Amazon DynamoDB를 사용한 데이터 모델링

AWS 규범적 지침

AWS 규범적 지침: Amazon DynamoDB를 사용한 데이터 모델링

Table of Contents

소개 1
프로세스 플로우 2
RACI 매트릭스 2
프로세스 단계 5
단계 1. 사용 사례 및 논리적 데이터 모델 식별 5
목표 5
프로세스 5
도구 및 리소스 6
RACI 6
결과 6
단계 2. 예비 비용 추정치 생성 6
목표 6
프로세스 6
도구 및 리소스 7
RACI 7
결과 7
단계 3. 데이터 액세스 패턴 파악 7
목표 7
프로세스 7
도구 및 리소스 8
RACI 8
결과 8
예 9
4단계. 기술 요구 사항 파악 9
목표 9
프로세스 9
도구 및 리소스 9
RACI 10
결과 10
5단계. DynamoDB 데이터 모델 생성 10
목표 10
프로세스 10
도구 및 리소스 11
RACI 12

결과 12
예 12
6단계. 데이터 쿼리 생성 13
목표 13
프로세스 13
도구 및 리소스 13
RACI 13
결과 14
예제 14
7단계. 데이터 모델 검증 14
목표 14
프로세스 14
도구 및 리소스 14
RACI 15
결과 15
8단계. 비용 추정 검토 15
목표 15
프로세스 15
도구 및 리소스 15
RACI 16
결과 16
9단계. 데이터 모델 배포 16
목표 16
프로세스 16
도구 및 리소스 16
RACI 16
결과 17
예 17
템플릿 19
비즈니스 요구 사항 평가 템플릿 19
기술 요구 사항 평가 템플릿 22
액세스 패턴 템플릿 26
템플릿 27
모범 사례 30
계층적 데이터 모델링 31
1단계: 사용 사례 및 논리적 데이터 모델 식별 31

2단계: 예비 비용 추정 생성 33
3단계: 데이터 액세스 패턴 파악 34
4단계: 기술 요구 사항 식별 35
5단계: DynamoDB 데이터 모델 생성 35
테이블에 구성 요소 저장 35
GSI1 인덱스 37
GSI2 인덱스 37
6단계: 데이터 쿼리 만들기 38
7단계: 데이터 모델 검증 41
8단계: 예상 비용 검토 42
목표 42
프로세스 43
9단계: 데이터 모델 배포 43
추가 리소스 45
기여자 47
문서 기록 48
용어집 49
..... 49
A 50
B 52
C 54
D 56
E 60
F 62
G 63
H 64
I 65
L 67
M 68
O 71
P 73
Q 75
R 75
S 78
T 81
U 82

V 83
W 83
Z 84
..... lxxxv

Amazon DynamoDB를 사용한 데이터 모델링

프로세스, 템플릿, 모범 사례

아마존 웹 서비스 (AWS)

2023년 12월(문서 기록)

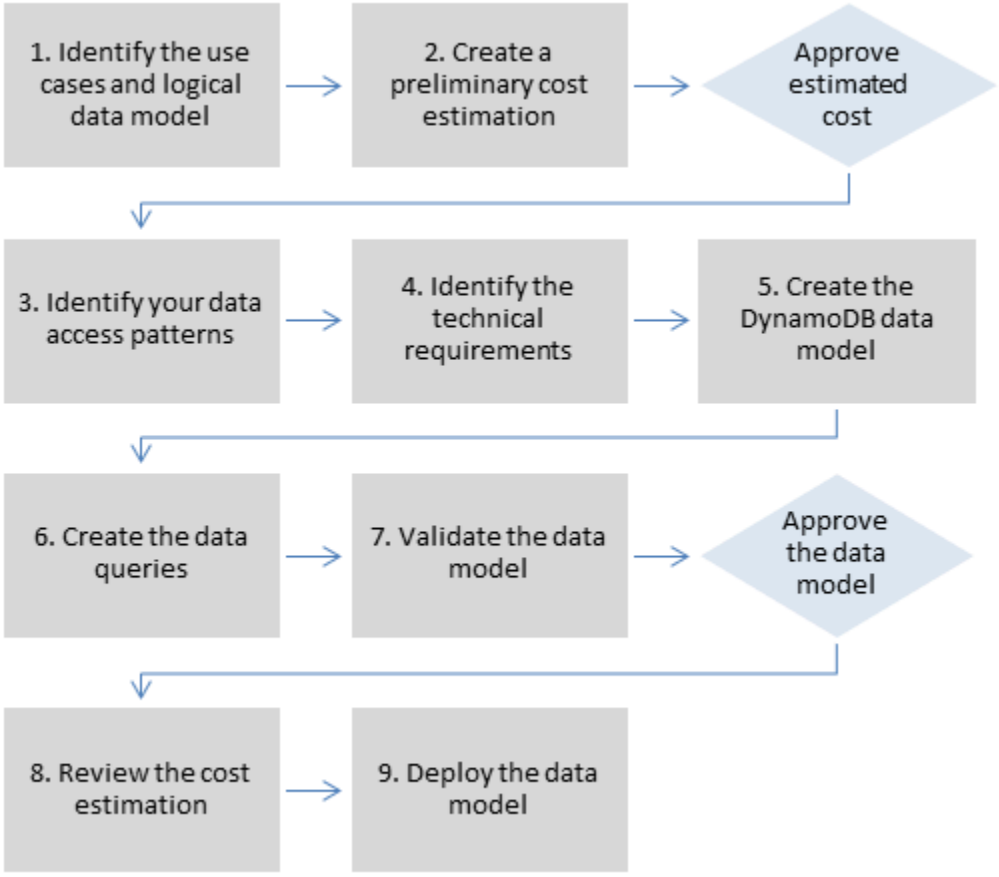
NoSQL 데이터베이스는 최신 애플리케이션을 구축하기 위한 유연한 스키마를 제공합니다. 다양한 규모에 맞는 개발의 용이성, 기능성 및 성능으로 널리 인정받고 있습니다. Amazon DynamoDB는 Amazon 웹 서비스 (AWS) 클라우드의 NoSQL 데이터베이스 서비스에서 원활한 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다. 완전관리형 데이터베이스 서비스인 DynamoDB는 분산 데이터베이스를 운영하고 확장하는 데 따르는 관리 부담을 덜어줍니다. 하드웨어 프로비저닝, 설정 및 구성, 복제, 소프트웨어 패치 또는 클러스터 확장에 대해 걱정할 필요가 없습니다.

NoSQL 스키마 설계에는 기존의 관계형 데이터베이스 관리 시스템(RDBMS)과는 다른 접근 방식이 필요합니다. RDBMS 데이터 모델은 데이터 구조 및 다른 데이터와의 관계에 중점을 둡니다. NoSQL 데이터 모델링은 액세스 패턴 또는 애플리케이션이 데이터를 소비하는 방식에 중점을 두므로 간단한 쿼리 작업을 지원하는 방식으로 데이터를 저장합니다. Microsoft SQL Server 또는 IBM Db2와 같은 RDBMS의 경우 액세스 패턴에 대해 많이 생각하지 않고도 정규화된 데이터 모델을 만들 수 있습니다. 나중에 패턴과 쿼리를 지원하도록 데이터 모델을 확장할 수 있습니다.

이 설명서에서는 기능 요구 사항, 성능, 유효 비용을 제공하는 DynamoDB 사용을 위한 데이터 모델링 프로세스를 제공합니다. 이 가이드는 DynamoDB를 실행 중인 애플리케이션의 운영 데이터베이스로 사용하려는 데이터베이스 엔지니어를 위한 것입니다. AWS AWS 전문 서비스는 권장 프로세스를 사용하여 대기업이 다양한 사용 사례 및 워크로드에 맞는 DynamoDB 데이터 모델링을 사용할 수 있도록 지원했습니다.

데이터 모델링 프로세스 흐름

Amazon DynamoDB를 사용하여 데이터를 모델링할 때는 다음 절차를 수행하는 것을 권장합니다. 이 설명서의 뒷부분에 있는 단원에서 자세히 설명합니다.



RACI 매트릭스

일부 조직에서는 책임 할당 매트릭스(RACI 매트릭스라고도 함)를 사용하여 특정 프로젝트 또는 비즈니스 프로세스에 관련된 다양한 역할을 설명합니다. 이 설명서는 조직이 DynamoDB 데이터 모델링 프로세스에 적합한 사람과 책임을 식별하는 데 도움이 될 수 있는 제안형 RACI 매트릭스를 제공합니다. 프로세스의 각 단계에 대해 이해관계자와 이들의 참여가 나열되어 있습니다.

- R - 단계 완료를 담당함
- A - 작업 승인 및 마무리에 대한 책무를 짐
- C — 작업에 대한 입력을 제공하기 위해 상담함

- I - 진행 상황은 알고 있지만 작업에 직접 관여하지는 않음

조직 및 프로젝트 팀의 구조에 따라 아래의 RACI 매트릭스의 역할을 동일한 이해 관계자가 수행할 수 있습니다. 어떤 경우에는 특정 단계에 대한 담당과 책무가 모두 이해관계자에게 있는 경우도 있습니다. 예를 들어, 데이터베이스 엔지니어는 데이터 모델 생성 및 승인을 모두 담당할 수 있는데, 이는 해당 영역이 데이터 모델의 영역이기 때문입니다.

프로세스 단계	비즈니스 사용자	비즈니스 분석가	솔루션 아키텍처	데이터베이스 엔지니어	애플리케이션 개발자	DevOps 엔지니어
1. 사용 사례 및 논리적 데이터 모델 식별	C	R/A	I	R		
2. 예비 비용 추정치 생성	C	A	I	R		
3. 데이터 액세스 패턴 파악	C	A	I	R		
4. 기술 요구 사항 파악	C	C	A	R		
5. DynamoDB 데이터 모델 생성	I	I	I	R/A		
6. 데이터 쿼리 생성	I	I	I	R/A	R	
7. 데이터 모델 검증	A	R	I	C		

프로세스 단계	비즈니스 사용자	비즈니스 분석가	솔루션 아키텍처	데이터베이스 엔지니어	애플리케이션 개발자	DevOps 엔지니어
8. 비용 추정 검토	C	A	I	R		
9. DynamoDB 데이터 모델 배포	I	I	C	C		R/A

데이터 모델링 프로세스 단계

이 단원에서는 Amazon DynamoDB에 대해 권장하는 데이터 모델링 프로세스의 각 단계를 자세히 설명합니다.

주제

- 단계 1. 사용 사례 및 논리적 데이터 모델 식별
- 단계 2. 예비 비용 추정치 생성
- 단계 3. 데이터 액세스 패턴 파악
- 4단계. 기술 요구 사항 파악
- 5단계. DynamoDB 데이터 모델 생성
- 6단계. 데이터 쿼리 생성
- 7단계. 데이터 모델 검증
- 8단계. 비용 추정 검토
- 9단계. 데이터 모델 배포

단계 1. 사용 사례 및 논리적 데이터 모델 식별

목표

- NoSQL 데이터베이스가 필요한 비즈니스 요구 사항과 사용 사례를 수집합니다.
- 개체-관계(ER) 다이어그램을 사용하여 논리적 데이터 모델을 정의합니다.

프로세스

- 비즈니스 분석가가 사용 사례와 예상 결과를 파악하기 위해 비즈니스 사용자를 인터뷰합니다.
- 데이터베이스 엔지니어가 개념적 데이터 모델을 만듭니다.
- 데이터베이스 엔지니어가 논리적 데이터 모델을 만듭니다.
- 데이터베이스 엔지니어가 항목 크기, 데이터 볼륨, 예상되는 읽기 및 쓰기 처리량에 대한 정보를 수집합니다.

도구 및 리소스

- 비즈니스 요구 사항 평가(템플릿 참고)
- 액세스 패턴 매트릭스(템플릿 참고)
- 다이어그램을 만들 때 선호하는 도구

RACI

비즈니스 사용자	비즈니스 분석가	솔루션 아키텍처	데이터베이스 엔지니어	애플리케이션 개발자	DevOps 엔지니어
C	R/A	I	R		

결과

- 문서화된 사용 사례 및 비즈니스 요구 사항
- 논리적 데이터 모델(ER 다이어그램)

단계 2. 예비 비용 추정치 생성

목표

- DynamoDB에 대한 예비 비용 추정치를 개발합니다.

프로세스

- 데이터베이스 엔지니어가 사용 가능한 정보와 DynamoDB 요금 페이지에 제시된 예제를 사용하여 초기 비용 분석을 생성합니다.
  - 온디맨드 용량에 대한 예상 비용을 생성합니다(예제 참고).
  - 제공된 용량에 대한 예상 비용을 생성합니다(예제 참고).
    - 프로비저닝된 용량 모델의 경우 계산기에서 예상 비용을 확인하고 예약 용량에 할인을 적용하십시오.
  - 두 용량 모델의 예상 비용을 비교해 보십시오.

  - 모든 환경 (개발, 제품, QA) 에 대한 추정치를 작성하세요.
- 비즈니스 분석가는 예비 비용 추정치를 검토한 후 승인하거나 거부합니다.

도구 및 리소스

- AWS 요금 계산기

RACI

비즈니스 사용자	비즈니스 분석가	솔루션 아키텍처	데이터베이스 엔지니어	애플리케이션 개발자	DevOps 엔지니어
C	A	I	R		

결과

- 예비 비용 추정

단계 3. 데이터 액세스 패턴 파악

액세스 패턴 또는 쿼리 패턴은 사용자와 시스템이 비즈니스 요구 사항을 충족하기 위해 데이터에 액세스하는 방법을 정의합니다.

목표

- 데이터 액세스 패턴을 문서화합니다.

프로세스

- 데이터베이스 엔지니어와 비즈니스 분석가가 최종 사용자를 인터뷰하여 데이터 액세스 패턴 매트릭스 템플릿을 사용하여 데이터를 쿼리하는 방법을 파악합니다.
  - 새 애플리케이션의 경우, 활동과 목표에 대한 사용자 사례를 검토합니다. 이들은 사용 사례를 문서화하고 사용 사례에 필요한 액세스 패턴을 분석합니다.

  - 기존 애플리케이션의 경우, 쿼리 로그를 분석해 현재 시스템을 사용하고 있는 사람의 수와 핵심 액세스 패턴을 파악해야 합니다.
- 데이터베이스 엔지니어는 액세스 패턴의 다음과 같은 속성을 파악합니다.
  - 데이터 크기: 저장해야 할 데이터의 양과 한 번에 요청할 데이터의 양을 알면 가장 효과적으로 데이터를 파티션(분할)하는 방법을 결정할 수 있습니다.(블로그 포스트 참조)
  - 데이터 모양: 쿼리를 처리할 때 데이터를 변화시키는 대신(RDBMS 시스템의 방식), NoSQL 데이터베이스는 데이터베이스의 모양이 쿼리 대상과 일치하도록 데이터를 구성합니다. 이는 속도와 확장성 향상에 중요한 요소입니다.
  - 데이터 속도: DynamoDB는 프로세스 쿼리에 사용할 수 있는 물리적 파티션의 수를 늘리고, 해당 파티션에 효율적으로 데이터를 배포해 조정합니다. 피크 쿼리 로드를 알면 I/O 용량을 가장 효과적으로 사용할 수 있는 데이터 파티션(분할) 방법을 결정하는 데 도움이 됩니다.
- 비즈니스 사용자가 액세스 또는 쿼리 패턴의 우선 순위를 지정합니다.
  - 우선 쿼리는 일반적으로 가장 많이 사용되거나 가장 관련성이 높은 쿼리입니다. 또한, 응답 지연 시간이 짧아야 하는 쿼리를 파악하는 것도 중요합니다.

도구 및 리소스

- 액세스 패턴 매트릭스(템플릿 참고)
- 올바른 DynamoDB 파티션 키 선택 (AWS 데이터베이스 블로그)
- DynamoDB를 위한 NoSQL 설계 (DynamoDB 설명서)

RACI

비즈니스 사용자	비즈니스 분석가	솔루션 아키텍처	데이터베이스 엔지니어	애플리케이션 개발자	DevOps 엔지니어
C	A	I	R		

결과

- 데이터 액세스 패턴 매트릭스

예

액세스 패턴	우선 순위	읽기 또는 쓰기	설명	유형 (단일 항목, 여러 항목 또는 전체)	주요 특성	필터	결과 정렬
사용자 프로필 생성	높음 (High)	쓰기	사용자가 새 프로필을 만듭니다.	단일 항목	사용자 이름	N/A	N/A
사용자 프로필 업데이트	중간 (Medium)	쓰기	사용자가 프로필을 업데이트합니다.	단일 항목	사용자 이름	사용자 이름 = 현재 사용자	N/A

4단계. 기술 요구 사항 파악

목표

- DynamoDB 데이터베이스의 기술 요구 사항을 수집합니다.

프로세스

- 비즈니스 분석가는 평가 설문지를 사용하여 기술 요구 사항을 수집하기 위해 비즈니스 사용자와 DevOps 팀을 인터뷰합니다.

도구 및 리소스

- 기술 요구 사항 평가 (예제 설문지 참조)

RACI

비즈니스 사용자	비즈니스 분석가	솔루션 아키텍처	데이터베이스 엔지니어	애플리케이션 개발자	DevOps 엔지니어
C	C	A	R		

결과

- 기술 요구 사항 문서

5단계. DynamoDB 데이터 모델 생성

목표

- DynamoDB 데이터 모델을 생성합니다.

프로세스

- 데이터베이스 엔지니어가 각 사용 사례에 필요한 테이블 수를 파악합니다. DynamoDB 애플리케이션에서는 가능한 적은 수의 테이블을 유지할 것을 권장합니다.
- 가장 일반적인 액세스 패턴을 기반으로 데이터를 식별하는 파티션 키가 있는 기본 키와 파티션 키와 정렬 키가 있는 기본 키의 두 가지 유형 중 하나일 수 있는 기본 키를 식별하십시오. 정렬 키는 파티션 내에서 효율적으로 쿼리할 수 있도록 데이터를 그룹화하고 구성하기 위한 보조 키입니다. 정렬 키를 사용해 계층 구조의 어느 수준에서 쿼리를 할 수 있도록 데이터의 계층적(일대다) 관계를 정의할 수 있도록 도와줍니다.(블로그 포스트 참조)
  - 파티션 키 설계
    - 파티션 키를 정의하고 그 분배를 평가합니다.
    - 워크로드를 골고루 분배하기 위해 쓰기 샤딩 필요성을 파악합니다.
  - 정렬 키 설계
    - 정렬 키를 파악합니다.
    - 복합 정렬 키가 필요한지 파악합니다.
    - 버전 제어의 필요성을 파악합니다.

- 액세스 패턴을 기반으로 쿼리 요구 사항을 충족하는 보조 인덱스를 파악합니다.
  - 로컬 보조 인덱스(LSI) 의 필요성을 파악합니다. 기본 테이블과 파티션 키는 동일하지만 정렬 키는 다른 인덱스가 존재합니다.
    - LSI가 있는 테이블의 경우 파티션 키 값당 10GB 크기 제한이 있습니다. LSI가 있는 테이블에는 한 파티션 키 값의 총 크기가 10GB를 초과하지 않는 한 원하는 수의 항목을 저장할 수 있습니다.
  - 글로벌 보조 인덱스(GSI) 의 필요성을 파악합니다. 파티션 키 및 정렬 키가 기본 테이블의 파티션 및 정렬 키와 다를 수 있는 인덱스입니다.(블로그 포스트 참조)
  - 인덱스 예측을 정의합니다. 적은 수의 속성을 프로젝션하여 인덱스에 쓸 항목 크기를 최소화하도록 고려하세요. 이 단계에서는 다음과 같은 사항을 사용할지 여부를 결정해야 합니다.
    - 희소 인덱스
    - 구체화된 집계 쿼리
    - GSI 오버로딩
    - GSI 샤딩
    - GSI를 사용한 최종적으로 일관성이 유지되는 복제본
- 데이터베이스 엔지니어가 데이터에 큰 항목이 포함될지 여부를 결정합니다. 큰 항목이 포함되는 경우에는 압축을 사용하거나 Amazon Simple Storage Service(S3)에 데이터를 저장하는 방식으로 솔루션을 설계합니다.
- 데이터베이스 엔지니어가 시계열 데이터가 필요한지 여부를 결정합니다. 시계열 데이터가 필요한 경우 시계열 설계 패턴을 사용하여 데이터를 모델링합니다.
- 데이터베이스 엔지니어는 ER 모델에 many-to-many 관계가 포함되는지 여부를 결정합니다. 포함되는 경우 인접 목록 설계 패턴을 사용하여 데이터를 모델링합니다.

도구 및 리소스

- Amazon DynamoDB용 NoSQL 워크벤치 — DynamoDB 데이터베이스를 설계하는 데 도움이 되는 데이터 모델링, 데이터 시각화, 쿼리 개발 및 테스트 기능을 제공합니다.
- DynamoDB를 위한 NoSQL 설계 (DynamoDB 설명서)
- 올바른 DynamoDB 파티션 키 선택 (AWS 데이터베이스 블로그)
- DynamoDB에서 보조 인덱스를 사용하는 모범 사례 (DynamoDB 설명서)
- Amazon DynamoDB 글로벌 보조 인덱스를 설계하는 방법(AWS 데이터베이스 블로그)

RACI

비즈니스 사용자	비즈니스 분석가	솔루션 아키텍처	데이터베이스 엔지니어	애플리케이션 개발자	DevOps 엔지니어
I	I	I	R/A		

결과

- 액세스 패턴 및 요구 사항을 충족하는 DynamoDB 테이블 스키마

예

다음 스크린샷은 NoSQL 워크벤치를 보여줍니다.

RetailDatabase | GSI: BundleProducts-VendorProducts-CustomerOrdersByOrderId | GSI: VendorOrdersByStatusDate-ProductInventory | GSI: ProductCatalog-CustomerOrdersByProduct

Primary Key		Attributes					
Partition Key: pk	Sort Key: sk						
P1	B1	GSI1-PK	GSI1-SK	name	desc		
		B1	P1	The Tiki Bundle	Everything you need for an island theme party.		
P4	B2	GSI1-PK	GSI1-SK	name	desc		
		B2	P4	Tiki Bar Set	Be the Mai Tai master with your very own Tiki Bar.		
P2	B1	name	desc	qty	GSI1-PK	GSI1-SK	location
		Tiki Torch	Bamboo tiki torch, 4 ft	6	B1	P2	W1-A9-S10-B52
	B2	name	desc	qty	GSI1-PK	GSI1-SK	location
		Tiki Torch	Bamboo tiki torch, 4 ft	2	B2	P2	W1-A9-S10-B52
	P2	name	desc	qty	location	reorderAt	GSI3-SK
		Tiki Torch	Bamboo tiki torch, 4 ft	656	W1-A9-S10-B52	100	/GardenOutdoor/OutdoorDecor/Lighting/LanternsT
	B1	name	desc	qty	GSI1-PK	GSI1-SK	location
		Tiki Statue - Pele	Tiki of the Hawaiian Fire Goddess Pele, 5 ft.	1	B1	P3	W1-A15-S6-B27

6단계. 데이터 쿼리 생성

목표

- 메인 쿼리를 생성하여 데이터 모델을 검증합니다.

프로세스

- 데이터베이스 엔지니어가 AWS 리전 또는 컴퓨터(DynamoDB 로컬)에서 DynamoDB 테이블을 수동으로 생성합니다.
- 데이터베이스 엔지니어가 DynamoDB 테이블에 샘플 데이터를 추가합니다.
- 데이터베이스 엔지니어가 Amazon DynamoDB용 NoSQL Workbench 나 Java 또는 Python용 AWS SDK를 사용하여 패싯을 구축하여 샘플 쿼리를 구축합니다(블로그 게시물 참고).

  패싯은 DynamoDB 테이블 뷰와 비슷합니다.
- 데이터베이스 엔지니어와 클라우드 개발자가 선호하 언어의 AWS Command Line Interface(AWS CLI) 또는 AWS SDK를 사용하여 샘플 쿼리를 작성합니다.

도구 및 리소스

- DynamoDB 콘솔에 대한 액세스 권한 얻기 위한 활성 AWS 계정
- DynamoDB 로컬(선택 사항), DynamoDB 웹 서비스에 액세스하지 않고 컴퓨터에 데이터베이스를 구축하려는 경우
- Amazon DynamoDB용 NoSQL Workbench (다운로드 및 설명서)
- AWS원하는 언어로 제공되는 SDK (PythonJavaScript, PHP, .NET, 루비, 자바, Go, Node.js, C++, SAP ABAP)

RACI

비즈니스 사용자	비즈니스 분석가	솔루션 아키텍처	데이터베이스 엔지니어	애플리케이션 개발자	DevOps 엔지니어
I	I	I	R/A	R	

결과

- DynamoDB 테이블을 쿼리하기 위한 코드

예제

- Java용 AWSSDK를 사용하는 DynamoDB 예제
- Python 예제
- JavaScript예시

7단계. 데이터 모델 검증

목표

- 데이터 모델이 요구 사항을 충족하는지 확인합니다.

프로세스

- 데이터베이스 엔지니어가 DynamoDB 테이블에 샘플 데이터를 추가합니다.
- 데이터베이스 엔지니어가 코드를 실행하여 DynamoDB 테이블을 쿼리합니다.
- 데이터베이스 엔지니어가 쿼리 결과를 수집합니다.
- 데이터베이스 엔지니어가 쿼리 성능 메트릭을 수집합니다.
- 비즈니스 사용자가 쿼리 결과가 비즈니스 요구 사항을 충족하는지 확인합니다.
- 비즈니스 분석가가 기술 요구 사항을 검증합니다.

도구 및 리소스

- DynamoDB 콘솔에 대한 액세스 권한 얻기 위한 활성 AWS 계정
- DynamoDB 로컬(선택 사항), DynamoDB 웹 서비스에 액세스하지 않고 컴퓨터에 데이터베이스를 구축하려는 경우
- 원하는 언어로 제공되는 AWS SDK

RACI

비즈니스 사용자	비즈니스 분석가	솔루션 아키텍처	데이터베이스 엔지니어	애플리케이션 개발자	DevOps 엔지니어
A	R	I	C		

결과

- 승인된 데이터 모델

8단계. 비용 추정 검토

목표

- 용량 모델을 정의하고 DynamoDB 비용을 추정하여 2단계의 비용 추정을 세부 조정합니다.
- 비즈니스 분석가 및 이해 관계자로부터 최종 재무 승인을 받으세요.

프로세스

- 데이터베이스 엔지니어가 예상 데이터 볼륨을 파악합니다.
- 데이터베이스 엔지니어가 데이터 전송 요구 사항을 파악합니다.
- 데이터베이스 엔지니어가 필요한 읽기 및 쓰기 용량 단위를 정의합니다.
- 비즈니스 분석가가 온디맨드 용량 모델과 프로비저닝 용량 모델 중 하나를 결정합니다.
- 데이터베이스 엔지니어가 DynamoDB Auto Scaling의 필요성을 식별합니다.
- 데이터베이스 엔지니어가 간단한 월간 계산기 도구에 매개변수를 입력합니다.
- 데이터베이스 엔지니어가 비즈니스 이해관계자에게 최종 가격 추정치를 제시합니다.
- 비즈니스 분석가와 이해 관계자가 솔루션을 승인하거나 거부합니다.

도구 및 리소스

- AWS 요금 계산기

RACI

비즈니스 사용자	비즈니스 분석가	솔루션 아키텍처	데이터베이스 엔지니어	애플리케이션 개발자	DevOps 엔지니어
C	A	I	R		

결과

- 용량 모델
- 수정된 비용 예측

9단계. 데이터 모델 배포

목표

- DynamoDB 테이블 (또는 테이블) 을 에 배포합니다. AWS 리전

프로세스

- DevOps 아키텍트는 DynamoDB 테이블 (또는 테이블) 을 위한 AWS CloudFormation 템플릿 또는 기타 코드형 인프라 (IaC) 도구를 생성합니다. AWS CloudFormation테이블 및 관련 리소스를 프로비저닝하고 구성하는 자동화된 방법을 제공합니다.

도구 및 리소스

- AWS CloudFormation

RACI

비즈니스 사용자	비즈니스 분석가	솔루션 아키텍처	데이터베이스 엔지니어	애플리케이션 개발자	DevOps 엔지니어
I	I	C	C		R/A

결과

- AWS CloudFormation 템플릿

예

```
mySecondDDBTable:
   Type: AWS::DynamoDB::
   Table DependsOn: "myFirstDDBTable"
   Properties:
   AttributeDefinitions:
      -  AttributeName: "ArtistId"
         AttributeType: "S"
      -  AttributeName: "Concert"
         AttributeType: "S"
      -  AttributeName: "TicketSales"
         AttributeType: "S"
   KeySchema:
      -  AttributeName: "ArtistId"
         KeyType: "HASH"
      -  AttributeName: "Concert"
         KeyType: "RANGE"
   ProvisionedThroughput:
      ReadCapacityUnits:
         Ref: "ReadCapacityUnits"
      WriteCapacityUnits:
         Ref: "WriteCapacityUnits"
   GlobalSecondaryIndexes:
      -  IndexName: "myGSI"
         KeySchema:
            - AttributeName: "TicketSales"
            KeyType: "HASH"
         Projection:
            ProjectionType: "KEYS_ONLY"
         ProvisionedThroughput:
         ReadCapacityUnits:
            Ref: "ReadCapacityUnits"
         WriteCapacityUnits:
            Ref: "WriteCapacityUnits"
   Tags:
      -  Key: mykey
```

```
        Value: myvalue
```

템플릿

이 섹션에 제공된 템플릿은 AWS 웹사이트의 Amazon DynamoDB를 사용한 게임 플레이어 데이터 모델링을 기반으로 합니다.

> **Note**
>
> 이 섹션의 표에서는 MM 백만의 약어로 사용하고 K를 천의 약어로 사용합니다.

주제

- 비즈니스 요구 사항 평가 템플릿
- 기술 요구 사항 평가 템플릿
- 액세스 패턴 템플릿

비즈니스 요구 사항 평가 템플릿

사용 사례에 대한 설명을 제공합니다.

> **설명**
>
> 온라인 멀티플레이어 게임을 만들고 있다고 가정해 보겠습니다. 이 게임에서는 50명의 플레이어로 구성된 그룹이 한 세션에 참여하여 게임을 플레이하고 플레이하는 데 보통 30분 정도 걸립니다. 게임이 진행되는 동안 특정 플레이어의 기록을 업데이트하여 플레이어가 플레이한 시간, 이들의 통계 또는 게임에서 이겼는지 여부를 표시해야 합니다. 사용자는 게임의 승자를 보거나 각 게임의 경기 리플레이를 보기 위해 이전에 플레이한 게임을 보고 싶어합니다.

사용자에 대한 정보를 제공하십시오.

User	설명	예상 횟수
게임 플레이어	온라인 게임 플레이어.	1 MM
개발 팀	게임 통계를 사용하여 게임을 개선할 내부 팀	100

	게임 경험.	

데이터 소스 및 데이터 수집 방법에 대한 정보를 제공하십시오.

소스(Source)	설명	User
온라인 게임	게임 플레이어는 프로필을 만들고 새 게임을 시작합니다.	게임 플레이어
게임 앱	게임 앱은 시작 및 종료 시간, 플레이어 수, 각 플레이어의 위치, 게임 맵 등 게임에 대한 통계를 자동으로 수집합니다.	

데이터 소비 방식에 대한 정보를 제공하십시오.

소비자	설명	User
온라인 게임	게임 플레이어는 프로필을 보고 게임 통계를 검토할 수 있습니다.	게임 플레이어
데이터 분석	게임 개발팀은 데이터 분석 및 사용자 경험 개선을 위해 게임 통계를 추출합니다. Spark 애플리케이션을 통한 분석을 지원하기 위해 데이터 스토어에서 데이터를 내보내고 Amazon S3로 가져옵니다.	개발 팀

엔티티 목록 및 식별 방법을 제공하십시오.

엔티티 이름	설명	식별자

게임 플레이어	각 사용자 (게이머) 의 ID, 주소, 인구 통계, 관심사와 같은 정보를 저장합니다.	사용자 이름
게임 인스턴스	작성자, 시작, 종료, 맵 Yplay를 포함하여 플레이한 각 게임에 대한 정보를 제공합니다.	게임 ID
게임 유저 매핑	사용자와 게임 간의 many-to-many 관계를 나타냅니다.	게임 ID 및 사용자 이름

엔티티에 대한 ER 모델을 생성하십시오.

엔티티에 대한 높은 수준의 통계를 제공하십시오.

Entity Name	예상 레코드 수	레코드 크기	참고
게임 플레이어	1 MM	1킬로바이트 미만	게임 플랫폼의 사용자는 약 1MM입니다.
게임 인스턴스	6 MM (10만 K/일 * 60일)	1킬로바이트 미만	매일 평균 10만 개의 게임이 있습니다. 지난

			60일을 저장해야 합니다.
게임 유저 매핑	300 밀리미터 (6MM 게임* 50명의 플레이어)	1킬로바이트 미만	평균적으로 각 게임에는 정보를 저장해야 하는 플레이어가 50명입니다.

기술 요구 사항 평가 템플릿

데이터 수집 유형에 대한 정보를 제공하십시오.

데이터 통합 유형	Y/N	설명	Frequency(주파수)
애플리케이션 액세스	Y		
API 게이트웨이	Y		
데이터 스트리밍	N		
Batch 프로세스	N		
ETL	N		
데이터 가져오기	N		
시계열	N		

데이터 소모 유형에 대한 정보를 제공하십시오.

데이터 소비 유형	Y/N	설명	Frequency(주파수)
애플리케이션 액세스			
API 게이트웨이			
데이터 내보내기			

데이터 분석

데이터 집계

보고

검색

데이터 스트리밍

ETL

데이터 볼륨 추정치를 제공하십시오.

엔티티 이름	예상 레코드 수	레코드 크기	데이터 볼륨
게임 플레이어	1 MM	1킬로바이트 미만	최대 1기가바이트 (1 밀리미터* 1킬로바이트)
게임 인스턴스	6 MM (100K/일 * 60일)	1킬로바이트 미만	최대 6기가바이트 (6MM * 1킬로바이트)
게임 유저 매핑	300 밀리미터 (6MM 게임* 50명의 플레이어)	1킬로바이트 미만	최대 300기가바이트 (300 밀리미터 * 1 킬로바이트)

> **Note**
>
> 백업 보존 기간은 60일로 설정합니다. 60일이 지나면 DynamoDB Time to Live(TTL)를 사용하여 데이터를 DynamoDB에서 Amazon S3로 자동 이동하여 분석을 위해 Amazon S3에 데이터를 저장해야 합니다.

시간 패턴에 대한 다음 질문에 답하십시오.

- 사용자가 애플리케이션을 사용할 수 있는 시간대는 언제입니까(예: 연중무휴 또는 평일 오전 9시~오후 5시)?
- 하루 중 사용량이 가장 많은 때가 있습니까? 몇 시간입니까? 애플리케이션 사용량 백분율이 어떻게 됩니까?

쓰기 처리량 요구 사항을 지정하십시오.

엔티티 이름	쓰기 횟수/일	시간/일	초당 쓰기 횟수
게임 플레이어	10,000개의 업데이트	18	< 1
게임 인스턴스	30만	18	5 미만
게임 유저 매핑	1,800,000,000	18	~ 27.777

> **참고**
>
> 게임 플레이어 쓰기 작업: 사용자의 1%가 매일 프로필을 업데이트하므로 1,000,000명의 사용자에 대해 10,000건의 업데이트가 있을 것으로 예상됩니다.
> 게임 인스턴스 쓰기 작업: 매일 100,000개의 게임 각 게임마다 생성, 시작, 종료 시 최소 3개의 쓰기 작업이 있으므로 총 쓰기 작업은 300,000번입니다.
> 게임 사용자 매핑 쓰기 작업: 50명의 플레이어가 참여하는 각 게임의 경우 매일 100,000개의 게임입니다. 평균 게임 시간은 30분이며 게이머 위치는 5초마다 업데이트됩니다. 게이머당 평균 360회의 업데이트가 발생하는 것으로 추정되므로 총 쓰기 작업은 100,000* 50* 360 = 1,800,000,000회 입니다.

읽기 처리량 요구 사항을 지정하십시오.

엔티티 이름	읽기/일	시간/일	읽기/초
게임 플레이어	20만	18	~ 3
게임 인스턴스	5백만	18	~ 77
게임 유저 매핑	1,800,000,000	18	~ 27.777

> **참고**
>
> 게임 플레이어 읽기 작업: 사용자의 20%가 게임을 시작하므로 1MM * 0.2 = 200,000회 입니다.
> 게임 인스턴스 쓰기 작업: 매일 100,000개의 게임 각 게임마다 플레이어당 최소 1회, 게임당 50명의 플레이어가 읽기 작업을 수행하므로 총 5,000,000회의 읽기 작업이 이루어집니다.
> 게임 사용자 매핑 읽기 작업: 50명의 플레이어가 참여하는 각 게임의 경우 매일 100,000개의 게임입니다. 평균 게임 시간은 30분이며 게이머 위치는 5초마다 업데이트됩니다. 게이머당 평균 360회의 업데이트가 발생하는 것으로 추정되므로 총 읽기 작업은 100,000* 50* 360 = 1,800,000,000회 입니다.

데이터 액세스 지연 요구 사항을 지정하십시오.

Operation	99개의 백분위수	최대 지연 시간
읽기(Read)	30밀리초	100 밀리세컨드
쓰기(Write)	10 밀리세컨드	50 밀리세컨드

데이터 가용성 요구 사항을 지정하십시오.

요구 사항	Y/N	지표	참고
높은 가용성	Y	99.9%	
RPO	Y	한 시간	복구 시간 목표
RPO	Y	1시간	복구 시점 목표
재해 복구	N		
지역 내 데이터 복제	N		
지역 간 데이터 복제	N	3초의 지연 시간	어느 AWS 리전 쪽이요?

보안 요구 사항을 지정하십시오.

요구 사항	Y/N	참고
민감한 데이터 저장소	N	보호 대상 의료 정보 (PHI), 결제 카드 업계 (PCI) 정보, 개인 식별 정보 (PII)?
저장된 데이터 암호화	Y	
전송 중 데이터 암호화	Y	
클라이언트측 암호화	N	
모든 독점 또는 제3자 공급업체 암호화 라이브러리	N	
데이터 액세스 로깅	N	
데이터 액세스 감사	N	

액세스 패턴 템플릿

다음 필드를 사용하여 사용 사례의 액세스 패턴에 대한 정보를 수집하고 문서화하십시오.

필드	설명
액세스 패턴	액세스 패턴의 이름을 제공하십시오.
설명	액세스 패턴에 대한 자세한 설명을 제공하십시오.
우선 순위	액세스 패턴의 우선 순위 (높음, 중간 또는 낮음)를 정의합니다. 이는 애플리케이션에 가장 관련성이 높은 액세스 패턴을 정의합니다.
읽기 또는 쓰기	읽기 액세스 패턴입니까 아니면 쓰기 액세스 패턴입니까?
Type	패턴이 단일 항목, 여러 항목 또는 모든 항목에 액세스합니까?

필드	설명
필터	액세스 패턴에 필터가 필요합니까?
정렬	결과를 정렬해야 합니까?

템플릿

액세스 패턴	설명	우선 순위	읽기 또는 쓰기	유형 (단일 항목, 멀티플 항목 또는 전체)	주요 특성	필터	결과 정렬
사용자 프로필 생성	사용자가 새 프로필을 만듭니다.	높음	쓰기 (Write)	단일 항목	사용자 이름	해당 사항 없음	해당 사항 없음
사용자 프로필 업데이트	사용자가 프로필을 업데이트합니다.	Medium	쓰기 (Write)	단일 항목	사용자 이름	사용자 이름 = 현재 사용자	해당 사항 없음
사용자 프로필 가져오기	사용자가 프로필을 검토합니다.	높음	읽기 (Read)	단일 항목	사용자 이름	사용자명 = 현재 사용자	해당 사항 없음
게임 만들기	사용자가 새 게임을 만듭니다.	높음	쓰기 (Write)	단일 항목	게임 ID	해당 사항 없음	해당 사항 없음
오픈 게임 찾기	사용자가 오픈 게임을 검색합	높음	읽기 (Read)	여러 항목		GameStatus = 오픈	시작 타임스탬프 내림차순

	니다. 검색 결과는 시작 타임스탬프를 기준으로 내림차순으로 정렬됩니다.						
맵으로 오픈 게임 찾기	사용자는 시작 타임스탬프를 기준으로 내림차순으로 정렬된 특정 맵을 사용하여 열린 게임을 검색합니다. 순서.	Medium	읽기 (Read)	여러 아이템		GameStatus = 열기 및 맵 = XYZ	시작 타임스탬프 내림차순
게임 보기	사용자가 게임 세부 정보를 검토합니다.	높음	읽기 (Read)	단일 항목	GameID	해당 사항 없음	해당 사항 없음
게임 내 사용자 보기	사용자는 게임 내 모든 사용자 목록을 가져옵니다.	Medium	읽기 (Read)	여러 아이템		게임 ID = XYZ	해당 사항 없음

유저를 게임에 참여시키세요	사용자가 오픈 게임에 참여합니다.	높음	쓰기 (Write)	단일 항목	게임 ID 및 사용자 이름	GameStatus = 열기	해당 사항 없음
게임 시작	사용자가 새 게임을 시작합니다.	높음	쓰기 (Write)	단일 항목	게임 ID	해당 사항 없음	해당 사항 없음
사용자용 게임 업데이트	게임 내 사용자 위치 업데이트.	Medium	쓰기 (Write)	단일 항목	게임 ID 및 사용자 이름	해당 사항 없음	해당 사항 없음
게임 업데이트	게임 종료, 통계 업데이트	Medium	쓰기 (Write)	단일 항목	게임 ID	해당 사항 없음	해당 사항 없음
사용자의 모든 과거 게임 찾기	사용자가 플레이한 모든 게임을 게임 시작 타임스탬프별로 순서대로 나열합니다.	낮음	읽기 (Read)	여러 아이템	사용자 이름 및 게임 ID	사용자 이름 = 현재 사용자	시작 타임스탬프
데이터 분석을 위한 데이터 내보내기	개발팀은 데이터를 Amazon S3로 내보내는 일괄 작업을 실행합니다.	낮음	읽기 (Read)	모두	해당 사항 없음	해당 사항 없음	해당 사항 없음

모범 사례

다음과 같은 DynamoDB 설계 모범 사례를 사용해 보십시오.

- 파티션 키 설계-카디널리티가 높은 파티션 키를 사용하여 로드를 균등하게 분산합니다.
- 인접 목록 디자인 패턴 - 이 디자인 패턴을 관리 one-to-many 및 관계에 사용합니다. many-to-many
- 스파스 인덱스 - 글로벌 보조 인덱스 (GSI) 에는 스파스 인덱스를 사용합니다. GSI를 생성할 때는 파티션 키와 정렬 키(선택 사항)를 지정해야 합니다. 해당 GSI 파티션 키를 포함하는 기본 테이블의 항목만 스파스 인덱스에 나타납니다. 이는 GSI를 더 작게 유지하는 데 도움이 됩니다.
- 인덱스 오버로딩 - 다양한 유형의 항목을 인덱싱하는 데 동일한 GSI를 사용합니다.
- GSI 쓰기 샤드 - 효율적이고 빠른 쿼리를 위해 파티션 전체에 데이터를 현명하게 샤드합니다.
- 대용량 항목 - 테이블 내에 메타데이터만 저장하고, Amazon S3에 blob을 저장하고, DynamoDB에 참조를 유지합니다. 대용량 항목을 여러 항목으로 나누고 정렬 키를 사용하여 효율적으로 인덱싱합니다.

설계 모범 사례에 대한 자세한 내용은 Amazon DynamoDB 설명서를 참조하십시오.

계층적 데이터 모델링의 예

다음 섹션에서는 예제 자동차 회사를 사용하여 데이터 모델링 프로세스 단계를 사용하여 DynamoDB에서 다단계 구성 요소 관리 시스템을 설계하는 방법을 보여줍니다.

주제

- 1단계: 사용 사례 및 논리적 데이터 모델 식별
- 2단계: 예비 비용 추정 생성
- 3단계: 데이터 액세스 패턴 파악
- 4단계: 기술 요구 사항 식별
- 5단계: DynamoDB 데이터 모델 생성
- 6단계: 데이터 쿼리 만들기
- 7단계: 데이터 모델 검증
- 8단계: 예상 비용 검토
- 9단계: 데이터 모델 배포

1단계: 사용 사례 및 논리적 데이터 모델 식별

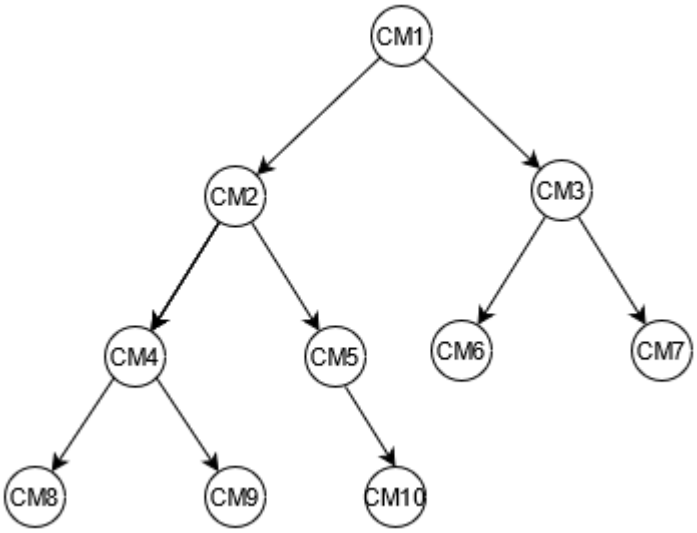
한 자동차 회사는 사용 가능한 모든 자동차 부품을 저장 및 검색하고 서로 다른 구성 요소와 부품 간의 관계를 구축하는 트랜잭션 구성 요소 관리 시스템을 구축하려고 합니다. 예를 들어, 자동차에는 여러 개의 배터리가 포함되고, 각 배터리에는 여러 개의 고급 모듈이 포함되고, 각 모듈에는 여러 개의 셀이 포함되고, 각 셀에는 여러 개의 하위 수준 구성 요소가 포함되어 있습니다.

일반적으로 계층적 관계 모델을 구축하려면 Amazon Neptune과 같은 그래프 데이터베이스가 더 적합합니다. 그러나 유연성, 보안, 성능 및 확장성 때문에 Amazon DynamoDB가 계층적 데이터 모델링의 더 나은 대안인 경우도 있습니다.

예를 들어, 80~90%의 쿼리가 트랜잭션이고 DynamoDB가 적합한 시스템을 구축할 수 있습니다. 이 예시에서는 나머지 10~ 20% 의 쿼리가 관계형이며, Neptune과 같은 그래프 데이터베이스가 더 적합합니다. 이 경우 쿼리의 10~ 20% 만 처리하도록 아키텍처에 추가 데이터베이스를 포함하면 비용이 증가할 수 있습니다. 또한 여러 시스템을 유지 관리하고 데이터를 동기화해야 하는 운영상의 부담도 가중됩니다. 대신 DynamoDB에서 10~20%의 관계형 쿼리를 모델링할 수 있습니다.

자동차 구성 요소의 예제 트리를 도식화하면 부품 간의 관계를 매핑하는 데 도움이 될 수 있습니다. 다음 다이어그램은 4개 수준의 종속성 그래프를 보여줍니다. CM1은 예제 자동차 자체의 최상위 구성 요

소로, 2개의 예시 배터리를 위한 2개의 하위 구성 요소인 CM2와 CM3이 있습니다. 각 배터리에는 모듈인 2개의 하위 구성 요소가 있습니다. CM2에는 모듈 CM4와 CM5가 있고 CM3에는 모듈 CM6과 CM7이 있습니다. 각 모듈에는 셀인 여러 하위 구성 요소가 있습니다. CM4 모듈에는 CM8과 CM9라는 2개의 셀이 있습니다. CM5에는 CM10이라는 셀이 1개 있습니다. CM6와 CM7에는 아직 관련 셀이 없습니다.



이 가이드에서는 이 트리와 해당 구성 요소 식별자를 참조로 사용합니다. 최상위 구성 요소를 상위라고 하고 하위 구성 요소를 하위라고 합니다. 예를 들어, 상단 구성 요소 CM1은 CM2 및 CM3의 상위입니다. CM2는 CM4 및 CM5의 상위입니다. 이는 상위-하위 관계를 그래프로 나타낸 것입니다.

트리에서 구성 요소의 전체 종속성 그래프를 볼 수 있습니다. 예를 들어, CM8은 CM4에 종속되고, CM4는 CM2에 종속되고, CM2는 CM1에 종속됩니다. 트리는 전체 종속성 그래프를 경로로 정의합니다. 경로는 다음 두 가지를 설명합니다.

- 종속성 그래프
- 트리에서의 위치

비즈니스 요구 사항에 맞는 템플릿 작성:

사용자에 대한 정보를 제공하십시오.

User	설명
직원	자동차 및 부품 정보가 필요한 자동차 회사의 내부 직원

데이터 소스 및 데이터 수집 방법에 대한 정보를 제공하십시오.

소스(Source)	설명	User
관리 시스템	사용 가능한 자동차 부품 및 다른 구성 요소 및 부품과의 관계와 관련된 모든 데이터를 저장하는 시스템입니다.	직원

데이터 소비 방식에 대한 정보를 제공하십시오.

소비자	설명	User
관리 시스템	상위 구성 요소 ID의 모든 직계 하위 구성 요소를 검색합니다.	직원
관리 시스템	구성 요소 ID에 대한 모든 하위 구성 요소의 반복 목록을 검색합니다.	직원
관리 시스템	구성 요소의 상위 항목 보기	직원

2단계: 예비 비용 추정 생성

솔루션이 재정적으로 실행 가능한지 확인할 수 있도록 애플리케이션의 모든 환경에 대한 예상 비용을 계산하는 것이 중요합니다. 가장 좋은 방법은 개발 및 배포를 진행하기 전에 높은 수준의 추정을 하고 비즈니스 분석가의 승인을 받는 것입니다.

- 데이터베이스 엔지니어가 사용 가능한 정보와 DynamoDB 요금 페이지에 제시된 예제를 사용하여 초기 비용 분석을 생성합니다.
  - 온디맨드 용량에 대한 예상 비용을 생성합니다(예제 참고).
  - 제공된 용량에 대한 예상 비용을 생성합니다(예제 참고).
    - 프로비저닝된 용량 모델의 경우 계산기에서 예상 비용을 확인하고 예약 용량에 할인을 적용하십시오.
  - 두 용량 모델의 예상 비용을 비교해 보십시오.

  - 모든 환경 (개발, 제품, QA) 에 대한 추정치를 작성하세요.
- 비즈니스 분석가는 예비 비용 추정치를 검토한 후 승인하거나 거부합니다.

이러한 참조 값을 사용하여 승인을 위해 제출할 예상 가격을 만들 수 있습니다. DynamoDB 요금 페이지와 AWS 요금 계산기를 사용하여 예산을 생성할 수 있습니다.

3단계: 데이터 액세스 패턴 파악

이 예제 사용 사례에는 다양한 자동차 구성 요소 간의 관계를 관리하기 위한 다음과 같은 액세스 패턴이 있습니다.

액세스 패턴	우선 순위	읽기 또는 쓰기	설명	Type	필터	결과 정렬
직계 자녀	높음(High)	읽기	상위 구성 요소 ID의 모든 직계 하위 구성 요소를 검색합니다.	다중	`Component ID`	N/A
모든 하위 구성 요소	높음(High)	읽기	구성 요소 ID에 대한 모든 하위 구성 요소의 반복 목록을 검색합니다.	다중	`Component ID`	N/A
상위 멤버	높음(High)	읽기	구성 요소의 상위 항목을 검색합니다.	다중	`Component ID`	N/A

4단계: 기술 요구 사항 식별

이 예제에는 이 예제의 범위를 벗어나는 특정 기술 요구 사항이 없습니다. 실제 경우에는 개발 및 배포를 진행하기 전에 이 단계를 완료하고 모든 기술 요구 사항이 충족되었는지 확인하는 것이 가장 좋습니다. 예제 설문지를 사용하여 비즈니스 사례의 이 단계를 완료할 수 있습니다. 또한 DynamoDB 서비스 할당량을 검증하여 설계된 솔루션에 엄격한 제한이 없는지 확인하는 것이 좋습니다.

5단계: DynamoDB 데이터 모델 생성

기본 테이블과 글로벌 보조 인덱스 (GSI) 의 파티션 키를 정의합니다.

- 주요 설계 모범 사례에 따라 이 예제에서는 를 기본 테이블의 파티션 `ComponentId` 키로 사용합니다. 고유하기 때문에 세분성을 제공할 `ComponentId` 수 있습니다. DynamoDB는 파티션 키의 해시 값을 사용하여 데이터가 물리적으로 저장되는 파티션을 결정합니다. 고유한 구성 요소 ID는 다른 해시 값을 생성하여 테이블 내부의 데이터 배포를 용이하게 할 수 있습니다. `ComponentId`파티션 키를 사용하여 기본 테이블을 쿼리할 수 있습니다.
- 구성 요소의 직계 하위 구성요소를 찾으려면 GSI를 생성하십시오. 여기서 `ParentId` 는 파티션 키이고 는 정렬 `ComponentId` 키입니다. 파티션 `ParentId` 키로 사용하여 이 GSI를 쿼리할 수 있습니다.
- 구성 요소의 모든 재귀 하위 항목을 찾으려면 `GraphId`가 파티션 키이고 `Path`가 정렬 키인 GSI를 생성합니다. `GraphId`를 파티션 키로 사용하고 정렬 키에 `BEGINS_WITH(Path, "$path")` 연산자를 사용하여 이 GSI를 쿼리할 수 있습니다.

	파티션 키	정렬 키	매핑 속성
기본 테이블	`ComponentId`		`ParentId`, `GraphId`, `Path`
GSI1	`ParentId`	`ComponentId`	
GSI2	`GraphId`	`Path`	`ComponentId`

테이블에 구성 요소 저장

다음 단계는 각 구성 요소를 DynamoDB 기본 테이블에 저장하는 것입니다. 예제 트리에서 모든 구성 요소를 삽입하면 다음과 같은 기본 테이블이 표시됩니다.

ComponentId	ParentId	GraphId	경로
CM1		CM1#1	CM1
CM2	CM1	CM1#1	CM1|CM2
CM3	CM1	CM1#1	CM1|CM3
CM4	CM2	CM1#1	CM1|CM2|CM4
CM5	CM2	CM1#1	CM1|CM2|CM5
CM6	CM3	CM1#1	CM1|CM3|CM6
CM7	CM3	CM1#1	CM1|CM3|CM7
CM8	CM4	CM1#1	CM1|CM2|CM4|CM8
CM9	CM4	CM1#1	CM1|CM2|CM4|CM9
CM10	CM5	CM1#1	CM1|CM2|CM5|CM10

GSI1 인덱스

구성 요소의 직계 하위 구성요소를 모두 검사하려면 파티션 키와 `ComponentId` 정렬 `ParentId` 키로 사용하는 인덱스를 만듭니다. 다음 피벗 테이블은 GSI1 인덱스를 나타냅니다. 이 인덱스를 사용하면 상위 구성 요소 ID로 모든 직계 하위 구성 요소를 검색할 수 있습니다. 예를 들어, 자동차에서 사용할 수 있는 배터리 수(CM1) 또는 모듈에서 사용할 수 있는 셀(CM4)을 확인할 수 있습니다.

ParentId	ComponentId
CM1	CM2
	CM3
CM2	CM4
	CM5
CM3	CM6
	CM7
CM4	CM8
	CM9
CM5	CM10

GSI2 인덱스

다음 피벗 테이블은 GSI2 인덱스를 나타냅니다. `GraphId`를 파티션 키로 사용하고, `Path`를 정렬 키로 사용하여 구성됩니다. `GraphId`와 정렬 키 (`Path`) `begins_with` 연산을 사용하면 트리에서 구성 요소의 전체 계보를 찾을 수 있습니다.

GraphId	경로	ComponentId
CM1#1	CM1	CM1
	CM1|CM2	CM2
	CM1|CM3	CM3

CM1|CM2|CM4	CM4
CM1|CM2|CM5	CM5
CM1|CM2|CM4|CM8	CM8
CM1|CM2|CM4|CM9	CM9
CM1|CM2|CM5|CM10	CM10
CM1|CM3|CM6	CM6
CM1|CM3|CM7	CM7

6단계: 데이터 쿼리 만들기

액세스 패턴을 정의하고 데이터 모델을 설계한 후 DynamoDB 데이터베이스에서 계층적 데이터를 쿼리할 수 있습니다. 비용을 절감하고 성능을 보장하는 데 도움이 되는 모범 사례로, 다음 예시에서는 쿼리 작업 없이 쿼리 작업만 사용합니다. `Scan`

- 구성 요소의 상위 구성 요소 찾기

  CM8 구성 요소의 상위 항목(상위, 차상위, 최상위 등)을 찾으려면 `ComponentId = "CM8"`을 사용하여 기본 테이블을 쿼리합니다. 쿼리는 다음 레코드를 반환합니다.

  결과 데이터의 크기를 줄이려면 프로젝션 표현식을 사용하여 `Path` 속성만 반환합니다.

ComponentId	ParentId	GraphId	경로
CM8	CM4	CM1#1	CM1|CM2|CM4|CM8

  경로

  CM1|CM2|CM4|CM8

  이제 파이프 (“|”) 를 사용하여 경로를 분할하고 첫 번째 N-1 구성 요소를 가져와서 조상을 구하십시오.

쿼리 결과: CM8의 상위 항목은 CM1, CM2, CM4입니다.

- 구성요소의 직계 하위 구성요소를 찾으세요.

  CM2 구성 요소의 모든 직계 하위 구성 요소 또는 한 수준 다운스트림 구성 요소를 가져오려면 를 사용하여 GSI1을 쿼리하십시오. `ParentId = "CM2"` 쿼리는 다음 레코드를 반환합니다.

ParentId	ComponentId
CM2	CM4
	CM5

- 최상위 구성 요소를 사용하여 모든 다운스트림 하위 구성 요소를 찾습니다.

  최상위 컴포넌트 CM1의 모든 하위 컴포넌트 또는 다운스트림 컴포넌트를 가져오려면 `GraphId = "CM1#1"` 및 를 사용하여 GSI2를 쿼리하고 `begins_with("Path", "CM1|")` 를 사용하여 프로젝션 표현식을 사용하십시오. `ComponentId` 해당 트리와 관련된 모든 구성 요소가 반환됩니다.

  이 예제에는 최상위 구성 요소가 CM1인 단일 트리가 있습니다. 실제로는 동일한 테이블에 최상위 구성 요소가 수백만 개 있을 수 있습니다.

GraphId	ComponentId
	CM2
CM1#1	CM3
	CM4
	CM5
	CM8
	CM9
	CM10
	CM6

CM7

- 중간 수준 구성 요소를 사용하여 모든 다운스트림 하위 구성 요소를 찾습니다.

  구성 요소 CM2에 대해 모든 하위 구성 요소 또는 다운스트림 구성 요소를 재귀적으로 가져오려면 두 가지 옵션이 있습니다. 한 수준씩 재귀적으로 쿼리하거나 GSI2 인덱스를 쿼리할 수 있습니다.

  - 하위 구성 요소의 마지막 수준에 도달할 때까지 한 수준씩 재귀적으로 GSI1을 쿼리합니다.

    1. `ParentId = "CM2"`를 사용하여 GSI1을 쿼리합니다. 이는 다음 레코드를 반환합니다.

ParentId	ComponentId
CM2	CM4
	CM5

    2. 다시 `ParentId = "CM4"`를 사용하여 GSI1을 쿼리합니다. 이는 다음 레코드를 반환합니다.

ParentId	ComponentId
CM4	CM8
	CM9

    3. 다시 `ParentId = "CM5"`를 사용하여 GSI1을 쿼리합니다. 이는 다음 레코드를 반환합니다.

       루프 계속: 마지막 수준에 도달할 때까지 각 `ComponentId`에 대해 쿼리합니다. `ParentId = "<ComponentId>"`를 사용한 쿼리가 결과를 반환하지 않는 경우 이전 결과가 트리의 마지막 수준에서 나온 것입니다.

ParentId	ComponentId
CM5	CM10

    4. 모든 결과를 병합합니다.

       결과= [CM4, CM5] + [CM8, CM9] + [CM10]

= [CM4, CM5, CM8, CM9, CM10]

- 최상위 구성 요소(자동차 또는 CM1)의 계층 트리를 저장하는 GSI2를 쿼리합니다.
  1. 먼저 최상위 구성 요소 또는 최상위 항목과 CM2의 `Path`를 찾습니다. 이를 위해 이 경우 `ComponentId = "CM2"`를 사용하여 기본 테이블을 쿼리하여 계층 트리에서 해당 구성 요소의 경로를 찾습니다. 및 h 속성을 선택합니다. `GraphId Pat` 쿼리는 다음 레코드를 반환합니다.

GraphId	경로
CM1#1	CM1|CM2

  2. 를 사용하여 GSI2를 쿼리합니다. `GraphId = "CM1#1" AND BEGINS_WITH("Path", "CM1|CM2|")` 쿼리가 다음 결과를 반환합니다.

GraphId	경로	ComponentId
CM1#1	CM1|CM2|CM4	CM4
	CM1|CM2|CM5	CM5
	CM1|CM2|CM4|CM8	CM8
	CM1|CM2|CM4|CM9	CM9
	CM1|CM2|CM5|CM10	CM10

  3. CM2에 대해 모든 하위 구성 요소를 반환하려면 `ComponentId` 속성을 선택합니다.

7단계: 데이터 모델 검증

이 단계에서 비즈니스 사용자는 쿼리 결과를 검증하고 비즈니스 요구 사항을 충족하는지 확인합니다. 다음 표를 사용하여 사용자의 요구 사항과 비교하여 액세스 패턴을 확인할 수 있습니다.

질문	기본 테이블/GSI	Query
사용자로서 상위 구성 요소 ID에 대한 모든 직계 하위 구성 요소를 검색하고 싶습니다.	GSI1	`ParentId = "<ComponentId>"`

		(구성 요소의 직계 하위 항목을 찾기)
사용자로서 구성 요소 ID에 대한 모든 하위 구성 요소의 반복 목록을 검색하고 싶습니다.	GSI1 또는 GSI2	GSI1: `ParentId = "<ComponentId>"` 또는 GSI2: `GraphId = "<TopLevelComponentId>#N" AND BEGINS_WITH("Path", "<PATH_OF_Component>")` (최상위 구성 요소를 사용하여 모든 하위 구성 요소를 찾습니다. 중간 수준 구성 요소를 사용하여 모든 아래 수준 하위 구성 요소를 찾습니다.)
사용자로서 구성 요소의 상위 항목을 보고 싶습니다.	기본 테이블	`ComponentId = "<ComponentId>"` 그런 다음 Path 속성을 선택합니다. (구성 요소의 상위 항목을 찾습니다.)

또한 모든 프로그래밍 언어로 스크립트 (테스트) 를 구현하여 DynamoDB를 직접 쿼리하고 결과를 예상 결과와 비교할 수 있습니다.

8단계: 예상 비용 검토

비용 추정치를 다시 검토하고 수정하십시오. 또한 비즈니스 이해 관계자와 함께 이를 검증하고 승인을 받아 다음 단계로 넘어가는 것도 좋은 방법입니다.

목표

- 용량 모델을 정의하고 DynamoDB 비용을 추정하여 2단계부터 추정 비용을 세분화합니다.

- 비즈니스 분석가와 이해 관계자로부터 최종 재무 승인을 받으세요.

프로세스

- 데이터베이스 엔지니어가 예상 데이터 볼륨을 파악합니다.
- 데이터베이스 엔지니어가 데이터 전송 요구 사항을 파악합니다.
- 데이터베이스 엔지니어가 필요한 읽기 및 쓰기 용량 단위를 정의합니다.
- 비즈니스 분석가가 온디맨드 용량 모델과 프로비저닝 용량 모델 중 하나를 결정합니다.
- 데이터베이스 엔지니어가 DynamoDB Auto Scaling의 필요성을 식별합니다.
- 데이터베이스 엔지니어가 매개 변수를 에 AWS Pricing Calculator 입력합니다.
- 데이터베이스 엔지니어가 비즈니스 이해 관계자에게 최종 가격 추정치를 제시합니다.
- 비즈니스 분석가와 이해 관계자가 솔루션을 승인하거나 거부합니다.

9단계: 데이터 모델 배포

이 특정 예제에서는 최신 데이터베이스 개발 및 운영을 위한 애플리케이션인 NoSQL Workbench를 사용하여 모델을 배포했습니다. 이 도구를 사용하면 데이터 모델을 만들고, 데이터를 업로드하고, 모델에 직접 배포할 수 있습니다. AWS 계정 이 예제를 구현하려면 NoSQL Workbench에서 생성한 다음 AWS CloudFormation 템플릿을 사용할 수 있습니다.

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  Components:
    Type: 'AWS::DynamoDB::Table'
    Properties:
      KeySchema:
        - AttributeName: ComponentId
          KeyType: HASH
      AttributeDefinitions:
        - AttributeName: ComponentId
          AttributeType: S
        - AttributeName: ParentId
          AttributeType: S
        - AttributeName: GraphId
          AttributeType: S
        - AttributeName: Path
          AttributeType: S
```

```
      GlobalSecondaryIndexes:
        - IndexName: GS1
          KeySchema:
            - AttributeName: ParentId
              KeyType: HASH
            - AttributeName: ComponentId
              KeyType: RANGE
          Projection:
            ProjectionType: KEYS_ONLY
        - IndexName: GSI2
          KeySchema:
            - AttributeName: GraphId
              KeyType: HASH
            - AttributeName: Path
              KeyType: RANGE
          Projection:
            ProjectionType: INCLUDE
            NonKeyAttributes:
              - ComponentId
      BillingMode: PAY_PER_REQUEST
      TableName: Components
```

추가 리소스

DynamoDB에 대한 추가 정보

- DynamoDB 요금
- DynamoDB 문서화
- DynamoDB를 위한 NoSQL 설계
- 쓰기 샤딩
- 로컬 보조 인덱스(LSI)
- 글로벌 보조 인덱스(GSI)
- GSI 오버로딩
- GSI 샤딩
- GSI를 사용한 최종적으로 일관성이 유지되는 복제본 생성
- 희소 인덱스
- 구체화된 집계 쿼리
- 시계열 데이터의 설계 패턴
- 인접 목록 설계 패턴
- 온디맨드 방식 및 프로비저닝 용량 모델
- DynamoDB Auto Scaling
- DynamoDB 라이브 시간(TTL)
- DynamoDB를 사용한 게임 플레이어 데이터 모델링(실험실)

AWS 서비스

- AWS CloudFormation
- Amazon S3

도구

- AWS Pricing Calculator
- DynamoDB용 NoSQL Workbench
- DynamoDB Local

- DynamoDB 및 AWS SDK

모범 사례

- DynamoDB를 사용한 설계 및 아키텍처 설계 모범 사례 (DynamoDB 문서화)
- DynamoDB의 보조 인덱스 사용에 대한 모범 사례 (DynamoDB 문서화)
- 큰 항목과 속성 저장 모범 사례 (DynamoDB 문서화)
- 올바른 DynamoDB 파티션 키 AWS 선택 (데이터베이스 블로그)
- Amazon DynamoDB 글로벌 보조 인덱스를 설계하는 방법 (데이터베이스 블로그) AWS
- Amazon DynamoDB용 NoSQL 워크벤치의 패싯이란(Medium 웹사이트)

AWS 일반 리소스

- AWS 권장 가이드 웹사이트
- AWS 설명서
- AWS 일반 참조

기여자

이 가이드의 기여자는 다음과 같습니다.

- 카밀로 곤잘레스, 선임 데이터 아키텍트 AWS
- 모이눌르 알-마문, 선임 빅데이터 아키텍트, AWS
- 산티아고 세구라, 전문 서비스 컨설턴트 AWS
- 사테이시 쿠마르 찬드라프라카삼, 클라우드 애플리케이션 아키텍트 AWS

문서 기록

아래 표에 이 가이드의 주요 변경 사항이 설명되어 있습니다. 향후 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하세요.

변경 사항	설명	날짜
계층적 데이터 모델링에 대한 모범 사례 섹션과 예제가 추가되었습니다.	DynamoDB 모범 사례의 요약과 계층적 모델을 설계하고 step-by-step 검증하는 예제를 추가했습니다.	2023년 12월 5일
최초 게시	—	2020년 10월 26일

AWS 권장 가이드 용어집

다음은 AWS 권장 가이드에서 제공하는 전략, 가이드, 패턴 등에서 일반적으로 사용되는 용어입니다. 용어집 항목을 제안하려면 용어집 끝에 있는 피드백 제공 링크를 사용하세요.

숫자

7가지 전략

애플리케이션을 클라우드로 이전하기 위한 7가지 일반적인 마이그레이션 전략 이러한 전략은 Gartner가 2011년에 파악한 5가지 전략을 기반으로 하며 다음으로 구성됩니다.

- 리팩터링/리아키텍트 - 클라우드 네이티브 기능을 최대한 활용하여 애플리케이션을 이동하고 해당 아키텍처를 수정함으로써 민첩성, 성능 및 확장성을 개선합니다. 여기에는 일반적으로 운영 체제와 데이터베이스 이식이 포함됩니다. 예를 들어, 온프레미스 Oracle 데이터베이스를 Amazon Aurora PostgreSQL 호환 버전으로 마이그레이션합니다.
- 리플랫포밍(리프트 앤드 리셰이프) - 애플리케이션을 클라우드로 이동하고 일정 수준의 최적화를 도입하여 클라우드 기능을 활용합니다. 예를 들어, 온프레미스 Oracle 데이터베이스를 AWS 클라우드의 Amazon RDS for Oracle로 마이그레이션합니다.
- 재구매(드롭 앤드 숍) - 일반적으로 기존 라이선스에서 SaaS 모델로 전환하여 다른 제품으로 전환합니다. 예를 들어, 고객 관계 관리(CRM) 시스템을 Salesforce.com으로 마이그레이션합니다.
- 리호스팅(리프트 앤드 시프트) - 애플리케이션을 변경하지 않고 클라우드로 이동하여 클라우드 기능을 활용합니다. 예를 들어, 온프레미스 Oracle 데이터베이스를 AWS 클라우드의 EC2 인스턴스에 있는 Oracle로 마이그레이션합니다.
- 재배치(하이퍼바이저 수준의 리프트 앤 시프트) - 새 하드웨어를 구매하거나, 애플리케이션을 다시 작성하거나, 기존 운영을 수정하지 않고도 인프라를 클라우드로 이동합니다. 이 마이그레이션 시나리오는 온프레미스 환경과 AWS 간의 가상 머신 호환성 및 워크로드 이동성을 지원하는 AWS의 VMware Cloud에 한정됩니다. 인프라를 AWS의 VMware Cloud로 마이그레이션할 때 온프레미스 데이터 센터에서 VMware Cloud Foundation 기술을 사용할 수 있습니다. 예를 들어, Oracle 데이터베이스를 호스팅하는 하이퍼바이저를 AWS의 VMware Cloud로 재배치합니다.
- 유지(보관) - 소스 환경에 애플리케이션을 유지합니다. 대규모 리팩터링이 필요하고 해당 작업을 나중으로 연기하려는 애플리케이션과 비즈니스 차원에서 마이그레이션할 이유가 없어 유지하려는 레거시 애플리케이션이 여기에 포함될 수 있습니다.
- 사용 중지 - 소스 환경에서 더 이상 필요하지 않은 애플리케이션을 폐기하거나 제거합니다.

A

ABAC

속성 기반 액세스 제어를 참조하십시오.

추상화된 서비스

매니지드 서비스를 참조하십시오.

산

원자성, 일관성, 격리성, 내구성을 참조하십시오.

능동-능동 마이그레이션

양방향 복제 도구 또는 이중 쓰기 작업을 사용하여 소스 데이터베이스와 대상 데이터베이스가 동기화된 상태로 유지되고, 두 데이터베이스 모두 마이그레이션 중 연결 애플리케이션의 트랜잭션을 처리하는 데이터베이스 마이그레이션 방법입니다. 이 방법은 일회성 전환이 필요한 대신 소규모의 제어된 배치로 마이그레이션을 지원합니다. 더 유연하지만 액티브-패시브 마이그레이션보다 더 많은 작업이 필요합니다.

능동-수동 마이그레이션

소스 데이터베이스와 대상 데이터베이스가 동기화된 상태로 유지되지만 소스 데이터베이스만 연결 애플리케이션의 트랜잭션을 처리하고 데이터는 대상 데이터베이스로 복제되는 데이터베이스 마이그레이션 방법입니다. 대상 데이터베이스는 마이그레이션 중 어떤 트랜잭션도 허용하지 않습니다.

집계 함수

행 그룹에서 연산을 수행하고 그룹에 대한 단일 반환값을 계산하는 SQL 함수입니다. 집계 함수의 예로는 및 등이 `SUM` 있습니다. `MAX`

AI

인공 지능을 참조하십시오.

AIOps

인공 지능 운영을 참조하십시오.

익명화

데이터세트에서 개인 정보를 영구적으로 삭제하는 프로세스입니다. 익명화는 개인 정보 보호에 도움이 될 수 있습니다. 익명화된 데이터는 더 이상 개인 데이터로 간주되지 않습니다.

안티 패턴

솔루션이 다른 솔루션보다 비생산적이거나 비효율적이거나 덜 효과적이어서 반복되는 문제에 자주 사용되는 솔루션입니다.

애플리케이션 제어

시스템을 멀웨어로부터 보호하기 위해 승인된 애플리케이션만 사용할 수 있는 보안 접근 방식입니다.

애플리케이션 포트폴리오

애플리케이션 구축 및 유지 관리 비용과 애플리케이션의 비즈니스 가치를 비롯하여 조직에서 사용하는 각 애플리케이션에 대한 세부 정보 모음입니다. 이 정보는 포트폴리오 검색 및 분석 프로세스의 핵심이며 마이그레이션, 현대화 및 최적화할 애플리케이션을 식별하고 우선순위를 정하는 데 도움이 됩니다.

인공 지능

컴퓨터 기술을 사용하여 학습, 문제 해결, 패턴 인식 등 일반적으로 인간과 관련된 인지 기능을 수행하는 것을 전문으로 하는 컴퓨터 과학 분야입니다. 자세한 내용은 What is Artificial Intelligence?를 참조하세요.

인공 지능 운영(AIOps)

기계 학습 기법을 사용하여 운영 문제를 해결하고, 운영 인시던트 및 사용자 개입을 줄이고, 서비스 품질을 높이는 프로세스입니다. AWS 마이그레이션 전략에서 AIOps가 사용되는 방법에 대한 자세한 내용은 운영 통합 가이드를 참조하세요.

비대칭 암호화

한 쌍의 키, 즉 암호화를 위한 퍼블릭 키와 복호화를 위한 프라이빗 키를 사용하는 암호화 알고리즘입니다. 퍼블릭 키는 복호화에 사용되지 않으므로 공유할 수 있지만 프라이빗 키에 대한 액세스는 엄격히 제한되어야 합니다.

원자성, 일관성, 격리성, 내구성(ACID)

오류, 정전 또는 기타 문제가 발생한 경우에도 데이터베이스의 데이터 유효성과 운영 신뢰성을 보장하는 소프트웨어 속성 세트입니다.

ABAC(속성 기반 액세스 제어)

부서, 직무, 팀 이름 등의 사용자 속성을 기반으로 세분화된 권한을 생성하는 방식입니다. 자세한 내용은 AWS Identity and Access Management(IAM) 설명서의 AWS용 ABAC란 무엇입니까?를 참조하세요.

신뢰할 수 있는 데이터 소스

가장 신뢰할 수 있는 정보 소스로 간주되는 기본 버전의 데이터를 저장하는 위치입니다. 익명화, 편집 또는 가명화와 같은 데이터 처리 또는 수정의 목적으로 신뢰할 수 있는 데이터 소스의 데이터를 다른 위치로 복사할 수 있습니다.

가용 영역

다른 가용 영역에 장애가 발생할 경우 분리되도록 설계된 AWS 리전 내의 개별적인 지점으로, 같은 리전 내의 다른 가용 영역에 비해 저렴하고 지연 시간이 짧은 네트워크 연결을 제공합니다.

AWS Cloud Adoption Framework(AWS CAF)

조직이 클라우드로 성공적으로 전환하기 위한 효율적이고 효과적인 계획을 개발하는 데 도움이 되는 AWS의 지침 및 모범 사례 프레임워크입니다. AWS CAF는 비즈니스, 사람, 거버넌스, 플랫폼, 보안 및 운영 관점이라는 6가지 중점 영역으로 지침을 구성합니다. 비즈니스, 사람 및 거버넌스 관점은 비즈니스 기술과 프로세스에 초점을 맞추고, 플랫폼, 보안 및 운영 관점은 전문 기술과 프로세스에 중점을 둡니다. 예를 들어, 사람 관점은 인사(HR), 직원 배치 기능 및 인력 관리를 담당하는 이해관계자를 대상으로 합니다. 이러한 관점에서 AWS CAF는 조직이 클라우드를 성공적으로 채택할 수 있도록 인력 개발, 교육 및 커뮤니케이션에 대한 지침을 제공합니다. 자세한 내용은 AWS CAF 웹 사이트와 AWS CAF 백서를 참조하세요.

AWS Workload Qualification Framework(AWS WQF)

데이터베이스 마이그레이션 워크로드를 평가하고, 마이그레이션 전략을 추천하고, 작업 추정치를 제공하는 도구입니다. AWS WQF는 AWS Schema Conversion Tool(AWS SCT)에 포함되어 있으며, 데이터베이스 스키마 및 코드 객체, 애플리케이션 코드, 종속성 및 성능 특성을 분석하고 평가 보고서를 제공합니다.

B

BCP

비즈니스 연속성 계획을 참조하십시오.

동작 그래프

리소스 동작과 시간 경과에 따른 상호 작용에 대한 통합된 대화형 뷰입니다. Amazon Detective에서 동작 그래프를 사용하여 실패한 로그온 시도, 의심스러운 API 호출 및 유사한 작업을 검사할 수 있습니다. 자세한 내용은 Detective 설명서의 Data in a behavior graph를 참조하세요.

빅 엔디안 시스템

가장 중요한 바이트를 먼저 저장하는 시스템입니다. 엔디안도 참조하십시오.

바이너리 분류

바이너리 결과(가능한 두 클래스 중 하나)를 예측하는 프로세스입니다. 예를 들어, ML 모델이 '이 이메일이 스팸인가요, 스팸이 아닌가요?', '이 제품은 책인가요, 자동차인가요?' 등의 문제를 예측해야 할 수 있습니다.

블룸 필터

요소가 세트의 멤버인지 여부를 테스트하는 데 사용되는 메모리 효율성이 높은 확률론적 데이터 구조입니다.

브랜치

코드 리포지토리의 포함된 영역입니다. 리포지토리에 생성되는 첫 번째 브랜치가 기본 브랜치입니다. 기존 브랜치에서 새 브랜치를 생성한 다음 새 브랜치에서 기능을 개발하거나 버그를 수정할 수 있습니다. 기능을 구축하기 위해 생성하는 브랜치를 일반적으로 기능 브랜치라고 합니다. 기능을 출시할 준비가 되면 기능 브랜치를 기본 브랜치에 다시 병합합니다. 자세한 내용은 브랜치 정보(GitHub 문서) 를 참조하십시오.

브레이크 글래스 액세스

예외적인 상황에서 승인된 프로세스를 통해 사용자가 일반적으로 액세스 권한이 없는 데이터에 빠르게 액세스할 수 AWS 계정 있는 수단입니다. 자세한 내용은 Well-Architected AWS 지침의 브레이크 글래스 절차 구현 표시기를 참조하십시오.

브라운필드 전략

사용자 환경의 기존 인프라 시스템 아키텍처에 브라운필드 전략을 채택할 때는 현재 시스템 및 인프라의 제약 조건을 중심으로 아키텍처를 설계합니다. 기존 인프라를 확장하는 경우 브라운필드 전략과 그린필드 전략을 혼합할 수 있습니다.

버퍼 캐시

가장 자주 액세스하는 데이터가 저장되는 메모리 영역입니다.

사업 역량

기업이 가치를 창출하기 위해 하는 일(예: 영업, 고객 서비스 또는 마케팅)입니다. 마이크로서비스 아키텍처 및 개발 결정은 비즈니스 역량에 따라 이루어질 수 있습니다. 자세한 내용은 백서의 컨테이너화된 마이크로서비스 실행AWS의 비즈니스 역량 중심의 구성화 섹션을 참조하세요.

비즈니스 연속성 계획(BCP)

대규모 마이그레이션과 같은 중단 이벤트가 운영에 미치는 잠재적 영향을 해결하고 비즈니스가 신속하게 운영을 재개할 수 있도록 지원하는 계획입니다.

C

CAF

클라우드 채택 프레임워크를 참조하십시오AWS.

CCoE

클라우드 센터 오브 엑설런스를 참조하십시오.

CDC

변경 데이터 캡처를 참조하십시오.

변경 데이터 캡처(CDC)

데이터베이스 테이블과 같은 데이터 소스의 변경 내용을 추적하고 변경 사항에 대한 메타데이터를 기록하는 프로세스입니다. 대상 시스템의 변경 내용을 감사하거나 복제하여 동기화를 유지하는 등의 다양한 용도로 CDC를 사용할 수 있습니다.

카오스 엔지니어링

시스템의 복원력을 테스트하기 위해 의도적으로 장애나 장애를 일으키는 이벤트를 발생시키는 행위 AWS Fault Injection Service(AWS FIS) 를 사용하여 AWS 워크로드에 스트레스를 주는 실험을 수행하고 응답을 평가할 수 있습니다.

CI/CD

지속적 통합 및 지속적 전달을 참조하십시오.

분류

예측을 생성하는 데 도움이 되는 분류 프로세스입니다. 분류 문제에 대한 ML 모델은 이산 값을 예측합니다. 이산 값은 항상 서로 다릅니다. 예를 들어, 모델이 이미지에 자동차가 있는지 여부를 평가해야 할 수 있습니다.

클라이언트측 암호화

데이터를 대상 AWS 서비스에서 수신하기 전에 로컬에서 암호화합니다.

클라우드 혁신 센터(CCoE)

클라우드 모범 사례 개발, 리소스 동원, 마이그레이션 타임라인 설정, 대규모 혁신을 통한 조직 선도 등 조직 전체에서 클라우드 채택 노력을 추진하는 다분야 팀입니다. 자세한 내용은 AWS 클라우드 엔터프라이즈 전략 블로그의 CCoE 게시물을 참조하세요.

클라우드 컴퓨팅

원격 데이터 스토리지와 IoT 디바이스 관리에 일반적으로 사용되는 클라우드 기술 클라우드 컴퓨팅은 일반적으로 엣지 컴퓨팅 기술과 연결됩니다.

클라우드 운영 모델

IT 조직에서 하나 이상의 클라우드 환경을 구축, 성숙화 및 최적화하는 데 사용되는 운영 모델입니다. 자세한 내용은 클라우드 운영 모델 구축을 참조하세요.

클라우드 채택 단계

조직이 AWS 클라우드로 마이그레이션할 때 일반적으로 거치는 4단계는 다음과 같습니다.

- 프로젝트 - 개념 증명 및 학습 목적으로 몇 가지 클라우드 관련 프로젝트 실행
- 기반 – 클라우드 채택 확장을 위한 기초 투자(예: 랜딩 존 생성, CCoE 정의, 운영 모델 구축)
- 마이그레이션 - 개별 애플리케이션 마이그레이션
- Re-invention - 제품 및 서비스 최적화와 클라우드 혁신

이러한 단계는 Stephen Orban이 AWS Cloud Enterprise Strategy Blog의 The Journey Toward Cloud-First & the Stages of Adoption 블로그 게시물에서 정의했습니다. AWS 마이그레이션 전략과 어떤 관련이 있는지에 대한 자세한 내용은 마이그레이션 준비 가이드를 참조하세요.

CMDB

구성 관리 데이터베이스를 참조하십시오.

코드 리포지토리

소스 코드와 설명서, 샘플, 스크립트 등의 기타 자산이 버전 관리 프로세스를 통해 저장되고 업데이트되는 위치입니다. 일반 클라우드 리포지토리에는 또는 이 포함됩니다 GitHub . AWS CodeCommit 코드의 각 버전을 브랜치라고 합니다. 마이크로서비스 구조에서 각 리포지토리는 단일 기능 전용입니다. 단일 CI/CD 파이프라인은 여러 리포지토리를 사용할 수 있습니다.

콜드 캐시

비어 있거나, 제대로 채워지지 않았거나, 오래되었거나 관련 없는 데이터를 포함하는 버퍼 캐시입니다. 주 메모리나 디스크에서 데이터베이스 인스턴스를 읽어야 하기 때문에 성능에 영향을 미치며, 이는 버퍼 캐시에서 읽는 것보다 느립니다.

콜드 데이터

거의 액세스되지 않고 일반적으로 과거 데이터인 데이터. 이런 종류의 데이터를 쿼리할 때는 일반적으로 느린 쿼리가 허용됩니다. 이 데이터를 성능이 낮고 비용이 저렴한 스토리지 계층 또는 클래스로 옮기면 비용을 절감할 수 있습니다.

컴퓨터 비전

기계가 이미지 속 사람, 장소, 사물을 인간 수준 이상의 정확도로 식별하는 데 사용하는 AI 분야입니다. 딥 러닝 모델로 구축되는 경우가 많으며 단일 이미지 또는 일련의 이미지에서 유용한 정보를 자동으로 추출, 분석, 분류 및 이해합니다.

구성 관리 데이터베이스(CMDB)

하드웨어 및 소프트웨어 구성 요소와 해당 구성을 포함하여 데이터베이스와 해당 IT 환경에 대한 정보를 저장하고 관리하는 리포지토리입니다. 일반적으로 마이그레이션의 포트폴리오 검색 및 분석 단계에서 CMDB의 데이터를 사용합니다.

규정 준수 팩

규정 준수 및 보안 검사를 사용자 지정하기 위해 조합할 수 있는 AWS Config 규칙 및 수정 작업 모음입니다. YAML 템플릿을 사용하여 AWS 계정 및 리전 또는 조직 전체에 단일 엔터티로 규정 준수 팩을 배포할 수 있습니다. 자세한 내용은 AWS Config 설명서의 Conformance packs를 참조하세요.

지속적 통합 및 지속적 전달(CI/CD)

소프트웨어 릴리스 프로세스의 소스, 빌드, 테스트, 스테이징 및 프로덕션 단계를 자동화하는 프로세스입니다. CI/CD는 일반적으로 파이프라인으로 설명됩니다. CI/CD를 통해 프로세스를 자동화하고, 생산성을 높이고, 코드 품질을 개선하고, 더 빠르게 제공할 수 있습니다. 자세한 내용은 지속적 전달의 이점을 참조하세요. CD는 지속적 배포를 의미하기도 합니다. 자세한 내용은 지속적 전달(Continuous Delivery)과 지속적인 개발을 참조하세요.

D

저장 데이터

스토리지에 있는 데이터와 같이 네트워크에 고정되어 있는 데이터입니다.

데이터 분류

중요도와 민감도를 기준으로 네트워크의 데이터를 식별하고 분류하는 프로세스입니다. 이 프로세스는 데이터에 대한 적절한 보호 및 보존 제어를 결정하는 데 도움이 되므로 사이버 보안 위험 관리

전략의 중요한 구성 요소입니다. 데이터 분류는 AWS Well-Architected Framework 보안 원칙의 구성 요소입니다. 자세한 내용은 데이터 분류를 참조하세요.

데이터 드리프트

프로덕션 데이터와 ML 모델 학습에 사용된 데이터 간의 상당한 차이 또는 시간 경과에 따른 입력 데이터의 의미 있는 변화. 데이터 드리프트는 ML 모델 예측의 전반적인 품질, 정확성 및 공정성을 저하시킬 수 있습니다.

전송 중 데이터

네트워크를 통과하고 있는 데이터입니다. 네트워크 리소스 사이를 이동 중인 데이터를 예로 들 수 있습니다.

데이터 최소화

꼭 필요한 데이터만 수집하고 처리하는 원칙입니다. AWS 클라우드에서 데이터 최소화를 실천하면 개인 정보 보호 위험, 비용 및 분석에 따른 탄소 발자국을 줄일 수 있습니다.

데이터 경계

신뢰할 수 있는 ID만 예상 네트워크에서 신뢰할 수 있는 리소스에 액세스하도록 하는 데 도움이 되는 AWS 환경 내 일련의 예방 가드레일입니다. 자세한 내용은 데이터 경계 구축을 참조하십시오. AWS

데이터 사전 처리

원시 데이터를 ML 모델이 쉽게 구문 분석할 수 있는 형식으로 변환하는 것입니다. 데이터를 사전 처리한다는 것은 특정 열이나 행을 제거하고 누락된 값, 일관성이 없는 값 또는 중복 값을 처리함을 의미할 수 있습니다.

데이터 출처

라이프사이클 전반에 걸쳐 데이터의 출처와 기록을 추적하는 프로세스(예: 데이터 생성, 전송, 저장 방법).

데이터 주체

데이터를 수집 및 처리하는 개인입니다.

데이터 웨어하우스

분석과 같은 비즈니스 인텔리전스를 지원하는 데이터 관리 시스템. 데이터 웨어하우스에는 일반적으로 대량의 과거 데이터가 포함되며 일반적으로 쿼리 및 분석에 사용됩니다.

데이터 정의 언어(DDL)

데이터베이스에서 테이블 및 객체의 구조를 만들거나 수정하기 위한 명령문 또는 명령입니다.

데이터베이스 조작 언어(DML)

데이터베이스에서 정보를 수정(삽입, 업데이트 및 삭제)하기 위한 명령문 또는 명령입니다.

DDL

데이터베이스 정의 언어를 참조하십시오.

딥 앙상블

예측을 위해 여러 딥 러닝 모델을 결합하는 것입니다. 딥 앙상블을 사용하여 더 정확한 예측을 얻거나 예측의 불확실성을 추정할 수 있습니다.

딥 러닝

여러 계층의 인공 신경망을 사용하여 입력 데이터와 관심 대상 변수 간의 매핑을 식별하는 ML 하위 분야입니다.

defense-in-depth

네트워크와 그 안의 데이터 기밀성, 무결성 및 가용성을 보호하기 위해 컴퓨터 네트워크 전체에 일련의 보안 메커니즘과 제어를 신중하게 계층화하는 정보 보안 접근 방식입니다. AWS에서 이 전략을 채택하면 AWS Organizations 구조의 다양한 계층에 여러 제어 기능을 추가하여 리소스를 보호할 수 있습니다. 예를 들어, defense-in-depth 접근 방식에는 다단계 인증, 네트워크 분할 및 암호화를 결합할 수 있습니다.

위임된 관리자

AWS Organizations에서 호환되는 서비스는 AWS 멤버 계정을 등록하여 조직의 계정을 관리하고 해당 서비스에 대한 권한을 관리할 수 있습니다. 이러한 계정을 해당 서비스의 위임된 관리자라고 합니다. 자세한 내용과 호환되는 서비스 목록은 AWS Organizations 설명서의 AWS Organizations와 함께 사용할 수 있는 AWS 서비스를 참조하세요.

배포

대상 환경에서 애플리케이션, 새 기능 또는 코드 수정 사항을 사용할 수 있도록 하는 프로세스입니다. 배포에는 코드 베이스의 변경 사항을 구현한 다음 애플리케이션 환경에서 해당 코드베이스를 구축하고 실행하는 작업이 포함됩니다.

개발 환경

환경을 참조하십시오.

탐지 제어

이벤트 발생 후 탐지, 기록 및 알림을 수행하도록 설계된 보안 제어입니다. 이러한 제어는 기존의 예방적 제어를 우회한 보안 이벤트를 알리는 2차 방어선입니다. 자세한 내용은 Implementing security controls on AWS의 Detective controls를 참조하세요.

개발 가치 흐름 매핑 (DVSM)

소프트웨어 개발 라이프사이클에서 속도와 품질에 부정적인 영향을 미치는 제약 조건을 식별하고 우선 순위를 지정하는 데 사용되는 프로세스입니다. DVSM은 원래 린 제조 방식을 위해 설계된 가치 흐름 매핑 프로세스를 확장합니다. 소프트웨어 개발 프로세스를 통해 가치를 창출하고 이동하는 데 필요한 단계와 팀에 중점을 둡니다.

디지털 트윈

건물, 공장, 산업 장비 또는 생산 라인과 같은 실제 시스템을 가상으로 표현한 것입니다. 디지털 트윈은 예측 유지 보수, 원격 모니터링, 생산 최적화를 지원합니다.

치수 표

스타 스키마에서 팩트 테이블의 양적 데이터에 대한 데이터 속성을 포함하는 작은 테이블입니다. 차원 테이블 속성은 일반적으로 텍스트처럼 동작하는 텍스트 필드 또는 불연속형 숫자입니다. 이러한 속성은 일반적으로 쿼리 제한, 필터링 및 결과 집합 레이블 지정에 사용됩니다.

재해

워크로드 또는 시스템이 기본 배포 위치에서 비즈니스 목표를 달성하지 못하게 방해하는 이벤트입니다. 이러한 이벤트는 자연재해, 기술적 오류, 의도하지 않은 구성 오류 또는 멀웨어 공격과 같은 사람의 행동으로 인한 결과일 수 있습니다.

재해 복구(DR)

재해로 인한 다운타임과 데이터 손실을 최소화하기 위해 사용하는 전략과 프로세스입니다. 자세한 내용은 AWS Well-Architected Framework의 AWS 기반 워크로로드의 재해 복구: 클라우드에서의 재해 복구를 참조하세요.

DML

데이터베이스 조작 언어를 참조하십시오.

도메인 기반 설계

구성 요소를 각 구성 요소가 제공하는 진화하는 도메인 또는 핵심 비즈니스 목표에 연결하여 복잡한 소프트웨어 시스템을 개발하는 접근 방식입니다. 이 개념은 에릭 에반스에 의해 그의 저서인 도

메인 기반 디자인: 소프트웨어 중심의 복잡성 해결(Boston: Addison-Wesley Professional, 2003)에서 소개되었습니다. Strangler Fig 패턴과 함께 도메인 기반 설계를 사용하는 방법에 대한 자세한 내용은 컨테이너 및 Amazon API Gateway를 사용하여 기존의 Microsoft ASP.NET(ASMX) 웹 서비스를 점진적으로 현대화하는 방법을 참조하세요.

DR

재해 복구를 참조하십시오.

드리프트 감지

기준 구성으로부터의 편차 추적 예를 들어 시스템 리소스의 편차를 감지하는 AWS CloudFormation 데 사용하거나 거버넌스 요구 사항 준수에 영향을 미칠 수 있는 착륙 지대의 변경을 탐지하는 AWS Control Tower 데 사용할 수 있습니다.

DVSM

개발 가치 흐름 매핑을 참조하십시오.

E

EDA

탐색적 데이터 분석을 참조하십시오.

엣지 컴퓨팅

IoT 네트워크의 엣지에서 스마트 디바이스의 컴퓨팅 성능을 개선하는 기술 클라우드 컴퓨팅과 비교할 때 엣지 컴퓨팅은 통신 대기 시간을 줄이고 응답 시간을 개선할 수 있습니다.

암호화

사람이 읽을 수 있는 일반 텍스트 데이터를 암호문으로 변환하는 컴퓨팅 프로세스입니다.

암호화 키

암호화 알고리즘에 의해 생성되는 무작위 비트의 암호화 문자열입니다. 키의 길이는 다양할 수 있으며 각 키는 예측할 수 없고 고유하게 설계되었습니다.

엔디안

컴퓨터 메모리에 바이트가 저장되는 순서입니다. 빅 엔디안 시스템은 가장 중요한 바이트를 먼저 저장합니다. 리틀 엔디안 시스템은 가장 덜 중요한 바이트를 먼저 저장합니다.

엔드포인트

서비스 엔드포인트를 참조하십시오.

엔드포인트 서비스

Virtual Private Cloud(VPC)에서 호스팅하여 다른 사용자와 공유할 수 있는 서비스입니다. AWS PrivateLink를 사용하여 엔드포인트 서비스를 생성하고 다른 AWS 계정 또는 AWS Identity and Access Management(IAM) 보안 주체에게 권한을 부여할 수 있습니다. 이러한 계정 또는 보안 주체는 인터페이스 VPC 엔드포인트를 생성하여 엔드포인트 서비스에 비공개로 연결할 수 있습니다. 자세한 내용은 Amazon Virtual Private Cloud(VPC) 설명서의 엔드포인트 서비스 생성을 참조하세요.

봉투 암호화

암호화 키를 다른 암호화 키로 암호화하는 프로세스입니다. 자세한 내용은 AWS Key Management Service(AWS KMS) 설명서의 Envelope encryption을 참조하세요.

환경

실행 중인 애플리케이션의 인스턴스입니다. 다음은 클라우드 컴퓨팅의 일반적인 환경 유형입니다.

- 개발 환경 - 애플리케이션 유지 관리를 담당하는 핵심 팀만 사용할 수 있는 실행 중인 애플리케이션의 인스턴스입니다. 개발 환경은 변경 사항을 상위 환경으로 승격하기 전에 테스트하는 데 사용됩니다. 이러한 유형의 환경을 테스트 환경이라고도 합니다.
- 하위 환경 - 초기 빌드 및 테스트에 사용되는 환경을 비롯한 애플리케이션의 모든 개발 환경입니다.
- 프로덕션 환경 - 최종 사용자가 액세스할 수 있는 실행 중인 애플리케이션의 인스턴스입니다. CI/CD 파이프라인에서 프로덕션 환경이 마지막 배포 환경입니다.
- 상위 환경 - 핵심 개발 팀 이외의 사용자가 액세스할 수 있는 모든 환경입니다. 프로덕션 환경, 프로덕션 이전 환경 및 사용자 수용 테스트를 위한 환경이 여기에 포함될 수 있습니다.

에픽

애자일 방법론에서 작업을 구성하고 우선순위를 정하는 데 도움이 되는 기능적 범주입니다. 에픽은 요구 사항 및 구현 작업에 대한 개괄적인 설명을 제공합니다. 예를 들어, AWS CAF 보안 에픽에는 ID 및 액세스 관리, 탐지 제어, 인프라 보안, 데이터 보호 및 인시던트 대응 등이 포함됩니다. AWS 마이그레이션 전략의 에픽에 대한 자세한 내용은 프로그램 구현 가이드를 참조하세요.

탐색 데이터 분석(EDA)

데이터 세트를 분석하여 주요 특성을 파악하는 프로세스입니다. 데이터를 수집 또는 집계한 다음 초기 조사를 수행하여 패턴을 찾고, 이상을 탐지하고, 가정을 확인합니다. EDA는 요약 통계를 계산하고 데이터 시각화를 생성하여 수행됩니다.

F

팩트 테이블

[스타 스키마의](#) 중앙 테이블. 비즈니스 운영에 대한 정량적 데이터를 저장합니다. 일반적으로 팩트 테이블에는 측정값이 포함된 열과 차원 테이블의 외부 키가 포함된 열 등 두 가지 유형의 열이 포함됩니다.

빨리 실패하세요

빈번하고 점진적인 테스트를 통해 개발 라이프사이클을 단축하는 철학. 이는 애자일 접근 방식의 중요한 부분입니다.

장애 격리 경계

장애 영향을 제한하고 워크로드의 복원력을 개선하는 데 도움이 되는 가용 영역AWS 리전, 컨트롤 플레인 또는 데이터 플레인과 같은 경계 AWS 클라우드 자세한 내용은 [AWS장애 격리](#) 경계를 참조하십시오.

기능 브랜치

[브랜치를](#) 참조하십시오.

기능

예측에 사용하는 입력 데이터입니다. 예를 들어, 제조 환경에서 기능은 제조 라인에서 주기적으로 캡처되는 이미지일 수 있습니다.

기능 중요도

모델의 예측에 특성이 얼마나 중요한지를 나타냅니다. 이는 일반적으로 SHAP(Shapley Additive Descriptions) 및 통합 그레디언트와 같은 다양한 기법을 통해 계산할 수 있는 수치 점수로 표현됩니다. 자세한 내용은 [다음을 AWS 사용한 기계 학습 모델 해석 가능성을](#) 참조하십시오.

기능 변환

추가 소스로 데이터를 보강하거나, 값을 조정하거나, 단일 데이터 필드에서 여러 정보 세트를 추출하는 등 ML 프로세스를 위해 데이터를 최적화하는 것입니다. 이를 통해 ML 모델이 데이터를 활용할 수 있습니다. 예를 들어, 날짜 '2021-05-27 00:15:37'을 '2021년', '5월', '목', '15일'로 분류하면 학습 알고리즘이 다양한 데이터 구성 요소와 관련된 미묘한 패턴을 학습하는 데 도움이 됩니다.

FGAC

[세분화된 액세스 제어를](#) 참조하십시오.

세분화된 액세스 제어(FGAC)

여러 조건을 사용하여 액세스 요청을 허용하거나 거부합니다.

플래시컷 마이그레이션

단계별 접근 방식 대신 변경 데이터 캡처를 통한 지속적인 데이터 복제를 통해 최단 시간에 데이터를 마이그레이션하는 데이터베이스 마이그레이션 방법입니다. 목표는 가동 중지 시간을 최소화하는 것입니다.

G

지리적 차단

지리적 제한을 참조하십시오.

지리적 제한(지리적 차단)

CloudFrontAmazon에서는 특정 국가의 사용자가 콘텐츠 배포에 액세스하지 못하도록 하는 옵션을 제공합니다. 허용 목록 또는 차단 목록을 사용하여 승인된 국가와 차단된 국가를 지정할 수 있습니다. 자세한 내용은 설명서의 콘텐츠의 지리적 배포 제한을 참조하십시오. CloudFront

Gitflow 워크플로

하위 환경과 상위 환경이 소스 코드 리포지토리의 서로 다른 브랜치를 사용하는 방식입니다. Gitflow 워크플로는 레거시로 간주되며 트렁크 기반 워크플로는 현대적이고 선호되는 접근 방식입니다.

브라운필드 전략

새로운 환경에서 기존 인프라의 부재 시스템 아키텍처에 대한 그린필드 전략을 채택할 때 브라운필드라고도 하는 기존 인프라와의 호환성 제한 없이 모든 새로운 기술을 선택할 수 있습니다. 기존 인프라를 확장하는 경우 브라운필드 전략과 그린필드 전략을 혼합할 수 있습니다.

가드레일

조직 단위(OU) 전체에서 리소스, 정책 및 규정 준수를 관리하는 데 도움이 되는 중요 규칙입니다. 예방 가드레일은 규정 준수 표준에 부합하도록 정책을 시행하며, 서비스 제어 정책과 IAM 권한 경계를 사용하여 구현됩니다. 탐지 가드레일은 정책 위반 및 규정 준수 문제를 감지하고 해결을 위한 알림을 생성하며, 이들은, Amazon AWS Config AWS Security Hub GuardDutyAWS Trusted Advisor, Amazon Inspector 및 사용자 지정 AWS Lambda 검사를 사용하여 구현됩니다.

H

하

고가용성을 확인하세요.

이기종 데이터베이스 마이그레이션

다른 데이터베이스 엔진을 사용하는 대상 데이터베이스로 소스 데이터베이스 마이그레이션(예: Oracle에서 Amazon Aurora로) 이기종 마이그레이션은 일반적으로 리아키텍트 작업의 일부이며 스키마를 변환하는 것은 복잡한 작업일 수 있습니다. AWS는 스키마 변환에 도움이 되는 AWS SCT를 제공합니다.

높은 가용성(HA)

문제나 재해 발생 시 개입 없이 지속적으로 운영할 수 있는 워크로드의 능력. HA 시스템은 자동으로 장애 조치되고, 지속적으로 고품질 성능을 제공하고, 성능에 미치는 영향을 최소화하면서 다양한 부하와 장애를 처리하도록 설계되었습니다.

히스토리언 현대화

제조 산업의 요구 사항을 더 잘 충족하도록 운영 기술(OT) 시스템을 현대화하고 업그레이드하는 데 사용되는 접근 방식입니다. 히스토리언은 공장의 다양한 출처에서 데이터를 수집하고 저장하는 데 사용되는 일종의 데이터베이스입니다.

동종 데이터베이스 마이그레이션

동일한 데이터베이스 엔진을 공유하는 대상 데이터베이스로 소스 데이터베이스 마이그레이션(예: Microsoft SQL Server에서 Amazon RDS for SQL Server로) 동종 마이그레이션은 일반적으로 리호스팅 또는 리플랫포밍 작업의 일부입니다. 네이티브 데이터베이스 유틸리티를 사용하여 스키마를 마이그레이션할 수 있습니다.

핫 데이터

자주 액세스하는 데이터(예: 실시간 데이터 또는 최근 번역 데이터). 일반적으로 이 데이터에는 빠른 쿼리 응답을 제공하기 위한 고성능 스토리지 계층 또는 클래스가 필요합니다.

핫픽스

프로덕션 환경의 중요한 문제를 해결하기 위한 긴급 수정입니다. 긴급성 때문에 핫픽스는 일반적으로 일반적인 DevOps 릴리스 워크플로 외부에서 만들어집니다.

하이퍼케어 기간

전환 직후 마이그레이션 팀이 문제를 해결하기 위해 클라우드에서 마이그레이션된 애플리케이션을 관리하고 모니터링하는 기간입니다. 일반적으로 이 기간은 1~4일입니다. 하이퍼케어 기간이 끝나면 마이그레이션 팀은 일반적으로 애플리케이션에 대한 책임을 클라우드 운영 팀에 넘깁니다.

I

IaC

인프라를 코드로 보세요.

ID 기반 정책

하나 이상의 IAM 보안 주체에 연결된 정책으로, AWS 클라우드 클라우드 환경 내에서 IAM 보안 주체의 권한을 정의합니다.

유휴 애플리케이션

90일 동안 평균 CPU 및 메모리 사용량이 5~20%인 애플리케이션입니다. 마이그레이션 프로젝트에서는 이러한 애플리케이션을 사용 중지하거나 온프레미스에 유지하는 것이 일반적입니다.

IIoT

산업용 사물 인터넷을 참조하십시오.

불변의 인프라

기존 인프라를 업데이트, 패치 또는 수정하는 대신 프로덕션 워크로드용 새 인프라를 배포하는 모델입니다. 변경 불가능한 인프라는 기본적으로 변경 가능한 인프라보다 더 일관되고 안정적이며 예측 가능합니다. 자세한 내용은 Well-Architected AWS 프레임워크의 변경 불가능한 인프라를 사용한 배포 모범 사례를 참조하십시오.

인바운드(수신) VPC

AWS 다중 계정 아키텍처에서 애플리케이션 외부의 네트워크 연결을 수락, 검사 및 라우팅하는 VPC입니다. AWS Security Reference Architecture에서는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정할 것을 권장합니다.

증분 마이그레이션

한 번에 전체 전환을 수행하는 대신 애플리케이션을 조금씩 마이그레이션하는 전환 전략입니다. 예를 들어, 처음에는 소수의 마이크로서비스나 사용자만 새 시스템으로 이동할 수 있습니다. 모든 것

이 제대로 작동하는지 확인한 후에는 레거시 시스템을 폐기할 수 있을 때까지 추가 마이크로서비스 또는 사용자를 점진적으로 이동할 수 있습니다. 이 전략을 사용하면 대규모 마이그레이션과 관련된 위험을 줄일 수 있습니다.

인프라

애플리케이션의 환경 내에 포함된 모든 리소스와 자산입니다.

코드형 인프라(IaC)

구성 파일 세트를 통해 애플리케이션의 인프라를 프로비저닝하고 관리하는 프로세스입니다. IaC는 새로운 환경의 반복 가능성, 신뢰성 및 일관성을 위해 인프라 관리를 중앙 집중화하고, 리소스를 표준화하고, 빠르게 확장할 수 있도록 설계되었습니다.

산업용 사물 인터넷(IIoT)

제조, 에너지, 자동차, 의료, 생명과학, 농업 등의 산업 부문에서 인터넷에 연결된 센서 및 디바이스의 사용 자세한 내용은 산업용 사물 인터넷(IoT) 디지털 트랜스포메이션 전략 구축을 참조하세요.

검사 VPC

AWS 다중 계정 아키텍처에서 동일한 또는 다른 AWS 리전에 있는 VPC, 인터넷 및 온프레미스 네트워크 간의 네트워크 트래픽 검사를 관리하는 중앙 집중식 VPC입니다. AWS Security Reference Architecture에서는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정할 것을 권장합니다.

사물 인터넷(IoT)

인터넷이나 로컬 통신 네트워크를 통해 다른 디바이스 및 시스템과 통신하는 센서 또는 프로세서가 내장된 연결된 물리적 객체의 네트워크 자세한 내용은 IoT란?을 참조하세요.

해석력

모델의 예측이 입력에 따라 어떻게 달라지는지를 사람이 이해할 수 있는 정도를 설명하는 기계 학습 모델의 특성입니다. 자세한 내용은 Machine learning model interpretability with AWS를 참조하세요.

IoT

사물 인터넷을 참조하십시오.

IT 정보 라이브러리(TIL)

IT 서비스를 제공하고 이러한 서비스를 비즈니스 요구 사항에 맞게 조정하기 위한 일련의 모범 사례 ITIL은 ITSM의 기반을 제공합니다.

IT 서비스 관리(TSM)

조직의 IT 서비스 설계, 구현, 관리 및 지원과 관련된 활동 클라우드 운영을 ITSM 도구와 통합하는 방법에 대한 자세한 내용은 운영 통합 가이드를 참조하세요.

ITIL

IT 정보 라이브러리를 참조하십시오.

ITSM

IT 서비스 관리를 참조하십시오.

L

레이블 기반 액세스 제어(LBAC)

사용자 및 데이터 자체에 각각 보안 레이블 값을 명시적으로 할당하는 필수 액세스 제어(MAC)를 구현한 것입니다. 사용자 보안 레이블과 데이터 보안 레이블 간의 교차 부분에 따라 사용자가 볼 수 있는 행과 열이 결정됩니다.

랜딩 존

랜딩 존은 확장성과 안전성을 갖춘 잘 설계된 다중 계정 AWS 환경입니다. 조직은 여기에서부터 보안 및 인프라 환경에 대한 확신을 가지고 워크로드와 애플리케이션을 신속하게 시작하고 배포할 수 있습니다. 랜딩 존에 대한 자세한 내용은 Setting up a secure and scalable multi-account AWS environment를 참조하세요.

대규모 마이그레이션

300대 이상의 서버 마이그레이션입니다.

LBAC

레이블 기반 액세스 제어를 참조하십시오.

최소 권한

작업을 수행하는 데 필요한 최소 권한을 부여하는 보안 모범 사례입니다. 자세한 내용은 IAM 설명서의 최소 권한 적용을 참조하세요.

리프트 앤드 시프트

7 R을 참조하십시오.

리틀 엔디안 시스템

가장 덜 중요한 바이트를 먼저 저장하는 시스템입니다. 엔디안도 참조하십시오.

하위 환경

환경 참조.

M

기계 학습(ML)

패턴 인식 및 학습에 알고리즘과 기법을 사용하는 인공 지능의 한 유형입니다. ML은 사물 인터넷(IoT) 데이터와 같은 기록된 데이터를 분석하고 학습하여 패턴을 기반으로 통계 모델을 생성합니다. 자세한 내용은 기계 학습을 참조하세요.

기본 브랜치

브랜치 참조.

매니지드 서비스

AWS 서비스인프라 계층, 운영 체제 및 플랫폼을 AWS 운영하며 사용자는 엔드포인트에 액세스하여 데이터를 저장하고 검색합니다. 관리형 서비스의 예로는 아마존 심플 스토리지 서비스 (Amazon S3) 와 아마존 DynamoDB가 있습니다. 이러한 서비스를 추상화된 서비스라고도 합니다.

MAP

Migration Acceleration 프로그램을 참조하십시오.

기구

도구를 만들고 도구 채택을 유도한 다음 결과를 검토하여 조정하는 전체 프로세스입니다. 메커니즘은 작동하면서 자체적으로 강화되고 개선되는 사이클입니다. 자세한 내용은 AWS Well-Architected 프레임워크에서의 메커니즘 구축을 참조하십시오.

멤버 계정

AWS Organizations의 조직에 속하는 관리 계정 이외의 모든 AWS 계정입니다. 하나의 계정은 한 번에 하나의 조직 멤버만 될 수 있습니다.

마이크로서비스

잘 정의된 API를 통해 통신하고 일반적으로 소규모 자체 팀이 소유하는 소규모 독립 서비스입니다. 예를 들어, 보험 시스템에는 영업, 마케팅 등의 비즈니스 역량이나 구매, 청구, 분석 등의 하위 영역

에 매핑되는 마이크로 서비스가 포함될 수 있습니다. 마이크로서비스의 이점으로 민첩성, 유연한 확장, 손쉬운 배포, 재사용 가능한 코드, 복원력 등이 있습니다. 자세한 내용은 AWS 서버리스 서비스를 사용하여 마이크로서비스 통합을 참조하세요.

마이크로서비스 아키텍처

각 애플리케이션 프로세스를 마이크로서비스로 실행하는 독립 구성 요소를 사용하여 애플리케이션을 구축하는 접근 방식입니다. 이러한 마이크로서비스는 경량 API를 사용하여 잘 정의된 인터페이스를 통해 통신합니다. 애플리케이션의 특정 기능에 대한 수요에 맞게 이 아키텍처의 각 마이크로서비스를 업데이트, 배포 및 조정할 수 있습니다. 자세한 내용은 AWS에서 마이크로서비스 구현을 참조하세요.

Migration Acceleration Program(MAP)

조직이 클라우드로 전환하기 위한 강력한 운영 기반을 구축하고 마이그레이션 초기 비용을 상쇄할 수 있도록 컨설팅 지원, 교육 및 서비스를 제공하는 AWS 프로그램입니다. MAP에는 레거시 마이그레이션을 체계적인 방식으로 실행하기 위한 마이그레이션 방법론과 일반적인 마이그레이션 시나리오를 자동화하고 가속화하는 도구 세트가 포함되어 있습니다.

대규모 마이그레이션

애플리케이션 포트폴리오의 대다수를 웨이브를 통해 클라우드로 이동하는 프로세스로, 각 웨이브에서 더 많은 애플리케이션이 더 빠른 속도로 이동합니다. 이 단계에서는 이전 단계에서 배운 모범 사례와 교훈을 사용하여 팀, 도구 및 프로세스의 마이그레이션 팩토리를 구현하여 자동화 및 민첩한 제공을 통해 워크로드 마이그레이션을 간소화합니다. 이것은 AWS 마이그레이션 전략의 세 번째 단계입니다.

마이그레이션 팩토리

자동화되고 민첩한 접근 방식을 통해 워크로드 마이그레이션을 간소화하는 다기능 팀입니다. 마이그레이션 팩토리 팀에는 일반적으로 운영, 비즈니스 분석가 및 소유자, 마이그레이션 엔지니어, 개발자, 스프린트에서 일하는 DevOps 전문가가 포함됩니다. 엔터프라이즈 애플리케이션 포트폴리오의 20~50%는 공장 접근 방식으로 최적화할 수 있는 반복되는 패턴으로 구성되어 있습니다. 자세한 내용은 이 콘텐츠 세트의 클라우드 마이그레이션 팩토리 가이드와 마이그레이션 팩토리에 대한 설명을 참조하세요.

마이그레이션 메타데이터

마이그레이션을 완료하는 데 필요한 애플리케이션 및 서버에 대한 정보 각 마이그레이션 패턴에는 서로 다른 마이그레이션 메타데이터 세트가 필요합니다. 마이그레이션 메타데이터의 예로는 대상 서브넷, 보안 그룹 및 AWS 계정이 있습니다.

마이그레이션 패턴

사용되는 마이그레이션 전략, 마이그레이션 대상, 마이그레이션 애플리케이션 또는 서비스를 자세히 설명하는 반복 가능한 마이그레이션 작업입니다. 예를 들어, AWS Application Migration Service를 사용하여 Amazon EC2로 마이그레이션을 리호스팅합니다.

Migration Portfolio Assessment(MPA)

AWS 클라우드로 마이그레이션의 비즈니스 사례를 검증하기 위한 정보를 제공하는 온라인 도구입니다. MPA는 상세한 포트폴리오 평가(서버 적정 규모 조정, 가격 책정, TCO 비교, 마이그레이션 비용 분석)와 마이그레이션 계획(애플리케이션 데이터 분석 및 데이터 수집, 애플리케이션 그룹화, 마이그레이션 우선순위 지정, 웨이브 계획)을 제공합니다. MPA 도구(로그인 필요)는 모든 AWS 컨설턴트와 APN 파트너 컨설턴트에게 무료로 제공됩니다.

마이그레이션 준비 상태 평가(MRA)

AWS CAF를 사용하여 조직의 클라우드 준비 상태에 대한 인사이트를 얻고, 장단점을 파악하고, 파악된 격차를 해소하기 위한 실행 계획을 수립하는 프로세스입니다. 자세한 내용은 마이그레이션 준비 가이드를 참조하세요. MRA는 AWS 마이그레이션 전략의 첫 번째 단계입니다.

마이그레이션 전략

워크로드를 AWS 클라우드로 마이그레이션하는 데 사용되는 접근 방식입니다. 자세한 내용은 이 용어집의 7R 항목을 참조하고 대규모 마이그레이션을 가속화하기 위한 조직 동원을 참조하십시오.

ML

기계 학습을 참조하십시오.

MPA

마이그레이션 포트폴리오 평가를 참조하십시오.

현대화

비용을 절감하고 효율성을 높이고 혁신을 활용하기 위해 구식(레거시 또는 모놀리식) 애플리케이션과 해당 인프라를 클라우드의 민첩하고 탄력적이고 가용성이 높은 시스템으로 전환하는 것입니다. 자세한 내용은 AWS 클라우드에서 애플리케이션 현대화 전략을 참조하세요.

현대화 준비 상태 평가

조직 애플리케이션의 현대화 준비 상태를 파악하고, 이점, 위험 및 종속성을 식별하고, 조직이 해당 애플리케이션의 향후 상태를 얼마나 잘 지원할 수 있는지를 확인하는 데 도움이 되는 평가입니다. 평가 결과는 대상 아키텍처의 청사진, 현대화 프로세스의 개발 단계와 마일스톤을 자세히 설명하는

로드맵 및 파악된 격차를 해소하기 위한 실행 계획입니다. 자세한 내용은 AWS 클라우드에서 애플리케이션의 현대화 준비 상태 평가를 참조하세요.

모놀리식 애플리케이션(모놀리식 유형)

긴밀하게 연결된 프로세스를 사용하여 단일 서비스로 실행되는 애플리케이션입니다. 모놀리식 애플리케이션에는 몇 가지 단점이 있습니다. 한 애플리케이션 기능에 대한 수요가 급증하면 전체 아키텍처 규모를 조정해야 합니다. 코드 베이스가 커지면 모놀리식 애플리케이션의 기능을 추가하거나 개선하는 것도 더 복잡해집니다. 이러한 문제를 해결하기 위해 마이크로서비스 아키텍처를 사용할 수 있습니다. 자세한 내용은 마이크로서비스로 모놀리식 유형 분해를 참조하세요.

멀티클래스 분류

여러 클래스에 대한 예측(2개 이상의 결과 중 하나 예측)을 생성하는 데 도움이 되는 프로세스입니다. 예를 들어, ML 모델이 '이 제품은 책인가요, 자동차인가요, 휴대폰인가요?' 또는 '이 고객이 가장 관심을 갖는 제품 범주는 무엇인가요?'라고 물을 수 있습니다.

변경 가능한 인프라

프로덕션 워크로드를 위해 기존 인프라를 업데이트하고 수정하는 모델입니다. 일관성, 안정성 및 예측 가능성을 개선하기 위해 AWS Well-Architected Framework는 변경 불가능한 인프라를 모범 사례로 사용할 것을 권장합니다.

O

OAC

원본 액세스 제어를 참조하십시오.

좋아요

원본 액세스 ID를 참조하십시오.

OCM

조직 변경 관리를 참조하십시오.

오프라인 마이그레이션

마이그레이션 프로세스 중 소스 워크로드가 중단되는 마이그레이션 방법입니다. 이 방법은 가동 중지 증가를 수반하며 일반적으로 작고 중요하지 않은 워크로드에 사용됩니다.

OI

운영 통합을 참조하십시오.

안녕하세요.

운영 수준 계약을 참조하십시오.

온라인 마이그레이션

소스 워크로드를 오프라인 상태로 전환하지 않고 대상 시스템에 복사하는 마이그레이션 방법입니다. 워크로드에 연결된 애플리케이션은 마이그레이션 중에도 계속 작동할 수 있습니다. 이 방법은 가동 중지 차단 또는 최소화를 수반하며 일반적으로 중요한 프로덕션 워크로드에 사용됩니다.

운영 수준 협약(OLA)

서비스 수준에 관한 계약(SLA)을 지원하기 위해 직무 IT 그룹이 서로에게 제공하기로 약속한 내용을 명확히 하는 계약입니다.

운영 준비 상태 검토 (ORR)

인시던트 및 발생 가능한 실패의 범위를 이해, 평가, 예방 또는 줄이는 데 도움이 되는 질문 및 관련 모범 사례로 구성된 체크리스트입니다. 자세한 내용은 Well-Architected AWS 프레임워크의 운영 준비 상태 검토 (ORR) 를 참조하십시오.

운영 통합(OI)

클라우드에서 운영을 현대화하는 프로세스로 준비 계획, 자동화 및 통합을 수반합니다. 자세한 내용은 운영 통합 가이드를 참조하세요.

조직 트레일

AWS Organizations의 조직 내 모든 AWS 계정에 대한 모든 이벤트를 로깅하기 위해 AWS CloudTrail에서 생성하는 트레일입니다. 이 트레일은 조직에 속한 각 AWS 계정에 생성되고 각 계정의 활동을 추적합니다. 자세한 내용은 설명서의 조직을 위한 트레일 만들기를 참조하십시오. CloudTrail

조직 변경 관리(OCM)

사람, 문화 및 리더십 관점에서 중대하고 파괴적인 비즈니스 혁신을 관리하기 위한 프레임워크입니다. OCM은 변화 채택을 가속화하고, 과도기적 문제를 해결하고, 문화 및 조직적 변화를 주도함으로써 조직이 새로운 시스템 및 전략을 준비하고 전환할 수 있도록 지원합니다. AWS 마이그레이션 전략에서는 클라우드 채택 프로젝트에 필요한 변화 속도 때문에 이 프레임워크를 인력 가속화라고 합니다. 자세한 내용은 사용 가이드를 참조하세요.

오리진 액세스 제어(OAC)

CloudFront에서는 Amazon Simple Storage Service (Amazon S3) 콘텐츠의 보안을 위해 액세스를 제한하는 향상된 옵션을 제공합니다. OAC는 모든 AWS 리전의 모든 S3 버킷, AWS KMS(SSE-KMS)를 사용한 서버측 암호화, S3 버킷에 대한 동적 PUT 및 DELETE 요청을 지원합니다.

오리진 액세스 ID(OAI)

CloudFront에서는 Amazon S3 콘텐츠 보안을 위해 액세스를 제한하는 옵션입니다. OAI를 사용하면 Amazon S3가 인증할 수 있는 보안 주체를 CloudFront 생성합니다. 인증된 보안 주체는 특정 배포를 통해서만 S3 버킷의 콘텐츠에 액세스할 수 있습니다. CloudFront 더 세분화되고 향상된 액세스 제어를 제공하는 OAC도 참조하세요.

또는

운영 준비 상태 검토를 참조하십시오.

아웃바운드(송신) VPC

AWS 다중 계정 아키텍처에서 애플리케이션 내에서 시작되는 네트워크 연결을 처리하는 VPC입니다. AWS Security Reference Architecture에서는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정할 것을 권장합니다.

P

권한 경계

사용자나 역할이 가질 수 있는 최대 권한을 설정하기 위해 IAM 보안 주체에 연결되는 IAM 관리 정책입니다. 자세한 내용은 IAM 설명서의 권한 경계를 참조하세요.

개인 식별 정보(PII)

직접 보거나 다른 관련 데이터와 함께 짝을 지을 때 개인의 신원을 합리적으로 추론하는 데 사용할 수 있는 정보입니다. PII의 예로는 이름, 주소, 연락처 정보 등이 있습니다.

PII

개인 식별 정보를 참조하십시오.

플레이북

클라우드에서 핵심 운영 기능을 제공하는 등 마이그레이션과 관련된 작업을 캡처하는 일련의 사전 정의된 단계입니다. 플레이북은 스크립트, 자동화된 런북 또는 현대화된 환경을 운영하는 데 필요한 프로세스나 단계 요약의 형태를 취할 수 있습니다.

정책

권한을 정의 (ID 기반 정책 참조) 하거나, 액세스 조건을 지정 (리소스 기반 정책 참조) 하거나, 조직의 모든 계정에 대한 최대 권한을 정의 AWS Organizations (서비스 제어 정책 참조) 할 수 있는 개체입니다.

다국어 지속성

데이터 액세스 패턴 및 기타 요구 사항을 기반으로 독립적으로 마이크로서비스의 데이터 스토리지 기술 선택. 마이크로서비스가 동일한 데이터 스토리지 기술을 사용하는 경우 구현 문제가 발생하거나 성능이 저하될 수 있습니다. 요구 사항에 가장 적합한 데이터 스토어를 사용하면 마이크로서비스를 더 쉽게 구현하고 성능과 확장성을 높일 수 있습니다. 자세한 내용은 마이크로서비스에서 데이터 지속성 활성화를 참조하세요.

포트폴리오 평가

마이그레이션을 계획하기 위해 애플리케이션 포트폴리오를 검색 및 분석하고 우선순위를 정하는 프로세스입니다. 자세한 내용은 마이그레이션 준비 상태 평가를 참조하세요.

조건자

일반적으로 조항에 있는 `true` `false` OR를 반환하는 쿼리 조건입니다. `WHERE`

조건부 푸시다운

전송하기 전에 쿼리의 데이터를 필터링하는 데이터베이스 쿼리 최적화 기법입니다. 이렇게 하면 관계형 데이터베이스에서 검색하고 처리해야 하는 데이터의 양이 줄어들고 쿼리 성능이 향상됩니다.

예방적 제어

이벤트 발생을 방지하도록 설계된 보안 제어입니다. 이 제어는 네트워크에 대한 무단 액세스나 원치 않는 변경을 방지하는 데 도움이 되는 1차 방어선입니다. 자세한 내용은 Implementing security controls on AWS의 Preventative controls를 참조하세요.

보안 주체

작업을 수행하고 리소스에 액세스할 수 있는 AWS의 객체입니다. 이 엔터티는 일반적으로 AWS 계정의 루트 사용자, IAM 역할 또는 사용자입니다. 자세한 내용은 IAM 설명서의 역할 용어 및 개념의 보안 주체를 참조하세요.

개인 정보 보호 중심 설계

전체 엔지니어링 프로세스에서 개인 정보를 고려하는 시스템 엔지니어링에서의 접근 방식입니다.

프라이빗 호스팅 영역

Amazon Route 53에서 하나 이상의 VPC 내 도메인과 하위 도메인에 대한 DNS 쿼리에 응답하는 방법에 대한 정보가 담긴 컨테이너입니다. 자세한 내용은 Route 53 설명서의 프라이빗 호스팅 영역 작업을 참조하세요.

사전 예방적 제어

규정을 준수하지 않는 리소스의 배포를 방지하도록 설계된 보안 제어입니다. 이러한 컨트롤은 리소스를 프로비저닝하기 전에 리소스를 스캔합니다. 리소스가 컨트롤과 호환되지 않으면 프로비저닝되지 않습니다. 자세한 내용은 AWS Control Tower 설명서의 컨트롤 참조 안내서를 참조하고 보안 제어 구현의 사전 제어를 참조하십시오. AWS

프로덕션 환경

환경을 참조하십시오.

가명화

데이터세트의 개인 식별자를 자리 표시자 값으로 바꾸는 프로세스입니다. 가명화는 개인 정보를 보호하는 데 도움이 될 수 있습니다. 가명화된 데이터는 여전히 개인 데이터로 간주됩니다.

Q

쿼리 계획

SQL 관계형 데이터베이스 시스템의 데이터에 액세스하는 데 사용되는 일련의 단계 (예: 지침).

쿼리 계획 회귀

데이터베이스 서비스 최적화 프로그램이 데이터베이스 환경을 변경하기 전보다 덜 최적의 계획을 선택하는 경우입니다. 통계, 제한 사항, 환경 설정, 쿼리 파라미터 바인딩 및 데이터베이스 엔진 업데이트의 변경으로 인해 발생할 수 있습니다.

R

RACI 매트릭스

RACI (책임, 책임, 상담, 정보 제공) 를 참조하십시오.

랜섬웨어

결제가 완료될 때까지 컴퓨터 시스템이나 데이터에 대한 액세스를 차단하도록 설계된 악성 소프트웨어입니다.

RASCI 매트릭스

책임, 책임, 상담, 정보 제공 (RACI) 을 참조하십시오.

RCAC

행 및 열 액세스 제어를 참조하십시오.

읽기 전용 복제본

읽기 전용 용도로 사용되는 데이터베이스의 사본입니다. 쿼리를 읽기 전용 복제본으로 라우팅하여 기본 데이터베이스의 로드를 줄일 수 있습니다.

재설계

7 R을 참조하십시오.

Recovery Point Objective(RPO)

마지막 데이터 복구 시점 이후 허용되는 시간입니다. 이에 따라 마지막 복구 시점과 서비스 중단 시점 사이에 허용 가능한 데이터 손실이 결정됩니다.

Recovery Time Objective(RTO)

서비스 중단과 서비스 복구 사이에 허용되는 최대 지연 시간입니다.

리팩터링

7 R을 참조하십시오.

리전

AWS 리소스를 지리적 영역에 모아 놓은 것입니다. 각 AWS 리전은 내결함성, 안정성 및 복원력을 제공하기 위해 격리되어 있으며 다른 리전과는 독립적입니다. 자세한 내용은 AWS 일반 참조의 AWS 리전 관리를 참조하세요.

회귀

숫자 값을 예측하는 ML 기법입니다. 예를 들어, '이 집은 얼마에 팔릴까?'라는 문제를 풀기 위해 ML 모델은 선형 회귀 모델을 사용하여 주택에 대해 알려진 사실(예: 면적)을 기반으로 주택의 매매 가격을 예측할 수 있습니다.

리호스팅

7 R을 참조하십시오.

release

배포 프로세스에서 변경 사항을 프로덕션 환경으로 승격시키는 행위입니다.

고쳐 놓다

7 R을 참조하십시오.

리플랫포밍

7 R을 참조하십시오.

환매

7 R을 참조하십시오.

리소스 기반 정책

Amazon S3 버킷, 엔드포인트, 암호화 키 등의 리소스에 연결된 정책입니다. 이 유형의 정책은 액세스가 허용된 보안 주체, 지원되는 작업 및 충족해야 하는 기타 조건을 지정합니다.

RACI(Responsible, Accountable, Consulted, Informed) 매트릭스

마이그레이션 활동 및 클라우드 운영에 참여하는 모든 당사자의 역할과 책임을 정의하는 매트릭스입니다. 매트릭스 이름은 매트릭스에 정의된 책임 유형에서 파생됩니다: 실무 담당자 (R), 의사 결정권자 (A), 업무 수행 조언자 (C), 결과 통보 대상자 (I). 지원자는 (S) 선택사항입니다. 지원자를 포함하면 매트릭스를 RASCI 매트릭스라고 하고, 지원자를 제외하면 RACI 매트릭스라고 합니다.

대응 제어

보안 기준에서 벗어나거나 부정적인 이벤트를 해결하도록 설계된 보안 제어입니다. 자세한 내용은 Implementing security controls on AWS의 Responsive controls를 참조하세요.

retain

7 R을 참조하십시오.

은퇴

7 R을 참조하십시오.

회전

공격자가 자격 증명에 액세스하는 것을 더 어렵게 만들기 위해 암호를 주기적으로 업데이트하는 프로세스입니다.

행 및 열 액세스 제어(RCAC)

액세스 규칙이 정의된 기본적이고 유연한 SQL 표현식을 사용합니다. RCAC는 행 권한과 열 마스크로 구성됩니다.

RPO

복구 지점 목표를 참조하십시오.

RPO

복구 시간 목표를 참조하십시오.

런북

특정 작업을 수행하는 데 필요한 일련의 수동 또는 자동 절차입니다. 일반적으로 오류율이 높은 반복 작업이나 절차를 간소화하기 위해 런북을 만듭니다.

S

SAML 2.0

많은 ID 제공업체 (IdPs) 가 사용하는 개방형 표준입니다. 이 기능은 페더레이션형 AWS Single Sign-On(SSO)을 활성화하므로 조직의 모든 멤버에 대해 IAM 사용자를 생성하지 않아도 사용자가 AWS Management Console에 로그인하거나 AWS API 작업을 직접적으로 호출할 수 있습니다. SAML 2.0 기반 페더레이션에 대한 자세한 내용은 IAM 설명서의 SAML 2.0 기반 페더레이션 정보를 참조하세요.

SCP

서비스 제어 정책을 참조하십시오.

secret

에는 AWS Secrets Manager 암호화된 형태로 저장하는 비밀번호나 사용자 자격 증명과 같은 기밀 또는 제한된 정보. 비밀 값과 해당 메타데이터로 구성됩니다. 비밀 값은 바이너리, 단일 문자열 또는 여러 문자열일 수 있습니다. 자세한 내용은 Secrets Manager 문서의 시크릿을 참조하십시오.

보안 제어

위협 행위자가 보안 취약성을 악용하는 능력을 방지, 탐지 또는 감소시키는 기술적 또는 관리적 가드레일입니다. 보안 제어에는 예방적, 탐정적, 대응적, 사전 예방적 등 네 가지 기본 유형이 있습니다.

보안 강화

공격 표면을 줄여 공격에 대한 저항력을 높이는 프로세스입니다. 더 이상 필요하지 않은 리소스 제거, 최소 권한 부여의 보안 모범 사례 구현, 구성 파일의 불필요한 기능 비활성화 등의 작업이 여기에 포함될 수 있습니다.

보안 정보 및 이벤트 관리(SIEM) 시스템

보안 정보 관리(SIM)와 보안 이벤트 관리(SEM) 시스템을 결합하는 도구 및 서비스입니다. SIEM 시스템은 서버, 네트워크, 디바이스 및 기타 소스에서 데이터를 수집, 모니터링 및 분석하여 위협과 보안 침해를 탐지하고 알림을 생성합니다.

보안 대응 자동화

보안 이벤트에 자동으로 대응하거나 보안 이벤트를 해결하도록 설계된 사전 정의되고 프로그래밍된 조치입니다. 이러한 자동화는 보안 모범 사례를 구현하는 데 도움이 되는 탐지 또는 대응형 보안 제어 역할을 합니다. AWS 자동 응답 조치의 예로는 VPC 보안 그룹 수정, Amazon EC2 인스턴스 패치, 자격 증명 교체 등이 있습니다.

서버측 암호화

데이터를 수신하는 AWS 서비스가 대상에서 데이터를 암호화하는 것입니다.

서비스 제어 정책(SCP)

AWS Organizations에 속한 조직의 모든 계정에 대한 권한을 중앙 집중식으로 제어하는 정책입니다. SCP는 관리자가 사용자 또는 역할에 위임할 수 있는 작업에 대해 제한을 설정하거나 가드레일을 정의합니다. SCP를 허용 목록 또는 거부 목록으로 사용하여 허용하거나 금지할 서비스 또는 작업을 지정할 수 있습니다. 자세한 내용은 AWS Organizations 설명서의 서비스 제어 정책을 참조하세요.

서비스 엔드포인트

AWS 서비스에 대한 진입점의 URL입니다. 엔드포인트를 사용하여 대상 서비스에 프로그래밍 방식으로 연결할 수 있습니다. 자세한 내용은 AWS 일반 참조의 AWS 서비스 엔드포인트를 참조하세요.

서비스 수준에 관한 계약(SLA)

IT 팀이 고객에게 제공하기로 약속한 내용(예: 서비스 가동 시간 및 성능)을 명시한 계약입니다.

서비스 수준 지표 (SLI)

오류율, 가용성 또는 처리량과 같은 서비스의 성능 측면을 측정하는 것입니다.

서비스 수준 목표 (SLO)

서비스 수준 지표로 측정되는 서비스 상태를 나타내는 대상 지표입니다.

공동 책임 모델

클라우드 보안 및 규정 준수를 위해 AWS와 공유하는 책임을 설명하는 모델입니다. AWS는 클라우드의 보안을 담당하고, 사용자는 클라우드에서 보안을 담당합니다. 자세한 내용은 공동 책임 모델을 참조하세요.

시엠

보안 정보 및 이벤트 관리 시스템을 참조하십시오.

단일 장애 지점 (SPOF)

응용 프로그램의 중요한 단일 구성 요소에서 발생한 오류로 인해 시스템이 중단될 수 있습니다.

SLA

SLA (서비스 수준 계약) 를 참조하십시오.

SLI

서비스 수준 표시기를 참조하십시오.

SLO

서비스 수준 목표를 참조하십시오.

split-and-seed 모델

현대화 프로젝트를 확장하고 가속화하기 위한 패턴입니다. 새로운 기능과 제품 릴리스가 정의되면 핵심 팀이 분할되어 새로운 제품 팀이 만들어집니다. 이를 통해 조직의 역량과 서비스 규모를 조정하고, 개발자 생산성을 개선하고, 신속한 혁신을 지원할 수 있습니다. 자세한 내용은 의 애플리케이션 현대화를 위한 단계별 접근 방식을 참조하십시오. AWS 클라우드

SPOF

단일 장애 지점 보기

스타 스키마

하나의 큰 팩트 테이블을 사용하여 트랜잭션 또는 측정 데이터를 저장하고 하나 이상의 작은 차원 테이블을 사용하여 데이터 속성을 저장하는 데이터베이스 구성 구조입니다. 이 구조는 데이터 웨어하우스에서 사용하거나 비즈니스 인텔리전스 용도로 설계되었습니다.

Strangler Fig 패턴

레거시 시스템을 폐기할 수 있을 때까지 시스템 기능을 점진적으로 다시 작성하고 교체하여 모놀리식 시스템을 현대화하기 위한 접근 방식. 이 패턴은 무화과 덩굴이 나무로 자라 결국 숙주를 압도하고 대체하는 것과 비슷합니다. Martin Fowler가 모놀리식 시스템을 다시 작성할 때 위험을 관리하는 방법으로 이 패턴을 도입했습니다. 이 패턴을 적용하는 방법의 예는 컨테이너 및 Amazon API Gateway를 사용하여 기존의 Microsoft ASP.NET(ASMX) 웹 서비스를 점진적으로 현대화하는 방법을 참조하세요.

서브넷

VPC의 IP 주소 범위입니다. 서브넷은 단일 가용 영역에 상주해야 합니다.

대칭 암호화

동일한 키를 사용하여 데이터를 암호화하고 복호화하는 암호화 알고리즘입니다.

합성 테스트

잠재적 문제를 감지하거나 성능을 모니터링하기 위해 사용자 상호 작용을 시뮬레이션하는 방식으로 시스템을 테스트합니다. Amazon CloudWatch Synthetics를 사용하여 이러한 테스트를 생성할 수 있습니다.

T

tags

AWS 리소스를 구성하는 메타데이터 역할을 하는 키-값 페어입니다. 태그를 사용하면 리소스를 손쉽게 관리, 식별, 정리, 검색 및 필터링할 수 있습니다. 자세한 내용은 AWS 리소스에 태그 지정을 참조하십시오.

대상 변수

지도 ML에서 예측하려는 값으로, 결과 변수라고도 합니다. 예를 들어, 제조 설정에서 대상 변수는 제품 결함일 수 있습니다.

작업 목록

런북을 통해 진행 상황을 추적하는 데 사용되는 도구입니다. 작업 목록에는 런북의 개요와 완료해야 할 일반 작업 목록이 포함되어 있습니다. 각 일반 작업에 대한 예상 소요 시간, 소유자 및 진행 상황이 작업 목록에 포함됩니다.

테스트 환경

환경을 참조하십시오.

훈련

ML 모델이 학습할 수 있는 데이터를 제공하는 것입니다. 훈련 데이터에는 정답이 포함되어야 합니다. 학습 알고리즘은 훈련 데이터에서 대상(예측하려는 답)에 입력 데이터 속성을 매핑하는 패턴을 찾고, 이러한 패턴을 캡처하는 ML 모델을 출력합니다. 그런 다음 ML 모델을 사용하여 대상을 모르는 새 데이터에 대한 예측을 할 수 있습니다.

전송 게이트웨이

VPC와 온프레미스 네트워크를 상호 연결하는 데 사용할 수 있는 네트워크 전송 허브입니다. 자세한 내용은 AWS Transit Gateway 설명서의 Transit Gateway란 무엇입니까? 섹션을 참조하세요.

트렁크 기반 워크플로

개발자가 기능 브랜치에서 로컬로 기능을 구축하고 테스트한 다음 해당 변경 사항을 기본 브랜치에 병합하는 접근 방식입니다. 이후 기본 브랜치는 개발, 프로덕션 이전 및 프로덕션 환경에 순차적으로 구축됩니다.

신뢰할 수 있는 액세스

사용자를 대신하여 AWS Organizations의 조직과 해당 계정에서 작업을 수행하도록 지정하는 서비스에 권한을 부여하는 것입니다. 신뢰할 수 있는 서비스는 필요할 때 각 계정에 서비스 연결 역할을 생성하여 관리 작업을 수행합니다. 자세한 내용은 AWS Organizations 설명서의 다른 AWS 서비스와 함께 AWS Organizations 사용을 참조하세요.

튜닝

ML 모델의 정확도를 높이기 위해 훈련 프로세스의 측면을 여러 변경하는 것입니다. 예를 들어, 레이블링 세트를 생성하고 레이블을 추가한 다음 다양한 설정에서 이러한 단계를 여러 번 반복하여 모델을 최적화하는 방식으로 ML 모델을 훈련할 수 있습니다.

피자 두 판 팀

피자 두 판이면 먹을 수 있는 소규모 DevOps 팀이죠. 피자 두 판 팀 규모는 소프트웨어 개발에 있어 가능한 최상의 공동 작업 기회를 보장합니다.

U

불확실성

예측 ML 모델의 신뢰성을 저해할 수 있는 부정확하거나 불완전하거나 알려지지 않은 정보를 나타내는 개념입니다. 불확실성에는 두 가지 유형이 있습니다. 인식론적 불확실성은 제한적이고 불완전한 데이터에 의해 발생하는 반면, 우연한 불확실성은 데이터에 내재된 노이즈와 무작위성에 의해 발생합니다. 자세한 내용은 Quantifying uncertainty in deep learning systems 가이드를 참조하세요.

차별화되지 않은 작업

애플리케이션을 만들고 운영하는 데 필요하지만 최종 사용자에게 직접적인 가치를 제공하거나 경쟁 우위를 제공하지 못하는 작업을 헤비 리프팅이라고도 합니다. 차별화되지 않은 작업의 예로는 조달, 유지보수, 용량 계획 등이 있습니다.

상위 환경

환경을 보세요.

V

정리

스토리지를 회수하고 성능을 향상시키기 위해 증분 업데이트 후 정리 작업을 수반하는 데이터베이스 유지 관리 작업입니다.

버전 제어

리포지토리의 소스 코드 변경과 같은 변경 사항을 추적하는 프로세스 및 도구입니다.

VPC 피어링

프라이빗 IP 주소를 사용하여 트래픽을 라우팅할 수 있게 하는 두 VPC 간의 연결입니다. 자세한 내용은 Amazon VPC 설명서의 VPC 피어링이란?을 참조하세요.

취약성

시스템 보안을 손상시키는 소프트웨어 또는 하드웨어 결함입니다.

W

웜 캐시

자주 액세스하는 최신 관련 데이터를 포함하는 버퍼 캐시입니다. 버퍼 캐시에서 데이터베이스 인스턴스를 읽을 수 있기 때문에 주 메모리나 디스크에서 읽는 것보다 빠릅니다.

웜 데이터

자주 액세스하지 않는 데이터입니다. 이런 종류의 데이터를 쿼리할 때는 일반적으로 적절히 느린 쿼리가 허용됩니다.

윈도우 함수

현재 레코드와 어떤 식으로든 관련된 행 그룹에 대해 계산을 수행하는 SQL 함수입니다. 윈도우 함수는 이동 평균을 계산하거나 현재 행의 상대적 위치를 기반으로 행 값에 액세스하는 등의 작업을 처리하는 데 유용합니다.

워크로드

고객 대면 애플리케이션이나 백엔드 프로세스 같이 비즈니스 가치를 창출하는 리소스 및 코드 모음입니다.

워크스트림

마이그레이션 프로젝트에서 특정 작업 세트를 담당하는 직무 그룹입니다. 각 워크스트림은 독립적이지만 프로젝트의 다른 워크스트림을 지원합니다. 예를 들어, 포트폴리오 워크스트림은 애플리케이션 우선순위 지정, 웨이브 계획, 마이그레이션 메타데이터 수집을 담당합니다. 포트폴리오 워크스트림은 이러한 자산을 마이그레이션 워크스트림에 전달하고, 마이그레이션 워크스트림은 서버와 애플리케이션을 마이그레이션합니다.

웜

한 번 쓰고, 많이 읽으세요.

WQF

AWS 워크로드 검증 프레임워크를 참조하십시오.

한 번 작성하고 여러 번 읽기 (WORM)

데이터를 한 번 쓰고 데이터가 삭제되거나 수정되지 않도록 하는 스토리지 모델입니다. 인증된 사용자는 필요한 만큼 데이터를 여러 번 읽을 수 있지만 변경할 수는 없습니다. 이 데이터 스토리지 인프라는 변경할 수 없는 것으로 간주됩니다.

Z

제로데이 익스플로잇

제로데이 취약점을 악용하는 공격 (일반적으로 멀웨어) 입니다.

제로데이 취약성

프로덕션 시스템의 명백한 결함 또는 취약성입니다. 위협 행위자는 이러한 유형의 취약성을 사용하여 시스템을 공격할 수 있습니다. 개발자는 공격의 결과로 취약성을 인지하는 경우가 많습니다.

좀비 애플리케이션

평균 CPU 및 메모리 사용량이 5% 미만인 애플리케이션입니다. 마이그레이션 프로젝트에서는 이러한 애플리케이션을 사용 중지하는 것이 일반적입니다.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.