Introduction to Azure Functions

Azure Global Bootcamp 2018

Robert Schlaeger
Developer
schlaeger@medialesson.de

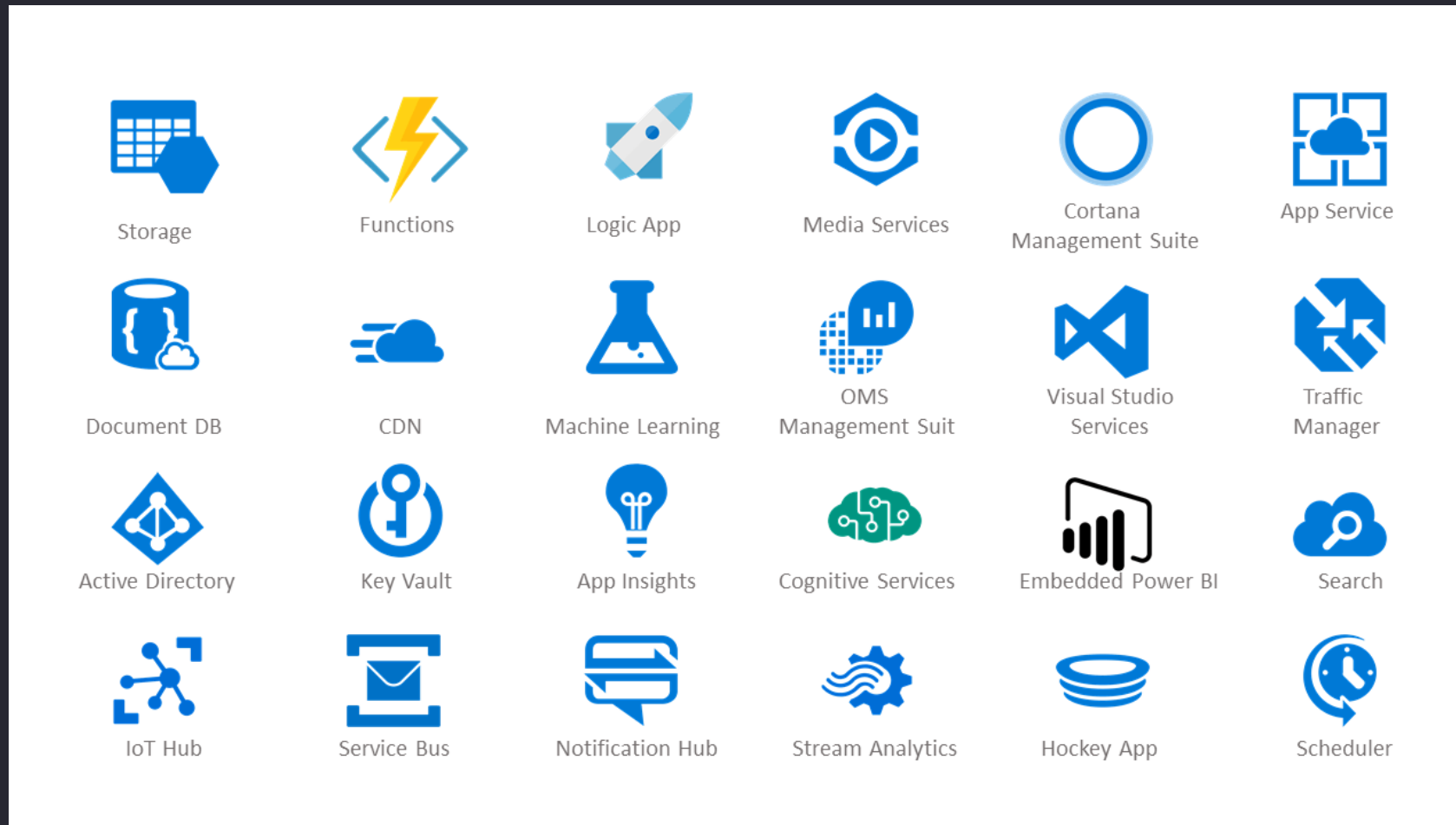🐦 @RobAnybody_

Sebastian Jensen
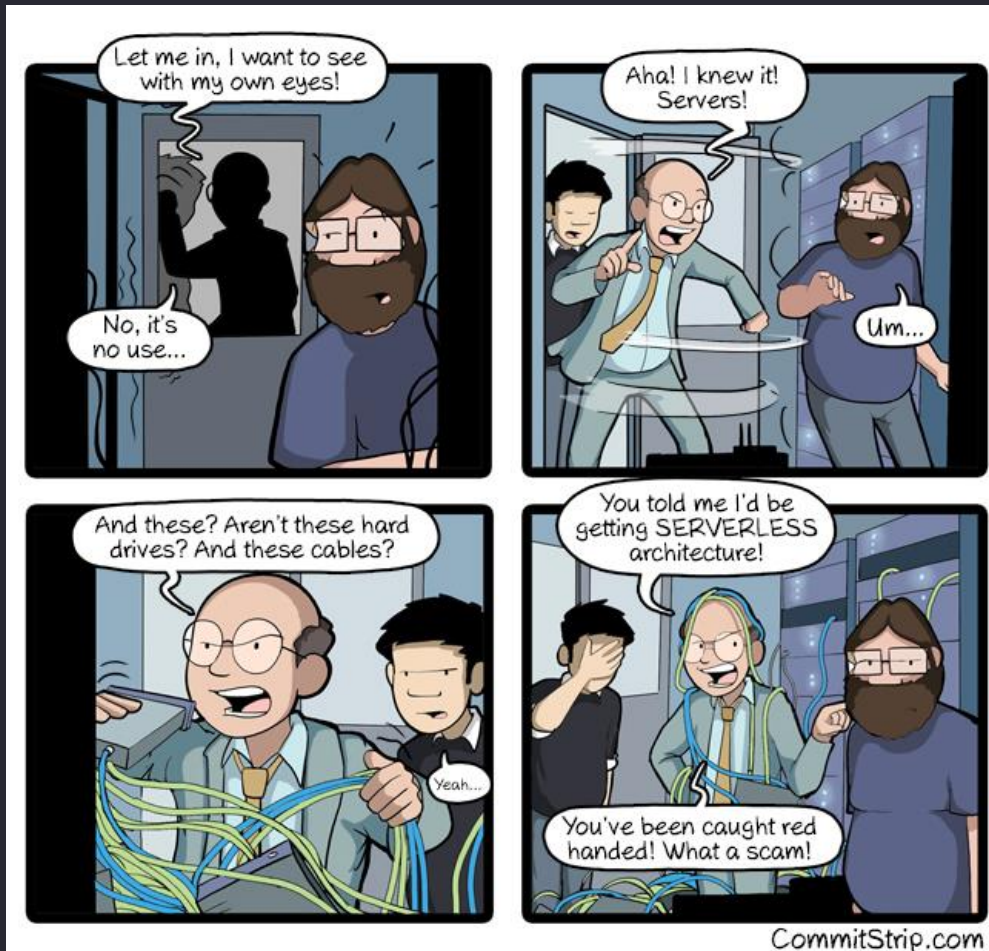Developer
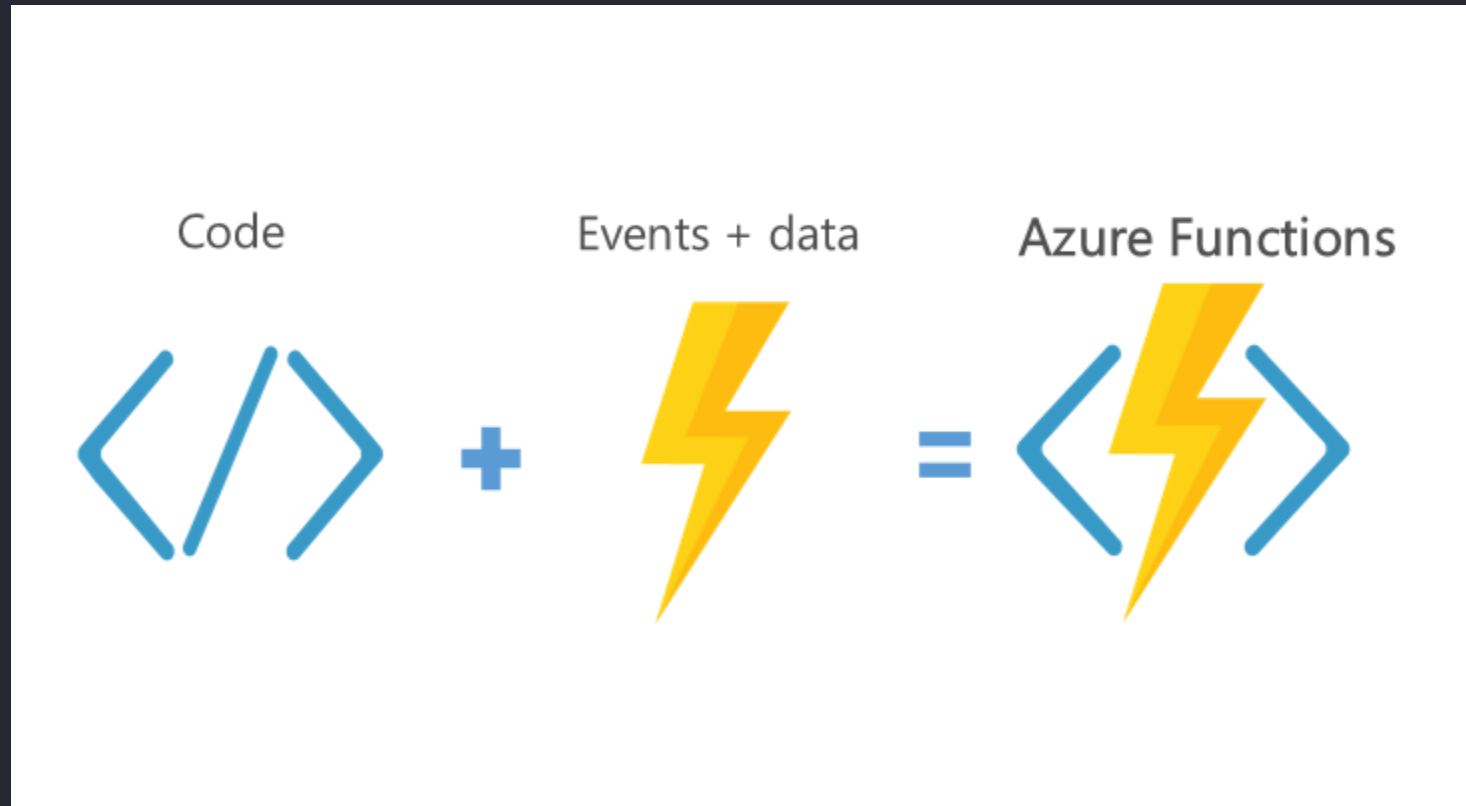jensen@medialesson.de

@tsjdevapps

Important Serverless Products

Serverless?!

What are Azure Functions?

Simplified Programming Model

- Write the code to respond to an event

- No need to write setup code

- Write code for only what you need to do

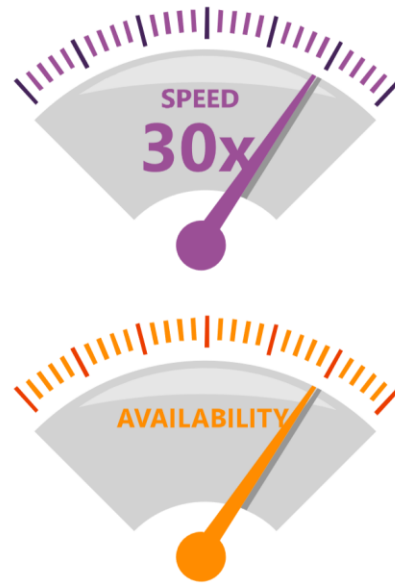- Focus on the business requirement

Language Support

Benefits

Abstraction of servers

Event-driven/ instant scale

Sub-second billing

New Pricing Model

- Pay only for what you use, no more

- If you are not using the functions, you are not paying

- Servers when you need them

- Serverless when you don't

Demo

Hello World – C#

```csharp
using System.Net;

public static async Task<HttpResponseMessage> Run(HttpRequestMessage req, TraceWriter log)
{
    log.Info("C# HTTP trigger function processed a request.");

    // parse query parameter
    string name = req.GetQueryNameValuePairs()
        .FirstOrDefault(q => string.Compare(q.Key, "name", true) == 0).Value;

    if (name == null)
    {
        // Get request body
        dynamic data = await req.Content.ReadAsAsync<object>();
        name = data?.name;
    }

    string response = "Hello " + name + ", welcome to the functions real world!";

    return name == null
        ? req.CreateResponse(HttpStatusCode.BadRequest,
                             "Please pass a name on the query string or in the request body")
        : req.CreateResponse(HttpStatusCode.OK, response);
}
```

Hello World – JS

```javascript
module.exports = function (context, req) {
    context.log('JavaScript HTTP trigger function processed a request.');

    if (req.query.name || (req.body && req.body.name)) {
        context.res = {
            // status: 200, /* Defaults to 200 */
            body: "Hello " + (req.query.name || req.body.name) +
                ", welcome to the functions real world!"
        };
    }
    else {
        context.res = {
            status: 400,
            body: "Please pass a name on the query " +
                "string or in the request body"
        };
    }
    context.done();
};
```

Hello World – F#


```fsharp
#r "System.Net.Http"
#r "Newtonsoft.Json"

open System.Net
open System.Net.Http
open Newtonsoft.Json

type Named = {
    name: string
}

let Run(req: HttpRequestMessage, log: TraceWriter) =
    async {
        log.Info(sprintf
            "F# HTTP trigger function processed a request.")

        // Set name to query string
        let name =
            req.GetQueryNameValuePairs()
            |> Seq.tryFind (fun q -> q.Key = "name")

        match name with
        | Some x ->
            return req.CreateResponse(HttpStatusCode.OK, "Hello " + x.Value
                                        + ", welcome to the functions real world!");
        | None ->
            let! data = req.Content.ReadAsStringAsync() |> Async.AwaitTask

            if not (String.IsNullOrEmpty(data)) then
                let named = JsonConvert.DeserializeObject<Named>(data)
                return req.CreateResponse(HttpStatusCode.OK, "Hello " + named.name
                                        + ", welcome to the functions real world!");
            else
                return req.CreateResponse(HttpStatusCode.BadRequest, "Specify a Name value");
    } |> Async.RunSynchronously
```

Triggers

- Defines the invocation of the function

- Must have exactly one trigger

- A trigger has some associated data with it

- Contains the payload that triggered the functions

Bindings

- Means of connecting to data from the code

- 2 types of bindings: Input Bindings and Output Bindings

- Bindings are optional

- Can have multiple input and output bindings

Demo

Voice Assistant: Alexa

Alexa

- Amazon

- April 2014

- Echo Family

- Windows-PCs later this year...

Wording: Skill

- Voice Experiences, which can be developed by third parties

- Extends the available functions

- "Voice Apps" with focused functionality

- Are currently available for free

- Be activated or deactivated via voice commands or companion apps

Demo

Pros of Azure Functions

- Azure Functions are cheap.

- Azure Functions are simple for simple scenarios.

- The amount of code you write in a function will probably be less than writing the same behavior outside of Azure Functions.

Cons of Azure Functions

- The languages and the runtimes for Azure Functions are not specialized.

- Deploying, authoring, testing, and executing a function is difficult outside of Azure.

- Startup time of the function is sometimes a little bit slow.

App Entwickler/in

**Ob als Praktikant, Werkstudent,
Trainee oder Direkteinstieg als
Junior App Entwickler/in,
bei uns bist Du genau richtig, wenn …**

**… Du richtig Bock
auf Programmieren hast!**

**Apps für Windows, iOS, Android,
IoT, Cloud, HoloLens, …**

**… mit .NET, C#, Xamarin,
Angular und Unity**

DAS ERWARTET DICH:

- Erweiterung Deiner Developer Skills
- Spannende und abwechslungsreiche Projekte
- Zugang zu neuesten Gadgets und Technologien, wie der Microsoft HoloLens
- Ein starkes Team und tolle Kollegen

* *Und nicht zu vergessen: regelmäßige Teamessen, wöchentliches Teamfrühstück, Kaffee oder Tee in Unmengen, frisches Obst und Rückenmassagen unserer Physiotherapeutin gehören auch mit dazu.*

Hört sich das nach Deinem Traumjob an?
Dann freuen wir uns auf Deine Bewerbung
an Petra Bauknecht **jobs@medialesson.de**

Any questions?

medialesson

Thank you for your interest!

www.medialesson.de

medialesson