



Miroslav Holec

Software & Cloud Architect

Microsoft MVP: Microsoft Azure

MCSD, MCSA, MTA

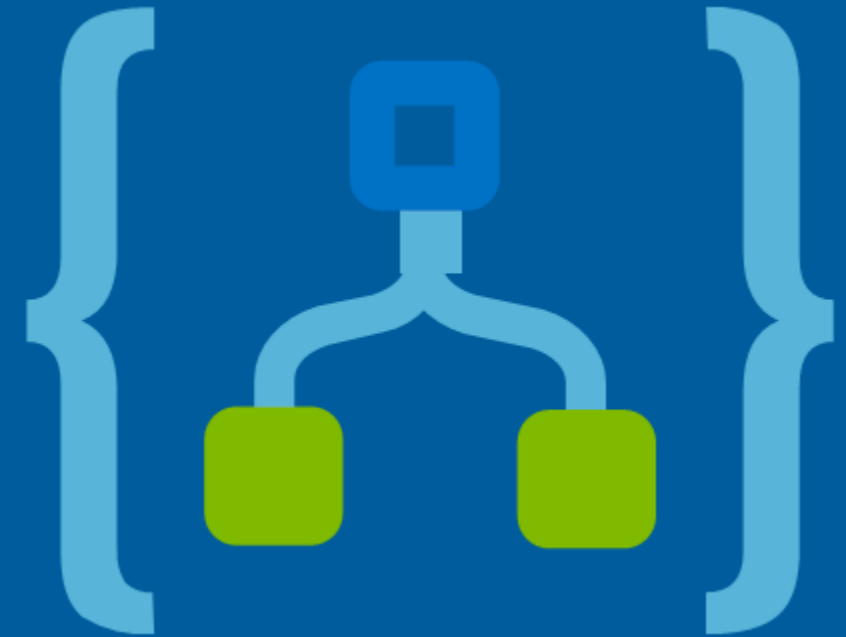
miroslavholec.cz



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

Serverless computing

Aplikace bez serveru s Microsoft Azure





odkaz.me/sc

Serverless architektura



Aplikační systémy nepoužívají standardní způsob vyřízení požadavku ale místo toho jsou **závislé na služnách třetích stran**, které řeší server-side logiku. Logika může být napsána i jako **kód**, který je iniciován událostmi a spouštěn v **bezstavových** kontejnerech pod správou třetí strany .

Specifika serverless



Abstrakce serverů

Kompletně pod správou 3rd party, neřešíme ani nastavení ani škálování (dynamické)



Event Driven

Flow nebo kód zahajují události vznikající v různých službách.



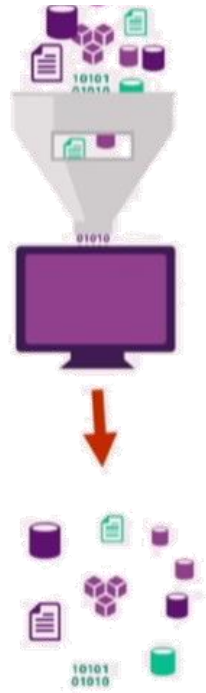
Pay for use

Platí se za čas na procesoru a/nebo za užitou operační paměť.

Serverless architektura

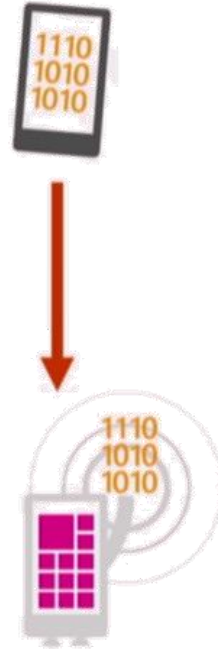


Monolitické aplikace



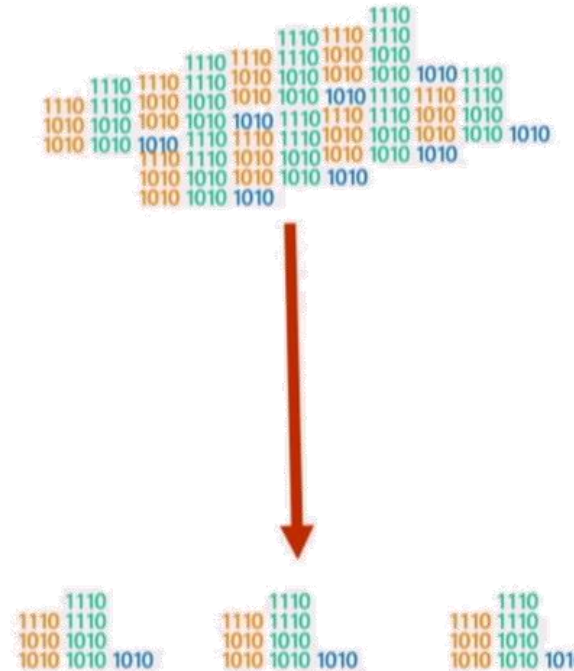
Samostatné komponenty

Logika uvnitř aplikace



Logika mimo aplikaci

Komplexní framework



Mikro funkcionalita

Výhody a nevýhody serverless řešení



Výhody

- Správa nanoslužby (prakticky žádná)
- Škálování (dynamické)
- Náklady (pay for use)
- Náklady na vývoj
- Business oriented (často bez kódu)
- Language independent (C# vs Node.js)



Nevýhody

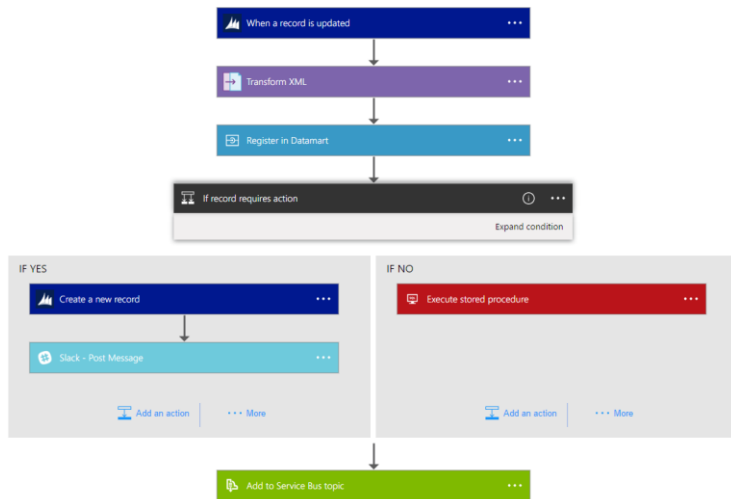
- Správa celého řešení
- Monitoring a testování
- Debugging
- Deployment a verzování
- Bezstavovost (pro někoho)
- Redundance v případě kódu

Nezahazujte všechnen současný kód. Serverless computing je jen alternativní přístup k řešení některých typických business problémů.

Serverless s Microsoft Azure



{ } Flow řízené událostmi



Microsoft Flow (iPaaS)

Logic Apps (iPaaS)

< > Funkce řízené událostmi



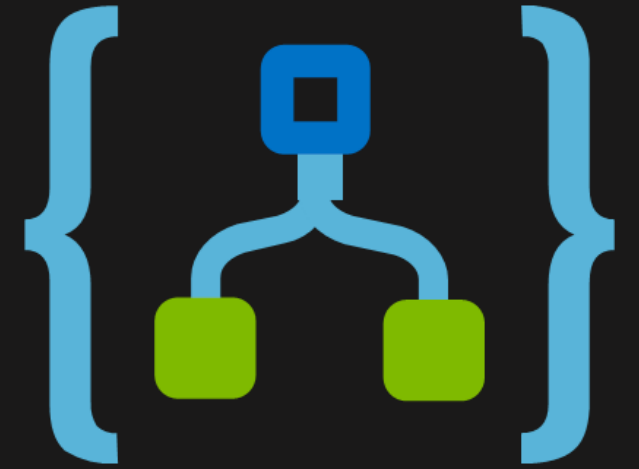
Inspired by Matthew Henderson @Microsoft

Azure Functions (FaaS)

App Service WebJobs (PaaS)

Logic Apps a Microsoft Flow

- Vizuální **klikátko** pro **modelování** business procesů
- **Integrace** s víc než stovkou služeb (Dropbox, Google Apps, O365..)
- Automaticky řešená **autentizace** proti službám (OAuth)
- Definují se události v 3rd party službách, nad kterými se spustí **flow**
- Absolutně **bez správy**, dynamické **škálování**, **pay per execute**



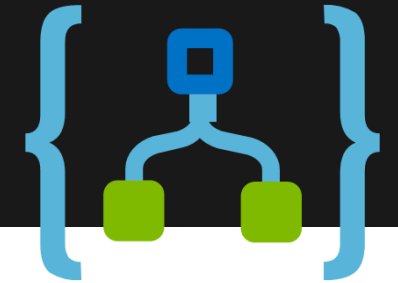
Microsoft Flow

- Není potřeba Azure subscription, zaměřena pro nevyvojáře
- Kompletně založena na Logic Apps
- Chybí některé connectory (např.: Azure Functions)

Verze zdarma

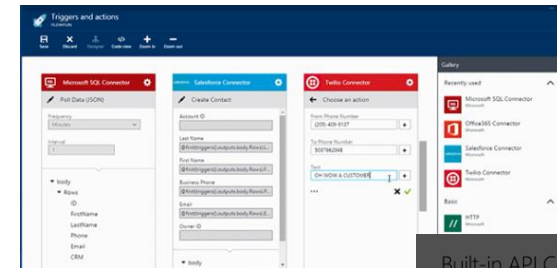
- 750 spuštění (= 1 ex. / h / m)
- 15-minutový check intv.

Logic Apps koncepty



Workflow

- GUI pro modelování business procesů jako série kroků



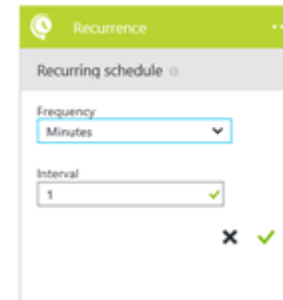
Managed Connectors

- Jednoduché připojení k 100+ službám včetně ověření



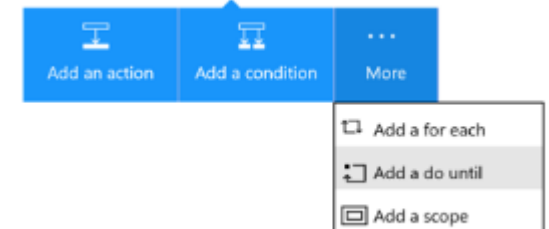
Triggers

- Řada konektorů vystupuje jako aktivátor workflow

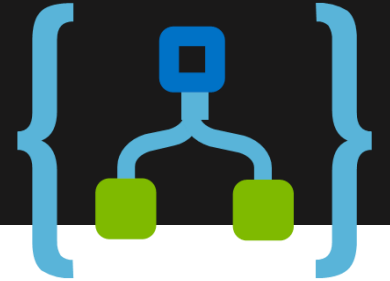


Actions

- Každý krok se jmenuje akce a lze ji mapovat na operátor (or, else, for)



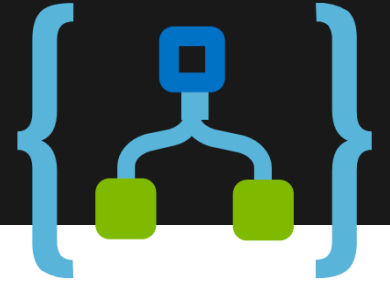
Logic Apps konektory



SaaS

- appFigures
- Asana
- Azure API Management
- Azure App Services
- Azure Automation
- Azure Cognitive Face API
- Azure Cognitive LUIS
- Azure Cognitive Text Analytics
- Azure Data Lake Store
- Azure Document DB
- Azure Functions
- Azure Machine Learning
- Azure Resource Manager
- Azure Service Bus
- Azure SQL
- Azure Storage Blob
- Azure Storage Queues
- Basecamp
- Bing Search
- BitBucket
- Bitly
- Blogger
- Box
- Buffer
- Campfire
- Chatter
- Common Data Service
- Disqus
- DocuSign
- Dropbox
- Dynamics AX Online
- Dynamics CRM Online
- Dynamics CRM Service Bus
- Dynamics Financials
- Dynamics Operations
- Easy Redmine
- Eventbrite
- Facebook
- Freshdesk
- GitHub
- Gmail
- Google Calendar
- Google Contacts
- Google Drive
- Google Sheets
- Google Tasks
- GoTo Meeting
- GoTo Training
- GoTo Webinar
- Harvest
- HelloSign
- Infusionsoft
- JIRA
- Insightly
- Instagram
- Instapaper
- MailChimp
- Mandrill
- Medium
- Microsoft Project Online
- Microsoft Translator
- MSN Weather
- Office 365
- Office 365 Users
- Office 365 Video
- OneDrive
- OneDrive for Business
- OneNote
- Outlook.com
- Outlook Tasks
- PagerDuty
- Pinterest
- Pipedrive
- Power BI
- Project Online
- Redmine
- Salesforce
- Salesforce Chatter
- SendGrid
- SharePoint Online
- Slack
- SmartSheet
- SparkPost
- Stripe
- Survey Monkey
- Todoist
- Trello
- Twilio
- Twitter
- Typeform
- UserVoice
- VS Team Services
- Wordpress
- Wunderlist
- Yammer
- YouTube

Logic Apps konektory



Protocols / Native

- HTTP, HTTPS
- HTTP Webhook
- FTP, SFTP
- SMTP
- RSS
- Compose, Query, Parse JSON
- Wait
- Terminate
- Workflow

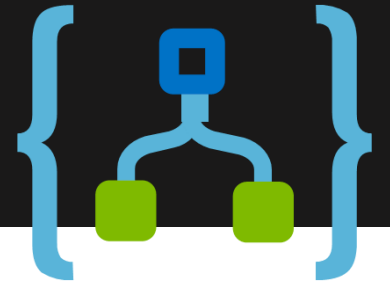
Hybrid

- BizTalk Server
- File System
- IBM DB2
- Informix
- SharePoint Server
- SQL Server
- SAP
- Websphere MQ

Azure Services and Power Apps Connectors

- API / WebApp
- Azure DocumentDB
- Azure ML
- Azure Blob Storage
- Resource Manager
- Azure Queues
- Cognitive Svc
- Common Data Service
- Event Hubs
- Nested LogicApp
- Power BI
- Service Bus
- SQL Azure

Logic Apps Monitoring / Logs



Azure Portal Monitoring

- Přehled všech spuštění workflow
- Monitoring view: pohled na jednotlivé kroky

Azure Diagnostics and Alerts

- Detailní pohled na chyby
- Možnost zaslat upozornění, pokud například selže workflow

Summary

All runs

STATUS	START TIME	DURAT
✓ Succeeded	7/22/2016, 7:48 PM	248 M
✓ Succeeded	7/22/2016, 7:48 PM	248 M
✓ Succeeded	7/22/2016, 7:48 PM	294 M
✓ Succeeded	7/22/2016, 7:48 PM	275 M
✓ Succeeded	7/22/2016, 7:48 PM	249 M
✓ Succeeded	7/22/2016, 7:48 PM	275 M

Workflow steps:

- When a new tweet appears (14s)
- Detect Tweet Sentiment (0s)
- Insert Row Into SQL (0s)
- List records (0s) - Error: ActionConditionFailed. Message: The execution of template action 'List_records' is skipped: the 'runAfter' condition for action 'Insert_Row_Into_SQL' is not satisfied. Expected status values 'Succeeded' and actual value 'Failed'.
- For Each Record Returned By CRM
 - Create a new record (0s)

Filter: Runs Failed

Period: Over the last 5 minutes

DEMO />

Propsání úkolu z Wunderlistu do Visual Studio TS

Jsem v dopravě a napadl mě úkol k vyřešení. Zapišu ho přes mobil do Wunderlistu do zvoleného seznamu a chci, aby se mi propsal do VSTS.

Výzvy pro non-serverless řešení

- Periodické spouštění služby
- Napojení na Wunderlist API
- Napojení na Visual Studio Team Services API
- Nalezení nové položky a propsání do VSTS
- Logování nebo monitoring

DEMO />

Odeslání SMS při nedostupnosti webové aplikace

Služba Application Insights mi kontroluje dostupnost webu. Pokud není dostupný, posílá se email. Spolu s emailem bych chtěl dostat i SMS zprávu.

Výzvy pro non-serverless řešení

- Endpoint pro zachycení AI webhooku
- Aplikační logika pro přečtení dat z těla
- Napojení na SMS poskytovatele a odeslání zprávy
- Logování nebo monitoring

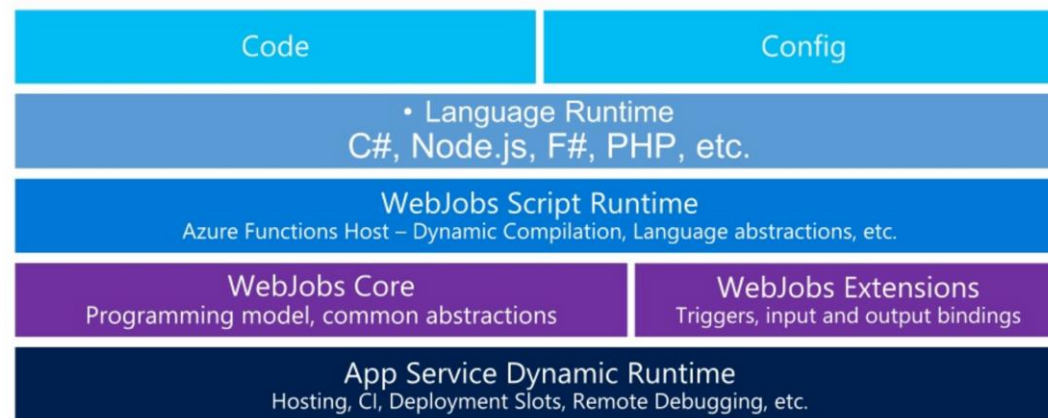
Azure Functions

- In-Browser **Azure WebJobs** (postaveno nad WebJobs SDK)
- Definují se události v Azure službách (**triggery**), nad kterými se spustí **kód**
- Kód může být v **různých jazycích** (C#, Node.JS, PHP..) – podpora **NuGet** a **npm**
- **Integrace** s 3rd party službami (nastavení v GUI místo kódu)
- Funkce má „injectnuté“ **vstupy a výstupy** – obojího může být více



- Dva **režimy placení**: app service plan vs consumption plan
- Podpora ve Visual Studio
- K Azure Functions obvykle přistoupíme tam, kde nám nestačí Logic Apps

<https://functions.azure.com>



Azure Functions koncepty



run.csx

Save

Run

```
1 using System.Net;
2 using Dapper;
3 using System.Data.SqlClient;
4 using System.Configuration;
5
6 public static async Task<HttpResponseMessage> Run(HttpRequestMessage req, TraceWriter log)
7 {
8     log.Info("C# HTTP trigger function processed a request.");
9
10    // Parse data from request
11    ClientMail data = await req.Content.ReadAsAsync<ClientMail>();
12    data.Date = DateTime.UtcNow;
13
14    var connectionString = ConfigurationManager.ConnectionStrings["SqlConnection"].ConnectionString;
15    try {
16        using(var connection = new SqlConnection(connectionString))
17        {
18            connection.Open();
19
20            connection.Execute("INSERT INTO [dbo].[ClientMail] (Date, Subject, [From], Body) VALUES
21                log.Info("New mail added succesfully!");
22        }
23    }
24    catch(Exception ex)
25    {
26        log.Info("Error: " + ex.Message);
27    }
28
29    return req.CreateResponse(HttpStatusCode.OK, "OK");
30 }
```

OUTPUT

TRIGGER

INPUT

CODE

OUTPUT

Azure Functions trigger a binding



Typ	Služba	Trigger	Input	Output
Schedule	Azure Functions	A		
HTTP (REST)	Azure Functions	A		A
Blob Storage	Azure Storage	A	A	A
Events	Azure Event Hub	A		A
Queues	Azure Storage	A		A
Tables	Azure Storage		A	A
Tables	Azure Mobile Apps		A	A
No-SQL DB	Azure DocumentDB		A	A
Push Notifications	Azure Notification Hub			A
Twilio SMS Text	Twilio			A
SendGrid email	SendGrid			A

Azure Functions správa kódu



Nástroje pro správu kódu

- Azure Functions Portal
- Visual Studio + deployment
- App Service Designer
- Kudu (Debug – CMD)
- FTP

Balíčkování, assemblies a kód

- Vlastní **metody a třídy** lze vytvářet přímo v .csx
- **Sdílený kód** v .csx lze načítat pomocí #load "file.csx"
- Import **namespaces** přes using, např.: System.Xml
- **Externí assembly** importujeme přes #r "Newtonsoft.Json"
- 3rd party (**NuGet**) se definují v project.json
- **Privátní balíčky** nahrajeme do bin #r "path/assembly.dll"

Azure Functions + Application Insights



- V současné době podpora pro C# a Node.JS
- <https://github.com/christopheranderson/azure-functions-app-insights-sample>

Postup

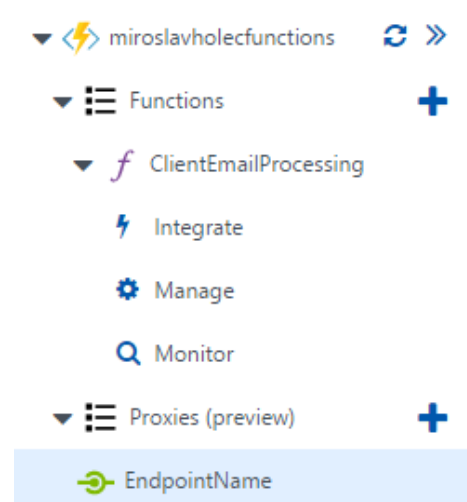
- Založím Application Insights
- Nastavím instrumentační klíč v Azure App Service (Settings)
`APPINSIGHTS_INSTRUMENTATIONKEY`
- V rootu aplikace musí být project.json (C#) s referencí na AI
- V rootu aplikace musí být package.json (Node.JS) s referencí na AI



Azure Functions proxies



- Umožňuje specifikovat **endpointy, které jsou odbaveny jiným resourcem**
- Na jiný resource (API) je odkázáno přes „**backend URL**“
- Lze definovat **povolené HTTP** metody a **routovací šablonu**
- Standardně nejsou povolené (nastavení ve Function App Settings)
- Aktuálně je tato funkce v **preview** módu



Proxy URL

`https://miroslavholecfunctions.azurewebsites.net/ClientEmailProcessingProxy`

Route template

`/ClientEmailProcessingProxy`

Allowed HTTP methods

Selected methods

GET POST DELETE HEAD

PATCH PUT OPTIONS TRACE

Backend URL

`https://miroslavholecfunctions.azurewebsites.net/ClientEmailProcessing`

Save Discard

Azure Functions cenové plány



Consumption plan

- Platba za výpočetní čas a operační paměť (jen pokud funkce běží)
- Automatické škálování, není potřeba přemýšlet nad výkonem strojů
- Nepodporuje App Service Extensions

Př.: Funkce spotřebuje 1536 MB a je spuštěna 2 mil. x za měsíc. Běh trvá průměrně 1 sec.

EXECUTIONS

=> 2 mil sec. běhu * (1536/1024) = 3,0 mil. GB-s

=> Zdarma od MS je 0,4 mil. GB-s

=> K platbě 2,6 mil. GB-s * \$0,000016 = \$41,60

RESOURCE

=> 2 mil. ex – 1 mil. zdarma od MS = 1 mil. spuštění

=> 1 mil. * \$0,20 = \$0,20

App Service Plan

- Běží nad App Service planem a platí se dle vybrané VM
- Je nutné hlídat si výkonnost stroje, nastavit případně autoscale
- App Service doporučeno nastavit na AlwaysOn

DEMO />

Uložení emailové zprávy od klienta do DB

Chci si mimo gmail propisovat všechny přijaté emaily od klientů do svého systému, abych je mohl poté snadno dohledat.

Výzvy pro non-serverless řešení

- Služba kontrolující periodicky stav emailu
- Napojení se na Google Mail API
- Připojení k relační databázi a načtení klienta
- Uložení emailu k existujícímu klientovi

Azure Functions doporučení



- Nepsat zbytečně to, co umí Logic Apps
- **Minimalistická** co do množství kódu
- Vyhýbat se redundanci kódu, využít **shared .csx**
- Pro sloužitější logiku volit **NuGet balíčky** nebo **private binárku**
- Funkce dělá jen **jednu věc**
- Ideálně **asynchronní** (hlavně u front) – NE Task.Result (wait na jiný thread, deadlocky)
- Nekombinovat produkční a testovací kód, využít **proxies**
- Funkce by měla být **bezstavová**, pro komunikaci používat Queues
- Vyřešení problému netrvá dlouho (**krátkotrvající**)
- Nezapomenout na CORS :)



Miroslav Holec

Software & Cloud Architect

Microsoft MVP: Microsoft Azure

MCSD, MCSA

odkaz.me/sc

Takeaways

- Serverless architektura používá 3rd party services
- Vytváříme funkce nebo flow řízené událostmi
- Staráme se jen o nanoslužbu, platíme za spotř. resources
- V Azure jsou Logic Apps a Azure Functions (propojitelné)
- Logic Apps mají variantu Microsoft Flow pro nevyvojáře
- Aktuálně vzniká mocný tooling kolem služeb

Q&A