통합 로그 모니터링 서비스

2025-1 AJOU SOFTCON

팀 명 로그

지도교수 고욱

팀 원 윤규민, 김도훈

멘 토 김병륜

개발동기 및 목적

- 멀티 클라우드 환경에서의 로그 관리 어려움 해결
- 개별 플랫폼별 모니터링 접근 불편함 해결 (AWS, GCP 등)
- 분산된 계정 및 리소스 관리 효율화
- 통합 모니터링 및 로그 관리 인터페이스

개발내용

중앙집중식 이벤트 기반 서버 관리 시스템 구축

- 인스턴스 별 로그를 실시간 수집 및 중앙 서버 통합 관리
- 웹 기반 대시보드를 통한 직관적이고 효율적인 로그 데이터 시각화
- 로그 데이터 실시간 모니터링 및 관리 효율성 극대화

통합 관리 인터페이스

- 다중 프로젝트를 하나의 대시보드에서 통합 모니터링 할 수 있는 중앙 집중형 관리 인터페이스 구현
- 서비스별 로그 현황을 직관적으로 파악할 수 있는 UX 설계

사용자 맞춤 로그 설정 지원

- 윈도우・리눅스 등 크로스 플랫폼 로그 수집 설정 지원
- 멀티라인 처리, JSON/PlainText 로그 등 다양한 포멧 수집 및 필터링 기능 제공

로그 수집 Agent 제공

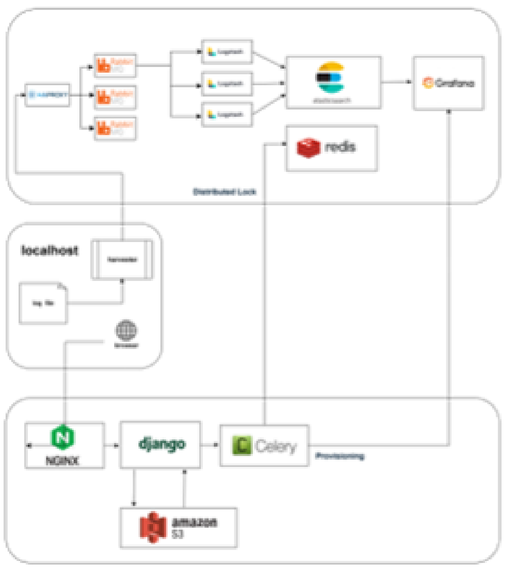
- 사용자가 입력한 설정에 맞는 agent config 및 설정을 동적으로 생성
- 모든 설정을 담은 agent 다운로드 및 실행가능한 스크립트 제공

모바일 반응형 대응 UI

- 모바일, 태블릿, PC 환경에 최적화된 반응형 대시보드 설계
- 운영 장소에 구애받지 않는 실시간 모니터링 제공

주요기술

시스템 아키텍처



[웹 백엔드]

- Django Rest Framework를 이용한 Clean Architecture 기반 백엔드를 구성함으로서 의존성 최소화

[데이터 파이프라인]

- RabbitMQ 를 이용한 이벤트 브로커 구성
- 전처리로 Log stash, 인덱스 저장으로 Elastic search 사용

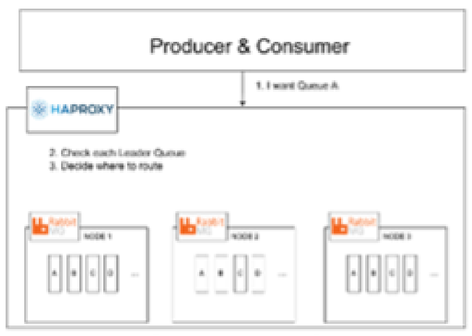
[사용자 UI]

- React를 통한 전반적인 웹 프론트 구성 및 Grafana를 통한 사용자 대시보드 동적 구성

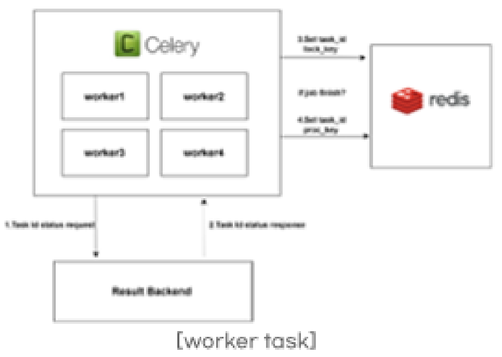
[로그 수집 Agent]

- File, Metric beats를 이용한 로그 수집 에이전트 구성

데이터 파이프라인 및 워커 프로비저닝



[Haproxy & MQ]



[worker task]

[데이터 파이프라인]

- Raft 알고리즘 기반 3노드 클러스터로 구성된 고가용성 시스템 제공
- quorum queue를 도입하여 시스템 다운 시 메모리 복구 및 상태 추적 제공
- Haproxy를 이용하여 노드 장애시 리더 재선정 이후 적절한 트래픽 라우팅 제공

[워커 프로비저닝]

- Celery 기반 병렬 비동기 프로비저닝을 통한 RPS 및 TPS 개선
- Celery chain을 이용한 Task 순차 보장 로직 설계
- Redis를 이용한 분산락 및 Task status 체크를 통한 중복처리 방어 로직 설계
- 워커 다운, 에러 시 retry 및 requeueing 로직 설계를 통한 At least once delivery 보장

결과 및 분석

[운영 효율성 향상]

- 모든 로그를 단일 서비스에서 실시간으로 관리 가능
- 장애 발생 시 즉각적인 로그 필터링 및 문제 분석으로 장애 대응 속도향상

[머신러닝 연계를 통한 분석 기능 확장 가능성 확보]

- Elasticsearch 기반의 이상 탐지 모델과 연동 가능
- 향후 로그 데이터 패턴 학습을 통한 자동 경고 시스템 구축 가능성 확보

[유연한 확장성 제공]

- 멀티 계정, 멀티 서비스 환경에 유연하게 대응 가능하도록 설계됨

오픈소스 URL

https://github.com/Log-Central

Github

시연 영상