# Renewing Kubernetes cluster certificates

Last Updated: 2021-03-03

The Kubernetes cluster certificates have a lifespan of one year. If the Kubernetes cluster certificate expires on the Kubernetes master, then the **kubelet** service will fail. Issuing a **kubectl** command, such as `kubectl get pods` or `kubectl exec -it` *`container_name`* `bash`, will result in a message similar to **`Unable to connect to the server: x509: certificate has expired or is not yet valid`**.

## Procedure

1. Log on to the Kubernetes master node as the `root` user and run the following command to check when the Kubernetes certificates will expire.

```
kubeadm alpha certs check-expiration
```

   The output will be similar to the following. In this case the certificates will expire in 273 days.

```
CERTIFICATE                EXPIRES                  RESIDUAL TIME   EXTERNALLY MANAGED
admin.conf                 Sep 17, 2020 21:24 UTC   273d            no
apiserver                  Sep 17, 2020 21:24 UTC   273d            no
apiserver-etcd-client      Sep 17, 2020 21:24 UTC   273d            no
apiserver-kubelet-client   Sep 17, 2020 21:24 UTC   273d            no
controller-manager.conf    Sep 17, 2020 21:24 UTC   273d            no
etcd-healthcheck-client    Sep 17, 2020 21:24 UTC   273d            no
etcd-peer                  Sep 17, 2020 21:24 UTC   273d            no
etcd-server                Sep 17, 2020 21:24 UTC   273d            no
front-proxy-client         Sep 17, 2020 21:24 UTC   273d            no
scheduler.conf             Sep 17, 2020 21:24 UTC   273d            no
```

2. Run the following commands to back up the existing Kubernetes certificates:

```
mkdir -p $HOME/fcik8s-old-certs/pki
/bin/cp -p /etc/kubernetes/pki/*.* $HOME/fcik8s-old-certs/pki
ls -l $HOME/fcik8s-old-certs/pki/
```

   The output will be similar to the following:

```
total 56
-rw-r--r-- 1 root root 1261 Sep  4  2019 apiserver.crt
-rw-r--r-- 1 root root 1090 Sep  4  2019 apiserver-etcd-client.crt
-rw------- 1 root root 1679 Sep  4  2019 apiserver-etcd-client.key
-rw------- 1 root root 1679 Sep  4  2019 apiserver.key
-rw-r--r-- 1 root root 1099 Sep  4  2019 apiserver-kubelet-client.crt
-rw------- 1 root root 1679 Sep  4  2019 apiserver-kubelet-client.key
-rw-r--r-- 1 root root 1025 Sep  4  2019 ca.crt
-rw------- 1 root root 1675 Sep  4  2019 ca.key
-rw-r--r-- 1 root root 1038 Sep  4  2019 front-proxy-ca.crt
-rw------- 1 root root 1675 Sep  4  2019 front-proxy-ca.key
-rw-r--r-- 1 root root 1058 Sep  4  2019 front-proxy-client.crt
-rw------- 1 root root 1679 Sep  4  2019 front-proxy-client.key
-rw------- 1 root root 1675 Sep  4  2019 sa.key
-rw------- 1 root root  451 Sep  4  2019 sa.pub
```

3. Run the following commands to back up the existing configurtion files:

```
/bin/cp -p /etc/kubernetes/*.conf $HOME/fcik8s-old-certs
ls -ltr $HOME/fcik8s-old-certs
```

   The output will be similar to the following:

```
total 36
-rw------- 1 root root 5451 Sep  4  2019 admin.conf
-rw------- 1 root root 5595 Sep  4  2019 kubelet.conf
-rw------- 1 root root 5483 Sep  4  2019 controller-manager.conf
-rw------- 1 root root 5435 Sep  4  2019 scheduler.conf
drwxr-xr-x 2 root root 4096 Dec 19 21:21 pki
```

4. Run the following commands to back up your home configuration:

```
mkdir -p $HOME/fcik8s-old-certs/.kube
/bin/cp -p ~/.kube/config $HOME/fcik8s-old-certs/.kube/.
ls -l $HOME/fcik8s-old-certs/.kube/.
```

   The output will be similar to the following:

```
-rw------- 1 root root 5451 Sep  4  2019 config
```

5. Run the following command to renew all the Kubernetes certificates:

```
kubeadm alpha certs renew all
```

   The output of the command will be similar to the following:

```
certificate embedded in the kubeconfig file for the admin to use and for kubeadm itself renewed
certificate for serving the Kubernetes API renewed
certificate the apiserver uses to access etcd renewed
certificate for the API server to connect to kubelet renewed
certificate embedded in the kubeconfig file for the controller manager to use renewed
certificate for liveness probes to healtcheck etcd renewed
certificate for etcd nodes to communicate with each other renewed
certificate for serving etcd renewed
certificate for the front proxy client renewed
certificate embedded in the kubeconfig file for the scheduler manager to use renewed
```

6. Run the following command to confirm the certificates have been renewed and will expire in 364 days:

```
kubeadm alpha certs check-expiration
```

   The output should look similar to the following:

```
CERTIFICATE                EXPIRES                  RESIDUAL TIME   EXTERNALLY MANAGED
admin.conf                 Dec 20, 2021 02:35 UTC   364d            no
apiserver                  Dec 20, 2021 02:35 UTC   364d            no
apiserver-etcd-client      Dec 20, 2021 02:35 UTC   364d            no
apiserver-kubelet-client   Dec 20, 2021 02:35 UTC   364d            no
controller-manager.conf    Dec 20, 2021 02:35 UTC   364d            no
etcd-healthcheck-client    Dec 20, 2021 02:35 UTC   364d            no
etcd-peer                  Dec 20, 2021 02:35 UTC   364d            no
etcd-server                Dec 20, 2021 02:35 UTC   364d            no
front-proxy-client         Dec 20, 2021 02:35 UTC   364d            no
scheduler.conf             Dec 20, 2021 02:35 UTC   364d            no
```

7. Confirm the **`kubelet`** services are running and communication between the worker nodes and the Kubernetes master is working.
8. After waiting a few minutes, run the following command from the Kubernetes master node to confirm that the worker nodes are available:

```
kubectl get nodes
```

   If you get a response similar to the following:

```
The connection to the server 9.37.21.119:6443 was refused - did you specify the right host or port?
```

continue with the next steps to resolve the issue. Otherwise, your Kubernetes cluster certificates have been successfully renewed.

9. Run the following command:

```
diff $HOME/fcik8s-old-certs/kubelet.conf /etc/kubernetes/kubelet.conf
```

If there is no output, the `kubelet.conf` file was not updated with the new certificate information.

10. Update the `/etc/kubernetes/kubelet.conf` file and display the difference from the old version to the new one:

```
cd /etc/kubernetes
sudo kubeadm alpha kubeconfig user --org system:nodes --client-name system:node:$(hostname) > kubelet.conf
diff $HOME/fcik8s-old-certs/kubelet.conf /etc/kubernetes/kubelet.conf
```

If the output shows a difference, the file `kubelet.conf` was updated with the new certificate information.

11. Run the following command:

```
diff ~/.kube/config $HOME/fcik8s-old-certs/.kube/config
```

If there is no output, the `config` file still has the outdated keys and certificate values in it.

12. Update `client-certificate-data` and `client-key-data` in `~/.kube/config` with the values from the updated file in `/etc/kubernetes/kubelet.conf`:
    - 
      ```
      cat /etc/kubernetes/kubelet.conf
      ```

      Select and copy the output after `client-key-data:`.
    - In the `~/.kube/config` file, replace the information after `client-key-data:` with the text copied in the previous step.
    - 
      ```
      cat /etc/kubernetes/kubelet.conf
      ```

      Select and copy the output after `client-certificate-data:`.
    - In the `~/.kube/config` file, replace the information after `client-certificate-data:` with the text copied in the previous step.

13. Restart the kubelet service:

```
systemctl daemon-reload&&systemctl restart kubelet
```

This command is successful if there is no output.

14. Verify master and worker nodes are available:

```
kubectl get nodes
```

15. Verify all pods are in the running state:

```
kubectl get pods
```

**Parent topic:**

→ Administering Kubernetes