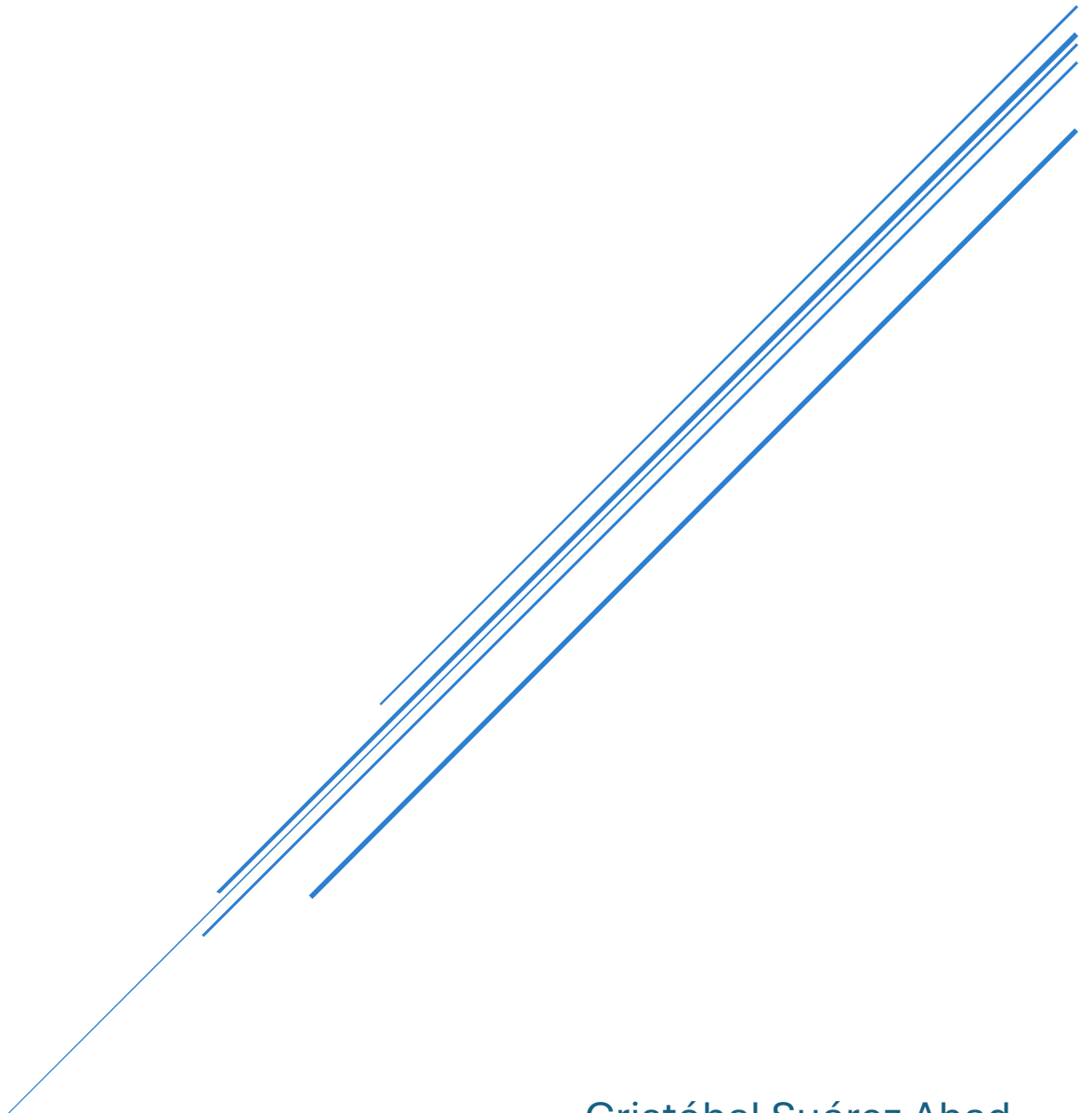


DOCKERFILE MYSQL Y POSTGRESQL

Actividad 5



Cristóbal Suárez Abad
Optativa – 2º ASIR

Contenido

Introducción:	2
1. Parte de una imagen ubuntu server o similar.	3
2. Instalamos paquetería: SSH, MySQL, PostgreSQL y sudo.	3
3. Configuración de variables.	4
4. Configurar SSH.	5
5. Configuración de bases de datos:	6
6. Script de inicialización:	7
7. CMD final.	7
8. “init.sh”	8
9. Capturas de pantalla.	9

Introducción:

1. Parte de una imagen ubuntu server o similar
2. Instalar SSH
3. Queremos instalar MySQL y PostgreSQL, y configurarlos. (Podéis reutilizar lo hecho en la actividad SGBD)
4. En tiempo de construcción debes poder elegir un usuario y contraseña que quieres crear para las dos bases de datos.
5. Cambiar los puertos por defecto de las bases de datos
6. Probar conexión con DBeaver

Tenéis que entregar el dockerfile y un documento explicando cada uno de los pasos pedidos.

1. Parte de una imagen ubuntu server o similar.

FROM ubuntu:22.04 # elegimos la imagen de Ubuntu, la más actual.

ENV DEBIAN_FRONTEND=noninteractive # Evita que los procesos de instalación de paquetes pidan interacción (como confirmaciones), así es más sencillo y rápido crear la imagen.

2. Instalamos paquetería: SSH, MySQL, PostgreSQL y sudo.

4. INSTALACIÓN DE PROGRAMAS

RUN apt-get update && apt-get install -y \

openssh-server \

mysql-server \

postgresql \

sudo \ # **algunos comandos necesitan de ejecutarse con sudo.**

&& rm -rf /var/lib/apt/lists/* # **este último limpia la cache de apt. Así la imagen será más pequeña.**

3. Configuración de variables.

2. ARGUMENTOS DE CONSTRUCCIÓN

ARG SSH_USER=user

ARG SSH_PASSWORD=pass

ARG DB_USER=dbadmin

ARG DB_PASSWORD=dbpass

ARG DB_NAME=prueba_db

ARG: Son variables que solo existen durante la construcción de la imagen (build time). Aquí defines usuarios y contraseñas predeterminados para SSH y bases de datos. También se pueden sobrescribir en el momento de usar **docker build** usando la opción “**--build-arg**”.

3. VARIABLES DE ENTORNO PARA CMD / INIT

ENV SSH_USER=\${SSH_USER} \

SSH_PASSWORD=\${SSH_PASSWORD} \

DB_USER=\${DB_USER} \

DB_PASSWORD=\${DB_PASSWORD} \

DB_NAME=\${DB_NAME}

ENV crea variables de entorno dentro del contenedor, es decir, valores que estarán disponibles para cualquier proceso que se ejecute dentro del contenedor mientras esté corriendo.

A diferencia de **ARG**, que solo existe durante la construcción de la imagen, las variables creadas con **ENV** permanecen durante la ejecución del contenedor y pueden ser accedidas por scripts, servicios o cualquier comando que se ejecute en él.

4. Configurar SSH.

5. CONFIGURACIÓN DE SSH Y USUARIO

RUN mkdir -p /var/run/sshd && \ ## **Crea la carpeta necesaria para SSH**

useradd -m -d /home/\${SSH_USER} -s /bin/bash -g root -G sudo -u 1000 \${SSH_USER}
&& \ # **crea un usuario (nombre, grupo y UID).**

echo "\${SSH_USER}:\${SSH_PASSWORD}" | chpasswd # **asigna la contraseña al usuario
creado en el archivo chpasswd**

EXPOSE 22 # **se documenta el puerto que se abre. Solo documenta.**

5. Configuración de bases de datos:

- MySQL: puerto y conexiones externas.

RUN sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/mysql.conf.d/mysqld.cnf && \ # **Cambia la dirección de enlace de MySQL de localhost a 0.0.0.0, permitiendo conexiones externas.**

sed -i 's/port\s*=.*port = 3307/' /etc/mysql/mysql.conf.d/mysqld.cnf # **abre el Puerto 3307**

EXPOSE 3307 # **documenta que MySQL escuchará en ese puerto.**

- PostgreSQL: puerto y conexiones externas.

RUN sed -i "s/#listen_addresses = 'localhost'/listen_addresses = '*'/'g" /etc/postgresql/14/main/postgresql.conf && \ # **Permite conexiones externas cambiando listen_addresses a *.**

sed -i "s/port = 5432/port = 5433/" /etc/postgresql/14/main/postgresql.conf && \ # **Cambia el puerto 5432 → 5433.**

echo "host all all 0.0.0.0/0 md5" >> /etc/postgresql/14/main/pg_hba.conf # **Añade una regla al pg_hba.conf para aceptar conexiones desde cualquier host con contraseña (md5).**

EXPOSE 5433 # **documenta que PostgreSQL escuchará en ese puerto.**

6. Script de inicialización:

COPY init.sh /init.sh # **Copia el archivo “init.sh” dentro del contenedor. Debemos crearlo previamente y ponerlo en la misma carpeta que el “Dockerfile”.**

RUN chmod +x /init.sh # **Le da permisos de ejecución.**

7. CMD final.

CMD ["/init.sh"] # **simplemente ejecuta el “init.sh”.**

8. “init.sh”

Inicia los servicios

```
service mysql start
```

```
service postgresql start
```

Da 5 segundos de margen para que los servicios se inicien. Esto evita problemas.

```
sleep 5
```

En MySQL se crea la base de datos, el usuario y su contraseña y se le da al usuario los privilegios en la base de datos creada.

```
mysql -u root -e "CREATE DATABASE IF NOT EXISTS ${DB_NAME};"
```

```
mysql -u root -e "CREATE USER IF NOT EXISTS '${DB_USER}'@ '%' IDENTIFIED BY  
'${DB_PASSWORD}';"
```

```
mysql -u root -e "GRANT ALL PRIVILEGES ON ${DB_NAME}.* TO '${DB_USER}'@ '%'; FLUSH  
PRIVILEGES;"
```

Lo mismo, pero en PostgreSQL

```
sudo -u postgres psql -c "CREATE USER ${DB_USER} WITH ENCRYPTED PASSWORD  
'${DB_PASSWORD}';"
```

```
sudo -u postgres psql -c "CREATE DATABASE ${DB_NAME} OWNER ${DB_USER};"
```

```
sudo -u postgres psql -c "GRANT ALL PRIVILEGES ON DATABASE ${DB_NAME} TO  
${DB_USER};"
```

Iniciar el servicio de SSH.

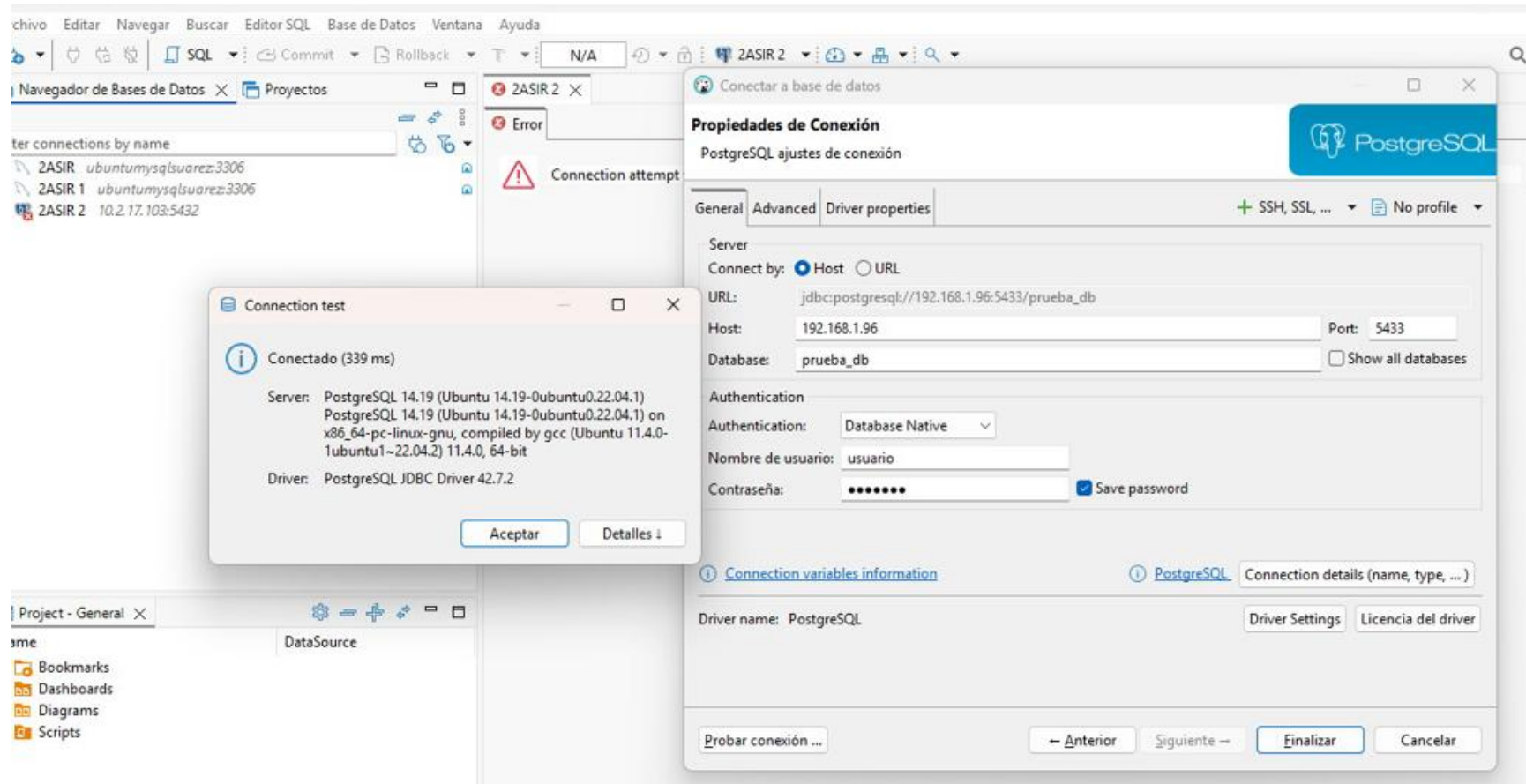
```
service ssh start
```

Comando para mantener el contenedor activo y evitar su cierre.

```
tail -f /dev/null
```

9. Capturas de pantalla.

Conexión con PostgreSQL en DBeaver.



Conexión con SSH.

```
C:\Users\Cristobal>ssh usuario@192.168.1.96 -p 22
usuario@192.168.1.96's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.12.41+deb13-amd64 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@17be75eaca41:~$
```

La conexión de MySQL con DBeaver no ha sido posible. No consigo solucionar el problema.

