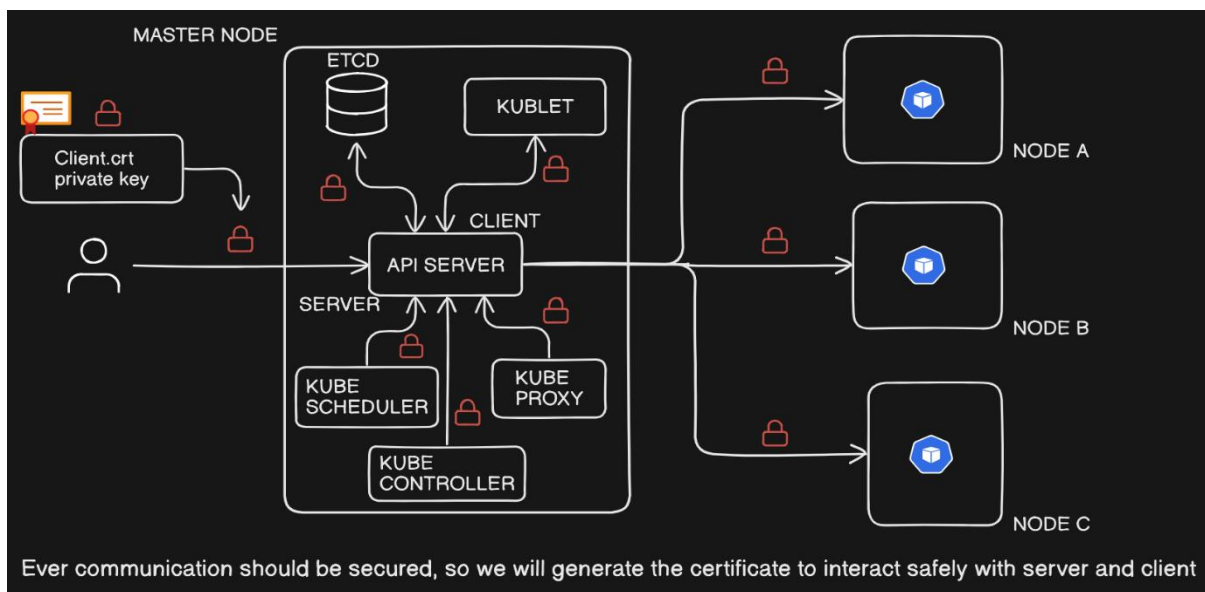


SSL AND TLS CERTIFICATE IN KUBERNETES

SSL (Secure Sockets Layer) and TLS (Transport Layer Security) certificates play a critical role in securing communications between components by encrypting data and verifying identities.

SSL and TLS Certificates in Kubernetes

- **SSL/TLS certificates** are digital certificates that provide authentication and encryption for secure data exchange over a network. They ensure the confidentiality and integrity of data transmitted between clients (like a user's web browser or service) and servers.
- In Kubernetes, TLS is primarily used for securing:
 - **API server communication:** Protects data exchanged between the Kubernetes API server and clients, such as kubectl.
 - **Intra-cluster communication:** Encrypts traffic between pods or services.
 - **Ingress traffic:** Secures external traffic entering the cluster via an Ingress Controller.



We Need SSL/TLS Certificates in Kubernetes

- **Security:** TLS certificates provide end-to-end encryption, preventing data from being intercepted by unauthorized parties.
- **Authentication:** Certificates validate the identity of clients and servers, reducing the risk of man-in-the-middle attacks.
- **Compliance:** Many regulations require data encryption, especially for sensitive or personal information.

When to use SSL/TLS Certificates in Kubernetes

- **API Communication:** Kubernetes by default secures the API server with TLS to ensure secure access for kubectl and other clients.
- **Service Communication:** When sensitive data is transmitted between services or pods, encrypting intra-cluster communication adds another security layer.
- **Ingress Traffic:** When exposing services to the internet, TLS certificates secure user access, especially when handling personal or sensitive information.

SSL AND TLS CERTIFICATE IN KUBERNETES

New employee joined inot the team I need to generate certificate to give access to the cluster.

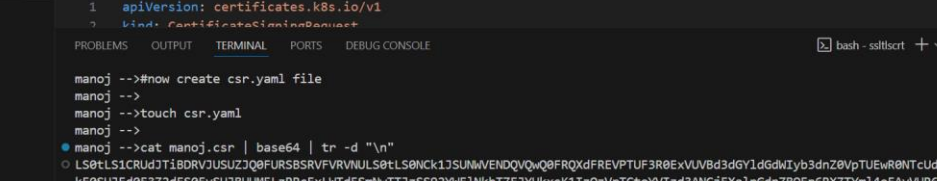
Generate 'key'

The screenshot shows a terminal window with a dark background. The terminal output is as follows:

```
manoj -->
manoj --># example:- let's say new employee joined into the team you need to generate certificate,
manoj --># to access the cluster, these are step we need to follow to generate certificate for new user
manoj -->
manoj -->openssl genrsa -out manoj.key 2048
manoj -->#i generated key, now i need to create request for certificate sign
manoj -->
manoj -->openssl req -new manoj.key -out manoj.csr -subj "//CN=manoj"
req: Extra option: "manoj.key"
req: Use -help for summary.
manoj -->openssl req -new -key manoj.key -out manoj.csr -subj "//CN=manoj"
manoj -->
```

The terminal window has a title bar that reads "bash - ssh:scrt". On the left side, there is a file explorer panel showing a directory structure. The "KUBERNETES" directory is expanded, and the "ssltlsrct" subdirectory is selected. Inside "ssltlsrct", the files "manoj.csr" and "manoj.key" are listed. The "manoj.key" file is highlighted with a green box.

Create the request for certification sign



```
EXPLORER ...
> OPEN EDITORS 2 unsaved
KUBERNETES
> autoscaling
> bash
> cluster
> configmap
> Daemonset
> healthprobes
> labels&selector
> multicontainer
> namespace
> podcreation
> requestsanlimits
> services
> sslcertificate
v sslsrt
  ! csr.yaml
  E manoj.csr
  A manoj.key
> tolerationandaffinityands...
E clusterrole.txt
$ ipaddress.sh
! role.yaml
! rolebind.yaml
```

```
sslsrt > ! csr.yaml
1 apiVersion: certificates.k8s.io/v1
2 kind: CertificateSigningRequest
manoj -->#now create csr.yaml file
manoj -->
manoj -->touch csr.yaml
manoj -->
manoj -->cat manoj.csr | base64 | tr -d "\n"
LS0tLS1CRUdJTiBDRVJUSUJzQ0FURSBSRVFVbWU50tS0NkNC1jSUNmVENDQVQwQ0FRQXZFREVPVFU3R0EXVUVB3d2Y1dGdWlYb3dnZ0VpTUEwR0NTcUdTSWlZIFFkFRVUENC
K0ESUJEd0F3Z2ZdSF0FSUJBUmU5LzRRcExLTd5SmMwTTJzSS92YmF1InkhTZE3YUksK1IzQzVrTct0YVlzd3ANCjFhXlpgdnZB0FpRkXZTm14eFAYVUR6bXVXZ2x0TjV1RHR0aT
UvT1ZhSUNQUTU4RjFhdW44ZDVuTkhXNzQvbHINCnRaS1aWjZ2ZjZjBKRjRwc2MvVFFsUmNlVGVxczQVNSVHFOl2pGanJ2OWY1mkJ0U1ETU4WlxlwRtUzEmYUemhhS5G8NCmNyZ10a1B
HfZShSFZ2V1VORHtdFhPQ1NCcE1KU2d2j1d5eE3am16aEwRW85a2hTBe14UnVmc0N0WmFETVoNCl3d3ZTnkwNE1kQndSQyrtZahH3BUEgzNXpVbXVZNG1FckV3Q2orMnY2cVZV
TtDrSm1ROW1z0Xp2MmJqVnRSUGNCn1UwpybTZXN0N1ajNIa0tFdgjMvNZ20OE5oYnFSTjArZE3J3d2ad2hB201CQFHZ0FENKsc7H3vaatp13MFTmZG5eATRN1QW0FFN1NM
I9dU180S0pwt290eVFSeGRbnV0RGtyYWFNSUFuRUK0WgdmdEdtdqWY2a3Q0bEZTaUgNcQ5d0paYj8XUePgVVL18IUmZMS
NgdTC0FRtWghpGdcNCmCUF1LT1rSVRYk0RC0G5hSmpvMSDVfphSDBje05L1Lld2zSNTN1Zw8rUDVudZ2Y2Rn1VdmNkan
vR01QVB8ZVGRSjY0t1dK1wMtducFFSeEdYUkGp52prY2ZNRDR3ZFMC1hw51VHREQxQ245ZVZ1K1v1F1AvNdc4N0hYVn
manoj -->
```

this is the user certificate
sign request, i have encode
with base64 key. now copy
this & paste in request in
"manoj.csr"

SSL AND TLS CERTIFICATE IN KUBERNETES

```

1  apiVersion: certificates.k8s.io/v1
2  kind: CertificateSigningRequest
3  metadata:
4    name: manoj
5  spec:
6    # encode certificate request
7    request: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBRSRVFRVNVULS0tLS0Nck1JSUNNVENDQVQwQ0FRQXdFREVPVUF3R0ExVUVBd3dGYldGdWIyb3dnZ0VpTUEwR0NTcUdTSWlzMkRFRF
8    signerName: kubernetes.io/kube-apiserver-client
9    expirationSeconds: 432000 # 50 days ,we can also give 1 year for validity
10   usages:
11     - client auth
12
13 #to decode the csr file use this command in powershell
14 # [Convert]::ToBase64String([System.IO.File]::ReadAllBytes("C:\Users\provide the path\filename.csr")) | % { $_.Replace("`n", "") }

```

Created certificate signing request still in pending state

```
manoj -->
manoj -->kubectl get csr
NAME    AGE    SIGNERNAME                                REQUESTOR    REQUESTEDDURATION   CONDITION
manu    30m    kubernet.es.io/kube-apiserver-client     kubernetes-admin    50d           Approved,Issued
manoj -->kubectl apply -f csr.yaml
certificatesigningrequest.certificates.k8s.io/manoj created
manoj -->
manoj -->kubectl get csr
NAME    AGE    SIGNERNAME                                REQUESTOR    REQUESTEDDURATION   CONDITION
manoj   8s    kubernet.es.io/kube-apiserver-client     kubernetes-admin    50d           Pending
manu    30m    kubernet.es.io/kube-apiserver-client     kubernetes-admin    50d           Approved,Issued
manoj -->
```

we still didn't approved yet. this certificate should be approved by authorized CA (certificate authority) this is for the internal k8's cluster

[illegible]

SSL AND TLS CERTIFICATE IN KUBERNETES

After checking very details then I need to approve the request

```
manoj -->kubectl describe csr manoj
Name:         manoj
Labels:       <none>
Annotations:  kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"certificates.k8s.io/v1","kind":"CertificateSigningRequest",
"metadata":{"annotations":{},"name":"manoj"},"spec":{"expirationSeconds":432000,"request":"LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNUbS0tL
S0NCK1JSUNWVENDQVQwQ0FRQXdfREVPTUF3R0ExVUVBd3dGYldGdWIyb3dnZ0VpTUeW0tCudTSWIZRFFQkFRVUENCkE0SUJEd0F3Z2dfS0FvSUJBUUM5LzRRcExLWtd5SmNwTT
JzSS92YWFIbnkxZ0Ykxsk1IzQzVrTctoYVZid3ANCjFkXAlpGdnZB0Fp6RXZTYm14eFAYVUR6bXVDZ2x0TjV1HRQaTUvTlZhsUNCQTU4RjFhdWd4ZDVuTkhXNzQvbHINCnRaS1h
qZ2Z2p2jBKRjRwcZMvVFFsUmNIVGxxQVNSVHFOl2pGanJZOWY1WkJoUU1ETUQ4WwXwRUTzemUyemhhSG8NCmNyZl0a1BhSFZhSFZ2V1VORHVtdFhP01NCcE1KU2dja1d5eEJ3am16
aEMuRW85a2hTbE14UnVmc0NHnFETVoNCldVa3Z0TnkwNE1kQndSQytZaHJBUEgzNXpVbXVZNG1FcVks3Q2orMnY2cVZVTDrSm1l
jNIa0tFdjMvN2Z0OE50YnFSTjArZEJ3d3Zad2hBZ01CQUFHZ0FEQU5CZ2txaGtpRz13MEINCKFRc0ZBQU9DQVFFQUNVmi9uVi80:
dmdEtqdwY2a3Q0bEZTaUgNCnQ5d0paYjBXUEpGdVVT11BIUmZMS1dzcZfQmWJCUg1FS3A4MHNIUuzsQvdVRkNSam15YUNGdTLCOI
vam5DVfPhSDBjeE05L1lWd2dSNTNiZW8rUDVUd3Y2RnV1dmNKanNoRm1UWm8NCng4N3dwcWQ2SnlwcXhxSittLeE9idGovR0lnQVl
Y2ZNRDR3ZFMNC1hwd1VHREQxQ245ZVZ1Vk1Fb1AvNdc4N0hIVnNzaEZiZDBVeFZHMfowR1FCRnk4U010VWFWQ1R0Yw01dXRSamEI
m1Y1J0b1E9PQ0KLS0tLS1FTkQgQ0VSVe1GSUNBVEUgKVRUVVTVc0tLS0tDQo=","signerName":"kubernetes.io/kube-a
h"}}}
```

```
CreationTimestamp: Wed, 30 Oct 2024 13:02:59 +0530
Requesting User:   kubernetes-admin
Signer:           kubernetes.io/kube-apiserver-client
Requested Duration: 50d
Status:           Approved,Issued
Subject:          Common Name: manoj
                  Serial Number:
Events:           <none>
```

certificate approved, valid till 50days. after the we can renew it. if we want

```
manoj -->
manoj -->kubectl get csr
NAME      AGE      SIGNERNAME              REQUESTOR    REQUESTEDDURATION   CONDITION
manoj     7m55s    kubernetes.io/kube-apiserver-client  kubernetes-admin  50d                 Pending
manu      38m      kubernetes.io/kube-apiserver-client  kubernetes-admin  50d                 Approved,Issued
```

```
manoj -->#let approve the certificate
manoj -->
manoj --># if everything is good you can approve the request
manoj -->
manoj -->kubectl certificate approve manoj
certificatesigningrequest.certificates.k8s.io/manoj approved
```

after checking everything is good then only approve the certificate

```
manoj -->kubectl get csr
NAME      AGE      SIGNERNAME              REQUESTOR    REQUESTEDDURATION   CONDITION
manoj     8m54s    kubernetes.io/kube-apiserver-client  kubernetes-admin  50d                 Approved,Issued
manu      39m      kubernetes.io/kube-apiserver-client  kubernetes-admin  50d                 Approved,Issued
```

```
manoj -->
manoj -->#now i want to share this certificate with user
manoj -->
manoj -->kubectl get csr manoj -o yaml > issusecrt.yaml
manoj -->
manoj -->
```

SSL AND TLS CERTIFICATE IN KUBERNETES

```

manoj -->
manoj -->#in this file "request" is in encode formatted we need to decode
manoj -->#copy the request from file
manoj -->
manoj --># echo "<past-request>" | base64 -d --> now decoded
manoj -->
manoj --># echo "LS0tLS1CRUdJTiBDRVJUSU5JZjQ0FURS5SRVFRVnVUS0tLS0kNk1JSUwVWENDQWQ0RFRXQzREVPUTF3R0E5VUVBZDZ3dGY1dGdwIyB3dnZ0VtVUEwR0NTcUdTSWIzRFFkQFRVUENCKE0S0JEdG93
Z2ZfS0F0VUJBUU5MLzRRcXkWTdS5mWUk3ZjTS5S92YmF1NkhTE3ZjYUxksK1IzQzvrTctoVYIzd3ANCjFxaJp6dnZB0FP6RkZTm14eFayVUR6bXVDZ2x0TjV1RHRQaTUvT1ZhSUNCQTU4RjFhdmd4ZDVuTklXNzQvbnHI
NCnR8a1hQz22ZpJgRkRwczmVVF5UmNlVgxxQNSVHF0L2p6ganJZ0MY1Wk3JoUUEtUQ4WkxwRUtzmYjemhHSG8NCmNyZl0ea1BhSFZhfZ2V1VORHVtJdFhPQ1NCcE1KUzDja1c
Umc0MWhnFETVoNc1dva3ZQTKmkWElK1Qc8dS0yTzAh7BUeJgznXpYbVZNGN1FcVk3Q2orMnY2cVZVTDrSm1ROW1i20Xp2Wm3QvNrSUGGhNCn1Nv4pYbTZ2XN0NLaJNiAetFdjMvNZ0OE
UfHzaBEQESCZ22cagTgPr213MEINCKFRcdZB0U9DQVFFQUNVMI9Uv1850p6wT298eVSeGRnbnVR0GTyYwNF5UFURUK0W6GmdEtqdwY2a3Q0bEzTAUghNCQ5d8paYjBXUEPg6dVVTL
MhM1Uuz2oqvdVRKNSam15YUNGdJ1C0FRFTW8hPdGcNCnCBMUF1T1zrSVRYK0RC0G5hSmpvams5DVfphSD8jEe0S11Wd2dSNTiI2W8rUDUVDj3Z2RnV1dmlKtaNoRm1Uw8MNCng4N3dv
nQ0B8ZUWVG8RSy0U1dL1wTducFSeEdYUkGv2PrY2ZNRDR3ZFwNc1hwd1VHRE0xQ245ZVZ1K1V1AvaIDc4N0hTvrN8aEzIZD8VeFZhHfowR1FCrnk4u0i0VwFwQ1R0Yw0idX6
MwGgr1BBSVh6m11Y30bLE9Q0KLS0tLS1FtKQ0vSV5E1GSUNBVEUkUgVRVUVTCv0tLS0tDQo=" | base64 -d
manoj -->
0 -----BEGIN CERTIFICATE REQUEST-----
MIICVTCATQCAQAwEDEOMAwGAUUEAwFbWub2owggEiMABGSCGSGIsB3DQEBAQUA
A4IDBwAwggEiAQAIAQAQ9/4QpLYk7yJcPmZsI/vaae6HSDbXRL1+R3C5Kl+thR3wp
1WjZfVvA8ZsEvSbixXP2UDzmuCk1Tn5uDtP15/NVCAIS5F1augxd5nNw74/1r
tZKXJgfiF0JF4p3/TQlRcHt1qASRTqh/fJf9rYf5ZB0QIDM0B1pEkSze2zhaHo
cnFz4kPaHvAhVwVUNDuStXOCS8pM3GikWyxBwjzCh0E9khS1MxRufSxCGZqDMZ
WUkVpNy04tDbWRc+YhrAPH35zUmuY4mEqY7Cj+2v6qVUM7Kj1Q9ms9zvZbJvRtPh
yMUjXm6W7CKj3HkKvE3/7fN8NhbqyN0+BdWwvZJwAgMBAAGGAADANBgkqhkiG9w0B
AQoFAAOCQAQEAU2/nV/4QjP0tyQRXdgndtKraaMIAnEi40gFtkJuf6kt41FSiH
t9wZ2b0WPJFuUs/PHRfLJl
pB1AKOVkITr+O88na3JjoJj
x87ypad6Jypqpx4Kx0bL:
XpuGDD1C9eVwVUEoP/4:
UdbCnQ6Y51H8khtSMtk+/I
-----END CERTIFICATE REQUEST-----
manoj -->

```

Now I can add this certificate to kube-config and assign certain roles to it. So that user will have certain permissions and user can access the server using that certificate.

```
manoj -->
manoj --># now i can add this certificate to kube-config and assign certain roles to it. so that user will have certain permissions and
manoj --># user can access the server using that certification
manoj -->
manoj -->kubectll get csr -o yaml > manoj.crt
manoj -->
manoj -->
```

SSL AND TLS CERTIFICATE IN KUBERNETES

```
sslscrt > manoj.crt
1  apiVersion: v1
2  items:
3  - apiVersion: certificates.k8s.io/v1
4    kind: CertificateSigningRequest
5    metadata:
6      annotations:
7        kubectl.kubernetes.io/last-applied-configuration: |
8          {"apiVersion":"certificates.k8s.io/v1","kind":"CertificateSigningRequest","metadata":{"annotations":{},"name":"manoj"},"spec":{"expirationSeconds":"4320000","request":
9      creationTimestamp: "2024-10-30T07:32:59Z"
10     name: manoj
11     resourceVersion: "105207"
12     uid: 82a6934e-fb13-4a03-b231-317f46e0e54f
13   spec:
14     expirationSeconds: 4320000
15     groups:
16     - kubeadm:cluster-admins
17     - system:authenticated
18     request: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBRSRVFRVJUNLS0tLS0Nkck1JSUMwVENDQWQwQ0F0RQXDFREVVTUF3R0EXVUVBd3dGdWJyb3dnZ0VpTUewR0NTcudTSWZFRFFQKFRVUENCKE0SUJEd0F3Z2dFS0FvS
19     signerName: kubernetes.io/kube-apiserver-client
20     usages:
21     - client auth
22     username: kubernetes-admin
23   status:
24     certificate: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tck1JSUM5akNDQM2Z0F3SUJBZ0tSQUtja3E2emdNTmZSHHVB0V1lUw5xbU13RFFZSkktvklodmNOQVFfTEJRQXCKRlRFE1CRUDBMVVFQXhNS2Z2ZVmlaW
25     conditions:
26     - lastTransitionTime: "2024-10-30T07:41:48Z"
27     - lastUpdateTime: "2024-10-30T07:41:48Z"
28     message: This CSR was approved by kubectl certificate approve.
29     reason: KubectlApprove
30     status: "True"
```

Comparison between SSL and TLS in table format:

Feature	SSL (Secure Sockets Layer)	TLS (Transport Layer Security)
Security	Vulnerable to various attacks, weaker encryption	Stronger encryption, more secure, supports PFS
Handshake Process	Slower and more complex	Faster, simplified handshake, especially in TLS 1.3
Cipher Suites	Limited, lacks modern algorithms	Expanded support for secure, modern cipher suites
Deprecation Status	Deprecated and unsupported	Current standard for secure communications
Usage in Certificates	Often referred to as "SSL certificates" due to legacy naming	Technically uses TLS encryption, even in "SSL certificates"