



Общество с ограниченной ответственностью «Ребреин»

ИНН 7727409582, ОГРН 1197746106161

Адрес: 123056, город Москва, Большая Грузинская ул, д. 36а стр. 5а, офис 13

Утверждено

Приказом № ПР-1 от 17.06.2025 г.

Генеральный директор

 Фролкина Е.А.  
«17» июня 2025 г.

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА  
– ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ  
**«КОНТЕЙНЕРИЗАЦИЯ И УПРАВЛЕНИЕ ОКРУЖЕНИЕМ С ПОМОЩЬЮ DOCKER»**

**Срок реализации:** 2 месяца

**Количество часов:** 133 акад. ч.

**Форма обучения:** заочная форма

**Формат обучения:** с применением  
исключительно дистанционных технологий

**Возраст обучающихся:** для лиц старше 17  
лет, имеющих или получающих среднее  
профессиональное и (или) высшее  
образование

Москва, 2025 г.

## **1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Настоящая дополнительная профессиональная программа – программа повышения квалификации «Контейнеризация и управление окружением с помощью Docker» (далее – Программа) разработана в соответствии с:

- Федеральным законом от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказом Министерства образования и науки РФ от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- Постановлением Правительства РФ от 11.10.2023 № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- Профессиональным стандартом 06.026 «Системный администратор информационно-коммуникационных систем», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 26.10.2020 года № 60580;
- ФГОС высшего образования – бакалавриат по направлению 09.03.02 Информационные системы и технологии, утв. приказом Минобрнауки России от 19.09.2017 №926;
- Локальными нормативными актами ООО «Ребреин».

В данной программе учтены основные идеи формирования универсальных учебных действий учащихся и соблюдена преемственность с программами высшего и/или среднего профессионального образования.

**Направленность программы:** Программа имеет техническую направленность.

**Адресат:**

- Разработчики программного обеспечения
- DevOps-инженеры
- Системные администраторы
- QA-специалисты, участвующие в DevOps-процессах
- Системные архитекторы

**Требования к входным знаниям обучающегося:**

- Базовые знания Linux
- Основы работы с командной строкой и терминалом
- Базовые знания сетевых протоколов и приложений
- Понимание концепций виртуализации и клиент-серверной архитектуры

**Актуальность реализации:**

Использование Docker позволяет стандартизировать окружение для разработки, тестирования и эксплуатации приложений. Контейнеризация ускоряет развертывание сервисов, облегчает управление зависимостями и повышает масштабируемость приложений.

**Отличительные особенности программы:**

- Практическая ориентация с большим количеством лабораторных работ и примеров
- Освещение всех ключевых аспектов Docker: образы, контейнеры, сети, storage, compose, swarm
- Включение продвинутых тем и опциональных Темаов для профессиональной подготовки

**Объем и срок освоения программы:** 133 академ. ч. в течение 2 мес. (8 недель)

Доступ к материалам Программы у обучающихся остаётся и после окончания периода обучения. Это позволяет повторять изученный материал в удобное время, восполнять пробелы в знаниях, а также возвращаться к практическим заданиям при решении рабочих задач. Такой формат способствует более глубокому закреплению навыков и поддерживает профессиональное развитие выпускников даже после завершения обучения.

**Выдаваемый документ о квалификации:** удостоверение о повышении квалификации и/или сертификат об успешном освоении программы.

**Цели и задачи программы:**

Подготовить специалистов к эффективной работе с Docker для контейнеризации приложений, управления окружением и организации отказоустойчивых и масштабируемых сервисов.

**Программа направлена на решение следующих основных задач:**

Обучающие:

- Изучить базовые и продвинутые команды Docker
- Научиться создавать, настраивать и оптимизировать образы и контейнеры
- Освоить работу с Docker Compose и Docker Swarm

Развивающие:

- Развить навыки построения отказоустойчивых и масштабируемых решений
- Научиться оптимизировать время сборки и размер образов
- Развить способность анализировать и логировать работу сервисов

Воспитательные:

- Формирование культуры DevOps-подхода
- Развитие системного мышления при проектировании контейнеризированных приложений
- Привитие привычки использовать стандартизованные инструменты для командной работы

**Планируемые результаты:**

Знания

- Понимание концепции контейнеризации и архитектуры Docker
- Знание принципов построения и оптимизации Docker-образов
- Знание инструментов управления зависимостями, сетями, хранилищем и логами

Умения

- Настраивать и запускать контейнеры и образы
- Использовать Docker Compose и Swarm для организации многоконтейнерных и многонодовых систем
- Настраивать постоянное хранилище и логирование сервисов

Навыки

- Работа с Docker CLI и Dockerfile
- Создание и управление контейнерами, сетями и томами
- Оптимизация образов и сборки, настройка restart policy и capabilities

- Мониторинг и отладка контейнеризированных приложений

**Перечень профессиональных компетенций, на получение которых направлено обучение:**

На основе профстандарта 06.026 «Системный администратор информационно-коммуникационных систем»:

- В/02.5 Обеспечение работы технических и программных средств информационно-коммуникационных систем;
- С/05.6 Выполнение обновления программного обеспечения сетевых устройств информационно-коммуникационных систем;
- С/08.6 Планирование и проведение работ по распределению нагрузки между имеющимися ресурсами, снятию нагрузки на сетевые устройства информационно-коммуникационных систем перед проведением регламентных работ, восстановлению штатной схемы работы в случае сбоев.

Таким образом, в результате освоения программы у обучающихся формируются следующие профессиональные компетенции:

- ОПК-5. Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;
- ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий;
- ОПК-7. Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем.

**Организационно-педагогические условия реализации программы дополнительного профессионального образования**

**Язык реализации образовательной программы:** обучение проводится на русском языке.

**Форма обучения:** заочная форма.

**Особенности реализации программы:** программа реализуется с использованием электронного обучения и исключительно дистанционных образовательных технологий.

**Условия набора:** на обучение принимаются все желающие лица, оплатившие обучение и заключившие договор об образовании. Обучение проходит в индивидуальном формате без формирования учебных групп. Обучающийся самостоятельно определяет время освоения Программы.

**Формы проведения занятий:**

- занятия в текстовом формате;
- практическая работа;
- самостоятельная работа с литературой;
- индивидуальные вопросы.

**Материально-техническое оснащение**

**Материальное обеспечение программы**

Занятия проводятся в системе дистанционного обучения «Rebrain». Каждый обучающийся и педагог оснащены доступом к системе дистанционного обучения: <https://rebrainme.com/>.

У педагога дополнительного профессионального образования имеется необходимое оборудование средства для реализации программы: ноутбук с подключением к интернету, программное обеспечение.

### **Методическое обеспечение программы**

Программа обеспечена:

- учебно-методическими материалами (текстовые занятия, полезными материалами);
- практическими заданиями.

### **Кадровое обеспечение:**

К реализации программы в качестве педагогов дополнительного образования допускаются лица:

- 1) отвечающее одному из требований:
  - a) имеющее высшее образование или среднее профессиональное образование в рамках укрупненных групп специальностей и направлений подготовки высшего образования и специальностей среднего профессионального образования «Образование и педагогические науки»;
  - b) имеющее высшее образование либо среднее профессиональное образование в рамках иных укрупненных групп специальностей и направлений подготовки высшего образования и специальностей среднего профессионального образования при условии его соответствия дополнительной общеобразовательной общеразвивающей программе, реализуемой ООО «Ребреин», и получение при необходимости дополнительного профессионального образования педагогической направленности;
  - в) успешно прошедшее промежуточной аттестации не менее чем за два года обучения по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующей направленности дополнительной общеобразовательной общеразвивающей программе;
- 2) не имеющее ограничений на занятие педагогической деятельностью, установленных законодательством Российской Федерации;
- 3) прошедшее обязательный предварительный (при поступлении на работу) и периодические медицинские осмотры (обследования), а также внеочередные медицинские осмотры (обследования) в порядке, установленном законодательством Российской Федерации.

Реализация Программы также возможна лицами, привлекаемыми на условиях гражданско-правового договора в соответствии с действующим законодательством РФ.

## **2. УЧЕБНЫЙ ПЛАН**

№ п/ п	Наименование модуля	Количество часов			Формы контроля / аттестация
		Всего	Теория	Практика	
1	Модуль 1. Онбординг	2	1	1	Входное тестирование
2	Модуль 2. Docker	120	40	80	Практическое задание
3	Итоговая аттестация	11		11	Итоговое практическое задание

### **3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК**

<b>№ п/ п</b>	<b>Наименование модуля</b>	<b>1 неделя</b>	<b>2 неделя</b>	<b>3 неделя</b>	<b>4 неделя</b>	<b>5 неделя</b>	<b>6 неделя</b>	<b>7 неделя</b>	<b>8 неделя</b>
1	Модуль 1. Онбординг	2							
2	Модуль 2. Docker	14	17	17	17	17	17	17	4
3	Итоговая аттестация								11   A

### **4. РАБОЧАЯ ПРОГРАММА**

#### **Модуль 1. Онбординг**

Теория 1 академ. ч. Практика 1 академ. ч.

Модуль состоит из следующих тем:

##### **Тема 1: Онбординг**

В модуле обучающемуся предоставляется вводный конспект, содержащий общую информацию о программе, структуре курса, форматах взаимодействия с материалами и ожидаемых результатах обучения.

Предусмотрено прохождение входного тестирования, включающего 7 вопросов, направленных на закрепление информации из онбординга. В рамках темы обучающийся выполняет задание по целеполаганию: формулирует свою цель прохождения программы, указывает желаемые навыки по окончании обучения, а также оценивает текущий уровень своих знаний по DevOps, выбрав один из предложенных вариантов.

#### **Модуль 2. Docker**

Теория 40 академ. ч. Практика 80 академ. ч.

Модуль состоит из следующих тем:

##### **Тема 1: Basics. Знакомство с Docker**

Содержание: Основы Виртуализации. Внутреннее устройство. Установка Docker. Практическое задание.

##### **Тема 2: Basics. Флаги запуска**

Содержание: Основные флаги запуска контейнеров. Настройка работы контейнера через CLI. Практическое задание.

##### **Тема 3: Basics. Запуск команд внутри контейнеров**

Содержание: Запуск команд в работающих контейнерах. Использование docker exec. Практическое задание.

##### **Тема 4: Basics. Внешнее хранилище**

Содержание: Работа с томами (Volumes) и bind-монтированием. Подключение внешнего хранилища. Практическое задание.

##### **Тема 5: Basics. Остановка, удаление контейнеров**

Содержание: Команды остановки, удаления и очистки контейнеров. Практическое задание.

## Тема 6: Basics. Логирование

Содержание: Просмотр логов контейнеров (docker logs). Настройка уровней логирования.  
Практическое задание.

## Тема 7: Images. Введение в Docker-образы

Содержание: Понятие образа. Работа с публичными и локальными образами. Практическое задание.

## Тема 8: Images. Введение в Dockerfile

Содержание: Назначение Dockerfile. Создание первого Dockerfile. Практическое задание.

## Тема 9: Images. Основные директивы Dockerfile

Содержание: FROM, RUN, COPY, CMD, ENTRYPOINT, WORKDIR. Практическое задание.

## Тема 10: Images. Параметризация Dockerfile

Содержание: ARG и ENV. Передача параметров сборки. Практическое задание.

## Тема 11: Images. Введение в понятие «слои»

Содержание: Структура образа. Слои и кэширование. Практическое задание.

## Тема 12: Images. Удаление образов

Содержание: Команды очистки образов (docker rmi). Практическое задание.

## Тема 13: Images. Кэширование

Содержание: Механизм кэширования слоев при сборке. Оптимизация сборки.  
Практическое задание.

## Тема 14: Images. Multistage

Содержание: Многоступенчатая сборка образов. Практическое задание.

## Тема 15: Images. Multistage. Advanced

Содержание: Продвинутые приёмы multistage для уменьшения объема и оптимизации.  
Практическое задание.

## Тема 16: Images. Использование Multistage для генерации нескольких образов

Содержание: Создание нескольких артефактов через multistage. Практическое задание.

## Тема 17: Images. Использование нескольких Dockerfile

Содержание: Разделение Dockerfile для разных целей (dev, prod). Практическое задание.

## Тема 18: Images. Минимизация объема образа

Содержание: Использование легковесных базовых образов, удаление лишних слоев.  
Практическое задание.

## Тема 19: Images. Внешние Docker-реестры

Содержание: Работа с Docker Hub, регистрация и push/pull образов. Практическое задание.

## Тема 20: Images. Минимизация времени сборки образа

Содержание: Оптимизация порядка инструкций, использование кэша. Практическое задание.

## Тема 21: Images. Dockerignore

Содержание: Настройка .dockerignore. Исключение лишних файлов из сборки.  
Практическое задание.

## Тема 22: Images. Различия базовых образов

Содержание: Alpine, Debian, Ubuntu, BusyBox. Практическое задание.

## Тема 23: Advanced. Restart policy

Содержание: Настройка политики перезапуска контейнеров (always, on-failure).  
Практическое задание.

## Тема 24: Advanced. Capabilities

Содержание: Назначение и управление Linux capabilities для контейнеров. Практическое задание.

## Тема 25: Advanced. Сети

Содержание: Bridge, Host, Overlay сети. Настройка сетевого взаимодействия контейнеров.  
Практическое задание.

## Тема 26: Advanced. Docker in Docker

Содержание: Запуск Docker внутри контейнера. Практическое задание.

## Тема 27: Compose. Знакомство с Docker compose

Содержание: Основы Docker Compose. Структура docker-compose.yml. Практическое задание.

## Тема 28: Compose. Запуск зависимого сервиса

Содержание: Связывание контейнеров через Compose. Практическое задание.

## Тема 29: Compose. Опции Compose-файла

Содержание: Настройка переменных среды, томов, сетей, restart policy. Практическое задание.

## Тема 30: Compose. Запуск Compose со сборкой образа

Содержание: Автоматическая сборка образа при запуске docker-compose up --build.  
Практическое задание.

## Тема 31: Compose. Хранение логов в EFK

Содержание: Настройка хранения логов контейнеров в Elasticsearch, Fluentd, Kibana.  
Практическое задание.

## Тема 32: Server. Insecure Registry

Содержание: Работа с небезопасными реестрами, настройка TLS и авторизации.  
Практическое задание.

## Тема 33: Swarm. Знакомство с Docker Swarm-режимом

Содержание: Основы кластеризации Swarm, архитектура. Практическое задание.

## Тема 34: Swarm. Многонодовый Docker Swarm-кластер

Содержание: Инициализация и подключение узлов. Практическое задание.

## Тема 35: Swarm. Многонодовый Docker Swarm-кластер

**Содержание:** Продвинутые настройки кластера и управление сервисами. Практическое задание.

**Тема 36: Swarm. Запускаем доступный извне сервис**

**Содержание:** Публикация сервисов Swarm на внешнем интерфейсе. Практическое задание.

**Тема 37: Swarm. Labels**

**Содержание:** Использование меток для управления развертыванием сервисов. Практическое задание.

**Тема 38: Swarm. Placement**

**Содержание:** Контроль размещения сервисов на узлах кластера. Практическое задание.

**Тема 39: Swarm. Построение постоянного хранилища для кластера на базе GlusterFS**

**Содержание:** Настройка распределенного хранилища для контейнеров. Практическое задание.

**Тема 40: Swarm. Собираем логи кластера через EFK**

**Содержание:** Централизованное логирование сервисов кластера с использованием EFK. Практическое задание.

Каждая тема модуля включает текстовое занятие с теоретическим материалом и пошаговыми инструкциями, после изучения которого предлагается практическое задание. Практические задания рассчитаны на 2 академических часа. Выполнение заданий предполагает отправку решения на проверку через личный кабинет обучающегося. Критерии оценки прописаны в описании к каждому заданию. В случае корректного выполнения выставляется зачёт. Если работа содержит ошибки, задание возвращается на доработку. При повторной неудачной попытке (после двух доработок) обучающийся получает «незачёт».

### **Итоговая аттестация.**

Модуль посвящён выполнению финального практического задания без предварительного теоретического блока.

## **5. МЕТОДИЧЕСКИЕ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ**

Программа обеспечена системой дистанционного обучения <https://rebrainme.com/>.

Педагогические технологии:

- технология дифференцированного обучения;
- технология разноуровневого обучения;
- технология развивающего обучения;
- технология проблемного обучения;
- технология дистанционного обучения.

Методы обучения:

- словесный, наглядный практический;
- объяснительно – иллюстративный;
- частично-поисковый, исследовательский проблемный;
- игровой, дискуссионный.

## **Электронно-библиотечные ресурсы и системы, информационно-справочные системы:**

1. Научная электронная библиотека eLIBRARY.RU.
2. Собственные учебные материалы: <https://rebrainme.com/docker/>
3. Официальная документация Docker [Электронный ресурс]: <https://docs.docker.com/engine/docker-overview/>
- 4.

## **6. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ**

### **Оценочные материалы:**

Для отслеживания результатов освоения программы среди слушателей проводится текущий контроль, промежуточный контроль и итоговое оценивание.

### **Текущий контроль**

Осуществление текущего контроля проводится после занятий в виде написания практических заданий или тестирований. Тематика и условия выполнения практических заданий расписаны в личном кабинете обучающегося в СДО. Педагог проверяет решение и принимает решение о принятии решения (зачет), о необходимости доработать решение или о незачете. Если промежуточный контроль представлен в виде тестирования, подсчет верных ответов и выставление оценки «зачёт» и «незачёт» происходят в автоматическим решиме в СДО.

### **Тема 6: Basics. Логирование**

#### **Задание**

1. Запустите контейнер со следующими параметрами:
  - должно работать в фоне,
  - слушает на хосте 127.0.0.1:8892,
  - имеет имя rbm-dkr-06-local,
  - образ - nginx:stable,
  - логи должны использовать драйвер local и объем файла лога не должен превышать 10 MiB.
2. Выведите список запущенных контейнеров.
3. Один раз обратитесь к запущенному nginx, чтобы были записаны логи (например, curl --silent http://127.0.0.1:8892 > /dev/null).
4. Выполните вывод содержимого файла на хостовой системе, в который записаны логи контейнера.
5. Настройте глобальное сохранение логов с драйвером local и объемом логов в 10 MiB через файл /etc/docker/daemon.json.
6. Запустите контейнер со следующими параметрами:
  - должно работать в фоне,
  - слушает на хосте 127.0.0.1:8893,
  - имеет имя rbm-dkr-06-global,
  - образ - nginx:stable,
  - в команде запуска НЕ должны присутствовать параметры драйвера.
7. Выведите список запущенных контейнеров.
8. Один раз обратитесь к запущенному nginx, чтобы были записаны логи (например, curl --silent http://127.0.0.1:8893 > /dev/null).
9. Выполните вывод содержимого файла на хостовой системе, в который записаны логи контейнера.
10. Отправьте решение на проверку.

## **Тема 18: Images. Минимизация объема образа**

### **Задание**

Мы изучили различные способы уменьшения образов, которые будут постоянно встречаться вам в реальной работе.

В качестве задания на примере репозитория awesome-compose предлагаем ознакомиться с best-practice по сборке различных языков программирования

1. Склонируйте репозиторий на исходную машину
2. Изучите Dockerfile для сборки
  - django/app и соберите, образ с тегом django:django (Для сборки проекта необходимо включить buildkit, как это сделать можно подсмотреть в "Полезных ссылках")
  - vuejs/vuejs и соберите, образ с тегом vuejs:vuejs (В процессе сборки вы можете столкнуться с ошибками совместимости версий пакетов, добавьте к команде yarn install флаг --ignore-engines)
  - react-java-mysql/backend и соберите, образ с тегом backend:java
  - react-java-mysql/frontend и соберите, образ с тегом frontend:react
3. Отправьте задание на проверку

## **Тема 37: Swarm. Labels**

### **Задание**

1. В репозитории dkr-30-voting из предыдущих заданий создайте новую ветку с именем dkr-38-labels.
2. Измените docker-compose.yml файл, убрав routing mesh.
3. Создайте в репозитории второй файл /home/user/dkr-30-voting/traefik-compose.yml, в котором опишите стек для запуска Traefik.
4. Запустите стек traefik (Traefik должен принимать запросы по 80 и 443 порту и перенаправлять их на сервисы внутри swarm).
5. Добавьте в /home/user/dkr-30-voting/docker-compose.yml лейблы для приема трафика через traefik на домене из nip.io (Чтобы автопроверки корректно отработали, будем использовать адреса вида voting.x.x.x.x.nip.io , kibana.x.x.x.x.nip.io. Сервис nip.io будет разрешить их в ip-адрес x.x.x.x).
6. Запустите новый стек из /home/user/dkr-30-voting/docker-compose.yml.
7. Выведите список стеков.
8. Выведите список сервисов.

[...]

### **Итоговое оценивание**

В конце программы обучающиеся сдают итоговую аттестацию.

### **Финальный презентационный проект**

### **Правила презентации проекта**

Доброго дня! Рады, что вы успешно завершаете практикум Docker. Рассказываем, каким образом мы будем проверять, как вы усвоили материал:

1. Вы выбираете удобное вам время для итоговой встречи, укажите в ответ на это задание три удобных вам таймслота длительностью 2 часа (желательно в разные дни) и удовлетворяющие таким условиям:
  - не менее чем через 3 дня (минимальный срок для подготовки проекта, рекомендуем брать неделю)
  - не более 20 дней от текущей даты
  - находится в таймслоте с 8:00 до 19:00 по МСК (можем рассматривать выходные дни)

- Если какой-то форс-мажор, напишите нам и мы перенесём дату презентации ;)

Мы сверим указанные вами слоты с нашим расписанием собеседований, подтвердим один из них.

2. Вы выполняете задание, а также готовите мини-презентацию о своем решении - почему сделали именно так, какие еще варианты решений рассматривали и т.д. (мы не требуем обязательного наличия слайдов, просто подготовьте небольшой доклад)
3. При подтверждении даты презентации, мы вместе с заданием отправляем вам ссылку на Zoom митинг, на запланированную дату.

### **Этапы презентации и правила оценивания**

Идентификация (вы показываете свой паспорт или другое удостоверение личности)

1 этап. Запуск, презентация и защита вашего проекта. Оценивается до 20 баллов. Во время презентации вам необходимо:

- Рассказать какие решения вы принимали во время реализации проекта
- С какими сложностями сталкивались и как их решили
- Ответить на вопросы куратора

2 этап - 3 вопроса по практикуму Docker. За каждый вы получите от 0 до 10 баллов.

- 0 баллов - если ответили неверно.
- 4 балла - если направление мысли (поиска ответа) определено с помощью подсказки кураторов или дополнительных источников ( google / etc ), но вы не сказали правильный ответ.
- 7 баллов - если направление мысли (поиска ответа) определено без помощи подсказки кураторов или дополнительных источников, но вы не сказали правильный ответ.
- 9 баллов - если направление мысли (поиска ответа) определено без помощи дополнительных источников, но правильный ответ найден с помощью дополнительных источников ( google / etc ) или подсказки кураторов.
- 10 баллов - если полностью самостоятельно раскрыли тему.

Все баллы суммируются. Максимальное количество баллов за встречу - 50.

Сертификат будет доступен в вашем профиле, в течении некоторого времени после защиты проекта (при условии успешной защиты).

Если вы не проходите собеседование, или не удовлетворены набранными баллами, то следующая попытка будет вам предоставлена не ранее, чем через 1 календарный месяц.

### **Задание**

В репозитории находится приложение, запуск которого необходимо выполнить для данного задания, соблюдая правила и условия указанные ниже в соответствующих разделах.

Данные для переменных указанных в теле задания:

Вы можете выбрать любой вариант, который вам понравится. [...]

### **Требования к реализации:**

#### **К запущенному приложению:**

- Базовый образ - "base\_image" (значение указано выше)
- Приложение должно быть запущено в frontend режиме с хранением телеметрии, обfuscацией ID и без редактирования IP адресов
- Конфигурация должна производиться исключительно через переменные окружения
- Отправлять логи в EFK средствами Docker
- Образ должен собираться из docker-compose файла

- Конфигурационный файл серверов должен подключаться как volume

К Dockerfile:

- Образ не должен содержать конфигурационный файл серверов
- Образ должен кешировать как можно больше при минимальном количестве шагов

[...]

Результаты текущего контроля и итогового оценивания отображаются в личном кабинете слушателя в системе дистанционного обучения <https://rebrainme.com/>.

По результатам сдачи текущего контроля и итогового оценивания педагог даёт обратную связь слушателям, отмечает их сильные стороны и обращает внимание на зоны для развития. При необходимости педагог может повторить пройденные темы со слушателями, если установлен факт плохого закрепления и усвоения темы у слушателей.