# Role-Based Access Control (RBAC) and Service Accounts in Kubernetes

Role-Based Access Control (RBAC) and Service Accounts are fundamental features in Kubernetes that enhance security by managing who can access what resources within the cluster.

## Role-Based Access Control (RBAC)

RBAC is a method of regulating access to computer or network resources based on the roles of individual users within an enterprise. In Kubernetes, RBAC allows you to control who can access the Kubernetes API and what permissions they have.

### Key Concepts

- Role: Contains rules that represent a set of permissions. It is namespace-scoped.

- ClusterRole: A role that applies to the entire cluster.

- RoleBinding: Grants the permissions defined in a role to a user or set of users. It is namespace-scoped.

- ClusterRoleBinding: Grants the permissions defined in a ClusterRole to a user or set of users across the entire cluster.

## Service Accounts

Service accounts are accounts used by processes within pods to interact with the Kubernetes API. Each service account is tied to a set of credentials stored as Secrets, which are mounted into pods allowing in-cluster processes to communicate with the Kubernetes API.

Example of creating a service account:

apiVersion: v1  
kind: ServiceAccount  
metadata:  
 name: myserviceaccount

## Use Cases

- RBAC: Managing user permissions across a Kubernetes environment, ensuring users have only the access they need.

- Service Accounts: Automatically managing API credentials for Kubernetes-native applications, typically those that run within pods and need to interact with other parts of Kubernetes.