

作りたい機能一覧

解説系の記事

- 赤石基礎論とかの解説記事やバグ、仕様をまとめたもの
- 最終的には**自作のマークダウン**で記事を作れたり自由度が高いため、赤石専用の見やすい記事が作れる。

回路の紹介集(サバイバルから論理回路まで)

- Youtubeなどに上がっている回路の紹介とURLの添付（著作権とかに注意!）
- **検索方法**を強化したい。
 - （関連するアイテム・エンティティで検索,クリーパーを検索条件に入れると、トラップタワー、帯電クリーパー製造機）
- 作り方などの動画に対する、コミュニティを作る
 - 上手く作れなかった人との会話や、詳細な情報の共有に**画像**や**動画**を付けながらやり取りできる。

赤石回路用語辞書

- 現在試作中

Web上で動作する回路シミュレーター

- 記事などにつけて、分かりやすく、楽しく学べるようにしたい

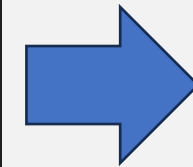
解説系の記事

- 赤石基礎論とかの解説記事やバグ、仕様をまとめたもの
- 最終的には**自作のマークダウン**で記事を作れたり自由度が高いため、赤石専用の見やすい記事が作れる。

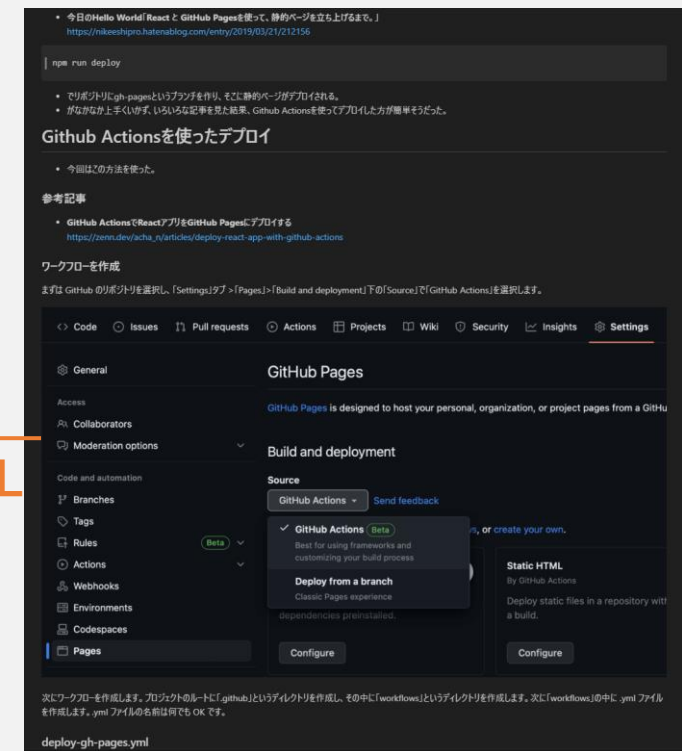
例) マークダウン記法

画像や動画はS3などのストレージに保存してURLを取得、マークダウン記法で書かれた記事を読み込むときにアクセスする。

```
1 # 静的ページの立ち上げ
14
15 # 初めにしようとしたこと(失敗)
16 gh-pages でデプロイしようとした
17
18 ### 参考記事
19 **今日のHello World「React」とGitHub Pagesを使って、静的ページを立ち上げるまで。」**<br>
20 https://nikeeshipro.hatenablog.com/entry/2019/03/21/212156
21
22 ...
23 npm run deploy
24 ...
25 **でリポジトリにgh-pagesというブランチを作り、そこに静的ページがデプロイされる。がなかなか上手くいかず、いろいろな記事を見た結果、Github Actionsを使ってデプロイした方が簡単そうだった。**
26
27
28 # Github Actionsを使ったデプロイ
29 **今回はこの方法を使った。**
30
31 ### 参考記事
32 **Github ActionsでReactアプリをGithub Pagesにデプロイする**<br>
33 https://zenn.dev/acha\_n/articles/deploy-react-app-with-github-actions
34
35 ### ワークフローを作成
36 まずは Github のリポジトリを選択し、「Settings」タブ > 「Pages」 > 「Build and deployment」下の「Source」で「Github Actions」を選択します。
37
38 ![[../images/GithubActions.png]]
39
40 次にワークフローを作成します。プロジェクトのルートに「.github」というディレクトリを作成し、その中に「workflows」というディレクトリを作成します。次に「workflows」の中に .yaml ファイルを作成します。 .yaml ファイルの名前は何でも OK です。
41
42 ### deploy-gh-pages.yaml
43 ---yaml
44 name: Deploy Github Pages #workflow-name
45
46 on:
47   push:
48     branches: ['main']
49   workflow_dispatch:
50
51 permissions:
52   contents: read
53   pages: write
54   id-token: write
55
56 concurrency:
```



画像のURL



回路の紹介集

個人的な課題

- Youtubeで目的の回路を探すのは難しく、検索が不便である
- 作り方動画で、上手く作れなかった人とやり取りするが、言葉だけでは難しく労力もかかる。



検索機能の強化

- 関連するアイテム・エンティティから、検索できるようにする。
- 動画の投稿主が検索しやすいように、投稿主の名前(id)と動画のURLから検索できるようにする。
- など、専門用語を知らなくても、だいたいは見つけれられるようにしたい。
- +専門用語や回路のパターンなどで、より精度の高い検索ができるようにもしたい。

コミュニティの作成

- 適当にユーザ名とパスワードを決めてログインするだけ(もしくは匿名)で、画像や動画を添付できるメッセージ機能が欲しい
- 画像や動画を使うことで、説明も簡単で分かりやすいし、**相手の回路を確認できる**
- 作り方動画を見て、作った人が画像を上げられるようにすると面白いし、いいかも
- こういうコミュニティがあると、初心者も参加しやすいし、界限としても盛り上がる

AWSについて

メリット

- Webアプリを作るときに必要なサービス一式が揃っている。
- Serverless Frameworkを使うと、ymlファイルに必要な情報を書き込み、**コマンド一つ**でデプロイ、運用できる。(サービス一式の管理も)
- プロトタイプぐらいなら、無料枠を使える。
- 運用を考えても、従量課金制で他のサービスと比べても安い方

注意項目

- ここまで、本格的な運用をしようとする、コスト問題は避けられない
- 場合によっては、広告とかチャリティー募金、Webアプリの一部有料化などが必要
- 集団での作業など、AWSの細かい仕様やロールなど詳しく調べて管理する必要がある

でもとりあえず、作らないことには何も始まらない...

```
You, yesterday | 1 author (You) | Serverless Framework Configuration - Serverless framework configuration file
service: RedstoneCircuitDictionary

provider:
  name: aws
  region: ap-northeast-1
  stage: ${opt:stage, 'dev'}
  environment:
    DYNAMODB_TABLE: ${opt:stage, 'dev'}-redstoneCircuitDictionary-words
    REGION: ${self:provider.region}
    COGNITO_POOL_ID: !Ref CognitoUserPool
  ecr:
    images:
      redstone-circuit-dictionary-https:
        path: .
        file: docs/Dockerfile.https
      redstone-circuit-dictionary-cognito:
        path: .
        file: docs/Dockerfile.cognito
  iamRoleStatements:
    - Effect: Allow
      Action:
        - dynamodb:Query
        - dynamodb:Scan
        - dynamodb:GetItem
        - dynamodb:PutItem
        - dynamodb:UpdateItem
        - dynamodb:DeleteItem
      Resource:
        - "arn:aws:dynamodb:${opt:region, self:provider.region}:*:table/${self:provider.region}/${opt:stage}/${self:service}"
        - "arn:aws:dynamodb:${opt:region, self:provider.region}:*:table/${self:provider.region}/${opt:stage}/${self:service}"
    - Effect: Allow
      Action:
        - cognito-idp:AdminConfirmSignUp
      Resource: "arn:aws:cognito-idp:${self:provider.region}:${self:aws:accountId}"
    - Effect: Allow
      Action:
        - logs:CreateLogGroup
        - logs:CreateLogStream
        - logs:PutLogEvents
      Resource: "*"

functions:
  httpHandler:
    image:
      name: redstone-circuit-dictionary-https
    events:
      - http:
```

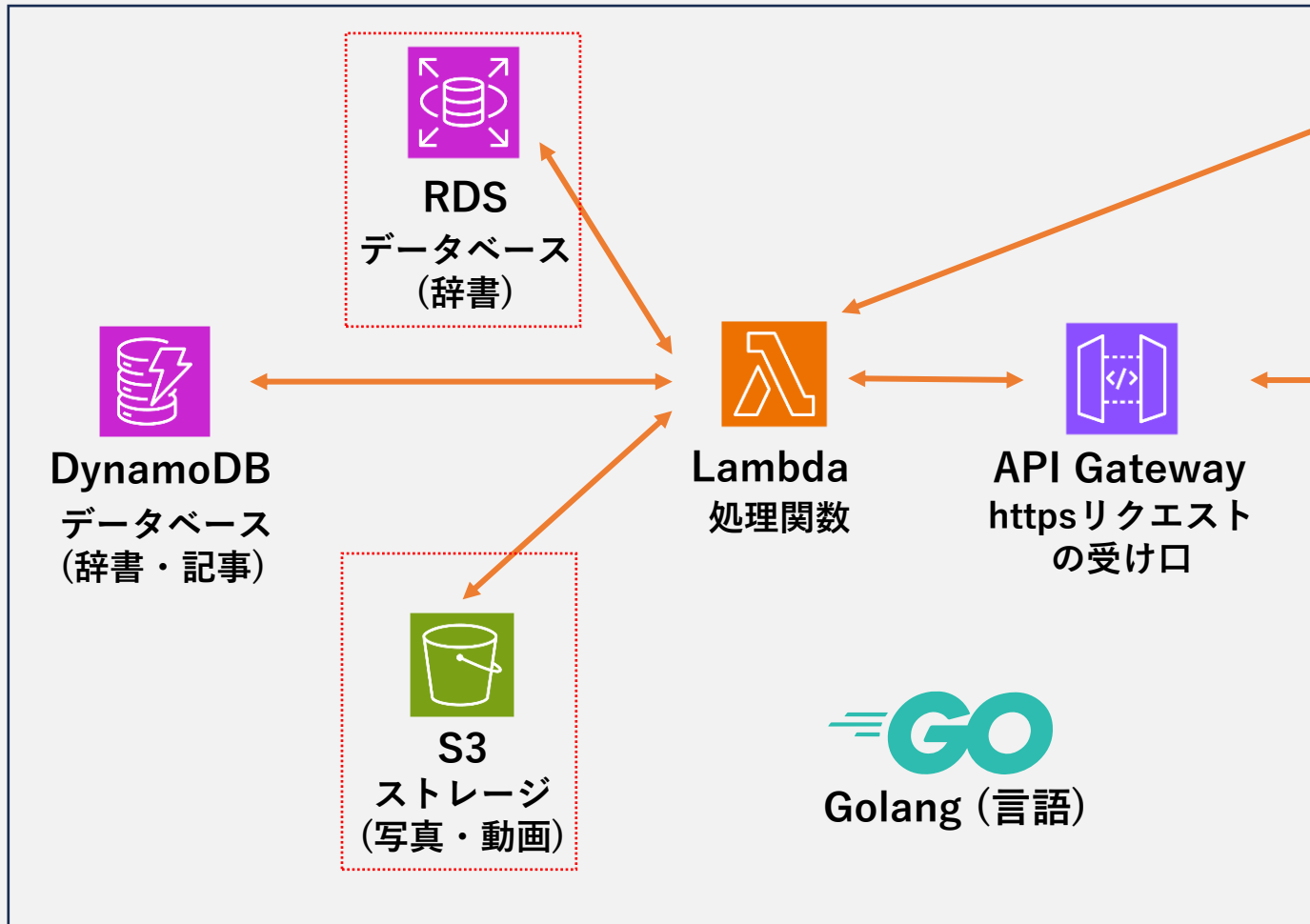
serverless.yml

sls deploy

デプロイコマンド

プロジェクト構成例


バックエンド



フロントエンド


Cognito
ユーザ管理


Amplify
ホスティング


React(言語)

未実装