



안상혁님을 응원해보세요!

🌿 INFCON 참여 # 지금 만족하고 있어요

안상혁

Server Developer

기술 스택

Spring, JPA, Kafka, Kubernetes, Python, ksqldb, Java

경력

라인플러스

Message Server Developer | Message Server Gateway | 재직 중
2022.07. ~ 재직 중 (2년 5개월)

주요 업무

- KSETL 프로젝트 (<https://engineering.linecorp.com/ko/blog/quickly-build-kafka-stream-etl-system-with-ksetl>)
 - Kafka에 수십 GB/초의 속도로 공급되는 메세지들을 실시간 ETL을 통해 real-time ML용 Output으로 공급하는 ksqldb, kafka streams, kubernetes 기반 데이터 플랫폼을 구축하는 프로젝트.
 - KSETL 개발 및 운영, 모니터링 고도화 작업, 문서화 작업, 장애 대응.
- LINE Message API Gateway
 - LINE 메신저의 Message API Gateway 서버 개발 및 운영. (API 호출 > 1MRps)
 - API Gateway 서버

추천해요 🍷

제안하기 ✉

댓글 💬

- API Gateway의 보조 컴포넌트용 k8s cluster 관리, High Availability 및 Observability 확보, DR 환경 구축.
- argoCD 도입을 통한 Gitops 문화 확립.

프로젝트

API Gateway

라인플러스

2023.12. ~ 진행 중

LINE Messaging Gateway Server 개발, 유지보수, 운영

- LINE의 Messaging Gateway 서버 개발 / 유지보수 업무 진행. (MAU 2억, API 호출 횟수 > 1M/s를 안정적으로 서비스 할 수 있는 서버 개발). 분산 클러스터 환경 및 비동기 환경에서 동작하는 서버 개발.
- 부하 테스트 진행 및 SMP 설정.

보조 Component 개발, 유지보수, 운영

- 100+ 라인 서버 개발자가 구독하는 얼럿 전파 서버 개발 및 운영.
- `fluent bit` 커스텀 plug-in 개발 / 유지 보수 / 기능 고도화.

Kubernetes 관련 작업

- On-premises 환경의 `kubernetes` 클러스터 관리
 - On-premises 환경의 `legacy k8s cluster` 를 최신 버전 동기화 작업.
 - kubernetes cluster의 Observability 확보/고도화. (prometheus, Istio, kiali 도입 및 설정 / 유지보수)
 - Multi IDC를 고려한 secret 관리 체계 도입.
 - Gitops 기반의 kubernetes 클러스터 관리 도입 및 고도화. (argoCD 도입 및 설정 / 유지보수)
 - `kubernetes` cluster 이중화 작업으로 High Availability 확보.
 - istio 도입을 통해 on-premises `kubernetes` cluster의 DR Simulation 환경 구축

KSETL

라인플러스

2022.09. ~ 2023.10.

KSETL

댓글 💬

- 모니터링 고도화 및 성능 개선
 - `Kafka` , `Kafka Streams` , `ksqlDB` 메트릭 추가 조사.
 - `JMX Exporter` 의 `CPU usage` 증가 원인 조사 및 개선. (CPU usage `60%` 개선)
 - KSETL Application별 대시보드 생성하는 Web UI 서버 제작 및 배포(`Spring MVC` , `tailwind-css` , `java`)
- 운영 기여
 - k8s 초기화 컨테이너 제작하여, KSETL Helm chart 배포 시 발생하는 Pod Initialized Fail 문제 해결.
 - `ksqlDB` → `Redis` 로 잘못 공급된 데이터를 제거하는 파이썬 기반 cli 툴 개발. 수십 GB 데이터 불러와 필터링 + 제거하는 작업을 분 단위로 처리.
 - `argoCD` 를 도입 및 설정하여 gitops 기반 관리체계 구축.
 - Jenkins를 이용해 CI/CD 환경의 60% 가량 구축 및 유지 보수.
- KSETL Prototype 환경 구성 (`docker` , `docker-compose` , `Spring` , `tailwind-css` , `Java`)
 - 사용자마다 `ksqlDB` 프로토타입 환경을 스크립트 하나로 구축해서 사용할 수 있도록 컨테이너 환경 및 필요 어플리케이션 개발.
 - UI 기반 설정으로 원하는 형태의 Dummy 메시지를 `Kafka` 토픽으로 제공할 수 있는 `FakerProducer` 개발.
- `ksqlDB` Internal (`ksqlDB` , `Kafka Streams` , `Grafana` , `Prometheus`)
 - `ksqlDB` 의 `record format` migration 작업을 통해 `G1 GC` 의 `Concurrent Cycle` 발생 빈도 12배 개선하여 `Concurrent Cycle` 에 의한 `Stop the world` 개선, `Network I/O` , `Network Transmit and Recieve Byte` 감소.
 - Pod 별 Task 의 `Topic Partition` 균형 분할을 위한 `TaskAssignor` 개발 기여. `TaskAssignor` 의 성능 테스트로 의사결정에 기여.
 - `ksqlDB` 의 `Consumer Assign` , `TaskAssignor` 코드 분석 작업 진행. 이를 통해 리밸런싱 시간 최적화(실험치 기준 최대 90% 개선) 및 가용한 Query 갯수 증가(최소 10%)

포트폴리오

URL

[백준](#)



[블로그](#)

댓글



외국어

영어

일상 회화 가능

일본어

일상 회화 가능

자기소개

안녕하세요. 저는 이런 사람입니다.

1. 새로운 기술을 공부하는데 거리낌이 없고 필요하다면 도입하는데 망설이지 않습니다.
2. 새로운 컴포넌트를 추가해야한다면 기존에 존재하는 기술로 해결할 수는 없는지를 따져보고 도입하고자 합니다.
3. 할 수 있는 범위 내에서 성능 테스트를 한 후, 근거를 확보하고 이야기 하는 것을 좋아합니다.
4. 부족한 지식들은 개발 서적과 공식 문서 등을 이용해 꾸준히 보충하고 있습니다.
5. 몇몇 오픈소스에 기여한 경험이 있습니다.

a. spring/spring-kafka : <https://github.com/spring-projects/spring-kafka/pull/3212>

b. kiali.io/kiali : <https://github.com/kiali/kiali.io/pull/765>

c. micrometer-metrics/context-propagation : <https://github.com/micrometer-metrics/context-propagation/pull/223>

d. apache/kafka : <https://github.com/apache/kafka/pull/15926>

e. apache/kafka : <https://github.com/apache/kafka/pull/15573>

f. reactor/reactor:core : <https://github.com/reactor/reactor-core/pull/3779>

g. spring/spring-kafka : <https://github.com/spring-projects/spring-kafka/pull/3172>

h. spring/spring-kafka : <https://github.com/spring-projects/spring-kafka/pull/3216>

i. spring/spring-kafka : <https://github.com/spring-projects/spring-kafka/pull/3237>

j. line/armeria : <https://github.com/line/armeria/pull/5628>

k. line/armeria : <https://github.com/line/armeria/pull/5628>

댓글 

스스로 좋은 동료가 되기 위해 모자란 지식을 보충하고, 부드러운 커뮤니케이션을 하기 위해 노력합니다.

Powered by Rallit.