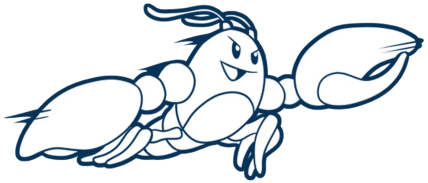
BUOYANT
SERVICE MESH
ACADEMY

# **Buoyant Cloud –** See What Your Mesh Is Telling You



# Agenda

- What is Buoyant Cloud?
- Architecture and how data flows
- Deploying the agent
- Dashboard walkthrough (metrics, topology, events, health)
- Golden metrics
- Alerts and integrations
- Multi-cluster visibility
- Data privacy and security
- Best practices
- Demo / Q&A

SERVICEMESH ACADEMY

# The Problem: Why Do You Need Buoyant Cloud?

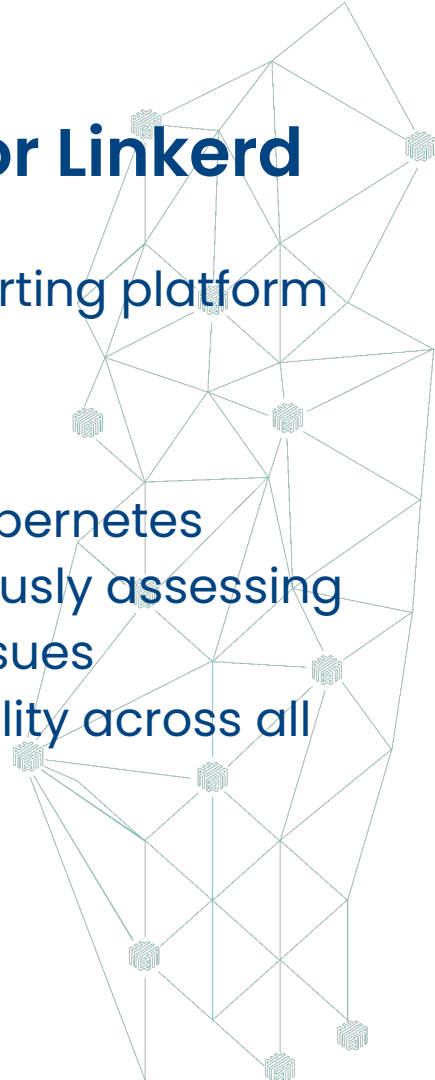
- **Brings multiple service mesh** deployments into a single pane of glass
- **Lack of unified visibility** across multiple clusters
- **Reactive troubleshooting** instead of proactive health management
- **Teams need SRE tooling** without building everything from scratch
- **Pain points**: "How do I know my mesh is healthy? How do I keep it healthy?"

SERVICEMESH ACADEMY

# Buoyant Cloud: Your Command Center for Linkerd

Buoyant Cloud is the SaaS dashboard, monitoring, and alerting platform for Buoyant Enterprise for Linkerd (BEL).

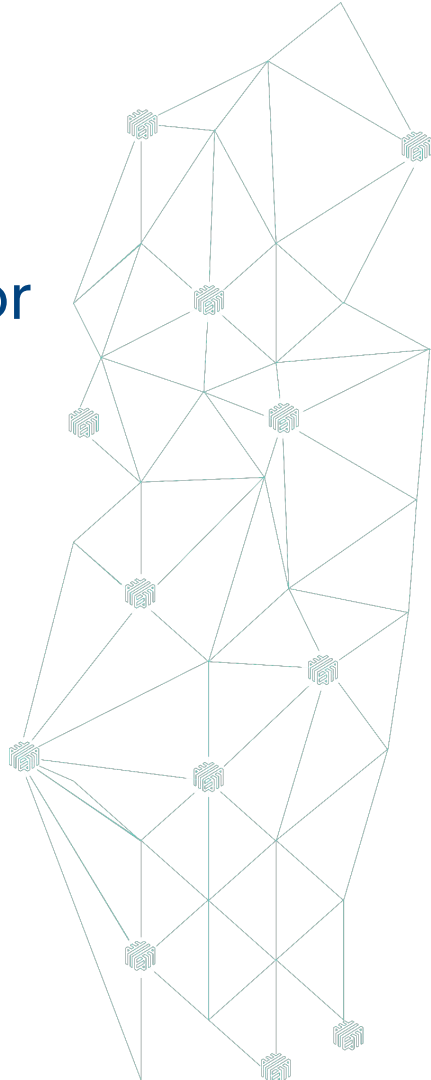
It provides a global, cross-cluster health dashboard for Kubernetes environments running Linkerd, automatically and continuously assessing the health of Linkerd deployments, pinpointing potential issues proactively, and providing global platform-wide observability across all connected clusters.



SERVICEMESH ACADEMY

# What is Buoyant Cloud?

- Dashboard, monitoring, and alerting tool for Buoyant Enterprise for Linkerd (BEL)
- Provides a single pane of glass across all your Linkerd-managed clusters
- Proactive diagnostics
- Performance management



SERVICEMESH ACADEMY

# Architecture Overview

- **Agent pushes data out** — no inbound connections required
- **Agent is lightweight** — minimal resource footprint
- **Deployed via separate Helm chart**, integrates with standard release cycle tooling

SERVICEMESH ACADEMY

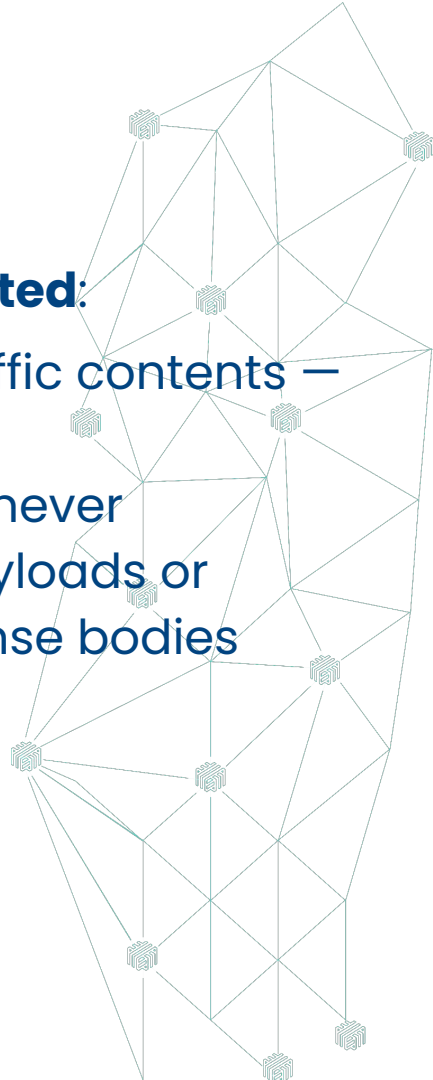
# What Data Does Buoyant Cloud Collect?

**What IS collected**:

- Metadata about Deployments, StatefulSets, DaemonSets
- Linkerd CRDs (ServerAuthorizations, Servers, Multiclusters etc)
- Proxy metrics about application traffic (success rates, latencies, request volumes)
- Proxy mTLS public key info (NOT private keys)

**What is NOT collected**:

- Application traffic contents — never
- Private keys — never
- Application payloads or request/response bodies



SERVICEMESH ACADEMY

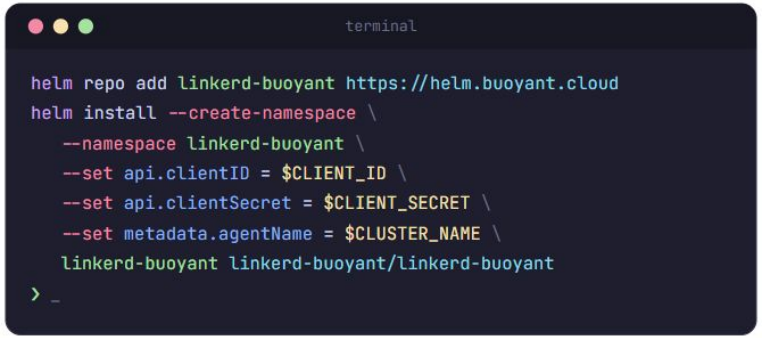
# Deploying the Buoyant Cloud Agent

Prerequisites:

- BEL Strategic Plan
- Linkerd control plane installed
- Helm

Steps:

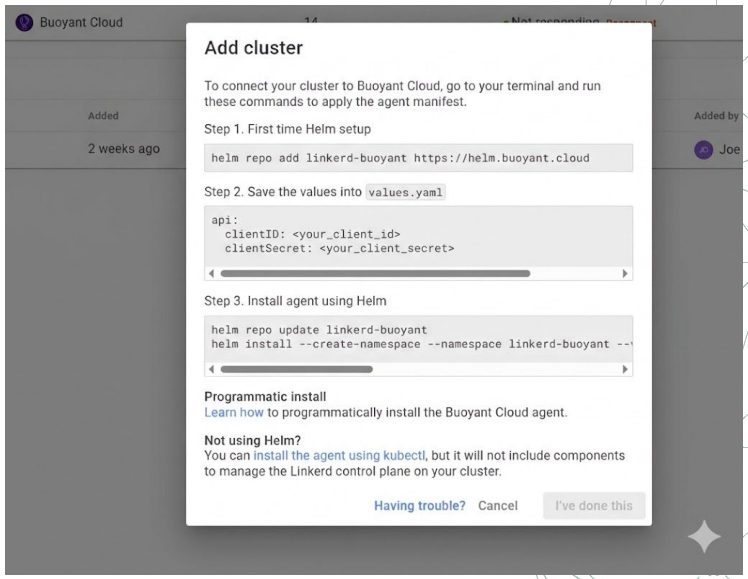
- Retrieve **client ID** and **secret** from Buoyant Cloud Dashboard and set these variables, along with the **cluster name**




SERVICEMESH ACADEMY

# Agent Configuration

- **Custom values file** from Buoyant Cloud Clusters page
- **Configurable** resource requests/limits for the agent pod
- **Network policies**: agent needs outbound HTTPS to Buoyant Cloud endpoints
- **Upgraded through Helm**

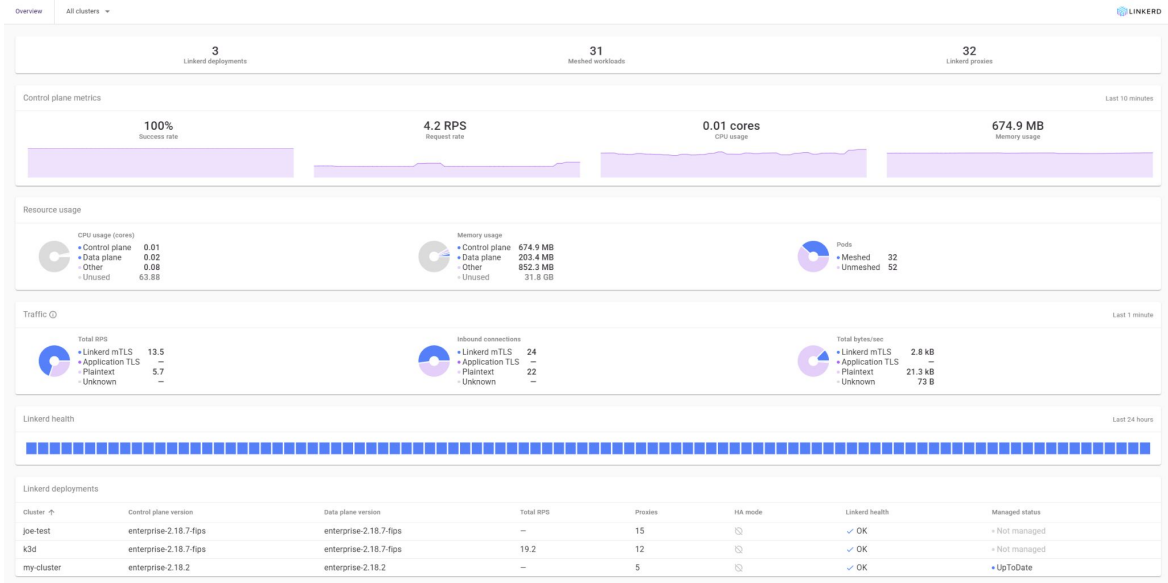



SERVICEMESH ACADEMY

https://artifacthub.io/packages/helm/linkerd-buoyant/linkerd-buoyant

# Dashboard: Overview

- **Unified** view across all connected clusters
- **At-a-glance** health status of the mesh
- **Quick indicators**: control plane health, data plane health, version info, resource usage
- **Number of meshed** vs unmeshed workloads
- **Recent events** and alerts




SERVICEMESH ACADEMY

# Dashboard: Workload Metrics

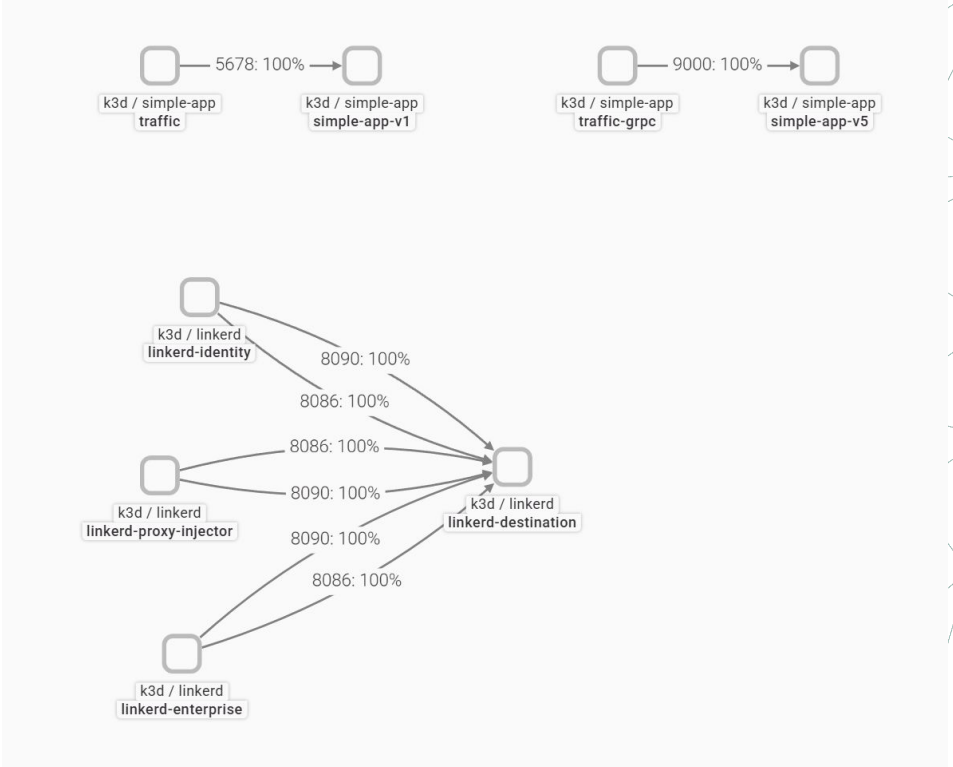
**Golden metrics for every meshed workload:**

- **HTTP/gRPC/TCP** success rates
- **Latency percentiles** (p50, p95, p99)
- **Request volume** (RPS)
- **Real-time** CPU and memory usage for any Kubernetes workload (meshed or not)
- **Historical graphs** — trend analysis
- **Drill down** from cluster to namespace to workload
- **Quickly identify** workloads in trouble


SERVICEMESH ACADEMY

# Dashboard: Topology

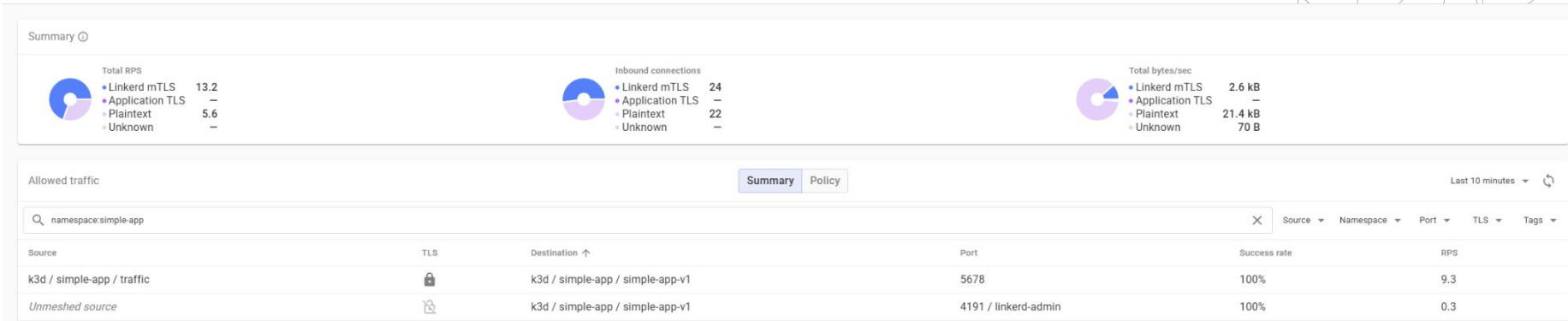
- **Visual dependency map** between workloads
- **See which services call which** — clients and dependencies
- **Detect new dependencies** or new clients automatically
- **Filter by namespace**, cluster, workload
- **Pin nodes** to focus on specific service chains
- **Use case**: "Where does traffic go after it hits my API gateway?"




SERVICEMESH ACADEMY

# Dashboard: Traffic

- **Traffic view** with success rate column
- **Time window selector**
- **Policy columns** showing authorization policy status
- **Same underlying data** source as the Topology page



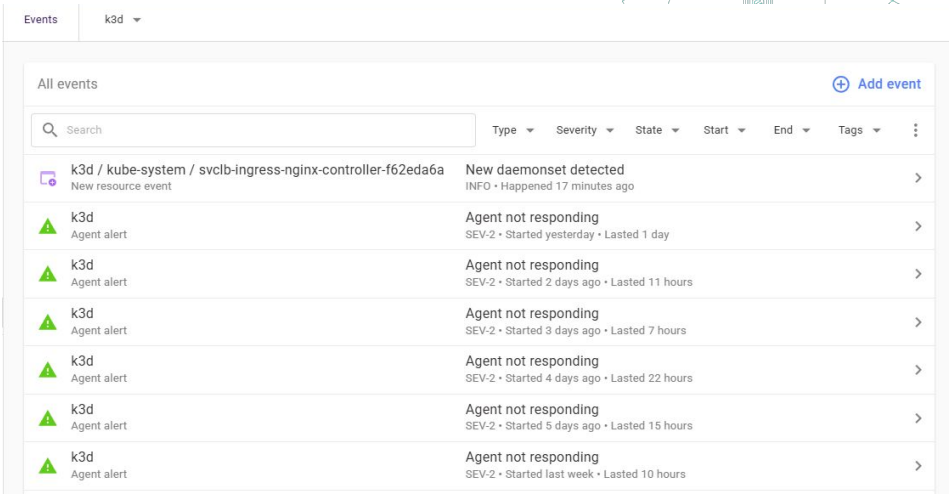

SERVICEMESH ACADEMY

# Dashboard: Events and Health Checks

- **Continuous health assessment** of the Linkerd deployment
- **Proactive diagnostics** — issues flagged before they become incidents

**Event types:**

- Version incompatibilities (control plane vs data plane)
- Certificate expiration warnings
- Configuration drift
- Rollout events (deployments, restarts)
- New workload meshed / unmeshed
- Sanity-checking mesh installs and upgrades




SERVICEMESH ACADEMY

# Alerts and Integrations

**Alert on:**

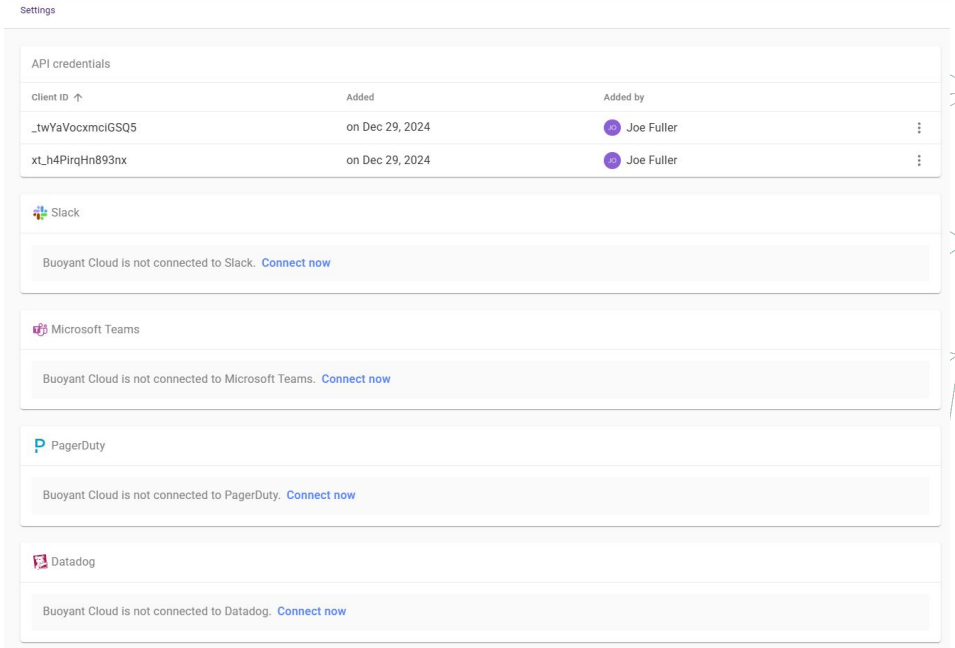
- Linkerd health issues
- Workload traffic anomalies
- Rollout events
- SLO breaches

**Notification channels:**

- Slack
- Pagerduty
- Microsoft Teams
- Datadog

Integrate into your existing incident response workflows

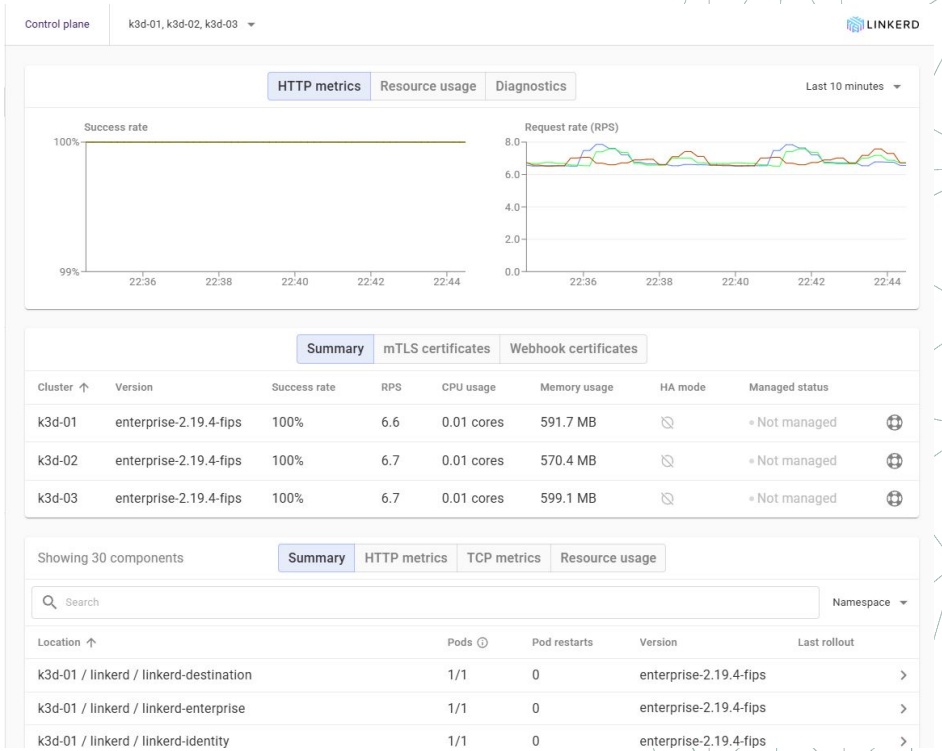
Tip: start with broad alerts, refine over time




SERVICEMESH ACADEMY

# Multi-Cluster Visibility

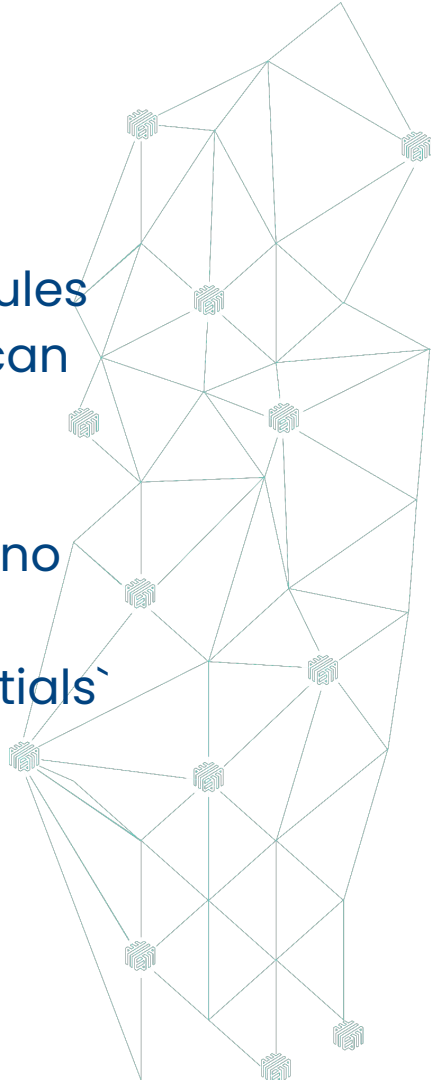
- **Single dashboard** across all clusters
- **Compare health and metrics** across environments (dev, staging, prod)
- **Identify cross-cluster** version drift
- **Multi-cluster** service mirroring visibility
- **Use case**: "Are all my production clusters running the same Linkerd version?"




SERVICEMESH ACADEMY

# Common Pitfalls / Troubleshooting

- **Agent can't connect**: check network policies / egress rules
- **Cluster name mismatch**: renaming after registration can cause confusion
- **Stale data**: agent pod not running or crashlooping
- **Missing metrics**: workloads not meshed (no sidecar = no proxy metrics)
- **Certificate issues**: ensure `buoyant-cloud-org-credentials` secret is correct
- **Resource**: link to Buoyant troubleshooting docs



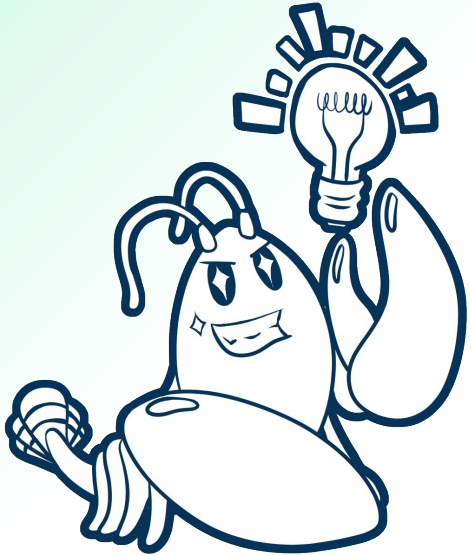
SERVICEMESH ACADEMY

# Best Practices

- **Naming clusters thoughtfully** — the `CLUSTER_NAME` persists in Buoyant Cloud; use a convention (e.g., `env-region-purpose`)
- **Deploy the agent to every cluster** — even dev/staging, for full visibility
- **Integrate alerts from day one** — connect Slack/PagerDuty before you need it
- **Use workload exclusion** — exclude noisy or irrelevant workloads to reduce clutter
- **Review topology regularly** — catch unexpected dependencies early
- **Keep the agent updated** — upgrade via Helm when new versions are released
- **Verify agent artifacts** — use signature verification for supply chain security
- **Use "Send Diagnostics"** — for troubleshooting with Buoyant support

SERVICEMESH ACADEMY

# Q&A



SERVICEMESH ACADEMY