**scenario (backup etcd database)**

**Task : Backup etcd database**

To backup the etcd data of a Kubernetes master node initialized with `kubeadm`, you can use the `etcdctl` command. Here's a step-by-step guide:

**solution (backup etcd database)**

1. \*\*Fetch and extract the tarball\*\*:

wget https://github.com/etcd-io/etcd/releases/download/v3.5.0/etcd-v3.5.0-linux-amd64.tar.gz

tar xvf etcd-v3.5.0-linux-amd64.tar.gz

2. \*\*Move the `etcdctl` binary to a directory in your PATH\*\*:

sudo mv etcd-v3.5.0-linux-amd64/etcdctl /usr/local/bin/

3. \*\*Clean up\*\*:

rm -r etcd-v3.5.0-linux-amd64

rm etcd-v3.5.0-linux-amd64.tar.gz

4. \*\*Verify the installation\*\*:

etcdctl version

This should display the version of `etcdctl` you just installed.

Remember to always check the official documentation or the project's GitHub page for the most up-to-date and detailed instructions.

1. \*\*Set environment variables\*\*:

When using `kubeadm`, the etcd cluster is typically secured using client certificates. The necessary credentials and endpoints are found in the etcd pod manifest on the master node. You can set them as environment variables to make the `etcdctl` commands easier:

export ETCDCTL\_API=3

export ETCD\_CERT\_DIR=/etc/kubernetes/pki/etcd

export ETCDCTL\_CACERT=${ETCD\_CERT\_DIR}/ca.crt

export ETCDCTL\_CERT=${ETCD\_CERT\_DIR}/peer.crt

export ETCDCTL\_KEY=${ETCD\_CERT\_DIR}/peer.key

export ETCDCTL\_ENDPOINTS=https://127.0.0.1:2379

# Notes CA certificate is ca.crt && client certificate is peer.crt & client key is peerkey &&

2. \*\*Backup etcd data\*\*:

Use the `etcdctl snapshot save` command to backup etcd data:

etcdctl snapshot save /home/ubuntu/snapshot.db

Or alternative command for above export

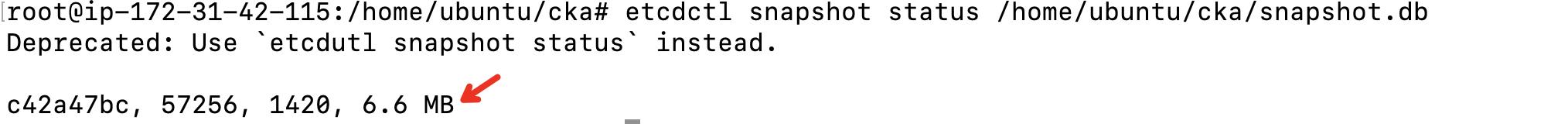
ETCDCTL\_API=3 etcdctl --endpoints=127.0.0.1:2379 --cacert=/etc/kubernetes/pki/etcd/ca.crt --cert=/etc/kubernetes/pki/etcd/peer.crt --key=/etc/kubernetes/pki/etcd/peer.key snapshot save /home/ubuntu/snapshot.db

Replace `/path/to/backup/` with your desired backup directory. This will create a backup file named `snapshot.db` in that directory.

3. \*\*Verify the backup\*\*:

It's a good practice to verify the integrity of the backup:

etcdctl snapshot status /home/ubuntu/snapshot.db



4. \*\*restore the backup\*\*:

etcdctl snapshot restore /home/ubuntu/snapshot.db