Migrace Pickey.cz z VPS do Kubernetes v cloudu

Ondřej Šika & David Slačálek

ondrej@sika.io
@ondrejsika

Posobota, 28. 5. 2022

Ondřej Šika

Jsem DevOps lektor, architekt a konzultant z Prahy.

Navrhnu a implementuji Vám na míru DevOps architekturu od verzování v Gitu po provoz v Cloudu.

Dělám populární školení, kde své znalosti předávám tak, abyste si mohli vše udělat sami a bez zbytečných přešlapů a slepých cest.

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Vypsané termíny školení

- **Kubernetes** (Praha & Online)
  30.-31.2022, 15 900 CZK
- **Cloudflare** (Praha & Online)
  1.6.2022, 8 900 CZK
- **Terraform** (Praha & Online)
  2.6.2022, 8 900 CZK
- **Úvod do Go (Golang)** (Praha & Online)
  6.-7.6.2022, 15 900 CZK

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Píšu knihu o DevOps

kniha.sika.io

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

David Slačálek

Jsem backend vývojář.

Mám trému.

S kamarádkou Denisou Hrubešovou jsme založili Pickey.cz.

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Pickey.cz

Platforma pro monetizaci digitálního obsahu.

Tvůrce digitálního obsahu se soustředí pouze na produkci a promo, veškeré technikálie zajišťuje Pickey.

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Gordíkey

@gordikey

5 přispěvatelů

Jsem pes (obviously), jmenuju se Gordíkey a stará se tu o mě @diseasezoe - zakladatelka @pickey
Jsem kříženec labradora s ridgebackem.
Cizí lidi mi říkaj "štěndo labrouše", ale páníčci to nesnášej.

Na granulkey

59 Kč /měsíc

Rád jim

Na steaky

99 Kč /měsíc

Hodně rád jim steaky

Na věčná loviště

199 Kč /měsíc

Ruku na srdce, je mi už 13 a

Problém

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Problém

- Spolehlivost / Dostupnost
- Zero time deployment
- Škálovatelnost
- Udržitelnost & technologický dluh

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Spolehlivost / Dostupnost

- Nepříjemné výpadky poskytovatel VPS
- Chyběla jednodnoduchá možnost HA řešení

Zero time deployment

- Současná migrace probíhá pomocí RSYNC
- Dočasně nekonzistentní kód na serveru (v průběhu nasazování)
- Deploy v noci (v době malé návštěvnosti)
- Žádná graceful termination long term jobů

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Škálovatelnost

- Očekávaný růst
- Škálování s výpadkem (nutný restart VM)
- Možnost autoscalingu (v budoucnu)

Udržitelnost & technologický dluh

- Příprava platformy na růst teamu
  - Použití standardních technologií a přístupů (Kubernetes, ...)
  - Docker Compose pro lokální vývoj
- Jednodušší onboarding

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Řešení

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Řešení

Přesunout Pickey.cz do Kubernetes v Cloudu

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

O čem to dnes bude

- Technologický stack
- Původní stav
- Nový stack
- Změny v aplikaci
- Změny v deploymentu
- Fuckupy
- Výsledek

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Technologický stack

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Technologický stack

- Backend
  - PHP 8 + Laravel
- Frontend
  - Vue.js + Nuxt
- Data
  - Postgres
  - Složka s uploads

Technologický stack

- 3rd party services
  - MailGun
  - Mux
  - Pusher
  - GP Webpay

O čem to dnes bude

- ~~Technologický stack~~
- Původní stav
- Nový stack
- Změny v aplikaci
- Změny v deploymentu
- Fuckupy
- Výsledek

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Původní stav

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Původní stav

- 2x VPS ON na Wedosu
  - Produkce + Dev prostředí na vlastním VM
- Vše instalované ručně na servery
- CD pomocí Bitbucket Pipelines
- Databáze (Postgres) na stejném VM jako PHP
- Uploads ve složce na VM

Proč na VPS bez fancy DevOps shitu?

- MVP!
- Launch ASAP
- Žádná znalost Dockeru, Kubernetes, Cloudu
- Engineering team: pouze David
- Cena

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

O čem to dnes bude

- ~~Technologický stack~~
- ~~Původní stav~~
- Nový stack
- Změny v aplikaci
- Změny v deploymentu
- Fuckupy
- Výsledek

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Nový stack

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Nový stack

- Cloud: Digital Ocean
- Docker + Kubernetes
- IaC: Terraform
- Managed Postgres, S3
- Gitlab + Gitlab CI
- Cloudflare
- Tergum

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Cloud: Digital Ocean

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Cloud: Digital Ocean

- Jednoduchý na ovládání (i Terraformem)
- Cena: Levnější než AWS
- Poskytuje vše, co potřebujeme
  - Managed Kubernetes, LoadBalancing
  - Managed Postgres (+Redis)
  - S3

Docker + Kubernetes

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Docker + Kubernetes

- Standard, co se týká provozu aplikací
  - Není důvod použít nic jiného ;)
- Build in škálování, HA
- Managed Kubernetes
- Oddělené Dev & Prod clustery
  - Dev + build: 2x 2CPU, 4RAM
  - Prod: 2x 4CPU, 8RAM

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

IaC: Terraform

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Co je IaC?

Infrastructure as code (IaC) je způsob, jakým spravovat veřejné i privátní cloudy pomocí deklarativně definovaných konfiguračních souborů, které používají nástroje pro správu infrastruktury jako zdroj pravdy.

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

IaC: Terraform

- Chceme spravovat infrastrukturu formou kódu
  - Jednodušší
  - Bezpečnější
  - Udržitelnější
- Terraform
  - Nejrozšířenější nástroj pro IaC

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

```
resource "digitalocean_droplet" "example" {
  image    = "debian-10-x64"
  name     = "example"
  region   = "fra1"
  size     = "s-1vcpu-1gb"
}
```

```
resource "digitalocean_record" "example" {
  domain = "example.com"
  type   = "A"
  name   = digitalocean_droplet.example.name
  value  = digitalocean_droplet.example.ipv4_address
}
```

DB: Managed Postgres

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

DB: Managed Postgres

- Nechceme se starat o databázový server
- Někdo jiný to umí mnohem lépe
- Oddělené DB servery
  - Dev: 1CPU, 1RAM
  - Prod: 1CPU, 2RAM
- Backup resi DO i my (Tergum)

S3

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

S3

- Ukládání dat v S3 (v DigitalOceanu je to spaces)
- Kubernetes neumožňuje ukládání uploads ve složce
- Škálování
- Možnost servírovat statické soubory mimo aplikaci

Gitlab a Gitlab CI

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Gitlab + Gitlab CI

- Gitlab (gitlab.com)
  - Potřebujeme rozumnou DevOps platformu (na místo Bitbucketu)
  - Podpora ukládání Terraform state
- Gitlab CI
  - Docker (Kaniko) buildy + nasazování do Kubernetes
  - Deploy infrastruktury (ještě nemáme :/)

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Cloudflare

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Cloudflare

- Při migraci jsme se rozhodli přejít s DNS ke Cloudflare
- Jednoduchá správa pomocí Terraformu
- Stabilní DNS provider (pryč z Wedosu a z Česka)
- DDoS ochrana, CDN, ...
- Zdarma

Tergum

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Tergum (declarative backup tool)

- github.com/sikalabs/tergum
- Deklarativní backup tool
- Běží v Kubernetes
- Backend je S3

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

tergum.yml

```yaml
Meta:
  SchemaVersion: 3
Backups:
  - ID: example-postgres-server
    Source:
      PostgresServer:
        Host: 127.0.0.1
        Port: 5432
        User: postgres
        Password: pg
        Database: postgres
    Targets:
      - ID: s3
        S3:
          AccessKey: XXX
          SecretKey: YYYY
          Region: eu-central-1
          BucketName: backups
          Prefix: postgres
          Suffix: html.zip
```

tergum-pickey-noreply@sikalabs.io
[tergum][pickey-all] Backup Summary -- OK
To: monitoring@ondrejsika.com

SUCCESS	BACKUP	BACKUP TIME	TARGET	UPLOAD TIME	FILE SIZE	ERROR	TIME TOTAL
OK	Postgres: db-dev	2s (+5s)	S3: spaces	0s (+0s)	28.7M		7s
OK	Postgres: db-prod	4s (+4s)	S3: spaces	0s (+0s)	17.8M		8s

BACKUP	TARGET	ERROR

-- tergum

O čem to dnes bude

- ~~Technologický stack~~
- ~~Původní stav~~
- ~~Nový stack~~
- Změny v aplikaci
- Změny v deploymentu
- Fuckupy
- Výsledek

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Změny v aplikaci

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Změny v aplikaci

- Soubory z FS do S3
- Sessions v DB / Redisu
- Cache do Redisu
- Optimalizace SQL

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Soubory z FS do S3

- Uživatelské soubory
- Temp
- Automaticky aktualizované soubory
- Intervention/image orientate!

Konfigurovatelné v .env pro lokální vývoj

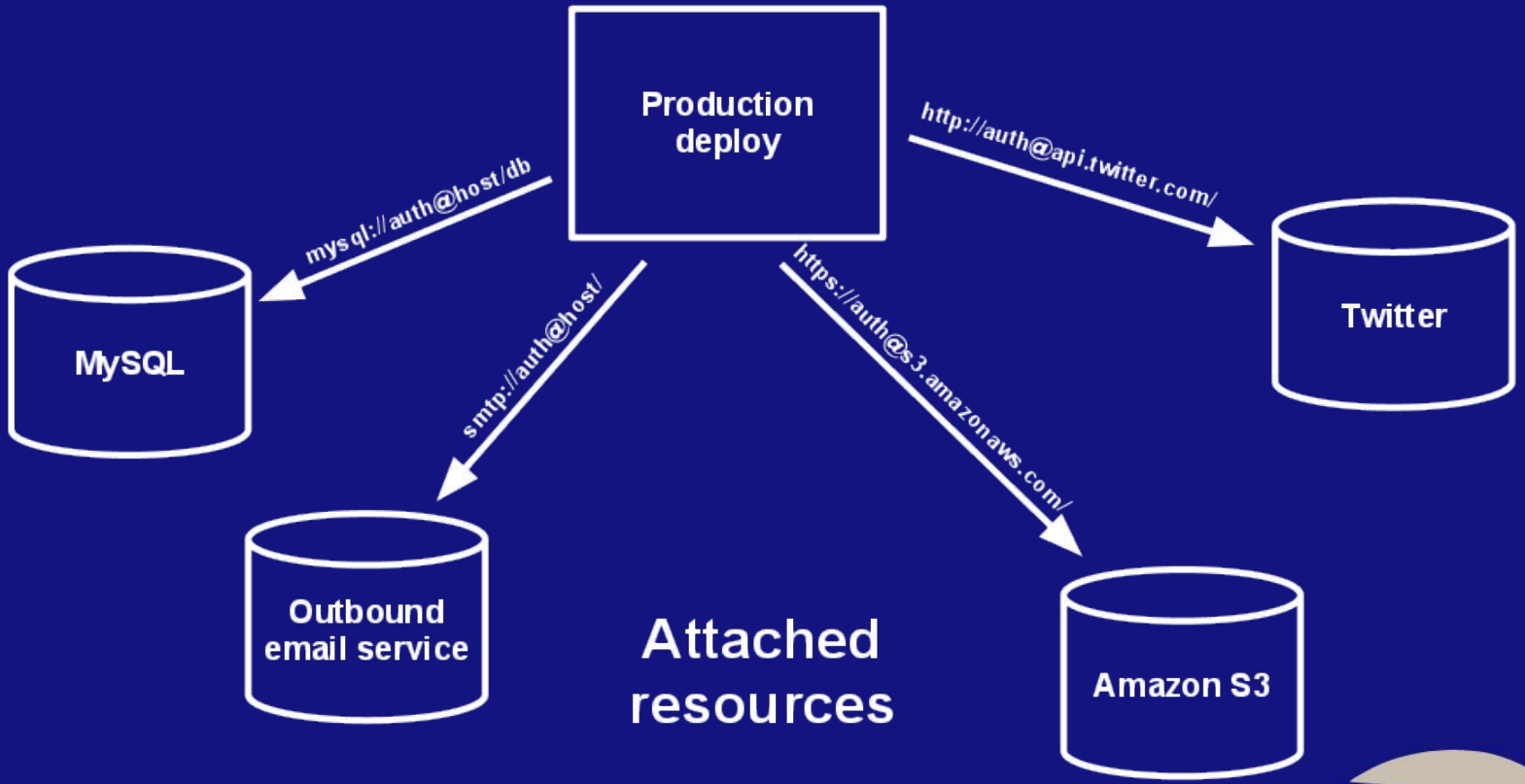
@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Sessions v DB

- Je nutné ukládat session na místo, odkud budou přístupné pro všechny instance aplikace
- Nemáme jednu instanci ani sdílený FS mezi instancemi

Cache do Redisu

- V Kubernetes není sdílený FS pro rychou cache
- Cache chceme k aplikaci připojit jako backing service (viz 12factor apps)
- Redis je ideální NoSQL DB / cache



Attached resources

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Optimalizace SQL

- Databáze nemá nulovou latenci
- Neoptimalizované dotazy se velmi znatelně projeví

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

O čem to dnes bude

- ~~Technologický stack~~
- ~~Původní stav~~
- ~~Nový stack~~
- ~~Změny v aplikaci~~
- Změny v deploymentu
- Fuckupy
- Výsledek

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Změny v deploymentu

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Změny v deploymentu

- Docker
- Kubernetes
- Bitbucket Pipelines do Gitlab CI
- Správa konfigurace & secrets

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Docker

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Docker

- Optimalizace buildu v Dockeru
  - Optimalizace vrstev v Docker images (caches)
    - Nutné pochopení, jak a co při build procesu funguje
- Frontend: Build 2 images (dev & prod)
- Build Docker Image bez Dockeru - Kaniko

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Kubernetes

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Kubernetes

- Deployment aplikace je rozdělen do nasazení tří Helm balíčků:
  - pickey-data
  - pickey-api (PHP backend)
  - pickey-app (Nuxt.js frontend)

Kubernetes: pickey-data

- Nasazení Redisu pro queue a cache
- Oddělený balíček, aby při celkové preinstalaci aplikace (helm uninstall) instance Redisu zůstala

Kubernetes: pickey-api

- Obsahuje
  - Deployment PHP backendu (2 instance)
  - Deployment workeru na zpracovávání fronty jobu (horizon)
  - Cronjob (task scheduler) ... TODO

Kubernetes: pickey-app

- Obsahuje
  - Deployment frontend JS aplikace (2 instance)
  - SSR: Node.js server pro render prvního dotazu na stránku

Z Bitbucket Pipelines do Gitlab CI

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Z Bitbucket Pipelines do Gitlab CI

- Přepis kompletně znova
  - Dříve deployment probíhal jako upload PHP souboru pomocí RSYNC
- Gitlab CI běží v dev Kubernetes clusteru
- Kaniko build + Helm install
  - Gitlab CI má nativní podporu pro Docker & Kubernetes

O čem to dnes bude

- ~~Technologický stack~~
- ~~Původní stav~~
- ~~Nový stack~~
- ~~Změny v aplikaci~~
- ~~Změny v deploymentu~~
- Fuckupy
- Výsledek

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Fuckupy

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Fuckupy

- Platební brána
- Velikost Redis PVC
- Build Nuxt.js bez ENV
- Výpadek DB
- Časový odhad

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Platební brána: špatná konfigurace

- Číslo není string - špatná expanze velkého integeru do stringu (helm)
- Platební brána špatnou konfiguraci akceptovala
- Neseděly podpisy při validaci transakcí

Velikost Redis PVC

- Neodhadli jsme, kolik místa potřebuje Redis pro uložení fronty jobu
- Shodili jsme redis, a rozbila se fronta tasku - některé emailové notifikace jsme poslali 8x 🤦
- Oprava byla jednoduchá, zvětšili jsme PVC (+ nasadíme monitoring)

Build Nuxt.js aplikace vyžaduje ENV

- Na produkci nám po nasazení nefungovalo Google Analytics
- Nuxt.js potřebuje při buildu aplikace konfiguraci 🤦
  - To znamená, že Docker image musí obsahovat při buildu konfiguraci, což je antipattern 🤦
  - Pokud víte, jak to vyřešit, rádi se přiučíme 🙏

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Výpadek DB

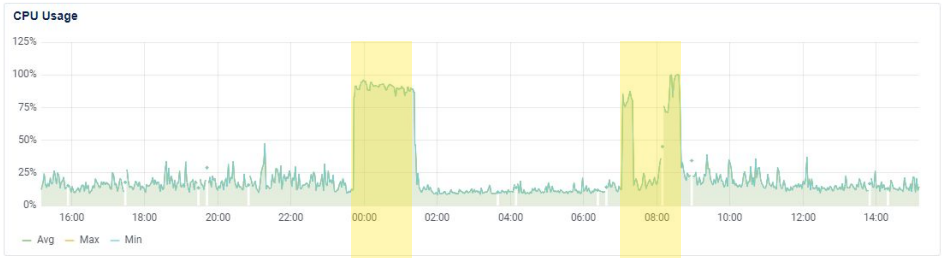
- Obrovské množství sekvenčních scanů
- 8 000 000 000 row reads

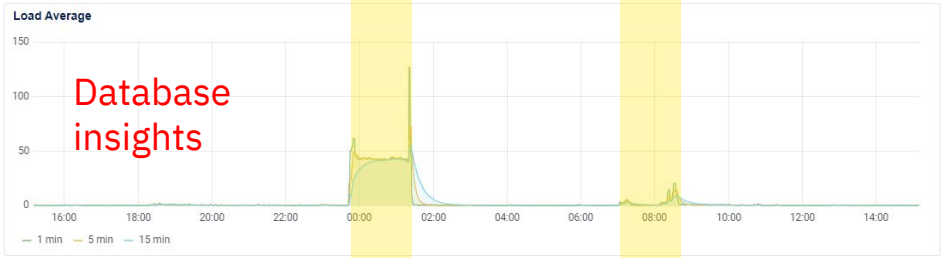
=> velká CPU zátěž

Eloquent - polymorfní relace

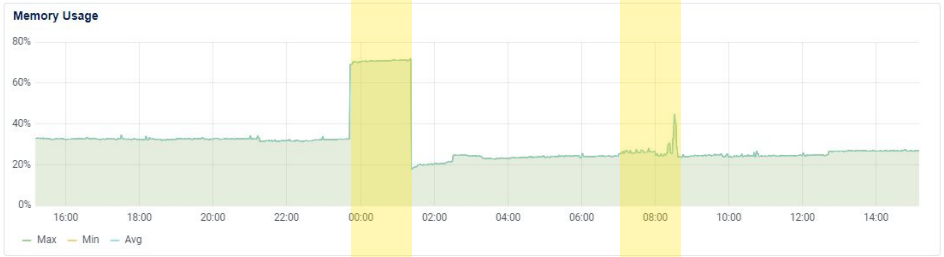
```php
//PHP: Global scope - zpracovaná média příspěvku
$builder->whereHas('detail', function (Builder $builder)
{
      $builder->where('has_media_processed', '=', true);
});

--SQL: redundant query
SELECT DISTINCT "detail_type" FROM "posts";
```

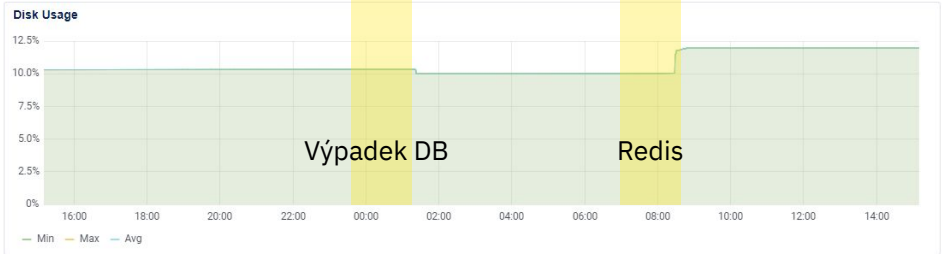
```php
$builder->whereRaw(<<<SQL
  posts.detail_id = (
    case
      when posts.detail_type = 'photo'
        then (select id from detail_photos where has_media_processed = true and id = posts.detail_id)
      when posts.detail_type = 'video'
        then (select id from detail_videos where has_media_processed = true and id = posts.detail_id)
      when posts.detail_type = 'text'
        then (select id from detail_texts where has_media_processed = true and id = posts.detail_id)
      when posts.detail_type = 'poll'
        then (select id from detail_polls where has_media_processed = true and id = posts.detail_id)
      when posts.detail_type = 'live'
        then (select id from detail_livestreams where has_media_processed = true and id =
posts.detail_id)
    end
  )
SQL);
```





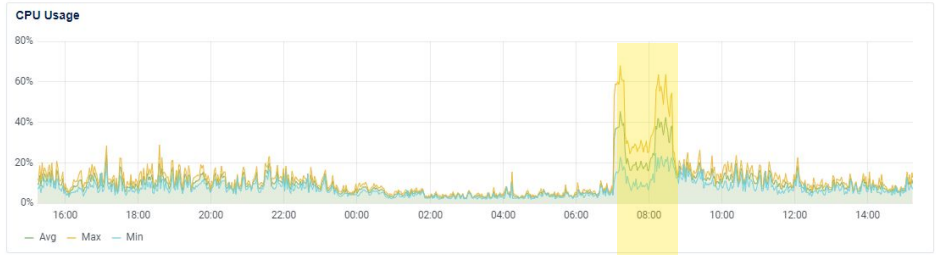
Database insights

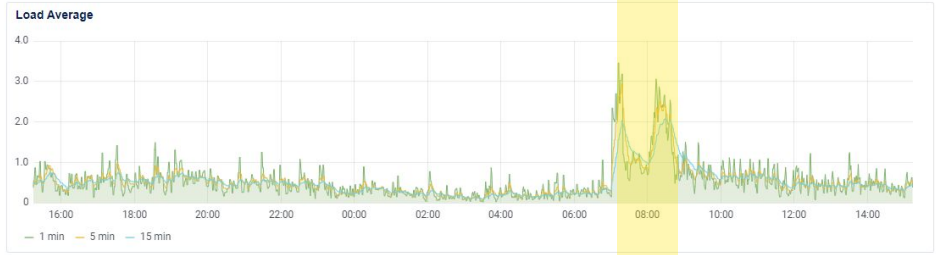


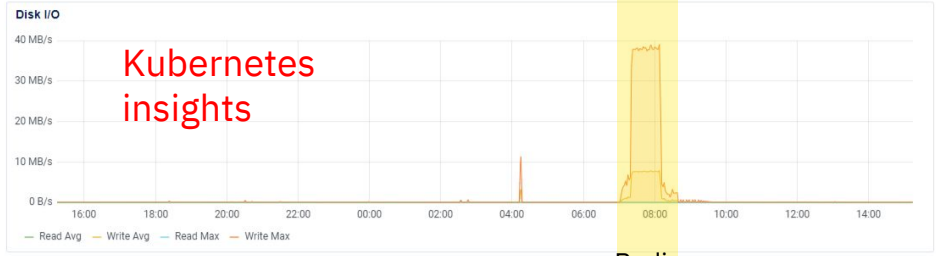


Výpadek DB

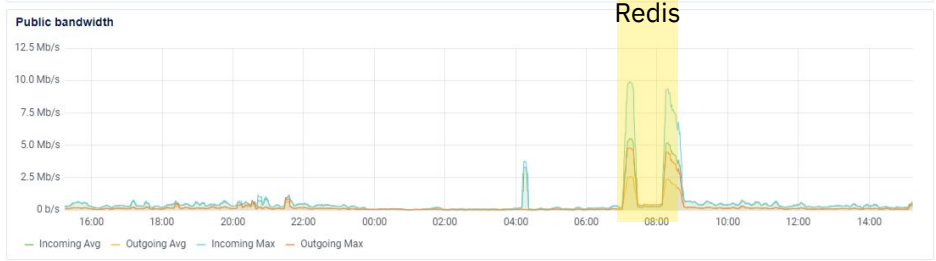
Redis







Kubernetes insights



Redis

Časový odhad & Learning new shit

- Z plánovaných cca 2 dnů bylo asi 6+ (překročení plánu o 300%)
- Engineering team (David) & operations team (opět David) se musí naučit:
  - Docker
  - Kubernetes
  - Cloud (+ Terraform)

O čem to dnes bude

- ~~Technologický stack~~
- ~~Původní stav~~
- ~~Nový stack~~
- ~~Změny v aplikaci~~
- ~~Změny v deploymentu~~
- ~~Fuckupy~~
- Výsledek

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Výsledek

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Shrnutí

- Migrace splnila očekávání (krom časové náročnosti)
- Škálovatelné HA řešení
- Unifikovaný deployment stack nad Kubernetes
- Odstranění velké části technologického dluhu
- Jednodušší onboarding vývojářů díky Dockerizaci

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

… ale

- Pomalejší, než na VPS díky latenci k DB
  - Na VPS nebyla potřeba tolik řešit optimalizace, fungovalo to

Do budoucna

- Optimalizace nejen SQL
- Monitoring
- GitOps

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Díky za pozornost

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

Otázky?

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

ondrej@sika.io

david@pickey.cz

www.pickey.cz

sika.link/slides

Slides

sika.link/slides

What's next?

sika.link/next

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika