

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа 6

Выполнил:

Сахно Ярослав

Группа К3341

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.

Ход работы

Схема взаимодействия.

1. build-and-push
 - a. собирает Docker-образы gateway, auth-service, course-service, admin-service;
 - b. пушит их в GitHub Container Registry (GHCR).
2. deploy
 - a. подключается по SSH (ключ берёт из secrets.SSH_PRIVATE_KEY);
 - b. обновляет docker-compose.yml на сервере (rsync);
 - c. делает docker compose pull && docker compose up -d — выкатывает свежие образы.

Был написал следующий файл для Github Actions (см. Листинг 1).

Листинг 1 - Модель docker-compose.yml:

```
name: CI / CD→ deploy to VPS

on:
  push:
    branches: [ "main" ]

jobs:

  # 1) BUILD+PUSH images to GHCR

  build-and-push:
```

```
runs-on: ubuntu-latest

env:

  REGISTRY: ghcr.io

  IMAGE_PREFIX: ghcr.io/${{ github.repository_owner }}/homework

  COMPOSE_PROFILES: "auth,course,admin,gateway" # какие
сервисы билдим

steps:

  - name: Checkout code

    uses: actions/checkout@v4

  - name: Login to GitHub Container Registry

    uses: docker/login-action@v3

    with:

      registry: ${ env.REGISTRY }

      username: ${ github.actor }

      password: ${ secrets.GITHUB_TOKEN }

  - name: Set up QEMU (multi-arch)

    uses: docker/setup-qemu-action@v3

  - name: Set up Docker Buildx

    uses: docker/setup-buildx-action@v3

  - name: Build & push service images (matrix)

    uses: docker/build-push-action@v5
```

```
with:

  push: true

  tags: |

    ${ env.IMAGE_PREFIX }/gateway:latest

    ${ env.IMAGE_PREFIX }/gateway:${ github.sha }

  context: ./gateway

  file: ./gateway/Dockerfile

- name: Build & push auth-service

  uses: docker/build-push-action@v5

  with:

    push: true

    tags: |

      ${ env.IMAGE_PREFIX }/auth-service:latest

      ${ env.IMAGE_PREFIX }/auth-service:${ github.sha }

    context: ./auth

    file: ./auth/Dockerfile

- name: Build & push course-service

  uses: docker/build-push-action@v5

  with:

    push: true

    tags: |

      ${ env.IMAGE_PREFIX }/course-service:latest

      ${ env.IMAGE_PREFIX }/course-service:${ github.sha
}}

  context: ./course
```

```
file: ./course/Dockerfile

- name: Build & push admin-service
  uses: docker/build-push-action@v5
  with:
    push: true
    tags: |
      ${{ env.IMAGE_PREFIX }}/admin-service:latest
      ${{ env.IMAGE_PREFIX }}/admin-service:${{ github.sha }}
    context: ./admin
    file: ./admin/Dockerfile

# 2) DEPLOY to remote server via SSH

deploy:
  runs-on: ubuntu-latest
  needs: build-and-push

  steps:
    - name: Setup SSH agent
      uses: webfactory/ssh-agent@v0.9.0
      with:
        ssh-private-key: ${{ secrets.SSH_PRIVATE_KEY }}

    - name: Copy updated docker-compose to server (rsync)
      run: |
        rsync -avz --delete \
```

```
-e "ssh -o StrictHostKeyChecking=no" \

docker-compose.yml \

    ${{ secrets.SSH_USER }}@${{ secrets.SSH_HOST }}:${{
secrets.REMOTE_APP_DIR }}/

- name: Redeploy via SSH

  run: |

    ssh -o StrictHostKeyChecking=no ${{ secrets.SSH_USER
}}@${{ secrets.SSH_HOST }} << 'EOF'

    set -e

    cd ${{ secrets.REMOTE_APP_DIR }}

    # Используем самые свежие тэги :latest, вытягиваем
образы из GHCR

    docker compose pull

    docker compose up -d --remove-orphans

    docker image prune -f

    EOF
```

Вывод

Был реализован автодеплой с использованием Github Actions.