etcdctl etcd command line

CONCEPTS

> Every key has a revision. Reads, watches and compaction all work in revisions, not timestamps.

put / del writes bump the revision
↓ raft
Cluster members agree on one history
↓ compact
History old revisions pile up
↓ defrag
Disk space is returned

> Without --endpoints etcdctl talks to 127.0.0.1:2379. Secure clusters need the TLS flags on every call.

CONNECTION

--endpoints=<*host:port,...*>
Cluster members to contact

--cacert=<*ca.crt*>
CA that signed the server cert

--cert=<*client.crt*> **--key=**<*client.key*>
Client certificate and key

-w json
Output as JSON (also table, fields)

KEYS

etcdctl put <*key*> <*value*>
Set a key

etcdctl get <*key*>
Read a key and its value

etcdctl get <*key*> **--print-value-only**
Value only, no key line

etcdctl get --prefix <*prefix*>
All keys starting with a prefix

etcdctl get --prefix --keys-only <*prefix*>
List key names only

KEYS (cont.)

etcdctl del <*key*>
Delete a key

etcdctl del --prefix <*prefix*>
Delete all keys under a prefix

WATCH

etcdctl watch <*key*>
Print changes to a key

etcdctl watch --prefix <*prefix*>
Watch every key under a prefix

LEASES

etcdctl lease grant <*ttl*>
Create a lease with a TTL in seconds

etcdctl put <*key*> <*value*> **--lease=**<*id*>
Attach a key to a lease

CLUSTER

etcdctl member list -w table
List members and IDs

etcdctl endpoint health --cluster
Check each member responds

etcdctl endpoint status --cluster -w table
Leader, DB size, raft term

MAINTENANCE

etcdctl snapshot save <*file*>
Write a point-in-time backup

etcdutl snapshot restore <*file*> **--data-dir=**<*dir*>
Restore into a new data dir

AUTH

etcdctl user add root
Create root before enabling auth

etcdctl auth enable
Turn authentication on