Project SCC 2024

Report 2

Work done by:

João Lima nº 60350

Diogo Nunes nº 70502

Motivation

It was given to us the task to adapt and deploy the version of Tukano developed in the context of the first project assignment, using Docker and Kubernetes, as IaaS facilities provided by Azure. This report's objective is to identify in what ways the goal was achieved, and explaining how it leverages the Azure IaaS portfolio, by performing an evaluation of the performance.

What was achieved

In our project, we implemented:

- Deployment of application server in Minikube
- Deployment of PostgreSQL database in Minikube
- Deployment of Persistent Volume in Minikube
- Testing with artillery
- Deployment of caching service (Redis) in Minikube

Note: We ran out of funds in Azure, so we had to test the entire project with Minikube, that is why nothing is deployed in Azure.

Impact on TuKano

We will be looking at the results from both assignments, comparing them:

user_register

- Azure

```
http.codes.400: ..... 400
http.codes.404: ..... 200
http.downloaded_bytes: ..... 15200
http.request_rate: ..... 6/sec
http.requests: ..... 600
http.response_time:
  min: ..... 2
  max: ..... 3446
  mean: ..... 94
  median: ..... 3
  p95: ..... 210.6
  p99: ..... 820.7
http.response_time.4xx:
  min: ..... 2
  max: ..... 3446
```

  mean: 94
  median: 3
  p95: 210.6
  p99: 820.7
http.responses: 600
plugins.metrics-by-endpoint./tukano/rest/users/.codes.400: 200
plugins.metrics-by-endpoint./tukano/rest/users/{{ userId }}?pwd={{ pwd }}.co... 200
plugins.metrics-by-endpoint./tukano/rest/users/{{ userId}}?pwd={{ pwd }}.cod... 200
plugins.metrics-by-endpoint.response_time./tukano/rest/users/:
  min: 2
  max: 356
  mean: 5
  median: 3
  p95: 4
  p99: 7
plugins.metrics-by-endpoint.response_time./tukano/rest/users/{{ userId }}?pwd={{ pwd }}:
  min: 2
  max: 10
  mean: 2.8
  median: 3
  p95: 4
  p99: 6
plugins.metrics-by-endpoint.response_time./tukano/rest/users/{{ userId}}?pwd={{ pwd }}:
  min: 181
  max: 3446
  mean: 274.3
  median: 198.4
  p95: 278.7
  p99: 2836.2
vusers.completed: 200
vusers.created: 200
vusers.created_by_name.TuKanoWholeUserFlow: 200
vusers.failed: 0
vusers.session_length:
  min: 192.8
  max: 3843.4
  mean: 287.9
  median: 206.5
  p95: 295.9
  p99: 2836.2

- Minikube

http.codes.200: 600
http.downloaded_bytes: 47137
http.request_rate: 6/sec
http.requests: 600

```
http.response_time:
  min: ..... 2
  max: ..... 309
  mean: ..... 7.4
  median: ..... 6
  p95: ..... 10.9
  p99: ..... 43.4
http.response_time.2xx:
  min: ..... 2
  max: ..... 309
  mean: ..... 7.4
  median: ..... 6
  p95: ..... 10.9
  p99: ..... 43.4
http.responses: ..... 600
plugins.metrics-by-endpoint./tukano-1/rest/users/.codes.200: ..... 200
plugins.metrics-by-endpoint./tukano-1/rest/users/{{ id }}?pwd={{ pwd }}.code...  200
plugins.metrics-by-endpoint./tukano-1/rest/users/{{ id}}?pwd={{ pwd }}.codes...  200
plugins.metrics-by-endpoint.response_time./tukano-1/rest/users/:
  min: ..... 5
  max: ..... 46
  mean: ..... 9.8
  median: ..... 10.1
  p95: ..... 12.1
  p99: ..... 43.4
plugins.metrics-by-endpoint.response_time./tukano-1/rest/users/{{ id }}?pwd={{ pwd }}:
  min: ..... 5
  max: ..... 309
  mean: ..... 9.6
  median: ..... 6
  p95: ..... 10.1
  p99: ..... 94.6
plugins.metrics-by-endpoint.response_time./tukano-1/rest/users/{{ id}}?pwd={{ pwd }}:
  min: ..... 2
  max: ..... 15
  mean: ..... 2.9
  median: ..... 3
  p95: ..... 4
  p99: ..... 6
vusers.completed: ..... 200
vusers.created: ..... 200
vusers.created_by_name.TuKanoWholeUserFlow: ..... 200
vusers.failed: ..... 0
vusers.session_length:
  min: ..... 15.4
  max: ..... 365.2
  mean: ..... 26.5
  median: ..... 21.5
```

p95: 39.3
p99: 141.2

- Conclusion

Minikube significantly outperforms Azure in handling the user_register tests, by having no error responses (400, 404), having generally lower latency and reduced variability in response times.

upload_shorts

- Azure

http.codes.403: 30
http.codes.404: 30
http.downloaded_bytes: 0
http.request_rate: 6/sec
http.requests: 60
http.response_time:
  min: 3
  max: 429
  mean: 105.3
  median: 175.9
  p95: 202.4
  p99: 210.6
http.response_time.4xx:
  min: 3
  max: 429
  mean: 105.3
  median: 175.9
  p95: 202.4
  p99: 210.6
http.responses: 60
plugins.metrics-by-endpoint./tukano/rest/blobs/{{ blobUrl }}.codes.403: 30
plugins.metrics-by-endpoint./tukano/rest/shorts/{{ userId }}?pwd={{ pwd }}.c... 30
plugins.metrics-by-endpoint.response_time./tukano/rest/blobs/{{ blobUrl }}:
  min: 3
  max: 429
  mean: 18
  median: 4
  p95: 5
  p99: 5
plugins.metrics-by-endpoint.response_time./tukano/rest/shorts/{{ userId }}?pwd={{ pwd }}:
  min: 176

```
  max: ..... 211
  mean: ..... 192.5
  median: ..... 190.6
  p95: ..... 202.4
  p99: ..... 206.5
vusers.completed: ..... 30
vusers.created: ..... 30
vusers.created_by_name.Upload short: ..... 30
vusers.failed: ..... 0
vusers.session_length:
  min: ..... 188.4
  max: ..... 671.7
  mean: ..... 220
  median: ..... 202.4
  p95: ..... 228.2
  p99: ..... 232.8
```

- Minikube

```
http.codes.200: ..... 29
http.codes.204: ..... 30
http.codes.404: ..... 1
http.downloaded_bytes: ..... 8753
http.request_rate: ..... 7/sec
http.requests: ..... 60
http.response_time:
  min: ..... 3
  max: ..... 82
  mean: ..... 10.5
  median: ..... 6
  p95: ..... 41.7
  p99: ..... 82.3
http.response_time.2xx:
  min: ..... 3
  max: ..... 82
  mean: ..... 10.6
  median: ..... 6
  p95: ..... 41.7
  p99: ..... 82.3
http.response_time.4xx:
  min: ..... 6
  max: ..... 6
  mean: ..... 6
  median: ..... 6
  p95: ..... 6
  p99: ..... 6
http.responses: ..... 60
```

```
plugins.metrics-by-endpoint./tukano-1/rest/blobs/{{ blobUrl }}.codes.204: ..... 30
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ id }}?pwd={{ pwd }}.cod...  29
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ id }}?pwd={{ pwd }}.cod...  1
plugins.metrics-by-endpoint.response_time./tukano-1/rest/blobs/{{ blobUrl }}:
  min: ..... 3
  max: ..... 82
  mean: ..... 8.6
  median: ..... 4
  p95: ..... 34.1
  p99: ..... 41.7
plugins.metrics-by-endpoint.response_time./tukano-1/rest/shorts/{{ id }}?pwd={{ pwd }}:
  min: ..... 6
  max: ..... 82
  mean: ..... 12.4
  median: ..... 7
  p95: ..... 13.1
  p99: ..... 49.9
vusers.completed: ..... 30
vusers.created: ..... 30
vusers.created_by_name.Upload short: ..... 30
vusers.failed: ..... 0
vusers.session_length:
  min: ..... 14.4
  max: ..... 187.2
  mean: ..... 29.4
  median: ..... 19.1
  p95: ..... 58.6
  p99: ..... 71.5
```

- Conclusion

  Like the first example, Minikube is a better alternative than Azure:

1. In the first assignment, there were a lot more 403 and 404 responses whereas in this one only has one 404 error.
2. Minikube has better performance in general, with significantly lower response times and session lengths

Note: Minikube's 204 error is caused by the files in Artillery not uploading correctly (that is why the test fails in the beginning), so when we do the download operation, it returns *No Content* but the functionality is working.

realistic_flow

- Azure

errors.No shorts exist yet.: 10
http.codes.404: 20
http.downloaded_bytes: 0
http.request_rate: 4/sec
http.requests: 20
http.response_time:
  min: 170
  max: 983
  mean: 284
  median: 190.6
  p95: 757.6
  p99: 757.6
http.response_time.4xx:
  min: 170
  max: 983
  mean: 284
  median: 190.6
  p95: 757.6
  p99: 757.6
http.responses: 20
plugins.metrics-by-endpoint./tukano/rest/shorts/{{ shortId }}/{{ userId }}/l... 1
plugins.metrics-by-endpoint./tukano/rest/shorts/{{ userId }}/feed?pwd={{ pwd... 10
plugins.metrics-by-endpoint./tukano/rest/shorts/{{ userId }}/followers?pwd={... 1
plugins.metrics-by-endpoint./tukano/rest/shorts/{{ userId }}/shorts.codes.404: . 6
plugins.metrics-by-endpoint./tukano/rest/shorts/{{ userId1 }}/{{ userId2 }}/... 2
plugins.metrics-by-endpoint.response_time./tukano/rest/shorts/{{ shortId }}/{{ userId }}/likes?pwd={{ pwd }}:
  min: 175
  max: 175
  mean: 175
  median: 175.9
  p95: 175.9
  p99: 175.9
plugins.metrics-by-endpoint.response_time./tukano/rest/shorts/{{ userId }}/feed?pwd={{ pwd }}:
  min: 170
  max: 241
  mean: 190.2
  median: 186.8
  p95: 198.4
  p99: 198.4
plugins.metrics-by-endpoint.response_time./tukano/rest/shorts/{{ userId }}/followers?pwd={{ pwd }}:
  min: 983
  max: 983
  mean: 983
  median: 982.6
  p95: 982.6

```
  p99: ..... 982.6
plugins.metrics-by-endpoint.response_time./tukano/rest/shorts/{{ userId }}/shorts:
  min: ..... 287
  max: ..... 762
  mean: ..... 376.2
  median: ..... 295.9
  p95: ..... 314.2
  p99: ..... 314.2
plugins.metrics-by-endpoint.response_time./tukano/rest/shorts/{{ userId1 }}/{{ userId2
}}/followers?pwd={{ pwd }}:
  min: ..... 175
  max: ..... 187
  mean: ..... 181
  median: ..... 175.9
  p95: ..... 175.9
  p99: ..... 175.9
vusers.completed: ..... 20
vusers.created: ..... 30
vusers.created_by_name.Download short: ..... 6
vusers.created_by_name.Follow user: ..... 2
vusers.created_by_name.Get Short Likes: ..... 4
vusers.created_by_name.Get User Follows: ..... 1
vusers.created_by_name.Get User's Shorts: ..... 6
vusers.created_by_name.Like short: ..... 1
vusers.created_by_name.View feed: ..... 10
vusers.failed: ..... 10
vusers.session_length:
  min: ..... 178.5
  max: ..... 1003.4
  mean: ..... 291.6
  median: ..... 194.4
  p95: ..... 772.9
  p99: ..... 772.9
```

- Minikube

```
errors.Cannot read properties of undefined (reading 'ownerId'): ..... 9
http.codes.200: ..... 12
http.codes.204: ..... 7
http.codes.404: ..... 2
http.downloaded_bytes: ..... 70
http.request_rate: ..... 5/sec
http.requests: ..... 21
http.response_time:
  min: ..... 4
  max: ..... 16
```

  mean: 7.2
  median: 6
  p95: 10.9
  p99: 10.9
http.response_time.2xx:
  min: 5
  max: 16
  mean: 7.5
  median: 6
  p95: 10.9
  p99: 10.9
http.response_time.4xx:
  min: 4
  max: 5
  mean: 4.5
  median: 4
  p95: 4
  p99: 4
http.responses: 21
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ id }}/feed?pwd={{ pwd }... 11
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ id }}/shorts.codes.200: ... 1
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ id }}/{{ ownerId }}/lik... 2
plugins.metrics-by-endpoint./tukano-1/rest/shorts/{{ userId1 }}/{{ userId2 }... 7
plugins.metrics-by-endpoint.response_time./tukano-1/rest/shorts/{{ id }}/feed?pwd={{ pwd }}:
  min: 5
  max: 16
  mean: 6.4
  median: 5
  p95: 6
  p99: 6
plugins.metrics-by-endpoint.response_time./tukano-1/rest/shorts/{{ id }}/shorts:
  min: 7
  max: 7
  mean: 7
  median: 7
  p95: 7
  p99: 7
plugins.metrics-by-endpoint.response_time./tukano-1/rest/shorts/{{ id }}/{{ ownerId
}}/likes?pwd={{ pwd }}:
  min: 4
  max: 5
  mean: 4.5
  median: 4
  p95: 4
  p99: 4
plugins.metrics-by-endpoint.response_time./tukano-1/rest/shorts/{{ userId1 }}/{{ userId2
}}/followers?pwd={{ pwd }}:
  min: 7

```
  max: ..... 11
  mean: ..... 9.3
  median: ..... 10.1
  p95: ..... 10.9
  p99: ..... 10.9
vusers.completed: ..... 21
vusers.created: ..... 30
vusers.created_by_name.Download short: ..... 6
vusers.created_by_name.Follow user: ..... 7
vusers.created_by_name.Get Short Likes: ..... 3
vusers.created_by_name.Get User's Shorts: ..... 1
vusers.created_by_name.Like short: ..... 2
vusers.created_by_name.View feed: ..... 11
vusers.failed: ..... 9
vusers.session_length:
  min: ..... 7.2
  max: ..... 26.7
  mean: ..... 14.1
  median: ..... 10.9
  p95: ..... 26.3
  p99: ..... 26.3
```

- Conclusion

Same as the other two, Minikube outperforms Azure significantly. This way has less errors, with faster performance and shorter sessions, resulting in a better efficiency. We suspect that the majority of errors come from “follow” or “like” because we had some problems in the naming with @Jsonproperty that is why we have an “ownerId” error, but on postman it works.

Final Conclusion

Same as mentioned in the beginning, the objective was to compare the results of using Kubernetes. Although we know Minikube is the local version so it would have better latency than deploying to Azure, the performance analysis showed significantly better results when using Minikube. We can only conclude that Minikube resulted in a **better performance**.