Microsoft

A Brief Journey from Device to Action …

hackster.io

IoT Virtual Bootcamp

December
12 – 14, 2017

http://aka.ms/IoTVirtualBootcamp

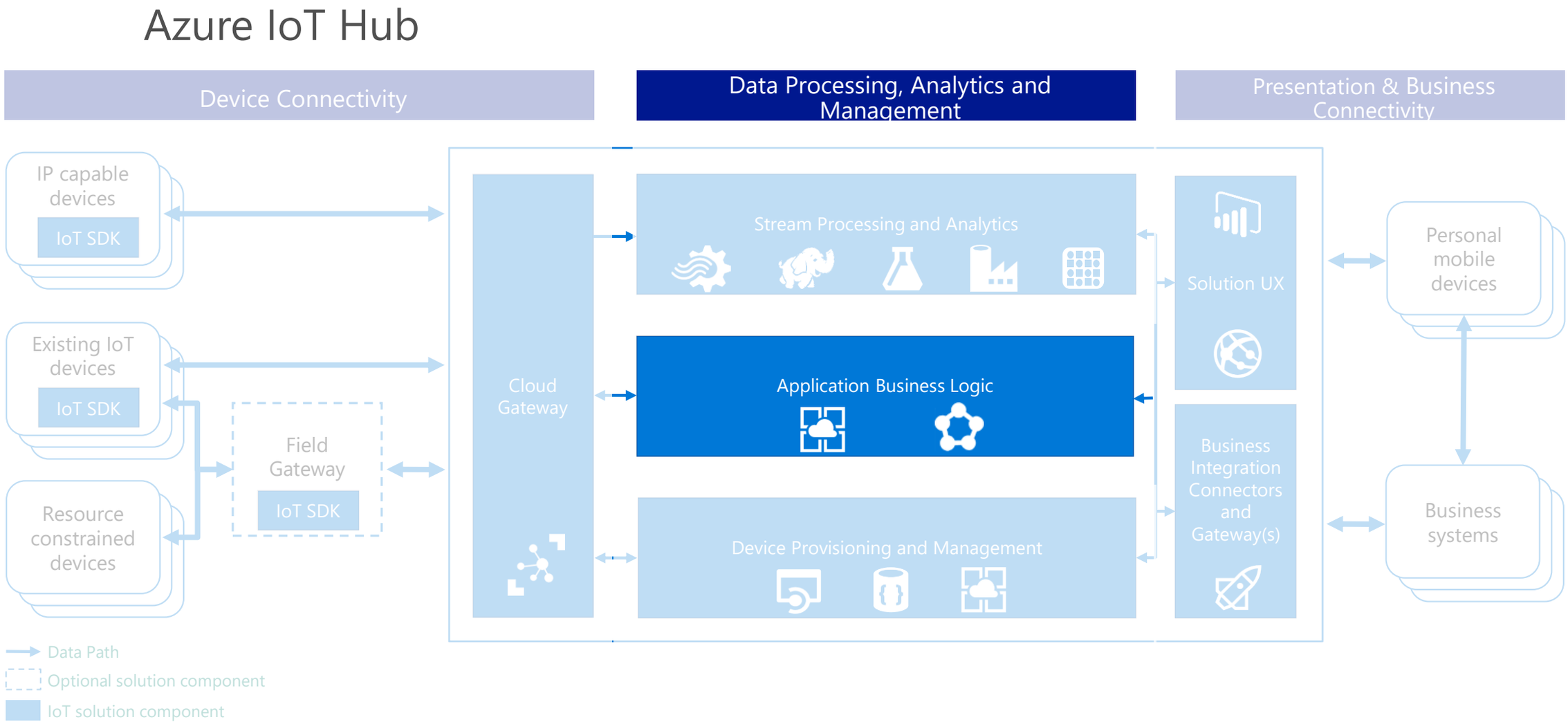
Hardware Partners : adafruit

Microsoft

Microsoft Azure Functions

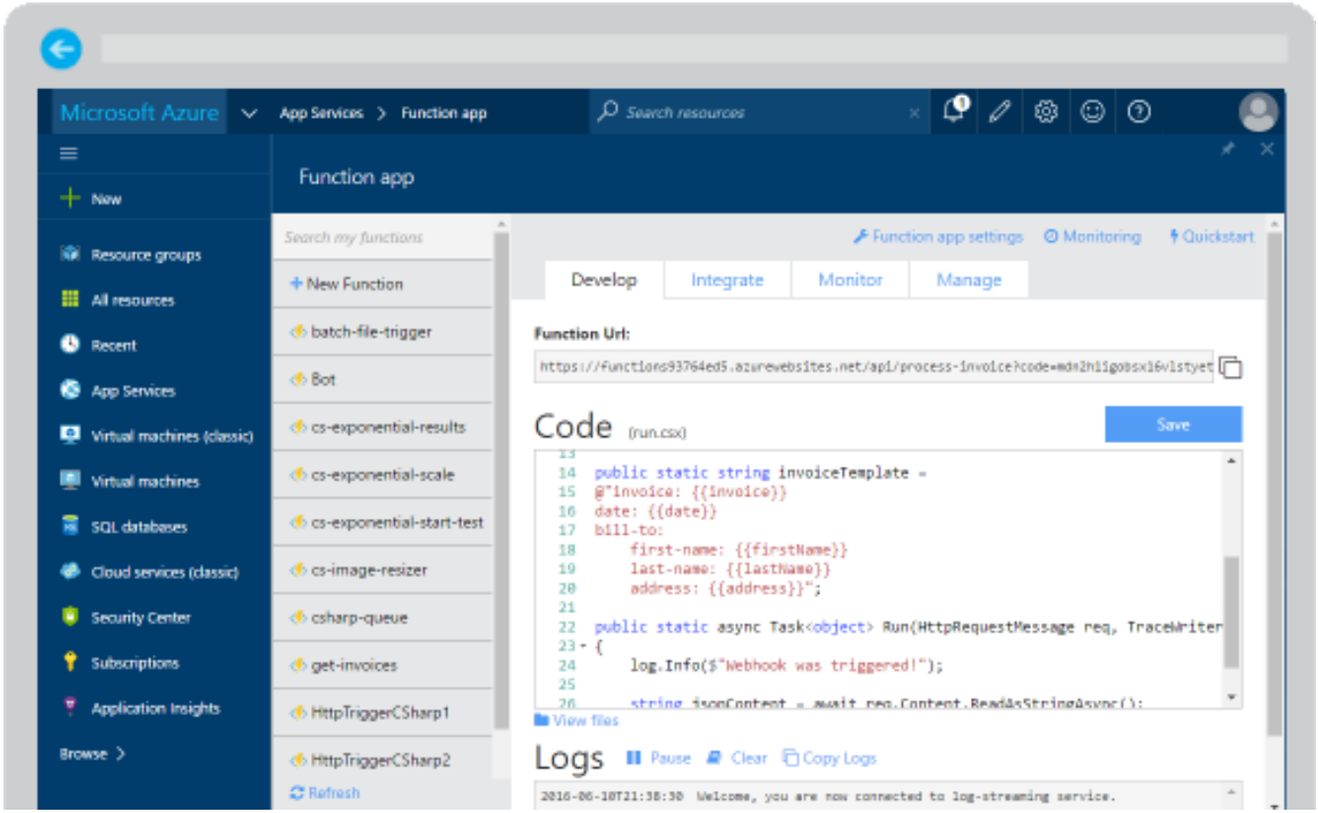
Kevin Saye

Azure IoT Hub



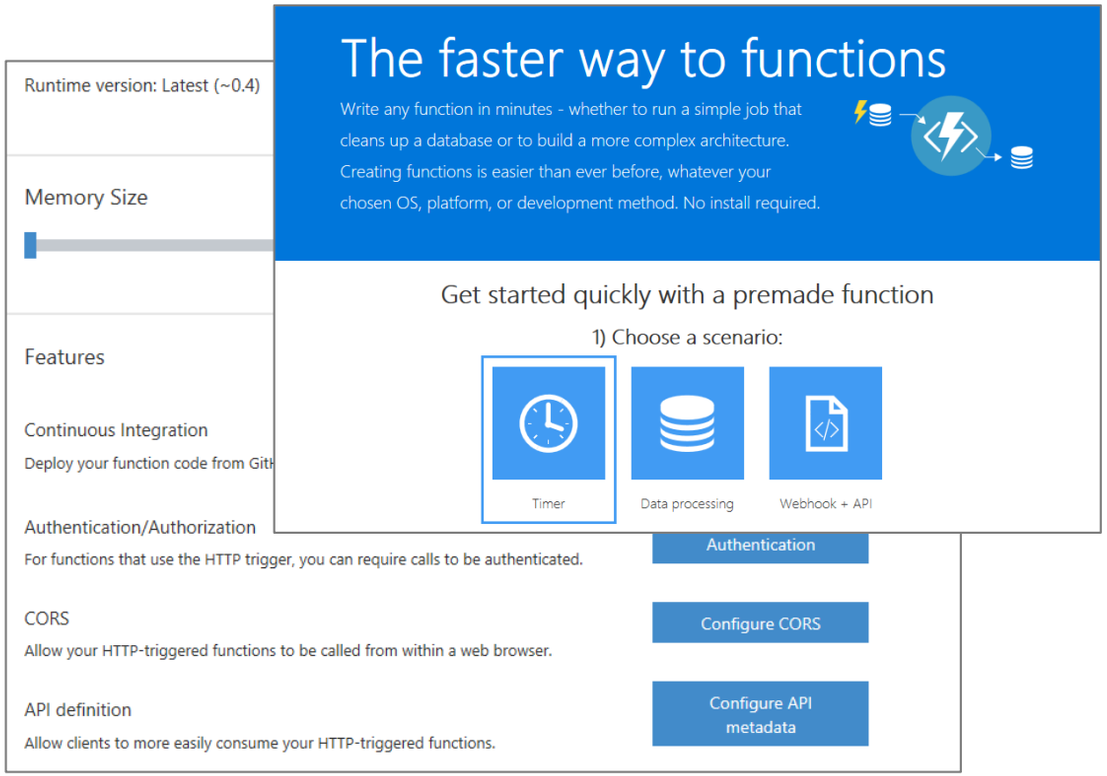
Azure Functions

Create a “serverless” event-driven experience that extends the existing Azure App Service platform by building “nanoservices” that can scale based on demand



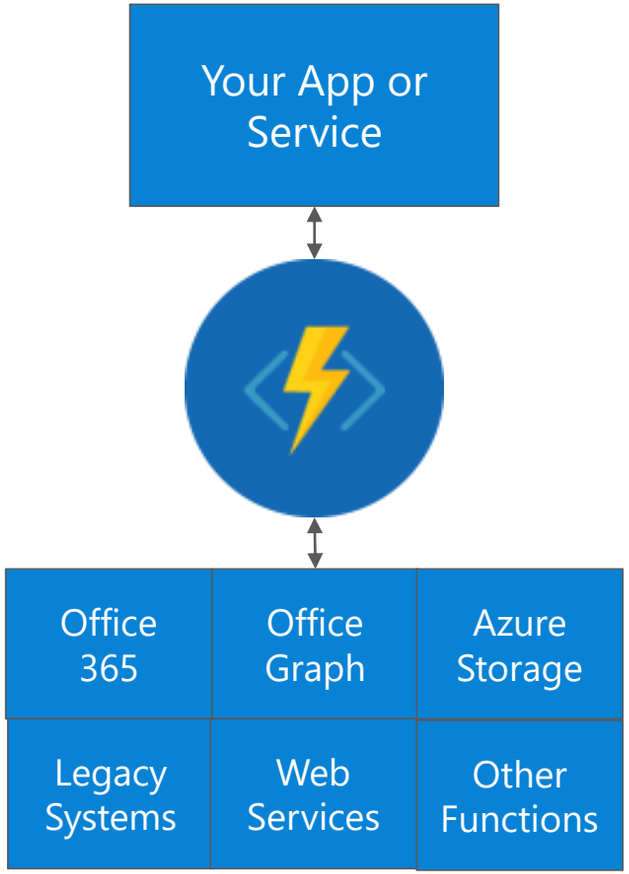
Supported Languages and Tools

Create functions in JavaScript, C#, Python, and PHP, as well as scripting options such as Bash, Batch, and PowerShell, that can be triggered by virtually any event in Azure, 3rd party services, or on premise systems



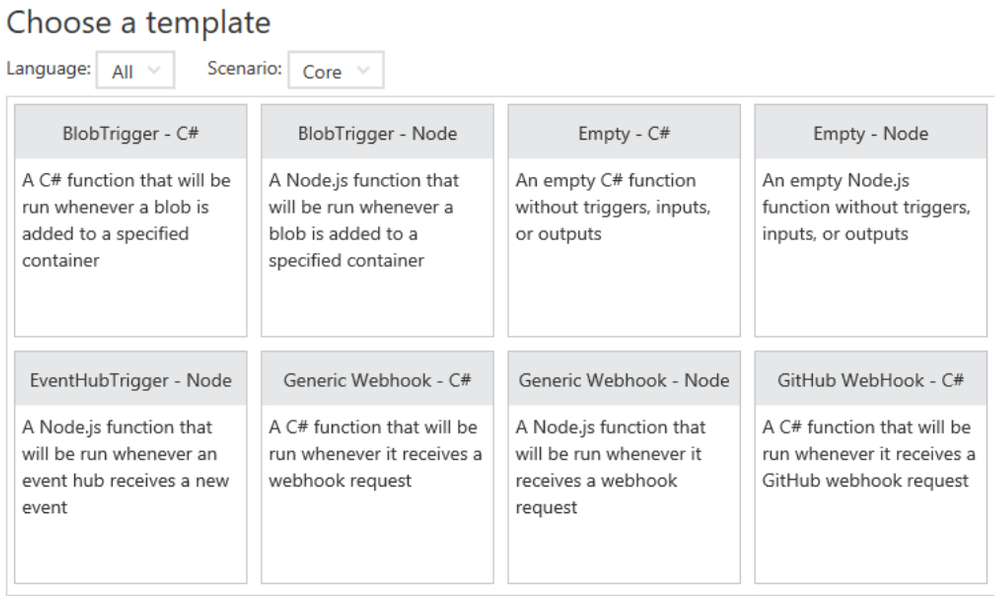
Common Scenarios

- Timer-based processing
- Azure service event processing
- SaaS event processing
- Serverless web application architectures
- Serverless mobile backends
- Real-time stream processing
- Real-time bot messaging



Function App Templates

Function App templates are categorized into general areas of Timer, Data Processing, and Webhook & API



- BlobTrigger
- EventHubTrigger
- Generic webhook
- GitHub webhook
- HTTPTrigger
- QueueTrigger
- ServiceBusQueueTrigger
- ServiceBusTopicTrigger
- TimerTrigger
- Blank & Experimental

Timer Function Apps

- Run at explicitly specified intervals, like every day at 2:00 am using CRON expressions, like “0 */5 * * * *” (every 5 minutes)
- Can send information to other systems, but typically don’t “return” information, only write to logs
- Great for redundant cleanup and data management
- Great for checking state of services
- Can be combined with other functions

Data Processing Function Apps

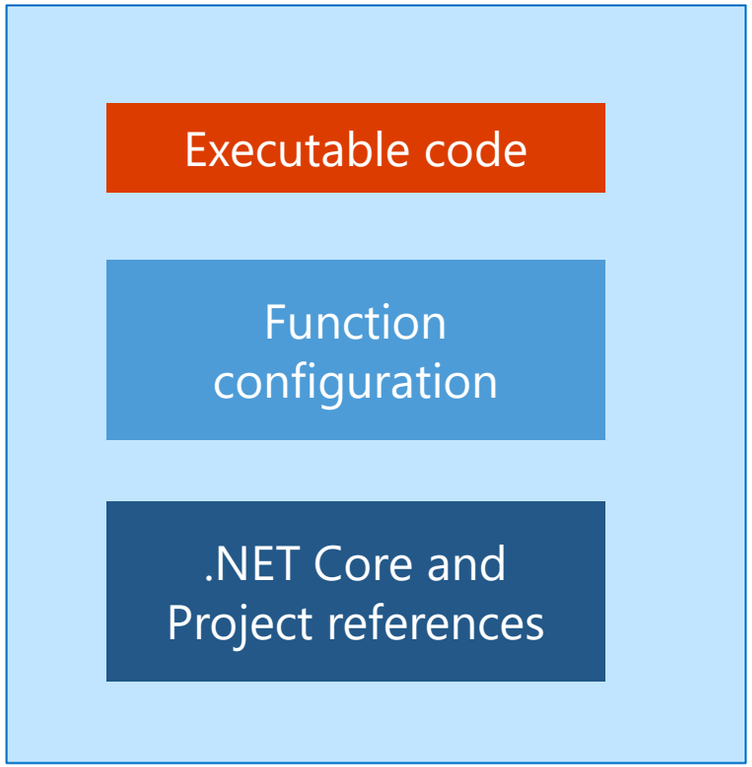
- Run when triggered by a data event, such as an item being added to a queue or container
- Typically have in and out parameters
- Great for responding to CRUD events
- Great for performing CRUD events
- Great for moving content
- Access data across services

Webhook & API Function Apps

- Triggered by events in other services, like GitHub, Team Foundation Services, Office 365, OneDrive, Microsoft PowerApps
- Takes in a request and sends back a response
- Often mimic Web API and legacy web services flows
- Typically need CORS settings managed
- Best for exposing functionality to other apps and services
- Great for building Logic Apps

Anatomy of a Function

- A “Run” file that containing the function code
- A “Function” file containing all service and trigger bindings and parameters
- A “Project” file containing project assembly and NuGet package references
- App Service settings, such as connection strings and API keys



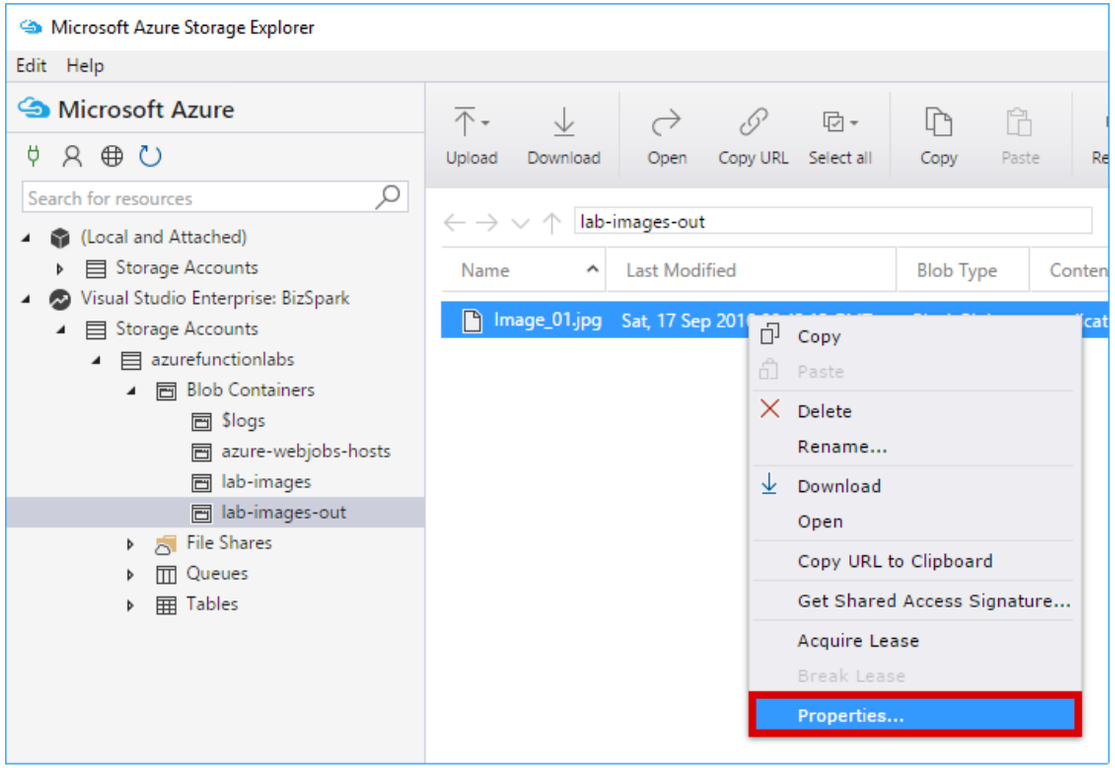
Function Bindings

Bindings serve as the basis for all connections to and from a function. Many bindings can be “bi-directional” as well.

Type	Service	Trigger	Input	Output
Schedule	Azure Functions	✓		
HTTP (REST or webhook)	Azure Functions	✓		✓*
Blob Storage	Azure Storage	✓	✓	✓
Events	Azure Event Hubs	✓		✓
Queues	Azure Storage	✓		✓
Tables	Azure Storage		✓	✓
Tables	Azure Mobile Apps		✓	✓
No-SQL DB	Azure Cosmos		✓	✓
Push Notifications	Azure Notification Hubs			✓

Testing Functions

- Command-line tools
- 3rd party products such as Postman and Swagger
- Direct web calls via cURL
- Nested functions
- Microsoft Azure Storage Explorer
- Visual Studio Cloud Explorer
- Visual Studio



Demonstration

- Creating a Function
- Show existing functions:
  - https://github.com/ksaye/IoTDemonstrations/blob/master/eGauge/run.csx
  - https://github.com/ksaye/IoTDemonstrations/blob/master/Senet/index.js
  - https://github.com/ksaye/IoTDemonstrations/blob/master/IoT2ADLS/index.js

Summary:

Azure Functions is a Serverless architecture for data processing in IoT solutions

Azure Functions supports multiple development languages such as: Node, C#, Python, PHP and etc

Azure Functions are initiated one of 3 ways:

- Timers
- Data Processing
- Webhooks / API

Additional References:

- www.InternetofYourThings.com
- https://azure.microsoft.com/en-us/blog/tag/azure-functions/
- https://azure.microsoft.com/en-us/services/functions/
- https://docs.microsoft.com/en-us/azure/azure-functions/

Microsoft

www.InternetofYourThings.com