```
kubectl port-forward nginx 8080:80
```

Implementing the service model in kubernetes.

It used to

1.Discovery

2.Load balancing

3. Exposing to outside the world.

Step 1:

We need to up the docker container and minikube cluster.

Step 2: clone the git repository using the below command.

```
git clone https://github.com/iam-veeramalla/Docker-Zero-to-Hero
cd Docker-Zero-to-Hero
cd example
cd python.app
```

Step 3:

Create the docker container image use the below command.

```
docker build  -t akhil626/python:v1 .
docker push   akhil626/python:v1
docker run  -d akhil626/python:v1
Image: docker.io/akhil626/python:v1
docker rmi  -f akhil626/python:v1(to delete the docker image).
```

Command: docker run -t Akhi626/python-app-sample:v .

Step 4:

Create the deployment in minikube cluster.

Vi deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

Save the file.

Then create the deployment using the below command

```
Kubectl -f apply deployment.yaml
```

Then we can create the pods and replica sets.

We can check it by using the below commands

```
Kubectl get pods -o wide
```

Step 5:

Login to minikube using the below command

```
Minikube ssh
```

```
Curl -L http://ip address of pod:portno
```

We can find the output.

Step 6:

But we are unable to get load balancing and node port mode.

So we need to create the service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: python-dnjgo
spec:
  type: NodePort
  selector:
      app: nginx

  ports:
      # By default and for convenience, the `targetPort` is set to the same value as the `port` field.
    - port: 80
      targetPort: 8000
      # Optional field
      # By default and for convenience, the Kubernetes control plane will allocate a port from a range (default: 30000-32767)
      nodePort: 30007
```

in service.yaml selector must have same name as in deployment.yaml file.

Then apply the below command to run services

```
Kubeclt apply -f service.yaml
Kubectl get svc(we can get the all the services)
```

```
Minikube ip
```

```
Curl
```

If we need change it to load balance we just change in service.yaml file. Replace nodeport mode to loadbalancer.

Load balancer can implement in cloud service like AWS, GCP, Azure.

We can access the application anywhere in the world.