Managing Role Based Access Controls

Kien Bui

DevOps & Platform Engineer

Course Overview

Kubernetes Security Fundamentals

Managing Certificates and kubeconfig Files

Managing Role Based Access Controls

Summary

What is Role Based Access Control (RBAC)

API Objects for configuring RBAC

- Role and ClusterRole
- RoleBinding and ClusterRoleBinding

Role Based Access Control (RBAC)

Authorization plugin enabled on the API Server

Allowing a requestor to perform actions on resources

RESTful API semantics

- Verb on Noun

Default deny, rules are written to permit actions on the resource

Subjects - users, groups or `ServiceAccounts`

API Objects for Implementing RBAC Rules

Role

ClusterRole

RoleBinding

ClusterRoleBinding

Roles

`Roles` are what can be done to Resources

`Roles` are made up of one or many `Rules`

Verbs on resources

- Get Pods, Create Deployment

Default deny, add permissions to Resources

There is no deny permission

Roles are namespaced

ClusterRoles

Similar to a `Role`, enables access to Resources

Cluster scoped resources

- `Nodes`, `PersistentVolumes`

Give access across more than one namespace or all namespaces

Defining `Roles` in each namespace can increase administrative overhead and can be error prone

RoleBinding

`Role/ClusterRole` only say what can be done

Defines the Subjects and refers to a `Role/ClusterRole`

Who can do what defined in a `Role/ClusterRole`

`Role` and `RoleBinding` are used in namespaced scoped security

`ClusterRole` and `RoleBinding` are used provide access to more than one namespace or the whole cluster

ClusterRoleBinding

`ClusterRoleBinding` grants access cluster-wide

Combing a `ClusterRole` with a `ClusterRoleBinding`

Will scope security independent of namespace

- Non-namespaced
- Cluster-scoped resources

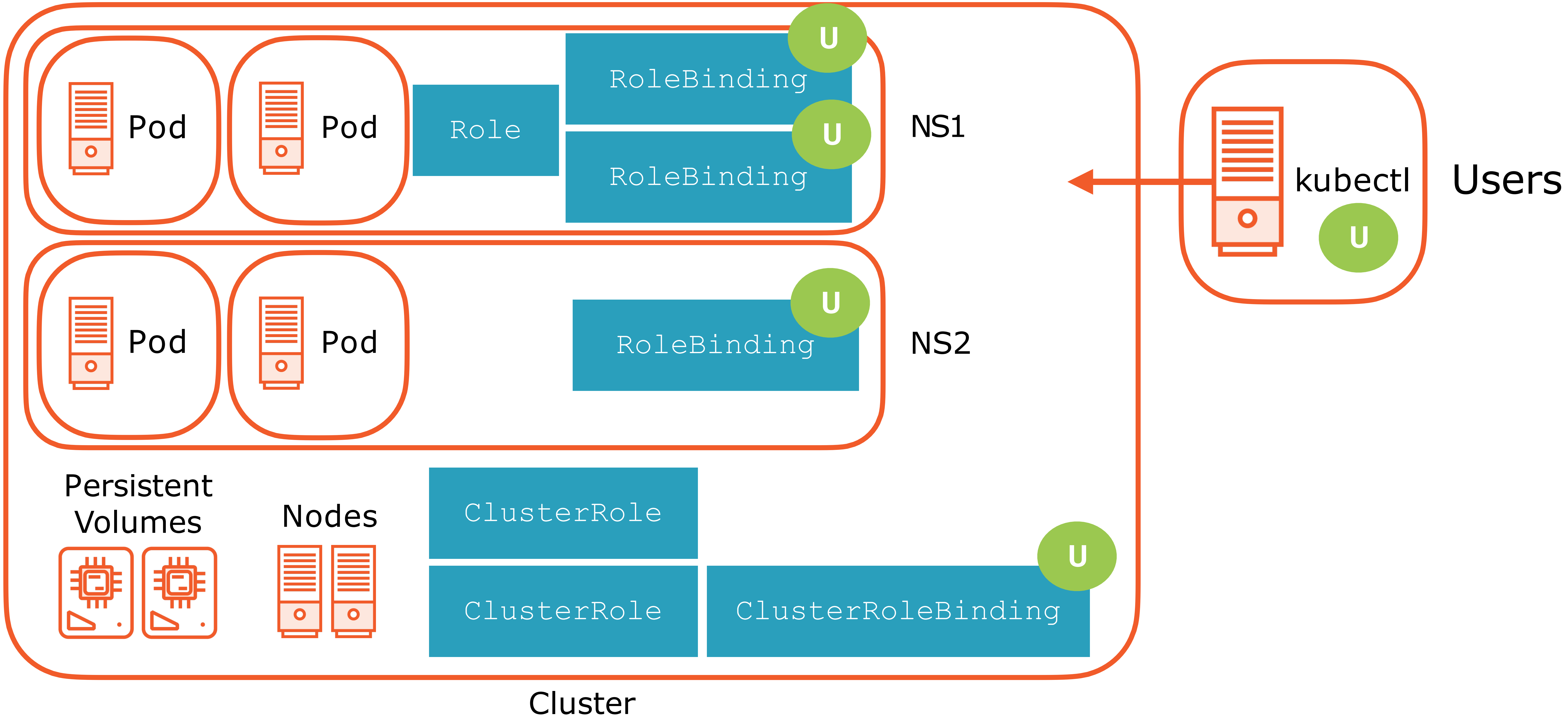
What to use when?

✓ Use `Role` and a `RoleBinding` to scope security to a single namespace

✓ Use `ClusterRole` and `RoleBinding` to scope security to several or all namespaces

✓ Use `ClusterRole` and `ClusterRoleBinding` to scope security to all namespaces OR cluster-scoped resources

Using RBAC


Pod
Pod
Role
RoleBinding
RoleBinding
U
U
NS1
Pod
Pod
RoleBinding
U
NS2
Persistent Volumes
Nodes
ClusterRole
ClusterRole
ClusterRoleBinding
U
Cluster
kubectl
U
Users

Default ClusterRoles

cluster-admin	admin	edit	view
Cluster-wide super user	Full access within a Namespace	Read/write within a Namespace	Read-only within a Namespace
RoleBinding - full admin within a Namespace	RoleBinding - full admin within a Namespace	NOT view/edit Roles RoleBindings Resource Quotas	NOT view/edit Roles RoleBindings Resource Quotas
Edit Roles RoleBindings Resource Quotas	Edit Roles RoleBindings	Access to Secrets	No Access to Secrets

Defining Roles and ClusterRoles

- `Rules`
  - `apiGroups`
    - An empty string designates the Core API group
  - Resources
    - `Pods`, `Services`, `Deployments`, `Nodes` and more
  - `Verbs`
    - `get, list, create, update, patch, watch, delete, deletecollection`
- `Roles`/`ClusterRoles` can have several `Rules` defined

https://bit.ly/314xJ24

Defining RoleBindings and ClusterRoleBindings

roleRef

- RoleBinding -> Role/ClusterRole
- ClusterRoleBinding -> ClusterRole

Subjects

- kind (User/Group/ServiceAccount)
- Name
- Namespace

Role and RoleBinding

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
 name: demorole
 namespace: ns1
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get", "list"]
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: demorolebinding
  namespace: ns1
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: demorole
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: User
  name: demouser
```

Role and RoleBinding

```
kubectl create role demorole \
  --verb=get,list \
  --resource=pods
  --namespace ns1

kubectl create rolebinding demorolebinding \
  --role=demorole \
  --user=demouser \
  --namespace ns1
```

Demo

Role-based Access Controls -RBAC

- Roles and RoleBindings
- ClusterRoles and ClusterRoleBindings
- ClusterRoles and RoleBindings

Review

What is Role Based Access Control (RBAC)

API Objects for configuring RBAC

- Role and ClusterRole
- RoleBinding and ClusterRoleBinding