NodePort  
  
POD get instantiated, terminated and moved from one Node to another, in doing so POD changes IP address so how would we keep track of that? Even if the POD isn’t moving how traffic from outside reach a certain POD   
the answer for both questions is Kubernetes services.   
Services is an abstraction that defines a logical set of Pods and a policy by which you can access them, you may think of Services as your waiter in a big restaurant, this waiter isn’t cooking nor preparing the food but he just abstract everything happing at the kitchen for you as you deal only with this waiter   
so let’s cover three different type of services Node port , ClusterIP and load balancer  
and we begin with Node port by creating this ( I need to create a diagram )

**NodePort**NodePort service is an easy way to expose an application running on POD by mapping a port in the node that host this POD with a port of the application the POD as shown in the diagram

A screenshot of a cell phone

Description automatically generated

there is 2 very important parts in this services YAML file ports and selector.  
targetPort is the actual port used by the application in here its port 80 as we are planning to run a web server and nodeport is port on the node hosting that PODs   
selector is the label selector which determine which set of pods targeted by this services, in here any POD with label app: FRONT-END will be serviced by this services

apiVersion: v1

kind: Service

metadata:

name: web-app

spec:

selector:

app: webserver   
 type: NodePort

ports:

- targetPort: 80

port: 80

nodePort: 32001

|  |  |
| --- | --- |
| Tip | * Kubernetes by default allocate node port from (3000-32767) range  it could be change using the flag --service-node-port-range * The default service type is ClusterIP * Be aware with the change of the Node ip address as it could effect your services |

now let’s expose the pod we created before with services shown after putting in web-app.yaml file

[root@ip-172-25-1-56 /]# kubectl create -f web-app.yaml

service "web-app" created

[root@ip-172-25-1-56 /]# kubectl describe services web-app

Name: web-app

Namespace: default

Labels: <none>

Annotations: <none>

Selector: app=webserver

Type: NodePort

IP: 10.98.21.191

Port: <unset> 80/TCP

TargetPort: 80/TCP

NodePort: <unset> 32001/TCP

Endpoints: 10.47.255.250:80

Session Affinity: None

External Traffic Policy: Cluster

Events: <none>

Now we can test that by just send CURL -i to sent http request using the CLI

[root@computeee centos]# curl -i 10.98.21.191:80

HTTP/1.1 200 OK

Server: nginx/1.15.12

Date: Tue, 14 May 2019 18:33:07 GMT

Content-Type: text/html

Content-Length: 612

Last-Modified: Tue, 16 Apr 2019 13:08:19 GMT

Connection: keep-alive

ETag: "5cb5d3c3-264"

Accept-Ranges: bytes

<!DOCTYPE html>

<html>

<head>

<title>Welcome to nginx!</title>

<style>

body {

width: 35em;

margin: 0 auto;

font-family: Tahoma, Verdana, Arial, sans-serif;

}

</style>

</head>

<body>

<h1>Welcome to nginx!</h1>

<p>If you see this page, the nginx web server is successfully installed and

working. Further configuration is required.</p>

<p>For online documentation and support please refer to

<a href="http://nginx.org/">nginx.org</a>.<br/>

Commercial support is available at

<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>

</body>

</html>