#Kubernetes In 30Days challenge :-

#Day12 :-

Today It's about RBAC in kubernetes.

Firstly RBAC means Role Based Access Control As the name suggest it is used to give the appropriate access according to the roles allocated people. similar to RBAC in azure cloud and AWS cloud.

For Example If you take a Teaching Institute There will be lot of people who are ideally do different work according to Roles they have

Principal - Manages Everything in college

subject1 Faculty - teaches subject1

subject2 Faculty - teaches subject2

Exactly, Kubernetes will provide provision to enhance security using the features of RBAC. This will eliminate unauthorized usage of Kubernetes services. If the RBAC isn't there in Kubernetes, anyone can delete/Modify/Update the cluster. This shouldn't be done in Prod.

RBAC Broadly divided as follows:

In Realtime Developer, QA, DevOps, Kubernetes Admin should have different Roles there by different levels of Access to cluster.

Users :- This is Just like IAM users in AWS or AAD Users in Azure. This used to get users access to cluster → User Management

→ We cannot create users directly on Kubernetes. We can use something called Identity Provider which will authenticate user to access cluster.

These Identity Provider may be AWS IAM, Azure AD, LDAP, SSO etc. There is OAuth to authenticate users (needs to integrate with Kubernetes) to access cluster.

So on High Note, Kubernetes won't do any kind of user management. It's the Identity Provider who has to do.

service accounts :- These are just like kubernetes objects (pod, Deploy, svc etc). It's an yaml file to be ran on cluster to create service A/c.

service Account is needed because think of Pod should be able to access various things such as secrets / configMap, it is possible through sa.

To Manage RBAC we have

① service Accounts / Users

② Role / cluster Role

③ Role Binding / cluster Binding

Note :- There's a default service Account that is associated with each Pod to establish communication b/w control plane & Pod.

① A Role will be created with details using Role.yml

② Attaching the Role to service Account or user is called Role Binding. using a file

③ If it is constrained to Namespace then we will call it as Role and Role Binding.

④ If it is at cluster level access then it is cluster Role and cluster Role Binding.

⑤ There will be default Roles present in Kubernetes. We can make use of them if it matches same access levels.

Demo:-

Role creation and Binding:-

① create a test namespace

```
# Kubectl create ns test
```

② create a service Account using below file:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: myaccount
  namespace: test
```

③ create the service Account with name myaccount

```
# Kubectl apply -f sa.yml
```

④ Now create Role which has access Rules.

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: test
  name: testadmin
rules:
- apiGroups: ["*"]
  resources: ["*"]
  verbs: ["*"]
```

→ This Role having access to Everything in the cluster. (yet Not attached)

⑤ Now create the file for Role Binding using below text.

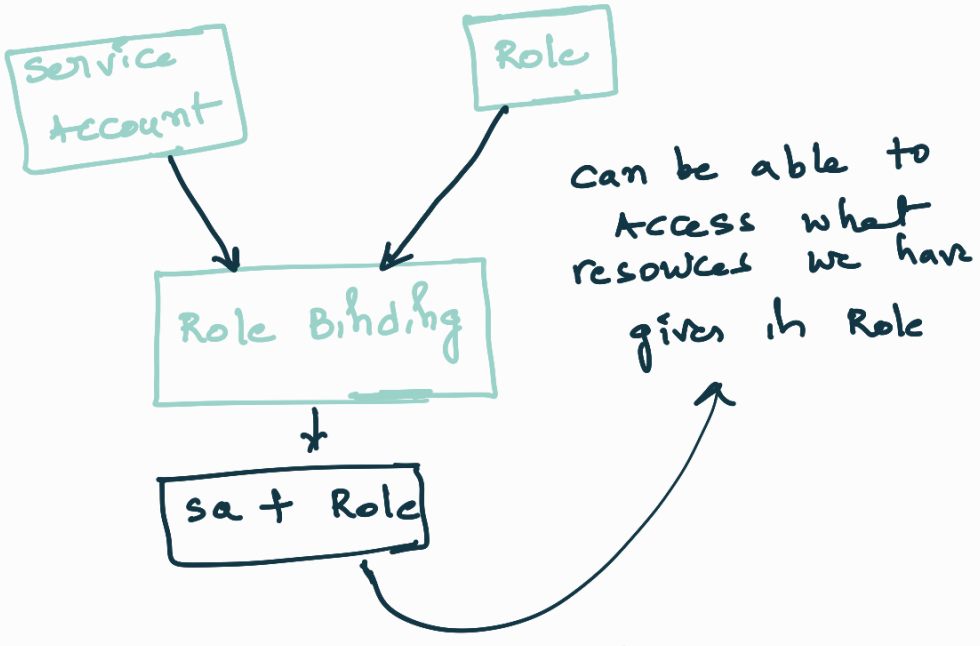
```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: testadminbinding
  namespace: test
subjects:
- kind: ServiceAccount
  name: myaccount
  apiGroup: ""
roleRef:
  kind: Role
  name: testadmin
  apiGroup: ""
```

Namespace

service Account created

Role we created

This how we can create sa, Roles and Role Bindings.



Authentication Vs **Authorization**

- It Authenticates user against creds whether correct user logged in or not

- It Authorizes the user have a specific user or not against the policies/Roles