Title: A Workshop-based training on MSA and Event Storming (DDD)

## 교육 개요, 목표, 특징

* 개요: MSA 아키텍쳐의 분석,설계,구현에 있어서 각광 받는 방법으로 이벤트 스토밍 (Event-Storming) 기법을 기반한 실질적 설계, 구현, 운영 기법에 대한 워크숍 방식의 교육
* 목표: 커머스 예제를 중심으로 Event 중심의 MSA 아키텍처에 대한 분석,설계기법을 이해하고 스프링클라우드, 쿠버네티스를 기반한 구현과 운영 (DevOps)기법을 이해함
* 특징: 기존 객체지향 및 DDD 등 복잡하고 무거운 학습의 과정을 최소화하고 마이크로서비스의 구현결과에 필요한 실질적 기법만을 최적화한 기법으로 이벤트스토밍과 스프링클라우드, 쿠버네티스 CI/CD 의 내용을 3일내에 실습예제를 기반하여 워크숍 방식으로 컴팩트하게 다룬다.

## 교육대상자가 사전에 학습하거나 준비해올 사항

* 장비: 개발 노트북
* 기초지식: 자바언어기본
* 개발환경: 구글 클라우드 플랫폼 (GCP) 계정을 실습자 개개인이 갖고 있어야 합니다. 사전에 공지를 통해서 가입 ([cloud.google.com](http://cloud.google.com/)) 을 유도해주시면 좋습니다. 기타 개발도구는 Java 와 IntelliJ 가 있으면 좋습니다.

## 

## Course Learning Objects

|  |  |  |
| --- | --- | --- |
| LO | 세부주제 | 상세 설명 |
| Why MSA and agile | * Success stories on MSA: Amazon and Netflix * Agile Methodology * Agile and MSA * MSA and EDA architectural style * DevOps Process * Values and effects * Future: Serverless and BizDevOps | Micro Service Architecture 의 출현 배경, 성공사례, 아키텍처적인 접근, 비즈니스 가치에 대한 이해 |
| Introduction to MSA Analysis and Design including Event Storming | * Domain-Driven Design * Service Decomposition and API Design Strategy * Introduction to Event Storming: Easy way to DDD and Event System * The Environment and Process of Event Storming | MSA 기반의 대형 시스템 분석 기법으로 MDD 의 일종인 DDD 를 적용한 분석/설계 방법 이해 |
| Workshop: Event Storming with a specific domain | * Explanation on the target Domain: A Commerce Example (Online Shopping mall) * Target User Scenario * Team Separation and KPI statements * Ice breaking with first step to event storming with instructor guide: What is Events, Policies, Commands and Aggregates * The first event storming by team’s autonomy: guided by facilitators (1~2 hours) * First review for the first results: Best teams will be complimented. * Correction and refinement (1 hour) * Finding Bounded Contexts and Microservices * Finding key events needed to be mutually understood and shared team-widely. And standardize the naming and statements of key events. * Finding read models for data projection and CQRS | 워크숍을 통한 이벤트 스토밍 첫번째 경험하기 |
| Workshop: Event driven integration | * How to use Kafka publisher and consumer with the command line tools * Publish event by a team, consumes the event by other teams. * Business Process Execution by choreography and Eventual transactions * Saga and compensation | 워크숍을 통한 이벤트 기반의 통합이 가능함을 직접 경험 |
| Micro Service Implementation Patterns | * Transforming Event Storming to Implementations with Hexagonal Architecture * Introduction to tools: Spring Boot, Kafka, JPA, Spring Data Rest, Spring Cloud Stream * Example implementation of a domain: * How to generate a spring cloud project with the Spring Initializer and IntelliJ * How to map the aggregate stickers to “Entity” classes * How to map the command stickers to “RestRespository” and service classes * How to capture the moment of event publishing: Aggregate’s CRUD moment by using the JPA lifecycle hooks. * Inserting domain event publish code in the aggregate with Spring Cloud Streams * Listening domain event by the “EventListener” | 단위 마이크로 서비스의 세부 구현 방법 설명 |
| Workshop: Unit Microservices Integration | * Each team implements the microservices in parallel and can commit the source code and the code changes triggers the automated CI/CD * Replays entire business process that was conducted manually in the previous workshop and find that it can be conducted by fully-automated manner. | 구현된 마이크로 서비스를 통한 실제 상호연동 |
| Proposed Microservice Integration Patterns | * Integrating mi-services with PubSub (Choreography with MQ/Kafka) * Integrating mi-services with Mediator (Orchestration with BPM) * Transaction Issues (Dynamic Discovery, API gateway, Event sourcing, CQRS) * Data Projection: HATEOAS (UI), GraphQL, Cache, Materialized View (CQRS) * Security with OAuth and IAM | 마이크로 서비스들을 통합하고 트랜잭션 처리하는 방법 |
| Making User Interfaces and direct call | * Integrating mi-services with User Interfaces: Client-side rendering, SPA, MVVM, CORS exceptions and API Gateway * Integrating micro services with direct-call (Request-Response, Synchronous)   (Dynamic Service Binding, Service Registry and Circuit Breaker) and Understanding why it is not recommended | 유저인터페이스 만들기, 직접적 호출 방법 이해하기와 |
| Legacy Migration Patterns | * Legacy Migration Strategies: Value-Focused Migration with Strangler Pattern and Anti-patterns * Legacy Migration Tools: Service Mesh and Event Sourcing | 점진적 레가시 통합에 적합한 방법 제시 |
| Workshop: Delivering Microservices and DevOps | * Container-based Application Patterns * Containerizing Services and Container orchestration with Docker * DevOps tool chains: CI/CD with Multi-phase Dockerfile, Kubernetes and KNative-build/serving * Zero down-time deploy (canary) with Istio and KNative-serving * Service Reliability Management with Circuit Breaking, Rate Limiting and Chaos Engineering (Istio) | 마이크로 서비스 기반의 무정지 반영, 자동복구, 운영자동화 |

## Architect Course (3 Days)

Target audiences:

* Business Analyst
* Software Architect
* Developers

|  |  |  |
| --- | --- | --- |
| Day | LO | 제약사항 |
| Day 1 | * Why MSA and agile (1 hour) * Introduction to MSA Analysis and Design including Event Storming (1 hour) * Workshop: Event Storming with a specific domain (4 hour) * Workshop: Event driven integration (2 hour) |  |
| Day 2 | * Micro Service Implementation Patterns (2 hour) * Workshop: Unit Microservices Integration (4 hour) * Proposed Microservice Integration Patterns (2 hour) |  |
| Day 3 | * Making User Interfaces and direct call (4 hour) * Legacy Migration Patterns (1 hour) * Workshop: Delivering Microservices and DevOps (2 hour) * [Additional] Organizational Change Management: Horizontal to Vertical and Metrix Organization models for changing to Agile, LeSS, SaFe, and Strangler pattern in organization | 푸른색: 추가 activity |

## 

## Developer Course (3 Days)

Target audiences:

* Server and UI Developers
* Software Architect

|  |  |  |
| --- | --- | --- |
| Day | LO | 제약사항 |
| Day 1 | * Why MSA and agile (1 hour) * Introduction to MSA Analysis and Design including Event Storming (1 hour) * Workshop: Event Storming with a specific domain (1 hour) * Workshop: Event driven integration (1 hour) * Micro Service Implementation Patterns (4 hour) * Lab-time: Coding with Spring-boot | 푸른색: 추가 activity |
| Day 2 | * Workshop: Unit Microservices Integration (4 hour) * Lab-time: Coding with Spring-boot * Proposed Microservice Integration Patterns (2 hour) * Making User Interfaces and direct call (2 hour - part1) * Lab-time: Coding Vue-JS |  |
| Day 3 | * Making User Interfaces and direct call (2 hour - part2) * Lab-time: Coding Vue-JS * Legacy Migration Patterns (optional) * Workshop: Delivering Microservices and DevOps (6 hour) * Lab-time: Docker, Kubernetes and Istio, KNative with GCP |  |

## 

## Operator Course (3 Days)

Target audiences:

* Operators
* Developers

|  |  |  |
| --- | --- | --- |
| Day | LO | 제약사항 |
| Day 1 | * Why MSA and agile (1 hour) * Introduction to MSA Analysis and Design including Event Storming (1 hour) * Micro Service Implementation Patterns (2 hour) * Lab-time: Coding with Spring-boot (optional) * Proposed Microservice Integration Patterns (2 hour) | 마이크로 서비스 아키텍처의 큰 그림상의 이슈를 중심으로 실제 구현은 최소화, 이미 완성된 예제를 기반으로 실습진행 |
| Day 2 | * Legacy Migration Patterns (1 hour) * Workshop: Delivering Microservices and DevOps (7 hour)   + Container-based Application Patterns   + Containerizing Services and Container orchestration with Docker   + DevOps tool chains: CI/CD with Multi-phase Dockerfile, Kubernetes and KNative-build/serving   + Lab-time: Registering GCP account and hello to Kubernetes (Basic K8S commands)   + Lab-time: Containerizing application with docker   + Lab-time: Pushing app to GCR and run with docker   + Lab-time: Launching app by Kubernetes, Self-healing, Auto-scaling   + Kubernetes Object Models and Declarative desired-state description   + Using Helm and creating Helm charts | 푸른색: 추가 activity |
| Day 3 | * Workshop: Delivering Microservices and DevOps (8 hour)   + Zero down-time deploy (canary) with Istio and KNative-serving   + Lab-time   + Service Reliability Management with Circuit Breaking, Rate Limiting and Chaos Engineering (Istio)   + Lab-time | 푸른색: 추가 activity |

## 

## 

## Quick Overview Course (1 Day)

MSA 와 이벤트스토밍에 전반에 대한 빠른 이해를 위함.

세부 기술등의 용어와 개념 정도 이해를 가지고 향후 본 과정들을 듣기 위한 Overview 가 목적

Target audiences:

* All

|  |  |  |
| --- | --- | --- |
| Day | LO | 제약사항 |
| Day 1 | * Why MSA and agile * Introduction to MSA Analysis and Design including Event Storming (1 hour) * Workshop: Event Storming with a specific domain (1 hour) * Workshop: Event driven integration (1 hour) * Micro Service Implementation Patterns (2 hour) * Workshop: Unit Microservices Integration (1 hour) * Proposed Microservice Integration Patterns (1 hour) * Making User Interfaces and direct call (1 hour) * Legacy Migration Patterns (1 hour) * Workshop: Delivering Microservices and DevOps | Hour 표시없음: Optional |