Lesson 02 Demo 04

Working with kubeadm

Objective: To generate kubeadm tokens, verify certificate expiration, and access certificate encryption key for Kubernetes cluster management

Tools required: kubeadm, kubectl, kubelet, and containerd

Prerequisites: A Kubernetes cluster (refer to Demo 01 from Lesson 01 for setting up a cluster)

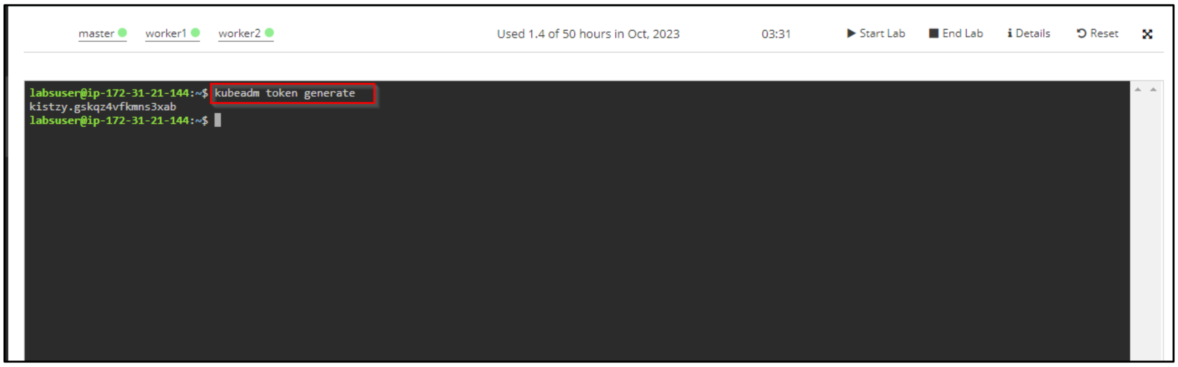
Steps to be followed:

1. Generate tokens for kubeadm

Step 1: Generate tokens for kubeadm

1.1 Generate a new token using the following command:

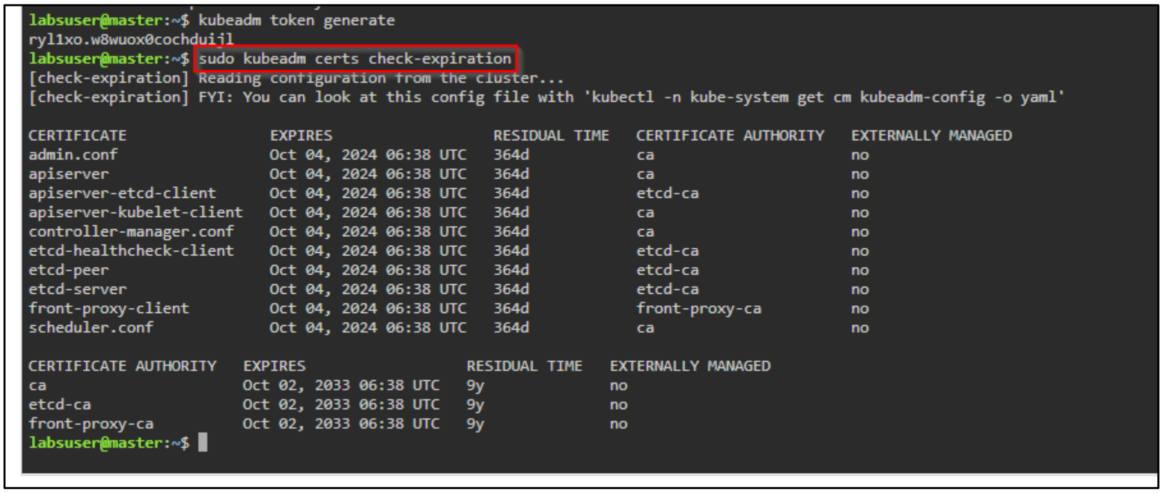
kubeadm token generate



```
labsuser@ip-172-31-21-144:~$ kubeadm token generate
kistzy.gskqz4vfkmns3xab
labsuser@ip-172-31-21-144:~$
```

This command will generate a new token for node join operations.

1.2 Check the expiration of all the certificates in the cluster using the command below:
sudo kubeadm certs check-expiration

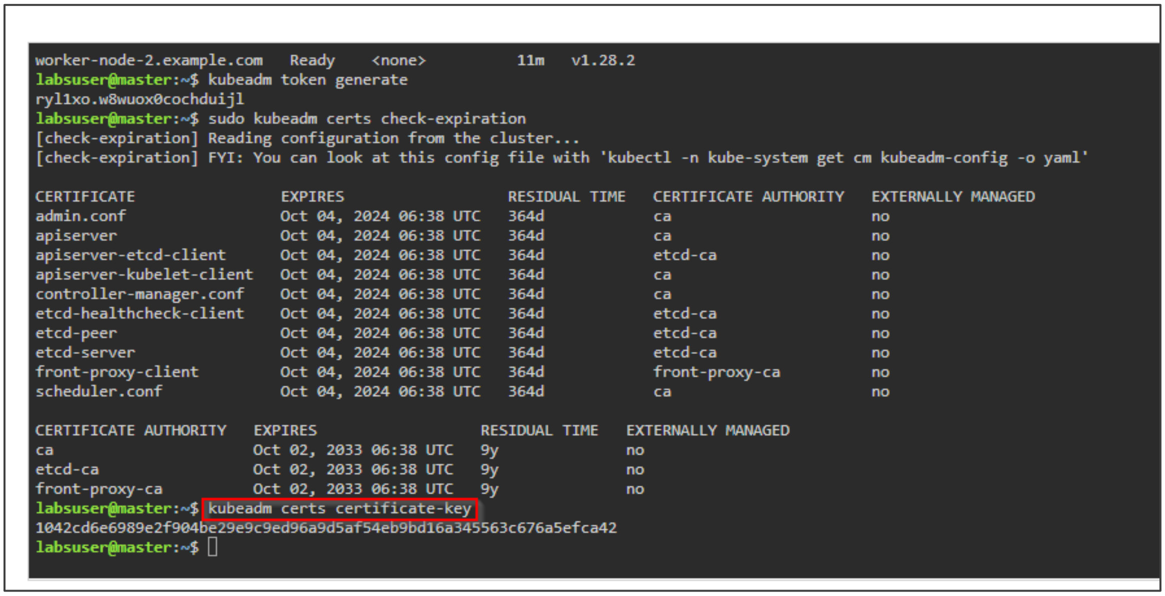


```
labsuser@master:~$ kubeadm token generate
ryl1xo.w8wuox0cochduijl
labsuser@master:~$ sudo kubeadm certs check-expiration
[check-expiration] Reading configuration from the cluster...
[check-expiration] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'

CERTIFICATE                EXPIRES                  RESIDUAL TIME   CERTIFICATE AUTHORITY   EXTERNALLY MANAGED
admin.conf                 Oct 04, 2024 06:38 UTC   364d            ca                      no
apiserver                  Oct 04, 2024 06:38 UTC   364d            ca                      no
apiserver-etcd-client      Oct 04, 2024 06:38 UTC   364d            etcd-ca                 no
apiserver-kubelet-client   Oct 04, 2024 06:38 UTC   364d            ca                      no
controller-manager.conf    Oct 04, 2024 06:38 UTC   364d            ca                      no
etcd-healthcheck-client    Oct 04, 2024 06:38 UTC   364d            etcd-ca                 no
etcd-peer                  Oct 04, 2024 06:38 UTC   364d            etcd-ca                 no
etcd-server                Oct 04, 2024 06:38 UTC   364d            etcd-ca                 no
front-proxy-client         Oct 04, 2024 06:38 UTC   364d            front-proxy-ca          no
scheduler.conf             Oct 04, 2024 06:38 UTC   364d            ca                      no

CERTIFICATE AUTHORITY   EXPIRES                  RESIDUAL TIME   EXTERNALLY MANAGED
ca                      Oct 02, 2033 06:38 UTC   9y              no
etcd-ca                 Oct 02, 2033 06:38 UTC   9y              no
front-proxy-ca          Oct 02, 2033 06:38 UTC   9y              no
labsuser@master:~$
```

1.3 Run the following command to obtain the key used to encrypt the certificates:
kubeadm certs certificate-key

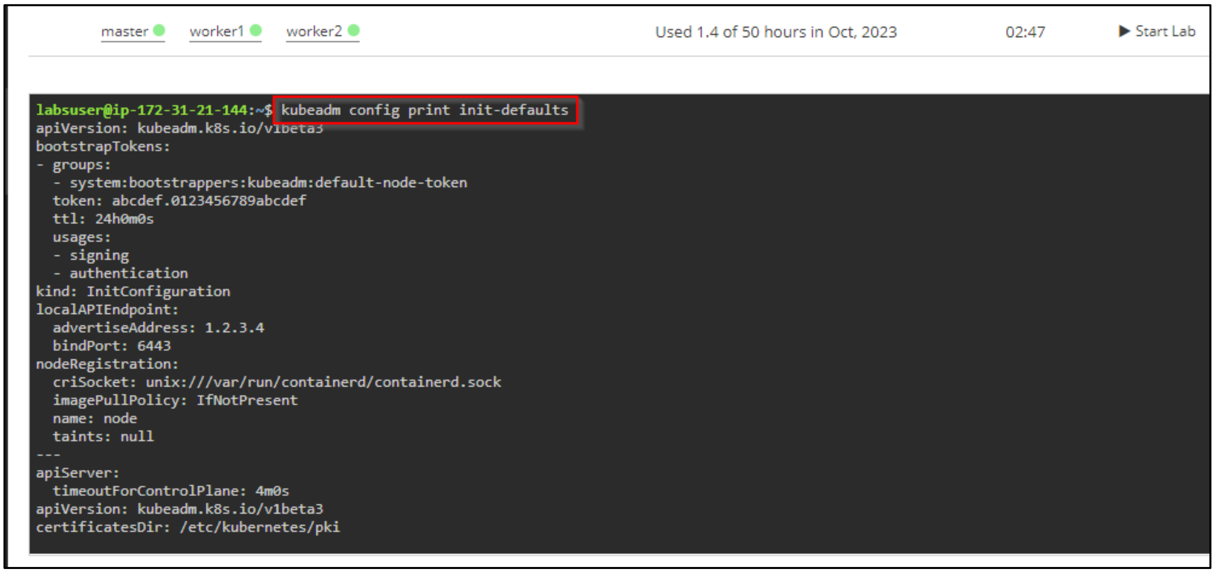


```
worker-node-2.example.com   Ready    <none>          11m   v1.28.2
labsuser@master:~$ kubeadm token generate
ryl1xo.w8wuox0cochduijl
labsuser@master:~$ sudo kubeadm certs check-expiration
[check-expiration] Reading configuration from the cluster...
[check-expiration] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'

CERTIFICATE                EXPIRES                  RESIDUAL TIME   CERTIFICATE AUTHORITY   EXTERNALLY MANAGED
admin.conf                 Oct 04, 2024 06:38 UTC   364d            ca                      no
apiserver                  Oct 04, 2024 06:38 UTC   364d            ca                      no
apiserver-etcd-client      Oct 04, 2024 06:38 UTC   364d            etcd-ca                 no
apiserver-kubelet-client   Oct 04, 2024 06:38 UTC   364d            ca                      no
controller-manager.conf    Oct 04, 2024 06:38 UTC   364d            ca                      no
etcd-healthcheck-client    Oct 04, 2024 06:38 UTC   364d            etcd-ca                 no
etcd-peer                  Oct 04, 2024 06:38 UTC   364d            etcd-ca                 no
etcd-server                Oct 04, 2024 06:38 UTC   364d            etcd-ca                 no
front-proxy-client         Oct 04, 2024 06:38 UTC   364d            front-proxy-ca          no
scheduler.conf             Oct 04, 2024 06:38 UTC   364d            ca                      no

CERTIFICATE AUTHORITY   EXPIRES                  RESIDUAL TIME   EXTERNALLY MANAGED
ca                      Oct 02, 2033 06:38 UTC   9y              no
etcd-ca                 Oct 02, 2033 06:38 UTC   9y              no
front-proxy-ca          Oct 02, 2033 06:38 UTC   9y              no
labsuser@master:~$ kubeadm certs certificate-key
1042cd6e6989e2f904be29e9c9ed96a9d5af54eb9bd16a345563c676a5efca42
labsuser@master:~$ 
```

1.4 Print the default initialization configuration using the following command:
kubeadm config print init-defaults



```
labsuser@ip-172-31-21-144:~$ kubeadm config print init-defaults
apiVersion: kubeadm.k8s.io/v1beta3
bootstrapTokens:
- groups:
  - system:bootstrappers:kubeadm:default-node-token
  token: abcdef.0123456789abcdef
  ttl: 24h0m0s
  usages:
  - signing
  - authentication
kind: InitConfiguration
localAPIEndpoint:
  advertiseAddress: 1.2.3.4
  bindPort: 6443
nodeRegistration:
  criSocket: unix:///var/run/containerd/containerd.sock
  imagePullPolicy: IfNotPresent
  name: node
  taints: null
---
apiServer:
  timeoutForControlPlane: 4m0s
apiVersion: kubeadm.k8s.io/v1beta3
certificatesDir: /etc/kubernetes/pki
```

By following these steps, you have successfully generated kubeadm tokens, verified certificate expiration, and accessed the certificate encryption key for Kubernetes cluster management.