snowflake®

SPARK에서 SNOWFLAKE로의 마이그레이션 가이드

Snowpark의 빠른 관리형 데이터 파이프라인에 대한 마이그레이션 모범 사례

챔피언
가이드

TABLE OF CONTENTS

3 마이그레이션을 해야 하는 이유는 무엇일까요?

4 마이그레이션 동기

6 Snowpark를 포함한 Snowflake

9 여러분의 워크로드는 적합할까요?

최소한의 노력으로 마이그레이션할 수 있는 워크로드와 프로세스는 무엇일까요?

10 Snowflake에서 더 쉽게 배포할 수 있는 어떤 새 워크로드를 추가하고 싶으신가요?

11 오늘날 프로세스에 어떤 문제가 있으며 재설계하면 혜택을 누릴 수 있을까요?

12 어떤 워크로드가 오래되어 완전 점검이 필요하거나 이 플랫폼의 이점을 누릴 수 있을까요?

14 마이그레이션 계획

17 Spark보다 높은 Snowflake의 가치

18 도움이 필요하신가요?

19 Snowflake 및 Mobilize.Net 소개

마이그레이션을 해야 하는 이유는 무엇일까요?

지난 몇 년간 Hive, Hadoop 및 Apache Spark와 같은 테크놀로지를 통해 수많은 데이터 파이프라인이 개발되었습니다.

엄청난 속도로 늘어나는 데이터는 조직에 새로운 기회를 제시하지만 현재의 인프라로는 많은 양의 데이터를 대규모로 수집, 변환, 분석하기가 어렵습니다.[1]

개발자들은 다음과 같은 다양한 문제가 있다고 언급해 왔습니다.

1. **복잡성:** 인프라를 구축 및 유지하고 문제를 해결하는 것이 복잡합니다. 구현 시 다양한 도구에 의존해 데이터를 처리하고 준비해야 합니다. 파이프라인 안정성을 달성하기가 어렵습니다. 이는 성능 병목 현상 및 가동 중지 시간을 초래합니다. 문제를 수정하는 데 막대한 노력과 투자가 필요한 경우도 있습니다.
2. **편의성 및 비용:** 데이터 팀은 이 복잡한 배포를 관리해야 합니다. 또한, 조직에서 채용해야 하는 전문가를 찾기 힘들고 이들을 유지하기 위한 비용도 높을 수 있습니다.
3. **다양한 데이터 소스:** 데이터 과학자는 수많은 소스에서 데이터를 찾고 가져온 다음, 이를 공유할 방법을 찾는 데 시간을 들여야 합니다.
4. **확장성 문제:** 용량 추가 및/또는 축소 시 지연이 발생합니다.
5. **가용성 위험:** 멀티 존(AZ) 솔루션을 설정하려면 기본 구성 요소를 자세히 알고 있어야 합니다.
6. **보안:** 자격 증명을 설정 및 유지해야 하는 장소가 너무 많습니다.

Spark란 무엇일까요?

Apache Spark™는 단일 노드 머신 또는 클러스터에서 데이터 엔지니어링, 데이터 사이언스 및 머신 러닝을 실행하기 위한 다국어 엔진입니다. 이는 대규모 데이터 세트를 대상으로도 비정형 정보를 수집, 셰이핑 및 처리하는 메커니즘을 제공합니다.

Spark는 컴퓨팅 집약적인 데이터 엔지니어링 및 분석 작업에 유용한 도구입니다. 사용자는 Spark에서 클러스터 확장을 위한 코드를 작성할 수 있지만 그렇게 한다고 해서 이를 관리하는 것에 대한 복잡성이 없어지지 않습니다. 또한, Spark를 추상화한다고 해서 메모리 및 컴퓨팅과 같은 분산 리소스의 사용에 대한 내재된 복잡성이 사라지지는 않습니다. 사용자가 데이터에 집중하는 대신 인프라에 엄청난 노력을 들여야 한다는 것입니다.

마이그레이션 동기

제로에 가까운 유지 관리

Snowflake와 함께하면 기본 제공 성능의 복잡성이 줄어드므로, 인프라를 변경하거나, 조정하거나, 장애(예: 시스템 업그레이드, 보안 패치, 용량 계획)를 해결할 필요가 없게 됩니다.

비즈니스 사용자의 필요를 충족하는 최신 아키텍처

고객은 기존 데이터 아키텍처 접근 방식(그림 1 참고)을 사용하는 경우에는 보통 별도의 클라우드 스토리지 계층, 데이터 웨어하우스, 처리 프레임워크(예: Spark)를 실행합니다. 이에 따라 잦은 데이터 이동, 불필요한 데이터 파이프라인 및 데이터 사일로가 발생하며, 완전한 액세스가 제한될 수 있습니다. 데이터 웨어하우스, 클라우드 스토리지 및 Spark 시스템에는 각기 다른 보안 도메인과 제어가 존재합니다. 또한, 각 종점은 잠재적 거버넌스 및 보안 취약점입니다. 이 모든 복잡성은 숙련된 직원이 수동으로 관리 및 유지해야 하는 다양한 잠재적 장애점과 워크로드 병목현상을 초래합니다.

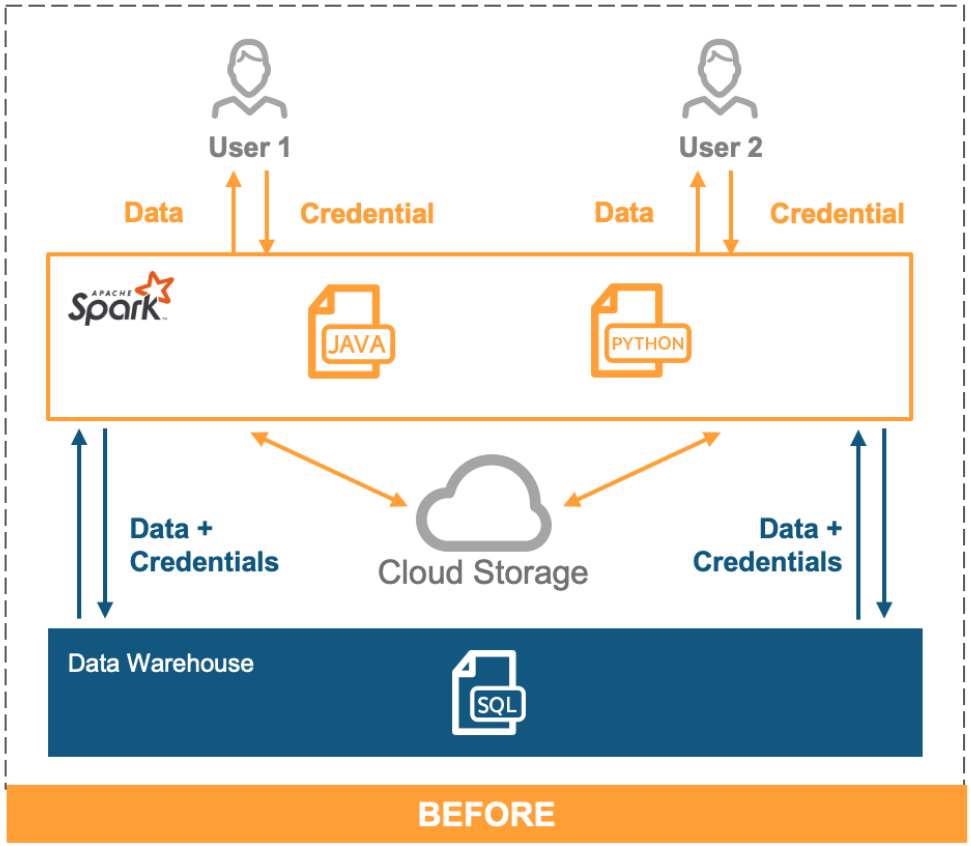


그림 1: 이전: 기존 접근 방식을 사용한 복잡한 데이터 파이프라인

더 빠른 데이터 액세스

Snowflake의 유연하고 거의 무제한인 규모 및 속도로 데이터 전문가는 언제든지 현재 및 과거 데이터에 모두 빠르게 액세스할 수 있습니다.

데이터 공유

Snowflake가 있으면 거의 실시간으로 직접적이고, 관리되는 안전한 데이터 공유가 가능하기에 엔터프라이즈는 일대일, 일대다 및 다대다 데이터 공유 관계를 구축할 수 있습니다.

원하는 프로그래밍 언어가 사용된 완전한 SQL 데이터베이스

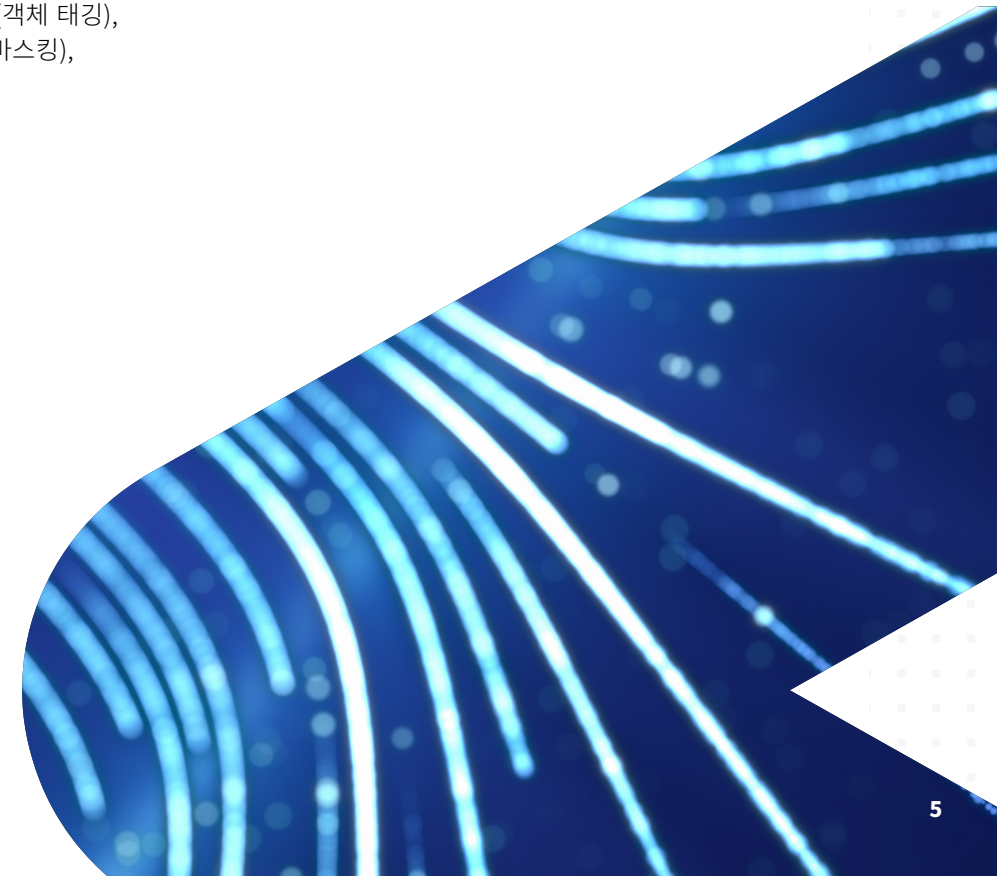
Snowflake는 수백만 명의 비즈니스 사용자가 이미 어떻게 사용하는지 알고 있는 도구를 지원합니다. 또한, 데이터 엔지니어와 데이터 과학자가 Scala, Java 또는 Python에서 코딩할 수 있도록 하는 개발자 프레임워크인 Snowpark를 함께 제공합니다. 이 가이드의 'Snowpark를 포함한 Snowflake' 섹션에서 Snowpark에 대해 자세히 알아보세요.

비용

Snowflake를 사용하면 운영 비용을 높여 총소유 비용(TCO)을 늘리는 기존 데이터 아키텍처의 복잡성 및 편의성 문제를 방지할 수 있습니다. 기존 기술과 관련된 성능 문제 또한 쿼리 완료 시간을 늘리고 필요한 클러스터 크기를 키워 비용을 키울 수 있습니다. 또한, 기존 기술을 사용하면 클러스터는 비활성 상태에서도 계속 실행되고(쉬운 클러스터 켜기/끄기의 부재가 원인임) 유료 여유 용량의 사용이 권장되는 등, 탄력성이 제한되어 있어 비용이 늘어납니다. 마지막으로 데이터 사일로가 생성되면 비즈니스 가치가 더욱더 떨어질 수 있습니다.

거버넌스 기능

Snowflake는 단일 진실 공급원을 제공하므로, 사용자가 단 하나의 플랫폼에서 데이터를 파악하고(객체 태깅), 데이터를 보호하고(익명화, 액세스 정책, 마스킹), 데이터 액세스를 추적할 수 있습니다.



SNOWPARK를 포함한 SNOWFLAKE

Snowpark는 원하는 방식으로 Snowflake 코드를 쓰고 Snowflake 내에서 직접 실행할 수 있는 새로운 개발자 경험을 선사합니다.

Snowpark는 데이터 프로그래밍 기능을 자연스럽고 쉽게 만듭니다. 조직은 기존의 자사 SQL 스킬을 사용하고 Scala, Java 또는 Python을 포함시킬 수 있습니다. 또한, 익숙한 데이터프레임 구조를 통해 강력하고 효율적인 파이프라인, 머신 러닝(ML) 워크플로우 및 데이터 애플리케이션을 구축할 수 있습니다. 즉, Spark 사용자가 Snowflake를 사용하면 동일한 방식으로 코딩하면서 더 빠르고 안전하게 실행할 수 있습니다(그림 2 참고).

조직은 데이터 증강, 변환, ML 스코어링 및 비즈니스 로직 등을 위해 선호하는 라이브러리 및 사용자 정의 로직으로 사용자 정의 함수(UDF)를 생성할 수 있습니다. 그런 다음 자동으로 무거운 작업을 수행하기 위해 이를 Snowflake 엔진으로 푸시할 수 있습니다. 또, 조직은 Snowpark를 통해 Snowflake 데이터 클라우드의 이점을 누릴 수 있습니다. 이는 사용이 쉽고, 연결되어 있으며, 비용 효율적이고, 관리되는 플랫폼으로, 기본적으로 클라우드에서 설계되었습니다.

Snowflake는 데이터 스토리지, 처리 및 공유에 걸쳐 클라우드 탄력성의 혜택을 완벽하게 활용합니다. 또한, 근본적으로 여러 팀에 걸친 강력한 협업을 통해 다양한 데이터 워크로드에 대한 접근 방식을 바꿉니다.

Snowflake의 아키텍처는 고객에게 필요에 따라 데이터 레이크, 데이터 웨어하우스나 데이터 메시 아키텍처를 사용할 수 있게 하는 등, 유연성을 제공합니다.

Snowflake의 소비 기반 가격 책정을 통해 고객은 사용한 만큼만 지불합니다. 또한, 가장 중요한 점은 Snowflake를 사용하면 고객이 불필요한 데이터 파이프라인을 구축하거나 이 데이터에 액세스하기 위해 기존 방법을 사용하는 대신, 데이터 공유 기능을 통해 쉽게 액세스할 수 있는 데이터 및 서비스 네트워크를 이용할 수 있다는 것입니다.

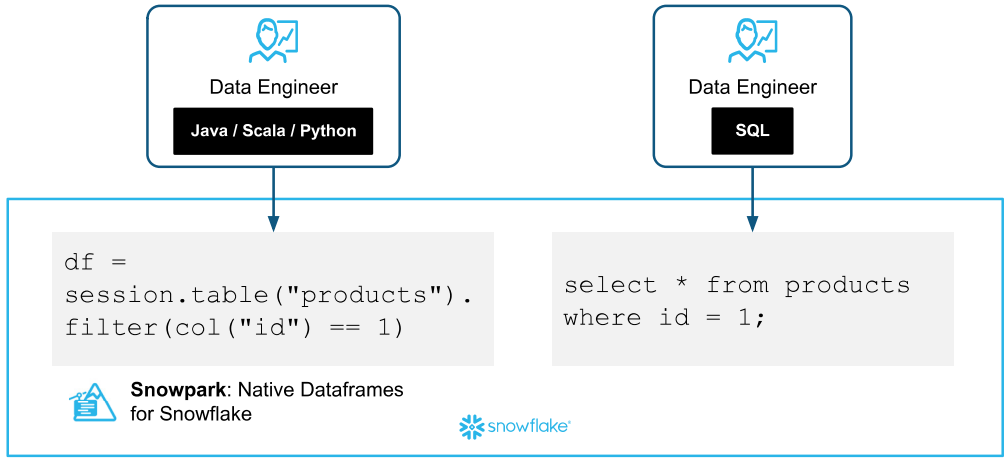


그림 2: 사용자는 Snowpark를 통해 Java, Scala 또는 Python으로 코딩하고 이를 Snowflake 내에서 바로 실행할 수 있습니다.

마지막으로 Snowflake의 지능형 인프라는 여러분의 팀이 인프라 설치 또는 유지 관리에 대해 걱정할 필요 없이 데이터에 접속합니다. Snowflake로 마이그레이션하면 다음과 같은 이점을 바로 얻을 수 있습니다.

- **보안:** 산업 모범 사례에 따른 보안 구성 및 지속적인 취약성 테스트가 제공됩니다. 클라우드 컴퓨팅의 경우, 데이터 자산이 적절한 보안 설정과 지속적으로 발생하는 새로운 보안 위협을 차단하기 위한 지속적인 교육과 노력으로 보호되어야 합니다. Snowflake는 언제든지 활용할 수 있는 강력한 RBAC 시스템으로 안전한 플랫폼을 제공합니다.
- **쉬운 사용:** 하나의 데이터 복사본과 단일 진실 공급원을 가진 간소화된 아키텍처가 제공됩니다.
- **데이터 공유:** 데이터를 쉽고 안전하게 조직 내에서 공유하거나, 직접 공유 또는 Snowflake 데이터 마켓플레이스를 통해 외부에서 고객과 공유합니다.
- **제로 카피 클론 생성:** 데이터를 실제로 복사하지 않고 테이블, 스키마 또는 데이터베이스의 '복사본'을 여러 개 생성합니다. 이는 복사에 걸리는 시간을 단축하고 데이터 스토리지 비용을 절감합니다.
- **거의 즉각적인 확장성:** 컴퓨팅과 스토리지의 규모를 개별적으로 조정하고 전용 웨어하우스가 필요한 작업을 위해 컴퓨팅 파워를 분리합니다.
- **가용성:** 단순한 멀티 AZ 지원 그리고 모든 지역과 클라우드에 걸친 복제 및 대체 작동을 제공합니다.

SNOWFLAKE 데이터 마켓플레이스

Snowflake 데이터 마켓플레이스는 모델에 정보를 알리기 위해 데이터 팀에 수백 개의 즉시 쿼리 가능한 데이터 세트 및 데이터 서비스에 대한 직접적인 액세스를 제공합니다. 여기에는 공중 보건, 날씨, 위치, 인구 통계 등을 비롯한 여러 부문과 범주에 걸친 다양한 공개 및 상업용 데이터가 포함됩니다.

매일 알루미늄 가격을 예측하기 위해 AI/ML을 사용하는 조직의 필요에 대해 생각해 보겠습니다. 이를 위해 데이터를 매일 받으려면 Bloomberg를 구독해야 합니다. 그러는 동안 옥수수 가격이 알루미늄 가격에 영향을 준다는 것을 알게 될 것입니다. 이는 데이터 팀의 예측을 추가적으로 개선할 외부 데이터입니다. 이러한 경우 데이터 과학자에게는 데이터에 대한 더 광범위한 온디맨드식 액세스가 필요합니다. Snowflake 데이터 마켓플레이스는 이를 240개 이상의 타사 데이터 공급자 및 데이터 서비스 공급자에 대한 액세스를 통해 제공합니다(2022년 1월 31일 기준).

Snowflake 데이터 마켓플레이스는 Snowflake 계정에서 직접 안전하게 관리되며 액세스할 수 있게 하여 타사 데이터 수집 및 변환의 기존 ETL 프로세스에 드는 비용과 노력을 줄입니다. 이는 자동 실시간 업데이트를 통해 더 새로운 데이터에 더 빠르게 액세스할 수 있게 하기에 불필요한 데이터 파이프라인을 구축할 필요가 없어집니다.

자신 있게 하는 마이그레이션

이제 Snowpark용 Snowflake가 제공하는 지원을 통해 그 어느 때보다 더 쉽게 데이터 클라우드의 이러한 요소가 선사하는 이점을 활용할 수 있습니다. 다음 섹션은 마이그레이션 프로세스를 안내하며 Spark에서 Snowpark로의 마이그레이션을 집중적으로 다룹니다. 하지만 여러분 스스로 전체 프로세스를 진행할 필요는 없습니다.

여러분을 도와드릴 Snowflake의 파트너 네트워크가 마련되어 있습니다. 또한, Mobilize.Net SnowConvert와 같은 도구도 마이그레이션 프로젝트를 더 빠르게 진행하는 데 도움이 됩니다.

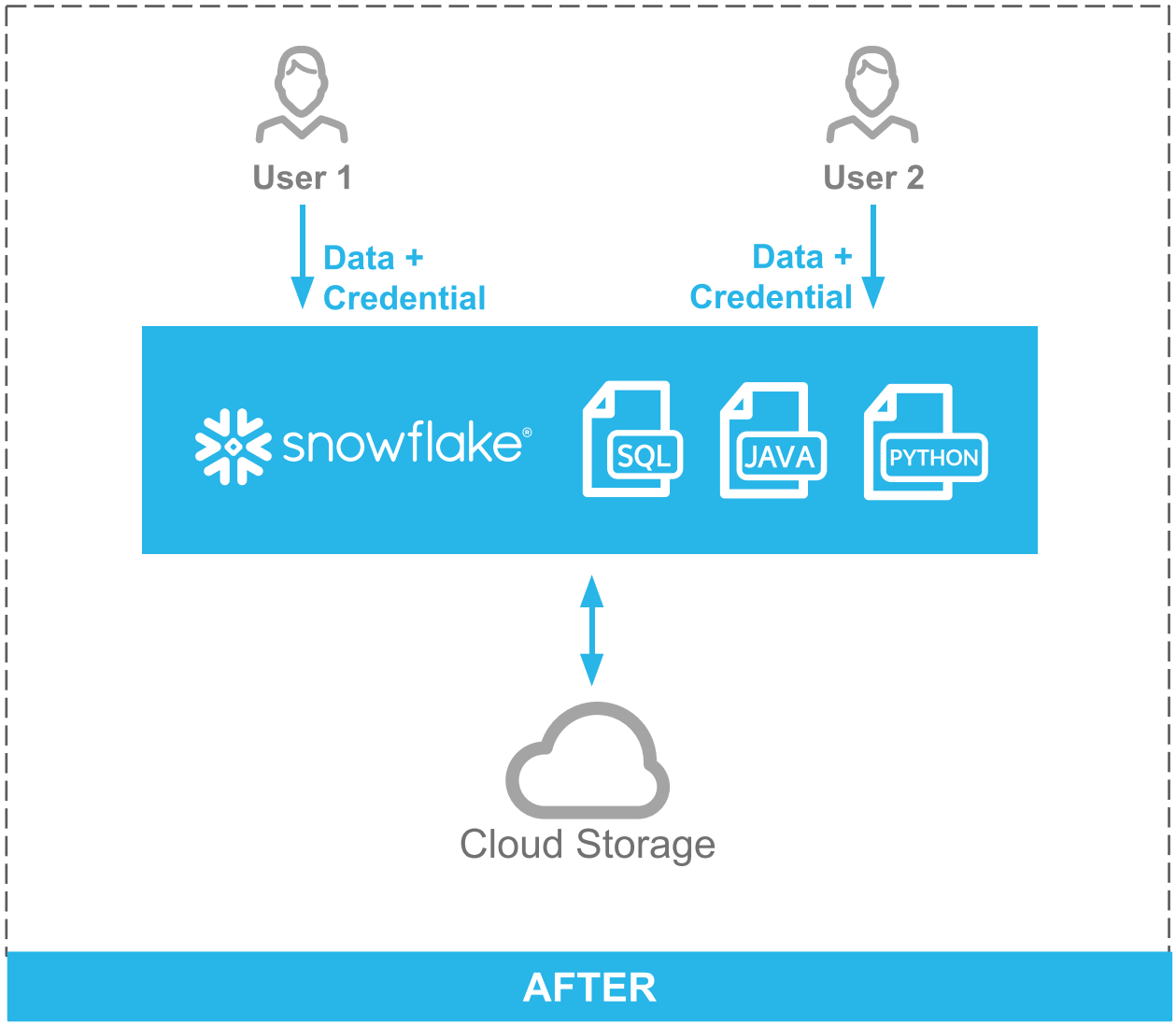


그림 3: Snowflake를 통해 간소화된 아키텍처

여러분의 워크로드는 적합할까요?

워크로드를 Spark에서 Snowflake로 마이그레이션하는 것을 고려하는 경우, 이 질문을 가장 먼저 던져야 합니다. 최상위 뷰에서는 워크로드를 구성하는 엔지니어링 또는 데이터 사이언스를 이해하는 것이 중요합니다. Spark 작업을 분류하면 어떤 워크로드가 Snowflake를 통해 업그레이드하는 데 가장 적합한지 올바르게 결정하는 데 도움이 될 수 있습니다.

대상으로 삼을 가장 적합한 워크로드로 시작할 수 있도록, 여러분의 워크로드 및 데이터에 대해 다음 4가지 질문을 해 보시기 바랍니다.

1. 최소한의 노력으로 마이그레이션할 수 있는 워크로드와 프로세스는 무엇일까요?

데이터 파이프라인

Spark를 사용하면 각기 다른 데이터 파이프라인을 설정하고 여러 소스에서 데이터를 찾고 가져오는 데 시간을 허비해야 합니다. Snowpark로 마이그레이션하면 이러한 문제 중 일부가 없어집니다.

Snowflake 고객은 데이터 클라우드와 간소화된 아키텍처의 이점을 모두 누리기 위해 Snowflake로 데이터를 이동하거나 객체 스토어를 유지할 뿐만 아니라, 외부 테이블을 통해 Snowflake에서 데이터에 바로 액세스하는 유연성을 갖습니다. Snowflake는 여러분의 환경에 있는 클라우드 스토리지의 파일을 사용할 수 있습니다.

대부분의 경우 이러한 데이터를 Snowflake 테이블로 이동하면 유연성과 성능이 개선됩니다. 예를 들어, 날짜 또는 회사에 따라 데이터 조각을 가져오는 작업은 Parquet 파일보다 유지 관리 오버헤드가 없는 테이블로 수행할 때 성능이 더 뛰어납니다. 마이그레이션 도구는 DDL 정의 변환 속도를 단축할 수 있습니다.

데이터가 이미 Snowflake에 있다면 데이터에 필요한 변환에만 집중할 수 있습니다. 여러 Snowflake 웨어하우스 사이에서 데이터 액세스하는 것은 간단하며 Snowflake 안전한 데이터 공유가 가진 위력의 또 다른 좋은 예입니다.

온프레미스 데이터의 경우 마이그레이션하기 위해 다양한 커넥터가 존재합니다. 또는, 이를 클라우드 스토리지에 업로드할 수 있습니다.

추출, 변환, 로드(ETL) 또는 추출, 로드, 변환(ELT) 작업

Snowpark 데이터프레임 API는 이 프로세스를 간소화합니다. 단 몇 줄의 코드로 여러 데이터 테이블을 가져와 이러한 데이터에서 중복을 제거하여 변환하고, 결합하고, 다른 테이블 또는 뷰에 로드할 수 있습니다.

데이터 세트를 필터링하고, 다른 데이터 세트의 열을 추가하여 이를 집계하고, 다양한 데이터를 결합하고, 그 결과를 저장한 다음, 하위 세트를 또 다른 계산 단계에서 사용할 수도 있습니다. 또한, 이 모든 작업을 원하는 언어로 수행할 수 있습니다. 그뿐만 아니라 이 작업은 SQL에서 진행할 수 있습니다. 여러분은 Python 및 Pandas에 대한 지식, 기본적인 Java 지식 또는 기능 Scala 데이터 프로그래밍을 활용하실 수 있습니다.

그리고 선호하는 언어로 작성한 이러한 파이프라인을 사용하고, Snowflake에서 작업 및 저장 프로시저로 이를 운영할 수 있습니다.

2. SNOWFLAKE에서 더 쉽게 배포할 수 있는 어떤 새 워크로드를 추가하고 싶으신가요?

Snowflake를 사용하여 다음을 포함한 문제를 해결할 수도 있습니다.

비즈니스 로직 표준화

여러분의 비즈니스에는 데이터 형식 처리 또는 크레딧 스코어링 알고리즘(그림 4 참고)과 같은 작업을 위한 가치 있으며 재사용 가능한 로직을 구현하는 Java/Python/Scala 코드가 있습니다.

데이터 셰이핑

여러분의 파이프라인에서 수행하는 일반적인 작업은 데이터를 처리하고 소비하기 위해 이를 일반적인 형식으로 만드는 것입니다. 몇몇 사용 사례를 검토해보겠습니다.

- **이진 형식에서 변환 또는 학습된 모델로 데이터 추출:** UDF는 특정 이진 형식(암호화되어 있을 수 있음)의 열에서 값을 읽고 사용자 지정 알고리즘을 적용할 수 있습니다. 그런 다음에는 파일을 스테이지에 업로드하고 이를 UDF에서 읽을 수 있게 됩니다. 또한, UDF는 문서에서 감정을 분석하는 것과 같은 사용 사례를 위해 이전에 학습된 모델을 활용하고 데이터를 집계하려 이를 입력 파일에 적용할 수 있습니다. UDF가 배포되었다면 이 작업을 단순한 SQL 쿼리에서도 수행할 수 있습니다.
- **JSON 정규화:** 다양한 소스에서 가져온 비정형 정보를 형성하려면 추가 처리를 위해 통합할 수 있도록 약간의 작업을 거쳐야 합니다. XML 또는 JSON 파일은 파이프라인의 추후 단계에서 소비될 지원되는 형식과 일치시키기 위해 처리할 수 있습니다.
- **사용자 지정 XML 처리:** 이 단계는 대규모 또는 복잡한 XML 데이터에 대한 분석을 수행하거나 다양한 스키마가 포함된 문서를 정규화하는 데 사용됩니다. 그리고 예를 들어, XML 처리에 모든 Python 패키지를 사용할 수 있습니다.

데이터 랭글링

데이터 정제, 데이터 수정 또는 데이터 먼징이라고도 하는 데이터 랭글링은 원시 데이터를 더 쉽게 사용할 수 있는 형식으로 변환하도록 설계된 다양한 프로세스를 지칭합니다. 여기에는 데이터 정화 작업과 데이터 유효성 검사가 포함될 수 있습니다.

- **기록 정화 및 수정:** 값이 예상 범위 내에 있는지 확인하기 위해 검사합니다. 이러한 중요한 작업은 기록 세트, 테이블 또는 데이터베이스에서 손상되었거나 잘못된 기록을 감지 및 수정하는 데 사용됩니다.
  - ***이상값 제거***는 범위에 속하지 않는 데이터 포인트를 제외하고 명확한 원인을 기록합니다.
  - ***철자 오류 수정***은 예를 들어 Python 맞춤법 검사기를 사용합니다.
  - ***봇 감지***는 애플리케이션이 사용자 행동을 추적하기 위해 메트릭을 수집하는 행위입니다. 일반적으로 인간 행동에 대한 학습을 위한 데이터가 손상되지 않도록 봇을 감지해야 합니다.
- **데이터 유효성 검사:** 유효성 검사의 일반적인 목표는 데이터를 모델 학습으로 이동하기 전에 정상적인지 확인하는 것입니다. 이러한 단계에는 다음이 포함됩니다.
  - ***데이터 룩업***은 열에 있는 데이터가 알려진 허용 값 세트에 속하는지 확인합니다. 또한, 추가 정보를 검색하는데 핵심적인 값을 사용합니다.
  - ***일반적인 유효성 검사***는 해당 값이 NULL이 아닌지 확인하거나 적절한 값에 NULL을 매핑합니다.

그림 4: 모든 사용자에게 일반적인 Java/Python 라이브러리 배포.

데이터 증강

Snowpark를 사용하면 Java 또는 Python UDF가 Snowflake 내부에서 전송되는 데이터를 증강함에 따라 (그림 5 참고) 학습된 ML 모델을 포함한 사용자 지정 함수를 가져올 수 있습니다.

더 많은 데이터 증강 사용 사례의 예로는 다음이 있습니다.

- **언어 감지:** 텍스트 조각의 언어를 결정합니다.
- **IP 지리적 위치:** 인터넷에 연결된 장치의 실제 지리적 위치에 IP 또는 MAC 주소를 매핑합니다. 예를 들어, 방문자의 국가 정보를 가져옵니다.
- **네트워크 분석:** 이는 소셜 네트워크 또는 자연스러운 프로세스 구조의 관계를 이해하는 데 사용하거나 유기체의 생물학적 시스템을 분석하는 것과 같은 복잡한 문제에 사용할 수 있습니다.
- **데이터 구간화:** 이 프로세스는 다수의 거의 연속적인 값을 더 적은 수의 '구간' 그룹으로 분류합니다. 예를 들어, 한 무리의 사람에 대한 데이터를 보유하고 있다면 이들의 연령을 더 적은 수의 연령 구간으로 정리하고 싶을 수 있습니다.

3. 오늘날 프로세스에 어떤 문제가 있으며 재설계하면 혜택을 누릴 수 있을까요?

데이터 보안

- **고객 정보에서 가져온 PII 데이터입니다.** 모든 개인 식별 정보(PII)를 감지 또는 제거하기 위해 UDF를 데이터 세트의 열에 있는 콘텐츠에 적용할 수 있습니다. 예를 들어, 신용 카드 번호 또는 SSN과 같은 중요한 열을 식별하고 나면 Snowflake 내부에 있는 규제를 받는 미사용 데이터를 암호화하기 위해 Java UDF를 적용할 수 있습니다(그림 6 참고).
- **데이터 마스킹, 필터링 및 행 수준 암호화를 활용할 수 있습니다.** 이는 GDPR, HIPAA, CCPA 등의 지배를 받는 데이터 흐름과 관련되어 있습니다(그림 7 참고).

데이터 게시

Snowflake에서 데이터를 쉽게 게시할 수 있습니다. 데이터를 변환한 다음에는 이를 클라우드 스토리지에 파일로 저장할 수 있습니다. 더 좋은 방법은 이를 Snowflake에 다시 저장한 다음 데이터 공유를 사용하여 데이터를 RBAC와 적절한 보안 및 거버넌스를 통해 다른 소비자에게 제공하는 것입니다.

Snowflake는 여러분의 클라우드 스토리지, 안전한 데이터 공유, 제로 카피 클론 생성 및 데이터 클린룸 기능을 포함한 기능 세트와 통합되어 여러분이 단일 관리형 플랫폼 내에서 데이터를 제공할 수 있도록 합니다.

그림 5: Snowflake 내부에서 Java 또는 Python UDF를 사용하여 데이터 증강.

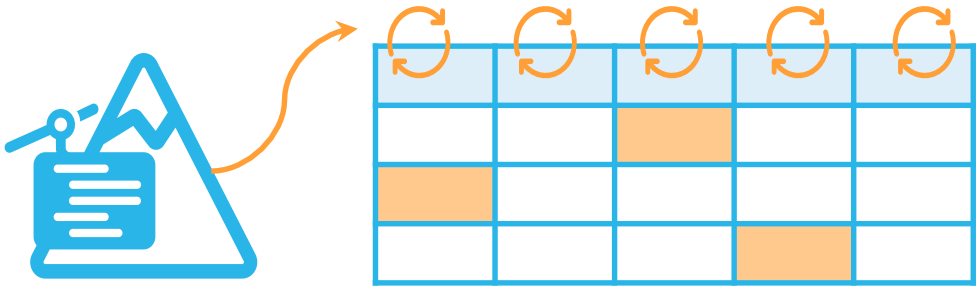


그림 6: 변칙 또는 PII를 찾기 위해 모든 데이터프레임 열에 걸쳐 반복하여 Snowpark UDF 사용.

4. 어떤 워크로드가 오래되어 완전 점검이 필요하거나 해당 플랫폼의 이점을 누릴 수 있을까요?

비정형 데이터 관리

Snowflake를 통해 비정형 데이터와 메타데이터를 저장하고, 액세스하고, 처리하고, 공유할 수 있습니다. 또한, 파일의 파인 그레인드 거버넌스를 사용하며 단순한 SQL 명령으로 모든 데이터 세트를 검색할 수 있습니다(그림 8 참고).

데이터 사이언스

또한, Snowpark는 다수의 데이터 사이언스 프로세스에 걸쳐 이점을 제공합니다.

- **기능 엔지니어링:** 이 작업은 도메인별로 매우 다르지만 일반적으로 데이터 변환, 데이터 구간화 및 데이터 세트에 데이터를 추가하거나 제거하기 위한 몇몇 사용자 지정 로직의 조합이 사용됩니다. 기능 엔지니어링은 때로는 프로세스의 한 단계이며 때로는 최종 목표입니다. 기능이 추가된 정제된 데이터 세트는 일반적으로 다른 프로세스에서 사용할 수 있도록 게시됩니다. 이러한 다른 프로세스는 다른 비즈니스 인텔리전스(BI) 도구이거나 모델 학습 알고리즘의 입력일 수 있습니다.

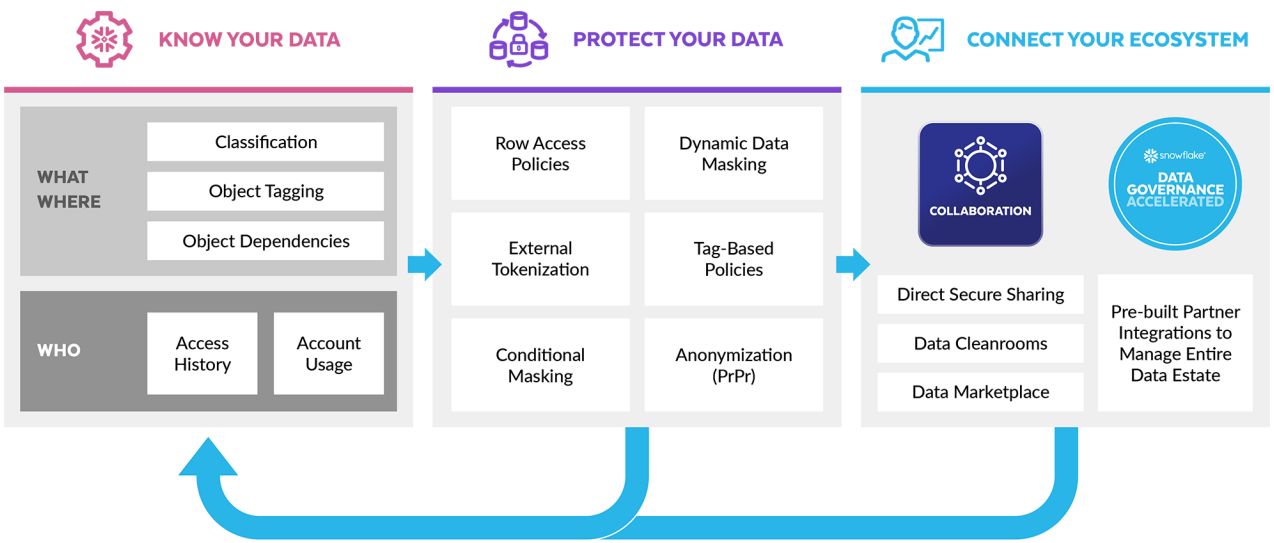


그림 7: DPR, HIPAA, CCPA 등과 같은 규제를 준수할 수 있도록 데이터 흐름에 데이터 마스킹, 필터링 및 행 수준 암호화 사용.

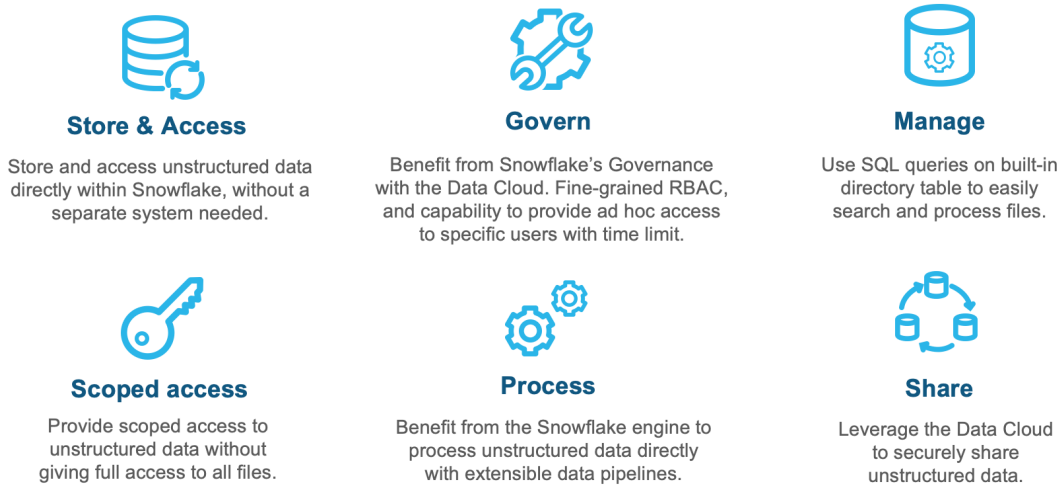


그림 8: Snowflake에서 비정형 데이터 저장, 액세스, 처리 및 공유.

- **데이터 증강:** Snowflake 데이터 마켓플레이스는 여러분의 데이터를 증강하는 데 사용할 수 있는 추가 데이터 세트의 편리한 소스를 제공합니다. 예를 들어, 더 나은 지리적 데이터, 시계열 개인 및 상용 컨텍스트로 데이터를 보강하여 예측 의사 결정의 정확도를 수량화할 수 있는 방식으로 더 빠르게 원하는 규모로 개선할 수 있습니다. 더 빠르고 개선된 통찰력으로 조직은 수익을 늘리고, 작업을 간소화하고, 위험을 줄일 수 있습니다.
- **실험 스냅샷:** 실험 데이터 스냅샷을 위한 제로 카피 클론 생성으로 모델 학습 실험을 쉽게 설정할 수 있습니다.
- **모델 학습:** Snowpark는 Anaconda와의 제품 통합의 일환으로 제공되는 사전 설치된 오픈 소스 라이브러리를 사용하여 분산 학습이나 GPU 인프라를 요구하지 않는 모델 학습 사용 사례를 지원합니다. Snowflake 사용자는 더 광범위한 학습 지원을 위해 사전 구축된 선도적인 ML 플랫폼과의 통합을 활용할 수도 있습니다. 또는, Snowflake와 여러분의 ML 학습 환경 사이에 빠른 데이터 전송을 가능하게 하는 Python용 Snowflake 커넥터를 활용할 수 있습니다.
- **학습된 모델 운영(MLOps):** Snowpark는 UDF 내 객체, 코드 및 기타 아티팩트에 대한 지원을 통해 실험에서 프로덕션까지의 경로를 간소화합니다. Snowflake에서 학습되었든 외부 플랫폼에서 학습되었든 머신 러닝 모델은 이러한 아티팩트 중 하나일 수 있습니다. Snowflake로 모델을 가져오면 데이터 파이프라인과 모델 사이의 통합이 간소화되어 프로덕션의 모델 추론까지 경로가 단축됩니다.

마이그레이션 계획

성공적인 마이그레이션 프로젝트의 경우, 마이그레이션을 위해 고려할 모든 구성 요소를 확인할 수 있으며 잘 설계된 계획이 필요합니다. 이 섹션은 실제 마이그레이션을 시작하기 전에 수집해야 하는 정보의 체크리스트를 제공합니다.

식별 및 평가

첫 번째 단계는 핵심 애플리케이션 정보를 수집하는 것입니다. 여기에는 파일 인벤토리 수행, 이미 사용 중인 테크놀로지 목록 작성, API 사용량 기록이 포함됩니다. 다음은 시작하기 위해 수집할 것이 권장되는 정보의 목록입니다.

- **완전한 파일 인벤토리**
- **테크놀로지 목록(워크로드가 Python, Java 또는 Scala로 작성되었나요?)**
- **사용 중:**
  - Spark UDF
  - Spark 사용자 지정 집계 함수
  - Spark 탄력적인 분산 데이터 세트(RDD) 및 파티션 API
  - 추가 라이브러리(오픈 소스, 타사)
  - 초기 진입점 목록

단계	목적	결과
식별 및 평가	애플리케이션 크기 및 종속성 결정	애플리케이션 인벤토리: • 파일 목록/크기 • 작업 목록 • 데이터 소스 종속성 • 다른 데이터베이스에 대한 단층 파일, CSV, Parquet, Avro, delta, JDBC • 읽기/쓰기 상태 • 벤치마크 • 대기 시간(배치 vs 실시간) • 데이터/컴퓨팅 시간 • CPU/메모리 사용량
프로젝트 개요	잠재적 위험 식별 및 프로젝트 타임라인 확립	범위/프로젝트 타임라인
애플리케이션 마이그레이션	기존 애플리케이션에서 새로운 코드베이스 생성성	제거, 대체 및 리매핑
마이그레이션 테스트	대상 애플리케이션 기능적 등가 검증	허용 오차 내에서 대상 데이터 및 성능을 검증하는 벤치마크

- **스트리밍 사용량**
- **SparkML 또는 테크놀로지와 관련된 기타 모델 학습 사용량**
- **가져오기 사용량 수집**
- **API 사용량 세부 사항**
- **Hive 또는 Spark SQL 명령 및 SparkSQL 메트릭 지시**
- **긴 데이터프레임 체인을 가진 파일 강조**
- **식별된 데이터 소스 패턴 나열**
- **Maven 종속성 및 호환성 추적**

이 모든 정보는 진행 중인 작업의 크기를 평가하는 데 도움이 됩니다. 그러나 이 프로세스 자체가 복잡하고 일부 수동 작업이 필요할 수 있습니다. 고객이 마이그레이션 프로젝트를 시작하는 데 도움이 되려면 Mobilize.Net에서 제공하는 것과 같은 파트너 솔루션이 유용할 수 있습니다. Snowflake는 Mobilize.Net과 협업하여 Spark용 SnowConvert를 구축했습니다. 이 마이그레이션 도구를 통해 여러분의 코드 대부분이 Snowpark와 호환 가능한 데이터프레임 API를 활용할지 판단하고, 어떤 파일에 지원되지 않는 기능이 많은지 식별할 수 있습니다. 이 도구는 준비 점수를 결정합니다. 여러분은 이에 따라 Snowpark로 이동하는 것이 얼마나 쉬울지 판단하실 수 있습니다.

또한, 생성된 평가는 모범 사례를 포함한 문서와 연결된 링크와 Snowpark로 이동하기 전에 조정이 필요한 코드 요소를 재설계할 수 있도록 여러분을 안내하는 참조 아키텍처 정보를 제공합니다.

데이터 소스 인벤토리

이 단계는 CSV, Parquet, flat files, Avro 및 Delta를 포함한 모든 정형, 반정형 및 비정형 형식을 평가합니다. 또한, 온프레미스의 데이터 소스(예: SQL server, Oracle, Teradata, Greenplum), 클라우드 데이터베이스(예: Redshift, BigQuery) 및 클라우드 스토리지(예: Google, Amazon, Azure)를 추적합니다. 각 데이터 소스에서는 특정 테이블/파일, 데이터 메트릭, 사용량 수준, 동기화 니즈, 데이터 소스의 읽기 또는 쓰기 여부를 결정하는 것이 중요합니다.

현재 데이터 레이크 구조를 결정하는 것이 중요합니다. 일반적인 데이터 레이크에는 유연성이 없는 아키텍처 또는 느린 데이터 이동과 같은 몇몇 문제가 존재할 수 있습니다. 이는 파이프라인 안정성에 영향을 줄 수 있기에 이러한 데이터를 Snowflake 테이블로 바로 이동하는 것을 고려하는 것이 좋습니다.

타사 공급업체 소프트웨어 또는 API 종속성

이는 문제의 데이터베이스를 보강하거나 변환하기 위해 타사 도구 또는 API가 현재 사용되고 있는지 식별합니다.

- **Java 및 Scala 워크로드의 경우 표준 라이브러리 대부분을 Snowflake에 업로드하고 걱정 없이 사용할 수 있습니다.**
- **Python의 경우 Snowflake는 이제 Anaconda를 활용하여 표준 Python 패키지에 안전한 액세스를 제공합니다.**
- **타사 API가 지리적 위치 정보를 가져오는 것과 같은 상황에서 사용되었다면 이는 Snowflake 데이터 마켓플레이스에서 가져온 지리 공간 데이터 세트를 사용하여 쉽게 대체될 수 있습니다.**

Snowflake는 API 호출보다 나은 대안을 제시할 수 있는 데이터 세트를 제공하는 수많은 테크놀로지 파트너를 보유하고 있습니다.

프로젝트 개요

프로젝트 타임라인을 결정하고 잠재적 위험 식별을 시작하는 것은 중요합니다. 이상적인 상황은 Snowflake 데이터 클라우드 내에서 모든 데이터를 소비하는 것입니다. 이렇게 하면 보안과 거버넌스를 모두 개선하는 동시에 최고의 성능을 얻게 됩니다.

몇몇 대체 대안과 더불어 가능한 비용 혜택을 이 단계에서 고려할 수 있습니다. 예를 들어, Apache Parquet 파일을 대체하면 성능 방면에서 혜택을 누리게 되고 일부 필터링 및 그룹화 작업이 더 쉬워집니다.

성공적인 마이그레이션을 위한 또 다른 핵심적인 측면은 테스트 데이터입니다. 허용되는 오류 허용 오차 수준(예: 라운딩으로 인해 허용되는 오류 여유)을 결정하는 것도 중요합니다. 또한, 일부 데이터 세트가 행 카운트에서 약간씩 다를 수 있다는 점을 고려해야 합니다 (예: 라운딩 차이로 인해 일부 행에 필터가 적용되지 않아 더 많은 행이 반환될 수 있음).

애플리케이션 마이그레이션

이는 기존 애플리케이션 소스 코드를 가져와 Snowpark에서 사용하기 위해 변환하는 단계입니다. 전체 애플리케이션을 마이그레이션하는 것이 아니라면 기존 코드에서 Snowpark로 업그레이드하고자 하는 하위 세트만 포함시켜 일부를 분리하는 것을 권장합니다.

Spark에서 Snowpark로의 실제 애플리케이션 마이그레이션의 경우 가능한 접근 방식은 3가지입니다.

- **제거:** 엔드 투 엔드 암호화 및 데이터 마스킹과 같이 Snowflake에서 이미 제공되는 기능을 제공하는 요소입니다.
- **대체:** 일부 애플리케이션 또는 특정 하위 세트 기능은 마이그레이션 후 제한된 가치를 제공할 수 있습니다. 이러한 기능을 이미 Snowflake 데이터 클라우드에서 제공하는 기능으로 대체하는 것이 더 비용 효율적일 수 있습니다.
- **리매핑:** 이 접근 방식은 소형 및 중형 워크로드에 적용할 수 있습니다. 이는 Spark 매핑 패턴을 Snowpark 패턴에 적용하는 것을 의미합니다. Snowpark는 Python과 Scala 언어를 모두 지원하며 호환성이 매우 높은 데이터프레임 API를 제공합니다.

Spark-Scala용 Mobilize.Net SnowConvert 또는 Spark-Python용 Mobilize.Net SnowConvert와 같은 도구는 마이그레이션 계획 프로세스를 단축합니다. 이러한 도구는 기존 소스 코드를 가져와 새롭게 리매핑된 프로젝트로 새로운 폴더를 생성할 수 있습니다. 이는 네임스페이스 또는 Python 가져오기를 조정합니다. 이는 모든 Spark API 사용량을 검사하고 이를 동등한 Snowpark API에 매핑할 수 있습니다. 또한, 필요할 경우 도구는 Spark와 Snowpark 사이에 다른 행동을 보이는 모든 영역에서 발생하는 변화의 횟수를 줄이기 위해 헬퍼를 도입합니다.

이러한 도구는 재설계나 대체가 필요한 영역을 추적하는 데 도움이 될 수도 있습니다. 그리고 많은 파티션 작업을 제거할 수 있으며 다른 작업을 Snowflake 윈도우 함수에 매핑할 수 있습니다. 이는 파티션 및 순서화에 대한 지원을 제공합니다.

그뿐만 아니라, 이러한 도구는 재작성이 필요할 수 있는 패턴에 대한 경고를 도입합니다. 여기에는 Spark-Scala용 Mobilize.Net SnowConvert 또는 Spark-Python용 Mobilize.Net SnowConvert 지식 베이스와 연결된 링크가 포함됩니다. 이는 패턴 라이브러리, 샘플 코드 조각 및 참조 소스 코드를 제공하여 일반적인 사용 사례의 모범 사례를 안내합니다.

마이그레이션 테스트

이는 Snowpark로 마이그레이션할 때 가장 중요한 단계라고 할 수 있습니다. 마이그레이션 테스트는 두 부분으로 계획할 것이 권장됩니다.

- **스모크 테스트:** 이 테스트는 소규모 데이터 하위 세트로 수행할 수 있습니다. 일반 로직이 동등하다는 것을 검증하고 적용된 변화에 기본 검증을 수행하기 위해 워크로드를 처음부터 끝까지 실행하기 위한 것입니다.
- **사용자 수용 테스트:** 이 테스트는 마이그레이션된 워크로드를 프로덕션과 가까운 환경으로 실행해야 하기에 더 복잡합니다. 이렇게 하는 이유는 대상 데이터 행 카운트 또는 워크로드 총 실행 시간이 허용 오차 내에 있는지와 같은 중요한 성공 요인을 결정하여 벤치마킹하기 위해서입니다.

SPARK보다 높은 SNOWFLAKE의 가치

데이터의 양이 늘어나고 속도가 빨라짐에 따라 데이터 과학자, 분석 전문가, BI 전문가, 데이터 엔지니어는 최신 워크로드를 위해 특별히 설계된 도구가 필요합니다(동시 성능, 확장성, 가치 및 보안을 위해 설계됨). 목표를 세우고 마이그레이션 프로젝트의 성공을 측정하기 위해 추적할 수 있는 일반적은 메트릭을 다음과 같습니다.

메트릭	현재 상태	미래 상태
데이터 유효성 검사 테스트 가정(최소한의 차이)	레거시 실행에 따라 데이터 세트 캡처	데이터 세트와 변환된 Snowpark 결과 일치
성능 • 처리 시간 및 빈도 • 최적화, 데이터 파이프라인 간소화 • 데이터 가용성	현재 벤치마크 캡처	Snowflake에서 처리되는 모든 처리, 개선된 성능 제공
비용 절감 • 실행 시간 • 지원 및 유지 관리 운영 라이선스 비용	Spark 인프라 및 인력 비용에 따라 현재 지출 캡처	간소화된 아키텍처를 위해 비용 절감

마이그레이션에 도움이 필요하신가요?

Snowflake는 마이그레이션 속도를 개선하고, 계획과 구현 활동을 구성 및 최적화하고, 여러분의 테크놀로지 및 비즈니스 목표를 충족하는 데 도움이 되는 고객 모범 사례를 적용하기 위해 준비되어 있습니다.

Snowflake 프로페셔널 서비스는 데이터 아키텍처 전문 지식과 플랫폼에 대한 고급 기술 지식을 강력하게 결합하여 배포합니다. 이를 통해 고성능 데이터 전략, 개념 증명 및 마이그레이션 프로젝트를 제공합니다.

또한, **Mobilize.Net**과 같은 Snowflake 기술 솔루션 파트너는 개념 증명 및 플랫폼 마이그레이션을 수행하는 방면에서 엄청난 경험을 가지고 있습니다. 기술 솔루션 파트너는 높은 수준의 아키텍처 권장 사항부터 마이그레이션 프로세스를 자동화하고 단축하기 위한 도구까지 다양한 서비스를 제공합니다.

여러분의 조직이 플랫폼 마이그레이션을 위한 모든 직원을 보유하고 있든 추가적인 전문 지식이 필요하든, Snowflake와 솔루션 파트너는 Snowflake 데이터 클라우드로의 여정을 단축하는 데 도움이 되는 기술과 도구를 보유하고 있습니다.따라서 여러분은 혜택을 더 빠르게 얻을 수 있습니다.

SNOWFLAKE 소개

Snowflake와 함께하면 모든 조직이 Snowflake의 데이터 클라우드를 통해 데이터를 집결할 수 있습니다. 고객은 데이터 클라우드를 사용하여 사일로화 된 데이터를 통합하고, 데이터를 검색하고 안전하게 공유하며, 다양한 분석 워크로드를 실행합니다. 데이터 또는 사용자가 어디에 있든 Snowflake를 통해 여러 클라우드와 지역에 걸쳐 단일 데이터를 경험합니다. 2023년 4월 30일 일요일 기준, 2022년 Forbes Global 2000 기업(G2K) 중 590개사를 비롯한 다양한 산업 분야의 수천여 고객이 Snowflake 데이터 클라우드를 사용하여 비즈니스를 강화하고 있습니다.

자세한 내용은 **snowflake.com**에서 확인할 수 있습니다.

MOBILIZE.NET 소개

Mobilize.Net은 세상에서 가장 정확한 소스 코드 번역 테크놀로지를 구축합니다. 수백만 명의 개발자가 Mobilize.Net 테크놀로지를 사용하여 수십억 줄의 코드를 성공적으로 현대화했습니다. Mobilize.Net 솔루션을 통해 고객은 위험을 낮추고, 비용을 절감하고, 시간을 절약하는 동시에 애플리케이션을 오늘날 플랫폼 비즈니스의 요구에 맞춰 변화시킬 수 있습니다. Mobilize.Net SnowConvert 마이그레이션 테크놀로지는 Snowflake가 Snowflake 프로페셔널 서비스 마이그레이션 노력을 지원하기 위해 선택한 솔루션입니다. 비공개 회사인 Mobilize.Net은 워싱턴주 벨뷰에 기반을 두고 있습니다.

자세한 내용은 **www.mobilize.net**에서 확인할 수 있습니다.

© 2023 Snowflake Inc. All rights reserved. 여기에 언급된 Snowflake, Snowflake 로고 및 기타 모든 Snowflake 제품, 기능 및 서비스 이름은 미국 및 기타 국가에서 Snowflake Inc.의 등록 상표 또는 상표입니다. 여기에 언급되거나 사용된 기타 모든 브랜드 이름 또는 로고는 식별 목적으로만 사용되며 해당 소유자의 상표일 수 있습니다. Snowflake는 그러한 소유자와 연관되거나 후원 또는 보증을 받지 않습니다.

인용

[1] bit.ly/38QnHYD