Lesson 02 Demo 07

Backing up and Restoring Etcd Cluster Data

Objective: To backup and restore the data of an etcd cluster

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. Back up etcd cluster data
2. Obtain data from the etcd cluster

Step 1: Back up etcd cluster data

1.1 Install the **etcd-client** using the command below:

sudo apt install etcd-client

```
labsuser@master:~$ sudo apt install etcd-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  etcd-client
0 upgraded, 1 newly installed, 0 to remove and 17 not upgraded.
1 not fully installed or removed.
Need to get 4575 kB of archives.
After this operation, 15.3 MB of additional disk space will be used.
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 etcd-client amd64 3.3.25+dfsg-7ubuntu0.22.04.1 [4575 kB]
Fetched 4575 kB in 1s (6703 kB/s)
Selecting previously unselected package etcd-client.
(Reading database ... 218319 files and directories currently installed.)
Preparing to unpack .../etcd-client_3.3.25+dfsg-7ubuntu0.22.04.1_amd64.deb ...
Unpacking etcd-client (3.3.25+dfsg-7ubuntu0.22.04.1) ...
Setting up grub-efi-amd64-signed (1.187.6+2.06-2ubuntu14.4) ...
Installing grub to /boot/efi.
Installing for x86_64-efi platform.
grub-install: warning: EFI variables cannot be set on this system.
grub-install: warning: You will have to complete the GRUB setup manually.
Installation finished. No error reported.
Setting up etcd-client (3.3.25+dfsg-7ubuntu0.22.04.1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...
```

1.2 List all the pods in the **kube-system** namespace using the following command:
kubectl get pods -n kube-system

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
labsuser@master:~$ kubectl get pods -n kube-system
NAME                                     READY   STATUS    RESTARTS          AGE
calico-kube-controllers-7ddc4f45bc-cvlvw 1/1     Running   1 (3h44m ago)     3d20h
calico-node-44f8d                        1/1     Running   1 (3h44m ago)     3d20h
calico-node-gbbnv                        1/1     Running   1 (3h44m ago)     3d20h
calico-node-rzzkd                        1/1     Running   1 (3h44m ago)     3d20h
coredns-5dd5756b68-f9j4v                 1/1     Running   1 (3h44m ago)     3d20h
coredns-5dd5756b68-j6p7z                 1/1     Running   1 (3h44m ago)     3d20h
etcd-master.example.com                  1/1     Running   3 (3h44m ago)     3d20h
kube-apiserver-master.example.com        1/1     Running   3 (3h44m ago)     3d20h
kube-controller-manager-master.example.com 1/1   Running   48 (3h44m ago)    3d20h
kube-proxy-6wdwh                         1/1     Running   1 (3h44m ago)     3d20h
kube-proxy-7ggt6                         1/1     Running   1 (3h44m ago)     3d20h
kube-proxy-82b48                         1/1     Running   1 (3h44m ago)     3d20h
kube-scheduler-master.example.com        1/1     Running   43 (3h44m ago)    3d20h
labsuser@master:~$
```

1.3 Describe the etcd pod in the **kube-system** namespace using the following command and note the IP address:
kubectl describe pods <etcd-pod-name> -n kube-system

```
labsuser@master:~$ kubectl describe pods etcd-master.example.com -n kube-system
Name:                 etcd-master.example.com
Namespace:            kube-system
Priority:             2000001000
Priority Class Name:  system-node-critical
Node:                 master.example.com/172.31.9.176
Start Time:           Tue, 10 Oct 2023 06:08:34 +0000
Labels:               component=etcd
                      tier=control-plane
Annotations:          kubeadm.kubernetes.io/etcd.advertise-client-urls: https://172.31.9.176:2379
                      kubernetes.io/config.hash: 3fde340195623832fd7e0f97757ee09a
                      kubernetes.io/config.mirror: 3fde340195623832fd7e0f97757ee09a
                      kubernetes.io/config.seen: 2023-10-06T13:42:20.738339102Z
                      kubernetes.io/config.source: file
Status:               Running
SeccompProfile:       RuntimeDefault
IP:                   172.31.9.176
IPs:
  IP:           172.31.9.176
Controlled By:  Node/master.example.com
Containers:
  etcd:
    Container ID:  containerd://370ed1940a462a3ed71a364480da293dcf8b4756bd777c1669e7e2d80392ffce
    Image:         registry.k8s.io/etcd:3.5.9-0
    Image ID:      registry.k8s.io/etcd@sha256:e013d0d5e4e25d00c61a7ff839927a1f36479678f11e49502b53a5e0b14f10c3
    Port:          <none>
    Host Port:     <none>
    Command:
```

1.4 Copy the client IP address

```
labsuser@master:~$ kubectl describe pods <etcd-pod-name> -n kube-system
bash: etcd-pod-name: No such file or directory
labsuser@master:~$ kubectl describe pods etcd-master.example.com -n kube-system
Name:                 etcd-master.example.com
Namespace:            kube-system
Priority:             2000001000
Priority Class Name:  system-node-critical
Node:                 master.example.com/172.31.9.176
Start Time:           Tue, 10 Oct 2023 06:08:34 +0000
Labels:               component=etcd
                      tier=control-plane
Annotations:          kubeadm.kubernetes.io/etcd.advertise-client-urls: https://172.31.9.176:2379
                      kubernetes.io/config.hash: 3fde340195623832fd7e0f97757ee09a
                      kubernetes.io/config.mirror: 3fde340195623832fd7e0f97757ee09a
                      kubernetes.io/config.seen: 2023-10-06T13:42:20.738339102Z
                      kubernetes.io/config.source: file
Status:               Running
SeccompProfile:       RuntimeDefault
IP:                   172.31.9.176
IPs:
```

1.5 Set it as an environment variable and confirm its value using the following commands, replacing **<advertise-client-url>** with the one you retrieved:

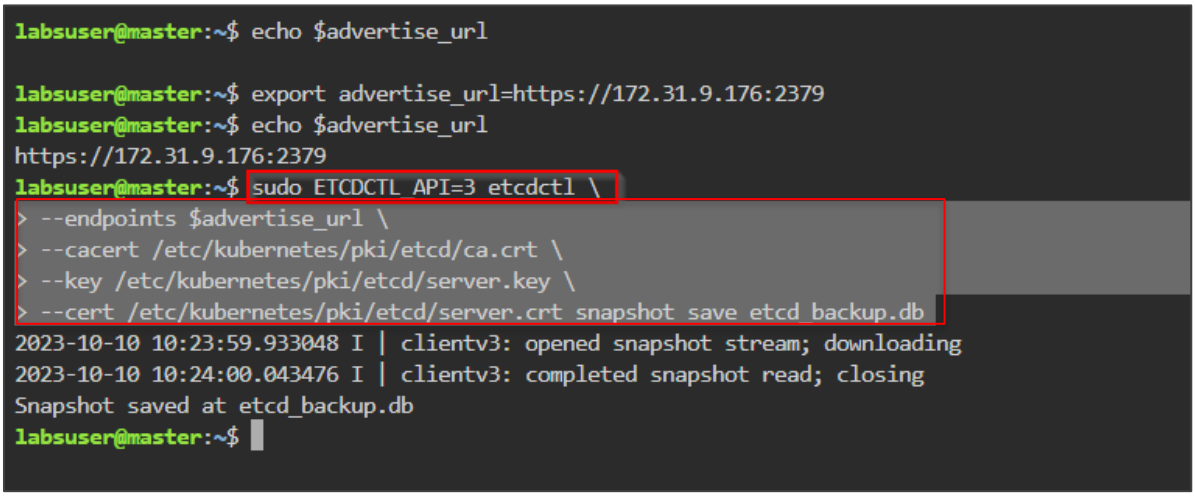
export advertise_url=<advertise-client-url>

echo $advertise_url

```
QoS Class:           Burstable
Node-Selectors:      <none>
Tolerations:         :NoExecute op=Exists
Events:              <none>
labsuser@master:~$ export advertise_url=https://172.31.9.176:2379
labsuser@master:~$ echo $advertise_url
https://172.31.9.176:2379
labsuser@master:~$
```

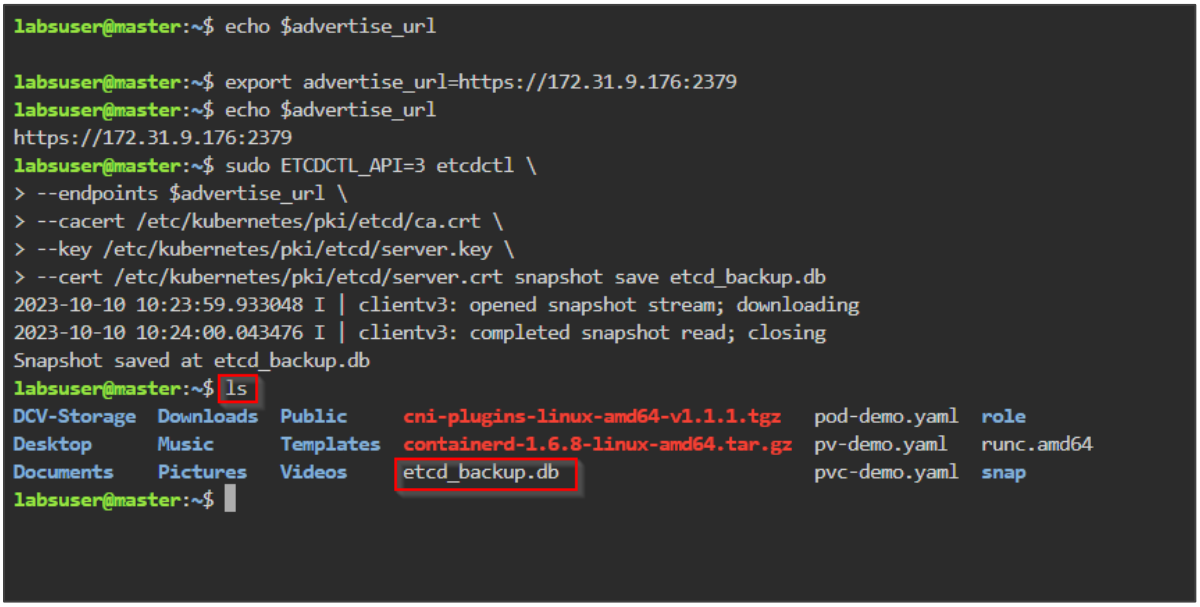
1.6 Back up the etcd data using the command below:

**sudo ETCDCTL_API=3 etcdctl **
**--endpoints $advertise_url **
**--cacert /etc/kubernetes/pki/etcd/ca.crt **
**--key /etc/kubernetes/pki/etcd/server.key **
--cert /etc/kubernetes/pki/etcd/server.crt snapshot save etcd_backup.db



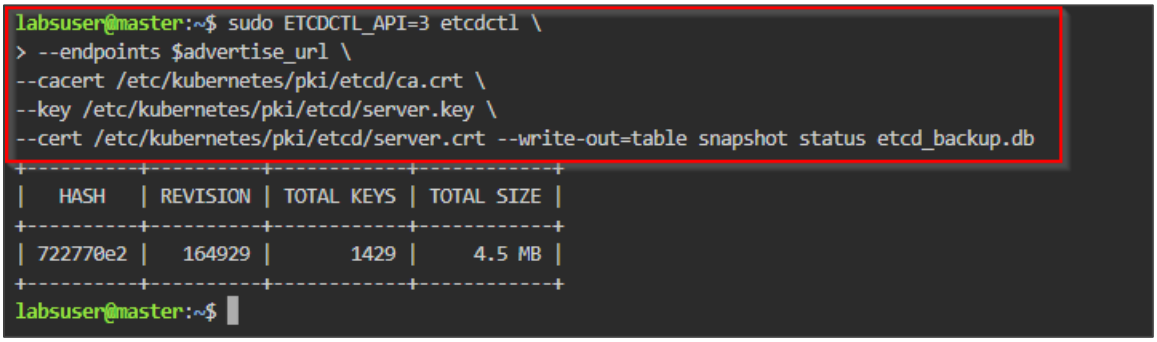
1.7 Check the presence of the newly created **etcd_backup.db** file using the following command:

ls



1.7 Run the following command to verify the etcd backup:

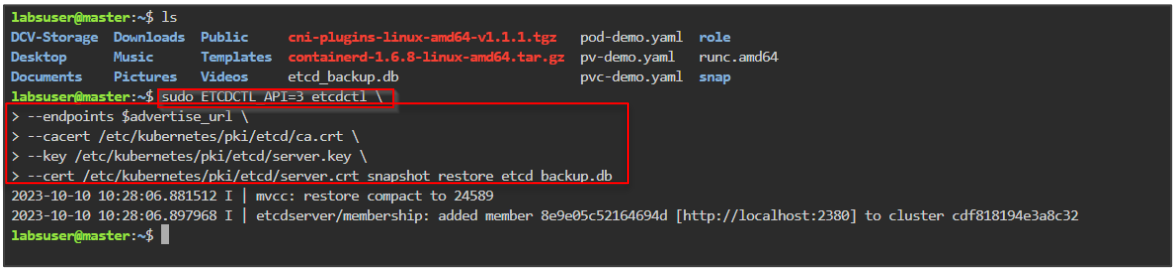
**sudo ETCDCTL_API=3 etcdctl **
**--endpoints $advertise_url **
**--cacert /etc/kubernetes/pki/etcd/ca.crt **
**--key /etc/kubernetes/pki/etcd/server.key **
--cert /etc/kubernetes/pki/etcd/server.crt --write-out=table snapshot status etcd_backup.db



Step 2: Obtain data from the etcd cluster

2.1 Restore the etcd cluster data using the following command:

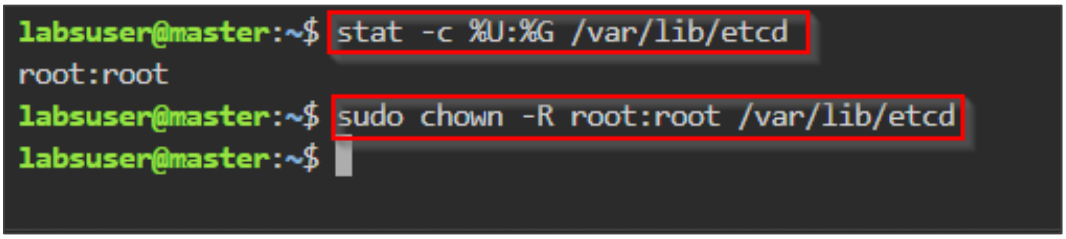
**sudo ETCDCTL_API=3 etcdctl **
**--endpoints $advertise_url **
**--cacert /etc/kubernetes/pki/etcd/ca.crt **
**--key /etc/kubernetes/pki/etcd/server.key **
--cert /etc/kubernetes/pki/etcd/server.crt snapshot restore etcd_backup.db



2.2 Set the proper ownership for the new data directory using the following command:

stat -c %U:%G /var/lib/etcd

sudo chown -R root:root /var/lib/etcd



2.3 Confirm the state of the cluster using the command below:

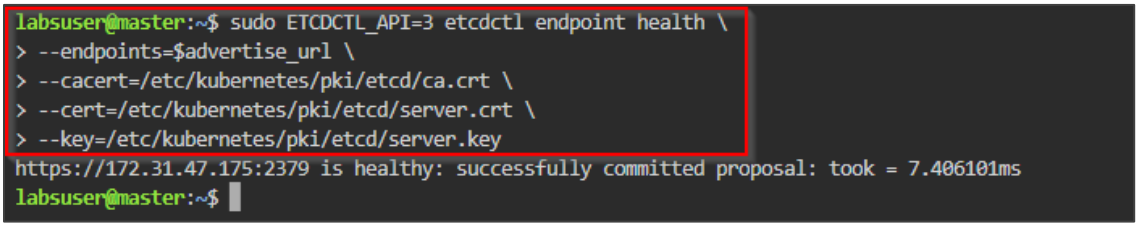
**sudo ETCDCTL_API=3 etcdctl endpoint health **

**--endpoints=$advertise_url **

**--cacert=/etc/kubernetes/pki/etcd/ca.crt **

**--cert=/etc/kubernetes/pki/etcd/server.crt **

--key=/etc/kubernetes/pki/etcd/server.key



By following these steps, you have effectively backed up and restored the data within an etcd cluster in a Kubernetes environment.