

Lesson 03 Demo 07

Backing Up and Restoring Etcd Cluster Data

Objective: To demonstrate how to efficiently backup and restore the data of an etcd cluster within a Kubernetes environment

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

Step 1: Back up the 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
```

1.4 After noting the client URL, set it as an environment variable and confirm its value using the following command:

```
export advertise_url=<advertise-client-url>
echo $advertise_url
```

```
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:
```

```
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 $advertise_url
https://172.31.9.176:2379
labsuser@master:~$
```

Note: Replace <advertise-client-url> with the actual advertise client URL value you retrieved.

1.5 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
```

```
labsuser@master:~$ echo $advertise_url

labsuser@master:~$ export advertise_url=https://172.31.9.176:2379
labsuser@master:~$ echo $advertise_url
https://172.31.9.176:2379
labsuser@master:~$ sudo ETCDCTL_API=3 etcdctl \
> --endpoints $advertise_url \
> --cacert /etc/kubernetes/pki/etcd/ca.crt \
> --key /etc/kubernetes/pki/etcd/server.key \
> --cert /etc/kubernetes/pki/etcd/server.crt snapshot save etcd_backup.db
2023-10-10 10:23:59.933048 I | clientv3: opened snapshot stream; downloading
2023-10-10 10:24:00.043476 I | clientv3: completed snapshot read; closing
Snapshot saved at etcd_backup.db
labsuser@master:~$
```

1.6 Check the presence of the newly created **etcd_backup.db** file:

ls

```
labsuser@master:~$ echo $advertise_url

labsuser@master:~$ export advertise_url=https://172.31.9.176:2379
labsuser@master:~$ echo $advertise_url
https://172.31.9.176:2379
labsuser@master:~$ sudo ETCDCTL_API=3 etcdctl \
> --endpoints $advertise_url \
> --cacert /etc/kubernetes/pki/etcd/ca.crt \
> --key /etc/kubernetes/pki/etcd/server.key \
> --cert /etc/kubernetes/pki/etcd/server.crt snapshot save etcd_backup.db
2023-10-10 10:23:59.933048 I | clientv3: opened snapshot stream; downloading
2023-10-10 10:24:00.043476 I | clientv3: completed snapshot read; closing
Snapshot saved at etcd_backup.db
labsuser@master:~$ ls
DCV-Storage  Downloads  Public      cni-plugins-linux-amd64-v1.1.1.tgz  pod-demo.yaml  role
Desktop      Music      Templates   containerd-1.6.8-linux-amd64.tar.gz  pv-demo.yaml   runc.amd64
Documents    Pictures   Videos     etcd_backup.db                       pvc-demo.yaml  snap
labsuser@master:~$
```

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
```

```
labsuser@master:~$ sudo ETCDCTL_API=3 etcdctl \
> --endpoints $advertise_url \
--cacert /etc/kubernetes/pki/etcd/ca.crt \
--key /etc/kubernetes/pki/etcd/server.key \
--cert /etc/kubernetes/pki/etcd/server.crt --write-out=table snapshot status etcd_backup.db
+-----+-----+-----+-----+
| HASH | REVISION | TOTAL KEYS | TOTAL SIZE |
+-----+-----+-----+-----+
| 722770e2 | 164929 | 1429 | 4.5 MB |
+-----+-----+-----+-----+
labsuser@master:~$
```

Step 2: Obtain the 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
```

```
labsuser@master:~$ echo $advertise_url
labsuser@master:~$ export advertise_url=https://172.31.9.176:2379
labsuser@master:~$ echo $advertise_url
https://172.31.9.176:2379
labsuser@master:~$ sudo ETCDCTL_API=3 etcdctl \
> --endpoints $advertise_url \
> --cacert /etc/kubernetes/pki/etcd/ca.crt \
> --key /etc/kubernetes/pki/etcd/server.key \
> --cert /etc/kubernetes/pki/etcd/server.crt snapshot save etcd_backup.db
2023-10-10 10:23:59.933048 I | clientv3: opened snapshot stream; downloading
2023-10-10 10:24:00.043476 I | clientv3: completed snapshot read; closing
Snapshot saved at etcd_backup.db
labsuser@master:~$ ls
DCV-Storage  Downloads  Public      cni-plugins-linux-amd64-v1.1.1.tgz  pod-demo.yaml  role
Desktop      Music      Templates  containerd-1.6.8-linux-amd64.tar.gz  pv-demo.yaml   runc.amd64
Documents    Pictures   Videos    etcd_backup.db                       pvc-demo.yaml  snap
labsuser@master:~$ sudo ETCDCTL_API=3 etcdctl \
> --endpoints $advertise_url \
> --cacert /etc/kubernetes/pki/etcd/ca.crt \
> --key /etc/kubernetes/pki/etcd/server.key \
> --cert /etc/kubernetes/pki/etcd/server.crt snapshot restore etcd_backup.db
2023-10-10 10:28:06.881512 I | mvcc: restore compact to 24589
2023-10-10 10:28:06.897968 I | etcdserver/membership: added member 8e9e05c52164694d [http://localhost:2380] to cluster cdf818194e3a8c32
labsuser@master:~$
```

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
```

```
labsuser@master:~$ stat -c %U:%G /var/lib/etcd
root:root
labsuser@master:~$ sudo chown -R root:root /var/lib/etcd
labsuser@master:~$
```

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
```

```
labsuser@master:~$ sudo ETCDCTL_API=3 etcdctl endpoint health \
> --endpoints=$advertise_url \
> --cacert=/etc/kubernetes/pki/etcd/ca.crt \
> --cert=/etc/kubernetes/pki/etcd/server.crt \
> --key=/etc/kubernetes/pki/etcd/server.key
https://172.31.47.175:2379 is healthy: successfully committed proposal: took = 7.406101ms
labsuser@master:~$
```

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