BDWatchdogFaaS: A Tool for Monitoring and Analysis of Functions-as-a-Service in Cloud Environment

Manuel Framil, Mario Carpente, David Fraga, Jonatan Enes, Roberto R. Expósito, Guillermo L. Taboada, and Juan Touriño

Centro de Investigación CITIC, Universidade da Coruña, 15071 A Coruña, Spain
Correspondence: m.framil.deamorin@udc.es

DOI: *https://doi.org/10.17979/spudc.000024.52*

Abstract: BDWatchdog is a framework to assist in the in-depth and real-time analysis of the execution of Big Data frameworks and applications. BDWatchdog was originally developed to monitor Hadoop ecosystems deployed on serverless containers, in order to detect bottlenecks and spot certain patterns that frameworks or applications may have. In this work we shift the focus to monitoring serverless functions in the public cloud, by proposing an extension of BDWatchdog which captures, transforms and analyzes logs from both AWS Cloudwatch and Azure Application Insights, which store the logs from AWS Lambda and Azure Functions (respectively), the FaaS (Function-as-a-Service) solutions of the two main public cloud providers, AWS and Azure. The extension, called BDWatchdogFaaS, builds and stores a common model to both providers, allowing to consult, analyze and monitor function logs from AWS and Azure indistinctly. The transformation of logs into the common model is done by a FaaS of the corresponding provider, which in near real-time ingests, processes and sends the data to a common storage. In addition, the data is forwarded to a Power BI dashboard so that the serverless functions can be monitored easily.

1 Introduction

Cloud computing refers to a paradigm of computing that has undergone a enormous growth in the last decade, and in which the computing resources and services are consumed on demand through the internet. This has resulted in a fundamental change in every stage of the software applications life cycle, from it's conception and development until the way they are used by the final user. In cloud computing, the cloud provider is responsible for allocating, managing and scaling the underlying resources of the application, releasing the developers from the responsibility of administrating the infrastructure, thus allowing them to focus only in the functionality.

In recent years a new model of cloud computing, called Function-as-a-Service (FaaS) has been presented. This new paradigm allows to run small fragments of code (functions) without the need for allocating virtual machines or deploy any kind of server. The code is uploaded to the cloud provider, which automatically allocates the resources needed, run the function and returns the results. Faas are commonly employed to build microservices applications, because they are easily integrated between them and with other cloud services, and their code can be written en different programming languages depending of the needs of the application. Besides, FaaS are event-driven, which means they can be activated in response to specific events, such as HTTP requests, database changes or messages in queues, among others.

This work is part of the project PICSA (Productivity Increase by Cloud Serverless Automation, for further details see the acknowledgement section), which main objective is to increase the productivity of developers and organizations when working with serverless technologies, particularly with FaaS.

2 BDWatchdogFaaS Architecture

The main contribution of this work is the design, implementation and publication of BDWatchdogFaaS, an application performance monitoring (APM) tool for FaaS in a multicloud environments. Previous relevant work is BDWatchdog (Enes et al., 2018), a monitoring and profiling tool for Big Data applications in Hadoop environments; and Serverless Containers (Enes et al., 2020), a framework for real-time container auto-scaling. BDWatchdogFaaS has been built as an cloud-native extension of BDWatchdog (Enes et al., 2018) to capture logs from AWS Lambda and Azure Functions, the FaaS solutions of Amazon Web Services and Azure, respectively. The main issue that BDWatchdogFaaS solves is the dependency of a particular provider due to the incompatibility between FaaS logs formats of different cloud providers. To overcome this problem, BDWatchdogFaaS proposes a common data model that enables to store the data in a single database, extract overall metrics and visualize the executions of FaaS from different hyperscalars simultaneously.

The architecture of BDWatchdogFaaS, presented in Figure 1 is based on microservices , which are independently deployable services, very loosely coupled and integrated through APIs.

BDWatchdogFaaS is composed of four modules, each of them being responsible for a specific task:

- **Azure monitoring module:** This module deals with the monitoring of Azure Functions and it has been implemented completely using Azure services. When an Azure Function is executed, Azure automatically generate a series of logs which are stored in Application Insights, the APM tool provided by Azure. Once in Application Insights, logs are immediately sent to a queue of EvenHub, which work as trigger to a Azure Function, called *AzureLogsForwarder*. This FaaS handles the task of processing logs to extract the relevant information, and then forwards it to the statistics and visualization modules.
- **AWS monitoring module:** This module deals with the monitoring of AWS Lambda. Analogous to the Azure monitoring module, this module has been built on top of AWS services. First, when a Lambda is executed, AWS registers the logs in a log group of Cloudwatch, the monitoring service of AWS. Using a subscription filter logs are automatically sent to a Lambda function, called *AWSLogsForwarder*, which is trigger as response to the incoming data. As for *AzureLogsForwarder*, this function parses logs to capture the most relevant information, and then forwards it to the statistics and visualization modules.
- **Statistic module:** This module has been developed to store the logs processed by the previous modules in an AWS DynamoDB Serverless table. DynamoDB is a fully managed, key-value NoSQL database designed to run high-performance applications provided by AWS.

  The table keeps all processed logs from both Azure Functions and AWS Lambda in a single point, so that another Lambda function, *LogStatsForwarder*, can read the data and obtain overall statistics about the FaaS being monitored in both cloud providers. These statistics are forwarded to the visualization module as well.
- **Visualization module:** Finally, the visualization module is built making use of the capabilities for real-time visualization of streaming datasets in Power BI Service. Up to four datasets have been defined: one for Azure logs, another for Lambda, the third dataset collects logs from both, and the last one receives the stats data. From these streaming datasets it has been created several visualizations that are updated in real-time. All of them have been gathered in a single dashboard panel in Power BI Service, available to users to review the behaviour of the running FaaS.

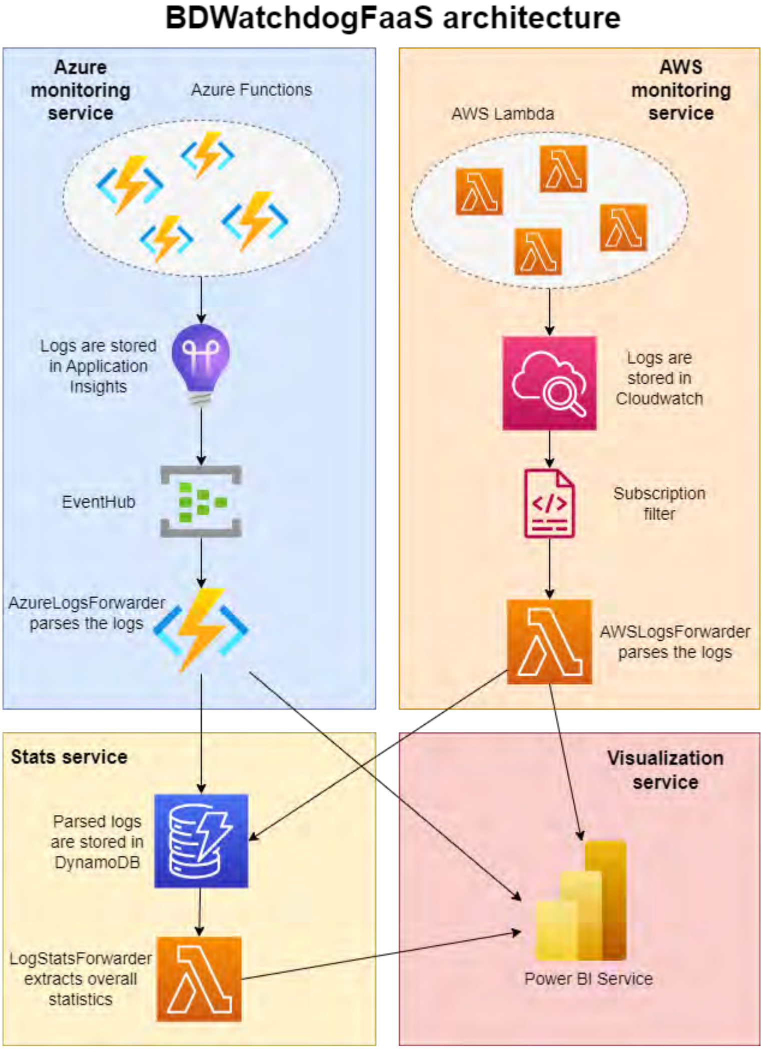
BDWatchdogFaaS architecture
Azure monitoring service
Azure Functions
Logs are stored in Application Insights
EventHub
AzureLogsForwarder parses the logs
AWS monitoring service
AWS Lambda
Logs are stored in Cloudwatch
Subscription filter
AWSLogsForwarder parses the logs
Stats service
Parsed logs are stored in DynamoDB
LogStatsForwarder extracts overall statistics
Visualization service
Power BI Service

Figure 1: Architecture of BDWatchdogFaaS

3 BDWatchdogFaaS Deployment

BDWatchdogFaaS has been publish in the Serverless Application Repository (SAR) of AWS, a platform where developers can find, share, and deploy serverless applications and components. SAR allows users to discover pre-built serverless applications or functions created by the AWS community or AWS partners. The objective is to ease its deployment and broad its application. The applications are described using Cloudformation templates. They are key to define the architecture of serverless applications in the AWS ecosystem, enabling its automatic deployment. Cloudformation templates have been written using the AWS Serverless Application Model (SAM), an open-source framework for building serverless applications. It provides shorthand syntax to express functions, APIs, databases, and event source mappings. During deployment, SAM transforms and expands the SAM syntax into AWS CloudFormation syntax, enabling developers to build serverless applications faster.

Therefore, a SAM template defining the tool's architecture and configuration parameters (variables that must be set by the final user, such as the Power BI Service URLs to the streaming datasets) has been developed in order to publish BDWatchdogFaaS.

4 BDWatchdogFaaS Evaluation

BDWatchdogFaaS has been developed employing a TDD (Test Driven Development) approach to test its individual components and module integration. By doing so, it is ensured that the final product is well-tested when it comes to functionality and integration, leaving to this stage load and stress tests. Another aspect that needs to be tested is the fit of BDWatchdogFaaS in the market of FaaS monitoring tools.

The tests carried on are aimed to measure the reliability of BDWatchdogFaaS, putting the tool through several experiments that simulate real world environments. Furthermore, there are other tests that has been designed to verify the usability and fit of BDWatchdogFaaS in the market, checking out how easily can it be discovered inside the SAR and how quickly can it be deployed by a potential user. The tests were the following:

1. **Publication and discovery of BDWatchdogFaaS:** The objective is to verify that an user can find BDWatchdogFaaS inside the SAR. To do so, we designed a set of queries that potential users would use to look for applications in the SAR, and measured in which position is BDWatchdogFaaS placed on each one. These queries include a combination of relevant terms that describe BDWatchdogFaaS, such as "monitoring", "FaaS" or "multicloud".

   The results show that for 8 of the 12 queries tested, BDWatchdogFaaS placed itself between the top 3 applications retrieved by the search engine.

2. **Deployment of BDWatchdogFaaS:** We tested that, once the user has found the application in the SAR, it can be deployed easily. We tried, successfully, to deploy BDWatchdogFaaS directly from the SAR interface, and were able to start running the tool in less than 5 minutes, proving that it can be effortlessly set up.

3. **Reliability during peaks of demand:** In order to validate that BDWatchdogFaaS can scale properly to respond peaks of demand, we have monitored a thousand executions of a Lambda in a period of 5 seconds. This test generated a large amount of logs. Rather than end up collapsing BDWatchdogFaaS, the tools was able to process all the logs without delays or errors.

4. **Long-running execution:** A selected set of four functions, has been executed in AWS and Azure, once per minute, during 24 hours straight so we can test whether BDWatchdogFaaS monitors continuously and seamless for such intervals of time. During the experiment, the operation of BDWatchdogFaaS was uninterrupted and no errors were yield, confirming that the tool can run for that, and probably longer, periods of time.

5 Conclusions

This paper has presented BDWatchdogFaaS, an extension of our previous work, BDWatchdog, which is able to watch over FaaS applications in a multi-cloud environment, allowing organizations to monitor simultaneously AWS Lambdas and Azure Functions, the main FaaS implementations. This tool, publicly available in AWS Serverless Repository (SAR), captures, transforms and analyzes logs from both AWS Cloudwatch and Azure Application Insights, which store the logs from AWS Lambda and Azure Functions. A common model for FaaS logs has been designed to unify data process and analysis. Moreover, data has been processed in near real-time and visualized in Power BI. This way, serverless functions can be monitored straightforwardly using a standard software stack, providing value to developers and stakeholders by increasing the productivity when working with FaaS.

As future work, it is planned to further develop BDWatchdogFaaS in the search of product market fit in the sector of FaaS monitoring tools. Moving from SAR to BDWatchdogFaaS as SaaS will increase its adoption and benefit from the lessons learnt analysing logs of a higher number of FaaS.

Acknowledgement

This work was funded by the Ministry of Science and Innovation of Spain (ref. PDC2021-121309-I00/MCIN/AEI/10.13039/501100011033) and by the European Union "NextGenerationEU/PRTR". CITIC is funded by the Xunta de Galicia through the collaboration agreement between the Consellería de Cultura, Educación, Formación Profesional e Universidades and the Galician universities for the reinforcement of the research centres of the Galician University System (CIGUS).

Bibliography

J. Enes, R. R. Expósito, and J. Touriño. Bdwatchdog: Real-time monitoring and profiling of big data applications and frameworks. *Future Generation Computer Systems*, 87:420–437, 2018. URL *https://www.sciencedirect.com/science/article/pii/S0167739X17316096*.

J. Enes, , R. R. Expósito, and J. Touriño. Real-time resource scaling platform for big data workloads on serverless environments. *Future Generation Computer Systems*, 105:361–379, 2020. URL *https://www.sciencedirect.com/science/article/pii/S0167739X19310015*.