Lab 2 K8s

1. Create a StatefulSet named web-statefulset with 2 replicas using the nginx image. The StatefulSet should have a Headless Service named web-service

statefulset-pod.yml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: web-service
spec:
  clusterIP: None
  selector:
    app: web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web-statefulset
spec:
  serviceName: web-service
  replicas: 2
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
        - name: nginx
          image: nginx
```

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl apply -f statefulset-pod.yml
service/web-service unchanged
statefulset.apps/web-statefulset configured
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl get statefulset
NAME             READY   AGE
web-statefulset   2/2     3m16s
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl get pods
NAME               READY   STATUS    RESTARTS   AGE
web-statefulset-0   1/1     Running   0          2m50s
web-statefulset-1   1/1     Running   0          2m54s
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$
```

2. How many DaemonSets are created in the cluster in all namespaces?

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl get daemonsets --all-namespaces
NAMESPACE     NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kube-system   kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   29h
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$
```

3. Create a DaemonSet named “nginx” with image “nginx”.

```
! nginx-daemonset.yml > {} spec > {} template > {} spec > [ ] containers > {} 0
      io.k8s.api.apps.v1.DaemonSet (v1@daemonset.json)
  1   apiVersion: apps/v1
  2   kind: DaemonSet
  3   metadata:
  4     name: nginx
  5   spec:
  6     selector:
  7       matchLabels:
  8         app: nginx
  9     template:
 10       metadata:
 11         labels:
 12           app: nginx
 13       spec:
 14         containers:
 15           - name: nginx
 16             image: nginx
 17
```

4. How many pods have been created within the nginx DaemonSet and why?

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl apply -f nginx-daemonset.yml
daemonset.apps/nginx created
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl get daemonsets
NAME    DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
nginx   1         1         1       1            1           <none>          21s
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
nginx-gpckz         1/1     Running   0          25s
web-statefulset-0   1/1     Running   0          6m
web-statefulset-1   1/1     Running   0          6m4s
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$
```

Only one pod was created because because DaemonSets run one pod on the node so the number of pods should be equal to that of the nodes.

6. Deploy a DaemonSet for FluentD Logging. Use the given specifications.
➔ Name: elasticsearch
➔ Image: k8s.gcr.io/fluentd-elasticsearch:1.20

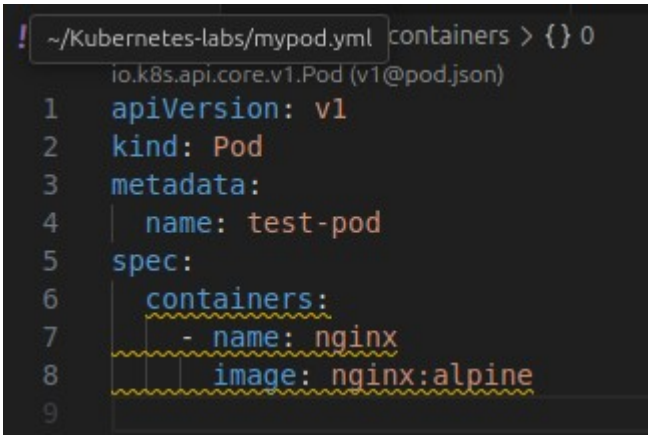
```
! fluentid-daemonset.yml > {} spec > {} template > {} spec > [ ] containers > {} 0
      io.k8s.api.apps.v1.DaemonSet (v1@daemonset.json)
  1   apiVersion: apps/v1
  2   kind: DaemonSet
  3   metadata:
  4     name: elasticsearch
  5   spec:
  6     selector:
  7       matchLabels:
  8         app: fluentd
  9     template:
 10       metadata:
 11         labels:
 12           app: fluentd
 13       spec:
 14         containers:
 15           - name: fluentd
 16             image: k8s.gcr.io/fluentd-elasticsearch:1.20
```

7. Deploy a pod named nginx-pod using the nginx:alpine image with the labels set to tier=backend.

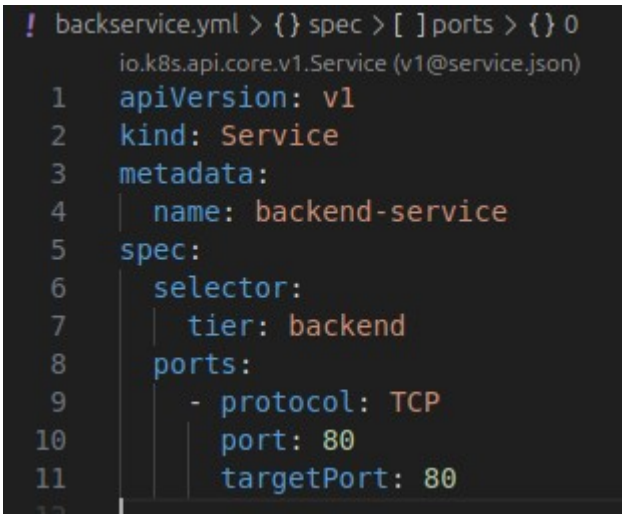
```
! nginx-back.yml > {} spec > [ ] containers > {} 0
      io.k8s.api.core.v1.Pod (v1@pod.json)
  1   apiVersion: v1
  2   kind: Pod
  3   metadata:
  4     name: nginx-pod
  5     labels:
  6       tier: backend
  7   spec:
  8     containers:
  9       - name: nginx
 10         image: nginx:alpine
 11
```

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
elasticsearch-cxqs6   1/1     Running   0          2m44s
nginx-gpckz         1/1     Running   0          20m
nginx-pod           1/1     Running   0          72s
web-statefulset-0   1/1     Running   0          26m
web-statefulset-1   1/1     Running   0          26m
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$
```

8. Deploy a test pod using the nginx:alpine image.



9. Create a service backend-service to expose the backend application within the cluster on port 80




```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl describe pod nginx-pod | grep IP
IP:               10.244.0.65
IPs:
  IP:  10.244.0.65
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl describe service backend-service | grep Endpoints
Endpoints:         10.244.0.65:80
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$
```

10. Try to curl the backend-service from the test pod. What is the response?

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl exec -it test-pod -- sh
/ # curl backend-service
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
```

11. Create a deployment named web-app using the image nginx with 2 replicas

webapp.yml > {} spec > {} template > {} spec > [ ] containers > {} 0
io.k8s.api.apps.v1.Deployment (v1@deployment.json)

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web-app
  template:
    metadata:
      labels:
        app: web-app
    spec:
      containers:
        - name: nginx
          image: nginx
```

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
web-app   2/2     2            2           8s
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$
```

12. Expose the web-app as service web-app-service application on port 80 and nodeport 30082 on the nodes on the cluster

webappservice.yml > {} spec > [ ] ports > {} 0
io.k8s.api.core.v1.Service (v1@service.json)

```yaml
apiVersion: v1
kind: Service
metadata:
  name: web-app-service
spec:
  selector:
    app: web-app
  type: NodePort
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30082
```

13.Access the web app from the node

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ minikube ip
192.168.59.100
^[[Ashehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ curl http://192.168.59.100:30082
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
```

14.Cread a deployment nginx with pod labels
➔ app:nginx
➔ tier:frontend
and set-based selectors on replicasets that allow filtering objects based on specific conditions. Given the valid operators In, Exists.

nginx-deployment.yml > {} spec > {} template > {} spec > [ ] containers > {} 0

io.k8s.api.apps.v1.Deployment (v1@deployment.json)

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 2
  selector:
    matchExpressions:
      - key: tier
        operator: In
        values: ["frontend"]
      - key: app
        operator: Exists
  template:
    metadata:
      labels:
        app: nginx
        tier: frontend
    spec:
      containers:
        - name: nginx
          image: nginx
```

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl apply -f nginx-deployment.yml
deployment.apps/nginx created
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
nginx     2/2     2            2           6s
web-app   2/2     2            2           12m
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$
```

15.When can we use the Loadbalancer service?

We use a Loadbalancer service when we want to expose a service externally and need to be accessed from the internet and to distribute incoming traffic from the outside on multiple pods.