

DEPLOY IN KUBERNETES CLUSTER

TEAM ID	PNT2022TMID38924
PROJECT NAME	CUSTOMER CARE REGISTRY
DATE	17.11.2022

KUBERNETES:

DRASHBOARD-ADMINUSER:

```
    apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
---
apiVersion: v1
kind: Secret
metadata:
  name: admin-user-token
  namespace: kubernetes-dashboard
  annotations:
    kubernetes.io/service-account.name: admin-user
type: kubernetes.io/service-account-token
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
```

FLASK DEPLOYMENT:

```
    apiVersion: apps/v1
    kind: Deployment
  metadata:
    name: flask-app

  spec:
    replicas: 3
    selector:
      matchLabels:
        app: flask-app
    template:
      metadata:
        labels:
```

DEPLOY IN KUBERNETES CLUSTER

```
app: flask-app

spec:
  containers:
  - name: webpage
    image: rajkiranss/flask
    imagePullPolicy: Never
    ports:
    - containerPort: 5000
      protocol: TCP
```

FLASK INGRESS:

```
    apiVersion: networking.k8s.io/v1
    kind: Ingress
    metadata:
      name: flask-app-ingress
    annotations:
      kubernetes.io/ingress.class: nginx
      nginx.ingress.kubernetes.io/ssl-redirect: "false"

    spec:
      # ingressClassName: nginx
      rules:
      - http:
          paths:
          - backend:
              service:
                name: flask-app-service
                port:
                  number: 5000

            path: /
              pathType: Prefix
```

FLASK SERVICE:

```
    apiVersion: v1
    kind: Service
    metadata:
      name: flask-app-service
    spec:
      type: ClusterIP
      ports:
      - port: 5000
      selector:
        app: flask-app
```

IBM DEPLOYMENT:

DEPLOY IN KUBERNETES CLUSTER

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-deploy

spec:
  replicas: 3
  selector:
    matchLabels:
      app: flask-deploy
  template:
    metadata:
      labels:
        app: flask-deploy

    spec:
      containers:
        - name: web
          image: jp.icr.io/webpage/web
          imagePullPolicy: Always
          ports:
            - containerPort: 5000
              protocol: TCP
```