

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Практическая работа № 6  
“CI/CD”

Выполнил:

Коротин А.М.

К33392

Проверил:

Добряков Д. И.

Санкт-Петербург

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

Для выполнения задачи была создана ВМ на платформе Yandex Cloud (рисунок 1).

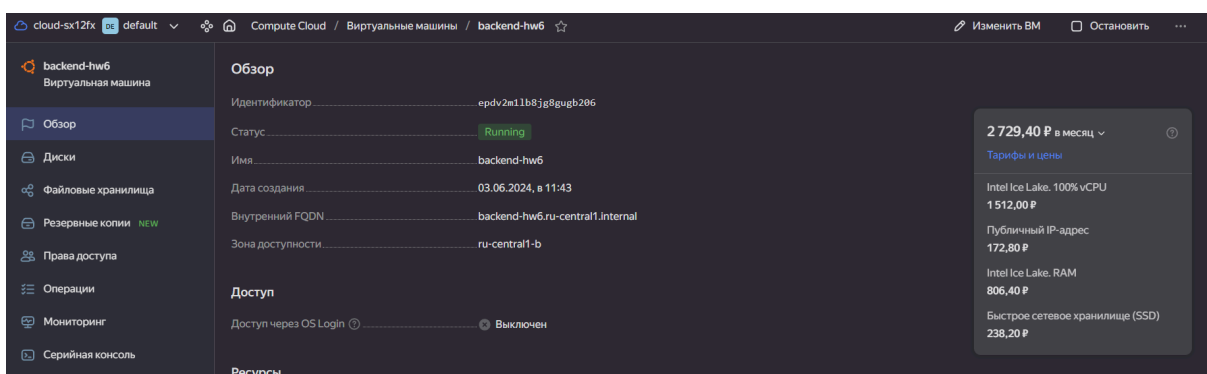


Рисунок 1 — Параметры ВМ в Yandex Cloud

Далее в секции Secrets репозитория GitHub были созданы секретные переменные, используемые для деплоя (рисунок 2):

- хост ВМ,
- пользователь ВМ,
- секретный ключ для подключения к ВМ.

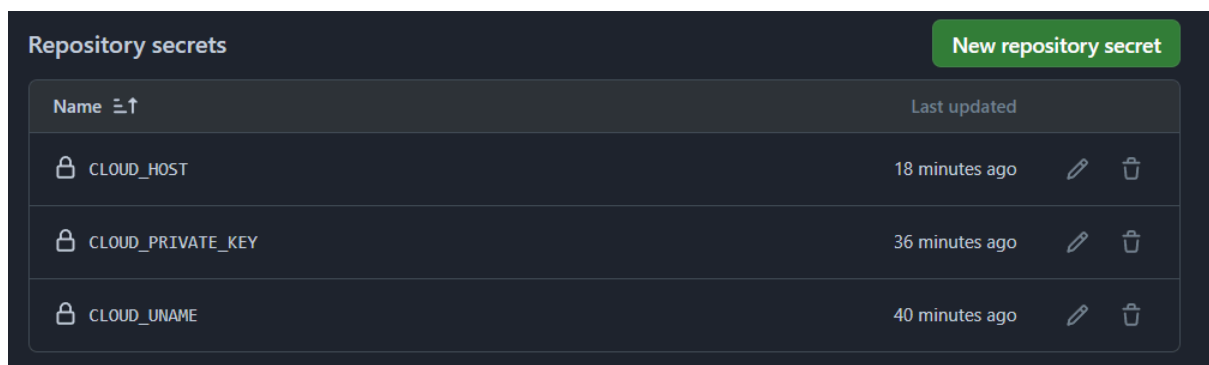


Рисунок 2 — Переменные репозитория

Далее был настроен сценарий GitHub Action для деплоя (рисунок 3).

```
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: actions/setup-node@v1
        with:
          node-version: 20.x
      - name: Preparation
        run: |
          sudo apt-get update
          sudo apt-get install rsync -y
          mkdir "$HOME/.ssh"
          echo "${{ secrets.CLOUD_PRIVATE_KEY }}" > "$HOME/.ssh/key"
          chmod 600 "$HOME/.ssh/key"
      - name: Sync
        run: |
          rsync -e "ssh -p 22 -i $HOME/.ssh/key -o StrictHostKeyChecking=no" -avz --exclude '.git'
            --chown=github:www-data --chmod=Dg=rwx,Fg=rwx --archive --compress --delete .
            "${{ secrets.CLOUD_UNAME }}"@${{ secrets.CLOUD_HOST }}:/home/alexey/deploy
      - name: Build
        run: |
          ssh -p 22 -i $HOME/.ssh/key "${{ secrets.CLOUD_UNAME }}"@${{ secrets.CLOUD_HOST }}
            'cd /home/alexey/deploy/labs/K33392/Korotin_Alexey/lab2 && sudo docker compose up -d --build'
```

Рисунок 3 — Настройка деплоя

После push запускается сценарий запуска, который можно видеть на вкладке Actions (рисунок 4).

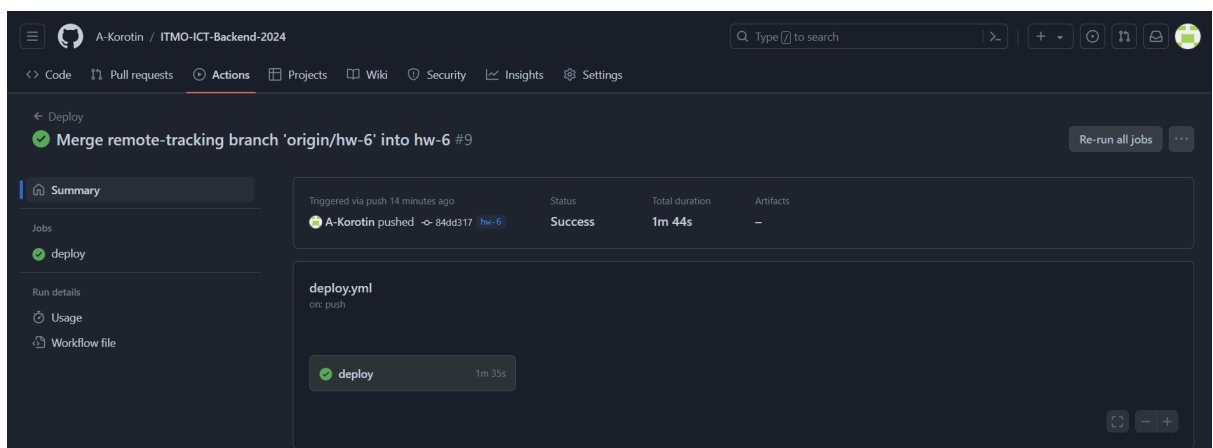


Рисунок 4 — Выполненный сценарий на вкладке Actions

Далее проверим правильность деплоя на ВМ через ssh. Подключаемся к ВМ (рисунок 5).

```
C:\Users\Alexey\.ssh>ssh alexey@158.160.92.246 -i id_backend
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-107-generic)
```

Рисунок 5 — Подключение к ВМ по ssh

Далее проверим список запущенных контейнеров командой `sudo docker ps` (рисунок 6).

```
alexey@backend:~$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
9d86a39822e1   provectuslabs/kafka-ui:latest      "/bin/sh -c 'java --"   17 minutes ago Up 16 minutes   0.0.0.0:8090->8080/tcp, :::8090->8080/tcp   lab2-kafka-ui-1
5ea9e39ff952   lab2-smart-devices-service         "npm run runApp"        17 minutes ago Up 16 minutes   0.0.0.0:12345->12345/tcp, :::12345->12345/tcp   lab2-smart-devices-service-1
1397cdf2116f   confluentinc/cp-kafka:latest       "/etc/confluent/dock-   17 minutes ago Up 16 minutes   9092/tcp, 0.0.0.0:29092->29092/tcp, :::29092->29092/tcp   lab2-kafka-1
6a38ef479909   lab2-auth-service                  "npm run runApp"        17 minutes ago Up 16 minutes   0.0.0.0:1234->1234/tcp, :::1234->1234/tcp   lab2-auth-service-1
30f18c9874db   postgres:latest                    "docker-entrypoint.s-"   17 minutes ago Up 16 minutes (healthy)   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   lab2-auth-db-1
467ed4bd69ab   confluentinc/cp-zookeeper:latest   "/etc/confluent/dock-   17 minutes ago Up 16 minutes   2888/tcp, 3888/tcp, 0.0.0.0:22181->22181/tcp, :::22181->22181/tcp   lab2-zookeeper-1
2ff70f498dd2   postgres:latest                    "docker-entrypoint.s-"   17 minutes ago Up 16 minutes (healthy)   0.0.0.0:5433->5432/tcp, :::5433->5432/tcp   lab2-app-db-1
```

Рисунок 6 — Список запущенных контейнеров

Можем видеть, что деплой произошёл успешно, все контейнеры были запущены.

## Вывод

В ходе выполнения работы были изучены основы CI/CD. Также был написан скрипт для облачного деплоя решения с использованием технологии GitHub Actions.