Serverless

in Microsoft Azure

Who we are

Robert Meyer

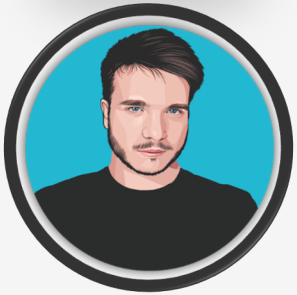
Fullstack Software Developer @ Cloudklabauter
Freelance Trainer

@roeb

Martin Brandl

Cloud Solution Architect @ white duck
Microsoft Azure MVP

@martin_jib

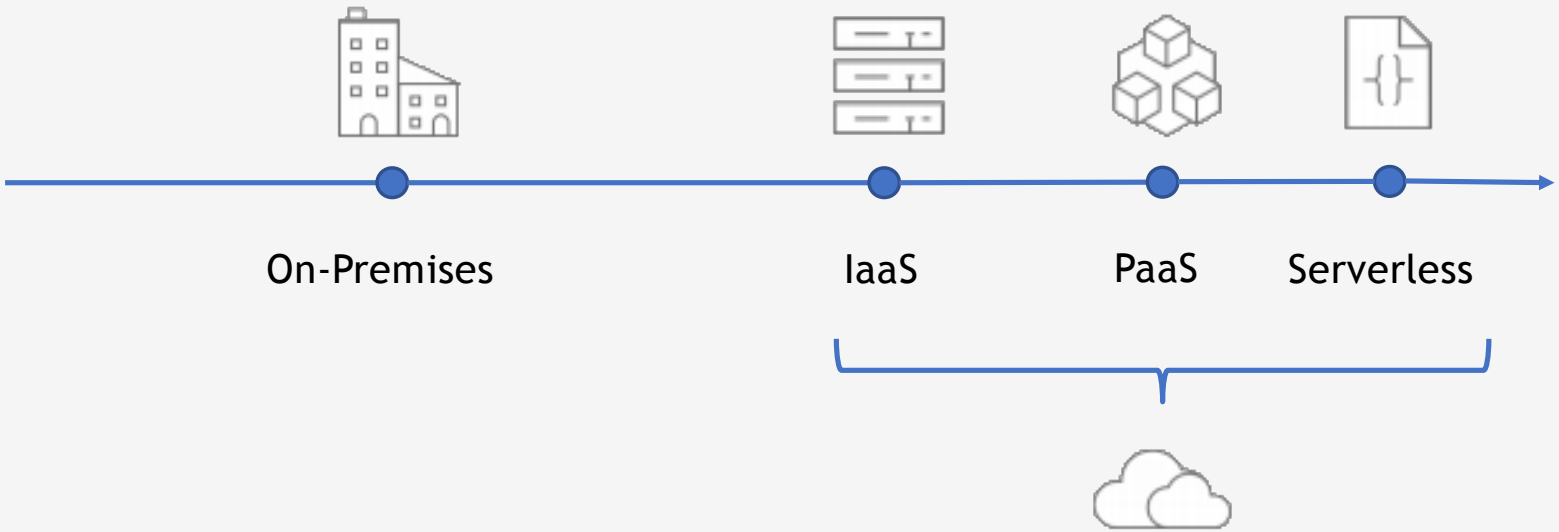


Suad Wolgram

Werkstudent @ white duck

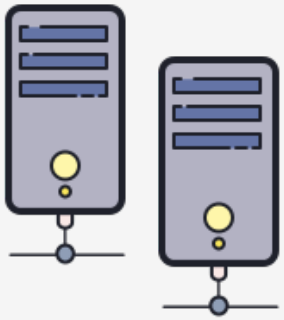
@SuadWolgram

Microsofts Serverless World



What is Serverless?

Deploy without having to worry about infrastructure



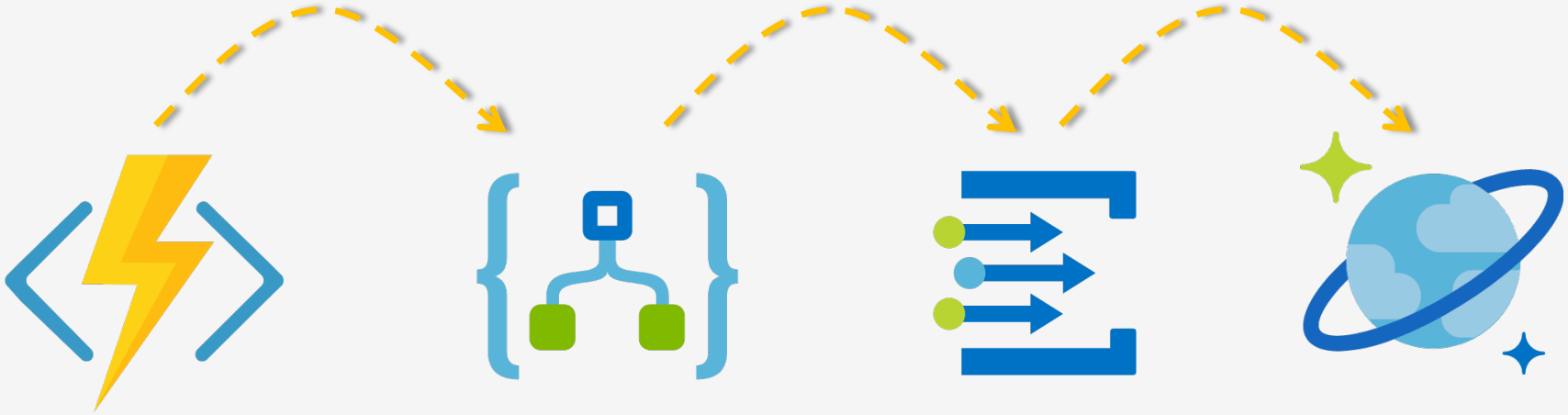
Automatic scaling

Consumption-based pricing model

Functions

in Microsoft Azure

Serverless Components in Azure



Functions

Serverless Compute

Logic Apps

Serverless Workflows

Event Grid

Serverless Events

CosmosDB

Serverless Documents Database

What is Azure Functions?

Azure Functions are serverless, event driven and extends the existing Azure App Service platform

Azure Functions are „nanoservices“ that can be scale based on demand

Azure Functions: Triggers and Bindings

Triggers are a way to start the execution

Bindings are a way to simplify code
for input and output of data

Azure Functions: Triggers and Bindings

Triggers

Timer
Queue Message
HTTP Trigger
…

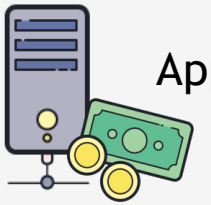
Languages

C#, F#
JavaScript, TypeScript
Java
PowerShell
Python

Bindings

Azure Storage (Queue/Blob)
CosmosDB
SendGrid
HTTP
SignalR
ServiceBus
… und 12 weitere

AppService vs. Consumption Plan



AppService Plan

Regular App Service Plan
- Pay for dedicated servers
- Predictable monthly cost
- Several pricing tiers
- No function duration constraints

Azure Functions premium plan
- VNet connectivity
- Improved performance
- Pre-warmed instances
- Set min and max instances

Consumption Plan

Billing Model
- Number of executions
- CPU Time (s) x RAM (GB)

Free monthly grant
- 1,000,000 executions
- 400,000 GBs

Infrastructure
- A1 Virtual Machine

Azure Functions: Scaling

	App Service	Premium	Consumption
Timeout Duration	30min – unlimited	30-60min	5-10min
Scale-Out	Manual/Autoscale	Event-Driven	Event-Driven
Max Instances	10-20	100	200
Function per AppPlan	Unbounded	100	100
Max Memory	1.75 – 14GB	3.5 – 14GB	1.5GB
Storage	50 – 1000GB	250GB	1GB

Azure Functions: Security

Authorization Key
- Function Keys
- Host Keys
- Master Key

Communication with other Azure resources through Managed Identity

Use Azure KeyVault references for sensitive data

Use API Management to control the Function calls

OpenID Connect Provider (Preview)

Azure Functions: Core Tools & Azurite

Azure Functions Core Tools

- CLI for local development, testing & deployment
- Allows Remote Debuging from Azure Functions
- Available for Windows, macOS and Linux
- Command: *func*

Azurite

- Azure Functions requires an Azure Storage account
- Azurite is a open-source Storage Emulator
- Azurite supersedes the Azure Storage Emulator
- Can use a Docker Container or with VSCode Extensions

Demo: Azure Functions Demo

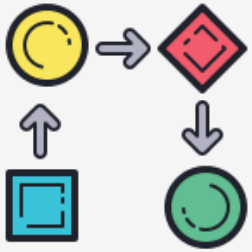
Create a JavaScript Function inside the Azure Portal

Create and deploy an Azure Function with Visual Studio Code

Azure Durable Functions

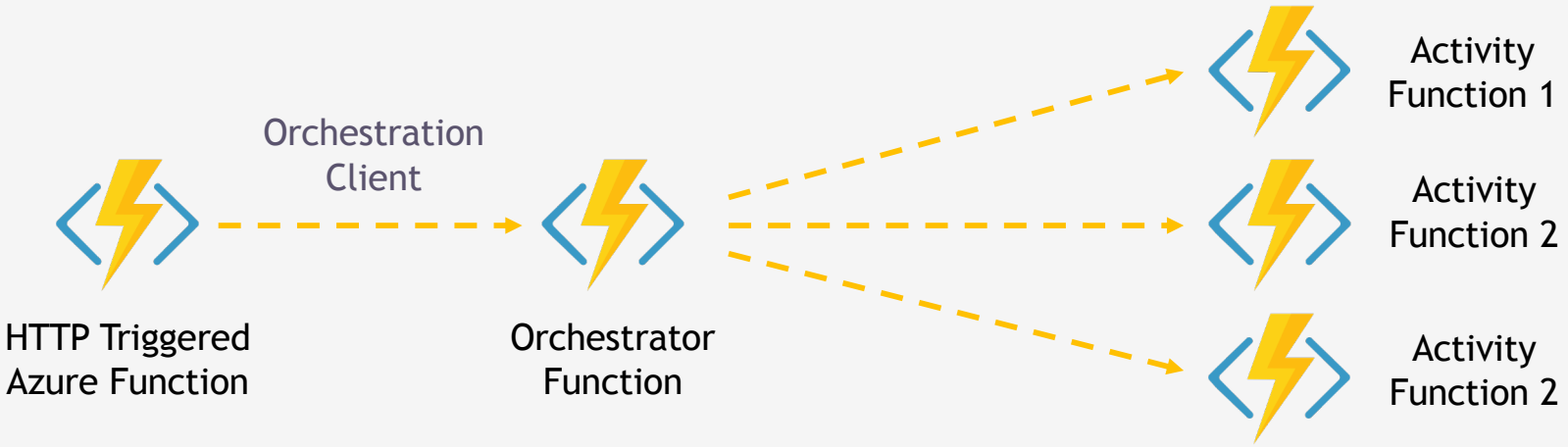
An extension to Azure Functions

Write “stateful” functions in a “serverless” environment



Define workflows in code

How Durable Functions work



Azure API Connections

Status: Private Preview

Takes Azure Logic App Connectors to Azure (Durable) Functions

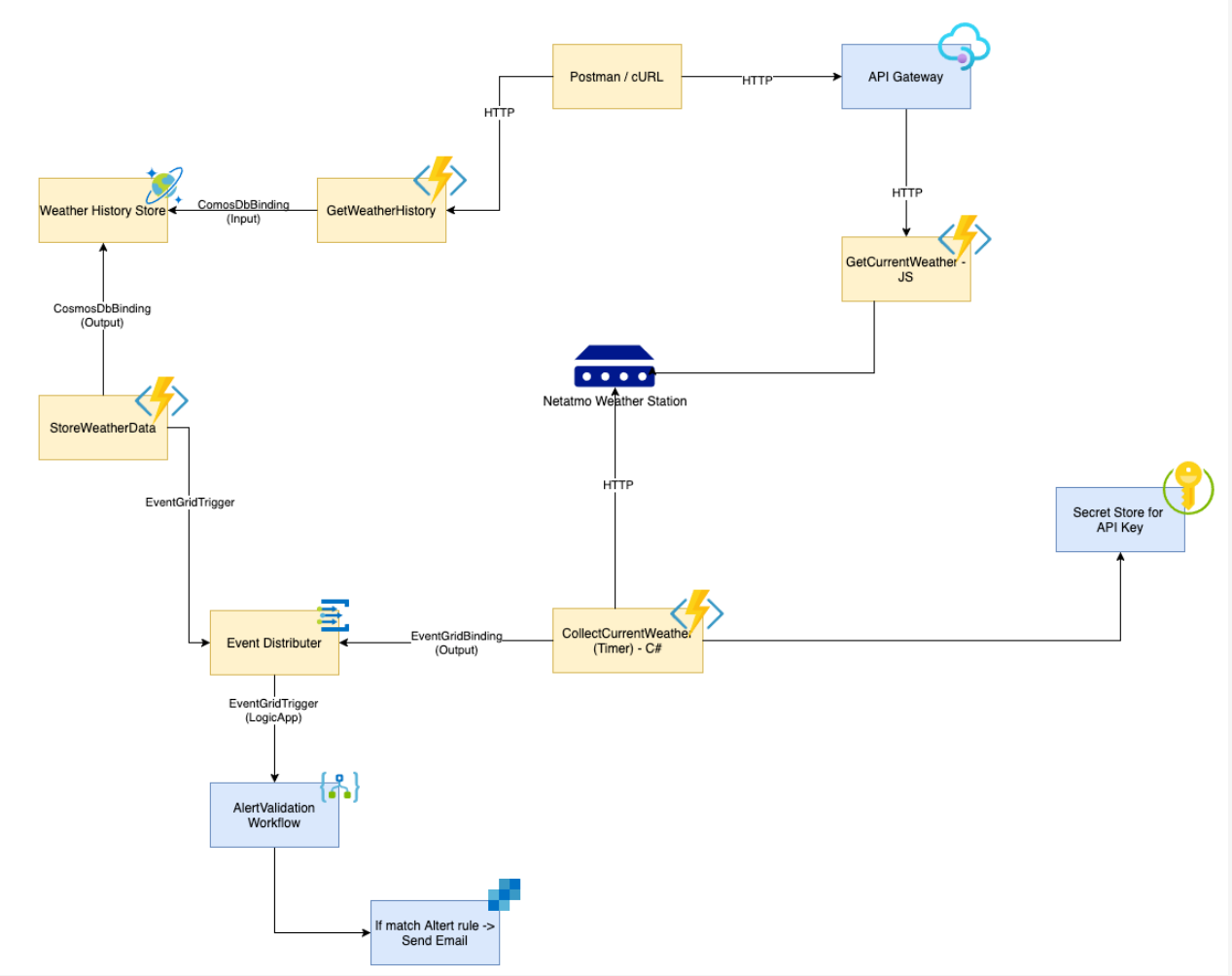
Currently only Microsoft Connectors

Only for C# & JavaScript

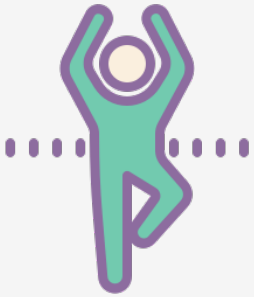
Benefits:

- Reduce Development coasts
- Easy & Secure Authentication over OAuth

The Scenario



Exercise: Azure Functions



Create and deploy your first Azure Function

We have a Netatmo weather station (Emulator) for you. The weather station generates new measurements every minute.

The measured values are provided by Netamto via REST API.

You should write a function which can be called via HTTP and retrieves the latest measured data from the Netatmo REST API Endpoint and makes it available to the portal.

Link: [GITHUB LINK HERE]