クライアント企業様

「生産部品の在庫管理システム」AWS環境構築手順書

| 作成者 | 三好　順樹 |
| --- | --- |
| 作成日 | 2024年12月8日 |
| 最終更新日 | 2024年12月8日 |

※以下すべて作成途中で概要のみ

目次

本書の目的

本手順書では「生産部品の在庫管理システム」をAWS App Runner上に構築し、ネロギング・モニタリングの設定を実施し、またコスト管理の設定することを目的としています。

環境情報

構築作業に必要な環境を示します。

利用するサービス

・GIt/Github

・Docker

・AWS

AWSリージョン

App Runner

CloudTrail

CloudWatch

AWS Budget

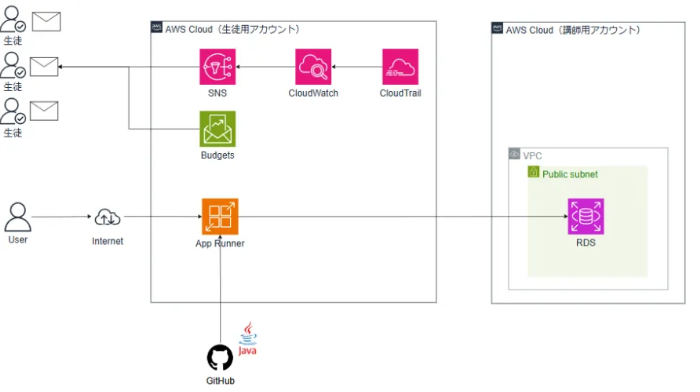
IAM

hogehoge

・デプロイ対象のリポジトリ

https://github.com/digital-ojt/infra-course-drone/

AWSリソース構成図



命名規則

本プロジェクトでは以下のような命名規則に従って構築します。

**リソースに対する命名規則**

| **要素** | **詳細** |
| --- | --- |
| sysname | 固定値としてdroneを使用 |
| env | 固定値としてdevを使用 |
| user | (個別) I AM user名 |
| service | (個別) 対象サービス |
| 予備 | 一意にできない場合に使用 |

例. drone-dev-yourname-apprunner

**タグに対する命名規則**

| **キー** | **値** |
| --- | --- |
| Cost | drone\_IAM user名 |
| Project | (固定)infra-course-drone |
| Name | リソース名 |
| createdBy | (個別)IAM user名 |

例.Cost:yourname

アプリ構築

※以下、内容は整理し、画像キャプチャもつける

Java 11

Spring 2.7.15

アプリ起動

$ docker-compose up

http://localhost:8080/ を開く

デプロイ： ECS Fargate by App Runner の構築

App Runnerはフルマネージド型のコンテナアプリケーションサービスであり、簡単に ECS・Fargate で Web アプリを構築できる。 AWSの管理画面を開いて、App Runnerで「サービスの作成」をする。

ソースおよびデプロイ

「ソースコードリポジトリ」を選択してGitHubと連携する

GitHub連携の「ソースディレクトリ」は「/dev」を指定する

デプロイ設定

「自動」を選択する

構築を設定

ランタイムは「Corretto 11」を選択する

AppRunner 上ではビルドコマンドとスタートコマンドを設定する。

$ mvn clean package

$ java -Xms256m -jar target/dev-0.0.1.jar

ロギングとモニタリングの設定

使うツールはCloudWatch,Cloud Watch Alerm, Cloud Tail

・ダッシュボードを作成

・AppRunner>メトリクス>ダッシュボード追加から一気に追加

・一時停止ログの取得設定

・SNSの設定

・Cloud Watch Alermの設定

インフラコスト管理の設定

・予算を作成して予算超過した場合に、設定したアドレス宛にアラートが通知される

・Cost Explorer でコストを監視、レポート・CSVデータで出力する