Kubernetes mentés, helyreállítás

Control plane

Mentés

ETCD mentés

K8S konfiguráció mentés

Egy control plane kiesés

Kettő control plane kiesés

Három control plane kiesés

Worker

ETCD környezet helyreállítása mentésből

A környezet helyreállítása előtt győződjünk meg arról, hogy minden control-plane gépen telepítve van az etcdctl és az etcdutl program.

```
# etcdctl version
etcdctl version: 3.6.12
API version: 3.6

# etcdutl version
etcdutl version: 3.6.12
API version: 3.6
```

Megjegyzés: 3.5-ös `etcdctl` verzió az utolsó olyan változat, amellyel a helyreállítás elvégezhető. 3.6-os verziótól a helyreállítási műveleteket az `etcdutl` utasítással kell elvégezni.

Mentés állapotának ellenőrzése

Állapot gyors ellenőrzése

```
# etcdutl snapshot status /root/etcd-snapshot-20260606.db -w table
```

```
+----------+----------+------------+------------+---------+
|   HASH   | REVISION | TOTAL KEYS | TOTAL SIZE | VERSION |
+----------+----------+------------+------------+---------+
| e1ac0dbb | 17572986 |        749 |      24 MB |   3.6.0 |
+----------+----------+------------+------------+---------+
```

Helyreállítási teszt

Győződjünk meg a mentés helyreállíthatóságáról. A mentést valamelyik control-plane gépen egy átmeneti könyvtárban állítsuk helyre.

```
# rm -rf /tmp/etcd-restore-test
etcdutl snapshot restore /root/etcd-snapshot-20260606.db --data-
dir=/tmp/etcd-restore-test
2026-07-06T08:54:44+02:00   info    snapshot/v3_snapshot.go:305    restoring
snapshot    {"path": "/root/etcd-snapshot-20260606.db", "wal-dir":
"/tmp/etcd-restore-test/member/wal", "data-dir": "/tmp/etcd-restore-test",
"snap-dir": "/tmp/etcd-restore-test/member/snap", "initial-memory-map-size":
10737418240}
2026-07-06T08:54:44+02:00   info    bbolt    backend/backend.go:203
Opening db file (/tmp/etcd-restore-test/member/snap/db) with mode -rw-------
and with options: {Timeout: 0s, NoGrowSync: false, NoFreelistSync: true,
PreLoadFreelist: false, FreelistType: , ReadOnly: false, MmapFlags: 8000,
InitialMmapSize: 10737418240, PageSize: 0, NoSync: false, OpenFile: 0x0,
Mlock: false, Logger: 0xc000304008}
2026-07-06T08:54:44+02:00   info    bbolt    bbolt@v1.4.3/db.go:322
Opening bbolt db (/tmp/etcd-restore-test/member/snap/db) successfully
2026-07-06T08:54:44+02:00   info    schema/membership.go:138    Trimming
membership information from the backend...
2026-07-06T08:54:44+02:00   info    bbolt    backend/backend.go:203
Opening db file (/tmp/etcd-restore-test/member/snap/db) with mode -rw-------
and with options: {Timeout: 0s, NoGrowSync: false, NoFreelistSync: true,
PreLoadFreelist: false, FreelistType: , ReadOnly: false, MmapFlags: 8000,
InitialMmapSize: 10737418240, PageSize: 0, NoSync: false, OpenFile: 0x0,
Mlock: false, Logger: 0xc0000621e0}
2026-07-06T08:54:44+02:00   info    bbolt    bbolt@v1.4.3/db.go:322
Opening bbolt db (/tmp/etcd-restore-test/member/snap/db) successfully
2026-07-06T08:54:44+02:00   info    membership/cluster.go:424    added
member    {"cluster-id": "cdf818194e3a8c32", "local-member-id": "0", "added-
peer-id": "8e9e05c52164694d", "added-peer-peer-urls":
["http://localhost:2380"], "added-peer-is-learner": false}
2026-07-06T08:54:45+02:00   info    bbolt    backend/backend.go:203
Opening db file (/tmp/etcd-restore-test/member/snap/db) with mode -rw-------
and with options: {Timeout: 0s, NoGrowSync: false, NoFreelistSync: true,
PreLoadFreelist: false, FreelistType: , ReadOnly: false, MmapFlags: 8000,
InitialMmapSize: 10737418240, PageSize: 0, NoSync: false, OpenFile: 0x0,
Mlock: false, Logger: 0xc000062228}
2026-07-06T08:54:45+02:00   info    bbolt    bbolt@v1.4.3/db.go:322
Opening bbolt db (/tmp/etcd-restore-test/member/snap/db) successfully
```

```
2026-07-06T08:54:45+02:00   info    snapshot/v3_snapshot.go:333    restored
snapshot    {"path": "/root/etcd-snapshot-20260606.db", "wal-dir":
"/tmp/etcd-restore-test/member/wal", "data-dir": "/tmp/etcd-restore-test",
"snap-dir": "/tmp/etcd-restore-test/member/snap", "initial-memory-map-size":
10737418240}
```

Amennyiben sikeresen lefut a helyreállítás, akkor a teszt sikeres.

Jelenlegi állapot mentése

Kubernetes objektumok állapotának mentése egy control-plane gépen. Erre a helyreállítás után az objektumok állapotának ellenőrzésénél lesz szükség.

```
# mkdir -p /root/k8s-backup
# kubectl get nodes -o wide > /root/k8s-backup/nodes.before.txt
# kubectl get pods -A -o wide > /root/k8s-backup/pods.before.txt
# kubectl get ns > /root/k8s-backup/namespaces.before.txt
# kubectl get cm -A > /root/k8s-backup/configmaps.before.txt
```

Szolgáltatások tiltása és leállítása az összes control-plane gépen.

```
# systemctl disable kubelet containerd
# systemctl stop kubelet containerd
# systemctl reboot
<code>

Korábbi környezet mentése.

<code>
# mkdir -p /root/k8s-backup/pre-restore
# mkdir -p /root/k8s-backup/pre-restore/etc/kubernetes
# mkdir -p /root/k8s-backup/pre-restore/var/lib
# cp -a /etc/kubernetes/manifests /root/k8s-backup/pre-
restore/etc/kubernetes/
# cp -a /etc/kubernetes/pki /root/k8s-backup/pre-restore/etc/kubernetes/
# cp -a /var/lib/etcd /root/k8s-backup/pre-restore/var/lib/
```

From:
https://wiki.r-l.hu/ - **Reverse-Logic wiki**

Permanent link:
https://wiki.r-l.hu/doku.php?id=kubernetes:mentes-helyreallitas&rev=1783575541

Last update: **2026/07/09 05:39**