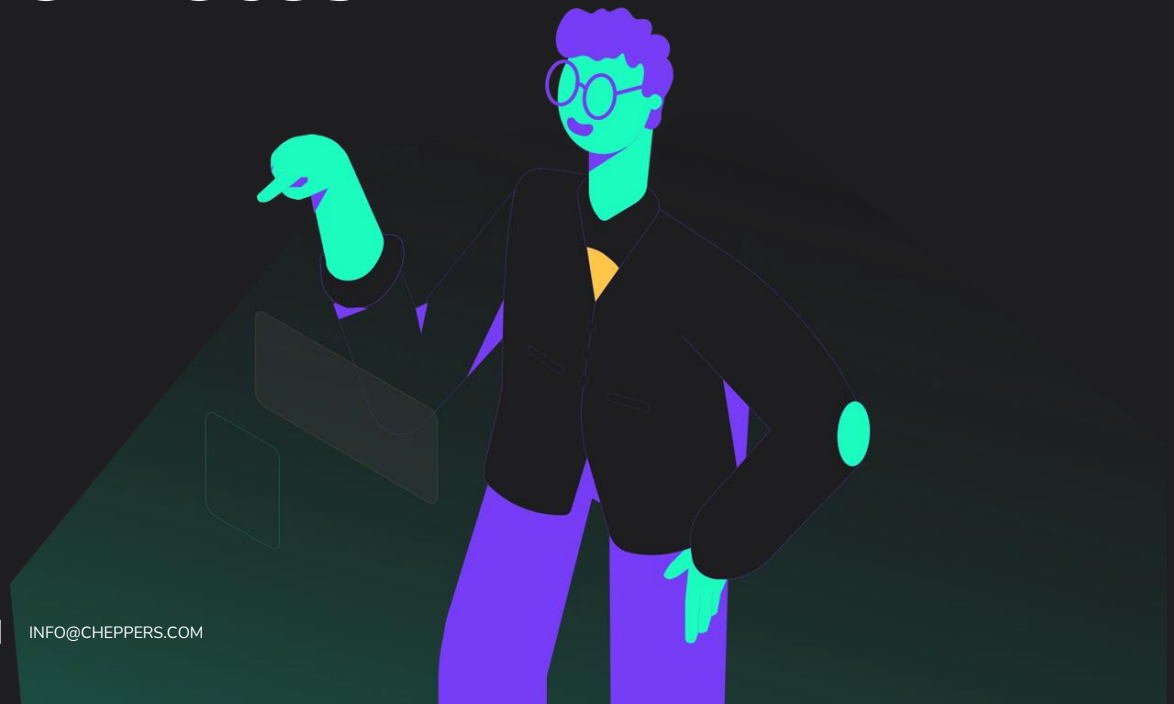
CHEPPERS

Konténer image készítés Kubernetes alatt



Martin Zsolt
cloud engineer

zsolt.martin@cheppers.com

Miről lesz szó?

CHEPPERS

1. Miért van erre szükség?
2. Kell-e a Docker?
3. Kaniko
4. Gitlab CI integráció

CHEPPERS

Az image készítés folyamata nem bonyolult...

CHEPPERS

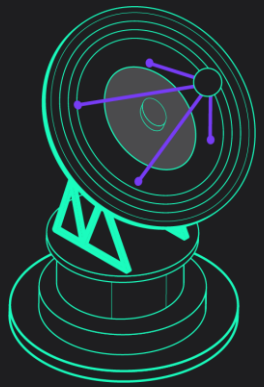
... kivéve ha konténerizálni akarjuk.

CHEPPERS

Docker-in-Docker

Megoldható...

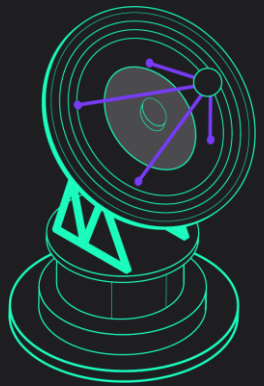
... de sosem ez volt a célja



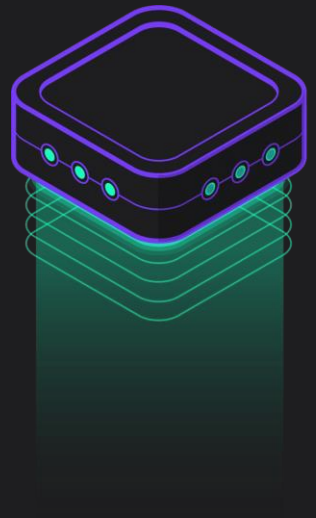
CHEPPERS

Docker-in-Docker

- Biztonsági kockázatok
- Teljesítmény problémák
- LSM inkompatibilitás



CHEPPERS



Docker socket mounting

Nem konténerizáljuk a Dockert, csupán a hoston futó Docker daemonra csatlakozunk.

CHEPPERS

Docker socket mounting

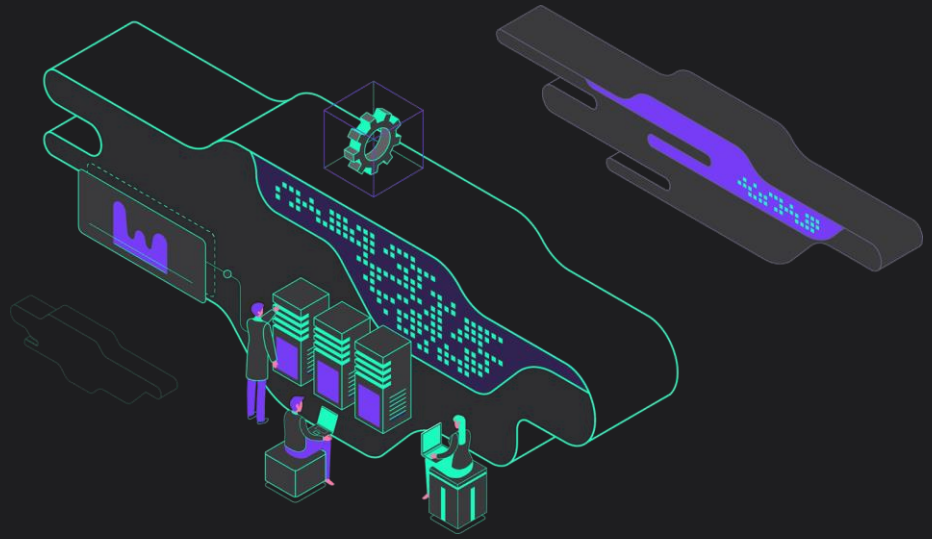
Működik a cache

Nincsenek mellékhatások

Biztonsági kockázatok

Kubernetes nehézségek

CHEPPERS



Docker és a Kubernetes

- Kubernetes 1.24: a Docker vége
- AWS Fargate támogatás hiánya

CHEPPERS

AWS Fargate

Automatikus számítási kapacitást biztosít a konténerek számára

Teljesen menedzselt worker node-ok EKS alatt

CHEPPERS

AWS Fargate

- Minden pod teljesen izoláltan fut
- Nem támogatott a privileged mód
- Nem támogatott a socketek felcsatolása
- Nem elérhető a Docker daemon

Automatikus számítási kapacitást biztosít a konténerek számára

Teljesen menedzselt worker node-ok EKS alatt

CHEPPERS

Dobjuk ki a Dockert!

CHEPPERS

Kaniko

https://github.com/GoogleContainerTools/kaniko

“kaniko is a tool to build container images from a Dockerfile, inside a container or Kubernetes cluster”

CHEPPERS

Kaniko

- Kizárólag konténerben működik
- A teljes build folyamat userspace-ben fut
- Nem daemon, minden build feladat külön Kaniko konténerben történik
- Build context alapján dolgozik, mint a Docker

CHEPPERS

Kaniko

IMAGE

Helper scriptek

Debug verzió busybox shellel

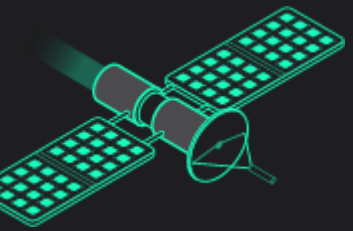
Minimalista

Futtatás más image-ben nem támogatott

Kaniko

CHEPPERS

MŰKÖDÉS



1. Dockerfile felolvasása
2. Kiinduló image letöltése és kitömörítése
3. Minden utasítás végén snapshot készítése a fájlrendszerről
4. Módosítások hozzáfűzése új rétegként a kiinduló image-hez
5. Az elkészült image feltöltése a registrybe

Kaniko

CHEPPERS

CACHE

Image rétegek

- RUN, COPY parancsok
- Konténer registryben tárolja
- Minden utasítás előtt ellenőrzi, hogy létezik-e már

Kiinduló image

- Lokális könyvtárban tárolja a szükséges image-eket
- Cache előtöltés szükséges (warmer)

Kaniko

CHEPPERS

CACHE

Image rétegek

- RUN, COPY parancsok
- Konténer registryben tárolja
- Minden utasítás előtt ellenőrzi, hogy létezik-e már

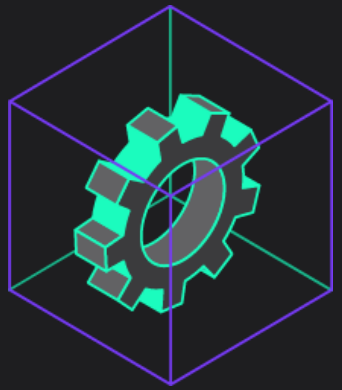
Kiinduló image

- Lokális könyvtárban tárolja a szükséges image-eket
- Cache előtöltés szükséges (warmer)

CHEPPERS

Kaniko

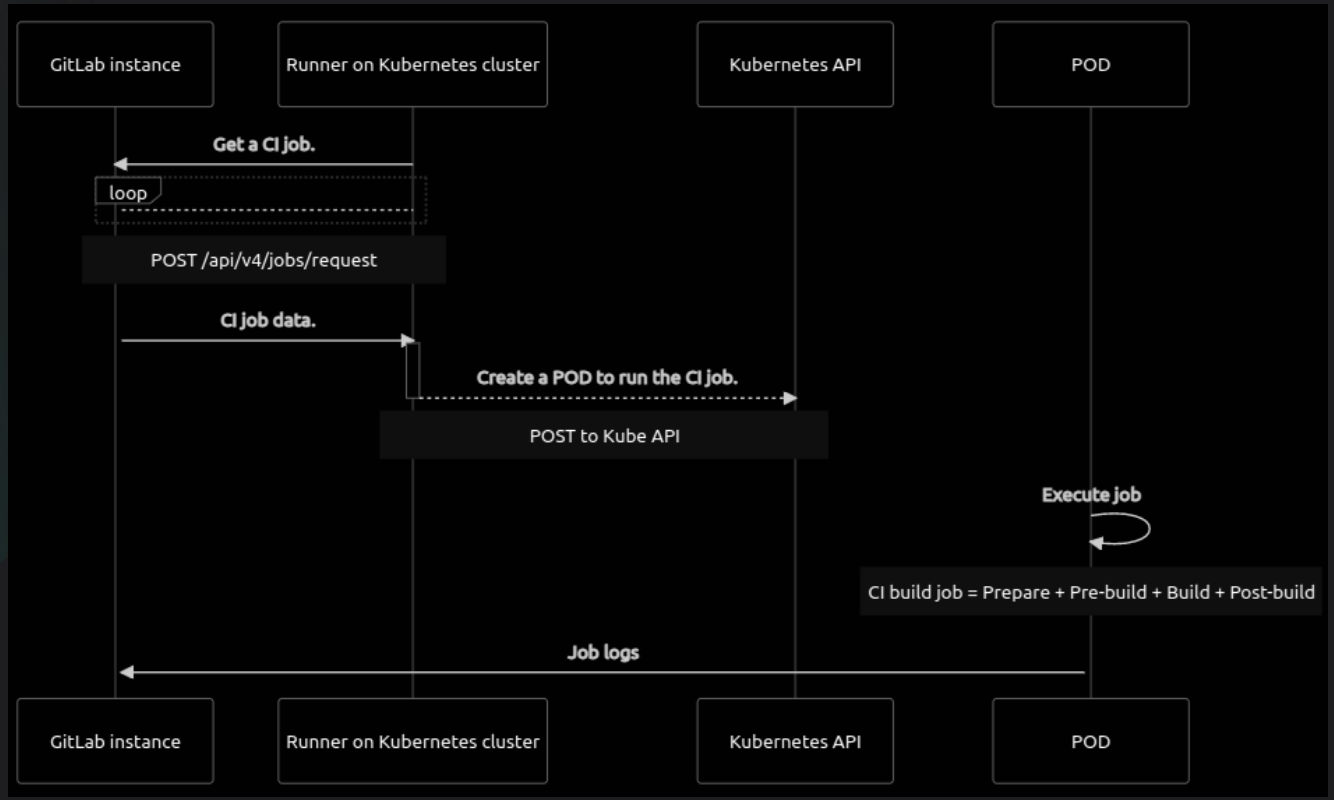
BIZTONSÁG



- Nincs szükség privileged módra
- Root userként fut (általában)
- Érdemes az éles alkalmazásoktól elszeparálva futtatni

Gitlab CI integráció

CHEPPERS



CHEPPERS

Gitlab CI

RUNNER

```
[[runners]]
  [runners.kubernetes]
    image = "gcr.io/kaniko-project/executor:v1.8.1-debug"
    allowed_images = ["gcr.io/kaniko-project/executor:*"]
    namespace = "gitlab-runner"
    privileged = false
    allow_privilege_escalation = false
    cap_drop = ["ALL"]
    cap_add = ["CHOWN", "DAC_OVERRIDE", "FOWNER", "SETFCAP", "SETGID", "SETUID"]
    memory_request = "1Gi"
    memory_request_overwrite_max_allowed = "4Gi"
    memory_limit = "1Gi"
    memory_limit_overwrite_max_allowed = "4Gi"
    cpu_request = "0.5"
    cpu_request_overwrite_max_allowed = "2"
    cpu_limit = "0.5"
    cpu_limit_overwrite_max_allowed = "2"
    [[runners.kubernetes.volumes.pvc]]
      name = "kaniko-cache"
      mount_path = "/cache"
```

CHEPPERS

Gitlab CI

JOB

```yaml
build-with-kaniko:
  stage: build
  tags:
    - eks-kaniko
  variables:
    KUBERNETES_CPU_REQUEST: 1
    KUBERNETES_CPU_LIMIT: 1
    KUBERNETES_MEMORY_REQUEST: 2Gi
    KUBERNETES_MEMORY_LIMIT: 2Gi
  before_script:
    - mkdir -p /kaniko/.docker
    - echo "{\"credsStore\":\"ecr-login\"}" > /kaniko/.docker/config.json
  script:
    - /kaniko/warmer --image "alpine:3.15" --image "composer:2.2"
    - >-
      /kaniko/executor
      --context "${CI_PROJECT_DIR}"
      --destination "${ECR_REPOSITORY}:${CI_COMMIT_SHORT_SHA}"
      --destination "${ECR_REPOSITORY}:${CUSTOM_TAG}"
      --cache=true
```

CHEPPERS

Köszönöm a figyelmet!

Martin Zsolt
zsolt.martin@cheppers.com