Create user In k8s

1. Generate certificates for the user.

```
root@FARDIN:/mnt/f/k8s_security# openssl genpkey -out fardin.key -algorithm Ed25519
root@FARDIN:/mnt/f/k8s_security# cat fardin.key
-----BEGIN PRIVATE KEY-----
MC4CAQAwBQYDK2VwBCIEIMKzGLPYp18YoQReTGCsqlRwiU/zMjbsTSxBQvCAJFaU
-----END PRIVATE KEY-----

root@FARDIN:/mnt/f/k8s_security# openssl req -new -key fardin.key -out fardin.csr -subj "/CN=fardin,/O=edit"
root@FARDIN:/mnt/f/k8s_security# ll
total 0
drwxrwxrwx 1 root root 512 Feb 29 11:32 ./
drwxrwxrwx 1 root root 512 Feb 29 11:32 fardin.csr*
-rwxrwxrwx 1 root root 294 Feb 29 11:32 fardin.csr*
-rwxrwxrwx 1 root root 119 Feb 29 11:12 fardin.key*
```

2. Create a certificate signing request (CSR).

root@FARDIN:/mnt/f/k8s_security# cat fardin.csr | base64 | tr -d "\n"
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0kTULHZ01GUUNBOUF3SVRFUUIBNEdBMVVFQXd3SFptRnlaR2x1TERFTK1Bc0dBMVVFQ2d3RVpXUnBkREFxTUFVR
wpBeXRSY0FNaEFNVNHRC0V0SXnjvYrSREJ0NVJJLTNRyvjdvYNRDTnhBoGZVMCSDWVXBFaGRVb0FBd0JRWURLMIZ3CkEwRUFs0HA0Y0haRE9zenV1UutIQUdTdHR2N1RkR1JDSm
ZUWU8yWFUzbVYxVC8yOENENjZKRmlmZmZ4WjdSYm0KdEE3cHlqSituZFlZQjBQTVAvQ312RGlKREE9PQotLS0tLVORCBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0Kroot@FA

```
root@FARDIN:/mnt/f/k8s_security# vim user_cert.yaml
root@FARDIN:/mnt/f/k8s_security# ll
total 1
drwxrwxrwx 1 root root 512 Feb 29 11:36 //
drwxrwxrwx 1 root root 512 Feb 29 11:12 ../
-rwxrwxrwx 1 root root 294 Feb 29 11:32 fardin.csr*
-rwxrwxrwx 1 root root 119 Feb 29 11:12 fardin.key*
-rwxrwxrwx 1 root root 616 Feb 29 11:36 user_cert.yaml*
```

```
apiVersion: certificates.k8s.io/v1
kind: CertificatesigningRequest
metadata:
name: fardin
spec:
request: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0KTULHZ01GUUNBUUF3SVRFUU1BNEdBMVVFQXd3SFptRnlaR2x1TERFTk1Bc0dBMVVFQ2d3RVpXUn
BKREFxTUPVRwpBexRsv0FNaEFNVnR4c0V6SXhjVXF5REJ0NVJTblRyYjdvVWRDTnhBbcZVMC9mVXBFaCRvb0FBddJRWURLM1Z3CkEwRUFs0HA0V0haRE9zenV1UUtTQUdTdHR
2N1RkR1JDSmZUWUByWFUzbVYxVC8y0ENENjZKRmlmZmZ4WjdSYm0KdEE3cHlqSituZFlZQjBQTVAvQ312RGlKREE9PQotLS0tLUVORCBDRVJUSUZJQ0FURSBSRVFVRVNULS0t
LS0K
signerName: kubernetes.io/kube-apiserver-client
expirationSeconds: 86400 # one day
usages:
- client auth
user_cert.yaml (END)
```

3. Sign the certificate using the cluster certificate authority.

```
root@FARDIN:/mnt/f/k8s_security# kaf user_cert.yaml
certificatesigningrequest.certificates.k8s.io/fardin created
root@FARDIN:/mnt/f/k8s_security# kubectl certificate approve fardin
certificatesigningrequest.certificates.k8s.io/fardin approved
```

```
rootgFARDIN:/mmt/f/k8s_security# kubectl describe csr/fardin
Name: fardin
Labels: <none>
Annotations: kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"certificates.k8s.io/v1", "kind":"CertificateSigningReque
st","metadata":{"annotations:"{}}, "name":"fardin"}, "spec":{"expirationSeconds":86400, "request":"LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRV
NULS0tLS0kTUHZ0IGUUNBUUF3SVRFUPUUBNEdBMVFQXd3SFptRhlaR2x1TERFTk1Bc0dBMVVFQ2d3RVpXUnBkREFXTUFVRwpBexR3v9FnhAEFNVnR4c0v6SXhjVXFSREJ0NVJ
TblRyYjdvVwRDTnhBbGZVMC9mVXBFaGRvb0FBd0JRWURLMT23CkEWBUFS0HA0*V0ABRE92enV1UUTQUdTdHR2NIRRAIJDSMZUWUByWFUzbVyxVcSyDENDIZKRAmlamZmZWUSV
Ym0KdEE3cHlqSituZFIZQjBQTVAvQ31ZRClKREE9PQotLS0tLUVORCBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0K", "signerName":"kubernetes.io/kube-apiserver-
client", "usages": ["client auth"]}}
CreationTimestamp: Thu, 29 Feb 2024 11:37:55 +0530
Requesting User: kubernetes-andmin
Signer: kubernetes-andmin
Signer: kubernetes.io/kube-apiserver-client
Status: Approved, Issued
Subject:

Common Name: fardin,
Serial Number:
Organization: edit
Events: <none>
```

4. Create a configuration specific to the user.

```
root@FARDIN:/mmt/f/k8s_security# kubectl get csr/fardin -o jsonpath="{.status.certificate}" | base64 -d > fardin.crt
root@FARDIN:/mmt/f/k8s_security# ll
total 5
drwxrwxrwx 1 root root 512 Feb 29 11:39 //
drwxrwxrwx 1 root root 512 Feb 29 11:12 ./
--wxrwxrwx 1 root root 512 Feb 29 11:139 fardin.crt*
--wxrwxrwx 1 root root 769 Feb 29 11:39 fardin.cs*
--wxrwxrwx 1 root root 294 Feb 29 11:32 fardin.ksy*
--wxrwxrwx 1 root root 616 Feb 29 11:23 fardin.ksy*
-rwxrwxrwx 1 root root 616 Feb 29 11:32 fardin.ksy*
-rwxrwxrwx 1 root root 512 Feb 29 11:40 ./
drwxrwxrwx 1 root root 512 Feb 29 11:40 ./
drwxrwxrwx 1 root root 512 Feb 29 11:40 ./
drwxrwxrwx 1 root root 512 Feb 29 11:12 ./
--wxrwxrwx 1 root root 512 Feb 29 11:39 fardin.crt*
--wxrwxrwx 1 root root 514 Feb 29 11:39 fardin.crt*
--wxrwxrwx 1 root root 5769 Feb 29 11:39 fardin.crt*
--wxrwxrwx 1 root root 5769 Feb 29 11:39 fardin.crt*
--wxrwxrwx 1 root root 519 Feb 29 11:32 fardin.csy*
--rwxrwxrwx 1 root root 519 Feb 29 11:32 fardin.csy*
--rwxrwxrwx 1 root root 519 Feb 29 11:36 user_cert.yaml*

root@FARDIN:/mmt/f/k8s_security# kubectl --kubeconfig fardin-kube-config config set-credentials fardin --client-key fardin.key --client-certificate fardin.csr --embed-certs=true
User "fardin" set.
root@FARDIN:/mmt/f/k8s_security# kubectl --kubeconfig fardin-kube-config config set-context fardin --client-key fardin.key --cot@FARDIN:/mmt/f/k8s_security# root@FARDIN:/mmt/f/k8s_security# root@FARDIN:/mt/f/k8s_security# root@FARDIN:/mt/f/k8s_security# root@FARDIN:/mt
```

5. Add RBAC rules for the user or their group.

Create a Role for John to manage pods in the default namespace:

```
root@FARDIN:/mnt/f/k8s_security# kubectl create role pod-manager --verb=create,list,get --resource=pods --namespace=default role.rbac.authorization.k8s.io/pod-manager created

root@FARDIN:/mnt/f/k8s_security# kubectl create clusterrole pod-manager --verb=get,list,watch --resource=pods --namespace=default clusterrole.rbac.authorization.k8s.io/pod-manager created root@FARDIN:/mnt/f/k8s_security# kubectl auth can-i get pods --namespace=default --as=fardin yes

root@FARDIN:/mnt/f/k8s_security# kubectl auth can-i get pods --namespace=default --as=fardin yes

root@FARDIN:/# kg

unable to connect to the server: dial tcp: lookup 0412A72FEB560AB29394EEAB1C4CA9E2.gr7.us-east-1.eks.amazonaws.com on 172.31.240.1:53
: no such host
root@FARDIN:~# |
```

```
root@FARDIN:/mnt/f/k8s_security# realpath fardin-kube-config
/mnt/f/k8s_security/fardin-kube-config
root@FARDIN:/mnt/f/k8s_security# export KUBECONFIG=/mnt/f/k8s_security/fardin-kube-config
root@FARDIN:/mnt/f/k8s_security# kgn
NAME STATUS ROLES AGE VERSION
                                                             v1.28.2
v1.28.2
                             control-plane
                                                     62m
master
                Ready
worker-0
                Ready
                                                     61m
                             <none>
worker-1
                                                             v1.28.2
                Ready
                             <none>
                                                     61m
```

```
root@FARDIN:/mnt/f/k8s_security# mkdir pod
root@FARDIN:/mnt/f/k8s_security# cd pod/
root@FARDIN:/mnt/f/k8s_security/pod# vim pod.yaml
root@FARDIN:/mnt/f/k8s_security/pod# kaf .
pod/nginx created
root@FARDIN:/mnt/f/k8s_security/pod# kgp
NAME
        READY
                STATUS
                                     RESTARTS
                                                AGE
                ContainerCreating
nginx
        0/1
                                                3s
root@FARDIN:/mnt/f/k8s_security/pod# kgp
                STATUS
                           RESTARTS
                                      AGE
NAME
        READY
                Running
nginx
        1/1
                                      5s
root@FARDIN:/mnt/f/k8s_security/pod#
```