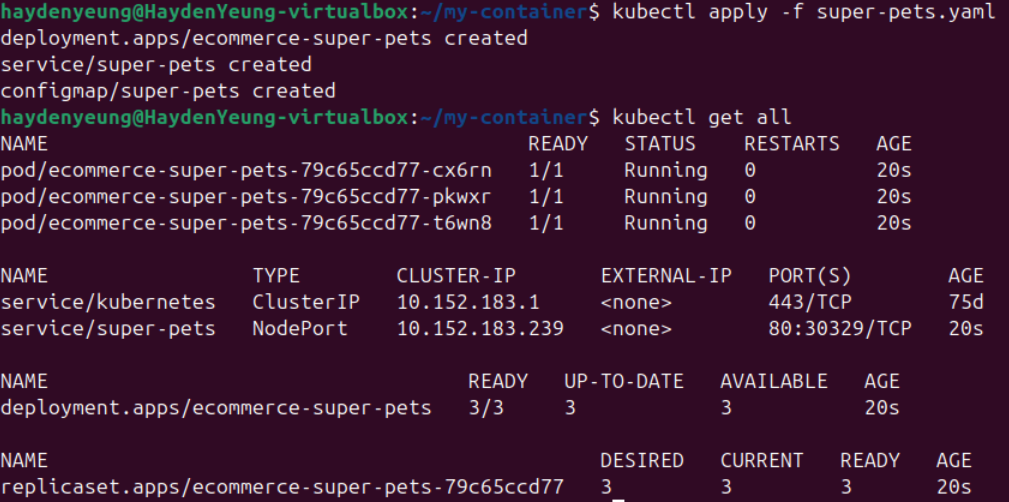
1. Unit Recap

* This week material and lab activities is a recap for what we have gone through this unit, it revisted from the basic elements like deploying simple NodeJs app, then configured this deployment to have a sidecar architecture, editing reasonable resources for both the container and sidecar container, scaling the deployment – which are the core elements / principle of DevOp, thus, they are all equally important as they are built on top of each other. This is reflected through errors I made through this lab: incorrect placing of codes in super-pets.yaml lead to deployment failure, miscalculation in setup for both memory and cpu requests and limits – also due to low spec computational VM that causing errors during “curling” loop and lead to further error during scaling as the readinessProbe mistakenly flagged the generated pods as unhealthy due to taking quite a period of time to fully running. All of these are expected in real life situations.

2. Lab Activities

Initial deployment

Followed the lab instructions, result below was obtained from “kubectl get all” command



Obtained three different pods from “curl 10.152.183.239” command

A screen shot of a computer

AI-generated content may be incorrect.

These errors was caused by forgot to include \n by the end of res.write(‘Hello ‘ + clientIP + ‘, this is ‘ + server\_name + ‘ v1 on ‘ + os.hostname());

Application data & sidecars

Edited the super-pets.yaml according to lab instructions and obtained the following results

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

API security & recource needs

Followed the instructions and applied “kubectl get all” to observe the results:

A computer screen shot of a program

AI-generated content may be incorrect.

A screen shot of a computer screen

AI-generated content may be incorrect.

Activity 1 – Try out the application

Because of experienced the above errors, I decided to remove resources from both the NodeJs container & Envoy Container and the results were at below

A screen shot of a computer program

AI-generated content may be incorrect.

Thus, I assume that running both containers is quite a heavy toll for my nodes.

A screenshot of a computer screen

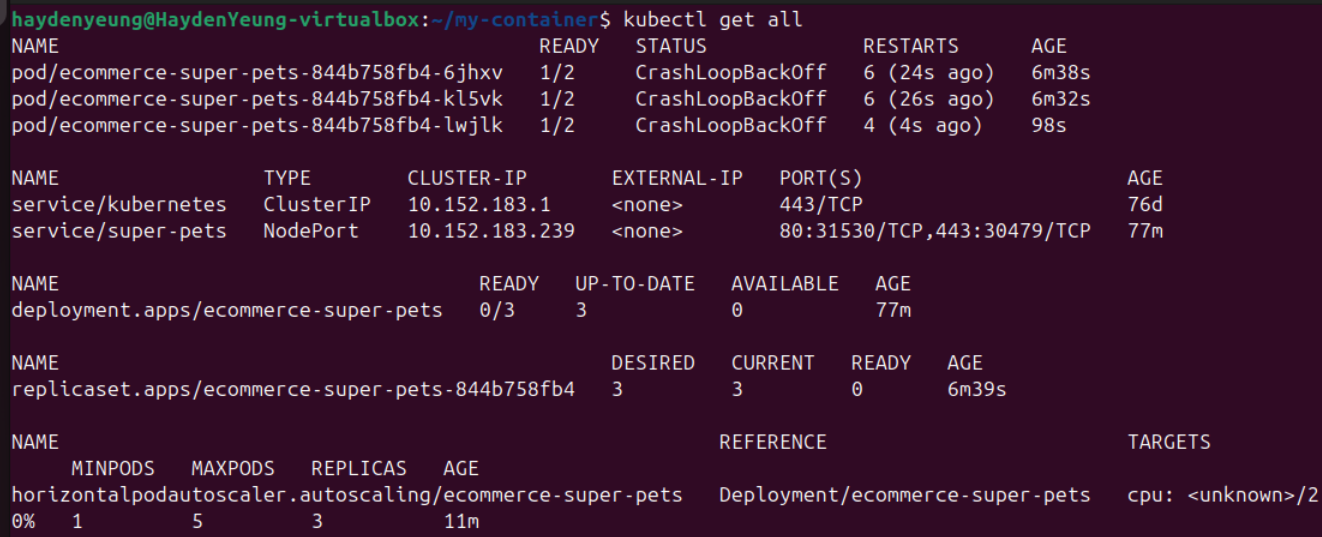
AI-generated content may be incorrect.

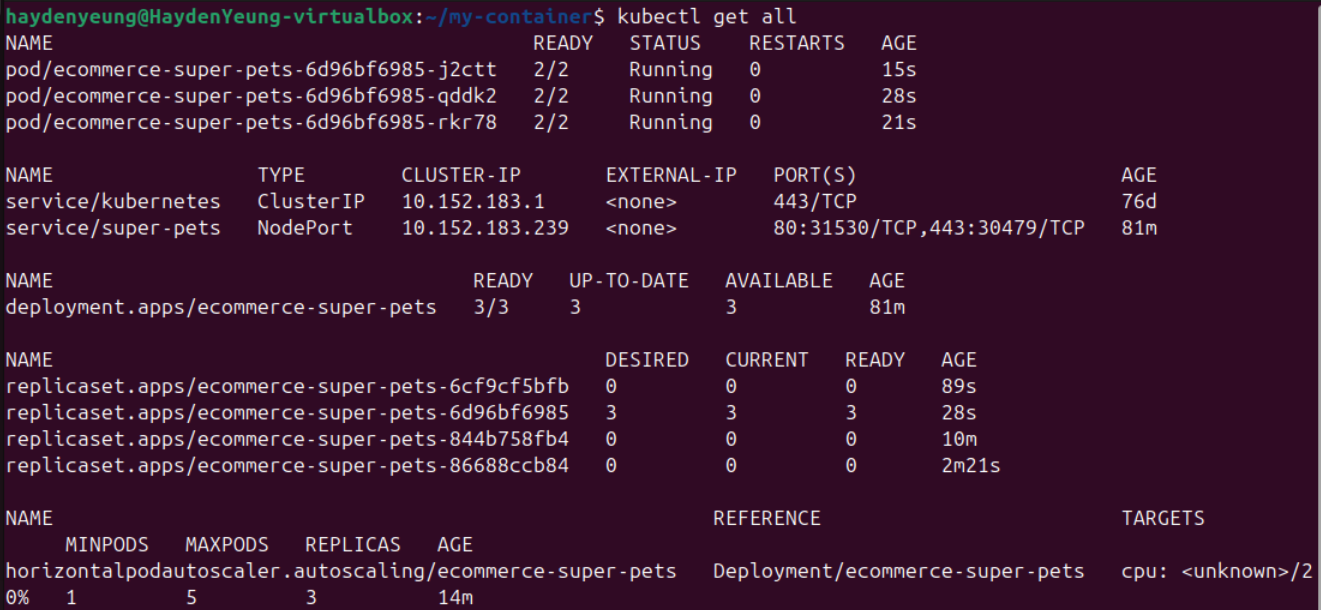
Hence, changing the image did not solve the problem.

Based on the previous tasks, theoretically speaking is that:

* When changing the image to v2 (or v3), the Deployment performs a rolling update, replacing v1 pods with v2 pods. During this process, the curl loop shows intermittent failures (e.g., "Connection refused" or timeouts) because some pods are terminating while others are starting. Once the update completes, the responses reflect the new version.
* With preference from Grok, I also found that: because there’s no readinessProbe defined for the ecommerce container at this stage.
  + Without a readiness probe, Kubernetes routes traffic to pods as soon as their containers start, even if the NodeJS application isn’t fully initialized (i.e., not yet listening on port 8080).
  + This leads to failed requests during the rolling update.
  + Additionally, the rolling update itself causes brief periods where fewer pods are available, increasing the chance of requests hitting a pod that’s not ready.

Task 2 – Is it fixed now?





A screenshot of a computer

AI-generated content may be incorrect.

I had to remove the readinessProbe in order for all 3 pods running – this can be explained that these pods are healthy (proved by curl-able). The readinessProbe, as introduced from previous lecture, is that it will check on the status of the pod upon its creation and the time was set too low (10s for an Ubuntu VM with low spec in RAM) will always result in failed (CrashLoopBackOff).

Theoretically speaking: in an ideal node with good computational spec having readinessProbe will prevent the error we encountered (expected to encountered in task 1 instead of error due to “spamming” curl command on low computing spec node).

Challenge Task

A screenshot of a computer

AI-generated content may be incorrect.

I had visited the mentioned page, and will repeat the visit frequently in the near future as I will try to practice deploying my personal projects with k8s.

3. Quiz Result

A screenshot of a computer

AI-generated content may be incorrect.