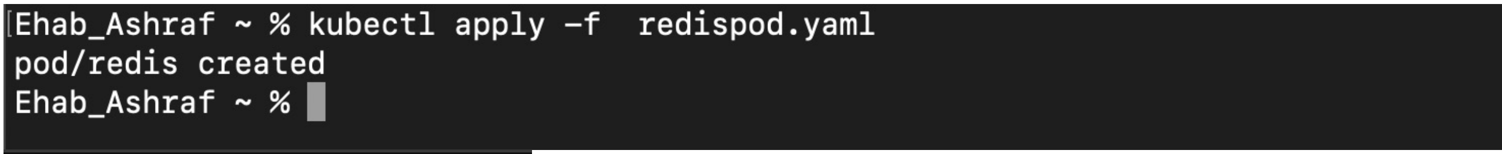
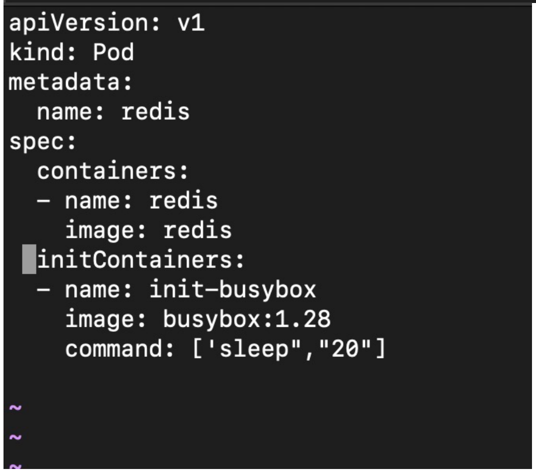
1- Create a pod red with redis image and use an init container that uses the busybox image and sleeps for 20 seconds



```
[Ehab_Ashraf ~ % kubectl apply -f  redispod.yaml
pod/redis created
Ehab_Ashraf ~ %
```



```
apiVersion: v1
kind: Pod
metadata:
  name: redis
spec:
  containers:
  - name: redis
    image: redis
  initContainers:
  - name: init-busybox
    image: busybox:1.28
    command: ['sleep","20"]

~
~
~
```

2- Create a pod named print-envars-greeting.

1. Configure spec as, the container name should be print-env-container and use bash image.

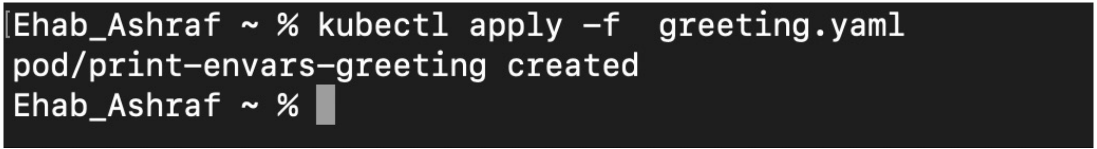
2. Create three environment variables:

   a. GREETING and its value should be “Welcome to”

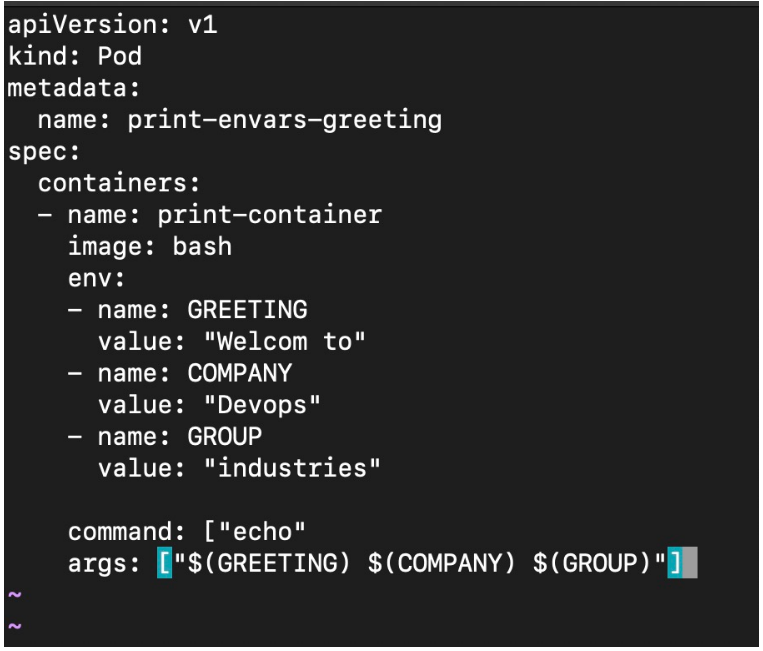
   b. COMPANY and its value should be “DevOps”

   c. GROUP and its value should be “Industries”

3. Use command to echo ["$(GREETING) $(COMPANY) $(GROUP)"] message.

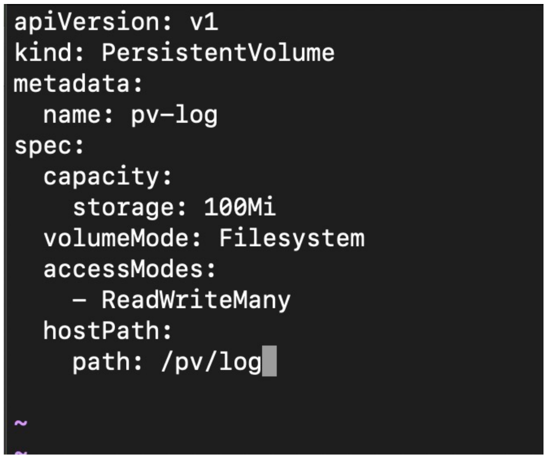


```
[Ehab_Ashraf ~ % kubectl apply -f  greeting.yaml
pod/print-envars-greeting created
Ehab_Ashraf ~ %
```

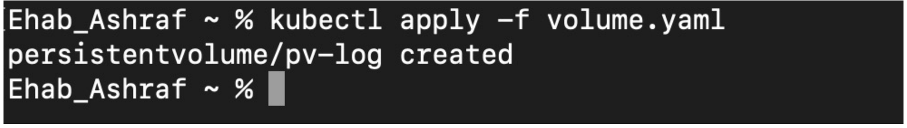


```
apiVersion: v1
kind: Pod
metadata:
  name: print-envars-greeting
spec:
  containers:
  - name: print-container
    image: bash
    env:
    - name: GREETING
      value: "Welcom to"
    - name: COMPANY
      value: "Devops"
    - name: GROUP
      value: "industries"

    command: ["echo"
    args: ["$(GREETING) $(COMPANY) $(GROUP)"]
~
~
```

3- Create a Persistent Volume with the given specification.
Volume Name: pv-log
Storage: 100Mi
Access Modes: ReadWriteMany
Host Path: /pv/log

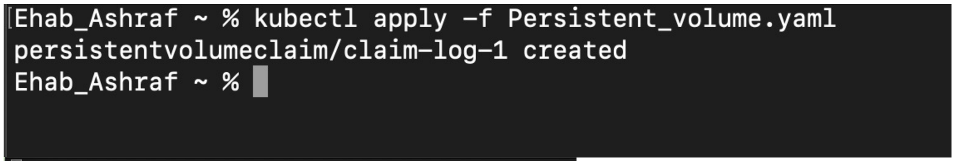


```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-log
spec:
  capacity:
    storage: 100Mi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteMany
  hostPath:
    path: /pv/log

~
~
```

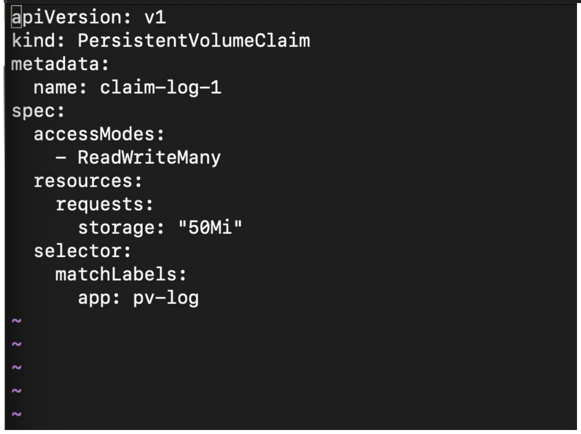


```
Ehab_Ashraf ~ % kubectl apply -f volume.yaml
persistentvolume/pv-log created
Ehab_Ashraf ~ %
```

4- Create a Persistent Volume Claim with the given specification.
Volume Name: claim-log-1
Storage Request: 50Mi
Access Modes: ReadWriteMany

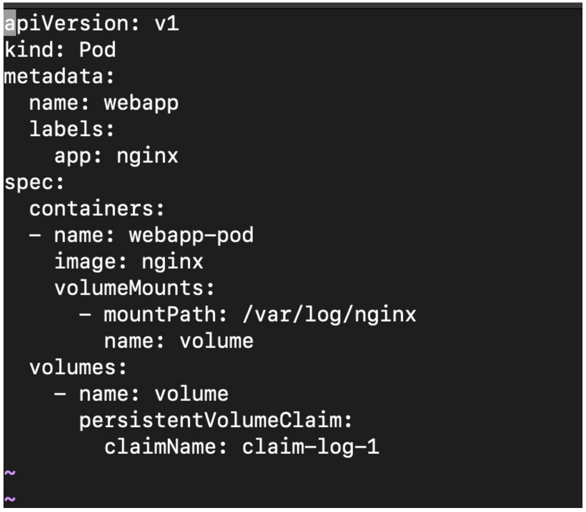


```
Ehab_Ashraf ~ % kubectl apply -f Persistent_volume.yaml
persistentvolumeclaim/claim-log-1 created
Ehab_Ashraf ~ %
```

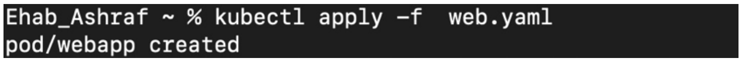


```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: claim-log-1
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: "50Mi"
  selector:
    matchLabels:
      app: pv-log
~
~
~
~
~
```

5- Create a webapp pod to use the persistent volume claim as its storage.
Name: webapp
Image Name: nginx
Volume: PersistentVolumeClaim=claim-log-1
Volume Mount: /var/log/nginx

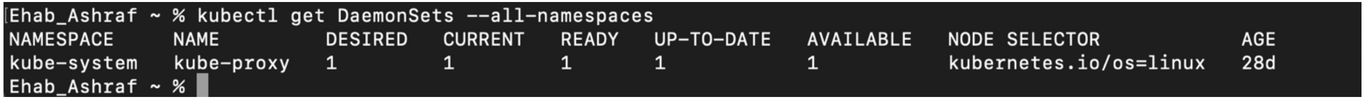


```
apiVersion: v1
kind: Pod
metadata:
  name: webapp
  labels:
    app: nginx
spec:
  containers:
  - name: webapp-pod
    image: nginx
    volumeMounts:
      - mountPath: /var/log/nginx
        name: volume
  volumes:
    - name: volume
      persistentVolumeClaim:
        claimName: claim-log-1
~
~
```



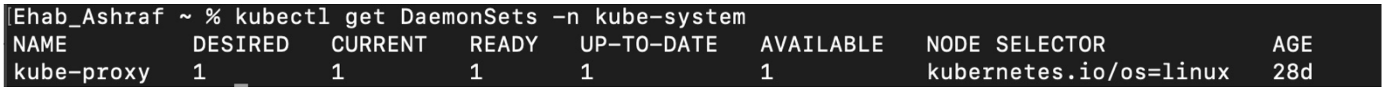
```
Ehab_Ashraf ~ % kubectl apply -f  web.yaml
pod/webapp created
```

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



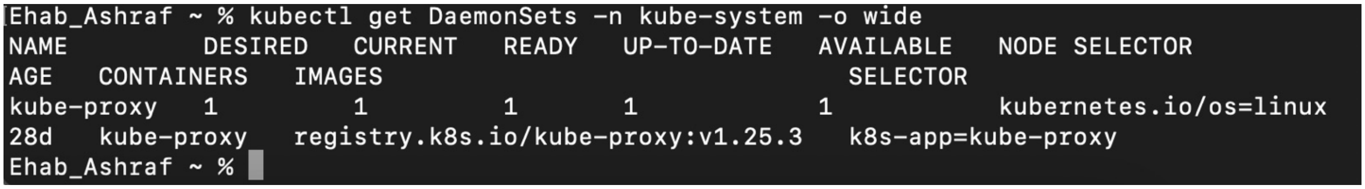
```
[Ehab_Ashraf ~ % kubectl get DaemonSets --all-namespaces
NAMESPACE     NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kube-system   kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   28d
Ehab_Ashraf ~ %
```

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



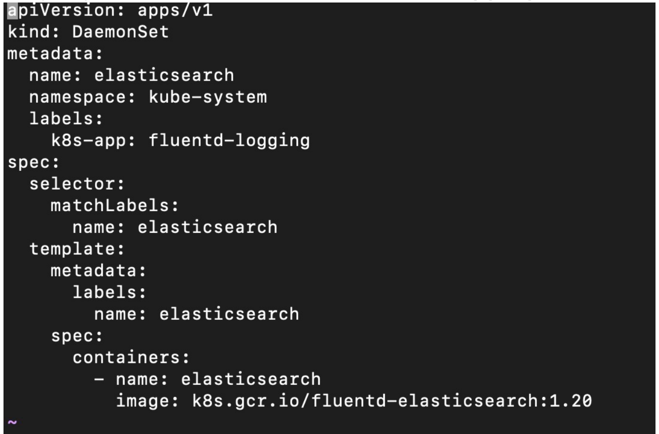
```
[Ehab_Ashraf ~ % kubectl get DaemonSets -n kube-system
NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   28d
```

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

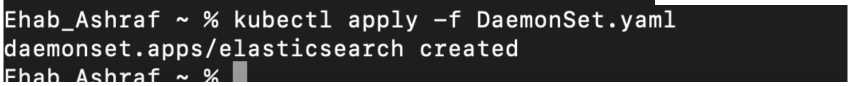


```
Ehab_Ashraf ~ % kubectl get DaemonSets -n kube-system -o wide
NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR
AGE   CONTAINERS   IMAGES                                 SELECTOR
kube-proxy   1         1         1       1            1           kubernetes.io/os=linux
28d   kube-proxy   registry.k8s.io/kube-proxy:v1.25.3   k8s-app=kube-proxy
Ehab_Ashraf ~ %
```

9- Deploy a DaemonSet for FluentD Logging. Use the given

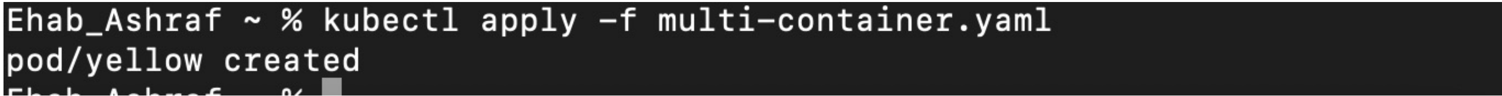


```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: elasticsearch
  namespace: kube-system
  labels:
    k8s-app: fluentd-logging
spec:
  selector:
    matchLabels:
      name: elasticsearch
  template:
    metadata:
      labels:
        name: elasticsearch
    spec:
      containers:
       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

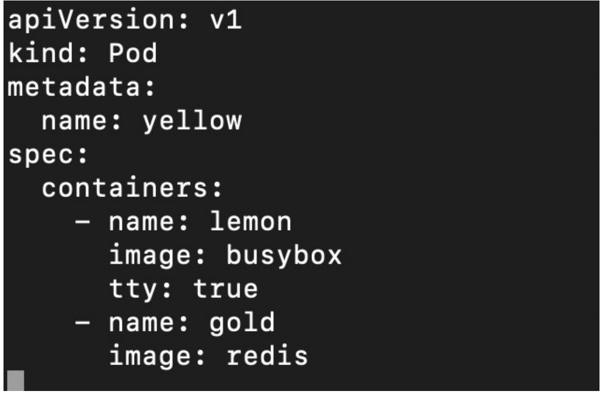


```
Ehab_Ashraf ~ % kubectl apply -f DaemonSet.yaml
daemonset.apps/elasticsearch created
Ehab_Ashraf ~ %
```

10- Create a multi-container pod with 2 containers.



```
Ehab_Ashraf ~ % kubectl apply -f multi-container.yaml
pod/yellow created
Ehab_Ashraf ~ %
```



```
apiVersion: v1
kind: Pod
metadata:
  name: yellow
spec:
  containers:
    - name: lemon
      image: busybox
      tty: true
    - name: gold
      image: redis
```