Faculty of Engineering and Applied Science

SOFE 4790U: Distributed Computing

Fall 2022

Dr. Mohamed El-darieby
TA: Ghadeer Abdelkader

Lab 3: Deploying a Circuit Breaking ambassador and a Function-as-a-Service (FaaS)

Lab Group 20

Adris Azimi (100704571),

Deepak Thangella (100695929),

Harris Athwal (100750935),

Krystian Rusin (100704636)

Discussion

Summarize the problem, solution, and requirements for the pattern given in part 1.

Constantly monitoring web applications and back-end services is required. This, however, is not as simple as it sounds when the system is running on the cloud. There are many factors and aspects of the systems that have to be constantly monitored to ensure smooth operation.

The solution to this problem is the health endpoint monitoring pattern. In simple terms, requests are sent to an endpoint, performing the necessary checks, and an indication of its status is returned.

Some requirements for this pattern include:

- Validating the response code.
- Checking the content of the response to detect errors, even when a 200 (OK) status code is returned.
- Validating the URL returned by the DNS lookup to ensure correct entries.
- Measuring the response time

Which of these requirements can be achieved by the procedures shown in parts 2 and 3?

Part 2 showcased the use of a circuit breaker. This feature can be implemented to achieve the requirement of monitoring for failures. A circuit breaker can actively watch for any failures and proactively act within the system to ensure the proper management of a failure.

Part 3 showcased the use of a decorator pattern. This feature can be used to proactively fix certain failures and problems with validation. If a function is missing any critical objects, the decorator pattern can append default values for the missing object, resulting in fewer failures in the system.

Design

Kubernetes provides persistent volumes. Why such a feature can be important? How to implement it? Provide an example in which persistent volumes are needed. Configure a YAML file to implement the example. Run it and test the creation of persistent volume and its ability to provide the required functionality within the example.

Why can such a feature be important?

The reason why kubernetes persistent volumes are important is because it provides persistent storage for containerized applications. These storages can retain the previously entered data so that pods can still have access to the stored data even after restarting an

application. This is useful because even if an application or pod has been shut down or restarted, the stored data would still be retained and retrieved upon restart. Normally, the stored data would be wiped out after an application or pod has been shut down and restarted, however persistent volumes prevent this.

How to implement it?

The persistent volume can be implemented by configuring a YAML file and creating the persistent volume by using the kubectl apply functionality command.

Provide an example in which persistent volumes are needed. Configure a YAML file to implement the example. Run it and test the creation of persistent volume and its ability to provide the required functionality within the example.

YAML file and testing available within the second audible video of configuring and implementing persistent volumes using GKE.

Deliverables

Group Submission Deliverable

https://drive.google.com/drive/folders/130fck6QFJTwtofSmb-Uw4GEDtfJMX_LS?usp=sharing

Individual Work

Deepak Thangella
https://drive.google.com/drive/folders/1sqiKXUk2C2LVjbO6ZRoQgrtGbF7vm6qt?usp=sharing

Adris Azimi
https://drive.google.com/drive/folders/130fck6QFJTwtofSmb-Uw4GEDtfJMX_LS?usp=sharing

Harris Athwal
https://drive.google.com/drive/folders/1T2sqxIEpUez1NIs-cuXMWp7bXLSR0Lq3?usp=share_link

Krystian Rusin
https://youtu.be/aeG8aim0qR8