LAB UI MATERIAL

DOCKER AND OPENSHIFT

---

1. DOCKER PLAYGROUND

Use the below link for our preconfigured Docker instance
https://labs.play-with-docker.com

a. **To List all the Images:-**
   docker images

b. **Pull our docker project from github and build the docker image:-**
   git clone https://github.com/ashwin25/kube-docker-demo.git

c. cd kube-docker-demo
   docker build –tag docker-service:[tagname] eg 1.0
   (To

d. **To push the built docker image to DockerHub (public docker image repository):-**
   docker login –u ashwinprakash
   docker tag docker-service:1.0 ashwinprakash25/docker-service:1.0
   docker push ashwinprakash/docker-service:1.0

   Due to docker limits Ashwin is using quay.io

   docker login quay.io

   Username: ashwinprakash25

   Password:

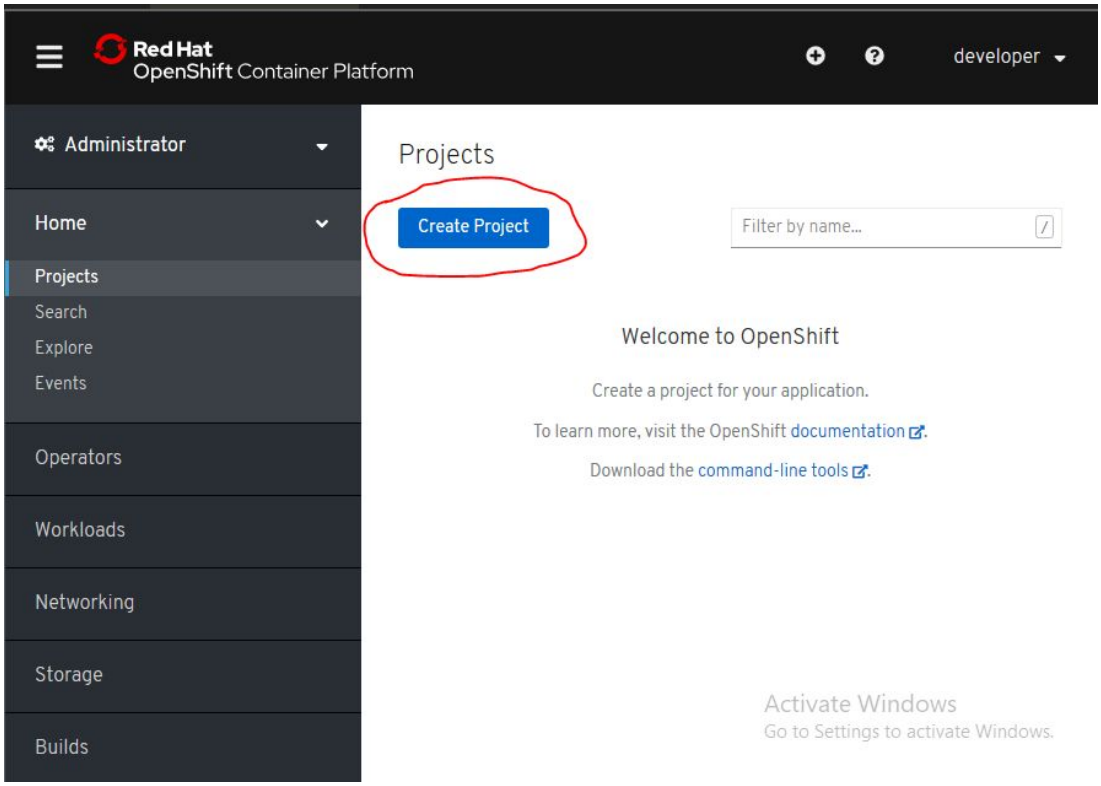
   docker tag docker-service:1.0 quay.io/ashwinprakash25/docker-service:1.0
   docker push quay.io/ashwinprakash25/docker-service:1.0

2. OPENSHIFT BASIC FEATURES

Use the below link for the instance with preconfigured Openshift environment
https://www.openshift.com/learn/courses/playground

a. **Steps to Login and create Namespace (project) in Openshift:**
   Go to Console
   Login using username – developer, password – developer

   Click on **Create Project**, assign a **Name** (eg :- demo-project) and click on **Create**

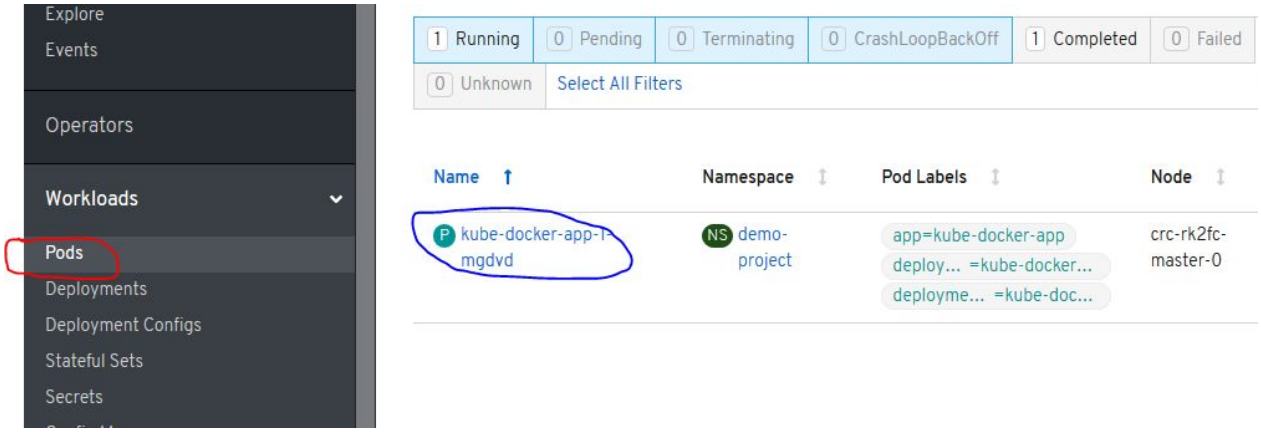


b. **Pull the image from DockerHub and deploy in openshift**
   oc new-app adi4196/kube-docker-app:40904 (My Image which pushed in DockerHub)

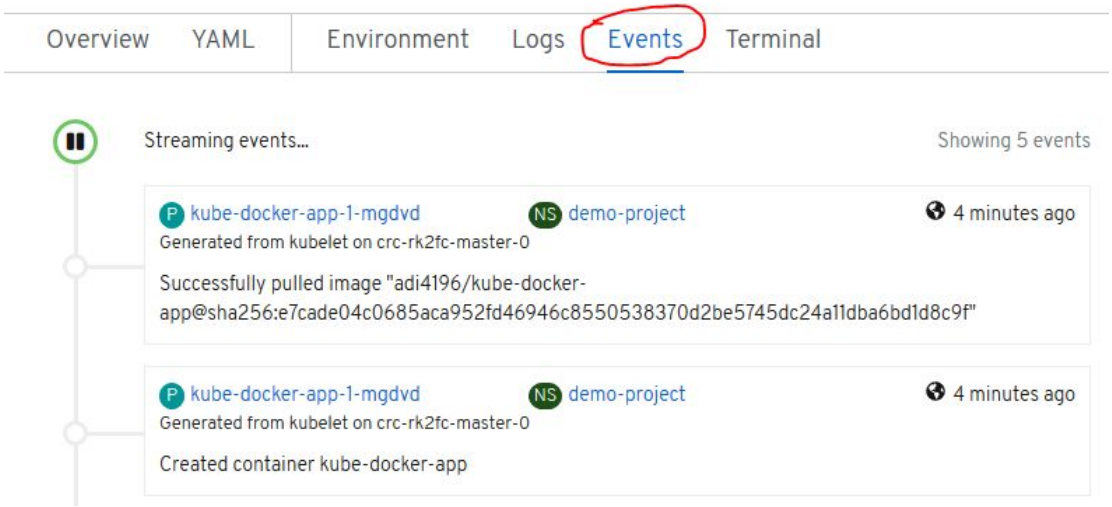
(**Note:- We can do the above step using UI but this one command reduces our work to deploy the image and create a service in just one command hence we will be using command line (CLI) only for this step)**

c. **Check the events of the running deployment (Process of deployment)**
   Go to Workloads -□ Pods

   Click on the Pod name as shown below



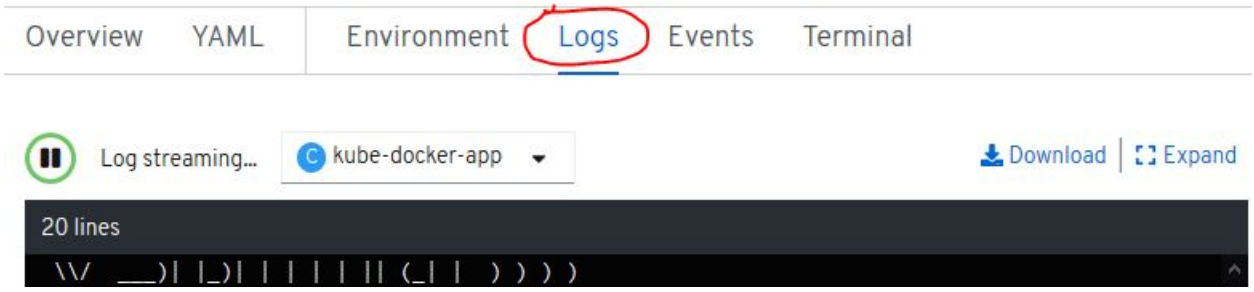
Click on the Events Section at the top as shown below



d. **Check the logs of your application running in the pod**
Go to Workloads - Pods

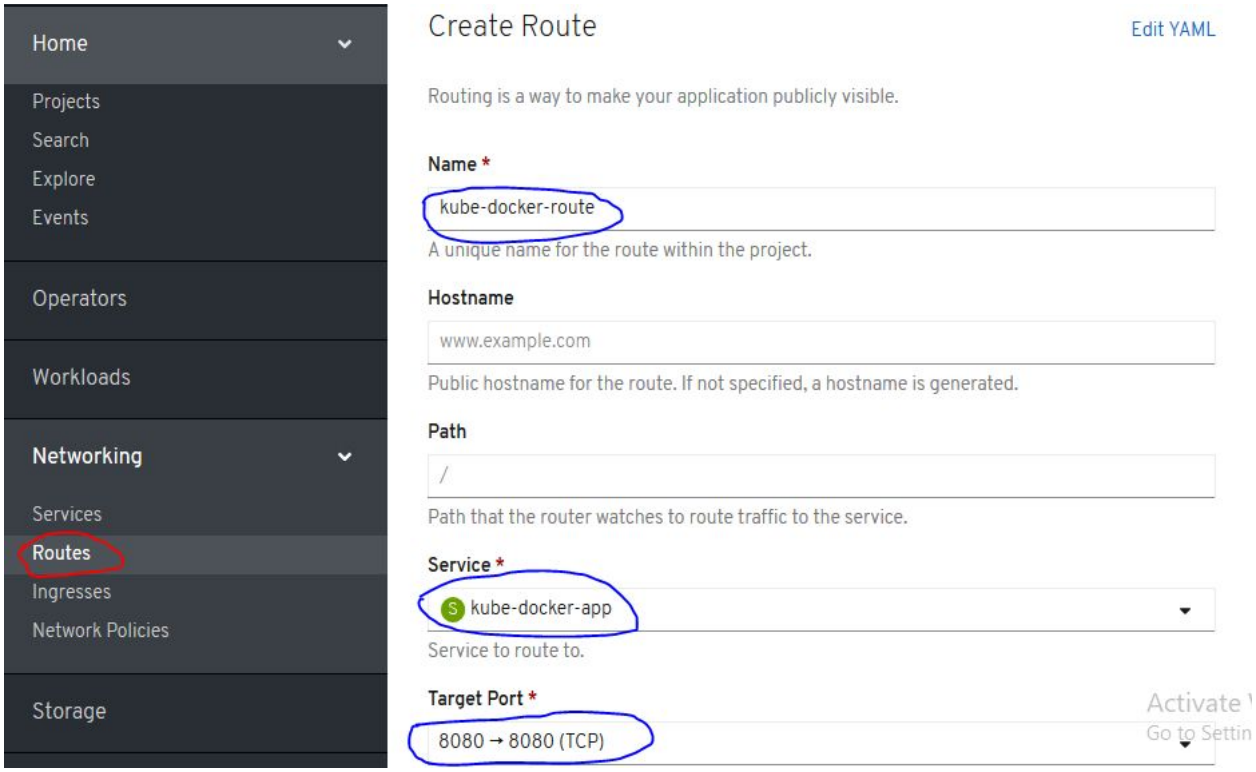
Click on the Pod name as shown below

Click on the Events Section at the top as shown below

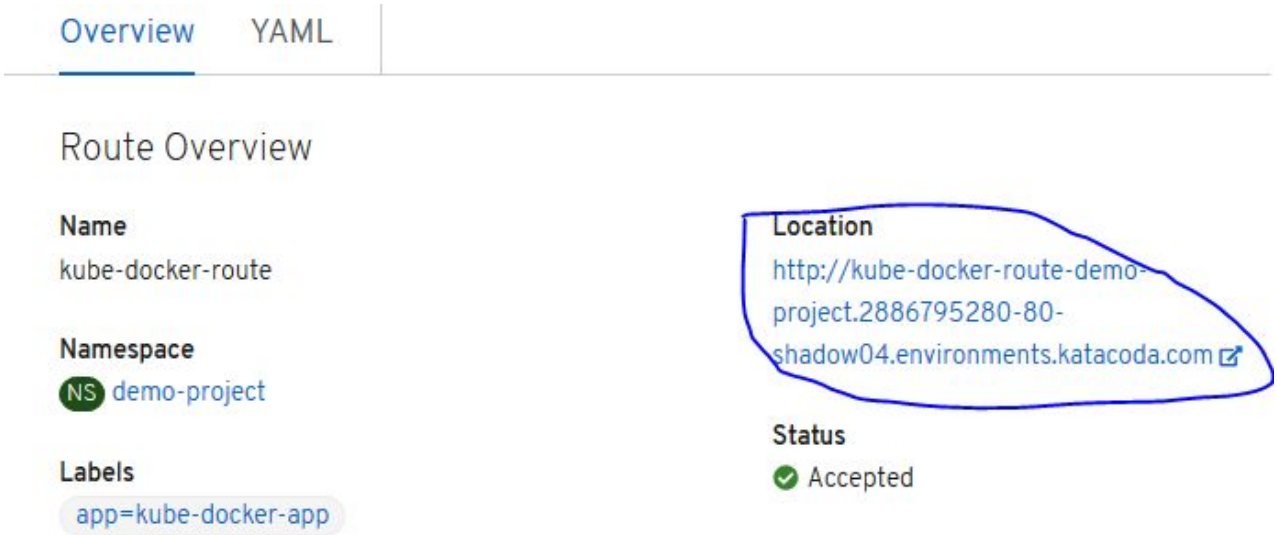


e. **Hit the service exposed for your application using Curl command or using the browser**
Go to Networking Routes

Click on Create Route, Give the Name, Select the Service created, Assign port 8080 -> 8080(TCP) as shown below



Click on the below hostname which will take you to the browser

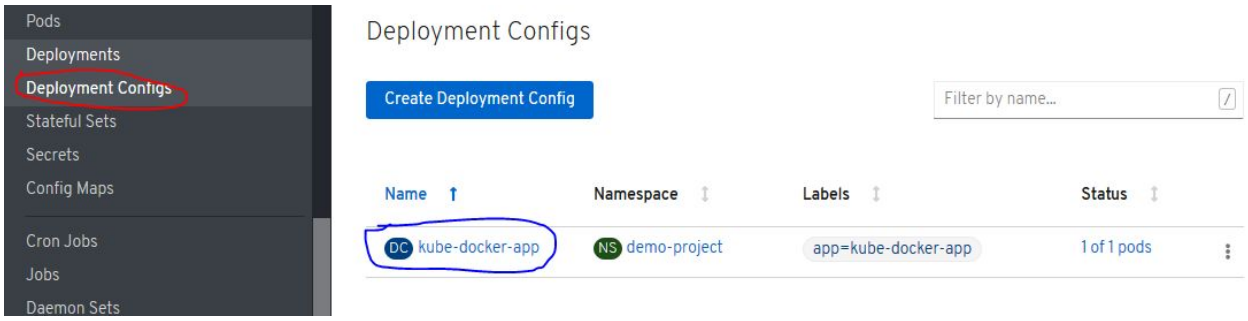


3. OPENSHIFT ADVANCED FEATURES

a. **Roll-up or Roll-down a pod of the deployed application**

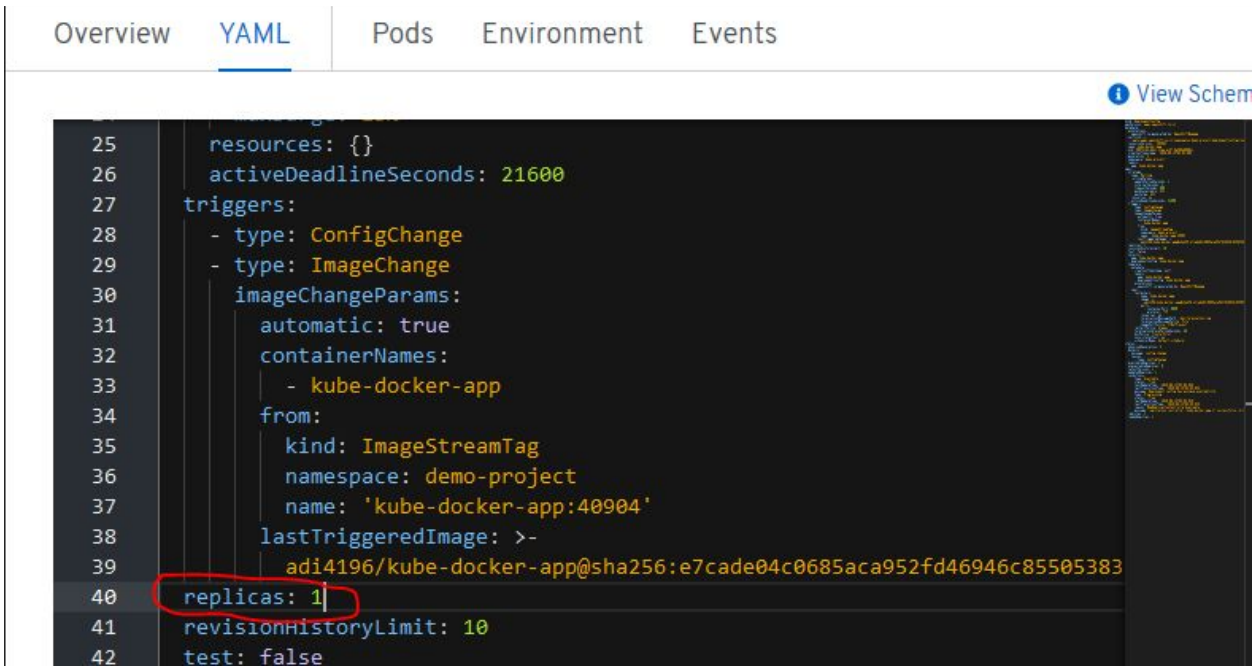
Go to Workloads Deployment Configs

Click on the Name **'kube-docker-app'** shown below



Go to YAML section and update the 'replicas' value from 1 to 2 in the editor section as shown below

Click on Save and Reload and then Go to Workloads Pods.

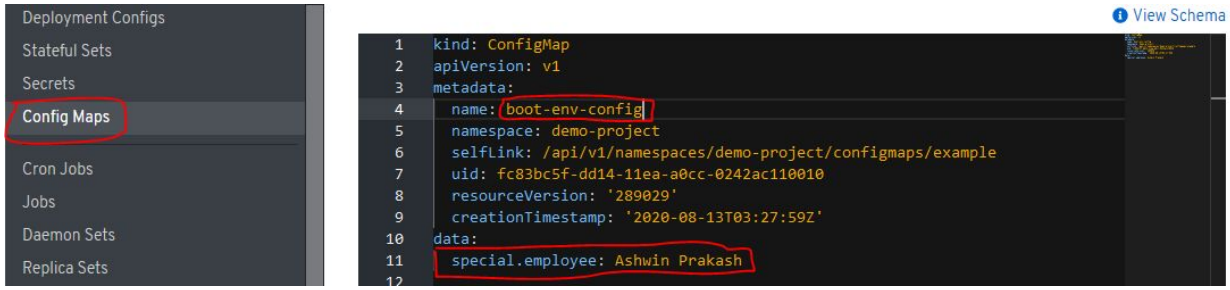


b. **Create Config Map as an Environment Variable**

Go to Workloads Config Maps, Click on **Create Config Map**

In the Editor section shown below, add the red highlighted text starting the tab space shown at line no 7 and change the name at line no 4 as shown.

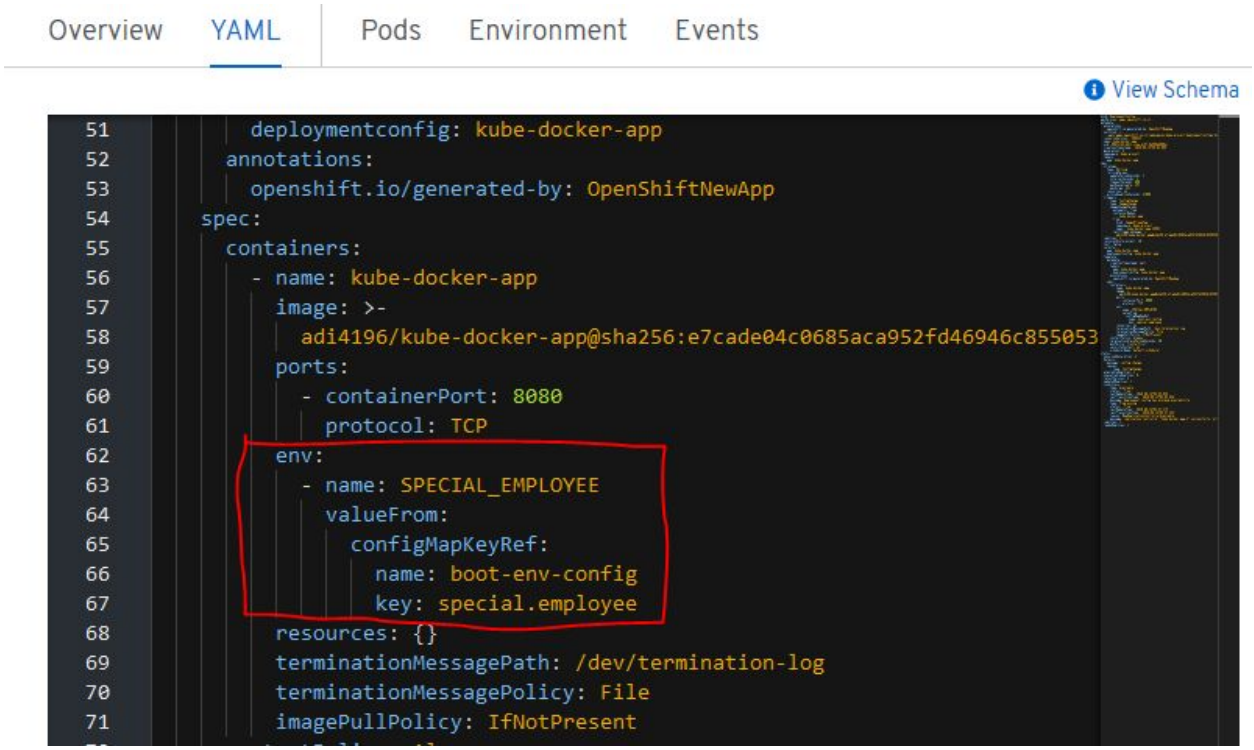
Click on **Create**



Go to Workloads □ Deployment Configs □ Name □ YAML

Update the highlighted changes as shown below in the Deployment Config Editor

(Note:- Please maintain the indentation as shown in the below editor)



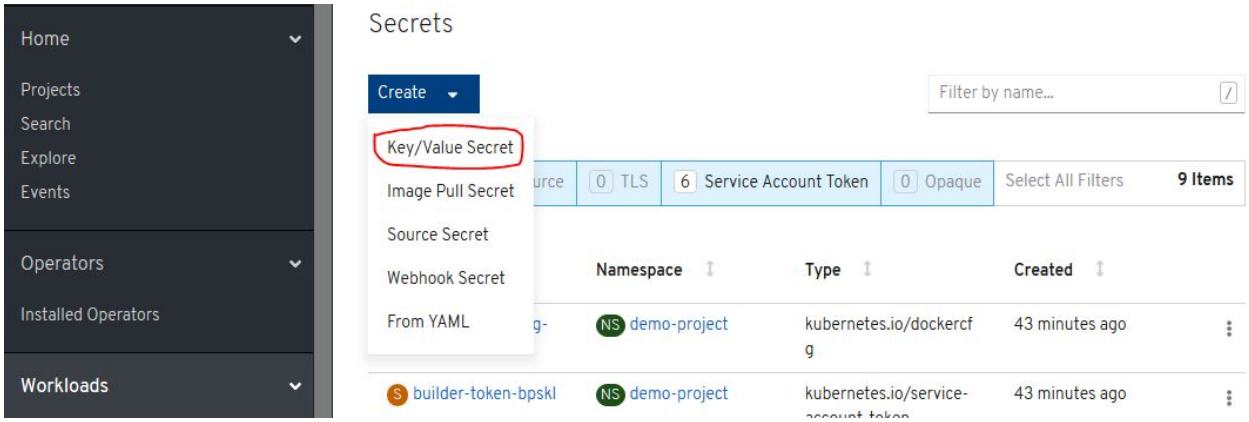
Use the same HostName which we created in the Route section and hit the below Url :-

'HostName/specialEmp'

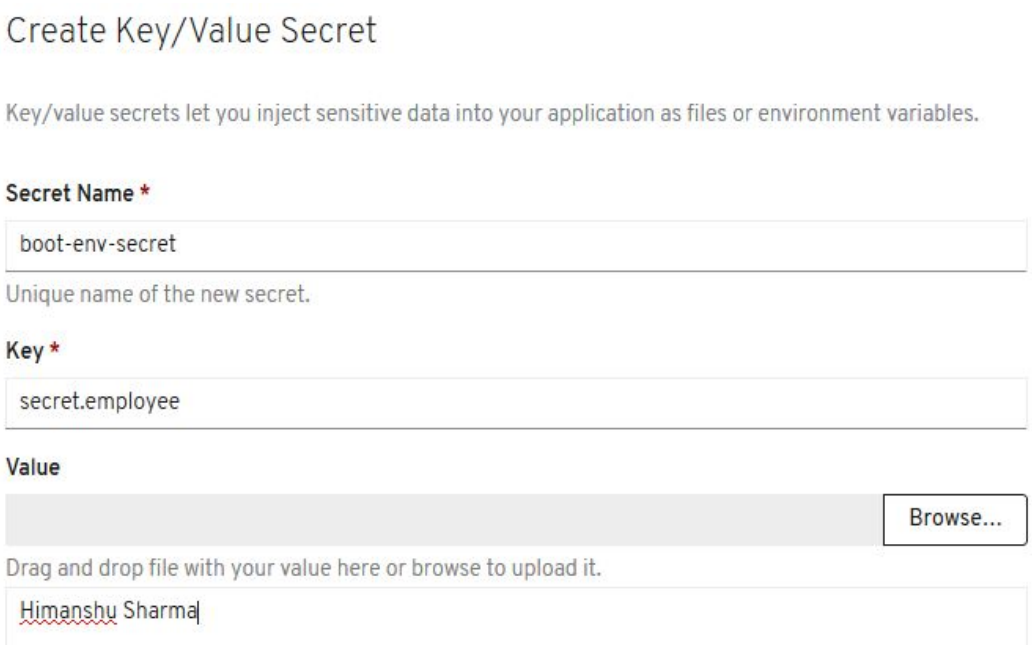
c. Create Secret as an Environment Variable

Go to Workloads Secrets,

Select Key/Value Pair as the type as shown below.



Put the details as shown below and then click on Create



Go to Workloads Deployment Configs Name YAML

Update the highlighted changes as shown below in the Deployment Config Editor

(Note:- Please maintain the indentation as shown in the below editor)

Overview YAML Pods Environment Events

View Schema

```
56          - name: kube-docker-app
57            image: >-
58              adi4196/kube-docker-app@sha256:e7cade04c0685aca952fd46946c855053
59            ports:
60              - containerPort: 8080
61                protocol: TCP
62            env:
63              - name: SECRET_EMPLOYEE
64                valueFrom:
65                  secretKeyRef:
66                    name: boot-env-secret
67                    key: secret.employee
68            resources: {}
69            terminationMessagePath: /dev/termination-log
70            terminationMessagePolicy: File
71            imagePullPolicy: IfNotPresent
```

Use the same HostName which we created in the Route section and hit the below Url :-

‘HostName/secretEmp’