

ชื่อคอร์ส : Microservices Design

ชื่อวิทยากร : พิเศษ เสรี มากเอียด (Seri Makiad)

ตำแหน่ง : Business Unit, Associate Director

วัน เวลา : 5 มีนาคม 2568 / 13:00 - 16:00 น.

สถานที่ : ICT Salaya (Onsite)

สิ่งที่คุณจะได้เรียนรู้จากคอร์สนี้ : เรียนรู้หลักการในการออกแบบสถาปัตยกรรมซอฟต์แวร์แบบ Microservices และสามารถนำไปใช้ต่อยอดในการทำงานได้

เนื้อหาในคอร์ส :

Module 1: Introduction to Microservices Architecture

- Lesson 1: Overview of Microservices
 - o What are microservices? Key concepts and principles
 - o Comparing monolithic vs. microservices architectures
 - o Benefits and challenges of microservices
- Lesson 2: Key Components of Microservices Architecture
 - o Service discovery, API gateways, and load balancing
 - o Inter-service communication (synchronous vs. asynchronous)
 - o Understanding data management in microservices (shared vs. distributed databases)

Module 2: Designing Microservices

- Lesson 1: Domain-Driven Design (DDD)
 - o Introduction to DDD and its relevance in microservices
 - o Identifying bounded contexts and designing services around them
 - o Aggregates, entities, and value objects
- Lesson 2: Service Decomposition
 - o Techniques for decomposing a monolithic application into microservices
 - o Designing microservices around business capabilities
 - o Best practices for defining service boundaries
- Lesson 3: API Design and Contracts
 - o Designing RESTful APIs for microservices
 - o Implementing API contracts and versioning
 - o Handling API evolution and backward compatibility

Module 3: Communication Between Microservices

- Lesson 1: Synchronous Communication
 - o Using HTTP/REST and gRPC for service communication
 - o Implementing client-side and server-side load balancing
 - o Handling latency and network issues

- **Lesson 2: Asynchronous Communication**
 - Message brokers and event-driven architectures
 - Implementing message queues (e.g., RabbitMQ, Kafka)
 - Patterns for event sourcing and CQRS (Command Query Responsibility Segregation)

Module 4: Data Management in Microservices

- **Lesson 1: Data Consistency and Transactions**
 - Managing distributed data and eventual consistency
 - Implementing distributed transactions (e.g., Saga pattern)
 - Handling data duplication and synchronization
- **Lesson 2: Database Design for Microservices**
 - Designing databases per service (polyglot persistence)
 - Patterns for data replication and sharding
 - Best practices for schema management and migrations

Module 5: Security and Resilience in Microservices

- **Lesson 1: Securing Microservices**
 - Implementing authentication and authorization (OAuth2, JWT)
 - Securing inter-service communication (SSL/TLS, mutual TLS)
 - Best practices for securing APIs and data
- **Lesson 2: Building Resilient Microservices**
 - Implementing resilience patterns (e.g., Circuit Breaker, Bulkhead)
 - Monitoring and logging in a microservices environment
 - Designing for fault tolerance and graceful degradation

Module 6: Deploying and Scaling Microservices

- **Lesson 1: Containerization and Orchestration**
 - Containerizing microservices with Docker
 - Orchestrating microservices with Kubernetes
 - Service mesh and advanced orchestration with Istio or Linkerd
- **Lesson 2: Continuous Integration and Continuous Deployment (CI/CD)**
 - Setting up CI/CD pipelines for microservices
 - Blue-green deployments, canary releases, and rolling updates
 - Automated testing and monitoring in CI/CD
- **Lesson 3: Scaling Microservices**
 - Horizontal scaling and load balancing strategies
 - Auto-scaling and capacity planning
 - Best practices for optimizing performance and resource usage

Module 7: Monitoring, Logging, and Observability

- **Lesson 1: Monitoring and Metrics**
 - Setting up monitoring tools (Prometheus, Grafana)
 - Collecting and analyzing performance metrics
 - Setting up alerts and dashboards
- **Lesson 2: Distributed Tracing and Logging**
 - Implementing distributed tracing (Jaeger, Zipkin)
 - Centralized logging solutions (ELK Stack, Fluentd)
 - Correlating logs and traces for debugging and analysis
- **Lesson 3: Observability in Microservices**
 - Building observability into microservices
 - Best practices for ensuring system reliability
 - Case studies and real-world examples
 -

ทำไมต้องเรียนคอร์สนี้ : ผู้เรียนจะได้รับความรู้ความเข้าใจเกี่ยวกับสถาปัตยกรรมซอฟต์แวร์แบบ Microservice ตั้งแต่พื้นฐานการออกแบบ และสามารถเลือกใช้รูปแบบสถาปัตยกรรมซอฟต์แวร์ของระบบที่เหมาะสมกับลักษณะของงานได้อย่างเหมาะสม

เรียนจบคอร์สนี้แล้วไปต่อยอดอะไรได้บ้าง : เข้าใจความแตกต่างระหว่างสถาปัตยกรรมซอฟต์แวร์แบบต่าง ๆ และเรียนรู้หลักการในการออกแบบสถาปัตยกรรมซอฟต์แวร์แบบ Microservices และสามารถนำไปใช้ต่อยอดในการทำงานได้

หลักสูตรนี้เหมาะสำหรับใคร : Backend Developer หรือผู้ที่ต้องการพัฒนาระบบงานแบบ Microservices สำหรับการพัฒนาระบบ

ประวัติผู้สอน : เสรี มากเอียด , Technology Director จาก Bluebik Vulcan

