Lesson 02 Demo 06

Implementing RBAC Using Namespaces

Objective: To set up and verify role-based access control (RBAC) in Kubernetes using namespaces for secure and efficient user management

Tools required: kubeadm, kubectl, kubelet, and containerd

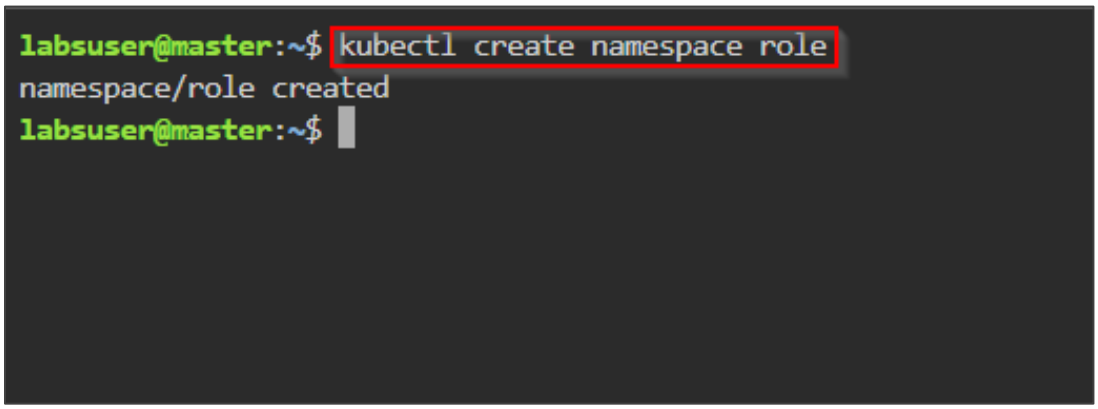
Prerequisites: A Kubernetes cluster (refer to Demo 01 from Lesson 01 for setting up a cluster)

Steps to be followed:

1. Create a namespace in the master node
2. Generate an RSA private key and certificate signing request
3. Create a role
4. Create a role binding service
5. Set up credentials for the user
6. Verify the roles

Step 1: Create a namespace in the master node

1.1 Use the following command to create a namespace:
kubectl create namespace role



1.2 Create and navigate to the **role** directory with these commands:

mkdir role

cd role

```
labsuser@master:~$ kubectl create namespace role
namespace/role created
labsuser@master:~$ mkdir role
labsuser@master:~$ cd role
labsuser@master:~/role$
```

Step 2: Generate an RSA private key and certificate signing request

2.1 Use the following commands to generate a 2048-bit RSA private key and create a CSR:

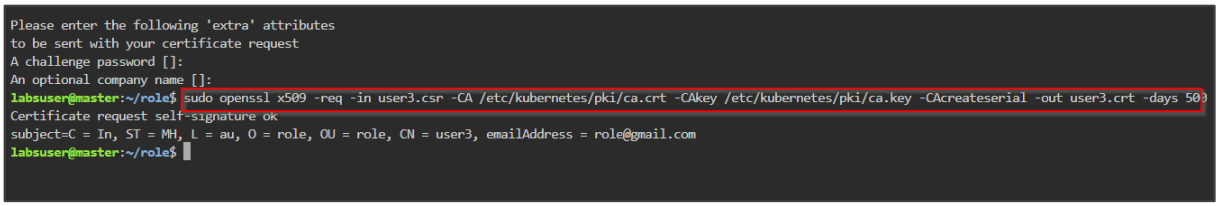
sudo openssl genrsa -out user3.key 2048

sudo openssl req -new -key user3.key -out user3.csr

```
labsuser@master:~/role$ sudo openssl genrsa -out user3.key 2048
labsuser@master:~/role$ sudo openssl req -new -key user3.key -out user3.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:In
State or Province Name (full name) [Some-State]:MH
Locality Name (eg, city) []:au
Organization Name (eg, company) [Internet Widgits Pty Ltd]:role
Organizational Unit Name (eg, section) []:role
Common Name (e.g. server FQDN or YOUR name) []:user3
Email Address []:role@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
labsuser@master:~/role$
```

2.2 Link an identity to the private key using a digital signature with the following command:
sudo openssl x509 -req -in user3.csr -CA /etc/kubernetes/pki/ca.crt -CAkey /etc/kubernetes/pki/ca.key -CAcreateserial -out user3.crt -days 500



Step 3: Create a role

3.1 Run the following command to create a **role.yaml** file:
vi role.yaml

3.2 Add the following code to the **role.yaml** file:

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: role
  name: user3-role
rules:
- apiGroups: ["", "extensions", "apps"]
  resources: ["deployments", "pods", "services"]
  verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]
```

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
   namespace: role
   name: user3-role
rules:
- apiGroups: ["", "extensions", "apps"]
  resources: ["deployments", "pods", "services"]
  verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]

~
~
```

3.3 Create a role with the following commands:

kubectl create -f role.yaml

kubectl get roles -n role

```
labsuser@master:~$ vi role.yaml
labsuser@master:~$ kubectl create -f role.yaml
role.rbac.authorization.k8s.io/user3-role created
labsuser@master:~$ kubectl get roles -n role
NAME         CREATED AT
user3-role   2023-10-09T09:25:16Z
labsuser@master:~$
```

Step 4: Create a role binding service

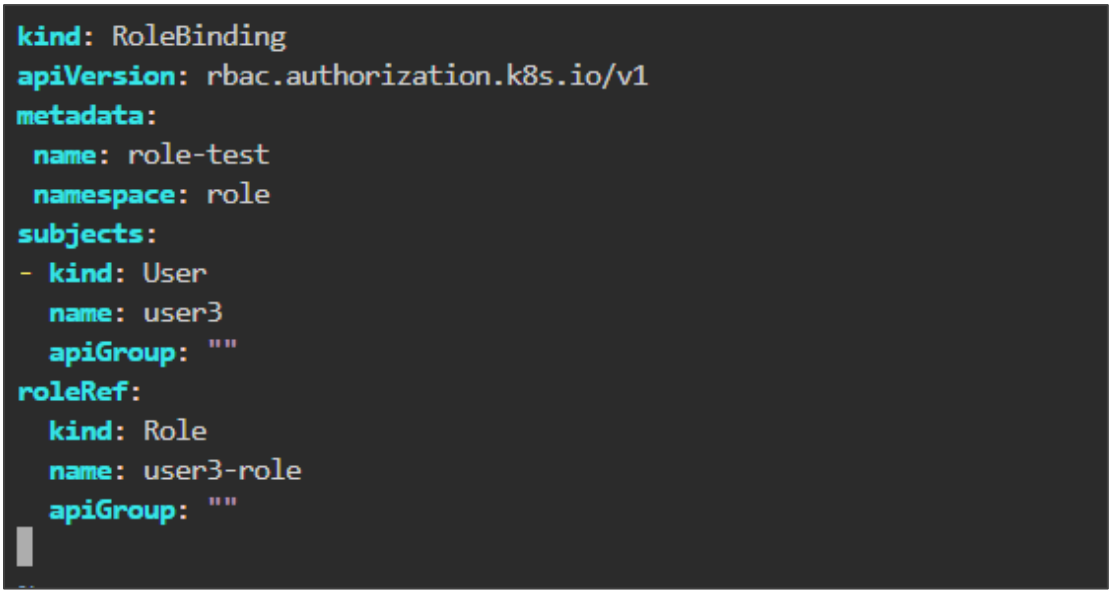
4.1 Run the following command to create a **rolebinding.yaml** file:

vi rolebinding.yaml

```
labsuser@master:~$ vi rolebinding.yaml
```

4.2 To create a role binding service, add the following code to the **rolebinding.yaml** file:

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
 name: role-test
 namespace: role
subjects:
- kind: User
  name: user3
  apiGroup: ""
roleRef:
  kind: Role
  name: user3-role
  apiGroup: ""
```



4.3 Create a role binding with the following commands:

kubectl create -f rolebinding.yaml

kubectl get rolebinding -n role

```
labsuser@master:~$ vi rolebinding.yaml
labsuser@master:~$ kubectl create -f rolebinding.yaml
rolebinding.rbac.authorization.k8s.io/role-test created
labsuser@master:~$ kubectl get rolebinding -n role
NAME        ROLE              AGE
role-test   Role/user3-role   6s
labsuser@master:~$
```

Step 5: Set up credentials for the user

5.1 Assign credentials to user3 using the following command:

kubectl config set-credentials user3 --client-certificate=/home/labsuser/role/user3.crt --client-key=/home/labsuser/role/user3.key

```
labsuser@master:~$ kubectl config set-credentials user3 --client-certificate=/home/labsuser/role/user3.crt --client-key=/home/labsuser/role/user3.key
User "user3" set.
labsuser@master:~$
```

5.2 Set the context for user3 using the following command:

kubectl config set-context user3-context --cluster=kubernetes --namespace=role --user=user3

```
labsuser@master:~$ kubectl config set-credentials user3 --client-certificate=/home/labsuser/role/user3.crt --client-key=/home/labsuser/role/user3.key
User "user3" set.
labsuser@master:~$ kubectl config set-context user3-context --cluster=kubernetes --namespace=role --user=user3
Context "user3-context" created.
labsuser@master:~$
```

5.3 Display current contexts using the following command:

kubectl config get-contexts

```
labsuser@master:~$ kubectl config set-credentials user3 --client-certificate=/home/labsuser/role/user3.crt --client-key=/home/labsuser/role/user3.key
User "user3" set.
labsuser@master:~$ kubectl config set-context user3-context --cluster=kubernetes --namespace=role --user=user3
Context "user3-context" created.
labsuser@master:~$ kubectl config get-contexts
CURRENT   NAME                          CLUSTER      AUTHINFO           NAMESPACE
*         kubernetes-admin@kubernetes   kubernetes   kubernetes-admin
          user3-context                 kubernetes   user3              role
labsuser@master:~$
```

5.4 Navigate to the home directory and view the **config** file using the following commands:

cd ..

cat .kube/config

```
labsuser@master:~/role$ cd ..
labsuser@master:~$ cat .kube/config
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJRUUzVjBJc0doemd3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1WlhSbGN6QWVGdzB5TXpFd01Ea3dOekU1TXpKYUZ3MHpNekV3TURZd05UazVNekphTUJVeApFekFFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU01iM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUURYbWRuYWpPZGpxU2s2YmllQzc4dmhBSmZQWUdscWxPY2FQS2ZtK2VaU2lLVHM0YXNUd1lNNHBFWjIKcDhuYW9NNTdpd0wrK0xaaXA2amRDQzhUcUV4M0JkSFhmTFF0ZlVJd2k2c2RmNWdQK0xRL3ZsQ0NjVVg0UF
dPdwpNbXExMzFOdWU2L0tBR0FkL0dVMVJwSkQ4TTlTM3JMNnlUd2FwcUcxb01CSUxKNC9MVFEwUVVyZWlkTW9pcUphClNQbDBKbHZZZk9DdkVrOTdOamVSdEFpa1VDb3ViZnMwTDJSOGVDdXhoUHl5RXl5SWttemNHOFdH
aEpPNUJscU0KQUczck1JWWVHYkRRM2RsbDE3RWVRUjhMbVk1UWlhZWtqTmhXenlkNUFkd0FLMlR5Nzc3emNFclV0VzZodUZ2cQo5WUJnTFB0SERDaE0wQmhncEgzNlZ5c28xZGxMQWdNQkFBR2pXVEJYTUE0R0ExVWREd0
VCL3dRRUF3SUNwREFQCkJnTlZIUk1CQWY4RUJUQURBUUgvTUIwR0ExVWREZ1FXQkJSVXo2enhSMU9sbVlwUUdPTm1IS1BvNVVpNTR6QVYKQmdOVkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdWek1BMEdDU3FHU01iM0RRRUJD
d1VBQTRJQkFRQ0FDaXNQTVB0VgorQkdjWnJaWkNIWitKK1kxbmVWd3BRTmw5R1NXOWtudTB2OGFxcVk0aUZOZTlIbUFwTEFQMXlWQ2xLc0FzUHJuClZUQU9GcWxiUm41b3c5VzdMK3Y2N0lFLzFRSXc2MTdBUzlJTytiQm
8zTmhnYVNiNmdqeGlEblBaa0FydC9EcFgKYWpVUDRJaHQ4VE90Um5hNlJmVFA3bXJzY05tb2ZFRUFZYjUzUDBKSXljNGNL05pdi9HQzNseFFENDRxWjcwMQpjUXVUODdFQi9SQnh4QWl3emFXZFlDOWNnMzgrM2JxVTMr
eEs1ZXVJNEh2NlBUK1ZFcFBqSHZwL0M2QlQ4ZmFHCi9PUlBpdEpLZytEbnp5SHoyc3JJcnRRM1p1eHAxWjdRUnBET0xETmk3MkIvLzJqVDJvbmludlNmQUE2SGZuNmoKS3dCNHdxZWtUWjYzCi0tLS0tRU5EIENFUlRJRk
lDQVRFLS0tLS0K
    server: https://172.31.3.59:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
- context:
    cluster: kubernetes
    namespace: role
    user: user3
  name: user3-context
```

5.5 Copy the **config** file as shown below:

```
labsuser@master:~$ cat .kube/config
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJRUUzVjBJc0doemd3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1WlhSbGN6QWVGdzB5TXpFd01Ea3dOekU1TXpKYUZ3MHpNekV3TURZd056STBNekphTUJVeApFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUURYbWRuYWpPZGpxU2s2YmllQzc4dmhBSmZQWUdscWxPY2FQS2ZtK2VaU2lLVHM0YXNUd1lNNHBFWjIKcDhuYW9NNTdpd0wrK0xaaXA2amRDQzhUcUV4M0JkSFhmTFF0ZlVJd2k2c2RmNWdQK0xRL3ZsQ0NjVVg0UF
dPdwpNbXExMzFOdWU2L0tBR0FkL0dVMVJwSkQ4TTlTM3JMNnlUd2FwcUcxb01CSUxKNC9MVFEwUVVyZWlkTW9pcUphClNQbDBKbHZZZk9DdkVrOTdOamVSdEFpa1VDb3ViZnMwTDJSOGVDdXhoUHl5RXl5SWttemNHOFdH
aEpPNUJscU0KQUczck1JWWVHYkRRM2RsbDE3RWVRUjhMbVk1UWlhZWtqTmhXenlkNUFkd0FLMlR5Nzc3emNFclV0VzZodUZ2cQo5WUJnTFB0SERDaE0wQmhncEgzNlZ5c28xZGxMQWdNQkFBR2pXVEJYTUE0R0ExVWREd0
VCL3dRRUF3SUNwREFQCkJnTlZIUk1CQWY4RUJUQURBUUgvTUIwR0ExVWREZ1FXQkJSVXo2enhSMU9sbVlwUUdPTm1ISlBvNVVpNTR6QVYKQmdOVkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJD
d1VBQTRJQkFRQ0FDaXNQTVB0VgorQkdjWnJaWkNIWitKK1kxbmVWd3BRTmw5R1NXOWtudTB2OGFxcVk0aUZOZTlIbUFwTEFQMXlWQ2xLc0FzUHJuClZUQU9GcWxiUm41b3c5VzdMK3Y2N0lFLzFRSXc2MTdBUzlJTytiQm
8zTmhnYVNiNmdqeGlEblBaa0FydC9EcFgKYWpVUDRJaHQ4VE90Um5hNlJmVFA3bXJzY05tb2ZFRUFZYjUzUDBKSXljNGNLN05pdi9HQzNseFFENDRxWjcwMQpjUXVUODdFQi95SQnh4QWl3emFXZFlDOWNnMzgrM2JxVTMr
eEs1ZXVJNEh2NlBUK1ZFcFBqSHZwL0M2QlQ4ZmFHCi9PUlBpdEpLZytEbnp5SHoyc3JJcnRRM1p1eHAxWjdRUnBET0xETmk3MkIvLzJqVDJvbmludlNmQUE2SGZuNmoKS3dCNHdxZWtUWjYzCi0tLS0tRU5EIENFULRJRk
lDQVRFLS0tLS0K
    server: https://172.31.3.59:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
- context:
    cluster: kubernetes
    namespace: role
    user: user3
  name: user3-context
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
```

5.6 On the client machine, run the following command:

vi myconf

```
labsuser@worker-node-1:~$ vi myconf
```

5.7 Then, paste the copied config file as shown below:

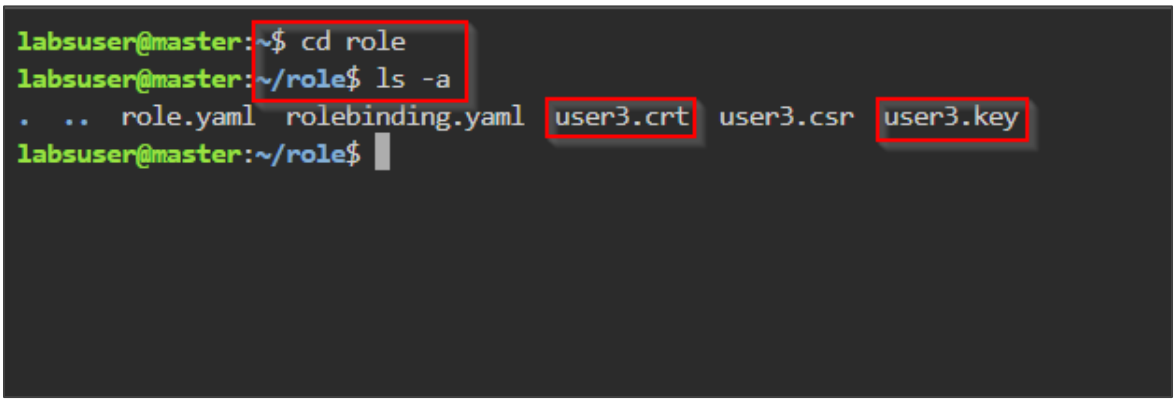
```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJRUUzVjBJc0doemd3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1WlhSbGN6QWVGdzB5TXpFd01Ea3dOekU1TXpKYUZ3MHpNekV3TURZd056STBNekphTUJVeApFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUURYbWRuYWpPZGpxU2s2YmllQzc4dmhBSmZQWUdscWxPY2FQS2ZtK2VaU2lLVHM0YXNUd1lNNHBFWjIKcDhuYW9NNTdpd0wrK0xaaXA2amRDQzhUcUV4M0JkSFhmTFF0ZlVJd2k2c2RmNWdQK0xRL3ZsQ0NjVVg0UF
dPdwpNbXExMzFOdWU2L0tBR0FkL0dVMVJwSkQ4TTlTM3JMNnlUd2FwcUcxb01CSUxKNC9MVFEwUVVyZWlkTW9pcUphClNQbDBKbHZZZk9DdkVrOTdOamVSdEFpa1VDb3ViZnMwTDJSOGVDdXhoUHl5RXl5SWttemNHOFdH
aEpPNUJscU0KQUczck1JWWVHYkRRM2RsbDE3RWVRUjhMbVk1UWlhZWtqTmhXenlkNUFkd0FLMlR5Nzc3emNFclV0VzZodUZ2cQo5WUJnTFB0SERDaE0wQmhncEgzNlZ5c28xZGxMQWdNQkFBR2pXVEJYTUE0R0ExVWREd0
VCL3dRRUF3SUNwREFQCkJnTlZIUk1CQWY4RUJUQURBUUgvTUIwR0ExVWREZ1FXQkJSVXo2enhSMU9sbVlwUUdPTm1ISlBvNVVpNTR6QVYKQmdOVkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJD
d1VBQTRJQkFRQ0FDaXNQTVB0VgorQkdjWnJaWkNIWitKK1kxbmVWd3BRTmw5R1NXOWtudTB2OGFxcVk0aUZOZTlIbUFwTEFQMXlWQ2xLc0FzUHJuClZUQU9GcWxiUm41b3c5VzdMK3Y2N0lFLzFRSXc2MTdBUzlJTytiQm
8zTmhnYVNiNmdqeGlEblBaa0FydC9EcFgKYWpVUDRJaHQ4VE90Um5hNlJmVFA3bXJzY05tb2ZFRUFZYjUzUDBKSXljNGNLN05pdi9HQzNseFFENDRxWjcwMQpjUXVUODdFQi95SQnh4QWl3emFXZFlDOWNnMzgrM2JxVTMr
eEs1ZXVJNEh2NlBUK1ZFcFBqSHZwL0M2QlQ4ZmFHCi9PUlBpdEpLZytEbnp5SHoyc3JJcnRRM1p1eHAxWjdRUnBET0xETmk3MkIvLzJqVDJvbmludlNmQUE2SGZuNmoKS3dCNHdxZWtUWjYzCi0tLS0tRU5EIENFULRJRk
lDQVRFLS0tLS0K
    server: https://172.31.3.59:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
- context:
    cluster: kubernetes
    namespace: role
    user: user3
  name: user3-context
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
"myconf" 28L, 5862B                                                                    1,1           Top
```

```
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURJVENDQWdtZ0F3SUJBZ0lJWFBXL05wTiszTEl3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0E
xVUUKQXhNS2EzVmlaWEp1WlhSbGN6QWVGdzB5TXpFd01UY3dOelE0TkRCYUZ3MHlOREV3TVRZd056VXpORE5hTURReApGekFWQmdOVkJBb1REbk41YzNSbGJUcHRZWE4wWlhKek1Sa3dGd1lEV1F
RREV4QnJkV0psY201bGRHVnpMV0ZrCmJXbHVNSUlCSWpBTkJna3Foa2lHOXcwQkFRRUZBQU9DQVE4QU1JSUJDZ0tDQVFFQXVCTElSdzg2VmRIL01lJS3UKOTdDbjNiQW5aNzJLYngxSksxSFUxWFp
3UzUyQmtkbnYyUHRMVGNscG1IaXk3WEViWmlKNHE2dXRrR253chNLdApLQXdNRVc0S0ZVYUR6ZVc1YVAzV3U1WFMwbmVpTXp4SnJ6VkxPbnJsdHBtREN5ZDNYV1A1NkVEeVBjV1RvUDBzCjlwVXh
lbUI2aW44Z2NGNEtqdE43WHI5T2g3eE5sMy94VVc2ZDAwcVRSR1BBL2xoSEQvWTl5RThhclU3NVpYcmMKMHVaMkJvYWlLODRUbmo3a3pSaHh6eXR1cHhRM2Q2UzZ4dndPWTBFbE1DdHNOTi8wNEV
PenRYZjdLYmRkVTVIQgo2SWRsQ0NUT3QvSFp6bHFVK3Q1MmJ1MXVVWxoV2trQjVRdzZWS0FiVmRyTG93cXhZZ2VuTFY2dmpKRW92TUxnCjlUODJkUUlEQVFBQm8xWXdWREFPQmdOVkhROEJBZjh
FQkFNQ0JhQXdFd1lEVlIwbEJBd3dDZ1lJS3dZQkJRVUgKQXdJd0RBWURWUjBUQVFIL0JBSXdBREFmQmdOVkhTTUVHREFXZ0JUcmRTdWt1ZSs1L3BFWHNmaW8yUU40cUNGVQpHVEFOQmdrcWhraUc
5dzBCQVFzRkFBT0NBUUVBQkp5ZWYzdkV5b25NSXh3Q29pWCtTYi9DS0JuSmt1VU9lcE15CjRFbGhudmozMU56VVB6eWVoNnZmTGM2NjVNM05IOVd5ZWkxTGtNN0xUTlp3L0VjQVRWdzYwYzhVdFh
HaFU3eUQKUTB5OGNJUDZlcHJhVjBSKytkSldKYzN0TVc2b1lVZkpocEFnTjNIR2hKbnhNQmYyNDNqTFNYR2VBbHZGKzY2MwpuNDJmQTAzT2pWV1hwcGZMemZIUHZXMWVzbjBNVEFTUXlRcnBxMWt
FRkFBT2NoN0dQaEhneEkrR2V4dm5zSUlOCit5Qm5ueUM3elpGcW9sZFloU3pwRzlQS29KQ2l2RkphcTVwZ3F4LzI2QjJmdHhzUEd3VGFhOGY5ZTUyU3lsR0IKTzgxL2ZqclhpYjZ6bitpcHphdWV
XUUdvNU5GOWd1ZUF2elp5Z04zd2pYek1heGk1VHc9PQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
    client-key-data: LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFb3dJQkFBS0NBUUVBdUJMSVJ3ODZWZEgvSUlLdTk3Q24zYkFuWjcyS2J4MUpLMUhVMVhad1M1MkJrZG5
2CjJQdExuY2xwbUhpeTdYRWJaaUo0cTZ1dGtHbndwc0t0S0F3TUVXNEtGVWFEemVXNWFQM1d1NVhTMG5laU16eEoKcnpWTE9ucmx0cG1EQ3lkM1hXUDU2RUR5UGNXVG9QMHM5cFV4ZW1CNmluOGd
jRjRLanRON1hyOU9oN3hObDMveApVVzZkMDBxVFJHUEEvbGhIRC9ZOXlFOGFyVTc1WlhyYzB1WjJCb2FpSzg0VG5qN2t6Umh4enl0dXB4UTNkNlM2Cnh2d09ZMEVsTUN0c050OLzA0RU96dFhmN0t
iZGRVNUhCNklkbENDVE90L0haemxxVSt0NTJidTF1VVVsaFdra0IKNVF3NlZLQWJWZHJMb3dxeFlnZW5MVjZ2akpFb3ZNTGc5VDgyZFFJREFRQUJBb0lCQUJlbGVnciszc3lWaklPQQpwZ0oxdEN
yYmJVUjdhZVE2UytML3dTcW02MmdiVk1ESzBXLzdZRUllNUt6TW5KZTZGdjM0NGEyVUFWank4eEpKCklmdHFDTWJZYnYwcXNOWHJjL1VUM3RqOE56Z0lYRkxlcE15WU9SRExKdVZkTlNOaXI0ZlV
4dzdhQVFTRWRDR1QKUklwc3dnMXBoWFJ6UGxTUkVDUW04TXRNSWlkM2htUzJoSnF4THIvM3VEc1Z2ZDJRUTVnUjQ5LzhobEl5K1h1cgpLYTA1OVVqbWxtc1JnWmp2aFYwY2JyMXFRVTI3aFVwTDd
HYjZ2R0I3SmVwbXQ0RzhrQTlPRmVFeDdJdmwyRlFJCkVuOWVBVWVIWlAxbGwzaDVYMFEvQnFJZllZNVlhNDBhTTYyN2tkbklsMkhhUUsvRGg5V0hPYjlhM3JYYm1FQkwKWUtvMW5RVUNnWUVBNFN
tZ0UzS093U0FOUmdkQTZZWFJEM2h2c1VwdTZDTG11ZGtLaUIxM3h6VTUxejRxZWxiZAplWWJTQWZOTFFLajN3OW9LZVFWS01QN2NMNDNacjRVMTNxWjgrYkcyRzNwbWNCektLTTFkS2p0eVYxZk5
4TXFWCnZJZE1scXA3THVGZ3FHMENJaHNobnVQVGtwRkRsNzNJbEpBUmM3Nlg2MVRudG1XYnVYUEtEdnNDZ1lFQTBVaUoKY0cyTmNkQkRwcHkxR2xQU2o5cTkzSTdrU3diCXBSYjhXTFh1RXd0eFh
QU0RjQmdBa0RVQW1IemlFczhMdWhNKwpDbE90TTU3NU5zZksya0pCcEwvT3gycHkrWElBdkVNRmpXTEtIMkxqcGxTNGNhSTZDeWIyODdYS0dzWXM5YklGCk0yaytaR3hCTzV4ZmEwd2VzejVzVTB
nemVKSnkrOURUQkt2ZUZVOENnWUJJZjh5cE5tTzh6NE0ybnpOSmZlcjQKM2tDb2gxejZwZkpBdVIvcEFBUUhRcDRqTGtNakxVRVd3Yl1R4SjlOOU9HcSt4MEJFQTI0R0pLbmxMZmwveUphMQo0RW1
4WklWTWg3YmdOb1RwYW1hcktJVUNscVVNem1ZMFZDUlhCd0d1OEtiYzFHeXhhcWNDc0R6NW5vaHpIWXF1Ck8wMURSWVZKSXdlUTlWcVU5akdEY1FLQmdGUmVYYW5KUUdCRmlVQTlwSFM4bmdJSU9
Nb1ZnTXNPT3NIYkJsZnMKMzlFbHA4bkZlTWRUcGNaODE0U2hzN0hKYmZ6UmpRMDB0bm84K0ZLeE1JSFVmQW9pbVFXalVENW5Td3lhS0Q3YgpPSWgyb1RlSXArS0hkOGJQS25jTkxqQ2srN01BMVQ
vb2laS0tZYkhhVkUwNUhwYzc3d2VUNzh3SG5mc09NRU9OCjFiWFpBb0dCQUpKSWMvalBLRE9BVzl3VzlRYWFNVVJqejlEbERlUU1MdU0yY0YvaDlkQzZiNzBrWG0yblZTbVQKNVpnb3o5VDJUcW1
2L2I1VFJ5TGRjb3hueWpUYnB5QzBOb0VNNiszUXNJSWRmZlJUVW4vNXhRM0d6YnZqT1hBdwo4WGpRd25XUUJqZE50NXhPWkVpQWE4anhJOVRpWmxGa0p0aWtDOVNGZ0dxOE5MK0FnWFp6Ci0tLS0
tRU5EIFJTQSBQUklWQVRFIEtFWS0tLS0tCg==
```


5.8 Copy the .**crt** and .**key** files from the master node to the client node in the **/role** directory using the following commands:

cd role

ls -a



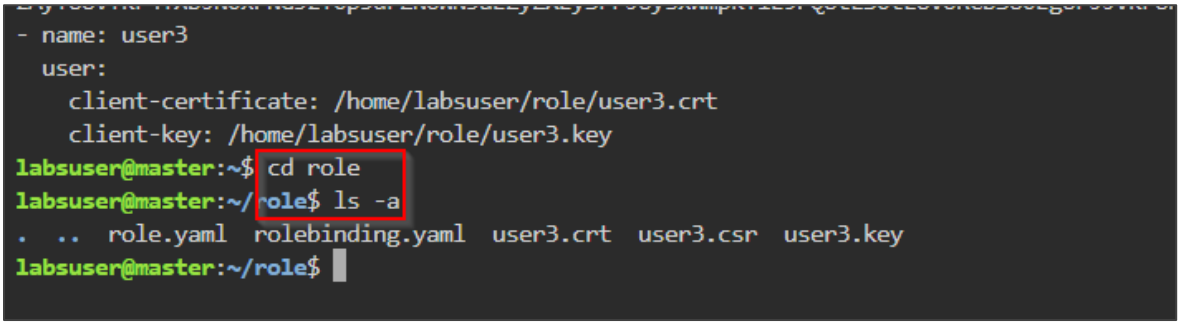
5.9 On the client machine, display the contents of the **myconf** file using the **cat** command:

cat myconf

5.10 Navigate to the **/role** directory and list its contents using the following commands:

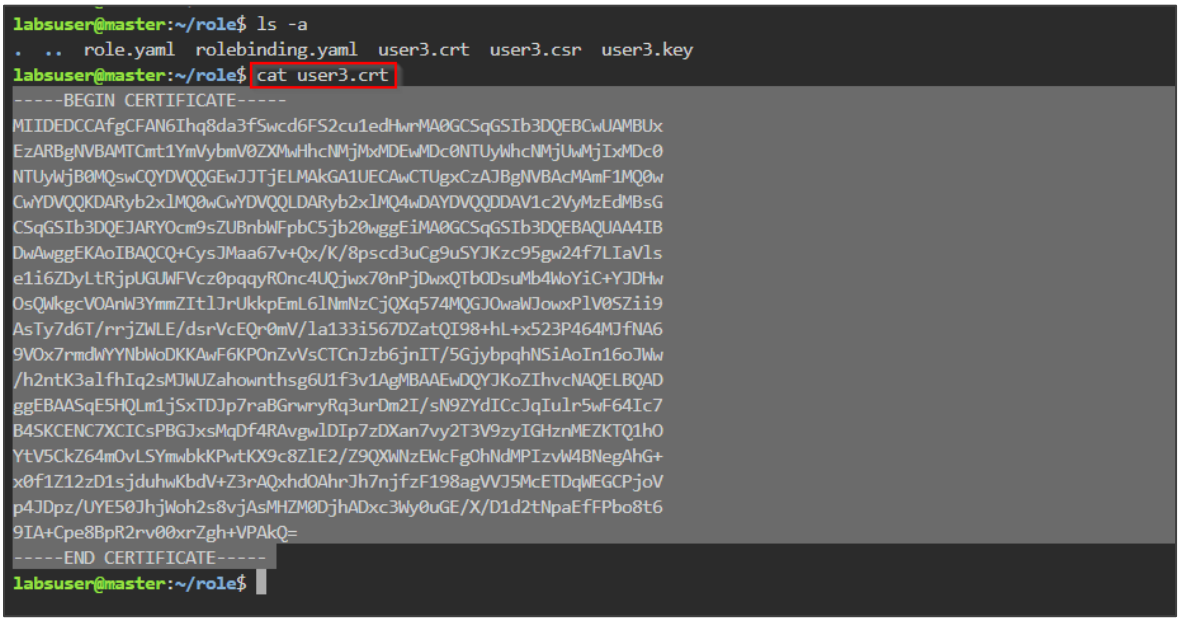
cd role

ls -a



5.11 View the **user3.crt** file on the master node using the command below and then copy the code as shown in the screenshot:

cat user3.crt



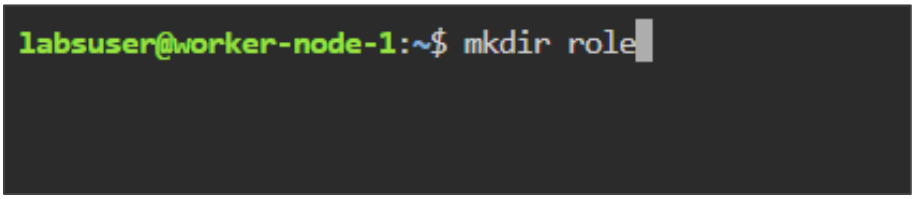
5.12 On **worker-node-1**, enter the following commands and then paste the copied code into the **/role** directory:

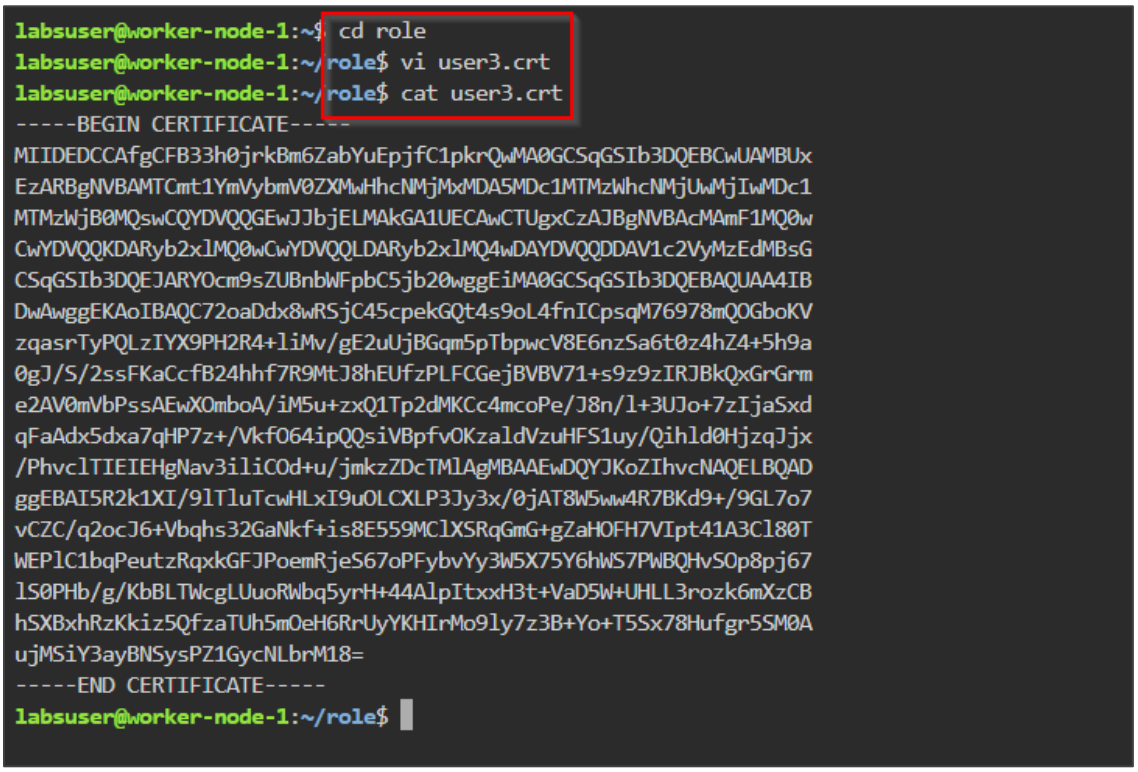
mkdir role

cd role

vi user3.crt

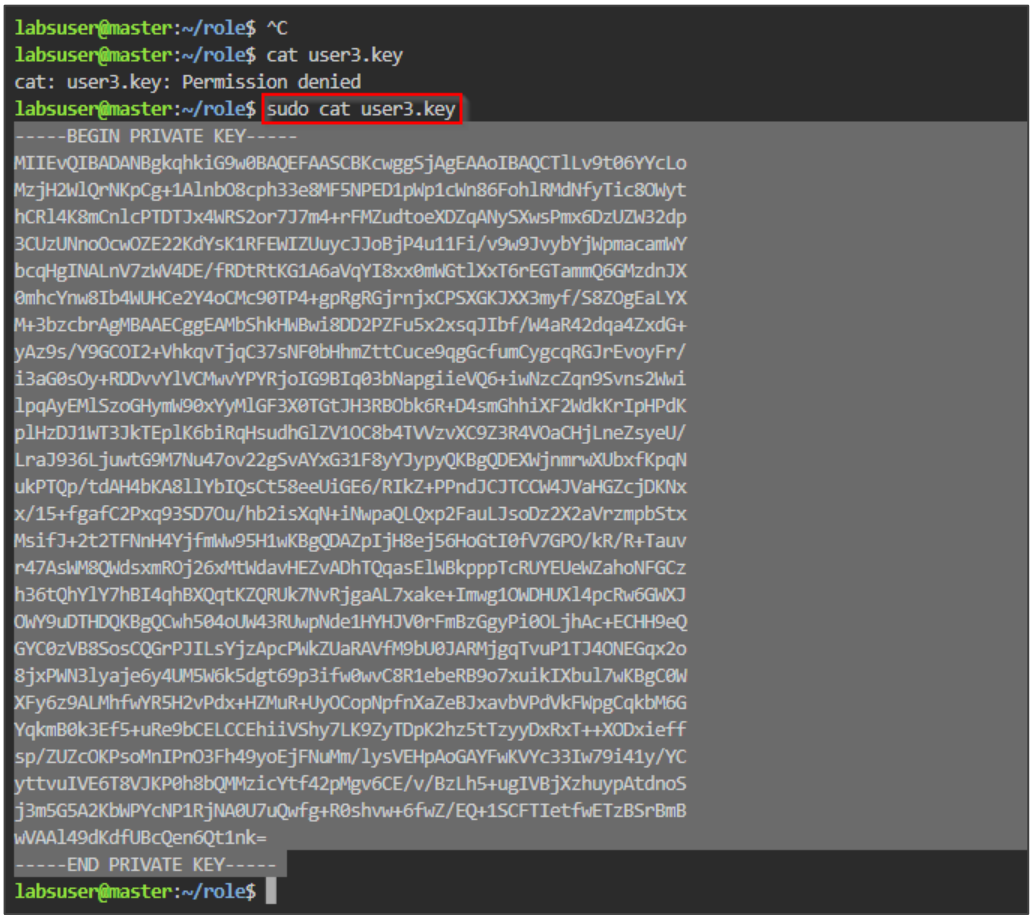
cat user3.crt





5.13 Navigate to the master node, use the following command to access the key, and then copy the code as shown in the screenshot below:

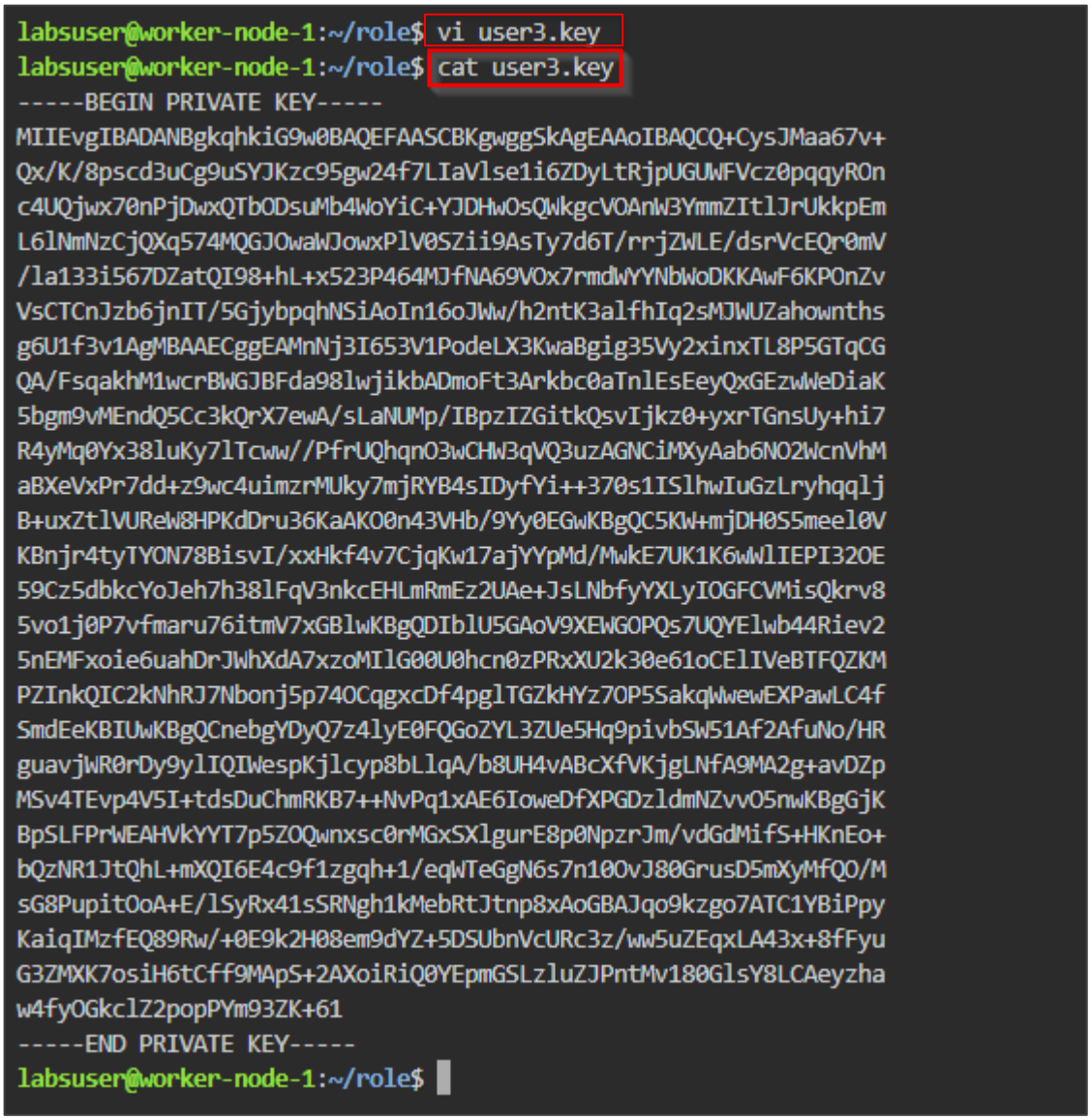
sudo cat user3.key



5.14 Copy and paste the key into **worker-node-1** using the following commands:

vi user3.key

cat user3.key



Step 6: Verify the roles

6.1 Navigate to the home directory using the following command:

cd ..

```
labsuser@worker-node-1:~$ cd ..
```

6.2 Validate role permissions by listing pods, initiating a test deployment, and checking deployment and pod statuses with the following commands:

kubectl get pods --kubeconfig=myconf

kubectl create deployment test --image=docker.io/httpd -n role --kubeconfig=myconf

kubectl get pods --kubeconfig=myconf

kubectl get deployment --kubeconfig=myconf

```
labsuser@worker-node-1:~$ kubectl get pods --kubeconfig=myconf
NAME                      READY   STATUS                  RESTARTS      AGE
admin-56d684dff9-7tzkl    0/1     ImagePullBackOff        0             160m
frontend-4ff8l            1/1     Running                 1 (18m ago)   45m
frontend-jvd98            1/1     Running                 1 (18m ago)   45m
mydep-548c7db5df-mxk4t    0/1     CreateContainerError    0             59m
mydep-6f74bcdf49-jjw2g    0/1     CreateContainerError    0             60m
nginx                     0/1     Pending                 0             72m
security-context-1        1/1     Running                 1 (18m ago)   41m
labsuser@worker-node-1:~$ kubectl create deployment test --image=docker.io/httpd -n role --kubeconfig=myconf
deployment.apps/test created
labsuser@worker-node-1:~$ kubectl get pods --kubeconfig=myconf
NAME                      READY   STATUS                  RESTARTS      AGE
admin-56d684dff9-7tzkl    0/1     ImagePullBackOff        0             161m
frontend-4ff8l            1/1     Running                 1 (19m ago)   46m
frontend-jvd98            1/1     Running                 1 (19m ago)   46m
mydep-548c7db5df-mxk4t    0/1     CreateContainerError    0             60m
mydep-6f74bcdf49-jjw2g    0/1     CreateContainerError    0             60m
nginx                     0/1     Pending                 0             73m
security-context-1        1/1     Running                 1 (19m ago)   42m
labsuser@worker-node-1:~$ kubectl get deployment --kubeconfig=myconf
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
admin   0/1     1            0           161m
mydep   0/1     1            0           61m
labsuser@worker-node-1:~$
```

6.3 Run the following commands to verify and create a ConfigMap named **my-config** in the **role** namespace using the **myconf kubeconfig** file:

kubectl create configmap my-config --from-literal=key1=config1 --kubeconfig=myconf

kubectl get configmaps --kubeconfig=myconf

kubectl get configmap my-config --kubeconfig=myconf -o yaml

```
labsuser@worker-node-1:~$ kubectl get deployment --kubeconfig=myconf
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
myhttpd       0/1     1            0           3h21m
redis-cache   1/3     3            1           3d4h
web-server    1/3     3            1           3d3h
labsuser@worker-node-1:~$ kubectl create configmap my-config --from-literal=key1=config1 --kubeconfig=myconf
configmap/my-config created
labsuser@worker-node-1:~$
```

```
labsuser@worker-node-1:~$ kubectl get configmaps --kubeconfig=myconf
NAME                  DATA   AGE
deployment-configmap  2      3h39m
kube-root-ca.crt      1      4d2h
my-config             1      17m
labsuser@worker-node-1:~$ kubectl get configmap my-config --kubeconfig=myconf -o yaml
apiVersion: v1
data:
  key1: config1
kind: ConfigMap
metadata:
  creationTimestamp: "2023-11-03T11:50:05Z"
  name: my-config
  namespace: default
  resourceVersion: "157363"
  uid: 62db8bbc-c41e-4445-9cb5-78ce259b2cbc
labsuser@worker-node-1:~$
```

By following these steps, you have successfully configured and verified RBAC in Kubernetes using namespaces.