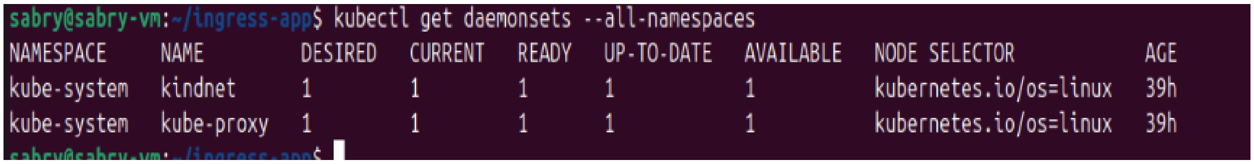
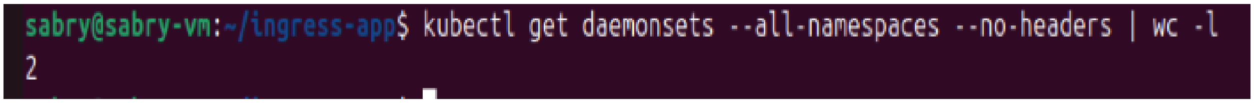
1- **How many DaemonSets are created in the cluster in all namespaces?**

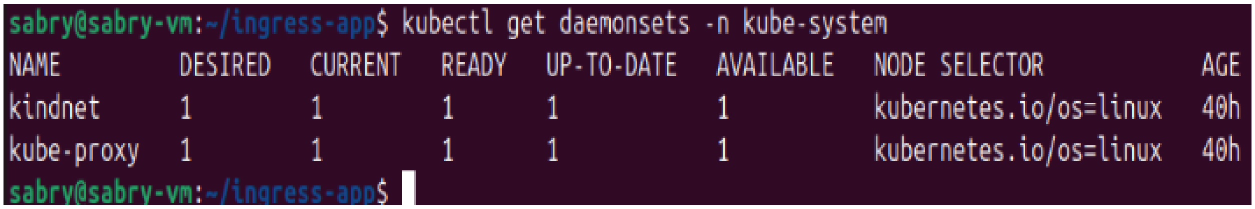


```
sabry@sabry-vm:~/ingress-app$ kubectl get daemonsets --all-namespaces
NAMESPACE     NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kube-system   kindnet      1         1         1       1            1           kubernetes.io/os=linux   39h
kube-system   kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   39h
sabry@sabry-vm:~/ingress-app$
```



```
sabry@sabry-vm:~/ingress-app$ kubectl get daemonsets --all-namespaces --no-headers | wc -l
2
```

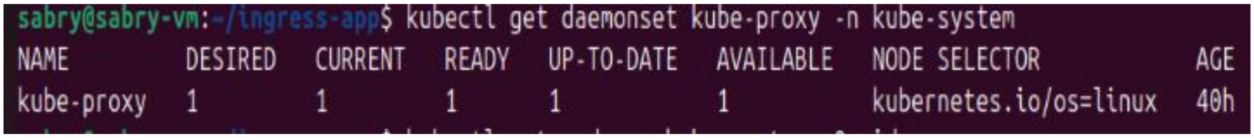
2- **what DaemonSets exist on the kube-system namespace?**



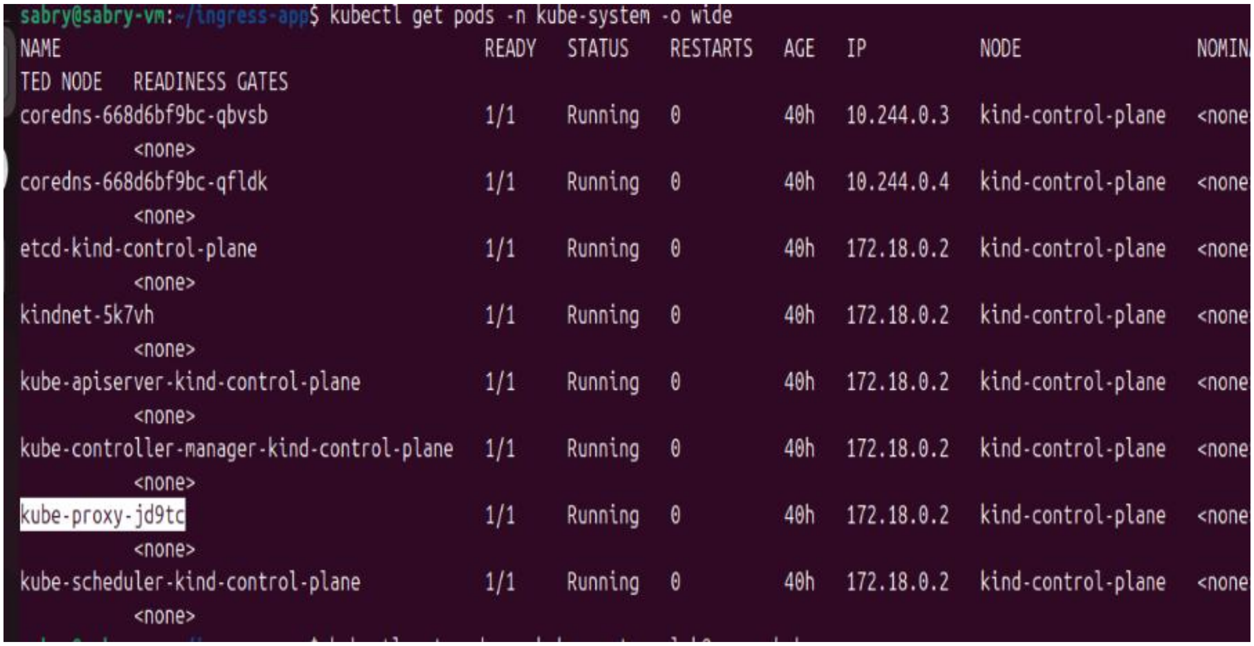
```
sabry@sabry-vm:~/ingress-app$ kubectl get daemonsets -n kube-system
NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kindnet      1         1         1       1            1           kubernetes.io/os=linux   40h
kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   40h
sabry@sabry-vm:~/ingress-app$
```

3- What is the image used by the POD deployed by the kube-proxy

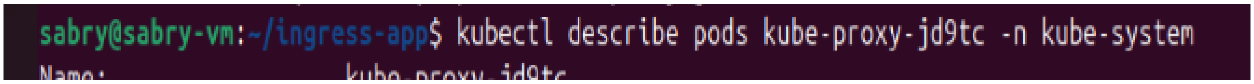
DaemonSet



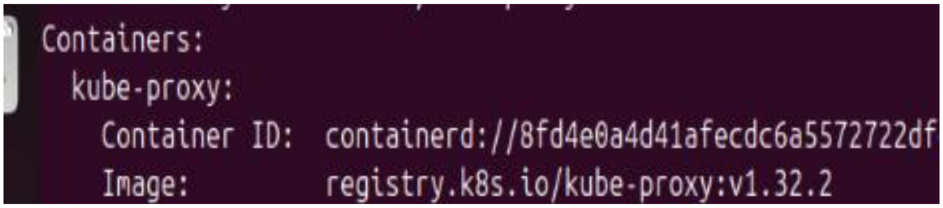
```
sabry@sabry-vm:~/ingress-app$ kubectl get daemonset kube-proxy -n kube-system
NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   40h
```



```
sabry@sabry-vm:~/ingress-app$ kubectl get pods -n kube-system -o wide
NAME                                         READY   STATUS    RESTARTS   AGE   IP           NODE                 NOMIN
TED NODE   READINESS GATES
coredns-668d6bf9bc-qbvsb                     1/1     Running   0          40h   10.244.0.3   kind-control-plane   <none
           <none>
coredns-668d6bf9bc-qfldk                     1/1     Running   0          40h   10.244.0.4   kind-control-plane   <none
           <none>
etcd-kind-control-plane                      1/1     Running   0          40h   172.18.0.2   kind-control-plane   <none
           <none>
kindnet-5k7vh                                1/1     Running   0          40h   172.18.0.2   kind-control-plane   <none
           <none>
kube-apiserver-kind-control-plane            1/1     Running   0          40h   172.18.0.2   kind-control-plane   <none
           <none>
kube-controller-manager-kind-control-plane   1/1     Running   0          40h   172.18.0.2   kind-control-plane   <none
           <none>
kube-proxy-jd9tc                             1/1     Running   0          40h   172.18.0.2   kind-control-plane   <none
           <none>
kube-scheduler-kind-control-plane            1/1     Running   0          40h   172.18.0.2   kind-control-plane   <none
           <none>
```



```
sabry@sabry-vm:~/ingress-app$ kubectl describe pods kube-proxy-jd9tc -n kube-system
Name:             kube-proxy-jd9tc
```



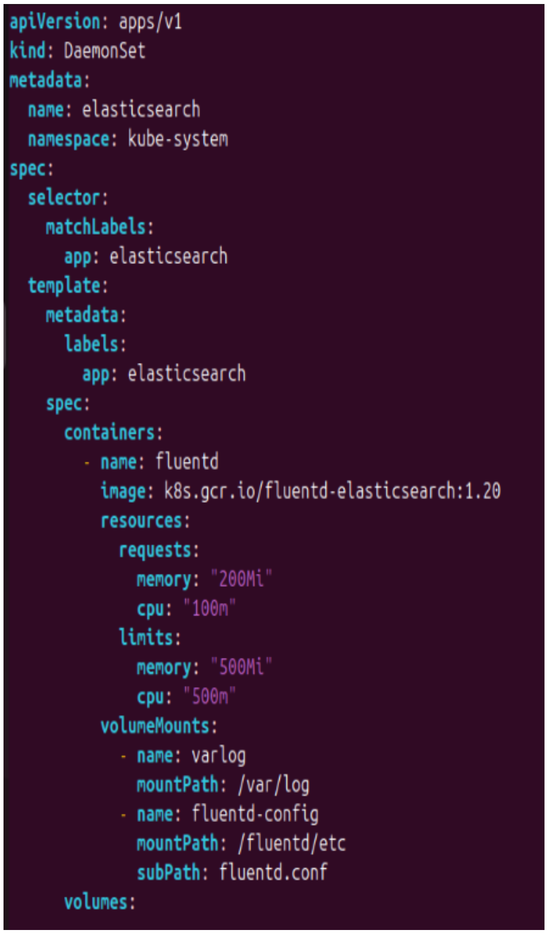
```
Containers:
  kube-proxy:
    Container ID:  containerd://8fd4e0a4d41afecdc6a5572722df
    Image:         registry.k8s.io/kube-proxy:v1.32.2
```

4- Deploy a DaemonSet for FluentD Logging. Use the given specifications.

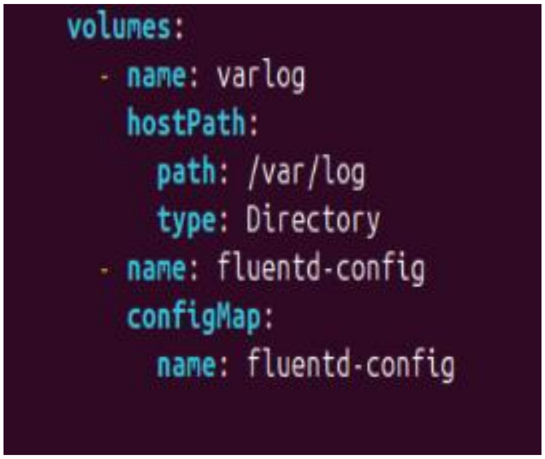
Name: elasticsearch

Namespace: kube-system

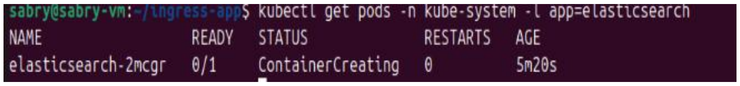
Image: k8s.gcr.io/fluentd-elasticsearch:1.20



```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: elasticsearch
  namespace: kube-system
spec:
  selector:
    matchLabels:
      app: elasticsearch
  template:
    metadata:
      labels:
        app: elasticsearch
    spec:
      containers:
        - name: fluentd
          image: k8s.gcr.io/fluentd-elasticsearch:1.20
          resources:
            requests:
              memory: "200Mi"
              cpu: "100m"
            limits:
              memory: "500Mi"
              cpu: "500m"
          volumeMounts:
            - name: varlog
              mountPath: /var/log
            - name: fluentd-config
              mountPath: /fluentd/etc
              subPath: fluentd.conf
      volumes:
```

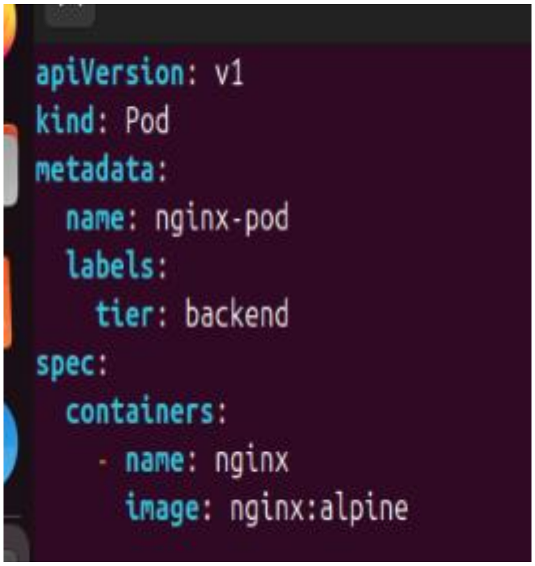


```
  volumes:
    - name: varlog
      hostPath:
        path: /var/log
        type: Directory
    - name: fluentd-config
      configMap:
        name: fluentd-config
```

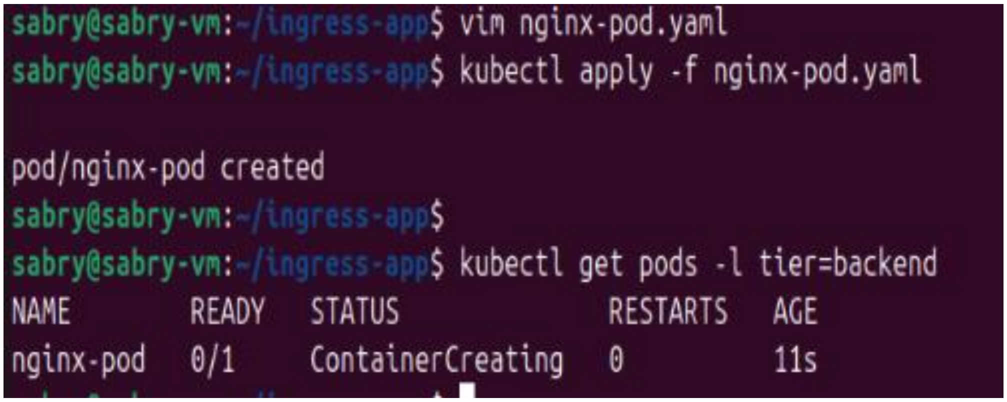


```
sabry@sabry-vm:~/ingress-app$ kubectl get pods -n kube-system -l app=elasticsearch
NAME                  READY   STATUS              RESTARTS   AGE
elasticsearch-2mcgr   0/1     ContainerCreating   0          5m20s
```

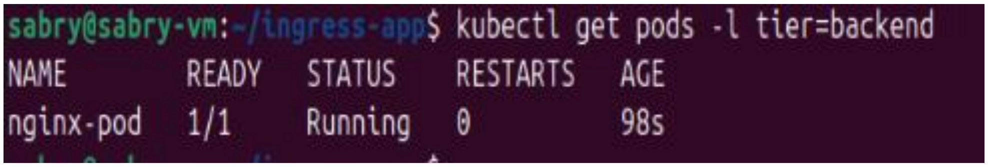
5- Deploy a pod named nginx-pod using the nginx:alpine image with the labels set to tier=backend.



```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    tier: backend
spec:
  containers:
    - name: nginx
      image: nginx:alpine
```

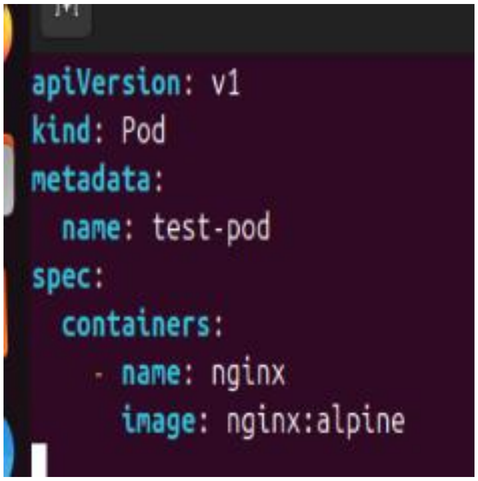


```
sabry@sabry-vm:~/ingress-app$ vim nginx-pod.yaml
sabry@sabry-vm:~/ingress-app$ kubectl apply -f nginx-pod.yaml

pod/nginx-pod created
sabry@sabry-vm:~/ingress-app$
sabry@sabry-vm:~/ingress-app$ kubectl get pods -l tier=backend
NAME        READY   STATUS              RESTARTS   AGE
nginx-pod   0/1     ContainerCreating   0          11s
```

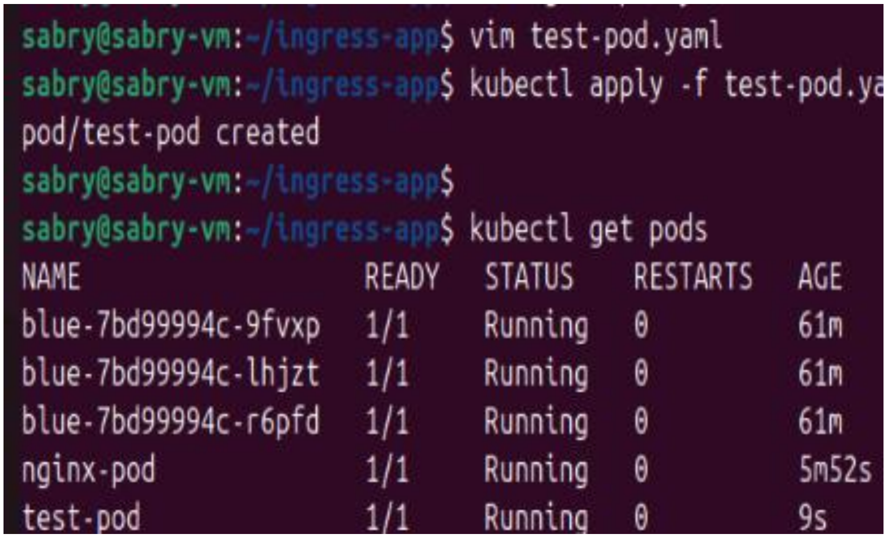


```
sabry@sabry-vm:~/ingress-app$ kubectl get pods -l tier=backend
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          98s
```

6- Deploy a test pod using the nginx:alpine image.

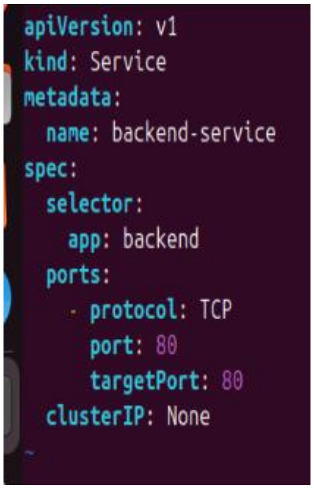


```
apiVersion: v1
kind: Pod
metadata:
  name: test-pod
spec:
  containers:
    - name: nginx
      image: nginx:alpine
```

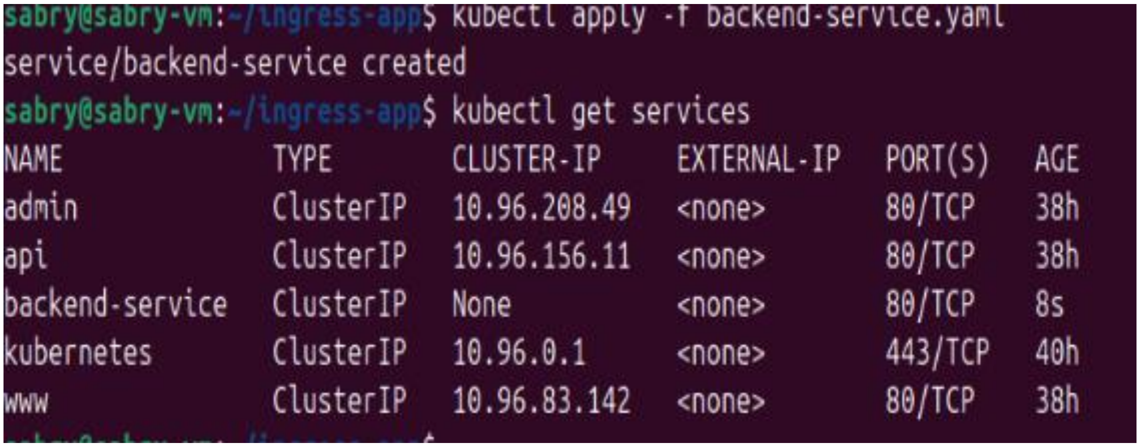


```
sabry@sabry-vm:~/ingress-app$ vim test-pod.yaml
sabry@sabry-vm:~/ingress-app$ kubectl apply -f test-pod.ya
pod/test-pod created
sabry@sabry-vm:~/ingress-app$
sabry@sabry-vm:~/ingress-app$ kubectl get pods
NAME                   READY   STATUS    RESTARTS   AGE
blue-7bd99994c-9fvxp   1/1     Running   0          61m
blue-7bd99994c-lhjzt   1/1     Running   0          61m
blue-7bd99994c-r6pfd   1/1     Running   0          61m
nginx-pod              1/1     Running   0          5m52s
test-pod               1/1     Running   0          9s
```

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

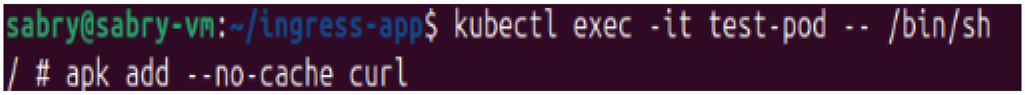


```
apiVersion: v1
kind: Service
metadata:
  name: backend-service
spec:
  selector:
    app: backend
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  clusterIP: None
~
```

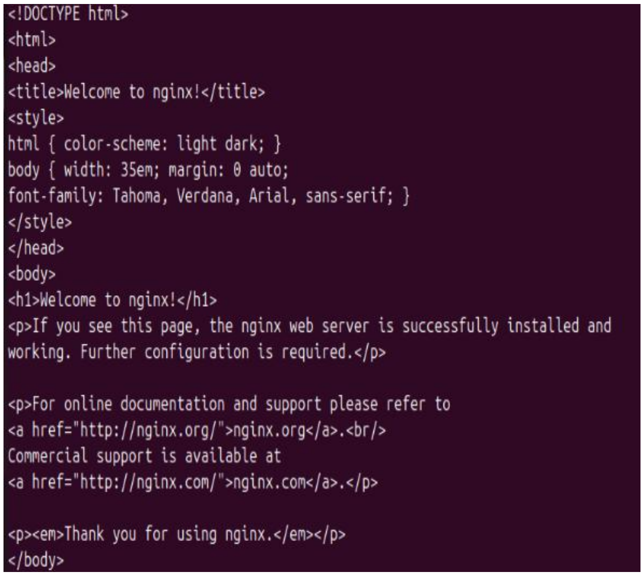


```
sabry@sabry-vm:~/ingress-app$ kubectl apply -f backend-service.yaml
service/backend-service created
sabry@sabry-vm:~/ingress-app$ kubectl get services
NAME              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
admin             ClusterIP   10.96.208.49    <none>        80/TCP    38h
api               ClusterIP   10.96.156.11    <none>        80/TCP    38h
backend-service   ClusterIP   None            <none>        80/TCP    8s
kubernetes        ClusterIP   10.96.0.1       <none>        443/TCP   40h
www               ClusterIP   10.96.83.142    <none>        80/TCP    38h
```

8- try to curl the backend-service from the test pod. What is the response?

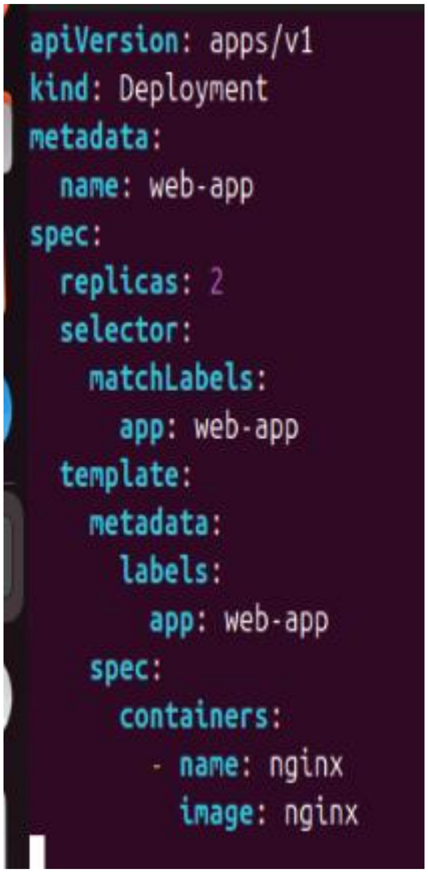


```
sabry@sabry-vm:~/ingress-app$ kubectl exec -it test-pod -- /bin/sh
/ # apk add --no-cache curl
```

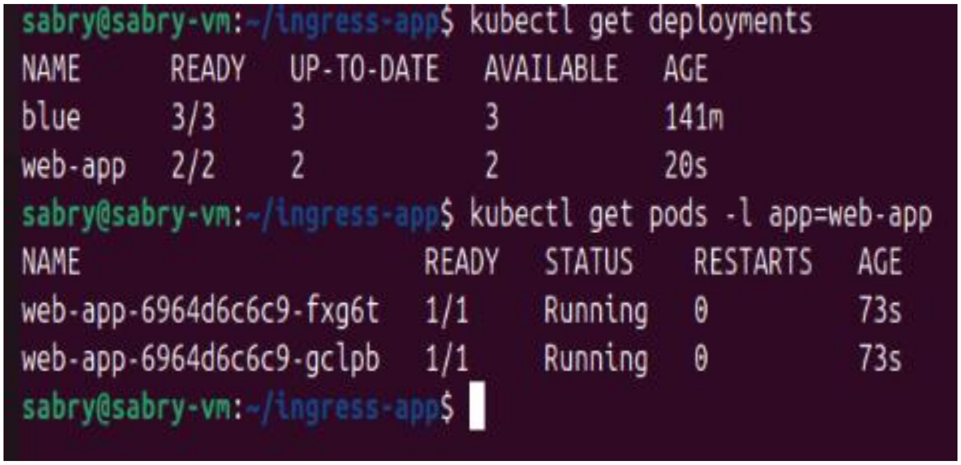


```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
```

9- Create a deployment named web-app using the image nginx with 2 Replicas

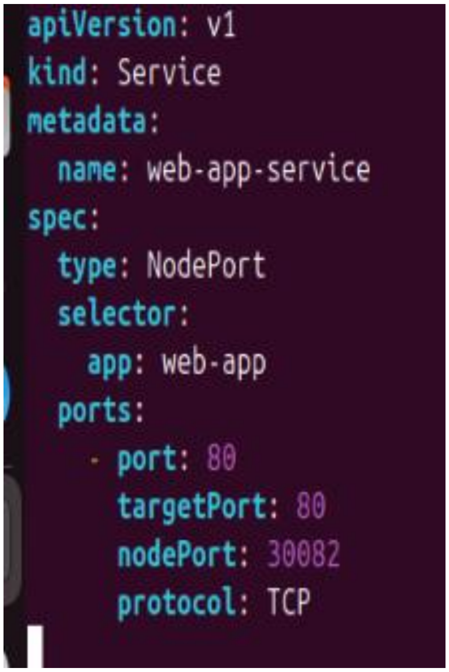


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web-app
  template:
    metadata:
      labels:
        app: web-app
    spec:
      containers:
        - name: nginx
          image: nginx
```

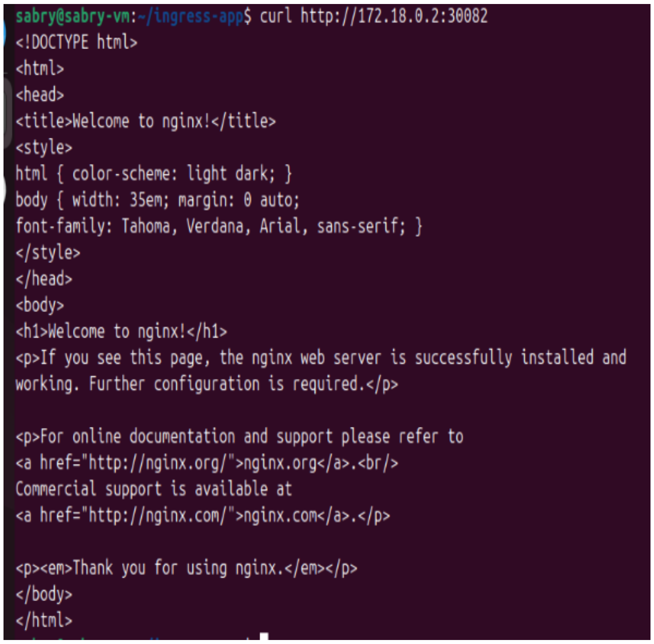


```
sabry@sabry-vm:~/ingress-app$ kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
blue      3/3     3            3           141m
web-app   2/2     2            2           20s
sabry@sabry-vm:~/ingress-app$ kubectl get pods -l app=web-app
NAME                       READY   STATUS    RESTARTS   AGE
web-app-6964d6c6c9-fxg6t   1/1     Running   0          73s
web-app-6964d6c6c9-gclpb   1/1     Running   0          73s
sabry@sabry-vm:~/ingress-app$ 
```

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

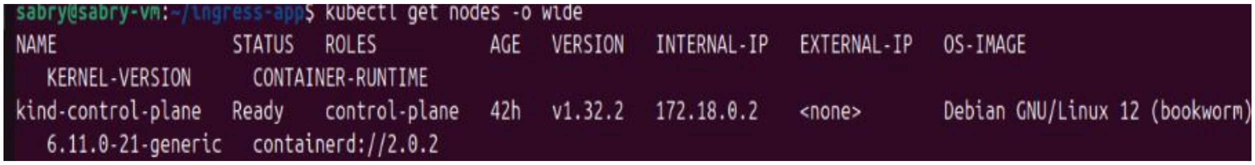


```
apiVersion: v1
kind: Service
metadata:
  name: web-app-service
spec:
  type: NodePort
  selector:
    app: web-app
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30082
      protocol: TCP
```

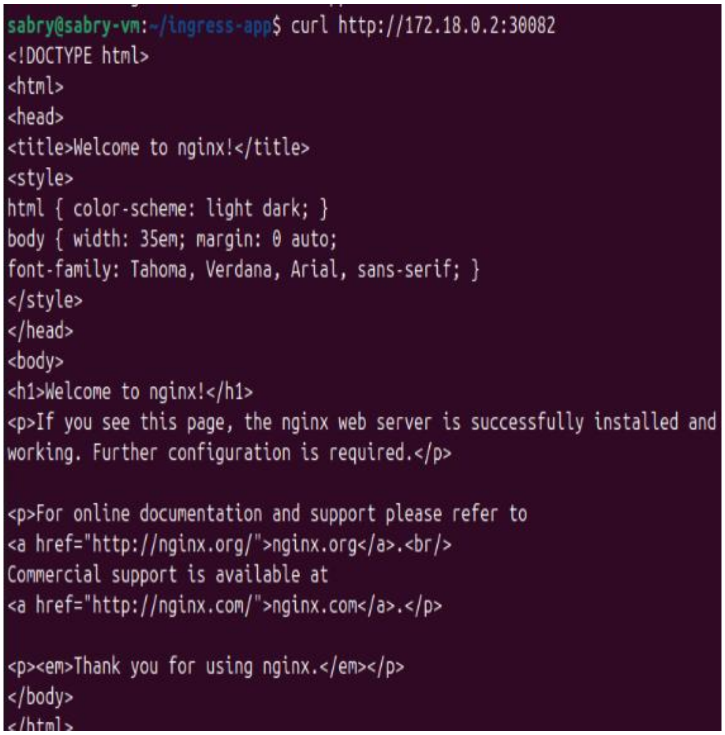


```
sabry@sabry-vm:~/ingress-app$ curl http://172.18.0.2:30082
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

11- access the web app from the node

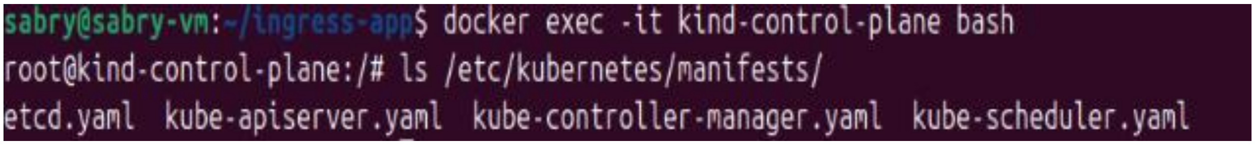


```
sabry@sabry-vm:~/ingress-app$ kubectl get nodes -o wide
NAME                 STATUS   ROLES           AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE
   KERNEL-VERSION     CONTAINER-RUNTIME
kind-control-plane   Ready    control-plane   42h   v1.32.2   172.18.0.2    <none>        Debian GNU/Linux 12 (bookworm)
   6.11.0-21-generic   containerd://2.0.2
```



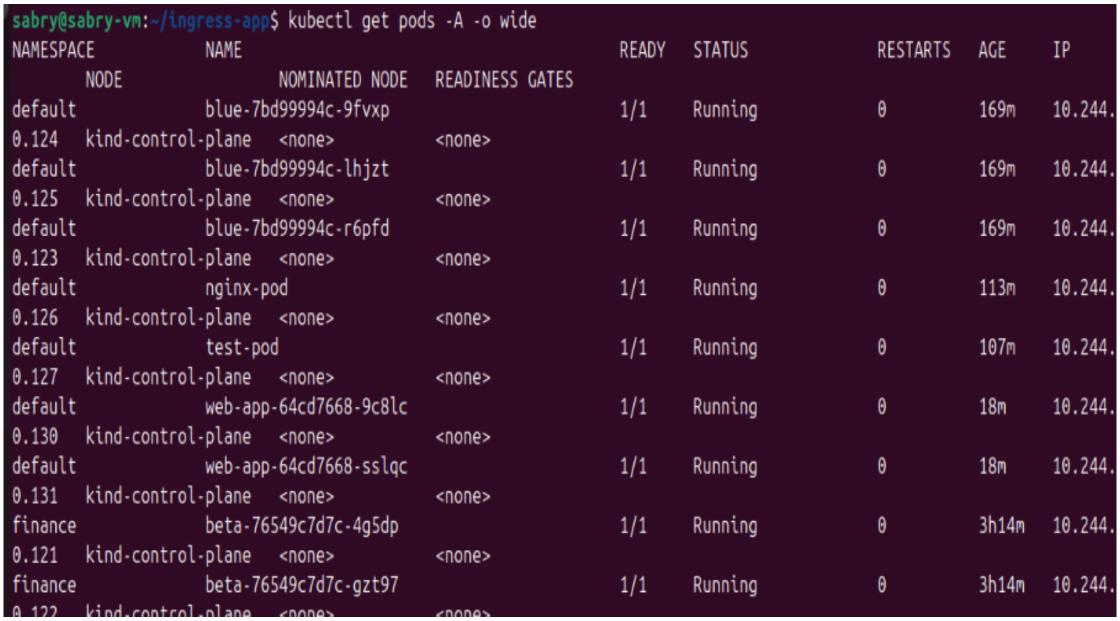
```
sabry@sabry-vm:~/ingress-app$ curl http://172.18.0.2:30082
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

12- How many static pods exist in this cluster in all namespaces?



```
sabry@sabry-vm:~/ingress-app$ docker exec -it kind-control-plane bash
root@kind-control-plane:/# ls /etc/kubernetes/manifests/
etcd.yaml  kube-apiserver.yaml  kube-controller-manager.yaml  kube-scheduler.yaml
```

13-On which nodes are the static pods created currently?



```
sabry@sabry-vm:~/ingress-app$ kubectl get pods -A -o wide
NAMESPACE          NAME                                      READY   STATUS             RESTARTS    AGE     IP
        NODE               NOMINATED NODE   READINESS GATES
default            blue-7bd99994c-9fvxp                      1/1     Running            0           169m    10.244.
0.124   kind-control-plane   <none>           <none>
default            blue-7bd99994c-lhjzt                      1/1     Running            0           169m    10.244.
0.125   kind-control-plane   <none>           <none>
default            blue-7bd99994c-r6pfd                      1/1     Running            0           169m    10.244.
0.123   kind-control-plane   <none>           <none>
default            nginx-pod                                 1/1     Running            0           113m    10.244.
0.126   kind-control-plane   <none>           <none>
default            test-pod                                  1/1     Running            0           107m    10.244.
0.127   kind-control-plane   <none>           <none>
default            web-app-64cd7668-9c8lc                    1/1     Running            0           18m     10.244.
0.130   kind-control-plane   <none>           <none>
default            web-app-64cd7668-sslqc                    1/1     Running            0           18m     10.244.
0.131   kind-control-plane   <none>           <none>
finance            beta-76549c7d7c-4g5dp                     1/1     Running            0           3h14m   10.244.
0.121   kind-control-plane   <none>           <none>
finance            beta-76549c7d7c-gzt97                     1/1     Running            0           3h14m   10.244.
0.122   kind-control-plane   <none>           <none>
```