## **Creating a service for an application running in five pods**

1. **Run a Hello World application in your cluster:**

kubectl run hello-world --replicas=5 --labels="run=load-balancer-example" --image=gcr.io/google-samples/node-hello:1.0 --port=8080

The preceding command creates a [Deployment](https://kubernetes.io/docs/concepts/workloads/controllers/deployment/) object and an associated [ReplicaSet](https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/) object. The ReplicaSet has five [Pods](https://kubernetes.io/docs/concepts/workloads/pods/pod/), each of which runs the Hello World application.

1. **Display information about the Deployment:**

kubectl get deployments hello-world  
kubectl describe deployments hello-world

1. **Display information about your ReplicaSet objects:**

kubectl get replicasets  
kubectl describe replicasets

1. **Create a Service object that exposes the deployment:**

kubectl expose deployment hello-world --type=LoadBalancer --name=my-service

1. **Display information about the Service:**

kubectl get services my-service

**The output is similar to this:**

NAME CLUSTER-IP EXTERNAL-IP PORT(S) AGE  
my-service 10.3.245.137 104.198.205.71 8080/TCP 54s

**Note:** If the external IP address is shown as <pending>, wait for a minute and enter the same command again.

1. **Display detailed information about the Service:**

kubectl describe services my-service

1. **The output is similar to this:**

Name: my-service  
Namespace: default  
Labels: run=load-balancer-example  
Annotations: <none>  
Selector: run=load-balancer-example  
Type: LoadBalancer  
IP: 10.3.245.137  
LoadBalancer Ingress: 104.198.205.71  
Port: <unset> 8080/TCP  
NodePort: <unset> 32377/TCP  
Endpoints: 10.0.0.6:8080,10.0.1.6:8080,10.0.1.7:8080 + 2 more...  
Session Affinity: None  
Events: <none>

1. Make a note of the external IP address (LoadBalancer Ingress) exposed by your service. In this example, the external IP address is 104.198.205.71. Also note the value of Port and NodePort. In this example, the Port is 8080 and the NodePort is 32377.
2. In the preceding output, you can see that the service has several endpoints: 10.0.0.6:8080,10.0.1.6:8080,10.0.1.7:8080 + 2 more. These are internal addresses of the pods that are running the Hello World application. To verify these are pod addresses, enter this command:

kubectl get pods –output=wide

**The output is similar to this:**

NAME ... IP NODE  
hello-world-2895499144-1jaz9 ... 10.0.1.6 gke-cluster-1-default-pool-e0b8d269-1afc  
hello-world-2895499144-2e5uh ... 10.0.1.8 gke-cluster-1-default-pool-e0b8d269-1afc  
hello-world-2895499144-9m4h1 ... 10.0.0.6 gke-cluster-1-default-pool-e0b8d269-5v7a  
hello-world-2895499144-o4z13 ... 10.0.1.7 gke-cluster-1-default-pool-e0b8d269-1afc  
hello-world-2895499144-segjf ... 10.0.2.5 gke-cluster-1-default-pool-e0b8d269-cpuc

1. **Use the external IP address (LoadBalancer Ingress) to access the Hello World application:**

curl http://<external-ip>:<port>

where <external-ip> is the external IP address (LoadBalancer Ingress) of your Service, and <port> is the value of Port in your Service description. If you are using minikube, typing minikube service my-service will automatically open the Hello World application in a browser.

**The response to a successful request is a hello message:**

Hello Kubernetes!

## **Cleaning up**

**To delete the Service, enter this command:**

kubectl delete services my-service

**To delete the Deployment, the ReplicaSet, and the Pods that are running the Hello World application, enter this command:**

kubectl delete deployment hello-world

* **AutoScale**

https://kubernetes.io/docs/concepts/workloads/controllers/deployment/#updating-a-deployment