



# AWS EMR Overview

BigData Immersion Day

Park, CheonGu / Solutions Architect

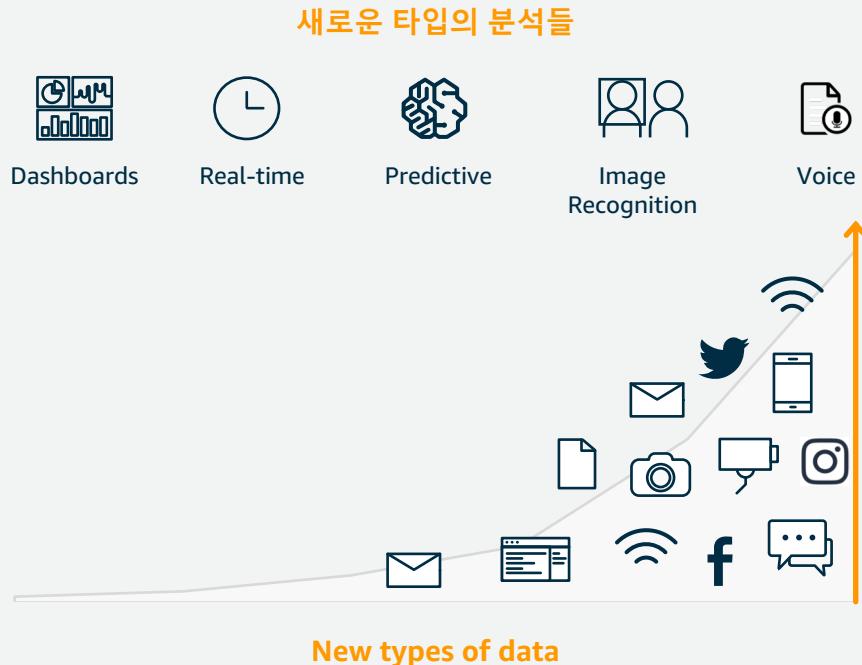
November, 2018

# Table of contents

- Data Lakes with EMR
- Amazon EMR 소개 및 Architecture
- Amazon S3와 EMRFS
- Amazon EMR Design Patterns
- Summary
- HOL : SQL-on-Hadoop

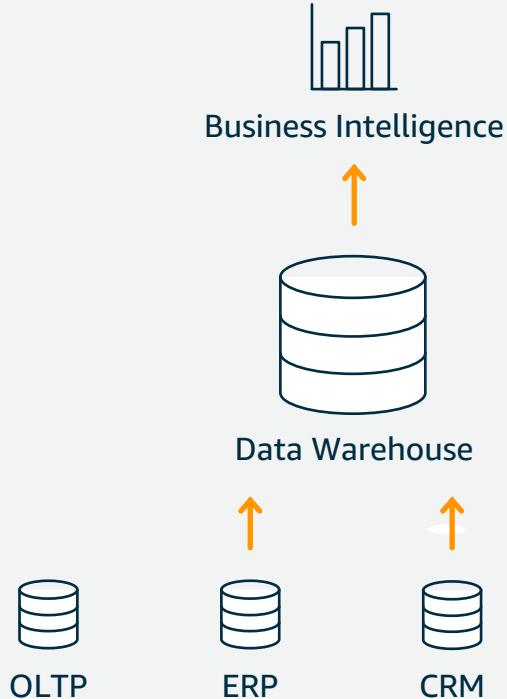
# Data Lakes with EMR

# 데이터가 차별화 요소가 되기 위해서 고객이 필요한 것들



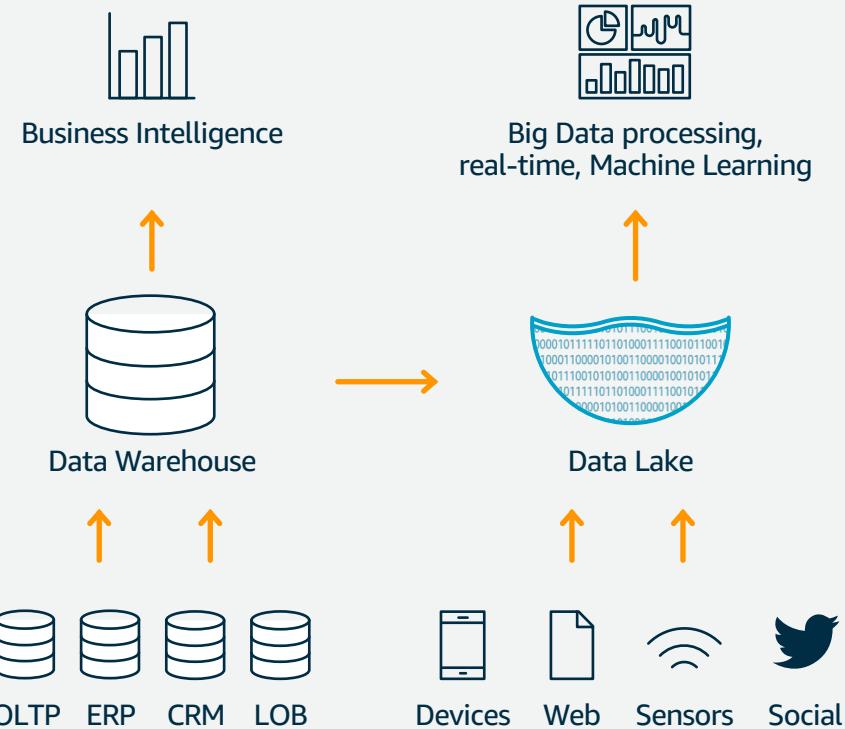
- PB에서 EB에 달하는 비관계형 데이터를 캡처 및 저장
- 배치 리포팅 만으로 할 수 없는 실시간, 예측, 음성 및 이미지 분석 등의 새로운 타입의 분석들
- 안전하고 통제된 환경하에서 여러 곳에서 동시에 쉽게 엑세스

# 전통적인 분석은 다음과 같이 진행됨



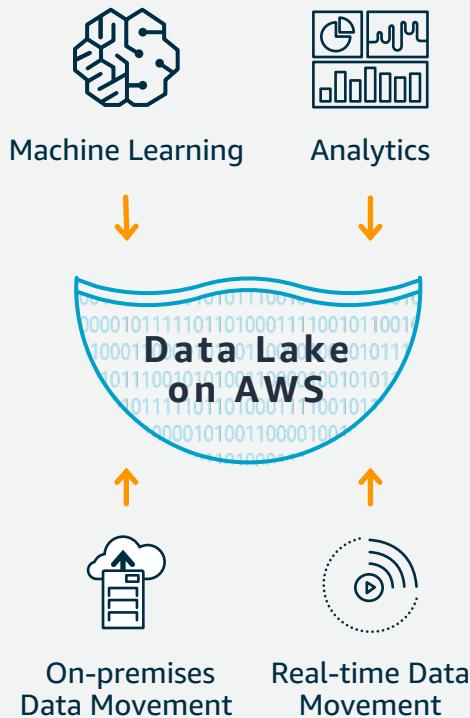
- 관계형 데이터
- 수 TB 부터 수 PB 까지의 크기
- 데이터 로딩 전에 정의되어야 하는 스키마
- DW상에서 가능해지는 운영 보고
- 매우 큰 초기 CAPEX / OPEX 비용

# 전통적인 접근법을 확장하는 Data Lakes



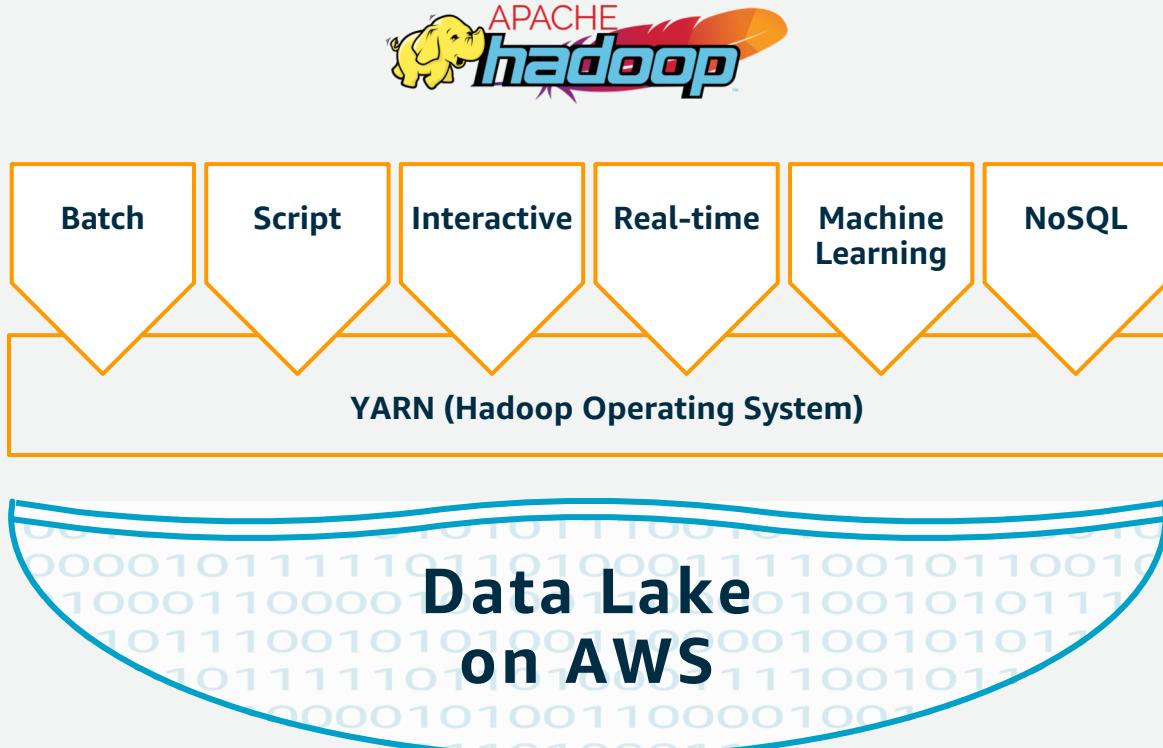
- '관계형' 과 '비관계형' 데이터
- 수 TB에서 수 EB의 크기
- 다양한 분석 엔진
- 저렴한 스토리지 및 분석 비용

# AWS의 Data Lakes 와 분석 도구



- 개방적이며 포괄적임
- 보안
- 확장성 및 내구성
- 비용 효율적

# Data Lake 위에서 Apache Hadoop 구축



- 분산처리
- 다양한 분석 도구들
  - Batch/Script (Hive/Pig)
  - Interactive (Spark, Presto)
  - Real-time (Spark)
  - Machine Learning (Spark)
  - NoSQL (HBase)
- 다양한 Use-Case들
  - Log and clickstream analysis
  - Machine Learning
  - Real-time analytics
  - Large-scale analytics
  - Genomics
  - ETL

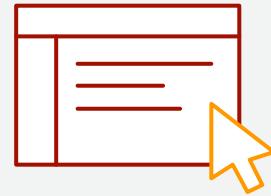
# EMR 소개

# Amazon EMR 소개

Managed Hadoop and Spark in the cloud



엔터프라이즈급



손쉬운 사용

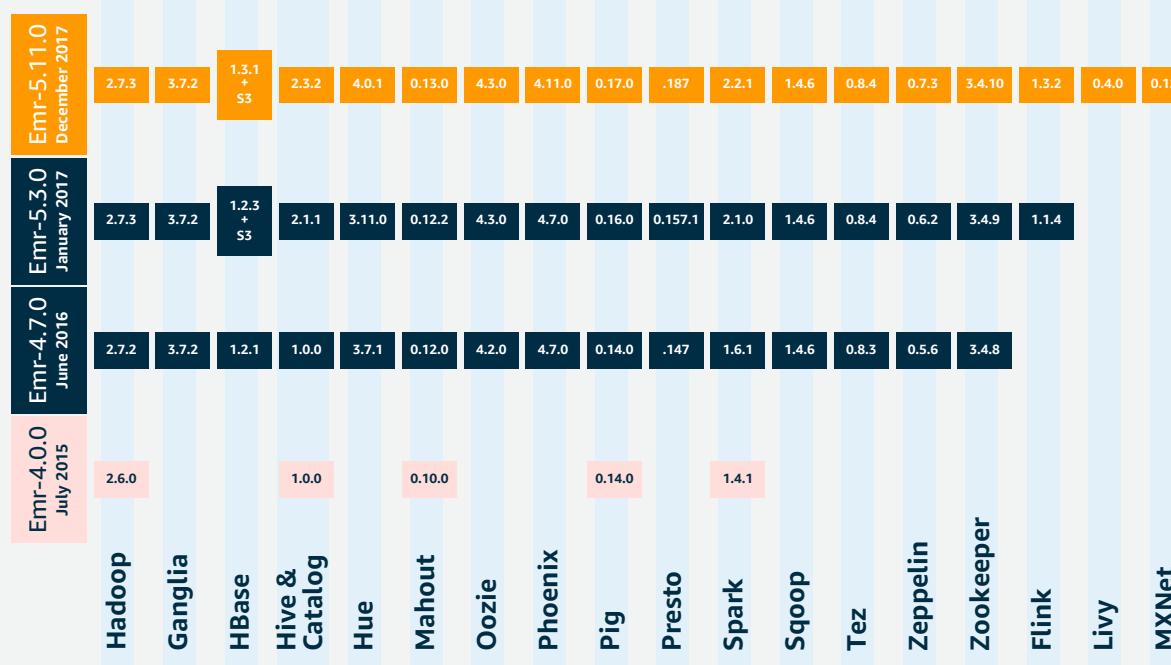


저렴한 비용

# 엔터프라이즈에서 사용하는 Hadoop & Spark

## Hadoop and Spark 생태계 SW의 최신버전을 배포함

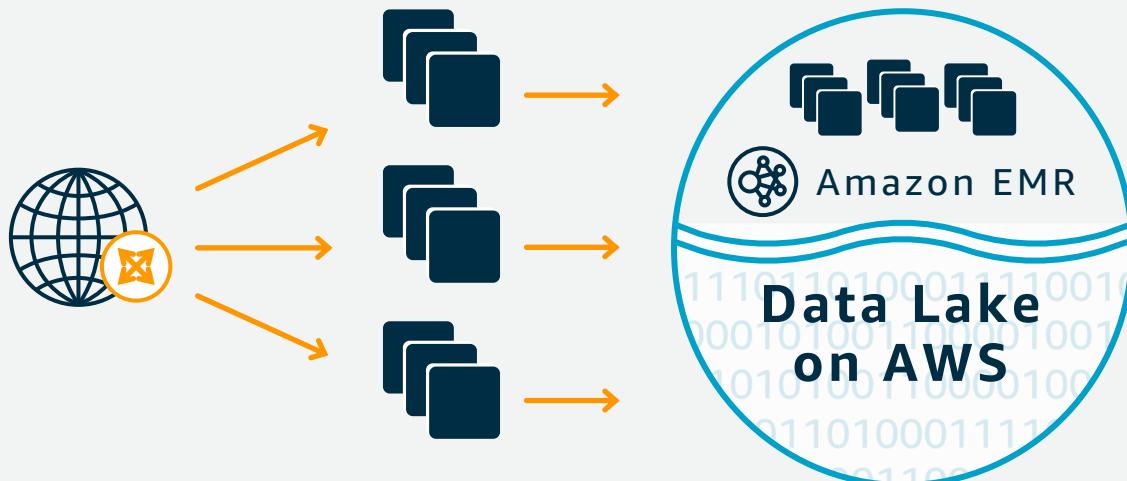
EMR Releases (<https://docs.aws.amazon.com/emr/latest/ReleaseGuide/images/emr-releases-5x.png>)



- 19 오픈소스 프로젝트:  
Apache Hadoop,  
Spark, HBase, Presto  
등등
- 각 프로젝트의 마지막  
버전을 출시후  
30일내에 EMR에  
포함

# 엔터프라이즈에서 사용하는 Hadoop & Spark

어느 크기로도 늘릴수 있음



- 컴퓨팅(EMR)과 스토리지(S3)가 독립적임.
- 어떠한 용량의 데이터로 저장 처리할 수 있음 : 수PB ~ 수EB
- 한 개에서 수 천개의 노드까지 프로비저닝 할 수 있음
- Auto-scaling

# 엔터프라이즈에서 사용하는 Hadoop & Spark

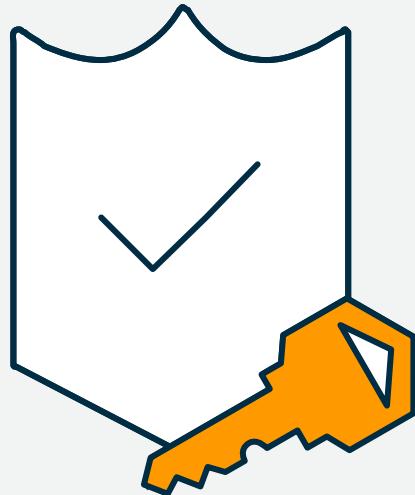
## 고가용성과 내구성

- S3 는 99.99999999% 내구성을 제공할 수 있도록 디자인 되어 있음
- EMR은 여러분들의 클러스터를 모니터링해서 실패한 노드나 성능에 문제가 있는 노드를 교체하고 시스템을 재시작함
- Amazon CloudWatch를 이용해서 모니터링
- Job history 나 browse logs 를 보기 위한 내장 Console
- 데이터 지속성을 위한 클러스터 HDFS를 가지고 있습니다.



# 엔터프라이즈에서 사용하는 Hadoop & Spark

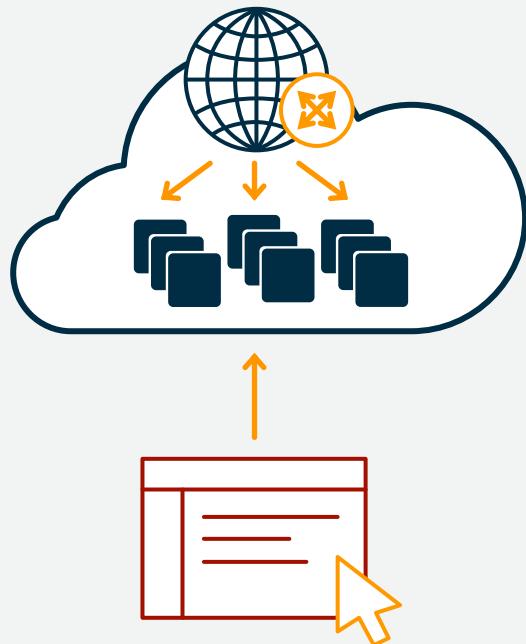
## 높은 보안성



- 저장 및 전송 중 데이터 암호화 지원
- Amazon Macie 로 ML 을 이용한 보안지원
- Amazon VPC 를 이용한 네트워크 격리
- IAM policies 를 이용한 Access 와 Permission 관리
- AWS CloudTrail 로 Log와 audit 수행
- Kerberos support 를 통한 Microsoft AD 통합

# 손쉬운 사용

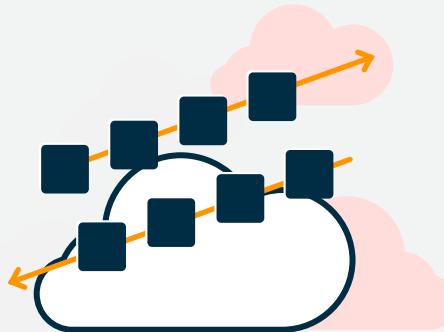
수 분 내에 클러스터를 기동 가능함



- Hadoop & Spark 클러스터를 수 분 내에 기동 가능
- Hadoop 을 유지보수하거나 설치할 필요가 없음
- Cluster 튜닝이나 설정이 완료되어 있음
- 발표된 지 30일 이내의 최신 Hadoop 버전

# 손쉬운 사용

## 유연하고 자동화된 확장(Scaling)



- Policy에 기반한 자동화된 클러스터의 규모
- 프로세스가 완료되었을 때 클러스터를 종료할 수 있음
- 한시적으로 사용해야 하는 클러스터나 장기간 실행 클러스터 모두에 최적화 되어 있음
- 수동 개입이 필요 없음

# 저렴한 비용

Reserved Instances 나 Spot Instance 를 사용해서 75% ~ 95% 비용 절감



- Reserved Instance 로 예약된 기간을  
커밋하고 최대 75% 절약
- Spot 인스턴스를 사용해서 최대  
90% 비용 절감
  - On-demand 가격의 일부만 지불함
  - 입찰가격 지정하고, 그 가격이  
시장가보다 크면 리소스를 얻을 수 있음
  - Spot과 on-demand의 인스턴스 타입  
리스트에서 프로비저닝 가능
  - 용량/가격 기준으로 최적의 AZ에서 런치  
할 수 있음
  - Spot Block 지원

# 저렴한 비용 낮은 TCO

## On-premises

### Support Costs

#### 서버 비용

Hardware—Server, Rack, Chassis, PDUs, Tor Switches (+Maintenance)  
Software—OS, Virtualization Licenses (+Maintenance)

#### 네트워크 비용

Network Hardware—LAN Switches, Load Balancer Bandwidth costs  
Software—Network Monitoring

#### IT 운영비

Server admin, virtualization admin, storage admin, network admin, support team

#### 기타

Project planning, advisors, legal, contractors, managed services, training, cost of capital

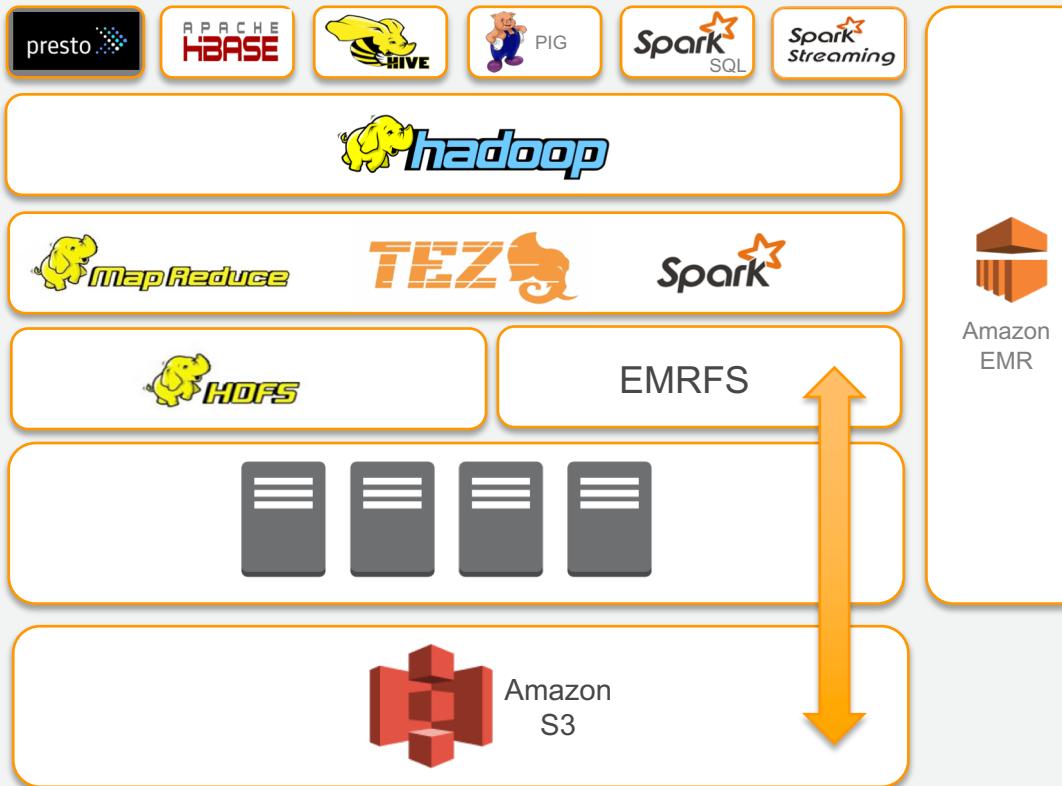
## EMR

### Subscription Fee Support Costs

- Hadoop cluster의 지원 및 관리 시간 단축
- 하드웨어 구입, 설치 등의 선행 비용 없음
- 데이터센터 공간, 전력, 냉각 등의 운영비용 절감
- 비즈니스 가치 : 자연비용, Risk 프리미엄, 경쟁력, 거버넌스 등

# EMR 아키텍쳐

# Amazon EMR 아키텍처



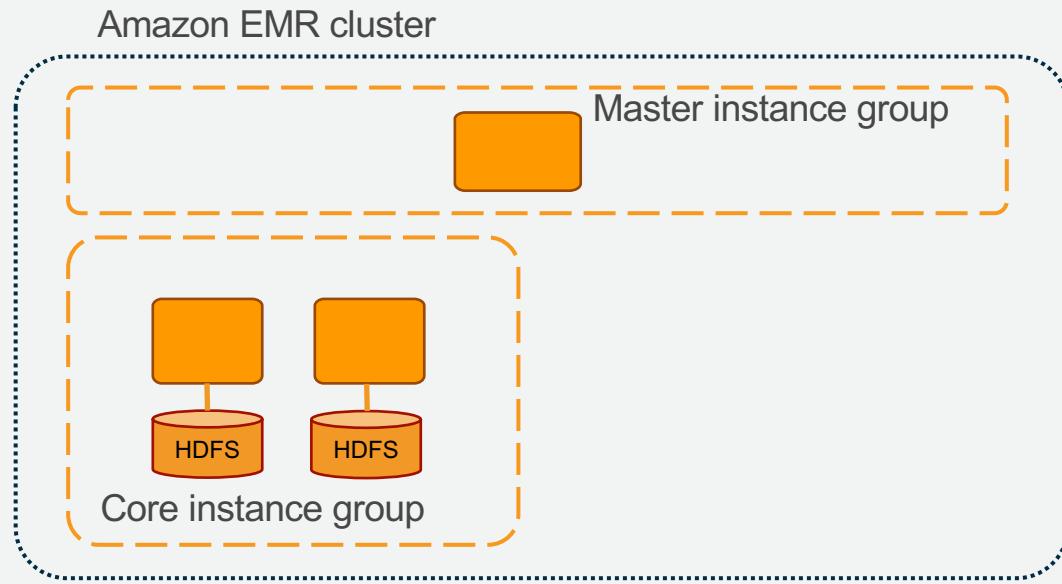
# 코어 노드와 태스크 노드



Run TaskTrackers  
(Compute)

Run DataNode (HDFS)

HDFS 또는 S3를 백엔드  
스토리지로 사용 가능



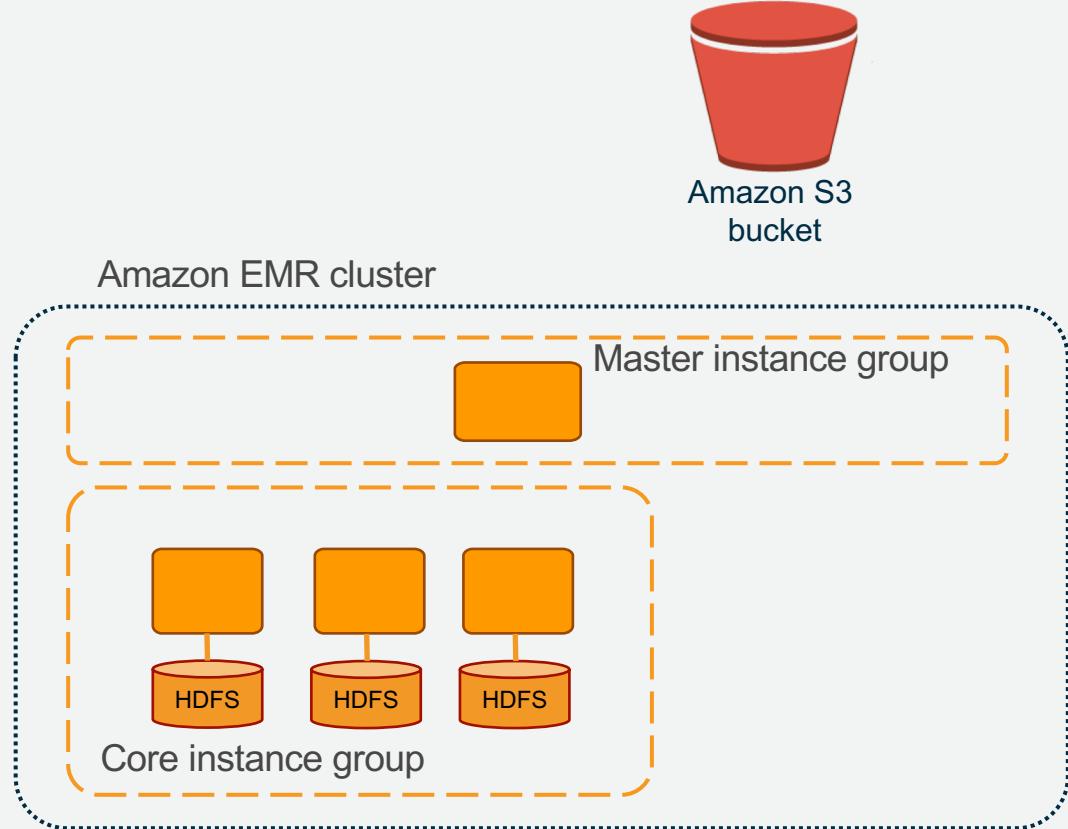
# 코어 노드

코어 노드 추가 가능

HDFS 공간 추가 필요  
(복제 세팅에 따라 용량  
결정됨)

더 많은 CPU/memory

운영 데이터가 HDFS에  
저장된 경우 스케일다운 시  
유의



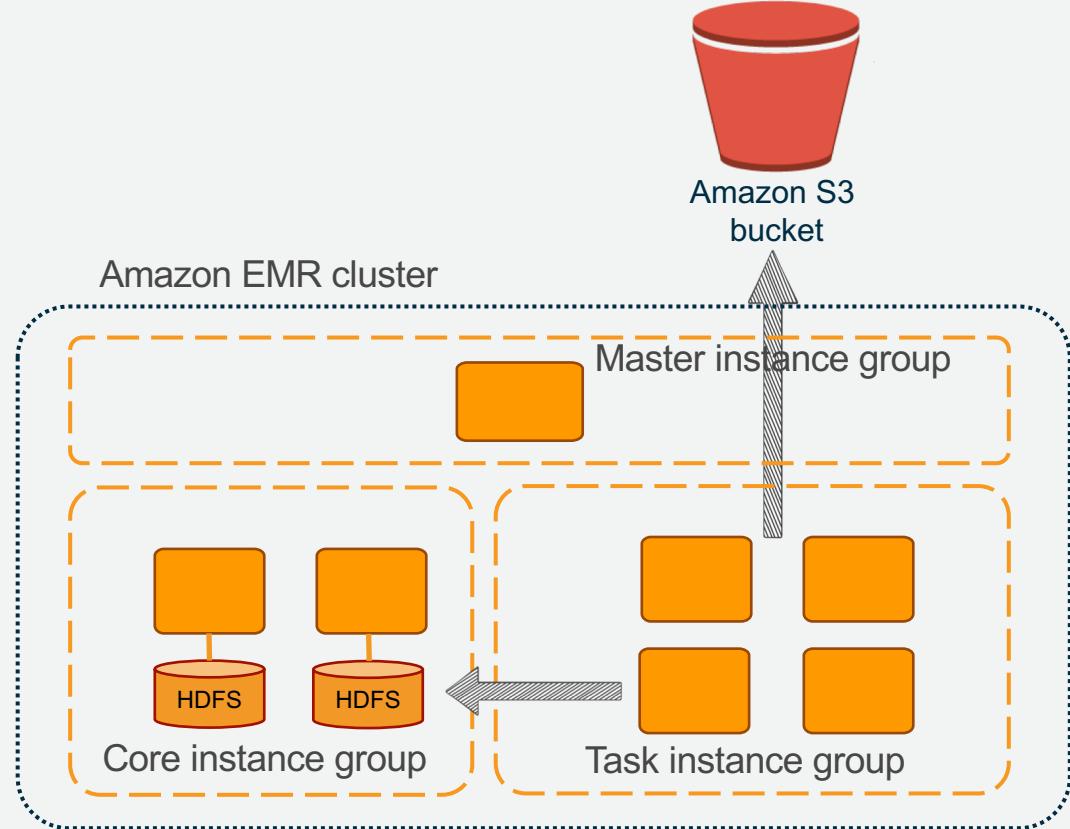
# 태스크 노드

Run TaskTrackers

데이터 저장을 위한 로컬 HDFS  
없음

코어 노드의 HDFS나 S3에서  
데이터를 읽어 옴

Spot 인스턴스 적극적 활용

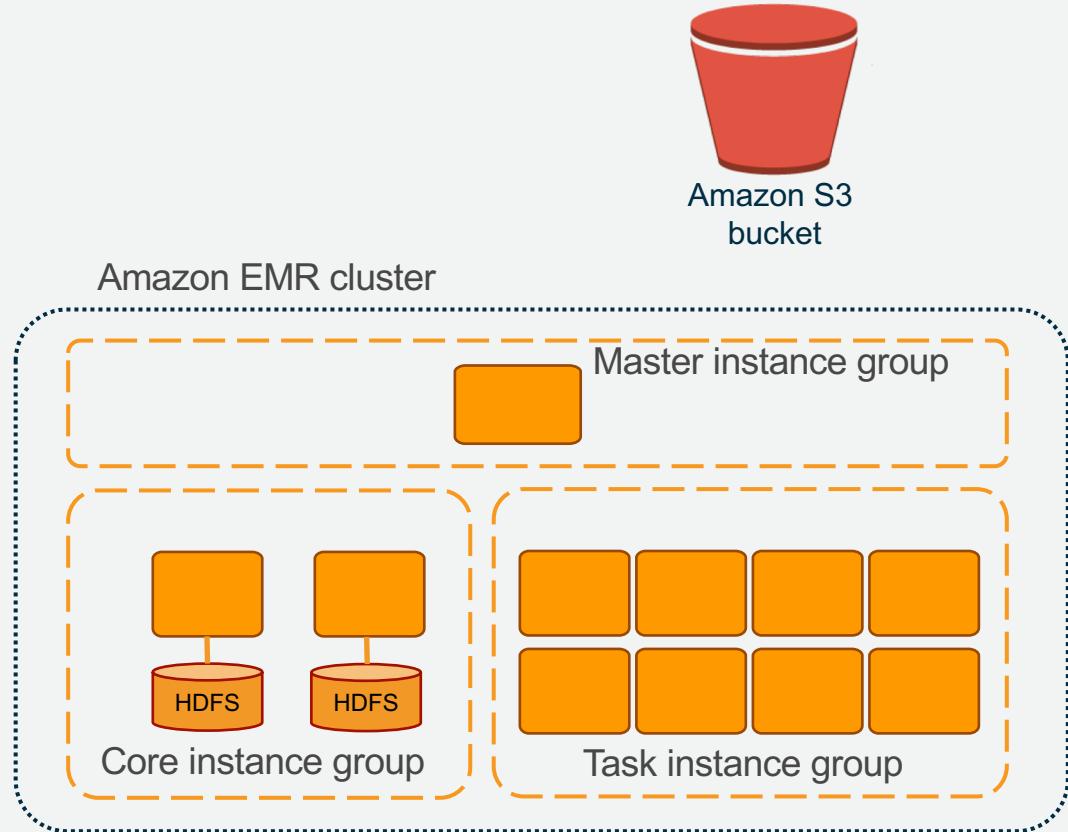


# 태스크 노드

더 많은 CPU power

더 많은 memory

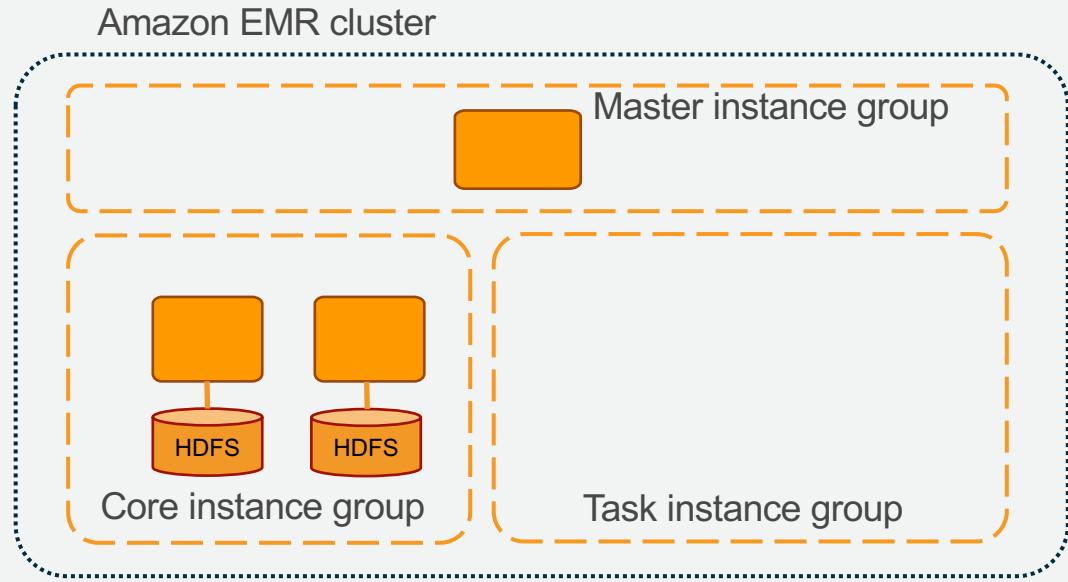
수동 또는 Auto-Scaling으로  
노드 추가/제거



# 태스크 노드

작업이 완료되면 작업 노드를 제거 가능

또는 Auto Scaling을 통해 자동으로 제거



# EMR Instance Fleets 활용

Node type	Fleet instance types	Target capacity	Advanced Spot options
Master Master - 1	<b>m3.xlarge</b> 8 vCPU, 15 GiB memory, 80 SSD GB storage EBS Storage: none Maximum bid price: <input type="text"/> % On-Demand <input type="text"/> 100 <a href="#">Add / remove instance types to fleet</a>	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot <i>The master fleet consists of one EC2 instance</i>	 Spot instances 중단시 온디맨드로 자동 전환
Core Core - 2	<b>m3.xlarge</b> 8 vCPU, 15 GiB memory, 80 SSD GB storage EBS Storage: none Maximum bid price: <input type="text"/> % On-Demand <input type="text"/> 30  <b>c4.large</b> 2 vCPU, 3.8 GiB memory, 80 SSD GB storage EBS Storage: 32 GiB Maximum bid price: <input type="text"/> <a href="#">Add / remove instance types to fleet</a>	<input type="text"/> 10 On-demand vCPUs <input type="text"/> 20 Spot vCPUs <b>30 Total vCPUs</b>	 Provisioning timeout Switch to On-Demand instances <input type="text"/> after <input type="text"/> min. of Spot unavailability
Task Task - 3	<b>m3.xlarge</b> 8 vCPU, 15 GiB memory, 80 SSD GB storage EBS Storage: none Maximum bid price: <input type="text"/> % On-Demand <input type="text"/> 100 <a href="#">Add / remove instance types to fleet</a>	<input type="text"/> 0 On-demand vCPUs <input type="text"/> 0 Spot vCPUs <b>0 Total vCPUs</b>	 Defined duration Not set   Provisioning timeout Terminate cluster <input type="text"/> after <input type="text"/> min. of Spot unavailability

# 클러스터 사이즈 조정 사용 예

- 업무 시간 동안 수동으로 확장되는, 작은 규모의 Always-on 클러스터
- 정해진 처리 시간 동안에 다양한 크기의 워크로드를 위해 작업 노드 추가
- 스팟 인스턴스로 구성된 작업 노드를 추가하여 온디맨드 코어 노드 줄이기
- 워크로드가 요구하는 클러스터 사이즈와 맞추기 위해 Auto Scaling 활용

# 향상된 관리 기능의 활용

Auto Scaling rules

Maximum instances: 20 ⓘ  
Minimum instances: 4 ⓘ

Scale out

Default-scale-out-1: Add 1 instance if YARNMemoryAvailablePercentage is less than 15  
1 five-minute period with a cooldown of 300 seconds

Default-scale-out-2: Add 1 instance if ContainerPendingRatio is greater than 0.75 for 1  
five-minute period with a cooldown of 300 seconds

+ Add rule

Scale in

Default-scale-in: Terminate 1 instance if YARNMemoryAvailablePercentage is greater than 75 for 1 five-minute period with a cooldown of 300 seconds

+ Add rule

인스턴스 용량 설정

YARN 리소스 사용기반 설정

자동확장/축소선택

# Amazon S3 와 EMRFS

# AWS 데이터 소스로 읽고 쓰기



# EMRFS는 S3를 더 쓰기 쉽게

일관성 보기

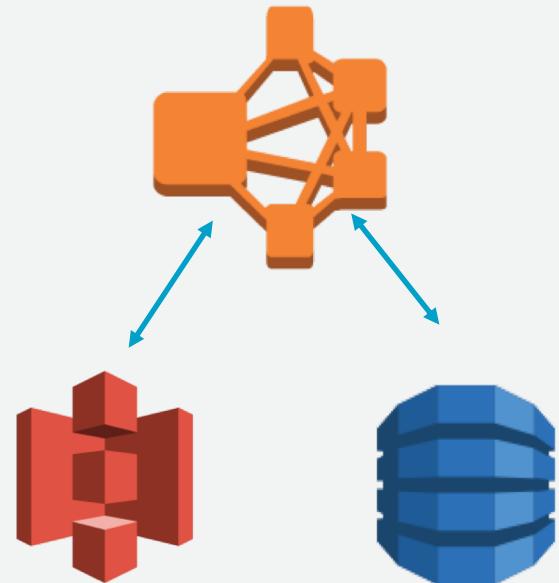
삭제 일관성

쓰기 후 읽기 일관성

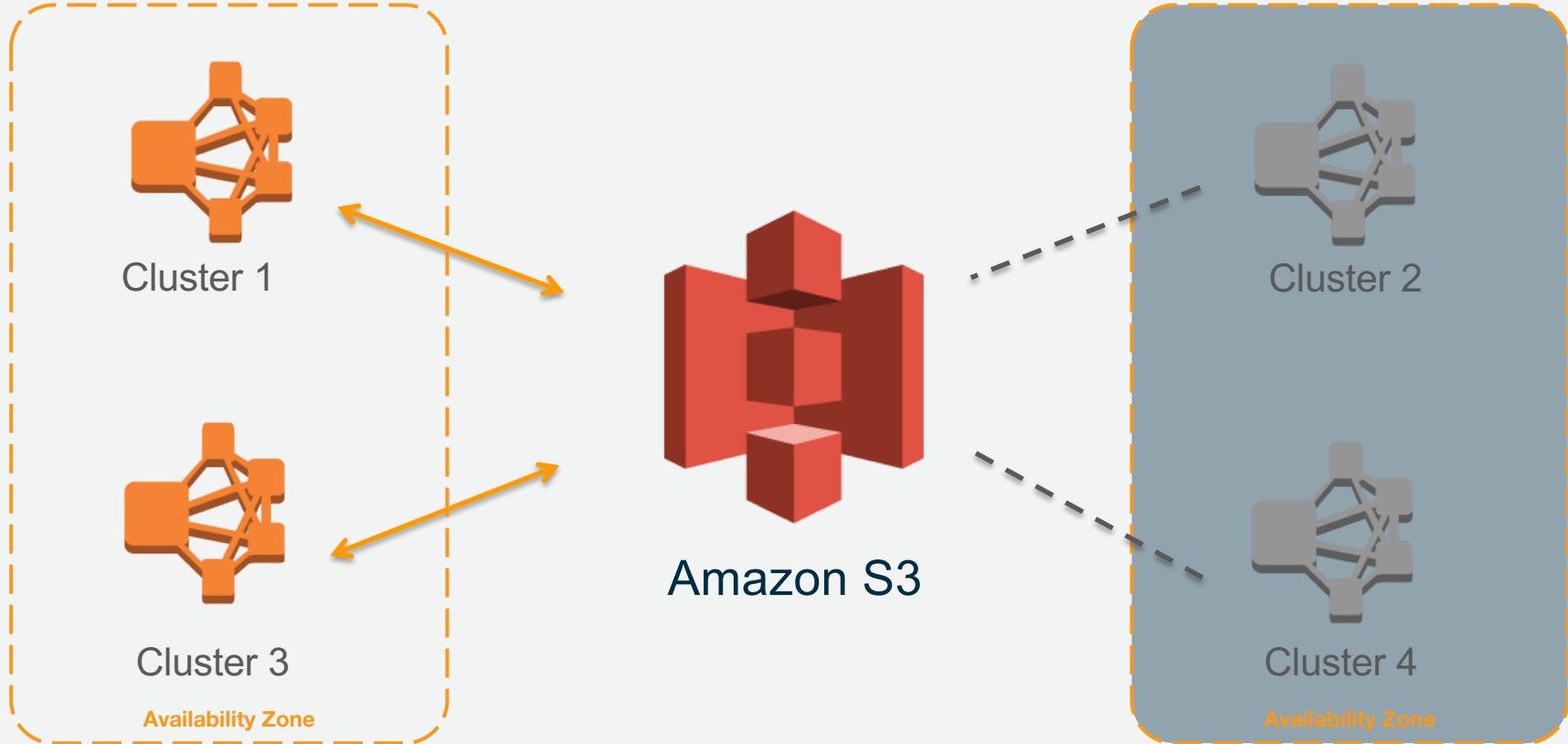
에러 핸들링 옵션

S3 암호화 지원

어플리케이션들이 투명하게 이용: s3://



# Disaster Recovery 빌트 인



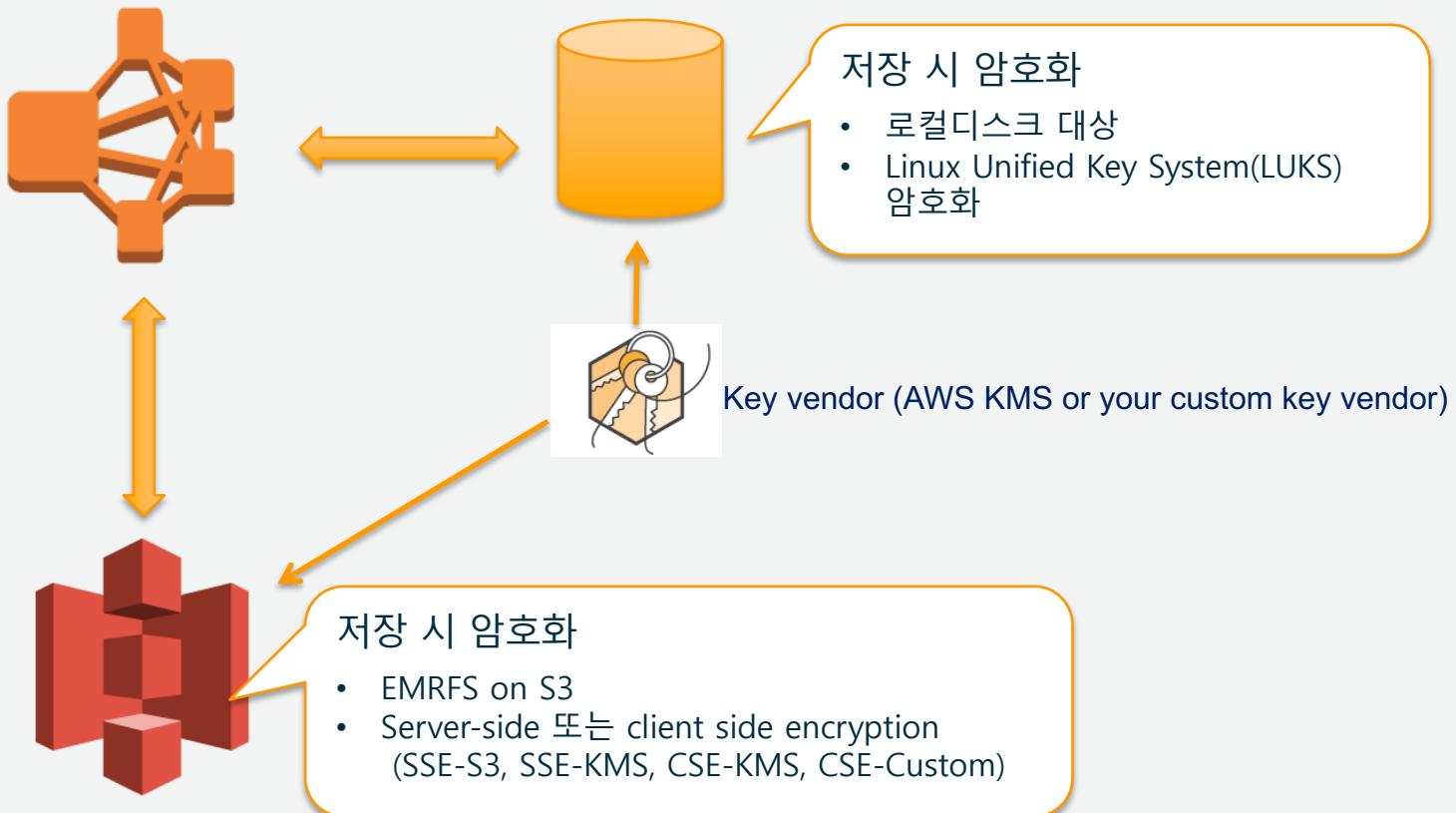
# Sqoop에서의 EMRFS 활용

```
sqoop import-all-tables  
--connect jdbc:mysql://ec2-34-xxx-xx-xxx.compute-1.amazonaws.com:3306/customer –  
username administrator –password password  
--warehouse-dir s3://Bucket_name/Directory_name  
--num-mappers 4  
--fields-terminated-by ''  
--lines-terminated-by '\n'  
--exclude-tables current_dept_emp,  
    dept_emp_latest_date
```

# 맵리듀스에서의 EMRFS 활용

```
loadlog = LOAD 's3://Bucket_name/Directory_name/Logfile_name' USING JsonLoader  
('host:chararray,user:chararray,method:chararray,path:chararray,  
code:INT,size:INT,referer:chararray,agent:chararray,time:chararray');  
ip_data = GROUP loadlog BY host ;  
ip_count = FOREACH ip_data GENERATE group AS time,COUNT(loadJson) as  
total_visits ;  
  
sort_data = RANK ip_count BY total_visits DESC ;  
store sort_data into 's3://Bucket_name/Logoutput_directory/' using PigStorage();
```

# EMR 저장 시 암호화 옵션

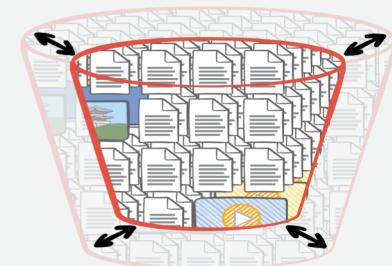
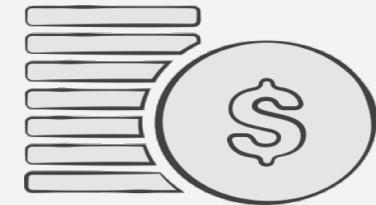


# HDFS 활용이 필요한 작업

- 반복적인 작업
  - 같은 데이터 세트를 한 번 이상 쓴다면
  - Spark 를 쓰는 것도 고려
- Disk I/O 집중적인 워크로드
- S3에 데이터 영구 보존, S3DistCp로 HDFS에 데이터를 복제하여 프로세싱

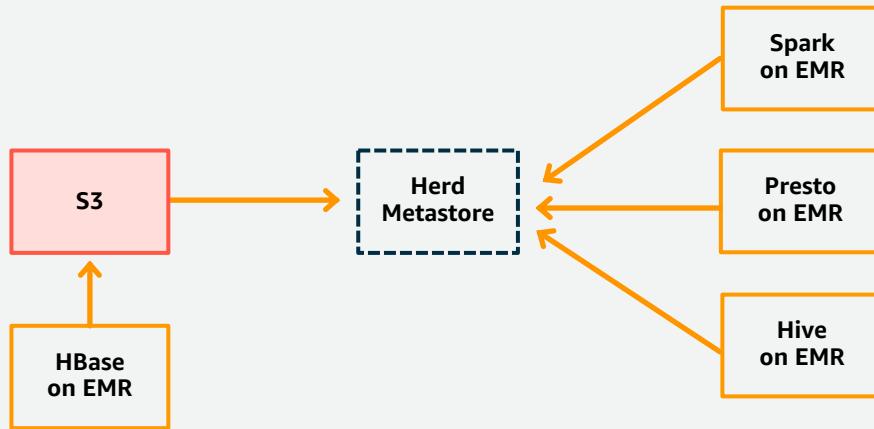
# 이점 : 주요 EMR 스토리지로서 Amazon S3

- 클러스터를 셧다운 가능
  - 비용 절감
- 데이터 내구성
  - 99.99999999%
- HDFS 확장을 위한 노드 추가 없음
  - 사실상 무제한 용량
- 데이터에 따라 Amazon S3 확장
  - IOPS와 데이터 저장량에 따라 자동 확장



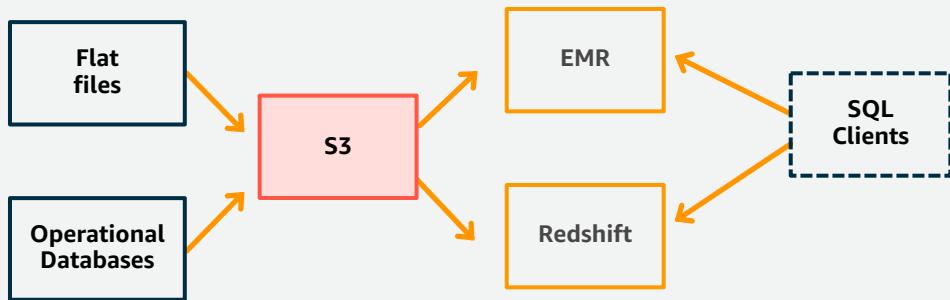
# EMR Design Pattern

# FINRA - S3를 EMR와 함께 Data Lake로 사용함



- 20PB 이상의 Trade 레코드에 대한 빠른 액세스 요구사항
- on-premise 시스템에서 마이그레이션
- Amazon EMR에서 Hbase를 이용해서 저장 및 서비스
- 데이터를 프로세스 하기위해 EMR 엔진(Spark, Presto, Hive) 사용
- On-premise 시스템 대비 60% 비용 절감

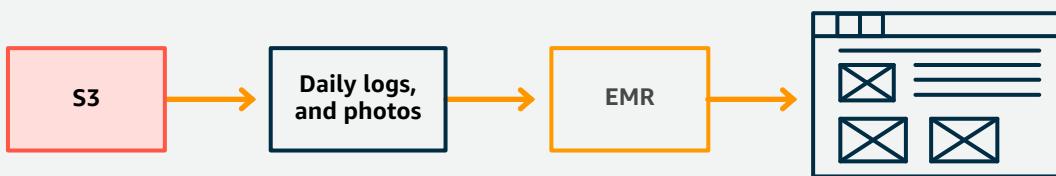
# Nasdaq - Data Lake 구축을 위해 AWS 사용



나스닥에서 운영하는 7개 거래소의  
데이터  
(orders, quotes, trade executions)

- 기존의 레거시 on-premises DW를 Amazon Redshift로 마이그레이션
- 매 거래일당 48억 행이 추가됨 (orders, trades, quotes)
- S3에 다중소스, 유효성 검사, 스테이지 데이터를 수집합니다.
- Redshift는 빠른 쿼리를 위해서 S3에서 데이터를 읽습니다.
- EMR의 Presto와 S3는 매우 큰 과거 데이터셋의 분석에 사용됩니다.

# Yelp - 개인화 웹사이트를 만들기 위해 AWS를 사용함



개인화  
(사이트를 방문한 사람,  
하이라이트 리뷰, 자동완성,  
인기 검색어, 광고)

- 매일 로그와 사진을 저장하고, 일당 1.2TB의 로그를 생성합니다.
- Amazon EMR 은 20개의 배치 스크립트를 수행합니다.
- 웹사이트에 개인화를 위한 스크립트의 결과물이 나타납니다.  
(사이트를 방문한 사람, 하이라이트 리뷰, 자동완성, 인기 검색어, 광고)
- 선행 하드웨어 비용을 \$55,000 감소

# Summary

# Amazon EMR 비용 절감 팁

- S3를 영구 데이터 저장용으로 쓰기. Presto, Hive, Spark 등으로 쿼리.
- 필요할 때만 컴퓨팅 리소스에 대해 비용 지불
- Amazon EC2 스팟 인스턴스로 80% 이상 비용 절감
- Amazon EC2 RI로 지속적인 워크로드 처리
- 워크로드에 적합한 클러스터 사이즈를 위한 Auto Scaling 활용

# Key Benefits of EMR

- 쓰기 쉽고, 수 분만에 클러스터 실행
- 손쉬운 스케일 업/다운 (10개 노드 -> 200개 노드 -> 10개 노드)
- 저 비용. 쓴 만큼만 지불
- 높은 보안성. VPC안에서 실행
- 손쉬운 클러스터 사이즈, 노드 타입, Hadoop 버전 등 변경
- DynamoDB, EC2, S3, VPC, Cloudwatch 등과 통합.
- Root 접근 + bootstrap 액션 = 유저가 자유롭게 관리
- 개발자가 하둡 관리에 쓰는 시간을 줄이고 데이터에 대해 더 가치를 생성하게끔 도와줌

# Q&A

AWS Solutions Architect