



Deloitte.
デロイトトーマツ



**Deloitte on AWS で加速する
データベースモダナイゼーションジャーニー**

目次

はじめに	02
はじめに	02
リレーショナルデータベース	07
非リレーショナルまたは NoSQL データベース	08
高度なトランザクション処理伴う金融サービスアプリケーション	12
IoT ベースの分析アプリケーション	13
リアルタイムマルチプレイヤーゲーム	14
「サーバーレス」はどのようにしてさらなる進化を遂げているのか？	15
Amazon Aurora Serverless	16
Amazon Neptune Serverless	17
デロイトのケーススタディ 1	19
デロイトのケーススタディ 2	20
まとめ	22
執筆者	23
改訂履歴	24



はじめに

はじめに

あらゆる企業や組織において保存や処理の必要なデータの進化、データ量、およびデータの種類の驚異的な速度で増加しています。また、データモデルの多様性も増えています。すべてのデータベースニーズを適切に管理できる統合データベースを保持する需要は少なくなっています。クラウドにおけるモダンアーキテクチャでは、適材適所のツールを必要とし、目的に応じたデータベース（DB）の活用が求められています。

本ホワイトペーパーでは、アマゾンウェブサービス（AWS）が提供する目的に応じた DB をどのようにさまざまなアプリケーションワークロード、データの形状と構造、パフォーマンス要件、および運用効率のために活用できるかを説明していきます。加えて、DB の歴史とアプリケーションおよび DB がどのように進化してきたかについても触れていきます。

AWS による目的に応じた DB の提供で実現可能なモダンアーキテクチャの例も紹介しています。最後に、デロイトのクライアントにおける導入事例を紹介いたします。本ホワイトペーパーは、AWS の目的に応じた DB サービスを使用してアプリケーションとデータワークロードをモダナイズするための戦略を提供することを目標としております。





多くの組織がアプリケーションの履歴を利活用しやすい形でモダライズしようとしています。デジタル時代においてデータは日常生活の一部となっており、スマートフォン、IoT（Internet of Things）デバイス、デジタル家電が大量のデータを生成しています。これに加えて、従来のシステムも大量のデータを生成しています。今日のデジタル組織は、膨大な量のデータを取り込み、保存し、処理して有用な情報を提供し、成長と拡大を促進するインサイトを得る必要があります。DBはこのデータの大部分を保存し、運用する上で重要な役割を果たします。

組織は、パフォーマンスやスケーリングに制限のないアジャイル特性をもつアプリケーションを構築したいと考えています。また、DB サービスが自己修復機能を持ち、日々のDB 管理を軽減し、ビジネスに集中できることが期待されます。

さらに、1 秒未満の遅延、場合によっては 1,000 分の 1 秒未満の遅延を必要とするアプリケーションも存在します。

この大規模な拡張を支える DB 技術も、広範なデータ処理と保存のニーズを満たし、現代のビジネス課題に特化した DB を構築するために進化しています。

データについて検討する際、データはさまざまな形式、構造、作業量が考えられます。これらすべての構造をリレーショナルデータベース（RDB）でモデリングすることは、不十分かつ非効率です。例えば、RDB に適さないデータパターンには、キー・バリュー、ドキュメント、ワイドカラム、グラフ、タイムシリーズなどがあります。アプリケーションをモダライズし、成功させていくためには、データパターンと時間とともに進化するアーキテクチャデザインを理解することが重要です。

データベースにおけるパラダイムシフト

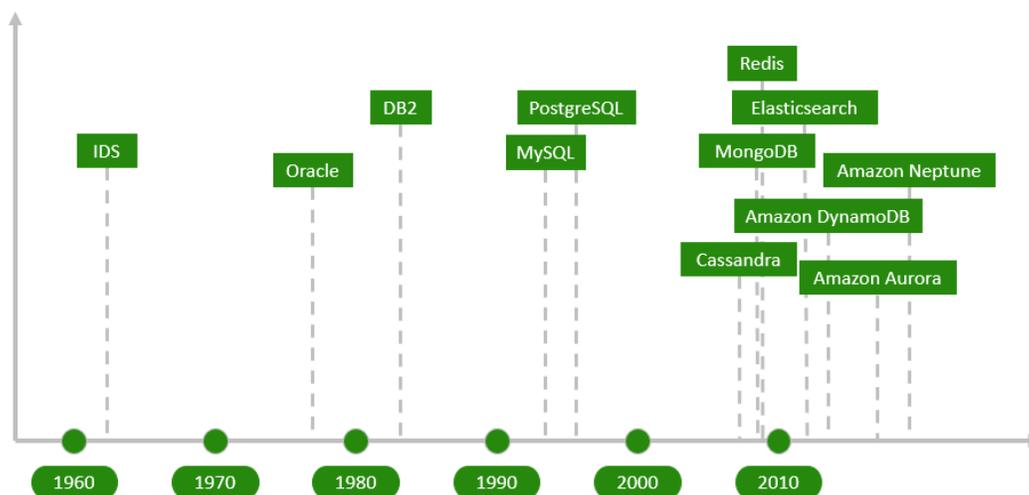


図 1: 1960 年代以降の DB のタイムライン¹

過去数十年にわたり、DB の様相は、保存できるデータの種類（多様性）、量（ボリューム）、そしてデータを処理・保存する頻度（速度）の点で進化してきました。インターネットを利用する 3 層アーキテクチャを採用したアプリケーションの導入により、DB に求められる要件が変わりました。その結果、開発者はより高速でグローバルに対応できる DB を求めるようになっています。さらに、クラウドコンピューティングの台頭により、技術的に可能なことが変わり、より堅牢でスケーラブルなアプリケーションを経済的に実現できるようになりました。アプリケーションアーキテクチャの進化をたどると、メインフレームはアプリケーションと DB の両方を集中的に管理して提供しました。

クライアントサーバー時代には、アプリケーションが DB から分離されました。

そして、1990 年代には、多くの組織が 3 層アーキテクチャ（Web、アプリケーション、データ）を採用しました。現在では、多くの組織が DB の作業量を小さく分割する際に、マイクロサービスやイベントドリブンアーキテクチャを活用しています。様々な機能を持つアプリケーションや DB を目的に応じて分離することはアジリティを高め、市場投入までの時間を短縮するための鍵となります。過去 10 年間でデータの発展は指数関数的に増加しており、その勢いは衰える兆しを見せていません。図 2 に示すように、単一の RDB で組織のすべての要件を管理できる時代は終わりました。

¹<https://s2.smu.edu/~fmoore/timeline.pdf>

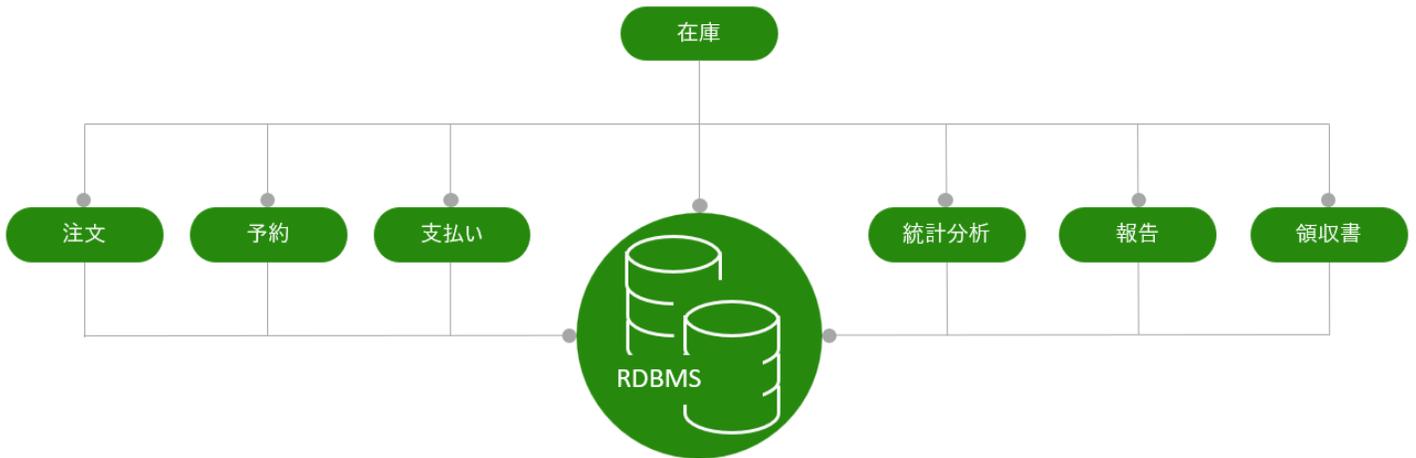


図 2: リレーショナルデータベース管理システム（RDBMS）があらゆる種類のデータを保存するためのソリューションであった時代

「単体ですべての用途に適合する」DB は、モダンなアプリケーションおよび分析エコシステムのニーズに対応するのが難しくなっています。目的に応じた DB の選択が求められる主な課題には以下の使用例が含まれます：

- データの指数関数的な増加、最適化された保存、および取得の必要性
- 自動スケーリング
- 自動バックアップ
- 複数種類のデータ（キー・バリューペア、複数キー・バリューペア、タイムシリーズ）の効果的な処理
- 入れ子構造のデータ（JSON（JavaScript Object Notation））の効果的な処理
- アプリケーションのさまざまな遅延要件に対応

これらの課題を解決するために、AWS は特定のユースケースに対応した 15 種類の DB サービスを提供するようになりました。さらに、それら目的別のフルマネージド AWS DB を使用することで、顧客は時間とコストを節約し、スケール時のパフォーマンスを向上させ、迅速にイノベーションを進めることができます。

開発者はデフォルトの RDB に頼る必要はありません。アプリケーションのニーズを考慮し、要件に最も適した DB を選択することができます。

DB の選択は、アプリケーションアーキテクチャの要素として重要です。

選択する DB の種類は、チームが責任を持つ管理アクセスパターン、パフォーマンス、および運用に影響を与えます。アプリケーションワークロード、データの形状、パフォーマンス要件など、さまざまな要素を考慮する必要があります。

異なるワークロードとデータベースの整合性

AWS は 15 の目的に応じた DB を提供しており、適材適所のツールを選択することができます。これらのフルマネージド DB サービスは、継続的な監視、自己修復ストレージ、および自動スケーリングを提供し、アプリケーションのビジネス推進に集中できるよう支援します。本セクションの目的は、AWS が提供する各 DB サービスの主な機能、特徴、および一般的な使用例を紹介することです。

図 3 は、AWS の目的に応じた DB の種類の幅広さを示しています：

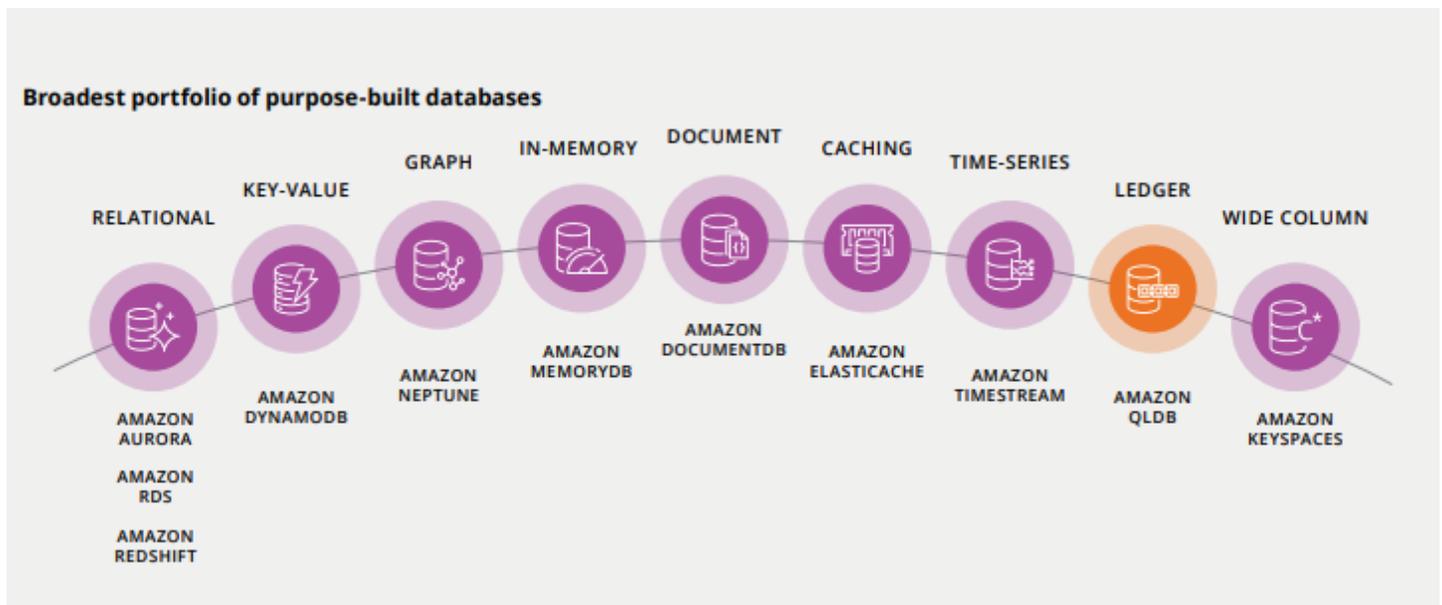


図 3: AWS の目的に応じた DB のポートフォリオ²

² <https://aws.amazon.com/products/databases/>

リレーショナルデータベース

AWS は 2 種類の RDB を提供しています：

1) Amazon Relational Database Service (Amazon RDS)

5 つの DB エンジン（商用およびオープンソースの両方）をマネージド環境で実行できる RDB サービスです。

2) Amazon Aurora：

AWS クラウド向けに設計された高性能な RDB で、MySQL または PostgreSQL と互換性があり、商用エンジンと比較して大幅に低コストでグローバルにスケールできます。

Amazon RDS

Amazon RDS は以下の 6 つのエンジンをサポートしています：

- MySQL
- MariaDB
- PostgreSQL
- Oracle
- Microsoft SQL Server
- DB2

これは、既存のアプリケーション、DevOps プロセス、およびツールがすでに定義されている一般的な DB エンジンを利用できることを意味します。

特徴:

- クラウド移行の容易化
- マルチ AZ（アベイラビリティゾーン）デプロイメント
- 自動バックアップとパッチ適用
- パフォーマンス制御の方針（汎用 vs プロビジョニング IOPS)

また、基盤となるハードウェアをカスタマイズする必要があるユーザー向けに Amazon RDS Custom もあります。これは、クラウドと互換性がない可能性のあるレガシーまたはパッケージアプリケーションに原則有用です。

使用例:

- エンタープライズリソースプランニング（ERP）
- Customer Relationship Management/顧客管理（CRM）
- 金融データ
- トランザクション

Amazon Aurora

Amazon Aurora は、クラウド向けに構築された RDBMS で、MySQL および PostgreSQL と完全な互換性があります。Aurora は、商用グレードの DB のパフォーマンスと可用性を 10 分の 1 のコストで提供します。顧客が現在使用しているコード、ツール、およびアプリケーションは、既存の MySQL および PostgreSQLDB と同様に Amazon Aurora でも使用できます。特定のワークロードでは、Amazon Aurora は既存のアプリケーションを変更せずに MySQL の 5 倍、PostgreSQL の 3 倍のスループットを提供できます。

特徴:

- 自己修復ストレージ
- ストレージの自動スケーリング
- 最大 15 のリードレプリカをサポート
- 自動、継続的、段階的なバックアップとポイントインタイムリカバリ
- グローバル DB
- 優れたパフォーマンスと耐久性
- サーバーレス

使用例:

- モバイルおよびウェブゲーム
- Software as a Service（SaaS）アプリケーション
- エンタープライズアプリケーション

非リレーショナルまたは NoSQL データベース

数十年間、唯一の DB の選択肢は RDB でした。そのため、アプリケーション内のデータの形状や機能に関わらず、RDB モデルを使用することが唯一の選択肢でした。

では、RDB はスキーマの正規化や DB 内の参照整合性を担保するものだったのでしょうか？確かに RDB はこうした目的も達成できていました。しかし、ここでの重要なポイントは、すべてのアプリケーションのデータモデルや使用場面がリレーショナルモデルに一致するわけではないということです。つまり、リレーショナルデータモデルはすべての使用例に適合するわけではありません。そこで、AWS が提供する NoSQL DB を紹介します。

Amazon DynamoDB

Amazon DynamoDB は、スケーリングの規模に関係なく高速で一貫したパフォーマンスを提供するフルマネージド NoSQL DB です。柔軟な課金モデル、Infrastructure as Code (IaC) との緊密な統合、およびハンズオフの運用モデルを備えています。また、Amazon DynamoDB は誤差 1,000 分の 1 秒のパフォーマンスを提供します。

顧客は通常、他の従来の DB システムでスケラビリティの問題を経験した場合に

Amazon DynamoDB を検討します。また、Amazon DynamoDB は高性能な読み取りと書き込みを処理し、変化する負荷の下でも予測可能なパフォーマンスを提供する場合に理想的です。

特徴:

- 自動スケーリング
- DB 行の TTL (Time To Live)
- オンデマンドバックアップとリストア
- 大規模スケールでの誤差 1,000 分の 1 秒のパフォーマンス

使用例:

- マルチプレイヤーゲームのデータ保存
- メタデータストアとキャッシュ
- ショッピングカート
- ユーザートラッキング

Amazon DocumentDB

ドキュメント DB は、アプリケーション内でリッチなドキュメントを保存およびクエリすることを可能にする NoSQL DB の一種です。

Amazon DocumentDB は MongoDB と互換性があり、組み込みのセキュリティベストプラクティス、継続的なバックアップ、および他の AWS サービスとのネイティブ統合を提供します。Amazon DocumentDB はまた、ほぼ瞬時のクラッシュリカバリを提供します。ログ構造ストレージを利用しており、トランザクションログの再生が不要であるため、再起動時間が通常 30 秒未満に短縮されます。

特徴:

- クラウド上のフルマネージドドキュメント DB
- MongoDB との互換性
- 高いスケラビリティとパフォーマンス
- 自動、継続的、増分バックアップとポイントインタイムリカバリ

使用例:

- コンテンツ管理システム
- ユーザープロフィール
- 商品カタログ

Amazon ElastiCache

一部のサービスやアプリケーションでは、主要なデータストレージとして耐久性があり、強力な DB が必要です。例えば、Amazon Aurora のような RDB や Amazon DocumentDB (MongoDB 互換) などの NoSQL DB を指します。これらの DB は強力なクエリ機能と強固なデータ保証を提供し、主要な DB として非常に適しています。

しかし、時にはアプリケーションユーザーが主要なデータストアから得られる遅延に対し、より低遅延であることが求められる場合があります。例えば、アプリケーション内の共通リクエストフローにおいて、読み取り処理が書き込み処理より著しく多い場合です。このような状況では、インメモリキャッシュを使用して支援することが望ましいかもしれません。頻繁に読み取られるクエリの結果をキャッシュに保存することで、応答時間を改善し、主要な DB への負荷を軽減できます。

Amazon ElastiCache は、2 つのオープンソースインメモリキャッシュエンジン (Redis と Memcached) をサポートしています。

Amazon ElastiCache を使用すると、フルマネージドのインメモリキャッシュを利用できます。インスタンスのフェイルオーバー、バックアップとリストア、ソフトウェアのアップグレードの責任を負う必要はありません。

使用例:

- セッションの保存
- リアルタイム分析
- チャットアプリ
- 商品カタログ

Amazon Neptune

Amazon Neptune は、AWS が提供するフルマネージドのグラフ DB サービスであり、エンティティとその関係を表すノードとエッジで構成されるグラフデータの保存とクエリに特化しています。また、Amazon VPC を使用したネットワーク分離、データの静止および転送時の暗号化、アイデンティティおよびアクセス管理 (IAM) ポリシーなどの強力なセキュリティ機能を提供します。Amazon Neptune は、グラフベースのアプリケーションを構築および実行するための強力なツールです。

使用例:

- レコメンドエンジン
- 不正検出
- ソーシャルネットワーク分析
- 新薬開発
- サプライチェーン管理

Amazon Timestream

時系列データを扱い、RDB で時系列データをモデル化しようとする、次のような共通の課題が常に存在します:

- 時系列データには適さない
- 時系列データの処理が非効率
- 時系列データの性質に合わない硬直したスキーマ

既存の時系列 DB における弊害には次のものがあります:

- スケーリングが難しい
- 可用性の維持が難しい
- データライフサイクル管理の制限

Amazon Timestream はこれらの課題を解決します。Amazon Timestream は、高速でスケール可能なサーバーレスの時系列 DB サービスで、1 日あたり数兆のイベントを保存および分析することを容易にします。

使用例:

- 重要な時系列アプリケーションの監視とアクションの実行
- テレメトリデータの保存と分析
- IoT データの保存と分析

Amazon Quantum Ledger Database (Amazon QLDB)

Amazon QLDB は、不変で暗号的に検証可能なトランザクションログを提供する、透明性の高いフルマネージド型の台帳 DB サービスです。Amazon QLDB は、一貫性のあるトランザクションによる透明性の高いジャーナルを使用してアプリケーションデータの変更履歴を追跡および維持するのに役立ちます。Amazon QLDB の ACID トランザクションを使用すると、正確なイベント駆動型システムを構築できます。

使用例:

- 銀行および金融業務で不変なトランザクションを記録する
- 人事・給与管理において完全かつ集中化された従業員の詳細の記録を作成する
- サプライチェーンシステムで材料の履歴を追跡および維持する

Amazon Keyspaces

Amazon Keyspaces は、スケーラブルで高可用性の Apache Cassandra 互換のマネージド DB サービスで、大規模アプリケーションに必要な高速な読み取りおよび書き込み性能を提供します。オープンソース技術を基に構築され、データをワイドコラム形式で保存します。

Apache Cassandra のデプロイ、設定、および管理には専門的な知識が必要です。バージョンアップグレードの煩雑さ、暗号化管理の複雑さ、およびリソースの最適化の課題は、通常、Apache Cassandra ベースのワークロードのデプロイを敬遠させます。Amazon Keyspaces はこうした複雑さを軽減し、顧客がビジネス問題の解決に集中できるようにします。

使用例:

- メッセージング
- デバイスメタデータ
- イベントログ

Amazon MemoryDB for Redis

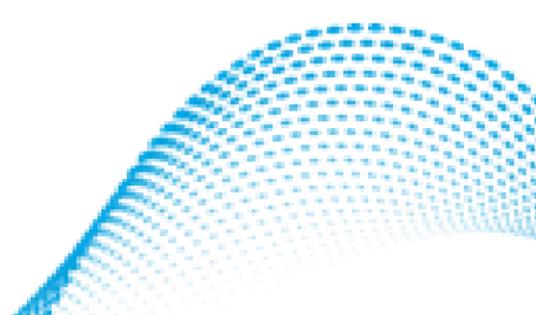
Amazon MemoryDB for Redis は、Redis 互換のインメモリ DB サービスで、非常に高速なパフォーマンスを提供します。キャッシュとは異なり、Amazon MemoryDB に保存されたデータは耐久性があります。Amazon MemoryDB は、誤差 1,000 分の 1 秒の遅延すら許容できないアプリケーション向けに設計されています。

Amazon MemoryDB はフルマネージドであり、即座に高性能かつ耐久性のある MemoryDB クラスタを稼働することができます。データの耐久性、一貫性、および回復性を提供するための分散トランザクションログを提供します。

また、Amazon MemoryDB はデータセット全体をメモリに保存するため、1,000,000 分の 1 秒単位の読み取り遅延と誤差 1,000 分の 1 秒の書き込み遅延を実現できます。1 日に 13 兆以上のリクエストを管理し、1 秒あたり数億のリクエストのピークをサポートできます。

使用例:

- オンラインゲーム
- ストリーミングメディアとエンターテインメント
- チャットおよび高性能なウェブアプリケーション



モダンアーキテクチャの種類

このセクションでは、AWS の目的に応じた DB を活用することで可能なアーキテクチャパターンの例を紹介します。



トランザクション性の高い 金融サービスアプリケーション

金融サービス業界のアプリケーション群が、オンプレミス環境の単一の大規模な OracleDB インスタンスに依存しているとイメージしてください。これらのアプリケーションのユースケースの例としては、金融取引、トレーディング、会計、不正検出などがあります。これには非常に複雑な DB クエリ、金融取引のための ACID（原子性、一貫性、分離性、耐久性）プロパティの要件、および高ボリュームの読み取りと書き込みの必要性が含まれます。また、キャッシングレイヤーを利用する 100 以上のテーブルがあります。このタイプの DB を AWS に移行するためのオプションは 3 つあります：

01. 現状のまま Amazon Elastic Compute Cloud (EC2) インスタンスに移行する。
02. Amazon RDS – Oracle に再プラットフォームする。
03. 大規模なモノリシックを複数の特定用途向け DB に分割する。

方法 1 と 2 は、ワークロードをクラウドに移行する際に組織が選択することが多いですが、時には不十分です。オンプレミス環境の既存の問題や課題をアプリケーションと一緒にクラウドへ移行してしまいます。潜在的には、さまざまな DB のユースケースをより詳しく見て、それらを目的に応じた DB に分割することで、アプリケーションのニーズをより効果的に解決することができます。方法 3 を詳しく見てみましょう。提案されるアーキテクチャは以下の図のようになります。



図 4: 金融サービスアプリケーションのための顧客の DB パターン

Aurora は、IOPS（1 秒あたりの入出力操作）をスケールする方法が非常に重要であるため、金融サービスのアプリケーションにおいては Amazon RDS の DB エンジンよりも適しています。

金融取引において不正を識別するためには、取引そのもののトランザクションは役に立ちません。取引の背後にいる関係者と、これらの関係者が他の関係者とどう繋がっているか、そして取引自体がどのように関連しているかを知る必要があります。また、こうした関係者の関連性を接続されたグラフとしてモデル化することが不可欠です。Amazon Neptune は、数十億の相互接続されたエンティティ間の関係を保存およびクエリすることができます。このようにデータをモデル化することで、明確には見えない関係を識別できます。そして、多くの場合、そこに不正取引を見つけることができます。

Amazon Aurora PostgreSQL は 10 分の 1 あるいは 100 分の 1 秒の遅延を提供することができますが、アプリケーションによっては 1,000,000 分の 1 秒レベルの遅延での提供が必要な場合があります。金融取引を扱うアプリケーションでは 1,000,000 分の 1 秒単位で取引を扱うことが不可欠です。

Amazon ElastiCache は、クラウド上で高性能かつ低遅延のキャッシングソリューションを提供します。Amazon Aurora PostgreSQL であまり変更されないミッションクリティカルなテーブルは、1,000 分の 1 秒未満の遅延のニーズに対して Amazon ElastiCache でフロントエンド化できます。Amazon ElastiCache は需要に応じてスケールし、数百万のオブジェクトを保存できるため、キャッシングレイヤーに最適な選択肢です。

IoT ベースの分析アプリケーション

センサーのデータをリアルタイムで読み取り、そのデータを分析してさまざまなアクションを実行するアプリケーションを探してみましょう。例えば、電力網のセンサーが信号を解析用の中央処理装置に送信し、電圧の変動がある場合に修理チームに警告を出すことが考えられます。このアプリケーションは、2つの特定用途向け DB を使用してモデル化できます。

Amazon Timestream は、数百から数千のセンサーによって生成される時系列データを取り込み、迅速に分析するために使用されます。実際、Amazon Timestream は 1 日に数兆のイベントを分析することができます。Amazon Timestream は完全にサーバーレスであり、需要に応じて自動的にスケールアップまたはスケールダウンするため、管理が不要です。Amazon Timestream には集計、ロールアップ、ウィンドウ関数、およびさまざまな数学関数などの組み込みの分析機能があります。これらの組み合わせを使用して、リアルタイムのダッシュボード、信号処理、および警告を行うことができます。時系列データを Amazon Timestream でモデル化することは、リレーショナルエンジンやその他の NoSQL オフオファリングを使用するよりも合理的です。

警告および警告のライフサイクル全体（例：新規、確認済み、作業中、解決済み）は、Amazon DocumentDB のような完全管理型の NoSQL DB でモデル化できます。Amazon DocumentDB はペタバイト規模のデータと毎秒数百万の読み取りおよび書き込みをサポートできます。Timestream と同様に、Amazon DocumentDB もサーバー管理、パッチ適用、ソフトウェアなどの面で顧客の負担を軽減します。Amazon DocumentDB はサーバーレスに設計されています。Amazon DocumentDB は AWS Lambda と統合しており、警告ステータスが変更されたときにカスタム関数を実行するためのトリガーを提供します。

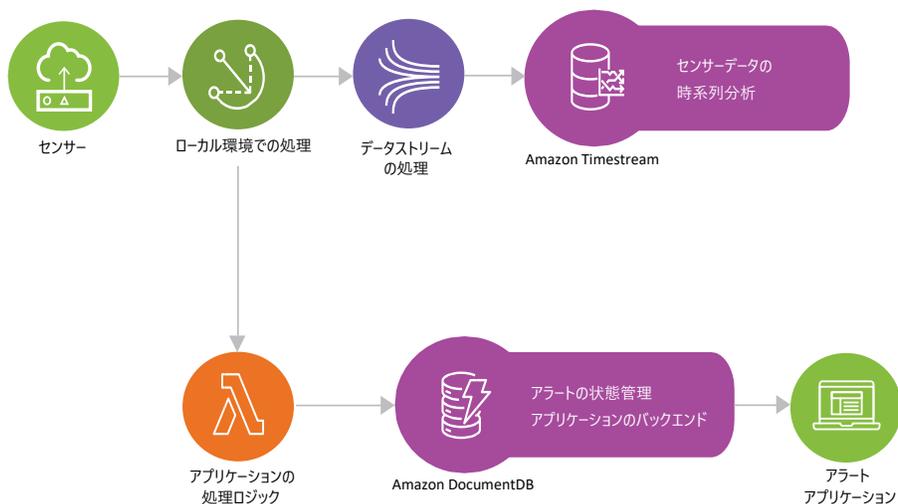


図 5: IoT ベースの警告アプリケーションのための DB パターン

リアルタイムマルチプレイヤーゲーム

マルチプレイヤーゲーム向けに DB の選択肢を評価してみましょう。

ユースケースは次の 2 つのみに限定します：

- 1 リアルタイムのランキング表示の保存と取得
- 2 ユーザープロフィールの保存と取得

ランキング表示は、マルチプレイヤーゲームにおいて高エンゲージメントを維持するための貴重な手段です。ランキング表示は耐久性が鍵となり、遅延も重要です。ランキング表示を Aurora MySQL のようなトランザクション DB に保存し、Amazon ElastiCache のようなキャッシングレイヤーでフロントエンド化するという強行策も考えられます。この方法でもランキング表示は機能しますが、Amazon Aurora とキャッシングレイヤーを単一のトランザクションで更新するコストがボトルネックになる可能性があります。より良い方法は、Amazon MemoryDB for Redis を使用することです。Amazon MemoryDB は高速な DB 復旧と再起動を可能にするマルチ AZ トランザクションログを持ち、強力な耐久性を提供します。Amazon MemoryDB は毎秒数億のリクエストに応じてスケールアップでき、リーダーが 1,000,000 分の 1 秒ごとに変わる可能性があるマルチプレイヤーゲームにとって重要です。

この大規模なスケールと低遅延、高い同時実行性を兼ね備えた Amazon MemoryDB は、ゲームアプリケーションに最適な選択肢です。同時に、ユーザープロフィールにはユーザー名や勝敗などの単純な属性や、獲得した実績の配列、ユーザーが持つインベントリの配列などの複雑な属性があります。これらの属性をモデル化する方法として JSON があり、Amazon DocumentDB (mongoDB 互換) を使用してネイティブフォーマットの JSON を保存するのに非常に適しています。RDB や他の NoSQL DB は、ドキュメントをその自然な形式で管理できる DB ほどのパフォーマンスを発揮しません。Amazon DocumentDB は、ドキュメントの読み取りおよび書き込みを数百万回にスケールし、ペタバイト規模のストレージにも対応できます。DocumentDB はフルマネージド型サービスであり、ライセンス料もかかりません。

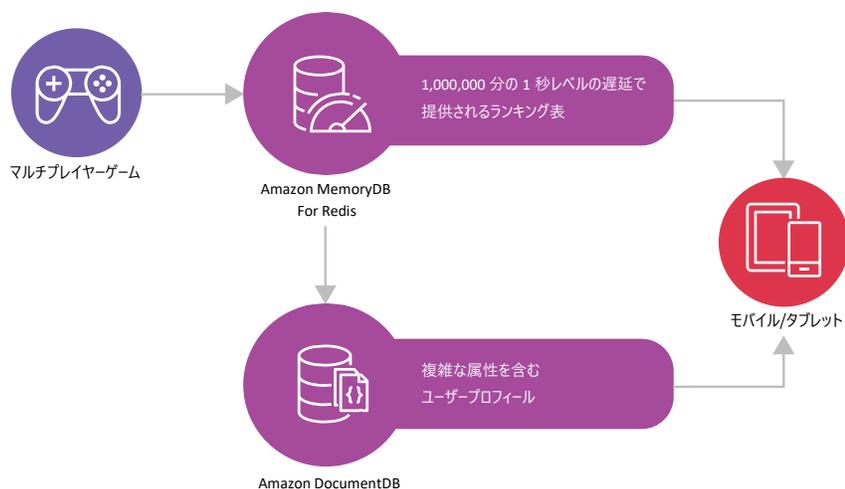


図 6: リアルタイムゲームアプリケーションに推奨される DB

Serverless Databases

「サーバーレスデータベース」はどのようにしてさらなる進化を遂げているのか？

アプリケーションの要求に合わせて DB の変更にかかる時間とリソースを心配していたときを思い出してください。時が経ち、長年の革新によって今日の DB はアプリケーションの需要に基づいて容量が自動的にスケールされるようになり、時間のかかる計画やアップグレードが不要になりました。これがサーバーレス DB の世界です。

サーバーレス DB は、DB インフラの管理と保守に関連する運用上の負担（スケーリング、容量計画、パッチ管理、バージョンのアップグレード）を取り除くことで、顧客は運用作業ではなく、自分たちのビジネスやイノベーションに集中できます。また、高可用性とフォールトトレランス（耐障害性）の利点もあります。これらのサーバーレス DB ワークロードに対して、従量課金モデルを利用することで、顧客は使用した期間のみ料金を支払い、それ以上の費用は発生しません。

サーバーレス DB の一般的なユースケースは以下の通りです：

- 頻繁に使用されないアプリケーション
- 開発/テスト/サンドボックス（テスト環境）のワークロード
- 予測不可能かつ変動するワークロード
- 急速にスケールする必要があるワークロード

ここで、負荷が急激に増加するワークロード、頻度が低いワークロード、周期的なワークロードを持つそれぞれのアプリケーションの例を図 7 に示します。

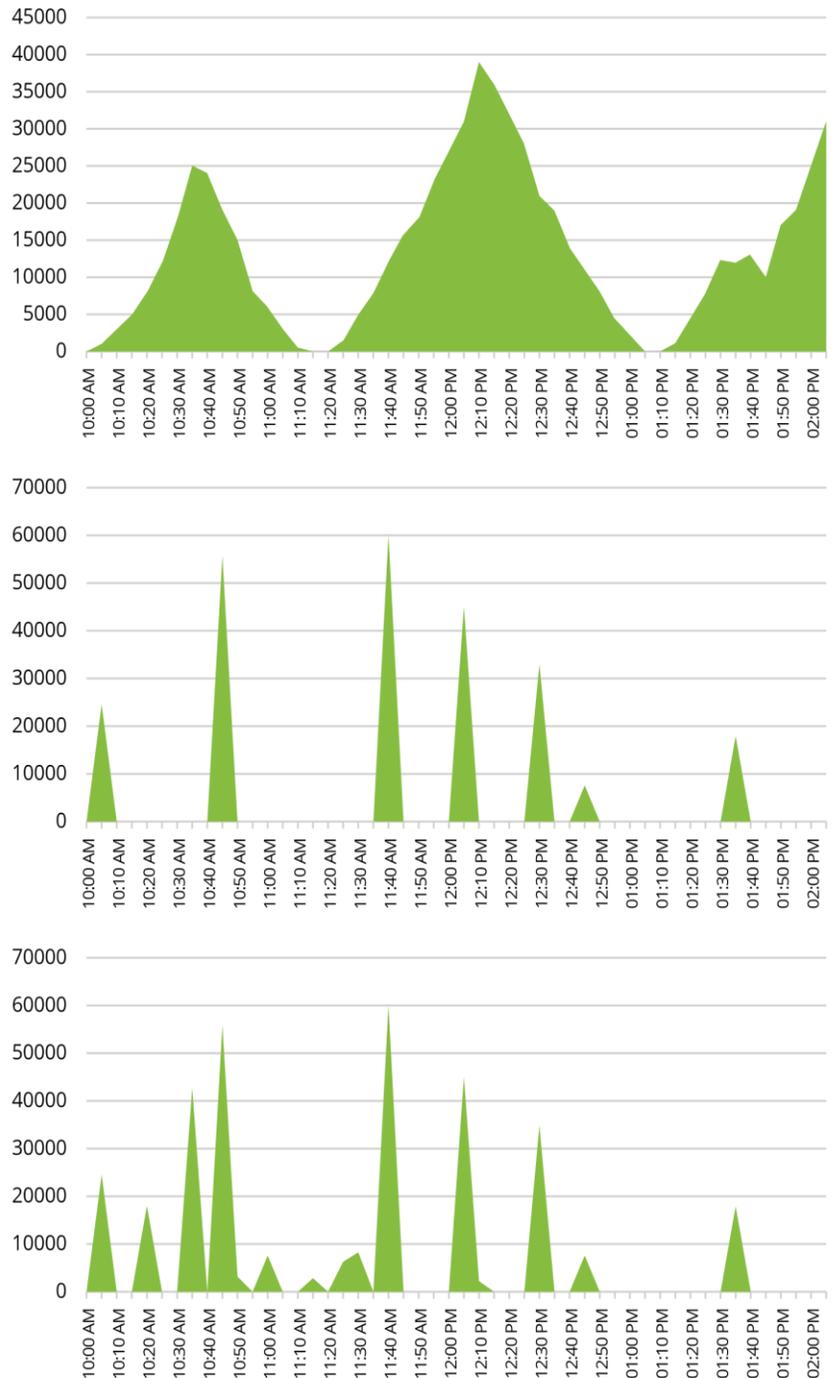


図 7：サーバーレス DB のユースケース³

³ <https://aws.amazon.com/rds/aurora/serverless>

上述したワークロードに対しては、ピーク時に十分でない容量をプロビジョニングすることがあります。これにより、エクスペリエンスの質の低下やパフォーマンスに影響が生じる可能性があります。他の選択肢としては、ピーク時に対応できる容量を提供することですが、これには高額な費用がかかり、ピーク時以外では過剰な容量が発生する可能性があります。さらに別の選択肢として、多数の人員配置・環境の継続的監視・必要に応じたスケールアップやスケールダウンという方法もあります。しかし、これには多くの労力、ダウンタイム、性能の低下が伴います。よって、サーバーレス DB を利用することが、合理的な選択肢と言えます。今日における多くの AWS 目的に応じた DB には、サーバーレスオプションがあります。その中には、Amazon DynamoDB、Amazon QLDB、Cassandra 用の Amazon Keyspaces、Amazon Timeseries など、完全にサーバーレスで提供されるものもあります。Amazon Aurora や Amazon Neptune のように、サーバーレスモードをオプションとして提供する他の DB もあります。

Amazon Aurora Serverless

Amazon Aurora は AWS で急速に成長しているサービスの一つです。Amazon Aurora Serverless は、要求レベルの厳しいアプリケーションや DB ワークロードにも対応しています。Amazon Aurora Serverless は、瞬時に数十万件のトランザクションをわずかな時間でスケールします。スケールする際には、細かい単位で容量を調整し、アプリケーションが必要とする適切な量の DB リソースを提供します。DB の容量を管理する必要はありません。スケールリングは、CPU（中央演算処理装置）の追加およびメモリリソースを追加することで行われ、現在のトランザクションには影響を与えません。コンピュールフリートは継続的に監視され、必要に応じて水平スケールされます。ストレージも需要に応じて増減します（図 8）。

Amazon Aurora Serverless の最も重要なユースケースの一つは、マルチテナントの Software-as-a-Service (SaaS) アプリケーションです。SaaS ベンダーは通常、単一のクラスタ内で数百から数千の AuroraDB を運用し、それぞれが異なる顧客をサポートすることで、利用率とコスト効率を改善します。それでも、各 DB を個別に管理する必要があり、同じクラスタ内に共存する DB が当初の計画以上に共有リソースを消費する場合に備える必要があります。

アプリケーション

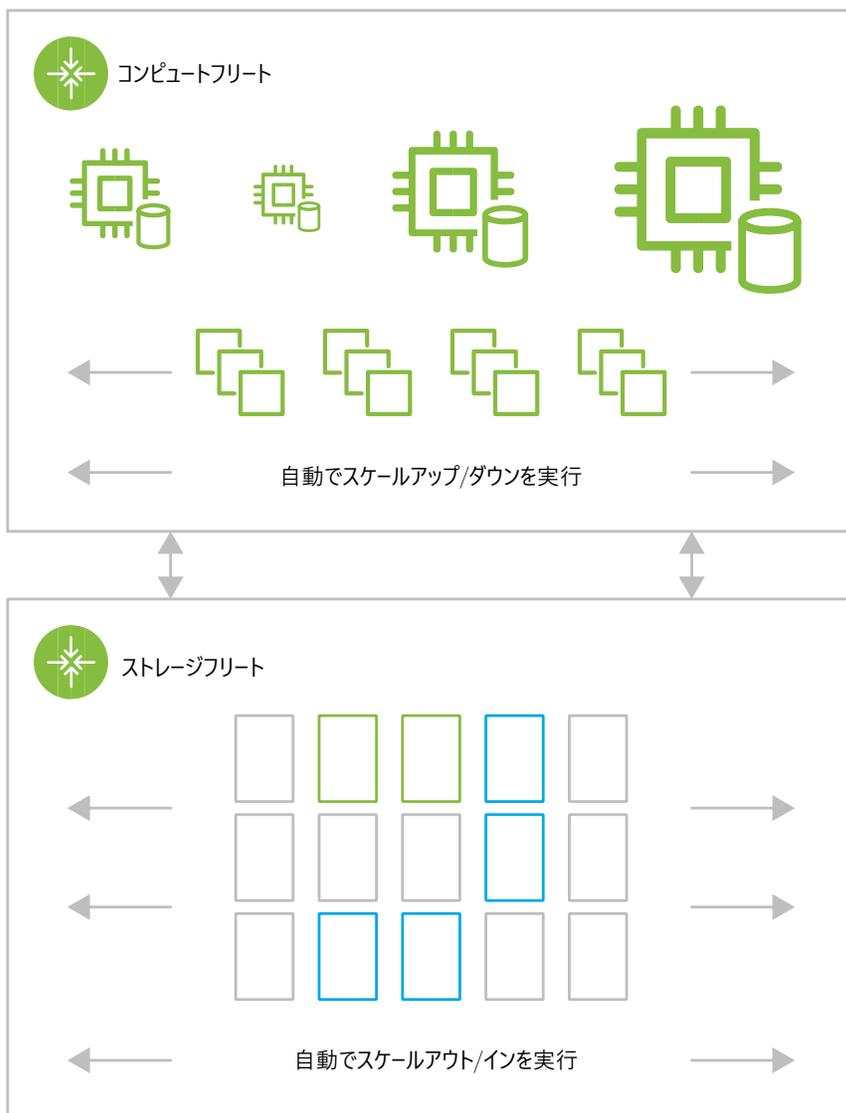


図 8 : Amazon Aurora Serverless のアーキテクチャ⁴

Amazon Aurora Serverless v2 を使用することで、SaaS ベンダーはプロビジョニングされた容量に対するコストを心配することなく、顧客ごとに AuroraDB クラスタをプロビジョニングすることができます。使用されていないときは、コスト削減のために自動で 0.5 Aurora Capacity Units (ACU) までスケールダウンし、アプリケーションの要件の変化に応じて DB の容量を即座に調整します。

⁴ <https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-serverless-v2.how-it-works.html>

Amazon Neptune Serverless

グラフ DB は今なお新しい技術であり、時折、顧客が典型的な OLAP/OLTPDB 上でグラフワークロードを実行しているのを見かけます。一部の顧客は、グラフ DB の容量管理が非常に難しいと感じています。Amazon Neptune は難しいグラフ DB の容量管理のためにサーバーレス DB の提供を開始しました。Amazon Neptune は、瞬時に容量を秒単位でスケールアップすることができます。容量は細かく増減できるため、ピーク容量に合わせてプロビジョニングする場合と比較して、最大 90%のコスト削減が可能です。

Amazon Neptune サーバーレスは以下に基づいてスケールを決定します：

- フォアグラウンドおよびバックグラウンドプロセスの CPU 使用率
- バッファプールのような内部データ構造のメモリ使用率

データベース 移行事例

このセクションでは、デロイトがクライアントのために実装した目的に応じた DB の 3 つの事例を紹介します。



デロイトのケーススタディ 1

Executive Summary

多国籍企業においては、専門家が情報に基づいた意思決定を行えるようにするために、ソフトウェアを利用した信頼できるコンテンツを提供します。同企業は、パフォーマンスとスケーラビリティを向上させ、外部依存とコストを削減し、顧客体験を充実させることを目標として、コンテンツおよびリサーチインテリジェンスの能力をモダナイズしようとしました。

目的に応じた DB の必要性

ソリューションの一環として、デロイトチームはコンテンツの簡素化と分類に注力しました。これには、取得、取り込み、処理、編集、そしてコンテンツを公開するという複数段階のアプローチが含まれており、全体的なパフォーマンスを向上させ、コンテンツのトレーサビリティを維持することを目的としています。ワークフローのためのメタデータを提供し、サービス実行間でコンテンツの動作や制約条件を維持するために、Amazon DocumentDB が選定されました。Amazon DocumentDB のグローバルテーブル機能を活用して、マルチリージョンデータ同期の必要性を排除し、パフォーマンスを最適化しました。デロイトはまた、全世界で 3 億以上の組織のトランザクションデータ（更新および取得）を効率的に処理するために Amazon RDS for PostgreSQL を導入しました。

解決策

構築されたソリューションは、現代の課題に適応するために取り組んでいるデジタルトランスフォーメーションに適しています。Amazon DocumentDB が選定された理由は次の通りです：

- ワークフローのメタデータを保存し、マイクロサービス間のコンテキストを維持するため
- 設定の保存場所として機能し、コードのデプロイ依存を減らし、すべての設定を集中管理するため
- 異なるリージョンにおけるグローバルテーブルは、複数のリージョン間でのデータ同期の必要性を排除します
- インフラのスケーラビリティを気にすることなく、迅速かつ容易にコンテンツ設定の変更を動的に行う

Amazon RDS for PostgreSQL も活用され、3 億以上の組織のトランザクションデータを処理するために、データ損失やパフォーマンスの遅延なく同時更新および取得を処理することが求められました。

成果と利点

デロイトはクライアントである世界中の法律、税務、および会計の専門家のグローバルな顧客基盤にサービスを提供するレガシーなコンテンツおよびリサーチインテリジェンスプラットフォームを変革しました。

インフラのスケーラビリティを気にすることなく、迅速かつ容易にコンテンツ設定の変更を動的に行う AWS の目的に応じた DB とクラウドネイティブサービスを使用することで、クライアントはこれからのビジネスに必要な不可欠な情報を得ることができるようになりました。成果には以下が含まれます：

- スケーラブルな Amazon RDS for PostgreSQL を使用することで、40%増加した 1 日あたりのトランザクション量
- 世界中の 3 億以上の組織のコンテンツを処理するための 1,000 分の 1 秒未満の応答時間
- DynamoDB グローバルテーブルを使用した可用性、耐久性、およびマルチリージョンのフォールトレランスの向上
- Amazon RDS for PostgreSQL を使用したクラスタ管理の自動化、暗号化、プライバシー、セキュリティ、および高速化と災害復旧のための複数のレプリカインスタンス

同時更新間で最後に書き込んだものを優先的に処理する（Last-Writer-Wins）調整を使用して、一貫性の向上と競合解決を改善させることで、すべてのレプリカが世界中で一貫したコンテンツ状態を提供することを保証します。

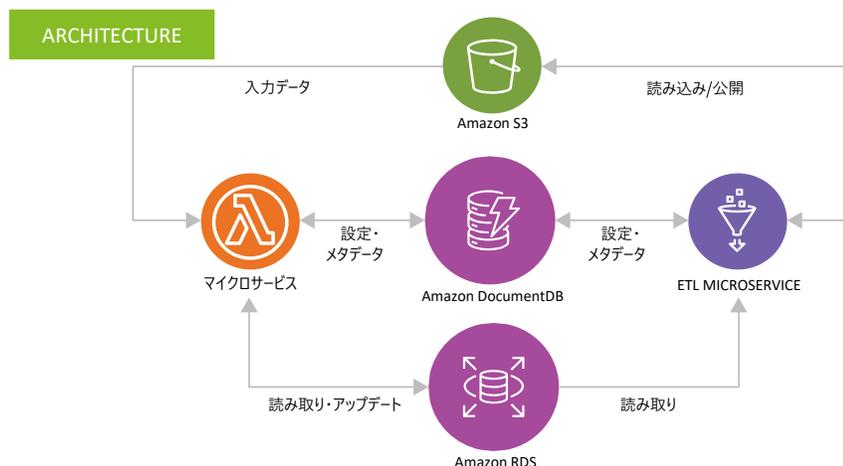


図 10: グローバルなコンテンツ管理のためのクライアント向け最新 DB アーキテクチャソリューション

デロイトのケーススタディ 2

Executive Summary

デロイトの ConvergeHEALTH™ CognitiveSpark™ for Clinical は、モジュール式、クラウドベース、およびメタデータ駆動のソリューションです。これらのソリューションは、臨床試験のライフサイクル全体にわたるデータ管理を自動化し、さまざまな入力文書や提供元から構造化、標準化された成果物をデジタルデータ上に生成するよう設計されています。こうした要素は、下流システムの設定、必要なレポートと分析の自動入力、重要な試験成果物のコンテンツ生成のために効率的に解釈され、変換されます。

目的に応じた DB の必要性

デロイトのチームは、コンテンツの簡素化と分類に注力し、取得、取り込み、処理、編集、およびメタデータの公開という多段階のアプローチを採用しました。この方法により、全体的なパフォーマンスの向上とコンテンツのトレーサビリティの維持が図られます。プロトコル要素、研究定義、生物医学的概念、内部（スポンサー）および外部管理基準（CDISC およびその他）のノード間における関係性（エッジリレーションシップ）を管理するために、Amazon Neptune が選ばれました。また、20 以上の臨床データソースのトランザクションデータ（更新および取得）を効率的に処理するために、Amazon RDS for PostgreSQL が導入されました。

解決策

このソリューションは、デジタルデータ変換を推進するための高度なメタデータ基盤を提供し、多くの組織が新薬を市場に投入するまでの時間を短縮するために取り組んでいます。Amazon Neptune が導入された理由は以下の通りです：

- データ収集フォーム、データ集計基準、および管理用語を設定するためのメタデータを保存するため
- 品質、統合性、コンプライアンスのためのコンピュータで読み取り可能なルールを保存し、臨床開発ライフサイクル全体の信頼できる情報源として機能するため
- 内部および外部のメタデータを接続し、ビジネスプロセス全体にわたるエンドツーエンドのトレーサビリティを提供するため
- 垂直および水平のスケーラビリティ：
 - 外部の属性、語彙、および辞書と統合する
 - 競争情報からのさらなるデータの追加を可能にする（多くのデータが非構造化コンテンツとして利用可能）
 - 基盤となるインフラストラクチャのスケーラビリティを気にせずに、メタデータの変更を動的に行うための迅速で手間のかからない方法

Amazon RDS for PostgreSQL も導入され、20 以上の臨床データ提供元からのトランザクションデータを処理するために使用されました。これにより、データ損失やパフォーマンスの遅延なく、同時更新とデータの取得が可能となりました。

成果と利点

デロイトは、AWS の目的に応じた DB とクラウドネイティブサービスを使用して、ConvergeHEALTH™ CognitiveSpark™ for Clinical platform を円滑に立ち上げ、臨床開発の利害関係者（臨床医、データ管理者、生物統計学者、医学ライター）に未来の臨床データフローを再構築するための情報を提供しました。その結果として、以下の成果が得られました：

- 自動化された品質評価のためのスケーラブルなデータ基盤、生データセット、強化データセット、分析データセットを統合したストレージ
- 20 以上の臨床データ提供元から送られるデータを処理するためにかかる 1,000 分の 1 秒未満の応答時間

- Amazon Neptune を使用した可用性、耐久性、およびマルチリージョンでのフォールトトレランス向上
- Amazon RDS for PostgreSQL を使用した自動クラスタ管理、暗号化、プライバシー保護、セキュリティ、および複数のレプリカインスタンスによる高速処理と災害復旧
- 全世界で一貫したコンテンツ状態を提供するために、同時更新時の最終書き込み者優先（last-writer-wins）調整を使用して、一貫性の向上と競合解決の改善

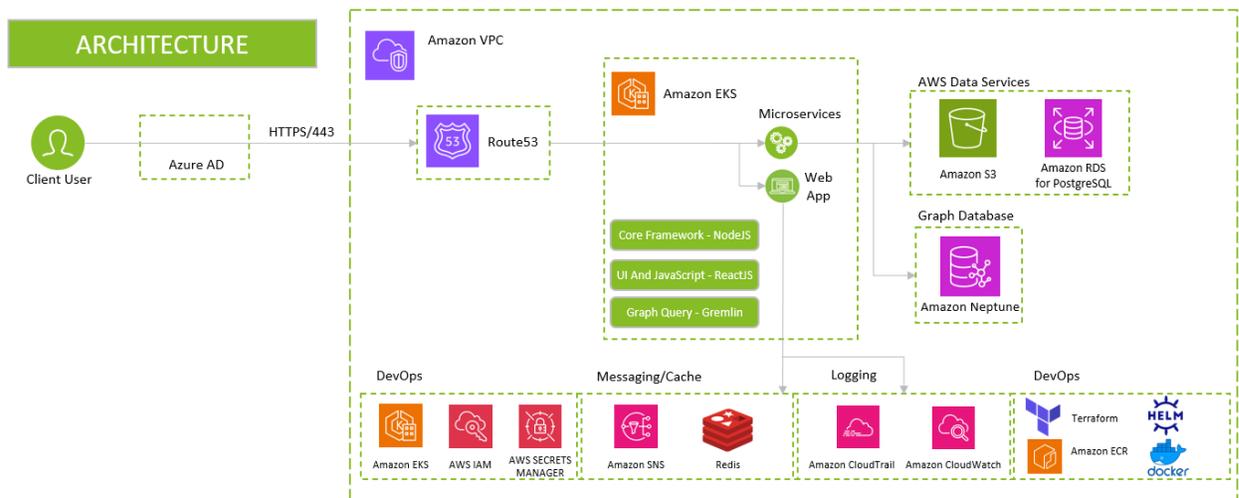


図 11：臨床開発のユースケースに特化した DB ソリューション

まとめ

数年前に構築された企業のレガシーデータプラットフォームをモダナイズすることは難しく思えるかもしれませんが、適切なアプローチとテクノロジープラットフォームを用いることで、その成果は非常に価値のあるものとなります。

今日の市場にはクラウドのモダナイゼーションを支援するために目的に応じたDBの選択肢が豊富にあり、モダナイゼーションの取り組みをより複雑にしています。しかし、その一方で、適切なツールを使用することで組織は多大なる利益を享受できます。

戦略的にDBモダナイゼーションを成功させるためには、各種ワークロードに対する利益とリスクの分析が必要です。オリジナルおよびオープンソースの両機能を提供し、エコシステムを強くサポートする技術パートナーを選ぶことが重要です。

DBのモダナイゼーションジャーニーを始める際、AWSは以下のガイダンスを提供します。

- ユースケースを用いてAWSが提供する目的に応じたDBを評価し、どのDBサービスが適しているかを絞り込むための要件を定義する
- 選択したDBオプションを、市場投入までのスピードやチームのスキルセット、将来のニーズとのバランスを考慮しながら重ね合わせる
- モダナイゼーションは単に革新的な技術に関するものだけでなく、組織文化の変革を伴うものであり、スキルアップやトレーニングが必要となることを認識する

執筆者

本文書の執筆者は以下の方々です。

Deloitte Consulting LLP

Tony Witherspoon

PPMD

Cloud Engineering (Architecture & Strategy)

Sameer Limaye

Specialist Leader

Cloud Native Development

Deloitte Tohmatsu Consulting LLC

Yamaguchi, Hirofumi

Partner

Engineering, AI & Data (Engineering)

Lee, Grace Jahyun

Director

Engineering, AI & Data (Engineering)

Nishida, Tomohiro

Analyst

Engineering, AI & Data (Engineering)

Majima, Hidesato

Analyst

Engineering, AI & Data (Engineering)

Amazon Web Services

Kasi Muthu

Senior Partner Solutions Architect

Data, ML

Vishal Srivastava

Senior Partner Solutions Architect

Databases

Grant Schonberg

Principal Tech BD

Databases

Amazon Web Services Japan G.K

Shinji Kusunuki

Sr. Partner Solution Architect

Japan Partner Management

改訂履歴

日付

改訂内容

2023 年 5 月 30 日

初稿完成

初版発行

2024 年 12 月 18 日

日本語版完成

日本語版発行



免責事項

本書における情報についてはお客様ご自身で判断を行う必要があります。本書は(a) 情報提供のみを目的としています。(b) 現在の AWS 製品の提供内容や事例を示しており、予告なく変更される場合があります。(c) AWS およびその関連会社、供給事業者、ライセンス契約事業者からのいかなる約束や保証も生じさせるものではありません。AWS の製品およびサービスは「現状のまま」提供され、明示または黙示を問わず、いかなる種類の保証、表明、条件も伴いません。AWS のお客様に対する責任と義務は AWS との契約によってのみ制御され、本書はその契約の一部ではなく、AWS とお客様との間でのいかなる契約も変更するものではありません。

© 2023 Amazon Web Services, Inc, or its affiliates.

Deloitte.

デロイト トーマツ

デロイト トーマツ グループは、日本におけるデロイト アジア パシフィック リミテッドおよびデロイト ネットワークのメンバーであるデロイト トーマツ 合同会社ならびにそのグループ法人（有限責任監査法人 トーマツ、デロイト トーマツ リスク アドバイザリー 合同会社、デロイト トーマツ コンサルティング 合同会社、デロイト トーマツ ファイナンシャル アドバイザリー 合同会社、デロイト トーマツ 税理士 法人、DT 弁護士 法人およびデロイト トーマツ グループ 合同会社を含む）の総称です。デロイト トーマツ グループは、日本で最大級のプロフェッショナル グループのひとつであり、各法人がそれぞれの適用法令に従い、監査・保証業務、リスク アドバイザリー、コンサルティング、ファイナンシャル アドバイザリー、税務、法務等を提供しています。また、国内約 30 都市に約 2 万人の専門家を擁し、多国籍企業や主要な日本企業をクライアントとしています。詳細はデロイト トーマツ グループ Web サイト、www.deloitte.com/jp をご覧ください。

Deloitte（デロイト）とは、デロイト トウシュート マツ リミテッド（“DTTL”）、そのグローバル ネットワーク 組織を構成するメンバー フォーム および それらの関係法人（総称して“デロイト ネットワーク”）のひとつまたは複数 を指します。DTTL（または“Deloitte Global”）ならびに各メンバー フォーム および 関係法人 はそれぞれ法的に独立した別個の組織体であり、第三者に関して相互に義務を課しまたは拘束させることはありません。DTTL および DTTL の各メンバー フォーム ならびに 関係法人 は、自らの作為および不作為 についてのみ責任を負い、互いに他のフォーム または 関係法人 の作為 および 不作為 について責任を負うものではありません。DTTL はクライアントへのサービス提供を行いません。詳細は www.deloitte.com/jp/about をご覧ください。デロイト アジア パシフィック リミテッドは DTTL のメンバー フォーム であり、保証 有限責任 会社です。デロイト アジア パシフィック リミテッドのメンバー および それらの関係法人 は、それぞれ法的に独立した別個の組織体であり、アジア パシフィック における 100 を超える都市（オーランド、バンコク、北京、ベンガルール、ハノイ、香港、ジャカルタ、クアラルンプール、マニラ、メルボルン、ムンバイ、ニューデリー、大阪、ソウル、上海、シンガポール、シドニー、台北、東京を含む）にてサービスを提供しています。

Deloitte（デロイト）は、監査・保証業務、コンサルティング、ファイナンシャル アドバイザリー、リスク アドバイザリー、税務・法務などに関連する最先端のサービスを、Fortune Global 500® の約 9 割の企業や多数のプライベート（非公開）企業を含むクライアントに提供しています。デロイトは、資本市場に対する社会的な信頼を高め、クライアントの変革と繁栄を促し、より豊かな経済、公正な社会、持続可能な世界の実現に向けて自ら率先して取り組むことを通じて、計測可能で継続性のある成果をもたらすプロフェッショナルの集団です。デロイトは、創設以来 175 年余りの歴史を有し、150 を超える国・地域にわたって活動を展開しています。“Making an impact that matters”をバース（存在理由）として標榜するデロイトの 45 万人超の人材の活動の詳細については、www.deloitte.com をご覧ください。

本資料は皆様への情報提供として一般的な情報を掲載するのみであり、デロイト トウシュート マツ リミテッド（“DTTL”）、そのグローバル ネットワーク 組織を構成するメンバー フォーム および それらの関係法人 が本資料をもって専門的な助言やサービスを提供するものではありません。皆様の財務または事業に影響を与えるような意思決定または行動をされる前に、適切な専門家に相談ください。本資料における情報の正確性や完全性に関して、いかなる表明、保証または確約（明示・黙示を問いません）をするものではありません。また DTTL、そのメンバー フォーム、関係法人、社員・職員または代理人のいずれも、本資料に依拠した人に関係して直接または間接に発生し得る損失および損害に対して責任を負いません。DTTL ならびに各メンバー フォーム および 関係法人 はそれぞれ法的に独立した別個の組織体です。

Member of
Deloitte Touche Tohmatsu Limited

© 2025. For information, contact Deloitte Tohmatsu Group.



IS 669126 / ISO 27001



BCMS 764479 / ISO 22301