FINAL LAB

1- create a namespace iti-devops

2- create a service account iti-sa-devops under the same namespace

```
hawila@hawila-pc: ~
hawila@hawila-pc:~$ kubectl create namespace iti-devops
namespace/iti-devops created
hawila@hawila-pc:~$ kubectl create serviceaccount iti-sa-devops  --namespace=iti-devops
serviceaccount/iti-sa-devops created
hawila@hawila-pc:~$
```

3- create a clusteRole which should be named as cluster-role-devops to grant permissions

```
Get Started    Rule.yaml
Lab5 > Rule.yaml > [ ] rules > {} 0 > [ ] resources > 8
all.json
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
 namespace: iti-devops
 name: cluster-role-devops
rules:
  - apiGroups: [""]
    resources: ["configMaps","secrets","endpoints","nodes","pods","services","namespaces","events","serviceAccounts"]
    verbs: ["get","list","watch","create","patch","update"]

hawila@hawila-pc: ~/test/Lab5
hawila@hawila-pc:~/test/Lab5$ kubectl apply -f Rule.yaml
role.rbac.authorization.k8s.io/cluster-role-devops created
hawila@hawila-pc:~/test/Lab5$
```

4- create a ClusterRoleBinding which should be named as cluster-role-binding-devops

```
ruleBind.yaml    Rule.yaml
Lab5 > ruleBind.yaml > [ ] subjects > {} 0 > apiGroup
all.json
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: cluster-role-binding-devops
  namespace: iti-devops
subjects:
- kind: ServiceAccount
  name: iti-sa-devops
  apiGroup: ""
roleRef:
  kind: ClusterRole
  name: cluster-role-devops
  apiGroup: rbac.authorization.k8s.io

hawila@hawila-pc: ~/test/Lab5
hawila@hawila-pc:~/test/Lab5$ kubectl apply -f ruleBind.yaml
rolebinding.rbac.authorization.k8s.io/cluster-role-binding-devops created
hawila@hawila-pc:~/test/Lab5$
```

5- What is the difference between statefulSets and deployments?

ASPECT	DEPLOYMENT	STATEFULSET
Data persistence	Stateless	Stateful
Pod name and identity	Pods are assigned an ID that consists of the deployment name and a random hash to generate a temporarily unique identity	Each pod gets a persistent identity consisting of the StatefulSet name and a sequence number
Interchangeability	Pods are identical and can be interchanged	Pods in a StatefulSet are neither identical nor interchangeable
Behavior	A pod can be replaced by a new replica at any time	Pods retain their identity when rescheduled on another node
Volume claim	All replicas share a PVC and a volume	Each pod gets a unique volume and PVC
Allowed volume access mode(s)	ReadWriteMany and ReadOnlyMany	ReadWriteOnce
Pod interaction	Requires a service to interact with the pods	The headless service handles pod network identities

6- Set up Ingress on Minikube with the NGINX Ingress Controller

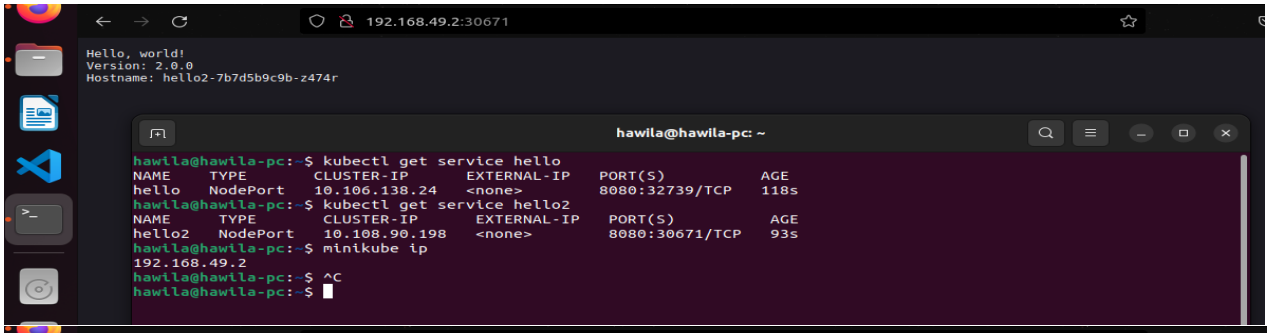
Enable ingress

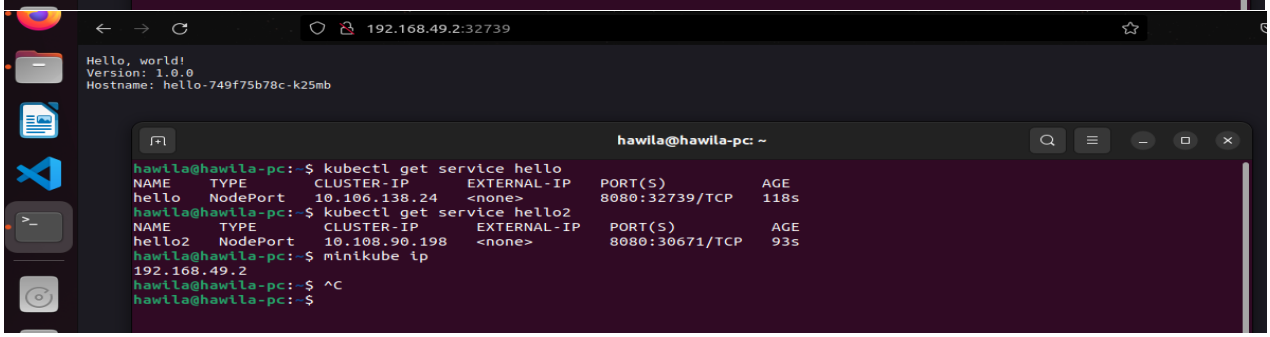
```
hawila@hawila-pc:~$ minikube addons enable ingress
💡  ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
    ▪ Using image registry.k8s.io/ingress-nginx/controller:v1.5.1
    ▪ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20220916-gd32f8c343
    ▪ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20220916-gd32f8c343
🔎  Verifying ingress addon...
🌟  The 'ingress' addon is enabled
hawila@hawila-pc:~$ kubectl get pods -n ingress-nginx
NAME                                        READY   STATUS      RESTARTS   AGE
ingress-nginx-admission-create-qszds        0/1     Completed   0          2m37s
ingress-nginx-admission-patch-mh4p9         0/1     Completed   1          2m37s
ingress-nginx-controller-77669ff58-dmhtv    1/1     Running     0          2m37s
hawila@hawila-pc:~$
```

Create 2 deployment

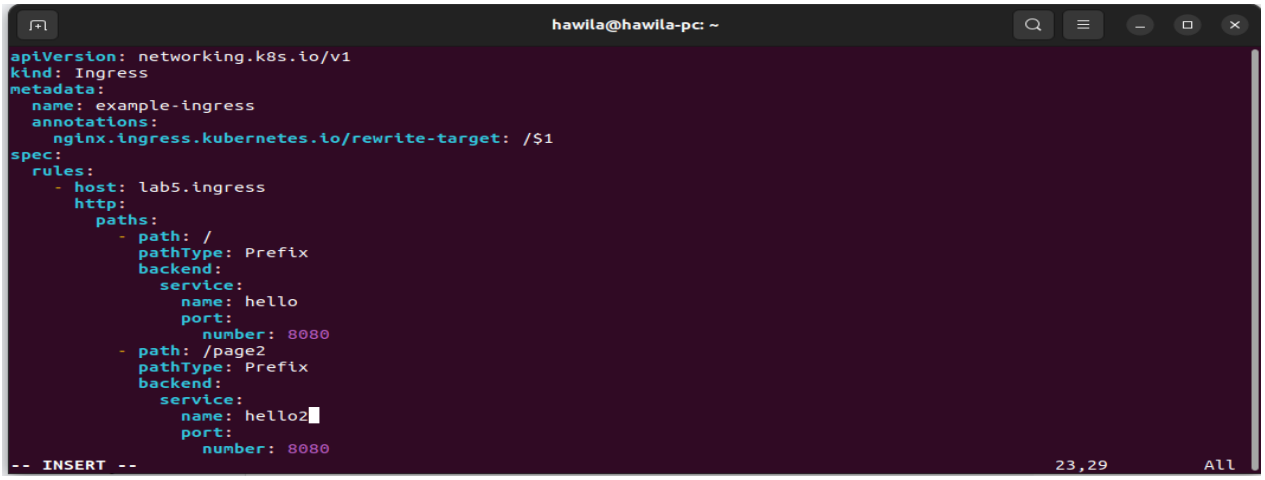
```
hawila@hawila-pc: ~
hawila@hawila-pc:~$ kubectl create deployment hello --image=gcr.io/google-samples/hello-app:1.0
deployment.apps/hello created
hawila@hawila-pc:~$ kubectl expose deployment hello --type=NodePort --port=8080
service/hello exposed
hawila@hawila-pc:~$ kubectl create deployment hello2 --image=gcr.io/google-samples/hello-app:2.0
deployment.apps/hello2 created
hawila@hawila-pc:~$ kubectl expose deployment hello2 --type=NodePort --port=8080
service/hello2 exposed
hawila@hawila-pc:~$
```

check service

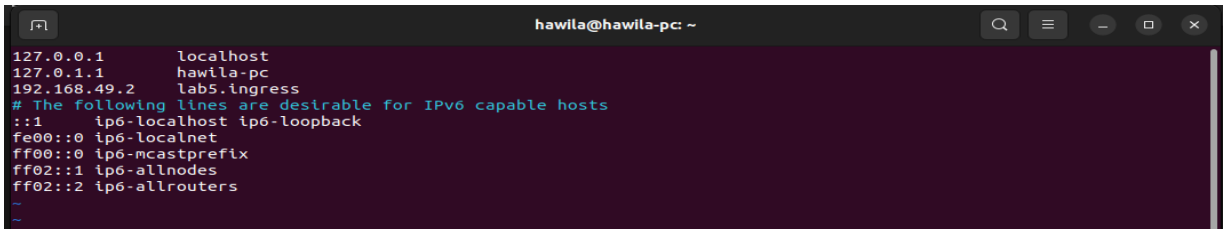




ingress-config



Host configuration



Test url