

GitHub Actions の痒いところを埋める サードパーティーランナー

2024/5/16 GitHub Actions Meetup Tokyo #3

どら (@dora1998)



自己紹介



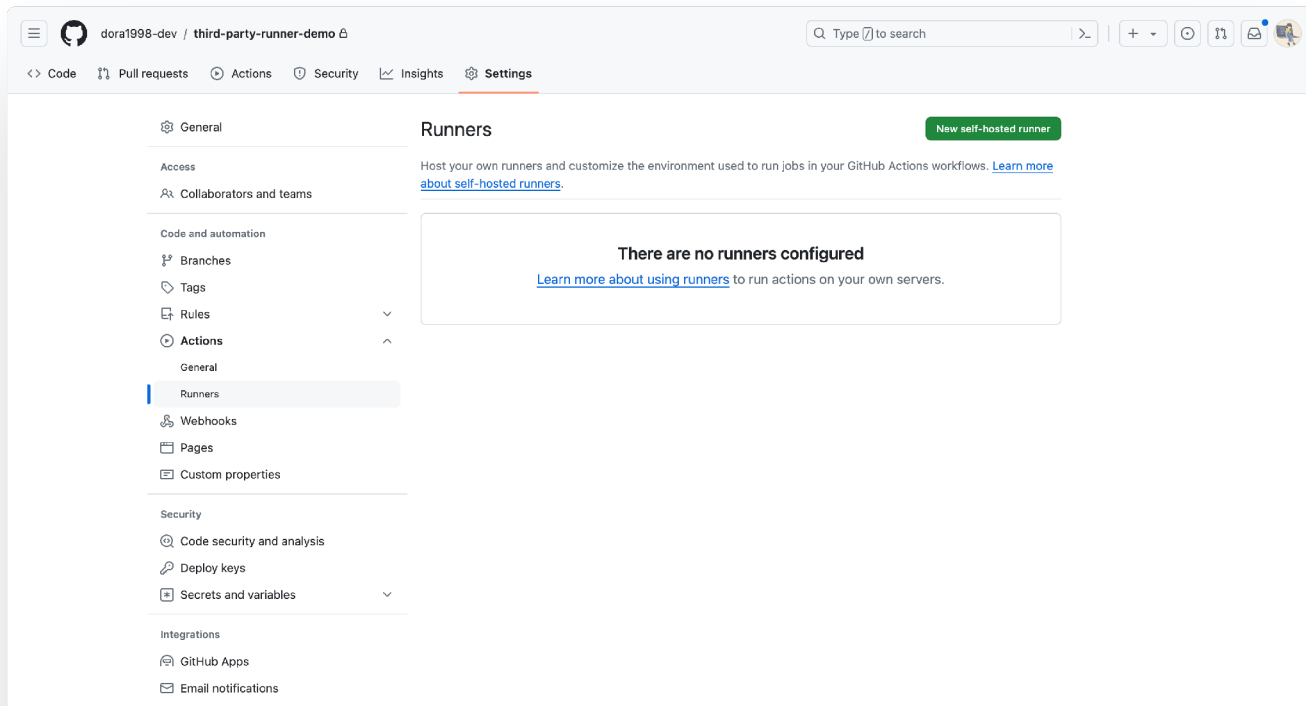
どら

🐼 @dora1998 ✂ @d0ra1998


- 株式会社サイバーエージェント
 - WINTICKET Web テックリードマネージャー
- CIの高速化が好き

セルフホステッド ランナーとは

- ユーザーが用意したマシンで GitHub Action の Job を実行することができる
- つまり、GitHub が提供していない環境も使える



サードパーティーランナーとは

- セルフホステッド ランナーの仕組みを使ってランナーを提供するプロバイダーが存在する
 - まとめているリポジトリもあります
 -  [neysonfu/awesome-github-actions-runners](https://github.com/neysonfu/awesome-github-actions-runners)
 - 筆者はこのブログ記事で知りました
 - [GitHub Actionsのサードパーティーマネージドランナーの紹介 - いけだや技術ノート](#)

サードパーティーランナーの例

- Namespace
- BuildJet
- WarpBuild
- AWS CodeBuild
- Cirun

どういう時に使うといいか

- 料金を抑えたい
- ARM CPU や GPU を使いたい
- 実行時間を短くしたい
- クラウドストレージへアップロード/ダウンロードしたい
- より詳細なジョブの分析を行いたい

料金を抑えたい

プライベートリポジトリで使う場合、GitHub-hosted runner の半額程度になる

サービス	2 vCPUで動かした場合
GitHub (Private) ^[1]	\$0.008 (2 vCPU, 7GB, 14GB)
Namespace ^[2]	\$0.003 (2 vCPU, 4GB, 48GB ^[3])
BuildJet ^[4]	\$0.004 (2 vCPU, 8GB)
WarpBuild ^[5]	\$0.004 (2 vCPU, 7GB, 150GB)
AWS CodeBuild ^[6]	\$0.0024 (lambda.x86-64.2GB)

1. <https://docs.github.com/ja/billing/managing-billing-for-github-actions/about-billing-for-github-actions#per-minute-rates>

2. <https://namespace.so/pricing> (※月額に含まれるクレジット分は2/3)

3. <https://namespace.so/docs/architecture/containers#machine-resource-shapes>

4. <https://buildjet.com/for-github-actions/docs/about/pricing>

5. <https://docs.warpbuild.com/runners> (※スポットインスタンスを使うとさらに25%安い)


6. <https://aws.amazon.com/jp/codebuild/pricing/>


ARM CPU や GPU を使いたい


- ARM64 や GPU を搭載したランナーを使うことができる
- ちなみに、これらは GitHub-hosted runner でもベータで提供されている
 - <https://github.com/github/roadmap/issues/961>
 - <https://github.com/github/roadmap/issues/836>

実行時間を短くしたい

- BuildJet や WarpBuild はシングルコア性能が高いことを強みとしている^{[1][2][3]}
- CPUヘビーなジョブであれば、同じコア数で実行時間が短くなるかも

1. <https://buildjet.com/for-github-actions/blog/a-performance-review-of-github-actions-the-cost-of-slow-hardware> 

2. <https://docs.warpbuild.com/#what-is-warpbuild> 

3. <https://www.warpbuild.com/blog/github-actions-speeding-up#a-note-on-github-hosted-runner-processors> 

クラウドストレージへアップロード/ダウンロードしたい

- 例えば、S3 を使う場合は AWS 上のランナーを使えば高速にアクセスできる
- 実際のベンチマークでもダウンロードが10倍ほど速かった（1 MB, 1000個, 16スレッド）
 - GitHub-hosted runner: 31.4 MB/s
 - AWS CodeBuild: 293.0 MB/s

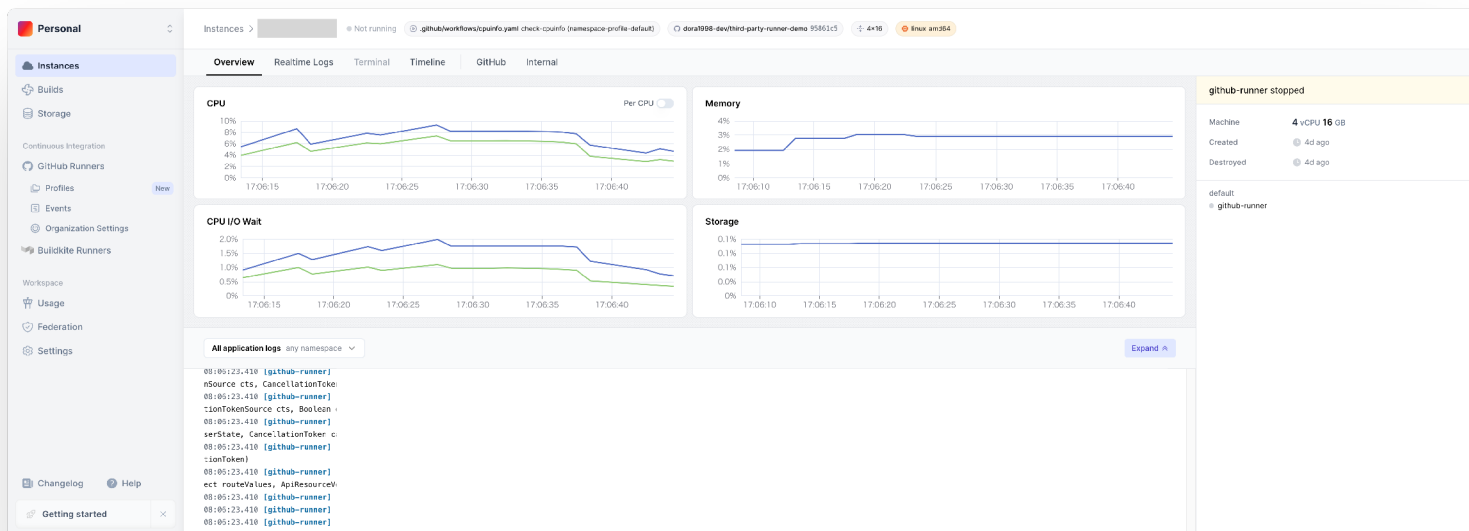
測定内容

 [dvassallo/s3-benchmark](https://github.com/dvassallo/s3-benchmark) をフォークして使用

- GitHub-hosted runner: プライベート リポジトリの標準 GitHub-hosted runner (2コア, 7GB RAM)
- AWS CodeBuild: lambda.x86-64.2GB

より詳細なジョブの分析を行いたい

- Namespace は特に管理画面が充実している
- GitHub 上では確認できない以下のメトリクスが確認できる
 - 実行中の CPU /メモリ使用率の推移
 - 日ごと・ワークフローごとの使用量



気をつけたいこと

- 起動時間がかかる
- Cache, Artifact が遅い
- ランナーの差異
- セキュリティ

起動時間がかかる

- GH-hosted runner よりもジョブの実行開始までラグがある
- 体感、速い順で以下のような印象
 - ↑ ほぼゼロ
 - GH-hosted runner
 - Namespace, BuildJet
 - AWS CodeBuild
 - Cirun (GCP)
 - ↓ 1分程度

Cache, Artifactが遅い


- Cache や Artifact の Action は GitHub のストレージとやり取りする
 - `actions/cache`
 - `actions/upload-artifact`
 - `actions/download-artifact`
- GH-hosted runner だと(おそらく内部で繋がってるので)速いが、セルフホステッドランナーだと遅くなる
- 各社置き換えを提供しているので、その使い勝手を見た方が良い


ランナーの差異

- そのまま `runs-on` だけ書き換えると微妙に動かないことがある
 - パッケージが足りない
 - Namespaceのランナーではlessが標準で存在しない
 - CodeBuild では AWS 操作時に Lambda に設定されたサービスロールで認証されている
- (GH-hosted runner の障害時などに)サクッと載せ替えるのは結構難しい

セキュリティ

- リポジトリの Administration 権限を渡す必要がある
 - これはランナーを登録するために必要^[1]
 - 恐らく `/repos/{owner}/{repo}/actions/runners` 以下のAPIを操作するため
 - 一方で、リポジトリの削除など重大な操作もできてしまう^[2]ので、万が一管理体制に不備があると重大インシデントにつながりうる

1. <https://discord.com/channels/975088590705012777/975088591183171597/1168482784696873010> 

2. <https://docs.github.com/en/rest/authentication/permissions-required-for-github-apps?apiVersion=2022-11-28#repository-permissions-for-administration> 

まとめ

- サードパーティランナーにはGitHub-hosted runnerにはない特徴がある
- 使う上では…
 - 起動まで若干ラグがある
 - Cache, Artifact をそのまま使用すると遅いので対策が必要
 - ランナーの差異で調整が必要かも
 - GitHub App が要求する権限が広いので注意