D_PES2UG20CS237_P K Navin Shrinivas_A2b

Details :

- Name : P K Navin Shrinivas
- Section : D
- SRN : PES2UG20CS237

Pre setup

- Install kubectl and minikube

```bash
sudo pacman -S kubectl minikube
```

- Start docker

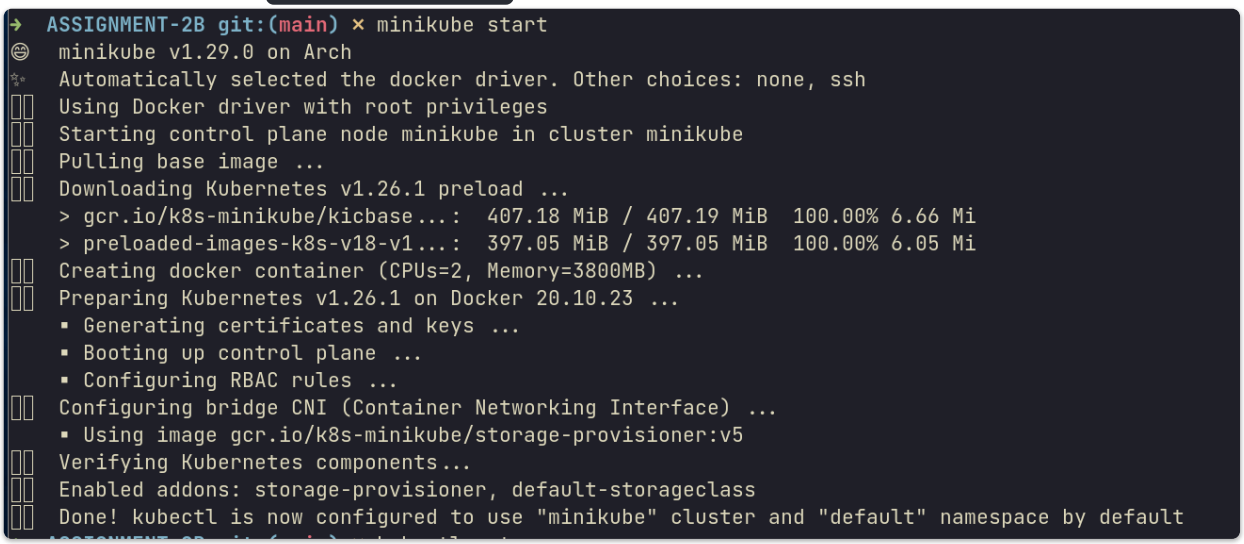
```bash
sudo systemctl start docker
```

- Run minikube :

```bash
minikube start
```

Task 1 : Deployments

- Screenshot of `minikube start` [1a] :



Task 2 : Deployments

- One can use `get` in kubectl to see various componenets [2a]:

```
→  ASSIGNMENT-2B git:(main) × kubectl get pods
No resources found in default namespace.
→  ASSIGNMENT-2B git:(main) × kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   33m   v1.26.1
→  ASSIGNMENT-2B git:(main) × kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   33m
→  ASSIGNMENT-2B git:(main) × _
```

- We create prods through deplyment, here is a minimal way to do it [2b]:

```
→  ASSIGNMENT-2B git:(main) × kubectl create deployment pes2ug20cs237 --image=nginx
deployment.apps/pes2ug20cs237 created
```

- We can now see the created deployment and pods [2c]:

```
→  ASSIGNMENT-2B git:(main) × kubectl get deployment
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
pes2ug20cs237   1/1     1            1           2m3s
→  ASSIGNMENT-2B git:(main) × kubectl get pods
NAME                             READY   STATUS    RESTARTS   AGE
pes2ug20cs237-5c878cb5fc-frxwz   1/1     Running   0          2m9s
→  ASSIGNMENT-2B git:(main) ×
```

- As mentioned in k8s docs, deploment in turn use `ReplicaSet` to spawn and manage pods. In the minimal command for deployment,

replicaset is set to 1. Can been seen here in screenshot :

```
➜  ASSIGNMENT-2B git:(main) ✗ kubectl get rs
NAME                          DESIRED   CURRENT   READY   AGE
pes2ug20cs237-5c878cb5fc   1         1         1       4m20s
➜  ASSIGNMENT-2B git:(main) ✗
```

- You can also see all details of the deployment using `kubectl describe deployment pes2ug20cs237` :

```
➜  ASSIGNMENT-2B git:(main) ✗ kubectl describe deployment pes2ug20cs237
Name:                   pes2ug20cs237
Namespace:              default
CreationTimestamp:      Thu, 23 Feb 2023 18:31:58 +0530
Labels:                 app=pes2ug20cs237
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=pes2ug20cs237
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=pes2ug20cs237
  Containers:
   nginx:
    Image:        nginx
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   pes2ug20cs237-5c878cb5fc (1/1 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  10m   deployment-controller  Scaled up replica set pes2ug20cs237-5c878cb5fc to 1
➜  ASSIGNMENT-2B git:(main) ✗ _
```

- As we had previously create a minimal deployment, we would like to modify it, you can do so by `export EDITOR=vim && kubectl edit deployment`

`pes2ug20cs237` [2d]:

```
  uid: f9932a07-51e7-422d-9c76-d047b59343bf
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: pes2ug20cs237
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: pes2ug20cs237
    spec:
      containers:
        - image: nginx:1.16
        imagePullPolicy: Always
        name: nginx
        resources: {}
INSERT  kubectl-edit-1762741128.yaml [+]
-- INSERT --
```

- Screenshot [2e] :

```
➜  ASSIGNMENT-2B git:(main) ✗ export EDITOR=nvim && kubectl edit deployment pes2ug20cs237
deployment.apps/pes2ug20cs237 edited
➜  ASSIGNMENT-2B git:(main) ✗
```

- Let's first see that the new deploment has been rolled out and then undo it [2f] :

```
→  ASSIGNMENT-2B git:(main) ✗ kubectl describe deployment pes2ug20cs237
Name:                   pes2ug20cs237
Namespace:              default
CreationTimestamp:      Thu, 23 Feb 2023 18:31:58 +0530
Labels:                 app=pes2ug20cs237
Annotations:            deployment.kubernetes.io/revision: 4
Selector:               app=pes2ug20cs237
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=pes2ug20cs237
  Containers:
   nginx:
    Image:        nginx:1.16
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   pes2ug20cs237-dcc7dc975 (1/1 replicas created)
```

```
→  ASSIGNMENT-2B git:(main) ✗ kubectl rollout undo deployment pes2ug20cs237
deployment.apps/pes2ug20cs237 rolled back
```

- We can also again check to see if the rollout undo has worked or not [2g] :

```
→  ASSIGNMENT-2B git:(main) ✗ kubectl describe deployment pes2ug20cs237
Name:                   pes2ug20cs237
Namespace:              default
CreationTimestamp:      Thu, 23 Feb 2023 18:31:58 +0530
Labels:                 app=pes2ug20cs237
Annotations:            deployment.kubernetes.io/revision: 5
Selector:               app=pes2ug20cs237
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=pes2ug20cs237
  Containers:
   nginx:
    Image:        nginx
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
```

Task 3 : Pods

- Get your pod name and let's use that to debug the pod [3a] :

```
→  ASSIGNMENT-2B git:(main) ✗ kubectl logs pes2ug20cs237-5c878cb5fc-pbfbr
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/02/23 13:20:43 [notice] 1#1: using the "epoll" event method
2023/02/23 13:20:43 [notice] 1#1: nginx/1.23.3
2023/02/23 13:20:43 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/02/23 13:20:43 [notice] 1#1: OS: Linux 6.1.8-arch1-1
2023/02/23 13:20:43 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/02/23 13:20:43 [notice] 1#1: start worker processes
2023/02/23 13:20:43 [notice] 1#1: start worker process 29
2023/02/23 13:20:43 [notice] 1#1: start worker process 30
2023/02/23 13:20:43 [notice] 1#1: start worker process 31
2023/02/23 13:20:43 [notice] 1#1: start worker process 32
2023/02/23 13:20:43 [notice] 1#1: start worker process 33
2023/02/23 13:20:43 [notice] 1#1: start worker process 34
2023/02/23 13:20:43 [notice] 1#1: start worker process 35
2023/02/23 13:20:43 [notice] 1#1: start worker process 36
2023/02/23 13:20:43 [notice] 1#1: start worker process 37
2023/02/23 13:20:43 [notice] 1#1: start worker process 38
2023/02/23 13:20:43 [notice] 1#1: start worker process 39
2023/02/23 13:20:43 [notice] 1#1: start worker process 40
```

- We can also see all the state changes that may have occured right after a new rollout [3b] :

```
Events:
  Type    Reason     Age    From               Message
  ----    ------     ----   ----               -------
  Normal  Scheduled  9m1s   default-scheduler  Successfully assigned default/pes2ug20cs237-5c878cb5fc-pbfbr to minikube
  Normal  Pulling    9m     kubelet            Pulling image "nginx"
  Normal  Pulled     8m57s  kubelet            Successfully pulled image "nginx" in 2.647343881s (2.647355475s including waiting)
  Normal  Created    8m57s  kubelet            Created container nginx
  Normal  Started    8m57s  kubelet            Started container nginx
→  ASSIGNMENT-2B git:(main) ✗
```

- It is also possible to debug by interacting with the pod [3c] :

> ⓘ **You have to create another deployment with mongo images to actually get an interactive shell.**

```
→  ASSIGNMENT-2B git:(main) ✗ kubectl exec -it pes2ug20cs237-mongo-57db9bc4d-glsz4 /bin/bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
root@pes2ug20cs237-mongo-57db9bc4d-glsz4:/# ls
bin   data  docker-entrypoint-initdb.d  home        lib    lib64   media  opt   root  sbin  sys  usr
boot  dev   etc                        js-yaml.js  lib32  libx32  mnt    proc  run   srv   tmp  var
root@pes2ug20cs237-mongo-57db9bc4d-glsz4:/# exit
exit
```

- delete all the deployments and use pod logs to see it happen [3d]:

```
→  ASSIGNMENT-2B git:(main) ✗ kubectl delete deployment pes2ug20cs237
deployment.apps "pes2ug20cs237" deleted
→  ASSIGNMENT-2B git:(main) ✗ kubectl delete deployment pes2ug20cs237-mongo
deployment.apps "pes2ug20cs237-mongo" deleted
→  ASSIGNMENT-2B git:(main) ✗ kubectl get pods
No resources found in default namespace.
→  ASSIGNMENT-2B git:(main) ✗
```

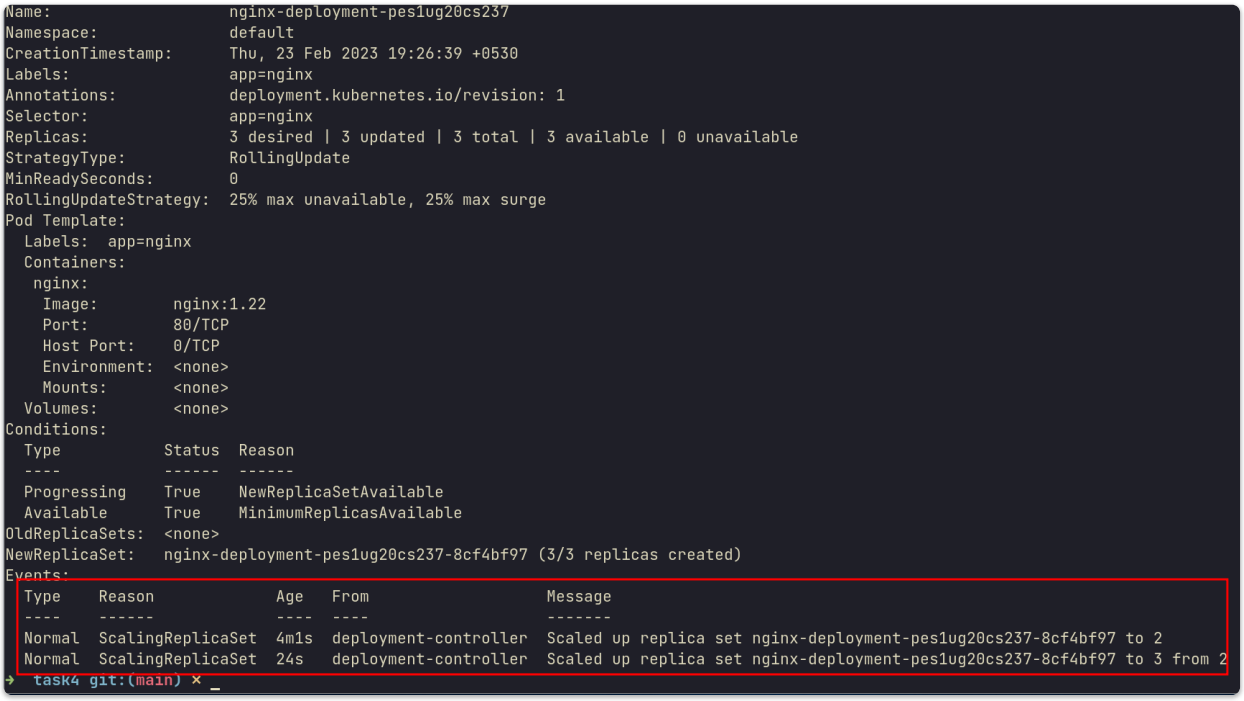
Task 4 : Manual deployments and scaling

- We have so far been using an minimal command to create deplyments, we can write a config file with all the flags before hand, create a new dir and create the following files [4a] :
  nginx-deployment.yaml :

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment-pes1ug20cs237
  labels:
    app: nginx ## The metadata of this Deployment will hold this app name
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx ## This replicaSet can handle all services with app name as nginx
  template: ## Template of the actual pod to be deployed.
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.22
        ports:
        - containerPort: 80
```

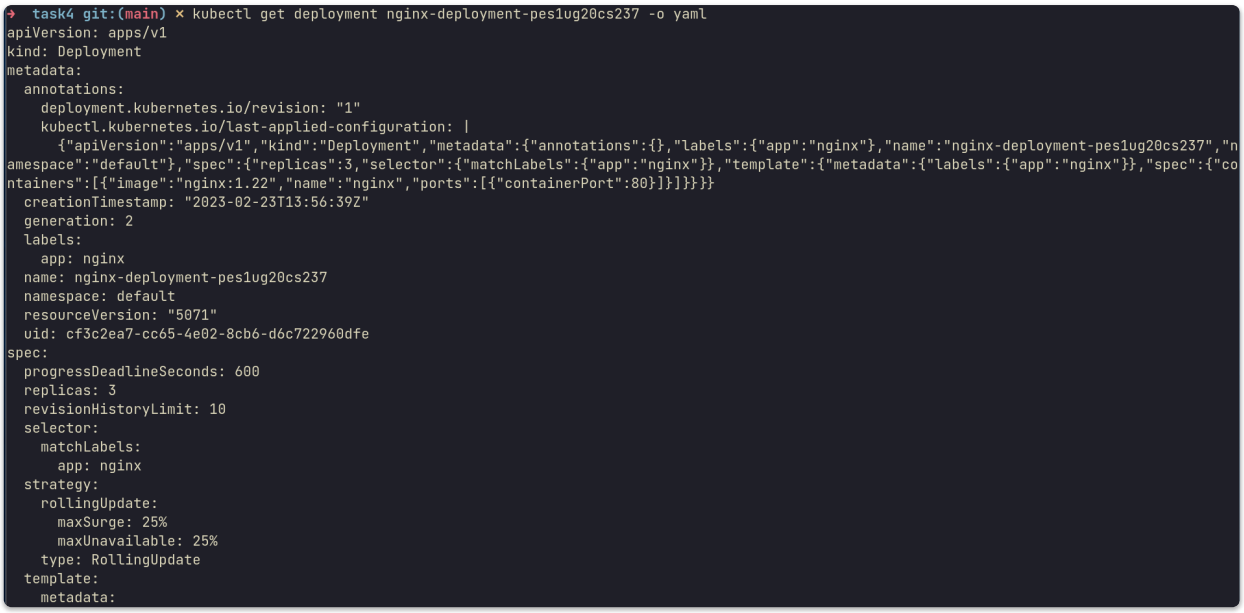
```
→  task4 git:(main) ✗ kubectl apply -f nginx-deplyoment.yaml
deployment.apps/nginx-deployment-pes1ug20cs237 created
→  task4 git:(main) ✗ kubectl get pods
NAME                                              READY   STATUS    RESTARTS   AGE
nginx-deployment-pes1ug20cs237-8cf4bf97-wxkt5   1/1     Running   0          28s
nginx-deployment-pes1ug20cs237-8cf4bf97-zk2pb   1/1     Running   0          28s
→  task4 git:(main) ✗ kubectl get deployments
NAME                              READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment-pes1ug20cs237   2/2     2            2           32s
→  task4 git:(main) ✗ kubectl get rs
NAME                                        DESIRED   CURRENT   READY   AGE
nginx-deployment-pes1ug20cs237-8cf4bf97   2         2         2       41s
→  task4 git:(main) ✗
```

- Now try updating the config file and reapplyign it, you should see the changes show up :



```
Name:                   nginx-deployment-pes1ug20cs237
Namespace:              default
CreationTimestamp:      Thu, 23 Feb 2023 19:26:39 +0530
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.22
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Progressing    True    NewReplicaSetAvailable
  Available      True    MinimumReplicasAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-deployment-pes1ug20cs237-8cf4bf97 (3/3 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  4m1s  deployment-controller  Scaled up replica set nginx-deployment-pes1ug20cs237-8cf4bf97 to 2
  Normal  ScalingReplicaSet  24s   deployment-controller  Scaled up replica set nginx-deployment-pes1ug20cs237-8cf4bf97 to 3 from 2
→  task4 git:(main) ✗ _
```

- You can view the deployment details like so [4b] :



```
→  task4 git:(main) ✗ kubectl get deployment nginx-deployment-pes1ug20cs237 -o yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"apps/v1","kind":"Deployment","metadata":{"annotations":{},"labels":{"app":"nginx"},"name":"nginx-deployment-pes1ug20cs237","n
amespace":"default"},"spec":{"replicas":3,"selector":{"matchLabels":{"app":"nginx"}},"template":{"metadata":{"labels":{"app":"nginx"}},"spec":{"co
ntainers":[{"image":"nginx:1.22","name":"nginx","ports":[{"containerPort":80}]}]}}}
  creationTimestamp: "2023-02-23T13:56:39Z"
  generation: 2
  labels:
    app: nginx
  name: nginx-deployment-pes1ug20cs237
  namespace: default
  resourceVersion: "5071"
  uid: cf3c2ea7-cc65-4e02-8cb6-d6c722960dfe
spec:
  progressDeadlineSeconds: 600
  replicas: 3
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
```

Task 5 : ReplicaSet behaviours

- Replicaset will autmatically delete and create pods to match the requirement! They use heartbreats to detect changes.

- Lets delete pods and see this [5a] :

```
→  task4 git:(main) ✗ kubectl get pods
NAME                                              READY   STATUS    RESTARTS   AGE
nginx-deployment-pes1ug20cs237-8cf4bf97-g9f72   1/1     Running   0          12m
nginx-deployment-pes1ug20cs237-8cf4bf97-wxkt5   1/1     Running   0          15m
nginx-deployment-pes1ug20cs237-8cf4bf97-zk2pb   1/1     Running   0          15m
→  task4 git:(main) ✗ kubectl delete pods nginx-deployment-pes1ug20cs237-8cf4bf97-zk2pb
pod "nginx-deployment-pes1ug20cs237-8cf4bf97-zk2pb" deleted
→  task4 git:(main) ✗ kubectl get pods
NAME                                              READY   STATUS    RESTARTS   AGE
nginx-deployment-pes1ug20cs237-8cf4bf97-2jcdz   1/1     Running   0          10s
nginx-deployment-pes1ug20cs237-8cf4bf97-g9f72   1/1     Running   0          12m
nginx-deployment-pes1ug20cs237-8cf4bf97-wxkt5   1/1     Running   0          15m
→  task4 git:(main) ✗ _
```

Task 6 : Connecting service to deployments

- Services are an asbtration of pods and describe how to access them.
- The way service targets pods is the same as replicaset, using `selector`
- Let's create a new folder and make a new service file : nginx-service.yaml :

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service-pes2ug10cs237
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 80
```

- We can apply the service and see the following [6a] :

```
→  task6 git:(main) ✗ ls
nginx-service.yaml
→  task6 git:(main) ✗ kubectl apply -f nginx-service.yaml
service/nginx-service-pes2ug20cs237 created
→  task6 git:(main) ✗ kubectl get service
NAME                          TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)    AGE
kubernetes                    ClusterIP   10.96.0.1     <none>        443/TCP    122m
nginx-service-pes2ug20cs237   ClusterIP   10.98.85.6    <none>        8080/TCP   38s
→  task6 git:(main) ✗ kubectl describe service nginx-service-pes2ug20cs237
Name:              nginx-service-pes2ug20cs237
Namespace:         default
Labels:            <none>
Annotations:       <none>
Selector:          app=nginx
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.98.85.6
IPs:               10.98.85.6
Port:              <unset>  8080/TCP
TargetPort:        80/TCP
Endpoints:         10.244.0.10:80,10.244.0.11:80,10.244.0.9:80
Session Affinity:  None
Events:            <none>
→  task6 git:(main) ✗ _
```

- What is cool here is those ip addresses in the above image are the ones of the nginx pods from the deployments, we can see this like so [6b] :

```
→  task6 git:(main) ✗ kubectl get pods -o wide
NAME                                          READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED NODE   READINESS GATES
nginx-deployment-pes1ug20cs237-8cf4bf97-2jcdz   1/1     Running   0          17m   10.244.0.11   minikube   <none>           <none>
nginx-deployment-pes1ug20cs237-8cf4bf97-g9f72   1/1     Running   0          29m   10.244.0.10   minikube   <none>           <none>
nginx-deployment-pes1ug20cs237-8cf4bf97-wxkt5   1/1     Running   0          33m   10.244.0.9    minikube   <none>           <none>
→  task6 git:(main) ✗ _
```

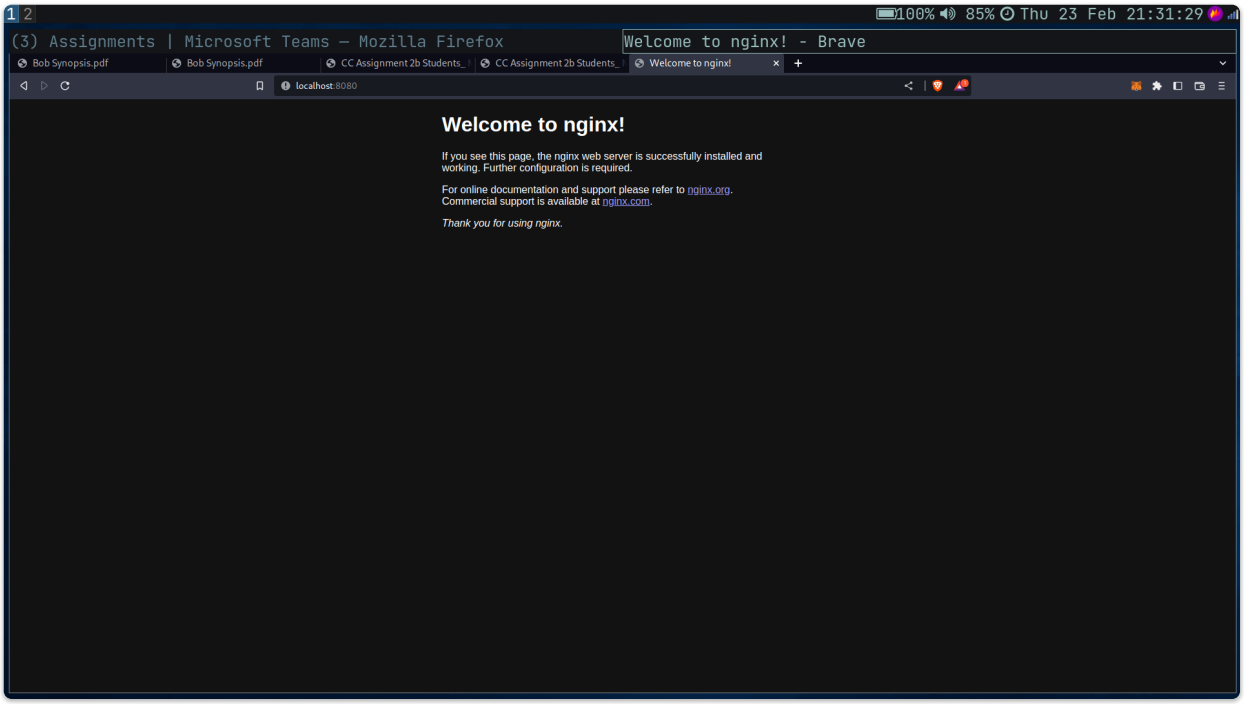
Task 7 : Port forwarding

- We can portforward any port to the service manually like so [7a]:

```
→  task6 git:(main) ✗ kubectl port-forward service/nginx-service-pes2ug20cs237 8080:8080
Forwarding from 127.0.0.1:8080 → 80
Forwarding from [::1]:8080 → 80
Handling connection for 8080
```

> ⓘ **The service exposes 8080 which redirects to 80 in pods. We are exposing 8080 of service to 8080 of host. Hence 8080->80**

- Here is the access of the webpage on localhost [7b]:



Task 8 : Clean up

- Lets first delete our deployments and services [8a]:

```
→  task6 git:(main) ✗ kubectl delete services nginx-service-pes2ug20cs237
service "nginx-service-pes2ug20cs237" deleted
→  task6 git:(main) ✗ kubectl delete deployments nginx-deployment-pes1ug20cs237
deployment.apps "nginx-deployment-pes1ug20cs237" deleted
→  task6 git:(main) ✗ _
```

- Now we can stop minikube [8b]:

```
→  task6 git:(main) ✗ minikube stop
✋  Stopping node "minikube"  ...
🛑  Powering off "minikube" via SSH ...
🛑  1 node stopped.
→  task6 git:(main) ✗
```

Task 9 : Self task

- Start your minikube cluster once again.

- Lets once again create an nginx deployment, the default deployment will do.

```
➜  task4 git:(main) ✗ minikube start
😄  minikube v1.29.0 on Arch
✨  Using the docker driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.26.1 on Docker 20.10.23 ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
➜  task4 git:(main) ✗ kubectl create deployment nginx-pes2ug20cs237 --image=nginx  --port=80
deployment.apps/nginx-pes2ug20cs237 created
```

- Now create a NodePort service with target port as 80 [9a] :

```
➜  task4 git:(main) ✗ kubectl expose pod nginx-pes2ug20cs237-cb884c689-2hp9f --type=NodePort --port=8080 --target-port=80
service/nginx-pes2ug20cs237-cb884c689-2hp9f exposed
```

- We can now see the ports exposed in the service as so [9b] :

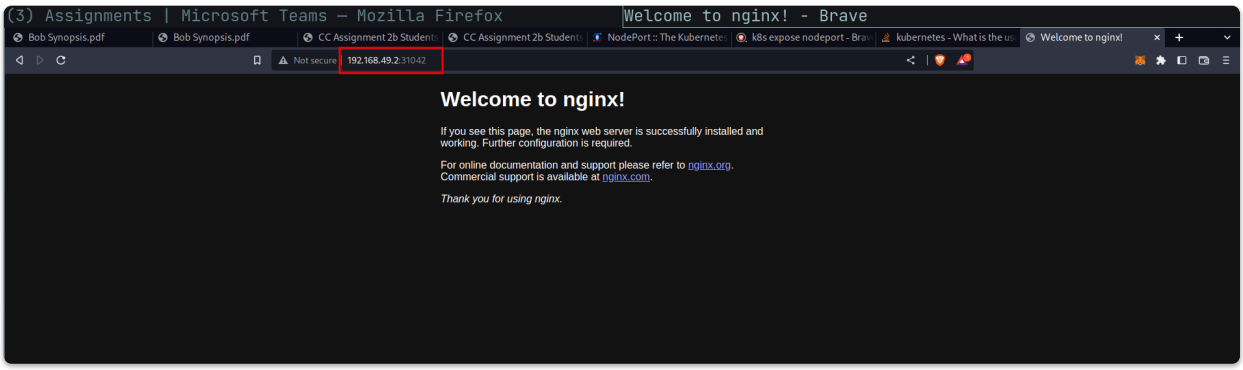
```
➜  task4 git:(main) ✗ kubectl get service -o wide
NAME                                  TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE     SELECTOR
kubernetes                            ClusterIP   10.96.0.1       <none>        443/TCP          4h14m   <none>
nginx-pes2ug20cs237-cb884c689-2hp9f   NodePort    10.107.95.172   <none>        8080:31042/TCP   3m16s   app=nginx-pes2ug20cs237,pod-templat
=cb884c689
➜  task4 git:(main) ✗ kubectl describe services nginx-pes2ug20cs237-cb884c689-2hp9f
Name:                     nginx-pes2ug20cs237-cb884c689-2hp9f
Namespace:                default
Labels:                   app=nginx-pes2ug20cs237
                          pod-template-hash=cb884c689
Annotations:              <none>
Selector:                 app=nginx-pes2ug20cs237,pod-template-hash=cb884c689
Type:                     NodePort
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.107.95.172
IPs:                      10.107.95.172
Port:                     <unset>  8080/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  31042/TCP
Endpoints:                10.244.0.13:80
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
```

> ⓘ **In the above ports : `Port` is the port through which other serives in the name space can access this pod. TargetPort is the port that is accessed in the pod when a requests is come in. Node port is mean to external access.**

- One can check the minikube ip address as so [9c]:

```
➜  task4 git:(main) ✗ minikube ip
192.168.49.2
➜  task4 git:(main) ✗ _
```

- And now try opening the minikube ip address with the Nodeport port, like so [9d]:


(3) Assignments | Microsoft Teams – Mozilla Firefox
Welcome to nginx! - Brave
Bob Synopsis.pdf
Bob Synopsis.pdf
CC Assignment 2b Students
CC Assignment 2b Students
NodePort :: The Kubernetes
k8s expose nodeport - Brav
kubernetes - What is the us
Welcome to nginx!
Not secure
192.168.49.2:31042
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.