tfnotify

Show Terraform execution plan beautifully on GitHub

@b4b4r07 (Sep 11, 2018) / HashiCorp Meetup #3

DevOps を支える今話題の HashiCorp ツール群について

BABAROT / @b4b4r07

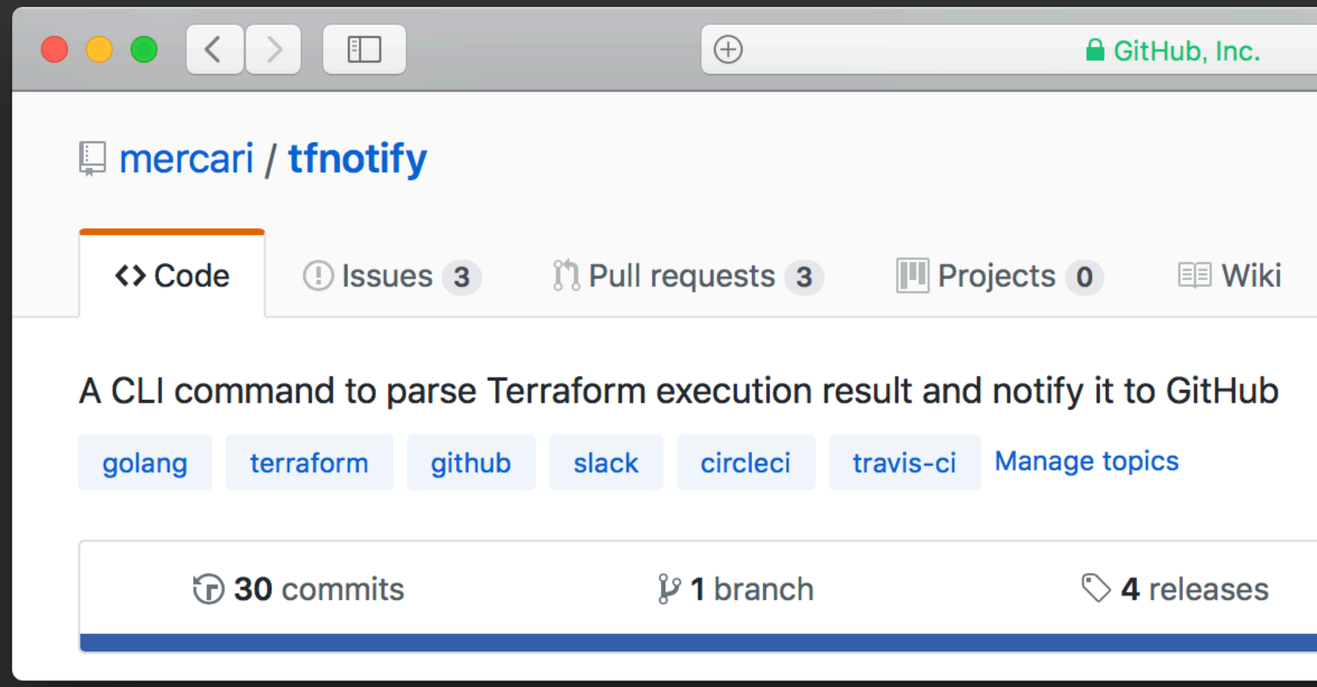
Mercari, Inc.
SRE, Microservices Platform

Blog / **tellme.tokyo**

Agenda

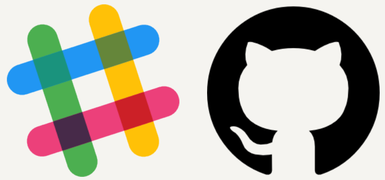
1. *mercari/tfnotify*
2. *Why tfnotify?*
3. *Implementations*

mercari/tfnotify



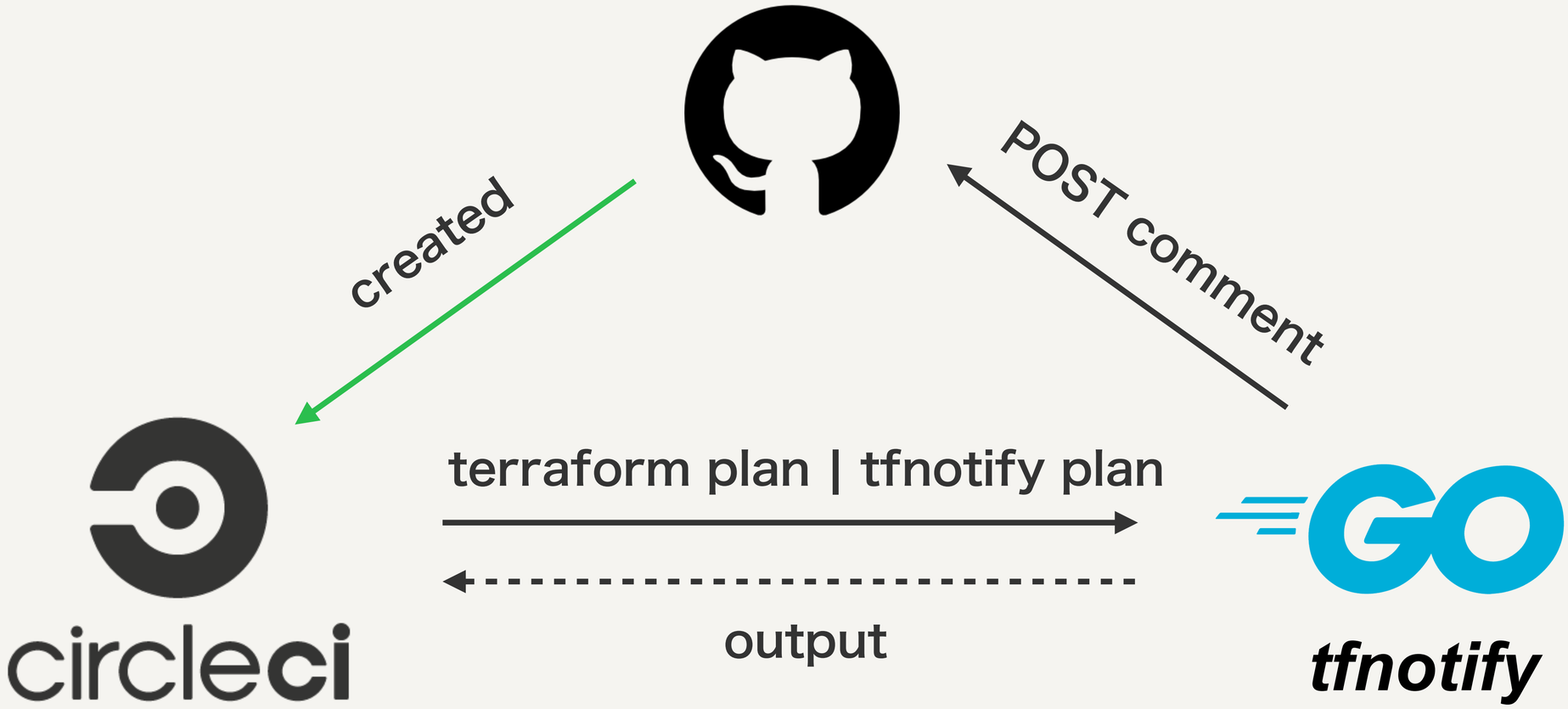
tfnotify とは

- Go 製 CLI ツール
- Terraform の実行結果をパースし、適切なフォーマットにあてはめて、任意の通知先 (GitHub のコメント等) へ通知する

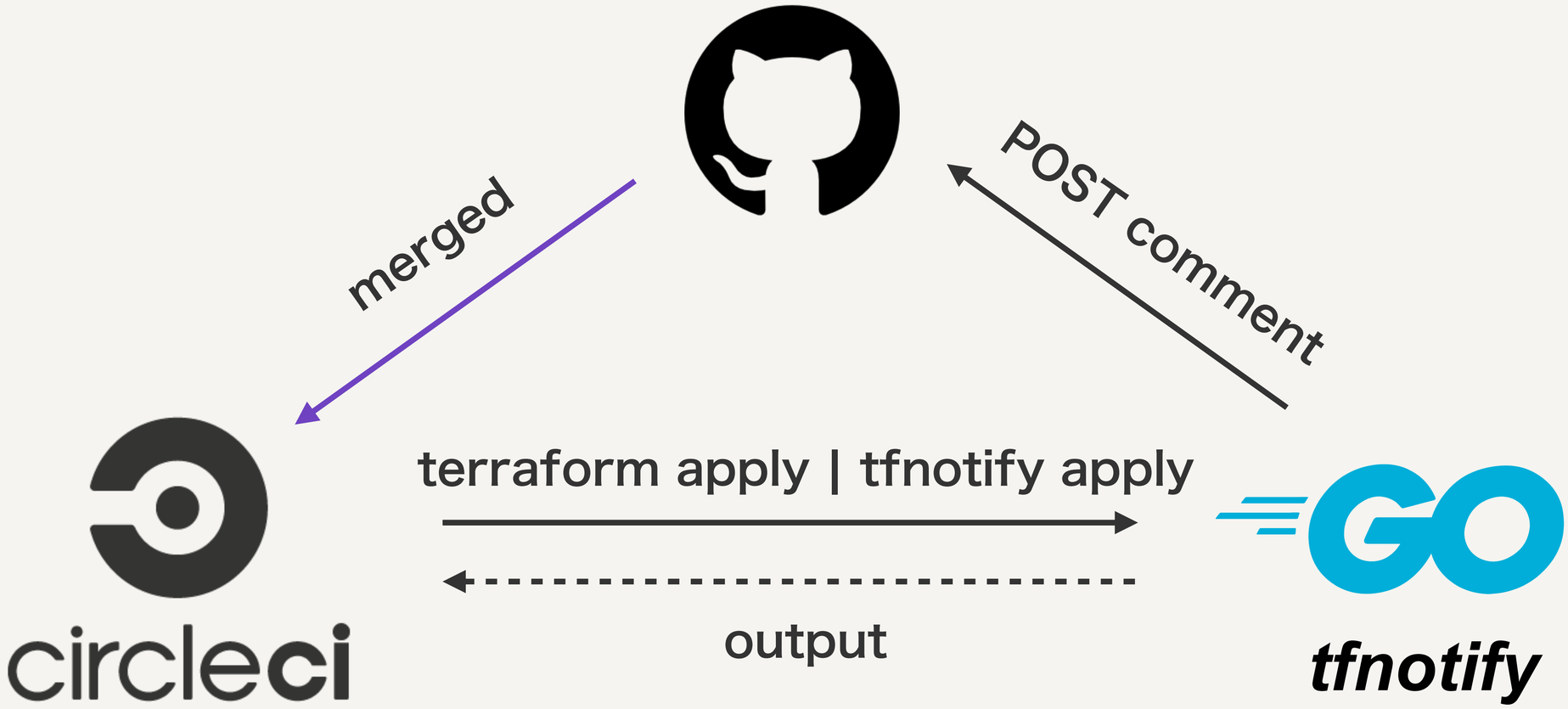


```
NAME:
   tfnotify - Notify the execution result of terraform com

USAGE:
   tfnotify [global options] command [command options] [ar

VERSION:
   0.1.0

COMMANDS:
     fmt      Parse stdin as a fmt result
     plan     Parse stdin as a plan result
     apply    Parse stdin as a apply result
     help, h  Shows a list of commands or help for one com

GLOBAL OPTIONS:
   --ci value         name of CI to run tfnotify
   --config value     config path
   --notifier value   notification destination
   --help, -h         show help
   --version, -v      print the version
```

Workflow

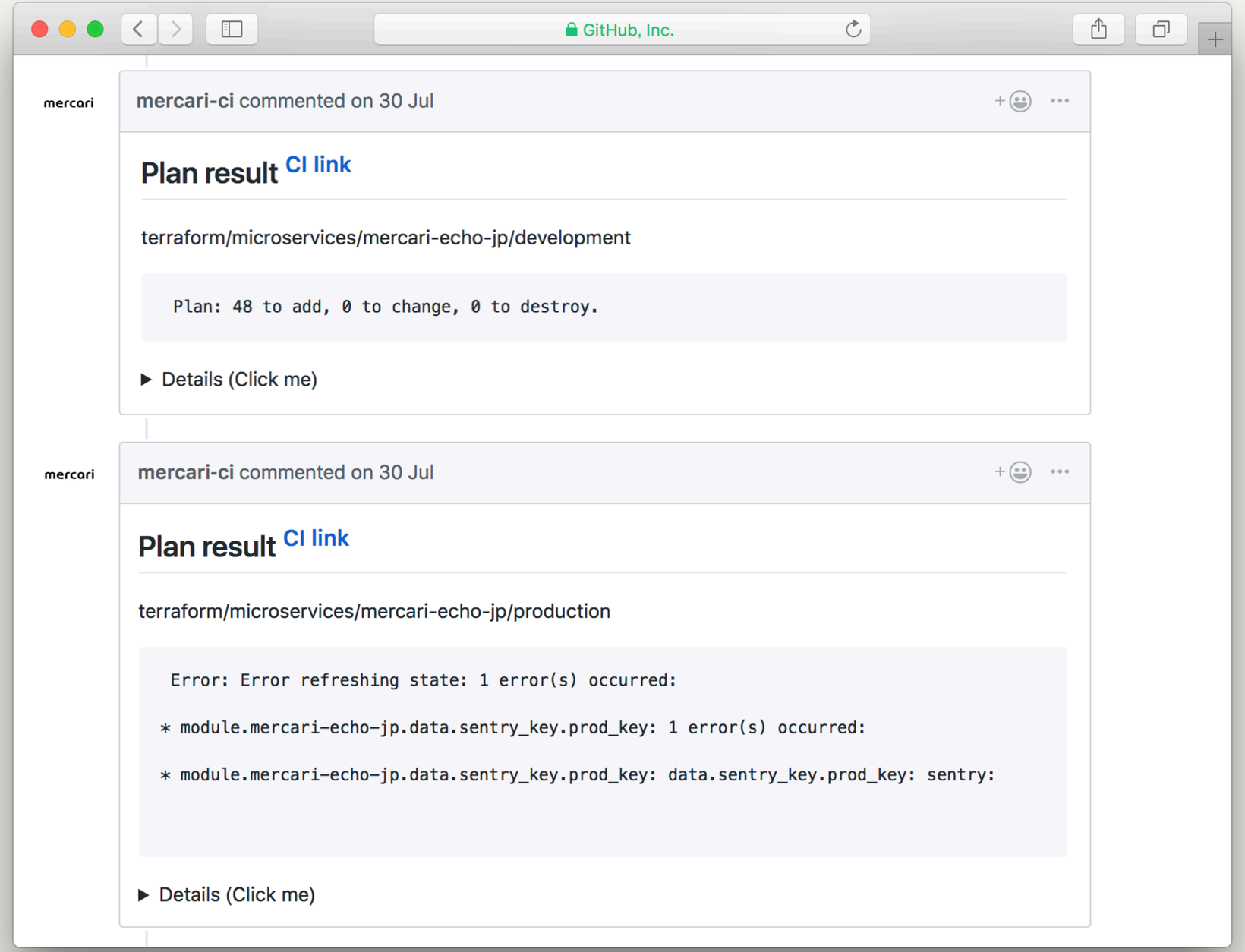

created
POST comment
terraform plan | tfnotify plan
output
circleci
tfnotify

Workflow

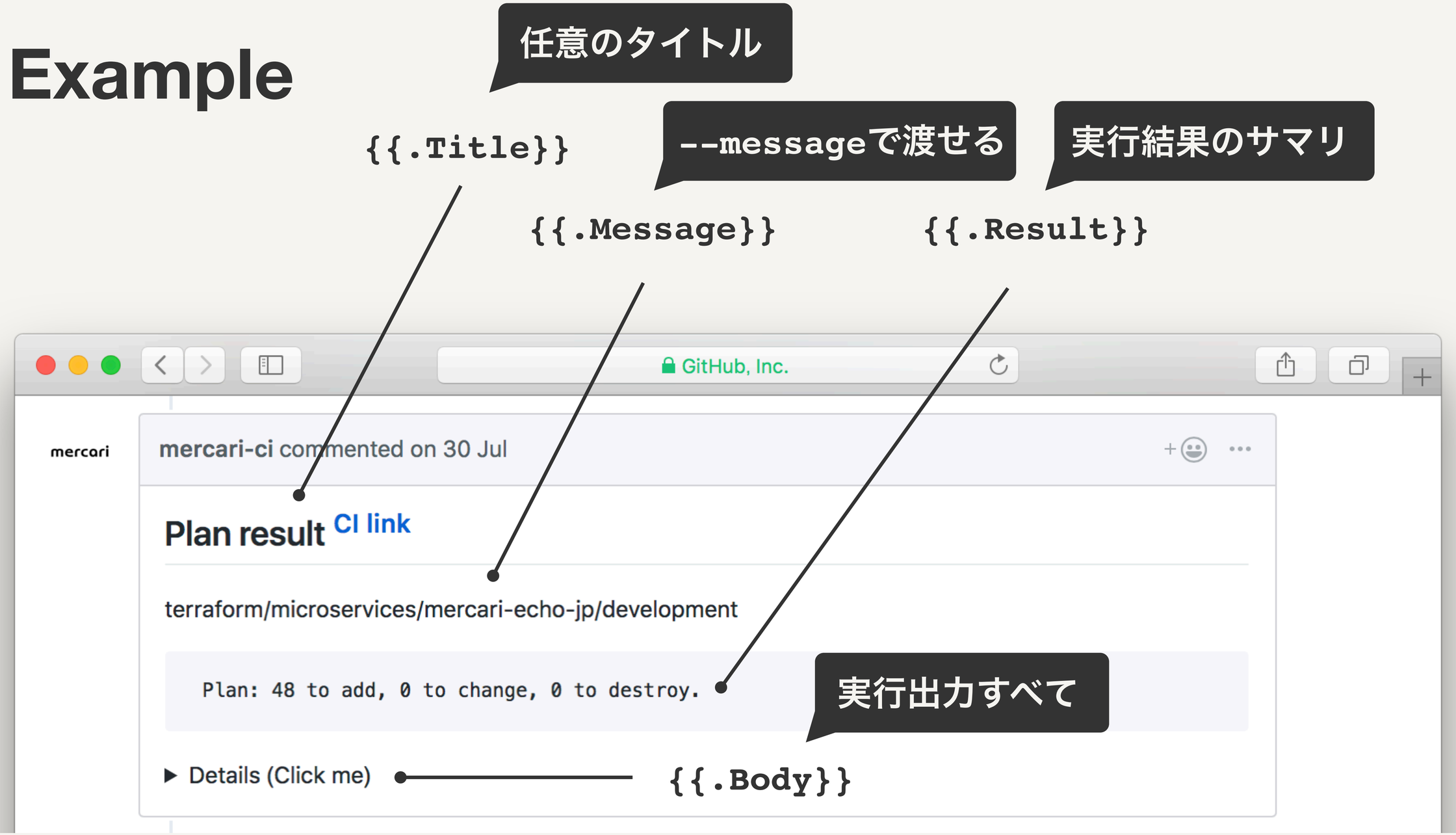

merged
POST comment
terraform apply | tfnotify apply
output
circleci
tfnotify

Example

- fmt
- plan
- apply



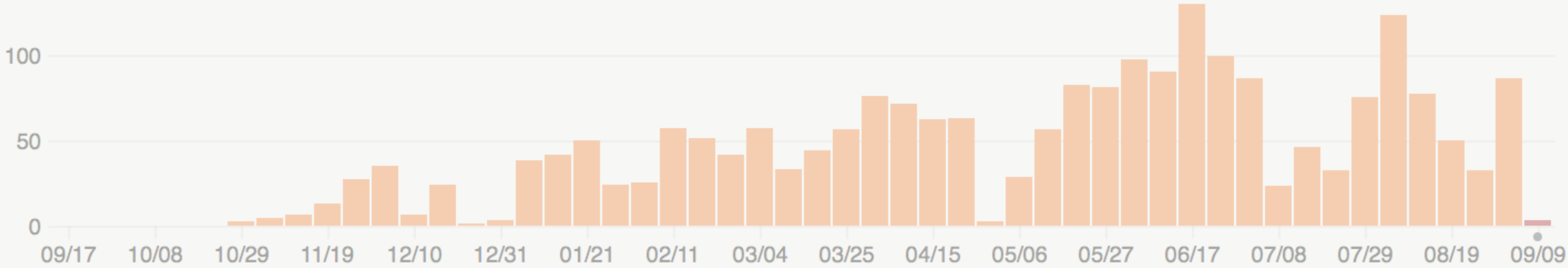
Example


任意のタイトル
{{.Title}}
--messageで渡せる
実行結果のサマリ
{{.Message}}
{{.Result}}
GitHub, Inc.
mercari
mercari-ci commented on 30 Jul
Plan result CI link
terraform/microservices/mercari-echo-jp/development
Plan: 48 to add, 0 to change, 0 to destroy.
実行出力すべて
▶ Details (Click me)
{{.Body}}

Why tfnotify?

なぜ必要となったのか

- メルカリでは Microservices 領域で Terraform を利用している
- Ownership の観点からインフラ管理においてもレビュー・マージは各 Micorservices チームによってされるべき
- とはいっても、Platform チームにレビューされたいケースもある

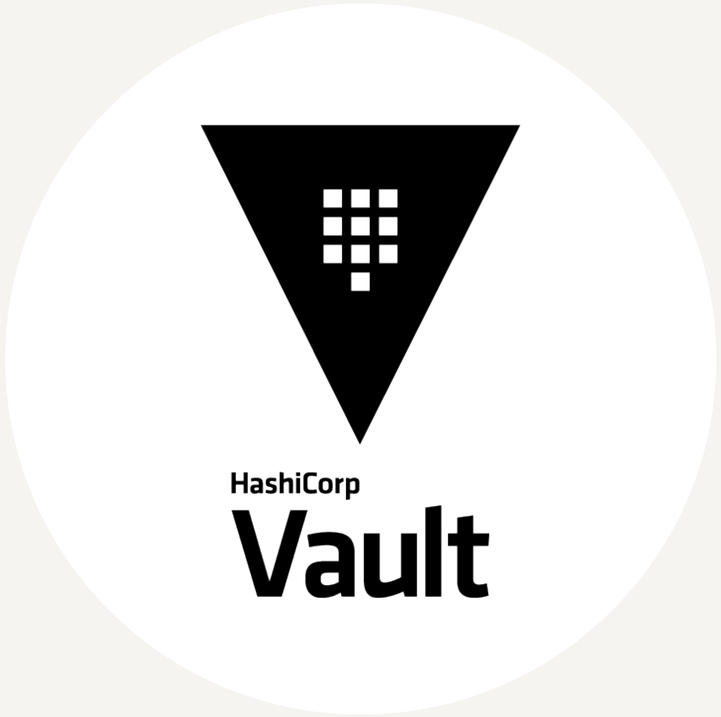


なぜ必要となったのか

- Platform チームとしても、各 Microservices チームとしても、Infrastructure as Code の重要性を理解し Terraform でコード化し、その実行計画を毎回見ることを習慣づけたい
- 毎日多くの P-R がある中、Circle CI に見にいく手間を省きたい

→レビューの流れで GitHub 上でクイックに確認したい

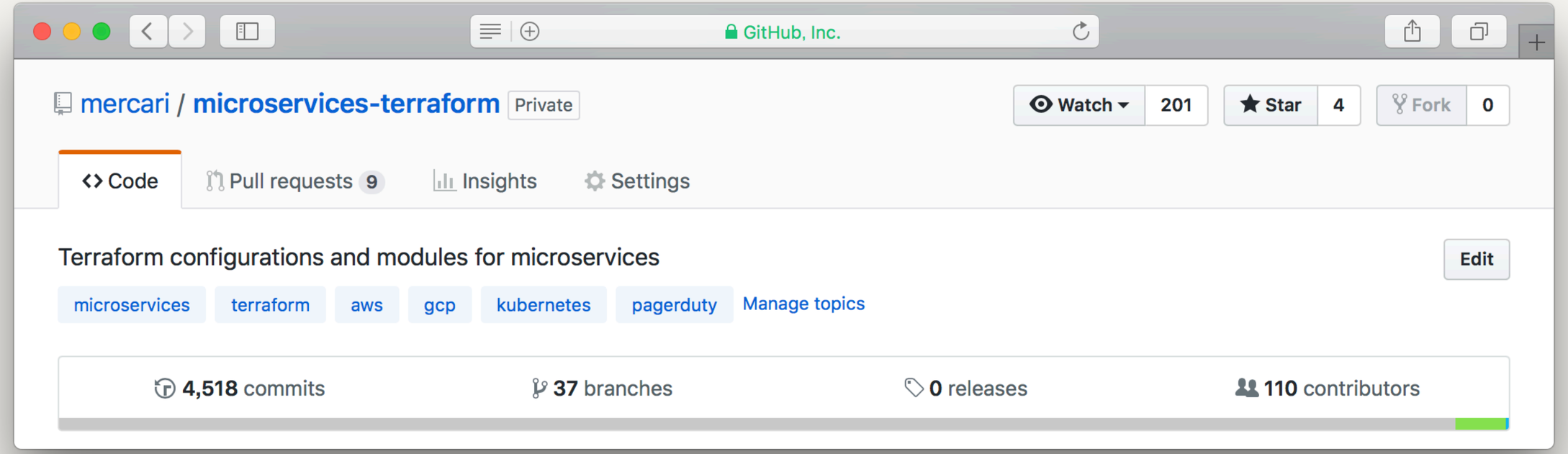
メルカリでの HashiCorp ツール



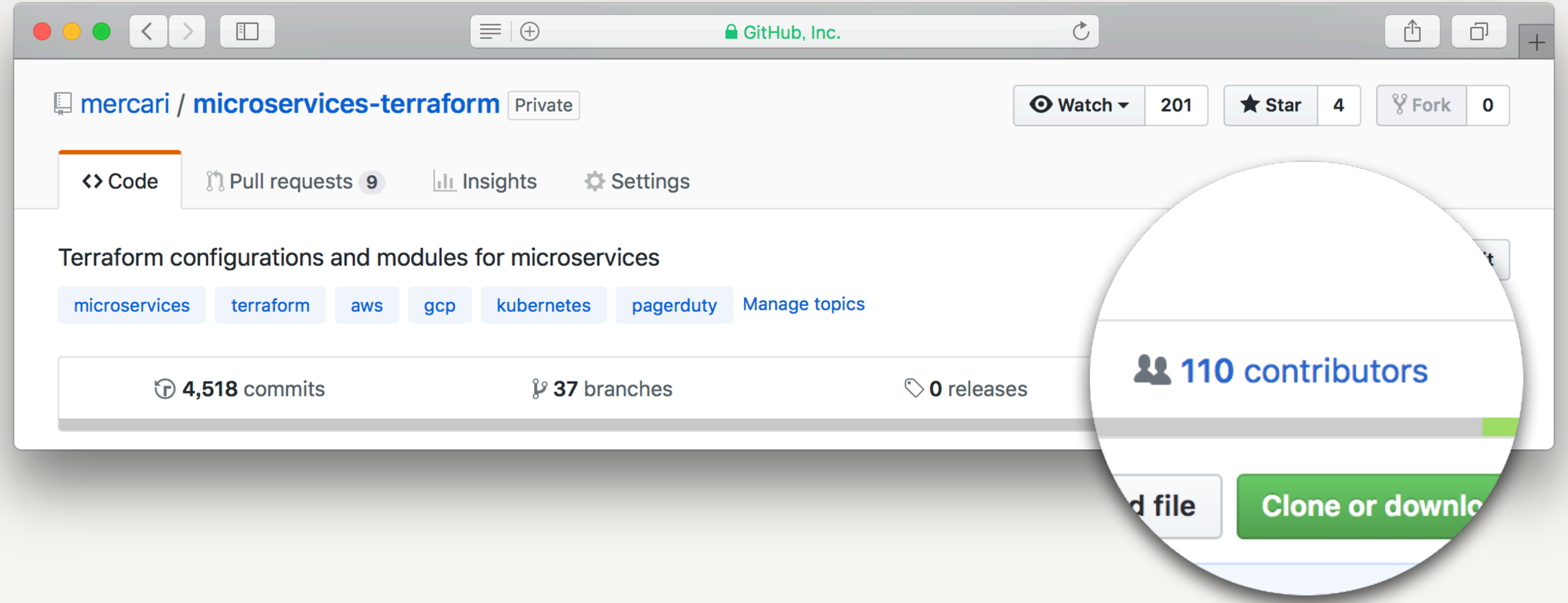
メルカリでの HashiCorp ツール

- メルカリには 70 以上の Microservices がある (今日現在．増え続けている)
- すべての Microservices とその Platform のインフラ構築には Terraform を利用している
- Developers に Infrastructure as Code を実践してもらう

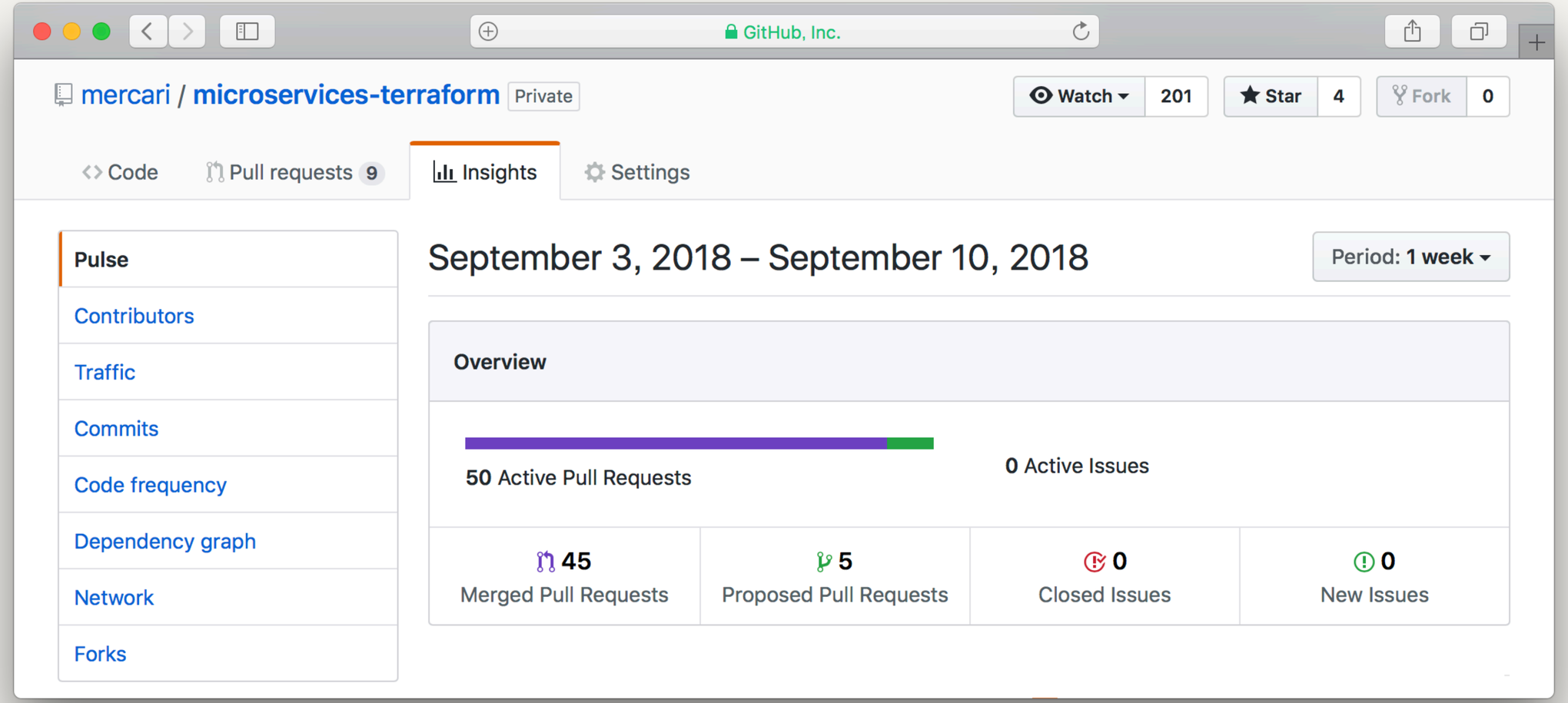
メルカリでの Terraform 利用事例



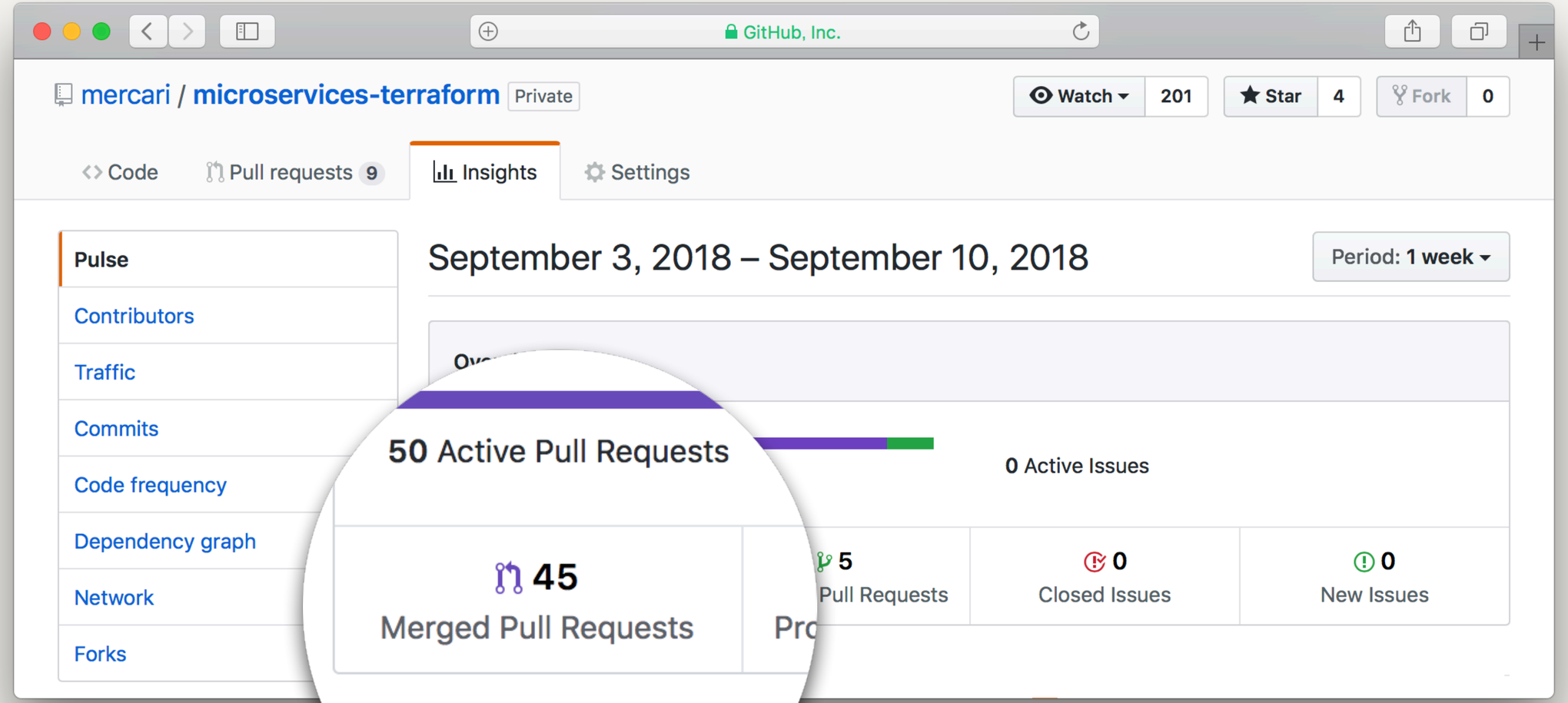
メルカリでの Terraform 利用事例



メルカリでの Terraform 利用事例



メルカリでの Terraform 利用事例



メルカリでの Terraform 利用事例

- Insights
  - 110+ Contributors
  - 8 ~ 10 Pull Requests / week (5 days)
  - 140+ state files (70+ Microservices * 2 Env)
- ひとつの中央リポジトリですべての Terraform コードを管理している
  - CI pipeline の構築が一度で済む
  - Platform チームがレビューに入りやすい

リポジトリ構成

- 各 Microservice ごとに
  ディレクトリを分ける
- Service ごとに tfstate を分ける
  - 影響を他へ波及させない
- Resource ごとに file を分ける
  - リソース定義場所を明確化
- CODEOWNERS で権限委譲

```
/terraform/microservices
├── mercari-echo-jp
│   ├── development
│   │   ├── backend.tf
│   │   ├── google_project_iam_member.tf
│   │   ├── google_storage_bucket.tf
│   │   ├── module_microservice_starter_kit.tf
│   │   ├── providers.tf
│   │   └── variables.tf
│   └── production
├── mercari-echo-us
│   ├── development
│   └── production
├── mercari-echo-gb
│   ├── development
│   └── production
└── merpay-echo
    ├── development
    └── production
```

リポジトリ構成

- 各 Microservice ごとに
  ディレクトリを分ける
- Service ごとに tfstate を分ける
  - 影響を他へ波及させない
- Resource ごとに file を分ける
  - リソース定義場所を明確化
- CODEOWNERS で権限委譲

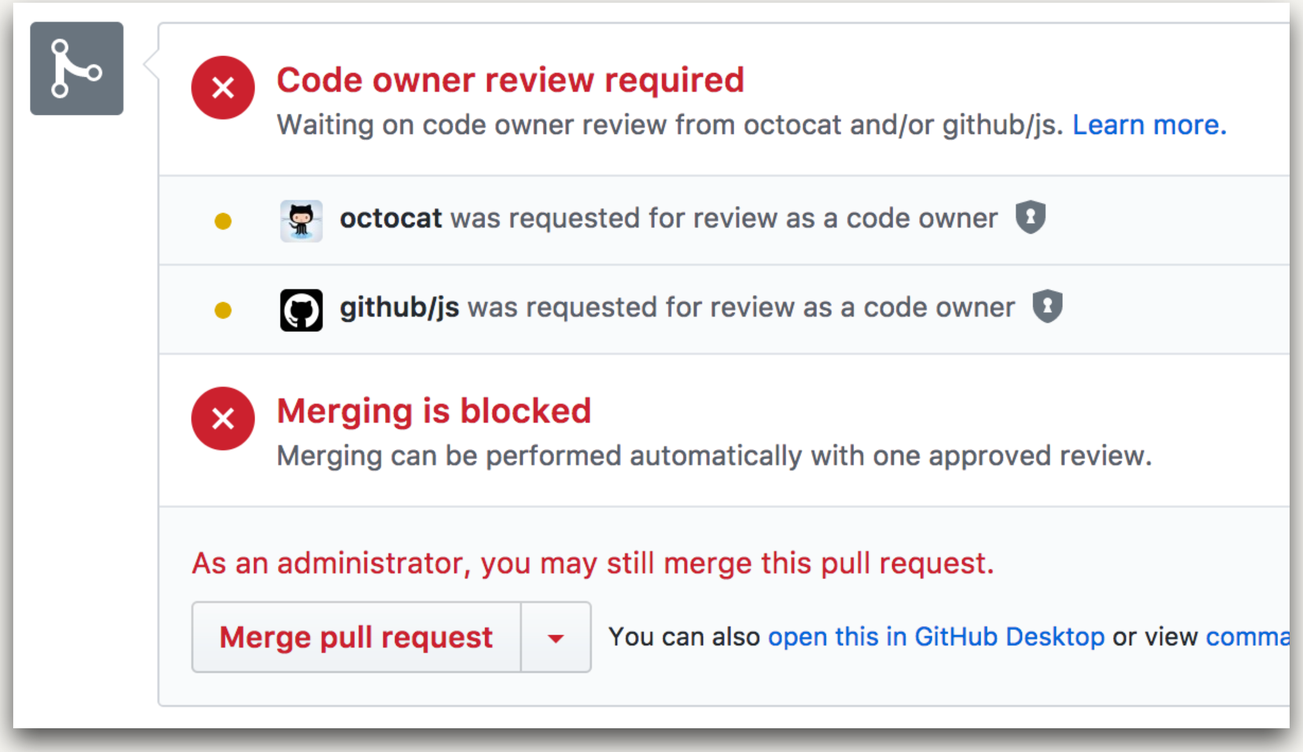
```
/terraform/microservices
├── mercari-echo-jp
│   ├── development
│   │   ├── backend.tf
│   │   ├── google_project_iam_member.tf
│   │   ├── google_storage_bucket.tf
│   │   ├── module_microservice_starter_kit.tf
│   │   ├── providers.tf
│   │   └── variables.tf
│   └── production
├── mercari-echo-us
│   ├── development
│   └── production
├── mercari-echo-gb
│   ├── development
│   └── production
└── merpay-echo
    ├── development
    └── production
```

自立分散

中央集権

権限委譲

- GitHub の機能
- CODEOWNERS に記載された人から Approve されるまでマージできないようにできる
- これにより権限委譲を実現
- (勝手に変更できない)



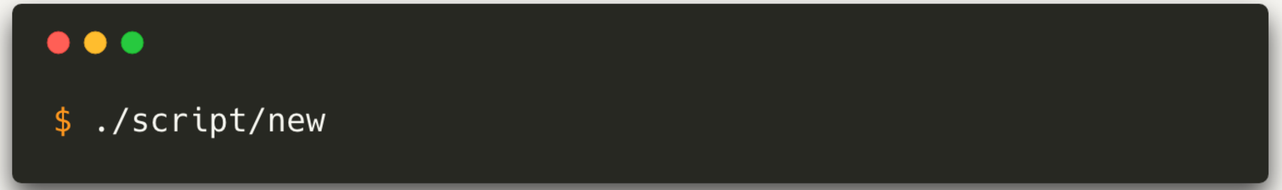
https://blog.github.com/2017-07-06-introducing-code-owners/

権限委譲

```
# /.github/CODEOWNERS
# For more detail about CODEOWNERS, refer to following articles:
#   - https://help.github.com/articles/about-codeowners/
#   - https://blog.github.com/2017-07-06-introducing-code-owners/

# These owners will be the default owners for everything in the repo.
* @mercari/microservices-platform

/terraform/microservices-platform/development/   @mercari/microservices-platform
/terraform/microservices-platform/production/    @mercari/microservices-platform

# mercari-echo-jp
/terraform/microservices/mercari-echo-jp/development/  @mercari/mercari-echo-jp-dev
/terraform/microservices/mercari-echo-jp/production/   @mercari/mercari-echo-jp-prod
```

https://help.github.com/articles/about-codeowners/

モジュール



- microservices-starter-kit (Terraform モジュール)
  - Template Provider を使いディレクトリ作成、ファイル吐き出し
  - Microservices の立ち上げに必要なリソースを Bootstrap する
    - GCP Project, Service account など
    - PagerDuty, DataDog, Kubernetes (Namespace, Secret) など
    - GitHub Teams (CODEOWNERS の権限委譲に使う)

https://speakerdeck.com/b4b4r07/terraform-ops-for-microservices

Implementations

実装

- io.TeeReader を使っている
  - GitHub に POST するだけではなく CI の Console にも出力
- Terraform の実行結果は自前のパーサで構造化する (regexp)
- POST するメッセージは Go のテンプレートで書くことができる
- 設定は YAML で持つ

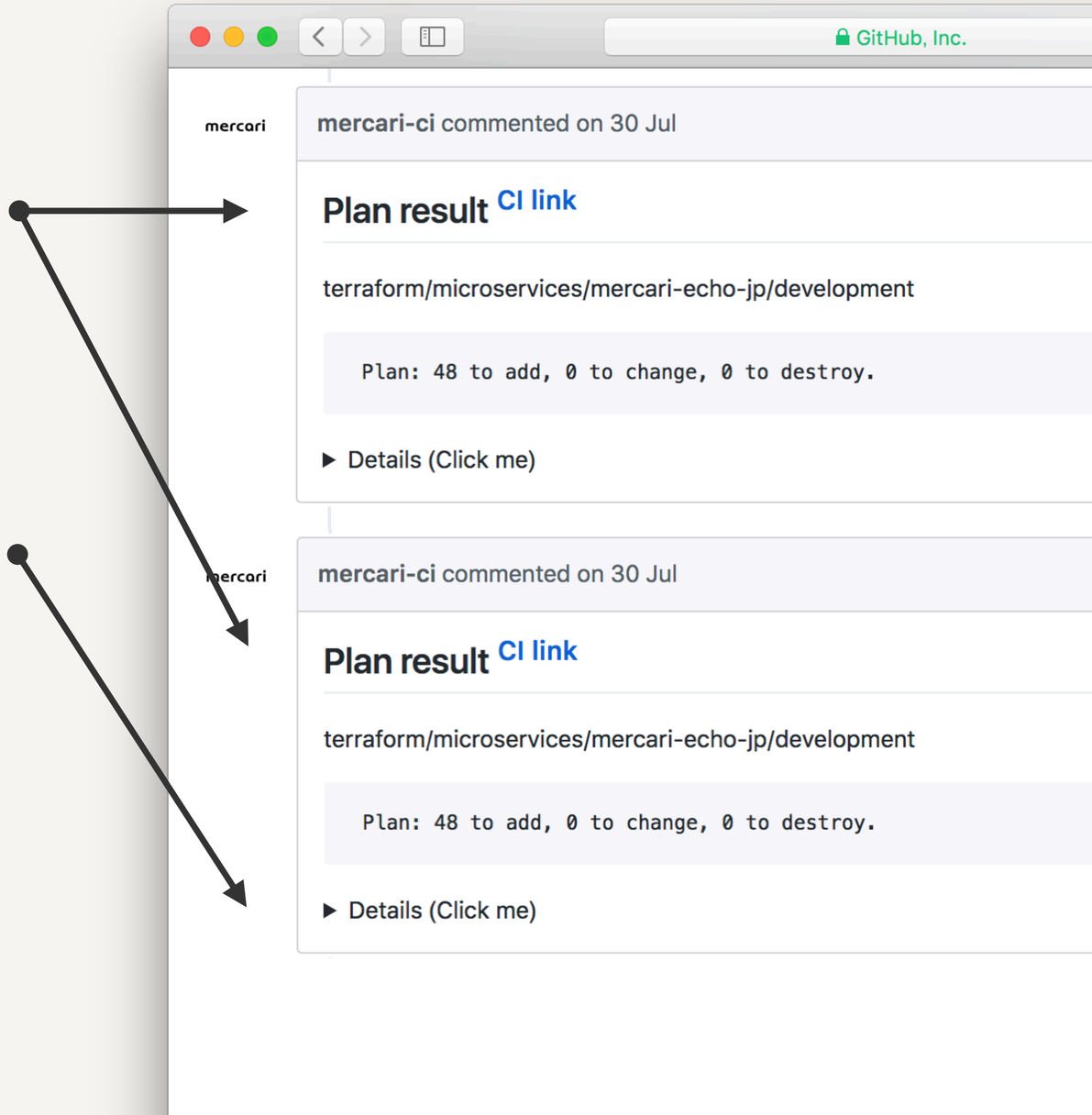
```
ci: circleci
notifier:
  github:
    token: $GITHUB_TOKEN
    repository:
      owner: "mercari"
      name: "tfnotify"
terraform:
  plan:
    template: |
      {{ .Title }}
      {{ .Message }}
      {{if .Result}}
      <pre><code> {{ .Result }}
      </pre></code>
      {{end}}
      <details><summary>Details (Click me)</summary>
      <pre><code> {{ .Body }}
      </pre></code></details>
```

実装

- 重複した内容がある場合、古いものが削除される

- 実行結果は長いので Details タグで囲って折りたたみ、一瞥に必要な情報だけが見やすく表示される

`{{.Title}}`, `{{.Message}}` の重複を見る



Conclusion

まとめ

- メルカリでは Microservices 領域で Terraform を利用している
  - 集権と分散のバランスで Terraform リポジトリを運用している
- たくさんの P-R を効率よくレビューするため tfnotify を書いた
  - Terraform の実行結果を手軽に確認したい
  - Infrastructure as Code の文化を根付かせる
  - 毎回 plan, apply 結果を見る習慣をもつ / もってもらう
- 先行実装がないため OSS にした

HashiCorp
Terraform

Thanks