

## Самостоятельная работа

1. Скопируйте репозиторий с практикой на машину с Docker и перейдите в директорию: `3.docker-compose_practice`

```
git clone https://gitlab.slurm.io/edu/docker.git
```

```
cd docker/3.docker-compose_practice
```

2. Самостоятельно напишите `Dockerfile` и `docker-compose.yml` файлы для сборки и запуска приложения в директории.

- В `Dockerfile` должны устанавливаться зависимости из файла `requirements.txt` (`pip install -r requirements.txt`)
- Само приложение должно запускаться командой `python setup.py && flask run --host=0.0.0.0 --port 8000`
- Приложение должно быть доступно на `8000` порту
- Приложение должно работать от имени пользователя `app`
- Рабочей директорией в контейнере должен быть путь `/app`
- В `Dockerfile` также необходимо передавать дополнительную переменную для работы приложения - `FLASK_APP=src/app.py`
- В `docker-compose.yml` файле должно быть два сервиса - `db` (`postgres`) и `app`.
- Приложение должно собираться автоматически при выполнении команды `docker-compose up`
- Приложение должно запускаться из локальных файлов проекта (то есть нужно монтировать директорию с кодом в рабочую директорию контейнера с приложением)
- Для работы приложения ему нужно передать следующие переменные:
  - `DB_NAME`
  - `DB_USER`
  - `DB_PASSWORD`
  - `DB_HOST`
  - `FLASK_ENV: development`

3. Приложение должно уметь сохранять пользователей с `lastName` и `firstName` и выводить данные пользователя по запросу с `id`.

Примеры запросов:

```
curl localhost:8000/api/v1/users -X POST -d  
'{"firstName":"John","lastName":"Cena"}' --header "Content-Type:  
application/json"
```

```
curl localhost:8000/api/v1/users/1
```

4. Создайте тестового пользователя, скопируйте ответ приложения и отправьте <sup>+2</sup> в качестве решения данного шага.