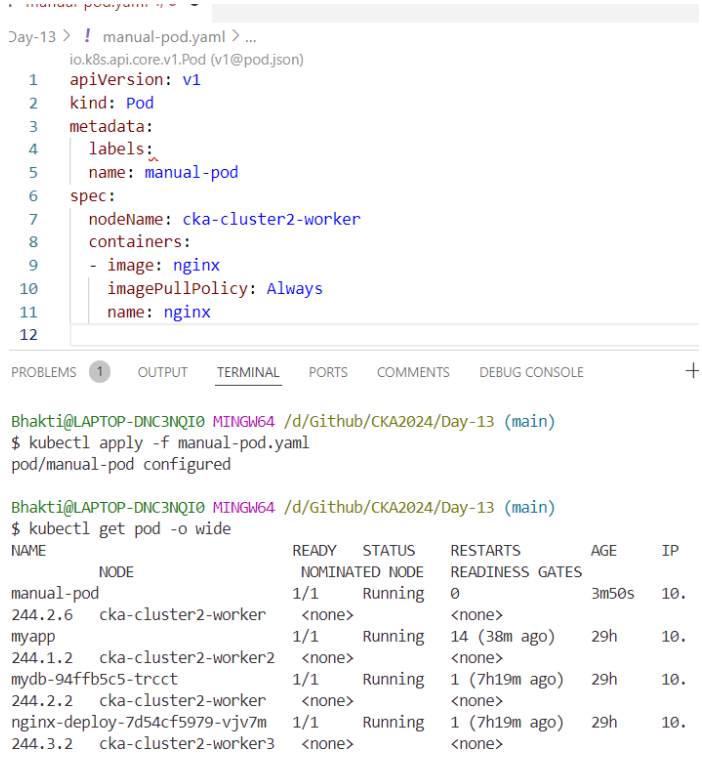
Task

1. Create a pod and try to schedule it manually without the scheduler.



```
Day-13 > ! manual-pod.yaml > ...
        io.k8s.api.core.v1.Pod (v1@pod.json)
  1  apiVersion: v1
  2  kind: Pod
  3  metadata:
  4    labels:
  5    name: manual-pod
  6  spec:
  7    nodeName: cka-cluster2-worker
  8    containers:
  9    - image: nginx
 10      imagePullPolicy: Always
 11      name: nginx
 12
```

```
PROBLEMS 1  OUTPUT  TERMINAL  PORTS  COMMENTS  DEBUG CONSOLE

Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024/Day-13 (main)
$ kubectl apply -f manual-pod.yaml
pod/manual-pod configured

Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024/Day-13 (main)
$ kubectl get pod -o wide
NAME                           READY   STATUS    RESTARTS        AGE     IP
          NODE                  NOMINATED NODE   READINESS GATES
manual-pod                     1/1     Running   0               3m50s   10.
244.2.6   cka-cluster2-worker    <none>           <none>
myapp                          1/1     Running   14 (38m ago)    29h     10.
244.1.2   cka-cluster2-worker2   <none>           <none>
mydb-94ffb5c5-trcct            1/1     Running   1 (7h19m ago)   29h     10.
244.2.2   cka-cluster2-worker    <none>           <none>
nginx-deploy-7d54cf5979-vjv7m  1/1     Running   1 (7h19m ago)   29h     10.
244.3.2   cka-cluster2-worker3   <none>           <none>
```

2. Login to the control plane node and go to the directory of default static pod manifests and try to restart the control plane components

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024/Day-13 (main)
$ docker ps | grep control
3dfb3b89ac3a   kindest/node:v1.30.0   "/usr/local/bin/entr…"   31 hours ago
 Up 7 hours   127.0.0.1:61265->6443/tcp   cka-cluster2-control-plane
c8dbfc3b8f26   kindest/node:v1.30.0   "/usr/local/bin/entr…"   2 weeks ago
 Up 7 hours   127.0.0.1:56038->6443/tcp   cka-cluster1-control-plane

Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024/Day-13 (main)
$ docker exec -it cka-cluster2-control-plane bash
root@cka-cluster2-control-plane:/# cd /etc/kubernetes/manifests/
root@cka-cluster2-control-plane:/etc/kubernetes/manifests# ls -lrt
total 16
-rw------- 1 root root 1463 Jul  9 13:12 kube-scheduler.yaml
-rw------- 1 root root 3436 Jul  9 13:12 kube-controller-manager.yaml
-rw------- 1 root root 3896 Jul  9 13:12 kube-apiserver.yaml
-rw------- 1 root root 2422 Jul  9 13:12 etcd.yaml
root@cka-cluster2-control-plane:/etc/kubernetes/manifests# mv kube-scheduler.
yaml /tamp
root@cka-cluster2-control-plane:/etc/kubernetes/manifests#
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
$ docker ps | grep scheduler

Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
$ kubectl get pod
NAME                          READY   STATUS    RESTARTS        AGE
manual-pod                    1/1     Running   0               14m
myapp                         1/1     Running   14 (48m ago)    29h
mydb-94ffb5c5-trcct           1/1     Running   1 (7h29m ago)   29h
nginx-deploy-7d54cf5979-vjv7m 1/1     Running   1 (7h29m ago)   30h

Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
$ kubectl get pod -n kube-system | grep scheduler
```

```
root@cka-cluster2-control-plane:/etc/kubernetes/manifests# mv /tamp kube-sche
duler.yaml
root@cka-cluster2-control-plane:/etc/kubernetes/manifests# ls -lrt
total 16
-rw------- 1 root root 1463 Jul  9 13:12 kube-scheduler.yaml
-rw------- 1 root root 3436 Jul  9 13:12 kube-controller-manager.yaml
-rw------- 1 root root 3896 Jul  9 13:12 kube-apiserver.yaml
-rw------- 1 root root 2422 Jul  9 13:12 etcd.yaml
root@cka-cluster2-control-plane:/etc/kubernetes/manifests#

Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
$ kubectl get pod -n kube-system | grep scheduler
kube-scheduler-cka-cluster2-control-plane            1/1     Running   0
          35s
```

3. Create 3 pods with the name as pod1, pod2 and pod3 based on the nginx image and use labels as env:test, env:dev and env:prod for each of these pods respectively.

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
$ kubectl run pod1 --image=nginx --labels=env=test && kubectl run pod2 --imag
e=nginx --labels=env=dev && kubectl run pod3 --image=nginx --labels=env=prod
pod/pod1 created
pod/pod2 created
pod/pod3 created

Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
$ kubectl get pod
NAME                          READY   STATUS    RESTARTS         AGE
myapp                         1/1     Running   15 (7m21s ago)   30h
mydb-94ffb5c5-trcct           1/1     Running   1 (7h48m ago)    30h
nginx-deploy-7d54cf5979-vjv7m 1/1     Running   1 (7h48m ago)    30h
pod1                          1/1     Running   0                15s
pod2                          1/1     Running   0                15s
pod3                          1/1     Running   0                15s
```

4. Then using the kubectl commands, filter the pods that have labels dev and prod.

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
$ kubectl get pod --selector='env in (dev,prod)'
NAME   READY   STATUS    RESTARTS   AGE
pod2   1/1     Running   0          65s
pod3   1/1     Running   0          65s
```