

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа №6

Выполнил:

Таначев Егор

Группа К33412

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

Задача

Необходимо настроить автодеплой (с триггером на обновление кода в нашем 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

Создадим аккаунт на Yandex.Cloud и перейдем Реестр Container Registry. Добавим новый каталог под названием backend, как показано на Рисунке 1.

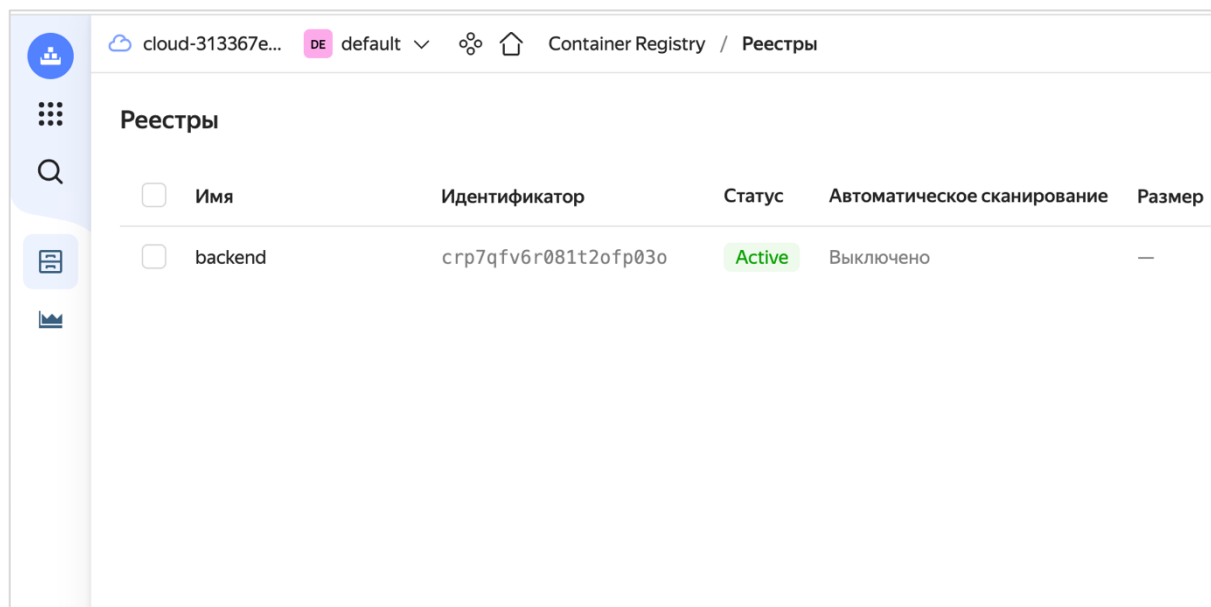


Рисунок 1 – Container Registry

Установим Yandex Cloud CLI и произведем все необходимые настройки. Для этого использовалась [документация](#) от Yandex Cloud. После этого сконфигурируем Docker, как показано на Рисунке 2.

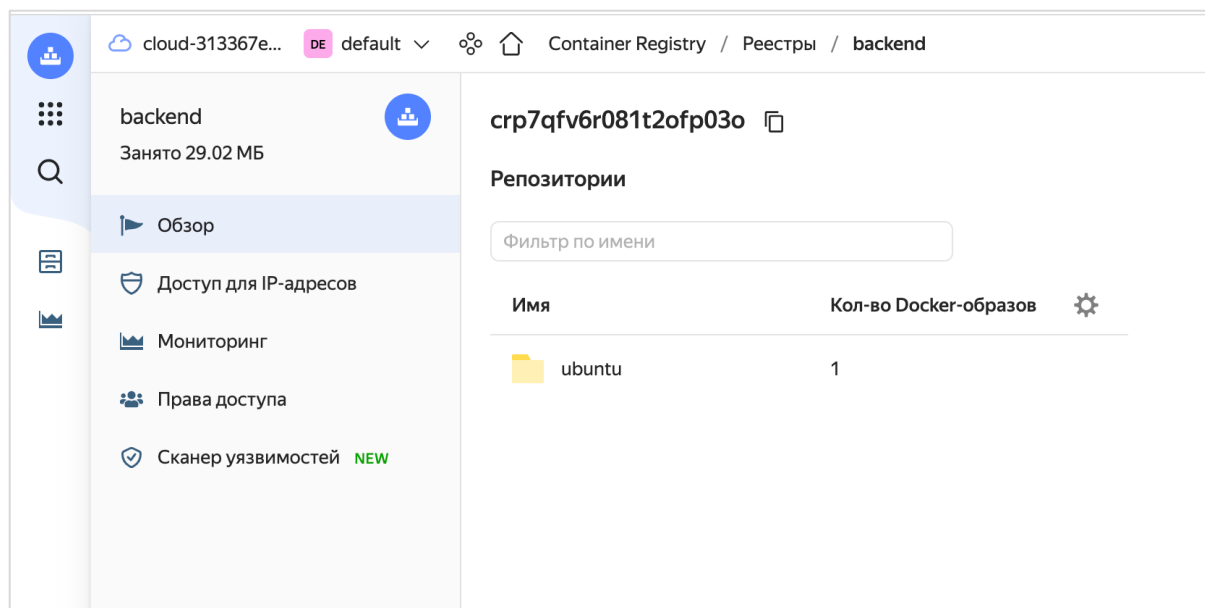


Рисунок 2 – Docker образ

Перейдем в раздел Сервисных аккаунтов и создадим новый аккаунт, для этого есть специальная кнопка на платформе. Назовем его register и назначим ему роль, как показано на Рисунке 3.

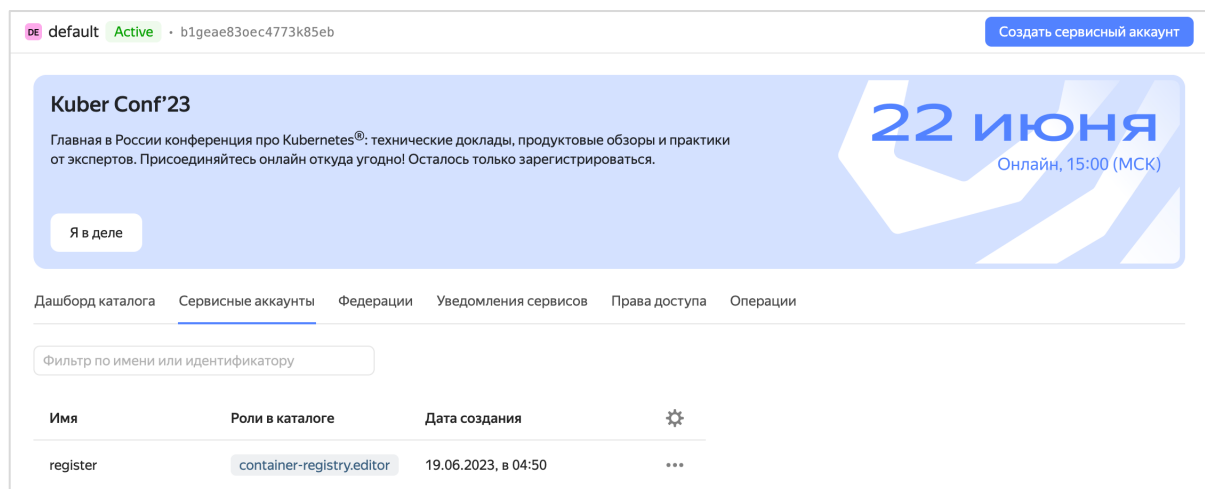


Рисунок 3 – Сервисный аккаунт

Создадим автоматизированный ключ и экспортируем его в json формате, как показано на Рисунке 4.

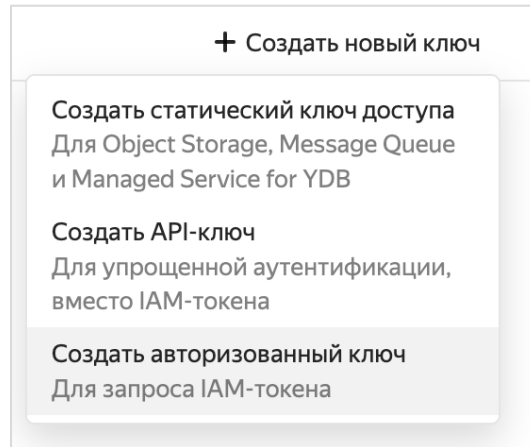


Рисунок 4 – Создание ключа

Перейдем в Compute Cloud и создадим виртуальную машину на Ubuntu 22.04, как показано на Рисунке 5.

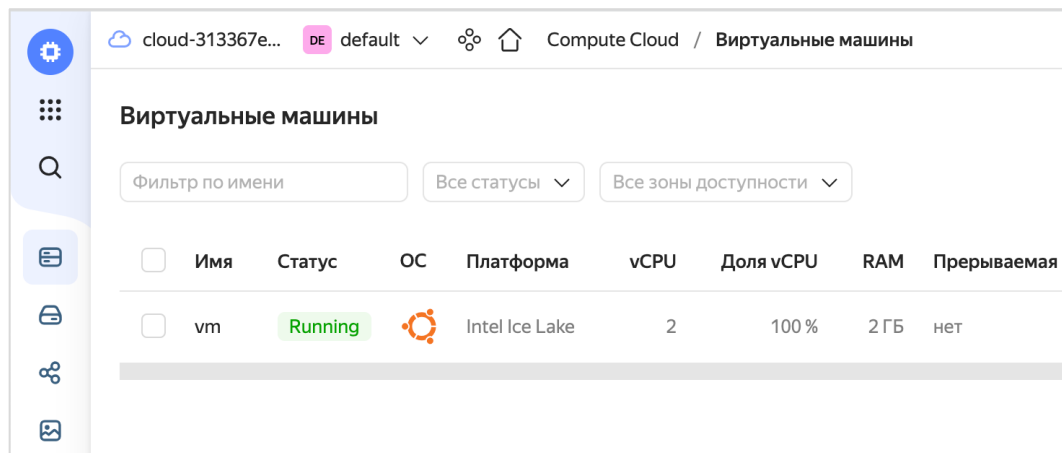


Рисунок 5 – Виртуальная машина

После этого создадим папку HW4 в ветке hw6 и добавим в него нашего приложение. Создадим autodeploy.yml для GitHub CI, как показано на Рисунке 6.

```

1  name: Docker deploy
2
3  on:
4    push:
5      branches:
6        - hw6
7
8  jobs:
9    build:
10     runs-on: ubuntu-latest
11
12     steps:
13       - uses: actions/checkout@v2
14
15       - name: Login (Yandex Cloud)
16         id: login-cr
17         uses: yc-actions/yc-cr-login@v1
18         with:
19           yc-sa-json-credentials: ${ secrets.AUTHORIZED_KEY }

```

```

20
21     - name: Build image (Yandex Cloud)
22       env:
23         CR_REGISTRY: crp7qfv6r081t2ofp03o
24         CR_REPOSITORY: backend
25         IMAGE_TAG: ${ github.sha }
26       run: |
27         docker build -t cr.yandex/$CR_REGISTRY/$CR_REPOSITORY:user$IMAGE_TAG ./homeworks/K33412/Tanachev\ Egor/HW6/task/user
28         docker push cr.yandex/$CR_REGISTRY/$CR_REPOSITORY:user$IMAGE_TAG
29         docker build -t cr.yandex/$CR_REGISTRY/$CR_REPOSITORY:market$IMAGE_TAG ./homeworks/K33412/Tanachev\ Egor/HW6/task/market
30         docker push cr.yandex/$CR_REGISTRY/$CR_REPOSITORY:market$IMAGE_TAG
31
32
33     - name: Connect to VM via SSH
34       uses: appleboy/ssh-action@master
35       with:
36         host: 51.250.111.202
37         username: admin
38         key: ${ secrets.KEY }
39         passphrase: ${ secrets.PASSPHRASE }
40         port: 22
41
42     - name: Run commands on VM
43       env:
44         YANDEX_KEY: ${ secrets.YANDEX_KEY }
45         IMAGE_TAG: ${ github.sha }
46       run: |
47         sudo docker login --username oauth --password $YANDEX_KEY cr.yandex
48         if [ "$(docker ps -q)" ]; then
49           sudo docker kill $(docker ps -q) && /dev/null
50         fi
51         if [ "$(docker ps -a -q)" ]; then
52           sudo docker rm $(docker ps -a -q) && /dev/null
53         fi
54         sudo docker run -dp 80:1111 cr.yandex/crp7qfv6r081t2ofp03o/backend:user${ github.sha } && sudo docker run -dp 81:2222 cr

```

Рисунок 6 – Файл autodeploy.yml

Теперь попробуем запустим изменение на ветку hw6 и проверим Actions. Как можем видеть у нас прошел деплой, как показано на Рисунке 7.

The screenshot shows the GitHub Actions interface for a repository named 'egortanachev / ITMO-ICT-Backend-2023'. The workflow 'Update autodeploy.yml #86' is displayed, with a summary of jobs on the left and a detailed log of the 'build' job on the right.

Summary:

- Jobs: build (successful)
- Run details
- Usage
- Workflow file

build job log:

- Set up job: 3s
- Build appleboy/ssh-action@master: 5s
- Run actions/checkout@v2: 1s
- Login (Yandex Cloud): 1s
- Build image (Yandex Cloud): 1m 38s
- Connect to VM via SSH: 2s
- Run commands on VM: 2s
- Post Login (Yandex Cloud): 0s
- Post Run actions/checkout@v2: 0s
- Complete job: 0s

Рисунок 7 – Успешное выполнение деплоя

Вывод

В результате работы был успешно настроен автодеплой для приложения на удалённый сервер с использованием CI-системы (Github Actions или Gitlab CI). Автодеплой настроен с использованием триггера, который реагирует на обновления кода в репозитории, при этом автодеплой активируется только для определённой ветки. Это позволяет автоматизировать процесс развертывания приложения на сервере и упростить процесс обновления приложения.