PRACTICAL NO: - 4

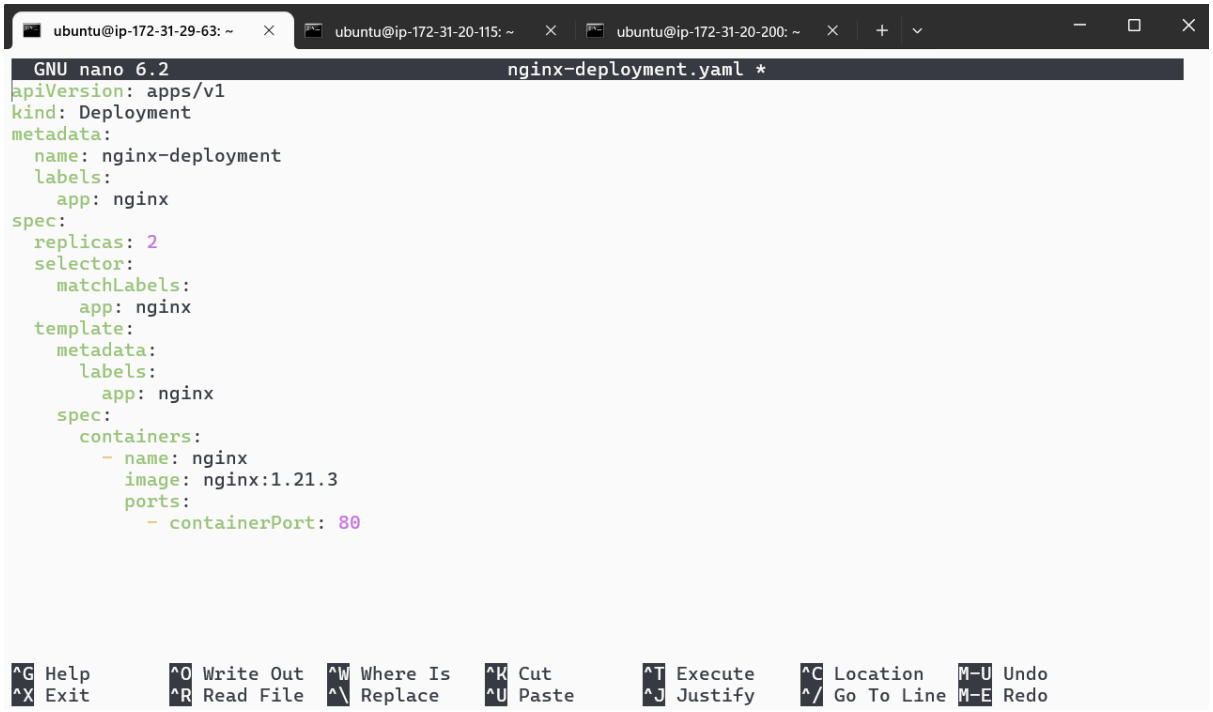
In previous practical, we have setup the Kubernetes cluster

➢ Deploying Your Application on Kubernetes

Create the YAML file: Use a text editor to create a file named nginx-deployment.yaml

```
ubuntu@ip-172-31-29-63:~$ nano nginx-deployment.yaml
ubuntu@ip-172-31-29-63:~$
```

Add the Deployment Configuration: Copy and paste the following YAML content into the file. Save and exit the editor (Press Ctrl+X, then Y, and Enter).



Create the YAML File: Create another file named nginx-service.yaml

```
ubuntu@ip-172-31-29-63:~$ nano nginx-service.yaml
ubuntu@ip-172-31-29-63:~$
```

Add the Service Configuration: Copy and paste the following YAML content into the file given below.

```
ubuntu@ip-172-31-29-63: ~   ubuntu@ip-172-31-20-115: ~   ubuntu@ip-172-31-20-200: ~
  GNU nano 6.2                                  nginx-service.yaml
apiVersion:v1
kind: Service
metadata:
   name:nginx-service
spec:
  selector:
    app: nginx
  ports:
       - protocol: TCP
         port: 80
         targetPort: 80
  type: LoadBalancer
```

Deploy the Application: Use kubectl to create the Deployment and Service from the YAML files.

```
ubuntu@ip-172-31-29-63:~$ kubectl apply -f nginx-deployment.yaml --validate=false
deployment.apps/nginx-deployment created

ubuntu@ip-172-31-29-63:~$ kubectl apply -f nginx-service.yaml --validate=false
service/nginx-service created
ubuntu@ip-172-31-29-63:~$
```

Verify the Deployment: Check the status of your Deployment,Pods and Services

```
ubuntu@ip-172-31-29-63:~$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   1/2     2            1           9m25s
ubuntu@ip-172-31-29-63:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS      AGE
nginx-deployment-6b4d6fdbf-6w4bm    1/1     Running   4 (98s ago)   9m18s
nginx-deployment-6b4d6fdbf-bhcwm    1/1     Running   4 (70s ago)   9m18s
ubuntu@ip-172-31-29-63:~$ kubectl get services
NAME            TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP      10.96.0.1       <none>        443/TCP        111m
nginx-service   LoadBalancer   10.110.88.111   <pending>     80:30132/TCP   110s
ubuntu@ip-172-31-29-63:~$
```

Describe the deployment(Extra)

```
ubuntu@ip-172-31-29-63:~$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           11m
ubuntu@ip-172-31-29-63:~$ kubectl describe deployment
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Sun, 15 Sep 2024 19:39:41 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx
Replicas:               2 desired | 2 updated | 2 total | 1 available | 1 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.21.3
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Progressing    True    NewReplicaSetAvailable
  Available      False   MinimumReplicasUnavailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-deployment-6b4d6fdbf (2/2 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  11m   deployment-controller  Scaled up replica set nginx-deployment-6b4d6fdbf to 2
ubuntu@ip-172-31-29-63:~$
```

Verify Service: Run the following command to check the services running in your cluster:

```
ubuntu@ip-172-31-29-63:~$ kubectl get service
NAME            TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP      10.96.0.1       <none>        443/TCP        114m
nginx-service   LoadBalancer   10.110.88.111   <pending>     80:30132/TCP   4m59s
ubuntu@ip-172-31-29-63:~$
```

Forward the Service Port to Your Local Machine

```
ubuntu@ip-172-31-45-227:~$ kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

```
ubuntu@ip-172-31-45-227:~$ kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
^Cubuntu@ip-172-31-45-227:~kubectl port-forward service/nginx-service 8081:8080
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
^Cubuntu@ip-172-31-45-227:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-776b8fd845-k9cx4   1/1     Running   0          113m
ubuntu@ip-172-31-45-227:~$ kubectl logs nginx-deployment-776b8fd845-k9cx4
/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
2024/09/12 06:35:51 [notice] 1#1: using the "epoll" event method
2024/09/12 06:35:51 [notice] 1#1: nginx/1.27.1
2024/09/12 06:35:51 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/09/12 06:35:51 [notice] 1#1: OS: Linux 6.5.0-1022-aws
2024/09/12 06:35:51 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/09/12 06:35:51 [notice] 1#1: start worker processes
2024/09/12 06:35:51 [notice] 1#1: start worker process 24
2024/09/12 06:35:51 [notice] 1#1: start worker process 25
```

Access the Application Locally

Open a Web Browser: Now open your web browser and go to the following URL: http://localhost:8080 You should see the application (in this case, Nginx) that you have deployed running in the Kubernetes cluster, served locally via port 8080.