

Lab 7 - Kubernetes Practice

Hugo Garrido-Lestache

February 27, 2025

CSC 5201 301

Introduction

In this lab I will be using kubernetes to deploy a multi-pod application. This application will be a simple flask app which will connect to a sql database. The flask app will have 3 pods and there will be 1 pod for the sql database.

The sencond part of the lab I will be researching and describing an intresting use case for kubernetes.

1 Kubernetes Practice

After following the instructions in the lab I was able to deploy the application. The application is a simple flask app that connects to a sql database. once I deployed the app I was able to access the app from the browser. Below is a screenshot of the app running in the browser in figure 1.

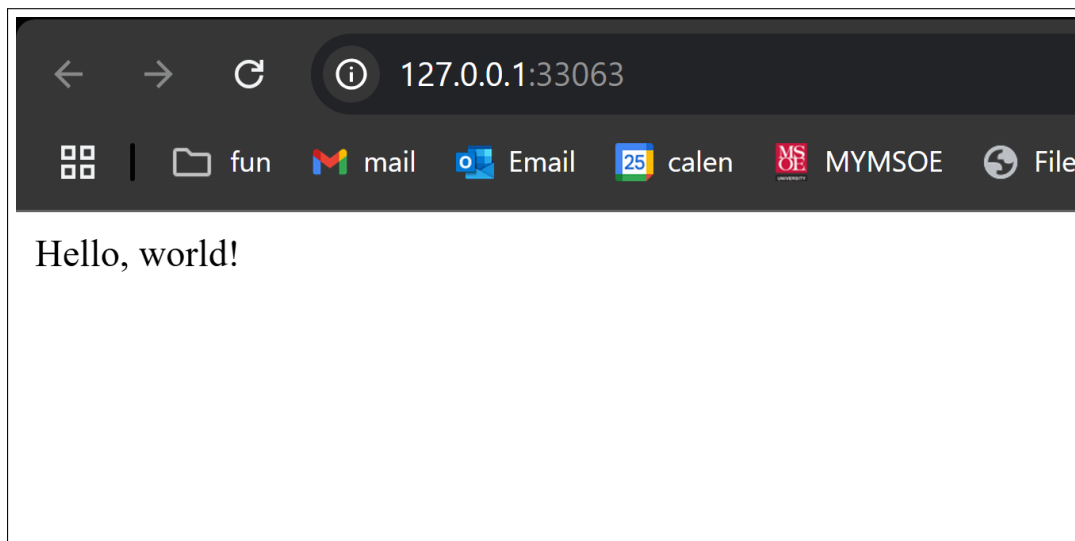


Figure 1: Flask app running in the browser

After ensuring the app was deployed I did curl request to the app to add a new user. Below is me adding a new user to the database in figure 2.

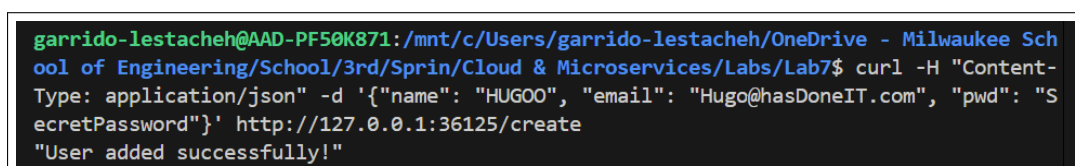


Figure 2: Curl request to add a new user to the database

I then checked the database to ensure the user was added. Below is a screenshot of the database with the new user in figure 3.

```
mysql> select * from users;
+-----+-----+-----+-----+
| user_id | user_name | user_email      | user_password |
+-----+-----+-----+-----+
|      1 | HUGOO    | Hugo@hasDoneIT.com | SecretPassword |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

Figure 3: Database with the new user added

All these screenshot show that the application was deployed correctly and the database was connected to the flask app. I also additionally checked the minikube dashboard to check on the status. It was cool to see that there is a dashboard that shows the status of the pods and the services which is easy to access and use. Below is a screenshot of the minikube dashboard in figure 4.

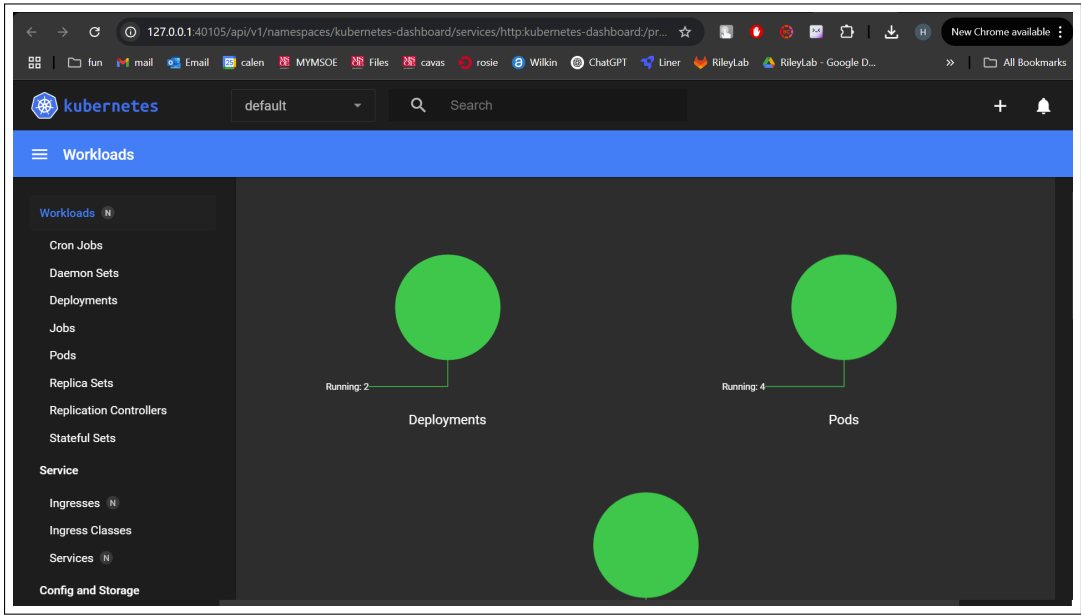


Figure 4: Minikube dashboard

2 Kubernetes Use Case

The use case I am going to describe is going to be for training complex reinforcement learning models. Reinforcement learning models differ from common machine learning models due to the fact that they don't just learn on data but also generate the data they learn from. This is usually done in a loop where they collect data and then learn from that data repeatedly. This process can be very computationally expensive and time consuming and could leverage from parallelization. Additionally Reinforcement learning has a lot of hyperparameters that need to be tuned and this can also be done in parallel. Additionally testing Reinforcement learning models may also be computationally expensive if the loss function is not directly tied to performance.

2.1 How Kubernetes can help

Kubernetes can help in many different ways and ideas.

1. Faster training: By using Kubernetes we can speed up both the training through the data collections process by scaling up the number of pods which simulate the environment and collect data.
2. Hyperparameter tuning: By using Kubernetes we can run multiple pods with different hyperparameters and then compare the results to see which hyperparameters work best. This can be done in quite interesting and complex methods such as terminating experiments early if they are not performing well and replacing them with new experiments. We can also allocate resources in a manner which is relative to the performance of the experiments allowing for more resources to be allocated to the better performing experiments. We can also be smart about how we select hyperparameters to try next by using bayesian optimization or other methods
3. Testing simultaneously: Some time measuring the quality of the model may also require some computational resources. Using Kubernetes we could create a new pod when a model needs to be tested and then delete the pod once the test is done. This could allow for testing to be done in parallel and not slow down the training process.

2.2 Challenge with Pod networking

Some potential challenges that may arise may be the communication between the pods. Pods would have to communicate with each other to share data and hyperparameters. This could be done through a shared volume or through a shared database but this could be a potential bottleneck.

2.3 Conclusion

Kubernetes could be a very useful tool for training complex reinforcement learning models. It could help speed up the final model training run time which may take days or weeks. Allowing for automatic managing of testing and experimenting with hyperparameters.

Additionally if this is done in the cloud it gives you the ability to scale as much as you may want. This could be useful as you could use a lot of resources quickly rather than having to wait for a long time to get the resources you need.

feedback

This Lab was quite interesting in understanding and actually getting experience with kubernetes and how it can be used. I think I still need more undirected practice with kubernetes to better understand it but I think getting some initial experience was very helpful. I had issues trying to deploy the app which boiled down to me not naming my python file correctly which was a bit frustrating but I was able to figure it out. I think the lab was well structured and the instructions were clear and easy to follow.

The use case idea was good in getting us to understand the benefits of kubernetes and how it can be used in a real world scenario. Thinking of your own usecase allows you to put your personal problems into perspective and see how kubernetes can help you solve them, making the lab more personal and interesting. Overall I think the lab was very good and I learned a lot from it.