

Cloud Runの紹介



Press Space for next page →





自己紹介

- 📝 飯野陽平 ([wheatandcat](https://www.wheatandcat.me/))
- 🏢 フリーランスエンジニア (シェアフル株式会社CTO)
- 💻 Blog: <https://www.wheatandcat.me/>
- 🛠️ 今までに作ったもの
 - [memoir](#)
 - [ペペロミア](#)
 - [Atomic Design Check List](#)

Cloud Runとは

- Cloud Run は、スケーラブルなインフラストラクチャ上でコンテナを直接実行できるマネージド コンピューティング プラットフォーム。
- コンテナ イメージをビルドしてデプロイして使用できる。
- 構成、スケーリングは設定で即座に変更可能。

実行形式

Cloud Runには以下の2つの実行形式がある

Cloud Run Service

- 主な用途: APIサーバー、webサイト、イベント処理
- できること
- 自動スケーリング、各種スケーリングの設定

Cloud Run Jobs

- 主な用途: バッチ処理、ツール
- Cloud Run Serviceとの差は以下の通り
 - 設定タイムアウトの時間を最大1時間まで設定可能
 - 最大再試行回数を設定できる
 - 並列処理

Cloud Runの使い方

Cloud Runでは以下の2つの方法でコンテナイメージの作成が可能。

- **Dockerfile** を使用する
- **Buildpacks** を使用する

Buildpacksとは

Buildpacksは、ソースコードから読み取りベストプラクティスに則ってDockerfileを書かずにイメージを生成してくれるツールで、現状だと以下のプラットフォームで対応している。

- Go
- Node.js
- Python
- Java
- .NET Core

[Learn More](#)

Cloud Run Serviceをデプロイしてみる①

以下のチュートリアルをベースに、デプロイを紹介

- [Cloud Run に Go サービスをデプロイする](#)

Cloud Run Serviceをデプロイしてみる②

以下のコマンドでイメージを作成

```
$ gcloud builds submit --pack image=IMAGE_URL
```

上記で、GCPの**Artifact Registry**にイメージが作成されるので、以下のコマンドでCloud Runにデプロイ

```
$ gcloud run deploy $SERVICE_NAME \  
    --project=$PROJECT_ID \  
    --image=$IMAGE_URL \  
    --region=$REGION
```

[Learn More](#)

環境変数の設定

環境変数の設定はコンソールから設定可能

- 環境変数の設定

コンテナ

変数とシークレット

接続

セキュリティ

環境変数

名前 1

APP_ENV

例: ENV

値 1

例: prod

名前 2

GCP_LOCATION_ID

例: ENV

値 2

例: prod

+ 変数を追加

環境変数で機密情報を設定する場合はSecret Managerの値をマウントできる。

- シークレットを使用する

継続的デプロイの方法

GitHub Actionsを使用して実装したので紹介

■ Cloud Runへ移行

- <https://github.com/wheatandcat/memoir-backend/pull/117/files>

Cloud Run Jobsを試してみる①

以下のチュートリアルをベースに紹介

- [Cloud Run で Go ジョブをビルドして作成する](#)

Cloud Run Jobsを試してみる②

以下のコマンドでイメージをデプロイ

```
$ gcloud builds submit --pack image=gcr.io/PROJECT_ID/logger-job
```

以下のコマンドで作成したコンテナを使用してジョブを作成する

```
$ gcloud beta run jobs create $JOB_NAME \  
  --image gcr.io/PROJECT_ID/logger-job \  
  --tasks 50 \  
  --set-env-vars SLEEP_MS=10000 \  
  --set-env-vars FAIL_RATE=0.5 \  
  --max-retries 5 \  
  --region REGION
```

以下のコマンドで実行

```
$ gcloud beta run jobs execute $JOB_NAME
```

Cloud Run Jobsを試してみる③

結果は、コンソールで確認できる

Cloud Run

ジョブの実行

プレビュー

削除

!

ジョブ: job-quickstart

実行日: 2022/06/19 14:18:08 実行 ID: リージョン: us-central1

タスクの概要 (50) :

49 成功

1 が失敗しました

0 実行中

タスク

ログ

設定

YAML

フィルタ

タスクのフィルタ

?

●

ToDo リスト

↑

0

1

2

3

4

5

6

7

8

9

10

11

12

13

再試行数: 3

再試行数: 4

再試行数: 1

再試行数: 5

再試行数: 3

再試行数: 2

再試行数: 2

再試行数: 2

✓

タスク 0

情報

ログ

全般

開始時刻

2022/06/19 14:18:15

終了時間

2022/06/19 14:18:27

再試行の回数

0

最後の終了コード

0

並列実行

以下の記事を元に紹介

- [Cloud Run jobs を解説する](#)

Cloud SchedulerにJobを設定する

以下のチュートリアルをベースに紹介

- [スケジュールに従ってジョブを実行する](#)

URLには以下を設定する

```
https://REGION-run.googleapis.com/apis/run.googleapis.com/v1/namespaces/PROJECT-ID/jobs/JOB-NAME:run
```

まとめ

- GCPの推しているサービスなだけあってスキの無い作りだった
- **Cloud Run Jobs**の登場でバッチ処理もカバーできるようになった
- **Cloud Run Jobs**は、まだ**プレビュー**の状態なので注意

ご清聴ありがとうございました