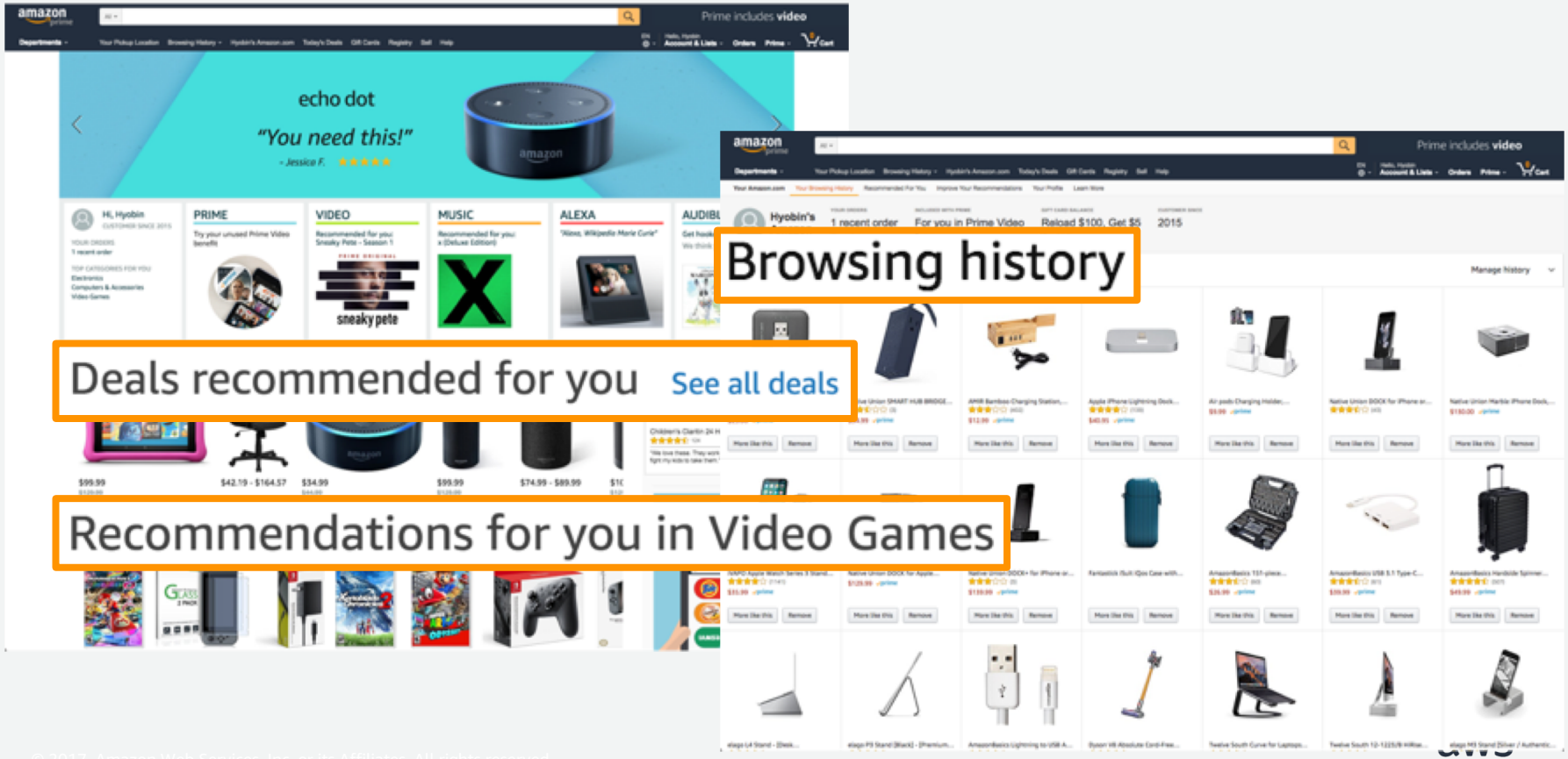
Data Life Cycle on AWS

Data Lab 소개

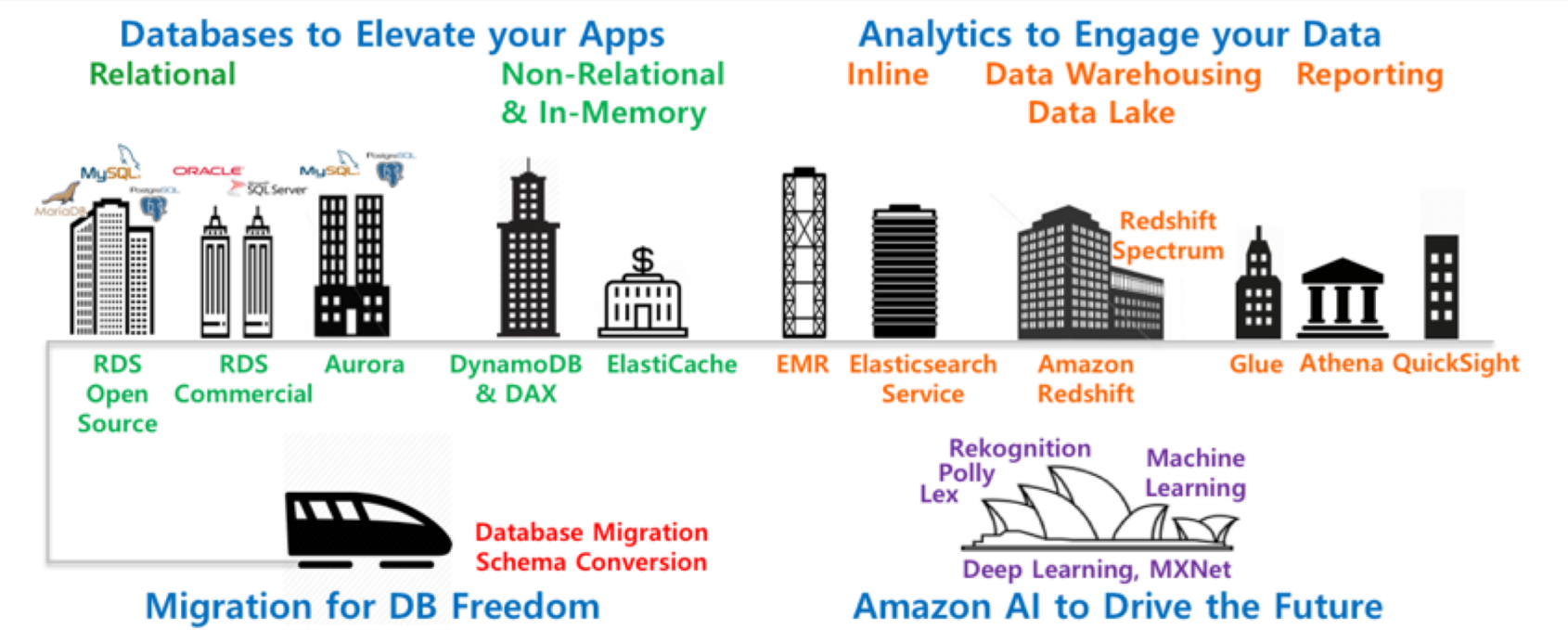
Choi, Yoojeong, Database SA, AWS

Background

실 환경에서 발생하는 다양한 종류의 워크로드



AWS에 존재하는 다양한 데이터 관련 서비스

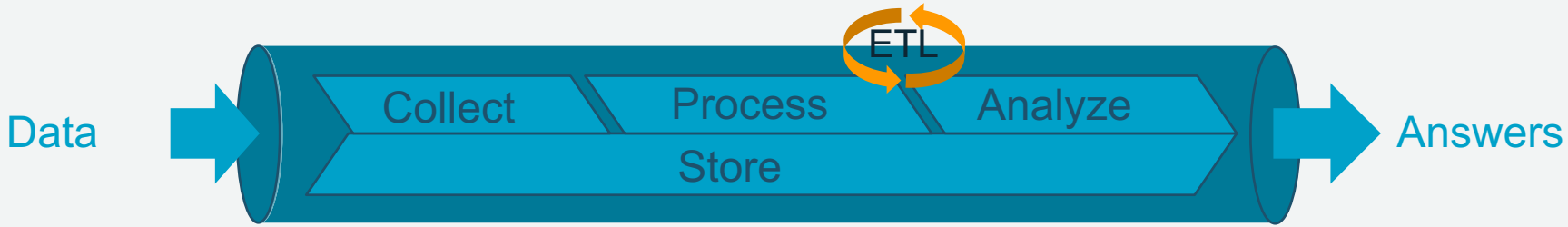


왜 Technical Bootcamp를 만들었을까요?

VoC

- 내 워크로드에 적합한 AWS 데이터서비스는 무엇인지 모르겠다..
- 어떤 서비스가 Data Warehouse 를 구축하는데 적합한가? Data Warehouse 구축 후 운영 데이터의 ETL작업은 어떻게 해야하나?
- 대용량의 데이터를 저장하고 분석하는데 가장 쉽고 비용효율적인 방법은 무엇인가?
- Data governance 측면에서 다양한 종류의 데이터들을 어떻게 처리해야 하나?
- 서비스 모니터링 방법은?
- 데이터레이크를 구축하고 싶은데, 뭐부터 시작해야 하나?
- AWS에서 데이터 분석 플랫폼 구축 시 best practice는 무엇인가?
- 다양한 툴들을 사용하여 대용량 데이터를 어떻게 분석할 수 있는가?
- **AWS 데이터 서비스는 각각 이해가 되지만, Data Life Cycle 측면에서 어떻게 서비스들을 효과적으로 연계하고 통합할 수 있는지 알고 싶다.**

ANALYTICS PIPELINE



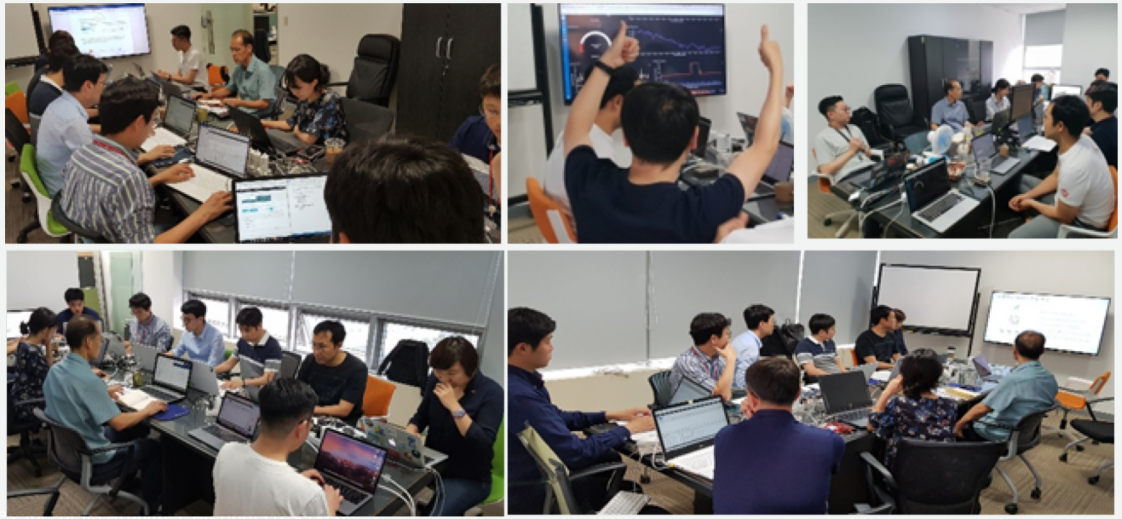
전형적인 분석 analytics pipeline은 다음 단계를 가짐

1. 데이터 수집
2. 데이터 저장
3. 데이터 처리 및 필요 시 ETL 수행
4. 데이터 분석 및 시각화

Local Data Lab의 진행 단계

타사 Local Data Lab 고객 Feedback

- 세션들이 실무적인 관점에서 실제 업무에서 어떻게 활용해야 하는지에 다뤄서 유용했다.
- 전반적인 기술적인 구성도 및 흐름을 설명해 주는 것이 와 닿았다.
- DW, BI를 17 년 동안 해 왔지만, DW에 대해 Cloud native 방식으로 더 공부해야 겠다는 생각이 들었다.



Technical Bootcamp

실제 e-commerce 데이터에 가까운 데이터 기반

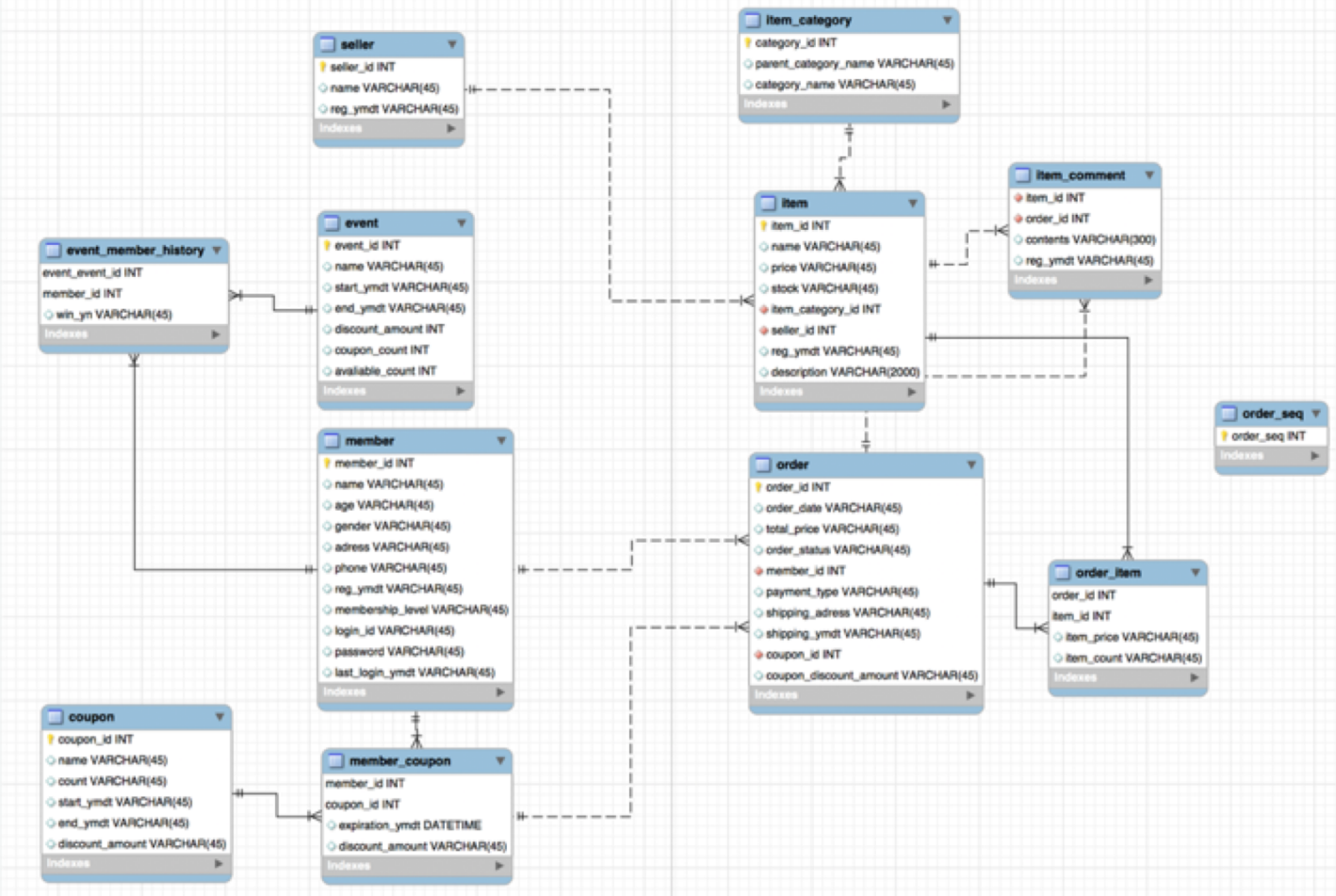
Results

	order_date	total_price	order_id	category_name	item_id	item_count	item_price	name	age	gender	city
1	20180322	1463	CA-2018-10284651	Machines	TEC-MA-10003626	3	411	Skye Norling	57	F	Peoria
2	20180322	1463	CA-2018-10284651	Storage	OFF-ST-10001328	3	16	Skye Norling	57	F	Peoria
3	20180322	1463	CA-2018-10284651	Phones	TEC-PH-10001468	1	182	Skye Norling	57	F	Peoria
4	20180322	330	CA-2018-10284652	Binders	OFF-BI-10002412	2	6	Kean Thornton	40	M	Denver
5	20180322	330	CA-2018-10284652	Labels	OFF-LA-10001771	3	5	Kean Thornton	40	M	Denver
6	20180322	330	CA-2018-10284652	Appliances	OFF-AP-10000938	3	101	Kean Thornton	40	M	Denver
7	20180322	1739	CA-2018-10284653	Copiers	TEC-CO-10001571	2	550	Dorris liebe	55	F	Seattle
8	20180322	1739	CA-2018-10284653	Phones	TEC-PH-10001835	3	158	Dorris liebe	55	F	Seattle
9	20180322	1739	CA-2018-10284653	Phones	TEC-PH-10004912	3	55	Dorris liebe	55	F	Seattle
10	20180322	8	CA-2018-10284654	Binders	OFF-BI-10004209	1	8	Alex Avila	46	F	San Francisco

Source : Tableau Community
https://community.tableau.com/docs/DOC-1236

ERD

RDB 대상 워크로드 생성을 위해 앞의 e-commerce 데이터 샘플 데이터를 정규화 함



가상 고객 요구사항

OLTP

- 최소한의 응답 시간을 요하는 RDB 기반의 기본 워크로드 (예 : 주문, 물품 조회 등)
- 가장 낮은 대기 시간의 NoSQL을 기반으로 예측할 수없고 확장 가능한 작업량 (예 : 프로모션 이벤트)
- 로그를 기반으로 한 실시간 스트리밍 데이터 (예 : 클릭 스트림)

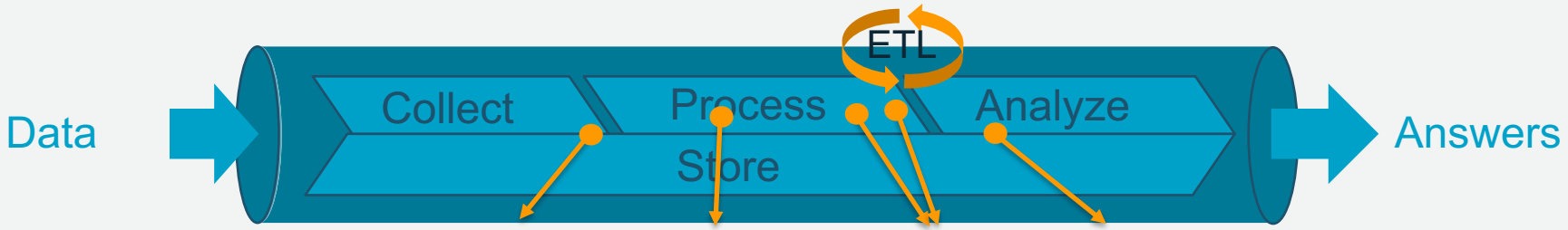
- Aurora
- DynamoDB
- Kinesis

Analytics

- 데이터 ETL 및 통합
- 다양한 스토리지 유형의 다양한 데이터를 처리하기위한 데이터 거버넌스
- 모던 데이터웨어하우스
- BI 용 단일 화면
- 로그 및 실시간 분석

- Glue ETL, Data Catalog
- Athena, Redshift w/Spectrum
- QuickSight
- ElasticSearch

ANALYTICS PIPELINE: AWS 서비스



데이터 수집 및 저장

- Amazon S3
- Amazon Kinesis
- Amazon Dynamo DB
- Amazon RDS (Aurora)

이벤트 처리

- AWS Lambda
- KCL Apps
- Amazon Kinesis Firehose

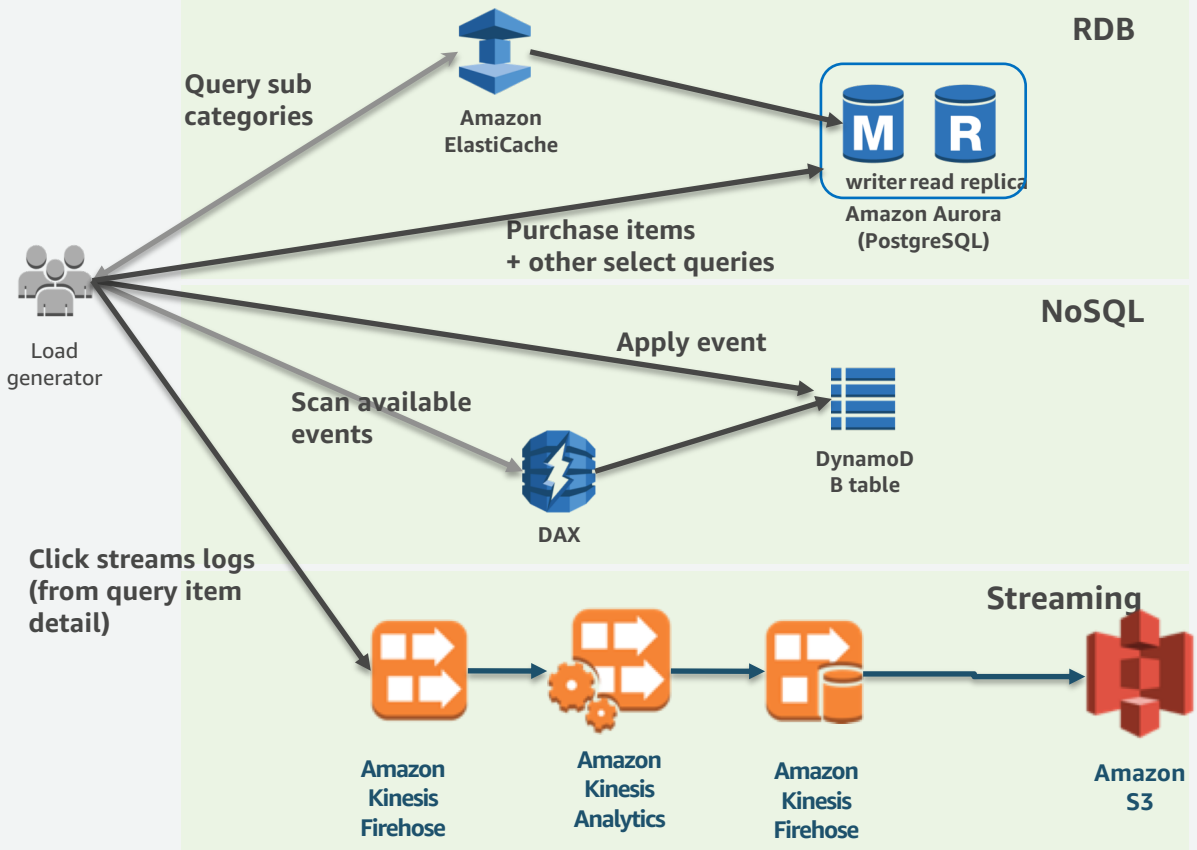
데이터 처리

- Amazon EMR
- Amazon Machine Learning
- AWS Glue

데이터 분석

- Amazon Redshift
- Amazon QuickSight

e-commerce 데이터를 기반으로 한 부하 생성기



[RDS-Aurora PostgreSQL]

- [SELECT] 상품 상세 조회
- [SELECT] 상품 리스트 조회
- [SELECT] 내 정보 조회
- [SELECT] 세부 카테고리 리스트 조회
- [INSERT] 구매
- [GET] 세부 카테고리 리스트 조회 (캐쉬이용) **(Elasticache Redis)**

[NoSQL-DynamoDB]

- [PUT] 이벤트 응모
- [SCAN] 오픈된 이벤트 리스트 조회
- [SCAN] 오픈된 이벤트 리스트 조회 **(DAX)**

[Streaming]

- 상품 상세 조회의 클릭스트림 로그 생성

RDB (Aurora) 데이터

order_item

order_id	item_id	item_price($)	item_count	order_date	order_time
CA-2018-12176056	FUR-CH-10001190	136	3	20180719	20180719023154
CA-2018-12176056	TEC-AC-10004353	63	3	20180719	20180719023154
CA-2018-12176057	OFF-BI-10001543	36	2	20180719	20180719023156
CA-2018-12176057	OFF-BI-10004253	16	3	20180719	20180719023156
CA-2018-12176057	FUR-FU-10003577	14	2	20180719	20180719023156

order

order_id	order_date	total_price ($)	order_ status	member_id
CA-2018-12176057	20180719	148	ordered	AA-10315

item

item_id	name	price($)	item_ category_ id
OFF-BI-10001543	GBC VeloBinder Manual Binding System	36	1

member

member_id	name	age	gender	country	postal_code	reg_ymdt
AA-10315	Alex Avila	46	F	United States	94122	20140113132700

aws

NoSQL (DynamoDB) 데이터

event_member_history

member_event_id	history_seq	reg_date
VP-21730_2	1	2018-07-19 02:33:12
TZ-21580_4	1	2018-07-19 02:33:10
TS-21655_2	1	2018-07-19 02:33:32
TS-21505_4	1	2018-07-19 02:33:23
TP-21415_3	1	2018-07-19 02:33:00

partition key = “member_id”_”event_id”
sort key = history_seq (지원 횟수)

event

event_id	available_count	available_yn	discount_amount	end_ymdt	name	start_ymdt
1	100	Y	10	20181201235959	new year event	20180101000000
2	100	Y	20	20181201235959	early bird event	20180101000000
3	100	Y	30	20181201235959	black friday event	20180101000000
4	100	Y	40	20181201235959	cyber monday event	20180101000000
5	100	N	50	20181201235959	thanks giving event	20180101000000

member

member_id	name	age	gender	country	postal_code	reg_ymdt
TZ-21580	Tracy Zic	28	F	United States	80027	20140807160743

실제 환경에서는, 로그인 후 세션 정보에 의해 member_id가 이미 식별되므로 부하 생성기는 전체 구성원 목록(파일) 중 member_id를 무작위로 선택합니다.

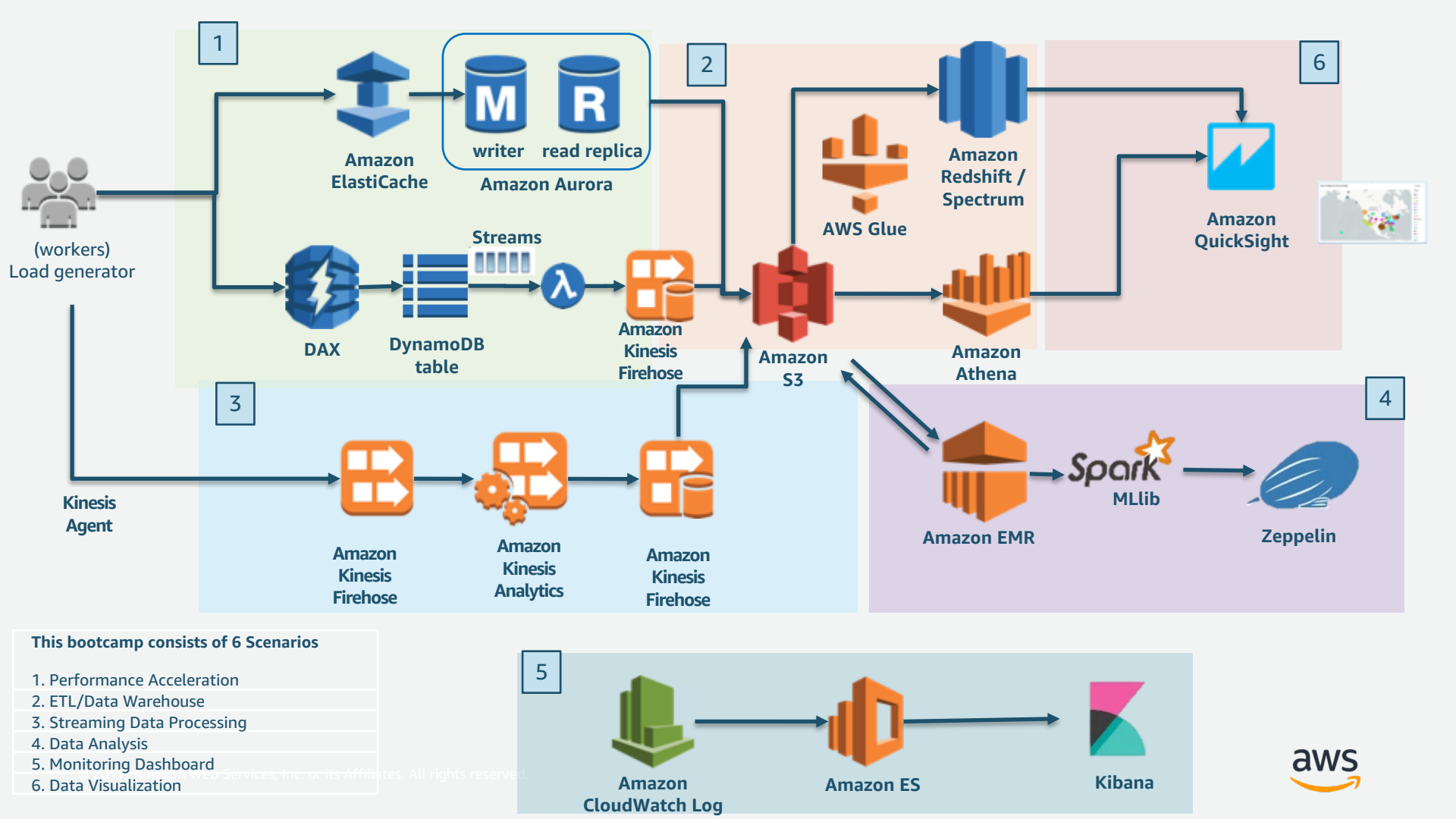
Streaming data 데이터

Streaming data –'특정 물품 상세 조회' 시마다 (1줄 씩) 생성

```
[ec2-user@ip-10-150-1-248 queryItemDetail]$ head 201807190233.log
2018-07-19 02:31:54$FUR-FU-10003919$Eldon Executive Woodline II Cherry Finish Desk Accessories$41$Execution time: 0.00251293182373 seconds
2018-07-19 02:31:54$FUR-BO-10001811$Atlantic Metals Mobile 5-Shelf Bookcases  Custom Colors$301$Execution time: 0.00313806533813 seconds
2018-07-19 02:31:54$OFF-PA-10004782$Xerox 228$6$Execution time: 0.0029[illegible]1800498962 seconds
2018-07-19 02:31:54$TEC-PH-10004959$Classic Ivory AntiqueÊTelephoneÊZL[illegible]1810$100$Execution time: 0.000820875167847 seconds
2018-07-19 02:31:54$OFF-AR-10002804$Faber Castell Col-Erase Pencils$5[illegible]Execution time: 0.00292205810547 seconds
2018-07-19 02:31:54$FUR-FU-10003192$Luxo Adjustable Task Clamp Lamp$8[illegible]$Execution time: 0.00215482711792 seconds
2018-07-19 02:31:54$TEC-AC-10004814$Logitech Illuminated Ultrathin Keyboard with Backlighting$57$Execution time: 0.00296306610107 seconds
```

로그 포맷 => **"Datetime of click" & "Item ID" & "Item name" & "Execution Time"**

Kinesis에서 실시간으로 이 로그를 기반으로 '인기 5 위 항목'을 찾을 수 있습니다.

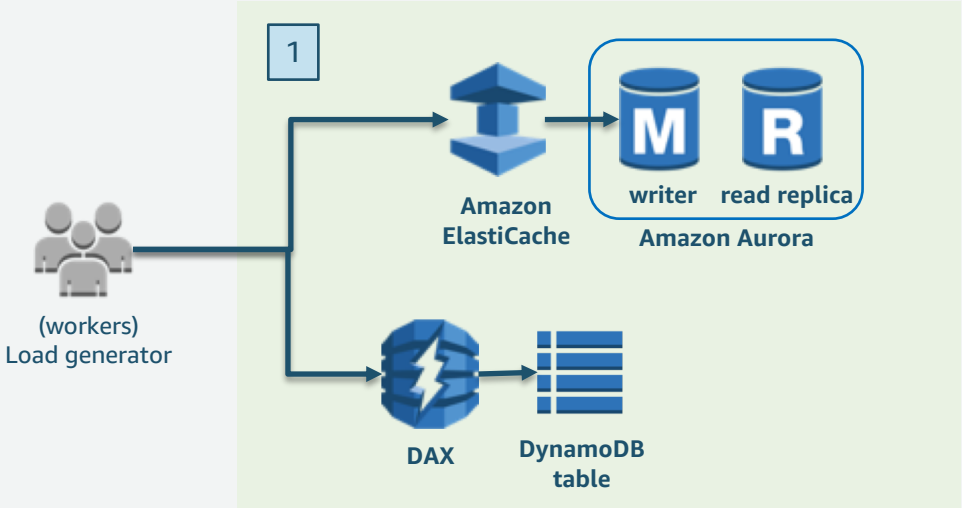


This bootcamp consists of 6 Scenarios

1. Performance Acceleration
2. ETL/Data Warehouse
3. Streaming Data Processing
4. Data Analysis
5. Monitoring Dashboard
6. Data Visualization

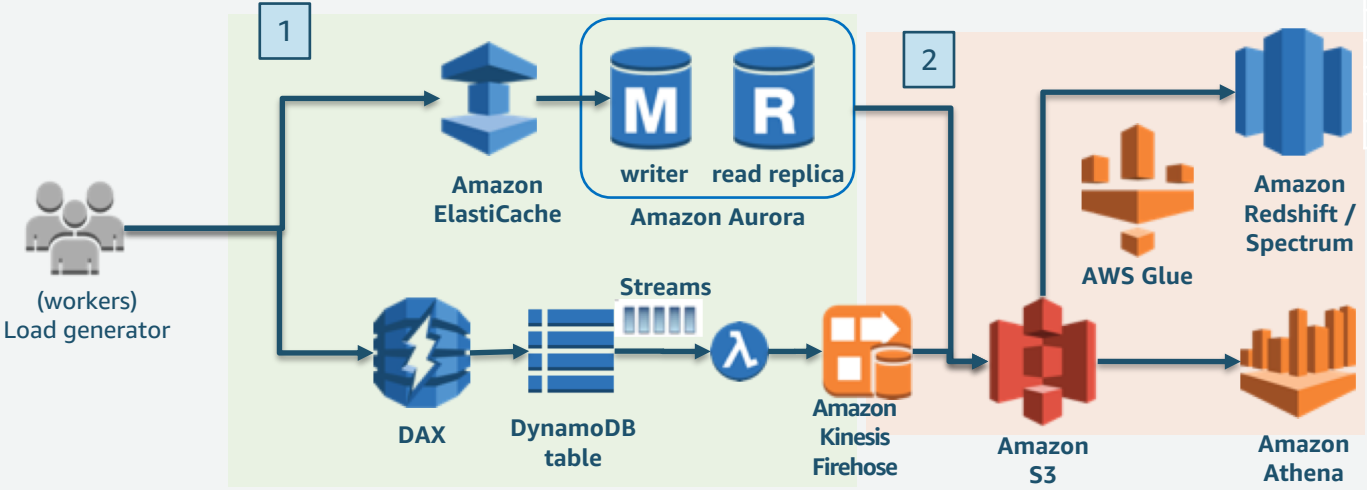
Scenario 1. Performance Acceleration

1. **Performance Acceleration**
2. ETL/Data Warehouse
3. Streaming Data Processing
4. Data Analysis
5. Monitoring Dashboard
6. Data Visualization



- You can **improve performance of Read Workload** when using OLTP data (RDB, NoSQL) through this lab.
- While generating workload against Aurora (RDB) and Dynamodb (NoSQL), you can check execution time and performance improvement by adding Cache Layer
  - ✓ Elasticache for RDB
  - ✓ DAX for DDB
- In this bootcamp, aurora cluster has only master instance for cost optimization ☺

Scenario 2. ETL / Data Warehouse

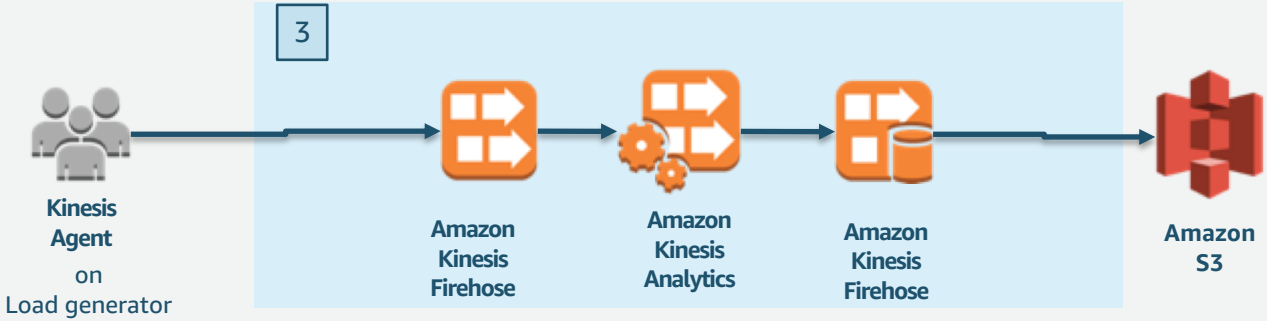


1. Performance Acceleration
2. **ETL/Data Warehouse**
3. Streaming Data Processing
4. Data Analysis
5. Monitoring Dashboard
6. Data Visualization

- You can copy raw data stored in RDB (Amazon Aurora) and NoSQL (DynamoDB) by service operation into S3 and transform data in S3 for analysis through this lab. Based on this analytic data in S3, you can build data warehouse with Athena, Redshift / Spectrum.
  - ✓ Glue ETL, a serverless ETL service **copy Aurora data to S3**.
  - ✓ DynamoDB streams, Lambda and Kinesis Firehose **copy DynamoDB data to S3**.
  - ✓ Glue can ETL **from raw data to analytic data in S3.**
  - ✓ Data prepared for analysis is generated through a Glue crawler, and **OLAP & analytic queries** is executed through Athena and Redshift / Spectrum.

Scenario 3. Streaming Data Processing

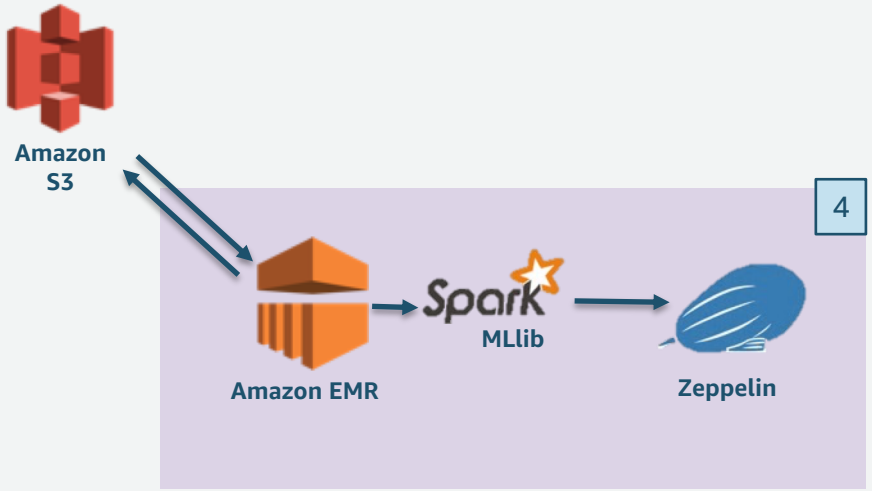
1. Performance Acceleration
2. ETL/Data Warehouse
3. Streaming Data Processing
4. Data Analysis
5. Monitoring Dashboard
6. Data Visualization



- Clickstream data is collected and analyzed in real-time using Kinesis Data Firehose and Kinesis Data Analytics. Collect and send stream data to Kinesis Data Firehose is via Kinesis agent and stored in S3.
  - ✓ Kinesis Data Firehose, **collect stream data to S3 in real-time.**
  - ✓ Kinesis Data Analytics, fastest way to analyze data in real-time and **collect top5 results**.

Scenario 4. Data Analysis

1. Performance Acceleration
2. ETL/Data Warehouse
3. Streaming Data Processing
4. Data Analysis
5. Monitoring Dashboard
6. Data Visualization



- You can **analyze the data stored in S3 Data Lake** using EMR on Spark.
- The **data catalog** created by Glue is compatible with Hive metastore of Hadoop, so you can easily load data into Spark.
- Zeppelin is a useful web notebook service for use with Spark. It can be installed easily by selecting it in check box when creating EMR clusters. This exercise will mainly use Zeppelin because it includes simple command execution and simple visualization.

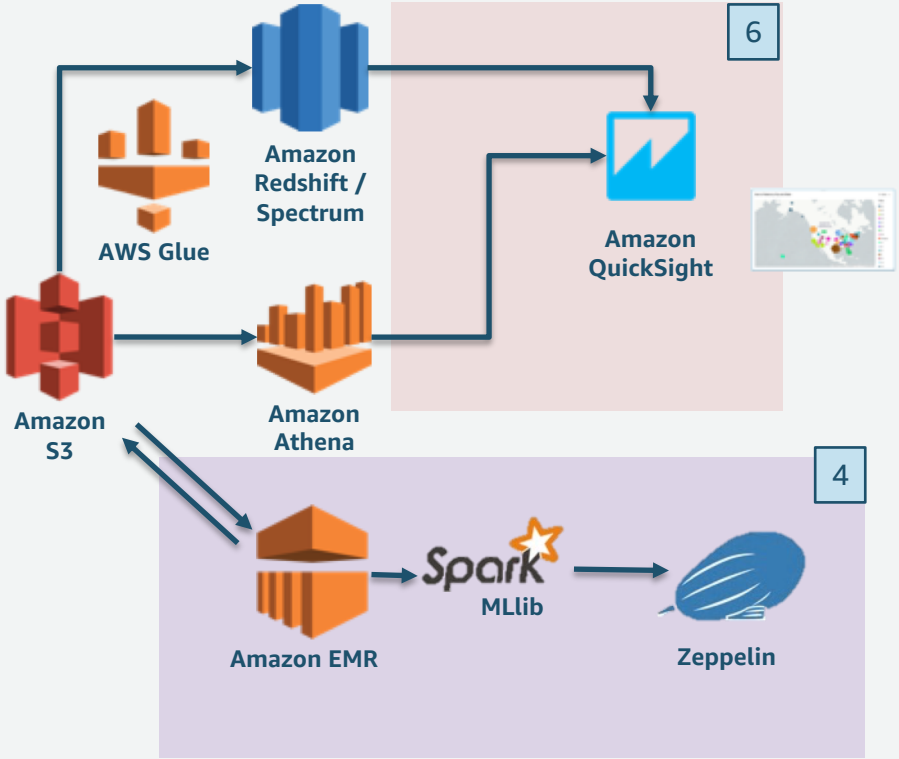
aws

Scenario 5. Monitoring Dashboard

1. Performance Acceleration
2. ETL/Data Warehouse
3. Streaming Data Processing
4. Data Analysis
5. **Monitoring Dashboard**
6. Data Visualization

- Elasticsearch indexes and visualizes metricsfor RDB (Amazon Aurora) collected in CloudWatch Logs. And simply configure RDS **monitoring dashboard** via Kibana.
  - ✓ RDS, metrics collected in CloudWatch Logs through enhanced monitoring.
  - ✓ Elasticsearch Service, index and analyze CloudWatch Logs.
  - ✓ Kibana, configure visualization dashboard to monitor RDS metrics.

Scenario 6. Data Visualization



1. Performance Acceleration
2. ETL/Data Warehouse
3. Streaming Data Processing
4. Data Analysis
5. Monitoring Dashboard
6. **Data Visualization**

- This Scenario is the step of **visualizing the analysis results of the data stored in Data Lake.**
- Various solutions can be selected according to the purpose of visualization. For Business Intelligence, you can use Quicksight to find Insights through a variety of approaches.
- Configure your Dashboard to use the Zeppelin web notebook to visualize, share, and analyze indicators.
- You can also use Zeppelin to run Hadoop Spark code and run ad-hoc queries via SQL.

Agenda –Technical Bootcamp – Day 1

09:30 ~ 10:10	Data Life Cycle on AWS (Data Lab 소개) (Lab 0. Cloudformation 수행)
10:20 ~ 10:40	Lab 0. Performance Acceleration for OLTP (Aurora, DynamoDB) with Cache layer
11:00 ~ 11:40	Data Lake on AWS
11:40 ~ 13:00	점심시간
13:00 ~ 13:20	실시간 스트리밍 데이터 처리
13:20 ~	Lab 1-1 : Kinesis Firehose for clickstream
~ 15:00	Lab 1-2 : Kinesis Analytics for real-time analytics
15:20 ~ 15:40	Data Engineering on AWS
15:40 ~	Lab 2-1 : RDS to S3 with Glue
	Lab 2-2 : DDB to S3 with DDB Streams, Lambda and Kinesis
	Lab 2-3 : Data Transforming with Glue
~ 18:00	Lab 2-4 : ETL to Redshift with Glue

Agenda –Technical Bootcamp – Day 2

09:30 ~ 10:00	Wrap up, Lab 3-1 : Glue Data Catalog for Analytics
10:20 ~ 11:00	Data Warehouse 구성
11:00 ~ 11:20	Lab 3-2 : Query with Redshift w/ spectrum and Athena
11:20 ~ 13:00	점심시간
13:00 ~ 13:40	Data Analytics on AWS
14:00 ~ 14:40	Lab 4. Exploratory Data Analysis with Spark and Zeppelin
15:00 ~ 15:30	Log analysis with Elasticsearch
15:30 ~ 16:00	Lab 5. CloudWatch Logs to ES and Kibana
16:20 ~ 17:00	Lab 6. Visualization with QuickSight

aws

예) Data 분석 결과들

spectrum vs athena | New query 2 | New query 3 | spectrum vs athena

```
select m.city, sum(oi.item_price * oi.item_price)
from "fact_order_json" oi, "item" i, "item_category" it, "member" m
where oi.item_id = i.item_id
and i.item_category_id = it.category_id
and oi.member_id = m.member_id
and oi.order_date > '20180301'
and i.name like '%Apple%'
group by m.city
order by 2 desc;
```

Run query | Save as | Format query | New query | (Run time: 5.85 seconds, Data scann

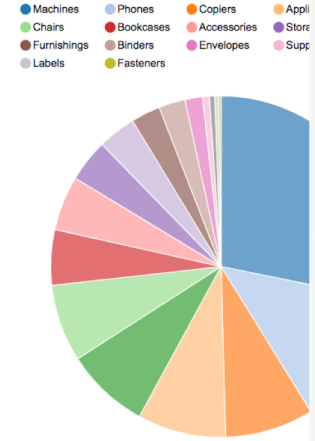
Results

	city
1	San Francisco
2	New York City
3	Elmhurst
4	Henderson
5	Garden City
6	Denver
7	Columbus

어떤 상품 카테고리의 매출이 높은지 확인 - Machines, Phones, Copiers 순서로 매출이 나타남 FINISHED

```
%sql
select item_category.category_name, sum(order_item.item_count
    *order_item.item_price) sales
from order_item, item, item_category
where order_item.item_id = item.item_id
and item.item_category_id = item_category.category_id
group by item_category.category_name
order by sales desc
```

settings

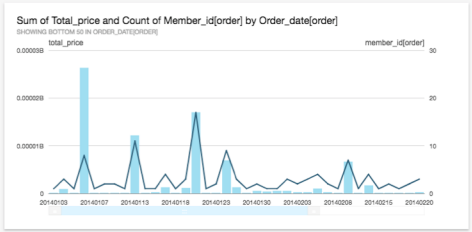


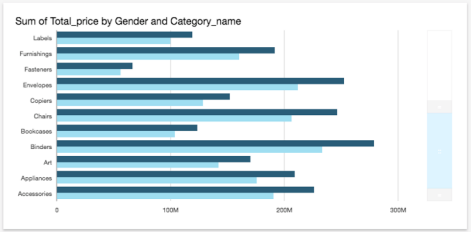
Took 22 sec. Last updated by anonymous at May 03 2018, 1:49:32 PM. (outd

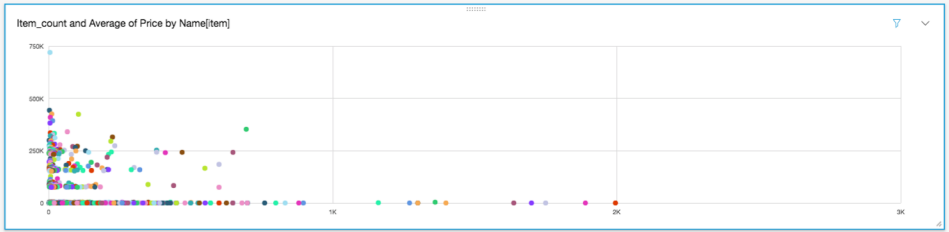
카테고리 매출 비중을 성별에 따라 분류해 보자 FINISHED

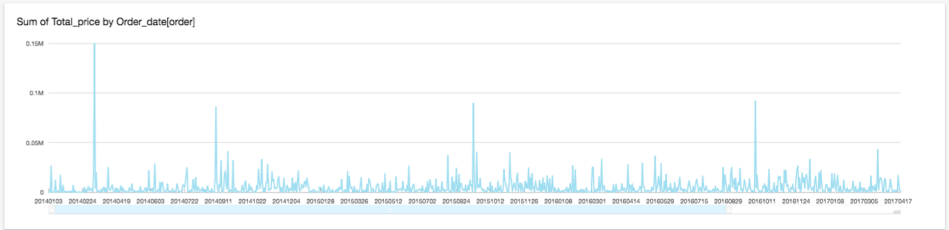
```
%sql
select item_category.category_name, member.gender, sum(order_item
    .item_count*order_item.item_price) sales
from order, order_item, item, item_category, member
where order_item.item_id = item.item_id
and item.item_category_id = item_category.category_id
and order.order_id = order_item.order_id
and order.member_id = member.member_id
group by item_category.category_name, member.gender
order by sales desc
```

settings









감사합니다