

# Тестовое задание для стажера на позицию «DevOps»

1. Написать скрипт под ОС Linux, выполняющий проверку свободного места корневого раздела диска. В случае использования более 85% - прислать алерт на email. Скриптовый язык использовать на свое усмотрение. Имя почтового ящика (адрес) должен задаваться через переменную/параметр.

1.1 Создать запись Cron для ежедневного запуска скрипта.

2. Реализовать автоматизированную сборку библиотеки под Linux на примере SQLite.

2.1 Скачать архив <https://www.sqlite.org/2018/sqlite-amalgamation-3260000.zip>

2.2 Изучить руководство по сборке sqlite <https://www.sqlite.org/howtocompile.html>

2.3 Написать CMakeLists.txt для компиляции исходников под Linux (x86/x86\_64) – только со при помощи gcc.

2.4 Запустить процесс компиляции. Если в процессе возникнут ошибки – желательно описать процесс их решения в README.

3. Написать конфиг для Nginx, публикующий сервис из локальной сети в интернет. Nginx выступает как reverse-проxy.

3.1 IP адрес сервиса в локальной сети: 10.0.0.10, порт сервиса TCP 5000, протокол http.

3.2 Nginx «слушает» все интерфейсы на порту 80 (http). И все запросы, приходящие на доменное имя devops.ru – перенаправляет на вышеуказанный сервер в локальную сеть.

4. Подготовить bash скрипт, который запускает компиляцию задачи из пункта 2 и создает файл лога процесса. В имени файла лога должна быть дата его создания.

5. Написать dockerfile и собрать docker image, который включает в себя приложение и веб-сервер Nginx, при запуске контейнера открыть порт 8080. В качестве приложения можно взять статичную html страницу.

5.1 Написать docker-compose файл для запуска контейнера на базе образа из пункта 5. Имя контейнера задать – my-devops. Порт публикации приложения – 80. Порт публикации должен задаваться через файл настроек (переменных окружения).

6. Есть условное приложение Node.js. Приложение работает в docker контейнере. Dockerfile написан неправильно/неэффективно. Необходимо переписать dockerfile в соответствии с best-practice.

6.1 Dockerfile:

```
FROM ubuntu:18.04
COPY ./src /app
RUN apt-get update -y
RUN apt-get install -y nodejs
RUN npm install
```

```
ENTRYPOINT ["npm"]  
CMD ["run", "prod"]
```

Пункты 1-4 обязательны для выполнения, пункты 5-6 выполняются по желанию.

#### **Требования к присылаемым решениям:**

Для проверки необходимо передать zip-архив, содержащий:

1. Код скрипта. Запись Cron в текстовом файле cron.txt
2. CMakeLists.txt, лог компиляции, файл readme с описанием проблем/их решения (при его наличии).
3. Конфиг файл для Nginx.
4. Код скрипта.
5. Dockerfile, docker-compose файл.
6. Dockerfile (исправленный вариант).

**Максимальное время на выполнение задания:** 1 неделя.