

Микросервисы на Java:

Kafka, Kubernetes, мониторинг

Пояснения к словам иностранного происхождения

Здесь собраны пояснения к словам иностранного происхождения, которые используются в программе курса, но пока не включены в словарную норму русского языка.

01	Кому подойдёт курс	• Middle Java-разработчики с опытом 2+ лет, готовые к росту на Senior • Senior Java-разработчики • Разработчики, умеющие строить отдельные сервисы, но не понимающие как их связывать, деплоить и мониторить • Тимлиды и архитекторы, систематизирующие практический опыт с распределёнными системами • Разработчики, которым на собеседованиях задают вопросы про микросервисы и Kubernetes, но не имеющие практического опыта
02	Чему научитесь на курсе Какие знания и навыки освоите	• Проектировать микросервисную архитектуру (Saga, Circuit Breaker, API Gateway, Service Discovery) • Применять паттерны межсервисного взаимодействия на практике • Работать со Spring Cloud (Consul) • Настраивать асинхронный обмен данными через Apache Kafka и Spring Kafka • Строить систему распределённого трейсинга через Spring Cloud Sleuth и Zipkin • Настраивать мониторинг и алерты с помощью Prometheus и Grafana • Строить log pipeline через Kafka → Logstash → Elasticsearch → Kibana • Разворачивать приложения в Kubernetes через Helm • Создавать декларативные CI/CD-пайплайны на Jenkins • Аргументировать архитектурные решения перед командой и бизнесом

03

Как проходит обучение

- Вебинары и сессия ответов на вопросы
- Необходимая теория
- Практика
- Проекты — для отработки навыков

Что вас ждёт на курсе

Удостоверение о повышении квалификации или сертификат	Практика, основанная на решении реальных рабочих задач	Обучение от экспертов из Яндекса и других крупных компаний
---	--	--

Микросервисы на Java: Kafka, Kubernetes, мониторинг

3 месяца

продолжительность курса

6 проектов

в портфолио (опционально)

5 ЧАСОВ



Аргументация в рабочей коммуникации

*дополнительный модуль

10 ЧАСОВ



Контейнеризация и управление многоконтейнерными Java-приложениями при помощи Docker

*дополнительный модуль

3 НЕДЕЛИ

01

Разработка приложений с использованием микросервисной архитектуры

2 НЕДЕЛИ

02

Apache Kafka

2 НЕДЕЛИ

03

Мониторинги и логи с помощью ELK-стека

3 НЕДЕЛИ

04

Развёртывание приложения с использованием CI/CD, Kubernetes, Helm

Аргументация в рабочей коммуникации

★

5 часов

Дополнительный модуль

Содержание

- Темы
- Аргументация;
 - Переговоры;
 - Контраргумент;
 - Делиберативный диалог;
 - Убеждение

Контейнеризация и управление многоконтейнерными Java-приложениями при помощи Docker

★

10 часов

Дополнительный модуль

Содержание

- Темы
- Dockerfile
 - Docker compose
 - Docker контейнер
 - Docker

Разработка приложений с использованием микросервисной архитектуры

01

3 недели

Содержание

Темы	• Паттерны микросервисной архитектуры; • Паттерны межсервисного взаимодействия; • Паттерны обеспечения безопасности, состояния, тестирования и UI сервисов; • стек технологий Spring Cloud; • Безопасность и тестирование в Spring Cloud; • Интеграции в Spring Cloud
Практика	Создание микросервисного приложения — инфраструктуру банка с использованием микросервисной архитектуры

Apache Kafka

02

2 недели

Содержание

Темы	• Стриминговая платформа Apache Kafka • Возможности Apache Kafka • Интеграция Apache Kafka со Spring • Возможности Spring Kafka • Неблокирующие ретрай обработки сообщений в Spring Kafka • Поддержка потоковой обработки данных и тестирование в Spring Kafka
Практика	Продолжите работу над микросервисным приложением банка — настройте взаимодействие микросервисов при помощи Apache Kafka

2 недели

Содержание

Темы	• Трейсинг с помощью Spring Cloud Sleuth и Zipkin; • Мониторинги и алерты с помощью Prometheus и Grafana; • Сбор логов с использованием Apache Kafka и Logstash; • Использование Elasticsearch для индексации, хранения и поиска логов; • Использование системы визуализации данных Kibana
Практика	Завершите работу над микросервисным приложением банка — добавьте логирование и мониторинг при помощи ELK-стека
Воркшоп	Воркшоп по мониторингам и логированию: какими мониторингами стоит обвесить систему; какие мониторинги поместить на фронт и бэк

3 недели

Содержание

Темы	• Оркестрация контейнеризованных приложений • Внутреннее устройство Kubernetes • Возможности Kubernetes • Пакетный менеджер Helm • Тестирование, обеспечение безопасности и лучшие практики в Helm • Непрерывная интеграция и доставка микросервисов • Возможности Jenkins • Интеграции Jenkins
Практика	Продолжите работу над микросервисным приложением банка — настройте его развёртывание при помощи CI/CD, Kubernetes, Helm
Воркшоп	Воркшоп по Kubernetes и CI/CD: отличия от docker-compose, swarm; когда использовать Kubernetes, а когда можно обойтись без него; как раскатывать микросервисное приложение, обеспечивая обратную совместимость