LAB 05

1- create a namespace iti-devops

```
[mohamed@Azzam lab-05]$ kubectl create namespace iti-devops
namespace/iti-devops created
[mohamed@Azzam lab-05]$
```

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

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: iti-sa-devops
  namespace: iti-devops
~
```

```
[mohamed@Azzam lab-05]$ kubectl apply -f service-acc.yaml
serviceaccount/iti-sa-devops created
[mohamed@Azzam lab-05]$
```

LAB 05

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

“get”,”list”,”watch”,”create”,”patch”,”update” to

“configMaps”,”secrets”,”endpoints”,”nodes”,”pods”,”services”,”namespaces”,”events”,”serviceAccou

nts”.

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: cluster-role-devops
rules:
- apiGroups: [""]
  resources: ["services", "endpoints", "pods", "configMaps", "secrets", "nodes", "namespaces", "events", "serviceAccounts"]
  verbs: ["get", "list", "watch", "create", "patch", "update"]
~
~
```

```
[mohamed@Azzam lab-05]$ vim cluster-role.yaml
[mohamed@Azzam lab-05]$ kubectl apply -f cluster-role.yaml
clusterrole.rbac.authorization.k8s.io/cluster-role-devops created
[mohamed@Azzam lab-05]$
```

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

namespace. Define roleRef apiGroup should be rbac.authorization.k8s.io . Kind should be ClusterRole,

name should be cluster-role-devops and subjects kind should be ServiceAccount: name should be iti-sa-

devops and namespace should be iti-devops

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: cluster-role-binding-devops
subjects:
- kind: ServiceAccount
  name: iti-sa-devops
  namespace: iti-devops
roleRef:
  kind: ClusterRole
  name: cluster-role-devops
  apiGroup: rbac.authorization.k8s.io
~
```

LAB 05

5- What is the difference between statefulSets and deployments?

StatefulSet	Deployment
Used to deploy stateful applications	Used to deploy stateless applications
Pods created by StatefulSets have unique names which remain constant across application rescheduling.	Pods created by Deployment have dynamic, random names and numbers that change across application rescheduling.
Its Pods are created in sequential order and deleted in reverse, sequential order.	Its Pods are created and deleted randomly.
Its Pods are not interchangeable and maintain their identities after restarts.	Its Pods are interchangeable and do not maintain their identities after restarts.
It does not allow shared volume. Thus, each Pod replica has its own sticky Volume and PersistentVolumeClaim.	It allows shared volume via Volume and PersistentVolumeClaim across all of the Pod replicas.
Replication is complex	Replication is easier

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

play around with paths , you can create more than 2 deployments if you like

https://kubernetes.io/docs/tasks/access-application-cluster/ingress-minikube/

```
File  Edit  View  Search  Terminal  Help
[mohamed@Azzam lab-05]$ vim ingress.yaml
[mohamed@Azzam lab-05]$ kubectl apply -f ingress.yaml
ingress.networking.k8s.io/example-ingress configured
[mohamed@Azzam lab-05]$
```

LAB 05

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$1
spec:
  rules:
    - host: hello-world.info
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: web
                port:
                  number: 8080
          - path: /v2
            pathType: Prefix
            backend:
              service:
                name: web2
                port:
                  number: 8080
```