데이터 엔지니어링 적재

아파치 하이브

Park Suhyuk


psyoblade


psyoblade

hive 2.3.2
docker-ce 19.03.13
ubuntu 20.04 LTS

NCSOFT®

목차

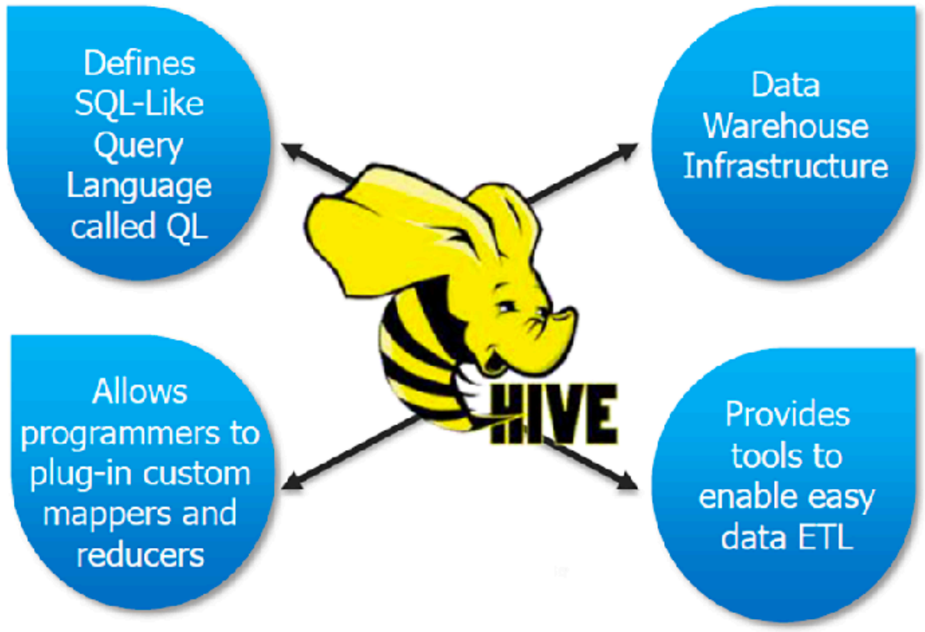
Introduction

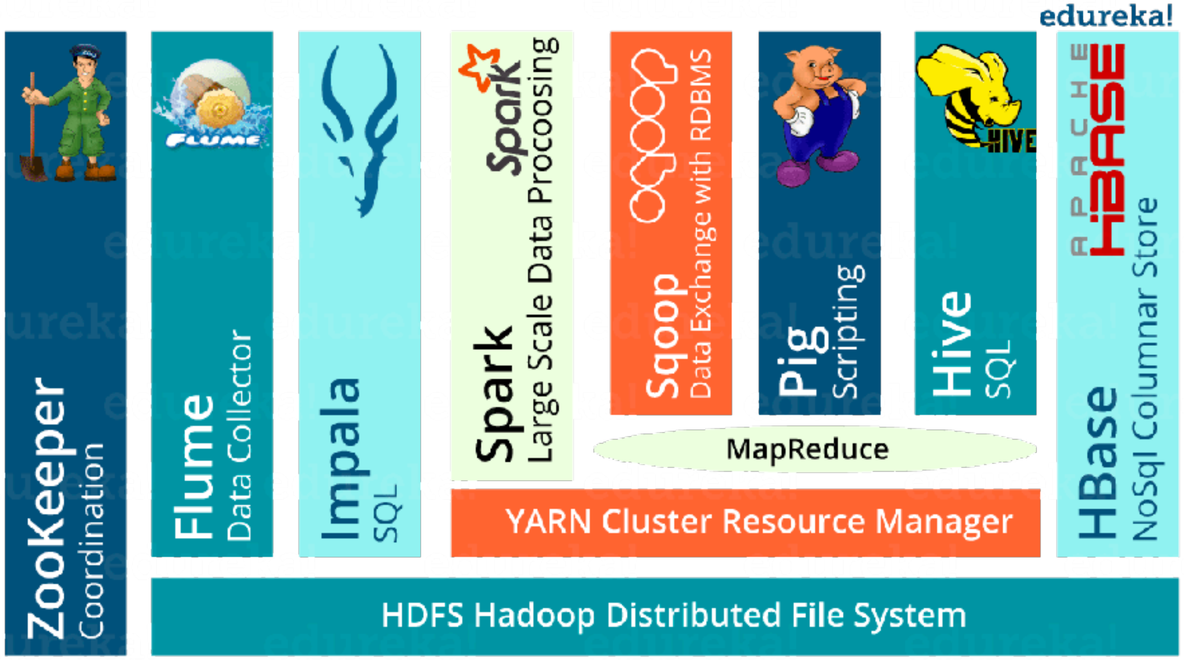
아파치 하이브

아파치 하이브 – 개요

What is 'Hive' ?

하이브는 분산 저장소에 존재하는 대용량 데이터를 SQL을 통해 관리할 수 있는 SQL on Hadoop 데이터 웨어하우스 엔진입니다. 2010년 부터 Facebook 에서 개발하기 시작하여, 현재는 아파치 프로젝트인 하둡 기반의 데이터 웨어하우스 프레임워크로써 복잡한 MapReduce 코드 대신 SQL 만으로 대용량 데이터 분산처리가 가능하여, 기존의 DW 기반의 BI 개발자들도 손쉽게 사용할 수 있으며, SQL 에서 제공하는 다양한 함수 및 Optimizer 등이 거의 대부분 적용되어 확장성과 유지보수성이 뛰어납니다.





아파치 하이브 – 사용 예제 (1/2)

Hello Hive using Beeline

Beeline Example

```
% bin/beeline
Hive version 0.11.0-SNAPSHOT by Apache
beeline> !connect jdbc:hive2://localhost:10000 scott tiger
!connect jdbc:hive2://localhost:10000 scott tiger
Connecting to jdbc:hive2://localhost:10000
Connected to: Hive (version 0.10.0)
Driver: Hive (version 0.10.0-SNAPSHOT)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://localhost:10000> show tables;
show tables;
+-------------------+
|     tab_name      |
+-------------------+
| primitives        |
| src               |
| src1              |
| src_json          |
| src_sequencefile  |
| src_thrift        |
| srcbucket         |
| srcbucket2        |
| srcpart           |
+-------------------+
9 rows selected (1.079 seconds)
```

- 테이블이 존재하면 제거하고, 실습을 위한 IMDB 영화(`imdb_movies`) 테이블을 생성합니다

```
# beeline>
drop table if exists imdb_movies;

create table imdb_movies (
  rank int
  , title string
  , genre string
  , description string
  , director string
  , actors string
  , year string
  , runtime int
  , rating string
  , votes int
  , revenue string
  , metascore int
) row format delimited fields terminated by '\t';
```

- 생성된 테이블에 로컬에 존재하는 파일을 업로드합니다

```
load data local inpath '/opt/hive/examples/imdb.tsv' into table imdb_movies;
```

https://cwiki.apache.org/confluence/display/Hive/HiveServer2+Clients#HiveServer2Clients-BeelineExample

아파치 하이브 – 사용 예제 (2/2)

Hello Hive using Beeline

▼ 📘 13. [중급] 2015년도 개봉된 영화 중에서 최고 매출 Top 3 영화 제목과 매출금액을 출력하세요

```
select title, cast(revenue as float) as rev from imdb_movies where year = '2015' order by rev desc limit 3;
```

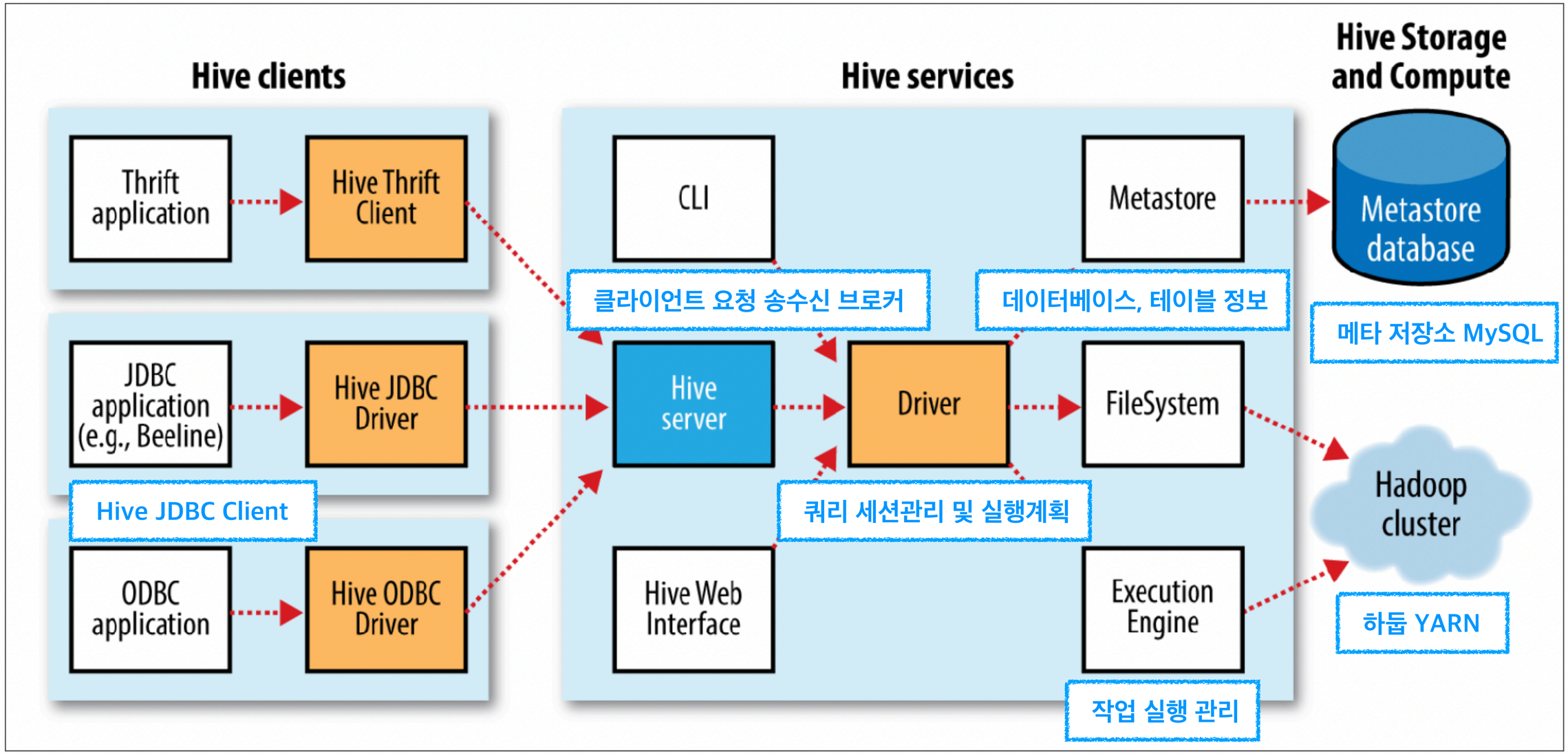
- 아래와 유사하게 나오면 정답입니다

```
+-----------------------------------------+---------+
|                  title                  |   rev   |
+-----------------------------------------+---------+
| Star Wars: Episode VII - The Force Awakens  | 936.63  |
| Jurassic World                          | 652.18  |
| Avengers: Age of Ultron                 | 458.99  |
+-----------------------------------------+---------+
```

Hive Components

아파치 하이브 내부 컴포넌트 소개

아파치 하이브 – 컴포넌트

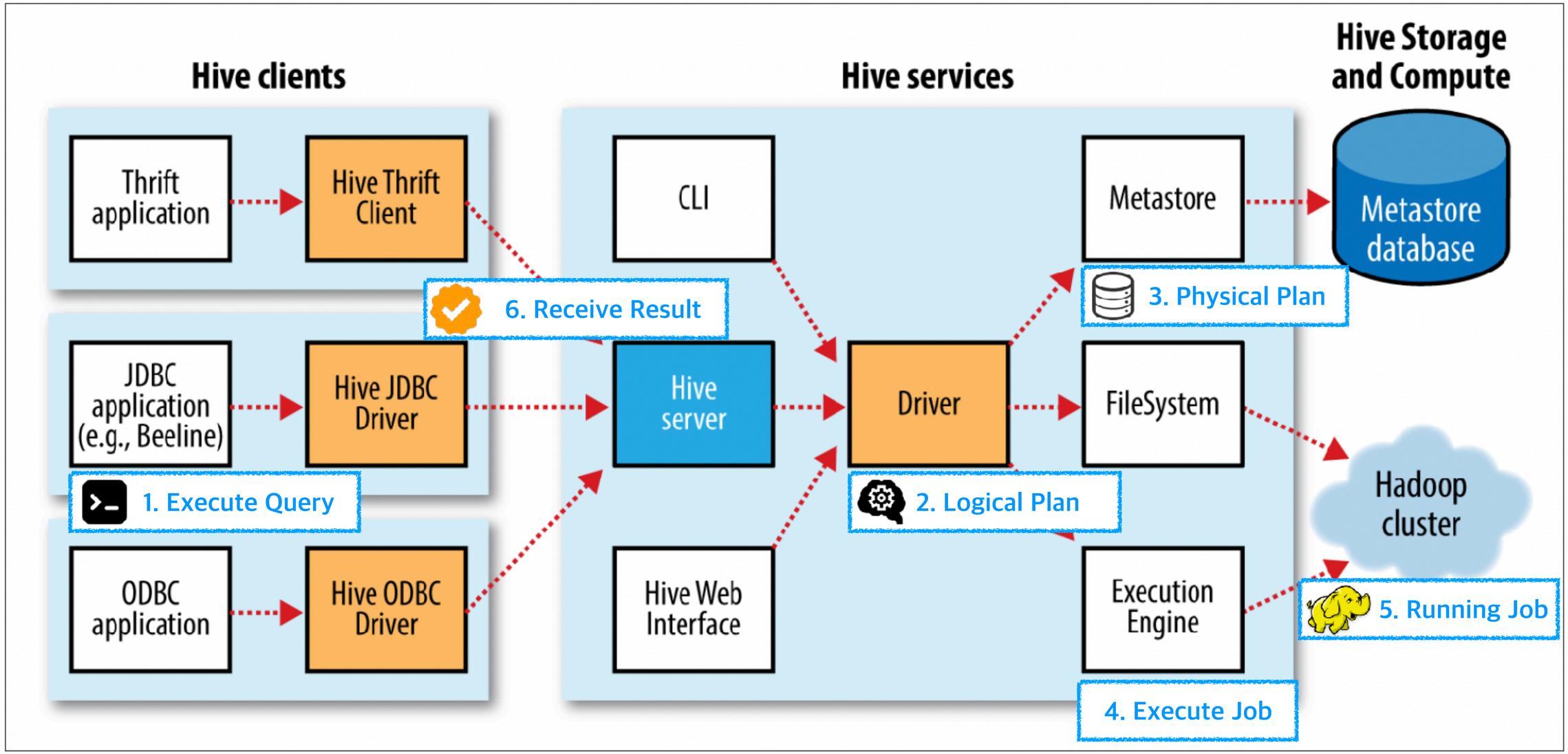


https://www.oreilly.com/library/view/hadoop-the-definitive/9780596521974/

Hive Internal

아파치 하이브 내부 동작 방식

아파치 하이브 – 동작순서



https://www.oreilly.com/library/view/hadoop-the-definitive/9780596521974/

아파치 하이브 – 동작 순서

Hive Internal

Query 수행 절차

1. Execute Query → CLI 혹은 웹 UI 를 통해 HiveQL 을 Driver 로 전달
2. Logical Plan → 세션을 생성하고 컴파일러에 해당 쿼리의 논리 계획을 생성
3. Physical Plan → 메타스토어로부터 메타를 가져와 물리 계획을 생성 및 전달
4. Execute Job → Yarn 혹은 MapReduce 작업을 통해서 실행 계획 수행
5. Running Job and Receive Result → 실행 결과를 반환합니다

Components

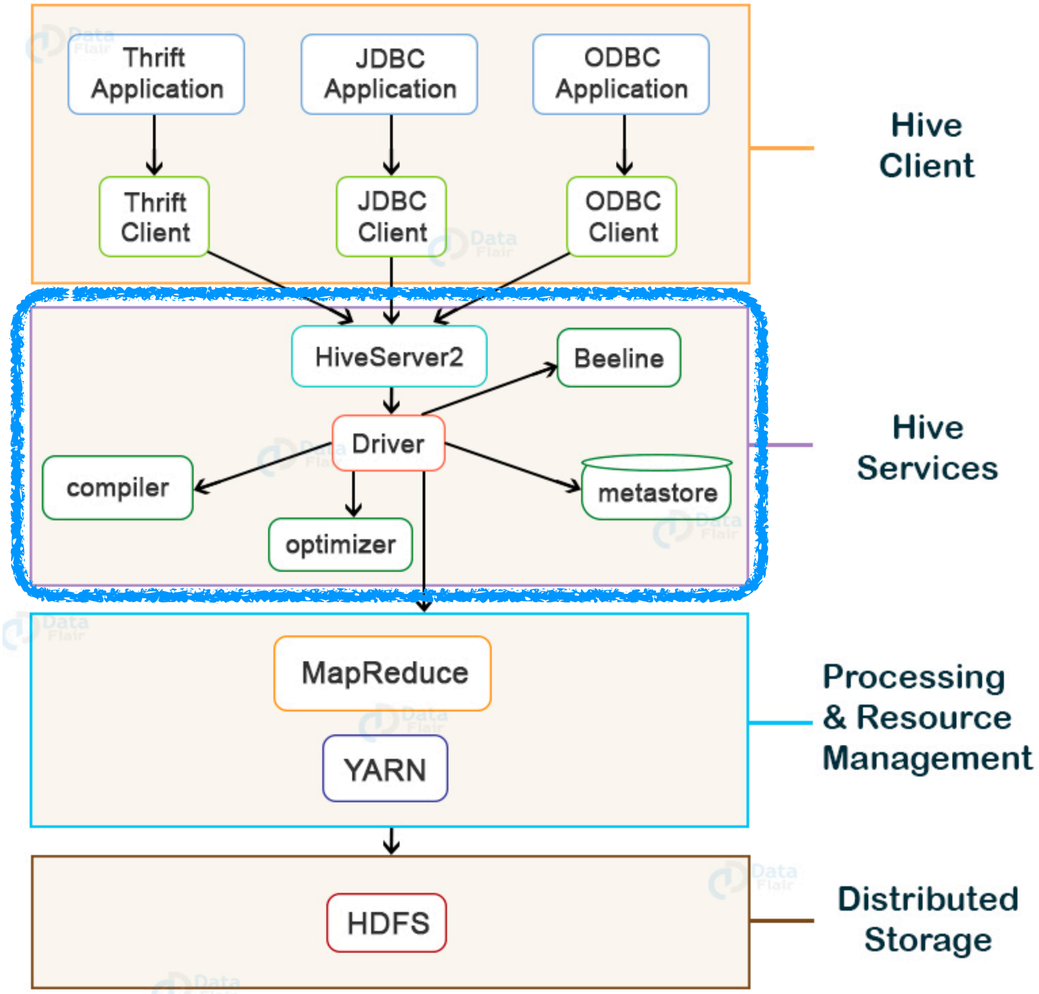
HiveServer2 : 클라이언트 쿼리 수신 및 생성된 결과를 전달하는 브로커

Hive Driver : 요청 쿼리에 대한 세션 생성 및 컴파일러 통한 실행계획 생성

Hive Compiler : 쿼리 구문분석, 데이터 타입 체크 및 메타 통한 물리 수행계획

Optimizer : 성능과 확장성을 검토하여 변환 작업을 통해 최적화 수행

Execution Engine : 물리 계획을 직접 수행하는 엔진



Hive Architecture & Its Components

https://data-flair.training/blogs/apache-hive-architecture/

아파치 하이브 - Hive vs. Spark

What is differences between Hive and Spark

Spark SQL 의 경우 Hive 엔진과 연동 시 Hive 의 Metastore 기반으로 동작하기 때문에 어플리케이션 수준에서 운영 시에 큰 차이점을 느끼지 못 합니다. 특히 JDBC, ODBC 지원이나, OLAP 데이터 처리 지향적이며 Batch 처리에 최적화 되어 있으며, 레코드 단위 업데이트를 지원하지 않는 점 등 Hive QL 과 Spark SQL 은 상당히 유사한 점이 많습니다. 하지만 실행 엔진은 설계 방향이 다르기 때문에 아래와 같은 다른 점들에 유의하여 사용 및 선택할 수 있습니다.

	Hive QL	Spark SQL
실행엔진	MapReduce or Tez 기반의 느리지만 안정적인 SQL 수행이 가능한 엔진	Memory 기반의 Spark Engine 을 통한 빠르지만 상대적으로 다소 민감한 엔진
플러그인	SQL 문법을 기본으로 UDF, UDAF 등은 다양한 언어를 통해 컴파일 후 배포	Scala, Java, Python 및 R 을 통한 런타임 시에 적용 가능
권한설정	유저, 그룹 별 접근 권한을 지정할 수 있습니다.	유저에 대한 권한 지정 기능은 없습니다
스키마	유연한 스키마와 추가된 스키마에 에볼루션 기능을 지원합니다	스키마 관련 기능은 없지만 Parquet 포맷 자체의 스키마 에볼루션을 활용합니다
메타데이터	테이블 수준에서 파티션, 버킷팅을 지원합니다	자체 메타 스토어가 없기 때문에 하이브와 연동하여 테이블 파티션, 버킷팅 지원

수고 하셨습니다