# Postgresql con docker

1. Abrir una terminal en la carpeta postgresql. Se recomienda el uso de Windows terminal, ya que permite abrir varias pestañas, así como el uso de diferentes terminales. <https://github.com/microsoft/terminal>
2. Crear una carpeta con nombre ca donde serán guardados los certificados generados y moverse a dicha carpeta.

1. mkdir ca

2. cd ca

1. Ejecutar los siguientes comandos para la creación de los certificados.

1. openssl req -new -text -out server.req

2. openssl rsa -in privkey.pem -out server.key

3. rm privkey.pem

4. openssl req -x509 -in server.req -text -key server.key -out server.crt

5. chmod og-rwx server.key

1. Moverse nuevamente a la carpeta postgresql

1. cd ..

1. Crear un volumen en docker.[[1]](#footnote-1)

1. docker volume create --name=db-data

1. Construir la imagen de postgresql con Docker. Esta imagen es necesario crearla por un problema con los permisos de los certificados[[2]](#footnote-2). La imagen queda con el tag <<postgres\_local>>.

1. docker build -t postgres\_local .

1. Lanzar el contenedor de postgresql con docker-compose.

1. docker-compose up -d

Con estos pasos se consigue lanzar una instancia de PostgreSQL con ssl habilitado. En el archivo docker-compose.yml podemos encontrar que el servicio escucha en el puerto 5432 y que las credenciales con user y password.

Con un gestor como DBeaver o Pgadmin se puede realizar la conexión a la base de datos con las credenciales definidas en el archivo docker-compose.yml. En la Fig. 1 se muestra un ejemplo de conexión utilizando DBeaver.

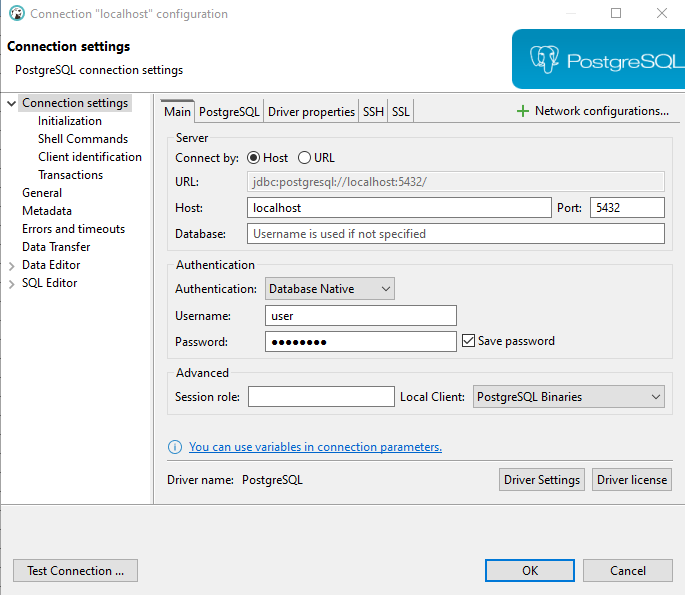


Fig. Configuración en DBeaver de la conexión a la base de datos

Tras conectarse a la base de datos, es necesario crear una base de datos para SSO. Para ello ejecutar el siguiente script:

1. CREATE DATABASE keycloak;

# Red Hat SSO Local

En el siguiente enlace están los pasos detallados para correr Red Hat SSO en ambiente local: <https://access.redhat.com/documentation/es-es/red_hat_single_sign-on/7.6/html-single/getting_started_guide/index#installing-server-product_>

Por el momento no es necesario ejecutar el paso para arrancar el servidor, ya que con la configuración por defecto se utiliza una base de datos H2. Para configurar el uso de postgresql en SSO se debe hacer lo siguiente:

1. Descargar el driver disponible en <https://jdbc.postgresql.org/download/>
2. Modificar el archivo adjunto sso-extensions.cli fijando las credenciales de la base de datos en caso de haberlas modificado en docker-compose.yml y modificando la ruta al driver (postgresql-42.6.0.jar)
3. En una terminal PowerShell moverse a la carpeta rh-sso-7.6.0-server-dist\rh-sso-7.6\bin (la ruta es relativa a la ubicación en donde se descomprimió SSO. Una vez ahí ejecutar:

1. .\jboss-cli.bat --connect --file=C:\Popular\rh-sso-7.6.0-server-dist\rh-sso-7.6\bin\sso-extensions.cli

Aquí el archivo sso-extensions está ubicado en al ruta C:\Popular\rh-sso-7.6.0-server-dist\rh-sso-7.6\bin. Este comando modifica el archivo standalone.xml y agrega el respectivo datasource.

1. Ejecutar el servidor con el comando:

1. .\standalone.bat

1. En caso de no tener Docker instalado, seguir este manual: <https://cuteprogramming.blog/2023/05/21/using-docker-and-kubernetes-without-docker-desktop-on-windows-11/> [↑](#footnote-ref-1)
2. <https://stackoverflow.com/questions/55072221/deploying-postgresql-docker-with-ssl-certificate-and-key-with-volumes> [↑](#footnote-ref-2)