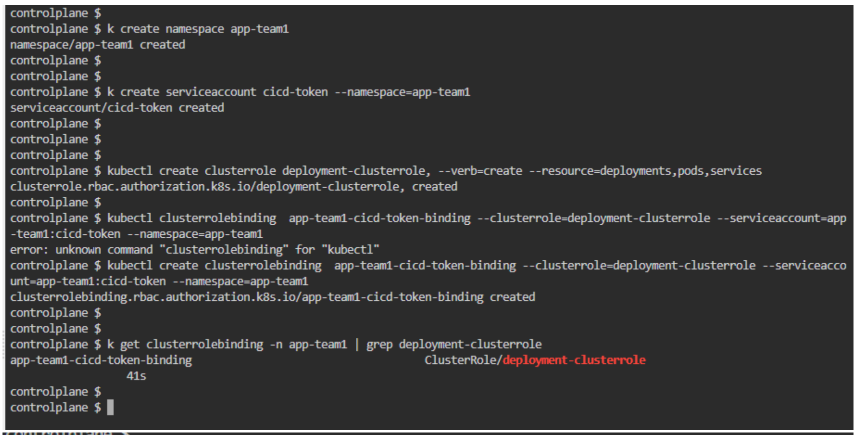
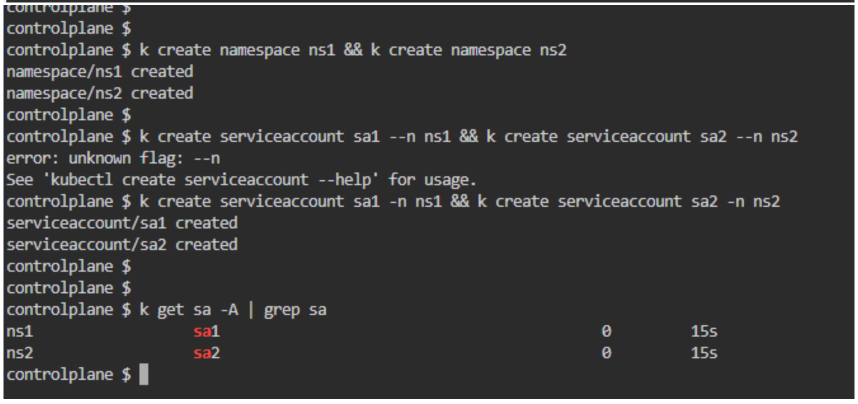
Mock Exam 3.0

Name: Mohamad Haffiz Bin Mohd Hissham
Date: 15 Sep 2024
---

1)

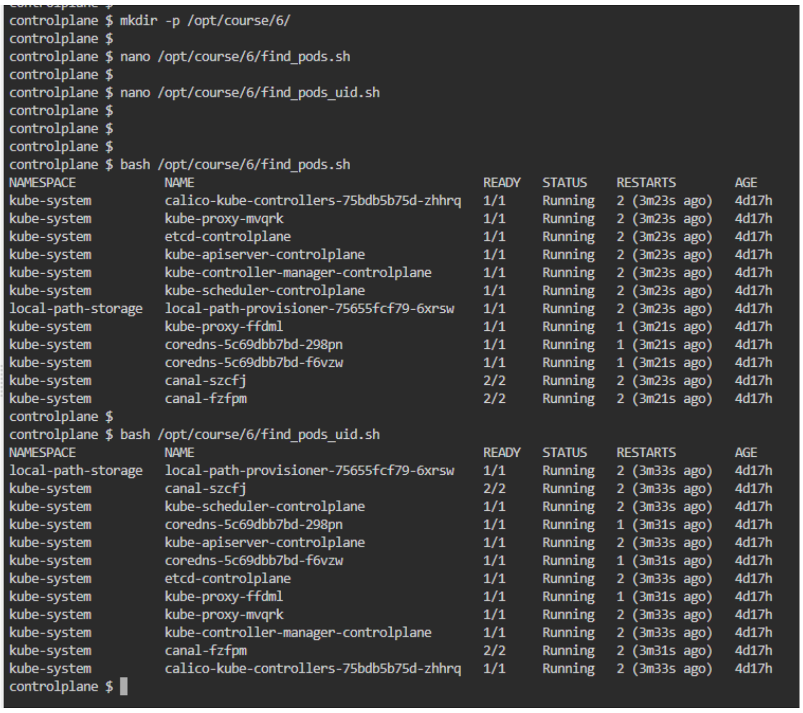


```
controlplane $
controlplane $ k create namespace app-team1
namespace/app-team1 created
controlplane $
controlplane $
controlplane $ k create serviceaccount cicd-token --namespace=app-team1
serviceaccount/cicd-token created
controlplane $
controlplane $
controlplane $
controlplane $ kubectl create clusterrole deployment-clusterrole, --verb=create --resource=deployments,pods,services
clusterrole.rbac.authorization.k8s.io/deployment-clusterrole, created
controlplane $
controlplane $ kubectl clusterrolebinding  app-team1-cicd-token-binding --clusterrole=deployment-clusterrole --serviceaccount=app
-team1:cicd-token --namespace=app-team1
error: unknown command "clusterrolebinding" for "kubectl"
controlplane $ kubectl create clusterrolebinding  app-team1-cicd-token-binding --clusterrole=deployment-clusterrole --serviceacco
unt=app-team1:cicd-token --namespace=app-team1
clusterrolebinding.rbac.authorization.k8s.io/app-team1-cicd-token-binding created
controlplane $
controlplane $
controlplane $ k get clusterrolebinding -n app-team1 | grep deployment-clusterrole
app-team1-cicd-token-binding                                     ClusterRole/deployment-clusterrole
                  41s
controlplane $
controlplane $
```



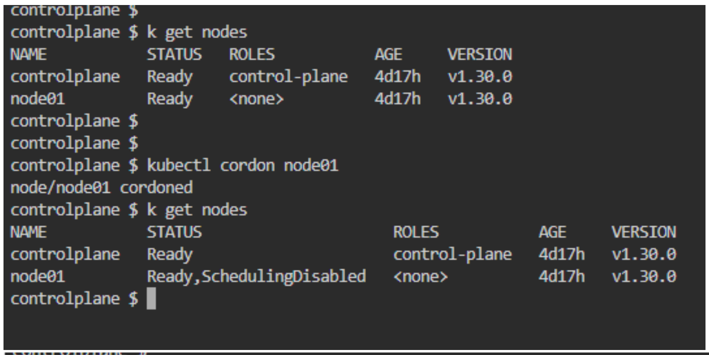
```
controlplane $
controlplane $
controlplane $ k create namespace ns1 && k create namespace ns2
namespace/ns1 created
namespace/ns2 created
controlplane $
controlplane $ k create serviceaccount sa1 --n ns1 && k create serviceaccount sa2 --n ns2
error: unknown flag: --n
See 'kubectl create serviceaccount --help' for usage.
controlplane $ k create serviceaccount sa1 -n ns1 && k create serviceaccount sa2 -n ns2
serviceaccount/sa1 created
serviceaccount/sa2 created
controlplane $
controlplane $
controlplane $ k get sa -A | grep sa
ns1                  sa1                                          0         15s
ns2                  sa2                                          0         15s
controlplane $
```

2)

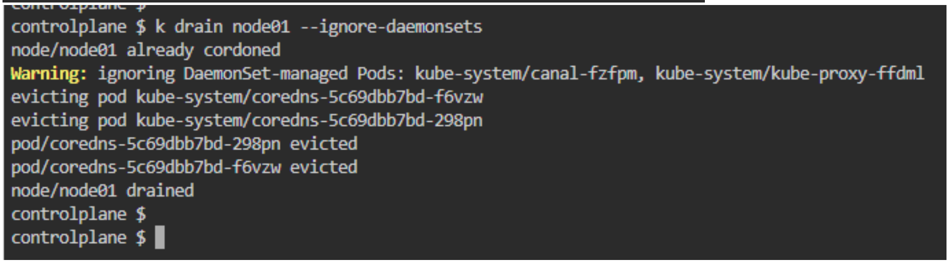


```
controlplane $ mkdir -p /opt/course/6/
controlplane $
controlplane $ nano /opt/course/6/find_pods.sh
controlplane $
controlplane $ nano /opt/course/6/find_pods_uid.sh
controlplane $
controlplane $
controlplane $
controlplane $ bash /opt/course/6/find_pods.sh
NAMESPACE            NAME                                       READY   STATUS    RESTARTS        AGE
kube-system          calico-kube-controllers-75bdb5b75d-zhhrq   1/1     Running   2 (3m23s ago)   4d17h
kube-system          kube-proxy-mvqrk                           1/1     Running   2 (3m23s ago)   4d17h
kube-system          etcd-controlplane                          1/1     Running   2 (3m23s ago)   4d17h
kube-system          kube-apiserver-controlplane                1/1     Running   2 (3m23s ago)   4d17h
kube-system          kube-controller-manager-controlplane       1/1     Running   2 (3m23s ago)   4d17h
kube-system          kube-scheduler-controlplane                1/1     Running   2 (3m23s ago)   4d17h
local-path-storage   local-path-provisioner-75655fcf79-6xrsw    1/1     Running   2 (3m23s ago)   4d17h
kube-system          kube-proxy-ffdml                           1/1     Running   1 (3m21s ago)   4d17h
kube-system          coredns-5c69dbb7bd-298pn                   1/1     Running   1 (3m21s ago)   4d17h
kube-system          coredns-5c69dbb7bd-f6vzw                   1/1     Running   1 (3m21s ago)   4d17h
kube-system          canal-szcfj                                2/2     Running   2 (3m23s ago)   4d17h
kube-system          canal-fzfpm                                2/2     Running   2 (3m21s ago)   4d17h
controlplane $
controlplane $ bash /opt/course/6/find_pods_uid.sh
NAMESPACE            NAME                                       READY   STATUS    RESTARTS        AGE
local-path-storage   local-path-provisioner-75655fcf79-6xrsw    1/1     Running   2 (3m33s ago)   4d17h
kube-system          canal-szcfj                                2/2     Running   2 (3m33s ago)   4d17h
kube-system          kube-scheduler-controlplane                1/1     Running   2 (3m33s ago)   4d17h
kube-system          coredns-5c69dbb7bd-298pn                   1/1     Running   1 (3m31s ago)   4d17h
kube-system          kube-apiserver-controlplane                1/1     Running   2 (3m33s ago)   4d17h
kube-system          coredns-5c69dbb7bd-f6vzw                   1/1     Running   1 (3m31s ago)   4d17h
kube-system          etcd-controlplane                          1/1     Running   2 (3m33s ago)   4d17h
kube-system          kube-proxy-ffdml                           1/1     Running   1 (3m31s ago)   4d17h
kube-system          kube-proxy-mvqrk                           1/1     Running   2 (3m33s ago)   4d17h
kube-system          kube-controller-manager-controlplane       1/1     Running   2 (3m33s ago)   4d17h
kube-system          canal-fzfpm                                2/2     Running   2 (3m31s ago)   4d17h
kube-system          calico-kube-controllers-75bdb5b75d-zhhrq   1/1     Running   2 (3m33s ago)   4d17h
controlplane $
```

3)

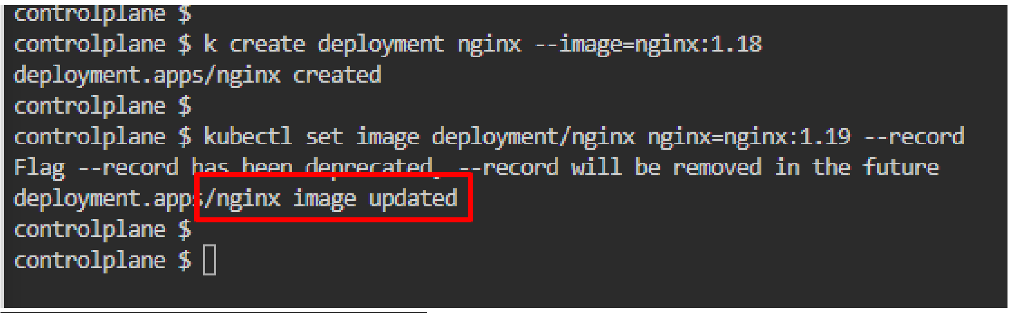


```
controlplane $
controlplane $ k get nodes
NAME           STATUS   ROLES           AGE     VERSION
controlplane   Ready    control-plane   4d17h   v1.30.0
node01         Ready    <none>          4d17h   v1.30.0
controlplane $
controlplane $
controlplane $ kubectl cordon node01
node/node01 cordoned
controlplane $ k get nodes
NAME           STATUS                     ROLES           AGE     VERSION
controlplane   Ready                      control-plane   4d17h   v1.30.0
node01         Ready,SchedulingDisabled   <none>          4d17h   v1.30.0
controlplane $
```

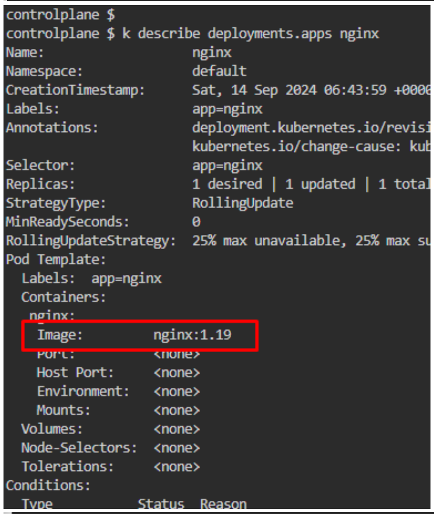


```
controlplane $
controlplane $ k drain node01 --ignore-daemonsets
node/node01 already cordoned
Warning: ignoring DaemonSet-managed Pods: kube-system/canal-fzfpm, kube-system/kube-proxy-ffdml
evicting pod kube-system/coredns-5c69dbb7bd-f6vzw
evicting pod kube-system/coredns-5c69dbb7bd-298pn
pod/coredns-5c69dbb7bd-298pn evicted
pod/coredns-5c69dbb7bd-f6vzw evicted
node/node01 drained
controlplane $
controlplane $
```

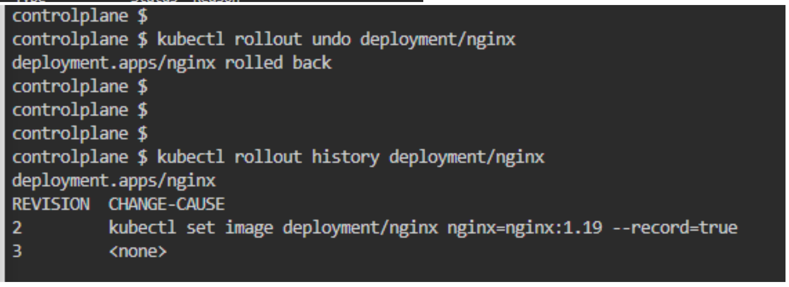
4)



```
controlplane $
controlplane $ k create deployment nginx --image=nginx:1.18
deployment.apps/nginx created
controlplane $
controlplane $ kubectl set image deployment/nginx nginx=nginx:1.19 --record
Flag --record has been deprecated, --record will be removed in the future
deployment.apps/nginx image updated
controlplane $
controlplane $
```

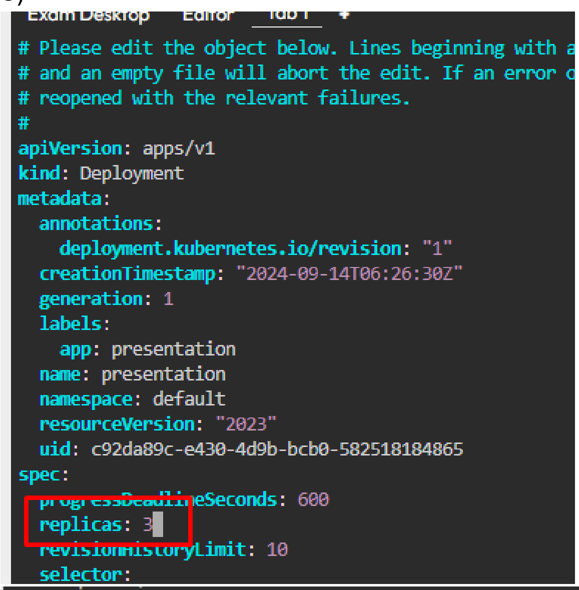


```
controlplane $
controlplane $ k describe deployments.apps nginx
Name:                   nginx
Namespace:              default
CreationTimestamp:      Sat, 14 Sep 2024 06:43:59 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revisi
                        kubernetes.io/change-cause: kub
Selector:               app=nginx
Replicas:               1 desired | 1 updated | 1 total
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max su
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.19
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
  Node-Selectors: <none>
  Tolerations:    <none>
Conditions:
  Type           Status  Reason
```

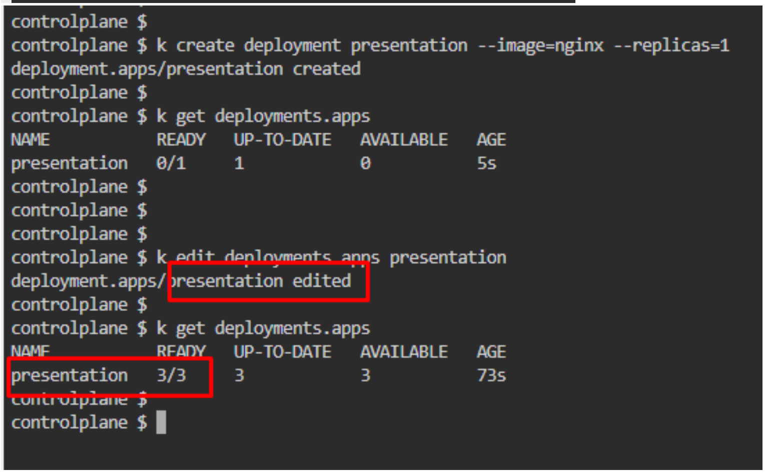


```
controlplane $
controlplane $ kubectl rollout undo deployment/nginx
deployment.apps/nginx rolled back
controlplane $
controlplane $
controlplane $
controlplane $ kubectl rollout history deployment/nginx
deployment.apps/nginx
REVISION  CHANGE-CAUSE
2         kubectl set image deployment/nginx nginx=nginx:1.19 --record=true
3         <none>
```

5)

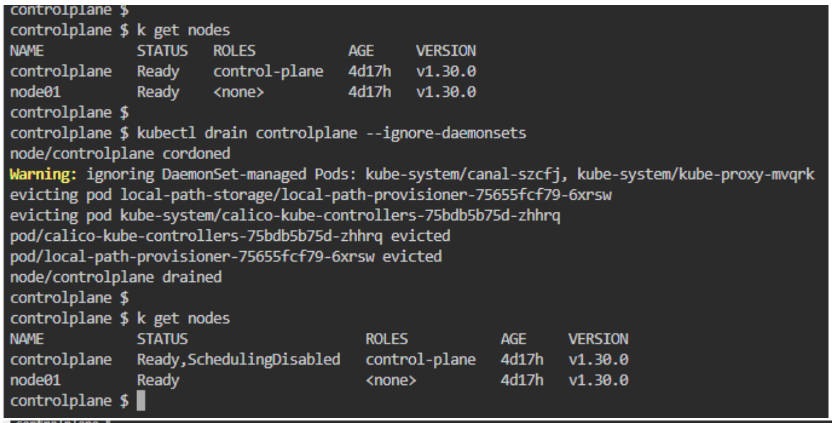


```
Exam Desktop    Editor    Tab 1    +
# Please edit the object below. Lines beginning with a
# and an empty file will abort the edit. If an error o
# reopened with the relevant failures.
#
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: "2024-09-14T06:26:30Z"
  generation: 1
  labels:
    app: presentation
  name: presentation
  namespace: default
  resourceVersion: "2023"
  uid: c92da89c-e430-4d9b-bcb0-582518184865
spec:
  progressDeadlineSeconds: 600
  replicas: 3
  revisionHistoryLimit: 10
  selector:
```

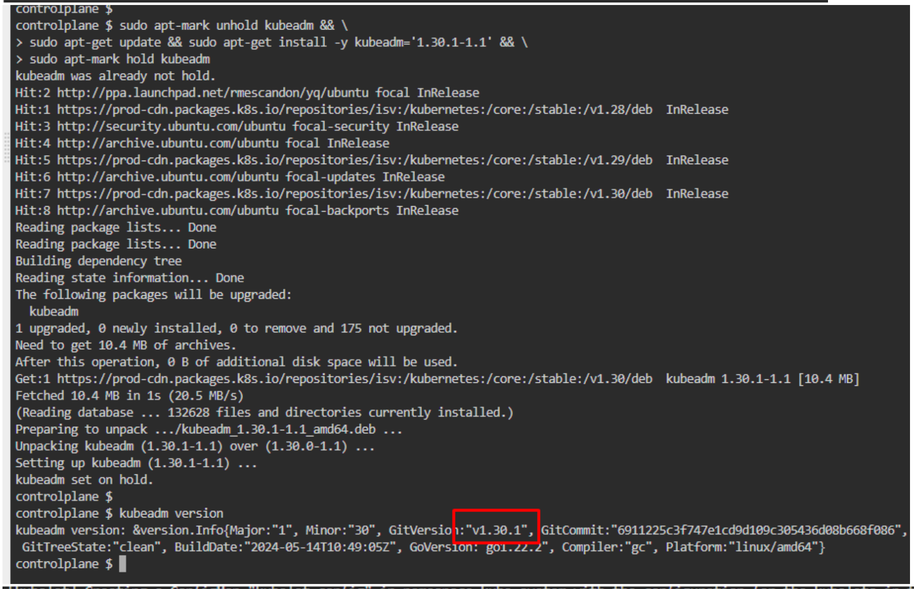


```
controlplane $
controlplane $ k create deployment presentation --image=nginx --replicas=1
deployment.apps/presentation created
controlplane $
controlplane $ k get deployments.apps
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
presentation   0/1     1            0           5s
controlplane $
controlplane $
controlplane $
controlplane $ k edit deployments.apps presentation
deployment.apps/presentation edited
controlplane $
controlplane $ k get deployments.apps
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
presentation   3/3     3            3           73s
controlplane $
controlplane $
```

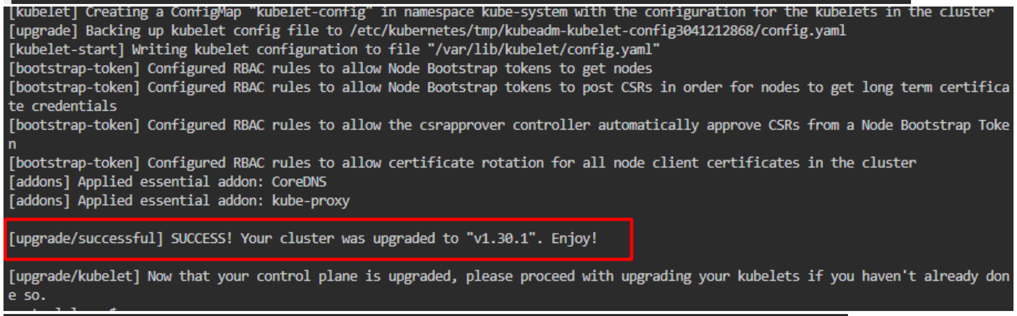
6)



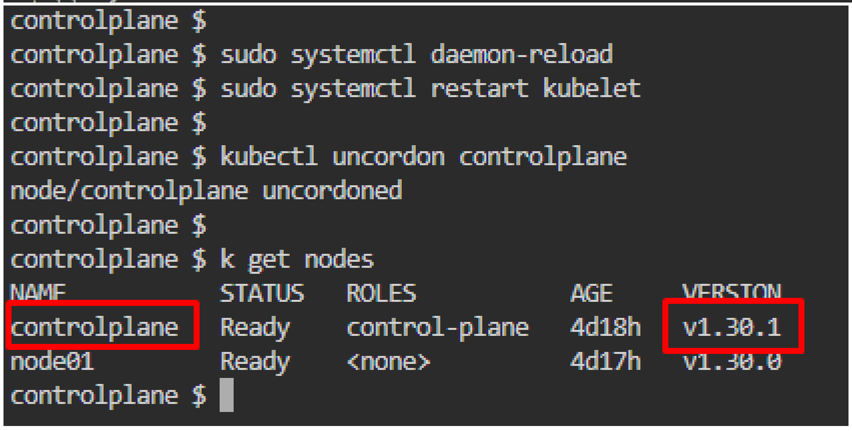
```
controlplane $
controlplane $ k get nodes
NAME           STATUS   ROLES           AGE     VERSION
controlplane   Ready    control-plane   4d17h   v1.30.0
node01         Ready    <none>          4d17h   v1.30.0
controlplane $
controlplane $ kubectl drain controlplane --ignore-daemonsets
node/controlplane cordoned
Warning: ignoring DaemonSet-managed Pods: kube-system/canal-szcfj, kube-system/kube-proxy-mvqrk
evicting pod local-path-storage/local-path-provisioner-75655fcf79-6xrsw
evicting pod kube-system/calico-kube-controllers-75bdb5b75d-zhhrq
pod/calico-kube-controllers-75bdb5b75d-zhhrq evicted
pod/local-path-provisioner-75655fcf79-6xrsw evicted
node/controlplane drained
controlplane $
controlplane $ k get nodes
NAME           STATUS                     ROLES           AGE     VERSION
controlplane   Ready,SchedulingDisabled   control-plane   4d17h   v1.30.0
node01         Ready                      <none>          4d17h   v1.30.0
controlplane $
```



```
controlplane $
controlplane $ sudo apt-mark unhold kubeadm && \
> sudo apt-get update && sudo apt-get install -y kubeadm='1.30.1-1.1' && \
> sudo apt-mark hold kubeadm
kubeadm was already not hold.
Hit:2 http://ppa.launchpad.net/rmescandon/yq/ubuntu focal InRelease
Hit:1 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.28/deb  InRelease
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.29/deb  InRelease
Hit:6 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  InRelease
Hit:8 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  kubeadm
1 upgraded, 0 newly installed, 0 to remove and 175 not upgraded.
Need to get 10.4 MB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  kubeadm 1.30.1-1.1 [10.4 MB]
Fetched 10.4 MB in 1s (20.5 MB/s)
(Reading database ... 132628 files and directories currently installed.)
Preparing to unpack .../kubeadm_1.30.1-1.1_amd64.deb ...
Unpacking kubeadm (1.30.1-1.1) over (1.30.0-1.1) ...
Setting up kubeadm (1.30.1-1.1) ...
kubeadm set on hold.
controlplane $
controlplane $ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"30", GitVersion:"v1.30.1", GitCommit:"6911225c3f747e1cd9d109c305436d08b668f086",
 GitTreeState:"clean", BuildDate:"2024-05-14T10:49:05Z", GoVersion:"go1.22.2", Compiler:"gc", Platform:"linux/amd64"}
controlplane $
```

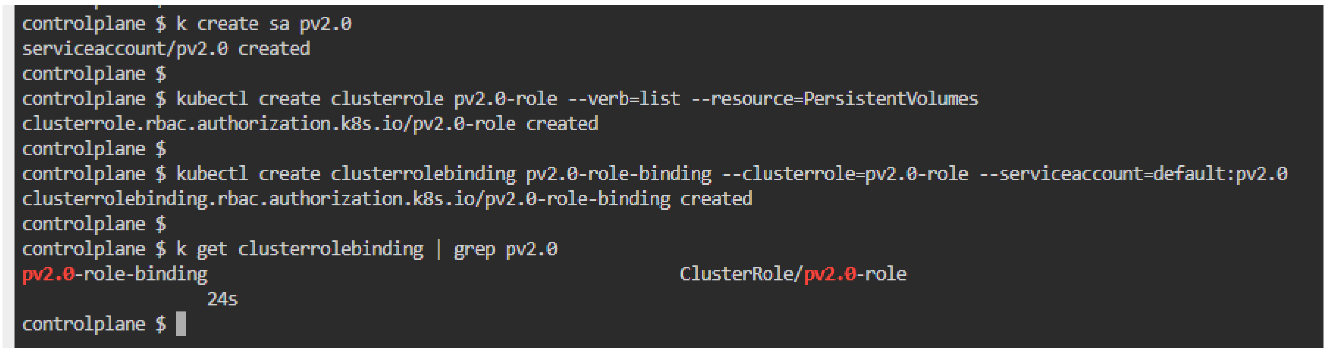


```
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upgrade] Backing up kubelet config file to /etc/kubernetes/tmp/kubeadm-kubelet-config3041212868/config.yaml
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certifica
te credentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Toke
n
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

[upgrade/successful] SUCCESS! Your cluster was upgraded to "v1.30.1". Enjoy!

[upgrade/kubelet] Now that your control plane is upgraded, please proceed with upgrading your kubelets if you haven't already don
e so.
```



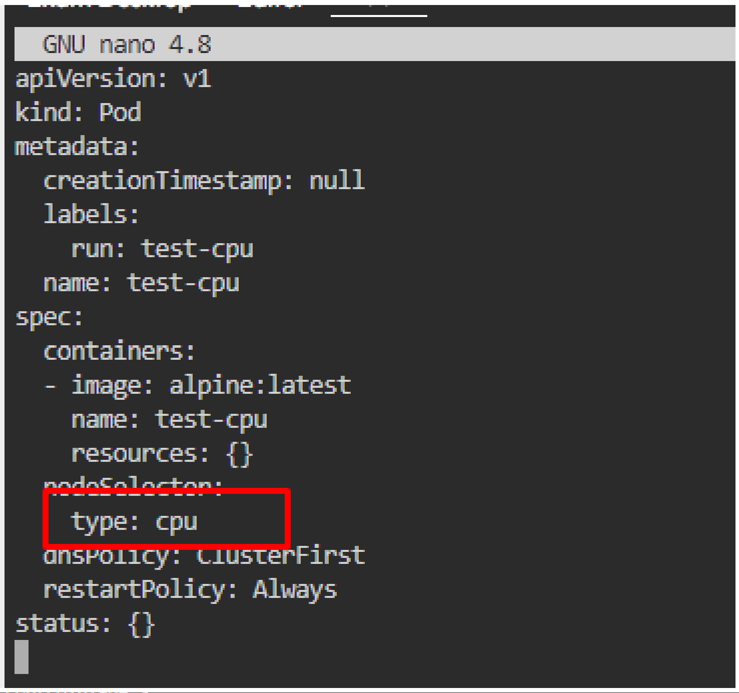
```
controlplane $
controlplane $ sudo systemctl daemon-reload
controlplane $ sudo systemctl restart kubelet
controlplane $
controlplane $ kubectl uncordon controlplane
node/controlplane uncordoned
controlplane $
controlplane $ k get nodes
NAME           STATUS   ROLES           AGE     VERSION
controlplane   Ready    control-plane   4d18h   v1.30.1
node01         Ready    <none>          4d17h   v1.30.0
controlplane $
```

7)

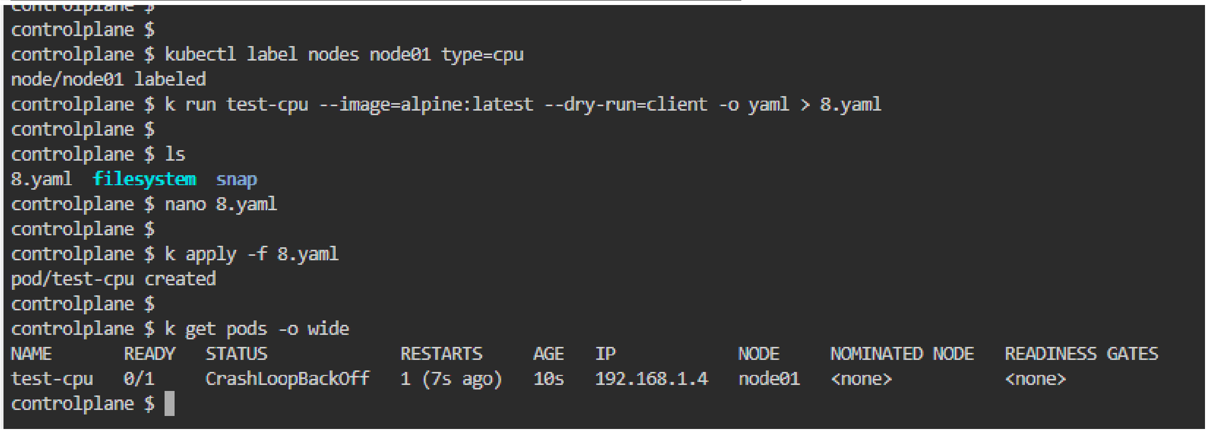


```
controlplane $ k create sa pv2.0
serviceaccount/pv2.0 created
controlplane $
controlplane $ kubectl create clusterrole pv2.0-role --verb=list --resource=PersistentVolumes
clusterrole.rbac.authorization.k8s.io/pv2.0-role created
controlplane $
controlplane $ kubectl create clusterrolebinding pv2.0-role-binding --clusterrole=pv2.0-role --serviceaccount=default:pv2.0
clusterrolebinding.rbac.authorization.k8s.io/pv2.0-role-binding created
controlplane $
controlplane $ k get clusterrolebinding | grep pv2.0
pv2.0-role-binding                                ClusterRole/pv2.0-role
                  24s
controlplane $
```

8)

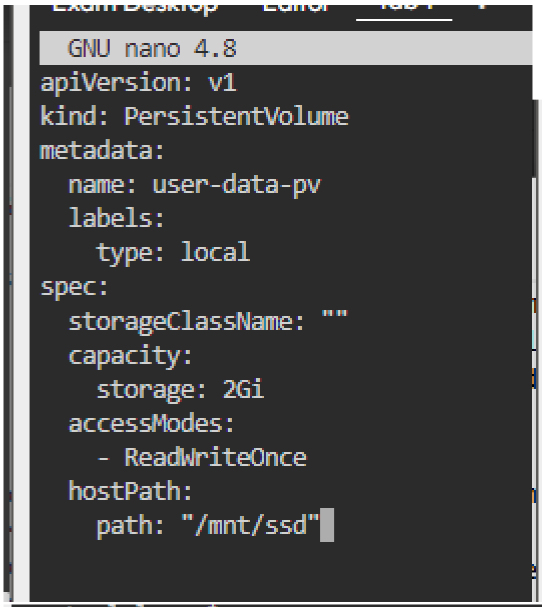


```
  GNU nano 4.8
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: test-cpu
  name: test-cpu
spec:
  containers:
  - image: alpine:latest
    name: test-cpu
    resources: {}
  nodeSelector:
    type: cpu
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
```

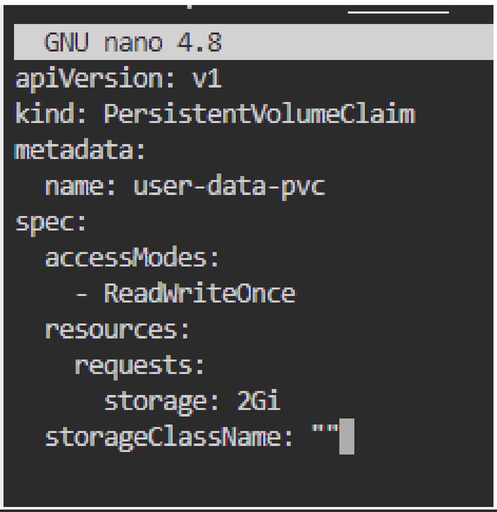


```
controlplane $
controlplane $ kubectl label nodes node01 type=cpu
node/node01 labeled
controlplane $ k run test-cpu --image=alpine:latest --dry-run=client -o yaml > 8.yaml
controlplane $
controlplane $ ls
8.yaml  filesystem  snap
controlplane $ nano 8.yaml
controlplane $
controlplane $ k apply -f 8.yaml
pod/test-cpu created
controlplane $
controlplane $ k get pods -o wide
NAME       READY   STATUS             RESTARTS     AGE   IP            NODE     NOMINATED NODE   READINESS GATES
test-cpu   0/1     CrashLoopBackOff   1 (7s ago)   10s   192.168.1.4   node01   <none>           <none>
controlplane $
```

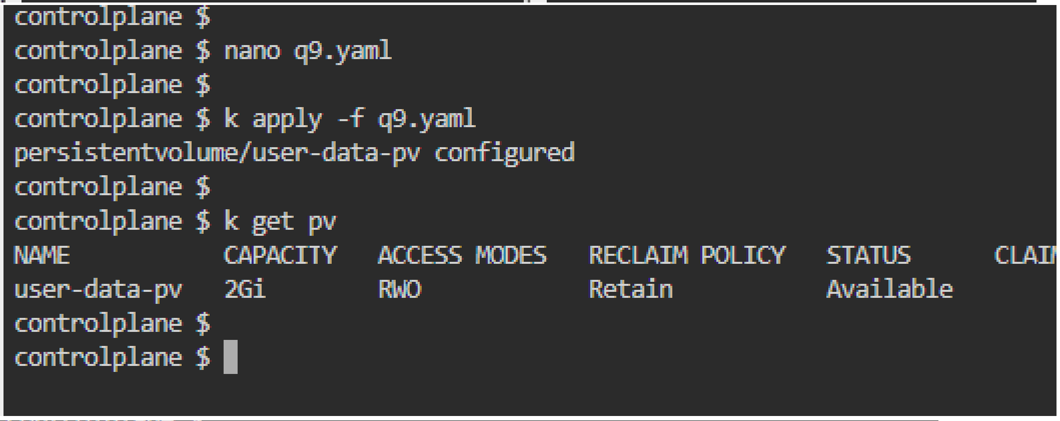
9)



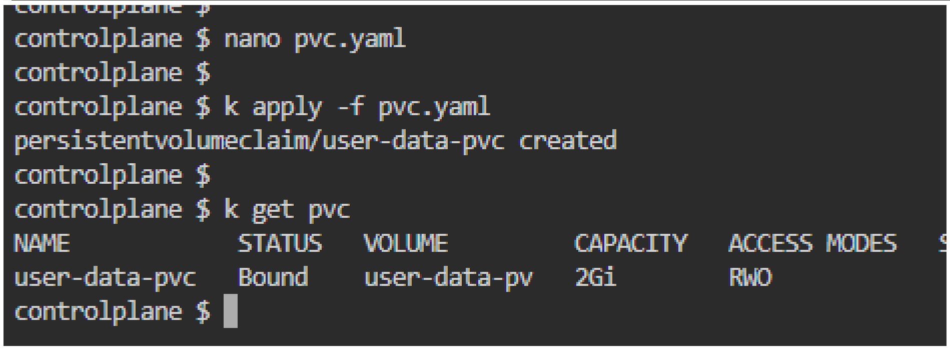
```
 GNU nano 4.8
apiVersion: v1
kind: PersistentVolume
metadata:
  name: user-data-pv
  labels:
    type: local
spec:
  storageClassName: ""
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/mnt/ssd"
```



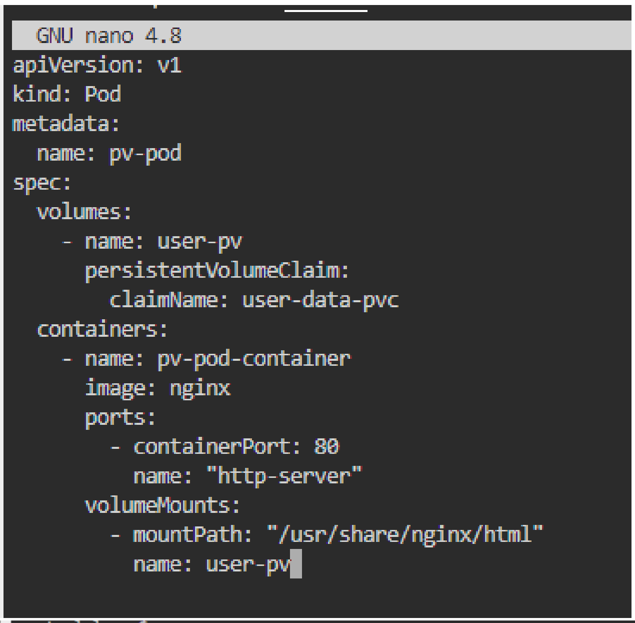
```
 GNU nano 4.8
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: user-data-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
  storageClassName: ""
```



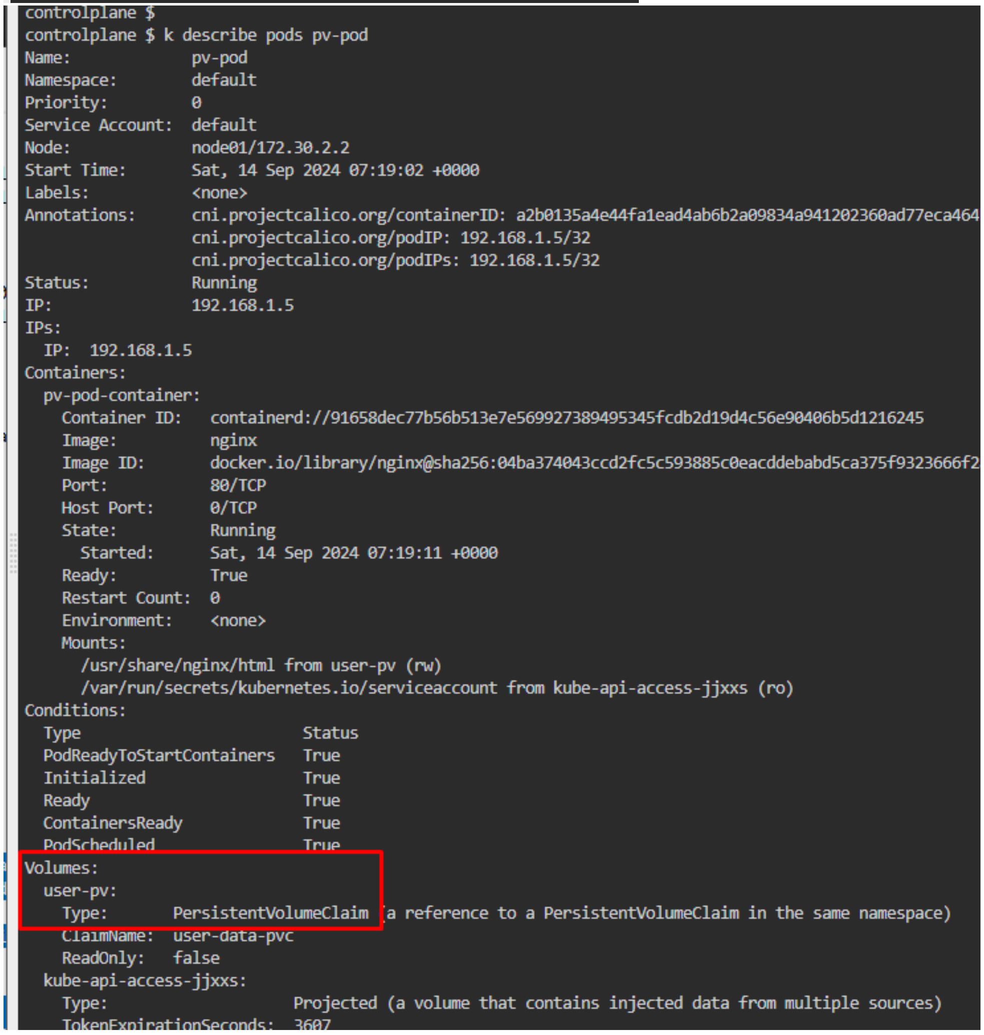
```
controlplane $
controlplane $ nano q9.yaml
controlplane $
controlplane $ k apply -f q9.yaml
persistentvolume/user-data-pv configured
controlplane $
controlplane $ k get pv
NAME           CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAI
user-data-pv   2Gi        RWO            Retain           Available
controlplane $
controlplane $
```



```
controlplane $ nano pvc.yaml
controlplane $
controlplane $ k apply -f pvc.yaml
persistentvolumeclaim/user-data-pvc created
controlplane $
controlplane $ k get pvc
NAME            STATUS   VOLUME         CAPACITY   ACCESS MODES   S
user-data-pvc   Bound    user-data-pv   2Gi        RWO
controlplane $
```

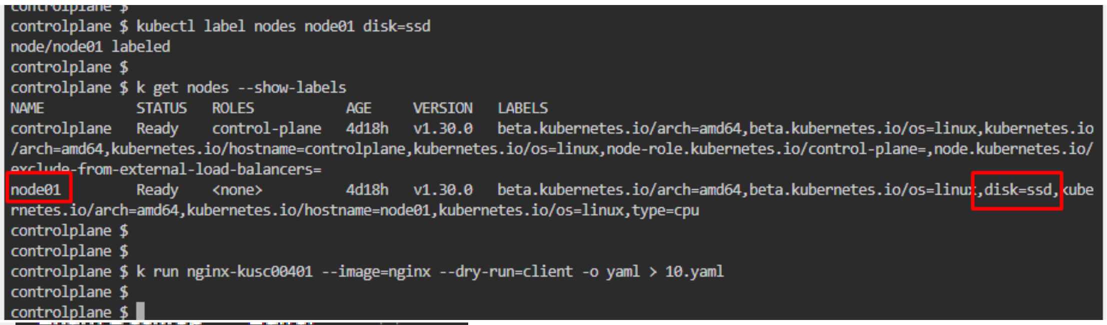


```
  GNU nano 4.8
apiVersion: v1
kind: Pod
metadata:
  name: pv-pod
spec:
  volumes:
    - name: user-pv
      persistentVolumeClaim:
        claimName: user-data-pvc
  containers:
    - name: pv-pod-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: user-pv
```

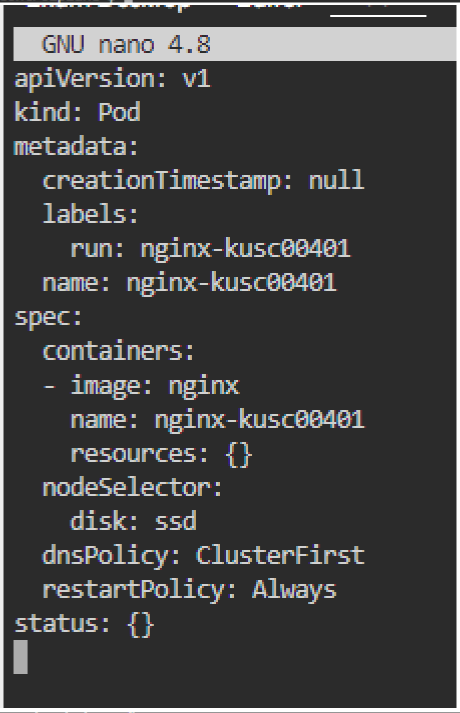


```
controlplane $
controlplane $ k describe pods pv-pod
Name:             pv-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             node01/172.30.2.2
Start Time:       Sat, 14 Sep 2024 07:19:02 +0000
Labels:           <none>
Annotations:      cni.projectcalico.org/containerID: a2b0135a4e44fa1ead4ab6b2a09834a941202360ad77eca464
                  cni.projectcalico.org/podIP: 192.168.1.5/32
                  cni.projectcalico.org/podIPs: 192.168.1.5/32
Status:           Running
IP:               192.168.1.5
IPs:
  IP:  192.168.1.5
Containers:
  pv-pod-container:
    Container ID:   containerd://91658dec77b56b513e7e569927389495345fcdb2d19d4c56e90406b5d1216245
    Image:          nginx
    Image ID:       docker.io/library/nginx@sha256:04ba374043ccd2fc5c593885c0eacddebabd5ca375f9323666f2
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sat, 14 Sep 2024 07:19:11 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /usr/share/nginx/html from user-pv (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-jjxxs (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  user-pv:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  user-data-pvc
    ReadOnly:   false
  kube-api-access-jjxxs:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
```

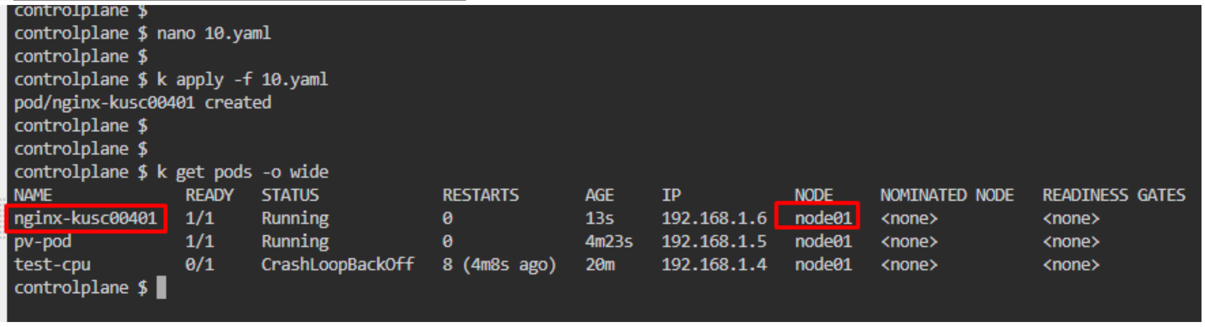
10)



```
controlplane $ kubectl label nodes node01 disk=ssd
node/node01 labeled
controlplane $
controlplane $ k get nodes --show-labels
NAME           STATUS   ROLES           AGE     VERSION   LABELS
controlplane   Ready    control-plane   4d18h   v1.30.0   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io
/arch=amd64,kubernetes.io/hostname=controlplane,kubernetes.io/os=linux,node-role.kubernetes.io/control-plane=,node.kubernetes.io/
exclude-from-external-load-balancers=
node01         Ready    <none>          4d18h   v1.30.0   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,disk=ssd,kube
rnetes.io/arch=amd64,kubernetes.io/hostname=node01,kubernetes.io/os=linux,type=cpu
controlplane $
controlplane $
controlplane $ k run nginx-kusc00401 --image=nginx --dry-run=client -o yaml > 10.yaml
controlplane $
controlplane $
```



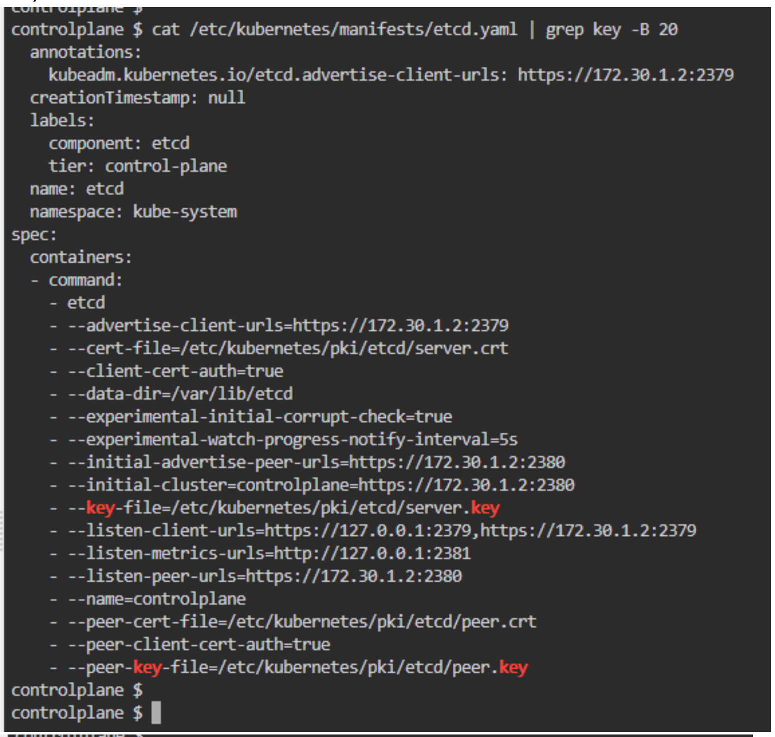
```
  GNU nano 4.8
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-kusc00401
  name: nginx-kusc00401
spec:
  containers:
  - image: nginx
    name: nginx-kusc00401
    resources: {}
  nodeSelector:
    disk: ssd
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
```



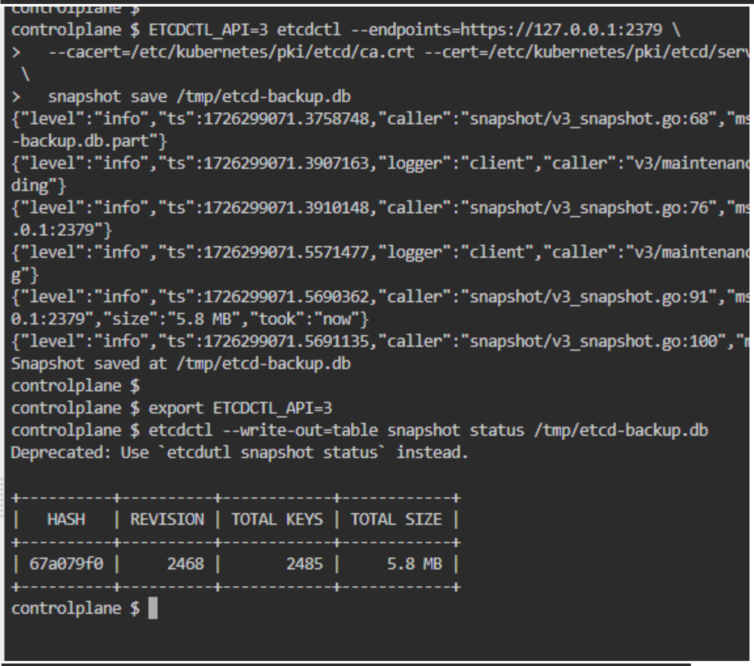
```
controlplane $ nano 10.yaml
controlplane $
controlplane $ k apply -f 10.yaml
pod/nginx-kusc00401 created
controlplane $
controlplane $
controlplane $ k get pods -o wide
NAME              READY   STATUS             RESTARTS       AGE     IP            NODE     NOMINATED NODE   READINESS GATES
nginx-kusc00401   1/1     Running            0              13s     192.168.1.6   node01   <none>           <none>
pv-pod            1/1     Running            0              4m23s   192.168.1.5   node01   <none>           <none>
test-cpu          0/1     CrashLoopBackOff   8 (4m8s ago)   20m     192.168.1.4   node01   <none>           <none>
controlplane $
```

11) No question

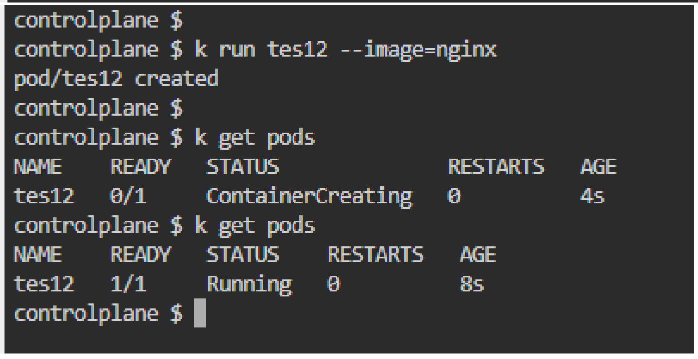
12)



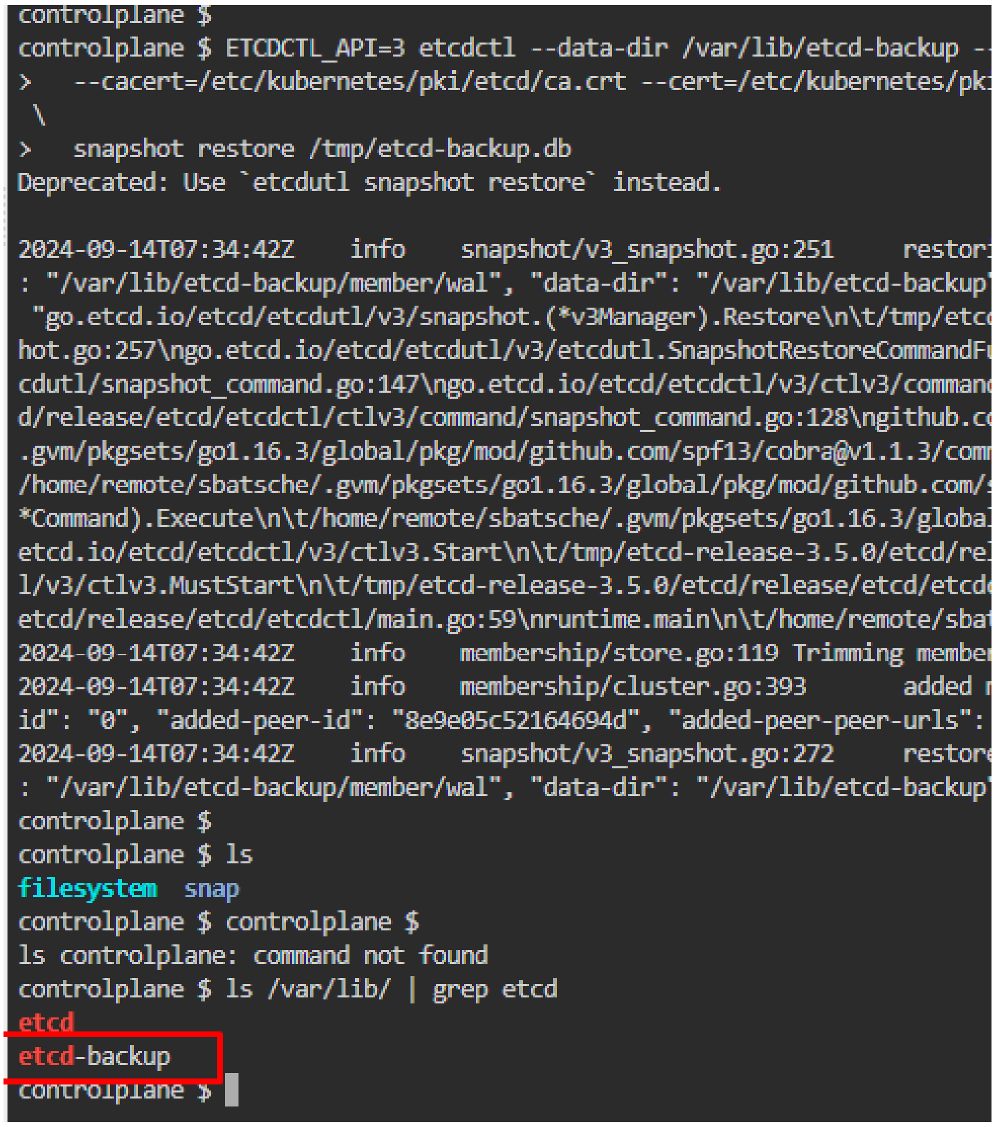
```
controlplane $
controlplane $ cat /etc/kubernetes/manifests/etcd.yaml | grep key -B 20
  annotations:
    kubeadm.kubernetes.io/etcd.advertise-client-urls: https://172.30.1.2:2379
  creationTimestamp: null
  labels:
    component: etcd
    tier: control-plane
  name: etcd
  namespace: kube-system
spec:
  containers:
  - command:
    - etcd
    - --advertise-client-urls=https://172.30.1.2:2379
    - --cert-file=/etc/kubernetes/pki/etcd/server.crt
    - --client-cert-auth=true
    - --data-dir=/var/lib/etcd
    - --experimental-initial-corrupt-check=true
    - --experimental-watch-progress-notify-interval=5s
    - --initial-advertise-peer-urls=https://172.30.1.2:2380
    - --initial-cluster=controlplane=https://172.30.1.2:2380
    - --key-file=/etc/kubernetes/pki/etcd/server.key
    - --listen-client-urls=https://127.0.0.1:2379,https://172.30.1.2:2379
    - --listen-metrics-urls=http://127.0.0.1:2381
    - --listen-peer-urls=https://172.30.1.2:2380
    - --name=controlplane
    - --peer-cert-file=/etc/kubernetes/pki/etcd/peer.crt
    - --peer-client-cert-auth=true
    - --peer-key-file=/etc/kubernetes/pki/etcd/peer.key
controlplane $
controlplane $
```



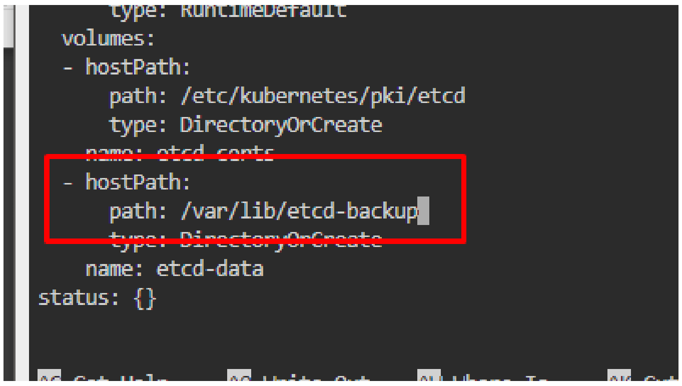
```
controlplane $
controlplane $ ETCDCTL_API=3 etcdctl --endpoints=https://127.0.0.1:2379 \
>   --cacert=/etc/kubernetes/pki/etcd/ca.crt --cert=/etc/kubernetes/pki/etcd/serv
 \
>   snapshot save /tmp/etcd-backup.db
{"level":"info","ts":1726299071.3758748,"caller":"snapshot/v3_snapshot.go:68","ms
-backup.db.part"}
{"level":"info","ts":1726299071.3907163,"logger":"client","caller":"v3/maintenanc
ding"}
{"level":"info","ts":1726299071.3910148,"caller":"snapshot/v3_snapshot.go:76","ms
.0.1:2379"}
{"level":"info","ts":1726299071.5571477,"logger":"client","caller":"v3/maintenanc
g"}
{"level":"info","ts":1726299071.5690362,"caller":"snapshot/v3_snapshot.go:91","ms
0.1:2379","size":"5.8 MB","took":"now"}
{"level":"info","ts":1726299071.5691135,"caller":"snapshot/v3_snapshot.go:100","r
Snapshot saved at /tmp/etcd-backup.db
controlplane $
controlplane $ export ETCDCTL_API=3
controlplane $ etcdctl --write-out=table snapshot status /tmp/etcd-backup.db
Deprecated: Use `etcdutl snapshot status` instead.

+----------+----------+------------+------------+
|   HASH   | REVISION | TOTAL KEYS | TOTAL SIZE |
+----------+----------+------------+------------+
| 67a079f0 |     2468 |       2485 |     5.8 MB |
+----------+----------+------------+------------+
controlplane $
```



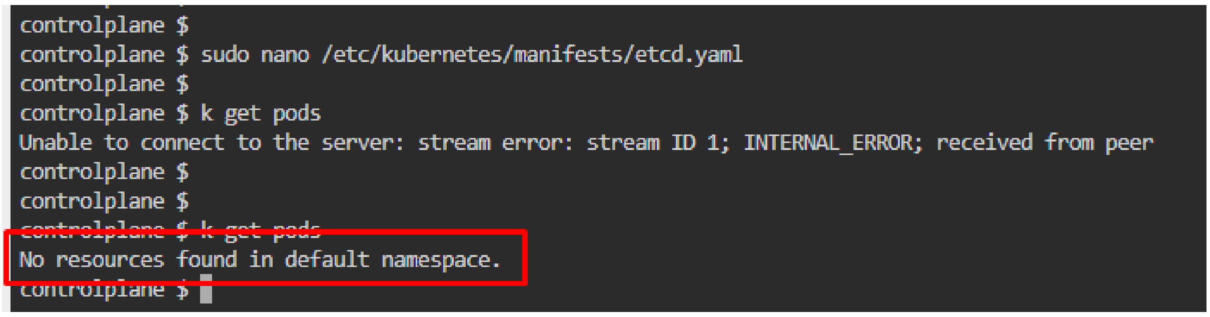
```
controlplane $
controlplane $ k run tes12 --image=nginx
pod/tes12 created
controlplane $
controlplane $ k get pods
NAME    READY   STATUS              RESTARTS   AGE
tes12   0/1     ContainerCreating   0          4s
controlplane $ k get pods
NAME    READY   STATUS    RESTARTS   AGE
tes12   1/1     Running   0          8s
controlplane $
```



```
controlplane $
controlplane $ ETCDCTL_API=3 etcdctl --data-dir /var/lib/etcd-backup --
>   --cacert=/etc/kubernetes/pki/etcd/ca.crt --cert=/etc/kubernetes/pki
 \
>   snapshot restore /tmp/etcd-backup.db
Deprecated: Use `etcdutl snapshot restore` instead.

2024-09-14T07:34:42Z    info    snapshot/v3_snapshot.go:251     restor
: "/var/lib/etcd-backup/member/wal", "data-dir": "/var/lib/etcd-backup
 "go.etcd.io/etcd/etcdutl/v3/snapshot.(*v3Manager).Restore\n\t/tmp/etc
hot.go:257\ngo.etcd.io/etcd/etcdutl/v3/etcdutl.SnapshotRestoreCommandFu
cdutl/snapshot_command.go:147\ngo.etcd.io/etcd/etcdctl/v3/ctlv3/comman
d/release/etcd/etcdctl/ctlv3/command/snapshot_command.go:128\ngithub.co
.gvm/pkgsets/go1.16.3/global/pkg/mod/github.com/spf13/cobra@v1.1.3/comm
/home/remote/sbatsche/.gvm/pkgsets/go1.16.3/global/pkg/mod/github.com/s
*Command).Execute\n\t/home/remote/sbatsche/.gvm/pkgsets/go1.16.3/global
etcd.io/etcd/etcdctl/v3/ctlv3.Start\n\t/tmp/etcd-release-3.5.0/etcd/rel
l/v3/ctlv3.MustStart\n\t/tmp/etcd-release-3.5.0/etcd/release/etcd/etcdc
etcd/release/etcd/etcdctl/main.go:59\nruntime.main\n\t/home/remote/sbat
2024-09-14T07:34:42Z    info    membership/store.go:119 Trimming member
2024-09-14T07:34:42Z    info    membership/cluster.go:393       added m
id": "0", "added-peer-id": "8e9e05c52164694d", "added-peer-peer-urls":
2024-09-14T07:34:42Z    info    snapshot/v3_snapshot.go:272     restore
: "/var/lib/etcd-backup/member/wal", "data-dir": "/var/lib/etcd-backup
controlplane $
controlplane $ ls
filesystem  snap
controlplane $ controlplane $
ls controlplane: command not found
controlplane $ ls /var/lib/ | grep etcd
etcd
etcd-backup
controlplane $
```

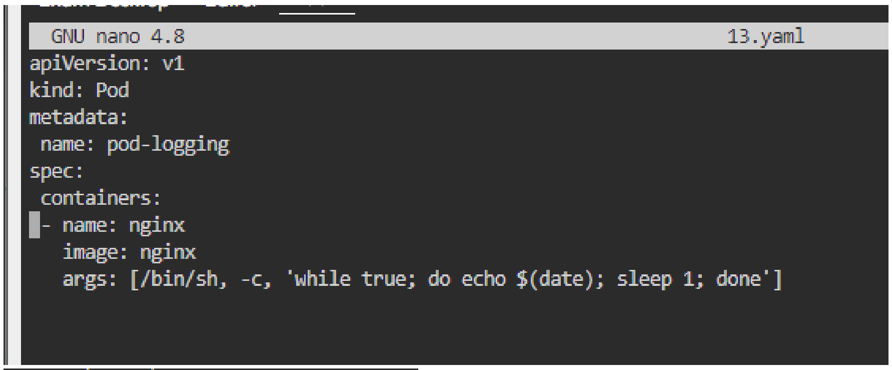


```
        type: RuntimeDefault
  volumes:
  - hostPath:
      path: /etc/kubernetes/pki/etcd
      type: DirectoryOrCreate
    name: etcd-certs
  - hostPath:
      path: /var/lib/etcd-backup
      type: DirectoryOrCreate
    name: etcd-data
status: {}
```

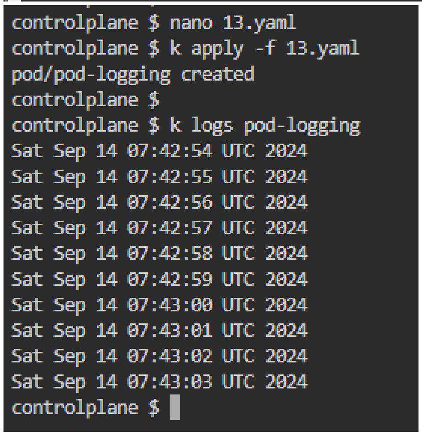


```
controlplane $
controlplane $ sudo nano /etc/kubernetes/manifests/etcd.yaml
controlplane $
controlplane $ k get pods
Unable to connect to the server: stream error: stream ID 1; INTERNAL_ERROR; received from peer
controlplane $
controlplane $
controlplane $ k get pods
No resources found in default namespace.
controlplane $
```

13)

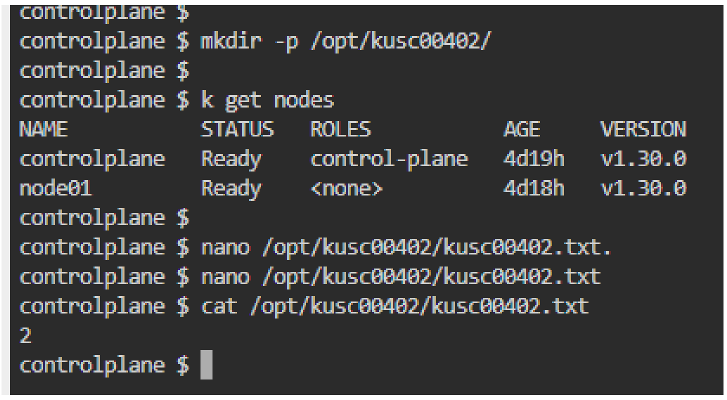


```
  GNU nano 4.8                                          13.yaml
apiVersion: v1
kind: Pod
metadata:
 name: pod-logging
spec:
 containers:
 - name: nginx
   image: nginx
   args: [/bin/sh, -c, 'while true; do echo $(date); sleep 1; done']
```



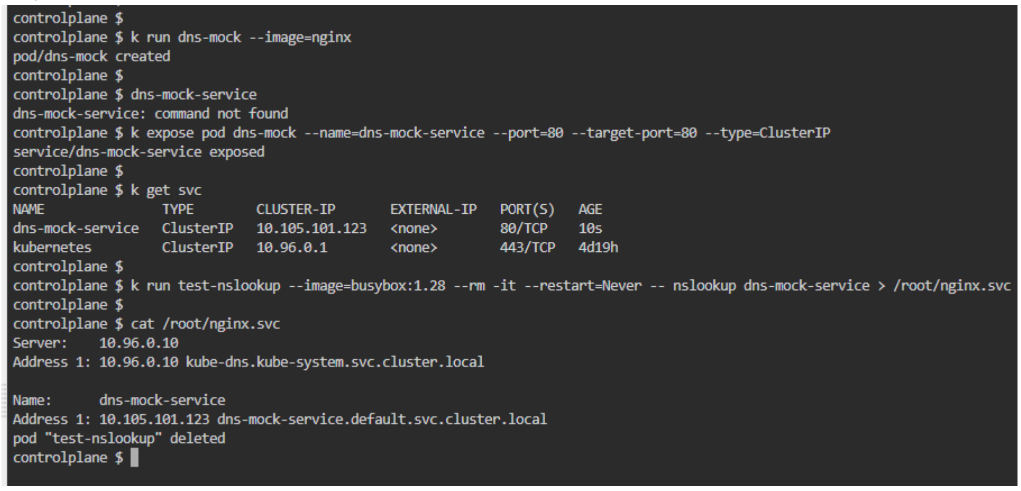
```
controlplane $ nano 13.yaml
controlplane $ k apply -f 13.yaml
pod/pod-logging created
controlplane $
controlplane $ k logs pod-logging
Sat Sep 14 07:42:54 UTC 2024
Sat Sep 14 07:42:55 UTC 2024
Sat Sep 14 07:42:56 UTC 2024
Sat Sep 14 07:42:57 UTC 2024
Sat Sep 14 07:42:58 UTC 2024
Sat Sep 14 07:42:59 UTC 2024
Sat Sep 14 07:43:00 UTC 2024
Sat Sep 14 07:43:01 UTC 2024
Sat Sep 14 07:43:02 UTC 2024
Sat Sep 14 07:43:03 UTC 2024
controlplane $
```

14)



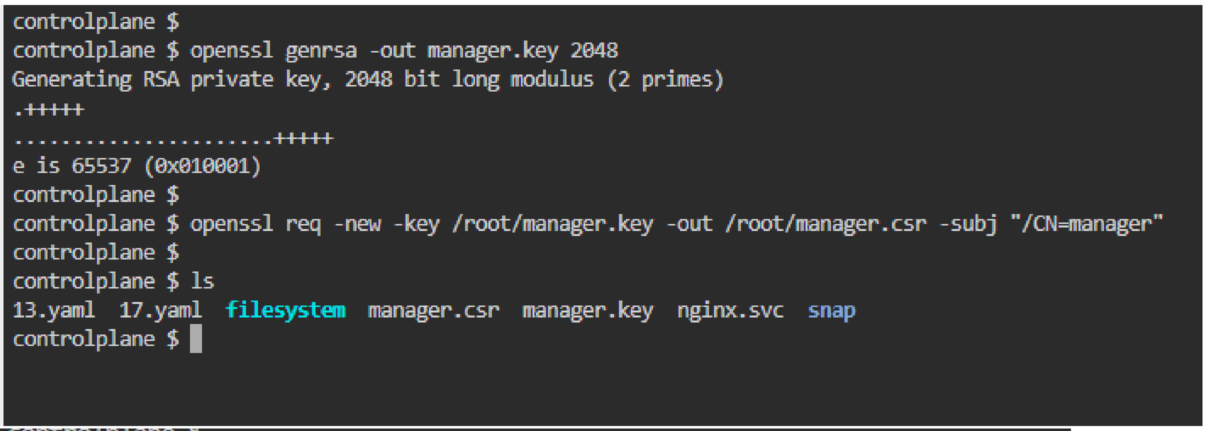
```
controlplane $
controlplane $ mkdir -p /opt/kusc00402/
controlplane $
controlplane $ k get nodes
NAME           STATUS   ROLES           AGE     VERSION
controlplane   Ready    control-plane   4d19h   v1.30.0
node01         Ready    <none>          4d18h   v1.30.0
controlplane $
controlplane $ nano /opt/kusc00402/kusc00402.txt.
controlplane $ nano /opt/kusc00402/kusc00402.txt
controlplane $ cat /opt/kusc00402/kusc00402.txt
2
controlplane $
```

15)

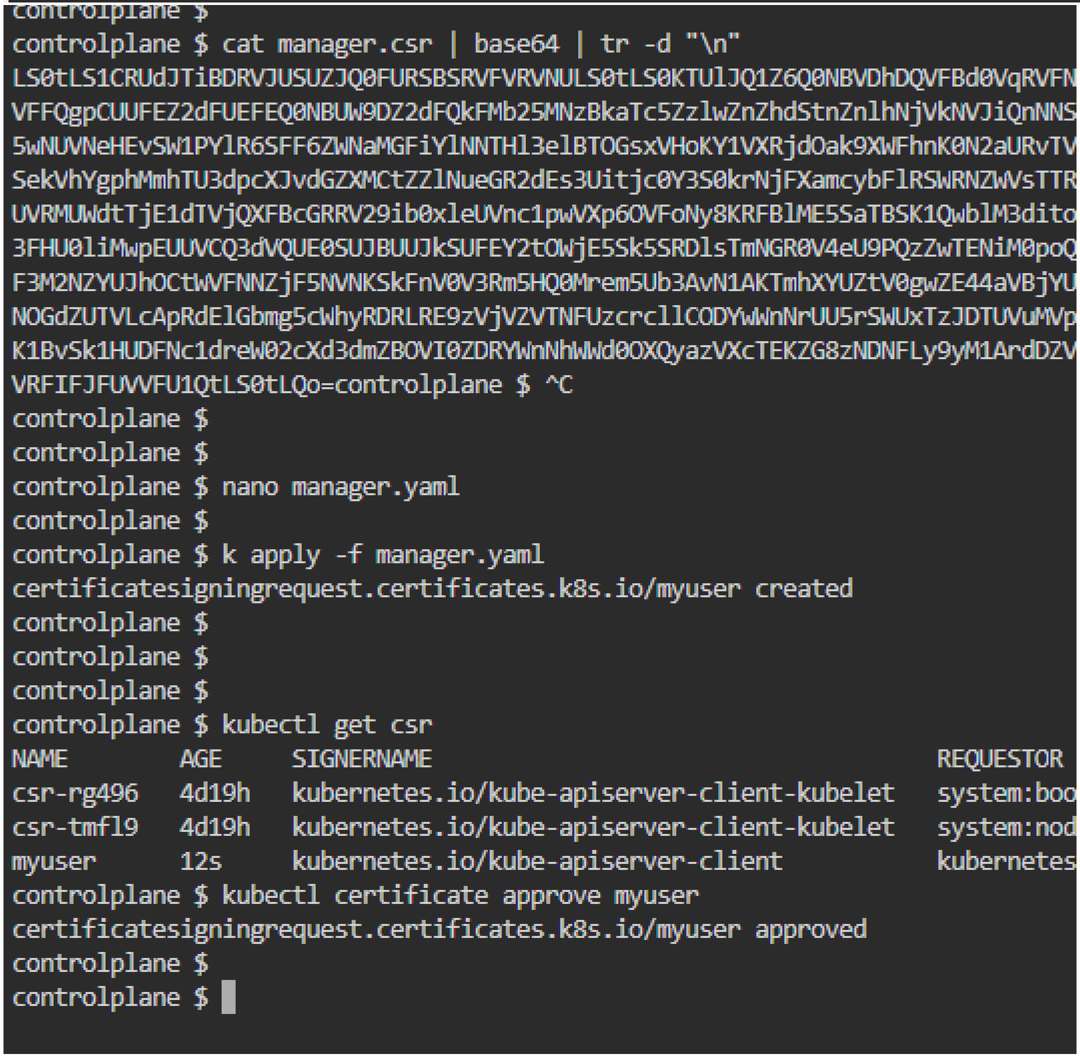


```
controlplane $
controlplane $ k run dns-mock --image=nginx
pod/dns-mock created
controlplane $
controlplane $ dns-mock-service
dns-mock-service: command not found
controlplane $ k expose pod dns-mock --name=dns-mock-service --port=80 --target-port=80 --type=ClusterIP
service/dns-mock-service exposed
controlplane $
controlplane $ k get svc
NAME               TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
dns-mock-service   ClusterIP   10.105.101.123   <none>        80/TCP    10s
kubernetes         ClusterIP   10.96.0.1        <none>        443/TCP   4d19h
controlplane $
controlplane $ k run test-nslookup --image=busybox:1.28 --rm -it --restart=Never -- nslookup dns-mock-service > /root/nginx.svc
controlplane $
controlplane $ cat /root/nginx.svc
Server:    10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name:      dns-mock-service
Address 1: 10.105.101.123 dns-mock-service.default.svc.cluster.local
pod "test-nslookup" deleted
controlplane $
```

16)



```
controlplane $
controlplane $ openssl genrsa -out manager.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.+++++
.......................+++++
e is 65537 (0x010001)
controlplane $
controlplane $ openssl req -new -key /root/manager.key -out /root/manager.csr -subj "/CN=manager"
controlplane $
controlplane $ ls
13.yaml  17.yaml  filesystem  manager.csr  manager.key  nginx.svc  snap
controlplane $
```

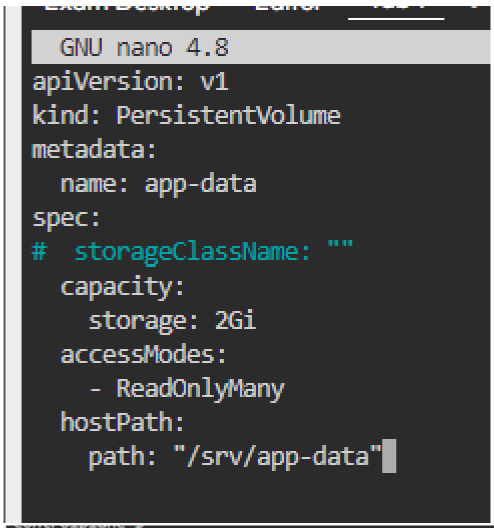


```
controlplane $
controlplane $ cat manager.csr | base64 | tr -d "\n"
```

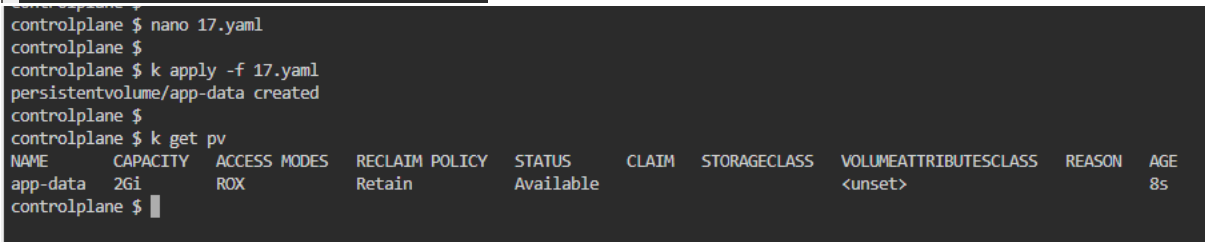
```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0KTUlJQ1Z6Q0NBVDhDQVFBd0VqRVFN
VFFQgpCUUFEZ2dFUEFEQ0NBUW9DZ2dFQkFMb25MNzBkaTc5ZzlwZnZhdStnZnlhNjVkNVJiQnNNS
5wNUVNeHEvSW1PYlR6SFF6ZWNaMGFiYlNNTHl3elBTOGsxVHoKY1VXRjdOak9XWFhnK0N2aURvTV
SekVhYgphMmhTU3dpcXJvdGZXMCtZZlNueGR2dEs3Uitjc0Y3S0krNjFXamcybFlRSWRNZWVsTTR
UVRMUWdtTjE1dTVjQXFBcGRRV29ib0xleUVnc1pwVXp6OVFoNy8KRFBlME5SaTBSK1QwblM3dito
3FHU0liMwpEUUVCQ3dVQUE0SUJBUUJkSUFEY2tOWjE5Sk5SRDlsTmNGR0V4eU9PQzZwTENiM0poQ
F3M2NZYUJhOCtWVFNNZjF5NVNKSkFnV0V3Rm5HQ0Mrem5Ub3AvN1AKTmhXYUZtV0gwZE44aVBjYU
NOGdZUTVLcApRdElGbmg5cWhyRDRLRE9zVjVZVTNFUzcrcllCODYwWnNrUU5rSWUxTzJDTUVuMVp
K1BvSk1HUDFNc1dreW02cXd3dmZBOVI0ZDRYWnNhWWd0OXQyazVXcTEKZG8zNDNFLy9yM1ArdDZV
VRFIFJFUVVFU1QtLS0tLQo=controlplane $ ^C
```

```
controlplane $
controlplane $
controlplane $ nano manager.yaml
controlplane $
controlplane $ k apply -f manager.yaml
certificatesigningrequest.certificates.k8s.io/myuser created
controlplane $
controlplane $
controlplane $
controlplane $ kubectl get csr
NAME        AGE     SIGNERNAME                                    REQUESTOR
csr-rg496   4d19h   kubernetes.io/kube-apiserver-client-kubelet   system:boo
csr-tmfl9   4d19h   kubernetes.io/kube-apiserver-client-kubelet   system:nod
myuser      12s     kubernetes.io/kube-apiserver-client           kubernetes
controlplane $ kubectl certificate approve myuser
certificatesigningrequest.certificates.k8s.io/myuser approved
controlplane $
controlplane $
```

17)



```
  GNU nano 4.8
apiVersion: v1
kind: PersistentVolume
metadata:
  name: app-data
spec:
#  storageClassName: ""
  capacity:
    storage: 2Gi
  accessModes:
    - ReadOnlyMany
  hostPath:
    path: "/srv/app-data"
```



```
controlplane $ nano 17.yaml
controlplane $
controlplane $ k apply -f 17.yaml
persistentvolume/app-data created
controlplane $
controlplane $ k get pv
NAME       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   VOLUMEATTRIBUTESCLASS   REASON   AGE
app-data   2Gi        ROX            Retain           Available                          <unset>                          8s
controlplane $
```