



# O kontejnerech, cloudu, vesmíru a vůbec se zvláštním zřetelem na Kubernetes

Tomáš Kubica, Azure

Blog: <https://tomaskubica.cz>

Email: [tomas.kubica@microsoft.com](mailto:tomas.kubica@microsoft.com)

LinkedIn: <https://www.linkedin.com/in/tkubica/>



Microsoft  Linux



# Co je Azure? Soustava moderních datových center



# 38

Azure regionů

## Nedávno spuštěné:

US Regions: West US 2 and West Central  
Germany – Launched in Sep 2016  
United Kingdom – Launched in Sep 2016

## Ohlášené:

France: France Central and France South  
Korea: Korea Central and Korea South  
DoD East and Central

## Platform Services

### Security & Management

- Portal
- Active Directory
- Multi-Factor Authentication
- Automation
- Key Vault
- Store / Marketplace
- VM Image Gallery & VM Depot

### Compute

- Cloud Services
- Service Fabric
- Batch
- Remote App

### Web and Mobile

- Web Apps
- API Apps
- API Management
- Mobile Apps
- Logic Apps
- Notification Hubs

### Developer Services

- Visual Studio
- Azure SDK
- Team Project
- Application Insights

### Hybrid Operations

- Azure AD Connect Health
- AD Privileged Identity Management
- Backup
- Operational Insights
- Import/Export
- Site Recovery
- StorSimple

### Integration

- Storage Queues
- Biztalk Services
- Hybrid Connections
- Service Bus

### Analytics & IoT

- HDInsight
- Machine Learning
- Data Factory
- Event Hubs
- Stream Analytics
- Mobile Engagement

### Data

- SQL Database
- SQL Data Warehouse
- Redis Cache
- Search
- DocumentDB
- Tables

### Media & CDN

- Media Services
- Content Delivery Network (CDN)

## Infrastructure Services

### Compute

- Virtual Machines
- Containers

### Storage

- BLOB Storage
- Azure Files
- Premium Storage

### Networking

- Virtual Network
- Load Balancer
- DNS
- Express Route
- Traffic Manager
- VPN Gateway
- Application Gateway

Datacenter Infrastructure (24 Regions, 19 Online)



Proč vůbec potřebujeme měnit  
způsob jakým nasazujeme a  
provozujeme aplikace?

# Nekonzistentní prostředí vedou k chybám



Nainstaluji  
Otestuji



TEST



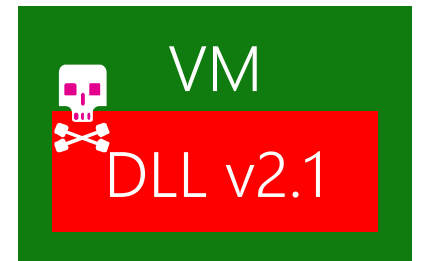
Nasadím



PROD



Vynadám



CRASH

# Nekonzistentní prostředí vedou k chybám



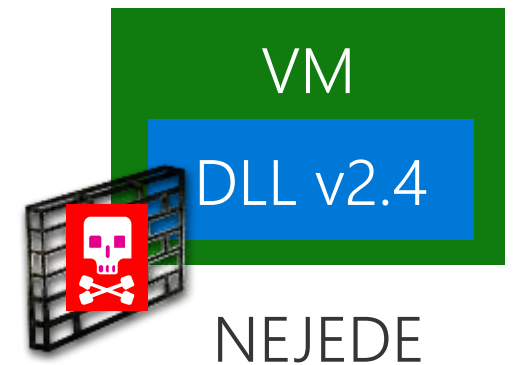
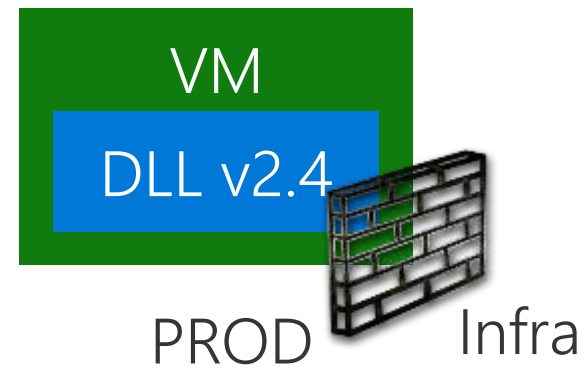
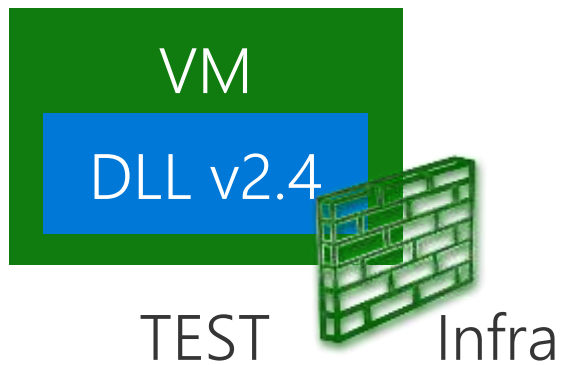
Nainstaluji  
Otestuji



Nasadím

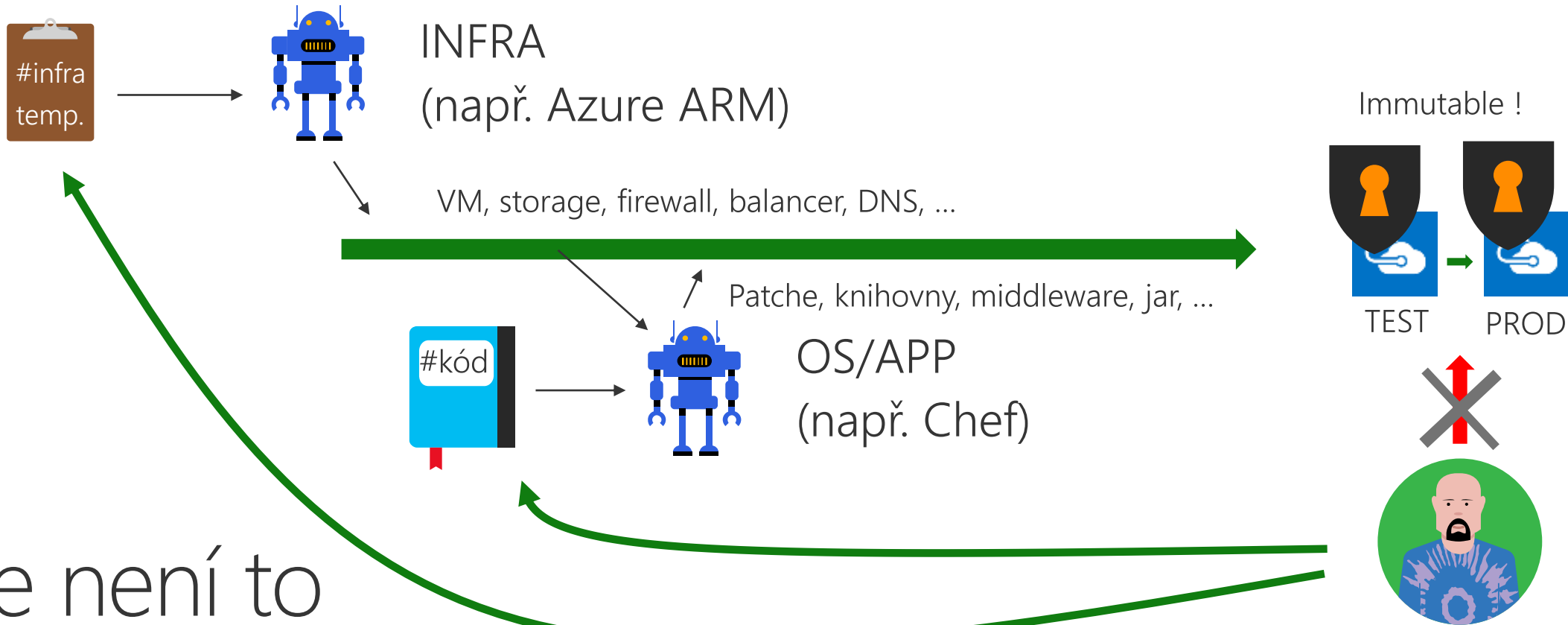


Vynadám



# Nekonzistentní prostředí vedou k chybám

Co nasadit immutable řešení (Phoenix „servery“)?

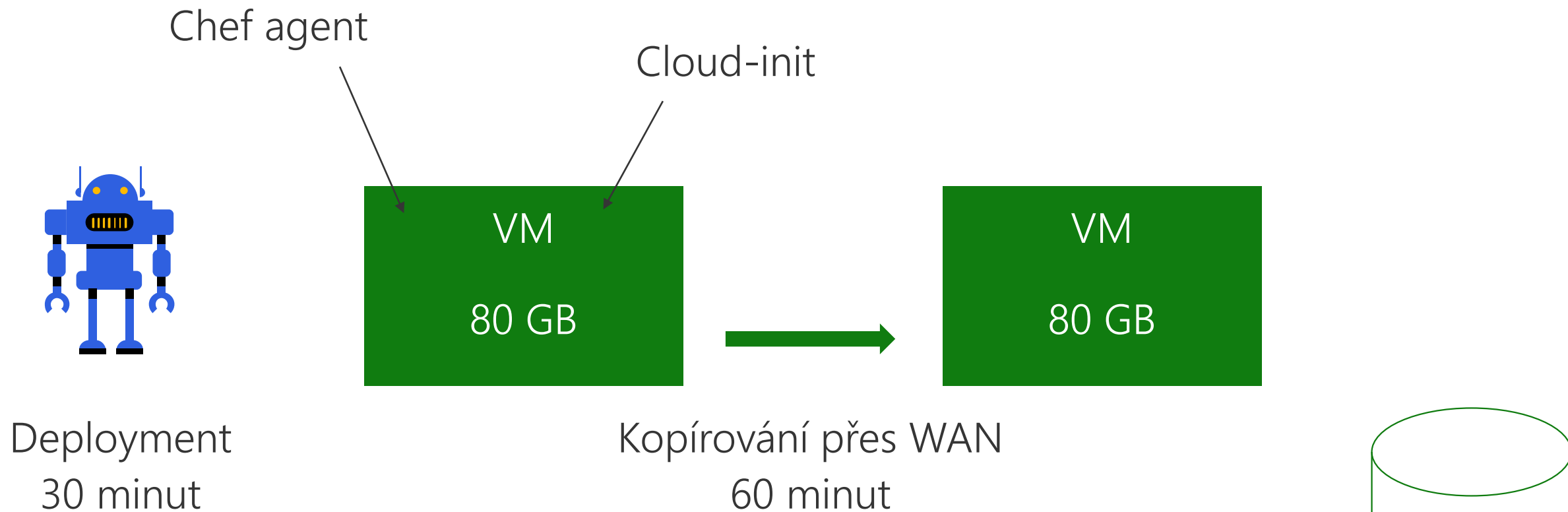


Potřebuji změnu, release, ...

Ale není to  
těžkopádné?



# Automatizace VM a OS je těžkopádná



A co teprve sítě, firewally nebo storage?

# Je hodně možností pro jednotku deploymentu, ale každá má mouchy

VM  
Image  
(OVA)

Middleware  
(WebLogic,  
WebDeploy)

Habitat (by Chef)

Package  
(RPM, MSI)

Specifické pro  
jazyk (např. Ruby)

CM (Chef,  
Puppet, Ansible)

# DevOps bez Dev a DevOps bez Ops



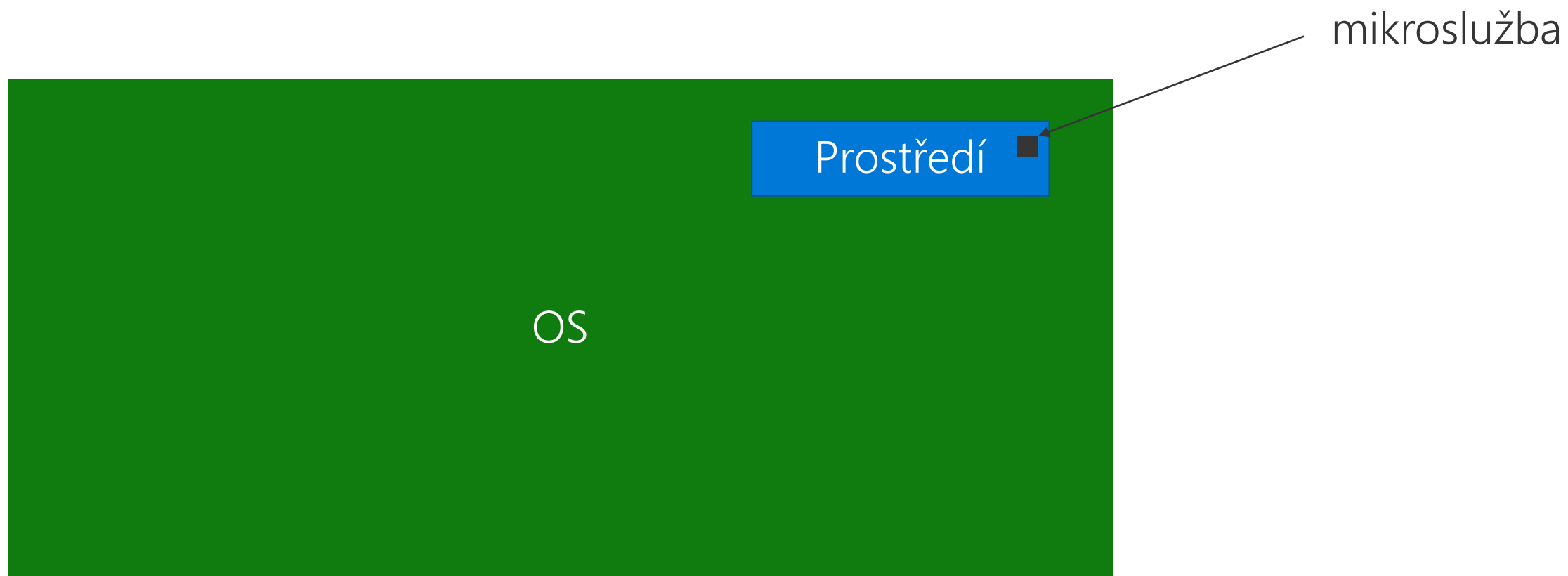
# DevOps



*„Jdeme do DevOps, ale provozu to říkat nebudeme, nepochopí to.“*

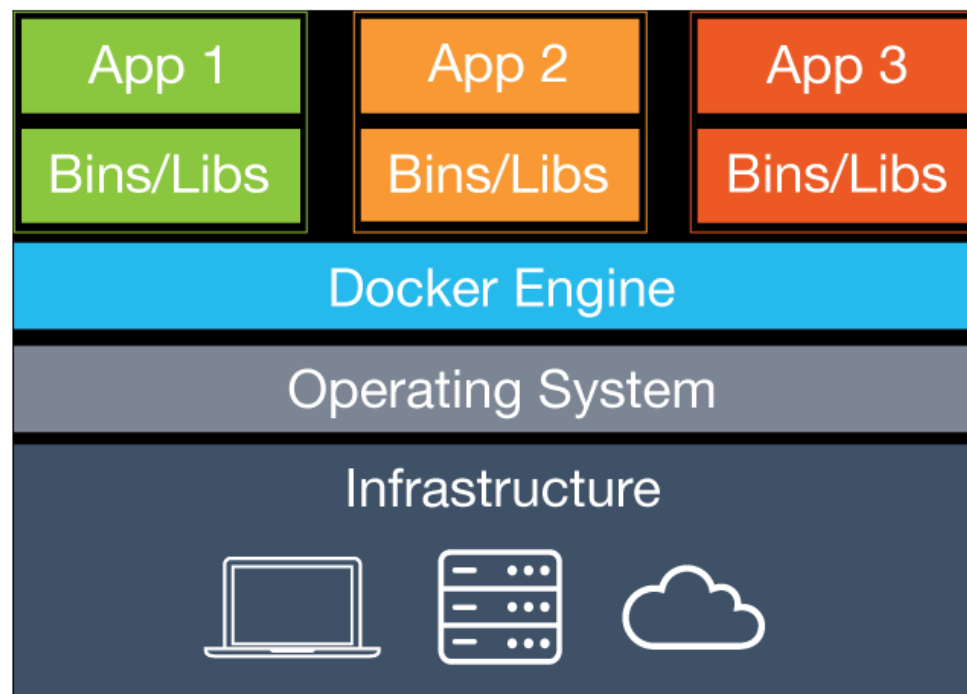
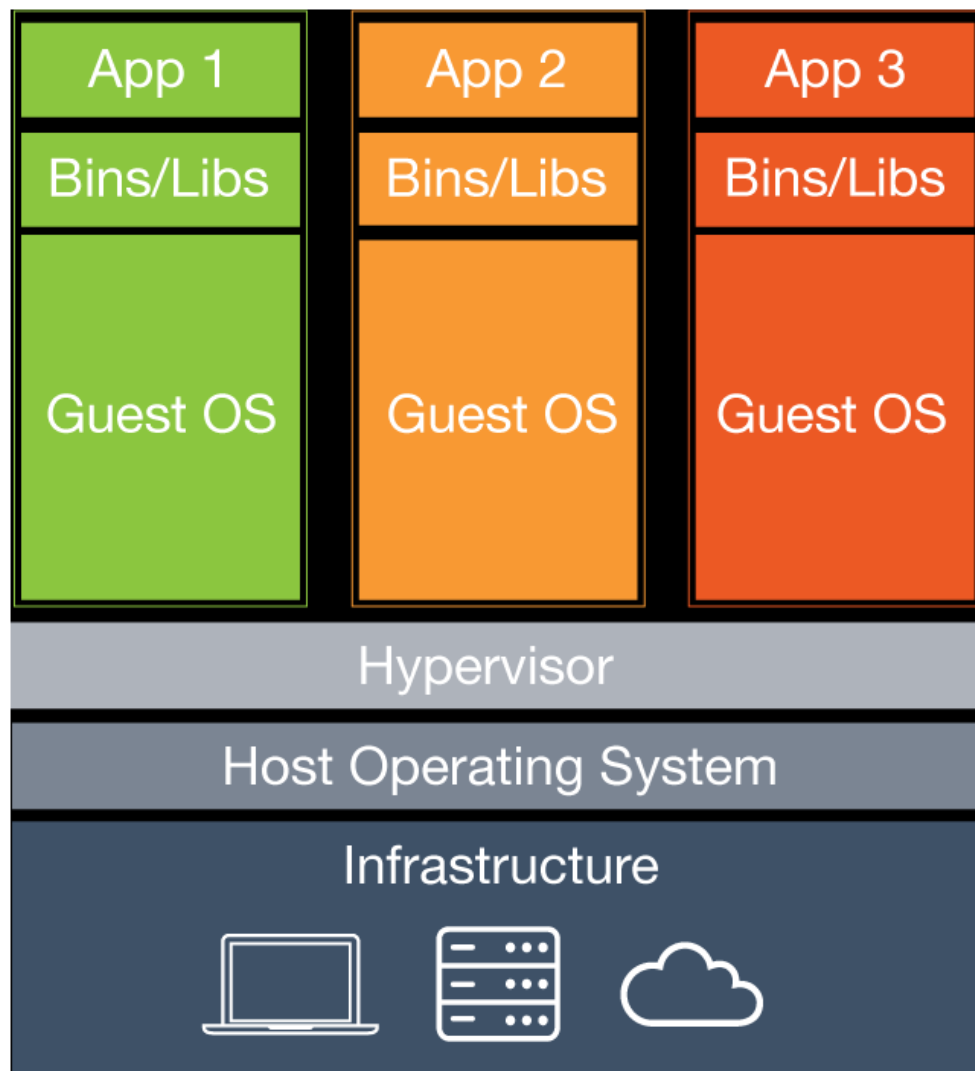
*„Chceme rozjet ten DevOps, vývojáře k tomu nepotřebujeme.“*

# Mikroslužby potřebují menší „servery“

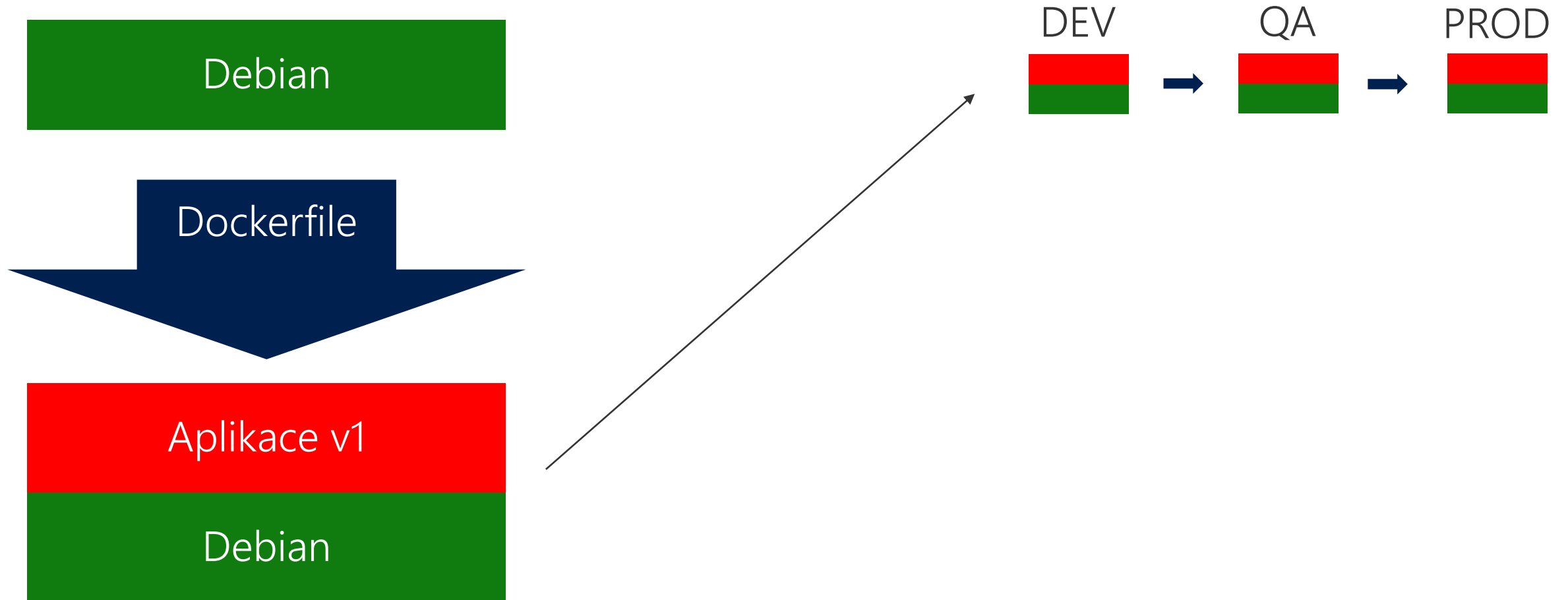


Proč jsou kontejnery tak populární odpověď?

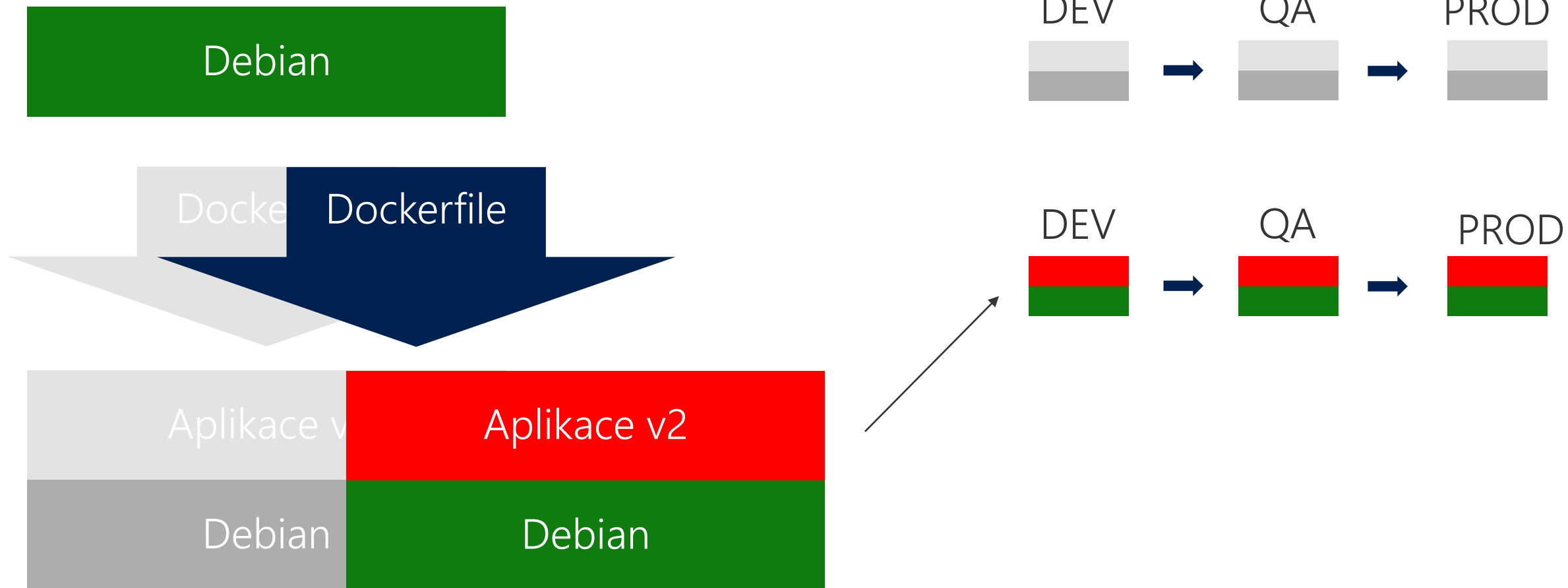
# Kontejner je malý, šikovný a efektivní



Kontejner je pomíjivý, je immutable,  
neoprašuje se



Kontejner je pomíjivý, je immutable,  
neopraňuje se

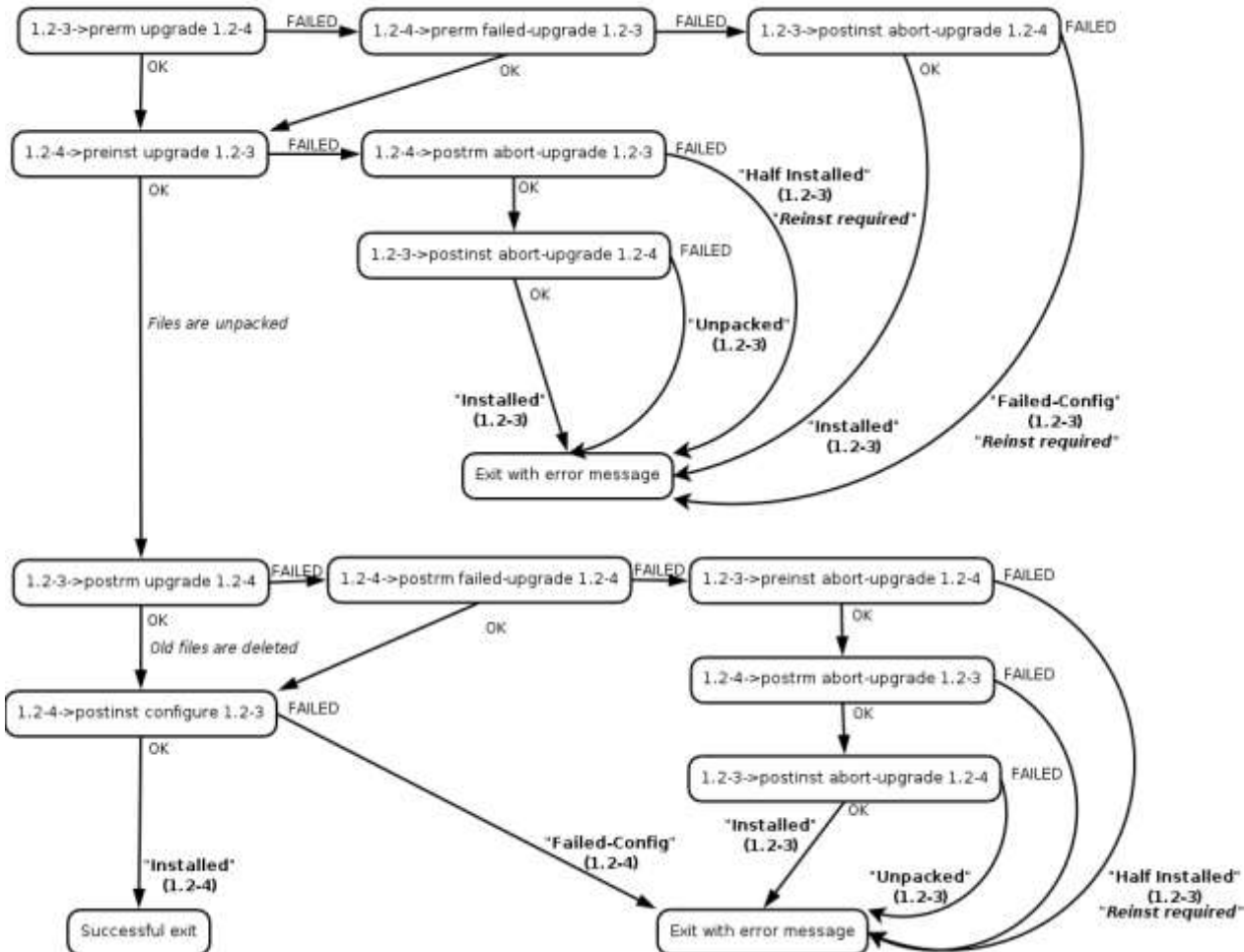




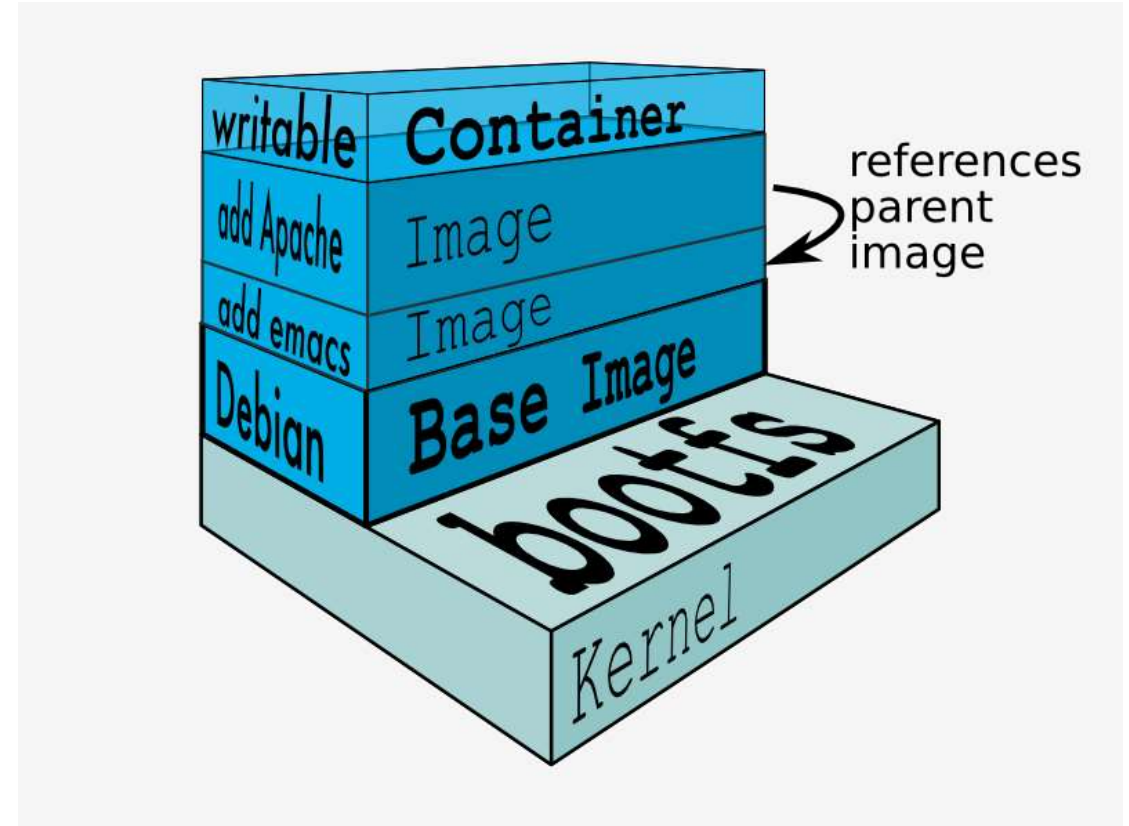
# Kontejner je jednotkou deploymentu

## dpkg

Upgrade of foo 1.2-3 (Installed) to 1.2-4



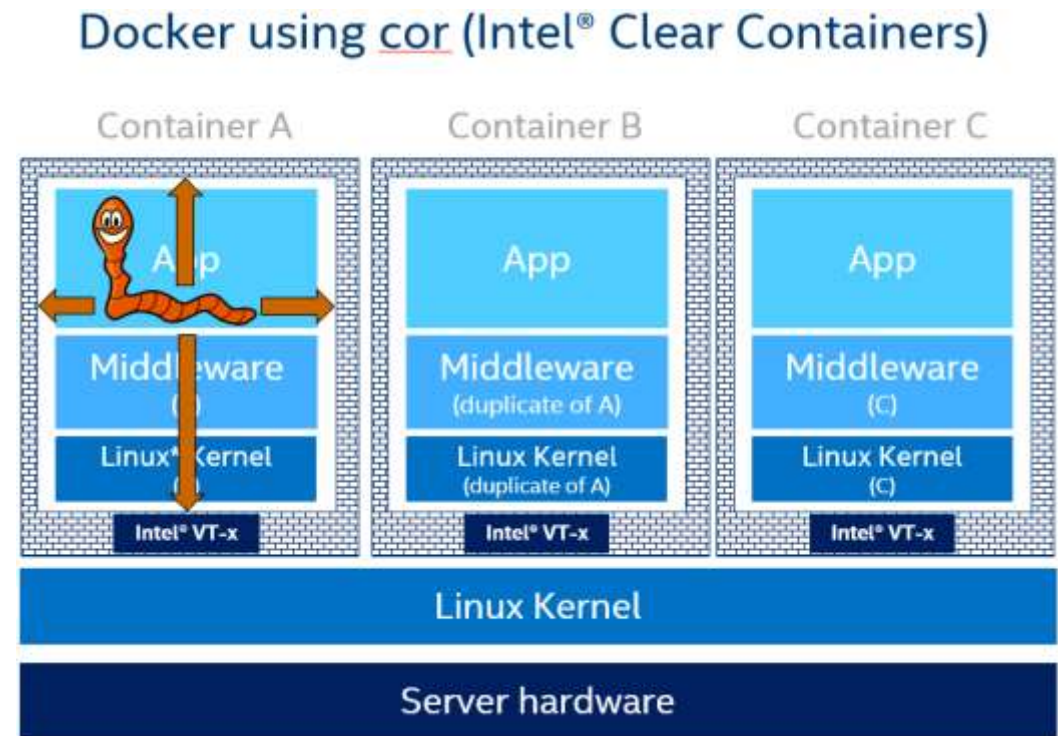
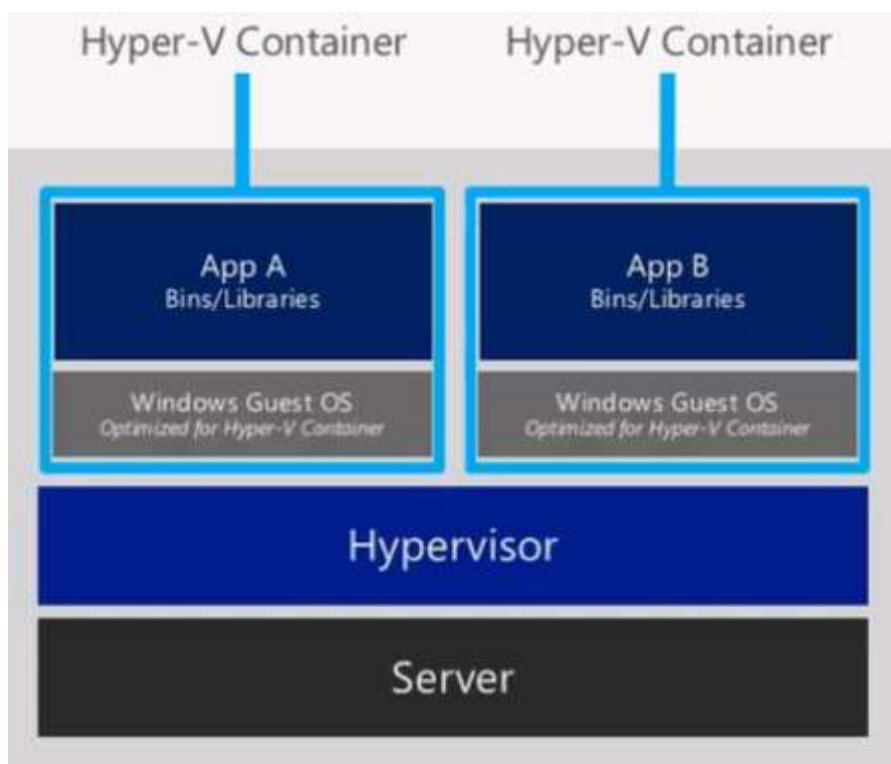
## kontejner



Dependency peklo  
nevzniká při deploymentu.

V ten moment už je vyřešené.

# Docker API je skvělé, tak co použít jeho efektivní ovládání se skutečnou izolací VT-x?



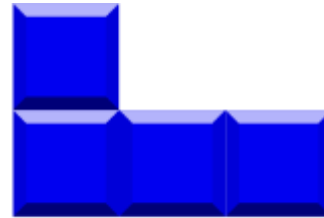
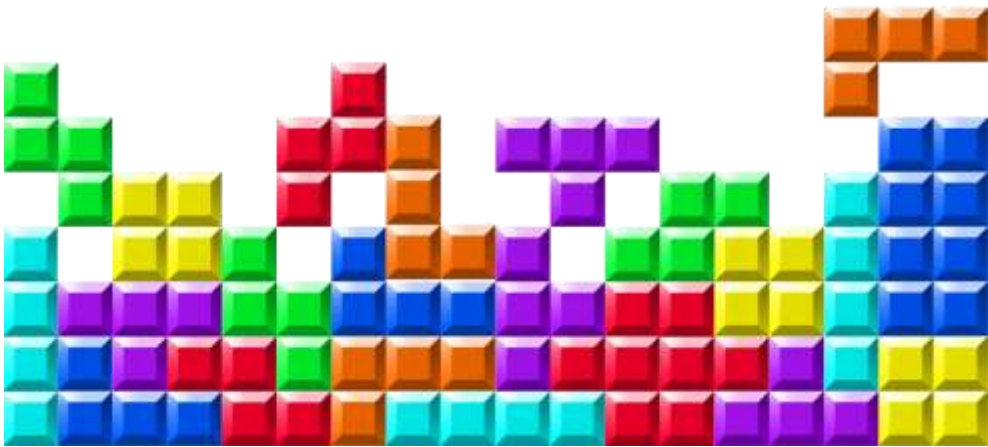
 Windows Server 2016



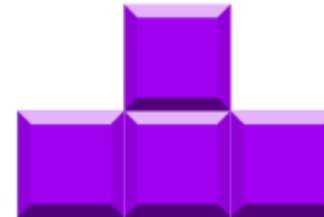
Proč bez orchestrátoru je  
kontejner spíš hračka?



# Kdo bude hrát Tetris?



Já potřebuji 128 MB RAM a rychlé SSD místo

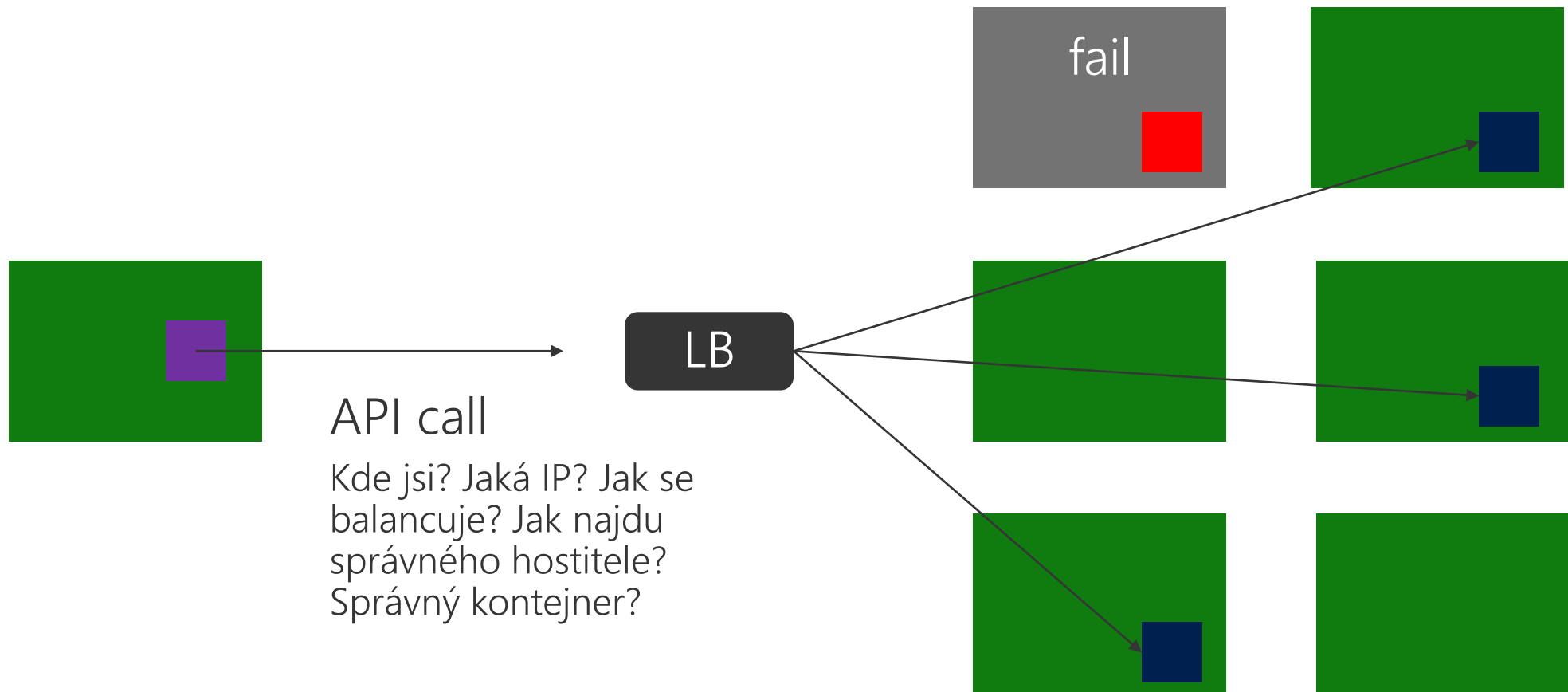


Já chci 512 MB RAM a rychlé síťové připojení



Já dělám single-threaded výpočty, potřebuji vysokofrekvenční CPU

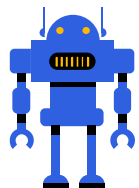
# Potřebuji balancovat provoz a služby se musí mezi sebou najít



# Jak budu nasazovat nové verze aplikace?

Už zase immutable?

Docker image

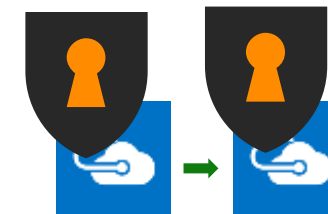


Orchestrátor  
(např. Kubernetes)

Kontejner, balancing, monitoring

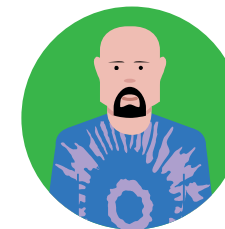


Immutable !



TEST

PROD



Ale kdo dá do image kód?

Potřebuji změnu, release, ...

Co orchestrátor nemá, a přesto  
to potřebuji aneb jak vypadá  
moderní PaaS?



# Svět není stateless, tak společnost nefunguje

## Distribuované systémy nejsou brnkačka

Azure Container Service  
Swarm, DC/OS, Kubernetes

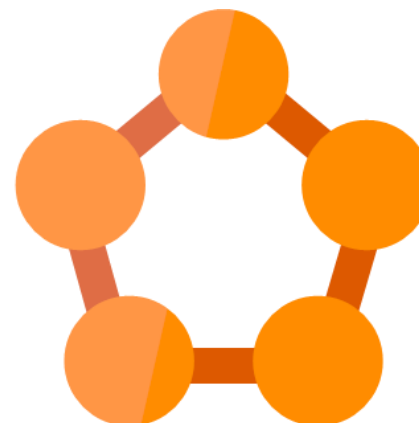


Externalizace state

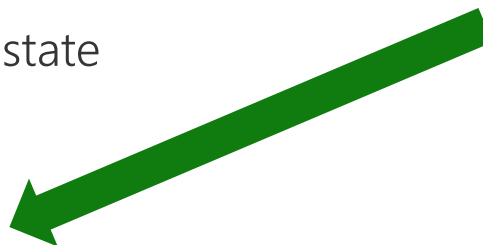
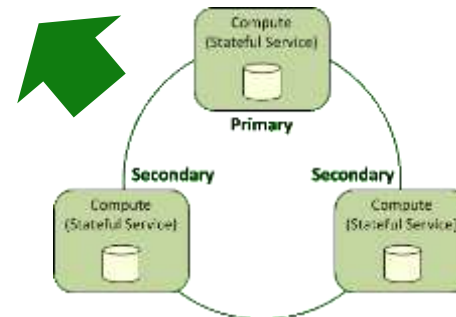


Azure PaaS datové služby

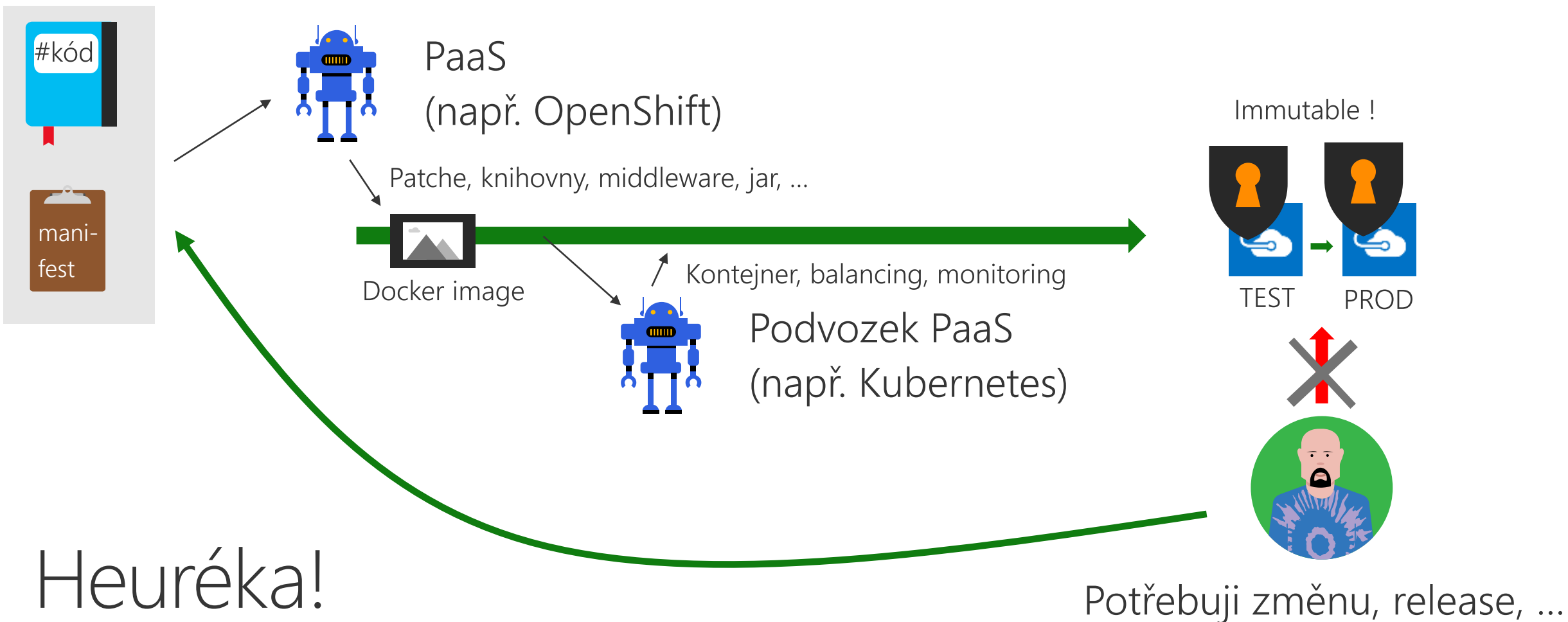
Azure Service Fabric



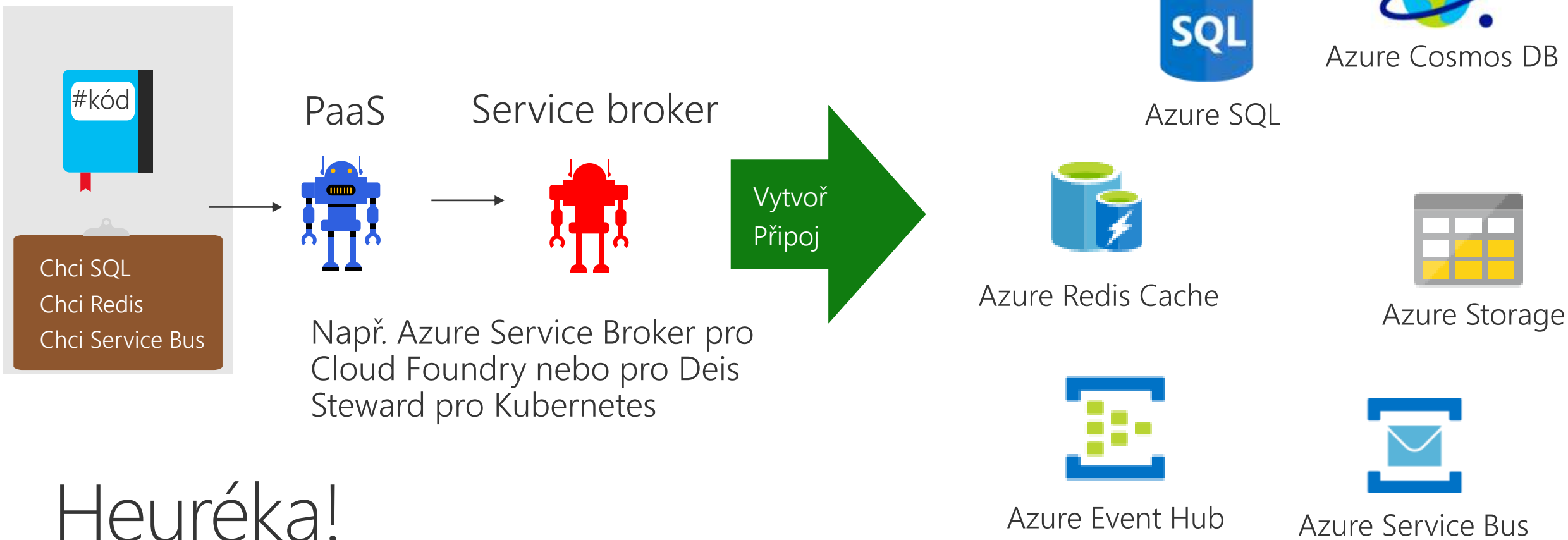
Stateful service  
Stateful actor



# Kontejner je jako binárka, ale já chci automatický kompilátor a packager



# Existuje svět tam venku (API, DBaaS, BigData as a Service, ...)

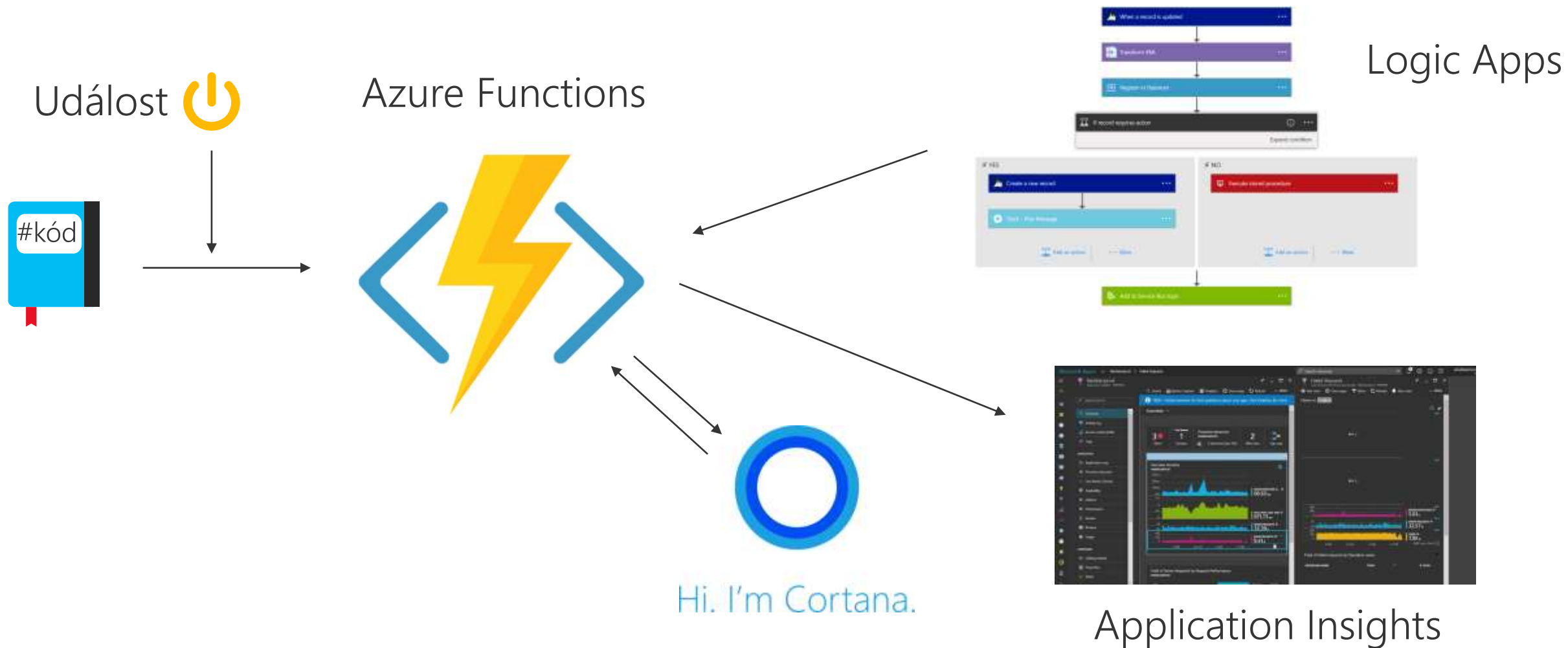


## Heuréka!

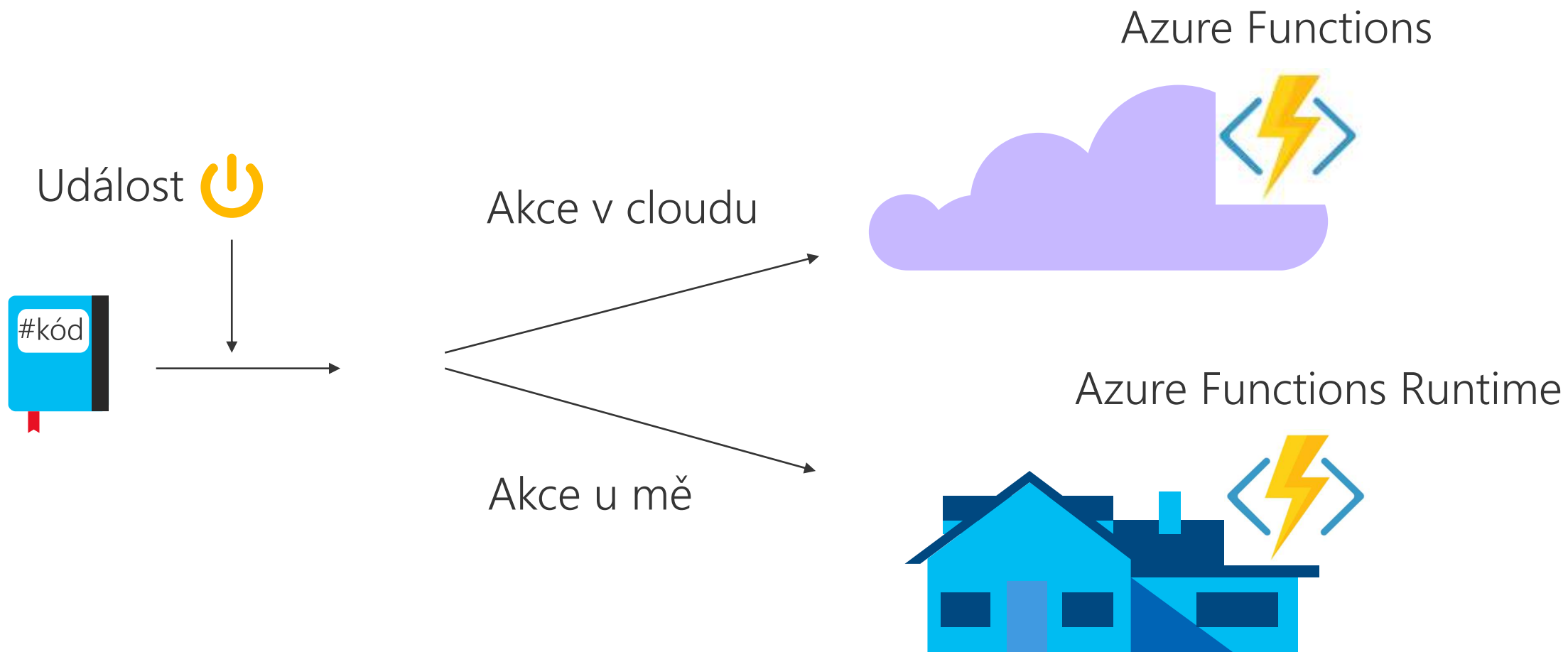
Směřuje zmenšování jednotky  
nasazení přes kontejner až k  
funkcím jako služba  
(serverless)?

# Ultimátní nanoslužby

## Událostmi řízený asynchronní model



# Ultimátní nanoslužby U mě nebo v cloudu?



Rozhodnuto, chci orchestrátor.  
Jaký si mám vybrat?

# Docker Swarm, výborný pro ty, co už Docker znají a chtějí používat stejné ovládání



- Chová se prakticky stejně jako jeden Docker node
- Stejně nástroje – Docker Compose, Docker Machine, ...
- Pluginy pro perzistentní volume, síťářinu

Podpora v





# DC/OS pro ty, co chtějí pracovat s kontejnery a daty současně (Hadoop, Spark, ...)



**DC/OS**

- Dvouúrovňový scheduler postavený na Mesos (kontejnerové orchestrátory, například Marathon, jsou jen podmnožinou možností)
- Podpora datově specializovaných orchestrátorů (Hadoop, Spark, ...)
- Je starší než Docker, prověřený

Podpora v



# Kubernetes tam, kde chcete to nejvíc rostoucí a nevádí vám se příští rok učit znova



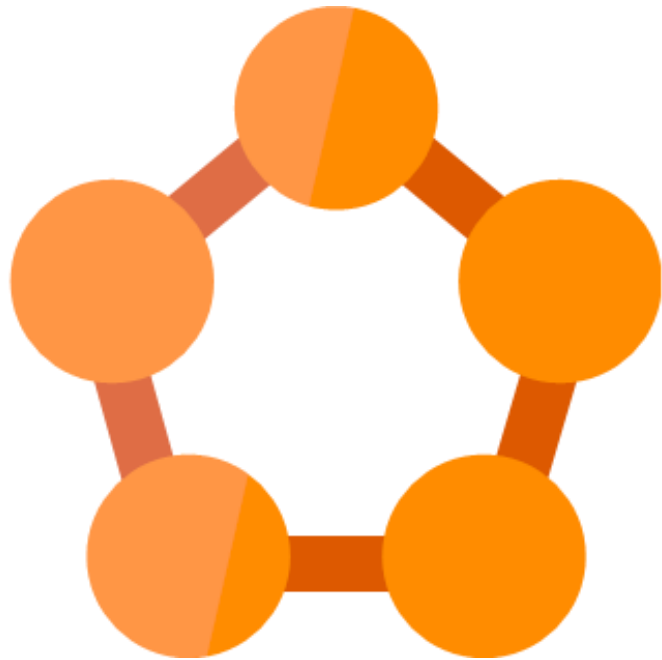
**kubernetes**

- Konceptuálně jiné (pody apod.)
- Průkopník IP-per-pod/kontejner (heuréka!)
- Nejvíc funkcí pro kontejnerový svět
- Výborná integrace do infrastruktury pod ním (například do Azure)
- V létě 2017 mu budou teprve 2 roky, mění se před očima
- Už dnes dobře podporuje Windows

Podpora v



# Azure Service Fabric pro .NET mikroslužby



Azure Service Fabric

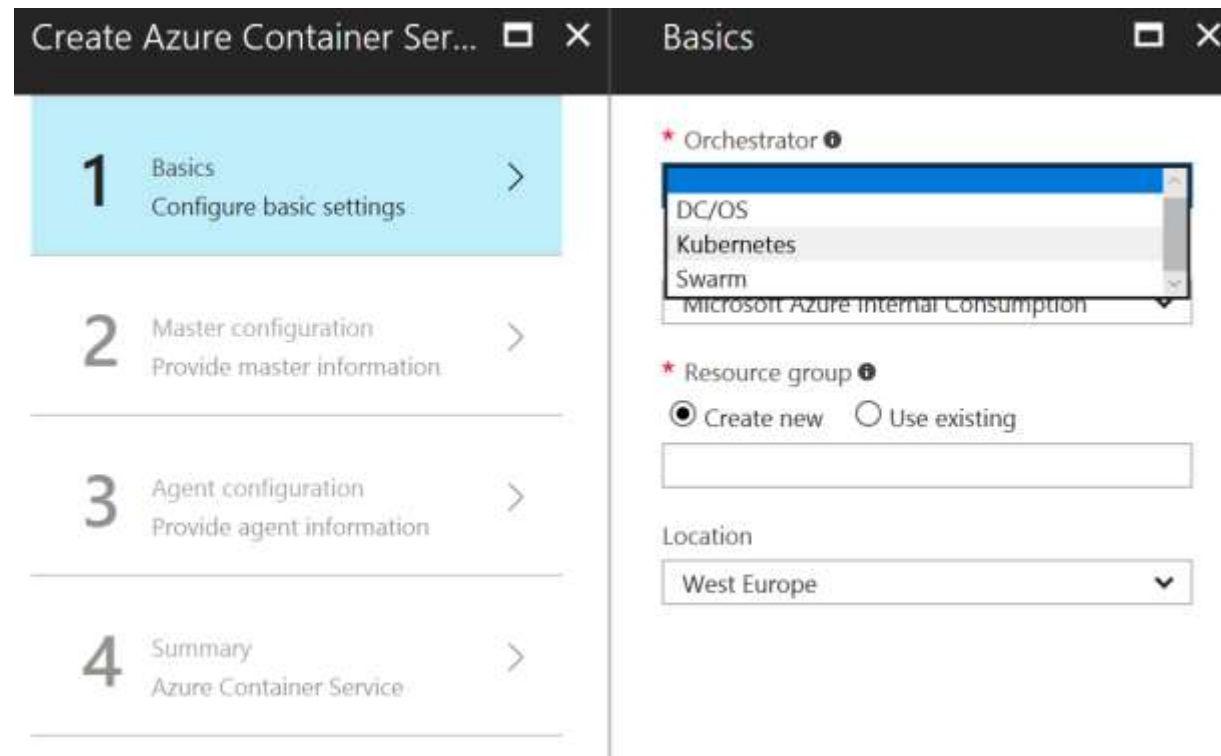
- Zaměřeno na pohodlí programátorů
- Stateful Services
- Reliable actor model
- Podpora kontejnerů
- Linux i Windows

Proč má smysl provozovat  
kontejnery v Azure?

# Vytvořte cluster podle best practice a bezpečnosti na kliknutí, platíte po minutách



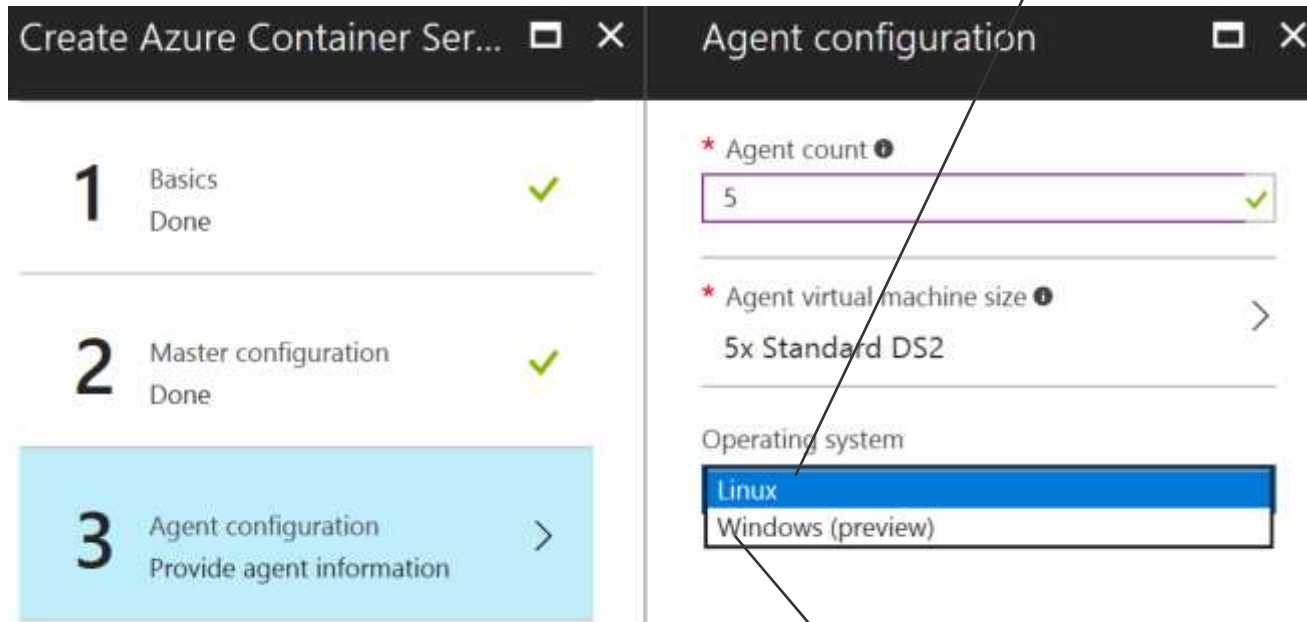
Get started with containers



<https://github.com/Azure/acs-engine>

# Linux nebo Windows kontejnery?

```
$ kubectl run mojenginx --image=nginx
```



```
$ kubectl run mojeiis --image=microsoft/iis
```

# Enterprise registr pro vaše Docker image



Azure Container Registry

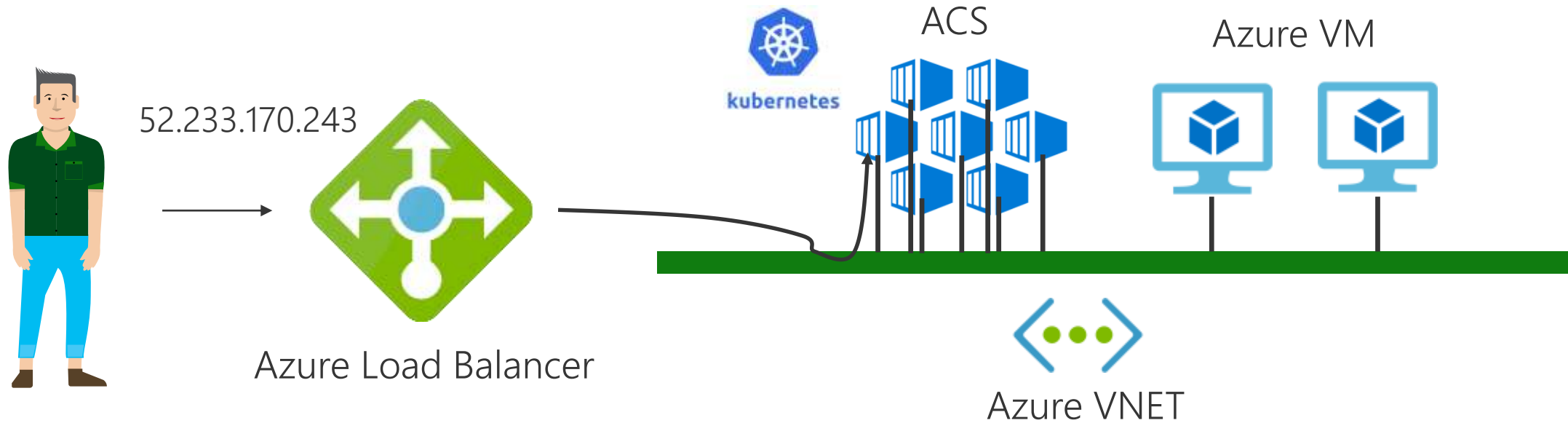


Azure Active Directory



On-premise AD

# Integrujte Kubernetes s infrastrukтурой aneb nezapomeňme na uživatele



```
$ kubectl get services
```

NAME	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	10.0.0.1	<none>	443/TCP	3h
mojeiis	10.0.196.35	52.233.170.243	80:32704/TCP	3m



# Oddělení na úrovni kontejnerů je bezpečákům málo, izolujte a škálujte vícero clusterů



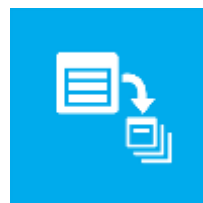
# Kontejnery mají v Azure širší použití, než jen s orchestrátory



Web App PaaS  
On Linux



```
docker pull microsoft/mssql-server-linux
```



Azure Batch  
(HPC)



docker  
Docker EE



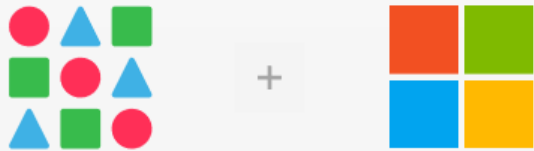
OPENSIFT<sup>™</sup>  
ENTERPRISE  
by Red Hat<sup>™</sup>



Pivotal **Cloud Foundry**<sup>™</sup>

Co přidat ke Kubernetes a získat tak moderní platformu jako služba pro vaše aplikace?

**Deis is joining Microsoft** - to  
continue our mission to make  
container technology easier to use.



<https://github.com/kubernetes/helm>

<https://github.com/kubernetes-incubator/service-catalog>

<https://github.com/deis/workflow>

# Package manager pro Kubernetes



<https://helm.sh/>

```
$ helm install --name my-release \  
  --set drupalUsername=admin,drupalPassword=password \  
  stable/drupal
```

<https://kubernetes.com/>

# Service broker pro Kubernetes



Vytvoř instanci  
(např. DBaaS)



Připoj k aplikaci  
(a předej connection string)

```
kubectl create -f instance.yaml
```

```
apiVersion: servicecatalog.k8s.io/v1alpha1
kind: Instance
metadata:
  name: mojeDB
spec:
  serviceClassName: smallDB
```

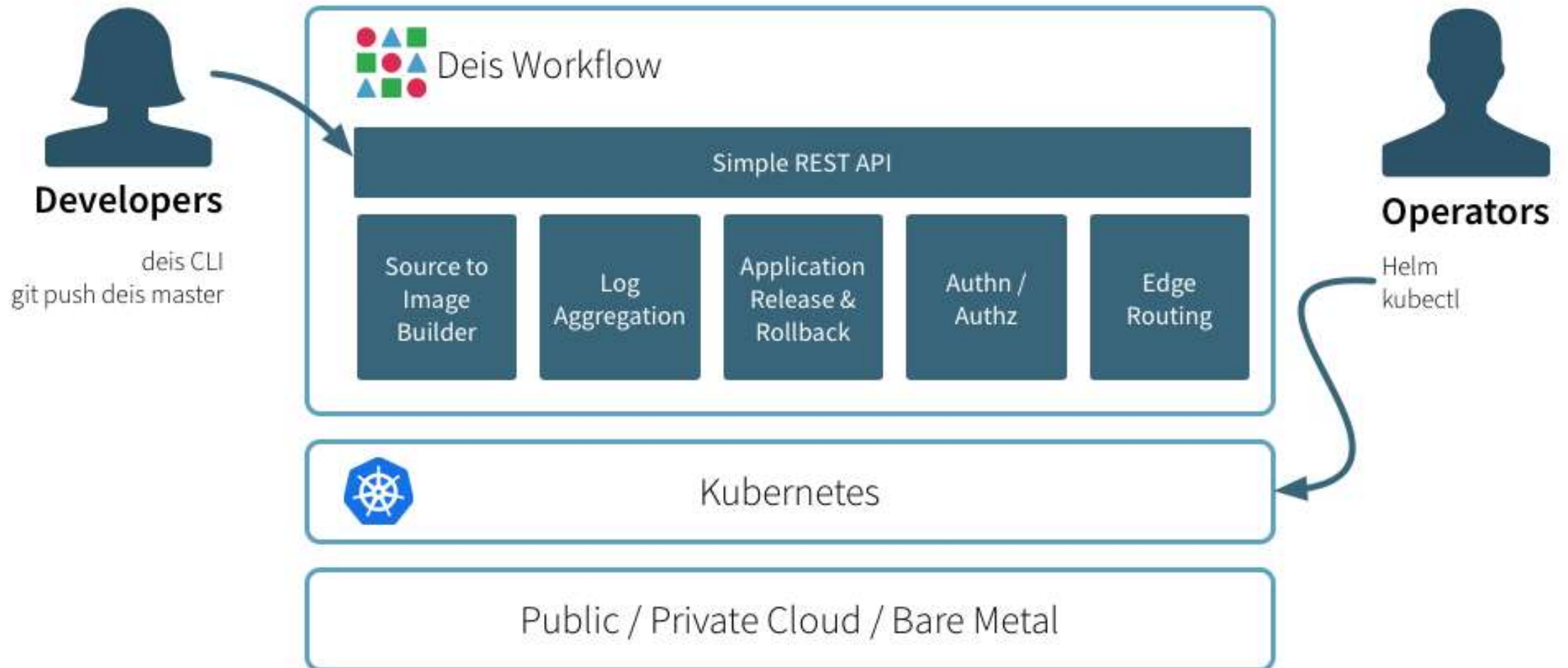
```
kubectl create -f binding.yaml
```

```
apiVersion: servicecatalog.k8s.io/v1alpha1
kind: Binding
metadata:
  name: johnsBinding
spec:
  secretName: johnSecret
  ...Pod selector labels...
```

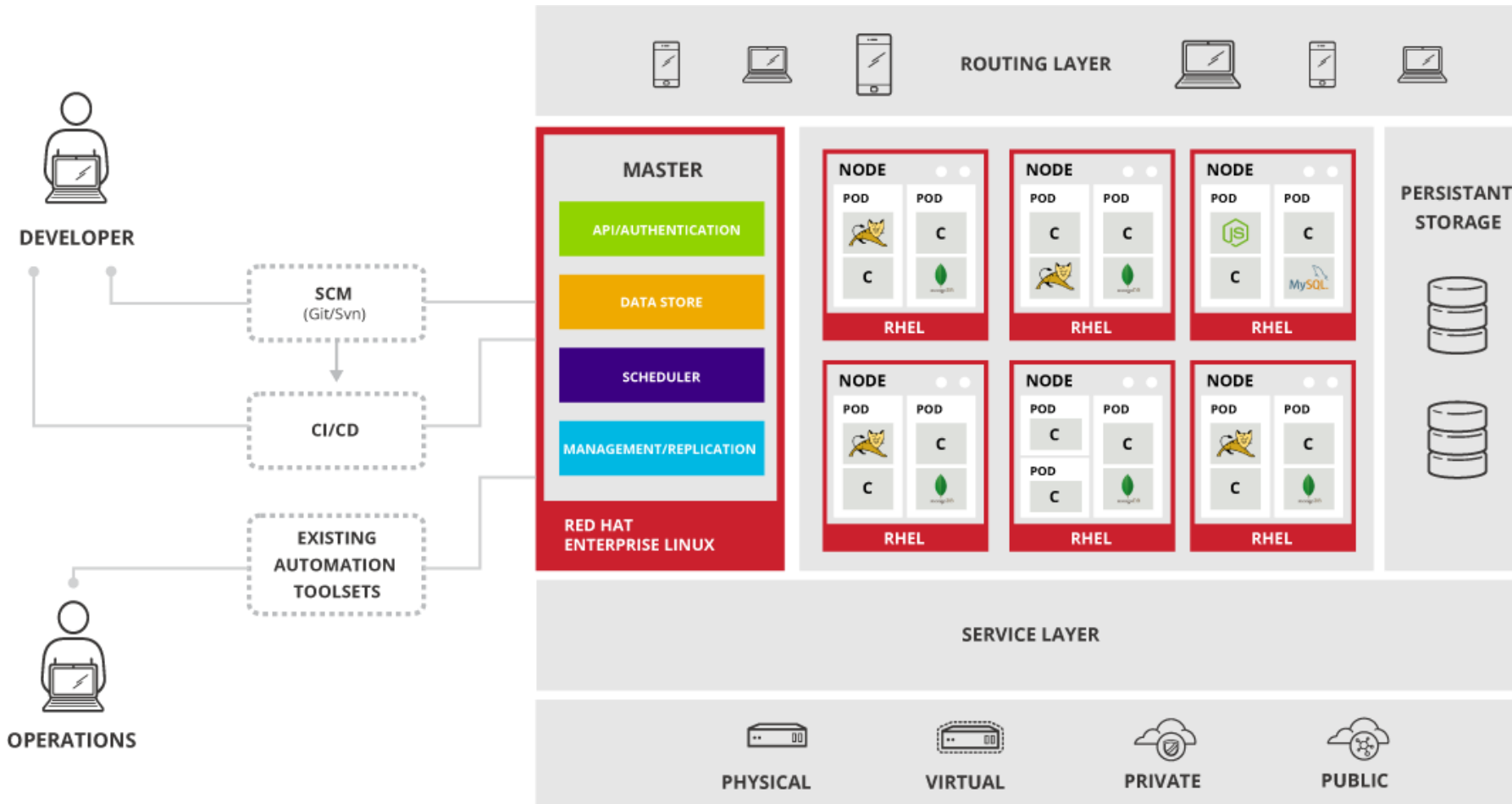
# PaaS nad Kubernetes



<https://deis.com/workflow>

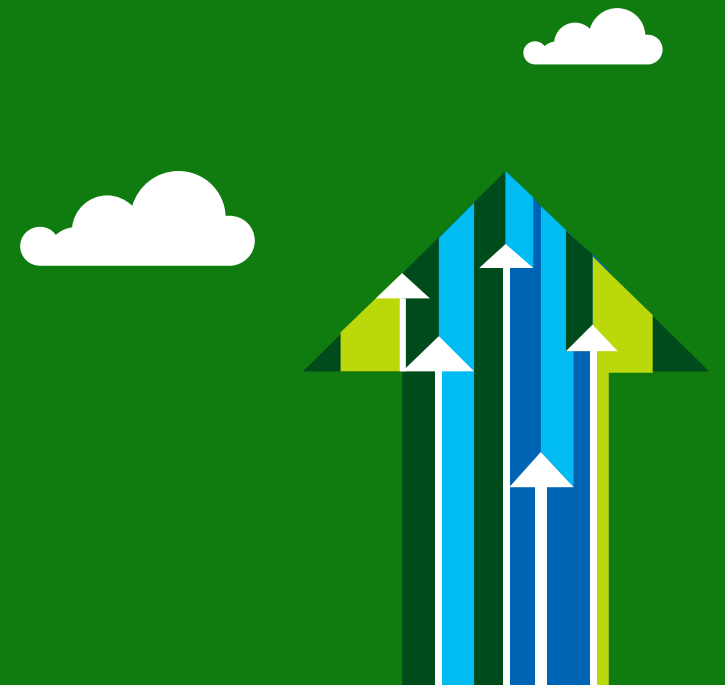


# OpenShift PaaS nad Kubernetes





Azure a kontejnery: skvělá kombinace  
Kubernetes: výborný orchestrátor  
PaaS: nejrychlejší cesta k cíli



Tomáš Kubica, Azure

Blog: <https://tomaskubica.cz>

Email: [tomas.kubica@microsoft.com](mailto:tomas.kubica@microsoft.com)

LinkedIn: <https://www.linkedin.com/in/tkubica/>

