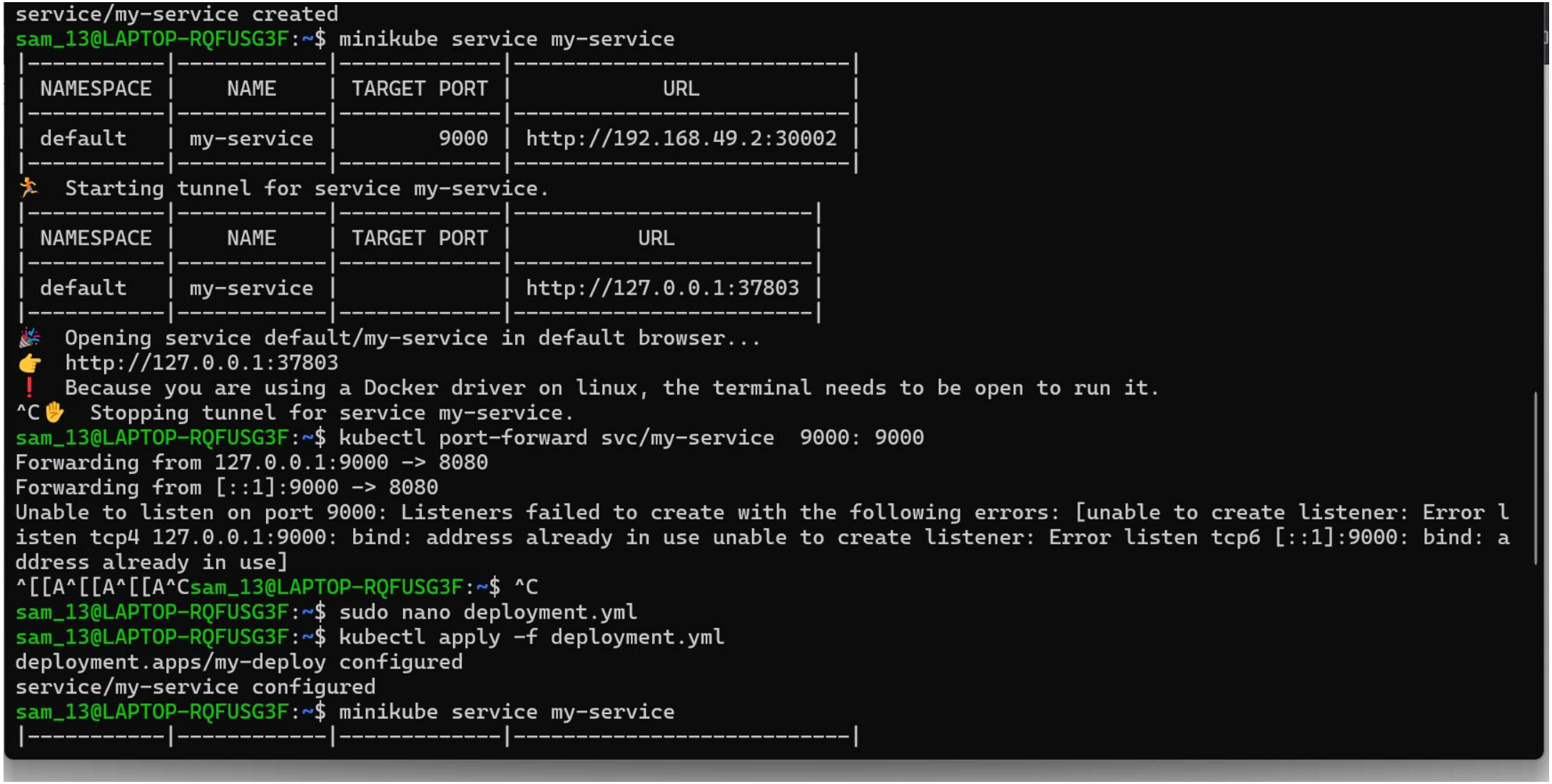
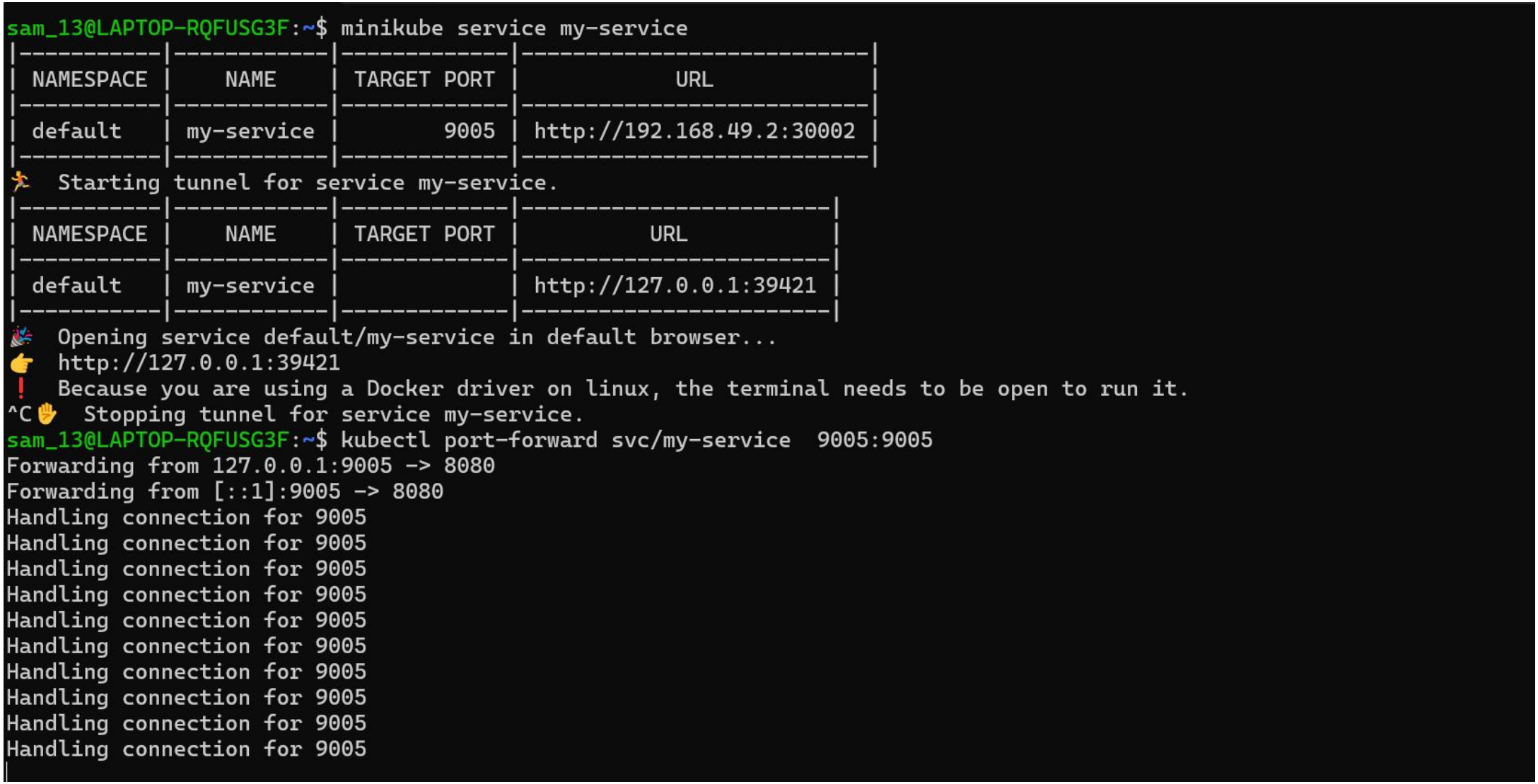
```
sam_13@LAPTOP-RQFUSG3F:~$ kubectl delete all --all
The connection to the server 127.0.0.1:32774 was refused - did you specify the right host or port?
The connection to the server 127.0.0.1:32774 was refused - did you specify the right host or port?
The connection to the server 127.0.0.1:32774 was refused - did you specify the right host or port?
The connection to the server 127.0.0.1:32774 was refused - did you specify the right host or port?
The connection to the server 127.0.0.1:32774 was refused - did you specify the right host or port?
The connection to the server 127.0.0.1:32774 was refused - did you specify the right host or port?
The connection to the server 127.0.0.1:32774 was refused - did you specify the right host or port?
The connection to the server 127.0.0.1:32774 was refused - did you specify the right host or port?
The connection to the server 127.0.0.1:32774 was refused - did you specify the right host or port?
The connection to the server 127.0.0.1:32774 was refused - did you specify the right host or port?
sam_13@LAPTOP-RQFUSG3F:~$ kubectl get pod
The connection to the server 127.0.0.1:32774 was refused - did you specify the right host or port?
sam_13@LAPTOP-RQFUSG3F:~$ minikube status
minikube
type: Control Plane
host: Stopped
kubelet: Stopped
apiserver: Stopped
kubeconfig: Stopped

sam_13@LAPTOP-RQFUSG3F:~$ minikube start
😄  minikube v1.35.0 on Ubuntu 24.04 (amd64)
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.46 ...
🔄  Restarting existing docker container for "minikube" ...
🤦  StartHost failed, but will try again: driver start: start: docker start minikube: exit status 1
stdout:
```

```
c46dff3245a05899617a6f6f/devices.allow: invalid argument: unknown
Error: failed to start containers: minikube

🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
🔎  Verifying Kubernetes components...
    ▪ 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
sam_13@LAPTOP-RQFUSG3F:~$ kubectl delete all --all
pod "my-deploy-67d7595d74-9fdql" deleted
service "kubernetes" deleted
service "my-service" deleted
deployment.apps "my-deploy" deleted
sam_13@LAPTOP-RQFUSG3F:~$ kubectl get pod
No resources found in default namespace.
sam_13@LAPTOP-RQFUSG3F:~$ kubectl get node
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   24m   v1.32.0
sam_13@LAPTOP-RQFUSG3F:~$ kubectl get deploy
No resources found in default namespace.
sam_13@LAPTOP-RQFUSG3F:~$ kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   84s
sam_13@LAPTOP-RQFUSG3F:~$ sudo nano deployment.yml
[sudo] password for sam_13:
sam_13@LAPTOP-RQFUSG3F:~$ kubectl apply -f deployment.yml
deployment.apps/my-deploy created
service/my-service created
sam_13@LAPTOP-RQFUSG3F:~$ minikube service my-service
```


service/my-service created
sam_13@LAPTOP-RQFUSG3F:~$ minikube service my-service
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | 9000 | http://192.168.49.2:30002 |
Starting tunnel for service my-service.
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | | http://127.0.0.1:37803 |
Opening service default/my-service in default browser...
http://127.0.0.1:37803
Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^C Stopping tunnel for service my-service.
sam_13@LAPTOP-RQFUSG3F:~$ kubectl port-forward svc/my-service 9000: 9000
Forwarding from 127.0.0.1:9000 -> 8080
Forwarding from [::1]:9000 -> 8080
Unable to listen on port 9000: Listeners failed to create with the following errors: [unable to create listener: Error listen tcp4 127.0.0.1:9000: bind: address already in use unable to create listener: Error listen tcp6 [::1]:9000: bind: address already in use]
^[[A^[[A^[[A^Csam_13@LAPTOP-RQFUSG3F:~$ ^C
sam_13@LAPTOP-RQFUSG3F:~$ sudo nano deployment.yml
sam_13@LAPTOP-RQFUSG3F:~$ kubectl apply -f deployment.yml
deployment.apps/my-deploy configured
service/my-service configured
sam_13@LAPTOP-RQFUSG3F:~$ minikube service my-service


sam_13@LAPTOP-RQFUSG3F:~$ minikube service my-service
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | 9005 | http://192.168.49.2:30002 |
Starting tunnel for service my-service.
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | | http://127.0.0.1:39421 |
Opening service default/my-service in default browser...
http://127.0.0.1:39421
Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^C Stopping tunnel for service my-service.
sam_13@LAPTOP-RQFUSG3F:~$ kubectl port-forward svc/my-service 9005:9005
Forwarding from 127.0.0.1:9005 -> 8080
Forwarding from [::1]:9005 -> 8080
Handling connection for 9005
Handling connection for 9005
Handling connection for 9005
Handling connection for 9005
Handling connection for 9005
Handling connection for 9005
Handling connection for 9005
Handling connection for 9005
Handling connection for 9005
Handling connection for 9005

```
^C
sam_13@LAPTOP-RQFUSG3F:~$ cat deployment.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: my-app
        image: sam1302/simple-web-app:latest
        ports:
        - containerPort: 9005

---

apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-service
spec:
  type: NodePort
  ports:
    - port: 9005
      targetPort: 8080
      nodePort: 30002
  selector:
    apptype: web-backend
sam_13@LAPTOP-RQFUSG3F:~$
```