Miroslav Holec
Software & Cloud Architect
Microsoft MVP: Microsoft Azure
MCSD, MCSA, MTA

miroslavholec.cz
@miroslavholec

Serverless in Azure

HAVIT SOFTWARE DEVELOPMENT

bit.ly/NSWI152

Serverless Computing

- abstraction of servers and infrastructure
- no servers provisioning, focus on the code
- driven by real-time reactions to events and triggers
- billing based on consumed resources (code running)

Benefit from a fully managed service

Spare your teams the burden of managing servers. By utilizing fully managed services, you focus on your business logic and avoid administrative tasks. With serverless architecture you simply deploy your code, and it runs with high availability.

Scale flexibly

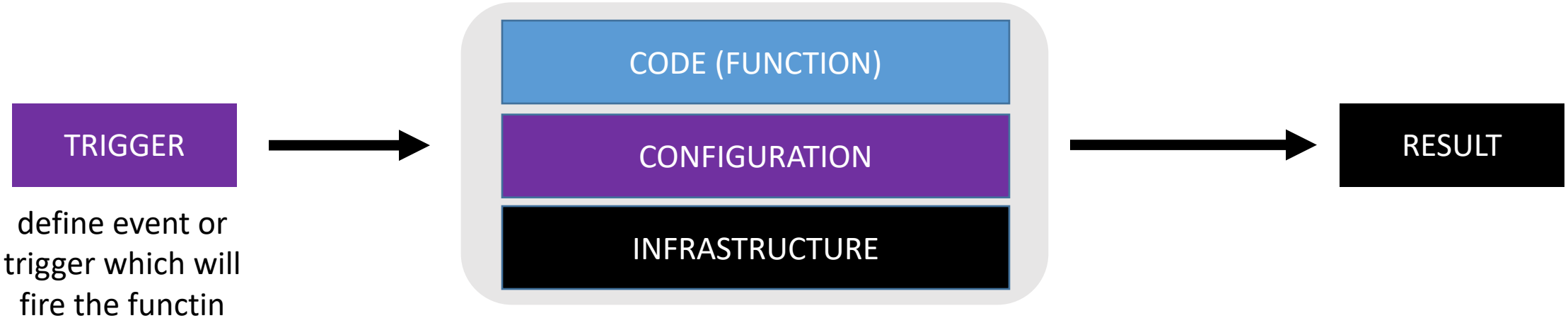
Serverless compute scales from nothing to handle tens of thousands of concurrent functions almost instantly (within seconds), to match any workload, and without requiring scale configuration—it reacts to events and triggers in near-real time.

Only pay for resources you use

With serverless architecture, you only pay for the time your code is running. Serverless computing is event-driven, and resources are allocated as soon as they're triggered by an event. You're only charged for the time and resources it takes to execute your code—through sub-second billing.

Azure Functions

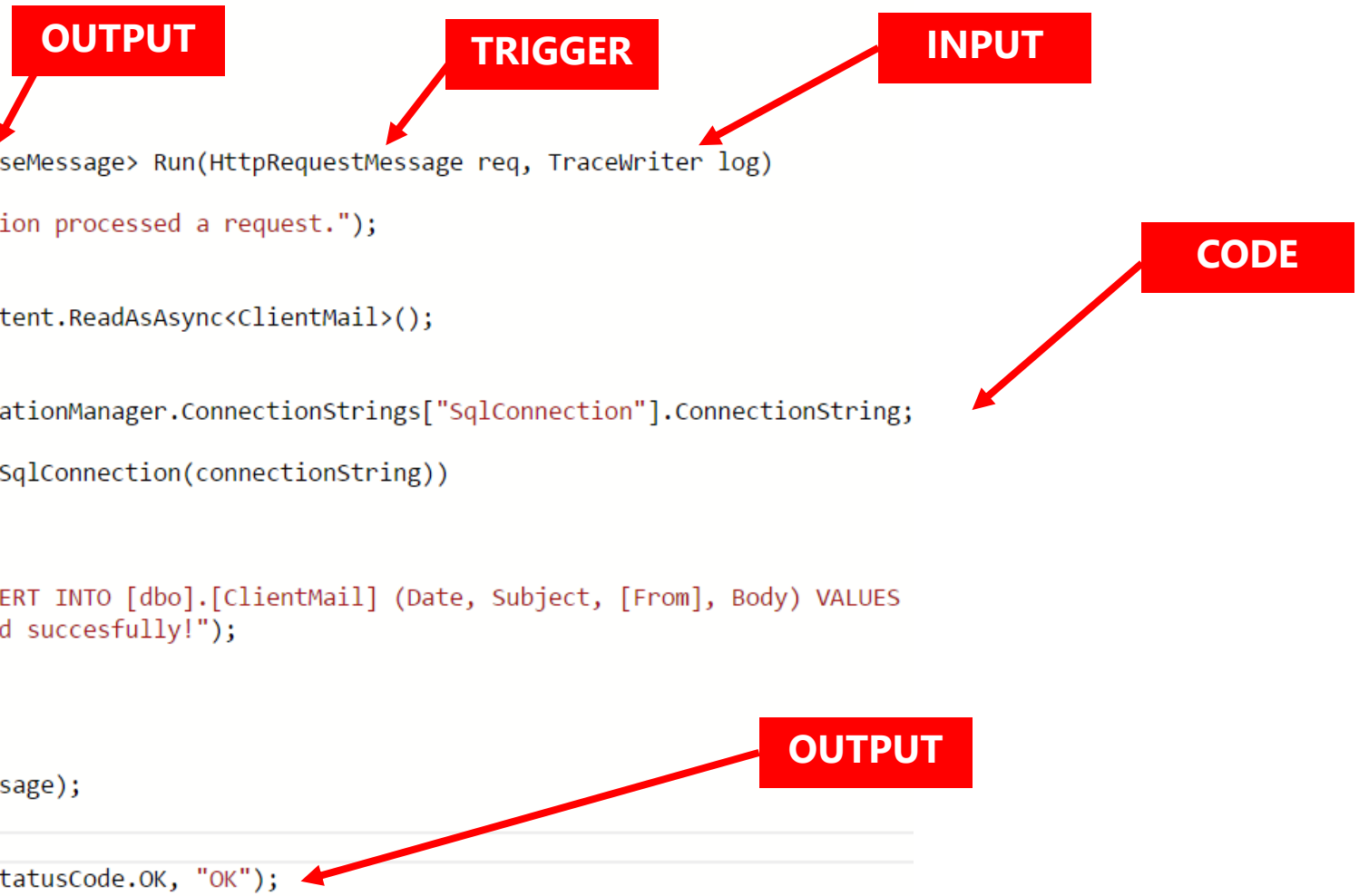
- running small piece of code in cloud (function)
- language liberty (C#, Java, F#, PHP, Node.js)
- dependencies support (NuGet, NPM...)
- pay-per-use (consumption plan) or “standard” app service plan
- technically runs on app service and webjobs SDK



Azure Functions

run.csx Save ▶ Run

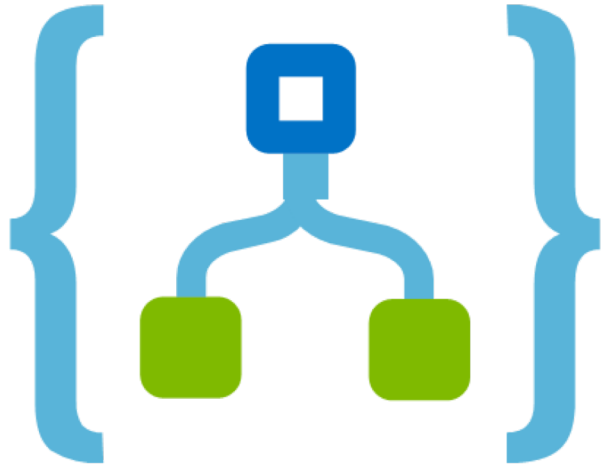
```
1  using System.Net;
2  using Dapper;
3  using System.Data.SqlClient;
4  using System.Configuration;
5
6  public static async Task<HttpResponseMessage> Run(HttpRequestMessage req, TraceWriter log)
7  {
8      log.Info("C# HTTP trigger function processed a request.");
9
10     // Parse data from request
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 }
```



Azure Functions

DEMO

Azure Logic Apps



- running business workflows in the cloud
- integrate with various SaaS
- out of the box connectors (reduces implementation time)
- autoscale, pay-per-use model

MANAGED CONNECTORS
(office 365, gmail, dropbox, asana, twitter, facebook)

TRIGGER
(new post, new file, new email message)

ACTIONS

CONDITIONS

WORKFLOW
(steps which will execute)

Logic Apps

DEMO

Lab

- WHEN a new RSS feed item is published (choose your favorite one)
- IF (title OR summary) CONTAINS “Azure“
  - PROCESS RSS item (Azure Storage, Send to mailbox…)

- or create your own workflow