

Going Serverless

mit Azure Functions

Tim Niklas Vinkemeier

- Software Developer @ prodot
- M. Sc. Informatik
- Microsoft Student Partner Alumnus



 @TimVinkemeier

 github.com/TimVinkemeier

“less servers need to care about”
“serverless”

Das Cloud-Kontinuum

oder: wo läuft mein Code?



Bare Metal

IaaS

PaaS

FaaS / "Serverless"



- VMs
- Scale Sets



- Web Apps
- Mobile Apps

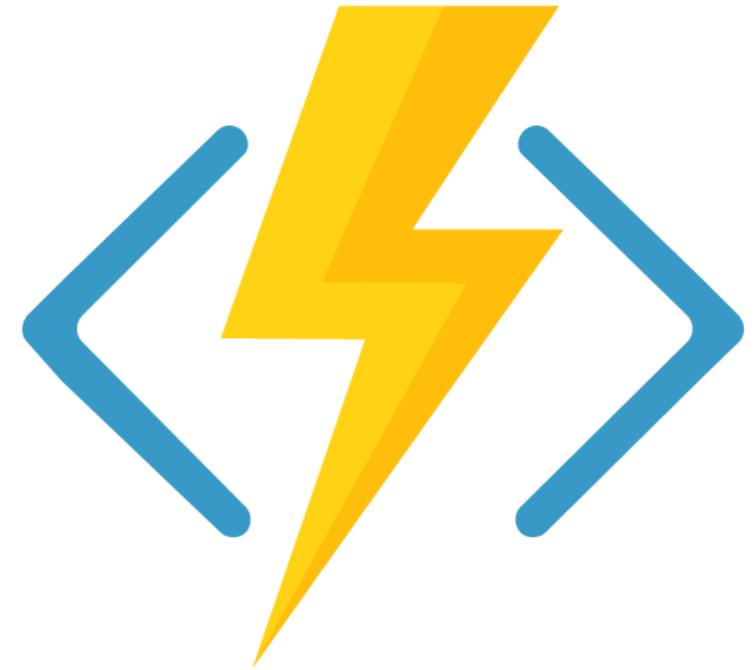


- Functions

Serverless Compute - Vorteile

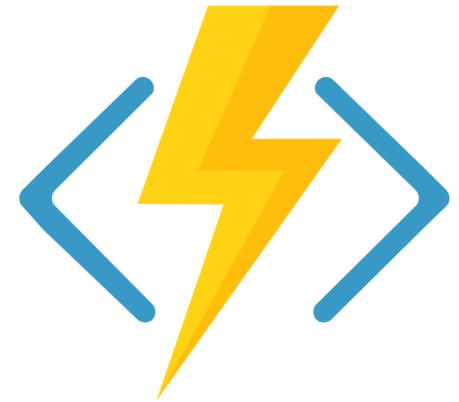
- Ergebnisorientierte Entwicklung mit einem einfachen Modell
- Viele vorhandene Integrationen mit Cloud-Diensten
- Nutzungsbasierte Bezahlung
- Flexible, quasi unbegrenzte Skalierung

Azure Functions



Was ist eine Function?

- Ein Stück Code in einer der unterstützten Sprachen
 - C#, F#, Node.js, Java, Python, PHP, Batch, Bash
- Metadaten, die die Trigger und Bindungen beschreiben
- Läuft auf der Functions-Runtime (Open Source)



Trigger

- Starten eine Function als Reaktion auf ein Event
 - Storage (Blob, Queue)
 - Azure Cosmos DB
 - Timer
 - HTTP
 - Webhook
 - Event Hub
 - Service Bus (Queues, Topics)
 - Microsoft Graph



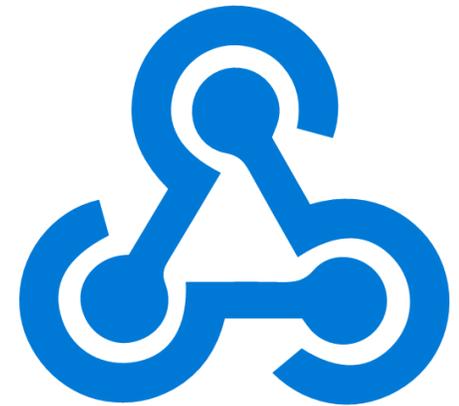
Bindings (Input)

- Zusätzliche Daten zur Verarbeitung
 - Storage (Blob, Table)
 - SQL DB
 - Cosmos DB
 - Microsoft Graph



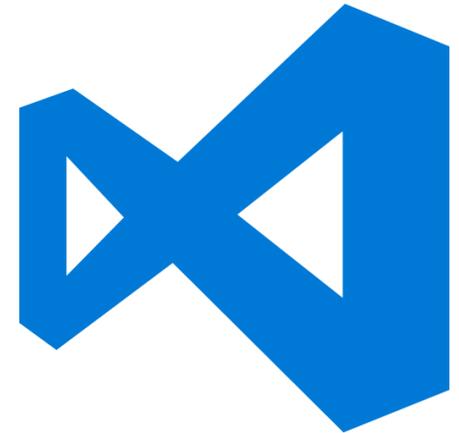
Bindings (Output)

- Storage (Blob, Table, Queue)
- HTTP-Antwort
- Event Hub
- ServiceBus (Queue, Topic)
- SQL DB
- Cosmos DB
- Notification Hub
- Twilio, SendGrid
- Microsoft Graph



Entwicklung von Funktionen

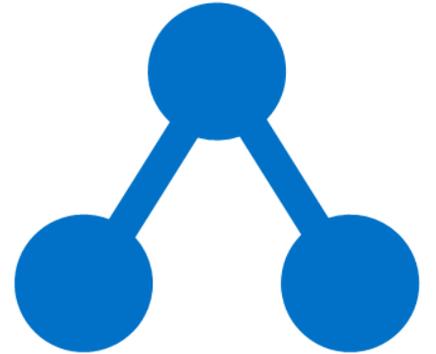
- Skript-basierte Funktionen im Azure Portal
 - C#, F#, Node.js, etc.
- Vorkompilierte Funktionen mit dem WebJobs SDK
 - C# in Visual Studio



DEMO

Azure Functions Proxies

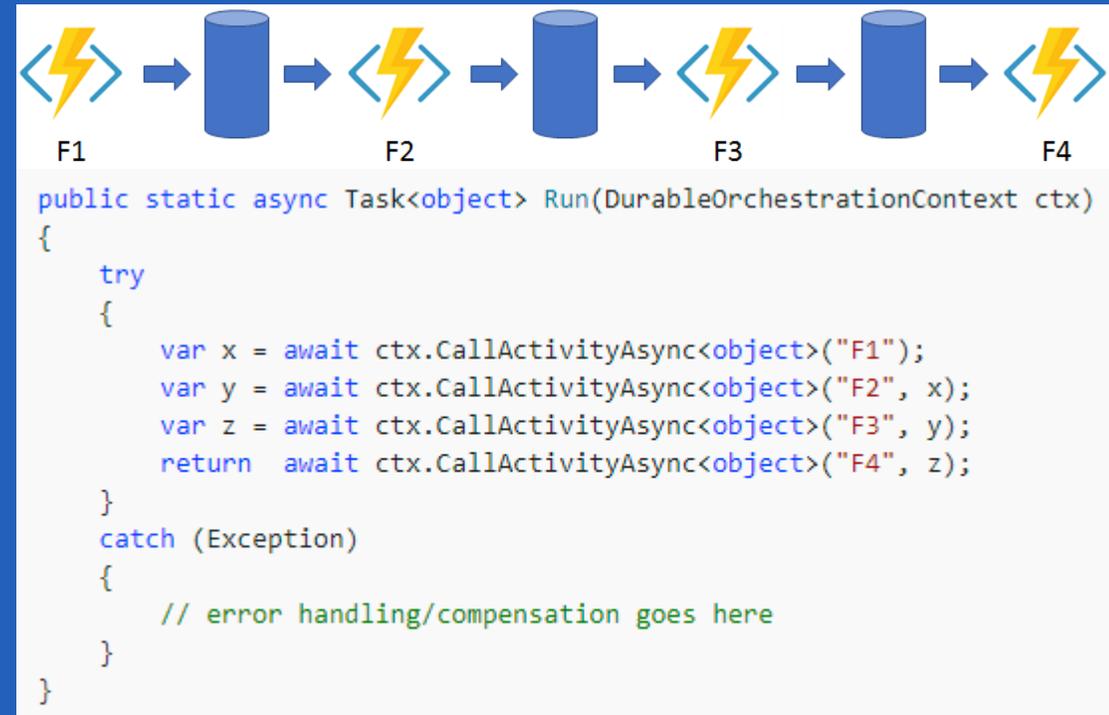
- Deklarativer Proxy
- Request- und Response-Rewrites
- Platzhalter und AppSettings in URLs und Rewrite-Regeln
- Abrechnung wie eine normale Function



DEMO

Durable Functions

- Orchestrierung von Functions durch asynchronen Code
- Checkpointing der Orchestrator-Function bis zum Eintreffen des Events
- Viele unterstützte Patterns
 - Function Chaining
 - Fan Out/Fan In
 - Async HTTP
 - Stateful Singleton
 - Human Interaction (e.g. Approval)



Was wir sahen (und was nicht)

- Azure Functions
 - Bindings, Triggers
 - Proxies
 - Durable Functions
 - Entwicklung im Portal
 - Precompiled Functions
 - Entwicklung mit Visual Studio
- Xplat-Tooling
- Application Insights Integration

Danke für die Aufmerksamkeit!

Fragen werden geduldet ;)

Ressourcen

- Why Serverless Architecture?

<https://cloudncode.blog/2017/05/15/why-serverless-architecture/>

- Der DemoBot ist auf GitHub!

<https://github.com/TimVinkemeier/DemoBot>