

Общество с ограниченной ответственностью «Ребрейн»  
ИНН 7727409582, ОГРН 1197746106161

---

Утверждено  
Приказом № ПР-1 от 01.06.2025 г.  
Генеральный директор  
Фролкина Е.А.  
«01» июня 2025 г.



**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА  
– ПРОГРАММА ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ  
«DEVOPS-ИНЖЕНЕР»**

**Срок реализации:** 14 месяцев  
**Количество часов:** 375 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

## 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Настоящая дополнительная профессиональная программа – программа профессиональной переподготовки «DevOps-инженер» (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 «Системный администратор информационно-коммуникационных систем», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 29 сентября 2020 г. № 680н;
- Профессиональным стандартом «Руководитель разработки программного обеспечения», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 20 июля 2022 г. № 423н;
- ФГОС высшего образования – бакалавриат по направлению 01.03.02 Прикладная математика и информатика, утв. приказом Минобрнауки России от 10.01.2018 № 9.

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:**

Программа рассчитана на обучающихся старше 17 лет, имеющих или получающих среднее профессиональное и (или) высшее образование.

Обучающийся на момент зачисления на Программу должен обладать базовыми знаниями LINUX, NGINX, сетевых протоколов.

### **Актуальность реализации:**

Переход компаний к облачным технологиям и микросервисной архитектуре требует высокой автоматизации процессов разработки и развертывания. DevOps-инженеры играют ключевую роль в обеспечении стабильности, масштабируемости и безопасности ИТ-инфраструктуры. Спрос на специалистов с компетенциями в Git, Docker, Kubernetes, CI/CD и системах мониторинга продолжает расти. Программа ориентирована на практическое освоение инструментов DevOps и отвечает требованиям современного ИТ-рынка. Завершение обучения позволяет участникам сразу применять полученные навыки в реальных проектах.

### **Отличительные особенности программы:**

Отличительными особенностями программы являются её реализация с применением дистанционных образовательных технологий и практико-ориентированный формат обучения. Программа включает систему взаимосвязанных модулей, охватывающих весь жизненный цикл DevOps-процессов — от разработки до мониторинга. Образовательный

процесс строится на использовании интерактивного контента, способствующего более качественному усвоению материала. Для самостоятельной отработки навыков предусмотрен банк практических заданий, а также финальное задание, максимально приближенное к реальным кейсам. Такой подход позволяет активизировать познавательную деятельность обучающихся и обеспечивает высокий уровень вовлечённости.

**Объем и срок освоения программы:** 375 академ. ч. в течение 14 мес.

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:**

Обучающиеся успешно прошедшие Программу и сдавшие итоговую аттестацию вправе получить диплом о профессиональной переподготовке установленного ООО «Ребрейн» образца.

**Цели и задачи программы:**

Цель дополнительной профессиональной программы «DevOps-инженер» – формирование у обучающихся комплекса актуальных теоретических знаний и практических навыков, необходимых для автоматизации процессов разработки, тестирования, развертывания и сопровождения программных продуктов с использованием современных инструментов и методологий DevOps.

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

Обучающие:

- Сформировать у обучающихся понимание принципов и процессов DevOps как культурной и технической практики в современной разработке программного обеспечения.
- Обучить работе с системами контроля версий (Git) и инструментами контейнеризации (Docker) как основами автоматизации и масштабируемости.
- Освоить принципы и практики непрерывной интеграции (CI), доставки (CD) и мониторинга (Observability) с использованием современных инструментов (GitLab, Kubernetes и др.).
- Развить навыки проектной деятельности через выполнение финального практического задания, приближенного к реальным задачам DevOps-инженера.

Развивающие:

- Сформировать у обучающихся способность к самостоятельному освоению новых технологий и инструментов в сфере DevOps.
- Развить аналитическое мышление для принятия обоснованных технических решений при работе с системами автоматизации.
- Повысить уверенность в работе с профессиональными инструментами и средами, используемыми в DevOps-практике.
- Сформировать устойчивые навыки самоконтроля, самоорганизации и эффективного планирования индивидуальной учебной траектории.

Воспитательные:

- Формировать ответственное отношение к качеству и безопасности разрабатываемых и сопровождаемых ИТ-продуктов, понимание значимости своей роли в процессе разработки.
- Воспитывать стремление к профессиональному развитию, непрерывному обучению и следованию современным стандартам и этике в сфере информационных технологий.

### **Планируемые результаты:**

Планируемыми результатами обучения по программе является приобретение следующих знаний, умений, навыков, участвующих в качественном изменении компетенций:

#### **Знания:**

- основные принципы и подходы DevOps, включая практики CI/CD и инфраструктуру как код;
- возможности и особенности работы с Git, Docker, GitLab CI, Kubernetes и инструментами мониторинга;
- этапы жизненного цикла разработки программного обеспечения в контексте DevOps-подхода.

#### **Умения:**

- использовать системы контроля версий для управления исходным кодом;
- создавать и использовать контейнеры с помощью Docker;
- настраивать процессы непрерывной интеграции и доставки с применением GitLab CI и других инструментов;
- применять базовые команды и манифесты Kubernetes для управления кластером;
- реализовывать мониторинг и логирование сервисов для оценки стабильности и производительности системы.

#### **Навыки:**

- самостоятельного освоения новых инструментов и технологий DevOps;
- анализа и оптимизации процессов разработки и развертывания;
- применения полученных знаний для решения практических задач и кейсов, приближенных к реальным условиям ИТ-проектов.

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

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

Обобщенные трудовые функции			Трудовые функции		
код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
А	Технические работы по обслуживанию информационно-коммуникационной системы	4	Выполнение работ по выявлению и устранению типичных инцидентов информационно-	А/01.4	4

			коммуникационных систем		
			Выполнение работ по управлению стандартными изменениями в технических и программных средствах информационно-коммуникационных систем по инструкции	A/02.4	4
В	Обслуживание информационно-коммуникационной системы	5	Выполнение работ по выявлению и устранению инцидентов в информационно-коммуникационных системах	B/01.5	5
			Обеспечение работы технических и программных средств информационно-коммуникационных систем	B/02.5	5
			Реализация схемы резервного копирования, архивирования и восстановления конфигураций технических и программных средств информационно-коммуникационных систем по утвержденным планам	B/03.5	5
			Внесение изменений в технические и программные средства информационно-коммуникационных систем по утвержденному плану работ	B/04.5	5
			Проведение обновления программного обеспечения технических средств информационно-коммуникационных систем	B/05.5	5

			систем по инструкциям производителей		
			Диагностика исчерпания типовых ресурсов информационно-коммуникационных систем с использованием прикладных программных средств и средств контроля	B/06.5	5
			Проведение предварительных испытаний при проведении работ с возможными рисками перерывов в предоставлении сервисов информационно-коммуникационных систем	B/07.5	5
D	Обслуживание серверных операционных систем информационно-коммуникационной системы	6	Выполнение работ по выявлению и устранению нетипичных инцидентов, возникающих в серверных операционных системах информационно-коммуникационной системы	D/01.6	6
			Проведение анализа и определение основных причин сложных проблем, возникающих на серверах и в серверных операционных системах	D/02.6	6
			Выполнение планирования резервного копирования, архивирования и восстановления конфигурации серверов и серверных операционных систем	D/03.6	6

		Планирование изменений параметров работы серверов и серверных операционных систем	D/04.6	6
		Выполнение обновления программного обеспечения серверных операционных систем	D/05.6	6
		Прогнозирование влияния внешних и внутренних воздействий на поведение серверных операционных систем	D/06.6	6
		Планирование и проведение работ по распределению нагрузки между имеющимися ресурсами, снятию нагрузки на серверы и серверные операционные системы перед проведением регламентных работ, восстановлению штатной схемы работы в случае сбоев	D/08.6	6

На основе профстандарта «Руководитель разработки программного обеспечения»:

Обобщенные трудовые функции			Трудовые функции		
код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
В	Организация процессов разработки компьютерного программного обеспечения	7	Управление запросами на изменения, дефектами и проблемами в компьютерном программном обеспечении	В/04.7	7
			Управление конфигурациями и выпусками программного продукта	В/05.7	7

			Разработка внутренних правил, методик и регламентов проведения работ	В/06.7	7
С	Управление программно-техническими, технологическими и человеческими ресурсами для разработки компьютерного программного обеспечения	7	Управление инфраструктурой коллективной среды разработки компьютерного программного обеспечения	С/01.7	7
			Управление рисками разработки компьютерного программного обеспечения	С/02.1	7
			Управление процессами оценки сложности, трудоемкости, сроков выполнения работ	С/03.7	7

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

- Владение инструментами контроля версий (Git), включая работу с ветками, разрешение конфликтов и организацию рабочих процессов.
- Компетентность в контейнеризации приложений с использованием Docker и Docker Compose.
- Умение создавать и оптимизировать Docker-образы с применением многоуровневой сборки и минимизацией размера.
- Владение навыками настройки и эксплуатации кластеров Docker Swarm и основ Kubernetes.
- Компетентность в применении практик непрерывной интеграции с использованием GitLab CI.
- Знание современных подходов к автоматическому тестированию: юнит-, интеграционное, нагрузочное и безопасностное тестирование.
- Владение современными методами деплоя, включая использование Helm и реализацию стратегий Canary-деплой в Kubernetes.
- Способность организовывать мониторинг и логирование приложений с применением ELK/EFK, Sentry, DataDog и других решений.
- Понимание архитектуры и процессов различных языков программирования и фреймворков, включая Python, PHP, Golang, Java, Ruby и др.
- Компетентность в обеспечении наблюдаемости и управлении инцидентами с использованием современных платформ (например, PagerDuty).

**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:

а) 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»;

б) 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 «Ребрейн», 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.	Онбординг	5	4	1	Текущий контроль
2.	Git	50	16	34	Текущий контроль, промежуточный контроль
3.	Development	37	13	24	Текущий контроль
4.	Docker	98	32	66	Текущий контроль, промежуточный контроль
5.	CI: Gitlab	36	12	24	Текущий контроль
6.	Kubernetes Base	56	18	38	Текущий контроль, промежуточный контроль
7.	CI: QA	17	5	12	Текущий контроль, промежуточный контроль
8.	CD: Modern Way	35	11	24	Текущий контроль, промежуточный контроль
9.	Observability	30	10	20	Текущий контроль
10.	Финальное задание	11	1	10	Итоговая аттестация
	<b>Итого:</b>	375	122	253	

## 3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование модуля	1 мес	2 мес	3 мес	4 мес	5 мес	6 мес	7 мес	8 мес	9 мес	10 мес	11 мес	12 мес	13 мес	14 мес
1.	Онбординг	X													
2.	Git	X	X												
3.	Development			X											
4.	Docker				X	X	X								
5.	CI: Gitlab							X							

6.	Kubernetes Base								X	X					
7.	CI: QA										X				
8.	CD: Modern Way											X	X		
9.	Observability													X	X
10	Финальное задание														X

#### 4. РАБОЧАЯ ПРОГРАММА

##### Модуль 1. Онбординг

Теория 4 академ. ч. Практика 1 академ. ч.

Модуль состоит из следующих блоков изучения тем:

Блок 01: Онбординг

Блок 02: Рекомендации по прохождению

Блок 03: Введение

В модуле обучающемуся предоставляется вводный конспект, содержащий общую информацию о программе, структуре курса, форматах взаимодействия с материалами и ожидаемых результатах обучения, а также рекомендации по прохождению программы.

Предусмотрено прохождение входного тестирования, включающего 6 вопросов, направленных на закрепление информации из онбординга. В рамках блока 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. Git

Теория 16 академ. ч. Практика 34 академ. ч.

Модуль состоит из следующих блоков изучения тем:

Блок 01: Основы работы с Git

Блок 02: Игнорирование и удаление файлов в репозитории

Блок 03: История изменений

Блок 04: Изменение и отмена коммитов

Блок 05: Работа с ветками и тегами

Блок 06: Слияние веток

Блок 07: Решение конфликтов при слиянии веток

Блок 08: Слияние веток через «rebase»

Блок 09: Объединение коммитов

Блок 10: Перенос коммитов из одной ветки в другую

Блок 11: Работа с «git stash»

Блок 12: Работа с удалёнными репозиториями

Блок 13: Работа с командным репозиторием (code review), fork

Блок 14: Рабочие процессы: GitHub Flow, GitFlow

Блок 15: Best practices и README-файл

Блок 16: Подмодули (submodules)

Блок 17: Финальное задание

Модуль направлен на формирование у обучающихся базовых и продвинутых навыков работы с системой контроля версий Git. Каждый блок модуля включает текстовое занятие с теоретическим материалом и пошаговыми инструкциями, после изучения которого предлагается практическое задание. Исключение составляют блоки 14, 15 и 16, где выполнение практического задания не предусмотрено, а закрепление материала происходит через тестирование или самостоятельную работу. Блок 17 посвящён выполнению финального практического задания без предварительного теоретического блока.

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».

### **Модуль 3. Development**

Теория 13 академ. ч. Практика 24 академ. ч.

Модуль состоит из следующих блоков изучения тем:

Блок 01: Введение в разработку

Блок 02: C/C++: компилирование и библиотеки (configure, make install)

Блок 03: Golang: компилирование и запуск проекта

Блок 04: PHP basic: composer & php-fpm

Блок 05: PHP framework: настройка проекта Laravel

Блок 06: Python basics: requirements, install, standalone app

Блок 07: Python frameworks: Django

Блок 08: Ruby: gems, standalone app

Блок 09: Ruby: фреймворк Ruby on Rails

Блок 10: JavaScript: Node.js

Блок 11: Java: Intro

Блок 12: Java: Spring Framework

Блок 13: c# dot net

Модуль направлен на формирование у обучающихся представления о многообразии языков программирования и сред разработки, с которыми взаимодействует DevOps-инженер в процессе работы. В рамках модуля слушатели знакомятся с особенностями компиляции, запуска, сборки и конфигурации проектов на различных языках: C/C++, Golang, PHP, Python, Ruby, JavaScript, Java и C#.

Каждый блок содержит теоретический материал с пояснениями по ключевым концепциям и практическими примерами настройки и запуска приложений. Исключение составляет блок 01 «Введение в разработку», где обучение ограничивается обзорной теорией без практического задания. Остальные блоки сопровождаются практическими заданиями, направленными на отработку изученных инструментов и конфигурационных особенностей.

Практические задания рассчитаны на 2 академических часа каждое. Выполнение заданий осуществляется через личный кабинет обучающегося с обязательной отправкой на проверку. Критерии выполнения чётко определены в описании к каждому заданию. При корректном выполнении выставляется зачёт. В случае ошибок задание возвращается на доработку, и если после двух попыток результат остаётся неудовлетворительным, педагог выставляет «незачёт».

## Модуль 4. Docker

Теория 32 академ. ч. Практика 66 академ. ч.

Модуль состоит из следующих блоков изучения тем:

- Блок 01: Основы: знакомство с Docker
- Блок 02: Основы: флаги запуска
- Блок 03: Основы: запуск команд внутри контейнеров
- Блок 04: Основы: внешнее хранилище
- Блок 05: Основы: остановка, удаление контейнеров
- Блок 06: Основы: логирование
- Блок 07: Образы: введение в Docker-образы
- Блок 08: Образы: введение в Dockerfile
- Блок 09: Образы: параметризация Dockerfile
- Блок 10: Образы: введение в понятие «слои»
- Блок 11: Образы: удаление образов
- Блок 12: Образы: кэширование
- Блок 13: Образы: multistage
- Блок 14: Образы: минимизация объёма образа
- Блок 15: Образы: внешние Docker-реестры
- Блок 16: Образы: Dockerignore
- Блок 17: Advanced: политика перезапуска
- Блок 18: Advanced: Capabilities
- Блок 19: Advanced: сети
- Блок 20: Advanced: Docker in Docker
- Блок 21: Compose: знакомство с Docker compose
- Блок 22: Compose: запуск зависимого сервиса
- Блок 23: Compose: опции compose-файла
- Блок 24: Compose: запуск Compose со сборкой образа
- Блок 25: Compose: хранение логов в EFK
- Блок 26: Swarm: знакомство с Docker Swarm-режимом
- Блок 27: Swarm: многонодовый Docker Swarm-кластер
- Блок 28: Docker Context
- Блок 29: Swarm: запускаем сервис, доступный извне
- Блок 30: Swarm: Labels
- Блок 31: Swarm: построение постоянного хранилища для кластера на базе GlusterFS
- Блок 32: Swarm: собираем логи кластера через EFK
- Блок 33: Финальное задание

Модуль направлен на формирование у обучающихся устойчивых практических навыков работы с контейнеризацией на базе технологии Docker. В процессе изучения модуля слушатели последовательно осваивают базовые и продвинутые возможности Docker, включая работу с контейнерами, образами, Dockerfile, Compose и Swarm, а также взаимодействие с внешними реестрами и средствами логирования.

Каждый блок модуля включает текстовое занятие с теоретическими пояснениями, примерами команд и конфигураций. Основное внимание уделяется практической части — каждое задание направлено на закрепление материала путём выполнения типовых задач, имитирующих реальные сценарии использования Docker в профессиональной среде. Исключение составляет только блок 33, который представляет собой финальное практическое задание без теоретического сопровождения.

Практические задания выполняются обучающимся поэтапно и проверяются преподавателем через личный кабинет. Каждое задание рассчитано на 2 академических часа. Критерии оценки указаны в описании задания. При корректном выполнении выставляется зачёт. В случае наличия ошибок задание возвращается на доработку. Если после двух доработок задание остаётся нерешённым, преподаватель вправе выставить «незачёт».

## **Модуль 5. CI: Gitlab**

Теория 12 академ. ч. Практика 24 академ. ч.

Модуль состоит из следующих блоков изучения тем:

- Блок 01: Введение в CI/CD
- Блок 02: Введение в GitLab
- Блок 03: Обзор GitLab
- Блок 04: Gitlab Projects
- Блок 05: Gitlab CI: Runners
- Блок 06: Runner в Kubernetes
- Блок 07: GitLab CI: Basics
- Блок 08: Gitlab CI: Variables
- Блок 09: Gitlab CI: Tips
- Блок 10: Gitlab CI: Packages and Registries
- Блок 11: Gitlab CI: Cache and Artifacts
- Блок 12: Gitlab CI: using external secrets

Модуль направлен на освоение обучающимися инструментов и практик непрерывной интеграции (CI) с использованием GitLab — одной из самых популярных платформ для автоматизации жизненного цикла разработки. В рамках модуля слушатели получают представление о концепции CI/CD, возможностях GitLab как системы управления проектами и основах настройки пайплайнов автоматизации.

Каждый блок модуля включает теоретическое занятие с подробным описанием ключевых понятий и пошаговыми инструкциями по настройке CI-инфраструктуры. После изучения теории обучающийся выполняет практическое задание, которое направлено на закрепление навыков настройки GitLab CI, работы с раннерами, переменными окружения, кэшем, артефактами, регистрами и внешними секретами. Практические задания предусмотрены во всех блоках, включая продвинутые темы — за исключением тех случаев, где материал носит обзорный характер или является вспомогательным.

На выполнение каждого задания отводится 2 академических часа. Работы сдаются через личный кабинет, где указаны критерии оценки. Если задание выполнено корректно — выставляется зачёт. При наличии ошибок работа отправляется на доработку. Если после двух доработок задание остаётся нерешённым, преподаватель вправе выставить «незачёт».

## **Модуль 6. Kubernetes Base**

Теория 18 академ. ч. Практика 38 академ. ч.

Модуль состоит из следующих блоков изучения тем:

- Блок 01: Оркестраторы, и всегда ли они нужны
- Блок 02: Обзор популярных оркестраторов
- Блок 03: Сравнение Kubernetes и Swarm
- Блок 04: Структура API. CRD
- Блок 05: Доступ к API. RBAC и ServiceAccounts

Блок 06: Подготовка рабочего окружения  
Блок 07: Поды (pods)  
Блок 08: Объекты-контроллеры: ReplicaSet и Deployment  
Блок 09: Планирование и размещение pods: Kube-scheduler  
Блок 10: Промежуточный тест  
Блок 11: Сервисы и сеть  
Блок 12: Configmaps и secrets  
Блок 13: Ingress  
Блок 14: Persistent storage  
Блок 15: Объекты-контроллеры: StatefulSet и DaemonSet  
Блок 16: Запуск задач: Jobs и CronJobs  
Блок 17: Базовые механизмы безопасности  
Блок 18: Горизонтальное масштабирование  
Блок 19: Финальное задание

Модуль направлен на освоение основ работы с системой оркестрации контейнеров Kubernetes — ключевым инструментом в современном DevOps-стеке. В процессе обучения слушатели знакомятся с архитектурой Kubernetes, принципами его работы, основными объектами и механизмами управления контейнеризованными приложениями.

Каждый блок модуля содержит теоретическое занятие с подробными объяснениями, схемами и примерами конфигураций. После изучения теории обучающийся выполняет практическое задание, направленное на закрепление полученных знаний. В рамках модуля разбираются как базовые, так и продвинутые темы: от структуры API и настройки доступа через RBAC до конфигурации сетей, хранилищ, контроллеров, масштабирования и безопасного развёртывания приложений. Блок 10 включает промежуточное тестирование для проверки уровня усвоения ключевых понятий. Блок 19 представляет собой итоговую практическую работу по модулю, обобщающую весь пройденный материал.

На выполнение каждого практического задания отводится 2 академических часа. Задания выполняются через личный кабинет обучающегося, где размещены критерии оценки. В случае успешного выполнения выставляется зачёт. Если задание содержит ошибки, оно возвращается на доработку. После двух неудачных попыток преподаватель вправе выставить «незачёт».

### **Модуль 7. CI: QA**

Теория 5 академ. ч. Практика 12 академ. ч.

Модуль состоит из следующих блоков изучения тем:

Блок 01: Введение в тестирование  
Блок 02: Классификация тестирования  
Блок 03: QA и CI/CD  
Блок 04: QA и Docker  
Блок 05: QA и отчёты тестирования  
Блок 06: Итоговый тест: DevOps и тестирование

Модуль направлен на формирование у обучающихся представления о роли тестирования в процессе непрерывной интеграции и доставки программного обеспечения (CI/CD), а также на освоение базовых подходов и инструментов, используемых в QA-практиках. Обучающиеся знакомятся с основами тестирования, его классификацией, интеграцией тестов в пайплайны CI, особенностями использования Docker для запуска тестов, а также с методами генерации и анализа отчётов о результатах тестирования.

Каждый блок включает теоретическое занятие, где рассматриваются ключевые понятия и практические сценарии. По большинству тем предусмотрены практические задания, исключение составляет блок 01, в конце которого предусмотрен тест, а также итоговый блок, который представляет собой завершающий тест на понимание связей между DevOps и QA.

Каждое практическое задание рассчитано на 2 академических часа. Выполнение заданий осуществляется через личный кабинет, где представлены критерии оценки. При корректном выполнении обучающийся получает зачёт. В случае ошибок задание отправляется на доработку. После двух неудачных попыток преподаватель вправе выставить «незачёт».

## **Модуль 8. CD: Modern Way**

Теория 11 академ. ч. Практика 24 академ. ч.

Модуль состоит из следующих блоков изучения тем:

Блок 01: Введение в CD

Блок 02: Простой деплой Recreate

Блок 03: Бесшовный простой деплой «recreate» [\*optional]

Блок 04: Деплой с помощью docker-compose

Блок 05: Деплой в системы оркестрации

Блок 06: Деплой в K8s

Блок 07: Helm

Блок 08: Шифрование секретов с использованием kubeseal

Блок 09: Деплой в K8s: Canary

Блок 10: ArgoCD

Блок 11: Argo Rollouts [\*optional]

Блок 12: Финальное тестирование

Модуль направлен на освоение современных практик непрерывной доставки (CD) и автоматизированного развёртывания приложений. Обучающиеся получают теоретические знания и практические навыки настройки процессов доставки программного обеспечения с использованием различных инструментов и подходов, применяемых в современной DevOps-инфраструктуре.

В рамках модуля изучаются базовые и продвинутое методы деплоя: от простого способа Recreate до развёртывания в оркестраторах, Kubernetes, с использованием Helm, kubeseal и стратегий Canary. Особое внимание уделено инструментам GitOps, таким как ArgoCD и Argo Rollouts. Каждый тематический блок включает теоретический материал и практическое задание, за исключением блока 01, в конце которого предусмотрен тест; блоков 03 и 11, которые являются опциональными; и блока 12, где проводится итоговое тестирование по модулю.

Практические задания рассчитаны на 2 академических часа каждое. Задания выполняются и направляются на проверку через личный кабинет обучающегося. Критерии успешного выполнения указаны в описании заданий. При соответствии требованиям выставляется зачёт. В случае ошибок работа отправляется на доработку. После двух неудачных попыток преподаватель вправе выставить «незачёт».

## **Модуль 9. Observability**

Теория 10 академ. ч. Практика 20 академ. ч.

Модуль состоит из следующих блоков изучения тем:

Блок 01: Мониторинг, логирование и обратная связь  
Блок 02: Мониторинг приложений  
Блок 03: Мониторинг приложений в K8s  
Блок 04: Сбор логов  
Блок 05: Трейсинг  
Блок 06: Что такое Observability center  
Блок 07: Облачная платформа DataDog  
Блок 08: Мониторинг ошибок с Sentry  
Блок 09: Мобильные приложения и Crashlitics [optional]  
Блок 10: PagerDuty платформа для Ops

Модуль направлен на формирование у обучающихся знаний и практических навыков в области наблюдаемости (observability) — ключевого аспекта поддержки, сопровождения и развития современных IT-систем. В ходе изучения обучающиеся знакомятся с основными компонентами наблюдаемости: мониторингом, логированием, трассировкой и системой оповещений, а также с инструментами, которые обеспечивают полноту и глубину анализа состояния приложений и инфраструктуры.

Каждый блок модуля включает теоретическое занятие с пояснениями по архитектуре и настройке инструментов, а также практическое задание, направленное на закрепление полученных знаний. Рассматриваются как базовые темы, такие как принципы мониторинга и логирования, так и конкретные решения: Prometheus, ELK/EFK-стек, Sentry, Crashlitics, DataDog, а также платформа для управления инцидентами PagerDuty. Блок 09 носит опциональный характер.

Каждое практическое задание рассчитано на 2 академических часа. Задания выполняются и сдаются через личный кабинет обучающегося, критерии выполнения представлены в описании. При корректном выполнении выставляется зачёт. В случае ошибок задание возвращается на доработку. После двух неудачных попыток преподаватель имеет право выставить «незачёт».

### **Модуль 10. Финальное задание**

Теория 1 академ. ч. Практика 10 академ. ч.

Модуль состоит из следующих блоков изучения тем:

Блок 01: Финальное задание  
Блок 02: Заключение

Финальное задание является формой итоговой аттестации и направлено на комплексную проверку освоения всех ключевых тем программы. Задача моделирует реальную профессиональную ситуацию, в которой DevOps-инженеру необходимо собрать, настроить и задеплоить приложение в Kubernetes, используя возможности GitLab CI, без подробной документации и прямой связи с разработчиком.

В качестве задания используется приложение speedtest. Обучающемуся необходимо самостоятельно организовать процесс сборки и развёртывания приложения с нуля, реализовать необходимую инфраструктуру и обеспечить её корректную работу.

### **Критерии выполнения задания:**

1. Приложение задеплоено в кластер Kubernetes средствами GitLab CI.
2. Разработан и оформлен .gitlab-ci.yml для автоматического деплоя Helm-чарта.
3. Создан Helm-чарт для приложения speedtest.

4. Настроено логирование (EFK stack), мониторинг (Prometheus) — размещены в папке internal/logging.
5. Настроена и развернута база данных (MySQL) — манифесты или чарты размещены в папке internal.
6. Обеспечено взаимодействие приложения с СУБД — после запуска создаётся запись с данными приложения.
7. Приложение протестировано, работает корректно, отображается измеренная скорость.
8. Сделаны и приложены скриншоты:
  - вывода команды kubectl get pods, services, deployments, statefulsets, secrets, configmaps в namespace final;
  - содержимого базы данных;
  - работающего приложения с измеренной скоростью.

Проверка и оценка задания:

Задание выполняется и оформляется в виде репозитория, ссылка на который направляется на проверку через личный кабинет. Оценивание осуществляется по системе:

- Зачёт — все критерии выполнены, приложение работает стабильно, структура проекта оформлена корректно.
- Доработка — задание содержит несущественные ошибки или неточности, возможно повторное представление после исправлений.
- Незачёт — критические нарушения, задание не демонстрирует достаточный уровень освоения программы. После двух попыток доработки преподаватель вправе выставить итоговую оценку «незачёт».

В случае успешной сдачи итоговой аттестации обучающийся получает диплом о профессиональной переподготовке установленного образца.

## 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.

**Дидактический материал:**

1. Проект «Феникс». Как DevOps устраняет хаос и ускоряет развитие компании. Авторы: Джин Ким, Кевин Бер, Джордж Спаффорд.

2. Python для сетевых инженеров. Автоматизация сети, программирование и DevOps. Автор: Эрик Чоу.
3. Ускоряйся! Наука DevOps: Как создавать и масштабировать высокопроизводительные цифровые организации. Авторы: Форстрен Николь, Хамбл Джек, Ким Джин.
4. DevOps для современного предприятия: учебное пособие. Автор: М. Херинг.

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

1. Научная электронная библиотека eLIBRARY.RU.
2. Собственные учебные материалы: <https://my.rebrainme.com/course/devops>.

#### **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 автоматическом режиме в СДО.

#### **Промежуточный контроль**

Промежуточный контроль проводится после изучения следующих модулей: 2, 4, 6, 7, 8. Тематика и условия выполнения работ в рамках промежуточного контроля 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 автоматическом режиме в СДО.

#### **Итоговое оценивание**

В конце программы обучающиеся сдают итоговую аттестацию. Для успешного прохождения итогового контроля слушатель должен выполнить критерии, указанные в разделе 4.

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

По результатам сдачи текущего контроля, промежуточного 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.