Exploring a Kubernetes Database Deployment

Outcomes

1.1 Access a shell prompt and issue commands with correct syntax
3.1 Use a container orchestration system to run a multi-container environment
3.2 Automate a deployment using popular automation tools

Background

In this exercise you will have to opportunity to apply Kubernetes objects from a YAML file and use `kubectl` to explore a running system.

Inside the `exercises/db-k8s` directory you will find a `db-k8s.yml` file that has all of the objects we discussed during the presentation. Your goal for this exercise is to `apply` those objects to your local Kubernetes instance using kubectl and explore the running system using the `get`, `describe`, and `logs` commands. Feel free to try some database operations as well, you can get a bash prompt on any pod using `kubectl exec` just like you would with Docker.

Questions

Please answer the following questions in the text box for this assignment.

1. A systems architect was using a stock Docker Hub image with a custom ENTRYPOINT point script she had designed. This required a Dockerfile, BASH script, and a directory to store them. When she migrated to Kubernetes she was able to do this all in one YAML file. Describe how this is possible.
2. Why are *Services* essential to replication?
3. Why do we define two *Deployments* for our example?
4. How can our database deployment be improved?
5. Compare and contrast Kubernetes *PersistentVolumeClaims* with Docker Compose named volumes.
6. What does Kubernetes do when the `db-r` pod fails because `db-rw` is not up yet?