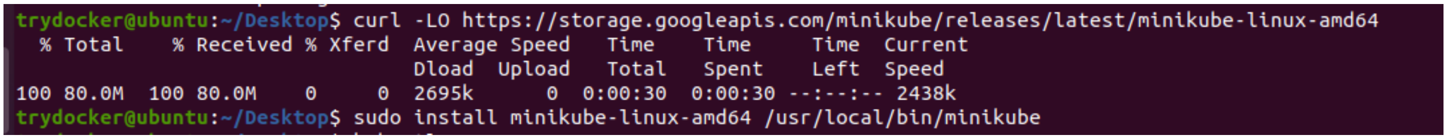
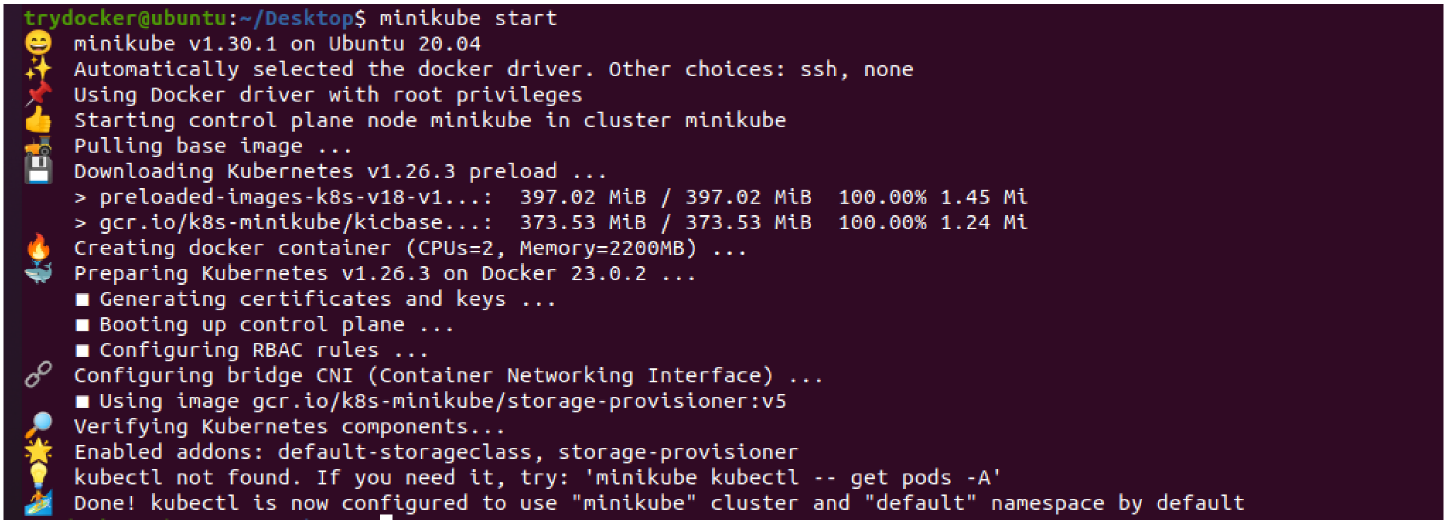
K8s assignment

- create minikube cluster

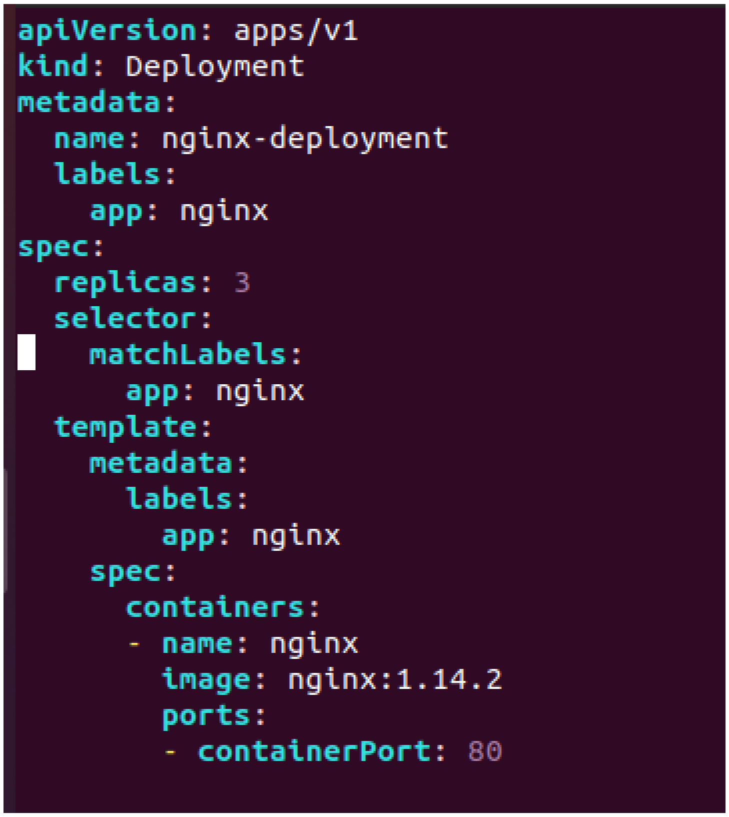


```
trydocker@ubuntu:~/Desktop$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 80.0M  100 80.0M    0     0  2695k      0  0:00:30  0:00:30 --:--:-- 2438k
trydocker@ubuntu:~/Desktop$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
```

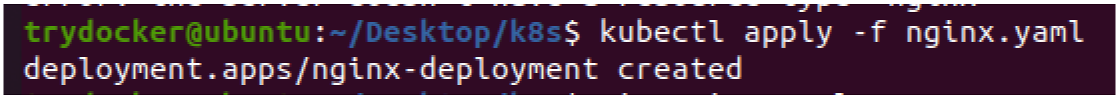


```
trydocker@ubuntu:~/Desktop$ minikube start
😄  minikube v1.30.1 on Ubuntu 20.04
✨  Automatically selected the docker driver. Other choices: ssh, none
📌  Using Docker driver with root privileges
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
💾  Downloading Kubernetes v1.26.3 preload ...
    > preloaded-images-k8s-v18-v1...:  397.02 MiB / 397.02 MiB  100.00% 1.45 Mi
    > gcr.io/k8s-minikube/kicbase...:  373.53 MiB / 373.53 MiB  100.00% 1.24 Mi
🔥  Creating docker container (CPUs=2, Memory=2200MB) ...
🐳  Preparing Kubernetes v1.26.3 on Docker 23.0.2 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🔎  Verifying Kubernetes components...
🌟  Enabled addons: default-storageclass, storage-provisioner
💡  kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

- create nginx deployment with 3 replicas

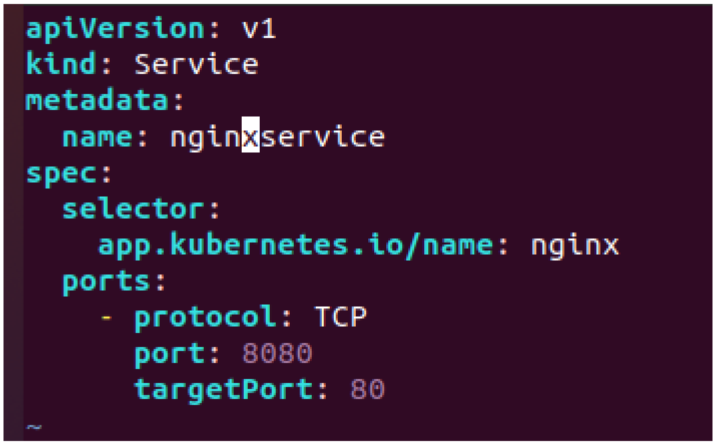


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

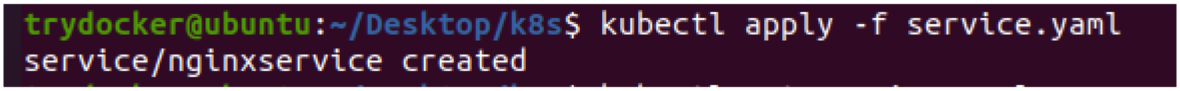


```
trydocker@ubuntu:~/Desktop/k8s$ kubectl apply -f nginx.yaml
deployment.apps/nginx-deployment created
```

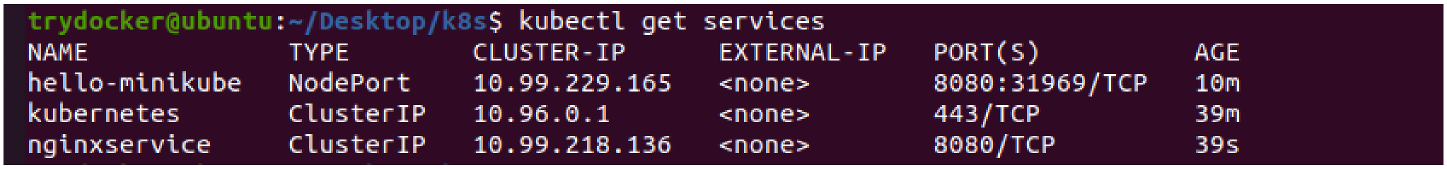
- create service to point to this deployment, type cluster IP



```
apiVersion: v1
kind: Service
metadata:
  name: nginxservice
spec:
  selector:
    app.kubernetes.io/name: nginx
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 80
~
```

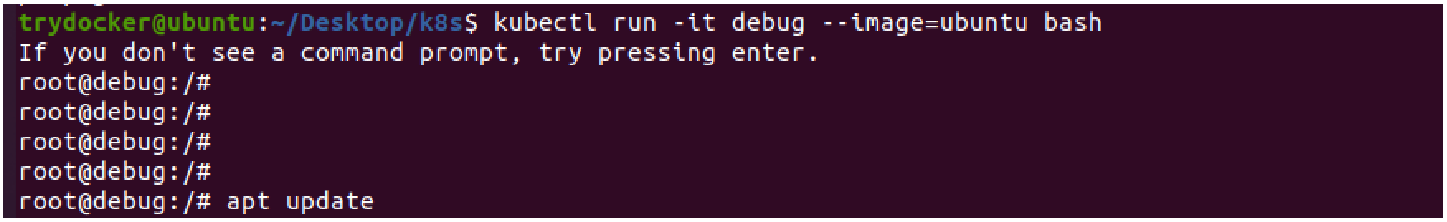


```
trydocker@ubuntu:~/Desktop/k8s$ kubectl apply -f service.yaml
service/nginxservice created
```

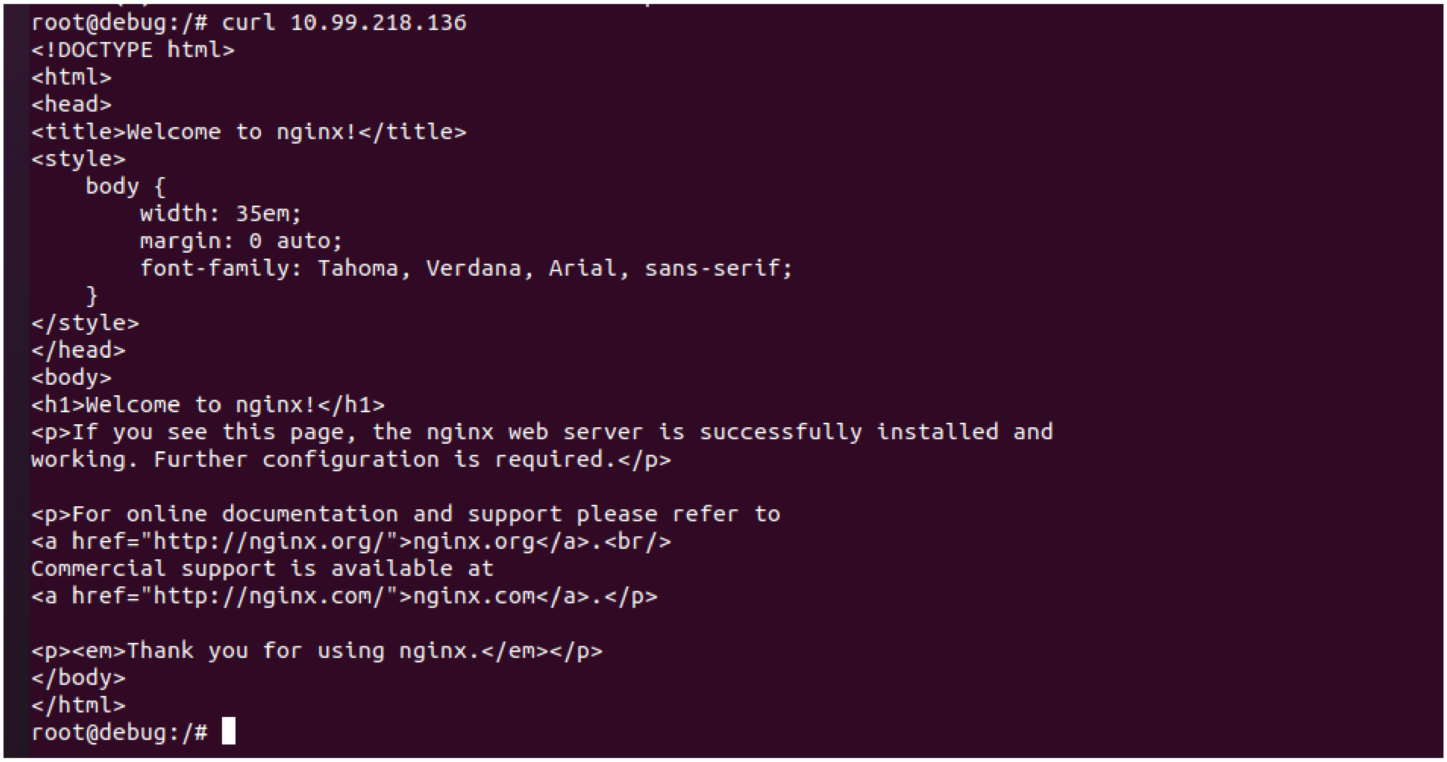


```
trydocker@ubuntu:~/Desktop/k8s$ kubectl get services
NAME             TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
hello-minikube   NodePort    10.99.229.165   <none>        8080:31969/TCP   10m
kubernetes       ClusterIP   10.96.0.1       <none>        443/TCP          39m
nginxservice     ClusterIP   10.99.218.136   <none>        8080/TCP         39s
```

- create debug pod to test the service



```
trydocker@ubuntu:~/Desktop/k8s$ kubectl run -it debug --image=ubuntu bash
If you don't see a command prompt, try pressing enter.
root@debug:/#
root@debug:/#
root@debug:/#
root@debug:/#
root@debug:/# apt update
```



```
root@debug:/# curl 10.99.218.136
<!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>
root@debug:/#
```