1.Started minikube and status checked. Manifest File created for pod creation

```
bala@ubuntu:~$ minikube start
😄  minikube v1.33.1 on Ubuntu 23.10 (vbox/amd64)
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.44 ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
🔎  Verifying Kubernetes components...
    ▪ Using image registry.k8s.io/ingress-nginx/controller:v1.10.1
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
    ▪ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
    ▪ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
🔎  Verifying ingress addon...
🌟  Enabled addons: storage-provisioner, default-storageclass, ingress
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
bala@ubuntu:~$ cd mini
bala@ubuntu:~/mini$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

bala@ubuntu:~/mini$ nano pod.yml
bala@ubuntu:~/mini$ cat pod.yml
apiVersion: v1
kind: Pods
metadata:
 name: demo-pod
 labels:
  app: nginx
specs:
 containers:
 - name: nginx-container
   image: nginx

bala@ubuntu:~/mini$
```

2.Pods created by using manifest file & CLI

```
bala@ubuntu:~/mini$ kubectl apply -f pod.yaml
pod/demo-pod created
bala@ubuntu:~/mini$ kubectl get pods
NAME       READY   STATUS    RESTARTS   AGE
demo-pod   1/1     Running   0          13s
bala@ubuntu:~/mini$ kubectl run demo-pod2 --image=nginx:latest
pod/demo-pod2 created
bala@ubuntu:~/mini$ kubectl get pods
NAME        READY   STATUS              RESTARTS   AGE
demo-pod    1/1     Running             0          103s
demo-pod2   0/1     ContainerCreating   0          5s
bala@ubuntu:~/mini$ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
demo-pod    1/1     Running   0          21m
demo-pod2   1/1     Running   0          20m
bala@ubuntu:~/mini$
```

And the created pods are running

3.Deployment created using manifest file, inside the namespace (demo-tasks)

```
bala@ubuntu:~/mini$ nano deployment.yaml
bala@ubuntu:~/mini$ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: demo-deploy
  namespace: demo-tasks
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx-container
        image: nginx:latest
        ports:
        - containerPort: 80
bala@ubuntu:~/mini$ kubectl apply -f deployment.yaml
deployment.apps/demo-deploy created
bala@ubuntu:~/mini$ kubectl get deploy
No resources found in default namespace.
bala@ubuntu:~/mini$ kubectl get deployments -n demo-tasks
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
demo-deploy   2/2     2            2           63s
bala@ubuntu:~/mini$
```

4.Created replica pods are running inside the namespace(demo-tasks)

```
bala@ubuntu:~/mini$ kubectl get pods -n demo-tasks
NAME                           READY   STATUS    RESTARTS   AGE
demo-deploy-5449cb55b-hh8f5    1/1     Running   0          4m22s
demo-deploy-5449cb55b-km4q9    1/1     Running   0          4m22s
bala@ubuntu:~/mini$
```

5.Manifest file created for Service and Applied. Created service is Running inside namespace(demo-tasks)

```
bala@ubuntu:~/mini$ kubectl apply -f namespace.yaml
namespace/demo-tasks created
bala@ubuntu:~/mini$ nano service.yaml
bala@ubuntu:~/mini$ cat service.yaml
apiVersion: v1
kind: Service
metadata:
  name: demo-service
  namespace: demo-tasks
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80

bala@ubuntu:~/mini$ kubectl apply -f service.yaml
service/demo-service created
bala@ubuntu:~/mini$ kubectl get svc -n demo-tasks
NAME           TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
demo-service   ClusterIP   10.110.136.109   <none>        80/TCP    18s
bala@ubuntu:~/mini$
```