

VPWA: Cvičenie DevOps, 3. časť - Nasadenie v cloude #2 (CDN, S3, RDS, Azure)

Ciele cvičenia

- Oboznámiť sa s nasadzovaním kontajnerizovanej aplikácie na platforme AWS - pokračovanie
 - AWS RDS (manažovaná relačná DB Postgresql)
 - AWS CloudFront (CDN)
 - AWS S3 (object storage)
 - Prepojenie služieb a architektúra (3. prednáška k DevOps časti ako doplnkový materiál - diagramy, teória).
- Vyskúšať si nasadenie na niektorú cloud computing platformu (MS Azure, Digital Ocean, AWS a i.)

Prerekvizity

- Zostavené a funkčné Docker images server + PWA klient, pripravené na nasadenie
 - Konfigurovateľné premenné prostredia (ENV)
 - Zabezpečená perzistencia dát (RDBMS)

Návod na získanie kreditov pre Digital Ocean, MS Azure (GitHub Student Developer Pack)

1. Zaregistrujte si [GitHub Student Developer Pack](#), využite univerzitnú @stuba mailovú adresu.
2. Dokončíte registráciu na Digital Ocean (alebo MS Azure) a prihláste sa do služby.

Úloha č. 1: Oboznámte sa s nasadzovaním na platforme AWS ECS (Elastic Container Service) - pokračovanie

V spolupráci s cvičiacim sa oboznámte s nasadzovaním kontajnerizovanej webovej aplikácie (klient, server, RDBMS databáza) na platformu AWS ECS.

- AWS RDS (manažovaná relačná DB Postgresql) - východisko pre vysokú dostupnosť,

replikáciu, zálohovanie...

- AWS CloudFront (CDN) - optimalizácia doručovania obsahu klientom + vstupná brána pre klientov (ochrana pred DDoS).
- AWS S3 (object storage) - úložisko pre statické assety (HTML, CSS, JS, media)
- Prepojenie služieb a architektúra

Úloha č. 2: Nasadte vašu aplikáciu na zvolenú cloudovú platformu (návod pre Azure)

Vyberte si niektorú cloud platformu (AWS, Digital Ocean, MS Azure, Google Cloud) a vykonajte nasadenie vašej aplikácie (postačuje nasadenie Docker kontajnerov na niektorú orchestračnú platformu - AWS ECS, Azure Container Instances a pod.). Stručný návod pre MS Azure:

1. Prihláste sa do svojho účtu a chodte na [Azure Home](#).
2. Vytvorte si repozitár pre Docker obrazy - [Container Registries](#).
 - Vytvorte nový repozitár
 - Chodte do sekcie Access Keys - povoľte Admin user, bude vám vygenerované používateľské meno a heslo.
 - Prihláste sa do vytvoreného repozitára príkazom `docker login vaserepo.azurecr.io`
 - Tagnite váš Docker image a odošlite ho do Azure repozitára

```
docker tag pwa-client vaserepo.azurecr.io/pwa-client
```

```
docker push vaserepo.azurecr.io/pwa-client
```

3. Vytvorte kontajnerizované inštancie pre BE časť vašej aplikácie - [Azure Container instances](#):
 - Postgresql - použitie Docker image z Docker Hubu (referencia: 'postgres'). Nezabudnite nastaviť premenné prostredia pre inicializáciu databázy. Pre jednoduchosť nemusíte zabezpečovať perzistenciu dát, ktorá prežije zmazanie kontajnera.
 - Slek-server - použite vami vytvorený image z Azure Container Registries. Nezabudnite nastaviť premenné prostredia, špeciálne endpoint vami nasadeného Postgresql (IP adresa alebo DNS).
4. Nasadte FE časť vašej aplikácie - [Azure App Services](#):
 - Slek-client - použite vami vytvorený image z Azure Container Registries. Nezabudnite, že Quasar musí byť zostavený s korektným parametrom API_URL (endpoint, kde ste nasadili Slek-server na Azure Container Instances).

5. Pomôcka - Docker compose:

```
version: '3.5'
services:
  slek-client:
    image: slekrepo.azurecr.io/slek-client:latest
```

restart: unless-stopped

ports:

- 80: 80

environment:

HOST: 0.0.0.0

PORT: 80

slek-server:

image: slekrepo.azurecr.io/slek-server:latest

restart: unless-stopped

ports:

- 3333: 3333

environment:

HOST: 0.0.0.0

PORT: 3333

APP_KEY: fgkjdfjgkdfjjdkflg

DRIVE_DISK: local

NODE_ENV: production

DB_CONNECTION: pg

PG_HOST: postgres

PG_PORT: 5432

PG_USER: docker

PG_PASSWORD: dockerpass

PG_DB_NAME: docker

postgres:

image: postgres

restart: unless-stopped

ports:

- "5432: 5432"

container_name: "postgres"

volumes:

- postgres: /var/lib/postgresql/data

environment:

POSTGRES_USER: "docker"

POSTGRES_PASSWORD: "dockerpass"

POSTGRES_DB: "docker"

volumes:

postgres:

name: pg- volume

Revision #5

Created Wed, May 4, 2022 6:34 PM by Adam Puskas

Updated Wed, May 4, 2022 4:34 PM by Adam Puskas