**Hướng dẫn renew certification elk 8 sử dụng cert-manager**

**Lộ trình đổi cert an toàn (không downtime)**

1. Tạo Certification mới (**bước 03** và **bước 04**)
2. Rollout elasticsearch (**bước 05**)
3. Rollout kibana (**bước 05**)
4. Kiểm tra lại dịch vụ

**Bước 01: Chuẩn bị & lưu ý**

- **Kiểm tra xem** đã cài hay chưa **cert-manager**   
PC1150:02-anh.pt8$ **kubectl -n cert-manager get pods**

# Tất cả phải Running: cert-manager, cainjector, webhook → **nếu chưa cài ta sẽ cài mới.**

- **Secret hiện tại cho ES:** **elasticsearch-master-certs**. Ta sẽ **giữ nguyên tên secret này** để không phải đổi values Helm.

- **Subject Alternative Name (SAN)** bắt buộc cho **Elasticsearch HTTP** (để Kibana/clients không báo sai hostname):

elasticsearch-master

elasticsearch-master.elk-stack-08

elasticsearch-master.elk-stack-08.svc

elasticsearch-master.elk-stack-08.svc.cluster.local

**Bước 02: Cài cert-manager (Helm)**

#### ==> Tạo ns cho cert-manager

PC1150:02-anh.pt8$ **kubectl create namespace cert-manager**

#### Cài CRDs (bắt buộc)

PC1150:02-anh.pt8$ **kubectl apply -f https://github.com/cert-manager/cert-manager/releases/download/v1.14.4/cert-manager.crds.yaml**

#### Add repo & install (chọn v1.14.x ổn định)

PC1150:02-anh.pt8$ **helm repo add jetstack https://charts.jetstack.io**

PC1150:02-anh.pt8$ **helm repo update**

PC1150:02-anh.pt8$ **helm upgrade --install cert-manager jetstack/cert-manager \**

**--namespace cert-manager \**

**--version v1.14.4**

#### Kiểm tra:

PC1150:02-anh.pt8$ **kubectl -n cert-manager get pods**

# Tất cả phải Running: cert-manager, cainjector, webhook

**Bước 03:** Tạo **Root CA nội bộ** và **ClusterIssuer** cấp phát: Cách này phù hợp cho TLS nội bộ giữa Kibana ↔ ES. Nếu bạn cần **public CA (Let's Encrypt)** cho Ingress, xem Bước 05 thêm.

#### Tạo file issuer-ca.yaml và áp dụng:

PC1150:02-anh.pt8$ **nano issuer-ca.yaml**

# 2.1 Bootstrap 1 self-signed ClusterIssuer để sinh Root CA

apiVersion: cert-manager.io/v1

kind: ClusterIssuer

metadata:

name: selfsigned-bootstrap

spec:

selfSigned: {}

---

# 2.2 Tạo Certificate là Root CA và lưu vào secret elk-root-ca (10 năm)

apiVersion: cert-manager.io/v1

kind: Certificate

metadata:

name: elk-root-ca

namespace: cert-manager

spec:

isCA: true

commonName: elk-root-ca

secretName: elk-root-ca

duration: 87600h # ~10 năm

issuerRef:

name: selfsigned-bootstrap

kind: ClusterIssuer

---

# 2.3 Dùng Root CA trên làm ClusterIssuer chính cho toàn cụm

apiVersion: cert-manager.io/v1

kind: ClusterIssuer

metadata:

name: elk-ca-issuer

spec:

ca:

secretName: elk-root-ca

#### Áp dụng:

PC1150:02-anh.pt8$ **kubectl apply -f issuer-ca.yaml**

#### Kiểm tra CA sẵn sàng:

PC1150:02-anh.pt8$ **kubectl -n cert-manager get secret elk-root-ca**

**Bước 04:** Cấp cert cho **Elasticsearch HTTP** (secret: elasticsearch-master-certs)

#### Tạo es-certificate.yaml

PC1150:02-anh.pt8$ **nano es-certificate.yaml**

apiVersion: cert-manager.io/v1

kind: Certificate

metadata:

name: elasticsearch-master-certs

namespace: elk-stack-08

spec:

secretName: elasticsearch-master-certs # giữ nguyên tên secret đang dùng

commonName: elasticsearch-master

dnsNames:

- elasticsearch-master

- elasticsearch-master.elk-stack-08

- elasticsearch-master.elk-stack-08.svc

- elasticsearch-master.elk-stack-08.svc.cluster.local

duration: 8760h # 1 năm

renewBefore: 720h # gia hạn trước 30 ngày

issuerRef:

name: elk-ca-issuer

kind: ClusterIssuer

**#Áp dụng & đợi cert sẵn sàng:**

PC1150:02-anh.pt8$ **kubectl apply -f es-certificate.yaml**

PC1150:02-anh.pt8$ **kubectl -n elk-stack-08 wait --for=condition=Ready certificate/elasticsearch-master-certs --timeout=120s**

PC1150:02-anh.pt8$ **kubectl -n elk-stack-08 get secret elasticsearch-master-certs**

**Ghi Chú:** cert-manager sẽ tự tạo/ghi đè secret **elasticsearch-master-certs** với tls.crt, tls.key, kèm ca.crt (Root CA).

**Nếu muốn ép cert-manager cấp lại ngay thực hiện lệnh sau:**

PC1150:02-anh.pt8$ **kubectl cert-manager renew -n elk-stack-08 elasticsearch-master-certs**

# theo dõi sự kiện

PC1150:02-anh.pt8$ **kubectl -n elk-stack-08 describe certificate elasticsearch-master-certs**

Sau khi ép cấp lại cert thực hiện tiếp theo bước 5 để nạp cert mới

**Bước 05: Rollout ELK để nạp cert mới & kiểm tra**

Làm theo các bước sau để tìm đúng resource rồi restart:

1) Tìm tên thực tế của Kibana

PC1150:02-anh.pt8$ **kubectl -n elk-stack-08 get deploy,sts | grep -i elasticsearch**

PC1150:02-anh.pt8$ **kubectl -n elk-stack-08 get deploy,sts | grep -i kibana**

Bạn sẽ thấy một trong các tên kiểu: **deployment.apps/kibana-kibana** hoặc **deployment.apps/kibana-01-kiban**a hoặc **statefulset.apps/kibana**

**2) Restart đúng resource tìm được:**

Nếu là **Deployment**:

PC1150:02-anh.pt8$ **kubectl -n elk-stack-08 rollout restart deploy/<tên-deployment>**

Nếu là **StatefulSet**:

PC1150:02-anh.pt8$ **kubectl -n elk-stack-08 rollout restart sts/<tên-statefulset>**

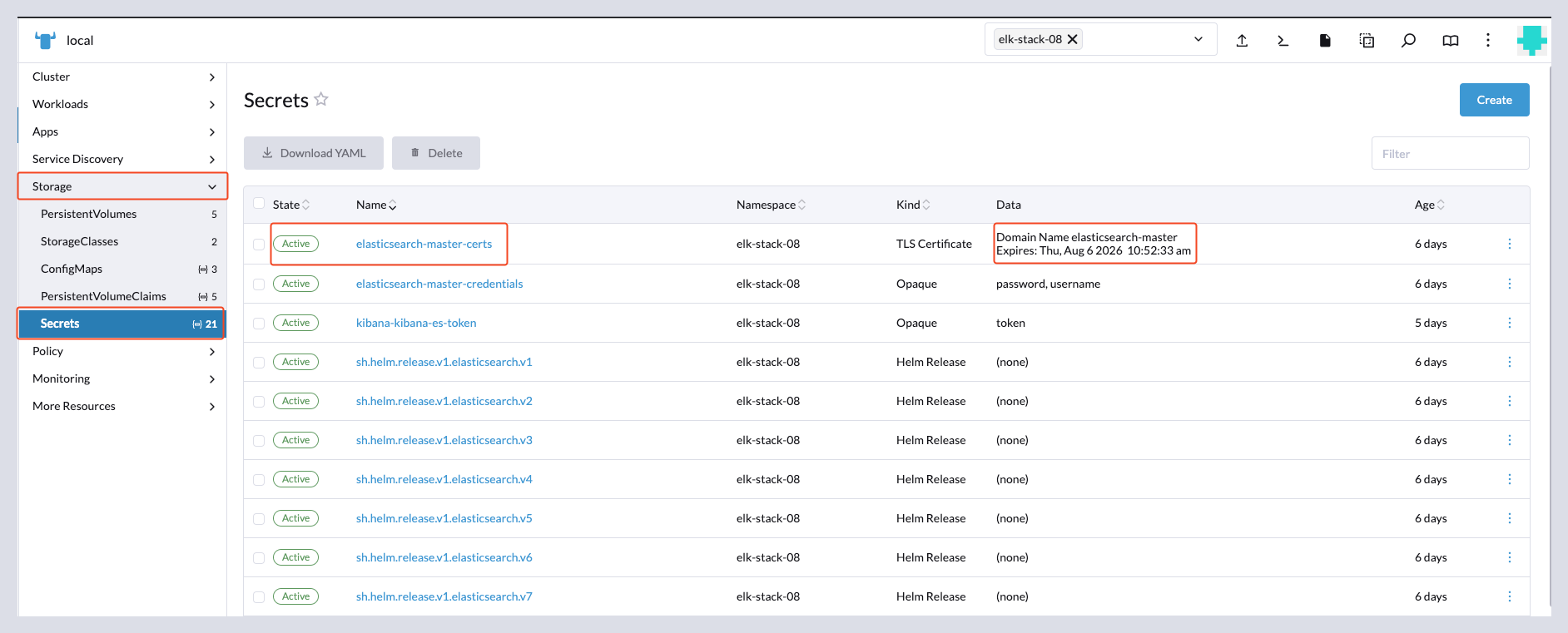
**Ghi Chú:**   
 - sts chính là statefulset

- Nếu Kibana đã được cấu hình trong file values của kibana như ở dưới:

elasticsearchCertificateSecret: elasticsearch-master-certs

elasticsearchCertificateAuthoritiesFile: ca.crt

thì Kibana sẽ nhận CA mới ngay. Nếu trước đây bạn từng dùng verificationMode: none/certificate, có thể bỏ cấu hình đó để bật verify đầy đủ

### Xem ngày hết hạn: có thể dùng rancher hoặc command line  
 Xem bằng rancher  
  
  
  
 Xem bằng command line

PC1150:02-anh.pt8$ kubectl -n elk-stack-08 get secret elasticsearch-master-certs \

-o jsonpath='{.data.tls\.crt}' | base64 -d | \

openssl x509 -noout -startdate -enddate -subject -issuer

**Bước 6:** (Tuỳ chọn) Cert public cho Kibana Ingress (Let's Encrypt)

Nếu bạn có Ingress NGINX và muốn public CA cho người dùng truy cập:

Tạo ClusterIssuer Let’s Encrypt (HTTP-01):  
PC1150:02-anh.pt8$ nano le-clusterissuer.yaml

apiVersion: cert-manager.io/v1

kind: ClusterIssuer

metadata:

name: letsencrypt-prod

spec:

acme:

email: your-email@citigo.com.vn

server: https://acme-v02.api.letsencrypt.org/directory

privateKeySecretRef:

name: letsencrypt-prod-key

solvers:

- http01:

ingress:

class: nginx

## Apply cấu hình

PC1150:02-anh.pt8$ kubectl apply -f le-clusterissuer.yaml

**Tạo Certificate cho Kibana** (ví dụ host kibana8-anhpt8.citigo.com.vn):

PC1150:02-anh.pt8$ nano kibana-certificate.yaml

apiVersion: cert-manager.io/v1

kind: Certificate

metadata:

name: kibana-tls

namespace: elk-stack-08

spec:

secretName: kibana-tls

issuerRef:

name: letsencrypt-prod

kind: ClusterIssuer

dnsNames:

- kibana8-anhpt8.citigo.com.vn

## Apply cấu hình

PC1150:02-anh.pt8$ kubectl apply -f kibana-certificate.yaml

PC1150:02-anh.pt8$ kubectl -n elk-stack-08 wait --for=condition=Ready certificate/kibana-tls --timeout=180s

**Ghi Chú:** Đảm bảo **Ingress Kibana** dùng **tls.secretName: kibana-tls**. Không cần đụng tới cert của ES nội bộ