k3d k3s in Docker

CONCEPTS

Docker host your machine
Cluster network k3d-<name>
Nodes serverlb, server-N, agent-N

One Docker network per cluster

Every node is a Docker container named k3d-<cluster>-<role>-<n>. Host ports reach the cluster through the k3d-<cluster>-serverlb container, so map ports with @loadbalancer.

Node filters pick targets: server:0, agent:*, agent:0,1, loadbalancer. Repeatable flags like -p, -v, -e and --k3s-arg accept them after an @.

CLUSTERS

k3d cluster create *<name>*
Create a cluster

k3d cluster list
List clusters and node counts

k3d cluster delete *<name>*
Delete a cluster

k3d cluster delete --all
Delete every cluster

CLUSTERS (cont.)

k3d cluster stop *<name>*
Stop the cluster containers

k3d cluster start *<name>*
Start a stopped cluster

k3d cluster create *<name>* **-s** *<n>* **-a** *<n>*
Set server and agent count

k3d cluster create *<name>* **-i** *<image>*
Pick the k3s version via image

PORTS AND NETWORKING

k3d cluster create -p "8080:80@loadbalancer"
Map host port to ingress

k3d cluster create -p "8082:30080@agent:0"
Map a NodePort via an agent

k3d cluster create --api-port 6550
Fix the API server host port

k3d node edit k3d-*<name>***-serverlb --port-add 8080:80**
Add a port to a live cluster

KUBECONFIG

k3d kubeconfig merge *<name>* **--kubeconfig-switch-context**
Merge and switch context

KUBECONFIG (cont.)

k3d kubeconfig get *<name>*
Print a cluster's kubeconfig

k3d kubeconfig merge *<name>* **--kubeconfig-merge-default**
Merge into ~/.kube/config

k3d kubeconfig merge *<name>* **-o** *<file>*
Write to a separate file

kubectl config use-context k3d-*<name>*
Switch context by hand

create already updates your default kubeconfig and switches to context k3d-<name>. Use merge for clusters made elsewhere.

NODES

k3d node list
List all nodes

k3d node create *<name>* **-c** *<cluster>*
Add an agent to a cluster

k3d node create *<name>* **--role server -c** *<cluster>*
Add a server node

k3d node stop *<node>*
Stop one node

k3d node start *<node>*
Start one node

NODES (cont.)

k3d node delete *<node>*
Remove a node

IMAGES AND REGISTRIES

k3d image import *<image>* **-c** *<name>*
Copy a local image into nodes

k3d image import *<archive.tar>* **-c** *<name>*
Import from a tarball

k3d registry create *<name>* **--port** *<port>*
Create a local registry

k3d registry list
List registries

k3d registry delete *<name>*
Delete a registry

k3d cluster create --registry-create *<name>*
Make a registry with the cluster

k3d cluster create --registry-use k3d-*<name>***:***<port>*
Use an existing registry

MOUNTS, ENV AND K3S ARGS

k3d cluster create -v *<host>***:***<node>***@***<filter>*
Mount a host path

MOUNTS, ENV AND K3S ARGS (cont.)

k3d cluster create -e "KEY=val@server:0"
Set a node env var

k3d cluster create --k3s-arg "*<arg>***@server:*"**
Pass a flag to k3s

k3d cluster create --k3s-node-label "k=v@agent:*"
Label k3s nodes

CONFIG FILE

k3d cluster create --config *<file>*
Create from a config file

k3d cluster create *<name>* **-c** *<file>*
Flags override the file

k3d cluster delete -c *<file>*
Delete the file's cluster

k3d config init
Print a default config

MAINTENANCE

k3d version
Show k3d and k3s versions

k3d *<cmd>* **--verbose**
Debug logging

docker ps -f name=k3d-
See the node containers