<https://thenewstack.io/three-realistic-approaches-to-kubernetes-rbac/>

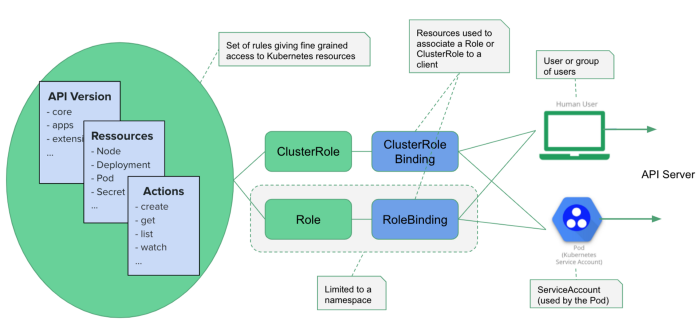
<https://trstringer.com/kubectl-from-within-pod/>

<https://itnext.io/running-kubectl-commands-from-within-a-pod-b303e8176088>

<https://jeremievallee.com/2018/05/28/kubernetes-rbac-namespace-user.html>

<https://www.stackrox.com/wiki/configuring-kubernetes-rbac/>

<https://medium.com/better-programming/k8s-tips-using-a-serviceaccount-801c433d0023>



# **Understanding Kubernetes RBAC**

<https://rancher.com/understanding-kubernetes-rbac>

Kubernetes introduced RBAC in 1.6 and has improved it in both 1.7 and 1.8. The Internet has many articles about how to activate and configure it, but I believe that we should know *why* we are doing things before we start doing them. For this post, I’ll take you back to a core concept of access control known as AAA (spoken as “triple A”). AAA defines the procedure for granting access to an application, network, building, or any other system as the combination of three components: authentication, authorization, and accounting/auditing.

RBAC model in Kubernetes consists of the three main components:

* Roles: defines permissions boundaries
* Subjects: *Users* (human or an application), or user groups
* RoleBingdings: specifies which *Subjects* have which *Roles*

Roles are a namespaced-resource consisting of rules that set permissions for individual namespaces, whereas ClusterRoles are non namespaced-resource that grant clusterwide permissions or permissions that span multiple namespaces. Each rule is a combination of verbs, resource types, and namespace selectors.