**Lesson 3 Demo 7**

**Kubernetes Dashboard Role-Based Access Control (RBAC)**

**Objective:** To implement a Role-based access control (RBAC) authorization on the

Kubernetes dashboard

**Tools required:** kubeadm, kubectl, kubelet, and etcd

**Prerequisites:** A Kubernetes cluster must be set up (follow steps of Lesson 2 Demo 1)

Steps to be followed:

1. Adding, deleting, and verifying cluster roles

**Step 1: Adding, deleting, and verifying cluster roles**

1. To find the resources available in the **kubernetes-dashboard**, use the following command:

**kubectl get sa -n kubernetes-dashboard**

Text

Description automatically generated

The following command will display all the existing **custerrolebindings**:

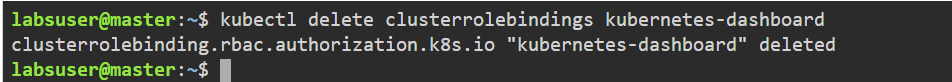
**kubectl get clusterrolebindings**

**Text

Description automatically generated**

1. To assign a new cluster role, delete the existing one if available.

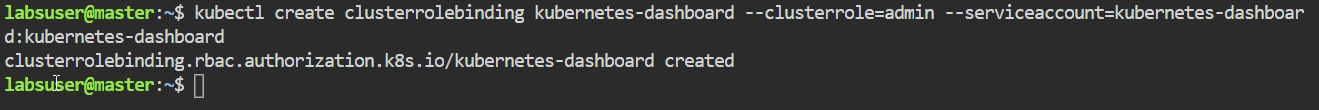
**kubectl delete clusterrolebindings kubernetes-dashboard**



|  |
| --- |
| Note: If a cluster role already exists, it will be deleted. |

1. Create a new cluster role and set the **kubernetes-dashboard** to admin.

**kubectl create clusterrolebinding kubernetes-dashboard --clusterrole=admin --serviceaccount=kubernetes-dashboard:kubernetes-dashboard**

****

1. Run the following command to verify the newly assigned admin cluster role:

**kubectl get clusterrolebindings**

**Text

Description automatically generated**

The **admin** cluster role is assigned to the **kubernetes-dashboard**, as seen in the screenshot above.