



깊이있는 개발자 윤상일입니다

새로운 걸 만드는 것, 문제가 있거나 개선점이 있을 때 Deep-dive하는 것을 좋아합니다.
다양한 경험을 통해 서버 개발자로 확신을 가지고 성장해 나가고자 합니다.

프로필

1995년 5월 21일
서울특별시 송파구 문정동 새말로 8길 12
문정건영아파트 102동 1508호

Phone. +82 010-9790-8783
E-mail. sangilyoon.dev@gmail.com

github. github.com/sangily
blog. meoru-tech.tistory.com

스킬셋

최적화 개선 노력	● ● ● ● ●
학습능력	● ● ● ● ●
Python	● ● ● ● ●
MySQL/MariaDB	● ● ● ● ●
Flutter	● ● ● ● ●
Java/Kotlin	● ● ● ● ●
Go	● ● ● ● ●
Kubernetes	● ● ● ● ●
Redis	● ● ● ● ●

학력사항

인하대학교 전기공학과
@2021 졸업
중앙대학교 사범대학 부속고등학교
@2014 졸업

경험사항

2021 삼성청년SW아카데미(SSAFY)
구미캠퍼스 전공반(모바일 - Kotlin)
2020 42seoul
본과교육생(C, Linux)

경력사항

2021.11 - 2022.02 인턴_당근마켓
플랫폼 서버개발실 계정서버팀
2022.07 - (재직중) TL_SK하이닉스
DRAM Design Verification 품질검증팀

경험

인터페이스 작성을 통한 TDD 및 코드 리팩토링 기여 - 당근마켓

@ 문제상황

- Mocking 작성 및 Unit Test 작성이 어려운 로직 존재
- 서버 shutdown시 DB connection 및 controller close 되지 못하고 exit

@ 문제원인파악

- Golang 특성상 인터페이스 정의가 중요한데, 적절히 관리되지 못하고 있었음

@ Action

- 인터페이스 재정의
 - Mocking model 활용도 증가
 - 외부 connection을 사용하는 인터페이스마다 shutdown 정의
- Unit Test가 필요한 Bundled Logic 또는 재사용성 높은 Logic 분리

@ 성과

- Test Coverage 증가 : Unit Test가 touch하지 못하고 있던 약 5개 함수 gotest 신규 작성
- Graceful Shutdown 구현 : 자원 누수 예방 및 Grafana 관련 Error log 80%감소

@ 어려움

- 여전히 남아있는 connection Error log 20% => SRE팀과 소통하여 해결
- 짧은 인턴 기간 내에 인터페이스 정의를 위해 전체 로직 파악 필요

언어 전환을 통한 개발효율성 및 데이터 가시성 증대 - SK하이닉스

@ 문제상황

- 빈번히 사용되는, 1000줄이 넘는 C++ 데이터처리 프로그램 존재(이하 A)
- 내부 로직 에러 개선, 기능 추가 및 테스트가 매우 어려움

@ 문제원인파악

- 컴파일 언어인 C++ 내부의 데이터 관찰이 아주 어려움

@ 해결방안 구상

- (X) A 프로그램을 나눠서 중간산출물로 데이터를 관찰한다.
 - 중간산출물 io 및 storage 비용이 매우 큼
- (X) 데이터를 쪼갬다. - 타사(Synopsys) 포맷의 암호화 파일이라 불가능
- (O) 스크립트 언어인 파이썬으로 A 프로그램을 변경한다.

@ Action

- A 프로그램에 타사 c++ api에 종속성 존재, 해당 api를 CFFI를 활용해 pythonize
- A 프로그램의 전체 로직을 파악하여 파이썬으로 Migration
- Class 설계

@ 성과

- 실시간으로 데이터 관측 가능
 - A 프로그램 내부 로직 에러 다수 발견 및 로직 개선 (의사예러 90% 이상 개선)
 - 기능 추가가 용이해져 추가 로직 다수 개발 (신규과제 3건 발굴)
- A 프로그램 확장성 증가
 - 파이썬 데이터프레임으로 활용할 수 있게 됨. 팀 신규과제 발굴에 기여 (1건)

@ 어려움

- 언어 자체의 성능상 런타임 증가 => 이후 프로세스의 최적화를 통해 오히려 전체 업무 속도 증가

개발 협업 문화 정착 - SK하이닉스

@ 문제상황

- 파트 내에서 개발, 관리되는 프로그램만 100개가 넘음
- 프로그램이 개발될 때마다 _r1.py, _r2.py, _final.py, _test.py, _230724.py... 등 버전이 분기됨
- 여러 프로그램에 여러 이슈 문의 => 이슈 관리가 전혀 이루어지지 않음

@ 문제원인파악

- SW 개발 협업 문화 및 톨에 대한 인식과 관심 자체의 부재

@ Action

팀 80여명, 파트 12명 중 개발자로 입사한 인원이 저 혼자였습니다.

- Jira 셋업 및 도입
- Confluence에 파트 Onboarding 문서, 회의록, Error&Solution 모음집 등 작성
- Git 셋업 및 도입, Rule 설정 및 전달
- 파트장님 및 파트원들에게 설파, 기획안 작성 및 전달

@ 성과

- 기존 PPT로 진행되던 주간회의가 현재 Jira로 진행됨
- 기존에 report되고 업무에 밀려 잊혀지곤 하던 이슈들이 Backlog로 관리됨
 - 장기적 파트 개발효율 지표 증가
- 기존에 직접전달식, 또는 일부 인원만 갖고 있는 ppt 자료로 이뤄지던 지식의 전달이 공용문서로 기록됨
- 버전 관리 도입
 - Release Version 관리를 통해 개발효율성 증대
 - 문제 발생시 신속한 수정이력 파악 및 hotfix 대처가 가능해짐

@ 어려움

- 기존 방식에서 잘 바뀌려고 하지 않는 사람들 => 지속적 노력 필요
- 산업보안으로 인해 폐쇄망 공용 Unix서버에서 모든 조직원이 공동작업하는 환경
 - Git Branch 전환시 내 branch만 변하는게 아니라 모든 사람의 branch가 변함
 - Branch 사용 불가능 및 코드저장소 불필요 => commit으로만 관리(SVN 방식)

RDBMS 도입 - SK하이닉스

@ 문제상황

- 수백TB의 데이터를 csv나 pkl로 추출 => 유저는 해당 파일을 load해서 사용
- GB단위의 Dataframe pkl파일은 load하는 데에만 수십분 소요
- 데이터 파편화

@ 문제원인파악

- RDBMS 가용 인원의 부재

@ Action

- RDBMS 서버 권한을 가진 DT 부서와 협업하여 스토리지 확보 및 테이블 설계 (MariaDB)
- Sqlalchemy API 작성

@ 성과

- Target Data Load시간 **최대 99% 이상 감소 (20분 -> 10초 이내)**
- 적절한 Normalization을 통해 데이터 사이즈 **약 40% 감소 (300TB -> 170TB)**
- 데이터 중앙 집중화 => 타 부서(DT) **신규 과제 발굴 기여(3건)**

@ 어려움

- 기존 프로그램들과의 호환을 위해 csv 및 pkl도 추출하고, Data도 upload하는 이중작업 발생

프로젝트

: 커피타임 (1인 프로젝트, 개발 中)

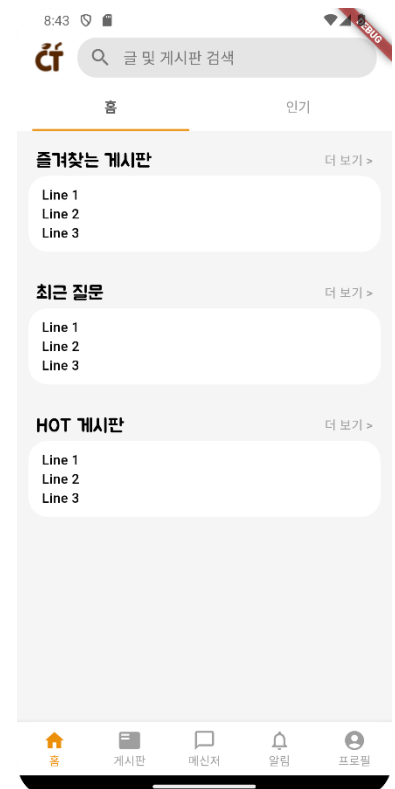
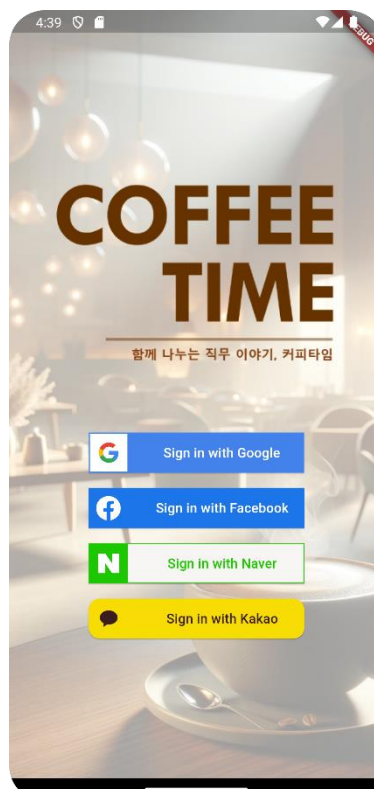
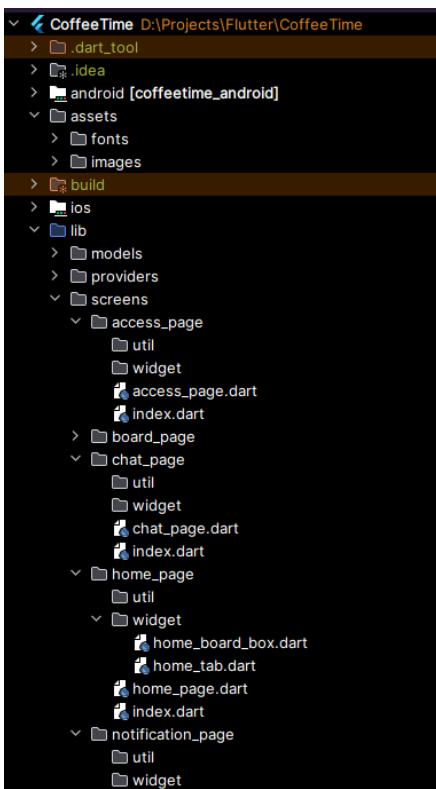
<https://github.com/the-coffeetime>

COFFEE TIME

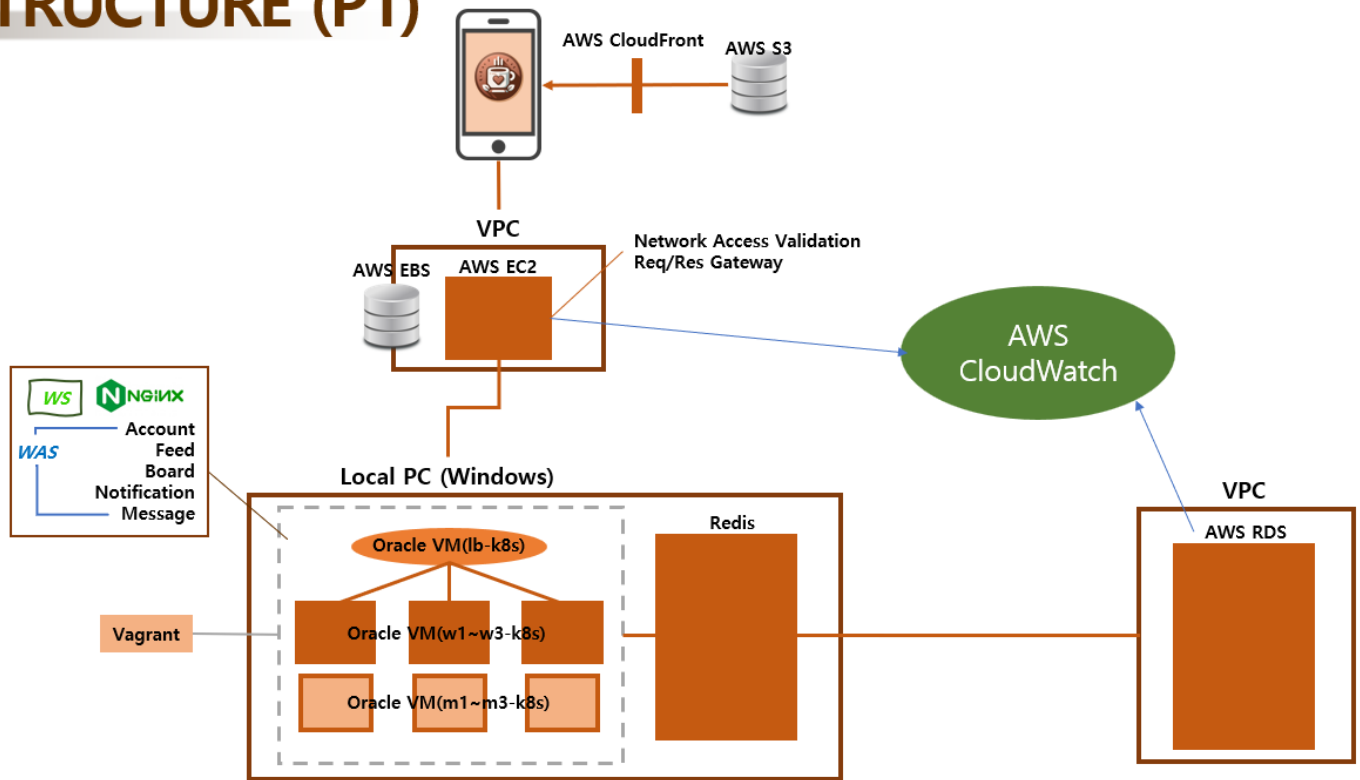


함께 나누는 직무 이야기, 커피타임

커피타임은 직무별로 다양한 이야기를 가볍게 나누는 공간입니다.
각 직무별로 주니어, 시니어, 취업준비생 등 폭넓은 경력의 사람들이 어우러지며,
소박한 인생 이야기, 직무 이야기, 커리어 고민 등 다양한 이야기거리로 소통합니다.
함께 고민하고, 도움을 주고받고, 직무 관련 유용한 정보를 공유하며 함께 성장하는
따뜻한 커뮤니티입니다.

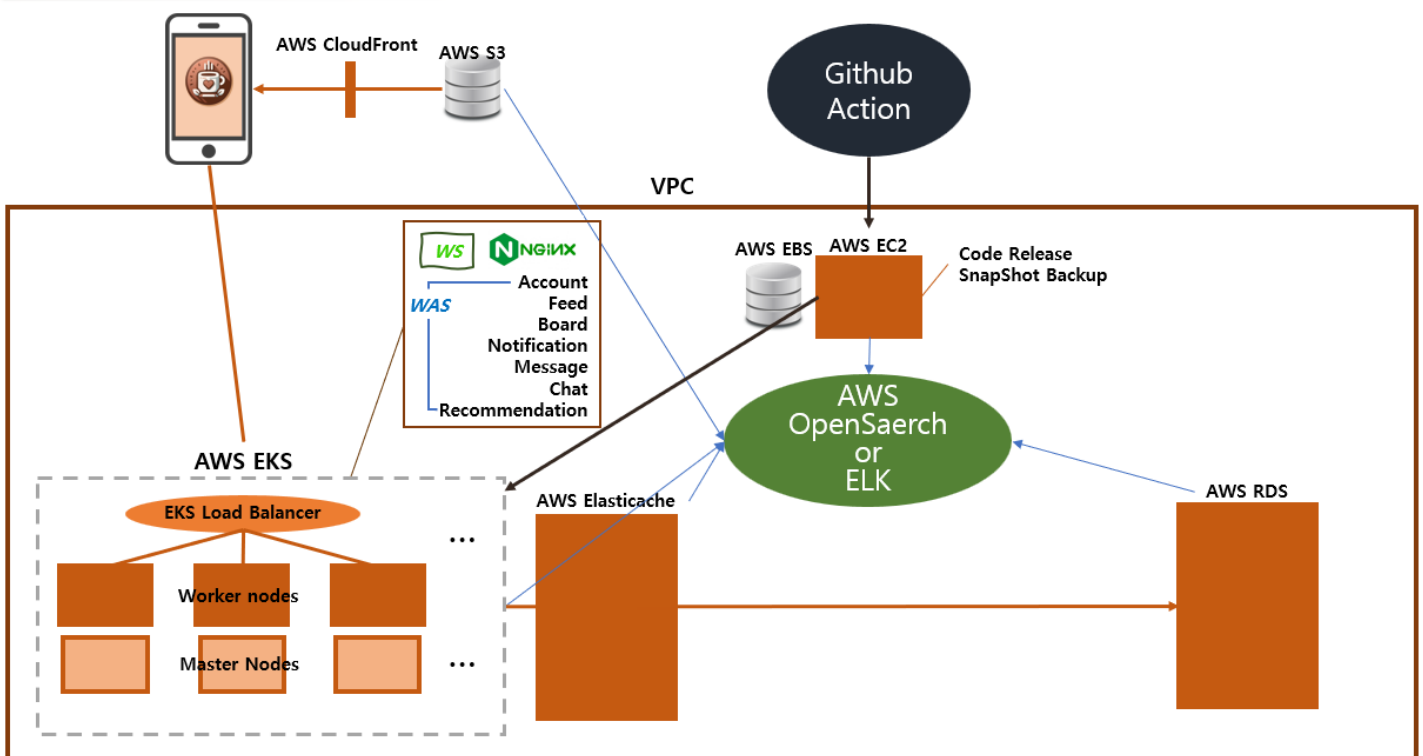


STRUCTURE (P1)

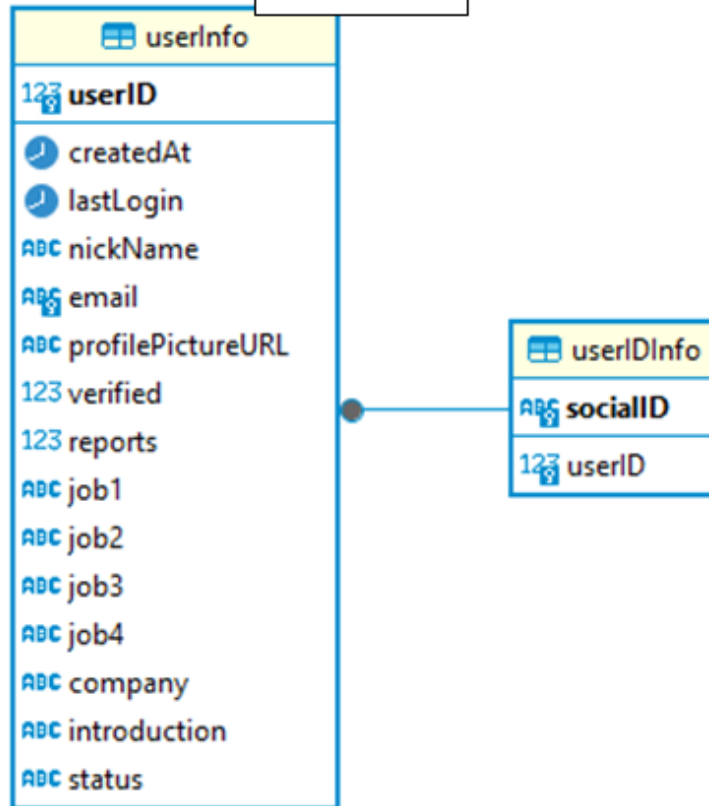


배포 및 유저 확보 이후
(계획)

STRUCTURE (P2)



user



board

