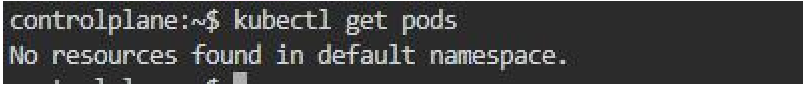
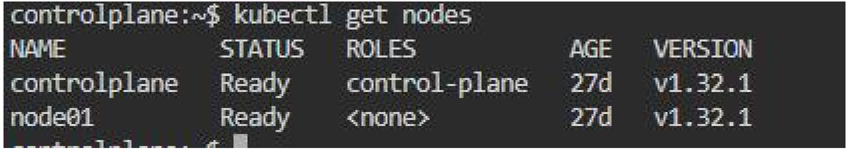
1- How many `pods` exist on the system?



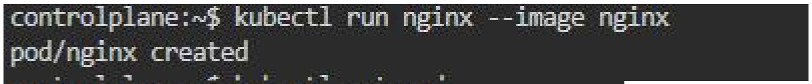
```
controlplane:~$ kubectl get pods
No resources found in default namespace.
```

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

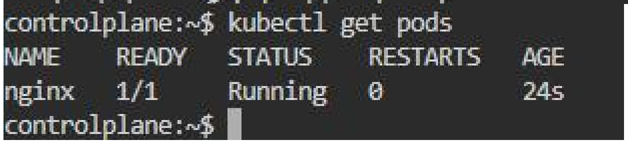


```
controlplane:~$ kubectl get nodes
NAME           STATUS   ROLES           AGE   VERSION
controlplane   Ready    control-plane   27d   v1.32.1
node01         Ready    <none>          27d   v1.32.1
```

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

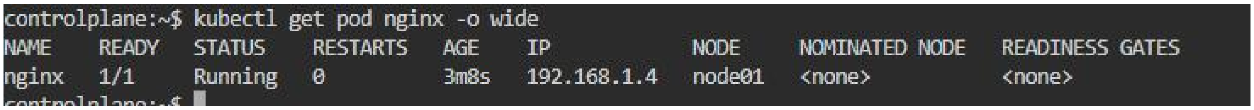


```
controlplane:~$ kubectl run nginx --image nginx
pod/nginx created
```



```
controlplane:~$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          24s
controlplane:~$
```

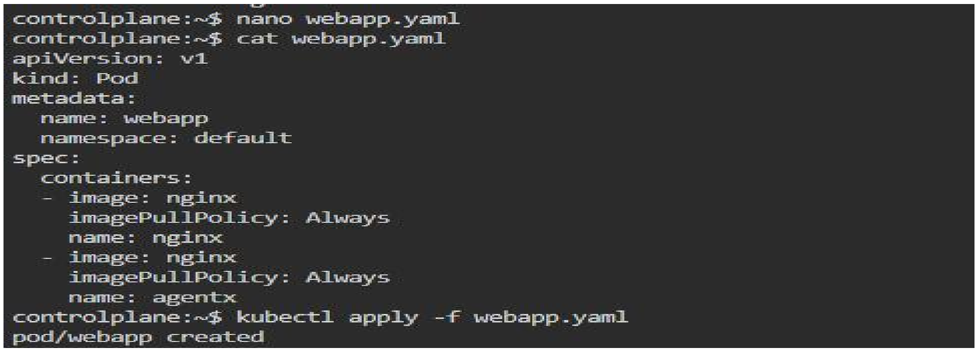
4- Which nodes are these pods placed on?



```
controlplane:~$ kubectl get pod nginx -o wide
NAME    READY   STATUS    RESTARTS   AGE    IP            NODE     NOMINATED NODE   READINESS GATES
nginx   1/1     Running   0          3m8s   192.168.1.4   node01   <none>           <none>
```

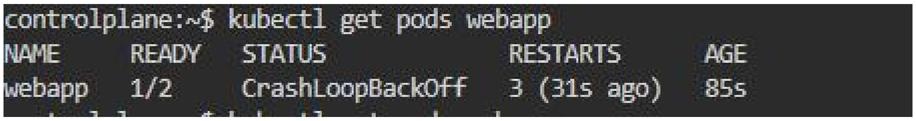
5- Create pod from the below yaml using kubectl apply command

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



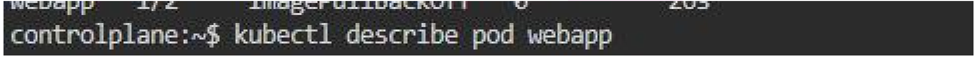
```
controlplane:~$ nano webapp.yaml
controlplane:~$ cat webapp.yaml
apiVersion: v1
kind: Pod
metadata:
  name: webapp
  namespace: default
spec:
  containers:
  - image: nginx
    imagePullPolicy: Always
    name: nginx
  - image: nginx
    imagePullPolicy: Always
    name: agentx
controlplane:~$ kubectl apply -f webapp.yaml
pod/webapp created
```

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

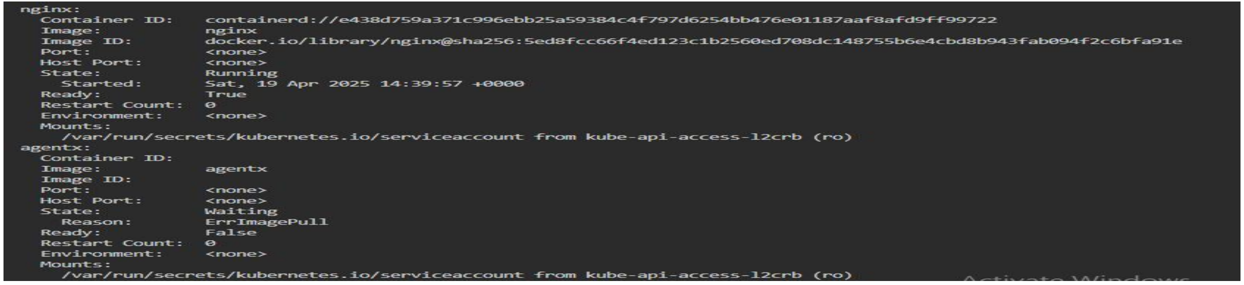


```
controlplane:~$ kubectl get pods webapp
NAME     READY   STATUS             RESTARTS      AGE
webapp   1/2     CrashLoopBackOff   3 (31s ago)   85s
```

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

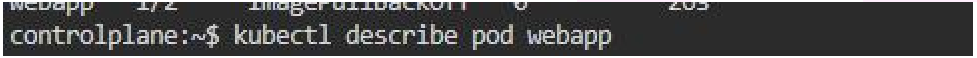


```
controlplane:~$ kubectl describe pod webapp
```

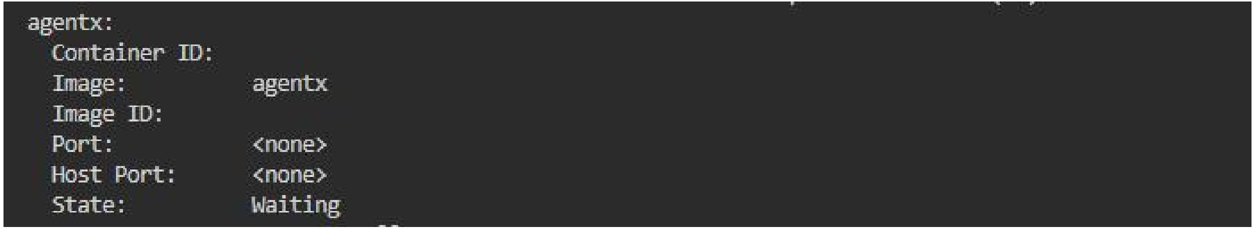


```
 nginx:
   Container ID:   containerd://e438d759a371c996ebb25a59384c4f797d6254bb476e01187aaf8afd9ff99722
   Image:          nginx
   Image ID:       docker.io/library/nginx@sha256:5ed8fcc66f4ed123c1b2560ed708dc148755b6e4cbd8b943fab094f2c6bfa91e
   Port:           <none>
   Host Port:      <none>
   State:          Running
     Started:      Sat, 19 Apr 2025 14:39:57 +0000
   Ready:          True
   Restart Count:  0
   Environment:    <none>
   Mounts:
     /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-l2crb (ro)
 agentx:
   Container ID:
   Image:          agentx
   Image ID:
   Port:           <none>
   Host Port:      <none>
   State:          Waiting
     Reason:       ErrImagePull
   Ready:          False
   Restart Count:  0
   Environment:    <none>
   Mounts:
     /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-l2crb (ro)
```

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



```
controlplane:~$ kubectl describe pod webapp
```

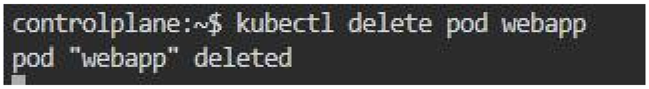


```
 agentx:
   Container ID:
   Image:          agentx
   Image ID:
   Port:           <none>
   Host Port:      <none>
   State:          Waiting
```

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

The name: agentx field is not correctly indented under the container definition. It should be part of the same container block.

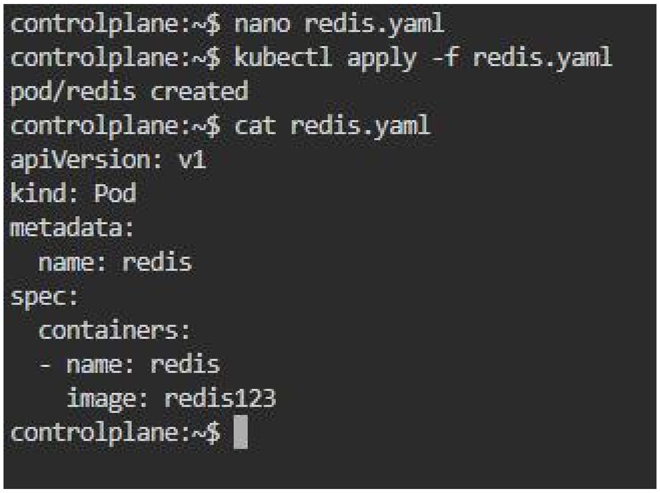
10- Delete the webapp Pod.



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

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

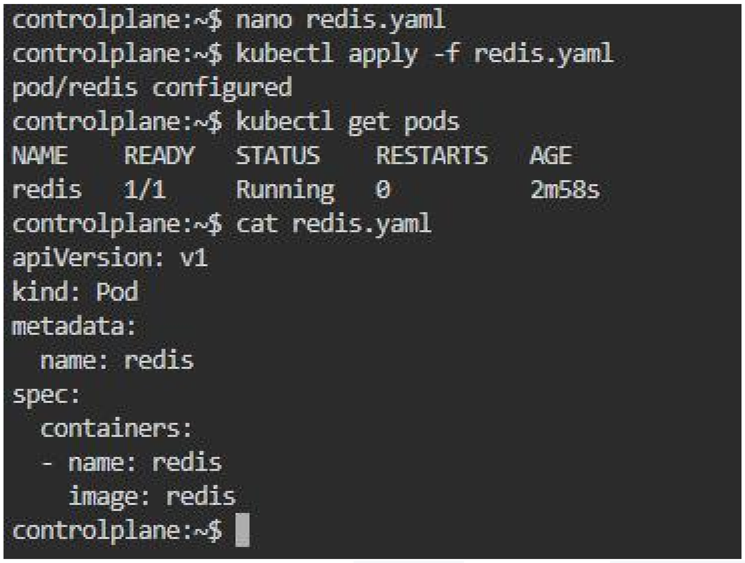
- Name: redis
- Image Name: redis123



```
controlplane:~$ nano redis.yaml
controlplane:~$ kubectl apply -f redis.yaml
pod/redis created
controlplane:~$ cat redis.yaml
apiVersion: v1
kind: Pod
metadata:
  name: redis
spec:
  containers:
  - name: redis
    image: redis123
controlplane:~$
```

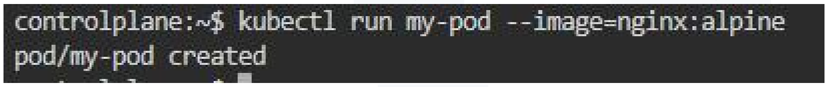
12- Now change the image on this pod to `redis`.

Once done, the pod should be in a `running` state.



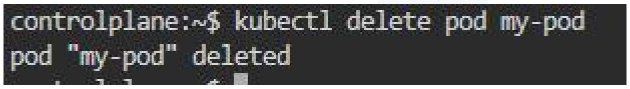
```
controlplane:~$ nano redis.yaml
controlplane:~$ kubectl apply -f redis.yaml
pod/redis configured
controlplane:~$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
redis   1/1     Running   0          2m58s
controlplane:~$ cat redis.yaml
apiVersion: v1
kind: Pod
metadata:
  name: redis
spec:
  containers:
  - name: redis
    image: redis
controlplane:~$
```

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



```
controlplane:~$ kubectl run my-pod --image=nginx:alpine
pod/my-pod created
```

14- Delete the pod called *my-pod*



```
controlplane:~$ kubectl delete pod my-pod
pod "my-pod" deleted
```