



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

«УПРАВЛЕНИЕ ПРИЛОЖЕНИЯМИ В KUBERNETES С ПОМОЩЬЮ HELM»

Срок реализации: 2 месяца

Количество часов: 33 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 «Управление приложениями в Kubernetes с помощью Helm» (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 06.003 «Архитектор программного обеспечения», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 30 августа 2021 г. № 579н;
- ФГОС высшего образования – бакалавриат по направлению 09.03.04 Программная инженерия, утв. приказом Минобрнауки России от 19.09.2017 № 920;
- 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:

Программа предназначена для ИТ-специалистов, работающих с инфраструктурой, развёртыванием и сопровождением приложений в Kubernetes, в том числе:

- DevOps-инженеров, стремящихся получить практические навыки управления конфигурациями и развёртывания приложений с помощью Helm;
- Системных администраторов, желающих автоматизировать процесс деплоя и обновления сервисов в контейнерной среде;
- Разработчиков, которым важно упрощённо упаковывать, версионировать и развёртывать приложения в Kubernetes без необходимости работы с громоздкими YAML-манифестами;
- Kubernetes-администраторов, заинтересованных в централизованном управлении конфигурациями, повторном использовании шаблонов и упрощении поддержки приложений в кластерной инфраструктуре.

Требования к входным знаниям обучающегося:

Для успешного освоения программы слушателям необходимо:

- обладать базовыми знаниями о принципах работы Kubernetes;
- уметь разворачивать кластеры Kubernetes и работать с YAML-манифестами для деплоя приложений.

Актуальность реализации:

Современные инфраструктуры требуют высокой степени автоматизации и гибкости при управлении приложениями в Kubernetes. Helm стал де-факто стандартом для упаковки, управления зависимостями и развёртывания приложений в контейнерной среде. Навыки работы с Helm позволяют DevOps-инженерам, администраторам Kubernetes и разработчикам ускорять процессы CI/CD, упрощать поддержку кластерных решений и

минимизировать ошибки при ручном управлении манифестами. Освоение Helm — важный шаг к зрелому и управляемому подходу к инфраструктуре как коду.

Отличительные особенности программы:

- Полный путь от теории к практике: от базовых команд и установки до финального проекта с развёртыванием собственного helm-чарта.
- Упор на Go-templating и расширенные возможности Helm: слушатели глубоко осваивают шаблонизацию, хуки, тестирование и структуру чарта.
- Фокус на эксплуатацию: работа с зависимостями, настройка плагинов, CI/CD-сценарии — всё, что важно для повседневной DevOps-практики.
- Финальный проект как подтверждение сформированных навыков — слушатели упаковывают и деплоят полноценное приложение в Kubernetes с использованием Helm.

Объем и срок освоения программы: 33 академ. ч. в течение 2 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: удостоверение о повышении квалификации и/или сертификат об успешном освоении программы.

Цели и задачи программы:

Сформировать у слушателей практические навыки работы с Helm для управления жизненным циклом приложений в Kubernetes — от установки чарта до его создания, тестирования и эксплуатации в продуктивной среде.

Программа направлена на решение следующих основных задач:

Обучающие:

- Ознакомить слушателей с архитектурой Helm и его ролью в экосистеме Kubernetes.
- Научить использовать helm-чарты для установки, обновления и удаления приложений.
- Освоить шаблонизацию с помощью Go-templating и расширенные возможности Helm (хуки, зависимости, тесты).
- Развить умения по упаковке, отладке и деплою собственных helm-чартов.

Развивающие:

- Развить системное мышление при управлении конфигурациями и инфраструктурой как кодом.
- Сформировать устойчивые практики автоматизации развёртывания и сопровождения приложений.
- Повысить уровень технической зрелости специалистов в DevOps-процессах.

Воспитательные:

- Сформировать ответственное отношение к развёртыванию и сопровождению приложений в продуктивной среде.
- Способствовать развитию культуры командной работы и обмена знаниями.
- Повысить готовность к самостоятельному принятию решений в условиях быстро развивающейся технологической среды.

Планируемые результаты:

Знания:

- архитектура Helm и его роль в Kubernetes-экосистеме;
- структура helm-чарта и его основные компоненты;
- синтаксис и особенности Go-templating;
- назначение и реализация Helm hooks, зависимостей и тестов;
- подходы к версионированию, деплою и откату приложений в Kubernetes с использованием Helm;
- типовые ошибки и подходы к их диагностике при работе с helm-чартами

Умения:

- устанавливать Helm и работать с готовыми чартами из общедоступных и локальных репозиториев;
- создавать и настраивать собственные helm-чарты для приложений;
- использовать шаблонизацию Go-templating для генерации манифестов;
- реализовывать хуки и тесты в helm-чартах;
- управлять зависимостями и переиспользовать конфигурации в Helm;
- применять плагины Helm для расширения функциональности.

Навыки:

- автоматизация процессов развёртывания приложений в Kubernetes с помощью Helm;
- отладка, тестирование и безопасный деплой helm-чартов;
- интеграция Helm в CI/CD пайплайны и процессы сопровождения приложений;
- адаптация helm-чартов под разные окружения и сценарии эксплуатации;
- реализация финального проекта, включающего полный цикл управления приложением через Helm.

Перечень профессиональных компетенций, на получение которых направлено обучение:

На основе профстандарта 06.003 «Архитектор программного обеспечения»:

- А/02.6 Выбор и моделирование архитектурного решения для реализации программной системы
- В/02.6 Выбор и моделирование архитектурных решений для реализации интегрированного программного обеспечения

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

- ОПК-2. Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности;
- ОПК-5. Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;
- ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;

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 «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”	1	0,5	0,5	Входное тестирование
2	Модуль 2 “Знакомство с Helm	4	1	3	Практическое задание
3	Модуль 3 “Разработка Helm Charts”	16	4	12	Практическое задание
4	Модуль 4 “Эксплуатация Helm charts”	8	2	6	Практическое задание
5	Модуль 5 “Финальная работа”	4		4	Итоговое практическое задание

3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование модуля	1 месяц	2 месяц
1	Модуль 1 “Онбординг”	1	
2	Модуль 2 “Знакомство с Helm”	4	
3	Модуль 3 “Разработка Helm Charts”	13	3
4	Модуль 4 “Эксплуатация Helm charts”		8
5	Модуль 5 “Финальная работа”		4 А

4. РАБОЧАЯ ПРОГРАММА

Модуль 1. Онбординг

Теория 0,5 академ. ч. Практика 0,5 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 блока обучающийся выполняет задание по целеполаганию: формулирует свою цель прохождения программы, указывает желаемые навыки по окончании обучения, а также оценивает текущий уровень своих знаний.

Модуль 2. Знакомство с Helm

Теория 1 академ. ч. Практика 3 академ. ч.

Модуль состоит из следующих тем:

Тема 1. Установка и базовые команды

Содержание: что такое Helm, как он работает, области применения. Установка приложения. Наиболее полезные команды. Практическое задание.

Тема 2. Работа с готовыми helm-чартами. Где берут чарты

Содержание: Http-репо. OCI-registry. Github/сайт проекта. Структура чарта. Практическое задание.

Модуль 3. Разработка Helm Charts

Теория 1 академ. ч. Практика 3 академ. ч.

Тема 1. Генерация стандартного чарта. Структура чарта

Содержание: Helm как шаблонизатор. Установка helm-чарта. Практическое задание.

Тема 2. Go-template: основы. Типы данных

Содержание: Принцип работы шаблонизатора. Объекты, доступные по умолчанию. Игра в Helm. Типы данных. Переменные внутри шаблона и функции. Борьба с пробелами. Практическое задание.

Тема 3. Go-template: Именованные шаблоны. Library Charts

Содержание: Named Templates — именованные шаблоны. Контексты и области видимости. Library charts. Практическое задание.

Тема 4. Go-template: control structure

Содержание: Ветвление. Циклы. Область видимости: новые горизонты. Построение ingress, с сервисами и портами. Практическое задание.

Тема 5. Тестирование чартов. Linter и дебаг

Содержание: Популярные ошибки при написании Helm charts. Рекомендации по разработке чартов. Механизмы валидации: Helm lint, JSON Schema, Helm unittest. Дополнительная валидация для Kubernetes. Практическое задание.

Тема 6. Helm hooks

Содержание: Предпосылки для появлению хуков. Работа с Helm hooks. Практическое задание.

Тема 7. Tips and tricks создания Helm charts

Содержание: Перезапуск подов при изменении configmap. Вызов шаблонов через функцию tpl. Работа с CRD (custom resource definitions). Post Rendering. Сетевые функции. Практическое задание.

Модуль 4. Эксплуатация Helm charts

Теория 1 академ. ч. Практика 3 академ. ч.

Тема 1. Деплой приложения через Helm

Содержание: Взаимодействие Helm с Kubernetes API. Релизы в Helm. Деплой приложения с Helm. Атомарный деплой. Применение Helm в CI/CD. Практическое задание.

Тема 2. Зависимости и слияние файлов

Содержание: Принципы 3-way-merge. Зависимости helm-чартов. Управление зависимостями через сторонние утилиты. Helmfile. Практическое задание.

Тема 3. Helm и плагины

Содержание: Введение в работу с плагинами. Helm fullstatus. Helm git. Helm diff. Helm dt. Helm secrets. External Secrets Operator. Практическое задание.

Модули 2-4 направлены на формирование у обучающихся базовых и продвинутых навыков работы с системой Helm. Каждый блок модуля включает текстовое занятие с теоретическим материалом и пошаговыми инструкциями, после изучения которого предлагается практическое задание.

Практические задания рассчитаны на 1-2 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 5. Финальная работа

Блок 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/helm/>
3. Официальный сайт Helm [Электронный ресурс]: <https://helm.sh/>

4. Погружение в Helm Package Manager. Часть первая [Электронный ресурс]: <https://habr.com/ru/companies/dataart/articles/588258/>
5. Погружение в Helm Package Manager. Часть вторая [Электронный ресурс]: <https://habr.com/ru/companies/dataart/articles/589539/>
6. Основы работы с Helm чартами и темплейтами — Часть 2 [Электронный ресурс]: <https://habr.com/ru/articles/548720/>

6. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

Оценочные материалы:

Для отслеживания результатов освоения программы среди слушателей проводится текущий 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.

Модуль 2.

Тема 1. Установка и базовые команды

1. Установите утилиту helm любым удобным способом и настройте autocomplete.
2. Найдите команду, которая выводит версию helm в консоль, и сохраните её вывод в файл \$HOME/version.
3. Найдите команду, которая выводит переменные среды helm в консоль, и сохраните её вывод в файл \$HOME/variables.
4. Найдите флаг, который ограничивает количество обращений helm к kube-apiserver, и напишите его имя в файл \$HOME/flag.
5. Отправляйте задание на проверку.

Модуль 3. Разработка Helm Charts

Тема 2. Go-template: основы. Типы данных

1. Создайте тестовый чарт с именем myhelmchart по примеру из материалов блока. Путь к чарту должен быть /home/user/myhelmchart, это важно для автопроверки.
2. Создайте файл templates/deployment.yaml: ...
3. Ваша задача — заполнить values.yaml и шаблонизировать в deployment.yaml:
 - Имя deployment. Тип данных — строка. Имя переменной в values: deploymentName. В итоговом манифесте значение переменной должно принудительно приводиться к нижнему регистру.
 - Метки и селекторы. Тип данных для переменной в values.yaml — структура. Имя переменной: labels.
 - replicas — целое число. Имя переменной: replicas.
 - Имя образа, должно задаваться переменной \$fullImageName. Переменная должна состоять из имени образа image и tag. Если тег не задан, в качестве его значения должна использоваться версия приложения (appVersion из метаданных чарта). Итоговое значение должно быть взято в кавычки. Не создавайте больше никаких переменных кроме \$fullImageName
 - resources — структура. Имя переменной аналогичное.
4. Отправляйте задание на проверку.

Модуль 4. Эксплуатация Helm charts

Тема 1. Деплой приложения через Helm

1. Загрузите чарт из репозитория командой:
helm pull oci://registry-1.docker.io/bitnamicharts/nginx --version 19.0.1
 - Распакуйте чарт.
 - Создайте файл values.dev.yaml, его вы будете использовать для параметризации при установке приложения.
2. Установите nginx со следующими параметрами:
 - Количество реплик 3
 - limits.cpu=100m, limits.memory=100Mi
 - requests.cpu=50m, requests.memory=50Mi
 - образ bitnami/nginx:1.27.4
 - имя релиза nginx-server-v1
3. Измените лимиты приложения и проведите повторную установку чарта с limits.cpu=500m, limits.memory=256Mi.
4. Измените количество реплик до 5 и проведите повторную установку чарта.
5. Проверьте количество подов.
6. Посмотрите количество релизов.
7. Откатитесь до релиза с количеством реплик равным 3 и новыми лимитами.
8. Проверьте, что новые параметры применились к подам, и их указанное количество.
9. У вас в кластере запущено приложение с именем my-broken-release, которое не работает, найдите его и откатитесь к предыдущей версии.
10. Отправляйте задание на проверку

Итоговое оценивание

Финальное задание

Поздравляем! Вы преодолели все задачи и дошли до финала практикума!

Теперь вам предстоит закрепить все полученные знания и навыки в большой комплексной задаче.

Вы создадите Helm-чарт для установки приложения speedtest с поддержкой различных баз данных, конфигурируемыми секретами и настройками, а также реализуете интеграцию с ingress, readiness/liveness пробами и другими важными компонентами.

Ваша задача — сделать полноценный helm-чарт с зависимостями для установки приложения.

В тексте задания присутствуют блоки «Дополнительно:» — они введены для тех, кто хочет извлечь максимум пользы из этого задания. Выполнение блоков не влияет на итоговую оценку.

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.