

# 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 Postgres) - databáza
  - AWS CloudFront (CDN) - klient
  - AWS S3 (object storage) - klient
  - AWS ECS (Docker containers) - server
  - Prepojenie služieb a architektúra (3. prednáška k DevOps časti - diagramy, teória).
- Vyskúšať si nasadenie na niektorú cloud computing platformu (MS Azure, AWS a pod.)

## 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 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 platforme AWS ECS.

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

- AWS CloudFront (CDN) - optimalizácia doručovania obsahu klientom (statické assety - obrázky, CSS a pod.) + vstupná brána pre klientov (ochrana pred DDoS).
- AWS S3 (object storage) - úložisko pre statické assety (HTML, CSS, JS, media)
- Prepojenie služieb, architektúra a konfigurácia - siete a bezpečnostné skupiny (VPC), cache policy (CDN), static website hosting (S3 + CloudFront), DB snapshots (RDS), service update (ECS) a i.

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

Vyberte si niektorú cloud platformu (AWS, MS Azure, Digital Ocean a pod.) a vykonajte nasadenie vašej aplikácie (postačuje nasadenie Docker kontajnerov na niektorú orchestračnú platformu - AWS ECS, Azure Container Apps a pod.). 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
  - Po rozkliknutí repozitára chodte do sekcie Access Keys - povoľte Admin user, bude vám vygenerované používateľské meno a heslo.
  - Prihláste sa do vytvoreného repozitára príkazom `docker login <vase_repo>.azurecr.io`
  - Tagnite váš Docker image a odošlite ho do Azure repozitára

```
docker tag pwa-client <vase_repo>.azurecr.io/slek-client
docker push <vase_repo>.azurecr.io/slek-client
```

**slekclient | Access keys** ☆ ...

Container registry

Search << Registry name slekclient

Overview Login server slekclient.azurecr.io

Activity log Admin user ☒ Enabled

Access control (IAM) Username slekclient

Tags

Quick start

Events

Settings

Access keys

Name	Password	Regenerate
password		
password2		

3. Vytvorte kontajner pre BE časť vašej aplikácie - [Azure Container Apps](#)
  - Basics - vyberte reosurce group, nastavte meno kontajnera, vyberte región pre nasadenie.
  - App Settings - odškrtnite "use quick start image", vyberte cestu k vášmu slek-server v **Azure Container Registries** repozitári, zvolte požadovanú **konfiguráciu inštancie** (postačuje 0,25 CPU, 0,5 GB RAM). Nastavte tiež **premenné prostredia** - pozor na cestu k

databáze (DB\_HOST, resp. PG\_HOST). Pre začiatok môžete nastaviť "localhost", po následnom spustení databázy na Azure (napr. ako ďalší kontajner) **nezabudnite túto premennú prostredia aktualizovať**. Povoľte tiež **Ingress - Accepting traffic from anywhere**, vystavíte tak službu, aby bola dostupná odkiaľkoľvek. Nezabudnite **špecifikovať port** (napr. 3333).

## Create Container App ...

Name \*

slek-server

Image source

☒ Azure Container Registry

☐ Docker Hub or other registries

Registry \*

slekserver.azurecr.io

Image \*

slek-server

Image tag \*

latest

OS type

Linux

Command override ⓘ

Example: /bin/bash, -c, echo hello; sleep 100000

Container resource allocation

CPU and Memory \*

0.25 CPU cores, 0.5 Gi memory

### Environment variables

+ Add

Name	Value	Delete
HOST	0.0.0.0	

## Application ingress settings

Enable ingress for applications that need an HTTP endpoint.

Ingress ⓘ	<input checked="" type="checkbox"/> Enabled
Ingress traffic	<div><input type="radio"/> <b>Limited to Container Apps Environment</b></div> <div><input type="radio"/> <b>Limited to VNet:</b> Applies if 'internalOnly' setting is set to true on the Container Apps environment</div> <div><input checked="" type="radio"/> <b>Accepting traffic from anywhere:</b> Applies if 'internalOnly' setting is set to false on the Container Apps environment</div>
Ingress type	<div><input checked="" type="radio"/> HTTP</div> <div><input type="radio"/> TCP</div>
Transport	<div>Auto</div>
Insecure connections	<input checked="" type="checkbox"/> Allowed
Target port * ⓘ	<div>3333</div>

4. Vytvorte konrajner pre databázu, použite pritom službu [Azure Container Instances](#). Nastavte inštanciu podľa obrázkov nižšie. Po rozkliknutí vytvorenej inštalácie nájdete endpoint v sekcii Overview - IP address (public).

Container details

Container name \* ⓘ

slek-postgres-2✓

Region \* ⓘ

(Europe) North Europe▼

Availability zones ⓘ

None▼

**i** The selected region does not support Availability Zones.

Image source \* ⓘ

☐ Quickstart images

☐ Azure Container Registry

☒ Other registry

Image type \* ⓘ

☒ Public

☐ Private

Image \* ⓘ

postgres✓

**i** If not specified, Docker Hub will be used for the container registry and the latest version of the image will be pulled.

OS type \*

☒ Linux

☐ Windows

**i** This selection must match the OS of the image chosen above.

Size \* ⓘ

1 vcpu, 1.5 GiB memory, 0 gpus  
[Change size](#)

Networking type

☒ Public

☐ Private

☐ None

DNS name label ⓘ

✓

DNS name label scope reuse ⓘ

Any reuse (unsecure)▼

Ports ⓘ

Ports	Ports protocol	
5432✓	TCP▼	🗑
	▼	

Restart policy ⓘ On failure

Environment variables

Mark as secure	Key	Value
No	POSTGRES_USER	slek
No	POSTGRES_PASSWORD	slekpass
<input type="text" value="No"/>	<input type="text" value="POSTGRES_DB"/>	<input type="text" value="slek"/>
<input type="text" value="No"/>	<input type="text"/>	<input type="text"/>

Command override ⓘ [ ] Example: [ "/bin/bash", "-c", "echo hello; sleep 100000" ]

Key management ⓘ ☒ Microsoft-managed keys (MMK) ☐ Customer-managed keys (CMK)

OS type : Linux

Copy to clipboard

IP address (Public) : 1.14

FQDN : ---

Container count : 1

5. Vytvorte kontajner pre klienta (slek-client). Nezabudnite, že slek-client musí byť zostavený s premennou API\_URL **smerujúcou na váš novonasadený backend** - krok 3 (pre vyššiu bezpečnosť môžete použiť HTTPS endpointy). Pomôcka: vonkajšiu adresu slek-server kontajnera nájdete po jeho rozkliknutí v sekcii **Ingress**. Pozor, po presadení kontajnera slek-server vám Azure môže vygenerovať novú ingress adresu.

#### Settings

Authentication

Secrets

Ingress

Continuous deployment

Custom domains

Dapr

Identity

Transport

Auto

Insecure connections

☒ Allowed

Target port \* ⓘ

5432

Endpoint(s)

<https://slek-postgres--a3noocl.internal.proudsand-northeurope.azurecontainerapps.io>  
-postgres--a3noocl.internal.proudsand-northeurope.azurecontainerapps.io

6. Aktualizujte premennú prostredia DB\_HOST (resp. PG\_HOST) pre váš slek-server

kontajner jeho rozkliknutím a zvolením -> Container -> Environment variables -> Edit and deploy. Použite IP adresu Postgres inštancie z kroku 4. V sekcii Scale zvolíte interval pre repliky 1-1.

Search

Secrets

Ingress

Continuous deployment

Custom domains

Dapr

Identity

Service Connector (preview)

Locks

Application

Revision management

Containers

Scale

Edit and deploy

Refresh

One or more containers, along with settings such as scale rules, can be specified in a container app. Edit the container app to change the configuration.

Container

slek-server

Properties

Environment variables

Health probes

Search

Name	Source	Value
APP_KEY	Manual entry	nInpGYSTleLKKrMtkZSPJfI8tHJWMla9
DB_HOST	Manual entry	localhost
HOST	Manual entry	0.0.0.0
PORT	Manual entry	3333

## 6. Rozkliknite nasadený kontajner slek-server, zvolíte sekciu Console - Bash terminál a zmigrujte databázu:

Select a container instance to connect to console.

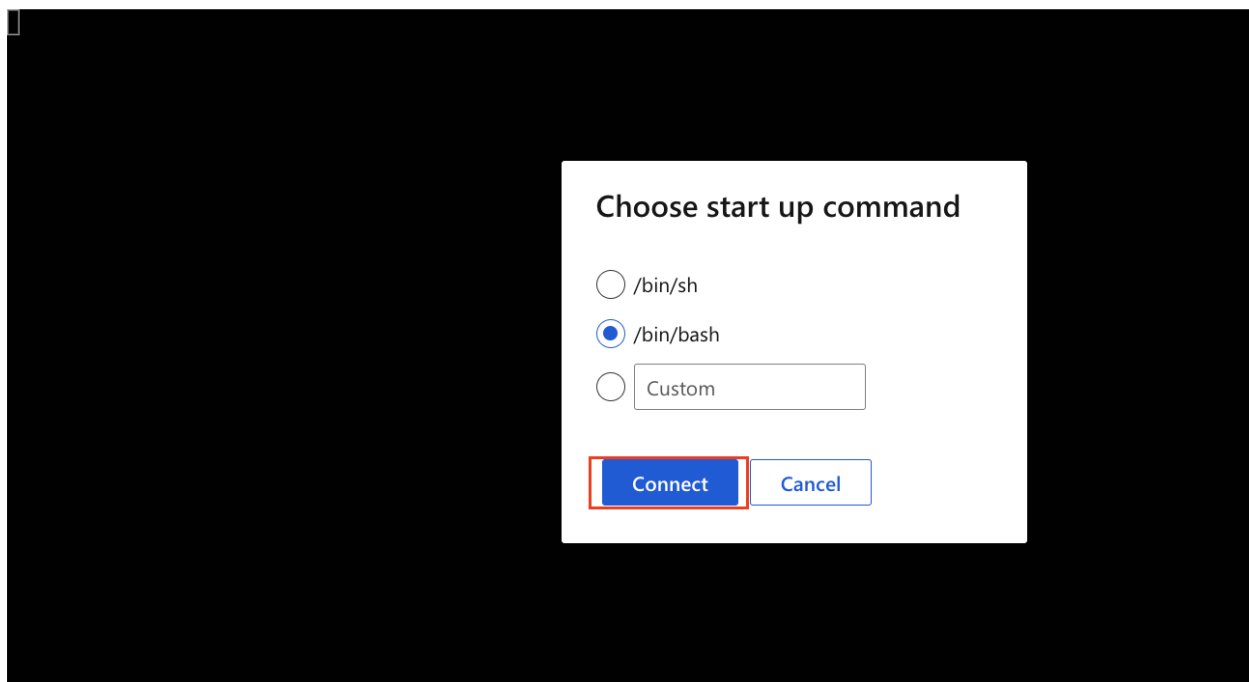
Replica

slek-server--ijqv

Container

slek-server

Reconnect



```
node ace migration:refresh --seed
```

## 7. Rozkliknite nasadený kontajner slek-client, skopírujte ingress adresu a preverte funkcionlitu vašej PWA aplikácie.

## Login

Email

---

Password



---

☐ Remember me

[CREATE ACCOUNT](#)

[LOGIN](#)

---

Revision #13

Created Sun, Dec 4, 2022 1:22 AM by Adam Puskas

Updated Sun, Dec 4, 2022 12:22 AM by Adam Puskas