

Контейнеризация урок 5. ДЗ.

Виктор Владыкин

Задание 1.

Запускается два контейнера *postgres* и *adminer*. Переменные окружения для БД берутся из файла *.env*. Для сохранения данных из БД пробрасывается директория. Данные будут доступны на хостовой машине по пути */var/lib/docker/volumes/s5hw_db_data/_data*.

файл *.env*

```
POSTGRES_USER="db_user"
POSTGRES_PASSWORD="pass you'll never guess"
POSTGRES_DB="app_db"
```

файл *docker-compose.yml*

```
version: "3.9"

services:

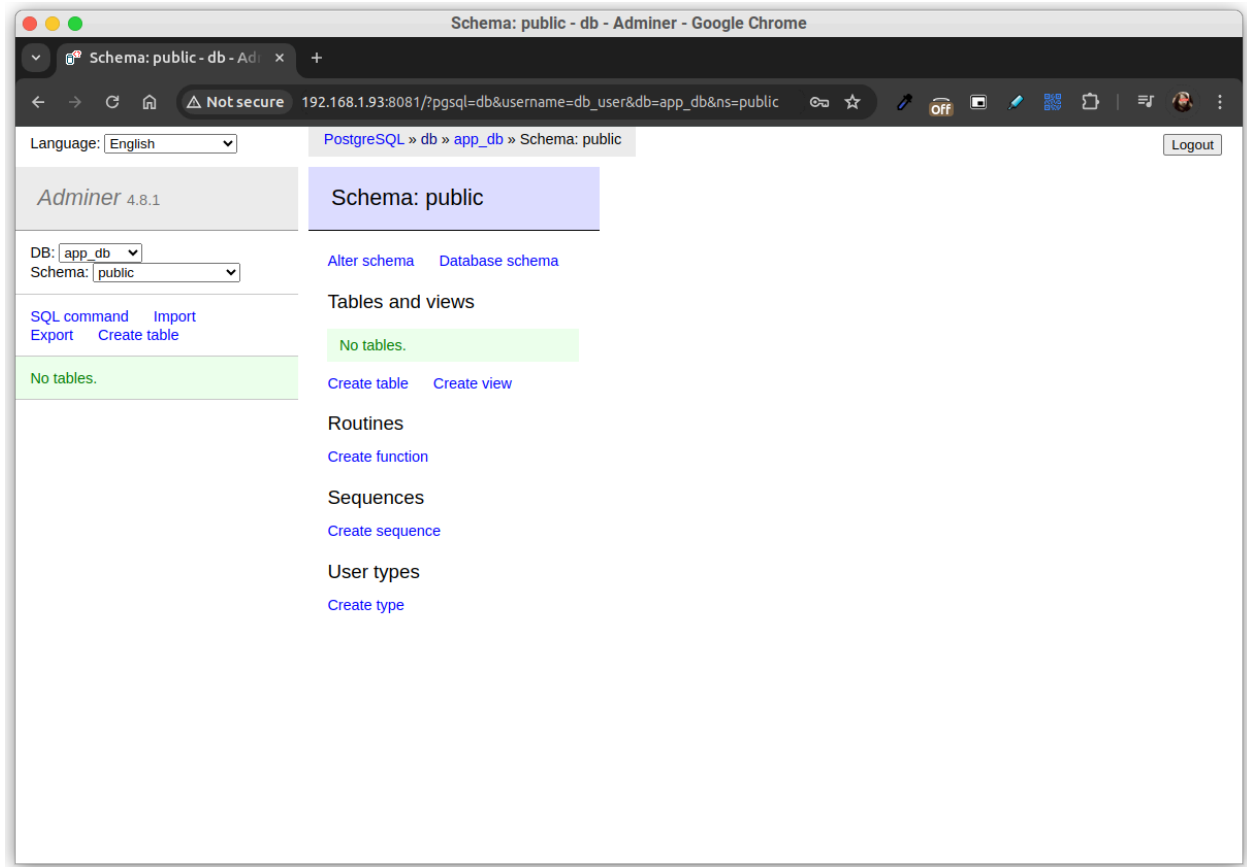
  db:
    image: postgres:16.2
    restart: unless-stopped
    volumes:
      - db_data:/var/lib/postgresql/data
    environment:
      POSTGRES_USER: ${POSTGRES_USER}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
      POSTGRES_DB: ${POSTGRES_DB}

  adminer:
    image: adminer:4.8.1
    restart: unless-stopped
    depends_on:
      - db
    ports:
      - 8081:8080

volumes:
  db_data: {}
```

```
zombiehunter@us: ~/s5hw
zombiehunter@us:~/s5hw$ micro .env
zombiehunter@us:~/s5hw$ micro docker-compose.yml
zombiehunter@us:~/s5hw$ docker-compose up -d
Creating network "s5hw_default" with the default driver
Creating s5hw_db_1 ... done
Creating s5hw_adminer_1 ... done
zombiehunter@us:~/s5hw$ docker-compose ps
      Name                    Command                                State      Ports
-----
s5hw_adminer_1    entrypoint.sh php -S [::]: ...    Up        0.0.0.0:8081->8080/tcp,:::8081->8080/tcp
s5hw_db_1         docker-entrypoint.sh postgres     Up        5432/tcp
zombiehunter@us:~/s5hw$
```

```
zombiehunter@us: ~/s5hw
zombiehunter@us:~/s5hw$ sudo ls /var/lib/docker/volumes/s5hw_db_data/_data
base          pg_hba.conf  pg_notify    pg_stat      pg_twophase  postgresql.auto.conf
global        pg_ident.conf pg_replslot  pg_stat_tmp  PG_VERSION   postgresql.conf
pg_commit_ts  pg_logical   pg_serial    pg_subtrans  pg_wal       postmaster.opts
pg_dynshmem   pg_multixact pg_snapshots pg_tblspc    pg_xact      postmaster.pid
zombiehunter@us:~/s5hw$
```



Задание 2.

В наличии 4 ноды: *us*, *us2*, *us3*, *us4* (*us* - менеджер, остальные воркеры). Далее размечаю ноды при помощи лейблов. Две ноды на *lab*, две на *dev*.

```
docker node update --label-add TAG=lab us
docker node inspect us | grep Spec -A 5
```

```
"Spec": {
  "Labels": {
    "TAG": "lab"
  },
  "Role": "manager",
  "Availability": "active"
```

```
docker node update --label-add TAG=lab us2
docker node update --label-add TAG=dev us3
docker node update --label-add TAG=dev us4
```

Создаю два docker compose файла, сервисы разношу по портам, в секции *deploy* определяю ноды для сервисов.

файл docker-compose.lab.yaml

```
version: "3.9"

services:
  adminer:
    image: adminer:4.8.1
    restart: unless-stopped
    networks:
      - app_lab_network
    ports:
      - "8087:8080"
    deploy:
      placement:
        constraints:
          - "node.labels.TAG==lab"

  db:
    image: postgres:16.2
    restart: unless-stopped
    volumes:
      - db_lab_data:/var/lib/postgresql/data
    environment:
      POSTGRES_USER: db_user
      POSTGRES_PASSWORD: pass
      POSTGRES_DB: app_db
    networks:
      - app_lab_network
    ports:
      - "5432:5432"
    deploy:
      placement:
        constraints:
          - "node.labels.TAG==lab"

volumes:
  db_lab_data: {}

networks:
  app_lab_network:
    driver: overlay
```

файл docker-compose.dev.yaml

```
version: "3.9"

services:
  adminer:
    image: adminer:4.8.1
    restart: unless-stopped
    networks:
      - app_dev_network
    ports:
      - "8089:8080"
    deploy:
      placement:
        constraints:
          - "node.labels.TAG==dev"

  db:
    image: postgres:16.2
    restart: unless-stopped
    volumes:
      - db_dev_data:/var/lib/postgresql/data
    environment:
      POSTGRES_USER: db_user
      POSTGRES_PASSWORD: pass
      POSTGRES_DB: app_db
    networks:
      - app_dev_network
    ports:
      - "5433:5432"
    deploy:
      placement:
        constraints:
          - "node.labels.TAG==dev"

volumes:
  db_dev_data: {}

networks:
  app_dev_network:
    driver: overlay
```

Запущь!

```
docker stack deploy -c docker-compose.dev.yaml stack_dev
docker stack deploy -c docker-compose.lab.yaml stack_lab
```

```
zombiehunter@us:~/s5hw2$ docker swarm init
Swarm initialized: current node (wep9llrxyxsm16ec8jap4b0v5) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-49vy6phlaih1qfhl8m932zlwantnodwijn7fnk0s648bm7rw3-56q0g6jsrlxp93h11ndgjqr0q 192.168.1.93:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

zombiehunter@us:~/s5hw2$ docker node ls
ID                HOSTNAME        STATUS    AVAILABILITY    MANAGER STATUS    ENGINE VERSION
wep9llrxyxsm16ec8jap4b0v5 *   us             Ready     Active           Leader             26.1.0
zombiehunter@us:~/s5hw2$ docker node ls
ID                HOSTNAME        STATUS    AVAILABILITY    MANAGER STATUS    ENGINE VERSION
wep9llrxyxsm16ec8jap4b0v5 *   us             Ready     Active           Leader             26.1.0
j9tbyxhcfjbtwui46bnuktx5     us2            Ready     Active           26.0.2
knrc6jcb8s1i7og44wbjovzn      us3            Ready     Active           26.1.0
pqkgdbpk1knzqxmcabo497ebp     us4            Ready     Active           26.1.0
zombiehunter@us:~/s5hw2$

zombiehunter@us2:~$ clear
zombiehunter@us2:~$ docker swarm join --token SWMTKN-1-49vy6phlaih1qfhl8m932zlwantnodwijn7fnk0s648bm7rw3-56q0g6jsrlxp93h11ndgjqr0q 192.168.1.93:2377
This node joined a swarm as a worker.
zombiehunter@us2:~$

zombiehunter@us3:~$ clear
zombiehunter@us3:~$ docker swarm join --token SWMTKN-1-49vy6phlaih1qfhl8m932zlwantnodwijn7fnk0s648bm7rw3-56q0g6jsrlxp93h11ndgjqr0q 192.168.1.93:2377
This node joined a swarm as a worker.
zombiehunter@us3:~$

This node joined a swarm as a worker.
zombiehunter@us4:~$

[tmx0] 0:ssh+ "zhm-msi" 18:45 26-Apr-24
```

```
Creating service stack_dev.db
zombiehunter@us:~/s5hw2$ docker stack deploy -c docker-compose.lab.yaml stack_lab
Ignoring unsupported options: restart

Since --detach=false was not specified, tasks will be created in the background.
In a future release, --detach=false will become the default.
Creating network stack_lab_app_lab_network
Creating service stack_lab.db
Creating service stack_lab.adminer
zombiehunter@us:~/s5hw2$ docker stack services stack_lab
ID                NAME                MODE                REPLICAS    IMAGE                PORTS
nmvt8dniac5      stack_lab.adminer    replicated          1/1          adminer:4.8.1       *:8087->8080/tcp
9q0um9me2vj0     stack_lab.db         replicated          1/1          postgres:16.2       *:5432->5432/tcp
zombiehunter@us:~/s5hw2$ docker stack services stack_dev
ID                NAME                MODE                REPLICAS    IMAGE                PORTS
bznhkuc4ma7d     stack_dev.adminer    replicated          1/1          adminer:4.8.1       *:8089->8080/tcp
zicteibrazds     stack_dev.db         replicated          1/1          postgres:16.2       *:5433->5432/tcp
zombiehunter@us:~/s5hw2$ docker ps
CONTAINER ID      IMAGE                COMMAND             CREATED         STATUS         PORTS             NAMES
e6711e83779      adminer:4.8.1       "entrypoint.sh php -" 43 seconds ago  Up 42 seconds  8080/tcp          stack_lab.adminer.1.ebjoe4s6df9obqef8wrmrdxk3
zombiehunter@us:~/s5hw2$

zombiehunter@us2:~$ docker ps
CONTAINER ID      IMAGE                COMMAND             CREATED         STATUS         PORTS             NAMES
163918f4b5a1     postgres:16.2       "docker-entrypoint.s" 29 seconds ago  Up 28 seconds  5432/tcp          stack_lab.db.1.418f2sxam785ukw1j6q4a5dmk
zombiehunter@us2:~$

zombiehunter@us3:~$ docker ps
CONTAINER ID      IMAGE                COMMAND             CREATED         STATUS         PORTS             NAMES
09685fa6710a     adminer:4.8.1       "entrypoint.sh php -" 46 seconds ago  Up 44 seconds  8080/tcp          stack_dev.adminer.1.8ua2bx8nhh8m3h5w55bs1hw07
zombiehunter@us3:~$

zombiehunter@us4:~$ docker ps
CONTAINER ID      IMAGE                COMMAND             CREATED         STATUS         PORTS             NAMES
9b24dec450ac     postgres:16.2       "docker-entrypoint.s" 49 seconds ago  Up 48 seconds  5432/tcp          stack_dev.db.1.qojd1rp4wgh1witngq9k56enk
zombiehunter@us4:~$

[tmx0] 0:ssh+ "zhm-msi" 21:55 26-Apr-24
```

