**2차 멘토링 피드백**

2024/05/18 2팀 김건호

1. **데이터 분석 방향성**

주제를 특정하기보다는 적절히 섞어서 진행하면 좋을 것 같음

R&R을 나눠서 진행할 것

task로 나눌 것인지, 아니면 서비스로 나눌 것인지 생각

WBS를 만들어서 detailed task를 나누고 status, issue 사항도 체크할 것

**3개의 순서로** 가이드라인 제시 받음

1순위로 데이터 마트 구성, Visualization까지의 과정을 완성할 것

* S3 – glue crawler – Data Catalog - Quicksight까지의 과정
* 데이터 분석을 우선 완수할 것. 시각화는 무조건 필수임
* 앞에 과정이 완성됐다고 가정하고 분석부터 진행

2순위로 데이터 수집 방법을 구현할 것

* 기본 인프라에서 데이터를 어떻게 가져올것이냐의 문제를 생각

3순위로 데이터 workflow를 설정할 것

* 워크플로우가 돌아갈 스크립트가 없는데 만드는 것은 의미 없음
* 제일 후순위로 미룰 것

데이터 분석에 집중하고 그에 따라 가상의 시나리오를 만들어서 앞단을 구성할 것

비용적, 성능적 부분이나 사용할 사람들의 스킬셋을 따져야 함.

특히 데이터 같은 경우는 구축보다는 **그들이 다루어야 하기 때문에 고객사의 스킬셋에 맞출 필요성이 있음**. 그래서 그것에 맞는 환경을 구축하는 것이 좋음

* 즉, 상대적으로 시나리오에 관해선 자유로움

원래대로라면 고객이 가지고 있는 데이터를 먼저 조사해야 함.

뽑아낼 수 있는 데이터냐에 집중하고 고객사에게 제공받거나 하는 형식으로 진행이 됨.

하지만, 지금은 교육적인 측면이기 때문에 그것에 집중하지 말고 분석에 집중할 것

일반적으로 MSP에서 말하는 운영이란 서버 이슈 등을 말함.

하지만, **데이터에서 말하는 운영은 개념이 다름**.

일례로 고객이 관리를 못하는 경우가 있음. 데이터 이슈가 있거나, 대시보드 추가 수정, 보고 싶은 인사이트에 대한 수정이 필요할 때가 있음.

이러한 작업이 꾸준히 일어나야 된다면 계속 수정하고 가공할 필요가 있음. 하지만, 그런 기술 내재화가 없다면 MSP에게 요청함. 이러한 운영적 측면을 말함.

그래서 종합적으로 고객의 모든 상황을 파악하는 것이 중요. 리소스나 서버 규모 등에 집중하는 운영이라는 개념을 다르게 보고 시나리오를 선정하는 것이 중요

실제로 고객들은 어떠한 관점을 많이 보는 것인지도 시나리오에 도움이 될 듯함.

* 성능은 당연한 관점으로 봄
* 이것을 베이스로 깔고 비용은 줄일 수 있으면 좋은 것임

여기서 말하는 성능은 대시보드를 의미할 수도 있음. 얼마나 최근 데이터를 빨리 볼 수 있는가를 의미하는 것임.

* Quicksight 경우는 사실 실시간성이 아니기 때문에 얼마나 자주 갱신하느냐를 따질 수 있음. 사람마다 실시간의 기준이 다르기 때문에 어떤 사람은 30분도 느리다 할 수 있고, 어떤 사람은 반나절만에 해도 빠르다고 느끼는 경우가 있음

그래서 이 모든 측면을 다 고려해서 얼마나 합당한 솔루션을 제시했냐를 시나리오에 잘 입혀야 함. 시나리오는 전체 틀을 잡아놓고 인사이트 결과를 낸 후 구체화 시키면 좋을 듯함.

1. **공통 과제 관련 개인 피드백**

액세스 로그 시간 지연 문제

* 버퍼 사이즈 및 시간 설정을 해야 됨

액세스 로그 파일에 있는 액세스 로그 자체는 볼 것도 없고 활용할 수 있는 것도 제한적임. ip 정도는 알 수는 있지만 누가 접근했는지 알 수가 없음.

기존 액세스 로그 활용 방식으로는 사용자 요구 및 행동 이해가 구체화될 수 없음

그래서 보통 여러 데이터를 활용 및 가공해서 클릭 스트림을 분석함

실무에서는 세션, ip 정보를 ip 테이블에 매핑해서 저장해 놨다가 반복적으로 들어온 ip는 동일하다고 생각. 만약 로그인이 안돼서 접근했을 때 이전의 정보를 보고 이 ip로 봤을 때는 이 유저일 것이라고 유추 가능. 그렇지만 폰으로 들어오면 ip가 달라지니까 다양한 ip로 들어올 수 있다는 문제가 있긴 함

이 사람이 사이트에서 어디로 이동하는지 분석이 되어 있어야 함. 그 다음에 경험 최적화가 가능함

로그를 쌓도록 스크립트를 구성할 것. 이 부분은 구글 analytics 참고

* 수집 서버를 만들고 화면에 일어난 이벤트를 수집 서버에 던지면 상세한 로깅 정보가 쌓일 것임. 이를 통해 데이터 마트를 만들 수 있음

더 쉽게 하려면 자바스크립트로 만들어서 현재 세션 id 정보 등을 넘겨 서버에서 받아와 저장하게끔 구성하면 됨. 이 부분은 그걸 구현한 작은 사이트들이 있을 것임. 찾아볼 것

* 클릭할 때 함수를 참고해서 필요한 정보를 받아 파라미터로 서버에 던짐
* 이런 식으로 호출되는 것을 액세스 로그와 유사하게 만들면 좋을 것임

샤드가 하나인데 굳이 kinesis data stream을 쓰는 이유?

* 일시적으로 데이터를 모아서 한 번에 처리. 계속해서 데이터를 보내는 통신이 일어나면 부하도가 높아짐. 일시적으로 저장해서 부하도를 낮추기 위함.
* 또한 전달이 안 되는 데이터가 있을 것임. 특정 시점에서부터 데이터를 가지고 올 수 있을텐데 스트림이나 kafka는 이런 역할을 하는 것임

QuickSight에서 페이지 네이션이라는 옵션을 조심할 것.

요즘은 이것이 기본으로 켜져 있어서 조심해야 함. 엄청 비싸니 무조건 끌 것

1. **공통 과제 종합 피드백**

데이터

* 수집
  + 설문 (excel)
    - 수기든 자동이든 s3로 가게끔
    - 엑셀 파싱은 lambda로 코딩해야 됨
  + DB
    - DMS
    - snapshot

→ 트랜잭션보다는 고객 정보 동기화용

* + 트랜잭션 로그 (access)
    - 실제 로그 수집에 대한 고민을 해봐야 함
      * 스크립트를 통해 수집서버에 모아서 s3로 보내야 함
      * 이 과정에서 firehose, datastream 사용

S3는 데이터 레이크임

하나의 버킷에 담든, 여러 개를 쓰든, 권한 제어적인 측면에서 보면 됨

민감 데이터 같은 경우는 따로 관리하는 게 나을 수도 있음

버킷을 왜 나눴는지에 대한 이유를 설명 가능하면 됨

* 데이터 마트 구성
  + glue crawler
    - meta data
    - partition

→ lambda 로직

* + Data catalog
* 시각화
  + Quicksight까지
  + **대시보드는 필수**

구성해 온 파이프라인을 전부 합쳐서 듣는 사람이 보기 좋은 인사이트가 나온다면 성공한 프로젝트가 될 것임

1. **개인 필수 과제 피드백**

github와 code commit 선정 문제

* code commit은 지원하지 않는 기능이 아직 많을 것임. 개발자들이 많이 쓰는 github를 두고 굳이 code commit을 쓸 필요는 없어 보임
* 또한 github로 사용하던 것을 굳이 code commit으로 이동할 필요는 없음

웹서버가 퍼블릭에 있어도 상관없나?

* 우선 안 됨. 퍼블릭에 있다는 것은 외부에 노출된다는 것을 의미함

웹 서버의 소스 코드 자체는 솔직히 털려도 큰 상관 없음

하지만, 외부 공격자가 퍼블릭에 있는 ec2에 다이렉트로 붙어서 공격할 수 있음

즉, 엔드포인트를 최소화하는 것처럼 관리적 측면을 줄일 필요가 있음. 다 관리할 수 있다면 해도 되지만 현실적으로는 불가능함