ACTIVIDAD 5: DOCKERFILE

ÍNDICE

1. Explicación..... 3
1.1 Base y Variables Iniciales.....3
1.2 Argumentos de Construcción.....3
1.3 Instalación de Programas.....3
1.4 Configuración de SSH y Usuario.....3
1.5 Configuración de Bases de Datos.....4
1.6 Puertos.....4
1.7 Comando Final.....4
2. Dockerfile.....5
3. Cómo Usarlo.....6
3.1 Construir la Imagen:.....6
3.2 Ejecutar el Contenedor:.....6
4. Pruebas.....7
4.1 MYSQL.....7
4.2 Postgres.....7
4.3 SSH.....8

1. Explicación

1.1 Base y Variables Iniciales

FROM ubuntu:22.04: Empezamos con una imagen limpia de Ubuntu 22.04. Es nuestro lienzo en blanco.

ENV DEBIAN_FRONTEND=noninteractive: Este es un truco clave. Evita que la instalación se detenga a hacernos preguntas (como la zona horaria). Así, la construcción es 100% automática.

1.2 Argumentos de Construcción

ARG ...: Aquí definimos variables para pasar nuestros usuarios y contraseñas al construir la imagen con docker build. Usamos ARG para no "quemar" las credenciales en el archivo. Es más seguro y reutilizable. Si no pasamos nada, usará los valores por defecto (user, pass, etc.).

1.3 Instalación de Programas

RUN apt-get update && apt-get install -y ...: Actualizamos la lista de paquetes e instalamos todo lo que necesitamos de una vez: el servidor SSH, MySQL, PostgreSQL y la herramienta sudo (para dar permisos de administrador a nuestro usuario).

1.4 Configuración de SSH y Usuario

RUN mkdir... && useradd...: En este bloque hacemos dos cosas:

- Creamos el usuario que nos conectará por SSH, usando las variables **${SSH_USER} y ${SSH_PASSWORD}** que pasamos antes.
- Lo metemos en el grupo **sudo** para que pueda ejecutar comandos como administrador.

1.5 Configuración de Bases de Datos

Por defecto, las bases de datos solo aceptan conexiones desde dentro del propio contenedor. Aquí lo arreglamos:

- MySQL: Modificamos su archivo de configuración (mysqld.cnf) para que escuche conexiones desde cualquier IP (0.0.0.0) y use el puerto 3307.

- PostgreSQL: Hacemos lo mismo, le decimos que escuche en todas partes (*) y use el puerto 5433. Además, añadimos una línea a pg_hba.conf, un archivo de seguridad de Postgres, para permitir explícitamente que cualquiera se conecte desde fuera con contraseña.

1.6 Puertos

EXPOSE 22 3307 5433: Le dice a Docker qué puertos son importantes en esta imagen. Ojo, no los abre al exterior, solo los documenta. Para abrirlos de verdad usamos el flag -p en el comando docker run.

1.7 Comando Final

CMD ...: Arranca el contenedor, ejecutando:

- Arranca los servicios de MySQL y PostgreSQL.

- Espera 5 segundos (muy importante, para darles tiempo a inicializarse bien).

- Ejecuta los comandos SQL para crear el usuario y la base de datos en ambos sistemas.

- Finalmente, arranca el servidor SSH.

- El tail -f /dev/null es un truco para que el contenedor se mantenga encendido y no se apague después de ejecutar los comandos.

2. Dockerfile

```
# 1. BASE Y VARIABLES INICIALES
FROM ubuntu:22.04
ENV DEBIAN_FRONTEND=noninteractive

# 2. ARGUMENTOS DE CONSTRUCCIÓN
ARG SSH_USER=user
ARG SSH_PASSWORD=pass
ARG DB_USER=dbadmin
ARG DB_PASSWORD=dbpass

# 3. INSTALACIÓN DE PROGRAMAS
RUN apt-get update && apt-get install -y \
    openssh-server \
    mysql-server \
    postgresql \
    sudo \
    && rm -rf /var/lib/apt/lists/*

# 4. CONFIGURACIÓN DE SSH Y USUARIO
RUN mkdir -p /var/run/sshd && \
    useradd -rm -d /home/${SSH_USER} -s /bin/bash -g root -G sudo -u 1000 ${SSH_USER} && \
    echo "${SSH_USER}:${SSH_PASSWORD}" | chpasswd

# 5. CONFIGURACIÓN DE BASES DE DATOS
# --- MySQL: Puerto 3307 y conexiones externas ---
RUN sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/mysql.conf.d/mysqld.cnf && \
    sed -i 's/port.*/port = 3307/' /etc/mysql/mysql.conf.d/mysqld.cnf

# --- PostgreSQL: Puerto 5433 y conexiones externas ---
RUN sed -i "s/#listen_addresses = 'localhost'/listen_addresses = '*'/"
/etc/postgresql/14/main/postgresql.conf && \
    sed -i "s/port = 5432/port = 5433/" /etc/postgresql/14/main/postgresql.conf && \
    echo "host all all 0.0.0.0/0 md5" >> /etc/postgresql/14/main/pg_hba.conf

# 6. PUERTOS
EXPOSE 22 3307 5433

# 7. COMANDO FINAL
CMD service mysql start && \
    service postgresql start && \
    sleep 5 && \
    mysql -u root -e "CREATE DATABASE IF NOT EXISTS ${DB_USER}db; CREATE USER IF NOT EXISTS
'${DB_USER}'@'%' IDENTIFIED BY '${DB_PASSWORD}'; GRANT ALL PRIVILEGES ON ${DB_USER}db.* TO
'${DB_USER}'@'%';" && \
    sudo -u postgres psql -c "CREATE DATABASE ${DB_USER}db; CREATE USER ${DB_USER} WITH
ENCRYPTED PASSWORD '${DB_PASSWORD}'; GRANT ALL PRIVILEGES ON DATABASE ${DB_USER}db TO
${DB_USER};" && \
    service ssh start && \
    tail -f /dev/null
```

3. Cómo Usarlo

3.1 Construir la Imagen:

Guarda el contenido anterior en un archivo llamado Dockerfile. Abre una terminal en esa misma carpeta y ejecuta:

```
docker build \
   --build-arg SSH_USER=usuario \
   --build-arg SSH_PASSWORD=Usuario.25 \
   --build-arg DB_USER=usuario \
   --build-arg DB_PASSWORD=Usuario.25 \
   -t actividad5.
```

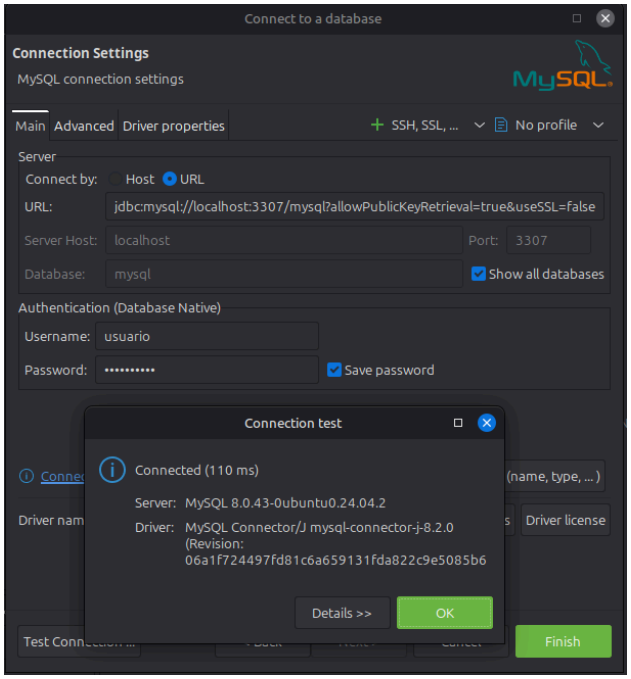
3.2 Ejecutar el Contenedor:

Una vez construida, ejecuta la imagen para crear un contenedor:

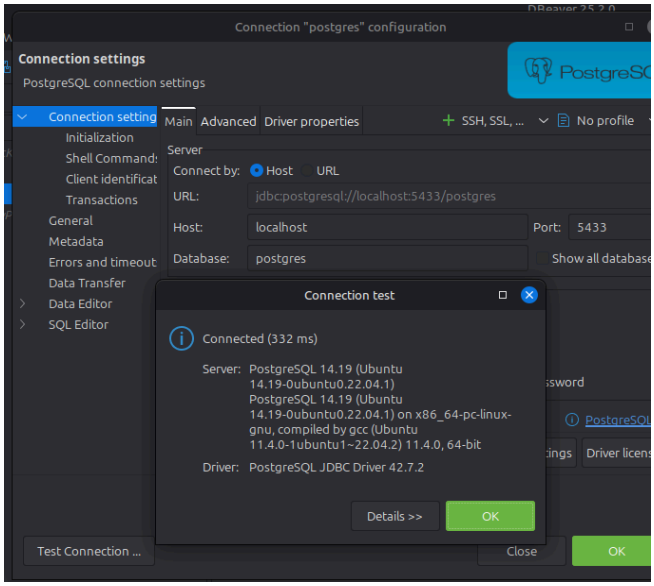
```
docker run -d \
   -p 2222:22 \
   -p 3307:3307 \
   -p 5433:5433 \
   --name act5\
   actividad5
```

4. Pruebas

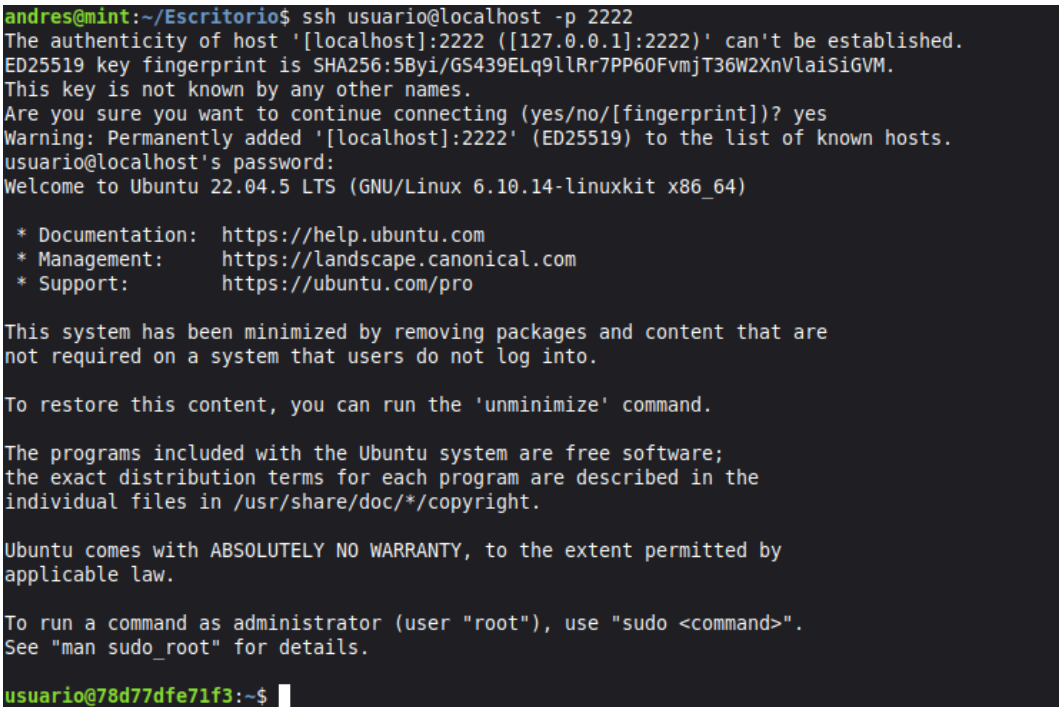
4.1 MYSQL



4.2 Postgres



4.3 SSH


andres@mint:~/Escritorio$ ssh usuario@localhost -p 2222
The authenticity of host '[localhost]:2222 ([127.0.0.1]:2222)' can't be established.
ED25519 key fingerprint is SHA256:5Byi/GS439ELq9llRr7PP6OFvmjT36W2XnVlaiSiGVM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2222' (ED25519) to the list of known hosts.
usuario@localhost's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.10.14-linuxkit x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.
To restore this content, you can run the 'unminimize' command.
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
usuario@78d77dfe71f3:~$