📊 Kubernetes RBAC Lab: ServiceAccounts & Roles

🎯 Objectives

- Create a namespace for isolation.
- Create a ServiceAccount.
- Create a Role with read-only pod permissions.
- Bind the Role to the ServiceAccount.
- Launch a pod using that ServiceAccount.
- Test permissions from inside the pod.

🔷 Step 1: Create Namespace

```
kubectl create namespace rbac-lab
```

🔷 Step 2: Create ServiceAccount

```
kubectl create serviceaccount viewer-sa -n rbac-lab
```

Verify:

```
kubectl get serviceaccounts -n rbac-lab
```

🔷 Step 3: Create Role

```yaml
# rbac-role.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: pod-reader
  namespace: rbac-lab
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get", "list", "watch"]
```

Apply:

```
kubectl apply -f rbac-role.yaml
```

🔷 Step 4: Create RoleBinding

```yaml
# rbac-rolebinding.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: read-pods-binding
  namespace: rbac-lab
subjects:
- kind: ServiceAccount
  name: viewer-sa
  namespace: rbac-lab
roleRef:
  kind: Role
  name: pod-reader
  apiGroup: rbac.authorization.k8s.io
```

Apply:

```
kubectl apply -f rbac-rolebinding.yaml
```

🔷 Step 5: Deploy Test Pod

```yaml
# busybox-test.yaml
apiVersion: v1
kind: Pod
metadata:
  name: curl-sa-test
  namespace: rbac-lab
spec:
  serviceAccountName: pod-reader
  containers:
  - name: curl
    image: curlimages/curl:8.7.1
    command: ["sleep"]
    args: ["3600"]
```

Apply:

```
kubectl apply -f busybox-test.yaml
```

🔷 Step 6: Test Access from Pod

```
kubectl exec -n rbac-lab -it curl-sa-test -- sh
```

Inside the pod:

```
# Set variables
TOKEN=$(cat /var/run/secrets/kubernetes.io/serviceaccount/token)
CACERT=/var/run/secrets/kubernetes.io/serviceaccount/ca.crt
NAMESPACE=rbac-lab

# ✅ Should work: GET pods in namespace
curl -s --cacert $CACERT --header "Authorization: Bearer $TOKEN" \
  https://kubernetes.default.svc/api/v1/namespaces/$NAMESPACE/pods

# ❌ Should fail: POST (create) a pod (no permission)
curl -s --cacert $CACERT --header "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -X POST \
  -d '{}' \
  https://kubernetes.default.svc/api/v1/namespaces/$NAMESPACE/pods
```

✅ Clean Up

```
kubectl delete namespace rbac-lab
```

🔧 Optional Enhancements

- Use ClusterRole instead of Role for cross-namespace access.
- Mount SA token manually and use `curl` with Bearer token.
- Validate with `kubectl auth can-i`.

Let me know if you want a GitHub repo scaffold or auto-marking script for this lab.