



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 ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ
«ПРОЕКТИРОВАНИЕ И СОПРОВОЖДЕНИЕ ВЫСОКОНАГРУЖЕННЫХ СИСТЕМ»**

Срок реализации: 6 месяцев.
Количество часов: 320 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 профессиональной переподготовки «Проектирование и сопровождение высоконагруженных систем» (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:

- Data-инженеры (Middle/Senior) – специалисты, работающие с высоконагруженными системами, потоками данных и требующие понимания масштабируемости и отказоустойчивости инфраструктуры.
- Backend-разработчики – создают распределённые приложения и микросервисные архитектуры, которым важно понимать нагрузку, балансировку, очереди сообщений и мониторинг.
- DevOps-инженеры (Middle/Senior) – ответственны за развертывание, поддержку и масштабирование высоконагруженных систем, настройку балансировщиков, очередей и мониторинга.

Требования к входным знаниям обучающегося:

- Базовые навыки системного администрирования Linux – работа в консоли, управление процессами, понимание файловой системы и прав пользователей.
- Основы сетевых технологий – IP, TCP/UDP, DNS, NAT, базовые команды для диагностики сети.
- Базовое понимание архитектуры веб-приложений – клиент-сервер, REST API, базы данных.
- Опыт работы с контейнерами и виртуализацией – Docker, понимание образов и контейнеров, базовые навыки работы с Kubernetes будут плюсом.
- Знание языков программирования – хотя бы один язык для написания скриптов и тестовых приложений (Python, Bash или Java).

Актуальность реализации:

Современные веб-приложения и сервисы постоянно сталкиваются с ростом нагрузки, увеличением объёма данных и требованиями к высокой доступности. Компании стремятся к масштабируемости своих решений, отказоустойчивости и низкой задержке отклика, что делает критически важным понимание принципов проектирования высоконагруженных систем. Знания в области мониторинга, балансировки, распределённых очередей и оптимизации ресурсов позволяют инженерам создавать стабильные и производительные сервисы, способные выдерживать миллионы запросов в секунду.

Отличительные особенности программы:

- Практическая направленность — акцент на тестирование, нагрузочное моделирование и мониторинг реальных приложений.
- Широкий охват инструментов — изучение JMeter, Siege, Zabbix, EFK, Prometheus, Grafana, RabbitMQ, Kafka, Nginx, HAProxy, Serverless и других технологий.
- Разбор архитектурных паттернов — от двухзвенной и трёхзвенной архитектуры до Serverless, включая отказоустойчивость и масштабирование.
- Фокус на производительность — инструменты анализа CPU, RAM, I/O, сети, а также индексы Apdex и методы оценки SLA.
- Интеграция с облачными и контейнерными решениями — Docker, облачные балансировщики, контейнеризация сервисов и мониторинг контейнеризированных приложений.
- Реальные кейсы и имитация нагрузок — работа с тестовыми стендами и сценариями, максимально приближёнными к промышленным условиям.

Объём и срок освоения программы: 320 академ. ч. в течение 6 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:

Цель программы — дать участникам системное понимание построения высоконагруженных, отказоустойчивых и масштабируемых сервисов, а также навыки мониторинга, нагрузочного тестирования и оптимизации производительности компонентов системы.

Программа направлена на решение следующих основных задач:

Обучающие:

- Освоение инструментов мониторинга (Zabbix, Prometheus, EFK, Grafana) и нагрузочного тестирования (JMeter, Siege).
- Изучение принципов балансировки нагрузки, настройки веб-серверов и app-серверов.
- Понимание архитектурных паттернов и распределённых систем (двухзвенная, трёхзвенная, Serverless).

Развивающие:

- Формирование навыков анализа узких мест и оптимизации производительности систем.
- Развитие умения работать с распределёнными системами и управлять кластеризацией сервисов.
- Совершенствование компетенций в создании отказоустойчивых и масштабируемых решений.

Воспитательные:

- Формирование ответственности за стабильность и производительность сервисов.
- Развитие культуры тестирования и мониторинга приложений в условиях высоких нагрузок.
- Осознание важности комплексного подхода при проектировании и сопровождении систем.

Планируемые результаты:

Знания:

- Основные принципы HighLoad-систем и их архитектур;
- Метрики нагрузки CPU, RAM, I/O и сети;
- Инструменты нагрузочного тестирования и мониторинга (JMeter, Siege, Zabbix, Prometheus, EFK/Graylog);
- Принципы построения отказоустойчивых и масштабируемых систем;
- Особенности работы с балансировкой, очередями сообщений (RabbitMQ, Kafka) и serverless-архитектурами;
- Основы настройки веб-серверов, app-серверов и распределённых систем хранения (Hadoop, Ignite, Memcached).

Умения:

- Анализировать и выявлять узкие места в производительности систем;
- Настраивать и проводить нагрузочные тесты для оценки стабильности и пропускной способности сервисов;
- Проектировать и реализовывать архитектурные решения для высоконагруженных систем;
- Настраивать балансировку, кластеры, очереди сообщений и serverless-компоненты;
- Осуществлять мониторинг и визуализацию показателей производительности;
- Оптимизировать работу приложений и систем на основе полученных данных.

Навыки:

- Работа с инструментами тестирования нагрузки (JMeter, Siege);
- Мониторинг и визуализация метрик (Zabbix, Prometheus, Grafana, EFK/Graylog);
- Настройка отказоустойчивых архитектур, веб- и app-серверов;
- Работа с кластерами данных и очередями сообщений (Hadoop, Ignite, Memcached, RabbitMQ, Kafka);
- Применение best practices для повышения производительности и стабильности сервисов.

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. Что такое HighLoad и где он начинается	6	3	3	Практическое задание
3	Модуль 3. Средства проверки гипотез и решений	146	48	98	Практическое задание
4	Модуль 4. Архитектурные паттерны при росте нагрузок	121	37	84	Практическое задание
5	Модуль 5. Средства обеспечения высокой производительности и отказоустойчивости	55	21	34	Практическое задание

6	Итоговая аттестация.	20		20	Итоговое практическое задание
---	----------------------	----	--	----	-------------------------------

3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование модуля	1 месяц	2 месяц	3 месяц	4 месяц	5 месяц	6 месяц
1	Модуль 1. Онбординг	2					
2	Модуль 2. Что такое HighLoad и где он начинается	6					
3	Модуль 3. Средства проверки гипотез и решений	50	60	36			
4	Модуль 4. Архитектурные паттерны при росте нагрузок			22	58	41	
5	Модуль 5. Средства обеспечения высокой производительности и отказоустойчивости					17	38
6	Итоговая аттестация.						20

4. РАБОЧАЯ ПРОГРАММА

Модуль 1. Онбординг

Теория 1 академ. ч. Практика 1 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. Что такое HighLoad и где он начинается

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

Модуль состоит из следующих тем:

Тема 1: Basics. Введение в HighLoad

Содержание: Терминология курса. Что такое HighLoad. С чего начинать работу. Как подготовиться к высоким нагрузкам. Практическое задание.

Тема 2: Общее понимание ресурсозатрат

Содержание: Базовые положения. Практическое задание.

Модуль 3. Средства проверки гипотез и решений

Теория 48 академ. ч. Практика 98 академ. ч.

Модуль состоит из следующих тем:

Тема 1: Метрики. CPU

Содержание: Основные метрики. CPU. Чем занят процессор. Load Average. Приоритеты процессов (Niceness). Сбор информации о CPU в Linux. Практическое задание.

Тема 2: Анализ нагрузки и управление RAM

Содержание: OOM killer. Cgroups. Практическое задание.

Тема 3: Анализ нагрузки и управление вводом-выводом. Диски.

Содержание: Типы дисков. Характеристики дисков. fio. Практическое задание.

Тема 4: Анализ нагрузки. Исторические данные

Содержание: Утилиты: Sar, Atop. Практическое задание.

Тема 5: Анализ нагрузки и управление вводом-выводом. Сеть.

Содержание: iftop. vnstat. Практическое задание.

Тема 6: Мониторинг. Установка тестового приложения

Содержание: Практическое задание.

Тема 7: Мониторинг. Время ответа

Содержание: Время ответа. Developer tools. Curl. Практическое задание.

Тема 8: Мониторинг. JMeter. Первое знакомство

Содержание: Установка JMeter. Основные компоненты интерфейса. Создание простого теста. Практическое задание.

Тема 9: Мониторинг. JMeter. Запись сценария. Поиск сложных запросов

Содержание: Запись сценариев через HTTP Proxy. Поиск и выделение проблемных запросов. Практическое задание.

Тема 10: Мониторинг. JMeter. Assertions, Variables, Pre- Post- processors, Logic Controllers

Содержание: Использование Assertions и Variables. Применение Pre- и Post-processors. Логические контроллеры. Практическое задание.

Тема 11: Мониторинг. JMeter. Plugins

Содержание: Установка и использование плагинов для расширения функционала JMeter. Практическое задание.

Тема 12: Мониторинг. JMeter. Мониторинг ресурсов

Содержание: Настройка мониторинга CPU, RAM, I/O в связке с JMeter. Практическое задание.

Тема 13: Мониторинг. JMeter. Стресс-тестирование

Содержание: Проведение стресс-тестов. Анализ поведения системы при перегрузке. Практическое задание.

Тема 14: Мониторинг. JMeter. Тестирование стабильности

Содержание: Тестирование устойчивости под длительной нагрузкой. Сбор статистики. Практическое задание.

Тема 15: Средства нагрузочного тестирования. Siege
Содержание: Установка и обзор возможностей Siege. Применение для нагрузочного тестирования. Практическое задание.

Тема 16: Средства нагрузочного тестирования. Siege. Нагрузочное тестирование
Содержание: Создание сценариев и проведение тестов с различной интенсивностью. Практическое задание.

Тема 17: Alpine Linux: легковесная сборка для контейнеров
Содержание: Особенности Alpine Linux. Минимальные образы для контейнеров. Практическое задание.

Тема 18: Средства нагрузочного тестирования. Alpine. Собственный имидж на базе Alpine
Содержание: Создание собственного Docker-образа на базе Alpine. Практическое задание.

Тема 19: Индексы производительности Apdex
Содержание: Понятие Apdex, формула расчета и интерпретация показателей. Практическое задание.

Тема 20: Мониторинг. Zabbix. Docker-appliance
Содержание: Развертывание Zabbix в Docker. Настройка базового мониторинга. Практическое задание.

Тема 21: Мониторинг. Zabbix. Колоночное хранилище
Содержание: Настройка колоночного хранилища для сбора метрик. Практическое задание.

Тема 22: Мониторинг. Zabbix. Автодобавление активного агента
Содержание: Настройка автообнаружения и автодобавления агентов. Практическое задание.

Тема 23: Мониторинг. Zabbix. Низкоуровневое обнаружение (LDD)
Содержание: Настройка Low-Level Discovery для динамических объектов. Практическое задание.

Тема 24: Мониторинг. Zabbix Proxy
Содержание: Настройка Zabbix Proxy для распределенного мониторинга. Практическое задание.

Тема 25: EFK. Единичный экземпляр
Содержание: Установка Elasticsearch, Fluentd и Kibana. Основные возможности. Практическое задание.

Тема 26: EFK. Настройка агента
Содержание: Конфигурация Fluentd/Fluent Bit для сбора логов. Практическое задание.

Тема 27: KQL (Kibana Query Language)
Содержание: Основы KQL. Составление запросов и фильтров в Kibana. Практическое задание.

Тема 28: Распределенный EFK кластер

Содержание: Настройка кластерного Elasticsearch. Репликация, шардирование и масштабирование. Практическое задание.

Тема 29: Graylog vs EFK

Содержание: Сравнение возможностей Graylog и EFK. Особенности работы с логами. Практическое задание.

Тема 30: Prometheus: установка

Содержание: Развертывание Prometheus. Основные компоненты и конфигурация. Практическое задание.

Тема 31: Prometheus: Node exporter

Содержание: Установка и настройка Node exporter. Сбор системных метрик. Практическое задание.

Тема 32: Prometheus: Alertmanager

Содержание: Настройка Alertmanager. Конфигурация оповещений. Практическое задание.

Тема 33: Prometheus: производительность Docker

Содержание: Сбор и анализ метрик Docker-контейнеров. Практическое задание.

Тема 34: Prometheus: производительность Kubernetes

Содержание: Мониторинг кластера Kubernetes с Prometheus. Практическое задание.

Тема 35: Prometheus: визуализация метрик в Grafana

Содержание: Настройка Grafana для отображения метрик. Создание дашбордов. Практическое задание.

Тема 36: Оповещения. Интеграция с Telegram

Содержание: Настройка оповещений Prometheus/Zabbix через Telegram. Практическое задание.

Тема 37: Оповещения. Интеграция со Slack

Содержание: Настройка уведомлений в Slack. Практическое задание.

Модуль 4. Архитектурные паттерны при росте нагрузок

Теория 37 академ. ч. Практика 84 академ. ч.

Модуль состоит из следующих тем:

Тема 1: Разделение приложения на двухзвенную архитектуру

Содержание: Понятие двухзвенной архитектуры. Логика разделения клиентской и серверной части. Практическое задание.

Тема 2: Двухзвенная архитектура в контейнерах. Сравнение по производительности с решением без Docker

Содержание: Развертывание двухзвенной архитектуры в Docker. Сравнение метрик производительности. Практическое задание.

Тема 3: Разделение приложения на трехзвенную архитектуру

Содержание: Понятие трехзвенной архитектуры. Логика разделения на веб-, прикладной и базовый слои. Практическое задание.

Тема 4: Сертификаты TLS RSA vs ECC. Подготовка

Содержание: Основы работы с TLS, отличие RSA и ECC. Генерация ключей и подготовка к установке. Практическое задание.

Тема 5: Сертификаты TLS RSA vs ECC. Nginx

Содержание: Настройка TLS в Nginx с использованием RSA и ECC. Практическое задание.

Тема 6: Настройка балансировки. Nginx

Содержание: Конфигурация Nginx для балансировки нагрузки. Round-robin, Least Connections. Практическое задание.

Тема 7: Настройка балансировки. HAProxy

Содержание: Установка и настройка HAProxy для балансировки TCP и HTTP трафика. Практическое задание.

Тема 8: Настройка балансировки. AWS ELB. Подготовка

Содержание: Обзор AWS ELB. Настройка окружения для балансировки через облако. Практическое задание.

Тема 9: Stateless. Тестовый стенд API — 1 нода

Содержание: Разработка и тестирование stateless API на одной ноде. Практическое задание.

Тема 10: Cloud. Тестовый стенд API + RDS

Содержание: Развертывание API с базой данных RDS. Тестирование производительности. Практическое задание.

Тема 11: Очереди. Отказоустойчивый RabbitMQ. Федерация

Содержание: Настройка федерации RabbitMQ для распределенной работы. Практическое задание.

Тема 12: Очереди. Отказоустойчивый RabbitMQ. Кластер

Содержание: Конфигурация RabbitMQ-кластера. Настройка репликации и устойчивости. Практическое задание.

Тема 13: Очереди. Отказоустойчивый RabbitMQ. Shovel

Содержание: Использование Shovel для переноса сообщений между брокерами. Практическое задание.

Тема 14: RabbitMQ. Резервное копирование

Содержание: Настройка бэкапов очередей и конфигурации RabbitMQ. Практическое задание.

Тема 15: Очереди. Kafka

Содержание: Введение в Kafka. Архитектура брокера, продюсеры и консумеры. Практическое задание.

Тема 16: Очереди. Zookeeper

Содержание: Установка и конфигурация Zookeeper для Kafka. Практическое задание.

Тема 17: Очереди. Kafka. Кластеризация

Содержание: Настройка Kafka-кластера. Репликация и разделение партиций. Практическое задание.

Тема 18: Очереди. Kafka Streams

Содержание: Обработка потоков данных с Kafka Streams. Создание простого приложения. Практическое задание.

Тема 19: Очереди. Kafka, отличия от RMQ, замер производительности

Содержание: Сравнение Kafka и RabbitMQ. Метрики производительности. Практическое задание.

Тема 20: Очереди. Мониторинг. Prometheus

Содержание: Настройка Prometheus для мониторинга очередей. Практическое задание.

Тема 21: Очереди. Мониторинг. Zabbix

Содержание: Настройка Zabbix для контроля очередей. Практическое задание.

Тема 22: Serverless. Знакомство с Serverless. Serverless Framework. Python Hello world

Содержание: Установка Serverless Framework. Разработка простого Python-функционала. Практическое задание.

Тема 23: Serverless. Кастомные слои. PHP

Содержание: Создание и подключение кастомных слоев для PHP. Практическое задание.

Тема 24: Serverless-архитектура: организация балансировки через ELB

Содержание: Настройка балансировки Serverless-функций через ELB. Практическое задание.

Тема 25: Serverless-архитектура: организация балансировки через API Gateway

Содержание: Настройка балансировки и маршрутизации через API Gateway. Практическое задание.

Модуль 5. Средства обеспечения высокой производительности и отказоустойчивости

Теория 21 академ. ч. Практика 34 академ. ч.

Модуль состоит из следующих тем:

Тема 1: HTTP/2

Содержание: Основы протокола HTTP/2. Мультиплексирование потоков, серверный push, компрессия заголовков. Практическое задание.

Тема 2: Apache - различные MPM (prefork, worker, event)

Содержание: Различия моделей многопроцессной и многопоточной обработки. Настройка MPM. Практическое задание.

Тема 3: Меняем Apache на Nginx. Сравнение функционала

Содержание: Плюсы и минусы Nginx и Apache. Миграция конфигураций. Практическое задание.

Тема 4: Nginx настройка и тюнинг

Содержание: Оптимизация конфигурации Nginx под высокую нагрузку. Настройка worker_processes, buffer, cache. Практическое задание.

Тема 5: NGINX Unit single node

Содержание: Установка NGINX Unit. Развертывание одного приложения на Node.
Практическое задание.

Тема 6: NGINX балансировка нагрузки

Содержание: Настройка балансировки HTTP и TCP. Методы round-robin, least connections.
Практическое задание.

Тема 7: App-сервера для Java - развертывание и инсталляция сервера приложений Java

Содержание: Установка и конфигурация Java Application Server (Tomcat, WildFly).
Практическое задание.

Тема 8: App-сервера для Java - сборка и инсталляция приложений Java

Содержание: Развёртывание WAR/JAR приложений. Настройка окружения. Практическое задание.

Тема 9: App-сервера для Java - подготовка тестового профиля нагрузки приложения Java

Содержание: Создание тестового профиля нагрузки. Конфигурация метрик JVM.
Практическое задание.

Тема 10: App-сервера для Java - нагрузка приложения Java по памяти

Содержание: Мониторинг потребления памяти, настройка heap и garbage collector.
Практическое задание.

Тема 11: App-сервера для Java. Нагрузка приложения Java - HTTP потоки

Содержание: Настройка максимальных потоков, управление соединениями. Практическое задание.

Тема 12: Hadoop - HDFS. Деплой и настройка

Содержание: Установка и конфигурация HDFS. Основные команды работы с HDFS.
Практическое задание.

Тема 13: Hadoop - HDFS. Мониторинг

Содержание: Метрики HDFS, использование JMX и встроенных инструментов.
Практическое задание.

Тема 14: Hadoop - Hive. Деплой и настройка

Содержание: Установка Hive, подключение к HDFS, настройка metastore. Практическое задание.

Тема 15: Hadoop - Hive. Мониторинг

Содержание: Мониторинг запросов Hive, отслеживание нагрузки. Практическое задание.

Тема 16: Hadoop - Hive. Кластер

Содержание: Настройка распределенного кластера Hive и HDFS. Практическое задание.

Тема 17: Hadoop - YARN

Содержание: Управление ресурсами через YARN. Настройка queue и scheduler.
Практическое задание.

Тема 18: Hadoop - MapReduce

Содержание: Основы MapReduce, запуск задач, мониторинг выполнения. Практическое задание.

Тема 19: Mysql мониторинг

Содержание: Метрики производительности MySQL. Настройка мониторинга. Практическое задание.

Тема 20: Apache Ignite. Установка и первичная настройка

Содержание: Развёртывание Apache Ignite. Основные конфигурации. Практическое задание.

Тема 21: Балансировка между репликами MySQL через HAProxy

Содержание: Настройка HAProxy для распределения нагрузки между MySQL репликами. Практическое задание.

Тема 22: Apache Ignite. Кластер (ZooKeeper)

Содержание: Настройка кластера Ignite с использованием ZooKeeper. Практическое задание.

Тема 23: Apache Ignite. Мониторинг

Содержание: Метрики кластера, настройка мониторинга и алертов. Практическое задание.

Тема 24: Memcached - Cluster

Содержание: Развёртывание и настройка Memcached в кластере. Практическое задание.

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.

Модуль 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/highload/>
3. Официальный сайт Highload [Электронный ресурс]: <https://highload.ru/>
4. JMeter GUI: Test Plan & Workbench [Электронный ресурс]: <https://www.guru99.com/hands-on-with-jmeter-gui.html>
5. Домашняя страница Siege [Электронный ресурс]: <https://www.joedog.org/siege-home/>
6. Официальный сайт Zabbix [Электронный ресурс]: <https://www.zabbix.com/>
7. Официальный сайт Prometheus [Электронный ресурс]: <https://prometheus.io/>
8. Официальная документация Nginx [Электронный ресурс]: <https://nginx.org/ru/docs/>
9. Официальный сайт Apache Kafka [Электронный ресурс]: <https://kafka.apache.org/>

6. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

Оценочные материалы:

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

Текущий контроль

Осуществление текущего контроля проводится после занятий в виде написания практических заданий или тестирований. Тематика и условия выполнения практических заданий расписаны в личном кабинете обучающегося в СДО. Педагог проверяет решение и принимает решение о принятии решения (зачет), о необходимости доработать решение или о незачете. Если промежуточный контроль представлен в виде тестирования, подсчет верных ответов и выставление оценки «зачёт» и «незачёт» происходят в автоматическом режиме в СДО.

Модуль 2. Что такое HighLoad и где он начинается

Тема 2: Общее понимание ресурсозатрат

Задание:

1. У вас есть проект веб-приложения, для работы которого требуются:
 - 1.1. Сервер логики (CPU: 2x1.7GHz, RAM: 16GB, Storage: 300GB)
 - 1.2. Сервер БД (CPU: 4x2.5GHz, RAM: 64GB, Storage: 1TB)
 - 1.3. Сервер-хранилище (CPU: 2x1.5GHz, RAM: 16GB, Storage: 3TB)
 - 1.4. Сервер мониторинга (CPU: 2x1.7GHz, RAM: 16GB, Storage: 300GB)

Обозначьте необходимые ресурсы для работы системы.

2. Выберите три типа размещения своей инфраструктуры (например, VDS, аренда физических серверов, облако). Произведите расчет стоимости для каждого варианта.
3. Какой вариант вы бы выбрали и почему? К этому выводу стоит вернуться в конце практикума, чтобы проверить свое предположение.

Модуль 3. Средства проверки гипотез и решений

Тема 4: Анализ нагрузки. Исторические данные

Практическое задание

1. Запустите top. Каково значение wa (iowait) на тестовой системе без нагрузки?

2. Запустите тест fio из задания «Анализ нагрузки и управление вводом-выводом. Диски. Часть 1», увеличив:
 - размер блока - до 256K;
 - Количество потоков - до 256.

Изменилось ли значение wa?

3. Установите пакет iotop.
4. Выясните, какие процессы в системе наиболее активно используют ввод-вывод.
5. Запустите тест fio из задачи «Анализ нагрузки и управление вводом-выводом. Диски. Часть 1». Найдите процесс fio в выводе iotop. Является ли он теперь самым IO-активным процессом в системе?
6. Создайте следующий файл профиля fio:

[...]

Тема 31: Prometheus: Node exporter

Практическое задание

1. Создайте виртуальную машину со следующими параметрами:
 - 2 CPU,
 - 4Гб оперативной памяти,
 - 25Гб - диск.
2. На созданной виртуальной машине:
 - обновите ОС;
 - установите Docker и docker-compose.
3. Создайте файл docker-compose.yml следующего содержания:

[...]

Модуль 4. Архитектурные паттерны при росте нагрузок

Тема 3: Разделение приложения на трехзвенную архитектуру

Практическое задание

1. Для выполнения задания необходимо создать такую инфраструктуру:
 - VM1 persona со следующими параметрами: 1 CPU / 2G memory;
 - VM2 wordpress со следующими параметрами: 1 CPU / 1G memory.

Обновите систему на VM1,2.

Добавьте адреса VM1,2 в файл /etc/hosts каждой из VM.

2. На VM1 persona:
 - добавьте репозиторий Persona;
 - установите Persona Server 5.7;
 - проверьте статус сервиса mysql;
 - подключитесь к Persona серверу пользователем root с заданным паролем.
3. Используя открытое подключение:
 - создайте пользователя БД wordpress@<wordpress> с паролем wordpress;
 - создайте БД wordpress;
 - задайте пользователю wordpress полные права на БД wordpress.

[...]

Тема 13: Очереди. Отказоустойчивый RabbitMQ. Shovel

Практическое задание

1. Создайте виртуальные машины VM1 rabbitmq-source и VM2 rabbit-destination следующей конфигурации:
 - 1 CPU

- 1G memory

Обновите систему на VM1,2.

Добавьте адреса VM1,2 в файл /etc/hosts каждой из VM.

2. На VM1,2 установите пакет rabbitmq-server. Убедитесь, что сервис rabbitmq-server запущен и работает.
3. На VM1,2 включите RabbitMQ-плагин shovel.
4. На VM1,2 добавьте пользователя shovel-user с паролем shovel-password. Задайте пользователю shovel-user полные права на корневой vhost.
5. На VM1,2 создайте скрипты send.py и receive.py. Пошлите и примите тестовые сообщения.
6. На VM1 rabbit-source создайте shovel-соединение. Для этого установите параметр shovel-test для компонента shovel:
 - src-uri равно amqp://shovel-user:shovel-password@rabbit-source;
 - src-queue равно hello;
 - dest-uri равно amqp://shovel-user:shovel-password@rabbit-destination;
 - dest-queue равно hello.
7. На VM1 проверьте статус shovel-соединения.

[...]

Модуль 5. Средства обеспечения высокой производительности и отказоустойчивости

Тема 6: NGINX балансировка нагрузки

Практическое задание

1. Создайте VM:
 - nginx-balancer
 - nginx-node1
 - nginx-node2
 - nginx-node3

следующей конфигурации:

- 1 CPU
- 1G memory

Обновите систему на созданных VM.

2. На все VM установите пакет nginx.
3. Модифицируйте файл /var/www/html/index.nginx-debian.html на VM nginx-node1,2,3 так, чтобы в заголовке отображался номер ноды.
4. Сконфигурируйте NGINX на VM nginx-balancer для работы в режиме load balancer, добавив в backend VM nginx-node1,2,3. Протестируйте и примените новую конфигурацию.
5. Выполните несколько HTTP-запросов к VM nginx-balancer. Убедитесь, что нагрузка распределяется между всеми VM nginx-node1,2,3.
6. Выключите VM nginx-node1. Снова выполните несколько HTTP-запросов к VM nginx-balancer. Убедитесь, что:
 - NGINX balancer справляется с ситуацией недоступности одной ноды в backend
 - Запросы, выпадающие на отсутствующую ноду занимают больше времени
7. Увеличьте время, в течение которого load balancer считает ноду недоступной и не маршрутизирует на нее запросы до 1 минуты.

[...]

Тема 17: Hadoop - YARN

Практическое задание

Мы будем использовать три абсолютно такие же ноды как создавали в задании 1:

3 виртуальных машины (ноды) u18-hadoop{1,2,3} следующей конфигурации:

- Операционная система: Ubuntu 18.04
- Количество ядер: 2 CPU
- Оперативная память: 2 GB
- Виртуальный диск: 20 ГБ

Полностью распределенный Hadoop кластер должен быть запущен вместе со всеми компонентами с помощью команды `$HADOOP_HOME/sbin/start-all.sh`.

1. Проверь работоспособность YARN кластера через WEB интерфейс.
2. Просмотри на странице, полученной в первом задании о отметь для себя информацию о состоянии кластера, нодах и приложениях, ошибках, сервисах и т.д.
3. Запусти приложение pi с помощью yarn - команды.
4. Проверить ход выполнения и результат третьего задания через WEB интерфейс Менеджера Ресурсов.
5. Просмотри список приложений.
6. Запусти NodeManager.
7. Получи состояние всех ресурсов, используя клиент Менеджера Ресурсов.

Итоговое оценивание

В конце программы обучающиеся сдают итоговую аттестацию.

Финальный презентационный проект

Правила презентации проекта:

Доброго дня! Рады, что вы успешно завершаете практикум Highload. Рассказываем, каким образом мы будем проверять, как вы усвоили материал:

1. Вы выбираете удобное вам время для итоговой встречи, укажите в ответ на это задание три удобных вам таймслота длительностью 2 часа (собеседование занимает около часа времени) и удовлетворяющие таким условиям:
 - не менее чем через 3 дня (минимальный срок для подготовки проекта)
 - не более 10 дней от текущей даты
 - находится в таймслоте с 9:00 до 19:00 по МСК с Понедельника по Пятницу

Мы сверим указанные вами слоты с нашим расписанием собеседований, подтвердим один из них и пришлем вам задание.

2. Вы выполняете задание, а также готовите мини-презентацию о своем решении - почему сделали именно так, какие еще варианты решений рассматривали и т.д. (мы не требуем обязательного наличия слайдов, просто подготовьте небольшой доклад)
3. При подтверждении даты презентации, мы вместе с заданием отправляем вам ссылку на Zoom митинг, на запланированную дату.

ВАЖНО: На собеседовании Вам нужно будет предъявить паспорт, чтобы подтвердить свою личность (ФИО должно совпадать с данными в личном кабинете). Это нужно для выдачи именного сертификата.

Этапы презентации и правила оценивания:

1. Идентификация (вы показываете свой паспорт или другое удостоверение личности)
1 этап. Запуск, презентация и защита вашего проекта. Оценивается до 20 баллов. Во время презентации вам необходимо:

- Рассказать какие решения вы принимали во время реализации проекта
- С какими сложностями сталкивались и как их решили
- Ответить на вопросы куратора

2 этап - 3 вопроса по практикуму Highload. За каждый вы получите от 0 до 10 баллов.

- 0 баллов - если ответили неверно.
- 4 балла - если направление мысли (поиска ответа) определено с помощью подсказки кураторов или дополнительных источников (google / etc), но вы не сказали правильный ответ.
- 7 балла - если направление мысли (поиска ответа) определено без помощи подсказки кураторов или дополнительных источников, но вы не сказали правильный ответ.
- 9 баллов - если направление мысли (поиска ответа) определено без помощи дополнительных источников, но правильный ответ найден с помощью дополнительных источников (google / etc) или подсказки кураторов.
- 10 баллов - если полностью самостоятельно раскрыли тему.

Все баллы суммируются. Максимальное количество баллов за встречу - 50.

Сертификат будет доступен в вашем профиле, в течении некоторого времени после защиты проекта (при условии успешной защиты).

Если вы не проходите собеседование, или не удовлетворены набранными баллами, то следующая попытка будет вам предоставлена не ранее, чем через 1 календарный месяц.

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.