**Lesson 3 Demo 8**

**Implementing Role-Based Access Controller**



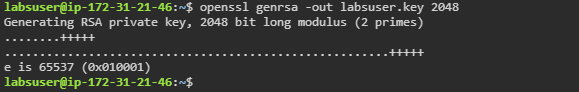
Steps to be followed:

1. Creating Client Certificate
2. Adding user credentials to the kubeconfig file
3. Creating Role and Role Binding
4. Checking the access for the new user

**Step 1: Creating Client Certificate**

1. Write the following to create a client key:

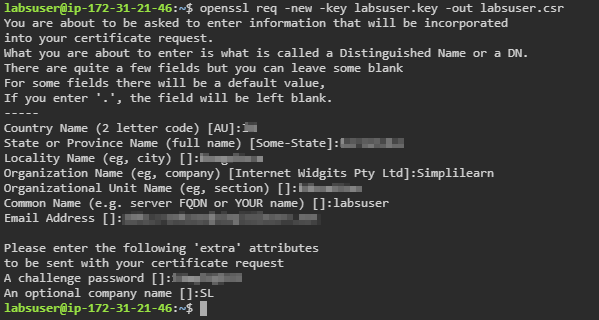
***openssl genrsa -out <<user>>.key 2048***



| **Note:** Labsuser is the regular user in our case. |
| --- |

1. Run the following command to create a certificate signing request:

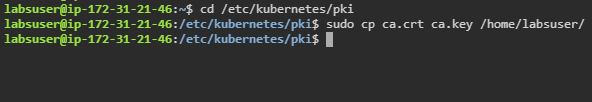
***openssl req -new -key labsuser.key -out labsuser.csr***



1. Execute the following commands to copy the CA certificate and key to the: **/home/labsuser/** folder:

***cd /etc/kubernetes/pki***

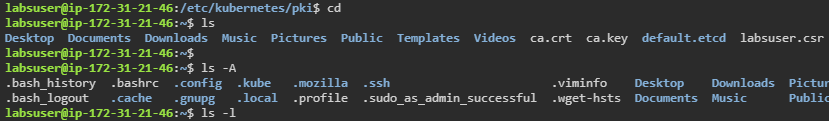
***sudo cp ca.crt ca.key /home/labsuser/***



1. Check if the certificate and key are copied to the root folder

***cd***

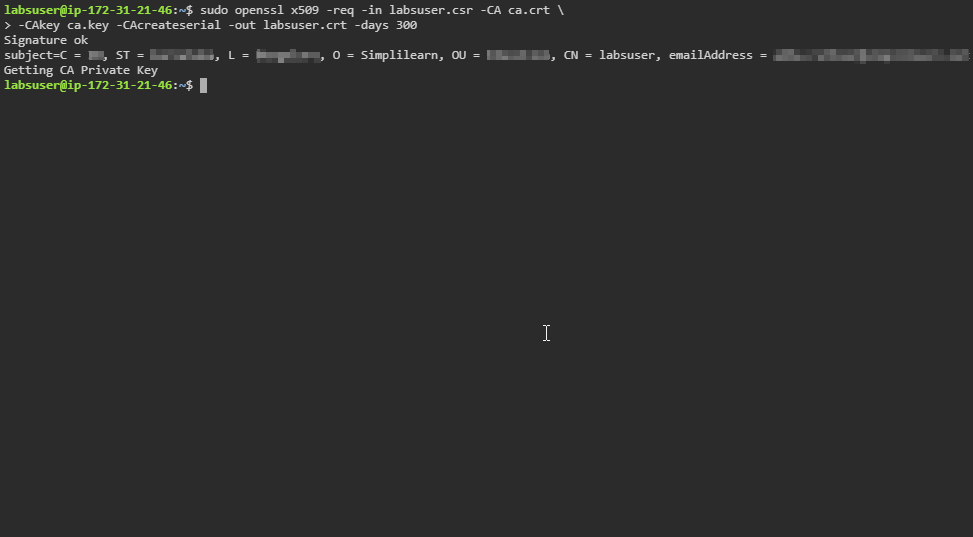
***ls***



1. Run the following command to sign the user key and request with cluster certificate and key:

***sudo openssl x509 -req -in labsuser.csr -CA ca.crt \***

***-CAkey ca.key -CAcreateserial -out labsuser.crt -days 300***

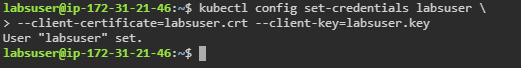


**Step 2: Adding user credentials to the kubeconfig file**

1. Use the following command to add user credentials:

***kubectl config set-credentials labsuser \***

***--client-certificate=labsuser.crt --client-key=labsuser.key***



1. Write the following command to test user permissions:

***sudo kubectl --user=labsuser get pods***



| **Note:** This will show errors as the roles are not bound yet. |
| --- |

**Step 3: Creating role and role binding**

1. Create a role using **role.yaml** file

***vi role.yaml***



1. Add the following code in the **role.yaml** file:

***kind: Role***

***apiVersion: rbac.authorization.k8s.io/v1***

***metadata:***

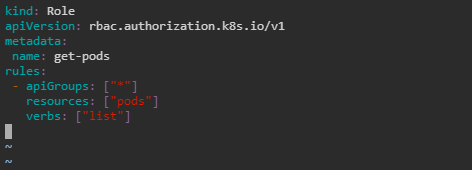
***name: get-pods***

***rules:***

***- apiGroups: ["\*"]***

***resources: ["pods"]***

***verbs: ["list"]***



1. Use the following command to create the role:

***kubectl apply -f role.yaml***



1. Create a role binding using **rolebinding.yaml** file

***vi rolebinding.yaml***



1. Add the following code in the **rolebinding.yaml** file:

***apiVersion: rbac.authorization.k8s.io/v1***

***kind: RoleBinding***

***metadata:***

***name: labuser-get-pods***

***subjects:***

***- kind: User***

***name: labsuser***

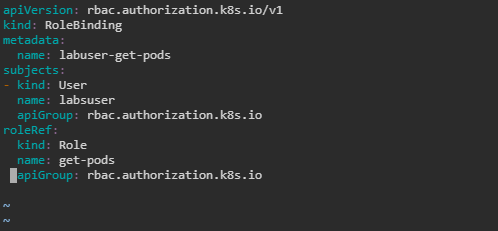
***apiGroup: rbac.authorization.k8s.io***

***roleRef:***

***kind: Role***

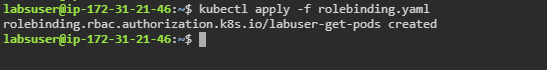
***name: get-pods***

***apiGroup: rbac.authorization.k8s.io***



1. Write the following command to create rolebinding.yaml:

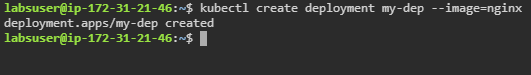
***kubectl apply -f rolebinding.yaml***



**Step 4: Checking the access for the new user**

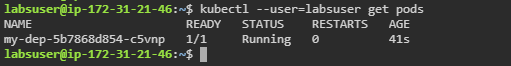
1. Use the following command to create a deployment using **nginx** image:

***kubectl create deployment my-dep --image=nginx***



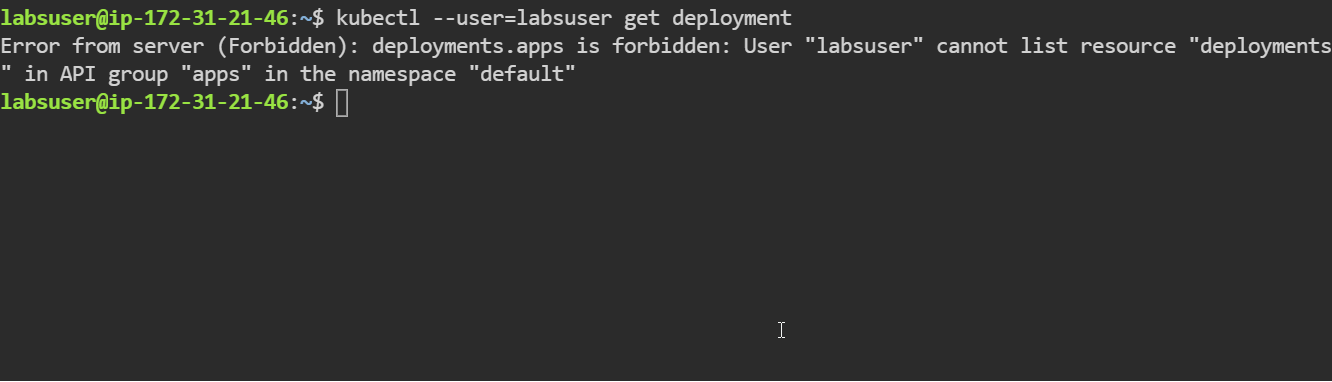
1. Write the following command to list pods:

***kubectl --user=labsuser get pods***



1. Write the following command to list the deployments:

***kubectl --user=labsuser get deployment***



| **Note:** Notice that the user **labsuser** can only list pods and cannot list other resources because we have defined the labsuser role to only list pod resources. |
| --- |