



Faculty of Engineering and Applied Science

SOFE 4790U Distributed Systems

Lab 2: Deploying a Circuit Breaking ambassador and a Function-as-Service (FaaS)

Group 19

Sunil Tumkur 100620430

Monil Patel 100727400

William Robinson 100751756

Michael Metry 100747141

Group GitHub Link:

<https://github.com/sunilt4/Distributed-Systems/tree/main/Lab%203>

Part 1:

Read the following document (Health Endpoint Monitoring pattern - Azure Architecture Center | Microsoft Learn). Focus on the problem being solved by the pattern, how is it solved? and the requirements needed for the solution.

- Problem being solved by the pattern
 - Monitor services running in the cloud than it is to monitor on-premises services
 - Not having full control of the hosting environment, and the services typically depend on other services provided by platform vendors and others
 - Network Latency
 - Performance + Availability of the underlying compute and storage systems
 - Network bandwidth
- How is it solved?
 - Verify services by running regular intervals and ensuring the system perform correctly and ensures availability
- Requirements needed for the solution
 - The application should perform necessary checks, and return an indication of its status
 - The checks performed by the application or service in response to the health verification endpoint.
 - Analysis of the results by the tool or framework that performs the health verification check
 - Checking cloud storage/database for availability and response time
 - Checking other resources or services located in the application, or located elsewhere but used by the application
 - The latency or response time check is performed by the monitoring tool or framework

- Types of checks that can be performed by the monitoring tools:
 - Validating the response code
 - Measuring the response time of network latency and the duration that the application took to execute the request
 - Checking for the expiration of SSL certificates
 - Checking resources or services located outside the application

Part 2: Deploying a Circuit Breaker pattern as an ambassador

Video Link:

https://drive.google.com/file/d/167YSX4yPPIhTQqCA_4fMktm9-Bpwbz7t/view?usp=sharing

Part 3: Using the Decorator pattern to implement FaaS that adds default values and performs transformations to the input of an HTTP RESTful API

Video Link 1:

https://drive.google.com/file/d/167YSX4yPPIhTQqCA_4fMktm9-Bpwbz7t/view?usp=sharing

Video Link 2:

<https://drive.google.com/file/d/1qA0JGbDK2BgJug4gOIb5kLvRpyuSSXfj/view?usp=sharing>

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.

Persistent Volume is a piece of storage in the cluster that has been provisioned by an administrator or dynamically provisioned using Storage Classes[1]. They are volume plugins, but have a lifecycle independent of separate pods[1]. The API then captures the details of the implementation of the storage by utilizing a cloud-provider-specific storage system.

In Persistent Volume Claim, it is a request for a storage by a user[1]. Like a pod, it consumes node resources with varying properties that can include performance, security, availability, and so on. They then request specific size and access modes such as ReadWriteOnce, ReadOnlyMany, or ReadWriteMany [1].

```
mikhaelsaad331@cloudshell:~ (distributed-systems-lab03)$ kubectl apply -f volume.yaml
persistentvolume/task-pv-volume created
mikhaelsaad331@cloudshell:~ (distributed-systems-lab03)$ kubectl get pv task-pv-volume
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS   CLAIM   STORAGECLASS  REASON   AGE
task-pv-volume  10Gi      RWO           Retain          Available  manual  manual                22s
mikhaelsaad331@cloudshell:~ (distributed-systems-lab03)$ kubectl apply -f pv-claim.yaml
persistentvolumeclaim/task-pv-claim created
mikhaelsaad331@cloudshell:~ (distributed-systems-lab03)$ kubectl get pv task-pv-volume
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS   CLAIM           STORAGECLASS  REASON   AGE
task-pv-volume  10Gi      RWO           Retain          Bound    default/task-pv-claim  manual                103s
mikhaelsaad331@cloudshell:~ (distributed-systems-lab03)$
```

Video Link:

<https://drive.google.com/file/d/1Jn119MQhBBGFkVu21FAeNedpaEiEdpfk/view?usp=sharing>

References:

- [1] “Persistent volumes,” *Kubernetes*, 22-Sep-2022. [Online]. Available: <https://kubernetes.io/docs/concepts/storage/persistent-volumes/>. [Accessed: 29-Oct-2022].
- [2] “Configure a pod to use a persistentvolume for storage,” *Kubernetes*, 21-Sep-2022. [Online]. Available: <https://kubernetes.io/docs/tasks/configure-pod-container/configure-persistent-volume-storage/>. [Accessed: 29-Oct-2022].
- [3] EdPrice-MSFT, “Health Endpoint Monitoring Pattern - Azure Architecture Center,” *Azure Architecture Center | Microsoft Learn*. [Online]. Available: <https://learn.microsoft.com/en-us/azure/architecture/patterns/health-endpoint-monitoring>. [Accessed: 29-Oct-2022].