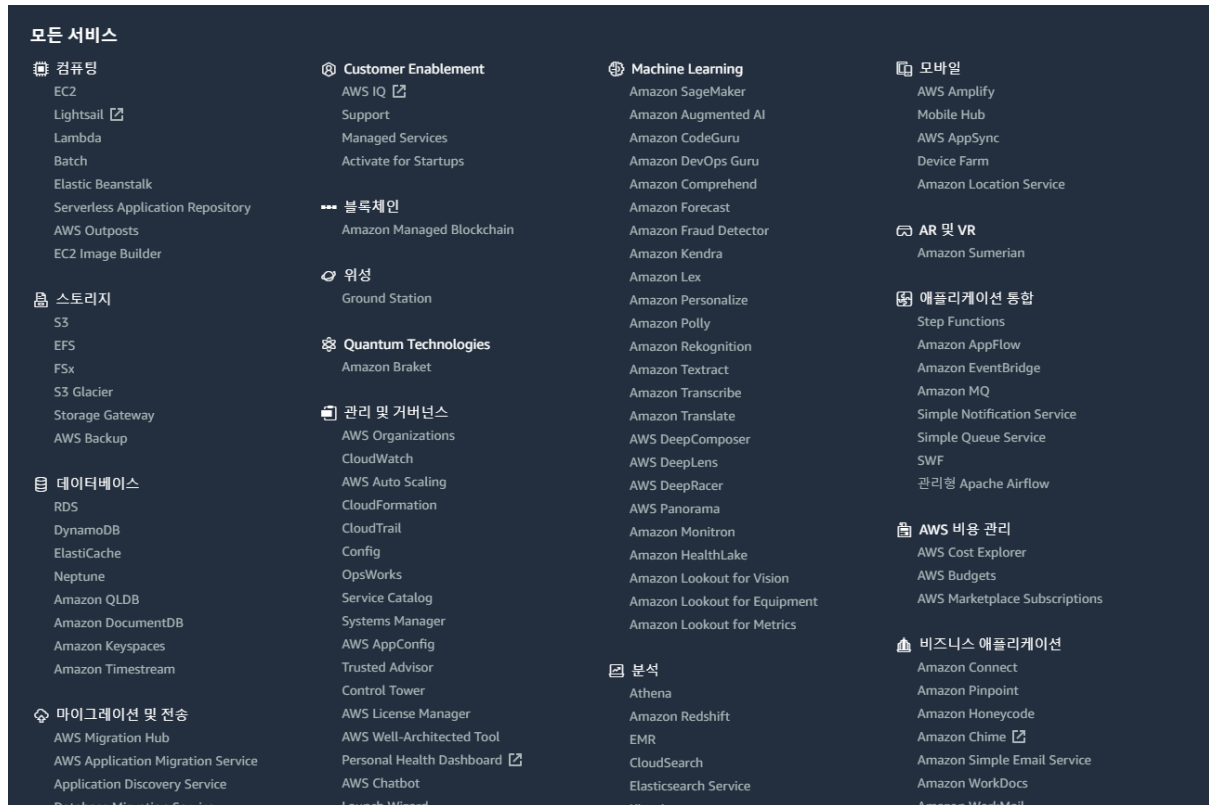


2021 Cloud IoT Services Assignment #2

[이름 : 유지원 / 학번 : 201812627]

Task #1 : 현재 제공되는 **AWS Service Solution**들에 대해서 조사하기



- AWS내부에는 Computing, AR&VR, BlockChain, Quantum Technologies, Database, Migration 및 전송, IOT, 게임 개발 등의 서비스들이 존재한다.

1) Computing

: IT 리소스를 인터넷을 통해 On Demand 로 제공하고 사용한 만큼만 비용을 지불하는 것. 물리적 데이터센터와 서버를 구입, 소유, 유지관리 해야했던 과거와는 달리 단순히 서비스 비용만 제공하면 컴퓨팅 파워, storage, database 등에 액세스 할 수 있다.

1-1) Computing의 장점

- 민첩성 : 광범위한 기술에 쉽게 액세스 가능하며 컴퓨팅, 스토리지 및 데이터베이스에서 사용 가능하다. 또한 사물 인터넷, 기계 학습, 데이터 레이크 및 분석 등의 리소스 등을 빠르게 구동 가능하다.

- 탄력성 : 오버 프로비저닝이 필요 없다. 실제로 필요한 만큼만 가상 서버를 이용하여 리소스를 프로비저닝할 수 있다. 이를 이용하여 비즈니스 요구 사항의 변화에 따른 탄력적 용량의 운용이 가능하다.
- 비용 절감 : 데이터 센터, 물리적 서버 구축 비용 등 절감 가능하며, 사용한 만큼만 비용을 지불하면 된다.

1-2) Computing의 유형













- IaaS(InfraStructure as a Service)
: 클라우드 IT 를 위한 기본 빌딩 블록이 포함되어 있으며 일반적으로 네트워크 기능, 컴퓨터 및 데이터 스토리지 공간에 대한 액세스를 제공한다.
- PaaS(Platform as a Service)
: 기본 인프라를 관리할 필요가 없으며, 리소스 구매, 용량 계획, 소프트웨어 유지 관리, 패치 작업 등에 대한 부담이 줄어든다.
- SaaS(Software as a Service)
: 웹 기반의 이메일과 같은 최종 사용자 어플리케이션으로 특정 소프트웨어를 어떻게 사용할지만 생각하면 됨

2) Database

: AWS 내부에서 지원하는 database서비스이다.

2-1) Database 장점

- 시간 및 비용 절감
- 성능 개선 및 규모 확장 : 마이크로서비스 아키텍처를 사용하여 최신 글로벌 분산 어플리케이션의 확장성, 성능 및 가용성 요구사항을 지원하도록 설계되었다.

데이터베이스 유형	사용 사례	AWS 서비스
관계형	기존 애플리케이션, ERP, CRM, 전자 상거래	 Amazon Aurora  Amazon RDS  Amazon Redshift
키-값	높은 트래픽의 웹 앱, 전자 상거래 시스템, 게임 애플리케이션	 Amazon DynamoDB
인 메모리	캐싱, 세션 관리, 게임 순위표, 지리 공간 애플리케이션	 Amazon ElastiCache for Memcached  Amazon ElastiCache for Redis
문서	콘텐츠 관리, 카탈로그, 사용자 프로필	 Amazon DocumentDB(MongoDB 호환)
와이드 컬럼	장비 관리, 플릿 관리 및 경로 최적화에 사용하는 대규모 산업용 앱	 Apache Cassandra  Amazon Keyspace
그래프	부정 탐지, 소셜 네트워킹, 추천 엔진	 Amazon Neptune
시계열	IoT 애플리케이션, DevOps, 산업용 텔레메트리	 Amazon Timestream
원장	레코드 시스템, 공급망, 등록, 은행 거래	 Amazon QLDB

3) 게임 개발

: 클라우드에서 게임 구축이 가능하다.

3-1) 게임 개발의 장점

- 가동 중단 시간 감소 및 비용 절감이 되며 생산성 및 효율성이 향상된다.
- 플레이어 생애 가치(LTV) 확장되고 신뢰할 수 있는 파트너를 통해 혁신 지원이 가능하다.

3-2) 사용 예시

ROVIO	CAPCOM	GAMELOFT
 <p>Rovio는 AWS의 기계 학습을 통한 예측을 기반으로 플레이어에게 완벽한 수준의 재미를 선사합니다.</p>	 <p>CAPCOM은 컨테이너, 데이터 및 기계 학습을 사용하여 재미있는 게임을 빠르게 구축합니다.</p>	 <p>Gameloft가 출시한 레고 레거시: 상자 속의 영웅들은 모두 AWS 클라우드에서 구축되었습니다.</p>

4) Migration&Transfer

: 축적되어온 데이터들을 바탕으로 Migration 을 단순화하고 가속화하는데 도움이 되는 포괄적인 자사 및 타사 도구 및 서비스 세트를 개발한다. 또한 투자가 원하는 비즈니스 성과를 달성할 수 있도록 보장하는 End to End 도구 세트도 포함된다.

4-1) Migration & Transfer 장점

- 31% 평균적인 인프라 비용 절감
- 62% 보다 효율적인 IT 인프라 관리
- 양자 기계 학습
- 69% 계획되지 않은 다운타임 감소
- 43% 연간 보안 사고 감소

5) IOT

: 다양한 곳에서 사용되는 디바이스들은 이제 서로를 연결하여 데이터를 수집하고 저장하고 분석할 수 있는 솔루션이 필요하므로 이러한 해결책을 위해 **AWS**가 제공하는 서비스이다.

5-1) IOT services 종류

- **Device Software**

- FreeRTOS

- : 소형 저전력 엣지 디바이스를 쉽게 프로그래밍, 배포, 보호, 연결 및 관리할 수 있는 마이크로컨트롤러용 운영체제

- AWS IoT Greengrass

- : 커넥티드 디바이스에서 로컬 컴퓨팅, 메시징, 데이터 캐싱, 동기화 및 기계학습 추론 기능을 안전한 방식으로 실행할 수 있는 소프트웨어

- 연결성 및 제어 서비스

- AWS IoT Core

- : 커넥티드 디바이스가 클라우드 애플리케이션 및 다른 디바이스와 쉽고 안전하게 상호작용할 수 있도록 지원

- AWS IoT Device Defender

- : 지속적으로 IoT 구성을 모니터링하고 감사하여 보안 모범 사례를 준수하도록 함

- AWS IoT Device Management

- : 대규모 디바이스들을 안전하게 관리할 수 있도록 도와줌

- 분석 서비스

- AWS IoT Analytics

- : 방대한 데이터에서 정교한 분석 가능

- AWS IoT SiteWise

- : 대규모 산업 데이터를 쉽게 수집 및 분석 가능

- AWS IoT Events

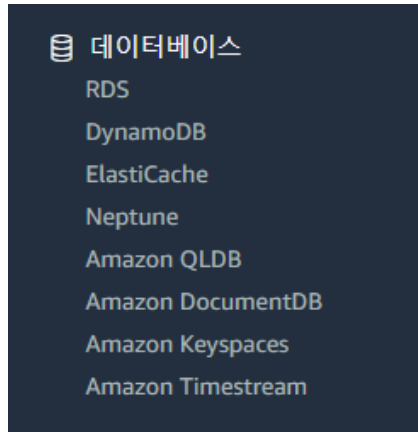
- : 수많은 센서와 애플리케이션에서 이벤트를 손쉽게 감지 가능

- AWS IoT Things Graph

- : 디바이스들과 클라우드를 연결하여 IoT Application 을 손쉽게 구축 가능

Task #2 : AWS Service Solution 조사

<Database>



1) RDS(Relational Database Service)

: RDS란 인프라 및 데이터베이스 업데이트 관리 뿐 아니라 까다로운 관계형 데이터베이스 설치, 운영 그리고 관리를 지원하는 시스템이다.

1-1) EC2 내부에 MySQL과 같은 프로그램을 설치해서 사용해도 되는데 그렇게 하지 않는 이유는 무엇일까?







- EC2에 바로 데이터베이스를 설치해서 사용하는 것이 비용만 놓고 비교해 보았을 때 확연히 저렴한 것을 알 수 있다. 하지만 RDS와 EC2직접 설치 중 하나를 선택해야 하는 개발자는 잠재적 기회 비용과 안정성을 함께 고려해야 한다. RDS를 사용한다면 프로젝트의 규모가 커졌을 때 **Striping**을 최적화하는데 노력과 시간을 쓸 필요가 없으며, 안정성 또한 보장된다. 추가적으로 데이터베이스 스냅샷, 복제본 등을 만들 수 있고 백업에 용이하다.

1-2) RDS 인스턴스의 “Multi-AZ” 옵션

해당 옵션을 사용한다면 다른 가용영역(Availability Zone)에 복제된 데이터베이스를 필요할 때 바로 사용할 수 있다. 즉 한쪽 가용영역에서 장애 및 에러가 발생할 경우, 문제에 영향을 받지 않게 독립된 다른 가용영역에 미리 복제된 데이터베이스를 자동으로 사용해서 복구 시키는 기능이다.

1-3) 서비스

- mysql 등 다양한 데이터베이스 서비스들 중 하나를 선택해서 사용 가능하다.

<input checked="" type="radio"/> Amazon Aurora 	<input type="radio"/> MySQL 	<input type="radio"/> MariaDB 
<input type="radio"/> PostgreSQL 	<input type="radio"/> Oracle 	<input type="radio"/> Microsoft SQL Server 

1-4) 퍼블릭 액세스 가능성

퍼블릭 액세스 가능 정보

- ☐ 예
VPC 외부의 Amazon EC2 인스턴스 및 디바이스는 데이터베이스에 연결할 수 있습니다. 데이터베이스에 연결할 수 있는 VPC 내부의 EC2 인스턴스 및 디바이스를 지정하는 하나 이상의 VPC 보안 그룹을 선택하십시오.
- ☒ 아니요
RDS는 데이터베이스에 퍼블릭 IP 주소를 할당하지 않습니다. VPC 내부의 Amazon EC2 인스턴스 및 디바이스만 데이터베이스에 연결할 수 있습니다.

예 – 전국에서 db주소만으로 접속 가능하다.

아니요 - VPC외부의 EC2인스턴스 등은 접속 불가하다.

1-5) 파라미터 그룹

ex) 데이터베이스가 한글 처리를 할 수 있도록 할 수 있다.

파라미터 그룹 생성

파라미터 그룹 세부 정보

파라미터 그룹을 생성하려면 파라미터 그룹 패밀리를 선택한 다음 파라미터 그룹의 이름을 지정하고 설명하십시오.

파라미터 그룹 패밀리
이 DB 파라미터 그룹이 적용될 DB 패밀리입니다.

mysql5.6

그룹 이름
DB 파라미터 그룹에 대한 식별자입니다.

Hangul

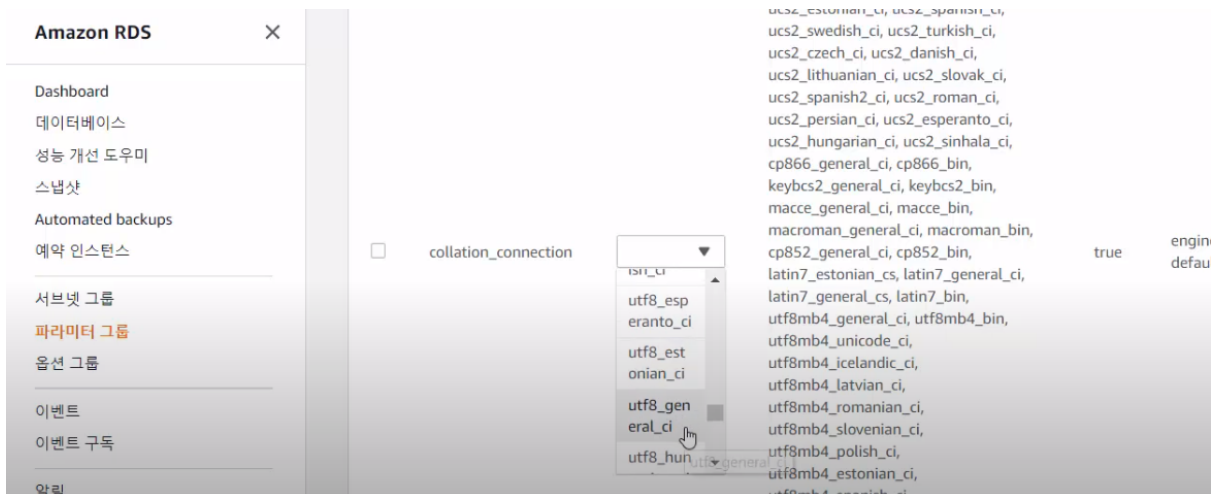
설명
DB 파라미터 그룹에 대한 설명입니다.

[취소](#) [생성](#)

> 파라미터 그룹 생성 > 파라미터 편집

> character를 검색 후 UTF-8로 설정,

> Collation_connection Collation_server 검색 후 utf8_general_ci 로 변경 후 저장



2) DynamoDB

: RDBMS가 아닌 **nosql** 데이터베이스이다. 하지만 기본키를 가지고 있으며 **indexing**과 **key** 설정이 매우 중요하다.

* **nosql** : 가용성과 확장성이 높고 고성능에 최적화된 비관계형 데이터베이스 시스템. 관계형 데이터 베이스와 다르게 데이터들의 저장에 초점을 맞추어 관계를 생각하지 않는다.

2-1) DynamoDB의 장점

- Full-managed

: 클러스터링, 백업정책, 성능상향, 다중**region**을 지원한다. 또한 **AWS**에서 지원하는 **serverless DB**이므로 운영 중 발생하는 장애를 **AWS support**의 도움으로 해결할 수 있다.

- 모든 데이터가 **SSD**에 저장되어 입출력 속도가 빠르다.

- **HTTP** 비연결형 통신으로 데이터를 가져와 **connection pool** 관리가 필요 없다.

- 파티션 관리가 백그라운드에서 이루어지며 사용자가 이를 관리할 필요는 없다.

2-2) DynamoDB의 단점

- 분량이 큰 **Rest API**를 사용해야 할 때 **nosql** 쿼리 코드를 관리하는 관리 전략이 필수 요소가 된다. 또한, 여러 쿼리에 대해 일관성, 원자성을 보장하지 않는다.

2-3) 테이블 생성

DynamoDB 테이블 만들기

[자습서](#)

DynamoDB는 스키마가 없는 데이터베이스로서, 테이블 이름과 기본 키만 필요로 합니다. 테이블의 기본 키는 각 파티션에서 항목을 고유하게 식별하고, 데이터를 분할하며, 데이터를 정렬하는 한 개 또는 여러 개의 속성으로 구성됩니다.

테이블 이름*

기본 키* 파티션 키

문자열

☐ 정렬 키 추가

테이블 설정

기본 설정을 사용하면 테이블을 가장 빠르게 시작할 수 있습니다. 이러한 기본 설정은 지금 또는 테이블을 만든 후에 수정할 수 있습니다.

- ☒ 기본 설정 사용
- 보조 인덱스 없음
 - 프로비저닝된 용량은 읽기 5개와 쓰기 5개로 설정
 - 기본 알람은 SNS 주제 "dynamodb"를 사용하여 80% 상한 설정
 - 유류 데이터를 기본 암호화 유형으로 암호화합니다.

기본적으로 Auto Scaling을 활성화하기 위해 필요한 권한이 없습니다.
설명서 [을\(를\)](#) 참조하십시오.

- 태그 추가 (1) [신규 기능!](#)

키	값
name	Jiwon
키 추가	비어 있는 값

테이블 만들기

테이블 삭제

테이블 이름으로 필터링

테이블 그룹 선택

작업

이름

tutorial

tutorial

닫기

개요

정책

구분지

알림

용량

인덱스

글로벌 테이블

백업

Contributor Insights

트래거

액세스 제어

태그

테이블 만들기 중

최근 알림

이 테이블에서 트리거된 CloudWatch 알림이 없습니다.

Kinesis 데이터 스트림 세부 정보

DynamoDB를 Amazon Kinesis Data Streams를 사용하여 테이블의 항목 수를 변경 사항을 Kinesis 데이터 스트림으로 합치합니다. [자세히 알아보기](#)

스트림 활성화됨

스트림 이름

Kinesis로의 스트리밍 관리

DynamoDB 스트림 세부 정보

스트림 활성화됨

보기 유형

최근 스트림 ARN

DynamoDB 스트림 관리

테이블 세부 정보

테이블 이름

기본 파티션 키

기본 정렬 키

특정 시점으로 복구

암호화 유형

KMS 마스터 키 ARN

암호화 상태

CloudWatch Contributor Insights

TTL(Time to live) 속성

테이블 상태

생성 중

태그

tutorial

key_ (문자열)

-

기본값

기본값

해당 사항 없음

-

선택

-

생성 중

2021년 5월 6일 오전 1시 9분 36초 UTC+9

3) ElasticCache

: 인 메모리 캐시를 간편하게 배포, 운영할 수 있도록 도와주는 **AWS** 서비스

3-1) ElasticCache 사용 이유

- 처리량이 많고 지연 시간이 짧은 인 메모리 데이터 스토어에서 데이터를 검색하여 데이터 집약적 앱을 구축하거나 기존 **DB** 성능을 강화할 수 있기 때문에 사용한다.
- 때문에 캐싱, 세션 스토어, 게이밍, 지리공간 서비스, 실시간 분석 및 대기열과 같은 실시간 사용 사례에 선택된다.

3-2) ElasticCache의 장점

- 탁월한 성능 : 1밀리초 미만의 응답 시간이 필요한 까다로운 어플리케이션을 지원할 수 있도록 인 메모리 데이터 스토어 및 캐시의 역할을 한다.
- 완전관리형 : **Amazon ElasticCache**에서 클러스터를 지속적으로 모니터링 하므로 개발자는 소프트웨어 패치, 설정, 구성, 모니터링, 장애 복구 등을 신경 쓰지 않아도 된다.
- 확장성 : 프로젝트의 규모가 커졌을 때 해당 수요에 맞춰 자동으로 **Scaleout, in, up** 된다.

The screenshot shows the AWS ElasticCache console. On the left is a navigation menu with links like Memcached, Redis, and ElasticCache clusters. The main area has the ElasticCache logo and a description: 'ElasticCache는 클라우드상의 분산된 인 메모리 캐시를 손쉽게 시작, 관리 및 확장할 수 있는 웹 서비스입니다.' Below this are three large buttons: 'Cluster 시작' (Create Cluster), '연결' (Connect), and '관리' (Manage). Each button has an icon and a brief description of the action. Below each button are links to documentation for Memcached and Redis.

ElasticCache 개요

Memcached
Redis
클라우드 데이터스토어
서비스 업데이트
예약 노드
백업
과라미터 그룹
사용자 관리
사용자 그룹 관리
서브넷 그룹
이벤트
ElasticCache 클러스터 클라이언트

ElasticCache
ElasticCache는 클라우드상의 분산된 인 메모리 캐시를 손쉽게 시작, 관리 및 확장할 수 있는 웹 서비스입니다.

[지금 시작](#)

클러스터 시작
몇 번의 클릭만으로 클러스터를 생성할 수 있습니다. 이 클러스터의 엔진 소프트웨어로 Memcached 또는 Redis 중 선택할 수 있습니다.
[설명서 링크\(Memcached\)](#)
[설명서 링크\(Redis\)](#)

연결
클러스터에 대한 액세스 권한을 부여받고 클러스터가 사용 가능한 상태가 되면, EC2 인스턴스에 로그인하여 클러스터의 노드에 연결할 수 있습니다.
[설명서 링크\(Memcached\)](#)
[설명서 링크\(Redis\)](#)

관리
AWS Management Console을 사용하여 손쉽게 리소스를 추가하고, 구성을 수정하고, ElasticCache 환경의 노드를 모니터링하여 애플리케이션 요구 사항을 충족할 수 있습니다.
[설명서 링크\(Memcached\)](#)
[설명서 링크\(Redis\)](#)

4) Neptune

: 빠르고 안정적인 그래프 데이터베이스 서비스로, 수십억 개의 관계를 저장하고 몇 밀리초의 지연 시간으로 그래프를 쿼리하는데 최적화된, 특수목적의 고성능 그래프 데이터베이스 엔진이다.

4-1) Neptune 장점

- 읽기 전용 복제본, 특정 시점으로 복구, Amazon S3으로 연속 백업, 가용 영역 간 복제 등의 기능을 갖추고 있다.
- 암호화를 지원하여 안전하다.
- 종합 관리형 솔루션이기 때문에 개발자가 하드웨어, 소프트웨어 패치, 설정, 구성, 백업과 같은 데이터베이스 관리 작업을 염려할 필요 없다.

Neptune > 데이터베이스 생성

데이터베이스 생성

엔진 옵션

엔진 유형
neptune

버전 정보
Neptune 1.0.4.1.R4 ▼

설정

DB 클러스터 식별자 [정보](#)
DB 클러스터 이름을 입력합니다. 이름은 현재 AWS 리전에서 AWS 계정이 소유하는 모든 DB 클러스터에 대해 고유해야 합니다.
database-1

DB 클러스터 식별자는 대소문자를 구분하지 않지만 모두 소문자로 저장됩니다(예: "mydbcluster"). 제약 조건: 1~60자의 영숫자 또는 하이픈으로 구성되어야 합니다. 첫 번째 문자는 글자여야 합니다. 하이픈은 연속으로 2개를 포함할 수 없습니다. 끝에 하이픈이 올 수 없습니다.

템플릿
귀하의 사용 사례를 충족하는 템플릿을 선택하십시오.

☒ 프로덕션
고가용성 및 빠르고 일관된 성능을 위해 기본값을 사용하십시오.

☐ 개발 및 테스트
이 인스턴스는 프로덕션 환경 외부에서 개발 용도로 마련되었습니다.

Neptune > 데이터베이스

데이터베이스 그룹 리소스 수정 작업 ▼ 데이터베이스 생성

Q 데이터베이스 필터

	DB 식별자 ▲	역할 ▼	엔진 ▼	리전 및 AZ ▼	크기 ▼	상태 ▼	CPU	현재 활동	유지 관리 ▼	VPC ▼	다중 AZ ▼
<input type="radio"/>	<input type="checkbox"/> database-1	클러스터	Neptune	ap-northeast-2	-	⌚ 생성 중	-	-	-	-	-
<input type="radio"/>	<input type="checkbox"/> database-1-instance-1	읽기	Neptune	ap-northeast-2a	db.r5.xlarge	⌚ 생성 중	-	-	-	vpc-2057da4b	2 영역
<input type="radio"/>	<input type="checkbox"/> database-1-instance-1-ap-northeast-2d	읽기	Neptune	ap-northeast-2d	db.r5.xlarge	⌚ 생성 중	-	-	-	vpc-2057da4b	2 영역

5) Amazone QLDB

: 완전 관리형 원장 데이터베이스로, 중앙의 신뢰할 수 있는 기관이 소유하는 투명하고 변경 불가하며 암호화 방식으로 검증 가능한 트랜잭션 로그를 제공한다. 또한 모든 어플리케이션 변경 내용을 추적하여 완전하고 검증 가능한 시간대별 변경 기록을 유지한다.

5-1) 주요 개념

- 원장 : 완전하고 변경 불가능한 테이블 변경 기록을 유지하는 일련의 **QLDB**테이블 및 저널로 구성된다. 원장에는 이름이 지정되며 태그를 지정할 수 있다.
- 저널(**Journal**) – 저널은 이전 블록에 암호 체인으로 연결되어 변경 사항을 확인할 수 있는 블록 시퀀스로 구성된다. 블록에는 테이블의 실제 변경 사항이 효율적인 검색을 위해 인덱싱된 상태로 포함된다. 이 추가 전용 모델에서는 이전 데이터를 편집하거나 삭제할 수 없으며 원장이 변경 불가능한 상태로 유지된다. 또, **QLDB**에서는 저널의 일부 또는 전체를 **S3**로 내보낼 수 있다.
- 테이블(**Table**) – 테이블은 원장 내에 존재하며 문서 수정 버전의 컬렉션이 포함된다. 테이블은 문서 필드에 대한 선택적 인덱스를 지원하며, 인덱스를 사용하면 등호(=) 조건자를 사용하는 쿼리의 성능을 개선할 수 있다.
- 문서(**Document**) – 문서는 테이블 내에 존재하며 **Amazon Ion** 형태여야 한다.. **QLDB**는 중첩된 **JSON** 요소가 포함된 문서를 지원하며 이러한 요소를 참조하고 포함하는 쿼리를 쓸 수 있는 기능을 제공한다. 문서는 특정 스키마를 준수하지 않아도 되므로 변경 사항에 따라 쉽게 조정 가능한 애플리케이션을 유연하게 구축할 수 있다.
- **PartiQL** – PartiQL은 특정 데이터 원본에 종속되지 않으면서 관계형, 반 구조화 및 중첩된 데이터에 대한 **SQL** 호환 액세스를 지원하는 새로운 개방형 표준 쿼리 언어다.
- 서버리스(**Serverless**) – 개발자는 용량 프로비저닝이나 읽기 및 쓰기 처리량 구성에 대해 걱정할 필요가 없다. 원장을 생성하고 테이블을 정의하면 **QLDB**가 애플리케이션의 요구 사항에 맞게 자동으로 크기 조정을 수행한다.

6) DocumentDB

: MongoDB와 호환된다.

(* MongoDB : JSON 과 유사한 스토리지를 지원하는 비 관계형 문서 데이터베이스로, 구조화되지 않은 데이터를 저장할 수 있는 유연한 데이터 모델이 있으며 풍부하고 직관적인 API 를 통해 전체 인덱싱 지원 및 복제를 제공한다.)

6-1) DocumentDB 장점

- 빠르고 확장 가능하며 가용성이 뛰어난 완전관리형 문서 데이터베이스 서비스이며 JSON data를 쉽게 저장, 쿼리 및 인덱싱 할 수 있다.
- 스토리지와 컴퓨팅이 분리되므로 각각 독립적으로 확장이 가능하다.
- 데이터 규모와 관계없이 몇 분만에 지연 시간이 짧은 읽기 전용 복제본을 최대 15개까지 추가하여 읽기 용량을 초당 수백만건의 요청으로 늘릴 수 있다.
- 3개의 가용 영역에 걸쳐 6개의 데이터 사본을 복제한다.

6-2) 사용 방법

- AWS cloud9 환경 생성 후 보안 그룹을 생성한다.
- Amazon DocumentDB 클러스터를 생성한다.
- mongo shell을 install 하고 Amazon DocumentDB클러스터에 연결한다.
- ex) 단일 문서 삽입, 작성한 문서 읽기의 순서로 쿼리를 수행한다.

7) Amazon Keyspaces

: 고가용성의 확장 가능한 관리형 Apache Cassandra 호환 데이터베이스 서비스이다.

서버를 프로비저닝, 패치 또는 관리할 필요가 없으며 소프트웨어를 설치, 유지관리, 관리 또는 운영해야 할 필요가 없어 개발자가 편리하다. 또한 사실상 무제한의 처리량과 스토리지로 초당 수천건의 요청을 처리하는 애플리케이션을 구축할 수 있다.

7-1) Keyspaces 장점

- 기존에 사용하던 CQL API 코드, Cassandra 드라이버 및 개발자 도구를 사용할 수 있다.
- 테이블을 자동으로 확장 및 축소할 수 있다.
- 모든 규모에서 일관되게 10 밀리초 미만의 응답 시간을 지원한다.
- 테이블은 기본적으로 암호화되고 고가용성을 위해 여러 AWS 가용 영역에서 세 번 복제된다.
- 애플리케이션 성능 저하 없이 수백 테라바이트 데이터가 있는 테이블을 지속적으로 백업하고 이전 35일 이내의 원하는 시점으로 복구 가능하다.

8) Amazon Timestream

: 최대 10배 저렴한 비용으로 1,000배 더 빠르게 매일 수조 건의 이벤트를 쉽게 저장하고 분석할 수 있다. 최신 데이터는 메모리에 저장하고 기록 데이터는 사용자 정의 정책을 기반으로 비용이 최적화된 스토리지에 저장하는 방식을 통해 시계열 데이터 수명 주기 관리 과정에서 시간과 비용을 절약할 수 있다.

8-1) Timestream 장점

- 메모리 저장소를 통해 특정 시점을 빠르게 쿼리할 수 있다.
- 비용 최적화 마그네틱 저장소를 통해 분석 쿼리도 신속하게 실행 가능하다.
- 서버리스 서비스이므로 용량을 프로비저닝 할 필요 없이 애플리케이션 구축 작업만 중점적으로 진행 가능하다.
- 스토리지 티어 전체에서 데이터를 투명하게 액세스하여 결합하므로 사용자가 데이터의 위치를 지정할 필요가 없다.

8-2) 데이터베이스 생성 과정

The screenshot displays the AWS Timestream 'Create database' console. At the top, the AWS logo and '서비스' (Services) are visible. A search bar contains '서비스, 기능, 마켓플레이스 제품, 설명서 검색' (Search services, features, marketplace products, documentation) and a '[Alt+S]' shortcut. The breadcrumb trail shows 'Timestream > Databases > Create database'.

The main heading is 'Create database' with an 'Info' link. Below this is the 'Database configuration' section, which includes the instruction: 'Create and configure a database or create a database with sample data to explore Timestream right away.'

Under 'Choose a configuration', there are two options: 'Standard database' (Create a new database with custom configuration) and 'Sample database' (Create a database and populate it with sample data to get started in a single click). The 'Sample database' option is selected with a blue radio button.

The 'Name' field is labeled 'sampleDB'. A note specifies: 'Specify a name that is unique for all Timestream databases in your AWS account in the current Region. You can not change this name once you create it.' A validation message states: 'Must be between 3 and 64 characters long. Must contain letters, digits, dashes, periods or underscores.'

The 'Tables with sample data (1/2)' section follows, with the instruction: 'Choose one or more sets of sample data to create tables within your sample database. Each sample data set is less than 50 MB.'

Under 'Choose sample data sets', there are two cards. The 'IoT' card is selected (checked checkbox) and contains the description: 'Contains IoT sensor data from one or more truck fleets to streamline fleet management and to identify cost optimization opportunities such as reduction in fuel consumption.' The 'Table name' field for IoT is 'IoT'. The 'DevOps' card is unselected (unchecked checkbox) and contains the description: 'Contains EC2 instance metrics such as CPU, network, and memory utilization to help monitor application performance and availability.' The 'Table name' field for DevOps is 'DevOps'.

At the bottom right, there are two buttons: 'Cancel' and 'Create database' (highlighted in orange).

출처

- <https://waspro.tistory.com/457>
- <https://aws.amazon.com/ko/products/databases/>
- <https://www.bespinglobal.com/bespins-pick-vol-11-aws-rds-vs-ec2/>
- <https://velog.io/@drakejin/DynamoDB에-대해-서-알아보자-2>
- <https://mattnk.tistory.com/13>
- <https://aws.amazon.com/ko/nosql/graph/>
- <https://aws.amazon.com/ko/qldb/>
- <https://aws.amazon.com/ko/blogs/korea/now-available-amazon-quantum-ledger-data-base-qldb/>