GitOps

- Sokkal több, mint először gondoltam -

Fogas László
@laszlocph

Ops szemléletű fejlesztő vagyok

- 4 év Lufthansa Systems
  - Java, 2007-2010
- 5 év Falcon.io@Dánia
  - Java, Ops
  - Early employee
  - Középvezetésbe belekostólva
- 3 éve szabadúszó DevOps tanácsadó
  - Kubernetes, Docker, CI

Mi a GitOps?

- Nincs wikipedia szócikk
- Google-ben csak vendor cikkek jelennek meg az elmúlt pár évből
- Majd a Quora megment https://www.quora.com/What-is-GitOps

“Nothing new here. [...] Since everything in DevOps is “code” - it lives in Git end to end. Hence GitOps. Here’s a good marketing pitch for it”

“GitOps literally means using Git everywhere from development to deployment.”

GitOps naív definíció

“Szoftverfejlesztői gyakorlatok alkalmazása infrastruktúra menedzselésére.”

“* As Code”

SSH és egyszeri parancsok helyett

- Kóddal leírva a kívánt állapotot
- Többnyire deklaratívan
- Verziókezelve Git-ben
- Együttműködve Pull Request-ben

… kezeljük az infrastruktúrát.

Mit kapunk ekkor?

- Audit log-ot
- Peer review-t
- Megismételhetőséget
- Magabiztosságot
- Sebességet

De hisz erre már voltak szavaink

- Infrastructure as code - még Wikipedia szócikk is van :)

És ekoszisztémánk:

- Pets vs cattle, Immutable infrastructure
- Twelve factor app
- Configuration as Code - már a Jenkins-t is lehet így telepíteni
- Pipeline as code

És eszközeink:

- Ansible, Puppet, Chef, Terraform..

Ennyi az egész?

- Csak egy újabb buzzword?
- Csak egy re-branding?

Mi tette lehetővé?

- A felhős API-k hozták el az innovációt
- és a deklaratív leírók

Mi az, ami új most?

- Containerek
- egy API
- és egy deklaratív leíró hogyan nézzen ki a deployált szoftver

Kubernetes

GitOps

- Ahogy a WeaveWorks 2017-ben definiálta egy blogpostban

  “Operations by pull requests”

  “our developers operate Kubernetes via Git”

GitOps

- Egy eszköz mindenekfelett az infrastruktúra kezelésére
- Ismert eszköz, fejlesztő központú
- Nagyfokú automatizálás
- Verzió kezelve, audit log, rollback
- Security

A teljes kívánt rendszerállapot git alatt.

Fejlesztők, fejlesztők, fejlesztők

Fejlesztők olyan ops feladatokat végeznek, amiket korábban nem tudtak:

- DNS név változtatás
- Hány példányban fusson az alkalmazás
- Milyen földrajzi elosztásban

Mind-mind részese a leírónak és az automatizációnak

CIOps vs GitOps

- Gyakori, hogy az automatizációt, és kimondottan a deploy-t a CI pipeline végére rakjuk. Push szemantika.
- A GitOps a Pull szemantikát helyezi előtérbe: A CI folyamat vége az artifactok elkészítése és a leíró frissítése. De nem a deploy.
- Új komponenst vezet be, melynek feladata a Git állapot Kubernetes-re való eljuttatása.
- Implementációk: ArgoCD, Weave Flux, Jenkins X (?)

Pull szemantika előnyei

- CD deklarativvá válik
- CI alapvetően procedurális
  - A script-ek nem fednek minden esetet
  - CI vagy sikerül vagy nem. Nem próbálja helyreállítani a hibát
  - Részleges sikerből a helyreállítás magas fokú rendszer ismeretet igényel
- Kubernetes is hasonlót csinál: control theory
- Bonyolult feladatok triviálissá válnak:
  - DR
  - Deploy lock
  - Signed commits only
- A CI nem kap hozzáférést az éles rendszerhez

GitOps - a jövő?

- Weave Flux + ArgoCD = Argo Flux
- A pull szemantika teret hódít
- Standardizáció indul el

Összefoglalva:

- Illeszkedik a “* as code” trendbe
- A Kubernetes teszi lehetővé
- Fejlesztők ops feladatokat végezzenek
- Pull szemantika új lehetőségeket nyit

Kérdések?