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

```
reem@reem-host:/$ 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   2d4h
reem@reem-host:/$
```

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

```
reem@reem-host:/$ kubectl get daemonsets -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   2d4h
reem@reem-host:/$
```

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

```
reem@reem-host:/$ kubectl describe daemonset kube-proxy -n kube-system
Name:           kube-proxy
Selector:       k8s-app=kube-proxy
Node-Selector:  kubernetes.io/os=linux
Labels:         k8s-app=kube-proxy
Annotations:    deprecated.daemonset.template.generation: 1
Desired Number of Nodes Scheduled: 1
Current Number of Nodes Scheduled: 1
Number of Nodes Scheduled with Up-to-date Pods: 1
Number of Nodes Scheduled with Available Pods: 1
Number of Nodes Misscheduled: 0
Pods Status:  1 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:           k8s-app=kube-proxy
  Service Account:  kube-proxy
  Containers:
   kube-proxy:
    Image:      registry.k8s.io/kube-proxy:v1.32.0
```

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

Name: elasticsearch

Namespace: kube-system

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

```
reem@reem-host:~$ sudo nano fluentd-daemonset.yaml
[sudo] password for reem:
reem@reem-host:~$ kubectl apply -f fluentd-daemonset.yaml
daemonset.apps/elasticsearch created
```

```
reem@reem-host:~$ sudo nano fluentd-daemonset.yaml
reem@reem-host:~$ kubectl get daemonsets --all-namespaces
NAMESPACE     NAME            DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kube-system   elasticsearch   1         1         0       1            0           <none>                   6m12
kube-system   kube-proxy      1         1         1       1            1           kubernetes.io/os=linux   2d4h
```

```
 GNU nano 7.2                                  fluentd-daemonset.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: elasticsearch
  namespace: kube-system
  labels:
    app: fluentd
spec:
  selector:
    matchLabels:
      app: fluentd
  template:
    metadata:
      labels:
        app: fluentd
    spec:
      containers:
      - name: fluentd
        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`.

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

```
reem@reem-host:~$ nano nginx-pod.yml
reem@reem-host:~$ kubectl apply -f nginx-pod.yml
pod/nginx-pod created
reem@reem-host:~$ kubectl get pod nginx-pod --show-labels
NAME        READY   STATUS    RESTARTS   AGE   LABELS
nginx-pod   1/1     Running   0          46s   tier=backend
```

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
```

```
reem@reem-host:~$ nano test-pod.yml
reem@reem-host:~$ kubectl apply -f test-pod.yml
pod/test-pod created
reem@reem-host:~$ kubectl get pod test
Error from server (NotFound): pods "test" not found
reem@reem-host:~$ kubectl get pod test-pod
NAME       READY   STATUS    RESTARTS   AGE
test-pod   1/1     Running   0          29s
reem@reem-host:~$
```

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

```yaml
apiVersion: v1
kind: Service
metadata:
  name: backend-service
spec:
  type: ClusterIP
  ports:
   - targetPort: 80
     port: 80
```

```
reem@reem-host:~$ nano backend-service.yml
reem@reem-host:~$ kubectl apply -f backend-service.yml
service/backend-service created
reem@reem-host:~$ kubectl get svc
NAME              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
backend-service   ClusterIP   10.96.127.215   <none>        80/TCP    22s
kubernetes        ClusterIP   10.96.0.1       <none>        443/TCP   2d4h
reem@reem-host:~$
```

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

```
reem@reem-host:~$ kubectl get pods -o wide
NAME                   READY   STATUS    RESTARTS   AGE   IP            NODE
nginx-pod              1/1     Running   0          50m   10.244.0.60   minikube
test-pod               1/1     Running   0          45m   10.244.0.61   minikube
web-app-64cd7668-g5xpp 1/1     Running   0          19m   10.244.0.62   minikube
web-app-64cd7668-j2v2c 1/1     Running   0          19m   10.244.0.63   minikube
reem@reem-host:~$ minikube ssh
docker@minikube:~$ curl 10.244.0.61
<!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>
```

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

```
reem@reem-host:~$ nano web-app.yml
reem@reem-host:~$ kubectl apply -f web-app.yml
deployment.apps/web-app created
reem@reem-host:~$ kubectl get deployment web-app
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
web-app   0/2     2            0           30s
```

```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
          ports:
          - containerPort: 80
```

```
reem@reem-host:~$ kubectl get deployment web-app
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
web-app   2/2     2            2           3m55s
reem@reem-host:~$ kubectl get pods -l app=web-app
NAME                       READY   STATUS    RESTARTS   AGE
web-app-64cd7668-g5xpp     1/1     Running   0          4m6s
web-app-64cd7668-j2v2c     1/1     Running   0          4m6s
```

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

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

```
reem@reem-host:~$ nano web-server.yml
reem@reem-host:~$ kubectl apply -f web-server.yml
service/webapp-service created
reem@reem-host:~$ kubectl get svc
NAME              TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
backend-service   ClusterIP   10.96.127.215    <none>        80/TCP         59m
kubernetes        ClusterIP   10.96.0.1        <none>        443/TCP        2d5h
webapp-service    NodePort    10.102.181.184   <none>        80:30082/TCP   35s
reem@reem-host:~$
```

11- access the web app from the node

12-

```
reem@reem-host:~$ kubectl get pods -o wide
NAME                     READY   STATUS    RESTARTS   AGE   IP            NODE
nginx-pod                1/1     Running   0          86m   10.244.0.60   minikube
test-pod                 1/1     Running   0          81m   10.244.0.61   minikube
web-app-64cd7668-g5xpp   1/1     Running   0          55m   10.244.0.62   minikube
web-app-64cd7668-j2v2c   1/1     Running   0          55m   10.244.0.63   minikube
reem@reem-host:~$ minikube ssh
docker@minikube:~$ curl 10.244.0.62
<!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>
```

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

```
controlplane:~$ cd /etc/kubernetes/manifests/
controlplane:/etc/kubernetes/manifests$ ls
etcd.yaml  kube-apiserver.yaml  kube-controller-manager.yaml  kube-scheduler.yaml
controlplane:/etc/kubernetes/manifests$
```

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

All static pods are created and running on the Master node