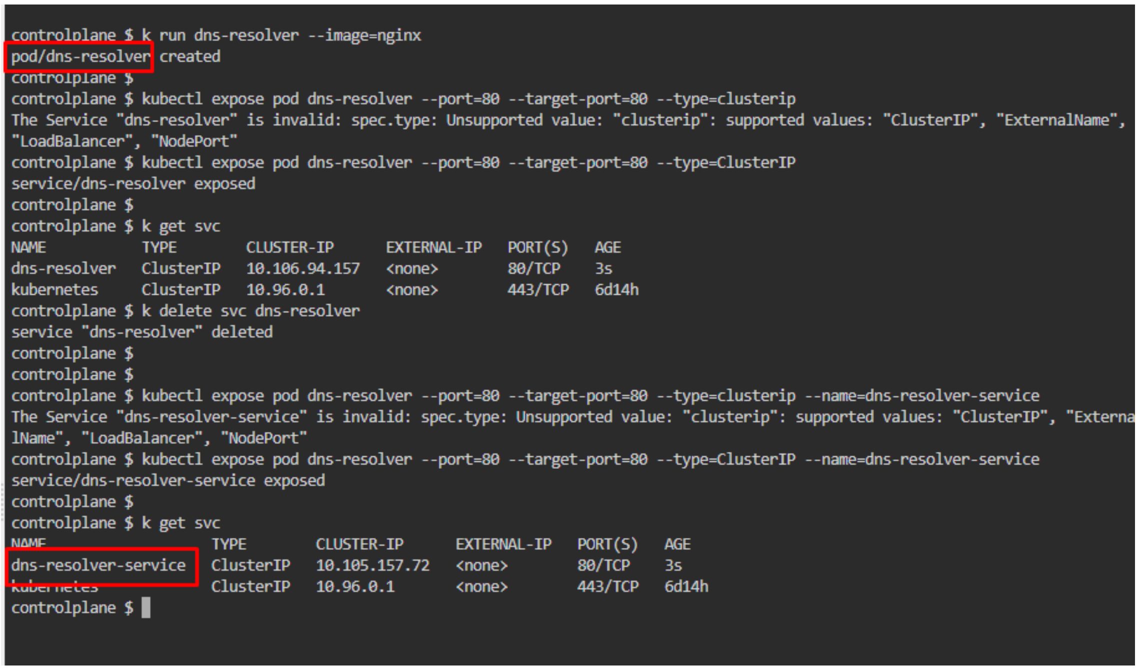
Mock Exam 4.0 – Revision JW C4 & C5

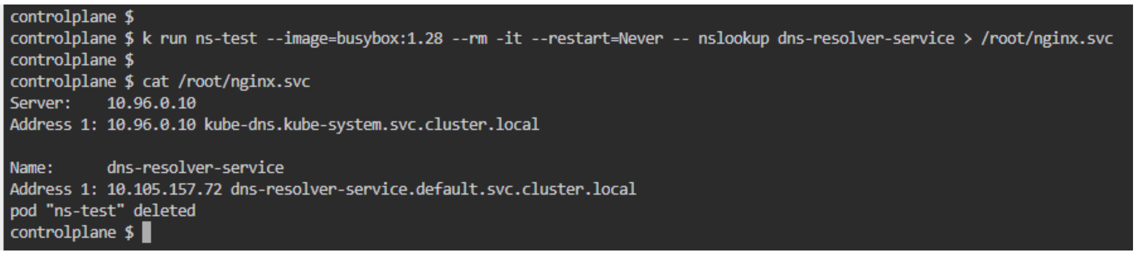
Name: Mohamad Haffiz Bin Mohd Hissham
Date: 15 Sep 2024

---

1)

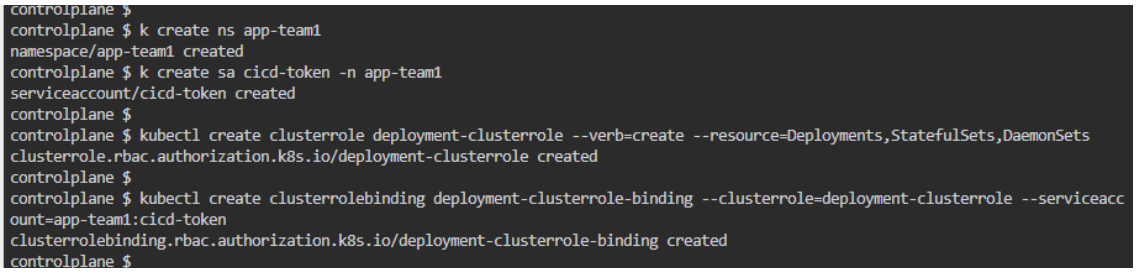


```
controlplane $ k run dns-resolver --image=nginx
pod/dns-resolver created
controlplane $
controlplane $ kubectl expose pod dns-resolver --port=80 --target-port=80 --type=clusterip
The Service "dns-resolver" is invalid: spec.type: Unsupported value: "clusterip": supported values: "ClusterIP", "ExternalName",
"LoadBalancer", "NodePort"
controlplane $ kubectl expose pod dns-resolver --port=80 --target-port=80 --type=ClusterIP
service/dns-resolver exposed
controlplane $
controlplane $ k get svc
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
dns-resolver   ClusterIP   10.106.94.157   <none>        80/TCP    3s
kubernetes     ClusterIP   10.96.0.1       <none>        443/TCP   6d14h
controlplane $ k delete svc dns-resolver
service "dns-resolver" deleted
controlplane $
controlplane $
controlplane $ kubectl expose pod dns-resolver --port=80 --target-port=80 --type=clusterip --name=dns-resolver-service
The Service "dns-resolver-service" is invalid: spec.type: Unsupported value: "clusterip": supported values: "ClusterIP", "Externa
lName", "LoadBalancer", "NodePort"
controlplane $ kubectl expose pod dns-resolver --port=80 --target-port=80 --type=ClusterIP --name=dns-resolver-service
service/dns-resolver-service exposed
controlplane $
controlplane $ k get svc
NAME                   TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
dns-resolver-service   ClusterIP   10.105.157.72   <none>        80/TCP    3s
kubernetes             ClusterIP   10.96.0.1       <none>        443/TCP   6d14h
controlplane $
```

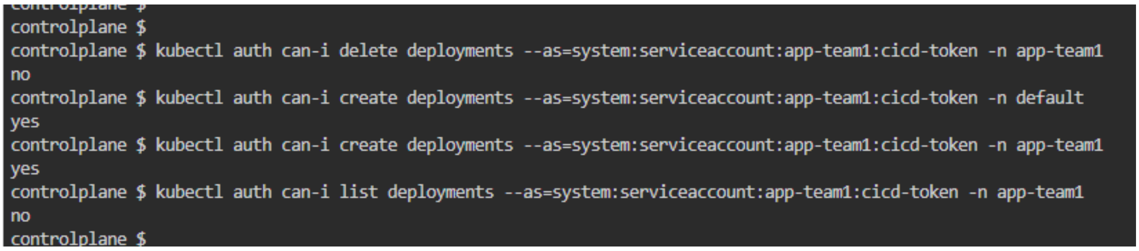


```
controlplane $
controlplane $ k run ns-test --image=busybox:1.28 --rm -it --restart=Never -- nslookup dns-resolver-service > /root/nginx.svc
controlplane $
controlplane $ cat /root/nginx.svc
Server:    10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name:      dns-resolver-service
Address 1: 10.105.157.72 dns-resolver-service.default.svc.cluster.local
pod "ns-test" deleted
controlplane $
```

2)



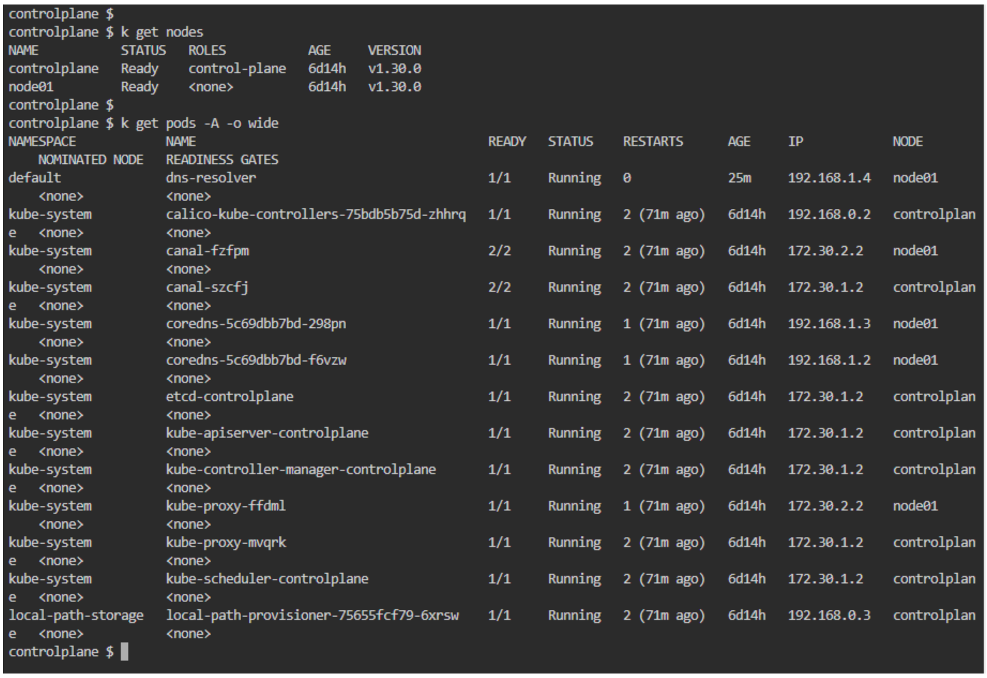
```
controlplane $
controlplane $ k create ns app-team1
namespace/app-team1 created
controlplane $ k create sa cicd-token -n app-team1
serviceaccount/cicd-token created
controlplane $
controlplane $ kubectl create clusterrole deployment-clusterrole --verb=create --resource=Deployments,StatefulSets,DaemonSets
clusterrole.rbac.authorization.k8s.io/deployment-clusterrole created
controlplane $
controlplane $ kubectl create clusterrolebinding deployment-clusterrole-binding --clusterrole=deployment-clusterrole --serviceacc
ount=app-team1:cicd-token
clusterrolebinding.rbac.authorization.k8s.io/deployment-clusterrole-binding created
controlplane $
```



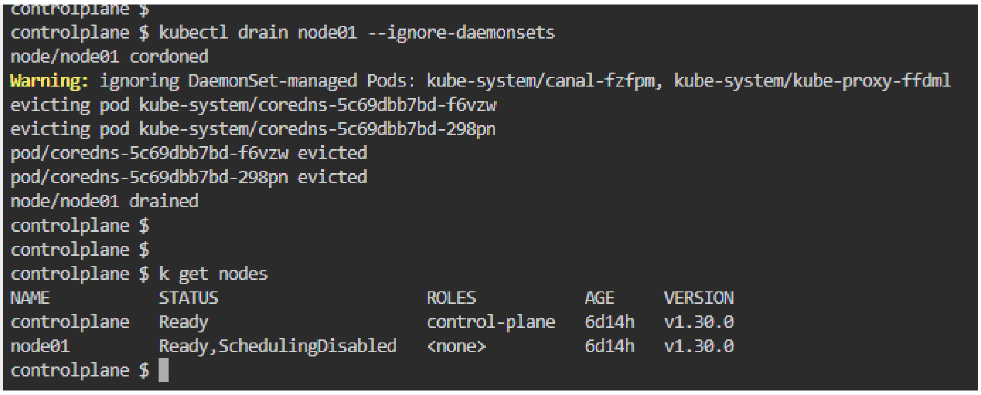
```
controlplane $
controlplane $
controlplane $ kubectl auth can-i delete deployments --as=system:serviceaccount:app-team1:cicd-token -n app-team1
no
controlplane $ kubectl auth can-i create deployments --as=system:serviceaccount:app-team1:cicd-token -n default
yes
controlplane $ kubectl auth can-i create deployments --as=system:serviceaccount:app-team1:cicd-token -n app-team1
yes
controlplane $ kubectl auth can-i list deployments --as=system:serviceaccount:app-team1:cicd-token -n app-team1
no
controlplane $
```

3

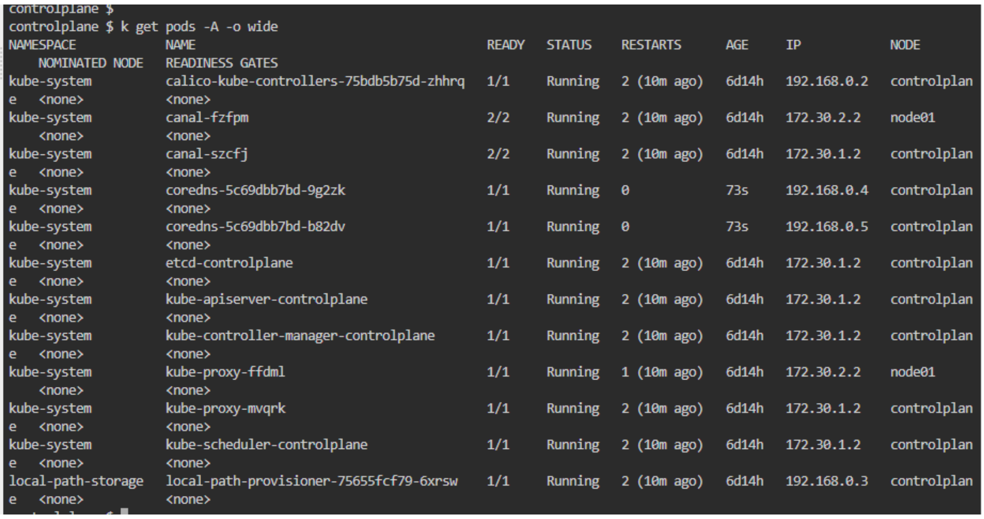
Assuming, **ek8s-node-0 =node01**



```
controlplane $
controlplane $ k get nodes
NAME           STATUS   ROLES           AGE     VERSION
controlplane   Ready    control-plane   6d14h   v1.30.0
node01         Ready    <none>          6d14h   v1.30.0
controlplane $
controlplane $ k get pods -A -o wide
NAMESPACE            NAME                                       READY   STATUS    RESTARTS      AGE     IP            NODE
    NOMINATED NODE   READINESS GATES
default              dns-resolver                               1/1     Running   0             25m     192.168.1.4   node01
    <none>           <none>
kube-system          calico-kube-controllers-75bdb5b75d-zhhrq   1/1     Running   2 (71m ago)   6d14h   192.168.0.2   controlplan
e   <none>           <none>
kube-system          canal-fzfpm                                2/2     Running   2 (71m ago)   6d14h   172.30.2.2    node01
    <none>           <none>
kube-system          canal-szcfj                                2/2     Running   2 (71m ago)   6d14h   172.30.1.2    controlplan
e   <none>           <none>
kube-system          coredns-5c69dbb7bd-298pn                   1/1     Running   1 (71m ago)   6d14h   192.168.1.3   node01
    <none>           <none>
kube-system          coredns-5c69dbb7bd-f6vzw                   1/1     Running   1 (71m ago)   6d14h   192.168.1.2   node01
    <none>           <none>
kube-system          etcd-controlplane                          1/1     Running   2 (71m ago)   6d14h   172.30.1.2    controlplan
e   <none>           <none>
kube-system          kube-apiserver-controlplane                1/1     Running   2 (71m ago)   6d14h   172.30.1.2    controlplan
e   <none>           <none>
kube-system          kube-controller-manager-controlplane       1/1     Running   2 (71m ago)   6d14h   172.30.1.2    controlplan
e   <none>           <none>
kube-system          kube-proxy-ffdml                           1/1     Running   1 (71m ago)   6d14h   172.30.2.2    node01
    <none>           <none>
kube-system          kube-proxy-mvqrk                           1/1     Running   2 (71m ago)   6d14h   172.30.1.2    controlplan
e   <none>           <none>
kube-system          kube-scheduler-controlplane                1/1     Running   2 (71m ago)   6d14h   172.30.1.2    controlplan
e   <none>           <none>
local-path-storage   local-path-provisioner-75655fcf79-6xrsw    1/1     Running   2 (71m ago)   6d14h   192.168.0.3   controlplan
e   <none>           <none>
controlplane $
```

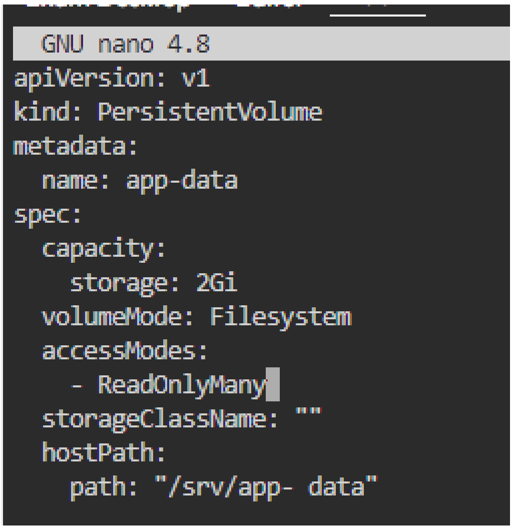


```
controlplane $
controlplane $ kubectl drain node01 --ignore-daemonsets
node/node01 cordoned
Warning: ignoring DaemonSet-managed Pods: kube-system/canal-fzfpm, kube-system/kube-proxy-ffdml
evicting pod kube-system/coredns-5c69dbb7bd-f6vzw
evicting pod kube-system/coredns-5c69dbb7bd-298pn
pod/coredns-5c69dbb7bd-f6vzw evicted
pod/coredns-5c69dbb7bd-298pn evicted
node/node01 drained
controlplane $
controlplane $
controlplane $ k get nodes
NAME           STATUS                     ROLES           AGE     VERSION
controlplane   Ready                      control-plane   6d14h   v1.30.0
node01         Ready,SchedulingDisabled   <none>          6d14h   v1.30.0
controlplane $
```

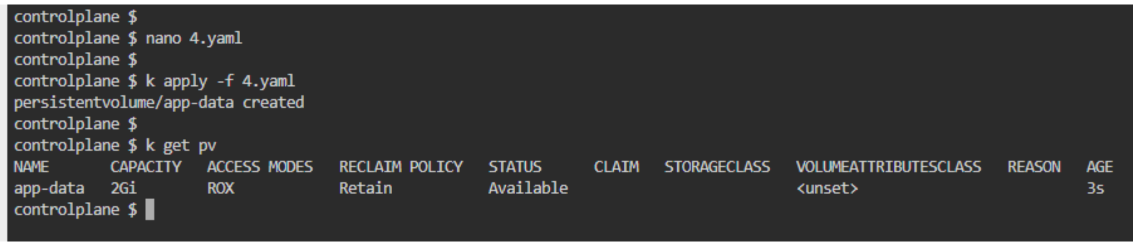


```
controlplane $
controlplane $ k get pods -A -o wide
NAMESPACE            NAME                                       READY   STATUS    RESTARTS      AGE     IP            NODE
    NOMINATED NODE   READINESS GATES
kube-system          calico-kube-controllers-75bdb5b75d-zhhrq   1/1     Running   2 (10m ago)   6d14h   192.168.0.2   controlplan
e   <none>           <none>
kube-system          canal-fzfpm                                2/2     Running   2 (10m ago)   6d14h   172.30.2.2    node01
    <none>           <none>
kube-system          canal-szcfj                                2/2     Running   2 (10m ago)   6d14h   172.30.1.2    controlplan
e   <none>           <none>
kube-system          coredns-5c69dbb7bd-9g2zk                   1/1     Running   0             73s     192.168.0.4   controlplan
e   <none>           <none>
kube-system          coredns-5c69dbb7bd-b82dv                   1/1     Running   0             73s     192.168.0.5   controlplan
e   <none>           <none>
kube-system          etcd-controlplane                          1/1     Running   2 (10m ago)   6d14h   172.30.1.2    controlplan
e   <none>           <none>
kube-system          kube-apiserver-controlplane                1/1     Running   2 (10m ago)   6d14h   172.30.1.2    controlplan
e   <none>           <none>
kube-system          kube-controller-manager-controlplane       1/1     Running   2 (10m ago)   6d14h   172.30.1.2    controlplan
e   <none>           <none>
kube-system          kube-proxy-ffdml                           1/1     Running   1 (10m ago)   6d14h   172.30.2.2    node01
    <none>           <none>
kube-system          kube-proxy-mvqrk                           1/1     Running   2 (10m ago)   6d14h   172.30.1.2    controlplan
e   <none>           <none>
kube-system          kube-scheduler-controlplane                1/1     Running   2 (10m ago)   6d14h   172.30.1.2    controlplan
e   <none>           <none>
local-path-storage   local-path-provisioner-75655fcf79-6xrsw    1/1     Running   2 (10m ago)   6d14h   192.168.0.3   controlplan
e   <none>           <none>
```

4)

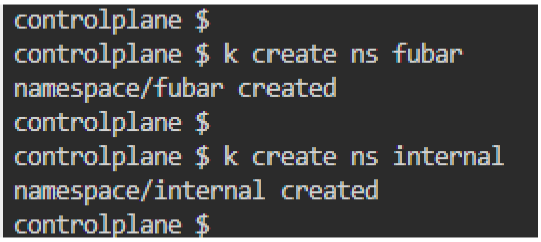


```
  GNU nano 4.8
apiVersion: v1
kind: PersistentVolume
metadata:
  name: app-data
spec:
  capacity:
    storage: 2Gi
  volumeMode: Filesystem
  accessModes:
    - ReadOnlyMany
  storageClassName: ""
  hostPath:
    path: "/srv/app- data"
```

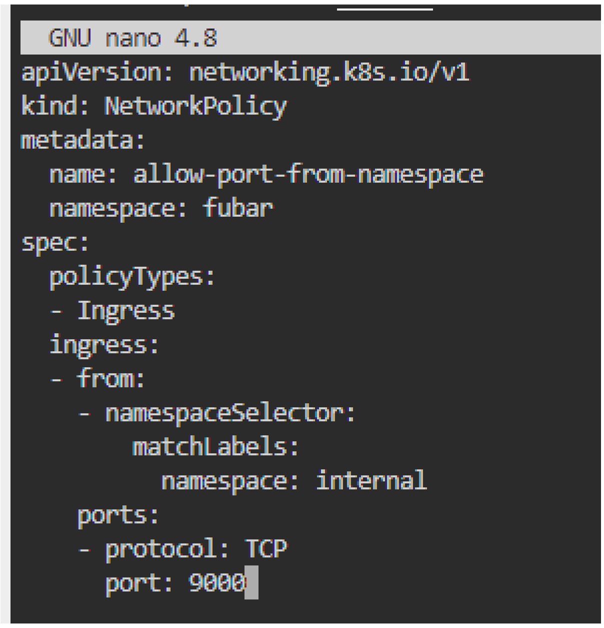


```
controlplane $
controlplane $ nano 4.yaml
controlplane $
controlplane $ k apply -f 4.yaml
persistentvolume/app-data created
controlplane $
controlplane $ k get pv
NAME       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   VOLUMEATTRIBUTESCLASS   REASON   AGE
app-data   2Gi        ROX            Retain           Available                          <unset>                          3s
controlplane $
```

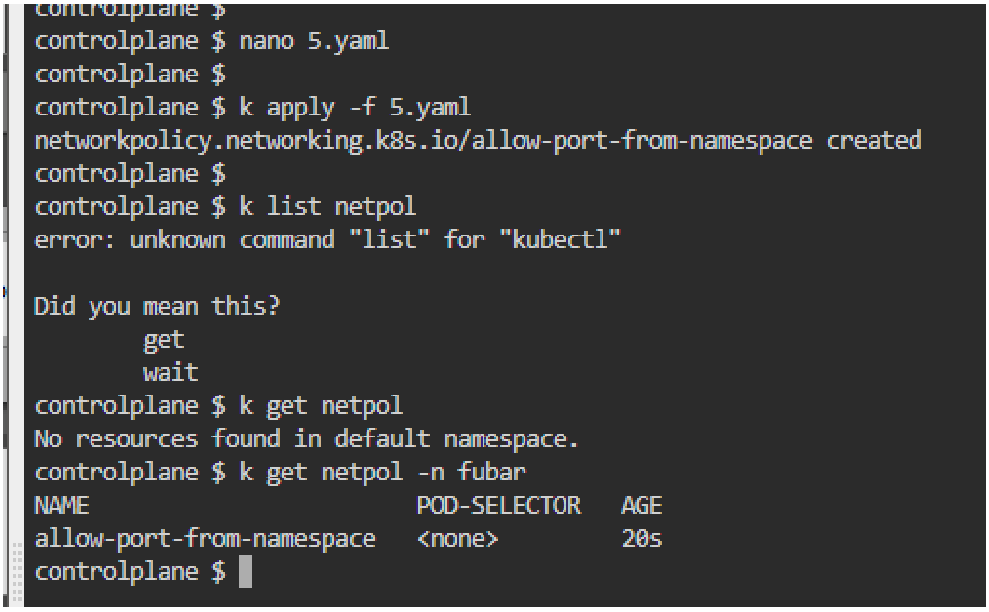
5)



```
controlplane $
controlplane $ k create ns fubar
namespace/fubar created
controlplane $
controlplane $ k create ns internal
namespace/internal created
controlplane $
```

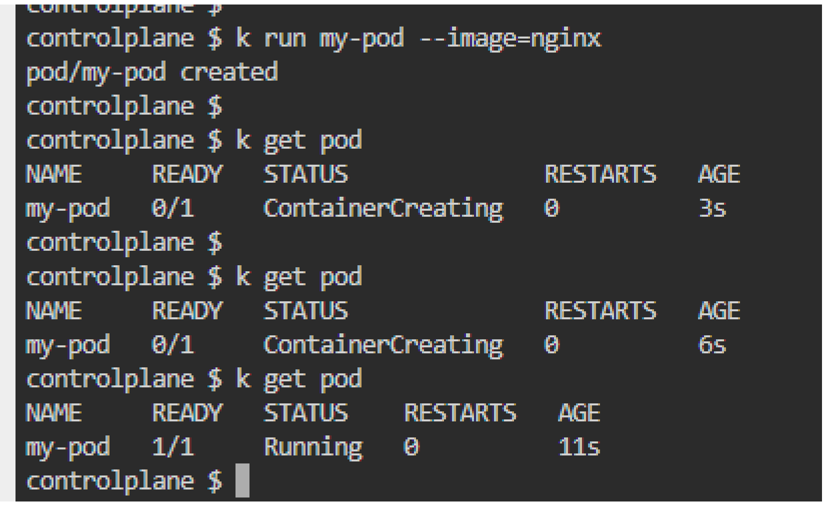


```
  GNU nano 4.8
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-port-from-namespace
  namespace: fubar
spec:
  policyTypes:
  - Ingress
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          namespace: internal
    ports:
    - protocol: TCP
      port: 9000
```

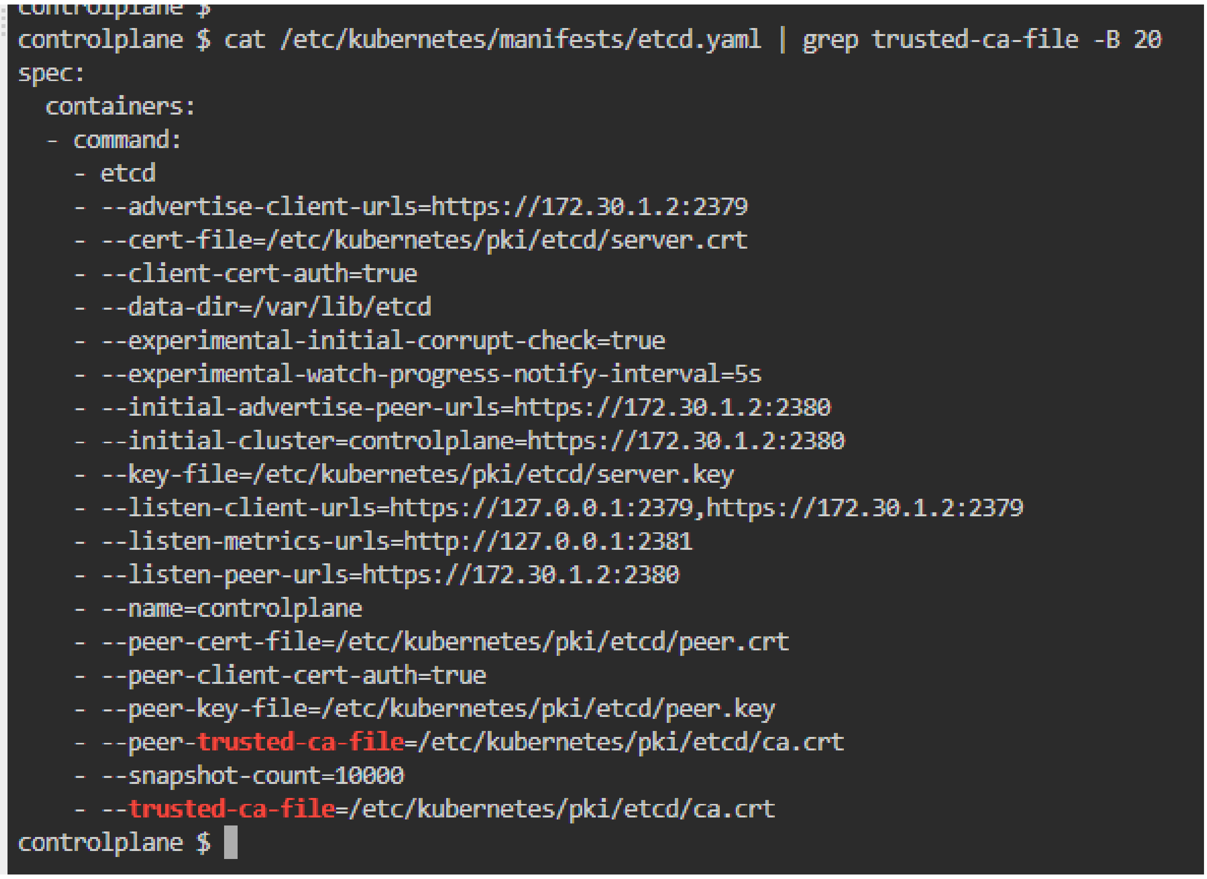


```
controlplane $
controlplane $ nano 5.yaml
controlplane $
controlplane $ k apply -f 5.yaml
networkpolicy.networking.k8s.io/allow-port-from-namespace created
controlplane $
controlplane $ k list netpol
error: unknown command "list" for "kubectl"

Did you mean this?
        get
        wait
controlplane $ k get netpol
No resources found in default namespace.
controlplane $ k get netpol -n fubar
NAME                        POD-SELECTOR   AGE
allow-port-from-namespace   <none>         20s
controlplane $
```

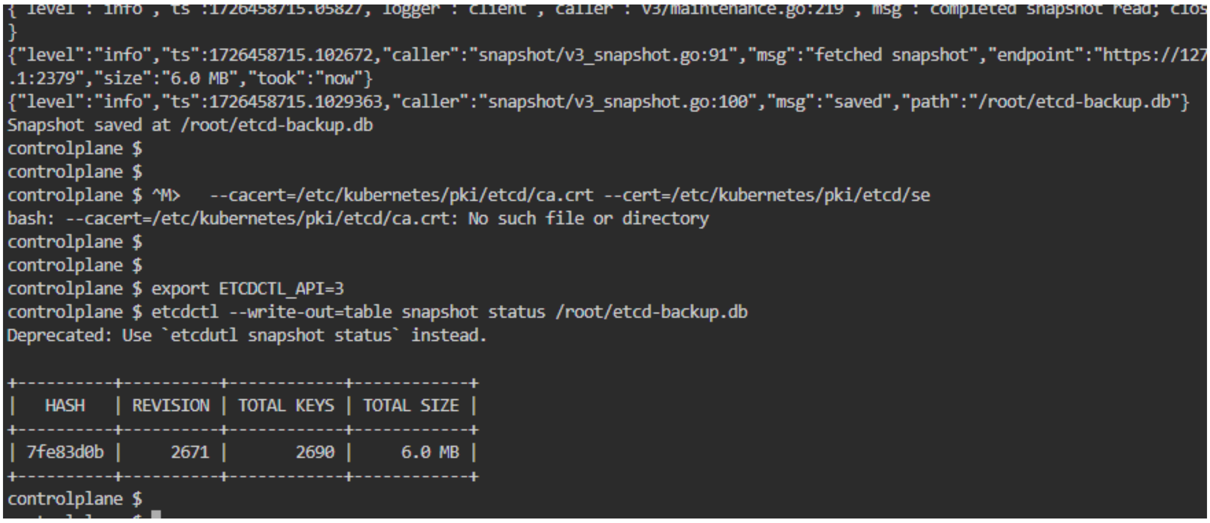
6)



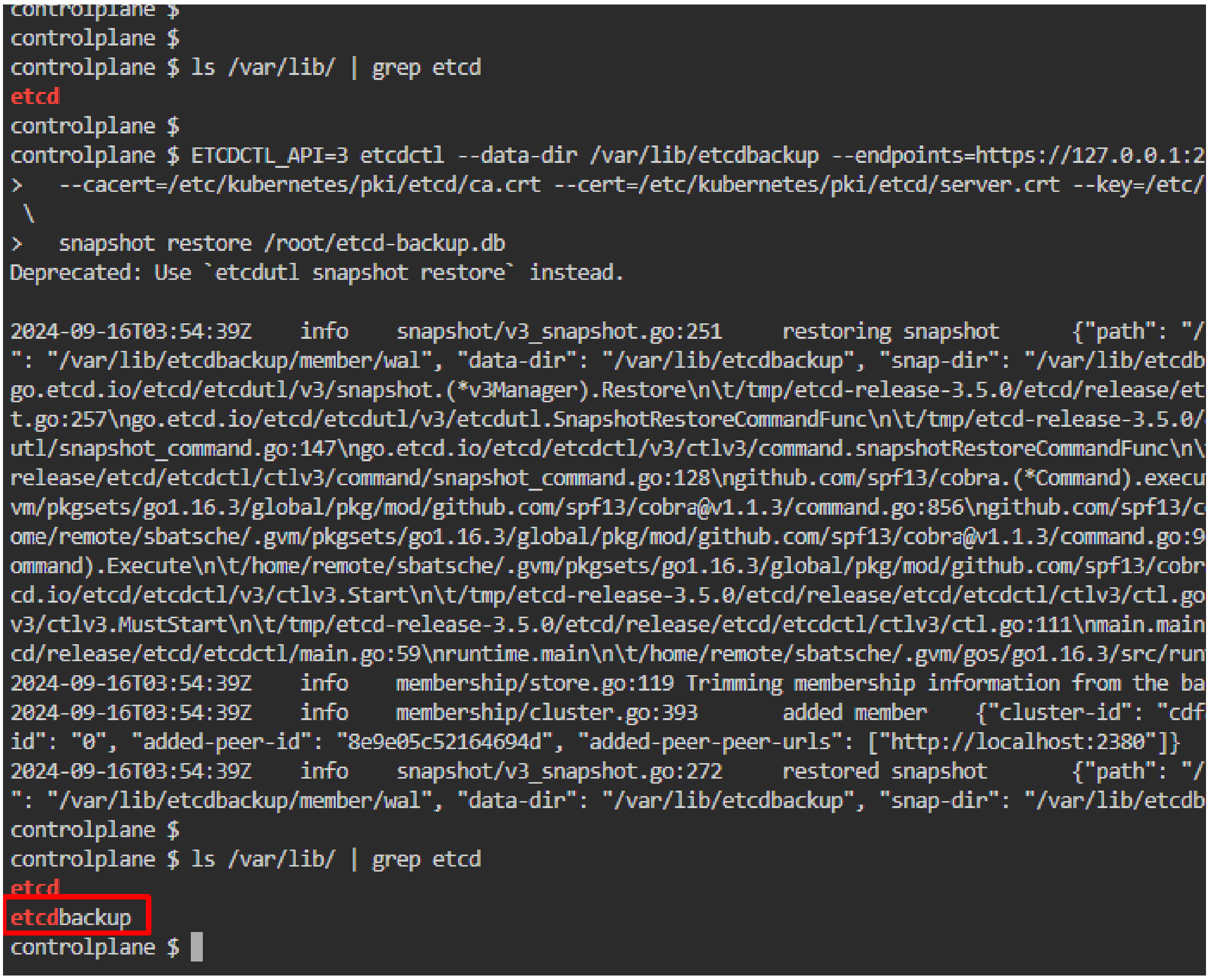
```
controlplane $
controlplane $ k run my-pod --image=nginx
pod/my-pod created
controlplane $
controlplane $ k get pod
NAME     READY   STATUS              RESTARTS   AGE
my-pod   0/1     ContainerCreating   0          3s
controlplane $
controlplane $ k get pod
NAME     READY   STATUS              RESTARTS   AGE
my-pod   0/1     ContainerCreating   0          6s
controlplane $ k get pod
NAME     READY   STATUS    RESTARTS   AGE
my-pod   1/1     Running   0          11s
controlplane $ 
```



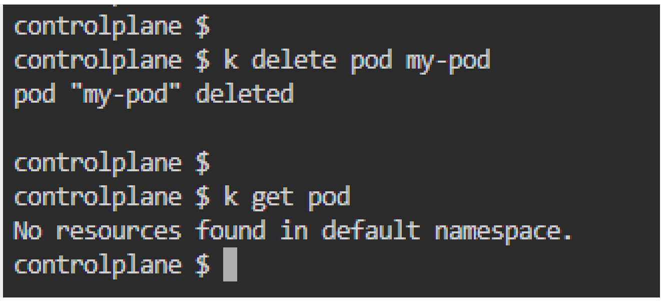
```
controlplane $
controlplane $ cat /etc/kubernetes/manifests/etcd.yaml | grep trusted-ca-file -B 20
spec:
  containers:
  - command:
    - etcd
    - --advertise-client-urls=https://172.30.1.2:2379
    - --cert-file=/etc/kubernetes/pki/etcd/server.crt
    - --client-cert-auth=true
    - --data-dir=/var/lib/etcd
    - --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
    - --peer-trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt
    - --snapshot-count=10000
    - --trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt
controlplane $ 
```



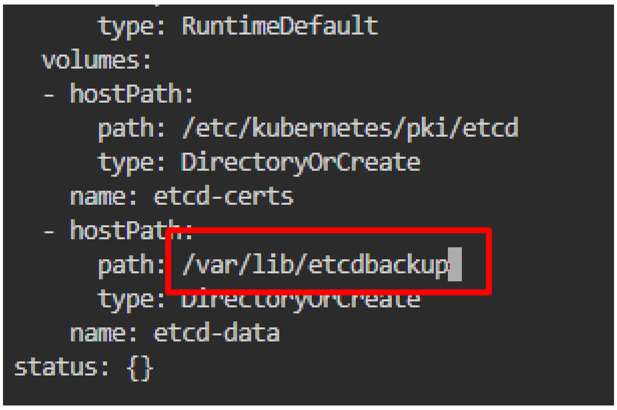
```
{"level":"info","ts":1726458715.05827,"logger":"client","caller":"v3/maintenance.go:219","msg":"completed snapshot read; clos
}
{"level":"info","ts":1726458715.102672,"caller":"snapshot/v3_snapshot.go:91","msg":"fetched snapshot","endpoint":"https://127
.1:2379","size":"6.0 MB","took":"now"}
{"level":"info","ts":1726458715.1029363,"caller":"snapshot/v3_snapshot.go:100","msg":"saved","path":"/root/etcd-backup.db"}
Snapshot saved at /root/etcd-backup.db
controlplane $
controlplane $
controlplane $ ^M>   --cacert=/etc/kubernetes/pki/etcd/ca.crt --cert=/etc/kubernetes/pki/etcd/se
bash: --cacert=/etc/kubernetes/pki/etcd/ca.crt: No such file or directory
controlplane $
controlplane $
controlplane $ export ETCDCTL_API=3
controlplane $ etcdctl --write-out=table snapshot status /root/etcd-backup.db
Deprecated: Use `etcdutl snapshot status` instead.

+----------+----------+------------+------------+
|   HASH   | REVISION | TOTAL KEYS | TOTAL SIZE |
+----------+----------+------------+------------+
| 7fe83d0b |     2671 |       2690 |     6.0 MB |
+----------+----------+------------+------------+
controlplane $
```



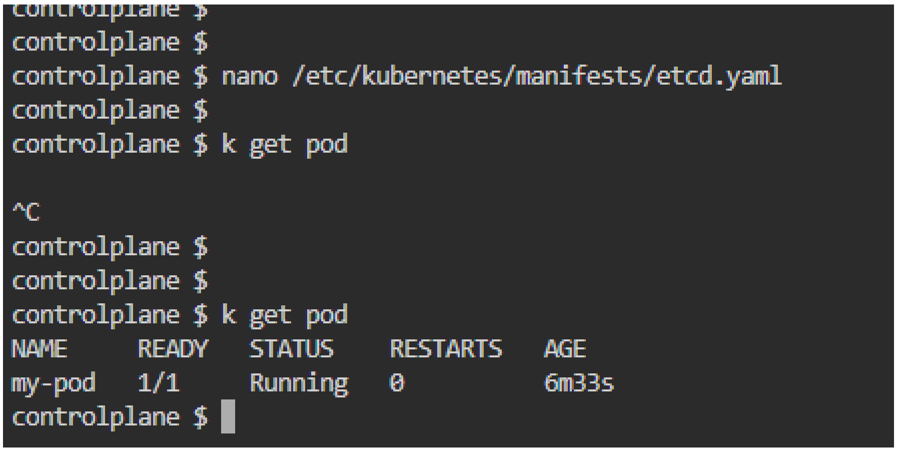
```
controlplane $
controlplane $
controlplane $ ls /var/lib/ | grep etcd
etcd
controlplane $
controlplane $ ETCDCTL_API=3 etcdctl --data-dir /var/lib/etcdbackup --endpoints=https://127.0.0.1:2
>   --cacert=/etc/kubernetes/pki/etcd/ca.crt --cert=/etc/kubernetes/pki/etcd/server.crt --key=/etc/
 \
>   snapshot restore /root/etcd-backup.db
Deprecated: Use `etcdutl snapshot restore` instead.

2024-09-16T03:54:39Z    info    snapshot/v3_snapshot.go:251     restoring snapshot      {"path": "/
": "/var/lib/etcdbackup/member/wal", "data-dir": "/var/lib/etcdbackup", "snap-dir": "/var/lib/etcdb
go.etcd.io/etcd/etcdutl/v3/snapshot.(*v3Manager).Restore\n\t/tmp/etcd-release-3.5.0/etcd/release/et
t.go:257\ngo.etcd.io/etcd/etcdutl/v3/etcdutl.SnapshotRestoreCommandFunc\n\t/tmp/etcd-release-3.5.0/
utl/snapshot_command.go:147\ngo.etcd.io/etcd/etcdctl/v3/ctlv3/command.snapshotRestoreCommandFunc\n\
release/etcd/etcdctl/ctlv3/command/snapshot_command.go:128\ngithub.com/spf13/cobra.(*Command).execu
vm/pkgsets/go1.16.3/global/pkg/mod/github.com/spf13/cobra@v1.1.3/command.go:856\ngithub.com/spf13/c
ome/remote/sbatsche/.gvm/pkgsets/go1.16.3/global/pkg/mod/github.com/spf13/cobra@v1.1.3/command.go:9
ommand).Execute\n\t/home/remote/sbatsche/.gvm/pkgsets/go1.16.3/global/pkg/mod/github.com/spf13/cobr
cd.io/etcd/etcdctl/v3/ctlv3.Start\n\t/tmp/etcd-release-3.5.0/etcd/release/etcd/etcdctl/ctlv3/ctl.go
v3/ctlv3.MustStart\n\t/tmp/etcd-release-3.5.0/etcd/release/etcd/etcdctl/ctlv3/ctl.go:111\nmain.main
cd/release/etcd/etcdctl/main.go:59\nruntime.main\n\t/home/remote/sbatsche/.gvm/gos/go1.16.3/src/run
2024-09-16T03:54:39Z    info    membership/store.go:119 Trimming membership information from the ba
2024-09-16T03:54:39Z    info    membership/cluster.go:393       added member    {"cluster-id": "cdf
id": "0", "added-peer-id": "8e9e05c52164694d", "added-peer-peer-urls": ["http://localhost:2380"]}
2024-09-16T03:54:39Z    info    snapshot/v3_snapshot.go:272     restored snapshot       {"path": "/
": "/var/lib/etcdbackup/member/wal", "data-dir": "/var/lib/etcdbackup", "snap-dir": "/var/lib/etcdb
controlplane $
controlplane $ ls /var/lib/ | grep etcd
etcd
etcdbackup
controlplane $
```



```
controlplane $
controlplane $ k delete pod my-pod
pod "my-pod" deleted

controlplane $
controlplane $ k get pod
No resources found in default namespace.
controlplane $
```

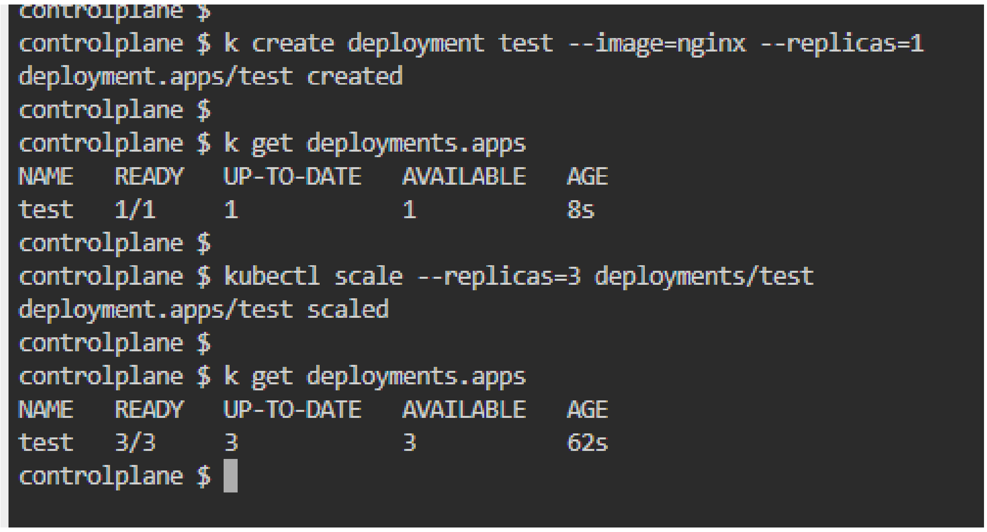


```
      type: RuntimeDefault
  volumes:
  - hostPath:
      path: /etc/kubernetes/pki/etcd
      type: DirectoryOrCreate
    name: etcd-certs
  - hostPath:
      path: /var/lib/etcdbackup
      type: DirectoryOrCreate
    name: etcd-data
status: {}
```



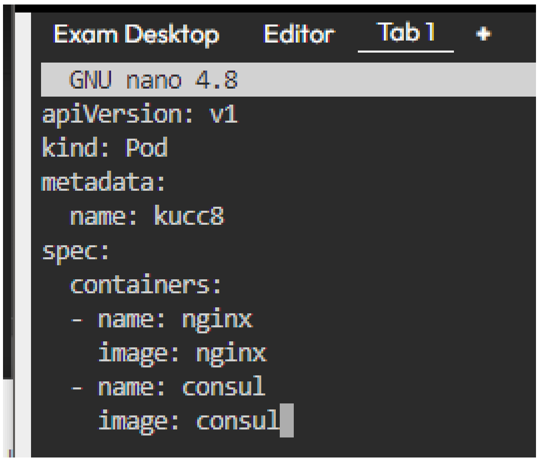
```
controlplane $
controlplane $
controlplane $ nano /etc/kubernetes/manifests/etcd.yaml
controlplane $
controlplane $ k get pod

^C
controlplane $
controlplane $
controlplane $ k get pod
NAME     READY   STATUS    RESTARTS   AGE
my-pod   1/1     Running   0          6m33s
controlplane $
```

7)

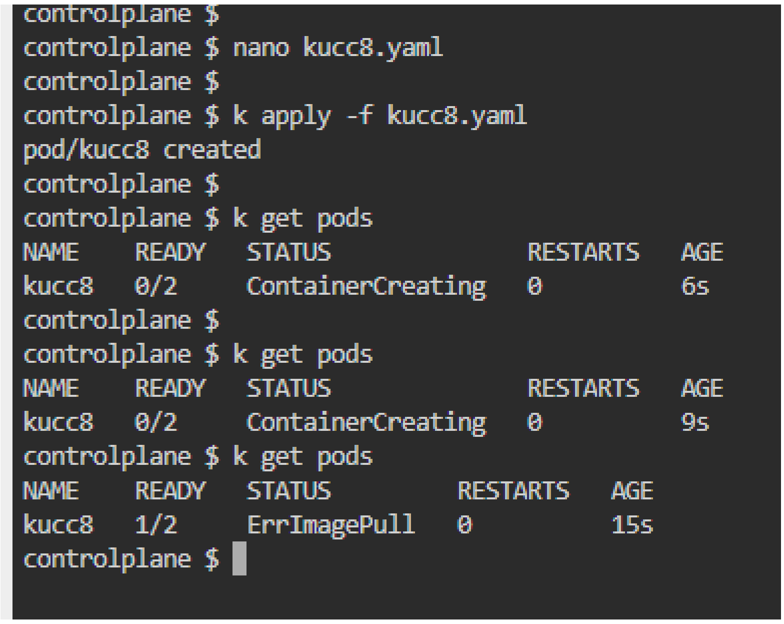


```
controlplane $
controlplane $ k create deployment test --image=nginx --replicas=1
deployment.apps/test created
controlplane $
controlplane $ k get deployments.apps
NAME   READY   UP-TO-DATE   AVAILABLE   AGE
test   1/1     1            1           8s
controlplane $
controlplane $ kubectl scale --replicas=3 deployments/test
deployment.apps/test scaled
controlplane $
controlplane $ k get deployments.apps
NAME   READY   UP-TO-DATE   AVAILABLE   AGE
test   3/3     3            3           62s
controlplane $
```

8)

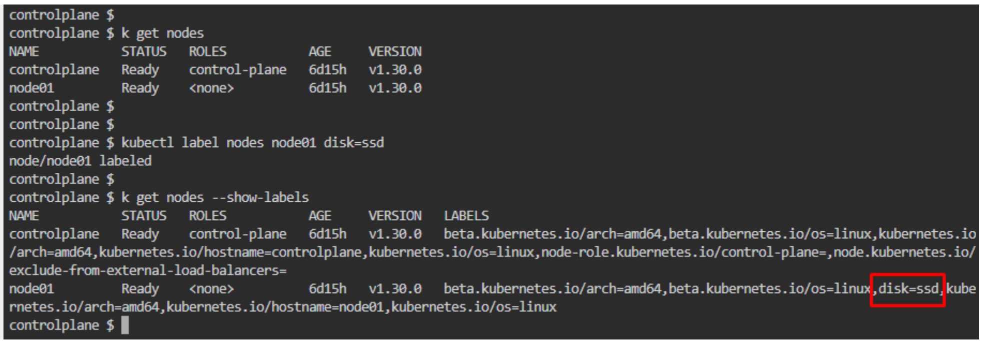


```
  GNU nano 4.8
apiVersion: v1
kind: Pod
metadata:
  name: kucc8
spec:
  containers:
  - name: nginx
    image: nginx
  - name: consul
    image: consul
```

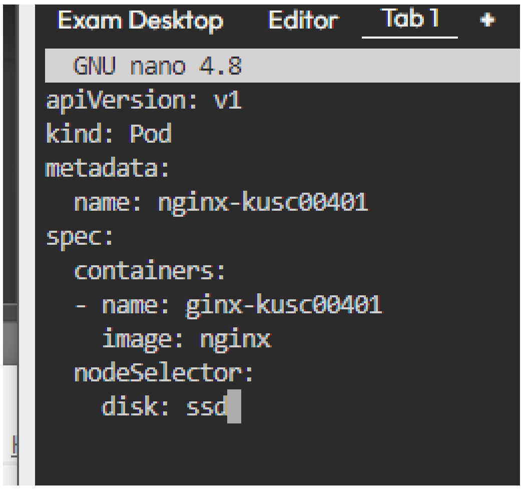


```
controlplane $
controlplane $ nano kucc8.yaml
controlplane $
controlplane $ k apply -f kucc8.yaml
pod/kucc8 created
controlplane $
controlplane $ k get pods
NAME    READY   STATUS              RESTARTS   AGE
kucc8   0/2     ContainerCreating   0          6s
controlplane $
controlplane $ k get pods
NAME    READY   STATUS              RESTARTS   AGE
kucc8   0/2     ContainerCreating   0          9s
controlplane $ k get pods
NAME    READY   STATUS         RESTARTS   AGE
kucc8   1/2     ErrImagePull   0          15s
controlplane $
```

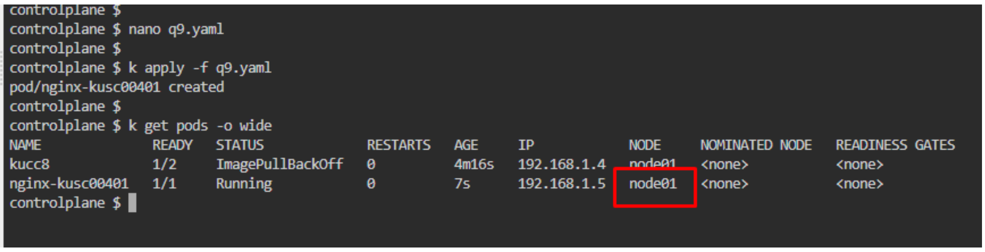
9)



```
controlplane $
controlplane $ k get nodes
NAME           STATUS   ROLES           AGE     VERSION
controlplane   Ready    control-plane   6d15h   v1.30.0
node01         Ready    <none>          6d15h   v1.30.0
controlplane $
controlplane $
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   6d15h   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>          6d15h   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
controlplane $
```

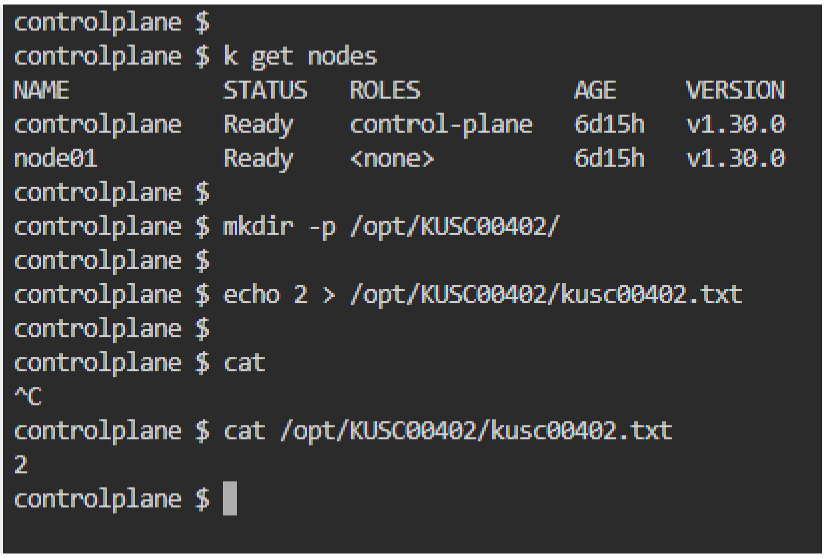


```
GNU nano 4.8
apiVersion: v1
kind: Pod
metadata:
  name: nginx-kusc00401
spec:
  containers:
  - name: ginx-kusc00401
    image: nginx
  nodeSelector:
    disk: ssd
```



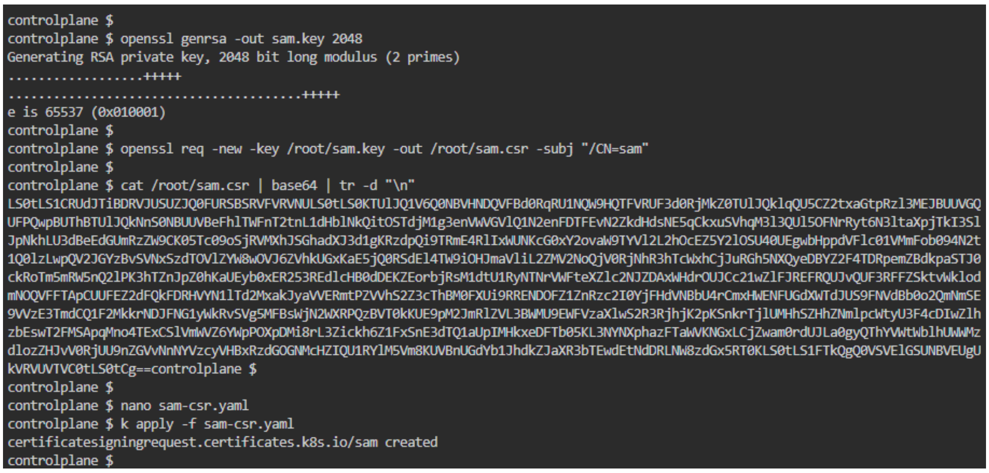
```
controlplane $
controlplane $ nano q9.yaml
controlplane $
controlplane $ k apply -f q9.yaml
pod/nginx-kusc00401 created
controlplane $
controlplane $ k get pods -o wide
NAME              READY   STATUS             RESTARTS   AGE     IP            NODE     NOMINATED NODE   READINESS GATES
kucc8             1/2     ImagePullBackOff   0          4m16s   192.168.1.4   node01   <none>           <none>
nginx-kusc00401   1/1     Running            0          7s      192.168.1.5   node01   <none>           <none>
controlplane $
```

10)

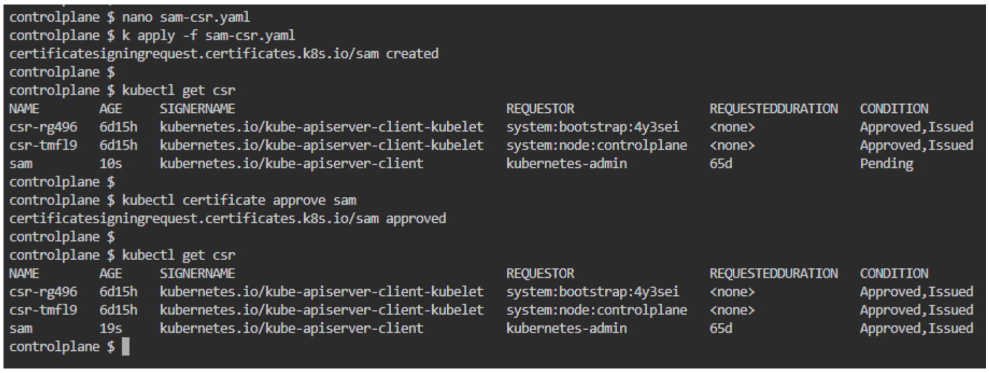


```
controlplane $
controlplane $ k get nodes
NAME           STATUS   ROLES           AGE     VERSION
controlplane   Ready    control-plane   6d15h   v1.30.0
node01         Ready    <none>          6d15h   v1.30.0
controlplane $
controlplane $ mkdir -p /opt/KUSC00402/
controlplane $
controlplane $ echo 2 > /opt/KUSC00402/kusc00402.txt
controlplane $
controlplane $ cat
^C
controlplane $ cat /opt/KUSC00402/kusc00402.txt
2
controlplane $
```

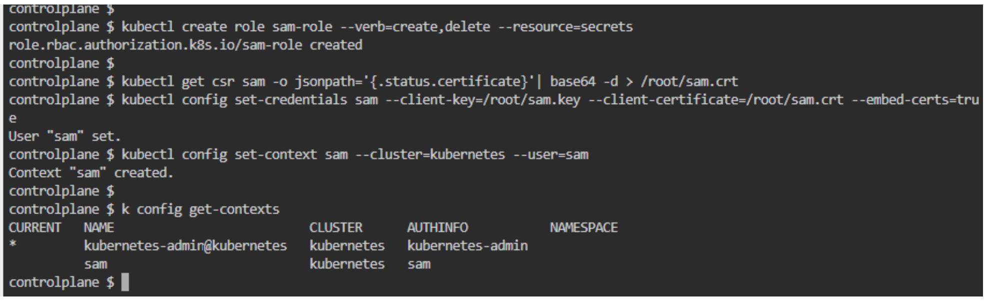
11)



```
controlplane $
controlplane $ openssl genrsa -out sam.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.................+++++
......................................+++++
e is 65537 (0x010001)
controlplane $
controlplane $ openssl req -new -key /root/sam.key -out /root/sam.csr -subj "/CN=sam"
controlplane $
controlplane $ cat /root/sam.csr | base64 | tr -d "\n"
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0KTUlJQ1V6Q0NBVHNDQVFBd0RqRU1NQW9HQTFVRUF3d0RjMkZ0TUlJQklqQU5CZ2txaGtpRzl3MEJBUUVGQ
UFPQwpBUThBTUlJQkNnS0NBUUVBeFhlTWFnT2tnL1dHblNkQitOSTdjM1g3enVWVGVlQ1N2enFDTFEvN2ZkdHdsNE5qCkxuSVhqM3l3QUl5OFNrRyt6N3ltaXpjTkI3Sl
JpNkhLU3dBeEdGUmRzZW9CK05Tc09oSjRVMXhJSGhadXJ3d1gKRzdpQi9TRmE4RlIxWUNKcG0xY2ovaW9TYVl2L2hOcEZ5Y2lOSU40UEgwbHppdVFlc01VMmFob094N2t
1Q0lzLwpQV2JGYzBvSVNxSzdTOVlZYW8wOVJ6ZVhkUGxKaE5jQ0RSdEl4TW9iOHJmaVliL2ZMV2NoQjV0RjNhR3hTcWxhCjJuRGh5NXQyeDBYZ2F4TDRpemZBdkpaSTJ0
ckRoTm5mRW5nQ2lPK3hTZnJpZ0hKaUEyb0xER253REdlcHB0dDEKZEorbjRsM1dtU1RyNTNrVWFteXZlc2NJZDAxWHdrOUJCc21wZlFJREFRQUJvQUF3RFFZSktvWklod
mNOQVFFTApCUUFEZ2dFQkFDRHVYN1lTd2MxakJyaVVERmtPZVVhS2Z3cThBM0FXUi9RRENDOFZ1ZnRzc2I0YjFHdVNBbU4rCmxHWENFUGdXWTdJUS9FNVdBb0o2QmNmSE
9VVzE3TmdCQ1F2MkkrNDJFNG1yWkRvSVg5MFBsWjN2WXRPQzBVT0kKUE9pM2JmRlZVL3BWMU9EWFVzaXlwS2R3RjhjK2pKSnkrTjlUMHhSZHhZNmlpcWtyU3F4cDIwZlh
zbEswT2FMSApqMno4TExCSlVmWVZ6YWpPOXpDMi8rL3Zickh6Z1FxSnE3dTQ1aUpIMHkxeDFTb05KL3NYNXphazFTaWVKNGxLCjZwam0rdUJLa0gyQThYVWtWblhUWWMz
dlozZHJvV0RjUU9nZGVvNnNYVzcyVHBxRzdGOGNMcHZIQU1RYlM5Vm8KUVBnUGdYb1JhdkZJaXR3bTEwdEtNdDRLNW8zdGx5RT0KLS0tLS1FTkQgQ0VSVElGSUNBVEUgU
kVRVUVTVC0tLS0tCg==controlplane $
controlplane $
controlplane $ nano sam-csr.yaml
controlplane $ k apply -f sam-csr.yaml
certificatesigningrequest.certificates.k8s.io/sam created
controlplane $
```

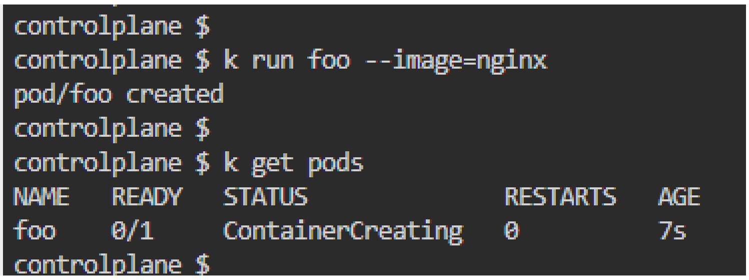


```
controlplane $ nano sam-csr.yaml
controlplane $ k apply -f sam-csr.yaml
certificatesigningrequest.certificates.k8s.io/sam created
controlplane $
controlplane $ kubectl get csr
NAME        AGE     SIGNERNAME                                    REQUESTOR                  REQUESTEDDURATION   CONDITION
csr-rg496   6d15h   kubernetes.io/kube-apiserver-client-kubelet   system:bootstrap:4y3sei    <none>              Approved,Issued
csr-tmfl9   6d15h   kubernetes.io/kube-apiserver-client-kubelet   system:node:controlplane   <none>              Approved,Issued
sam         10s     kubernetes.io/kube-apiserver-client           kubernetes-admin           65d                 Pending
controlplane $
controlplane $ kubectl certificate approve sam
certificatesigningrequest.certificates.k8s.io/sam approved
controlplane $
controlplane $ kubectl get csr
NAME        AGE     SIGNERNAME                                    REQUESTOR                  REQUESTEDDURATION   CONDITION
csr-rg496   6d15h   kubernetes.io/kube-apiserver-client-kubelet   system:bootstrap:4y3sei    <none>              Approved,Issued
csr-tmfl9   6d15h   kubernetes.io/kube-apiserver-client-kubelet   system:node:controlplane   <none>              Approved,Issued
sam         19s     kubernetes.io/kube-apiserver-client           kubernetes-admin           65d                 Approved,Issued
controlplane $ █
```

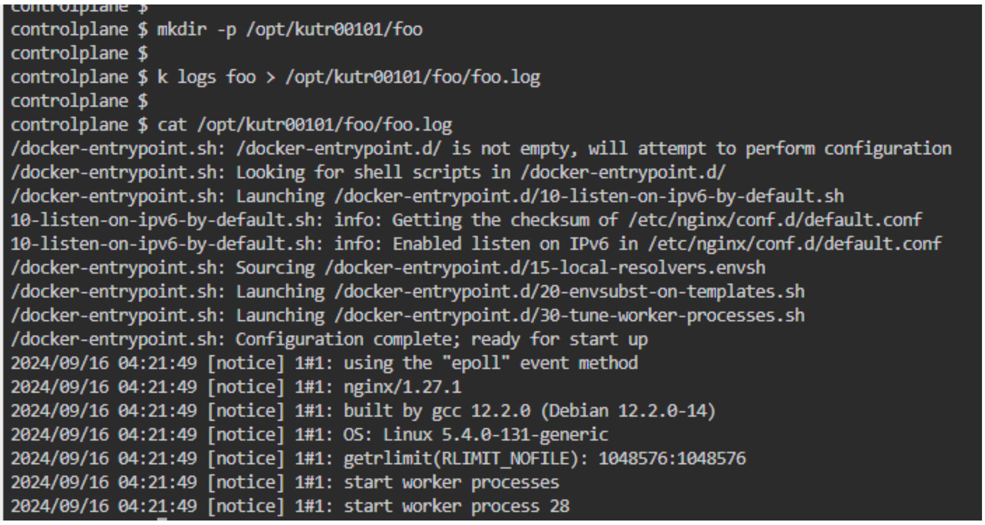


```
controlplane $
controlplane $ kubectl create role sam-role --verb=create,delete --resource=secrets
role.rbac.authorization.k8s.io/sam-role created
controlplane $
controlplane $ kubectl get csr sam -o jsonpath='{.status.certificate}'| base64 -d > /root/sam.crt
controlplane $ kubectl config set-credentials sam --client-key=/root/sam.key --client-certificate=/root/sam.crt --embed-certs=tru
e
User "sam" set.
controlplane $ kubectl config set-context sam --cluster=kubernetes --user=sam
Context "sam" created.
controlplane $
controlplane $ k config get-contexts
CURRENT   NAME                          CLUSTER      AUTHINFO           NAMESPACE
*         kubernetes-admin@kubernetes   kubernetes   kubernetes-admin
          sam                           kubernetes   sam
controlplane $ █
```

12)

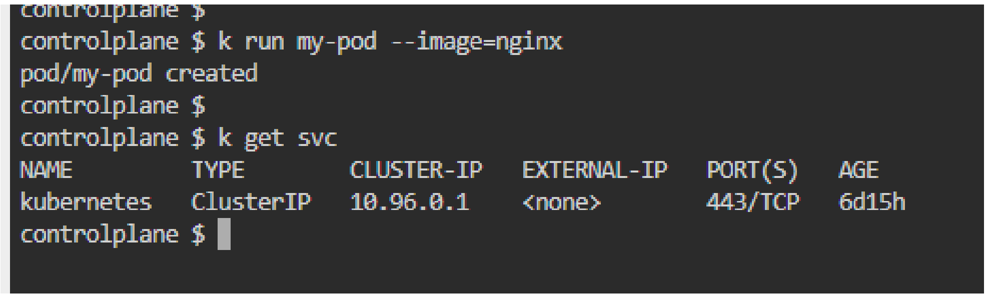


```
controlplane $
controlplane $ k run foo --image=nginx
pod/foo created
controlplane $
controlplane $ k get pods
NAME   READY   STATUS              RESTARTS   AGE
foo    0/1     ContainerCreating   0          7s
controlplane $
```

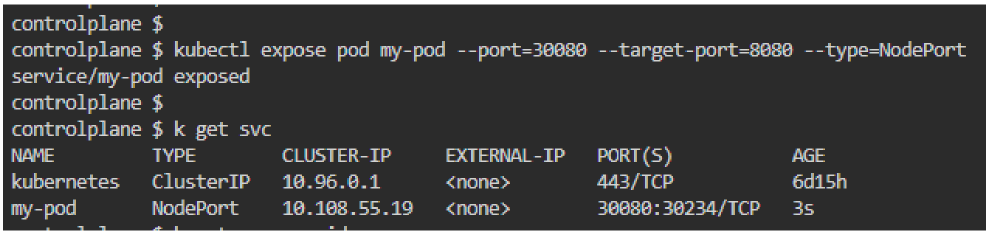


```
controlplane $
controlplane $ mkdir -p /opt/kutr00101/foo
controlplane $
controlplane $ k logs foo > /opt/kutr00101/foo/foo.log
controlplane $
controlplane $ cat /opt/kutr00101/foo/foo.log
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/09/16 04:21:49 [notice] 1#1: using the "epoll" event method
2024/09/16 04:21:49 [notice] 1#1: nginx/1.27.1
2024/09/16 04:21:49 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/09/16 04:21:49 [notice] 1#1: OS: Linux 5.4.0-131-generic
2024/09/16 04:21:49 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/09/16 04:21:49 [notice] 1#1: start worker processes
2024/09/16 04:21:49 [notice] 1#1: start worker process 28
```

13)

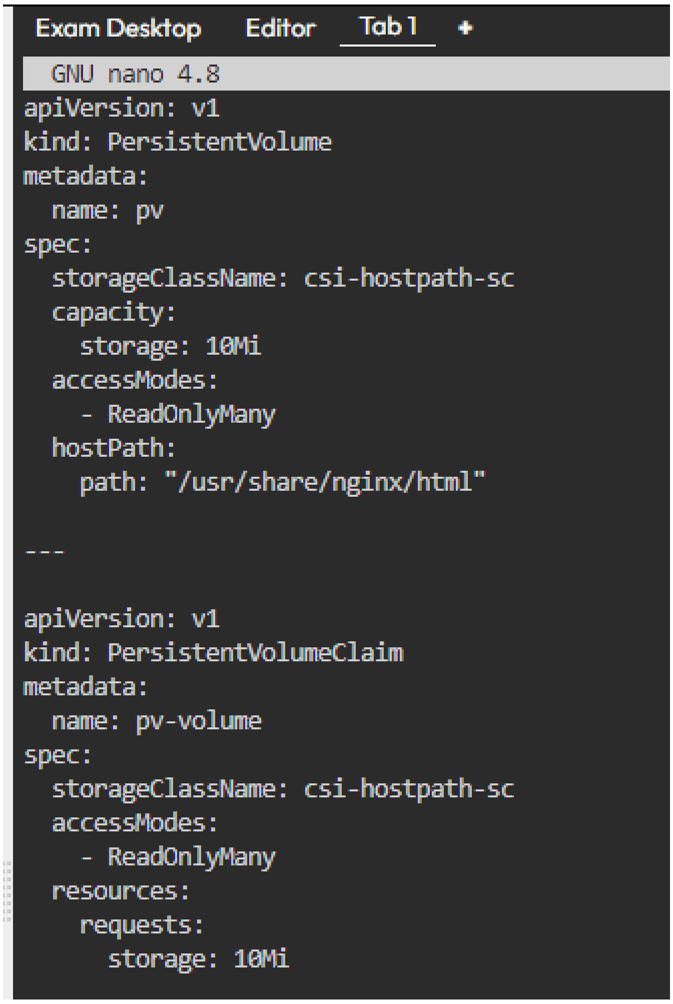


```
controlplane $
controlplane $ k run my-pod --image=nginx
pod/my-pod created
controlplane $
controlplane $ k get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   6d15h
controlplane $
```

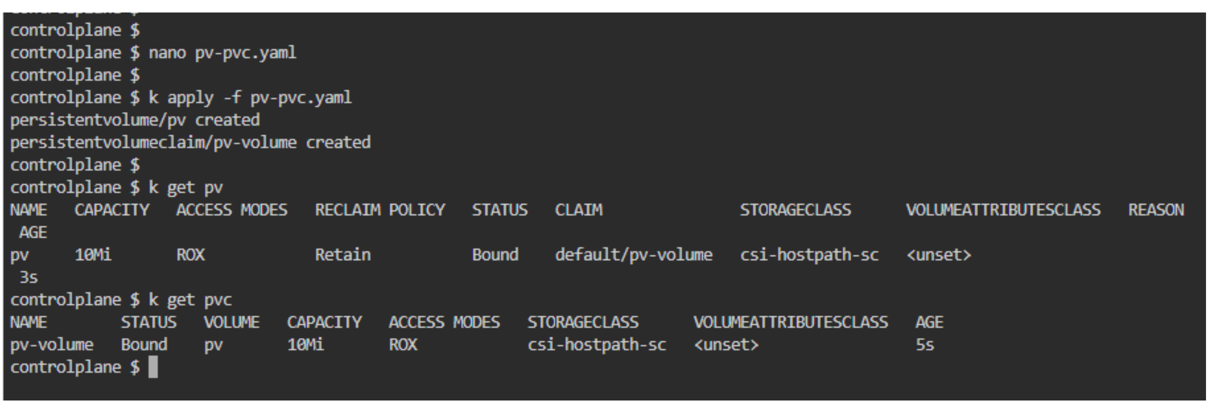


```
controlplane $
controlplane $ kubectl expose pod my-pod --port=30080 --target-port=8080 --type=NodePort
service/my-pod exposed
controlplane $
controlplane $ k get svc
NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)           AGE
kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP           6d15h
my-pod       NodePort    10.108.55.19   <none>        30080:30234/TCP   3s
```

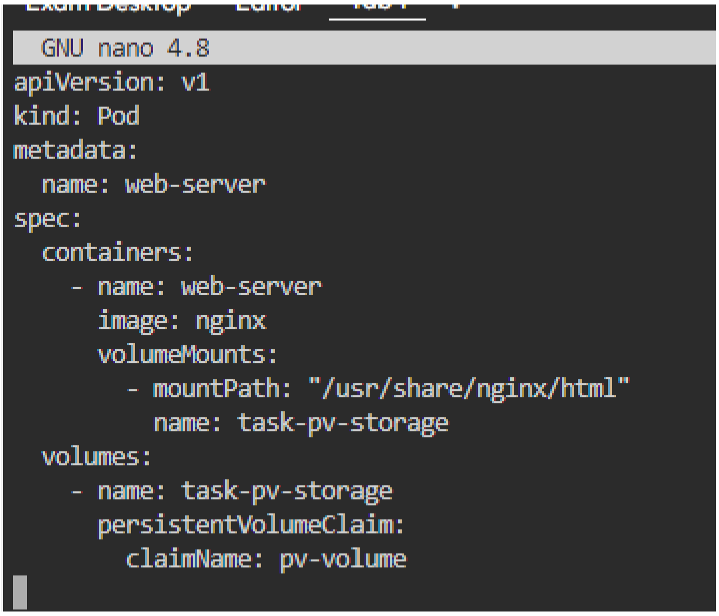
14)



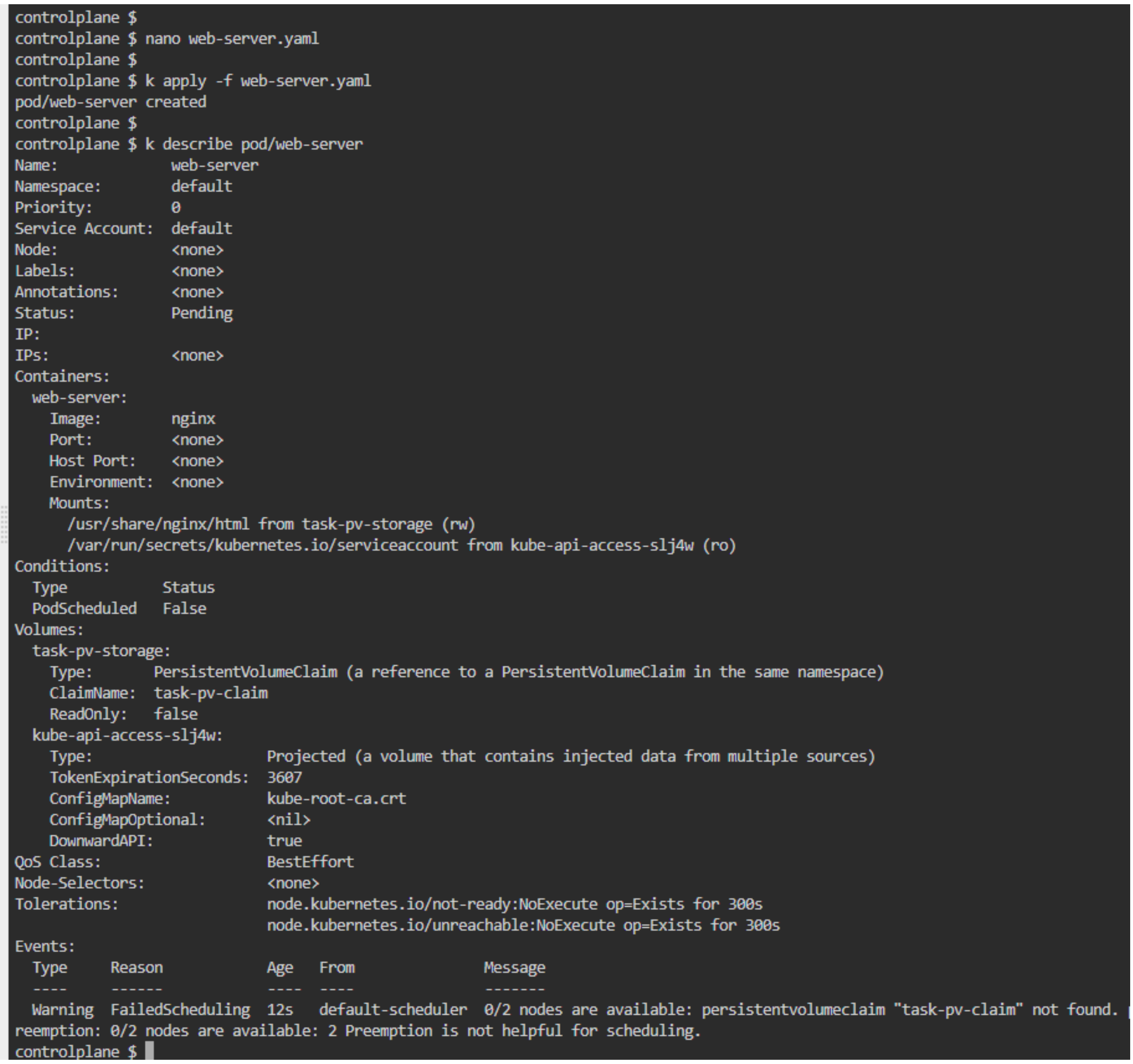
```
  GNU nano 4.8
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv
spec:
  storageClassName: csi-hostpath-sc
  capacity:
    storage: 10Mi
  accessModes:
    - ReadOnlyMany
  hostPath:
    path: "/usr/share/nginx/html"

---

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pv-volume
spec:
  storageClassName: csi-hostpath-sc
  accessModes:
    - ReadOnlyMany
  resources:
    requests:
      storage: 10Mi
```



```
controlplane $
controlplane $ nano pv-pvc.yaml
controlplane $
controlplane $ k apply -f pv-pvc.yaml
persistentvolume/pv created
persistentvolumeclaim/pv-volume created
controlplane $
controlplane $ k get pv
NAME   CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                STORAGECLASS      VOLUMEATTRIBUTESCLASS   REASON
 AGE
pv     10Mi       ROX            Retain           Bound    default/pv-volume    csi-hostpath-sc   <unset>
 3s
controlplane $ k get pvc
NAME        STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS      VOLUMEATTRIBUTESCLASS   AGE
pv-volume   Bound    pv       10Mi       ROX            csi-hostpath-sc   <unset>                 5s
controlplane $ █
```

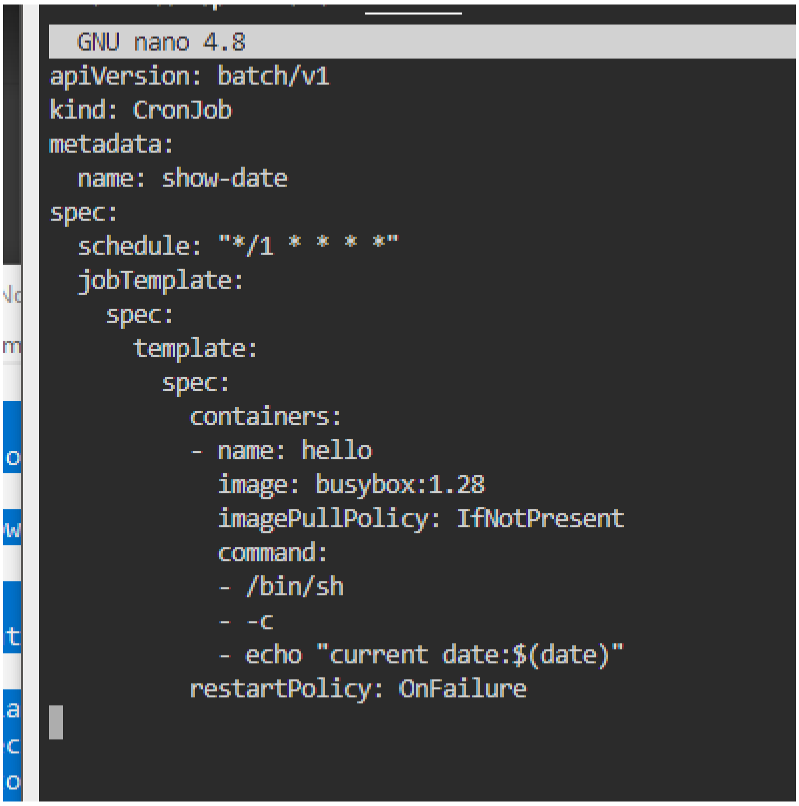


```
apiVersion: v1
kind: Pod
metadata:
  name: web-server
spec:
  containers:
    - name: web-server
      image: nginx
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: task-pv-storage
  volumes:
    - name: task-pv-storage
      persistentVolumeClaim:
        claimName: pv-volume
```

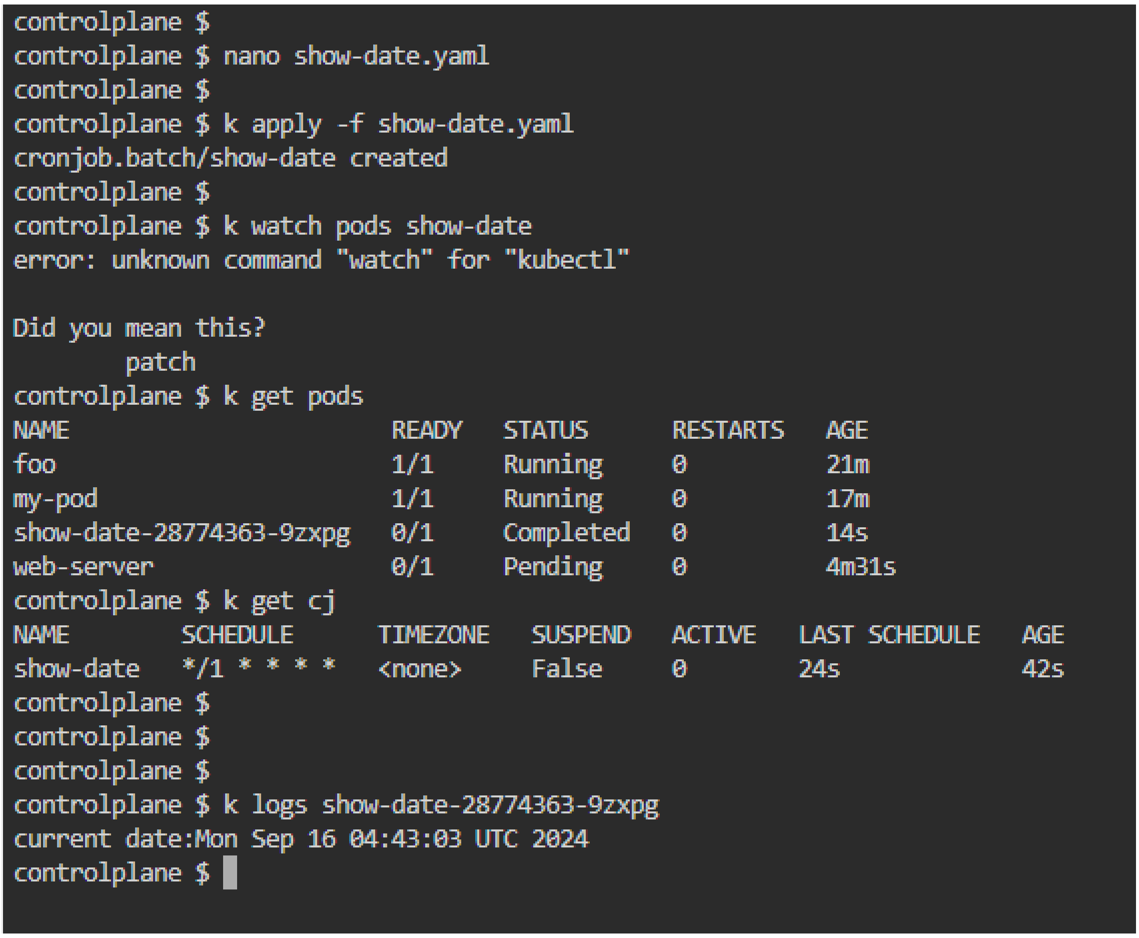


```
controlplane $
controlplane $ nano web-server.yaml
controlplane $
controlplane $ k apply -f web-server.yaml
pod/web-server created
controlplane $
controlplane $ k describe pod/web-server
Name:             web-server
Namespace:        default
Priority:         0
Service Account:  default
Node:             <none>
Labels:           <none>
Annotations:      <none>
Status:           Pending
IP:
IPs:              <none>
Containers:
  web-server:
    Image:        nginx
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:
      /usr/share/nginx/html from task-pv-storage (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-slj4w (ro)
Conditions:
  Type           Status
  PodScheduled   False
Volumes:
  task-pv-storage:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  task-pv-claim
    ReadOnly:   false
  kube-api-access-slj4w:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason            Age   From               Message
  ----     ------            ----  ----               -------
  Warning  FailedScheduling  12s   default-scheduler  0/2 nodes are available: persistentvolumeclaim "task-pv-claim" not found.
reemption: 0/2 nodes are available: 2 Preemption is not helpful for scheduling.
controlplane $
```

15)



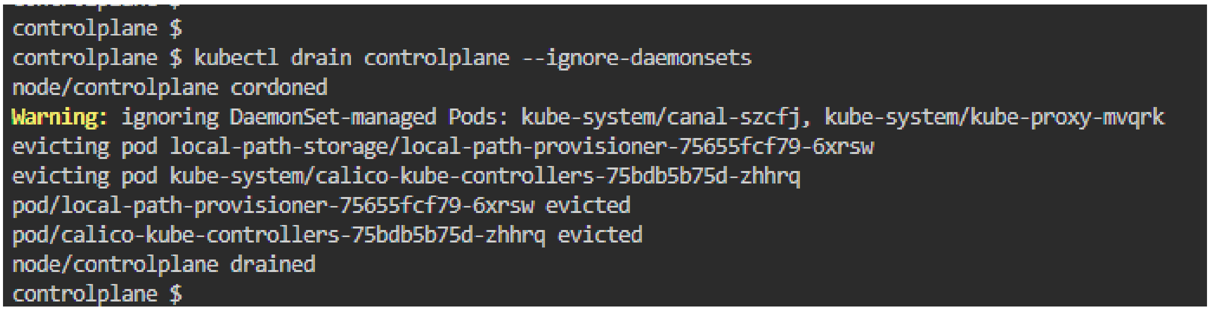
```
  GNU nano 4.8
apiVersion: batch/v1
kind: CronJob
metadata:
  name: show-date
spec:
  schedule: "*/1 * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: hello
            image: busybox:1.28
            imagePullPolicy: IfNotPresent
            command:
            - /bin/sh
            - -c
            - echo "current date:$(date)"
          restartPolicy: OnFailure
```



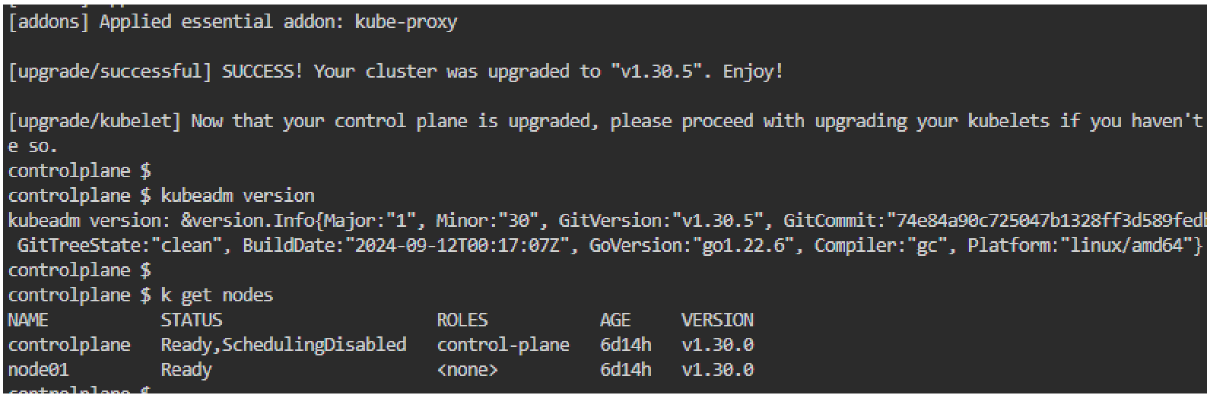
```
controlplane $
controlplane $ nano show-date.yaml
controlplane $
controlplane $ k apply -f show-date.yaml
cronjob.batch/show-date created
controlplane $
controlplane $ k watch pods show-date
error: unknown command "watch" for "kubectl"

Did you mean this?
        patch
controlplane $ k get pods
NAME                         READY   STATUS      RESTARTS   AGE
foo                          1/1     Running     0          21m
my-pod                       1/1     Running     0          17m
show-date-28774363-9zxpg     0/1     Completed   0          14s
web-server                   0/1     Pending     0          4m31s
controlplane $ k get cj
NAME        SCHEDULE      TIMEZONE   SUSPEND   ACTIVE   LAST SCHEDULE   AGE
show-date   */1 * * * *   <none>     False     0        24s             42s
controlplane $
controlplane $
controlplane $
controlplane $ k logs show-date-28774363-9zxpg
current date:Mon Sep 16 04:43:03 UTC 2024
controlplane $
```

16)

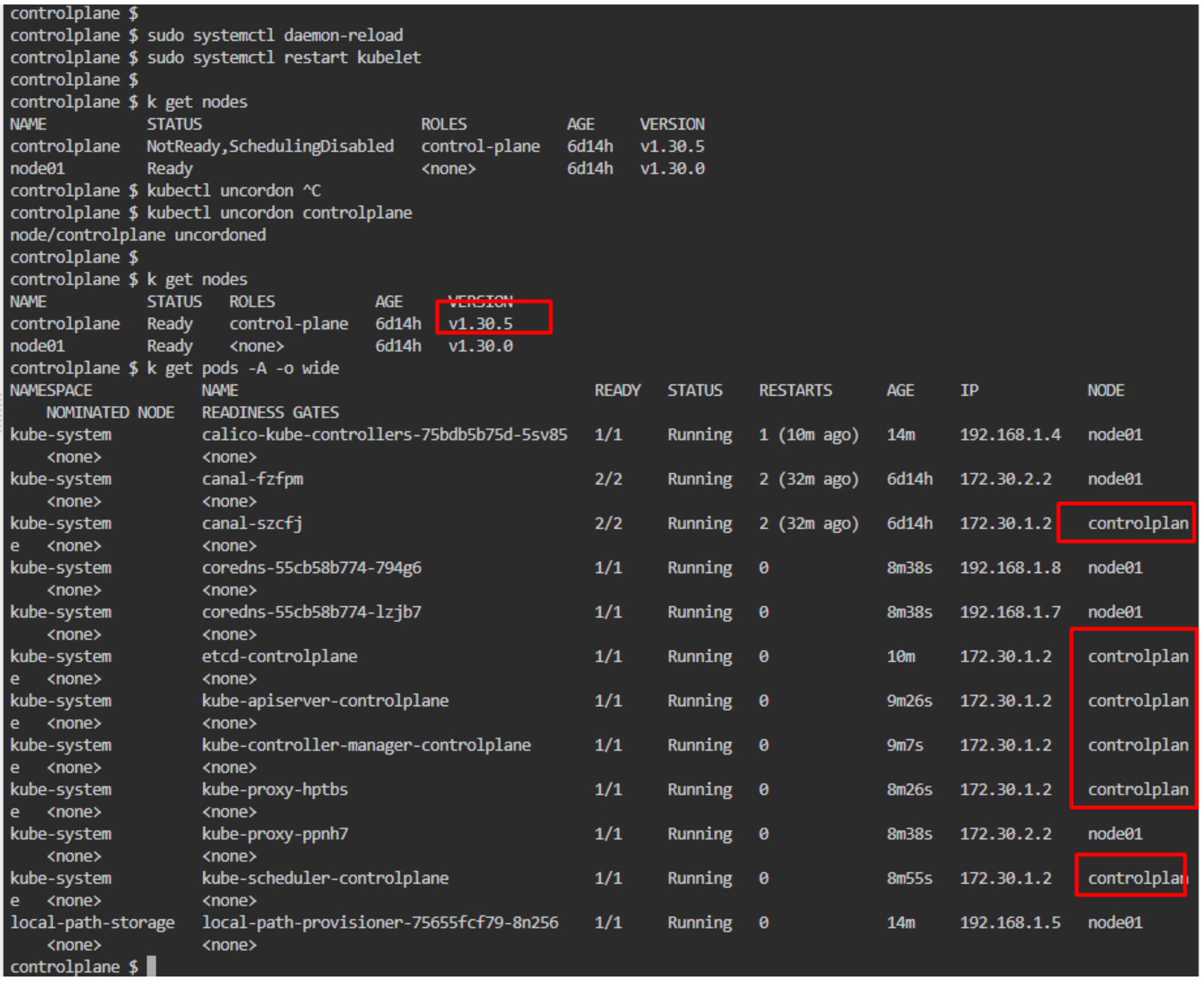
Assuming, **master =controlplane**



```
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/local-path-provisioner-75655fcf79-6xrsw evicted
pod/calico-kube-controllers-75bdb5b75d-zhhrq evicted
node/controlplane drained
controlplane $
```

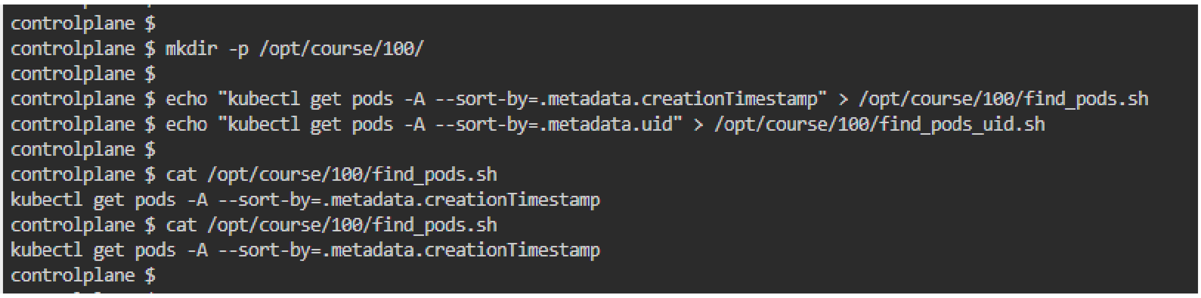


```
[addons] Applied essential addon: kube-proxy

[upgrade/successful] SUCCESS! Your cluster was upgraded to "v1.30.5". Enjoy!

[upgrade/kubelet] Now that your control plane is upgraded, please proceed with upgrading your kubelets if you haven't
e so.
controlplane $
controlplane $ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"30", GitVersion:"v1.30.5", GitCommit:"74e84a90c725047b1328ff3d589fed
 GitTreeState:"clean", BuildDate:"2024-09-12T00:17:07Z", GoVersion:"go1.22.6", Compiler:"gc", Platform:"linux/amd64"}
controlplane $
controlplane $ k get nodes
NAME           STATUS                     ROLES           AGE     VERSION
controlplane   Ready,SchedulingDisabled   control-plane   6d14h   v1.30.0
node01         Ready                      <none>          6d14h   v1.30.0
```

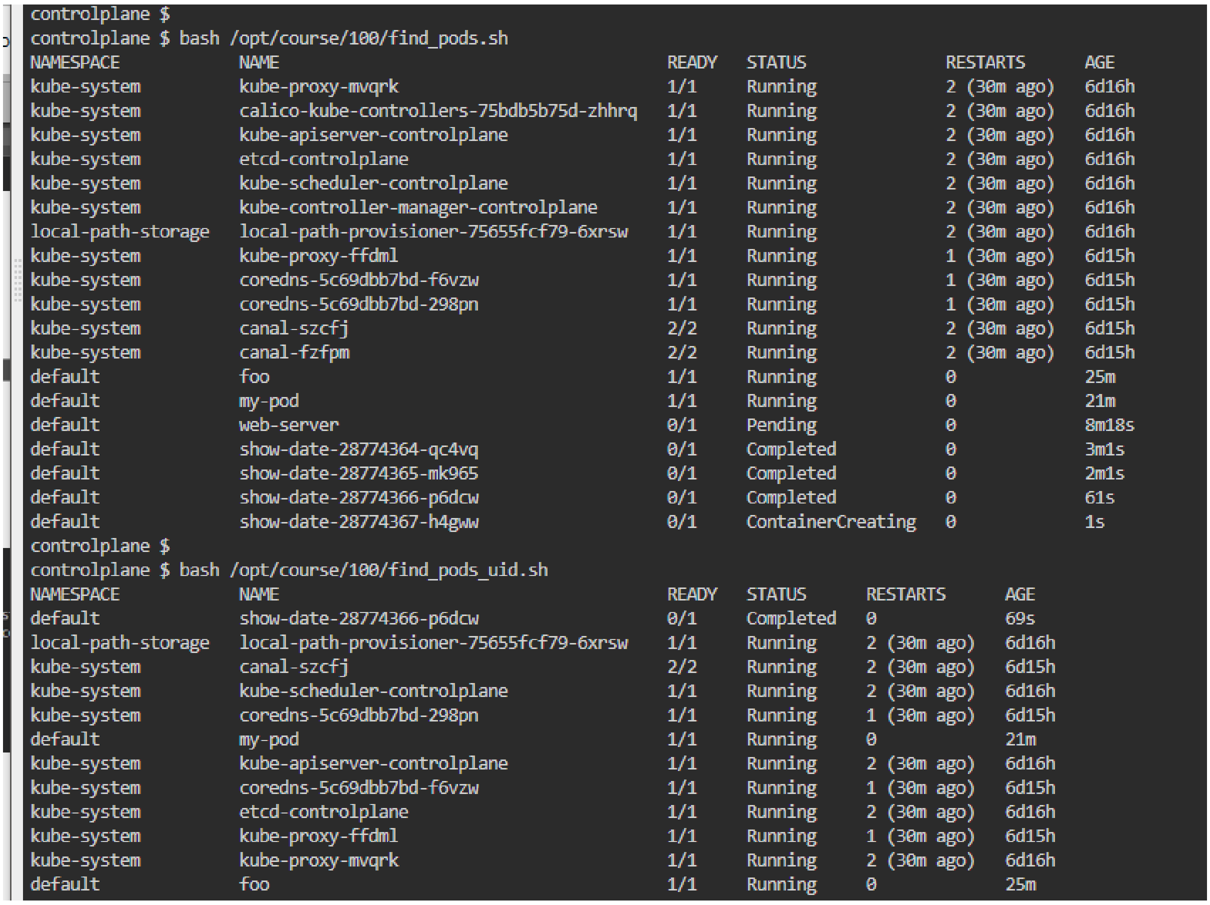


```
controlplane $
controlplane $ sudo systemctl daemon-reload
controlplane $ sudo systemctl restart kubelet
controlplane $
controlplane $ k get nodes
NAME           STATUS                        ROLES           AGE     VERSION
controlplane   NotReady,SchedulingDisabled   control-plane   6d14h   v1.30.5
node01         Ready                         <none>          6d14h   v1.30.0
controlplane $ kubectl uncordon ^C
controlplane $ kubectl uncordon controlplane
node/controlplane uncordoned
controlplane $
controlplane $ k get nodes
NAME           STATUS   ROLES           AGE     VERSION
controlplane   Ready    control-plane   6d14h   v1.30.5
node01         Ready    <none>          6d14h   v1.30.0
controlplane $ k get pods -A -o wide
NAMESPACE            NAME                                       READY   STATUS    RESTARTS       AGE     IP            NODE
     NOMINATED NODE   READINESS GATES
kube-system          calico-kube-controllers-75bdb5b75d-5sv85   1/1     Running   1 (10m ago)    14m     192.168.1.4   node01
     <none>           <none>
kube-system          canal-fzfpm                                2/2     Running   2 (32m ago)    6d14h   172.30.2.2    node01
     <none>           <none>
kube-system          canal-szcfj                                2/2     Running   2 (32m ago)    6d14h   172.30.1.2    controlplan
e    <none>           <none>
kube-system          coredns-55cb58b774-794g6                   1/1     Running   0              8m38s   192.168.1.8   node01
     <none>           <none>
kube-system          coredns-55cb58b774-lzjb7                   1/1     Running   0              8m38s   192.168.1.7   node01
     <none>           <none>
kube-system          etcd-controlplane                          1/1     Running   0              10m     172.30.1.2    controlplan
e    <none>           <none>
kube-system          kube-apiserver-controlplane                1/1     Running   0              9m26s   172.30.1.2    controlplan
e    <none>           <none>
kube-system          kube-controller-manager-controlplane       1/1     Running   0              9m7s    172.30.1.2    controlplan
e    <none>           <none>
kube-system          kube-proxy-hptbs                           1/1     Running   0              8m26s   172.30.1.2    controlplan
e    <none>           <none>
kube-system          kube-proxy-ppnh7                           1/1     Running   0              8m38s   172.30.2.2    node01
     <none>           <none>
kube-system          kube-scheduler-controlplane                1/1     Running   0              8m55s   172.30.1.2    controlplan
e    <none>           <none>
local-path-storage   local-path-provisioner-75655fcf79-8n256    1/1     Running   0              14m     192.168.1.5   node01
     <none>           <none>
controlplane $
```

17)



```
controlplane $
controlplane $ mkdir -p /opt/course/100/
controlplane $
controlplane $ echo "kubectl get pods -A --sort-by=.metadata.creationTimestamp" > /opt/course/100/find_pods.sh
controlplane $ echo "kubectl get pods -A --sort-by=.metadata.uid" > /opt/course/100/find_pods_uid.sh
controlplane $
controlplane $ cat /opt/course/100/find_pods.sh
kubectl get pods -A --sort-by=.metadata.creationTimestamp
controlplane $ cat /opt/course/100/find_pods.sh
kubectl get pods -A --sort-by=.metadata.creationTimestamp
controlplane $
```



```
controlplane $
controlplane $ bash /opt/course/100/find_pods.sh
NAMESPACE            NAME                                       READY   STATUS              RESTARTS      AGE
kube-system          kube-proxy-mvqrk                           1/1     Running             2 (30m ago)   6d16h
kube-system          calico-kube-controllers-75bdb5b75d-zhhrq   1/1     Running             2 (30m ago)   6d16h
kube-system          kube-apiserver-controlplane                1/1     Running             2 (30m ago)   6d16h
kube-system          etcd-controlplane                          1/1     Running             2 (30m ago)   6d16h
kube-system          kube-scheduler-controlplane                1/1     Running             2 (30m ago)   6d16h
kube-system          kube-controller-manager-controlplane       1/1     Running             2 (30m ago)   6d16h
local-path-storage   local-path-provisioner-75655fcf79-6xrsw    1/1     Running             2 (30m ago)   6d16h
kube-system          kube-proxy-ffdml                           1/1     Running             1 (30m ago)   6d15h
kube-system          coredns-5c69dbb7bd-f6vzw                   1/1     Running             1 (30m ago)   6d15h
kube-system          coredns-5c69dbb7bd-298pn                   1/1     Running             1 (30m ago)   6d15h
kube-system          canal-szcfj                                2/2     Running             2 (30m ago)   6d15h
kube-system          canal-fzfpm                                2/2     Running             2 (30m ago)   6d15h
default              foo                                        1/1     Running             0             25m
default              my-pod                                     1/1     Running             0             21m
default              web-server                                 0/1     Pending             0             8m18s
default              show-date-28774364-qc4vq                   0/1     Completed           0             3m1s
default              show-date-28774365-mk965                   0/1     Completed           0             2m1s
default              show-date-28774366-p6dcw                   0/1     Completed           0             61s
default              show-date-28774367-h4gww                   0/1     ContainerCreating   0             1s
controlplane $
controlplane $ bash /opt/course/100/find_pods_uid.sh
NAMESPACE            NAME                                       READY   STATUS      RESTARTS      AGE
default              show-date-28774366-p6dcw                   0/1     Completed   0             69s
local-path-storage   local-path-provisioner-75655fcf79-6xrsw    1/1     Running     2 (30m ago)   6d16h
kube-system          canal-szcfj                                2/2     Running     2 (30m ago)   6d15h
kube-system          kube-scheduler-controlplane                1/1     Running     2 (30m ago)   6d16h
kube-system          coredns-5c69dbb7bd-298pn                   1/1     Running     1 (30m ago)   6d15h
default              my-pod                                     1/1     Running     0             21m
kube-system          kube-apiserver-controlplane                1/1     Running     2 (30m ago)   6d16h
kube-system          coredns-5c69dbb7bd-f6vzw                   1/1     Running     1 (30m ago)   6d15h
kube-system          etcd-controlplane                          1/1     Running     2 (30m ago)   6d16h
kube-system          kube-proxy-ffdml                           1/1     Running     1 (30m ago)   6d15h
kube-system          kube-proxy-mvqrk                           1/1     Running     2 (30m ago)   6d16h
default              foo                                        1/1     Running     0             25m
```