GENESYS

Kubernetes Certificate Renewal

To confirm the Kubernetes certificate is expired or not use any one of the below commands.

1. openssl x509 -in /etc/kubernetes/pki/apiserver.crt -noout -text |grep ' Not '
2. find /etc/kubernetes/pki/ -type f -name "*.crt" -print|egrep -v 'ca.crt$'|xargs -L 1 -t -i bash -c 'openssl x509 -noout -text -in {}|grep After'
3. kubeadm alpha certs check-expiration
4. kubeadm certs check-expiration

If the Certificate is expired, then you have to renew the Kubernetes certificate with the below steps.

1. First backup your meta data.
2 cd /etc/kubernetes/pki/
3. mv {apiserver.crt,apiserver-etcd-client.key,apiserver-kubelet-client.crt,front-proxy-ca.crt,front-proxy-client.crt,front-proxy-client.key,front-proxy-ca.key,apiserver-kubelet-client.key,apiserver.key,apiserver-etcd-client.crt} /tmp ---------- you can use any other folder if you'd like
4. kubeadm init phase certs all
5. cd /etc/kubernetes/
6. mv {admin.conf,controller-manager.conf,kubelet.conf,scheduler.conf} /tmp ---------- again, you can choose any other folder here
7. kubeadm init phase kubeconfig all
8. reboot

After run the following:
1. cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
2. systemctl daemon-reload
3. systemctl start kubelet

If we get a healthy kubelet here run the following:
kubectl config set-context $(kubectl config current-context) --namespace=genesys

And finally test your pods by running
kubectl get all --all-namespaces

In **Kubernetes v1.15** this "alpha certs renew all" command was added. So if k8s version is 1.15 and newer it is possible to use it.

kubeadm alpha certs renew all

Certificates used by Kubernetes control plane components such as the Kubernetes API server have a lifetime of 1 year. Replicated has several mechanisms to ensure these certificates are rotated before they expire.
Automatic certificate renewal --> If you have more complex requirements for certificate renewal, you can opt out from the default behavior by passing --certificate-renewal=false to kubeadm upgrade apply or to kubeadm upgrade node.

```
[root@gcxi ~]# kubeadm upgrade node --certificate-renewal=false
[upgrade] Reading configuration from the cluster...
[upgrade] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
[upgrade] Upgrading your Static Pod-hosted control plane instance to version "v1.20.15"...
Static pod: kube-apiserver-gcxi.rptgenlab.com hash: 5878d82882d8a2334c7972ea1333ef39
Static pod: kube-controller-manager-gcxi.rptgenlab.com hash: 0109ce5dab4c57f822884357d67bc068
Static pod: kube-scheduler-gcxi.rptgenlab.com hash: e8f872f9a07112e96e684366d7248982
[upgrade/etcd] Upgrading to TLS for etcd
Static pod: etcd-gcxi.rptgenlab.com hash: a6fe12362ad6276dbe22f21febbe5ad7
[upgrade/staticpods] Preparing for "etcd" upgrade
[upgrade/staticpods] Current and new manifests of etcd are equal, skipping upgrade
[upgrade/etcd] Waiting for etcd to become available
[upgrade/staticpods] Writing new Static Pod manifests to "/etc/kubernetes/tmp/kubeadm-upgraded-manifests482750610"
[upgrade/staticpods] Preparing for "kube-apiserver" upgrade
[upgrade/staticpods] Current and new manifests of kube-apiserver are equal, skipping upgrade
[upgrade/staticpods] Preparing for "kube-controller-manager" upgrade
[upgrade/staticpods] Current and new manifests of kube-controller-manager are equal, skipping upgrade
[upgrade/staticpods] Preparing for "kube-scheduler" upgrade
[upgrade/staticpods] Current and new manifests of kube-scheduler are equal, skipping upgrade
[upgrade] The control plane instance for this node was successfully updated!
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[upgrade] The configuration for this node was successfully updated!
[upgrade] Now you should go ahead and upgrade the kubelet package using your package manager.
[root@gcxi ~]#
```

Manual certificate renewal --> You can renew your certificates manually at any time with the **kubeadm alpha certs renew** command

```
[root@gcxi ~]# kubeadm alpha certs renew all
Command "all" is deprecated, please use the same command under "kubeadm certs"
[renew] Reading configuration from the cluster...
[renew] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'

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 healthcheck 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

Done renewing certificates. You must restart the kube-apiserver, kube-controller-manager, kube-scheduler and etcd, so that they can use the new certificates.
[root@gcxi ~]#
```

Please refer below link for more details
https://kubernetes.io/docs/tasks/administer-cluster/kubeadm/kubeadm-certs/#check-certificate-expiration

Most Preventive way for certificate issue in kubernetes :
It is a best practice to upgrade your cluster frequently in order to stay secure, this will auto renew your certificate.
https://kubernetes.io/docs/tasks/administer-cluster/kubeadm/kubeadm-upgrade/