K3s Lightweight Kubernetes

CONCEPTS

K3s is one small binary: the install script drops it in /usr/local/bin, adds kubectl, crictl and ctr links, and sets up a systemd service.

Server control plane + datastore
↓ node-token
Agent K3S_URL + K3S_TOKEN
↓ registers
Cluster kubectl, manifests, snapshots
Install, join, operate

INSTALL

`curl -sfL https://get.k3s.io | sh -`
Install a single-node server

`curl -sfL https://get.k3s.io | K3S_URL=https://`*`<server>`*`:6443 K3S_TOKEN=`*`<token>`* `sh -`
Install as agent and join

`curl -sfL https://get.k3s.io | INSTALL_K3S_VERSION=`*`<version>`* `sh -`
Pin a version

`INSTALL_K3S_EXEC="server --cluster-init"`
Set the command and flags

`K3S_KUBECONFIG_MODE=644`
Make the kubeconfig readable

Install variables are saved into the service. Re-running the installer without them loses the old values.

SERVER AND AGENT

`k3s server`
Run a server node

`k3s agent`
Run a worker node

`--cluster-init`
Start an embedded-etcd cluster

`--server https://`*`<server>`*`:6443`
Join an existing cluster

`--token` *`<token>`*
Shared join secret

`--tls-san` *`<host-or-ip>`*
Add a name to the API cert

SERVER AND AGENT (cont.)

`--disable traefik`
Skip a packaged component

CONFIG FILE

`/etc/rancher/k3s/config.yaml`
Main config file

config.yaml

```
# flag names without the dashes
write-kubeconfig-mode: "0644"
tls-san:
  - k3s.example.com
disable:
  - traefik
node-taint+:
  - "dedicated=infra:NoSchedule"
```

CLI flags override the file. A key ending in + appends to the earlier value instead of replacing it.

`/var/lib/rancher/k3s/server/manifests/`
YAML here is auto-applied

CLUSTER AND NODES

`/etc/rancher/k3s/k3s.yaml`
Server kubeconfig

`export KUBECONFIG=/etc/rancher/k3s/k3s.yaml`
Use it with plain kubectl

`k3s kubectl get nodes`
Embedded kubectl

`k3s kubectl get pods -A`
All pods

`/var/lib/rancher/k3s/server/node-token`
Token to join agents

`k3s crictl ps`
Containers via the CRI

Copy k3s.yaml to your laptop and change the server address from 127.0.0.1 to the server's name.

SERVICE

`systemctl status k3s`
Server service state

`systemctl status k3s-agent`
Agent service state

`sudo systemctl restart k3s`
Restart after config changes

SERVICE (cont.)

`journalctl -u k3s -f`
Follow server logs

`/usr/local/bin/k3s-uninstall.sh`
Remove a server

`/usr/local/bin/k3s-agent-uninstall.sh`
Remove an agent

TOKENS AND CERTIFICATES

`k3s certificate check`
Show cert expiry

`k3s certificate rotate`
Rotate certs (stop K3s first)

ETCD AND BACKUPS

`k3s etcd-snapshot save`
Snapshot now

`k3s etcd-snapshot ls`
List snapshots

`k3s server --cluster-reset --cluster-reset-restore-path=`*`<snapshot>`*
Restore a snapshot

`/var/lib/rancher/k3s/server/db/snapshots/`
Default snapshot directory

RECIPES

Add a worker

```
# on the server
sudo cat /var/lib/rancher/k3s/
server/node-token
# on the worker
curl -sfL https://get.k3s.io |
K3S_URL=https://<server>:6443
K3S_TOKEN=<token> sh -
# back on the server
sudo k3s kubectl get nodes
```

HA with embedded etcd

```
# first server
curl -sfL https://get.k3s.io |
K3S_TOKEN=<secret> sh -s - server
--cluster-init
# more servers
curl -sfL https://get.k3s.io |
K3S_TOKEN=<secret> sh -s - server
--server https://<first>:6443
```

Restore from a snapshot

```
sudo systemctl stop k3s
sudo k3s server --cluster-reset \
  --cluster-reset-restore-
path=<snapshot-path>
sudo systemctl start k3s
```