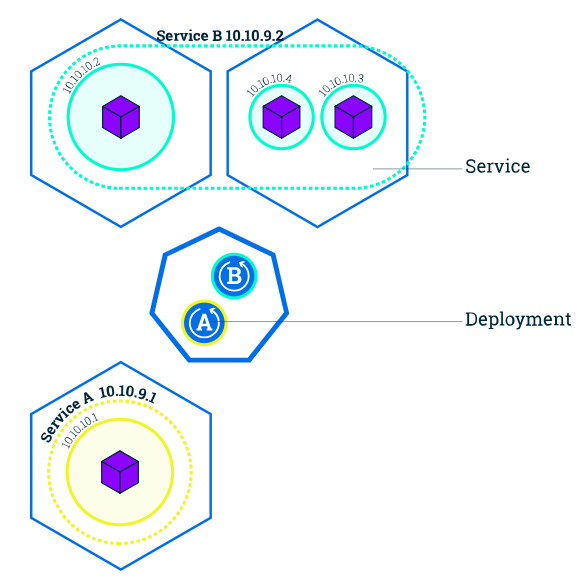
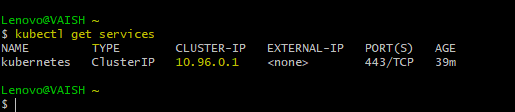
In the previous tutorial we have seen that **Pods have their own unique IP across the cluster, those IP’s are not exposed outside Kubernetes**. Taking this into account that over time *Pods may be terminated*, *deleted* or *replaced* by other Pods, we need a way to let other Pods and applications automatically discover each other. Kubernetes addresses this by **grouping Pods in Services**. A Kubernetes Service is an abstraction layer which defines a logical set of Pods and enables external traffic exposure, load balancing and service discovery for those Pods.

This abstraction will allow us to expose Pods to traffic originating from outside the cluster. **Services have their own unique cluster-private IP address and expose a port to receive traffic.**







We can see a listed service called kubernetes that is created by default when minikube starts the cluster.

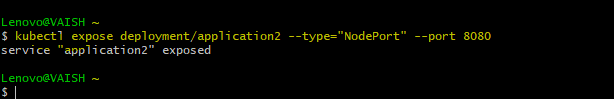
So, basically the above pod is virtually inside this service called kubernetes.

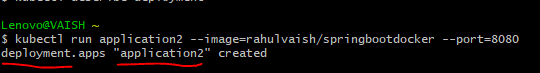
**Service IP = 10.96.0.1**

**Pod IP = 172.17.0.5 (unexposed)**

If you choose to expose the service outside the cluster, we have two options:

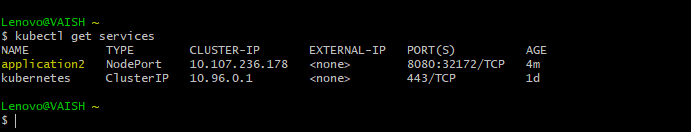
* **LoadBalancer** - provides a public IP address (what you would typically use when you run Kubernetes on GKE or AWS. minikube does not support the LoadBalancer option yet).
* **NodePort** - exposes the Service on the same port on each Node of the cluster using NAT (available on all Kubernetes clusters, and in Minikube)Let’s see how to do this-



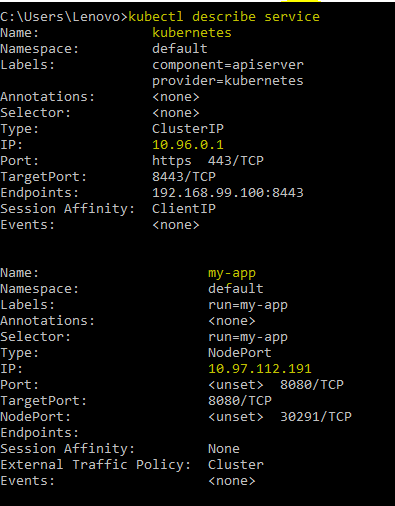
**NOTE: Hope you remember this deployment step from Chapter2. **

**Now the IMPORTANT POINT to note is that our containerized application name was hellodocker. When we deployed it on Kubernetes, we gave the name of its running instance as application2. This application2 resides inside a POD which is** **application2-874d69665-724hp. To expose service we will have to refer the name deployment/application2**

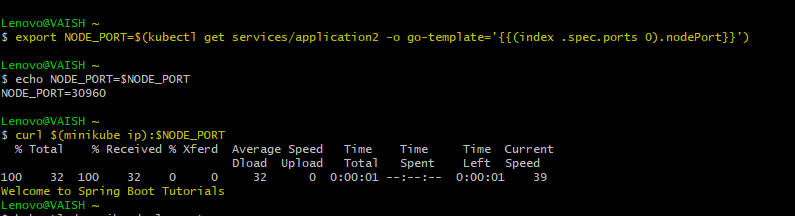
Now, if we check kubectl get services. We will find our default Kubernetes service + the new one we just hosted ☺

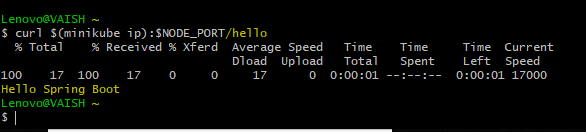


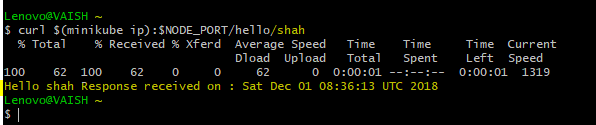
What to know about your services? kubectl describe services



Now let’s test this service:

****

****

****

The services are Working :D

