Kalpa Publications in Computing

Volume XXX, 2022, Pages 15–17

Proceedings of V XoveTIC Conference. XoveTIC 2022

Automatic issue generation

Francisco Cedron[1], Michael Añon-Valera[2], Sara Alvarez-Gonzalez[1], Carlos Fernandez-Lozano[1], and Adrián Carballal[1]

[1] Research Center on Information and Communication Technologies
University of A Coruña, 15071 A Coruña, Spain
[2] Software Engineering Laboratory, Computer Science Faculty
University of A Coruña, 15071 A Coruña, Spain

Abstract

It is undeniable that computing has settled in the modern world. Many elements have enabled global digitization and products are being developed faster, largely through the use of third-party services. However, the systems developed, whether a proprietary or third party, can have degradation or service outages. Having an automatic system for the generation of incidents would allow acting as soon as possible on specific problems in the services and the development of a system that allows this automation is the main objective of this work.

1 Introduction

Gone are the days when web services were products that could be considered handcrafted, today the trend is to automate all the elements that can derive from the services and not depend on human beings and thus avoid errors inherent to the human condition itself.

A vitally important aspect is the creation of an incident generation system. It is true that today certain incidents can be resolved automatically. However, incidents that cannot be solved automatically require manual action and for this, it is necessary to have a system that warns of system crashes or anomalous operations in a system. Thanks to a notification system the time to resolve a problem can be reduced, which can mean reducing losses due to the shutdown of a service [4]. An example was the Fastly service outage, which resulted in an estimated loss of 32M $ to Amazon [3].

2 Objective

The objective of this work is to develop a service that allows the monitoring of the services that a company could have. If any check detects a drop or degradation in any of the services, the objective will be to carry out the launching of the corresponding incident in an automated way. This will be done with the help of the Instatus API that allows the publication of incidents on a status page. Another objective of this work is the application of techniques such as backoff in monitoring, to perform different checks repeatedly before indicating that a service is down [1].

3 Results

Finally, a service has been developed that allows checks to be entered into the system (figure 1). The system of checks has finally been realized using a messaging system with RabbitMQ that allows to have different processes to evaluate the checks periodically and also providing a horizontal scaling [5, 6]. The operation of the periodic checks is shown in figure 2.

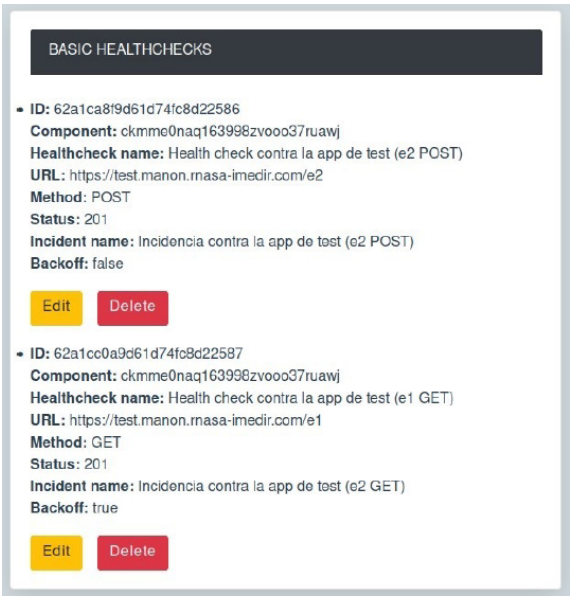

Figure 1: Example of incidents created in the application.

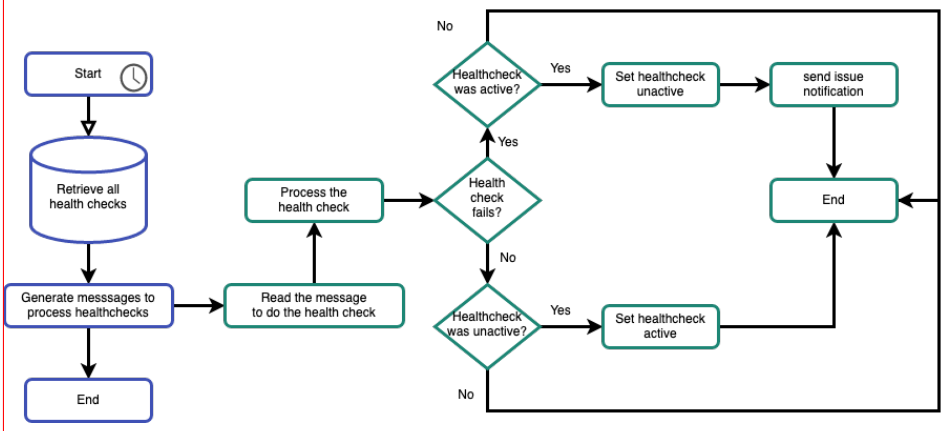

Figure 2: Flowchart of the periodic realizations. The blue elements are generated from a cron process while the green elements are executed by different workers, allowing the monitoring to be horizontally scalable.

4 Conclusions

The automatic incident monitoring website has been developed and allows multiple checks to be performed automatically on a regular basis. In addition, the code can be deployed on any company's private network and perform checks on points that are not available on the Internet, something that is not possible with commercial status pages such as Atlassian StatusPage [2].

5 Acknowledgments

- Financial support from the Xunta de Galicia and the European Union (European Social Fund -ESF), is gratefully acknowledged.
- CITIC, as Research Center accredited by Galician University System, is funded by "Consellería de Cultura, Educación e Universidades from Xunta de Galicia, Spain", supported in an 80% through ERDF Funds, Spain, ERDF Operational Programme Galicia 2014–2020, and the remaining 20% by "Secretaría Xeral de Universidades, Spain" (Grant ED431G 2019/01).
- We are particularly grateful to CESGA (Galician Supercomputing Centre) for providing access to their infrastructure.

References

[1] Amazon. Error retries and exponential backoff in aws. https://docs.aws.amazon.com/general/latest/gr/api-retries.html, last viewed September 2022.

[2] Atlassian. Statuspage. https://www.atlassian.com/software/statuspage, last viewed September 2022.

[3] Status Cacke. What happened to fastly and why did so many websites experience downtime? https://www.statuscake.com/blog/fastly-global-outage/, last viewed September 2021.

[4] Stephen Elliot. Devops and the cost of downtime: Fortune 1000 best practice metrics quantified. *International Data Corporation (IDC)*, 2014.

[5] Xian Jun Hong, Hyun Sik Yang, and Young Han Kim. Performance analysis of restful api and rabbitmq for microservice web application. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 257–259. IEEE, 2018.

[6] Catherine Kell. Weighing the scales: recontextualization as horizontal scaling. *Globalization and Language in Contact: Scale, Migration and Communicative Practices. London: Continuum*, pages 252–274, 2009.