

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Практическая работа №6  
CI/CD

Выполнила:  
Зайцева А. А.  
Группа К33402

Проверил:  
Добряков Д. И.

Санкт-Петербург

2023 г.

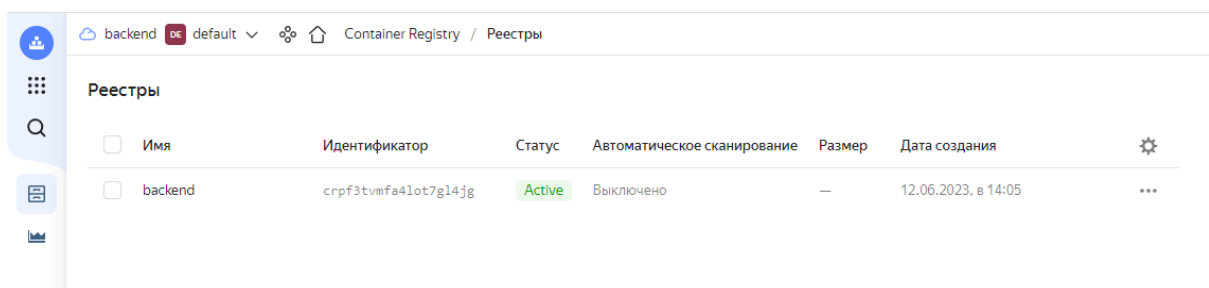
## Задача

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 работы были выбраны сервисы Github Actions и Yandex Cloud (Yandex Container Registry). Сконфигурированная система должна работать следующим образом: при пуше в ветку Github репозитория **hw6** должен запускаться **Github Action**, который собирает и загружает **Docker** образов в реестр **Yandex Container Registry**, а затем, подключаясь к виртуальной машине по SSH, обновляет и перезапускает контейнеры с приложением.

Сначала был создан аккаунт сервиса Yandex Cloud, в рамках которого были созданы организация, облако и каталог. В каталог включена сеть по умолчанию. Затем был создан реестр контейнеров Yandex Container Registry для хранения образа приложения.



Затем была установлена и сконфигурирована утилита **Yandex Cloud CLI**. С ее помощью Docker был сконфигурирован для работы с Yandex Container Registry.


Через консоль управления Yandex Cloud был создан сервисный аккаунт **registry-sa** с правами на редактирование реестра контейнеров.

Дашборд каталога    Сервисные аккаунты    Федерации    Уведомления сервисов    Права доступа    Операции

Фильтр по имени или идентификатору

Имя	Роли в каталоге	Дата создания	
registry-sa	container-registry.editor	12.06.2023, в 14:53	...

Была создана виртуальная машина на базе операционной системы Ubuntu 18. На созданной виртуальной машине был установлен Docker.

backend default Compute Cloud / Виртуальные машины											
Виртуальные машины											
Фильтр по имени    Все статусы    Все зоны доступности											
<input type="checkbox"/>	Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Зона доступности	Внутренний IPv4
<input type="checkbox"/>	backend-vm	Running	 Intel Ice Lake	2	100%	2 Гб	нет		13 Гб	ru-central1-a	10.128.0.29

Через Yandex Cloud CLI был создан ключ доступа для сервисного аккаунта registry-sa в формате JSON. Созданный ключ был загружен в раздел Secret Github репозитория с названием YC\_SA\_JSON\_CREDENTIALS.

Затем в репозитории Github был создан новый Github Action с содержимым:

```
1. name: Docker deploy
2.
3. on:
4.   push:
5.     # Запускаем пайплайн при пушах в ветку hw6
6.     branches: [ "hw6" ]
7.
8. jobs:
9.   build:
10.    runs-on: ubuntu-latest
11.
12.    steps:
13.      # Переходим в корень актуального состояния ветки
14.      - uses: actions/checkout@v2
15.
16.      # Авторизация в сервисе Yandex Cloud Container Registry
17.      - name: Login to Yandex Cloud Container Registry
18.        id: login-cr
19.        uses: yc-actions/yc-cr-login@v1
20.        with:
21.          yc-sa-json-credentials: ${ secrets.YC_SA_JSON_CREDENTIALS }
22.
23.      # Собираем и загружаем образы в реестр
24.      - name: Build, tag, and push image to Yandex Cloud Container Registry
25.        env:
26.          CR_REGISTRY: crpf3tvmfa4lot7gl4jg
27.          CR_REPOSITORY: backend
28.          IMAGE_TAG: ${ github.sha }
29.        run: |
```

```

30.     docker build -t cr.yandex/$CR_REGISTRY/$CR_REPOSITORY:main$IMAGE_TAG
homeworks/K33402/AZaytseva/6/main
31.     docker push cr.yandex/$CR_REGISTRY/$CR_REPOSITORY:main$IMAGE_TAG
32.     docker build -t cr.yandex/$CR_REGISTRY/$CR_REPOSITORY:stat$IMAGE_TAG
homeworks/K33402/AZaytseva/6/stat-service
33.     docker push cr.yandex/$CR_REGISTRY/$CR_REPOSITORY:stat$IMAGE_TAG
34.
35.     # Запускаем собранные образы в виртуальной машине
36.     - name: Deploying images to VM
37.       uses: appleboy/ssh-action@v0.1.10
38.       env:
39.         CR_REGISTRY: crpf3tvmfa4lot7gl4jg
40.         CR_REPOSITORY: backend
41.         IMAGE_TAG: ${ github.sha }
42.       with:
43.         host: ${ secrets.HOST }
44.         username: ${ secrets.USERNAME }
45.         key: ${ secrets.KEY }
46.         port: ${ secrets.PORT }
47.         script: script: sudo docker kill $(docker ps -q) && /dev/null ; sudo
docker rm $(docker ps -a -q) && /dev/null ; sudo docker run -dp 8000:8000
cr.yandex/crpf3tvmfa4lot7gl4jg/backend:main${ github.sha } && sudo docker run --
env MAIN_HOST=localhost --env MAIN_PORT=8000 -dp 8008:8008
cr.yandex/crpf3tvmfa4lot7gl4jg/backend:stat${ github.sha }

```

Созданный пайплайн можно запустить вручную или осуществив пуш в ветку hw6. Попробуем изменить комментарий в файле Github Action и проверим работу созданного пайплайна.

chiclassie / ITMO-ICT-Backend-2023 Public

forked from kantegory/ITMO-ICT-Backend-2023

<> Code Pull requests Actions Projects Wiki Security Insights Settings

← Docker deploy to Yandex Container Registry

Update docker.yml #4

Summary

Jobs

- build

Run details

- Usage
- Workflow file

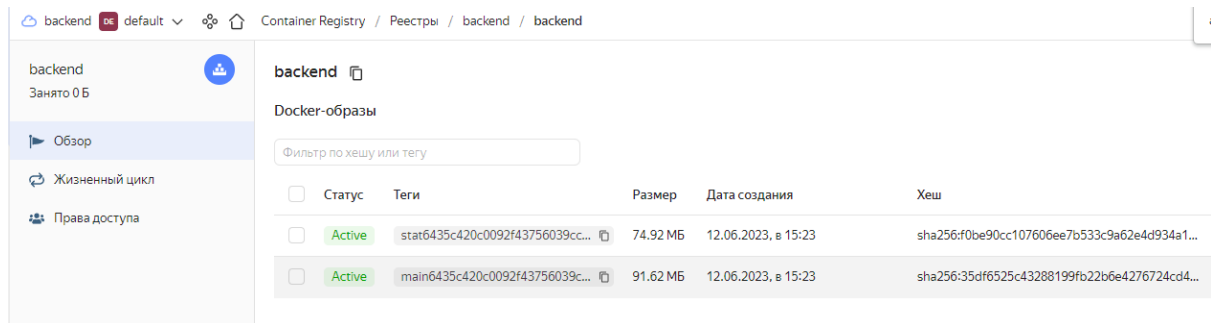
**build**

succeeded 1 minute ago in 1m 55s

- > Set up job
- > Run actions/checkout@v2
- > Login to Yandex Cloud Container Registry
- > Build, tag, and push image to Yandex Cloud Container Registry
- > Post Login to Yandex Cloud Container Registry
- > Post Run actions/checkout@v2
- > Complete job

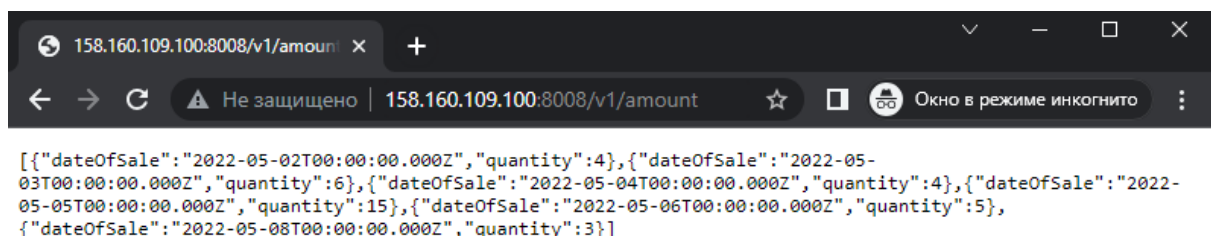
Пайплайн прошел успешно. Он доступен по ссылке <https://github.com/chiclassie/ITMO-ICT-Backend->

[2023/actions/runs/5244832393/jobs/9471386921](https://2023/actions/runs/5244832393/jobs/9471386921). Теперь проверим Yandex Container Registry на наличие собранных образов:



Для проверки работы запущенных образов в виртуальной машине перейдем по адресу <http://158.160.109.100:8008/v1/amount>, где

- 158.160.109.100 – это публичный ip адрес созданной виртуальной машины;
- 8008 – порт, на котором запущен микросервис статистики;
- /v1/amount – эндпоинт, возвращающий статистику по количеству проданных товаров.



## Вывод

В рамках шестой практической работы я освоила технологию Github Actions, а также познакомилась с некоторыми сервисами Yandex Cloud, а именно Yandex Container Registry, Yandex Cloud Computing, Yandex CLI. Мне удалось настроить пайплайн Github Actions для полной автоматизации процесса сборки образов из ветки репозитория, загрузки их в Yandex Container Registry и запуске обновленных образов в виртуальной машине.