Задача 2.5: Создание многоконтейнерного приложения с помощью Docker Compose

Подготовка

Образ для Nginx

1. В директории приложения создадим папку для nginx:
   `mkdir nginx`
2. Внутри папки создадим файл index.html для нашей будущей home страницы и отредактируем:
   `nano index.html`

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Docker Nginx</title>
</head>
<body>
  <h2>ITransition, hello from Nginx container!</h2>
</body>
</html>
```

3. Далее создаем Dockerfile для сборки образа:
   `nano Dockerfile`

```
FROM nginx:latest
COPY ./nginx/index.html /usr/share/nginx/html/index.html
```

4. Возвращаемся в корневую папку проекта и запускаем сборку образа nginx с указанием пути к Dockerfile:
   `sudo docker build . -t nginx -f nginx/Dockerfile`
5. Убеждаемся в успешной сборке:
   `sudo docker images`

```
ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
nginx        latest    5bda324dcdba   25 seconds ago   187MB
webapp       latest    3f6c00a96cfd   3 days ago       1.02GB
ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$
```

Образ для PostgreSQL

1. Для примера загрузим готовый образ из dockerhub:
   `docker pull postgres`
2. Убеждаемся, что образ успешно создан:
   `sudo docker images`

```
ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED              SIZE
nginx        latest    5bda324dcdba   About a minute ago   187MB
webapp       latest    3f6c00a96cfd   3 days ago           1.02GB
postgres     latest    fbd1be2cbb1f   7 weeks ago          417MB
ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$
```

Настройка docker-compose.yml

1. Для одновременного управления несколькими контейнерами необходимо подготовить файл с инструкциями:
   `nano docker-compose.yml`
2. Указываем тег версии. Используем "3.8", так как она является самой свежей на данный момент:
   `version: "3.8"`
3. Указываем раздел, в котором будем описывать сервисы (контейнеры) - в нашем случае это nginx и postgresql:
   `services:`
4. Начинаем описывать первый контейнер - nginx:
   `nginx:`
5. Указываем образ для создания контейнера:
   `image: nginx:latest`
6. Указываем порт для прослушки контейнером и порт для проброски запросов внутрь контейнера:

```
ports:
  - "80:80"
```

7. Проброс каталогов. Смонтируем локальный каталог с файлом index.html внутрь контейнера в качестве каталога /usr/share/nginx/html:

```
volumes:
  - ./nginx:/usr/share/nginx/html
```

8. Указываем докеру автоматически перезапускать приложение, если оно по какой-либо причине останавливается:
   `restart: always`
9. Финальный вид настройки для nginx:

```
nginx:
  image: nginx:latest
  ports:
    - "80:80"
  volumes:
    - ./nginx:/usr/share/nginx/html
  restart: always
```

10. Начинаем описывать второй контейнер - postgresql:
    `postgresql:`

11. Указываем образ для создания контейнера:
    `image: postgres:latest`
12. Указываем порты:

```
ports:
  - "5432:5432"
```

13. Указываем переменный окружения, в нашем случае это логин и пароль для пользователя postgres:

```
environment:
  - POSTGRES_PASSWORD=itransition
  - POSTGRES_USER=itransition
```

14. Устанавливаем флаг автоматического перезапуска:
    `restart: always`
15. Финальный вид настройки для postgresql:

```
postgresql:
  image: postgres:latest
  ports:
    - "5432:5432"
  environment:
    - POSTGRES_PASSWORD=itransition
    - POSTGRES_USER=itransition
  restart: always
```

Финальный вид docker-compose.yml

```
version: "3.8"
services:
  nginx:
    image: nginx:latest
    ports:
      - "80:80"
    volumes:
      - ./nginx:/usr/share/nginx/html
    restart: always
  postgresql:
    image: postgres:latest
    ports:
      - "5432:5432"
    environment:
      - POSTGRES_PASSWORD=itransition
```

```
      - POSTGRES_USER=itransition
    restart: always
```

Запуск многоконтейнерного приложения

1. Запускаем сборку и запуск контейнеров:

   ```
   sudo docker-compose up -d
   ```

   ```
   ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$ sudo docker-compose up -d
   Creating network "multicontainerapp_default" with the default driver
   Creating multicontainerapp_nginx_1      ... done
   Creating multicontainerapp_postgresql_1 ... done
   ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$
   ```

2. Проверяем состояние:

   ```
   sudo docker ps
   ```

   ```
   ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$ sudo docker ps
   CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS                                     NAMES
   6e2a43d034c0   postgres:latest    "docker-entrypoint.s…"   35 seconds ago   Up 34 seconds   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   multicontainerapp_postgresql_1
   0885b60b893c   nginx:latest       "/docker-entrypoint.…"   35 seconds ago   Up 34 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp           multicontainerapp_nginx_1
   ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$
   ```

3. Откроем страницу веб-приложения в браузере:

   

   ITransition, hello from Nginx container!

4. Убедимся, что приложения находятся в одной сети и могут друг друга видеть:

   ```
   sudo docker container inspect nginx
   ```


   ```
            "Networks": {
                "multicontainerapp_default": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": [
                        "nginx",
                        "9d0629c67f7f"
                    ],
                    "NetworkID": "eac435448d46538bd66af5c3f733803a8df1f42f7796bf7adc2ca139ee867cda",
                    "EndpointID": "e861ebecbafd6d1b21c6d6c93cdbf1e6360ec47f8152a3df8a42006471b60a2f",
                    "Gateway": "172.18.0.1",
                    "IPAddress": "172.18.0.2",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "MacAddress": "02:42:ac:12:00:02",
                    "DriverOpts": null
                }
            }
        }
    }
]
   ```


```
sudo docker container inspect postgresql
```


```
            "Networks": {
                "multicontainerapp_default": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": [
                        "9fc38ac7f9a6",
                        "postgresql"
                    ],
                    "NetworkID": "eac435448d46538bd66af5c3f733803a8df1f42f7796bf7adc2ca139ee867cda",
                    "EndpointID": "1b74b691424e61a6e53b4e24409f7ca1e85104c5f49c34169bf462fe0fe9ecdc",
                    "Gateway": "172.18.0.1",
                    "IPAddress": "172.18.0.3",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "MacAddress": "02:42:ac:12:00:03",
                    "DriverOpts": null
                }
            }
        }
    }
]
```


5. Остановим и удалим контейнеры:

```
sudo docker-compose down
```

```
ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$ sudo docker-compose down
Stopping multicontainerapp_postgresql_1 ... done
Stopping multicontainerapp_nginx_1      ... done
Removing multicontainerapp_postgresql_1 ... done
Removing multicontainerapp_nginx_1      ... done
Removing network multicontainerapp_default
ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$
```

6. Снова запустим контейнеры:

```
sudo docker-compose up
```

```
ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$ sudo docker-compose up
Creating network "multicontainerapp_default" with the default driver
Creating multicontainerapp_nginx_1      ... done
Creating multicontainerapp_postgresql_1 ... done
Attaching to multicontainerapp_nginx_1, multicontainerapp_postgresql_1
nginx_1       | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
nginx_1       | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
nginx_1       | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
nginx_1       | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
nginx_1       | 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
nginx_1       | /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
nginx_1       | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
nginx_1       | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
nginx_1       | /docker-entrypoint.sh: Configuration complete; ready for start up
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: using the "epoll" event method
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: nginx/1.25.3
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: OS: Linux 6.2.0-36-generic
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: start worker processes
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: start worker process 29
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: start worker process 30
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: start worker process 31
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: start worker process 32
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: start worker process 33
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: start worker process 34
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: start worker process 35
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: start worker process 36
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: start worker process 37
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: start worker process 38
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: start worker process 39
nginx_1       | 2023/11/06 17:41:00 [notice] 1#1: start worker process 40
```

```
postgresql_1  |
postgresql_1  | Success. You can now start the database server using:
postgresql_1  |
postgresql_1  |     pg_ctl -D /var/lib/postgresql/data -l logfile start
postgresql_1  |
postgresql_1  | initdb: warning: enabling "trust" authentication for local connections
postgresql_1  | 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.
postgresql_1  | waiting for server to start....2023-11-06 17:41:01.433 UTC [48] LOG:  starting PostgreSQL 16.0 (Debian 16.0-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2
postgresql_1  | 2023-11-06 17:41:01.435 UTC [48] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
postgresql_1  | 2023-11-06 17:41:01.445 UTC [51] LOG:  database system was shut down at 2023-11-06 17:41:01 UTC
postgresql_1  | 2023-11-06 17:41:01.451 UTC [48] LOG:  database system is ready to accept connections
postgresql_1  |  done
postgresql_1  | server started
postgresql_1  | CREATE DATABASE
postgresql_1  |
postgresql_1  |
postgresql_1  | /usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*
postgresql_1  |
postgresql_1  | waiting for server to shut down...2023-11-06 17:41:01.608 UTC [48] LOG:  received fast shutdown request
postgresql_1  | .2023-11-06 17:41:01.610 UTC [48] LOG:  aborting any active transactions
postgresql_1  | 2023-11-06 17:41:01.612 UTC [48] LOG:  background worker "logical replication launcher" (PID 54) exited with exit code 1
postgresql_1  | 2023-11-06 17:41:01.612 UTC [49] LOG:  shutting down
postgresql_1  | 2023-11-06 17:41:01.613 UTC [49] LOG:  checkpoint starting: shutdown immediate
postgresql_1  | 2023-11-06 17:41:01.662 UTC [49] LOG:  checkpoint complete: wrote 923 buffers (5.6%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.021 s, sync=0.021 s, total=0.051 s;
st=0.003 s, average=0.001 s; distance=4257 kB, estimate=4257 kB; lsn=0/19130D8, redo lsn=0/19130D8
postgresql_1  | 2023-11-06 17:41:01.672 UTC [48] LOG:  database system is shut down
postgresql_1  |  done
postgresql_1  | server stopped
postgresql_1  |
postgresql_1  | PostgreSQL init process complete; ready for start up.
postgresql_1  |
postgresql_1  | 2023-11-06 17:41:01.737 UTC [1] LOG:  starting PostgreSQL 16.0 (Debian 16.0-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
postgresql_1  | 2023-11-06 17:41:01.737 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
postgresql_1  | 2023-11-06 17:41:01.737 UTC [1] LOG:  listening on IPv6 address "::", port 5432
postgresql_1  | 2023-11-06 17:41:01.741 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
postgresql_1  | 2023-11-06 17:41:01.746 UTC [64] LOG:  database system was shut down at 2023-11-06 17:41:01 UTC
postgresql_1  | 2023-11-06 17:41:01.751 UTC [1] LOG:  database system is ready to accept connections
```

7. Удалим контейнеры:

```
sudo docker-compose rm
```

```
ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$ sudo docker-compose rm
Going to remove multicontainerapp_postgresql_1, multicontainerapp_nginx_1
Are you sure? [yN] y
Removing multicontainerapp_postgresql_1 ... done
Removing multicontainerapp_nginx_1      ... done
ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$
```

8. Убедимся, что удаление прошло успешно:

```
sudo docker ps -a
```

```
ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED      STATUS                     PORTS     NAMES
3c2ed13ae575   webapp    "gunicorn --bind 0.0…"   3 days ago   Exited (0) 3 days ago              angry_haslett
ruslan@ruslan-Z690-UD:~/IT-2023/task_2/task_2.5/multicontainerapp$
```