

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.

Move to Kubernetes	
Describe the progress you made moving your app to Kubernetes	<p>The process of moving over to Kubernetes was slightly difficult at first regarding tagging the containers I wanted to be stored by docker.</p> <p>Once set up though I created files with all my deployments and an handy readme to refer back to for useful console commands, screenshots of it running in Kubectl below</p>
Describe what additional functionality you added, e.g. 2 nd data source; include decisions made around architecture to include the extra functionality.	<p>I added a Digimon dataset as most of the data was like the Pokémon data. This meant originally that I wouldn't have to change the analytics client however this changed as I couldn't get the server to accept command line input to switch its datasets on start.</p> <p>Instead I had two server containers and two analytic containers that sent their data to Redis. Minimal changes had to be made to the clients and servers, just changing the ports they used and renaming some of the fields.</p> <p>I added a new section to the website using the new data. With Digimon data now available along with Pokémon data.</p>
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	<p>I settled on 2 servers, 2 for Pokémon data and 2 for Digimon data. Then I had 1 analytics client for Pokémon and 1 for Digimon data. This then got stored on Redis which I found out can handle a lot more website clients than my PC could ever replicate.</p> <p>This also allows for the data servers to be changed during a rolling update to introduce new data without stopping the new data.</p> <p>Too many web containers slowed the entire system down due to a configuration error on my side not balancing containers between CPU cores properly etc.</p> <p>While testing I had too many pods for my system to handle and 22gb of Ram was used by the docker Vmmem along with my 24 CPU threads being maxed out.</p>

Testing	
Briefly describe the test you created and what type of test it is?	I used K6 for testing as it allowed me to simply load test the website with users, I hit a limit within Linux and couldn't scale past 4000 concurrent website users due to a limit on open files that was difficult to change.
Why did you choose that test?	It allowed me to simply load test the website, it was simple to setup and operate. The scripts is below along with the testing results
How did or how would you automate the test?	<p>The tests runs itself once started and it can be configured for different run times. It could be setup to run for the day with simulated peaks and dips in users etc.</p> <p>It's started via command-line so a test can be easily automated. Another script can be setup just to immediately start with a number of numbers users and times etc so it can be easily started via command line.</p>

Monitoring	
Briefly describe the monitor you created and what type of monitor it is?	In order to monitor my Kubernetes, I used Splunk. I did try Kubernetes dashboard but setting it up was difficult and broke my connection to docker Kubernetes when I set up the proxy.
Why did you choose that monitor?	It installed very fast and immediately gave me a web dashboard where I could see all the details about my pods.
Serverless Function	
What serverless function and functionality did you implement?	Kubeless is seemingly deprecated and the QuickStart guides are tutorials have been taken down.
Where in your application did you or would you slot in this functionality?	I intended to write a simple python program that would return if the server was working as intended or to automatically run a load test
Any other comments? (e.g. you may have had to opt for Plan B, using Compose to implement extra functionality)	<p>I tried to setup 3 different serverless systems, openfaas, fission and faas-netes, none of which worked.</p> <p>Open Faas was more promising but required a login in that I couldn't find</p>

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

Pokemon Metrics:

Average HP of Pokemon	68.75961538461539
Weakest attack of all Pokemon so far	5
Strongest attack of all Pokemon so far	181
Toughest Pokemon (HP + Attack)	Guzzlord with a score of 324

[Open Pokedex Entry for Guzzlord](#)

Digimon Metrics:


Average HP of Digimon	1210.8835341365461
Weakest attack of all Digimon so far	52
Strongest attack of all Digimon so far	318
Toughest Digimon (HP + Attack)	Gankoomon with a score of 2268

```
eoghan@DESKTOP-SSG04AC:/mnt/c/Users/sceer/Desktop/College Work Sem 2
NAME                READY    UP-TO-DATE    AVAILABLE    AGE
digimon-client-deploy 1/1      1              1            170m
digimon-server-deploy 2/2      2              2            170m
pokemon-client-deploy 1/1      1              1            170m
pokemon-server-deploy 2/2      2              2            170m
pokemon-website-deploy 8/8      8              8            170m
redis-deploy          1/1      1              1            170m
```

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

```
import http from 'k6/http';
import { check, sleep } from 'k6';
export let options = {
  stages: [
    { duration: '15s', target: 1000 },
    { duration: '30s', target: 3500 },
    { duration: '20s', target: 0 },
  ],
};
export default function() {
  let res = http.get('http://127.0.0.1:30000');
  check(res, { 'status was 200': r => r.status == 200 });
  sleep(1);
}
```

```
eoghan@DESKTOP-SSG94AC:/mnt/c/Users/sceer/Desktop/College Work Sem 2/Data-driven-Microservices-Assignment$ k6 run Deployment/Testing/script.js
```



```

execution: local
  script: Deployment/Testing/script.js
  output: -

scenarios: (100.00%) 1 scenario, 3500 max VUs, 1m35s max duration (incl. graceful stop):
  * default: Up to 3500 looping VUs for 1m5s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

running (1m05.9s), 0000/3500 VUs, 62039 complete and 0 interrupted iterations
default ✓ [=====] 0000/3500 VUs 1m5s

  ✓ status was 200

checks.....: 100.00% ✓ 62039      × 0
data_received.....: 3.2 GB  49 MB/s
data_sent.....: 5.0 MB  76 kB/s
http_req_blocked.....: avg=26.34µs min=1.38µs med=3.78µs max=68.51ms p(90)=6.55µs p(95)=108.18µs
http_req_connecting.....: avg=18.7µs min=0s med=0s max=68.44ms p(90)=0s p(95)=63.67µs
http_req_duration.....: avg=825.86ms min=3.97ms med=615.87ms max=8.71s p(90)=1.96s p(95)=2.32s
  { expected_response:true }...: avg=825.86ms min=3.97ms med=615.87ms max=8.71s p(90)=1.96s p(95)=2.32s
http_req_failed.....: 0.00% ✓ 0      × 62039
http_req_receiving.....: avg=1.78ms min=11.79µs med=435.52µs max=145.5ms p(90)=4.62ms p(95)=7.78ms
http_req_sending.....: avg=16.58µs min=4.11µs med=12.03µs max=12.09ms p(90)=25.56µs p(95)=42.45µs
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=824.06ms min=3.91ms med=612.92ms max=8.71s p(90)=1.95s p(95)=2.31s
http_reqs.....: 62039  940.867374/s
iteration_duration.....: avg=1.82s min=1s med=1.61s max=9.71s p(90)=2.96s p(95)=3.32s
iterations.....: 62039  940.867374/s
vus.....: 20 min=20 max=3500
vus_max.....: 3500 min=3500 max=3500

```

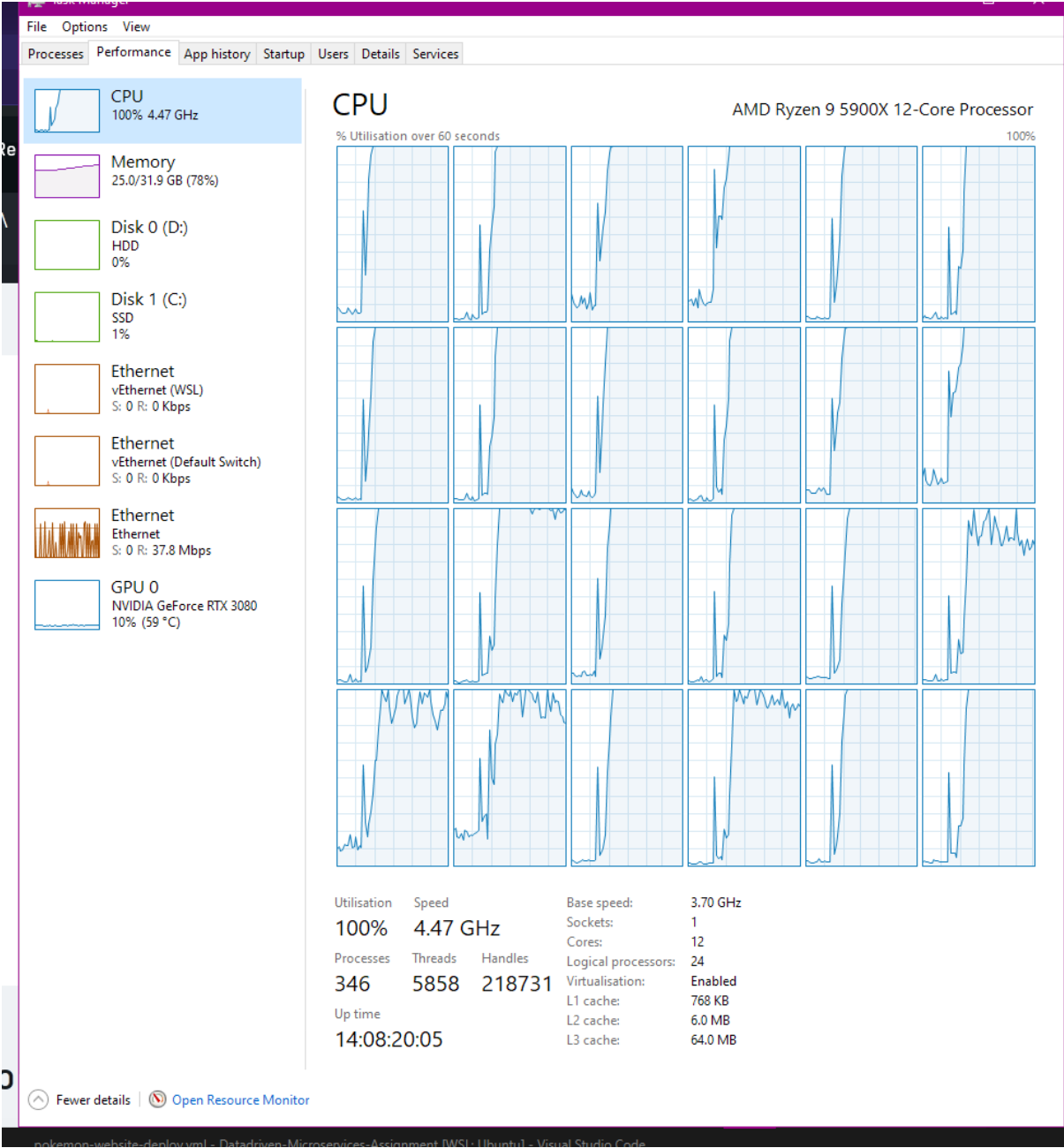
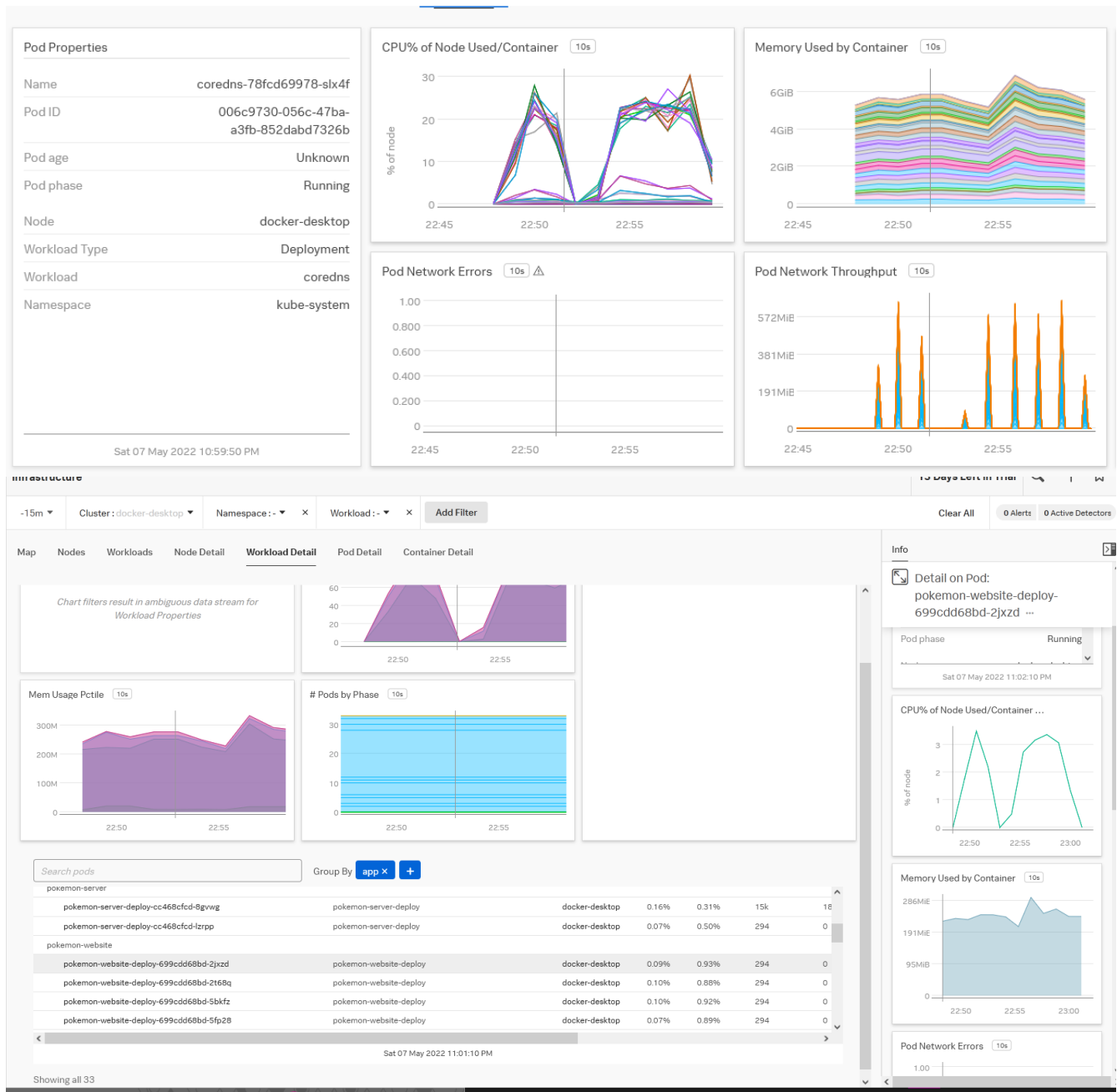


Figure 1 With 4 servers, 4 clients, 2 Redis and 16 webservers being used by 3500 simulated users. Computer was not too happy

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



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

```
eoghan@DESKTOP-SS694AC:/mnt/c/Users/sceer/Desktop/Collge Work Sem 2/Datadriven-Microservices-Assignment$ curl -XPOST https://127.0.0.1:30000/testserver
curl: (35) error:140BF108:SSL routines:ssl3_get_record:wrong version number
eoghan@DESKTOP-SS694AC:/mnt/c/Users/sceer/Desktop/Collge Work Sem 2/Datadriven-Microservices-Assignment$ curl -XPOST https://127.0.0.1:30000/testserverHandling connection for 8080
eoghan@DESKTOP-SS694AC:/mnt/c/Users/sceer/Desktop/Collge Work Sem 2/Datadriven-Microservices-Assignment$ echo -n $PASSWORD | faas-cli secret create fn-basic-auth-password
Creating secret: fn-basic-auth-password
Handling connection for 8080
Created: 202 Accepted
eoghan@DESKTOP-SS694AC:/mnt/c/Users/sceer/Desktop/Collge Work Sem 2/Datadriven-Microservices-Assignment$ faas-cli up
2022/05/08 15:42:35 No templates found in current directory.
2022/05/08 15:42:36 Attempting to expand templates from https://github.com/openfaas/templates.git
2022/05/08 15:42:37 Fetched 16 template(s) : [csharp dockerfile go javall javall-vert-x node node12 node12-debian node14 node16 node17 php7 python python3 python3-debian ruby] from https://github.com/openfaas/templates.git
Please provide a valid --image name for your Docker image
eoghan@DESKTOP-SS694AC:/mnt/c/Users/sceer/Desktop/Collge Work Sem 2/Datadriven-Microservices-Assignment$ Handling connection for 8080
^C
eoghan@DESKTOP-SS694AC:/mnt/c/Users/sceer/Desktop/Collge Work Sem 2/Datadriven-Microservices-Assignment$ curl $OPENFAAS_URL/function/basic-auth --basic --user=admin:password
curl: option --user=admin:password: is unknown
curl: try 'curl --help' or 'curl --manual' for more information
eoghan@DESKTOP-SS694AC:/mnt/c/Users/sceer/Desktop/Collge Work Sem 2/Datadriven-Microservices-Assignment$ curl $OPENFAAS_URL/function/basic-auth --basic --user=admin:02df94c5711780249dafd5dc1f3458532df21053
curl: option --user=admin:02df94c5711780249dafd5dc1f3458532df21053: is unknown
curl: try 'curl --help' or 'curl --manual' for more information
eoghan@DESKTOP-SS694AC:/mnt/c/Users/sceer/Desktop/Collge Work Sem 2/Datadriven-Microservices-Assignment$ fission environment create - name env-name - image fission/python-env
```

