Summary

- Revision of the previous session on the certificate, role binding & role
- Role-based access control is an authorization mechanism for managing permissions around Kubernetes resources
- The actions a user can perform within a cluster or namespace is called role
- The role contains resources (pod, deployment) & verbs (get, list)
- Practical: Role-based access control
    - Creating certificate
        - Generating private key

```
# pwd
/var/lib/minikube/certs
#
# openssl  genrsa  -out eric.key  1024
Generating RSA private key, 1024 bit long modulus (2 primes)
.....................................+++++
...+++++
e is 65537 (0x010001)
# ls
apiserver-etcd-client.crt     eric.key                proxy-client.crt
apiserver-etcd-client.key     etcd                    proxy-client.key
apiserver-kubelet-client.crt  front-proxy-ca.crt      sa.key
apiserver-kubelet-client.key  front-proxy-ca.key      sa.pub
apiserver.crt                 front-proxy-client.crt  vimal.crt
apiserver.key                 front-proxy-client.key  vimal.csr
ca.crt                        proxy-client-ca.crt     vimal.key
ca.key                        proxy-client-ca.key
```

        - Creating certificate signing request

```
# openssl req  -new -key eric.key  -out eric.csr  -subj '/CN=eric/O=tech'
#
# ls
apiserver-etcd-client.crt     eric.csr                proxy-client-ca.key
apiserver-etcd-client.key     eric.key                proxy-client.crt
apiserver-kubelet-client.crt  etcd                    proxy-client.key
apiserver-kubelet-client.key  front-proxy-ca.crt      sa.key
apiserver.crt                 front-proxy-ca.key      sa.pub
apiserver.key                 front-proxy-client.crt  vimal.crt
ca.crt                        front-proxy-client.key  vimal.csr
ca.key                        proxy-client-ca.crt     vimal.key
```

        - Using CA for signing CSR

```
# openssl  x509 -req  -in  eric.csr   -out eric.crt  -CAkey ca.key  -CA ca.crt
   -CAcreateserial
Signature ok
subject=CN = eric, O = tech
Getting CA Private Key
```

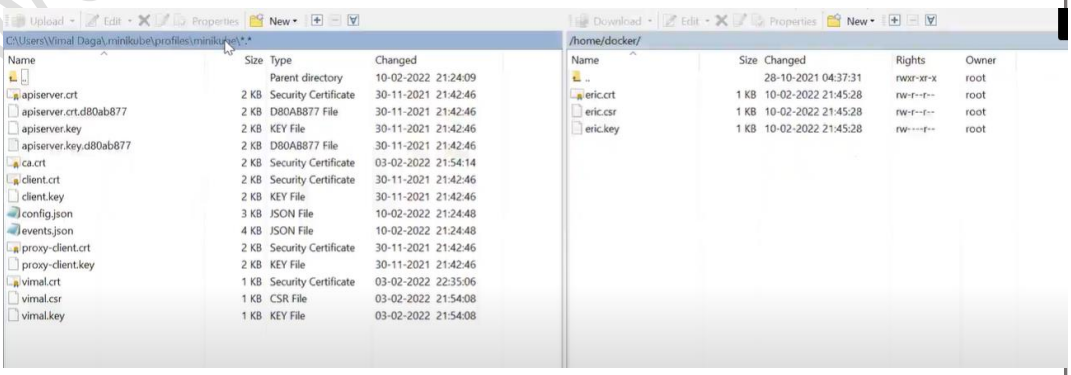
    - Creating role
        - Creating namespace

```
C:\Users\Vimal Daga>kubectl create ns testing
namespace/testing created

C:\Users\Vimal Daga>kubectl get ns
NAME              STATUS   AGE
default           Active   72d
kube-node-lease   Active   72d
kube-public       Active   72d
kube-system       Active   72d
teama             Active   22d
testing           Active   1s
```

- Changing file permission & transferring certificate to user

```
$ whoami
docker
$ pwd
/home/docker
$ sudo su - root
# cd /var/lib/minikube/certs/
# cp eric.* /home/docker/
# cd /home/docker/
# ls
eric.crt  eric.csr  eric.key
# ls -l
total 12
-rw-r--r-- 1 root root 826 Feb 10 16:15 eric.crt
-rw-r--r-- 1 root root 550 Feb 10 16:15 eric.csr
-rw-------- 1 root root 887 Feb 10 16:15 eric.key
# chmod o+r eric.key
# ls -l
total 12
-rw-r--r-- 1 root root 826 Feb 10 16:15 eric.crt
-rw-r--r-- 1 root root 550 Feb 10 16:15 eric.csr
-rw----r-- 1 root root 887 Feb 10 16:15 eric.key
#
```



- Setting credential

```
C:\Users\Vimal Daga>
C:\Users\Vimal Daga>kubectl config set-credentials eric --client-key="C:\Users\Vimal Daga\.minikube\profil
inikube\eric.key"  --client-certificate="C:\Users\Vimal Daga\.minikube\profiles\minikube\eric.crt"
User "eric" set.
```

- Manifest file for context

```
apiVersion: rbac.authorization.k8s
kind: Role
metadata:
  namespace: testing
  name: mymonitor-role
rules:
  - apiGroups: [""]
    resources: [ "pods", "deploy"]
    verbs: [ "list", "create"]
```

- Switching context

```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl config use-context minikube
Switched to context "minikube".
```

- Creating role

```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl apply -f myrole.yml
role.rbac.authorization.k8s.io/mymonitor-role created

C:\Users\Vimal Daga\Documents\Container2021-ws>
```

- Creating role binding
  - Manifest file for role binding

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: eric-user-montoring-binding
  namespace: testing
subjects:
  - kind: User
    name: eric
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: mymonitor-role
  apiGroup: rbac.authorization.k8s.io
```

  - Creating role binding

```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl apply -f myrolebinding.yml
rolebinding.rbac.authorization.k8s.io/eric-user-montoring-binding created

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get role
No resources found in default namespace.

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get rolebinding  -n testing
NAME                          ROLE                  AGE
eric-user-montoring-binding   Role/mymonitor-role   37s

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl describe  rolebinding  -n testing
Name:         eric-user-montoring-binding
Labels:       <none>
Annotations:  <none>
Role:
  Kind:  Role
  Name:  mymonitor-role
Subjects:
  Kind  Name  Namespace
  ----  ----  ---------
  User  eric

C:\Users\Vimal Daga\Documents\Container2021-ws>
```

- Applying role

```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl apply -f myrole.yml
role.rbac.authorization.k8s.io/mymonitor-role configured

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl describe role -n testing
Name:         mymonitor-role
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources  Non-Resource URLs  Resource Names  Verbs
  ---------  -----------------  --------------  -----
  deploy     []                 []              [list create]
  pods       []                 []              [list create]
```

- Checking

```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl  get  pods    -n testing
NAME   READY   STATUS    RESTARTS   AGE
myp    1/1     Running   0          16s

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl  get  pods    -n testing   -w
NAME   READY   STATUS    RESTARTS   AGE
myp    1/1     Running   0          24s
Error from server (Forbidden): unknown (get pods)
```