Assignment 2b - KUBERNETES

Submitted by: Mahika Gupta

SRN: PES1UG20CS243

Date: 25/02/2023

Tasks:

1 **Installation**

2 **Creating pods and Deployments**

3 **Debugging pods**

4 **Applying configuration files**

5 **Self-healing feature**

6 **Connecting Services to Deployments**

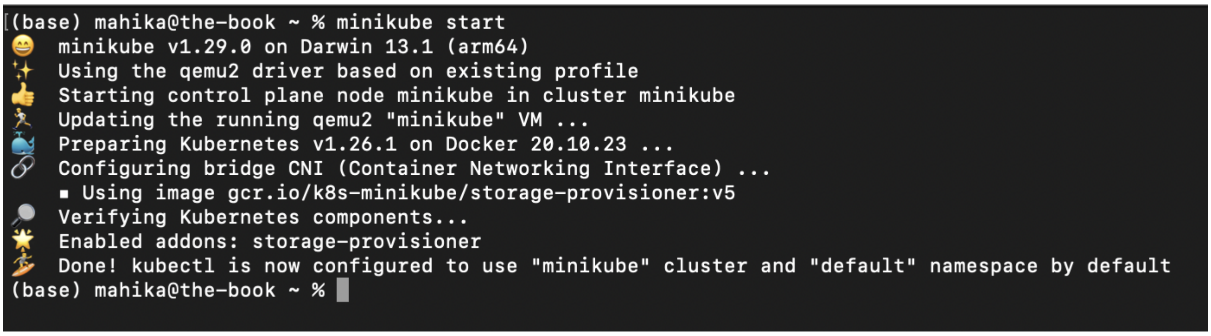
7 **Exposing a Service externally**

8 **Deletion and Cleanup**

9 **Exposing an external IP address to access an Application in a cluster**

Screenshot 1a:

```
minikube start
```



```
(base) mahika@the-book ~ % minikube start
😄  minikube v1.29.0 on Darwin 13.1 (arm64)
✨  Using the qemu2 driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🏃  Updating the running qemu2 "minikube" VM ...
🐳  Preparing Kubernetes v1.26.1 on Docker 20.10.23 ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🔎  Verifying Kubernetes components...
🌟  Enabled addons: storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
(base) mahika@the-book ~ % █
```

Quick Points:

1. Ensure Docker is installed, up and running before using minikube.
2. Pods may take some time (a few minutes in a bad network) initially to start up, this is normal.
3. All resources mentioned in all the tasks must be created only in the **default Kubernetes namespace**, modifying other namespaces such as kube-system may cause Kubernetes to stop working.

Assignment 2 deliverables:

1. Section 1: Installation
   - Screenshot 1a - Minikube running successfully
2. Section 2: Creating pods and deployments, Editing them and observing Rollback:-
   - Screenshot 2a - get nodes, pod and services command.
   - Screenshot 2b- Deployment created.
   - Screenshot 2c- get deployment and pod command .
   - Screenshot 2d- editing '-image:nginx.'
   - Screenshot 2e- showing edited deployment.
   - Screenshot 2f- deployment is rolled back.
   - Screenshot 2g- showing original nginx image.
3. Section 3:Debugging Pods:-
   - Screenshot 3a - Kubectl logs displayed.
   - Screenshot 3b- Kubectl 'describe pod ' command.
   - Screenshot 3c - Create mongo deployment.
   - Screenshot 3d - Delete both requirements.
4. Section 4: Applying configuration files:-
   - Screenshot 4a - Kubectl apply command on yaml file.
   - Screenshot 4b- Kubectl get on yaml file
5. Section 5: Delete a pod to observe the self-healing feature.
   - Screenshot 5a - Deleted pod:-
6. Section 6 : Connecting Services to Deployments
   - Screenshot 6a- Kubectl apply and get command.
   - Screenshot 6b-kubectl get pod -o wide command
7. Section 7: Port Forwarding:-
   - Screenshot 7a -Kubectl port-forward command
   - Screenshot 7b- Display welcome to nginx on web page
8. Section 8: Deleting service/deployment and Cleanup
   - Screenshot 8a - Delete nginx deployments
   - Screenshot 8b - stop minikube
9. Section 9: Expose an external IP address to access an Application in a cluster
   - Screenshot 9a- the command which exposes specifies the type of service (NodePort)
   - Screenshot 9b - kubectl get service command which displays the node port
   - Screenshot 9c - minikube IP address
   - Screenshot 9d - the webpage with the IP Address visible. (If the IP Address is not visible in the screenshot, you will lose significant portion of marks w.r.t. Section 9)

Section 2: Creating pods and deployments, Editing them and observing Rollback

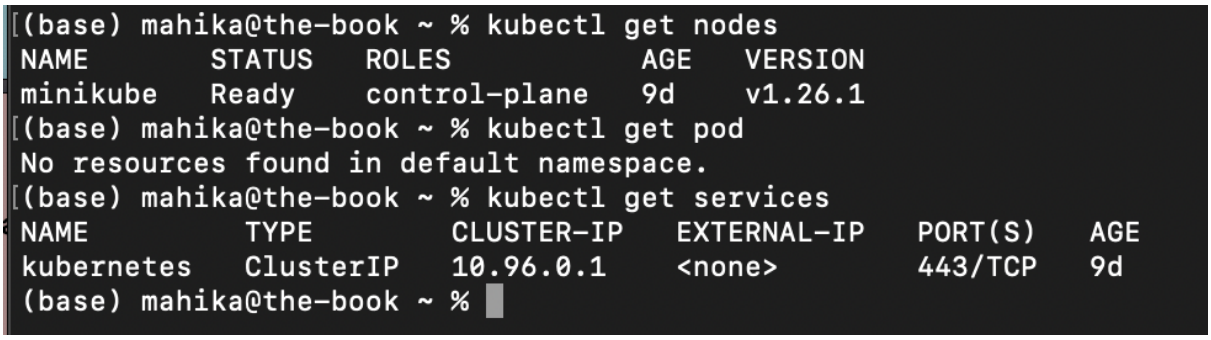
Firstly, view the nodes, pods and services present currently.

```
kubectl get nodes
```

```
kubectl get pod
```

```
kubectl get services
```

Screenshot 2a:

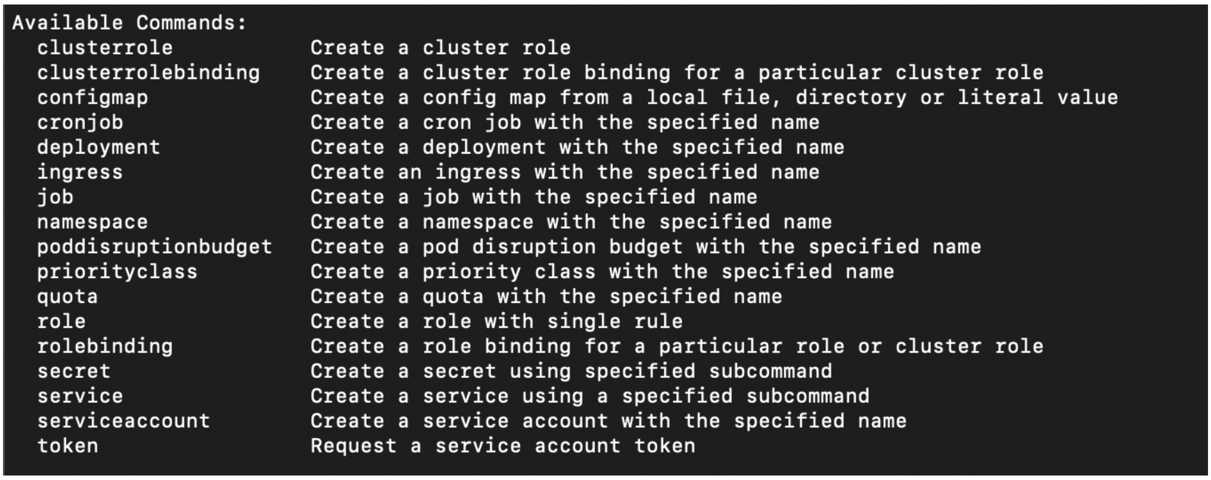


```
[(base) mahika@the-book ~ % kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   9d    v1.26.1
[(base) mahika@the-book ~ % kubectl get pod
No resources found in default namespace.
[(base) mahika@the-book ~ % kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   9d
(base) mahika@the-book ~ %
```

Currently, only one default service is running. We will explore how to create a service later on.

To see what all we can create using kubectl:

```
kubectl create -h
```



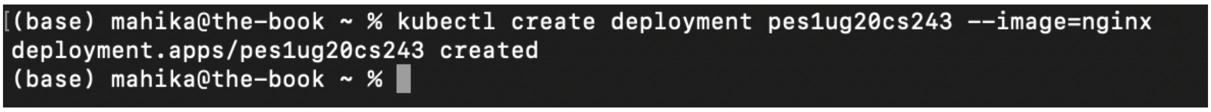
```
Available Commands:
  clusterrole           Create a cluster role
  clusterrolebinding    Create a cluster role binding for a particular cluster role
  configmap             Create a config map from a local file, directory or literal value
  cronjob               Create a cron job with the specified name
  deployment            Create a deployment with the specified name
  ingress               Create an ingress with the specified name
  job                   Create a job with the specified name
  namespace             Create a namespace with the specified name
  poddisruptionbudget   Create a pod disruption budget with the specified name
  priorityclass         Create a priority class with the specified name
  quota                 Create a quota with the specified name
  role                  Create a role with single rule
  rolebinding           Create a role binding for a particular role or cluster role
  secret                Create a secret using specified subcommand
  service               Create a service using a specified subcommand
  serviceaccount        Create a service account with the specified name
  token                 Request a service account token
```

Create a deployment with the name of the deployment being your SRN and the image nginx

```
kubectl create deployment pes1ug20csxxx --image=nginx
```

*SRN has to be in lowercase.

Screenshot 2b:



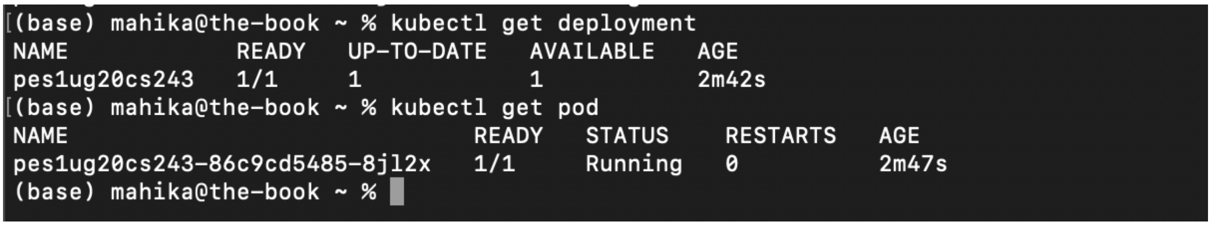
```
[(base) mahika@the-book ~ % kubectl create deployment pes1ug20cs243 --image=nginx
deployment.apps/pes1ug20cs243 created
(base) mahika@the-book ~ %
```

This command downloads the latest nginx image from DockerHub

Now run

`kubectl get deployment` and `kubectl get pod`

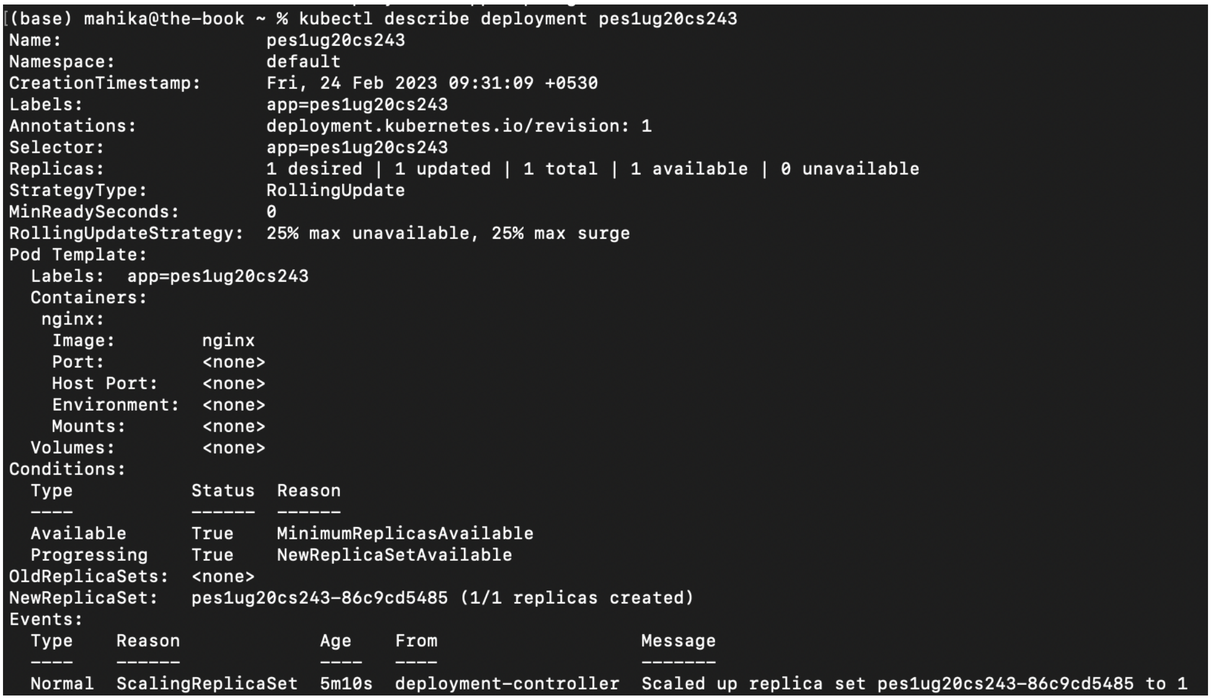
Screenshot 2c:



```
[(base) mahika@the-book ~ % kubectl get deployment
 NAME            READY   UP-TO-DATE   AVAILABLE   AGE
 pes1ug20cs243   1/1     1            1           2m42s
[(base) mahika@the-book ~ % kubectl get pod
 NAME                             READY   STATUS    RESTARTS   AGE
 pes1ug20cs243-86c9cd5485-8jl2x   1/1     Running   0          2m47s
 (base) mahika@the-book ~ %
```

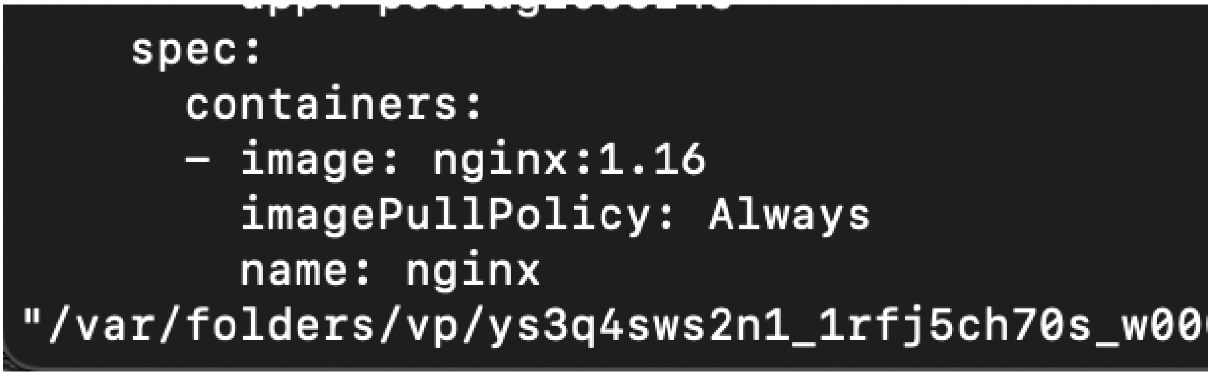
To see further details about the deployment, run

`kubectl describe deployment pes1ug20csxxx`



```
[(base) mahika@the-book ~ % kubectl describe deployment pes1ug20cs243
Name:                   pes1ug20cs243
Namespace:              default
CreationTimestamp:      Fri, 24 Feb 2023 09:31:09 +0530
Labels:                 app=pes1ug20cs243
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=pes1ug20cs243
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=pes1ug20cs243
  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:   pes1ug20cs243-86c9cd5485 (1/1 replicas created)
Events:
  Type    Reason             Age    From                   Message
  ----    ------             ----   ----                   -------
  Normal  ScalingReplicaSet  5m10s  deployment-controller  Scaled up replica set pes1ug20cs243-86c9cd5485 to 1
```

Screenshot 2d:

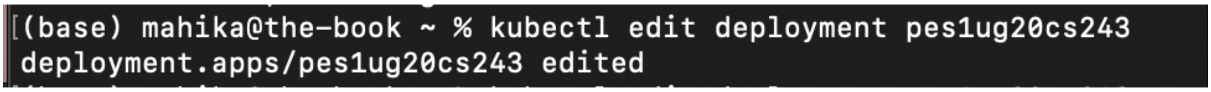


```
    spec:
      containers:
      - image: nginx:1.16
        imagePullPolicy: Always
        name: nginx
"/var/folders/vp/ys3q4sws2n1_1rfj5ch70s_w00
```

- Then type the Esc key followed by :wq followed by Enter key. This ensures that your edit is saved

- Incase you wish to exit without saving the changes: Esc key followed by typing ‘q!’ followed Enter key.
- ‘x’ is the key used for Backspace. Make sure to use the Esc key before using the ‘x’ key.

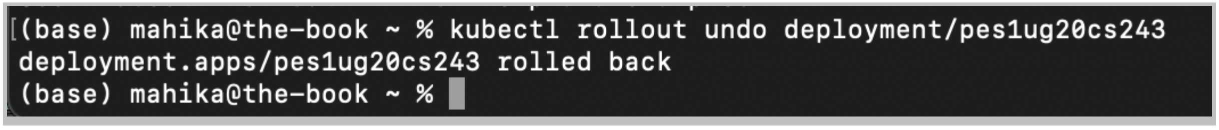
Screenshot 2e:



```
[(base) mahika@the-book ~ % kubectl edit deployment pes1ug20cs243
deployment.apps/pes1ug20cs243 edited
```

Screenshot 2f:

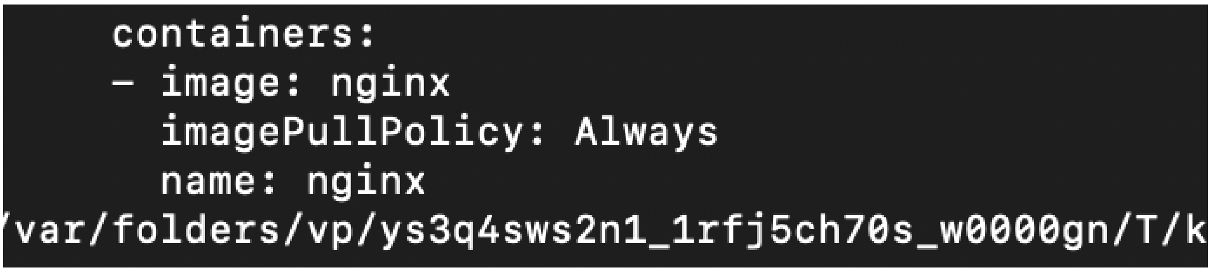
```
kubectl rollout undo deployment/pes1ug20csxxx
```



```
[(base) mahika@the-book ~ % kubectl rollout undo deployment/pes1ug20cs243
deployment.apps/pes1ug20cs243 rolled back
(base) mahika@the-book ~ %
```

Observe the version has been reverted from 1.16 to the latest version

Screenshot 2g:

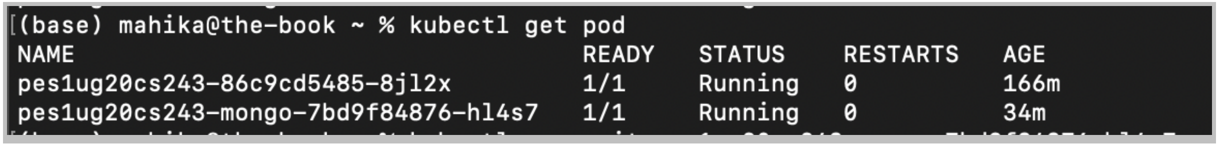


```
    containers:
    - image: nginx
      imagePullPolicy: Always
      name: nginx
/var/folders/vp/ys3q4sws2n1_1rfj5ch70s_w0000gn/T/k
```

Section 3: Debugging pods

A common way to debug is to look at logs. Copy paste your pod name after running the below command and run command 3a by replacing the pod name with yours. Please note that the pod name would have changed after editing the configuration file.

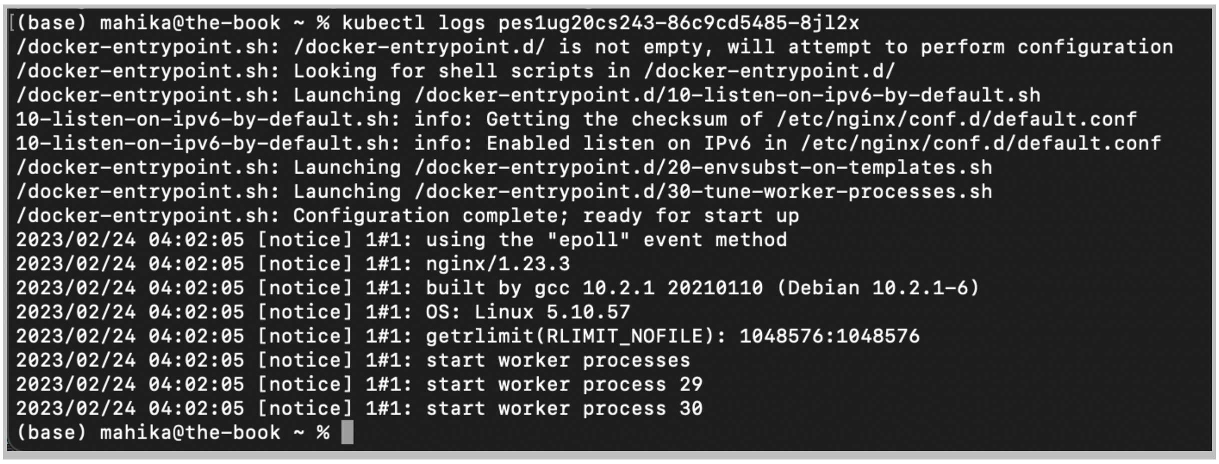
```
kubectl get pod
```



```
[(base) mahika@the-book ~ % kubectl get pod
NAME                                 READY   STATUS    RESTARTS   AGE
pes1ug20cs243-86c9cd5485-8jl2x       1/1     Running   0          166m
pes1ug20cs243-mongo-7bd9f84876-hl4s7 1/1     Running   0          34m
```

Screenshot 3a:

```
kubectl logs <pod_name>
```

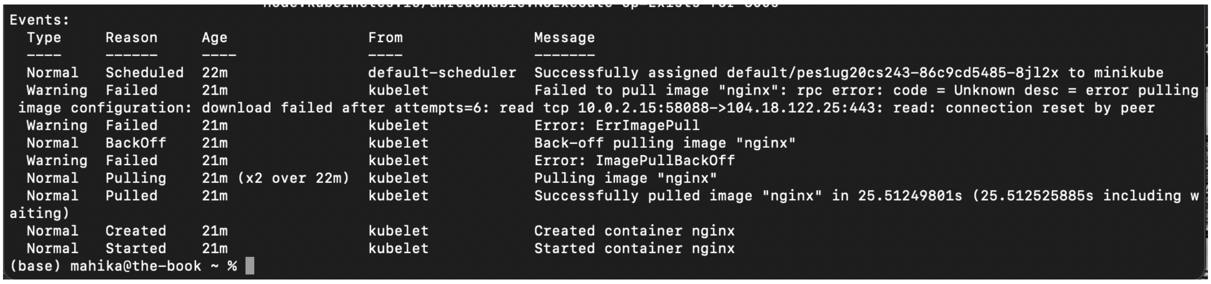


```
(base) mahika@the-book ~ % kubectl logs pes1ug20cs243-86c9cd5485-8jl2x
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
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
2023/02/24 04:02:05 [notice] 1#1: using the "epoll" event method
2023/02/24 04:02:05 [notice] 1#1: nginx/1.23.3
2023/02/24 04:02:05 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/02/24 04:02:05 [notice] 1#1: OS: Linux 5.10.57
2023/02/24 04:02:05 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/02/24 04:02:05 [notice] 1#1: start worker processes
2023/02/24 04:02:05 [notice] 1#1: start worker process 29
2023/02/24 04:02:05 [notice] 1#1: start worker process 30
(base) mahika@the-book ~ %
```

You can see all the state changes for a pod to debug it in the following manner (Under the "Events" section – scroll down to view it):

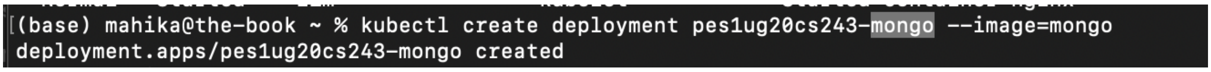
```
kubectl describe pod <pod_name>
```

Screenshot 3b: (Screenshot of the "Events" section



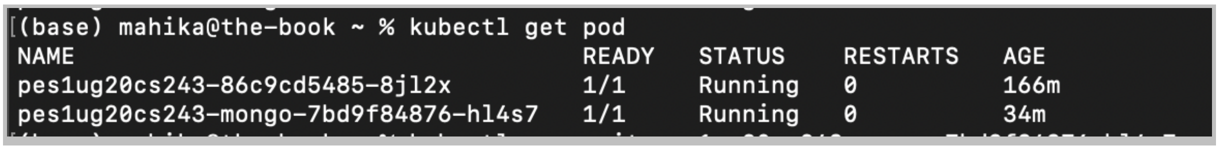
```
Events:
  Type     Reason     Age                 From               Message
  ----     ------     ----                ----               -------
  Normal   Scheduled  22m                 default-scheduler  Successfully assigned default/pes1ug20cs243-86c9cd5485-8jl2x to minikube
  Warning  Failed     21m                 kubelet            Failed to pull image "nginx": rpc error: code = Unknown desc = error pulling
 image configuration: download failed after attempts=6: read tcp 10.0.2.15:58088->104.18.122.25:443: read: connection reset by peer
  Warning  Failed     21m                 kubelet            Error: ErrImagePull
  Normal   BackOff    21m                 kubelet            Back-off pulling image "nginx"
  Warning  Failed     21m                 kubelet            Error: ImagePullBackOff
  Normal   Pulling    21m (x2 over 22m)   kubelet            Pulling image "nginx"
  Normal   Pulled     21m                 kubelet            Successfully pulled image "nginx" in 25.51249801s (25.512525885s including w
aiting)
  Normal   Created    21m                 kubelet            Created container nginx
  Normal   Started    21m                 kubelet            Started container nginx
(base) mahika@the-book ~ %
```

```
kubectl create deployment pes1ug20csxx-mongo --image=mongo
```



```
(base) mahika@the-book ~ % kubectl create deployment pes1ug20cs243-mongo --image=mongo
deployment.apps/pes1ug20cs243-mongo created
```

```
kubectl get pod
```



```
(base) mahika@the-book ~ % kubectl get pod
NAME                                  READY   STATUS    RESTARTS   AGE
pes1ug20cs243-86c9cd5485-8jl2x        1/1     Running   0          166m
pes1ug20cs243-mongo-7bd9f84876-hl4s7  1/1     Running   0          34m
```

Wait till the container is created and the pod is in Running State. Once it is, run the following command to get the mongo terminal:

Screenshot 3c:

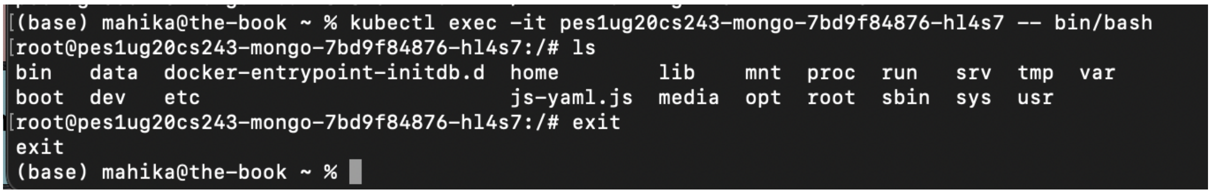
```
kubectl exec -it <pod_name> -- bin/bash
```

`-it` stands for Interactive terminal

Run

```
ls
```

```
exit
```

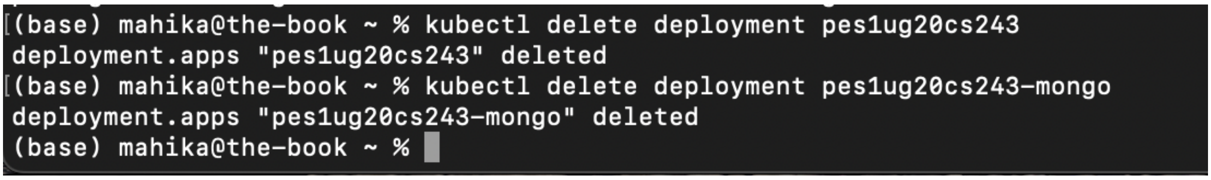


```
[(base) mahika@the-book ~ % kubectl exec -it pes1ug20cs243-mongo-7bd9f84876-hl4s7 -- bin/bash
[root@pes1ug20cs243-mongo-7bd9f84876-hl4s7:/# ls
 bin   data  docker-entrypoint-initdb.d  home        lib    mnt  proc  run   srv  tmp  var
 boot  dev   etc                         js-yaml.js  media  opt  root  sbin  sys  usr
[root@pes1ug20cs243-mongo-7bd9f84876-hl4s7:/# exit
 exit
 (base) mahika@the-book ~ %
```

`kubectl delete deployment <deployment_name>`

Delete both your deployments. Now run kubectl get pod and observe the output.

Screenshot 3d:



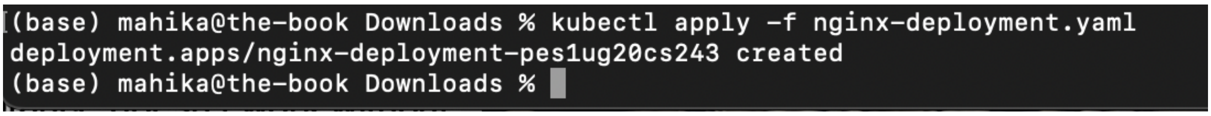
```
[(base) mahika@the-book ~ % kubectl delete deployment pes1ug20cs243
deployment.apps "pes1ug20cs243" deleted
[(base) mahika@the-book ~ % kubectl delete deployment pes1ug20cs243-mongo
deployment.apps "pes1ug20cs243-mongo" deleted
(base) mahika@the-book ~ %
```

Section 4: Applying configuration files and Scaling

Screenshot 4a:

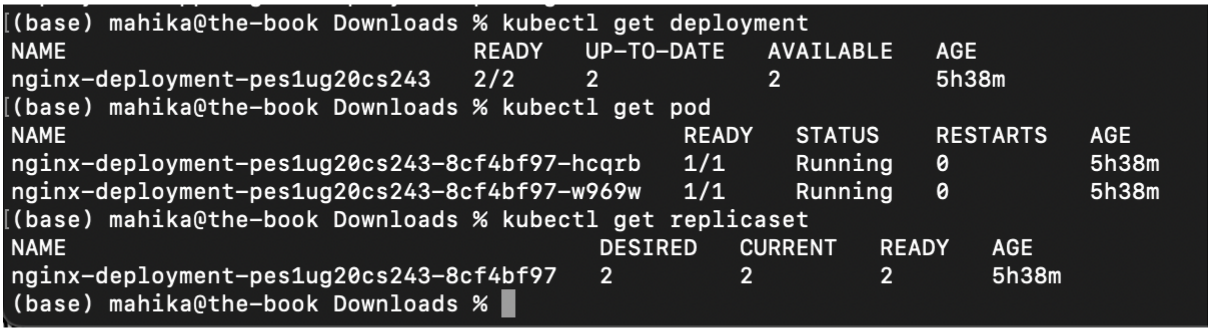
`kubectl apply -f <filename>`

Note: For Windows, the filename should be within double quotes



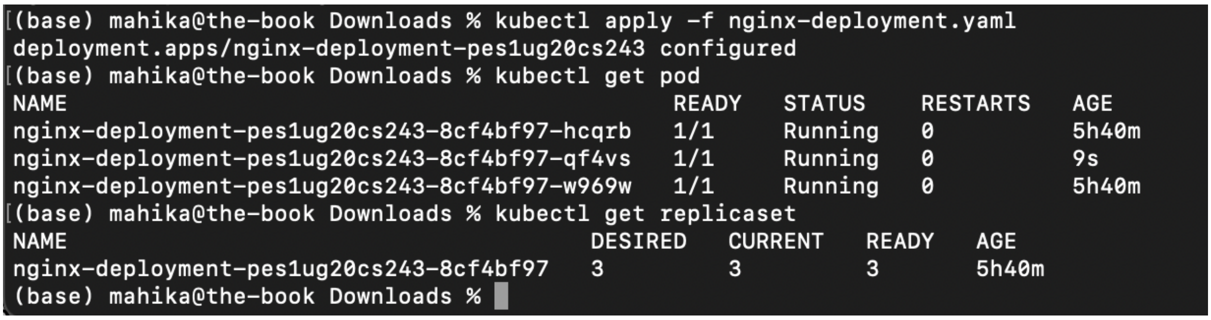
```
[(base) mahika@the-book Downloads % kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment-pes1ug20cs243 created
(base) mahika@the-book Downloads %
```

Check the pods, deployment and replicaset.



```
[(base) mahika@the-book Downloads % kubectl get deployment
NAME                             READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment-pes1ug20cs243   2/2     2            2           5h38m
[(base) mahika@the-book Downloads % kubectl get pod
NAME                                              READY   STATUS    RESTARTS   AGE
nginx-deployment-pes1ug20cs243-8cf4bf97-hcqrb   1/1     Running   0          5h38m
nginx-deployment-pes1ug20cs243-8cf4bf97-w969w   1/1     Running   0          5h38m
[(base) mahika@the-book Downloads % kubectl get replicaset
NAME                                      DESIRED   CURRENT   READY   AGE
nginx-deployment-pes1ug20cs243-8cf4bf97   2         2         2       5h38m
(base) mahika@the-book Downloads %
```

Now change the replicas to 3 in the file and run the command again. (Line 8 in the .yaml file). Notice that it says “configured” and not “created” this time. Check the pods and replicaset again.

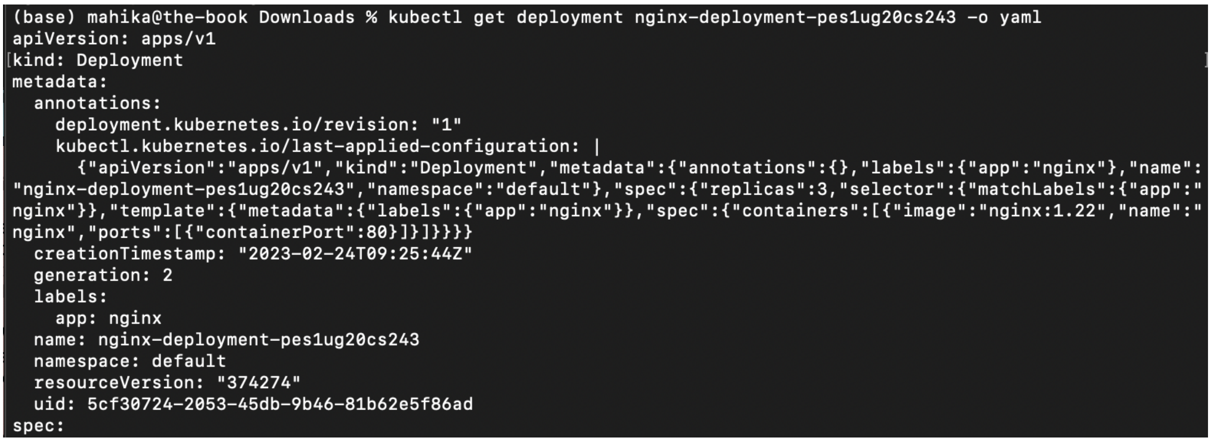


```
[(base) mahika@the-book Downloads % kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment-pes1ug20cs243 configured
[(base) mahika@the-book Downloads % kubectl get pod
NAME                                            READY   STATUS    RESTARTS   AGE
nginx-deployment-pes1ug20cs243-8cf4bf97-hcqrb   1/1     Running   0          5h40m
nginx-deployment-pes1ug20cs243-8cf4bf97-qf4vs   1/1     Running   0          9s
nginx-deployment-pes1ug20cs243-8cf4bf97-w969w   1/1     Running   0          5h40m
[(base) mahika@the-book Downloads % kubectl get replicaset
NAME                                      DESIRED   CURRENT   READY   AGE
nginx-deployment-pes1ug20cs243-8cf4bf97   3         3         3       5h40m
(base) mahika@the-book Downloads %
```

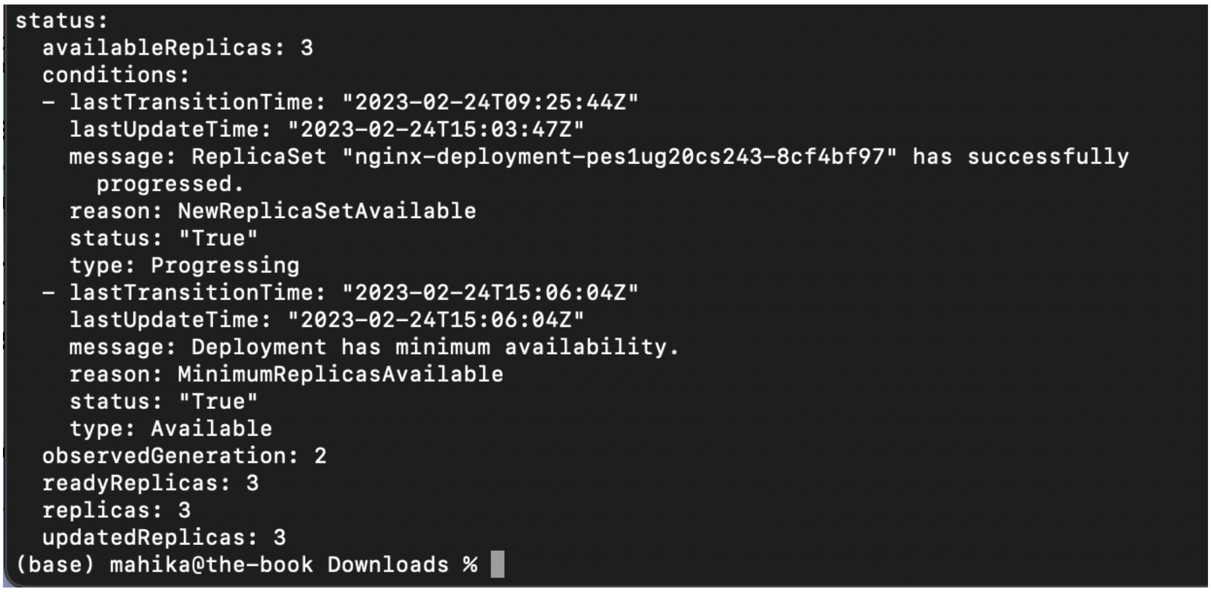
There are 3 parts to any configuration file as discussed above. The third part is Status. It is added to the configuration file automatically by Kubernetes and it can be viewed by:

Screenshot 4b:

```
kubectl get deployment nginx-deployment-pes1ug20csxxx -o yaml
```



```
(base) mahika@the-book Downloads % kubectl get deployment nginx-deployment-pes1ug20cs243 -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-pes1ug20cs243","namespace":"default"},"spec":{"replicas":3,"selector":{"matchLabels":{"app":"
nginx"}},"template":{"metadata":{"labels":{"app":"nginx"}},"spec":{"containers":[{"image":"nginx:1.22","name":"
nginx","ports":[{"containerPort":80}]}]}}}}
  creationTimestamp: "2023-02-24T09:25:44Z"
  generation: 2
  labels:
    app: nginx
  name: nginx-deployment-pes1ug20cs243
  namespace: default
  resourceVersion: "374274"
  uid: 5cf30724-2053-45db-9b46-81b62e5f86ad
spec:
```



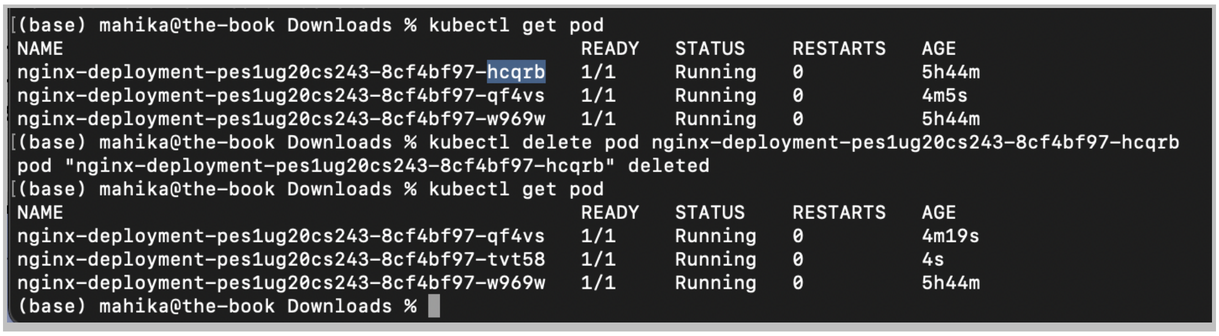
```
status:
  availableReplicas: 3
  conditions:
  - lastTransitionTime: "2023-02-24T09:25:44Z"
    lastUpdateTime: "2023-02-24T15:03:47Z"
    message: ReplicaSet "nginx-deployment-pes1ug20cs243-8cf4bf97" has successfully
      progressed.
    reason: NewReplicaSetAvailable
    status: "True"
    type: Progressing
  - lastTransitionTime: "2023-02-24T15:06:04Z"
    lastUpdateTime: "2023-02-24T15:06:04Z"
    message: Deployment has minimum availability.
    reason: MinimumReplicasAvailable
    status: "True"
    type: Available
  observedGeneration: 2
  readyReplicas: 3
  replicas: 3
  updatedReplicas: 3
(base) mahika@the-book Downloads %
```

Section 5: Delete a pod to observe the self-healing feature.

Screenshot 5a: (Should contain the list of pods before deleting, deletion command and list of pods after deletion)

```
kubectl delete pod <pod_name>
```

```
kubectl get pod
```



```
[(base) mahika@the-book Downloads % kubectl get pod
 NAME                                          READY   STATUS    RESTARTS   AGE
 nginx-deployment-pes1ug20cs243-8cf4bf97-hcqrb   1/1     Running   0          5h44m
 nginx-deployment-pes1ug20cs243-8cf4bf97-qf4vs   1/1     Running   0          4m5s
 nginx-deployment-pes1ug20cs243-8cf4bf97-w969w   1/1     Running   0          5h44m
[(base) mahika@the-book Downloads % kubectl delete pod nginx-deployment-pes1ug20cs243-8cf4bf97-hcqrb
 pod "nginx-deployment-pes1ug20cs243-8cf4bf97-hcqrb" deleted
[(base) mahika@the-book Downloads % kubectl get pod
 NAME                                          READY   STATUS    RESTARTS   AGE
 nginx-deployment-pes1ug20cs243-8cf4bf97-qf4vs   1/1     Running   0          4m19s
 nginx-deployment-pes1ug20cs243-8cf4bf97-tvt58   1/1     Running   0          4s
 nginx-deployment-pes1ug20cs243-8cf4bf97-w969w   1/1     Running   0          5h44m
 (base) mahika@the-book Downloads %
```

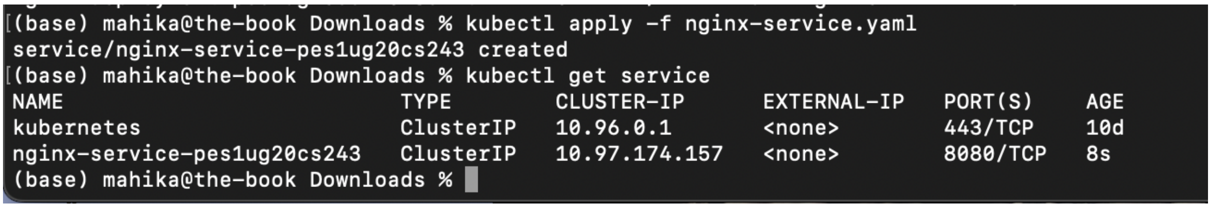
Notice that the pod has been replaced. This is part of the self-healing feature of Kubernetes.

Section 6: Connecting Services to Deployments

```
kubectl apply -f <filename>
```

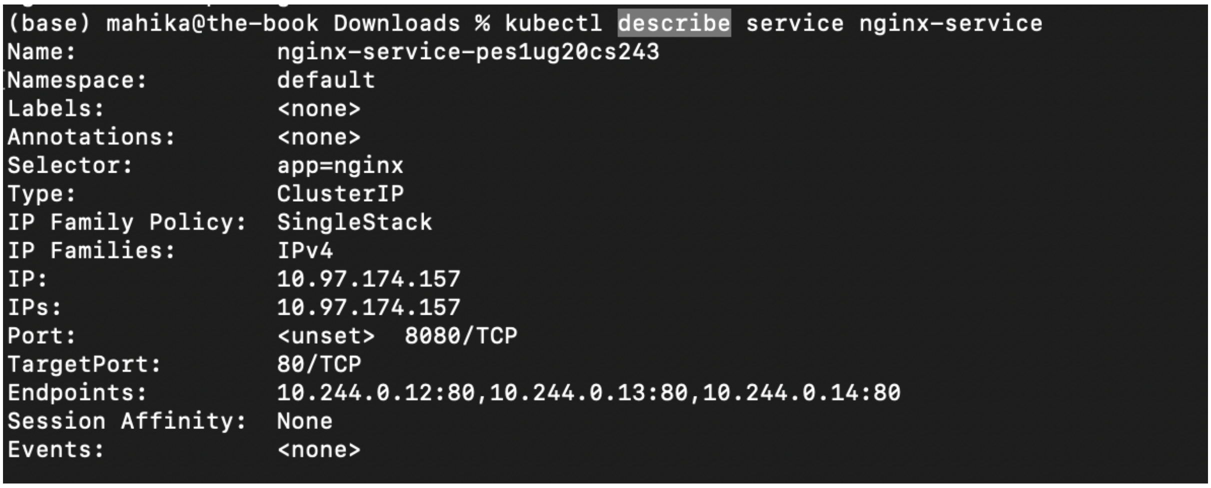
```
kubectl get service
```

Screenshot 6a:



```
[(base) mahika@the-book Downloads % kubectl apply -f nginx-service.yaml
 service/nginx-service-pes1ug20cs243 created
[(base) mahika@the-book Downloads % kubectl get service
 NAME                          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
 kubernetes                    ClusterIP   10.96.0.1       <none>        443/TCP    10d
 nginx-service-pes1ug20cs243   ClusterIP   10.97.174.157   <none>        8080/TCP   8s
 (base) mahika@the-book Downloads %
```

```
kubectl describe service nginx-service
```

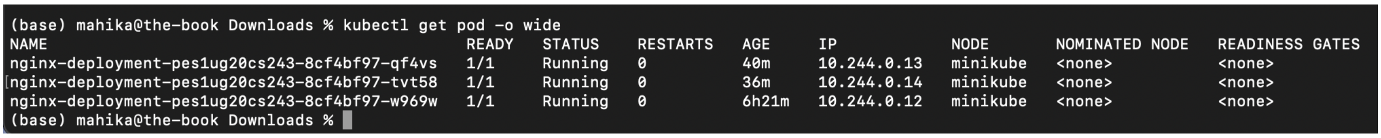


```
(base) mahika@the-book Downloads % kubectl describe service nginx-service
Name:              nginx-service-pes1ug20cs243
Namespace:         default
Labels:            <none>
Annotations:       <none>
Selector:          app=nginx
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.97.174.157
IPs:               10.97.174.157
Port:              <unset>  8080/TCP
TargetPort:        80/TCP
Endpoints:         10.244.0.12:80,10.244.0.13:80,10.244.0.14:80
Session Affinity:  None
Events:            <none>
```

This shows the end points of the service. To see which pod it forwards the requests to, we can look at individual pods' information using the command:

```
kubectl get pod -o wide
```

Screenshot 6b:

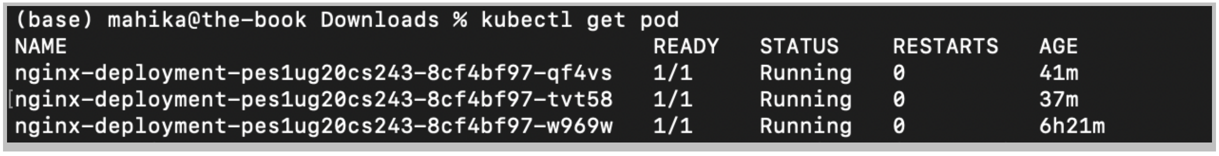


```
(base) mahika@the-book Downloads % kubectl get pod -o wide
NAME                                          READY   STATUS    RESTARTS   AGE    IP            NODE       NOMINATED NODE   READINESS GATES
nginx-deployment-pes1ug20cs243-8cf4bf97-qf4vs   1/1     Running   0          40m    10.244.0.13   minikube   <none>           <none>
nginx-deployment-pes1ug20cs243-8cf4bf97-tvt58   1/1     Running   0          36m    10.244.0.14   minikube   <none>           <none>
nginx-deployment-pes1ug20cs243-8cf4bf97-w969w   1/1     Running   0          6h21m  10.244.0.12   minikube   <none>           <none>
(base) mahika@the-book Downloads %
```

Section 7: Port Forwarding:k

Make sure all pods of the deployment are up and running by running

```
kubectl get pod
```

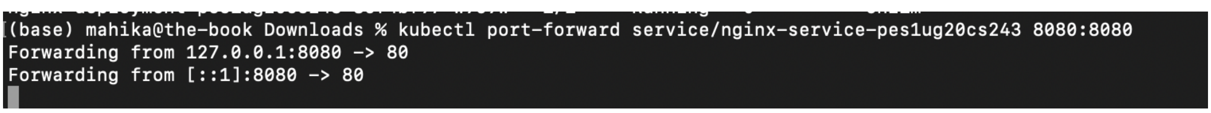


```
(base) mahika@the-book Downloads % kubectl get pod
NAME                                          READY   STATUS    RESTARTS   AGE
nginx-deployment-pes1ug20cs243-8cf4bf97-qf4vs   1/1     Running   0          41m
nginx-deployment-pes1ug20cs243-8cf4bf97-tvt58   1/1     Running   0          37m
nginx-deployment-pes1ug20cs243-8cf4bf97-w969w   1/1     Running   0          6h21m
```

Expose the service using the command:

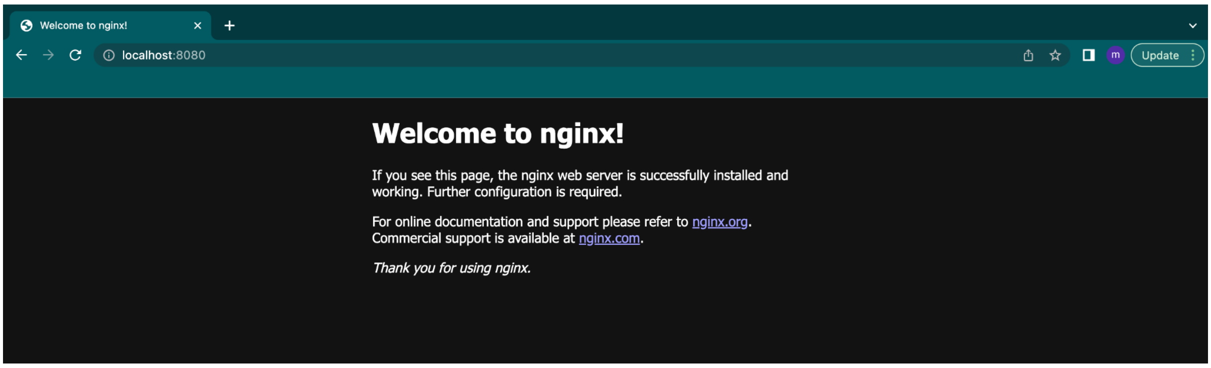
Screenshot 7a:

```
kubectl port-forward service/nginx-service-pes1ug20csxxx 8080:8080
```



```
(base) mahika@the-book Downloads % kubectl port-forward service/nginx-service-pes1ug20cs243 8080:8080
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

Screenshot 7b:



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.

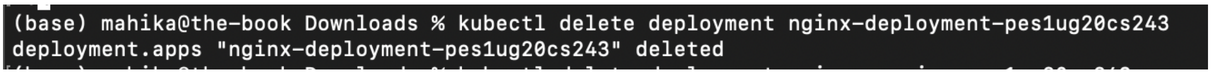
Thank you for using nginx.

Section 8: Deleting service/deployment and Cleanup

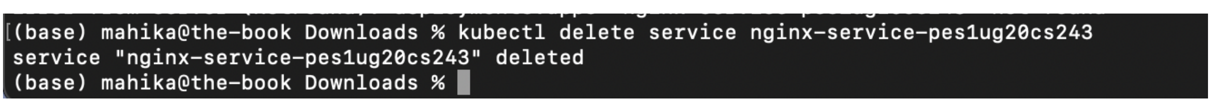
Screenshot 8a:

```
kubectl delete deployment nginx-deployment-pes1ug20csxx
```

```
kubectl delete service nginx-service-pes1ug20csxxx
```



```
(base) mahika@the-book Downloads % kubectl delete deployment nginx-deployment-pes1ug20cs243
deployment.apps "nginx-deployment-pes1ug20cs243" deleted
```

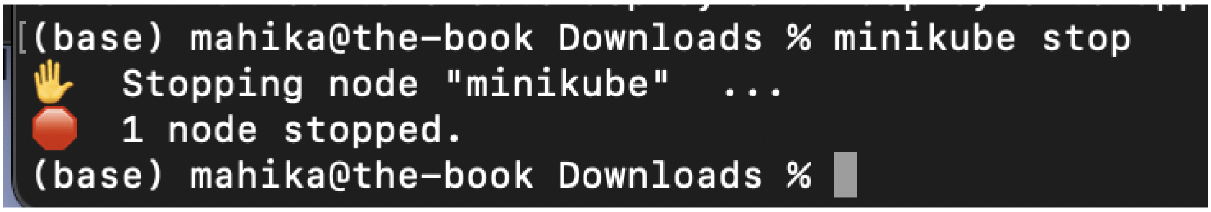


```
[(base) mahika@the-book Downloads % kubectl delete service nginx-service-pes1ug20cs243
service "nginx-service-pes1ug20cs243" deleted
(base) mahika@the-book Downloads % 
```

Run this command after Section 9!!

Screenshot 8b:

```
minikube stop
```



```
[(base) mahika@the-book Downloads % minikube stop
✋  Stopping node "minikube"  ...
🛑  1 node stopped.
(base) mahika@the-book Downloads % 
```

This ensures all resources are stopped and not seen when running the Kubernetes commands.

Section 9: Expose an external IP address to access an Application in a cluster
(To be done by the student)

Problem Statement:

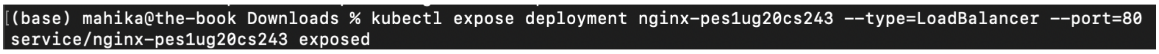
1) Create a deployment named nginx-<srn> with image as nginx.
2) Expose the deployment which automatically creates the service to be exposed
3) The service should be of type NodePort or LoadBalancer
4) Access the service on your browser using minikube ip address and the Nodeport or LoadBalancer and the respective port

Note:

1) CLI alone is sufficient to achieve this. .yaml configuration files are not required.
2) Do not try to display the webpage by using port-forwarding as demonstrated in Section 7. Note: Read the difference between the two ways for better understanding.
3) Make sure the firewall is turned off. Open the browser in incognito mode in case the page doesn't load.

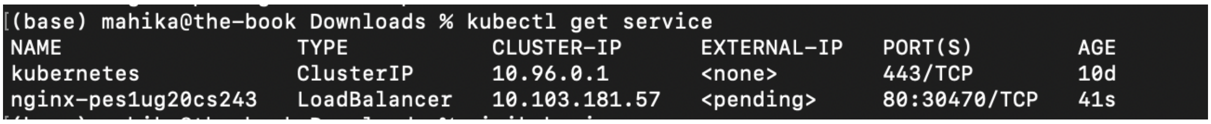
Deliverables of Section 9:

- **Screenshot 9a**: Screenshot of the command which exposes specifies the type of service (NodePort)



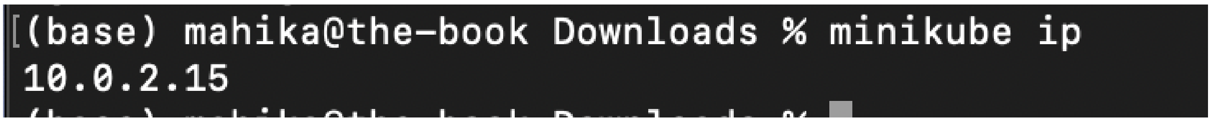
```
[(base) mahika@the-book Downloads % kubectl expose deployment nginx-pes1ug20cs243 --type=LoadBalancer --port=80
service/nginx-pes1ug20cs243 exposed
```

- **Screenshot 9b**: Screenshot of `kubectl get service command` which displays the node port



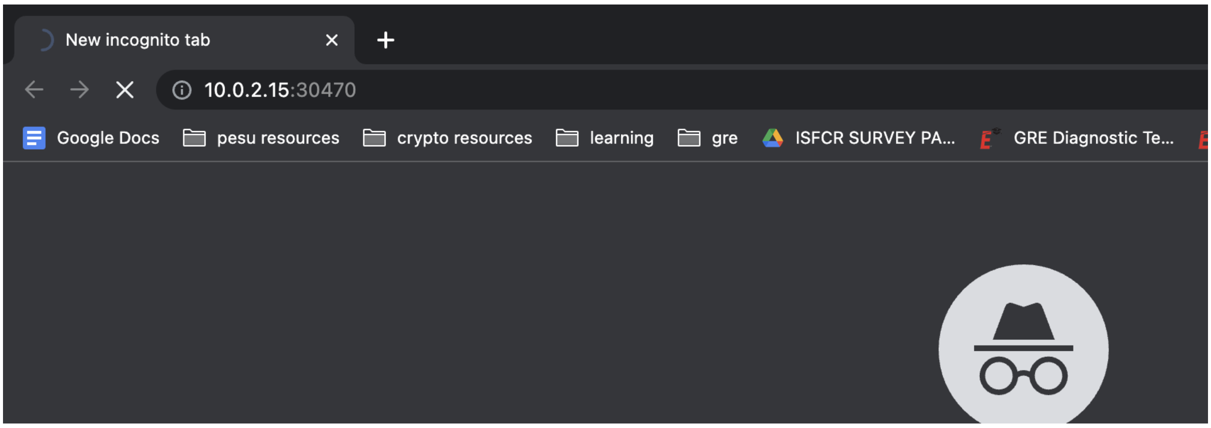
```
(base) mahika@the-book Downloads % kubectl get service
NAME                  TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes            ClusterIP      10.96.0.1       <none>        443/TCP        10d
nginx-pes1ug20cs243   LoadBalancer   10.103.181.57   <pending>     80:30470/TCP   41s
```

- **Screenshot 9c**: Screenshot of minikube IP address



```
(base) mahika@the-book Downloads % minikube ip
10.0.2.15
```

- **Screenshot 9d**: Screenshot of the webpage with the IP Address visible. (If the IP Address is not visible in the screenshot, you will lose significant portion of marks w.r.t. Section 9)



10.0.2.15:30470