

DevOps Internship

-Exercițiul 3-

1. Am creat Dockerfile

```
ex3 > Dockerfile
1 # Am folosit imagine de postgres
2 FROM postgres:14
3
4 #variabilele de mediu luate din .env
5 ENV POSTGRES_USER=${POSTGRES_USER}
6 ENV POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
7 ENV POSTGRES_DB=${POSTGRES_DB}
8
9 #volum persistent
10 VOLUME ["/var/lib/postgresql/data"]
11
12 EXPOSE 5432
13
14 #rulam postgres
15 CMD ["postgres"]
--
```

2. Pentru a stoca variabilele de mediu într-un mod sigur, am creat un fișier .env în care le stocăm. Fișierul .env este urcat pe github fără date confidențiale, apoi modificările sunt ignorate folosind comanda `git update-index --assume-unchanged`

```
ex3 > .env
1 POSTGRES_USER=ituser
2 POSTGRES_PASSWORD=****
3 POSTGRES_DB=company_db
```

3. Am dat build la imagine

```
PS C:\Users\Gabriela\Desktop\DevOps\ex3> docker build -t postgres .
[+] Building 164.9s (6/6) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 280B
=> [internal] load metadata for docker.io/library/postgres:14
=> [auth] library/postgres:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/1] FROM docker.io/library/postgres:14@sha256:4e8c2e92ad1fa2c3ea5efca42540e626a362d467e50bec75f9afecf2666f2f5b
=> => resolve docker.io/library/postgres:14@sha256:4e8c2e92ad1fa2c3ea5efca42540e626a362d467e50bec75f9afecf2666f2f5b
=> => sha256:e9d1d4c20f1d179dd339b4c5d4fd264e10d4fb00bbe293e73f7ec7f52f20866d 116B / 116B
=> => sha256:97f8e7617a0f2695d47b8cdfef2bc6cf9a721a25ede7ecdb7ee656e89e4fd4a1 9.53kB / 9.53kB
=> => sha256:d7125325320af1fcc7782cbcefa349c1e9da14df39ad7b92258167881b11f0d4 129B / 129B
=> => sha256:17bae54169d8534ccbd9b57ed09ad8e14cc770acd6978588ff04f9a363b3df 185B / 185B
=> => sha256:13aad1eb8ef9af7b7e7ccbcfef0987e2dbfc6826eca212c4bbe206af8484ad84 108.60MB / 108.60MB
=> => sha256:9421c3677c88ec77aa882630a65431ec03de4943a011d4f5a45765565143d7b1 8.07MB / 8.07MB
=> => sha256:b32c48c7612c63005e7a88a7d25c0575b824bf1ec8c3cd857773a3016649d1e9 3.14kB / 3.14kB
=> => sha256:6d08ba40cea036e40932be8f326a88fb79dd4f8d801a50e49013147c862e1385 1.45MB / 1.45MB
=> => sha256:cbf06ab180b7df345139d7145ac2f73b9cfc152924e9678601fc2cca75efac7d 1.20MB / 1.20MB
=> => sha256:a86ca505ec22779053226793596cfff413604d1d585547f2463ca065d01ebe01e 5.47kB / 5.47kB
=> => sha256:9f43c5189c92348826227b001fd8cb57e42264c4eb552801d8601a45e3d5c2e3 168B / 168B
=> => sha256:50050b48d80cc0fd6f650255439f2d1c66c18187f2b2197cfa993983453762fba 1.17kB / 1.17kB
=> => sha256:9e050e74cc973808bfa56db1238bcf4f22e9919319c615e165c386e5970c3ff2 4.53MB / 4.53MB
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:70d225d54d63fa24a907396e6af6f280e96ed6cc030d702e436dc3e3e1c0f6ae
=> => exporting config sha256:98c194a4efc51b73e39ed1e7eb5f48b4519bb445ef847d126fc87b4ef0840145
=> => exporting attestation manifest sha256:c020cef5c076066844d4c8694c7f82773ec7c1356cbf2955def35867cd2a6710
=> => exporting manifest list sha256:5d2e16c27b5fe69340c8d6a783348a30abf0ba7b8d3a5038e721b527ebcae78d
=> => naming to docker.io/library/postgres:latest
=> => unpacking to docker.io/library/postgres:latest
```

View build details: [docker-desktop://dashboard/build/desktop-linux/desktop-linux/3xsn828in2rf8xkzyujgk58](https://dashboard/build/desktop-linux/desktop-linux/3xsn828in2rf8xkzyujgk58)

4. Am rulat imaginea

```
PS C:\Users\Gabriela\Desktop\DevOps\ex3> docker run --name postgres -p 5432:5432 -v persistent:/var/lib/postgresql/data --env-file .env postgres
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.
```

```
The database cluster will be initialized with locale "en_US.utf8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".
```

```
Data page checksums are disabled.
```

```
fixing permissions on existing directory /var/lib/postgresql/data ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... Etc/UTC
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... initdb: warning: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.
ok
```

```
Success. You can now start the database server using:
```

```
pg_ctl -D /var/lib/postgresql/data -l logfile start
```

5. Am copiat fișierul populatedb în container și l-am rulat. A trebuit să șterg insert-urile de la salarii cu id>53 pentru că angajații nu existau.

```
PS C:\Users\Gabriela\Desktop\DevOps\ex3> docker cp C:\Users\Gabriela\Desktop\DevOps\ex3\populatedb.sql postgres:/home/
Successfully copied 5.12kB to postgres:/home/
PS C:\Users\Gabriela\Desktop\DevOps\ex3> docker exec -it postgres psql -U ituser -d company_db -f /home/populatedb.sql
psql:/home/populatedb.sql:2: ERROR: syntax error at or near "USE"
LINE 1: USE company_db;
```

```
^
CREATE TABLE
CREATE TABLE
CREATE TABLE
INSERT 0 8
INSERT 0 53
INSERT 0 53
```

```
PS C:\Users\Gabriela\Desktop\DevOps\ex3> docker exec -it postgres psql -U ituser -d company_db
psql (14.17 (Debian 14.17-1.pgdg120+1))
Type "help" for help.
```

```
company_db=# select * from employees;
```

employee_id	first_name	last_name	department_id
1	Alice	Smith	1
2	George	Anderson	1
3	Mia	Rodriguez	1
4	Ethan	Harris	1
5	Sophia	King	1
6	Bob	Johnson	2
7	Charlie	Brown	2
8	Hannah	Martinez	2
9	Noah	Lewis	2
10	Liam	Clark	2
11	Emma	Davis	2
12	Olivia	Taylor	2
13	Lucas	Baker	2
14	Ava	Evans	2
15	William	Nelson	2
16	David	Williams	3
17	Isaac	Thomas	3
18	Jack	White	3
19	Grace	Robinson	3
20	Daniel	Scott	3
21	Victoria	Lopez	3
22	Frank	Miller	4
23	Karen	Lopez	4
24	Henry	Wright	4
25	Zoe	Allen	4
26	Elijah	Parker	4
27	Charlotte	Adams	4
28	Samuel	Gonzalez	5
29	Eleanor	Carter	5
30	Jacob	Mitchell	5
31	Michael	Perez	5
32	Scarlett	Turner	5

```
:|
```

6. Am rulat query-urile pentru informațiile cerute:

a. Pentru numărul de angajați:

```
PS C:\Users\Gabriela\Desktop\DevOps\ex3> docker exec -it postgres psql -U ituser -d company_db
psql (14.17 (Debian 14.17-1.pgdg120+1))
Type "help" for help.
```

```
company_db=# select count(employee_id) from employees;
 count
-----
      53
(1 row)
```

```
company_db=#
```

b. Numele angajaților dintr-un departament anume (Marketing)

```
company_db=# select first_name, last_name from employees where department_id=(select department_id from departments where department_name='Marketing');
 first_name | last_name
-----
 Frank      | Miller
 Karen      | Lopez
 Henry      | Wright
 Zoe         | Allen
 Elijah     | Parker
 Charlotte  | Adams
(6 rows)
```

c. Cel mai mare și cel mai mic salariu pe departament

```
company_db=# SELECT
 departments.department_name,
 MAX(salaries.salary) AS maxim,
 MIN(salaries.salary) AS minim
 FROM departments
 JOIN employees ON departments.department_id=employees.department_id
 JOIN salaries ON employees.employee_id=salaries.employee_id
 GROUP BY departments.department_name;
 department_name |    maxim    |    minim
-----
 Customer Support | 119000.00 | 109000.00
 Marketing        |  91000.00 |  78000.00
 Operations       | 131000.00 | 121000.00
 Sales            | 107000.00 |  93000.00
 Legal            | 143000.00 |  91000.00
 IT               |  94000.00 |  67000.00
 Finance          |  76000.00 |  62000.00
 HR               |  60000.00 |  50000.00
(8 rows)
```

7. Pentru dump am folosit pg_dump apoi am copiat dump-ul pe calculator

```
PS C:\Users\Gabriela\Desktop\DevOps\ex3> docker exec -it postgres pg_dump -U ituser -d company_db -f /home/dump.sql
PS C:\Users\Gabriela\Desktop\DevOps\ex3> docker cp postgres:/home/dump.sql dump.sql
Successfully copied 8.7kB to C:\Users\Gabriela\Desktop\DevOps\ex3\dump.sql
PS C:\Users\Gabriela\Desktop\DevOps\ex3>
```