



BUOYANT

The creators of  
**LINKERD**



# Linkerd and ingress controllers: bringing the outside world in

Flynn, Tech Evangelist

[flynn@buoyant.io](mailto:flynn@buoyant.io) | [@flynn on slack.linkerd.io](#)



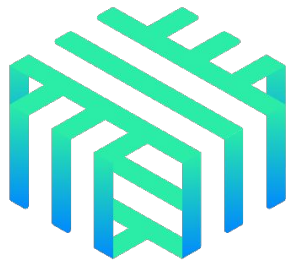


KubeCon



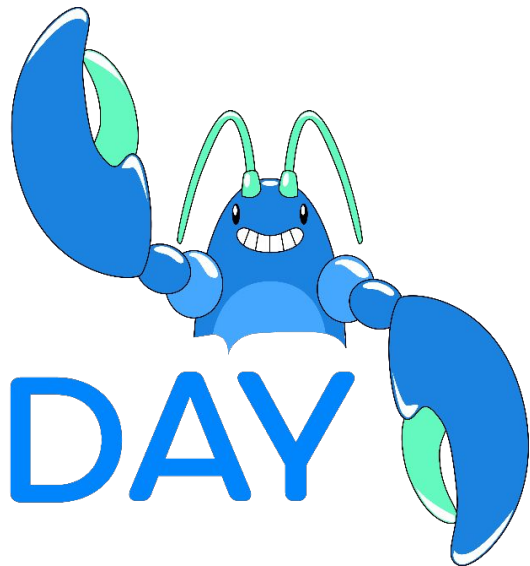
CloudNativeCon

Europe 2023



# LINKERD DAY

## EUROPE



BUOYANT

# What's on the agenda?

- The Ingress Problem™
- How it's solved
- How Linkerd works with ingress controllers
- **Demo!**

# What's needed for the hands-on part?

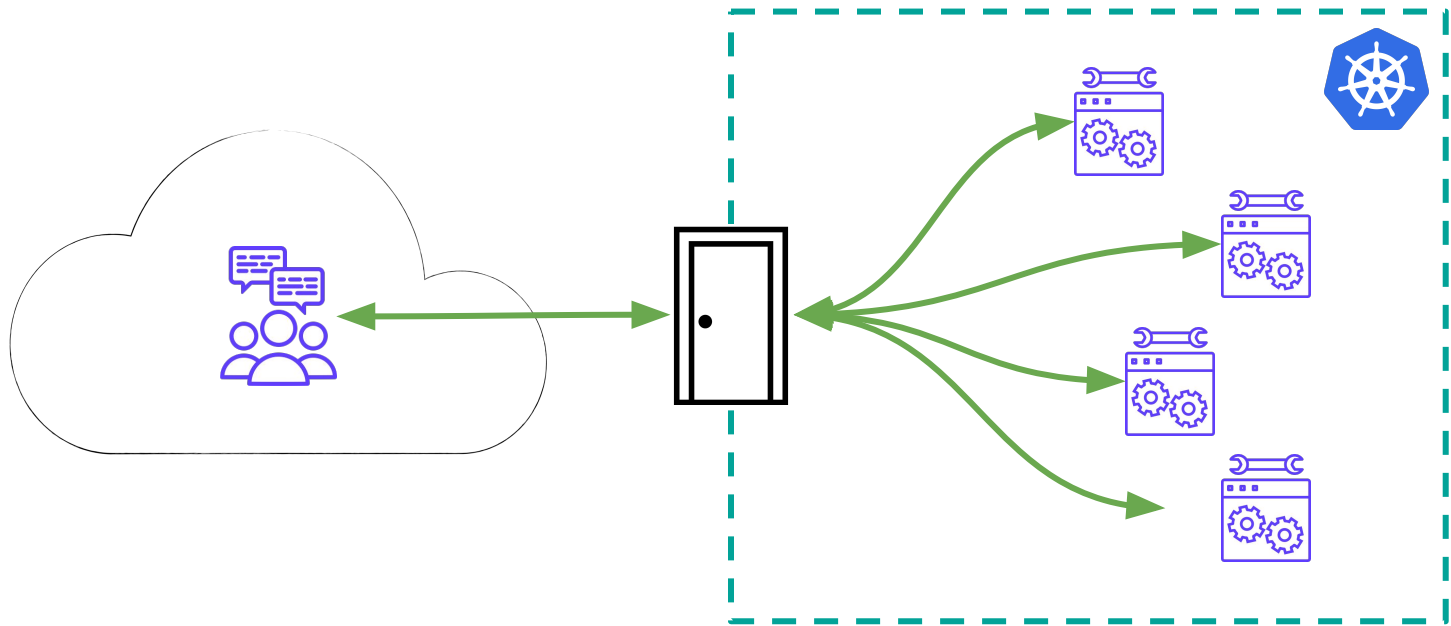
- A Kubernetes cluster, ideally supporting services of type LoadBalancer.
- To follow along, you'll also need `kubectl`, `linkerd`, and `helm`.
- Check out the workshop source at

<https://github.com/BuoyantIO/service-mesh-academy/tree/main/linkerd-and-ingress>

# The Ingress Problem



# How do you let people use your services?



# Ingress controller and service mesh

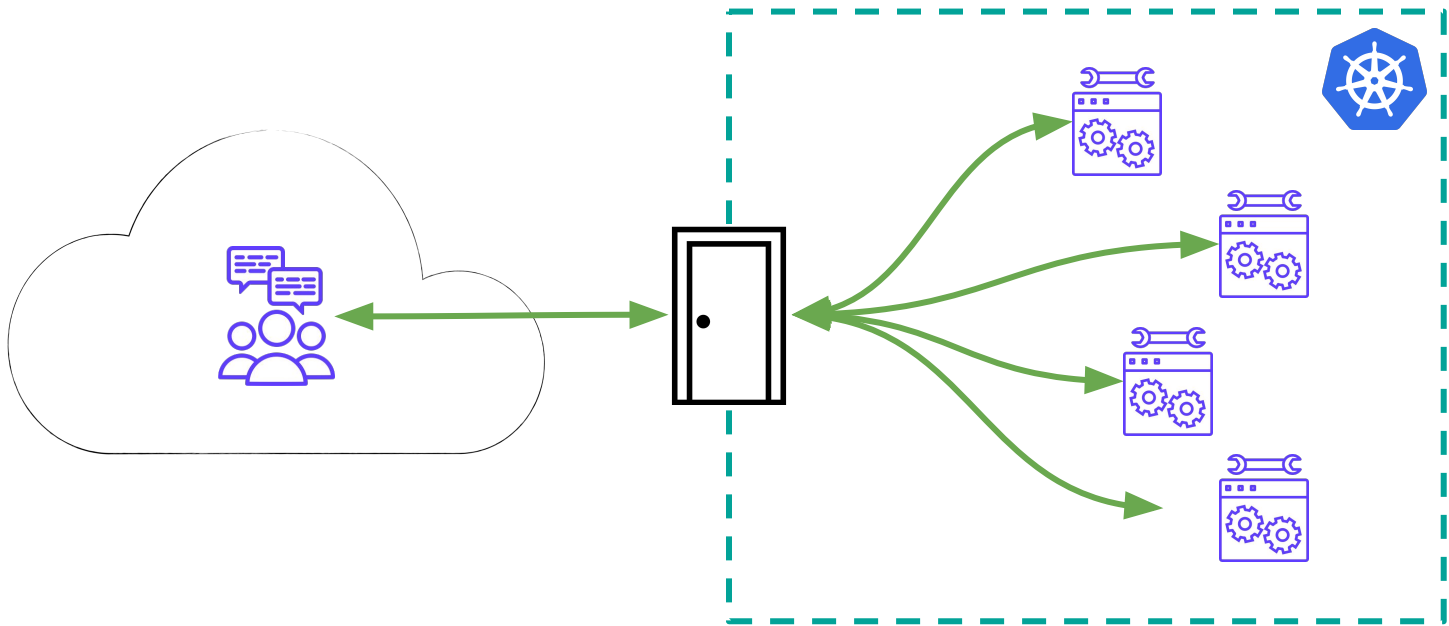
- The ingress controller provides **control** as well as **access**.
- Both the ingress controller and the service mesh can provide security, reliability, and observability
  - ◆ e.g. both can be great at gRPC load balancing
  - ◆ (when doing things like this, you tend to call it an API gateway instead of just an ingress controller)
- **However**. The ingress controller can **only** affect things at the edge of the cluster; the mesh has influence **everywhere**.
  - ◆ This is why you generally want both for real-world production use.

# A note: ingress **concept** != Ingress **resource**

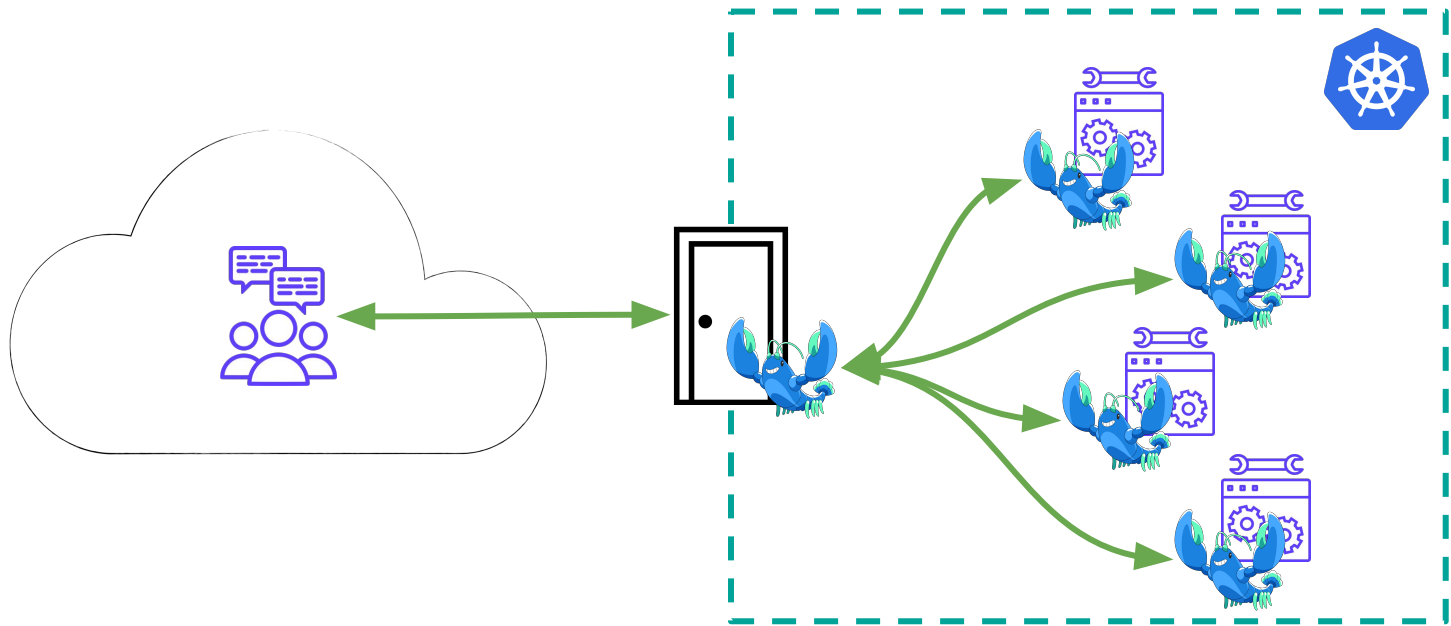
- Kubernetes has a resource called Ingress
  - ◆ It's old and limited; the Gateway API is meant to replace it
- "Ingress" as a concept means the function of carefully providing a mechanism for traffic from outside the cluster to find its way into the cluster.
  - ◆ This does **not** necessarily mean that you're using the Ingress resource.



# How does Linkerd work with ingress controllers?



# How does Linkerd work with ingress controllers?



# Ingress Controller Gotchas



# Skipping Incoming Ports

- If your ingress controller needs to know the IP address of incoming connections, you must tell Linkerd to skip the incoming port.
  - ◆ If you don't, the ingress controller will only ever see the Linkerd sidecar proxy's IP address.
- Skipping the incoming ports is always harmless, since the **purpose** of the ingress controller is managing connections from outside.

# Skipping Incoming Ports

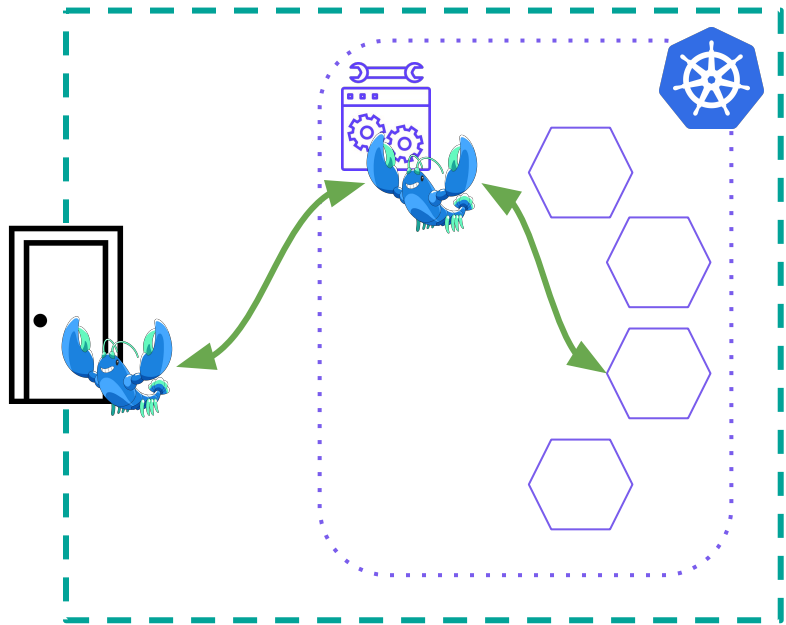
→ Do this with an annotation on the ingress controller Pods:  
`config.linkerd.io/skip-incoming-ports: 8080,8443`

- **Very important:** skip the targetPort, not the Service port!
- ◆ For the Service shown, you'd need to skip 8080, not 80

```
---
apiVersion: v1
kind: Service
metadata:
  name: fancy-ingress
spec:
  ports:
    - name: http
      port: 80
      targetPort: 8080
  selector:
    app: fancy-ingress
```

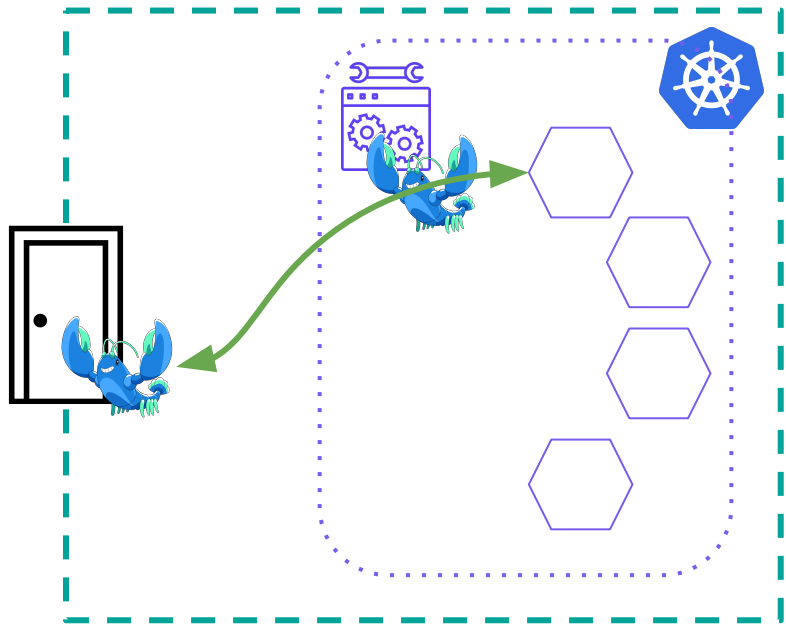
# “Ingress Mode”

- Linkerd expects connections to the *cluster IP* for a workload.
- When connecting to the cluster IP, Linkerd’s load balancing gets to choose which of the workload’s endpoints to use.



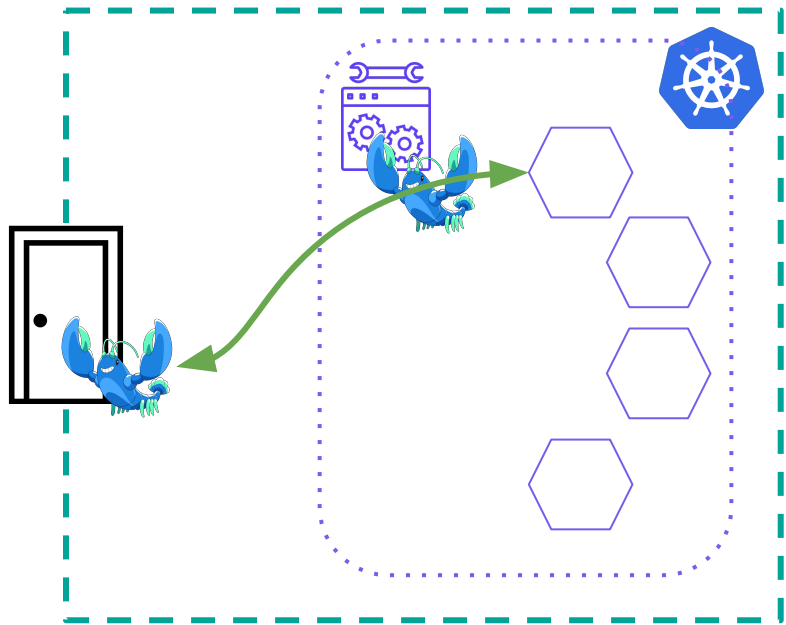
# “Ingress Mode”

- A connection straight to an endpoint will bypass Linkerd's load balancing.
- Sometimes this is what you want (e.g. Emissary using its endpoint resolver).



# “Ingress Mode”

- If not (e.g. Istio Gateway), use ingress mode!
- Annotate the ingress Pod with `linkerd.io/inject: ingress` instead of `linkerd.io/inject: enabled`
- **Note:** you **don't** want to use this if you don't really have to.





# Demo Time!



NGINX





# Q&A



BUOYANT'S

SERVICE MESH  
ACADEMY

The creators of  
LINKERD

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

Jan 24

Intro to GitOps with Weave GitOps, Flagger and Linkerd (hosted by Weaveworks)

Feb 16

Real-World GitOps with Flagger and Linkerd

**SIGN UP TODAY!**

[buoyant.io/sma](https://buoyant.io/sma)



# Fully managed LINKERD on any Kubernetes cluster

Buoyant Cloud automated upgrades, data plane  
version tracking, mesh health alerts, and much,  
much more.

**BOOK A DEMO**

[buoyant.io/demo](https://buoyant.io/demo)





# Thanks so much!

Flynn  
Tech Evangelist, Buoyant  
[flynn@buoyant.io](mailto:flynn@buoyant.io)

