

# 교육 과정 소개서.

---

76가지 예제로 완벽 대비하는 AWS 기반 DevOps 실전



## 강의정보

강의장	온라인 강의   데스크탑, 노트북, 모바일 등
수강 기간	평생 소장
상세페이지	<a href="https://fastcampus.co.kr/dev_online_100devops">https://fastcampus.co.kr/dev_online_100devops</a>
강의시간	75시간 22분
문의	<a href="#">고객센터</a>

## 강의특징

나만의 속도로	낮이나 새벽이나 <b>내가 원하는 시간대</b> 에 나의 스케줄대로 수강
------------	---

원하는 곳 어디서나	시간을 쪼개 먼 거리를 오가며 오프라인 강의장을 찾을 필요 없이 <b>어디서든 수강</b>
---------------	---

무제한 복습	무엇이든 반복적으로 학습해야 내것이 되기에 이해가 안가는 구간 <b>몇번이고 재생</b>
-----------	--



## 강의목표

- 76가지 실무 초밀착 실습 주니어-미드레벨 채용 공고 속 공통 우대사항 & 자격 요건을 76가지 예제로 한 번에! DevOps 업무 전 범위를 완벽 커버하세요.
- 대표 스택 37개를 활용한 환경 구축부터, 더 효율적인 스택 활용 방법과 장애 대응까지 모두 익혀봅시다
- 수십 건의 IT 대기업 채용 공고에서 공통적으로 등장하는 '자격 요건'과 '우대 사항'. 파이프라인 구축과 운영 등 기본적인 업무부터 장애 대응, 인프라 설계 등 심화 업무까지 배워봅시다.
- 현업의 표준 아키텍처 4가지 위에서 실습하는 76가지 기초-심화 예제로 알려드립니다.

## 강의요약

- 실제 현업과 유사하게 구축한 환경에서 예제를 구현하고, 마지막으로 복습 & 응용을 위한 과제까지 3단계 반복 학습법으로 완전하게 DevOps의 실무를 학습해보세요.
- 수백 건의 채용 공고, 그리고 IT 대기업 출신 현직 DevOps 엔지니어가 엄선한 100가지 예제를 현업의 표준 아키텍처 4가지 위에서 실습해보세요. 자주 쓰이는 아키텍처를 직접 구축하고 운영, 유지보수하며 DevOps 실무를 99% 유사하게 경험할 수 있습니다
- 5년차 이상 DevOps 엔지니어들이 꼽은 실무 단골 상황! 가상의 상황을 가정하여 하나의 큰 목표(대용량 트래픽 처리, 빌드 속도 개선 등)를 위해 환경을 구축하고 운영해보세요.
- 각 환경 위에서 현업에서 가장 많이 쓰는 DevOps 대표 스택 37개를 활용해 직접 구축해본 다음, 각 스택에서 발생하는 오류를 잡고, 더 효율적으로 쓰는 방법까지 익혀보세요.



## 강사

오성근	과목	- Part 1
약력		<ul style="list-style-type: none"> <li>- 현) 대기업 S사 IT Infra 엔지니어</li> <li>- 현) 쿠버네티스 강의 다수</li> </ul>
정현석	과목	- Part 2
약력		<ul style="list-style-type: none"> <li>- 현) 메디트 DevOps Developer</li> <li>- 전) SK C&amp;C Cloud Architect</li> <li>- 전) 메가존클라우드 Solutions Architect</li> </ul>
김용현	과목	- Part 3
약력		<ul style="list-style-type: none"> <li>- 현) 스타트업 DevOps Engineer</li> <li>- 전) 레저여행 어플 Y사 DevOps Engineer</li> </ul>

## CURRICULUM

## 01.

## 선수 지식

파트별 수강시간 21:25:53

<b>CH01. Terraform</b>
01. Terraform 소개
02. Terraform 장/단점
03. HCL 문법 및 formatter 소개
04. resource 와 data
05. module
06. variable & local & output
07. count & for each & for
08. 조건문(condition)
09. 상태 저장소(backend)
10. 상태 관리
11. 리소스 강제 교체(taint/untaint)
12. Provisioner
13. 워크스페이스
14. Terraform Cloud 사용
15. Terraform 외부 module 소개 및 사용
16. Terraform 운영 전략(Backend, Workspace, Directory, Module)
<b>CH02. Docker 개요</b>
01. 강사 소개 및 강의 목표
02. container란?
03. docker란?
04. docker 설치 및 간단 사용 실습
<b>CH03. Kubernetes 시작하기</b>
01. kubernetes_구조
02. kubernetes _설치_실습
03. kube-apiserver
04. etcd
05. kube-scheduler
06. controller-manager
07. kubelet
08. kube-proxy
09. container-runtime
10. coredns
11. 다시_전체보기



CURRICULUM

01.

선수 지식

파트별 수강시간 21:25:53

<b>CH04. AWS EKS 알아보기</b>
01. AWS EKS란?
02. kubernetes와의 차이점(장점)
03. EKS를 생성하는 방법
04. Terraform EKS module로 클러스터 생성하기
05. EKS클러스터 접근
<b>CH05. Kubernetes 오브젝트</b>
01. kubernetes 오브젝트
02. node
03. namespace
04. pod
05. replicaset
06. deployment
07. daemonset
08. service
09. ingress
10. CA
11. metricsserver_hpa_vpa
12. 환경 변수
13. configmap
14. secret
15. volume
16. 클러스터 제어 관련 오브젝트(RBAC, role, · serviceaccount, netowrk policy)
<b>CH06. Kubernetes 관련 도구</b>
01. k9s
02. lens
03. kubectx_kubens
04. helm
05. karpenter
06. ECR
07. kubecost
08. 그외의 다양한 예시들

## CURRICULUM

## 01.

## 선수 지식

파트별 수강시간 21:25:53

<b>CH07. AWS EKS로 어플리케이션 구축</b>
01. 클러스터 추가 설정
02. 샘플 어플리케이션 코드
03. 이미지빌드 및 푸시
04. manifest 작성
05. 서비스 기동 및 확인
<b>CH08. AWS EKS로 어플리케이션 운영</b>
01. helm chart 구성
02. githubAction으로 빌드구성
03. argocd 구성
04. 버전 업데이트 테스트
05. 서비스 스케일링
06. probe 추가
07. pod termination 정책의 이해
08. spot fargate로 비용 아끼기
09. fluentbit 구성하여 서버 로그 확인
10. EKS의 튜닝 포인트
11. kubectl 변형 jsonpath
12. 강의 마무리
<b>CH09. Github Actions</b>
01. Github Actions 소개
02. Github Actions 실습 내용 소개
03. Github Actions 실습(+Slack 연동)
<b>CH10. ArgoCD</b>
01. ArgoCD 소개
02. ArgoCD 설치
03. ArgoCD 설정 - declarative-setup
04. ArgoCD 설정 - app-of-apps
05. ArgoCD 인증
06. 실습에 필요한 인프라 구축(AWS CodeSeries + EKS)
07. ArgoCD CI/CD 실습

## CURRICULUM

## Part 1-1.

# 컨테이너 오케스트레이션 기초

파트별 수강시간 08:26:20

<b>CH00. OT &amp; 실습 환경 세팅</b>
01. 오리엔테이션
02. EKS 클러스터 구성(1)
03. EKS 클러스터 구성(2)
04. EKS 클러스터 구성(3)
<b>CH01. Docker 컨테이너 네트워킹</b>
01. 컨테이너 네트워크 구성
02. Pod 간 네트워크 연결 알아보기
03. Node 부터의 네트워크 알아보기
<b>CH02. Kubernetes 클러스터 내부 DNS 설정</b>
01. DNS 설정과 쿠버네티스의 DNS
02. 서비스와 DNS (1)
03. 서비스와 DNS (2)
04. External Name
<b>CH03. 복제 컨트롤러 사용</b>
01. Pod, Deployment, ReplicaSet
02. Rolling Update, NameSpace
03. Static Pod, MultiContainer Pod
<b>CH04. ConfigMap 및 Secret 사용</b>
01. 볼륨 개요
02. PV, PVC, Configmap (1)
03. PV, PVC, Configmap (2)
04. Secret 활용
<b>CH05. StatefulSet 구성</b>
01. Deployment vs StatefulSet
02. StatefulSet과 Headless Service
03. DB 활용 실습



CURRICULUM

Part 1-1.

컨테이너  
오케스트레이션  
기초

파트별 수강시간 08:26:20

<b>CH06. 스케줄링 지정 및 DaemonSet 구현</b>
01. Kubernetes 스케줄링 소개
02. Node Selector, Affinity, Taint, Toleration
03. DaemonSet, Cordon, Drain
04. Daemon Set
<b>CH07. Ingress 컨트롤러 설정</b>
01. Service 및 Ingress 소개
02. Service 및 Ingress - Instance 모드
03. Ingress - IP 모드
<b>CH08. 리소스 쿼터 설정</b>
01. Request, Limit 설정
02. Limit Range 설정
<b>CH09. 라이브니스 및 레디니스 프로브 구성</b>
01. 프로브 소개
02. 라이브니스 프로브 설정
03. 레디니스, 스타트업 프로브 설정
<b>CH10. Job 및 CronJob 구현</b>
01. Pod 및 Job의 차이 확인
02. Crontab
03. Cronjob
<b>CH11. AutoScale 구현</b>
01. Top 및 메트릭 모니터링 구성
02. AutoScale 구현 (HPA)
03. AutoScale 구현 (VPA)
<b>CH12. [미션] 쇼핑물을 쿠버네티스 기반으로 생성</b>
01. 쿠버네티스 기반 쇼핑물 생성
02. 생성된 쇼핑물에 접속 확인

## CURRICULUM

## Part 1-2.

## 기본 모니터링 및 로깅

파트별 수강시간 03:48:52

<b>CH01. 애플리케이션 로그 설정</b>
01. 애플리케이션 로그 설정
02. 로그 경로 안내
03. 로그 경로 실습
<b>CH02. 로그 레벨 관리</b>
01. 로그 레벨 관리
<b>CH03. 로그 회전 설정</b>
01. 로그 회전 설정
<b>CH04. 로그 중앙 집중화</b>
01. Elastic Search 소개
02. Elastic Search 활용
<b>CH05. 애플리케이션 성능 모니터링</b>
01. Prometheus&Grafana
02. Prometheus&Grafana 실습
<b>CH06. 기본 시스템 모니터링</b>
01. 기본 시스템 모니터링
<b>CH07. 시스템 로그 분석</b>
01. 시스템 로그 분석
<b>CH08. 네트워크 모니터링</b>
01. 네트워크 모니터링
02. 네트워크 모니터링 실습 (1)
03. 네트워크 모니터링 실습 (2)
<b>CH09. 로그 시각화</b>
01. 로그 시각화 툴 소개
<b>CH10. 알림 및 경고 설정</b>
01. 알림 및 경고 설정
<b>CH11. [미션] 쇼핑몰의 모니터링 시스템을 구현</b>
01. 메트릭 기반 모니터링 시스템 구현
02. 로그 기반 모니터링 시스템 구현



CURRICULUM

Part 1-3.

자동화 스크립트  
작성

파트별 수강시간 02:55:06

<b>CH01. 기본 배치 스크립트 작성</b>
01. 기본 배치 스크립트 작성
02. 스크립트 기본 소개
<b>CH02. 서버 상태 체크 스크립트</b>
01. 서버 상태 체크 스크립트
02. 서버 상태 체크 스크립트 확인
<b>CH03. 간단한 배포 스크립트</b>
01. 간단한 배포 스크립트
<b>CH04. 데이터 백업 스크립트</b>
01. 데이터 백업 스크립트
<b>CH05. 로그 파일 분석 스크립트</b>
01. 로그 파일 분석 스크립트
<b>CH06. 사용자 관리 스크립트</b>
01. 사용자 관리 스크립트
02. 사용자 관리 스크립트 실습
<b>CH07. 시스템 모니터링 스크립트</b>
01. 시스템 모니터링 스크립트
02. 시스템 모니터링 스크립트 실습
<b>CH08. 네트워크 트래픽 모니터링 스크립트</b>
01. 네트워크 트래픽 모니터링 스크립트
<b>CH09. 시스템 보안 체크 스크립트</b>
01. 시스템 보안 체크 스크립트
02. 시스템 보안 체크 스크립트 실습
<b>CH10. 서비스 상태 모니터링 스크립트</b>
01. 서비스 상태 모니터링 스크립트
<b>CH11. [미션] 쇼핑몰의 노드 추가시 사용할 스크립트 생성</b>
01. 쿠버네티스 노드 추가시 필요한 명령어 확인
02. 증설된 노드에서 kubectl 명령어 실행가능토록 스크립트 생성

## CURRICULUM

## Part 2-1.

CI/CD 안정적  
운영

파트별 수강시간 03:07:53

## CH01. 자동화된 배포 환경 만들어 보기(Github Action)

01. 자동화된 배포 스크립트 만들기
02. 배포 스크립트 결과물(Artifact) 만들기
03. CI/CD 배포 Task 만들기
04. ECS 환경을 위한 배포파이프라인 구성하기
05. 배포스크립트 빠르게 동작 시키기

## CH02. 배포 후 모니터링과 문제점 원인 파악해보기

01. 배포 스크립트는 정상적으로 동작했으나 문제가 될 경우
02. 배포 마무리 전에 스트레스 테스트 진행해보기

## CH03. [미션] CI/CD 배포 파이프라인에 스트레스 테스트 자동화 하기

01. [미션] CI/CD 배포 파이프라인에 스트레스 테스트 자동화 하기



CURRICULUM

Part 2-2.

운영환경고도화

파트별 수강시간 01:57:59

<b>CH01. 새로운취약점탐지와대응</b>
01. 새로운취약점 탐지와 그에 따른 대응하기
<b>CH02. 로깅 시스템 점검 및 대응</b>
01. 로깅 시스템을 점검하고 설정해보기
<b>CH03. 컴퓨팅 환경에 따른 로깅 시스템 변경</b>
01. 컴퓨팅 환경에 따른 로깅 시스템 변경을 해보자
<b>CH04. 안정적인 배포를 위한 배포 전략 설정</b>
01. 다양한 배포 전략에 대해 고민하는 DevOps 엔지니어들
02. ECS 환경에서의 Blue/Green 배포 전략 설정하기
03. RDS 업데이트 진행 시 중단 시간 줄이기
<b>CH05. 안정적인 운영을 위해 고려해야 하는 설정들</b>
01. 서버 업데이트에 대해 대응하는 방법(ECS)
02. 예상치 못한 부하 공격에 대해 빠르게 대응하는 방법
<b>CH06. [미션] 시스템에 블루/그린 배포 환경 적용하고 모니터링 하기</b>
01. [미션] 시스템에 블루/그린 배포 환경 적용하고 모니터링 하기




---

CURRICULUM

Part 2-3.

플랫폼 운영 및  
설정

파트별 수강시간 02:05:35

---

<b>CH01. Terraform 배포 환경과 관리</b>
01. Terraform 환경과 State, Lock 파일 관리의 중요성
02. Terraform Backend 환경 Migration해보기
03. Terraform Atlantis 환경 구성과, Terraform Cloud 연동해보기
<b>CH02. EKS 도입 시 가장 먼저 겪는 경험들</b>
01. EKS와 AWS IAM에 대해 이해하고 관리자 추가하기
02. EKS와 VPC CNI에 대해 이해해보기
03. EKS환경에서 ALB 설정과 여러가지 설정 적용해보기
04. EKS 환경에서의 CNI 설정
<b>CH03. [미션] Atlantis 환경 구성 후 EKS 어플리케이션 배포하기</b>
01. [미션] Atlantis 환경 구성 후 EKS 어플리케이션 배포하기



CURRICULUM

Part 3-1.

쿠버네티스  
안정적 운영

파트별 수강시간 09:38:57

<b>CH01. 사전준비</b>
01. 사전준비
<b>CH02. 노드의 예약 대기 시간과 인프라 비용을 줄이기 위해 karpenter 사용</b>
01. 오토 스케일링이란
02. karpenter가 각광받는 이유
03. 시나리오 설명 및 실습
<b>CH03. 서비스의 안정적인 운영을 위하여 파드의 안정적인 종료 지원</b>
01. 파드의 라이프사이클
02. 파드의 graceful shutdown
03. 시나리오 설명 및 실습
<b>CH04. 치솟는 트래픽에 유연하게 대응하기 위해서 오버프로비저닝 적용</b>
01. 파드의 스케줄링
02. 파드와 노드의 스케일링에 대한 비교
03. 시나리오 설명 및 실습
<b>CH05. 리소스 별 에러메시지에 대한 트러블 슈팅</b>
01. 리소스 별 에러메시지 유형 파악
02. 리소스 별 에러메시지 유형 분석
03. 리소스 별 에러메시지 유형 해결
<b>CH06. 삭제가 되지 않는 Stuck 상태인 리소스를 finalizer 초기화를 사용하여 제거</b>
01. 파드의 stuck 상태
02. finalizer 설명
03. 시나리오 설명 및 실습



CURRICULUM

Part 3-1.

쿠버네티스  
안정적 운영

파트별 수강시간 09:38:57

<b>CH07. kustomize, helm, kustomize + helm은 언제 사용해야 하는가?</b>
01. 기존 메니페스트 관리
02. kustomize와 helm 설명
03. 상황별 적합한 솔루션을 선택해보자
<b>CH08. 특정 애플리케이션에만 특정 권한을 세팅하기 위한 IRSA 적용</b>
01. k8s 환경 기본적인 애플리케이션의 권한
02. IRSA란 무엇인가?
03. 시나리오 설명 및 실습
<b>CH09. 안정적인 클러스터 버전 업그레이드</b>
01. 클러스터 버전을 해야하는 이유
02. 클러스터 버전과 노드 버전 업그레이드의 차이
03. 싱글 클러스터 버전 업그레이드 프로세스
04. 멀티 클러스터 버전 업그레이드 프로세스
<b>CH10. 쿠버네티스 격리 수준 및 운영 정책 수립</b>
01. 네임스페이스란?
02. 격리 수준의 범위 산정
03. 격리 수준에 맞는 운영 정책 수립
<b>CH11. livenessProbe의 치명적 단점</b>
01. livenessProbe의 치명적 단점
02. 시나리오 설명 및 실습
<b>CH12. 실무자의 운영성을 높이기 위하여 Umbrella Helm Chart 적용</b>
01. Umbrella Helm Chart는 무엇인가?
02. 시나리오 설명 및 실습
<b>CH13. 실무미션</b>
01. 쇼핑몰 서비스에서 대용량 트래픽이 들어왔을 때, 유연하게 처리해본다.



CURRICULUM

Part 3-2.

CI&CD 안정적 운영

파트별 수강시간 05:38:50

<b>CH01. 사전준비</b>
01. 사전준비
<b>CH02. CI 빌드속도 개선</b>
01. CI란 무엇인가
02. CI의 빌드속도가 느릴 경우 어떤 결과를 초래할까
03. 시나리오 설명 및 실습
<b>CH03. 도커라이징 최적화</b>
01. 도커라이징이란 무엇인가
02. 도커라이징을 최적화 하기 위한 방법
03. 시나리오 설명 및 실습
<b>CH04. 도커 이미지 풀 제약에 대해서 private 저장소를 사용하여 해결</b>
01. 퍼블릭 레지스트리에 대해서 알아본다
02. 프라이빗 레지스트리에 대해서 알아본다
03. 시나리오 설명 및 실습
<b>CH05. ArgoCD Image Updater를 사용하여 완전한 pull 방식을 지향</b>
01. 기존 push 방식의 gitops 플로우
02. 지향하는 pull 방식의 gitops 플로우
03. 시나리오 설명 및 실습
<b>CH06. secret값 변경시, ArgoCD에서 동기화 하는 방법</b>
01. ArgoCD에서 동기화 프로세스에 대해 알아본다
02. 외부 3rd 파티 서비스를 사용할 경우 동기화에 대해서 알아본다
03. 시나리오 설명 및 실습
<b>CH07. ArgoCD로 GitOps 운영시, orphaned resource에 대한 관리</b>
01. orphaned resource란 무엇인가
02. argocd에서 orphaned resource를 어떻게 모니터링 할까
<b>CH08. 실무미션</b>
01. 도커라이징시 최적화하여 CI 빌드속도를 높여 실질적인 서비스 안정성을 높여본다.



CURRICULUM

Part 3-3.

운영 고도화

파트별 수강시간 06:04:42

<b>CH01. 사전준비</b>
01. 사전준비
<b>CH02. Terraform 사용시 주의사항</b>
01. 테라폼에 대한 기본 아키텍처 및 플로우를 알아본다
02. 테라폼에서 어떤 문제가 발생할지 분석해본다
03. 시나리오 설명 및 실습
<b>CH03. Terraform vs Ansible vs ArgoCD 영역 구분</b>
01. Terraform, Ansible, ArgoCD 각 서비스의 역할에 대해서 설명한다
02. Terraform, Ansible, ArgoCD 각 서비스를 비교하여 장단점을 살펴본다
03. Terraform, Ansible, ArgoCD 각 서비스의 적합한 상황을 분석해본다
<b>CH04. 운영시 AWS Service quotas 문제 해결</b>
01. AWS 서비스 쿼타에 대해서 알아본다.
02. AWS 서비스 쿼타의 문제가 발생하는 원인이 해결책은 무엇인지 살펴본다
03. 시나리오 설명 및 실습
<b>CH05. 컨테이너 서비스를 운영하는 인스턴스의 EBS Volume 최적화</b>
01. 컨테이너 데이터 라이프사이클에 대해서 알아본다.
02. 컨테이너를 운영 중인 인스턴스의 볼륨 용량 초과와 원인과 해결책을 알아본다.
03. 시나리오 설명 및 실습
<b>CH06. 안정적이고 개발생산성을 높이기 위한 Github repo, branch, PR 정책 수립</b>
01. 다양한 Git flow에 대해서 알아본다.
02. Github pr 프로세스에 대해서 알아본다.
03. 시나리오 설명 및 실습
<b>CH07. 비용절감을 위하여 람다를 이용하여 유희상태인 RDS 사용정지</b>
01. 비용절감을 위하여 람다를 이용하여 유희상태인 RDS 사용정지하는 시나리오 설명 및 실습
<b>CH08. gitops-bridge를 사용하여 한 큐에 EKS Best Practice를 적용한 인프라, 애드온, 서비스 구축</b>
01. 기존의 EKS 애드온 구축 시나리오
02. 개선된 GitOps 기반 EKS 구축 시나리오
<b>CH09. github의 모든 리소스를 테라폼으로 관리</b>
01. github의 모든 리소스를 테라폼으로 관리
02. 시나리오 설명 및 실습



---

CURRICULUM

부록.  
부록

파트별 수강시간 01:14:53

---

CH01. 부록
01. 부록1
02. 부록2
03. 부록3
04. 부록4
05. 부록5
06. 부록6
07. 부록7



CURRICULUM

# 실무 장애 대응 프로세스.

## Part 1. 실전에서의 장애 대응과 문제 해결

파트별 수강시간 01:38:14

<b>CH01. 장애 후속 조치를 위한 개괄적 프로세스 및 장애 보고서 작성 요령 (Write incident report)</b>
01. 장애 후속 조치 프로세스와 프레임워크에 대한 개론
02. 예시를 통한 장애 보고서 작성 요령 및 후속 조치의 이해
03. 심각도에 대한 이해, 우선순위와 심각도의 차이
04. 장애 등급과 심각도 수준을 정의하는 방법
05. 장애의 근본 원인 파악하기
06. Fishbone Diagram and 5 whys
07. 건강한 코드 리뷰와 온콜 문화 만들기
08. 배포 프로세스의 고도화
09. 모니터링 지표 설계
<b>CH02. 장애 리뷰와 회고 (incident review and retrospective)</b>
01. 리뷰와 회고의 차이, 목적과 프로세스
<b>CH03. 장애 지표 관리 요령 (incident metrics)</b>
01. 장애지표 설계 및 관리 요령
<b>CH04. 실전 장애 관리 및 개선 (Incident in Action)</b>
01. 사례 연구 및 시나리오 기반 학습



CURRICULUM

# 실무 장애 대응 프로세스.

## 실전 장애 케이스 실습 전 준비

파트별 수강시간 01:04:47

<b>CH01. 장애 대응 방법론 요약 1 - 평상시</b>
01. OT : 커리큘럼 소개
02. OT : 강의방식 소개
03. 컴포넌트별로 어떤 문제가 발생할 수 있는지 알기
04. 전체 시스템의 구조를 이해하고 취약점을 분석하기
05. 장애 상황 발생시 누가 어떻게 행동해야 하는지 지침을 정하기
06. 장애 상황을 빠르게 감지할 수 있도록 모니터링을 수행하기
<b>CH02. 장애 대응 방법론 요약 2 - 장애 상황</b>
01. 가장 먼저 무엇을 해야 하는지
02. 무엇을 보아야 하는지
03. 장애 상황 대응 중 가져야 할 마음가짐
<b>CH03. 장애 대응 방법론 요약 3 - 상황 종료 이후</b>
01. 장애를 돌이켜 보기 - 장애 보고서 작성
02. 장애를 돌이켜 보기 - 회고 세션
<b>CH04. 실습에서 제시하는 서비스의 시스템 구조</b>
01. 서비스 소개 및 구조 설명
02. 클라우드 인프라 설정 실습
03. 서비스 애플리케이션 배포 실습
04. AWS 비용 설명 및 tear down

## CURRICULUM

# 실무 장애 대응 프로세스.

## 케이스 1: 배포 후 네트워크 장애 (1)

파트별 수강시간 00:58:31

<b>CH01. 상황 이해에 필요한 네트워크 개념</b>
01. 네트워크 요청과 응답의 흐름
02. 각 컴포넌트에서 발생할 수 있는 네트워크 문제
<b>CH02. 장애 상황 예시</b>
01. 시스템 구조 간략 설명
02. 인프라 프로비저닝 및 서비스 배포
03. 장애 유발 및 증상 관찰
<b>CH03. AWS 환경에서의 DNS 트러블슈팅</b>
01. DNS resolution 문제에 대한 이해
02. DNS 트러블슈팅을 위한 툴
03. DNS 레코드 수정을 통한 증상 해소
<b>CH04. 근본 원인 분석 및 장애 보고서 작성</b>
01. 장애 보고서 템플릿 소개 및 작성시 주의 사항
02. 직접적 원인 파악 및 기록
03. 5 Whys를 통한 근본 원인 파악
04. 방지 대책 및 시스템 개선책 제시
<b>CH05. 케이스 요약 및 중요 포인트</b>
01. 케이스 요약
02. 포인트 - 실수에 대한 자책보다는 시스템 개선 고민하기
03. 포인트 - 휴먼 에러로 인한 장애 후 더 좋은 회고를 하기 위해
04. 인프라 자원 tear down 및 비용 설명



CURRICULUM

# 실무 장애 대응프로세스.

## 케이스 2: 배포 후 네트워크 장애 (2)

파트별 수강시간 00:57:22

<b>CH01. 상황 이해에 필요한 Kubernetes 개념</b>
01. Kubernetes의 네트워크와 트래픽 라우팅
02. Helm과 Helm Template
<b>CH02. 장애 상황 제시</b>
01. 시스템 구조 간략 설명
02. 인프라 프로비저닝 및 서비스 배포
03. 장애 유발 및 증상 관찰
<b>CH03. Kubernetes 환경에서의 트러블슈팅</b>
01. Kubernetes 트러블슈팅을 위한 툴
02. 네트워크 모니터링 지표를 통한 원인 좁히기
03. 클러스터 내부 라우팅 플로우를 따라 원인 파악 및 조치
<b>CH04. 근본 원인 분석 및 장애 보고서 작성</b>
01. 장애 보고서 템플릿 소개 및 작성시 주의 사항
02. 직접적 원인 파악 및 기록
03. 5 Whys를 통한 근본 원인 파악
04. 방지 대책 및 시스템 개선책 제시
<b>CH05. 케이스 요약 및 중요 포인트</b>
01. 케이스 요약
02. 포인트 - 중요한 컴포넌트를 리팩터링할 때 주의할 점
03. 포인트 - Helm Template을 만들 때 고려할 점

## CURRICULUM

실무 장애 대응  
프로세스.케이스 3: 배포 후  
네트워크 장애  
(3)

파트별 수강시간 01:01:09

## CH01. 장애 상황 제시

01. 시스템 구조 간략 설명
02. 인프라 프로비저닝 및 서비스 배포
03. 장애 유발 및 증상 관찰

## CH02. 모니터링 지표를 통한 상황 이해

01. 장애 상황에서 살펴봐야 할 데이터베이스 지표와 로그
02. 장애 상황에서 살펴봐야 할 애플리케이션 지표와 로그

## CH03. Kubernetes 환경에서의 애플리케이션 트러블슈팅

01. 컨테이너까지 외부 환경 변수가 전달되는 플로우
02. Kubernetes 툴을 사용한 애플리케이션 로그 및 컨텍스트 파악
03. Kubernetes 매니페스트 문제에 대한 임시 조치

## CH04. 근본 원인 분석 및 장애 보고서 작성

01. 장애 보고서 템플릿 소개 및 작성시 주의 사항
02. 직접적 원인 파악 및 기록
03. 5 Whys를 통한 근본 원인 파악
04. 방지 대책 및 시스템 개선책 제시

## CH05. 케이스 요약 및 중요 포인트

01. 케이스 요약
02. 포인트 - 서비스 컴포넌트를 올바르게 마이그레이션하는 법
03. 포인트 - 안전한 배포를 위해 검증해야 하는 시스템의 범위
04. AWS 비용 설명 및 tear down



CURRICULUM

# 실무 장애 대응 프로세스.

## 케이스 4: 트래픽으로 인한 서버 애플리케이션 장애

파트별 수강시간 01:01:29

<b>CH01. 상황 이해를 위한 Kubernetes 자원 할당 개념</b>
01. 워크로드 리소스의 자원 할당 개념
02. 자원이 부족한 상황에서 발생할 수 있는 이상 현상
<b>CH02. 장애 상황 제시</b>
01. 시스템 구조 간략 설명
02. 인프라 프로비저닝 및 서비스 배포
03. Kubernetes 워크로드 리소스에 대한 모니터링 및 주요 지표
04. 장애 유발 및 증상 관찰
<b>CH03. Kubernetes 자원 관련 트러블슈팅</b>
01. 워크로드에 대한 수동 스케일링을 통한 임시 조치
02. 자원 할당에 관한 트러블슈팅 기법
<b>CH04. 근본 원인 분석 및 장애 보고서 작성</b>
01. 장애 보고서 템플릿 소개 및 작성시 주의 사항
02. 직접적 원인 파악 및 기록
03. 5 Whys를 통한 근본 원인 파악
04. 방지 대책 및 시스템 개선책 제시
<b>CH05. 케이스 요약 및 중요 포인트</b>
01. 케이스 요약
02. 포인트 - 적절한 워크로드 자원 할당과 스케일링
03. 포인트 - 시스템 디자인 측면의 문제 해결 방법
04. AWS 비용 설명 및 tear down



CURRICULUM

# 실무 장애 대응 프로세스.

## 케이스 5: 트래픽으로 인한 데이터베이스 장애

파트별 수강시간 01:17:12

<b>CH01. 상황에 따른 DB 원상 복구 방법</b>
01. RDBMS의 기본 구조
02. 웹 서비스에서 데이터베이스에 접근하는 과정
<b>CH02. 장애 상황 제시</b>
01. 시스템 구조 간략 설명
02. 인프라 프로비저닝 및 서비스 배포
03. 트래픽 피크에서 살펴볼 모니터링 지표
04. 장애 유발 및 증상 관찰
<b>CH03. DB 원상 복구 실습</b>
01. 데이터베이스 지표에서 이상을 확인하기
02. 커넥션 관련 문제에 대한 증상 해결법
<b>CH04. 근본 원인 분석 및 장애 보고서 작성</b>
01. 장애 보고서 템플릿 소개 및 작성시 주의 사항
02. 직접적 원인 파악 및 기록
03. 5 Whys를 통한 근본 원인 파악
04. 방지 대책 및 시스템 개선책 제시
<b>CH05. 케이스 요약 및 중요 포인트</b>
01. 케이스 요약
02. 포인트 - 데이터베이스 커넥션에 관련한 best practice
03. 포인트 - 데이터베이스 고가용성을 확보하기 위한 전략
04. AWS 비용 설명 및 tear down

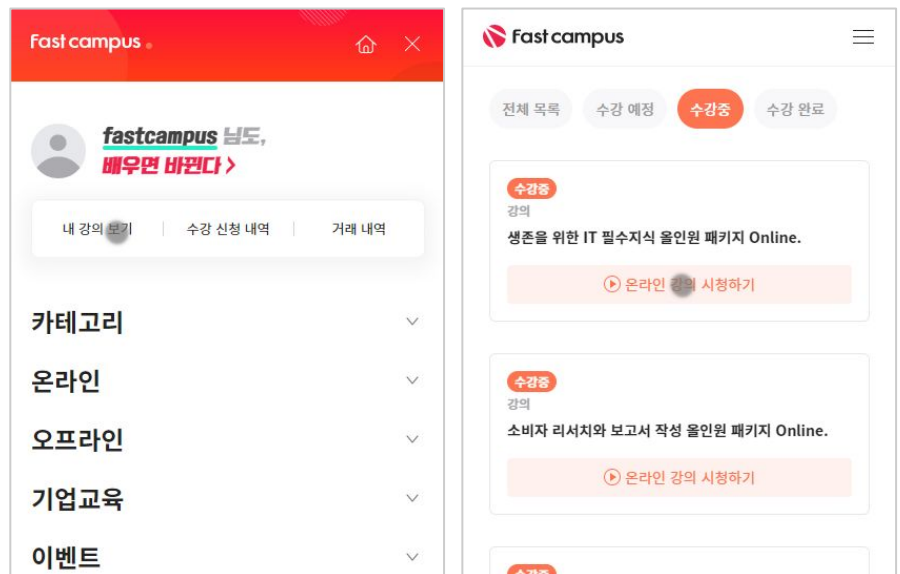


## 주의 사항

- 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.