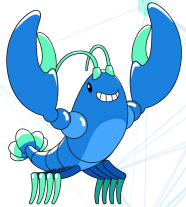
BUOYANT

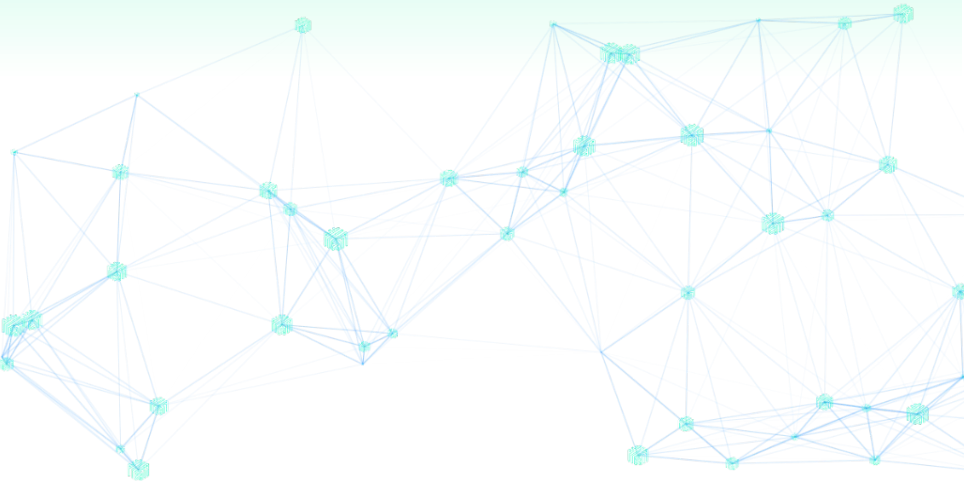
Creators of LINKERD

Kyverno 101 with Linkerd

Cortney Nickerson, Community at Nirmata
Flynn, Technical Evangelist for Linkerd



What's on the agenda?



- ➔ Intro to Policy as Code and Kyverno
- ➔ Quick intro to Linkerd
- ➔ Why you should care about this stuff 🙂
- ➔ DEMOS!
- ➔ Gotchas

BUOYANT

How do you follow along?

- https://github.com/BuoyantIO/service-mesh-academy/tree/main/kyverno-101-and-linkerd
- I'll be using a `k3d` cluster, but pretty much any cluster that can support LoadBalancer services should work

BUOYANT

How do you follow along?

➔ **For this demo, we'll use Buoyant Enterprise for Linkerd 2.18!**

To use Buoyant Enterprise for Linkerd to follow along, you will need a free Buoyant ID from https://enterprise.buoyant.io/.

We promise it's worth it and we won't sell your information to anyone! 🙂

(Linkerd edge-25.4.4 or later will work, too.)

BUOYANT

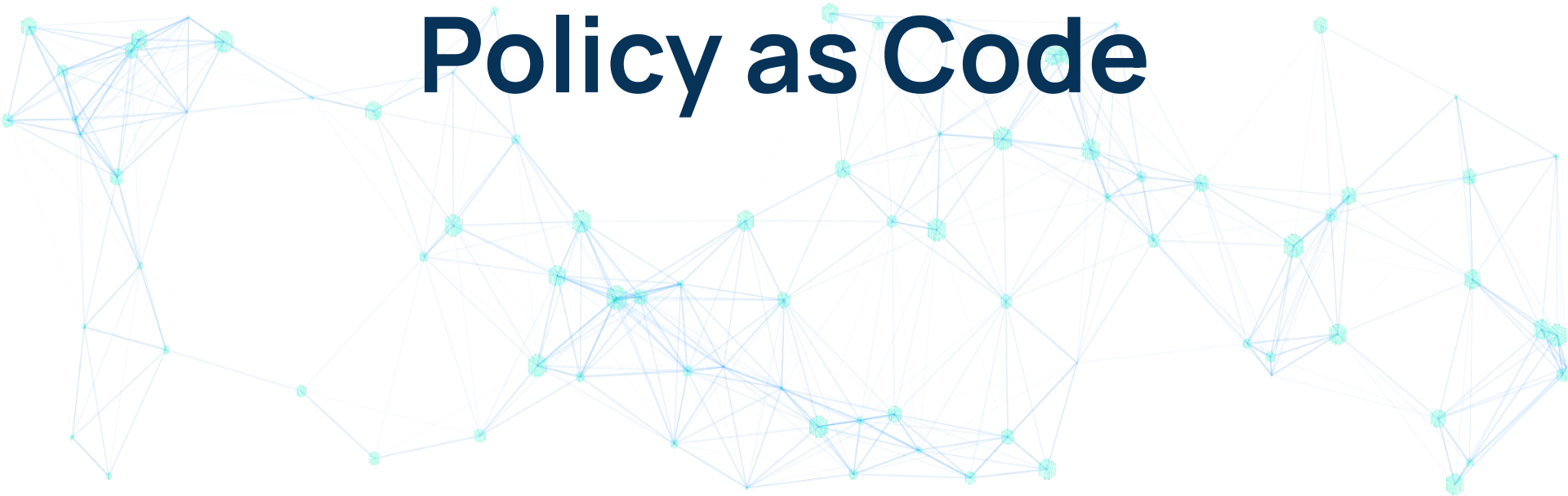
How do you follow along?

- **kubectl**
  https://kubernetes.io/docs/tasks/tools/
- **linkerd CLI**
  https://linkerd.io/2/getting-started
- **helm**
  https://helm.sh/docs/intro/quickstart
- **bat**
  https://github.com/sharkdp/bat
- **jq**
  https://github.com/jqlang/jq
- **yq**
  https://github.com/mikefarah/yq
- **tree**
  `brew install tree` on MacOS

BUOYANT

BUOYANT
SERVICE MESH
ACADEMY

Policy as Code



What is Policy as Code?

01

Declarative Rule Definition

Write rules in declarative formats like YAML and CEL (Common Expression Language). These human-readable formats make policies accessible to both developers and operations teams.

02

Version Control Integration

Store, review, and version policies exactly like application code. This brings familiar development practices to infrastructure governance.

03

Automated Enforcement

Deploy policies across all environments automatically, ensuring consistent rule application from development through production.

04

Consistency & Auditability

Achieve reproducible compliance and maintain complete audit trails of policy changes and enforcement actions.

Why Do Policies Matter?

- **Standardization at Scale**

  Ensure consistent configurations across multiple teams, clusters, and environments without manual oversight.

- **Safe Developer Self-Service**

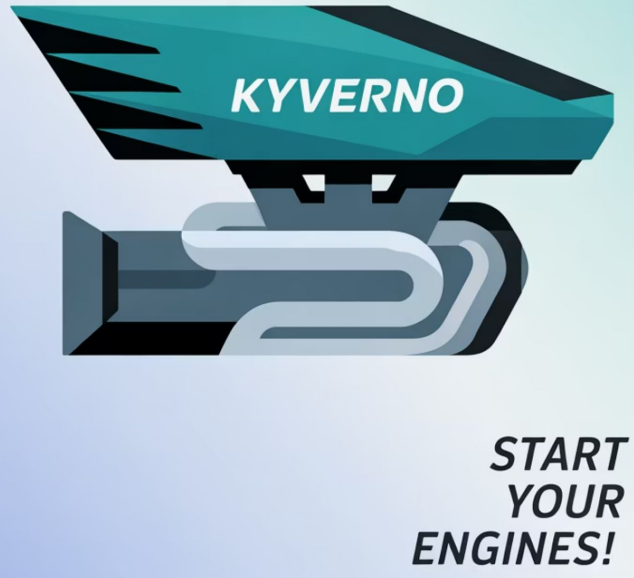
  Enable development teams to deploy confidently while automated policies enforce security and compliance guardrails.

- **Eliminate Review Bottlenecks**

  Reduce dependency on manual security reviews through automated policy validation and enforcement.

- **Governance Without Friction**

  Maintain regulatory compliance and organizational standards without slowing down feature delivery cycles.



Enter Kyverno: Your Policy Engine

Kubernetes-Native

Built specifically for Kubernetes environments, no external dependencies or complex integrations required.

Multi-Stage Execution

Operates across admission control, background scans, CI/CD pipelines, and Infrastructure as Code workflows.

Familiar Syntax

Write policies in YAML or CEL (Common Expression Language) - no need to learn specialized policy languages.

Five Policy Types

Comprehensive policy framework covering validation, mutation, generation, image verification, and cleanup operations.

Kyverno Architecture

1

Admission Controller

Runs as a Kubernetes admission controller, intercepting all API requests before resources are persisted to etcd.

2

Policy Engine

Webhook passes requests through Kyverno's engine, which applies validate, mutate, generate, and verify logic based on active policies.

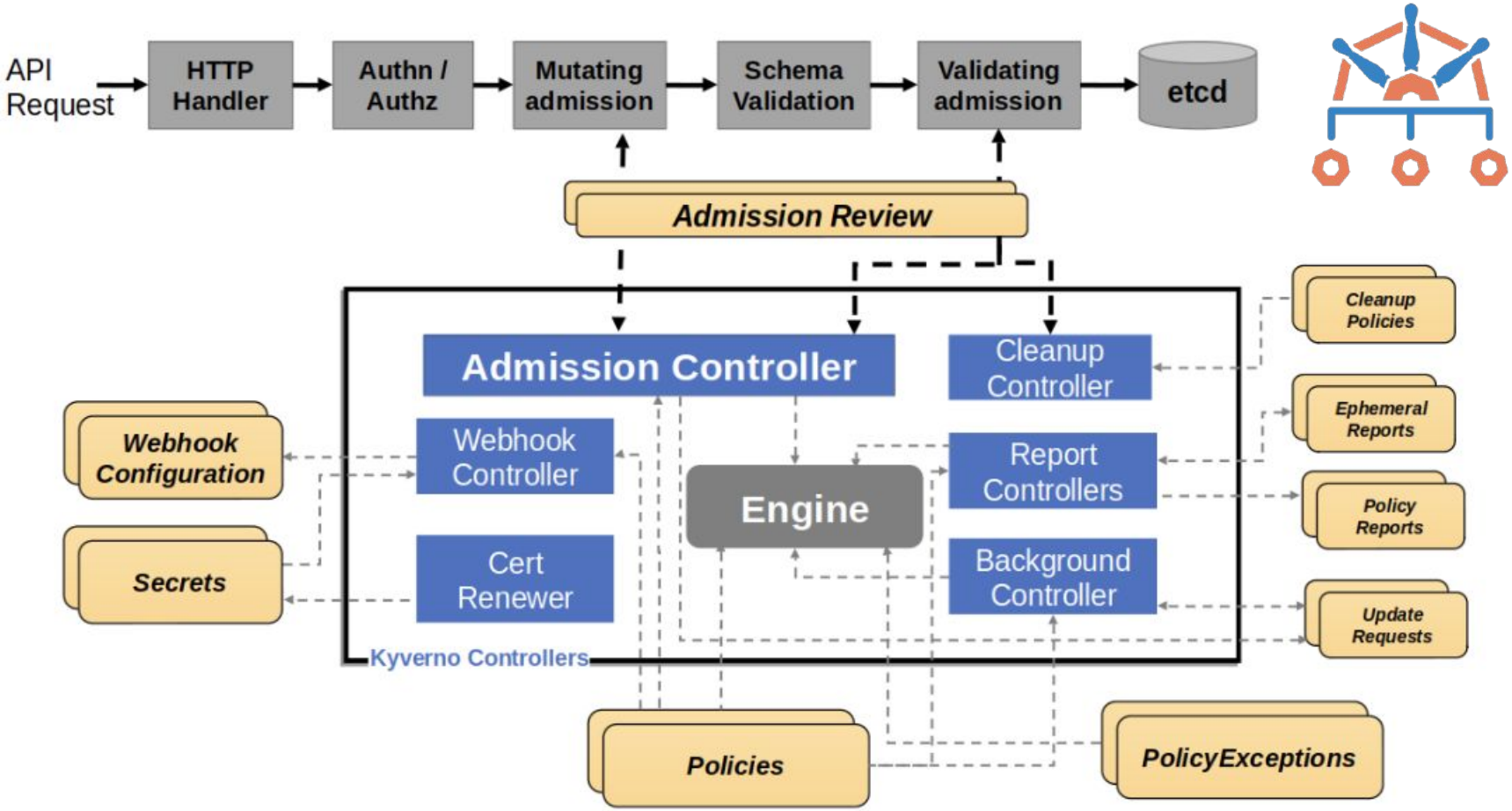
3

Controllers & Reporting

Background controllers handle cleanup operations, policy reporting, and exception management for comprehensive governance.

Policy Reports provide complete visibility into policy enforcement actions and audit trails, essential for compliance and troubleshooting.

The architecture ensures minimal performance impact while providing comprehensive policy coverage.


API Request
HTTP Handler
Authn / Authz
Mutating admission
Schema Validation
Validating admission
etcd
Admission Review
Admission Controller
Cleanup Controller
Webhook Controller
Report Controllers
Engine
Cert Renewer
Background Controller
Kyverno Controllers
Webhook Configuration
Secrets
Cleanup Policies
Ephemeral Reports
Policy Reports
Update Requests
Policies
PolicyExceptions

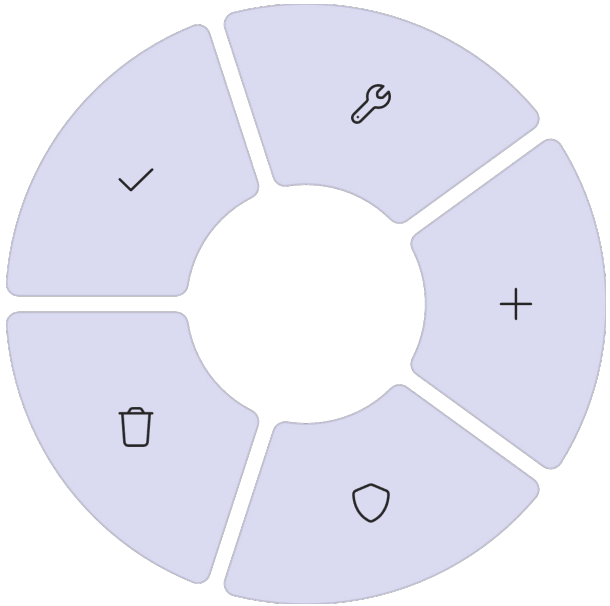
Five Policy Types: Your Complete Toolkit

Validate

Check and verify resource configurations against security and compliance requirements before deployment.

Cleanup

Remove stale, temporary, or expired resources automatically based on schedules, conditions, or lifecycle events.



Mutate

Automatically modify resources to add missing labels, security contexts, or compliance metadata during creation.

Generate

Auto-create supporting resources like NetworkPolicies, ResourceQuotas, or ConfigMaps when new resources are deployed.

VerifyImages

Ensure container images are digitally signed and trusted before allowing them to run in your clusters.

Why CEL Matters

Alignment with Kubernetes Future

Kubernetes APIs like ValidatingAdmissionPolicy and MutatingAdmissionPolicy already use CEL as their standard expression language, making this transition essential for long-term compatibility.

Kyverno's CEL Extensions

Image verification functions - Built-in functions for signature and attestation validation
Resource fetch functions - `resource.Get()` and `resource.List()` for dynamic policy decisions
Generator functions - Advanced templating and resource creation capabilities

Future-proof your policy investment by adopting the same expression language that will power the next generation of Kubernetes admission controls.

The transition to CEL ensures that your Kyverno policies remain compatible as Kubernetes evolves, while providing immediate benefits in expressiveness and capability.

Beyond Kubernetes

Kyverno policies can enforce rules on **any JSON payload** by setting `mode: JSON`, expanding governance beyond cluster boundaries into your entire delivery pipeline.

1

Terraform Plan Validation

Enforce infrastructure policies directly on Terraform plans. For example, prevent creation of EKS clusters with public endpoints or ensure all RDS instances have encryption enabled.

2

CI/CD Configuration Governance

Validate pipeline configurations, deployment manifests, and other structured data before they reach production environments.

Pipeline Integration: Kyverno becomes a universal policy engine for your entire delivery pipeline, not just Kubernetes clusters. This unified approach ensures consistent governance from code to production.

This capability transforms Kyverno from a Kubernetes-only tool into a comprehensive policy platform for modern infrastructure and application delivery.

Example Use Cases

The following real-world scenarios demonstrate Kyverno's practical applications across security, compliance, and operational excellence:

Mandatory Labels

Ensure all resources have required labels for cost allocation, team ownership, and environment classification.

Naming Conventions

Enforce consistent resource naming patterns that support automation and operational clarity.

Resource Limits

Mandate CPU requests and limits to ensure fair resource distribution and prevent resource starvation.

Auth Policy Protection

Prevent modification of critical authentication and authorization policies by unauthorized users.

Certificate Security

Protect TLS certificates and ensure they meet security standards and rotation requirements.

Service Mesh Coverage

Verify that all applications participate in the service mesh for consistent security and observability.

Advantages of Kyverno + Kubernetes

Kubernetes Foundation

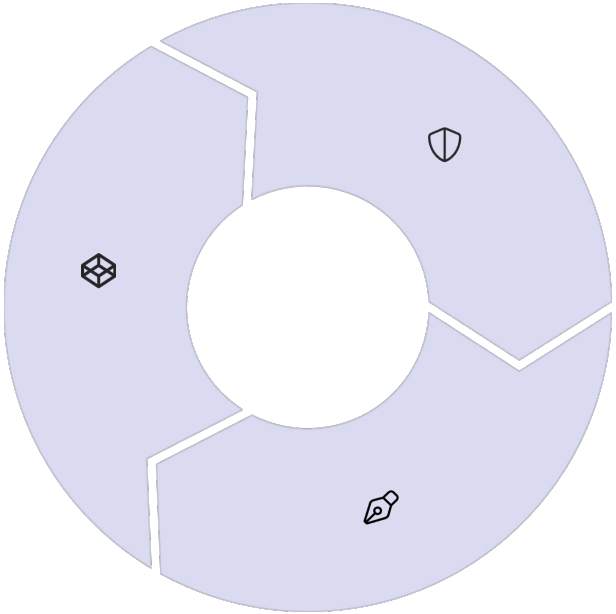
Kubernetes provides robust **RBAC** and mature **API machinery** that forms the foundation for secure, scalable policy management.

Kyverno Enhancement Layer

Kyverno adds comprehensive policy capabilities on top of Kubernetes' solid foundation:

Policy as Code

Version-controlled, reviewable policies that integrate with your existing development workflows and practices.



Comprehensive Enforcement

Validation, mutation, generation, and cleanup capabilities that cover the entire resource lifecycle.

Cross-Platform Governance

Consistent policy enforcement across clusters, CI/CD pipelines, and infrastructure as code workflows.

Result: A safer, more compliant, and highly automated platform that scales with your organization while maintaining security and governance standards.

Where to Start

Kyverno Playground

Experiment with policies without needing a cluster. Test CEL expressions, validate policy logic, and learn the syntax in a safe environment.

Start Small

Begin with fundamental policies that provide immediate value and build confidence with the platform.

Advanced Implementation

Graduate to sophisticated policies that provide comprehensive governance and security enforcement.

Beginner Policies

Require labels - Basic metadata governance

Add defaults - Automatic resource configuration

Audit image signatures - Supply chain visibility

Advanced Policies

SBOM enforcement - Comprehensive supply chain security

Cross-namespace generation - Complex resource relationships

Lifecycle cleanup - Automated resource management

Common Challenges

- **Policy Overload**

  Problem: Implementing too many policies simultaneously overwhelming teams and causing deployment friction.

  Solution: Start with 3-5 critical policies, validate their effectiveness, then gradually expand your policy coverage.

- **Overly Restrictive Rules**

  Problem: Blocking legitimate use cases with policies that are too strict or don't account for edge cases.

  Solution: Begin in audit mode to understand impact, then progressively tighten enforcement based on real usage patterns.

- **Missing CI/CD Integration**

  Problem: Policy violations discovered only at runtime, causing production surprises and emergency rollbacks.

  Solution: Integrate policy validation into your CI/CD pipelines to catch issues during development and testing phases.

Policy as Code with 

Consistency + Automation

Policy as Code eliminates manual processes and ensures consistent enforcement across all environments and teams.

Evolution to CEL

Kyverno's transition from YAML ClusterPolicy to CEL specialized policies aligns with Kubernetes' future and provides more power.

Beyond Kubernetes

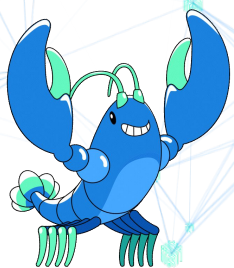
Universal JSON payload support extends governance to your entire delivery pipeline and infrastructure stack.

> **Key Takeaway:** Kyverno makes policies simple to write, powerful to execute, and future-proof for your evolving infrastructure needs. Start your policy as code journey today.

Try Kyverno Playground | Explore Documentation

BUOYANT
SERVICE MESH
ACADEMY

What is Linkerd?



What is Linkerd?

Linkerd is a **service mesh**.

***service mesh**, n:*

- An infrastructure layer providing security, reliability, and observability at the platform level, uniformly, across an entire application.

What is Linkerd?

Linkerd is a **service mesh**.

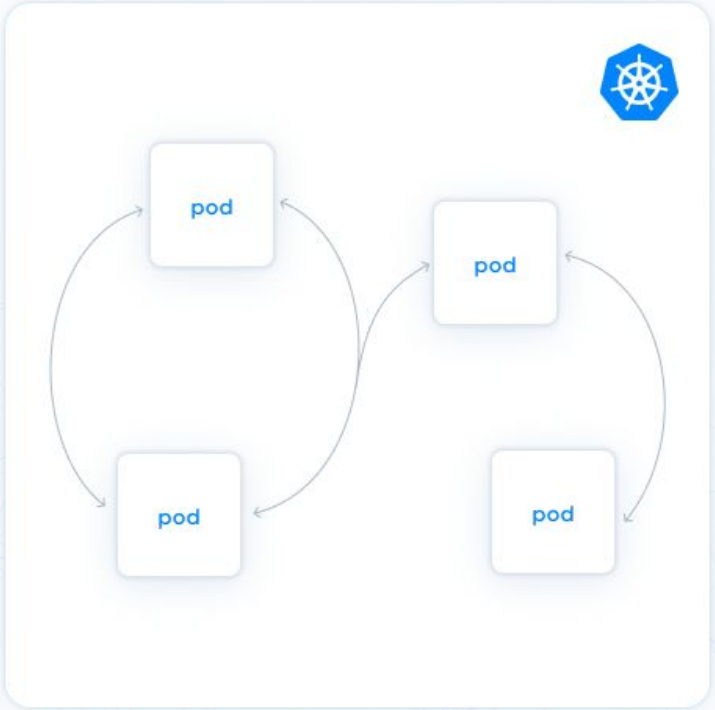
***service mesh**, n:*

- An infrastructure layer providing security, reliability, and observability at the platform level, uniformly, across an entire application.

The Microservices Architecture

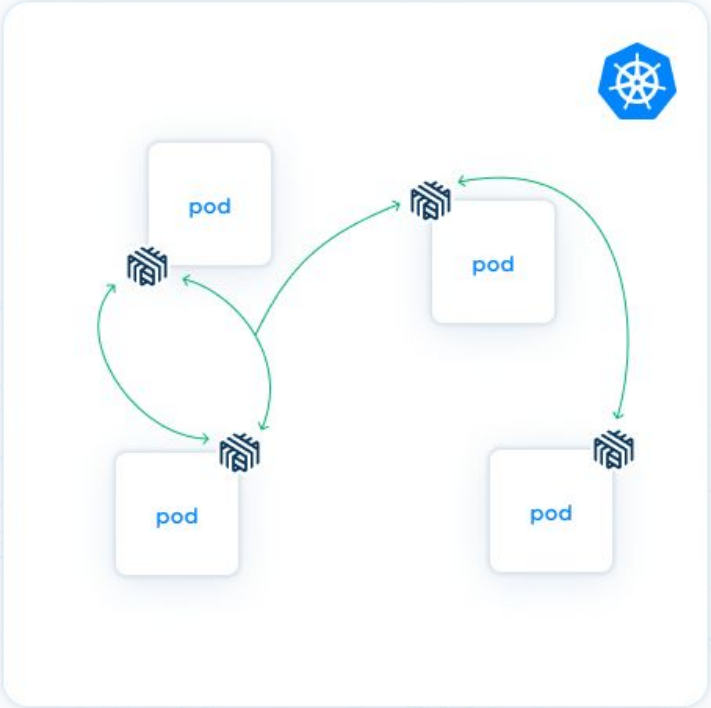
- Microservices communicate over an insecure, unreliable network.
- These are *fundamental characteristics* of the way real networking is built; they cannot be changed.

- Service meshes like Linkerd exist to make this situation better.



Microservices and the Mesh

- Like most other meshes, Linkerd works by adding a proxy (a *sidecar)* next to each application pod.
- *Unlike* any other mesh, Linkerd uses a purpose-built, lightweight, ultrafast Rust microproxy.
- These microproxies mediate and measure all communications in the mesh, which allows for all the mesh's functionality.



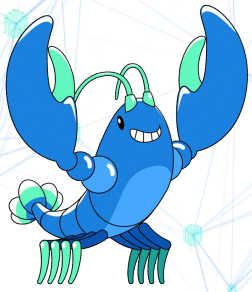
Why is this important?

Security, reliability, and observability are not optional.

- You can get them from a mesh.
- You can get them by writing a lot of application code.
- You can't do without them.

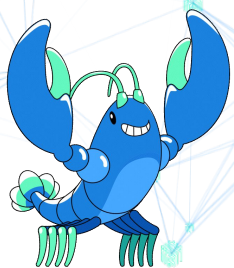
BUOYANT
SERVICE MESH
ACADEMY

DEMO 🤞



BUOYANT
SERVICE MESH
ACADEMY

Gotchas



Gotchas

- The big one: **use `kyverno test` and the Playground!**
  - Kyverno isn't always vocal about errors in policies
  - Even more importantly, Kyverno will do what you tell it to, which may not be what you meant! 🙂
- Be careful about resource types
  - For example, *examining* Pods might make sense, but then you might need to *mutate* Deployments

Tell us how we can improve!

Your feedback matters!

(We promise it won't take more than a few minutes, and it will help us tremendously — thank you! 🙂)

BUOYANT
Creators of LINKERD

Buoyant Enterprise for LINKERD

Rust-based network security and reliability for modern applications. Built on open source and designed for the enterprise.

- ➔ Zero-trust security and compliance across your entire network
- ➔ Global traffic management and control
- ➔ Full L7 application observability
- ➔ Built for the enterprise

Learn more & try it for free at buoyant.io/**enterprise-linkerd**

BUOYANT
Creators of LINKERD

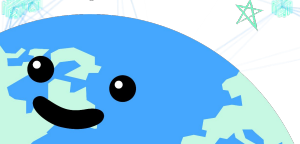
Updated Courses!

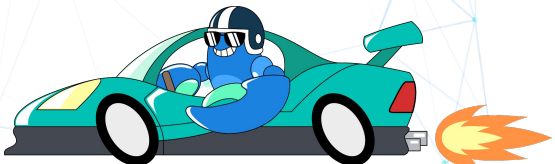
BUOYANT
Creators of LINKERD

Get Certified!

With hands-on LINKERD self-paced courses

- Service Mesh 101
- Linkerd in Production



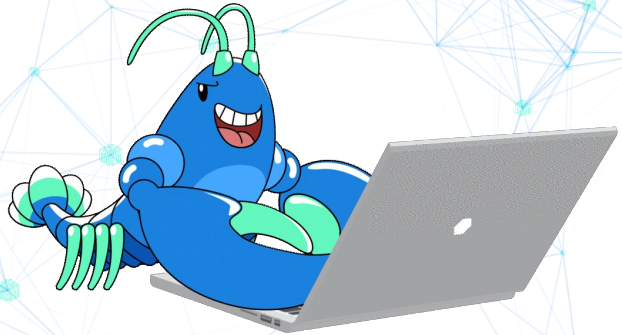


SERVICE MESH ACADEMY

Up Next on October 16

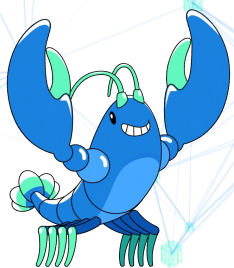
Managing Linkerd Certificates without Losing Your Mind

SIGN UP TODAY!
buoyant.io/sma



BUOYANT
SERVICE MESH
ACADEMY

Q&A



BUOYANT
SERVICE MESH
ACADEMY

Thanks much!

flynn@buoyant.io
@flynn on slack.linkerd.io