1- How many pods exist on the system?

- At the beginning, there are no pods.

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

2- How many Nodes exist on the system?

- There is one worker node named node01.

```
controlplane:~$ k 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:~$ k run nginx --image nginx
pod/nginx created
controlplane:~$ k get po
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          13s
```

4- Which nodes are these pods placed on?

- The node name is node01.

```
controlplane:~$ k get po -o wide
NAME    READY   STATUS    RESTARTS   AGE   IP           NODE     NOMINATED NODE   READINESS GATES
nginx   1/1     Running   0          70s   192.168.1.5   node01   <none>           <none>
```

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

- There is an error in the pod Status ErrImagePull.

```
controlplane:~$ vi pod-definition.yaml
controlplane:~$ k apply -f pod-definition.yaml
pod/webapp created
controlplane:~$ k get po
NAME     READY   STATUS         RESTARTS   AGE
webapp   1/2     ErrImagePull   0          10s
```

6- How many containers are part of the pod webapp

- There are two containers (nginx, agentx but it failed to start).

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

- Two images are used (nginx) and (agentx but Kubernetes failed to pull it from Dockerhub.

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

- The status of agentx is ErrImagePull.

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

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

- The container agentx failed to run because Kubernetes was **unable to pull the image** from Docker Hub.

```
Events:
  Type     Reason     Age                  From               Message
  ----     ------     ----                 ----               -------
  Normal   Scheduled  5m14s                default-scheduler  Successfully assigned default/webapp to node01
  Normal   Pulling    5m15s                kubelet            Pulling image "nginx"
  Normal   Pulled     5m14s                kubelet            Successfully pulled image "nginx" in 268ms (268ms including waiting). Image size: 72403299 bytes.
  Normal   Created    5m14s                kubelet            Created container: nginx
  Normal   Started    5m14s                kubelet            Started container nginx
  Normal   Pulling    2m12s (x5 over 5m14s)  kubelet          Pulling image "agentx"
  Warning  Failed     2m9s (x5 over 5m11s)   kubelet          Failed to pull image "agentx": failed to pull and unpack image "docker.io/library/agentx:latest": failed to resolve
 reference "docker.io/library/agentx:latest": pull access denied, repository does not exist or may require authorization: server message: insufficient_scope: authorization failed
  Warning  Failed     2m9s (x5 over 5m11s)   kubelet          Error: ErrImagePull
  Warning  Failed     61s (x15 over 5m10s)   kubelet          Error: ImagePullBackOff
  Normal   BackOff    12s (x19 over 5m10s)   kubelet          Back-off pulling image "agentx"
controlplane:~$
```

10- Delete the webapp Pod.

```
controlplane:~$ k delete pod webapp
pod "webapp" deleted
controlplane:~$ k get po
No resources found in default namespace.
```

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

- Name: redis
- Image Name: redis123

- The status ErrImagePull because the image name (redis123) is not found on Dockerhub

```
controlplane:~$ k run redis --image=redis123 --dry-run=client -o yaml > redis-pod-def.yaml
controlplane:~$ cat redis-pod-def.yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: redis
  name: redis
spec:
  containers:
  - image: redis123
    name: redis
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
controlplane:~$ k apply -f redis-pod-def.yaml
pod/redis created
controlplane:~$ k get po
NAME    READY   STATUS         RESTARTS   AGE
redis   0/1     ErrImagePull   0          5s
controlplane:~$
```

C2
Gene
ral

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

```
controlplane:~$ vi redis-pod-def.yaml
controlplane:~$ k apply -f redis-pod-def.yaml
pod/redis configured
controlplane:~$ cat redis-pod-def.yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: redis
  name: redis
spec:
  containers:
  - image: redis
    name: redis
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
controlplane:~$ k get po
NAME    READY   STATUS    RESTARTS   AGE
redis   1/1     Running   0          4m37s
```

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

```
controlplane:~$ k run my-pod --image=nginx:alpine --dry-run=client -o yaml > my-pod-def.yaml
controlplane:~$ cat my-pod-def.yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: my-pod
  name: my-pod
spec:
  containers:
  - image: nginx:alpine
    name: my-pod
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
controlplane:~$ k apply -f my-pod-def.yaml
pod/my-pod created
controlplane:~$ k get po
NAME     READY   STATUS    RESTARTS   AGE
my-pod   1/1     Running   0          5s
redis    1/1     Running   0          12m
```

14- Delete the pod called my-pod

```
controlplane:~$ k get po
NAME     READY   STATUS    RESTARTS   AGE
my-pod   1/1     Running   0          5s
redis    1/1     Running   0          12m
controlplane:~$ k delete pod my-pod
pod "my-pod" deleted
controlplane:~$ k get po
NAME    READY   STATUS    RESTARTS   AGE
redis   1/1     Running   0          13m
```