

riganti

 Update Conference  
Prague 2020



# Cloud-Native in a Day

**Tomáš Herceg**

CEO @ RIGANTI

Co-founder of Update Conference

Microsoft MVP

[tomas.herceg@riganti.cz](mailto:tomas.herceg@riganti.cz)

[@hercegtomas](https://twitter.com/hercegtomas)

[www.tomasherceg.com/blog](http://www.tomasherceg.com/blog)



# Proč cloud-native?

- Pohodlnost pro nás vývojáře
  - „Jakou infrastrukturu budete potřebovat?“
  - „Pošlete nám ZIP balíček s aplikací, my si to nasadíme a otestujeme“
  - „Nemůžeme použít Mongo DB, naši admini tomu nerozumí“
- Bezpečnost
  - Cloudové platformy mají zkušené týmy
  - Díky množství aplikací jsou efektivnější
- Spolehlivost
  - Většina služeb má dostupnost >99,9%
  - Geografická redundance

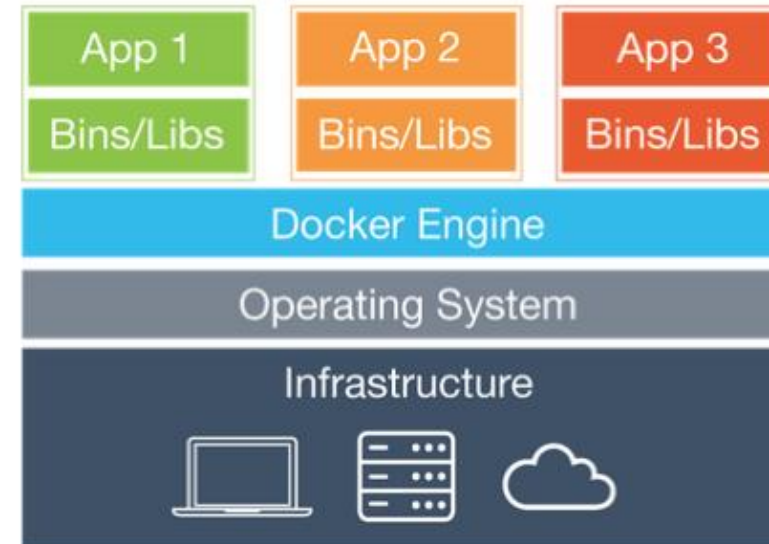
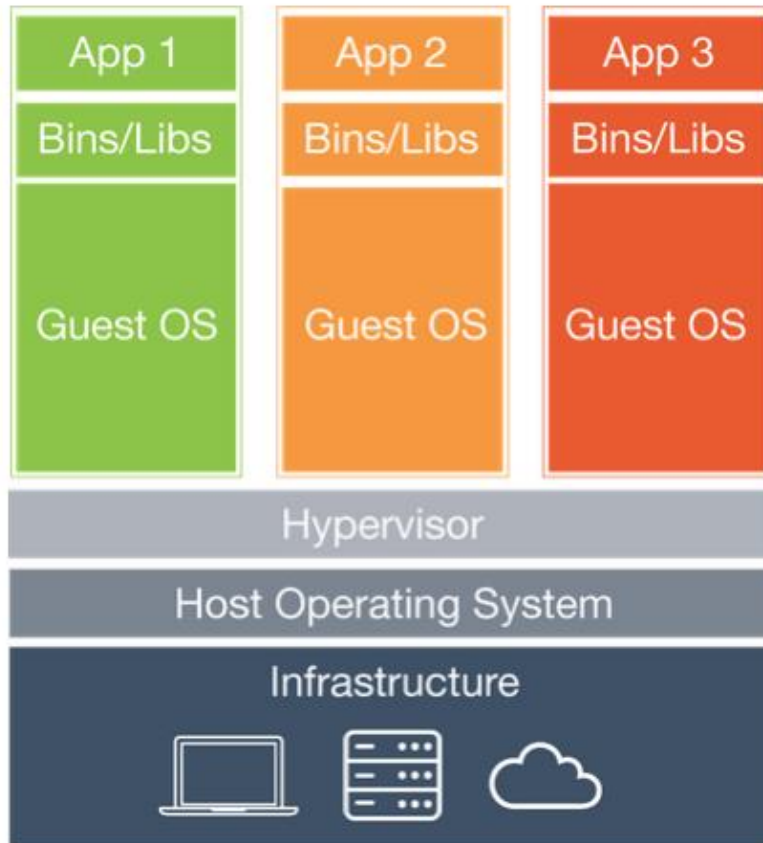


# Agenda

- **10:00 – 11:00**  
**Kontejnery v Azure**
- 11:15 – 12:15  
Azure Cosmos DB
- 13:00 – 15:00  
DEMO: Nasazení aplikace do Azure pomocí CI/CD



# Co je kontejner?





# Vlastnosti kontejnerů

- Nižší paměťové nároky
- Jednotný způsob deploymentu bez technologie
- Rychlý start
- Snapshot v libovolném okamžiku
- Stejné prostředí jako v produkci
  - „Works on my machine“ syndrom
- Copy on write a „vrstvy“ filesystému



# Kdy kontejnery použít

- Microservices architektura
  - Ne vždy chceme nasazovat vše
  - Různé výkonnostní nároky a škálování
- Zjednodušení nasazování
  - Vyšší spolehlivost, méně chyb



# Kdy kontejnery použít

- Instalace a distribuce aplikací
  - Jak složité bylo napsat instalátor k SQL Serveru?
- Mnoho dalších scénářů se objeví časem
- Konzistentní buildy
  - Stáhnete zdrojáky od kolegy a jde to zkompilevat na první dobrou



# Kontejnery v Azure

- Azure App Service (Linux)
  - Jeden nebo několik málo kontejnerů
  - Před kontejnerem je reálný webserver – pokročilá konfigurace
- Azure Container Instance
  - Samostatně běžící kontejner, nebo skupina kontejnerů
  - Různá omezení (síťování mezi kontejnery apod.)





# Kontejnery v Azure

- Azure Kubernetes Service
  - Vhodné pro velké aplikace postavené na kontejnerech
  - Standardní Kubernetes cluster hostovaný v Azure
    - Různé extensions pro Azure-specific functionality
  - Definujete si počet a parametry worker nodů
  - Azure řeší management clusteru, upgrady atd.

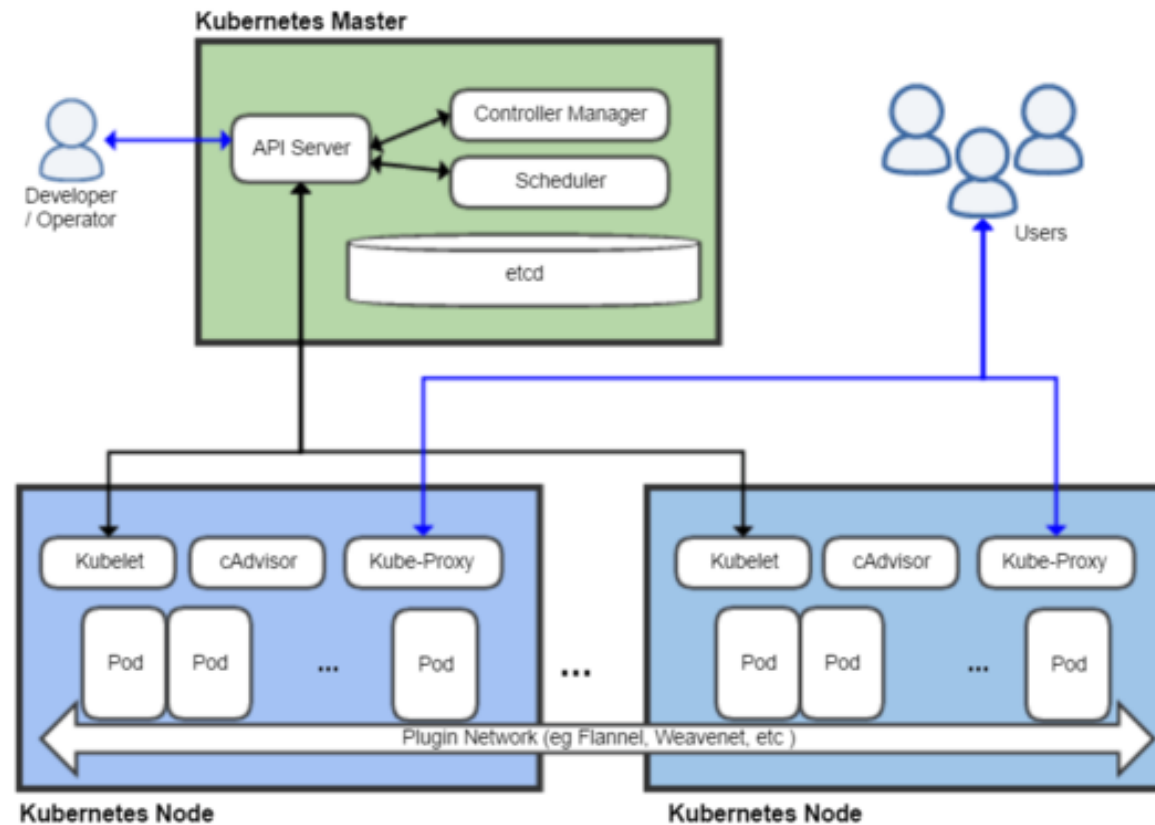


# DEMO

Kontejnerizace .NET aplikace



# Kubernetes





# Pod

- Samostatně nasaditelná jednotka
- Má svoji IP adresu
- Typicky obsahuje jeden kontejner
  - Může jich být víc, pokud patří k sobě
    - např. aplikace + proxy server
- Bezstavový, může být kdykoliv přesunut, zabit nebo nahrazen
  - Přijde o změny ve filesystému, které udělal
- Definuje labels
  - Dvojice klíč=hodnota



# Service

- „Proxy“ pro skupinu podů
- Selector
  - Sada labelů – definuje, které pody do služby patří
- Služba je dostupná pod svým názvem v DNS
  - Load balancuje traffic mezi svými pody



# Ingress

- Vstupní „proxy“
- Routuje HTTP požadavky na konkrétní služby
  - /api do služby A
  - /backend do služby B
  - cokoliv jiného do služby C
  - ...
- Ingress controller je typicky nginx
  - Pokročilé možnosti konfigurace



# Vytvoření clusteru

```
# instalace Azure CLI - vyžaduje Chocolatey
```

```
choco install azure-cli -y
```

```
# připojení k Azure
```

```
az login
```

```
az account set --subscription "subscription_name"
```

```
# založení clusteru
```

```
az aks create --resource-group resource_group --name cluster_name \  
  --node-count 1 --node-vm-size Standard_B2ms --node-osdisk-size 32 \  
  --generate-ssh-keys --location westeurope --kubernetes-version xxx
```



# Připojení ke clusteru

# otevře browser s portálem Kubernetes

az aks browse --resource-group *resource\_group* --name *cluster\_name*

# namapuje kubectl na Azure cluster

az aks get-credentials --resource-group *resource\_group* --name  
*cluster\_name*

# práce s kubectl

kubectl get pods

kubectl apply -f *file.yaml*





# Agenda

- 10:00 – 11:00  
Kontejnery v Azure
- **11:15 – 12:15**  
**Azure Cosmos DB**
- 13:00 – 15:00  
DEMO: Nasazení aplikace do Azure pomocí CI/CD



# Úvod

- Globálně distribuovaná, vícemodelová databáze
- Podpora různých API
- Garantovaná nízká latence
- SLA
- Indexy spravovány na pozadí
  - Lze definovat i manuálně
- Multi-master writes




# Podporovaná API


- SQL (Document DB)
  - Mongo DB
  - Graph (Gremlin)
  - Table (Azure Table Storage)
  - Cassandra
- 
- API vybíráte při vytváření CosmosDB accountu

Azure Cosmos DB  
New account


\* ID  
<enter a unique name>

\* API   
SQL

\* Subscription  
<subscription ID>

\* Resource Group   
☒ Create new ☐ Use existing  
<enter same unique name as used in ID above>

\* Location  
<choose the location closest to users>

☐ Enable geo-redundancy 

☒ Pin to dashboard

**Create** [Automation options](#)



# Základní poznatky

- Je to dokumentová databáze
  - Cosmos DB se vyvinula z Azure Document DB
- Efektivita dotazů má zásadní vliv na cenu
  - Rychlé (= levné) hledání podle primárního klíče
  - Rychlé (= levné) hledání v rámci partition
    - Jeden ze sloupců můžete nastavit jako partition key



# Propustnost

- Jednotka RU/s
  - Každý dotaz má jinou náročnost
    - Lze zjistit v portálu
  - Čtení
    - 1kB dokument = 1 RU
    - 100kB dokument = 10 RU
  - Zápis
    - 1kB dokument = 5 RU
    - 100kB dokument = 50 RU
- Lze provisionovat na databázi (sdílené RU/s)  
nebo na konkrétní kontejner



# Ceny

- Standard provisioned throughput
  - Kolik si navolíte, tolik budete mít
  - Minimum 400 RU/s
    - Cca 23\$ / měsíc
    - Dříve bylo nutné mít 400 RU/s pro každou kolekci, ale lze je již sdílet
- Storage
  - Cca 0.25\$ / GB
- Platí se za každý region (pokud chcete škálovat)
- V případě multi-master writes, platí se víc (cca 2x tolik)



# Ceny

- Autoscale provisioned throughput
  - Minimálně 4000 RU/s
  - Automaticky škáluje mezi 10 – 100% nastavených RU/s
    - Platíte jen za to, co využijete
  - Cena
    - 4000 RU/s – průměrné využití 10% – 35\$
    - 4000 RU/s – průměrné využití 100% – 350\$
- Free tier
  - Prvních 400 RU/s a 5GB místa je zdarma



# DEMO

Interakce s Cosmos DB z .NETu





# Kolik stojí dotazy

- Dotazování podle primárního klíče je levné
  - 2-3 RU (záleží na velikosti dokumentu, indexech atd.)
- Dotazování v rámci jedné partition
  - Relativně levné, obvykle jednotky RU
  - Záleží na velikosti partitions
- Dotazování napříč partitions
  - Většinou desítky RUs



# Denormalizace

- V relačních databázích se typicky dodržují normální formy
  - Motivace
    - Ušetření místa na disku
    - Nejsou třeba hromadné updaty
    - Je jednodušší zajistit transakčnost
  - Nevýhody
    - Složité (a pomalé) dotazy s mnoha joiny
    - Čtení je typicky mnohem častější než zápis



# Denormalizace

- Cosmos DB je stavěna na globální použití a multi-master writes
  - Tyto požadavky jdou proti sobě
- Datový model je jednodušší
  - JOINy se v podstatě nedělají
    - Dokumenty by měly obsahovat vše, co bude potřeba
  - Data jsou uložena v několika různých podobách
    - Klidně jen dočasně (TTL)
  - Eventuální konzistence



# Change Feed

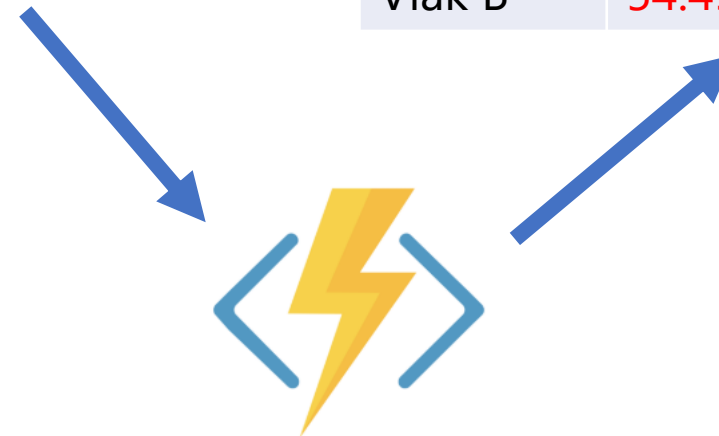
TrainPositions

TrainId	Date	X	Y
Vlak A	12:34:56.322	12.23456	23.45678
Vlak B	12:34:56.467	54.45878	44.66872
Vlak A	12:34:57.497	12.23454	23.45682
Vlak B	12:34:57.601	54.45879	44.66875

...

Trains

TrainId	X	Y
Vlak A	12.23456	23.45678
Vlak B	54.45878	44.66872





# Change Feed

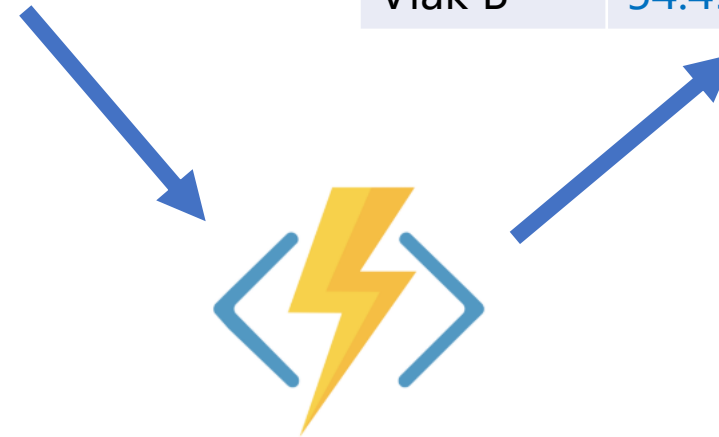
TrainPositions

TrainId	Date	X	Y
Vlak A	12:34:56.322	12.23456	23.45678
Vlak B	12:34:56.467	54.45878	44.66872
Vlak A	12:34:57.497	12.23454	23.45682
Vlak B	12:34:57.601	54.45879	44.66875

...

Trains

TrainId	X	Y
Vlak A	12.23454	23.45682
Vlak B	54.45879	44.66875





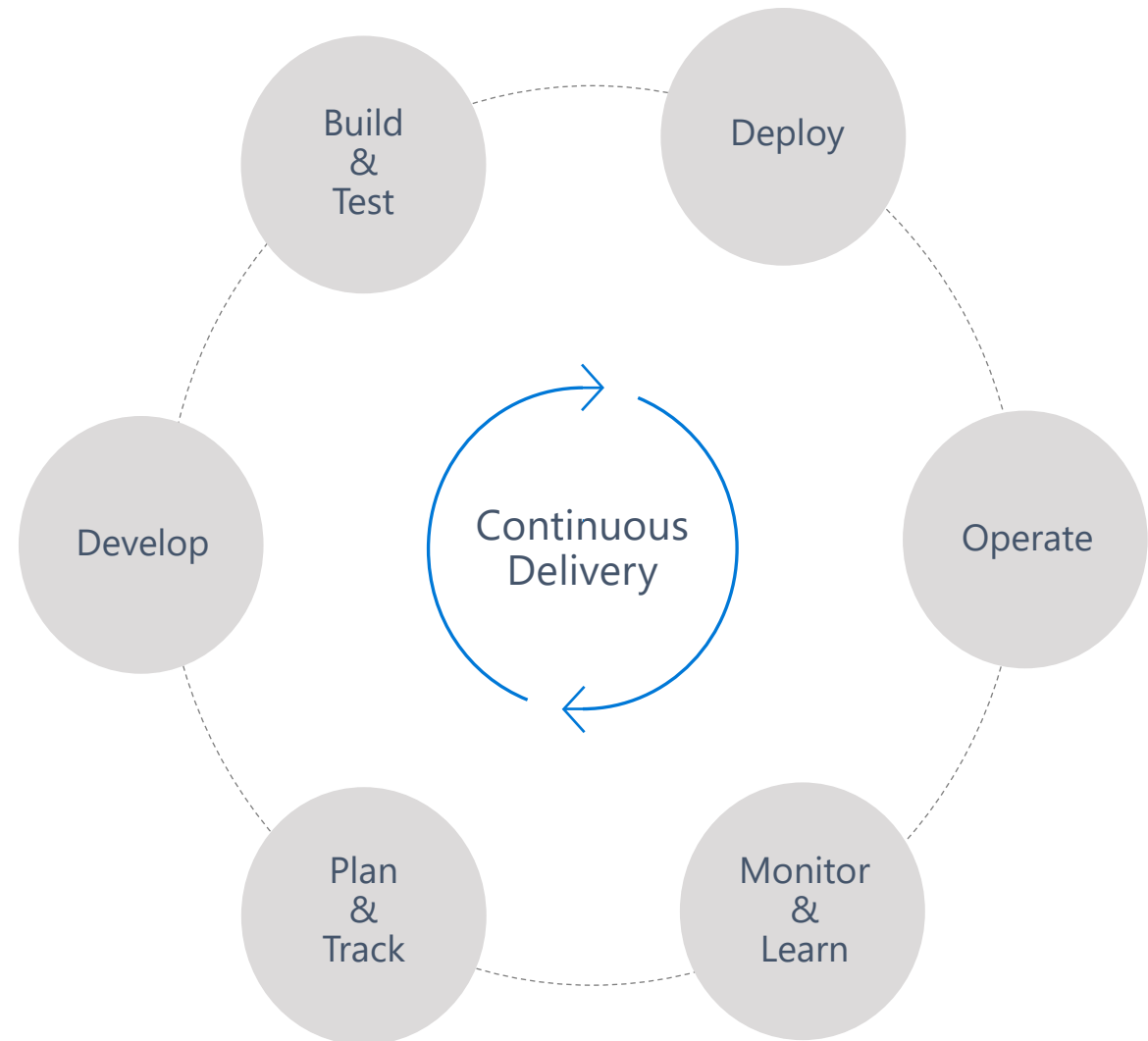
# Agenda

- 10:00 – 11:00  
Kontejnery v Azure
- 11:15 – 12:15  
Azure Cosmos DB
- **13:00 – 15:00**  
**DEMO: Nasazení aplikace do Azure pomocí CI/CD**



# Co je DevOps?

- Lidé
- Procesy
- Nástroje
- „Continuous Delivery“
- Zkrácení času k uvedení nové funkcionality





# K čemu DevOps?

- Lepší feedback od uživatelů a z produkčního prostředí
- Boří se hranice mezi vývojáři a IT správci
- Snížení chybovosti release
- "Dokumentace release procesu"
- Víte, co se s aplikací děje (monitoring)





# Součásti Azure DevOps



Azure Boards



Azure Pipelines



Azure Repos



Azure Test Plans



Azure Artifacts



# Azure Pipelines

- Kontinuální integrace a nasazení
- Podpora vlastních i cloudových build agentů
- Build navázaný na každý commit
- Release management s podporou více prostředí



# Azure Boards

- Kanban boardy, backlog, týmové dashboard
- Podpora různých metodologií
  - Agile
  - Scrum
  - ...
- Možnosti customizace



# Azure Repos

- Správa verzí zdrojového kódu
  - Git
  - TFVC (Team Foundation Server)
- Pokročilé nástroje pro pull requesty a integrace

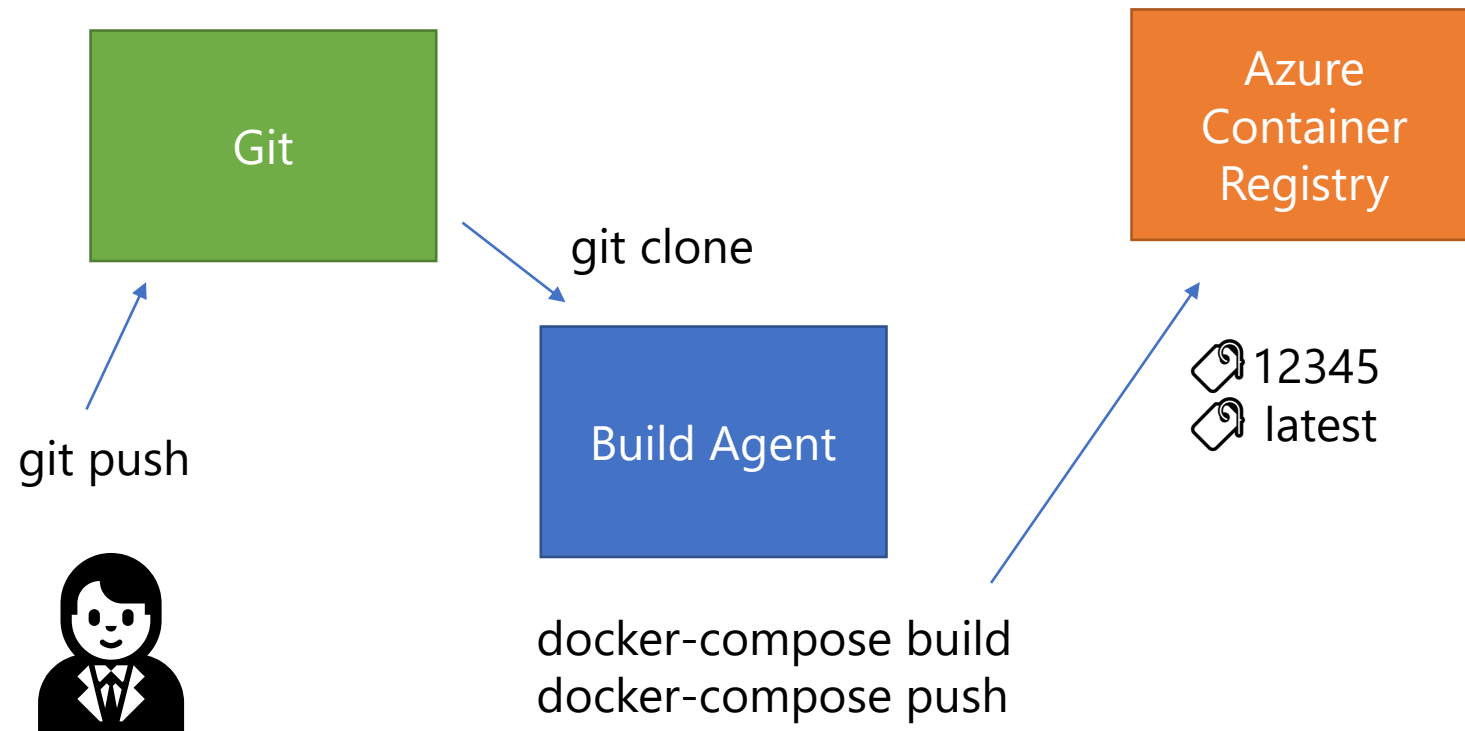


# Azure Artifacts

- Správa interních balíčků pro vývojáře
  - NuGet
  - Npm
  - Maven
- Snadné sdílení kódu napříč projekty

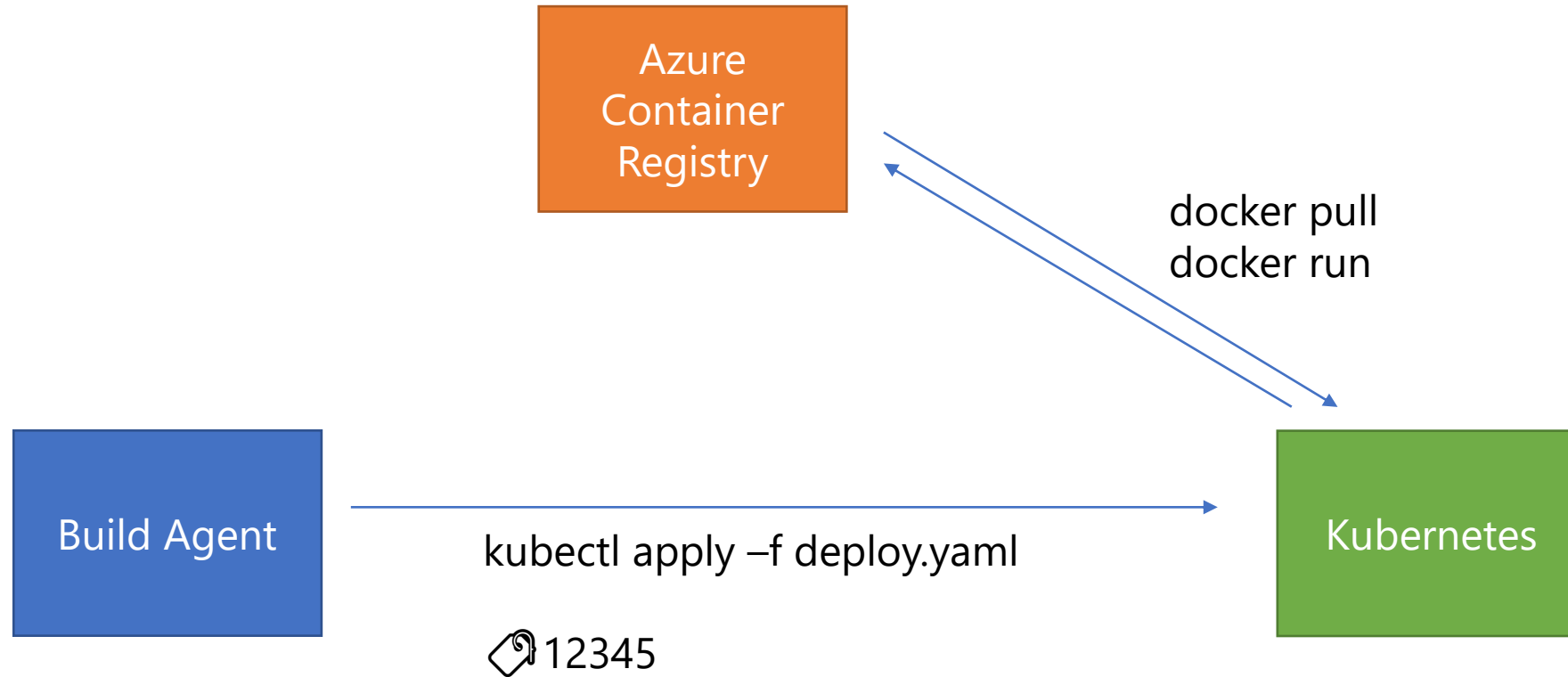


# Build





# Release





# DEMO

Nasazení aplikace do Kubernetes pomocí Azure DevOps





# Co byste si měli odnést

- Kontejnery umožňují jednotný typ deploymentu aplikace
  - Bez ohledu na programovací jazyk, runtime nebo knihovny
- V microservices aplikacích se SQL databáze ne vždy hodí
  - Dokumentové databáze vyžadují jiný způsob přemýšlení a návrhu
- Kubernetes je pokročilá platforma pro kontejnerová řešení
  - Složité, ale tooling kolem toho se neustále vyvíjí



Nebojte se toho!

riganti

 Update Conference  
Prague 2020



# Q&A

**Tomáš Herceg**

CEO @ RIGANTI

Co-founder of Update Conference

Microsoft MVP

[tomas.herceg@riganti.cz](mailto:tomas.herceg@riganti.cz)

[@hercegtomas](https://twitter.com/hercegtomas)