Step-by-Step Guide

Step 1: Access the etcd Pod

etcd runs on the control plane nodes in your Kubernetes cluster. To back it up, you need to access the etcd pod.

Find the etcd Pod:

```
kubectl get pods -n kube-system -l component=etcd
```

- This command lists the etcd pod(s) running in your cluster.

Access the etcd Pod:

```
kubectl exec -it <etcd-pod-name> -n kube-system -- /bin/sh
```

- Replace `<etcd-pod-name>` with the actual pod name from the previous step.

Step 2: Back Up etcd

Once inside the etcd pod, you can create a snapshot of the etcd database.

Create an etcd Snapshot:

```
ETCDCTL_API=3 etcdctl snapshot save /var/lib/etcd/snapshot.db \
  --endpoints=https://127.0.0.1:2379 \
  --cacert=/etc/kubernetes/pki/etcd/ca.crt \
  --cert=/etc/kubernetes/pki/etcd/server.crt \
  --key=/etc/kubernetes/pki/etcd/server.key
```

- This command saves a snapshot of etcd to the file `/var/lib/etcd/snapshot.db`.

Copy the Snapshot to a Safe Location:

Exit the etcd pod and copy the snapshot file to a safe location:

```
kubectl cp kube-system/<etcd-pod-name>:/var/lib/etcd/snapshot.db
./snapshot.db
```

- This copies the snapshot file to your local machine.

Step 3: Restore etcd from a Backup

In case of data corruption or loss, you can restore etcd from the backup snapshot.

Stop the Kubernetes API Server:

```
sudo systemctl stop kube-apiserver
```

- This stops the Kubernetes API server to prevent any new writes during the restoration.

Restore the etcd Snapshot:
On the control plane node, run:

```
ETCDCTL_API=3 etcdctl snapshot restore snapshot.db \
  --data-dir /var/lib/etcd \
  --initial-cluster <etcd-node-name>=https://127.0.0.1:2380 \
  --initial-advertise-peer-urls https://127.0.0.1:2380
```

- Replace `<etcd-node-name>` with the name of your etcd node.

Restart the Kubernetes API Server:
After restoring etcd, restart the API server:

```
sudo systemctl start kube-apiserver
```

- This will bring your Kubernetes cluster back online with the restored etcd data.

Step 4: Automate etcd Backups

Regular backups of etcd can be automated using a CronJob in Kubernetes.

Create a CronJob for etcd Backups:

```
apiVersion: batch/v1
kind: CronJob
metadata:
  name: etcd-backup
  namespace: kube-system
spec:
  schedule: "0 3 * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: etcd-backup
            image: bitnami/etcd:latest
            command:
            - /bin/sh
            - -c
            - |
              ETCDCTL_API=3 etcdctl snapshot save /snapshot/snapshot-$(date +\%F-\%T).db \
              --endpoints=$(ETCD_ENDPOINTS) \
              --cacert=/etc/kubernetes/pki/etcd/ca.crt \
              --cert=/etc/kubernetes/pki/etcd/server.crt \
              --key=/etc/kubernetes/pki/etcd/server.key
            env:
            - name: ETCD_ENDPOINTS
              value: "https://<etcd-service>:2379"
            volumeMounts:
            - name: etcd-backup
              mountPath: /snapshot
            - name: etcd-certs
              mountPath: /etc/kubernetes/pki/etcd
              readOnly: true
          restartPolicy: OnFailure
          volumes:
          - name: etcd-backup
            hostPath:
              path: /var/lib/etcd-backup
              type: DirectoryOrCreate
          - name: etcd-certs
            hostPath:
              path: /etc/kubernetes/pki/etcd
              type: Directory
```

- This CronJob runs every day at 3 AM and saves etcd snapshots to the `/var/lib/etcd-backup` directory.