

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

```
menna@docker:~$ 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	7d5h

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

```
menna@docker:~$ 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	7d5h

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

```
menna@docker:~$ kubectl get daemonset kube-proxy -n kube-system -o jsonpath='{.spec.template.spec.containers[0].image}'
registry.k8s.io/kube-proxy:v1.32.0menna@docker:~$
```

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

Name: elasticsearch

Namespace: kube-system

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

```
menna@docker:~$ vim elasticsearch.yaml
```

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: elasticsearch
  namespace: kube-system
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      containers:
        - name: fluentd-elasticsearch
          image: k8s.gcr.io/fluentd-elasticsearch:1.20
~
~
~
~
~
~
~
"elasticsearch.yaml" 17L, 355B
menna@docker:~$ kubectl apply -f elasticsearch.yaml
daemonset.apps/elasticsearch created
```

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

```
menna@docker:~$ vim nginx-pod.yaml
```

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

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
"nginx-pod.yaml" 10L, 141B
```

```
menna@docker:~$ kubectl apply -f nginx-pod.yaml
pod/nginx-pod created
```

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

```
menna@docker:~$ vim test-pod.yaml
```

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

```
menna@docker:~$ kubectl apply -f test-pod.yaml
pod/test-pod created
```

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

```
menna@docker:~$ vim backend-service.yaml
```

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

```
menna@docker:~$ kubectl apply -f backend-service.yaml
service/backend-service created
```

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

```
menna@docker:~$ kubectl exec -it test-pod -- curl http://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>
```

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

```
menna@docker:~$ vim web-app-deployment.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
~
```

```
menna@docker:~$ kubectl apply -f web-app-deployment.yaml
deployment.apps/web-app created
```

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

```
menna@docker:~$ vim web-app-service.yaml
```

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

```
menna@docker:~$ kubectl apply -f web-app-service.yaml
service/web-app-service created
```

11- access the web app from the node

```
menna@docker:~$ kubectl get nodes -o wide
NAME          STATUS    ROLES          AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION   CONTAINER-RUNTIME
minikube      Ready     control-plane   7d6h   v1.32.0   192.168.49.2   <none>         Ubuntu 22.04.5 LTS   6.11.0-17-generic docker://27.4.1

menna@docker:~$ curl http://192.168.49.2:30082
<!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?

```
menna@docker:~$ kubectl get pods --all-namespaces -o jsonpath='{.items[?(@.metadata.ownerReferences[*].kind=="Node")].metadata.name}' | wc -w
4
```

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

```
menna@docker:~$ kubectl get pods --all-namespaces -o jsonpath='{.items[?(@.metadata.ownerReferences[*].kind=="Node")].spec.nodeName}' | sort | uniq
minikube minikube minikube minikube
```