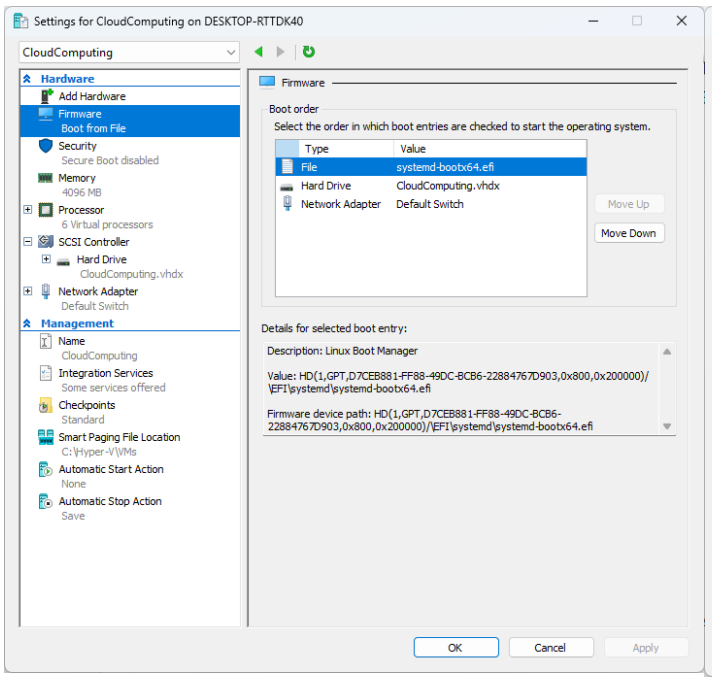
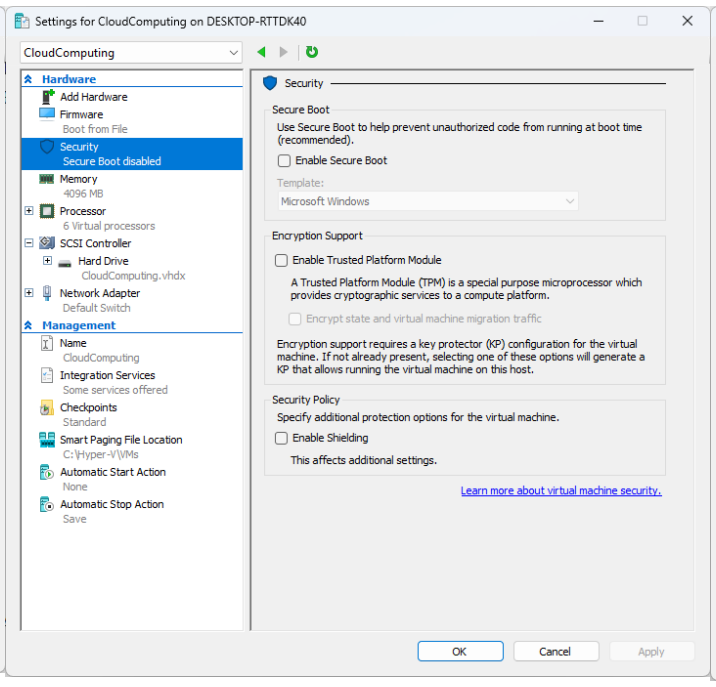
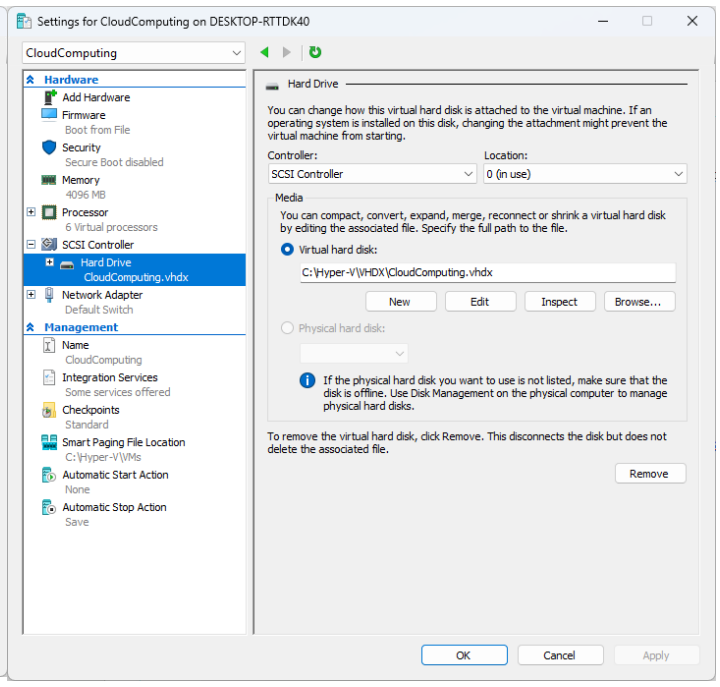
Опис середовища

Середовищем виконання лабораторних робіт 1 і 2 є Arch Linux VM. Технологія віртуалізації - Hyper-V.
Деякі налаштування віртуальної машини наведено на скріншотах:





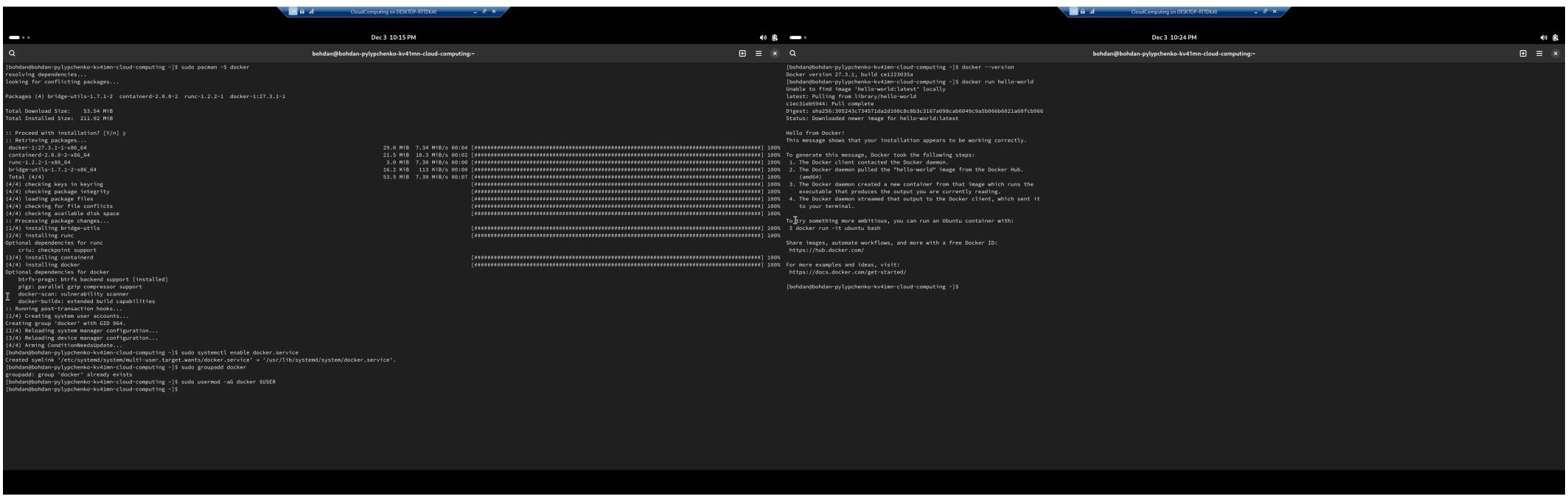


Встановлення Arch Linux у віртуальній машині виконувалось за допомогою офіційного скрипту archinstall.

Реалізація лабораторних робіт використовує додаткові утиліти:

1. Docker (engine, compose).— з метою реалізації принципу infrastructure as a code, зручного керування всіма компонентами рішення.
2. PgAdmin—для окремого доступу до баз даних щоб додатково перевіряти дію «додатку» на базу даних.

Встановлення Docker engine та Docker compose:



Для потреб лабораторних робіт достатньо використовувати pgAdmin docker image:

CloudComputing on DESKTOP-RTTDK40

Dec 3 10:59 PM

bohdan@bohdan-pylypchenko-kv41mn-cloud-computing:~

```
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing ~]$ docker pull dpage/pgadmin4
Using default tag: latest
latest: Pulling from dpage/pgadmin4
43c4264eed91: Pull complete

bf21c2f58347: Pull complete

71389c0d821c: Pull complete

ae4e538450a4: Pull complete

2d9b01174118: Pull complete

3bd152afc50c: Pull complete

ee953f6212a6: Pull complete

c6e13029cc02: Pull complete

d3c69bd584f2: Pull complete

c299fa72acbb: Pull complete

c9524e25ca26: Pull complete

9d0b3ea70124: Pull complete

cec06255d74b: Pull complete

dd62840fe7de: Pull complete

1ac3d0782310: Pull complete

2ff470a2408c: Pull complete

Digest: sha256:561c1f8f99f2fe24bc63814db9e30292cf4360331de9182da21e3554ce61bdaa
Status: Downloaded newer image for dpage/pgadmin4:latest
docker.io/dpage/pgadmin4:latest
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing ~]$ docker run -p 80:80 \

    -e 'PGADMIN_DEFAULT_EMAIL=bp@lab1.com' \

    -e 'PGADMIN_DEFAULT_PASSWORD=123' \

    -d dpage/pgadmin4
24ae63ab2fba38404a7e26c49adac542d8bc9c887a3d42e2224ba61222adcba1
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing ~]$ docker ps
CONTAINER ID   IMAGE            COMMAND            CREATED          STATUS          PORTS                                                  NAMES
24ae63ab2fba   dpage/pgadmin4   "/entrypoint.sh"   42 seconds ago   Up 42 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp, 443/tcp   inspiring_buck
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing ~]$
```

pgAdmin 4

localhost/browser/

pgAdmin File Object Tools Help bp@lab1.com (internal)

Object Exp

Servers

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes ×

Welcome

pgAdmin

Management Tools for PostgreSQL

Feature rich | Maximises PostgreSQL | Open Source

pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.

Quick Links

Add New Server

Configure pgAdmin

Getting Started

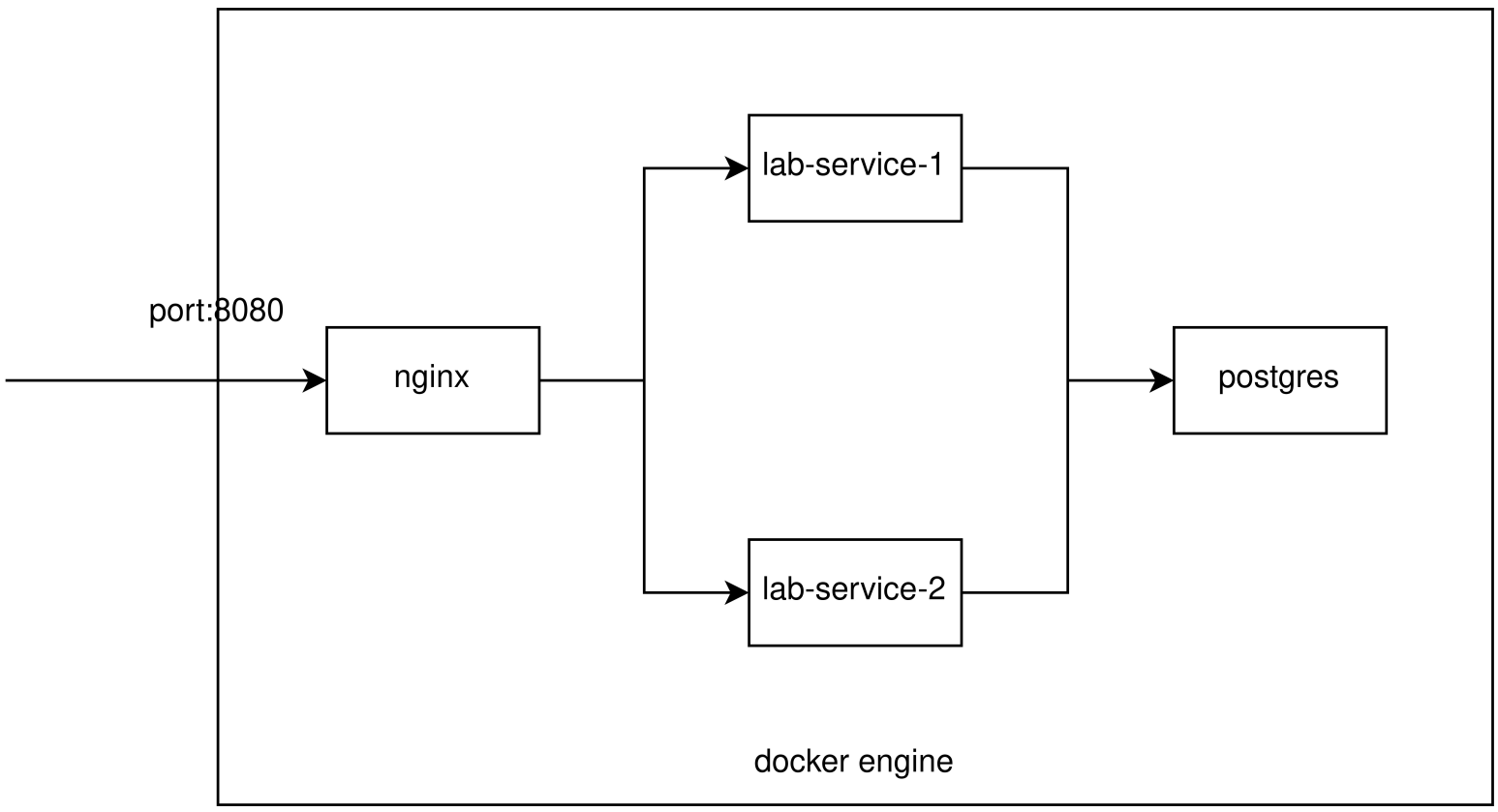
PostgreSQL Documentation

pgAdmin Website

Planet PostgreSQL

Community Support

Опис реалізації



Код реалізації доступний на github: https://github.com/Bohdan628318ylypchenko/CloudComputing (тег lab1).

```
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ tree
.
├── docker-compose.yml
├── lab-service
│   ├── Dockerfile
│   ├── main.py
│   └── requirements.txt
├── nginx
│   ├── Dockerfile
│   └── nginx.conf
└── postgres
    ├── Dockerfile
    └── init.sql
```

Рішення складається з компонентів:

- postgres: сервер-база даних, містить базу “lab1”, що має 1 таблицю “logs”. Схема logs: id (primary key), writer_ip (varchar(32)), message (varchar(255)). Таблиця logs створюється автоматично під час запуску контейнера шляхом виконання init.sql:

init.sql:

```sql
lab1 > postgres > init.sql
1   CREATE TABLE logs (
2       id SERIAL PRIMARY KEY,
3       writer_ip VARCHAR(32),
4       message VARCHAR(255)
5   );
```

Dockerfile (postgres):

```
lab1 > postgres > Dockerfile
1   FROM postgres:17.2
2
3   COPY init.sql /docker-entrypoint-initdb.d/
4
```

- lab-service: python fastapi додаток. Містить 2 ендпоінти: POST /action і GET /logs. Action ендпоінт генерує випадкове повідомлення, записує свою ip адресу та згенероване повідомлення в таблицю logs, повертає ip і згенероване повідомлення. Logs ендпоінт повертає всі записи з таблиці logs. Весь код додатку міститься у файлі main.py:

lab1 > lab-service > main.py

```python
from fastapi import FastAPI
import random
import psycopg2
import socket


app = FastAPI()


conn = psycopg2.connect(
    host="db",
    port=5432,
    dbname="lab1",
    user="postgres",
    password="postgres"
)


@app.post("/action")
def action():
    message = " ".join(random.choices(["cat", "dog", "apple", "orange", "banana"], k=5))
    writer_ip = socket.gethostbyname(socket.gethostname())

    try:
        cursor = conn.cursor()
        cursor.execute("INSERT INTO logs (writer_ip, message) VALUES (%s, %s)", (writer_ip, message))
        conn.commit()
        cursor.close()
    except Exception as e:
        return {"error": str(e)}

    return {"writer_ip": writer_ip, "message": message}


@app.get("/logs")
def get_logs():
    try:
        cursor = conn.cursor()
        cursor.execute("SELECT id, writer_ip, message FROM logs")
        rows = cursor.fetchall()
        cursor.close()
    except Exception as e:
        return {"error": str(e)}

    return {"logs": [{"id": row[0], "writer_ip": row[1], "message": row[2]} for row in rows]}

```

Залежності додатку описано у requirements.txt:

lab1 > lab-service > requirements.txt

```
fastapi==0.103.1
uvicorn==0.22.0
psycopg2-binary==2.9.10

```

Dockerfile додатку:

lab1 > lab-service > Dockerfile

```dockerfile
FROM python:3.12-slim

WORKDIR /app

COPY requirements.txt /app/
RUN pip install -r requirements.txt

COPY main.py /app

CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]
```

- nginx: розподіляє вхідне навантаження, що йде на порт 8080, між 2 інстансами lab-service.

nginx.conf:

lab1 > nginx > nginx.conf

```
events {}

http {
    upstream lab_service {
        server lab-service1:8000;
        server lab-service2:8000;
    }

    server {
        listen 8080;

        location / {
            proxy_pass http://lab_service;
        }
    }
}
```

Dockerfile (nginx):

lab1 > nginx > Dockerfile

```
FROM nginx:1.27.3

COPY nginx.conf /etc/nginx/nginx.conf
```

Структура рішення описана за допомогою docker-compose.yml:

lab1 > docker-compose.yml

```
services:
  db:
    build:
      context: ./postgres
    container_name: db
    ports:
      - "5432:5432"
    environment:
      POSTGRES_DB: lab1
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
    healthcheck:
      test: ["CMD-SHELL", "pg_isready --username=postgres --dbname=lab1"]
      interval: 5s
      timeout: 5s
      retries: 5

  lab-service1:
    build:
      context: ./lab-service
    container_name: lab-service1
    depends_on:
      db:
        condition: service_healthy

  lab-service2:
    build:
      context: ./lab-service
    container_name: lab-service2
    depends_on:
      db:
        condition: service_healthy

  nginx:
    build:
      context: ./nginx
    container_name: nginx
    ports:
      - "8080:8080"
    depends_on:
      - lab-service1
      - lab-service2
```

Варто виділити наявність healthcheck у сервісі бази даних. Сервер бази даних не є доступним відразу по старту db service. За

відсутності healthcheck підняття сервісів, які depends_on db, відбувається відразу по запуску db сервісу, зокрема lab-service1/2. Тоді Lab-service інтсанси намагатимуться створити з’єднання з неініціалізованою базою даних, що призведе до runtime error і, як наслідок, “падіння” lab-service1/2. Ця ситуація унеможливлюється завдяки використанню healthcheck та depends_on із condition: lab-service1/2 запустяться лише тоді, коли сервер бази даних ЗАКІНЧИТЬ ініціалізувати базу даних lab1 (“pg_isready –username=postgres –dbname=lab1”).

Демонстрація роботи реалізації

Зберемо рішення виконавши команду docker compose build:

```
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ ls -al
total 4
drwxr-xr-x 1 bohdan bohdan 100 Dec  4 13:05 .
drwxr-xr-x 1 bohdan bohdan  38 Dec  4 13:00 ..
-rw-r--r-- 1 bohdan bohdan 818 Dec  4 01:44 docker-compose.yml
drwxr-xr-x 1 bohdan bohdan  66 Dec  4 01:44 lab-service
drwxr-xr-x 1 bohdan bohdan  40 Dec  4 01:44 nginx
drwxr-xr-x 1 bohdan bohdan  36 Dec  4 01:44 postgres
drwxr-xr-x 1 bohdan bohdan 370 Dec  4 14:29 звіт
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ docker compose build
[+] Building 0/0
[+] Building 0/1uilding                                                                                                                                0.1s
[+] Building 30.2s (7/8)                                                                                                                    docker:default
 => [db internal] load build definition from Dockerfile                                                                                                0.2s
[+] Building 53.6s (23/26)                                                                                                                  docker:default
[+] Building 63.3s (28/33)                                                                                                                  docker:default
 ✔ Service db            Built                                                                                                                        30.4s
 => => transferring context: 2B                                                                                                                        0.0s
 => [lab-service1 internal] load .dockerignore                                                                                                         0.7s
[+] Building 67.9s (32/35)                                                                                                                  docker:default
 => [db internal] load build definition from Dockerfile                                                                                                0.2s
 => => transferring dockerfile: 100B                                                                                                                   0.0s
 => [db internal] load metadata for docker.io/library/postgres:17.2                                                                                    2.0s
 => [db internal] load .dockerignore                                                                                                                   0.2s
 => => transferring context: 2B                                                                                                                        0.0s
 => [db internal] load build context                                                                                                                   0.4s
 => => transferring context: 137B                                                                                                                      0.0s
 => [db 1/2] FROM docker.io/library/postgres:17.2@sha256:fe4efc6901dda0d952306fd962643d8022d7bb773ffe13fe8a21551b9276e50c                             26.1s
 => => resolve docker.io/library/postgres:17.2@sha256:fe4efc6901dda0d952306fd962643d8022d7bb773ffe13fe8a21551b9276e50c                                 0.3s
 => => sha256:67e602605d99b3135f3828366b182306f4cd3e9a4b0c98eb725fe6d21d45c8c4 3.63kB / 3.63kB                                                         0.0s
 => => sha256:810c36706d001955d088d9ab907e2d5b9937ecafb23e56aa01073fdafad6f4a4 9.92kB / 9.92kB                                                         0.0s
 => => sha256:fe4efc6901dda0d952306fd962643d8022d7bb773ffe13fe8a21551b9276e50c 10.26kB / 10.26kB                                                       0.0s
 => => sha256:a24f300391ed560110f7db7fb4da16df19606c8beeb61936365711cc7609e4a9 4.53MB / 4.53MB                                                         0.9s
 => => sha256:002e1a8eb6f979b0de364bf1657a9259b3523a1c9e8a6d6ac6b9e3a0777afb93 1.17kB / 1.17kB                                                         0.2s
 => => sha256:bc0965b23a04fe7f2d9fb20f597008fcf89891de1c705ffc1c80483a1f098e4f 28.23MB / 28.23MB                                                       4.5s
 => => sha256:627f580b7ad7eae5de5a735e41bd210ba17351399d3d2c211ecfbaadf1c3ae17 1.45MB / 1.45MB                                                         1.4s
 => => sha256:cfb3c2203f88521b7100200c81b57aa1ec9cb9b1a1921ab3e5e39934ae619cf1 8.07MB / 8.07MB                                                         3.1s
 => => sha256:9e592465b24339421141d5c8ebf8b8cf50ca92f94e3590ad3a616fbfa890e833 1.20MB / 1.20MB                                                         2.4s
 => => sha256:8d4265d09d9cdac15612ec9d293497e286089ff0777daffa8b5638b734c7d316 114B / 114B                                                             2.8s
 => => sha256:e3a8293e92fdc474ad5bc8500daa2a3208051960cbfaafd185919bef75a9e1d0 3.14kB / 3.14kB                                                         3.2s
 => => sha256:2cb801c394368ce9eead7a09e97dcee1a81ad0848751fcd7ad9bd8287e7a0d47 110.30MB / 110.30MB                                                     13.8s
 => => sha256:c5fdb20d8658f2e85ccc7008be082fa3e86f378f007cf387971486f816fb7743 10.23kB / 10.23kB                                                       3.4s
 => => sha256:67c5fe618f0cf74ad1a0d31965795dc73bb1de11f7a9e8f26b4886e00e5868a5 128B / 128B                                                             3.7s
 => => sha256:c9cdd1fe82e4c0cac2f8dfbd562ce2d9b79a03700c6bdee3bab7dcfd925b836b 167B / 167B                                                             3.9s
 => => sha256:8f152c4aceed2dd762792866e573f6c078abb88b18778224e05b71f8d3bbac37 5.42kB / 5.42kB                                                         4.2s
 => => sha256:2cd360f3b7db41d99ee40210a8b6801589ce86f26b0ecc063a7ba8416ffc2d7a 185B / 185B                                                             4.4s
 => => extracting sha256:bc0965b23a04fe7f2d9fb20f597008fcf89891de1c705ffc1c80483a1f098e4f                                                              3.3s
 => => extracting sha256:002e1a8eb6f979b0de364bf1657a9259b3523a1c9e8a6d6ac6b9e3a0777afb93                                                              0.0s
 => => extracting sha256:a24f300391ed560110f7db7fb4da16df19606c8beeb61936365711cc7609e4a9                                                              0.3s
 => => extracting sha256:627f580b7ad7eae5de5a735e41bd210ba17351399d3d2c211ecfbaadf1c3ae17                                                              0.1s
 => => extracting sha256:cfb3c2203f88521b7100200c81b57aa1ec9cb9b1a1921ab3e5e39934ae619cf1                                                              0.9s
 => => extracting sha256:9e592465b24339421141d5c8ebf8b8cf50ca92f94e3590ad3a616fbfa890e833                                                              0.2s
 => => extracting sha256:8d4265d09d9cdac15612ec9d293497e286089ff0777daffa8b5638b734c7d316                                                              0.0s
```

Повний вивід команд:

```
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ ls -al
total 4
drwxr-xr-x 1 bohdan bohdan 100 Dec  4 13:05 .
drwxr-xr-x 1 bohdan bohdan  38 Dec  4 13:00 ..
-rw-r--r-- 1 bohdan bohdan 818 Dec  4 01:44 docker-compose.yml
drwxr-xr-x 1 bohdan bohdan  66 Dec  4 01:44 lab-service
drwxr-xr-x 1 bohdan bohdan  40 Dec  4 01:44 nginx
drwxr-xr-x 1 bohdan bohdan  36 Dec  4 01:44 postgres
drwxr-xr-x 1 bohdan bohdan 370 Dec  4 14:29 звіт
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ docker compose build
[+] Building 0/0
[+] Building 0/1uilding
0.1s
[+] Building 30.2s (7/8)
docker:default
 => [db internal] load build definition from Dockerfile
0.2s
[+] Building 53.6s (23/26)
docker:default
[+] Building 63.3s (28/33)
docker:default
 ✔ Service db            Built
30.4s
 => => transferring context: 2B
0.0s
 => [lab-service1 internal] load .dockerignore
0.7s
[+] Building 67.9s (32/35)
docker:default
 => [db internal] load build definition from Dockerfile
0.2s
 => => transferring dockerfile: 100B
0.0s
 => [db internal] load metadata for docker.io/library/postgres:17.2
2.0s
 => [db internal] load .dockerignore
0.2s
 => => transferring context: 2B
0.0s
 => [db internal] load build context
0.4s
 => => transferring context: 137B
0.0s
 => [db 1/2] FROM docker.io/library/postgres:17.2@sha256:fe4efc6901dda0d952306fd962643d8022d7bb773ffe13fe8a21551b9276e50c
26.1s
 => => resolve docker.io/library/postgres:17.2@sha256:fe4efc6901dda0d952306fd962643d8022d7bb773ffe13fe8a21551b9276e50c
0.3s
 => => sha256:67e602605d99b3135f3828366b182306f4cd3e9a4b0c98eb725fe6d21d45c8c4 3.63kB / 3.63kB
0.0s
 => => sha256:810c36706d001955d088d9ab907e2d5b9937ecafb23e56aa01073fdafad6f4a4 9.92kB / 9.92kB
0.0s
 => => sha256:fe4efc6901dda0d952306fd962643d8022d7bb773ffe13fe8a21551b9276e50c 10.26kB / 10.26kB
0.0s
 => => sha256:a24f300391ed560110f7db7fb4da16df19606c8beeb61936365711cc7609e4a9 4.53MB / 4.53MB
0.9s
 => => sha256:002e1a8eb6f979b0de364bf1657a9259b3523a1c9e8a6d6ac6b9e3a0777afb93 1.17kB / 1.17kB
0.2s
 => => sha256:bc0965b23a04fe7f2d9fb20f597008fcf89891de1c705ffc1c80483a1f098e4f 28.23MB / 28.23MB
4.5s
 => => sha256:627f580b7ad7eae5de5a735e41bd210ba17351399d3d2c211ecfbaadf1c3ae17 1.45MB / 1.45MB
1.4s
 => => sha256:cfb3c2203f88521b7100200c81b57aa1ec9cb9b1a1921ab3e5e39934ae619cf1 8.07MB / 8.07MB
3.1s
 => => sha256:9e592465b24339421141d5c8ebf8b8cf50ca92f94e3590ad3a616fbfa890e833 1.20MB / 1.20MB
2.4s
 => => sha256:8d4265d09d9cdac15612ec9d293497e286089ff0777daffa8b5638b734c7d316 114B / 114B
2.8s
 => => sha256:e3a8293e92fdc474ad5bc8500daa2a3208051960cbfaafd185919bef75a9e1d0 3.14kB / 3.14kB
3.2s
 => => sha256:2cb801c394368ce9eead7a09e97dcee1a81ad0848751fcd7ad9bd8287e7a0d47 110.30MB / 110.30MB
13.8s
 => => sha256:c5fdb20d8658f2e85ccc7008be082fa3e86f378f007cf387971486f816fb7743 10.23kB / 10.23kB
3.4s
 => => sha256:67c5fe618f0cf74ad1a0d31965795dc73bb1de11f7a9e8f26b4886e00e5868a5 128B / 128B
3.7s
 => => sha256:c9cdd1fe82e4c0cac2f8dfbd562ce2d9b79a03700c6bdee3bab7dcfd925b836b 167B / 167B
3.9s
 => => sha256:8f152c4aceed2dd762792866e573f6c078abb88b18778224e05b71f8d3bbac37 5.42kB / 5.42kB
4.2s
 => => sha256:2cd360f3b7db41d99ee40210a8b6801589ce86f26b0ecc063a7ba8416ffc2d7a 185B / 185B
4.4s
 => => extracting sha256:bc0965b23a04fe7f2d9fb20f597008fcf89891de1c705ffc1c80483a1f098e4f
3.3s
 => => extracting sha256:002e1a8eb6f979b0de364bf1657a9259b3523a1c9e8a6d6ac6b9e3a0777afb93
0.0s
```

```
 => => extracting sha256:a24f300391ed560110f7db7fb4da16df19606c8beeb6193636571lcc7609e4a9
0.3s
 => => extracting sha256:627f580b7ad7eae5de5a735e41bd210ba17351399d3d2c211ecfbaadf1c3ae17
0.1s
 => => extracting sha256:cfb3c2203f88521b7100200c81b57aa1ec9cb9b1a1921ab3e5e39934ae619cf1
0.9s
 => => extracting sha256:9e592465b24339421141d5c8ebf8b8cf50ca92f94e3590ad3a616fbfa890e833
0.2s
 => => extracting sha256:8d4265d09d9cdac15612ec9d293497e286089ff0777daffa8b5638b734c7d316
0.0s
 => => extracting sha256:e3a8293e92fdc474ad5bc8500daa2a3208051960cbfaafd185919bef75a9e1d0
0.0s
 => => extracting sha256:2cb801c394368ce9eead7a09e97dcee1a81ad0848751fcd7ad9bd8287e7a0d47
7.9s
 => => extracting sha256:c5fdb20d8658f2e85ccc7008be082fa3e86f378f007cf387971486f816fb7743
0.0s
 => => extracting sha256:67c5fe618f0cf74ad1a0d31965795dc73bb1de11f7a9e8f26b4886e00e5868a5
0.0s
 => => extracting sha256:c9cdd1fe82e4c0cac2f8dfbd562ce2d9b79a03700c6bdee3bab7dcfd925b836b
0.0s
 => => extracting sha256:8f152c4aceed2dd762792866e573f6c078abb88b18778224e05b71f8d3bbac37
0.0s
 => => extracting sha256:2cd360f3b7db41d99ee40210a8b6801589ce86f26b0ecc063a7ba8416ffc2d7a
0.0s
 => [db 2/2] COPY init.sql /docker-entrypoint-initdb.d/
0.3s
 => [db] exporting to image
1.0s
 => => exporting layers
0.9s
 => => writing image sha256:65934f1f591c355706ede74cc076aef40b082a589ed99a34e2561cc111b7c33a
0.0s
 => => naming to docker.io/library/lab1-db
0.0s
 => [db] resolving provenance for metadata file
0.0s
 => [lab-service1 internal] load build definition from Dockerfile
0.5s
 => => transferring dockerfile: 227B
0.0s
 => [lab-service2 internal] load build definition from Dockerfile
0.5s
 => => transferring dockerfile: 227B
0.0s
 => [lab-service1 internal] load metadata for docker.io/library/python:3.12-slim
2.1s
 => [lab-service2 internal] load .dockerignore
0.6s
 => => transferring context: 2B
0.0s
 => [lab-service1 internal] load .dockerignore
0.7s
 => => transferring context: 2B
0.0s
 => [lab-service2 internal] load build context
0.6s
 => => transferring context: 1.19kB
0.0s
 => [lab-service2 1/5] FROM docker.io/library/python:3.12-slim@sha256:60d9996b6a8a3689d36db740b49f4327be3be09a21122bd02fb8895abb38b50d
5.6s
 => => resolve docker.io/library/python:3.12-slim@sha256:60d9996b6a8a3689d36db740b49f4327be3be09a21122bd02fb8895abb38b50d
0.4s
 => => sha256:60d9996b6a8a3689d36db740b49f4327be3be09a21122bd02fb8895abb38b50d 9.12kB / 9.12kB
0.0s
 => => sha256:1c44018d7eb40488f29e7c6ad4991d3200507e14dca71b94fe61011815e98155 1.75kB / 1.75kB
0.0s
 => => sha256:e35f80b558775b0903c34523ef3422645c10e06a66b151f228acf9cf8b862f66 5.17kB / 5.17kB
0.0s
 => => sha256:5f86e0092844f78afa84a3b63059ea82e82fa93a28eedde55386370d924e8cf3 13.63MB / 13.63MB
1.9s
 => => sha256:5e739003c33414360cf2cec9452e1935b1bfcae79f279c33009bf4c0d5596f52 3.32MB / 3.32MB
1.3s
 => => sha256:46271edb89aec00565d974118707aaded82928c7ea7cb0a2bcc94b1f17e17e3d 249B / 249B
0.8s
 => => extracting sha256:5e739003c33414360cf2cec9452e1935b1bfcae79f279c33009bf4c0d5596f52
0.4s
 => => extracting sha256:5f86e0092844f78afa84a3b63059ea82e82fa93a28eedde55386370d924e8cf3
2.0s
 => => extracting sha256:46271edb89aec00565d974118707aaded82928c7ea7cb0a2bcc94b1f17e17e3d
0.0s
 => [lab-service1 internal] load build context
0.7s
 => => transferring context: 1.19kB
0.0s
 => [lab-service2 2/5] WORKDIR /app
0.2s
 => [lab-service2 3/5] COPY requirements.txt /app/
0.3s
 => [lab-service2 4/5] RUN pip install -r requirements.txt
11.3s
 => [lab-service2 5/5] COPY main.py /app
0.3s
 => [lab-service2] exporting to image
1.3s
 => => exporting layers
1.2s
 => => writing image sha256:307079b74c26af4a96b354abfe6ea77f8f7fb3a1950653ee2774c915448ce78c
0.0s
 => => naming to docker.io/library/lab1-lab-service2
0.0s
 => [lab-service1] exporting to image
1.3s
 => => exporting layers
1.2s
 => => writing image sha256:5dd5a3798f98d5c4cfcac4f955f36cb3415b22058d0b4845b097fd0b5efdb35a
0.0s
 => => naming to docker.io/library/lab1-lab-service1
0.0s
 => [lab-service1] resolving provenance for metadata file
0.2s
 => [lab-service2] resolving provenance for metadata file
0.0s
 => [nginx internal] load build definition from Dockerfile
0.2s
 => => transferring dockerfile: 94B
0.0s
 => [nginx internal] load metadata for docker.io/library/nginx:1.27.3
2.2s
 => [nginx internal] load .dockerignore
0.3s
 => => transferring context: 2B
0.0s
 => [nginx internal] load build context
0.3s
 => => transferring context: 276B
0.0s
 => [nginx 1/2] FROM docker.io/library/nginx:1.27.3@sha256:fb197595ebe76b9c0c14ab68159fd3c08bd067ec62300583543f0ebda353b5be
10.4s
 => => resolve docker.io/library/nginx:1.27.3@sha256:fb197595ebe76b9c0c14ab68159fd3c08bd067ec62300583543f0ebda353b5be
0.4s
 => => sha256:fb197595ebe76b9c0c14ab68159fd3c08bd067ec62300583543f0ebda353b5be 10.27kB / 10.27kB
0.0s
 => => sha256:3d696e8357051647b844d8c7cf4a0aa71e84379999a4f6af9b8ca1f7919ade42 2.29kB / 2.29kB
0.0s
 => => sha256:66f8bdd3810c96dc5c28aec39583af731b34a2cd99471530f53c8794ed5b423e 8.58kB / 8.58kB
0.0s
 => => sha256:8cc1569e58f52d008e232130d8fca2411f417ea423305cd7d7b513fb96d22947 629B / 629B
0.2s
 => => sha256:650ee30bbe5efddbef9cc0245ba52b133d3c8897a6565faa6c5c87bc552b5305 43.84MB / 43.84MB
4.3s
 => => sha256:362f35df001b4bd6f8733cd4abe8e1493582782404fefc2393129a5dfb5e72df 955B / 955B
0.4s
 => => extracting sha256:362f35df001b4bd6f8733cd4abe8e1493582782404fefc2393129a5dfb5e72df
0.0s
 => => extracting sha256:13e320bf29cd3ef51b06a3dfe259b2582d48be27a9ac4c6b7af6fbb99429d210
0.0s
[+] Building 4/4b64962dd94d4818372adf30dc0e2ca4803c46d4f638b7712fe01a149c705c5 1.40kB / 1.40kB
0.9s
```

```
 => => extracting sha256:7b50399908e1c0958c409f3c844d61736fd41e37a58dca4832927715508dd3aa
0.0s
 => => extracting sha256:57b64962dd94d4818372adf30dc0e2ca4803c46d4f638b7712fe01a149c705c5
0.0s
 => [nginx 2/2] COPY nginx.conf /etc/nginx/nginx.conf
0.2s
 ✔ Service db             Built
30.4s
 => [nginx] exporting to image
0.4s
 => => exporting layers
0.4s
 ✔ Service lab-service2  Built
23.4s
[+] Building 4/4mage sha256:35c5e9ecab0287faadeb31319f37a908274e3ca804f9161a8e93571941aca7e0
0.0s
 ✔ Service db             Built
30.4s
 ✔ Service lab-service2  Built
23.4s
 ✔ Service lab-service1  Built
23.3s
 ✔ Service nginx          Built
14.3s
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$
```

Запустимо рішення командою docker compose up:

```
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ docker compose up
[+] Running 5/5
 ✔ Network lab1_default    Created                                                    0.1s
 ✔ Container db            Created                                                    0.1s
 ✔ Container lab-service2  Created                                                    0.2s
 ✔ Container lab-service1  Created                                                    0.2s
 ✔ Container nginx         Created                                                    0.1s
Attaching to db, lab-service1, lab-service2, nginx
db            | The files belonging to this database system will be owned by user "postgres".
db            | This user must also own the server process.
db            |
db            | The database cluster will be initialized with locale "en_US.utf8".
db            | The default database encoding has accordingly been set to "UTF8".
db            | The default text search configuration will be set to "english".
db            |
db            | Data page checksums are disabled.
db            |
db            | fixing permissions on existing directory /var/lib/postgresql/data ... ok
db            | creating subdirectories ... ok
db            | selecting dynamic shared memory implementation ... posix
db            | selecting default "max_connections" ... 100
db            | selecting default "shared_buffers" ... 128MB
db            | selecting default time zone ... Etc/UTC
db            | creating configuration files ... ok
db            | running bootstrap script ... ok
db            | performing post-bootstrap initialization ... ok
db            | syncing data to disk ... ok
db            |
db            |
db            | Success. You can now start the database server using:
db            |
db            |     pg_ctl -D /var/lib/postgresql/data -l logfile start
db            |
db            | initdb: warning: enabling "trust" authentication for local connections
db            | initdb: hint: 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.
db            | waiting for server to start....2024-12-04 14:38:31.468 UTC [55] LOG:  starting PostgreSQL 17.2 (Debian 17.2-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
db            | 2024-12-04 14:38:31.475 UTC [55] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
db            | 2024-12-04 14:38:31.510 UTC [58] LOG:  database system was shut down at 2024-12-04 14:38:23 UTC
db            | 2024-12-04 14:38:31.524 UTC [55] LOG:  database system is ready to accept connections
db            |  done
db            | server started
db            | CREATE DATABASE
db            |
db            |
db            | /usr/local/bin/docker-entrypoint.sh: running /docker-entrypoint-initdb.d/init.sql
db            | CREATE TABLE
db            |
db            |
db            | waiting for server to shut down....2024-12-04 14:38:31.888 UTC [55] LOG:  received fast shutdown request
db            | 2024-12-04 14:38:31.895 UTC [55] LOG:  aborting any active transactions
db            | 2024-12-04 14:38:31.898 UTC [55] LOG:  background worker "logical replication launcher" (PID 61) exited with exit code 1
db            | 2024-12-04 14:38:31.901 UTC [56] LOG:  shutting down
db            | 2024-12-04 14:38:31.911 UTC [56] LOG:  checkpoint starting: shutdown immediate
```

Повний вивід команди:

```
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ docker compose up
[+] Running 5/5
 ✔ Network lab1_default    Created
0.1s
 ✔ Container db            Created
0.1s
 ✔ Container lab-service2  Created
0.2s
 ✔ Container lab-service1  Created
0.2s
 ✔ Container nginx         Created
0.1s
Attaching to db, lab-service1, lab-service2, nginx
db            | The files belonging to this database system will be owned by user "postgres".
db            | This user must also own the server process.
db            |
db            | The database cluster will be initialized with locale "en_US.utf8".
db            | The default database encoding has accordingly been set to "UTF8".
db            | The default text search configuration will be set to "english".
db            |
db            | Data page checksums are disabled.
db            |
db            | fixing permissions on existing directory /var/lib/postgresql/data ... ok
db            | creating subdirectories ... ok
db            | selecting dynamic shared memory implementation ... posix
db            | selecting default "max_connections" ... 100
db            | selecting default "shared_buffers" ... 128MB
db            | selecting default time zone ... Etc/UTC
db            | creating configuration files ... ok
db            | running bootstrap script ... ok
db            | performing post-bootstrap initialization ... ok
db            | syncing data to disk ... ok
db            |
db            |
db            | Success. You can now start the database server using:
db            |
db            |     pg_ctl -D /var/lib/postgresql/data -l logfile start
db            |
db            | initdb: warning: enabling "trust" authentication for local connections
db            | initdb: hint: 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.
db            | waiting for server to start....2024-12-04 14:38:31.468 UTC [55] LOG:  starting PostgreSQL 17.2 (Debian 17.2-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
db            | 2024-12-04 14:38:31.475 UTC [55] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
db            | 2024-12-04 14:38:31.510 UTC [58] LOG:  database system was shut down at 2024-12-04 14:38:23 UTC
db            | 2024-12-04 14:38:31.524 UTC [55] LOG:  database system is ready to accept connections
db            |  done
db            | server started
db            | CREATE DATABASE
db            |
db            |
db            | /usr/local/bin/docker-entrypoint.sh: running /docker-entrypoint-initdb.d/init.sql
db            | CREATE TABLE
db            |
db            |
db            | waiting for server to shut down....2024-12-04 14:38:31.888 UTC [55] LOG:  received fast shutdown request
db            | 2024-12-04 14:38:31.895 UTC [55] LOG:  aborting any active transactions
db            | 2024-12-04 14:38:31.898 UTC [55] LOG:  background worker "logical replication launcher" (PID 61) exited with exit code 1
db            | 2024-12-04 14:38:31.901 UTC [56] LOG:  shutting down
db            | 2024-12-04 14:38:31.911 UTC [56] LOG:  checkpoint starting: shutdown immediate
db            | .2024-12-04 14:38:33.147 UTC [77] FATAL:  the database system is shutting down
db            | .2024-12-04 14:38:34.326 UTC [56] LOG:  checkpoint complete: wrote 929 buffers (5.7%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.091 s, sync=2.270 s, total=2.425 s; sync files=304,
longest=0.017 s, average=0.008 s; distance=4256 kB, estimate=4256 kB; lsn=0/190D158, redo lsn=0/190D158
db            | 2024-12-04 14:38:34.335 UTC [55] LOG:  database system is shut down
db            |  done
db            | server stopped
```

```
db            |
db            | PostgreSQL init process complete; ready for start up.
db            |
db            | 2024-12-04 14:38:34.465 UTC [1] LOG:  starting PostgreSQL 17.2 (Debian 17.2-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
db            | 2024-12-04 14:38:34.465 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
db            | 2024-12-04 14:38:34.465 UTC [1] LOG:  listening on IPv6 address "::", port 5432
db            | 2024-12-04 14:38:34.480 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
db            | 2024-12-04 14:38:34.501 UTC [81] LOG:  database system was shut down at 2024-12-04 14:38:34 UTC
db            | 2024-12-04 14:38:34.527 UTC [1] LOG:  database system is ready to accept connections
nginx         | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
nginx         | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
nginx         | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
nginx         | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
nginx         | 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
nginx         | /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
nginx         | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
nginx         | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
nginx         | /docker-entrypoint.sh: Configuration complete; ready for start up
lab-service1  | INFO:     Started server process [1]
lab-service1  | INFO:     Waiting for application startup.
lab-service1  | INFO:     Application startup complete.
lab-service1  | INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
lab-service2  | INFO:     Started server process [1]
lab-service2  | INFO:     Waiting for application startup.
lab-service2  | INFO:     Application startup complete.
lab-service2  | INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```

Окремо перевіримо, чи активні контейнери рішення:

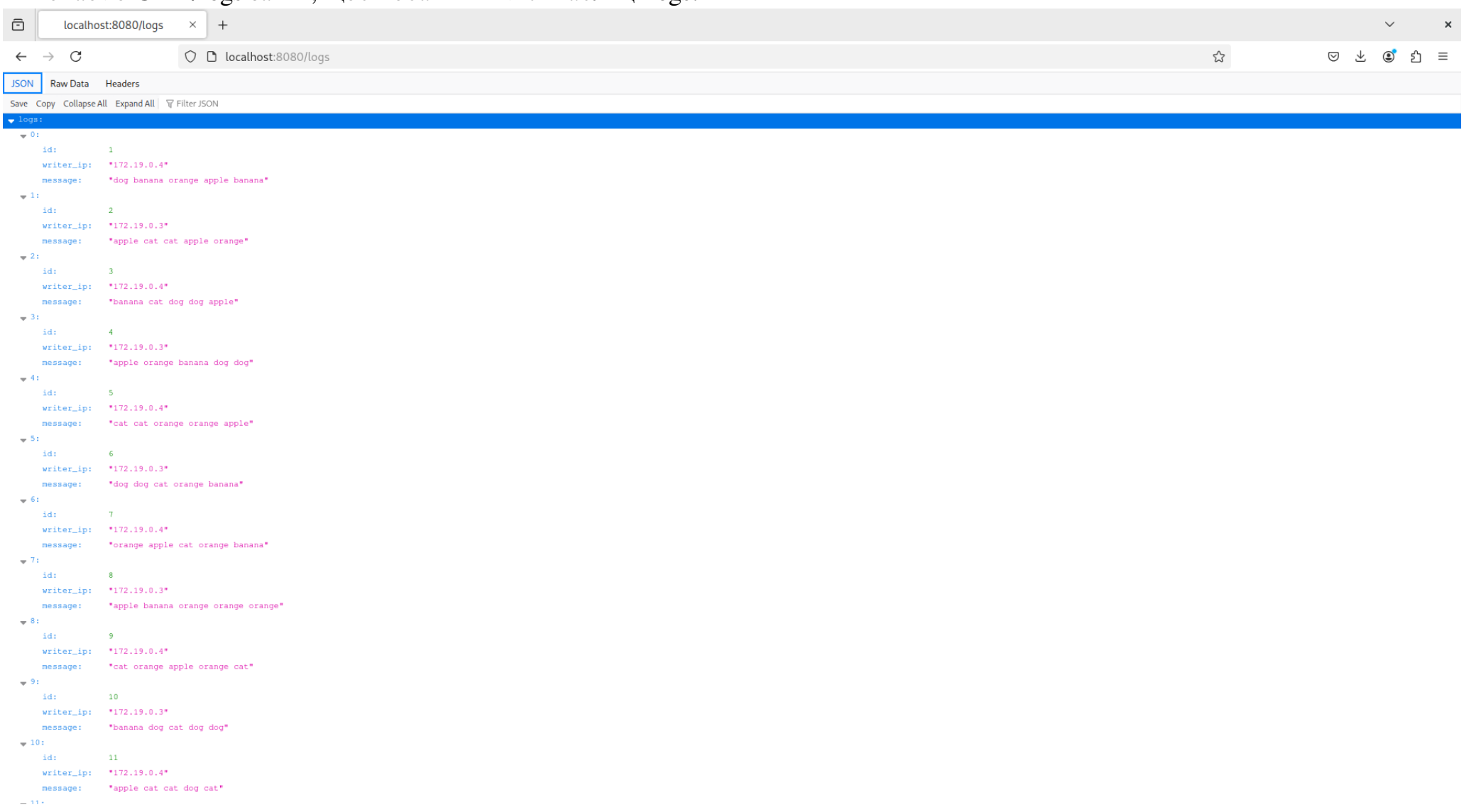
```
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED              STATUS                        PORTS                                                  NAMES
097137e46582   lab1-nginx          "/docker-entrypoint.…"   About a minute ago   Up About a minute             80/tcp, 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp      nginx
685be585bfea   lab1-lab-service2   "uvicorn main:app --…"   About a minute ago   Up About a minute                                                                    lab-service2
f01d415f2134   lab1-lab-service1   "uvicorn main:app --…"   About a minute ago   Up About a minute                                                                    lab-service1
40e44b093e5c   lab1-db             "docker-entrypoint.s…"   About a minute ago   Up About a minute (healthy)   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp              db
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$
```

Виконаємо декілька POST /action запитів:

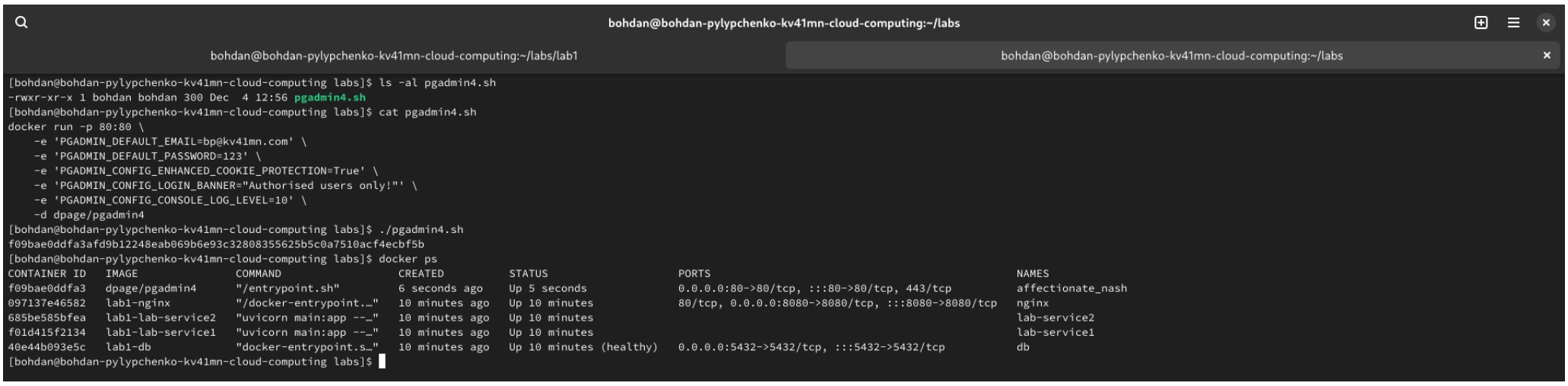
```
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.4","message":"dog banana orange apple banana"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.3","message":"apple cat cat apple orange"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.4","message":"banana cat dog dog apple"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.3","message":"apple orange banana dog dog"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.4","message":"cat cat orange orange apple"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.3","message":"dog dog cat orange banana"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.4","message":"orange apple cat orange banana"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.3","message":"apple banana orange orange orange"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.4","message":"cat orange apple orange cat"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.3","message":"banana dog cat dog dog"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.4","message":"apple cat cat dog cat"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.3","message":"orange apple apple banana cat"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.4","message":"cat dog apple apple banana"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.3","message":"dog orange apple banana cat"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.4","message":"apple apple orange orange dog"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.3","message":"banana orange cat cat cat"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.4","message":"cat orange orange dog orange"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.3","message":"cat dog dog apple dog"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.4","message":"apple banana dog banana banana"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.3","message":"cat dog cat banana dog"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$ curl -X POST http://localhost:8080/action
{"writer_ip":"172.19.0.4","message":"apple dog banana dog apple"}[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing lab1]$
```

Бачимо, що writer_ip відрізняється від запиту до запиту: nginx дійсно балансує навантаження між 2 інстансами.

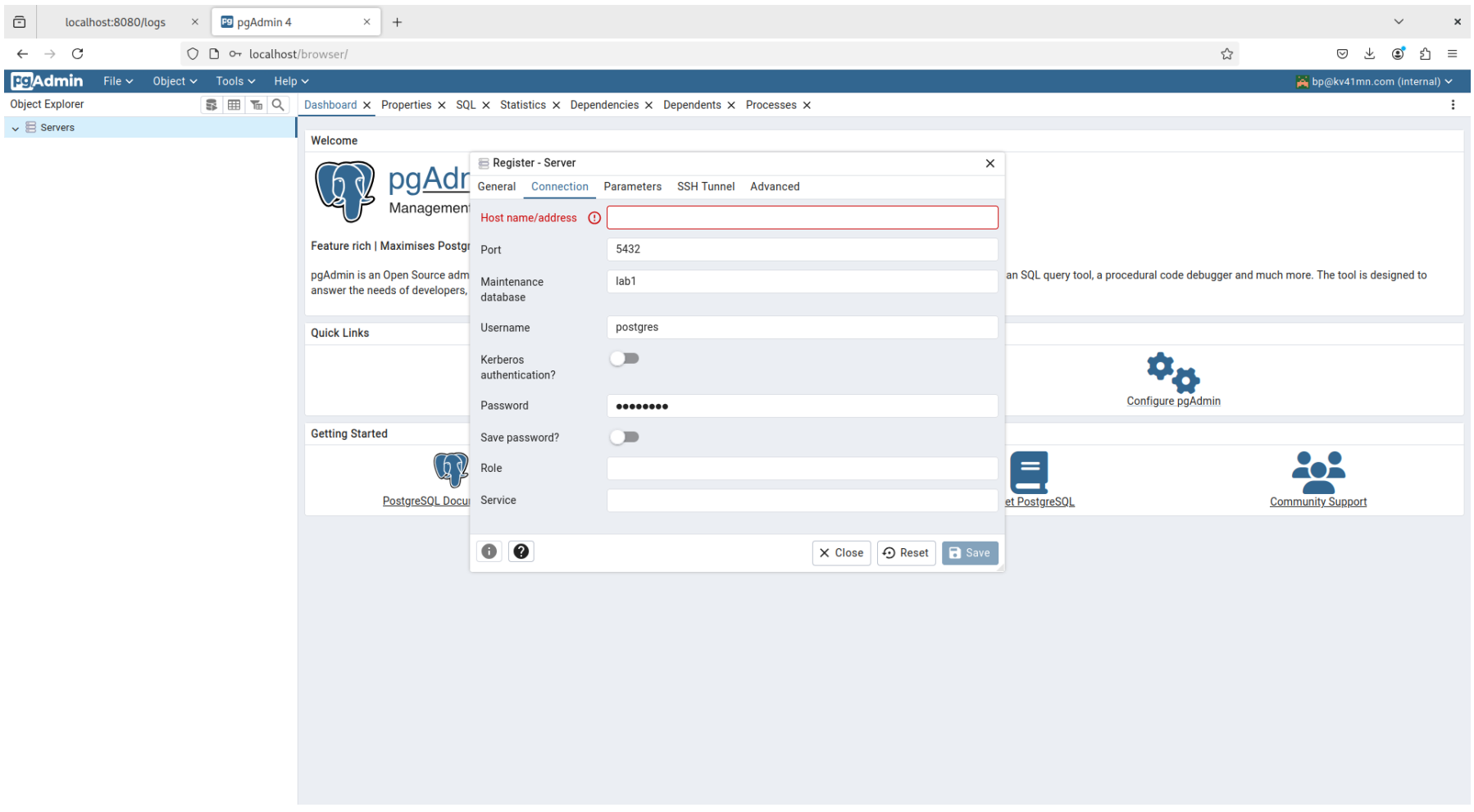
Виконаємо GET /logs запит, щоб побачити вміст таблиці logs:



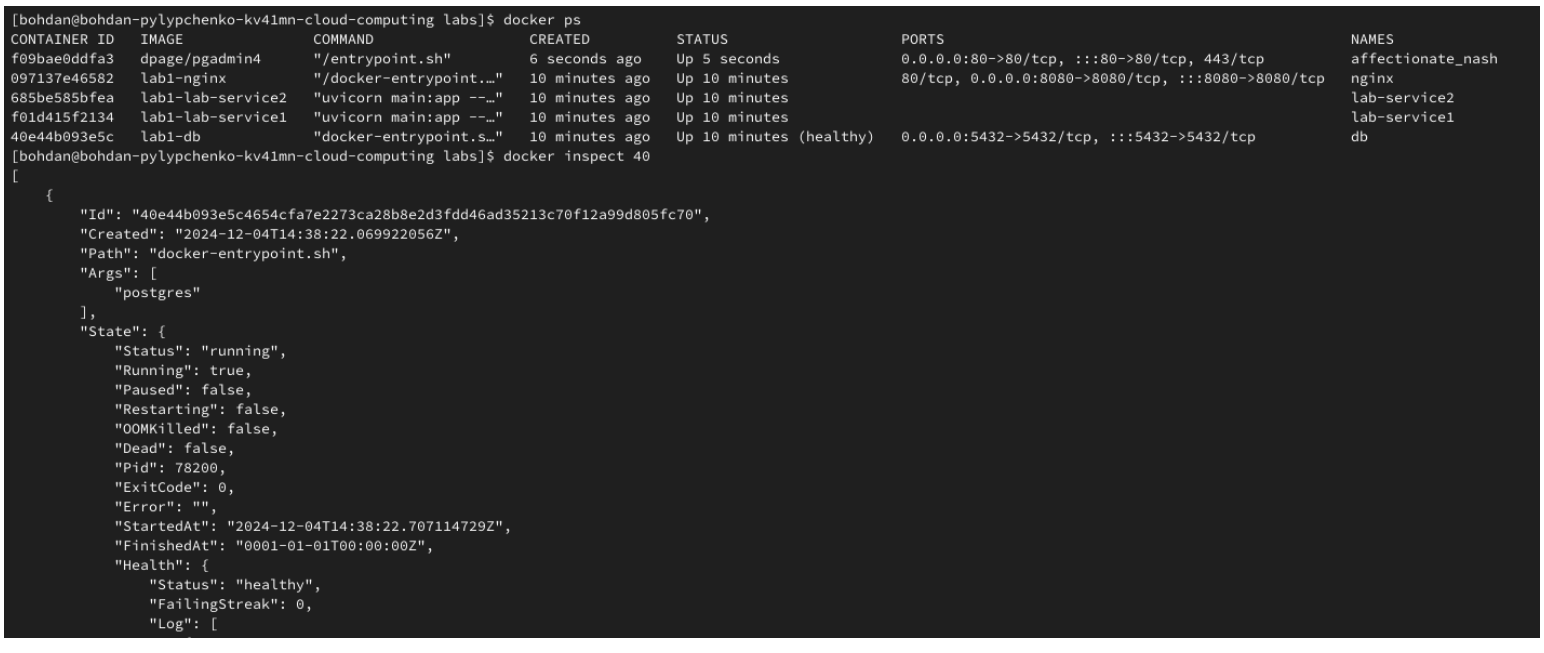
Повідомлення збереглись у базі даних.
Додатково перевіримо записи у базі даних за допомогою pgadmin4.
Для цього запустимо pgadmin за допомогою скрипта pgadmin4.sh:



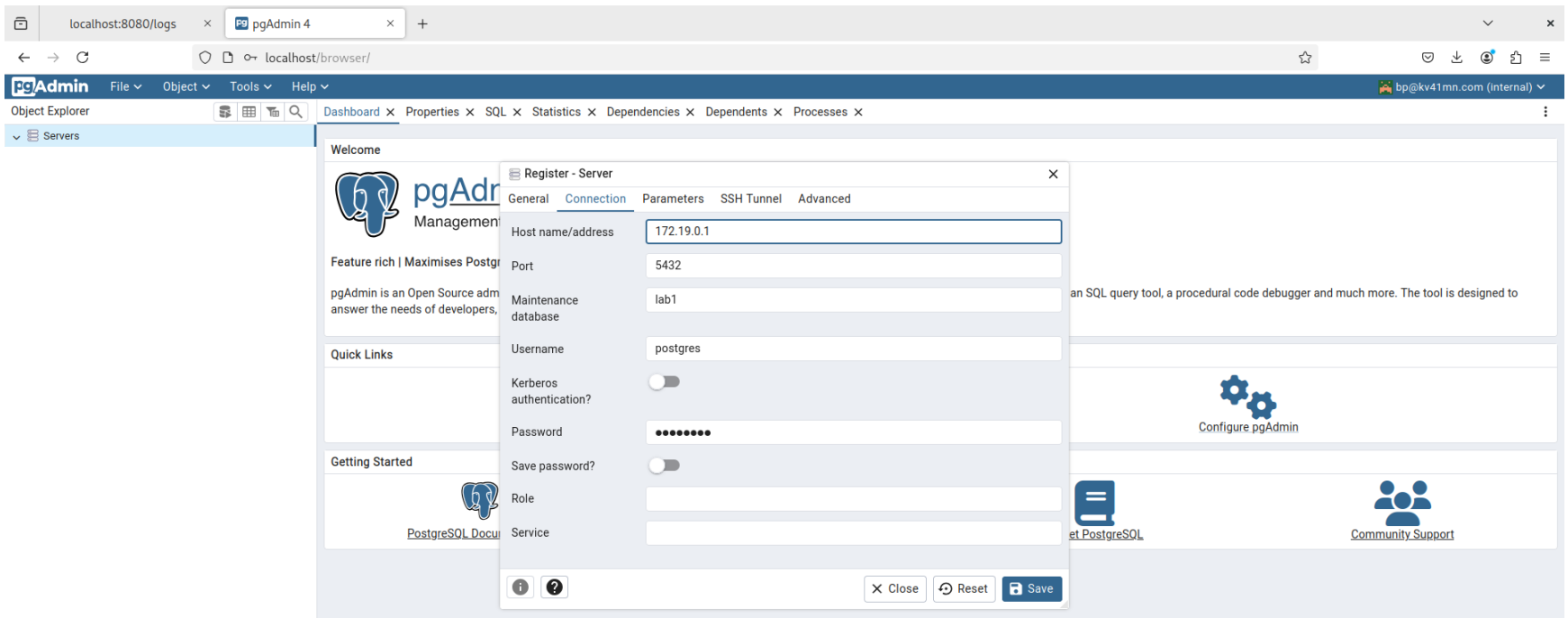
Для з’єднання pgadmin із сервером лабораторної роботи маємо вказати ip серверу із docker network:



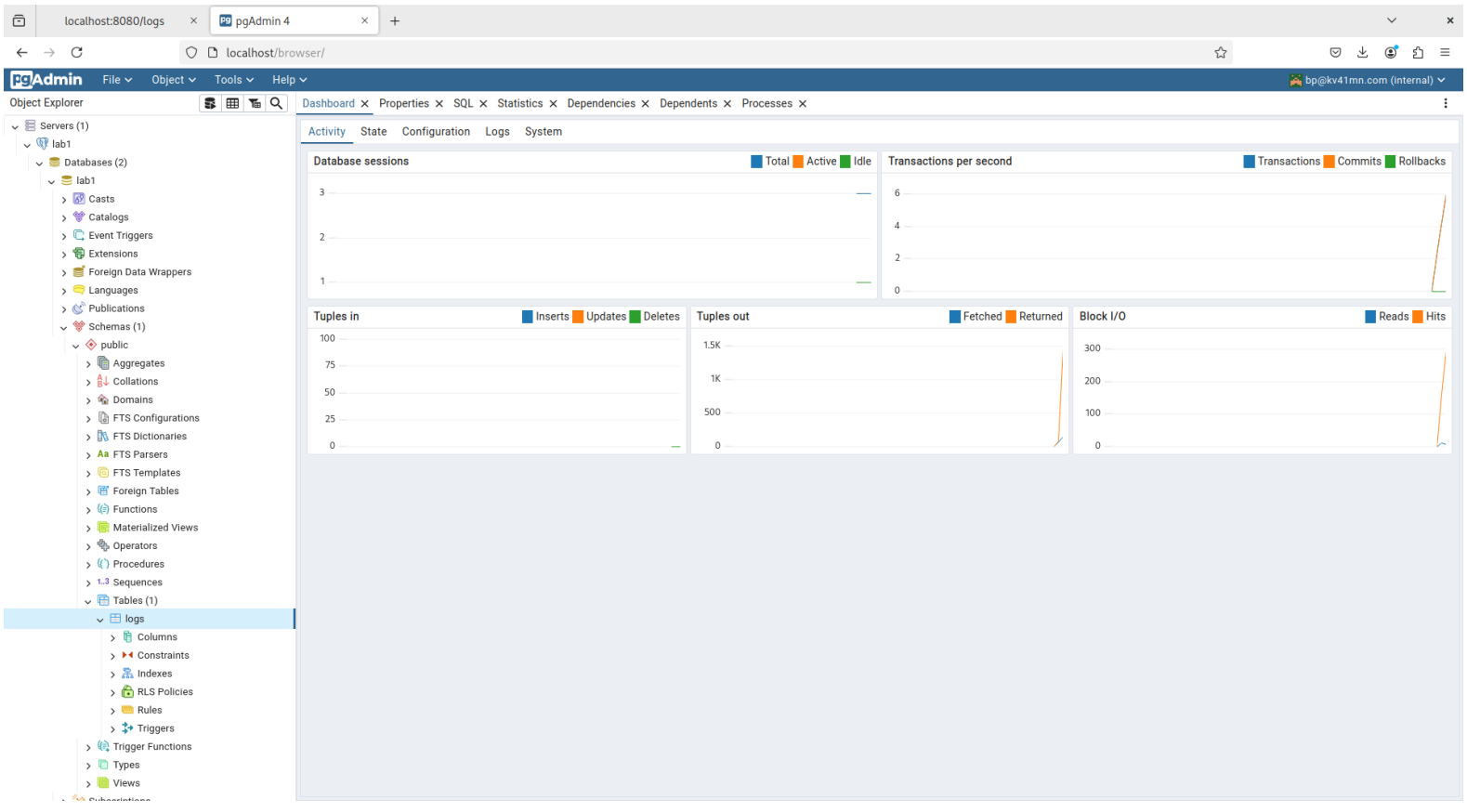
Отримаємо docker network ip серверу роботи за допомогою docker inspect:



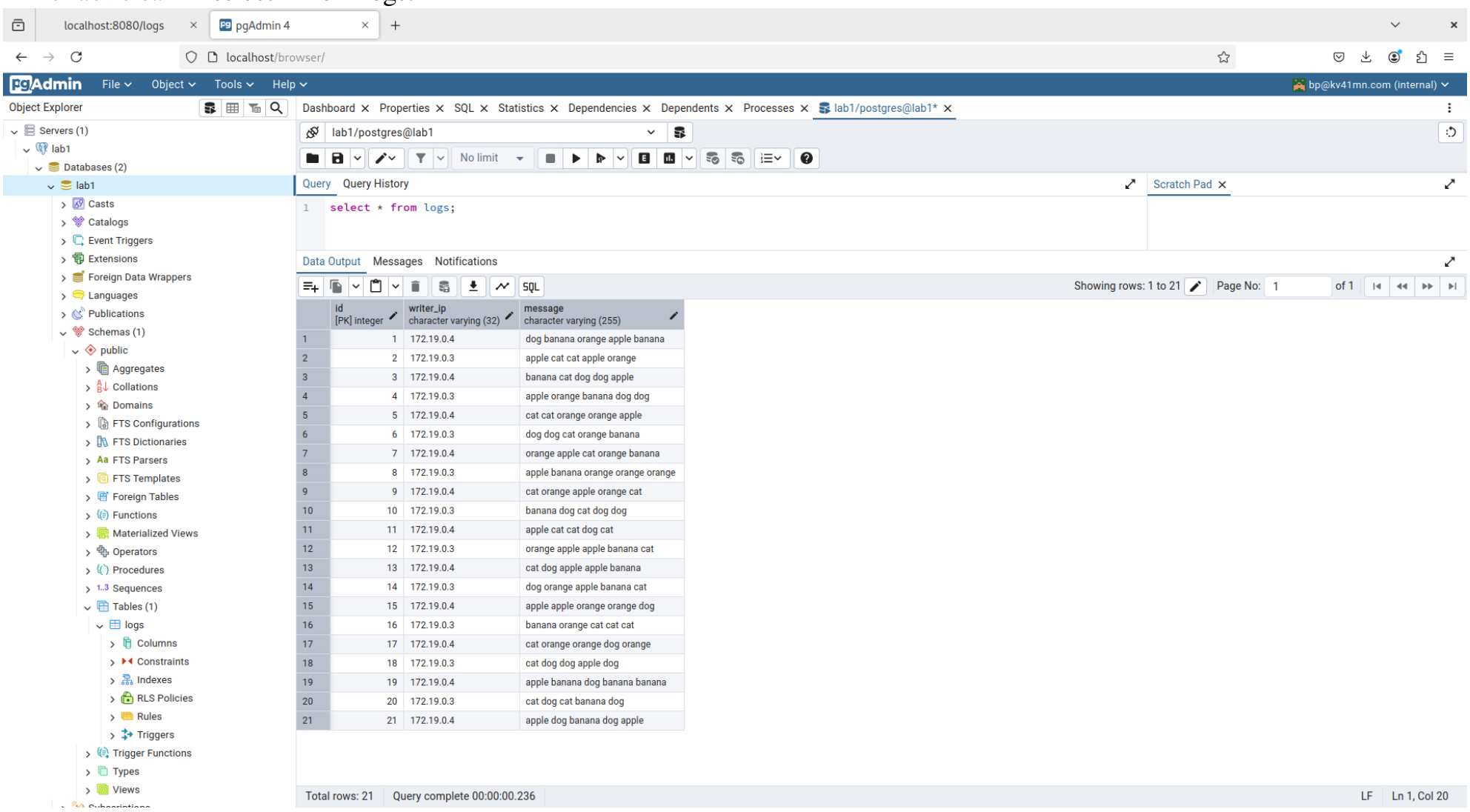
```
            "Networks": {
                "lab1_default": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": [
                        "db",
                        "db"
                    ],
                    "MacAddress": "02:42:ac:13:00:02",
                    "DriverOpts": null,
                    "NetworkID": "a718e02e233907b5b70b0576082d10d896558bf7971eb17e071c1717d346f913",
                    "EndpointID": "282b9cb94c35edbe1ef4b2b1bf02f8c39a2bb265ac9fb9cb48fe4da0cf2ad710",
                    "Gateway": "172.19.0.1",
                    "IPAddress": "172.19.0.2",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "DNSNames": [
                        "db",
                        "40e44b093e5c"
                    ]
                }
            }
        }
    }
]
[bohdan@bohdan-pylypchenko-kv41mn-cloud-computing labs]$
```

З’єднання встановлено:



Виконаємо запит select * from logs:



Додатки lab-service1/2 дійсно виконували записи / читання бази даних.