Portfolio 전유진

🗂️ **전유진 | Server Engineer / Network Engineer**
ujin5712@gmail.com

🚀 핵심 프로젝트

지원 직무와 직접 관련된 Kubernetes / AWS / Docker 기반 인프라 구축 프로젝트입니다.

1. Docker Swarm 기반 3-Tier 웹 서비스 배포

📅 **기간** 2026.04.27 ~ 2026.05.15 · **참여인원** 4명, 팀장

프로젝트 개요

- 기존 웹 서비스를 Docker 기반으로 컨테이너화하고, 4대의 Ubuntu VM으로 구성된 Docker Swarm 환경에 분산 배포
- Frontend·Backend·Database를 분리한 3-Tier Architecture를 기반으로 서비스 가용성 및 운영 효율성 개선
- CI/CD와 모니터링 환경을 구축하여 배포 자동화 및 서비스 상태 관측 체계 구현

담당

- 기존 웹 서비스 Docker Image 및 Container 환경 구성
- Docker Swarm 기반 다중 노드 클러스터 및 서비스 배포 구조 설계
- Frontend·Backend Replica 및 Load Balancing 구성

- GitHub Actions 기반 CI/CD Pipeline 구축
- Prometheus·Grafana 등을 활용한 모니터링 환경 구축

개발환경 / Tech Stack

Infra	CI/CD·모니터링	App
`Ubuntu` `VirtualBox` `Docker` `Docker Swarm` `Nginx`	`GitHub Actions` `Harbor` `SonarQube` `Prometheus` `Grafana` `Loki` `Tempo`	`React` `TypeScript` `FastAPI` `MariaDB` `MongoDB` `Redis`

구현기능 및 개발과정

- 4대의 Ubuntu VM으로 1 Manager · 3 Worker Docker Swarm Cluster 구성
- Frontend·Backend를 각각 2개 Replica로 배포하여 서비스 이중화
- Nginx Reverse Proxy 및 Load Balancing을 적용하여 Replica 간 요청 분산
- web-net, db-net, monitor-net 등 Overlay Network를 역할별로 분리하여 서비스 간 접근 범위 제어
- MariaDB·MongoDB·Redis의 데이터 저장소를 별도 노드에 배치하고 Volume을 활용하여 데이터 영속성 확보
- Health Check 및 Rolling Update/Rollback 정책을 적용하여 배포 중 서비스 안정성 확보
- GitHub Actions → SonarQube → Docker Build → Harbor → Docker Swarm으로 이어지는 CI/CD Pipeline 구축
- Prometheus·Grafana·Loki·Tempo 기반으로 Metric, Log, Trace 통합 모니터링 환경 구성

성과

- 기존 웹 서비스를 4-Node Docker Swarm 기반 분산 서비스 구조로 전환
- Frontend·Backend 이중화 및 Load Balancing을 통해 단일 컨테이너 장애에 대응 가능한 구조 구현
- 코드 변경부터 Image Build·Registry Push·Service Update까지 이어지는 자동 배포 프로세스 구축
- Container·Host·Application의 상태를 확인할 수 있는 통합 Observability 환경 구축

▼ 느낀점

- 단순한 Docker Container 실행을 넘어 네트워크, 데이터 영속성, 가용성, 배포 및 모니터링을 함께 고려해야 안정적인 서비스 운영이 가능함을 경험
- Docker Swarm을 활용해 직접 오케스트레이션 환경을 구성하며 분산 서비스와 Container Orchestration에 대한 이해를 심화
- CI/CD와 Observability를 구축하면서 개발 이후의 배포·운영 과정까지 고려하는 DevOps 관점의 중요성을 체감

---

2. Kubernetes 기반 헬스용품 이커머스 플랫폼 구축

기간 2026.05.26 ~ 2026.06.12 · **참여인원** 4명, 팀장

프로젝트 개요

- 온프레미스 환경에서 Kubernetes 기반 MSA 이커머스 플랫폼 구축
- CI/CD, 네트워크, Observability를 포함한 컨테이너 운영 환경 구성

담당

- Kubernetes Infrastructure / Network / Observability
- 클러스터 설계·구축 및 장애 분석·복구

개발환경 / Tech Stack

`Kubernetes` `Kubespray` `Ansible` `Calico` `MetalLB` `NGINX Ingress` `Prometheus` `Grafana` `Loki` `Tempo` `Alloy` `Jenkins` `Harbor` `Docker`

구현기능 및 개발과정

- Kubespray 기반 Control Plane·etcd·Worker 다중 노드 Kubernetes 클러스터 구축
- External ETCD 구조와 Multi-Master 환경의 Kubernetes 클러스터 운영
- Calico(VXLAN)·MetalLB·NGINX Ingress 기반 Pod 네트워크 및 외부 트래픽 라우팅 구성
- Prometheus·Grafana·Loki·Tempo·Alloy 기반 Metric·Log·Trace 통합 Observability 환경 구축
- Jenkins → Harbor → Kubernetes CI/CD 파이프라인 구성

- etcd·Node·Network·Metrics Server 장애를 로그 및 계층별 통신 점검을 통해 분석·복구

성과

- 제한된 온프레미스 환경에서 Kubernetes 인프라 전반을 직접 설계·구축·운영
- Cluster, Network, Observability 장애 대응을 통해 Kubernetes 트러블슈팅 역량 강화

▼ 느낀점

- Kubernetes 운영에는 Pod 배포뿐 아니라 Network·etcd·Resource·Observability 등 인프라 전반에 대한 이해가 중요함을 경험

---

3. AI 러닝 종합 플랫폼 'DAI RUN' – 온프레미스 K8s 구축부터 AWS EKS 이전까지

🏃 **기간** 2026.06.30 ~ 2026.08.26
참여인원 4명
팀장 · DevOps/Platform Engineer
https://dairun.site

프로젝트 개요

- 위치, 날씨·대기질, 러닝 기록, 사용자 선호 데이터를 활용하여 사용자의 당일 러닝 거리와 강도, 러닝 코스를 추천하는 AI 기반 러닝 종합 플랫폼
- 초기 Docker 기반 서비스를 MSA 및 Kubernetes 환경으로 전환하고, 온프레미스 Kubernetes 환경에서 서비스 안정화 및 관측 체계를 구축한 뒤 AWS EKS 기반 클라우드 환경으로 확장

개발환경 / Tech Stack

`Kubernetes` `Docker` `AWS EKS` `Terraform` `Argo CD` `Istio` `Prometheus` `Grafana` `Loki` `Tempo` `KEDA` `Karpenter` `PostgreSQL` `Redis` `Kafka` `SNS/SQS`

구현기능 및 개발과정

- 온프레미스 및 AWS 전체 인프라·네트워크 아키텍처 설계

- Kubernetes 기반 MSA 배포 환경 구축 및 운영, Istio 기반 서비스 네트워크 및 외부 트래픽 라우팅 구성
- 애플리케이션·DB·Kafka·Observability 워크로드를 Kubernetes에 분리 배치
- Gateway → Service → Pod → DB 단위로 트래픽 흐름을 분석하여 장애 구간 특정
- etcd 디스크 장애 발생 시 Snapshot을 이용해 Kubernetes Cluster 복구
- Argo CD 기반 GitOps 배포 환경 구축
- NAT 의존도를 줄이고 VPC Endpoint 중심의 Private Network 구조 설계
- Site-to-Site VPN을 이용한 온프레미스-AWS Hybrid 환경 구성
- VPC·Subnet·EKS·VPC Endpoint 등을 Terraform으로 관리하고, Kafka → SNS/SQS, MinIO → S3, Harbor → ECR 등 클라우드 환경에 적합한 구조로 전환

성과

- 온프레미스 Kubernetes 구축부터 AWS EKS Migration까지 전체 인프라 전환 과정 경험
- Observability 환경을 기반으로 실제 Gateway 및 etcd 장애 원인 분석·복구
- Terraform·Argo CD를 활용한 IaC 및 GitOps 기반 운영 경험 확보
- 워크로드 특성에 따라 HPA·KEDA·Karpenter를 비교하고 Auto Scaling 구조 설계

▼ 느낀점

- 기술을 많이 사용하는 것보다 서비스 요구사항과 운영 환경에 맞는 기술을 선택하는 것이 중요하다는 점을 배웠습니다.
- 또한 실제 장애 대응을 통해 Network → Gateway → Service → Pod → DB 순서로 시스템을 계층적으로 분석하며 문제 범위를 좁혀가는 DevOps/SRE의 트러블슈팅 방식을 경험했습니다.

🧩 네트워크·프로토콜 기본기

TCP/IP, 보안 통신 등 인프라 엔지니어의 기초 체력을 보여주는 프로젝트입니다.

✉️ **Java 소켓 기반 SMTP 메일 전송 프로그램**
2024.09.27 ~ 2024.11.21 (4인)

🔐 **VPN 구성 (OpenVPN)**
2026.02.26 ~ 2026.03.30
Linux↔Linux,
Linux↔Windows 환경에서

Java Socket/SSLSocket으로 Gmail SMTP 서버와 직접 통신하는 메일 클라이언트를 구현. HELO·AUTH LOGIN·MAIL FROM·DATA 등 SMTP 명령과 상태 코드(220/250/354/235/221)를 단계별로 검증하며 애플리케이션 계층 프로토콜의 동작 원리를 직접 구현으로 체득.

`Java` `Socket/SSLSocket` `SSL/TLS` `RFC 5321`

OpenVPN 서버·클라이언트를 구축. 인증서·비대칭키 기반 인증, iptables 패킷 포워딩, 라우팅 구성까지 직접 설정해 외부 클라이언트가 폐쇄망 내부 서버에 접근하는 환경을 완성.

`OpenVPN` `iptables` `TCP/IP` `Rocky Linux`

📎 기타 프로젝트

지원 직무와의 연관성은 낮지만 데이터베이스·백엔드 기초를 보여주는 프로젝트입니다.

프로젝트	기간	기술 스택
군자동 음식점 데이터베이스 관리 프로그램 — ER 모델링부터 MFC/ODBC 인터페이스까지 구현	2024.03 ~ 2024.06	MSSQL, C++ MFC, ODBC
MariaDB + MongoDB 구성 — 법정동 코드 기반 부동산 시세·생활인프라 DB 설계	2026.03.31 ~ 2026.04.13	MariaDB, MongoDB
상권추천 서비스 '상추' — 서울시 상권 데이터 분석·시각화 백엔드 개발	2026.04.14 ~ 2026.04.24	Python, FastAPI, MariaDB, MongoDB