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.

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 uninstall --purge
Remove every Istio resource

STATUS & VERSION

istioctl version
Show client and control plane versions

istioctl proxy-status
Sync state of every proxy

ANALYZE & VALIDATE

istioctl analyze
Analyze the current namespace

istioctl analyze -A
Analyze all namespaces

ANALYZE & VALIDATE (cont.)

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

PROXY 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

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