

# Databases

일시

2023.05.06. (토)

작성자

김민경

## 1. Database Types

RDBMS	-SQL을 사용하거나 OLTP 처리시 사용 -RDS, Aurora -조인에 유용함
NoSQL DB	-유연한 DB -조인 기능 x -일반적으로 SQL 쿼리 언어 사용 x -DynamoDB, ElastiCache, Neptune
Object Store (객체 스토어)	-S3(대용량 객체 저장용), Glacier(백업, 아카이브 용)
Data Warehouse	-SQL 분석, BI(비즈니스 인텔리전스) 사용
Search (검색 유형의 DB)	-OpenSearch: 자유롭게 텍스트 입력 or 비정형 데이터를 검색
Graphs	-Neptune: 데이터 세트 간의 관계를 표시
Ledger (원장 DB)	-트랜잭션 목록 & 원장 기록 가능 -퀀텀 원장 DB
Time Series (시계열 DB)	-Amazon Timestream

### \*OLTP (온라인 트랜잭션)

:네트워크상의 여러 이용자가 실시간으로 데이터베이스의 데이터를 갱신하거나 조회하는 등의 단위작업을 처리하는 방식,  
ATM 인출과 신용카드 결제, 표 예매 및 예약 시스템, 온라인 쇼핑, 일반 전자상거래를 위한 실시간 데이터 처리  
ex) 카드, 은행 거래에서 일어나는 트랜잭션 단위의 활동

### \*트랜잭션

:DB의 상태를 변화시키기 위해 수행하는 작업의 단위(SELECT, INSERT, DELETE, UPDATE)

### \*조인

:두 개 이상의 테이블이나 데이터베이스를 연결하여 데이터를 검색하는 방법

### \*아카이브

:사진, 동영상, 문서 등 볼트에 저장되는 모든 객체

### \*BI

:조직이 좀 더 데이터 기반의 의사 결정을 하도록 지원하는 비즈니스 분석, 데이터 마이닝, 데이터 시각화, 데이터 도구,  
인프라, 모범 사례가 모두 포함

## 2. RDS

• DB 엔진 유형: 관리형 PostgreSQL, MySQL, Oracle, SQL서버, MariaDB 용 사용자 지정 RDS

• RDS 인스턴스 크기 & EBS 볼륨 유형 및 크기 프로비저닝해야함

• 읽기 전용 복제본 지원 (읽기 용량 확장 위해)

• 고가용성 목적으로 대기 DB를 다중 AZ에 둘 수 있음  
(대기 DB: 재해 발생 대비용, 쿼리 수행x)

• 보안

: IAM, SG, KMS(저장 데이터 암호화), SSL/TLS(저장 데이터 암호화)

• 자동 백업 옵션

: 최대 35일(지정시간 복구 기능 지원)

• 장기 보존 백업 필요시⇒ 수동 DB 스냅샷 사용

• 유지 관리 기능 예약 가능→ DB 다운타임 발생 가능성

(DB 엔진 업데이트, 프로비저닝할 때, EC2 인스턴스에 보안 패치 실행시)

• 프록시 강제해 IAM 인증 추가 가능, Secret Manager와 통합해 DB 자격 증명 관리 가능

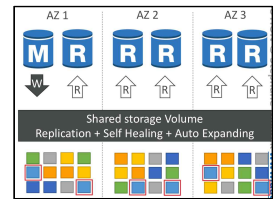
### \*SSL

:보안 소켓 계층,  
두 서버 사이 전달되는 데이터 암호화

### \*TLS

::전송계층 보안, SSL보다 더 안전

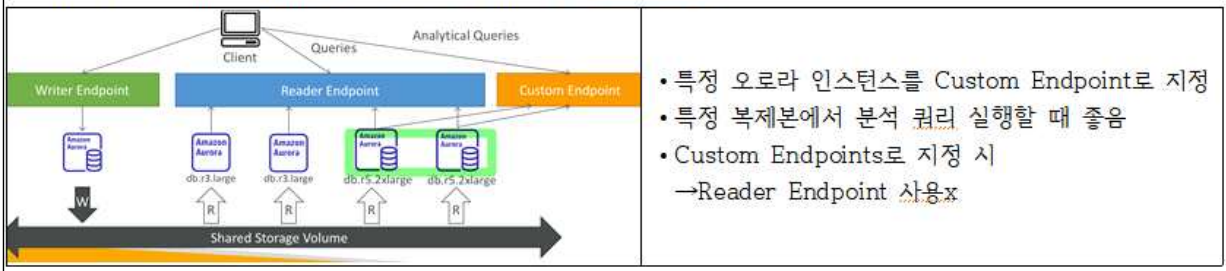
- Oracle, SQL 서버 유형의 DB에서 지원됨
- Use case: 관계형 DB 저장시(RDBMS & OLTP 처리), SQL 쿼리 실행, 트랜잭션에 활용



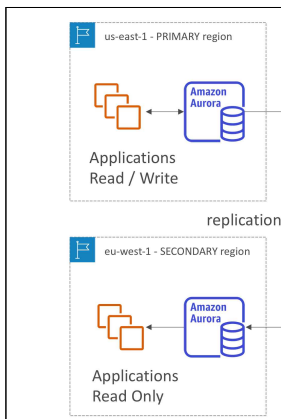
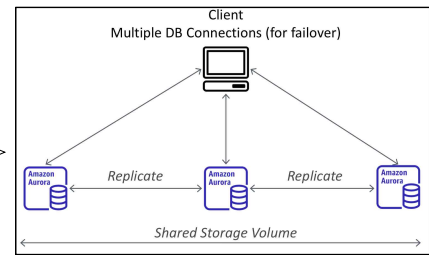
### 3. Aurora

- DB 엔진인 PostgreSQL, MySQL과 호환되는 API
- 컴퓨팅 & 스토리지가 분리된 특별한 서비스
- 스토리지: 데이터를 3 AZ에 걸쳐 6개 복제본 저장(default), 문제 발생 시 자가 복구 처리, 오토 스케일링 내장
- 클러스터: write & reader DB 인스턴스를 위해 사용자 지정 엔드포인트 필요  
(클러스터⇒ 클라이언트 있을 때 많은 인스턴스와 어떻게 접속하는지)

#### 5) Aurora Custom Endpoints (Aurora 사용자 지정 엔드포인트)



- 보안, 모니터링, 유지관리 기능 = RDS
- Aurora Serverless  
: 워크로드가 간헐적 or 예측x⇒ 용량 계획 x아도 되므로 유용
- Aurora Multi-Master (다중 마스터)  
: DB 인스턴스 여러 개가 스토리지에 쓰기할 수 있도록 설정 가능 ⇒
- Aurora Global  
: 글로벌 DB 원할 시 사용,  
DB가 복제된 리전마다 최대 16개 DB 읽기 전용 인스턴스 제공, 리전 간 스토리지 복제 1s 미만,  
기존 리전에 문제 발생 시 보조 리전을 새 리전으로 승격 가능



- all 읽기 및 쓰기가 발생하는 하나의 기본 리전 존재
- 복제 지연이 1s 미만인 보조 읽기 전용 리전을 5개까지 설정 가능
- 각 보조 지역마다 읽기 전용 복제본을 16개까지 생성 가능
- 한 리전 DB 장애 발생  
→복구 시간 목표(RTO) 1m 미만
- 리전에 걸쳐 데이터 복제⇒ 1s 미만

#### Aurora Machine Learning

: SageMaker & Comprehend 사용해 ML 사용 가능

**\*SageMaker**

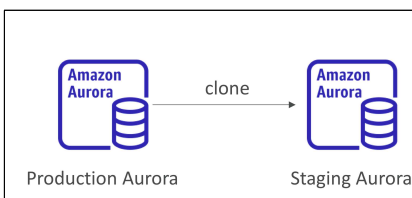
:기계 학습 서비스

**\*Comprehend**

:자연어 처리 서비스

#### Aurora DB Cloning

: 기존 클러스터에서 새 클러스터 만드는 것, 스냅샷 사용해 복구하는 것보다 fast,



- Production Aurora를 새로운 Aurora DB로 복제  
ex\_ Production Aurora에 영향 주지 x고,  
Staging Aurora로 복제해 개발 or 테스트 수행 o

#### Use case

- RDS와 같지만 Aurora는 유지 관리 小, 유연성↑, 성능↑, 내장된 기능 多

## 4. ElastiCache

- 관리형 Redis/Memcached (캐시)
- RDS와 비슷한 기능 제공하지만 캐싱 작업에 활용됨
- In-memory data store(데이터 읽을 때 1밀리초 미만 지연h)
- EC2 인스턴스 유형 프로비저닝 필수
- Redis 용에서는 클러스터 생성, 다중 AZ와 샤딩을 위한 읽기 전용 복제본 사용 가능

클러스터는 하나 이상의 캐시 노드 모음으로서, 이 모든 노드는 Redis 캐시 엔진 소프트웨어의 인스턴스 하나를 실행합니다. 클러스터를 만들 때 모든 노드에서 사용할 엔진과 버전을 지정합니다.

### \*샤딩

:대규모 DB를 빠르고 쉽게 관리할 수 있도록 작게 나누는 것

- 보안: IAM, SG, KMS, Redis Auth 사용

### -Redis AUTH

:Redis 클러스터 생성 시 비밀번호 or 토큰 설정 (캐시에 들어갈 때 비번 사용).  
전송 중 암호화 위해 SSL 보안 지원 0

- 백업, 스냅샷 지정, 지정 시간 복구 기능 0 (=RDS)
- 관리형 및 예약된 유지 관리 가능
- RDS와 결합된 DB에서 캐싱 작업을 수행하려면 app 코드가 Elasticache 활용하도록 코드 수정해야 함
- SQL 사용 x
- Use case

-키-값 스토어 있고, read 빈번, write 小 시⇒ DB 쿼리를 캐싱하는 것이 좋음

-웹사이트 사용자 위해 세션 데이터 저장

### \*세션

:웹 사이트의 여러 페이지에 걸쳐 사용되는 사용자 정보 저장

## 5. DynamoDB

- AWS 독점 기술, 관리형 서버리스 NoSQL DB, 밀리초 수준의 지연시간 제공
- 용량모드

### ① 프로비저닝 용량모드

-선택형 오토 스케일링 탑재

-이중 워크로드 있을 때(워크로드 ↑ or ↓), 로드 예측 가능 시

### ② 온디맨드 용량모드

-용량 프로비저닝x

-오토스케일링 실행 → 워크로드 예측 diff or 수요 급증 시 유용

- ElastiCache 대신 키-값 저장 가능

(웹사이트 세션 데이터 저장하는 좋은 방법, TTL 가능과 결합하면 일정 h 후에 행이 만료되게 설정 가능)

- (default) HA, 다중 AZ에 걸쳐있음
- read & write 완전 분리, 트랜잭션 처리 가능
- DAX (DynamoDB와 호환되는 읽기 캐시) ⇒ 읽기 지연h 백만분의 1s

-DynamoDB 응답 시간⇒한 자릿수 밀리초 단위

DAX⇒ 캐시 데이터에 마이크로초 단위의 응답 시간이 필요할 때

### \*TTL

:Time To Live, 캐시가 폐기되기 전까지의 시간

- 보안, 인증, 권한 부여⇒ IAM 통해 처리됨
- 이벤트 처리 용량
- DynamoDB Streams로 DB의 all 변경사항 스트리밍 가능,  
람다와 통합 가능(DynamoDB 테이블에서 변경 발생시 람다 실행)
- Kinesis data streams로 전송 가능

- Global Table 기능

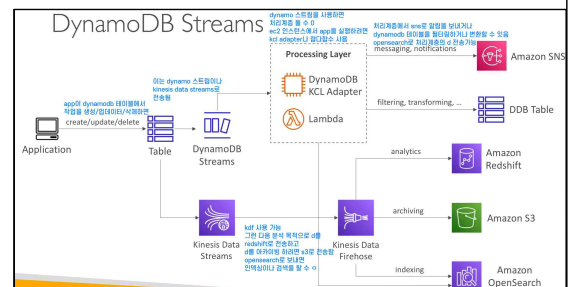
: 여러 리전 간 다중 활성 복제 가능

- 백업 옵션

① 자동 백업 : PITR(지정시간 복구) 활성 시⇒ 테이블 복구 가능, 35일까지 가능

### \*다중 활성

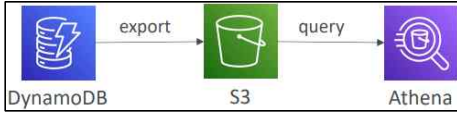
:어느 리전에서든 읽고 쓸 수 있음



② 온디맨드 백업 : 백업 보유기간 ↑시 사용 (복구 진행 시 새로운 테이블 생성)

- PITR 내에 읽기 용량 단위 사용x고 DynamoDB 테이블을 S3로 내보낼 수 있음

(35일 이내에 S3로 내보내는 기능 사용해서 쓰기 용량 단위 사용하지 x고 S3에서 새 테이블로 불러올 수 있음)



- 스키마 빠르게 전개할 때 사용
- SQL 쿼리 언어 사용x
- Use Case

-400kb 미만 문서 다루는 작은 서버리스 app 개발 서버리스 캐시 분산

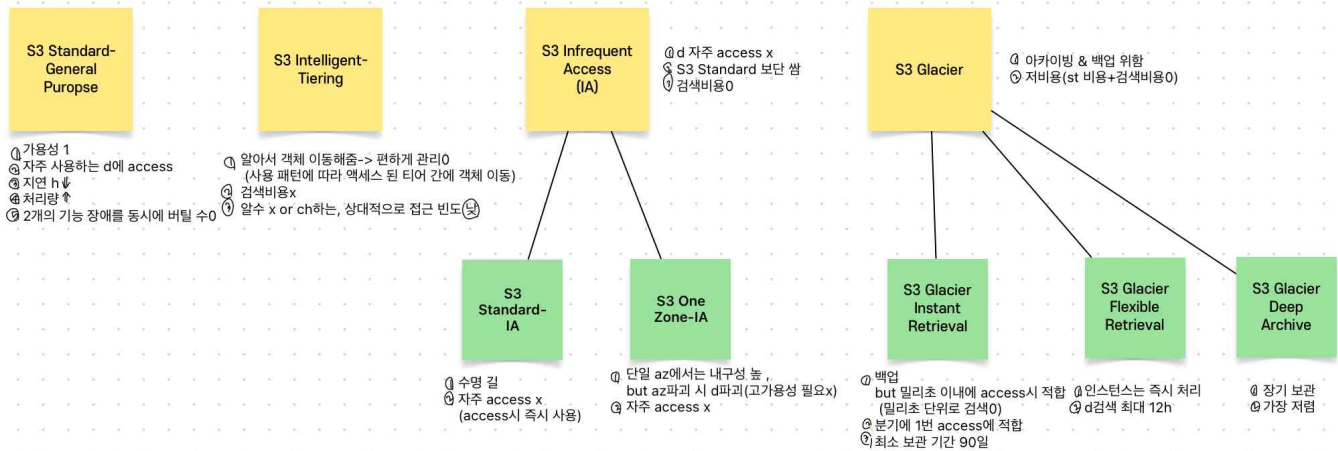
## 6. S3

### \*버저닝

:특정 버킷의 객체들에 관해 버전 관리 기능, 복원 가능

- 키-값 ⇒ 큰 객체 저장 시 유용 (작은 객체 저장 시 유용x)
- 서버리스, 확장성 무한, 객체 최대 크기 5TB, 시간 경과 따라 계속 버저닝함
- Tiers(계층) + 수명주기 정책

## [S3 Storage Class]



- 버저닝, 암호화, 복제, MFA-삭제, 액세스 로그

### \*MFA-삭제

:실수로 버킷 삭제하는 것 방지 가능

- 보안

: IAM, 버킷 정책, ACL, S3에 액세스 포인트 생성, S3 Object Lambda 사용해 객체를 app에 전송 가능, CORS, Glacier의 객체 잠금 or 볼트 잠금 개념 활용 가능

### \*ACL(Access Contrl List)

:접근 제어 목록, 개체나 개체 속성에 적용되어 있는 허가 목록

### \*액세스 포인트

:S3 액세스 포인트를 사용하면 고객이 각 액세스 포인트의 고유한 액세스 제어 정책을 생성하여 공유 데이터 세트에 대한 액세스를 손쉽게 제어, 버킷 정책 대신 사용하는 것(버킷 정책 너무 많을 때 액세스 포인트 사용)

### \*S3 Object Lambda

:S3 버킷의 객체를 호출한 app이 회수하기 전에 수정하기 위함, 원래 주 버킷에서 일부 내용만 보이게 데이터 편집해 연결

### \*CORS

:Cross-Origin Resource Sharing, 출처가 다른 자원들 공유하는 것

### \*S3 Glacier Vault Lock

:S3 Glacier 저장소에 대한 규정 준수 제어

- 암호화

: SSE-S3, SSE-KMS(자체 키 가져옴), SSE-C, 클라이언트 측 암호화, 전송 중 암호화, S3 버킷 암호화

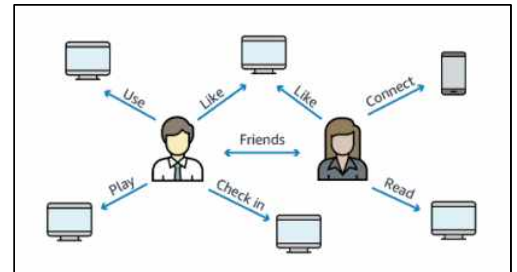
- 배치 작업
  - S3 버킷에 있는 all 파일 한 번에 작업 시 유용(대량 작업 수행)
  - 기존 s3 버킷의 비암호화 객체를 암호화할 때, s3 복제 활성화하기 전 기존 파일을 다른 버킷으로 복사할 때
- 성능
  - 멀티 파트 업로드(파일 쪼개서 병렬로 업로드)
  - S3 Transfer Acceleration(전송 가속화, 다른 리전으로 빠르게 전송)
  - S3 Select(S3에서 필요 데이터만 검색)
- 자동화
  - S3 Event Notifications(S3 이벤트 알림 기능)
    - ⇒S3 버킷에서 특정 이벤트가 발생할 때 알림을 받음(SNS, SQS, Lambda, EventBridge)
- Use cases
  - 정적 파일, 큰 파일의 키-값 스토어, 웹사이트 호스팅

## 7. DocumentDB

- 빠르고 안전하고 완전 관리형 데이터베이스 서비스
- Aurora로 PostgreSQL과 MySQL 대규모 클라우드 네이티브 버전을 구현하는 것과 비슷
- **몽고DB용 오로라** (클라우드에서 MongoDB 호환 데이터베이스를 쉽게 설치, 운영 및 조정)
  - 몽고DB ⇒ NoSQL DB
- JSON 데이터 저장, 쿼리, 인덱스 하는데 사용
- HA, 데이터 3 AZ에 복제
- 스토리지⇒ 10GB 단위로 최대 64TB까지 ↑
- 초당 수백만 건의 요청이 있는 워크로드에 확장될 수 있게 설계됨

## 8. Amazon Neptune

- 완전관리형 **그래프** DB
- **그래프 데이터 셋 ex) 소셜 네트워크**
  - 사용자는 친구가, 게시물에는 댓글이, 댓글에는 좋아요가 있는 것처럼 모든 것이 서로 연결되어 있고 그래프를 만듦
- 3 AZ에 걸쳐 최대 15개의 읽기 전용 복제본으로 복제함
- 소셜 네트워크 같이 고도로 연결된 데이터 셋 사용하는 app 실행시 사용됨, 그래프 데이터 셋에서 복잡하고 어려운 쿼리 실행하기에 최적화
- 수십억개의 관계 저장 가능, 그래프 쿼리시 지연h 밀리초



## 9. Amazon Keyspaces

- **Amazon Keyspaces와 관련 있는 것**
- 고가용성의 확장 가능한 관리형 Apache Cassandra 호환 데이터베이스 서비스 (Cassandra: 오픈소스 NoSQL 분산 DB)
- 서버리스, 확장성, HA, 완전관리형
- app 트래픽 따라 테이블 자동 확장/축소
- 테이블 데이터⇒ 여러 AZ에 걸쳐 3번 복제
- Keyspaces에서 쿼리 수행시 Cassandra 쿼리 언어(CQL) 사용하기
- 어떤 규모에서도 지연시간 10밀리초 미만, 초당 수천건 요청 처리
- 용량: 온디맨드 모드, 프로비저닝 모드
- 암호화, 백업 기능 제공, PITR 최대 35일
- Use cases: IoT 장치 정보, 시계열 데이터 저장 등

## 10. Amazon QLDB

- Quantum Ledger DB(퀀텀 레저 DB) 약자
- 금융 트랜잭션 or 원장 나눌 때 사용  
(원장⇒ 금융 트랜잭션 기록하는 장부)
- 완전관리형 원장 데이터베이스로,  
투명하고, 변경 불가능하며, 암호화 방식으로 검증 가능한 트랜잭션 로그를 제공하는 것
- 완전 관리, 서버리스, HA, 3 AZ에 걸쳐 데이터 복제 가능
- app 데이터의 h에 따른 all 변경 내역 검토시 사용 (∴장부라고 함)
- 불변 시스템 (Immutable system)  
-DB에 쓰면 삭제 & 수정 불가
- 블록체인 프레임워크보다 2-3배 더 나은 성능 얻을 수 있음. sql 사용해 데이터 관리 가능
- 탈중앙화 개념x (아마존 소유의 중앙 DB에서만 저널 작성 가능=중앙 권한 구성 요소)  
⇔ Amazon Managed Blockchain

### \*블록체인

:비즈니스 네트워크 내에서 정보를 투명하게 공유할 수 있도록 하는 고급 데이터 베이스 메커니즘

## 11. Amazon Timestream

- 시계열 DB  
(시계열: 시간 정보를 포함하는 포인트의 모음)
- 완전관리형, 빠름, 확장성, 서버리스
- DB 용량 자동 확장, 축소 가능
- 매일 수조 건의 이벤트 저장 및 분석 가능
- 시계열 데이터 있을 시⇒ 관계형 DB보다 시계열 DB 사용이 빠르고 저렴함
- 쿼리 예약하고 다중 척도 레코드 얻을 수 있음, SQL과도 완벽히 호환됨
- 최신 데이터⇒ 메모리에 저장,  
과거 데이터⇒ 비용 효율적인 스토리지 계층에 저장
- 시계열 분석 기능 있음→ 거의 실시간으로 데이터 분석 & 패턴 찾을 수 있음
- 전송 중 데이터 & 저장 데이터 암호화 지원
- Use case: IoT app, 운영 app, 실시간 분석 등

