Task 7.1

Jeremy Pedersen

Student ID: 217593144

This report details how I deployed a Node.js application to a Kubernetes cluster and made it accessible via a web service. The deployment aimed to demonstrate the use of Kubernetes for managing and scaling containerized applications.

Setup and Configuration

- **Kubernetes Cluster**: I utilized a Kubernetes cluster, either locally or on a cloud platform, to host the application. This provided the infrastructure necessary for deploying and managing containerized applications.
- **Deployment Configuration (deployment.yaml)**: This configuration file outlined the desired state of the application, including which Docker image to run and the number of replicas.
- **Service Configuration (service.yaml)**: This configuration detailed how the application was exposed within the cluster, specifying the type of service (e.g., NodePort or LoadBalancer) and the rules for accessing the application.
- **Cluster role binding and dashboard adminuser**.yaml config files were also used.

Deployment Process

- **Cluster Initialization**: The process began by ensuring the Kubernetes cluster was ready and capable of deploying containers.
- **Applying Configurations**: The deployment and service configurations were implemented to set up the application according to the specifications in the YAML files.
- **Verification**: I verified the successful deployment by ensuring the application components were properly created and operational.

Interaction

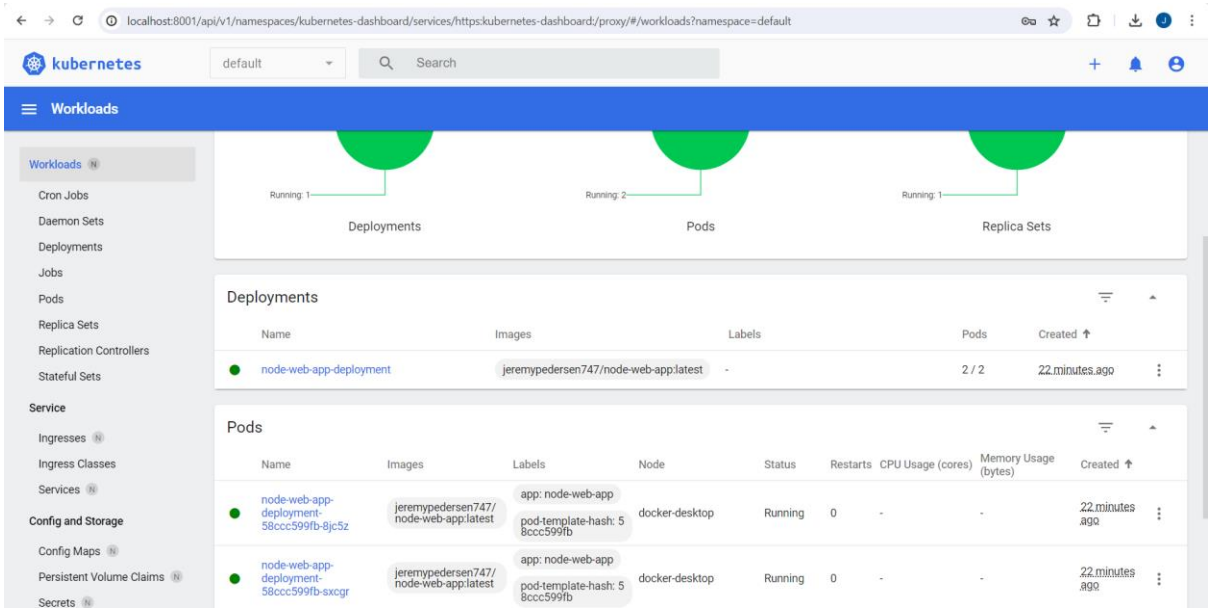
- **Accessing the Application**: The application was accessed through a designated URL or IP address, which was determined by the type of service defined in the service configuration.
- **Using the Application**: Interaction with the application was demonstrated by accessing specific endpoints through a web browser or other tools suited for web applications.

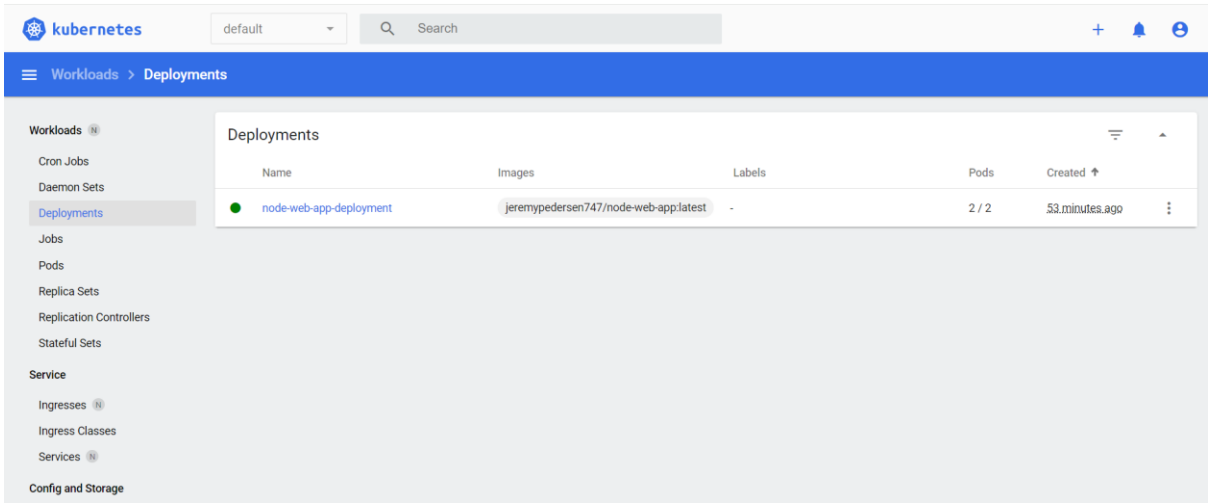
```
C:\temp>kubectl apply -f deployment.yaml
deployment.apps/node-web-app-deployment created

C:\temp>kubectl apply -f service.yaml
service/node-web-app-service created

C:\temp>kubectl get deployments
NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
node-web-app-deployment   2/2     2            2           116s

C:\temp>kubectl get services
NAME                 TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes           ClusterIP      10.96.0.1      <none>        443/TCP        125m
node-web-app-service   LoadBalancer   10.96.45.23    localhost     80:30918/TCP   24s
```






```
C:\temp>kubectl get nodes -o wide
NAME             STATUS   ROLES           AGE    VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE         KERNEL-VERSION                     CONTAINER-RUNTIME
docker-desktop   Ready    control-plane   143m   v1.29.1   192.168.65.3   <none>        Docker Desktop   5.15.146.1-microsoft-standard-WSL2   docker://25.0.3

C:\temp>
```