Kubernetes mentés, helyreállítás

Dokumentum verzió	1.00
Dokumentum kelte	2026.07.09
Kubernetes verzió	1.34 - 1.36

Control plane

A control plane gépek helyreállítása előtt győződjünk meg arról, hogy a mentések rendelkezésre állnak. A gépek konfigurációját tartalmazó mentés csak azon a gépen állítható helyre, ahol készült!

Konfiguráció mentése

A mentés az alábbi állományokat tartalmazza:

- /etc/cni
- /etc/containerd
- /etc/kubernetes/manifests/*.yaml
- /etc/kubernetes/pki
- /etc/kubernetes/*.conf
- /root/kubeadm-config.yaml
- /var/lib/kubelet/pki
- /var/lib/kubelet/config.yaml
- /var/lib/kubelet/instance-config.yaml
- /var/lib/kubelet/kubeadm-flags.env

A `/var/lib/kubelet/kubeadm-flags.env` fájlban `kubernetes v1.35` verziótól nincs inicializálási paraméter, ezért kihagyható a mentésből.

A mentést `root` felhasználóként kell futtatni. az alábbi példa napi mentést készít a helyi gép /var/backup alkönvtárába.

```
# mkdir -p /var/backup && tar -czvf /var/backup/k8s-config-$(hostname)-$(date +'%Y%m%d').tar.gz /etc/cni /etc/containerd /etc/kubernetes/manifests/*.yaml /etc/kubernetes/pki /etc/kubernetes/*.conf /root/kubeadm-config.yaml /var/lib/kubelet/pki /var/lib/kubelet/config.yaml /var/lib/kubelet/instance-config.yaml /var/lib/kubeadm-flags.env
```

ETCD mentés

Az ETCD adatbázisáról egyetlen control plane gépen kell snapshot-ot készíteni. A snapshot `root` felhasználóként az `etcdctl` utasítással készíthető el. A snapshot készítése után visszaállítási tesztet kell végezni. Az ETCD 3.6-os verziójától kezdve a helyreállítási műveleteket az `etcdutl` utasítással tudjuk végrehajtani.

```
# etcdctl version
```

```
etcdctl version: 3.6.12
API version: 3.6

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

```
# mkdir -p /var/backup/ && etcdctl snapshot save /var/backup/etcd-snapshot-
$(hostname)-$(date +'%Y%m%d').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
...
Snapshot saved at /var/backup/etcd-snapshot-HOSTNAME-YYYYMMDD.db
Server version 3.6.0
```

Snapshot állapot ellenőrzése.

```
# etcdutl snapshot status /var/backup/etcd-snapshot-HOSTNAME-YYYYMMDD.db -w
table
+----------+----------+------------+------------+---------+
|   HASH   | REVISION | TOTAL KEYS | TOTAL SIZE | VERSION |
+----------+----------+------------+------------+---------+
| aac67e22 | 18385870 |        516 |      24 MB |   3.6.0 |
+----------+----------+------------+------------+---------+
```

Egy átmeneti könyvtárban helyreállítási tesztet futtatva győződhetünk meg a snapshot állapotáról.

```
# rm -rf /tmp/etcd-restore-test-$(date +'%Y%m%d') && etcdutl snapshot
restore /var/backup/etcd-snapshot-$(hostname)-$(date +'%Y%m%d').db --data-
dir=/tmp/etcd-restore-test-$(date +'%Y%m%d')

# [ $? -eq 0 ] && echo "OK - Sikeres helyreállítás" || echo "FAILED -
Sikertelen helyreállítás"
OK - Sikeres helyreállítás

# rm -rf /tmp/etcd-restore-test-$(date +'%Y%m%d')
```

Állapot ellenőrzése

Az alábbi utasításokkal ellenőrizhető a klaszter állapota.

```
# kubectl get nodes -l node-role.kubernetes.io/control-plane
NAME     STATUS   ROLES           AGE   VERSION
kube01   Ready    control-plane   66d   v1.36.2
kube02   Ready    control-plane   66d   v1.36.2
kube03   Ready    control-plane   66d   v1.36.2
```

Mindhárom node estében érdemes ellenőrizni a működési feltételeket.

```
# for node in kube01 kube02 kube03 ; do echo "### ${node} állapot ###" ;
kubectl describe node ${node} | egrep
'(Ready|MemoryPressure|DiskPressure|PIDPressure|NetworkUnavailable)' | awk
'{print $1": "$2}' ; echo ; done
### kube01 állapot ###
NetworkUnavailable: False
MemoryPressure: False
DiskPressure: False
PIDPressure: False
Ready: True

### kube02 állapot ###
NetworkUnavailable: False
MemoryPressure: False
DiskPressure: False
PIDPressure: False
Ready: True

### kube03 állapot ###
NetworkUnavailable: False
MemoryPressure: False
DiskPressure: False
PIDPressure: False
Ready: True
```

Ellenőrizzük a POD-ok állapotát.

```
# kubectl -n kube-system get pods
NAME                              READY   STATUS    RESTARTS       AGE
coredns-589f44dc88-5ppfq          1/1     Running   7 (46h ago)    66d
coredns-589f44dc88-nmvss          1/1     Running   7 (46h ago)    66d
etcd-kube01                       1/1     Running   14 (47h ago)   66d
etcd-kube02                       1/1     Running   2 (46h ago)    21d
etcd-kube03                       1/1     Running   2 (46h ago)    21d
kube-apiserver-kube01             1/1     Running   31 (47h ago)   21d
kube-apiserver-kube02             1/1     Running   2 (46h ago)    21d
kube-apiserver-kube03             1/1     Running   5 (46h ago)    21d
kube-controller-manager-kube01    1/1     Running   5 (47h ago)    21d
kube-controller-manager-kube02    1/1     Running   7 (46h ago)    21d
kube-controller-manager-kube03    1/1     Running   4 (46h ago)    21d
kube-proxy-5v48r                  1/1     Running   2 (46h ago)    21d
kube-proxy-hj68x                  1/1     Running   2 (47h ago)    21d
kube-proxy-xrk5x                  1/1     Running   2 (46h ago)    21d
kube-scheduler-kube01             1/1     Running   7 (47h ago)    21d
kube-scheduler-kube02             1/1     Running   4 (46h ago)    21d
kube-scheduler-kube03             1/1     Running   5 (46h ago)    21d
metrics-server-55bf4495db-tsgtq   1/1     Running   6 (46h ago)    26d
```

Áttekintő ellenőrzés.

```
# kubectl get --raw='/readyz?verbose'
[+]ping ok
[+]log ok
[+]etcd ok
[+]etcd-readiness ok
[+]informer-sync ok
[+]poststarthook/start-apiserver-admission-initializer ok
[+]poststarthook/generic-apiserver-start-informers ok
[+]poststarthook/priority-and-fairness-config-consumer ok
[+]poststarthook/priority-and-fairness-filter ok
[+]poststarthook/storage-object-count-tracker-hook ok
[+]poststarthook/start-apiextensions-informers ok
[+]poststarthook/start-apiextensions-controllers ok
[+]poststarthook/crd-informer-synced ok
[+]poststarthook/start-system-namespaces-controller ok
[+]poststarthook/peer-endpoint-reconciler-controller ok
[+]poststarthook/start-cluster-authentication-info-controller ok
[+]poststarthook/start-kube-apiserver-identity-lease-controller ok
[+]poststarthook/start-kube-apiserver-identity-lease-garbage-collector ok
[+]poststarthook/storage-readiness ok
[+]poststarthook/start-legacy-token-tracking-controller ok
[+]poststarthook/start-service-ip-repair-controllers ok
[+]poststarthook/rbac/bootstrap-roles ok
[+]poststarthook/scheduling/bootstrap-system-priority-classes ok
[+]poststarthook/priority-and-fairness-config-producer ok
[+]poststarthook/bootstrap-controller ok
[+]poststarthook/start-kubernetes-service-cidr-controller ok
[+]poststarthook/aggregator-reload-proxy-client-cert ok
[+]poststarthook/start-kube-aggregator-informers ok
[+]poststarthook/apiservice-status-local-available-controller ok
[+]poststarthook/apiservice-status-remote-available-controller ok
[+]poststarthook/apiservice-registration-controller ok
[+]poststarthook/apiservice-discovery-controller ok
[+]poststarthook/kube-apiserver-autoregistration ok
[+]autoregister-completion ok
[+]poststarthook/apiservice-openapi-controller ok
[+]poststarthook/apiservice-openapiv3-controller ok
[+]shutdown ok
readyz check passed
```

ETCD tagok listája.

```
# etcdctl --
endpoints=https://kube01:2379,https://kube02:2379,https://kube03:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key member list -w table
+------------------+---------+--------+------------------------------+------
-----------------------+------------+
|        ID        | STATUS  |  NAME  |          PEER ADDRS          |
```

```
CLIENT ADDRS         | IS LEARNER |
+------------------+---------+--------+-----------------------------+------
-----------------------+------------+
| 1546a2d2ac61bd8d | started | kube02 | https://192.168.110.162:2380 |
https://192.168.110.162:2379 |      false |
| 18e2fb4d02a30b2e | started | kube03 | https://192.168.110.163:2380 |
https://192.168.110.163:2379 |      false |
| 20e2ce3086b2e277 | started | kube01 | https://192.168.110.161:2380 |
https://192.168.110.161:2379 |      false |
+------------------+---------+--------+-----------------------------+------
-----------------------+------------+
```

ETCD endpoint állapot.

```
# etcdctl --
endpoints=https://kube01:2379,https://kube02:2379,https://kube03:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key endpoint status -w table
+---------------------+------------------+---------+-----------------+------
---+--------+-----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+-------------------------
+-------------------+
|      ENDPOINT       |        ID        | VERSION | STORAGE VERSION | DB
SIZE | IN USE | PERCENTAGE NOT IN USE | QUOTA  | IS LEADER | IS LEARNER |
RAFT TERM | RAFT INDEX | RAFT APPLIED INDEX | ERRORS | DOWNGRADE TARGET
VERSION | DOWNGRADE ENABLED |
+---------------------+------------------+---------+-----------------+------
---+--------+-----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+-------------------------
+-------------------+
| https://kube01:2379 | 20e2ce3086b2e277 |   3.6.8 |           3.6.0 |   24
MB | 7.2 MB |                   71% | 2.1 GB |     false |      false |
4 |    6990447 |           6990447 |        |                           |
false |
| https://kube02:2379 | 1546a2d2ac61bd8d |   3.6.8 |           3.6.0 |   24
MB | 7.2 MB |                   71% | 2.1 GB |      true |      false |
4 |    6990447 |           6990447 |        |                           |
false |
| https://kube03:2379 | 18e2fb4d02a30b2e |   3.6.8 |           3.6.0 |   24
MB | 7.2 MB |                   71% | 2.1 GB |     false |      false |
4 |    6990447 |           6990447 |        |                           |
false |
+---------------------+------------------+---------+-----------------+------
---+--------+-----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+-------------------------
+-------------------+
```

Tanúsítványok érvényességének vizsgálata.

```
# kubeadm certs check-expiration
```

```
[check-expiration] Reading configuration from the "kubeadm-config" ConfigMap
in namespace "kube-system"...
[check-expiration] Use 'kubeadm init phase upload-config kubeadm --config
your-config-file' to re-upload it.

CERTIFICATE                EXPIRES                  RESIDUAL TIME
CERTIFICATE AUTHORITY   EXTERNALLY MANAGED
admin.conf                 Feb 15, 2126 06:54 UTC   99y             ca
no
apiserver                  Feb 15, 2126 06:54 UTC   99y             ca
no
apiserver-etcd-client      Feb 15, 2126 06:54 UTC   99y             etcd-ca
no
apiserver-kubelet-client   Feb 15, 2126 06:54 UTC   99y             ca
no
controller-manager.conf    Feb 15, 2126 06:54 UTC   99y             ca
no
etcd-healthcheck-client    Feb 15, 2126 06:54 UTC   99y             etcd-ca
no
etcd-peer                  Feb 15, 2126 06:54 UTC   99y             etcd-ca
no
etcd-server                Feb 15, 2126 06:54 UTC   99y             etcd-ca
no
front-proxy-client         Feb 15, 2126 06:54 UTC   99y             front-
proxy-ca          no
scheduler.conf             Feb 15, 2126 06:54 UTC   99y             ca
no
super-admin.conf           Feb 15, 2126 06:54 UTC   99y             ca
no

CERTIFICATE AUTHORITY   EXPIRES                  RESIDUAL TIME   EXTERNALLY
MANAGED
ca                      Sep 01, 2225 09:27 UTC   199y            no
etcd-ca                 Sep 01, 2225 09:27 UTC   199y            no
front-proxy-ca          Sep 01, 2225 09:27 UTC   199y            no
```

Amennyiben a fenti ellenőrzések sikeresek, akkor a klaszter control plane gépei egészségesek. Nincs további teendő.

Egy control plane kiesése

Egy control plane gép kiesését a klaszter probléma nélkül elviseli. Amennyiben a kiesés az ETCD leader-t érinti, akkor a két megmaradt tagból az egyik átveszi a leader szerepet.

A leader szerepkörű ETCD tagot az alábbi utasítással kaphatjuk meg.

```
# etcdctl --
endpoints=https://kube01:2379,https://kube02:2379,https://kube03:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --
```

```
cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key endpoint status -w json | jq -r '.[]
| select(.Status.leader == .Status.header.member_id) | .Endpoint'

https://kube02:2379
```

A leader kiesésekor az ETCD ellenőrző utasítás nem tud kapcsolódni a korábbi leader-hez. A két működő tag új leader-t választ.

```
# etcdctl --
endpoints=https://kube01:2379,https://kube02:2379,https://kube03:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key endpoint status -w json | jq -r '.[]
| select(.Status.leader == .Status.header.member_id) | .Endpoint'
{"level":"warn","ts":"2026-07-09T09:24:07.518541+0200","logger":"etcd-
client","caller":"v3@v3.6.12/retry_interceptor.go:68","msg":"retrying of
unary invoker failed","target":"etcd-
endpoints://0xc0002103c0/kube02:2379","peer":"Peer{Addr: <nil>, LocalAddr:
<nil>, AuthInfo:
<nil>}","method":"/etcdserverpb.Maintenance/Status","attempt":0,"error":"rpc
error: code = DeadlineExceeded desc = latest balancer error: connection
error: desc = \"transport: Error while dialing: dial tcp
192.168.110.162:2379: connect: connection refused\""}
Failed to get the status of endpoint https://kube02:2379 (context deadline
exceeded)

https://kube01:2379
```

A gépek listázásakor egy idő után a hiányzó gép NotReady állapotba kerül.

```
# kubectl get nodes -l node-role.kubernetes.io/control-plane
NAME     STATUS     ROLES           AGE   VERSION
kube01   Ready      control-plane   66d   v1.36.2
kube02   NotReady   control-plane   66d   v1.36.2
kube03   Ready      control-plane   66d   v1.36.2
```

ETCD endpoint állapot ellenőrzésekor hiányzik a kiesett tag.

```
# etcdctl --
endpoints=https://kube01:2379,https://kube02:2379,https://kube03:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key endpoint status -w table
{"level":"warn","ts":"2026-07-09T09:34:48.926554+0200","logger":"etcd-
client","caller":"v3@v3.6.12/retry_interceptor.go:68","msg":"retrying of
unary invoker failed","target":"etcd-
endpoints://0xc00027f860/kube02:2379","peer":"Peer{Addr: <nil>, LocalAddr:
<nil>, AuthInfo:
<nil>}","method":"/etcdserverpb.Maintenance/Status","attempt":0,"error":"rpc
error: code = DeadlineExceeded desc = context deadline exceeded while
```

```
waiting for connections to become ready"}
Failed to get the status of endpoint https://kube02:2379 (context deadline
exceeded)
+---------------------+------------------+---------+-----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+-------------------+--------+-------------------------
+-------------------+
|      ENDPOINT       |        ID        | VERSION | STORAGE VERSION | DB
SIZE | IN USE | PERCENTAGE NOT IN USE | QUOTA  | IS LEADER | IS LEARNER |
RAFT TERM | RAFT INDEX | RAFT APPLIED INDEX | ERRORS | DOWNGRADE TARGET
VERSION | DOWNGRADE ENABLED |
+---------------------+------------------+---------+-----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+-------------------+--------+-------------------------
+-------------------+
| https://kube01:2379 | 20e2ce3086b2e277 |   3.6.8 |           3.6.0 |   24
MB | 8.0 MB |                   68% | 2.1 GB |      true |      false |
5 |    6996663 |           6996663 |        |                           |
false |
| https://kube03:2379 | 18e2fb4d02a30b2e |   3.6.8 |           3.6.0 |   24
MB | 8.0 MB |                   67% | 2.1 GB |     false |      false |
5 |    6996681 |           6996678 |        |                           |
false |
+---------------------+------------------+---------+-----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+-------------------+--------+-------------------------
+-------------------+
```

Rövid idejű kiesés

Rövid ideig tartó kiesés esetén a hiba megszűnése után érdemes ellenőrizni az állapotokat.

Gépek listája.

```
# kubectl get nodes -l node-role.kubernetes.io/control-plane
NAME     STATUS   ROLES           AGE   VERSION
kube01   Ready    control-plane   66d   v1.36.2
kube02   Ready    control-plane   66d   v1.36.2
kube03   Ready    control-plane   66d   v1.36.2
```

ETCD endpoint állapot.

```
# etcdctl --
endpoints=https://kube01:2379,https://kube02:2379,https://kube03:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key endpoint status -w table
+---------------------+------------------+---------+-----------------+------
---+--------+----------------------+--------+-----------+------------+-----
```

```
------+------------+--------------------+--------+--------------------------
+-------------------+
|      ENDPOINT       |        ID        | VERSION | STORAGE VERSION | DB
SIZE | IN USE | PERCENTAGE NOT IN USE | QUOTA  | IS LEADER | IS LEARNER |
RAFT TERM | RAFT INDEX | RAFT APPLIED INDEX | ERRORS | DOWNGRADE TARGET
VERSION | DOWNGRADE ENABLED |
+---------------------+------------------+---------+-----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+--------------------------
+-------------------+
| https://kube01:2379 | 20e2ce3086b2e277 |   3.6.8 |           3.6.0 |   24
MB | 8.4 MB |                   66% | 2.1 GB |      true |      false |
5 |    6996862 |            6996862 |        |                           |
false |
| https://kube02:2379 | 1546a2d2ac61bd8d |   3.6.8 |           3.6.0 |   24
MB | 8.0 MB |                   68% | 2.1 GB |     false |      false |
5 |    6996863 |            6996861 |        |                           |
false |
| https://kube03:2379 | 18e2fb4d02a30b2e |   3.6.8 |           3.6.0 |   24
MB | 8.3 MB |                   66% | 2.1 GB |     false |      false |
5 |    6996864 |            6996863 |        |                           |
false |
+---------------------+------------------+---------+-----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+--------------------------
+-------------------+
```

Újabb ellenőrzéssel meggyőződhetünk a RAFT indexek állapotáról.

```
# etcdctl --
endpoints=https://kube01:2379,https://kube02:2379,https://kube03:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key endpoint status -w table
+---------------------+------------------+---------+-----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+--------------------------
+-------------------+
|      ENDPOINT       |        ID        | VERSION | STORAGE VERSION | DB
SIZE | IN USE | PERCENTAGE NOT IN USE | QUOTA  | IS LEADER | IS LEARNER |
RAFT TERM | RAFT INDEX | RAFT APPLIED INDEX | ERRORS | DOWNGRADE TARGET
VERSION | DOWNGRADE ENABLED |
+---------------------+------------------+---------+-----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+--------------------------
+-------------------+
| https://kube01:2379 | 20e2ce3086b2e277 |   3.6.8 |           3.6.0 |   24
MB | 9.8 MB |                   60% | 2.1 GB |      true |      false |
5 |    6998333 |            6998333 |        |                           |
false |
| https://kube02:2379 | 1546a2d2ac61bd8d |   3.6.8 |           3.6.0 |   24
```

```
MB | 9.6 MB |                   61% | 2.1 GB |     false |      false |
5 |    6998333 |           6998333 |        |                          |
false |
| https://kube03:2379 | 18e2fb4d02a30b2e |   3.6.8 |           3.6.0 |   24
MB | 9.8 MB |                   60% | 2.1 GB |     false |      false |
5 |    6998333 |           6998333 |        |                          |
false |
+---------------------+------------------+---------+-----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+-------------------------
+-------------------+
```

Tartós kiesés

Egy control plane node tartós kiesése esetén az ETCD tömöríti a RAFT naplókat, így azok alkalmazására már nincs lehetőség. Ilyenkor az alábbi lehetőségek vannak.

Automatikus helyreállás

Ebben az esetben automatikus helyreállás történik egy friss leader snapshot elküldésével. Ellenőrzéseken kívül nincs további teendő.

Helyreállítás ugyanazon a gépen

A legtisztább eljárás, hogy a két megmaradó gép egyikén töröljük az ETCD tagok és a gépek listájából a gépet. Fontos, hogy a törlegdő gépen ne fusson a `containerd` és a `kubelet` szolgáltatás valamint a kuberneteshez kapcsolódó statikus pod-ok sem. Ezt legegyszerűbben a gép leállításval tudjuk megoldani.

Ellenőrizzük az ETCD tagok listáját.

```
# etcdctl --endpoints=https://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 member list -w table
+------------------+---------+--------+------------------------------+------
-----------------------+------------+
|        ID        | STATUS  |  NAME  |          PEER ADDRS          |
CLIENT ADDRS         | IS LEARNER |
+------------------+---------+--------+------------------------------+------
-----------------------+------------+
| 1546a2d2ac61bd8d | started | kube02 | https://192.168.110.162:2380 |
https://192.168.110.162:2379 |      false |
| 18e2fb4d02a30b2e | started | kube03 | https://192.168.110.163:2380 |
https://192.168.110.163:2379 |      false |
| 20e2ce3086b2e277 | started | kube01 | https://192.168.110.161:2380 |
```

```
https://192.168.110.161:2379 |      false |
+------------------+---------+--------+------------------------------+------
-----------------------+------------+
```

Távolítsuk el a hibás tagot az azonosító megadásával (ID mező).

```
# etcdctl --endpoints=https://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 member remove 18e2fb4d02a30b2e
Member 18e2fb4d02a30b2e removed from cluster adeb3c2ca3243377
```

Újra ellenőrizzük az ETCD tagok listáját.

```
# etcdctl --endpoints=https://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 member list -w table
+------------------+---------+--------+------------------------------+------
-----------------------+------------+
|        ID        | STATUS  |  NAME  |          PEER ADDRS          |
CLIENT ADDRS         | IS LEARNER |
+------------------+---------+--------+------------------------------+------
-----------------------+------------+
| 1546a2d2ac61bd8d | started | kube02 | https://192.168.110.162:2380 |
https://192.168.110.162:2379 |      false |
| 20e2ce3086b2e277 | started | kube01 | https://192.168.110.161:2380 |
https://192.168.110.161:2379 |      false |
+------------------+---------+--------+------------------------------+------
-----------------------+------------+
```

Ellenőrizzük a kubernetes gépek listáját.

```
# kubectl get nodes -l node-role.kubernetes.io/control-plane
NAME     STATUS     ROLES           AGE   VERSION
kube01   Ready      control-plane   66d   v1.36.2
kube02   Ready      control-plane   66d   v1.36.2
kube03   NotReady   control-plane   66d   v1.36.2
```

Töröljük a hibás gépet.

```
# kubectl delete node kube03
node "kube03" deleted
```

Ellenőrizzük a gépek listáját.

```
# kubectl get nodes
NAME     STATUS   ROLES           AGE   VERSION
kube01   Ready    control-plane   66d   v1.36.2
kube02   Ready    control-plane   66d   v1.36.2
```

Generáljunk új tokent a gép csatlakoztatásához.

```
# kubeadm token create --print-join-command
kubeadm join 192.168.110.160:6443 --token gef9jk.oafnicbu6hgwhhy7 --
discovery-token-ca-cert-hash
sha256:838e04036d629fbf3e152fc6606bf9a143f8e1ca0b0bc00d747fe1caba5df8f1

# kubeadm init phase upload-certs --upload-certs
[upload-certs] Storing the certificates in Secret "kubeadm-certs" in the
"kube-system" Namespace
[upload-certs] Using certificate key:
7aa4b634e4f0986669dbd19b972ec3c14a80e3ab1909a54d26ca58b0b7ec15ac
```

Ha szeretnénk előállítani a control plane csatlakoztatásának utasítását, akkor ezt a megoldást használhatjuk. Az utasítás létrehozza a tokent és a tanúsítvány kulcsot.

```
# JOIN_CMD="$(kubeadm token create --print-join-command)"
# CERT_KEY="$(kubeadm init phase upload-certs --upload-certs | tail -n 1)"
# echo "$JOIN_CMD --control-plane --certificate-key $CERT_KEY"
kubeadm join 192.168.110.160:6443 --token 7q469t.81sbouok0em1pvjo --
discovery-token-ca-cert-hash
sha256:838e04036d629fbf3e152fc6606bf9a143f8e1ca0b0bc00d747fe1caba5df8f1  --
control-plane --certificate-key
10babb9fd52e8ae866eabae323e7f486ae6c5fa26613ab93004738cab8432e87
```

A snapshot-ból helyreállított gépet indítsuk el oly módon, hogy a `containerd` és a `kubelet` szolgáltatások biztosan **NE** induljanak el. Ezt a legegyszerűbben a gép boot folyamatának módosításával érjük el. A GRUB menüjében a kernel paraméterezését egészítsük ki az alábbiak szerint.

`linux ….. **systemd.mask=containerd.service systemd.mask=kubelet.service**`

Az elindult rendszerben tudjuk ellenőrizni.

```
# systemctl status containerd kubelet
○ containerd.service
     Loaded: masked (Reason: Unit containerd.service is masked.)
     Active: inactive (dead)

○ kubelet.service
     Loaded: masked (Reason: Unit kubelet.service is masked.)
    Drop-In: /usr/lib/systemd/system/kubelet.service.d
             └─10-kubeadm.conf
     Active: inactive (dead)
```

Amennyiben snapshot-ból állt helyre a gép, futtassuk a `kubeadm reset` utasítást. Ez törölni fogja a korábbi klaszter állományait.

```
# kubeadm reset
[reset] Reading configuration from the "kubeadm-config" ConfigMap in
```

```
namespace "kube-system"...
[reset] Use 'kubeadm init phase upload-config kubeadm --config your-config-
file' to re-upload it.
...
The reset process does not perform cleanup of CNI plugin configuration,
network filtering rules and kubeconfig files.

For information on how to perform this cleanup manually, please see:
    https://k8s.io/docs/reference/setup-tools/kubeadm/kubeadm-reset/
```

Töröljük a felesleges egyéb állományokat.

```
# rm -rf /etc/cni/net.d/* /root/.kube /var/lib/cni /var/lib/etcd
/var/lib/kubelet
```

Indítsuk újra a gépet.

```
# reboot
```

A gép újraindulása után csatlakoztassuk a gépet control plane szerepkörben a meglevő klaszterhez.

```
# kubeadm join 192.168.110.160:6443 --token gef9jk.oafnicbu6hgwhhy7 --
discovery-token-ca-cert-hash
sha256:838e04036d629fbf3e152fc6606bf9a143f8e1ca0b0bc00d747fe1caba5df8f1 --
control-plane --certificate-key
7aa4b634e4f0986669dbd19b972ec3c14a80e3ab1909a54d26ca58b0b7ec15ac
[preflight] Running pre-flight checks
[preflight] Reading configuration from the "kubeadm-config" ConfigMap in
namespace "kube-system"...
[preflight] Use 'kubeadm init phase upload-config kubeadm --config your-
config-file' to re-upload it.
...

This node has joined the cluster and a new control plane instance was
created:

* Certificate signing request was sent to apiserver and approval was
received.
* The Kubelet was informed of the new secure connection details.
* Control plane label and taint were applied to the new node.
* The Kubernetes control plane instances scaled up.
* A new etcd member was added to the local/stacked etcd cluster.

To start administering your cluster from this node, you need to run the
following as a regular user:

    mkdir -p $HOME/.kube
    sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
    sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
Run 'kubectl get nodes' to see this node join the cluster.
```

Állítsuk be a kubernetes konfigurációt a root felhasználónak.

```
# rm -rf /root/.kube/
# mkdir -p $HOME/.kube
# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
# sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Ellenőrizzük a klaszter állapotát.

```
# kubectl get nodes -l node-role.kubernetes.io/control-plane
NAME     STATUS   ROLES           AGE     VERSION
kube01   Ready    control-plane   67d     v1.36.2
kube02   Ready    control-plane   67d     v1.36.2
kube03   Ready    control-plane   5m21s   v1.36.2
```

Ellenőrizzük az ETCD tagok listáját.

```
# etcdctl --endpoints=https://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 member list -w table
+------------------+---------+--------+-----------------------------+------
------------------------+------------+
|        ID        | STATUS  |  NAME  |          PEER ADDRS         |
CLIENT ADDRS         | IS LEARNER |
+------------------+---------+--------+-----------------------------+------
------------------------+------------+
| 1546a2d2ac61bd8d | started | kube02 | https://192.168.110.162:2380 |
https://192.168.110.162:2379 |      false |
| 20e2ce3086b2e277 | started | kube01 | https://192.168.110.161:2380 |
https://192.168.110.161:2379 |      false |
| 23fb75a79460e3c9 | started | kube03 | https://192.168.110.163:2380 |
https://192.168.110.163:2379 |      false |
+------------------+---------+--------+-----------------------------+------
------------------------+------------+
```

Ellenőrizzük az ETCD endpointok állapotát.

```
# etcdctl --
endpoints=https://kube01:2379,https://kube02:2379,https://kube03:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key endpoint status -w table
+---------------------+------------------+---------+-----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+--------------------------
+-------------------+
|      ENDPOINT       |        ID        | VERSION | STORAGE VERSION | DB
```

```
SIZE | IN USE | PERCENTAGE NOT IN USE | QUOTA  | IS LEADER | IS LEARNER |
RAFT TERM | RAFT INDEX | RAFT APPLIED INDEX | ERRORS | DOWNGRADE TARGET
VERSION | DOWNGRADE ENABLED |
+---------------------+------------------+---------+-----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+-------------------------
+------------------+
| https://kube01:2379 | 20e2ce3086b2e277 |   3.6.8 |           3.6.0 |   24
MB | 7.5 MB |                   70% | 2.1 GB |      true |      false |
5 |    7031342 |           7031342 |        |                           |
false |
| https://kube02:2379 | 1546a2d2ac61bd8d |   3.6.8 |           3.6.0 |   24
MB | 7.5 MB |                   70% | 2.1 GB |     false |      false |
5 |    7031343 |           7031343 |        |                           |
false |
| https://kube03:2379 | 23fb75a79460e3c9 |   3.6.8 |           3.6.0 |   24
MB | 7.5 MB |                   70% | 2.1 GB |     false |      false |
5 |    7031343 |           7031343 |        |                           |
false |
+---------------------+------------------+---------+-----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+-------------------------
+------------------+
```

A helyreállítás sikeres.

Két control plane kiesése

Két control plane gép kiesése esetén a kunernetes klaszter elveszíti a kvórumot. Ebben a helyzetben a worker gépeken futó pod-ok a legutolsó állapotukban maradnak. Nem működik a HPA. Nem lehet módosítani a konfiguráción. Ilyen esetben a control plane gépek mielőbbi helyreállítása szükséges.

Rövid idejű kiesés

Rövid iejű kiesés esetén amennyiben a control plane gépek fájlrendszere nem sérül, akkor a gépek újraindulásakor megtörténik a szinkron. Ilyen esetben ellenőrizni kell a klaszter állapotot.

A következő példában két gép rövid időre leáll (pl.: tápellátás hiba miatt). Ebben az esetben a megszokott ellenőrzési műveletek nem futnak, mivel nincs stabil klaszter állapot.

ETCD endpoint állapot vizsgálata.

```
# etcdctl --endpoints=https://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 member list -w table
{"level":"warn","ts":"2026-07-09T12:23:48.745371+0200","logger":"etcd-
client","caller":"v3@v3.6.12/retry_interceptor.go:68","msg":"retrying of
```

```
unary invoker failed","target":"etcd-
endpoints://0xc00027e960/127.0.0.1:2379","peer":"Peer{Addr:
'127.0.0.1:2379', LocalAddr: '127.0.0.1:45328', AuthInfo:
'tls'}","method":"/etcdserverpb.Cluster/MemberList","attempt":0,"error":"rpc
error: code = DeadlineExceeded desc = context deadline exceeded"}
Error: context deadline exceeded
```

A gépek listázása időtúllépés miatt sikertelen, vagy érvénytelen választ kap.

```
# kubectl get nodes
Error from server (Forbidden): nodes is forbidden: User "kubernetes-admin"
cannot list resource "nodes" in API group "" at the cluster scope
```

Újra kell indítani a kiesett gépeket. Ellenőrizzük a gépek elérhetőségét.

```
[root@kube01 ~]# ping -c 5 kube02
PING kube02 (192.168.110.162) 56(84) bytes of data.
64 bytes from kube02 (192.168.110.162): icmp_seq=1 ttl=64 time=0.208 ms
64 bytes from kube02 (192.168.110.162): icmp_seq=2 ttl=64 time=0.192 ms
64 bytes from kube02 (192.168.110.162): icmp_seq=3 ttl=64 time=0.202 ms
64 bytes from kube02 (192.168.110.162): icmp_seq=4 ttl=64 time=0.204 ms
64 bytes from kube02 (192.168.110.162): icmp_seq=5 ttl=64 time=0.234 ms

--- kube02 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4122ms
rtt min/avg/max/mdev = 0.192/0.208/0.234/0.014 ms
[root@kube01 ~]# ping -c 5 kube03
PING kube03 (192.168.110.163) 56(84) bytes of data.
64 bytes from kube03 (192.168.110.163): icmp_seq=1 ttl=64 time=0.247 ms
64 bytes from kube03 (192.168.110.163): icmp_seq=2 ttl=64 time=0.295 ms
64 bytes from kube03 (192.168.110.163): icmp_seq=3 ttl=64 time=0.190 ms
64 bytes from kube03 (192.168.110.163): icmp_seq=4 ttl=64 time=0.207 ms
64 bytes from kube03 (192.168.110.163): icmp_seq=5 ttl=64 time=0.179 ms

--- kube03 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4084ms
rtt min/avg/max/mdev = 0.179/0.223/0.295/0.042 ms
```

Ha minden gép helyreállt, akkor futtassuk az ellenőrzéseket.

Control plane gépek listája.

```
# kubectl get nodes -l node-role.kubernetes.io/control-plane
NAME     STATUS   ROLES           AGE   VERSION
kube01   Ready    control-plane   67d   v1.36.2
kube02   Ready    control-plane   67d   v1.36.2
kube03   Ready    control-plane   32m   v1.36.2
```

ETCD klaszter tagok listála.

```
# etcdctl --endpoints=https://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 member list -w table
+------------------+---------+--------+-----------------------------+------
-----------------------+------------+
|        ID        | STATUS  |  NAME  |          PEER ADDRS         |
CLIENT ADDRS         | IS LEARNER |
+------------------+---------+--------+-----------------------------+------
-----------------------+------------+
| 1546a2d2ac61bd8d | started | kube02 | https://192.168.110.162:2380 |
https://192.168.110.162:2379 |      false |
| 20e2ce3086b2e277 | started | kube01 | https://192.168.110.161:2380 |
https://192.168.110.161:2379 |      false |
| 23fb75a79460e3c9 | started | kube03 | https://192.168.110.163:2380 |
https://192.168.110.163:2379 |      false |
+------------------+---------+--------+-----------------------------+------
-----------------------+------------+
```

ETCD endpoint állapot ellenőrzése.

```
# etcdctl --
endpoints=https://kube01:2379,https://kube02:2379,https://kube03:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key endpoint status -w table
+---------------------+------------------+---------+-----------------+------
---+--------+-----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+--------------------------
+-------------------+
|      ENDPOINT       |        ID        | VERSION | STORAGE VERSION | DB
SIZE | IN USE | PERCENTAGE NOT IN USE | QUOTA  | IS LEADER | IS LEARNER |
RAFT TERM | RAFT INDEX | RAFT APPLIED INDEX | ERRORS | DOWNGRADE TARGET
VERSION | DOWNGRADE ENABLED |
+---------------------+------------------+---------+-----------------+------
---+--------+-----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+--------------------------
+-------------------+
| https://kube01:2379 | 20e2ce3086b2e277 |   3.6.8 |           3.6.0 |   24
MB |  12 MB |                   50% | 2.1 GB |      true |      false |
6 |    7036311 |           7036311 |        |                           |
false |
| https://kube02:2379 | 1546a2d2ac61bd8d |   3.6.8 |           3.6.0 |   24
MB |  12 MB |                   50% | 2.1 GB |     false |      false |
6 |    7036312 |           7036312 |        |                           |
false |
| https://kube03:2379 | 23fb75a79460e3c9 |   3.6.8 |           3.6.0 |   24
MB |  12 MB |                   50% | 2.1 GB |     false |      false |
6 |    7036312 |           7036312 |        |                           |
false |
+---------------------+------------------+---------+-----------------+------
```

```
---+--------+----------------------+--------+-----------+------------+-----
------+------------+--------------------+--------+-------------------------
+-------------------+
```

És egy utolsó ellenőrzés.

```
# etcdctl --
endpoints=https://kube01:2379,https://kube02:2379,https://kube03:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key endpoint health -w table
+----------------------+--------+--------------+-------+
|      ENDPOINT        | HEALTH |     TOOK     | ERROR |
+----------------------+--------+--------------+-------+
| https://kube01:2379 |   true |  37.207191ms |       |
| https://kube02:2379 |   true |  39.074018ms |       |
| https://kube03:2379 |   true | 197.541227ms |       |
+----------------------+--------+--------------+-------+
```

A worker gépek állapota eltérhet a control plane gépekben tárolt állapottól. Ebben az esetben érdemes minden worker gépet újraindítani.

Végleges kiesés

Amennyiben a két kiesett gép többé már nem érhető el, akkor a meglevő node-ból egy gépes klasztert kell készíteni, és utánna csatlakoztatni a többi gépet. Az alábbi műveleteket a megmaradt gépen kell futtatni.

Ellenőrizzük aze etcd állapotát.

```
# etcdctl --
endpoints=https://kube01:2379,https://kube02:2379,https://kube03:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key endpoint status -w table
{"level":"warn","ts":"2026-07-09T12:59:52.835472+0200","logger":"etcd-
client","caller":"v3@v3.6.12/retry_interceptor.go:68","msg":"retrying of
unary invoker failed","target":"etcd-
endpoints://0xc00027f2c0/kube02:2379","peer":"Peer{Addr: <nil>, LocalAddr:
<nil>, AuthInfo:
<nil>}","method":"/etcdserverpb.Maintenance/Status","attempt":0,"error":"rpc
error: code = DeadlineExceeded desc = latest balancer error: connection
error: desc = \"transport: Error while dialing: dial tcp
192.168.110.162:2379: connect: no route to host\""}
Failed to get the status of endpoint https://kube02:2379 (context deadline
exceeded)
{"level":"warn","ts":"2026-07-09T12:59:57.837654+0200","logger":"etcd-
client","caller":"v3@v3.6.12/retry_interceptor.go:68","msg":"retrying of
unary invoker failed","target":"etcd-
```

```
endpoints://0xc00025e960/kube03:2379","peer":"Peer{Addr: <nil>, LocalAddr:
<nil>, AuthInfo:
<nil>}","method":"/etcdserverpb.Maintenance/Status","attempt":0,"error":"rpc
error: code = DeadlineExceeded desc = latest balancer error: connection
error: desc = \"transport: Error while dialing: dial tcp
192.168.110.163:2379: connect: no route to host\""}
Failed to get the status of endpoint https://kube03:2379 (context deadline
exceeded)
+---------------------+------------------+---------+----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+--------------------+----------------------+-----------
--------------+------------------+
|      ENDPOINT       |        ID        | VERSION | STORAGE VERSION | DB
SIZE | IN USE | PERCENTAGE NOT IN USE | QUOTA  | IS LEADER | IS LEARNER |
RAFT TERM | RAFT INDEX | RAFT APPLIED INDEX |        ERRORS         |
DOWNGRADE TARGET VERSION | DOWNGRADE ENABLED |
+---------------------+------------------+---------+----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+--------------------+----------------------+-----------
--------------+------------------+
| https://kube01:2379 | 20e2ce3086b2e277 |   3.6.8 |           3.6.0 |   24
MB | 8.2 MB |                  67% | 2.1 GB |     false |      false |
6 |    7041707 |            7041707 | etcdserver: no leader |
|             false |
+---------------------+------------------+---------+----------------+------
---+--------+----------------------+--------+-----------+------------+-----
------+------------+--------------------+----------------------+-----------
--------------+------------------+
```

Amennyiben működik az etcd, akkor snapshot-ot kell készíteni az adatbázisról.

```
# etcdctl snapshot save /root/etcd-snapshot-disaster.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
{"level":"info","ts":"2026-07-09T13:02:05.335290+0200","caller":"snapshot/v3
_snapshot.go:83","msg":"created temporary db file","path":"/root/etcd-
snapshot-disaster.db.part"}
{"level":"info","ts":"2026-07-09T13:02:05.341997+0200","logger":"client","ca
ller":"v3@v3.6.12/maintenance.go:236","msg":"opened snapshot stream;
downloading"}
{"level":"info","ts":"2026-07-09T13:02:05.415922+0200","caller":"snapshot/v3
_snapshot.go:96","msg":"fetching
snapshot","endpoint":"https://127.0.0.1:2379"}
{"level":"info","ts":"2026-07-09T13:02:05.666551+0200","logger":"client","ca
ller":"v3@v3.6.12/maintenance.go:302","msg":"completed snapshot read;
closing"}
{"level":"info","ts":"2026-07-09T13:02:05.997155+0200","caller":"snapshot/v3
_snapshot.go:111","msg":"fetched
snapshot","endpoint":"https://127.0.0.1:2379","size":"24
MB","took":"661.672699ms","etcd-version":"3.6.0"}
```

```
{"level":"info","ts":"2026-07-09T13:02:05.997294+0200","caller":"snapshot/v3
_snapshot.go:121","msg":"saved","path":"/root/etcd-snapshot-disaster.db"}
Snapshot saved at /root/etcd-snapshot-disaster.db
Server version 3.6.0
```

Le kell tiltani és le kell állítani a `kubelet` és a `containerd` szolgáltatásokat.

```
# systemctl stop kubelet containerd
# systemctl disable kubelet containerd
Removed '/etc/systemd/system/multi-user.target.wants/containerd.service'.
Removed '/etc/systemd/system/multi-user.target.wants/kubelet.service'.
```

Újra kell indítani a gépet.

```
# reboot
```

Újraindítás után a gépen nem szabad futnia a szolgáltatásoknak.

```
# systemctl is-active containerd kubelet
inactive
inactive
```

A snapshot helyreállítását el kell végezni egy másik könyvtárban `/var/lib/etcd-new`

```
# etcdutl snapshot restore /root/etcd-snapshot-disaster.db --name kube01 --
data-dir /var/lib/etcd-new --initial-cluster
kube01=https://192.168.110.161:2380  --initial-cluster-token new-cluster --
initial-advertise-peer-urls=https://192.168.110.161:2380
2026-07-09T13:06:50+02:00   info    snapshot/v3_snapshot.go:305    restoring
snapshot    {"path": "/root/etcd-snapshot-disaster.db", "wal-dir":
"/var/lib/etcd-new/member/wal", "data-dir": "/var/lib/etcd-new", "snap-dir":
"/var/lib/etcd-new/member/snap", "initial-memory-map-size": 10737418240}
2026-07-09T13:06:50+02:00   info    bbolt    backend/backend.go:203
Opening db file (/var/lib/etcd-new/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: 0xc000062190}
2026-07-09T13:06:50+02:00   info    bbolt    bbolt@v1.4.3/db.go:322
Opening bbolt db (/var/lib/etcd-new/member/snap/db) successfully
2026-07-09T13:06:50+02:00   info    schema/membership.go:138    Trimming
membership information from the backend...
2026-07-09T13:06:50+02:00   info    bbolt    backend/backend.go:203
Opening db file (/var/lib/etcd-new/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: 0xc00011c128}
2026-07-09T13:06:50+02:00   info    bbolt    bbolt@v1.4.3/db.go:322
Opening bbolt db (/var/lib/etcd-new/member/snap/db) successfully
```

```
2026-07-09T13:06:50+02:00   info    membership/cluster.go:424    added
member    {"cluster-id": "3cd4235e48cb8de9", "local-member-id": "0", "added-
peer-id": "234c167f7d0f641f", "added-peer-peer-urls":
["https://192.168.110.161:2380"], "added-peer-is-learner": false}
2026-07-09T13:06:51+02:00   info    bbolt    backend/backend.go:203
Opening db file (/var/lib/etcd-new/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: 0xc0000621d8}
2026-07-09T13:06:51+02:00   info    bbolt    bbolt@v1.4.3/db.go:322
Opening bbolt db (/var/lib/etcd-new/member/snap/db) successfully
2026-07-09T13:06:51+02:00   info    snapshot/v3_snapshot.go:333    restored
snapshot    {"path": "/root/etcd-snapshot-disaster.db", "wal-dir":
"/var/lib/etcd-new/member/wal", "data-dir": "/var/lib/etcd-new", "snap-dir":
"/var/lib/etcd-new/member/snap", "initial-memory-map-size": 10737418240}
```

Sikeres helyreállítás után a korábbi könyvtárat el kell mozgatni.

```
# mv /var/lib/etcd /var/lib/etcd-old
```

A helyreállított snapshot könyvtárat a helyére kell másolni.

```
# cp -a /var/lib/etcd-new /var/lib/etcd
```

Újra engedélyezni kell a szolgáltatásokat.

```
# systemctl enable kubelet containerd
Created symlink '/etc/systemd/system/multi-
user.target.wants/kubelet.service' →
'/usr/lib/systemd/system/kubelet.service'.
Created symlink '/etc/systemd/system/multi-
user.target.wants/containerd.service' →
'/usr/lib/systemd/system/containerd.service'.
```

Újra kell indítani a gépet.

```
# reboot
```

Az újraindítást követően a control plane elindul. Az indulást követően ellenőrizzük a klaszer állapotát.

Control plane gépek listája.

```
# kubectl get nodes -l node-role.kubernetes.io/control-plane
NAME     STATUS     ROLES           AGE   VERSION
kube01   Ready      control-plane   67d   v1.36.2
kube02   NotReady   control-plane   67d   v1.36.2
kube03   NotReady   control-plane   90m   v1.36.2
```

ETCD endpoint lista.

```
# etcdctl --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 endpoint status -w table
+------------------------+------------------+---------+-----------------+---
------+--------+----------------------+--------+-----------+------------+--
---------+------------+--------------------+--------+----------------------
---+-------------------+
|        ENDPOINT        |        ID        | VERSION | STORAGE VERSION | DB
SIZE | IN USE | PERCENTAGE NOT IN USE | QUOTA  | IS LEADER | IS LEARNER |
RAFT TERM | RAFT INDEX | RAFT APPLIED INDEX | ERRORS | DOWNGRADE TARGET
VERSION | DOWNGRADE ENABLED |
+------------------------+------------------+---------+-----------------+---
------+--------+----------------------+--------+-----------+------------+--
---------+------------+--------------------+--------+----------------------
---+-------------------+
| https://127.0.0.1:2379 | 234c167f7d0f641f |   3.6.8 |           3.6.0 |
24 MB |  13 MB |                  49% | 2.1 GB |      true |      false |
3 |        966 |                966 |        |                           |
false |
+------------------------+------------------+---------+-----------------+---
------+--------+----------------------+--------+-----------+------------+--
---------+------------+--------------------+--------+----------------------
---+-------------------+
```

ETCD tagok listája.

```
# etcdctl --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 member list -w table
+------------------+---------+--------+-----------------------------+------
------------------------+------------+
|        ID        | STATUS  |  NAME  |          PEER ADDRS         |
CLIENT ADDRS         | IS LEARNER |
+------------------+---------+--------+-----------------------------+------
------------------------+------------+
| 234c167f7d0f641f | started | kube01 | https://192.168.110.161:2380 |
https://192.168.110.161:2379 |      false |
+------------------+---------+--------+-----------------------------+------
------------------------+------------+
```

Az etcd adatbázisa rendben. Új példány indult.

Töröljük a két hiányzó gépet.

```
# kubectl delete node kube02
node "kube02" deleted
# kubectl delete node kube03
node "kube03" deleted
```

Hozzunk létre tokent és tanúsítvány kulcsot.

```
# JOIN_CMD="$(kubeadm token create --print-join-command)"
# CERT_KEY="$(kubeadm init phase upload-certs --upload-certs | tail -n 1)"
# echo "$JOIN_CMD --control-plane --certificate-key $CERT_KEY"
kubeadm join 192.168.110.160:6443 --token 7q469t.81sbouok0em1pvjo --
discovery-token-ca-cert-hash
sha256:838e04036d629fbf3e152fc6606bf9a143f8e1ca0b0bc00d747fe1caba5df8f1  --
control-plane --certificate-key
10babb9fd52e8ae866eabae323e7f486ae6c5fa26613ab93004738cab8432e87
```

Két új control plane gép telepítése után csatlakoztatjuk a gépeket a klaszterhez.

```
kubeadm join 192.168.110.160:6443 --token 7q469t.81sbouok0em1pvjo --
discovery-token-ca-cert-hash
sha256:838e04036d629fbf3e152fc6606bf9a143f8e1ca0b0bc00d747fe1caba5df8f1  --
control-plane --certificate-key
10babb9fd52e8ae866eabae323e7f486ae6c5fa26613ab93004738cab8432e87
[preflight] Running pre-flight checks
[preflight] Reading configuration from the "kubeadm-config" ConfigMap in
namespace "kube-system"...
[preflight] Use 'kubeadm init phase upload-config kubeadm --config your-
config-file' to re-upload it.
...

This node has joined the cluster and a new control plane instance was
created:

* Certificate signing request was sent to apiserver and approval was
received.
* The Kubelet was informed of the new secure connection details.
* Control plane label and taint were applied to the new node.
* The Kubernetes control plane instances scaled up.
* A new etcd member was added to the local/stacked etcd cluster.

To start administering your cluster from this node, you need to run the
following as a regular user:

    mkdir -p $HOME/.kube
    sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
    sudo chown $(id -u):$(id -g) $HOME/.kube/config

Run 'kubectl get nodes' to see this node join the cluster.
```

Mindkét gépen állítsuk be a kubernetes konfigurációt a root felhasználónak.

```
# rm -rf /root/.kube/
# mkdir -p $HOME/.kube
# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
# sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Ellenőrizzük a gépeket.

```
# kubectl get nodes -l node-role.kubernetes.io/control-plane
NAME     STATUS   ROLES           AGE    VERSION
kube01   Ready    control-plane   67d    v1.36.2
kube02   Ready    control-plane   10m    v1.36.2
kube03   Ready    control-plane   10m    v1.36.2
```

```
# etcdctl --
endpoints=https://kube01:2379,https://kube02:2379,https://kube03:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key endpoint health -w table
+----------------------+--------+--------------+-------+
|      ENDPOINT        | HEALTH |     TOOK     | ERROR |
+----------------------+--------+--------------+-------+
| https://kube01:2379 |   true |  13.402358ms |       |
| https://kube03:2379 |   true |  90.974734ms |       |
| https://kube02:2379 |   true | 173.267605ms |       |
+----------------------+--------+--------------+-------+
```

A helyreállítás sikeres.

Három control plane kiesése

Három control plane gép elvesztésekor a legegyszerűbb megoldás három azonos nevű és IP című gép telepítése. A telepítés után vissza kell állítani a konfiguráció mentéseket a megfelelő gépeken.

Mindhárom gépre be kell másolni a mentéseket a `/root/k8s-config-$(hostname)-YYYYMMMDD.tar.gz` fájlként. A másolás után az összes gépek ki kell csomagolni a fájlok tartalmát.

```
# cd /
# tar -xzf /root/k8s-config-$(hostname)-YYYYMMMDD.tar.gz
```

Ezzel a konfiguráció helyreállítása megtörtént.

ETCD snapshot helyreállításához mindhárom gépre másoljuk be a snapshot állományt a `/root/etcd-snapshot-$(hostname)-YYYYMMDD.db` néven. A fájl mindhárom gépen ugyanaz legyen.

```
# (kube01) md5sum /root/etcd-snapshot-$(hostname)YYYYMMDD.db
6a66ab3ffa291dcb9d199817858b5617 /root/etcd-snapshot-$(hostname)-YYYYMMDD.db

# (kube02) md5sum /root/etcd-snapshot-$(hostname)YYYYMMDD.db
6a66ab3ffa291dcb9d199817858b5617 /root/etcd-snapshot-$(hostname)-YYYYMMDD.db

# (kube03) md5sum /root/etcd-snapshot-$(hostname)YYYYMMDD.db
6a66ab3ffa291dcb9d199817858b5617 /root/etcd-snapshot-$(hostname)-YYYYMMDD.db
```

Gépenként állítsuk helyre a snapshot tartalmát.

kube01

```
etcdutl snapshot restore /root/etcd-snapshot-$(hostname)-YYYYMMDD.db --name
kube01 --data-dir /var/lib/etcd --initial-cluster
kube01=https://192.168.110.161:2380,kube02=https://192.168.110.162:2380,kube
03=https://192.168.110.163:2380 --initial-advertise-peer-urls
https://192.168.110.161:2380 --initial-cluster-token new-cluster
```

kube02

```
etcdutl snapshot restore /root/etcd-snapshot-$(hostname)-YYYYMMDD.db --name
kube02 --data-dir /var/lib/etcd --initial-cluster
kube01=https://192.168.110.161:2380,kube02=https://192.168.110.162:2380,kube
03=https://192.168.110.163:2380 --initial-advertise-peer-urls
https://192.168.110.162:2380 --initial-cluster-token new-cluster
```

kube03

```
etcdutl snapshot restore /root/etcd-snapshot-$(hostname)-YYYYMMDD.db --name
kube03 --data-dir /var/lib/etcd --initial-cluster
kube01=https://192.168.110.161:2380,kube02=https://192.168.110.162:2380,kube
03=https://192.168.110.163:2380 --initial-advertise-peer-urls
https://192.168.110.163:2380 --initial-cluster-token new-cluster
```

Jogosultságok beállítása a `/var/lib/etcd` könyvtárra.

```
# chown -R root:root /var/lib/etcd
```

Minden gépen engedélyezni kell a `containerd` és a `kubelet` szolgáltatásokat.

```
# systemctl enable containerd kubelet
Created symlink '/etc/systemd/system/multi-
user.target.wants/containerd.service' →
'/usr/lib/systemd/system/containerd.service'.
Created symlink '/etc/systemd/system/multi-
user.target.wants/kubelet.service' →
'/usr/lib/systemd/system/kubelet.service'.
```

Újra kell indítani a control plane gépeket.

```
# reboot
```

Amikor a gépek elindultak ellenőrizni kell a k8s klaszter állapotát.

```
# kubectl get nodes -l node-role.kubernetes.io/control-plane
NAME     STATUS   ROLES           AGE    VERSION
kube01   Ready    control-plane   45m    v1.36.2
kube02   Ready    control-plane   45m    v1.36.2
kube03   Ready    control-plane   45m    v1.36.2
```

```
# etcdctl --
endpoints=https://kube01:2379,https://kube02:2379,https://kube03:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key endpoint health -w table
+----------------------+--------+--------------+-------+
|      ENDPOINT        | HEALTH |     TOOK     | ERROR |
+----------------------+--------+--------------+-------+
| https://kube01:2379 |   true |  10.749263ms |       |
| https://kube03:2379 |   true |  32.846299ms |       |
| https://kube02:2379 |   true |  45.252051ms |       |
+----------------------+--------+--------------+-------+
```

A helyreállítás sikeres.

Worker

Új worker gép létrehozása

A worker gépek nevei és az IP beállítások egyezzenek meg a korábbi gép azonosítóival.

A telepítés után valamelyik control plane gépen futtassuk a nem működő gép törlését.

```
# kubectl delete node kube04
node "kube04" deleted
```

A törlés után hozzunk létre új tokent a gép csatlakoztatásához.

```
# kubeadm token create --print-join-command
kubeadm join 192.168.110.160:6443 --token sfk438.u9jyjz6cdpz1puk5 --
discovery-token-ca-cert-hash
sha256:838e04036d629fbf3e152fc6606bf9a143f8e1ca0b0bc00d747fe1caba5df8f1
```

Csatlakoztassuk az új worker gépet a klaszterhez.

```
# kubeadm join 192.168.110.160:6443 --token sfk438.u9jyjz6cdpz1puk5 --
discovery-token-ca-cert-hash
sha256:838e04036d629fbf3e152fc6606bf9a143f8e1ca0b0bc00d747fe1caba5df8f1
[preflight] Running pre-flight checks
[preflight] Reading configuration from the "kubeadm-config" ConfigMap in
namespace "kube-system"...
```

```
[preflight] Use 'kubeadm init phase upload-config kubeadm --config your-
config-file' to re-upload it.
...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was
received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the
cluster.
```

Minden további műveletet a telepítési leírás szerint kell végrehajtani.

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

Permanent link:
https://wiki.r-l.hu/doku.php?id=kubernetes:mentes-helyreallitas&rev=1783623319

Last update: **2026/07/09 18:55**