

# 교육 과정 소개서.

---

초격차 패키지: 9개 프로젝트로 경험하는 대용량 트래픽 & 데이터 처리



## 강의정보

강의장	온라인 강의   데스크탑, 노트북, 모바일 등
수강 기간	평생 소장
상세페이지	<a href="https://fastcampus.co.kr/dev_online_traffic_data">https://fastcampus.co.kr/dev_online_traffic_data</a>
강의시간	107시간 22분 38초
문의	<a href="#">고객센터</a>

## 강의특징

나만의 속도로 **낮이나 새벽이나 내가 원하는 시간대에** 나의 스케줄대로 수강

원하는 곳 어디서나 **시간을 쪼개 먼 거리를 오가며 오프라인 강의장을 찾을 필요 없이 어디서든 수강**

무제한 복습 **무엇이든 반복적으로 학습해야 내것이 되기에 이해가 안가는 구간 몇번이고 재생**



## 강의목표

- 네카라쿠배 같은 사용자가 많은 서비스 뿐만 아니라, 마케팅 이벤트를 앞둔 서비스, 비즈니스 데이터를 활용한 신규 기능을 개발하려는 서비스 등 3년차 이상의 백엔드 개발자라면, 대용량 처리 역량이 매우 중요합니다.
- 어떤 강의에서도 볼 수 없는 대용량 처리 경험을 쌓을 수 있는 총 9가지 초격차 프로젝트를 구성했습니다
- 실제 비즈니스에서 발생한 예기치 못한 트래픽에 자유자재로 대응할 수 있는 능력을 키우려면, 각 단계에서 사용할 수 있는 처리 방법을 모두 알고 있어야 합니다
- 대용량 트래픽 & 데이터 처리가 핵심이 되는 9가지 주제의 프로젝트

## 강의요약

- 성능 테스트 외에도 계획적으로 요구사항 정리부터 인터페이스 정의 및 스펙 작성까지 한 후 이를 토대로 구현을 진행하는 실무 프로세스에 익숙해질 수 있습니다
- 마케팅 혹은 이벤트로 인한 예상치 못한 트래픽부터, MAU가 늘어난 대규모 서비스까지! 일반적으로 경험하기 어려운 수준의 대용량 데이터와 트래픽을 총 8개 시나리오 경험해보세요
- 이번 프로젝트는 Kafka와 Redis, Cassandra 같은 최신 기술 스택을 사용하여 대용량 처리에 대한 경험을 쌓을 뿐만 아니라, 각 기술에 적용되는 설계 원칙을 알고, 다양한 소프트웨어를 조합해 여러 종류의 문제를 해결하는데서 오는 재미를 느낄 수 있기를 바랍니다.
- 카카오, 당근, N사, Gmarket, SK텔레콤 등 국내 IT 대기업의 최전선에서 트래픽과 데이터에 맞서고 계신 개발자 분들이 가진 경험을 녹여낸 프로젝트입니다



## 강사

이동준	과목	- Project 8. 광고 데이터 마이그레이션(CDC)
	약력	- 현) 카카오 발표] - 6억 건의 광고데이터 마이그레이션 / if(kakao) 2021 - 대량의 광고 데이터 필터 기능 개발기: ES한테 뽀맛고 Citus한테 안긴 썬 / if(kakao)2022
김병규	과목	- Project 7. 피쳐플래그
	약력	- 전) 당근마켓 - 전) 유안타증권 등
김한성	과목	- Project 6. Serverless (ML/프론트엔드 연동)
	약력	- 현) SK텔레콤
이형구	과목	- Project 2. 아이돌 티켓팅 접속자 대기열 시스템
	약력	- 현) 카카오모빌리티 발표] - JVM warm up / if(kakao)2022



## 강사

Alex	과목	- Project 1. 대규모 트래픽을 고려한 실무의 게시판
약력		<ul style="list-style-type: none"> <li>- 현) N사</li> <li>- 전) 카카오 계열사 교육</li> <li>- 한국공학대학교, 몰입캠프 특강</li> </ul>
JCH	과목	- Project 5. 블랙프라이데이 (EDA/분산처리)
약력		<ul style="list-style-type: none"> <li>- 전) 네카라쿠배 중 한 기업</li> <li>- 전) 엔씨 소프트</li> </ul>
안창현	과목	- Project 4. 최저가 서비스와 상품 추천 서비스
약력		<ul style="list-style-type: none"> <li>- 현) G* 국내 대기업(이커머스) Tech Leader</li> <li>- 전) S* 국내 대기업(제조업) Tech Leader</li> </ul>
개발자 J	과목	- Project 3. 네고왕 이벤트 선착순 쿠폰 시스템
약력		<ul style="list-style-type: none"> <li>- 현) 이커머스 플랫폼</li> <li>- 전) 엔씨 소프트</li> </ul>



CURRICULUM

01.

**대규모 트래픽을  
고려한 실무  
게시판 프로젝트**

파트별 수강시간 12:14:04

<b>CH01. 프로젝트 소개</b>
01. 소개 - 프로젝트 및 강사 소개
02. 소개 - 프로젝트 기획 설명 및 요구사항 정의
<b>CH02. 프로젝트 준비</b>
01. 프로젝트 준비 - 깃허브 연동
02. 프로젝트 준비 - 브랜치 전략
03. 프로젝트 준비 - 프로그램 설치
04. 프로젝트 준비 - 시퀀스 다이어그램 작성
05. 프로젝트 준비 - 시퀀스 다이어그램 실습 (1)
06. 프로젝트 준비 - 시퀀스 다이어그램 실습 (2)
07. 프로젝트 준비 - DB ERD 작성
08. 프로젝트 준비 - DB ERD 작성 실습
<b>CH03. issue1: 사용자 API 개발</b>
01. 프로젝트 개발 - 사용자 API 개발
02. 프로젝트 개발 - 사용자 API 개발 실습 (1)
03. 프로젝트 개발 - 사용자 API 개발 실습 (2)
04. 프로젝트 개발 - 사용자 API 개발 실습 (3)
<b>CH04. issue2: 로그인 기능에 AOP 적용</b>
01. 프로젝트 개발 - 로그인 기능에 AOP 적용
02. 프로젝트 개발 - 로그인 기능에 AOP 적용 실습
<b>CH05. issue3: 게시판 API 개발</b>
01. 프로젝트 개발 - 게시판 API 개발
02. 프로젝트 개발 - 게시판 API 개발 실습 (1)
03. 프로젝트 개발 - 게시판 API 개발 실습 (2)

## CURRICULUM

## 01.

# 대규모 트래픽을 고려한 실무 게시판 프로젝트

파트별 수강시간 12:14:04

## CH06. issue4: 검색 API 개발

01. 프로젝트 개발 - 검색 API 개발 실습
02. 프로젝트 개발 - 검색 API 개발 실습 (1)
03. 프로젝트 개발 - 검색 API 개발 실습 (2)

## CH07. issue5: 댓글, 태그 기능 개발

01. 프로젝트 개발 - 댓글, 태그 기능 개발
02. 프로젝트 개발 - 댓글, 태그 기능 개발 실습

## CH08. issue6: 성능 테스트 진행

01. 성능 테스트 - 이론
02. 성능 테스트 - 시나리오 작성
03. 성능 테스트 - 진행 및 결과 도출

## CH09. issue7: 성능 테스트 리팩토링

01. 성능테스트 - 리팩토링 이론
02. 성능테스트 - 리팩토링 (1) - 아키텍처 관점
03. 성능테스트 - 리팩토링 (2) - 어플리케이션 관점
04. 성능테스트 - 리팩토링 (3) - 데이터베이스 관점
05. 성능테스트 - 리팩토링 (4) - 성능 테스트 재진행 및 결과 도출

## CH10. issue8: 로깅, 예외처리 개발

01. 로깅, 예외 처리 - 이론
02. 로깅, 예외 처리 (1) - 로깅 레벨 설정
03. 로깅, 예외 처리 (2) - 실습
04. 로깅, 예외 처리 (3) - 데모 및 테스트



CURRICULUM

01.

대규모 트래픽을 고려한 실무 게시판 프로젝트

파트별 수강시간 12:14:04

<b>CH11. issue9: 알람 기능 개발</b>
01. 알람 기능 개발 - 이론
02. 알람 기능 개발 - AWS SNS 연동
03. 알람 기능 개발 - SLACK 알람 연동
<b>CH12. issue10: 배포 자동화</b>
01. 배포 자동화 - 이론
02. 배포 자동화 - 실습
03. 배포 자동화 - 실습 (2)
<b>CH13. 프로젝트 정리</b>
01. 프로젝트 정리 - 비즈니스 로직 정리
02. 프로젝트 정리 - 가독성 및 유지보수 관점 코드 정리
03. 프로젝트 정리 - 가독성 및 유지보수 관점 코드 정리 (2)
04. 프로젝트 정리 - 프레임워크 및 디자인 패턴 관점 정리
05. 프로젝트 정리 - 프레임워크 및 디자인 패턴 관점 정리 (2)
06. 프로젝트 정리 - 성능테스트 및 리팩토링 내용 정리
07. 프로젝트 정리 - 성능테스트 및 리팩토링 내용 정리 (2)
08. 프로젝트 정리 - 예외처리 및 로깅 관점 정리
09. 프로젝트 정리 - 예외처리 및 로깅 관점 정리 (2)
10. 프로젝트 정리 - 모니터링 및 알람 관점 정리
11. 프로젝트 정리 - 모니터링 및 알람 관점 정리 (2)
12. 프로젝트 정리 - 배포 자동화 정리
13. 프로젝트 정리 - 배포 자동화 정리 (2)
<b>CH14. 부록</b>
01. 성장하는 개발자 마음가짐 탐구
02. 기초 CS 공부를 해야 하는 이유
03. 연봉 협상 꿀팁
04. 기술 면접 잘보는 방법
05. 개발 블로그 작성하는 방법



## CURRICULUM

## 02.

# 아이돌 티켓팅 접속자 대기열 시스템

파트별 수강시간 16:34:43

CH01. 들어가며
01. 강의 소개
02. 개발 환경 준비
CH02. Redis
01. in-memory database
02. Reids 소개
03. Redis 설치
04. Redis CLI을 통한 접속
05. Data types에 대한 이해
06. Data types String 실습
07. Data types List, Set 실습
08. Data types Hash 실습
09. Data types Sorted Set 실습
10. Data types Geospatial 실습
11. Data types Bitmap 실습
12. Transactions
13. Keys, Scan 명령어
14. Cache 이론
15. Cache 실습
16. Spring Boot Cache
17. Spring Boot Session Store
18. Spring Boot Pub/Sub
19. Monitoring
20. Replication

## CURRICULUM

## 02.

# 아이돌 티켓팅 접속자 대기열 시스템

파트별 수강시간 16:34:43

<b>CH03. Webflux</b>
01. Spring Webflux 소개
02. CPU bound vs IO Bound
03. sync/async와 block/non-block
04. Spring MVC vs Webflux
05. Reactor 이론
06. Reactor 실습
07. Spring Webflux 실습(1)
08. Spring Webflux 실습(2)
09. R2DBC 이론
10. R2DBC 실습(1)
11. R2DBC 실습(2)
12. Reactive Redis 이론
13. Reactive Redis 실습
14. Spring MVC vs Webflux 성능
15. Blockhound
<b>CH04. 프로젝트 - 접속자대기 시스템</b>
01. 요구사항 분석
02. 아키텍처 설계
03. 개발 환경 준비
04. 대기열 등록 API 개발
05. 진입 요청 API 개발
06. 접속 대기 웹페이지 개발
07. 대기열 스케줄러 개발
08. 대기열 이탈
09. 테스트
10. 마무리

## CURRICULUM

## 03.

# 네고왕 이벤트 선착순 쿠폰 시스템

파트별 수강시간 08:40:18

<b>CH00. Intro</b>
01. 강의 개요
<b>CH01. 프로젝트 개요 및 요구사항 분석</b>
01. 프로젝트 개요 및 요구사항 분석
<b>CH02. 프로젝트 환경 설정</b>
01. 프로젝트 구성
02. mysql, redis 설정
03. 부하 테스트툴 설정
04. 부하 테스트 시도해보기
<b>CH03. 쿠폰 발급 관련 도메인 설계</b>
01. 쿠폰 발급 흐름 정리 및 엔티티 설계
02. 쿠폰 엔티티 작성
<b>CH04. Mysql기반 선착순 쿠폰 발급 기능 개발</b>
01. 쿠폰 엔티티 발급 기능
02. 쿠폰 발급 기능
03. 쿠폰 발급 API
04. 트래픽이 높아지는 경우 문제점 파악
05. 동시성 이슈 해결 - synchronized
06. 동시성 이슈 해결 - redis lock
07. 동시성 이슈 해결 - mysql lock
<b>CH05. Redis기반 선착순 쿠폰 발급 서버 설계</b>
01. Redis 데이터 구조 학습
02. 쿠폰 발급 서버 구조 학습
03. 적절한 Redis 데이터 구조 결정하기
<b>CH06. Redis기반 API 서버 구현</b>
01. 쿠폰 발급 요청 API 개발 - sorted set
02. 쿠폰 발급 요청 API 개발 - set
03. 쿠폰 발급 요청 API 개발 - 캐시 적용
04. 쿠폰 발급 요청 API 개발 - script
<b>CH07. Redis기반 쿠폰 발급 서버 구현</b>
01. 쿠폰 발급 서버 구현
02. 쿠폰 발급 기능 테스트 및 로컬 캐시 적용
<b>CH08. 배포 및 성능 테스트</b>
01. 모니터링 환경 구축
02. 서버 배포
03. 서버 모니터링 설정
04. 성능 테스트



CURRICULUM

04.

최저가 서비스와  
상품 추천 서비스

파트별 수강시간 12:30:00

<b>CH01. 이커머스 비즈니스의 이해</b>
01. 이커머스 비즈니스 타입 및 환경
02. 이커머스의 데이터, 트래픽 특징 1
02. 이커머스의 데이터, 트래픽 특징 2
03. 이커머스의 Key value 비즈니스
04. 이커머스내 구현되는 B/E 서비스 타입
05. 현업 이커머스 개발담당자 시각에서 본 Redis, Kafka Use case
<b>CH02. Redis 활용 서비스 설계</b>
01. Use Case에 기반한 서비스 요구사항 정의 for Redis
02. Use Case에 기반한 서비스 품질요건 정의 for Redis
03. 서비스 논리 설계 for Redis
<b>CH03. Redis의 기본개념</b>
01. redis 기본 설명 (개념, 장점, 단점, 목적)
02. [실습] 가격비교 Service에 사용될 redis 구조 (Master-Replica, Sentinel, Cluster)
03. redis 와 비슷한 다른 solution - memcache
<b>CH04. Redis 설치 및 Configuration</b>
01. [실습] 가격비교 API에 사용될 redis vm 생성, Redis 설치 (on EC2 of AWS)
02. [실습] 가격비교 API에 사용될 redis 주요 설정-1
03. [실습] 가격비교 API에 사용될 redis 주요 설정-2
<b>CH05. Redis 활용 서비스 개발</b>
01. [실습] 가격비교 REST API 기본틀 개발
02. [실습] 가격비교 API redis Library 종류와 선택
03. 가격정보 데이터의 dataType로 분할 필요
04. [실습] 가격정보 데이터의 dataType
05. [실습] redis sample data 생성과 입력, Key, Value Design
06. [실습] redis 관련 code 작성 및 배포 -1
07. [실습] redis 관련 code 작성 및 배포 -2
08. [실습] redis 관련 code 작성 및 배포 -3
09. [실습] redis 관련 code 작성 및 배포 -4
10. [실습] redis 관련 code 작성 및 배포 -5
11. [실습] redis 관련 code 작성 및 배포 -6
12. [실습] redis 관련 code 작성 및 배포 -7
13. [실습] redis 관련 code 작성 및 배포 -8



CURRICULUM

04.

최저가 서비스와  
상품 추천 서비스

파트별 수강시간 12:30:00

<b>CH06. Redis 서비스 Test</b>
01. [실습] redis Stress Test
<b>CH07. Redis 장애상황과 대응</b>
01. [실습] redis 장애 Case 및 대응
02. redis server command
03. redis advanced setting
<b>CH08. 추가활용 사례</b>
01. 사례 1 : redis 를 이용한 session 관리
02. 사례 2 : redis 를 이용한 배송서비스
<b>CH09. Kafka 활용 서비스 설계</b>
01. Use Case에 기반한 서비스 요구사항 정의 for Kafka
02. Use Case에 기반한 서비스 품질요건 정의 for Kafka
03. 서비스 논리 설계 for Kafka
<b>CH10. Kafka의 기본개념</b>
01. Kafka 기본 설명 (개념, 장점, 단점, 목적)
02. Zookeeper와 Kafka 역할
03. 가격비교 구매실적 Kakfa Cluster 구조
04. Partition, Replica, Offset, Consumer, Producer 개념
05. Kafka 와 비슷한 다른 Solution - rabbitMQ, redisQueue, 그 외 상용 툴
<b>CH11. Kafka 설치 및 Configuration</b>
01. [실습] 구매실적 Service 에서 사용될 Kafka vm. 생성, 설치
<b>CH12. Kafka 활용 서비스 개발</b>
01. [실습] Kakfa Produce, Consuming 간단개발
02. [실습] Kafka Stream을 활용한 KStream 개발-1
03. [실습] Kafka Stream을 활용한 KStream 개발-2
04. [실습] Kafka Stream을 활용한 KStream 개발-3
05. [실습] Kafka Stream을 활용한 Ktable 개발
06. [실습] 광고-구매 Join을 통한 광고효과 Streaming
07. [실습] 대량 Kafka Message Join Test
08. [실습] KStream Configuration
<b>CH13. Kafka 서비스 Test</b>
01. Kafka Stress Test
<b>CH14. Kafka 장애상황 대응과 운영</b>
01. [실습] 장애 Case 및 대응
<b>CH15. Kafka 추가활용 사례</b>
01. 사례1 : Kafka를 이용한 Dynamic Customer Chatting
02. 사례 2 : Kafka를 이용한 Big Data 적재



CURRICULUM

05.

**블랙프라이데이  
(EDA/분산처리)**

파트별 수강시간 11:06:34

<b>CH00. Intro</b>
01. 강의 및 강사 소개
<b>CH01. 대용량 트래픽에 대처하는 전략</b>
01. 서비스의 확장성이란?
02. 확장성을 위한 복제와 분산
03. 남은 문제들과 해결책
<b>CH02. EDA(Event Driven Architecture)의 소개</b>
01. EDA의 개념과 범위
02. EDA의 장단점과 사용 시나리오
03. EDA를 구성하기 위해 필요한 요소들
<b>CH03. 실습 프로젝트 요구사항 분석 및 설계 (블랙 프라이데이의 대용량 트래픽 주문 처리)</b>
01. 프로젝트 요구사항 분석 및 정리
02. 상위 수준 설계(주요 컴포넌트)
03. 하위 수준 설계(각 컴포넌트별 인터페이스)
04. 아키텍처 고려사항을 설계에 적용하기
<b>CH04. 기반 기술 학습(Kafka, Cassandra)</b>
01. 메시지 브로커란?
02. Kafka 둘러보기
03. 관계형 DB와 NoSQL
04. Cassandra 둘러보기
<b>CH05. 개발환경 구성하기</b>
01. IntelliJ로 개발 환경 구성
02. Docker와 Docker Compose를 이용해 테스트 환경 구성
03. Kafka 설치 및 실습
04. Cassandra 설치 및 실습

## CURRICULUM

## 05.

블랙프라이데이  
(EDA/분산처리)

파트별 수강시간 11:06:34

**CH06. 블랙 프라이데이 프로젝트 코드 구현**

01. 회원 서비스 개발 - REST API와 JPA 구현
02. 결제 서비스 개발 - 어댑터 패턴을 이용한 외부 인터페이스 연동
03. 배송 서비스 개발 - 스케줄링 잡을 통한 상태 업데이트 구현
04. 검색 서비스 개발 - 캐싱을 사용한 성능 향상
05. 카탈로그 서비스 개발 - NoSQL을 이용한 상품 정보 관리
06. 주문 서비스 개발 - 외부 저장소를 통한 작업 세션 관리
07. Docker Compose 환경에서 Postman을 이용한 서비스 연동 테스트

**CH07. Event를 이용한 리팩토링**

01. 이벤트 구조 정의 및 작성
02. Kafka를 이용한 카탈로그 변경 이벤트의 발행과 적용
03. 주문 과정을 이벤트 드리븐으로 재구성

**CH08. 테스트 전략과 테스트 코드 작성**

01. 소프트웨어 테스트 전략과 도구(JUnit, Mockito)
02. Junit과 Mockito를 이용해 테스트 코드 작성
03. 테스트 결과 확인 및 개선

**CH09. 성능 테스트 및 분석**

01. 성능 테스트의 개념 및 방법
02. 성능 테스트 도구 JMeter 사용해보기
03. JMeter를 이용해 부하 테스트 작성하고 성능 측정하기

## CURRICULUM

## 06.

# 대용량 트래픽을 견딜 수 있는 서버리스 아키텍처의 쇼핑몰

파트별 수강시간 16:33:01

<b>CH01. 강의소개</b>
01. 강의 및 강사 소개
<b>CH02. 요구사항 분석</b>
01. 쇼핑몰 서비스 소개 및 요건 분석
<b>CH03. 아키텍처 설계</b>
01. Monolithic Architecture란?
02. 일반적인 Monolithic Architecture 설계
03. Monolithic Architecture 방식의 장/단점
04. Serverless 란?
05. Serverless 방식의 장/단점
06. AWS 소개 및 기본적인 개념 설명
07. Lambda Architecture 소개 및 설계
<b>CH04. 개발 환경 구성</b>
01. 실습시 사용할 Github Repository 안내
02. IntelliJ 설치 및 설정
03. VPC 란?
04. IAM Role & Policy 란?
05. AWS 계정 및 IAM User 생성
06. Terraform 설치 및 설정
07. Terraform 동작 방식 설명 및 리소스 생성
08. code-server 설치 및 설정
09. SSH Tunneling 구성
<b>CH05. spring boot 프로젝트 생성</b>
01. 기본적인 REST API 설계
02. spring boot 프로젝트 생성
03. spring boot 프로젝트 빌드 및 배포



## CURRICULUM

## 06.

# 대용량 트래픽을 견딜 수 있는 서버리스 아키텍처의 쇼핑몰

파트별 수강시간 16:33:01

## CH06. REST API 개발(기본 기능)

01. 구현할 기능에 대한 설명
02. RDS란?
03. JPA를 통한 RDS Table 생성
04. Cognito 란?
05. 회원가입 API 추가
06. Cognito 를 통한 로그인 로그아웃 기능 추가
07. S3 란?
08. 상품 등록 조회 API 추가
09. 상품 구매 API 추가
10. DynamoDB 란
11. DynamoDB Table 를 통한 구매 이력 기능 추가
12. 비용 추정 및 계산

## CH07. 상품 등록 API 고도화1(검색엔진, 알림)

01. 구현할 기능에 대한 설명
02. OpenSearch, SQS, SES 생성(by Terraform)
03. SQS란?
04. Lambda + SQS 이벤트 연동
05. OpenSearch란?
06. SES란?
07. 상품 등록시 메일 알림 발송하는 기능 추가
08. 상품 등록시 검색엔진에 상품 정보 업데이트 기능 추가
09. 비용 추정 및 계산

## CH08. 상품 등록 API 고도화2(썸네일 이미지, CDN)

01. 구현할 기능에 대한 설명
02. Lambda + S3 이벤트 연동
03. CloudFront 란 ?
04. CloudFront + S3 연동
05. 비용 추정 및 계산

## CH09. front-end 배포 및 연동

01. React 프로젝트 설명
02. CloudFront vs S3 정적 웹 사이트 호스팅 차이
03. 비용 추정 및 계산



CURRICULUM

06.

**대용량 트래픽을 견딜 수 있는 서버리스 아키텍처의 쇼핑몰**

파트별 수강시간 16:33:01

<b>CH10. 추천 상품 리스트 API 추가</b>
01. 구현할 기능에 대한 설명
02. 추천 상품 리스트 생성용 Lambda 생성
03. Java Lambda에 invoke 권한 및 기능 추가
04. Lambda에서 Cold Start 문제 설명
05. Lambda Warm-up 방식 설명
06. Lambda Provisioned Concurrency 기능 설명 및 설정
07. Lambda Snapstart 기능 설명 및 설정
08. 비용 추정 및 계산
<b>CH11. 시스템 모니터링용 알림 시스템 구성</b>
01. 구현할 기능에 대한 설명
02. CloudWatch Alarm 생성 및 SNS 연동
03. CloudWatch 통합 Dashboard 생성
04. 비용 추정 및 계산
<b>CH12. 부가적인 설명</b>
01. MSA 기본 개념
02. 구축된 Architecture와 비교
03. auto-scaling 장/단점
04. 추가로 확장될 수 있는 요소(기능, 보안)
05. 구축된 Architecture의 한계점
06. 예상되는 트래픽에 대한 비용 추정
07. 예상되는 트래픽에 대한 실제 부하테스트
08. 생성된 AWS 리소스 삭제
<b>CH13. 실습 비용 관리 가이드</b>
01. 주로 많이 사용되는 AWS 리소스 삭제 가이드
02. 리소스 삭제시 비용이 발생할 수 있는 포인트
03. 간단한 계정 보안 가이드
04. 계정 비용 관리 및 모니터링 가이드
05. 차수별 오픈에 따른 중단/재개 가이드



CURRICULUM

07.

피쳐플래그

파트별 수강시간 06:39:52

<b>CH01. 피쳐 플래그란?</b>
01. 피쳐 플래그의 목적과 효용성
02. 피쳐 플래그 레퍼런스와 인터페이스 (Open Feature Flag)
03. 피쳐 플래그와 Remote Configuration 의 차이
<b>CH02. 피쳐 플래그가 필요한 문제 정의와 의사결정</b>
01. 피쳐 플래그가 필요한 상황인지 판단하기
02. 집중할 문제가 어떤 것인지에 대해 정의하기
03. 캐싱 - 어떤 캐싱 정책알고리즘이 좋을까
<b>CH03. 피쳐플래그 플랫폼 설계하기 1</b>
01. 기술 스택 결정하기
02. 논리적 플랫폼 아키텍처 설계하기
<b>CH04. 피쳐플래그 플랫폼 설계하기 2</b>
01. 물리적 플랫폼 아키텍처 상세 설계 (w/Open Featureflag)
<b>CH05. 실습 1: 프로젝트 세팅</b>
01. Hello World! 서버 구현 및 Github 연동하기
02. ResolveFeatureflag API Skeleton 서버 구현
<b>CH06. 실습 2: Openfeatureflag Provider 개발</b>
01. Openfeatrueflag Provider 개발 (w flagd)
02. Featureflag 플랫폼과 Provider 인테그레이션
<b>CH07. 피쳐 플래그 플랫폼 검증 플래닝</b>
01. 검증해야 할 대상과 변인 정의
02. 각 상황 별 예상되는 결과 도출하기
<b>CH08. 실습 3: 검증을 위한 환경 만들기</b>
01. 모니터링을 위한 인프라 구축하기 (Prometheus, Grafana)
02. 검증을 위한 프로그램 개발하기
<b>CH09. 실습 4: 캐싱 알고리즘 변경 및 실험과 검증</b>
01. 캐싱 알고리즘 변경 및 결과 재검증
02. 상황별 효과적인 캐싱 알고리즘 실측
<b>CH10. 프로젝트 회고</b>
01. featrueflag 플랫폼이 나아가야할 방향



CURRICULUM

08.

**대용량 광고  
데이터  
마이그레이션  
프로젝트**

파트별 수강시간 15:57:21

<b>CH01. 개요</b>
01. 강사 및 강의 소개
<b>CH02. 디지털 광고의 이해</b>
01. 광고 기초
02. 디지털 광고 기술의 역사
03. 디지털 광고 생태계의 주요 기술 플랫폼 및 중개자
04. 주요 디지털 광고 매체 및 채널
<b>CH03. CDC(Change Data Capture)의 이해</b>
01. CDC의 정의
02. CDC 작동 방식
03. CDC의 이점
04. CDC 구현 방식
<b>CH04. 프로젝트 준비</b>
01. 프로젝트 시나리오
02. 요구사항 분석과 아키텍처 설계
03. 개발환경 구성
<b>CH05. 레거시 DSP 개발</b>
01. 레거시 DSP 모듈 추가
02. MySQL DB 연결
03. 광고 도메인 엔티티 구현
04. 광고 데이터 CRUD api 구현
<b>CH06. 레거시 DSP에 CDC 적용</b>
01. Domain event 추가
02. RabbitMQ에 메시지 전송
<b>CH07. 점진적 마이그레이션 구현</b>
01. 점진적 마이그레이션 애플리케이션 모듈 추가
02. 이중 DB 연결 (MySQL, PostgreSQL)
03. RabbitMQ Message Consumer 구현
04. 광고 도메인별 단건 마이그레이션 로직 구현



CURRICULUM

08.

**대용량 광고  
데이터  
마이그레이션  
프로젝트**

파트별 수강시간 15:57:21

<b>CH08. 마이그레이션 서비스 모듈 분리</b>
01. 마이그레이션 서비스 모듈 분리
<b>CH09. 마이그레이션 internal API 구현</b>
01. 마이그레이션 internal API 모듈 추가
02. 마이그레이션 동의 API 구현
03. 동의를 광고주만 점진적 마이그레이션 진행되도록 수정
04. 단건 재처리 API 구현
05. Kafka로 마이그레이션 동의 메시지 발행
<b>CH10. 배치성 마이그레이션 구현</b>
01. 배치성 마이그레이션 구현
02. Kafka Message consumer 구현
03. 사용자별 마이그레이션 단계 진행 이벤트 처리로직 구현
04. 도메인별 페이징 마이그레이션 이벤트 처리 로직 구현
05. 배치 재처리 API 구현
06. 마이그레이션 진행상황 확인 API 구현
<b>CH11. 마이그레이션 모니터링 및 성능 튜닝</b>
01. 레거시 DSP에 성능테스트용 더미 광고데이터 생성
02. 모니터링 환경 설정
03. 배치성 마이그레이션 시뮬레이션
04. Kafka Message Consumer 비동기 처리
05. Batch Insert 설정
06. 배치성 마이그레이션 로직 수정
07. 점진적 마이그레이션과 배치성 마이그레이션 동시에 시뮬레이션
<b>CH12. 프로젝트 정리 및 회고</b>
01. 프로젝트 정리 및 회고
<b>CH13. 실서비스 마이그레이션 진행 후기</b>
01. 6억 건의 광고데이터 마이그레이션 진행 후기
<b>CH14. 강의 정리 및 회고</b>
01. 강의 정리 및 회고



CURRICULUM

09.

**쿠버네티스 활용  
MSA 기반 SNS  
프로젝트**

파트별 수강시간 07:09:45

<b>CH01. 프로젝트 개요 및 전체 구조</b>
01. Kubernetes를 이용한 MSA 기반 SNS 백엔드 개발
02. 프로젝트 개발을 위한 인프라 설정 개요
03. AWS와 EKS를 이용한 쿠버네티스 클러스터 설정
04. EFS를 이용한 StorageClass 정의
05. ECR 구성 및 MySQL, Redis, Kafka 설치
<b>CH02. Social Feed 서버 개발</b>
01. 스프링 프로젝트 구성
02. Kubernetes를 위한 기본 스프링 설정 및 EKS 배포
03. Social Feed 기능 개발
04. Telepresence를 이용한 마이크로서비스 개발환경 구성
<b>CH03. User 서버 개발</b>
01. 사용자 가입, 로그인 기능 개발
02. Follow, Unfollow 기능 개발
03. Social Feed 서버와 User 서버 연계
04. 배포한 Pod를 무중단으로 업데이트 하기
<b>CH04. Image 서버 개발</b>
01. 이미지 서버를 위한 볼륨 구성
02. 사진 업로드, 리사이즈 및 조회 기능 개발
<b>CH05. Notification Batch 개발</b>
01. Follower 메일 알림 기능 개발
02. Job과 CronJob을 활용한 배치 수행
<b>CH06. 스케일 가능한 서비스 환경 구성과 성능 테스트</b>
01. Prometheus와 Grafana를 이용한 서비스 모니터링
02. OpenLens와 Kubecost를 이용한 서비스 관리
03. k6를 이용한 애플리케이션 성능 테스트와 Pod 스케일링
<b>CH07. Timeline 서버 개발</b>
01. Kafka를 활용한 애플리케이션 성능 개선
02. 타임라인 API 만들기
03. 프론트엔드 배포 및 Ingress를 통한 API 공개
<b>CH08. 서비스 배포와 운영</b>
01. Helm을 이용한 서비스 배포와 환경 구분

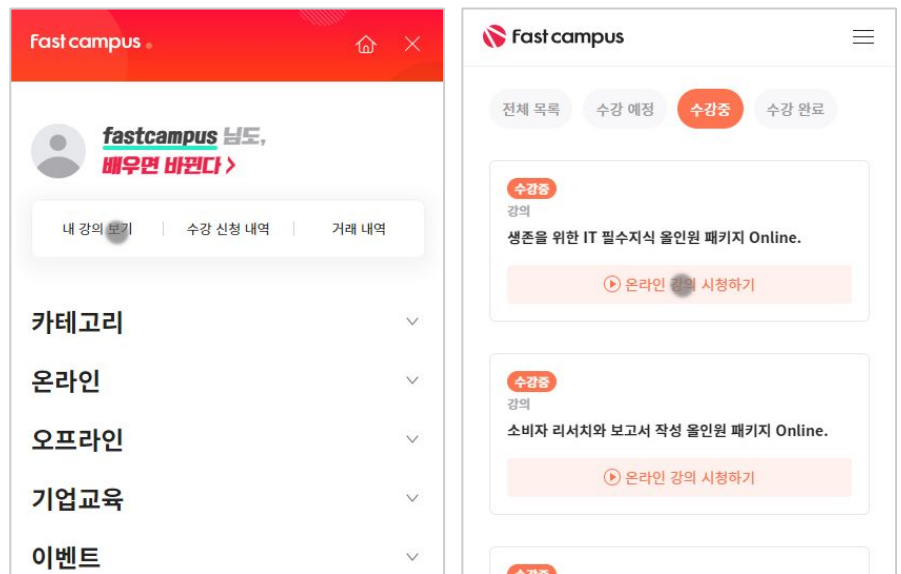


## 주의 사항

- 상황에 따라 사전 공지 없이 할인이 조기 마감되거나 연장될 수 있습니다.
- 패스트캠퍼스의 모든 온라인 강의는 아이디 공유를 금지하고 있으며 1개의 아이디로 여러 명이 수강하실 수 없습니다.
- 별도의 주의사항은 각 강의 상세페이지에서 확인하실 수 있습니다.

## 수강 방법

- 패스트캠퍼스는 크롬 브라우저에 최적화 되어있습니다.
- 사전 예약 판매 중인 강의의 경우 1차 공개일정에 맞춰 '온라인 강의 시청하기'가 활성화됩니다.



## 환불 규정

- 온라인 강의는 각 과정 별 '정상 수강기간(유료수강기간)'과 정상 수강기간 이후의 '복습 수강기간(무료수강기간)'으로 구성됩니다.
- 환불금액은 실제 결제금액을 기준으로 계산됩니다.

수강 시작 후 7일 이내	100% 환불 가능 (단, 수강하셨다면 수강 분량만큼 차감)
수강 시작 후 7일 경과	정상(유료) 수강기간 대비 잔여일에 대해 환불규정에 따라 환불 가능

※ 강의별 환불규정이 상이할 수 있으므로 각 강의 상세페이지를 확인해 주세요.