ClusterIP Service (Default)

This is the default service type. It only exposes the service within the Kubernetes cluster.

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx  # Service selects Pods with this label
  ports:
    - protocol: TCP
      port: 80        # Port that the service exposes
      targetPort: 80   # Port on the Pod
  type: ClusterIP  # This is the default type (optional)
```

NodePort Service

Exposes the service on a static port on each node's IP, allowing access from outside the cluster.

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-nodeport
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80          # Service Port
      targetPort: 80     # Pod Port
      nodePort: 30001    # Static port exposed on each node
  type: NodePort       # Exposed externally
```

📄 3. LoadBalancer Service

Exposes the service externally via a load balancer (usually cloud-specific).

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-loadbalancer
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80          # Service Port
      targetPort: 80     # Pod Port
  type: LoadBalancer   # Exposed externally via a load balancer
```

📄 4. ExternalName Service

Maps the service to an external DNS name, useful for referring to services outside the cluster.

```
apiVersion: v1
kind: Service
metadata:
  name: external-service
spec:
  ports:
    - port: 80
  externalName: example.com  # External DNS name to map the service to
  type: ExternalName         # This type doesn't need selectors
```

```
# Services commands

# 1. Create a Service from a YAML file
kubectl apply -f service.yaml

# 2. List all Services in the current namespace
kubectl get services

# 3. Get detailed information about a specific Service
kubectl describe service <service-name>

# 4. Expose a Pod as a Service (e.g., exposing port 80)
kubectl expose pod <pod-name> --port=80 --target-port=80 --name=<service-name>

# 5. Expose a Deployment as a Service
kubectl expose deployment <deployment-name> --port=80 --target-port=80 --name=<service-name>

# 6. Scale a Deployment (change the number of replicas)
kubectl scale deployment <deployment-name> --replicas=5

# 7. Delete a Service
kubectl delete service <service-name>

# 8. Get the ClusterIP of a specific Service
kubectl get service <service-name> -o wide

# 9. Watch for the External IP of a LoadBalancer Service
kubectl get service <service-name> --watch
```