ECS\_ECR演習1(Cloud9/docker)

クラス名　　　　　No.　　　氏名

次のようなコンテナサービスを利用した環境を構築しなさい。

※Leaner Labで実施すること。

タイムライン が含まれている画像

自動的に生成された説明

■事前設定

コンテナを展開するためのVPCやサブネット、セキュリティグループを作成する。ここでは、 CloudFormationでベースの環境を作成する。CloudFormationによりVPC (クラス名\_ecs\_vpc)、サブネット(クラス名\_pub\_sub、クラス名\_pub\_sub2)、ルートテーブル(ecc\_rt1)、インターネットゲートウェイ(クラス名\_ecs\_igw)、セキュリティグループ(クラス名\_ecs\_websg)を作成する。

1.ecs.ymlファイルを使用して、スタックを作成しなさい。

　スタック名：クラス名　LabName：クラス名　※キーペアは使用できるものを使用する

［　作成できた　］

2.RDSのデータベースを作成する。

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-ecs-sg　　説明：クラス名番号-ecs-sg

VPC：クラス名\_ecs\_vpc インバウンドドルール：タイプ(SSH、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-05eea6b9e0cf048ac ]

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-ecs-subg 説明：クラス名番号-ecs-subg　　VPC：クラス名\_ecs\_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\_pub\_sub、クラス名\_pub\_sub2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-ecs-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.medium

　・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\_ecs\_vpc サブネットグループ：クラス名番号-esc-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-ecs-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)

■Cloud9の設定

1.Cloud9を次の仕様に従い起動しなさい。

　 名前：クラス名番号\_cloud9 説明：クラス名番号\_cloud9 環境のタイプ：新しいEC2インスタンス

インスタンスタイプ：t2.micro プラットフォーム：Amazon Linux 2023

ネットワーク設定 接続：セキュアシェル（SSH） VPC：クラス名\_ecs\_vpc

サブネット：ecc\_pub\_sub

2.Cloud9のコーンソールを開くことができたか確認しなさい。［　確認できた　］

■AuroraDBの設定

1.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-ecs-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-ecs-db.cluster-c59kyyscbffa.us-east-1.rds.amazonaws.com ］

2.Cloud9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　］

3.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　］

4.index.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。

5.index.phpファイルを確認しなさい。

データベース名、エンドポイントが正しいか確認しておくこと。［　確認できた　］

■Dockerコンテナの作成

1.次の内容のDockerfileファイルを作成しなさい。作成は、PCで行い、Cloud9へコピーすること。

|  |
| --- |
| FROM php:8.2-apache  RUN apt-get update && docker-php-ext-install pdo\_mysql mysqli  WORKDIR /var/www/html  COPY index.php index.php |

2.index.phpファイルをCloud9へコピーしなさい。

3.次のコマンドを使用してコンテナイメージを作成しなさい。

|  |
| --- |
| docker build --tag php\_web . |

4.3で作成したイメージを確認しなさい。

|  |
| --- |
| docker images |

5.3で作成したイメージを使用してコンテナを作成しなさい。コンテナ名：クラス名番号sv、ポート番号：80。

|  |
| --- |
| docker run -d -p 80:80 --name クラス名番号sv php\_web |

6.5でコンテナが作成されたか確認しなさい。

|  |
| --- |
| docker ps |

コンテナID：[ 717d24d2ff502f0296c2f88ec63cb728b3d64ebeaf12d05f131ce1c9b76a9ccc ]

7. curlコマンドでindex.phpファイルが表示されるか確認しなさい。

|  |
| --- |
| curl localhost |

［　確認できた　］

|  |
| --- |
| <html>  <head>  <title>AWS DB test</title>  </head>  <body>  <header><h1>商品一覧</h1></header>  <table border="1">  <tr><td>商品名</td><td>カテゴリ</td><td>価格</td></tr>  <tr><td>クリスマスプディング</td><td>ホール</td><td>2000</td></tr>  <tr><td>ブッシュド・ノエル</td><td>シートケーキ</td><td>250</td></tr>  <tr><td>イチゴとシブースドケーキ</td><td>ショートケーキ</td><td>400</td></tr>  </table></body>  </html> |

8.5で作成したコンテナからイメージ(クラス名番号sv\_image)を作成しなさい。

|  |
| --- |
| docker commit クラス名番号sv クラス名番号sv\_image |

［　確認できた　］

9.8で作成したイメージを確認しなさい。

|  |
| --- |
| docker images |

［　確認できた　］

　イメージID：[ ef2e9b9482c5fa4ca62536ef8116db1eba78fae9fa6bdc93bfb2d5ccda76586a ]

ECR\_ECS演習2(ECR/ECSの設定)

クラス名　　　　　No.　　　氏名

次の仕様に従いECRを作成し、コンテナイメージをアップロードしなさい。

■ECR(プライベートリポジトリ)の作成 => ECS

1.次の仕様に従いプライベートリポジトリを作成しなさい。

　リポジトリ名：クラス名番号repo

リポジトリは作成できたか。［　作成できた　］

2.1で作成したリポジトリのURIをコピーしなさい。

　　URI:［ 666629120999.dkr.ecr.us-east-1.amazonaws.com/sk3a04repo ］

3.[プッシュコマンドを表示]をクリックして、イメージファイルをアップロードする手順を確認する（Cloud9で実行する）。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, メール

自動的に生成された説明

4.3で確認した手順に従い、イメージをアップロードしなさい。

　　イメージはリポジトリにアップロードできたか。［　確認できた　］

次の仕様に従いECSを設定しなさい。

■クラスターの作成

1.クラスターとは：［ タスクとサービスを実行する基盤のこと ］

2.次の仕様に従い、クラスターを作成しなさい。

　　クラスター名：クラス名番号Cluster デフォルトの名前空間：クラス名番号cluster

　　インフラストラクチャ：AWS Fargate (サーバーレス)

3.2で作成したクラスターを確認しなさい。［　確認できた　］

■タスク定義の作成

1.タスクとは：［ タスク定義に基づいて起動するコンテナの集まりのこ　］

2.次の仕様に従いタスク定義を作成しなさい。

　タスク定義ファミリー名：クラス名番号\_tasks 起動タイプ：AWS Fargate

　タスクロール：LabRole タスク実行ロール：LabRole

　・コンテナの詳細

　　名前：websv イメージURI：クラス名番号repo(ECRのプライベートリポジトリ)のURI

3.2で作成したタスク定義を確認しなさい。［　確認できた　］

■サービスの作成

1.サービスとは：［ クラスター内で実行中のタスクを管理す　］

2.次の仕様にしたがい「クラス番号Cluster」にサービスを作成しなさい。

　・デプロイ設定

　　ファミリー：クラス名\_tasks サービス名：クラス名番号serv 必要なタスク：2

　・ネットワーキング

　　VPC：クラス名\_ecs\_vpc　　　サブネット：ecc\_pub\_sub、 ecc\_pub\_sub2

　　セキュリティグループ：既存のセキュリティグループを使用(クラス名\_ecs\_websv)

・ロードバランサー

　　ロードバランサーの種類：Application Load Balancer ロードバランサー名：クラス名番号-elb

ターゲットグループ：新しいターゲットグループの作成(ターゲットグループ名：クラス名番号-tg)

3.2のサービスが起動して、正常に動作しているか確認しなさい。

　　［　確認できた　］

■動作確認

Webページを作成して、ECSで起動したコンテナが正常に起動したか確認しなさい。

1.サービスで指定したコンテナが起動しているか確認しなさい。

　ブラウザで［サービス＞設定とネットワーク］のDNS名を指定してWebページが表示されるか確認しなさい。

　DNS名：［ sk3a04-elb-219516912.us-east-1.elb.amazonaws.com ］

テキスト, テーブル

自動的に生成された説明

■提出

　このファイルを「クラス名番号名前\_ECS\_ECR」、index.phpを「クラス名番号名前\_index.php」、Dockerfileを「クラス名番号名前\_Dockerfile」で保存しsv23へ提出しなさい。

■削除処理

次の手順で削除処理を行いなさい。

1.ECSの作成したサービス(クラス名番号serv)を削除しなさい。

2.ECSの作成したタスク(クラス名番号\_tasks)を削除しなさい。

3.ECSの作成したクラスター(クラス名番号Cluster)を削除しなさい。

4.ECRの作成したリポジトリ(クラス名番号repo)を削除しなさい。

5.作成したデータベース(クラス名番号-ecs-db)を削除しなさい。（インスタンス→クラスターの順番）

6.作成したスタック(クラス名)を削除しなさい。