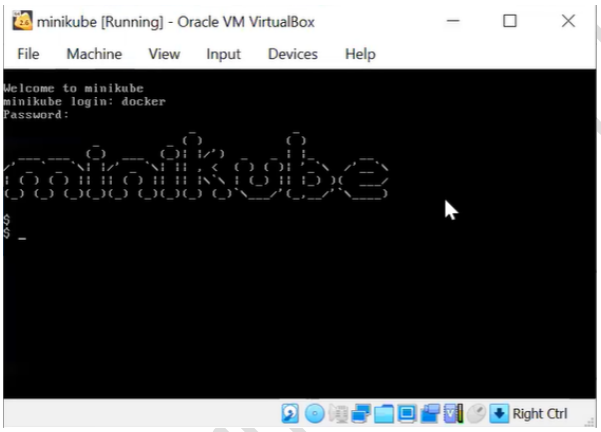
Summary

- How to log in to the Minikube cluster
  - From the virtual box
    - Minikube login : docker
    - Password: tcuser



  - Minukube ssh

- Whatever pod we launch inside Kubernetes is launched by the docker
  - Login to minikube
  - Command:- docker ps

```
$
$
$ docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED              STATUS
    NAMES
830abd471ecd   httpd                  "httpd-foreground"       About a minute ago   Up About a minute
    k8s_mypod1_mypod1_default_66d2513e-f8a6-4ce0-9183-3a9243ccafed_0
14a026fc35c0   k8s.gcr.io/pause:3.5   "/pause"                 About a minute ago   Up About a minute
    k8s_POD_mypod1_default_66d2513e-f8a6-4ce0-9183-3a9243ccafed_0
35f56e37cc0e   8d147537fb7d           "/coredns -conf /etc…"   4 hours ago          Up 4 hours
    k8s_coredns_coredns-78fcd69978-scd2p_kube-system_95472e64-f328-47df-a09d-ee00730842d7_2
8a4c3b0c5ff8   6e38f40d628d           "/storage-provisioner"   4 hours ago          Up 4 hours
    k8s_storage-provisioner_storage-provisioner_kube-system_fb995623-1ad6-45a7-b09d-fe4a9ccc75a5_3
ec77e7831ebc   6120bd723dce           "/usr/local/bin/kube…"   4 hours ago          Up 4 hours
    k8s_kube-proxy_kube-proxy-mvggn_kube-system_0b330d34-4179-4670-9aef-8529e3eebbe7_2
29ebb55b9d2e   k8s.gcr.io/pause:3.5   "/pause"                 4 hours ago          Up 4 hours
    k8s_POD_coredns-78fcd69978-scd2p_kube-system_95472e64-f328-47df-a09d-ee00730842d7_2
052395cb7ab7   k8s.gcr.io/pause:3.5   "/pause"                 4 hours ago          Up 4 hours
    k8s_POD_storage-provisioner_kube-system_fb995623-1ad6-45a7-b09d-fe4a9ccc75a5_2
077661c4ab6f   k8s.gcr.io/pause:3.5   "/pause"                 4 hours ago          Up 4 hours
```

- How to go inside the docker container which is running in the Kubernetes cluster
    - Login to minikube
    - Command:- docker attach (container id /name)

```
$ docker attach  830abd471ecd



date
```

- A good practice is whatever we are doing with Kubernetes always do with the kubectl command
- Practical:- Two pods in one single cluster by default have connectivity
    - Launching pods

```
C:\Users\Vimal Daga>kubectl run myweb --image=vimal13/apache-webserver-php
pod/myweb created

C:\Users\Vimal Daga>kubectl get pods
NAME     READY   STATUS              RESTARTS        AGE
mypod1   1/1     Running             1 (8m18s ago)   13m
myweb    0/1     ContainerCreating   0               4s
```

    - Entering inside the container

```
C:\Users\Vimal Daga>kubectl  exec -it myweb   --  bash
[root@myweb /]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.4  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:04  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
[root@myweb html]# cat index.php
<body bgcolor='aqua'>
<pre>

<?php

print "welcome to vimal web server for testing";


print `ifconfig`;

?>

</pre>
[root@myweb html]#
```

- Pinging another pod

```
[root@myweb html]# ping  172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.496 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.083 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.093 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.061 ms
64 bytes from 172.17.0.2: icmp_seq=5 ttl=64 time=0.164 ms
^C
--- 172.17.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4089ms
rtt min/avg/max/mdev = 0.061/0.179/0.496/0.162 ms
```

- Every pod has connectivity with each other in a cluster

- How to expose the pod to clients outside the cluster
  - Launching pods

```
C:\Users\Vimal Daga>kubectl expose pod myweb  --type=NodePort --port=80
service/myweb exposed

C:\Users\Vimal Daga>kubectl  get svc
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP        134m
myweb        NodePort    10.110.161.140   <none>        80:30185/TCP   10s
```

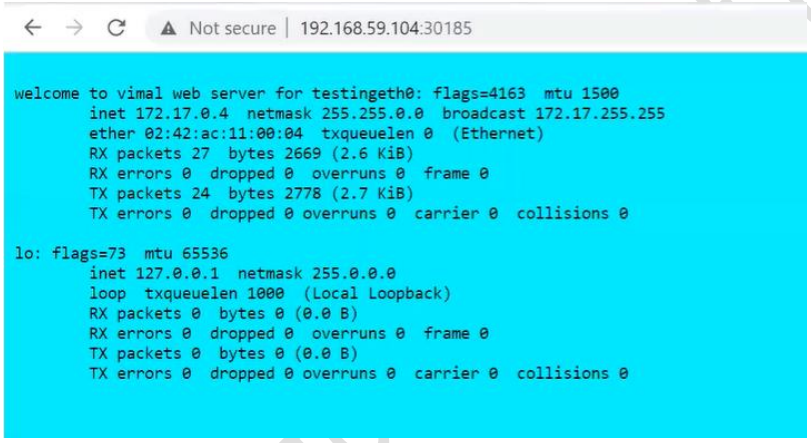
- Getting minikube IP

```
C:\Users\Vimal Daga>minikube ip
192.168.59.104

C:\Users\Vimal Daga>
```

- Getting port no from service (kubectl get svc)

```
C:\Users\Vimal Daga>kubectl  get svc
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP        134m
myweb        NodePort    10.110.161.140   <none>        80:30185/TCP   10s
```

- Client connecting from outside of the cluster



- Practical:- Setting multi-container with Node port load balancer
  - YAML file for Node port load balancer

```yaml
apiVersion: v1
kind: Service
metadata:
  name: mywebservice
spec:
  type: NodePort
  selector:
    dc: IN
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30000
```

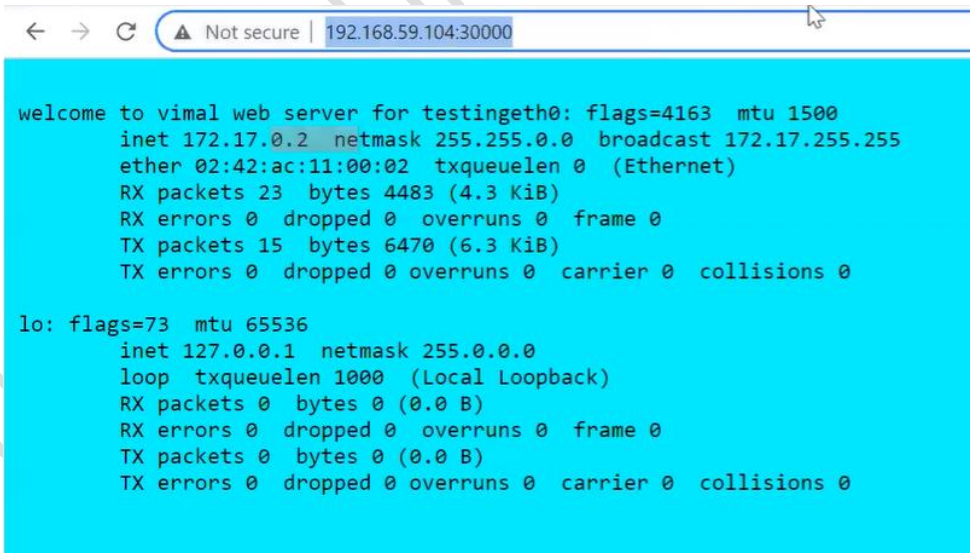
- Scaling pods
  - Command:- kubectl scale rc (Name of replication controller) --replicas=3

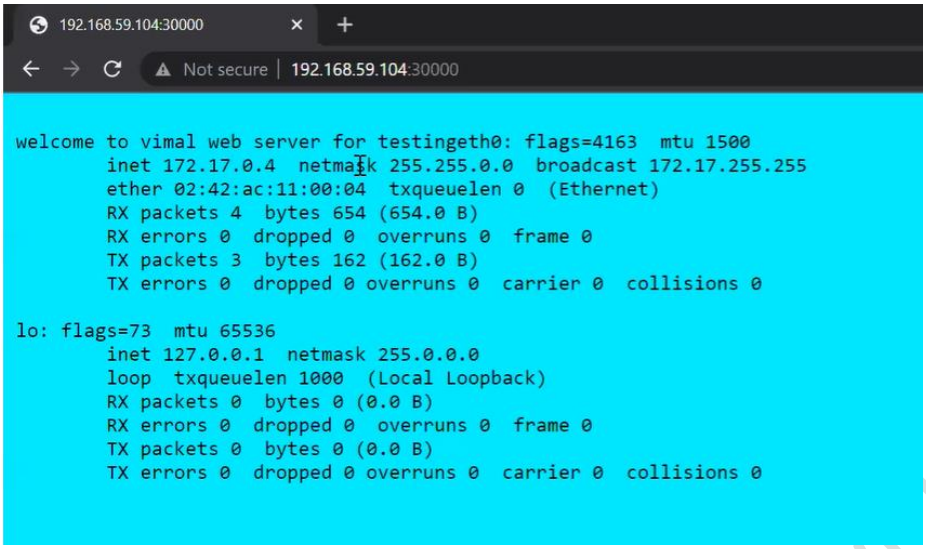
```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl scale rc myrc1 --replicas=3
replicationcontroller/myrc1 scaled

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get rc
NAME    DESIRED   CURRENT   READY   AGE
myrc1   3         3         1       20m

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get pods
NAME          READY   STATUS              RESTARTS   AGE
myrc1-85whm   1/1     Running             0          20m
myrc1-db9hm   0/1     ContainerCreating   0          7s
myrc1-xr9ds   0/1     ContainerCreating   0          7s
```

- Checking endpoints

```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl describe svc/mywebservice
Name:                     mywebservice
Namespace:                default
Labels:                   <none>
Annotations:              <none>
Selector:                 dc=IN
Type:                     NodePort
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.99.234.11
IPs:                      10.99.234.11
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  30000/TCP
Endpoints:                172.17.0.2:80,172.17.0.4:80,172.17.0.5:80
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
```

- Client hitting the server





- Client connecting to different pods having different IP

- How to launch the cluster IP load balancer with the YAML file

```yaml
apiVersion: v1
kind: Service
metadata:
  name: mywebsvcpriv
spec:
  type: ClusterIP
  selector:
    dc: IN
  ports:
    - port: 80
      targetPort: 80
```