Data Engineering

Data engineering is the development, implementation, and maintenance of processes and systems that ingest raw data and produce high quality and consistent data to be used for analysis , machine learning, and more. As a data engineer, you must incorporate security, data management, orchestration, data architecture, software engineering , and operations to manage the lifecycle of data. The goal is to take raw data and make it easy and reliable to work with and to integrate across datasets and domains. To accomplish this goal, you use methods, tools, and services such as streaming, extract, transform, and load, or ETL, data warehouses, and data lakes.

Data Modeling, Data Lineage

Concept

- **Data Modeling** : 논리적 데이터 모델을 구성하는 작업. 산출물은 보통 ERD. : DEA
- **Data Lineage** : a Visual representation that traces the flow and transformation of data through its Lifecycle : DEA
- **Schema Evolution** : Ablity to adapt and change the schema of a dataset over time without disrupting existing process or systems . ex) Glue Schema Registry : DEA

Database Performance Optimization

indexing

- Avoid full table scan

Partitioning

- 샤딩과 같이 큰 데이터셋을 서브셋으로 분리하여 관리
- 매우 큰 테이블을 여러 테이블로 분할
- 기본적으로 스캔하는 데이터의 양을 줄임
- 병렬처리 용이
- 샤딩은 기본적으로 수평 파티셔닝. -> 로우 단위로 움직인다. ? 이거 맞음?

Compression

- Speed up data transfer, reduce storage , disk reads
- GZIP , LZOP , BZIP, ZSTD -> various trade offs between compression and speed
- 열 압축(Columnar compression)

Data Sampling Techniques

Concept

- **Random Sampling** : 모든 샘플이 같은 추출 확률을 가지는 것 : DEA | Statistics
- **Stratified Sampling** : 층화추출. 카테고리별 서브셋 안에서 각각 랜덤추출 : DEA | Statistics
- **Systemic Sampling** : 계통추출. 일정한 순서를 정하고 매번 그 순서에 해당하는 요소를 표본으로 추출 : DEA | Statistics

---

Data Skew Mechanism

Concept

- **Data skew** : 편포된 데이터. unequal distribution or imbalance of data across various nodes or partitions in distributed computing systems. Partition key를 무작위로 생성하게끔 해서 데이터를 균형있게 분산시킬 수 있다. : DEA

편포된 데이터 처리하기

1. Adaptive Partitioning : Dynamically adjust Partitioning based on data characteristics to ensure a more balanced distribution
2. Salting : Introduce a random factor or salt to the data to distribute it more uniformly
3. Repartitioning: Regularly redistribute the data based on its current distribution characteristic
4. Sampling : Use a sample of the data to determine the distribution and adjust the processing strategy accordingly

Steps

- find features of services
- look for aws native services that can handle the features

Concept

- **AWS Graviton** : 고성능 워크로드를 위해 개발된 서버 프로세서 : AWS
- **Materialized View** : 쿼리 결과를 미리 계산해 물리적으로 저장해두는 데이터 베이스 객체. 자주 쓰는 복잡한 쿼리에 적합함.미리 계산된 쿼리의 결과를 캐시해 성능을 크게 향샹시킬 수 있다. : AWS

Key Words

- least operational overhead : try to use the features instead of building your own

DEA Scope of Services

- Analytics
- Application Intergration

- Cloud Financial Management
- Compute
- Containers
- Database
- Developer Tools
- Frontend Web and Mobile
- Machine Learning
- Management and Governance
- Migration and Transfer
- Networking and Content Delivery
- Security, Identity and Compliance
- Storage

Analytics

Amazon Athena

- Query result reuse feature

S3-> Athena

Sales table -> S3 Bucket (S3 prefix path) -> Partitioning and bucketing

Definition: A serverless, interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. Notable characteristics:

- Serverless: No infrastructure to manage

Pay-per-que uery: Only pay for the queries you run

Works directly with data stored in S3

Supports various data formats (CSV, JSON, ORC, Avro, Parquet)

JDBC/ODBC driver support for connecting with BI tools

Concept

- **Amazon Athena** : 서버리스, 인터랙티브 쿼리 지원 분석 서비스. S3에서 직접 데이터를 쿼리할 수 있다. Pay-per-query : DEA ; AWS
- **Athena Workgroup** : Athena 내에서 쿼리를 구성하고 관리해주는 서비스. 업무 관련 쿼리를 그룹화 하여 관리. 그룹별 비용한도 설정. IAM 기반 엑세스 제어. 작업 그룹별 쿼리 성능 및 사용량 모니터링. Athena에서 Spark 엔진을 사용하기 위해 만들어야 한다. : DEA ; AWS
- **Parquet** : 하둡 에코시스템을 위해 설계된 오픈 소스 파일형식. 열 기반저장. 압축효율성, 복합 중첩 데이터 구조 지원이 특징. 보통 Cost-Effective 한 옵션임 : DEA ; AWS
- **Data Buketinge** : Dataset을 Bucket이라는 일종의 범주로 분류하는 것. Bucket간 데이터는 균등하게 분배된다. : DEA ; AWS
- **Query Result Reuse** : Query 결과를 재사용하여 쿼리 성능을 향상시키는 기능 : DEA ; AWS

Spark and Athena

- You must first create a Spark enabled Workgroup in Athena

Query Optimization 전략

- Use an S3 bucket that is in the same AWS Region where the company runs Athena queries
- Preprocess csv data to Apache Parquet format by fetching only data blocks that are needed for predicates

Amazon EMR

AWS Glue

- 서버리스 이므로 설정하거나 관리할 인프라가 없다.
- 원본 데이터의 변경 및 변경 데이터의 저장을 위한 별도의 저장소가 필요없고, 메타데이터 만으로 ETL작업을 수행합니다.
- 정형데이터와 더불어 반정형 데이터도 함께 작동하도록 설계되었습니다.
- ETL 스크립트에서 사용할 수 있는 Dynamic Frame이라는 구성 요소 사용하여 Apache Spark의 Data Frame과 완벽 호환 되고, 스키마가 필요 없고 Dynamic Frame용 고급 변환 세트 이용할 수 있습니다.
- 고성능의 워커로 빠른 작업수행이 가능합니다.
- 스케쥴링 기능으로 주기적인 작업 실행을 자동화할 수 있습니다.
- 북마크 기능으로 작업상태를 저장하여 중단된 시점부터 작업 재개 가능합니다.
- 작업에 대한 모니터링을 지원합니다.

Concept

- **AWS GLue** : 완전관리형 ETL 서비스. 서버리스 서비스 : DEA ; AWS
- **AWS Glue Catalog** : 중앙집중형 메타데이터 저장소. 테이블정의, 작업정의 등 기타 관리정보 포함. 다른 AWS 서비스들과 통합 지원. Persistence(지속성). 기본적으로 모든 데이터 자산에 대한 메타데이터 저장을 지원한다. : DEA ; AWS
- **AWS Glue Crawler** : 메타데이터를 검색하고 카탈로그화하기 위한 자동화된 도구. data schema 변경을 감지 : DEA ; AWS
- **AWS Glue Workflow** : 복잡한 ETL 파이프라인을 위한 ETL 오케스트레이션 툴. Glue Job이 포함된 ETL에 적합. Cost-Effective Solution. 프로세스 관리 기능 : DEA ; AWS
- **AWS Glue Studio** : AWS Glue의 새로운 비주얼 인터페이스.ETL 작업을 손쉽게 실행 및 모니터링 가능 : DEA ; AWS
- **AWS Glue job bookmark** : ETL job Progress 를 추적하고 중단한 지점부터 실행하기 위한 Glue 기능. 기본적으로 job을 추적하기 위한 기능이기 때문에 incremental data processing에 적합 : DEA ; AWS
- **AWS Glue Data Quality** : Glue에서 제공하는 Validation checking 서비스. built-in check 과 custom ruleset 기반 check이 가능하다. : DEA ; AWS
- **AWS Glue Dynamic Frame File Grouping** : 여러 작은 파일을 보다 큰 청크로 묶어서 작업할 수 있는 ETL 기능 : DEA ; AWS
- **Glue ExcutionClass** : AWS Glue job 실행시 설정할 수 있는 파라미터. 사용 가능한 설정값은 standard or flexible. standard 는 빠르게 완료해야 하는 time-sensitive job에 설정. 종료시간이 정해지지 않는 Job에는 flexible 설정 : DEA ; AWS
- **Dynamic Frame** : Glue에서 사용되는 분산 데이터 처리를 위한 데이터 구조. 동적 스키마 및 자체 파티셔닝 기능 : DEA ; AWS

- **Upsert Operation** : 데이터베이스에 새 레코드를 삽입하거나 기존 레코드를 업데이트하는 것을 하나의 작업으로 수행하는 것. 데이터 중복을 막기 위해 스테이징 테이블을 만들고 데이터를 카피한 후 업데이트하는 작업도 Upsert에 해당 : DEA ; AWS

AWS Glue Data Catalog

Use Lamada to create Glue Partitions

- use code that writes data to Amazon S3 to invoke the Boto3 AWS Glue create_partition_API call.
- Use Glue Data Catalog as the central metatdata repository

AWS Glue Crawler

- Use AWS Glue Crawlers to connect to multiple data stores and to Update Data Catalog with metadata changes.

AWS Glue Workflow

AWS Glue DataBrew

Concept

- **AWS Glue DataBrew** : Glue에서 제공하는 Visual Data Transformation Tool. raw 데이터에서 인사이트 추출. 서버리스 솔루션. 데이터 품질진단, 프로파일링 전처리 등 수행 : DEA ; AWS
- **NEST_TO_ARRAY** : Glue DataBrew에서 여러 컬럼을 하나의 배열로 만들 때 사용 : DEA ; AWS
- **NEST_TO_MAP** : Glue DataBrew에서 여러 컬럼을 하나의 Map으로 만들 때 사용 : DEA ; AWS

EventBridge and Glue Workflow

- Use an EventBridge rule to trigger a Glue Workflow when a new file is uploaded to an S3 bucket
- Use EventBridge rule every 15 minutes to trigger a Glue Workflow

AWS Glue DataBrew

- Data Quality Validation이 가능하다.
- PII detection이 가능하다.
- DataBrew gives you the Ability to create a data quality ruleset that automatically performs data quality validations as part of a profiling job.
- You can use DataBrew in combination with a Step Functions state machine to automate data validation in an ingestion pipeline
- 고객 이름이나 성이 포함된 컬럼을 찾아서 unique한 값들을 확인할 수 있다.

AWS Lake Formation

Concept

- **AWS Lake Formation** : 데이터 레이크를 쉽게 설정, 보안설정및 관리하는 서비스. 중앙집중식 권한관리를 통한 세분화된 엑세스 제어 가능. IAM과 통합되어 RBAC 가능. : DEA ; AWS
- **Data Mesh** : 분산된 데이터 소유권을 통해 당면 과제를 해결하는 아키텍처 프레임워크 : AWS

- row level data access and column level data access to specific teams
- Security Policies
- Tag-Based access Control
- Integration with AWS analytics and machine learning services
- Simplifies data lake creation and management
- Centralized permissions management
- Automated data discovery and classification
- Supports governed tables for ACID transactions

리소스 접근제한 걸기. PII 시나리오

- Enabling fine-grained access control in AWS Lake Formation
- Register S3 path as an AWS Lake Formation location

Analytics_Stream

Amazon Kinesis Data Firehose

Concept

- **Kinesis Data Firehose** : 스트림 데이터를 변환하여 S3, Redshift, Open Search등 분석 서비스에 전달하는 ETL 서비스 완전관리형 서버리스, 준실시간(near real time). 데이터 포맷 변환 지원. 데이터 전송에 중점 : DEA ; AWS

Producer, Source, Writing Kinesis Data Streams Kinesis Agent Kinesis Data Firehose API CloudWatch Logs & Events Destination S3 Redshift Elasticsearch Splunk HTTP 엔드포인트

Amazon Kinesis Data Streams

Concept

- **Kinesis Data Streams** : 대규모 레코드 스트림을 실시간으로 수집하고 처리하는 AWS 서비스.보통 로그 데이터, 앱로그, 시장 데이터 피드, 웹 클릭스트림 데이터 .실시간 처리 및 분석에 중점 : DEA ; AWS
- **Data Record(KDS)** : Kinesis Stream에 저장되는 단위 : DEA ; AWS
- **Shard(KDS)** : Data Stream에서 고유하게 식별되는 레코드 시퀀스-> 스트림은 하나 이상의 샤드로 구성되며 샤드는 고정된 용량 제공. Stream 용량 제한이 샤드에 의해 결정되기 때문에 샤드의 수를 늘림으로서 늘어난 데이터 쓰루풋을 감당할 수 있다. : DEA ; AWS

- **Partition Key(KDS)** : 스트림 내의 Shard 별로 데이터를 그룹화 하기 위한 Key : DEA ; AWS
- **Sequence Number(Data Streams)** : 각 데이터 레코드에는 샤드 내에 파티션-키 마다 고유한 sequence number가 존재 : DEA ; AWS
- **WriteThroughputExeeded exceptions** : AWS 서비스, 특히 DynamoDB에서 프로비저닝된 처리량(읽기 또는 쓰기 용량)을 초과했을 때 발생하는 오류 : DEA ; AWS
- **Batch Messages(KDS)** : Record를 Batch로 처리하는 것. overhead를 줄이고 처리량을 늘리기 위해 KDS에서 사용할 수 있는 옵션. ProvisionedThroughputExeededException 방지 : DEA ; AWS

Amazon Kinesis Agent:

A standalone Java application that collects and sends data to Amazon Kinesis Data Streams Typically installed on servers generating log data Monitors specified files and streams new data to Kinesis Handles file rotation, checkpointing, and retry upon failures Simplifies the process of getting data into Kinesis streams

Kinesis Client Library (KCL):

A Java library that helps consume and process data from Kinesis Data Streams Manages the interaction between your application and Kinesis Handles shard assignment, checkpointing, and load balancing Allows developers to focus on processing logic rather than stream management Available in multiple programming languages

Kinesis Data Streams and DynamoDB

- User Kinies Data Streams to capture and store data in DynamoDB ex)sensor data -> Kinesis Data Streams -> DynamoDB
- query from datastore with latancy of less than 10ms

Amazon Managed Service for Apache Flink

Concept

- **Amazon Managed Service for Apache Flink** : Apache Flink를 사용해 실시간 스트리밍 어플리케이션을 빌드하고 실행할 수 있는 완전관리형 서버리스 서비스 . 과거 Kinesis Data Analytics란 이름이었다. 데이터 변환, 대화형 쿼리, 실시간 분석 및 다른 AWS와의 통합이 특징 : DEA ; AWS

Amazon Managed Streaming for Apache Kafka(Amazon MSK)

Concept

- **Amazon Managed Service for Apache Kafka** : Apache Kafka 인프라와 운영을 관리하는 AWS 스트리밍 데이터 서비스. Kafka와 같이 데이터 Publisher와 Consumer를 분리하는 역할을 한다. : DEA ; AWS
- **Broker Node** : 카프카 클러스터 구성요소중 하나로 메시지를 저장하고 관리하는 역할을 한다. Producer로부터 메시지를 수신하고 Consumer에게 전달함. 토픽과 파티션을 관리한다. : DEA ; AWS
- **Zookeepr Node** : 분산시스템 조정을 위한 노드. 브로커의 상태관리, 클러스터 설정 저장 : DEA ; AWS

- **Producer(Kafka)** : 메시지를 생성하고 브로커에 전송하는 어플리케이션. 필요시 메시지 직렬화 수행 : DEA ; AWS
- **Consumer(Kafka)** : 브로커로부터 메시지를 읽는 어플리케이션 : DEA ; AWS
- **Topic** : Producer가 게시한 메시지가 저장되는 공간. Kafka에서 메시지를 구분하는 논리적인 단위. Kafka는 기본적으로 하나의 Topic을 여러 브로커로 파티셔닝하는 전략을 취한다. : DEA ; AWS

- 기본적으로 kafka와 동일하기 때문에 파티션 수의 증가는 가능하지만 감소는 불가능하다.
- 파티션 수를 증가시키면 데이터의 분산이 증가하고 처리량이 증가한다.
- Kinesis Data Streams의 경우 Shard의 수를 증가시키고 감소시킬 수 있다.

Amazon OpenSearch Service

Concept

- **OpenSearch** : 시계열 데이터와 준실시간 데이터 분석에 적합한 대시보드 서비스. 낮은 지연시간이 필요할 경우 적합 : DEA ; AWS
- **OpenSearch Dashboards** : AWS 에서 제공하는 Kibana 서비스 : DEA ; AWS

Amazon QuickSight

Concept

- **AWS QuickSight** : 대시보드 생성 및 리포팅을 위한 서버리스 툴. 쿼리가 필요 : DEA ; AWS
- **AWS QuickSight SPICE** : Super-fast, Parallel, In-memory Calculation Engine. QuickSight에서 빠른 쿼리 성능 및 시각화 렌더링 제공. 컬럼기반 저장 및 오토스케일링이 특징. ad-hoc 분석을 위해 설계됨 : DEA ; AWS

S3 -> Athena -> QuickSight

기본적으로 대시보드 툴

Permission Issue

권한 문제시 체크

- QuickSight does not have permission to access the data in S3
- QuickSight does not have permission to decrypt the data in S3

Application Integration

Amazon AppFlow

Concept

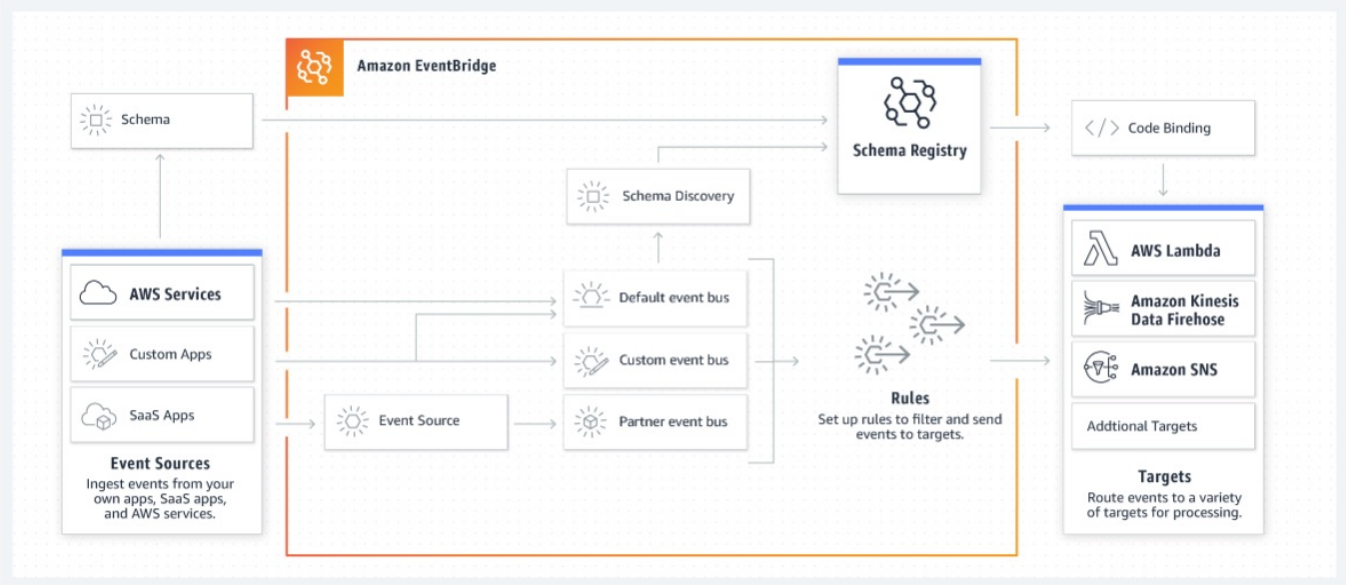
- **Amazon Appflow** : 주로 SaaS 어플리케이션과 AWS 서비스간 안전한 데이터 전송을 지원하는 완전관리형 통합 서비스. 주로 여러 서드파티 어플리케이션과 S3, Redshift등 AWS 서비스와의 연결을 지원 : DEA ; AWS

Amazon EventBridge

Concept

- **Amazon EventBridge** : 어플리케이션 간에 이벤트를 라우팅 하는 서버리스 이벤트 버스 서비스. 강력하고 유연한 이벤트 필터링 및 패턴 매칭 기능 제공. 서드파티 Saas와의 연결을 지원함. decoupling application에 활용. : DEA ; AWS

Amazon EventBridge is recommended when you want to build an application that reacts to events from SaaS applications and/or AWS services. Amazon EventBridge is the only event-based service that integrates directly with third-party SaaS partners. Amazon EventBridge also automatically ingests events from over 90 AWS services without requiring developers to create any resources in their accounts. Further, Amazon EventBridge uses a defined JSON-based structure for events and allows you to create rules that are applied across the entire event body to select events to forward to a target. Amazon EventBridge currently supports over 15 AWS services as targets, including AWS Lambda, Amazon SQS, Amazon SNS, Amazon Kinesis Streams, and Firehose, among others. At launch, Amazon EventBridge has limited throughput (see Service Limits) which can be increased upon request, and typical latency of around half a second.



Amazon Managed Workflows for Apache Airflow

Concept

- **Apache Airflow** : 데이터 오케스트레이션 툴. 자동화된 스케줄링 구축에 용이 : DEA ; AWS

Amazon SNS

Concept

- **Amazon SNS** : Publish-Subscribe 메시징 서비스. 기본적으로 하나의 메시지를 여러 구독자에게 동시에 브로드캐스팅 하는데 사용(fan-out). 메시지 지속성 없음. 실시간 이벤트 알림에 적합. : DEA ; AWS

Amazon SQS

Concept

- **AWS SQS** : Simple Queue Service. 대기열 기반 메시징 서비스. 마이크로서비스, 분산시스템간 통신을 위해 사용. 작업 대기열 관리 및 부하분산에 적합.결제, 주문처리 등 안정적인 메시지 처리가 필요한 경우 보다 적합함. 메시지를 최대 14일간 보존: DEA ; AWS
- **FIFO queue** : First-In-First-Out Queue. 메시지 순서를 보장하는 대기열. 메시지 그룹화 및 순서 보장이 필요한 경우 사용. : DEA ; AWS
- **dead-letter queue** : DLQ. 일정 횟수 이상 실패한 메시지를 보관하는 격리된 대기열. 재처리 및 디버깅에 사용 : DEA ; AWS

SNS -> SQS -> EC2 패턴

이벤트 발생 -> SNS Topic -> SQS -> EC2

AWS Step Functions

Concept

- **AWS Step Functions** : 서버리스 워크플로우 오케스트레이션 툴. Visual Workflow Designer. Error Handling and Retry Mechanism: DEA ; AWS

- Step Functions uses a state machine, which is a workflow defined using the Amazon States Language (ASL), a JSON-based structured language. This state machine defines a series of steps, with the output of one step acting as input to the next step.
- The workflow in Step Functions is represented as a series of states. Each state represents a unit of work that can perform some action, make a choice, or pass control to another state. The state machine handles error checking, retry logic, and can parallelize tasks, making it easier to build and manage complex workflows.

Step Functions and IAM

- Make Sure that Step Functions state machine code has all IAM permissions that are necessary to create and run the EMR jobs

Step Functions and EMR

- Query the flow logs for the VPC. Determine whether the traffic that originates from EMR cluster is reach the data providers.

Cloud Financial Management

AWS Budgets

AWS Cost and Usage Report

AWS Cost Explorer

Compute

AWS Batch

Amazon EC2

Application Auto Scaling

- to schedule the scaling of EC2 instances

AWS Lambda

Concept

- **Lambda** : 서버를 프로비저닝 하거나 관리할 필요 없이 서버를 실행할 수 있게 해주는 서버리스 컴퓨팅 서비스. 오토 스케일링, 이벤트 기반 실행 가능 : DEA ; AWS
- **Lambda Layer** : 라이브러리, 커스텀 런타임 또는 기타 함수 종속성을 포함할 수 있는 ZIP 아카이브. 코드 재사용 및 배포 최적화, 코드 버전관리용으로 사용 : DEA ; AWS

Lambda Layer

update the lambda functions

less manual way to update the lambda functions

=> package the custom Python scripts into Lambda Layers. Apply the Lambda Layers to the Lambda functions

Lambda@Edge

AWS Serverkess Application Model(SAM)

Containers

Amazon Elastic Container Registry(ECR)

Amazon Elastic Container Service(ECS)

Concept

- **AWS Fargate** : 쿠버네티스와 ECS를 위한 서버리스 컴퓨팅 엔진. processing lazyer에서 scaling이 필요한 경우 적합 : DEA ; AWS

AWS Fargate Fargate는 Amazon EC2 인스턴스의 서버나 클러스터를 관리할 필요 없이 컨테이너를 실행하기 위해 Amazon ECS에 사용할 수 있는 기술입니다. AWS Fargate를 사용하면 더 이상 컨테이너를 실행하기 위해 가상 머신의 클러스터를 프로비저닝, 구성 또는 조정할 필요가 없습니다. 따라서 서버 유형을 선택하거나, 클러스터를 조정할 시점을 결정하거나, 클러스터 패킹을 최적화할 필요가 없습니다.

Amazon Elastic Kubernetes Service(EKS)

Database

Amazon Aurora

Concept

- **Database Activity Streams** : Aurora에서 제공하는 데이터베이스 동작을 모니터링 하기 위한 기능. 기본적으로 Kinesis Data Streams를 사용한다. 클러스터 수준에서 동작한다. : AWS

Amazon Timestream DB

Concept

- **Amazon Timestream DB** : 빠르고 확장성이 좋은 서버리스 시계열 DB. 테라바이트 단위의 시계열 데이터를 높은 가용성으로 몇초 안에 쿼리할 수 있다. : AWS

Amazon DocumentDB

Amazon DynamoDB

Concept

- **DynamoDB** : 완전관리형 NoSQL DB 서비스. 규모에 상관없이 한자리수 밀리초의 지연시간을 제공하게끔 설계. 서버리스, 오토스케일링 : DEA ; AWS
- **DynamoDB Streams** : DynamoDB 테이블의 수정 변경사항을 캡처하는 서비스. Lambda와 함께 사용하여 중요한 변경사항에 대해 이벤트를 실행하게끔 할 수 있다. : DEA ; AWS
- **DynamoDB Dax** : DynamoDB 성능 향상을 위한 인메모리 캐싱기술 : DEA ; AWS
- **PartiQL** : SQL 호환 쿼리 언어로, 관계형, 비관계형, 중첩 데이터를 쿼리할 수 있게 해주는 Amazon의 오픈 소스 쿼리 언어. 대규모 분산시스템에서도 효율적으로 작동 : DEA ; AWS

- Fast Querying
- Caching

Amazon Keyspaces(for Apache Cassandra)

Amazon MemoryDB for Redis

Amazone Neptune

Amazon RDS

Amazon Redshift

Concept

- **AWS Redshift** : 대규모 데이터 웨어하우징을 위한 완전 관리형 페타바이트 규모의 클라우드 DW 서비스. 대규모 병렬처리(MPP) 아키텍처 사용.보안을 위한 암호화 지원 : DEA ; AWS
- **Redshift Spectrum** : Redshift의 확장 기능으로 S3에 저장된 데이터를 직접 쿼리할 수 있게끔 함. 대규모 데이터 세트 분석. 유연한 스키마 설계 가능 : DEA ; AWS
- **Redshift Serverless** : 인프라 관리 없이 DW클러스터를 구축할 수 있는 서버리스 서비스. PPU(Pay-Per-Use).자동 용량 관리 및 스케일링. : DEA ; AWS
- **Redshift Federated Query** : Redshift에서 외부 데이터 소스의 데이터를 직접 쿼리할 수 있게끔 하는 기능. 데이터를 Redshift로 이동하지 않고도 다양한 소스의 데이터를 분석할 수 있다.여러 소스의 데이터를 단일 가상 테이블로 통합 : DEA ; AWS
- **Redshift Query Editor** : Redshift 에서 제공하는 쿼리 편집기. DB, 스키마 테이블 및 프로시저 생성 가능. 예약된 실행 기능으로 프로시저 실행을 스케줄링 할 수 있음. : DEA ; AWS
- **Distribution key** : Redshift에서 데이터를 클러스터의 노드에 분산시키는 기준이 되는 열. 같은 distribution key를 가진 행들은 같은 노드에 저장. 조인 최적화 및 워크로드 밸런싱 : DEA ; AWS
- **VACCUM** : 공간을 확보하고 행을 재정렬하는 절차.특정 테이블 및 전체 테이블에 대해 실행가능. 삭제 플래그 표시된 행을 삭제하고 정렬키 기준으로 재정렬. 기본적으로 자동으로 수행되지만 수동으로 직접 실행할 수도 있다. : DEA ; AWS
- **VACCUM FULL** : 테이블 공간 확보 및 모든 행 재정렬 명령어. 테이블에 대해 실행 : DEA ; AWS
- **VACCUM SORT ONLY** : 테이블 정렬키 기준으로 재정렬 : DEA ; AWS
- **VACCUM DELETE ONLY** : 테이블 공간 확보만 진행 : DEA ; AWS
- **VACCUM REINDEX** : VACCUM FULL 에 더해 인덱스 재생성까지 진행. 인덱스 깨졌을 경우 진행 : DEA ; AWS
- **Distribution Style** : Redshift에서 분산 스타일은 데이터가 컴퓨팅 노드에 어떻게 분산되는지를 의미 : DEA ; AWS
- **ALL Distribution** : Full Copy of the Table is stored on each distribution. 전체 테이블이 모든 컴퓨팅 노드에 분산됨. 자주 업데이트 되지 않는 테이블에 적합. 공통으로 join되어 사용되는 표준공통디멘션에 적합. 작은 테이블에 적합 : DEA ; AWS
- **Key Distribution** : 특정 열의 값을 기준으로 행이 분산됨. 연관 데이터를 같은 슬라이스에 있도록 하기때문에 조인 성능 최적화. join이 자주 일어나는 fact와 dimension 테이블에 적합 : DEA ; AWS
- **Even Distribution** : 행들이 라운드 로빈 방식으로 슬라이스에 분산됨. 명확한 분산 키가 없는 테이블에 적합 : DEA ; AWS
- **AWS DataExchange** : 클라우드에서 서드파티 데이터 세트에 대한 구독 검색, 사용을 위한 서비스 : AWS
- **AWS Redshift Data API** : Lambda, SageMaker, Cloud9 등의 다른 AWS 서비스에서 Redshift에 접근할 수 있게끔 하는 인터페이스. 실시간 쿼리가 가능하다. : AWS
- **Redshift Data Sharing** : 데이터를 수동으로 이동하거나 복사하지 않고도 Redshift 클러스터, 작업 그룹, 계정 및 리전 전체에서 라이브 데이터에 대한 엑세스를 안전하게 공유 가능. 기존 분석 작업을 방해하지 않고 라이

브 데이터에 엑세스 : AWS
- **Ad-hoc query** : 미리 정의되거나 예약된 것이 아닌 필요에 따라 즉석에서 생성되고 실행되는 DB쿼리를 의미함. 주로 표준 보고서가 다루지 않는 방식으로 데이터를 탐색하기 위해 설계됨 : AWS
- **Concurrency Scaling mode** : RW 성능을 향상시키기 위한 Redshift의 동시성 스케일링 모드 : AWS

Redshift and S3

Redshift and Lambda

Redshift Streaming Ingestion

Use Streaming Ingestion to ingest data from Amazon Kinesis Data Streams into Amazon Redshift Serverless.

Redshift Cluster

EVEN distribution

The leader node distributes the rows across the slices in a round-robin fashion, regardless of the values in any particular column. EVEN distribution is appropriate when a table doesn't participate in joins. It's also appropriate when there isn't a clear choice between KEY distribution and ALL distribution.

KEY distribution

The rows are distributed according to the values in one column. The leader node places matching values on the same node slice. If you distribute a pair of tables on the joining keys, the leader node collocates the rows on the slices according to the values in the joining columns. This way, matching values from the common columns are physically stored together.

ALL distribution

A copy of the entire table is distributed to every node. Where EVEN distribution or KEY distribution place only a portion of a table's rows on each node, ALL distribution ensures that every row is collocated for every join that the table participates in.

ALL distribution multiplies the storage required by the number of nodes in the cluster, and so it takes much longer to load, update, or insert data into multiple tables. ALL distribution is appropriate only for relatively slow moving tables; that is, tables that are not updated frequently or extensively. Because the cost of redistributing small tables during a query is low, there isn't a significant benefit to define small dimension tables as DISTSTYLE ALL.

Redshift Spectrum

Amazon Redshift Spectrum is a feature of Amazon Redshift that enables you to run SQL queries directly against exabytes of unstructured data in Amazon S3 without having to load or transform the data.

Key Features

1. Query Data in S3 Directly

- Run queries on data stored in S3 without loading it into Redshift tables.
- Support for various file formats (Parquet, ORC, JSON, CSV, Avro, etc.).

2. Separation of Storage and Compute

   - Store data cost-effectively in S3 and scale compute in Redshift independently.
   - Pay only for the queries you run.

3. Massive Scalability

   - Query exabytes of data in S3.
   - Thousands of Redshift Spectrum nodes can work in parallel.

4. Integration with Redshift

   - Seamlessly query both local Redshift data and external S3 data in the same SQL query.
   - Use the same BI and analytics tools you use with Redshift.

5. Performance Optimization

   - Intelligent query optimization pushes down predicates to S3.
   - Partition pruning to minimize data scanned.

6. Data Lake Queries

   - Query data in your S3-based data lake without moving the data.
   - Combine with Redshift data for data lake and data warehouse integration.

7. Schema Flexibility

   - Supports schema evolution.
   - Add or remove columns in S3 without affecting existing queries.

8. Security Integration

   - Integrates with AWS IAM for access control.
   - Supports encryption of data at rest in S3.

9. Cost-Effective

   - Pay per query based on the amount of data scanned.
   - No need to pay for storage of data queried in S3.

How It Works

1. Define external tables in Redshift that point to data in S3.
2. Use standard SQL to query these external tables.
3. Redshift Spectrum parallelizes queries across thousands of nodes.
4. Results are returned to the Redshift cluster and then to the client.

Use Cases

1. Historical Data Analysis: Query historical data stored in S3 without loading into Redshift.

2. Data Lake Analytics: Run analytics on your entire S3 data lake.
3. Infrequently Accessed Data: Query cold data in S3 without keeping it in Redshift.
4. ETL Data Processing: Transform data in S3 before loading into Redshift.

Best Practices

1. Use columnar formats like Parquet for better performance.
2. Partition data in S3 for improved query efficiency.
3. Compress data to reduce scan times and costs.
4. Use appropriate data types in external table definitions.
5. Monitor query performance and adjust S3 data organization as needed.

Limitations

1. Limited write capabilities (can't perform INSERT, UPDATE, DELETE on external tables).
2. Some Redshift features are not available for external tables.
3. Query performance can be slower compared to querying data in Redshift tables.

Understanding Redshift Spectrum is crucial for designing cost-effective and scalable data analytics solutions on AWS, especially when dealing with large amounts of data in S3 data lakes.

Developer Tools

AWS CLI

AWS Cloud9

AWS CDK

AWS CodeBuild

AWS CodeCommit

AWS CodeDeploy

AWS CodePipeline

Front-End Web & Mobile

Amazon API Gateway

Machine Learning

Amazon SageMaker

Amazon Comprenhend

- NLP solution
- to detect sentiment, entities, and key phrases in text.

Concept

- **Amazon SageMaker** : 머신 러닝 모델 구축, 훈련, 배포를 위한 완전관리형 서비스 : DEA ; AWS
- **SageMaker Data Wrangler** : Data Sampling Using AWS SageMaker : DEA ; AWS

SageMaker Notebook : DynamoDB에서 직접 boto3를 사용하여 데이터를 쿼리할 수 있으며, SageMaker에서 데이터를 쿼리할 수 있습니다.

Management and Governace

AWS CloudFormation

템플릿을 사용한 AWS 리소스의 자동화된 프로비저닝 및 관리 인프라 변경 사항의 버전 관리 및 추적 다양한 AWS 서비스 및 리소스의 통합 관리

Concept

- **CloudFormation** : IaC 서비스. 템플릿을 사용한 AWS 리소스의 자동화된 프로비저닝 및 관리. 인프라 변경 사항의 버전 관리 및 추적. : DEA ; AWS

AWS CloudTrail

Concept

- **CloudTrail** : AWS 계정의 거버넌스, 규정준수, 감사를 위한 서비스. API 호출 및 계정활동 로그 처리. AWS 계정 활동의 상세한 이벤트 기록 제공 : DEA ; AWS

CloudTrail can send logs to CloudTrail Lake without the need to develop a custom solution. CloudTrail Lake automatically converts the JSON event type to Apache ORC format, and stores the data in an event data store. CloudTrail Lake gives you the ability to run SQL queries across multiple event data stores automatically. You can use this solution to analyze the event data automatically.

- S3로 로그를 저장하고, Athena를 통해 쿼리할 수 있다.

CloudTrail Lake

- Automatic Data intergration from CloudTrail
- AWS Config configuration items
- AWS Audit Manager evidence

Amazon CloudWatch

- 어플리케이션 모니터링을 위한 커스텀 지표 생성가능

Concept

- **CloudWatch** : AWS 어플리케이션 및 관리 서비스. 실시간 로그 및 지표 수집. 알림설정 및 자동화된 작업 트리거 . 대시보드를 통한 성능 시각화. : DEA ; AWS

Amazon CloudWatch Logs

You can use Amazon CloudWatch Logs to monitor, store, and access your log files from Amazon Elastic Compute Cloud (Amazon EC2) instances, AWS CloudTrail, Route 53, and other sources.

CloudWatch Logs enables you to centralize the logs from all of your systems, applications, and AWS services that you use, in a single, highly scalable service. You can then easily view them, search them for specific error codes or patterns, filter them based on specific fields, or archive them securely for future analysis. CloudWatch Logs enables you to see all of your logs, regardless of their source, as a single and consistent flow of events ordered by time.

CloudWatch Logs also supports querying your logs with a powerful query language, auditing and masking sensitive data in logs, and generating metrics from logs using filters or an embedded log format.

CloudWatch Logs supports two log classes. Log groups in the CloudWatch Logs Standard log class support all CloudWatch Logs features. Log groups in the CloudWatch Logs Infrequent Access log class incur lower ingestion charges and support a subset of the Standard class capabilities. For more information, see Log classes.

CLoudWatch Logs insights

CloudWatch Log Insights automatically discovers fields in logs from AWS services, such as Amazon Route 53, Lambda, CloudTrail, and Amazon VPC, and any application or custom log that emits log events as JSON.

- Amazon Route 53

Amazon Athena CloudWatch connector

- LogGroups as schemas
- LogStreams as tables
- all_log_streams view (all LogStreams in LogGroup)

AWS Config

When you turn on AWS Config, it first discovers the supported AWS resources that exist in your account and generates a configuration item for each resource. AWS Config keeps track of all changes to your resources by invoking the describe or the list API call for each resource in your account. The service uses those same API calls to capture configuration details for all related resources.

Amazon Managed Grafana

AWS Systems Manager

AWS Well-Architected Tool

Migration and Transfer

AWS Application Discovery Service

AWS Application Migration Service

AWS DMS

Concept

- **AWS Database Migration Service** : 데이터베이스를 AWS 클라우드로 안전하게 마이그레이션하거나 데이터베이스 간에 지속적으로 데이터를 복제할 수 있게 해주는 완전 관리형 서비스. 이기종 DB, CDC, KMS 적용 : AWS
- **Reflatform Migration Strategy** : lift and reshape 마이그레이션 전략. 아키텍쳐와 코드를 거의 손대지 않고 새로운 플랫폼으로 옮기는 것 : AWS
- **Refactor Migration Strategy** : re-architecting. 신규 환경에 맞게끔 어플리케이션을 재구축 하는것 : AWS

AWS DataSync

Concept

- **AWS DataSync** : a managed data transfer service that simplifies and accelerates moving large amounts of data online between on-premises storage and Amazon S3, EFS, or FSx for Windows File Server. DataSync is optimized for efficient, incremental, and reliable transfers of large datasets, making it suitable for transferring 5 TB of data with daily updates. : AWS

AWS Schema Conversion Tool (SCT)

AWS Snow Family

AWS Transfer Family

- Transfer Family can update security policy of the Transfer Family server endpoint tp specify a minimum protocol version of TLS 1.2

SSL/TLS는 Secure Sockets Layer와 Transport Layer Security(전송 계층 보안)를 의미하는 용어로, 컴퓨터 시스템이 인터넷에서 안전하게 서로 통신할 수 있도록 하는 프로토콜 또는 통신 규칙

Networking and Content Delivery:

Amazon CloudFront

AWS PrivateLink

Amazon Route 53

Amazon VPC

Concept

- **VPC gateway endpoint** : 다른 AWS 서비스와 VPC를 연결해주는 서비스 inbound route 과 outbound route이 연결되어 있어야 한다. : AWS
- **VPC inferface endpoint** : on premise 환경과 vpc 망을 연결할 경우 사용 : AWS
- **Data in Transit** : 특정 지점에서 다른 지점으로 이동하는 데이터를 지칭. SSL/TLS 같은 프로토콜, VPN으로의 전송, AWS PrivateLink와 관련 : AWS
- **Data at Rest** : 클라우드 스토리지나 하드 드라이브에 저장되어 이동하지 않는 데이터를 지칭. SSE, CSE, KMS를 통한 키 관리와 관련되어 있다. : AWS

Security, Identity, and Compliance:

Projecting Data

- Transport Layer Security(TLS) : DEA ; AWS

AWS Identity and Access Management (IAM)

Concept

- **AWS IAM** : AWS 리소스에 대한 엑세스를 안전하게 제어하는 웹 서비스. 전반적인 엑세스 관리 시스템 제공. 멀티 팩터 인증 지원 : DEA ; AWS
- **AWS STS** : Security Token Services. AWS 리소스에 대한 임시 보안 자격 증명 생성 및 제공 서비스. : DEA ; AWS
- **AssumeRole** : AWS STS의 API 작업 중 하나로, 지정된 역할을 수임하여 임시 보안 자격 증명을 얻는 프로세스. 자격 증명에 대한 유효기간 설정 가능 : DEA ; AWS
- **Amazon Resource Name** : ARN. AWS 리소스를 고유하게 식별할 수 있는 일종의 식별자.IAM 정책, Amazon Relational Database Service(RDS) 태그 및 API 호출과 같은 모든 AWS에서 리소스를 명료하게 지정해야 하는 경우 사용 : DEA; AWS
- **Security Groups** : 네트워크 보안 및 접근제어를 위한 서비스. 인스턴스 수준(4계층)에서 방화벽처럼 동작한다. Stateful 서비스 : DEA; AWS
- **Network ACL** : 네트워크 보안 및 접근제어를 위한 서비스. 서브넷 수준에서 동작. Stateless 서비스 : DEA; AWS

IAM Access Analyzer

IAM Identity Center

- Workforce identities
- Application assignments
- Identity center

- Multi-account permission

Identities in AWS

- IAM user
- IAM group
- IAM role

AWS Certiticate Manager

IAM policies

- Trust Policy
- Identity-based policy
- Resource-based policies

AWS managed policy : Creatd and managed by AWS Customer managed policy : Created and managed by You Inline policy : Created for a single IAM Identity

Access Models

Concept

- **RBAC** : Access based on user Role. Business logic : DEA ; AWS
- **ABAC** : Attribute Based Access Control. Access based on attributes or tags : DEA ; AWS
- **Client-Side Encryption** : Encrypt data before sending it to AWS : DEA ; AWS
- **Server-Side Encryption** : Encrypt data after sending it to AWS : DEA ; AWS
- **Encryption** : In-Transit and At-Rest : DEA ; AWS
- **Tokenization** : Plain Text into a string of Characters. Replace sensitive data with non-sensitive data : DEA ; AWS
- **Token Vault** : Store and manage tokens : DEA ; AWS
- **Salt** : Random data that is used as an additional input to a one-way function that hashes data

---

AWS Key Management Service (AWS KMS)

Concept

- **AWS KMS** : 데이터 암호화에 사용되는 암호화 키를 쉽게 생성, 제어, 관리할 수 있게 해주는 관리형 서비스. 중앙집중식 암호화 키 관리. 자동 또는 수동으로 키를 주기적으로 교체 가능.IAM과 통합된 세부적 권한관리.CloudHSM 클러스터와 통합하여 자체 키 스토어 생성 가능 : DEA ; AWS

- to give find-grained control over who can use the keys

AWS Key Management Service (KMS) AWS Key Management Service (KMS) is a managed service that makes it easy to create and control the encryption keys used to encrypt your data. Key Features

Centralized Key Management: Create, import, rotate, disable, delete, define usage policies for, and audit the use of encryption keys. Integrated with AWS Services: Seamlessly integrates with many AWS services for data encryption. Secure: Uses Hardware Security Modules (HSMs) that are validated under FIPS 140-2. Auditable: Integrated with AWS CloudTrail to provide logs of all key usage. Highly Available: Built on systems that are designed for 99.999999999% durability.

Types of Keys

Customer Master Keys (CMKs):

Can be used to encrypt up to 4KB of data directly More commonly used to encrypt data keys (envelope encryption)

Data Keys:

Used to encrypt large amounts of data Can be generated using a CMK

Key Concepts

Envelope Encryption: Process of encrypting data with a data key, then encrypting the data key with a master key. Key Rotation: Automatic yearly rotation of CMKs to enhance security. Key Policies: JSON documents that control access to a CMK. Grants: Alternative to key policies for providing temporary permissions to use CMKs.

Use Cases

Encrypt data at rest in AWS services (S3, RDS, etc.) Encrypt sensitive application data Manage keys for client-side encryption Securely share data across AWS accounts

Best Practices

Use separate CMKs for different applications or types of data Regularly audit key usage and permissions Use the principle of least privilege when granting access to keys Enable automatic key rotation for CMKs Use aliases to refer to keys instead of key IDs

AWS KMS is a managed service to create and control the cryptographic keys that are used to protect your data. AWS KMS integrates with most other AWS services that encrypt your data. For example, AWS KMS integrates with CloudTrail to log use of your AWS KMS keys for auditing, regulatory, and compliance, and you can use the AWS KMI API to create and manage AWS KMS keys and features such as custom key stores and use AWS KMS keys in cryptographic operations. Here's a question. What AWS service can you use if you have a requirement to directly manage the AWS CloudHSM device that generates, stores, and uses encryption keys?

KMS -> S3

- User Server-Side Encryption with KMS(SS3-KMS) to encrypt data that contains PII. Configure IAM policies to allow access to the KMS key.

Encrption fundamentals

Lambda -> DynamoDB -> KMS

- DynamoDB Client-Side Encryption : DEA ; AWS

One design would be a serverless application that uses API Gateway, Lambda, Amazon Cognito, DynamoDB, and AWS KMS. For this design, the client authenticates with Amazon Cognito and receives an authorization token. The token is used to validate calls to the customer order Lambda function. The function calls the tokenization layer providing sensitive information in the request. This layer includes the logic to generate unique random tokens and store encrypted text in a cipher database. Lambda calls AWS KMS to obtain an encryption key. It then uses the DynamoDB client-side encryption library to encrypt the original text and store the ciphertext in the cipher database. The Lambda function retrieves the generated token in the response from the tokenization layer. This token is then stored in the application database for future reference. AWS KMS manages the creation and management of cryptographic keys. It provides logs of all key usage to help you meet regulatory and compliance needs.

Project Sensitive Data in DW

Concept

- **Dynamic Data Masking** : 동적 데이터 마스킹. DBMS에서 민감한 데이터에 대한 실시간 엑세스를 제어하는 보조기능. 실시간 처리 및 원본데이터 보존. 사용자 인가수준별 차등적용: DEA ; AWS

Here's another question. How do you protect sensitive data in your data warehouse? You can use Dynamic Data Masking, or DDM, in Amazon Redshift and manipulate how Amazon Redshift shows sensitive data to the user at query time without transforming it in the database. When attached to a table, the masking expression is applied to one or more of its columns. You can further modify masking policies to only apply them to certain users or users defined roles that you create with RBAC. You can also apply DDM on the cell level by using conditional columns when creating your masking policy, and you can apply multiple masking policies with varying levels of obfuscation to the same column in a table and assign them to different roles.

AWS CloudHSM

CloudHSM provides hardware security models in AWS. AWS automates the hardware provisioning, software patching, network routing, and creating encrypted backups of key stores. You are responsible for scaling your CloudHSM environment and managing the crypto accounts and credentials within the HSM. Like AWS KMS, CloudHSM is designed so that plain text keys cannot be used outside the HSM by anyone.

Amazon Macie

Concept

- **Amazon Macie** : AWS에서 제공하는 완전관리형 데이터 보안 및 프라이버시 서비스. 기계학습과 패턴 매칭을 활용해 민감 데이터 감지 및 분류. PII 식별에 적합 : DEA ; AWS

AWS Secrets Manager

> AWS Secrets Manager를 사용하면 수명 주기 동안 데이터베이스 보안 인증, 애플리케이션 보안 인증, OAuth 토큰, API 키 및 기타 암호를 관리, 검색, 교체할 수 있습니다. 다수의 AWS 서비스는 Secrets Manager에 보

> 안 암호를 저장하고 사용합니다.

Concept

- **AWS Secrets Manager** : Credential data를 보관하는 저장소. credentials을 일정 주기마다 교체하는built in 기능이 있다. 수명주기 동안 데이터베이스 보안 인증, 어플리케이션 보안 인증, OAuth토큰, API 키 및 기타 암호를 관리, 검색, 교체 가능. 다수의 AWS 서비스는 Secrets Manager에 보안 암호를 저장하고 사용 : DEA ; AWS

A secrets lifecycle has four phases, create, store, use, and destroy. And a secrets management solution protects the secrets in each of these phases from unauthorized access. AWS Secrets Manager is a secrets management service to help you protect access to your application services and resources.

Secrets Manager offers built-in integration with IAM and you can attach access control policies to IAM principals or to secrets themselves by using resource-based policies. Secrets Manager also integrates with AWS Key Management Service, or AWS KMS. Secrets are encrypted at rest by using an AWS managed key or customer managed key. And Secrets Manager supports a rotation of secrets securely, and you can schedule automatic database credential rotation for Amazon RDS, Amazon Redshift, and Amazon DocumentDB. You can also use customized Lambda functions to extend the Secrets Manager rotation feature to other secrets types such as API keys and OAuth tokens.

Parameter Store

AWS Shield

AWS WAF

Storage

AWS Backup

AWS Backup Audit Manager

AWS Backup Audit Manager lets you audit and report on the compliance of your data protection policies to help meet business and regulatory compliance requirements. To monitor the backup activity of the S3 resources you identified earlier, you can create a custom framework in AWS Backup Audit Manager with a targeted set of controls configured.

Amazon EBS

- Root Volume 에서만 DeleteOnTermination이 True로 설정되어있다.

When an instance terminates, the value of the DeleteOnTermination attribute for each attached EBS volume determines whether to preserve or delete the volume. By default, the DeleteOnTermination attribute is set to True for the root volume. It is set to False for all other volume types.

Concept

- **Amazon EBS** : 컴퓨팅 인스턴스에 부착하는 네트워크 드라이브의 일종 : DEA ; AWS
- **EBS root volume** : persistent data by default : DEA ; AWS
- **Provisioned IOPS SSD** : IO intensive DB에 적합한 고성능 SSD : DEA ; AWS
- **General Perpose SSD** : 범용 SSD : DEA ; AWS
- **Cold HDD** : large, colde dataset에 적합한 드라이브 : DEA ; AWS
- **throughput Optimized SSD** : large dataset의 throughput intensive workload의 적합. Mapreducing, Kafka, DW 등 : DEA ; AWS

Amazon EFS

Amazon S3

Concept

- **Amazon S3** : 다양한 유형의 객체를 관리하고 저장하는 웹기반 스토리지 서비스.높은 내구성과 가용성. 사용 빈도에 따른 티어링이 특징 : DEA ; AWS
- **Amazon S3 Select** : S3애 저장된 객체에서 직접 쿼리를 실행할 수 있는 S3 기능.단일객체에 적합. Join은 지원하지 않는다. : DEA ; AWS
- **Amazon S3 Storage lens** : AWS에서 제공하는 스토리지 분석 서비스. 대화형 대시보드를 통해 스토리지를 이해, 분석 및 최적화하여 조직 전체, 특정 계정, 리전, 버킷 또는 접두사에 대한 데이터를 집계 : DEA ; AWS
- **Amazon S3 notification feature** : S3 버킷에 대한 이벤트 알림 서비스 SNS, SQS Lambda로 이벤트를 보낼 수 있다. SQS는 FIFO SQS가 아닌 Standard SQS만 가능 : DEA ; AWS
- **S3 standard** : 자주 엑세스 하는 데이터를 위한 기본 스토리지 클래스. 낮은 레이턴시와 높은 쓰루풋 : DEA ; AWS
- **S3 Standard IA** : Infrequent Access. 수명이 길고 엑세스 빈도가 낮은 데이터를 위한 클래스 : DEA ; AW
- **S3 One Zone-Infrequent Acess** : Standard IA와 유사하지만 데이터가 단일 가용 영역에 저장됨 : DEA ; AWS
- **S3 Glacier Instant retreval** : 즉시 엑세스가 필요한 아카이브 데이터를 위한 스토리지 클래스. 최소 90일의 저장기간. : DEA ; AWS
- **S3 Glacier Deep Archive** : 최소 180일의 저장기간을 가지고 최저 비용의 아카이브 데이터를 위한 스토리지 클래스. : DEA ; AWS
- **S3 Intelligent Tiering** : 자동으로 데이터를 가장 적합한 스토리지 클래스로 이동시키는 스토리지 클래스. : DEA ; AWS
- **SSE-S3** : AWS가 키를 완전히 관리하며 추가비용 없이 사용할 수 있다. : DEA ; AWS
- **SSE-C** : Server-Side Encryption with Customer Provided keys : DEA ; AWS
- **SSE-KMS** : SSE-S3와 유사하지만 Key에 대한 감사추적을 제공한다. : DEA ; AWS

S3 Select

S3 는 간단한 쿼리를 지원한다. S3 Select를 사용하면 S3에 저장된 객체에서 쿼리를 실행할 수 있다. 이를 통해 객체의 일부만 읽을 수 있으며, 객체의 크기가 클 경우 데이터 전송 비용을 줄일 수 있다. Join은 지원하지 않는다.

S3 접근 관리

특정 국가, 지역에서만 접근가능하게 하기

LakeFormation을 사용하여 S3 버킷에 대한 접근을 제한할 수 있다.

data lake location in AWS Lake Formation. You can use Lake Formation to restrict access to the S3 bucket to specific countries or regions. You can also use Lake Formation to restrict access to specific IP addresses or IP address ranges. This way, you can ensure that only users in specific countries or regions can access the data lake.

S3 Storage Classes

IAM

ACL(Access Control List)

Bucket Policy

Storage lens

S3 Storage Lens delivers organization-wide visibility into object storage usage, activity trends, and makes actionable recommendations to optimize costs and apply data protection best practices.

Amazon Glacier

Unsorted

Stream Processing Style

Simple event processing

각각의 이벤트가 직접적으로 수행해야할 action과 매핑되어 처리 된다.실시간으로 작업의 흐름을 처리할 때 사용되며, 이벤트 처리 시간과 비용의 손실이 적다. Event Stream Processing

이벤트를 중요도에 따라 필터링하여 걸러진 이벤트만을 수신자에게 전송.실시간으로 정보의 흐름을 처리할 때 사용되며, 기업에 적용될 경우 신속한 의사 결정을 가능하게 한다.(BAM) Complex event processing

일상적인 이벤트의 패턴을 감지하여 더 복잡한 이벤트의 발생을 추론하는 것. 예를 들어 ‘주식의 등락’이라는 일상적인 이벤트의 패턴을 감지하여 ‘투자 적기’ 라는 상위의 이벤트를 추론해 낼 수 있다.