



# クラウド基礎

RDS・DynamoDB

## 目次

AWS のリレーショナルデータベース .....	2
RDS .....	3
ハンズオン： Oracle DB インスタンスの構築.....	7
AWS の NoSQL データベース .....	21
Dynamo DB .....	22
ハンズオン： NoSQL テーブルでのクエリの実行.....	25

# AWS のリレーショナルデータベース

---

# RDS

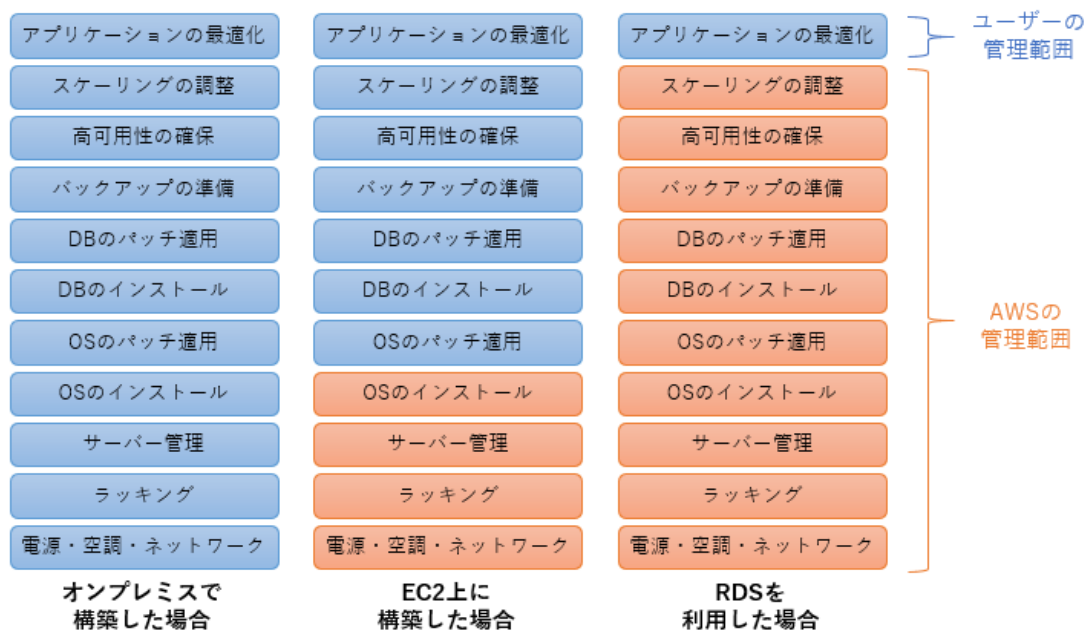
## 1. RDS の概要

Amazon Relational Database Service (RDS) は、AWS が提供するデータベースサービスの中でもリレーショナルデータベースを簡単に利用することができるサービスです。



一般的にリレーショナルデータベースは、データを複数のテーブルに保存し、それぞれのテーブルの関係性を管理することで全体のデータを管理します。データの書き込みを行う時は、処理結果の整合性が取れるように処理のまとまりと順番(トランザクション)を厳密に管理し、処理を記述するために SQL を用いることも特徴です。

RDS を使用すると、ハードウェアのプロビジョニング、データベースのセットアップ、OS のパッチ適用、データベースエンジンのバージョンアップやバックアップ、高可用性の確保、スペック変更時のチューニングといった時間がかかる管理タスクを AWS に任せることができるため開発者はアプリケーションの開発に集中することができます。



## 2. RDS の特徴

RDS の主な特徴には次のようなものがあります。

- 運用・管理の容易さ

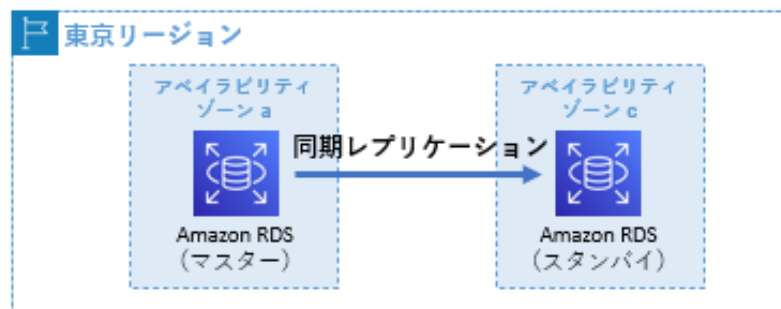
RDS は AWS が管理を行うマネージドサービスであるため、ユーザーは OS やデータベースエンジンのパッチ更新などを行わなくても常に最新の状態でデータベースを利用することができます。OS のメンテナンスは週に 1 回指定した時間に自動で行われ、データベースエンジンのマイナーアップグレードは自動適用と手動適用を選択することができます。

- 自動バックアップ

RDS はデフォルトで 7 日間の自動バックアップ機能を持っています。自動バックアップを設定している期間であれば、ポイントタイムリカバリの機能によって秒数まで指定をしてデータベースを復元することができるため、故障があった直前の状態にリカバリーすることが可能です。

- 高可用性

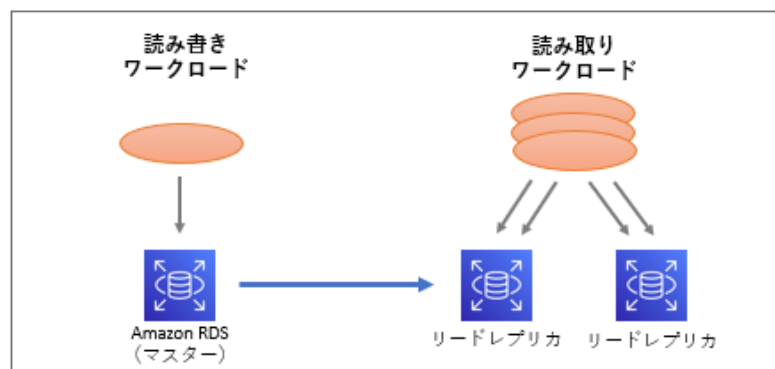
RDS はマルチ AZ 配置を設定することができ、設定をオンにしておくとアベイラビリティゾーンをまたいだレプリケーションを作成し、自動でデータを同期するスタンバイデータベースを作成することができます。



さらに、マルチ AZ 配置を有効にしておくと、マスターに障害が発生した場合には自動でスタンバイへの切り替え（フェイルオーバー）が行われて、スタンバイデータベースがマスターになります。

- リードレプリカの利用

RDS では、データベースエンジンとして Amazon Aurora、MySQL、PostgreSQL、MariaDB を利用した場合、リードレプリカと呼ばれる読み取り専用のデータベースを作成し、マスターデータベースの読み込み負荷を軽減させることができます。



### 3. RDS で利用可能なデータベースエンジン

RDS では以下のデータベースエンジンを利用することができます。

- MySQL
- PostgreSQL
- Oracle
- SQL Server
- MariaDB
- Amazon Aurora

Amazon Aurora は、AWS がクラウドでの利用に最適化して設計したデータベースエンジンです。MySQL 及び PostgreSQL との互換性を持っているため、MySQL もしくは PostgreSQL で動いているアプリケーションは Aurora を利用してそのまま実行することができます。

また、性能としては RDS の MySQL エンジンと比べて 5 倍の性能があり、リードレプリカを 15 個作成できるなどといった多くのメリットが存在しています。

### 4. RDS のスケーリング

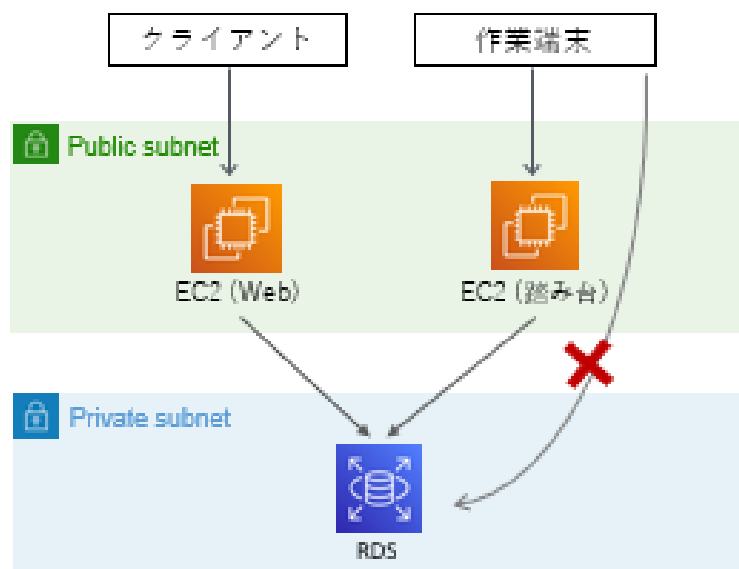
RDS がスケーリングを行う際はスケールアップ/スケールダウンによる垂直スケーリングを行います。これはリレーショナルデータベースで処理を行う際はトランザクションを厳密に管理するため、複数のサーバーに処理を分けてしまうと、サーバー間でデータの矛盾が発生してしまう恐れがあるためです。



## 5. 踏み台サーバーを利用した RDS への接続

RDS には個人情報を含む重要なデータが保管されるため、不正アクセスを許してしまった場合には大きな被害が発生してしまいます。そのため、RDS を起動するときはインターネット経由での接続ができないようにプライベートサブネット内に RDS を配置する構成を取ります。

しかし、RDS への接続が必要となる場合もあるため、そういった場合にはパブリックサブネットに踏み台サーバー（Bastion サーバー）を配置して、プライベートサブネットの RDS には踏み台サーバーを経由してアクセスすることで RDS インスタンスを保護します。



### ■ 参考

- ・ Amazon RDS (マネージドリレーショナルデータベース)  
<https://aws.amazon.com/jp/rds/>

# ハンズオン： Oracle DB インスタンスの構築

## 1. はじめに

### 1-1. ハンズオンのゴール

- Amazon RDS を利用して Oracle DB インスタンスを作成して接続を行う

### 1-2. 事前準備

- 安心サンドボックスを利用して AWS マネジメントコンソールへサインインしていること
- Oracle SQL Developer がインストールされていること

### 1-3. ハンズオンのステップ

ステップ 1, Oracle DB インスタンスとセキュリティグループを作成する  
ステップ 2, SQL Developer を利用して Oracle DB インスタンスへ接続する

※ 今回のハンズオンでは簡易な設定で RDS の動作を確認するために  
RDS へのパブリックアクセスを有効にしてハンズオンを実施します。

ハンズオンで入力する設定値は最低限の動作を確保するための値を採用しています。



## 2. ハンズオンの実施

### ステップ 1. Oracle DB インスタンスとセキュリティグループを作成する

RDS を利用して Oracle DB インスタンスを作成します。

1. AWS マネジメントコンソールで [RDS] を選択する。



2. RDS のダッシュボードに遷移したら [データベースの作成] をクリックします。



3. データベースの作成画面で以下のように設定を行います。

- データベース作成方法：[標準作成]
- エンジンのタイプ：[Oracle]
- エディション：[Oracle Standard Edition Two]
- ライセンス：[license-included]
- バージョン：[デフォルトで選択されているバージョン]

RDS > データベースの作成

## データベースの作成

データベース作成方法を選択 [情報](#)

☒ **標準作成**  
可用性、セキュリティ、バックアップ、メンテナンスといったすべての設定オプションを設定します。

☐ **簡単に作成**  
推奨されるベストプラクティス設定を使用します。一部の設定オプションは、データベースの作成後に変更できます。

### エンジンのオプション

エンジンのタイプ [情報](#)

☐ Amazon Aurora  


☐ MySQL  


☐ MariaDB  


☐ PostgreSQL  


☒ **Oracle**  


☐ Microsoft SQL Server  


Database management type [情報](#)

☒ **Amazon RDS**  
RDS fully manages your database, including automatic patching. Choose this option if you don't need to customize your environment.

☐ Amazon RDS Custom  
RDS manages your database and gives you privileged access to the OS. Use this option if you want to customize the database, OS, and infrastructure.

## エディション

- ☐ Oracle Enterprise Edition  
最大 16 個の vCPU をサポートする手頃な価格のフル機能データベースです。

- ☒ Oracle Standard Edition Two  
最大 16 個の vCPU をサポートする手頃な価格のフル機能データベースです。Oracle Database Standard Edition Two は、Standard Edition および Standard Edition One に代わるものです。

## ライセンス

license-included ▼

## バージョン

Oracle 19.0.0.0.ru-2022-10.rur-2022-10.r1 ▼

- テンプレート：[開発/テスト]
- DB インスタンス識別子：[OracleDBHandsOnInstance]
- マスターユーザー名：[oracleadmin]
- パスワード自動生成：[有効]
- DB インスタンスクラス：[バースト可能クラス(t クラスを含む)] [db.t3.small]

## テンプレート

お客様のユースケースに合わせてサンプルテンプレートを選択します。

- ☐ 本番稼働用  
高い可用性と、高速で安定したパフォーマンスのために、デフォルト値を使用します。

- ☒ 開発/テスト  
このインスタンスは本番稼働環境ではない開発で使われます。

## 設定

## DB インスタンス識別子 情報

DB インスタンスの名前を入力します。この名前は、AWS アカウントが現在の AWS リージョンで所有しているすべての DB インスタンスにおいて一意である必要があります。

OracleDBHandsOnInstance

DB インスタンス識別子は、大文字と小文字の区別がありませんが、すべて小文字で保存されます (例: "mydbinstance")。制約事項: 1~60 文字の英数字またはハイフン。1 文字目は文字である必要があります。連続する 2 つのハイフンを含めることはできません。ハイフンで終わることはできません。

## ▼ 認証情報の設定

## マスターユーザー名 情報

DB インスタンスのマスターユーザーのログイン ID を入力します。

oracleadmin

1~16 文字の英数字。1 文字目は文字である必要があります。

- ☒ パスワードの自動生成

Amazon RDS がパスワードを生成するか、お客様がご自身でパスワードを指定することができます。

### インスタンスの設定

以下の DB インスタンスの設定オプションは、上記で選択したエンジンでサポートされているものに制限されています。

#### DB インスタンスクラス [情報](#)

- ☐ 標準クラス (m クラスを含む)
- ☐ メモリ最適化クラス (r クラスと x クラスを含む)
- ☒ **バースト可能クラス (t クラスを含む)**

db.t3.small

2 vCPUs 2 GiB RAM ネットワーク: 2,085 Mbps

- ☒ 追加のメモリ設定を含める [情報](#)
- ☐ 以前の世代のクラスを含める

- ストレージタイプ：[汎用 (SSD)]
- ストレージ割り当て：[20]
- ストレージの自動スケーリングを有効にする：[有効]
- 最大ストレージしきい値：[1000]

## ストレージ

### ストレージタイプ 情報


汎用 (SSD)

### ストレージ割り当て

20

GiB

(最小: 20 GiB、最大: 32,768 GiB) より高い割り当て済みストレージは、IOPS のパフォーマンスを改善する場合があります。

-  100 GiB 未満の汎用 (SSD) ストレージを高スループットワークロード用にプロビジョニングした場合、初期の汎用 (SSD) IO クレジットバランスを使い切った時点で、レイテンシーが大きくなる場合があります。詳細はこちら [🔗](#)

### ストレージの自動スケーリング 情報

アプリケーションのニーズに基づいて、データベースのストレージに対する動的なスケーリングのサポートを提供します。

- ☒ ストレージの自動スケーリングを有効にする  
この機能を有効にすると、指定したしきい値を超えた場合にストレージを増やすことができます。

### 最大ストレージしきい値 情報

データベースが指定されたしきい値に自動スケールされると、料金が適用されます。

1000

GiB

最小: 21 GiB、最大: 32,768 GiB

## 可用性と耐久性

### マルチ AZ 配置 情報

- ☐ スタンバイインスタンスを作成する (本稼働環境向けに推奨)  
データの冗長性を提供し、I/O のフリーズを防ぎ、システムバックアップの間のレイテンシーの急上昇を最小限に抑えるために、別のアベイラビリティゾーン (AZ) にスタンバイを作成します。

- ☒ スタンバイインスタンスを作成しないでください

- コンピューティングリソース：EC2 コンピューティングリソースに接続しない
- ネットワークタイプ：IPv4
- Virtual Private Cloud：[Default VPC (vpc-xxxxxx)]（デフォルト VPC を選択）
- サブネットグループ：デフォルト
- パブリックアクセス可能：[あり]
- VPC セキュリティグループ：[新規作成]
- 新しい VPC セキュリティグループ名：[ **YYMMDD**HandsOnDBInstanceSecurityGroup ]  
( **YYMMDD** は本日の日付 )
- アベイラビリティゾーン：指定なし
- 認証機関：デフォルト

**接続 情報**

**コンピューティングリソース**  
このデータベースのコンピューティングリソースへの接続を設定するかどうかを選択します。接続を設定すると、コンピューティングリソースがこのデータベースに接続できるように、接続設定が自動的に変更されます。

☒ **EC2 コンピューティングリソースに接続しない**  
このデータベースのコンピューティングリソースへの接続を設定しないでください。後でコンピューティングリソースへの接続を手動で設定できます。

☐ **EC2 コンピューティングリソースに接続**  
このデータベースの EC2 コンピューティングリソースへの接続を設定します。

**ネットワークタイプ 情報**  
デュアルスタックモードを使用するには、IPv6 CIDR ブロックを指定した VPC のサブネットに関連付けてください。

☒ **IPv4**  
リソースは IPv4 アドレス指定プロトコル経由でのみ通信できます。

☐ **デュアルスタックモード**  
リソースは、IPv4、IPv6、またはその両方を介して通信できます。

**Virtual Private Cloud (VPC) 情報**  
VPC を選択します。VPC は、この DB インスタンスの仮想ネットワーク環境を定義します。

**Default VPC (vpc-0a47c702272a2f21b)**

対応する DB サブネットグループがある VPC のみが表示されます。

**データベースの作成後に、VPC を変更することはできません。**

**DB サブネットグループ 情報**  
DB サブネットグループを選択します。DB サブネットグループは、選択した VPC で DB インスタンスが使用できるサブネットと IP 範囲を定義します。

**デフォルト**

**パブリックアクセス 情報**

☒ **あり**  
RDS はデータベースにパブリック IP アドレスを割り当てます。Amazon EC2 インスタンスと VPC 外の他のリソースはデータベースに接続できます。VPC 内のリソースもデータベースに接続できます。データベースに接続できるリソースを指定する VPC セキュリティグループを 1 つ以上選択します。

☐ **なし**  
RDS はデータベースにパブリック IP アドレスを割り当てません。Amazon EC2 インスタンスと VPC 内の他のリソースのみがデータベースに接続できます。データベースに接続できるリソースを指定する VPC セキュリティグループを 1 つ以上選択します。

## VPC セキュリティグループ (ファイアウォール) 情報

データベースへのアクセスを許可する VPC セキュリティグループを 1 つ以上選択します。セキュリティグループのルールで適切な受信トラフィックが許可されていることを確認します。

☐ 既存の選択  
既存の VPC セキュリティグループの選択

☒ 新規作成  
新しい VPC セキュリティグループの作成

## 新しい VPC セキュリティグループ名

## アベイラビリティゾーン 情報

指定なし

## 認証機関 - 任意 情報

Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-2019 (デフォルト)

If you don't select a certificate authority, RDS chooses one for you.

## ▶ 追加設定

- データベース認証: [パスワード認証]

### データベース認証

データベース認証オプション 情報

☒ パスワード認証  
データベースのパスワードを使用して認証します。

☐ パスワードと Kerberos 認証  
承認されたユーザーに、Kerberos 認証を使ってこの DB インスタンスで認証を行うことを許可するディレクトリを選択します。

データベースの設定が完了したら [データベースの作成] をクリックします。

4. データベースの作成が始まったら [認証情報の詳細を表示] から自動生成されたデータベースのマスターストランパスワードを確認します。

※ パスワードを表示できるのはこのタイミングだけなので確認を忘れないようにしましょう。



5. データベースの作成が完了しステータスが [利用可能] になっていることを確認します。





## ステップ 2. SQL Developer を利用して Oracle DB インスタンスへ接続する

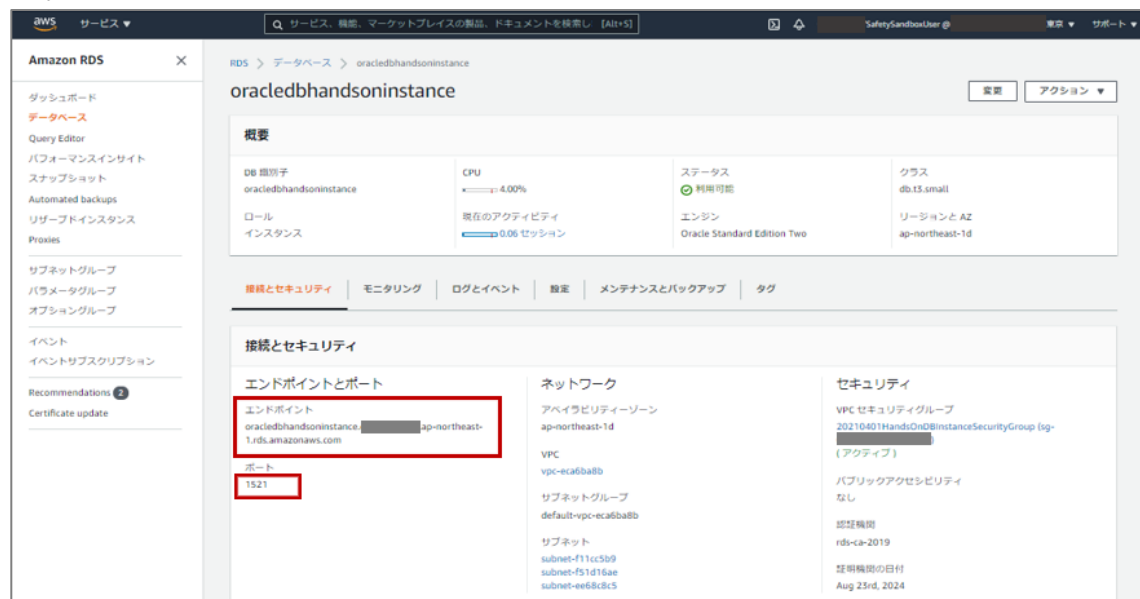
続いて Oracle SQL Developer を使用して DB インスタンスに接続します。

1. DB インスタンスに接続するには、エンドポイントとポート番号が必要なため最初に確認を行います。

データベースの一覧から作成を行った DB インスタンスの名前をクリックします。



DB インスタンスの詳細画面の「接続とセキュリティ」タブからエンドポイントとポート番号を控えます。

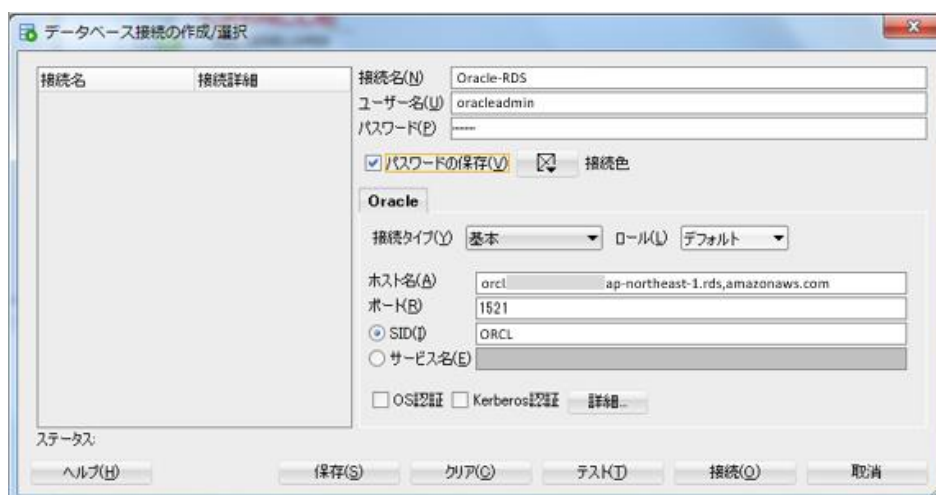


2. Oracle SQL Developer を起動して、接続タブにある「+」のアイコンから [接続の作成] を選択します。



3. [New/Select Database Connection] ダイアログボックスで、DB インスタンスの情報を入力して [接続] をクリックします。

- Connection Name (接続名) : [Oracle-RDS]
- Username (ユーザー名) : [oracleadmin]
- Password (パスワード) : [データベース管理者のパスワード]
- Hostname (ホスト名) : [DB インスタンスのエンドポイント]
- Port (ポート) : [1521]
- SID : [ORCL]



以上で、Oracle DB インスタンスにアクセスして、独自のデータベースを作成したり、DB インスタンスに対してクエリを実行したりすることができるようになりました。

試しにデータベース名を取得するクエリを実行してみましょう。

```
SELECT NAME FROM V$DATABASE
```

Oracle DB インスタンスへ接続することができない場合は、以下の資料を参考にしてください。

- ・ Amazon RDS DB インスタンスに接続できない

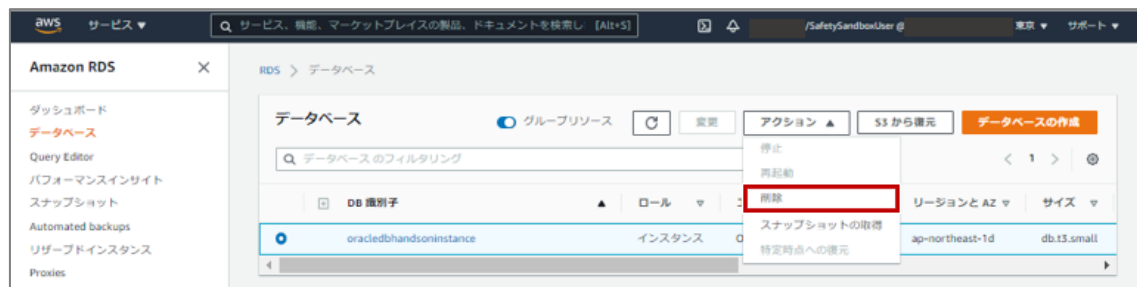
[https://docs.aws.amazon.com/ja\\_jp/AmazonRDS/latest/UserGuide/CHAP\\_Troubleshooting.html#CHAP\\_Troubleshooting.Connecting](https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/UserGuide/CHAP_Troubleshooting.html#CHAP_Troubleshooting.Connecting)

### 3. 後片付け

後片付けを行います。

この作業まで完了していない場合、継続して課金が発生してしまいますので、必ず実施してください。

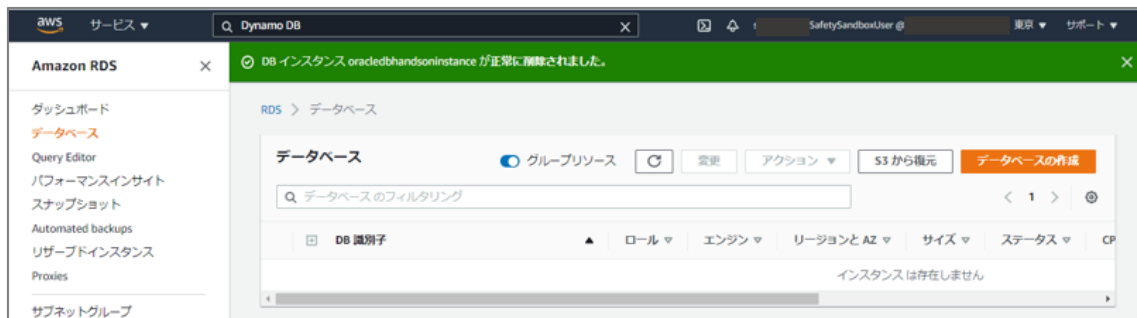
1. 作成した DB インスタンスを選択して [アクション] > [削除] とクリックします。



2. 確認ウィンドウで [インスタンスの削除後、システムスナップショットとポイントインタイムの復元を含自動バックアップが利用不可となることを了承しました] を選択し、[delete me] と入力して k ら [削除] をクリックします。



3. DB インスタンスが削除されたことを確認します。



ハンズオンの終了後、時間に余裕がある場合は EC2 と RDS を組み合わせて利用する公式チュートリアル（所要時間約 50 分）にも取り組んでみるのも一案です。

ここまでの学習範囲で実施することができるため、学習した内容をしっかりと理解できている方は是非取り組んでみてください。

- ・ Amazon RDS で WordPress をデプロイ

<https://aws.amazon.com/jp/getting-started/hands-on/deploy-wordpress-with-amazon-rds/>

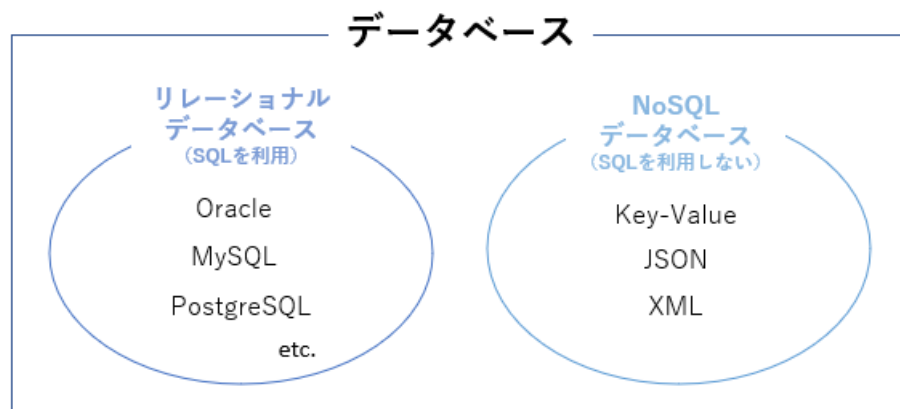
# AWS の NoSQL データベース

---

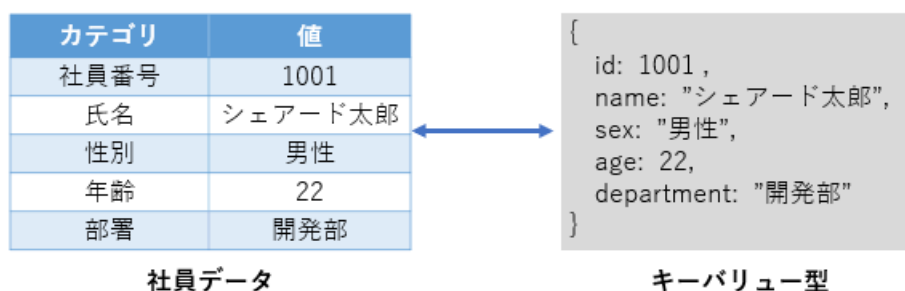
# Dynamo DB

## 1. NoSQL データベース

データベースの中には、SQL を利用して表形式でデータの関連性を管理するリレーショナルデータベース以外にも、SQL を使用しない NoSQL データベースが存在します。NoSQL データベースには様々な種類がありますが、AWS の NoSQL データベースサービスで利用されるのはキーバリュー型の NoSQL データベースです。



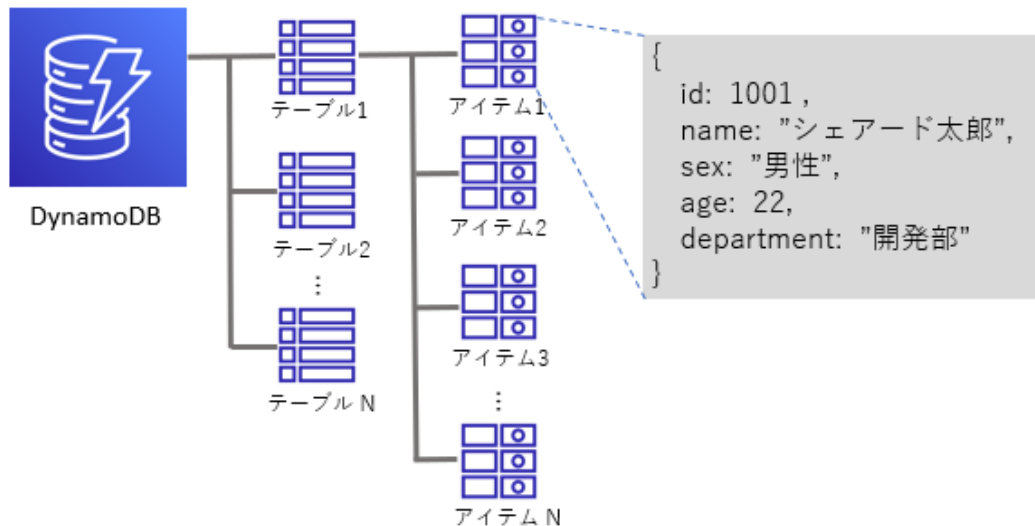
キーバリュー型では、1つのデータをキーとバリュー（値）のペアで扱い、1つのデータを表記する際は、キーとバリューをコロンでつないで「"キー":"バリュー"」と表記します。また、データが複数あり、それらが関連性を持つ場合は、各データをカンマで繋げて、それらを{}で括ることで、複数のデータを持つ1つのデータを表現することもできます。



リレーショナルデータベースと異なり、NoSQL データベースはテーブルの関係性によってデータを管理せず、トランザクションを厳密に管理することはありませんがその分高速にデータ処理を行うことが可能です。この特徴を活かして、NoSQL データベースではオンラインゲームのステータス管理や IoT 機器のログ収集といった大量かつ頻繁に通信される簡素な構造のデータを蓄積する際などに利用されます。

## 2. DynamoDB の概要

Amazon DynamoDB (DynamoDB) はフルマネージド型の No SQL データベースです。データはテーブルの中のアイテム (項目) として扱います。



DynamoDB の主な特徴には次のようなものがあります。

- 完全フルマネージド

DynamoDB はフルマネージド型のデータベースサービスであるため、ユーザーの管理は RDS よりも更に少なく、OS やデータベースエンジン、可用性については一切考える必要がありません。データベースサーバーとしてインスタンスを作成・起動しなくても、テーブルを作ることが可能です。また、DynamoDB の利用時はリージョンだけ指定すればデータは 3 ヶ所のアベイラビリティゾーンに保存されます。

- 高速な処理

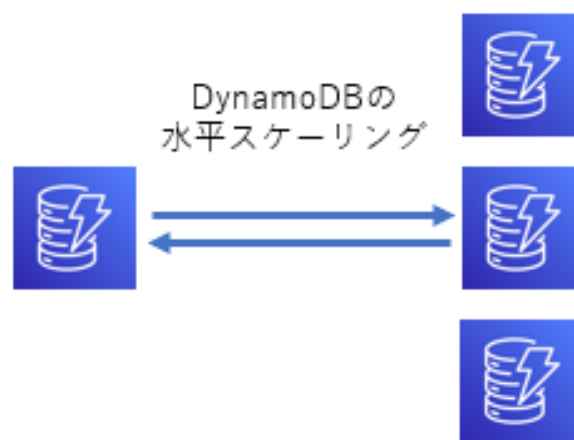
DynamoDB は、NoSQL の特徴を生かして規模に関わらず一貫して数ミリ秒のアクセス速度で処理を行うことができるように設計されています。公式ページでは DynamoDB は「1 日に 10 兆件以上のリクエストを処理することができ、毎秒 2,000 万件を超えるリクエストをサポートします」と紹介されているように、大規模かつ低レイテンシーなアクセスが要求されるデータを扱うことに長けています。



#### 4. DynamoDB のスケーリング

DynamoDB がスケーリングを行う際はスケールアウト/スケールインによる水平スケーリングを行います。NoSQL は、RDS と違ってトランザクションを厳密に管理しないため、事前の更新処理が完了していなくても次の更新処理を実行して、更新が行われなくなったタイミングで結果的にデータが一貫性を持つことを目指す結果整合性モデルを採用しています。

この結果整合性モデルによって大量のアクセスがあってもパフォーマンスを保ったまま処理を行うことができるため、DynamoDB のスケーリングは複数のサーバーに処理を分散する水平スケーリングの形で行われます。



##### ■ 参考

- ・ Amazon DynamoDB (マネージド NoSQL データベース)  
<https://aws.amazon.com/jp/dynamodb/>

# ハンズオン： NoSQL テーブルでのクエリの実行

## 1. はじめに

### 1-1. ハンズオンのゴール

- DynamoDB を利用してテーブルを作成しデータの追加・削除を行う

### 1-2. 事前準備

- 安心サンドボックスを利用して AWS マネジメントコンソールへサインインしていること

### 1-3. ハンズオンのステップ

- ステップ 1, NoSQL テーブルを作成する
- ステップ 2, NoSQL テーブルにデータを追加する
- ステップ 3, NoSQL テーブルでクエリを実行する

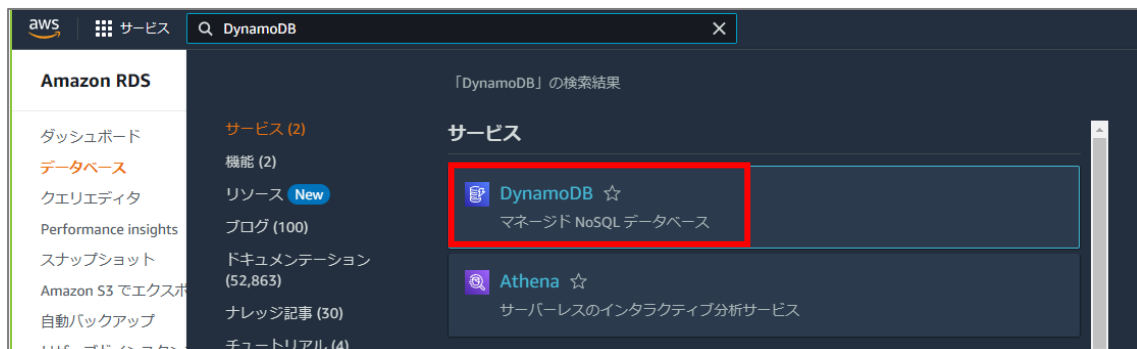
ハンズオンで入力する設定値は最低限の動作を確保するための値を採用しています。

## 2. ハンズオンの実施

### ステップ 1. NoSQL テーブルを作成する

DynamoDB コンソールを使用してテーブルを作成します。

1. AWS マネジメントコンソールで [DynamoDB] を選択します。



2. DynamoDB のサービスページが標示されたら [テーブルの作成] をクリックします。



3. DynamoDB テーブルの作成画面で次のように設定を行い [作成] をクリックします。

- テーブル名：[Music]
- パーティションキー：[Artist]
- ソートキー：[songTitle]
- デフォルト設定の使用：設定をカスタマイズ

※ デフォルト設定の仕様を無効化することでデフォルトで Auto Scaling を有効にするために必要な IAM ロールを作成することができます。

※ テーブル設定については、[デフォルト設定の使用] チェックを外す以外には操作不要です。



aws サービス 検索 [Alt+S]

DynamoDB > テーブル > テーブルの作成

### テーブルの作成

**テーブルの詳細** 情報  
DynamoDB は、テーブルの作成時にテーブル名とプライマリキーのみを必要とするスキーマレスデータベースです。

**テーブル名**  
テーブルを識別するために使用されます。  
  
0~255 文字で、文字、数字、アンダースコア、ハイフン、ピリオド (.) のみを使用できます。

**パーティションキー**  
パーティションキーは、テーブルのプライマリキーの一部です。これは、テーブルから項目を取得し、スケーラビリティと可用性のためにホスト間でデータを割り当てるために使用されるハッシュ値です。  
 文字列  
0~255 文字 (大文字と小文字が区別されます)。

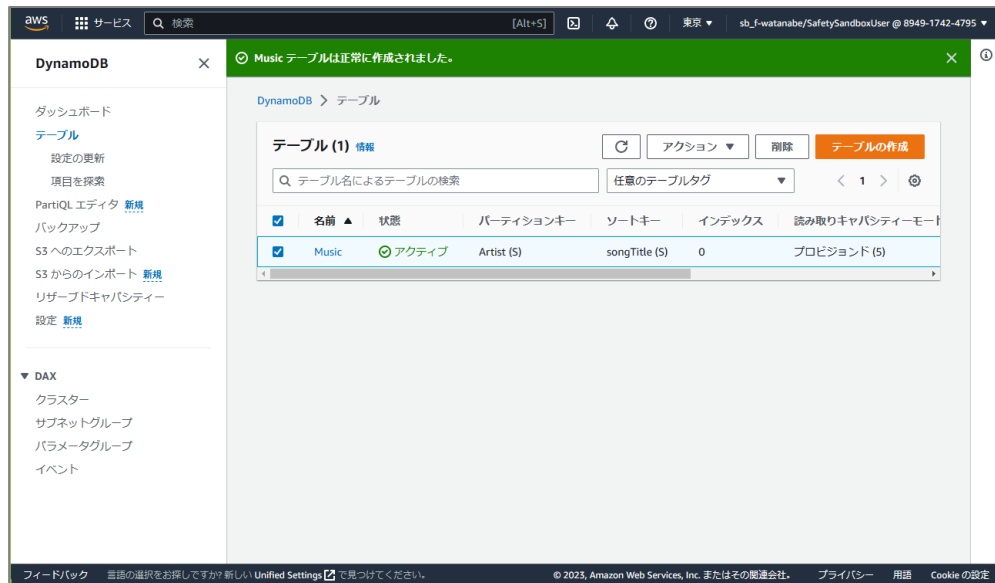
**ソートキー - オプション**  
ソートキーは、テーブルのプライマリキーの 2 番目の部分として使用できます。ソートキーにより、同じパーティションキーを共有するすべての項目をソートまたは検索できます。  
 文字列  
0~255 文字 (大文字と小文字が区別されます)。

**テーブル設定**

☐ デフォルト設定  
最も短時間でテーブルを作成します。これらの設定は、今すぐ変更するか、テーブルの作成後に変更できます。

☒ 設定をカスタマイズ  
これらの高度な機能を使用して、DynamoDB をニーズに合わせて設定します。

## 4. テーブルが作成されたことを確認します。



## ステップ 2. NoSQL テーブルにデータを追加する

作成した DynamoDB のテーブルにデータを追加します。

1. 作成したテーブルを選択し、「アクション」>「項目を作成」をクリックする。



2. データ入力ウィンドウで次のように情報を記入して [保存] をクリックします。

- Artist : [No One You Know]
- songTitle : [Call Me Today]

**項目を作成**

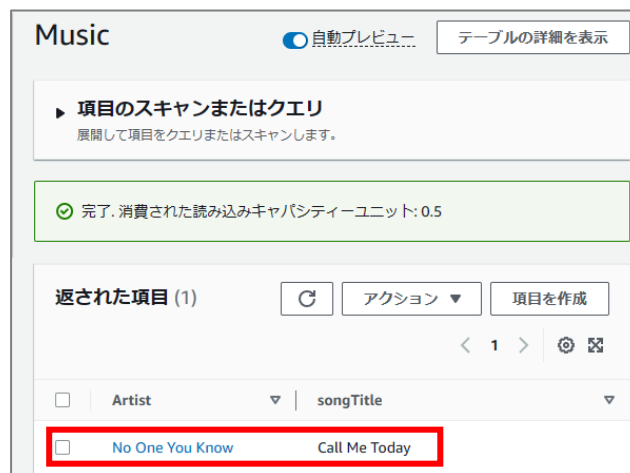
項目の属性を追加、削除、または編集できます。属性は、最大 32 レベルの深さまで他の属性内にネストできます。 [詳細はこちら](#)

新しい属性の追加 ▼

属性名	値	タイプ
Artist - パーティションキー	No One You Know	文字列
songTitle - ソートキー	Call Me Today	文字列

キャンセル **項目を作成**

3. テーブルに項目が追加されたことを確認します。



4. 「項目の作成」をクリックし、同様の手順で以下の項目を追加します。

- Artist : [No One You Know]、songTitle : [My Dog Spot]
- Artist : [No One You Know]、songTitle : [Somewhere Down The Road]
- Artist : [The Acme Band]、songTitle : [Still in Love]
- Artist : [The Acme Band]、songTitle : [Look Out, World]



返された項目 (5)

アクション ▼ 項目を作成

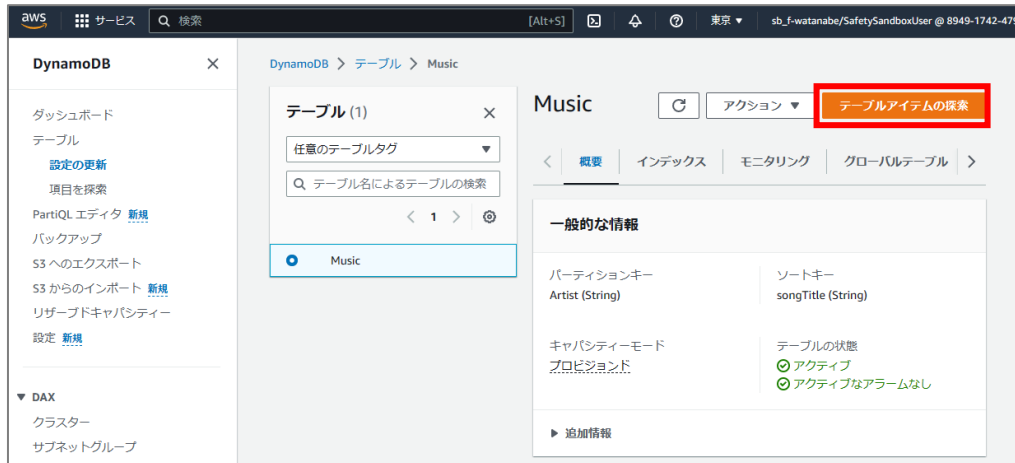
< 1 > ⚙️

<input type="checkbox"/>	Artist ▼	songTitle ▼
<input type="checkbox"/>	No One You Know	Call Me Today
<input type="checkbox"/>	No One You Know	My Dog Spot
<input type="checkbox"/>	No One You Know	Somewhere Down The Road
<input type="checkbox"/>	The Acme Band	Look Out, World
<input type="checkbox"/>	The Acme Band	Still in Love

### ステップ 3. NoSQL テーブルでクエリを実行する

クエリを使用してテーブルに追加したデータの検索を行います。

1. 「テーブルの詳細」をクリックし、詳細画面に遷移後、「テーブル相手の探索」をクリックする。



2. 「クエリ」を選択し、[Artist]ボックスに [No One You Know] を指定して [開始] をクリックします。  
すると条件に該当するデータのみを表示することができます。





3. 同様に [Artist] ボックスに [The Acme Band]、[songTitle] ボックスは [次で始まる] を選択して、[S] と入力を行い [実行する] をクリックします。
- すると、「The Acme Band」の「S」から始まる曲のデータのみが表示されます。



4. テーブル内の項目を削除する場合は、[スキャン] にチェックを入れ、[実行する] をクリックする。該当の項目を選択した上で [アクション] > [削除] とクリックします。



返された項目 (5)		アクション ▲	項目を作成
<input type="checkbox"/>	No One You Know	My Dog Spot	
<input type="checkbox"/>	No One You Know	Somewhere Down The Road	
<input type="checkbox"/>	The Acme Band	Look Out, World	
<input checked="" type="checkbox"/>	The Acme Band	Still in Love	

### 3. 後片付け

後片付けを行います。

この作業まで完了していない場合、継続して課金が発生してしまいますので、必ず実施してください。

1. ダッシュボードから[テーブル]をクリックする。



2. 作成したテーブルを選択し、[削除]をクリックする。



3. テーブルの削除ウィンドウが標示されるため [このテーブルに対応するすべての CloudWatch アラームを削除します] にチェックを入れた上で [削除] と入力して、[削除] をクリックします。

テーブルの削除

×

テーブルを削除しようとしています。

- Music

☒

このテーブルのすべての CloudWatch アラームを削除します。

☐

削除する前にこのテーブルのバックアップを作成します。  
このチェックボックスをオンにしないと、削除中のデータを復元できません。

このテーブルの削除を確認するには、ボックスに「削除」と入力します。

削除

キャンセル

テーブルの削除



クラウド基礎 2-1 第1版

---

2021年1月25日 初版 発行

---

著者 株式会社 SystemShared 東京 IT スクール

---

文中の社名、商品名等は各社の商標または登録商標である場合があります。