TASK 4

Step 1: Open Ubuntu and create two .txt files with the following code:

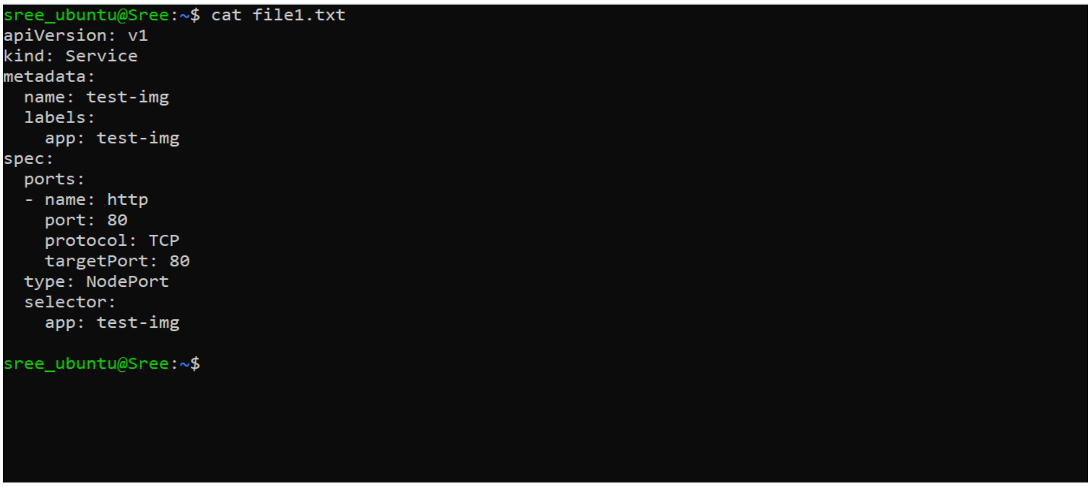
File1.txt

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: test-img
  name: test-img
spec:
  replicas: 1
  selector:
    matchLabels:
      app: test
  template:
metadata:
  labels:
    app: test
spec:
  containers:
  - name: test-img
    image: sreevadhani/task
    imagePullPolicy: Always
    ports:
    - containerPort: 80
      name: http
      protocol: TCP
```

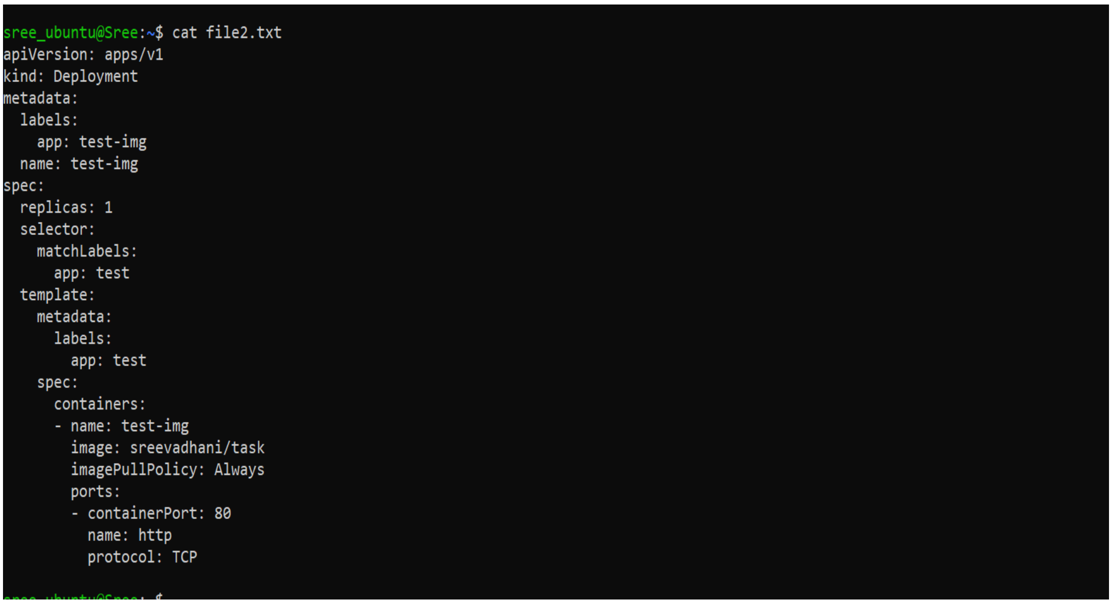
File2.txt

```
# service type loadbalancer
---
apiVersion: v1
kind: Service
metadata:
  labels:
    app: test-img
    name: test-img
spec:
ports:
- name: http
  port: 80
  protocol: TCP
  targetPort: 80
type: NodePort
selector:
  app: test-img
```

Step 2: Open vim editor and save the files.



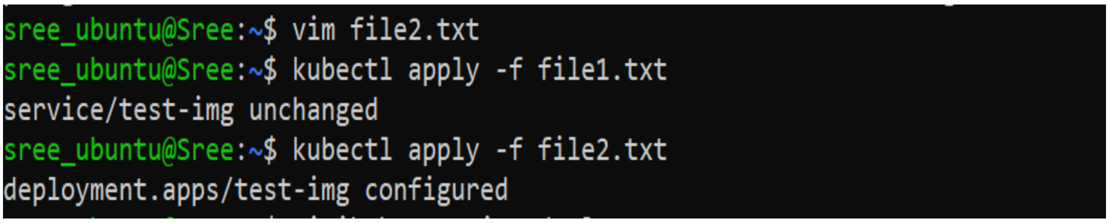
```
sree_ubuntu@Sree:~$ cat file1.txt
apiVersion: v1
kind: Service
metadata:
  name: test-img
  labels:
    app: test-img
spec:
  ports:
  - name: http
    port: 80
    protocol: TCP
    targetPort: 80
  type: NodePort
  selector:
    app: test-img

sree_ubuntu@Sree:~$
```



```
sree_ubuntu@Sree:~$ cat file2.txt
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: test-img
  name: test-img
spec:
  replicas: 1
  selector:
    matchLabels:
      app: test
  template:
    metadata:
      labels:
        app: test
    spec:
      containers:
      - name: test-img
        image: sreevadhani/task
        imagePullPolicy: Always
        ports:
        - containerPort: 80
          name: http
          protocol: TCP
```

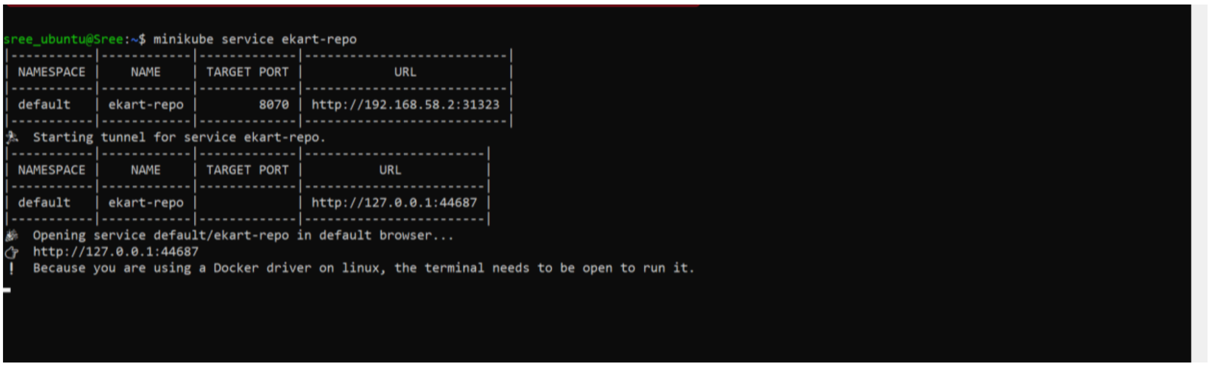
Step 3: Create deployment configuration

kubectl apply -f file1.txt - This command applies the Deployment configuration and creates the deployment named test.



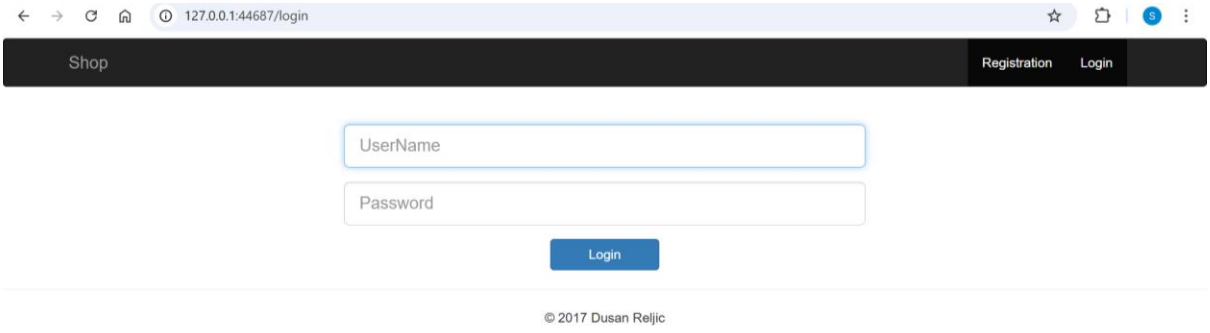
```
sree_ubuntu@Sree:~$ vim file2.txt
sree_ubuntu@Sree:~$ kubectl apply -f file1.txt
service/test-img unchanged
sree_ubuntu@Sree:~$ kubectl apply -f file2.txt
deployment.apps/test-img configured
```

Step 4: Starting the tunnel



```
sree_ubuntu@Sree:~$ minikube service ekart-repo
|-----------|------------|-------------|---------------------------|
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
|-----------|------------|-------------|---------------------------|
| default   | ekart-repo |        8070 | http://192.168.58.2:31323 |
|-----------|------------|-------------|---------------------------|
🏃  Starting tunnel for service ekart-repo.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | ekart-repo |             | http://127.0.0.1:44687 |
|-----------|------------|-------------|------------------------|
🎉  Opening service default/ekart-repo in default browser...
👉  http://127.0.0.1:44687
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```

Step 5: Go to http://localhost:44687 to access the deployed application



127.0.0.1:44687/login
Shop
Registration
Login
UserName
Password
Login
© 2017 Dusan Reljic