Levy Andrew Documentation Kube assignment

Task 1

Task 1: Dockerize your flask app and create an image you can push up to DockerHub. Use this link to help you create your flask app in a container: https://runnable.com/docker/python/dockerize-your-flask-application

(Use the url shortener or your flask app you deployed to elastic beanstalk).

1. Downloaded the URL-Shortener from github

   https://github.com/kura-labs-org/DEPLOY4_FLASK_APP.git

2. Unzip the file

3. Add a line of command to the application.py

   ```
   if __name__ == '__main__':
   app.run(host='0.0.0.0', port=5000, debug=True)
   ```

4. Create a Dockerfile with the content below:

   ```
   From python:3.10
   WORKDIR /urlshortener
   COPY . .
   run pip install -r requirements.txt
   ENV FLASK_APP=application.py
   EXPOSE 5000
   CMD flask run --host=0.0.0.0
   ```

5. Run CMD and cd to the url-shorterner folder location in my case it was in my document

   ```
   cd Document/urlshorterner
   ```

6. Run this command to build the image name urlshortener with the docker file well use above.

   ```
   docker build –t urlshortener .
   ```

7. To check if the image was build run this command

   ```
   docker images
   ```

8. To run this image and open port 8080 and redirect 5000 to it: run this command

   ```
   docker run -ti -p 8080:5000 urlshortener
   ```

9. Open a browser and type in the address

   ```
   localhost:8080
   ```

10. Retagging the image to send to docker-hub. Run the command below

    ```
    docker tag urlshortener la22/urlshortener
    ```

11. To check and to login in docker-hub. Run the command below

    ```
    docker login
    ```

12. To push the image to docker-hub. Run the command below

    ```
    docker push la22/urlshortener
    ```

Task 2

Task 2: Deploy your Flask app in Kubernetes.

- Create your cluster with a Load Balancer.
- Create a deployment yaml file for your flask app.
- Use the yaml file from yesterday's class to help you create a yaml file for your Flask app.

Through the WSL command line or the Ubuntu VM run those command

13. ***mkdir urlshortener***

14. ***cd urlshortener***

15. ***k3d cluster create urlshortener-cluster -p "8081:8080@loadbalancer"***

16. ***nano urlshortener.yml***

17. paste the following code in that file

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: urlshortener-deployment
spec:
  selector:
    matchLabels:
      app: urlshortener
  replicas: 1
  template:
    metadata:
      labels:
        app: urlshortener
    spec:
      containers:
      - name: urlshortener-container
        image: la22/urlshortener:latest
        ports:
        - containerPort: 5000

---

apiVersion: v1
kind: Service
metadata:
  name: urlshortener-service
spec:
  type: LoadBalancer
  ports:
  - port: 8080
    protocol: TCP
    targetPort: 5000
  selector:
    app: urlshortener
```

18. ***kubectl create -f urlshortener.yml***

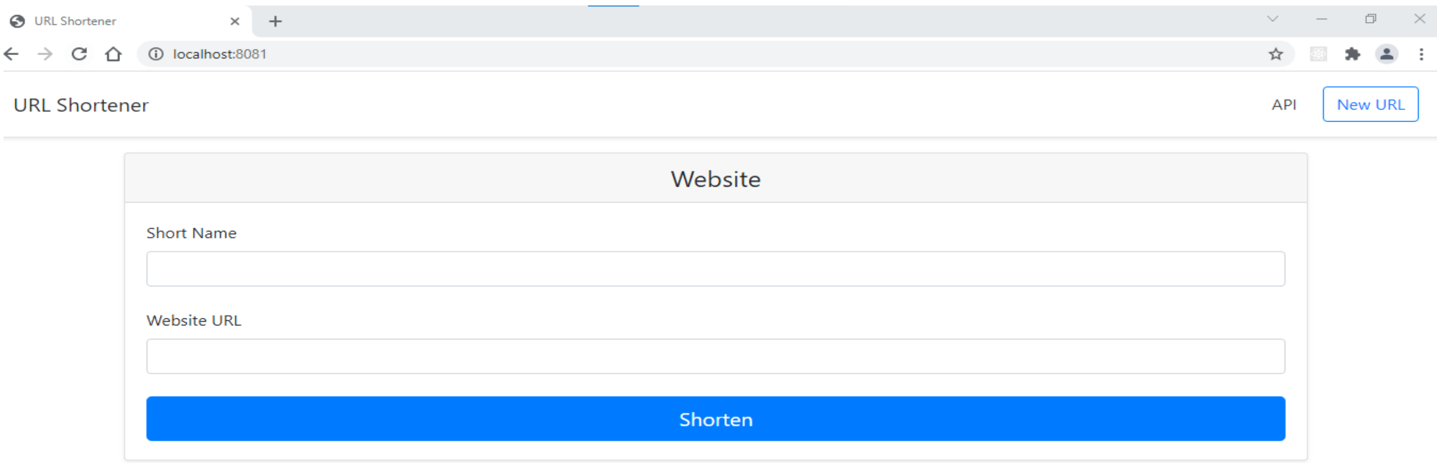
19. To check the status for the container. Run the follow command

kubectl get pod

kubectl get all

Levy Andrew Documentation Kube assignment

20. Screenshot of localhost:8081



We are now going to delete the services/deployment/cluster for this pod

21. Run this command to remove the service

    kubectl delete service urlshortener-service

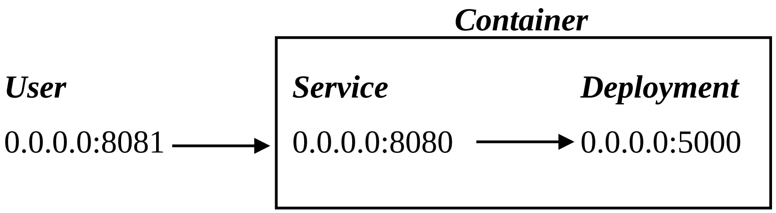
22. Run this command to remove the deployment

    kubectl delete deployment urlshortener-deployment

23. Run this command to delete the cluster

    k3d cluster delete urlshortener-cluster

Topology



Container

User

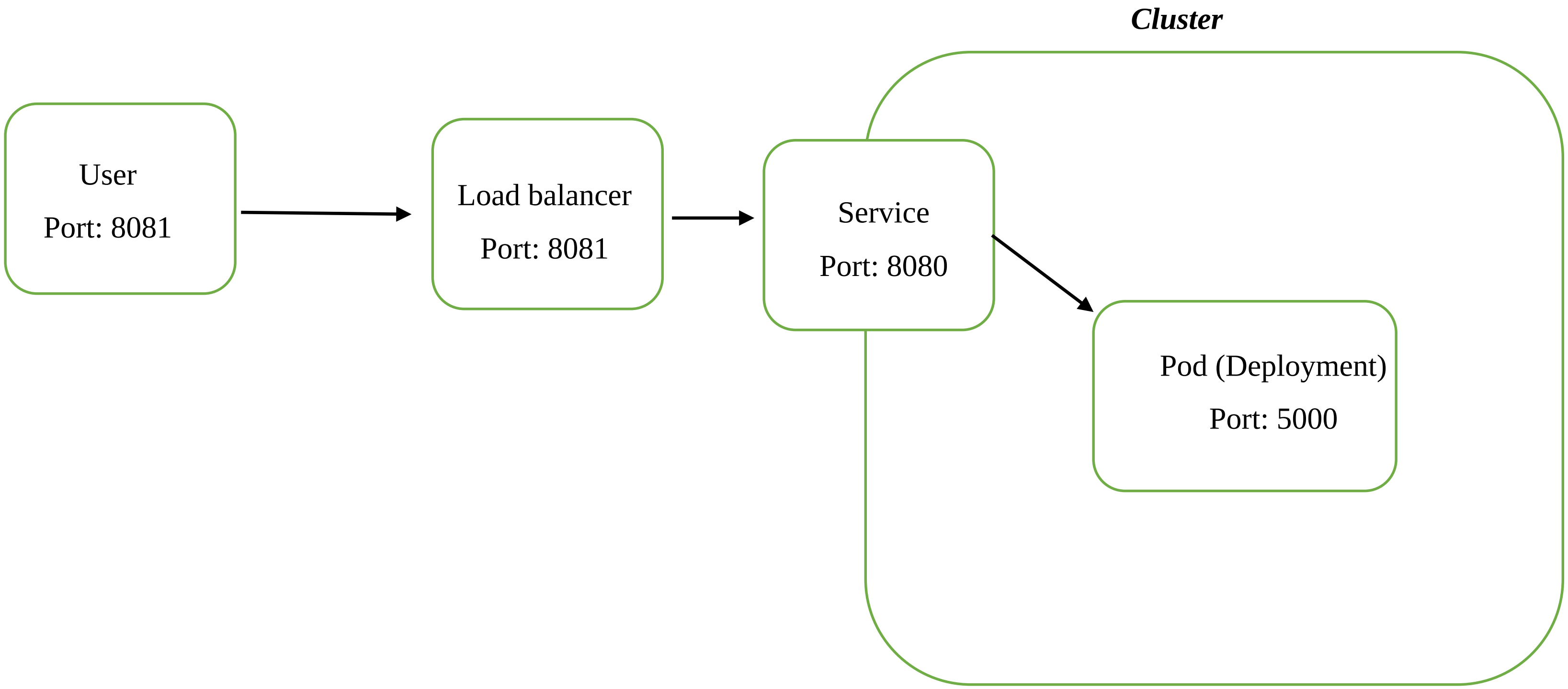
0.0.0.0:8081

Service

0.0.0.0:8080

Deployment

0.0.0.0:5000



Cluster

User

Port: 8081

Load balancer

Port: 8081

Service

Port: 8080

Pod (Deployment)

Port: 5000