Loadbalancer in minikube

A LoadBalancer service in Kubernetes exposes an application to the internet using an external IP. It is commonly used in cloud environments (AWS, GCP, Azure) to distribute traffic across multiple pods.

C: > Users > Niree > ! service.yml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80        # Exposes the service on port 80
      targetPort: 80  # Forwards traffic to container port 80
  type: LoadBalancer  # Exposes the service with an external IP
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

minikube

```
PS C:\Users\Niree> kubectl apply -f service.yml
service/nginx-service configured
PS C:\Users\Niree> kubectl get svc
NAME              TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
example-service   LoadBalancer   10.100.164.58   <pending>     8765:31040/TCP   8h
kubernetes        ClusterIP      10.96.0.1       <none>        443/TCP          3d21h
nginx-service     LoadBalancer   10.108.58.137   <pending>     80:30080/TCP     40m
```

```
PS C:\Users\Niree> minikube tunnel
✅  Tunnel successfully started

📌  NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

🏃  Starting tunnel for service example-service.
❗  Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more information, see: https://minikube.sigs.k
8
s.io/docs/handbook/accessing/#access-to-ports-1024-on-windows-requires-root-permission
🏃  Starting tunnel for service nginx-service.
```

- Apply Service.yml file
- then get svc
- it shows loadbalancer in svc
- if need to expose in minikube means
- need to use (minikube tunnel) command
- This will create an external IP that can be accessed locally.

kubectl apply -f service.yml

kubectl get svc

minikube tunnel