

# CO NÁS DNES ČEKÁ

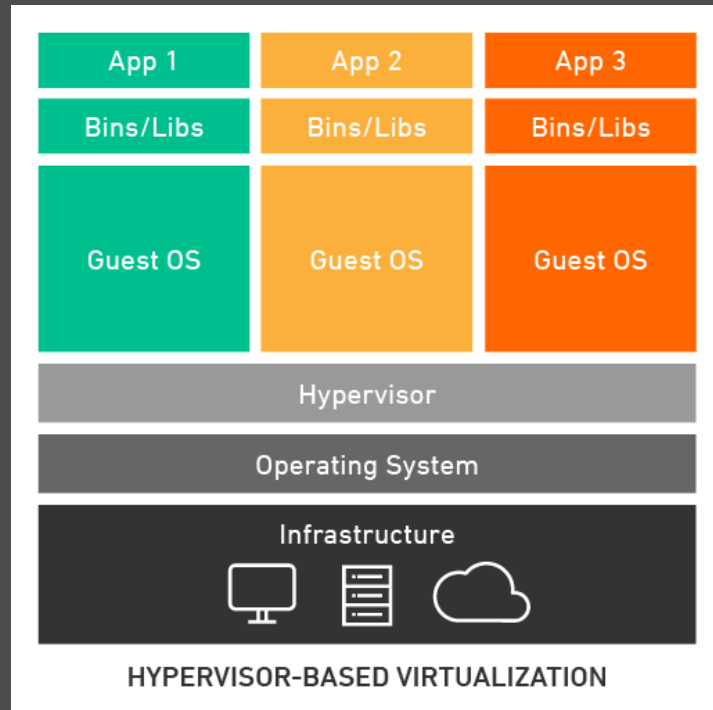
## Agenda webináře

- Srovnání VM vs. Kontejnery
  - Univerzálnost virtualizace a zjednodušování u kontejnerů
  - Typy služeb v kontejnerech
  - Spolupráce kontejnerů a virtualizace
  - VMware Tanzu – nový cluster
  - Plánování nového clusteru
  - Komponenty
  - Kam s čím
- 
- Q & A

## VIRTUALIZACE VS KONTEJNERIZACE

### Kontejnizace jako systém se sdíleným OS

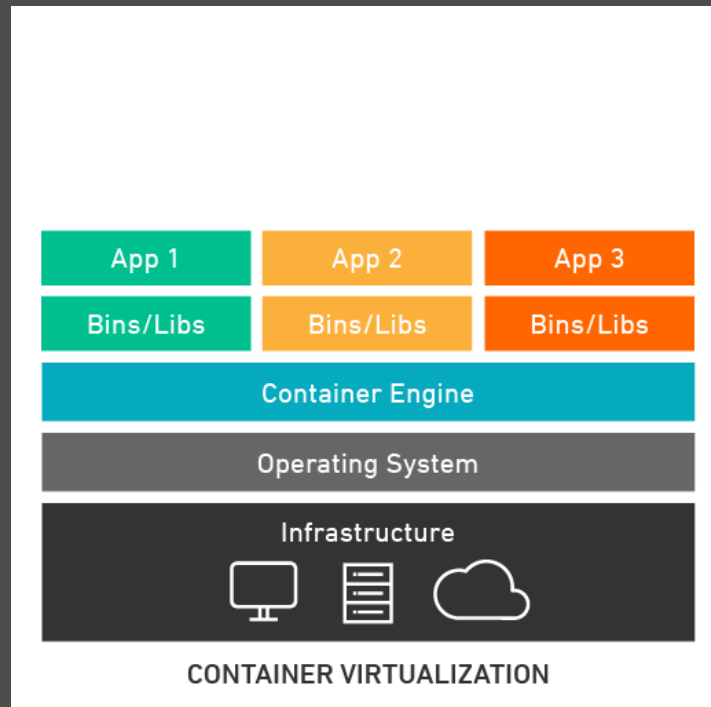
- Vymezuje se např. instrukční sadou
- Standardně řešíme výkonnost komponent
  - ptáme se na frekvenci CPU
- Řešíme použitou virtualizaci
  - VMware, HyperV, KVM
  - malá přenositelnost
- Virtuální servery jsou velké
  - Začínáme na 10GB, standardně mezi 30-50 GB
  - Systém bobtná
- Upgrade systému = upgrade OS + Libs + Apps



# VIRTUALIZACE VS KONTEJNERIZACE

## Kontejnarizace jako systém se sdíleným OS

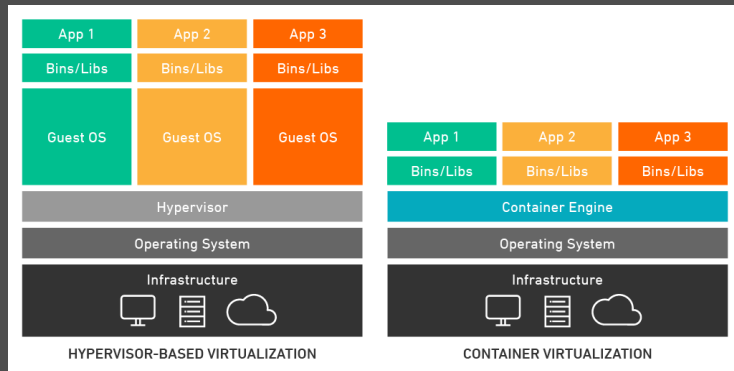
- Vymezujeme se především OS
  - Windows, Linux
- Kontainer nenese vlastní „Guest OS“
  - využívá služeb jádra Host OS
- HW pro containerizaci
  - dál si nese vymezení – např. instrukční sadou
  - neomezuje nás na virtualizaci ani ji nevyžaduje
- Aplikace tvoří podstatnou část



# VIRTUALIZACE VS KONTEJNERIZACE

## Kontejnizace jako systém se sdíleným OS

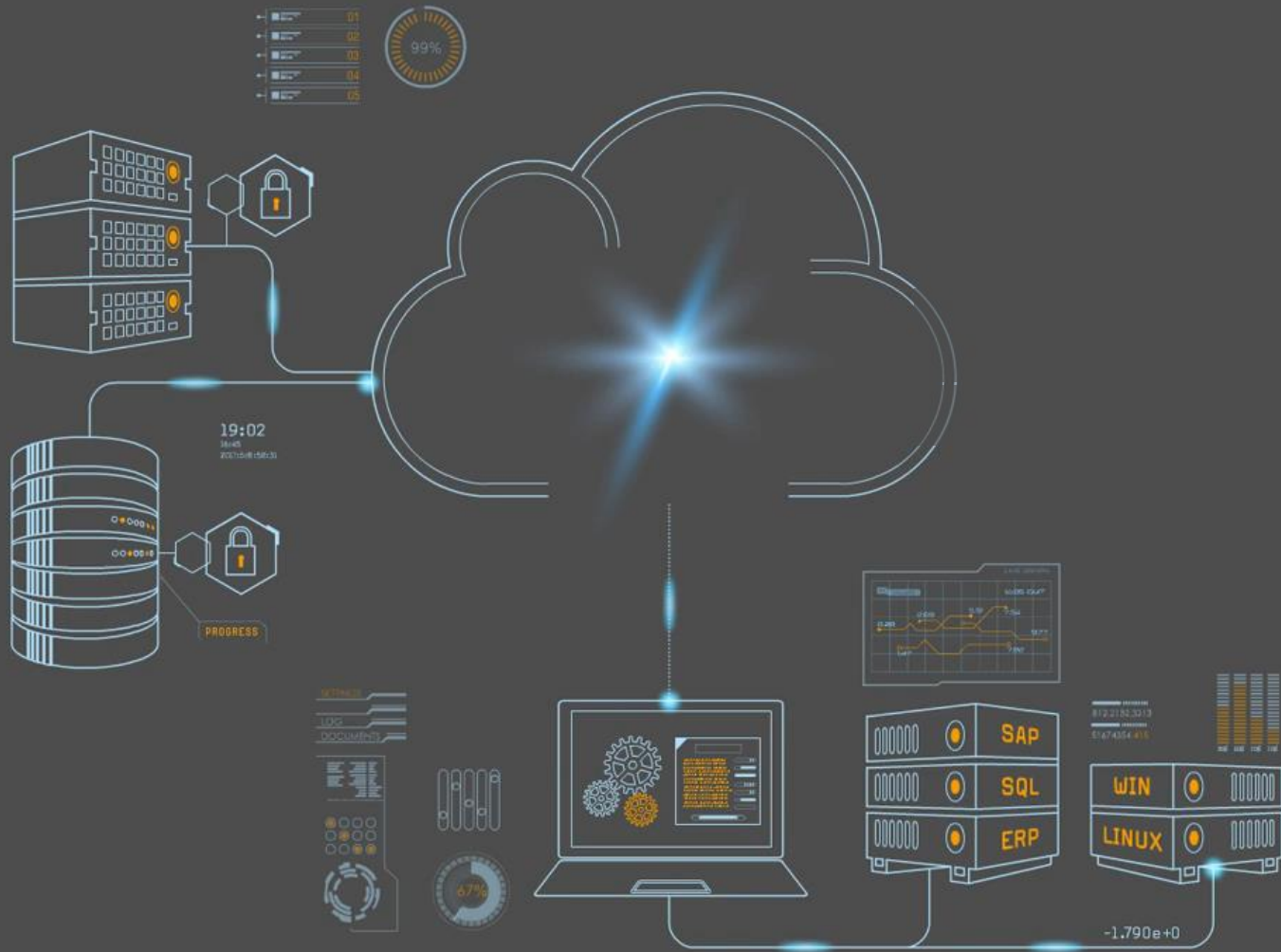
- Typicky x0 – x00 serverů
- Stabilně udržované – long-lifecycle
- Spravováno ručně se smyslupnými pravidly
- Velká míra individualizace správců
- Pomalá změna při změně prostředí
- Velké nároky při upgrade, často nutné



virtualizace

kontejnizace

- Až x000 služeb
- Short-lifecycle – častý progress
- Automatický deployment i testování
- Spravováno nástrojem
- Pevné předpisy pro deployment
- Velká dynamika ve vývoji i v adaptabilitě prostředí
- Upgrade jako nativní součást běhu

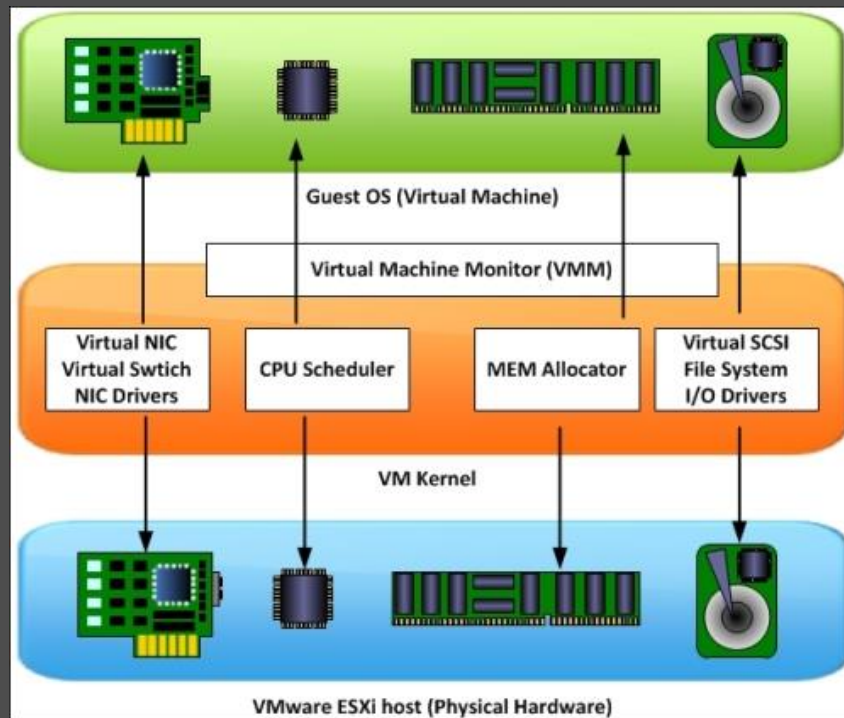


# UNIVERZÁLNOST VS PŘENOSITELNOST

## VIRTUALIZACE SE SNAŽÍ BÝT UNIVERZÁLNÍ

Virtualizace ve snaze o maximální užitečnost a podobu vůči BareMetalu

- Řešíme množství IO device
  - síťovky
  - disky
  - potenciálně HUD
- Pinování a rezervace zdrojů
  - Pinování CPU
  - Rezervace vs. Alokace CPU a RAM
- Passthru zdrojů
  - Síťové karty
  - Grafické karty



## KONTEJNER ZJEDNODUŠUJE

Standardně propagujeme port(y) + protokol a synchronizovanou složku(y)

- Volumes
  - synchronizuje složku s Host filesystemem
- Ports (Expose)
  - propaguje síťovou službu kontejneru
- Image
  - definuje obsah kontejneru
  - běžně popsán skrze Dockerfile

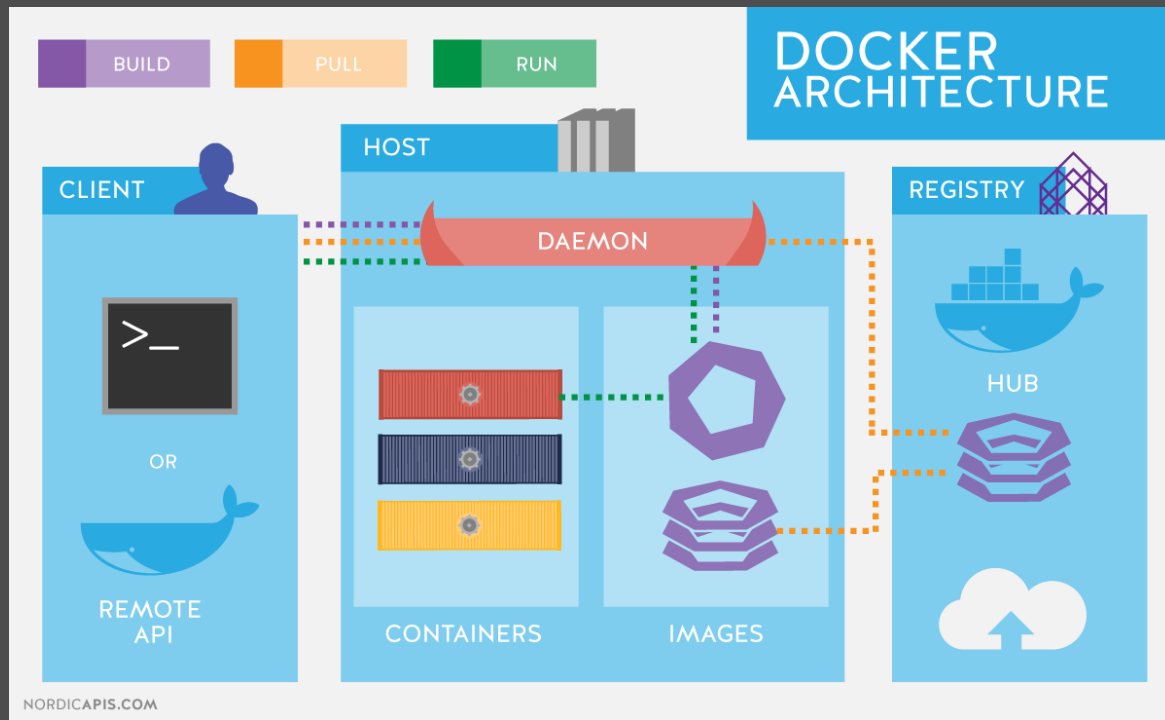
```
docker-compose.yml
1  version: "3"
2  services:
3  server:
4    image: golang:1.11.1
5    volumes:
6      - './server:/go'
7    ports:
8      - '8080:8080'
9
10   command: '-ti'
11   entrypoint:
12     - '/bin/bash'
13
```



# DOCKER JAKO DNEŠNÍ FENOMÉN

## Kontejnizace jako systém se sdíleným OS

- Dnes nejrozšířenější engine
  - používá se jich více
- Přímou podporuje znovupoužitelnost
- Snadné pro testování a vývoj
- Nevyhovující jako kopie velkých produkčních systémů



## KONTEJNERY V PRODUKČNÍM NASAZENÍ

Potřeby se neliší od jiné platformy, liší se jen forma



### Aaaah! Something went wrong

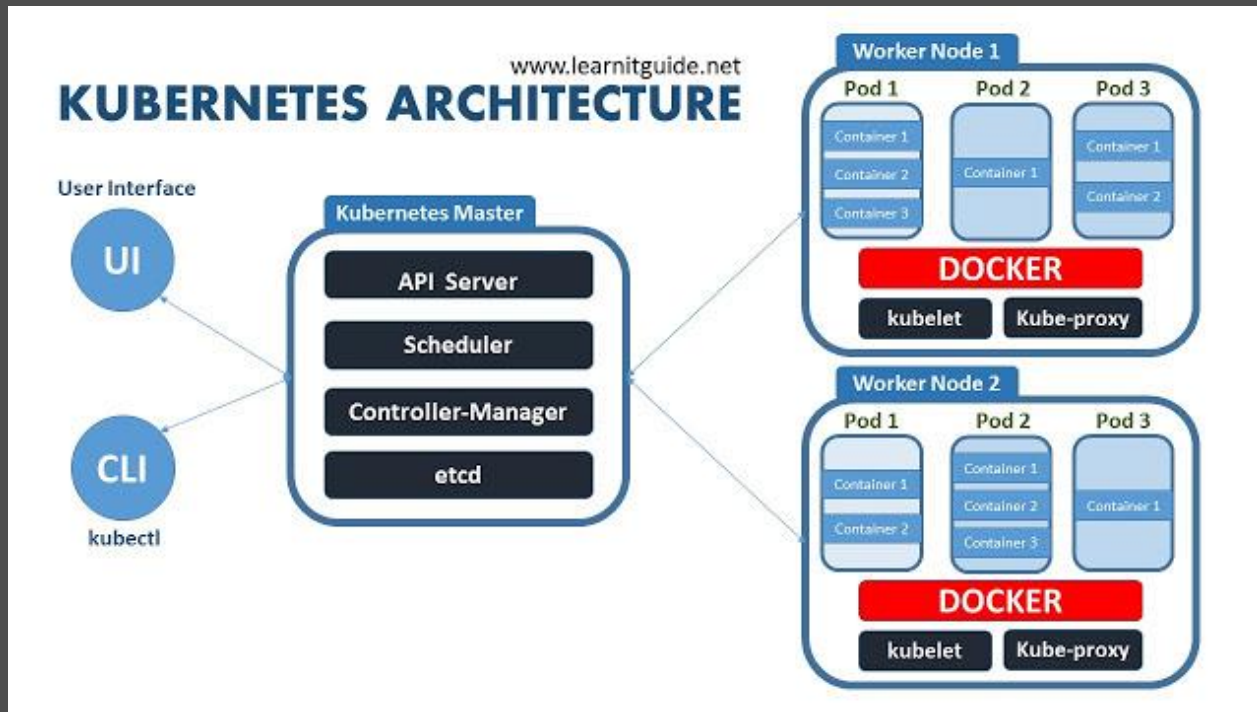
Brace yourself till we get the error fixed.  
You may also refresh the page or try again later

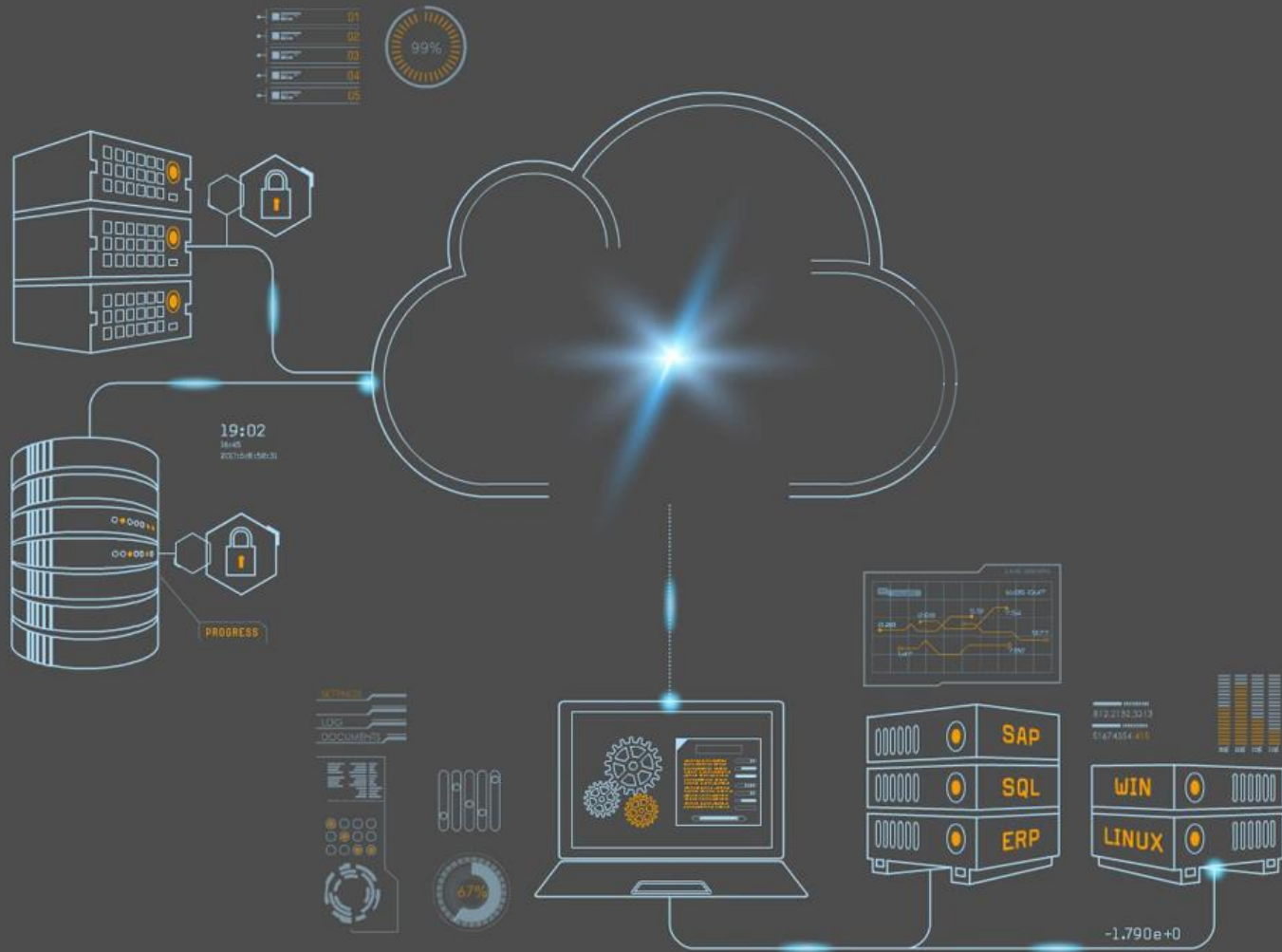
- Produkce není test
- Jako uživatelé chceme mít služby dostupné
- I DevOps týmy mají svoji potřebu kvality, ačkoliv jsou jistě tolerantnější
- Přichází komplikace
  - Síťování
  - Ukládání a práce s daty

# KUBERNETES JAKO „CLUSTER DOCKERŮ“

Kubernetes jsou velkou orchestrací kontejnerů s pevným předpisem

- Oddělení rolí
  - Master
  - Worker
- Dostupnost API
  - typicky kubectl
- Mechanismus k zajištění HA
- Služba je typicky POD
- Problémy
  - řízení přístupu
  - množství komponent



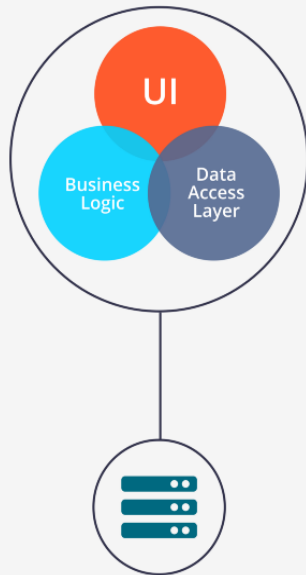


## TYPY SLUŽEB

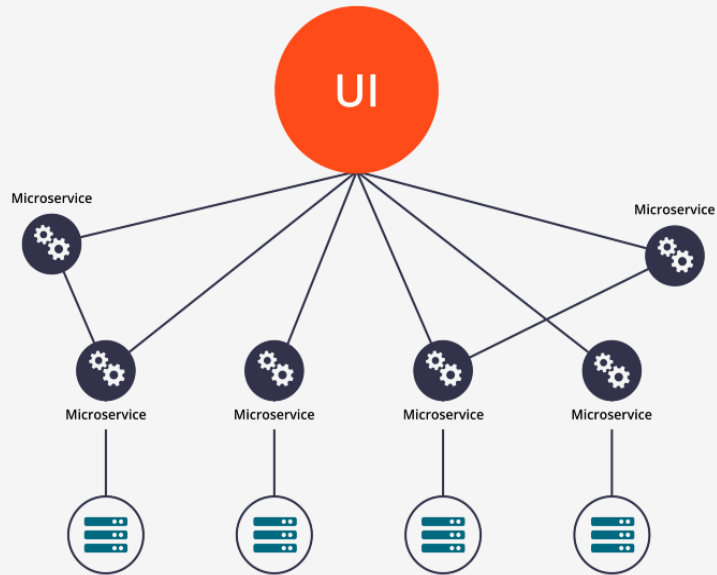
# MONOLITHIC VS MICROSERVICE

Rozdíl je známý, změna je velmi komplikovaná

- poměr např. 1 : 100
- perioda obměny >10 let
- Agilní vs. WaterFall vývoj
- Znovu-použitelnost
  - zdrobnění
  - zobecnění
- Škálovatelnost vs Rychlost



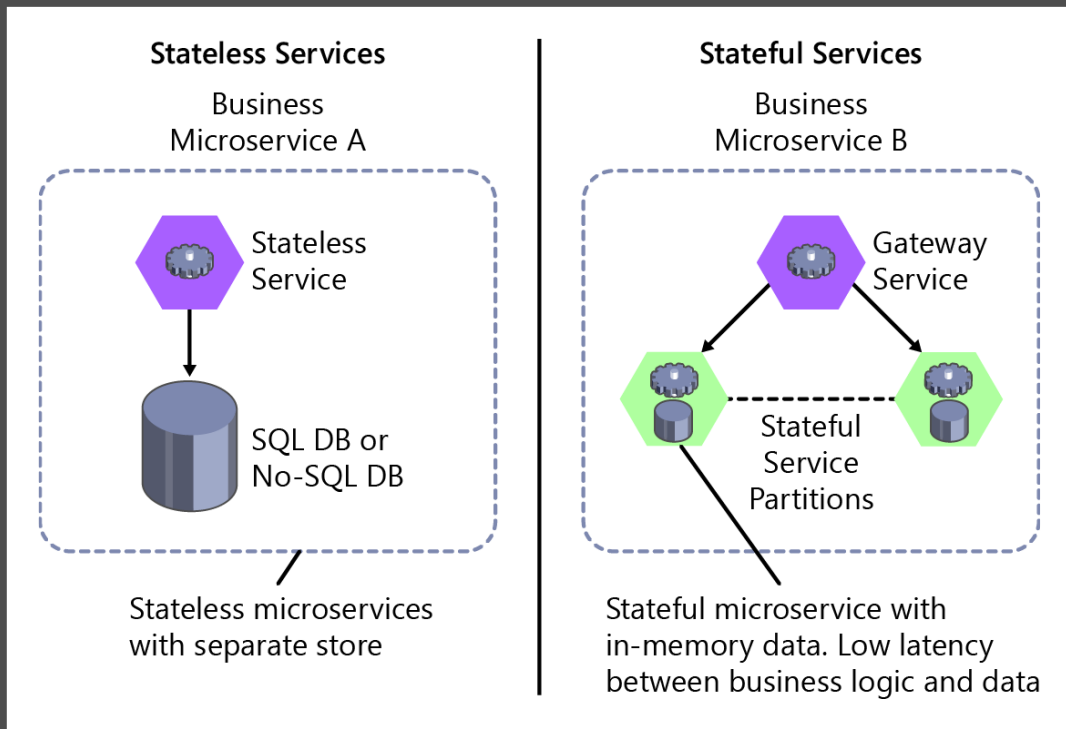
Monolithic Architecture



Microservice Architecture

# STAVOVÉ VS NESTAVOVÉ

Stavový kontejner vyžaduje speciální zacházení



- Stavové (stateful)
  - synchronizace stavu
  - specifické škálování
- Důležitost (obnovitelnost) stavu
  - Session Data
  - SQL Databáze
- Nestavové (stateless)
  - výkonnost
  - event processing
  - opakovatelnost
  - FaaS

# MONOLITICKÝ KONTEJNER

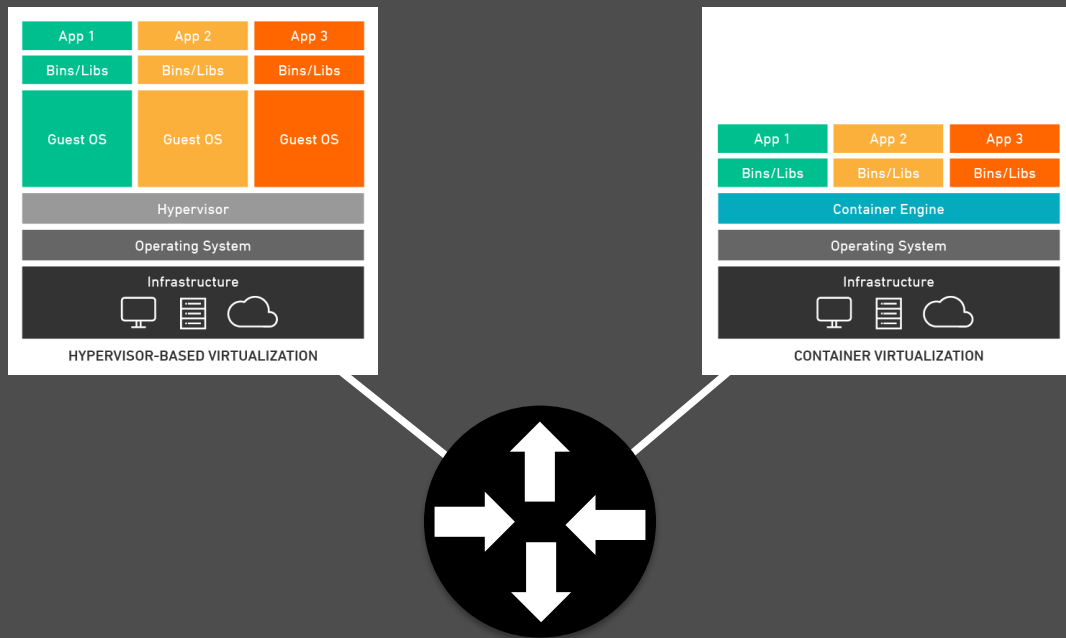
Existující téma vypadající jako cesta správným směrem

- Typicky stavový
- Omezený výkonem na HW
  - zásadně vstupuje do plánování zdrojů
- Neschopný efektivně škálovat
  - bez distribuce
  - plná redundance
- Složité opustit daný cyklus
  - I využívanou platformu
- Běžný problém
  - Legacy IT => CloudNative Apps



# KONTEJNERY VE SPOLUPRÁCI S VIRTUALIZACÍ

Skvělý začátek jiného přístupu



- Vzájemná propojitelnost
- Využívání společných zdrojů
  - sítě
  - síťových služeb
  - úložišť
  - CPU/memory
- Prostor pro vytvoření nového vývojového cyklu
- Více přístupů
  - Oddělená infrastruktura
  - Container Engine na VM
  - Container Engine jako součást hypervisoru



# VMWARE TANZU

## Založení kontejner clusteru přímo v prostředí IaaS

- vCloud Director
- Prostředí VDC
  - Virtualizace
  - Kontejnerizace
- Společné prostředky
  - CPU/RAM
  - Síťové služby
  - Sdílená storage
- Jedna z možných cest
  - API (např. Terraform)
  - MaaS

The screenshot displays the VMware Cloud Director interface. The top navigation bar includes 'VMware Cloud Director', 'Data Centers', 'Applications', 'Networking', 'Libraries', 'Administration', 'Monitor', and 'More'. The main content area is titled 'Virtual Data Center' and 'Dedicated vSphere Data Centers'. It shows an overview of the environment with 1 Site, 1 Organization, and 1 Virtual Data Center. The 'Running Applications' section shows 0 VMs and 0 vApps. The 'Used Resources' section shows 20 GHz CPU, 30 GB Memory, and 0 MB Storage. A table below shows the resource usage for the 'marek\_flex-vdc' VDC, which is owned by 'Marek\_Erneker' and 'vcd.lab.cloud'. The table has columns for Applications, CPU, Memory, and Storage. The CPU usage is 20 GHz, Memory is 30 GB, and Storage is 0 MB. A note indicates that 1.2 TB of storage is allocated for running VMs. At the bottom, there is a 'Recent Tasks' section with a table that currently shows 'No recent tasks found'.

Applications	CPU	Memory	Storage
0 vApps	20 GHz	30 GB	0 MB
0 of 0 Running VMs			1.2 TB allocated

# VYTVÁŘÍME KUBERNETY – VERZE A LOKALITA

## Co řešíme při vytváření kubernetes clusteru?

- Volba Virtual Data Center
  - enabler pro využívání více lokalit
  - georedundance + přenositelnost = vyšší dostupnost + nezávislost

### Select the Virtual Data Center for the cluster

Name	Organization
<input checked="" type="radio"/> marek_flex-vdc	Marek_Erneker

1 - 1 of 1 Virtual Data Center

### Select the Kubernetes Policy and Version to use

Name	Description
<input checked="" type="radio"/> vcd-marek-ns01	vcd-marek-ns01

1 - 1 of 1 Kubernetes Policy

Kubernetes Version \*

- ✓ v1.17.17
- v1.16.8
- v1.16.12
- v1.16.14
- v1.17.7
- v1.17.8
- v1.17.11
- v1.17.13
- v1.18.5
- v1.18.10
- v1.18.15
- v1.19.7

- Volba Kubernetes Version
  - Agilní vývoj
  - 3 podporované Major verze
  - Replacement-Upgrade vs Rolling-Upgrade

# VYTVÁŘÍME KUBERNETY – PLÁNOVÁNÍ ZDROJŮ

Co řešíme při vytváření kubernetes clusteru?

- Počet Master & Worker nodů
  - Master – definuje dostupnost a operabilitu
  - Worker – definuje dostupnost výkonu samotného

Input number of control plane/worker nodes, and select the machine classes to use

Number of Control Plane Nodes *	1	
Number of Worker Nodes *	3	

## Select Worker Machine Class

	Name	CPU Count	CPU Reservation	Memory	Memory Reservation
<input type="radio"/>	best-effort-xsmall	2	0 MHz	2 GB	0 MB

- Master/Worker Machine Class
  - Alokuje/Rezervuje výkon na hypervisoru
  - Klíčové z hlediska dosažitelného maxima
    - Na celek
    - Na container
  - Jsou předdefinované providerem

# VYTVÁŘÍME KUBERNETY – PŘIPRAVENOST

Vytvořený cluster je připraven k operativě

- Zaměřeno na rychlost a jednoduchost
- Zpřístupňuje kubeconfig

The screenshot displays the VMware Cloud Director interface. At the top, there is a navigation bar with the VMware logo and menu items: VMware Cloud Director, Data Centers, Applications, Networking, Libraries, Administration, Monitor, and More. Below the navigation bar, the main content area is titled "Kubernetes Container Clusters". There are four action buttons: NEW, RESIZE, DOWNLOAD KUBE CONFIG, and DELETE. A table lists the clusters:

















	Name	Status	Kubernetes Provider	Kubernetes Version	Virtual Data Center	Owner
<input type="radio"/>	devops-1-17	running	vSphere with Tanzu	v1.17.17	marek_flex-vdc	marek.erneker
<input type="radio"/>	vdc-marek-cluster01	running	vSphere with Tanzu	v1.17.13	marek_flex-vdc	miroslav.kis

At the bottom right of the table area, it says "1 - 2 of 2 Kubernetes Clusters". Below the table is a "Recent Tasks" section with a table:

Task	Status	Type	Initiator	Start Time	Completion Time
Created devops-1-17 - urn:vcloud.entity:vmware:tkgcluster:1.0.0:0ac5a336-30b8-4b7...	Succeeded	definedEntity	marek.erneker	03/23/2021, 01:25:28 PM	03/23/2021, 01:29:58 PM

# VYTVÁŘÍME KUBERNETY – KOMPONENTY

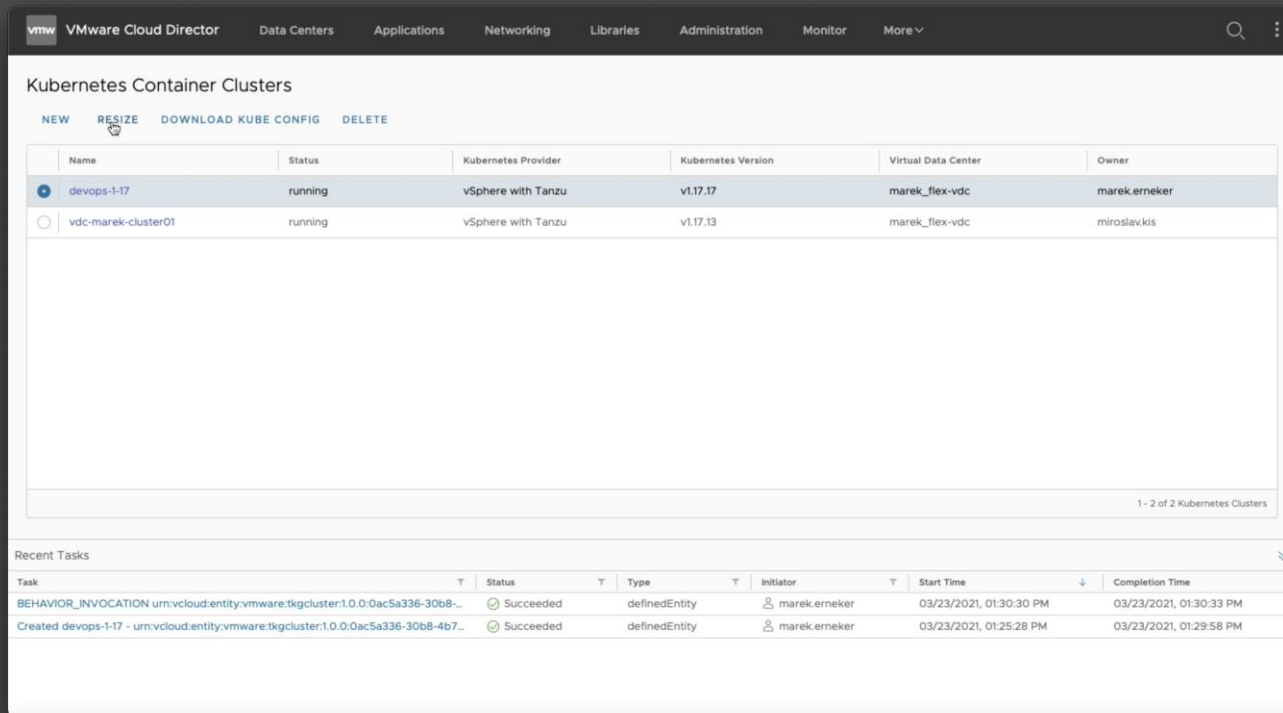
Vytvořený cluster je připraven k operativě

 <b>Networking</b> 	 <b>Ingress</b> 	 <b>Egress</b> 	 <b>Registry</b> 	 <b>Volume</b> 	 <b>CI/CD</b> 	 <b>Monitoring</b> 	 <b>Logging</b> 
<b>Antrea</b>	<b>Avi over NSX-T IPv4, IPv6</b>	<b>Avi over NSX-T</b>	<b>Harbor 1.3 Harbor 2</b>	<b>vSAN NFS</b>	<b>Concourse- CI GitLab</b>	<b>Promethe us + Grafana</b>	<b>Grafana- Loki</b>

# VYTVÁŘÍME KUBERNETY – ŠKÁLOVÁNÍ

Drobnost nebo nutnost?

- Podporuje škálování vůči infrastruktuře
- Pouze Worker nody
- Dostupné skrze API
- Size Up i Down



The screenshot displays the VMware Cloud Director interface for managing Kubernetes Container Clusters. The top navigation bar includes 'Data Centers', 'Applications', 'Networking', 'Libraries', 'Administration', 'Monitor', and 'More'. The main content area is titled 'Kubernetes Container Clusters' and features a table with the following columns: Name, Status, Kubernetes Provider, Kubernetes Version, Virtual Data Center, and Owner. Two clusters are listed: 'devops-1-17' (running, vSphere with Tanzu, v1.17.17, owned by marek.erneker) and 'vdc-marek-cluster01' (running, vSphere with Tanzu, v1.17.13, owned by miroslav.kis). A 'RESIZE' button is highlighted above the table. Below the table, a 'Recent Tasks' section shows two successful tasks: 'BEHAVIOR\_INVOCATION um:vcloud:entity:vmware:tkgcluster:1.0.0:0ac5a336-30b8-...' and 'Created devops-1-17 - um:vcloud:entity:vmware:tkgcluster:1.0.0:0ac5a336-30b8-4b7-...'.

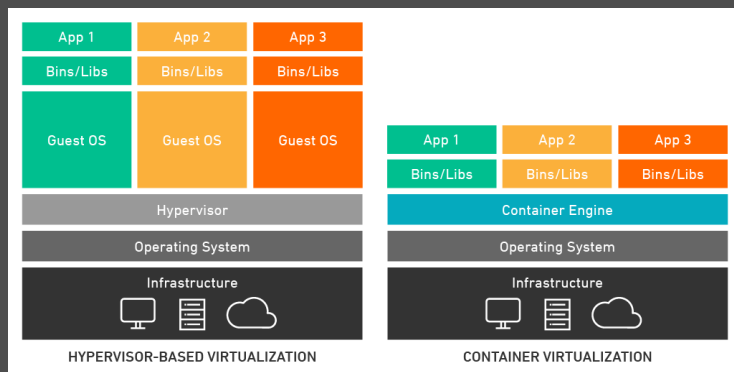
Name	Status	Kubernetes Provider	Kubernetes Version	Virtual Data Center	Owner
devops-1-17	running	vSphere with Tanzu	v1.17.17	marek_flex-vdc	marek.erneker
vdc-marek-cluster01	running	vSphere with Tanzu	v1.17.13	marek_flex-vdc	miroslav.kis

Task	Status	Type	Initiator	Start Time	Completion Time
BEHAVIOR_INVOCATION um:vcloud:entity:vmware:tkgcluster:1.0.0:0ac5a336-30b8-...	Succeeded	definedEntity	marek.erneker	03/23/2021, 01:30:30 PM	03/23/2021, 01:30:33 PM
Created devops-1-17 - um:vcloud:entity:vmware:tkgcluster:1.0.0:0ac5a336-30b8-4b7-...	Succeeded	definedEntity	marek.erneker	03/23/2021, 01:25:28 PM	03/23/2021, 01:29:58 PM

# KAM S ČÍM

Mohu-li si vybrat, dosáhnou lepšího chování své aplikace

- Monolitické celky
- Stavové nedistribučitelné aplikace
  - SQL servery M+S
- Aplikace s neschopností automatického deploymentu
  - Např. vyžadují ruční vstup
  - Nejsou na to připraveny



virtualizace  
+  
bare-metal

kontejnerizace

- Nestavové aplikace
  - Data Processing
  - Cache Systémy
- Distribuované aplikace
  - řada NoSQL DB
- Aplikace se schopností škálování
  - Škálovací mechanismus
- Aplikace s normalizovatelnou podobou
  - Nevymýšlejíme kolo



Q & A





DĚKUJEME ZA POZORNOST