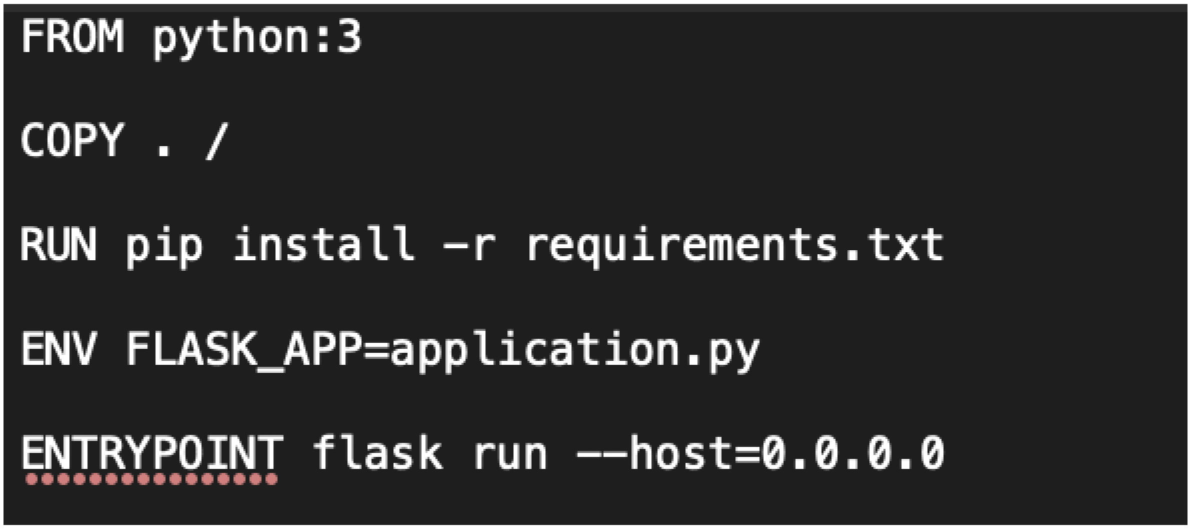
Objective: To dockerize a python flask application, push the image to Dockerhub and deploy the application to Kubernetes.

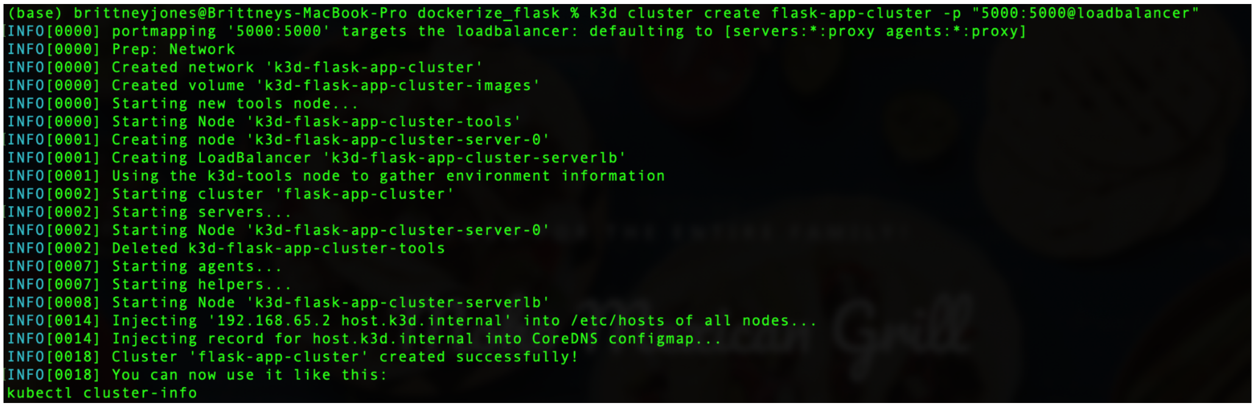
1. In order to create a Docker container with your flask application you first need to create a Docker image. To create a docker image you need a dockerfile:



```
FROM python:3

COPY . /

RUN pip install -r requirements.txt

ENV FLASK_APP=application.py

ENTRYPOINT flask run --host=0.0.0.0
```

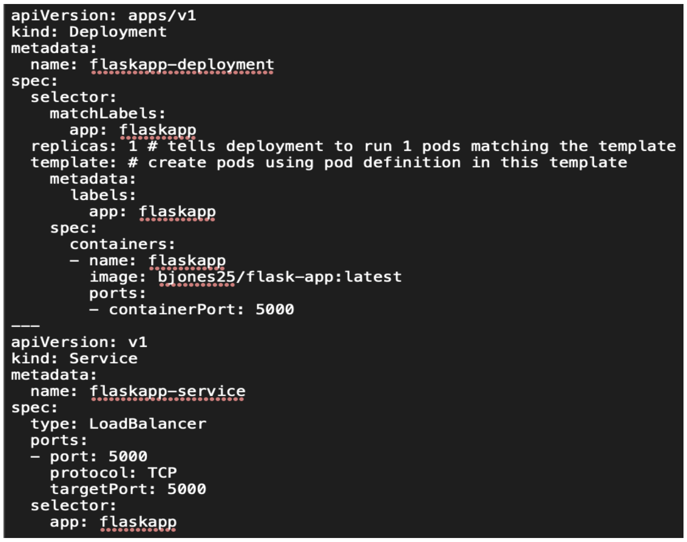
   You create the image using the docker build command(docker build -t flask-app:latest .). You can use the docker images command to make sure your image was created

2. You then need to retag the image by using the docker command (docker tag flask-app bjones25/flask-app:latest)
3. To push the image to Dockerhub I would use the docker command (docker push bjones25/flask-app:latest)
4. To deploy the application in Kubernetes you can do so by creating a cluster with a loadbalancer using: k3d cluster create -p "5000:5000@loadbalancer"



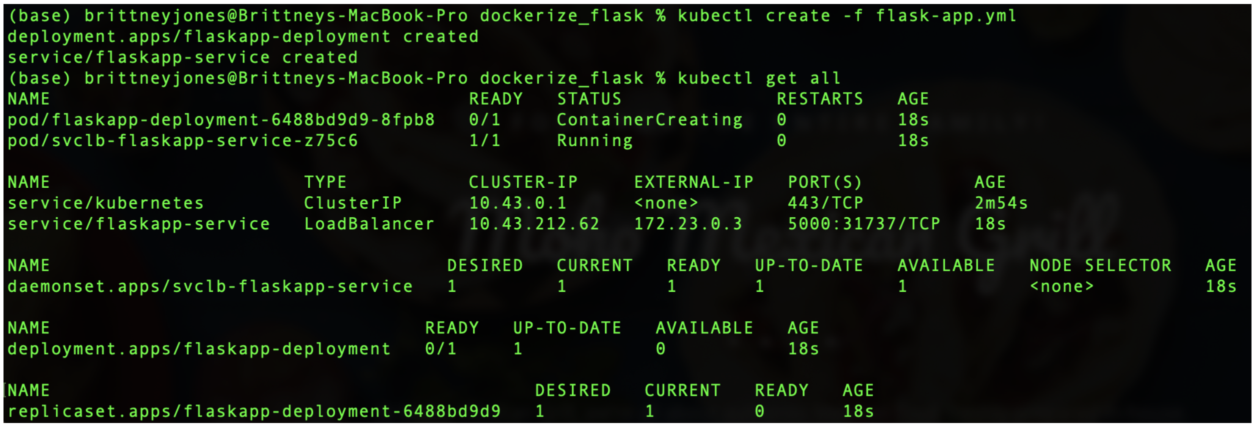
```
(base) brittneyjones@Brittneys-MacBook-Pro dockerize_flask % k3d cluster create flask-app-cluster -p "5000:5000@loadbalancer"
INFO[0000] portmapping '5000:5000' targets the loadbalancer: defaulting to [servers:*:proxy agents:*:proxy]
INFO[0000] Prep: Network
INFO[0000] Created network 'k3d-flask-app-cluster'
INFO[0000] Created volume 'k3d-flask-app-cluster-images'
INFO[0000] Starting new tools node...
INFO[0000] Starting Node 'k3d-flask-app-cluster-tools'
INFO[0001] Creating node 'k3d-flask-app-cluster-server-0'
INFO[0001] Creating LoadBalancer 'k3d-flask-app-cluster-serverlb'
INFO[0001] Using the k3d-tools node to gather environment information
INFO[0002] Starting cluster 'flask-app-cluster'
INFO[0002] Starting servers...
INFO[0002] Starting Node 'k3d-flask-app-cluster-server-0'
INFO[0002] Deleted k3d-flask-app-cluster-tools
INFO[0007] Starting agents...
INFO[0007] Starting helpers...
INFO[0008] Starting Node 'k3d-flask-app-cluster-serverlb'
INFO[0014] Injecting '192.168.65.2 host.k3d.internal' into /etc/hosts of all nodes...
INFO[0014] Injecting record for host.k3d.internal into CoreDNS configmap...
INFO[0018] Cluster 'flask-app-cluster' created successfully!
INFO[0018] You can now use it like this:
kubectl cluster-info
```

5. To create a deployment you first need a .yaml file containing



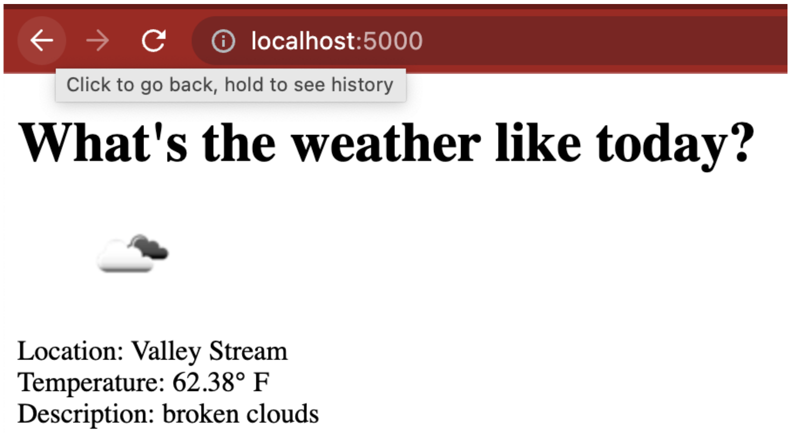
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flaskapp-deployment
spec:
  selector:
    matchLabels:
      app: flaskapp
  replicas: 1 # tells deployment to run 1 pods matching the template
  template: # create pods using pod definition in this template
    metadata:
      labels:
        app: flaskapp
    spec:
      containers:
      - name: flaskapp
        image: bjones25/flask-app:latest
        ports:
        - containerPort: 5000
---
apiVersion: v1
kind: Service
metadata:
  name: flaskapp-service
spec:
  type: LoadBalancer
  ports:
  - port: 5000
    protocol: TCP
    targetPort: 5000
  selector:
    app: flaskapp
```

6. Then you apply the yaml using kubectle create -f filename.yaml. Running kubectl get all shows you the status of your pods, services and deployments



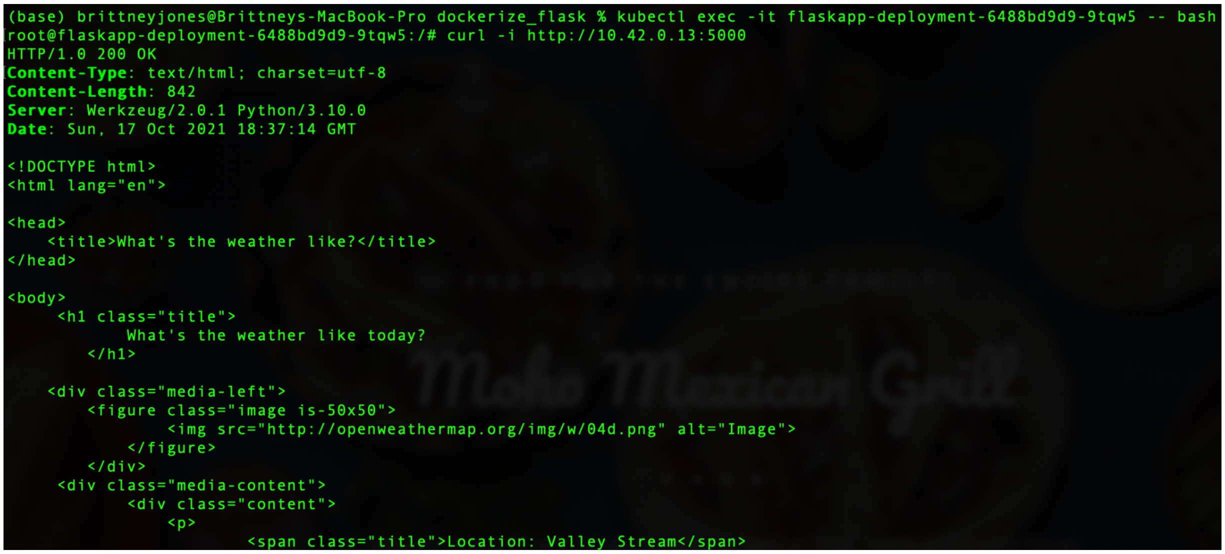
```
(base) brittneyjones@Brittneys-MacBook-Pro dockerize_flask % kubectl create -f flask-app.yml
deployment.apps/flaskapp-deployment created
service/flaskapp-service created
(base) brittneyjones@Brittneys-MacBook-Pro dockerize_flask % kubectl get all
NAME                                     READY   STATUS              RESTARTS   AGE
pod/flaskapp-deployment-6488bd9d9-8fpb8  0/1     ContainerCreating   0          18s
pod/svclb-flaskapp-service-z75c6         1/1     Running             0          18s

NAME                        TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
service/kubernetes          ClusterIP      10.43.0.1      <none>        443/TCP          2m54s
service/flaskapp-service    LoadBalancer   10.43.212.62   172.23.0.3    5000:31737/TCP   18s

NAME                                    DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/svclb-flaskapp-service   1         1         1       1            1           <none>          18s

NAME                                   READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/flaskapp-deployment   0/1     1            0           18s

NAME                                            DESIRED   CURRENT   READY   AGE
replicaset.apps/flaskapp-deployment-6488bd9d9   1         1         0       18s
```

7. To see if your application is running you can go to localhost:5000 or execute command into the container



What's the weather like today?

Location: Valley Stream
Temperature: 62.38° F
Description: broken clouds



```
(base) brittneyjones@Brittneys-MacBook-Pro dockerize_flask % kubectl exec -it flaskapp-deployment-6488bd9d9-9tqw5 -- bash
[root@flaskapp-deployment-6488bd9d9-9tqw5:/# curl -i http://10.42.0.13:5000
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 842
Server: Werkzeug/2.0.1 Python/3.10.0
Date: Sun, 17 Oct 2021 18:37:14 GMT

<!DOCTYPE html>
<html lang="en">

<head>
    <title>What's the weather like?</title>
</head>

<body>
     <h1 class="title">
            What's the weather like today?
        </h1>

    <div class="media-left">
        <figure class="image is-50x50">
                <img src="http://openweathermap.org/img/w/04d.png" alt="Image">
            </figure>
        </div>
     <div class="media-content">
            <div class="content">
                <p>
                        <span class="title">Location: Valley Stream</span>
```

8. When finished you should delete the service(kubectl delete service servicename), deployment(kubectl delete deployment deploymentname) and cluster(k3d cluster delete clustername)