

Flutter開発とGitHub Actions

YORIFUJI MITSUNORI

ORION SALON #31

自己紹介

- YORIFUJI MITSUNORI
 - @yorifuji on X/GitHub/Zenn
- 金融系スタートアップのiOSエンジニア





CIサービス戦国時代？



アイデアからリリースまでのワークフローを自動化

GitHub Actionsを使用すると、ワールドクラスのCI/CDですべてのソフトウェアワークフローを簡単に自動化できます。GitHubから直接コードをビルド、テスト、デプロイでき、コードレビュー、ブランチ管理、問題のトリアージを希望どおりに機能させます。

GitHub Actionsを使ってみる >



GitHub Actions

<https://github.co.jp/features/actions>

GitHub Actionsで自動化している作業

- Check(formatter/lint)
 - flutter format
 - flutter analyze
- Test
 - Unit/Widget/Integration Test
 - VRT(Visual Regression Test)
- Build & Deploy
 - Staging(Firebase App Distribution)
 - Production(Google Play, App Store)
- Bump up
 - pubspec.yml の version更新&PR作成
- Nightly build

**FlutterのCI環境として
GitHub Actionsを
採用する理由を考えてみた**

推し要素をリストアップ！

GitHub Actionsのおすすめポイント

- GitHubに付属しているのですぐに使い始められる
- GitHubで発生するイベントが利用できる
- パブリックリポジトリは無料、毎月一定の無料枠あり
- サードパーティーのアクションが豊富
- カスタマイズ性が高い
- 無料プランで最大20並列までジョブが実行できる
- Self-hosted runnerが使える
- VSCode Extensionが便利
- ドキュメントが豊富、日本語翻訳あり

Flutter開発でのおすすめポイント

- FlutterSDKのインストールが簡単
- ワークフローやターゲットに合わせてRunnerの使い分けができる
- 開発環境（Xcodeなど）は複数バージョンがインストール済み
- Windows, Linux, macOSを使ってビルドできる
- 複数のJobを並列に同時実行できる
 - CheckやAndroidとiOSの build & deploy

一つだけ挙げるとしたら

パブリックリポジトリ

無料



GitHubはオープンソースを支援します。

プライベートリポジトリ

含まれる利用時間

Free

2,000分
1か月あたり

Pro

3,000分
1か月あたり

Team

1か月あたり

Enterprise

50,000分
1か月あたり

追加購入

Linux

2 cores, 7GB

\$0.008

1分あたり

Windows

2 cores, 7GB

\$0.016

1分あたり

macOS

3 cores, 7GB

\$0.08

1分あたり

セルフホスト

Free

GitHub Actionsはレガシープランユーザーではご利用できません。

パブリックリポジトリは無料、毎月一定の無料枠あり

GitHubで発生するイベントとの連携

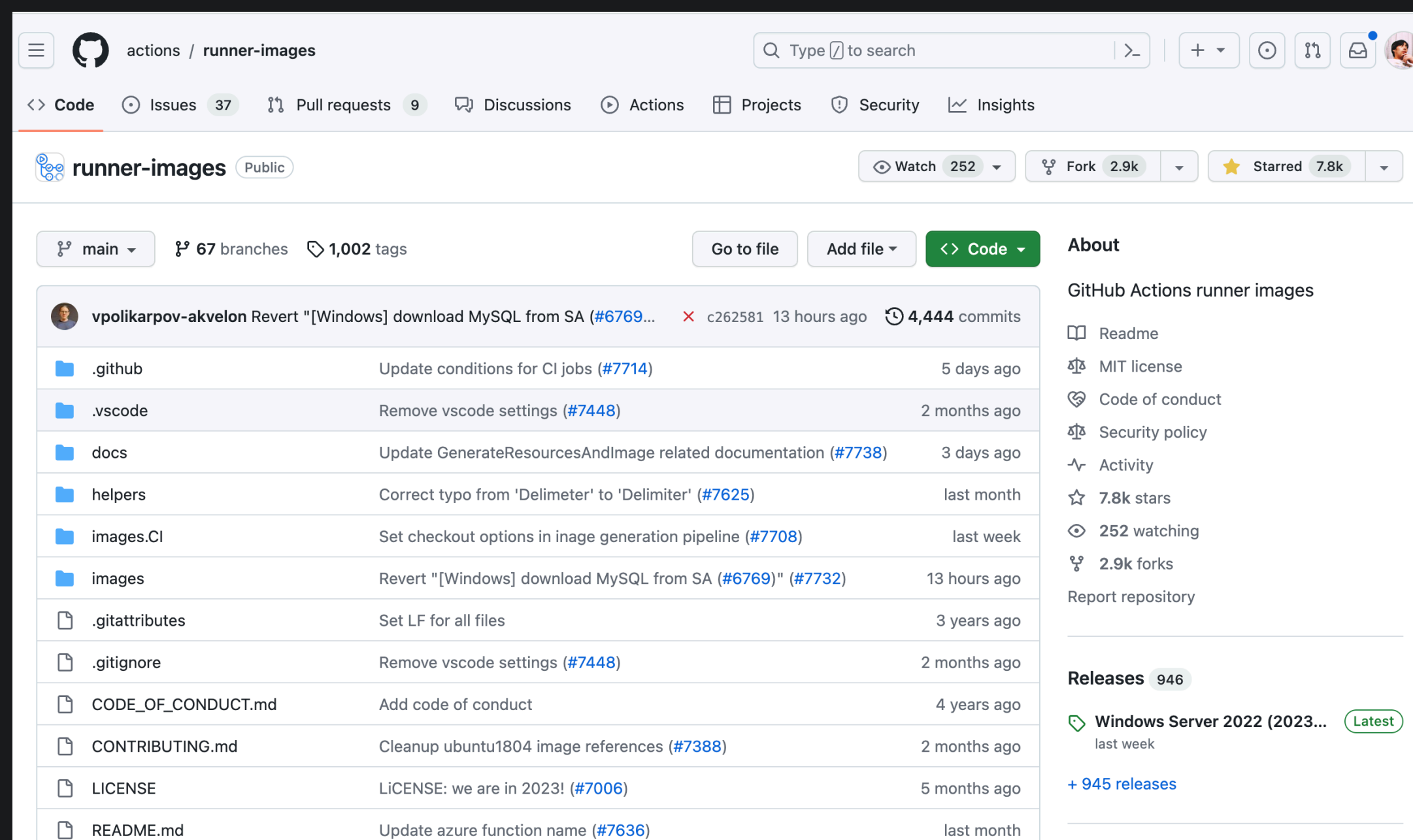
- 他のCIサービスではPUSH・PR・Tagなどの**Git**に関するトリガーをサポート
- GitHub Actionsでは**GitHub**で発生するイベントをワークフロー（ジョブ）実行のトリガーにすることができる
 - 例：IssueやPRに内容に合わせてLabelをつける
- GitHub Actionsはテストやビルドだけでなく開発プロセス全体で利用できる

GitHub-hosted runner

- GitHub-hosted runnerはGitHubが標準で提供しているCIの実行環境（VM）
 - Windows, Linux(ubuntu), macOSから選択可能
 - macOSはMonterey, Venturaが提供されている
- GitHubが運用管理しているため自分でメンテナンスは不要
- ジョブの実行単位でまっさらなインスタンスが割り当てられる
- 分単位で課金
 - macOSは最も安いLinuxの10倍の値段がするので適材適所で使い分けする

参考：GitHub-hosted runnerのソフトウェアの構成

- CI環境にインストールされているソフトの一覧などはGitHub上で公開されている
- <https://github.com/actions/runner-images>



macOS 13

- OS Version: macOS 13.6 (22G120)
- Kernel Version: Darwin 22.6.0
- Image Version: 20231002.1

Xcode

Version	Build	Path
15.0	15A240d	/Applications/Xcode_15.0.app
14.3.1 (default)	14E300c	/Applications/Xcode_14.3.1.app
14.2	14C18	/Applications/Xcode_14.2.app
14.1	14B47b	/Applications/Xcode_14.1.app

(2023.10.8時点)

Self-hosted runner

- プライベートリポジトリでGitHub-hosted runnerでmacOSを使うと利用時間が x10 換算のため無料枠の消費が早い
- **Self-hosted runner**は所有しているマシンでGitHub Actionsのワークフロー実行するシステム
- メリット
 - プライベートリポジトリでRunnerの利用時間に対する料金が発生しない
- デメリット
 - マシンの管理・運用コストは全て自腹（OSやツールのアップデート、通信費用など）
 - GitHub-hostedとは異なり標準ではVMの機能が提供されていないのでゴミが残ったりする

Flutterのビルド環境のセットアップ

- GitHubが提供するRunnerにはFlutter SDKがインストールされていない
- テストやビルドを行うためにSDKのインストールが必要
 - third-partyのアクションがある
 - <https://github.com/marketplace/actions/flutter-action>
 - Gitから直接clone
- XcodeやAndroid SDKなどは複数バージョンがインストール済みで切り替えも簡単
 - Xcodeなら環境変数 DEVELOPER_DIR で切り替え可能

ワークフローの目的でRunnerを切り替える

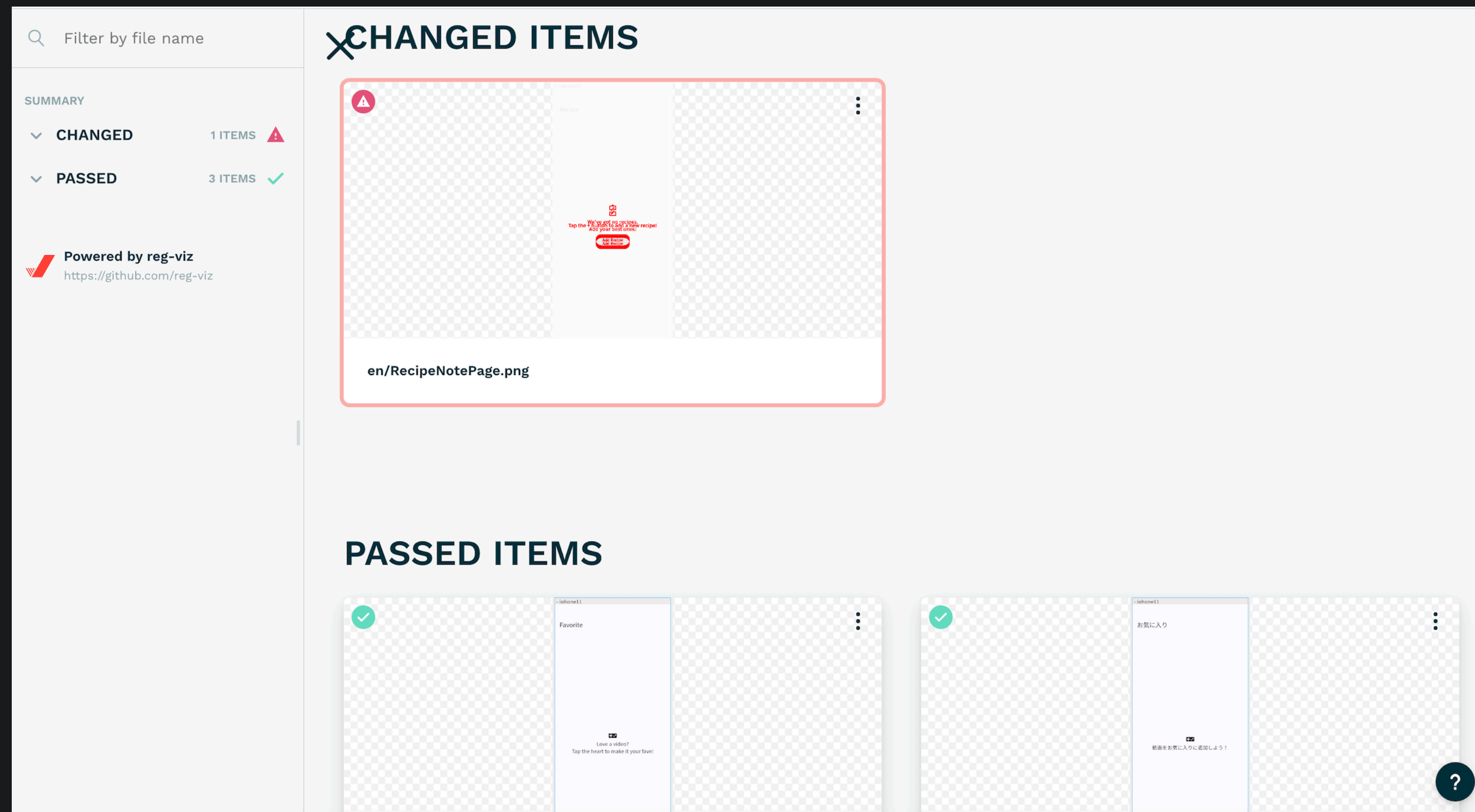
- 静的解析（flutter analyze）やテスト、ビルド&デプロイで利用するケース
 - 静的解析やテストは最も安いLinuxで実行
 - ビルドはAndroidやWebならLinux、iOSはmacOS
- ワークフローやビルドターゲットに合わせてRunnerを使い分けることで費用をコントロールできる

ワークフローの再利用性が高い

- GitHub Actionsのワークフローの定義はyamlファイルなので再利用性が高い
- 新しいプロジェクトを開始したら既存のファイルをコピーするだけCI/CDのセットアップが完了する
- GUIのワークフローエディタをポチポチしなくて良い（一長一短）

ワークフローのカスタマイズ性が高い

- CIジョブの中でJavaScriptを使って処理を記述できる
 - <https://github.com/actions/github-script>
- Job Summaries
 - ワークフローの実行結果の画面にMarkdown形式などで出力できる



reg-suitを使ったVRT（Visual Regression Test）

参考：reg-suitのサマリーをJob Summariesに出力

```
- name: Generate Comment
  uses: actions/github-script@v6
  with:
    script: |
      const fs = require('fs');
```



(途中省略)

```
await core.summary
  .addHeading(`${icon} reg-suit has checked for visual changes`, 3)
  .addTable([
    ["🔴 Changed", "🟡 New", "⚫ Deleted", "🟢 Passing"],
    [stats.changed, stats.newItems, stats.deleted, stats.passing]
  ])
  .addHeading("📝 Log", 3)
  .addCodeBlock(log, "log")
  .write()
```

compare summary

...

! reg-suit has checked for visual changes

 Changed	 New	 Deleted	 Passing
1	0	0	3

Log

```
[reg-suit] info version: 0.12.2
[reg-suit] info Skipped to detect the previous snapshot key because key generator plugin is not set up.
[reg-suit] info Comparison Complete
[reg-suit] info   Changed items: 1
[reg-suit] info   New items: 0
[reg-suit] info   Deleted items: 0
[reg-suit] info   Passed items: 3
```



Job summary generated at run-time

GitHub Actionsのruns（実行ログ）のページに表示された内容

今後の改善を期待するポイント

- ファイルアップローダー機能
 - iOSアプリのビルドでは各種証明書やProvisioning Profileが必要
 - GitHub Actionsはファイルアップローダーが提供されていない
 - ファイルをbase64に変換してsecretに登録 -> CIジョブの中でファイルに書き出す作業が必要
 - Bitriseなどはファイルアップロード機能からアップロードができる
- デバッグしやすくしてほしい

GitHub Actions で Automatically
manage signing を使って Flutte...



yorifuji

// Zenn

Flutter開発で使っているGitHub Actions
のワークフロー



yorifuji

// Zenn

具体的なワークフローの例はこちらの記事を参考に🙏

<https://zenn.dev/yorifuji>

ありがとうございました