



# LED 조 프로젝트 최종 보고서 : 3-Tier Architecture

조원 : 김주한, 김효섭, 이호진, 임효경, 최민영, 최한주

## 진행 일정

2024년 3월

< 오늘 >

일	월	화	수	목	금	토
3	4	5	6	7	8	9
📅 최종 완료 보고서						
👤 팀 구성도 및 계획도					✅ 중간점검	
🔧 공통 시나리오 & 요구사						
📋 개인 구성도 및 계획서						
10	11	12	13	14	15	16
📅 최종 완료 보고서						
		✅ 중간점검			✅ 중간점검	
17	18	19	20	21	22	23
📅 최종 완료 보고서						

## 목표

1. 단합과 협동을 통한 팀워크 증진(Hot-Spot Line을 이용하여 전체 인프라 연결)
2. 학습 내용의 활용을 통한 복습(현재까지 진행된 교육 과정의 교육 내용을 참고)
3. 배우지 않은 기술의 개인 학습 및 적용(관련 자료 및 기술 문서 확인)
4. Wi-Fi 공유기를 경유한 로컬 PC 간의 연결을 통해 구축하는 확장 네트워크 및 자원 공유
5. 교육과정에서 학습한 기술의 실제 적용을 통한 실무 경험

## 프로젝트 시나리오



기업 A는 온라인 쇼핑몰 운영 사업을 시작한 신규 IT 서비스 사업자로, 신규 고객 유입을 위한 정기 특가 상품 판매 행사의 시작을 앞두고 있습니다. 이 행사는 향후 일정 기간 동안 정기적으로 시행 될 예정이며, 시행될 때 마다 고객들은 단 한 시간 동안만 특가로 상품을 구매할 수 있습니다. 이를 위해, 기업 A의 인프라 팀에서는 원활한 행사 진행 및 활발한 DAU 유입을 위해 서비스 중단 최소화 인프라 구축을 목표로 쇼핑몰 이용 고객들의 행사 간 배송 및 상품 정보에 관한 원활한 문의 등록과 함께 최종 서비스 사용자들이 자사 서비스에 대한 요구사항을 깊이 있게 이해하고, 신속하게 대응할 수 있도록 기존 고객 Q&A 게시판이 필요하다고 저희 LED Company에 인프라 구축을 의뢰하였습니다. 행사 진행 중에는 많은 접속이 예상되기 때문에, 다수의 서버를 준비할 계획입니다. 하지만, 하나의 서버에 요청이 몰릴 경우 준비한 의미가 없기 때문에, 이를 분산시키기 위한 방법이 필요하다고 판단하였습니다.

최근, 국내 최대 온라인 전자상거래 플랫폼을 운영하는 기업 B의 상품 데이터 베이스 고장을 일으킨 적이 있습니다. 이에 판매자들이 서비스를 위해 업로드한 이미지 파일이 삭제돼 불편을 겪은 것으로 확인되었습니다. 서버가 하드웨어 장애로 판매자들이 업로드한 이미지가 삭제되는 일이 발생하는 등의 문제에 대비하기 위해 기업 A는 서버 장애 발생 시에도 판매자들이 업로드한 이미지 파일이 안전하게 보호될 수 있도록 대비하기로 결정했습니다. 이를 위해 Secondary DB 서버와 다운된 Primary DB 서버의 복구를 도울 수 있는 NFS(Network File System) 서버를 구축할 계획이며, 이를 통해 기업 A는 서버 장애로 인한 데이터 손실을 최소화하고, 판매자 및 고객들에게 안정적인 서비스를 제공할 수 있게끔 인프라 구축 방향을 설정하였습니다.

기업 A의 인프라 기존 구성과 변경 시 고려해야 할 요소는 다음과 같습니다. 정기 특가 상품 판매 행사의 시작을 앞두고 있는 기업 A는 첫 행사의 시작과 동시에 폭증할 트래픽의 규모를 정확히 예측해 수치로 제시하기 어려운 상황입니다. 기업 A는 현재 온프레미스 환경에서 가상화 기술을 활용해 WEB 서비스용 머신과 WAS용 머신을 별도로 운영하고 있으며, 자원의 효율성과 고가용성, 복구력 강화를 위해 Kubernetes 도입을 원하고 있습니다. 고객들의 서비스 접속이 폭발적으로 증가할 수 있는 특가 상품 판매 이벤트 특성 상 트래픽을 분산 처리해야 합니다.

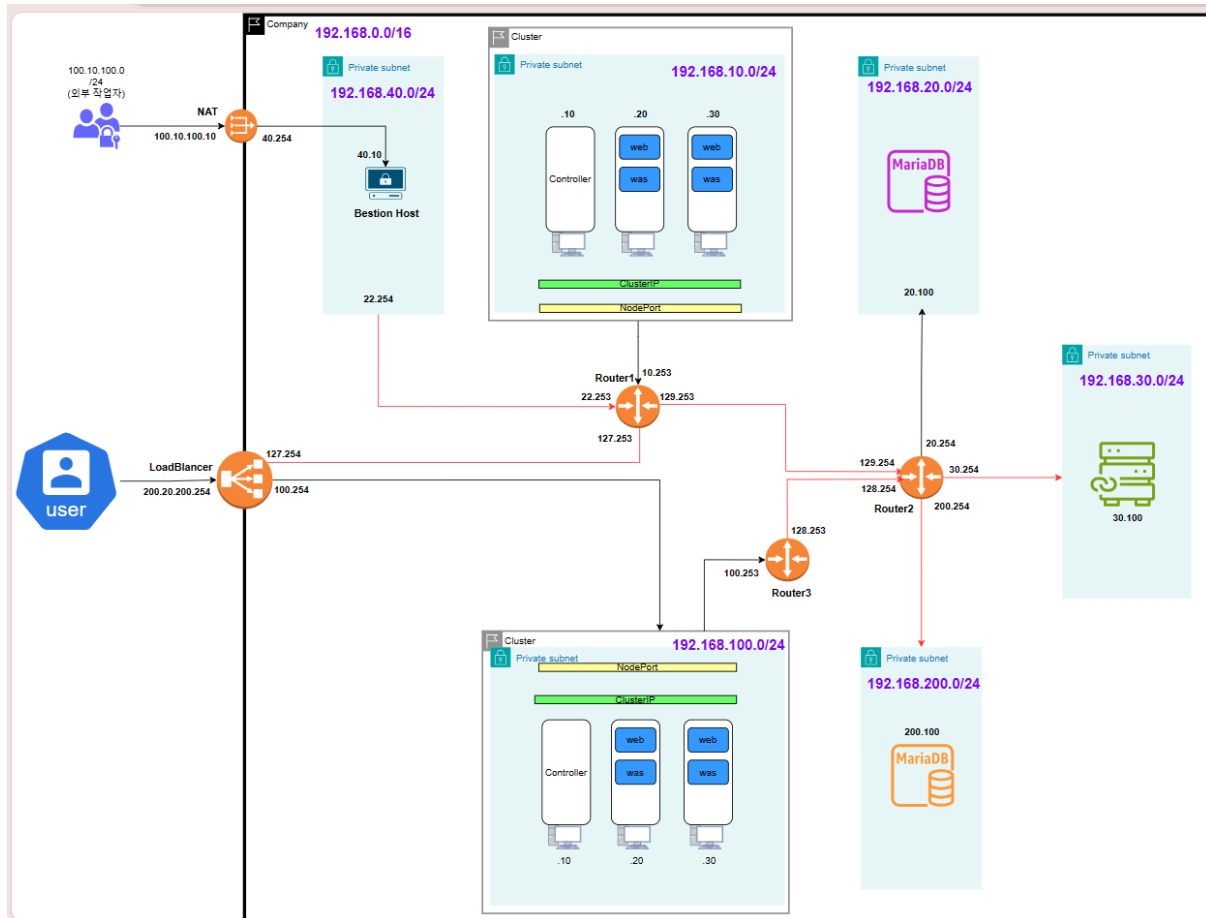
모든 고객 데이터는 외부에 공개할 수 없는 민감한 정보로 취급하며, 보안을 위해 고객 데이터용 데이터베이스는 독립된 온프레미스 서버에 저장해야 합니다. 고객 데이터 손실을 방지하고 장애에 대비하기 위해 데이터베이스는 동일 데이터센터 내 별도의 서버에 주기적으로 백업을 진행해야 합니다. 웹 어플리케이션의 업데이트가 빈번하게 자주 이루어질 예정이기 때문에 버전 업데이트를 용이하게 해야 하며, 외부에 있는 관리자를 위한 방법을 준비해야 합니다. 보안성 향상 및 검색 엔진 최적화(SEO)를 위해

웹 페이지에 SSL을 적용해야 합니다.

## 간트차트

조명		LED 조명																			
No	subnet	구분	5	6	7	8	9	10	11	12	13	14	15	16	17	18					
1		공통 시나리오 및 요구사항 명세 작성	LED 조명																		
2		팀 구성도 및 계획서																			
3		서브넷 분리 작업																			
4	192.168.10.0/24	쿠버네티스 환경 설정																			
5		베스컨 호스트 구축																			
6		NAT게이트 구축	김주한	김효섭	김주한	김효섭	김주한	김효섭	이호진	최한주	최민영	임효경									
7		Router1 구축																			
8		하스팟 설정 변경																			
9		마스터노드 구축																			
10	192.168.10.0/24	쿠버네티스 환경 설정																			
11		워커노드 구축 및 복제	김효섭	김주한	김효섭	김주한	김효섭	김주한	최한주	이호진	임효경	최민영									
12		쿠버네티스 클러스터 구축																			
13		Router1 연결 및 Yaml 파일 작성																			
14		서브넷 분리																			
15	192.168.30.0/24	NFS서버 구축	최민영	임효경	최민영	임효경	최민영	임효경	김주한	김효섭	이호진	최한주									
16		Router2번 연결																			
17		클러스터/NFS 연결 확인																			
18	192.168.20.0/24	MariaDB구축																			
19		Router2번 구축,연결	임효경	최민영	임효경	최민영	임효경	최민영	김효섭	김주한	최한주	이호진									
20		서브넷 분리																			
21		Router1/2 연결																			
22		쿠버네티스 환경 설정																			
23	192.168.100.0/24	쿠버네티스 클러스터 구축	이호진	최한주	이호진	최한주	이호진	최한주	최민영	임효경	김주한	김효섭									
24		LoadBalancer 구축																			
25		마스터노드 구축																			
26	192.168.100.0/24	쿠버네티스 환경 설정																			
27		워커노드 구축 및 복제	최한주	이호진	최한주	이호진	최한주	이호진	임효경	최민영	김효섭	김주한									
28		쿠버네티스 클러스터 구축																			
29		파트 로테이션 구축											LED 조명								
30		개인 구성도 및 계획서 제작																			
31		개인 파트 PPT 제작																			
32		최종 PPT 제작																			
33		발표 준비																			

## 프로젝트 최종 구성도



## 장비 목록

역할	장비명	주소	제공 서비스
LoadBalancer (200.20.200.0/24) (192.168.0.0/16)	LB	200.20.200.254  192.168.127.254 192.168.100.254	Apache2 (프록시, HTTP, SSL)
NAT (100.10.100.0/24) (192.168.0.0/16)	NAT	100.10.100.10.	Network Address Translation
		192.168.40.254	

Bastion Host  (192.168.40.0/24)  (192.168.22.0/24)	Bastion-Host	192.168.40.10	Bastion Host
		192.168.22.254	
Router  (192.168.0.0/16)	R1	192.168.22.253	Routing
		192.168.10.253	
		192.168.129.253	
		192.168.127.253	
	R2	192.168.129.254	Routing
		192.168.20.254	
		192.168.30.254	
		192.168.200.254	
		192.168.128.254	
	R3	192.168.128.253	Routing
		192.168.100.253	
Database  (192.168.20.0/24)  (192.168.200.0/24)	MariaDB-A	192.168.20.100	Primary Database
	MariaDB-B	192.168.200.100	Secondary Database
NFS  (192.168.30.0/24)	NFS	192.168.30.100	Network File System
Cluster A  (192.168.10.0/24)	Controller-A	192.168.10.10	Kubernetes Controller
	Worker-A1	192.168.10.20	Kubernetes Worker
	Worker-A2	192.168.10.30	Kubernetes Worker
Cluster B  (192.168.100.0/24)	Controller-B	192.168.100.10	Kubernetes Controller

	Worker-B1	192.168.100.20	Kubernetes Worker
	Worker-B2	192.168.100.30	Kubernetes Worker

## 구역 별 구현 방식

### 1. Bastion Host

- A의 기업에서 외부에 있는 관리자가 접근할 수 있는 방법을 요청했기 때문에,

NAT Gateway를 통한 Bastion Host의 접근을 구현

- 관리 대상 서버 : K8S Controller A, DataBase, NFS

### 2. K8S Cluster A

- A의 기업에서 자원의 효율성과 고가용성, 복구력 강화를 위해 Kubernetes 도입을 원하고 있으며, 버전 업데이트가 용이해야 하기 때문에

Kubernetes Cluster를 구축하였습니다.

### 3. Primary DB

- 고객의 데이터는 외부에 공개할 수 없는 민감한 정보로 취급하며, 보안을 위해 고객 데이터용 데이터베이스는 독립된 온프레미스 서버에 저장을 요청했기 때문에, 온프레미스 DataBase를 사용하였습니다.

### 4. NFS

- 다운된 DB 서버 복구를 도울 수 있는 NFS 서버의 구축을 요구했기 때문에

온프레미스 기반 NFS 서버를 구축하였습니다.

- DB서버와 마운트된 NFS서버가 있기 때문에, DB의 데이터가 손실되었을 경우, 마운트된 경로의 백업 파일을 사용하여 복구 할 수 있도록 구축하였습니다.

### 5. Replica DB

- Primary DB에 장애가 발생하였을 경우, 고객 데이터 손실을 방지하고 빠른 대응이 가능하도록 Replica DB를 구축하였습니다.

## 6. LB & K8S Cluster B

- 보안성 향상 및 검색 엔진 최적화(SEO)를 위해 웹 페이지에 SSL을 적용을 원했기 때문에 LB에 SSL 적용
- 트래픽을 분산 처리하기 위해 LB를 구현 하였고, Health Check 기능을 통해

비 정상 상태 서버로의 트래픽을 전송하지 않도록 구현하였습니다.

# 과정

## 1. Duplicated

- a. 가상 머신에 와이파이 어댑터를 여러 개 연결하여 사용하려 했음.  
ping을 보냈을 때 신호가 복제되는 문제가 발생
- b. 쿠버네티스 클러스터에 Wifi 어댑터 연결 시 , Pod 간 통신이 원활하게 이루어 지지 않는 문제가 발생
- c. 분석 결과, 물리적으로 하나의 장치인 와이파이 어댑터를 논리적으로는 가상머신에 여러 개 부착했다는 것이 원인 임을 알아냈습니다.
- d. 따라서, 라우터 역할을 하는 가상 머신에 다수의 물리적 어댑터를 연결하는것이 아닌, 하나의 물리적 어댑터에 다수의 논리적 주소를 할당하는 방식으로 우회하였습니다.
- e. Pod간의 통신 문제는 쿠버네티스 클러스터에 연결된 라우터에 Wifi를 연결하는 방식으로 우회하였습니다.

## 2. Ingress

- a. 인그레스를 사용하려고 하였으나 인그레스의 주요 목적은  
SSL Termination 적용, 도메인 사용이라는 것을 인지하여,  
목적에 맞지 않다고 판단하였습니다.
- b. Ingress 오브젝트 대신, NodePort를 사용하는 방향으로 전환하였습니다.

## 3. Health Check



- a. 로드밸런서가 비정상 상태인 웹 서버에 요청을 전달하면 안되기 때문에, 헬스체크를 이용하여 비정상 상태인 서버는 타겟 그룹에서 제외해야 했습니다.
- b. 아파치 헬스체크 중에 응답코드의 첫 자리를 이용해 상태를 판단하는 기능을 사용하려 했으나, 웹 서버의 상태에 무관하게 응답코드가 400이었기 때문에 다른 방법을 찾아 보았습니다.
- c. 아파치 헬스체크는 응답을 보내지 않는 서버를 타겟 그룹에서 제외하기 때문에, 웹 서버에 이상이 생기는 경우 쿠버네티스의 파드 상태를 Ready 0/1로 바꾸는 방법으로 우회를 시도했습니다.
- d. 쿠버네티스의 readinessProbe가 httpGet를 이용했을 때 , Ready 0/1가 표시되는 경우는 다음과 같습니다
  - i. web server up, was server up, db server down
  - ii. web server up, was server down
  - iii. web server down

다시 말하면, 모든 서버가 정상인 경우에는 Ready 1/1로 표시되어 요청을 수신하게 됩니다.

## 결과

1. 외부에서 공개된 IP를 통하여 내부의 Bastion Host로 SSH연결을 수행할 수 있게 되었습니다.
2. K8S 클러스터를 구현하여 자원의 효율성과 고가용성, 복구력을 향상시켰습니다.
3. 온프레미스로 DB를 구현하여 보안과, 복구능력, 고 가용성을 향상시켰습니다.
4. NFS를 구축하여 다운된 DB 서버의 복구를 도울 수 있게 되었습니다.
5. Replica DB를 구현하여 고객 데이터 손실을 방지하고 장애에 대비할 수 있게 되었습니다.
6. LB를 구축하여 트래픽을 분산 처리할 수 있게 되었고, SSL을 LB에 적용하여

보안성 향상 및 검색 엔진 최적화(SEO)를 향상시킴