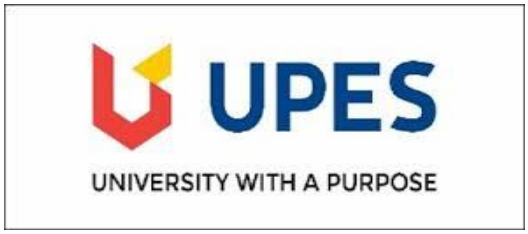

Lab Experiment – 13

NAME -Divyaansh Jain

ROLL NO – R171218040

SAP ID – 500067134

COURSE – B. Tech CSE- DevOps Batch 1

SUBJECT – Application Containerization

SEMESTER – 6th semester

Submitted To:

Dr. Hitesh Kumar Sir

Deploying Pods and Services on minikube

- Use command **minikube start** to start your minikube cluster.

```
D:\CSE>minikube start
* minikube v1.9.2 on Microsoft Windows 10 Home Single Language 10.0.18363 Build 18363
* Using the virtualbox driver based on existing profile
* Starting control plane node m01 in cluster minikube
* Restarting existing virtualbox VM for "minikube" ...
* Preparing Kubernetes v1.18.0 on Docker 19.03.8 ...
* Enabling addons: dashboard, default-storageclass, storage-provisioner
* Done! kubectl is now configured to use "minikube"
```

- For creating pods, we will create a deployment. Deployment is used to manage and monitor the pods. For launching deployment, we will create a yaml file. We will use **kubectl create -f <filename>** to run these yaml files.

```
apiVersion: apps/v1
kind: Deployment
metadata:
    name: grafana-dpl
    labels:
      run: grafana
spec:
    replicas: 1
    selector:
      matchLabels:
        run: grafana
    template:
      metadata:
        labels:
          run: grafana
      spec:
        containers:
        - name: grafana-container
          image: grafana/grafana:latest

          ports:
          - containerPort: 3000

          volumeMounts:
          - name: grafana-volume
            mountPath: /var/lib/grafana

        volumes:
        - name: grafana-volume
          persistentVolumeClaim:
           claimName: pvc-grafana
```

- Similarly, to create a service, we will create a yaml file and use kubectl command to launch it.

```
apiVersion: v1
kind: Service
metadata:
    name: grafana-svc
    labels:
      run: grafana
spec:
    selector:
      run: grafana
    type: NodePort
    ports:
    - nodePort: 30009
      port: 3000
      targetPort: 3000
      name: port-grafana
```

- After launching the yaml files, use command **kubectl get pods** to check all the running pods.

```
D:\CSE\kubernetes>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
grafana-dpl-77fb7d89d7-xnnhw        1/1     Running   0          7m51s
html-deployment-6ffb959974-4zgg8    1/1     Running   2          264d
php-deployment-577679d5c-t6xl2      1/1     Running   2          264d
prometheus-dpl-76bb5fbf9f-q64pj     1/1     Running   0          7m51s
```

- Use command **kubectl get deployment** to check all the active deployments.

```
D:\CSE\kubernetes>kubectl get deployment
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
grafana-dpl       1/1     1            1           2m39s
html-deployment   1/1     1            1           273d
php-deployment    1/1     1            1           273d
prometheus-dpl    0/1     1            0           2m39s
```

- Use command **kubectl get svc** to check all the active services.

```
D:\CSE\kubernetes>kubectl get svc
NAME                TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
grafana-svc         NodePort    10.97.175.108    <none>        3000:30009/TCP   9m21s
html-deployment     NodePort    10.108.233.234   <none>        80:31084/TCP     273d
kubernetes          ClusterIP   10.96.0.1        <none>        443/TCP          273d
php-deployment      NodePort    10.103.63.117    <none>        80:30466/TCP     273d
prometheus-svc      NodePort    10.111.193.26    <none>        9090:30008/TCP   9m21s
```

- We can also check these using minikube dashboard.