**Topics:**

1. **Renew Expired Kubernetes Certificates**
2. **Fix Not Ready worker Node Issue**

## **Renew Expired Kubernetes Certificates**

Client certificates generated by [kubeadm](https://kubernetes.io/docs/reference/setup-tools/kubeadm/) expire after 1 year. This document explains how to manage certificate renewals with kubeadm.

You should regularly check the certificate expiration by the following command to make sure to renew the certs before expiration

kubeadm certs check-expiration

You can renew your certificates manually at any time with the kubeadm certs renew command.

This command performs the renewal using CA (or front-proxy-CA) certificate and key stored in /etc/kubernetes/pki.

After running the command you should restart the control plane Pods. This is required since dynamic certificate reload is currently not supported for all components and certificates. [Static Pods](https://kubernetes.io/docs/tasks/configure-pod-container/static-pod/) are managed by the local kubelet and not by the API Server, thus kubectl cannot be used to delete and restart them. To restart a static Pod you can temporarily remove its manifest file from /etc/kubernetes/manifests/ and wait for 20 seconds (see the fileCheckFrequency value in [KubeletConfiguration struct](https://kubernetes.io/docs/reference/config-api/kubelet-config.v1beta1/). The kubelet will terminate the Pod if it's no longer in the manifest directory. You can then move the file back and after another fileCheckFrequency period, the kubelet will recreate the Pod and the certificate renewal for the component can complete.

Finally copy the kube-config to the ~/.kube/config

cp /etc/kubernetes/admin.conf ~/.kube/config

kubectl get po

Congratulation you have successfully renew the kubernetes certificates.

## **Fix Not Ready worker Node Issue**

Before start working on any worker node first make it drain or unscheduled and put it on maintenance mode

kubectl drain <node name>

For fixing the node issue we need to make sure the following checklist

1. Version of docker, kubeadm, kubelet are same and supported as master
2. Kubelet service is running fine
3. Docker is ruining fine
4. Check the kubelet logs

In your case problem is with the docker, kubelet is not able to connect with the docker because docker needs sudo permission to run. We run the following commands

sudo groupadd docker

sudo usermod -aG docker $USER

newgrp docker

This fix the docker sudo permission issue.

Then we stop the docker service and restart it again.

Now docker issue has been resolved

Now I need to remove this node from the cluster by running the following command on worker

sudo kubeadm reset

Now this node is reset we need to add back to the cluster.

Run the following command:

Master node:

1. kubeadm token create --print-join-command

Copy the join command from the above command and run it on worker node.

Worker node:

kubeadm join 192.168.1.130:6443 --token qt57zu.wuvqh64un13trr7x --discovery-token-ca-cert-hash sha256:5ad014cad868fdfe9388d5b33796cf40fc1e8c2b3dccaebff0b066a0532e8723

kubectl get nodes

Now you can see that you node is available and ready.

Thank you