

이상희 교사

# AWS를 활용한 웹 서비스 구성하기

## 4단원 강의

▶ 1차수: DB서비스(RDS)

2차수: DB서비스(DynamoDB)

3차수: 서버리스 서비스

## 1차수

### 학습목표

- AWS의 DATABASE 서비스 개요와 용도 그리고 RDS 서비스를 이해 하고 구축 할 수 있다

## 1차수

### 학습내용

- AWS의 DATABASE 서비스
- 관계형 DB VS 비관계형 DB
- 목적별 데이터베이스
- DATABASE 선택 기준
- Amazon RDS/제공엔진 /단순 아키텍처/다중 AZ배포
- Amazon RDS 다중 AZ장애조치 이벤트/읽기전용복제본/암호화



Amazon Relational Database Service(Amazon RDS)



Amazon Aurora



Amazon Redshift



Amazon DocumentDB  
(MongoDB 호환)



Amazon  
DynamoDB



Amazon ElastiCache



Amazon Keyspaces  
(Apache Cassandra용)



Amazon Timestream



Amazon Quantum Ledger  
Database(Amazon QLDB)



Amazon Neptune



Amazon Relational Database Service(Amazon RDS)



Amazon Aurora



Amazon Redshift



Amazon DocumentDB  
(MongoDB 호환)



Amazon  
DynamoDB



Amazon ElastiCache



Amazon Keyspaces  
(Apache Cassandra 호환)



Amazon Timestream



Amazon Quantum Ledger  
Database(Amazon QLDB)



Amazon Neptune








- Amazon Relational Database Service(Amazon RDS)는 AWS 클라우드에서 관계형 데이터베이스(RDBMS)를 더 쉽게 설치, 운영 및 확장할 수 있는 웹 서비스

참고 -관계형 데이터베이스란?

1969년 IBM의 연구원으로 있던 E.F.Codd 박사가 수학적 기초이론에 근거를 두고 고안한 것이 관계형 데이터베이스 (Relational Database) 이다.

데이터베이스는 최소한의 의미를 가지는 테이블들로 구성되며 그 테이블들에 있는 필드들로 연결한 것이다.

참고: 2000년대중반에 다른 데이터모델들이 채택되고 사용되는 현상이 발생  
이러한 새데이터베이스와 데이터모델등급을 차별화하고 분류하기위해  
NoSQL이란용어가 대두  
NoSQL이란 용어는 비관계형과 같은 의미로 사용

	관계형(SQL) 데이터베이스	비관계형(NoSQL) 데이터베이스
AWS 데이터베이스 예	 Amazon RDS  Amazon Redshift  Aurora	 DynamoDB  ElastiCache
데이터 스토리지	행과 열로 이뤄진 테이블	키 값, 와이드 컬럼, 그래프, 문서 또는 기타 모델
스키마	고정	동적

## 관계형DB

- 데이터 항목간에 사전정의된 관계가 있을때의 데이터 항목의모음을 의미
- 항목은 열과행으로 이루어진 테이블 집합으로 구성  
테이블의 각 열은 특정종류의 데이터를 수록
- 필드는 속성의 실제값을저장
- 테이블의행은 한 객체 또는 엔터티와 관련된 값의  
모음
- 데이터는 DB 테이블 자체를 재구성하지 않고도  
여러가지방법 으로 액세스 가능

## 비관계형 DB(NoSQL)

- 가용성과확장성이뛰어나며  
높은성능을유지하도록 최적화된  
비관계형데이터베이스시스템을설명  
하는데 사용되는용어
- NoSQL 데이터베이스는 키값페어 또는  
문서스토리지 같은 데이터관리에  
관계형모델대신 대체모델을 사용



- 데이터는 테이블로 구분됩니다.
- 관계형 데이터는 고도로 정형화되어 있습니다.
- 관계는 시스템에서 적용하는 키를 통해 설정됩니다.
- 관계형 데이터는 데이터 정확성 및 일관성을 제공합니다.





얼마나 많은 처리량이  
필요한가?

단순 데이터 구조, 대용량  
데이터 객체, 또는 모두를  
저장해야 하는가?

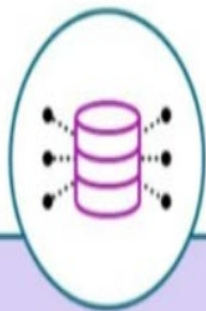
어떤 수준의 데이터 내구성,  
데이터 가용성 및 복구성이  
필요한가?

## 고려사항

- 읽기/쓰기요구사항
- 총스토리지요구사항
- 일반적인 객체크기 및 객체 액세스의 특성
- 내구성요구사항
- 대기시간요구사항
- 최대동시사용자수
- 쿼리의특성
- 필요한 무결성제어의 강도



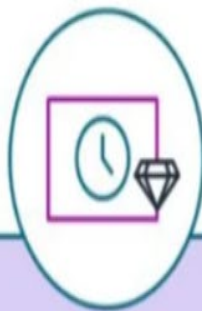
관계형 데이터베이스	비관계형(NoSQL) 데이터베이스
엄격한 스키마 규칙 및 데이터 품질 적용이 필요합니다.	데이터베이스를 수평 스케일링해야 합니다.
데이터베이스가 과도한 읽기/쓰기 용량을 필요로 하지 않습니다.	데이터가 기존 스키마에 적합하지 않습니다.
최상의 성능을 필요로 하지 않는 관계형 데이터 세트의 경우 관계형 데이터베이스 관리 시스템이 자원 소비가 적은 최고의 솔루션이 될 수 있습니다.	읽기/쓰기 속도가 기존 SQL 데이터베이스에서 경제적으로 지원할 수 있는 범위를 초과합니다.



하드웨어, OS 및  
데이터베이스  
소프트웨어 배포 및  
유지 관리  
기본 제공 모니터링



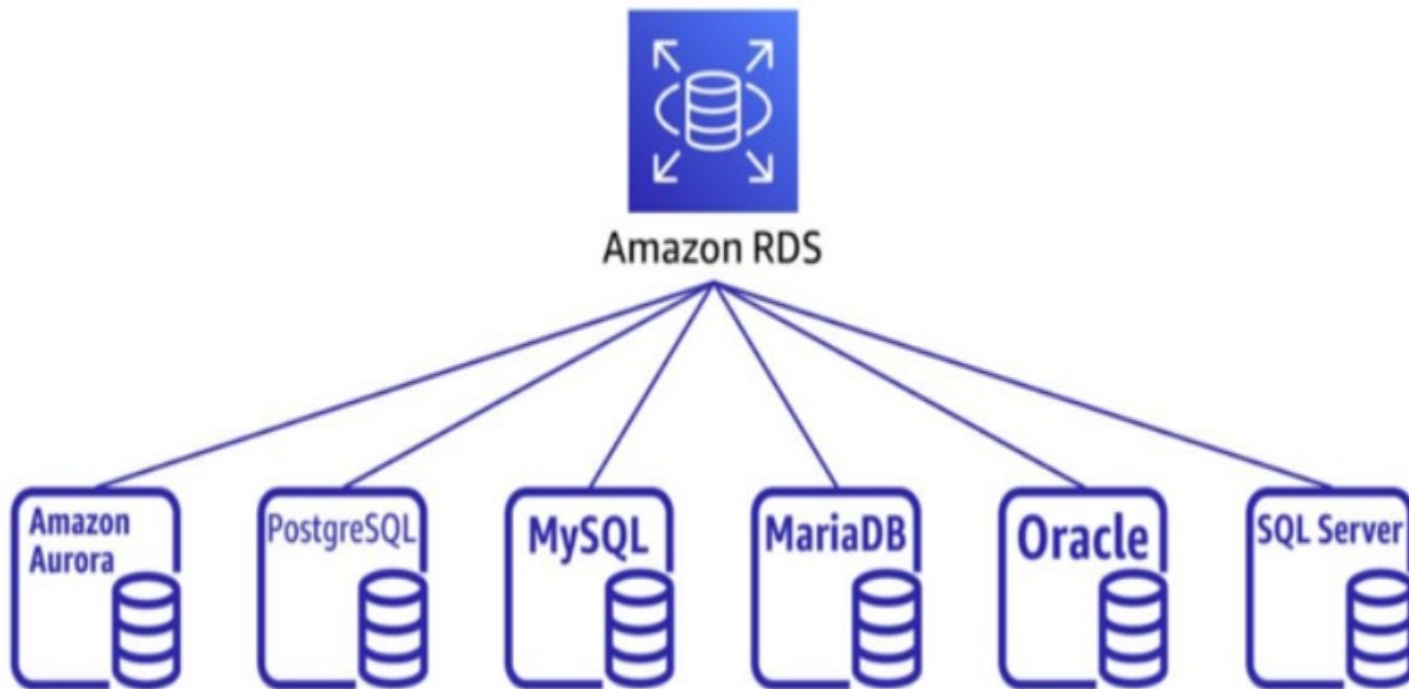
저장 시 및 전송 중  
데이터 암호화  
업계 규정 준수

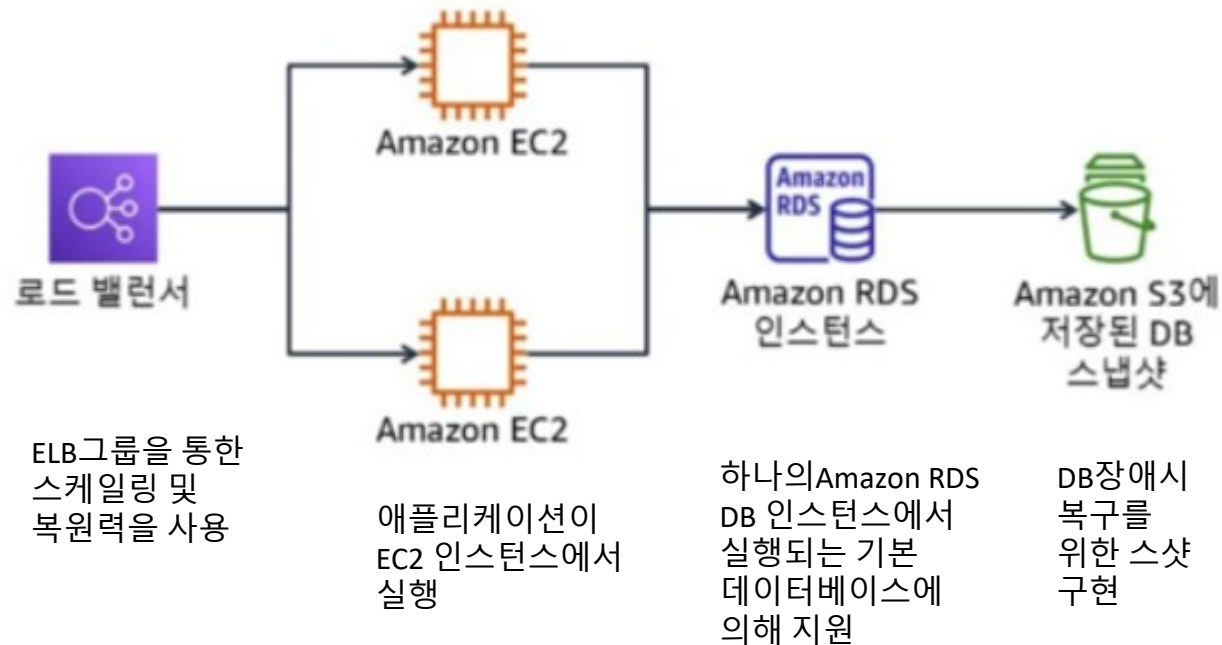


자동 다중 AZ 데이터  
복제



컴퓨팅 및 스토리지  
크기 조정  
최소한의 애플리케이션  
가동 중단

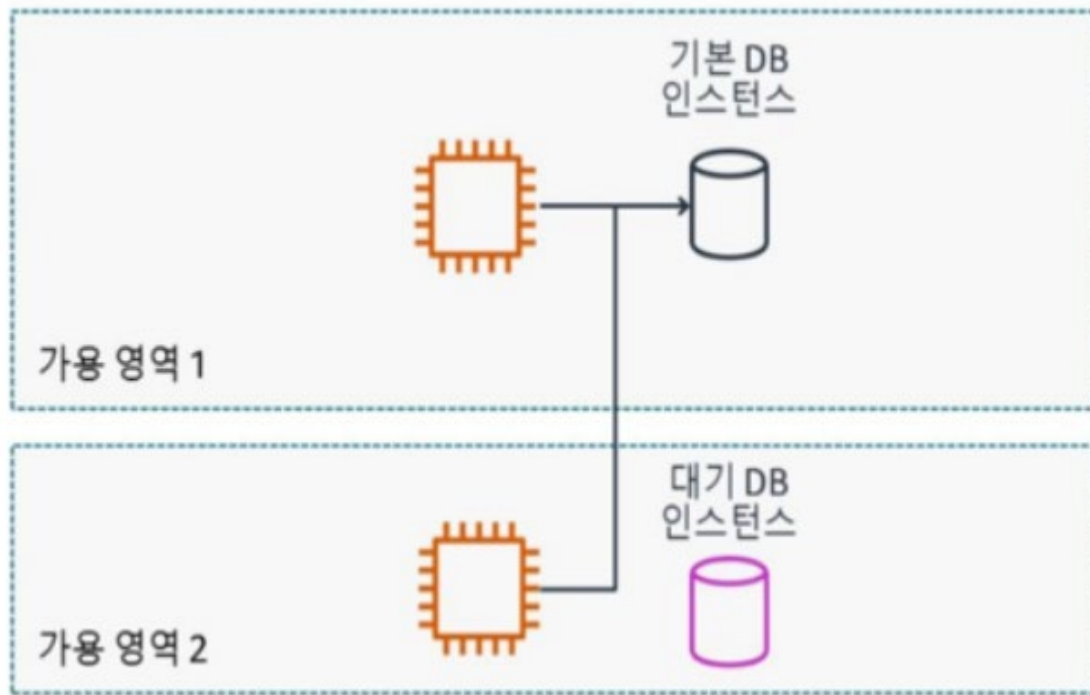






## 다중 AZ 배포:

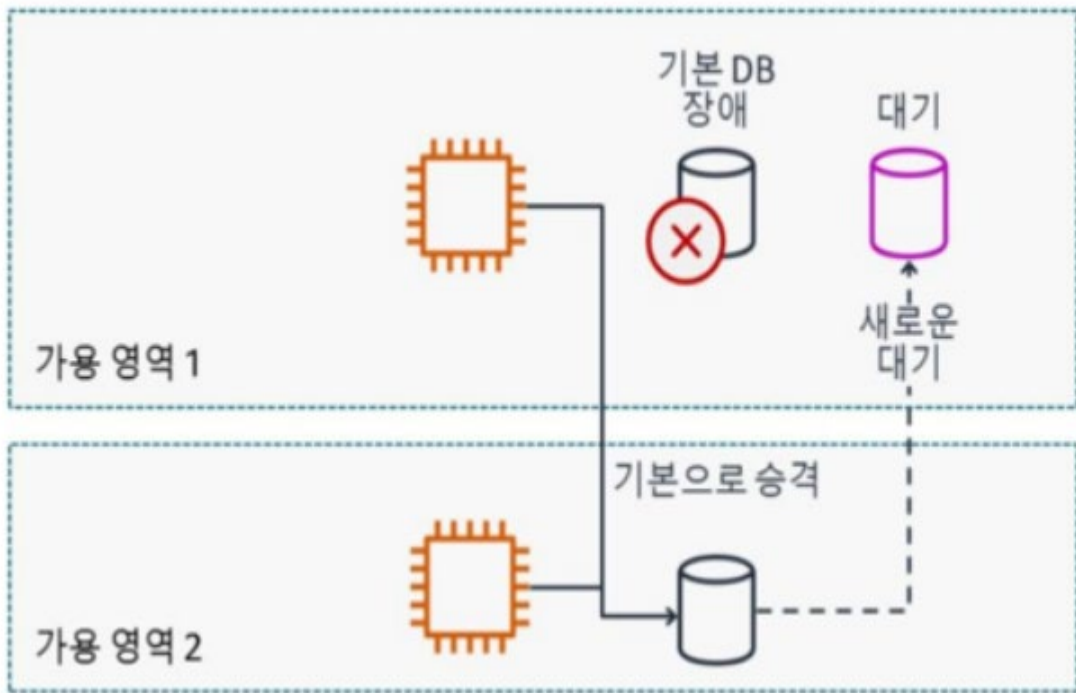
- 다른 가용 영역의 대기 DB 인스턴스에 데이터를 복제
- 읽기 전용 시나리오에서는 사용되지 않음



다중 AZ DB 인스턴스를 프로비저닝하는 경우  
Amazon RDS에서 다른가용영역에 있는  
예비인스턴스에 데이터를 동기식으로 복제함



- 장애 발생 시 보조 서버가 로드를 처리



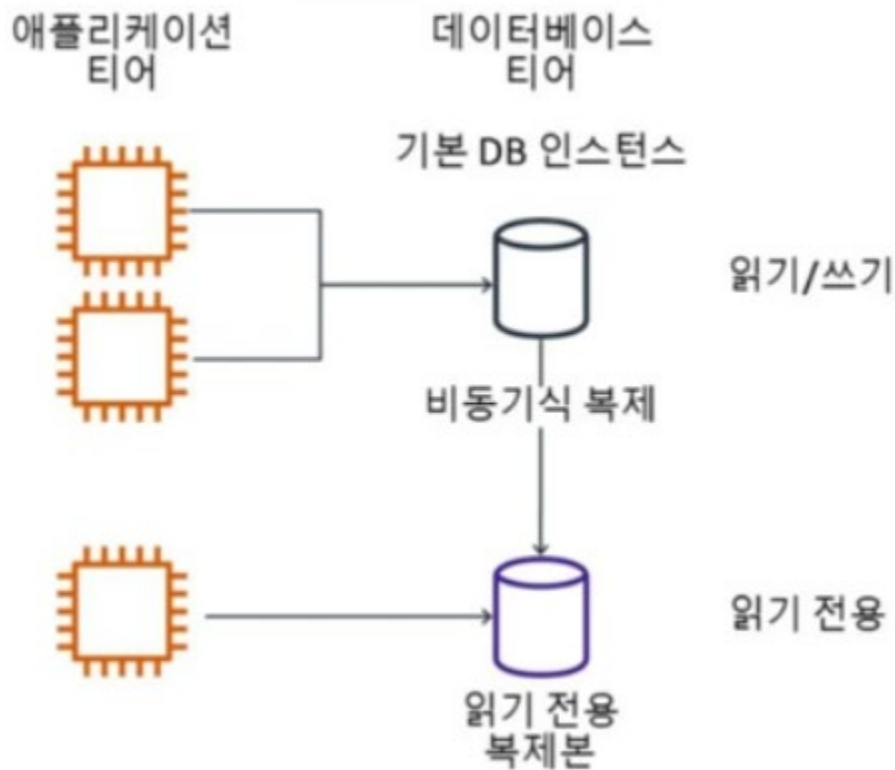
장애조치 완료 즉시 DB작업재개 됨

장애조치후에도DB 인스턴스의 엔드포인트는 변경되지 않으므로,  
관리자 개입없이 애플리케이션이 데이터베이스작업 재개함



읽기 전용  
복제본으로 다음이  
가능합니다.

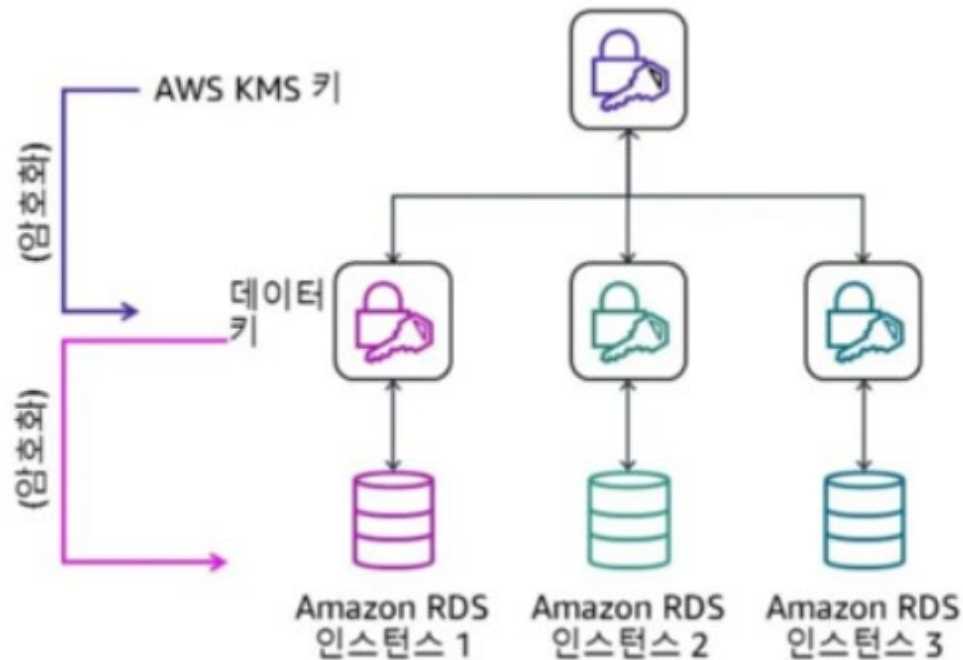
- 읽기 중심의 워크로드 처리를 위해 수평 스케일링
- 오프로드 리포팅
- AWS 리전 간 복제



Amazon에서 자동으로복제본을 기본DB 인스턴스와 동기화  
읽기전용복제본은Amazon RDS for Aurora, MySQL, MariaDB, PostgreSQL, Oracle,  
Microsoft SQL Server에서 사용가능



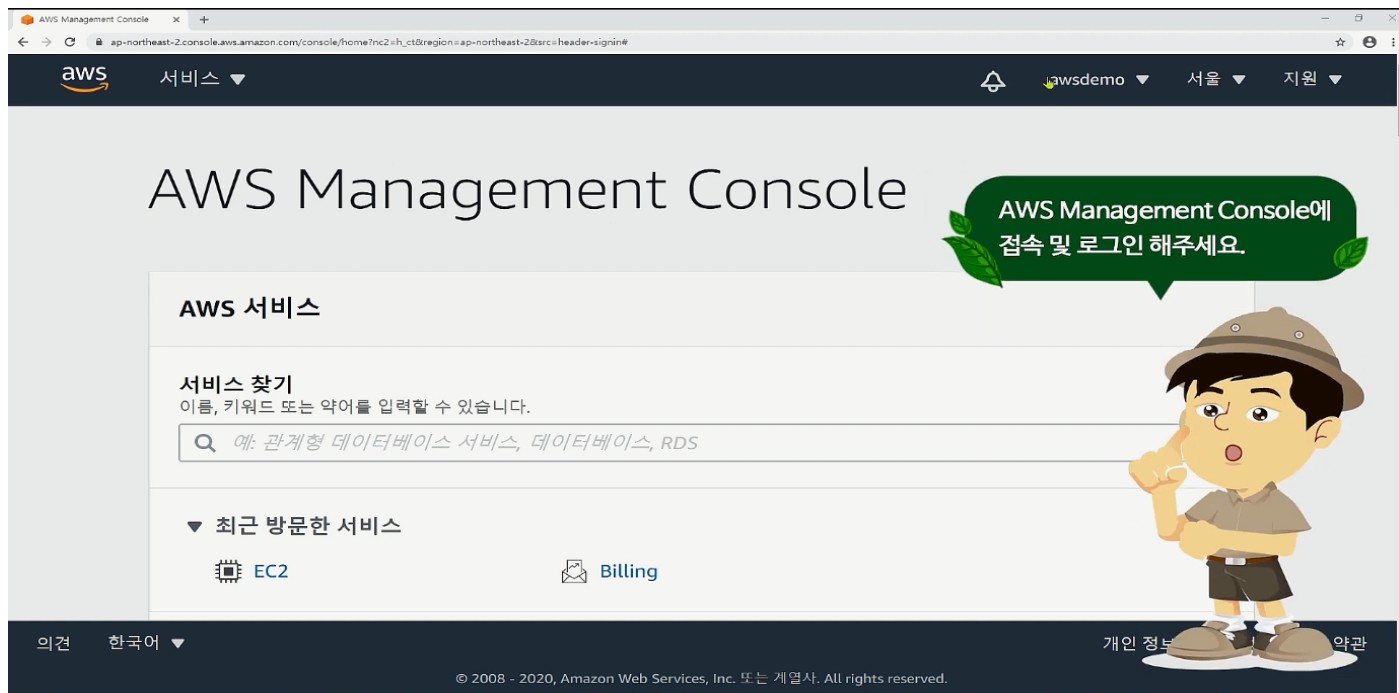
- AWS KMS에서 관리
- 고유 데이터 키로 데이터 암호화
- AWS KMS 키로 데이터 키 암호화
- 모든 RDS 엔진에서 사용 가능





## RDS 서비스 실습 시연

<https://rumble.com/v33ukx6-aws-rdsmysql-2023.html>



Q1

다음 중 AWS 가 제공하는 RDS 데이터베이스엔진이 아닌것은?

가.Amazon RDS for MySQL

나.Amazon RDS for Oracle

다.Amazon RDS for SQL Server

라.Amazon RDS for IBM DB2

A

라 (Amazon RDS for IBM DB2는 지원하지 않음)

Q2

다음 관계형 데이터베이스의 정의를 설명한 내용중 괄호 안에 들어갈 용어를 쓰시오

1969년 IBM의 연구원으로 있던 E.F.Codd 박사가 수학적 기초이론에 근거를 두고 고안한 것이 관계형 데이터베이스 (Relational Database) 이다.

데이터베이스는 최소한의 의미를 가지는 ( )들로 구성되며 그 ( )들에 있는 필드들로 연결한 것이다.

A

테이블(Table)

이상희 교사

# AWS를 활용한 웹 서비스 구성하기

## 4주차 강의

1차수: DB서비스(RDS)

▶ 2차수: DB서비스(DynamoDB)

3차수: 서버리스 서비스

## 2차수

### 학습목표

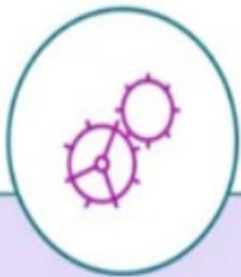
- AWS Dynamo DB 서비스를 이해하고 간단히 구현해볼 수 있다

## 2차수

### 학습내용

- Dynamo DB 특징
- NoSQL 데이터베이스
- DynamoDB 테이블
- 간단한 애플리케이션 아키텍처
- 사용사례
- 읽기/쓰기용량 오토스케일링
- 두가지 스케일링
- DB에 대한 일관성 옵션
- DynamoDB 글로벌테이블

## 완전관리형 NoSQL AWS 데이터베이스 서비스



### 규모에 따른 성능

어떤규모에서든 일관되게 수  
밀리초미만의 응답시간 제공  
거의무제한의 처리량 필요한  
애플리케이션 구축가능



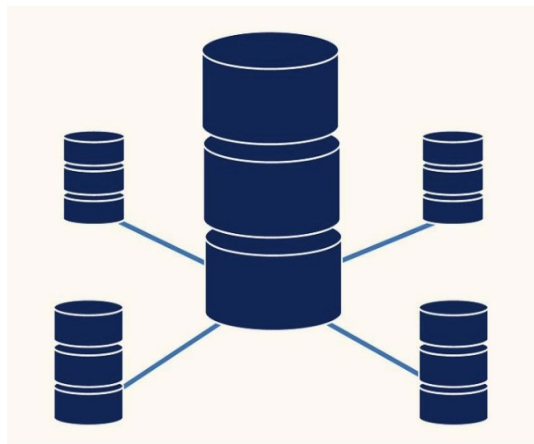
### 관리할 서버 없음

서버프로비저닝, sw패치적  
용 및 관리 불필요, 용량에  
맞게 테이블 자동확장/축소  
하여 성능 유지



### 엔터프라이즈에 사용 가능

기본적으로 모든 데이터  
암호화 강력한 보안을 위해  
aws id 및 접근관리와 완벽히  
통합



- ✓ 빅데이터 기술과 더불어 많이 알려진 데이터베이스 기술
- ✓ 하나의 논리적 데이터베이스가 통신 네트워크로 연결되어 여러대의 컴퓨터에 분산되어 저장되어 관리되는 DB
- ✓ 각각의 컴퓨터에 local DBMS와 분산 DBMS를 내장
- ✓ NOSQL은 빅데이터 시스템이 분산시스템에서 구축되기 때문에 필수로 활용
- ✓ IBM, 아파치, 하둡, 페이스북 카산드라에서 활용되고 있음
- ✓ 스마트 디바이스의 급속한 보급과 더불어 디지털 빅데이터가 늘어남에 따라 널리 사용



- 필수 키 값 액세스 패턴
- 파티션 키가 데이터 분산을 결정
- 정렬 키가 다양한 쿼리 기능을 제공

복합 기본 키

파티션 키	정렬 키(선택 사항)			
배우	역할	연도	장르	신장
Tom Hanks	Chuck Noland	2000	드라마	
Tim Allen	Buzz Lightyear	1995		5' 10"

항목 {

속성

핵심구성요소- 테이블, 항목 및 속성

테이블-항목의 모음 항목-속성의 모음

DynamoDB는 기본키를 사용하여 테이블의 각 항목을 고유하게 식별  
테이블 생성시 필수적으로 이름과 기본키를 지정해야 함



EC2인스턴스에서 가동중인 애플리케이션에서 처리하는 데이터를 DynamoDB에 저장



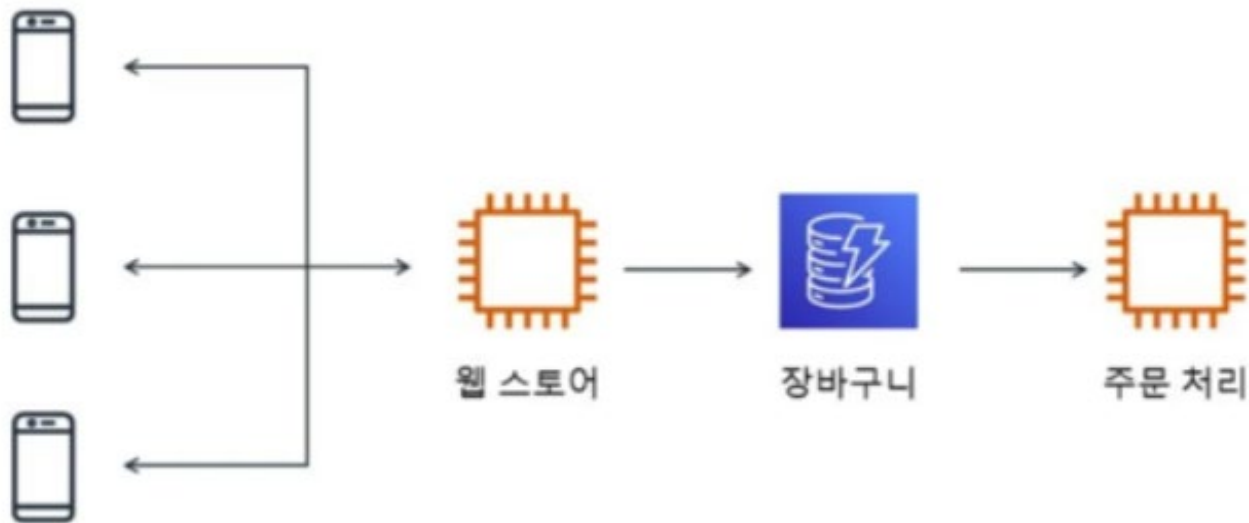
## 순위표 및 점수 매기기



Userid	GameTitle	TopScore	TopScoreDateTime	승	패
101	"Galaxy Invaders"	5,842	"2017-09-15:17:24:31"	21	72
101	"Meteor Blasters"	1,000	"2017-10-22:23:18:01"	12	3

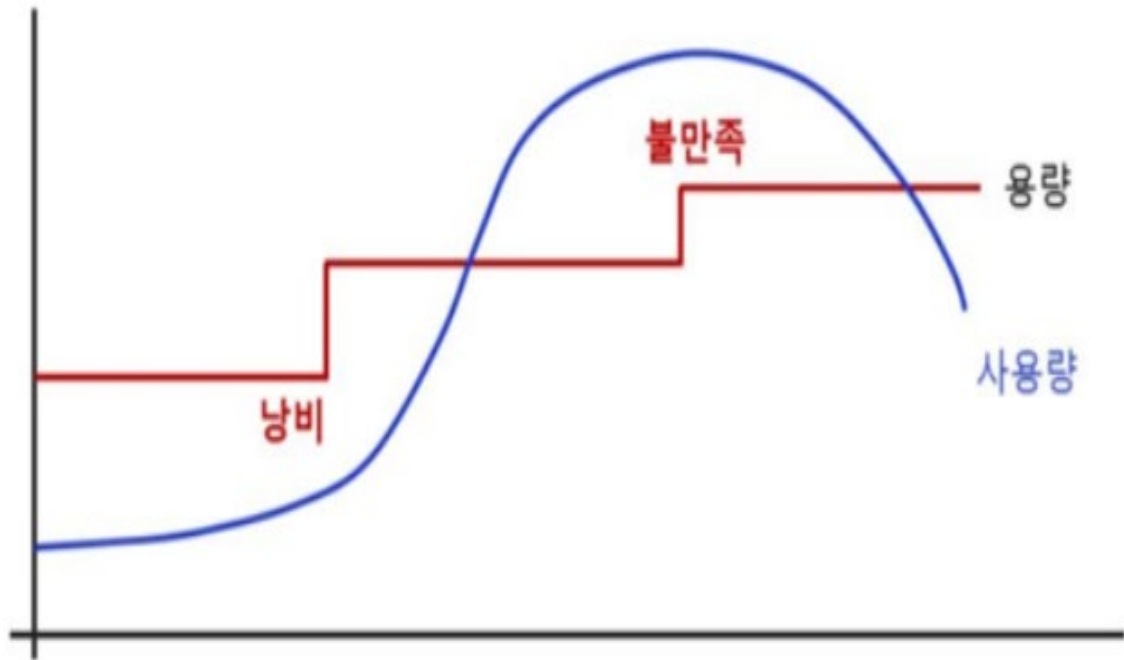
게임제작자는 DynamoDB를 사용하여 간단한 순위표를 지원가능  
이경우 게임의 최고점수를 표시하는 기능을 제공하기 위해  
글로벌 보조인덱스를 사용할 수 있음

## 임시 데이터(온라인 장바구니)



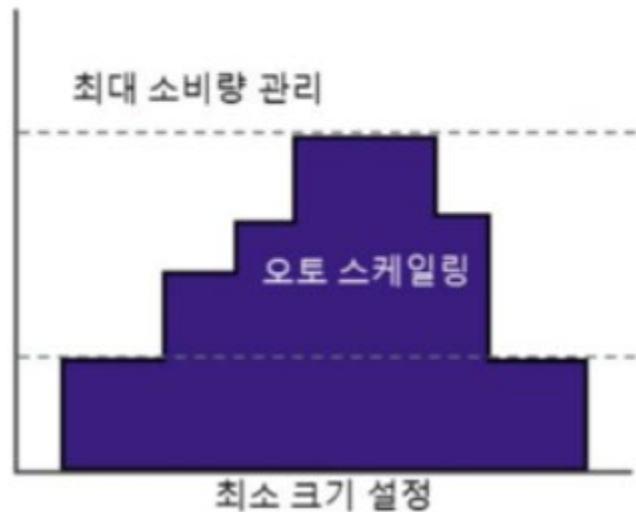
DynamoDB 는 장바구니내 항목에 대한 유연한 스키마를 제공  
조건을 사용하여 낙관적 동시성 제어를 구현함으로써 장바구니에서  
항목을 추가 또는 제거할 때 데이터 일관성을 보장함

- 프로비저닝 용량을 추측할 필요 없음
- 수요가 감소하면 용량을 감소
- 비용 절감
- 스케일링 활동에 대한 전체 가시성 확보

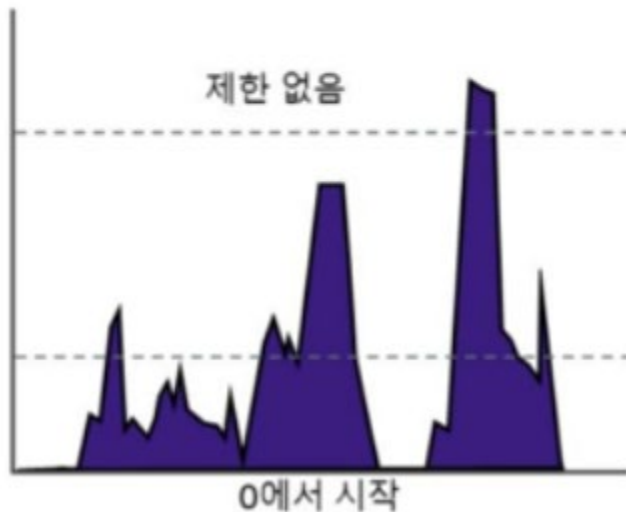


용량 프로비저닝시 추측 불필요(관리형 자동기능)

프로비저닝

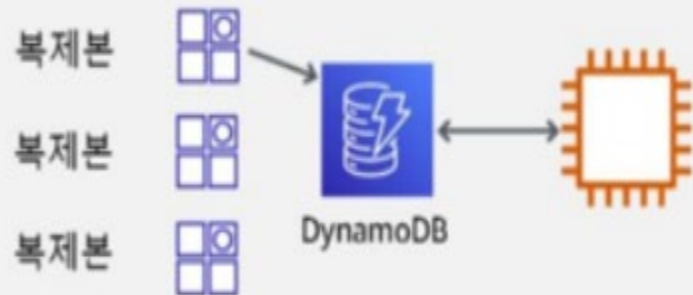


온디맨드



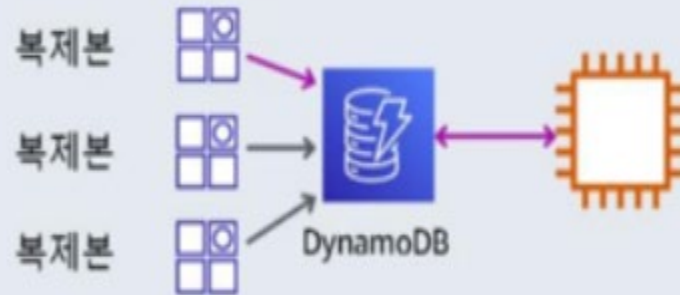
오토스케일링 규칙은 테이블사용을 모니터링하고 EC2 인스턴스와 유사하게 스케일링함 스케일링은 몇분정도 소요됨  
워크로드 예측이 어려운 경우 DynamoDB에 온디맨드스케일링을 사용을 권장

## 최종 일관성



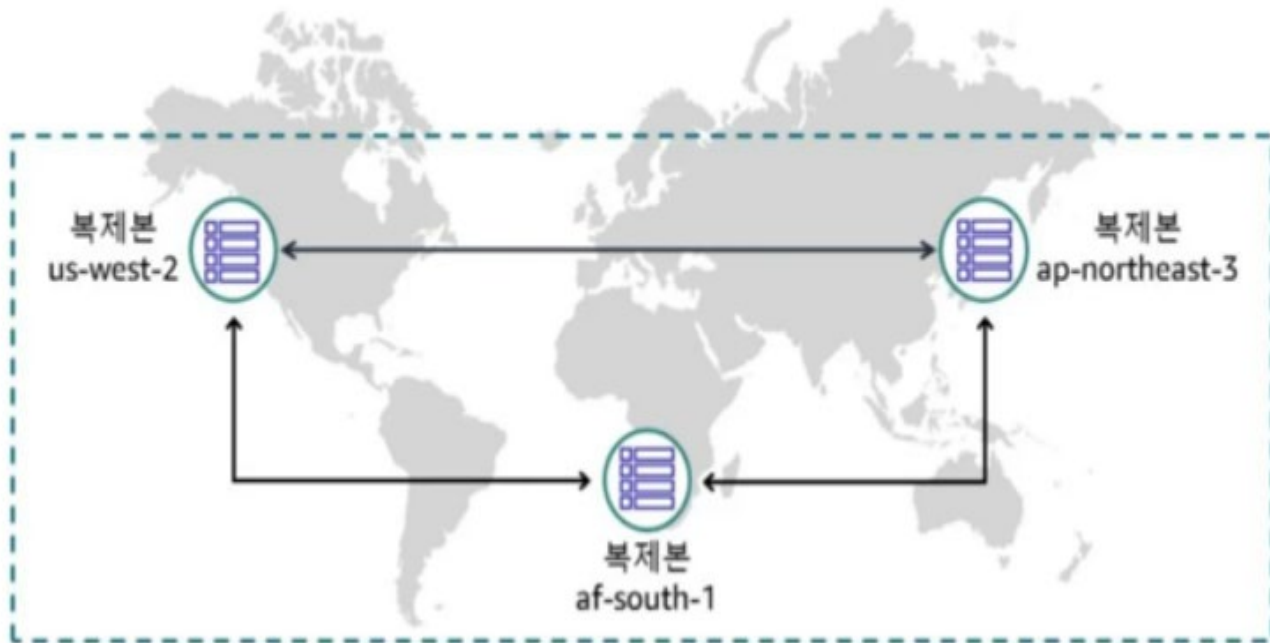
읽기 용량 단위 0.5 사용

## 강력한 일관성



읽기 용량 단위 1 사용

## 글로벌 테이블은 리전 간 복제를 자동화



### 글로벌테이블

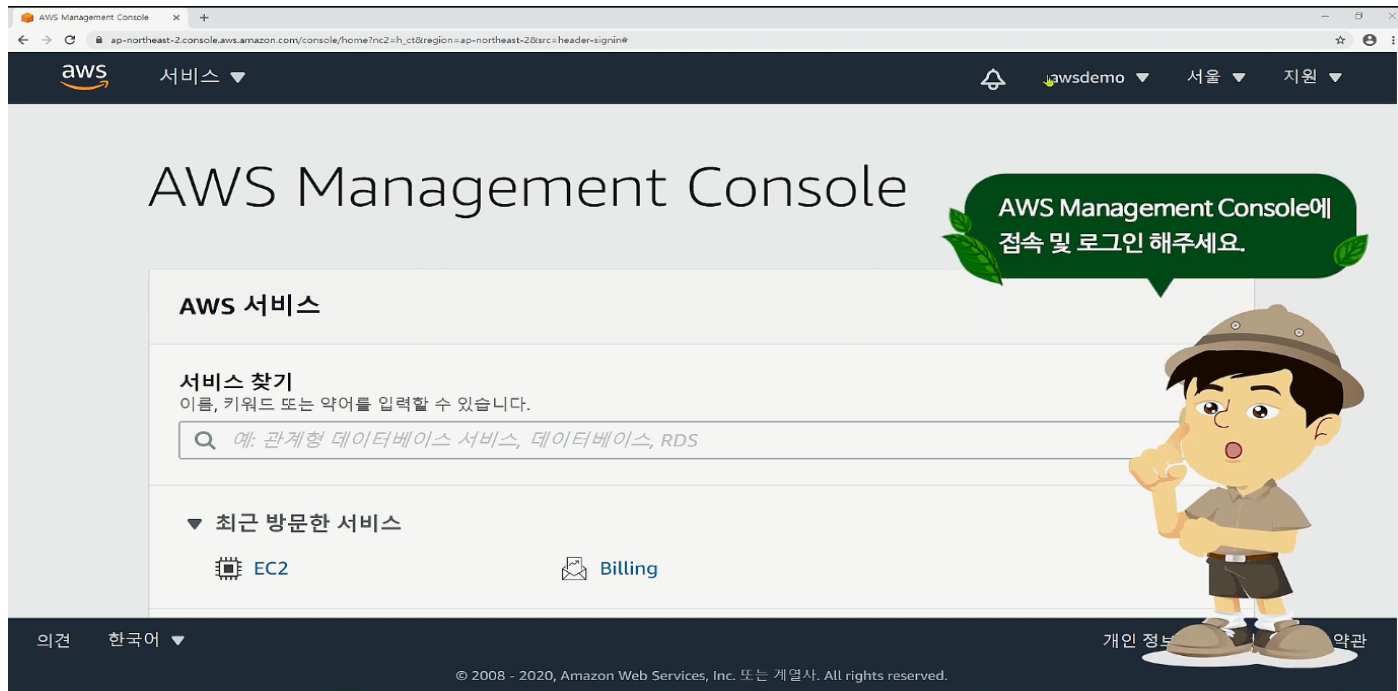
- 단일AWS 계정이 소유하고 복제테이블로 식별되는 한개 이상의 DynamoDB 테이블의 모음

### 복제본

- 글로벌테이블의 일부로 기능하는 단일DynamoDB 테이블

## 다이나모 DB 서비스 실습 시연

<https://rumble.com/v33ukz4-aws-dynamodb-2023.html>



Q1

Amazon RDS를 다중 AZ에서 사용하는 장점은 무엇인가?

- 가.동시에 실행되는 DB 라이브 복사본 2개를 제공
- 나.여러 가용영역에 걸쳐 자동장애 조치를 제공
- 다.자동 교차 리전 복제를 제공
- 라.읽기전용 복제본이 불필요

A

나.여러 가용영역에 걸쳐 자동장애 조치를 제공

Q2

다음 중 NoSQL 데이터베이스의 특징을 설명한 것 중 잘못된것은?

- 가.빅데이터 기술과 더불어 많이 알려진 데이터베이스 기술
- 나.하나의 논리적 데이터베이스가 통신 네트워크로 연결되어 강력한 단일서버에 집중되어 저장되어 관리되는 DB
- 다.각각의 컴퓨터에 local DBMS와 분산 DBMS를 내장
- 라.NOSQL은 빅데이터 시스템이 분산시스템에서 구축되기 때문에 필수로 활용

A

나  
강력한 단일서버에 집중되어 저장되는 것이 아닌 여러대의 컴퓨터에 분산되어 저장되어 관리되는 DB이다



이상희 교사

# AWS를 활용한 웹 서비스 구성하기

## 4주차 강의

1차수: DB서비스(RDS)

2차수: DB서비스(DynamoDB)

▶ 3차수: 서버리스 서비스

## 3차수

### 학습목표

- AWS 서비스중 Serverless의 개념과 Ramda 이용사례를 학습한다

## 3차수

### 학습내용

- 서버리스란?
- 용어정리
- AWS서버리스 포트폴리오
- 서버리스 아키텍처
- 서버리스환경을 고려한 DB 운영방안에 따른 관리영역
- AWS Rambda(PaaS)



## ✓ 서버리스

클라우드를 이용하여 더욱 신속하게 애플리케이션을 구축하는데 사용할수있는 서비스를 의미

## ✓ 장점

인프라를 혁신하고 변화에 신속히 대응 가능

AWS에서 인프라관리작업(용량 프로비저닝 및 패칭)을 처리함으로 코드 작성에 역량 집중



- 서버리스(Serverless) 종류
  - BaaS (Backend as a Service)
  - FaaS (Function as a Service)
  - PaaS 와의 차이점
    - PaaS의 경우, 전체 애플리케이션을 배포하면 특정 서버에서 애플리케이션이 24시간동안 계속 동작함
    - 반면 FaaS 는, 애플리케이션이 아닌 함수를 배포하며, 계속 실행되고 있는 것이 아닌, 특정 이벤트가 발생했을 때만 실행되며, 실행이 되었다가 작업을 마치면 (혹은 최대 타임아웃 시간을 지나면) 종료됨

## 컴퓨팅



Lambda



Fargate

## API 프록시



API  
Gateway



AWS  
AppSync

## 스토리지



Amazon S3

## 데이터베이스



DynamoDB



Aurora

## 프로세스 간 메시징



Amazon SNS



Amazon SQS

## 오케스트레이션



Step Functions

## 분석



Kinesis



Amazon  
Athena

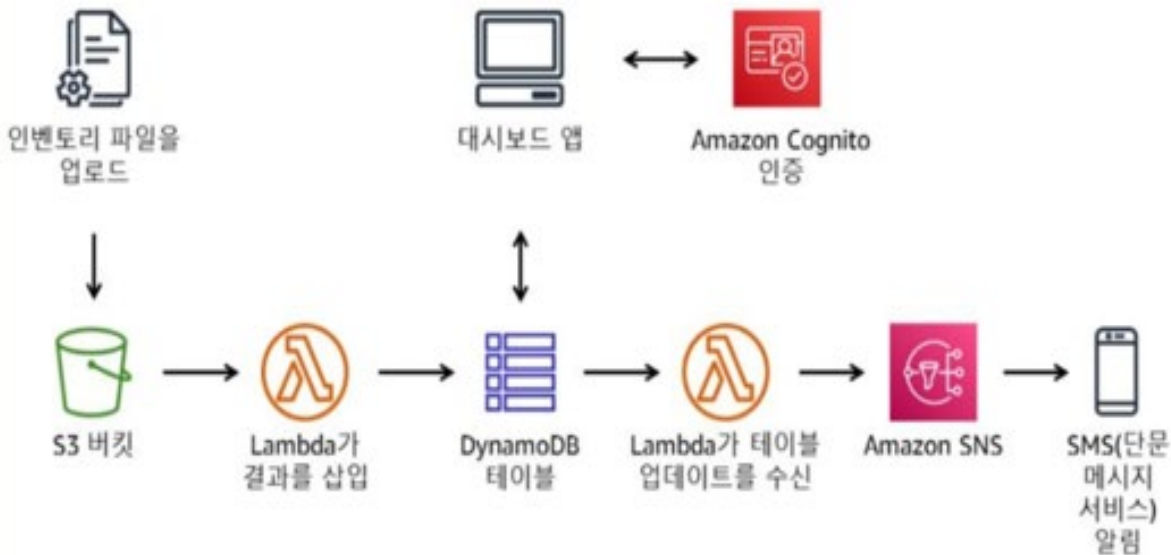
## 개발자 도구



프레임워크, SDK  
및 라이브러리

서버리스애플리케이션을 사용하면 컴퓨팅, 데이터베이스,  
스토리지, 스트림처리, 메시지대기열등의백엔드구성 요소를위한  
서버를 프로비저닝, 유지및 관리 필요성 제거  
애플리케이션의 내결함성 및 가용성에 구현에 대한 우려제거

- 유지 관리가 아닌 개발에 집중합니다.
- 소결합되고 확장성이 뛰어난 워크로드를 생성합니다.
- 총 소유 비용을 절감합니다.



클라우드기반 배포

애플리케이션이 클라우드내에 완전배포됨

하이브리드배포

클라우드기반 리소스와 온프레미스 리소스간에 인프라와 애플리케이션이 연결

서버리스 솔루션 이용시 인프라 관리부담이 제로

## Self Managed (On-Premise)



## DBs on Amazon EC2



## Fully Managed (AWS DBs)



AWS가 제공하는 서버리스 FaaS 솔루션으로 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스

## ✓ AWS Lambda의 특징

서버를 프로비저닝하거나 관리할 필요 없이 작성한 코드를 백엔드 서비스로서 배포할 수 있게 함

Lambda 함수를 실행

애플리케이션 또는 백엔드 서비스의 코드를 작성후 이벤트 트리거만 정의하면 됨

이벤트 트리거의 사례

- Amazon S3 업로드

- DynamoDB 업데이터

- API 게이트웨이 요청 등

이벤트 주도 아키텍처(Event Driven Architecture)를 구성가능

## ✓ AWS Lambda의 단점

24시간 내내 돌아가고 있는 상태가 아니고, 요청이 올 때 AWS가 Lambda를 깨우는데 시간을 사용하기 때문에 응답의 속도에 차이가 존재

요청이 적을 때는 의미가 없지만 요청이 많을 때는 응답 속도에 영향을 줌

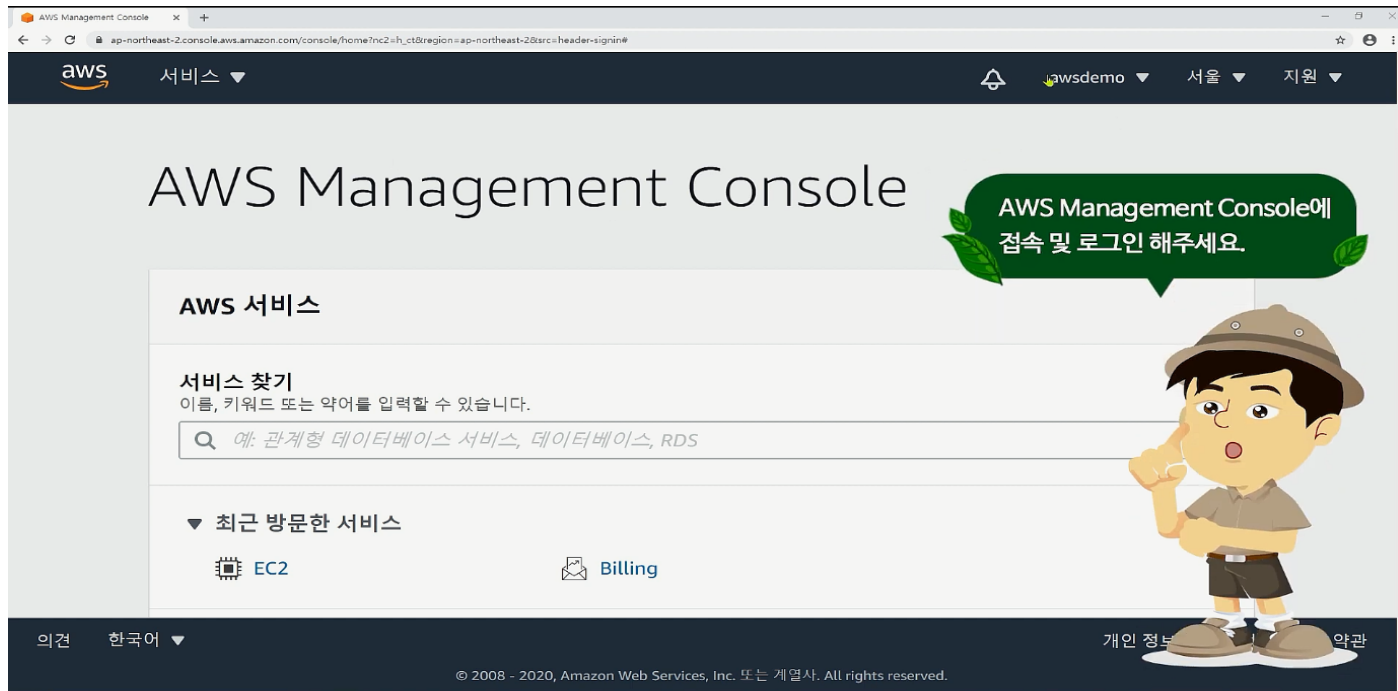


신속히 Scale-Up해야 하고 수요부재시 리소스 양을 0으로  
Scale-down해야 하는 애플리케이션 운영시나리오에 적용권  
장

- ✓ 파일처리
- ✓ 스트림처리
- ✓ 웹 애플리케이션의 탄력적 운영
- ✓ IoT 백엔드 서비스 지원(ex:API요청처리)
- ✓ 모바일백엔드 구축 및 요청처리

# Serverless서비스 (Rambda실습)

<https://rumble.com/v3unks3-aws-serverless-2023.html>



Q1

다음 중 서버리스 컴퓨팅과 직접적으로 상관없는 요소는?

가. BaaS

나. PaaS

다. FaaS

라. Lambda

A

나 PaaS

Q2

다음 중 람다를 사용해야하는 사례가 아닌 것을 골라보시오

가. 파일처리

나. 스트림처리

다. 웹 애플리케이션의 탄력적 운영

라. Aws사용요금의 청구 자동화

A

라. Aws사용요금의 청구 자동화는 람다 사용사례와 상관없다