Container Orchestration and Security

ASSIGNMENT – 2

SESSION: January 2026 – May 2026

SUBMITTED TO-

Dr. Hitesh Kumar Sharma

SUBMITTED BY –

NAME: Devanshi Jain

SAP ID: 500123468

ROLL NO.: R2142230278

Batch : 01 DevOps

SEMESTER: 6th

Scenario Description

An organization wants to deploy a production-ready web application on Kubernetes with the following requirements:

• Application must run inside a custom namespace

• Application should be deployed using a Deployment with multiple replicas

• Configuration must be managed using a ConfigMap

• Application data should persist using Volumes

• Application must be exposed using a Service

• Deployment should support scaling and rolling updates

• Application health must be verified

Tasks to be Performed

1. Create a custom Namespace

2. Create a ConfigMap to store application configuration

3. Create a Deployment with:

o Nginx container

o Multiple replicas

o Volume mount

4. Verify Pods and Replica management

5. Create a Service to expose the application

6. Scale the Deployment

7. Perform a Rolling Update

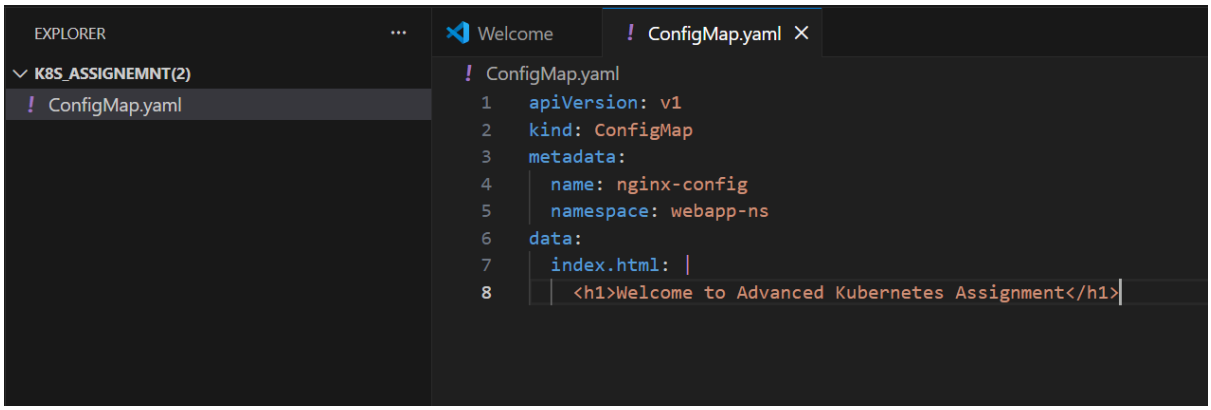
8. Validate application availability after updates

Github Link of both assignments(1 and 2) - https://github.com/Devanshii-git/COAS_Assignments.git

1. Create Custom Namespace

```
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl create namespace webapp-ns
namespace/webapp-ns created
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl get namespaces
NAME                   STATUS   AGE
default                Active   8d
kube-node-lease        Active   8d
kube-public            Active   8d
kube-system            Active   8d
kubernetes-dashboard   Active   8d
webapp-ns              Active   8s
```

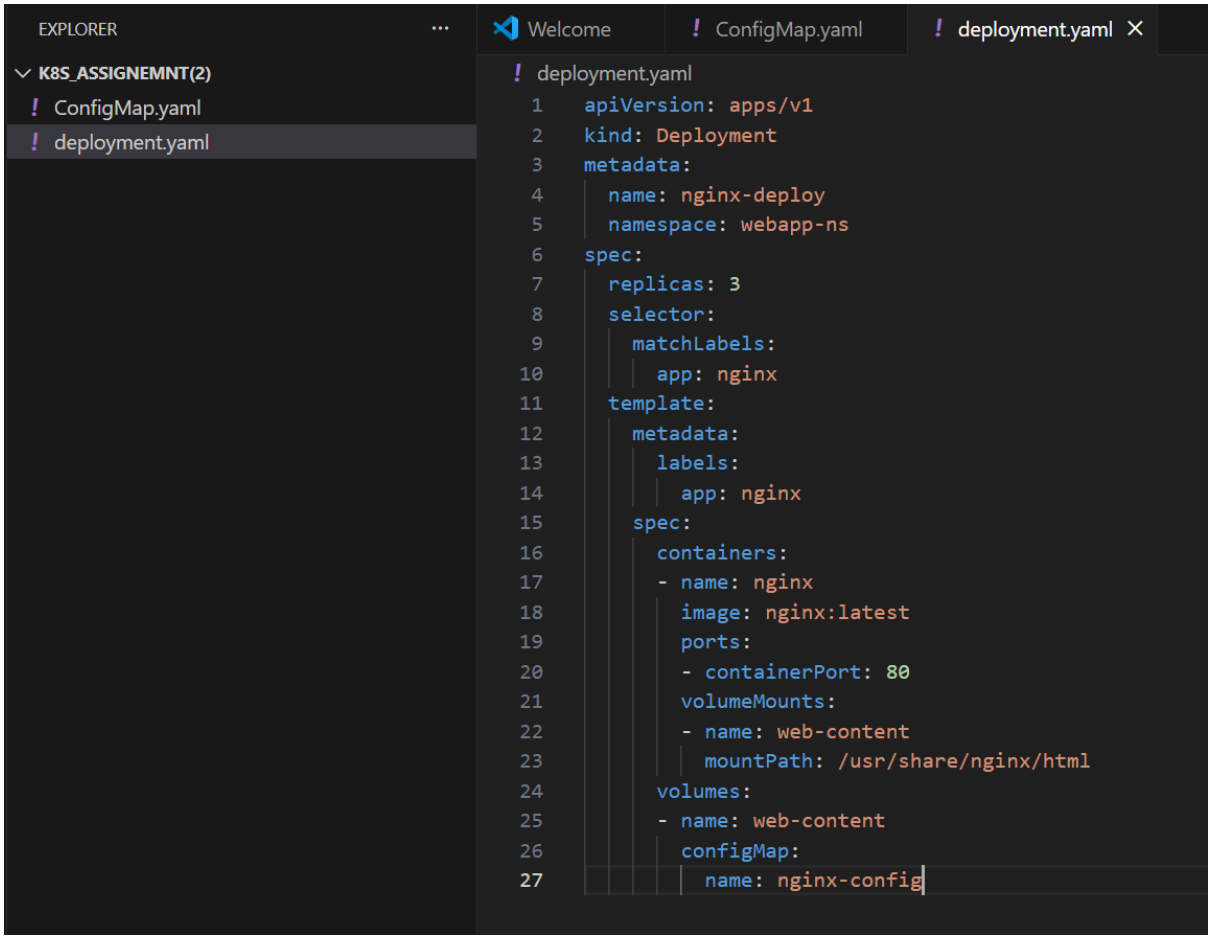
2. Creating ConfigMap




```
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl apply -f configmap.yaml
configmap/nginx-config created
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl get configmap -n webapp-ns
NAME               DATA   AGE
kube-root-ca.crt   1      49s
nginx-config       1      6s
```

3. Creating Deployment (Nginx + Replicas + Volume)

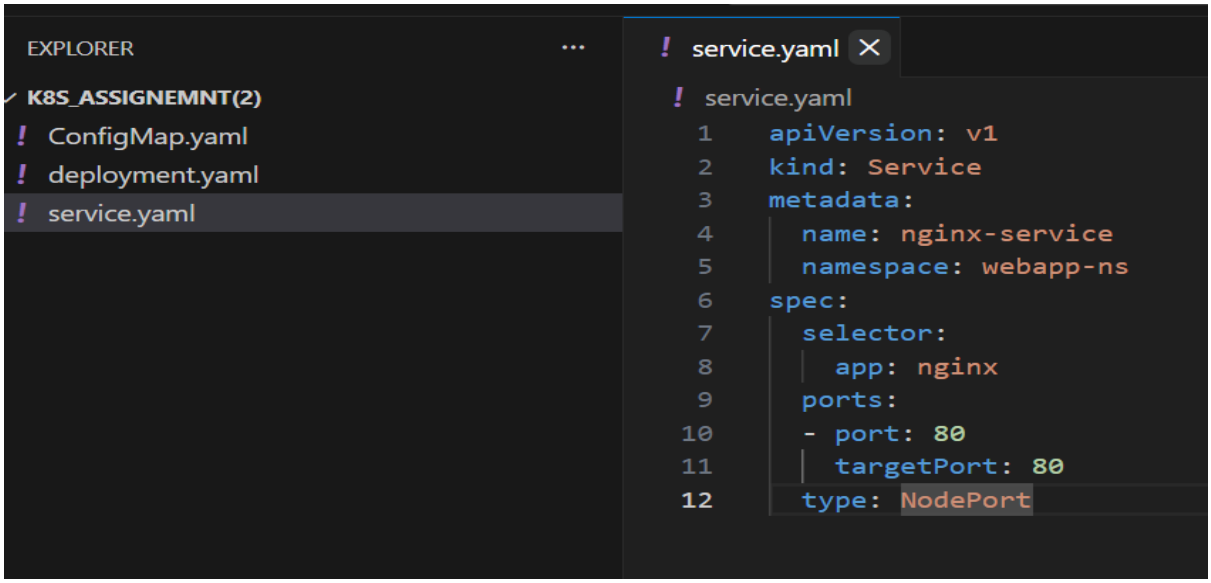
```
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl apply -f deployment.yaml
deployment.apps/nginx-deploy created
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl get deployments -n webapp-ns
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deploy   0/3     3            0           15s
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl get pods -n webapp-ns
NAME                            READY   STATUS              RESTARTS   AGE
nginx-deploy-58c8fcdf4b-92gfm   0/1     ContainerCreating   0          24s
nginx-deploy-58c8fcdf4b-dkz5v   1/1     Running             0          24s
nginx-deploy-58c8fcdf4b-hmvh4   1/1     Running             0          24s
```



4. Verifying Replica Management

```
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl describe deployment nginx-deploy -n webapp-ns
Name:                   nginx-deploy
Namespace:              webapp-ns
CreationTimestamp:      Tue, 17 Feb 2026 14:47:40 +0530
Labels:                 <none>
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:latest
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /usr/share/nginx/html from web-content (rw)
  Volumes:
   web-content:
    Type:          ConfigMap (a volume populated by a ConfigMap)
    Name:          nginx-config
    Optional:      false
  Node-Selectors:  <none>
  Tolerations:     <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-deploy-58c8fcdf4b (3/3 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  74s   deployment-controller  Scaled up replica set nginx-deploy-58c8fcdf4b from 0 to 3
```

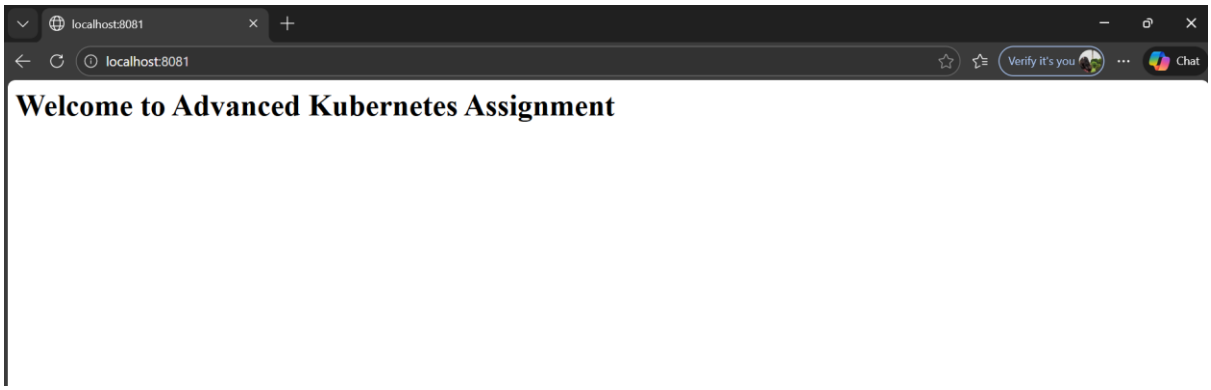
5. Creating Service (Expose Application)




```
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl apply -f service.yaml
service/nginx-service created
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl get service -n webapp-ns
NAME            TYPE       CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
nginx-service   NodePort   10.102.146.142   <none>        80:30436/TCP   6s
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl get nodes -o wide
NAME             STATUS   ROLES           AGE   VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE         KERNEL-VERSION                  CONTA
INER-RUNTIME
docker-desktop   Ready    control-plane   8d    v1.32.2   192.168.65.3   <none>        Docker Desktop   6.6.87.2-microsoft-standard-WSL2   docke
r://28.3.2
```

In local kubeadm clusters, NodePort IPs are internal to the VM, so I used kubectl port-forward to expose services on localhost.

```
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl port-forward svc/nginx-service 8081:80 -n webapp-ns
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
Handling connection for 8081
Handling connection for 8081
```



6. Scaling Deployment

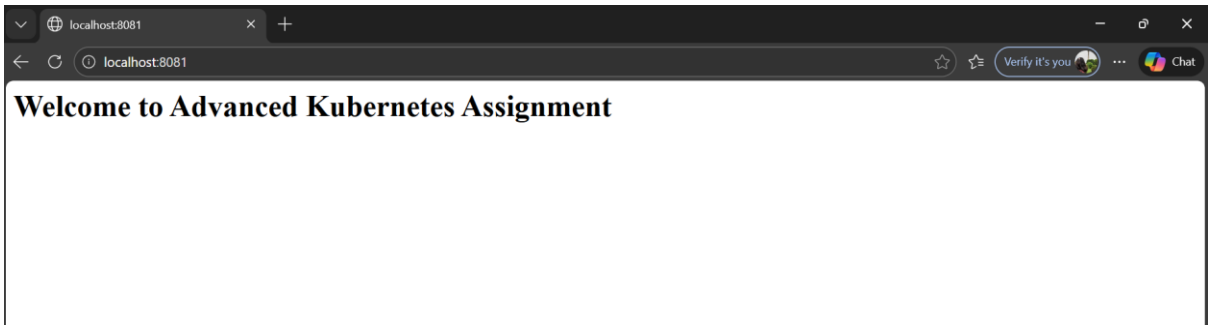
```
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl scale deployment nginx-deploy --replicas=5 -n webapp-ns
deployment.apps/nginx-deploy scaled
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl get pods -n webapp-ns
NAME                            READY   STATUS    RESTARTS   AGE
nginx-deploy-58c8fcdf4b-92gfm   1/1     Running   0          9m9s
nginx-deploy-58c8fcdf4b-dkz5v   1/1     Running   0          9m9s
nginx-deploy-58c8fcdf4b-hmvh4   1/1     Running   0          9m9s
nginx-deploy-58c8fcdf4b-n4t7v   1/1     Running   0          8s
nginx-deploy-58c8fcdf4b-wjcxz   1/1     Running   0          8s
```

7. Performing Rolling Update

```
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl set image deployment/nginx-deploy nginx=nginx:1.25 -n webapp-ns
deployment.apps/nginx-deploy image updated
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl rollout status deployment nginx-deploy -n webapp-ns
Waiting for deployment "nginx-deploy" rollout to finish: 3 out of 5 new replicas have been updated...
Waiting for deployment "nginx-deploy" rollout to finish: 3 out of 5 new replicas have been updated...
Waiting for deployment "nginx-deploy" rollout to finish: 3 out of 5 new replicas have been updated...
Waiting for deployment "nginx-deploy" rollout to finish: 3 out of 5 new replicas have been updated...
Waiting for deployment "nginx-deploy" rollout to finish: 4 out of 5 new replicas have been updated...
Waiting for deployment "nginx-deploy" rollout to finish: 4 out of 5 new replicas have been updated...
Waiting for deployment "nginx-deploy" rollout to finish: 4 out of 5 new replicas have been updated...
Waiting for deployment "nginx-deploy" rollout to finish: 4 out of 5 new replicas have been updated...
Waiting for deployment "nginx-deploy" rollout to finish: 2 old replicas are pending termination...
Waiting for deployment "nginx-deploy" rollout to finish: 2 old replicas are pending termination...
Waiting for deployment "nginx-deploy" rollout to finish: 2 old replicas are pending termination...
Waiting for deployment "nginx-deploy" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "nginx-deploy" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "nginx-deploy" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "nginx-deploy" rollout to finish: 4 of 5 updated replicas are available...
deployment "nginx-deploy" successfully rolled out
```

8. Validating Availability After Update

```
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl get pods -n webapp-ns
NAME                            READY   STATUS    RESTARTS   AGE
nginx-deploy-84ff5dd89c-jf4b4   1/1     Running   0          60s
nginx-deploy-84ff5dd89c-kjnpr   1/1     Running   0          87s
nginx-deploy-84ff5dd89c-mlrqm   1/1     Running   0          87s
nginx-deploy-84ff5dd89c-nzt6f   1/1     Running   0          87s
nginx-deploy-84ff5dd89c-p7wvq   1/1     Running   0          63s
```



The application can still be accessed after rolling update

9. Health Verification

```
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl logs nginx-deploy-84ff5dd89c-kjnpr -n webapp-ns
/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: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/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
2026/02/17 09:27:39 [notice] 1#1: using the "epoll" event method
2026/02/17 09:27:39 [notice] 1#1: nginx/1.25.5
2026/02/17 09:27:39 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2026/02/17 09:27:39 [notice] 1#1: OS: Linux 6.6.87.2-microsoft-standard-WSL2
2026/02/17 09:27:39 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2026/02/17 09:27:39 [notice] 1#1: start worker processes
2026/02/17 09:27:39 [notice] 1#1: start worker process 29
2026/02/17 09:27:39 [notice] 1#1: start worker process 30
2026/02/17 09:27:39 [notice] 1#1: start worker process 31
2026/02/17 09:27:39 [notice] 1#1: start worker process 32
2026/02/17 09:27:39 [notice] 1#1: start worker process 33
2026/02/17 09:27:39 [notice] 1#1: start worker process 34
2026/02/17 09:27:39 [notice] 1#1: start worker process 35
2026/02/17 09:27:39 [notice] 1#1: start worker process 36
2026/02/17 09:27:39 [notice] 1#1: start worker process 37
2026/02/17 09:27:39 [notice] 1#1: start worker process 38
2026/02/17 09:27:39 [notice] 1#1: start worker process 39
2026/02/17 09:27:39 [notice] 1#1: start worker process 40
2026/02/17 09:27:39 [notice] 1#1: start worker process 41
2026/02/17 09:27:39 [notice] 1#1: start worker process 42
2026/02/17 09:27:39 [notice] 1#1: start worker process 43
2026/02/17 09:27:39 [notice] 1#1: start worker process 44
```

```
PS C:\Users\Devanshi\Desktop\k8s_assignemnt(2)> kubectl describe pod nginx-deploy-84ff5dd89c-kjnpr -n webapp-ns
Name:             nginx-deploy-84ff5dd89c-kjnpr
Namespace:        webapp-ns
Priority:         0
Service Account:  default
Node:             docker-desktop/192.168.65.3
Start Time:       Tue, 17 Feb 2026 14:57:11 +0530
Labels:           app=nginx
                  pod-template-hash=84ff5dd89c
Annotations:      <none>
Status:           Running
IP:               10.1.0.25
IPs:
  IP:           10.1.0.25
Controlled By:  ReplicaSet/nginx-deploy-84ff5dd89c
Containers:
  nginx:
    Container ID:   docker://376e6c33123b90191ce8939296d7b46a265242154efdfe2cc3be924a4979f8fe
    Image:          nginx:1.25
    Image ID:       docker-pullable://nginx@sha256:a484819eb60211f5299034ac80f6a681b06f89e65866ce91f356ed7c72af059c
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Tue, 17 Feb 2026 14:57:39 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /usr/share/nginx/html from web-content (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-8qwvr (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  web-content:
    Type:      ConfigMap (a volume populated by a ConfigMap)
    Name:      nginx-config
    Optional:  false
  kube-api-access-8qwvr:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
```