl", "-f", "http://127.0.0.1:9081/eventos"]

      interval: 30s

      timeout: 15s

      retries: 5

      start\_period: 30s

  ms-participantes:

    image: ms-participantes:1.0

    container\_name: ms-participantes

    ports:

      - "9084:9084"

    environment:

      MS\_EVENTOS\_URL: http://ms-eventos:9081/eventos

    networks:

      - g4-network

    depends\_on:

      ms-eventos:

        condition: service\_healthy

    healthcheck:

      test: ["CMD", "curl", "-f", "http://127.0.0.1:9084/participantes"]

      interval: 30s

      timeout: 15s

      retries: 5

      start\_period: 30s

volumes:

  g4-volume:

networks:

  g4-network:

    external: true

1. Crear la carpeta “scripts” en la misma ruta donde se ha creado el archivo “docker-compose.yml”. Dentro de la carpeta script crear el archivo “init\_schema.sql” con el siguiente contenido:

ALTER SESSION SET CONTAINER = XEPDB1;

CREATE USER g4user IDENTIFIED BY g4password

DEFAULT TABLESPACE users

TEMPORARY TABLESPACE temp;

GRANT CONNECT, RESOURCE TO g4user;

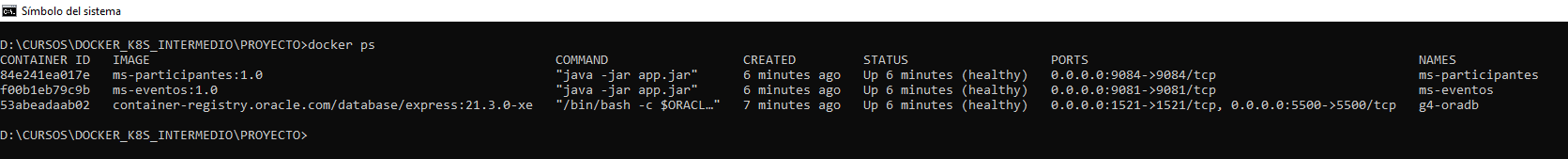
ALTER USER g4user QUOTA UNLIMITED ON users;

ALTER USER g4user QUOTA UNLIMITED ON users;

1. Ejecutar el comando:

docker-compose up -d

1. Verificar que los contenedores se encuentren levantados



KUBERNETES

Para conectarse localmente al pod de k8s

kubectl port-forward pod/oracle-xe 1521:1521

Para probar el ms

kubectl exec -it ms-participantes-59c646b94c-4sjxn-- curl http://localhost:9084/participantes

ms-participantes-59c646b94c-5kj4s

