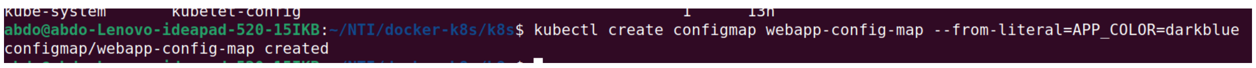
## 1. How many ConfigMaps exist in the environment?

kubectl get configmaps --all-namespaces

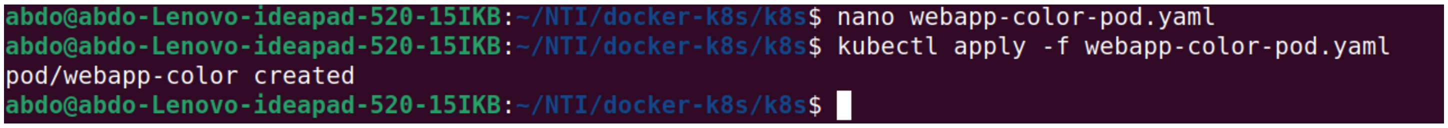
## 2. Create a new ConfigMap

- Name: `webapp-config-map`
- Data: `APP_COLOR=darkblue`



```
kube-system       kubelet-config                                          1      13h
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl create configmap webapp-config-map --from-literal=APP_COLOR=darkblue
configmap/webapp-config-map created
```

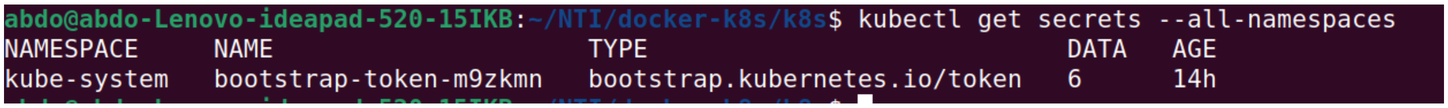
## 3. Create a `webapp-color` pod using nginx image and attach ConfigMap



```
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ nano webapp-color-pod.yaml
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl apply -f webapp-color-pod.yaml
pod/webapp-color created
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ 
```

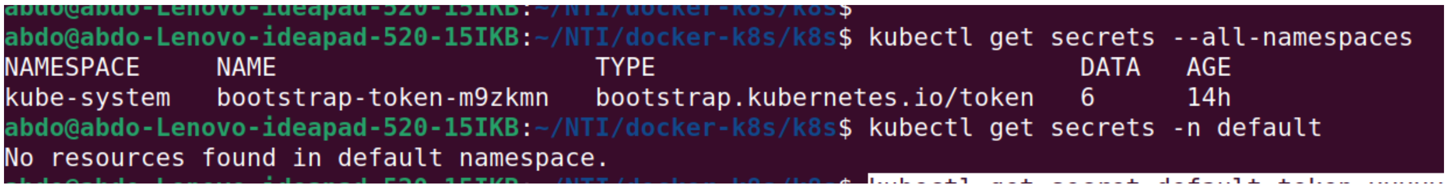
```
apiVersion: v1
kind: Pod
metadata:
  name: webapp-color
spec:
  containers:
  - name: nginx
    image: nginx
    envFrom:
    - configMapRef:
        name: webapp-config-map
```

## 4. How many Secrets exist in the system?



```
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl get secrets --all-namespaces
NAMESPACE     NAME                        TYPE                                  DATA   AGE
kube-system   bootstrap-token-m9zkmn      bootstrap.kubernetes.io/token         6      14h
```

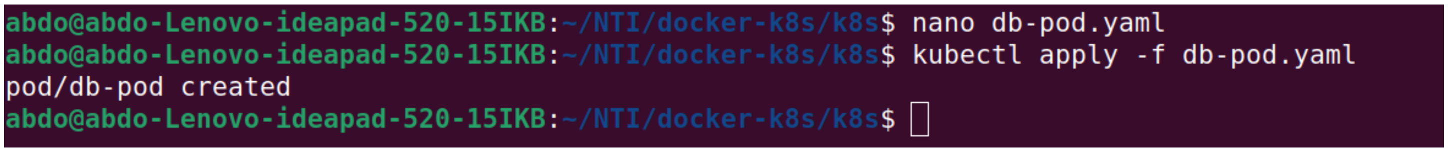
## 5. How many keys are defined inside the `default-token` Secret?



```
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl get secrets --all-namespaces
NAMESPACE     NAME                        TYPE                                  DATA   AGE
kube-system   bootstrap-token-m9zkmn      bootstrap.kubernetes.io/token         6      14h
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl get secrets -n default
No resources found in default namespace.
```

## 6. Create a Pod `db-pod` with image `mysql:5.7`

```
# db-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: db-pod
spec:
  containers:
  - name: mysql
    image: mysql:5.7
```

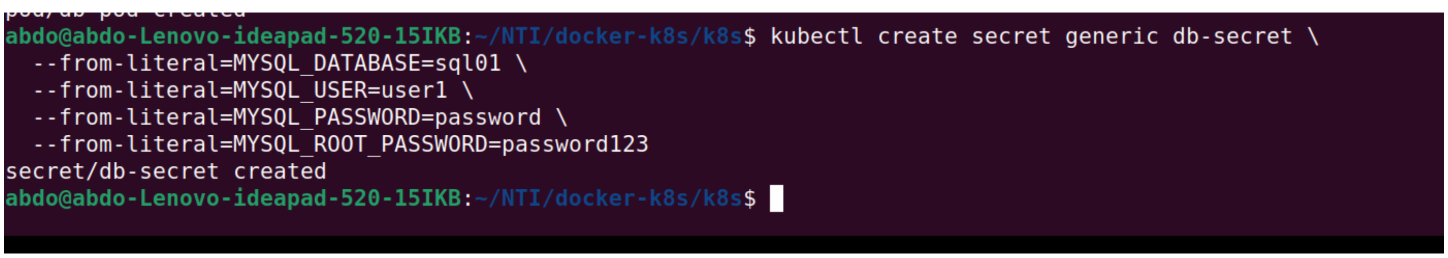


```
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ nano db-pod.yaml
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl apply -f db-pod.yaml
pod/db-pod created
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ 
```

## 7. Why is `db-pod` status not ready?

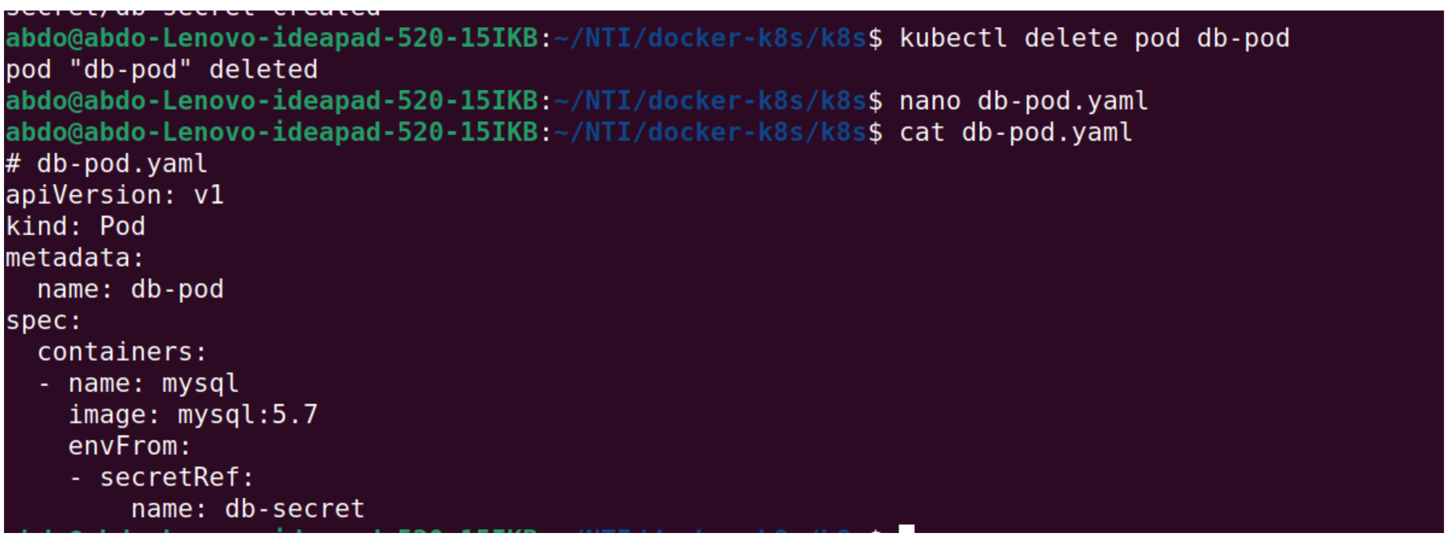
Because MySQL image **needs environment variables** like `MYSQL_ROOT_PASSWORD` to be set at startup. Otherwise, it will fail to start correctly.

## 8. Create a Secret named `db-secret` with given data

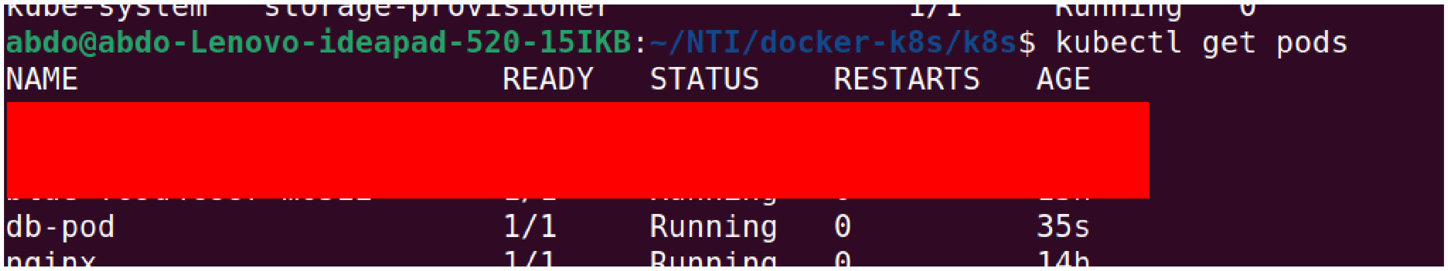


```
pod/db-pod created
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl create secret generic db-secret \
  --from-literal=MYSQL_DATABASE=sql01 \
  --from-literal=MYSQL_USER=user1 \
  --from-literal=MYSQL_PASSWORD=password \
  --from-literal=MYSQL_ROOT_PASSWORD=password123
secret/db-secret created
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ 
```

## 9. Configure `db-pod` to load environment variables from Secret

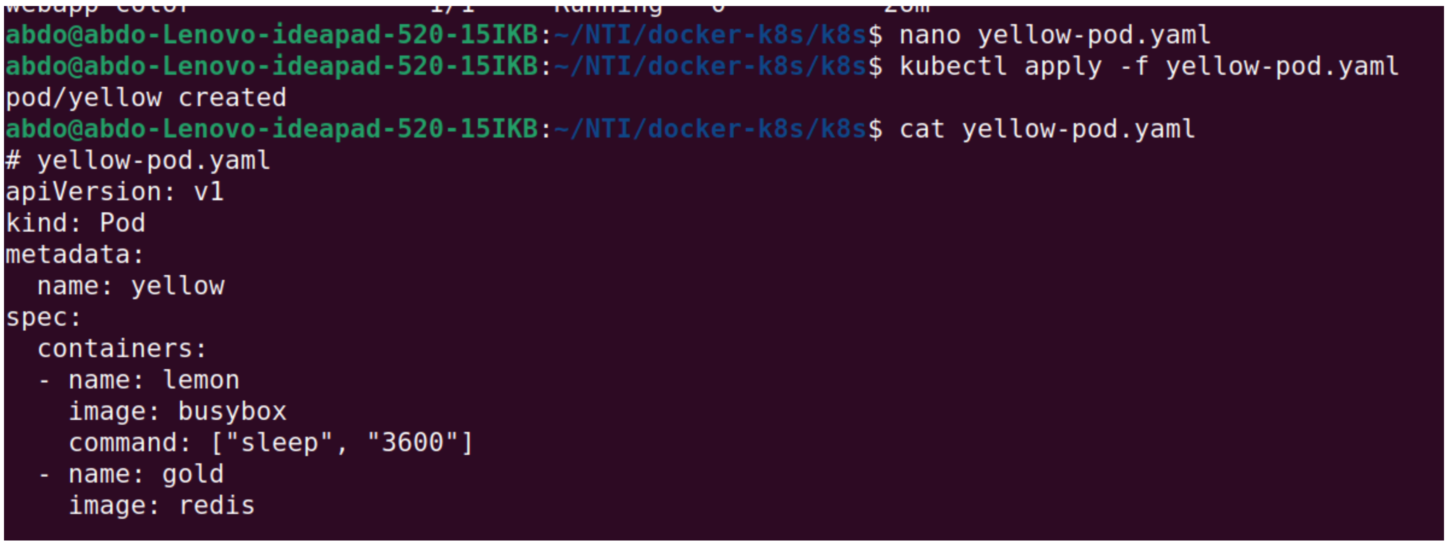


```
secret/db-secret created
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl delete pod db-pod
pod "db-pod" deleted
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ nano db-pod.yaml
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ cat db-pod.yaml
# db-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: db-pod
spec:
  containers:
  - name: mysql
    image: mysql:5.7
    envFrom:
    - secretRef:
        name: db-secret
```



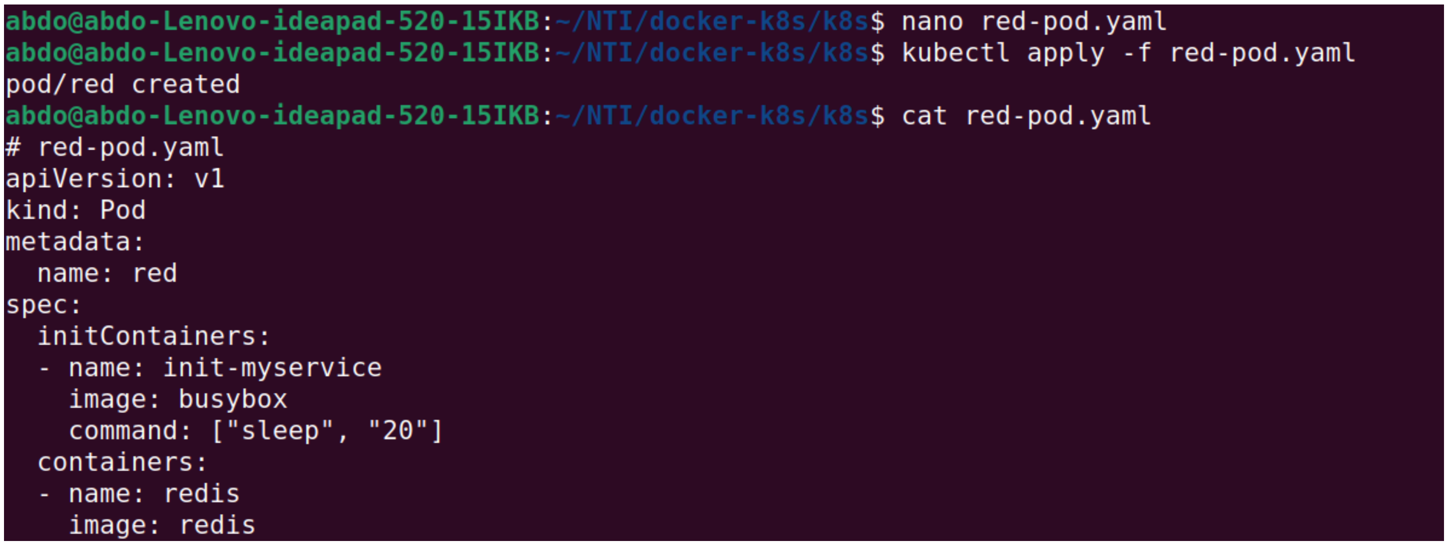
```
kube-system   storage-provisioner              1/1     Running   0
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl get pods
NAME                          READY    STATUS     RESTARTS    AGE
[redacted]
db-pod                        1/1      Running    0           35s
nginx                         1/1      Running    0           14h
```

**10. Create a multi-container pod `yellow`**



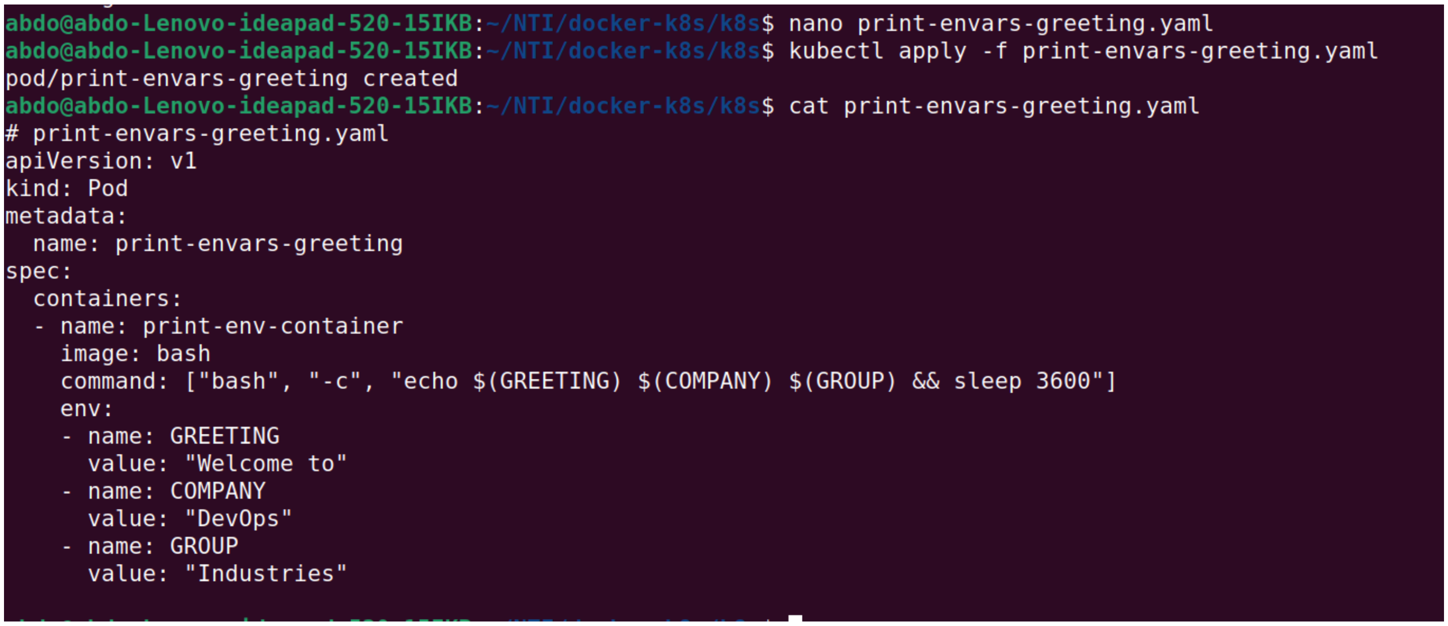
```
webapp-color                 1/1     Running   0          20m
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ nano yellow-pod.yaml
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl apply -f yellow-pod.yaml
pod/yellow created
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ cat yellow-pod.yaml
# yellow-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: yellow
spec:
  containers:
  - name: lemon
    image: busybox
    command: ["sleep", "3600"]
  - name: gold
    image: redis
```

**11- Create a pod red with redis image and use an initContainer that uses the busybox image and sleeps for 20 seconds**

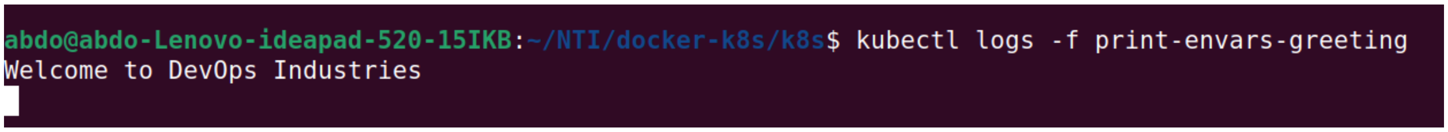


```
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ nano red-pod.yaml
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl apply -f red-pod.yaml
pod/red created
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ cat red-pod.yaml
# red-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: red
spec:
  initContainers:
  - name: init-myservice
    image: busybox
    command: ["sleep", "20"]
  containers:
  - name: redis
    image: redis
```

**12- 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”**
**4. Use command to echo ["$(GREETING) $(COMPANY) $(GROUP)"] message.**
**5. You can check the output using command.**

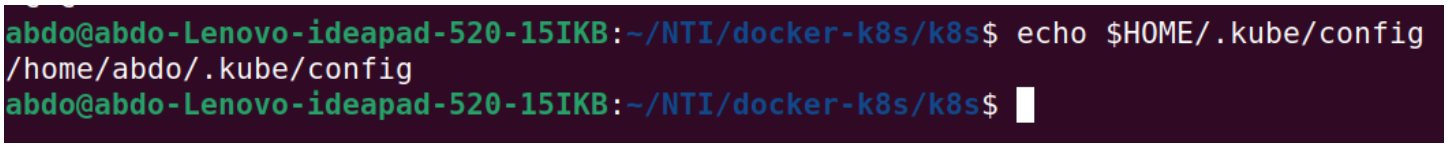


```
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ nano print-envars-greeting.yaml
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl apply -f print-envars-greeting.yaml
pod/print-envars-greeting created
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ cat print-envars-greeting.yaml
# print-envars-greeting.yaml
apiVersion: v1
kind: Pod
metadata:
  name: print-envars-greeting
spec:
  containers:
  - name: print-env-container
    image: bash
    command: ["bash", "-c", "echo $(GREETING) $(COMPANY) $(GROUP) && sleep 3600"]
    env:
    - name: GREETING
      value: "Welcome to"
    - name: COMPANY
      value: "DevOps"
    - name: GROUP
      value: "Industries"
```



```
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl logs -f print-envars-greeting
Welcome to DevOps Industries
```

**13. Where is the default kubeconfig file?**

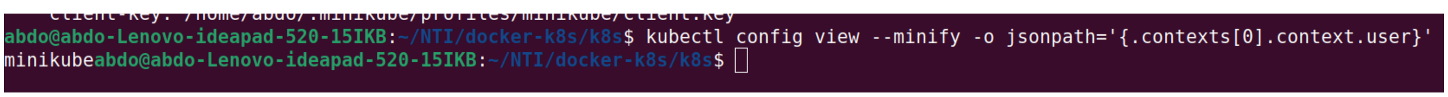


```
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ echo $HOME/.kube/config
/home/abdo/.kube/config
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$
```

**14. How many clusters are defined in the default kubeconfig?**

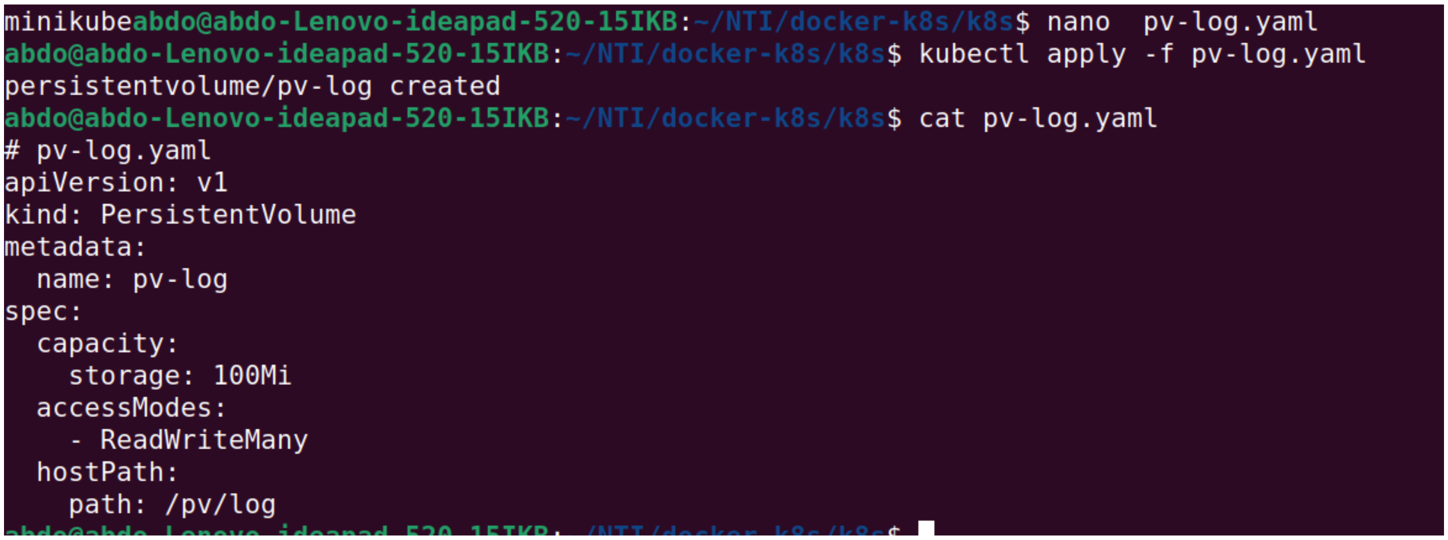
kubectl config view --minify=false

**15. What is the user configured in the current context?**



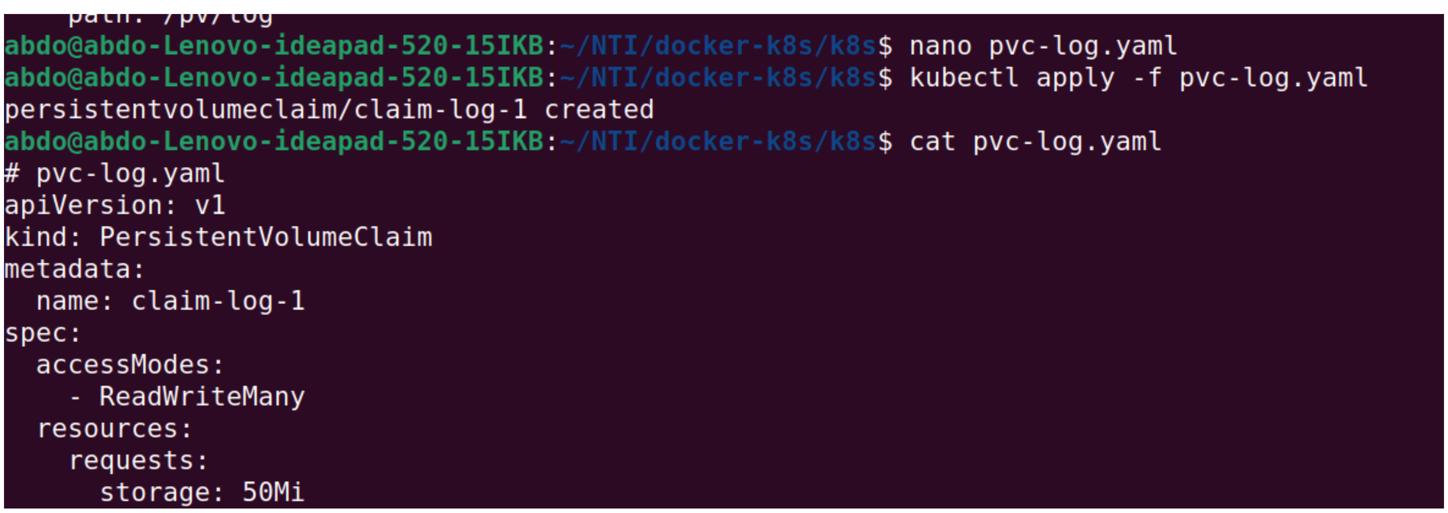
```
    client-key: /home/abdo/.minikube/profiles/minikube/client.key
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl config view --minify -o jsonpath='{.contexts[0].context.user}'
minikubeabdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$
```

## 16. Create a Persistent Volume (PV)



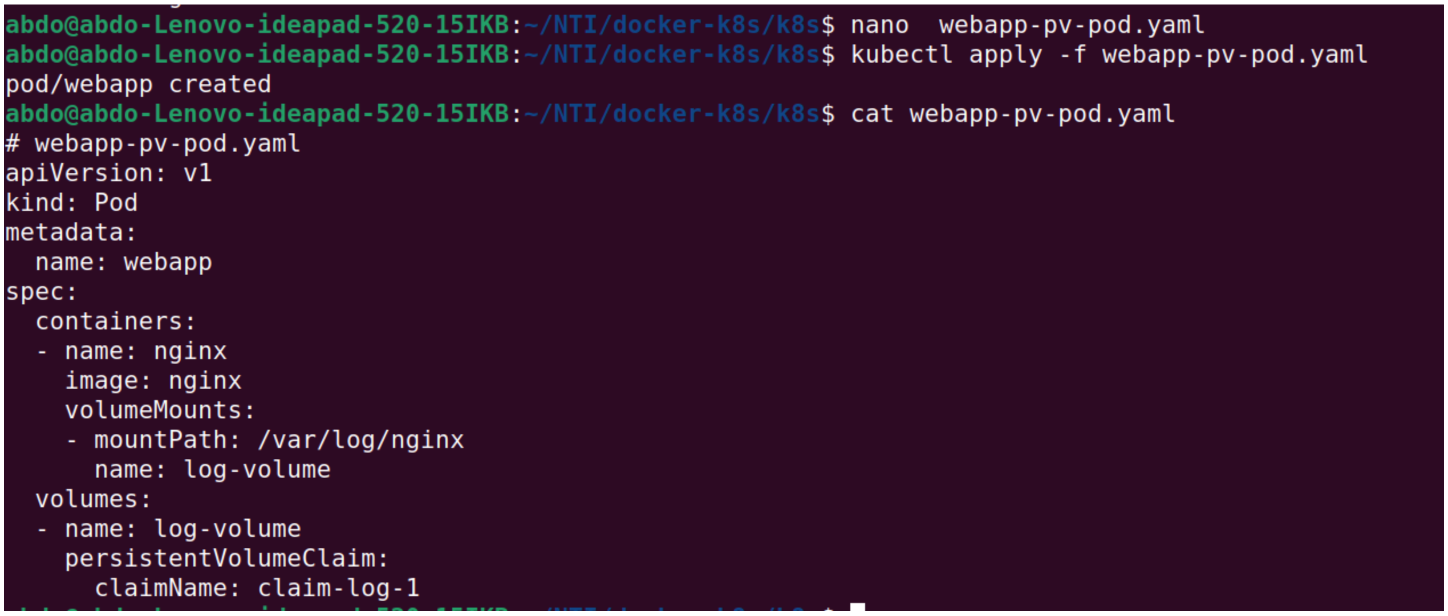
```
minikubeabdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ nano  pv-log.yaml
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl apply -f pv-log.yaml
persistentvolume/pv-log created
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ cat pv-log.yaml
# pv-log.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-log
spec:
  capacity:
    storage: 100Mi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: /pv/log
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$
```

## 17. Create a Persistent Volume Claim (PVC)



```
    path: /pv/log
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ nano pvc-log.yaml
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl apply -f pvc-log.yaml
persistentvolumeclaim/claim-log-1 created
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ cat pvc-log.yaml
# pvc-log.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: claim-log-1
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 50Mi
```

## 18. Create a webapp pod that uses the PVC



```
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ nano  webapp-pv-pod.yaml
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ kubectl apply -f webapp-pv-pod.yaml
pod/webapp created
abdo@abdo-Lenovo-ideapad-520-15IKB:~/NTI/docker-k8s/k8s$ cat webapp-pv-pod.yaml
# webapp-pv-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: webapp
spec:
  containers:
  - name: nginx
    image: nginx
    volumeMounts:
    - mountPath: /var/log/nginx
      name: log-volume
  volumes:
  - name: log-volume
    persistentVolumeClaim:
      claimName: claim-log-1
```