



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  
**«РАБОТА С УПРАВЛЯЕМЫМИ KUBERNETES-КЛАСТЕРАМИ В  
YANDEX CLOUD»**

**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 «Работа с управляемыми Kubernetes-кластерами в Yandex Cloud» (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:

Программа предназначена для специалистов, которые работают с облачной инфраструктурой и контейнерными приложениями:

- DevOps-инженеры — смогут с помощью Yandex Managed Service for Kubernetes автоматизировать развёртывание, масштабирование и обновление кластеров, минимизировать затраты на администрирование, а также повысить отказоустойчивость сервисов.
- Разработчики — научатся быстро запускать и масштабировать приложения в Kubernetes без необходимости ручного управления кластерами, использовать гибкие возможности настройки и интеграции с сервисами Yandex Cloud для ускорения процессов разработки и тестирования.
- Системные администраторы — получат единый инструмент управления инфраструктурой, который позволит заменить разрозненные виртуальные машины централизованной облачной платформой, облегчая сопровождение и эксплуатацию.

### Требования к входным знаниям обучающегося:

Для успешного освоения программы слушателям рекомендуется:

- владеть базовыми навыками работы в Linux (командная строка, управление процессами и сервисами, работа с файлами и сетевыми утилитами);
- иметь опыт работы с Docker или другими технологиями контейнеризации;
- понимать основные принципы работы Kubernetes (pods, контроллеры, сервисы, конфигурационные объекты);
- обладать базовыми знаниями в области сетевых технологий (TCP/IP, DNS, балансировка нагрузки);

- ориентироваться в облачных сервисах и их архитектуре (будет преимуществом опыт работы с Yandex Cloud или аналогами — AWS, GCP, Azure).

#### **Актуальность реализации:**

Использование Kubernetes стало стандартом в современной разработке и эксплуатации приложений. Однако самостоятельное администрирование кластеров требует значительных усилий: от настройки инфраструктуры до обеспечения отказоустойчивости и обновлений. Управляемый сервис Kubernetes в Yandex Cloud снимает с инженеров большую часть рутинных задач и позволяет сосредоточиться на разработке и эксплуатации приложений.

Программа актуальна для специалистов, которые хотят:

- освоить особенности работы с Managed Service for Kubernetes в Yandex Cloud;
- научиться интегрировать Kubernetes с другими сервисами облачной платформы;
- повысить эффективность и скорость развёртывания приложений;
- обеспечить масштабируемость и безопасность инфраструктуры без лишних затрат времени на администрирование.

#### **Отличительные особенности программы:**

- Практическая направленность — все ключевые темы сопровождаются примерами и лабораторными работами в Yandex Cloud.
- Фокус на управляемом сервисе — изучаются специфические возможности Yandex Managed Service for Kubernetes, включая UI, интеграции и облачные инструменты.
- Сравнение с self-hosted решениями — слушатели поймут, какие задачи решает облачный сервис и в каких случаях он эффективнее классических подходов.
- Интеграции с другими сервисами Yandex Cloud — хранение секретов, CI/CD, балансировка нагрузки, мониторинг и логирование.
- Актуальные практики DevOps и SRE — масштабирование, управление ресурсами, безопасность и «best practices» разработки приложений под Kubernetes.

**Объем и срок освоения программы:** 50 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 целостное представление об управляемом Kubernetes в Yandex Cloud.
- Научить применять Yandex Managed Service for Kubernetes для развёртывания, масштабирования и администрирования приложений.
- Развить навыки интеграции Kubernetes-кластеров с другими сервисами Yandex Cloud.
- Подготовить специалистов к эффективной эксплуатации облачной инфраструктуры в условиях высоких требований к производительности и отказоустойчивости.

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

### Обучающие:

- дать представление о базовых абстракциях Kubernetes и принципах работы управляемого сервиса;
- обучить работе с UI и API Яндекс Облака для управления кластерами;
- освоить практику настройки сетевых сервисов, хранилищ и балансировщиков;
- научить использовать Helm и разрабатывать собственные чарты;
- отработать интеграцию с CI/CD и сервисами хранения секретов.

### Развивающие:

- развить способность проектировать и поддерживать отказоустойчивые решения на базе управляемого Kubernetes;
- сформировать умение применять DevOps-практики в работе с облачными кластерами;
- улучшить навыки анализа и устранения проблем в работе приложений и кластера.

### Воспитательные:

- сформировать ответственное отношение к вопросам безопасности и надёжности облачной инфраструктуры;
- развить культуру документирования и применения best practices в команде;
- воспитать стремление к оптимизации процессов и снижению ручного труда за счёт автоматизации.

## Планируемые результаты:

### Знания:

- архитектура и основные абстракции Kubernetes;
- особенности управляемого сервиса Kubernetes в Yandex Cloud;
- принципы организации хранения данных, сетевых взаимодействий и балансировки нагрузки;
- механизмы безопасности: RBAC, ServiceAccounts, secrets, ограничение ресурсов;
- возможности интеграции Kubernetes с другими сервисами Yandex Cloud (хранилища, CI/CD, секреты).

### Умения:

- работать с UI и API Яндекс Облака для администрирования кластеров;
- развёртывать и масштабировать приложения в управляемых Kubernetes-кластерах;
- конфигурировать сервисы, ingress, балансировщики и persistent storage;
- использовать Helm для установки приложений и разработки собственных чартов;
- отлаживать работу приложений и кластера, устранять ошибки и настраивать мониторинг.

### Навыки:

- практическая эксплуатация управляемых кластеров Kubernetes в Yandex Cloud;
- проектирование отказоустойчивых решений с учётом масштабирования и безопасности;
- применение best practices по организации инфраструктуры как кода и CI/CD;
- интеграция управляемых кластеров с корпоративной инфраструктурой и облачными сервисами.

## 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:
  - 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»;
  - 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;
  - 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 “Введение в Kubernetes. Базовые абстракции”	8	2	6	Практическое задание
3	Модуль 3 “Хранение данных и сеть”	8	2	6	Практическое задание
4	Модуль 4 “Продвинутые практики”	16	6	10	Практическое задание
5	Модуль 5 “Helm”	5	1	4	Практическое задание

6	Модуль 6 “Интеграции Kubernetes”	5	1	6	Практическое задание
7	Итоговая аттестация	6		6	Итоговое практическое задание

### 3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование модуля	1 неделя	2 неделя	3 неделя	4 неделя
1	Модуль 1 “Онбординг”	2			
2	Модуль 2 “Введение в Kubernetes. Базовые абстракции”	8			
3	Модуль 3 “Хранение данных и сеть”	2	6		
4	Модуль 4 “Продвинутые практики”		7	9	
5	Модуль 5 “Helm”			5	
6	Модуль 6 “Интеграции Kubernetes”				5
7	Итоговая аттестация				6   А

### 4. РАБОЧАЯ ПРОГРАММА

#### Модуль 1. Онбординг

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

Модуль состоит из следующих тем:

##### Тема 1. Входное тестирование

Тестирование состоит из трёх частей, проверяющих компетенции, лежащие в основе практикума Kubernetes x Yandex Cloud:

- Базовые абстракции Kubernetes;
- Хранение и сети;
- Продвинутые практики.

Каждый блок состоит от 4 до 11 вопросов, призванных проверить ваш уровень владения той или иной компетенцией на момент старта программы. Всего в тесте 20 вопросов.

##### Тема 2: Онбординг

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

Предусмотрено прохождение 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.

## **Модуль 2. Введение в Kubernetes. Базовые абстракции**

Теория 2 академ. ч. Практика 6 академ. ч.

Модуль состоит из следующих тем:

Тема 1: Знакомство с Kubernetes. UI Яндекс Облака

Содержание: Что делает и как работает Kubernetes. Архитектура Kubernetes. Разница Docker Swarm и Kubernetes. Kubernetes и облачные провайдеры. Как самостоятельно создать кластер в UI и получить к нему доступ из консоли. Практическое задание.

Тема 2: Структура API

Содержание: API-сервер в структуре Kubernetes. Взаимодействие с кластером Kubernetes. Альтернативные способы общения с API. Kubeconfig. Способы расширения API. Практическое задание.

Тема 3: Поды (pods)

Содержание: Краткая сводка: поды. Создание пода. Жизненный цикл пода. Диагностика ошибок при запуске подов. Health-check-проверки. Практическое задание.

Тема 4: Объекты-контроллеры: ReplicaSet и Deployment

Содержание: ReplicaSet. Типичный Deployment. Механизмы трафикшутинга. Практическое задание.

Модуль направлен на формирование фундаментальных знаний и практических навыков работы с управляемыми Kubernetes-кластерами. В результате прохождения модуля участники смогут самостоятельно создавать кластеры в UI Яндекс Облака, подключаться к ним из консоли, разбираться в структуре API и конфигурационных файлах, выявлять и устранять ошибки при запуске подов, а также организовывать масштабируемые и отказоустойчивые приложения в Kubernetes.

## **Модуль 3. Хранение данных и сеть**

Теория 2 академ. ч. Практика 6 академ. ч.

Модуль состоит из следующих тем:

Тема 1: Configmaps и secrets

Содержание: что такое configmap и зачем оно нужно. Как монтировать configmap в под. Что такое secret и чем он отличается от configmap. Практическое задание.

Тема 2: Сервисы и сеть

Содержание: Что такое сервис; типы сервисов. Сетевое взаимодействие. External Traffic Policy. hostPort. Практическое задание.

Тема 3: Ingress

Содержание: Ingress как компонент Kubernetes. Ingress-nginx controller. Gateway API. Поиск ошибок. Практическое задание.

Тема 4: Yandex Application Load Balancer

Содержание: как установить и задеплоить Yandex Application Load Balancer. Как ресурсы Application Load Balancer соотносятся со стандартными ресурсами Kubernetes. Как сделать canary deploy на ALB. Как установить два ingress-контроллера. Практическое задание.

## Тема 5: Persistent storage

Содержание: Устройства Volume. PersistentVolumes. Storageclass. Создание и подключение дисков. Особенности работы с дисками. Изменение размеров диска. Создание снимков. Поиск ошибок. Практическое задание.

Модуль направлен на освоение инструментов работы с конфигурационными данными и секретами, а также на получение практических навыков организации сетевого взаимодействия в Kubernetes.

## Модуль 4. Продвинутые практики

Теория 6 академ. ч. Практика 10 академ. ч.

Модуль состоит из следующих тем:

### Тема 1: Планирование и размещение pods: Kube-scheduler

Содержание: Kube-scheduler и способы управления размещением. Affinity и anti-affinity. Taints. Ручные блокировки. Pod disruption budget. Практическое задание.

### Тема 2: Объекты-контроллеры: StatefulSet и DaemonSet

Содержание: контроллер StatefulSet. Манифест DaemonSet. Практическое задание.

### Тема 3: Устройство кластера

Содержание: Архитектура Kubernetes. Промежуточная аттестация.

### Тема 4: Доступ к API. RBAC и ServiceAccounts

Содержание: как работает доступ к API. Что выбрать: user или serviceaccount. Как задать нужную роль пользователю и как работает ролевой доступ в kubernetes (RBAC). Практическое задание.

### Тема 5: Запуск задач: Jobs и CronJobs

Содержание: Ресурс для запуска пода Job. Cronjob. Init-containers. Sidecars. Практическое задание.

### Тема 6: Горизонтальное масштабирование и способы ограничения потребления ресурсов

Содержание: Манифест HPA. Установка metrics-server для self-hosted кластера. Создание HPA. Способы ограничения ресурсов. Практическое задание.

### Тема 7: Базовые механизмы безопасности

Содержание: SecurityContext как механизм управления параметрами безопасности для подов и контейнеров, работающих внутри пода. NetworkPolicy как механизм ограничения сетевых доступов. Практическое задание.

### Тема 8: Дебаг кластера и приложений

Содержание: Ошибки работы подов. kubectl debug. Механизмы траблшутинга объектов-контроллеров. Устранение ошибок при работе с сервисами. Поиск ошибок при работе с ingress. Практическое задание.

### Тема 9: Best Practice по написанию приложений для K8s

Содержание: Концепция Twelve-Factor App. Cloud native. Про разницу отказоустойчивости и НА. Serverless. Best practice для приложений в Kubernetes. Тестирование.

Модуль направлен на углублённое освоение ключевых механизмов Kubernetes, которые позволяют управлять размещением рабочих нагрузок, обеспечивать масштабируемость, безопасность и отказоустойчивость приложений.

## **Модуль 5. Helm**

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

Модуль состоит из следующих тем:

Тема 1: Знакомство с Helm

Содержание: Краткая справка о Helm. Работа с helm: установка, команды. Как добыть чарт. Практическое задание.

Тема 2: Создание Helm-чарта

Содержание: как выглядит helm chart по умолчанию. Как установить helm chart и что будет, если вносить в него изменения. Как самостоятельно создать helm chart. Практическое задание.

Модуль направлен на освоение инструментов пакетного менеджмента в Kubernetes с помощью Helm. Слушатели познакомятся с принципами работы Helm, научатся устанавливать и использовать готовые чарты, а также разрабатывать собственные. Практические задания помогут закрепить навыки автоматизации развёртывания и управления приложениями через Helm-чарты, что позволит ускорять внедрение изменений и упрощать сопровождение инфраструктуры.

## **Модуль 6. Интеграции Kubernetes**

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

Модуль состоит из следующих тем:

Тема 1: Интеграция с хранилищем секретов

Содержание: SOPS. Передача ключа. Helm plugins. Lockbox для хранения секретов. Интегрируем Kubernetes с Lockbox. Практическое задание.

Тема 2: CI/CD

Содержание: Понятие CI/CD-процесса. Gitlab для CI/CD. Практическое задание.

Модуль направлен на освоение практик интеграции Kubernetes с внешними сервисами и инструментами, необходимыми для безопасного и автоматизированного управления инфраструктурой. Слушатели научатся работать с секретами с использованием Lockbox и Helm-плагинов, а также внедрять CI/CD-процессы на базе GitLab. Практика позволит закрепить навыки безопасного хранения данных и автоматизации жизненного цикла приложений.

Каждая тема модулей 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.**

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/kubernetes-yandex-managed-service/>
3. Документация Yandex Managed Service for Kubernetes [Электронный ресурс]: <https://yandex.cloud/ru/docs/managed-kubernetes/>
4. Кластерная архитектура Kubernetes [Электронный ресурс]: <https://kubernetes.io/ru/docs/concepts/architecture/>

## 6. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

**Оценочные материалы:**

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

**Вводное тестирование.**

Тестирование состоит из трёх частей, проверяющих компетенции, лежащие в основе практикума Kubernetes x Yandex Cloud.

Каждый блок состоит от 4 до 11 вопросов, призванных проверить ваш уровень владения той или иной компетенцией на момент старта программы. Всего в тесте 20 вопросов.

Каждый вопрос проверяет один из необходимых аспектов компетенции, чтобы вы могли определить свои сильные стороны и понимать, на что нужно будет обратить внимание в процессе прохождения практикума.

За каждый правильный ответ вам начисляется по одному баллу. Всего вы можете набрать 29 баллов (так как некоторые вопросы содержат несколько верных ответов).

Порога прохождения теста нет, так как это входная диагностика — этот результат вы сможете сравнить с результатом прохождения теста в заключительном блоке и оценить ваш личный прогресс.

### Рекомендации по прохождению

- выделите на прохождение теста 30 минут;
- найдите место, где будет тихо и вас никто не будет отвлекать;
- внимательно читайте инструкции к заданиям;
- если не знаете правильный ответ, выберите тот вариант, который кажется вам наиболее вероятным. Не гуглите ответы, ведь тестирование нужно, чтобы определить ваш реальный уровень знаний;
- сохраняйте спокойствие — это не экзамен, а часть учебного процесса;
- не переживайте, если неверно ответили на какие-то вопросы: вы пришли получать знания и навыки, и мы вам в этом поможем!

### Базовые абстракции

1. Базовые абстракции. Сфера применения Kubernetes. Какие из указанных возможностей поддерживает стандартный Kubernetes? Выберите 3 верных ответа.
  - А. Защита приложений от DDoS-атак
  - Б. Автоматическое масштабирование
  - С. Использование механизмов ресурсных пулов и квот
  - Д. Мониторинг и отрисовка метрик
  - Е. Управление секретами
2. Базовые абстракции. Namespaces, labels, annotations. Какие утверждения верны относительно создания неймспейсов в Kubernetes? Выберите 2 верных утверждения.
  - А. Неймспейс - это группа контейнеров, разделяющих общее процессное пространство
  - Б. Неймспейс можно создать с помощью команды kubectl
  - С. Неймспейсы предназначены для разделения ресурсов внутри кластера
  - Д. Один под может быть одновременно в нескольких неймспейсах
  - Е. Удаление неймспейса автоматически удаляет всее ресурсы в нём

### Хранение данных и сеть

1. Хранение данных и сеть. Configmap, Secrets. Вам нужно передать НЕКОНФИДЕНЦИАЛЬНУЮ информацию, например, URL API, в контейнер приложения. Какой механизм в Kubernetes предназначен для передачи «открытых» параметров конфигураций? Выберите 1 верный ответ.
  - А. Добавление URL API в Secrets для обеспечения безопасности
  - Б. Внедрение URL API в исходный код приложения
  - С. Указание URL API в манифесте Pod напрямую
  - Д. Хранение URL API в ConfigMap и подключение его как переменную окружения
2. Хранение данных и сеть. Типы сервисов. Какой тип сервиса в Kubernetes автоматически создаёт внешний IP-адрес для доступа к приложению из интернета в облачных средах? Выберите 1 верный ответ.
  - А. ClusterIP
  - Б. NodePort
  - С. LoadBalancer
  - Д. ExternalName

### Продвинутые практики

1. Продвинутые практики. Управление контроллерами: StatefulSet, DaemonSet. Какие из перечисленных характеристик отличают StatefulSet от других абстракций Kubernetes, таких как Deployment, ReplicaSet или DaemonSet? Выберите 2 варианта ответа.

- A. Автоматическое создание и удаление PersistentVolume для каждого пода
- B. Гарантия уникальных предсказуемых имен подов для каждого экземпляра
- C. Сохранение диска за подом после его пересоздания
- D. Возможность автоматического масштабирования приложения
- E. Автоматическое распределение подов по узлам кластера

### 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. Введение в Kubernetes. Базовые абстракции

### Тема 2: Структура API

#### Практическое задание

1. Установите yc-scli, инициализируйте профиль и получите kubeconfig (кластер мы создали в первом задании)
2. Установите kubectl и настройте autocomplete (чтобы он загружался при старте профиля). Шпаргалка по kubectl из полезных материалов вам в помощь.
3. Создайте неймспейс с именем yck2389.
4. Получите список всех неймспейсов командой kubectl get ns
5. Добавьте ему метку kubernetes=tulezz через команду kubectl label.
6. Сохраните манифест вашего неймспейса через команду kubectl get ..... -o yaml.
7. Отредактируйте файл, добавив дополнительно 2 лейбла: second="2" , third="3" — и примените ваш новый манифест. Через команду kubectl describe проверьте, что изменения применились. При помощи команды kubectl добавьте аннотации annotations=likelabels для неймспейса.
8. Отредактируйте манифест, удалив из него second=2, и примените. Получите ошибку
9. Снова сохраните неймспейс в файл. Удалите из него second=2, и примените.

## Модуль 3. Хранение данных и сеть

### Тема 2: Сервисы и сеть

#### Практическое задание

1. Создайте namespace yck2393 и переключитесь в него.
2. Создайте деплоймент httpd-dp в namespace yck2393 с образом httpd.
3. Создайте сервис http-svc-int в namespace yck2393 с типом ClusterIP, который будет смотреть на поды из пункта 2.
4. Создайте сервис http-svc-nodeport в namespace yck2393 с типом nodePort с портом 32501, который будет смотреть на поды из пункта 2.
5. Создайте сервис http-svc-ext в namespace yck2393 с типом LoadBalancer, который будет смотреть на поды из пункта 2.
6. При помощи curl и port-forward проверьте работоспособность всех сервисов.
7. Также вы можете посмотреть созданные объекты через UI. Найдите их в разделах ( Managed Service for Kubernetes/Кластер/Сеть и Network Load Balancer/Балансирующие)

8. Отправьте задание на проверку.

## Модуль 4. Продвинутые практики

### Тема 4: Доступ к API. RBAC и ServiceAccounts

#### Практическое задание

1. Создайте namespace с именем yck2399.
2. Создайте сервис-аккаунт с именем test-user в namespace yck2399 и токен.
3. Создайте сервис-аккаунт с именем test-admin в namespace yck2399 и токен.
4. В namespace yck2399 создайте роль test-role такую, чтобы сервис аккаунт получивший её:
  - Получил все возможные права на serviceaccounts, кроме удалений.
  - На pods – get, list, watch.
  - Не имел прав на создание/изменение rolebindings в данном неймспейсе.
5. Создайте rolebinding с именем test-binding и привяжите роль test-role к serviceaccount test-user в namespace yck2399.

[...]

## Модуль 5. Helm

### Тема 1: Знакомство с Helm

#### Практическое задание

В данном задании вам могут понадобиться свободные ресурсы на нодах, поэтому стоит удалить часть деплойментов, развёрнутых ранее.

1. Установите утилиту helm.
2. Возьмите helm-чарт с postgres отсюда.
3. Установите его в неймспейс yck2405:
  - Имя релиза postgres-db.
  - Включите создание дисков. Используйте storageclass: yc-network-hdd, размер диска 1 ГБайт.
  - Найдите блок, который добавляет метки и добавьте в него mypostgres=true.
  - База и имя пользователя mypostgres. Пароль для пользователя и для админа mypassword.
4. Отправляйте задание на проверку.

## Модуль 6. Интеграции Kubernetes

### Тема 2: CI/CD

#### Практическое задание

1. Данное задание предполагает ручную проверку ментором, после создания окружения НЕ НАЖИМАЙТЕ кнопку автопроверки. Для отправки результатов предусмотрена отдельная форма.
2. Создайте репозиторий в группе [https://gitlab.rebrainme.com/kubernetes\\_yandex\\_users\\_repos/<your\\_gitlab\\_id>/](https://gitlab.rebrainme.com/kubernetes_yandex_users_repos/<your_gitlab_id>)
3. Склонируйте код приложения, расположенный в репозитории, и сохраните его в созданный репозиторий в п.2. Проект реалистичный, поэтому сложилась следующая ситуация: чарт устарел, поэтому вы можете столкнуться с проблемами. Учитывая, что вы знакомы с Helm, при необходимости внесите корректировки самостоятельно в своём репозитории.
4. Запустите базу данных PostgreSQL в Kubernetes-кластере с созданием Persistent Volume для данных (можно использовать готовые helm-чарты из публичных репозиториев).

5. Запустите очередь сообщений RabbitMQ в Kubernetes-кластере с созданием Persistent Volume для данных (можно использовать готовые helm-чарты из публичных репозиториев).

[...]

### Промежуточный контроль

Промежуточный контроль проводится при изучении модуля 4. 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 автоматическим решиме в СДО.

## Модуль 4. Продвинутые практики

### Тестирование:

1. Выберите НЕВЕРНОЕ утверждение об объектах-контроллерах.
  - Deployment может управлять rolling updates
  - Deployment порождает поды
  - Replicaset порождает поды
  - Deployment порождает Replicaset
2. Выберите корректное продолжение для утверждения. «Данные поля nodeSelector помогают»:
  - компоненту “scheduler” определить, куда размещать поды
  - компоненту “api-server” осуществить поиск ноды
  - компоненту “kubelet” выбрать ноду, где запускать контейнер
  - сервис-аккаунтам определять доступные ноды
3. Какова роль scheduler?
  - Анализирует лимиты и запросы подов и размещает под на ноде
  - Анализирует лимиты и запросы подов и добавляет в манифест имя ноды, где должен быть запущен под
  - Анализирует лимиты и запросы подов и отправляет задание kubelet на старт пода
  - Анализирует количество реплик подов и поддерживает их в актуальном состоянии

### Итоговое оценивание

В конце программы обучающиеся сдают итоговую аттестацию, которая состоит из тестирования и практического задания.

### Тестирование:

- поможет оценить ваш личный прогресс по программе «Kubernetes: Yandex Cloud»;
- подсветит темы, которые стоит повторить перед финальным практическим заданием;
- позволит сравнить ваши текущие знания с результатами входного теста.

### Что ждёт в тестировании?

Тестирование включает три модуля, отражающих ключевые компетенции, заложенные в программе:

- базовые абстракции Kubernetes;
- работа с хранением и сетями;
- продвинутые практики.

Вы ответите на 20 вопросов, охватывающих указанные темы. Тестирование состоит из вопросов с одним или несколькими верными ответами.

За каждый правильный ответ вы получаете 1 балл. Максимально можно набрать 29 баллов, так как баллы начисляются за каждый верный ответ (если в вопросе несколько верных опций, суммарно за вопрос можно получить более 1 балла).

В тесте нет минимального порога прохождения. Его цель — помочь вам лучше понять свои сильные стороны и области, требующие повторения, прежде чем вы начнёте практическое задание.

### **Рекомендации по прохождению**

- Выделите 30 минут и найдите место, где вас не будут отвлекать.
- Внимательно читайте инструкции перед каждым вопросом.
- Выбирайте наиболее подходящий ответ, если затрудняетесь. Не ищите ответы в интернете — это искажает оценку вашего уровня знаний.
- Помните: тест — это часть обучения, а не экзамен.

Обратите внимание: тест доступен для прохождения только один раз, чтобы отразить ваш реальный прогресс.

### **Базовые абстракции.**

1. Базовые абстракции. Сфера применения Kubernetes. Какие из указанных функций может выполнять Kubernetes? Выберите 3 верных ответа.

- A. Автоматическое восстановление данных из резервных копий после сбоя
  - B. Добавление подов при возрастании нагрузки на приложение
  - C. Автоматическое создание контейнеров при хранении кода в Git-репозитории
- [...]

3. Базовые абстракции. Namespaces, labels, annotations. Какими характеристиками обладают лейблы в отличии от аннотаций? Выберите 2 верных ответа.

- A. Используются для группировки объектов
  - B. Изменяют поведение узлов
  - C. Являются обязательными атрибутами
- [...]

### **Хранение данных и сеть.**

1. Хранение данных и сеть. Configmap, Secrets. Ваше приложение в Kubernetes использует ConfigMap, смонтированный как файл, для хранения конфигурации. Вы изменили ConfigMap, но приложение продолжает использовать старую конфигурацию. Что нужно сделать, чтобы приложение начало использовать новую конфигурацию? Выберите 1 верный ответ.

- A. Перезапустить Pod, использующий ConfigMap
- B. Пересоздать ConfigMap
- C. Изменить имя ConfigMap
- D. Вручную внести изменения в Pod

2. Хранение данных и сеть. Типы сервисов. Вы разворачиваете веб-приложение, которое должно быть доступно извне, но у вас нет поддержки внешнего IP. Какой тип сервиса подойдёт в этой ситуации? Выберите 1 верный ответ.

- A. ClusterIP
- B. NodePort
- C. LoadBalancer
- D. ExternalName

## Продвинутые практики

1. Продвинутые практики. Управление контроллерами: StatefulSet, DaemonSet. Какие политики по обновлению своих подов поддерживает Daemonset? Выберите 2 верных ответа.

- A. recreate
  - B. OnDelete
  - C. RollingUpdate
- [...]

2. Продвинутые практики. Планирование и размещение подов. Какие команды могут остановить/возобновить scheduling подов на ноду? Выберите 2 верных ответа: один на остановку, другой на возобновление.

- A. kubectl drain node
- B. kubectl undrain node
- C. kubectl run node
- D. kubectl uncordon node

В зависимости от суммы баллов за тестирование изучите обратную связь, при необходимости повторите необходимые темы и переходите к решению финального практического задания.

Если вы набрали 23 и более баллов, поздравляем с успешным завершением итогового тестирования! 🎉 Вы продемонстрировали высокий уровень владения материалом практикума «Kubernetes: Yandex Cloud». Используйте ваши результаты как ориентир перед финальным практическим заданием, а также в работе с Kubernetes в будущем. Успехов и переходите на следующую страницу к задаче!

Если вы набрали менее 23 баллов, ваш результат показывает области, которые стоит повторить перед финальным практическим заданием. Не расстраивайтесь, это всего лишь один из этапов обучения. Проанализируйте свои ответы, освежите знания, уточните детали у менторов и двигайтесь дальше! Вы на правильном пути! 🚀 Как будете готовы, переходите на следующую страницу к задаче!

## Финальное задание

Данное задание предполагает ручную проверку ментором, после создания окружения НЕ НАЖИМАЙТЕ кнопку автопроверки. Для отправки результатов предусмотрена отдельная форма.

У вас есть приложение (<https://github.com/librespeed/speedtest>)

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).

Руководитель ставит задачу, чтобы через пару дней оно само выкатывалось на прод, и ещё где-то он слышал про helm, его тоже необходимо использовать. И чтобы секреты секретные были.

## Требования к настройкам приложения

- Приложение должно быть запущено во frontend-режиме с хранением телеметрии, обfuscацией 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 провиженер 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

[...]

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.