BUOYANT

The creators of LINKERD

Multi-cluster failover with Linkerd

Eliza Weisman
Linkerd Maintainer, Buoyant

@BuoyantIO

buoyant.io

Eliza Weisman

Systems Engineer, Buoyant
Linkerd Maintainer

@mycoliza

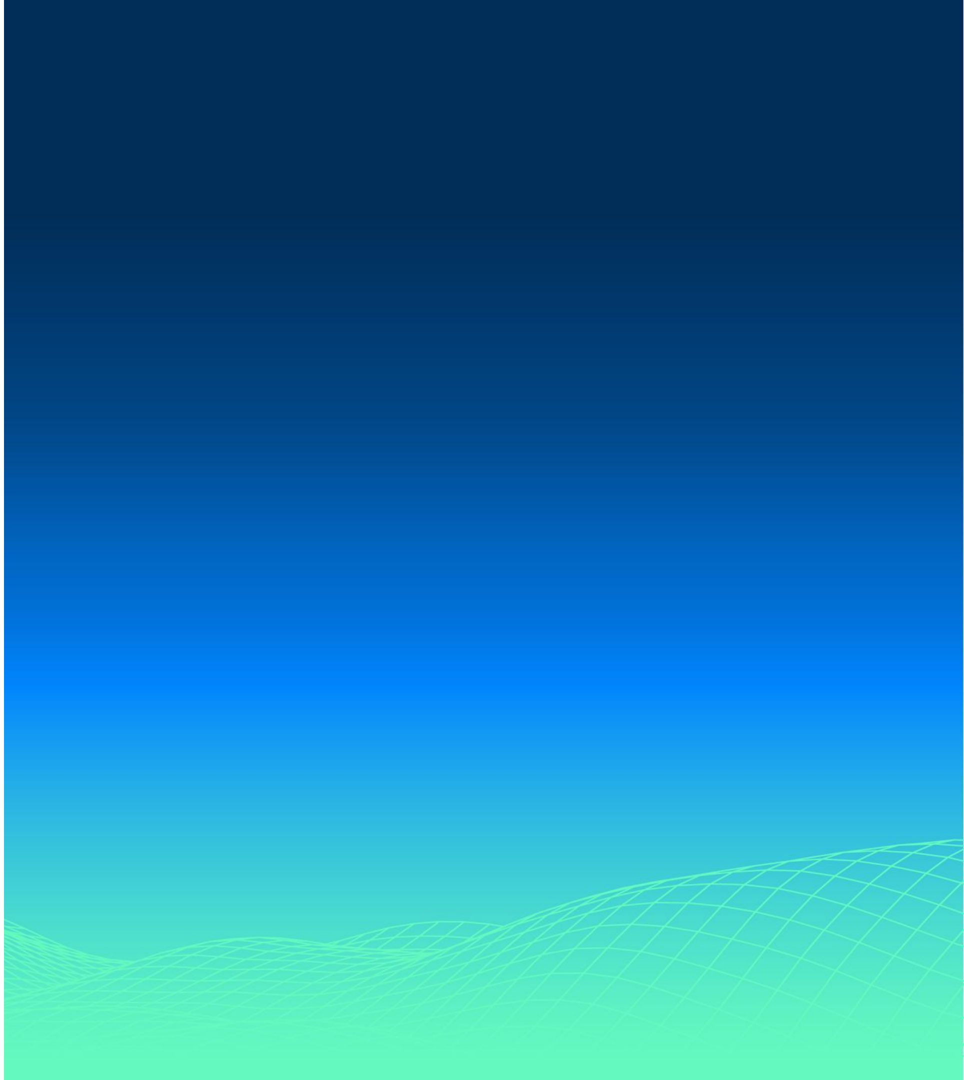
@hawkw

@eliza

Topics

- → What is Linkerd?
- → Why Linkerd enables failover support
- → The linkerd-failover extension
- → How it works
- → Walkthrough

What is Linkerd?

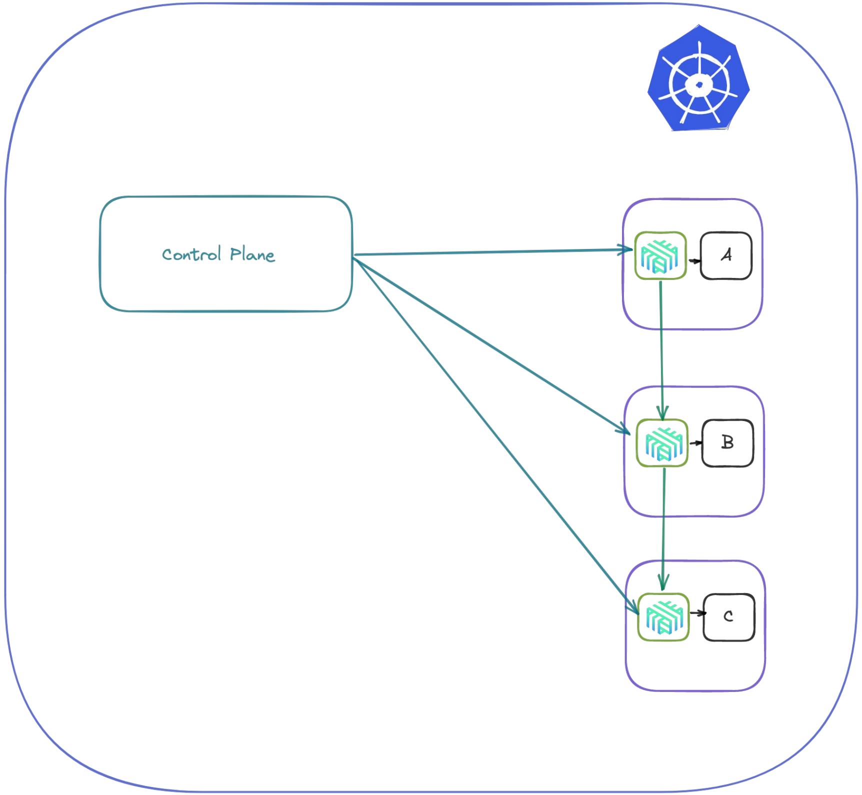


Linkerd is a service mesh

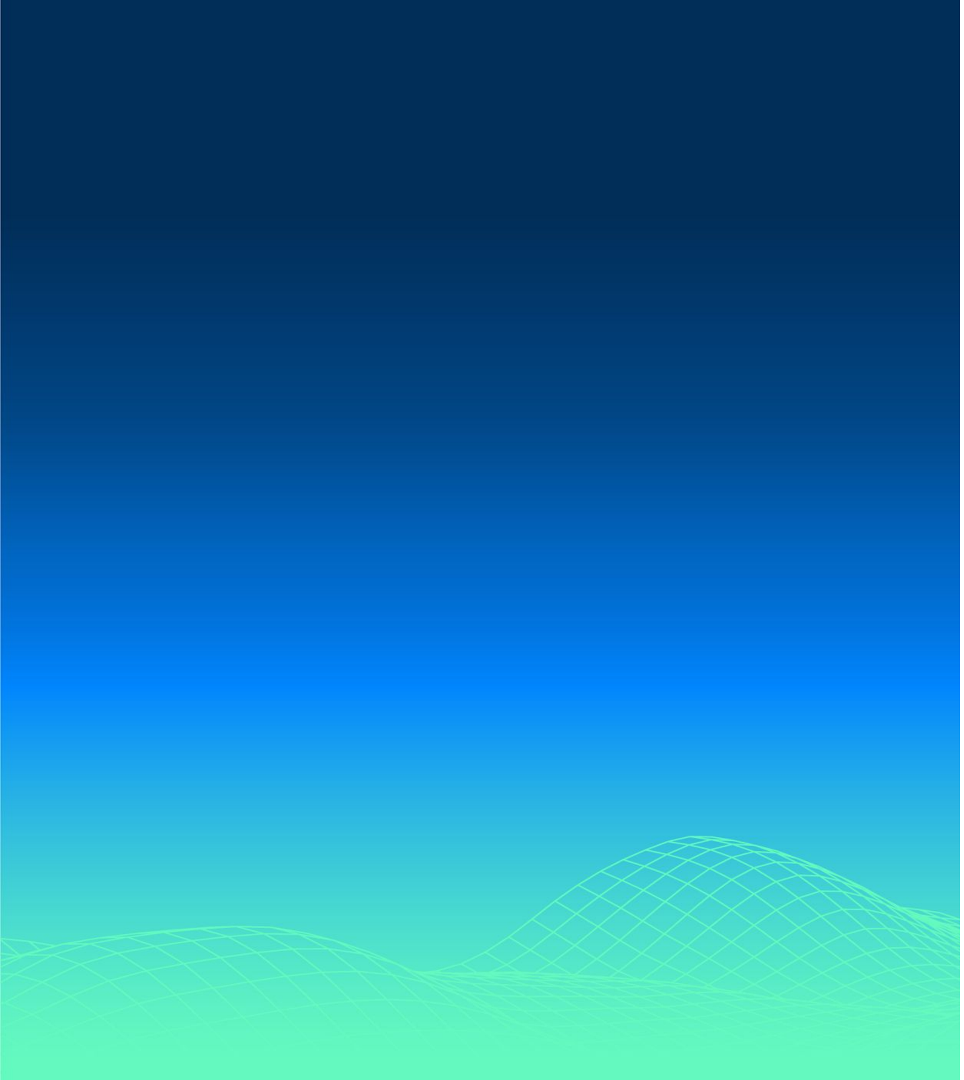
- Observability
- Security
- Reliability

The failover extension is a category of reliability because it ensures that requests are handled and that overall uptime is maintained

Linkerd Architecture



BUOYANT

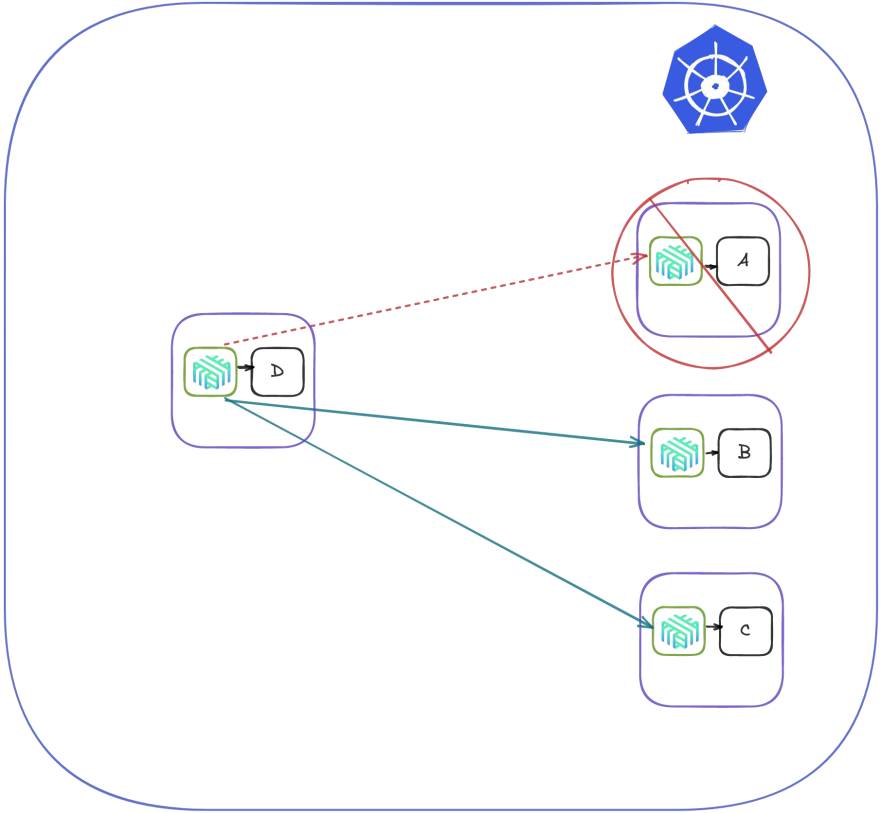


Failover

- → What is failover?
- → Why use Linkerd?

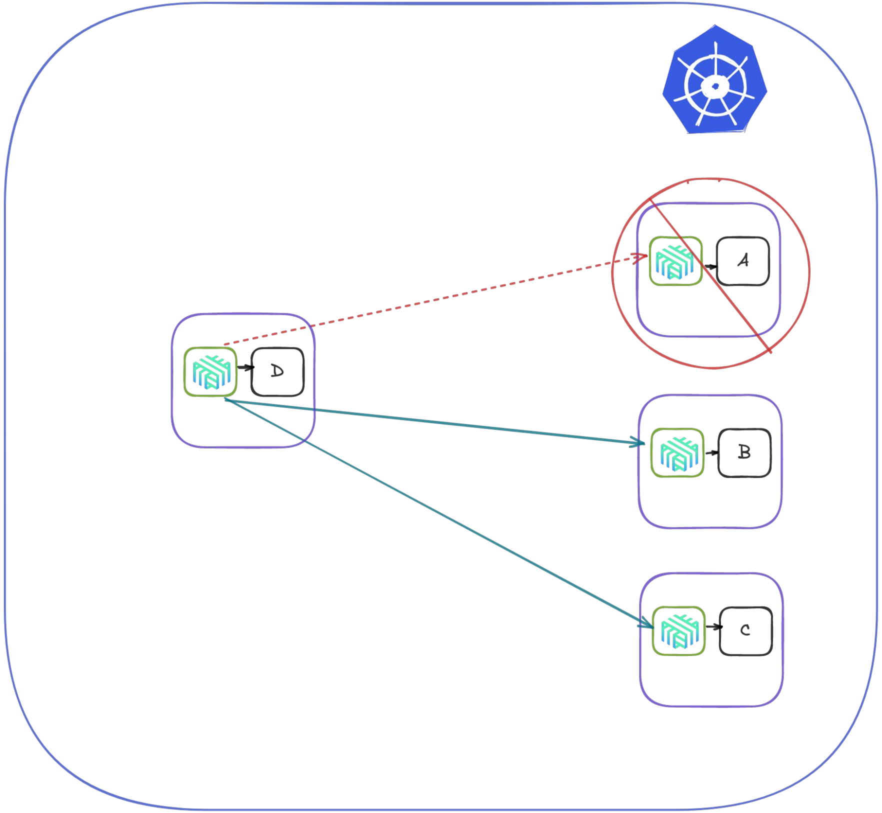
What is "failover"?

- Part of a broader category of circuit breaking
- A pattern that has been in practice for a looooooong time (there's nothing new here about what failover is, just the context)



What is "failover"?

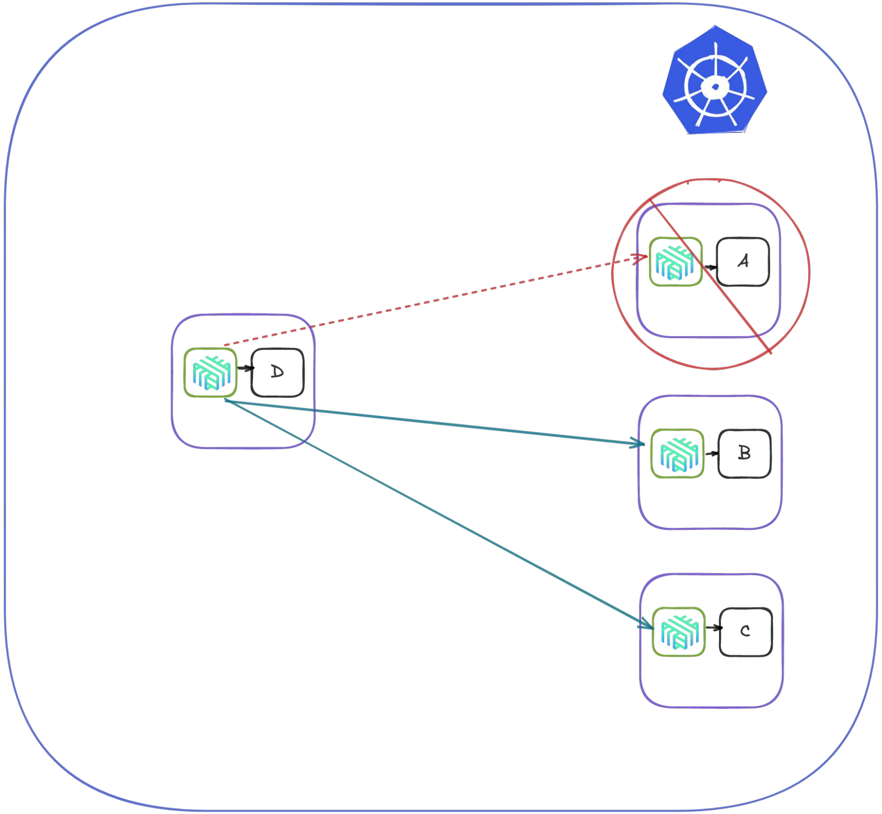
- A is the "primary" service
- If there are no Endpoint resources for A, services B and C act as failover services for traffic from D



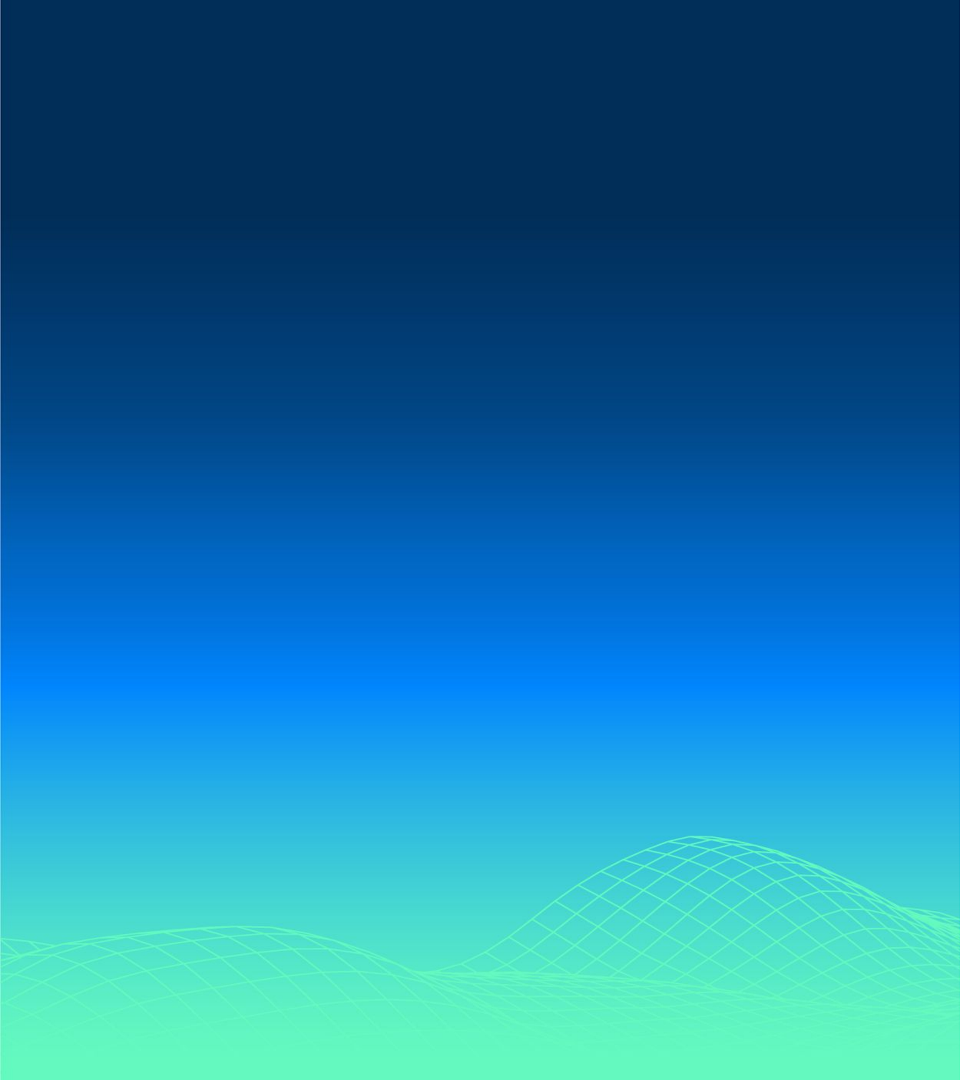
BUOYANT

Why use Linkerd?

- The Linkerd proxies are positioned perfectly to make routing decisions for failover
- Configuration is simple through the TrafficSplit resource provided by the Service Mesh Interface Specification



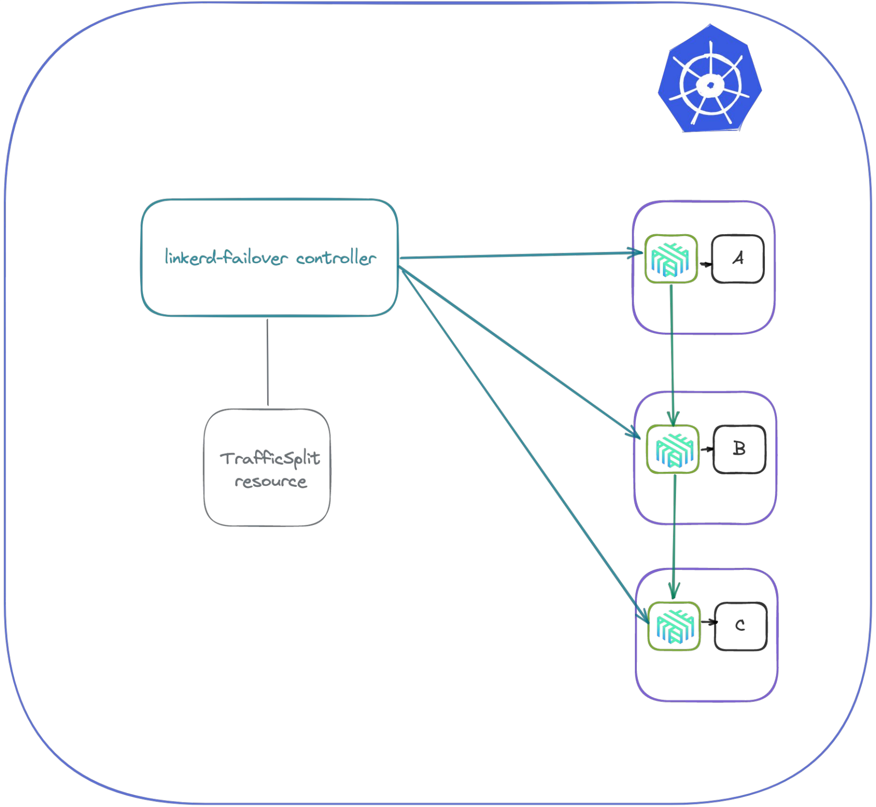
BUOYANT



Linkerd-failover extension

The linkerd-failover extension

- ➔ Controller written in Rust that watches for TrafficSplit resources with the `failover.linkerd.io/controlled-by: linkerd-failover` label
- ➔ Proxies are configured with details about how to handle traffic in the event that a service becomes unavailable



BUOYANT

TrafficSplit YAML for failover

```
apiVersion: split.smi-spec.io/v1alpha2
kind: TrafficSplit
metadata:
    name: sample-svc
    annotations:
        failover.linkerd.io/primary-service: A
    labels:
        failover.linkerd.io/controlled-by: linkerd-failover
spec:
    service: A
    backends:
        - service: A
          weight: 1
        - service: B
          weight: 0
        - service: C
          weight: 0
        - service: B-east
          weight: 0
        - service: C-east
          weight: 0
```

Let’s try it!

https://github.com/BuoyantIO/service-mesh-academy

Questions?

BUOYANT'S SERVICE MESH ACADEMY

The creators of LINKERD

Monthly hands-on, engineer-focused training from the creators of the service mesh

Up next (8-18): **Service mesh observability with Linkerd**

Sign up at
buoyant.io/service-mesh-academy

BUOYANT

The creators of LINKERD

Thank you!

Eliza Weisman
Linkerd Maintainer, Buoyant
eliza@buoyant.io

@BuoyantIO

buoyant.io