Week 3

Learnings

Exposing Ports

I want to be able to access my backend server from localhost.

Service

Creating a service creates an internal networking structure to the cluster.

Deployment vs Service

In looking at the Kubernetes documentation, they recommend creating a service to expose your pots. The service template looks very similar to the deployment template, and I don't understand why there are two configurations, and why we can't just put the networking configuration into the deployment configuration.

According to Kubernetes, a Service is evidently "a method for exposing a network application that is running as one or more Pods in your cluster." A service exists so that we can change the networking of our application independent of the application. For example, if you have a deployment, you can redeploy Pods while keeping the same networking in tact. If you keep redeploying pods, how do other connecting pods keep track of the IPs of the new pods being brought up. This is down through services.

Ingress

In order to map your cluster to localhost, you need to define an ingress configuration to be mapped to a service.

> Ingress exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. Traffic routing is controled by rules define on the Ingress resource.

Problems

Exposing my service

When trying to expose my service to localhost, nothing was working. I tried using load balancer services, node port services but nothing seemed to be working. Eventually I stumbled on a tutorial that showed minikube has it’s own ip which you can use to access your services.

```
minikube ip
```

You can get a list of all the services running, and how to access them (including your ports) by running

```
$ minikube service list
|---------------------|---------------------------|--------------|---------------------------|
|      NAMESPACE      |           NAME            | TARGET PORT  |            URL            |
|---------------------|---------------------------|--------------|---------------------------|
| default             | kubernetes                | No node port |                           |
| default             | webapplication-service    |         8080 | http://192.168.49.2:30000 |
| kube-system         | kube-dns                  | No node port |                           |
| kubernetes-dashboard | dashboard-metrics-scraper | No node port |                           |
| kubernetes-dashboard | kubernetes-dashboard      | No node port |                           |
|---------------------|---------------------------|--------------|---------------------------|
```

There is probably many ways of doing this, but this works for me for now.

Extra

Possible good article for next step: https://kubernetes.io/docs/tasks/access-application-cluster/connecting-frontend-backend/