riganti

Update Conference
Prague 2019

DOTVVM

Microsoft® Most Valuable Professional MVP

Microsoft Azure Training Day

Modernizing Data,
Applications & APIs to the Cloud

Tomáš Herceg

CEO @ RIGANTI

Co-founder of Update Conference

Microsoft MVP

tomas.herceg@riganti.cz

@hercegtomas

Who are we?

Custom Software Development

Solution Architecture

Consulting & Training

Tools & Products for .NET developers

riganti

- Founded in 2011
- Based in Prague & Brno
- Focus on .NET & Microsoft
  - Software development
  - Training & consulting

Tomáš Herceg

- Microsoft MVP
- Author of technical articles at DotNetPortal
- Conference speaker
  - *.NET Conf 2019*
  - *Update Conference Prague*
  - *DevSum Stockholm*
  - *DotNext St. Petersburg*
  - *.NET Dev Days Warsaw…*

Agenda

- **9:30 – 10:30** **Migrating web applications to Azure**
- 10:50 – 11:50 Moving your database to Azure
- 11:50 – 12:50 Deploying your application faster and safer
- 13:30 – 14:30 Modernizing your application with containers and serverless
- 14:50 – 15:50 Consolidating infrastructure with Azure Kubernetes Service

Web Application options in Azure

- **Azure App Service**
  - Similar to traditional „web hosting“
  - Windows & Linux support
  - ASP.NET, Java, PHP, Node.js, Python, Ruby…
- **Azure Container Instance, Azure Kubernetes Service**
  - For containerized applications
- **Azure Virtual Machines**
  - Classic Infrastructure as a Service scenario

Azure App Service

- App Service Plan
  - „Represents“ a virtual server instance (“hosted IIS”)
  - Can host multiple web applications
  - Pricing corresponds with the selected tier
    - X GB RAM
    - X virtual CPUs
    - X GBs of available filesystem storage
  - Windows & Linux support

- Web App
  - A single web application running in a specific App Service Plan

Azure App Service

- There is no direct control of the server
  - Cannot RDP to it
  - Cannot change configuration or install anything on the server
  - Can access only your own space in the filesystem
  - Everything is set up through the Azure portal

- Automatic or manual scaling
- Deployment slots
- Geo redundancy
- Web Jobs

Background jobs in Azure App Service

- Azure Web Jobs
  - Small console apps running side by side with the web app
  - Easy deployment

- Continuous or manually triggered
- Can be triggered based on an external event
  - New item in Azure Storage Queue
  - New entry in Cosmos DB
  - Timer
  - …

Filesystem in App Service

- Persistent
- Backed in Azure Storage
- Limited capacity
  - Based on App Service Plan
- Shared between all instances of the application (scaling)
- Be careful about performance and large files
  - Azure Blob Storage will work better

Azure Storage

- Blob Storage
  - Simple storage for binary data
  - Not a real filesystem – no folders, limited listing capabilities
- Table Storage
  - Very simple document database
  - Limited filtering and sorting capabilities, no foreign keys
  - Quick key-based lookup
- Storage Queues
  - Reliable publisher-subscriber flows

Blob Storage

- Cheapest way to store large data
- Great scalability

- One or more Blob Containers
  - Each container can contain multiple blobs
    - Use / in blob name to simulate hierarchies (they are no folders)
  - Can enable public access via HTTPS
    - The URL of the blob will be publicly accessible

Storage Queues

- Anyone can send messages to the queue
  - Max size of a message: 64kB

- Queue can have multiple readers
  - The message will be delivered to one of them
  - If the message is not processed within a timeout, it’ll reappear in the queue

- WebJobs can be triggered by queue message

DEMO

Migrating web app in Azure

Other options for App Service

- App Settings and Connection Strings
- Virtual directories and applications
  - Run sub-application at **/something**
- HTTPS
- Automatic and manual scaling
- Diagnostic tools (Kudu)
  - *yourappname*.scm.azurewebsites.net

Agenda

9:30 – 10:30	Migrating web applications to Azure
10:50 – 11:50	**Moving your database to Azure**
11:50 – 12:50	Deploying your application faster and safer
13:30 – 14:30	Modernizing your application with containers and serverless
14:50 – 15:50	Consolidating infrastructure with Azure Kubernetes Service

Database options in Azure

- Azure SQL Database
- Microsoft SQL Server in Virtual Machines
- Azure Cosmos DB
- MySQL
- MariaDB
- PostgreSQL
- Redis

Azure SQL Database

- Hosted SQL Server
- Some limitations
  - No cross-database queries…

- Managed Instance
  - Pricing based on server configuration – RAM, disk storage etc.
- Single Database, Elastic Pools
  - Pricing based on DTU (database transaction unit)

Azure SQL Database

- DTU
  - “Average” database transaction
  - 5 DTU = 5 average database transactions per second
- Serverless option
  - Pay less when the database is not used for a longer period of time
  - Auto-pause interval – 60 minutes or more

Azure SQL Database configuration

- Azure SQL Database Server
    - Just a logical container for multiple databases
        - Quick provisioning, nothing is installed
    - Holds configuration of firewall rules and user accounts

- Azure SQL Database
    - Assigned to the specific server

Azure SQL Database deployment options

- Export/import using SQL Management Studio
  - BACPAC format
  - SQL scripts
- Transactional Replication
  - No downtime, but more complicated setup
- Managed Instances: Azure Database Migration Service

DEMO

Moving the database in Azure

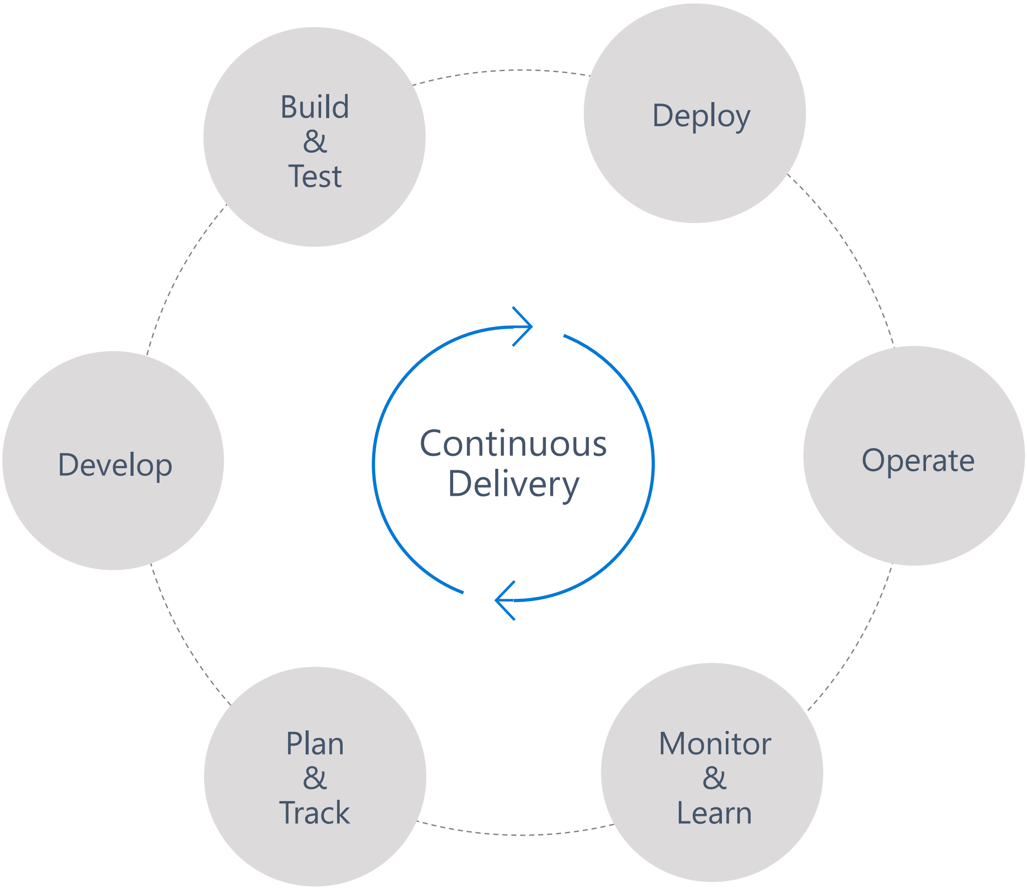
Agenda

- 9:30 – 10:30 Migrating web applications to Azure
- 10:50 – 11:50 Moving your database to Azure
- **11:50 – 12:50 Deploying your application faster and safer**
- 13:30 – 14:30 Modernizing your application with containers and serverless
- 14:50 – 15:50 Consolidating infrastructure with Azure Kubernetes Service

DevOps

- People
- Processes
- Tools

- Shared goal of **continuous delivery**



Why DevOps?

- Better and faster feedback from the users
- Breaking the gap between devs and IT admins
- Higher quality of releases
  - Release smaller changes often
- The build & release process is “documented”
- More insights in the application

Azure DevOps

- Formerly Visual Studio Team Services
- Formerly Visual Studio Online
- Formerly Visual Studio Team Foundation Server
- …

Azure DevOps functionalities

Azure Boards

Azure Pipelines

Azure Repos

Azure Test Plans

Azure Artifacts

Azure DevOps is modular

- You don’t need to use everything
  - Use only things that actually help

- Use external source control
- Use external task management
- Use external build system
- …

Azure Pipelines

- Automated builds and releases
  - Various triggers for build
    - Commit
    - Pull request
    - Regular schedule
  - Multiple release environments
    - Automatic deployment or manual approvals
  - Secret management
    - Passwords, connection strings
    - Certificates for signing

Azure Pipelines agents

- The build and release has to run somewhere

- Build agent
  - Hosted in Azure (check your pricing options for free minutes)
  - Your own VM
- Release agent
  - The same as BA
  - Can be also installed on the target server
    - E.g. deployments to on-premises IIS

Build

- Grab source code
- Restore packages, run npm install etc.
- Build
- Publish
  - create packages to be deployed

- Build output: Build artifacts

Release

- Get the build artifact
- Substitute variables and secrets
- Sign code
- Make the actual deployment

Build and release tasks

- UI for designing builds and releases
  - Easy to understand and discover
  - Tasks are versioned

- YAML definition for builds
  - Releases don’t support it right now, but it is planned
  - Versioned with the source code
    - Great if you use feature branches
  - Easy copy-paste

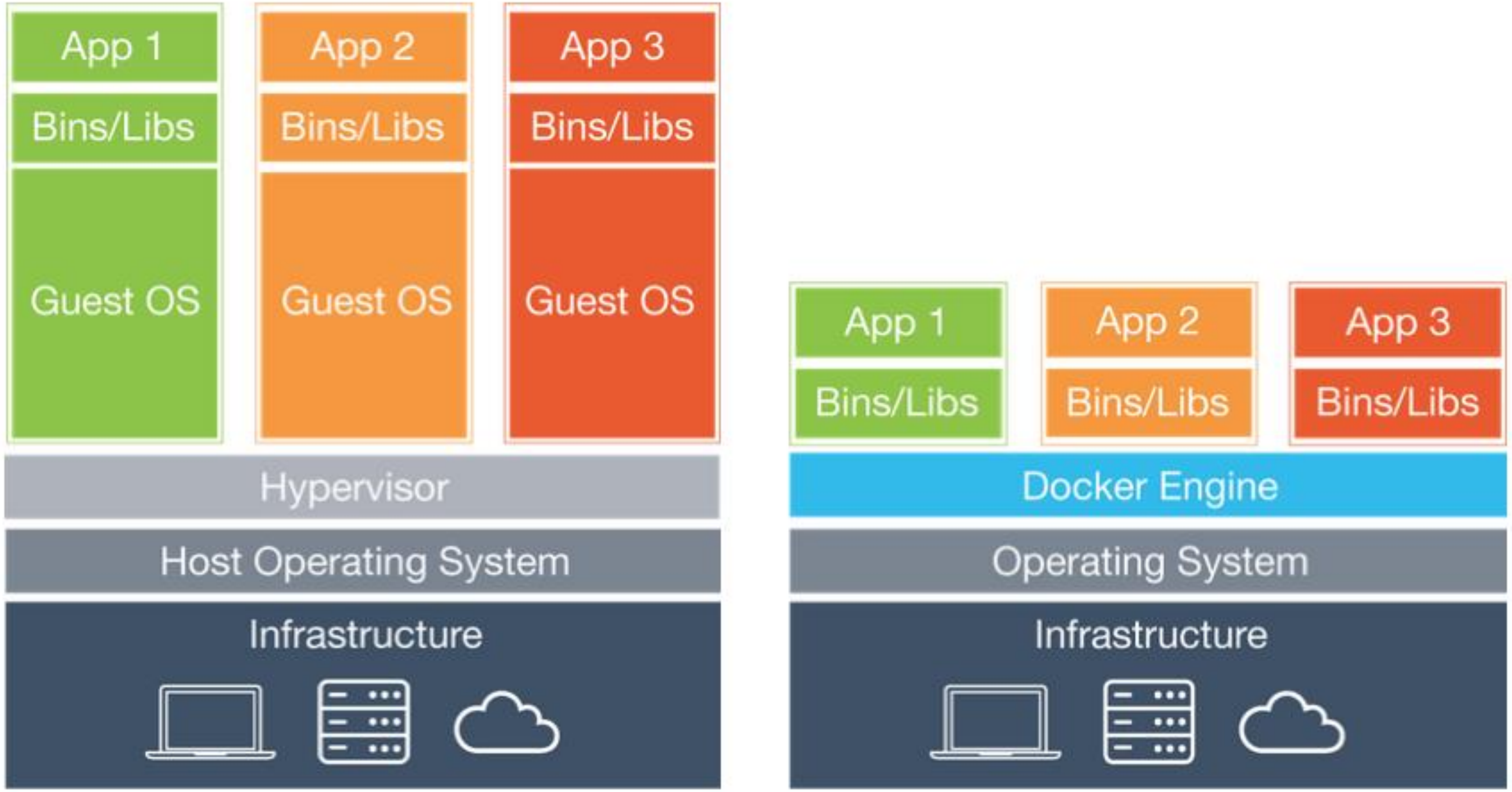
DEMO

Automated builds and releases

Agenda

9:30 – 10:30	Migrating web applications to Azure
10:50 – 11:50	Moving your database to Azure
11:50 – 12:50	Deploying your application faster and safer
13:30 – 14:30	**Modernizing your application with containers and serverless**
14:50 – 15:50	Consolidating infrastructure with Azure Kubernetes Service

What is container?



Containers

- More isolation than process
  - File system, network, registry (on Windows)…
- Less isolation than VM
  - Shared kernel, drivers…

- Quick start
- Copy-on-write file system
- No more “works on my machine” because of missing libraries, frameworks or SDKs

Container filesystem

- Layered approach
- Same layers are shared
  - Identified by their hash

- Avoid changes in the filesystem of containers
  - Configuration is supplied by environment variables
  - The changes are not persistent by default – containers are ephemeral
    - You need to configure a persistent volume for databases etc.

Container filesystem

- 10 containers with 10 different .NET Core apps
  - Underlying OS … same
  - .NET runtime … same
  - App and its libraries … different – they'll have their individual layers

- Example: ASP.NET Core 2.1 container on Linux
  - microsoft/dotnet:2.1-aspnetcore-runtime-alpine3.7
  - microsoft/dotnet:2.1-runtime-deps-alpine3.7
  - alpine:3.7

When to use containers?

- Microservices architecture
  - Different pieces of code in different technologies
  - IT admin doesn’t need to know how to install PHP, Python, Ruby and .NET in correct versions
- Consistent builds
  - All build dependencies are defined in the container
  - Build works for everyone

Container OS

- Linux
  - Very small images – just a few MBs

- Windows
  - Windows Server Core
  - Windows Nano Server
    - Supports also .NET Framework and IIS

How to start

- Install Docker for Windows
  - https://docs.docker.com/docker-for-windows/install/
  - Requires Hyper-V
- Allow access to volume with your source code
  - Debug builds need to do this
- Choose if you want Windows or Linux containers

Azure Container Instance

- “Here is the container – host it for me”

- Simple container applications
  - Single-container apps, or small number of containers
  - No HTTP proxy in front of the app

- Short-time scenarios
  - docs.microsoft.com – runnable C# code samples
  - Per second pricing

Azure App Service (Linux)

- Ideal for containerized web apps
- HTTP proxy in front of the container

- *App Service Plan* = the server
  - Priced by HW specs
  - Multiple applications

- No multi-container support yet
  - In a roadmap

DEMO

Containerizing web app and publishing it as Azure App Service on Linux

Agenda

▪ 9:30 – 10:30	Migrating web applications to Azure
▪ 10:50 – 11:50	Moving your database to Azure
▪ 11:50 – 12:50	Deploying your application faster and safer
▪ 13:30 – 14:30	Modernizing your application with containers and serverless
▪ **14:50 – 15:50**	**Consolidating infrastructure with Azure Kubernetes Service**

Azure Container Service

- Fully managed service for container orchestration

- Supports
  - Kubernetes (AKS)
  - DC/OS
  - Docker Swarm

Kubernetes in Azure

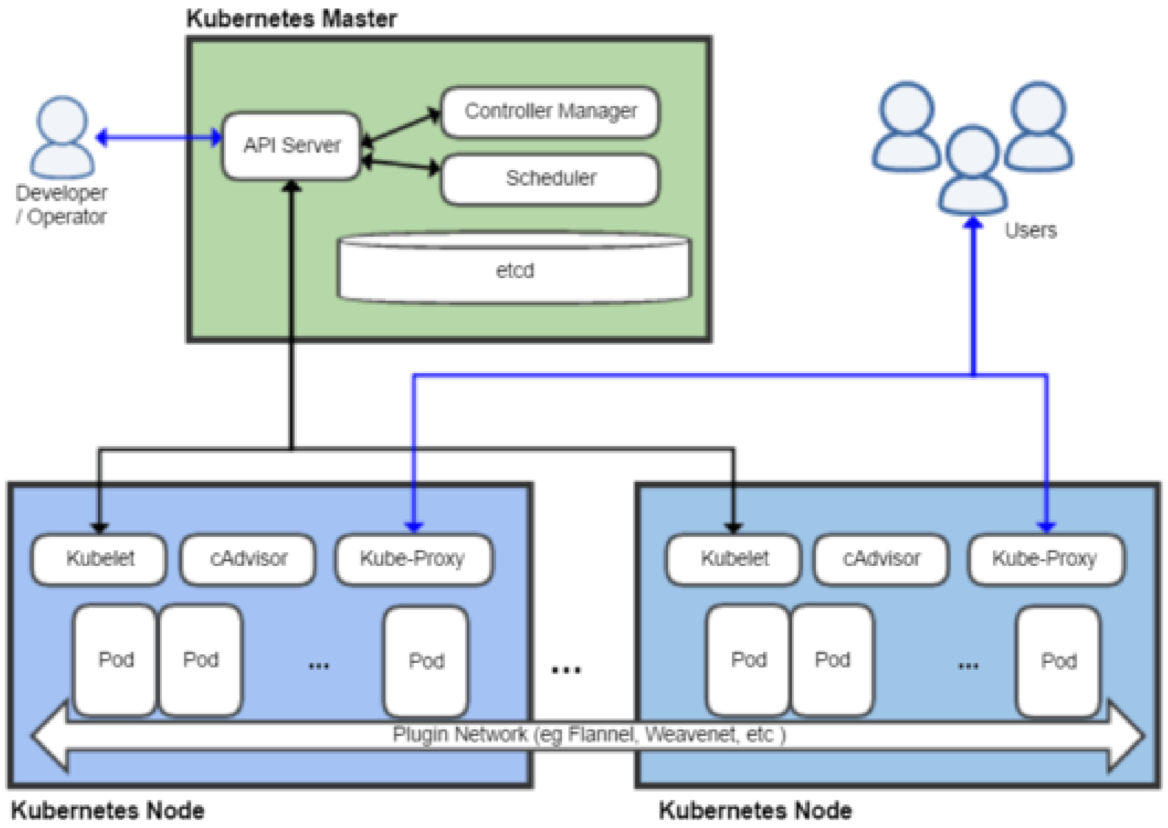
- First-class citizen
- You pay only for worker nodes
  - Management infrastructure is free of charge
- Standard tooling
- Portal integration for diagnostics

Kubernetes

- “K8s”

- Automated deployment and scaling of container workloads
- Supported in all cloud platforms
  - Amazon
  - Microsoft
  - Google
- Can be run on-premises

Kubernetes – architektura



Zdroj: https://en.wikipedia.org/wiki/Kubernetes#/media/File:Kubernetes.png

Pod

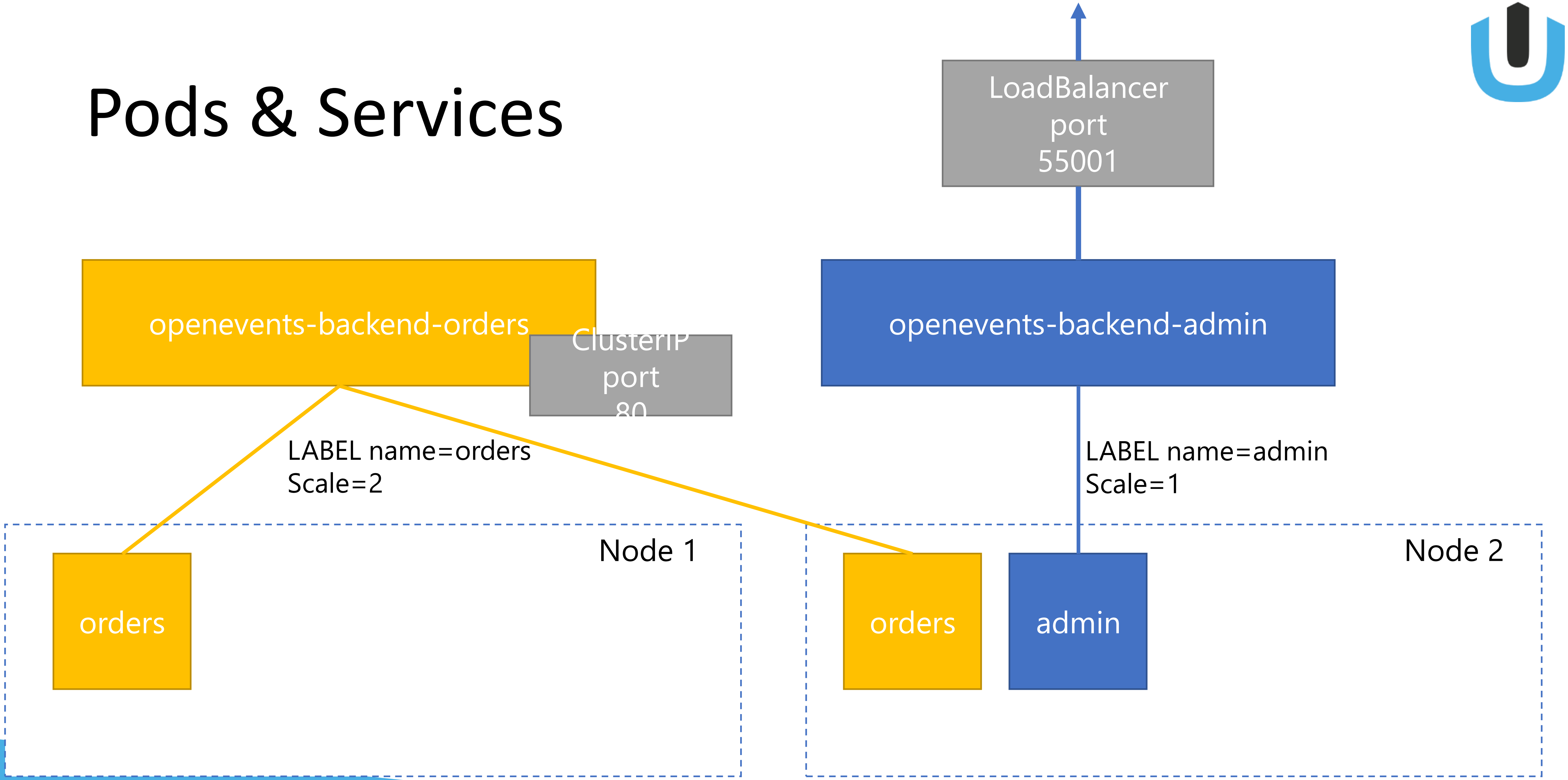
- Stand-alone independent unit
- Has its own IP address
- Commonly Pod = On container
  - But there can be more of them in the pod
    - E.g. application + its proxy server
- Ephemeral
  - Can be killed, restarted or moved at any time
  - Changes made to the file system are lost (unless you allocate volume)
- Labels
  - Key-value pairs with metadata

Service

- „Proxy“ for a group of pods of the same type
- Selector
  - Labels which help to match the pods

- Service exposes its name in DNS to all the pods
  - Load balancing of the traffic between assigned pods

Pods & Services



Management of Kubernetes clusters

- Azure Portal
  - Basic configuration, logs

- CLI
  - kubectl

- Kubernetes portal
  - `az aks browse …`

DEMO

Kubernetes in Azure

Summary

- Azure offers many options for hosting web apps
- Be careful about using local filesystem
  - Blob storage may be a better choice
- Use Storage Queues or Azure Service Bus to notify other apps
- Use Web Jobs or Azure Functions for serverless approach
- You can use containers to run isolated workloads
- You can use Kubernetes to orchestrate container-based solutions

riganti

Update Conference
Prague 2019

DOTVVM

Microsoft®
MVP
Most Valuable
Professional

Q&A

Tomáš Herceg

CEO @ RIGANTI

Co-founder of Update Conference

Microsoft MVP

tomas.herceg@riganti.cz

@hercegtomas