# Services

Dado que los pods en Kubernetes son mortales, es decir, cuando mueren se genera otro nuevo con una IP diferente, necesitamos tener una abstracción que nos permita identificarlos sin atacar a una IP concreta.

En Kubernetes existen los ‘services’ que es una abstracción que nos permite generar políticas de tráfico. De esta forma, podemos definir que un determinado tráfico se curse a través de los pods marcados con una determinada etiqueta.

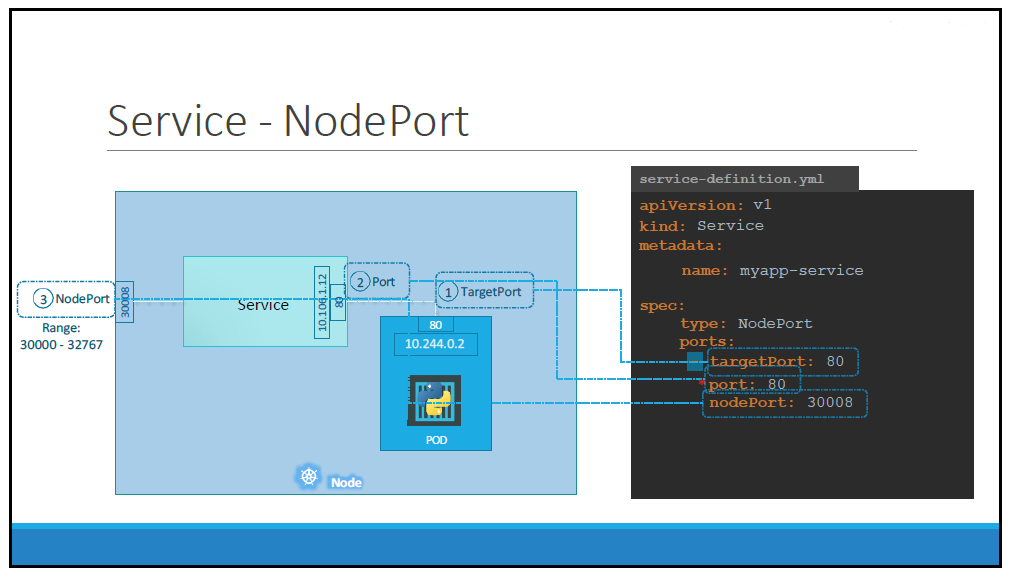
Exsiten tres formas de hacer esto, mediante NodePorts, ClusteIps y LoadBalancers.

## NodePort

La idea es abrir un puerto en todos los nodos a partir del cual podemos cursar tráfico a los pods usando un puerto del pod determinado.

Si generamos varios pods seleccionables por el ‘NodePort’, se balancea el tráfico de forma aleatoria entre todos ellos.

Para el caso en el que tengamos varios nodos, por defecto y de forma transparente, K8S genera el ‘service’ para todos ellos de forma cross. Por lo que dispondremos del ‘service’ en todos ellos y por tanto tendremos el NodePort configurado en cada nodo de forma automática.

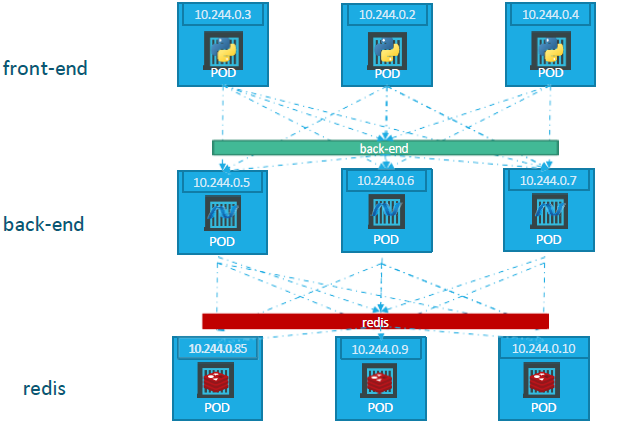


Las desventajas son:

1. Que solo podemos tener un servicio por puerto.
2. SOLO podemos usar los puertos 30000 a 32767.
3. Existe la posibilidad de que la IP del Nodo cambie y por tanto perdamos la conectividad.

## ClusterIP

Una aplicación típica dispone de varias capas que deben comunicarse entre sí como las que aparecen en la siguiente imagen:



En K8S, como no podemos confiar en las IPs de los pods dado que van cambiando según mueren y se levantan, necesitamos una forma fiable de establecer las conexiones. Un ClusterIP sirve precisamente a este propósito, tener un punto centralizado a partir del cual enrutar tráfico a pods que sirvan al mismo propósito.