

RBAC and Multi-User Access

Lab 7: RBAC and Multi-User Access

Objective:

- Understand and apply Kubernetes Role-Based Access Control (RBAC) using Roles, RoleBindings, and ServiceAccounts.

Steps

1. Create a Namespace

```
kubectl create namespace dev
```

2. Create a ServiceAccount

```
kubectl create serviceaccount dev-user --namespace dev
```

3. Create YAML file: `role.yaml` to define a Role with Limited Permissions

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

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

4. Create YAML file: `rolebinding.yaml` to bind the Role to the ServiceAccount

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: read-pods-binding
  namespace: dev
subjects:
- kind: ServiceAccount
  name: dev-user
  namespace: dev
roleRef:
  kind: Role
  name: pod-reader
  apiGroup: rbac.authorization.k8s.io
```

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

5. Create YAML file: `rbac-test.yaml` to test Pod That Uses the ServiceAccount

```
apiVersion: v1
kind: Pod
metadata:
  name: rbac-test
  namespace: dev
spec:
  serviceAccountName: dev-user
  containers:
  - name: busybox
    image: busybox
    command: ["sh", "-c", "sleep 3600"]
```

Apply it:

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

6. Test Permissions

Open a shell inside the pod:

```
kubectl exec -it rbac-test -n dev -- sh
```

✓ Allowed action:

```
kubectl get pods -n dev
```

✗ Forbidden action:

```
kubectl get secrets -n dev
```

7. Clean Up

```
kubectl delete pod rbac-test -n dev  
kubectl delete role pod-reader -n dev  
kubectl delete rolebinding read-pods-binding -n dev  
kubectl delete sa dev-user -n dev  
kubectl delete ns dev
```