

## Lesson 03 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 should already be set up (refer to the steps provided in Lesson 02, Demo 01 for guidance)

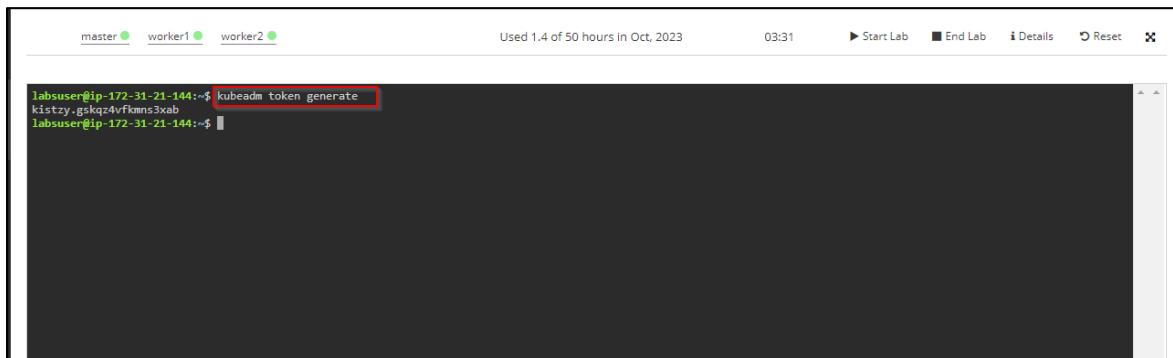
Steps to be followed:

1. Generate tokens for kubeadm

#### Step 1: Generate tokens for kubeadm

1.1 Generate a new token using the following command:

**kubeadm token generate**



The screenshot shows a terminal window with a dark background. At the top, there is a status bar with labels 'master', 'worker1', and 'worker2' each followed by a green dot. To the right of these labels, it says 'Used 1.4 of 50 hours in Oct, 2023', '03:31', and buttons for 'Start Lab', 'End Lab', 'Details', 'Reset', and a close icon. The terminal prompt is 'labsuser@ip-172-31-21-144:~\$'. The command 'kubeadm token generate' has been entered and is highlighted with a red rectangular box. Below the command, the output 'k1strz.gskqz4vfkms3xnb' is visible. The prompt is now '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**

```

master.example.com      Ready    control-plane   14m   v1.28.2
worker-node-1.example.com Ready    <none>          12m   v1.28.2
worker-node-2.example.com Ready    <none>          11m   v1.28.2
labsuser@master:~$ kubeadm token generate
ryl1xo.w8wuox0cochdui1j
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 To obtain the key used to encrypt the certificates, enter the following command: **kubeadm certs certificate-key**

```

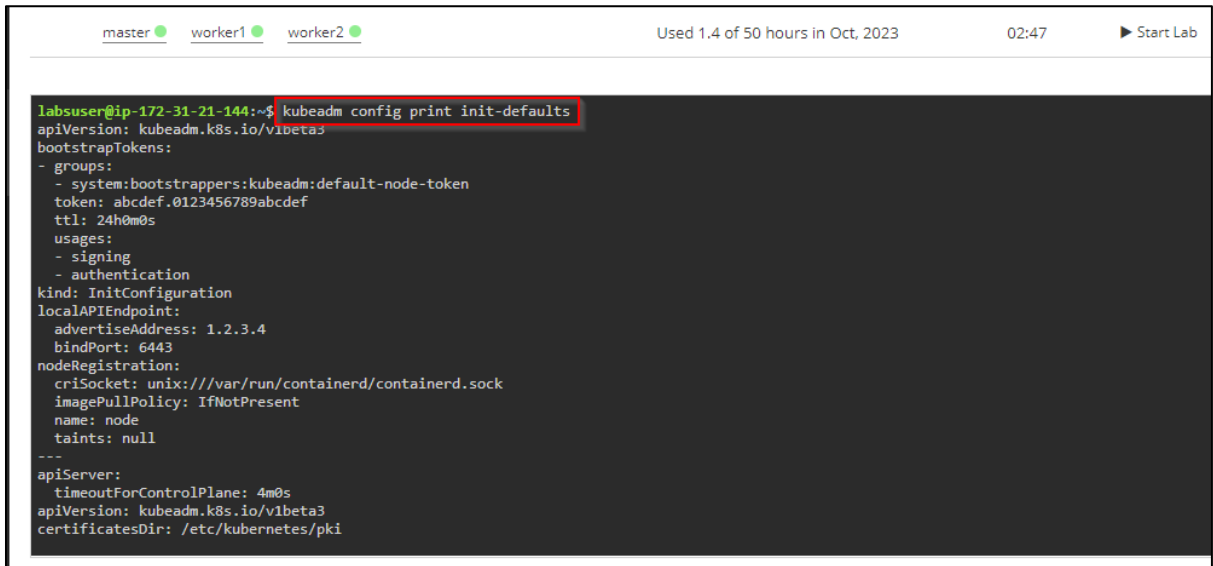
worker-node-2.example.com Ready    <none>          11m   v1.28.2
labsuser@master:~$ kubeadm token generate
ryl1xo.w8wuox0cochdui1j
labsuser@master:~$ sudo kubeadm certs check-expiration
[check-expiration] Reading configuration from the cluster...
[check-expiration] FYI: You can look at this config file 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**



The screenshot shows a terminal window with a dark background. At the top, there are three tabs labeled 'master', 'worker1', and 'worker2', each with a green dot. To the right of the tabs, it says 'Used 1.4 of 50 hours in Oct, 2023'. Further right is a timer '02:47' and a button 'Start Lab'. The terminal prompt is 'labsuser@ip-172-31-21-144:~\$'. The command 'kubeadm config print init-defaults' is entered and highlighted with a red box. The output is as follows:

```
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 prepared your Kubernetes cluster for secure node joins, ensured certificate validity, and obtained the essential certificate encryption key, enhancing the security and functionality of your cluster.