



[AWS] 5. 데이터베이스 서비스

• References

- AWS 공인 솔루션 아키텍트 스터디 가이드 - 어소시에이트 3/e - 5장
- <https://inpa.tistory.com/entry/AWS-RDS-개념-아키텍처-정리-이론편?category=947445>
- <https://velog.io/@songa29/AWS-DynamoDB란> (DynamoDB 란)
- https://docs.aws.amazon.com/ko_kr/amazondynamodb/latest/developerguide/SecondaryIndexes.html (DynamoDB의 보조인덱스)

• 관계형 데이터베이스

- https://inpa.tistory.com/entry/DB-데이터베이스-기초-개념?category=890807#관계형_데이터베이스의_종류 (참고)

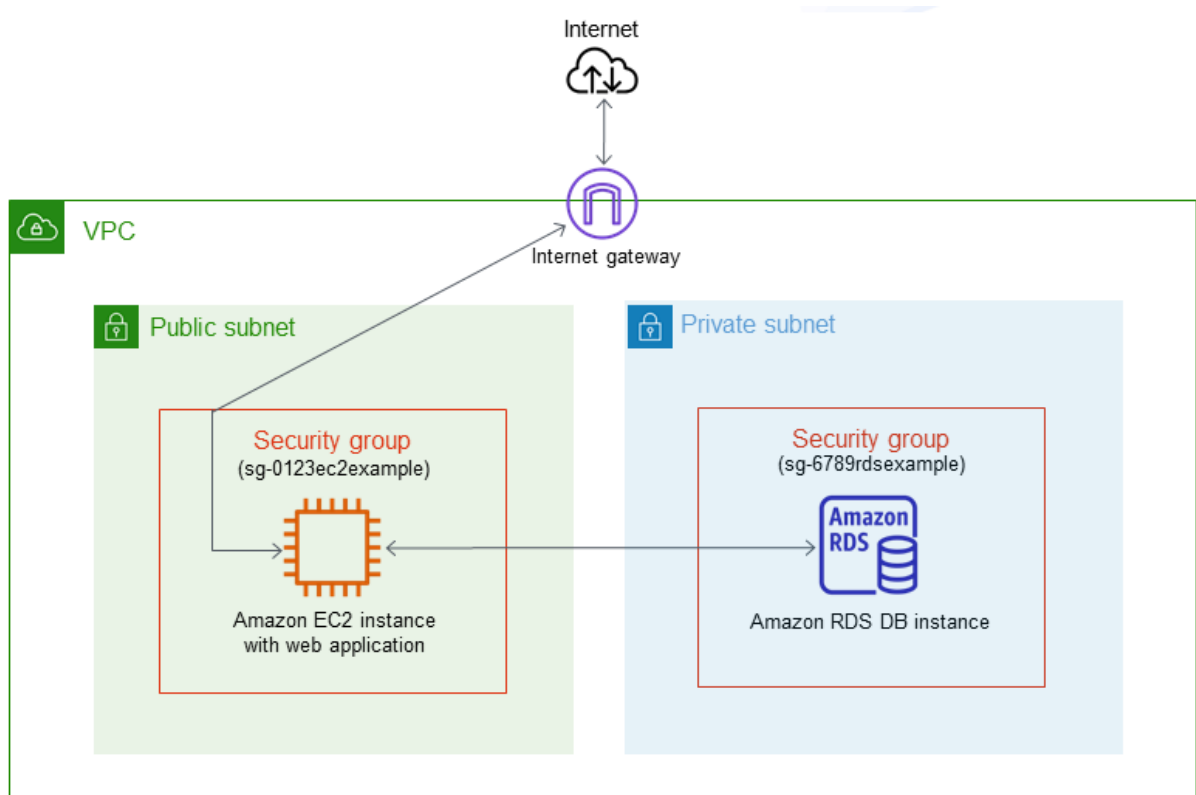
• OLTP 온라인 입출력 처리 vs OLAP 온라인 분석 처리

| OLTP (Online Transaction Processing) | OLAP (Online Analytic Processing) |
|---|--|
| 빈번한 데이터 읽기 및 쓰기 작업이 요구되는 애플리케이션에 적합 | 대규모 데이터에 대한 복잡한 쿼리 작업에 적합 |
| 비교적 정형화 된 쿼리를 주로 사용 | 복잡한 쿼리를 주로 사용 |
| 충분한 수준의 메모리 및 컴퓨터 용량을 지닌 단일 서버를 통해 모든 쓰기 및 읽기 작업 수행 | 대규모 데이터베이스의 경우 파티셔닝과 샤딩 작업을 통해 각 서버가 처리해야 할 부분만 맡아서 처리 |
| 분당 수백건 주문을 처리하는 온라인 주문 시스템 등에 주로 사용 | 빈번하지는 않지만 정기적으로 데이터에 접근하는 인사관리 시스템 등에 사용한다. |

• Amazon RDS

- Amazon Relational Database Service (RDS) 는 관리형 데이터베이스 서비스로서 클라우드 기반 관계형 데이터베이스

- 데이터베이스 인스턴스를 생성하여 구성, EC2와 달리 SSH 접속 불가능 하며 EC2 인스턴스에서 바로 접근할 수 없다. (기본적으로 Public IP를 부여하지 않음) → 다만 설정에 따라 public 으로 오픈이 가능하며 로드밸런서 같이 DNS로만 접근이 가능
- 데이터베이스 설치, 모니터링, 백업, 알람 등의 관리를 대신해주며, 하드웨어 프로비저닝, 데이터베이스 설정, 패치 및 백업 과 같은 잦은 운영작업을 자동화 하여 비용효율적이고 크기조정 가능한 DB 서비스를 제공
- EC2와 마찬가지로 인스턴스 타입을 지정해 줘야 하며 스토리지는 EBS를 그대로 활용하기 때문에 EBS 타입 선택이 필요
- RDS 는 기본적으로 VPC 내에서 동작하는 서비스(인스턴스) 이기 때문에 서브넷과 보안그룹 지정이 필요하다
- DB의 설정값들을 모아 그룹화한 파라미터 그룹시스템을 갖고 있다. (인스턴스 수정이 불가능 하기에 우회적으로 설정을 변경 가능)



• RDS 의 특징 및 기능 개요

| | |
|-----------|---|
| 관리 부담 감소 | 사용 편의성, 자동 소프트웨어 패치, 모범 사례 권장 시스템 |
| 확장성 | 즉각적인 컴퓨팅 규모 조정, 간편한 스토리지 규모 조정, *읽기 전용 복제본 시스템 |
| 가용성 및 내구성 | 자동백업, 데이터베이스 스냅샷, *다중 AZ구성 배포 , 자동 호스트 교체 |
| 보안 | 저장 데이터 및 전송데이터 암호화, 네트워크 격리, IAM 리소스 수준 권한 |
| 관리 효율성 | 모니터링 및 지표, 이벤트 알림 |
| 비용 효율성 | 사용한 만큼만 비용 지불, 예약 인스턴스, 인스턴스 중지 및 시작 |

- **관리부담 감소**

- 사용 편의성 : 몇 분 이내 인스턴스를 실행하고 애플리케이션을 연결 가능하다. 파라미터 그룹으로 쉽게 설정
- 자동 소프트웨어 패치 : 소프트웨어가 최신 상태로 유지되도록 한다. 패치 시기를 선택적으로 제어 가능
- 모범 사례 권장 시스템 : 인스턴스의 구성과 사용 지표 분석으로 모범 사례 지침을 제공, 고객은 해당 권장사항을 수행하거나 예약, 무시가 가능하다.

- **확장성**

- 즉각적인 컴퓨팅 규모 조정 : 배포에 사용할 컴퓨팅 (최대 vCPU32) 및 메모리(최대 244G) 리소스를 확장 축소 가능
- 간편한 스토리지 규모 조정 : 스토리지 요구 증가에 따라 추가 스토리지를 프로비저닝 DB 가동 중단 없이 스토리지 확장 가능
- 읽기 전용 복제본 시스템 : 인스턴스의 복제본을 하나 이상 생성하여 대량의 애플리케이션 읽기 트래픽을 처리할 수 있는 기능 제공

- **가용성 및 내구성**

- 자동 백업 : RDS는 데이터베이스와 트랜잭션 로그를 백업하고 사용자가 지정한 보존 기간 동안 이를 모두 저장할 수 있다. 특정 시점으로 복원 가능 (최근 5분 전까지 가능) 자동 백업 보존기간은 최대 35일로 구성 가능하다.
- 데이터베이스 스냅샷 : 사용자는 원하는 경우 언제든지 데이터베이스 스냅샷으로 새 RDB 인스턴스를 생성할 수 있다.
- 다중 AZ 구성 배포 : 다중 AZ 구성은 현재 서비스 되는 RDB에 문제가 되어도 다른 AZ에 있는 RDB로 빠르게 장애 복구 할 수 있다. 가용성과 내구성을 높여주므로 프로덕션 데이터베이스 워크로드에 적합하다.
- 자동 호스트 교체 : 하드웨어에 장애 발생 시, 배포를 지원하는 컴퓨팅 인스턴스를 자동으로 교체한다.

- **보안**

- 저장 데이터 및 전송 데이터 암호화 : AWS Key Management Service 를 통해 관리하는 키를 사용해 데이터베이스 암호화를 할 수 있다.
- 네트워크 격리 : VPC 에서 데이터베이스 인스턴스를 실행하여 방화벽 설정을 구성하고 데이터베이스 인스턴스에 대한 네트워크 액세스를 제어할 수 있다.
- IAM 리소스 수준 권한 : RDS는 IAM과 통합되어 사용자 및 그룹이 특정 RDS 리소스에서 수행할 수 있는 작업을 제어하는 기능을 제공한다.

- **관리 효율성**

- 모니터링 및 지표 : RDS는 추가비용 없이 Amazon CloudWatch와 연동해 지표를 제공한다. RDS 관리 콘솔을 사용하면 컴퓨팅/메모리/스토리지 용량 사용률, I/O 작업, 인스턴스 연결 등 주요 작업 지표를 보고 성능 문제를 신속하게 감지할 수 있는 편리한 도구인 성능 개선 도우미에 액세스할 수 있다.
- 이벤트 알림 : Amazon SNS를 통해 데이터베이스 이벤트를 알람받을 수 있다.

• 비용 효율성

- 사용한 만큼 비용 지불
- 예약 인스턴스 : RDS 예약 인스턴스는 1년 또는 3년 약정기간 동안 DB 인스턴스를 예약할 수 있는 옵션을 제공하여, 온디맨드 요금보다 저렴하게 이용가능하다.
- 인스턴스 중지 및 시작 : RDS 를 활용하면 최대 7일 까지 데이터베이스 인스턴스를 쉽게 중지했다 시작할 수 있다. 상시 구동하지 않아도 되는 개발 및 테스트 작업에 데이터베이스를 저렴하게 이용 가능하다.

• 데이터베이스 엔진 종류

| | |
|----------------------|---|
| MySQL | 가장 많이 사용되는 오픈소스 관계형 데이터베이스, 스토리지 엔진으로 RDS 의 자동 백업 기능과 호환성이 있는 InnoDB를 사용 |
| MariaDB | MySQL의 업그레이드판, 스토리지 엔진으로 XtraDB 및 InnoDB를 지원하지만 RDS 와의 호환성을 위해 InnoDB를 사용하는것을 권장 |
| Oracle | 오라클사의 유료 관계형 데이터베이스, 가장 널리 사용되는 DBMS, 유료 데이터베이스이기 때문에 라이선스 비용이 든다. |
| PostgreSQL | Oracle 호환 오픈소스 관계형 데이터베이스로 기능도 많고 성능도 가장 좋은 데이터베이스이다. Oracle 용으로 개발된 내부용 애플리케이션을 저렴하게 실행할 수 있는 방법이다 |
| Amazon Aurora | MySQL PostgreSQL 호환 데이터베이스로 가상화스토리지레이어를 이용하여 좀 더 높은 수준의 쓰기 성능을 제공한다. MySQL의 5배, PostgreSQL의 3배 정도 빠르다, MySQL 호환형, PostgreSQL 호환형 두가지 에디션으로 나뉜다. |
| Microsoft SQL Server | Microsoft 에서 개발한 관계형 데이터베이스 관리 시스템으로 RDS를 통해 쉽게 배포, 운영, 확장이 가능하다. |

• 라이선스 고려 사항

| | |
|-----------------------|---|
| 라이선스 포함 모델 | MariaDB, MySQL 는 GPL2.0, PostgreSQL 에는 PostgreSQL 라이선스의 적용을 받으며, RDS에서 실행되는 SQL Server의 모든 버전과 에디션, Oracle Database Standard Edition One, Two (SE1, SE2) 에는 라이선스비가 포함되어 있다. |
| 자체 라이선스 보유 모델 (*BYOL) | Oracle 데이터 베이스에만 해당하며 자체 보유 라이선스를 사용해야 한다. (EE, SE, SE1, SE2) |

* BYOL = Bring your own license

• 데이터베이스 옵션 그룹 (Parameter Group)

- 각 데이터베이스 엔진은 데이터베이스 관리 및 보안을 위해 다양한 기능과 옵션을 제공하는데, 사용자는 옵션 그룹을 사용하여 하나 이상의 인스턴스의 이들 기능과 옵션을 쉽게 적용 가능하다.
- 옵션 적용을 위해서는 상당한 양의 메모리가 필요하므로 인스턴스가 옵션 실행에 충분한 메모리가 있는지 확인해야 한다.
- 옵션은 엔진마다 다르다.
 - Oracle - Amazon S3 통합 기능 제공

- SQL Server, Oracle - 저장전 데이터 암호화 TDE 암호화 옵션 제공 (Transparent Data Encryption)
- MySQL, MariaDB - 쿼리 작업전 로그인을 요구하는 Audit 플러그인 제공

• 데이터베이스 인스턴스 클래스 종류

| | |
|---------------------|--|
| 스탠다드 데이터베이스 인스턴스 | 대부분 사용자의 데이터베이스에대한 요구수준에 맞춘 클래스, dbm5 등 |
| 메모리 최적화 데이터베이스 인스턴스 | 높은 수준의 성능을 요구하는 데이터베이스에 적합한 인스턴스 타입, 충분한 메모리를 제공하여 쿼리 속도를 높여 준다, db.x1e, db.z1d, db.r5 등 |
| 성능가속 데이터베이스 인스턴스 | 개발, 테스트, 비 상용화를 고려한 데이터베이스 인스턴스 db.t3, db.m5, db.r5 등은 AWS Nitro System 기반의 클래스로 최대 디스크 처리용량으로 디스크 읽기 및 쓰기 작업을 처리한다. |

• 스토리지 선택 방법 - IOPS (초당 입출력 작업량) 의 이해

- 데이터베이스 인스턴스의 스토리지를 선택하는데 있어 용량을 확보하는 것도 중요하지만, 데이터베이스 기반 애플리케이션이 원활히 작동 할 수 있는 스토리지의 처리속도도 중요한 요소
- 스토리지의 성능을 초당 입출력 작업량, IOPS (Input/Output operations Per Seconds)
- 단일 I/O 작업에서 전송 가능한 데이터의 양은 데이터베이스 엔진이 사용하는 페이지 크기에 따라 달라짐 (MySQL MariaDB : 16KB, Oracle, PostgreSQL, SQL Server : 8KB)
- 페이지가 클수록 한번의 I/O 작업에서 전송할 수 있는 데이터의 양이 많다는 것
- 즉, 필요한 데이터베이스 엔진, 디스크 처리용량(Disk Throughput), 스루풋을 지원하는 충분한 IOPS 를 측정하여 알맞은 스토리지를 선택한다.
- IOPS 계산 연습 문제
 - MariaDB, 2000Mbps의 디스크 스루풋을 지원하는 IOPS 계산
 - MariaDB의 페이지 크기는 16 KB = 128Kb 또는 0.128Mb
 - $2000Mbps / 0.128Mb = 15625 \text{ IOPS} \Rightarrow 16000 \text{ IOPS}$ 정도로 충분히 할당

• 스토리지 종류

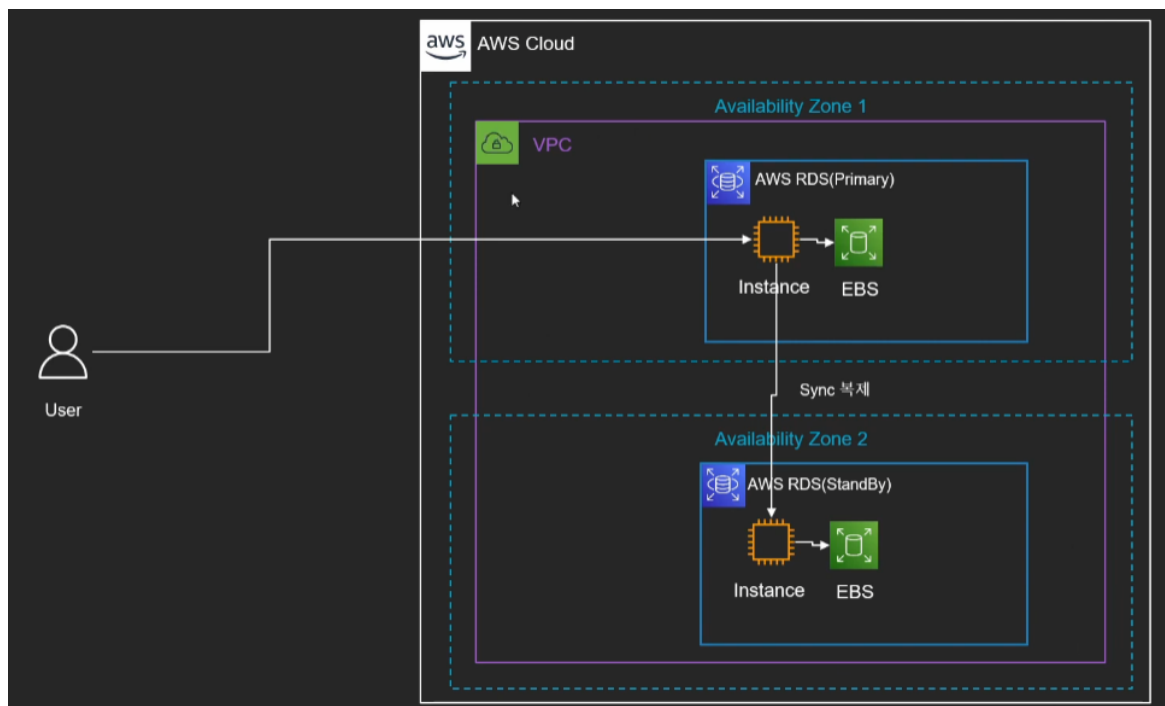
| | |
|--------------------------|--|
| 범용 SSD 스토리지 | 충분한 성능 및 밀리초 수준의 저지연성 제공 최대 64TB 볼륨 할당 (최소 20GB) 볼륨의 기가바이트 당 3 IOPS 기본성능 제공 E.g > 20GB - 60 IOPS, 5334 GB (5.33TB) - 16000 IOPS 볼륨당 최대 16,000 IOPS 처리성능 활용 가끔 최대 높은 수준의 IOPS 성능이 필요하다면 지나치게 많은 스토리지 볼륨을 할당하지 않고, 1TB 미만의 스토리지에 제공되는 최대 3,000 IOPS 순간 가속(burst) 성능을 활용 할 수 있다. $5,400,000 \text{ IOPS (크레딧 잔고)} / [3000 - 3 * (\text{스토리지 용량 GB})] = \text{초당 가속 성능}$ 사용된 크레딧 잔고는 볼륨의 초당 1 IOPS로 다시 충전된다. |
| 프로비전 IOPS SSD (io1) 스토리지 | 인스턴스 생성시 원하는 수준의 IOPS를 직접 할당 할 수 있다. 성능 가속 개념은 없으며, 일관되게 높은 수준의 성능이 필요한 경우 유용한 옵션 io1 볼륨 당 64,000 IOPS 까지 프로비전할 수 있다. 최대 IOPS 수준 및 할당 가능한 스토리지 양은 데이터베이스 엔진에 따라 달라짐 Oracle, PostgreSQL, MariaDB, MySQL, SQL Server, |

| | |
|--------------------------------|---|
| | Aurora 는 4GB에서 16TB 까지 스토리지 선택 가능하며 64,000 IOPS 까지 할당할 수 있다 GB당 최소 IOPS 비율은 50:1 이다 |
| 처리용량 최적화 HDD (st1) 스토리지 | 마그네틱 스토리지로 볼륨 크기 500MB에서 16TB 제공 TB당 40MBps 의 처리용량을 제공 16TB의 최대 처리용량은 500MBps |
| 쿨드 HDD (sc1) 스토리지 | 마그네틱 스토리지로 볼륨크기 500MB에서 16TB 제공 TB당 1MBps의 처리용량 제공 16TB의 최대 처리용량은 192MBps st1 보다 비용이 저렴하다는 장점 |
| 마그네틱 스토리지 | 역호환성 유지를 위해 구형 인스턴스를 위한 마그네틱 스토리지 제공 1TBG 용량에 100 IOPS 처리성능 제공 지원을 줄이고 있으며, 사용을 권장하지 않음 |

• RDS 구성 아키텍처 개요

| | Multi-AZ | Multi-Region | Read Replica |
|-----------------|---------------------------|----------------|----------------|
| 목적 | 고가용성 | DB / 로컬 퍼포먼스 | 확장성 / 성능 |
| 복제방식 | Sync | Async | Async |
| 액티브 | Primary DB만 Read/Write 가능 | Read 만 가능 | Read 만 가능 |
| 백업 | 자동 백업 (Standby 기준) | 자동백업 가능 | 기본적으로 백업 기능 X |
| 엔진 업데이트 | Primary만 업데이트 | 각 리전별 다른 업데이트 | DB별로 다른 업데이트 |
| FailOver | 자동으로 Standby로 Fail-Over | 수동으로 Fail-Over | 수동으로 Fail-Over |

• 고가용성 구현 (멀티 AZ)

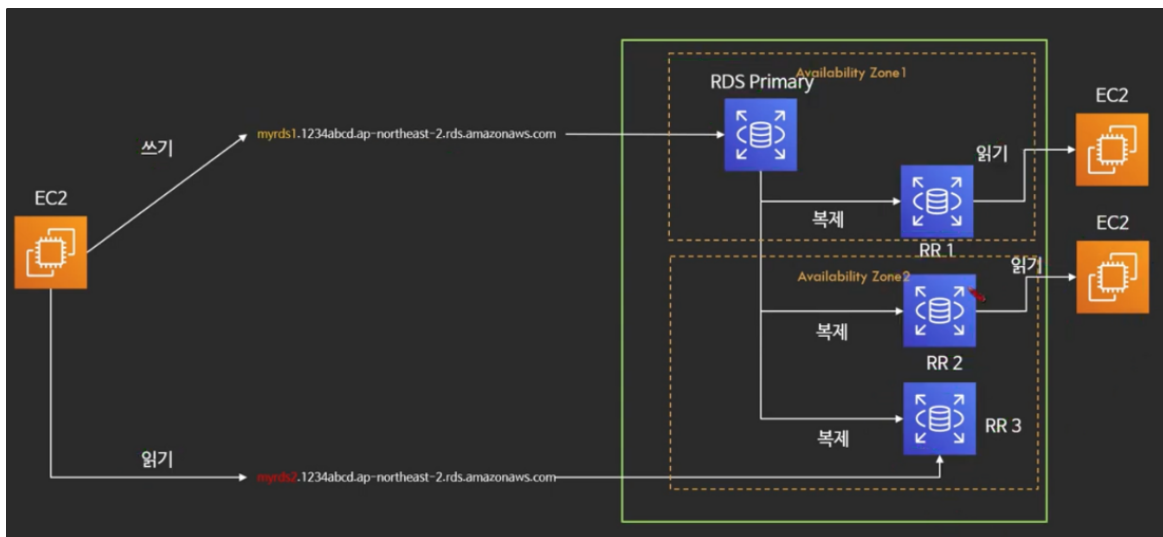


- 두개 이상의 AZ에 걸쳐 데이터베이스를 구축하고 원본과 다른 DB를 Standby 하여 자동으로 동기화
- 위 그림에서 AZ Zone 1 의 RDS 장애 시 AZ Zone 2 의 RDS 가 자동 승격 및 재해복구 수행

- DNS를 자동으로 Standby 인스턴스로 연결하기에 빠른 속도로 장애복구, 사용자는 장애를 느끼지 못함
- 다만 Muti-AZ 는 예비용 인스턴스가 하나 더 있기에 비용이 두배
- 성능개선의 목적이 아닌 안정성을 위한 서비스

• Read Replica (읽기 전용 복제본) - Scale-up / Scale-Out

| | |
|---------------------------|--|
| Scale-Up (수직적 확장) | 인스턴스와 관련된 리소스만 추가하는 방법 |
| Scale-Out (수평적 확장) | 읽기 전용 복제본 이라고 불리는 Read Replica 인스턴스를 추가하여 쿼리 작업 성능 향상, SQL Server 를 제외한 모든 데이터베이스 엔진이 지원하며 Aurora의 경우 Aurora replica를 제공한다 |



- Read Replica 는 RDS Primary 를 복제해서 DB 쓰기는 Primary에서, DB 읽기는 복제본에서 처리하는 구조
- DB 읽기 위주의 작업이 많은 경우 복제본을 생성하여 부하 분산
- 복제본은 같은 AZ (Scale-up) 또는 다른 AZ (Scale-out) 에 있을 수 있고 심지어는 다른 리전에도 존재할 수 있으며 각각의 복제본 인스턴스에 DNS가 부여되어 접근이 가능
- 이렇게 읽기와 쓰기를 분별하여 트래픽 분산하는 기술을 데이터베이스 Replication 이라 한다.
- 약간의 시간차를 갖고 데이터가 복제되기 때문에 복제본의 데이터가 일치하지 않을 수 있다.
- AWS 에서는 총 5개의 복제본이 생성 가능하며, 복제본의 복제본도 생성 가능
- 원본 DB의 장애 발생 시 수동으로 DNS 변경이 필요하다

• Amazon Aurora 의 Multi AZ 구현

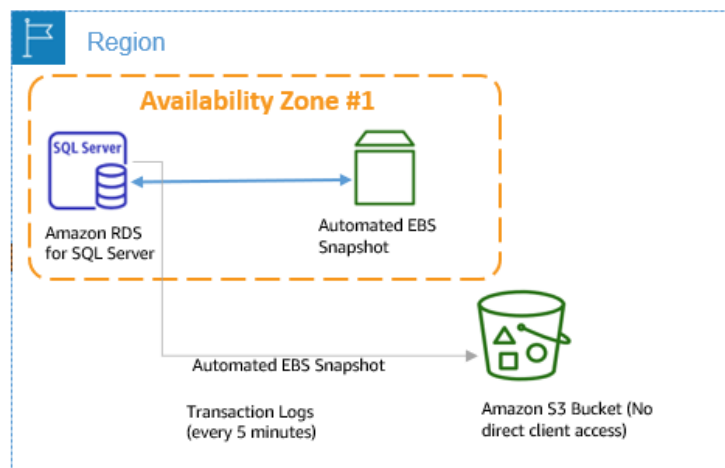
| | |
|---------------|---|
| 싱글 마스터 | 프라이머리 인스턴스를 구성하여, 해당 인스턴스를 가리키는 클러스터 엔드포인트를 제공 프라이머리 인스턴스와 레플리카는 AZ에서 동기적으로 복제되는 클러스터 볼륨을 공유하고 클러스터 볼륨은 필요에 따라 64TB까지 확장 프라이머리 인스턴스가 실패 할 때 → Aurora 레플리카가 없다면 새 프라이머리 인스턴스를 생성, Aurora 레플리카가 있다면 레플리카를 프라이머리로 승격 |
| 멀티 마스터 | 모든 인스턴스가 쓰기작업을 할 수 있으며, 별도의 페일오버 작업이 실행 되지 않음, 이를 지속적 가용성이라 부른다. |

- **RDS Multi Region 구성**

- 다른 리전에 지속적으로 동기화 하는 DB 클러스터 생성 (Async 복제)
- 마스터 리전과 Read Replication을 다른 리전에 두어 Latency를 줄임
- 주로 로컬 퍼포먼스 또는 DR (Disaster Recovery) 시나리오로 활용
 - 로컬 퍼포먼스 : 지역 사람들이 빨리 읽을 수 있게 함
 - DR 시나리오 : 재해복구 시나리오
- 각 리전별 자동백업 가능
- 리전별 Multi-AZ 구성 가능

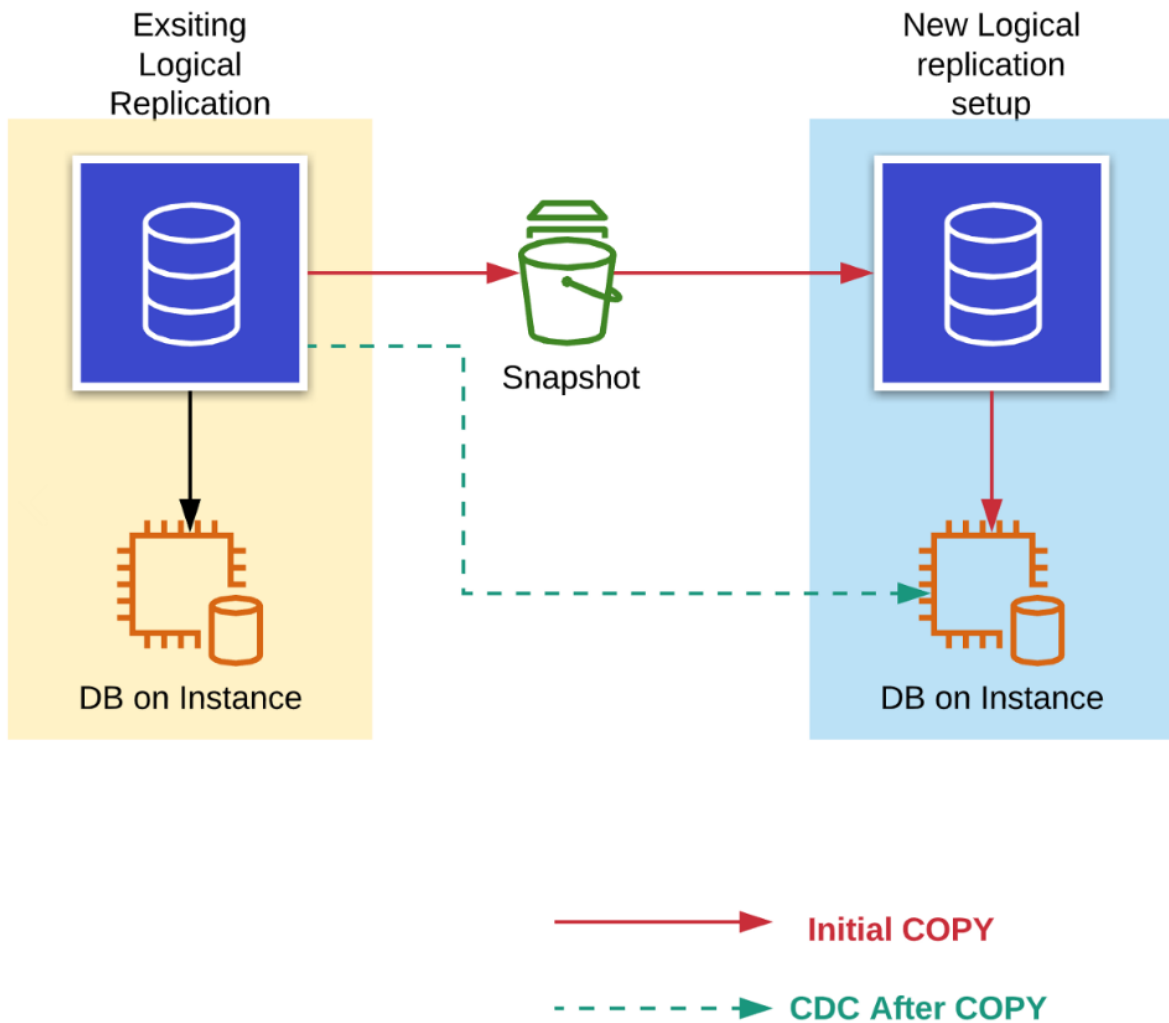
- **백업과 복원**

- 자동 백업(Automated Backups = AB)



- 매일마다 EBS 스냅샷과 트랜잭션 로그를 참고하여 자동으로 백업
- Default로 AB 기능 설정
- Retention Priod(1~35일) 안의 과거 특정 시간으로 되돌아갈 수 있다.
- 다른 DB 인스턴스를 새로 생성하여 스냅샷을 적용 시키는 형식
- RDB 백업 정보는 S3에 저장되며, AB 동안 약간의 IO suspension(딜레이) 가 존재
- Multi-AZ 로 구성 시 Standby를 통해 백업을 수행하기에 딜레이가 줄어듬
- 수동 백업 (DB 스냅샷)
 - AB는 인스턴스를 삭제 시 스냅샷도 모두 삭제되지만 수동 백업은 원본 RDS 를 삭제 하더라도 스냅샷은 S3에 그대로 남아있다.
 - 수동 백업은 유저 혹은 다른 프로세스로부터 요청에 따라 만들어지는 DB 스냅샷

- 스냅샷의 복구는 항상 새로운 인스턴스를 생성하여 수행된다.



- RDS 백업 복원시 도메인 변경

- 원본 RDS 인스턴스를 가지고 새로운 DB 복원 시 새로운 인스턴스와 Endpoint 생성.
- 원본 DNS는 앞에 Original인 반면, 복원된 것은 앞에 restored가 붙게 된다.

`original.ap-northeast-2.rds.amazonaws.com` `restored.ap-northeast-2.rds.amazonaws.com`

- 유지보수

- RDS 는 관리형 서비스 이므로 패치 및 업그레이드 등 데이터베이스 관리와 관련된 모든 작업을 AWS 가 처리한다.

- **RDS 인증방식**

- 유저/패스워드 방식
 - AWS Secret Manager(기타 유저 패스워드 인증관리 서비스) 와 연동하여 자동 로테이션 가능
 - 일정 주기마다 패스워드 변경
- IAM DB 인증
 - IAM 유저 Credential/Role을 통해 관리 가능
- Kerberos 인증

- **Amazon Redshift**

- 관리형 데이터 웨어하우스 서비스
- PostgreSQL을 기반으로 하지만 RDS와 별개로 존재
- 칼럼 단위의 칼럼형 스토리지로 저장속도가 빠르고 효율적이며 개별 칼럼에 신속하게 쿼리작업을 수행한다.
- 스토리지에서 소요되는 칼럼의 수를 줄이기 위해 압축 인코딩 기법을 사용
- 컴퓨트 노드
 - Redshift 클러스터는 하나 이상의 컴퓨터 노드를 지니고, 덴스(Dense) 리더 (Leader) 두가지 종류가 있다.
 - Dense 컴퓨트 노드 : 마그네틱 스토리지에 최대 326TB의 데이터 저장, SSD에 최대 8192TB 데이터 저장 가능
 - Leader 컴퓨트 노드 : 클러스터에 하나이상의 컴퓨트 노드가 있는 경우 리더 노드를 추가하여 컴퓨트 노드 간 커뮤니케이션 조정 및 소통할 수 있도록 하며, 사용 이용료는 없다
- 데이터 분산 유형
 - 행(row)은 다수의 컴퓨트 노드에 분산 저장
 - 분산 방식으로 EVEN(균일분산), KEY(단일컬럼내 값에 따라 분산), ALL(모두 동일하게 분산) 유형
- Redshift Spectrum
 - S3에 저장된 파일에서 데이터를 쿼리할 수 있는 서비스로 Redshift 클러스터 내로 데이터를 임포트할 필요가 없다
- AWS DMS (데이터베이스 마이그레이션 서비스)
 - 기존 데이터베이스와 스키마를 자동으로 복사해서 다른 데이터베이스에 마이그레이션
 - 관계형 비 관계형 데이터베이스간, 서로 다른 데이터베이스 엔진 간 마이그레이션을 지원한다.
 - DMS가 지원하는 데이터베이스 엔진 : Aurora, DynamoDB, IBM DB2, MariaDB, MongoDB, MySQL, Oracle, PostgreSQL, Redshift, S3, SAP

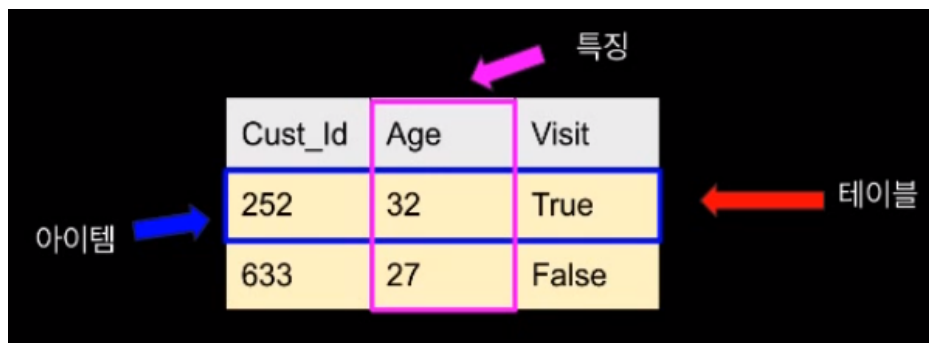
- **비 관계형 데이터베이스**

- 키/밸류 스토어타입과 도큐먼트 지향 스토어 타입이 있지만 기본적으로 키/밸류 스토어 데이터베이스 이다.
- Amazon Neptune : 그래프 데이터베이스 → 아이템 속성 간 관계 분석

- **DynamoDB**

- 각종 관리 업무를 지원하는 비관계형 데이터베이스 서비스
- 다수의 파티션에 분산된 데이터 구조를 이용하여 초당 수천회의 읽기 및 쓰기 작업을 처리
- 파티션이란 테이블을 위한 스토리지 할당 영역이며, 다수의 AZ에 존재하는 SSD 장치를 이용
- 테이블 생성시 스키마 생성이 필요 없고, 기본 키와 데이터 타입만 명시
- Auto Scaling 기능
- 실시간 데이터를 보관하는데 탁월
- 모바일, 웹, IoT 사용시 추천

- DynamoDB의 구성



- 테이블
- 아이템(Items) - 행(row)과 개념이 비슷, 최대 용량 400KB
- 특징(Attributes) - 하나의 키/벨류 쌍 의미
- Key-Value

- **DynamoDB의 키**

- PK를 사용하여 데이터 쿼리
- 파티션키(해시키)와 복합키가 있다.

| | |
|---|--|
| 파티션 키 (Partition Key) (해시 키, Hash Key) | 데이터를 나누고 분리 시키는 키 고유 특징 실제 데이터가 들어가는 위치를 결정 → 파티션키 내부에 있는 해시함수를 돌리고 해시값이 반환되면 이 값이 데이터가 저장 되는 주소 값 |
| 복합 키 (Composite Key) | 파티션 키 + 정렬 키 |

- **DynamoDB의 데이터 접근 권한**

- AWS IAM으로 관리 할 수 있다.
- 테이블의 생성과 접근 권한을 부여할 수 있고, 특정 테이블만, 특정 데이터만 접근 가능하도록 해주는 별도의 IAM 역할이 존재

- **DynamoDB 의 스루풋(Throughput) 옵션 선택**

- 테이블 생성 시 온디맨드 / 프로비전 모드로 선택하여 사용 가능
- 테이블 생성 시 읽기 용량 유닛(RCU), 쓰기 용량 유닛 수(WCU)를 기준으로 파티션을 제공
- 프로비전 모드에서 강한 일관성 읽기, 종국적 일관성 읽기 방식 선택

- **DynamoDB 의 Auto Scaling**

- 활성화 비율과 RCU 및 WCU의 최대 최소값을 지정하여 미리 지정한 활성화 비율에 따라 RCU 및 WCU를 자동으로 조절
- 예약 처리 용량 : 100유닛 이상의 WCU 또는 RCU를 필요로 하는 경우 예약 처리 용량(reserved throughput capacity)을 구매 해 비용 절감

- **DynamoDB 의 데이터 읽기**

- 읽기 방식은 스캔(Scan)과 쿼리(Query) 두가지의 테이블 읽기 방식을 제공

| | |
|------------|---|
| 스캔 (Scan) | 테이블 내 모든 아이템 반환, 읽기 집약적 방식으로 읽기 유닛을 모두 소모할 수 있다, ProjectionExpression 파라미터로 보고싶은 컬럼만 볼 수 있도록 수정 (필터링) |
| 쿼리 (Query) | 파티션 키 값에 따라 아이템을 반환, 소트 키가 있는 경우 소트 키를 함께 이용하여 쿼리 작업을 할 수 있다, ProjectionExpression 파라미터 |

- **DynamoDB 의 Secondary Index (보조 인덱스)**

- 많은 애플리케이션에서는 주요 키가 아닌 속성을 가진 데이터에 효율적으로 액세스할 수 있다는 장점을 활용하기 위해 하나 이상의 보조(또는 대체) 키를 사용
- 보조 인덱스는 쿼리 작업을 지원하는 대체 키와 함께 테이블의 속성 하위 집합을 포함하는 데이터 구조
- 두가지 유형의 Index 존재

| | |
|-----------------------------|--|
| LSI(Local Secondary Index) | 테이블 생성 시에만 정의, 테이블 생성 후 변경 삭제 불가능, 같은 파티션 키 & 다른 정렬 키 사용 |
| GSI(Global Secondary Index) | 테이블 생성 후에도 추가 변경 삭제 가능, 다른 파티션 키 & 정렬 키 사용 |

- **DynamoDB 의 전역 테이블**
















- 가용성을 높이기 위해 다수의 리전에서 해당 테이블을 복제하여 사용 가능하다.
- 리전당 하나의 읽기전용 테이블만 지닐 수 있다.
- Auto Scaling 활성화 필요
- 리전간 강한 일관성 읽기는 지원하지 않음

- **DynamoDB 의 백업**

- 백업은 언제나 가능하며, RCU를 소모하지 않고 DynamoDB의 성능에 영향을 주지 않음

- 횟수 제한이 없고 인덱스 및 암호화 옵션도 설정 가능하다.

• 이외 다양한 AWS 데이터베이스 서비스

| | | | | | | | |
|---|---|---|---|---|--|---|---|
| |  |  |  |  |  |  |  |
| | 관계형 DB | 키-값 | 문서 | 인-메모리 | 그래프 | 시계열 | 원장 |
|  |  |  |  |  |  |  |  |
| Amazon RDS for MySQL, PostgreSQL, MariaDB, Oracle, SQL Server | Amazon Aurora for MySQL & PostgreSQL | Amazon DynamoDB | Amazon DocumentDB with MongoDB compatibility | Amazon ElastiCache for Redis & Memcached | Amazon Neptune | Amazon Timestream | Amazon QLDB |
| 주요 특징 | 참조 무결성, ACID 트랜잭션, Schema-on-Write | Key lookup 빠른 응답시간, 빠른 처리량 | 각종 문서 저장 인덱싱 | 1/1000 초 미만의 응답시간. 메모리 스토어 | 쉽고 빠르게 데이터 관계 생성 및 탐색 | 시간에 따른 순차 데이터 수집, 저장 및 처리 | 모든 변경에 대한 완전, 불변, 검증 가능한 기록 |
| 요구 사항 | 웹 서비스, 상품 카탈로그, ERP, CRM | 실시간 입찰 장바구니 OLTP | 콘텐츠 관리, 개인에 맞춰 편집된 환경 관리 | 리더 보드, 실시간 분석, 캐싱 | 소셜 네트워킹 추천 서비스 | 사물 인터넷, 시계열 로그 분석 | 헬스케어, 금융 정보, 공급망 관리, 판매 이력 |

- Amazon DocumentDB : 문서 기반 데이터베이스 제공
- Amazon Neptune : 그래프 데이터베이스 제공
- Amazon Timestream : 시계열 데이터베이스 제공
- Amazon Quantum Ledger Database : 장부 데이터베이스 제공
- Amazon ElastiCache : 인메모리 데이터베이스, 메모리를 활용하여 처리속도가 빠르고 Redis와 Memcached 용이 있다.