

IT・AI を活用した食品安全情報収集システムに関する実証
事業

B)システム開発

(イ)システム構築ドキュメント

令和7年3月31日

みずほリサーチ&テクノロジーズ株式会社

情報通信研究部

目次

1. システム構築.....	1
1.1. 前提.....	1
1.1.1. AWS CLI のインストール	2
1.1.2. Node.js のインストール.....	2
1.1.3. AWS CDK のインストール及び初期設定	2
1.1.4. Docker のインストール	2
1.2. AWS リソースのデプロイ	3
1.2.1. パラメータ設定.....	3
1.2.2. デプロイ.....	4
1.2.3. Bedrock の設定.....	4
1.3. フロントエンドアプリのデプロイ	7
1.3.1. パラメータ設定.....	7
1.3.2. ビルド	9
1.3.3. デプロイ.....	9
1.4. 動作確認.....	9

1. システム構築

本案件で構築したシステムを AWS 上に構築（デプロイ）する手順を記した。

システム構築には、AWS から提供される AWS Cloud Development Kit (以降 CDK)使用しており、多くのリソースはコマンド実行でデプロイが可能となる。一部は AWS マネジメントコンソールから手動で操作を行う。

1.1. 前提

1.2 節以降の操作に必要な前提や準備事項を以下に示した。なお、以下の手順におけるコマンド実施箇所は Windows のターミナル上で行い、図 1-1 を前提とする。AWS の操作に使用する AWS アカウント及びアカウントに対する権限付与は実行されているものとする。

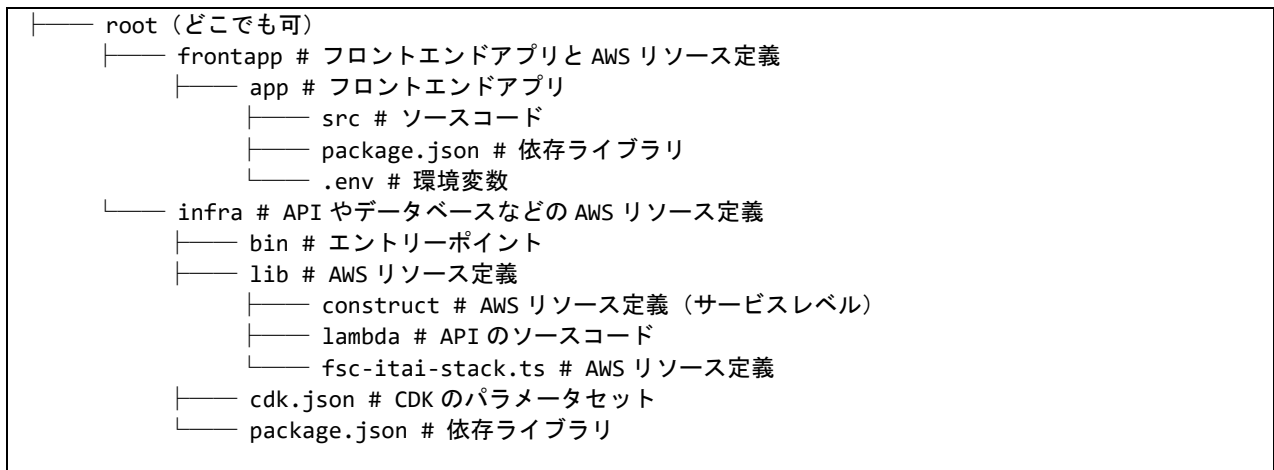


図 1-1 ディレクトリ構成

開発環境で使用しているソフトウェアやライブラリのバージョン情報を表 1-1 に示した。以上と記載のあるもの以外は同じバージョンのものを使用することを前提としている。

表 1-1 ソフトウェアバージョン

ソフトウェア名	バージョン
AWS CLI	2.8.3
Node.js	18.20.7 (バージョン 18 台の LTS)
AWS CDK	2.162.1
Rancher Desktop	1.16.0 以上

1.1.1. AWS CLI のインストール

AWS CLI の 2.8.3 をインストールする。

一般的なインストール手順で問題ないため、以下のサイトなどを参考にインストールを行う。

また、以降の CDK 実行などでは、認証情報の設定が適切にされていることを前提としているため、参考サイトに記載されている方法などで、環境に応じた設定を行うこと。

参考：

https://docs.aws.amazon.com/ja_jp/cli/latest/userguide/getting-started-version.html

<https://dev.classmethod.jp/articles/install-aws-cli-on-the-windows-11-terminal-at-hand-and-execute-aws-cli-commands/>

1.1.2. Node.js のインストール

AWS CDK の動作に必要なため、Node.js (バージョン 18 台の LTS を利用) をインストールする。

一般的なインストール手順で問題ないため、公式サイトから指定バージョンのインストーラーをダウンロードし、以下のサイトなどを参考にインストールを行う。

参考：

<https://nodejs.org/ja/download>

<https://qiita.com/yasushi-jp/items/fd885d38f11d95e81328>

1.1.3. AWS CDK のインストール及び初期設定

以下のコマンドで AWS CDK をインストールする。

```
npm install -g aws-cdk@2.162.1

# インストール完了後に以下を実行し、バージョン番号が表示されればインストール完了
cdk -version

# インストール完了後、以下を実行し、cdk の初期設定を行う。
# profile は AWS CLI の認証情報をデフォルトプロファイルとしていない場合に指定が必要
cdk bootstrap aws://<AWS アカウント番号>/us-east-1 --profile <プロファイル名>
```

1.1.4. Docker のインストール

AWS 環境上での Python 動作環境を構築するため、Docker を利用できる状態にする。

Windows 環境の場合、WLS2 及び Rancher Desktop を導入し、Docker が利用できる環境を構築する。一般的なインストール手順で問題ないため、以下のサイトなどを参考にインストールを行う。インストール完了後、RancherDesktop は起動しておく。

参考：

<https://rancherdesktop.io/>

<https://qiita.com/IoriGunji/items/f2a86ffdc629b6edc549>

Rancher Desktop のインストール

RancherDesktop の起動

1.2. AWS リソースのデプロイ

CDK を利用して API やデータベースなどの AWS リソースをデプロイする。

1.2.1. パラメータ設定

infra/cdk.json の以下の箇所を変更する。

```
# AWS アカウント、アクセスを許可する IP アドレスを設定
"dev": {
  "account": <AWS アカウント番号>,
  "allowAddresses": [          ←IP アドレスのリスト（初期設定は開発時に設定したもの）
    "114.141.124.142/32",
    "114.141.123.80/32",
    "114.141.120.205/32",
    "114.141.120.206/32",
    "103.157.243.124/32",
    "103.157.243.125/32",
    "103.157.243.126/32",
    "103.157.243.127/32",
    "103.157.243.228/32"
  ]
}

# CDK を実行する認証情報を設定
"profile": <プロファイル名>
```

1.2.2. デプロイ

以下のコマンドを実行する。許可を求められる場合、画面の指示に従って許可する。

```
cd infra

# 依存ライブラリのインストール
npm install

# CDK でのデプロイ
cdk deploy
```

1.2.3. Bedrock の設定

LLM を利用するために、AWS マネジメントコンソールで Bedrock のモデルを使用できるように設定する。

A) AWS マネジメントコンソールから Bedrock の画面に遷移

Bedrock の設定手順を図 1-2～図 1-5 に示した。

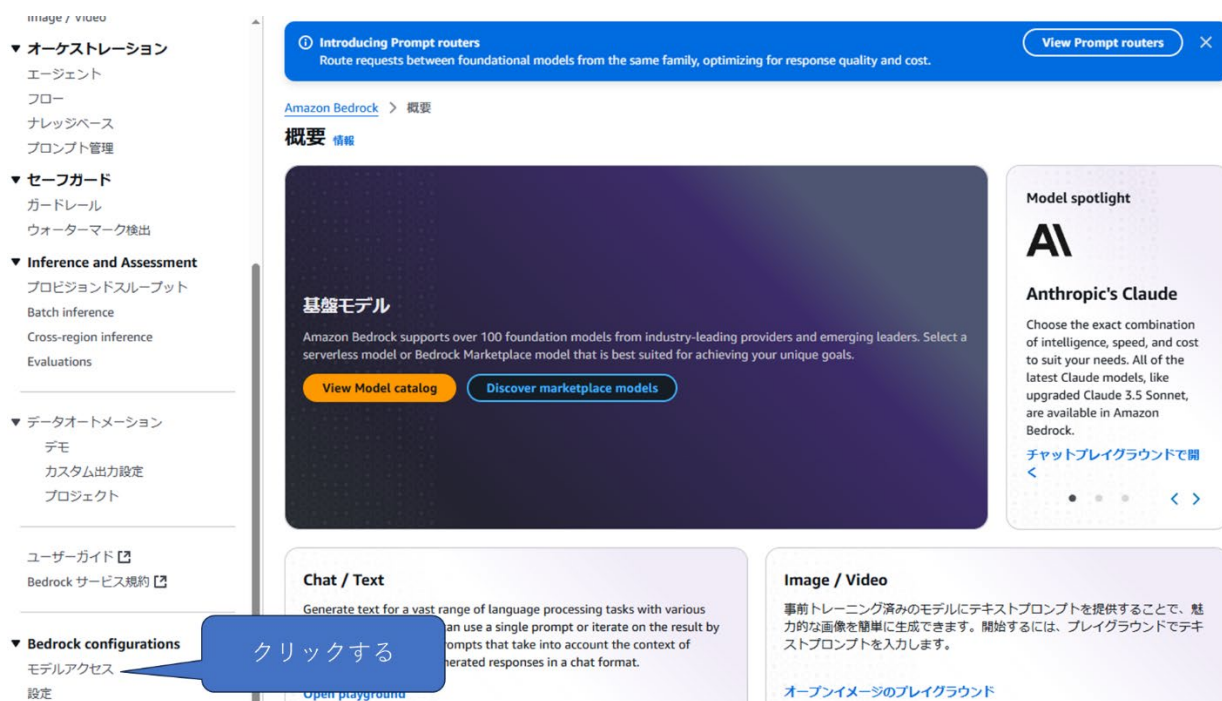


図 1-2 Bedrock①

- ▼ **プレイグラウンド**
 - Chat / Text
 - Image / Video
 - ▼ **オーケストレーション**
 - エージェント
 - フロー
 - ナレッジベース
 - プロンプト管理
 - ▼ **セーフガード**
 - ガードレール
 - ウォーターマーク検出
 - ▼ **Inference and Assessment**
 - プロビジョンドスループット
 - Batch inference
 - Cross-region inference
 - Evaluations
 - ▶ **データオートメーション**
- ユーザーガイド

Amazon Bedrock > モデルアクセス

モデルアクセスとは?

Bedrock を使用するには、正しい IAM 許可 を持つアカウントユーザーが利用可能な Bedrock 基盤モデル (FM) へのアクセスを有効にする必要があります。Bedrock モデル利用規約 のすべての Bedrock FM を表示します。

[モデルアクセスを変更](#)

Visit [Amazon Bedrock Quotas](#) for a quick guide to the default quotas and limits that apply to Amazon Bedrock.

ベースモデル (47)
Not seeing a model you're interested in

1. Claude 3.5 Sonnetで検索

Find model

2. v2がついていないほうの「リクエスト可能」をクリック

3. 「モデルアクセスをリクエスト」をクリック

モデル	アクセスのステータス
▼ Anthropic (2)	2/2 access granted
Claude 3.5 Sonnet v2 Cross-region inference	アクセスが付与されました
Claude 3.5 Sonnet	リクエスト可能

リクエスト可能

このモデルに対するアクセスをリクエストできます。アクセスが付与され、Bedrock でモデルの使用を開始すると、請求が開始されます。[モデルアクセスをリクエスト](#)

図 1-3 Bedrock②

<input type="checkbox"/>	DeepSeek-R1 Cross-region inference	リクエスト可能	テキスト	EULA
<input type="checkbox"/>	▼ Meta (9)	0/9 access granted		
<input type="checkbox"/>	Llama 3.3 70B Instruct Cross-region inference	リクエスト可能	テキスト	EULA
<input type="checkbox"/>	Llama 3.2 1B Instruct Cross-region inference	リクエスト可能	テキスト	EULA
<input type="checkbox"/>	Llama 3.2 3B Instruct Cross-region inference	リクエスト可能	テキスト	EULA
<input type="checkbox"/>	Llama 3.2 11B Vision Instruct Cross-region inference	リクエスト可能	テキストとビジョン	EULA
<input type="checkbox"/>	Llama 3.2 90B Vision Instruct Cross-region inference	リクエスト可能	テキストとビジョン	EULA
<input type="checkbox"/>	Llama 3.1 70B Instruct Cross-region inference	リクエスト可能	テキスト	EULA
<input type="checkbox"/>	Llama 3.1 8B Instruct Cross-region inference	リクエスト可能	テキスト	EULA
<input type="checkbox"/>	Llama 3 8B Instruct	リクエスト可能	テキスト	EULA
<input type="checkbox"/>	Llama 3 70B Instruct	リクエスト可能	テキスト	EULA
<input type="checkbox"/>	▼ Mistral AI (4)	0/4 access granted		
<input type="checkbox"/>	Mistral 7B Instruct	リクエスト可能	テキスト	EULA
<input type="checkbox"/>	Mixtral 8x7B Instruct	リクエスト可能	テキスト	EULA
<input type="checkbox"/>	Mistral Large (24.02)	リクエスト可能	テキスト	EULA
<input type="checkbox"/>	Mistral Small (24.02)	リクエスト可能	テキスト	EULA
<input type="checkbox"/>	▼ Stability AI (1)	0/1 access granted		
<input type="checkbox"/>	SDXL 1.0	リクエスト可能		

クリックする

キャンセル

図 1-4 Bedrock③

1.3. フロントエンドアプリのデプロイ

フロントエンドアプリのビルド及び AWS 上へのデプロイを行う。

1.3.1. パラメータ設定

frontapp/app/.env の以下の箇所を変更する。

設定する値は、1.2 節で構築されたリソースの値となるため、AWS マネジメントコンソールから各サービスの値を取得し、転記する。

```
VITE_API_ENDPOINT=①
VITE_USER_POOL_ID=②
VITE_USER_POOL_CLIENT_ID=③
VITE_IDENTITY_POOL_ID=④
```

A) AWS マネジメントコンソールから API Gateway の画面に遷移

遷移画面を図 1-6、図 1-7 に示した。



図 1-6 API Gateway①



図 1-7 API Gateway②

B) AWS マネジメントコンソールから Cognito の画面に遷移

Cognito の設定方法を図 1-8～図 1-14 に示した。



図 1-8 Cognito①



図 1-9 Cognito②



図 1-10 Cognito③



図 1-11 Cognito④

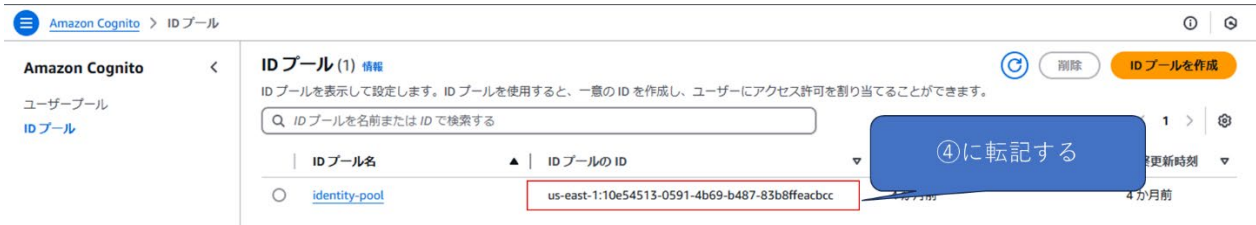


図 1-12 Cognito⑤

1.3.2. ビルド

以下のコマンドを実行する。

```
cd frontapp/app

# 依存ライブラリのインストール
npm ci --loglevel verbose --legacy-peer-deps

# CDK でのデプロイ
npm run build
```

1.3.3. デプロイ

ビルド結果のファイルを S3 のバケットにアップロードする。

バケット名 : 「dev-front-app-bucket」

対象 : frontapp/app/dist/index.html、frontapp/app/dist/assets

1.4. 動作確認

構築した環境の確認のため、フロントエンドアプリへのアクセスを行う。

A) AWS マネジメントコンソールから Cloudfront の画面に遷移

Cloudfront の画面遷移を図 1-15 及び図 1-16 に示した。



図 1-13 Cloudfront①



図 1-14 Cloudfront②

- B) コピーした「ディストリビューションドメイン名」に Web ブラウザでアクセスし、ホーム画面が表示されることを確認

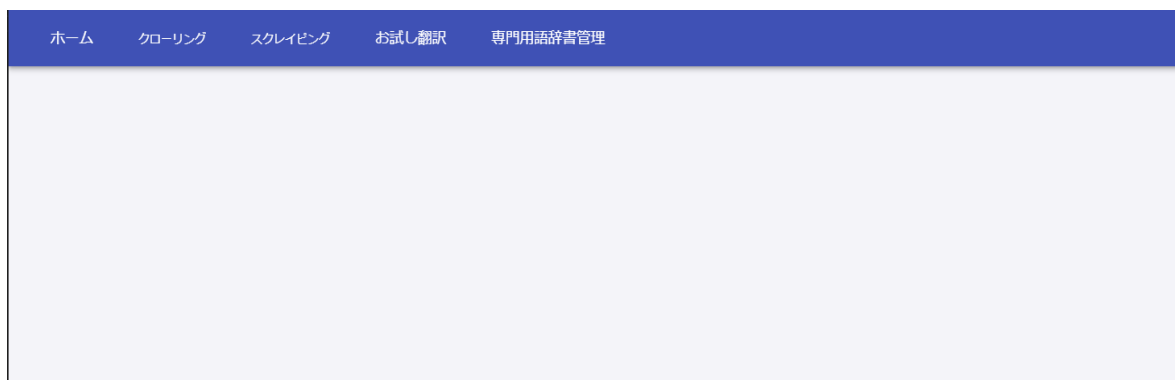


図 1-15 フロントエンドアプリ画面①

- C) お試し翻訳が機能することを確認

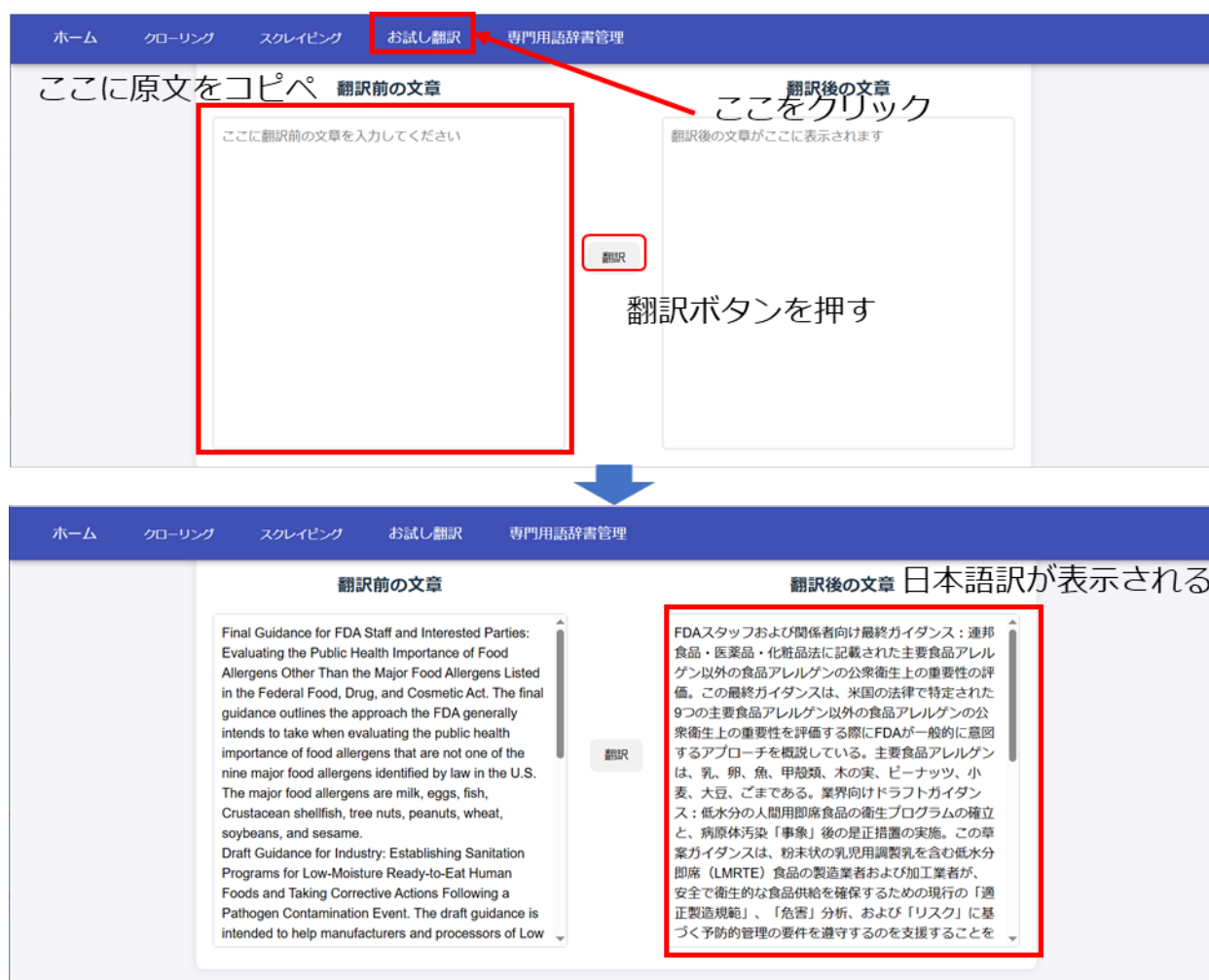


図 1-16 フロントエンドアプリ画面②