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

```
root@manar-VirtualBox:/home/manar/linux/kuber/lab2# 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   13d
kube-system   kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   13d
root@manar-VirtualBox:/home/manar/linux/kuber/lab2#
```

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

```
root@manar-VirtualBox:/home/manar/linux/kuber/lab2# 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   13d
kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   13d
root@manar-VirtualBox:/home/manar/linux/kuber/lab2#
```

```
              apiVersion: v1
              fieldPath: spec.nodeName
        image: registry.k8s.io/kube-proxy:v1.32.3
        imagePullPolicy: IfNotPresent
```

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

```
          fieldRef:
            apiVersion: v1
            fieldPath: spec.nodeName
      image: registry.k8s.io/kube-proxy:v1.32.3
      imagePullPolicy: IfNotPresent
      name: kube-proxy
```

```
root@manar-VirtualBox:/home/manar/linux/kuber/lab2# kubectl get daemonset kube-proxy -n kube-system -o yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
```

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

Name: elasticsearch

Namespace: kube-system

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

```
io.k8s.api.apps.v1.DaemonSet (v1@daemonset.json)
# - Deploy a DaemonSet for FluentD Logging. Use the given
# specifications.
# Name: elasticsearch
# Namespace: kube-system
# Image: k8s.gcr.io/fluentd-elasticsearch:1.20
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: elasticsearch
  namespace: kube-system
spec:
  selector:
    matchLabels:
      name: elasticsearch
  template:
    metadata:
      labels:
        name: elasticsearch
    spec:
      containers:
      - name: fluentd
        image: k8s.gcr.io/fluentd-elasticsearch:1.20
```

```
root@manar-VirtualBox:/home/manar/linux/kuber/lab3# kubectl apply -f log.yml
daemonset.apps/elasticsearch created
root@manar-VirtualBox:/home/manar/linux/kuber/lab3#
```

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

```yaml
# 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:    One or more containers do not have resources - this ca
  - name: nginx
    image: nginx:alpine
```

```
root@manar-VirtualBox:/home/manar/linux/kuber/lab3# kubectl apply -f deploy5.yml
pod/nginx-pod created
root@manar-VirtualBox:/home/manar/linux/kuber/lab3#
```

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

```yaml
# 6- Deploy a test pod using the nginx:alpine image.
apiVersion: v1
kind: Pod
metadata:
  name: test
spec:
  containers:    One or more containers do not have resourc
  - name: nginx
    image: nginx:alpine
```

```
root@manar-VirtualBox:/home/manar/linux/kuber/lab3# kubectl apply -f deploy6.yml
pod/test created
root@manar-VirtualBox:/home/manar/linux/kuber/lab3#
```

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

```
io.k8s.api.core.v1.Service (v1@service.json)
# 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
  # to expose app on cluster so we need to connect cluster with service (by label )
spec:
  selector:
    tier: backend
  ports:
  - port: 80 #port container
    targetPort: 80 #service
```

```
root@manar-VirtualBox:/home/manar/linux/kuber/lab3# kubectl apply -f deploy7.yml
service/backend-service created
```

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

```
root@manar-VirtualBox:/home/manar/linux/kuber/lab3# kubectl exec -it test -- /bin/sh
/ # curl http://backend-service
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
```

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

```
# 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:    One or more containers do not have resources - this
      - name: nginx
        image: nginx
```

```
root@manar-VirtualBox:/home/manar/linux/kuber/lab3# kubectl apply -f deploy8.yml
deployment.apps/web-app created
root@manar-VirtualBox:/home/manar/linux/kuber/lab3#
```

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

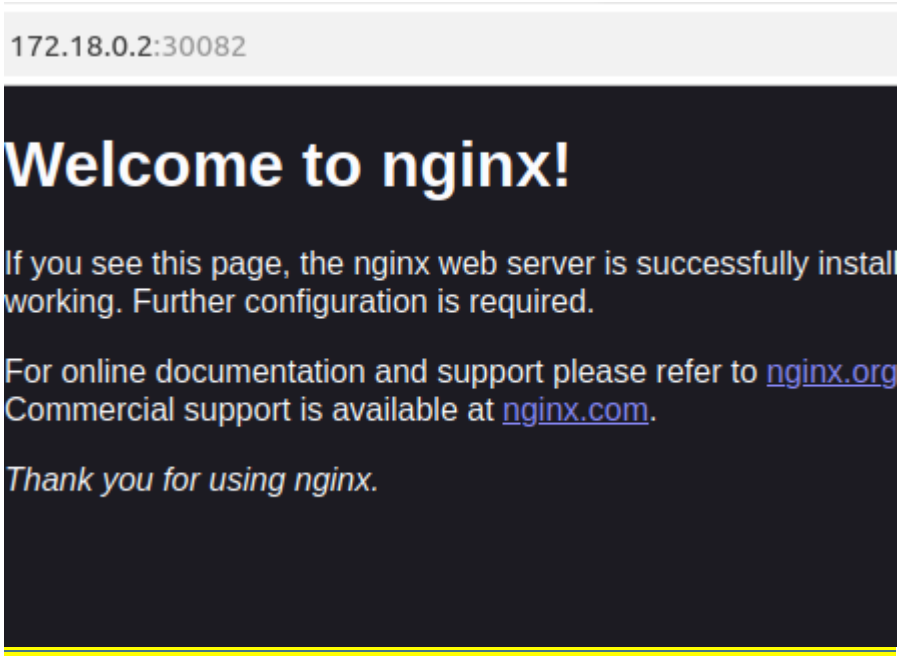
```yaml
# port 80 and nodeport 30082 on the nodes on the cluster
apiVersion: v1
kind: Service
metadata:
  name: web-app-service
spec:
  type: NodePort
  selector:
    app: web-app
  ports:
  - port: 80
    targetPort: 80
    nodePort: 30082
```

```
PROBLEMS 6  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

</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
root@manar-VirtualBox:/home/manar/linux/kuber/lab3# kubectl apply -f deploy8.yml
deployment.apps/web-app created
root@manar-VirtualBox:/home/manar/linux/kuber/lab3# kubectl apply -f nodeport.yml
service/web-app-service created
root@manar-VirtualBox:/home/manar/linux/kuber/lab3#
```

11- access the web app from the node>>kubectl get nodes -o wide , http://172.18.0.2:30082/,http://<IP>:<NodePort>



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

```
root@manar-VirtualBox:/home/manar/linux/kuber/lab3# kubectl get pods --all-namespaces -o wide
NAMESPACE           NAME                                READY   STATUS             RESTARTS
  NODE                NOMINATED NODE   READINESS GATES
default             blue-7bd99994c-6qz27                1/1     Running            0
  kind-control-plane   <none>           <none>
default             blue-7bd99994c-qkqqp                1/1     Running            0
  kind-control-plane   <none>           <none>
default             blue-7bd99994c-wvdph                1/1     Running            0
  kind-control-plane   <none>           <none>
default             curl-pod                            0/1     ImagePullBackOff   0
  kind-control-plane   <none>           <none>
default             nginx-pod                           1/1     Running            0
  kind-control-plane   <none>           <none>
default             redis                               0/1     ImagePullBackOff   0
  kind-control-plane   <none>           <none>
default             test                                1/1     Running            0
  kind-control-plane   <none>           <none>
default             web-app-6964d6c6c9-f7ljn            1/1     Running            0
  kind-control-plane   <none>           <none>
default             web-app-6964d6c6c9-kfd4b            1/1     Running            0
  kind-control-plane   <none>           <none>
default             webapp                              1/2     ErrImagePull       0
  kind-control-plane   <none>           <none>
finance             beta-76549c7d7c-7br7t               1/1     Running            0
```

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

```
root@manar-VirtualBox:/home/manar/linux/kuber/lab3# kubectl get pods --all-namespaces -o wide
NAMESPACE           NAME                                READY   STATUS             RESTARTS
  NODE                NOMINATED NODE   READINESS GATES
default             blue-7bd99994c-6qz27                1/1     Running            0
  kind-control-plane   <none>           <none>
default             blue-7bd99994c-qkqqp                1/1     Running            0
  kind-control-plane   <none>           <none>
default             blue-7bd99994c-wvdph                1/1     Running            0
  kind-control-plane   <none>           <none>
default             curl-pod                            0/1     ImagePullBackOff   0
  kind-control-plane   <none>           <none>
default             nginx-pod                           1/1     Running            0
  kind-control-plane   <none>           <none>
default             redis                               0/1     ImagePullBackOff   0
  kind-control-plane   <none>           <none>
default             test                                1/1     Running            0
  kind-control-plane   <none>           <none>
default             web-app-6964d6c6c9-f7ljn            1/1     Running            0
  kind-control-plane   <none>           <none>
default             web-app-6964d6c6c9-kfd4b            1/1     Running            0
  kind-control-plane   <none>           <none>
default             webapp                              1/2     ErrImagePull       0
  kind-control-plane   <none>           <none>
finance             beta-76549c7d7c-7br7t               1/1     Running            0
```