Kubernetes through Minikube

- K8s.
- Kubernetes is a container management tool.
- Why does the kubernetes logo have 7 sticks?
  Because google made a project named “project seven”.

Step :

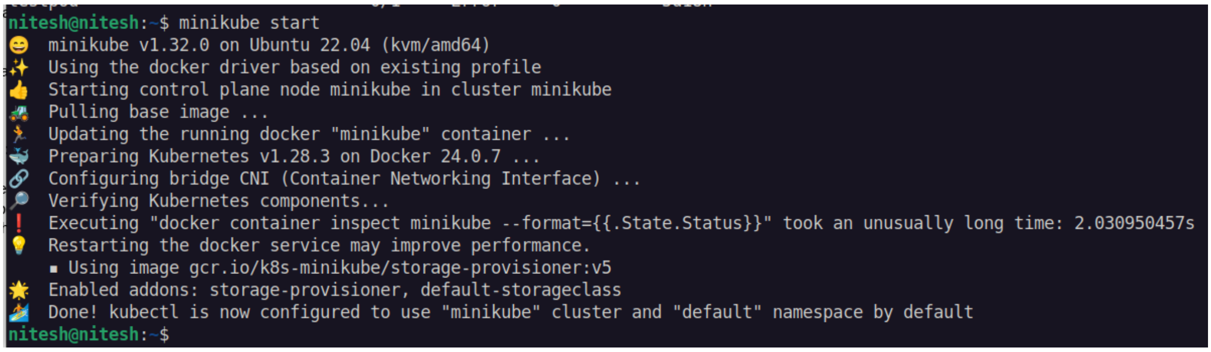
1) Install docker

Basic Kubernetes concepts like pods, services, and deployments.

How to deploy and manage applications on Minikube.

Command : minikube start

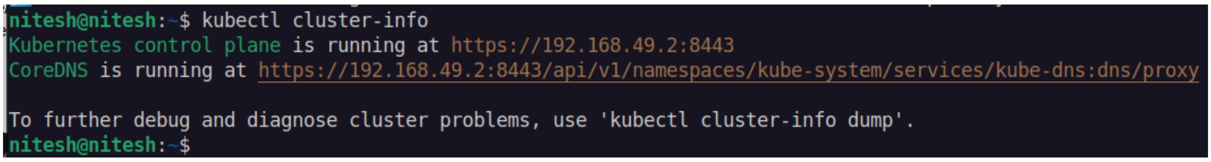
```
nitesh@nitesh:~$ minikube start
😄  minikube v1.32.0 on Ubuntu 22.04 (kvm/amd64)
✨  Using the docker driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
nitesh@nitesh:~$
```



```
nitesh@nitesh:~$ minikube start
😄  minikube v1.32.0 on Ubuntu 22.04 (kvm/amd64)
✨  Using the docker driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🏃  Updating the running docker "minikube" container ...
🐳  Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
❗  Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 2.030950457s
💡  Restarting the docker service may improve performance.
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
nitesh@nitesh:~$
```

Command : kubectl cluster-info

```
nitesh@nitesh:~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
nitesh@nitesh:~$
```



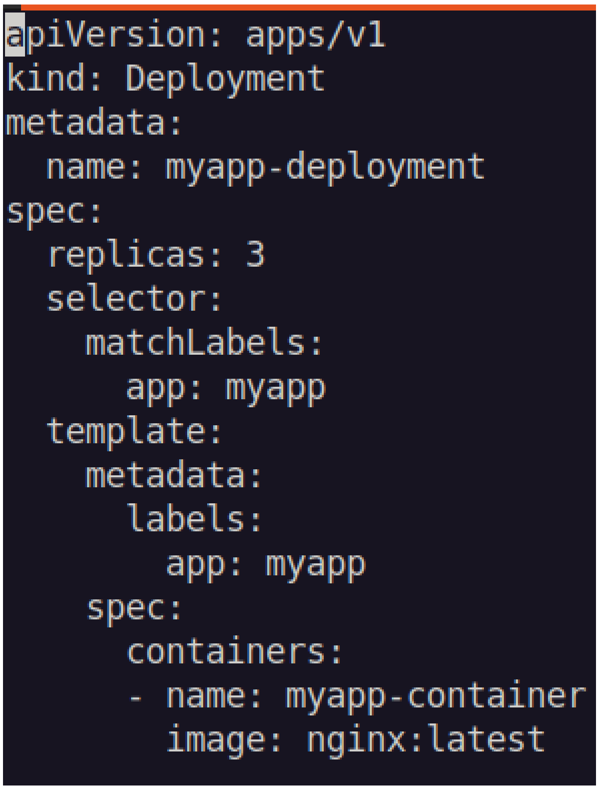
```
nitesh@nitesh:~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
nitesh@nitesh:~$
```

Create deployment yaml file
Command : vi app-deployment.yaml

Press i for “Write the following script”

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: myapp-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: myapp
  template:
    metadata:
      labels:
        app: myapp
    spec:
      containers:
      - name: myapp-container
        image: nginx:latest
```

Press “Esc :wq” to save



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: myapp-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: myapp
  template:
    metadata:
      labels:
        app: myapp
    spec:
      containers:
      - name: myapp-container
        image: nginx:latest
```

Command :kubectl apply -f app-deployment.yaml

kubectl : This is the command-line tool for interacting with Kubernetes clusters.

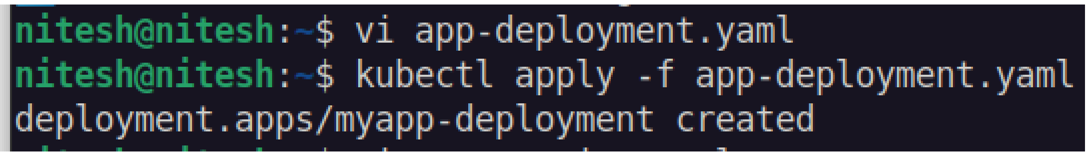
apply : This subcommand is used to apply a configuration to a resource. It's commonly used to create, update, or delete resources described in a YAML or JSON file.

-f : This flag is followed by the filename or URL of the resource configuration file in YAML or JSON format. It specifies the file to be applied to the cluster.

app-deployment.yaml **:** This is the YAML file that contains the specifications for your application deployment, such as the container image, replicas, ports, etc.

app-deployment.yaml : It is the deployment file name.

```
nitesh@nitesh:~$ vi app-deployment.yaml
nitesh@nitesh:~$ kubectl apply -f app-deployment.yaml
deployment.apps/myapp-deployment created
nitesh@nitesh:~$
```



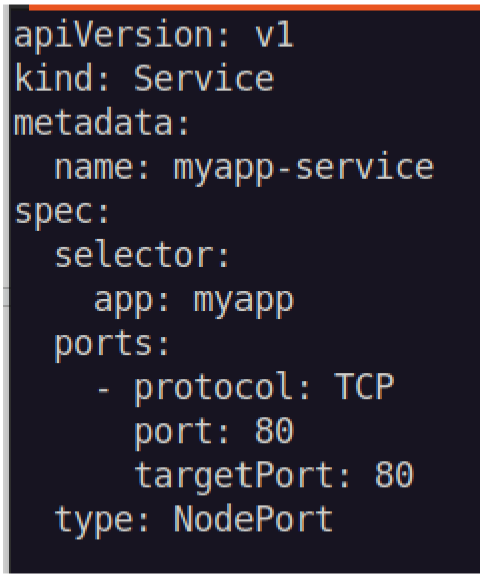
```
nitesh@nitesh:~$ vi app-deployment.yaml
nitesh@nitesh:~$ kubectl apply -f app-deployment.yaml
deployment.apps/myapp-deployment created
```

Create service.yaml

Press i for “Write the following script”

```
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  selector:
    app: myapp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: NodePort
```

Press “Esc :wq” to save



```
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  selector:
    app: myapp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: NodePort
```

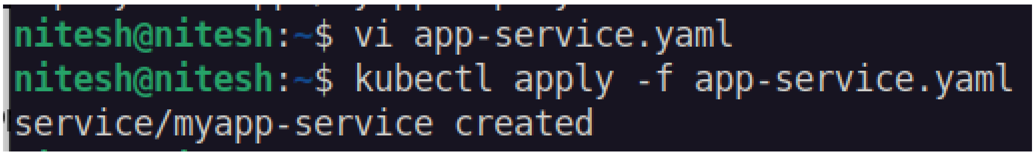
Command : kubectl apply -f app-service.yaml

kubectl : This is the command-line tool for interacting with Kubernetes clusters.

apply : This subcommand is used to apply a configuration to a resource. It's commonly used to create, update, or delete resources described in a YAML or JSON file.

-f : This flag is followed by the filename or URL of the resource configuration file in YAML or JSON format. It specifies the file to be applied to the cluster.

app-service.yaml : This is the YAML file that contains the configuration for your Kubernetes service. It specifies how the service should be set up, including details like the service type (NodePort, LoadBalancer, ClusterIP), ports, selectors to route traffic to pods, etc.

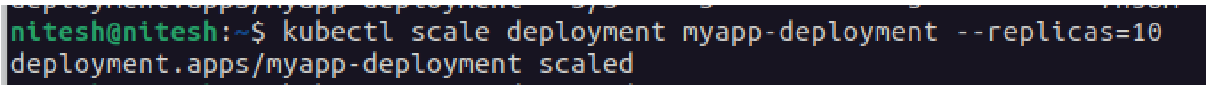


```
nitesh@nitesh:~$ vi app-service.yaml
nitesh@nitesh:~$ kubectl apply -f app-service.yaml
service/myapp-service created
```

minikube service myapp-service

kubectl scale deployment myapp-deployment --replicas=10

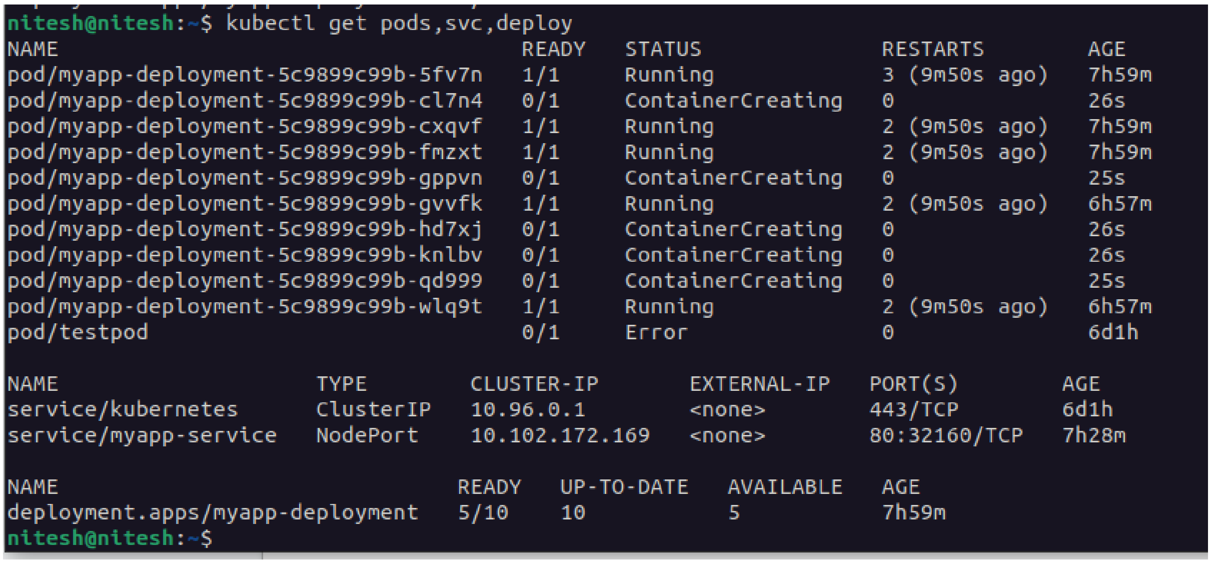
```
nitesh@nitesh:~$ kubectl scale deployment myapp-deployment --replicas=10
deployment.apps/myapp-deployment scaled
nitesh@nitesh:~$
```



```
nitesh@nitesh:~$ kubectl scale deployment myapp-deployment --replicas=10
deployment.apps/myapp-deployment scaled
```

kubectl get pods,svc,deploy

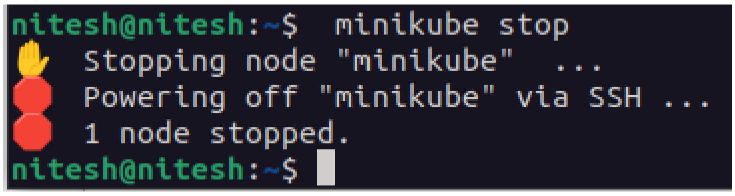
```
nitesh@nitesh:~$ kubectl get pods,svc,deploy
NAME                                READY   STATUS              RESTARTS     AGE
pod/myapp-deployment-5c9899c99b-5fv7n   1/1     Running             3 (9m50s ago)   7h59m
pod/myapp-deployment-5c9899c99b-cl7n4   0/1     ContainerCreating   0               26s
pod/myapp-deployment-5c9899c99b-cxqvf   1/1     Running             2 (9m50s ago)   7h59m
pod/myapp-deployment-5c9899c99b-fmzxt   1/1     Running             2 (9m50s ago)   7h59m
pod/myapp-deployment-5c9899c99b-gppvn   0/1     ContainerCreating   0               25s
pod/myapp-deployment-5c9899c99b-gvvfk   1/1     Running             2 (9m50s ago)   6h57m
pod/myapp-deployment-5c9899c99b-hd7xj   0/1     ContainerCreating   0               26s
pod/myapp-deployment-5c9899c99b-knlbv   0/1     ContainerCreating   0               26s
pod/myapp-deployment-5c9899c99b-qd999   0/1     ContainerCreating   0               25s
pod/myapp-deployment-5c9899c99b-wlq9t   1/1     Running             2 (9m50s ago)   6h57m
pod/testpod                         0/1     Error               0               6d1h

NAME                  TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)
AGE
service/kubernetes      ClusterIP   10.96.0.1        <none>        443/TCP        6d1h
service/myapp-service   NodePort    10.102.172.169   <none>        80:32160/TCP   7h28m

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/myapp-deployment   5/10    10           5           7h59m
nitesh@nitesh:~$
```



```
nitesh@nitesh:~$ kubectl get pods,svc,deploy
NAME                                      READY   STATUS              RESTARTS       AGE
pod/myapp-deployment-5c9899c99b-5fv7n     1/1     Running             3 (9m50s ago)  7h59m
pod/myapp-deployment-5c9899c99b-cl7n4     0/1     ContainerCreating   0              26s
pod/myapp-deployment-5c9899c99b-cxqvf     1/1     Running             2 (9m50s ago)  7h59m
pod/myapp-deployment-5c9899c99b-fmzxt     1/1     Running             2 (9m50s ago)  7h59m
pod/myapp-deployment-5c9899c99b-gppvn     0/1     ContainerCreating   0              25s
pod/myapp-deployment-5c9899c99b-gvvfk     1/1     Running             2 (9m50s ago)  6h57m
pod/myapp-deployment-5c9899c99b-hd7xj     0/1     ContainerCreating   0              26s
pod/myapp-deployment-5c9899c99b-knlbv     0/1     ContainerCreating   0              26s
pod/myapp-deployment-5c9899c99b-qd999     0/1     ContainerCreating   0              25s
pod/myapp-deployment-5c9899c99b-wlq9t     1/1     Running             2 (9m50s ago)  6h57m
pod/testpod                               0/1     Error               0              6d1h

NAME                       TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
service/kubernetes         ClusterIP   10.96.0.1        <none>        443/TCP        6d1h
service/myapp-service      NodePort    10.102.172.169   <none>        80:32160/TCP   7h28m

NAME                               READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/myapp-deployment   5/10    10           5           7h59m
nitesh@nitesh:~$
```

minikube stop

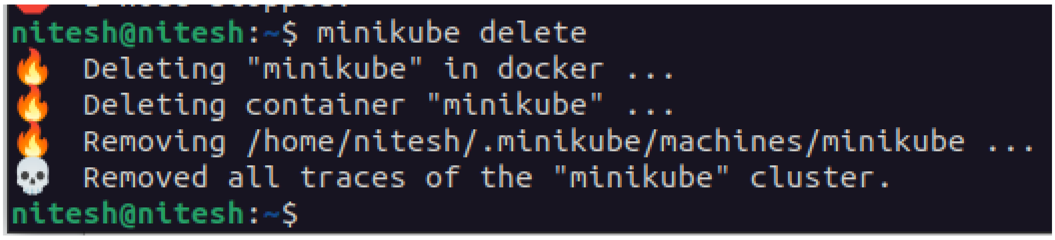
```
nitesh@nitesh:~$  minikube stop
✋  Stopping node "minikube"  ...
🛑  Powering off "minikube" via SSH ...
🛑  1 node stopped.
nitesh@nitesh:~$
```



```
nitesh@nitesh:~$  minikube stop
✋  Stopping node "minikube"  ...
🛑  Powering off "minikube" via SSH ...
🛑  1 node stopped.
nitesh@nitesh:~$
```

minikube delete

```
nitesh@nitesh:~$ minikube delete
🔥  Deleting "minikube" in docker ...
🔥  Deleting container "minikube" ...
🔥  Removing /home/nitesh/.minikube/machines/minikube ...
💀  Removed all traces of the "minikube" cluster.
nitesh@nitesh:~$
```



```
nitesh@nitesh:~$ minikube delete
🔥  Deleting "minikube" in docker ...
🔥  Deleting container "minikube" ...
🔥  Removing /home/nitesh/.minikube/machines/minikube ...
💀  Removed all traces of the "minikube" cluster.
nitesh@nitesh:~$
```