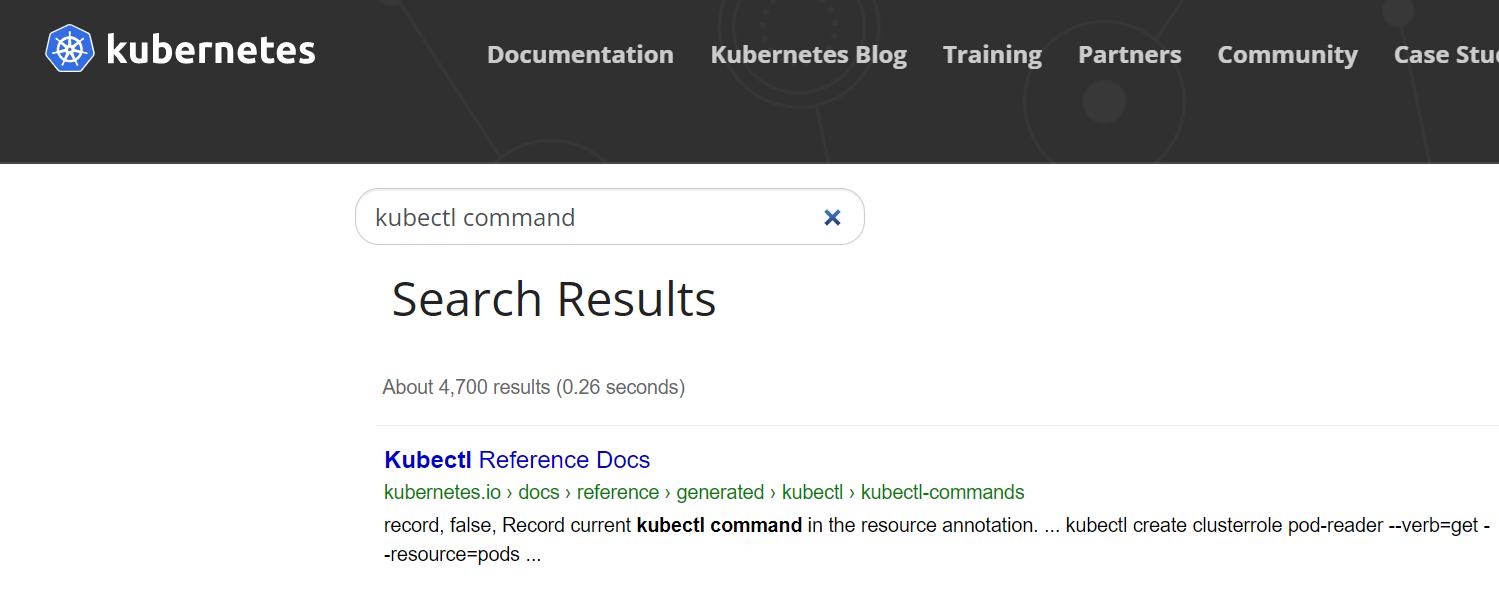
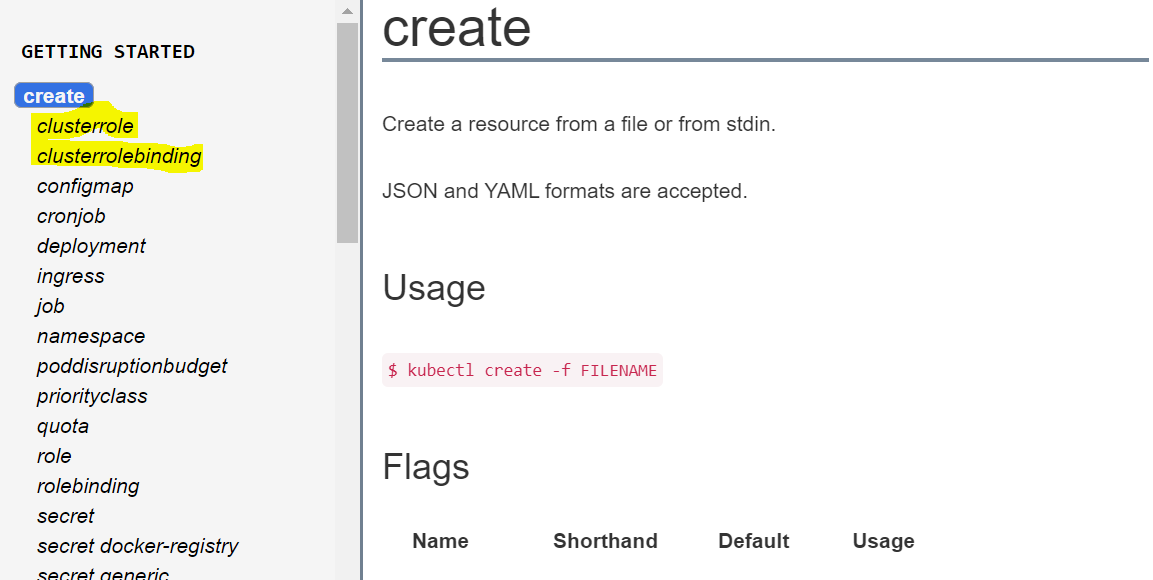
1. Create a Cluster Role and create a service account in a specific namespace. After that create the cluster role binding using that.

Search kubectl command in doc page.



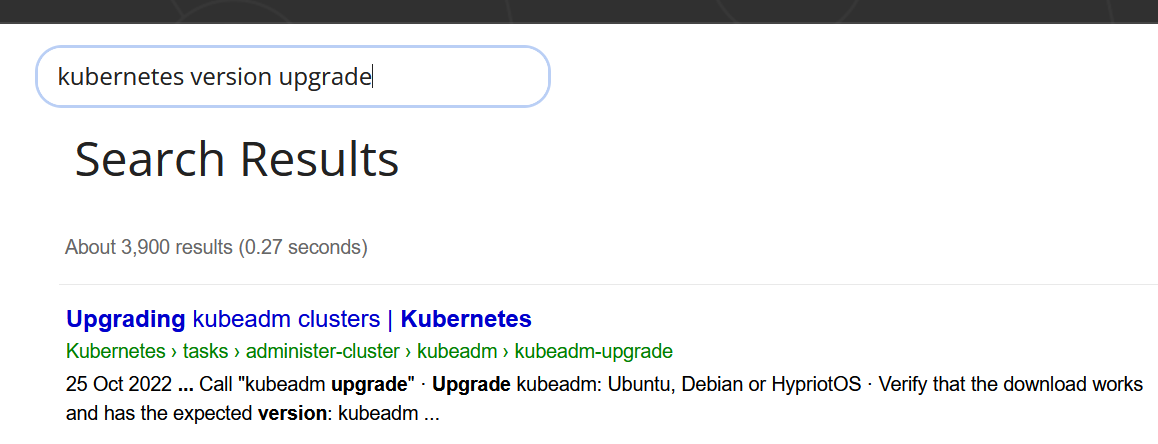


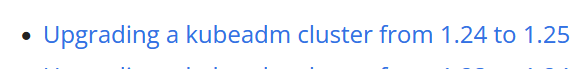
kubectl create clusterrole pod-reader **--verb**=get,list,watch **--resource**=pods

kubectl create serviceaccount <serviceAccountname> --namespace=app

kubectl create clusterrolebinding cluster-admin **--clusterrole**=cluster-admin **–serviceaccount=app:serviceaccountname**

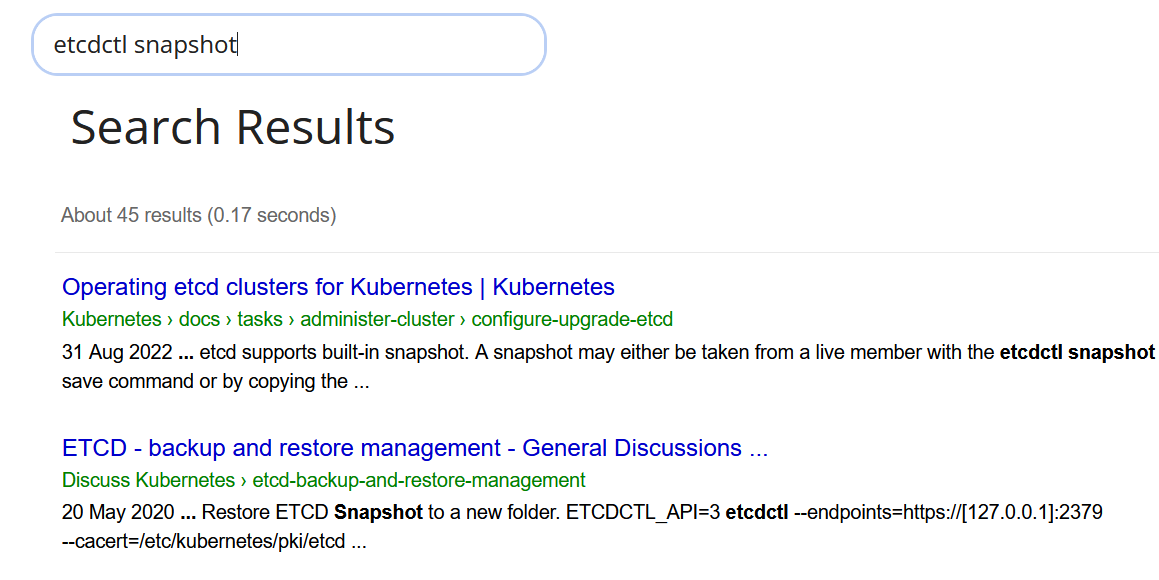
1. Kubectl drain <nodename> --ignore-daemonsets
2. Kubernetes version upgrade





Follow the above link. We will get request to upgrade only master node , drain and uncordon the node without fail.

1. Etcdctl snapshot and restore



Backup Command

ETCDCTL\_API=3 etcdctl --endpoints=https://127.0.0.1:2379 **\**

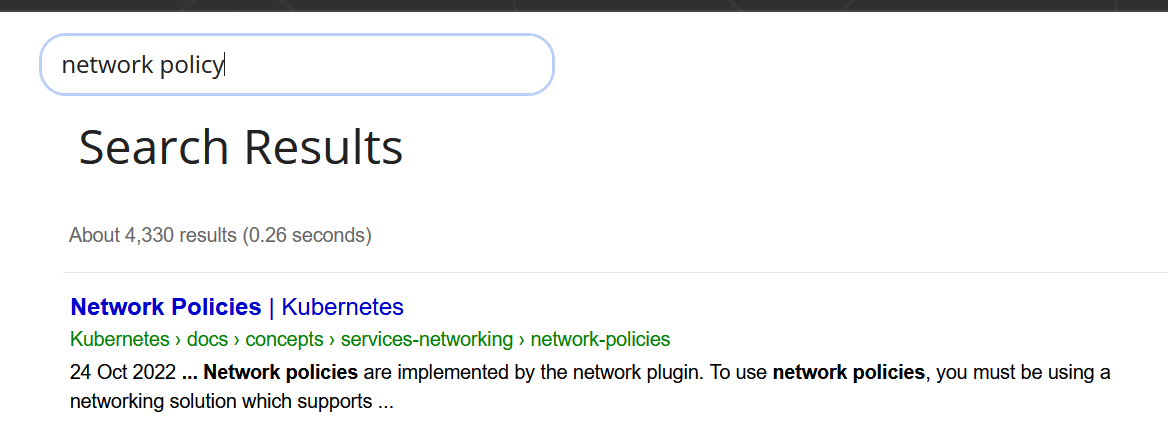
--cacert=<trusted-ca-file> --cert=<cert-file> --key=<key-file> **\**

snapshot save <backup-file-location>

Restore Command

ETCDCTL\_API=3 etcdctl --endpoints 10.2.0.9:2379 snapshot restore snapshotdb

1. Network Policy to allow the pods from a particular namespace



**apiVersion**: networking.k8s.io/v1

**kind**: NetworkPolicy

**metadata**:

**name**: test-network-policy

**namespace**: default

**spec**:

**podSelector**: {}

**policyTypes**:

- Ingress

**ingress**:

- **from**:

- **namespaceSelector**:

**matchLabels**:

**namespace**: <givennamespacein question>

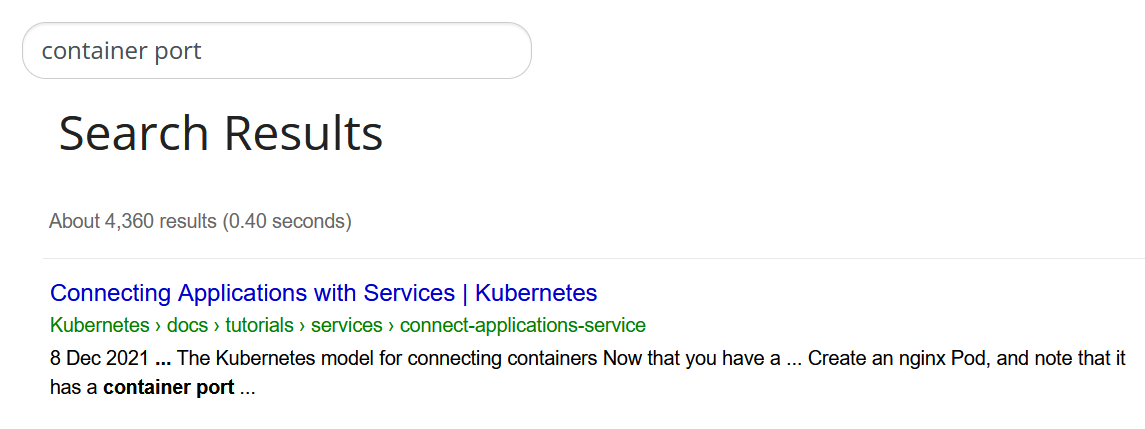
**ports**:

- **protocol**: TCP

**port**: 5978

1. Edit the deployment and set the container port and then create the nodeport service for the same deployment using kubectl expose.

Kubectl edit deployment and set the container port



First example as reference

**apiVersion**: apps/v1

**kind**: Deployment

**metadata**:

**name**: my-nginx

**spec**:

**selector**:

**matchLabels**:

**run**: my-nginx

**replicas**: 2

**template**:

**metadata**:

**labels**:

**run**: my-nginx

**spec**:

**containers**:

- **name**: my-nginx

**image**: nginx

**ports**:

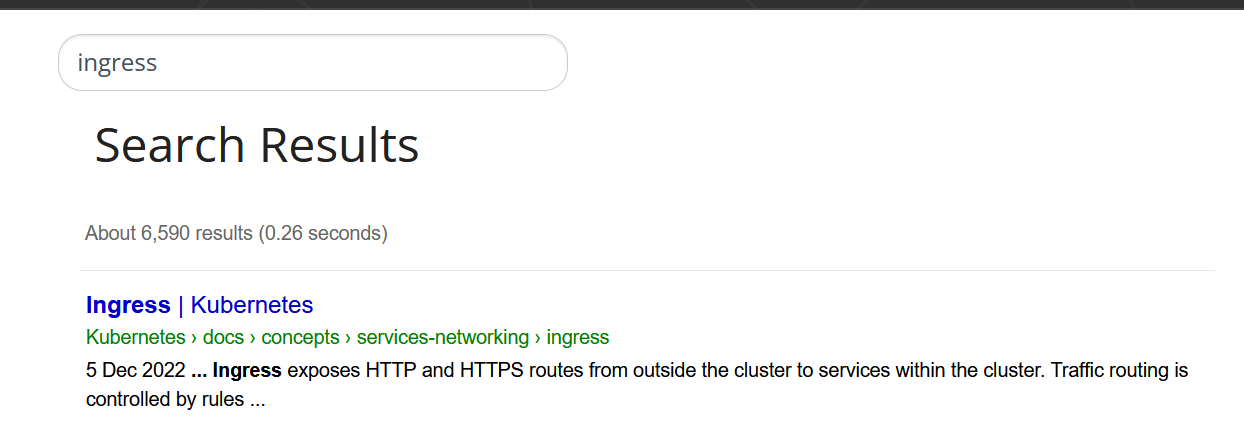
- **containerPort**: <givenportnumber>

**name: <givenname>**

**protocol: TCP**

check the kubectl expose command to create the nodeport service

1. Create the ingress for the given service



**apiVersion**: networking.k8s.io/v1

**kind**: Ingress

**metadata**:

**name**: minimal-ingress

**annotations**:

**nginx.ingress.kubernetes.io/rewrite-target**: /

**spec**:

**ingressClassName**: nginx-example

**rules**:

- **http**:

**paths**:

- **path**: /testpath

**pathType**: Prefix

**backend**:

**service**:

**name**: test

**port**:

**number**: 80

1. Scale the deployment replicas.

Kubectl scale deployment <deploymentname> --replicas=5

1. Create a deployment with nodeselector.
2. Create a multicontainer pod with 2 container and 2 container image.

**containers**:

- **name**: nginx

**image**: nginx

- **name**: redis

**image**: redis

1. Find an error from the pod logs.

Kubectl logs <podname> | grep “<given word>”

1. Create a pod with sidecar container to save the logs of the Pod
2. Troubleshooting question.

Verify the status of the kubelet . Kubelet will be stopped . Need to start the kubelet service.

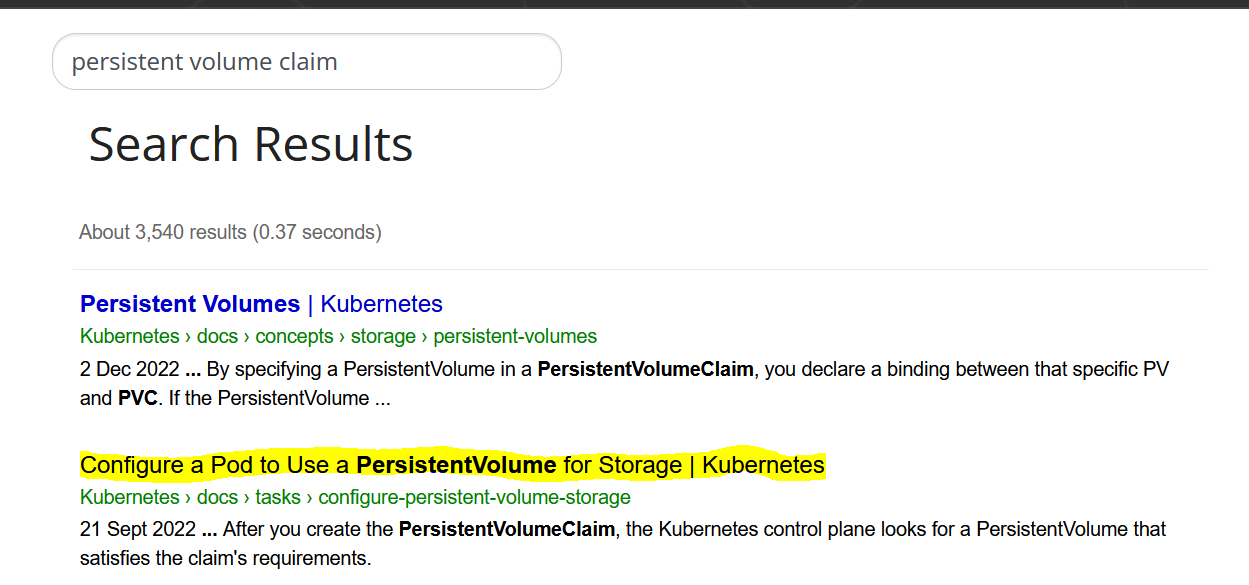
Systemctl status kubelet (if it is not started)

Systemctl start kubelet (start the service)

1. Create a persistent volume.

Either storage class they will provide (or) they will give the hostpath details

Note: Search Persistent Volume Claim in Kubernetes Documentation



**apiVersion**: v1

**kind**: PersistentVolume

**metadata**:

**name**: task-pv-volume

**labels**:

**type**: local

**spec**:

**storageClassName**: manual

**capacity**:

**storage**: 10Gi

**accessModes**:

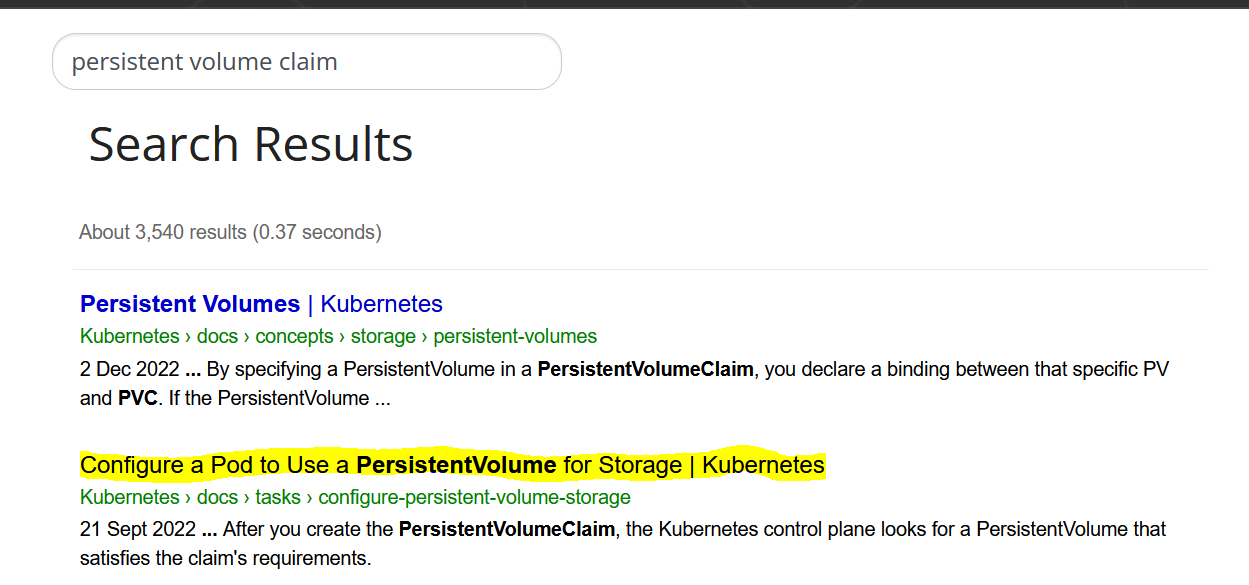
- ReadWriteOnce

**hostPath**:

**path**: "/mnt/data"

1. Create a persistent volume claim and configure a pod to use the PVC.

Note : In question we may have request to increase the storage value .



**apiVersion**: v1

**kind**: PersistentVolumeClaim

**metadata**:

**name**: task-pv-claim

**spec**:

**storageClassName**: manual

**accessModes**:

- ReadWriteOnce

**resources**:

**requests**:

**storage**: 3Gi

Configure a pod using the PVC

**apiVersion**: v1

**kind**: Pod

**metadata**:

**name**: task-pv-pod

**spec**:

**volumes**:

- **name**: task-pv-storage

**persistentVolumeClaim**:

**claimName**: task-pv-claim

**containers**:

- **name**: task-pv-container

**image**: nginx

**ports**:

- **containerPort**: 80

**name**: "http-server"

**volumeMounts**:

- **mountPath**: "/usr/share/nginx/html"

**name**: task-pv-storage

1. Need to find a pod which is running with high CPU and we may need to search in pods which is running with certain label.

Use : Kubectl top command with sort by CPU and remove the headers and make that one pod saved in the mentioned file path

1. Need to find the list of nodes which are in ready state using json query from kubectl get nodes result.