

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа №6

Выполнил:
Золотухин Артем
Группа К33392
Проверил:
Добряков Д. И.

Санкт-Петербур

г 2024 г.

Задача

Необходимо настроить автодеплой (с триггером на обновление кода в вашем репозитории, на определённой ветке) для вашего приложения на удалённый сервер с использованием Github Actions или Gitlab CI (любая другая CI-система также может быть использована).

Ход работы

Создал файл build.yml

```
name: Deploy

on:
  push:
    branches:
      - lab-4

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: actions/setup-node@v1
        with:
          node-version: 20.x
      - name: Copy
        run: |
          sudo apt-get update
          mkdir -p ~/.ssh/
          echo "${{ secrets.SSH_PRIVATE_KEY }}" >
~/.ssh/private.key
          chmod 600 ~/.ssh/private.key
          ssh -o StrictHostKeyChecking=no -i
~/.ssh/private.key ubuntu@213.219.215.18 'cd
/home/ubuntu/ITMO-ICT-Backend-2024 && git pull && cd
/home/ubuntu/ITMO-ICT-Backend-2024/labs/K33392/Золотухин\
Артём/lab4/ && echo "${{ secrets.AUTH_SERVICE_ENV }}" >
./auth-service/.env && echo "${{ secrets.MAIN_SERVICE_ENV
}}" > ./main/.env && sudo docker compose up -d --build'
```

```

=> [main] exporting to image
=> exporting layers
=> writing image sha256:7bfee71c926dccb57e684ee084439ba3240b91615dbba747d98da5f7e0c0f74a
=> naming to docker.io/library/lab4-main
[*] Running 5/5
✓Container lab4-db-main-1 Running
✓Container lab4-rabbitmq-1 Running
✓Container lab4-main-1 Recreated
✓Container lab4-db-auth-1 Running
✓Container lab4-auth-1 Recreated
Attaching to auth-1, db-auth-1, db-main-1, main-1, rabbitmq-1
Environment variables loaded from .env
main-1 | Prisma schema loaded from prisma/schema.prisma
main-1 | Datasource "db": PostgreSQL database "db_auth", schema "public" at "db_auth:5432"
auth-1 | Prisma schema loaded from prisma/schema.prisma
auth-1 | Datasource "db": PostgreSQL database "db_main", schema "public" at "db_main:5432"
auth-1 |
auth-1 | 1 migration found in prisma/migrations
main-1 |
main-1 | 1 migration found in prisma/migrations
main-1 |
main-1 | Applying migration '20240410135851_init'
auth-1 | Applying migration '20240416113508_init'
auth-1 |
auth-1 | The following migration(s) have been applied:
auth-1 |
auth-1 | migrations/
auth-1 |   └─ 20240416113508_init/
auth-1 |       └─ migration.sql
auth-1 |
auth-1 | All migrations have been successfully applied.
main-1 |
main-1 | The following migration(s) have been applied:
main-1 |
main-1 | migrations/
main-1 |   └─ 20240410135851_init/
main-1 |       └─ migration.sql
main-1 |
main-1 | All migrations have been successfully applied.
auth-1 | > auth-service@1.0.0 dev
auth-1 | > tsnd --respawn --transpile-only --exit-child app.ts
auth-1 |
main-1 | > FastifyTryout@1.0.0 dev
main-1 | > tsnd --respawn --transpile-only --exit-child src/app.ts
main-1 |
[INFO] 17:10:04 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.4.5)
[INFO] 17:10:04 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.4.5)
rabbitmq-1 | 2024-06-18 17:10:30.836693+00:00 [info] <0.776.0> accepting AMQP connection <0.776.0> (172.18.0.5:50632 → 172.18.0.4:5672)
rabbitmq-1 | 2024-06-18 17:10:31.084900+00:00 [info] <0.776.0> connection <0.776.0> (172.18.0.5:50632 → 172.18.0.4:5672): user 'guest' authenticated and granted access to vhost '/'
auth-1 | Auth service running on port 3081
rabbitmq-1 | 2024-06-18 17:10:32.468264+00:00 [info] <0.796.0> accepting AMQP connection <0.796.0> (172.18.0.6:40330 → 172.18.0.4:5672)
rabbitmq-1 | 2024-06-18 17:10:32.515794+00:00 [info] <0.796.0> connection <0.796.0> (172.18.0.6:40330 → 172.18.0.4:5672): user 'guest' authenticated and granted access to vhost '/'
main-1 | Server running on 3080

```

Этот gh Actions workflow автоматически разворачивает приложение при каждом пуше в ветку `lab-4`. Он выполняет клонирование репозитория, устанавливает Node.js, а затем подключается к удалённому серверу, где обновляет репозиторий, устанавливает переменные окружения и запускает контейнеры через Docker Compose. Это все обеспечивает автоматическое обновление и разворачивание приложения на удалённом сервере без необходимости ручного вмешательства каждый раз, когда вносятся изменения в кодовую базу.

Вывод:

В ходе данной работы была изучена технология CI/CD и в частности был изучен github action