Azure Functions

By: Khushboo Baheti, Kiruthiga Gunasekaran, Siri Sadashiva, Suganya Jeyaram

Azure Functions

Microsoft released Azure Functions in March 2016

Serverless compute service that enables you to run code on-demand without having to explicitly provision or manage infrastructure

Azure Functions lets you develop serverless applications on Microsoft Azure.

Functions can make development even more productive

Features

Choice of language : C#, F#, JavaScript

Pay-per-use pricing model : Pay only for the time spent running your code

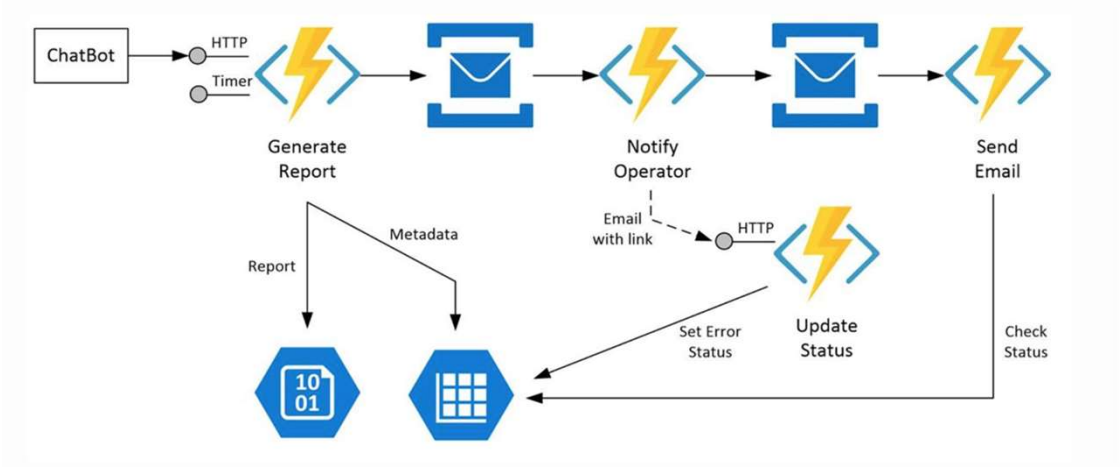
Bring your own dependencies: NuGet, NPM

Open-source: The Functions runtime is open-source and available on Github

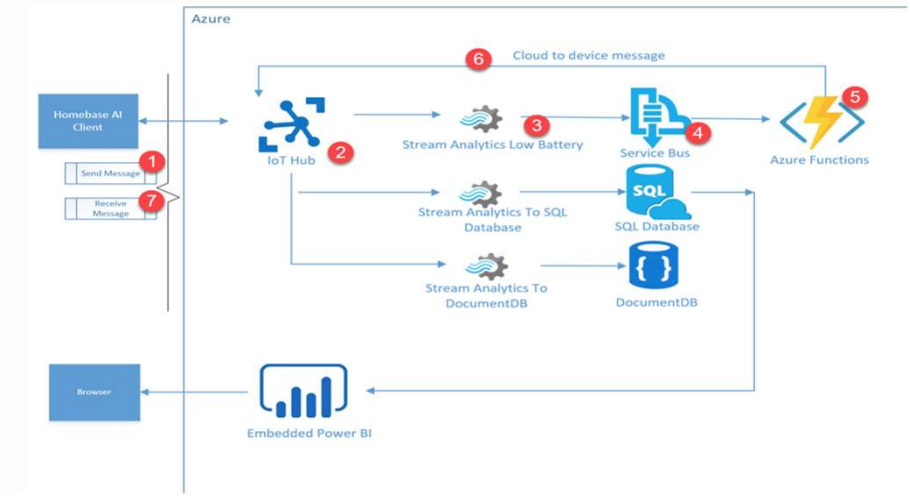
Integrated security: Protect HTTP-triggered functions with OAuth providers such as Azure Active Directory, Facebook, Google, Microsoft Account.

Flexible development - Code functions in the portal or set up continuous integration and deploy code through Github,Visual Studio Team Services

Use Case: Automating a furniture factory



Use Case: Homebase



Advantages

Every function automatically maps to an HTTP endpoint if enabled

Event grid trigger for Azure functions

Disadvantages

Slower to scale when compared to AWS Lambda

One other aspect is testing. Unlike working in a Paas environment where you can use the automated tools that are part of your development lifecycle, the tooling for Visual Studio 2015 is not the best.

Cost Discussion

Consumption Plan:

•Billed based on per-second resource consumption and executions.

•

*includes a monthly free grant of 1 million requests and 400,000 GB-s of resource consumption per month

•Memory is measured by rounding up to the nearest 128 MB, up to the maximum of 1,536 MB.

•Minimum execution time and memory for a single function execution is 100ms and 128mb.

App Service Plan:

•Best choice if existing, underutilized VMs that are already running other App Service instances.

•App Service Plan can be more cost-effective if function are running continuously.

•The service scales out manually or automatically depending on the options chosen. For example, based on CPU load on a five-minute window.

Cost Discussion

Consumption Plan:

- Billed based on per-second resource consumption and executions.

METER	PRICE	FREE GRANT (PER MONTH)
Execution Time*	$0.000016/GB-s	400,000 GB-s
Total Executions*	$0.20 per million executions	1 million executions

*includes a monthly free grant of 1 million requests and 400,000 GB-s of resource consumption per month

- Memory is measured by rounding up to the nearest 128 MB, up to the maximum of 1,536 MB.
- Minimum execution time and memory for a single function execution is 100ms and 128mb.

App Service Plan:

- Best choice if existing, underutilized VMs that are already running other App Service instances.
- App Service Plan can be more cost-effective if function are running continuously.
- The service scales out manually or automatically depending on the options chosen. For example, based on CPU load on a five-minute window.

Cost Example

Monthly billing (consumption plan) calculated as follows for a function with

- 512 MB memory consumption
- 1sec execution time
- 3,000,000 executions in one month

- Resource Consumption Billing Calculation

  Execution time: 3,000,000 * 1 sec = 3 million seconds

  Resource consumption: 3 million seconds * 0.5 GB = 1.5 million GB-s

  Total billable consumption: 1.5 million GB-s - 400,000 GB-s = 1.1 million GB-s

  Resource consumption cost: 1.1 million GB-s * $0.000016/GB-s = **$17.60**

- Executions billing calculation

  Billable executions: 3 million – 1 million = 2 million

  Execution cost: $0.20 (per million executions) * 2 = **$0.40**

Total cost: **$17.60 + $0.40 = $18**

Cost Example – Worst case

2 threads running continuously, taking 512 MB memory each on B1S VM (1 vCPU, 1 GB RAM)

Cost of running B1S VM = $0.015/hour * 24 * 30 = **$10.8**

Resource consumption for 2 threads = (3 million GB-s – 400,000 GB-s) * $0.000016/GB-s = $41.6

Execution cost for 2 threads = $0.20 (per million executions) * 5 = $1

Total cost for Azure Function: $41.6 + $1 = **$42.6 (~4 times expensive)**

Alternatives

- **AWS Lambda**
  - Launched in 2014.
  - Supports a range of runtime environments including NodeJS, Python, Java and C#
- **Google Cloud Functions**
  - Launched in 2016.
  - Only supports a single runtime environment using NodeJS.
- **IBM Cloud Functions**
  - Launched in 2017.
  - Based on Apache OpenWhisk and supports Node.js, Python, Swift, Java, and PHP.

Comparing alternatives

Feature	Azure Function	AWS Lambda	Google Cloud Function
Scalability	Automatic scaling (Consumption Plan) Manual or metered scaling (App Service Plan)	Automatic scaling	Automatic scaling
Max # of functions	Unlimited functions	Unlimited functions	1000 functions/project
Concurrent executions	No limit	1000 parallel executions per account, per region (request to increase)	No limit
Max execution	5 mins	5 mins	9 mins
Supported languages	C#, JavaScript, F#, Python, Batch, PHP, PowerShell	JavaScript, Java, C#, and Python	Only JavaScript
Deployments	Visual Studio Team Services, OneDrive, Local Git repository, GitHub, Bitbucket, Dropbox, External repository	Only ZIP upload (to Lambda or S3)	ZIP upload, Cloud Storage or Cloud Source Repositories
Pricing	1M requests for free then $0.20/1M invocations, plus $0.000016/GB-s	1M requests for free then $0.20/1M invocations, plus $0.00001667/GB-sec	1M requests for free then $0.40/1M invocations, plus $0.00000231/GB-sec

Conclusion

- Faster development and time to production is less than a day.
- Less overhead in scaling or maintaining applications. Moving from DevOps model to almost No-Ops.
- A wide range of triggering options.
- Continuous integration: Use Git to push your code, Azure function redeploys automatically.
- Azure Functions does not have limit on concurrent connections unlike AWS Lambda, which can happen due to latency, retries, throttling from underlying services.

Demo

- **1.Create Function and testing via HTTP REST call**
- **using Azure Portal**
- **Using Azure CLI**
- **Using Visual Studio IDE**
- **2.Durable Functions**
- **Function Chaining**
- **Fan-out/Fan-in**
- **3. Triggers**
- **- Create a function triggered by timer**
- **- Create a function triggered by github webhook**
- **4.Integration**
- **- Add messages to an Azure Storage queue using Functions**

References

- https://azure.microsoft.com/en-us/services/functions/
- https://read.acloud.guru/aws-lambda-vs-google-cloud-functions-vs-azure-functions-who-has-the-serverless-advantage-f6c2535e72f4

Thank you