Mohanad Mahmoud Sayed

1- How many `pods` exist on the system?

```
controlplane:~$ k get po
No resources found in default namespace.
controlplane:~$
```

2- How many `Nodes` exist on the system?

→ There is one worker node which is node01

```
NAME           STATUS   ROLES           AGE   VERSION
controlplane   Ready    control-plane   29d   v1.32.1
node01         Ready    <none>          29d   v1.32.1
controlplane:~$
```

3- Create a new pod with the `nginx` image.
Image name: nginx

```
controlplane:~$ kubectl run nginx --image=nginx
pod/nginx created
controlplane:~$ k get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          23s
controlplane:~$
```

4- Which nodes are these pods placed on?

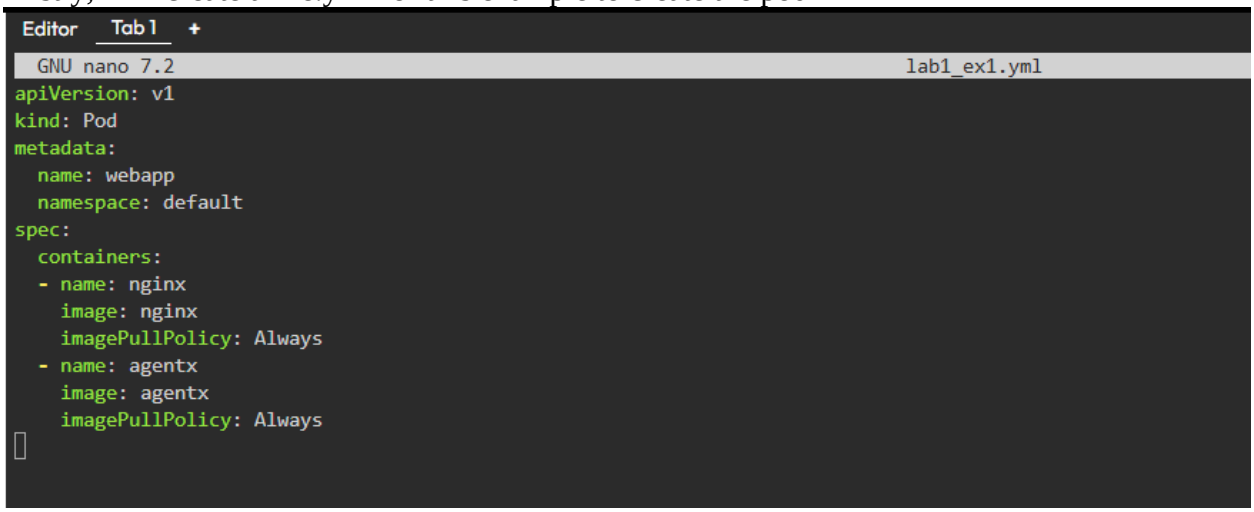
➔ On node: node01

```
controlplane:~$ k get pods -o wide
NAME    READY   STATUS    RESTARTS   AGE   IP           NODE     NOMINATED NODE   READINESS GATES
nginx   1/1     Running   0          56s   192.168.1.4   node01   <none>           <none>
controlplane:~$
```

5- Create pod from the below yaml using kubectl apply command

```
apiVersion: v1
kind: Pod
metadata:
  name: webapp
  namespace: default
spec:
  containers:
  - image: nginx
    imagePullPolicy: Always
    name: nginx
  - image: agentx
    imagePullPolicy: Always
    name: agentx
```

Firstly, let's create a file.yml for this example to create the pod



Then, let's apply

6- How many containers are part of the pod `webapp`

➔ Webapp pod has 2 containers: nginx and agentx

```
controlplane:~$ kubectl get pod webapp -o jsonpath='{.spec.containers[*].name}'
nginx agentxcontrolplane:~$
controlplane:~$
```

7- What images are used in the new `webapp` pod?

➔ nginx and agentx
(but nginx only is used, agentx is not pulled from dockerhub as shown in the ss in question 5)

```
controlplane:~$ kubectl get pod webapp -o jsonpath='{.spec.containers[*].image}'
nginx agentxcontrolplane:~$
controlplane:~$
```

8- What is the state of the container `agentx` in the pod `webapp`

```
agentx:
  Container ID:
  Image:          agentx
  Image ID:
  Port:           <none>
  Host Port:      <none>
  State:          Waiting
    Reason:       ImagePullBackOff
  Ready:          False
  Restart Count:  0
  Environment:    <none>
```

- **State:** Waiting
- **Reason:** ImagePullBackOff

9- Why do you think the container `agentx` in pod `webapp` is in error?

→**Reason:** Kubernetes is trying to pull the image agentx but **fails** because it likely does **not exist on a public registry like Docker Hub**

This causes the container to go into the **ImagePullBackOff** state — which means it tried to pull the image, failed, and now backs off from trying again immediately.

10- Delete the `webapp` Pod.

```
controlplane:~$ kubectl delete pod webapp
pod "webapp" deleted
controlplane:~$
```

11- Create a new pod with the name `redis` and with the image `redis123`.

- Name: redis
- Image Name: redis123

Firstly, let's create a file.yml for this example to create the pod

```
Editor   Tab 1   +
  GNU nano 7.2                                          redis_pod.yml *
apiVersion: v1
kind: Pod
metadata:
  name: redis
spec:
  containers:
  - name: redis
    image: redis123
```

Then, let's apply

```
controlplane:~$ kubectl apply -f redis_pod.yml
pod/redis created
controlplane:~$ k get pods
NAME    READY   STATUS              RESTARTS   AGE
nginx   1/1     Running             0          35m
redis   0/1     ContainerCreating   0          3s
controlplane:~$
```

→Note: the image is not running because redis123 is not found on dockerhub to pull

12- Now change the image on this pod to `redis`.
Once done, the pod should be in a `running` state.

Firstly: let's edit the pod file to change the image name from redis123 to redis

```
Editor   Tab 1   +
  GNU nano 7.2                                          redis_pod.yml *
apiVersion: v1
kind: Pod
metadata:
  name: redis
spec:
  containers:
  - name: redis
    image: redis
```

Then, let's apply

```
controlplane:~$ nano redis_pod.yml
controlplane:~$ kubectl apply -f redis_pod.yml
pod/redis configured
controlplane:~$ k get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          38m
redis   1/1     Running   0          2m44s
controlplane:~$
```

- ✓ The pod is in the running state

13- Create a pod called `my-pod` of image `nginx:alpine`

```
controlplane:~$ kubectl run my-pod --image=nginx:alpine
pod/my-pod created
controlplane:~$ k get pods
NAME     READY   STATUS    RESTARTS   AGE
my-pod   1/1     Running   0          7s
```

14- Delete the pod called `my-pod`

```
controlplane:~$ k delete pod my-pod
pod "my-pod" deleted
controlplane:~$ k get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          43m
redis   1/1     Running   0          7m31s
controlplane:~$
```