클라우드 관계형 데이터베이스
Amazon Aurora
&
오픈소스를 활용한 실시간 데이터 분석

양승도 | 솔루션즈 아키텍트 | Amazon Web Services

*"When we speak of free software, we are referring to **freedom**, not price."*

Richard Stallmann
Free Software Foundation, GNU Project

View on GitHub

Amazon GitHub Projects

http://amzn.github.io

Amazon	https://github.com/amzn
Amazon Labs	https://github.com/amznlabs
AWS	https://github.com/aws and https://github.com/amazonwebservices
AWS Labs	https://github.com/awslabs
The Blindsight Corporation	https://github.com/blindsightcorp
Carbonado	https://github.com/Carbonado
Goodreads	https://github.com/goodreads
Ivona	https://github.com/IvonaSoftware
Twitch	https://github.com/twitchtv, https://github.com/TwitchScience, and https://github.com/justintv
Zappos	https://github.com/Zappos

Amazon

Repositories | People 0

Filters | Find a repository...

login-and-pay-with-amazon-sdk-java Java

Amazon Payments - Login and Pay with Amazon Java SDK

Updated a day ago

exoplayer-amazon-port Jav

Official port of ExoPlayer for Amazon devices

Updated a day ago

alexa-skills-kit-js

SDK and example code for building voice-enabled skills for the Amazon Echo.

Updated 6 days ago

amazon-payments-magento-plugin PHP ★ 49 ⑂ 26

Amazon Payments for Magento Plugin

Updated 13 days ago

Amazon - Labs

Seattle, WA http://amzn.github.io/

Repositories | People 1

Filters | Find a repository...

Amazon DSSTNE: Deep Scalable Sparse Tensor Network Engine

DSSTNE (pronounced "Destiny") is an open source software library for training and deploying deep neural networks using GPUs. Amazon engineers built DSSTNE to solve deep learning problems at Amazon's scale. DSSTNE is built for production deployment of real-world deep learning applications, emphasizing speed and scale over experimental flexibility.

DSSTNE was built with a number of features for production workloads:

- **Multi-GPU Scale**: Training and prediction both scale out to use multiple GPUs, spreading out computation and storage in a model-parallel fashion for each layer.
- **Large Layers**: Model-parallel scaling enables larger networks than are possible with a single GPU.
- **Sparse Data**: DSSTNE is optimized for fast performance on sparse datasets. Custom GPU kernels perform sparse computation on the GPU, without filling in lots of zeroes.

oss-dashboard Ruby ★ 32 ⑂ 4

A dashboard for viewing many GitHub organizations at once.

Updated on Apr 28

Amazon Web Services

Seattle, WA http://aws.amazon.com

Repositories People 29

Filters Find a repository…

aws-sdk-go

AWS SDK for the Go programming language.

Updated 9 hours ago

aws-sdk-java-samples

Handwritten and generated code samples for the AWS SDK for Java documentation

Updated 15 hours ago

aws-toolkit-eclipse

Official mirror of the AWS Toolkit for Eclipse. For more information on the toolkit, see web site:

Updated 16 hours ago

aws-cli

Python ★ 3,904 ⑂ 603

Universal Command Line Interface for Amazon Web Services

Updated 21 hours ago

Amazon Web Services - Labs

Seattle, WA http://aws.amazon.com/

Repositories People 23

Filters Find a repository…

s2n is a C99 implementation of the TLS/SSL protocols that is designed to be simple, small, fast, and with security as a priority. It is released and licensed under the Apache Software License 2.0.

aws-quickstart

Shell ★ 75 ⑂ 43

Official repository for AWS Quick Start

Updated 21 hours ago

aws-sdk-android-samples

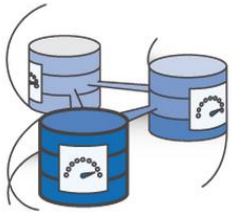
Java ★ 230 ⑂ 286

This repository has samples that demonstrate various aspects of the AWS Mobile SDK for Android, you can get the SDK source on Github

Amazon
RDS

- 관계형 데이터베이스
- 쉽고 빠른 구성
- 반복 적인 관리작업을 대신 수행
- 다양한 관계형 데이터베이스 옵션 제공
- 쉽고 빠른 확장
- 손쉬운 고 가용성 구성

Amazon RDS 데이터베이스 엔진



Aurora

PostgreSQL

MySQL®

MariaDB

Microsoft® SQL Server®

ORACLE®

Amazon Aurora는?

MySQL 호환 관계형 데이터베이스 엔진

상용 데이터베이스의 **성능**과 **가용성** 제공

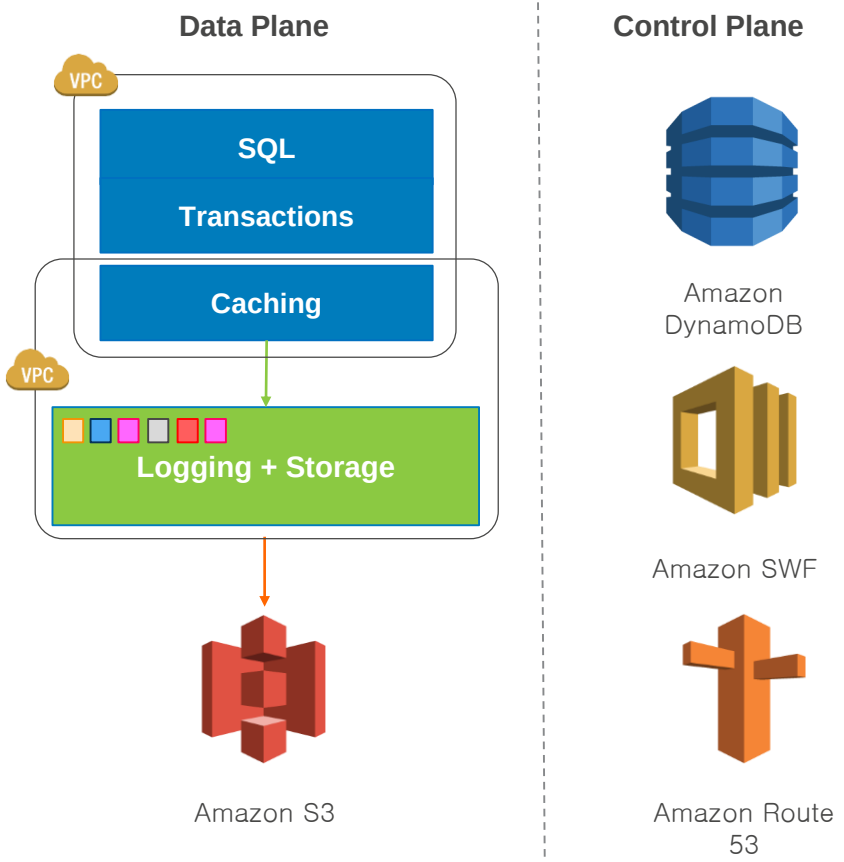
오픈소스 데이터베이스의 **효율성**과 **비용**

클라우드를 위한 데이터베이스 아키텍처

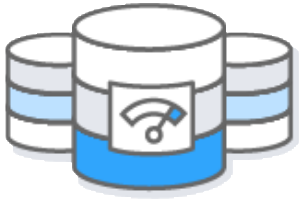
1. 로깅 및 스토리지를 멀티-테넌시 스케일-아웃 기반 DB 최적화 스토리지 서비스로 전환

2. 서비스 내부에 Amazon EC2, VPC, DynamoDB, SWF 및 Route 53 등 다른 AWS 서비스들 사용

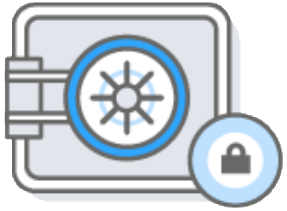
3. 연속적인 백업을 위한 Amazon S3와 통합으로 99.999999999% 내구성 제공



Amazon Aurora 주요 특징



고성능

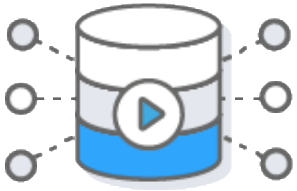


뛰어난 보안

MySQL과 호환

뛰어난 확장성

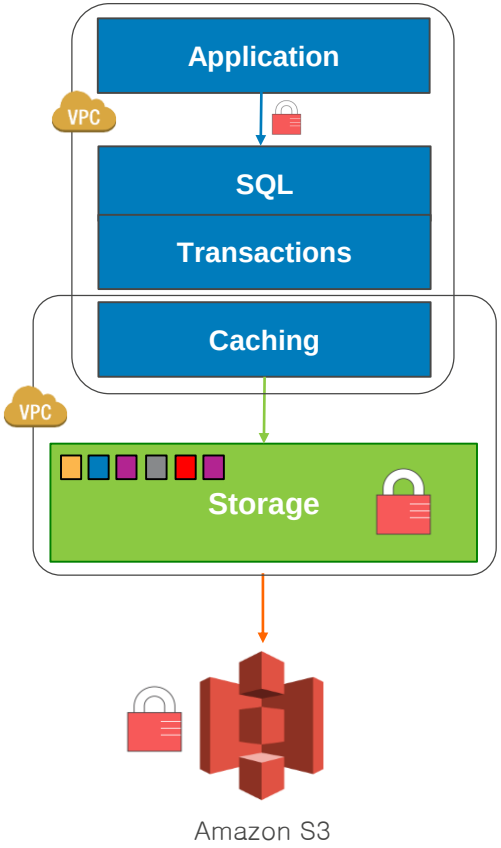
높은 가용성 및
내구성



완전 관리형

뛰어난 보안

- 저장 시 암호화
  - AES-256 및 하드웨어 가속
  - 디스크 및 S3 내 모든 블록들은 암호화
  - AWS KMS 를 통한 키 관리
- 전송 시 암호화 – SSL
- Amazon VPC를 통한 네트워크 격리
- 노드에 직접 접근 없음
- 산업 표준의 보안 및 데이터 보호 인증서 지원

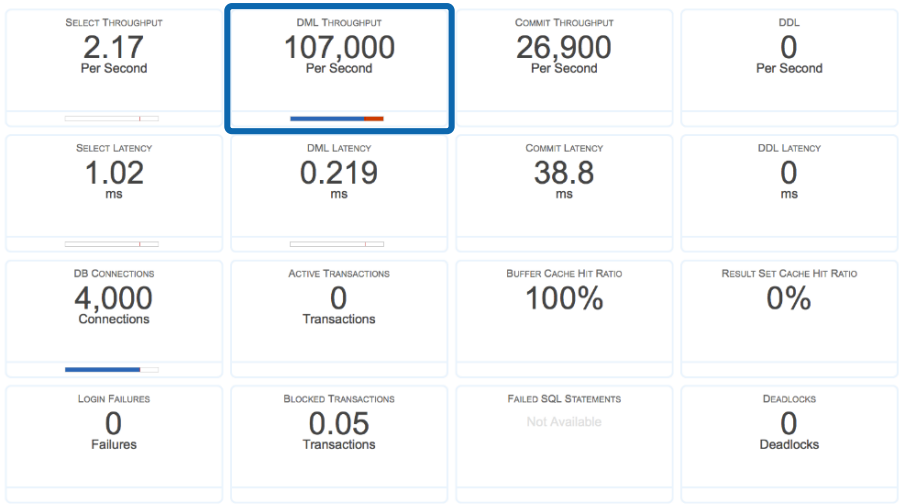


You’ve probably heard about our benchmark numbers…

SQL 성능 테스트 결과

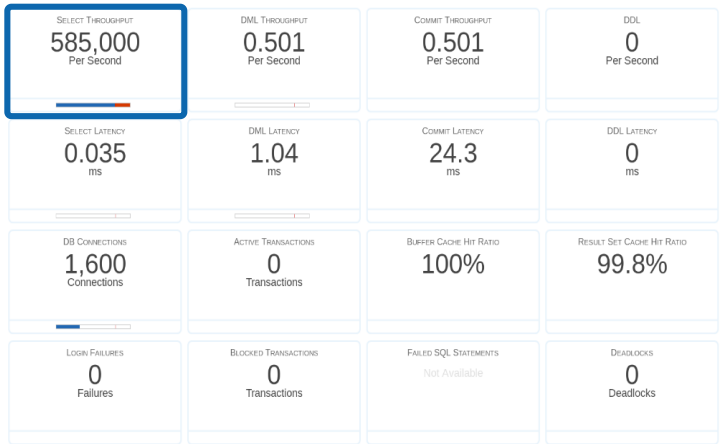
Amazon Aurora r3.8xl (32 vCPU, 244 GiB RAM) 사용 MySQL SysBench 성능 테스트

WRITE PERFORMANCE



4 클라이언트 머신당 각 1,000 connections

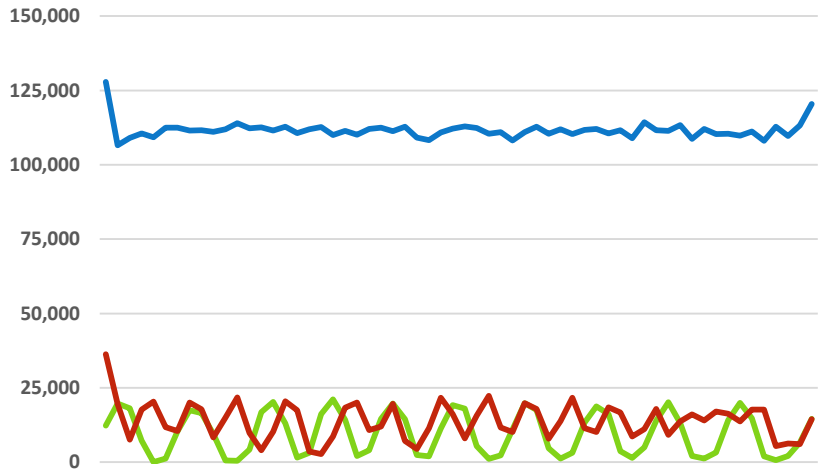
READ PERFORMANCE



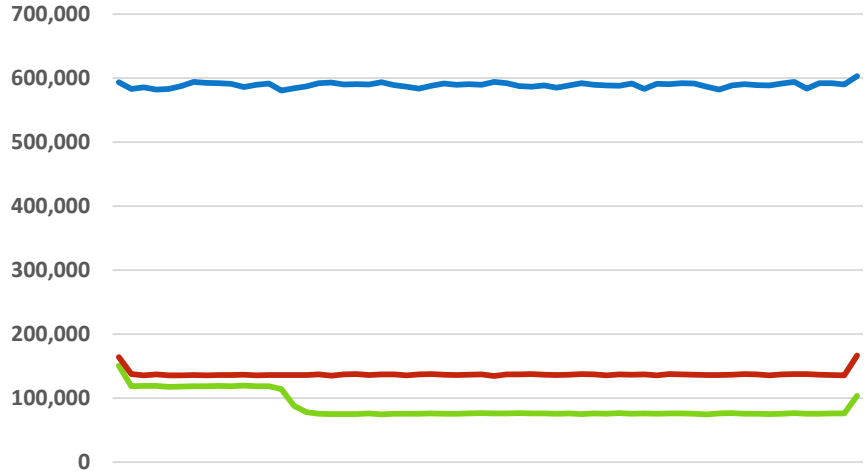
단일 클라이언트 머신 1,600 connections

RDS MySQL 5.6 & 5.7 보다 5X 빠른

WRITE PERFORMANCE



READ PERFORMANCE



MySQL SysBench results

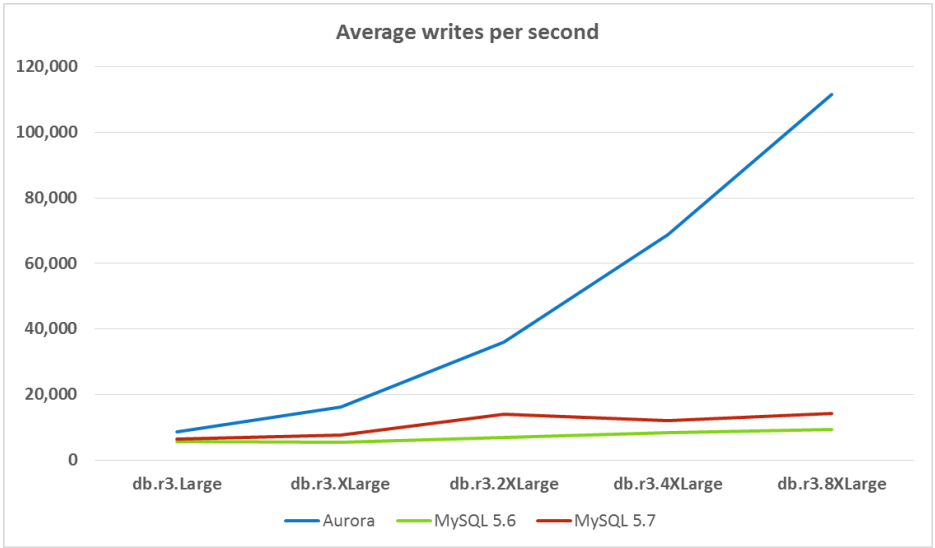
R3.8XL: 32 cores / 244 GiB RAM

Aurora — MySQL 5.6 — MySQL 5.7

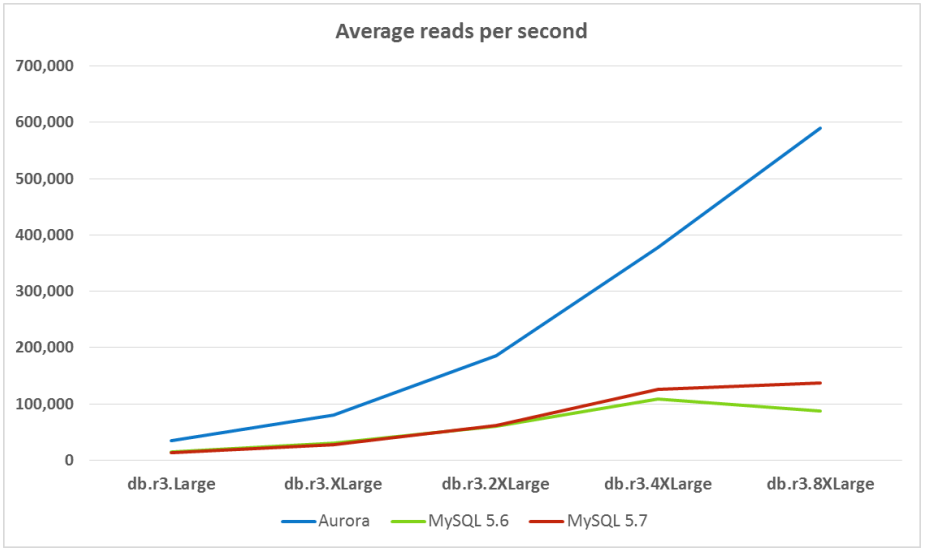
Five times higher throughput than stock MySQL based on industry standard benchmarks.

인스턴스 사이즈에 따른 성능

WRITE PERFORMANCE



READ PERFORMANCE



Aurora **MySQL 5.6** **MySQL 5.7**

Aurora scales with instance size for both read and write.

읽기 복제에 따른 지연 감소

Updates per second	Amazon Aurora	RDS MySQL 30 K IOPS (single AZ)
1,000	2.62 ms	0 s
2,000	3.42 ms	1 s
5,000	3.94 ms	60 s
10,000	5.38 ms	300 s

SysBench OLTP 워크로드
250 테이블

UP TO

500x

LOWER LAG

성능을 위한 Aurora 아키텍처

DO LESS WORK

I/O의 감소

네트워크 패킷 최소화

기존 결과를 캐시

데이터베이스 엔진 오프로드

BE MORE EFFICIENT

비동기식 처리

응답속도 경로 감소

락-없는 데이터 구조 사용

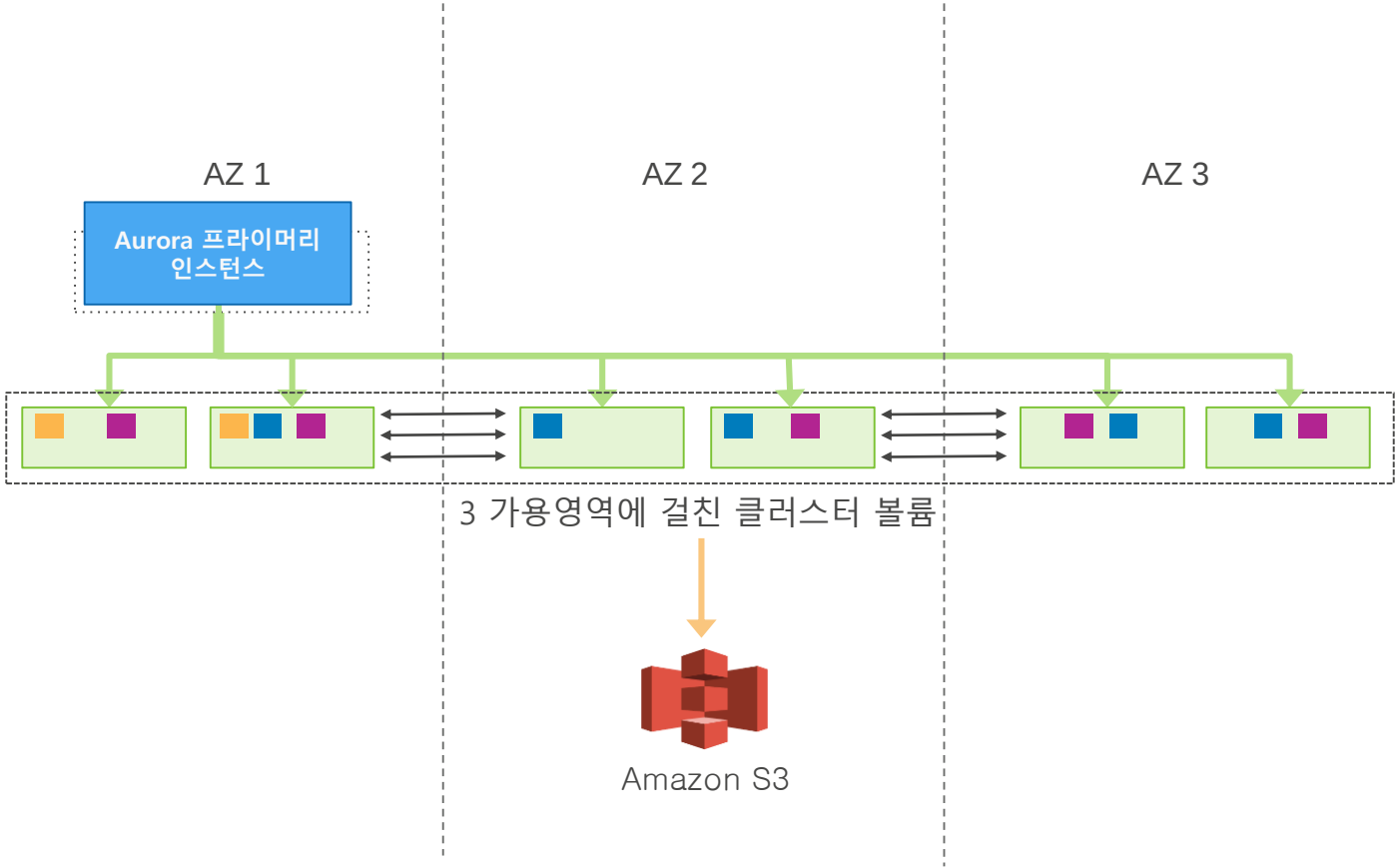
배치 수행 동시 처리

DATABASES ARE ALL ABOUT I/O

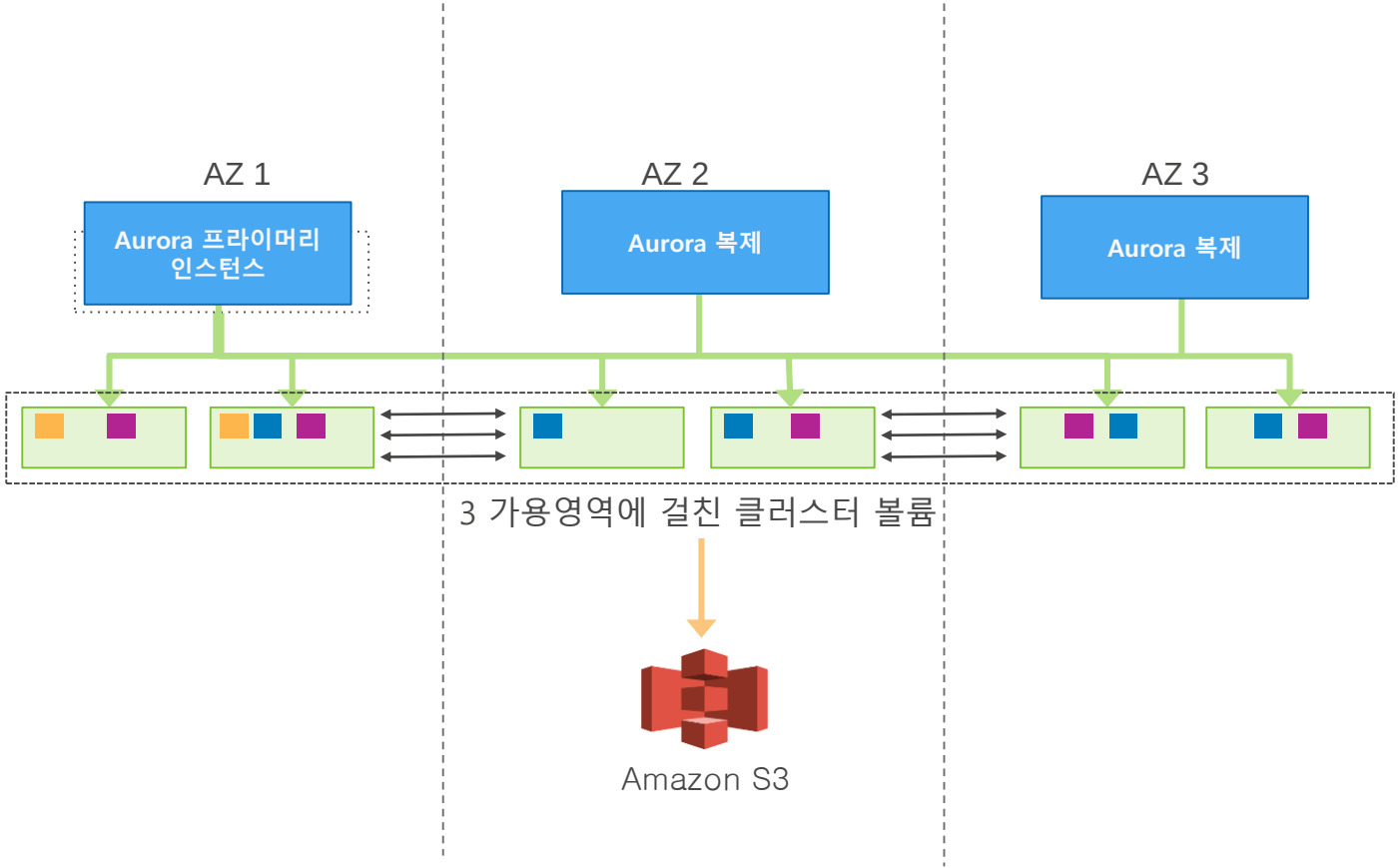
NETWORK-ATTACHED STORAGE IS ALL ABOUT PACKETS/SECOND

HIGH-THROUGHPUT PROCESSING DOES NOT ALLOW CONTEXT SWITCHES

Aurora 클러스터

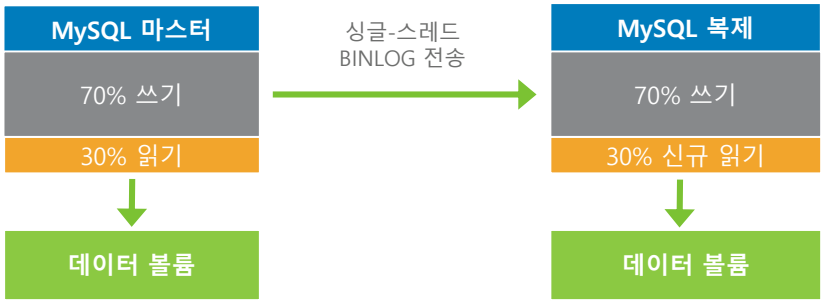


Aurora 클러스터 및 읽기 복제



Aurora I/O 트래픽

MYSQL READ SCALING



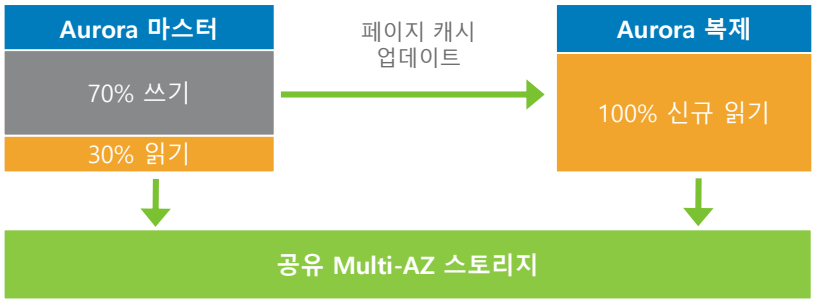
Logical: SQL 문을 복제에 적용

쓰기 부하는 양쪽 노드에서 유사

별도 스토리지

마스터 및 복제 사이에 데이터 차이 존재

AMAZON AURORA READ SCALING



Physical: 마스터에서 복제로 redo를 전송

복제는 스토리지를 공유.

쓰기 수행 없음

캐시된 페이지는 Redo 적용

Amazon Aurora의 고가용성

Amazon Aurora의 스토리지

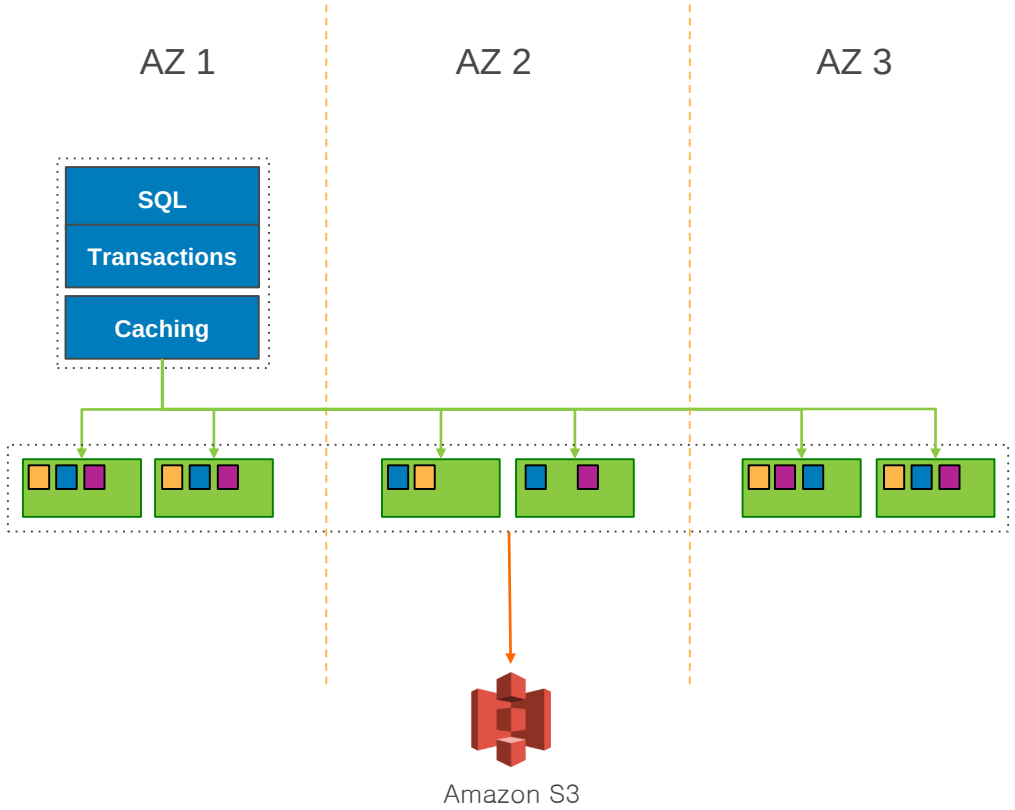
기본 고가용성

- 3 가용영역에 6-way 복제
- 4 / 6 쓰기, 3 / 6 읽기 쿼럼
- S3 저장소에 연속 백업

SSD, 스케일-아웃, 멀티-테넌트 스토리지

- 연속적 스토리지 확장
- 최대 64TB 크기
- 사용한만큼만 지불

로그-구조 기반 스토리지

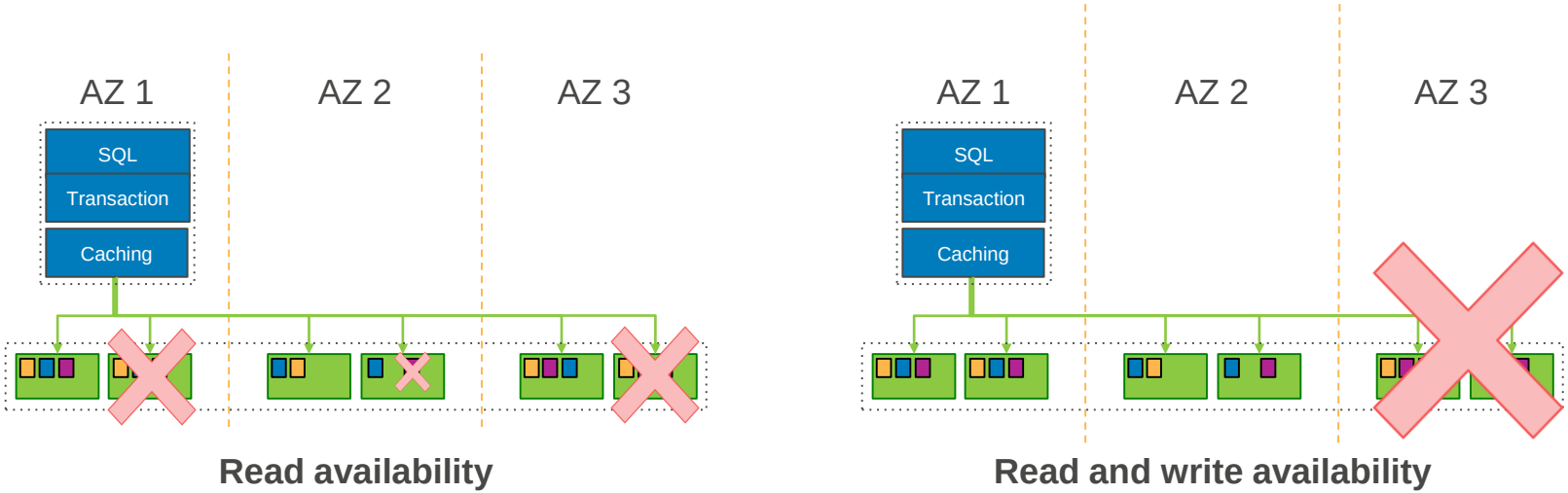


스토리지 자가 치유 및 장애 내구성

자동 장애 감지, 복제, 복구

2개의 복제 및 1개 가용 영역 장애는 읽기 및 쓰기 가용성에 영향 없음

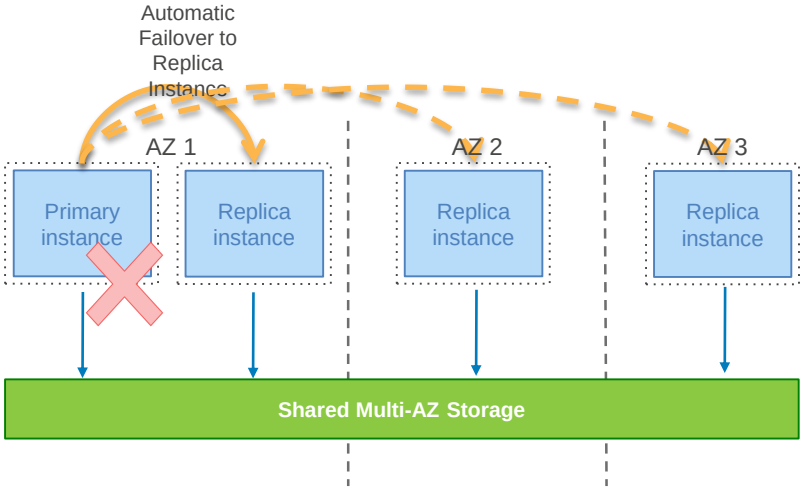
3개의 복제 장애에도 읽기 가용성에 영향 없음



Amazon Aurora의 인스턴스 자동 페일-오버

읽기 복제 있는 경우

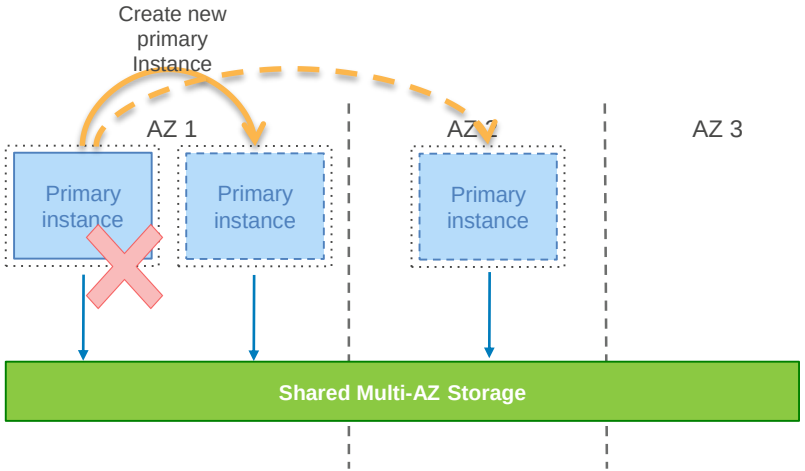
- 기존 복제를 새 기본 인스턴스로 승격
- 페일오버 대상 인스턴스 **우선 순위 지정 가능**
- DB 클러스터 엔드포인트 유지하며, 신규 기본 인스턴스로 DNS 레코드 변경
- 일반적으로 1분 이내에 완료



Aurora Replica가 있는 경우

읽기 복제 없는 경우

- 동일 가용 영역에 새 DB 인스턴스 생성 시도
- 생성 불가 시 다른 가용 영역에 신규 DB 인스턴스 생성 시도
- 일반적으로 15분 이내에 완료



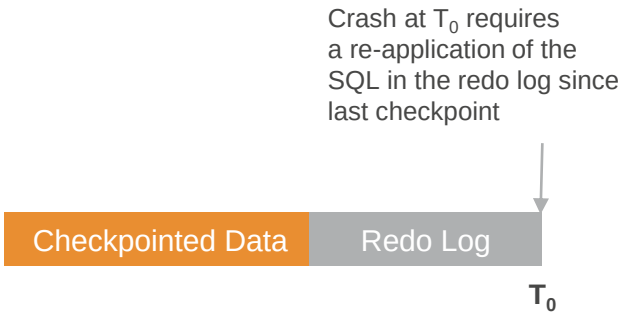
Aurora Replica가 없는 경우

신속한 크래시 복구

기존 데이터베이스

최종 체크포인트 이후 로그 재생 필요

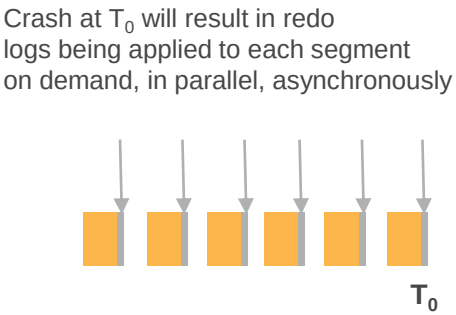
MySQL은 싱글-쓰레드 동작 및 다량의 디스크 억세스 필요



Amazon Aurora

스토리지 수준에서 읽기 시 온-디맨드 형태로 Redo 레코드 재생

병렬, 분산, 비동기



캐시 유지

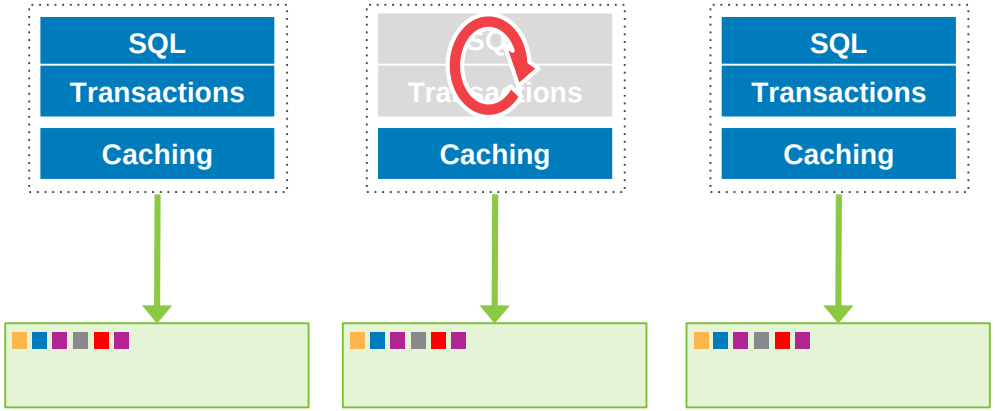
데이터베이스 프로세스와 캐시의 분리

데이터베이스 재기동 이벤트 시에도 캐시 웜(warm) 상태 유지

전체 캐시 활성화가 신속

즉각적인 크래시 복구 + 캐시 유지 = 빠르고 손쉬운 DB 장애 복구

Caching process is outside the DB process and remains warm across a database restart.



Compatible with the MySQL ecosystem

Well established MySQL ecosystem

“We ran our compatibility test suites against Amazon Aurora and everything just worked." - Dan Jewett, Vice President of Product Management at Tableau

Business Intelligence

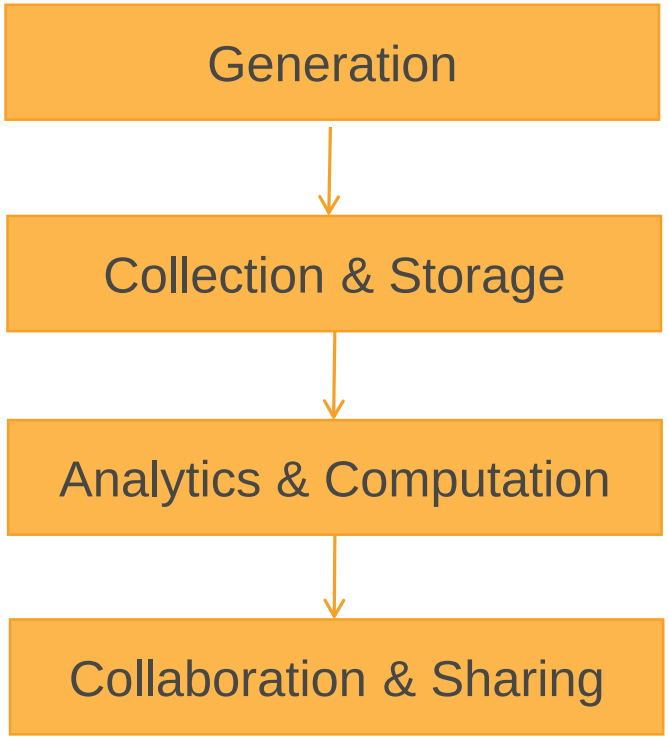
Data Integration

Query and Monitoring

SI and Consulting

Source: Amazon

How does Open-Source & Cloud fit into Data Analytics?


Generation
Collection & Storage
Analytics & Computation
Collaboration & Sharing

More devices
Lower cost
Higher throughput

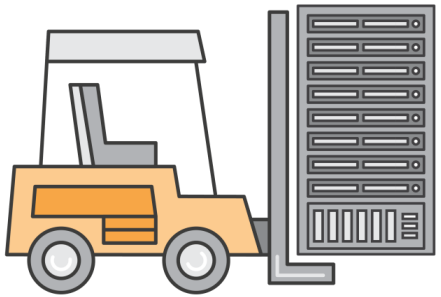
Generation

Collection & Storage

Analytics & Computation

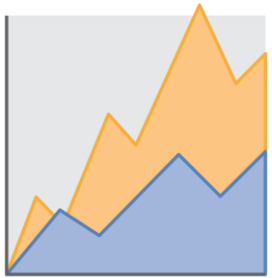
Collaboration & Sharing

제약 사항



Amazon Web Services helps remove constraints

데이터 분석의 세가지 유형



Retrospective

분석 또는 보고

Here-and-now

실시간 분석 및
대쉬보드

Predictions

보다 스마트한
서비스

데이터 분석의 세가지 유형

Retrospective

분석 또는 보고

Here-and-now

실시간 분석 및 대쉬보드

Predictions

보다 스마트한 서비스

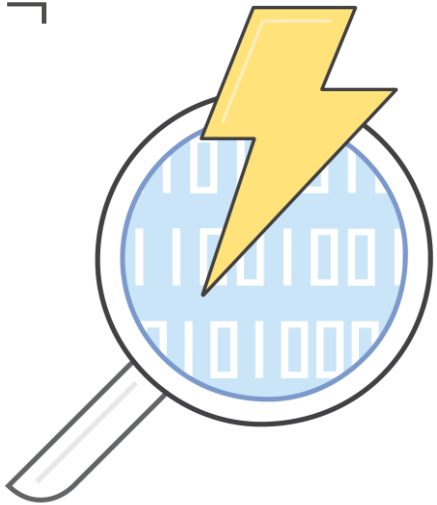
How Fast is Real-Time?

*“There’s no such thing as real time, only **near-real time**. Typically when we talk about real-time, we mean architectures that allow to respond to data **without persisting** it to a database first!”*

John Akred
CTO, Silicon Valley Data Science

So what is near real-time?

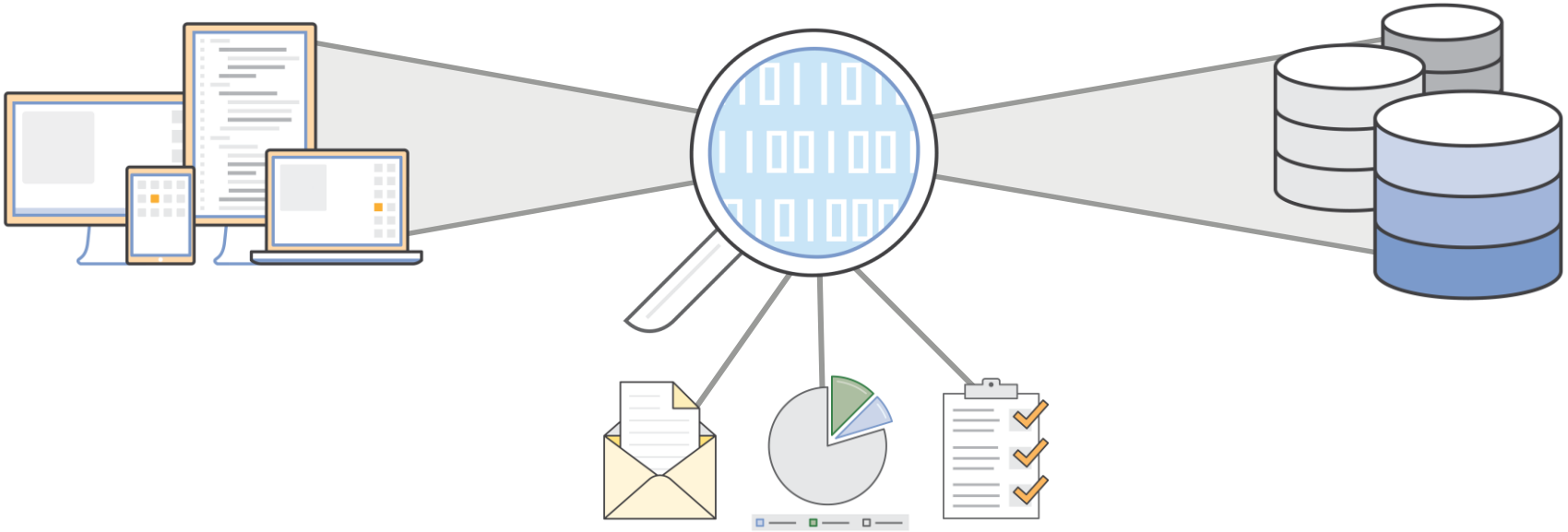
- 데이터가 도착하자마자 처리할 수 있는 능력
- *다시 말하면, "미래"가 아닌 "현재" 상태의 데이터를 처리하는 것*



- *그렇다면 "현재"란?*
  - *eCommerce – Attention span of a potential customer*
  - *Options Trader – Milliseconds*
  - *Guided Missile – Microseconds*

Solution: 스트림 프로세싱

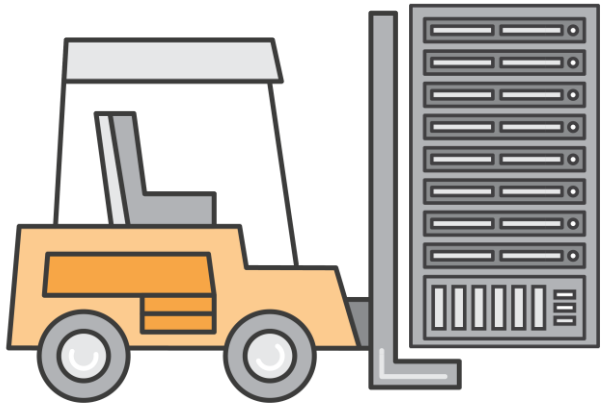
- Stream “storage” which allows processing events as they come in and react accordingly



What do we expect from a real-time data stream?

Real-Time Data Stream 에 대한 기대

- Real-time 데이터 스트림에 무엇을 기대합니까?
  - 고 가용성
  - 확장성
  - 장애복구 능력
  - 내구성(임시)
- 어떻게 가능한가요?
  - 다수의 데이터 센터 설비
  - 자동으로 확장가능한 인프라
  - 글로벌 부하 분산
  - 기타.



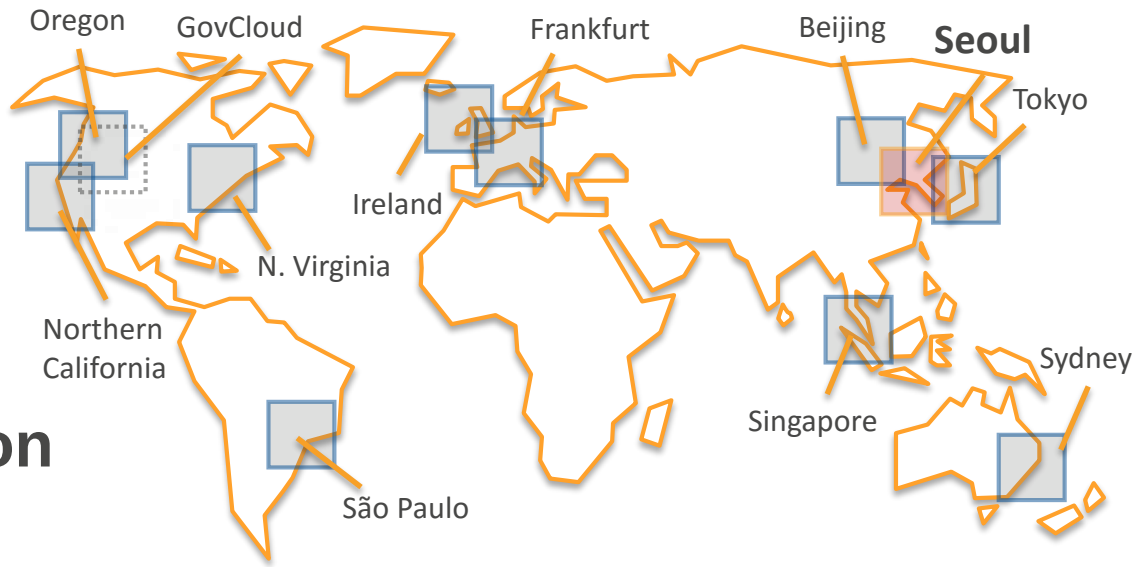
AWS Global Infrastructure

12 Regions

33 Availability Zones

55 Edge Locations

Continuous Expansion



Ingest

Amazon Mobile Analytics

AWS Import/Export

Store

Amazon RDS

Amazon Kinesis

Amazon DynamoDB

Amazon CloudSearch

Amazon S3

Amazon Glacier

Process

Amazon EMR

Amazon Machine Learning

Amazon Redshift

AWS Data Pipeline

Amazon Lambda

Amazon EC2

Visualize

Ingest

Store

Process

Visualize

Amazon Mobile Analytics

AWS Import/Export

Amazon RDS

Amazon Kinesis

Amazon DynamoDB

Amazon CloudSearch

Amazon S3

Amazon Glacier

Amazon EMR

Amazon Machine Learning

Amazon Redshift

AWS Data Pipeline

Amazon Lambda

Amazon EC2

Fluentd: 오픈소스 로그 수집

- Fluentd is an **open source data collector** to unify data collection and consumption
- Integration into **many data sources** (App Logs, Syslogs, Twitter etc.)
- Direct integration into **AWS** such as S3 & Kinesis


```
<source>
  type tail
  format apache2
  path /var/log/apache2/access_log
  tag s3.apache.access
</source>
<match s3.*.*>
  type s3
  s3_bucket myweblogs
  path logs/
</match>
```

https://github.com/fluent/fluentd/

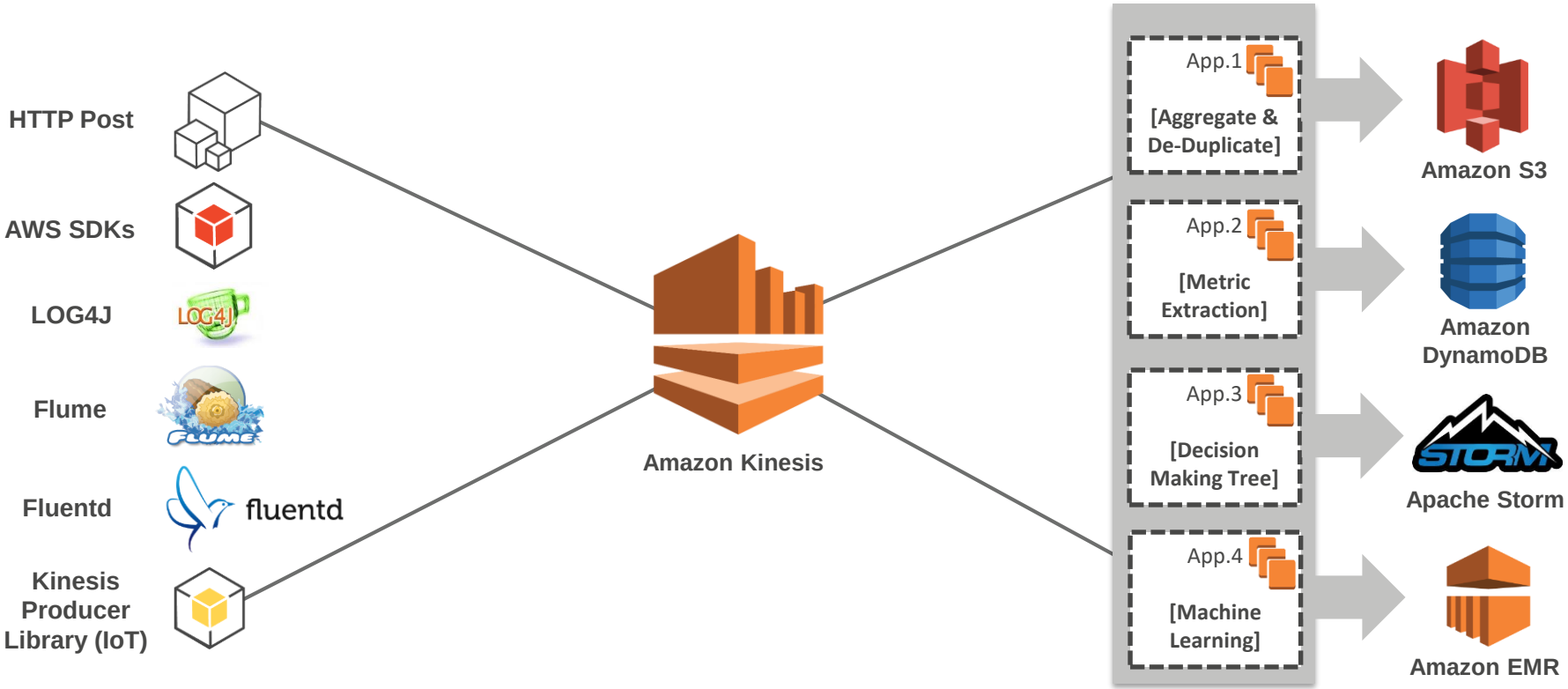

Ingest
Store
Process
Visualize
LOG4J
Flume
fluentd
Amazon Mobile Analytics
AWS Import/Export
Amazon RDS
Amazon Kinesis
Amazon DynamoDB
Amazon CloudSearch
Amazon S3
Amazon Glacier
Amazon EMR
Amazon Machine Learning
Amazon Redshift
AWS Data Pipeline
Amazon Lambda
Amazon EC2
awsmarketplace
pentaho
kibana

Real Time Data Stream: Amazon Kinesis

- 대용량 분산 스트림에 대한 Real-Time 데이터 분석
- 초당 수백만 이벤트를 처리할 수 있는 탄력적인 용량
- 스트림에 입력되는 이벤트에 따라 Real-Time 으로 반응
- 3군데 저장소에 복제하는 신뢰할 수 있는 스트림


CLASH of CLANS
4TB/day Amazon S3
Archival - Amazon Glacier
24/7 Hadoop -clusters EC2
Kinesis for Real-Time
SUP
ERC
ELL

Amazon Kinesis: 생산자와 소비자



Apache Spark Streaming

- Apache Spark is an **in-memory analytics** cluster using RDD for fast processing
- Spark streaming can read **directly** from an **Amazon Kinesis stream**


```
KinesisUtils.createStream('twitter-stream')
    .filter(_.getText.contains("Open-Source"))
    .countByWindow(Seconds(5))
```

Counting tweets on a sliding window

Ingest

Store

Process

Visualize

Amazon Mobile Analytics

AWS Import/Export

Amazon RDS

Amazon Kinesis

Amazon DynamoDB

Amazon CloudSearch

Amazon S3

Amazon Glacier

Amazon EMR

Amazon Machine Learning

Amazon Redshift

AWS Data Pipeline

Amazon Lambda

Amazon EC2

React in Real-Time: Amazon Lambda

- 완벽하게 관리되고 고가용성이 지원되는 “서버없는 컴퓨팅” & “클라우드 함수” 서비스
- 호출 또는 상태 변화를 통해 트리거
- 수신 이벤트 비율에 맞게 자동적으로 확장
- 모든 수신 이벤트에 반응할 수 있게 Amazon Kinesis 스트림에 연결 가능

Ingest

Store

Process

Visualize

Amazon Mobile Analytics

AWS Import/Export

Amazon RDS

Amazon Kinesis

Amazon DynamoDB

Amazon CloudSearch

Amazon S3

Amazon Glacier

Amazon EMR

Amazon Machine Learning

Amazon Redshift

AWS Data Pipeline

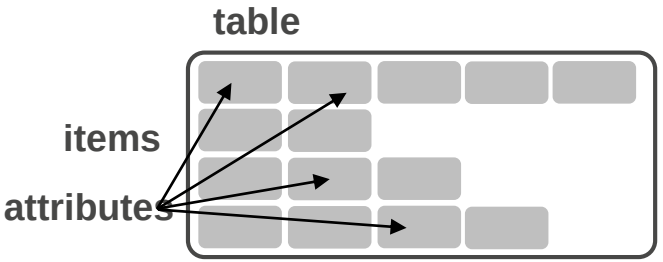
Amazon Lambda

Amazon EC2

Amazon DynamoDB

완전 관리형 **NoSQL** 데이터베이스 **서비스**

- Schemaless Data Model
- Seamless scalability
- No storage or throughput limits
- Consistent low latency performance
- High durability and availability



SHAZAM

Search Music

Experience more

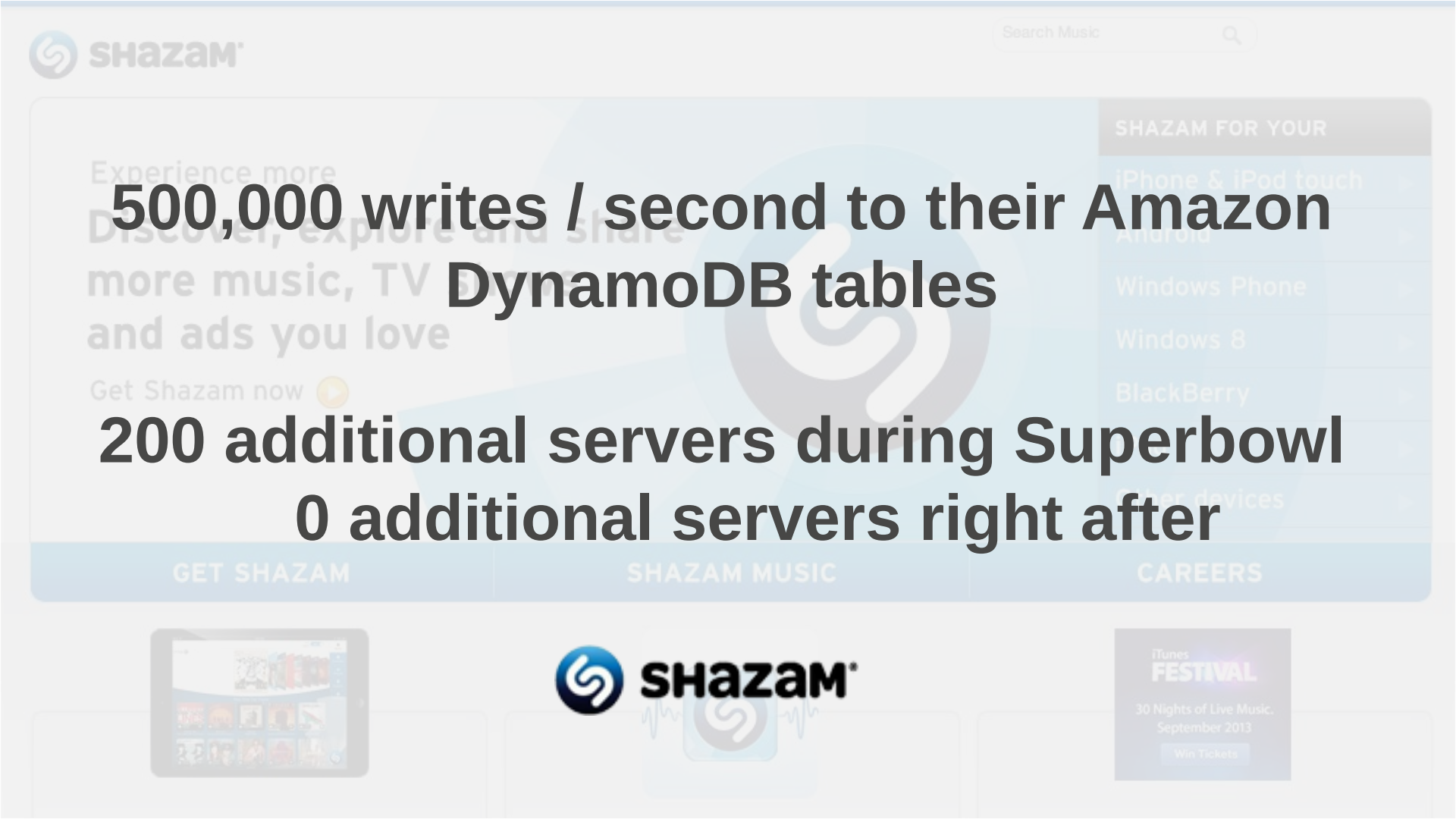
Discover, explore and share more music, TV shows and ads you love

Get Shazam now

SHAZAM FOR YOUR

- iPhone & iPod touch
- Android
- Windows Phone
- Windows 8
- BlackBerry
- iPad
- Other devices

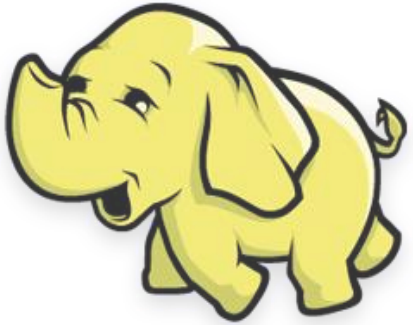
GET SHAZAM | SHAZAM MUSIC | CAREERS

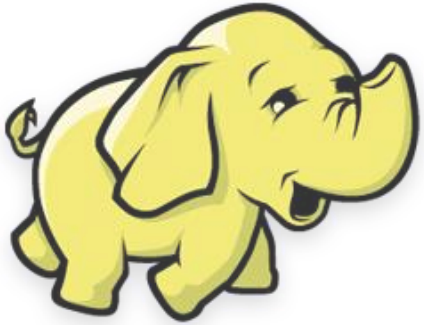


500,000 writes / second to their Amazon DynamoDB tables

200 additional servers during Superbowl

0 additional servers right after





1 instance x 100 hours = 100 instances x 1 hour

Ingest

Store

Process

Visualize

Amazon Mobile Analytics

AWS Import/Export

Amazon RDS

Amazon Kinesis

Amazon DynamoDB

Amazon CloudSearch

Amazon S3

Amazon Glacier

Amazon EMR

Amazon Machine Learning

Amazon Redshift

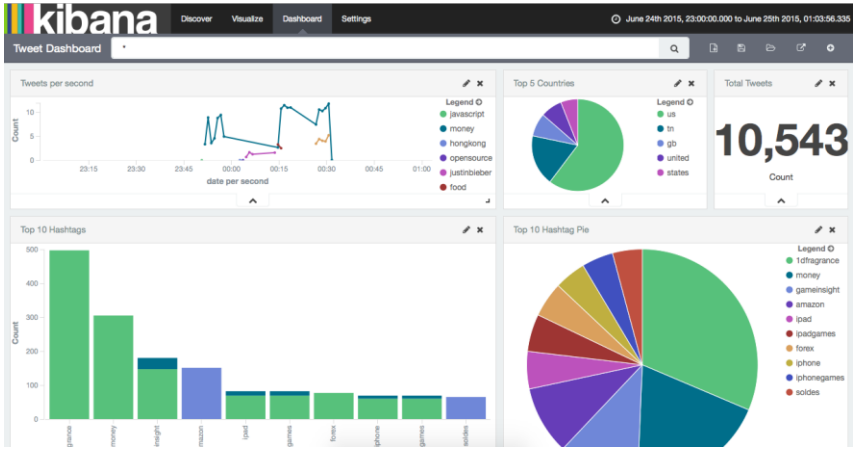
AWS Data Pipeline

Amazon Lambda

Amazon EC2

Kibana: 오픈소스 시각화 도구

- Kibana is an open-source project of Elastic.IO to **visualize data** in browser
- Uses **Elasticsearch** as indexing engine (based on Apache Lucene)
- Elasticsearch on **Hadoop** available (es-hadoop)



https://github.com/elastic/kibana

Let’s put it all together !

Live Twitter Feed Analysis

Twitter Blog* - On a typical day:

- More than 500 million Tweets sent
- Average 5,700 TPS

* https://blog.twitter.com/2013/new-tweets-per-second-record-and-how

오픈소스 & 클라우드

효율적인 데이터 저장/처리