

4-24-22
12:36:18

PLAY >



REC

AWS入門②

知っておくと強い用語集



データベース (DB)

構造的に整理されているデータの集合体。データが整理されていて、データを検索したり、特定のデータだけを取り出したたりするなど、プログラムからデータを扱いやすくなる

データベースの種類

リレーションナル型(RDB)

- Amazon RDS

- RDB (Relational DataBase / リレーションナルデータベース)

データを「表（テーブル）」として管理し、行と列で構造化して扱うデータベースの方式。

テーブル同士は「キー」で関連付けられ、複雑な検索や結合が可能。トランザクション処理や整合性維持に強い。

- RDS (Relational Database Service) **は、

AWSが提供する「RDBを簡単に使えるようにしたクラウドサービス」

- RDBMS (Relational Database Management System / リレーションナルデータベース管理システム)

- DBMS (Database Management System / データベース管理システム)

非リレーションナル型(NoSQL)

- Amazon DynamoDB(キーバリュー型)
- Amazon DocumentDB(ドキュメント型)
- Amazon ElastiCache(インメモリ型)
- Amazon Neptune(グラフ型)
- Amazon Timestream(時系列型)

- NoSQL (Not Only SQL)

テーブル型ではなく、柔軟な形式でデータを保存・管理するデータベースの総称。

JSON形式やキーと値のペア、カラム指向、グラフ構造など、用途に応じた多様な構造を持つ。

スキーマ（列の定義）が固定されない場合が多く、柔軟にデータを追加できる。

それぞれは、リレーションナル型ではないという共通点しかない

- キーバリュー型 (Key-Value Store)

NoSQLの一種で、「キー」と「値」をセットで保存するデータベースです。

辞書や連想配列のイメージで、キーを指定すると対応する値を高速に取得できます。

- ドキュメント型 (Document Store)

NoSQLの一種で、JSONやXMLなどの「ドキュメント」単位でデータを保存するデータベースです。

1つのドキュメント内に複数のデータを自由に入れられ、構造も柔軟に変えられます。

- インメモリ型 (In-Memory Database)

データをディスクではなくメモリ (RAM) 上に置いて管理するデータベースです。

そのため、読み書きが非常に高速で、リアルタイム処理に向いています。

DBMSとRDBMSの違いをシンプルにまとめるとこうなります。		
項目	DBMS	RDBMS
管理するデータ	構造化されているとは限らない	行と列で整理されたリレーションナルデータ
データ操作言語	プログラミック/簡単なAPIが多い	SQL (標準化された言語)
データの整合性	必ずしも保証されない	トランザクション管理 (ACID) で整合性を保証
テーブル間の関係	なし、または限定的	外部キーなどでテーブル同士の関係を定義できる
例	ファイルベースDB、初期のDBソフト	MySQL、PostgreSQL、Oracle、SQL Server

ざっくり言うと：

- DBMS = 「データを管理するシステム全般」
- RDBMS = 「きちんと整理された表（リレーションナル型）を管理するシステム」

つまり、RDBMSはDBMSの一種で、RDBに特化したものです。

**リレーションナル型 (RDB) と非リレーションナル型 (NoSQL) **の違い		
項目	リレーションナル型 (RDB)	非リレーションナル型 (NoSQL)
構造	行と列で構成されたテーブル	JSON・キーと値・グラフなど多様
スキーマ	固定（事前に列を定義）	スキーマレス（柔軟に追加変更可）
操作言語	SQL	DBごとに異なるAPIやクエリ
強み	データ整合性・複雑な検索に強い	高速アクセス・大量データ処理・スケール容易
例 (AWS)	Amazon RDS, Aurora	DynamoDB, DocumentDB, Neptune
利用例	会員管理、在庫管理、決済処理	セッション管理、リアルタイム分析、IoTデータ

ざっくり言えば、

- RDB = 「きちんと整理された表計算シート」
- NoSQL = 「自由に書き足せるメモ帳やグラフノート」

優劣があるわけではなく、必要に応じて使い分けるもの

現在のシステムの主流は「Webシステム」という形式

WebサーバーやHTMLなどWebの仕組みを利用してシステムを構築すること。AWSで構築するのはWebシステムが多い

よくある組み合わせ

- Webサーバー上にプログラム、表示するHTML、画像をおく
- データはデータベースサーバーに保存

Webサーバーとデータベースサーバーを組み合わせることが多い



LAMPサーバー

Webアプリを動かすための代表的なサーバー構成の1つで、Linux + Apache + MySQL + PHP の頭文字を取った名前です。

構成要素

- Linux : OS
- Apache : Webサーバー（ブラウザからのリクエストを処理）
- MySQL : データベース（DB）
- PHP : プログラム言語（Webアプリの処理を実行）

AWSならEC2やECS／EKSと組み合わせることが多い

Amazon EC2 (Elastic Compute Cloud)

AWSが提供する**仮想サーバー（クラウド上のコンピュータ）**のサービスです。
物理サーバーを用意する必要がなく、クリックだけで必要な性能・台数のサーバーを作れます。

Amazon ECS (Elastic Container Service)

AWSが提供するコンテナ管理サービスで、Dockerコンテナを簡単にデプロイ・管理・スケーリングできるようにしたものです。

Amazon EKS (Elastic Kubernetes Service)

AWSが提供するKubernetes（コンテナオーケストレーション）を簡単に運用できるマネージドサービスです。

データベースの役割は、データを効率的かつ安全に管理・格納し、必要に応じて取り出せるようにすること

ネットワーク

サーバーが建物だとすれば、
ネットワークはそこにつながる道

AWSのEC2やRDSなどのサービスは、ネットワークに接続する
ことが必須
そのためのAWSのネットワークがAmazon VPC

Amazon VPC (Virtual Private Cloud)

AWS上に自分専用の仮想ネットワークを作れるサービスです。

インターネットや他のAWSリソースから隔離されたネットワーク空間を用意でき、ネットワーク構成を
自由に設計できます。

DNS

インターネットの住所録のような役割

Amazon Route 53

AWSが提供するDNS (Domain Name System) サービスです。

ドメイン名とIPアドレスを結びつけ、インターネット上でWebサイトやサービスにアクセスできるよう
にします。

特徴

- ドメイン名登録とDNS管理が可能
- 高可用性・スケーラブルな設計
- ヘルスチェック機能で、異常なサーバーを自動で切り離す
- トラフィックルーティング（地理的分散、重み付け、フェイルオーバー）に対応

使う場面

- Webサイトやアプリのドメイン管理
- 複数リージョンにまたがるサービスのトラフィック分散
- フェイルオーバー構成での高可用性確保

ざっくり言うと、

「ドメイン名とサーバーのIPをつなぐ、AWSの電話帳兼交通整理係」です。

情報セキュリティ

情報のCIA

「機密性(Confidentiality)」「完全性(Integrity)」「可用性(Availability)」

機密性(Confidentiality)

正当な権利を持つ人だけがアクセスできること
=権利のない人は見られない

完全性(Integrity)

データが正当であること
=改竄されていない、壊れていない

可用性(Availability)

必要な時にアクセスできること
=きちんと使える

「サービスが正常な状態で提供できなくなる」事態は全て
セキュリティを脅かすもの=インシデント

コンテナ

プログラムの実行環境を隔離する仕組み

代表的な技術

- Docker

特徴

- OS全体を仮想化するVMより軽量
- どの環境でも同じように動作（開発環境＝本番環境）
- 起動・停止が高速
- 複数のコンテナを1台のサーバー上で並行実行可能

Amazon ECS (Elastic Container Service)

AWSが提供するコンテナ管理サービスで、Dockerコンテナを簡単にデプロイ・管理・スケーリングできるようにしたものです。

Amazon EKS (Elastic Kubernetes Service)

AWSが提供するKubernetes（コンテナオーケストレーション）を簡単に運用できるマネージドサービスです。

サーバーレス

利用者がサーバーを意識することなくアプリケーション（プログラムを動かせる仕組み）

AWS Lambda

AWSが提供するサーバーレスでコードを実行できるサービスです。

サーバー自分で用意せず、コードをアップロードするだけで、自動的に実行環境を用意してくれます。

AI分野

Amazon Bedrock

AWSが提供する生成AI（Generative AI）をAPI経由で利用できるサービスです。複数のAIモデル（テキスト生成、画像生成、要約、チャットなど）を、サーバー構築やモデル学習なしで使えます。

特徴

- 複数のAIモデルプロバイダー（Anthropic、Meta、Mistral、Amazon Titanなど）を選んで利用可能
- 自分のデータを使ったカスタマイズ（ファインチューニング、RAG）に対応
- サーバー構築やモデル学習は不要、API呼び出しだけで利用可能
- AWSのセキュリティ・ネットワークと統合されて安全に運用できる

使う場面

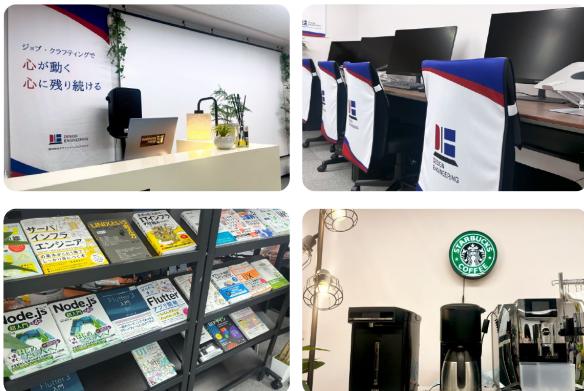
- チャットボットや文章生成
- コード補完やバグ修正提案
- 検索結果の要約
- 画像生成やコンテンツ作成支援

ざっくり言うと、

「AWS版の生成AIプラットフォーム。いろんなAIモデルを安全にまとめて使える入口」です。



LINE公式アカウントにて
最新情報配信中！



無料で自由に使える
学習＆カフェスペース開放中！

変化を楽しみ、自分らしく未来へ。

デザインエンジニアリングは、
挑戦するエンジニアの一歩を応援する会社です。

“好き”や“ワクワク”をそのままキャリアに変え、
自分の可能性を信じて前へ進む人には、無限のチャンスが広がっています。

失敗も学びに変え、仲間と共に笑い、共に成長しながら、
毎日が少しずつ楽しくなる未来へ。
未経験でも大丈夫。あなたの最初の一歩を、心からお待ちしています！



イベント・セミナー開催中！

カジュアル面談・エントリーは
こちらから！



LINE ID:@749gaovb

Q デザインエンジニアリング

URL : <https://design-engineering.jp/>