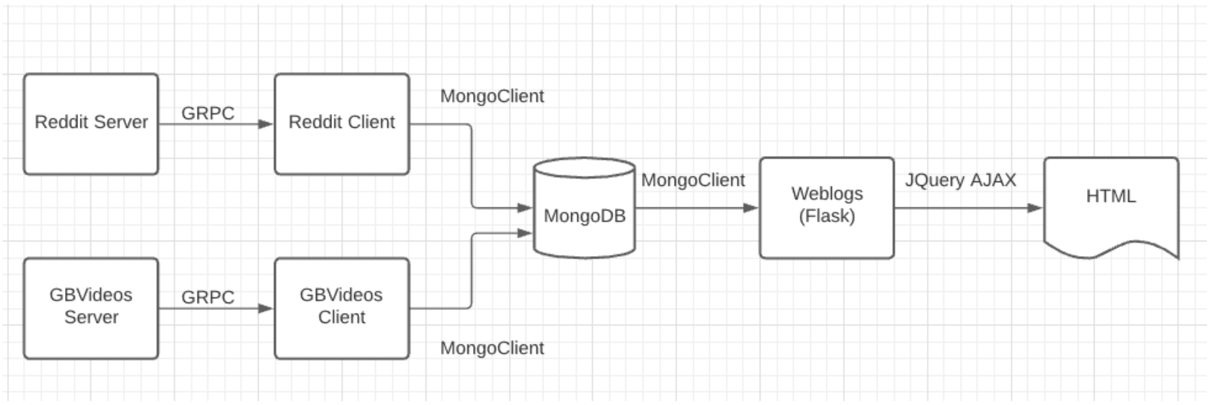
SOFT8026 - Data-driven Microservices
Assignment 2 Form

Instructions
Please complete the following form and include in the zip file you submit. Include screenshots / images in the appendices below the form.

Architecture Diagram

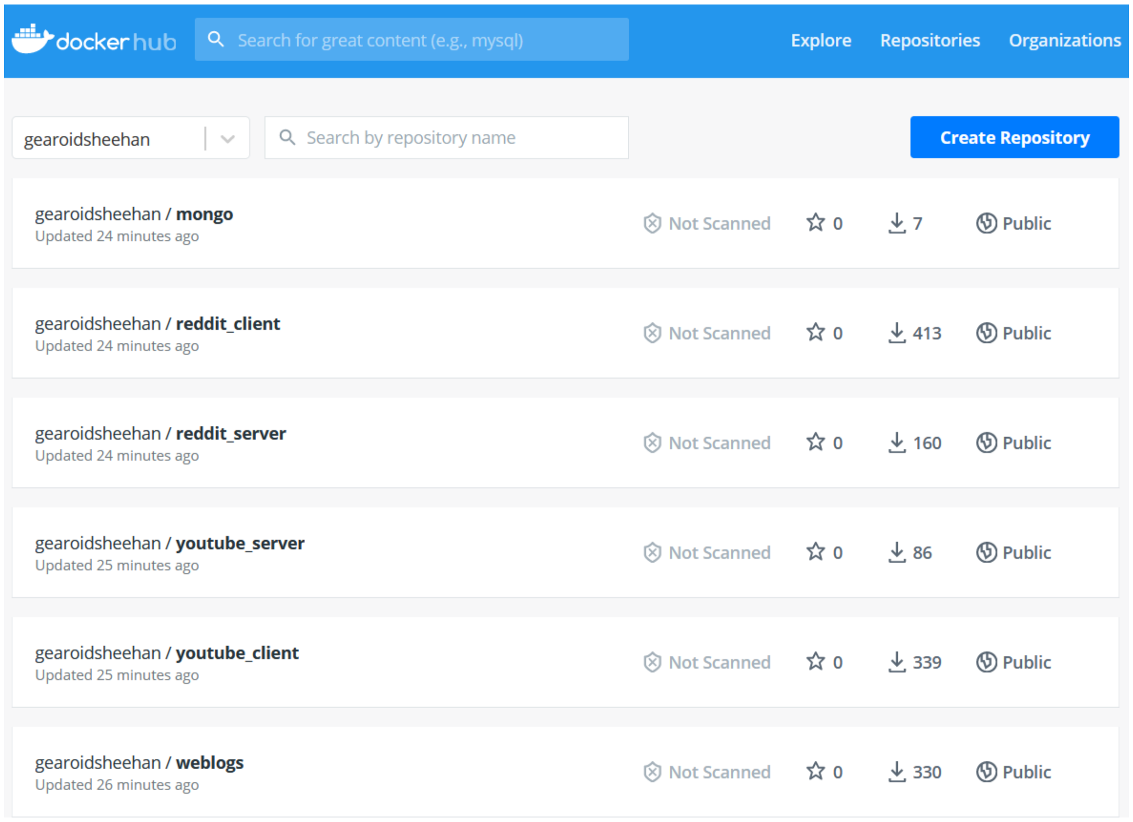


Move to Kubernetes	
Describe the progress you made moving your app to Kubernetes	**Running 'kubectl get pods'** When I created the pods for the first time Kubernetes ran as expected. I went back and changed a few bits in my client's code and pushed the changes to docker hub. Whenever I tried to get Kubernetes running again after that the images were failing to pull from docker hub for some reason (Except the MongoDB pod).
Describe what additional functionality you added, e.g. 2nd data source; include decisions made around architecture to include the extra functionality.	**Architecture (Repository Pattern)** For the architecture of the application, I used the repository pattern which is commonly used in REST applications (Repository Layer/Service Layer/ REST Controller). The data is read into the servers using a CSV reader, then streamed to the clients using GRPC. Analytics are then performed in the clients and are saved to a MongoDB instance. **Database chosen:** The reason I chose MongoDB over Redis is because MongoDB supports multi-threading out of the box. I began by using Redis but was having concurrency issues due to its single threaded nature, as this did not allow connections from both clients to write to the database at the same time. The data from the reddit client is saved with keys which are only even number, and the YouTube data similarly uses odd keys. This allowed both to be saved to the database from different clients without duplicating keys.

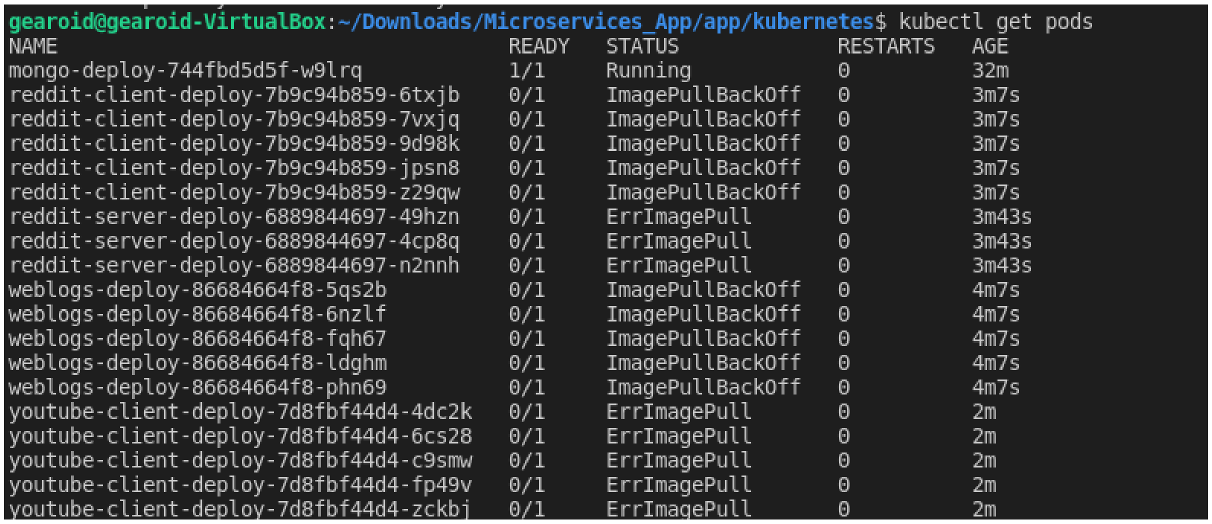
	Data being pulled from MongoDB container: The flask application then reads from the MongoDB instance, using a loop to check for an existing key then incrementing. Here the importance of using numerical keys came to play, as the data could now be retrieved in order of when it was first added to the database, by the key.
Describe your scaling and update strategy (as implemented in your Kubernetes deployments) – e.g. include why you chose to scale in the ratio among microservices that you did	For scaling the Kubernetes pods, I used 3 replicas for the servers and 5 replicas for the clients. The reason I used more replicas for the clients is because the servers are constantly streaming the same amount of data from the CSV, whereas the client are both receiving this data, responding to the servers and completing analytics, as well as uploading to the MongoDB database.
Testing	
Briefly describe the test you created and what type of test it is?	For the testing I used the Gatling load testing tool. I ran my application for a few seconds and saved a .har file from the developer console, under the network tab. This contained data about the web browsers interaction with my site. I then converted the .har file into a Scala file using the Gatling recorder software. Once this was done, I load tested the Scala file using the Gatling tool and below is the screenshot of the results.
Why did you choose that test?	I chose Gatling as it appears to be one of the most popular and reputable tests for load balancing available. It also provided results in chart form.
How did or how would you automate the test?	I would automate the test so that when a certain amount of traffic is using the application, the test is ran and the results are emailed to the developers so they can view whether the large amount of traffic is effecting the applications performance.
Monitoring	
Briefly describe the monitor you created and what type of monitor it is?	**YouTube Video showing the stream working dynamically:** https://www.youtube.com/watch?v=icmSy5ALmYQ On the template side, three bar graphs exist for each of the type of analytics made back in the client. The data is streamed in dynamically using jQuery Ajax requests made every 1 second. The key which states if the data is for reddit posts or gbvideos is then checked and is sorted into the correct bar in each chart then.
Why did you choose that monitor?	I have previous knowledge of jQuery and I feel that coloured graphs give off a clearer depiction of performance than just text.
Serverless Function	
What serverless function and functionality did you implement?	For the serverless function, I made a simple function which returned the number of records which have been added to the MongoDB database at a

	given time. Unfortunately, the function did not work correctly. After investigation it appeared the Kubeless Kubernetes pod was not deploying, similarly to earlier when attempting to move my main application to Kubernetes.
Where in your application did you or would you slot in this functionality?	In the flask application, so it could be called like a rest endpoint by a frontend application.
Any other comments? (e.g. you may have had to opt for Plan B, using Compose to implement extra functionality)	In the end I had to use docker-compose, which was disappointing as I originally managed to get Kubernetes working.

Appendix A – Screenshot(s) of your application running (e.g. Kubernetes log output, any web pages, etc.)

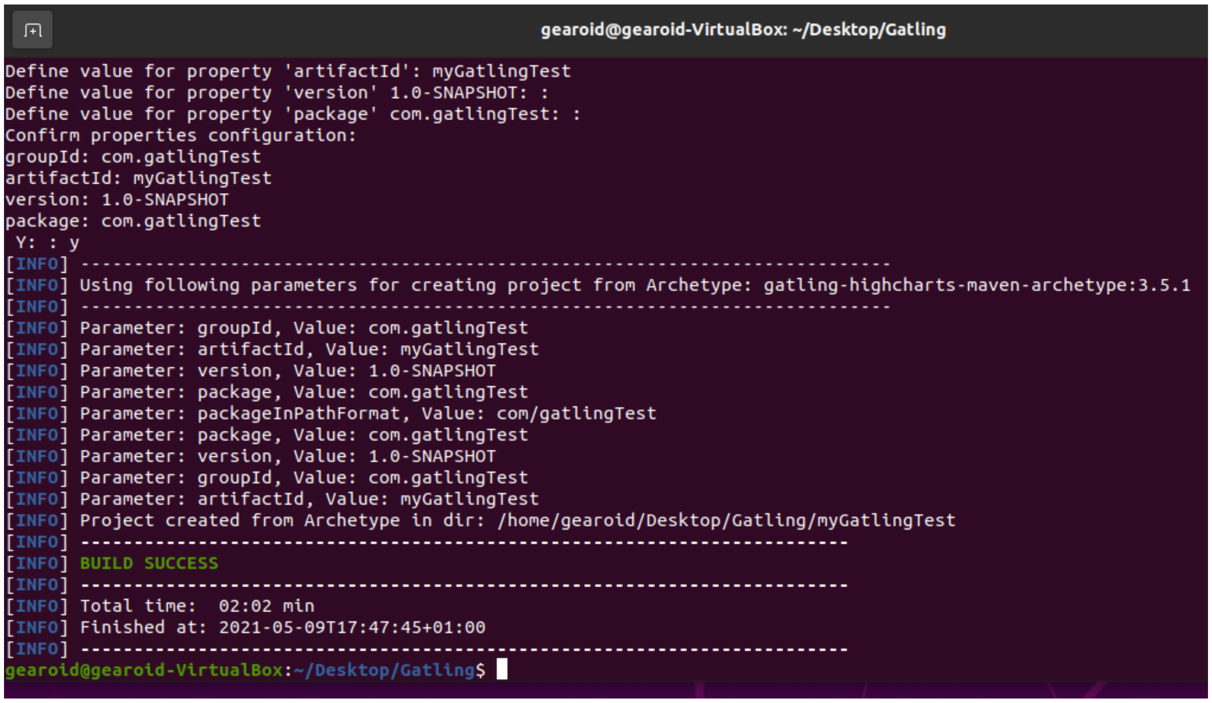


gearoidsheehan / **mongo** Updated 24 minutes ago	Not Scanned	0	7	Public
gearoidsheehan / **reddit_client** Updated 24 minutes ago	Not Scanned	0	413	Public
gearoidsheehan / **reddit_server** Updated 24 minutes ago	Not Scanned	0	160	Public
gearoidsheehan / **youtube_server** Updated 25 minutes ago	Not Scanned	0	86	Public
gearoidsheehan / **youtube_client** Updated 25 minutes ago	Not Scanned	0	339	Public
gearoidsheehan / **weblogs** Updated 26 minutes ago	Not Scanned	0	330	Public

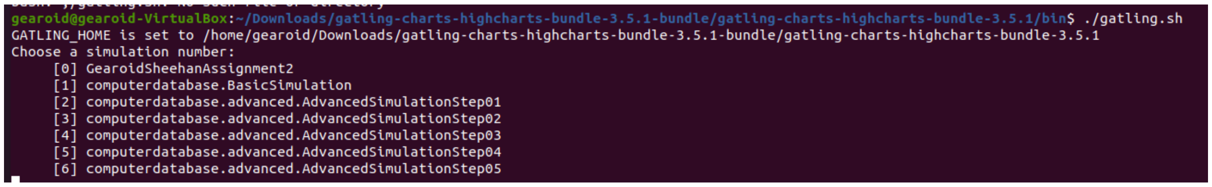


```
gearoid@gearoid-VirtualBox:~/Downloads/Microservices_App/app/kubernetes$ kubectl get pods
NAME                                    READY   STATUS             RESTARTS   AGE
mongo-deploy-744fbd5d5f-w9lrq           1/1     Running            0          32m
reddit-client-deploy-7b9c94b859-6txjb   0/1     ImagePullBackOff   0          3m7s
reddit-client-deploy-7b9c94b859-7vxjq   0/1     ImagePullBackOff   0          3m7s
reddit-client-deploy-7b9c94b859-9d98k   0/1     ImagePullBackOff   0          3m7s
reddit-client-deploy-7b9c94b859-jpsn8   0/1     ImagePullBackOff   0          3m7s
reddit-client-deploy-7b9c94b859-z29qw   0/1     ImagePullBackOff   0          3m7s
reddit-server-deploy-6889844697-49hzn   0/1     ErrImagePull       0          3m43s
reddit-server-deploy-6889844697-4cp8q   0/1     ErrImagePull       0          3m43s
reddit-server-deploy-6889844697-n2nnh   0/1     ErrImagePull       0          3m43s
weblogs-deploy-86684664f8-5qs2b         0/1     ImagePullBackOff   0          4m7s
weblogs-deploy-86684664f8-6nzlf         0/1     ImagePullBackOff   0          4m7s
weblogs-deploy-86684664f8-fqh67         0/1     ImagePullBackOff   0          4m7s
weblogs-deploy-86684664f8-ldghm         0/1     ImagePullBackOff   0          4m7s
weblogs-deploy-86684664f8-phn69         0/1     ImagePullBackOff   0          4m7s
youtube-client-deploy-7d8fbf44d4-4dc2k  0/1     ErrImagePull       0          2m
youtube-client-deploy-7d8fbf44d4-6cs28  0/1     ErrImagePull       0          2m
youtube-client-deploy-7d8fbf44d4-c9smw  0/1     ErrImagePull       0          2m
youtube-client-deploy-7d8fbf44d4-fp49v  0/1     ErrImagePull       0          2m
youtube-client-deploy-7d8fbf44d4-zckbj  0/1     ErrImagePull       0          2m
```

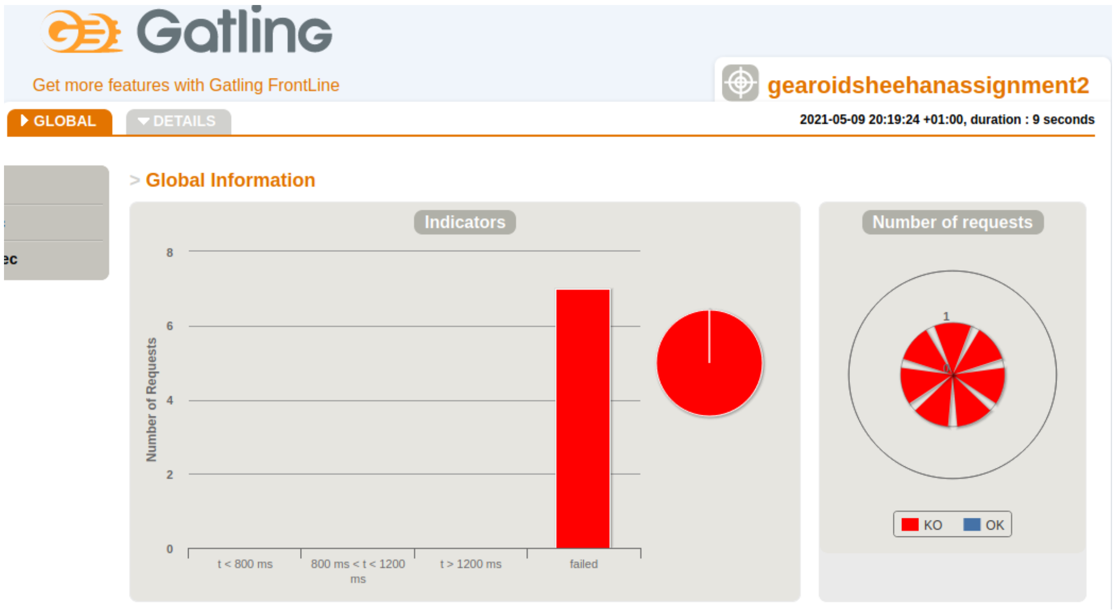
Appendix B – Screenshot(s) of your application being tested

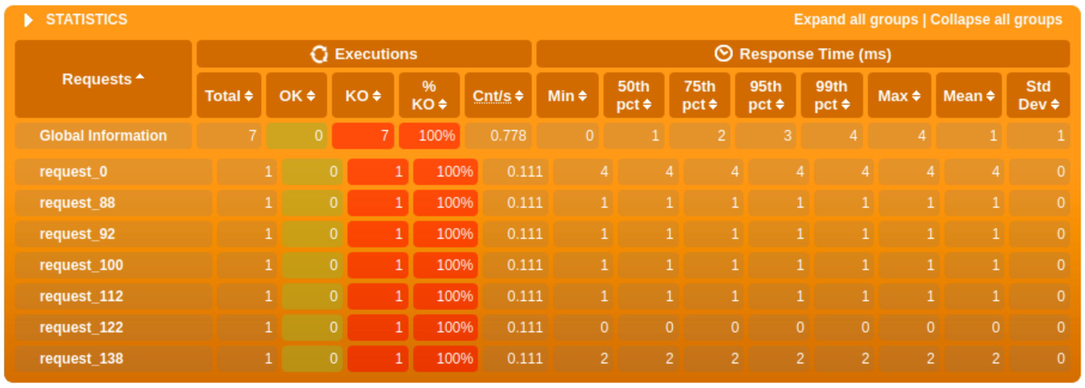


```
Define value for property 'artifactId': myGatlingTest
Define value for property 'version' 1.0-SNAPSHOT: :
Define value for property 'package' com.gatlingTest: :
Confirm properties configuration:
groupId: com.gatlingTest
artifactId: myGatlingTest
version: 1.0-SNAPSHOT
package: com.gatlingTest
 Y: : y
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Archetype: gatling-highcharts-maven-archetype:3.5.1
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: groupId, Value: com.gatlingTest
[INFO] Parameter: artifactId, Value: myGatlingTest
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.gatlingTest
[INFO] Parameter: packageInPathFormat, Value: com/gatlingTest
[INFO] Parameter: package, Value: com.gatlingTest
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: groupId, Value: com.gatlingTest
[INFO] Parameter: artifactId, Value: myGatlingTest
[INFO] Project created from Archetype in dir: /home/gearoid/Desktop/Gatling/myGatlingTest
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  02:02 min
[INFO] Finished at: 2021-05-09T17:47:45+01:00
[INFO] ------------------------------------------------------------------------
gearoid@gearoid-VirtualBox:~/Desktop/Gatling$ 
```



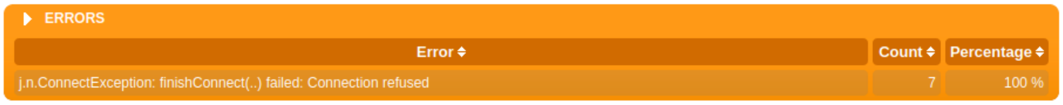
```
gearoid@gearoid-VirtualBox:~/Downloads/gatling-charts-highcharts-bundle-3.5.1-bundle/gatling-charts-highcharts-bundle-3.5.1/bin$ ./gatling.sh
GATLING_HOME is set to /home/gearoid/Downloads/gatling-charts-highcharts-bundle-3.5.1-bundle/gatling-charts-highcharts-bundle-3.5.1
Choose a simulation number:
     [0] GearoidSheehanAssignment2
     [1] computerdatabase.BasicSimulation
     [2] computerdatabase.advanced.AdvancedSimulationStep01
     [3] computerdatabase.advanced.AdvancedSimulationStep02
     [4] computerdatabase.advanced.AdvancedSimulationStep03
     [5] computerdatabase.advanced.AdvancedSimulationStep04
     [6] computerdatabase.advanced.AdvancedSimulationStep05
```





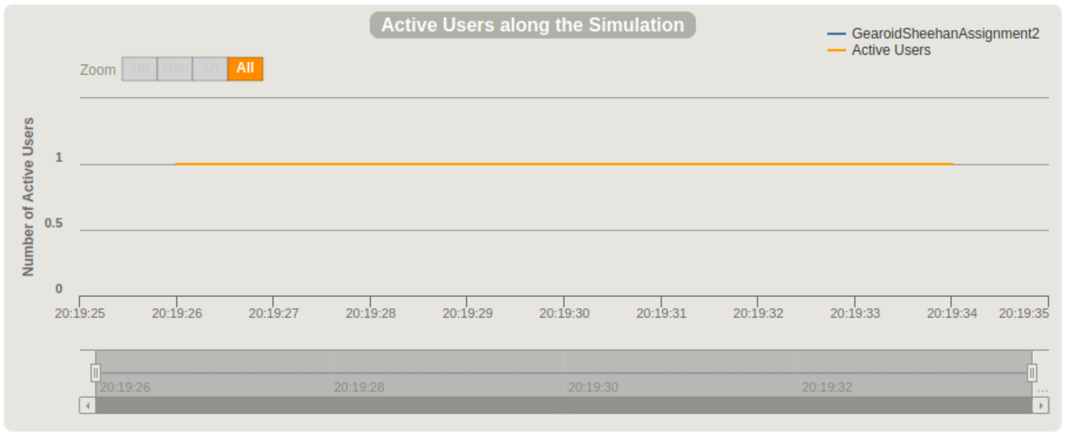
STATISTICS — Expand all groups | Collapse all groups

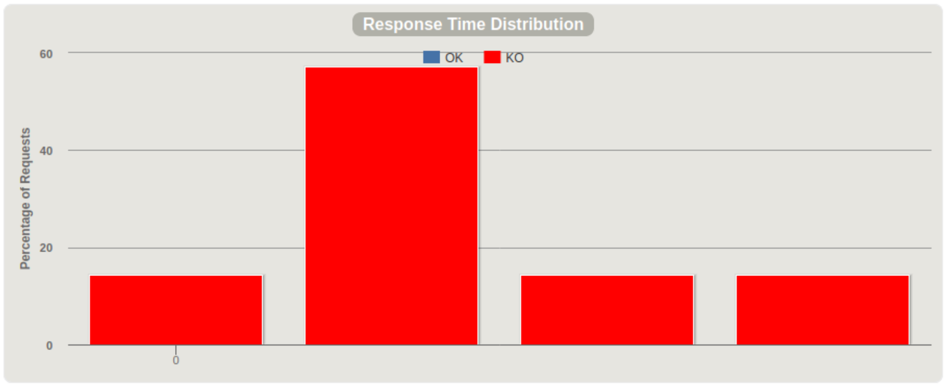
Requests	Executions					Response Time (ms)							
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
Global Information	7	0	7	100%	0.778	0	1	2	3	4	4	1	1
request_0	1	0	1	100%	0.111	4	4	4	4	4	4	4	0
request_88	1	0	1	100%	0.111	1	1	1	1	1	1	1	0
request_92	1	0	1	100%	0.111	1	1	1	1	1	1	1	0
request_100	1	0	1	100%	0.111	1	1	1	1	1	1	1	0
request_112	1	0	1	100%	0.111	1	1	1	1	1	1	1	0
request_122	1	0	1	100%	0.111	0	0	0	0	0	0	0	0
request_138	1	0	1	100%	0.111	2	2	2	2	2	2	2	0

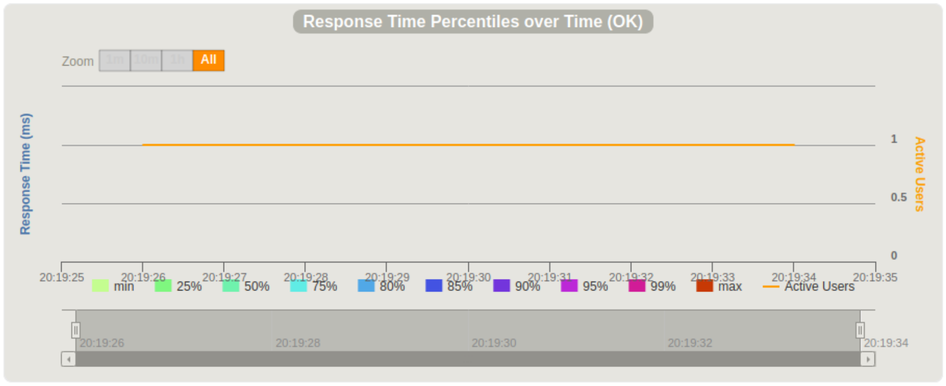


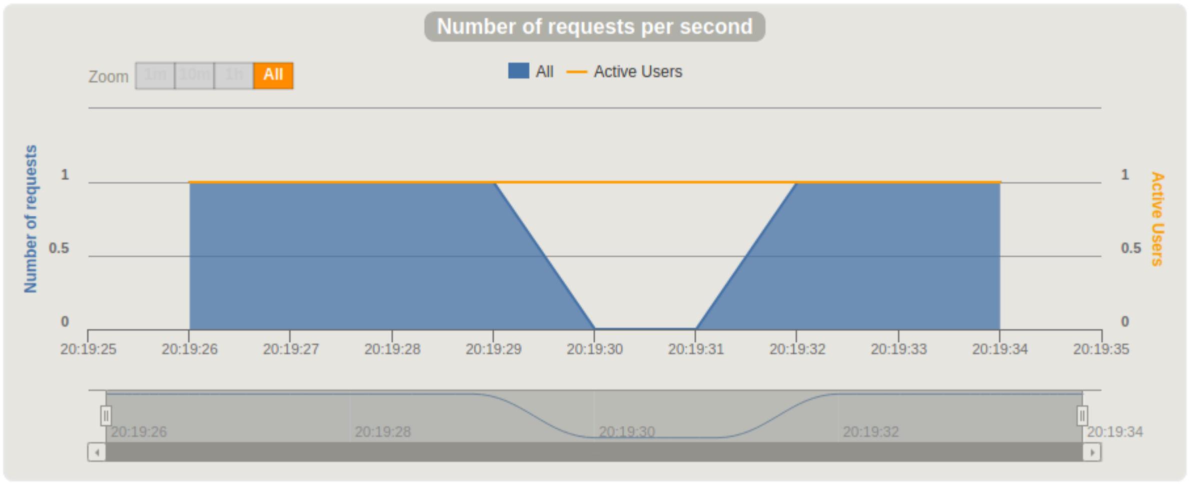
ERRORS

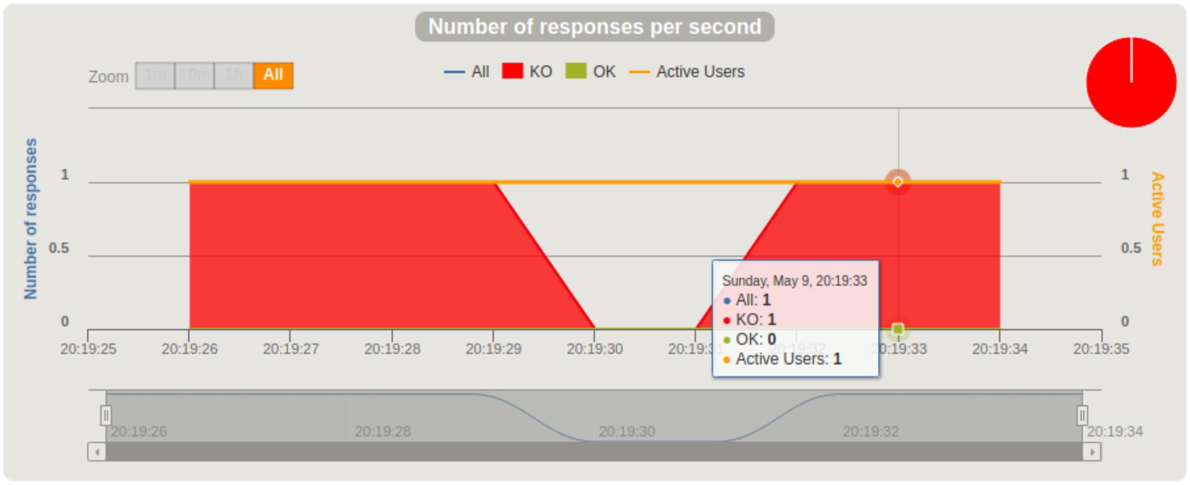
Error	Count	Percentage
j.n.ConnectException: finishConnect(..) failed: Connection refused	7	100 %



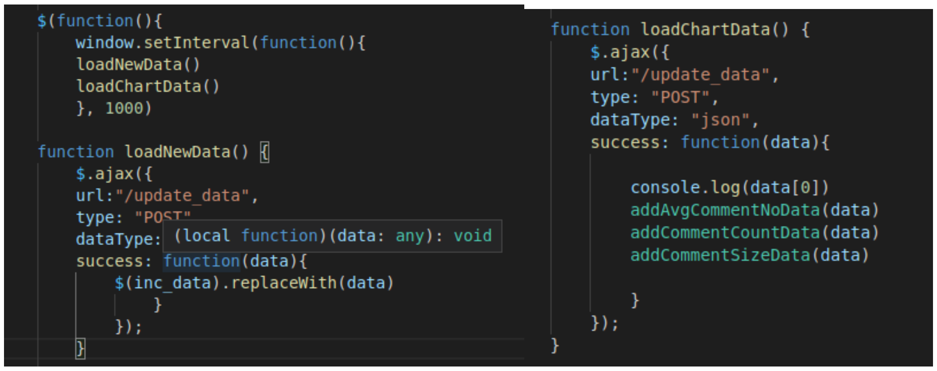

Response Time Distribution
OK
KO
Percentage of Requests
60
40
20
0
0


Response Time Percentiles over Time (OK)
Zoom 1m 10m 1h All
Response Time (ms)
Active Users
1
0.5
0
20:19:25
20:19:26
20:19:27
20:19:28
20:19:29
20:19:30
20:19:31
20:19:32
20:19:33
20:19:34
20:19:35
min 25% 50% 75% 80% 85% 90% 95% 99% max Active Users


Number of requests per second
Zoom 1m 10m 1h All
All
Active Users
Number of requests
Active Users
1
0.5
0
20:19:25
20:19:26
20:19:27
20:19:28
20:19:29
20:19:30
20:19:31
20:19:32
20:19:33
20:19:34
20:19:35


Number of responses per second
Zoom 1m 10m 1h All
All
KO
OK
Active Users
Number of responses
Active Users
1
0.5
0
Sunday, May 9, 20:19:33
All: 1
KO: 1
OK: 0
Active Users: 1
20:19:25
20:19:26
20:19:27
20:19:28
20:19:29
20:19:30
20:19:33
20:19:34
20:19:35

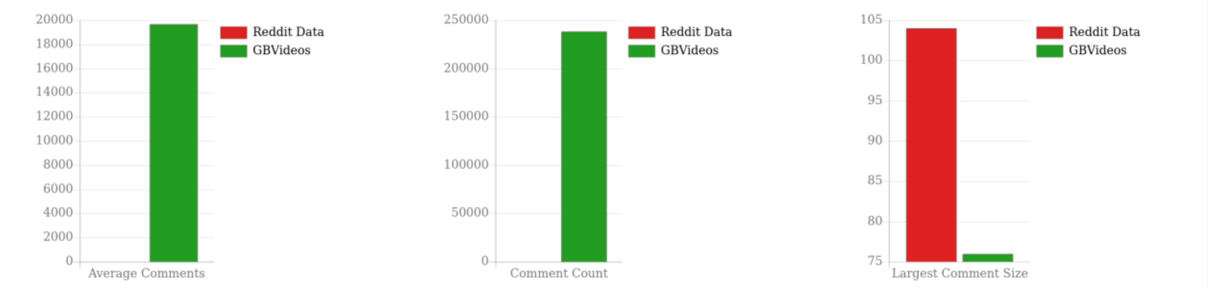
Appendix C – Screenshot(s) of your application being monitored

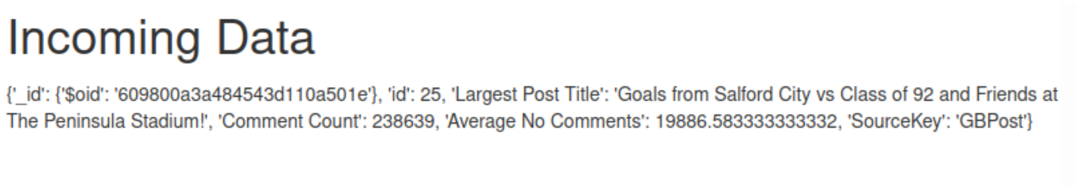


```
$(function(){
    window.setInterval(function(){
    loadNewData()
    loadChartData()
    }, 1000)

function loadNewData() {
    $.ajax({
    url:"/update_data",
    type: "POST"
    dataType: (local function)(data: any): void
    success: function(data){
        $(inc_data).replaceWith(data)
            }
        });
    }
```

```
function loadChartData() {
    $.ajax({
    url:"/update_data",
    type: "POST",
    dataType: "json",
    success: function(data){

        console.log(data[0])
        addAvgCommentNoData(data)
        addCommentCountData(data)
        addCommentSizeData(data)

        }
    });
}
```

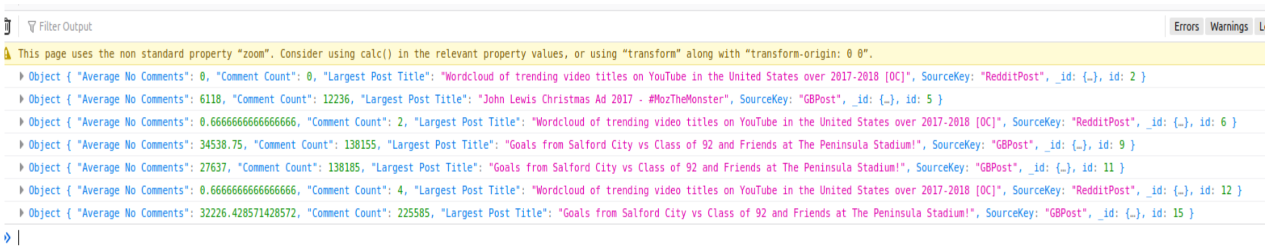
Reddit Data vs GBVideos Data





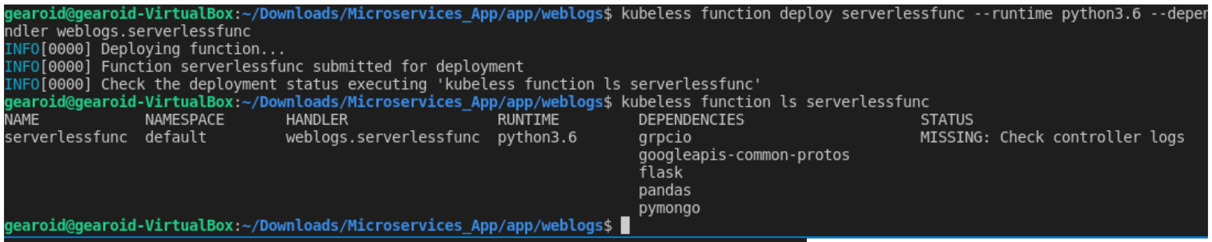
Incoming Data

{'_id': {'$oid': '609800a3a484543d110a501e'}, 'id': 25, 'Largest Post Title': 'Goals from Salford City vs Class of 92 and Friends at The Peninsula Stadium!', 'Comment Count': 238639, 'Average No Comments': 19886.583333333332, 'SourceKey': 'GBPost'}

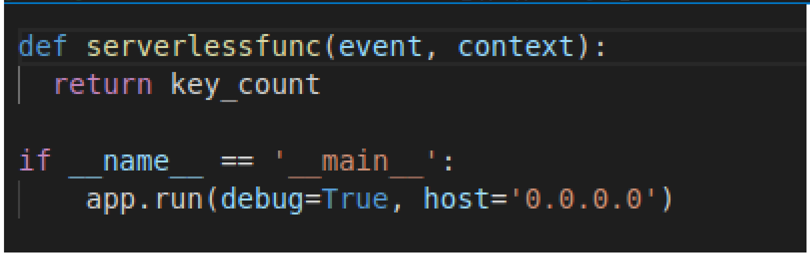


```
This page uses the non standard property "zoom". Consider using calc() in the relevant property values, or using "transform" along with "transform-origin: 0 0".
Object { "Average No Comments": 0, "Comment Count": 0, "Largest Post Title": "Wordcloud of trending video titles on YouTube in the United States over 2017-2018 [OC]", SourceKey: "RedditPost", _id: {…}, id: 2 }
Object { "Average No Comments": 6118, "Comment Count": 12236, "Largest Post Title": "John Lewis Christmas Ad 2017 - #MozTheMonster", SourceKey: "GBPost", _id: {…}, id: 5 }
Object { "Average No Comments": 0.6666666666666666, "Comment Count": 2, "Largest Post Title": "Wordcloud of trending video titles on YouTube in the United States over 2017-2018 [OC]", SourceKey: "RedditPost", _id: {…}, id: 6 }
Object { "Average No Comments": 34538.75, "Comment Count": 138155, "Largest Post Title": "Goals from Salford City vs Class of 92 and Friends at The Peninsula Stadium!", SourceKey: "GBPost", _id: {…}, id: 9 }
Object { "Average No Comments": 27637, "Comment Count": 138185, "Largest Post Title": "Goals from Salford City vs Class of 92 and Friends at The Peninsula Stadium!", SourceKey: "GBPost", _id: {…}, id: 11 }
Object { "Average No Comments": 0.6666666666666666, "Comment Count": 4, "Largest Post Title": "Wordcloud of trending video titles on YouTube in the United States over 2017-2018 [OC]", SourceKey: "RedditPost", _id: {…}, id: 12 }
Object { "Average No Comments": 32226.428571428572, "Comment Count": 225585, "Largest Post Title": "Goals from Salford City vs Class of 92 and Friends at The Peninsula Stadium!", SourceKey: "GBPost", _id: {…}, id: 15 }
»
```

Appendix D – Screenshot(s) of your serverless function running



```
gearoid@gearoid-VirtualBox:~/Downloads/Microservices_App/app/weblogs$ kubeless function deploy serverlessfunc --runtime python3.6 --deper
ndler weblogs.serverlessfunc
INFO[0000] Deploying function...
INFO[0000] Function serverlessfunc submitted for deployment
INFO[0000] Check the deployment status executing 'kubeless function ls serverlessfunc'
gearoid@gearoid-VirtualBox:~/Downloads/Microservices_App/app/weblogs$ kubeless function ls serverlessfunc
NAME            NAMESPACE       HANDLER                 RUNTIME         DEPENDENCIES                    STATUS
serverlessfunc  default         weblogs.serverlessfunc  python3.6       grpcio                          MISSING: Check controller logs
                                                                        googleapis-common-protos
                                                                        flask
                                                                        pandas
                                                                        pymongo
gearoid@gearoid-VirtualBox:~/Downloads/Microservices_App/app/weblogs$ 
```



```
def serverlessfunc(event, context):
  return key_count

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')
```