1- How many ConfigMaps exist in the environment?

2- Create a new ConfigMap Use the spec given below.
ConfigName Name: webapp-config-map
Data: APP_COLOR=darkblue

3- Create a webapp-color POD with nginx image and use the created ConfigMap

4- How many Secrets exist on the system?

5- How many secrets are defined in the default-token secret?

6- create a POD called db-pod with the image mysql:5.7 then check the POD status

7- why the db-pod status not ready

8- Create a new secret named db-secret with the data given below.
Secret Name: db-secret
Secret 1: MYSQL_DATABASE=sql01
Secret 2: MYSQL_USER=user1
Secret3: MYSQL_PASSWORD=password
Secret 4: MYSQL_ROOT_PASSWORD=password123

9- Configure db-pod to load environment variables from the newly created secret.
Delete and recreate the pod if required.

10-Create a multi-container pod with 2 containers.
Name: yellow
Container 1 Name: lemon
Container 1 Image: busybox
Container 2 Name: gold
Container 2 Image: redis

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

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 `<kubctl logs -f [ pod-name ]>` command.

13−Where is the default kubeconfig file located in the current environment?

14−How many clusters are defined in the default kubeconfig file?15- What is the user configured in the current context?

16- Create a `Persistent Volume` with the given specification.

Volume Name: pv-log

Storage: 100Mi

Access Modes: ReadWriteMany
Host Path: /pv/log

17- Create a `Persistent Volume Claim` with the given specification.

Volume Name: claim-log-1

Storage Request: 50Mi

Access Modes: ReadWriteMany

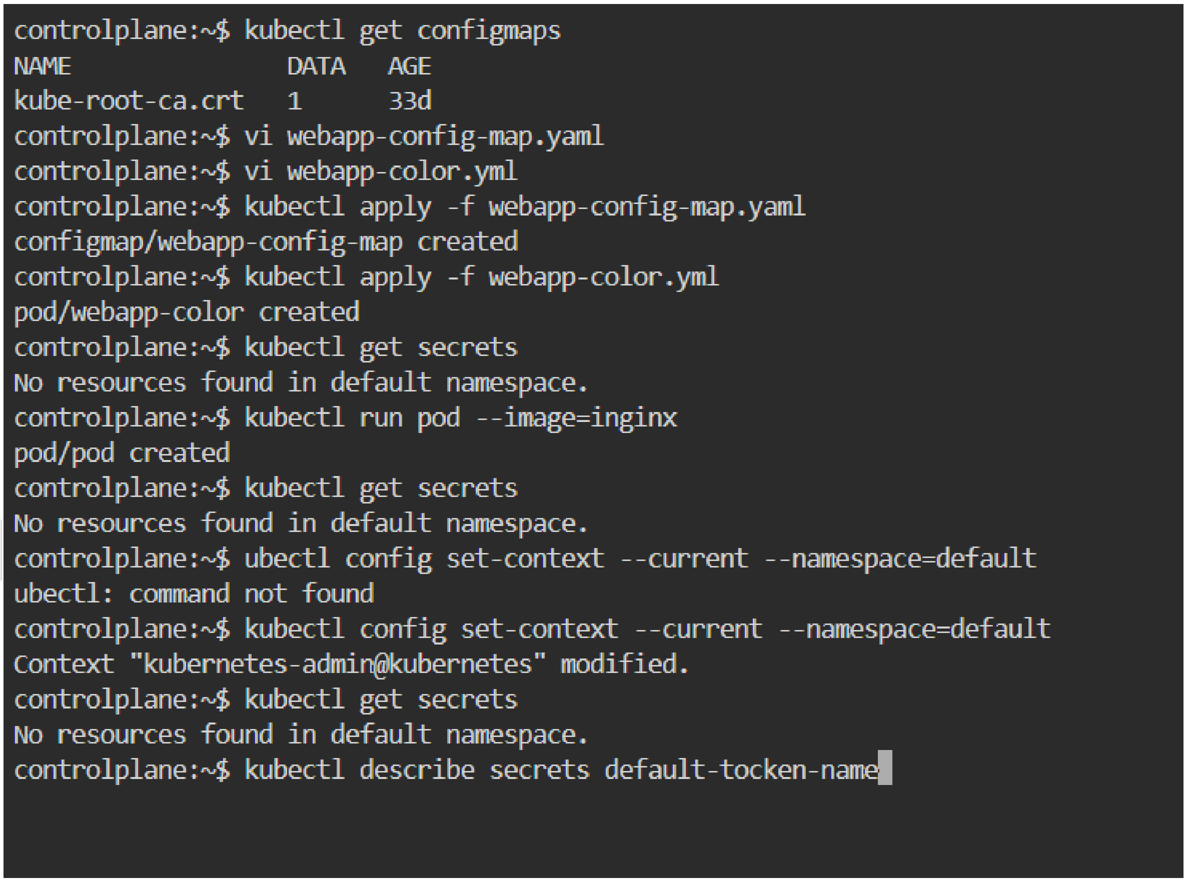
18- Create a `webapp` pod to use the persistent volume claim as its storage.

Name: webapp

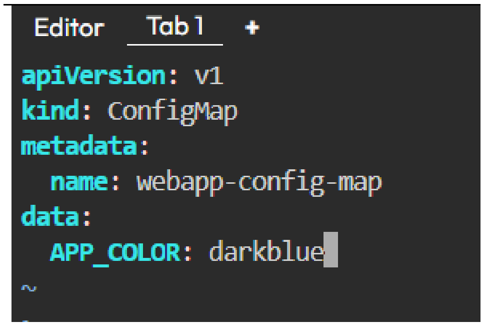
Image Name: nginx

Volume: PersistentVolumeClaim=claim-log-1

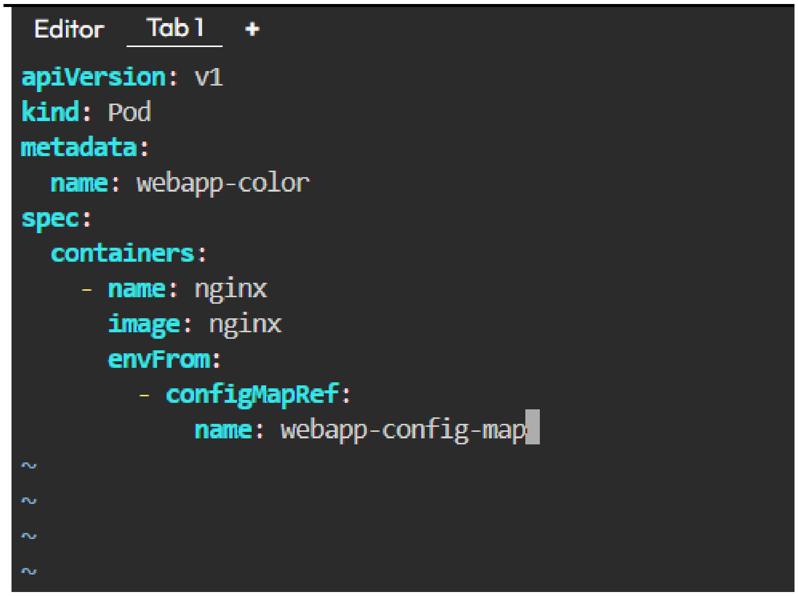
Volume Mount: /var/log/nginx



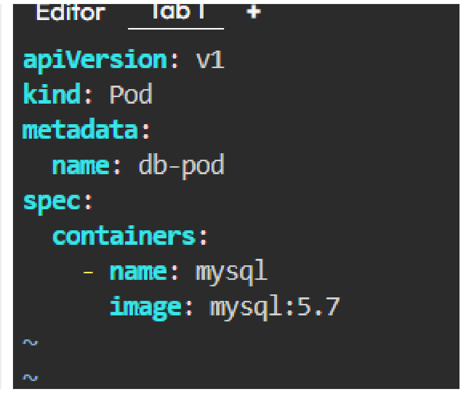
```
controlplane:~$ kubectl get configmaps
NAME               DATA   AGE
kube-root-ca.crt   1      33d
controlplane:~$ vi webapp-config-map.yaml
controlplane:~$ vi webapp-color.yml
controlplane:~$ kubectl apply -f webapp-config-map.yaml
configmap/webapp-config-map created
controlplane:~$ kubectl apply -f webapp-color.yml
pod/webapp-color created
controlplane:~$ kubectl get secrets
No resources found in default namespace.
controlplane:~$ kubectl run pod --image=inginx
pod/pod created
controlplane:~$ kubectl get secrets
No resources found in default namespace.
controlplane:~$ ubectl config set-context --current --namespace=default
ubectl: command not found
controlplane:~$ kubectl config set-context --current --namespace=default
Context "kubernetes-admin@kubernetes" modified.
controlplane:~$ kubectl get secrets
No resources found in default namespace.
controlplane:~$ kubectl describe secrets default-tocken-name
```



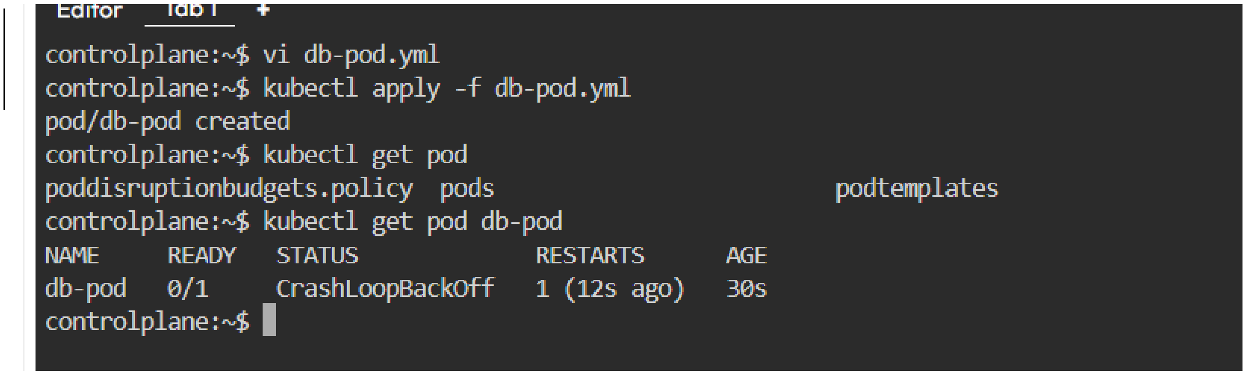
```
apiVersion: v1
kind: ConfigMap
metadata:
  name: webapp-config-map
data:
  APP_COLOR: darkblue
~
```



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

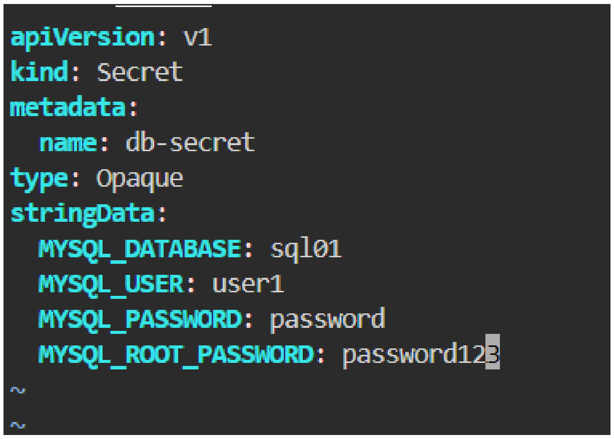


```
apiVersion: v1
kind: Pod
metadata:
  name: db-pod
spec:
  containers:
    - name: mysql
      image: mysql:5.7
~
~
```

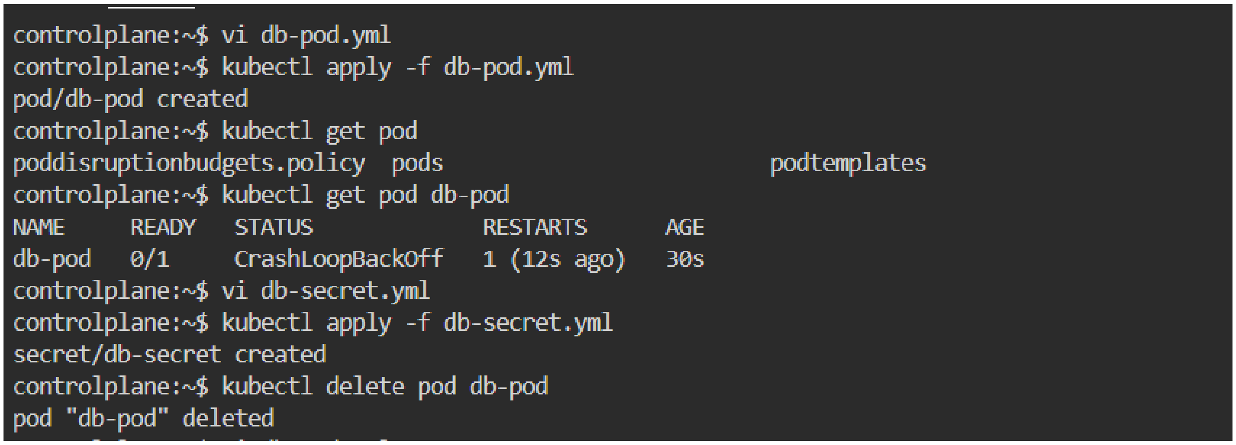


```
controlplane:~$ vi db-pod.yml
controlplane:~$ kubectl apply -f db-pod.yml
pod/db-pod created
controlplane:~$ kubectl get pod
poddisruptionbudgets.policy  pods                        podtemplates
controlplane:~$ kubectl get pod db-pod
NAME     READY   STATUS             RESTARTS      AGE
db-pod   0/1     CrashLoopBackOff   1 (12s ago)   30s
controlplane:~$
```

.

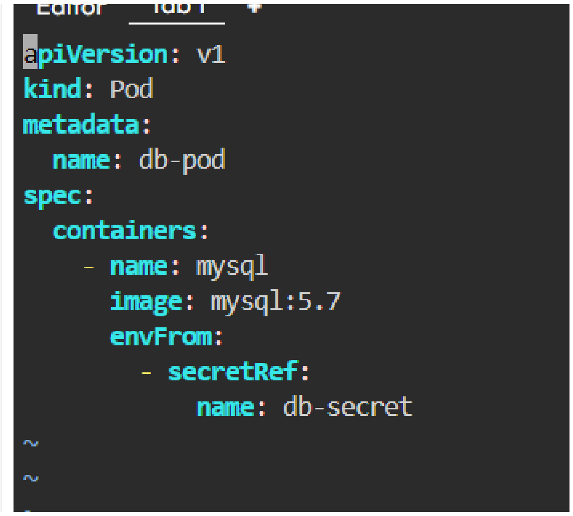


```
apiVersion: v1
kind: Secret
metadata:
  name: db-secret
type: Opaque
stringData:
  MYSQL_DATABASE: sql01
  MYSQL_USER: user1
  MYSQL_PASSWORD: password
  MYSQL_ROOT_PASSWORD: password123
~
~
```

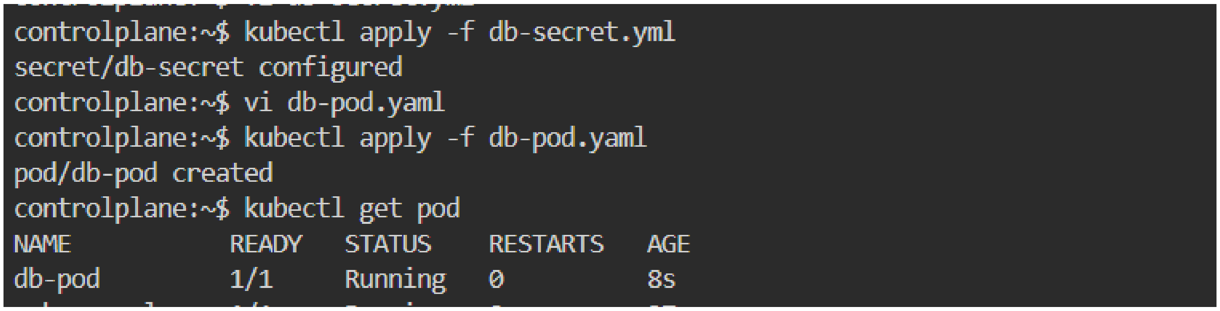


```
controlplane:~$ vi db-pod.yml
controlplane:~$ kubectl apply -f db-pod.yml
pod/db-pod created
controlplane:~$ kubectl get pod
poddisruptionbudgets.policy  pods                          podtemplates
controlplane:~$ kubectl get pod db-pod
NAME     READY   STATUS             RESTARTS      AGE
db-pod   0/1     CrashLoopBackOff   1 (12s ago)   30s
controlplane:~$ vi db-secret.yml
controlplane:~$ kubectl apply -f db-secret.yml
secret/db-secret created
controlplane:~$ kubectl delete pod db-pod
pod "db-pod" deleted
```

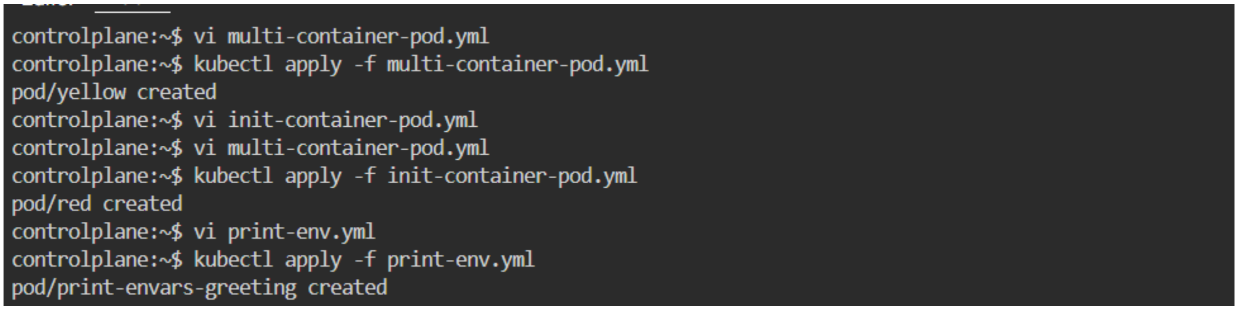
لو مش موجودين البود مش هيشتغل environment variables بتحتاج mysql البود مش ريدي لان



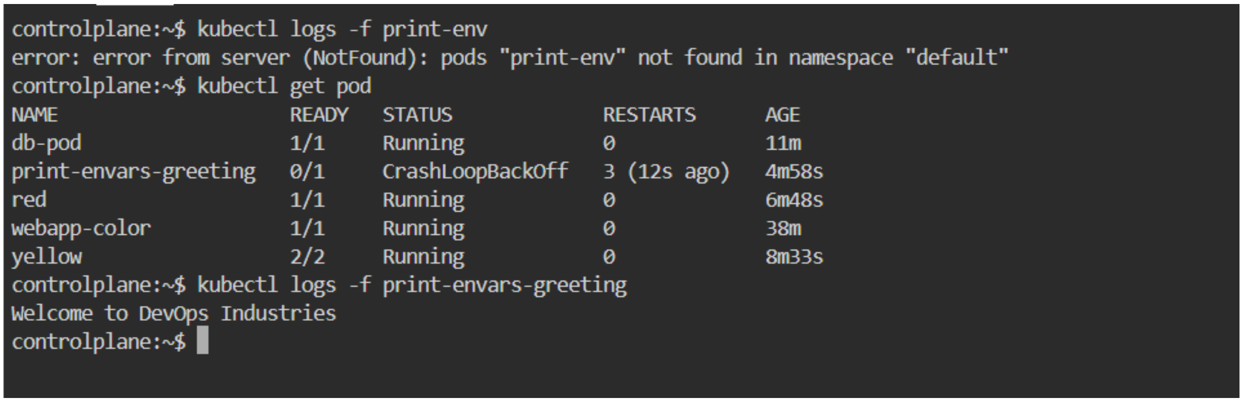
```
apiVersion: v1
kind: Pod
metadata:
  name: db-pod
spec:
  containers:
    - name: mysql
      image: mysql:5.7
      envFrom:
        - secretRef:
            name: db-secret
~
~
```



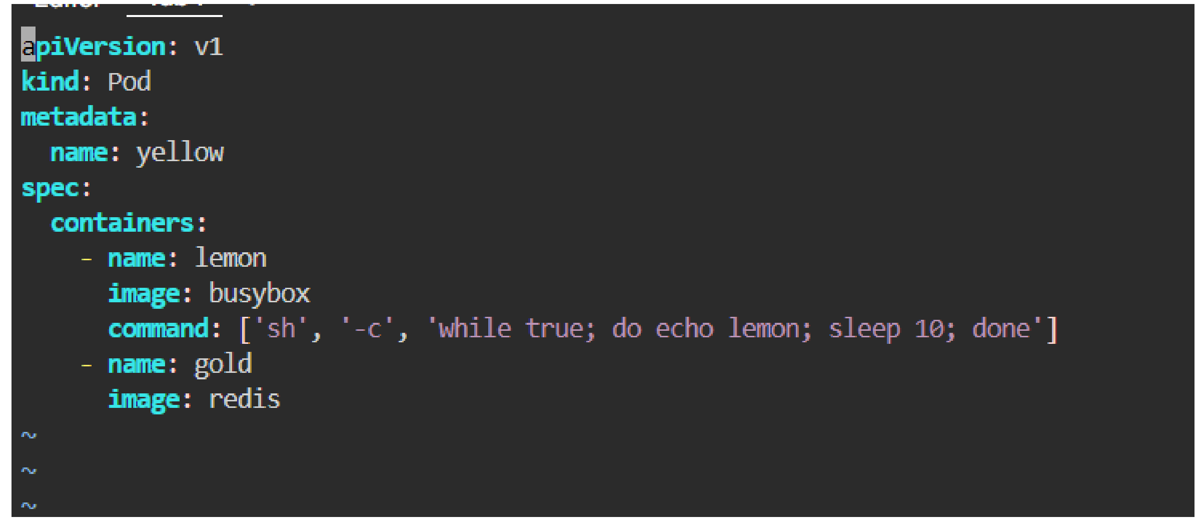
```
controlplane:~$ kubectl apply -f db-secret.yml
secret/db-secret configured
controlplane:~$ vi db-pod.yaml
controlplane:~$ kubectl apply -f db-pod.yaml
pod/db-pod created
controlplane:~$ kubectl get pod
NAME            READY   STATUS    RESTARTS   AGE
db-pod          1/1     Running   0          8s
```



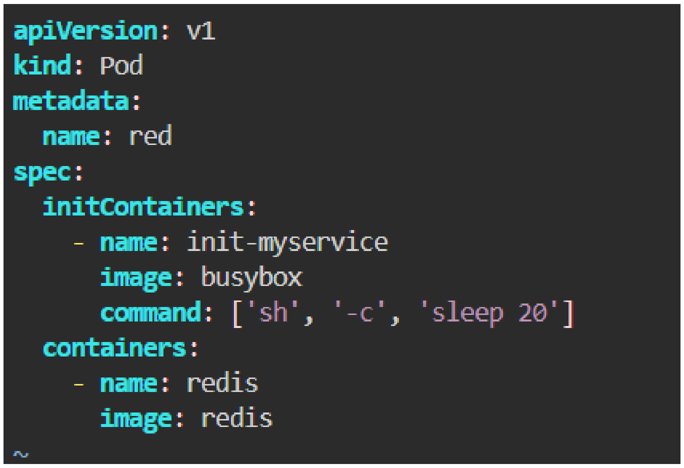
```
controlplane:~$ vi multi-container-pod.yml
controlplane:~$ kubectl apply -f multi-container-pod.yml
pod/yellow created
controlplane:~$ vi init-container-pod.yml
controlplane:~$ vi multi-container-pod.yml
controlplane:~$ kubectl apply -f init-container-pod.yml
pod/red created
controlplane:~$ vi print-env.yml
controlplane:~$ kubectl apply -f print-env.yml
pod/print-envars-greeting created
```



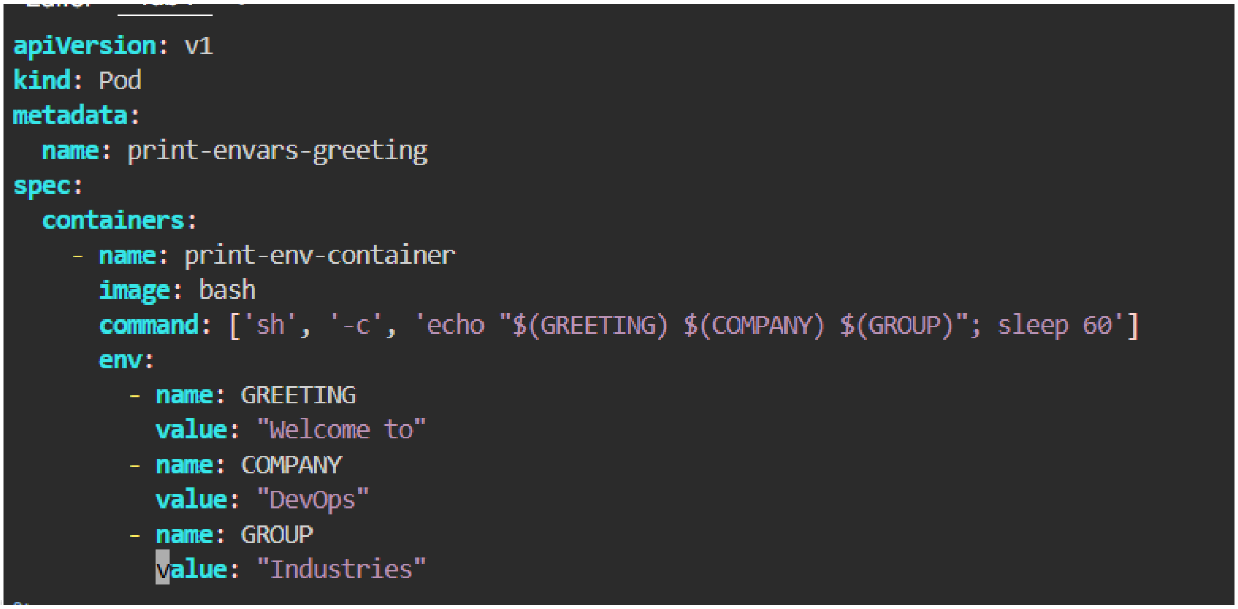
```
controlplane:~$ kubectl logs -f print-env
error: error from server (NotFound): pods "print-env" not found in namespace "default"
controlplane:~$ kubectl get pod
NAME                    READY   STATUS             RESTARTS      AGE
db-pod                  1/1     Running            0             11m
print-envars-greeting   0/1     CrashLoopBackOff   3 (12s ago)   4m58s
red                     1/1     Running            0             6m48s
webapp-color            1/1     Running            0             38m
yellow                  2/2     Running            0             8m33s
controlplane:~$ kubectl logs -f print-envars-greeting
Welcome to DevOps Industries
controlplane:~$ 
```



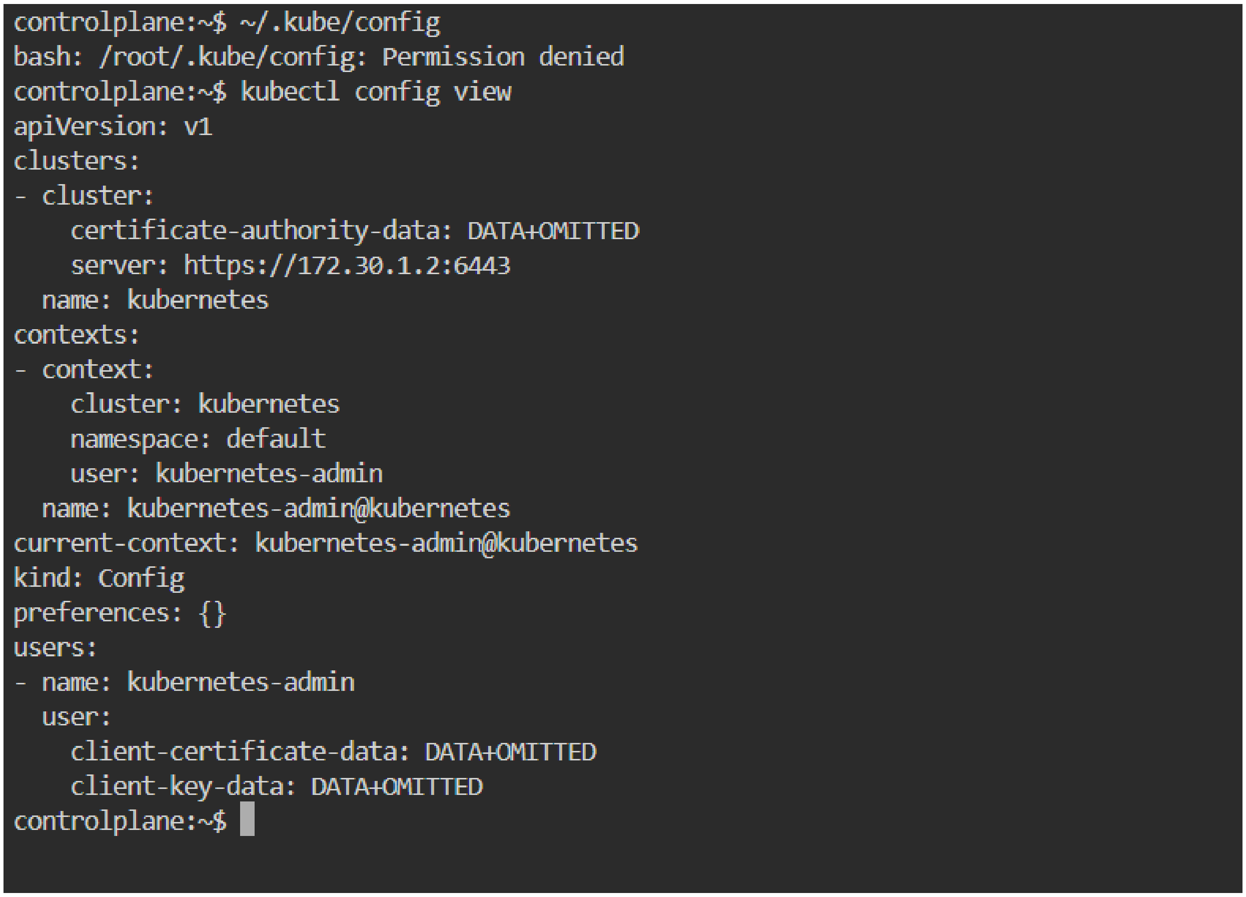
```
apiVersion: v1
kind: Pod
metadata:
  name: yellow
spec:
  containers:
    - name: lemon
      image: busybox
      command: ['sh', '-c', 'while true; do echo lemon; sleep 10; done']
    - name: gold
      image: redis
~
~
~
```



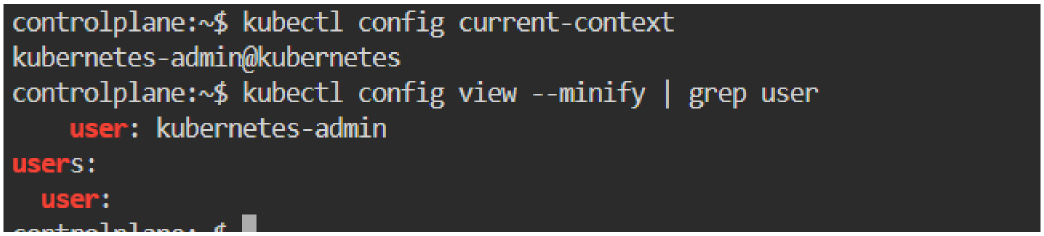
```
apiVersion: v1
kind: Pod
metadata:
  name: red
spec:
  initContainers:
    - name: init-myservice
      image: busybox
      command: ['sh', '-c', 'sleep 20']
  containers:
    - name: redis
      image: redis
~
```



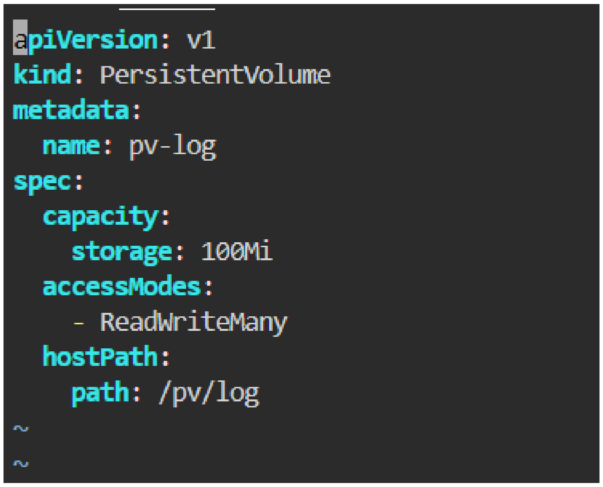
```
apiVersion: v1
kind: Pod
metadata:
  name: print-envars-greeting
spec:
  containers:
    - name: print-env-container
      image: bash
      command: ['sh', '-c', 'echo "$(GREETING) $(COMPANY) $(GROUP)"; sleep 60']
      env:
        - name: GREETING
          value: "Welcome to"
        - name: COMPANY
          value: "DevOps"
        - name: GROUP
          value: "Industries"
```



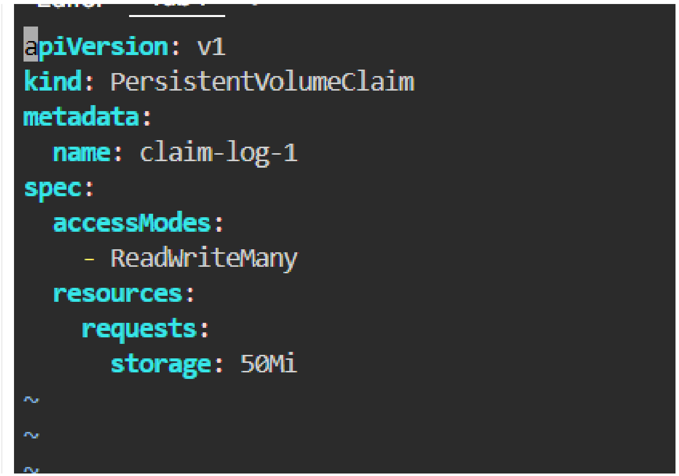
```
controlplane:~$ ~/.kube/config
bash: /root/.kube/config: Permission denied
controlplane:~$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://172.30.1.2:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    namespace: default
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: DATA+OMITTED
    client-key-data: DATA+OMITTED
controlplane:~$
```



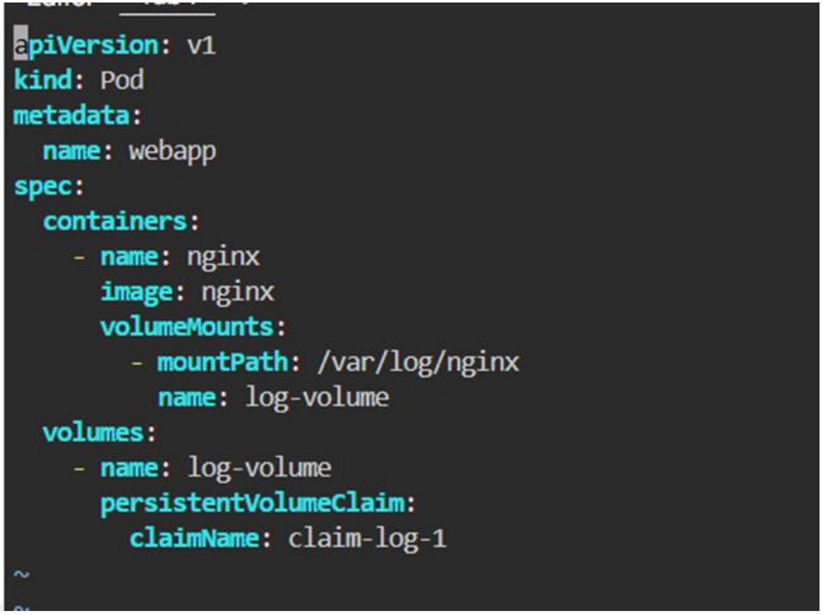
```
controlplane:~$ kubectl config current-context
kubernetes-admin@kubernetes
controlplane:~$ kubectl config view --minify | grep user
    user: kubernetes-admin
users:
  user:
controlplane:~$
```



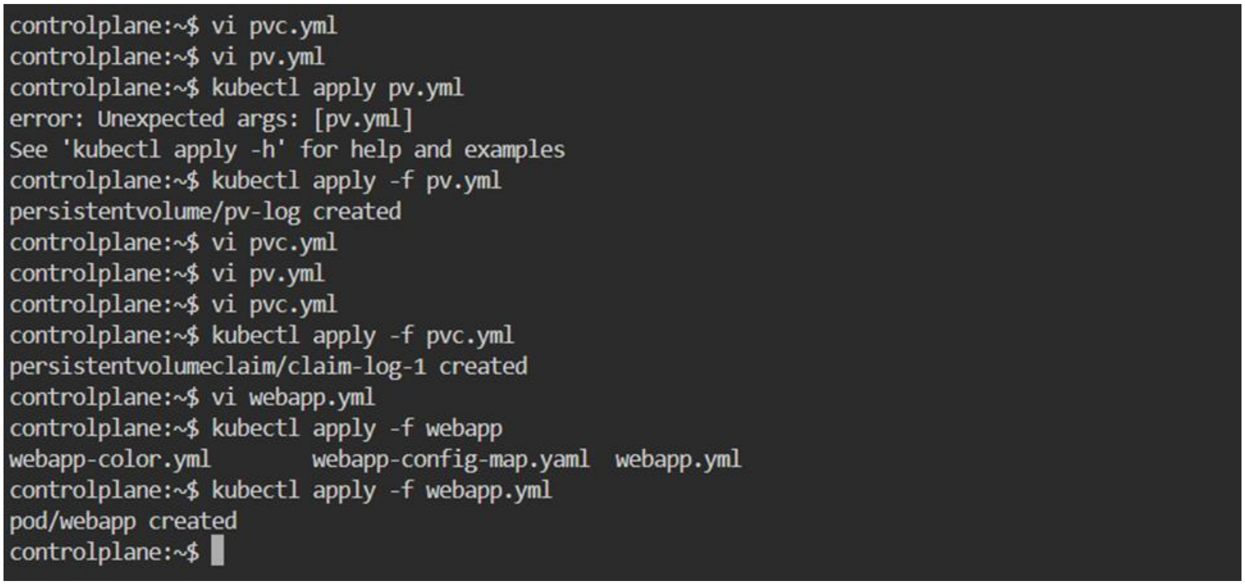
```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-log
spec:
  capacity:
    storage: 100Mi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: /pv/log
~
~
```



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



```
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
~
```



```
controlplane:~$ vi pvc.yml
controlplane:~$ vi pv.yml
controlplane:~$ kubectl apply pv.yml
error: Unexpected args: [pv.yml]
See 'kubectl apply -h' for help and examples
controlplane:~$ kubectl apply -f pv.yml
persistentvolume/pv-log created
controlplane:~$ vi pvc.yml
controlplane:~$ vi pv.yml
controlplane:~$ vi pvc.yml
controlplane:~$ kubectl apply -f pvc.yml
persistentvolumeclaim/claim-log-1 created
controlplane:~$ vi webapp.yml
controlplane:~$ kubectl apply -f webapp
webapp-color.yml        webapp-config-map.yaml  webapp.yml
controlplane:~$ kubectl apply -f webapp.yml
pod/webapp created
controlplane:~$ 
```