GitHub Actions a vlastní stroje

Je to bezpečné?

Autor: Jiří Konečný

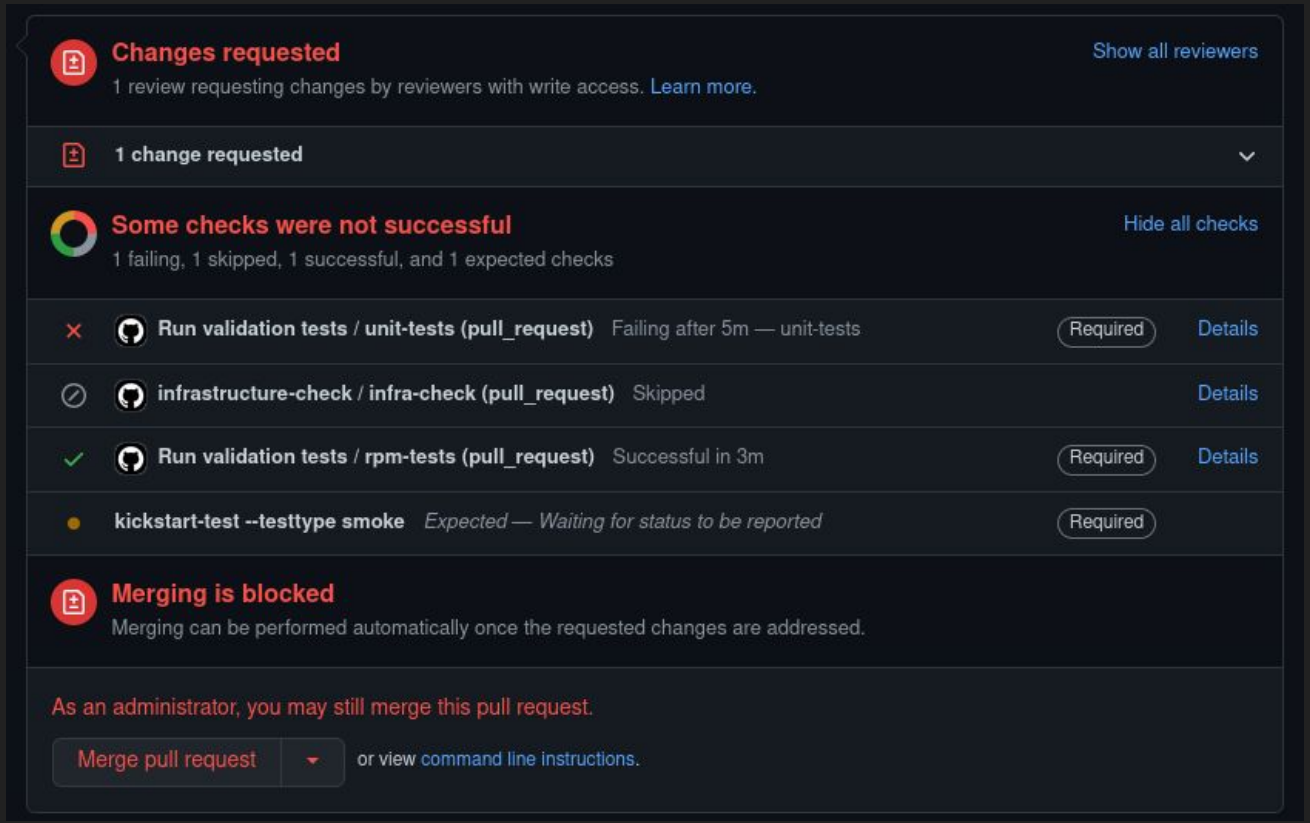
Agenda:

- Co jsou GitHub Actions?
- Na co si dát pozor s GitHub Actions
- Vlastní stroje pro GitHub Actions?
- Na co si dát pozor s vlastnímy stroji

Co jsou GitHub Actions?

- GitHub automatizace
- Použitelná prakticky na cokoliv
- Actions záložka

Co jsou GitHub Actions?



Co jsou workflow u GitHub Actions?

- Veřejné v repozitáři
- Kompletní popis automatizace
- .github/workflows/*.yml
- secrets.GITHUB_TOKEN / $GITHUB_TOKEN
- Workflow soubor

Na co si dát pozor

Na co si dát pozor s GitHub actions

Jaký je rozdíl mezi trigger *pull_request* a *pull_request_target*?

- Dokumentace

Na co si dát pozor s GitHub actions

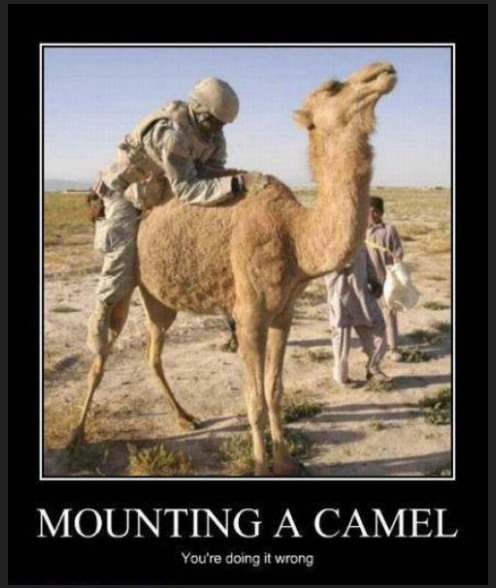
Jaký je rozdíl mezi trigger *pull_request* a *pull_request_target*?

“Ok, tak v tom případě jen vezmu pull_request_target, protože je bezpečnější.”

Na co si dát pozor s GitHub actions

Jaký je rozdíl mezi trigger *pull_request* a *pull_request_target*?

~~“Ok, tak v tom případě jen vezmu pull_request_target, protože je bezpečnější.”~~



Na co si dát pozor s GitHub actions

Jaký je rozdíl mezi trigger *pull_request* a *pull_request_target*?

- Dokumentace
- *pull_request* používá token přispěvatele (contributor)
  - Lepší vývoj Workflow souborů
  - Bezpečnější
- Nepoužívat *pull_request_target* pokud to není nutné (ale!)

Na co si dát pozor s GitHub actions

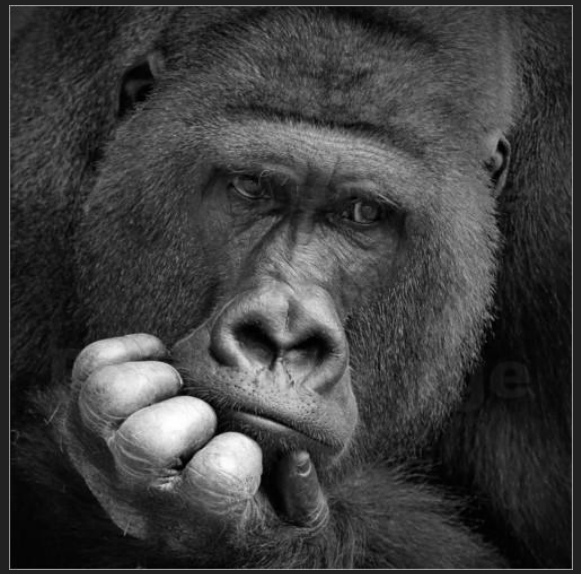
“Uživatel potřebuje mít možnost si pustit custom nastavení workflow. Hmm, tak co kdybych mu umožnil to říci komentářem a předal obsah komentáře aplikaci!”

Na co si dát pozor s GitHub actions

“Uživatel potřebuje mít možnost si pustit custom nastavení workflow. Hmm, tak co kdybych mu umožnil to říci komentářem a předal obsah komentáře aplikaci!”

To by šlo, ale!

Jak to děláme my a proč.



Vlastní stroje

Vlastní stroje pro GitHub Actions?

- Proč vlastní stroje?
- Výhody
  - Není nutná veřejná IP
  - Zdroje
  - Přístup k /dev/kvm (virtualizace)
- Nevýhody a omezení
  - Nutnost mít stroje a starat se o ně
  - Jednoduché se “střelit do nohy”
- **Doporučení od GitHubu je používat je pouze pro private repozitáře!**

Potencionální nebezpečí

“Však co, mám ten stroj pouze pro jeden workflow, který není spuštěný na PR. Co se může stát?”

Potencionální nebezpečí

~~“Však co, mám ten stroj pouze pro jeden workflow, který není spuštěný na PR. Co se může stát?”~~

Demo time (use-runner-on-another-workflow).

Potencionální nebezpečí

~~“Však co, mám ten stroj pouze pro jeden workflow, který není spuštěný na PR. Co se může stát?”~~

Jak tomu předejít?

- Používat všude pouze *pull_request_target* ale...

Potencionální nebezpečí

“Tak použijeme všude *pull_request_target* a je to, co by se mohlo stát?”

Potencionální nebezpečí

~~“Tak použijeme všude *pull_request_target* a je to, co by se mohlo stát?”~~

Demo time (attack-from-pull-request-target).

Potencionální nebezpečí

Jak se obecně chránit?

- Omezení permissions ve všech workflow.
- Vyžadovat potvrzení pro každého externího přispěvatele.
  - Staré řešení
  - Nové řešení
- GitHub se snaží situaci aktivně zlepšovat.
  - First time contributor approval.

Potencionální nebezpečí

Nejlepší řešení ze všech??

Potencionální nebezpečí

Nejlepší řešení ze všech??

Nepoužívat vlastní stroje!!!

Reference:

- https://docs.github.com/en/actions
- https://github.com/rhinstaller/anaconda/