1- How many static pods exist in this cluster in all namespaces?

```
controlplane $ kubectl get pods -n kube-system
NAME                                    READY   STATUS    RESTARTS   AGE
coredns-fb8b8dccf-bszpc                 1/1     Running   1          62m
coredns-fb8b8dccf-fhjr5                 1/1     Running   1          62m
etcd-controlplane                       1/1     Running   0          62m
katacoda-cloud-provider-75b6544856-s4bmd   1/1     Running   19         62m
kube-apiserver-controlplane             1/1     Running   0          61m
kube-controller-manager-controlplane    1/1     Running   0          62m
kube-keepalived-vip-9lwvh               1/1     Running   0          62m
kube-proxy-6m29h                        1/1     Running   0          62m
kube-proxy-nj9xh                        1/1     Running   0          62m
kube-scheduler-controlplane             1/1     Running   0          62m
weave-net-gb2r7                         2/2     Running   1          62m
weave-net-tdnn6                         2/2     Running   1          62m
controlplane $
```

2-On which nodes are the static pods created currently?

Master node “controlplane”

3- What is the path of the directory holding the static pod definition files?

/etc/Kubernetes/manifests

4- Create a static pod named static-busybox that uses the busybox image and the command sleep 1000

pod.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: static-bb-pod
  labels:
    app: nginx
spec:
  containers:
  - image: busybox
    name: nginx-demo
    command: ["sleep"]
    args: ["1000"]
```

```
controlplane $ kubectl get pods
NAME            READY   STATUS    RESTARTS   AGE
static-bb-pod   1/1     Running   0          7s
controlplane $
```

5- Edit the image on the static pod to use busybox:1.28.4

```
Terminal
apiVersion: v1
kind: Pod
metadata:
  name: static-bb-pod
  labels:
    app: nginx
spec:
  containers:
  - image: busybox:1.28.4
    name: nginx-demo
    command: ["sleep"]
    args: ["1000"]
~
~
~
~
:wq
```

6- How many ConfigMaps exist in the environment?

```
controlplane $ kubectl get configmaps
No resources found.
controlplane $ kubectl get configmaps -n kube-system
NAME                                 DATA   AGE
coredns                              1      73m
extension-apiserver-authentication   6      73m
kube-proxy                           2      73m
kubeadm-config                       2      73m
kubelet-config-1.14                  1      73m
vip-configmap                        0      73m
weave-net                            0      73m
controlplane $
```

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

```
controlplane $ kubectl create configmap webapp-config-map --from-literal=APP_COLOR=darkblue
configmap/webapp-config-map created
controlplane $ kubectl get configmaps
NAME                 DATA   AGE
webapp-config-map    1      5s
controlplane $
```

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

pod.yaml | web-pod.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: web-app
spec:
  containers:
  - image: nginx
    name: web-app
    envFrom:
    - configMapRef:
        name: webapp-config-map
```

Terminal

```
    Image:          nginx
    Image ID:       docker-pullable://nginx@sha256:644a70516a26004c97d0d85c7fe1d0c3a
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Wed, 03 Nov 2021 19:23:44 +0000
    Ready:          True
    Restart Count:  0
    Environment Variables from:
      webapp-config-map  ConfigMap  Optional: false
    Environment:        <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-zwhc6 (ro)
Conditions:
  Type              Status
```

9- How many Secrets exist on the system?

```
controlplane $ kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-zwhc6   kubernetes.io/service-account-token   3      80m
controlplane $
```

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

```
controlplane $ kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-zwhc6   kubernetes.io/service-account-token   3      80m
controlplane $ kubectl describe secret default-token-zwhc6
Name:         default-token-zwhc6
Namespace:    default
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: default
              kubernetes.io/service-account.uid: a32bbcdd-3cd0-11ec-88df-0242ac11002a

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1025 bytes
namespace:  7 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50
C9uYW1lc3BhY2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6Ir
zLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC5uYW1lIjoiZGVmYXVsdCIsImt1YmVybmV0ZXMuaW8v
nVpZCI6ImEzMmJiY2RkLTNjZDAtMTFlYy04OGRmLTAyNDJhYzExMDAyYSIsInN1YiI6InN5c3RlbTpzZXJ2aWNlY
hbZVPkkiygPepJbrT7c21NMgYYD1f_2PLzBYE7LesKNiCsFfs0I2Q1hQlgTV62yTWnxeJlWTMIXfGfzUxEBWmPdo_
tztO4QluRk84ukAH2Ffed6CVK3f5Ap00b7XWk0uulurn6YQocEhsagdl-odnwwdbjVzTAUFF_EPKTdgDQM2bPmdsa
SEZ7uWnUQwf4Hga8YW0z2xck8XB4LK2XktktHbtE0x0PMu_52gybJ-fde4IXWYwP0XDr4cbZlQ
controlplane $
```

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

```
controlplane $ kubectl get pods
NAME         READY   STATUS              RESTARTS   AGE
static-bb-pod   1/1     Running             0          16m
web-app      1/1     Running             0          8m2s
web-db       0/1     ContainerCreating   0          10s
```

12- why the db-pod status not ready

Secrets not found

13- 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

pod.yaml × web-pod.yaml × db-pod.yaml × db-secret.yaml ×

```
1  apiVersion: v1
2  kind: Secret
3  metadata:
4    name: db-secret
5  data:
6    MYSQL_DATABASE: c3FsMDE=
7    MYSQL_USER: dXN1cjE=
8    MYSQL_PASSWORD: cGFzc3dvcmQ=
9    MYSQL_ROOT_PASSWORD: cGFzc3dvcmQxMjM=
10
```

```
controlplane $ kubectl get secrets
NAME                  TYPE                                  DATA   AGE
db-secret             Opaque                                4      11s
default-token-zwhc6   kubernetes.io/service-account-token   3      98m
```

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

pod.yaml × web-pod.yaml × db-pod.yaml ×

```
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: web-db
5  spec:
6    containers:
7    - image: mysql:5.7
8      name: web-db
9      envFrom:
10       -  secretRef:
11            name: db-secret
12
```

```
controlplane $ kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
static-bb-pod 1/1     Running   1          32m
web-app       1/1     Running   0          23m
web-db        1/1     Running   0          3s
```

15- 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

multi-cont-pod.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: yellow
spec:
  containers:
  - image: busybox
    name: lemon

  - image: redis
    name: gold
```

```
controlplane $ kubectl get pod
NAME            READY   STATUS    RESTARTS   AGE
static-bb-pod   1/1     Running   2          36m
web-app         1/1     Running   0          28m
web-db          1/1     Running   0          4m36s
yellow          1/2     Running   0          13s
```

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

red-pod.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: red
spec:
  containers:
    - image: redis
      name: red-cont

  initContainers:
    - name: mycont
      image: busybox
      command: ["sleep 20"]
```

```
controlplane $ kubectl get pod
NAME            READY   STATUS                  RESTARTS   AGE
red             0/1     Init:CrashLoopBackOff   4          3m23s
```

17- Create a Persistent Volume with the given specification.
Volume Name: pv-log
Storage: 100Mi
Access Modes: ReadWriteMany
Host Path: /pv/log

pv-def.yaml

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

```
controlplane $ kubectl apply -f pv-def.yaml
persistentvolume/pv-log created
controlplane $ kubectl get pv
NAME     CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   REASON   AGE
pv-log   100Mi      RWX            Retain           Available                                   7s
```

18- Create a Persistent Volume Claim with the given specification.
Volume Name: claim-log-1
Storage Request: 50Mi
Access Modes: ReadWriteMany

pv-def.yaml pvc-def.yaml

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

```
controlplane $ kubectl apply -f pvc-def.yaml
persistentvolumeclaim/claim-log created
controlplane $ kubectl get pvc
NAME        STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE
claim-log   Bound    pv-log   100Mi      RWX                           4s
controlplane $
```

19- 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