**Lab Exercise 22- Authentication and Authorization in Kubernetes**

Service Accounts in Kubernetes allow you to authenticate and authorize applications and services running within a cluster. They provide a way to grant specific permissions and access control to pods and containers.

In this practical, we will cover the following steps:

1. Creating a Service Account



1. Creating a token for the Service Account



1. Creating a Role to define permissions



1. Creating a RoleBinding to associate the Role with the Service Account



1. Using the Service Account in a Pod



1. Verifying access permissions



**Setting Up Your Service Account**

To create a Service Account, use the following commands:

kubectl create sa mynewsa

To create a token for the Service Account "mysa" :

kubectl create token mynewsa

**Defining Permissions with Roles**

To define permissions for the Service Account, we need to create a Role **role.yaml** . Use the following YAML file:

**role.yaml**

apiVersion: rbac.authorization.k8s.io/v1

kind: Role

metadata:

namespace: default

name: pod-reader

rules:

- apiGroups:

- ''

resources:

- pods

verbs:

- get

- watch

- list

Run

kubectl apply -f role.yaml

To associate the Role with the Service Account, create a RoleBinding **rolebinding.yaml** . Use the following YAML file:

**rolebinding.yaml**

apiVersion: rbac.authorization.k8s.io/v1

kind: RoleBinding

metadata:

name: read-pods

namespace: default

subjects:

- kind: ServiceAccount

name: mynewsa

namespace: default

roleRef:

kind: Role

name: pod-reader

apiGroup: rbac.authorization.k8s.io

**Run**

kubectl apply -f rolebinding.yaml

**Putting It All Together: Using Service Accounts in Pods**

To use the Service Account in a Pod, update the Pod **pod.yaml** definition with the appropriate serviceAccountName. Use the following YAML file:

**pod.yaml**

apiVersion: v1

kind: Pod

metadata:

name: nginx

spec:

serviceAccountName: mynewsa

containers:

- name: nginx

image: nginx:1.14.2

ports:

- containerPort: 80

**Run**

kubectl apply -f pod.yaml

**Ensuring Access: Verifying Permissions**

To verify the access permissions of the Service Account, use the following command:

kubectl auth can-i get pods --as=system:serviceaccount:default:mynewsa

**Explanation:**

* The command checks if the Service Account "mysa" has permission to get pods.
* The output will indicate whether the access is allowed or denied.

**Conclusion**

Congratulations! You have successfully created and configured a Service Account in Kubernetes. You learned how to create a Service Account, associate it with a Role, and use it in a Pod. You also verified the access permissions of the Service Account. Feel free to explore further and customize the roles and permissions based on your specific requirements.