

셀러박스 고도화

☰ 태그	AWS	Crawling	서클플랫폼	업무개선
📅 날짜	@2024년 6월 1일			

서비스

서비스에서 맡고 있는 업무

진행 프로젝트

모든 수집 로직 비 동기화

수집서버 Serverless 로 전환 (AWS Lambda)

단계적 전환을 위한 API Gateway 도입

서비스

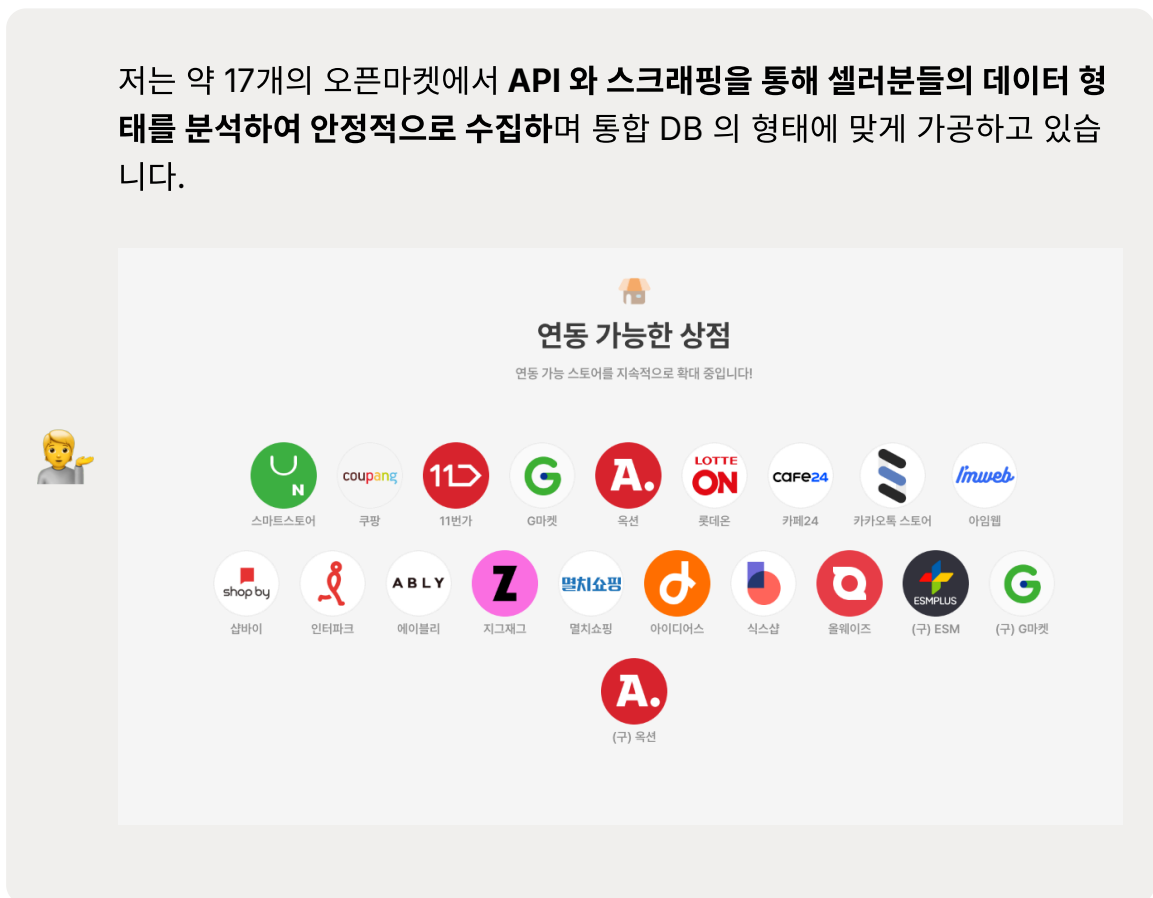
- [서비스 링크](#)
- 제공 서비스
 - 여러 오픈마켓에서 판매하는 셀러분들에게 주문, 정산, 리뷰, 문의 같은 데이터를 한 곳에 모아 관리할 수 있도록 만든 서비스입니다.



서비스에서 맡고 있는 업무

1. 데이터 스크래핑

저는 약 17개의 오픈마켓에서 **API**와 스크래핑을 통해 셀러분들의 데이터 형태를 분석하여 안정적으로 수집하며 통합 DB의 형태에 맞게 가공하고 있습니다.

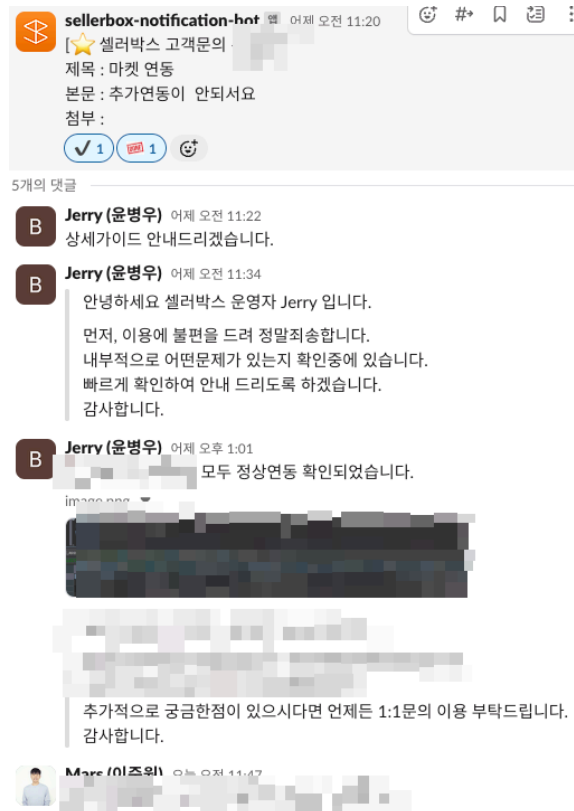


데이터의 빠른 수집주기를 가져가기 위해 **API 방식의 경우 RPS**를 최대 활용하기 위한 비동기 코드 구성과

스크래핑 방식의 경우 로그인 방식 분석 후 **Cookie 발급**을 통해 로그인 및 2차인증을 해결하여 데이터 수집을 진행하고 있습니다.

2. 고객 C/S 담당

내부 정책이나 예상치 못한 에러 또는 고객 요청사항등의 문의를 직접 담당하고 있으며 해당 문제를 직접 해결하거나 도움이 필요할경우는 다른팀과 소통하며 처리하고 있습니다.



3. 지속적인 서비스 개발

오픈마켓 추가 또는 셀러분들에게 신규 기능을 제공하기 위해 새로운 서비스를 프로젝트 단위로 계속 진행하고 있습니다.

진행 프로젝트

모든 수집 로직 비 동기화

- 기존의 수집 방식은 Flask 를 기반으로한 동기방식의 수집 방식이었습니다.
 - 이로 인해 10 분마다 주문수집을 진행해야하는 정책을 만족하기위해 주문 수집기간을 1일로 단축하였습니다.



비동기로 전환한 덕분에 모든 셀러의 7일치 주문을 10분안에 수집할 수 있게되었으며 전일치 주문건에 대해서도 셀러분들께 진행상황을 계속 전달드릴 수 있게 되었습니다.

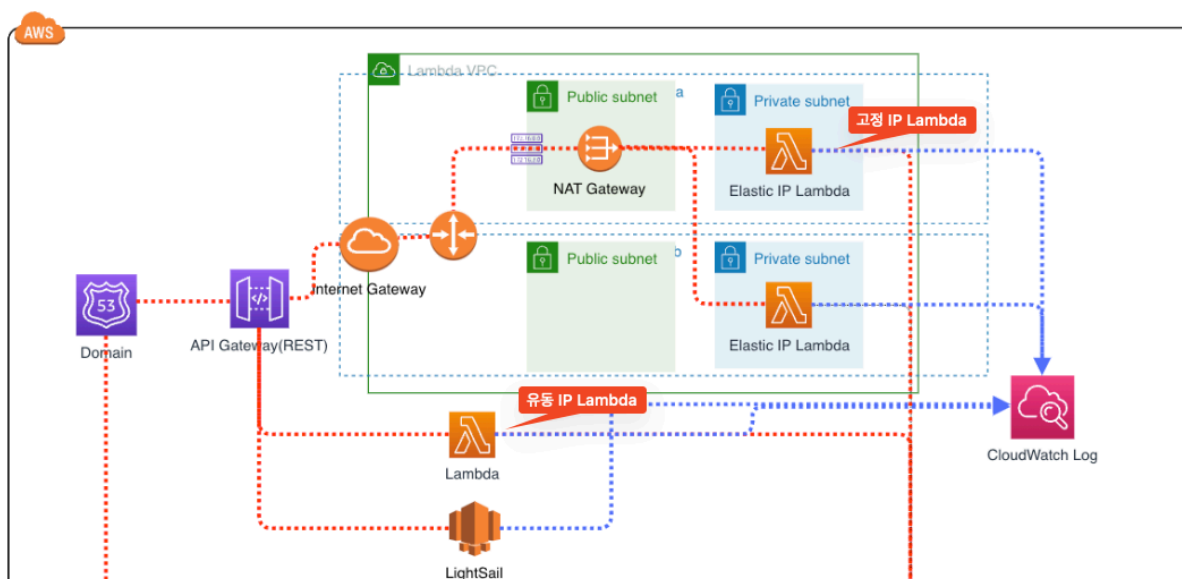
- FastAPI 로 전환 및 비동기 라이브러리 도입



aiohttp 와 aiolimiter 를 활용하여 요청 API 또는 스크래핑 URL 단위에 최대 RPS 를 적용하여 현 상태에서 최대의 수집속도를 적용하였습니다.

- 성과
 - 수집시간 단축으로 10분내로 **1일치 주문수집 → 7일치 주문수집** 변경

수집서버 Serverless 로 전환 (AWS Lambda)



- 수집서버는 **EC2 인스턴스를 여러개 사용**하였으며 **LoadBalancer** 를 앞에 두어 부하를 분산하고 있었습니다.
- 하지만 수집서버는 **10, 30분 단위로 대량 요청을 받는 특징이 있었으며 새벽시간대에는 요청이 없었습니다.**
 - EC2 서버는 부하가 심할때는 레이턴시가 길어졌으며 (심지어는 다운되는경우도 있었습니다.) **요청이 없는 시간대에는 의미없는 비용만 지불**하고 있었습니다.
 - 심지어 **고정 IP 로 운영되어야하는 인스턴스에는 부하가 매우많아** 항상 불안한 상태였습니다.



이에 저는 FastAPI 수집서버를 AWS Lambda 로 변경하여 주기적으로 생기는 부하에 유동적으로 대응할 수 있는 인프라로 변경하였으며

Lambda 에 고정 IP 를 부여할 수 있도록해 IP 화이트리스트를 강제하는 오픈마켓의 데이터도 대량으로 수집할 수 있게되었습니다.



Lambda 특성상 인스턴스 저장소에 로그를 보관하는 것은 불가능하기 때문에 **watchtower 라이브러리를 사용하여 모든 Lambda 의 로그를 하나의 그룹으로 묶어 관리**할 수 있도록 했습니다.

- 대량수집을 위해 Serverless 서버가 있음
- 또한 **Flask** 에서 **FastAPI** 로 한번에 전환할수 없어 단계적으로 전환하고 있는데요
 - 이때 **API** 엔드포인트 단위로 요청 도메인이 변경됩니다.
 - 이럴 때마다 스케줄링 서버에서 엔드포인트 그룹을 계속 수정해줘야하는 불편함이 있습니다.

이 상황을 해결하기 위해 Worker 의 모든 API 를 하나의 API Gateway 로 묶어서 관리하기로 했습니다.



- 모든 스케줄러는 하나의 **API Gateway** 도메인만을 바라보게 되어 관리포인트가 줄어들게 되었습니다.
- **Flask** 에서 **FastAPI** 전환 시 단계적으로 조금씩 변경을 할 수 있게 되었습니다.
 - 해당 과정에서 **Canary** 배포를 사용하여 전환하기전에 미리 테스트를 해볼 수 있어 매우 편리했습니다.