istioctl Service mesh CLI

CONCEPTS

Run analyze after every config change: it catches missing gateways, bad hosts and unlabelled namespaces before traffic breaks.

istiod control plane
↓ xDS push
Envoy / ztunnel data plane
↓ proxy-status
proxy-config what it received

A pod is <name>.<namespace>; use deployment/<name> in place of a pod name where the command accepts it.

proxy-config prints the xDS config a proxy holds; proxy-status tells you whether it is current.

INSTALL

istioctl install --set profile=*<name>*
Install a profile

istioctl install -f *<iop.yaml>*
Install from an IstioOperator file

istioctl install --set *<path>*=*<value>*
Override a single value

istioctl x precheck
Check the cluster before install

istioctl manifest generate -f *<iop.yaml>*
Render manifests to stdout

istioctl install -r *<rev>*
Install a revision alongside others

istioctl install --dry-run
Preview without applying

istioctl upgrade
Alias for install: apply new version

istioctl uninstall --revision *<rev>*
Remove one revision

istioctl uninstall --purge
Remove every Istio resource

STATUS & VERSION

istioctl version
Show client and control plane versions

istioctl version --remote=false
Client version only

istioctl proxy-status
Sync state of every proxy

istioctl ps *<pod>*.*<ns>*
Sync state of one proxy

ANALYZE & VALIDATE

istioctl analyze
Analyze the current namespace

istioctl analyze -A
Analyze all namespaces

istioctl analyze *<file.yaml>*
Check a file before applying

istioctl analyze -n *<ns>*
Analyze one namespace

istioctl analyze --use-kube=false *<file.yaml>*
Files only, ignore the cluster

istioctl analyze --failure-threshold Error
Only fail CI on errors

istioctl validate -f *<file.yaml>*
Validate resource syntax

istioctl bug-report
Collect logs and state into an archive

PROXY CONFIG

istioctl proxy-config all *<pod>*
Dump all Envoy config

istioctl pc cluster *<pod>*
Upstream clusters of a proxy

istioctl pc listener *<pod>*
Listeners of a proxy

istioctl pc route *<pod>*
Route tables of a proxy

istioctl pc endpoint *<pod>*
Endpoints the proxy knows

istioctl pc secret *<pod>*
Certificates loaded in the proxy

PROXY CONFIG (cont.)

istioctl pc listener *<pod>* **--port** *<n>*
Filter by port

istioctl pc cluster *<pod>* **--fqdn** *<host>*
Filter by service host

istioctl pc endpoint *<pod>* **--cluster** *<name>*
Endpoints of one cluster

istioctl pc route *<pod>* **--name** *<n>* **-o json**
Full JSON of one route

istioctl pc cluster deployment/*<name>*
Use a deployment instead of a pod

istioctl pc log *<pod>* **--level** *<lvl>*
Change Envoy log level

INJECT & DESCRIBE

istioctl x describe pod *<pod>*
Explain routing and policy for a pod

istioctl x check-inject *<pod>*
Why was sidecar injected or not?

istioctl kube-inject -f *<file.yaml>*
Inject the sidecar into a manifest

DASHBOARDS

istioctl dashboard kiali
Open Kiali

REVISIONS & TAGS

istioctl tag set *<tag>* **--revision** *<rev>*
Point a tag at a revision

AMBIENT

istioctl waypoint apply -n *<ns>*
Create a waypoint proxy

GLOBAL FLAGS

-n *<ns>*
Target namespace

--context *<name>*
Kubeconfig context to use