

# LOG8415 Personal Project

## Complete CI/CD setup on Azure to deploy serverless functions

1897277 - Benjamin Archambault

Friday, December 17<sup>th</sup> 2021

November 28, 2021

## 1 Introduction

### 1.1 Problem

As cloud computing becomes more popular, developers want easy way to implement their application using this new programming paradigm without having to worry about the infrastructure of their code. Too often than not, it is the programmers' fault if flaws or bugs arise. It is therefore essential to offer them a basic infrastructure to write good code.

In addition, fierce competition forces companies to develop their project more and more quickly. It would then be useful to offer them an intuitive interface to facilitate the development of their projects.

Also, several startups are trying to develop digital solutions using the cloud to reduce their costs. Not all companies can invest in hardware to host their projects. This is why they use the cloud to reduce operating costs.

### 1.2 Literature Review

- **Serverless functions app development and deployment**[1]: This article presents the general idea of how to implement and deploy a serverless functions app. The authors also give references to the event-driven architecture style because an application using serverless functions will most likely be built around this design pattern. In my opinion, this article is complete because it presents all the aspects of a serverless functions application and because it gives guidelines how to implement good code.
- **What Are Serverless Functions?**[2]: This article presents the utilities of serverless functions and why to use them. It is easier to understand for someone who does not know what is a serverless functions application

because it only gives a general overview of the architecture. What I also like about this article is that they give example use-cases of a serverless app. It helps me understand why use this solution instead of existing solutions and what are the benefits of serverless applications.

### 1.3 Proposed solution

The idea behind serverless functions is to focus on writing back-end code and let the cloud vendor (Azure) handle the operational infrastructure. It makes deployment easier because you can integrate CI/CD when pushing code to a certain branch or after a pull request. Also, if you have to make modification to the code, there will be practically no downtime. With Azure's *Functions App*, you can have your serverless application managed entirely in the cloud. You only have to worry about developing new code and paying the bills. Also, because Azure has its own security team and because it is used by millions of users, you can be sure that your API will be fairly secured.

## 2 Architecture

Serverless functions is « a piece of business logic that is both stateless (does not maintain data) and ephemeral (is used and destroyed), as it potentially only lasts for seconds, and is designed to be triggered by a specific condition. » [1] As stated in its name, serverless functions are part of the back-end of an application. They can be triggered by an HTTP request or scheduled to run automatically at a certain time. Serverless functions are independent from each other and do not interact with other serverless functions. Also, serverless functions are interoperable because they get triggered by an HTTP request and the result can be any HTTP response. (HTML, JSON documents, HTTP response). The interoperability is a good property of a cloud component because it can be used by all your project components.

## 3 Azure Services

To deploy a serverless functions app, you must create some Azure components to host your application.

The first component needed to run a serverless application, is the *Resource Group*. This component is necessary because it contains all the Azure components for your application. Those resources share the same life cycle to the other resources in the *Resource Group*.

Another important resource is the *Functions App* itself. This component represents the serverless functions that are used for the project. Also, this component scales with the usage of the functions. It is an on-demand service which means that you pay for what you use so if the function is highly used, Azure will take

care of the scaling but you will have to pay more credits.

The last service used is the *Storage Account*. This resource is used to store any type of data. (Blobs, files, queues, tables and many more) Azure offer many types of storage account for different uses and price range. The basic storage account is the *Standard general-purpose v2* and they all include redundancy options.

## 4 Data flow

The current section will express the data flow of the solution proposed in this paper.

- **Implement the function:** The developer implements a function that can be triggered by an HTTP request for example.
- **Define a trigger event:** The developer defines a trigger for the function to run. This trigger can be from an HTTP request (GET, POST, ...), from a scheduler (At a given time), from a queue and many more.
- **Trigger the event:** A user triggers the event using the function trigger set in the step above.
- **Deploy the function:** The cloud provider (Azure) create a new instance that runs the given function. (Developers can set custom properties like timeout, hardware utilisation and instance location) The result of the function is returned as an HTTP response to the user.

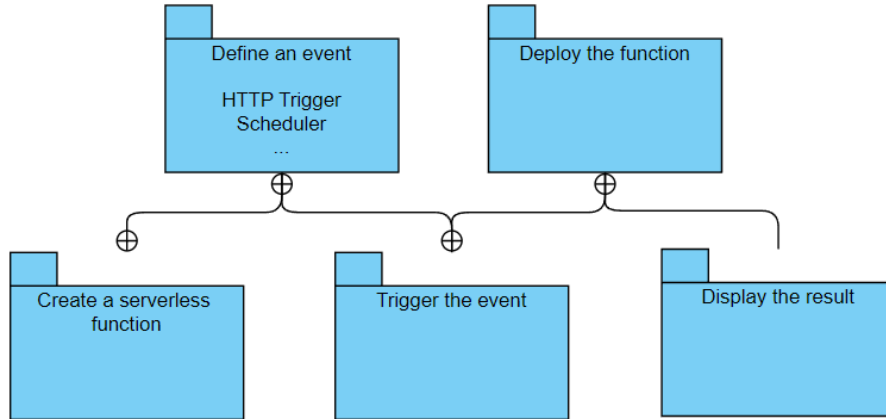


Figure 1: Azure Functions App data flow

## 5 Instructions for the demo

<https://github.com/ReptarK/log8415-project/blob/master/README.md>

## 6 Summary

With the rise of cloud computing, serverless functions are widely used for many reasons.

First, the cloud provider handles the administration of the infrastructure. This makes the solution easier to develop. The programmer can focus more on the code instead of having to manage the code infrastructure.

Also, because the majority of cloud providers work on a pay-as-you-go basis, the costs of maintaining code and hardware are often less expensive than having to provision your own infrastructure to store the functions.

Subsequently, as the back-end is mostly managed by the cloud provider, less code has to be implemented by the programmer. This reduces the risk of errors and decreases the workload.

Finally, because we are in the cloud, we are offered a virtually unlimited resource. This makes it easy to scale your application dynamically if demand changes over time. Also, your footprint is likely to decrease, because the resources used by cloud providers are optimized to reduce energy consumption.

## References

- [1] Rogeriohc, Alexbuckgit, dsk-2015, DCtheGeek, and v-thepet, “Serverless functions app development and deployment - azure example scenarios,” Serverless Functions app development and deployment - Azure Example Scenarios, 11-Dec-2021. [Online]. Available: <https://docs.microsoft.com/en-us/azure/architecture/serverless-quest/application-development>. [Accessed: 28-Nov-2021].
- [2] Splunk, “Serverless functions: A beginner’s guide,” What Are Serverless Functions? [Online]. Available: [https://www.splunk.com/en\\_us/data-insider/what-are-serverless-functions.html](https://www.splunk.com/en_us/data-insider/what-are-serverless-functions.html). [Accessed: 28-Nov-2021].