**Scenario 1**

\*\*Task 1 \*\* Establish a new ClusterRole for a deployment process and associate it with a distinct ServiceAccount within a designated namespace.

\*\*Instructions:\*\*

1. Construct a fresh ClusterRole titled `deploy-role-cluster` that grants permission exclusively to generate the subsequent resource kinds:

- Deployment

- StatefulSet

- DaemonSet

2. create a namespace by name dev-group1

3. Formulate a new ServiceAccount by the name `pipeline-auth` inside the pre-existing namespace `dev-group1`.

4. Link the recently created ClusterRole `deploy-role-cluster` to the ServiceAccount `pipeline-auth`, restricting its scope only to the `dev-group1` namespace.

**Solution**

\*\*Instructions:\*\*

1. Construct a fresh ClusterRole titled `deploy-role-cluster` that grants permission exclusively to generate the subsequent resource kinds:

- Deployment

- StatefulSet

- DaemonSet

kubectl create clusterrole deploy-role-cluster --verb=create --resource=deployments,statefulsets,daemonsets -n apps

2. create namespace if not exist

kubectl create ns dev-group1

3. Formulate a new ServiceAccount by the name `pipeline-auth` inside the pre-existing namespace `dev-group1`.

kubectl create sa pipeline-auth -n dev-group1

3. Link the recently created ClusterRole `deploy-role-cluster` to the ServiceAccount `pipeline-auth`, restricting its scope only to the `dev-group1` namespace.

kubectl create rolebinding deploy-role-cluster-binding --clusterrole=deploy-role-cluster --serviceaccount=dev-group1:pipeline-auth -n dev-group1