

# CloudflareとGoogle Cloud Runによる ML・推論実行アプリ制作

## 自己紹介

### 1. ITエンジニアではありません

- ・コロナ禍の在宅ワーク期間、通勤の時間が浮いたので、趣味でPythonの学習を開始。
- ・セキュリティとか、ネットワークとか、今だによく分からぬ。
- ・新規事業に於けるPoCレベルのものは自作可能。

### 2. データサイエンティストではありません

- ・Kaggleは、コロナ禍の在宅ワーク期間、時間があったので趣味の一環として遊びました。
- ・数学を勉強したのは、大学院時代が最後(20年以上前)で、高度な数学的な素地がある訳ではない。
- ・半導体技術者時代に「シックスシグマ」に触れて以来、データの活用自体には興味があった。

### 3. 主なキャリアパスは、半導体開発設計、コンサルタント、新規事業開発とマーケティングです

- ・グローバルメーカーで、開発・設計・品質管理・生産技術管理等に従事。
- ・コンサルタント時代は、金融業界、保険業界、メディア業界、IT業界で、様々なPJに従事。
- ・直近の10年以上は、新規事業開発(本業・副業含む)とマーケティング、海外市場開拓に従事。

## なぜ、Cloudflareに目を付けたか？

### 1. そもそも、僕がしたいことは？

- ・新規事業開発のPSFフェーズに於けるPoCを自作したい。（このフェーズでは、出戻りが多いため）
- ・できれば、無料(or低料金)で、爆速で制作し、想定ジョブに対する仮説検証を行いたい。

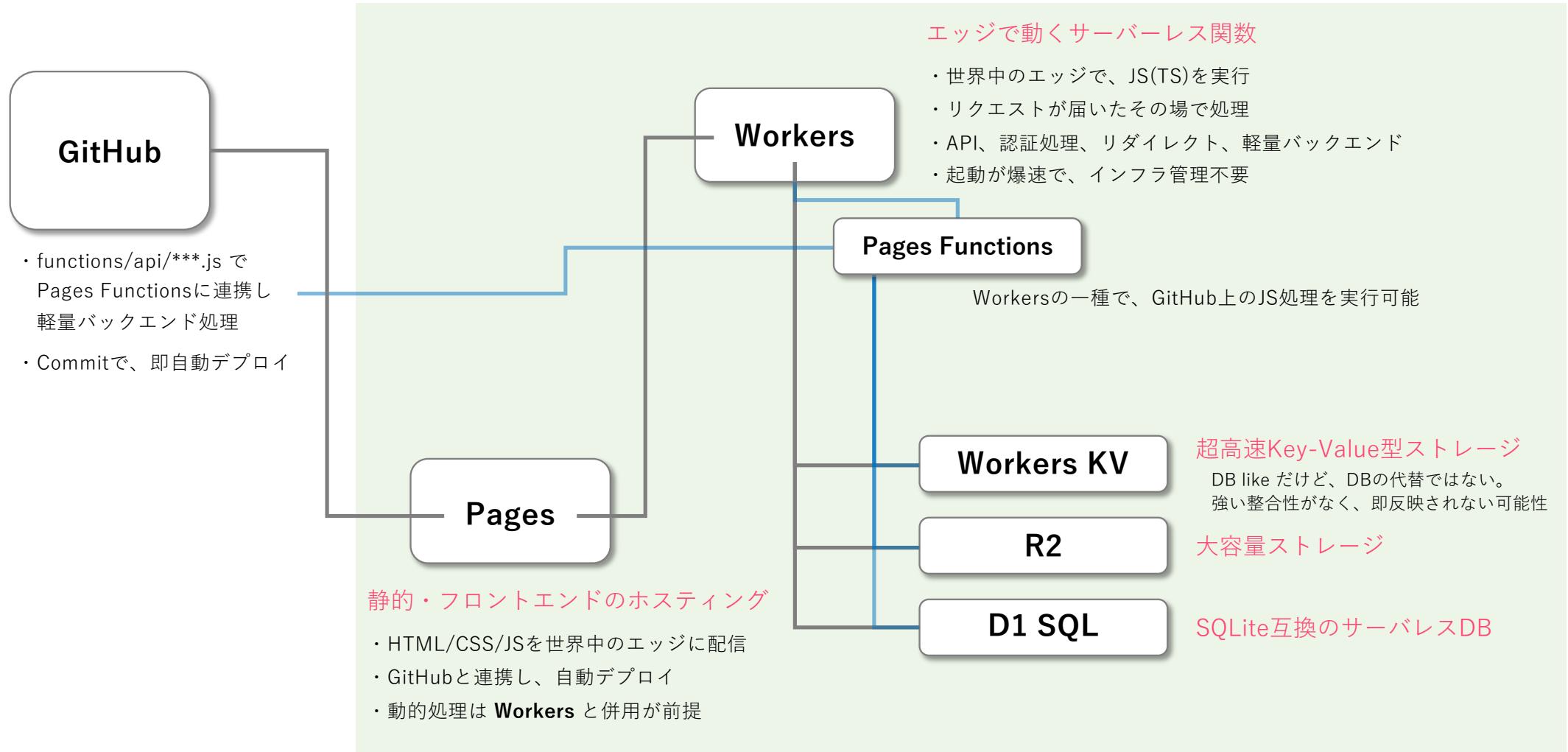
### 2. Cloudflareとは？

- ・無料でもPoC前提の「フロントエンド、軽量バックエンド、簡易DB、ストレージ」を単体で完結できる。
- ・GitHubと連携可能で、Commitしたら、すぐに自動デプロイ。
- ・Zero Trust等の認証機能もある。
- ・生成AI API連携ももちろん可能。

### 3. デメリット/制約条件は？

- ・Pythonは正式サポートされておらず、JavaScript/TypeScriptが前提。
- ・軽量バックエンドなので、機械学習関連のタスクは不可（Google Cloud Runで処理するAPI連携は可能）

## Cloudflareのイメージ



## なぜ、Google Cloud Runに目を付けたか？

### 1. そもそも、僕がしたいことは？

- Pythonによるテーブルデータの前処理や、機械学習推論タスク、Excel業務自動化を実行できるWebアプリの制作。  
(Cloudflareは軽量バックエンドのみであり、Pythonが正式サポートされていない)
- Streamlitでも良いが、もっと柔軟なUI設計を実現したい。(できれば無料 or 低料金で、バイブルコーディングしたい)

### 2. Google Cloud Runとは？

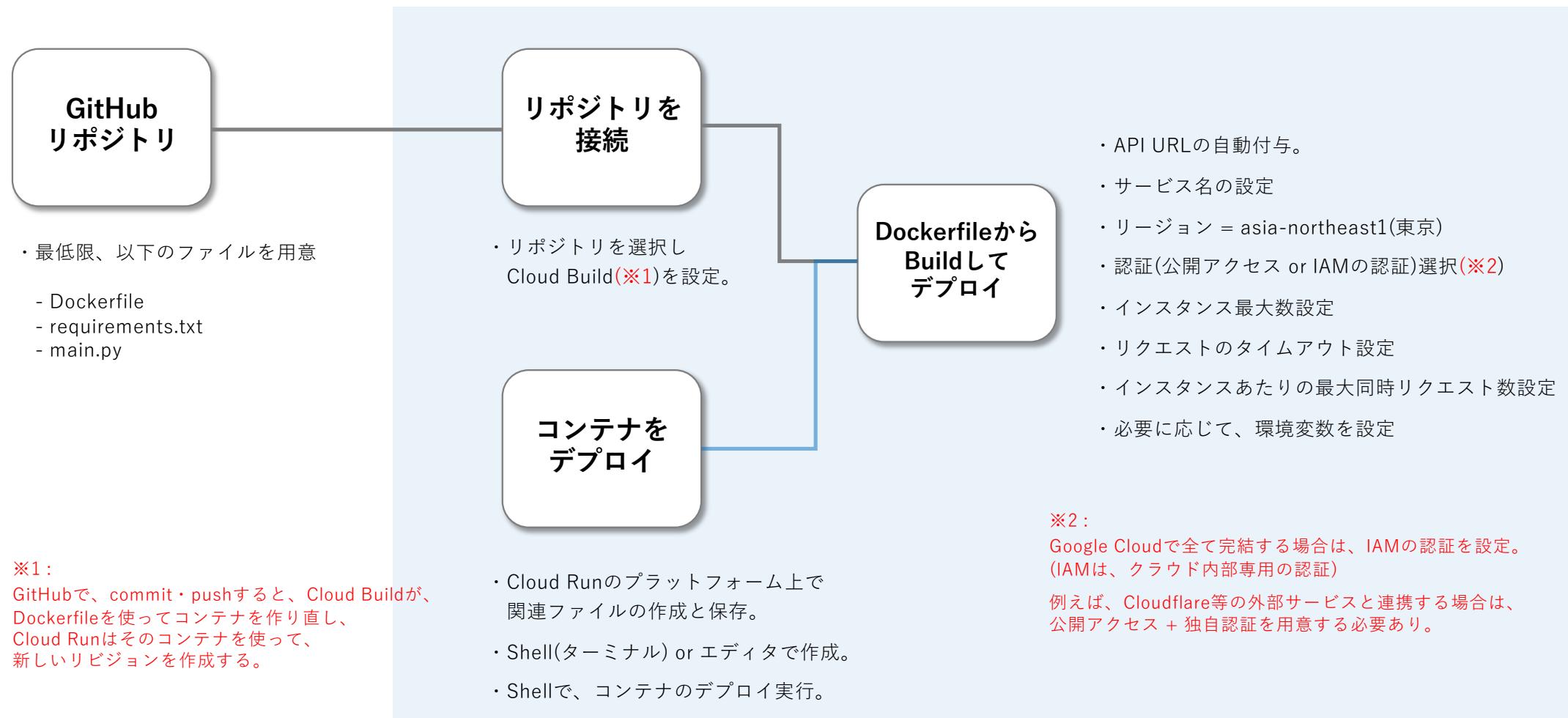
- Dockerコンテナを、URL付のWebサービスとして動かしてくれる「サーバ運用不要の実行基盤」(DockerをURLに)。
- HTTPで呼べるサービス (APIサービスに固定のURLが付与されるため、Cloudflare等から呼びやすい)。
- GitHub連携でデプロイしても、裏で「リビジョン」が作成され、履歴管理が可能(即ロールバック可能)。
- FastAPIをそのまま本番公開できるため、NotebookからAPIへの橋渡しが簡単)
- Dockerで実行環境を固定できるため、依存関係が壊れにくい。
- 実行時間、同時実行数、インスタンス数を設定でき、サーバ管理が不要で、自動スケール運用が楽。

## なぜ、Google Cloud Runに目を付けたか？

### 3. デメリット/制約条件は？

- ・コールドスタートで、Cloud Runは、リクエストが無いとコンテナが止まる。(低レイテンシAPIには不向き)
- ・Cloud Runはデフォルトで、1コンテナで同時に最大80リクエスト(メモリ大量消費等が同時に来ると課金地獄)(concurrency=1~5にすると、同時アクセス増加時には待ち行列発生。リアルタイム大量処理は不向き)
- ・実行時間制限(Timeout)が最大60分、レコード数が多く特徴量が多い、複数モデルの処理になると現実的な壁に。
- ・CPU=最大8vCPU、メモリ=最大32GBで、巨大モデルの処理になると限界が…。
- ・ステートレス前提なので、ローカルファイルは消え、メモリキャッシュも保証されない。
- ・Cloud Run単体では、DBもなくストレージもないため、ファイルの保存は不可(単体完結は無理)。
- ・Bearer Tokenは可能で手軽だが、エンタープライズでは、認証・権限の追加設計が必要と思う。

## Google Cloud Run のイメージ



## テーブルデータの推論実行スキーム例(概要)

Google Cloud Run公開アクセス許可 + API\_TOKEN認証。  
(Cloudflare、Cloud RunのAPI\_TOKENは同じものを設定)

### ③ API TOKEN認証、モデルload、前処理、予測

- ・公開アクセスを許可  
(API-TOKEN認証処理)
- ・リージョン設定
- ・Timeout短め、Concurrency低め
- ・APIのURL発行
- ・対象データの前処理と予測のみ実行

- ・モデル学習用データを用意。
- ・Scikit-learnやGBDT等の学習済みモデルを用意。  
(model.joblib)

### GitHub リポジトリ-1 Cloud Run用

リポジトリから継続的にデプロイ  
(Dockerfileを使用してビルド)

model.joblib

- ・Google Cloud Run上で動作するFastAPI アプリ
- ・コンテナ用DockerFileや、ML推論タスク(同期処理)を実行するmain.py等を用意。  
(API TOKENの認証処理有)
- ・Google Colabで作成した学習済みモデルも格納

### Google Cloud Run

②  
データ転送  
(プロキシ)

④  
推論データ返却  
(HTTPレスポンス/Fast API)



管理者

- ・学習済みモデルを更新する時は、新しいmodel.joblibをリポジトリへ

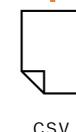
### GitHub リポジトリ-2 Cloudflare用

Pages&Workersでデプロイ

- ・Cloudflare Pages、Pages functions用のリポジトリ
- ・主な機能は、
  - csvファイルのアップロード用UI
  - pages functions 経由で、Google cloud runに転送(プロキシ)
  - cloud runから返却されたcsvをブラウザダウンロード
  - API TOKENの認証処理

### Cloudflare Pages Pages functions

①  
アップロード



ユーザー

- ・公開されたCloud Run URLを設定
- ・pages functionsからcsvを、Cloud Runへ転送(APIプロキシ)。
- ・返却されたcsvをブラウザに返却
- ・API URL(Cloud Run)とAPI-TOKENを環境変数として管理

⑤  
pages functions自動でダウンロード

## Google Colaboratory で学習済みモデルを用意

マイド... > Google Cloud R... > 学習済みモデル生... ▾

種類 ユーザー 最終更新 ソース

名前	オーナー	更新日時	ファイルサイズ	⋮
model.joblib	自分	12月27日	2 KB	⋮
train_df.csv	自分	2023/12/30	40 KB	⋮
学習済みモデル生成用_scikit-learn_Prediction of sp...	自分	12月27日	42 KB	⋮

学習済みモデル生成用\_scikit-learn\_Prediction of spam with Bayesian model.ipynb

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

コマンド + コード + テキスト ▶ すべてのセルを実行

```
model = Pipeline([
    ("scaler", StandardScaler()),
    ("clf", LogisticRegression(max_iter=2000))
])
```

1

```
model.fit(X_tr, y_tr)
```

1

```
proba = model.predict_proba(X_va)[:, 1]
print("AUC:", roc_auc_score(y_va, proba))
```

AUC: 0.8357366771159874

1

```
joblib.dump(
    model,
    "/content/drive/MyDrive/Google Cloud Run用/学習済みモデル生成用/model.joblib"
)
print("saved to Google Drive")
```

saved to Google Drive