데이터 엔지니어링 교육과정 강의 일정

Apache Spark 소개

01 아파치 스파크란?

02 아파치 스파크의 기본 아키텍처

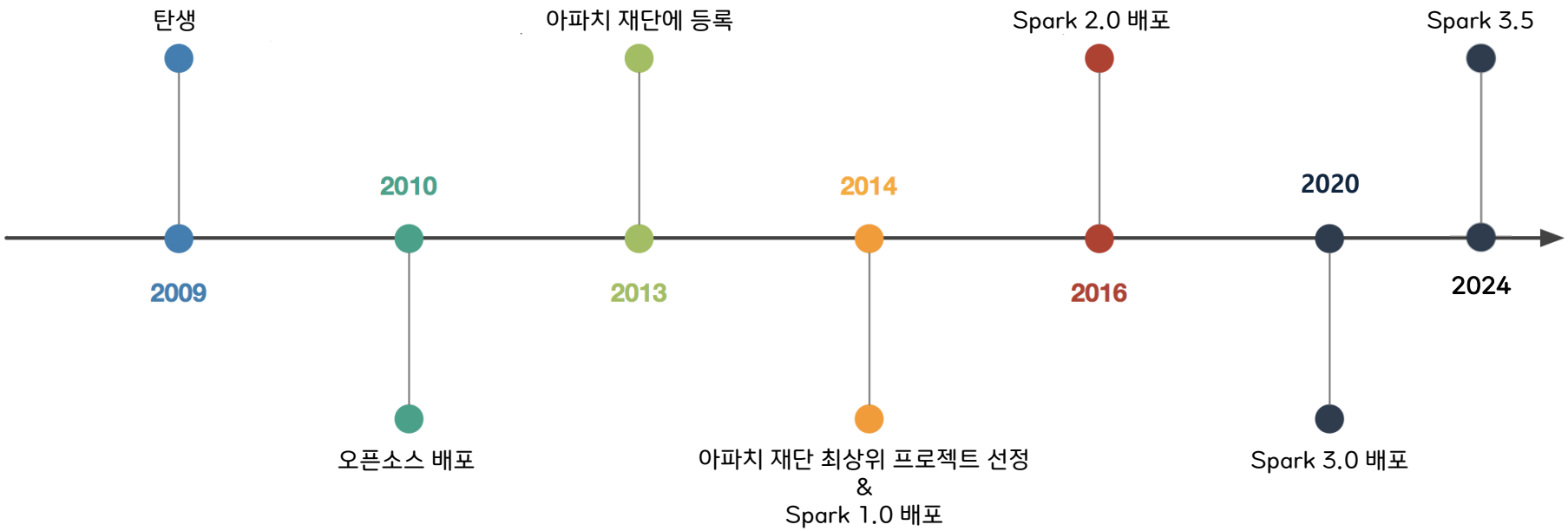
03 아파치 스파크 작업의 특성

04 Pyspark 실습

아파치 스파크란?

Apache Spark

- 빅데이터 처리를 위한 오픈소스 병렬분산처리 플랫폼으로 2009년 UC 버클리 대학교에서 개발



아파치 스파크란?

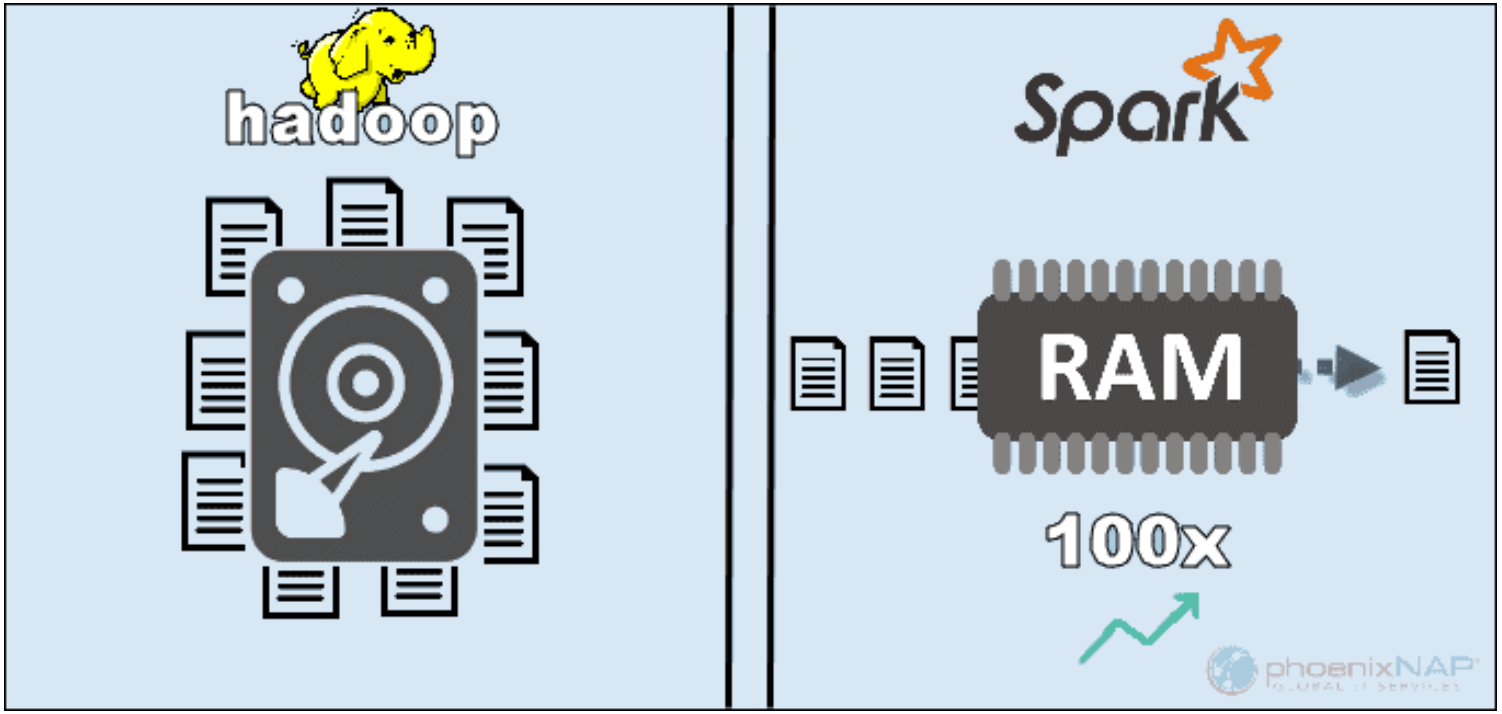
Apache Spark

- 빅데이터 처리를 위한 오픈소스 병렬분산처리 플랫폼으로 2009년 UC 버클리 대학교에서 개발
- 빅데이터 애플리케이션 개발이 필요한 통합 플랫폼을 제공하자! -> 데이터 분석에 필요한 거의 모든 기능 제공

아파치 스파크란?

Apache Spark

- 빅데이터 처리를 위한 오픈소스 병렬분산처리 플랫폼으로 2009년 UC 버클리 대학교에서 개발
- 빅데이터 애플리케이션 개발이 필요한 통합 플랫폼을 제공하자! -> 데이터 분석에 필요한 거의 모든 기능 제공
- 메모리 기반 처리로 디스크 기반 하둡에 비해 10-100배 빠른 데이터 처리 속도



아파치 스파크란?

Apache Spark

- 빅데이터 처리를 위한 오픈소스 병렬분산처리 플랫폼으로 2009년 UC 버클리 대학교에서 개발
- 빅데이터 애플리케이션 개발이 필요한 통합 플랫폼을 제공하자! -> 데이터 분석에 필요한 거의 모든 기능 제공
- 메모리 기반 처리로 디스크 기반 하둡에 비해 10-100배 빠른 데이터 처리 속도
- 현대 빅데이터 분석 업무를 하는 사람들에게는 표준과 같은 도구로 자리매김

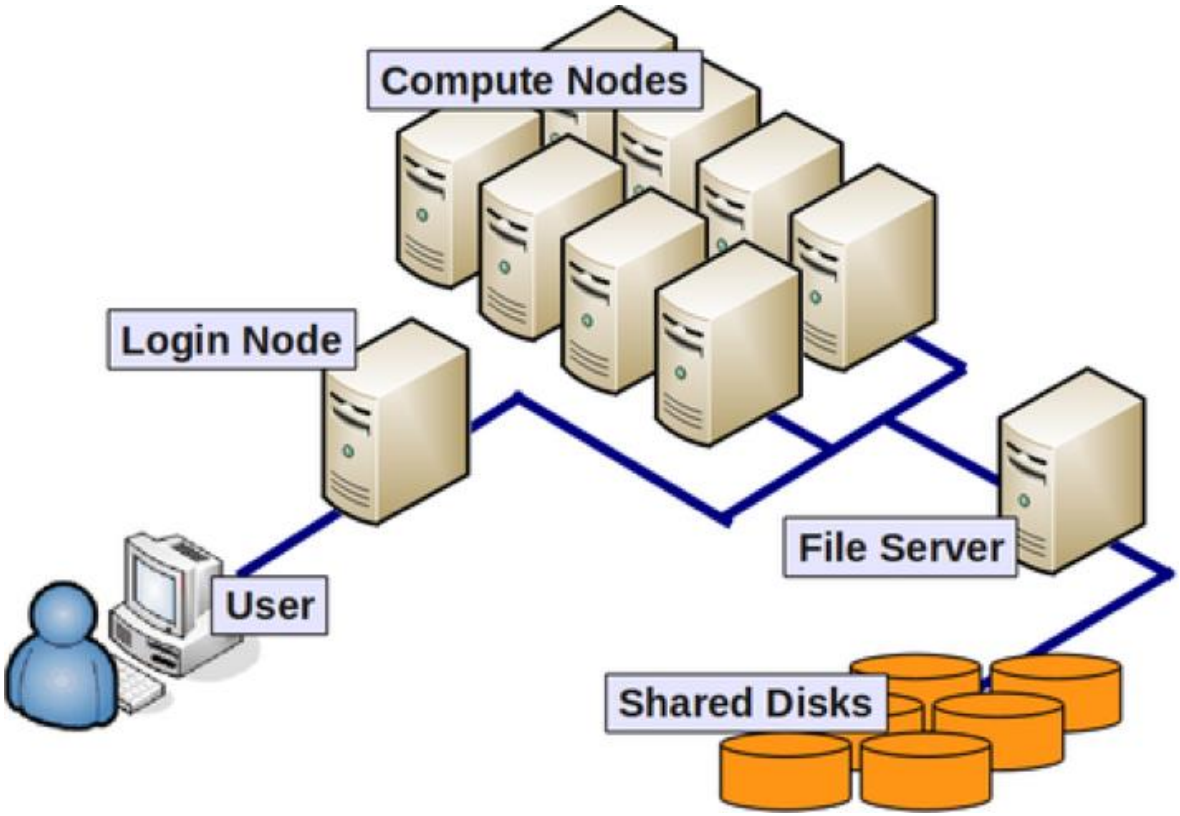
아파치 스파크 관련 용어

Cluster Computing

- 여러 대의 컴퓨터들이 연결되어 하나의 시스템처럼 동작하는 컴퓨터들의 집합
- 클러스터 컴퓨터가 계산을 수행하는 방식을 대략 요약하자면

1. 크고 복잡한 계산을 적당한 크기로 나누고 각 노드(Node)에 배분한다.
2. 각 노드들은 계산을 수행한다.
3. 그 결과를 한 컴퓨터(Frontend)에 수합하고 결과를 사용자에게 반환한다.

-나무위키-

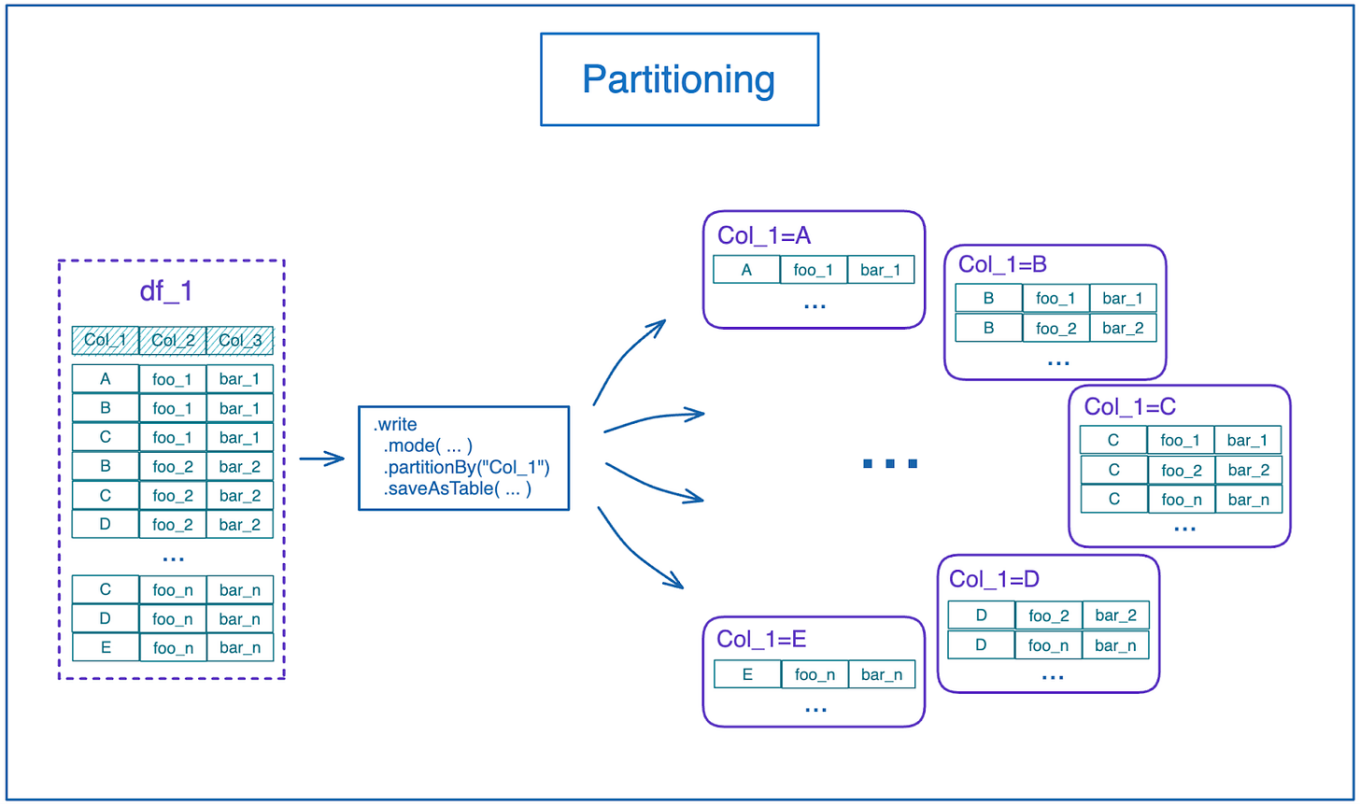


아파치 스파크 관련 용어

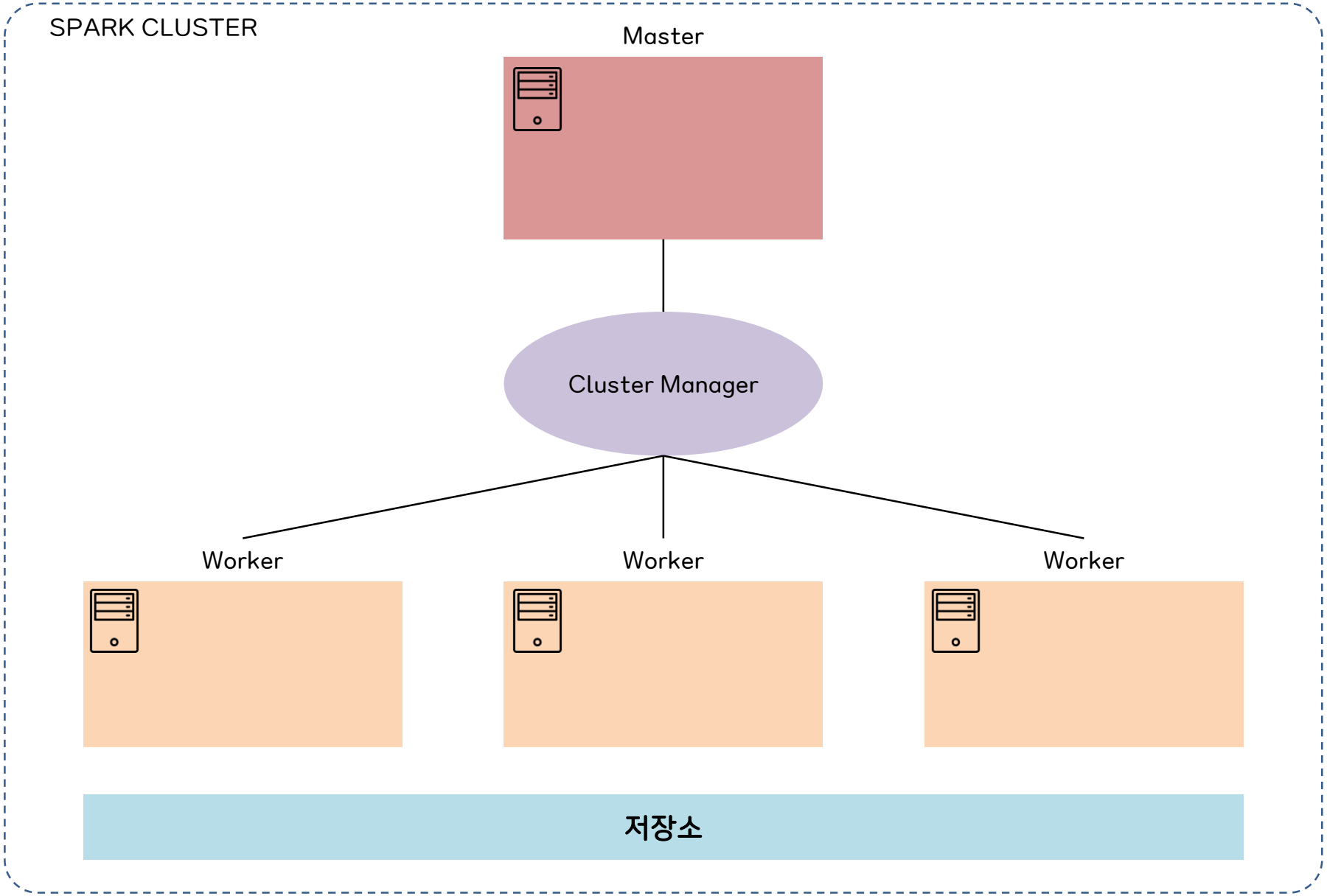
Partitioning (파티셔닝)

- 데이터를 특정 기준에 따라 관리하기 쉬운 partition이라는 작은 단위로 물리적으로 분할하는 것

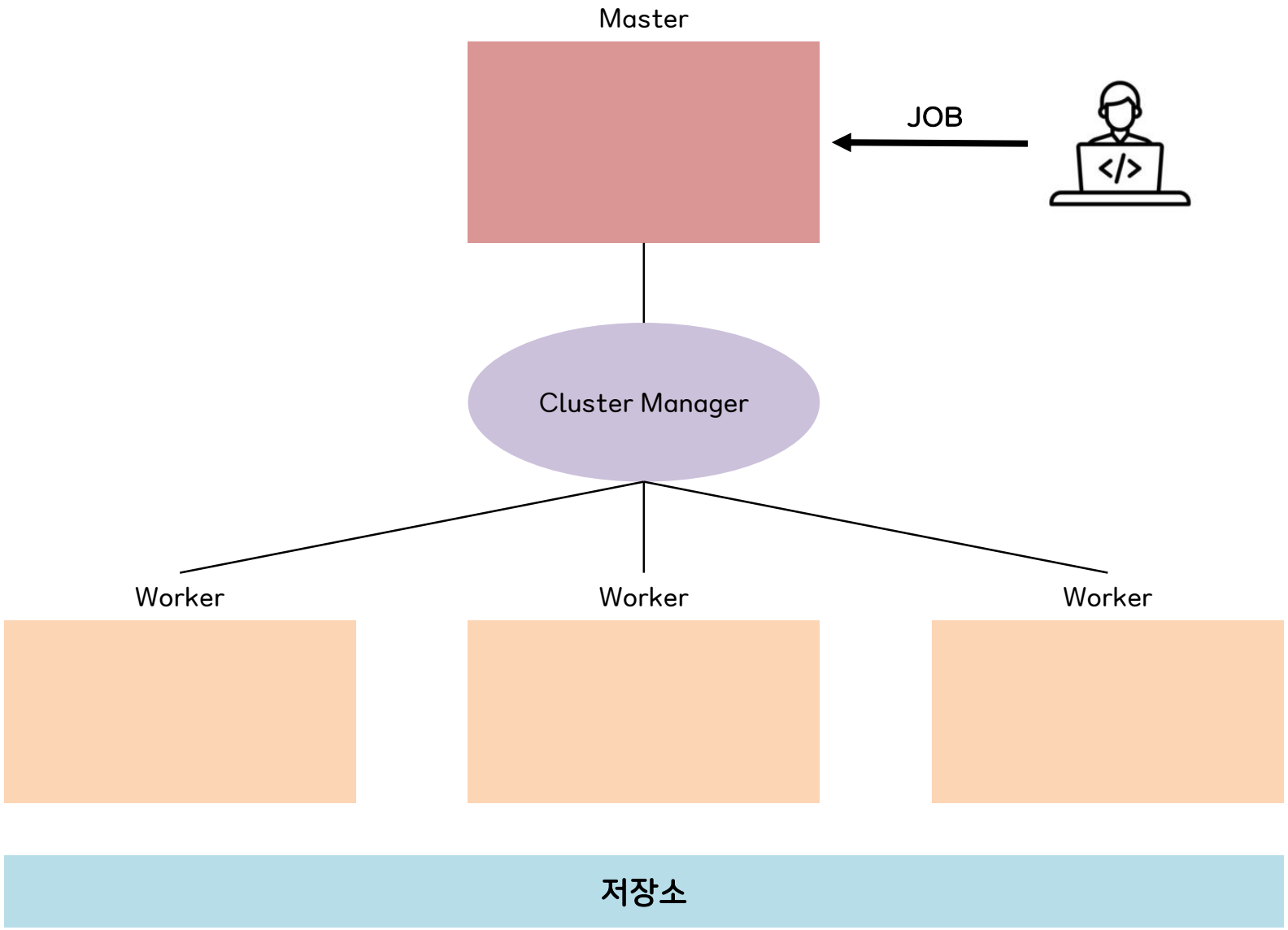
1. 데이터 셔플 최소화를 통한 컴퓨팅 성능 향상
2. 동일한 키의 데이터를 같은 노드에 배치하여 조회 성능 향상



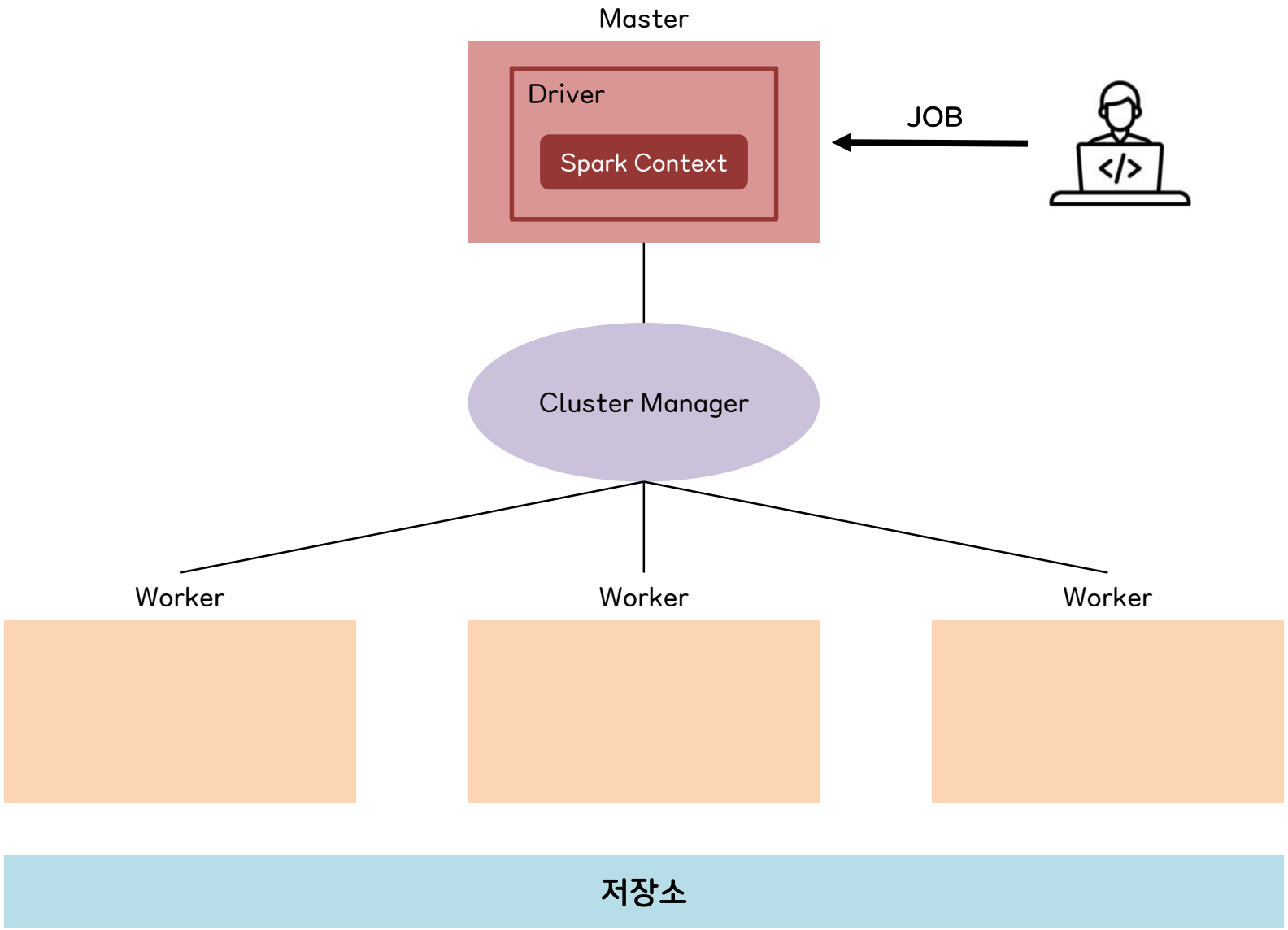
아파치 스파크의 기본 아키텍처



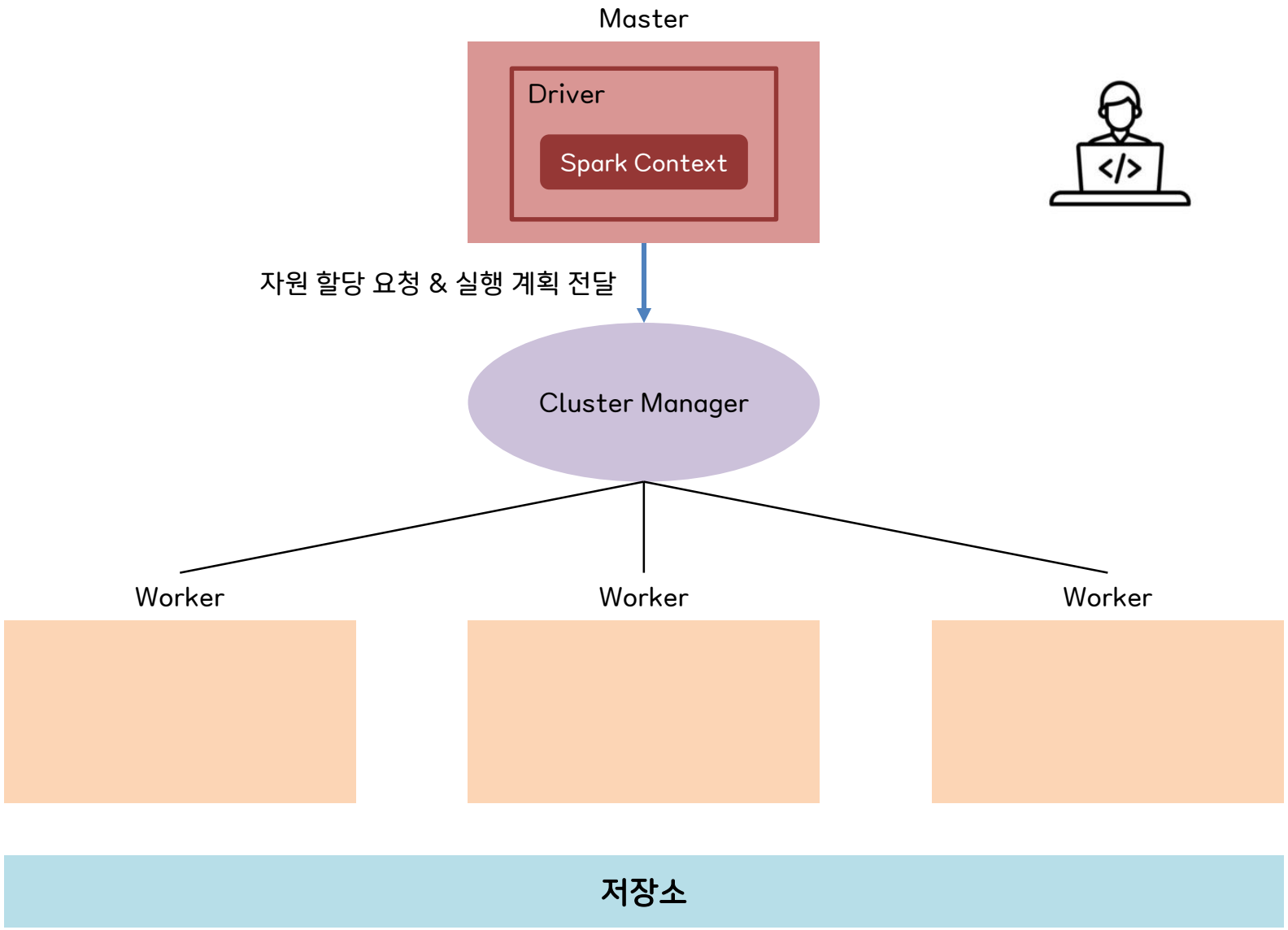
아파치 스파크의 기본 아키텍처



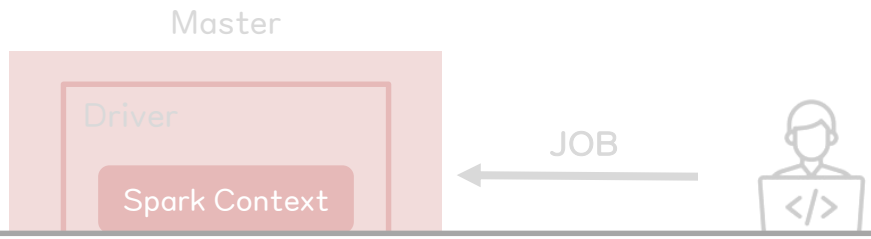
아파치 스파크의 기본 아키텍처



아파치 스파크의 기본 아키텍처



아파치 스파크의 기본 아키텍처



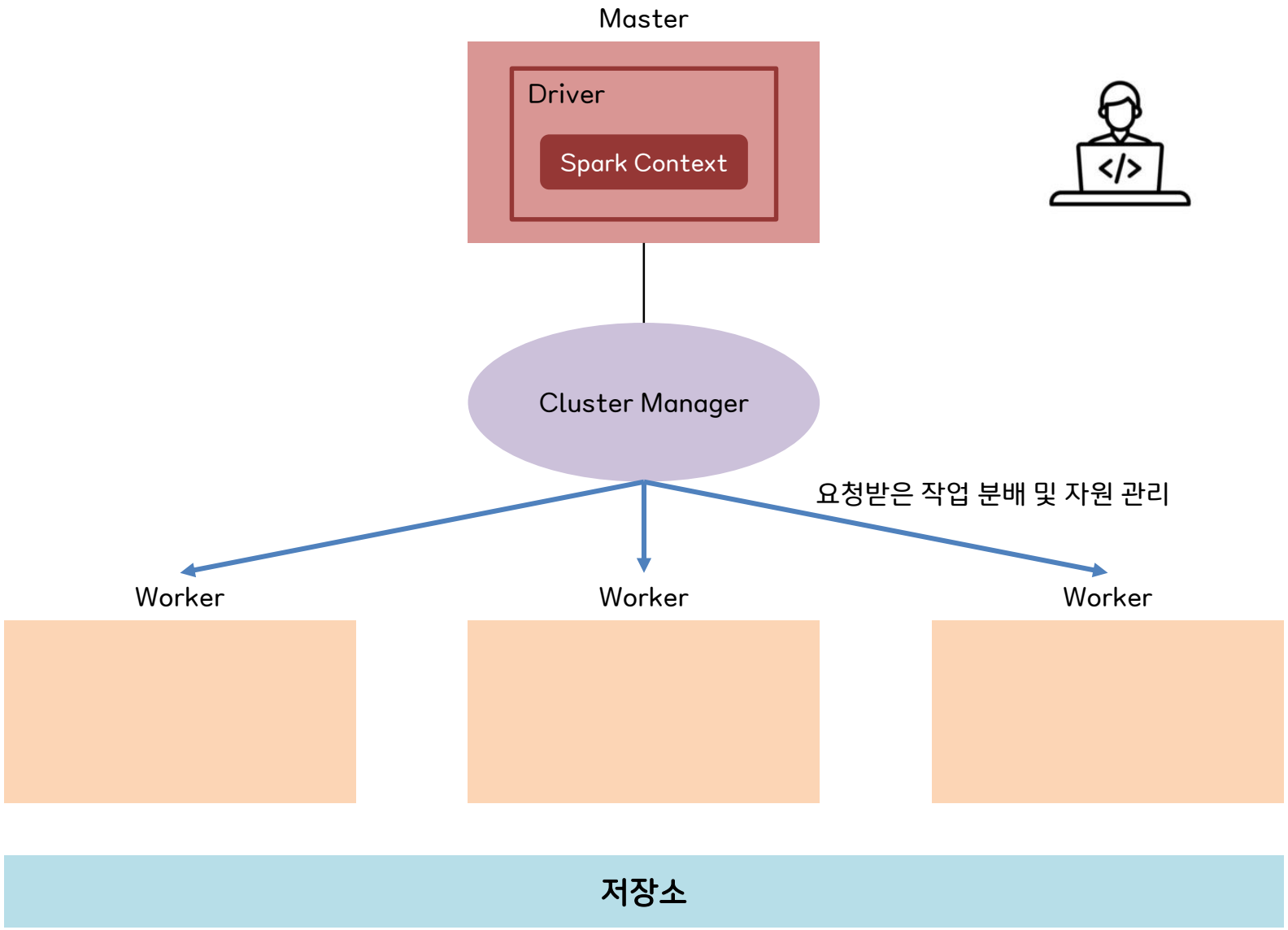
- **Driver**
  - Spark 애플리케이션을 실행하는 프로세스로, 사용자가 작성한 Spark 애플리케이션의 진입점(entry point)
  - Main 함수를 실행하고 애플리케이션의 코드를 분석하고 태스크(task)를 생성하여 클러스터 매니저에 제출

- **Spark Context**
  - Spark 클러스터와의 통신을 관리하는 객체로 클러스터와의 연결을 설정하고 RDD, Accumulators 등 필요 자원 구축

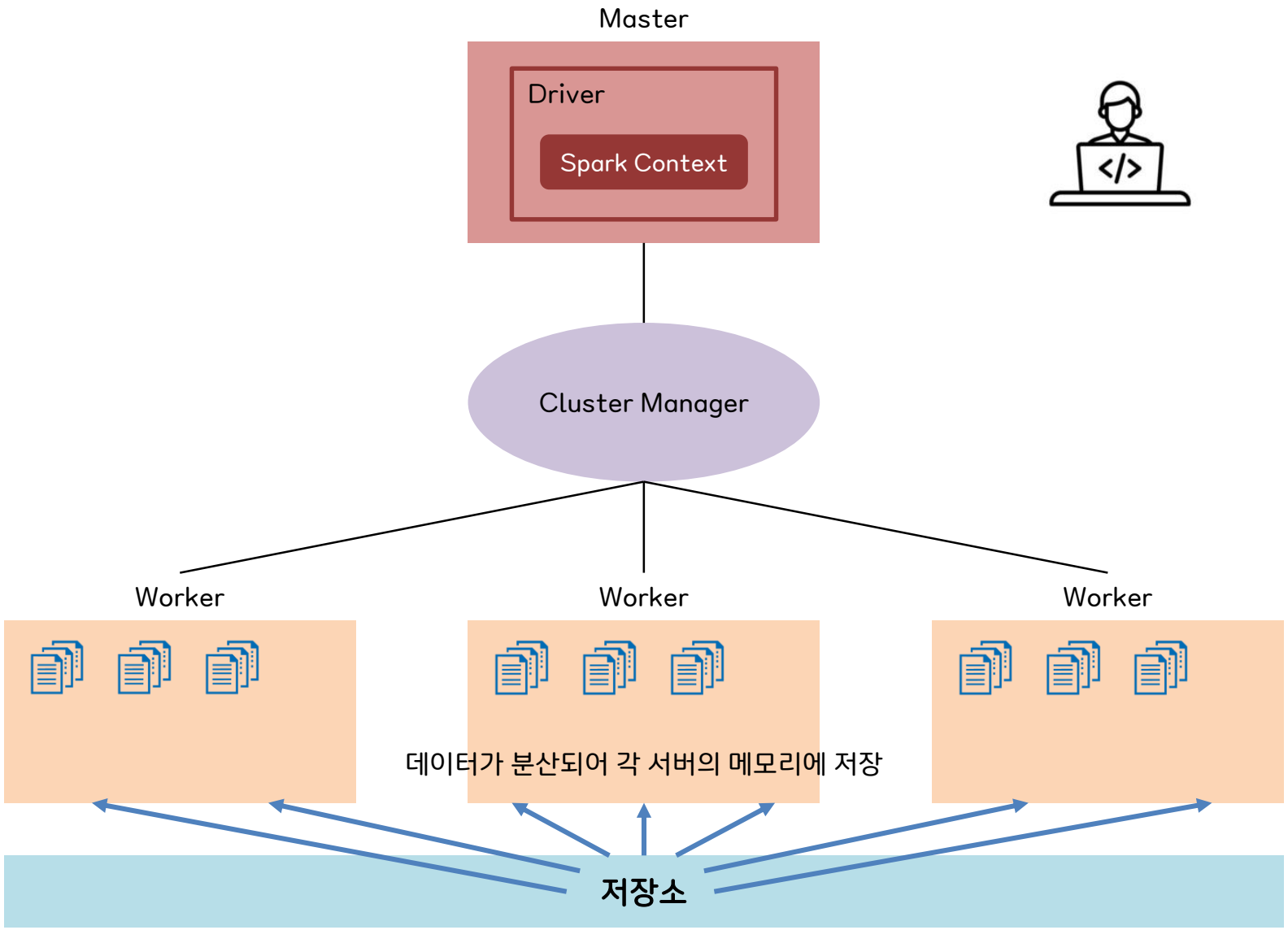
- **Cluster Manager**
  - 클러스터 리소스를 관리하고 Spark 애플리케이션의 Executor를 시작하고 관리
  - Yarn, Kubernetes, Apache Mesos 등이 역할을 수행할 수 있다.

저장소

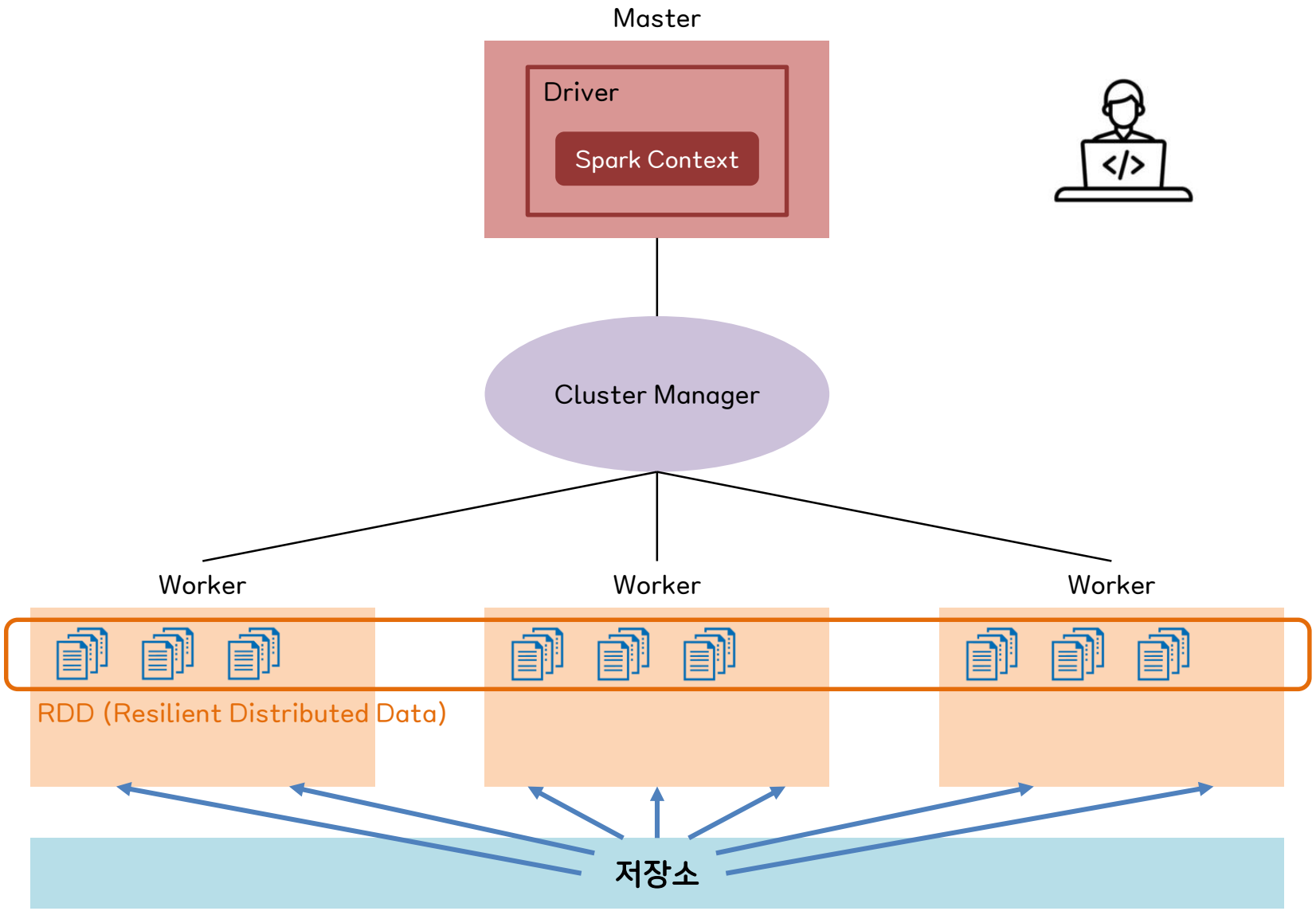
아파치 스파크의 기본 아키텍처



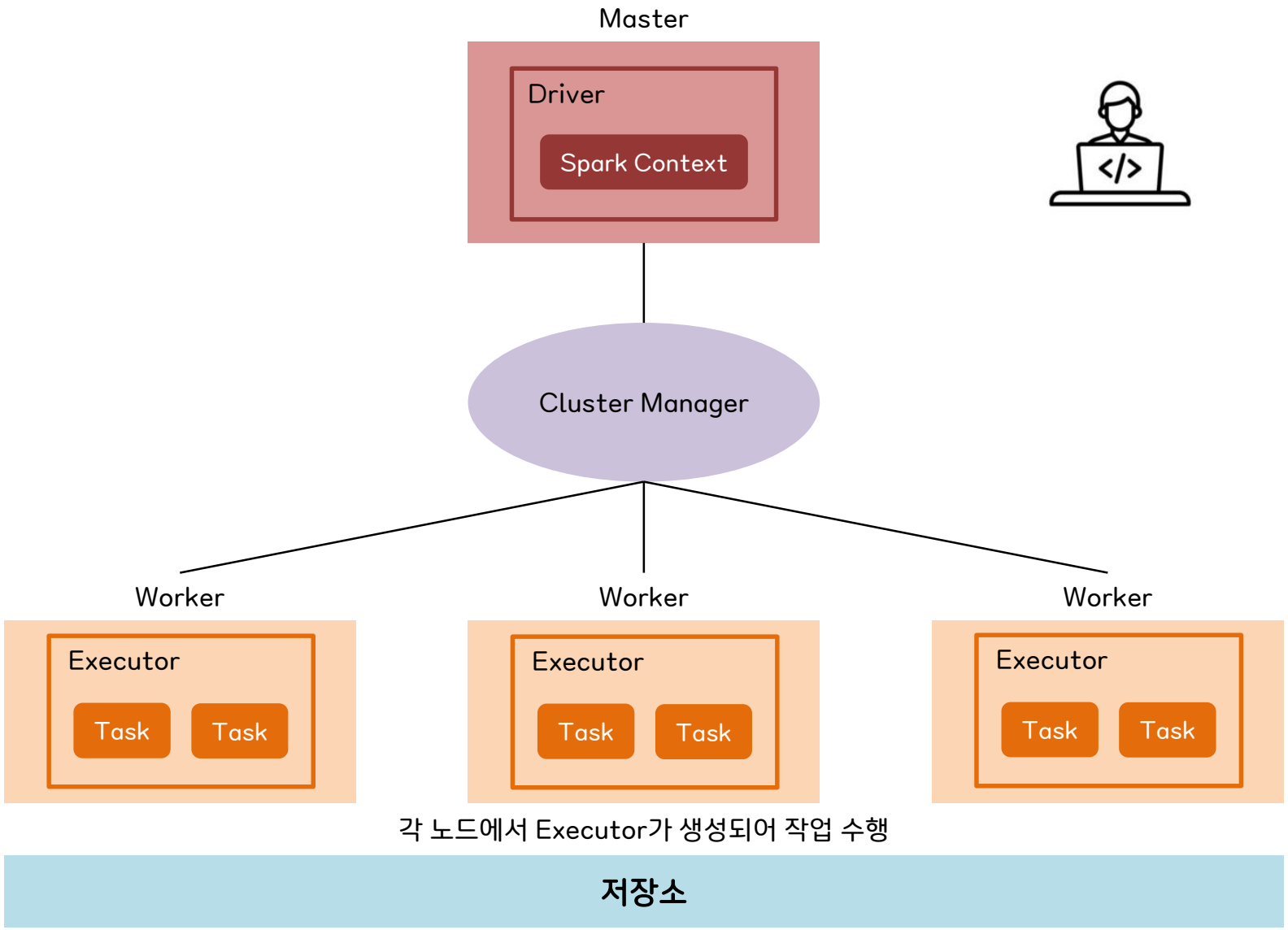
아파치 스파크의 기본 아키텍처



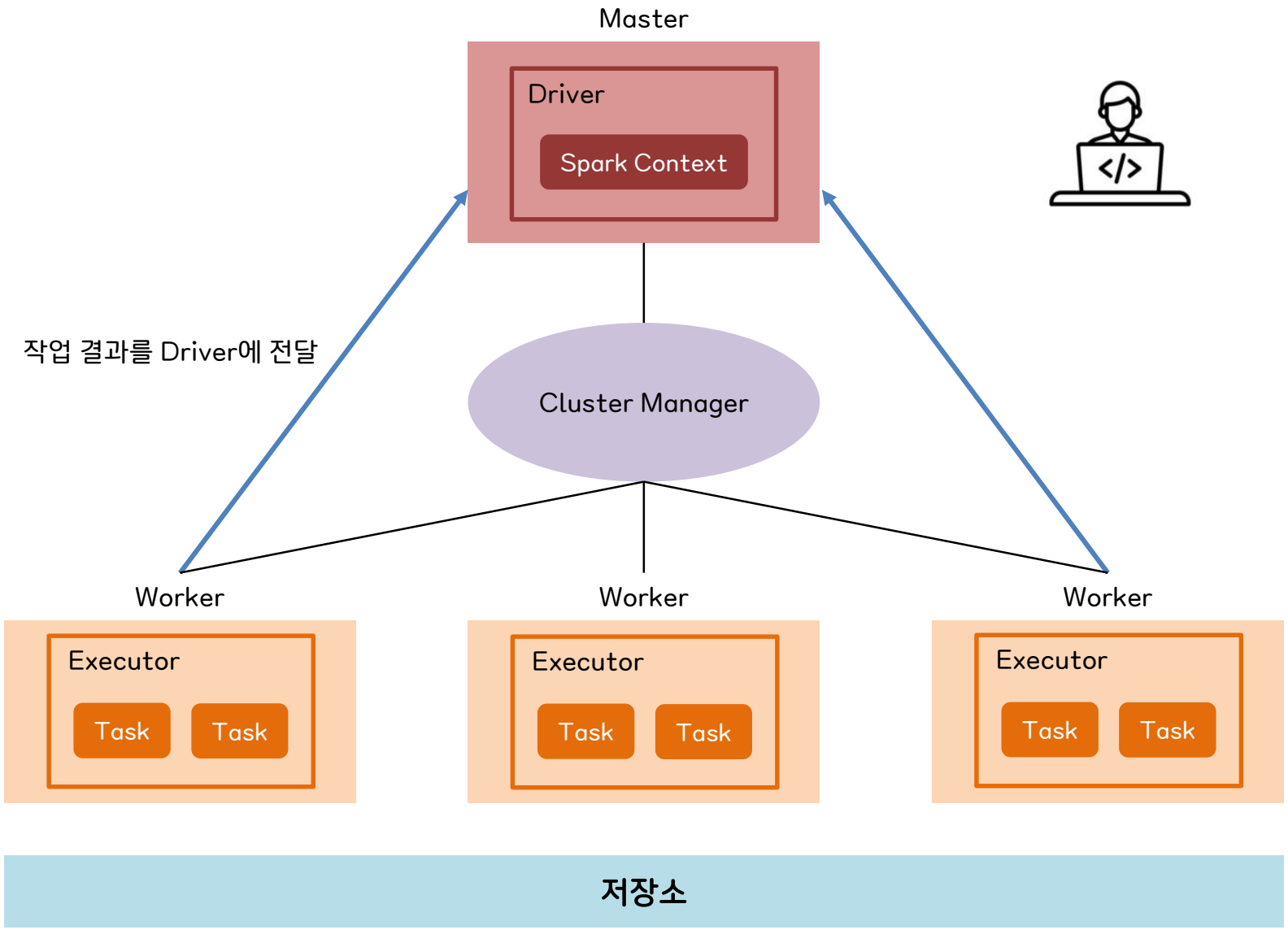
아파치 스파크의 기본 아키텍처



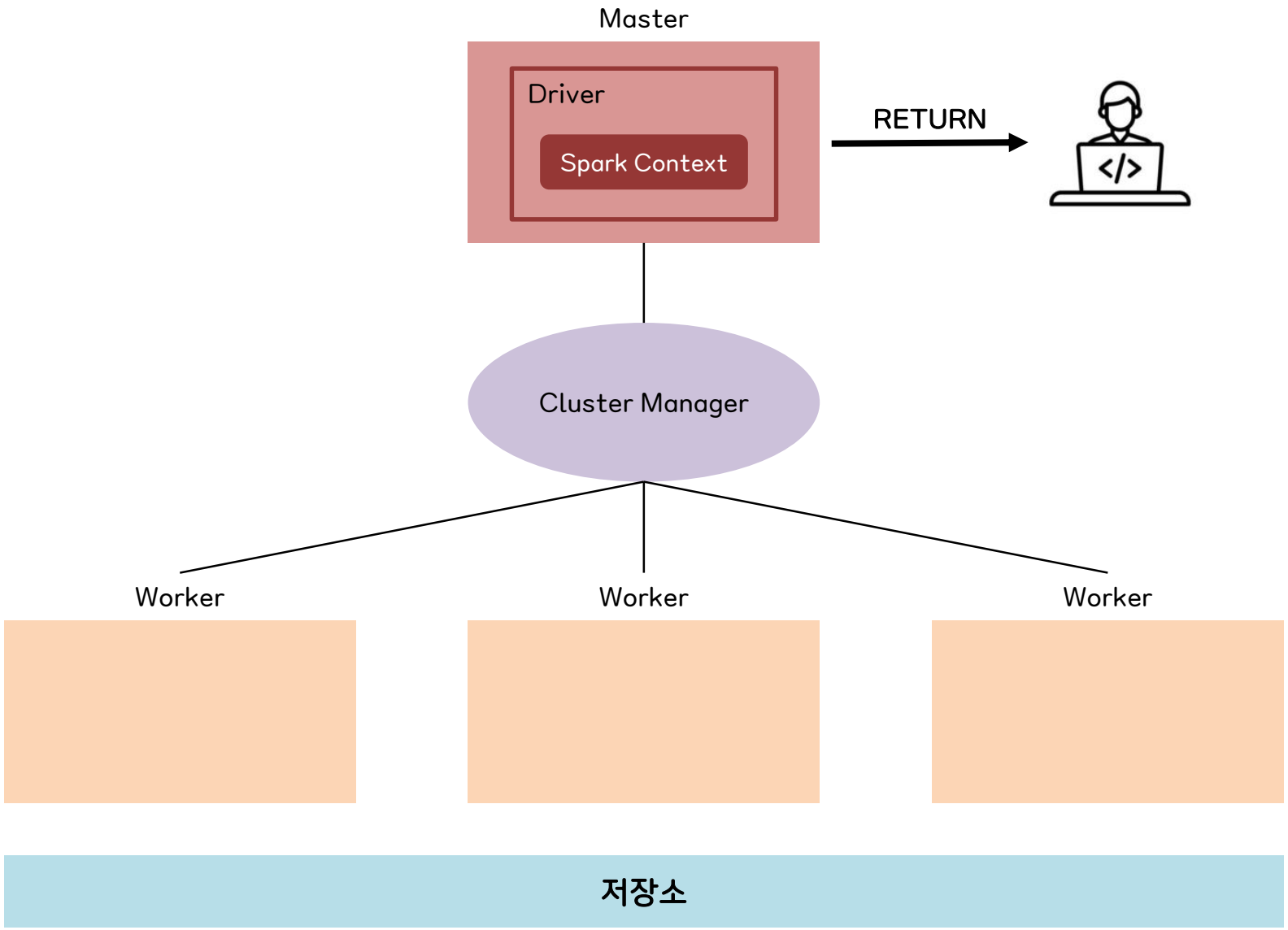
아파치 스파크의 기본 아키텍처



아파치 스파크의 기본 아키텍처



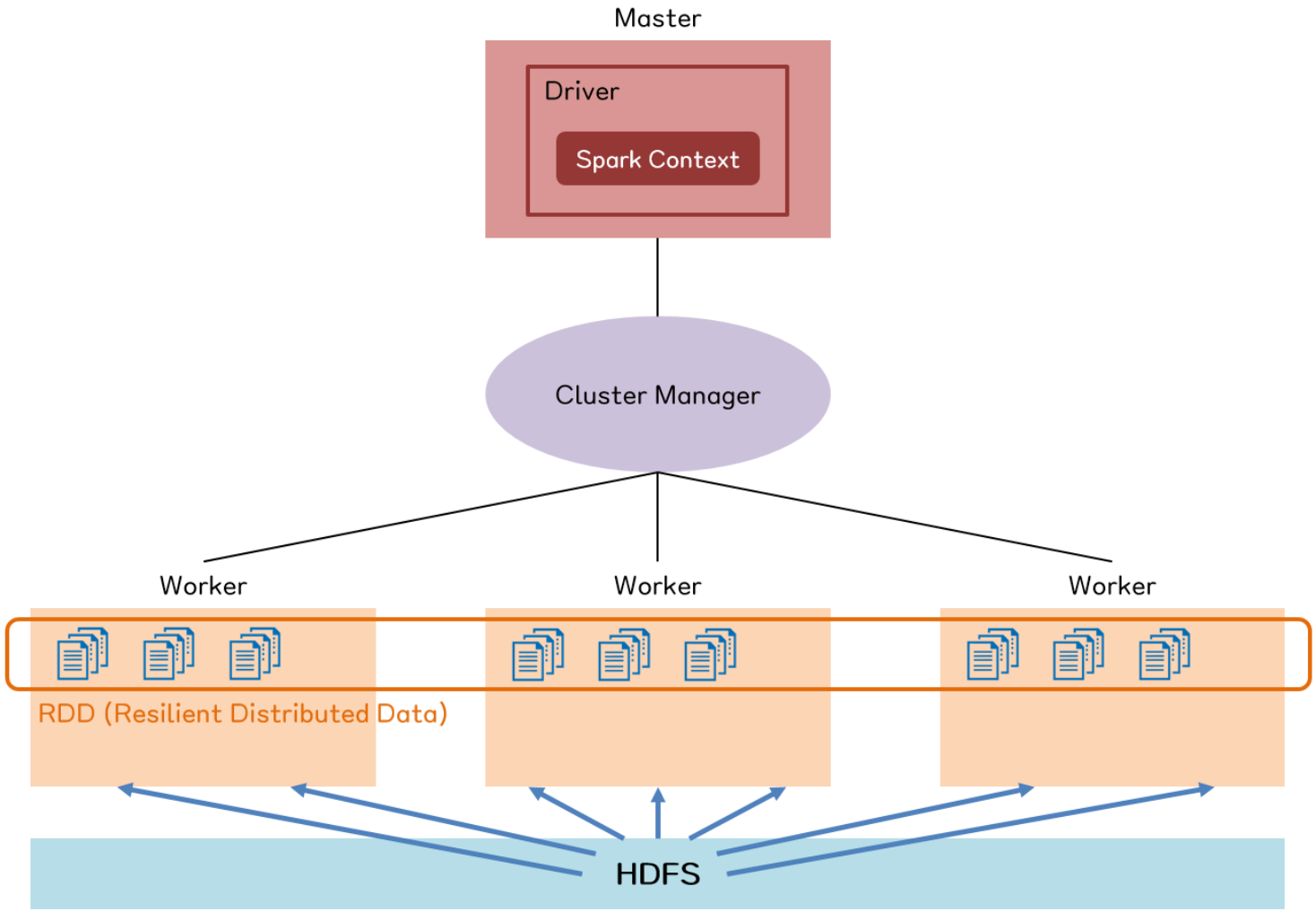
아파치 스파크의 기본 아키텍처



아파치 스파크 작업의 특성

Apache Spark

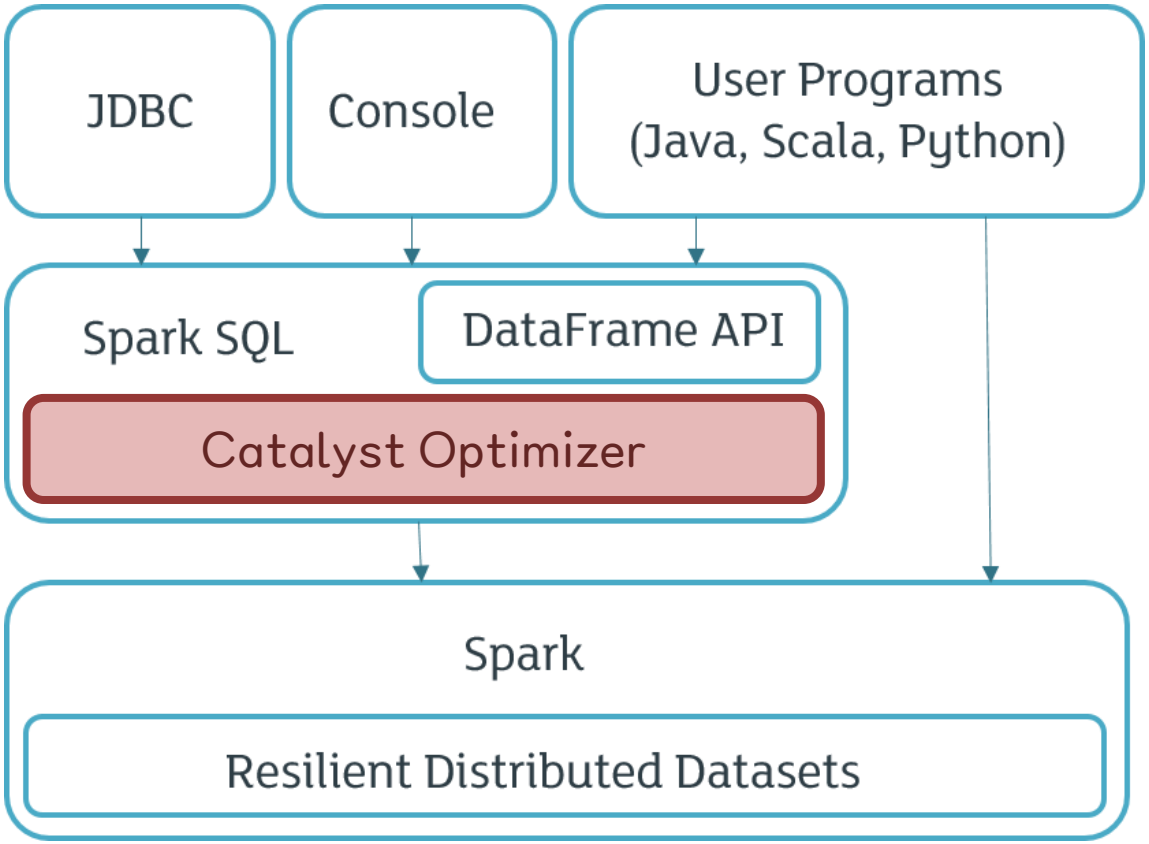
- Resilient Distributed Data (RDD) → 다수의 서버에 분산 방식으로 저장된 요소들의 집합. 병렬처리 및 장애 복구 가능



아파치 스파크 작업의 특성

Apache Spark

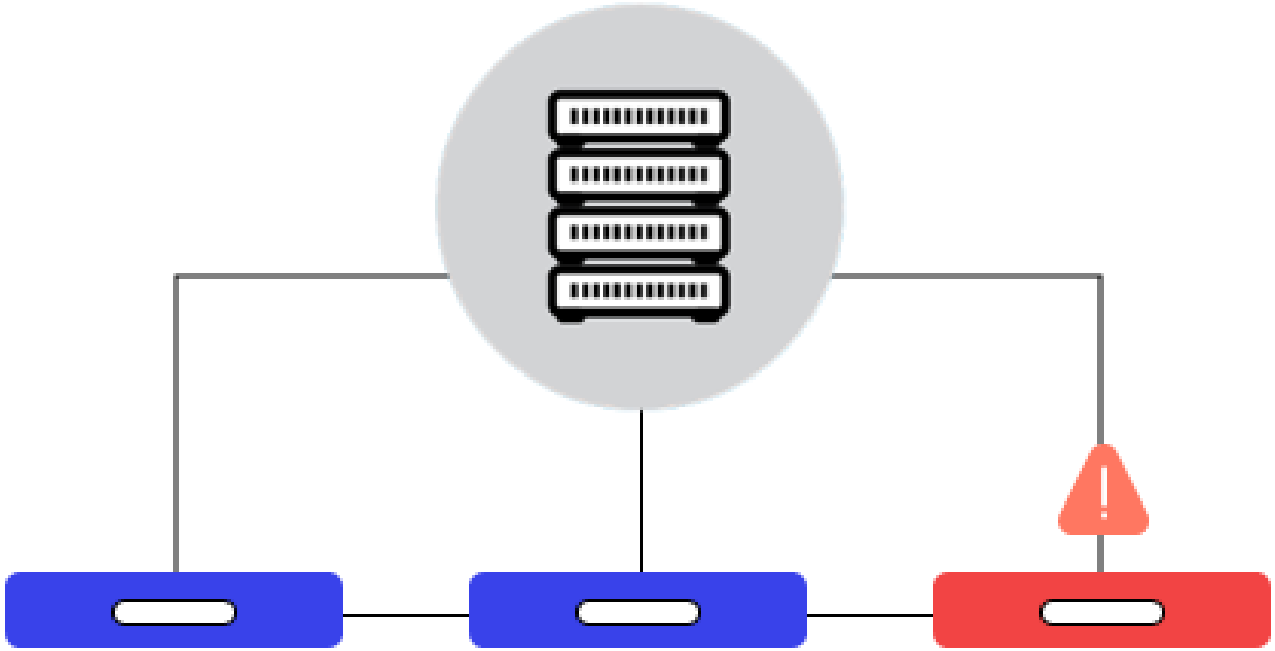
- Resilient Distributed Data (RDD) → 다수의 서버에 분산 방식으로 저장된 요소들의 집합. 병렬처리 및 장애 복구 가능
- Catalyst Optimizer → Optimizer를 통해 동작될 코드의 논리적/물리적 실행 계획을 최적화



아파치 스파크 작업의 특성

Apache Spark

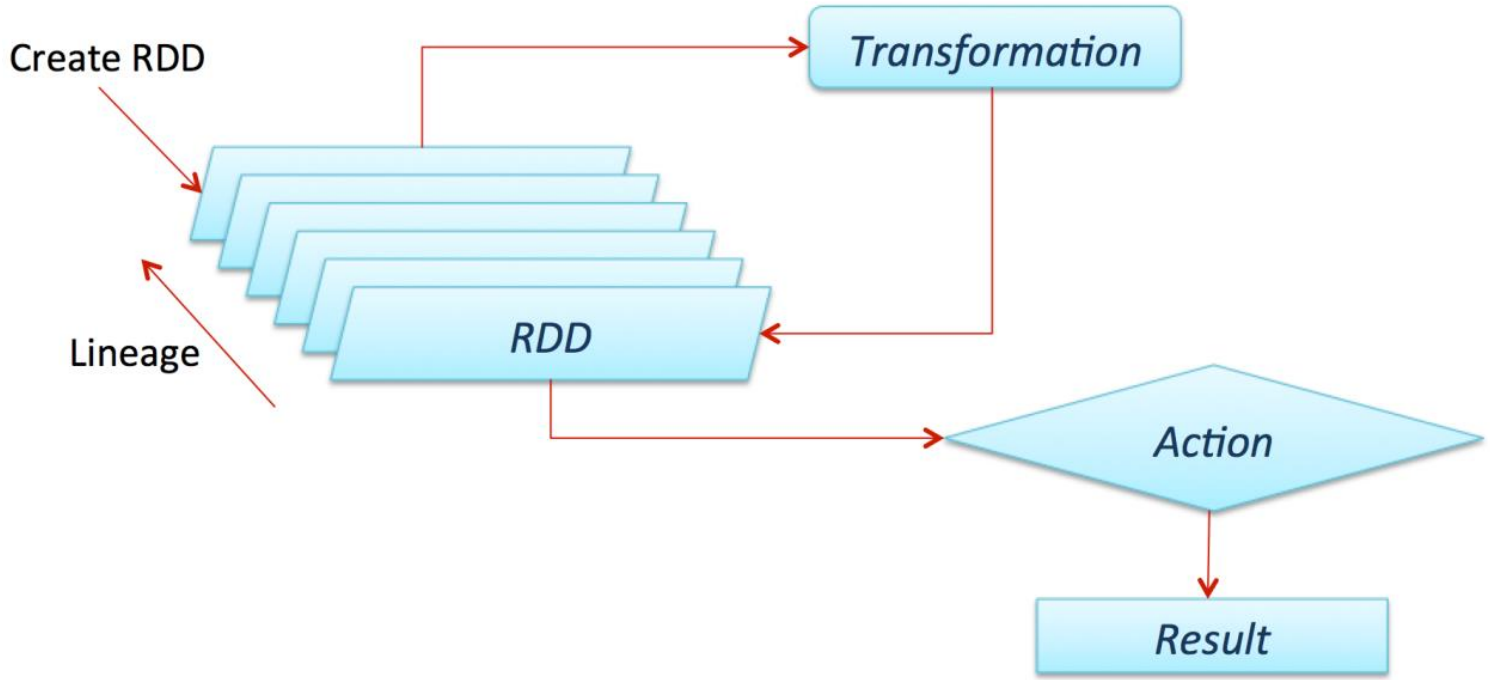
- Resilient Distributed Data (RDD) → 다수의 서버에 분산 방식으로 저장된 요소들의 집합. 병렬처리 및 장애 복구 가능
- Catalyst Optimizer → Optimizer를 통해 동작될 코드의 논리적/물리적 실행 계획을 최적화
- Fault Tolerance → 작업 실패시 자동으로 재시작하여 데이터 손실을 최소화하고 안정성을 보장



아파치 스파크 작업의 특성

Apache Spark

- Resilient Distributed Data (RDD) → 다수의 서버에 분산 방식으로 저장된 요소들의 집합. 병렬처리 및 장애 복구 가능
- Catalyst Optimizer → Optimizer를 통해 동작될 코드의 논리적/물리적 실행 계획을 최적화
- Fault Tolerance → 작업 실패시 자동으로 재시작하여 데이터 손실을 최소화하고 안정성을 보장
- Lazy Evaluation → 변환 작업을 선언한 후에도 실제 실행되는 시점까지 작업을 연기하는 방식



아파치 스파크 작업의 특성

Apache Spark

- Resilient Distributed Data (RDD) → 다수의 서버에 분산 방식으로 저장된 요소들의 집합. 병렬처리 및 장애 복구 가능
- Catalyst Optimizer → Optimizer를 통해 동작될 코드의 논리적/물리적 실행 계획을 최적화
- Fault Tolerance → 작업 실패시 자동으로 재시작하여 데이터 손실을 최소화하고 안정성을 보장
- Lazy Evaluation → 변환 작업을 선언한 후에도 실제 실행되는 시점까지 작업을 연기하는 방식

Transformation	Action
filter()	count()
groupby()	show()
orderBy()	collect()
join()	first()
union()	take()

df.groupby('col_a').agg(F.count('col_b'), F.sum('col_c')).orderBy('col_a') → 0.1 s

df.groupby('col_a').agg(F.count('col_b'), F.sum('col_c')).orderBy('col_a').show() → 37.7 s

아파치 스파크 작업의 특성

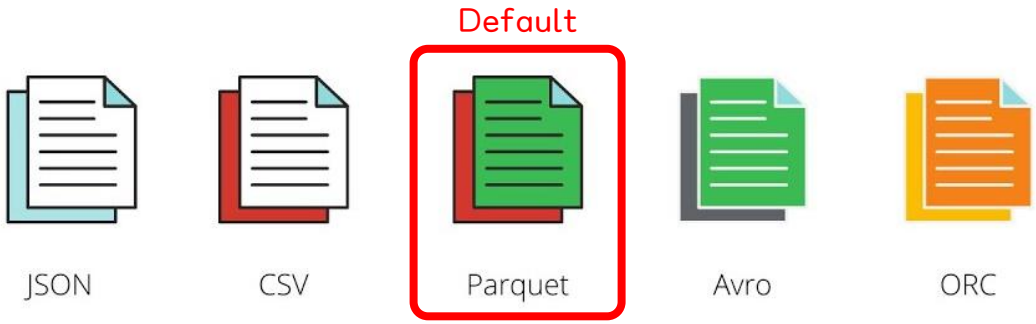
Apache Spark

- Resilient Distributed Data (RDD) → 다수의 서버에 분산 방식으로 저장된 요소들의 집합. 병렬처리 및 장애 복구 가능
- Catalyst Optimizer → Optimizer를 통해 동작될 코드의 논리적/물리적 실행 계획을 최적화
- Fault Tolerance → 작업 실패시 자동으로 재시작하여 데이터 손실을 최소화하고 안정성을 보장
- Lazy Evaluation → 변환 작업을 선언한 후에도 실제 실행되는 시점까지 작업을 연기하는 방식
- Columnar Format → .parquet 형식을 기본으로 하는 열 기반 데이터 처리

아파치 스파크 작업의 특성

Apache Spark

- Resilient Distributed Data (RDD) → 다수의 서버에 분산 방식으로 저장된 요소들의 집합. 병렬처리 및 장애 복구 가능
- Catalyst Optimizer → Optimizer를 통해 동작될 코드의 논리적/물리적 실행 계획을 최적화
- Fault Tolerance → 작업 실패시 자동으로 재시작하여 데이터 손실을 최소화하고 안정성을 보장
- Lazy Evaluation → 변환 작업을 선언한 후에도 실제 실행되는 시점까지 작업을 연기하는 방식
- Columnar Format → .parquet 형식을 기본으로 하는 열 기반 데이터 처리



아파치 스파크 작업의 특성

Apache Spark

- Resilient Distributed Data (RDD) → 다수의 서버에 분산 방식으로 저장된 요소들의 집합. 병렬처리 및 장애 복구 가능
- Catalyst Optimizer → Optimizer를 통해 동작될 코드의 논리적/물리적 실행 계획을 최적화
- Fault Tolerance → 작업 실패시 자동으로 재시작하여 데이터 손실을 최소화하고 안정성을 보장
- Lazy Evaluation → 변환 작업을 선언한 후에도 실제 실행되는 시점까지 작업을 연기하는 방식
- Columnar Format → .parquet 형식을 기본으로 하는 열 기반 데이터 처리