# 리눅스 기초 및 하둡

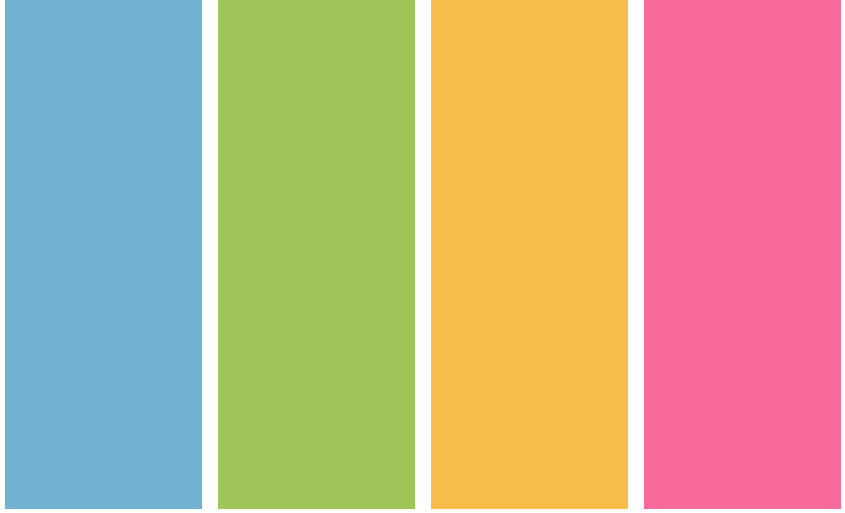
1.리눅스 기초
    1.VMWare를 이용한 가상화
    2.디렉터리와 파일 사용하기
    3.리눅스 vi 에디터 사용법
2.빅데이터 개념
    1.빅데이터 처리 시스템 개요
    2.빅데이터 처리 인프라 및 S/W
    3.하둡클러스터 동작 방식
2. 하둡
    1. Hadoop 종류와 구성요소
    2. 하둡 클러스터 구축을 위한 설정
        ; Full Distributed Mode 하둡 클러스터 설치
    1. Spark를 이용한 데이터 분석
    2. 하둡 클러스터 구성 관련 참고사항

4. 호튼웍스 샌드박스
    1. 호튼웍스 샌드박스 설치
    2. 하둡 기본 명령어
    3. Sqoop
    4. Pig
    5. Hive

https://bit.ly/33ISGQk

# 02 빅데이터 개념

1.빅데이터 처리 시스템 개요
2.빅데이터 처리 인프라 및 S/W
3.하둡클러스터 동작 방식



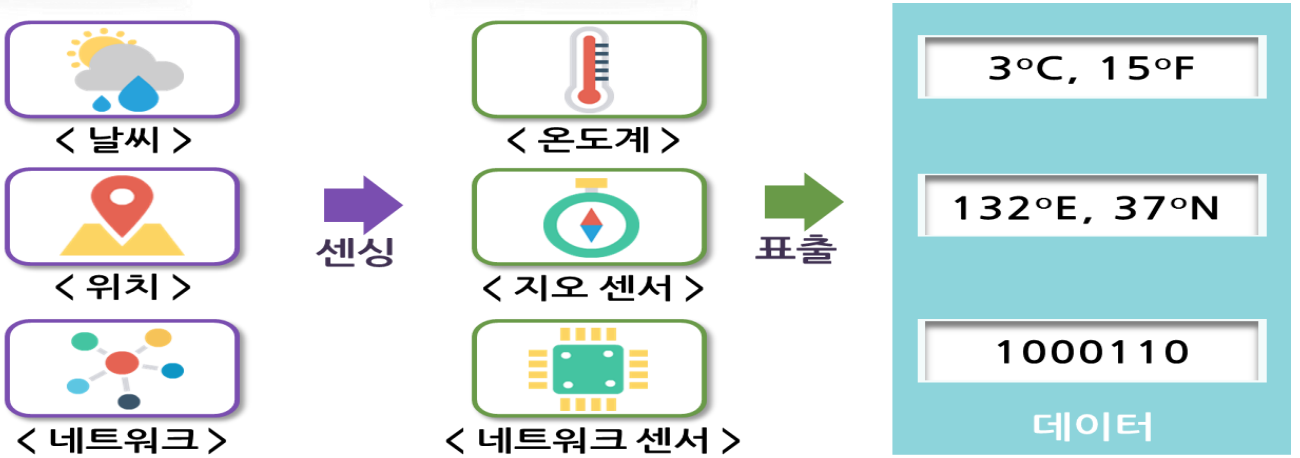
# 1. 빅데이터 처리 시스템 개요

# 빅데이터 처리 기술의 필요성

1. 빅데이터 처리 시스템 개요

## 1) 인간과 데이터

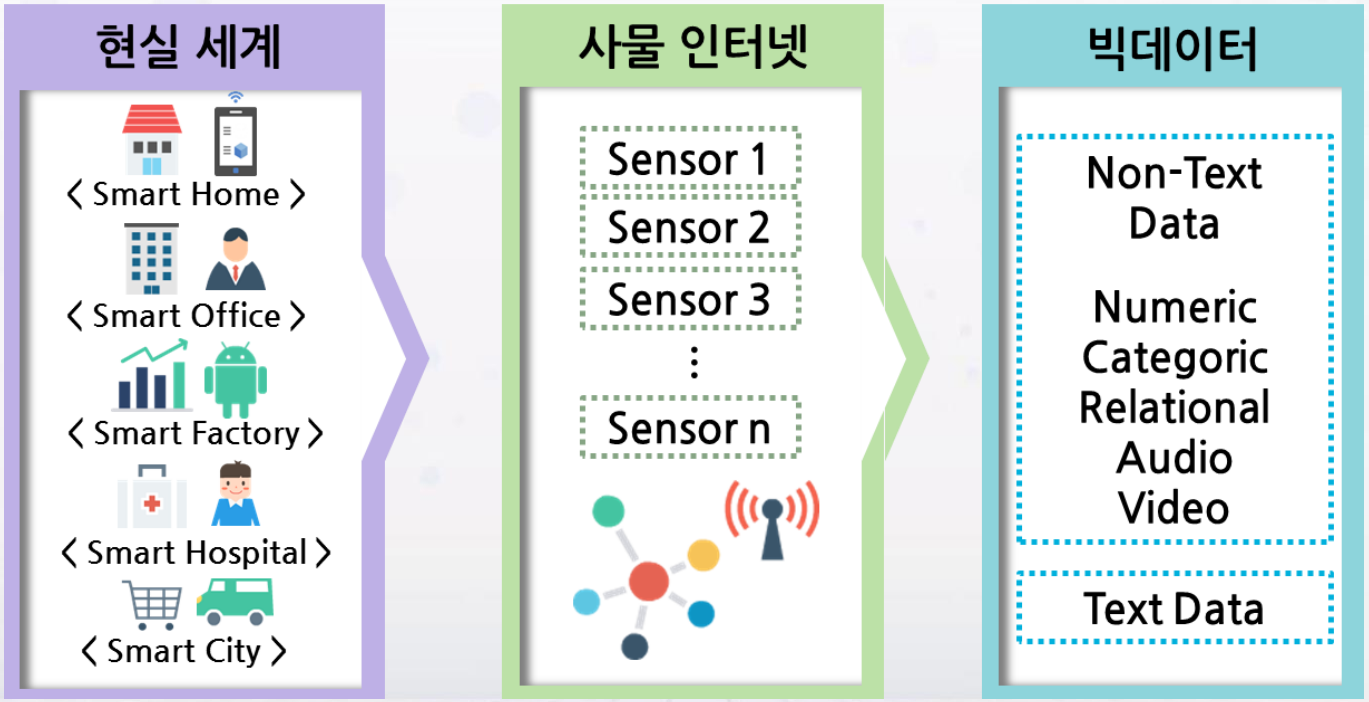

## 2) 센서와 데이터



# 빅데이터 처리 기술의 필요성



## 3) 사물인터넷과 빅데이터



# 빅데이터 처리 기술의 필요성



## 3) 빅데이터 특성 (3V)

### ① 데이터 볼륨 증가(Volume)


- 2020년에는 전 세계 디지털 데이터의 양이 약 40제타 바이트가 될 것이다 - IDC Digital University의 보고서 ; 모래알의 57배

VS

40ZB

# 빅데이터 처리 기술의 필요성



## 3) 빅데이터 특성 (3V)

### ① 데이터 볼륨 증가(Volume)

| $10^n$ | 접두어 | 기호 | 배수 | 십진수 |
|---|---|---|---|---|
| $10^{24}$ | 요타(yotta) | Y | 자 | 1 000 000 000 000 000 000 000 000 |
| $10^{21}$ | 제타(zetta) | Z | 십해 | 1 000 000 000 000 000 000 000 |
| $10^{18}$ | 엑사(exa) | E | 백경 | 1 000 000 000 000 000 000 |
| $10^{15}$ | 페타(peta) | P | 천조 | 1 000 000 000 000 000 |
| $10^{12}$ | 테라(tera) | T | 조 | 1 000 000 000 000 |
| $10^{9}$ | 기가(giga) | G | 십억 | 1 000 000 000 |
| $10^{6}$ | 메가(mega) | M | 백만 | 1 000 000 |
| $10^{3}$ | 킬로(kilo) | k | 천 | 1 000 |
| $10^{2}$ | 헥토(hecto) | h | 백 | 1 00 |
| $10^{1}$ | 데카(deca) | da | 십 | 1 0 |
| $10^{0}$ | | | 일 | 1 |

# 빅데이터 처리 기술의 필요성



## 3) 빅데이터 특성 (3V)

② 데이터 발생 속도 증가(Velocity)

모바일 및 SNS, 스캐너, 센서, RFID 태그 장치를 통해 과거와 비교할 수 없는 속도로 생성


- **빅데이터 생성 속도**
  - ✓하루 250경 바이트의 비정형 데이터
  - ✓매달 10억여 개의 트윗
  - ✓매달 350억여 개 페이스북 메시지
  - ✓1조대 이상 모바일 기기

# 빅데이터 처리 기술의 필요성



## 3) 빅데이터 특성 (3V)

③ 데이터 포맷 다양성 증가(Variety)

**기존 데이터웨어 하우스**

- ✓ 관계형 데이터 구조를 가진 데이터베이스 관리가 가능한 **정형화 데이터**
- ✓ 텍스트 위주 데이터

**최근 데이터**

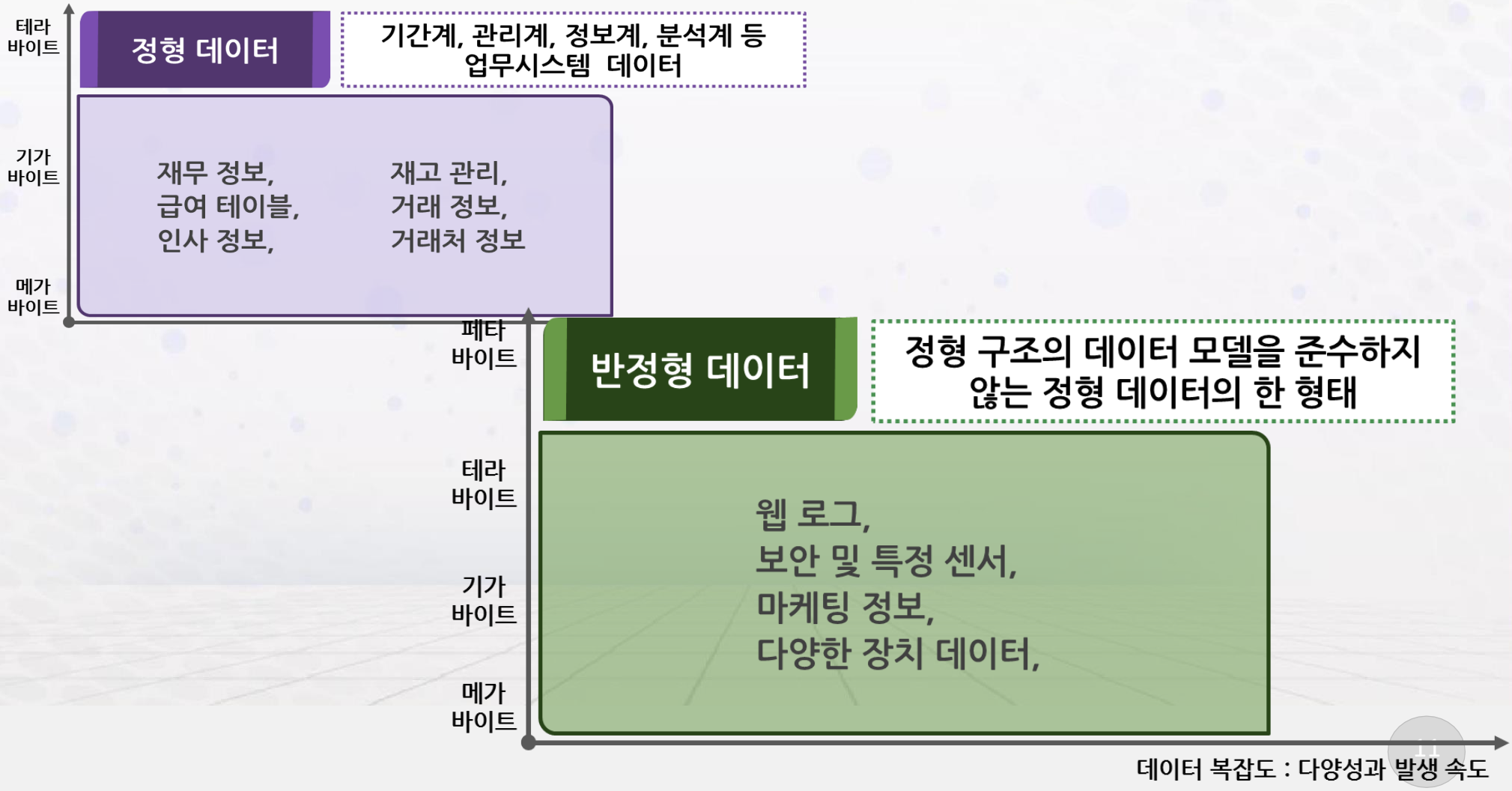
- ✓ **반정형 및 비정형 데이터**
- ✓ 그림, 동영상, 음성, 로그, 센서 데이터 스트림 등

다양한 데이터들이 생성됨에 따라 **기존 관계형 데이터 베이스로 처리하기에는 어려움**

# 빅데이터 처리 기술의 필요성



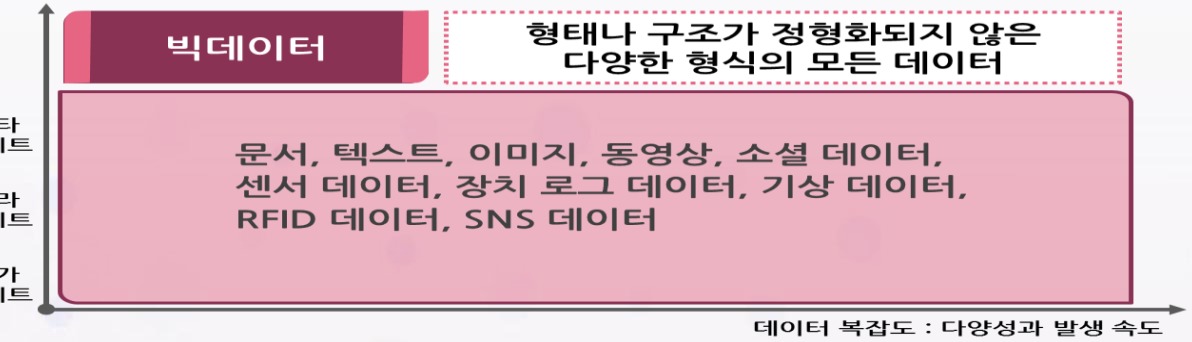
## 4) 빅데이터 유형

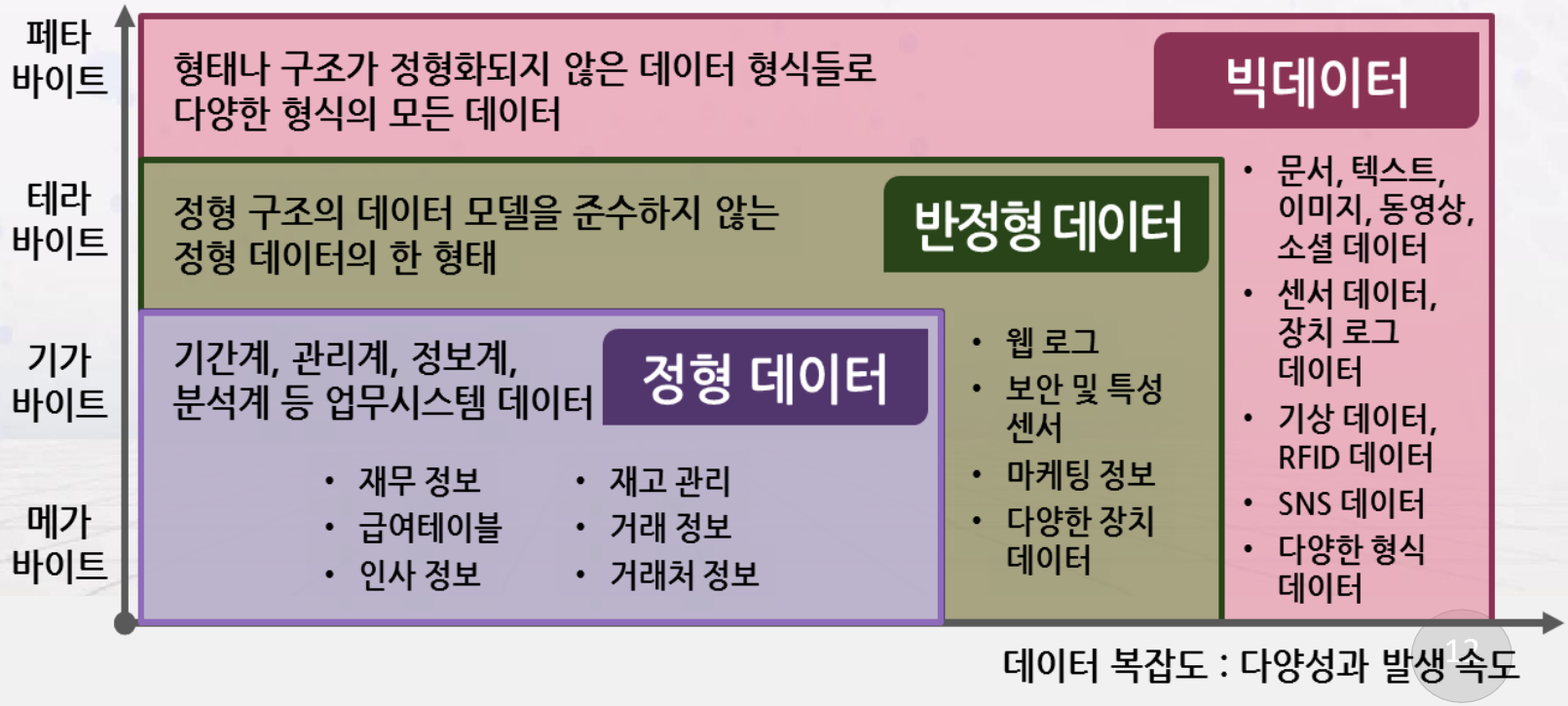


# 빅데이터 처리 기술의 필요성

1. 빅데이터 처리 시스템 개요

## 4) 빅데이터 유형



# 빅데이터 처리 기술의 필요성



## 4) 빅데이터 유형

| 유형 | 특징 | 종류 | 저장 시스템 예시 |
|---|---|---|---|
| 정형 데이터 | 정형화된 스키마를 가진 데이터 | RDB, File | RDB |
| 반정형 데이터 | 메타 구조를 가지는 데이터 | HTML, XML, JOSN, RSS, 웹로그, 센서 데이터, csv | RDB, NoSQL |
| 비정형 데이터 | 이미지나 동영상으로 존재하는 데이터 | 이진파일, 동영상, 이미지, 텍스트 | NoSQL, 분산파일시스템 |

# 빅데이터의 정의



- 일반적인 데이터베이스 SW가 저장, 관리, 분석할 수 있는 범위를 초과하는 규모의 데이터
  - Mckinsey, 2011

- 조직은 실시간 또는 거의 실시간으로 증가하는 더 크고 더 복잡한 데이터 집합의 분석을 통해 비즈니스 혜택을 파생한다.
  - McKinsey May 2011 article Big Data: The next frontier for innovation, competition, and productivity

- 기존 데이터베이스 관리도구의 데이터 수집·저장·관리·분석의 역량을 넘어서는 대량의 정형 또는 비정형 데이터 세트 및 이러한 데이터로부터 가치를 추출하고 결과를 분석하는 기술을 의미한다.
  - 위키백과

- 대용량 데이터를 활용/분석하여 가치 있는 정보를 추출하고, 생성된 지식을 바탕으로 능동적으로 대응하거나 변화를 예측하기 위한 정보화 기술
  - 국가정보화 전략위원회, 2011

**정말 큰 데이터 + 처리(분석)의 난이도가 높은 것**

# 빅데이터 처리 기술의 필요성



## 5) 빅데이터 시스템이란

대용량 데이터를 **분산 병렬 처리**하고 **관리**하는 시스템

1. 사용자에게 데이터의 유형에 따라서 실시간 (Real-time) 처리나 배치(Batch) 처리를 할 수 있도록 하는 프레임워크

2. 대규모 양의 데이터의 수집, 관리, 유통, 분석을 처리하는 일련의 분산 병렬 처리 프레임워크

# 빅데이터 처리 시스템의 정의



**대규모 양의 데이터의 수집, 관리, 유통, 분석을 처리하는 일련의 분산 병렬 처리 프레임워크**

**프레임워크 (Framework)**

**소프트웨어의 구체적인 부분에 해당하는 설계와 구현을 재사용이 가능한 협업 형태로 제공하는 소프트웨어 환경**

# 빅데이터 처리 시스템의 목표



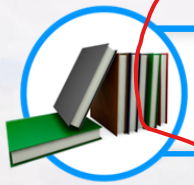
## 1 결함 허용 시스템(Fault Tolerance)

모든 시스템은 애플리케이션 문제 또는 하드웨어 리소스 문제로 인해 장애가 발생할 수 있음

장애가 발생해도 시스템이 대체 시스템이나 고장 대응체계를 통해서 시스템 운영을 계속하는 것

빅데이터 처리를 위해서는 장애가 발생해도 버티고 수행하는 능력인 '결함허용 시스템'을 갖춤

### 하둡(Hadoop)에서의 결함 허용 전략



클러스터 내의 노드가 수행 중에 죽거나, 실행이 실패하는 경우

다른 노드에 작업(Job)을 재할당

작업 재수행(Restart)을 자동으로 수행

Load Balancing

Active/Standby

Data Mirroring

Data Replication

# 빅데이터 처리 시스템의 목표



## 2 저비용 시스템(Cost Effective System)

작업에 적합한 시스템을 적절하게 선택하여 비용절감

### 하둡(Hadoop)에서의 비용 절감 절략

다양한 유틸리티로 시스템 처리 비용 감소

저렴한 서버를 사용하여 분산 처리

전체 장애 가능성 감소

수평 확장(Scale Out)이 가능한 구조로 설계되어 저비용 시스템 구축 가능

# 빅데이터 처리 시스템의 목표



## 2 저비용 시스템(Cost Effective System)

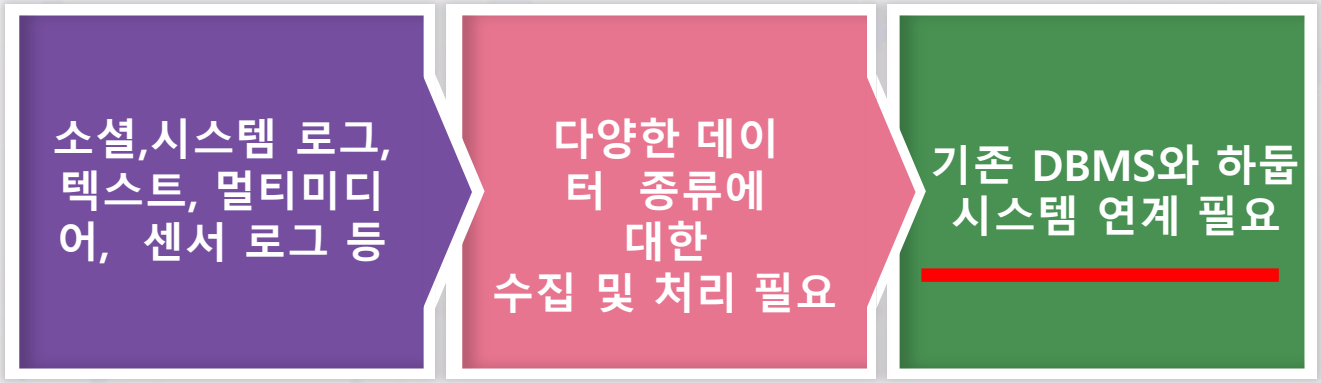

수평 확장(Scale Out)이 가능한 구조로 설계되어
저비용 시스템 구축 가능

# 빅데이터 처리 시스템의 목표



## 3 기존 시스템과 연계성

스쿱

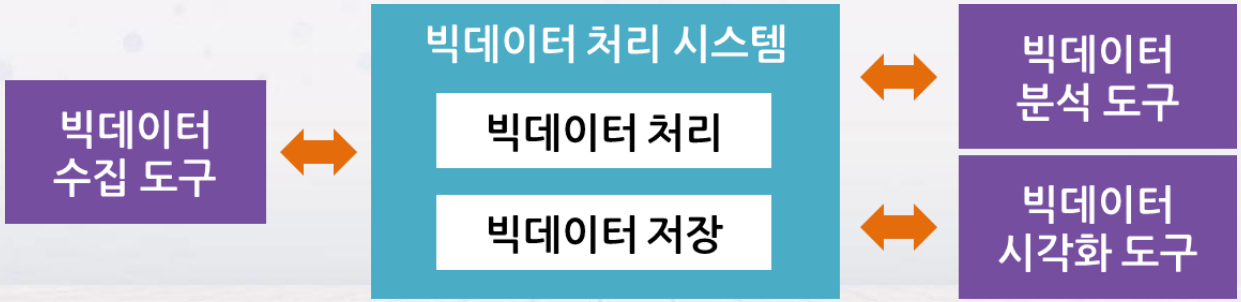


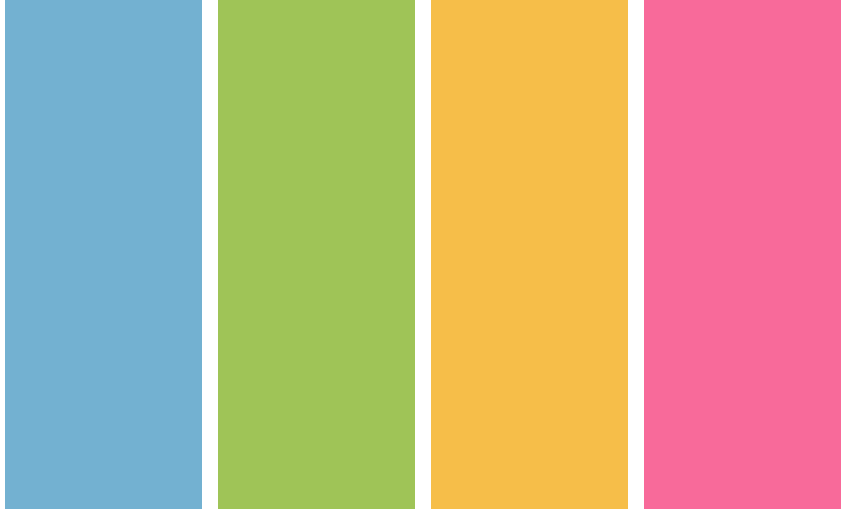
# 빅데이터 처리 시스템의 목표



## 4 다양한 도구 지원

1. 빅데이터 수집, 저장, 처리, 분석, 시각화 도구 등 다양한 도구 지원 필요
2. 다양한 도구들 간 시스템 호환성 지원 필요





# 2. 빅데이터 처리 인프라 및 S/W

# 빅데이터 프로세스 과정



## 1 빅데이터 프로세스 과정

| 처리 | 분석 | 시각화 |
| --- | --- | --- |
| 배치 | 통계 분석 | 시간 시각화 |
| 실시간 | 데이터 마이닝 | 분포 시각화 |
| 분산병렬 | 텍스터 마이닝 | 관계 시각화 |
| | 머신러닝 기반 | 비교 시각화 |
| | | 공간 시각화 |

# 빅데이터 처리 인프라



## 2 빅데이터 처리 인프라

**클라우드 컴퓨팅을 사용하는 방식**

1. 인터넷 상의 서버를 통해 IT 관련 서비스를 사용할 수 있는 컴퓨팅 환경
   - 데이터 저장, 네트워크, S/W, 콘텐츠 사용 등
2. 초기 투자비용이 적게 발생
3. 시스템에 대한 구축 오버헤드 절감
4. 향후 시스템이 확장되어야 할 때 확장성 확보 가능

**개별적인 시스템 구축 방식**

대표적인 하둡 배포판

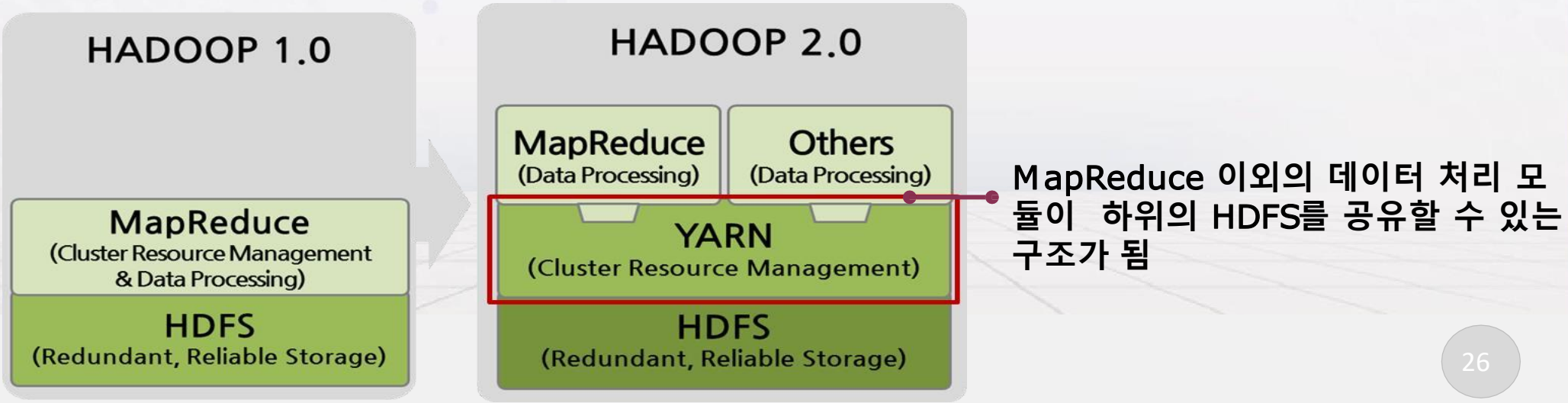
- 아파치 하둡 배포판
- 맵알 하둡 배포판
- 클라우데라 하둡 배포판
- 호튼웍스 하둡 배포판

# 빅데이터 처리 S/W



## 1 하둡(Hadoop)

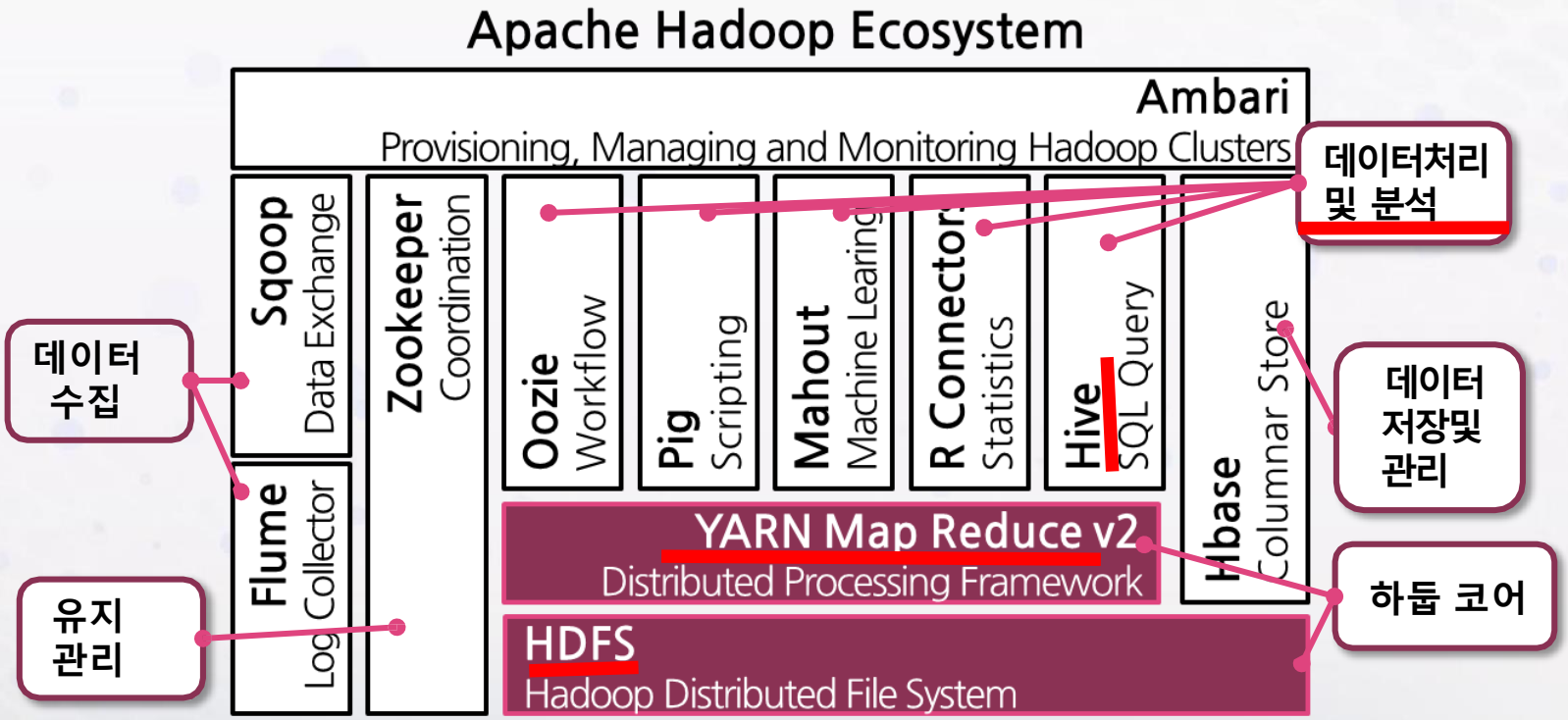
- ✓ 오픈 소스 S/W
- ✓ 하둡 분산 파일 시스템(HDFS) + 맵리듀스(MapReduce)/YARN
- ✓ 빅데이터 처리 프레임워크
- ✓ 다양한 하둡 에코 시스템으로 구성
- ✓ 결함 허용
- ✓ 데이터 블록의 복사본을 중복 저장하고 유지



# 빅데이터 처리 S/W



## ② 아파치 하둡 에코 시스템



# 빅데이터 처리 S/W



## ③ 스파크

**인-메모리 방식의 분산 처리 시스템**

UC 버클리의 AMP 랩에서 개발

메모리 사용으로 **반복 작업이나 스트리밍 데이터를** 효율적으로 처리

배치, 스트리밍 처리, SQL 기반 쿼리 수행 기능, 머신러닝 라이브러리

스칼라 쉘(Scala Shell)을 제공하여 **사용자와 대화형**으로 데이터 관리 가능

✓ 스칼라(Scala) 언어로 구현되어 있지만 **다양한 언어를 지원하는 SDK**를 가지고 있음

파이썬 | 자바 | R | ...

✓ 다양한 데이터 스토리지와 **연동 가능**

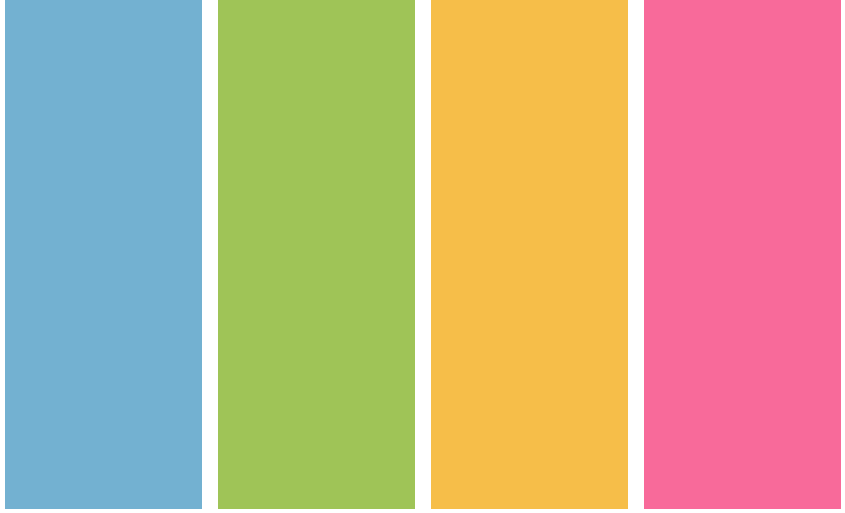
HDFS | 아마존 S3 | 카산드라 | Hbase | ...

1. 스파크 오픈 소스 엔진 위에 파이썬을 사용하기 위한 절차
    1) 자바설치
    2) 하둡 설치 및 설정
    3) 하둡 구동
    4) Spark 설치 및 설정
    5) Pyspark를 이용한 jupyter notebook 이용
2. 관계형 데이터 베이스의 제약 조건에 대해 5가지 이상 기술하시오

    Unique : 동일한 컬럼값을 허용하지 않는다 / Primary Key : 특정 레코드를 선택하는 컬럼

    Foreign Key : 다른 테이블을연결하는 컬럼 / Not Null : 컬럼값에 널을 허용하지 않음

    Check : 레코드의 컬럼값 조건을 만족 / Default : 커럼값이 없을 경우 기본값 지정
3. 딥러닝 프로그래밍 작성시 프로그래밍 절차
    1) 데이터셋 준비하기
    2) 훈련셋과 검증셋을 분류하거나 원핫인코딩 작업, 정규화작업 등 데이터 전처리하기
    3) 모델구성하기
    4) 모델 학습과정 설정하기
    5) 모델 학습시키기
    6) 모델 평가하기
    7) 모델사용하기

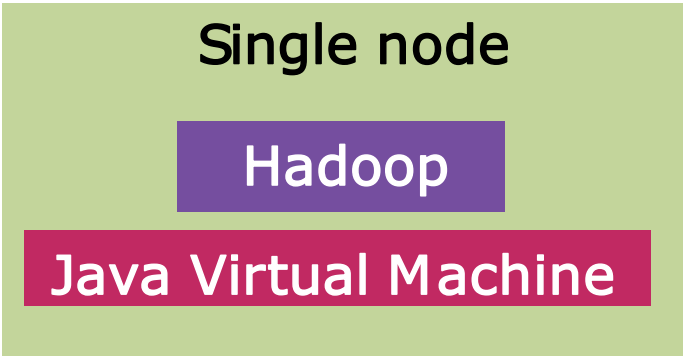


# 3. 하둡 클러스터 동작 방식

# 독립모드 (Standalone Mode)



데몬 프로세스 없이 모든 프로그램이 하나의
JVM(Java Virtual Machine)에서 동작하는 모드

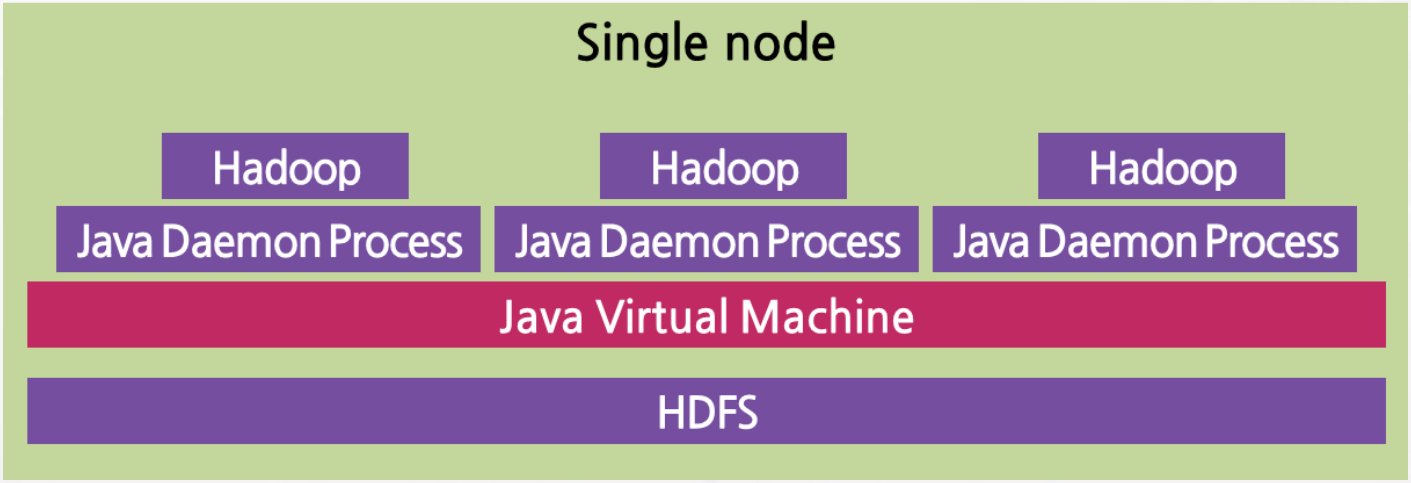



1. 맵리듀스 프로그램을 동작시키고 개발 테스트하는 동안에 사용하는 모드
2. 분산 운영 모드가 아니므로 실제 빅데이터 처리 환경으로 부적합
3. HDFS를 사용하지 않고 로컬 파일 시스템을 사용

# 의사분산모드 (Pseudo-distributed Mode)

3. 하둡 클러스터 동작방식

1대의 컴퓨터에 하둡 데몬 프로세스가
여러 개 분리되어 동작하는 모드

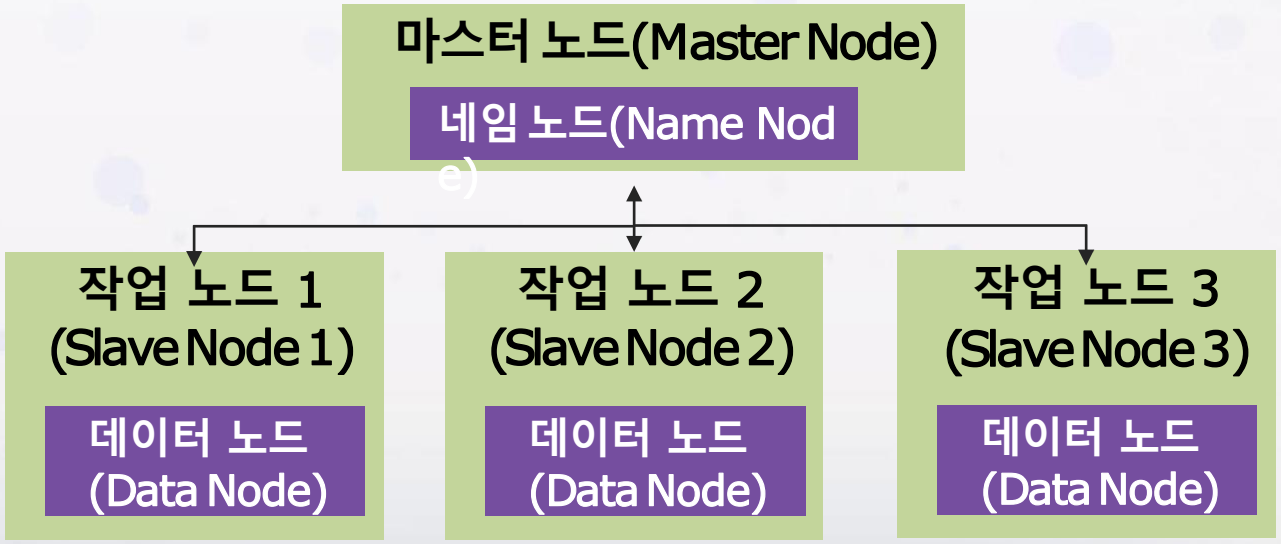



1. 작은 규모의 클러스터를 테스트, 디버깅, 프토로 타이핑 하는 경우에 주로 사용
2. 1대의 컴퓨터를 사용해서 가상 분산 운영 모드로 사용
3. HDFS를 사용

# 완전분산모드 (Fully distributed Mode)

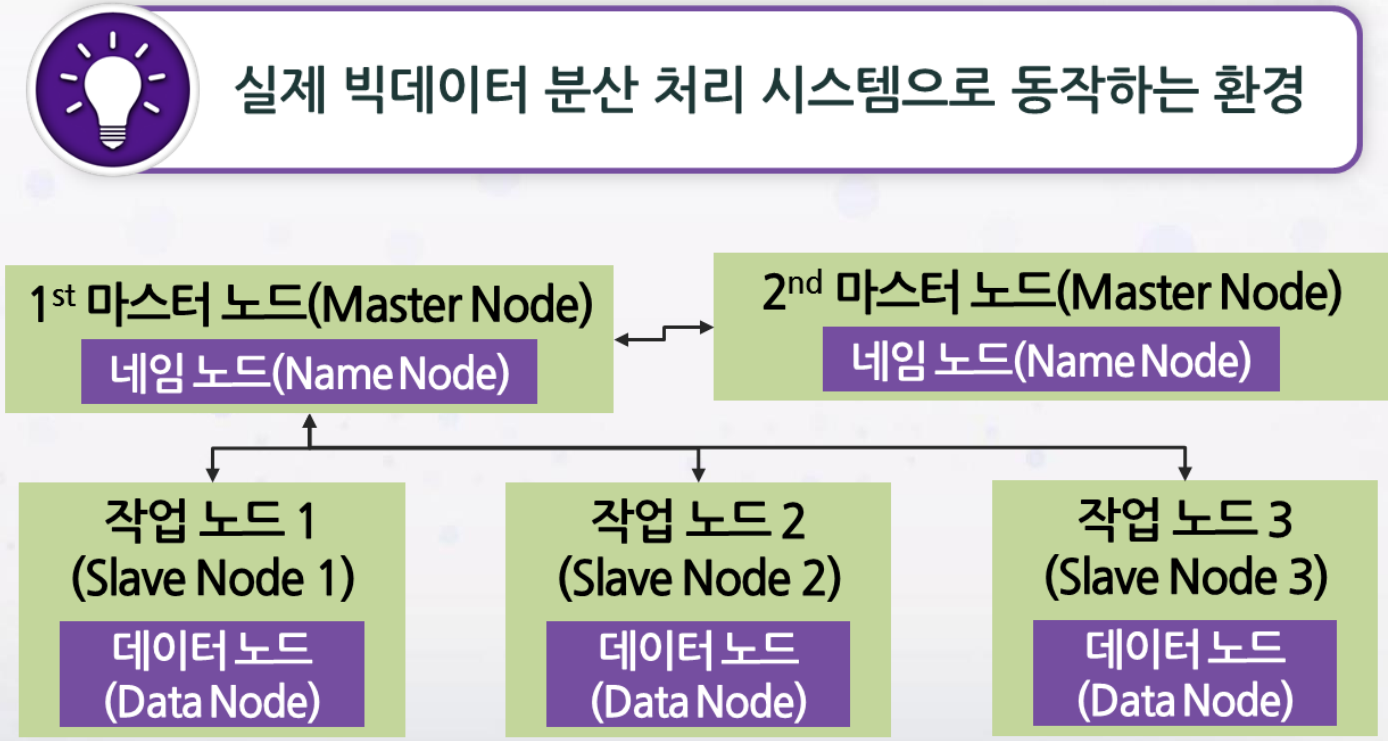
3. 하둡 클러스터 동작방식

하둡 데몬 프로세스가 클러스터로 구성된
여러 개의 컴퓨터에 나누어 동작



# 완전분산모드 (Fully distributed Mode)

3. 하둡 클러스터 동작방식



# 완전분산모드 (Fully distributed Mode)

3. 하둡 클러스터 동작방식

데이터들은 실제 데이터 노드에 분산 저장되며 이들에 대한 메타정보는 네임 노드에서 관리하는 운영 모드