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

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

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

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

Name: elasticsearch
Namespace: kube-system
Image: k8s.gcr.io/fluentd-elasticsearch:1.20

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

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

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

8- try to curl the backend-service from the test pod. What is the response?9- Create a deployment named `web-app` using the image `nginx` with `2` replicas

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

11- access the web app from the node

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

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

```
Initialising Kubernetes... done

controlplane:~$ kubectl get daemonsets --all-namespaces
NAMESPACE     NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kube-system   canal        2         2         2       2            2           kubernetes.io/os=linux   31d
kube-system   kube-proxy   2         2         2       2            2           kubernetes.io/os=linux   31d
controlplane:~$ kubectl get daemonsets -n kube-system
NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
canal        2         2         2       2            2           kubernetes.io/os=linux   31d
kube-proxy   2         2         2       2            2           kubernetes.io/os=linux   31d
controlplane:~$ kubectl describe daemonset kube-proxy -n kube-system | grep Image
    Image:      registry.k8s.io/kube-proxy:v1.32.1
controlplane:~$ vi daemonset.yaml
controlplane:~$ kubectl apply -f daemonset.yaml
daemonset.apps/elasticsearch created
controlplane:~$ vi daemonset.yaml
controlplane:~$ vi nginx-pod.yaml
controlplane:~$ vi nginx-pod.yaml
controlplane:~$ kubectl apply -f nginx-pod.yaml
pod/nginx-pod created
controlplane:~$ vi test-pod.yaml
controlplane:~$ kubectl apply -f test-pod.yaml
pod/test-pod created
controlplane:~$ vi test-pod.yaml
controlplane:~$ vi backend-service.yaml
controlplane:~$ kubectl apply -f backend-service.yaml
service/backend-service created
```

Editor Tab 1 +

```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: elasticsearch
  namespace: kube-system
spec:
  selector:
    matchLabels:
      name: fluentd
  template:
    metadata:
      labels:
        name: fluentd
    spec:
      containers:
      - name: fluentd
        image: k8s.gcr.io/fluentd-elasticsearch:1.20
~
~
~
~
```

Editor Tab 1 +

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

Editor Tab 1 +

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

Editor Tab 1 +

```yaml
apiVersion: v1
kind: Service
metadata:
  name: backend-service
spec:
  selector:
    tier: backend
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
~
~
```

```
controlplane:~$ kubectl exec -it test-pod -- sh
/ # apk add curl
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/community/x86_64/APKINDEX.tar.gz
OK: 45 MiB in 68 packages
/ # 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>
<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>
/ #
```

Editor Tab 1 +

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

```
controlplane:~$ vi web-app.yaml
controlplane:~$ kubectl apply -f web-app.yaml
deployment.apps/web-app created
controlplane:~$ vi web-app-service.yaml
controlplane:~$ kubectl apply -f web-app-service.yaml
service/web-app-service created
controlplane:~$ kubectl get nodes
NAME           STATUS   ROLES           AGE   VERSION
controlplane   Ready    control-plane   31d   v1.32.1
node01         Ready    <none>          31d   v1.32.1
controlplane:~$ kubectl get nodes -o wide
NAME           STATUS   ROLES           AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION     CONTAINER-RUNTIME
controlplane   Ready    control-plane   31d   v1.32.1   172.30.1.2    <none>        Ubuntu 24.04.1 LTS   6.8.0-51-generic   containerd://1.7.24
node01         Ready    <none>          31d   v1.32.1   172.30.2.2    <none>        Ubuntu 24.04.1 LTS   6.8.0-51-generic   containerd://1.7.24
controlplane:~$ curl http://172.30.1.2:30082
<!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>
```

Editor Tab 1 +

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

```
</html>
controlplane:~$ vi web-app-service.yaml
controlplane:~$ ls /etc/kubernetes/manifests/
etcd.yaml  kube-apiserver.yaml  kube-controller-manager.yaml  kube-scheduler.yaml
controlplane:~$ kubectl get pods -A -o wide | grep static
```