



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

A handwritten signature in blue ink, appearing to be 'Е.А. Фролова'.

Фролкина Е.А.

«17» июня 2025 г.

**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА  
– ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ  
«АДМИНИСТРИРОВАНИЕ КЛАСТЕРОВ KUBERNETES:  
НАСТРОЙКА, УПРАВЛЕНИЕ, БЕЗОПАСНОСТЬ»**

**Срок реализации:** 1 месяц

**Количество часов:** 51 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 «Администрирование кластеров Kubernetes: настройка, управление, безопасность» (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, уже имеющих опыт работы с контейнеризацией и базовыми возможностями Kubernetes. Она будет полезна:

- системным администраторам и DevOps-инженерам, отвечающим за эксплуатацию и стабильность инфраструктуры;
- сетевым инженерам и SRE-инженерам, участвующим в построении отказоустойчивых систем;
- техническим и релизным менеджерам, взаимодействующим с командами разработки и эксплуатации;
- разработчикам и тестировщикам, работающим с приложениями, развернутыми в Kubernetes, и заинтересованным в глубоком понимании инфраструктуры.

### **Требования к входным знаниям обучающегося:**

Для успешного освоения программы слушатели должны:

- владеть основами Linux (работа с командной строкой, базовая администрирование);
- иметь практический опыт работы с контейнерами (Docker или Podman);
- понимать архитектуру Kubernetes и уметь работать с его базовыми объектами (Pods, Deployments, Services, ConfigMaps, Secrets);
- разбираться в сетевых концепциях (IP, DNS, балансировка нагрузки, фаерволы);
- иметь опыт работы с системами мониторинга и журналирования будет преимуществом.

### **Актуальность реализации:**

Kubernetes стал де-факто стандартом для оркестрации контейнеров, и всё больше компаний выбирают self-hosted кластеры для гибкости, контроля и экономии ресурсов.

Однако их администрирование связано с высокой сложностью: настройка, обновление, обеспечение безопасности и мониторинга требуют глубоких знаний и практики.

Программа ориентирована на специалистов, которым необходимо не только понимать прикладную работу с Kubernetes, но и уметь управлять самим кластером: от инициализации и масштабирования до обеспечения отказоустойчивости и мониторинга. Полученные навыки позволяют повысить стабильность и надёжность инфраструктуры, снизить риски сбоев и обеспечить выполнение SLA.

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

- Практическая направленность: акцент на администрирование кластеров Kubernetes в реальных сценариях, включая обновления, масштабирование и HA-конфигурации.
- Фокус на инструменты развёртывания: изучение kubectl и Kubespray как ключевых подходов к установке и поддержке кластеров.
- Интеграция с экосистемой: работа с Helm, CertManager, Ingress-контроллером, а также развёртывание ELK и Prometheus stack для мониторинга и логирования.
- Безопасность и надёжность: внимание уделяется вопросам обновления сертификатов, настройке квот, вертикальному масштабированию и работе с custom metrics.
- Совмещение базовых и продвинутых задач: от инициализации до построения отказоустойчивой инфраструктуры, что позволяет системно освоить администрирование Kubernetes.

**Объем и срок освоения программы:** 51 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 комплексные навыки администрирования кластеров Kubernetes, включая установку, настройку, обновление, обеспечение отказоустойчивости и безопасность, а также освоение инструментов мониторинга и автоматизации.

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

Обучающие:

- освоить механизмы развёртывания кластеров с помощью kubectl и kubespray;
- изучить процедуры обновления и обслуживания кластеров, включая управление сертификатами;
- научиться организовывать HA-конфигурации для повышения отказоустойчивости;
- овладеть инструментами управления ресурсами (квоты, масштабирование);
- освоить внедрение ключевых инструментов экосистемы: Helm, Ingress-контроллер, CertManager, ELK-стек, Prometheus stack и Prometheus adapter.

Развивающие:

- развить умение проектировать и поддерживать надежные и масштабируемые инфраструктурные решения;

- научиться анализировать производительность и стабильность кластера на основе метрик и логов;
- сформировать практические навыки работы с инструментами обеспечения безопасности и отказоустойчивости.

#### **Воспитательные:**

- формировать ответственное отношение к эксплуатации инфраструктуры и данным пользователей;
- прививать культуру DevOps-практик: автоматизация, прозрачность процессов, командная работа;
- развивать стремление к профессиональному росту и внедрению современных технологий в повседневную практику.

#### **Планируемые результаты:**

##### **Знания:**

- архитектуру Kubernetes и особенности self-hosted кластеров;
- роль и устройство etcd в хранении состояния кластера;
- возможности и различия инструментов развертывания (kubeadm, kubespray);
- методы обеспечения отказоустойчивости и HA-конфигураций;
- подходы к управлению ресурсами: квоты, вертикальное масштабирование;
- принципы работы и применения Helm, Ingress-контроллера, CertManager, ELK-стека, Prometheus stack и Prometheus adapter;
- механизмы обеспечения безопасности и обновления сертификатов.

##### **Умения:**

- устанавливать и обслуживать кластеры Kubernetes с помощью kubeadm и kubespray;
- обновлять кластеры и сертификаты, обеспечивая минимальные простои;
- организовывать и поддерживать отказоустойчивые конфигурации;
- применять механизмы распределения ресурсов и масштабирования в кластере;
- интегрировать инструменты мониторинга и логирования для контроля состояния инфраструктуры;
- использовать Helm для пакетного развертывания приложений и сервисов.

##### **Навыки:**

- администрирования Kubernetes-кластера в реальных условиях;
- диагностики и устранения проблем, связанных с инфраструктурой и приложениями;
- практической настройки и эксплуатации инструментов мониторинга, логирования и сертификации;
- построения безопасной, стабильной и масштабируемой 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 «Ребреин», и получение при необходимости дополнительного профессионального образования педагогической направленности;

в) успешно прошедшее промежуточной аттестации не менее чем за два года обучения по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующей направленности дополнительной общеобразовательной общеразвивающей программе;

2) не имеющее ограничений на занятие педагогической деятельностью, установленных законодательством Российской Федерации;

3) прошедшее обязательный предварительный (при поступлении на работу) и периодические медицинские осмотры (обследования), а также внеочередные медицинские осмотры (обследования) в порядке, установленном законодательством Российской Федерации.

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

## 2. УЧЕБНЫЙ ПЛАН

№ п/ п	Наименование модуля	Количество часов			Формы контроля / аттестация
		Всего	Теория	Практика	
1	Модуль 1 “Онбординг”	2	1	1	Входное тестирование
2	Модуль 2 “Kubernetes Admin”	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”	2			
2	Модуль 2 “Kubernetes Admin”	12	13	13	2
3	Итоговая аттестация				7   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, необходимые для прохождения программы Kubernetes Admin:

- Основы Kubernetes;
- Работа с Git;
- Работа с Ansible.

Каждый блок состоит из 8 вопросов, призванных убедиться, что слушатель владеет достаточными навыками для освоения программы.

#### 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. Kubernetes Admin**

Теория 15 академ. ч. Практика 20 академ. ч.

Модуль состоит из следующих тем:

#### Тема 1. Зачем вам self-hosted

Содержание: Обзор применимости такого подхода. Какие плюсы/минусы. Зачем нужно уметь устанавливать self-hosted, даже если вам этот метод не подходит сейчас. Тестирование.

#### Тема 2. Установка и обслуживание etcd

Содержание: Как управлять хранилищем. Инициализация кластера. Алгоритм работы при возникновении проблем с нодами etcd. Как делать резервные копии. Практическое задание.

#### Тема 3. Kubeadm. Инициализация кластера kubernetes.

Содержание: Основные команды kubeadm. Инициализация кластера при помощи kubeadm. Установка containerd. Инициализация мастер-ноды. Присоединение worker-нод. Сертификаты. Трудности при развёртывании через kubeadm. Практическое задание.

#### Тема 4. Kubeadm. Обновление сертификатов и обслуживание кластера

Содержание: Кластер каким он должен быть. Обновление сертификатов. Добавление/удаление нод. Практическое задание.

#### Тема 5. Kubeadm. Обновление кластера

Содержание: на что следует обратить внимание при обновлении кластера. Какие шаги включает в себя процесс обновления кластера. Каким образом можно обновить кластер. Практическое задание.

#### Тема 6. Инструменты развёртывания. Kubespray

Содержание: что такое kubespray и его особенности. Как развернуть кластер с kubespray. Практическое задание.

Тема 7. Kubespray. Обновление кластера

Содержание: Возьмём определенную ветку kubespray, установим кластер. Обновимся до новой версии. Практическое задание.

Тема 8. Kubespray и kubeadm. HA-конфигурация

Содержание: Варианты обеспечения высокой доступности. Рассмотрим как настраивается HA-конфигурация. Модуль kube-vip.

Тема 9. Распределение ресурсов. Квоты

Содержание: Зачем нужны квоты. LimitRange. Как применяются квоты по ресурсам. ResourceQuota. Практическое задание.

Тема 10. Вертикальное масштабирование

Содержание: Что такое вертикальное масштабирование. Когда и зачем нужно вертикальное масштабирование. Как осуществляется вертикальное масштабирование сторонним сервисом. Практическое задание.

Тема 11. Введение в Helm

Содержание: Основные концепции и применение Helm в управлении приложениями в Kubernetes. Базовая настройка среды. Основные команды Helm. Как использовать плагины и команду helm create для создания собственных чартов Helm. Практическое задание.

Тема 12. Развёртывание Ingress-контроллера

Содержание: Как можно расширять возможности кластера. Установка и настройка ingresscontroller. Практическое задание.

Тема 13. Certmanager

Содержание: Зачем нужен certmanager. Когда нужен certmanager. Как установить и настроить certmanager. Как выпустить сертификаты с помощью Let's encrypt. Практическое задание.

Тема 14. Развёртывание ELK-стека

Содержание: Необходимость логирования. Что и как логировать. Развертывание полного решения по логированию кубер на elastic stack. Практическое задание.

Тема 15. Развёртывание prometheus stack

Содержание: Для чего настраивать мониторинг на Kubernetes. Развертывание prometheus и grafana с Kubernetes. Какие метрики отслеживать. Настройка алертов. Практическое задание.

Тема 16. Развёртывание Prometheus adapter. Custom metrics

Содержание: Настройка адаптера для сбора метрик. Настройка масштабирования.

Модуль направлен на формирование у обучающихся 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 1-3 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/>
3. Официальный сайт Kubernetes [Электронный ресурс]: <https://kubernetes.io/>
4. Официальная документация по kubectl [Электронный ресурс]: <https://kubernetes.io/docs/setup/production-environment/tools/kubectl/create-cluster-kubectl/>
5. Документация по prometheus-адаптер [Электронный ресурс]: <https://github.com/kubernetes-sigs/prometheus-adapter/blob/master/docs/walkthrough.md>

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

### **Оценочные материалы:**

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

### **Входное тестирование**

Тестирование состоит из трёх частей, проверяющих компетенции, необходимые для прохождения программы Kubernetes Admin:

- Основы Kubernetes;
- Работа с Git;
- Работа с Ansible.

Каждый блок состоит из 8 вопросов, призванных убедиться, что вы владеете достаточными навыками для освоения программы, соответственно, что вам будет комфортно изучать материал.

Каждый вопрос проверяет один из необходимых аспектов компетенции, чтобы вы могли определить свои сильные стороны и понимать, что нужно освежить в памяти при необходимости.

За каждый правильный ответ вам начисляется по одному баллу. Всего вы можете набрать 26 баллов (так как в одном вопросе есть несколько верных ответов). Тест считается успешно пройденным, если вы выберете более 85% верных ответов — наберёте не менее 22 баллов.

### **Основы Kubernetes**

1. Какой объект Kubernetes используется для управления обновлением и масштабированием приложений?
  - Pod
  - Deployment
  - Service
  - Ingress
2. Вы создали Ingress ресурс, но при выполнении команды `kubectl get ingresses` IP-адрес не появился. Какова может быть причина этого?
  - Неверно настроен ConfigMap
  - Ошибка в конфигурации деплоя
  - Неправильно настроены лейблы на подах
  - Неверный ingressClass

### **Работа с Git**

1. Какой командой вы можете клонировать репозиторий из удалённого источника?
  - `git fork $REPO`
  - `git clone $REPO`
  - `git init $REPO`
  - `git pull $REPO`
2. Какой командой вы можете создать новую ветку `new-branch` и переключиться на неё?
  - `git checkout -b new-branch`
  - `git branch new-branch`
  - `git switch new-branch`
  - `git create new-branch`

### **Работа с Ansible.**

1. Какой командой запустить плейбук `cluster.yml` с использованием инвентори `inventory`?
  - `ansible-playbook cluster.yml`
  - `ansible-playbook inventory cluster.yml`
  - `ansible-playbook -i inventory cluster.yml`
  - `ansible-playbook -inventory inventory cluster.yml`
2. Какой формат используется для описания статического инвентори файла в Ansible?
  - CSV
  - INI
  - JSON
  - XML

### **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 2. Установка и обслуживание etcd**

### **Практическое задание**

Обратите внимание, в данной инфраструктуре вы будете получать доступ к виртуальным машинам через балансировщик. Внешних ip-адресов на интерфейсах не будет. При настройке везде указывайте внутренние.

- Инициализируйте трёхнодовый кластер etcd.
- Не забудьте поменять статус кластера на existing в конфигурационных файлах
- Запишите в него значение etcdctl put key "nice job".
- Далее рекомендуем отправить задание на проверку и после того как она пройдёт, потренируйтесь в ломании и поднимании кластера etcd (это на тот случай, если починить не получится).

## **Тема 9. Распределение ресурсов. Квоты**

### **Практическое задание**

1. Ограничьте использование ресурсов в namespace default:
  - Имя манифеста должно быть ns-limit;
  - Максимальное потребление CPU — 4 ядра, memory — 4 G;
  - Не более 20 подов.
2. Ограничьте потребление ресурсов на под:
  - Имя манифеста должно быть pod-limit;
  - Не более 1 ядра;
  - Не более 1 гигабайта памяти;
  - Минимальные запросы: 200m и 128 Mi.
3. Отправьте задание на проверку.

### **Итоговое оценивание**

В конце программы обучающиеся сдают итоговую аттестацию.

### **Финальное задание**

Итак, настало время последнего рывка в рамках текущего практикума — финального практического задания.

За время прохождения программы вы узнали многое про администрирование Kubernetes, а теперь вас ждёт комплексная рабочая задача, где всё как в реальной жизни: мало конкретики, размытое описание желаемого результата и свобода творчества.

### **Условия задачи**

Ваша организация купила ПО, и в процессе выяснилось, что оно ориентировано только на эксплуатацию в Kubernetes, и у вендора нет альтернативных способов инсталляции.

Вам нужно подготовить надёжный кластер, чтобы внедрение ПО прошло успешно. При этом вендор не хочет ничего делать. Поэтому вы должны дать ему базовые инструменты, иначе он будет просить вас почитать логи из подов.

Напомним, что ноды переименовывать нельзя, это может сломать интеграцию

#### Ваши действия

1. Используя kubespray, разверните кластер, где один сервер выделен для control plane, два для worker nodes.
2. Включите HA-режим. IP-адрес балансировщика будет предоставлен при создании инфраструктуры.
3. Включите cert-manager. Создайте clusterissuer на продуктовые сервера let's encrypt.
4. Включите local-path-provisioner.

#### Дополнительные модули

1. Разверните Ingress-контроллер ingress-nginx из официального helm-чарта (после установки запустите автопроверку разово, она настроит интеграцию с облаком)
2. Настройте сборку логов при помощи EFK/Opensearch.
  - Fluentd должен быть развёрнут в namespace logging, elastic/kibana в namespace elasticsearch.
3. Подготовьте и разверните helm-чарты, проверьте работоспособность.
4. Разверните в namespace prometheus prometheus-оператор и проверьте наличие стандартных правил алертинга.
5. Если будете публиковать какие-либо веб-интерфейсы на ingress, закройте их при помощи basic-auth.
  - По желанию также можете получить сертификаты на эти эндпоинты.

#### Резервное копирование

1. Настройте резервное копирование etcd. Пусть сохраняется в папку и хранится 5 последних копий.
  - Чтобы сделать это, напишите скрипт, добавьте его в cron и проверьте, что происходит выгрузка etcd.

#### Немспейс team-one

Создайте неймспейс team-one и настройте LimitRange и ResourceQuota, где:

- суммарное ограничение на неймспейс не более 4x CPU и 4 GB RAM;
- минимальные запросы для контейнера 100m и 64 MB RAM;
- стандартный запрос контейнеров на ресурсы минимален;
- максимальное потребление для контейнера — 1 CPU и 1 GB.

#### Автоматическая проверка

- наличие 1 control plane и 2 воркер-ноды,
- работу балансировщика,
- ресурсы и лимиты в неймспейсе team-one,
- наличие cert-manager,
- наличие Ingressclasses.

#### Ручная проверка

Для детальной проверки ментором приложите скриншоты всех команд из списка ниже:

- kubectl get nodes -o wide;
- kubectl get all -n elasticsearch;
- kubectl get all -n prometheus;
- kubectl get all -n logging;
- kubectl get pv;
- kubectl get pvc -A.

Также просьба оформить в репозиторий все использованные компоненты и прислать ссылку.

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.