Sistemas de Computação em Cloud / Cloud Computing Systems

2024/2025

Project Assignment 2

Autores: Daniel Pojega, 71911

Tiago Neutel, 71903

09 dezembro 2024

Índice

1. Authentication

To deal with user authentication for the Blobs Services, we took advantage of our already implemented *createUser* endpoint. Upon successfully creating a user, a cookie is issued which will be included in subsequent requests, where it will be of use to endpoints where user session verification is required.

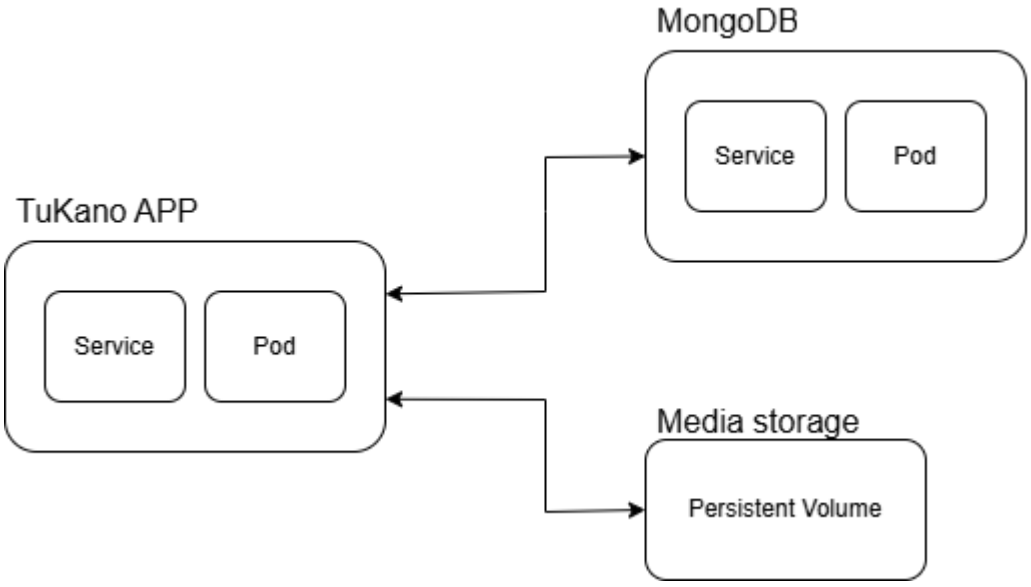
This cookie will then be stored locally in the respective thread handling the request, which can then be accessed from any point of the code.

For each request where user authentication is required, a call to the *validateSession()* function is made where the locally store cookie is validated, checking the user value stored in cache and comparing its Id to the userId passed as parameter. If values do not match, a 401 Unauthorized error is returned. If not, the request proceeds as usual.

2. Azure Kubernetes deployments

Prior to defining the deployments required, a Dockerfile was defined with the purpose of deploying our application in the Kubernetes environment. To achieve this, we built a container image of our application to expose the packaged version of our application (tukano.war) to the port 8080, where it will listen to HTTP requests.

Furthermore, to help visualize the architecture of the TuKano application and the corresponding services within the cluster, the following diagram was drawn:



2.1. TuKano Application

The deployment of the TuKano Application is defined in the tukano-app.yaml file, where we advantage of the containerized image previously created (a49521/tukano-sv:v1) to deploy the app.

Besides that, the PersistentVolumeClaim responsible for the blob media storage is mounted inside the container, ensuring persistence for the blobs. In addition, to connect to the MongoDB pod, we have defined an environment variable pointed to the instance URL.

Finally, to expose the application externally, a LoadBalancer service which listens on port 80 is created. The external-ip address that originates from this service allows users to connect to the application inside the cluster. Example: http://4.207.32.107:80/...

2.2. MongoDB Database

In relation to the database deployment, we opted for the NoSQL version which involved creating a MongoDB version of our database implementation, which previously was adapted to support CosmosDB.

Consequently, we set up a MongoDB instance in Kubernetes defining the main pod, the service to connect to the TuKano application pod and a PersistentVolumeClaim to ensure data persistence. Besides that, the MongoDB container uses an official MongoDB image and is exposed internally on the port 27017 within the cluster.

The Service, as previously mentioned, has the purpose of exposing the database internally within the cluster, allowing it to connect to the TuKano application and to forward traffic.

2.3. Persistent volume for blob storage

This yaml file specifies the PersistentVolume and PersistentVolumeClaim configurations to provide data persistence to the blobs.

3. Performance evaluation

3.1. Tests

In this section, we present a performance evaluation based on the tests provided by the professor on the project 1, for a better comparison. Initially, we used Postman to verify that each endpoint functioned as expected, with all tests passing successfully. Following these tests, we conducted performance testing with Artillery.

This subsection presents a table comparing the Average Time for each test, highlighting the performance differences between the current solution and the solution from Project 1.

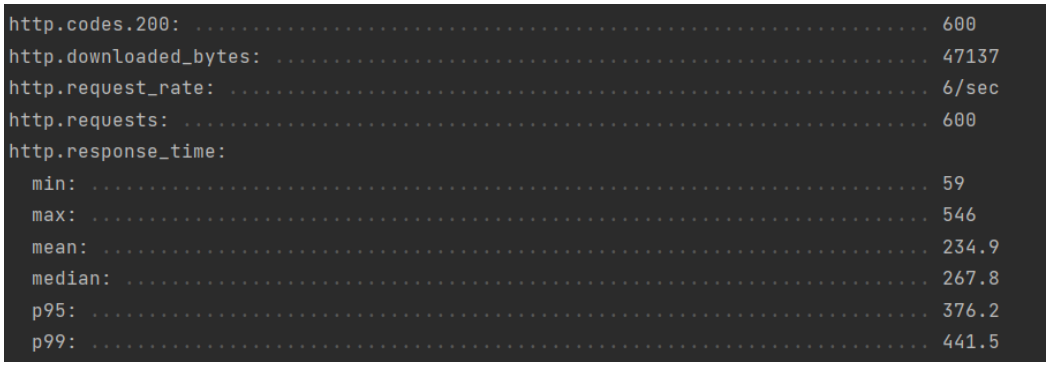
Tests	Average Time Project 1 (ms)	Average Time Project 2 (ms)
User_register.yaml	234.9	161.2
User_delete.yaml	268.5	113.8
Upload_shorts.yaml	1022.9	504
Realistic_flow.yaml	622.2	248.1

By analyzing the table, we can conclude that the solution that leverages Azure's IaaS portfolio improves the response time of any HTTP request by cutting nearly half the average time needed in the first solution.

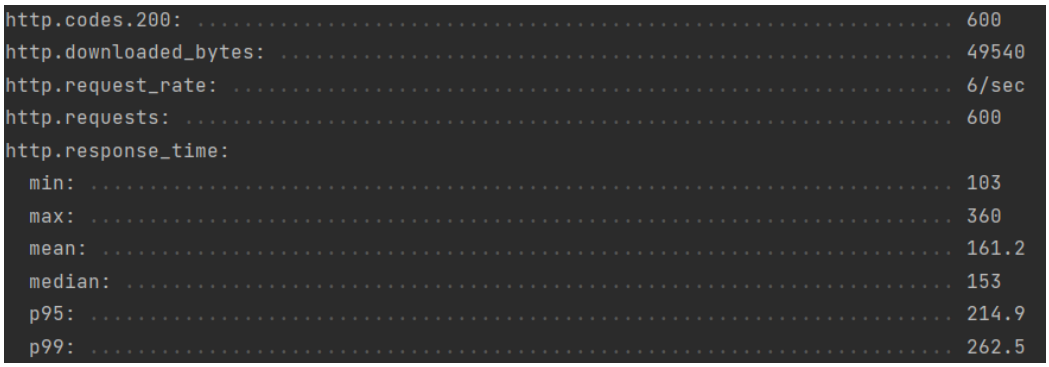
3.2. Screenshots of both solutions

User_register.yaml:

Project 1:



Project 2:



Upload_shorts.yaml:

Project 1:

```
http.codes.200: ..... 175
http.codes.204: ..... 75
http.codes.401: ..... 25
http.codes.409: ..... 25
http.downloaded_bytes: ..... 25044
http.request_rate: ..... 8/sec
http.requests: ..... 300
http.response_time:
  min: ..... 2
  max: ..... 2255
  mean: ..... 1022.9
  median: ..... 1064.4
  p95: ..... 1863.5
  p99: ..... 2059.5
```

Project 2:

```
http.codes.200: ..... 200
http.codes.204: ..... 100
http.downloaded_bytes: ..... 27194
http.request_rate: ..... 8/sec
http.requests: ..... 300
http.response_time:
  min: ..... 164
  max: ..... 1164
  mean: ..... 504.4
  median: ..... 424.2
  p95: ..... 1022.7
  p99: ..... 1153.1
```

Realistic_flow.yaml:

Project 1:

```
http.codes.200: ..... 6
http.codes.404: ..... 5
http.codes.500: ..... 10
http.downloaded_bytes: ..... 555
http.request_rate: ..... 5/sec
http.requests: ..... 21
http.response_time:
  min: ..... 2
  max: ..... 1293
  mean: ..... 622.2
  median: ..... 518.1
  p95: ..... 1249.1
  p99: ..... 1249.1
```

Project 2:

```
errors.ECONNRESET: ..... 1
errors.ETIMEDOUT: ..... 13
http.codes.200: ..... 4
http.codes.204: ..... 3
http.codes.401: ..... 3
http.downloaded_bytes: ..... 8
http.request_rate: ..... 3/sec
http.requests: ..... 24
http.response_time:
  min: ..... 49
  max: ..... 1280
  mean: ..... 248.1
  median: ..... 159.2
  p95: ..... 214.9
  p99: ..... 214.9
```

User_delete.yaml:

Project 1:

```
http.codes.204: ..... 200
http.downloaded_bytes: ..... 0
http.request_rate: ..... 2/sec
http.requests: ..... 200
http.response_time:
  min: ..... 249
  max: ..... 445
  mean: ..... 268.5
  median: ..... 262.5
  p95: ..... 284.3
  p99: ..... 347.3
```

Project 2:

```
http.codes.204: ..... 200
http.downloaded_bytes: ..... 0
http.request_rate: ..... 2/sec
http.requests: ..... 200
http.response_time:
  min: ..... 104
  max: ..... 202
  mean: ..... 113.8
  median: ..... 111.1
  p95: ..... 159.2
  p99: ..... 186.8
```