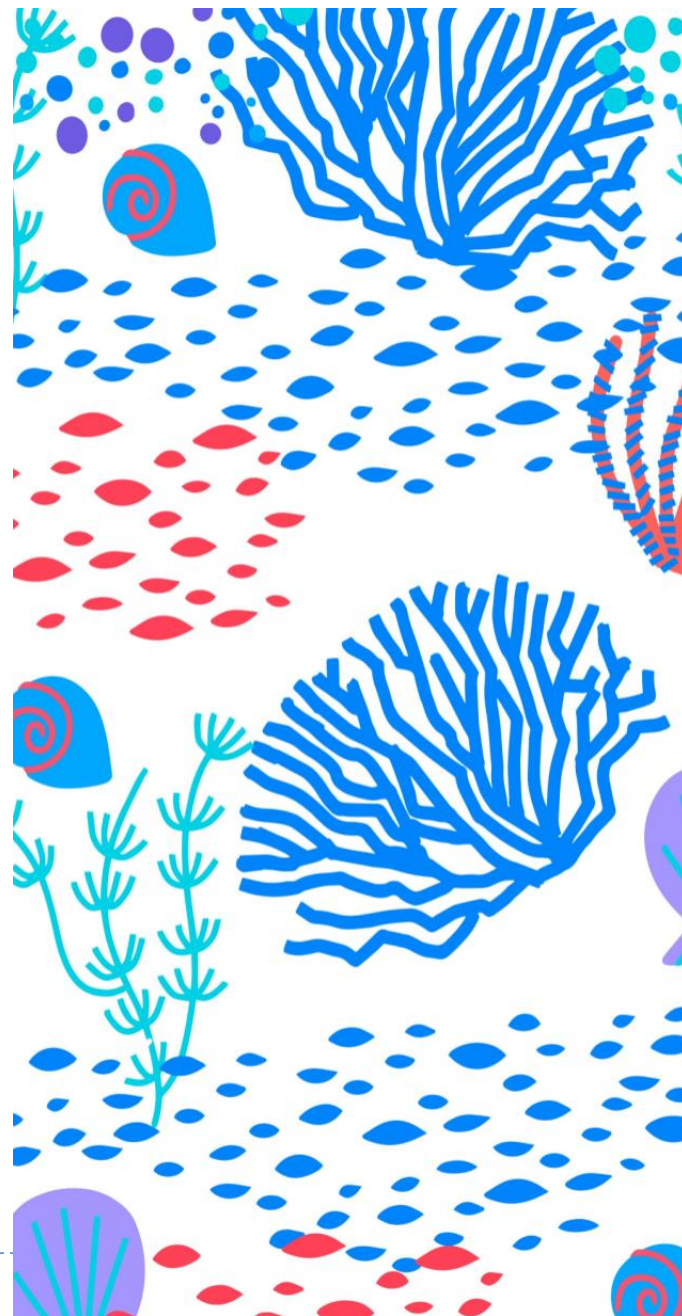


AWSのマネージド データベースを知る

AWS上でシステムを開発したいけれど、どのデータベースが適切なのかわからない。RDSなのかAuroraなのか？どちらを採用すべきなのか、知識が足りなくて選択できない。そんなお客様のためのホワイトペーパーです。



partner
network



目次

第一章	AWSが扱うデータベースを広く理解する	P3
第二章	Amazon RDSの特徴を理解して活用しよう	P8
第三章	AWS Auroraとは？その特徴は？	P14
第四章	AWS Aurora Serverlessを最新機能(v2)を含めて解説	P19
第五章	Auroraか(非Auroraな) RDSか。比較した上でどちらを使う？ . . .	P24
第六章	AWSのNoSQLの代表、Amazon DynamoDBとは	P29
	※だれでもわかるスタイルズのオンラインウェビナーのご案内	P35

第一章 AWSが扱うデータベースを広く理解する

1. マネージドサービスとしてのデータベースとは？

AWS(Amazon Web Services)の利用におけるメリットの一つとして、マネージドサービスの存在があります。また、これらマネージドサービスでは様々な種類のサービスがリリースされていますが、その中でも最も利用されているものとしては、データベースのマネージドサービスが存在しています。以降では、AWSにおけるマネージドサービスとしてのデータベースのメリットや種類について、どのようなものがあるのか、記載していきます。

① マネージドサービスとは

マネージドサービスとは、クラウドが提供するサービス形態の一つです。クラウドが提供するサービス形態としては、IaaS (Infrastructure as a Service)、PaaS (Platform as a Service)、SaaS (Software as a Service)、FaaS (Function as a Service) などがあります。このうち、クラウドサービス提供側が実際のサーバーを用意し、管理を行うPaaS、SaaS、FaaSが、『マネージドサービス』に該当します。つまり、クラウドサービス提供者が物理サーバを用意し、設定、運用管理を行っているため、“マネージド（運用管理された状態にある）サービス”と呼びます。

② データベースの運用の難しさ

データベースの運用はほかのミドルウェアと比べて下記の面で難易度が高いと言われています。

- キャパシティ予測が難しい

データベースがどのような頻度で利用され、どのくらいのデータを保持するか、といった予測を正確に事前に立てることは非常に難しいです。また、一時的なアクセスのスパイク（急上昇）などもいつ発生するか、どのくらいのアクセスが行われるかは予測ができません。

- 設定項目が多い

セキュリティやパフォーマンスの設定が多岐にわたり、設定項目が非常に多く、多くのエンジニアを悩ませています。安全かつ適切に利用するためには深い知見を持った専門的なエンジニアが必要です。

③マネージドサービスとしてのデータベースのメリットとは？

マネージドサービスとしてのデータベースのメリットとは、『データベースの難しい運用をAWSに所属する凄腕エンジニアに委任できる』ということにつきます。したがって、運用にかかるコストを少なくするとともに、データベースの安定稼働を実現することができます。また、運用に割いていた人的リソースやコストを他に注力させることができます。

マネージドデータベースのメリット



最大のメリットは運用
コスト（＝人件費）
の削減

運用コストを大幅に削減

2. データベースの適材適所

AWSが提供するデータベースサービスには多様な選択肢があり、アプリケーションに応じて最適なサービスを選択する必要があります。どのような種類があるのか、具体的に見ていきましょう。

①RDBMS（リレーショナル・データベース）

RDBMSとは、Relational DataBase Management Systemの略で、属性を列、組を行とする表（テーブル）の形でデータを格納するデータベースです。SQLと呼ばれる言語を用いてデータの参照や加工をおこないます。RDBMSは不整合のあるデータの記録を拒否するなどしてデータベースの整合性を保つ機能や、権限のない利用者による不正な読み出しや改竄などからデータを保護する仕組みを持っています。また、関連する複数の処理を一体化して矛盾なく実行する機能（トランザクション管理機能）もあり、データの正確性に強みを持っています。一方で、SQLによる読み書きに時間がかかってしまうデメリットがあり、非常に高速にデータを読み書きするようなアプリケーションには不向きな場合もあります。

②NoSQLとは？

NoSQLはRDBMSではないデータベース全般を指します。キーバリューストア型データベース（Amazon DynamoDB）や、グラフ型（Amazon Neptune）など、さまざまな種類があり、RDBMSと比較してデータの整合性には多少の弱みがありますが、データの格納形式がRDBMSよりも複雑ではないため、データの書き込みやデータ参照が膨大になるシステムに向いています。

③Amazonが提供するデータベースの種類

Amazonでは、『万能のデータベースは存在しない』（出典：<https://www.allthingsdistributed.com/2018/06/purpose-built-databases-in-aws.html>）の思想から、様々なマネージドデータベースサービスが提供されています。以下に具体的なサービス名と、分類について記載していきます。

- リレーショナルデータベース(RDBMS)

Amazon RDS (MySQL、MariaDB、PostgreSQL、商用としてのOracle、SQL Server)

Amazon Aurora (MySQL互換、PostgreSQL互換)

※AWS上の区分としては、AuroraもRDSに含まれるのですが、ここでは別個のものとして区分しています。

- NoSQLの様々な種類

- ・ キーバリューストア型データベース

Amazon DynamoDB

- ・ ドキュメントデータベース

Amazon DocumentDB

- ・ インメモリデータストア

Amazon ElastiCache

- ・ グラフデータベース

Amazon Neptune

- ・ 時系列データベース

Amazon Timestream

- ・ 台帳データベース

Amazon QLDB

リレーショナルとKey-valueの特徴

	リレーショナル	Key-Value (NoSQL)
特徴	・ 参照整合性 ・ ACIDトランザクション ・ Schema-On-Write	高スループット 低レイテンシーでの読み取り、書き込み 高いスケール性
利用局面	既存システムの移行 業務系システム 基幹システム	ソーシャル ゲーム ショッピングカート ECでの商品DB

3. Amazon Relational Database Service(Amazon RDS)の特徴

Amazon RDSは上記の通り、RDBMSとして提供されるマネージドデータベースです。6つのデータベースエンジン MySQL、PostgreSQL、MariaDB、MicrosoftSQLServer（商用）、Oracle（商用）から選択が可能となっており、様々な用途や要件に対応しています。Amazon RDSはPaaSの形式で提供されており、ストレージ容量の拡張と縮小が自在にできたり、自動でバックアップの取得やパッチ適用が可能となる機能があたりと、一般的なデータベースと比較して大幅に運用が楽になっています。

4. Amazon Auroraの特徴

Amazonがクラウド時代に向けて再設計した、高性能の商用データベースの速度や信頼性と、オープンソースデータベースのシンプルさや費用対効果を兼ね備えたリレーショナルデータベースエンジンです。選択できるデータベースエンジンはMySQL、PostgreSQLがあり利用可能なバージョンも制限されますが、機能は大幅に向上しています。大きな違いとしては、Auroraは（非Auroraの）RDSと違って、データベースが稼働しているサーバーとストレージが分離していることと、そのストレージが6箇所のストレージにコピーされる点です。したがって、1、2個のストレージでネットワーク障害、ディスク破損などの問題が起こったとしても、アプリケーションに影響がないように、高速な読み書きや可用性の向上が可能になっています。

5. AWSのNoSQLの代表選手 Amazon DynamoDB

RDSやAuroraの次にAWSのデータベースとして有名なサービスが、Amazon DynamoDBです。Amazon DynamoDBはキーバリューストア型のマネージドデータベースで、データをキーとバリューという一対一の構造で保持します。データの格納と取得に特化しており、IoTアプリケーションやゲームアプリケーションなどの、1桁ミリ秒単位の読み書きを要求する場合に適しています。こちらでも、データのバックアップやストレージの拡張などを自動実施する機能があり、運用における負荷が軽減されています。

第二章 Amazon RDSの特徴を理解して活用しよう

1. Amazon RDSとは？

Amazon RDS (Amazon Relational Database Service) とは、その名称のとおりAWS(Amazon Web Services)が提供するリレーショナルデータベースのサービスです。一般的に「データベース」というと、サーバーにミドルウェアとしてインストールし、稼働させるケースが考えられますが、RDSはAWSのマネージドサービスとして提供されています。

リレーショナルデータベースはその中の1形式で、属性を列、組を行とする表（テーブル）の形でデータを格納するデータベースです。Excelのような形式でデータが格納されていく、とイメージするといいいでしょう。データベースの中では最も使用されている、一般的な形式です。データの参照や加工といった操作は、SQLと呼ばれる言語を用いておこないます。リレーショナルデータベースは不整合なデータの記録を拒否するなどしてデータベースの整合性を保つ機能や、権限のない利用者による不正な読み出しや改竄などからデータを保護する仕組みを持っています。また、関連する複数の処理を一体化して矛盾なく実行する機能（トランザクション管理機能）もあり、データの正確性に強みを持っています。

2. Amazon RDSで利用できるデータベースエンジンの種類

Amazon RDSでは、Amazon Aurora（MySQL互換、PostgreSQL互換）、MySQL、PostgreSQL、MariaDBが選択可能で、商用データベースでもMicrosoft SQL Server、Oracleが選択可能です。一般的にOracleのような商用データベースを利用する場合はライセンス料の支払いが必要ですが、基本的にRDSの料金にライセンス料も含まれています。また、BYOL（所有しているライセンスの持ち込み）も可能です。

3. Amazon RDSのインスタンスタイプと料金体系

EC2と同様に、Amazon RDSもまったく自由にCPUのコア数やメモリの容量を変えられるわけではありませんが、さまざまなインスタンスタイプが用意されており、用途に合わせて最適なインスタンスタイプを選択可能です。インスタンスタイプとしては、大枠で汎用（幅広い用途向け）タイプのほかに、データベースはメモリを非常によく使うためメモリ最適化タイプが用意されています。



Amazon RDSの課金はインスタンスの起動時間、ストレージに保存しているデータ容量、データ転送量に応じて行われます。後述するマルチAZ配置の場合、スタンバイインスタンスの費用も発生するので、シングルAZに比べ料金はおおよそ倍になります。なお、OracleやSQL Serverといった商用製品は、BYOL（自前で調達したライセンスを利用すること）が可能で、その場合は料金が安くなります。また、EC2と同じようにリザーブドインスタンスやSavings Plansが利用可能で、リザーブドインスタンスを利用する場合は最大7割程度利用料金を節約することが可能です。

4. マルチAZ配置、リードレプリカ

Amazon RDSの大きな特徴として、マルチAZ配置オプション、リードレプリカオプションが用意されていることがあります。それぞれどのような機能なのか、具体的に記載していきます。

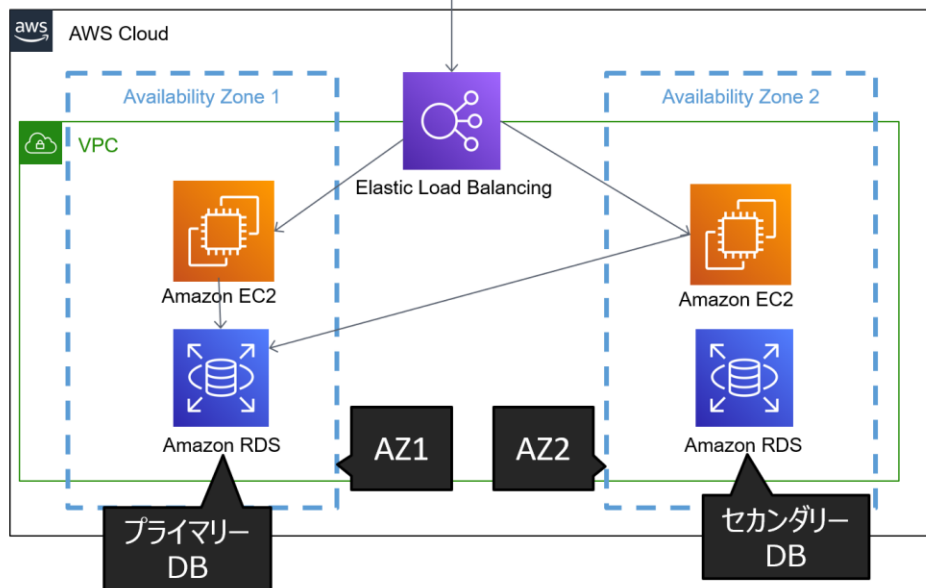
①可用性とスケーラビリティも高める新「マルチAZ配置オプション」

「マルチAZ」とは、「マルチ(複数)のAZ(データセンター群)」を意味しています。AWSでは、電力系統がそれぞれ独立しており、かつ高速のNWで接続されている複数のデータセンター群をアベイラビリティゾーン(AZ:Availability Zone)と呼んでいます。マルチAZ構成を行うと、スタンバイDBが別のAZに構築され、データが同期更新されるようになります。1つのAZで障害が発生して、プライマリのデータベースに障害が発生したりすると、自動的にスタンバイに切り替わります。このように、マルチAZ構成を用いることによって一つのAZで障害が発生しても他の正常なAZを使用して稼働を継続させることができ、データベースの可用性を向上させることができます。例えば、稼働率が99%のデータベースをシングルAZで稼働させた場合と、マルチAZで稼働させた場合を比較してみましょう。

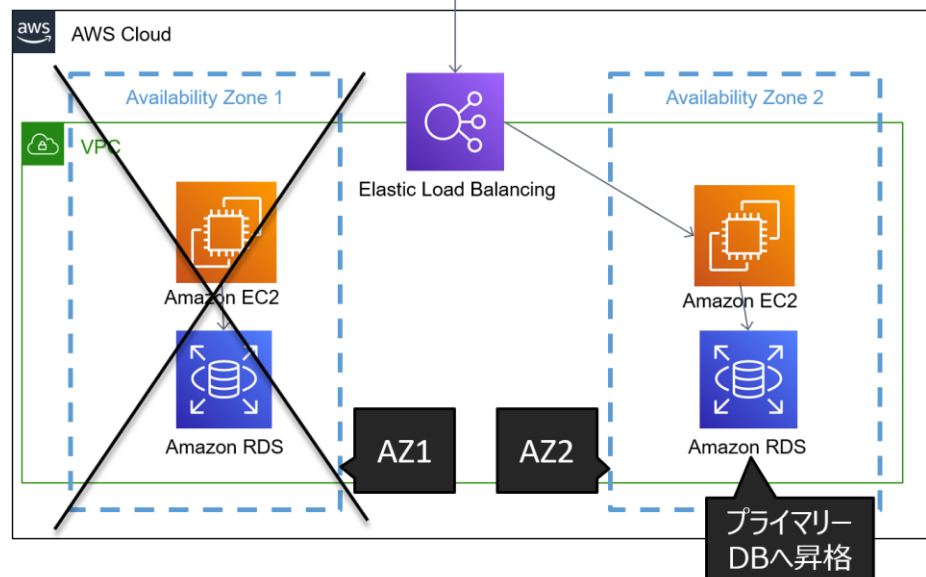
- ・1つのAZで稼働させた場合の可用性 = 99%（年間約87.60時間程度停止）
- ・2つのAZで稼働させた場合の可用性 = $1 - (1 - 0.99) * (1 - 0.99) = 99.99\%$ （年間で約0.87時間停止）

上記を見てみると、停止時間で考えた場合、マルチAZ構成の場合は約100倍可用性が向上しています。デメリットとしては、先述の通り利用料金が倍になることが挙げられます。

RDSのマルチAZ配置：障害発生前



RDSのマルチAZ配置：障害が発生すると。。。

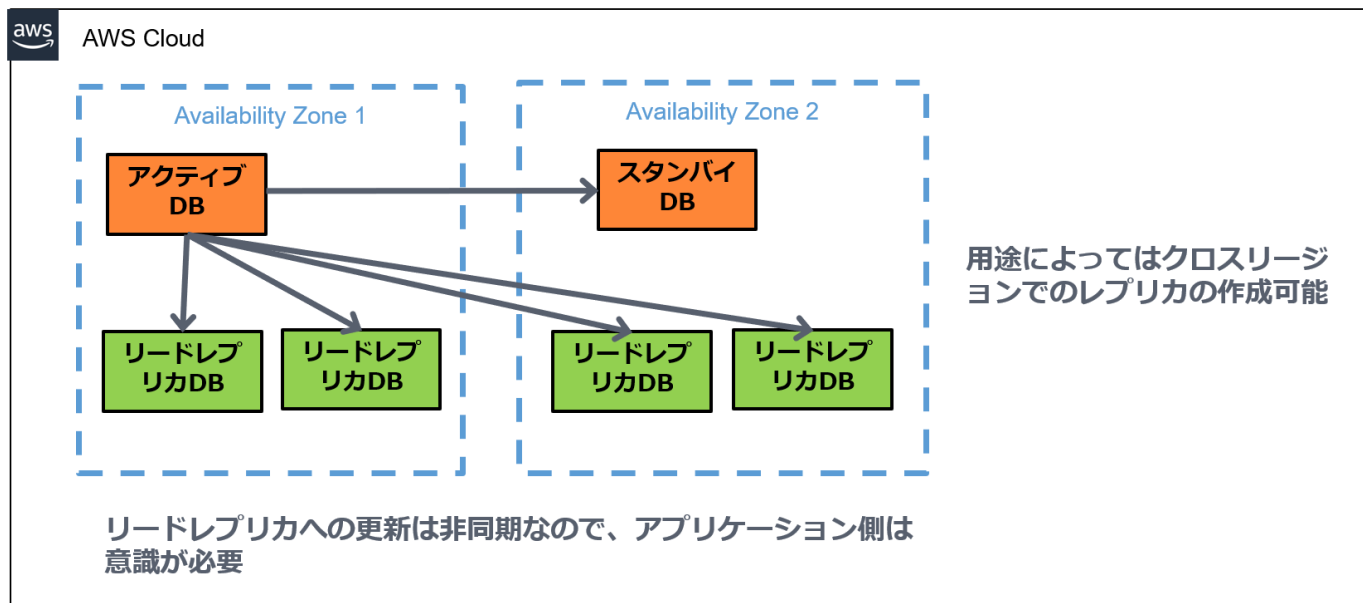


②データの読み取り性能を向上させる「リードレプリカオプション」

リードレプリカは読み取り専用のDBのことです。アクセス数が多くDBに読み取り負荷が起きている場合などは、リードレプリカを構築することでプライマリデータベースへの負荷を分散することができます。マルチAZ構成との違いは、リードレプリカへのデータ更新は非同期で行われる点です。あくまでもリードレプリカの作成目的は読み取り性能の向上であり、読み取り整合性が保証されているわけではありません。

リードレプリカオプション、マルチAZオプションともに、数クリックと数分で導入することが可能です。通常オンプレミスで同様の構築を行おうとすると最低でも数週間かかるかと思いますが、AWSでは上記のような仕組みを簡単に構築することが可能です。

リードレプリカの構成例



5. RDSのメンテナンスとは？

RDSは『メンテナンスウインドウ』という時間を設定することで、パッチ適用やバックアップなどのメンテナンス作業を自動で実施することができます。尚、インスタンスタイプの変更などは利用者自身で実施する必要があります。責任共有モデルを考慮して、AWS側で実施されるメンテナンスと、利用者自身で実施する必要があるメンテナンスについて、項目を把握しておきましょう。RDS側で実施されるメンテナンス項目については具体例を記載します。

①Amazon RDSにおけるバックアップ

Amazon RDSでは、非常に手間のかかるバックアップ作業についてもAWS側で自動的に実行させることが可能です。スケジュールに従いスナップショットを作成します。なお、自動で取得したバックアップの保存期間には制約（35日まで保持）があるため、要件に合致するかについては事前に確認しましょう。

②Amazon RDSにおけるパッチ適用

Amazon RDSではパッチ適用を自動で実施するため、万が一データベースエンジンの脆弱性が発覚しても自動かつ迅速に対応することが可能です。パッチ適用の種別は、『利用可能（延期が可能）』『必須（必ず行われる）』の2種類が存在します。後者については必ず実施されるため、なお、セキュリティ上緊急のパッチ適用はメンテナンスウィンドウの時間外でも実施されることがあります。

6. オンプレミスMySQLからAmazon RDS for MySQLへのデータ移行方式

Amazon RDSに移行したい場合については様々な手段が存在しますが、MySQLを例に、移行方法について3点簡単に記載します。

①MySQLレプリケーションの利用

MySQLのレプリケーション機能を利用し、ダウンタイムを最小限で移行することができます。こちらは無料ですが、MySQLのバージョンがRDS側でも利用可能か確認する必要があります。

②mysqldumpの利用

コマンドにてデータをエクスポートし、RDSにインポートする移行方式です。こちらも無料ですが、適宜踏み台サーバーなどを構築する必要があります。加えて手動で実施する必要があり、工数がかかります。

③DMS（AWS Database Migration Service）の利用

AWSが提供している機能を利用する形式で、こちらもダウンタイムを最小限で移行することができます。非常に多機能で、安全な移行を実現できますが、有償です。

第三章 AWS Auroraとは？その特徴は？

1. Amazon Auroraとは？

Amazon Auroraとは、AWS(Amazon Web Services)が開発した、MySQLおよびPostgreSQLと互換性のあるクラウド向けのリレーショナルデータベースのことです。従来、MySQLやPostgreSQLはオープンソースであり安価ではあるものの、拡張性や性能を十分なものにするには高度なスキルが必要であり、大規模利用向けではないと言われていました。Amazon Auroraはそのような弱点を克服するような形で性能面や拡張に関する機能を充実させ、従来のエンタープライズデータベースのパフォーマンスを出すことが可能になっています。加えて、MySQLとPostgreSQLなどの特徴であったシンプルさやコスト効率も維持したまま利用することが可能になっています。開発したAWSによると、MySQLと互換性のあるバージョンと、PostgreSQLと互換性のあるバージョンではそれぞれMySQLの最大5倍、PostgreSQLの最大3倍高速であるとされています。

① (Auroraではない) RDSとAuroraの違い

Aurora以外のRDSを、Auroraと比較したメリットは下記の通りです。

- コストが安い

Auroraと比較してRDSはコストを安く抑えながら、マネージドデータベースとしてのメリットを享受することができます。したがって、パブリッククラウドを利用し始めた段階でメンバーのスキルレベルが十分ではなかったり、性能面に不満がなく単純に運用コストを減らしたい場合はRDSを選択するといいいでしょう。

- 幅広いデータベースエンジンが利用可能であり、OracleやMicrosoft SQL Serverなどの商用データベースエンジンも利用可能
OracleやMicrosoft SQL Serverなどのデータベースエンジンを使いたい場合はAuroraは使用できません。また、MySQLやPostgreSQLにおいても、利用できるバージョンに差が存在します。例えばAuroraで利用可能なPostgreSQLのメジャーバージョンは9.6、10、11、12、13となります。いっぽうで、RDSの場合は14も利用可能です（※出典：https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/PostgreSQLReleaseNotes/postgresql-versions.html）。したがって、使用しているデータベースエンジンの種別や、バージョンにおいてAuroraで提供がない場合はRDSを使用することになります。

- MySQLに限りストレージエンジンが選択できない

これはMySQLの場合のみの話ですが、現在AuroraがサポートしているMySQLのストレージエンジンはInnoDBのみとなっています。MyISAMをどうしても利用したい場合は、RDSを選択することになります。

2. Amazon Auroraのさまざまな特性

Amazon AuroraにはRDSや通常のマネージドデータベースとは異なる様々な特性があります。どのような特性があるのか、具体的に見ていきましょう。

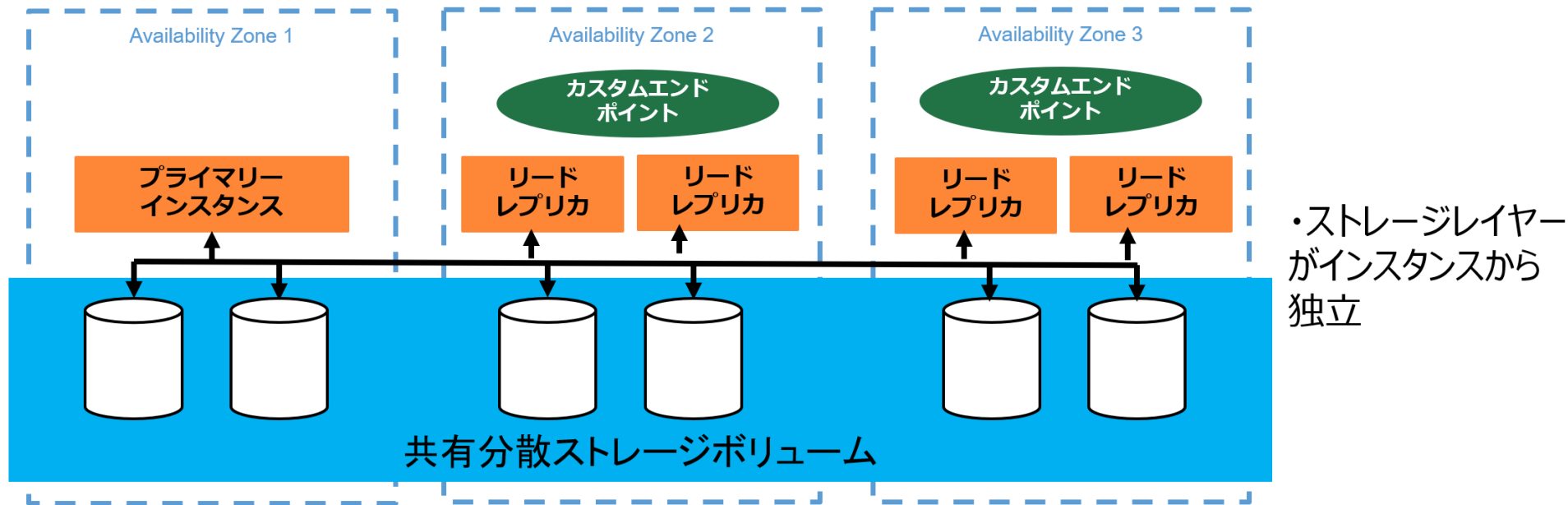
①Auroraのアーキテクチャについて

まずは、Auroraのアーキテクチャについてです。Auroraはクラスタという単位で構成されています。クラスタの要素は1つ以上のDBサーバーで構成されるインスタンス層とデータを管理するストレージ層の2層で成り立っています。つまり、SQLなどの処理を行うインスタンスとデータを保存するストレージが分離しています。また、データを保管するストレージは、1AZあたり2箇所、3AZに渡りコピーされています。このような構成にすることで、仮にネットワーク障害、ディスク破損などの問題が起こっていたとしても、データベースの稼働に影響がないようにし、可用性の向上に貢献しています。

②Auroraの特異性：ストレージについて

ストレージ自体にも特異性があります。AuroraのストレージはProtection Groupsという10GBごとのグループに分かれて処理がされます。そして、各Protection Groupに対して並列に同時に処理を行なえるようになっています。これによって得られるメリットとしては、バックアップやリストアといった処理が並列処理により実施されることで、高速に行うことが可能になる、という点です。

Auroraのアーキテクチャー



- ・複数のAZに最大 15 の昇格可能なリードレプリカ
- ・レプリカが低遅延
- ・フェイルオーバーの順序設定が可能
- ・カスタムリーダーポイントによるレプリカの使い分け

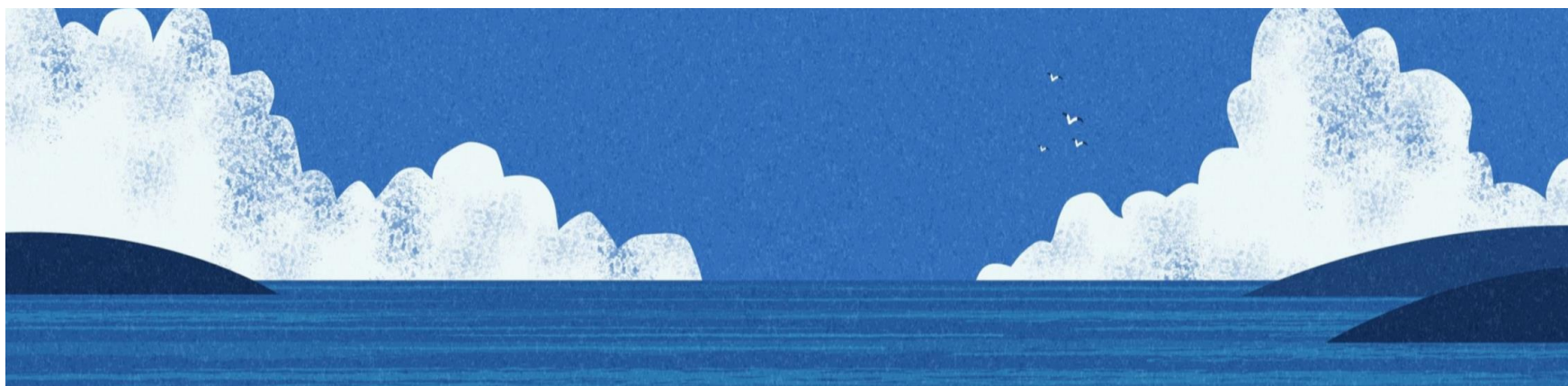
- ・データは3つのAZに渡って6個レプリケーション (1AZで2つ) され、かつS3にバックアップすることで99.99%を超える高可用性、耐障害性

③Auroraの特異性：バックトラック

より高速にデータベースの状態をリカバリする機能として搭載されているのが、バックトラックです。正確には、「データを巻き戻す/進める」ことができる機能です。例えばデータベースの操作中に誤ったSQLを実行してしまい、テーブルのデータを破壊してしまった場合でも巻き戻すことが可能です。また、データを巻き戻す/進めることが可能なので、テーブルのデータを破壊してしまった正確な時点を特定して作業を再開することができます。通常のデータベースは特定の時点にデータを戻すことしかできないため、1時間30分前にテーブルを誤操作してしまったのに、2時間分戻してしまうと30分間の更新が失われてしまうことがあります。この機能を使うことでデータの更新を極力失わずに操作を行うことができます。

④Auroraの特異性：クラスタキャッシュ管理

プライマリインスタンスが持っているキャッシュを、特定のリードレプリカと共有することができる機能です。仮にデータベースの障害が発生しフェールオーバーが行われた際に、キャッシュがフェールオーバーされないで、一時的にDBの処理能力が下がってしまいます。この機能を使うことでフェールオーバーが発生してもキャッシュをそのまま利用することが可能になるため、DBの処理能力低下を抑えることが可能です。



⑤Auroraの特異性：並列クエリ

これは、SQLなどの処理を行うインスタンスとデータを保存するストレージが分離しているアーキテクチャによって実現されている機能です。インスタンス層がやっているクエリ実行などの処理をストレージ層のリソースにオフロードして実行させる機能が並列クエリです。このことによってクエリ実行処理が非常に高速化します。

⑥Auroraの特異性：Global Database

複数のAWSリージョンにまたがり高速でレプリケーションとフェイルオーバーが可能になるオプションです。1つのプライマリAWSリージョンと、最大5つのセカンダリAWSリージョンで構築することが出来ます。Global DatabaseのRTO/RPOはそれぞれ、RTOが1分未満、RPOが5秒未満となっており、非常に高速な障害からの回復が可能になっています。世界中で展開されているアプリケーションのデータベースとしての利用や、リージョン障害に備えた災害対策環境としての利用が考えられます。

⑦Auroraの特異性：マルチマスター

従来のシングルマスター構成とは異なり、複数のプライマリDBを使用可能になります。この機能によって、片方のプライマリDBに障害が発生した場合でも、もう片方のプライマリDBで継続的な書き込みができます。データベース障害が発生した際のフェールオーバー中にも書き込み処理が可能になるため、より可用性が求められるシステムにおいても利用が可能になります。

Auroraの特異性まとめ

- ・ ストレージのインスタンスからの独立とProtection Groups
- ・ バックトラックによる時間指定の巻き戻し
- ・ クラスターキャッシュによる高速フェールオーバー
- ・ 並列クエリーによる高速化
- ・ Global Databaseによるリージョン障害対策
- ・ マルチマスターによる可用性

第四章 AWS Aurora Serverlessを最新機能(v2)を含めて解説

1. Amazon AuroraとAurora Serverless

AWS (Amazon Web Services) で提供されているマネージドデータベースの種類の1つに、Aurora Serverlessというものがあります。Aurora Serverlessは2022年4月にバージョンアップされAurora Serverless v2として提供が始まっています。

(出典：<https://aws.amazon.com/jp/blogs/news/amazon-aurora-serverless-v2-is-generally-available-instant-scaling-for-demanding-workloads/>)、

本記事では、Aurora Serverlessがどのようなサービスなのか、また、V2へのバージョンアップによりどのような機能が追加になったのかを、記載していきます。

①Amazon AuroraとAurora Serverlessの違いとは？

性能面と運用面で大きな違いがあります。性能面では、Amazon AuroraはEC2やコンテナなどのサービスと通信する際にクラスターエンドポイントと呼ばれるエンドポイント経由である一方、Aurora ServerlessはVPCエンドポイント経由で通信が行われます。簡単に言えば、Aurora Serverlessのほうが少々回り道をする通信経路となっており、Amazon Auroraのほうが、若干性能が高いです。運用面の違いは、「インスタンスの管理が必要かどうか」という点になります。Amazon Auroraはユーザー側でインスタンス管理が必要になりますが、Aurora Serverlessは完全にAWS側でインスタンスが管理されるため、運用コストが若干低くなります。また、サービスの利用料も、Amazon Auroraはインスタンスタイプによる課金であるのに対し、Aurora Serverlessでは秒単位の利用数や、DBへのリクエスト数がメインとなり、Aurora Serverlessのほうが（使い方にもよりますが）大抵の場合コスト効率がよく、低い料金で利用できます。

② (v2になる前の) Aurora Serverlessの特徴

Aurora Serverlessの大きな特徴は下記の通りです。

- アーキテクチャ
- API接続
- 自動スケーリング
- 課金体系

それぞれについて、詳細に見ていきます。

- ・アーキテクチャ

Aurora ServerlessはS3などと同様にAWSによりサーバーが管理されているため、VPCエンドポイント経由で通信が行われます。そうすることにより、Aurora側でインスタンスの増減が行われていても、AWS側で通信の制御を行い、通信が安定して行えるようになっています。

- ・API接続

Aurora ServerlessはAPIを使って他のサーバレスサービスとの連携ができます。APIを使うメリットとしては、APIの通信はHTTPSで行われ、DBとの直接通信を行わなくて済む、という点です。従来のRDBMSであれば、利用者が多いときに設定したコネクション数の上限に達すると通信ができなくなってしまう、というケースに遭遇することがあります。HTTPSであればセッション管理の必要性はあるにせよ、コネクション数の上限といったボトルネックはなく、アプリケーションの拡張性に一層柔軟に対応可能です。

- ・自動スケーリング

Auroraはストレージが10GB～128TBまで自動的に拡張されます。したがって、大抵の場合なストレージ管理を気にすることなく利用を継続することが可能になっています。

- ・課金体系

Aurora Serverlessではスケーリングした性能を表す単位をACU(Aurora capacity unit)+DBに対するリクエスト数+データ量となっています。1ACUは約2GBのRAMとそれに相応するCPU、ネットワークで構成されています。ACUについては、0から256ACUまでスケール変換ができますし、最大～最小を事前に設定しておき、スケールする範囲を調整することも可能です。DBの使用がなく、ACUが0の場合には課金が発生しないため、DBインスタンスを完全中止しなければ課金され続けるAmazon Auroraとは違いコスト効率に優れています。

③ (v2になる前の) Aurora Serverless導入時の問題点やデメリット

Aurora Serverlessはほかのサーバーレスサービスと同様、様々な制限や制約があり、導入前には考慮に入れる必要性があります。具体的な問題点やデメリットについて記載します。

- バックアップの制約

バックアップ期間を変更することはできず、特殊な要件で長期のバックアップを擁するシステムには利用できません。

- スケールが遅い

Aurora Serverlessはクエリが処理されている間はスケーリングできません。つまり継続的にクエリが実行されるようなアプリケーションではスケーリングが実行できず、想定以上の処理性能が出ない可能性があります。

- シングルAZのみ対応、別AZへのフェールオーバーが遅い

処理は基本的にシングルAZで行われます。障害が発生した際には別のAZに自動的にフェールオーバーするのですが、プロビジョニングされたAuroraよりも時間がかかります。したがって、Amazon Auroraと比較して、RTOが遅いというデメリットがあります。

- Data APIの制限

1秒間あたり最大1000コールまでとなっており、十分な処理性能を持っていますがそれ以上の性能を求められる場合は考慮が必要です。また、レスポンスのサイズも最大1MBとなっています。

2. Aurora Serverlessはv2で何が変わったのか？

Aurora Serverless v2は先述のデメリット・問題点のうち、『2. スケールが遅い』『3. フェールオーバーが遅い』が改善されています。まず、クエリが実行中でも瞬時にスケールすることが可能になっており、性能上のボトルネックが改善されています。また、従来はシングルAZだったのですがマルチAZ対応も可能になっています。それにより、グローバルデータベース、リードレプリカなどのAmazon Auroraの既存機能をAurora Serverlessでも利用可能になっており、可用性やRTOの大幅な改善が行われます。コストの効率性はそのまま継続して特徴として残っているので、性能を大幅に向上しつつ、コスト効率が良いAuroraとして利用することができます。

RDS、Aurora、Aurora Serverless、同v2の対比

	RDS	Aurora	Aurora Serverless	同v2
コスト	○	△ インスタンス数による	? ワークロードに大きく依存	? ワークロードに大きく依存
性能	○	◎	△ スケールアップが遅い等	◎ スケールアップが高速化 クエリー実行中もスケール可
可用性	○	◎	△ シングルAZのみ対応 他AZへの切り替わりが遅い	◎ マルチAZに対応 フェールオーバーが高速化
メンテナンス性	○	◎	◎	◎

3. 結局、Aurora Serverlessはどんなところで使えるのか？

Aurora Serverlessの使用ケースは下記が考えられます。

- 非常に変化の大きいワークロードの場合
非常にアクセスを多い時と少ない時の落差が大きかったりの予測できないワークロードに対して、データベースは負荷のニーズに合わせて容量を自動的にスケーリングできます。
- システム開発時のAmazon Auroraの代替
オンプレミス上のシステムをリファクタリングして改修したりする際に、Amazon Auroraを利用するとコスト効率が悪い場合があります。そのような場合には開発中の段階では、コスト効率に優れているAurora Serverlessを利用すると開発コスト削減につながる可以考虑されます。

Aurora Serverlessは使い方次第

- ・ 高負荷時が非常に限定的なケース
- ・ 利用時間が（一日のうちで）少ないケース
→RDSやAuroraではインスタンスが動作している限り課金されるのもつたいない
- ・ 開発時で、本番ではAuroraを使う予定の場合
- ・ 需要予測がたてられないような新しい用途
→いずれは、RDSやAuroraを使う、そのための代替の場合



第五章 Auroraか(非Auroraな) RDSか。比較した上でどちらを使う？

1. (非Auroraな) RDSのメリット

一般的に「データベース」というと、サーバーにミドルウェアとしてインストールし、稼働させるケースが考えられますが、Amazon RDSは、データベースをあらかじめセットアップされた状態で提供しており、パッチ適用やバックアップもAWSがユーザーの代わりに運営（マネージ）してくれます。このようなクラウド提供者によりある程度運用管理が代行されているサービスをマネージドサービスといいます。マネージドサービスとしてのデータベースのメリットとは、『データベースの難しい運用をAWSに所属する凄腕エンジニアに委任できる』ということにつきます。一般的にデータベースの運用は、データのバックアップや容量の拡張・予測、レプリケーションの設定・運用など、様々なコストがかかります。マネージドサービスを使うことによって、このようなデータベースの運用において困難な部分に対してコストをかけること少なくすることができだけでなく、データベースの安定稼働を実現することができます。また、運用に割いていた人的リソースやコストを他に注力させることができるので、経営を効率化することができます。

2. (非Auroraな) RDSの具体的な機能

先述のメリットを実現する具体的な機能としては、下記の2点があります。下記のような設定を、1クリックで導入することができます。

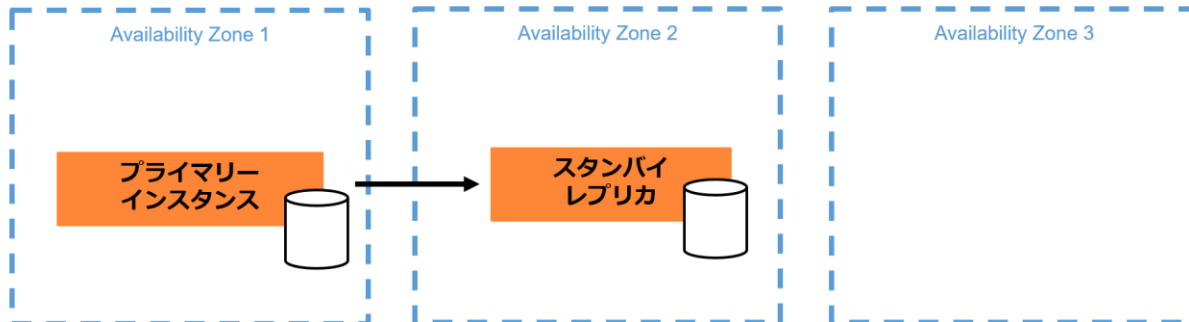
- 可用性とスケーラビリティも高める新「マルチAZ配置オプション」

別のAZにスタンバイデータベースを構成することができます。マルチAZ構成を用いることによって一つのAZで障害が発生しても他の正常なAZを使用して稼働を継続させることができ、データベースの可用性を向上させることができます。データは同期レプリケーションです。

- データの読み取り性能を向上させる「リードレプリカオプション」

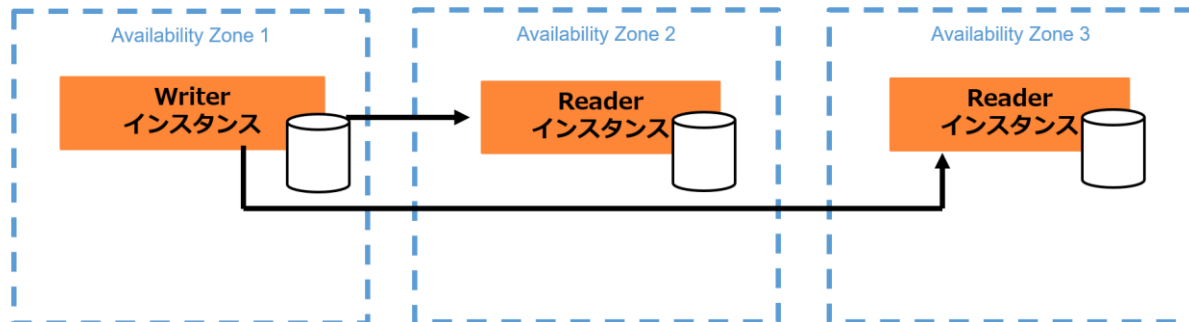
リードレプリカは読み取り専用のDBのことです。アクセス数が多くDBに読み取りが負荷が起きている場合などは、リードレプリカを構築することでプライマリデータベースへの負荷を分散することができます。スタンバイと異なり、データは非同期でレプリケーションされます。

RDSのマルチ AZ DB インスタンス（以前からの構成）

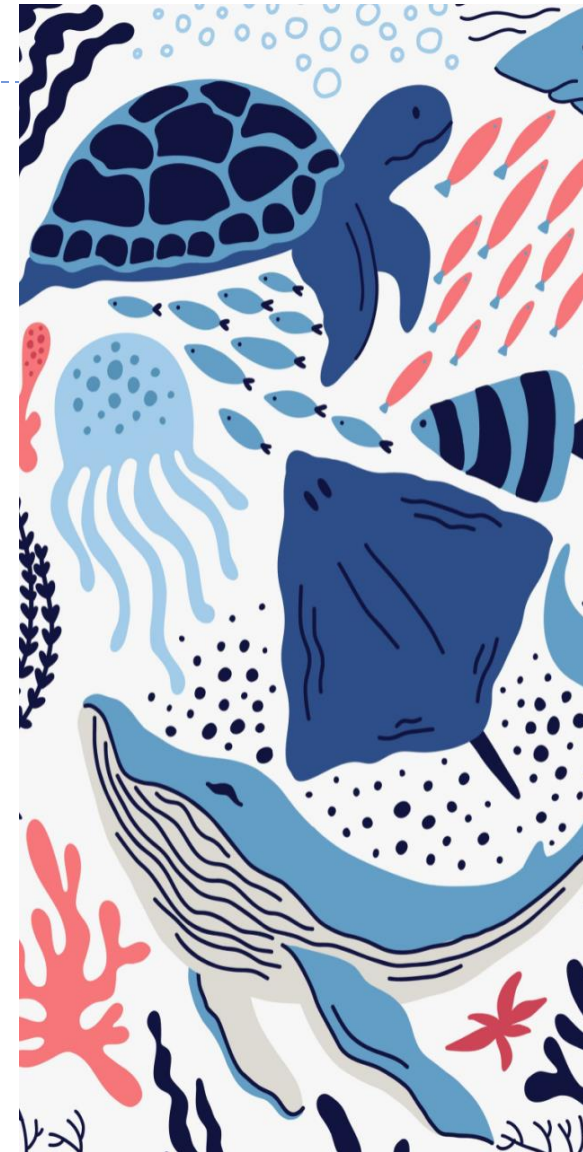


- ・スタンバイレプリカは待機のみで、リードできない（別途リードレプリカは作成可能）
- ・レプリケーションは同期
- ・フェイルオーバーが若干遅い（通常 60～120 秒）

RDSのマルチ AZ DB クラスター（2022/3の新機能）



- ・2台のスタンバイインスタンスへ読み取りアクセスが可能。追加のリードレプリカは構成できない
- ・レプリケーションは準同期で書き込みレイテンシーが低減（最大2倍）
- ・フェールオーバーが高速化（通常 35 秒未満。）



3. Auroraのアーキテクチャー(RDSとの違い)

AuroraとAuroraでないRDSの大きな違いは、『クラスタ』と『ストレージ』にあります。まずは、『クラスタ』についてです。Auroraはクラスタという単位で構成されています。クラスタとは、Auroraのデータベースを作成した際に作られるコンポーネント全体、つまりデータベースの管理単位と考えてください。次に『ストレージ』についてです。クラスタの要素は1つ以上のDBサーバーで構成されるインスタンス層とデータを管理するストレージ層の2層で成り立っています。つまり、SQL解析などの処理を行うインスタンスとデータを保存するストレージが分離しています。また、データを保管するストレージは、1AZあたり2箇所、3AZに渡り、合計6箇所にコピーされています。このような構成にすることで、仮にネットワーク障害、ディスク破損などの問題が起こっていたとしても、データベースの稼働に影響がないようにし、可用性の向上に貢献しています。

4. Auroraの特徴

Auroraは、主に下記のような特徴を持っており、AuroraでないRDSと比較して高性能になっています。

- 豊富なオプション

従来のシングルマスター構成とは異なり、複数のプライマリDBを使用可能なマルチマスタークラスター、複数のAWSリージョンにまたがり高速でレプリケーションとフェイルオーバーが可能になるグローバルデータベースといった、様々なオプションが用意されており、RDSよりも高可用性が実現できるオプションが用意されています。

- 高速化の仕組み

プライマリインスタンスが持っているキャッシュを、他のインスタンスと共有することができるクラスタキャッシュという機能があります。これにより、障害でフェールオーバーが発生した先でもキャッシュを共有することができ、フェールオーバー先でも高速なデータベースの処理を継続することが可能です。

5. Auroraの料金体系と(非Auroraな) RDSの料金体系

AuroraとRDSについて、料金体系について比較をしてみます。それぞれ、従量課金であることには変わらないのですが、従量課金の対象となる項目が異なっています。

- ・ RDS…DBインスタンスの起動時間+ストレージの容量に応じた料金
- ・ Aurora…DBインスタンスの起動時間+ストレージの容量に応じた料金+I/Oリクエスト数

このように、Auroraの料金ではI/Oのリクエストによる課金も発生します。東京リージョンでは、100万件のリクエストあたり0.24ドル（2022年5月）となっていますが、データベースは読み書きの頻度が多く発生するという特徴もあり、I/Oリクエスト部分は無視できない課金項目となります。また、インスタンスのタイプによる料金もAuroraのほうが高くなっています。

6. RDSの方が絶対安いのか？

料金体系を鑑みると、Auroraが割高となり、RDSの方が安くなることが多いです。ただし、要件によってはAuroraのほうが割安になる場合もあります。具体的な要件について見ていきましょう。

- データの耐久性は必要だが、データベースの可用性は必要ない場合

Auroraは、DBインスタンスが一台つまりシングルAZであっても、ストレージ部分は3AZにまたがって6つのコピーを持つようになっています。また、シングルAZでも、Auroraはインスタンスの自動復旧を行なう機能が備わっています。したがって、データの耐久性は必要だがある程度障害が許容できるような要件であれば、RDSでマルチAZ構成をとるよりもAuroraのシングルAZのほうが要件に合致しており、コスト効率も高いと考えることができます。

- 性能を出したい場合

リードレプリカを設置、かつマルチAZ構成にしたいケースを考えます。このケースでは、RDSでは最低インスタンス3台（プライマリ、スタンバイ、リードレプリカ）から始める必要がありますが、Auroraでは2台（プライマリ、リードレプリカ（スタン

バイの役割兼務)) のインスタンスで済みます。また、クラスタキャッシュ機能により、フェールオーバーが発生しても Auroraの場合は性能を落とさずに利用を継続することができます。このように、性能をある程度担保しつつ、可用性を高めた場合はAuroraのほうが向いている場合があります。

- 構築時の工数を削減したい場合

開発したAWSによると、MySQLと互換性のあるバージョンと、PostgreSQLと互換性のあるバージョンではそれぞれMySQLの最大5倍、PostgreSQLの最大3倍高速であるとされています。したがって、ある程度チューニングなどの工数を削っても一定以上の性能をだすことができます。チューニングなどの有識者がいない場合でも、新たに人を雇うコストと時間をかけるよりは、Auroraを選択するといいいでしょう。

RDSとAuroraの比較 (RDSは商用DB除く)

項目		RDS	Aurora
デプロイオプション		①シングルDBインスタンス ②マルチAZ DBインスタンス ③マルチAZ DBクラスター (2022/3新機能)	なし
ストレージ	サイズ	20GiB – 64TiB	10GiB – 128TiB
	スケーリング	手動 (オートスケーリングで自動化)	自動
	縮小	不可 (インスタンス再作成が必要)	データ削除で動的に縮小
	可用性	シングル構成の場合、AZ内で冗長化 マルチAZを指定すればマルチAZで冗長化	3つのAZに⑥つのボリュームを置いて冗長化
フェイルオーバーできるDB		スタンバイDBは可能	すべてのリードレプリカが可能
リードレプリカ	数	最大5	最大15
	フェールオーバー	マルチAZ DBインスタンス 不可 マルチAZ DBクラスター 可能	可能
	クロスリージョン	可能	可能 (MySQLのみ)
レプリケーションの同期・非同期		マルチAZ DBインスタンス スタンバイDB : 同期 レプリカ : 非同期 マルチAZ DBクラスター 可能 1台のスタンバイDB : 同期 以外は非同期	4個は同期、残りは非同期

第六章 AWSのNoSQLの代表、Amazon DynamoDBとは

1. Amazon DynamoDBとは？

Amazon DynamoDBは、Amazon Web Services (AWS) が開発したマネージドデータベースサービスの1つです。NoSQLというタイプのデータベースであり、通常のデータベースとは大きな違いがあります。以下では、NoSQLの特徴や、DynamoDBの使いどころについて、記載してきます。

2. NoSQLデータベースとリレーショナルデータベースの比較

NoSQLと通常のデータベースはどのような違いがあるのでしょうか。それぞれのデータベースの形式の特徴について記載していきます。

①リレーショナルデータベースとは？

リレーショナルデータベース（以下RDBMS、Relational DataBase Management System）とは、属性を列、組を行とする表（テーブル）の形でデータを格納するデータベースです。エクセルのような、行と列にデータを格納しながら、格納したデータに対して様々な操作が可能なデータベースであるとイメージするといいいでしょう。RDBMSではSQLと呼ばれる言語を用いてデータの参照や加工をおこないます。RDBMSは不正なデータの記録を拒否するなどしてデータベースの整合性を保つ機能や、権限のない利用者による不正な読み出しや改竄などからデータを保護する仕組みを持っています。また、関連する複数の処理を一体化して矛盾なく実行する機能（トランザクション）もあり、データの整合性に強みを持っています。一方で、SQLによる読み書きに時間がかかってしまうデメリットがあり、非常に高速にデータを読み書きする必要があるアプリケーションには不向きです。

②NoSQLとは？

NoSQLはRDBMSではないデータベース全般を指します。特にデータの格納・読み出しを大量かつ高速に行う作業において、RDBMSではデメリットが出てしまうケースを解決する為に開発されてきました。そのため、キーバリューストア型データベースなど様々な種類があり、要件に応じて選択することが可能です。

RDBMSと比較してデータの正確性（ACID特性といいます。ACIDはAtomicity（不可分性）、Consistency（一貫性）、Isolation（独立性）、Durability（永続性）の頭文字をつなぎ合わせたものです。）には多少の弱みがあることが多いですが、データの格納形式がRDBMSよりも複雑ではないため追加処理やデータ参照が非常に多いシステムに向いています。NoSQLは代表的なもので下記の種類があります。

NoSQLで利用できるデータモデルの例として、以下にいくつか挙げます。

- ・キーバリュー型：キーとバリューのみのシンプルな組み合わせのモデル
- ・カラム型：列方向にまとめてデータを扱うことモデル
- ・ドキュメント型：JSONやXML形式で記述されたドキュメントの形で管理するモデル
- ・グラフ型：データとデータ間のつながりを管理するモデル

RDBMSとNoSQL

ポイント	RDBMS	NoSQLデータベース
データの格納形式	行と列で構成されるテーブルのみ	キー値やグラフなどの様々なデータを格納する形式がある
データの一貫性	高い	低い
パフォーマンス	NoSQLデータベースに比べると遅い	データの一貫性と引き換えに高速なパフォーマンスを実現
スケールアウト	大変	容易
データ操作の手段	SQL	オブジェクトベースの API

3. Amazon DynamoDBは何を目的として生まれたのか？

Amazon DynamoDBは、Werner Vogels（AmazonのCTO）により開発されました。分散システム（複数のコンピュータ・サーバーにより構成されるシステム）においては、CAP定理という、「一貫性（Consistency）」「可用性（Availability）」「ネットワーク分断耐性（Partition-tolerance）」をすべて達成することはできない、という決まりがあります。このなかで、「可用性（Availability）」「ネットワーク分断耐性（Partition-tolerance）」を重視した結果開発されたのがDynamoDBです。

AmazonはECが主要なビジネス産業の一つですが、DynamoDBを使うことによって、ECのカートにおいて、“いつでもカートに追加できる＝可用性”、“カートに入れたデータを失わない＝ネットワーク分断耐性”を、実現しています。

4. DynamoDBの特徴

DynamoDBの特徴は以下の通りです。

- ・ **フルマネージドサービスであり、自動的に容量の拡張やレプリケーションが実施される**

データベースの運用はほかのミドルウェアと比べて、格納されるデータ容量の予測やバックアップといった運用が非常に難しく、運用コストがかさんでしまう原因となっています。DynamoDBは自動で容量の拡張やレプリケーションが行われるマネージドサービスとして提供されており、運用負荷の軽減が可能です。また、レプリケーションや自動バックアップが、「ネットワーク分断耐性（Partition-tolerance）」の向上につながっています。

- ・ **「値」とそれを取得するための「キー」だけを格納するというシンプルな機能を持った「キーバリュー型」の形式をもつ**

NoSQLで、高速なデータの格納と取得に特化されている

データの格納形式をシンプルに行うことで、データの読み書きを高速に行うことができるようになっています。これにより、データの読み書きの際にDynamoDBはサーバー側の負荷を上げることがないため、障害の低減や高速なデータの読み書きが可能になっています。一方で、表の結合など複雑なデータのクエリは不得意、という特徴もあります。

- ・ **1桁ミリ秒単位のレイテンシーを要求するアプリケーションにも対応**

DynamoDBは読み書きの性能がRDBMSと異なり非常に高速に実施することが可能になっています。そのため、レイテンシーに厳格なアプリケーションにおけるデータベースとして利用することが可能です。

得意なこと	キーに対するバリューのCRUD操作 簡易なクエリやオーダー 大規模なデータの保存（容量制限なし） データの信頼性（3つのAZに保存）
苦手なこと	テーブルのジョイン トランザクション/コミット/ロールバック 柔軟な検索 集計処理

上記の特性から、DynamoDBでは下記のような特徴を持つシステム・アプリケーションのデータベースとして利用することがいいでしょう。

- **ミリ秒単位のアクセスレイテンシーが求められる**
- **RDBMSのような複雑なデータ構造は扱わない**
- **データ容量の予測がしにくい**

具体的には、モバイルゲームにおけるデータ収集やアカウント管理、Web広告におけるクリックストリーミングログの格納、IoTのようなセンサー機器とデータを管理するシステムなどでの利用が適しているといえます。詳細は Amazon DynamoDB: ゲームのユースケースと設計パターン (<https://aws.amazon.com/jp/blogs/news/amazon-dynamodb-gaming-use-cases-and-design-patterns/>) を確認するといいいでしょう。

5. DynamoDBの設計上の考慮点

DynamoDBでアプリケーションを構成しようとする場合、最も重要な考慮点がキーの設計です。DynamoDBでは、テーブルのデータをパーティションと呼ばれる領域に分けて保持します。データがどのパーティションへ配置されるかは”パーティションキー”によって決定されます。また、パーティションにデータを格納する場合は”ソートキー”として指定した順序でデータを格納します。データが格納されすぎているパーティションがある場合（そのようなパーティションをホットパーティションといいます）、データのクエリ速度が遅くなる場合があります。また、データの検索手段として、フルスキャンと、パーティションキーを指定した上でのソートキーによる範囲を指定したスキャンの方法があり、ソートキーを指定している場合は後者の検索が可能となり、検索速度が向上します。また、Amazon DynamoDBはインデックスとして、LSI（Local Secondary Index）とGSI（Global Secondary Index）の2種類が利用できます。LSIは、Partition Keyが同一なアイテムを、ほかのアイテムからの検索するために利用します。いずれにせよDynamoDBの設計は、パーティションキーおよびソートキーその他のキー設計が非常に重要な要素になります。データ構造だけ定義しクエリーを柔軟に発行できたRDBMSと違い、DynamoDBはクエリーが柔軟ではありません。また、データ集計にも強くありません。そのため、設計段階で各種データへどのようなアクセスがされるかをきちんと考えておく必要があります。また、システム全体をDynamoDBだけで無理やり実現しようとせずに、DynamoDBに向いている機能に限定して採用しようとする考え方が大切です。



DynamoDBのキーとインデックス

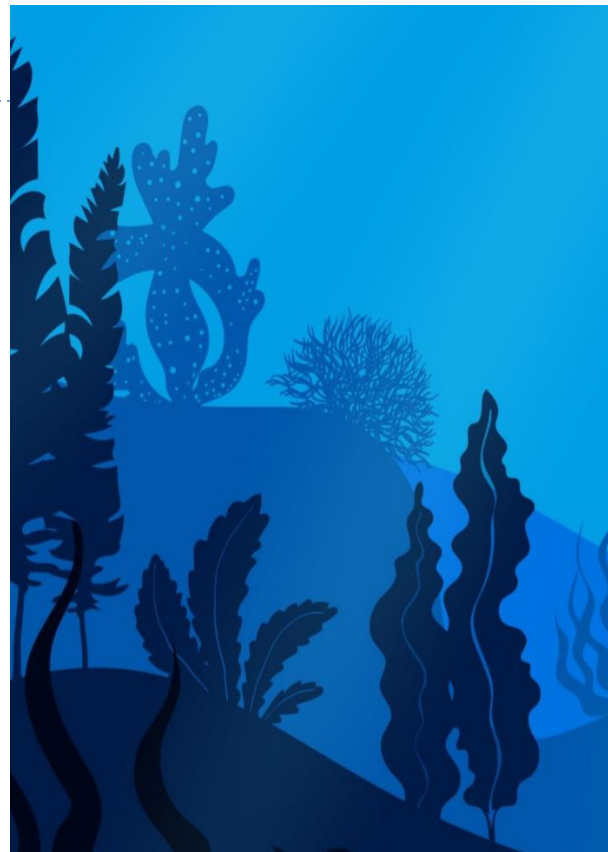
パーティションキー	テーブルの Item をパーティションと呼ばれる領域別に保存される。Item がどのパーティションへ配置されるかはパーティションキー（ハッシュ関数を実行し、ハッシュ文字列を生成）によって決定されます。各パーティションへのアクセスがなるべく均一になるようパーティションキーを設計する。
ソートキー	ソートキー が設定されている場合、データはパーティション内でソートキーを元に並べ替えられて物理的に近くに配置されます。
プライマリキー	「パーティションキー」または「パーティションキーとソートキーの複合キー」のこと。プライマリキーによってデータは一意に識別される。
グローバルセカンダリインデックス（GSI）	あるテーブルをベースに、異なるパーティションキー・ソートキーのテーブルを作成する仕組み
ローカルセカンダリインデックス（LSI）	あるテーブルをベースに、パーティションキーはそのままだに、異なるソートキーのテーブルを作成する仕組み

5. DynamoDBの料金体系

DynamoDBの料金を決める要素は大きく3つです。

- ・キャパシティーユニット
- ・ストレージ容量
- ・データ転送量

このなかで、DynamoDB特有のキャパシティーユニットについて解説します。キャパシティーユニットとは、事前に設定しておく読み込み/書き込み容量を指します。1キャパシティーユニットは1秒間に決められた回数の読み込み/書き込み容量を提供します。書き込みはWCU、読み込みRCUと呼ばれており、1WCUは1秒間に1回の書き込み、1RCUは1RCUは1秒間に2回の読み込み、または結果整合性のある1回の読み込みが可能です。このキャパシティーユニットを設定した数量に応じて課金が生じます。また、1キャパシティーユニットには書き込みが1KB、読み込みが4KBまでと制限があることも考慮しておく必要があります。



だれでもわかるスタイルズのオンラインウェビナーのご案内

スタイルズでは、AWSセキュリティに関する、30分で誰でもわかるオンラインウェビナーを開催しています。

このホワイトペーパーの内容に即して

- ・ 30分でわかる！ AWS基本的なセキュリティ
- ・ 30分でわかる！ AWS セキュリティグループ設定
- ・ 30分でわかる！ AWSにおけるDDoS対策
- ・ 30分でわかる！ AWS WAF設定
- ・ 30分でわかる！ Amazon Inspectorで脆弱性診断
- ・ 30分でわかる！ Amazon GuardDutyによるセキュリティ脅威検知
- ・ 30分でわかる！ AWS Security Hubによるセキュリティチェック

のウェビナーがいつでも無料でご覧いただけます。

以下よりお申し込みください。

<https://www.stylez.co.jp/webinar/>



はじめてでも
よくわかる

一回30分だけで
AWSが学べる！
スタイルズの
AWS無料ウェビナー

- ・ AWSセキュリティ
- ・ AWSシステム監視
- ・ その他

をテーマにした多くの
過去動画をいつでも
視聴できます。

▶ スタイルズ (Stylez) はAWSパートナーです

スタイルズのAWS関連サービス

AWS導入・移行サービス

・自社の基幹システムやWebサービスをクラウドへ移行したいがどのように進めていけばいいのか悩まれているお客様に向けたベストプラクティスをご提供します。

AWSセキュリティサービス

・データ漏洩、不正アクセス、マルウェア感染、DDoS攻撃、改ざんなどのセキュリティインシデント対策から、可視化、自動復旧対処の仕組み開発まで、AWSのセキュリティ対策をすべてご影響可能です。

AWS上のサーバーレス開発

・ECSを利用したコンテナ構成やサーバーレス構成により、オートスケール（自動拡張）やオートヒーリング（自動復旧）など、クラウド利用のメリットを最大限に活用したシステムをご提案します。

AWS監視、運用・保守サービス

・AWS、Zabbixのパートナー企業として多くのお客様に監視サービスを提供させていただいた実績と長年培ったノウハウをベースにお客様環境に合わせた最適な監視運用をご提案します。

AWSコンテナ構築運用サービス

・Amazon ECS（AWSが提供するコンテナサービス）をベースにお客様のシステムをコンテナ化して運用するまでのご支援をさせていただくソリューションです。

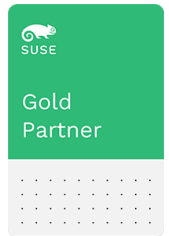
AWS DevOps導入サービス

・インフラのコード化、さらにソースコードのビルドやテスト、および、アプリケーションのデリバリーまで、開発工程全体を自動化するCI/CDの仕組みの構築をAWSの環境上での実現をサポートします。

お問合せ、ご相談は株式会社スタイルズ <https://www.stylez.co.jp> まで



partner
network



実績豊富なエンジニア集団の技術と開発ツールで
「開発期間/コスト削減」「品質向上」を実現

株式会社スタイルズ

<https://www.stylez.co.jp>

東京都千代田区神田小川町1-2 風雲堂ビル6階

AWSパートナーの一員として、CloudShift（クラウドシフト）というブランドにて、AWS導入の支援を提供しております。

