# Run Postgres on Docker

## Download image from docker hub

Texto

El contenido generado por IA puede ser incorrecto.

Now, we can run the command:

Texto, Carta

El contenido generado por IA puede ser incorrecto.

We’ll get previous messages because of the lack of mandatory parameters to initialize the container.

To check the required variables, go to the URL:

<https://github.com/docker-library/docs/blob/master/postgres/README.md>

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Finally, we can run a container with the image:

Texto, Carta

El contenido generado por IA puede ser incorrecto.

Every time we execute: “docker run -e POSTGRES\_PASSWORD=0000 postgres”

a **new container** is created from the postgres image. The previous container is **not reused** unless you explicitly specify the --name option or use docker start on an existing stopped container.

To see all the containers “running”, execute:

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

This won’t show the stopped containers, to check even the stopped containers use:

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

## Get into the container

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

docker run --name some-postgres-container -e POSTGRES\_USER=jrge -e POSTGRES\_DB=docker\_course\_db -e POSTGRES\_PASSWORD=0000 -p 5431:5432 -d postgres

**Breakdown of Each Part:**

1. **docker run** → This command runs a new container from an image.
2. **--name some-postgres-container** → Assigns the container a specific name (some-postgres-container). This makes it easier to manage instead of using auto-generated container names.
3. **-e POSTGRES\_USER=jrge** → Sets the PostgreSQL username to jrge.
4. **-e POSTGRES\_DB=docker\_course\_db** → Creates a default database named docker\_course\_db inside the PostgreSQL container.
5. **-e POSTGRES\_PASSWORD=0000** → Sets the PostgreSQL password to 0000. This is required for authentication.
6. **-p 5431:5432** → Maps port **5432** (the default PostgreSQL port inside the container) to port **5231** on your host machine. This means you can connect to the database using localhost:5231.
7. **-d** → Runs the container in **detached mode** (in the background).
8. **postgres** → The name of the Docker image being used. If the postgres image is not available locally, Docker will pull it from Docker Hub.

**What This Command Does:**

It starts a **PostgreSQL container** with:

* A custom database (docker\_course\_db).
* A specific user (jrge) with password authentication.
* Port forwarding, allowing access from the host machine via localhost:5231.
* Running in the background (-d).

To verify if the postgres server is running we’ll use Dbeaver.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Succeeded

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

# Run keycloak in Docker

Dockerfile to do so:

# Use the official Keycloak image as the builder stage

FROM quay.io/keycloak/keycloak:latest AS builder

# Enable health and metrics support

ENV KC\_HEALTH\_ENABLED=true

ENV KC\_METRICS\_ENABLED=true

# Configure the database vendor

ENV KC\_DB=postgres

WORKDIR /opt/keycloak

# Generate a self-signed certificate (for demonstration purposes only, not for production)

RUN keytool -genkeypair -storepass password -storetype PKCS12 -keyalg RSA -keysize 2048 \

-dname "CN=server" -alias server -ext "SAN:c=DNS:localhost,IP:127.0.0.1" \

-keystore conf/server.keystore

# Build the Keycloak distribution

RUN /opt/keycloak/bin/kc.sh build

# Use the official Keycloak image as the runtime stage

FROM quay.io/keycloak/keycloak:latest

# Copy the built Keycloak distribution from the builder stage

COPY --from=builder /opt/keycloak/ /opt/keycloak/

# Set environment variables for database connection

ENV KC\_DB=postgres

ENV KC\_DB\_URL=jdbc:postgresql://some-postgres-container:5432/docker\_course\_db

ENV KC\_DB\_USERNAME=jrge

ENV KC\_DB\_PASSWORD=0000

ENV KC\_HOSTNAME=localhost

# Expose Keycloak's default HTTP and HTTPS ports

EXPOSE 8080 8443

# Start Keycloak

ENTRYPOINT ["/opt/keycloak/bin/kc.sh", "start", "--optimized"]

docker build -t my-keycloak .

This shows the following:

Texto, Carta

El contenido generado por IA puede ser incorrecto.

We have the image built

Texto

El contenido generado por IA puede ser incorrecto.

Run the container:

docker run -p 8180:8080 -p 8443:8443 my-keycloak

I got the error:

Imagen que contiene Calendario

El contenido generado por IA puede ser incorrecto.

When you run multiple Docker containers (like Keycloak and PostgreSQL), they need to be able to communicate with each other. You can achieve this by creating a custom Docker network, so both containers can resolve each other's hostnames.

## Creating a Custom Network

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

1. Run the command to create a Postgres container using the network created:

docker run --name some-postgres-container --network keycloak-network -e POSTGRES\_USER=jrge -e POSTGRES\_DB=docker\_course\_db -e POSTGRES\_PASSWORD=0000 -p 5431:5432 -d postgres

1. Create container for keycloak in the new network as well.

docker run --name my-keycloak --network keycloak-network -p 8180:8080 -p 8443:8443 my-keycloak

**Explanation:**

* **-p 8180:8080**:  
  This maps port 8080 of the container (Keycloak's default HTTP port) to port 8180 on your **local machine** (host).  
  This means that when you access http://localhost:8180 on your machine, it is forwarded to the Keycloak container's port 8080.
* **-p 8443:8443**:  
  This maps port 8443 of the container (Keycloak's default HTTPS port) to port 8443 on your **local machine** (host).  
  This means that when you access https://localhost:8443, it is forwarded to the Keycloak container's port 8443.

Succeed:

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

We can access to UI:

<http://localhost:8180>

https://localhost:8443

## List available networks

You can list the available networks in Docker using the following command:

docker network ls

This command will display a table with the network ID, name, driver, and scope.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

# The docker compose to run postgres and keycloak

version: '3.8'

services:

postgres:

image: postgres

container\_name: some-postgres-container

environment:

POSTGRES\_USER: jrge

POSTGRES\_DB: docker\_course\_db

POSTGRES\_PASSWORD: "0000"

ports:

- "5432:5432"

networks:

- keycloak-network

keycloak:

image: my-keycloak

container\_name: my-keycloak

depends\_on:

- postgres

ports:

- "8180:8080"

- "8443:8443"

networks:

- keycloak-network

networks:

keycloak-network:

driver: bridge

**Explanation:**

* Defines two services: postgres and keycloak.
* Both services are assigned to the same keycloak-network.
* PostgreSQL is exposed on port 5432:5432 and Keycloak on 8180:8080 & 8443:8443.
* The depends\_on ensures Keycloak starts after PostgreSQL.

**Running the Containers:**

1. Save the file as docker-compose.yml.
2. Run the following command to start the services:

docker-compose up -d

1. To check the running containers:

docker ps

I got this:

Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Observe when executing the command it creates a new network:

Texto

El contenido generado por IA puede ser incorrecto.

## Create a Realm

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

## Create a client

Will be created inside the Realm

Una captura de pantalla de una computadora

El contenido generado por IA puede ser incorrecto.

* Client ID: spring-boot-client
* Client Protocol: openid-connect
* Access Type: confidential
* Set Redirect URIs: [http://localhost:8180/\*](http://localhost:8180/*)

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

## Creating a Spring Boot Application