

# AWS Black Belt Online Seminar

## Amazon RDS Proxy 概要

永末 健太

Database Specialist Solutions Architect

2025/12



# AWS Black Belt Online Seminar とは

- ・ 「サービス別」「ソリューション別」「業種別」などのテーマに分け、  
アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- ・ AWS の技術担当者が、 AWS の各サービスやソリューションについてテーマごとに動画を公開します
- ・ 以下の URL より、過去のセミナー含めた資料などをダウンロードすることができます
  - > <https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/>
  - > <https://www.youtube.com/playlist?list=PLzWGOASvSx6FlwIC2X1nObr1KcMCBBlqY>



ご感想は X (Twitter) へ！ハッシュタグは以下をご利用ください  
#awsblackbelt



# 内容についての注意点

- ・ 本資料では資料作成時点のサービス内容および価格についてご説明しています。AWS のサービスは常にアップデートを続けているため、最新の情報は AWS 公式ウェブサイト (<https://aws.amazon.com/>) にてご確認ください
- ・ 資料作成には十分注意しておりますが、資料内の価格と AWS 公式ウェブサイト記載の価格に相違があった場合、AWS 公式ウェブサイトの価格を優先とさせていただきます
- ・ 価格は税抜表記となっています。日本居住者のお客様には別途消費税をご請求させていただきます
- ・ 技術的な内容に関しましては、有料の [AWS サポート窓口](#)へお問い合わせください
- ・ 料金面でのお問い合わせに関しましては、[カスタマーサポート窓口](#)へお問い合わせください (マネジメントコンソールへのログインが必要です)

# 自己紹介

## 永末 健太

アマゾンウェブサービスジャパン  
データベーススペシャリスト SA

エンタープライズのお客様を中心に  
データベース移行のご支援をしています。

好きな AWS サービス  
Amazon RDS / Amazon Aurora / DynamoDB



# 本セミナーの対象者

- AWS で Amazon Aurora / RDS の利用を検討中、または今後検討予定の方
- 接続プーリングの実装やフェイルオーバーの高速化を実現したいが、  
アプリケーションは極力変更したくない方

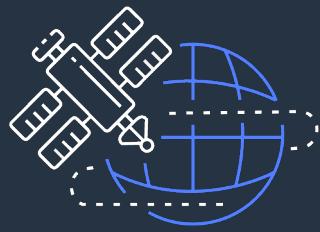
# アジェンダ

1. Amazon RDS Proxy 概要
2. Amazon RDS Proxy の主な機能
3. Amazon RDS Proxy を利用する上でのポイント
4. まとめ

# Amazon RDS Proxy 概要

# Amazon RDS Proxy

Amazon Aurora / RDS 向けの高可用性フルマネージド型データベースプロキシ  
アプリケーションのスケーラビリティやデータベース障害に対する回復力と  
安全性の向上を実現



データベース接続を  
プールおよび共有する事で  
アプリケーションの  
スケーリングを改善



アプリケーションの可用性を  
高め、データベースの  
フェイルオーバー時間を短縮

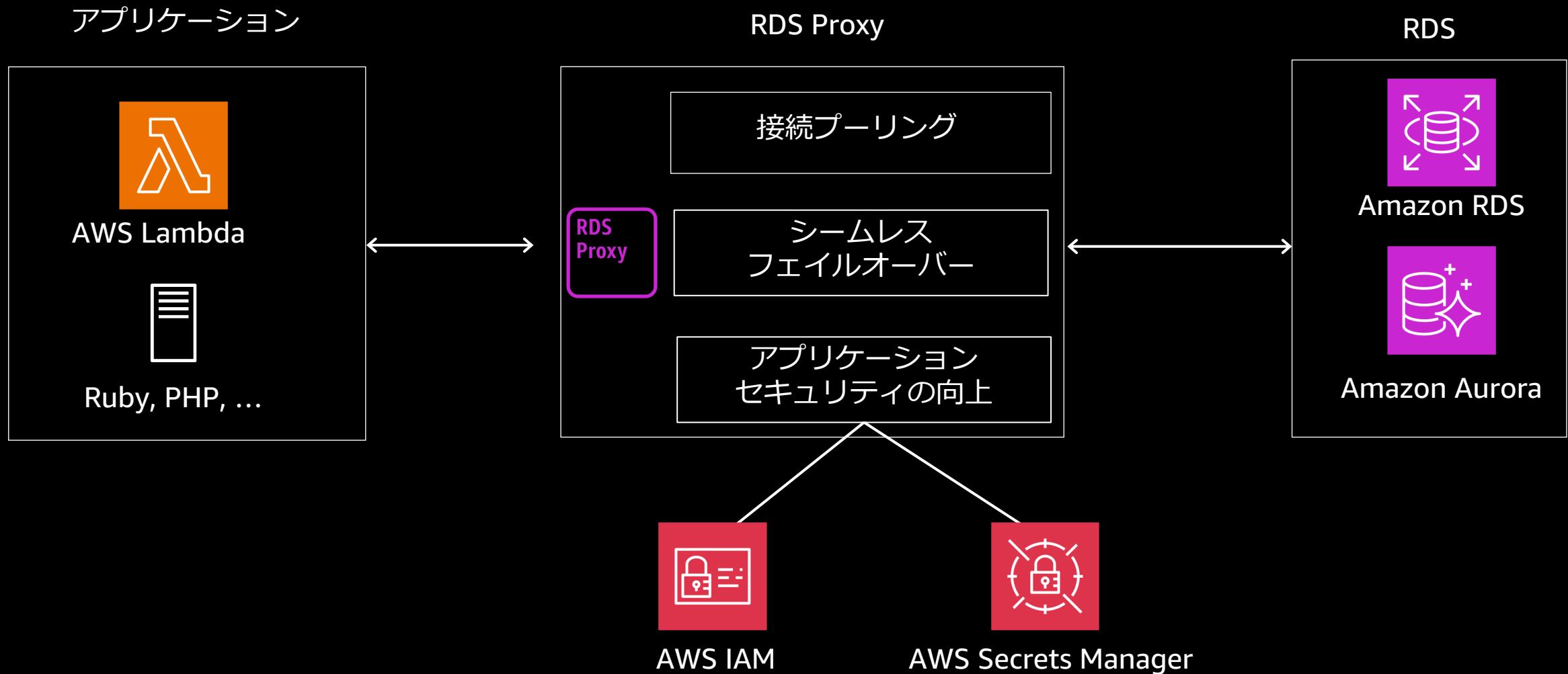


データベースアクセス制で、  
アプリケーションデータの  
セキュリティを管理



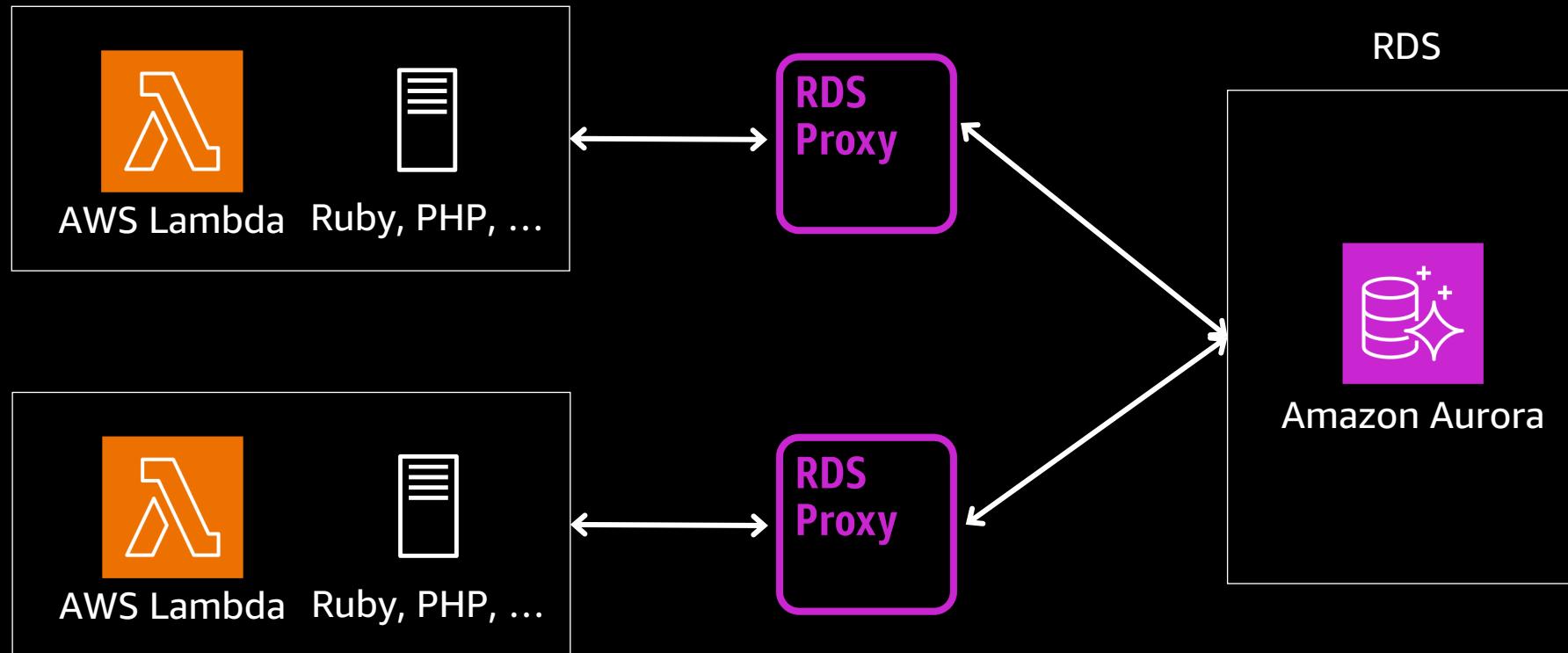
フルマネージド  
データベースプロキシ、  
データベースとの  
完全な互換性

# Amazon RDS Proxy 全体像



# Amazon RDS Proxy の構成例

各プロキシは、1つのターゲット DB インスタンスに関連付けることができます。  
また、同じ DB インスタンスに複数のプロキシを関連付けることができます。



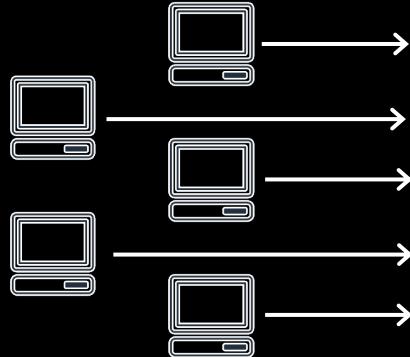
# Amazon RDS Proxy の主な機能



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# ①接続プーリング

アプリケーション



Connection  
Pool

RDS

thd_id	conn_id	pid	program_name	cpu_latency	current_memory
41018	40732	NULL	rds proxy	0 ps	37.93 KiB
40879	40593	NULL	rds proxy	0 ps	1.23 MiB
40877	40591	NULL	rds proxy	0 ps	1.23 MiB
40889	40603	NULL	rds proxy	0 ps	1.23 MiB
40968	40682	NULL	rds proxy	0 ps	5.18 MiB
41071	40785	NULL	rds proxy	0 ps	9.58 MiB

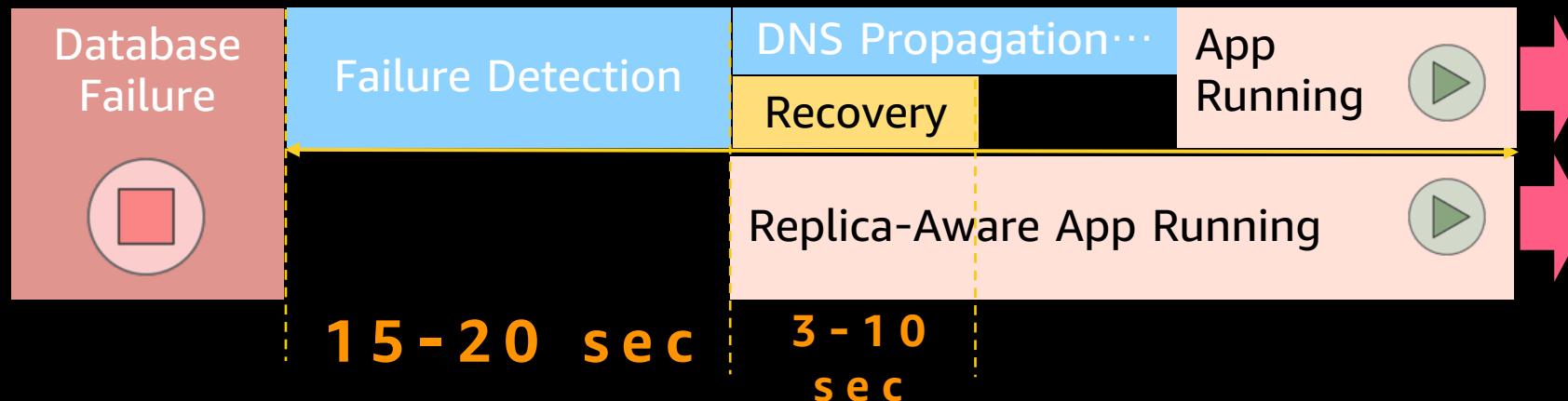
→ より多くの処理が  
実行可能になる

- 接続プーリング
  - 接続の開閉に伴うデータベースの負荷軽減 (TLS/SSL のハンドシェイク、認証、ネゴシエーション機能などの CPU 負荷等) を削減
- 接続の多重化
  - 接続の再利用により、データベース接続に必要なコンピューティングリソース (主にメモリ) を削減
  - max\_connections エラーの発生頻度の抑制

## ②シームレスかつ高速なフェイルオーバー

Amazon Aurora で1つ以上のリーダーインスタンスがある場合、フェイルオーバー時の一般的なサービスの復元時間は60秒未満、多くの場合30秒未満です<sup>(\*)1)</sup>

- Aurora インスタンスのフェイルオーバー時の挙動イメージ

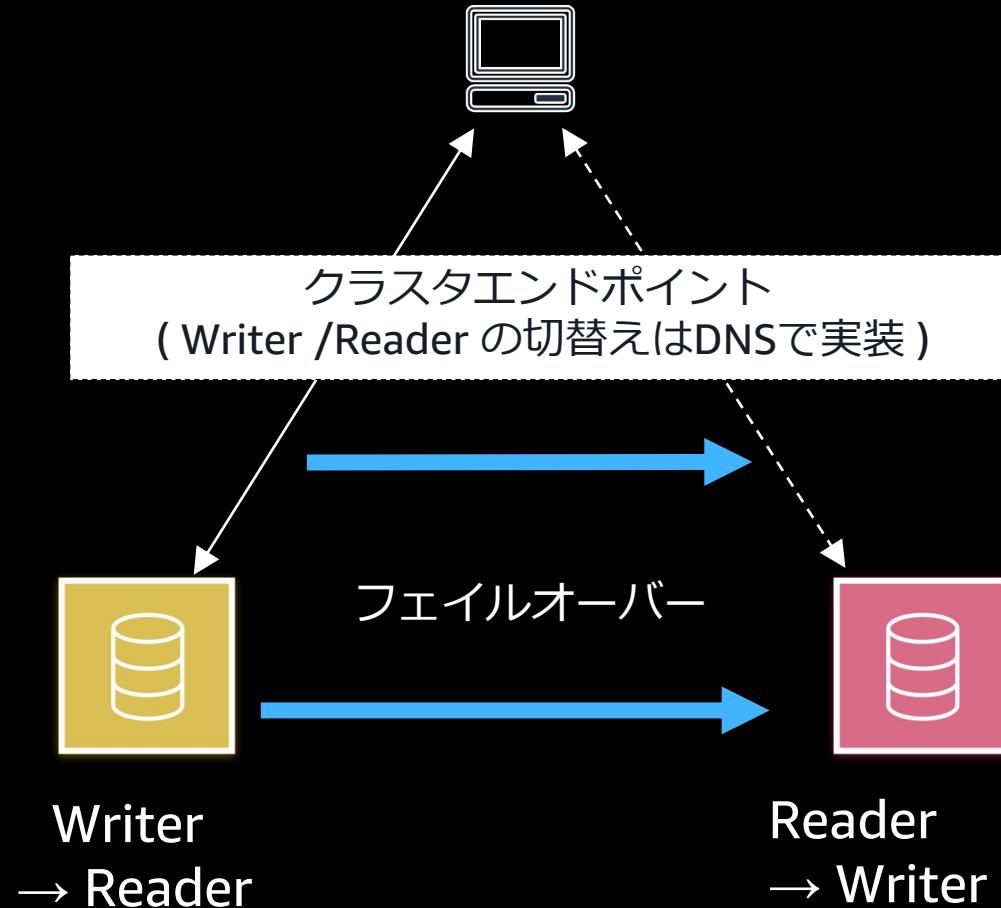


高速にフェイルオーバーを実行するためのポイント<sup>(\*)2)</sup>

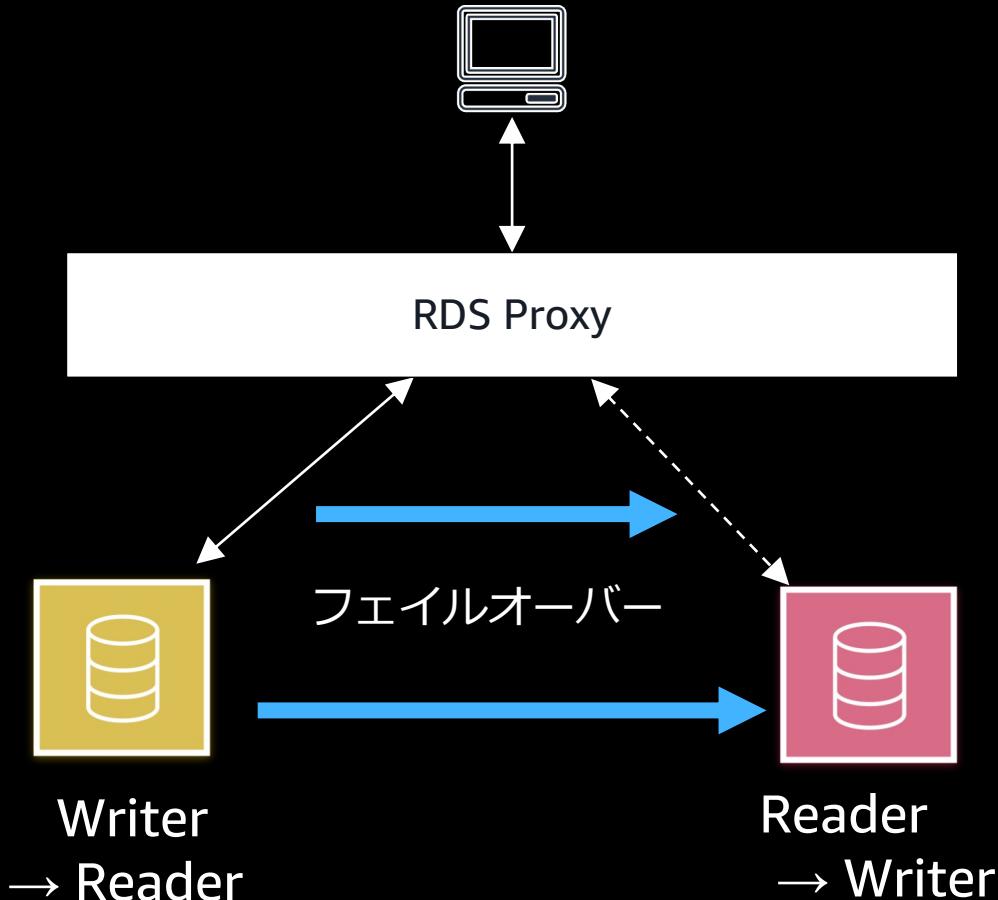
- TCP キープアライブを短い時間に設定する
- Java DNS キャッシュの有効期限を短い値に設定する
- **RDS Proxy を経由して接続する**
- AWS Advanced JDBC Wrapper Driver を使用する

# フェイルオーバー動作 概念図

【RDS Proxy を利用していない時】

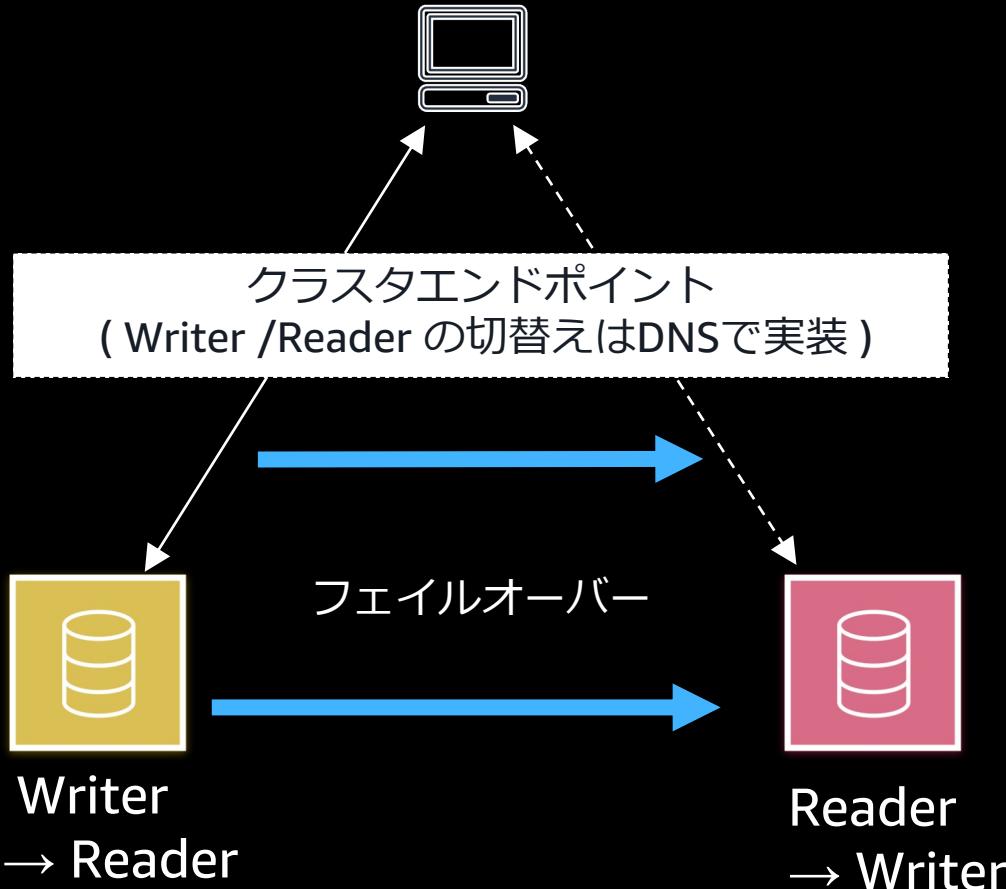


【RDS Proxy 利用時】



# フェイルオーバー動作 概念図

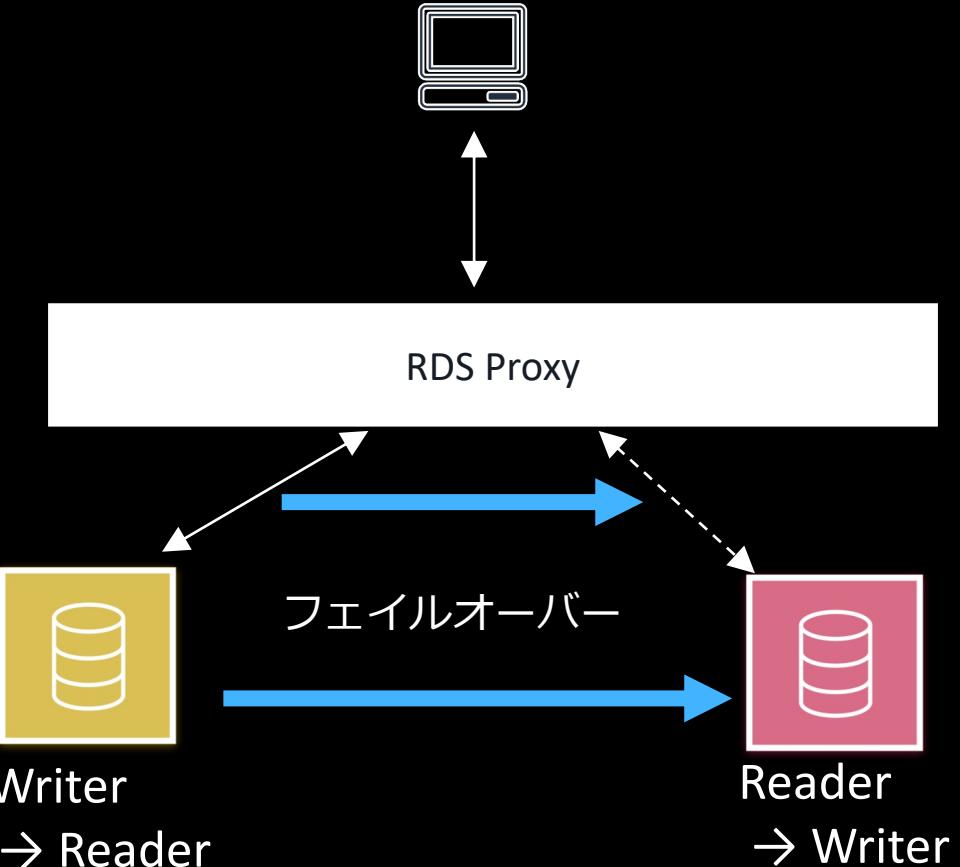
【RDS Proxy を利用していない時】



- DNS レコード変更に伴う影響を考慮する必要がある
- DNS キャッシュの考慮が必要 (TTL)  
キャッシュする時間が長い場合
  - フェイルオーバー後に接続するまでの時間がかかる
  - キャッシュする時間が短い
  - フェイルオーバー後のReaderに接続してしまう
- フェイルオーバー後、既存の接続の再接続が必要

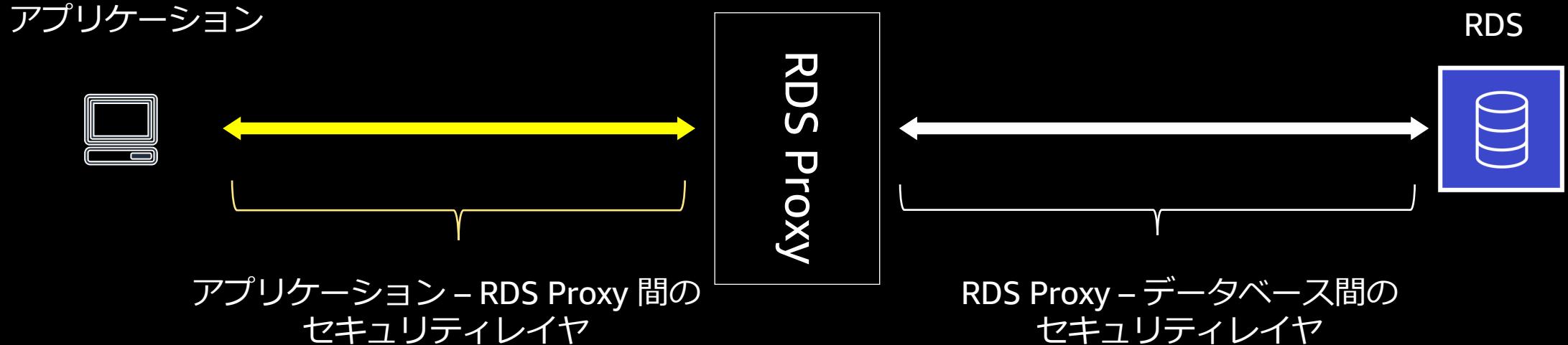
# フェイルオーバー動作 概念図

## 【RDS Proxy 利用時】



- DNS レコード変更に伴う影響を考慮する必要がない
- DNS キャッシュの考慮不要  
→ 高速なフェイルオーバーの実現
- RDS Proxy がバックエンドデータベースのフェイルオーバーを検知  
→ 確実にWriter インスタンスに接続  
→ アイドル状態の接続はアプリケーション  
– RDS Proxy間の接続を維持した状態で、  
フェイルオーバー先のデータベースに接続。  
アプリケーションの再接続は不要  
→ クエリ処理中・トランザクション中等の  
接続に関しては再接続が必要

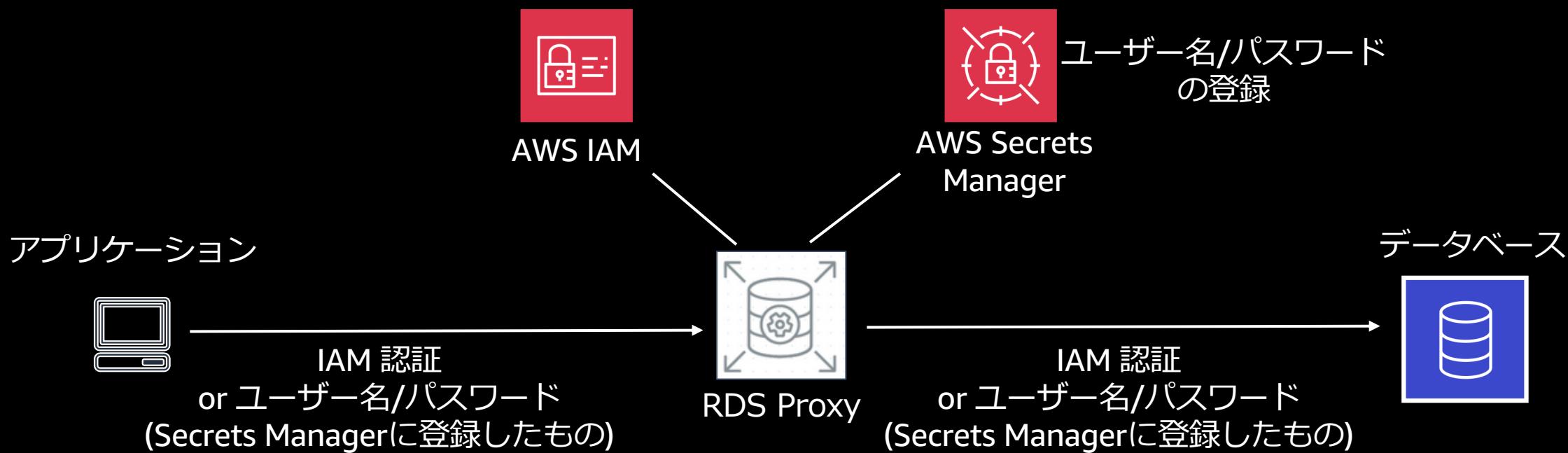
### ③アプリケーションセキュリティの向上



データベースサーバーがネイティブのユーザー名 / パスワード認証しか対応していない場合でも、RDS Proxy への接続は IAM 認証を利用可能。アプリケーションコードに認証情報を埋め込まない事によるセキュリティ向上を実現  
RDS Proxy は TLS 1.0-1.3をサポートし、基盤DBエンジンより高いバージョンでの接続が可能

### ③アプリケーションセキュリティの向上

- データベースが IAM 認証に対応している場合、RDS Proxy – データベース間も IAM 認証可能
- IAM 認証以外に AWS Secrets Manager によるデータベース認証情報の一元管理も可能



# Amazon RDS Proxy を 利用する上でのポイント



# 接続設定

**プロキシ設定**  
プロキシにより、アプリケーションのスケーラビリティが向上し、データベースの障害に対して透過性が高くなり、安全性が向上します。

**エンジンファミリー 情報**

- MariaDB and MySQL  
Supports Aurora MySQL, RDS for MariaDB, and RDS for MySQL
- PostgreSQL  
Supports Aurora PostgreSQL and RDS for PostgreSQL
- SQL Server  
Supports RDS for SQL Server

**プロキシ識別子**  
プロキシ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 リージョンで所有する、すべてのプロキシ間で一意である必要があります。

制約として、使用できるのは1~60 文字以内で英数字またはハイフンのみです。先頭の文字は英文字である必要があります。また、ハイフンを連続で 2 つ使ったり、最後の文字をハイフンにしたりすることはできません。

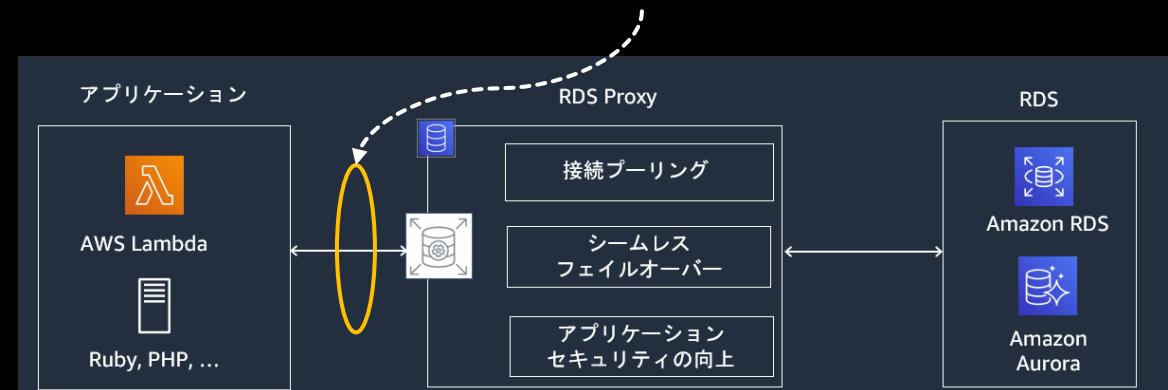
**アイドルクライアントの接続タイムアウト**  
アプリケーションからのアイドル接続は、指定した時間が経過すると閉じられます。  
 01  時間  30  分

最小は 1 分、最大で 8 時間です。

## アイドルクライアントの接続タイムアウト (IdleClientTimeout)

### クライアントとRDS Proxy間のタイムアウト設定

- コネクションのアイドル時間
- アイドル状態がこの設定時間を超えると接続はプールに戻され、新しいクライアント接続で再利用できる



# 接続設定

ターゲットグループの設定  
ターゲットグループは、プロキシが接続できるデータベースのコレクションです。現在、各ターゲットグループを 1 つの RDS DB インスタンスまたは Aurora DB クラスターに関連付けることができます。

データベース  
プロキシに関連付ける RDS DB インスタンスまたは Aurora DB クラスターを選択します。  
データベース名を選択

接続プールの最大接続数 情報  
データベースの最大接続制限に対する割合として、許可される最大接続数を指定します。  
100 パーセント  
例えは、最大接続数を 5,000 接続に設定した場合、50% を指定すると、プロキシはデータベースに対して最大 2,500 個の接続を作成できます。

リーダーエンドポイントを含める 情報  
リーダーエンドポイントを追加  
RDS には、このプロキシのリーダーエンドポイントが含まれています。

▼ 追加のターゲットグループの設定

セッション固定フィルタ 情報  
セッション固定は、アプリケーション接続の有効期間中、アプリケーション接続をプロキシからのデータベース接続に関連付けます。  
なし

接続借用タイムアウト 情報  
プールからの借用 DB 接続のタイムアウトです。  
02 分 00 秒  
最大: 5 分

初期化クエリ - オプション 情報  
1 つ以上の SQL 構文を指定して、各接続の初期セッション状態を設定します。ステートメントはセミコロンで区切れます。

文字数: 0/200,000

## 接続プールの最大接続数 (MaxConnectionsPercent)

ターゲット DB の max\_connections に対する割合 (1%~100%) の値を指定

- 1つのデータベースに複数のプロキシを割り当てる事が可能
- Proxyを経由しない接続も可能

※ 同じデータベースに割り当てるプロキシの割合とProxyを経由しない接続の合計を100%以下に設定する事

```
mysql> select @@max_connections;
+-----+
| @@max_connections |
+-----+
|          1000 |
+-----+
```

## 接続借用タイムアウト (Connection Borrow Timeout)

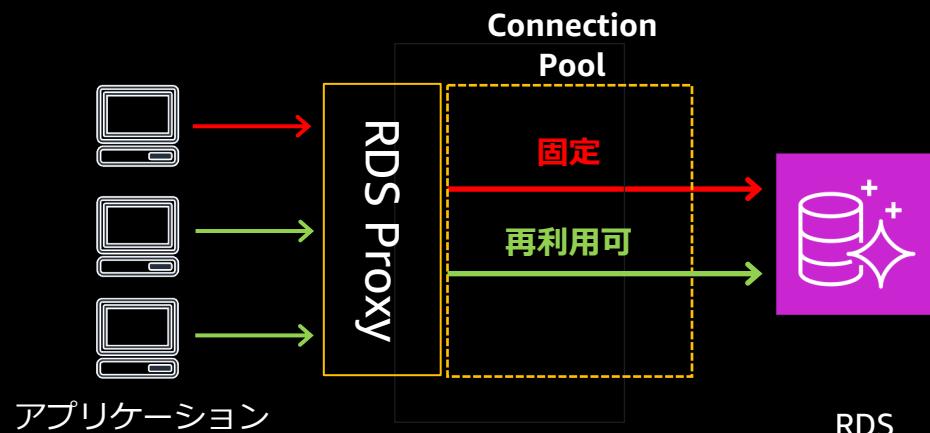
アプリケーションからの接続要求に対して接続プールが割当てられない場合に待機する時間 (デフォルト120秒)

- 接続数が最大値に達し、接続プールで利用可能な接続がなくなった場合
- フェイルオーバーオペレーションが進行中である為に WRITER インスタンスが使用できない場合

# RDS Proxy の固定 (ピン留め)

RDS Proxy は、他のセッションに不適切なセッション状態の変化を検出すると、クライアント接続を特定の DB 接続に自動的に固定します。固定によって接続の再利用の有効性が低下します。

固定化発生時のイメージ



不適切なセッション状態の例( MySQL の場合) :

- LOCK TABLE などの明示的なロックステートメント
- 一時テーブルの作成
- ROW\_COUNT、FOUND\_ROWS、および LAST\_INSERT\_ID の実行
- プリペアドステートメントの使用など

以下の方法で固定を最小化し、トランザクションレベルの接続の再利用 (多重化) を最大化可能

- 固定の原因となる可能性のある不要なデータベースリクエストを避ける
- 初期化クエリやセッションの固定フィルタオプションを使用する

参考) [https://docs.aws.amazon.com/ja\\_jp/AmazonRDS/latest/UserGuide/rds-proxy-pinning.html](https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/UserGuide/rds-proxy-pinning.html)

# RDS Proxy の料金

RDS Proxy は、基となるインスタンスの容量に基づいて料金が設定されています。

- プロビジョニングインスタンスの場合 : **vCPU ごとに1時間単位**
- Serverless v2 の場合 : データベースが消費する **Aurora Capacity Unit (ACU) ごとに1時間単位**

## 東京リージョンの料金

インスタンス種別	料金	最低料金
Aurora Serverless v2	USD 0.025/ACU 時間	8 ACU
プロビジョニングインスタンス	USD 0.018/vCPU 時間	2 vCPU

※1時間未満の場合、10分を最低料金として秒単位で請求されます

料金例 : Aurora クラスター内に db.r8g.large でライターおよびリーダー1台を構成した場合

1ヶ月を30日とした場合、以下の計算式により **USD 51.84** が課金されます

計算式 : Total 4 vCPU × 24 時間 × 30 日 × USD 0.018 = USD 51.84



# まとめ



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# 【再掲】 Amazon RDS Proxy

Amazon RDS 向けの高可用性フルマネージド型データベースプロキシ  
アプリケーションのスケーラビリティやデータベース障害に対する回復力と  
安全性の向上を実現



データベース接続を  
プールおよび共有する事で  
アプリケーションの  
スケーリングを改善

アプリケーションの可用性を  
高め、データベースの  
フェイルオーバー時間を短縮

データベースアクセス制で、  
アプリケーションデータの  
セキュリティを管理

フルマネージド  
データベースプロキシ、  
データベースとの  
完全な互換性

# Thank you!

