Create and Manage Containers on Minikube.

1. **Create an yml file for POD creation**
2. **Create an yml file for Service creation**
3. **Run the yml files**
4. **Check whether the env is up and running, if not troubleshoot**
5. **Check the output of the Env.**
6. **Test the POD stability**
7. **Delete the POD and Services.**

1.Create an yml file for POD creation

```
apiVersion: v1
kind: Pod
metadata:
  name: client-pod
  labels:
    component: web
spec:
  containers:
    - name: client
      image: tomcat
      ports:
        - containerPort: 8080
```

the file is saved as "**client-pod.yaml**"

```
[G:\Hypverv-VM\minikube]$ dir
 Volume in drive G is New Volume
 Volume Serial Number is 7890-388C

 Directory of G:\Hypverv-VM\minikube

10/06/2019  03:49 PM    <DIR>          .
10/06/2019  03:49 PM    <DIR>          ..
10/06/2019  11:08 AM               204 client-node-port.yaml
10/06/2019  11:17 AM               208 client-pod.yaml
10/06/2019  03:11 PM     3,351,498,752 disk.vhd
10/06/2019  04:13 PM       834,248,704 disk_36DC5EB5-4612-4F90-B6A
               4 File(s)  4,185,747,868 bytes
               2 Dir(s)  51,052,457,984 bytes free

[G:\Hypverv-VM\minikube]$
```

Note: -- The yaml files are stored on the local desktop

2.Create an yml file for Service creation

```
apiVersion: v1
kind: Service
metadata:
  name: client-nodeport
spec:
  type: NodePort
  ports:
    - port: 8000
      targetPort: 8080
      nodePort: 30303
  selector:
    component: web
```

the file is saved as "**client-node-port.yaml**"

3. Run the yml files

c:/> kubectl apply -f client-pod.yaml

c:/> kubectl apply -f client-node-port.yaml

```
[G:\Hypverv-VM\minikube]$ kubectl apply -f client-pod.yaml
pod/client-pod created

[G:\Hypverv-VM\minikube]$ kubectl apply -f client-node-port.yaml
service/client-nodeport created

[G:\Hypverv-VM\minikube]$
```

4.Checking the status

```
[G:\Hypverv-VM\minikube]$ kubectl get nodes
NAME       STATUS   ROLES    AGE     VERSION
minikube   Ready    master   5h51m   v1.16.0

[G:\Hypverv-VM\minikube]$
```

C:/> kubectl get nodes
This would list all the nodes (VM's) in the kube cluster.

C:/> kubectl get pod
This would list all the pods running in the kube cluster

```
[G:\Hypverv-VM\minikube]$ kubectl get pod
NAME         READY   STATUS    RESTARTS   AGE
client-pod   1/1     Running   0          3m23s

[G:\Hypverv-VM\minikube]$
```

The 1/1 under 'Ready' says that there is '1' POD up and running out of '1' pod that needs to be running.

Age→ is showing for how long it is running.

Troubleshooting.
Note: -- if the Ready says '0/1', which means the pod is not up.

Thinks that could have gone wrong are.

1. There is no sufficient resource available to create the POD's.
2. The image required to run the POD is not getting downloaded, means in that case login to the minikube Vm and download the image manually like below.

```
[G:\Hypverv-VM\minikube]$ kubectl get pods
NAME         READY   STATUS             RESTARTS   AGE
client-pod   0/1     ImagePullBackOff   0          16m
```

```
[G:\Hypverv-VM\minikube]$ kubectl delete -f client-pod.yaml
pod "client-pod" deleted
```

```
[G:\Hypverv-VM\minikube]$ minikube ssh
                         _             _
            _         _ ( )           ( )
  ___ ___  (_)  ___  (_)| |/')  _   _ | |_      __
/' _ ` _ `\| |/' _ `\| || , <  ( ) ( )| '_`\  /'__`\
| ( ) ( ) || || ( ) || || |\`\ | (_) || |_) )(  ___/
(_) (_) (_)(_)(_) (_)(_)(_) (_)`\___/'(_,__/''`\____)

$ docker image ls
REPOSITORY                                  TAG
```

```
$ docker pull tomcat
Using default tag: latest
latest: Pulling from library/tomcat
Digest: sha256:bb4ceffaf5aa2eba6c3ee0db46d863c8b23b263cb547dec0942e757598fd0c24
```

$ docker pull tomcat
This should most of the time resolve it.

C:/> kubectl get service

```
[G:\Hypverv-VM\minikube]$ kubectl get service
NAME              TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
client-nodeport   NodePort    10.109.153.153   <none>        8000:30303/TCP   17m
kubernetes        ClusterIP   10.96.0.1        <none>        443/TCP          6h7m

[G:\Hypverv-VM\minikube]$
```

```
[G:\Hypverv-VM\minikube]$ kubectl apply -f client-pod.yaml
pod/client-pod created
```

Lets create the POD again.

```
[G:\Hypverv-VM\minikube]$ kubectl get pods
NAME         READY   STATUS    RESTARTS   AGE
client-pod   1/1     Running   0          116s
```

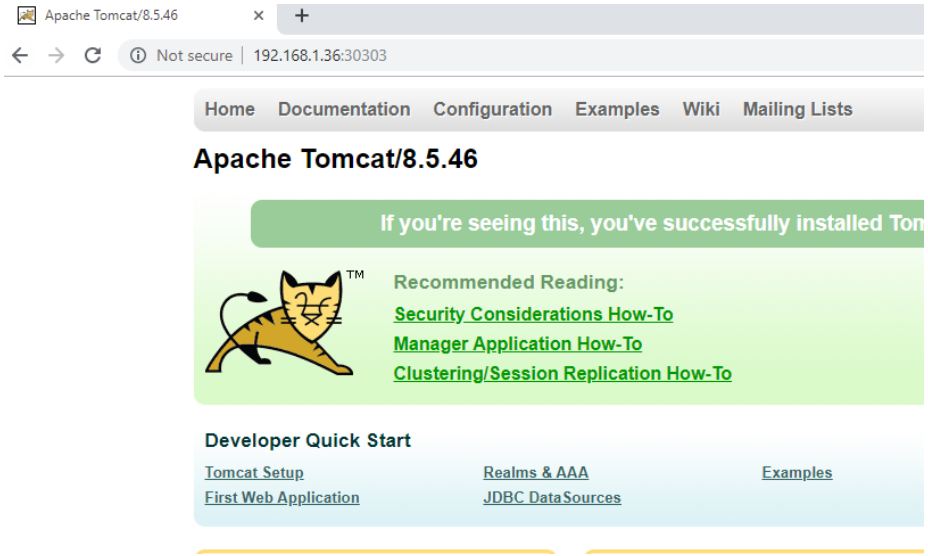
5. Check the output of the Env.

We would need the ip of the minikube vm, and use the "nodeport" port number defined in the services yaml file

```
$ ifconfig
docker0   Link encap:Ethernet  HWaddr 02:42:8D:A4:EC:A5
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          inet6 addr: fe80::42:8dff:fea4:eca5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9525 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10078 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:655538 (640.1 KiB)  TX bytes:2991010 (2.8 MiB)

eth0      Link encap:Ethernet  HWaddr 00:15:5D:38:01:00
          inet addr:192.168.1.36  Bcast:192.168.1.255  Mask:255.255.25
          inet6 addr: fe80::215:5dff:fe38:100/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:403503 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20690 errors:0 dropped:0 overruns:0 carrier:0
```

http://192.168.1.36:30303



Which means, we are successfully able to launch the tomcat inside the POD and access it from the external world.

6. Test the POD stability

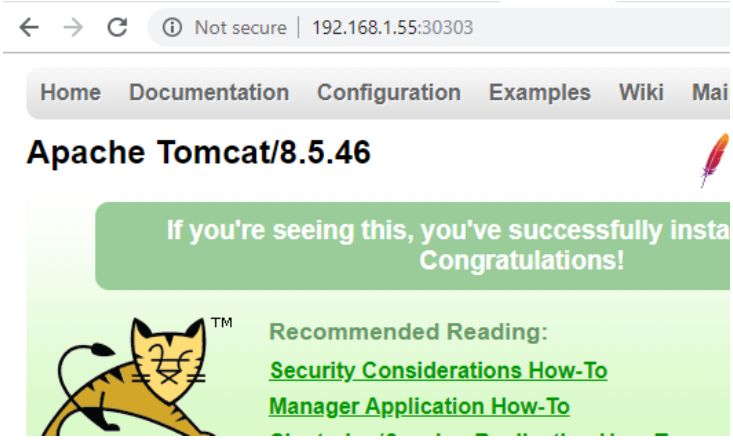
```
$ docker ps | grep tomcat
71383fff7197        tomcat                  "catalina.sh run"        3 minutes ago       Up 3 minutes
              k8s_client_client-pod_default_50e679be-5e3c-4b39-853b-2bbc772c39d1_0
$
```

The container is running.

Lets try to delete the container and lets see what happens

```
$ docker rm 71383fff7197
Error response from daemon: You cannot remove a running container 71383fff71975762979bd2097
1ce69e7a3a3053c918d. Stop the container before attempting removal or force remove
$ docker rm 71383fff7197 -f
71383fff7197
```

```
$ docker ps | grep tomcat
90b2a3ed3807        tomcat                  "catalina.sh run"        2 seconds ago       Up 1 second
              k8s_client_client-pod_default_50e679be-5e3c-4b39-853b-2bbc772c39d1_0
$
```



The container is working fine.

7. Delete the Env.

C:/> kubectl delete -f client-pod.yaml

```
[G:\Hypverv-VM\minikube]$ kubectl delete -f client-node-port.yaml
service "client-nodeport" deleted

[G:\Hypverv-VM\minikube]$ kubectl delete -f client-pod.yaml
pod "client-pod" deleted

[G:\Hypverv-VM\minikube]$
```

C:/> kubectl delete -f client-node-port.yaml

```
[G:\Hypverv-VM\minikube]$ kubectl get pods
No resources found in default namespace.

[G:\Hypverv-VM\minikube]$ kubectl get service
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   6h18m

[G:\Hypverv-VM\minikube]$
```