



Lab: Deploy Metrics Server with Custom TLS SANs

Goal

Deploy a working `metrics-server` instance with a **valid certificate** including the SANs needed to talk to the kubelets securely (`--kubelet-preferred-address-types=InternalIP,Hostname`).

Prerequisites


- Kubernetes cluster (Minikube, Kind, etc.)
 - `kubectl`, `openssl`
 - Cluster admin permissions
-

Lab Steps Overview

1. Generate a TLS certificate with proper SANs
 2. Patch or install `metrics-server` using the new certificate
 3. Test metrics collection with `kubectl top`
-

Generate Certificate with Required SANs

```
mkdir -p metrics-server-cert && cd metrics-server-cert
```

 `csr.conf`

```
[req]
req_extensions = v3_req
distinguished_name = req_distinguished_name
prompt = no

[req_distinguished_name]
CN = metrics-server

[v3_req]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names

[alt_names]
DNS.1 = metrics-server
DNS.2 = metrics-server.kube-system
DNS.3 = metrics-server.kube-system.svc
```

Generate cert:

```
openssl genrsa -out metrics-server.key 2048

openssl req -new -key metrics-server.key -out metrics-server.csr -config csr.conf

openssl x509 -req -in metrics-server.csr \
    -signkey metrics-server.key \
    -out metrics-server.crt -days 365 \
    -extensions v3_req -extfile csr.conf
```

You now have:

- `metrics-server.crt` (signed cert)
- `metrics-server.key` (private key)

Deploy Metrics Server with Custom TLS

Download and edit the deployment YAML:

```
kubectl apply -f https://github.com/kubernetes-sigs/metrics-
server/releases/latest/download/components.yaml --dry-run=client -o yaml >
metrics-server.yaml
```

Modify `metrics-server.yaml`

- Add volume mounts for the certificate
- Add `--tls-cert-file` and `--tls-private-key-file` flags
- Optionally: `--kubelet-insecure-tls=false` if SANs are correct

🔗 Example patch to Deployment:

```
spec:
  containers:
    - name: metrics-server
      image: ...
      args:
        - --cert-dir=/certs
        - --secure-port=4443
        - --kubelet-preferred-address-types=InternalIP,Hostname
        - --tls-cert-file=/certs/metrics-server.crt
        - --tls-private-key-file=/certs/metrics-server.key
      volumeMounts:
        - name: certs
          mountPath: /certs
          readOnly: true
  volumes:
    - name: certs
      secret:
        secretName: metrics-server-certs
```

Create the Certificate as a Secret

```
kubectl create secret generic metrics-server-certs \
  --from-file=metrics-server.crt=metrics-server.crt \
  --from-file=metrics-server.key=metrics-server.key \
  -n kube-system
```

Apply the Modified Deployment

```
kubectl apply -f metrics-server.yaml
```

Verify

Wait for the pod to be ready:

```
kubectl get pods -n kube-system | grep metrics-server
```

Then check metrics:

```
kubectl top nodes  
kubectl top pods --all-namespaces
```

If you still get TLS or SAN errors, double-check the cert SAN entries match the service DNS.

Optional Cleanup

```
kubectl delete -f metrics-server.yaml  
kubectl delete secret metrics-server-certs -n kube-system
```

 August 22, 2025