

Troubleshoot Unable to Connect to the Server X509 Certificate Expired

Contents

[Introduction](#)

[Problem](#)

[Solution](#)

Introduction

This document describes steps to resolve the **Unable to connect to the server: x509: certificate has expired or is not yet valid** ERROR.

Problem

Connections to Ultra Cloud Core Subscriber Microservices Infrastructure (SMI) Kubectl throw the error.

Kubernetes control plane node communication happens through SSL tunnel. SSL tunnel typically relies on a set of trusted third-party certificate authorities to establish the authenticity of certificates.

When the certificate expired, control plane node communication stops.

To check certificates expiration: `kubectl get secrets --all-namespaces | grep 'kubernetes.io/tls' | awk '{print $2, $1}' | xargs -n2 sh -c 'echo container $0 namespace $1;kubectl -n $1 get secret $0 -o jsonpath="{.data.tls.crt}" | base64 -d | openssl x509 -noout -enddate; echo -----'`

```
cloud-user@k8-rcdn-primary-1:~$ kubectl get secrets --all-namespaces | grep 'kubernetes.io/tls'
| awk '{print $2, $1}' | xargs
gs -n2 sh -c 'echo container $0 namespace $1;kubectl -n $1 get secret $0 -o
jsonpath="{.data.tls.crt}" | base64 -d | open
ssl x509 -noout -enddate; echo -----'
container cert-cli-cee-k8-rcdn-ops-center-ingress namespace cee-k8-rcdn
notAfter=May 1 16:54:39 2023 GMT
-----
container cert-docs-cee-k8-rcdn-product-documentation-ingress namespace cee-k8-rcdn
notAfter=May 1 16:56:04 2023 GMT
-----
container cert-grafana-ingress namespace cee-k8-rcdn
notAfter=May 1 16:56:06 2023 GMT
-----
container cert-restconf-cee-k8-rcdn-ops-center-ingress namespace cee-k8-rcdn
notAfter=May 1 16:54:40 2023 GMT
-----
container cert-show-tac-cee-k8-rcdn-ops-center-ingress namespace cee-k8-rcdn
notAfter=May 1 16:54:40 2023 GMT
-----
```

```
container cert-show-tac-cee-k8-rcdn-smi-show-tac-ingress namespace cee-k8-rcdn
notAfter=May 1 16:56:07 2023 GMT
-----
container cert-cli-smf-rcdn-ops-center-ingress namespace smf-rcdn
notAfter=May 1 16:54:56 2023 GMT
-----
container cert-restconf-smf-rcdn-ops-center-ingress namespace smf-rcdn
notAfter=May 1 16:54:57 2023 GMT
-----
container cert-show-tac-smf-rcdn-ops-center-ingress namespace smf-rcdn
notAfter=May 1 16:54:57 2023 GMT
-----
container cert-cli-smf-rcdn1-ops-center-ingress namespace smf-rcdn1
notAfter=May 1 16:55:07 2023 GMT
-----
container cert-restconf-smf-rcdn1-ops-center-ingress namespace smf-rcdn1
notAfter=May 1 16:55:08 2023 GMT
-----
container cert-show-tac-smf-rcdn1-ops-center-ingress namespace smf-rcdn1
notAfter=May 1 16:55:08 2023 GMT
-----
container cert-cli-smf-rcdn2-ops-center-ingress namespace smf-rcdn2
notAfter=May 3 18:11:26 2023 GMT
-----
container cert-restconf-smf-rcdn2-ops-center-ingress namespace smf-rcdn2
notAfter=May 3 18:11:28 2023 GMT
-----
container cert-show-tac-smf-rcdn2-ops-center-ingress namespace smf-rcdn2
notAfter=May 3 18:11:27 2023 GMT
-----
container cert-cli-smf-rcdn3-ops-center-ingress namespace smf-rcdn3
notAfter=May 3 18:11:41 2023 GMT
-----
container cert-restconf-smf-rcdn3-ops-center-ingress namespace smf-rcdn3
notAfter=May 3 18:11:43 2023 GMT
-----
container cert-show-tac-smf-rcdn3-ops-center-ingress namespace smf-rcdn3
notAfter=May 3 18:11:42 2023 GMT
-----
```

Solution

1. Verify that the apiserver.crt show the correct end date.

```
ubuntu@labnode-cnat-cnat-core-primary1:~$ cd /data/kubernetes/pki
ubuntu@labnode-cnat-cnat-core-primary1:/data/kubernetes/pki$ sudo su
root@labnode-cnat-cnat-core-primary1:/data/kubernetes/pki# sudo cat
/data/kubernetes/pki/apiserver.crt | openssl x509 -enddate -noout
notAfter=Feb 17 08:22:04 2022 GMT
```

2. Check the end date in SSL.

```
ubuntu@labnode-cnat-cnat-core-primary1:~$ echo | openssl s_client -showcerts -servername
gnupg.org -connect localhost:6443 2>/dev/null | openssl x509 -inform pem -noout -text
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 44335566778899aabba (0xabcdef0123456789)
Signature Algorithm: sha256WithRSAEncryption
Issuer: CN = kubernetes
Validity
```

Not Before: Mar 17 11:59:23 2020 GMT

Not After : Mar 19 10:37:35 2021 GMT

3. Check the docker container state.

```
root@labnode-cnat-cnat-core-primary1:/data/kubernetes/pki# docker ps -f "name=k8s_kube-apiserver"
```

```
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
f988867819ed c2c9a0406787 "kube-apiserver --ad..." 12 months ago Up 12 months k8s_kube-apiserver_kube-apiserver-labnode-cnat-cnat-core-primary1_kube-system_00112233445566778899aabbccddeeff_0
```

```
root@labnode-cnat-cnat-core-primary1:/data/kubernetes/pki#
```

```
root@labnode-cnat-cnat-core-primary1:/data/kubernetes/pki# docker ps -f "name=k8s_kube-controller"
```

```
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
929a8f1ef716 6e4bffa46d70 "kube-controller-man..." 3 days ago Up 3 days k8s_kube-controller-manager_kube-controller-manager-labnode-cnat-cnat-core-primary1_kube-system_112233445566778899aabbccddeeff00_2
```

```
root@labnode-cnat-cnat-core-primary1:/data/kubernetes/pki# docker ps -f "name=k8s_kube-scheduler"
```

```
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
32783a2c3a71 eba1ae204a2 "kube-scheduler --au..." 12 months ago Up 12 months k8s_kube-scheduler_kube-scheduler-labnode-cnat-cnat-core-primary1_kube-system_2233445566778899aabbccddeeff0011_1
```

```
root@labnode-cnat-cnat-core-primary1:/data/kubernetes/pki#
```

4. Restart docker containers of kube-apiserver and kube-scheduler on all three control plane nodes.

```
docker ps -f "name=k8s_kube-apiserver" -q | xargs docker restart
```

```
docker ps -f "name=k8s_kube-scheduler" -q | xargs docker restart
```

5. Confirm that the apiserver.crt show the correct end date.

```
root@labnode-cnat-cnat-core-primary1:/data/kubernetes/pki# sudo cat
```

```
/data/kubernetes/pki/apiserver.crt | openssl x509 -enddate -noout
```

```
notAfter=Feb 17 08:22:04 2022 GMT
```

6. Verify that the end date is updated in SSL and it has the correct end date.

```
echo | openssl s_client -showcerts -servername gnupg.org -connect localhost:6443 2>/dev/null |
```

```
openssl x509 -inform pem -noout -text
```

7. Verify that the cluster is healthy

Please refer to the [Cisco Ultra Cloud Core - Subscriber Microservices Infrastructure guides](#) for details on the operations.