DATABRICKS

Azure에 최적화 된 Spark 기반 데이터 분석 플랫폼

SaaS (하드웨어, 가상서버, 소프트웨어를 모두 지원하는 서비스)에 가까운 서비스 형태

SNOWFLACK - DATA WAREHOUSE 기반의 회사

데이터 웨어하우스- 목적에 맞게 분류된 데이터 창고로서 정형 데이터를 보관한다.

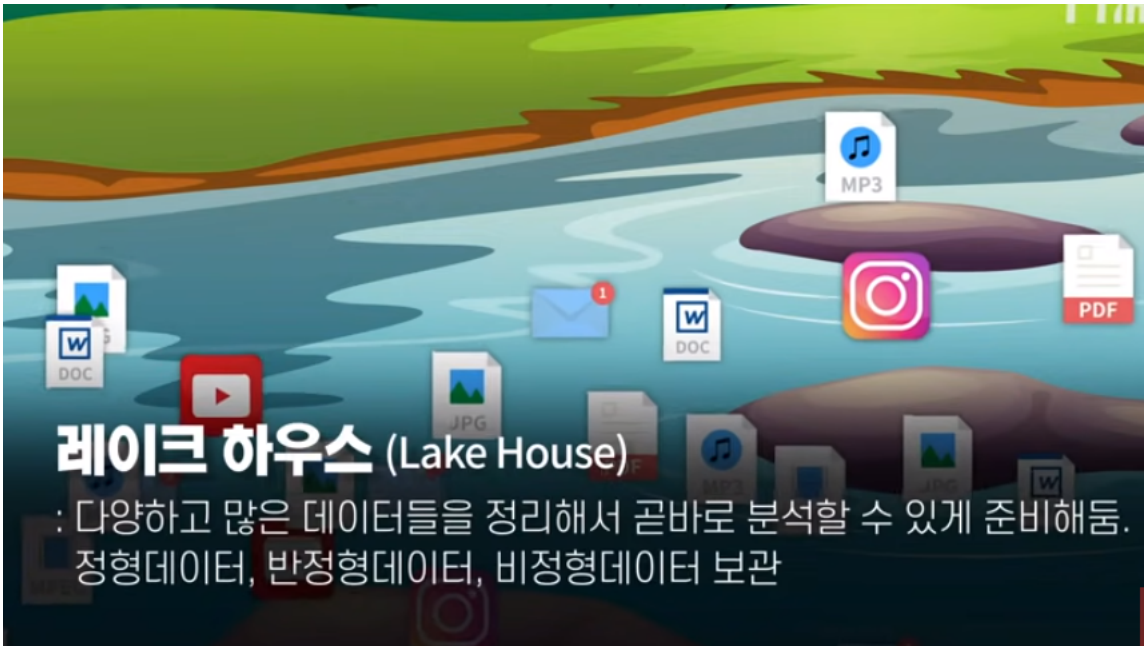
분석 과정에 특화되어 있음.

DATABRICKS - LAKE HOUSE 기반의 회사

DATA LAKE - 목적이 없는 데이터까지 무분별하게 모으게 되는 빅데이터 시대의 현상

LAKE HOUSE - DATA LAKE의 데이터들을 정리해서 곧바로 분석할 수 있게 준비해두는 것

저장 처리 과정에 특화되어 있음



***서로가 다른 과정에 특화되어 있어 두 기능을 같이 활용하기도 한다.**

DATABRICKS 사용하는 이유(요약)

1.작업의 동시성 (하나의 노트북에서 여러명이 글 작성하던거 생각하시면 됩니다.)

2.데이터 가용성 확장

데이터 웨어하우스에서 정형데이터만 보관가능했던 것을 넘어 반정형, 비정형까지 저장이 가능

3.실시간 분석 기능

데이터를 실시간으로 받아 동시에 분석이 가능한 작업 환경

4.손쉬운 접근성

기존 pyspark처럼 파이썬을 spark에서 이용할 수 있었던 거랑 같은 느낌입니다.

Databricks 주요 명령어

```
#Mount - Azure Blob Storage 파일 공유를 탑재하는 방법
dbutils.fs.mount(
  source = "wasbs://firstdata@hanml2storage1.blob.core.windows.net",
  mount_point = "/mnt/samtest",
  extra_configs = {"fs.azure.account.key.hanml2storage1.blob.core.windows.net":"3IUl5pnPF6XjRJp4rfz5Aphg8M/MmQiNZMptt47ATMecbaaviSxxjPDku61

#Data확인 - Mount된 DB 확인
display(dbutils.fs.ls("dbfs:/mnt/samtest"))
#os.listdir('/dbfs/mnt/data')

#DataFram을 SQL Table로 변환하기
#변환 후에는 %sql을 이용해 쿼리문 사용이 가능
.createOrReplaceTempView() #df.createOrReplaceTempView('생성할 테이블 명')
%sql
select * from 지정한 테이블명

#Azure dbfs에 있는 파일 불러오기
spark.read.csv() or spark.read.parquet() ...

#display() - show()와 같은 기능 *DataFrame의 경우에만 출력이 가능
#describe() - 요약 통계량 출력
display(df_train.describe())

#결측치 drop
df_train.na.drop()

#printSchema() - 모든 칼럼 tpye 출력
df_train.printSchema()

#withColumnRenamed(칼럼명, 수정할 칼럼명) - 칼럼명 변경
df.withColumnRenamed("y_0", "label")

#NoteBook에서 DBFS 파일 나열하기
%fs ls /my-file-path

#json 파일 데이터 형식 및 열 이름 조회
spark.read.option("inferSchema", "true").json(jsonFile)

#중복 제거하기
distinct() or dropDuplicates
```

클러스터에 포함되는 드라이버는? - 1개

spark(분산 컴퓨팅 환경)에서 병렬 처리가 발생하는 수준은? - 실행기 및 슬롯

드라이버는 어떤 프로그램 기반인가? - JAVA

Databricks에 사용되는 Notebook 형식은 다음 중 어느 것인가요? - .dbc

Azure Databricks 작업 영역에 새 클러스터를 만들 때 백그라운드에서 수행되는 작업은 무엇인가요?

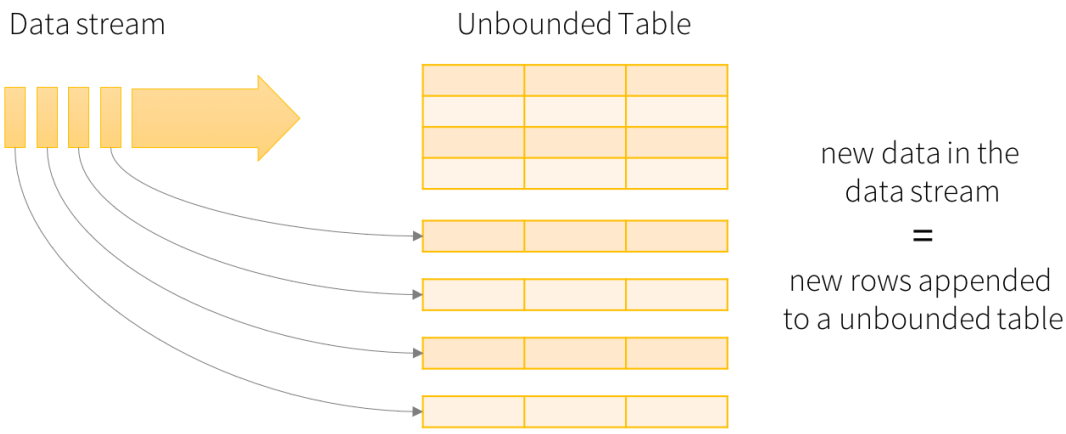
-Azure Databricks는 VM 유형 및 크기 선택에 따라 드라이버 및 작업자 노드의 클러스터를 생성합니다.

프로그래밍 모델 - Structed Streaming data의 핵심 아이디어

실시간 데이터를 지속적으로 추가하는 테이블로 처리하는 것

기본 개념

새롭게 추가되는 input data를 입력 테이블의 새 행이 추가되는 것



Data stream as an unbounded table

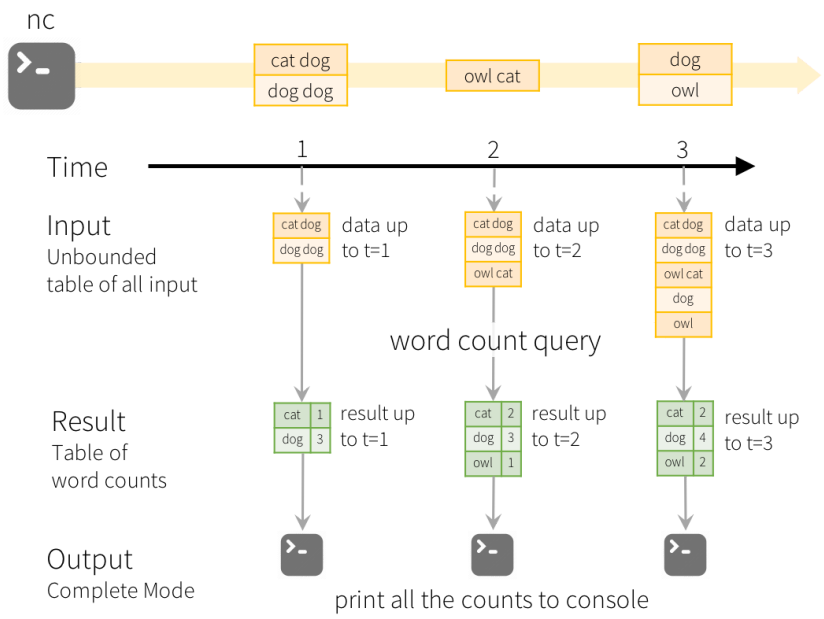
설정된 Trigger 값에 따라서 데이터가 Result Table에 업데이트 된다.

Output은 결과를 출력하는 외부 저장소로 정의

Complete Mode - 기존 + 업데이트 된 행 전체를 출력하는 모드

Append Mode - 새롭게 업데이트 된 행만 출력하는 모드

Update Mode - 마지막 트리거 결과에서 업데이트 된 행만 출력하는 모드



Model of the Quick Example

***위 그림의 first step은 기존에 알고 있는 DataFrame과 정확히 동일한 형태를 띈다.**

***Structed Streaming data는 전체 테이블을 구체화하지 않는다.**

***위 그림과 같이 점진적으로 처리를 실행하며 원본 데이터는 삭제하게 된다.**

***원본 데이터를 삭제한 후 최소환의 중간 과정만을 남겨둔다.**