

서버_개발자_이동엽_경력기술서

밑줄이 있는 문장에 관련 링크를 첨부하였습니다.

Contact.

Name. Dongyeop Lee

Email. ldy_1204@naver.com

Phone. 010-5171-1184

Channel.

Github : <https://github.com/2dongyeop>

LinkedIn : [linkedin.com/in/2dongyeop](https://www.linkedin.com/in/2dongyeop)

blog : <https://velog.io/@dongvelop>

Introduce.

의료 서비스 스타트업에서 Java & Spring 기반의 2년차 서버 개발자로 근무하고 있습니다.

헬스케어 앱(송아리당뇨, 송아리에어 IoT)의 API 서버를 개발하며, 의료 데이터 연동 및 안정성 개선을 경험하고 있습니다.

Mindset.

- '창피함은 한 번이지만, 얻는 기억은 영원하다.'라는 마음으로 모르는 것을 부끄러워하지 않고, 자기객관화를 통해 성장하려 합니다.
- 부족한 지식이라도 특정 주제에 대해 정리하여 사내 기술 공유를 진행하고, 피드백과 토론을 통해 기술 역량을 발전시키고 있습니다.
- 기능 개발 전에 구현 방안을 팀과 논의하고, 개발 후에 코드 리뷰를 통해 다양한 시각에서 검토하며 더 나은 코드를 고민합니다.

Culture.

- 소규모 개발팀에서 기술 문화를 정착시키기 위해, 사내 기술 위키를 활성화하고 온보딩 문서/개발 히스토리 문서화를 주도하고 있습니다.
- 레거시 코드의 신뢰성을 높이기 위해 테스트 코드를 도입하고, 부하 테스트 환경을 구축하여 최대 수용 가능 트래픽을 측정하고 있습니다.
- 최신 기술 변화와 트러블슈팅에 관심이 많아, 여러 테크 기업들의 기술 블로그를 매일 아침 정독하며 흐름을 따라가고 있습니다.

Experience.

2023. 07 ~

(주)송아리아이티 (Songaree IT)

1년 8개월, 재직 중

약 6만명의 회원 보유. 당뇨 및 산소포화도를 관리하는 헬스케어 앱/웹 서비스를 제공하는 스타트업

서버 개발자

- 송아리당뇨, 송아리에어 IoT 서버 운영 및 고도화(모놀리식 → MSA 전환)
 - JDK 1.8, Spring Boot 2.7 → JDK 17, Spring Boot 3.0.13, Spring Cloud 2022.0.4
 - 신기능 개발 주도 : 요구사항 분석, DB 설계, API 스펙 정의, 구현 및 배포 일정 관리
 - nGrinder를 이용한 마이크로서비스 성능 분석 및 트랜잭션 처리량(TPS) 측정 후, 쿼리 및 코드 최적화
 - Pinpoint를 이용해 외부 API 및 마이크로서비스 간 호출 시 실패하는 병목 지점 식별 : Retry 정책 설계 및 구현을 통해 API 호출 실패율 90% 감소
- 사내 기술 발표 15회, 강의/컨퍼런스 후기 3건 작성
 - [JDK 1.5부터 JDK 22까지의 주요 문법 정리](#)
 - [JDK 21에서만 발생하는 Out Of Memory Error](#)
 - [JVM Cold Start 최적화하기](#)
 - [Spring Boot with GraphQL](#)
- Skill Keyword
 - Java, Spring Boot, Spring Cloud, JPA, JUnit5, Docker, Nginx, Jenkins, n8n, Git
 - AWS(EC2, S3, SES, RDS, CloudWatch), Grafana, Prometheus, Pinpoint, nGrinder

Project.

2024. 12

송아리아이티
(SongareelT)

문자 전송 서비스 변경

진행 배경

- 송아리 당뇨, 송아리 에어 IoT에서 사용되는 문자 전송 기능 변경 작업(DB Agent 기반으로 동작)
- DB Agent가 주기적으로 DB로부터 문자 테이블을 폴링하여 새로운 데이터가 있을 경우, 문자를 보내는 방식
- 폴링 방식이므로 실시간 성이 떨어지며, 폴링 주기를 짧게할 경우 DB 부하가 증가하는 문제 발생
- 또한, DB Agent에 대한 헬스체크가 어려워 장애 감지가 어렵다는 단점이 존재

기술 스택

- Java 17, Spring Boot 3.0, Spring Cloud 2022.0, JPA, NCP SENS, Docker, MySQL

당시 역할 및 기여

- 문자 전송 기능을 제공하는 플랫폼 선정(비용, 카톡 전송 여부, 해외 문자 지원 여부 등을 비교)
- 현재 문제점 분석, 개선 방안 설립 및 DB Agent 방식의 문자 전송 기능을 API 방식으로 변경
- 선언형 코드 유지 및 비동기 처리를 위한 Reactive Feign 도입 (관련 게시물 일부)
 - Retry 로직 및 최종 완료되지 않은 작업들에 대한 배치 처리 구현
 - **OpenFeign & @Async** 을 이용한 방식보다 약 90% 성능 향상 : 200ms → 9ms

성과

- 더이상 DB Agent가 폴링하지 않으므로, **DB 부하 감소**
- API 호출 방식으로 변경함으로써, **실시간성 증가**
 - 해당 과정에서 문자 전송 실패시 케이스별 Retry 로직을 작성하여 문자 전송 성공률 개선
 - Retry 최종 실패시에 Alert를 추가하여, 모니터링 개선

2024. 11

송아리아이티
(SongareelT)

송아리에어 서버사용자 모니터링 기능 개발

진행 배경

- 환자의 산소포화도/맥박수 측정 결과를 보호자 및 의료진이 실시간으로 모니터링할 수 있도록 하기 위한

기술 스택

- Java 17, Spring Boot 3.0, Spring Cloud 2022.0, JPA, Docker, MySQL 8.0, Jenkins

당시 역할

- DB 스키마 설계 및 API 스펙 정의, 서버사용자 매핑 기능 및 실시간 알림 기능 개발
- 시나리오 별 Flow Chart 문서화 및 공유

고민 내용 및 해결 방법

- 실시간 알림 처리 구현 중, 서버에서 클라이언트로 일방적인 데이터 전송이므로 SSE를 고려
- **SSE를 사용하기 위해선 커넥션 관리 측면에서 JPA OSIV 옵션 비활성화가 필수적이라고 판단**
 - OSIV가 활성화되어있을 경우, 클라이언트에게 응답이 완료될 때까지 트랜잭션이 닫히지 않아 DB Connection Pool을 반납하지 않는 문제가 있음을 파악
 - 단, 기존 프로젝트들에서는 OSIV 옵션이 활성화되어 있기에 기존 서비스에 영향이 발생
 - 따라서 약간의 **시간 차를 감안하고 REST 방식으로 DB에 저장하고, 조회하는 방식으로 안정성을 선택**

- 아쉬웠던 점은 당시 클라이언트 간 실시간 통신을 지원하는 WebRTC라는 기술을 알지 못해, 선택지로 고려하지 못했다는 점이 있습니다. 추후 이 기술을 학습해놓는다면, 개선할 수 있다고 생각합니다.

2024. 10

송아리아이티
(SongareelT)

서비스 복구 및 백업 프로세스 설립

진행 배경

- EC2 오토스케일링 그룹을 삭제한 탓에, 연관된 사용 서비스중인 EC2 인스턴스가 종료되었던 장애가 발생
- 당시 복구 프로세스가 없었기에, 서버를 구성/기동하기까지 약 3시간 소요
- 따라서 서비스를 복구한 뒤, 동일한 문제가 발생했을 때 바로 복구할 수 있도록 프로세스를 설립

기술 스택

- Linux Crontab, AWS CLI

당시 역할

- **AWS CLI를 이용한 EC2 시작템플릿 갱신 자동화 (관련 게시물 일부)**
 - 기존 인스턴스의 사양, 네트워크 및 보안 설정, EBS까지 동일한 스펙으로 AMI 백업 후 버저닝 관리
- **n8n을 이용해 Web/API Server 헬스체크 추가 구축**
 - Prometheus & Grafana를 이용한 헬스체크와 달리 **즉각적인 장애 대응** 등의 동적 작업이 가능하여 채택
 - → n회 연속 API 서버의 헬스체크가 실패하면, 자동으로 **시작템플릿을 이용해 EC2 인스턴스 추가 기동**

성과

- **사내 최초 AWS CLI 도입** 및 Linux Crontab을 이용한 주기적인 백업 프로세스 구축
- 장애 발생시, 즉각적으로 최신화된 EC2 인스턴스를 기동할 수 있도록 안정성 증가

2024. 09 ~ 2024. 10

송아리아이티
(SongareelT)

사내 온프레미스 서버 이관

진행 배경

- 개발/테스트 환경으로 사용중인 온프레미스(구서버, 신서버) 서버 2대를 1대로 통합하는 작업

기술 스택

- Linux, Docker, Nginx, mysqldump, Jenkins

당시 역할

- 개발/테스트 환경의 API/Web 서버, DB(MySQL 5.7) 이관 (관련 게시물 일부)
 - 프로젝트별 JDK 버전 상이 문제를 해결하기 위해 **Docker 컨테이너 환경 구축 및 운영**
 - **mysqldump를 이용한 DB 백업 자동화 프로세스 구축**
 - Nginx(Web Server) **Brotli** 압축 적용
- 레거시 프로젝트에서 이용하던 Gitlab CI/CD를 고도화 프로젝트와 동일하게 Jenkins로 변경
- 정전 상황에서 리부트 시 **crontab**을 이용한 각 서버 및 DB 기동 프로세스 설계

2024. 07
~ 2024. 09

송아리아이티
(SongareelT)

송아리 프리미엄(정기 구독 서비스) 개발

진행 배경

- 송아리당뇨 앱에서 광고 제거 등의 일반 회원과 차별화된 기능을 제공하는 **구독형 유료 서비스** 개발

기술 스택

- Java 17, Spring Boot 3.0, Spring Cloud 2022.0, JPA, Openfeign, MySQL 8.0, Jenkins

당시 역할

- 기획 참여, 요구사항 분석, DB 스키마 설계, API 스펙 정의 및 기능 개발
 - 스토어별 구독/해지 Sequence Diagram 작성 및 공유
- Google/Apple 각 스토어 별 **결제 내역 검증 시, 결제 상태 반영 지연 문제 발생**
 - Backoff 방식의 Retry 정책을 적용하여 **처리 실패율 80% 감소**
- 여러 스토어 별 공통 구독 상태 추출 후 사내 구독 상태 정의(e.g. 활성, 유예, 취소, 만료)
 - 스토어별 구독 상태 변경 감지 스케줄링 기능 개발

성과

- 서비스 출시와 동시에 Google/Apple 스토어 앱 광고를 진행하여, 신규 회원 500명 증가

2024. 04

송아리아이티
(SongareelT)

AWS RDS 모니터링 시스템 구축

진행 배경

- 상용 DB로 사용 중인 AWS RDS(MySQL 8.0) 모니터링 구축 작업
- AWS CloudWatch의 경우 경고 생성 및 발생시 비용이 발생하여, 비용 절감을 위해

percona/rds_exporter(오픈소스)를 이용하여 메트릭을 수집

기술 스택

- Linux, Docker, rds_exporter, Grafana, Prometheus, Go(소스코드 일부 수정)

당시 역할

- AWS RDS 메트릭을 수집할 수 있는 RDS Exporter 조사 및 비교
- 수집한 메트릭을 Prometheus & Grafana를 이용해 모니터링 및 Alert 구축
 - Grafana Template을 가져온 뒤, 메트릭 명을 수정하여 대시보드 최적화

고민 내용 및 해결 방법

- rds_exporter는 내부적으로 AWS CloudWatch API를 호출하여 메트릭을 검사
- AWS CloudWatch API는 메트릭을 60초마다 갱신하지만, rds_exporter를 10초마다 메트릭을 조회
 - 이로 인해, 중복된 메트릭을 가져오면서 불필요한 비용을 발생시키는 문제 발생
- 원본 소스코드로부터 주기를 조정하여 중복 조회를 해결하고, 사용하지않는 불필요한 메트릭은 조회하지 않도록 수정

성과

- AWS CloudWatch 이용 시보다 약 99% 비용 절감**
- 196대의 인스턴스를 기준으로 메트릭을 수집 시, 하루에 약 0.25\$의 요금 부과

2024. 03

송아리아이티
(SongareelT)

이미지 리사이징 처리 서버 구축

진행 배경

- 기존에 썸네일 등과 같은 작은 이미지를 웹에서 S3로부터 불러와, 사이즈를 조정하는 작업이 존재
- 이미지 크기 가공 과정에서 느끼는 불편함을 해소하기 위해 오픈소스 도입

기술 스택

- Linux, Docker Compose, Thumbor(오픈소스)

당시 역할

- 현재의 불편함 및 요구사항 정리, 오픈소스 조사, 이미지 처리 서버 구축
- Thumbor 내부 동작 분석
 - 내장된 Redis가 얼굴 및 특징 감지 기능과 실패 작업의 대기열로 사용함을 파악
 - 불필요한 기능으로 판단 후, Redis 관련 설정 비활성화 처리

성과

- 썸네일 용 이미지 크기 조정 및 회전 등의 기능 제공
- 내장 Nginx를 이용해 이미지 캐싱(S3 호출 횟수 감소) 및 응답 압축 전달
- 더이상 Client에서 AWS S3로 직접 접근하지 않고, 이미지 처리 서버를 호출하도록 보안 강화

2023. 10 ~ 2024. 04

송아리아이티
(SongareelT)

송아리당뇨 API Server 마이그레이션

진행 배경

- 메인 서비스인 송아리당뇨 앱의 API 서버를 MSA로 전환하며 고도화하는 작업
 - AS-IS : Java 8, Spring Boot 2.7
 - To-BE : Java 17, Spring Boot 3.0, Spring Cloud 2022.0

기술 스택

- Java 17, Spring Boot 3.0, JPA, MySQL 8.0, Jenkins

당시 역할

- **Spring Boot 2.7 → 3.0** 마이그레이션 가이드 정리
- 레거시 코드의 신뢰성을 높이기 위해 테스트 코드를 도입
- FCM Legacy API에서 FCM HTTP v1 API로 마이그레이션
- 약 250개의 관련 테이블 ERD 및 테이블 명세서 작성

성과

- 약 60만건의 3개월치 데이터 조회 API 성능 95% 개선
 - 8개의 내부 동작을 비동기 처리 & 필요한 컬럼만 조회하도록 일부 쿼리 개선 : **7.73s → 362ms**
 - 약 5만명의 사용자 통계시 소요시간 단축 : **12분 → 2분**
- 모놀리식에서 MSA로 전환함으로써 배포 소요 시간 40% 단축

Skill.

Language.

- **Java 17, 21**: 최신 Java 문법에 관심이 많아, JDK 1.5부터 22까지 주요 기능들에 대해 정리 중이며, 계속해서 내용을 업데이트 중입니다.

Framework.

- **Spring Boot 3.x & Spring Cloud 2023.x & Spring Data JPA**
- **JUnit5 & Rest Assured & Mockito & Autoparams & Jacoco**
 - 단위테스트 및 통합테스트를 작성할 수 있고, Jacoco를 이용한 테스트 커버리지를 측정할 수 있습니다.

Performance Test.

- nGrinder & Apache JMeter & Apache Bench(ab : Apache HTTP server benchmarking tool)
 - nGrinder를 이용한 부하테스트 환경을 구축하고, 최대 수용 가능 트래픽을 측정하고 있습니다.

Infrastructure.

- AWS EC2, RDS, S3, CloudWatch, SES
- Pinpoint, Grafana, Prometheus, n8n
 - Pinpoint v2.2.2(JDK 1.8)에서 v3.0.0(JDK 17)로 재구축한 경험이 있습니다.

Database & ETC.

- MySQL, Redis
- Docker, Nginx

Collaboration.

- Git, Jira, Slack, Notion, IntelliJ, YouTrack, Asana, Writerside, Postman, Bruno

Education.

2018. 03
~ 2024. 02

한밭대학교 정보통신공학과

- 학점 4.0 / 4.5
- Java, Database, Network, Linux System, OS, Data Structure 수강