NOVA SCHOOL OF SCIENCE & TECHNOLOGY

Sistemas de Computação em Cloud

2024/2025

Project 2
TuKano - Part 2

Authors:

70525, Francisco Silva
70596, Martim Latas

Practical class Nº 3

December 21st, 2024

Introduction to the Project

In this second part of the project, we aim to adapt and deploy the version of *TuKano* developed in the first assignment using Docker and Kubernetes, leveraging Azure’s IaaS capabilities. The challenge is to replace the Azure PaaS services used previously with Kubernetes-based alternatives, while maintaining the original functionalities.

This project is composed of four dockers, whose communication will be explained later. These dockers have distinct functionalities: one for the Tukano main application, another for the Blob storage (with Authentication), a docker for the Redis Cache and another one for the Postgres (DataBase of the application).

Two of these are from public images, the Cache and Postgres, which were imported from hub.docker.com. The last two were created by us and uploaded (after building) to the same platform.

We will deploy the application server and the Hibernate + SQL database using Azure Kubernetes Service, ensuring scalability and performance. Additionally, persistent volumes will be set up to replace Azure Blob Storage for media data storage, allowing continuity in user uploads. By using Kubernetes, we will manage the entire infrastructure, enabling greater flexibility and control over our deployment.

In this project, we added authentication and access control to the *TuKano* application, running in a separate container for modularity.

Users log in through an endpoint, where their credentials are verified, and a session cookie is issued. This cookie is used to track authenticated users in later requests. Session data is stored in a Redis-like system to maintain consistency across instances.

The cookie itself is stored in a Map located in the Local Thread, meaning that every user connected to this system would have his own dedicated thread.

Access control would ensure only authorized users could perform certain actions. For example, in this system, only authenticated users can upload or download blobs, while only 'admin' users can delete said blobs (which is not something that should occur in the real world).

We also used a filter to manage the cookies and authentication on this system, like it was taught in class.

We used Artillery to test the application, like the last project, but since session data is stored in thread-local storage, it couldn't be accessed during testing. To work around this, we simplified the login and authentication process for testing purposes, allowing every request to be treated as valid.

This ensured the tests could run successfully without affecting the core authentication logic used in production. This might have some impact on performance, bettering the results, even if only a little.

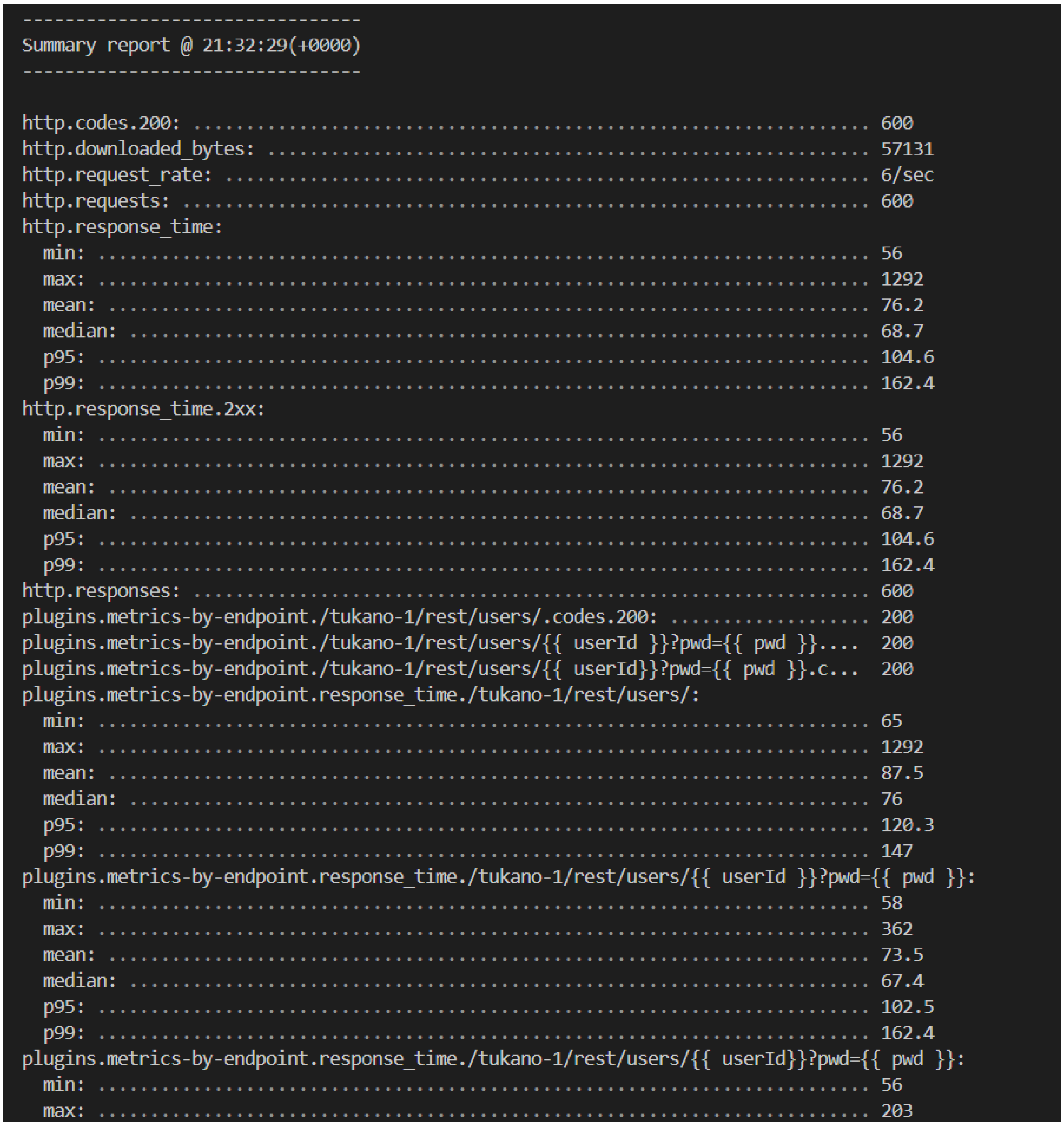
This project is separated in different containers, all communicating thanks to the EXPOSE instruction. Then, we have Kubernetes yaml files that instruct how each is deployed, with some environmental variables in order to communicate with other hosts.

Performance Analysis via Artillery

For the performance test, we once again used the Artillery testing platform, after performing the changes mentioned in the introduction above.

As for the test's content, we decided to use the same tests as the previous project, as the main functionalities of our system didn't change from the previous project.

Register Users



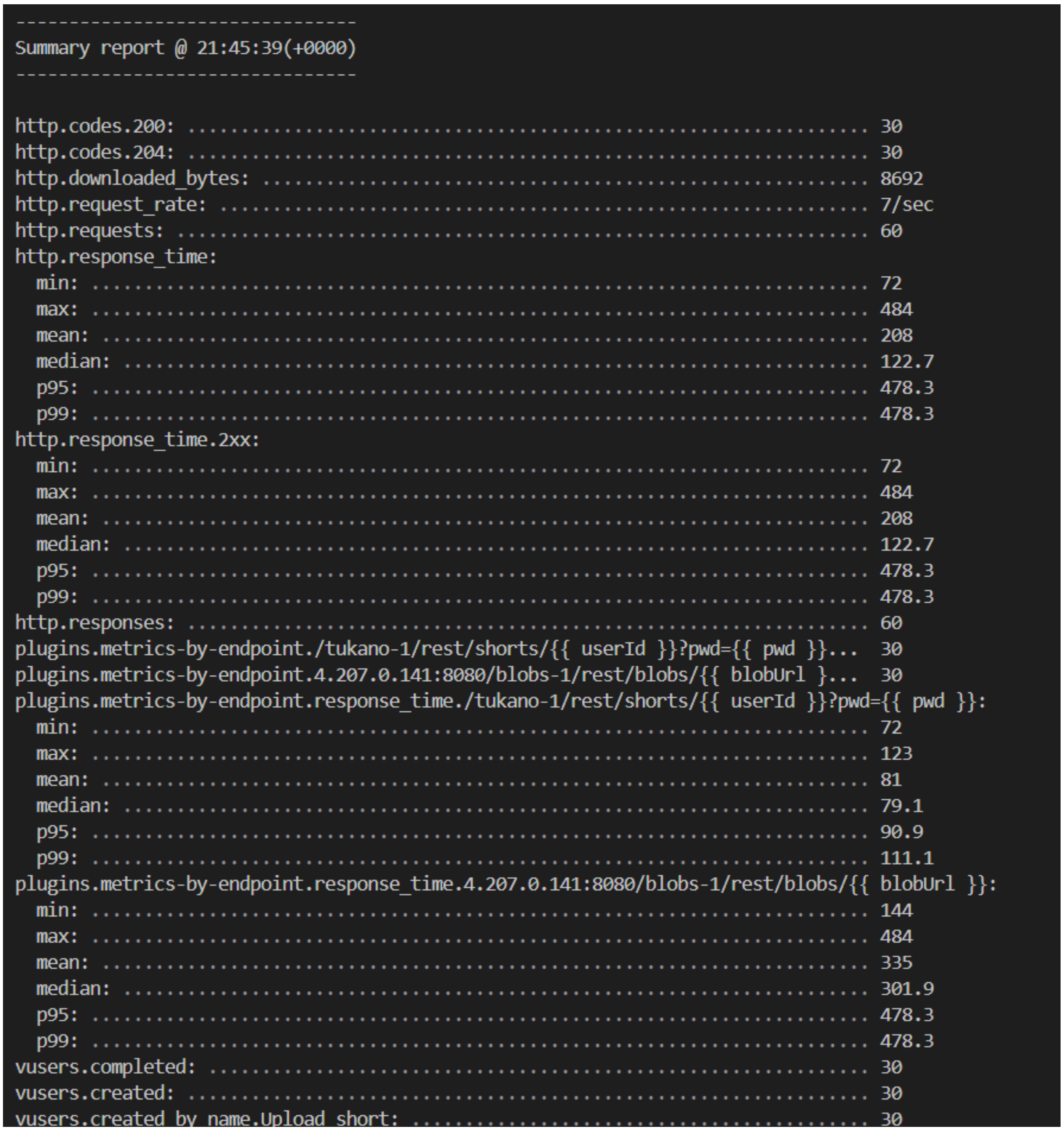
```
--------------------------------
Summary report @ 21:32:29(+0000)
--------------------------------

http.codes.200: ................................................................ 600
http.downloaded_bytes: ......................................................... 57131
http.request_rate: ............................................................. 6/sec
http.requests: ................................................................. 600
http.response_time:
  min: ......................................................................... 56
  max: ......................................................................... 1292
  mean: ........................................................................ 76.2
  median: ...................................................................... 68.7
  p95: ......................................................................... 104.6
  p99: ......................................................................... 162.4
http.response_time.2xx:
  min: ......................................................................... 56
  max: ......................................................................... 1292
  mean: ........................................................................ 76.2
  median: ...................................................................... 68.7
  p95: ......................................................................... 104.6
  p99: ......................................................................... 162.4
http.responses: ................................................................ 600
plugins.metrics-by-endpoint./tukano-1/rest/users/.codes.200: ................... 200
plugins.metrics-by-endpoint./tukano-1/rest/users/{{ userId }}?pwd={{ pwd }}....  200
plugins.metrics-by-endpoint./tukano-1/rest/users/{{ userId}}?pwd={{ pwd }}.c...  200
plugins.metrics-by-endpoint.response_time./tukano-1/rest/users/:
  min: ......................................................................... 65
  max: ......................................................................... 1292
  mean: ........................................................................ 87.5
  median: ...................................................................... 76
  p95: ......................................................................... 120.3
  p99: ......................................................................... 147
plugins.metrics-by-endpoint.response_time./tukano-1/rest/users/{{ userId }}?pwd={{ pwd }}:
  min: ......................................................................... 58
  max: ......................................................................... 362
  mean: ........................................................................ 73.5
  median: ...................................................................... 67.4
  p95: ......................................................................... 102.5
  p99: ......................................................................... 162.4
plugins.metrics-by-endpoint.response_time./tukano-1/rest/users/{{ userId}}?pwd={{ pwd }}:
  min: ......................................................................... 56
  max: ......................................................................... 203
```

The minimum response time of 56 ms highlights the system's capability to handle basic user creation operations with minimal latency. However, the maximum response time reached 1292 ms, indicating a notable outlier that may require additional investigation into potential bottlenecks during peak resource usage.

The 95th and 99th percentile response times (104.6 ms and 162.4 ms, respectively) suggest that while the system generally handles requests efficiently, occasional high-latency cases exist.

Overall, the system demonstrated robust performance for user creation operations, efficiently handling requests with minimal latency for most cases.

Upload Shorts



```
--------------------------------
Summary report @ 21:45:39(+0000)
--------------------------------

http.codes.200: ................................................................ 30
http.codes.204: ................................................................ 30
http.downloaded_bytes: ......................................................... 8692
http.request_rate: ............................................................. 7/sec
http.requests: ................................................................. 60
http.response_time:
  min: ......................................................................... 72
  max: ......................................................................... 484
  mean: ........................................................................ 208
  median: ...................................................................... 122.7
  p95: ......................................................................... 478.3
  p99: ......................................................................... 478.3
http.response_time.2xx:
  min: ......................................................................... 72
  max: ......................................................................... 484
  mean: ........................................................................ 208
  median: ...................................................................... 122.7
  p95: ......................................................................... 478.3
  p99: ......................................................................... 478.3
http.responses: ................................................................ 60
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId }}?pwd={{ pwd }}...  30
plugins.metrics-by-endpoint.4.207.0.141:8080/blobs-1/rest/blobs/{{ blobUrl }}...  30
plugins.metrics-by-endpoint.response_time./tukano-1/rest/shorts/{{ userId }}?pwd={{ pwd }}:
  min: ......................................................................... 72
  max: ......................................................................... 123
  mean: ........................................................................ 81
  median: ...................................................................... 79.1
  p95: ......................................................................... 90.9
  p99: ......................................................................... 111.1
plugins.metrics-by-endpoint.response_time.4.207.0.141:8080/blobs-1/rest/blobs/{{ blobUrl }}:
  min: ......................................................................... 144
  max: ......................................................................... 484
  mean: ........................................................................ 335
  median: ...................................................................... 301.9
  p95: ......................................................................... 478.3
  p99: ......................................................................... 478.3
vusers.completed: .............................................................. 30
vusers.created: ................................................................ 30
vusers.created_by_name.Upload short: ........................................... 30
```

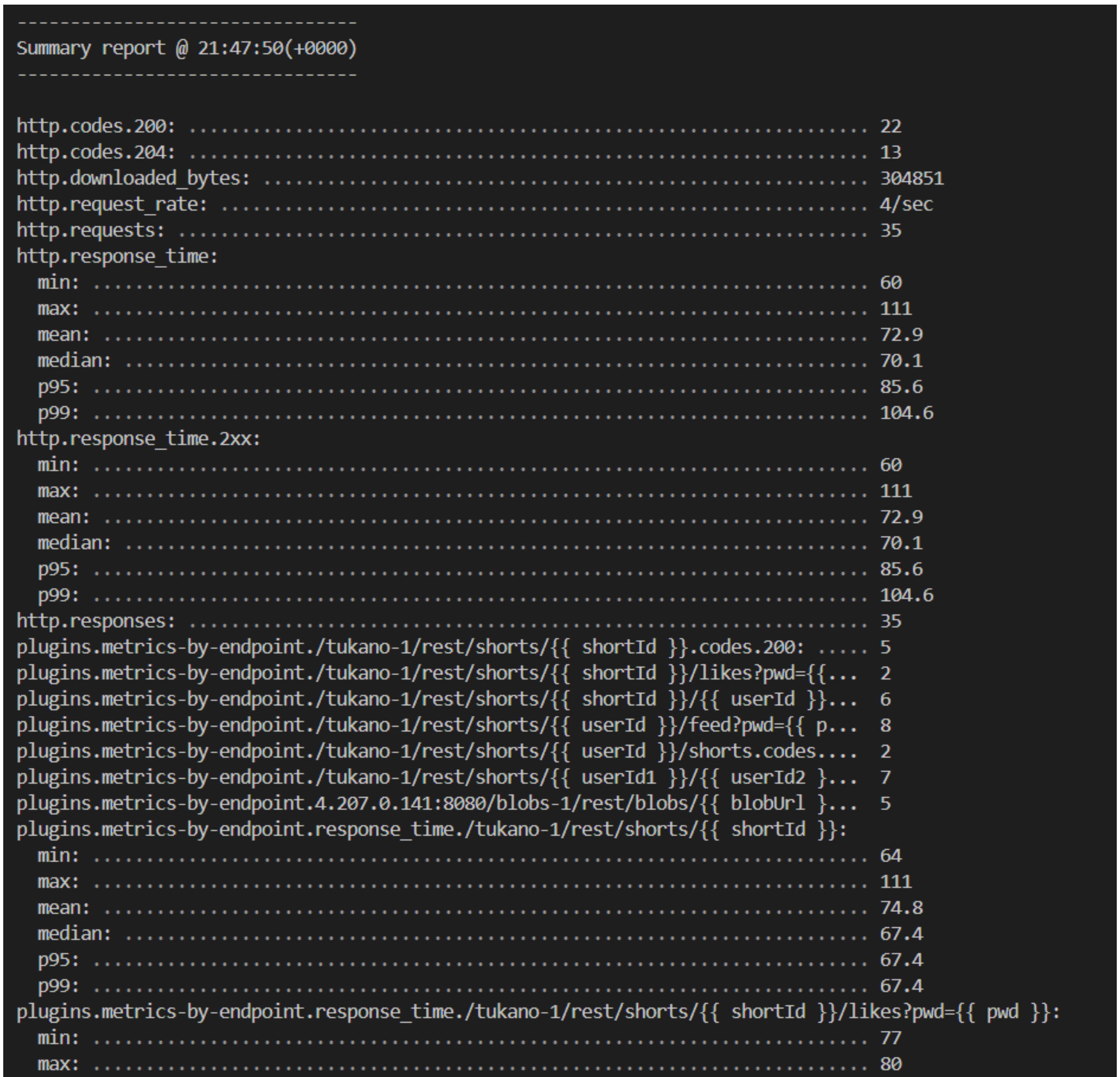
In this test, which focused on creating and uploading shorts alongside their corresponding blobs achieving a combined request rate of 7 requests per second.

The average response time for all requests was 208 ms, with a median response time of 122.7 ms, suggesting consistent performance across the majority of requests. The minimum response time of 72 ms reflects the baseline efficiency for these operations, while the maximum response time of 484 ms indicates occasional high-latency outliers. The 95th and 99th percentile response times, both at 478.3 ms, further highlight the presence of these higher-latency cases

The disparity in response times between these endpoints suggests that the blob upload process is more resource-intensive compared to the creation of shorts, which is to be expected. Still, the absence of failed requests demonstrates the system’s reliability in handling the operations.

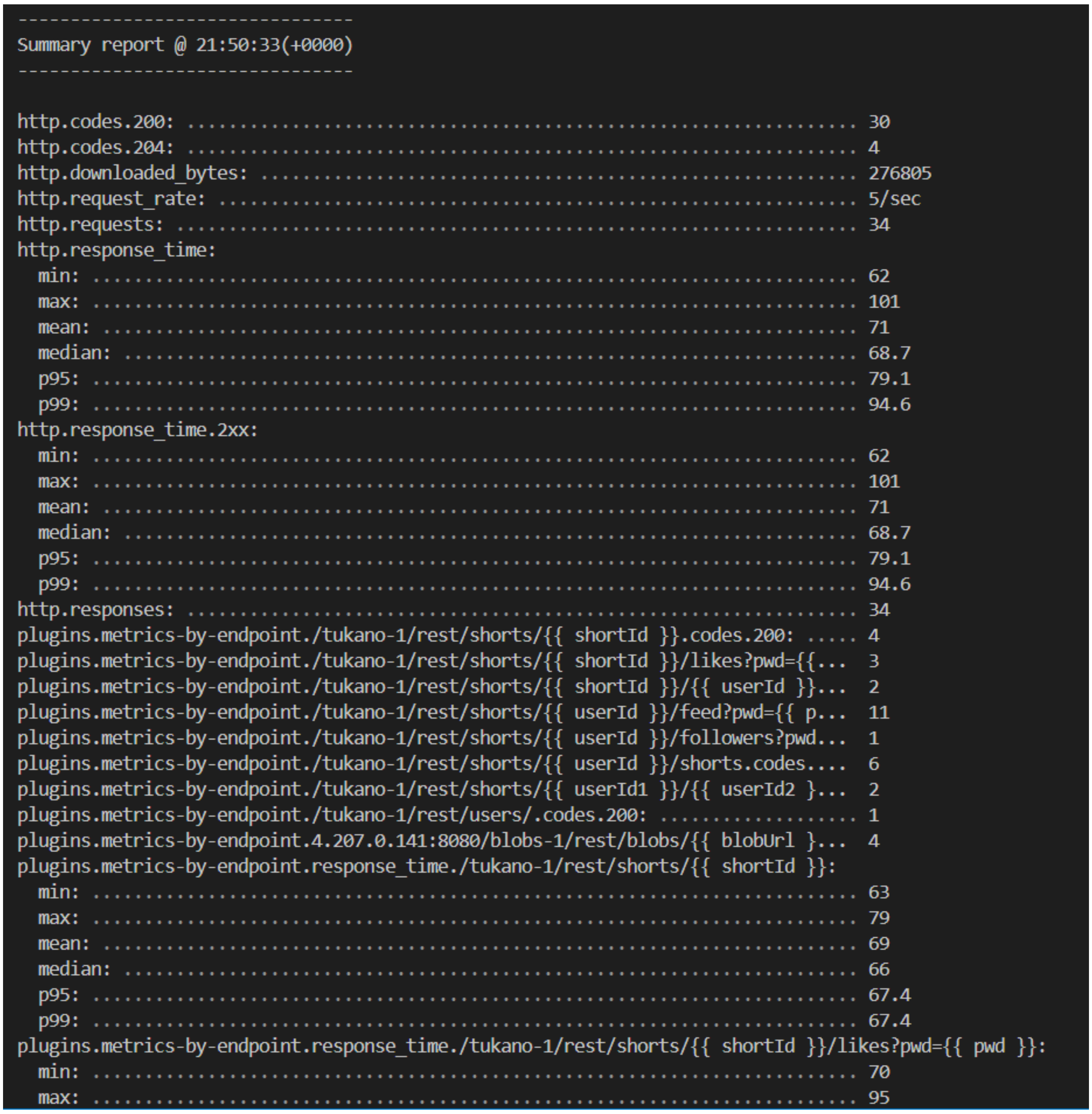
Realistic Flow

First Flow -



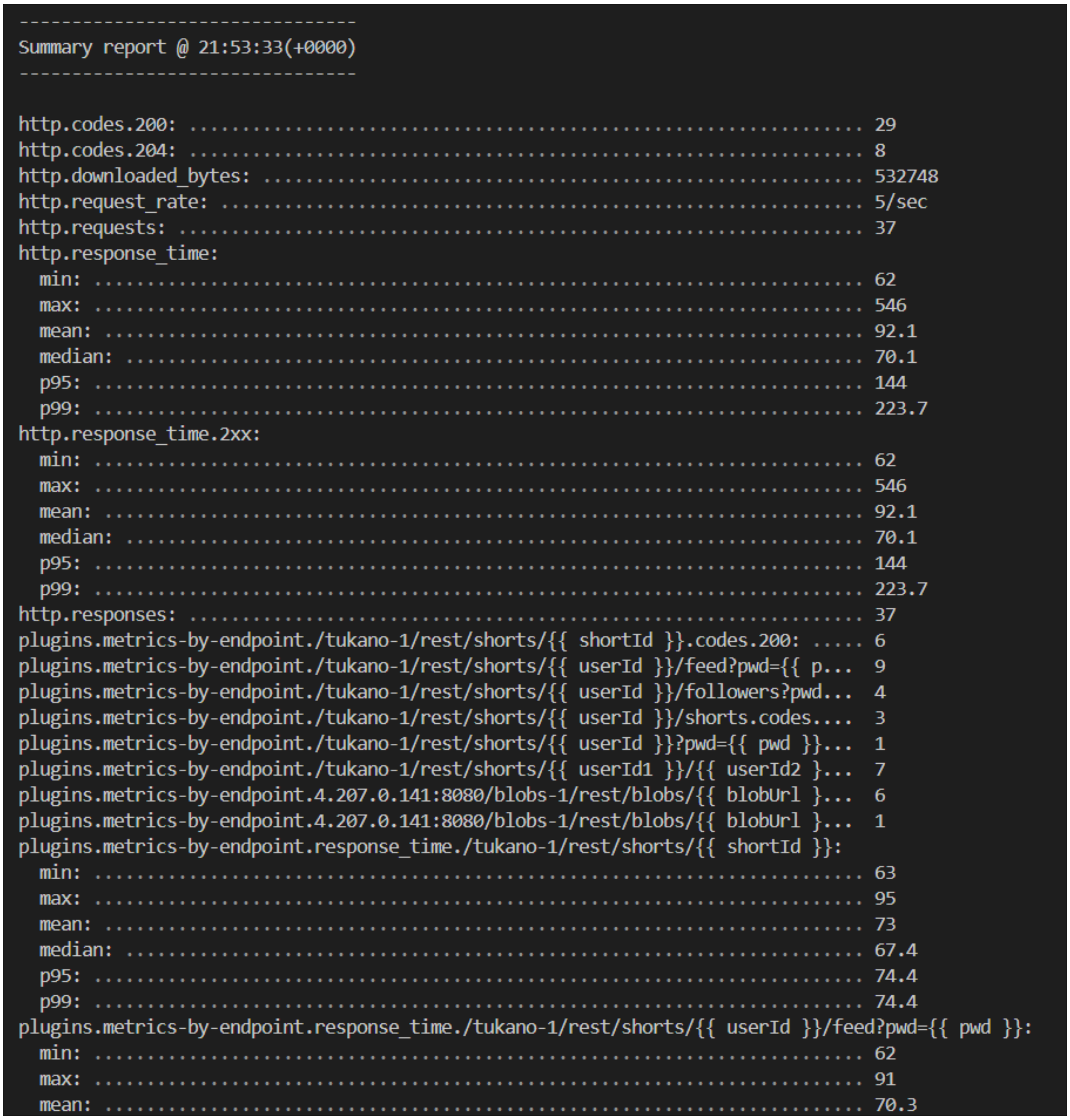
```
--------------------------------
Summary report @ 21:47:50(+0000)
--------------------------------

http.codes.200: ................................................................ 22
http.codes.204: ................................................................ 13
http.downloaded_bytes: ......................................................... 304851
http.request_rate: ............................................................. 4/sec
http.requests: ................................................................. 35
http.response_time:
  min: ......................................................................... 60
  max: ......................................................................... 111
  mean: ........................................................................ 72.9
  median: ...................................................................... 70.1
  p95: ......................................................................... 85.6
  p99: ......................................................................... 104.6
http.response_time.2xx:
  min: ......................................................................... 60
  max: ......................................................................... 111
  mean: ........................................................................ 72.9
  median: ...................................................................... 70.1
  p95: ......................................................................... 85.6
  p99: ......................................................................... 104.6
http.responses: ................................................................ 35
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ shortId }}.codes.200: ..... 5
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ shortId }}/likes?pwd={{...  2
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ shortId }}/{{ userId }}...  6
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId }}/feed?pwd={{ p...  8
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId }}/shorts.codes....  2
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId1 }}/{{ userId2 }...  7
plugins.metrics-by-endpoint.4.207.0.141:8080/blobs-1/rest/blobs/{{ blobUrl }...  5
plugins.metrics-by-endpoint.response_time./tukano-1/rest/shorts/{{ shortId }}:
  min: ......................................................................... 64
  max: ......................................................................... 111
  mean: ........................................................................ 74.8
  median: ...................................................................... 67.4
  p95: ......................................................................... 67.4
  p99: ......................................................................... 67.4
plugins.metrics-by-endpoint.response_time./tukano-1/rest/shorts/{{ shortId }}/likes?pwd={{ pwd }}:
  min: ......................................................................... 77
  max: ......................................................................... 80
```

Second Flow -



```
--------------------------------
Summary report @ 21:50:33(+0000)
--------------------------------

http.codes.200: ................................................................ 30
http.codes.204: ................................................................ 4
http.downloaded_bytes: ......................................................... 276805
http.request_rate: ............................................................. 5/sec
http.requests: ................................................................. 34
http.response_time:
  min: ......................................................................... 62
  max: ......................................................................... 101
  mean: ........................................................................ 71
  median: ...................................................................... 68.7
  p95: ......................................................................... 79.1
  p99: ......................................................................... 94.6
http.response_time.2xx:
  min: ......................................................................... 62
  max: ......................................................................... 101
  mean: ........................................................................ 71
  median: ...................................................................... 68.7
  p95: ......................................................................... 79.1
  p99: ......................................................................... 94.6
http.responses: ................................................................ 34
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ shortId }}.codes.200: ..... 4
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ shortId }}/likes?pwd={{... 3
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ shortId }}/{{ userId }}... 2
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId }}/feed?pwd={{ p... 11
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId }}/followers?pwd... 1
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId }}/shorts.codes.... 6
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId1 }}/{{ userId2 }... 2
plugins.metrics-by-endpoint./tukano-1/rest/users/.codes.200: ................... 1
plugins.metrics-by-endpoint.4.207.0.141:8080/blobs-1/rest/blobs/{{ blobUrl }... 4
plugins.metrics-by-endpoint.response_time./tukano-1/rest/shorts/{{ shortId }}:
  min: ......................................................................... 63
  max: ......................................................................... 79
  mean: ........................................................................ 69
  median: ...................................................................... 66
  p95: ......................................................................... 67.4
  p99: ......................................................................... 67.4
plugins.metrics-by-endpoint.response_time./tukano-1/rest/shorts/{{ shortId }}/likes?pwd={{ pwd }}:
  min: ......................................................................... 70
  max: ......................................................................... 95
```

Third Flow -



```
--------------------------------
Summary report @ 21:53:33(+0000)
--------------------------------

http.codes.200: ............................................................... 29
http.codes.204: ............................................................... 8
http.downloaded_bytes: ........................................................ 532748
http.request_rate: ............................................................ 5/sec
http.requests: ................................................................ 37
http.response_time:
  min: ........................................................................ 62
  max: ........................................................................ 546
  mean: ....................................................................... 92.1
  median: ..................................................................... 70.1
  p95: ........................................................................ 144
  p99: ........................................................................ 223.7
http.response_time.2xx:
  min: ........................................................................ 62
  max: ........................................................................ 546
  mean: ....................................................................... 92.1
  median: ..................................................................... 70.1
  p95: ........................................................................ 144
  p99: ........................................................................ 223.7
http.responses: ............................................................... 37
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ shortId }}.codes.200: ..... 6
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId }}/feed?pwd={{ p... 9
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId }}/followers?pwd... 4
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId }}/shorts.codes.... 3
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId }}?pwd={{ pwd }}... 1
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId1 }}/{{ userId2 }... 7
plugins.metrics-by-endpoint.4.207.0.141:8080/blobs-1/rest/blobs/{{ blobUrl }... 6
plugins.metrics-by-endpoint.4.207.0.141:8080/blobs-1/rest/blobs/{{ blobUrl }... 1
plugins.metrics-by-endpoint.response_time./tukano-1/rest/shorts/{{ shortId }}:
  min: ........................................................................ 63
  max: ........................................................................ 95
  mean: ....................................................................... 73
  median: ..................................................................... 67.4
  p95: ........................................................................ 74.4
  p99: ........................................................................ 74.4
plugins.metrics-by-endpoint.response_time./tukano-1/rest/shorts/{{ userId }}/feed?pwd={{ pwd }}:
  min: ........................................................................ 62
  max: ........................................................................ 91
  mean: ....................................................................... 70.3
```

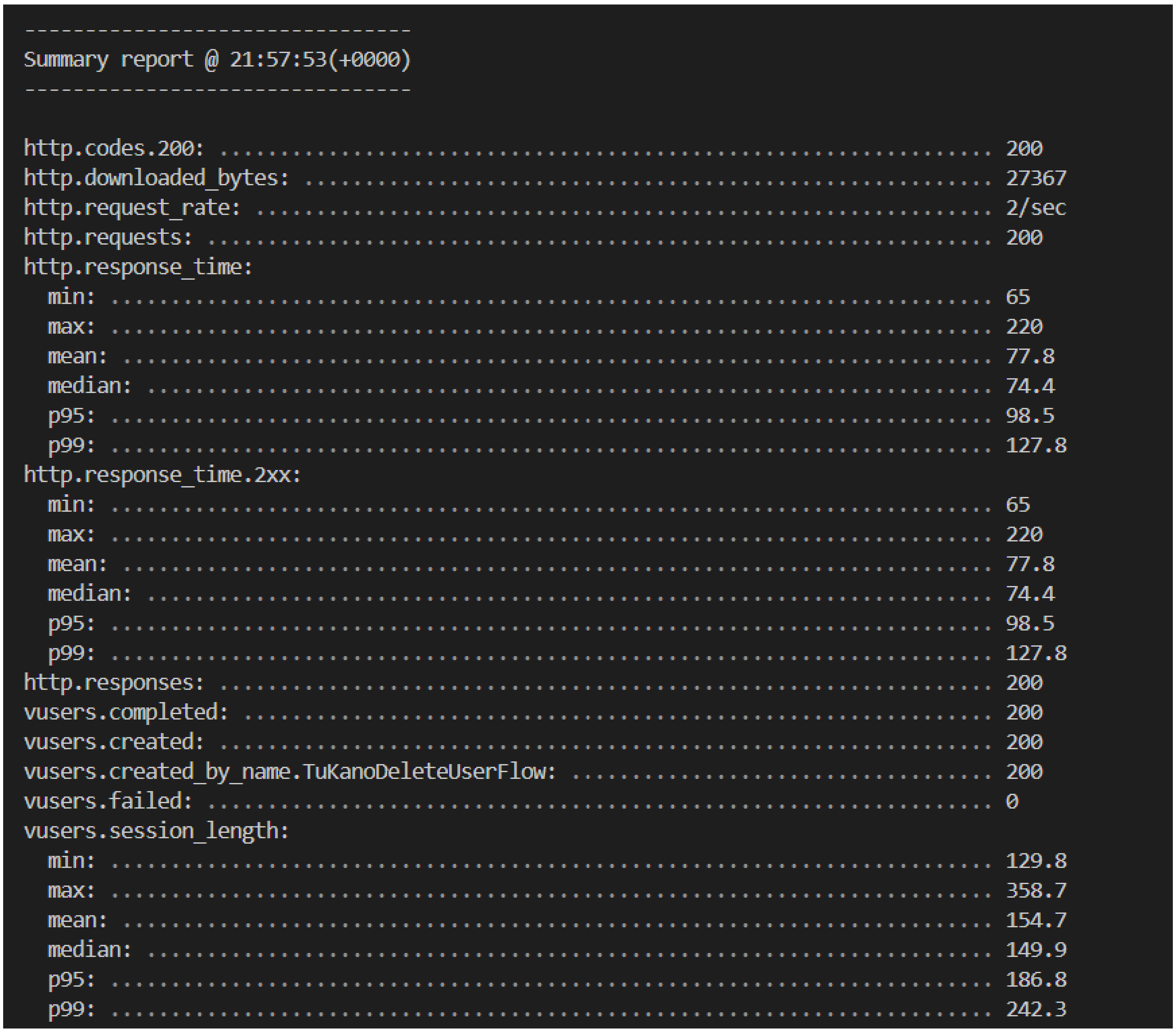
In the three performance tests conducted, the request rate achieved a consistent 4-5 requests per second, showcasing the system's ability to handle moderate traffic effectively. The steady request rate across the tests indicates that the Kubernetes deployment is well-configured to manage a stable load.

The average response time remains fairly consistent, ranging from 70 ms to 92 ms across the tests, reflecting reliable system performance. The highest response time varied, with a notable spike in the third test (546 ms). This increase in maximum response time may point to a transient bottleneck or outlier.

The steady mean response time for responses with a 200 status code further highlights the reliability of the application under normal conditions. The 95th and 99th percentile response times exhibited slight increases, particularly in the third test, but remained within acceptable limits. This suggests that the system is capable of handling most requests efficiently, with only occasional latency outliers affecting peak demand scenarios.

Certain endpoints, such as ‘/tukano-1/rest/shorts/{{shortId}}/feed’, experienced slightly higher latencies, which is expected to be one of the more resource demanding methods, but still indicating potential areas for optimization, such as query tuning. Overall, the use of persistent storage and Redis caching contributed to the system's ability to maintain stable performance, reducing latency for frequently accessed resources.

Delete Users



```
--------------------------------
Summary report @ 21:57:53(+0000)
--------------------------------

http.codes.200: ................................................................ 200
http.downloaded_bytes: ......................................................... 27367
http.request_rate: ............................................................. 2/sec
http.requests: ................................................................. 200
http.response_time:
  min: ......................................................................... 65
  max: ......................................................................... 220
  mean: ........................................................................ 77.8
  median: ...................................................................... 74.4
  p95: ......................................................................... 98.5
  p99: ......................................................................... 127.8
http.response_time.2xx:
  min: ......................................................................... 65
  max: ......................................................................... 220
  mean: ........................................................................ 77.8
  median: ...................................................................... 74.4
  p95: ......................................................................... 98.5
  p99: ......................................................................... 127.8
http.responses: ................................................................ 200
vusers.completed: .............................................................. 200
vusers.created: ................................................................ 200
vusers.created_by_name.TuKanoDeleteUserFlow: .................................. 200
vusers.failed: ................................................................. 0
vusers.session_length:
  min: ......................................................................... 129.8
  max: ......................................................................... 358.7
  mean: ........................................................................ 154.7
  median: ...................................................................... 149.9
  p95: ......................................................................... 186.8
  p99: ......................................................................... 242.3
```

In this test, focusing solely on removing users generated from earlier tests, the system exhibited stable performance, achieving a steady request rate of 2 requests per second and a total of 200 successful deletions (HTTP 200). This suggests that the system is efficiently handling delete operations without any errors or issues.

The typical response time was recorded at 77.8 ms, while the median response time was similarly positioned at 74.4 ms. The minimum response time was 65 ms. However, the peak response time hit 220 ms, which, while still acceptable, indicates sporadic spikes probably caused by temporary issues like resource contention or database latency.

The 95th and 99th percentile response times (98.5 ms and 127.8 ms, respectively) indicate that the system can handle most deletion requests promptly, with only a few outliers exhibiting slightly higher latency.

In general, the test validates the system's capacity to handle user deletion tasks reliably, with periodic latency increases that are manageable within the existing infrastructure. This shows that the Kubernetes deployment and database setup are effectively supporting the application’s deletion processes.

Conclusion

By extending the capabilities of our project, we can conclude how an application with these types of services, like Authentication and Session management, would operate within a Kubernetes-managed infrastructure, through the communication of different image containers.

Our performance testing showed that even with simplified authentication for testing purposes, the system handled requests efficiently, reflecting the benefits of Kubernetes. These enhancements make TuKano more robust, scalable, and closer to real-world applications that require secure access to data.