기존환경

기존환경이란 vm/cloud 같이 가상환경에 배포된 모놀리틱한 구조를 가진 환경일 수도 있음

새로운 서비스가 생겨서 인프라를 배포해야하거나 아니면 이미 배포된 인프라를 사용할 때 리스크가 많이 소요됨

만약 이게 IaC로 배포된 환경이라도 버전관리를 주기적으로 해줘야하고 컴퓨팅 용량 확장계획도 늘 대비하고 수립하고 시행해야함

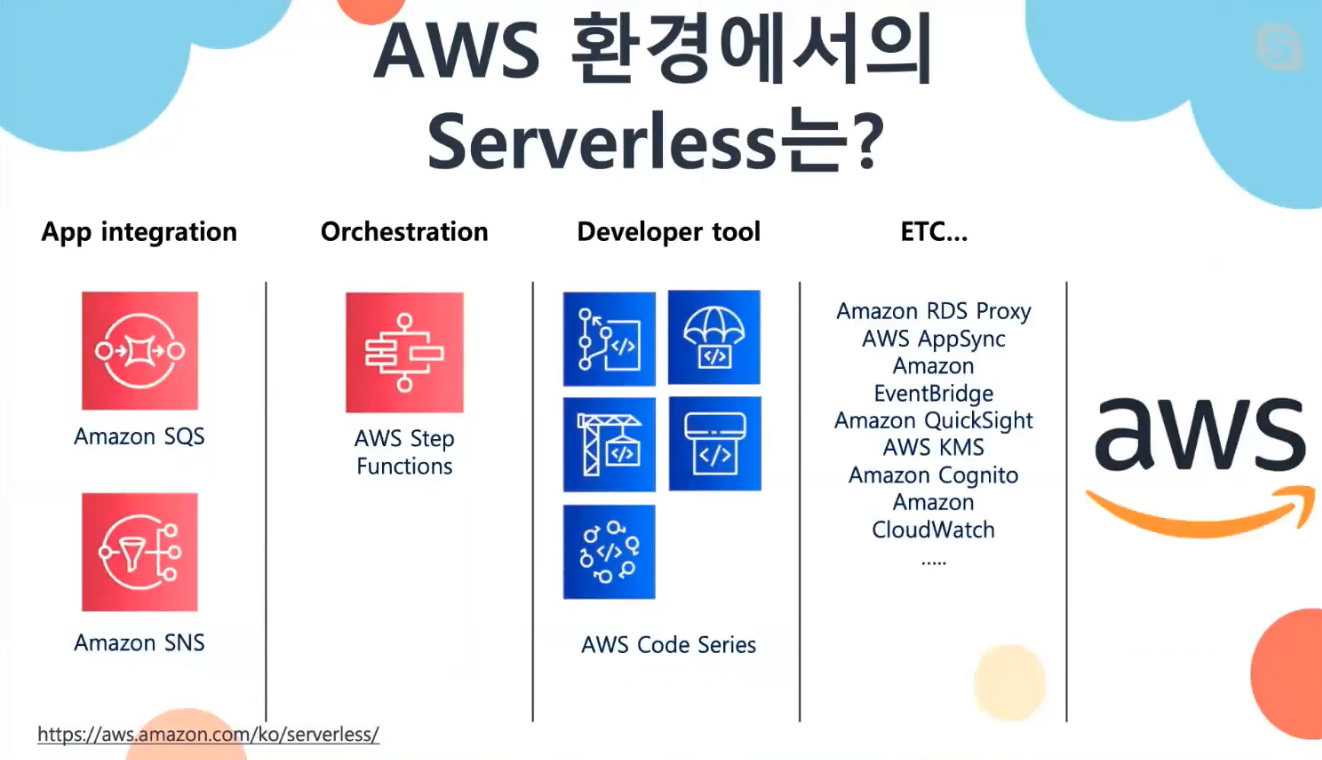
만약에 취약점이 발견되면은 patching을 통해서 주기적으로 관리했어야 했음 따라서 it에서는 한사람이 전지전능하게 뛰어나거나 업무분담을 했어야했음 근데 이게 잘 되지는 않음

근데 aws에서의 서버리스는 관리자가 훨씬 많은 일을 하지 않아도 됨

클라우드 서비스를 제공하는 vendor들이 여러 스펙들을 대비하게 해줌 사용자 대신 관리하게 해줌

그래서 서버리스 환경을 사용하면 리소스들이 많이 줄어들음

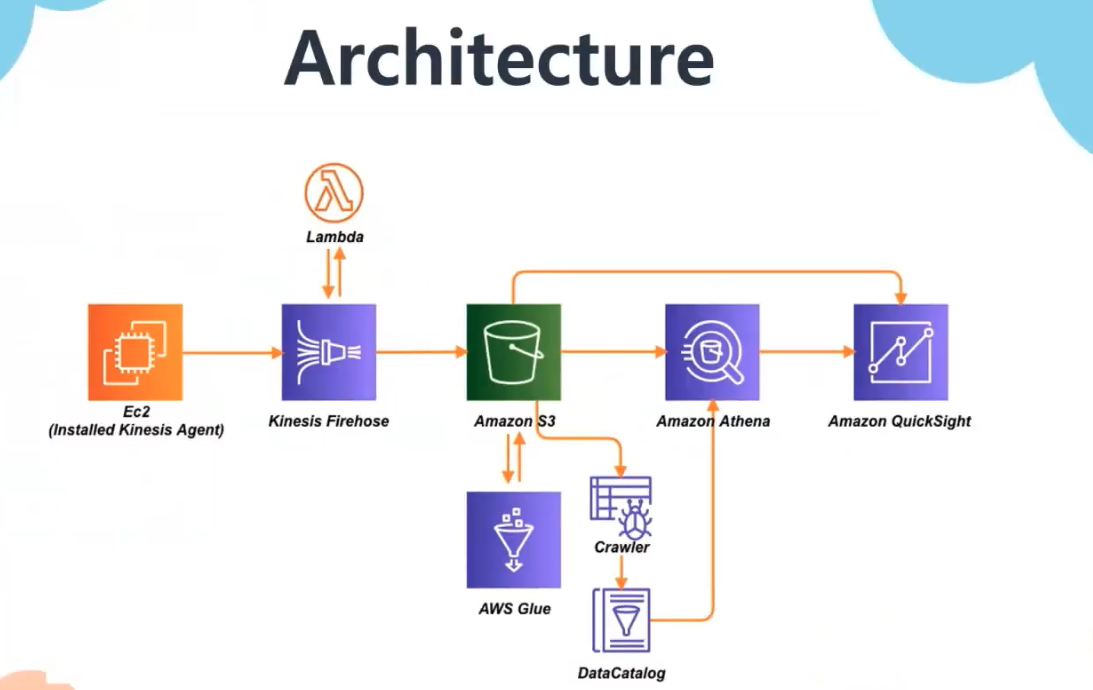
서버환경 구성이나 관리, 운영이나 취약점 패치 같은 것들을 신경 쓸 필요없이 좀 더 빠르게 개발하고 배포하고 상대적으로 신경을 덜 쓰게 됨

그래서 aws에서는 아래 사진처럼 다양한 서비스 환경을 제공하고 있음

이처럼 aws에서는 개발자가 많은 노력을 기울이지 않도록 신경쓰고 있다

이런 서비스들을 활용하려면 레고블럭처럼 흩어져있는 조각들을 잘 조립해서 하나의 서비스로 만들어야함.

오늘의 주제는 aws 서버리스 로 실시간 데이터 분석 플랫폼을 만들자

해당 아키텍쳐는 아래와 같음.

EC2 - 위 아키텍쳐에서 Ec2는 우리의 비지니스와 관련된 서비스가 있다고 가정하자 여기에는 앱이나 웹 또는 로그(스트림성 로그)들 같은 것들을 kinesis agent를 통해서 실시간으로 fireholes라는 서비스로 전달하게 됨 트래픽 이용을 줄이기 위해 vpc endpoint라는 서비스를 사용함

Kenesis Firehose - 데이터 수집단계에서 kenesis라는 서비스를 사용할 수 있음 수집하면서 서버리스 서비스인 Lamda를 통해서 간단한 transform을 실행함

kenesis 서비스에서는 클릭스트림이나 이런 클릭성 데이터들을 받아서 뒷단에 emr 또는 하드클러서 같은데에 보낼 수 있음 그래서 사실 큐인거임

버퍼크기 단위로 gz, zip, snappy 등으로 압축할 수 있고 포맷도 변환할 수 있음

Amazon S3 - 수집된 데이터는 s3라는 storage data에 저장되게 될 것 s3는 향후 데이터 분석을 위한 데이터 저장소(data store)로 계속 사용될 예정임

AWS Glue - detail job을 처리하거나 저장된 데이터에 대해서 스키마를 저장하고 aws의 다른 서비스에서 Catalog를 보여주는 서비스임

여기서 csv 파일로 저장된 데이터를 칵테일(탑테일?) 형태로 변환을 하고 그걸 데이터 카탈로그로 만들어서 그걸 스키마 형태로 다른 서비스에게 쿼리할 수 있게끔 제공을 할 것

Amazon Atena - 아테나라는 서비스는 아마존 row data 형태로 csv 파일로 저장되있고 탑테일 형태로 저장되어 있음

그런 데이터들을 프레스 기반의 데이터 엔진을 통해서 데이터를 따로 노드하지 않고 엔씨(ansi ?) SQL로 따로 로드할 수 있는 상태임

그래서 sv에 저장된 특정 데이터들을 sql 쿼리문으로 쿼리를 해서 결과값을 얻는 것

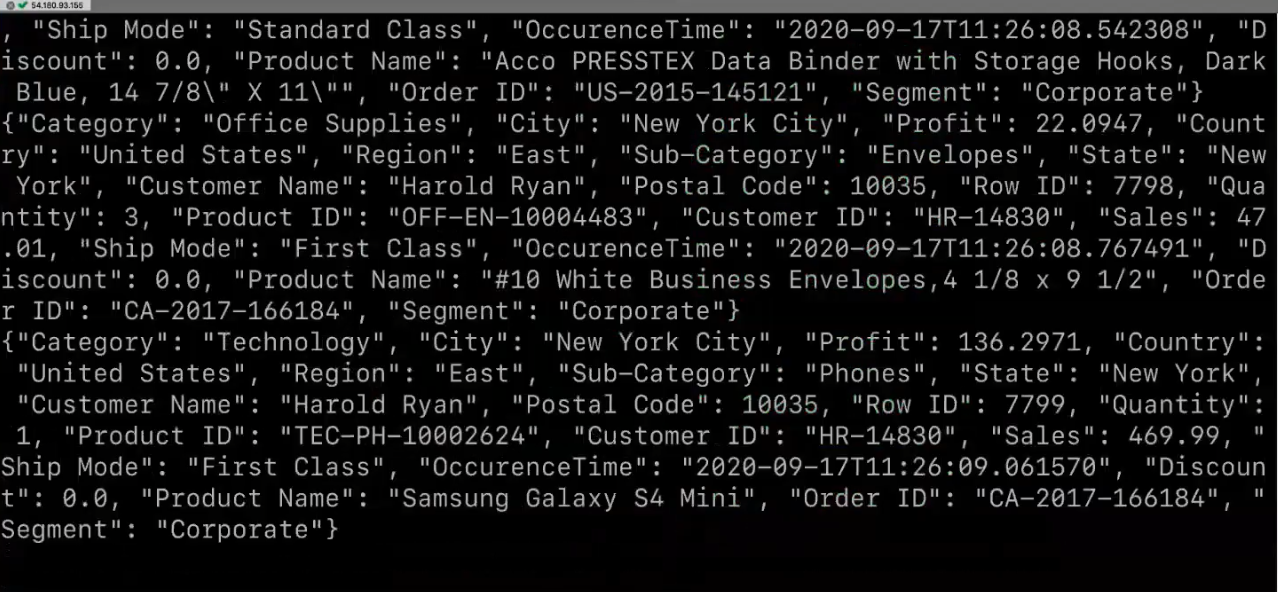
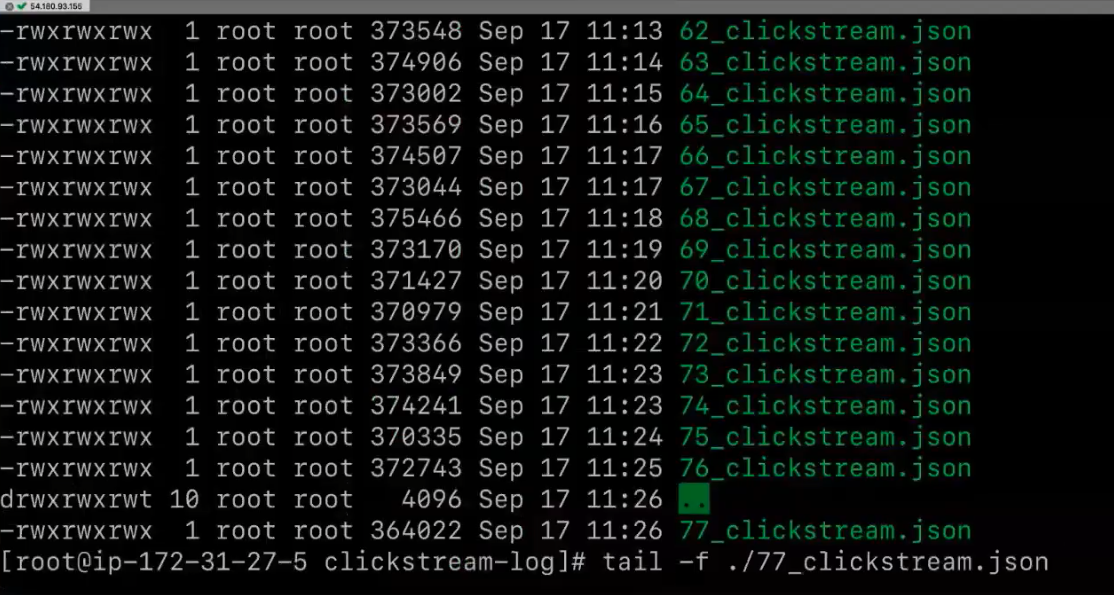
데이터는 스캔한 데이터의 트랩 용량(TB) 비용당 비용이 발생함

가장 심플하게 검색을 할 수 있는 서비스임

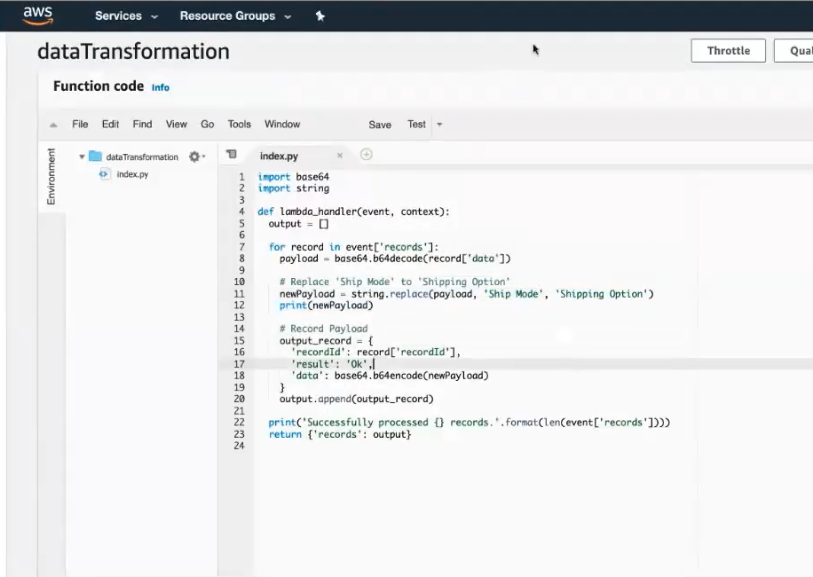
Amazon QuickSight - Amazon QuickSight는 AWS에서 제공하는 Business intelligence tool임 QuickSight를 통해 수집된 데이터를 가시화하고 그래프 형태로 보여줄 수 있음

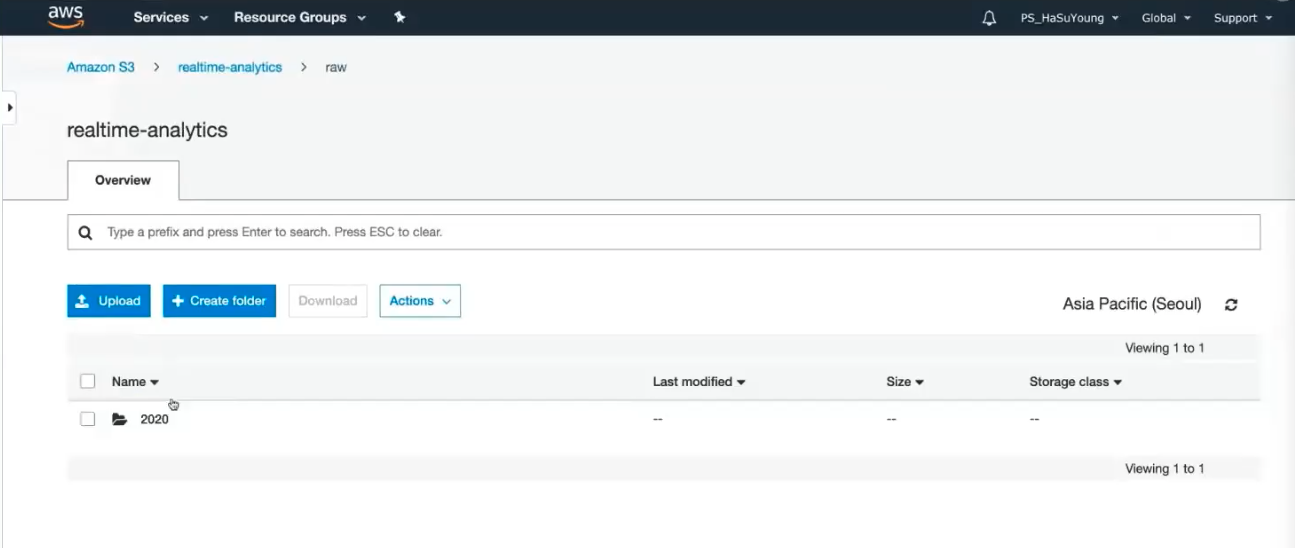
여기까지가 총 아키텍쳐의 흐름임

8분~26분

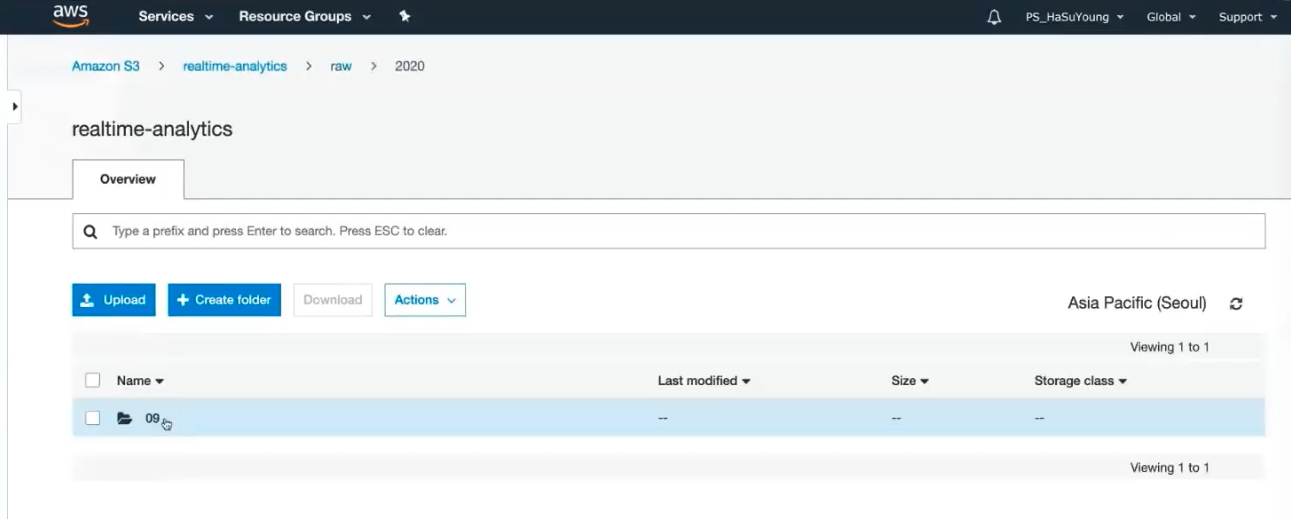
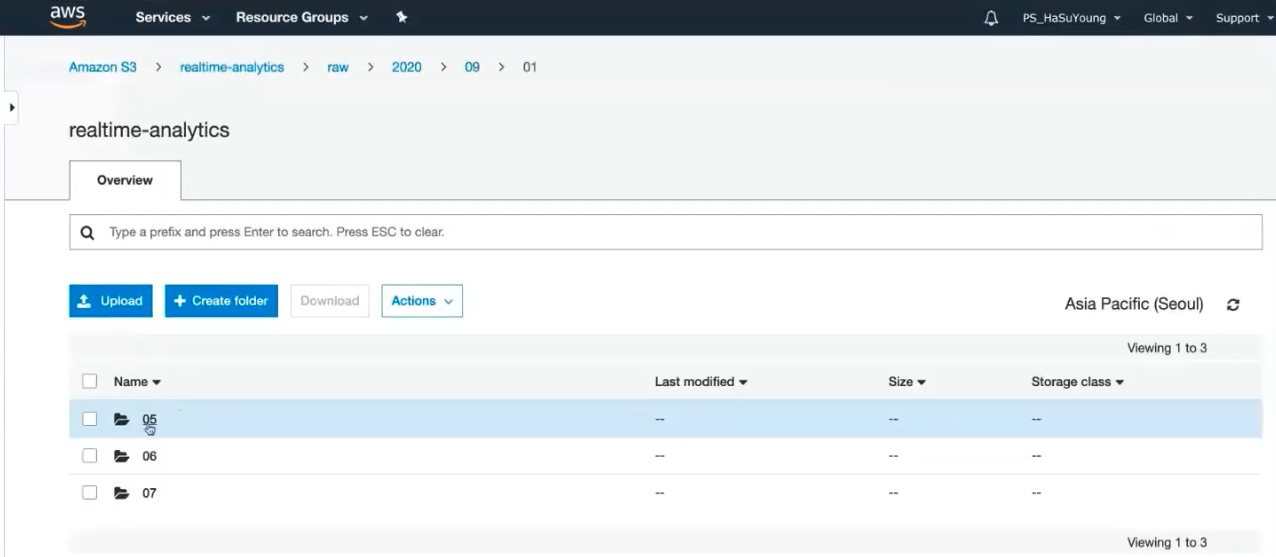
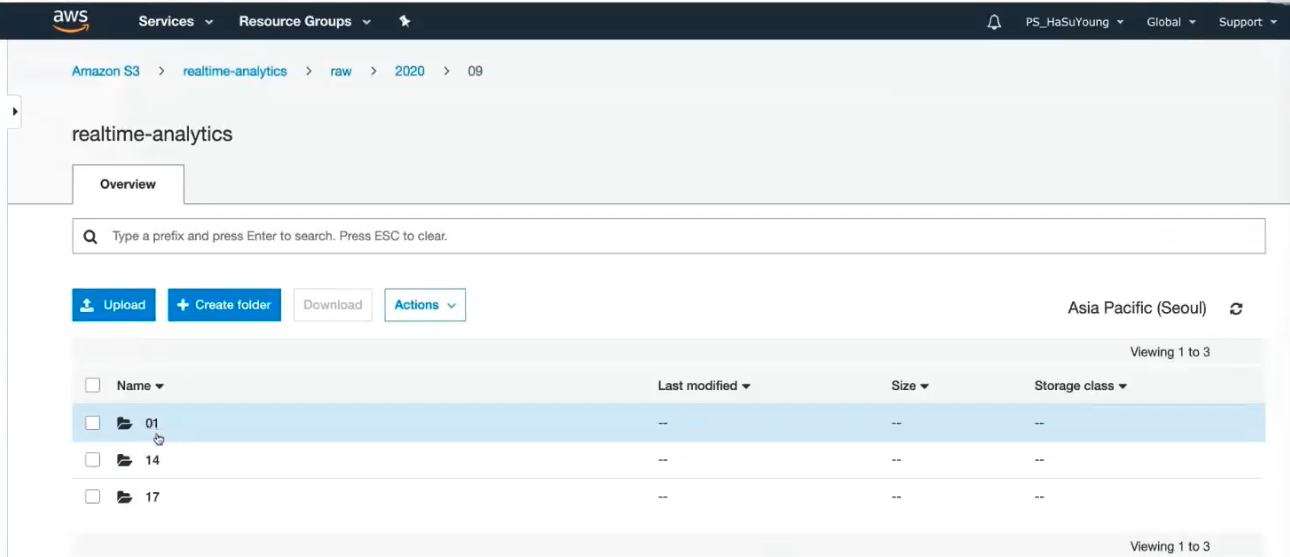


tail을 사용하여 구매 db를 Kenesis로 올림

Kenesis로 들어오는 데이터에 대해서는 지금 람다가 돌고 있어서 간단하게 transform을 한다고 했음. 여기서는 ship mode를 shipping option으로 변환해서 그냥 흘려보내줌

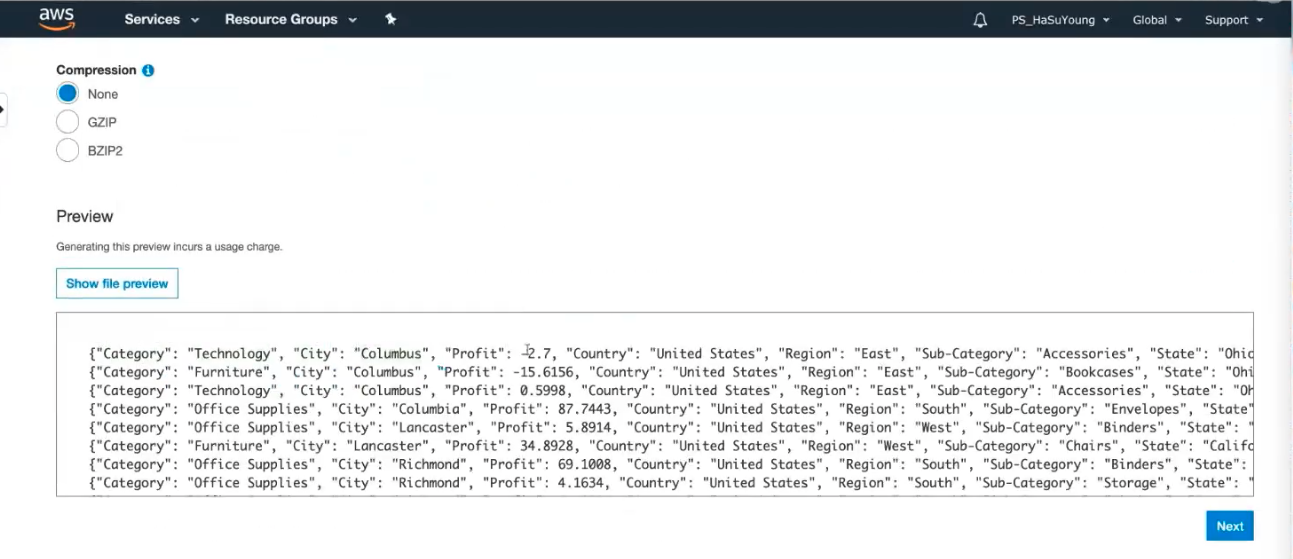


약간의 transform을 거친 데이터는 s3에 row data 형태로 저장하게 됨

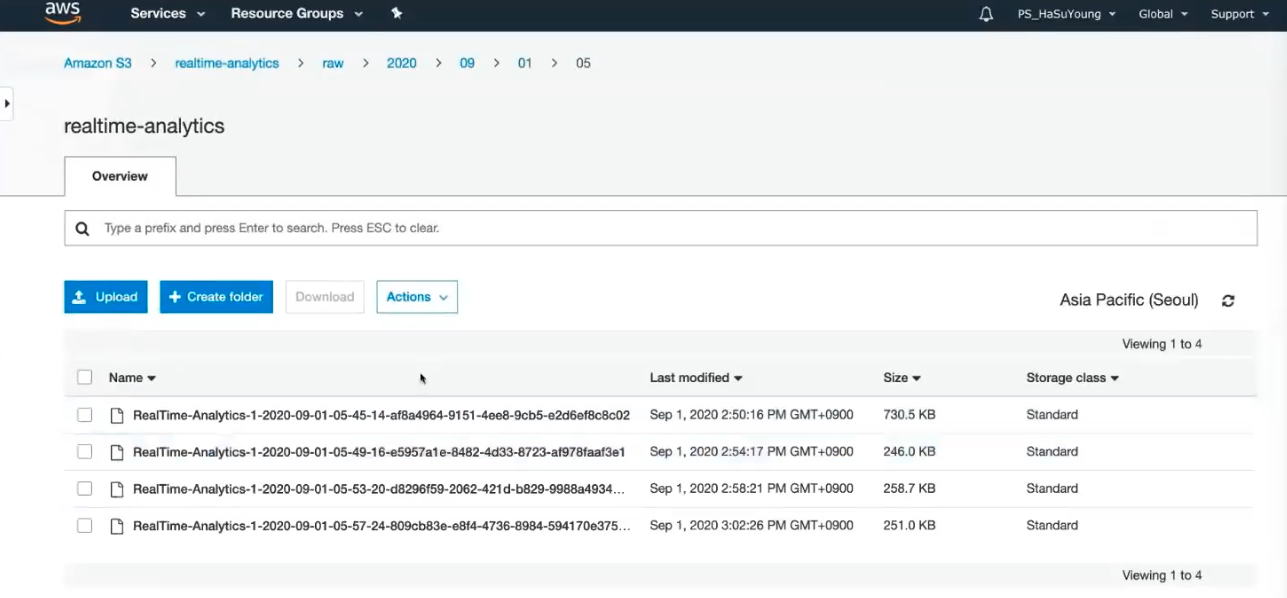
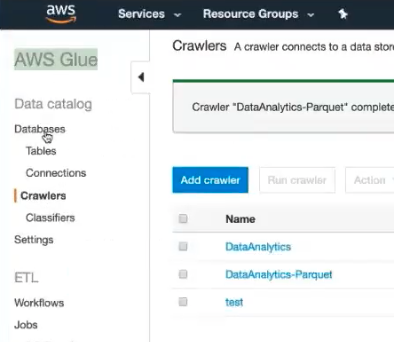
저장될때 firehores에서 자동으로 partitioning을 해줌

연,월,일,시간으로 나눠줌

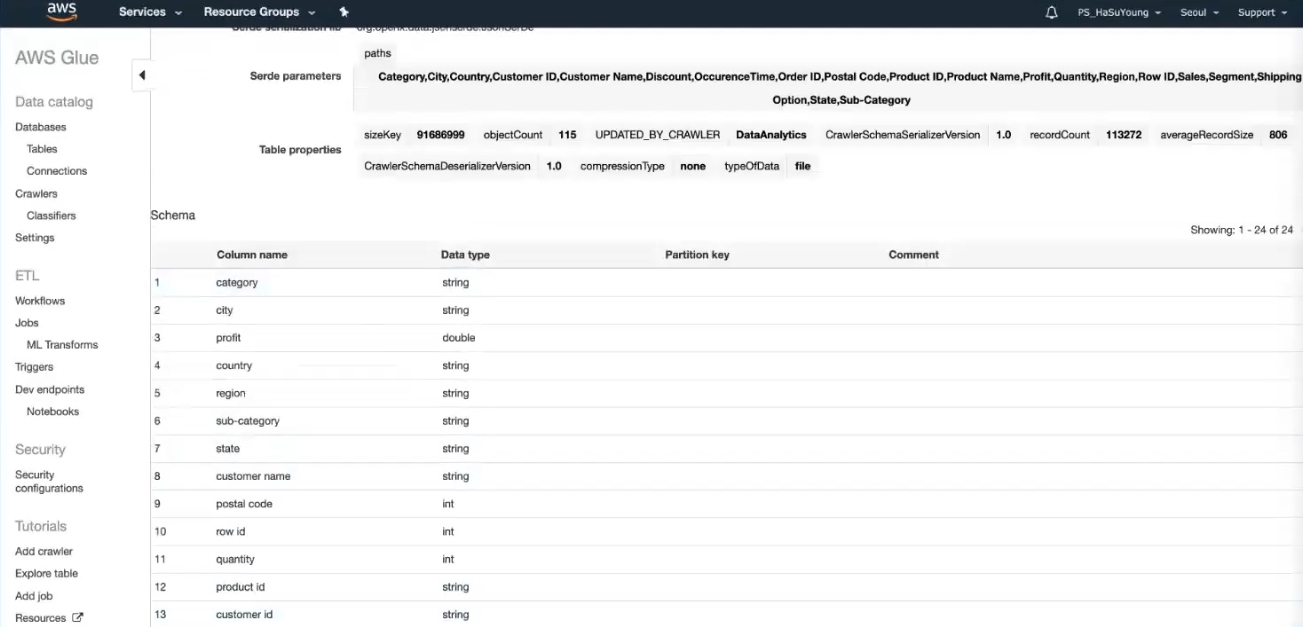
이런 식으로 데이터가 csv파일 형태로 저장됨



파일을 하나 들어가보면 이렇게 어떤 데이터가 들어가있는지 알 수 있음

그럼 이런 csv 파일을 스키마 형태로 어딘가 정의를 해줘야 어딘가에서 쿼리를 하던 가져오던 할것임. 이것에 대해서는 glue라는 서비스를 사용

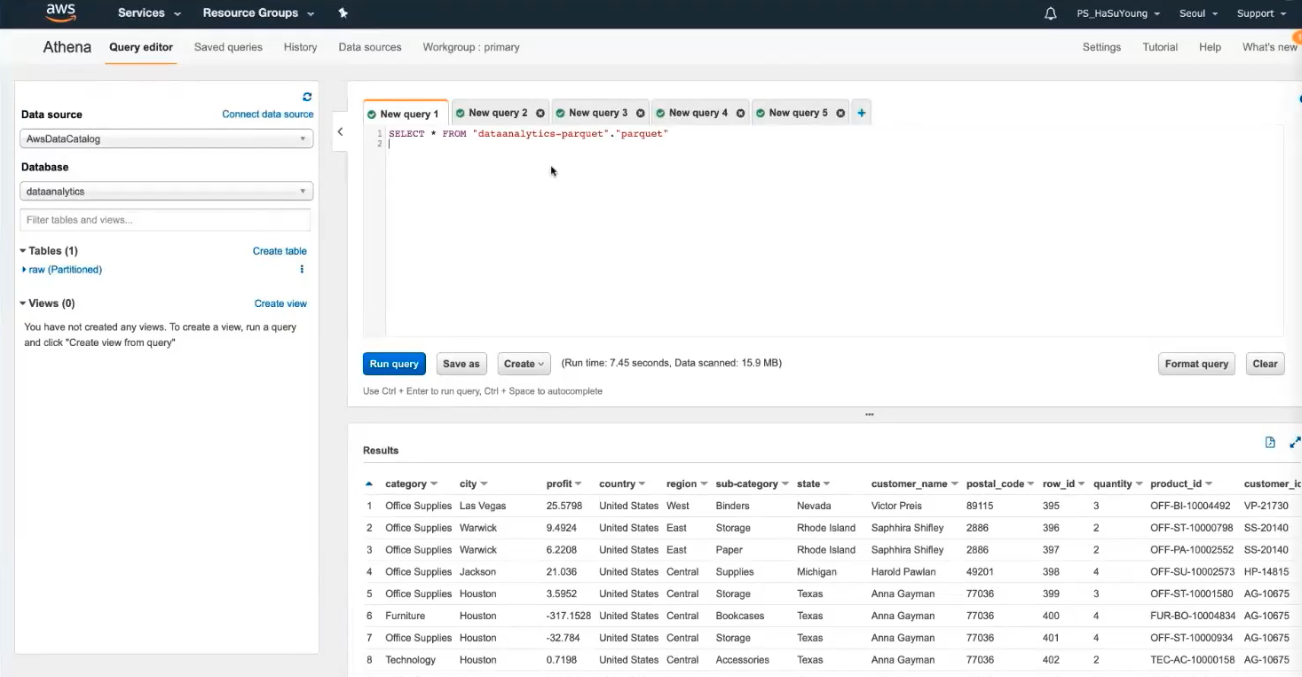
글루에 crawlers라는 서비스란 저장되어있는 csv 데이터 파일을 계속 크롤링하면서 스키마를 업데이트할 것임



이건 모두 자동화되어있음

그래서 이 데이터를 가지고 job을 통해 etl처리를 할 것임

etl은 이 스크립트를 기준으로 etl이 됨



그리고 atena라는 서비스를 통해서 저장된 데이터에 대해서 ansi쿼리로 쿼리할 수 있음 그래서 우리는 두개의 데이터가 있음 row data와 팟케이 형태의 데이터가 있음

그럼 사람들이 데이터를 조회해서 데이터가 잘 들어왔는지 알 수 있어야함 s3에 저장된 데이터를 이런 쿼리 형태로 알 수 있는게 atena 서비스임

parquet으로 저장된 데이터는 스캔을 적은양임

같은 데이터를 저장하고 있음에도 데이터 포맷을 달리함으로써 데이터 스캔의 저장량이 줄음. 이 뜻은 아테나에서 스캔하는 용량만큼 비용을 지불하게 되는데 이런 것들이 비용을 절감하게 할 수 있는 것들임