

**おうちCO-OP在庫管理システムを作ってみた（途中）**

# 要望

- おうちCOOPで届いた物で、いま何が残っているかを簡単に知りたい
- 冷蔵庫をいちいち漁るのはめんどくさい
- どうにかして

# 初期構想

- Webサイトを色々見たけどデータをAPIとかで抜いてくるのはできなさそう
- お届け商品のご案内メールが届くのでこれをどうにかできないか
- ご案内メールをAWSに転送してゴニョニョすればできるんじゃないか
  - AWSならすべてを解決してくれる、きっと

# 検討

- 自分に届いたメールをシステムに転送？
  - [Amazon WorkMail](#)なら独自のメールアカウントを発行できそう
    - ただし1ユーザー辺り 4 USD/Month 掛かる
  - [Amazon SES](#)にメール受信する仕組みがありそう
    - これを試してみる
- 独自ドメインが必要
  - SESでメールを受け取るには独自ドメインが必要そう
  - Webページも公開する予定なのであった方が便利そう
  - Route53でゴニョニョすれば簡単にドメインが取得できる
  - 取得した (.netで年1100円くらい)

# Amazon SESでメール受信する

- Email receivingという仕組みがある
- 特定のメールアドレスに届いたメールに対して、処理を行うことができる
  - ドメインでも良いっぽいけど試していない
  - S3に置いたりlambdaをキックしたり
  - 最終的にはlambdaで捌くけど、一旦S3に置くようにした

最終的に、自分のメールアドレスに届いたご案内メールをシステムのアドレスに転送するように設定して、それがSES経由でS3のバケットに置かれることを確認した

# S3に置かれたメールから商品情報を取得する

- S3に置かれる **メールファイル** は、メールヘッダなどが含まれる
- 本文はbase64でエンコードされてる、とか細々したルールがある
  - そういえばそんなだった
- pythonの **email** パッケージを使用して日本語文章を取得
- そこからは力技で商品情報を抜き出していく

```
■110:産地指定チリ産塩銀鮭切身（甘口）  
  4切（240g）  
  価格:520円x1
```

```
■${商品コード}:${商品名}  
  4切（240g） # この行は無視  
  価格:${価格}円x${個数}
```

# (余談) GitHub Pagesにスライドを公開する

- いま見ているこのスライドのこと
- GitHub Actionsで.mdから.pdf、.htmlへ変換
  - [基ファイル](#)はこれ
- それらを GitHub Pagesに公開するようにした
- と、偉そうに書いているけど、実際はいくつかのactionsを組み合わせただけ
- [KoharaKazuya/marp-cli-action](#)
- [peaceiris/actions-gh-pages](#)
- こんな[定義ファイル](#)でOK

# (余談) GitHub Pagesでカスタムドメインを使用する

- これも [GitHubの機能](#) にある
- 自分でやったのは、Route53にカスタムサブドメインを追加しただけ

```
docs.pug89.net  CNAME  シンプル - m-namiki.github.io
```



ここまでが連休でやったこと

# これからやること

- 商品情報をDBに登録する
  - RDS or DynamoDB
  - lambdaからRDSを触るのはアンチパターンと言われているが、どうせ自分用だから良いかな
  - DynamoDBは触ったことないのでそっちでやってみるのも良いかな
  - .devcontainerでDynamoDBとpythonのdocker-composeを書いてみたけど起動しない
    - dynamodb-localはちゃんと起動するんだけどpython側が謎の死亡を繰り返す
- 在庫を表示する画面を作る
  - TSとか使ってやろうかなと思案中
- 仕事だと使い慣れた言語やらサービスを使う優先度が高いけど、趣味だと触ったことないのを選ぶのが楽しい
  - そしてその分ちゃんと動かなくことが多くて苦しむ