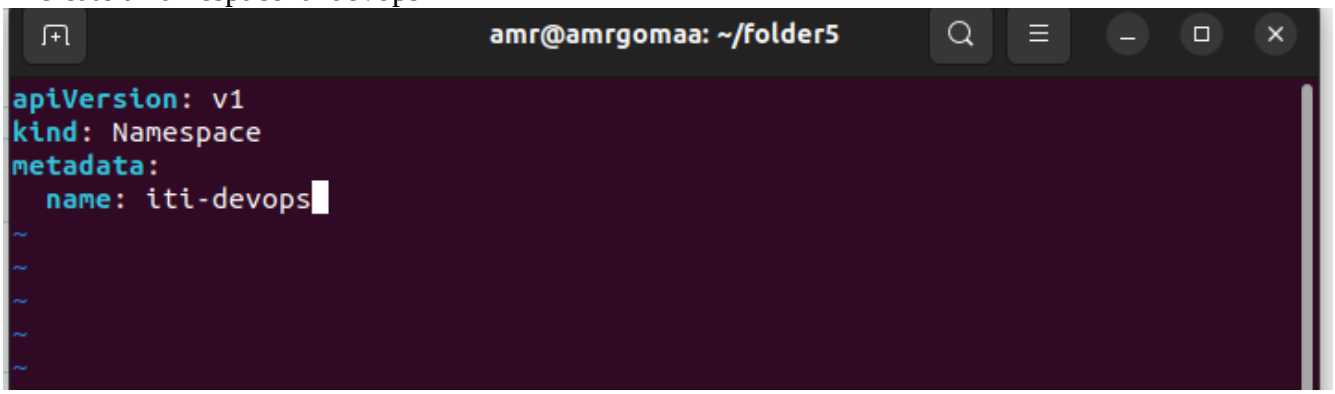
FINAL LAB

1- create a namespace iti-devops




```
mr@amrgomaa:~/folder5$ kubectl apply -f ns.yml
amespace/iti-devops created
mr@amrgomaa:~/folder5$
```

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

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

```
amr@amrgomaa:~/folder5$ kubectl apply -f svc.yml
serviceaccount/iti-sa-devops created
amr@amrgomaa:~/folder5$
```

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”,”serviceAccounts”.

```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"]
~
```

```
amr@amrgomaa:~/folder5$ kubectl apply -f clusterrole.yml
clusterrole.rbac.authorization.k8s.io/cluster-role-devops created
amr@amrgomaa:~/folder5$
```

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
~
```

```
amr@amrgomaa:~/folder5$ kubectl apply -f rolebinding.yml
clusterrolebinding.rbac.authorization.k8s.io/cluster-role-binding-devops created
amr@amrgomaa:~/folder5$
```

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
Order of pod creation	Pods are created and deleted randomly	Pods are created in a strict sequence and cannot be deleted randomly

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/
Best of Luck

```
amr@amrgomaa:~/folder5$ curl hello-world.info
Hello, world!
Version: 1.0.0
Hostname: web-84fb9498c7-5wmw4
amr@amrgomaa:~/folder5$ curl hello-world.info/v2
Hello, world!
Version: 2.0.0
Hostname: web2-7df4dcf77b-4g8qm
amr@amrgomaa:~/folder5$
```

```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
~
```