

サーバーレスなRSSリーダーをつくる

#python

#AWS

#Lambda

#Docker

#Github Actions

#VSCode Remote Container

@mini_koharu

もくじ

- やりたいこと
- やってみる
 - 開発環境
 - RSS取得
 - デプロイ
 - Lambdaをつくる
 - RSS取得先URLを設定する
 - 定期実行

やりたいこと1/3

rss

 mini-rss-reader-webhook ボット 2021/03/05

サイトやアプリへの動画埋め込みを簡単に！デバイスを選ばないプレイヤー：THEOplayer

<https://dev.classmethod.jp/news/theoplayer-webinar/>

2021-03-05 14:37:18+09:00

クラスメソッド発「やってみた」系技術メディア | DevelopersIO

サイトやアプリへの動画埋め込みを簡単に！デバイスを選ばないプレイヤー：THEOplayer | DevelopersIO

THEOplayerはユーザーの使用するデバイスやOS、アプリケーション環境を問わず、動画再生体験を提供できるサービスです。Webサイトやモバイルアプリへの動画埋め込みを可能とするだけでなく、ストリーミング配信やDRM（デジタル著作権管理）にも対応しており、企業独自のデジタルコンテンツ配信サービスを構築することができます。

CLASSMETHOD EVENT



サイトやアプリへの
動画埋め込みを簡単に！
デバイスを選ばないプレイヤー

 THEOplayer

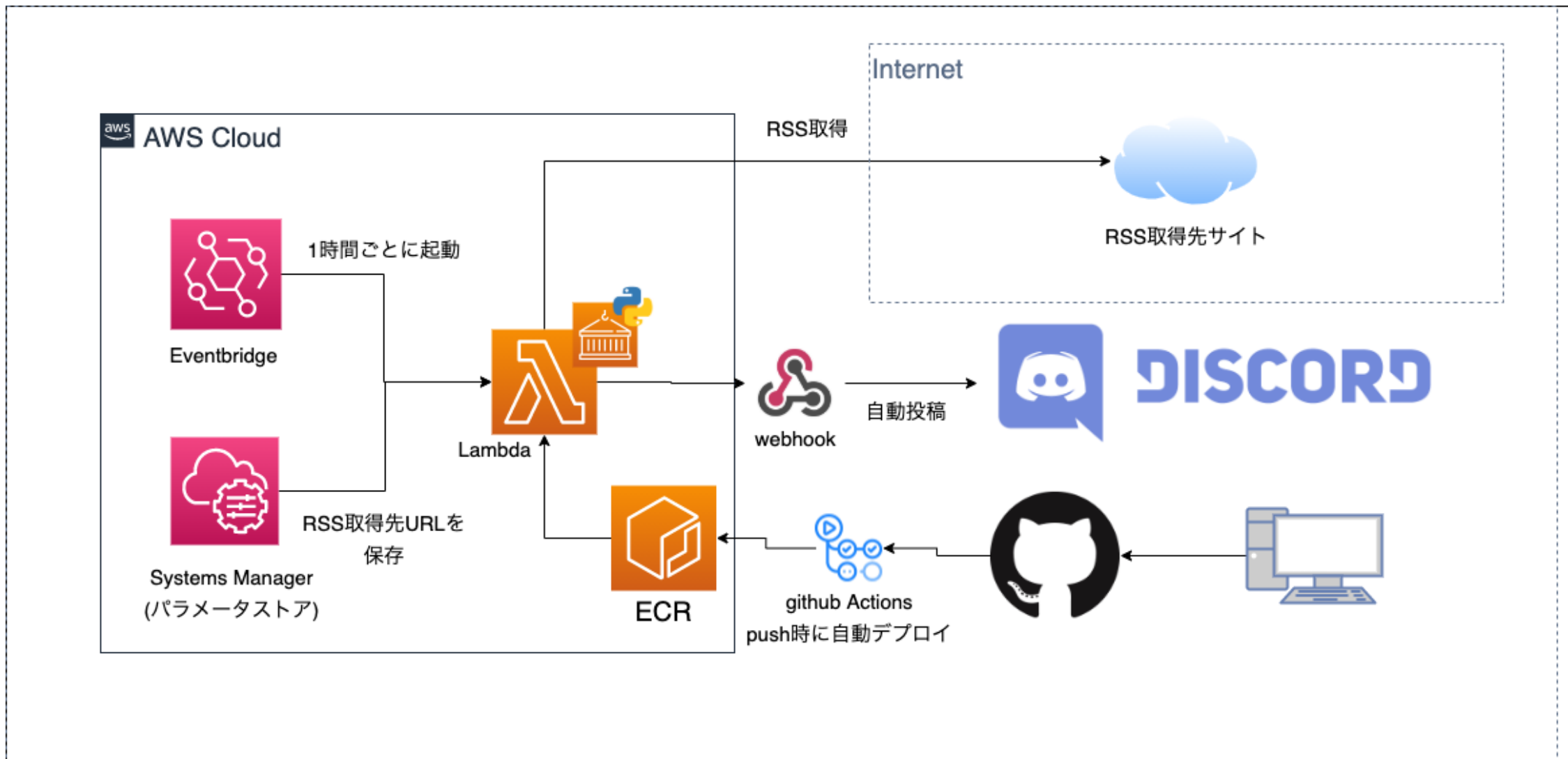
ウェビナー

やりたいこと2/3

+新しそうなこと全部やる+

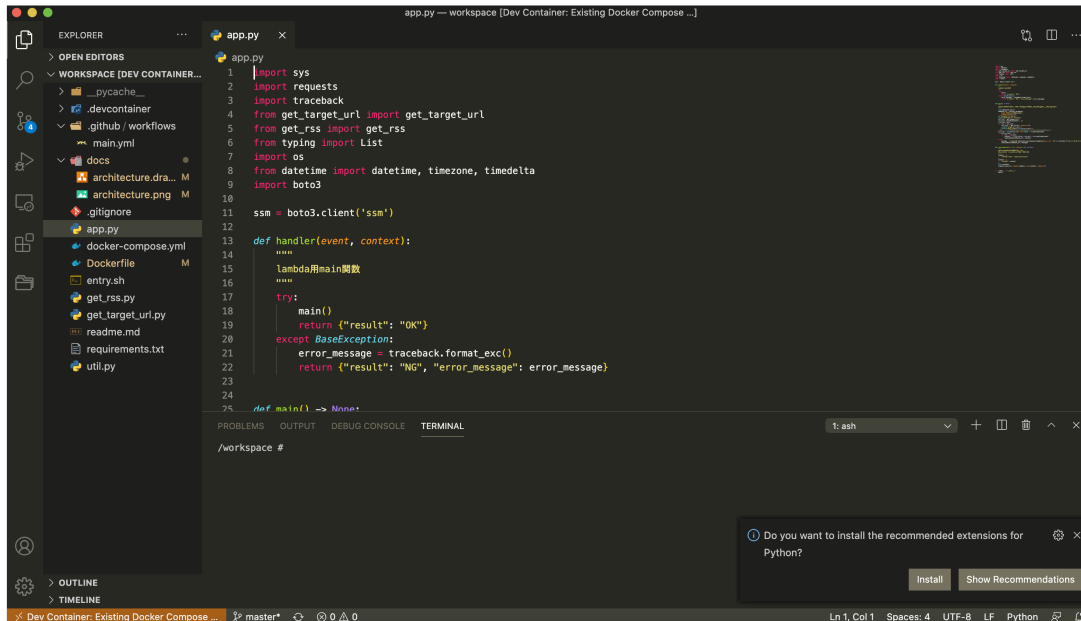
- Lambdaをコンテナで動かす
- Lambdaを定期実行する(Eventbridge)
- コンテナで開発する(VSCode Remote Container)
- CIの実装(Github Actions+ECR)
- クロール先URLをハードコーディングしない
(Systems Managerパラメータストア)

やりたいこと3/3(構成図)



やってみる:開発環境

- VSCode Remote Containerを使う
 - コンテナの中で開発ができる
 - ローカルにはdocker(-compose)とVSCodeだけ入っていればよい
 - Lambdaで動かすコンテナでそのまま開発できる
 - 実行環境と開発環境が完全に一致



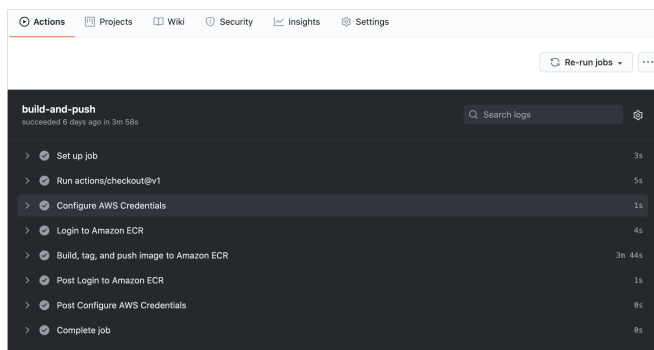
やってみる:RSS取得

- RSSをとってくる
 - pythonのfeedparserというライブラリが使える

```
def get_rss(url: str) -> List[RssContent]:  
    feed = feedparser.parse(url)  
    rss_list: List[RssContent] = []  
    for entry in feed.entries:  
        if not entry.get("link"):  
            continue  
        rss_content = RssContent(  
            title=entry.title,  
            url=entry.link  
        )  
        rss_list.append(rss_content)  
    return rss_list
```

やってみる:デプロイ

- GithubへのpushをトリガーにGithub Actionsが走り、自動でコンテナをビルド→ECRへアップロード
 - 参考:[GitHub ActionでDockerコンテナをビルドしてAmazon ECRに保存する](#)
- Github Actions #とは
 - Github上でCI/CD操作を実行できる機能(設定はyaml)
 - ほぼ無料
 - [AWS公式のActionsが公開されている](#)



やってみる:Lambdaをつくる

- コンテナ利用の場合もGUIからつくれる
- ソースコードやzipをアップロードする代わりにECRのURIを指定する

☰

関数の作成 Info

以下のいずれかのオプションを選択して、関数を作成します。

一から作成 ☐

シンプルな Hello World の例で開始します。

設計図の使用 ☐

一般的ユースケース用のサンプルコードと設定プリセットから Lambda アプリケーションを構築します。

コンテナイメージ ☒

関数にデプロイするコンテナイメージを選択します。

Serverless Application Repository の参照 ☐

AWS Serverless Application Repository からサンプル Lambda アプリケーションをデプロイします。

基本的な情報

関数名
関数の目的を名前として入力します。

半角英数字、ハイフン、アンダースコアのみを使用でき、スペースは使用できません。

コンテナイメージ URI Info
関数に使用するコンテナイメージの場所。

有効な Amazon ECR イメージ URI が必要です。

イメージを参照

①

やってみる:RSS取得先URLを設定する

- Systems Managerパラメータストアを利用する
 - デプロイし直さなくてもRSS取得先URLをGUIから追加/削除できる

名前

🔍 RSSURLList

説明 — *Optional*

利用枠

パラメータストアは標準パラメータおよび詳細パラメータを提供します。

☒ 標準

パラメータの上限は 10,000 です。パラメータ値サイズの上限値は 4 KB です。パラメータポリシーは使用できません。追加料金はありませ

☐ 詳細

10,000 以上のパラメータを作成できます。パラメータ値サイズの上限値は 8 KB です。パラメータポリシーを使用するように設定できます。詳細パラメータごとに請求が発生します。

タイプ

String

データ型

text ▼

値

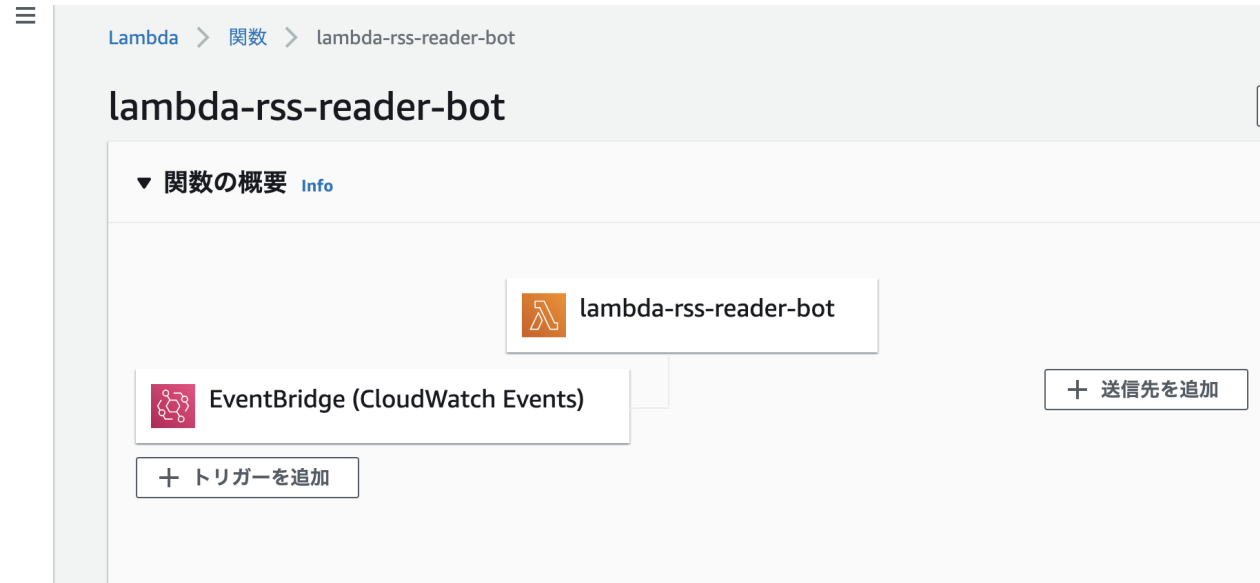
https://dev.classmethod.jp/feed/
https://zenn.dev/topics/aws/feed

最大長は 4096 文字です。

- パラメータ内の改行は保存される
- パラメータの取得はboto3(python用aws接続ライブラリ)から

やってみる:定期実行

- EventbridgeをLambdaのトリガーとして利用する



- 周期はrate(1 hour)
 - 開始タイミングは指定できないらしい

できた

rss

 mini-rss-reader-webhook ボット 2021/03/05

サイトやアプリへの動画埋め込みを簡単に！デバイスを選ばないプレイヤー：THEOplayer

<https://dev.classmethod.jp/news/theoplayer-webinar/>

2021-03-05 14:37:18+09:00

クラスメソッド発「やってみた」系技術メディア | DevelopersIO

サイトやアプリへの動画埋め込みを簡単に！デバイスを選ばないプレイヤー：THEOplayer | DevelopersIO

THEOplayerはユーザーの使用するデバイスやOS、アプリケーション環境を問わず、動画再生体験を提供できるサービスです。Webサイトやモバイルアプリへの動画埋め込みを可能とするだけでなく、ストリーミング配信やDRM（デジタル著作権管理）にも対応しており、企業独自のデジタルコンテンツ配信サービスを構築することができます。

CLASSMETHOD EVENT



サイトやアプリへの
動画埋め込みを簡単に！
デバイスを選ばないプレイヤー

 THEOplayer

ウェビナー

おわり

- おまけ
 - このスライドはMarpによる
 - markdownからスライドが生成できる
 - 画像はhtmlタグで埋め込むのがよい