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 `kubectel 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

To regenerate a new certificate and update worker nodes:

Create a configuration file in `/etc/root` named `kubeadm.yaml` with `advertiseAddress` set to the IP address of your Kubernetes master node. For example:

```
apiVersion: kubeadm.k8s.io/v1alpha1
kind: MasterConfiguration
api:
  advertiseAddress: 10.165.80.110
kubernetesVersion: v1.14.1
```

Remove the existing certificate and key files:

```
rm /etc/kubernetes/pki/apiserver.key
rm /etc/kubernetes/pki/apiserver.crt
rm /etc/kubernetes/pki/apiserver-kubelet-client.crt
rm /etc/kubernetes/pki/apiserver-kubelet-client.key
rm /etc/kubernetes/pki/front-proxy-client.crt
rm /etc/kubernetes/pki/front-proxy-client.key
```

Create new certificates:

```
kubeadm --config /root/kubeadm.yaml alpha phase certs apiserver
kubeadm --config /root/kubeadm.yaml alpha phase certs apiserver-kubelet-cl
kubeadm --config /root/kubeadm.yaml alpha phase certs front-proxy-client
```

Remove the old configuration files:

```
rm /etc/kubernetes/admin.conf
rm /etc/kubernetes/kubelet.conf
rm /etc/kubernetes/controller-manager.conf
rm /etc/kubernetes/scheduler.conf
```

Generate new configuration files:

```
kubeadm --config /root/kubeadm.yaml alpha phase kubeconfig all
```

Ensure that your `kubectl` service is using the correct configuration files:

```
cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
export KUBECONFIG=.kube/config
```

Reboot the Kubernetes master node.

After the server restarts, check to ensure that the **kubelet** service is running:

```
systemctl status kubelet
```

To rejoin the worker nodes, you require a cluster token.
For Kubernetes 1.7 the cluster token is preset. For Kubernetes versions later than 1.7, you must create a new token since the token generated at installation has a limited lifetime:

```
kubeadm token create
```

SSH into each of the worker nodes and reconnect them to the Kubernetes master node.

Join the worker nodes back into the Kubernetes cluster:

```
kubeadm join --token=cluster_token master_ip:6443
```

Where *cluster_token* is the token created in Step 9 and *master_ip* is the IP address of the Kubernetes master node.

> **Note**
>
> Some versions of **kubeadm** use a `--print-join-command` command line parameter. In these cases, **kubeadm** outputs the **kubeadm join** command required to reconnect with the Kubernetes master. If this occurs, enter this command (copy and paste) on each worker node.

Confirm that **kubelet** services are running and communication between the worker nodes and Kubernetes master is working.

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
```