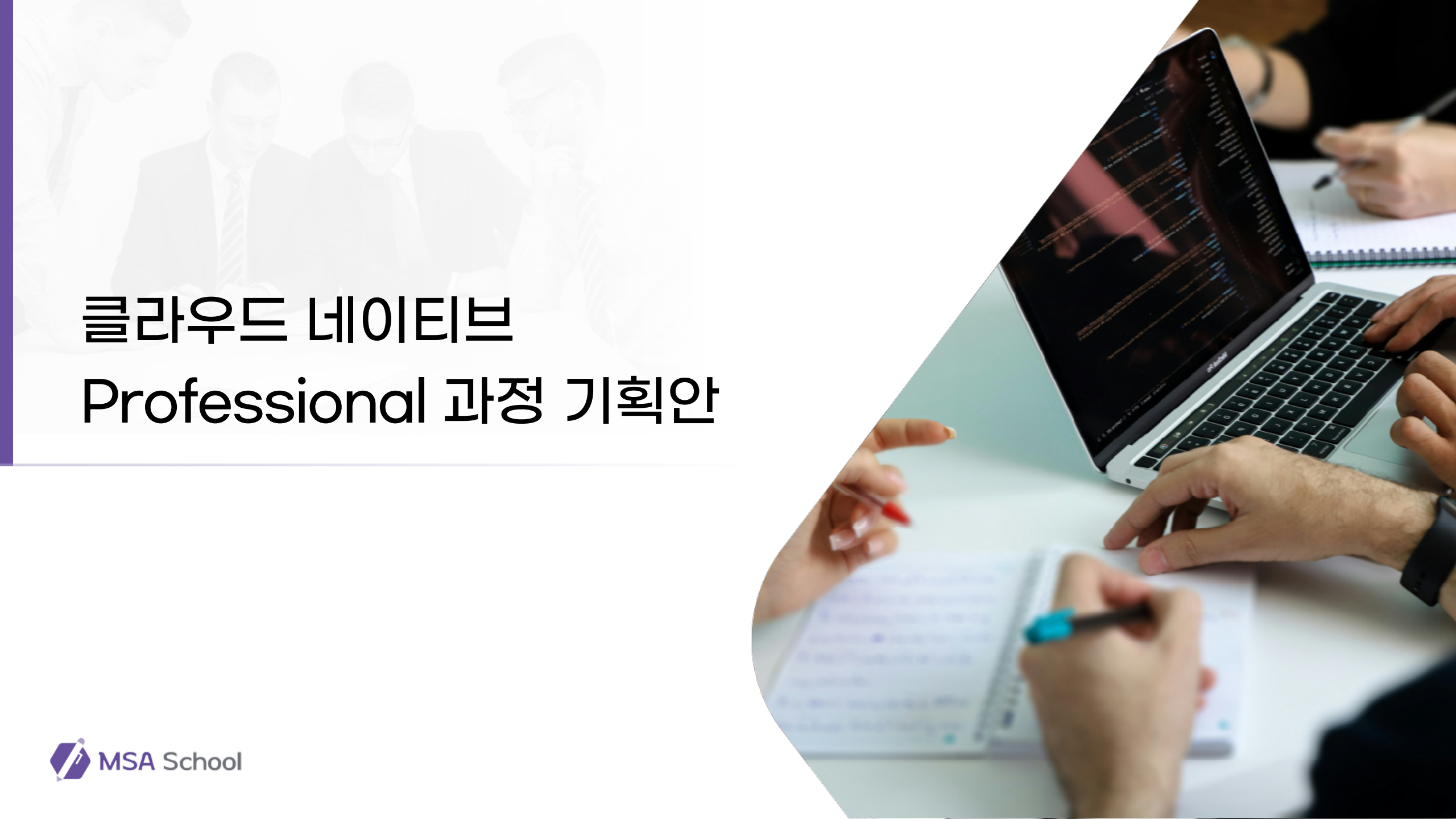




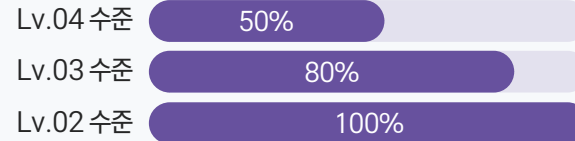
클라우드 네이티브 Professional 과정 기획안

인재육성 성숙도모형

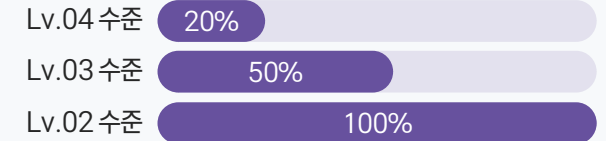
- ✓ 성숙도 모형은 CNCF, Scaled Agile Framework 등을 통합하여 자체 개발 클라우드 네이티브 전문가에게 필요한 지식 성숙도 모형입니다.
- ✓ 각 수준별 목표에 필요한 LO가 연결되어 훈련이 진행되고 각 단계를 통과하기 위한 체크포인트들을 기준으로 자격 인증이 이루어집니다.
- ✓ 개개인별 성취도에 따른 성숙도 모형을 과정중 확인할 수 있으며, 개개인의 역량개발에 대한 로드맵으로 작용합니다.

인재 배출 목표

아키텍트



개발자

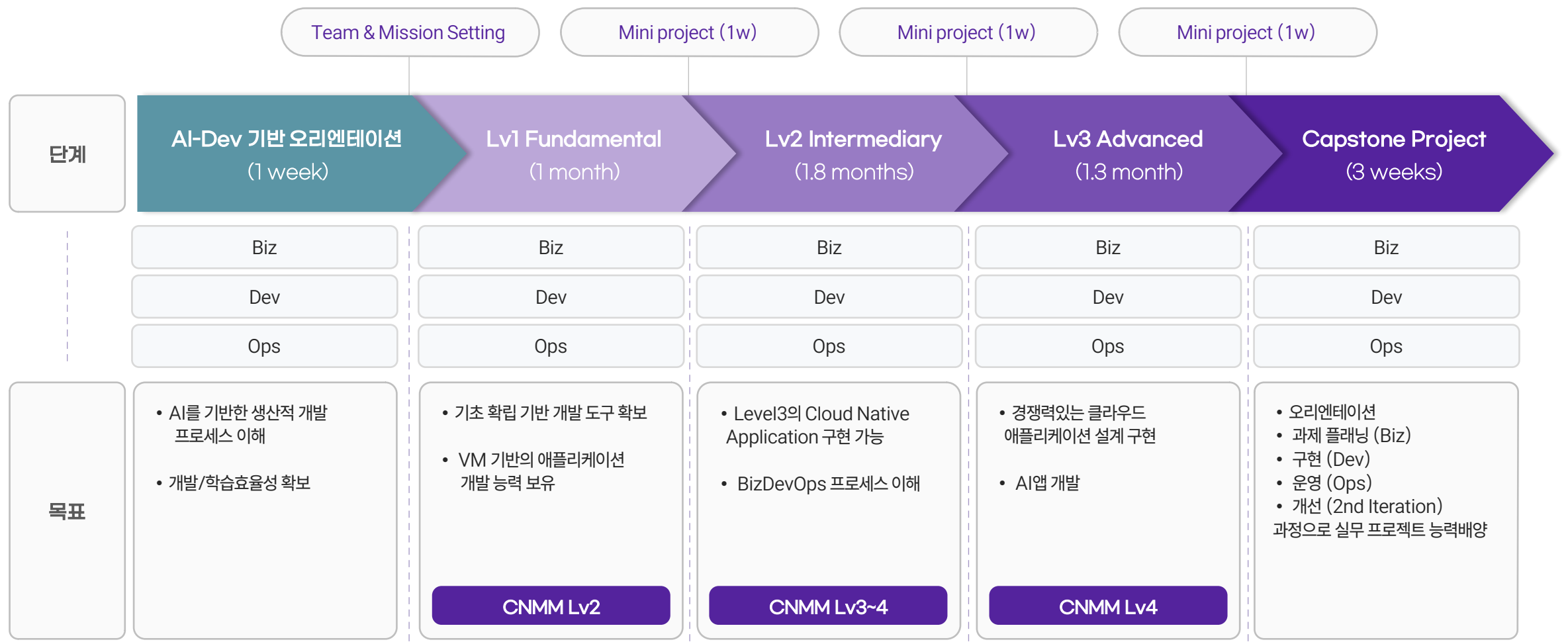


구현 관점	Early (Lv1)	Inception (Lv2)	Expanding (Lv3)	Mature (Lv4)
기능 분해	기능과 유즈케이스 단위로 분리 • 비즈니스 역량은 도출한 비즈니스 기능과 유즈케이스 단위로 분리가능	서비스별 인터페이스 정의 • 추출된 각 유즈케이스와 인터페이스를 통해 액세스 할 데이터에 대해 잘 정의된 인터페이스를 가짐	도메인 Context 분리 • Ubiquitous language가 다른 bounded context 간의 커뮤니케이션시, Anti-corruption layer를 통해 수행	도메인 기반 이벤트 • 구체화된 보기, 읽기 쓰기를 위한 별도의 모델 (CQRS) 구축
데이터	2PC 사용 가능 • ACID 기반의 트랜잭션을 유지합니다. Canonical Data Model 를 지향	Scheme per Service • 각각의 서비스는 자신만의 Scheme를 가짐 • 서비스들과 다중 엔터프라이즈 데이터 저장소 간의 트랜잭션이 적은 조정으로 이루어짐	DBMS per Service • 완전히 분산된 데이터 관리 • 서비스별 다른 유형의 DBMS 사용이 가능한 폴리글랏 퍼시스턴스를 지향	Event-Driven Archi. • 이벤트 기반 데이터 관리, 이벤트 소싱 및 커맨드 쿼리 • 일시적으로 데이터에 일관성이 없는 상태가 존재할 수 있으나, 일정 시간이 지나면, 데이터가 도착하여 다시 Consistency를 충족
SW 아키텍처	UI/UX : Server Side rendering, Session based 보안, 단일 언어(eg. Spring framework)	UI/UX : Server Side rendering, Session cluster 적용, MSA 지향 언어(eg. Spring Boot)	UI/UX : Client Side rendering, Token based 보안, OAuth2, 폴리글랏 Language	UI/UX : Client Side rendering+MVVM, Token based 보안, OAuth2, 폴리글랏 Language + Service Mesh
Infra 스트럭처	지속적인 빌드, 지속적인 통합 운용	지속적인 딜리버리와 배포, 로그의 중앙 집중화	컨테이너 사용(도커), 컨테이너 오케스트레이터(k8s), 외부 구성(유레카, 주키퍼)	자동 프로비저닝을 갖춘 PaaS기반 솔루션
배포	설치 스크립트 구동, 호스트 당 멀티 서비스 인스턴트	VM 당 하나의 서비스 인스턴스 클라이언트 사이드 로드 밸런싱 서버사이드 로드 밸런싱	Immutable 서버, 컨테이너 당 하나의 서비스 인스턴스, blue/green 배포	멀티 클라우드 및 멀티 데이터 센터 지원
팀구조	개발, QA, 릴리즈, 운영이 분리된 하나의 기능 팀	공유된 서비스 모델로 팀 공동 작업 내부 소스 공개	서비스별 프로젝트 팀(PO, UI/UX 디자이너, 개발자) Cross Functional한 플랫폼 팀	업무 기능별 혹은 도메인별 팀들이 모든 관점에서 책임을 수반. “네가 구축한 것은 네가 운영합니다.”

Overview - Vendor Neutral Curriculum

	AI-based Quick Overview	Lv1-Fundamental	Lv2-Intermediate	Lv3-Advanced
Biz	AI-based Agile SW Planning Tools	UML & Design Patterns - Object Oriented Analysis & Design - OOP Design Principles	Domain Driven Design - Business Analysis with Event Storming - Aggregate, Entity, Value Object Design	Agile Software Planning & Analysis - Customer Journey Mapping - Business Model Canvas - User Story Mapping / Example Mapping
Dev	AI-aided Development Environment - AI Coding Skills - Github Copilot+ +/Cursor IDE - Agent-based Development (GPT-Engineer) - Multi-agent Development Frameworks (CrewAI, LangGraph)	Development Environment - Version Control (Github) - AI IDE: VSCode with Copilot+ +/Cursor IDE Obejct Oriented Programming - Object Oriented Framework - Backend Dev with Spring boot, JDBC and JPA - Frontend Development (VueJS/React) Testing - Unit Test, E2E Testing - BDD / Mocking Software Conf. Mgmt & Build - Build automation tool (Gradle/Maven/npm) - GitHub & GitPod - Maven, Gradle	Microservice Design - Microservice Design Patterns - Hexagonal / Clean Architecture Request-Response Communication - Req/Res with FeignClient - Circuit Breaker Pattern - Data Project with GraphQL and Composite Services Event-driven Communication - Event Pub/Sub (Sagas), Compensation - Data Projection CQRS - Message Broker – Kafka Authentication/ Authorization - Token-based Authz/n (OAuth, JWT) - Centralized Session (Redis, Memcached) Application Packaging & Dockerizing	Advanced Event driven Microservices - Idempotency, Transaction, Parallel Processing, Deadline-handling - Integration with CDC - EDA Frameworks (Eventuate) - EDA Frameworks (Axon) Advanced Communication Style - gRPC Client/Server - Spring Reactive Developing LLM Applications - Python Basics - LLM app with Langchain - RAG - Multi-agent with LangGraph
Ops	AI-based Operation Tools - Kubectl AI - Kubectl GPT - k8s GPT	Deploying App on Cloud Infra - Multi-CSP Configuration (AWS, Azure, GCP) - Deploying App on VMs - Configuring Cloud DB, Storage, Networks	Container Orchestrator - Kubernetes Concepts & Objects - Managing SLA with Readiness & Liveness - Resource Mgmt., Persistence Volume Service Mesh (Istio) - Traffic Management (Retry, Timeout, Circuit breaker) - Security, Monitoring, EnvoyFilter, Tracing GitOps & CI/CD - Workflow with Jenkins & ArgoCD - Canary Deployment with Canary Scoring	Serverless Computing - KNative - Amazon Lambda Private Cloud Environment - Installing on-prem Kubernetes - Installing & Configuring DevOps Environment (Backstage, GitLab, Argo CD, Kubersphere) Private Gen-AI Environment - Installing Local LLM (Ollama) & Models - Installing Local AI-Coding Env

과정진행일정



교육생 선발

강의/수강관리

캡스톤 프로젝트

과정종료

1. 대상 선정을 위한 사전 평가

- Language, MSA, PaaS

2. 사전 설문을 통한 역량진단

- 모집 공고 및 안내
- 사전 역량 진단 설문

☑ 교육 목표 설정 및 선발기준 정의

☑ 선발 기준에 따른 사전 평가 문항
역량평가 설문서 작성

☑ Java 총 30 문항
Spring Framework : 총 15 문항
PaaS 사전역량 진단 : 총 60문항

☑ 지원자 모집 및 평가 실시

☑ 평가결과에 따른 선발자 확정 및 개별 통보

사전 평가문항 예시 : Spring Framework

사전평가 - Spring Framework

- 객체 지향 언어의 특징이 아닌 것을 고르시오.
 - 캡슐화(Encapsulation)
 - 상속(Inheritance)
 - 모듈화(Modularity)
 - 다형성(Polymorphism)
- 다음 스프링의 주요 특징 중에서 DI의 설명으로 틀린 것을 고르시오.
 - 각 클래스의 의존 관계를 참조하여 컨테이너가 자동으로 연결해준다.
 - 컨테이너 API에 종속되는 것을 줄일 수 있다.
 - 의존성 주입 종류로 Setter, Constructor, Method가 있다.
 - 각 객체의 생성부터 생명주기의 관리까지 모든 제어권을 제어하여 관리한다.
- IOC의 특징 중 알맞지 않은 것을 고르시오.
 - 개발자가 객체를 직접 생성하여 Framework에 제공한다.
 - POJO 제어권이 개발자에서 컨테이너가 전담한다.
 - Spring 컨테이너 혹은 IoC 컨테이너라고 부른다.
 - 컨테이너의 종류로 GenericXmlApplicationContext, WebApplicationContext 등이 있다.

사전 평가문항 예시 : Java

사전평가 - Java Fundamental

- 다음에 무엇에 대한 설명인지 적으시오. (중)

오버로딩 Overloading 중복정의

같은 일을 하는 메소드들의 매개변수의 타입이나 개수가 다를 경우 같은 이름을 사용할 수 있다.
- 다음은 무엇을 설명하는지 올바른 것을 고르시오. (하)

일반적으로 관련된 데이터와 기능을 클래스로 모아서, 데이터 및 기능을 보호하는 것

 - 상속 (Inheritance)
 - 다형성 (Polymorphism)
 - 추상화 (Abstraction)
 - 캡슐화 (Encapsulation)

사전 설문 - Cloud PaaS(Kubernetes)

Cloud Infrastructure Automation (공통)	1	Cloud 인프라 관리 자동화 장점과 특징을 설명할 수 있다.
	2	IaC에 대한 개념과 장단점을 이해하고 있는가?
	3	Cloud Infrastructure의 자동화 적용 영역을 이해하고 있는가?
	4	CSP에서 제공하는 자동화 툴을 제시하고 설명할 수 있는가?
	5	Terraform을 이용하여 AWS EC2 또는 Azure 인스턴스를 생성할 수 있는가?
	6	Terraform을 이용하여 VPC 및 네트워크 서비스를 생성할 수 있는가?
	7	Terraform을 이용하여 Load Balancer를 구성할 수 있는가?
	8	Terraform을 이용하여 Autoscaling 환경을 구성할 수 있는가?
	9	Terraform을 이용하여 보안 그룹 및 규칙을 생성할 수 있다.
	10	Terraform을 이용하여 EKS Cluster를 생성하고 운영할 수 있다.
	11	Terraform을 이용하여 CI/CD 파이프라인을 구성할 수 있다.
Kubernetes Operations (AWS, Azure)	12	멀티 Cloud 환경에서 Terraform을 이용하여 Cloud 인프라 자동화를 구현할 수 있는가?
	1	VM과 Container의 비교 설명을 할 수 있는가?
	2	Docker, Container, Kubernetes 관계를 이해하고 있는가?
	3	Kubernetes 구성 요소에 대해 설명할 수 있는가?
	4	Kubernetes의 아키텍처를 이해하고 있는가?
	5	Azure EKS의 아키텍처와 특징을 이해하고 있는가?
	6	Kubernetes의 기본 오브젝트(Pod, Replicaset, Deployment, Service)를 설명할 수 있는가?
	7	Kubernetes IAM을 통한 인증(Authentication)과 인가(Authorization)를 설명할 수 있는가?
	8	EKS 구축에 사용하는 도구들 설명할 수 있는가?
	9	Kubernetes Cluster를 구축할 수 있는가?
	10	EKS 클러스터에 API 애플리케이션을 배포할 수 있는가?
	11	Pod의 리소스를 명시하는 requests와 limits를 이해하고 있는가?
	12	Kubernetes Cluster 모니터링 방법을 이해하고 있는가?

교육생 선발

강의/수강관리

캡스톤 프로젝트

과정종료

1. 근태 및 운영 환경 관리

- 출결 관리 (결석, 지각, 조퇴)
- 교과정 안내 및 QnA

2. 강의 관리

- 진도관리 및 교육생 지원
- 검증된 교육 플랫폼 환경

☑ 운영담당자

- 고용노동부 HRD-Net 기반 출결 관리
- 데일리 만족도 조사를 통한 개선 의견 수렴
- 학습지원 및 교육장 내 시설(PC 등) 문제 수렴
- 진행 시간 안내

☑ 교육강사

- 검증된 MSA 교육 플랫폼 활용
- 실시간 교육생 진도 파악 및 질의 관리
- 수강생 화면지원

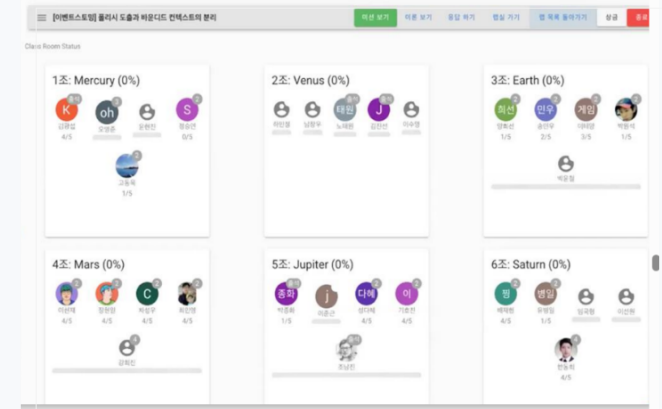
☑ 교육생

- 브라우저 기반 편리한 실습환경
- 실습결과 제출 및 시 기반 자체 문제해결

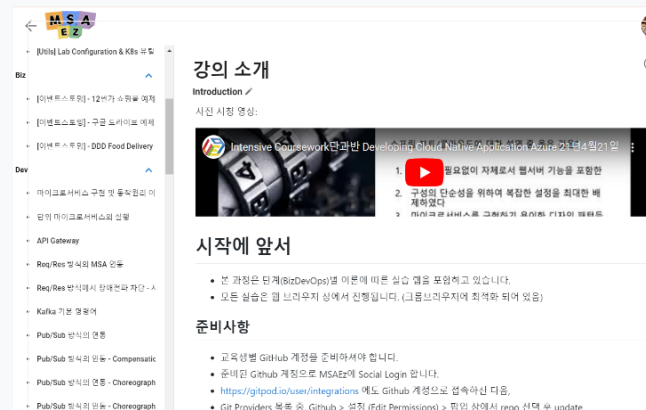
출결 관리 (운영담당자)



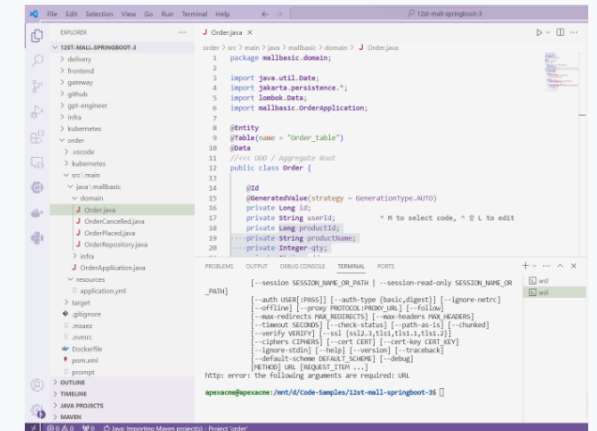
진도관리 (수강생 화면 제출 → 강사)



다년간 검증된 교육 플랫폼 활용 (강사)



시 기반 문제 자체 해결 (교육생)



교육운영 03. 단계별 미니 프로젝트 및 최종 Final Project 실시



1. 단계별 미니 프로젝트 수행

- RFP기반 미니 프로젝트
- 수행 결과물 수렴/ 평가

2. 파이널 프로젝트 및 평가

- 평가 플래닝(주제, 개인/그룹)
- 최종 발표 및 Assessment

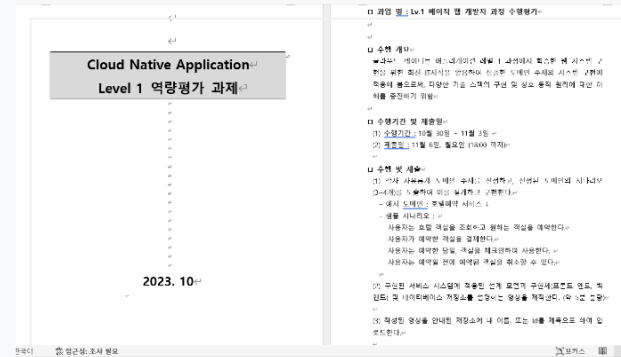
☑ 매 단계 종료 시 마다 미니 프로젝트 실시

☑ 미니 프로젝트를 위한 RFP 작성 및 배포 기간 내 수행 및 평가 실시

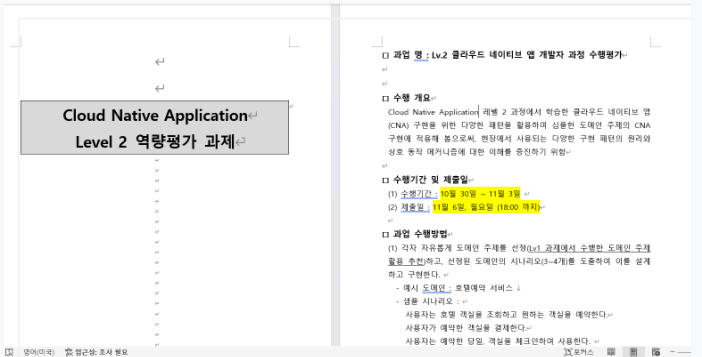
☑ 종합 필기시험 및 실습평가를 위한 캡스톤 프로젝트 병행 실시

☑ 필기 문항지 구성 : 총 30 문항
캡스톤 프로젝트 플래닝(개인별, 팀별)
수행평가를 위한 가중치가 설정된
체크 포인트 작성
평가위원 구성 및 발표 평가

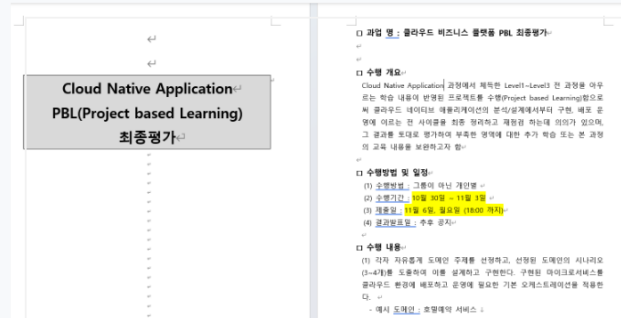
Level 1 미니프로젝트 RFP



Level 2 미니프로젝트 RFP



Level 3 미니프로젝트 RFP



클라우드 네이티브 역량강화 Final Assessment Checkpoints			
구분	Cloud Native 역량	구성	배점
클라우드 네이티브 아키텍처 (IaaS)	클라우드 아키텍처 설계	클라우드 아키텍처 구성, MSA 아키텍처 구성도	10
클라우드 네이티브 모델링 (Biz)	Data Modeling/ 서비스 분리/설계 역량	도메인분석 - 이벤트스트로밍	10
클라우드 네이티브 개발 MSA (Dev)	MSA 개발 또는 개발관리 역량	분산트랜잭션 - Saga	25
		보상처리 - Compensation	
		단일 진입점 - Gateway	
클라우드 네이티브 운영 (PaaS)	클라우드 배포 역량	분산 데이터 프로젝트션 - CQRS	45
		클라우드 배포 - Container 운영	
		컨테이너 자동확장 - HPA	
		컨테이너로부터 환경분리 - CofigMap/Secret	
		클라우드스트로리지 활용 - PVC	
클라우드 네이티브 운영 (PaaS)	컨테이너 인프라 설계 및 구성 역량	생프 윗팅/무정지배포 - Liveness/Rediness Probe	45
		서비스 메쉬 응용 - Mesh	
		통합 모니터링 - Loggregation/Monitoring	
총점 (100점)			



1. 최종 평가 결과 사후처리

2. 종료관리

- 만족도 평가
- 수료여부 확인
- 현업 담당부서에 전달

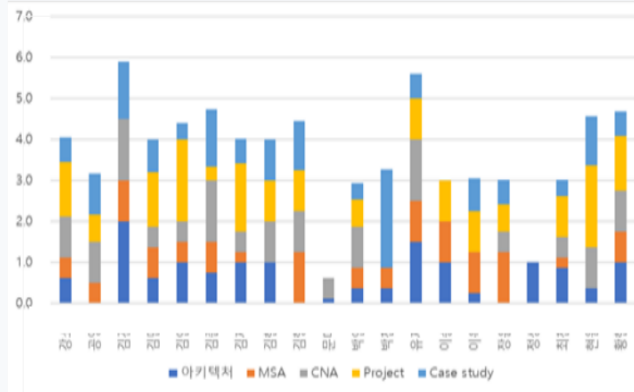
✓ 과정 종료에 따른 최종 종합 평가

✓ 수료 여부에 따른 수료증 발급

✓ 개별/팀별 Radar Chart 기반 Skill Set 분석 및 보완 내용 추천

✓ 평가를 통해 상위 10%인력은 즉시 현장 프로젝트에 투입

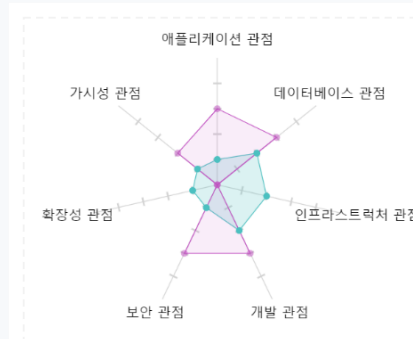
종합 평가



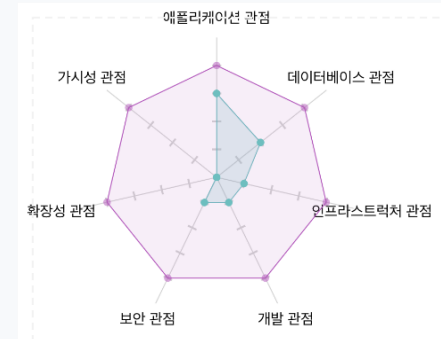
종합평가 상세

단계	구분	체크 포인트
분석/설계	마이크로 서비스 모델링	이벤트스트림의 스티커를 이해하고 있다. ^{1,2} 비즈니스 시나리오를 이벤트스트림으로 모델링할 수 있다. ^{3,4} 바운딩 컨텍스트간 매핑 관계를 이해하고 적용할 수 있다. ⁵
	구현	이벤트스트림을 활용한 서비스간 통신 원리를 이해하고 있다. ^{1,2} 도메인 이벤트에 관심있는 마이크로서비스의 전수명 (Listener Subscriber) 로직을 이해하고 설명할 수 있다. ^{3,4} Saga의 개념과 Saga 트랜잭션의 동작원리를 이해하고 있다. ⁵
데이터 로직성	구현	구현하는 서비스에 Read Model 스티커를 활용하는 CQRS 데이터 모델이 존재하여야 한다. ^{1,2} CQRS 개념을 이해하고 있으며, 구현한 서비스를 통해 설명할 수 있다. ^{3,4}
	게이트웨이 (Gateway)	구현하는 서비스에 게이트웨이 서비스가 포함되어야 한다. ^{1,2} 서비스들의 단일 진입점인 게이트웨이를 이해하고 있다. ^{3,4} 게이트웨이에 트래픽 라우팅을 적용하고 설명할 수 있다. ⁵
배포/운영	자동 스케일 아웃	클라우드에 배포된 서비스에 사용자 요청이 가중 되었을 때, 트래픽을 분산 처리하기 위한 컨테이너의 자동 증가가 설정되어 운영되고 있는가? ^{1,2}
	무정지 배포	새로운 서비스가 배포될 때, 기존 서비스의 중단 없는 Seamless한 배포가 수행되도록 서비스가 배포되어 동작하고 있는가? ^{3,4}
자동 복구	데이터 백업	배포된 마이크로서비스가 비정상적인 경우, 자체적으로 회복하여 최대한 안정적인 서비스가 유지되도록 컨테이너 플랫폼과 배포된 서비스간 이를 지원하는 메커니즘이 설정되어 운영되고 있는가? ^{1,2}

개별평가 결과 - 홍길동



팀 평가 결과



Mentors and Coaches

MSA School 대표 강사



장진영

유엔진솔루션즈 대표이사

MSA School 대표 강사



박용주

유엔진솔루션즈 이사

MSA School 대표 강사



윤성열

드림플로우 대표이사

MSA School 대표 강사



손영수

어니컴 CPO

MSA School 대표 코치



Coaches

- SFAe 공인 컨설턴트
- **SDS, SK, LGU+ 클라우드 아키텍트 과정** 주강사
- (전) OMG / SEMAT 국제 표준 도구 분과 위원
- 現 디지털플랫폼정부 기술자문위원
- 유엔진BPMS / 오픈클라우드 엔진 파운더 클라우드 네이티브 (MSA, DDD) 강의
- MSA와 SW모델링 Facebook그룹 운영 (www.facebook.com/groups/cloudswmoding)

- 現 MSA App. Engineering 기업과정
- 現 세종사이버대학교 컴퓨터/AI 공학과 겸임교수
- **SDS, SK, LGU+ 클라우드 아키텍트 과정** 주강사
- 한국소프트웨어기술진흥협회 전문강사
- '21 : SK MSA App. Engineering 과정
- '21.06 : KT Microservice 직무전환과정
- '20.09 : Doosan Microservices 교육
- '19.09 : KOSTA Microservices 교육
- '19.02 : LG CNS 이벤트스토밍 교육

- 現 드림플로우 대표이사
- 現 한국소프트웨어기술진흥협회 BAPF
- **SDS, 신한, KT 클라우드 아키텍트** 주강사
- 포럼, 교육과정위원회 및 전문강사
- '18.10 : 원오원글로벌 디지털팀 팀장
- '18.04 : TTA 사물인터넷 특별기술위원회 사물인터넷 융합서비스 프로젝트그룹 (SPG11) 부의장
- '17.03 : 가천대학교 겸임교수

- 現 Onycom 성능솔루션
- 사업부 상무/CPO
- **SDS 아키텍트 과정** 주강사
- NHN NEXT 모바일/창업 과정 교수
- 삼성전자 소프트웨어 센터 책임
- 삼성전자 삼성페이 서버 아키텍팅 과정 강의
- KT CleanCode 및 소프트웨어 아키텍트 과정

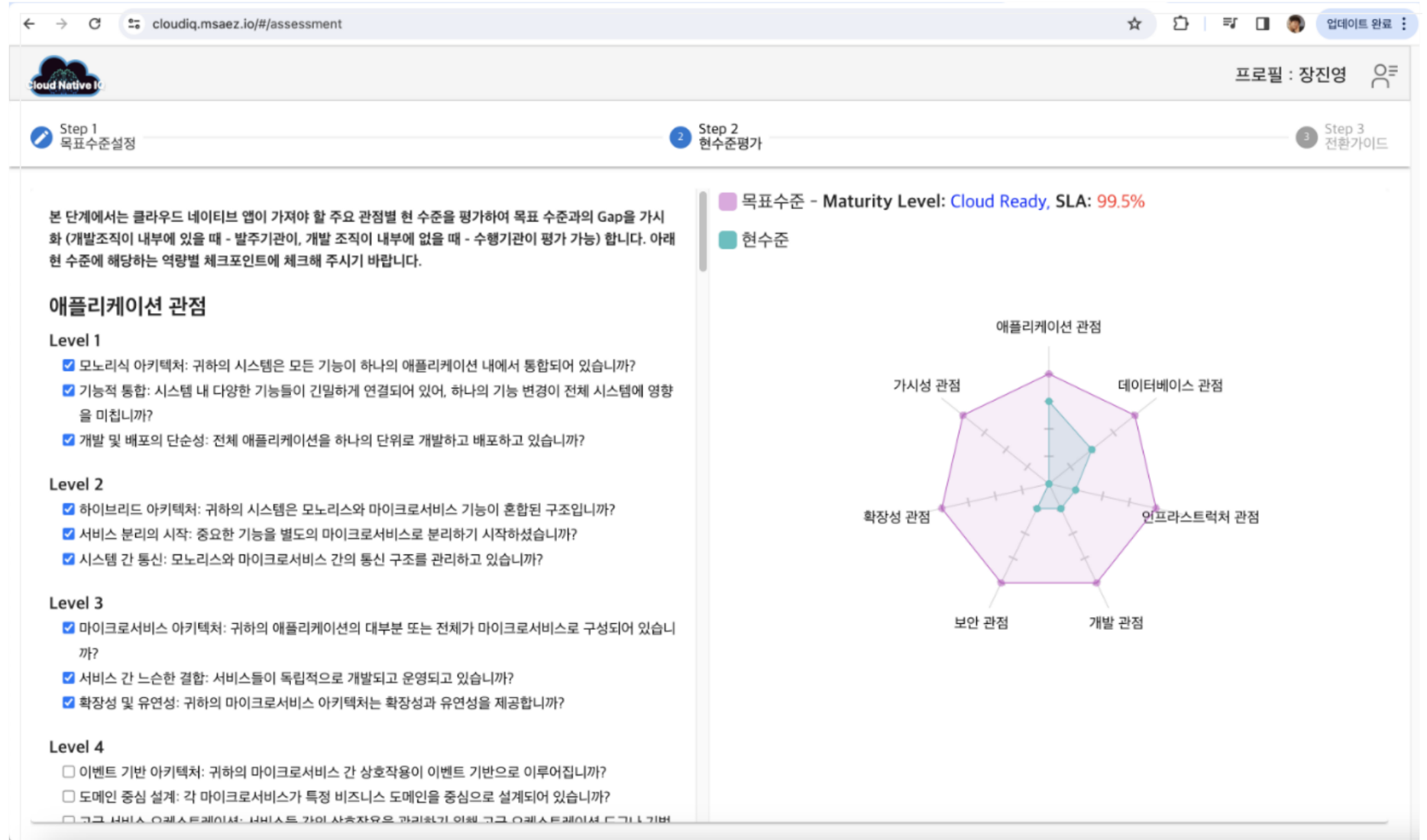
- 現 MSA 정규 교과정 실습 강사 및 코칭
- 現 MSA App. Engineering 기업과정 실습 강사 및 코칭
- SDS, Doosan, LG, SK 외 다수의 교육과정 실시



✓ CNCF CMM 성숙도 모형에 근거한
팀별 / 개인별 수준 평가

✓ 평균/중간 값에 개인 수준 비교

✓ 개개인 역량 개선을 위한 맞춤형
보완 LO추천

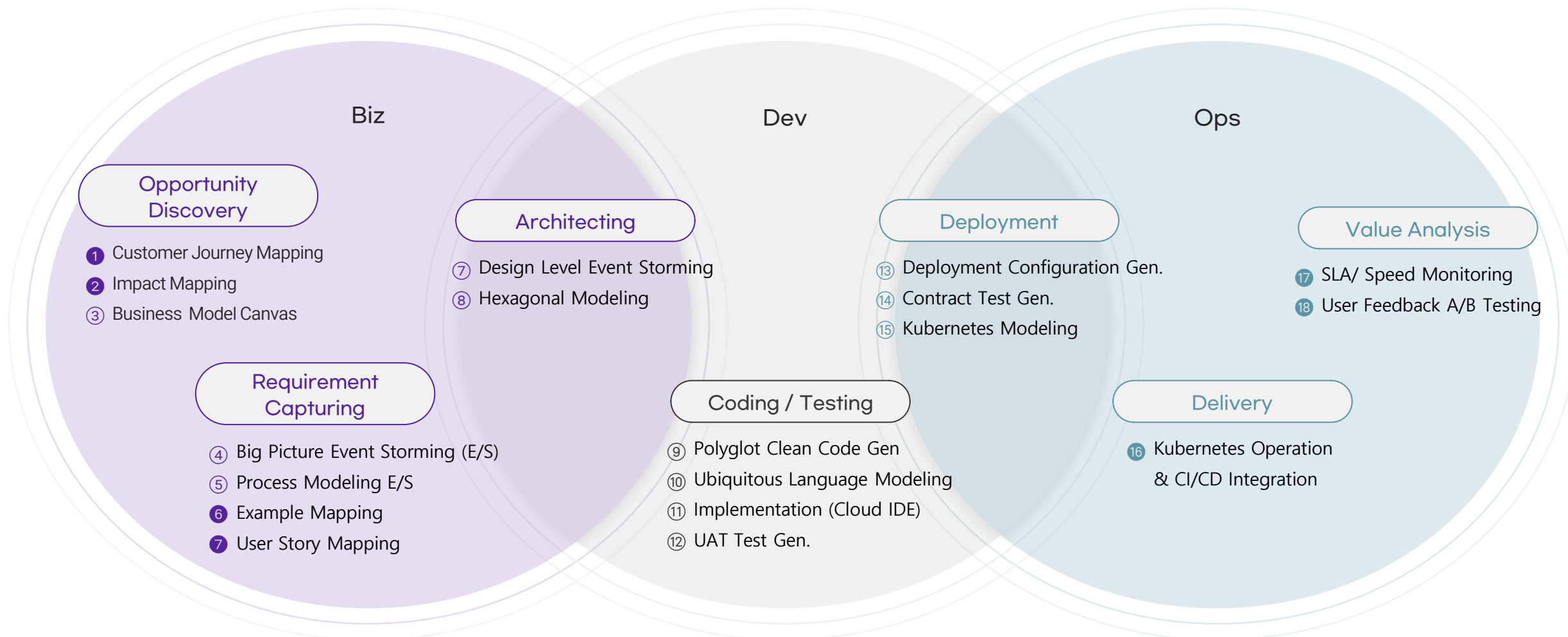




“ 본 강의와 같은 진행 및 교육 방식이라면 온라인 강의여서 더 좋다고 느꼈습니다.
MSA Easy 에서 개발하고 직접 내 환경에 강사가 들어와 봐주는 방식, 강사가 일일이 봐주려고 하는 느낌!”



✓ The OpenGroup의 Agile Architecture Framework의 주요 방법론으로 채택된 방법론들을 협업도구를 통하여 실습

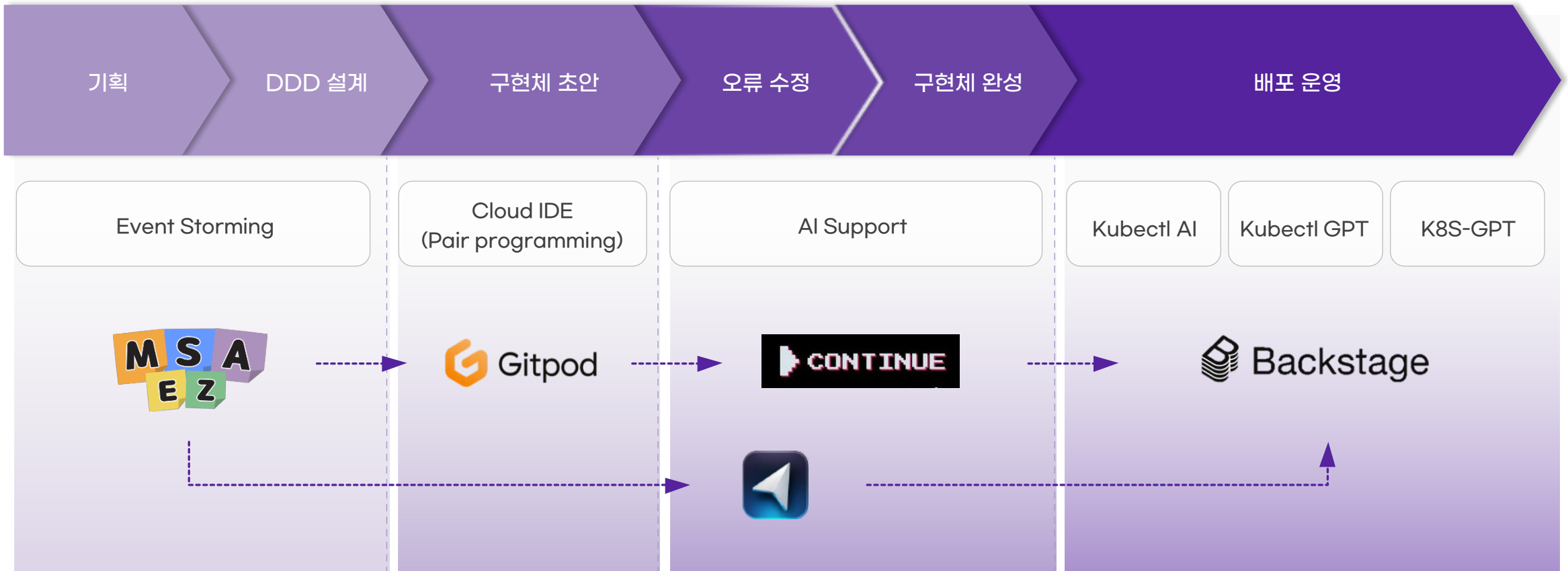


주요교구 클라우드 IDE 기반 실시간 강사 지원 및 AI Pair 프로그래밍

☑ 클라우드 IDE를 통한 수강생 전원 동일한 실습환경 구성과 생성 및 폐기가 용이하여 강의 진행 원활

☑ 구현단계에 애로가 있는 수강생의 IDE에 강사가 직접 문제해결 혹은 Pair Coding 통하여 이끌어 줄 수 있는 환경

☑ AI 기반으로 강사에게 질문하기 어려운 간단한 문제나 기반 개발 상식등을 부끄럼 없이 질문할 수 있는 환경



평점 **4.7**
★★★★★

총 교육횟수 **257회**

총 교육수강생 **7,410명**

총 교육시간 **9,840시간**

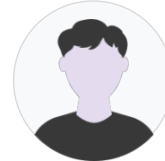
교육 콘텐츠

교재 내용이 다양하게 준비되어 있어 이해에 도움이 많이 되었고, 짧은 시간 안에 클라우드 환경 설정부터 배포까지 실습을 진행할 수 있는 부분이 좋았습니다.



실습 환경

MSAEZ 툴과 GitPod를 활용해 번거로운 설치 과정이나 환경설정 필요 없이 바로 실습을 진행할 수 있는 점이 좋았습니다.



강사 평가

새로운 개념을 계속 배우다 보니 따라가기 벅찬 경우가 있는데, 이론 및 실습 진행에 있어 적당한 진행 속도와 전문적인 지식으로 좋은 강의 해주셨습니다.



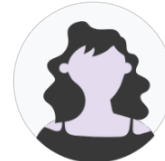
교육 콘텐츠

예제와 실습이 충분해서 좋았고, DDD가 실제 코드로 이행되는 과정과 그 과정을 위한 툴의 연계 등 유익한 강의였습니다.



실습 환경

웹 브라우저 환경에서 곧바로 이벤트스토밍부터 코드 생성, 개발 작업까지 연계되는 과정이 마음에 들었습니다.



강사 평가

강사님과 보조강사님 모두 질문에 친절하게 답해주시고 낙오자가 없도록 쉽고 상세한 설명을 해주십니다.



Appendix: 세부커리큘럼

커리큘럼 : AI 기반 구현 (이용자 관점)

교육목표	AI 를 이용하여 효율적으로 클라우드 네이티브 수준의 애플리케이션을 기획-분석-설계-구현-운영 하는 전과정을 이해하고 실습함				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	소프트웨어 개발 일반, 자바 문법 이해 (초급) 수준

교육내용	기간	주제	시간	주요 학습내용
	Week 1	AI 기반 기획/설계	2 Days	- AI 기반의 소프트웨어 개발 도구들의 이해 (1단계~5단계) - Cursor IDE 를 이용한 피자 주문 앱 구현 - 실습 - Cursor IDE 를 이용한 레가시 분석 및 모더나이징 - 실습 - 클라우드 네이티브 아키텍처 / MSA / 클린 아키텍처의 이해 - 도메인주도설계 방식과 AI 기반 애플리케이션 개발의 효율성 - ChatGPT 를 통한 큰 규모의 레가시 애플리케이션 분석
		AI 기반 구현/운영	3 Days	- GPT-Engineer 이용한 클라우드 네이티브 애플리케이션 완성 - 실습 - Kubectl ai 를 통한 쿠버네티스 스크립트 생성 - 실습 - Kubectl gpt 를 통한 쿠버네티스 명령 - 실습 - K8S-GPT를 통한 운영 / 트러블 슈팅 자동화 - 실습 예제) EDA(이벤트 드리븐 아키텍처) 기반 음식 배달앱

AI 기반 기획/설계

- ☑ AI 도구를 활용하여 MSA의 기획 및 설계 단계 이해
- ☑ 프로젝트 아이디어 구상 및 구현 계획 수립
- ☑ 사용자 요구사항을 기반으로 한 데이터 및 모델 설계

AI 기반 구현/운영

- ☑ 클라우드 네이티브 환경에서 AI 기반 애플리케이션의 구현 및 운영 방법 학습
- ☑ 도커와 쿠버네티스를 활용한 환경 설정 및 애플리케이션 배포 방법 이해
- ☑ 실시간 데이터 처리를 위한 이벤트 드리븐 아키텍처 개념 및 적용 방법 습득

커리큘럼 : LV1 Fundamental

교육목표	클라우드 네이티브 프로그래밍 기초 확립과 VM 기반의 애플리케이션 개발 능력을 이해하고 실습함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 2	자바 및 객체지향	2 Days	- 객체지향 프로그래밍 개요 및 리뷰 그리고 마이크로서비스와의 관계 - 객체지향 프로그래밍의 특징 - 클래스와 객체 개념 - 클래스 생성과 활용 - 인터페이스 개념 및 활용 - 인터페이스의 다중 구현 - 추상 클래스와 인터페이스 실습 - 실습 람다 표현식 소개 람다 표현식을 이용한 인터페이스 구현 람다 문법 실습 - 실습 스트림 개념 스트림 연산 및 활용 병렬 스트림 처리 스트림 실습 - 실습
		디자인 패턴	2 Days	- 디자인 패턴의 개요 - 디자인 패턴의 종류 - 생성 패턴 : 싱글턴 패턴 - 생성 패턴 : 팩토리 메소드 패턴 - 구조 패턴 : 어댑터 패턴 - 구조 패턴 : 퍼사드 패턴 - 행동 패턴 : 옵저버 패턴 - 행동 패턴 : 스트레티지 패턴
		레이어드 아키텍처와 클린 아키텍처의 관계	1 Day	- 레이어드 아키텍처 개요 - 레이어드 아키텍처의 구성 요소 - 레이어 간의 상호작용

커리큘럼 : LV1 Fundamental

교육목표	클라우드 네이티브 프로그래밍 기초 확립과 VM 기반의 애플리케이션 개발 능력을 이해하고 실습함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 3	UML	1 Days	• UML 소개 및 역할 • UML 의 기본 요소 • UML 작성 툴 소개 및 설정 • 유즈케이스 다이어그램 • 유즈케이스 다이어그램 실습 - 실습 • 클래스 다이어그램 • 클래스 다이어그램 실습 - 실습 • 시퀀스 다이어그램 • 시퀀스 다이어그램 실습 - 실습
		마이크로 서비스 프레임워크 : 스프링 부트	3 Days	• 스프링 부트 개요 및 구성 • 스프링 부트 개발 환경 설정 및 프로젝트 생성 - 실습 • 스프링 빌드 도구(Maven, Gradle) • IDE 세팅과 스프링 구성 실습 - 실습 • 스프링 컨테이너와 빈 • 의존성 주입(Dependency Injection) • 스프링 AOP(Aspect-Oriented-Programming) • 스프링 MVC 소개 및 구조 • HTTP와 웹서버 개념 • 컨트롤러와 요청 매핑 • 뷰와 뷰 리졸버
	Week 4	도메인 클래스 모델링과 데이터베이스	2 Days	• 도메인 클래스 모델링 • Java Persistence APIs(ORM)과 데이터베이스 • 스프링 트랜잭션 관리 @Transactional • 스프링 데이터 소개 • 스프링 데이터 설정 - 실습 • 스프링 데이터 실습(spring initializer) - 실습 • 스프링 데이터 JPA • JPA 엔티티 매핑 • JOA 엔티티 실습 - 실습

커리큘럼 : LV1 Fundamental

교육목표	클라우드 네이티브 프로그래밍 기초 확립과 VM 기반의 애플리케이션 개발 능력을 이해하고 실습함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 4	스프링 데이터와 REST	2 Days	- JPA 쿼리 애소드 - JPA 페이징 및 정렬 - JPA Repository 실습 - 실습 - JPA 연관 관계 매핑 - JPA 연관 관계 매핑 실습 - 실습 - 스프링 데이터 고급 기능 : 캐시, 동적 쿼리, 이벤트 처리 - 스프링 데이터 고급 기능 : 캐시, 동적 쿼리, 이벤트 처리 - RESTful API 소개 및 설계 원칙 - RESTful API 응답 형식 및 버전 관리 - 스프링 데이터 REST - 스프링 데이터 REST 실습 - 실습
		웹 서버 테스트	0.5 Days	- JUnit 소개 - JUnit 문법 및 테스트 코드 작성 방법 - JUnit 실습 - 실습
		Swagger 및 API	0.5 Days	- Swagger 소개 및 설치 - Swagger 문서 작성 및 UI 사용법 - Swagger 실습 - 실습

커리큘럼 : LV1 Fundamental

교육목표	클라우드 네이티브 프로그래밍 기초 확립과 VM 기반의 애플리케이션 개발 능력을 이해하고 실습함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 5	웹 기초	3 Days	• HTML/CSS/Javascript • JavaScript 기본 문법 • JavaScript DOM 조작 • JavaScript 이벤트 처리 • JavaScript 비동기 처리 (AJAX) • Javascript 클래스
		React.js	4 Days	• React.js 이벤트 처리 실습 - 실습 • React.js 컨텍스트 API • React.js 라우팅 (React Router) • React.js 라우팅 실습 - 실습 • React.js 비동기 데이터 처리 (Ajax, axios) • React.js 컨텍스트 API와 비동기 처리 실습 - 실습
	Week 6	MUI	2 Days	• MUI 기본 및 UI 컴포넌트 구성 • MUI 컴포넌트 실습 - 실습 • MUI 를 사용한 레이아웃 및 반응형 디자인 • MUI 레이아웃 실습 - 실습 • MUI 를 사용한 폼 및 입력 컴포넌트 구성 • MUI 를 사용한 데이터 표시 컴포넌트 구성 • MUI 폼/데이터 실습 - 실습
		가상머신	1 Days	• 가상 머신 환경 구성 소개 • 가상 머신 설치 및 설정 - 실습 • 가상 머신 네트워크 구성 • 가상 머신 네트워크 구성 실습 - 실습 • 가상 머신 리소스 관리 및 최적화 • 가상 머신 리소스 관리 및 최적화 실습 - 실습

커리큘럼 : LV1 Fundamental



교육목표	클라우드 네이티브 프로그래밍 기초 확립과 VM 기반의 애플리케이션 개발 능력을 이해하고 실습함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 7	미니프로젝트	5 Days	• 웹 및 Java를 이용한 서버 구축 • 요구사항에 맞추어 프로젝트 기능 명세 개발 • 스프링 부트 기반 백엔드 서버 개발 • React.js 기반 프론트엔드 서버 개발 • VM 환경에 배포

객체지향적 사고와 설계 마스터하기

- ✔ 객체지향 프로그래밍의 원리와 마이크로서비스 아키텍처와의 연관성 이해
- ✔ 다양한 디자인 패턴을 통한 효율적인 소프트웨어 설계 방법 학습
- ✔ UML을 사용한 시스템 설계 문서화

스프링 기반의 마이크로서비스 구축 및 관리

- ✔ 스프링 부트와 스프링 기반의 RESTful API 개발 및 배포 방법 습득
- ✔ 도메인 주도 설계(DDD)와 JPA를 활용한 데이터베이스 통합 방법 이해
- ✔ 테스트 주도 개발(TDD)을 위한 웹 서버 테스트 방법과 Swagger를 사용한 API 문서화

모던 웹 애플리케이션 개발

- ✔ HTML, CSS, JavaScript를 이용한 웹 기초와 프론트엔드 개발 스킬 습득
- ✔ React.js와 MUI를 활용한 동적이고 반응형 웹 애플리케이션 구축
- ✔ 가상머신을 활용한 개발 환경 구성과 미니 프로젝트를 통한 종합적인 실습 및 응용

커리큘럼 : LV2 Intermediate

교육목표	업계 표준 개발 툴과 실제 클라우드 프로젝트에서 요구되는 실무 중심의 최신 MSA 기술 및 글로벌 개발표준을 구현하고 실습함				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 8	마이크로서비스, MSA	2.5 Days	- MSA 소개, 모노리식서비스와 마이크로서비스 - REST 연동과 비동기 메시지 연동, MSA 장단점 - 쇼핑몰 (12 번가)을 주제로 한 개발 서비스 목표 및 조직의 전화 - 도메인 문제해결을 위한 접근 전략 - 이벤트 드리븐 아키텍처 및 애니메이션을 통한 이해 - 도메인 주도 설계 (Domain-Driven Design) - 마이크로서비스 서열과 역학관계
		마이크로서비스 설계 (DDD & 이벤트스토밍)	2.5 Days	- 이벤트스토밍 (Eventstorming) 소개 - 글로벌 기업들의 마이크로서비스 설계 방법론 엿보기 - 이벤트스토밍 온라인 도구 소개 - 실습 - 이벤트스토밍 (Eventstorming) 활용 - Event, Command, Actor - 실습 - 이벤트스토밍 (Eventstorming) 활용 - Policy, Aggregate, External - 실습

커리큘럼 : LV2 Intermediate



교육목표	업계 표준 개발 툴과 실제 클라우드 프로젝트에서 요구되는 실무 중심의 최신 MSA 기술 및 글로벌 개발표준을 구현하고 실습함				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 9	마이크로서비스 설계 (DDD & 이벤트스토밍)	5 Days	- 이벤트스토밍 (Eventstorming) 활용 - Bounded Context, Context Mapping - 실습 - 이벤트스토밍 (Eventstorming) 활용 - 모델 저장 / 불러오기 / 복제하기 - 실습 - 시나리오 기반 이벤트스토밍 쉽게 적용하기 - 호텔 도메인 12st 쇼핑몰 도메인 이벤트스토밍 - 실습 - Google Drive 도메인 이벤트스토밍 - 실습 - Food Delivery App 도메인 이벤트스토밍 - 실습

커리큘럼 : LV2 Intermediate

교육목표	업계 표준 개발 툴과 실제 클라우드 프로젝트에서 요구되는 실무 중심의 최신 MSA 기술 및 글로벌 개발표준을 구현하고 실습함				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 10	마이크로서비스 설계 (DDD & 이벤트스토밍)	5 Days	- 이벤트스토밍 확장 - 디자인 레벨 이해 - 실습 12st 쇼핑몰 도메인 디자인 레벨 이벤트스토밍 - 실습 - Google Drive 도메인 디자인 레벨 이벤트스토밍 - 실습 - Food Delivery App 디자인 레벨 이벤트스토밍 적용 - 실습 - 교구를 활용한 협업 이벤트스토밍(Collaborative Eventstorming) 맛보기 - 실습

커리큘럼 : LV2 Intermediate

교육목표	업계 표준 개발 툴과 실제 클라우드 프로젝트에서 요구되는 실무 중심의 최신 MSA 기술 및 글로벌 개발표준을 구현하고 실습함				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 11	마이크로서비스 구현	5 Days	• 마이크로서비스 구현 패턴 및 구현기술 • 마이크로서비스 개발도구 소개 - VSCode 설치 및 설정 - 실습 • 빌더 도구 (Maven - 메이븐) 소개 - 실습 • 랩에 사용되는 Spring-Boot 라이브러리 소개 및 활용 - 실습 • REST API 소개 및 기본 Commands 이해 - 실습 • 단위 마이크로서비스 구현 및 동작원리 이해 - 실습 • 단위 마이크로서비스 구현 및 동작원리 이해 - Kafka 이벤트스토퍼 엮보기 - 실습

커리큘럼 : LV2 Intermediate

교육목표	업계 표준 개발 툴과 실제 클라우드 프로젝트에서 요구되는 실무 중심의 최신 MSA 기술 및 글로벌 개발표준을 구현하고 실습함				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 12	마이크로서비스 인터-커뮤니케이션 (Request /Response)	2.5 Days	• 모노리스에서 마이크로서비스 전환시 고려요소 - 1, Strangler 패턴 • 동기호출 (Request/Response) 방식의 연동 (코드 리뷰 및 실행) - 실습 • 동기호출(Request/Response) 방식의 연동 (서비스 테스트) - 실습 • 동기호출(Request/Response) 방식에서의 장애전파차단 - 실습
		마이크로서비스 인터-커뮤니케이션 (Pub/ Sub)	2.5 Days	• 모노리스에서 마이크로서비스 전환시 고려요소 - 2, Saga 패턴 • 이벤트 스토어 - Kafka 소개 • Kafka 로컬 설치 및 서버 기동하기 - 실습 • Kafka 기본 명령어 실습 - 실습

커리큘럼 : LV2 Intermediate

교육목표	업계 표준 개발 툴과 실제 클라우드 프로젝트에서 요구되는 실무 중심의 최신 MSA 기술 및 글로벌 개발표준을 구현하고 실습함				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 13	마이크로서비스 인터-커뮤니케이션 (Pub/ Sub)	5 Days	• Pub /Sub 방식의 인터 커뮤니케이션 실습 - 코드리뷰 및 실행 - 실습 • Pub /Sub 방식의 인터 커뮤니케이션 실습 - 실습 및 Kafka 토픽확인 - 실습 • Pub /Sub 방식의 인터 커뮤니케이션 실습 - 확장시나리오 - 실습 • Pub /Sub 방식의 인터 커뮤니케이션 실습 (Compensation, Correlation) - 실습 • Pub /Sub 방식의 인터 커뮤니케이션 실습 (Compensation, Correlation) - 확장시나리오 - 실습 • Kafka 동적 확장 (Scaling) - 실습 • Kafka 확장과 동시성 처리 (Concurrency Handling) - 실습 • CDC(Change Data Capture) with Kafka - 실습

커리큘럼 : LV2 Intermediate



교육목표	업계 표준 개발 툴과 실제 클라우드 프로젝트에서 요구되는 실무 중심의 최신 MSA 기술 및 글로벌 개발표준을 구현하고 실습함				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 14	마이크로서비스 인터-커뮤니케이션 (Pub/ Sub)	5 Days	• CDC(Change Data Capture) 실습 - 실습 • Axon 프레임워크 소개 • Axon Framework 기반 마이크로서비스 구현 - 코레오그래피 Saga - 실습 • Axon Framework 기반 마이크로서비스 구현 - 오케스트레이션 Saga 모델링 - 실습 • Axon Framework 기반 마이크로서비스 구현 - 오케스트레이션 Saga - 실습

커리큘럼 : LV2 Intermediate



교육목표	업계 표준 개발 툴과 실제 클라우드 프로젝트에서 요구되는 실무 중심의 최신 MSA 기술 및 글로벌 개발표준을 구현하고 실습함				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 15	API 게이트웨이 및 인증인가	5 Days	• API 게이트웨이 소개 • API 게이트웨이 실습 - 실습 • 마이크로서비스에 통합인증 적용하기 - 실습 • 마이크로서비스 통합인증 (Password GrantType) 실습 - 실습 • 마이크로서비스 통합인증 (Authorization Code) 실습 - 실습

커리큘럼 : LV2 Intermediate

교육목표	업계 표준 개발 툴과 실제 클라우드 프로젝트에서 요구되는 실무 중심의 최신 MSA 기술 및 글로벌 개발표준을 구현하고 실습함				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 16	서비스 통합 (Integration)	2.5 Days	• 프론트엔드에서의 서비스 통합 • 프론트엔드 실습 - 실습 • Req /Res, Pub/ Sub 기반 서비스 통합 • 마이크로서비스 분산환경에서 데이터 프로잭션 • 데이터 통합 패턴 : GraphQL 실습 - 실습 • CQRS 기반 데이터 통합 실습 - 실습
		마이크로서비스 테스트	2.5 Days	• 컨트랙트 테스트란 ? • 컨트랙트 테스트 실습 (Request/ Response 방식) - 실습 • 컨트랙트 테스트 실습 (Message driven 방식) - 실습

커리큘럼 : LV2 Intermediate



교육목표	업계 표준 개발 툴과 실제 클라우드 프로젝트에서 요구되는 실무 중심의 최신 MSA 기술 및 글로벌 개발표준을 구현하고 실습함				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 17	미니프로젝트	5 Days	• DDD를 이용한 마이크로서비스 설계 • MSA 설계 적용 및 구현 • 인프라 구축 및 서버 배포 • 인증 적용

마이크로서비스 개념 이해 및 설계

- ☑ 모노리식 서비스와 마이크로서비스 비교 분석
- ☑ RESTful API 및 비동기 이벤트 기반 통신 매커니즘 학습
- ☑ 이벤트 드리븐 아키텍처와 DDD를 통한 마이크로서비스 설계 원칙 이해

마이크로서비스 개발 프로세스와 도구 습득

- ☑ 이벤트스토밍을 통한 복잡한 도메인 문제 해결 접근 전략 습득
- ☑ 실제 비즈니스 시나리오에 적용하는 이벤트스토밍 실습
- ☑ 마이크로서비스의 개발 도구 및 프레임워크 활용 기술 습득

마이크로서비스 운영 및 테스트 전략

- ☑ API 게이트웨이를 통한 서비스 연동 및 인증인가 처리 방법 학습
- ☑ 마이크로서비스 기반 아키텍처에서의 데이터 통합 패턴 구현
- ☑ 컨트랙트 테스트를 활용한 마이크로서비스 테스트 전략과 실습

커리큘럼 : LV3 Advanced

교육목표	확장된 애자일 SW 개발 프랙티스들을 학습한다.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	없음

교육내용	기간	주제	시간	주요 학습내용
	Week 18	애자일 SW 기획/설계	1.5 Days	- 애자일 SW 개발 프로세스의 이해 - 페르소나를 통한 애자일 소프트웨어 기획 - 고객 여정맵을 통한 사용자 경험 분석 - 사용자 경험의 고통점(페인포인트)와 솔루션 기회 분석 - 임팩트 매핑을 통한 개선 목표 수립 - 비즈니스 모델 캔버스를 통한 비즈니스 전략 수립 - 도메인 별 Prioritization Map (Core Domain Chart)을 통한 우선개발 범위 식별
		애자일 SW 분석	1 Days	- 사용자 스토리 매핑을 통한 프로젝트 계획 수립 - 예제 매핑 (Example Mapping)과 테스트 주도 개발 - BDD (Behavior Driven Development) 의 이해와 적용 - Scaled Agile Framework (SAFe) 의 이해
		데브옵스 프로세스 이해	0.5 Days	- 데브옵스 프로세스와 컨테이너 기반 DevOps 이해 - 클라우드 네이티브 앱 전 생명주기 이해
		도커기반 어플리케이션 패키징	2.5 Days	- 도커개념 및 이미지 이해하기 (이미지 규칙 소개) - 도커 허브와 도커 런타임 - 도커 이미지 생성 & Dockerfile - 이미지와 컨테이너 관계 실습 - 실습 - 이미지 생성 및 저장소 푸쉬 - 실습 - 이미지 기반 컨테이너 생성 및 새버전 배포 - 실습

커리큘럼 : LV3 Advanced

교육목표	경쟁력 있는 클라우드 애플리케이션 설계를 구현하고 실전 응용 가능한 클라우드 인프라를 이해하고 실습함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 19	클라우드 플랫폼	2 Days	- 컨테이너 오케스트레이션 개념과 쿠버네티스 - 인프라 관점에서의 클라우드 플랫폼 [AWS] CLI기반 AWS 및 쿠버네티스 Client 설치 및 소개 - 실습 [AWS] 관리콘솔을 통한 IAM 계정 생성 및 권한할당 - 실습 [AWS] 보안자격증명 생성 및 클라이언트 설정 준비 - 실습 [AWS] 기반 컨테이너 오케스트레이션 환경 설정 -1 EKS 생성 - 실습 [AWS] 기반 컨테이너 오케스트레이션 환경 설정 -2 k8s 컨텍스트 설정 & ECR 확인 - 실습
		클라우드 플랫폼	2 Days	[AWS] 이미지 저장소에 이미지 푸쉬해 보기 - 실습 [AWS] 기반 컨테이너 오케스트레이션 환경 설정 - EBS CSI Configuration - 실습 [Azure] CLI기반 Azure 및 쿠버네티스 Client 설치 및 소개 - 실습 [Azure] 기반 컨테이너 오케스트레이션 환경 설정 -1 (레이아웃, 리소스그룹 생성) - 실습 [Azure] 기반 컨테이너 오케스트레이션 환경 설정 -2 GUI 기반 (AKS, ACR) 생성 - 실습 [Azure] 기반 컨테이너 오케스트레이션 환경 설정 -3 (콘솔에서 연동하기) - 실습 [Azure] 이미지 저장소에 이미지 푸쉬해 보기 - 실습
	Week 20	쿠버네티스(Kubernetes) 이해	3 Days	- 쿠버네티스 기본 오브젝트 모델 - 내가 만든 서비스 쿠버네티스에 배포하고 조회하기 - 실습 - YAML기반으로 내 서비스 배포 및 업그레이드 하기 - 실습 - 배포된 마이크로서비스 삭제와 알아두면 유용한 명령어 소개 - 실습 - Pod Status에 따른 컨테이너 트러블 슈팅 - ImagePullBackOff, CrashLoopBack - 실습 - Pod Status에 따른 컨테이너 트러블 슈팅 - CrashLoopBack, (kubectl exec) - 실습 12st 어플리케이션 쿠버네티스에 배포하기-1 (주문서비스 Profile 확인 및 이미지생성) - 실습 12st 어플리케이션 쿠버네티스에 배포하기-2 (kafka 설치 12st 이미지 배포) - 실습 12st 어플리케이션 쿠버네티스에 배포하기-3 (서비스 생성 및 12st Mail 테스트) - 실습

커리큘럼 : LV3 Advanced

교육목표	경쟁력 있는 클라우드 애플리케이션 설계를 구현하고 실전 응용 가능한 클라우드 인프라를 이해하고 실습함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 20	마이크로서비스 SLA를 위한 쿠버네티스 Features	3 Days	- 레이블과 어노테이션 - 레이블을 활용한 쿠버네티스 객체 식별 - 실습 - 어노테이션을 활용한 마이크로서비스 롤백 응용 - 실습 - 리플리카셋 (ReplicaSet) 이론과 Manual Scale Out - 실습 - 오토 스케일 아웃(Auto Scale-out) 랩준비(Metric Server & Workload Generator) - 실습 - 주문 서비스에 오토 스케일 아웃(Auto Scale-out) 적용하기 - 실습 - Deployments 객체 종합 리뷰 - 네임스페이스 개념 및 기본 네임스페이스 응용 - 실습 - 마이크로서비스 로더밸런서 - Service 개념 - 마이크로서비스 로더밸런서(Service) - Access Scope : Type 정리 - 실습 - Service - ClusterIP 타입 생성과 컨테이너와의 관계 - 실습 - Service - ClusterIP, NodePort 타입으로 서비스 접근 - 실습 - Service - LoadBalancer 타입 및 서비스 이름으로 Web URL 활용 - 실습 - 셀프힐링(Liveness) & 무정지 배포(Rediness) - 실습 - 셀프힐링(Liveness) Exec Action 실습 - 실습 - 무정지 배포(Rediness) 실습을 위한 준비(kafka, Seige 설치) - 실습

커리큘럼 : LV3 Advanced

교육목표	경쟁력 있는 클라우드 애플리케이션 설계를 구현하고 실전 응용 가능한 클라우드 인프라를 이해하고 실습함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 21	마이크로서비스 SLA를 위한 쿠버네티스 Features	3 Days	- 무정지 배포(Rediness) 실습 - 실습 - 쿠버네티스 볼륨 인터페이스 - emptyDir, hostPath 유형 - 쿠버네티스 볼륨 인터페이스 - emptyDir, hostPath 실습 - 실습 [AWS] 쿠버네티스 볼륨 인터페이스 - EBS, EFS and PVC [AWS] 쿠버네티스 볼륨 인터페이스 - 관리콘솔을 통한 파일시스템 생성 - 실습 [AWS] 쿠버네티스 볼륨 인터페이스 - PVC기반 NFS - 실습 [Azure] 쿠버네티스 볼륨 인터페이스 - PVC기반 NFS - 실습 - 파일시스템 연결과 데이터베이스 설정 - 실습 - 쿠버네티스에 환경변수 구성하기 - 쿠버네티스에 환경변수(ConfigMap) 구성하기 실습 - 실습 - 쿠버네티스에 환경변수(Secret) 구성하기 실습 - 실습 - Ingress를 통한 진입점 통일 - Ingress를 통한 진입점 통일 실습 - 실습
		서비스 메시를 통한 Improved 오케스트레이션	2 Days	- 서비스 메시 정의 - 서비스 메시 아키텍처 및 메커니즘 이해 - 쿠버네티스에 서비스 메시(Istio) 설치 - 실습 - 마이크로서비스에 서비스 메시 적용하기 - 실습 - Istio를 통한 동적트래픽 라우팅 대상 배포 - 실습 - Istio를 통한 동적트래픽 라우팅 - 1. Kiali와 Jaeger를 통한 모니터링 - 실습 - Istio를 통한 동적트래픽 라우팅 - 2. 심플 라우팅 - 실습 - Istio를 통한 동적트래픽 라우팅 - 3. Advanced 라우팅 - 실습 - Istio를 통한 서비스 회복성 (Timeout & Retry) - 실습 - Istio를 통한 서비스 회복성 (Circuit Breaker) - 실습

커리큘럼 : LV3 Advanced

교육목표	경쟁력 있는 클라우드 애플리케이션 설계를 구현하고 실전 응용 가능한 클라우드 인프라를 이해하고 실습함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 22	Kubernetes와 메시를 활용한 통합 모니터링	2 Days	- Prometheus 설치와 PromQL 질의어 이해 - 실습 - Grafana를 활용한 대쉬보드 생성하기 - 실습 - Grafana 대쉬보드 커스터마이징 - 실습 - 이상감지를 위한 Alert 설정 (Slack 채널) - 실습 - 이상 상황 연출 및 Alert 수신하기 - 실습 - SRE(Site Reliability Engineering) 이란 - SRE(Site Reliability Engineering) 대쉬보드 실습 - 실습
		마이크로서비스 통합 로깅	1 Days	- 쿠버네티스에 통합로깅을 위한 EFK 스택 설치 - 실습 - 통합로깅 대상 서비스 배포 - 실습 - 클러스터에서 통합로깅 매카니즘 이해 - Kibana 접속을 통한 통합 로깅 확인하기 - 실습
		마이크로서비스 배포	2 Days	- 쿠버네티스 배포전략 - Argo Rollout과 Istio를 통한 카나리 배포 - 실습 - Argo CD를 활용해 카나리 배포해 보기 - 실습 - Argo CD를 활용한 카나리 배포 확장미션 - 실습 - A/B 테스트 방식을 통한 배포 실습 - 실습

커리큘럼 : LV3 Advanced

교육목표	경쟁력 있는 클라우드 애플리케이션 설계를 구현하고 실전 응용 가능한 클라우드 인프라를 이해하고 실습함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용	
	Week 23	파이프라인을 통한 배포 자동화 (벤더에 따라 상이)	3 Days	[AWS] Code Build Pipeline - 둘러보기 [AWS] Code Build Pipeline - 대상 서비스 준비 및 연동 - 실습 [AWS] Code Build Pipeline - Build Spec 이해 - 실습 [AWS] Code Build Pipeline - CI 환경설정 및 빌드 준비 - 실습 [AWS] Code Build Pipeline - Cache 적용 - 실습 [AWS] Code Build Pipeline - CD 환경설정 및 빌드 준비 - 실습 [AWS] Code Build Pipeline - CI/CD 테스트 - 실습 [AWS] Code Build Pipeline - 추가 기능 - 실습	[Azure] DevOps Portal - 둘러보기 [Azure] DevOps Portal - Azure Cloud Configuration - 실습 [Azure] DevOps Portal - 대상 서비스 준비 및 연동 - 실습 [Azure] DevOps Portal - CI 단계 Job 및 Task 생성 - 실습 [Azure] DevOps Portal - CI 빌드 준비 및 실행 - 실습 [Azure] DevOps Portal - CD 단계 Job 및 Task 생성 - 실습 [Azure] DevOps Portal - CD 빌드 준비 및 실행 - 실습 [Azure] DevOps Portal - CI/CD 폴 테스트 - 실습 [Azure] DevOps Portal - 추가 기능 - 실습
		Python 기초와 개발환경 설정	1 Day	- Python 프로그래밍 언어 소개 및 기초 문법 이해 - 개발 환경 설치 및 구성: Python 인터프리터, 텍스트 에디터, IDE 사용 - Python 가상 환경 설정 및 패키지 관리자 (pip) 사용법	
		Developing LLM Applications (Basics)	1 Day	- LLM Application Architecture - Prompt Engineering: CoT, ReAct, Plan-and-Solve - LLM Application 구현 고려사항: Context Window Limitations - LLM Frameworks: Langchain, Semantic Kernel	

커리큘럼 : LV3 Advanced

교육목표	경쟁력 있는 클라우드 애플리케이션 설계를 구현하고 실전 응용 가능한 클라우드 인프라를 이해하고 실습함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 24	Developing LLM Applications (Langchain)	2 Days	• 간단한 Tool 호출을 하는 챗봇 만들기 • LCEL (RunnablePassthrough, RunnableParallel, RunnableLambda) • Langsmith를 이용한 디버깅 • 문서기반 RAG 애플리케이션 만들기 (벡터 데이터베이스와 Pinecone을 기반한 RAG) • RDB를 기반한 RAG (Text2SQL)
		프라이빗 클라우드 환경 구축	2 Days	• 하드웨어 및 네트워크 구성 • OS 및 소프트웨어 환경 구성 • Docker & Kubernetes 설치 • DevOps 환경 구축(Backstage, GitLab, Argo CD, Kubersphere)
		폐쇄형 생성형 AI 환경 구축	1 Days	• 로컬 LLM 모델 개요 • LLM Inference • 모델 다운로드 및 적용, 배포 • AI 개발 환경 적용(폐쇄형 LLM 서버 연동)

커리큘럼 : LV3 Advanced

교육목표	경쟁력 있는 클라우드 애플리케이션 설계를 구현하고 실전 응용 가능한 클라우드 인프라를 이해하고 실습함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 25	서버리스 컴퓨팅 개론	1 Day	• 서버리스 컴퓨팅의 이해: 정의, 장점 및 사용 사례 소개 • 서버리스 아키텍처의 주요 구성 요소와 작동 원리 • 클라우드 서비스와 서버리스 컴퓨팅의 관계 이해
		KNative	2 Days	• KNative 개념 소개와 서버리스 컴퓨팅에서의 역할 • KNative 설치 및 기본 설정: 시작하기 위한 준비 • 간단한 서버리스 어플리케이션 구축을 위한 KNative 사용법 • KNative를 이용한 서비스 배포 및 관리 방법 • 트래픽 관리 및 오토 스케일링 설정 • KNative를 활용한 CI/CD 통합 실습
		Amazon Lambda	2 Days	• Amazon Lambda 서비스 소개 및 기본 개념 • Lambda 함수 생성, 설정 및 배포 과정 실습 • Amazon API Gateway와 Lambda를 통한 REST API 구축 실습

커리큘럼 : Capstone Project

교육목표	실제 학습한 내용을 Biz-Dev-Ops-2nd Iteration으로 실무 프로젝트를 진행하여 현업에 즉시 투입 가능한 수준의 Project를 생성함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 26	미니프로젝트	5 Days	• 미니 프로젝트 수행 • 클라우드 애플리케이션 설계 및 구현 • 필요에 따라 AI 연동을 통한 앱을 개발할 수 있음

클라우드와 도커를 활용한
효율적인 개발 환경 및 배포

- ☑ 도커를 이용한 어플리케이션 컨테이너화 및 이미지 관리
- ☑ AWS와 Azure 클라우드 서비스를 통한 컨테이너 오케스트레이션 환경 구축
- ☑ 쿠버네티스를 이용한 마이크로서비스 관리 및 무정지 배포(Rolling Update)

통합 모니터링 및 로깅을 통한
서비스 운영 최적화

- ☑ Prometheus와 Grafana를 이용한 메트릭스 기반 모니터링 시스템 구축
- ☑ EFK 스택을 활용한 통합 로깅 시스템 구축 및 관리
- ☑ Istio 서비스 메시를 통한 고급 오케스트레이션 기능 실습

파이프라인을 통한 지속적인
통합 및 배포 자동화

- ☑ AWS Code Build와 Azure DevOps를 활용한 CI/CD 파이프라인 구축
- ☑ Argo CD를 이용한 선언적 배포 및 A/B 테스트 방식의 배포 실습
- ☑ 쿠버네티스 아키텍처 이해 및 프라이빗 클라우드 환경 구축

보안과 개인 정보 보호를
중시하는 AI 개발 환경 설정

- ☑ 로컬 LLM 모델을 활용한 폐쇄형 생성형 AI 환경 구축 및 연동
- ☑ 데이터와 모델의 안전한 적용 및 배포 방법 학습

커리큘럼 : Capstone Project

교육목표	실제 학습한 내용을 Biz-Dev-Ops-2nd Iteration으로 실무 프로젝트를 진행하여 현업에 즉시 투입 가능한 수준의 Project를 생성함.				
교육대상	Business, Architect, Developer	교육기간	1일, 7시간 실습 (60%), 이론 (40%)	선수지식(수강 요건)	리눅스, 클라우드 인프라, 초급 자바스크립트/파이썬

교육내용	기간	주제	시간	주요 학습내용
	Week 27~29	Capstone Project	15 Days	• 캡스톤 프로젝트 평가 기준 및 레퍼런스 설명 • 베스트 과제 결과물 공유 - 음식배달앱, 숙소예약앱, 차량예약앱 등 • 팀별 과제 도출 및 DDD 모델 설계 - 실습 • 설계 결과 공유 - 팀별 - 실습 • 자체 구현 및 코칭 - 실습 • 평가