



Общество с ограниченной ответственностью «Ребрейн»

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
«ВВЕДЕНИЕ В АРХИТЕКТУРУ И БАЗОВЫЕ ПРИНЦИПЫ KUBERNETES»

Срок реализации: 1 месяц

Количество часов: 50 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 «Введение в архитектуру и базовые принципы Kubernetes» (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 «Ребрейн».

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:

- Начинающим DevOps- и системным инженерам, которые хотят получить базовые знания о Kubernetes и начать практическую работу с контейнерной оркестрацией.
- IT-специалистам, знакомым с Docker и контейнеризацией, которые стремятся освоить управление кластерами Kubernetes.
- Студентам и начинающим разработчикам, интересующимся современными технологиями развертывания и сопровождения приложений.

Требования к входным знаниям обучающегося:

- Уверенное владение Linux на уровне работы с командной строкой.
- Базовые навыки написания Bash-скриптов.
- Понимание принципов контейнеризации (желательно знание Docker).
- Общие представления о сетевых технологиях (адресация, протоколы, сервисы).

Актуальность реализации:

Kubernetes стал де-факто стандартом в мире контейнерной оркестрации. Сегодня его применяют в стартапах, корпорациях и госсекторе для управления инфраструктурой, повышения отказоустойчивости и автоматизации развертывания приложений. Знание архитектуры и базовых принципов Kubernetes — это необходимый навык для специалистов, работающих с DevOps-процессами, облачными системами и современными распределенными приложениями. Освоение Kubernetes позволяет быть востребованным в проектах любого масштаба и обеспечивает уверенное развитие карьеры в сфере инфраструктурных технологий.

Отличительные особенности программы:

- Пошаговое введение в архитектуру Kubernetes: от API и объектов-контроллеров до сетевых и storage-механизмов.
- Фокус на практическом применении: все теоретические темы сопровождаются заданиями и промежуточным тестом.
- Изучение ключевых объектов и инструментов Kubernetes: Pods, ReplicaSet, Deployment, StatefulSet, DaemonSet, Jobs и CronJobs.
- Освоение базовых механизмов безопасности и масштабирования приложений.
- Подготовка к дальнейшему углубленному изучению Kubernetes и работе с продвинутыми сценариями (CI/CD, сервис-меш, observability).

Объем и срок освоения программы: 48 академ. ч. в течение 1 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 базовое представление об архитектуре Kubernetes и научить применять его ключевые объекты и механизмы для развертывания и управления приложениями.

Программа направлена на решение следующих основных задач:

Обучающие:

- Познакомить с архитектурой Kubernetes и его API.
- Научить работать с базовыми объектами: Pods, ReplicaSet, Deployment, StatefulSet, DaemonSet.
- Дать знания о механизмах сетевого взаимодействия, хранения данных и безопасности в Kubernetes.

Развивающие:

- Сформировать навыки практической работы с kubectl и манифестами.
- Развить умение применять принципы автоматизации и масштабирования при работе с приложениями.
- Содействовать развитию системного мышления при проектировании инфраструктуры.

Воспитательные:

- Воспитать ответственное отношение к надежности и безопасности информационных систем.
- Сформировать культуру работы в команде с использованием современных DevOps-инструментов.
- Развить готовность к самообучению и адаптации к быстро меняющимся технологиям.

Планируемые результаты:

В результате прохождения программы у слушателя будут сформированы:

Знания:

- архитектура Kubernetes, структура API и роль CRD;
- основные объекты: Pods, ReplicaSet, Deployment, StatefulSet, DaemonSet, Jobs и CronJobs;
- механизмы работы сервисов, сетевого взаимодействия, ConfigMaps, Secrets и Ingress;
- основы обеспечения безопасности и масштабирования приложений в Kubernetes.

Умения:

- использовать kubectl для управления кластерами и приложениями;
- разворачивать приложения с использованием различных контроллеров;
- организовывать хранение данных с помощью Persistent Volumes;
- настраивать доступ, разграничение прав и базовые механизмы безопасности;
- диагностировать и устранять проблемы в прикладной части работы с Kubernetes.

Навыки:

- практическая работа с манифестами Kubernetes;
- применение механизмов балансировки, масштабирования и отказоустойчивости;
- настройка и сопровождение приложений в кластере;
- использование Kubernetes как основы для построения современной инфраструктуры.

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 устанавливать 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:

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	Модуль 1 “Онбординг”	1	0,5	0,5	Входное тестирование
2	Модуль 2 “Kubernetes Base”	40	15	25	Практическое задание
3	Итоговая аттестация	7		7	Итоговое практическое задание

3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование модуля	1 неделя	2 неделя	3 неделя	4 неделя
1	Модуль 1 “Онбординг”	1			
2	Модуль 2 “Kubernetes Base”	12	13	13	2
3	Итоговая аттестация				7 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 темы обучающийся выполняет задание по целеполаганию: формулирует свою цель прохождения программы, указывает желаемые навыки по окончании обучения, а также оценивает текущий уровень своих знаний, выбрав один из предложенных вариантов.

Модуль 2. Kubernetes Base

Теория 15 академ. ч. Практика 25 академ. ч.

Модуль состоит из следующих тем:

Тема 1. Структура API и CRD

Содержание: Архитектура Kubernetes API. Custom Resource Definitions (CRD) и Gateway API. Создание собственного namespace. Работа с метками и аннотациями. Расширение функциональности кластера с помощью CRD. Практическое задание.

Тема 2. Доступ к API. RBAC и ServiceAccounts

Содержание: Управление доступом с помощью RBAC и ServiceAccounts. Создание ролей, назначение прав, настройка безопасного доступа для пользователей и сервисов. Практическое задание.

Тема 3. Подготовка рабочего окружения

Содержание: Развёртывание локального кластера (Minikube/Kind), настройка kubectl, проверка подключения и подготовка среды для работы с Kubernetes. Практическое задание.

Тема 4. Pods

Содержание: Создание и управление подами. Диагностика проблем при запуске, устранение ошибок и проверка состояния подов. Практическое задание.

Тема 5. Объекты-контроллеры: ReplicaSet и Deployment

Содержание: Разница между ReplicaSet и Deployment. Создание, обновление и масштабирование деплоймов, восстановление некорректных Deployment'ов. Практическое задание.

Тема 6. Планирование и размещение Pods: Kube-scheduler

Содержание: Настройка размещения подов с помощью affinity, taints/tolerations и ресурсов. Оптимальное распределение нагрузки в кластере. Практическое задание.

Тема 7. Промежуточный тест

Содержание: Проверка освоенных навыков: создание подов, деплоймов, управление доступом и базовая диагностика. Практическое задание.

Тема 8. Сервисы и сеть

Содержание: Создание ClusterIP, NodePort и LoadBalancer сервисов. Предоставление сетевого доступа к приложениям и устранение сетевых проблем. Практическое задание.

Тема 9. ConfigMaps и Secrets

Содержание: Создание и использование ConfigMaps и Secrets. Управление конфигурациями, обеспечение безопасности и конфиденциальности данных в подах. Практическое задание.

Тема 10. Ingress

Содержание: Настройка Ingress для маршрутизации внешнего трафика к сервисам. Создание правил, настройка TLS и устранение ошибок маршрутизации. Практическое задание.

Тема 11. Persistent Storage

Содержание: Создание PersistentVolume и PersistentVolumeClaim. Подключение хранилищ к подам для сохранения данных при перезапуске контейнеров. Практическое задание.

Тема 12. Объекты-контроллеры: StatefulSet и DaemonSet

Содержание: Создание StatefulSet (например, для Kafka) и DaemonSet для развёртывания контейнеров на всех nodes. Управление состоянием подов и хранилищ. Практическое задание.

Тема 13. Запуск задач: Jobs и CronJobs

Содержание: Создание одноразовых и периодических задач с Jobs и CronJobs. Мониторинг выполнения и управление историей задач. Практическое задание.

Тема 14. Базовые механизмы безопасности

Содержание: Ограничение привилегий подов и пользователей. Настройка сетевых политик и контроль доступа к ресурсам кластера. Практическое задание.

Тема 15. Горизонтальное масштабирование

Содержание: Настройка Horizontal Pod Autoscaler. Автоматическое масштабирование подов в зависимости от нагрузки для поддержания стабильной работы приложений. Практическое задание.

Модуль направлен на формирование у обучающихся базовых и продвинутых навыков работы с Kubernetes. Каждая тема 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».

Итоговая аттестация.

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 [Электронный ресурс]: <https://rebrainme.com/kubernetes/>
3. Официальный сайт Kubernetes [Электронный ресурс]: <https://kubernetes.io/>

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 автоматическом режиме в СДО.

Тема 1. Структура API. CRD

1. Установите `kubectl` и настройте autocompletion (чтобы он загружался при старте профиля). Шпаргалка по kubectl вам в помощь.
2. Посмотрите список нод.
3. Создайте неймспейс с именем `my-ns`.
4. Добавьте ему метку `kubernetes=rulezz` через команду `kubectl label`.
5. Сохраните манифест вашего неймспейса через команду `kubectl get -o yaml`.
6. Отредактируйте файл, добавив дополнительно 2 лейбла: `second=2`, `third=3` — и примените ваш новый манифест. Через команду `kubectl describe` проверьте, что изменения применились.
7. Добавьте аннотации `annotations=likeLabels` для неймспейса.
8. Отредактируйте файл из пункта 5, удалив из него `second=2`, и примените. Проверьте результаты работы через `kubectl get`. Обратите внимание на аннотации!

Тема 13. Запуск задач: Jobs и CronJobs

Задание

- Добавьте двум узлам вашего кластера метки `usecase=job-node`.
- Создайте `CronJob` с именем `cron-test` в namespace `default`, который будет раз в 10 минут (`*/10 * * * *`) на нод с меткой `usecase=job-node` запускать контейнер с образом `busybox:1.36` и командой `curl <https://google.com>`.
- Дождитесь запуска контейнера. Изучите логи контейнера и `describe`.
- Создайте `Job` с именем `job-test` в namespace `default`. Пусть у него будет `InitContainer` с именем `pre-ping` и командой `ping 8.8.8.8`, а основной контейнер с командой `curl <https://google.com>`. Образа у обоих контейнеров `busybox:1.36`.
- Изучите логи контейнеров и `describe`.
- Оправьте задание на проверку.

Итоговое оценивание

В конце программы обучающиеся сдают итоговую аттестацию.

Задание

У вас есть приложение <https://github.com/librespeed/speedtest> (кому-то оно может быть знакомо из практикума по Docker).

Ситуация типичная: вам дали репозиторий, который нужно задеплоить в кубер. Информации ноль, позвонить и узнать, как работает приложение некому (если автор не вкоммитил свой номер телефона где-то в репозитории).

Требуется итоговый набор манифестов, чтобы кто-то сделал `kubectl apply -f` и приложение заработало.

Для удобства помещайте все объекты в разные файлы.

Требования к настройкам приложения

- Приложение должно быть запущено во frontend-режиме с хранением телеметрии, обфускацией ID и без возможности редактирования IP-адресов (для этого придётся немного поизучать репозиторий приложения и найти настройки — см. Dockerfile и `docker/entrypoint.sh`).
- Конфигурация должна производиться через переменные окружения (кроме тех случаев, когда нам явно требуется файл).
- СУБД — MySQL, пример развёртывания можно взять из документации.
- По желанию можно развернуть провизионер дисков (либо просто не создавайте их и данные будут писаться в файловую систему контейнера).

Требования к СУБД

- Разверните СУБД MySQL в namespace db.
- На всех манифестах, касающихся СУБД, должна присутствовать метка `app=db`.
- Имя манифеста с основной нагрузкой - `mysql` (тип вы без труда определите сами).
- Определите, где нужны secrets, а где configmaps, и создайте их. Присвойте имена `mysql-configmap` и `mysql-secret`.
- Создайте 2 сервиса: `clusterip` с именем `mysql` и `headless` с именем `mysql-headless`.
- Не забудьте инициализировать структуру базы из sql-скрипта. Его нужно смонтировать внутрь контейнера. Чтобы узнать куда, посмотрите описание на ваш образ на hub.docker.com

Требования при работе с kubernetes

- Приложение должно быть развёрнуто в namespace final.
- На всех манифестах, касающихся приложения, должна присутствовать метка `app.kubernetes.io/instance=librespeed`.
- Имя деплоймента `librespeed`.
- Образ должен скачиваться с нашего закрытого registry. Токен pull-creds пароль `gldt-LUE7NKmJu3F55r_fxsZC` для регистрии `registry.rebrainme.com`. Секрет должен называться `registrysecret`.
- В качестве списка серверов необходимо использовать файл `servers.json`. Он должен быть смонтирован в ваш под в `/servers.json`. В качестве имени конфигапы используйте `librespeed-servers`.

[...]

Результаты текущего контроля 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.