Cilium CLI

Kubernetes networking CLI

CONCEPTS

The cilium CLI talks to the cluster in your kubeconfig and manages the install; hubble is a separate binary for reading flows.

cilium CLI install and manage
↓ deploys
Agent + operator per node, eBPF
↓ flows to
Hubble Relay cluster-wide
↓ queried by
hubble CLI / UI observe flows

Every command takes --context to target another cluster in your kubeconfig.

INSTALL

`cilium install`
Install Cilium with detected defaults

`cilium install --version <ver>`
Install a specific release

`cilium install --set <key>=<value>`
Pass a Helm value

`cilium install --dry-run`
Show resources without applying

`cilium install --dry-run-helm-values`
Show non-default Helm values

`cilium upgrade --version <ver>`
Upgrade to a release

`cilium upgrade --reuse-values --set <key>=<value>`
Change one value on upgrade

`cilium uninstall`
Remove Cilium

STATUS AND CONFIG

`cilium status`
Show component health

`cilium status --wait`
Block until Cilium is ready

`cilium config view`
Print the agent configuration

`cilium sysdump`
Collect a debug archive

CONNECTIVITY

`cilium connectivity test`
Run the end-to-end test suite

`cilium connectivity test --single-node`
Test on a one-node cluster

`cilium connectivity test --context <a> --multi-cluster <b>`
Test across a cluster mesh

HUBBLE

`cilium hubble enable`
Deploy Hubble and Relay

`cilium hubble port-forward`
Forward Relay to localhost:4245

`hubble status`
Check Relay and flow counts

`hubble observe`
Stream recent flows

`hubble observe -P`
Auto port-forward Relay first

`hubble observe --pod <name>`
Flows for one pod

`hubble observe --verdict DROPPED`
Only dropped traffic

`hubble observe --protocol http`
Only L7 HTTP flows

`cilium hubble ui`
Open the Hubble UI

CLUSTER MESH

`cilium install --set cluster.name=<name> --set cluster.id=<n>`
Give each cluster a unique name and ID

`cilium clustermesh enable --context <ctx>`
Deploy the mesh API server

`cilium clustermesh status --wait`
Wait for the mesh to be ready

`cilium clustermesh connect --destination-context <ctx>`
Connect two clusters

Cluster names and IDs must be unique, and pod CIDRs must not overlap.