**Linh tinh**

**Kiểm tra Secret nexus-tls**

kubectl get secret nexus-tls -n nexus -o yaml

→ Phải thấy:

type: kubernetes.io/tls

data:

tls.crt: <nội dung base64>

tls.key: <nội dung base64>

Nếu không có hoặc trống → là do cert-manager chưa xin được cert.

Xem chi tiết certificate đã cấp:

kubectl describe certificate -n nexus nexus-tls

**Xem logs cert-manager**

kubectl logs -n cert-manager deploy/cert-manager -f

**Cách vào pod cert-manager kiểm tra:**

kubectl run curl-test --rm -i --tty --image=alpine --restart=Never – sh

**Bên trong shell của pod, cài curl:**

apk add --no-cache curl

**Sau đó test truy cập ACME server:**

curl -v <https://acme-v02.api.letsencrypt.org/directory>

Xem DNS config trong pod:

cat /etc/resolv.conf

Kiểm tra log CoreDNS:

kubectl logs -n kube-system -l k8s-app=kube-dns

Kiểm tra trang thái certificate:

kubectl get certificate -A -o wide

kubectl describe certificate -A

kubectl get certificate -A

Kiểm tra cert-manager log

kubectl logs -n cert-manager -l app.kubernetes.io/name=cert-manager

**Xóa Certificate để cert-manager tạo lại từ đầu** — việc này an toàn:

kubectl delete certificate nexus-tls -n nexus

Rồi apply lại ingress:

kubectl apply -f ingress-nexus.yaml

Hoặc nếu bạn thích, xóa luôn cái secret cũ (nếu có):

kubectl delete secret nexus-tls -n nexus

kubectl run -n cert-manager curl-client --rm -it --image=alpine -- sh

cat /etc/resolv.conf

Vì DNS options như ndots được set từ kubelet khi pod được deploy (thường default từ kubelet hoặc từ deployment yaml).

Bạn có thể chạy pod tạm:

kubectl run -n cert-manager dns-test --rm -it --image=busybox:1.28 -- sh

Rồi bên trong:

cat /etc/resolv.conf

→ Giá trị này sẽ giống với pod cert-manager, vì cùng 1 namespace và kubelet config.

**Tạo forward từ windows host đến máy ảo:**

netsh interface portproxy add v4tov4 listenport=80 listenaddress=0.0.0.0 connectport=31338 connectaddress=192.168.98.141

**Xóa một forward:**

netsh interface portproxy delete v4tov4 listenport=443 listenaddress=0.0.0.0

**Xem các forward:**

netsh interface portproxy show all

***Một số lệnh kiểm tra dns:***

nslookup jenkins.jenkins.svc (trả về ip của domain)

curl -v http:// nslookup jenkins.jenkins.svc.cluster.local.:8080 (nếu không dùng full url thì cần chú ý “ndots”, “ndots” lớn hơn số dấu “.” Trong domain thì dns sẽ tự gắn thêm phần searches (ví dụ: cluster.local), gây sai domain và dns phân giải không chính xác). Để xem lại “ndots” dùng lệnh: cat /etc/resolv.conf

kiểm tra chi tiết yaml của một service, pod, … với lệnh:

kubectl get svc jenkins -n jenkins -o yaml

kubectl get svc jenkins -n jenkins -o yaml | grep noi\_dung\_tim\_kiem [-A so\_dong\_can\_xem]

kubectl get deployment jenkins -n jenkins -o yaml | grep containerPort -A 2

kubectl get svc jenkins -n jenkins -o yaml | grep selector -A 2

Xem jenkins đăng lăng nghe ở những port nào:

kubectl exec -it jenkins-bb955c969-xdm7r -n jenkins -- ss -tuln

Chạy một pod test:

kubectl run txu-test --rm -i -t --image=ubuntu:22.04 -n jenkins -- /bin/bash

Get Endpoits:

kubectl get endpoints -n jenkins jenkins -o yaml

Get Config map

kubectl -n kube-system get configmap coredns -o yaml

Kiểm tra cấu hình dns:

cat /var/lib/kubelet/config.yaml

cat /var/lib/kubelet/config.yaml | grep clusterDNS

Restart kubelet:

sudo systemctl restart kubelet

Xem nhanh một pod đang chạy ở node nào

kubectl get pod -l app=jenkins -o wide -n jenkins

Vào lại container restore-from-minio:

kubectl exec -it <jenkins-pod-name> -n jenkins -c backup-to-minio – sh

Kiểm tra alias:

export MC\_CONFIG\_DIR=/mc-config

mc alias list

Kiểm tra butket trên minio theo đường dẫn đã set trong alias (trong container restore-from-minio, trong deployment của jenkins):

mc ls minio

Nếu vẫn báo Bucket does not exist, bạn có thể tạo bucket một lần bằng tay:

mc mb minio/jenkins-data

# Thử tạo file nhỏ

echo test > /tmp/test.txt

# Thử upload thủ công

mc cp /tmp/test.txt minio/jenkins-data/test.txt

**Thử nghiệm nhanh**

Bạn cũng có thể test bằng cách:

curl -v https://minio.txuapp.com/minio/health/ready

→ Đây là một API mặc định của MinIO → nếu trả về 200 OK là đã trỏ đúng API

**Giải pháp dứt điểm: Xoá cưỡng bức namespace kong**

**Bước 1: Xuất file JSON của namespace**

kubectl get namespace kong -o json > kong-ns.json

**Bước 2: Sửa file kong-ns.json**

Tìm đoạn:

"spec": {

"finalizers": [

"kubernetes"

]

}

**Xoá toàn bộ đoạn "finalizers"** hoặc chỉnh lại thành:

"spec": {

"finalizers": []

}

**Bước 3: Áp dụng lại để xoá cưỡng bức namespace**

kubectl replace --raw "/api/v1/namespaces/kong/finalize" -f kong-ns.json

Sau vài giây, namespace kong sẽ biến mất khỏi hệ thống.