САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Домашнее задание №6

Выполнил:

Соловьева П.А.

Группа К3344

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

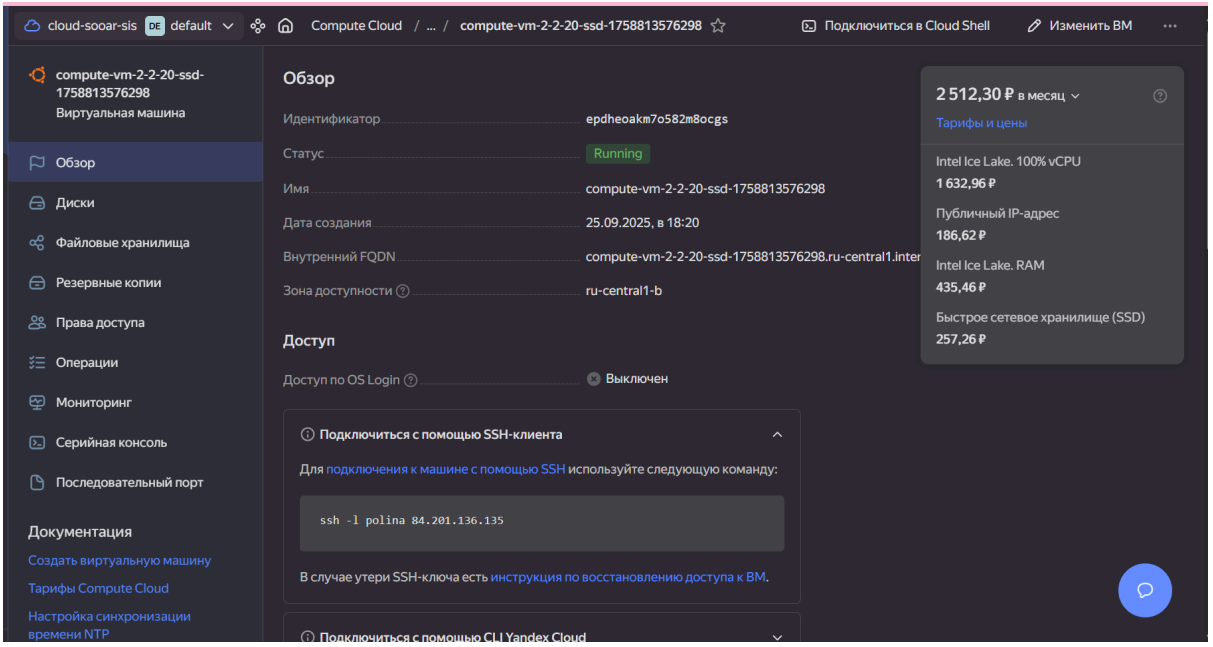
Задача

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

1. Подготовка сервера

Сначала подготовила сервер на Yandex Cloud и локальную среду для деплоя. Сгенерировала SSH-ключ для безопасного подключения к серверу и добавила его в GitHub secrets. На сервере убедилась, что установлены все необходимые зависимости и настроена папка для микросервисов.



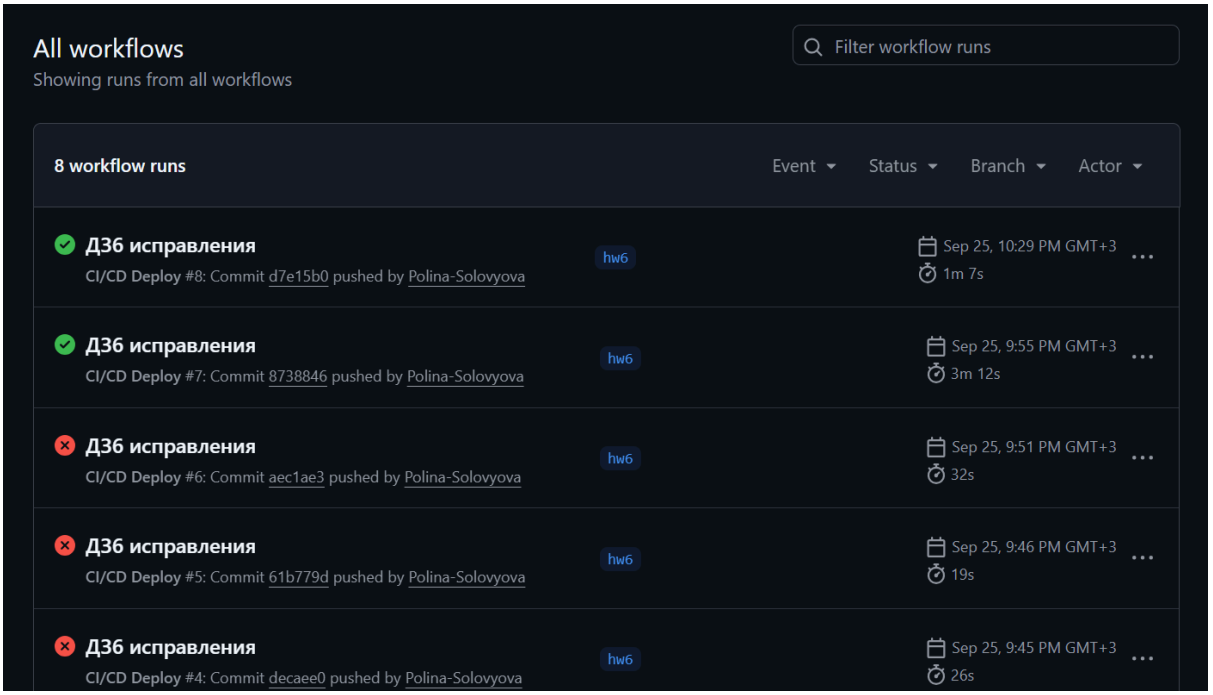
2. Настройка workflow GitHub Actions

Далее настроила workflow GitHub Actions для автоматического деплоя при пуше в ветку hw6. В workflow указала шаги: клонирование репозитория, копирование кода на сервер с помощью SCP и перезапуск сервисов через Docker Compose. Так же прописала скрипты для создания .env файлов для каждого сервиса, предварительно добавив необходимые данные в GitHub secrets.

```yaml
name: CI/CD Deploy

on:
  push:
    branches:
      - hw6

jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
    - name: Checkout code
      uses: actions/checkout@v3

    - name: Copy code to server
      uses: appleboy/scp-action@v0.1.3
      with:
        host: ${{ secrets.SERVER_IP }}
        username: ${{ secrets.SERVER_USER }}
        key: ${{ secrets.SERVER_SSH_KEY }}
        source: "БР2/Соловьева_Полина/homeworks/rental_service/*"
        target: "~/rental_service"
        strip_components: 3

    - name: Create .env files on server for users_service
      uses: appleboy/ssh-action@v0.1.7
      with:
          host: ${{ secrets.SERVER_IP }}
          username: ${{ secrets.SERVER_USER }}
          key: ${{ secrets.SERVER_SSH_KEY }}
          script: |
            mkdir -p ~/rental_service/rental_service/users_service
            touch ~/rental_service/rental_service/users_service/.env
            cat <<-EOT > ~/rental_service/rental_service/users_service/.env
            PORT=5001
            API_URL=http://${{ secrets.API_HOST }}:5001
            DATABASE_URL=${{ secrets.USERS_SERVICE_DB_URL }}
            RABBITMQ_URL=${{ secrets.RABBITMQ_URL }}
            EOT

    - name: Create .env files on server for property_service
      uses: appleboy/ssh-action@v0.1.7
      with:
          host: ${{ secrets.SERVER_IP }}
          username: ${{ secrets.SERVER_USER }}
          key: ${{ secrets.SERVER_SSH_KEY }}
          script: |
            mkdir -p ~/rental_service/rental_service/property_service
            touch ~/rental_service/rental_service/property_service/.env
            cat <<-EOT >
~/rental_service/rental_service/property_service/.env
            PORT=5002
            INTERNAL_SERVICE_TOKEN=${{ secrets.INTERNAL_SERVICE_TOKEN }}
            API_URL=http://${{ secrets.API_HOST }}:5002
            DATABASE_URL=${{ secrets.PROPERTY_SERVICE_DB_URL }}
            RABBITMQ_URL=${{ secrets.RABBITMQ_URL }}
            USERS_SERVICE_URL=${{ secrets.USERS_SERVICE_URL }}
            EOT

    - name: Create .env files on server for rental_service
```

```
    uses: appleboy/ssh-action@v0.1.7
    with:
        host: ${{ secrets.SERVER_IP }}
        username: ${{ secrets.SERVER_USER }}
        key: ${{ secrets.SERVER_SSH_KEY }}
        script: |
          mkdir -p ~/rental_service/rental_service/rental_service
          touch ~/rental_service/rental_service/rental_service/.env
          cat <<-EOT > ~/rental_service/rental_service/rental_service/.env
          PORT=5003
          INTERNAL_SERVICE_TOKEN=${{ secrets.INTERNAL_SERVICE_TOKEN }}
          API_URL=http://${{ secrets.API_HOST }}:5003
          DATABASE_URL=${{ secrets.RENTAL_SERVICE_DB_URL }}
          RABBITMQ_URL=${{ secrets.RABBITMQ_URL }}
          USERS_SERVICE_URL=${{ secrets.USERS_SERVICE_URL }}
          PROPERTY_SERVICE_URL=${{ secrets.PROPERTY_SERVICE_URL }}
          EOT

  - name: Run Docker Compose on server
    uses: appleboy/ssh-action@v0.1.7
    with:
      host: ${{ secrets.SERVER_IP }}
      username: ${{ secrets.SERVER_USER }}
      key: ${{ secrets.SERVER_SSH_KEY }}
      script: |
        cd ~/rental_service/rental_service
        docker compose down
        docker compose up -d --build
```

После настройки workflow сделала тестовый пуш в ветку hw6. Деплой провалился несколько раз, но я исправила ошибки и все поднялось. На сервере убедилась, что все контейнеры подняты и работают корректно.



Проверила доступность сервисов через порты, убедилась, что база данных и очереди сообщений функционируют, а микросервисы слушают свои порты.

Вывод

В ходе выполнения работы была успешно настроена система автоматического деплоя приложения на удалённый сервер с использованием GitHub Actions. В процессе работы были выполнены следующие шаги:

1. Подготовка сервера — создан сервер на Yandex Cloud, настроено безопасное подключение через SSH, установлены необходимые зависимости, подготовлены каталоги для микросервисов.
2. Настройка workflow GitHub Actions — разработан и протестирован workflow для автоматического деплоя при пуше в ветку hw6. В workflow реализованы шаги: клонирование репозитория, копирование файлов на сервер через SCP, создание .env файлов для каждого микросервиса с использованием секретов GitHub, а также перезапуск сервисов через Docker Compose.
3. Тестирование и отладка — выполнен тестовый пуш, выявлены и исправлены ошибки деплоя. После успешного запуска проверена корректная работа всех микросервисов: контейнеры запущены, сервисы доступны по своим портам, базы данных и очереди сообщений функционируют.

Таким образом, цель работы достигнута: настроена полностью автоматизированная система CI/CD, обеспечивающая развертывание приложения на сервере при обновлении кода в репозитории, что позволяет ускорить процесс разработки и снизить риск ошибок при ручном деплое.