1- create a namespace iti-devops

```
apiVersion: v1
kind: Namespace
metadata:
  name: iti-devops
  labels:
    name: iti-devops
```

```
controlplane $ k create -f name-s
namespace/iti-devops created
controlplane $
```

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

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

```
controlplane $ k create -f sa
serviceaccount/iti-sa-devops created
```

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","service Accou nts".

```
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"]
```

```
controlplane $ k create -f cr
role.rbac.authorization.k8s.io/cluster-role-devops created
```

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-sadevops and namespace should be iti-devops

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

```
controlplane $ k create -f crb
rolebinding.rbac.authorization.k8s.io/cluster-role-binding-devops created
```

5- What is the difference between statefulSets and deployments?
A StatefulSet is a Kubernetes resource object that manages a set of pods with unique identities. By assigning a persistent ID that is maintained even if the pod is rescheduled, a

StatefulSet *helps maintain the uniqueness and ordering of pods. With unique pod identifiers, administrators can efficiently attach cluster volumes to new pods across failures. Although the StatefulSet controller deploys pods using similar specifications, pods are not interchangeable. As a StatefulSet does not create a ReplicaSet, the pod replicas cannot be rolled back to previous versions. StatefulSets are typically used for applications that require persistent storage for stateful workloads, and ordered, automated rolling updates.*

A Deployment *is a Kubernetes resource object that provides declarative updates for pods that encapsulate application containers. A Deployment represents a number of identical pods without unique IDs, while specifying the pods' desired state and attributes. Deployments are typically used to autoscale the number of pod replicas, perform controlled rollouts for application code, and perform rollbacks when necessary.*

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

```
apiVersion: networking.k8s.io/\1
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:
              ser\ice:
                name: web
                port:
                  number: 8080
```

```
]$ minikube addons enable ingress
```