AWS Aurora(RDS)演習

クラス　　　　No. 氏名

※Leaner Labでの実施

□AWS Aurora

今までの課題ではMySQLを使用してサイトを構築しました。今回は、データベースサーバをMySQLからAuroraに変更します。Auroraは MySQLとの互換性があるため、データベースの内容については、MySQLと同じものを使用します。

1.Amazon Auroraの特徴をあげなさい。

|  |
| --- |
| MySQLおよびPostgreSQL互換の高パフォーマンス、高可用性、耐障害性を持つフルマネージドデータベースサービスで、自動バックアップ、スケーラビリティ、セキュリティ機能を提供し、コスト効率よく運用できます。 |

2.次のAuroraに関する用語について説明しなさい。

　①Auroraレプリカ：［　プライマリDBインスタンスのデータを読み取り専用で同期するインスタンス　］

　②プライマリDBインスタンス：［　Auroraクラスタ内で書き込みと読み取りの両方を処理するメインインスタンス　］

　③エンドポイント：［　Auroraデータベースクラスタへの接続用URLで、クライアントがデータベースにアクセスするために使用します　］

　④リーダーインスタンス：［ Auroraクラスタ内で読み取り専用のクエリを処理するインスタンス　］

　⑤ライターインスタンス：［ Auroraクラスタ内で書き込み操作を担当するインスタンスです　］

　⑥Auroraは［　　3　　］つのAZに保存される。バックアップは［　　S3　］に保存される。

□事前準備と確認

CloudFormationを使用して、Webサーバーの環境を構築します。

1.aurora.ymlファイルを使用して環境を構築しなさい。スタック名：クラス名番号　キーペア：作成済みのもの

2.Webサーバー(クラス名\_aurora\_websv)とPHPが動作しているか確認しなさい。http://IPアドレス/test.php

［　確認できた　・　確認できない　］　＊test.phpは各自で作成すること。

IPv4アドレス：[ 3.86.47.50 ]　インスタンスID：[ i-06801002b3bc34716 ]

3.2で確認したWebサーバへTera Termを使用して接続しなさい。

4.今回はAuroraにアクセスするため、MySQLクライアントが必要となる。次のコマンドを使用してインストールしなさい。

|  |
| --- |
| sudo yum install -y mariadb105 |

5.現在EC２(クラス名\_aurora\_websv)で使用しているアベイラビリティゾーンは：［ us-east-1c ］

6.次の仕様に従い、RDSを使用するためのサブネットを作成しなさい。

VPC：クラス名\_aurora\_vpc サブネット名：クラス名\_aurora\_sub

アベイラビリティゾーン：使用していないものIPv4 CIDRブロック：10.2.11.0/24

サブネットは作成できたか確認しなさい。［　確認できた　］

サブネットID：[ subnet-0172b39d65f151e72 ]

□Auroraデータベースの作成

1.次の仕様に従いAurora用のセキュリティグループを作成しなさい。(VPC>セキュリティグループ)

　セキュリティグループ名：クラス名番号-aurora-sg　　説明：クラス名番号-aurora-sg

　VPC：クラス名\_aurora\_vpc インバウンドドルール：タイプ(Aurora/MySQL) ソース：10.2.0.0/16

セキュリティグループは作成できたか確認しなさい。［　確認できた ］

　セキュリティグループID：[ sg-03b680dd91824c078 ]

2.サブネットグループを作成しなさい(RDS>サブネットグループ)。

　名前：クラス名番号-subg 説明：クラス名番号-subg　　VPC：クラス名\_aurora\_vpc

　サブネットを追加：アベイラビリティゾーン(設定されているもの)

　サブネット：クラス名\_web\_sub、クラス名\_aurora\_subがあるAZを選択する。

　※パラメータグループはデフォルトのものを使用します。

3.次の仕様に従いAuroraにデータベースを作成しなさい。(RDSで構築を行う)

　・エンジンのオプション

　エンジンのタイプ：Amazon Aurora 　エディション：Aurora MySQL 3.05.2

・認証情報管理：セルフマネージド　　・テンプレート：開発/テスト

・設定

DBクラスター識別子：クラス名番号-aurora-db　マスターユーザー名：root パスワード：123qwecc

・インスタンスのタイプ　「バースト可能クラス(tクラスを含む)」　db.t3.small

・可用性と耐久性　「Auroraレプリカを作成しない」を選ぶ。

・接続

VPC：クラス名\_aurora\_vpc サブネットグループ：クラス名番号-subg (自動で選択される)

セキュリティグループ：「既存の選択」　クラス名番号-aurora-sg

・モニタリング　「拡張モニタリングの有効化」のチェックを外す(外さない場合、エラーになってしまうため)

4.Amazon Auroraの確認

①Amazon Auroraは作成できたか確認しなさい。　［　作成できた　］

②ライターインスタンス名：  
［ sk3a04-aurora-db.cluster-c59kyyscbffa.us-east-1.rds.amazonaws.com ］  
③ﾗｲﾀｰｲﾝﾝｽﾀﾝｽのｴﾝﾄﾞﾎﾟｲﾝﾄ：  
［ sk3a04-aurora-db-instance-1.c59kyyscbffa.us-east-1.rds.amazonaws.com ］

5.EC２からAuroraのエンドポイントへ接続しなさい。接続方法はMySQLと同じ。

|  |
| --- |
| mysql -h Auroraのエンドポイント(ライターインスタンス) -u root -p |

［　接続できた　］

6.データベース、テーブルを作成後、データを追加しなさい。＊DB.txtファイルを参照すること。

|  |
| --- |
| select \* from products: |

上のSQLを実行して、正しい結果か確認しなさい。［　確認できた　］

7.aurora-db.phpのエンドポイント内容をAmazon Auroraのエンドポイントに変更しなさい。

8.auroa-db.phpをアップロードして、ドキュメントルートへ移動しなさい。

|  |
| --- |
| sudo mv aurora-db.php /var/www/html |

9.ブラウザでWebページが表示されたか確認しなさい。　［　確認できた　］

http://IPアドレス/aurora-db.php

テキスト, テーブル, 手紙

自動的に生成された説明

10.変更したphpファイルを「クラス名番号名前\_aurora-db.php」、このファイルを「クラス名番号名前\_auroradb」に変更してsv23に提出しなさい。

□終了処理

作成したAuroraデータベース、Webサーバーなどの削除を行う。

1.作成したデータベース「クラス名番号-aurora-db」を削除しなさい。[　削除できた　]

2.サブネットグループ「クラス名番号-subg」を削除しなさい。[　削除できた　]

※2まで完全に削除できるまで待ってください。2までの削除が完了しないと3から以降を削除ができません。

3.サブネット「クラス名\_aurora\_sub」を削除しなさい。[　削除できた　]

4.セキュリティグループ「クラス名番号-aurora-sg」を削除しなさい。[　削除できた　]

5.CloudFormationのスタック「クラス名番号」を削除しなさい。[　削除できた　]

6.「End Lab」で終了しなさい。