

Микросервисная архитектура

Код: ARC-015

Длительность: 24 ч.

Описание:

Микросервисная архитектура (МСА) сравнительно недавно завоевала славу самого популярного архитектурного подхода. Ее использование обеспечивает множество преимуществ на различных этапах жизненного цикла продукта. В частности, МСА позволяет получить меньшее время реакции на требования бизнеса, что зачастую обеспечивает ее превосходство над прочими архитектурными стилями.

Однако, в процессе разработки МСА систем архитектор сталкивается со множеством вызовов, многие из которых заставляют новичков откатываться назад.

Ошибки декомпозиции могут привести к серьёзному увеличению сложности разработки. Производительность страдает из-за непреодолимых сетевых задержек. Распределенное хранение данных не позволяет использовать транзакционную согласованность. Большое количество взаимодействующих по сети компонент снижает надежность. Сложно обеспечивать безопасность. Кроме того, отдельного разговора требует тестирование и обслуживание распределенной системы.

Данный курс покажет преимущества и недостатки использования МСА. Поможет принимать обоснованное решение о применимости МСА в зависимости от потребностей бизнеса и возможностей команды.

В курсе рассматриваются основные проблемы, связанные с реализацией микросервисной архитектуры. Стратегии, механизмы, паттерны, решающие озвученные выше задачи. По каждой рассматриваемой проблеме будут представлены наиболее распространённые инструменты (такие, как Kubernetes, Istio, Histryx, Kibana, и многие другие).

Практические упражнения позволят приобрести навык самостоятельного проектирования МСА систем.

Цели:

- Дать представление о МСА и о её связи с остальными архитектурами системы;
- Дать представление о проектировании МСА систем;
- Выработать практический навык проектирования МСА систем;

- Познакомить слушателей с наиболее распространёнными шаблонами MSA;

Разбираемые темы:

1. Архитектурные стили (2 часа).

- Понятия монолита, SOA, MSA.
- Сравнение и обоснование выбора.

2. Паттерны декомпозиции MSA (2 часа).

- Оптимальный размер микросервиса;
- Разбиение на сервисы по бизнес-возможностям;
- Разбиение на сервисы по проблемным областям;
- Практики определения предметных областей;
- Декомпозиция по техническим и организационным аспектам.

3. Организация разработки под MSA (1 час).

- Организация команд под MSA;
- Организация репозитория кода под MSA.

4. Интеграция микросервисов (3 часа).

- Паттерны взаимодействия;
- Синхронные и асинхронные механизмы взаимодействия;
- Оркестровка и хореография;
- Основные протоколы и технологии (REST, gRPC, GraphQL, Kafka, RabbitMQ и др.);
- Взаимодействие с внешними системами (API Gateway, BFF);
- Реактивные системы;
- Архитектуры управляемые событиями;
- Разделение команды и запроса.

5. Организация работы с данными в MSA (3 часа).

- Паттерны работы с данными;
- Порождение событий;
- Справочные данные в MSA;
- Единый источник истины (Source of Truth) в MSA.

6. Основные шаблоны MSA (9 часов).

1. Модифицируемость.
 - Решение проблемы высокой связанности. Версионирование интерфейсов и событий. Типы контрактов. Изменения событийной модели. Паттерны организации запросов. Микросервисное шасси

- (Microservice chassis). Сетки сервисов (Service Mesh).
2. Масштабируемость.
 - Паттерны построения балансировщиков. Сервисы обнаружения и паттерны работы с ними.
 3. Производительность.
 - Паттерны производительности в MSA (переборка, изящная деградация и пр.).
 4. Согласованность.
 - Проблемы обеспечения согласованности в MSA. CAP теорема. Решения проблем согласованности. Двухфазные коммиты. Паттерн SAGA. Понижение уровня согласованности.
 5. Надежность.
 - Механизмы отказоустойчивости. Паттерны предохранитель (Circuit Breaker), дросселировка, зависимые таймауты и пр.
 6. Безопасность.
 - Механизмы обеспечения безопасности в MSA. Аутентификация. Авторизация. Защита периметра. Защита каналов связи. Основные протоколы и паттерны (OAuth2, JWT, привратник, ключ камердинера и пр.).
 7. Тестируемость.
 - Пирамида и квадрат тестирования. Особенность тестирования микросервисов.
 - Модульное тестирование. Интеграционное тестирование. Компонентное тестирование. Тестирование E2E. Паттерны тестирования MSA.
 8. Удобство обслуживания.
 - Возможность наблюдения (Observability). Паттерны наблюдения (распределенная трассировка, агрегация журналов). Мониторинг и журналирование. Конфигурирование микросервисов. Вынесение конфигурации во вне. Поддержка системы.

7. Развертывание микросервисов (2 часа).

- Паттерн «Deployment pipeline»;
- Паттерны развертывания;
- Использование технологий Docker и Kubernetes;
- Использование сетки Istio;
- Бессерверное развертывание;
- Стратегии развертывания (синие-зеленые, канареечные и т.д.).

8. Стратегии миграции от монолита к MSA (2 часа).

- Стратегии перехода с монолита на микросервисы;
- Удушение монолита (Strangler monolith);
- Паттерны удушения;
- Связь микросервиса с монолитом;
- Миграция базы данных.

Целевая аудитория:

- Архитекторы;
- DevOps специалисты;
- Ведущие разработчики;
- Руководители разработки.

Предварительная подготовка - общее:

Опыт разработки программных систем.

Желательно опыт разработки архитектуры или прохождение курса [ARC-001 «Основные практики архитектора ПО»](#), знание подхода DDD или прохождение курса [ARC-003 «Domain Driven Design»](#).