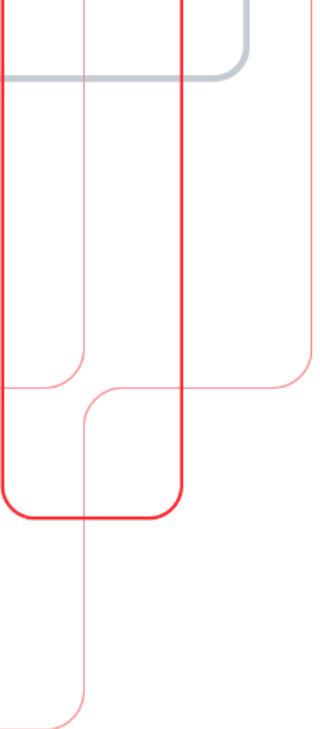


The logo for Fastly, featuring the word "fastly" in a bold, red, sans-serif font. The letter "a" has a small, white clock icon integrated into its center. A registered trademark symbol (®) is positioned at the top right of the letter "y".



OBSERVING FASTLY'S NETWORK AT SCALE THANKS TO K8S AND THE STRIMZI OPERATOR

Agenda

- 
-
- Introduction
 - Project and platform overview
 - Solution description
 - Problems, issues and learnings
 - Closure

INTRODUCTION



Fernando Crespo Grávalos

- Staff SREngineer @ **Fastly**
- Previously Senior SRE@**Tuenti, NAGRA, Bitnami...**
- OSS contributions (see [tuenti secrets manager](#), [instaclustr_exporter](#))
- *Medium blog posts*



Daniel Caballero Rodríguez

- Principal SREngineer @ **Fastly**
- Previously part time (Devops) lecturer, Devops@**Schibsted** (now Adevinta), **NTT**, **Oracle**...
- *Eventual OSS contributions (see [tcpgoon](#))*



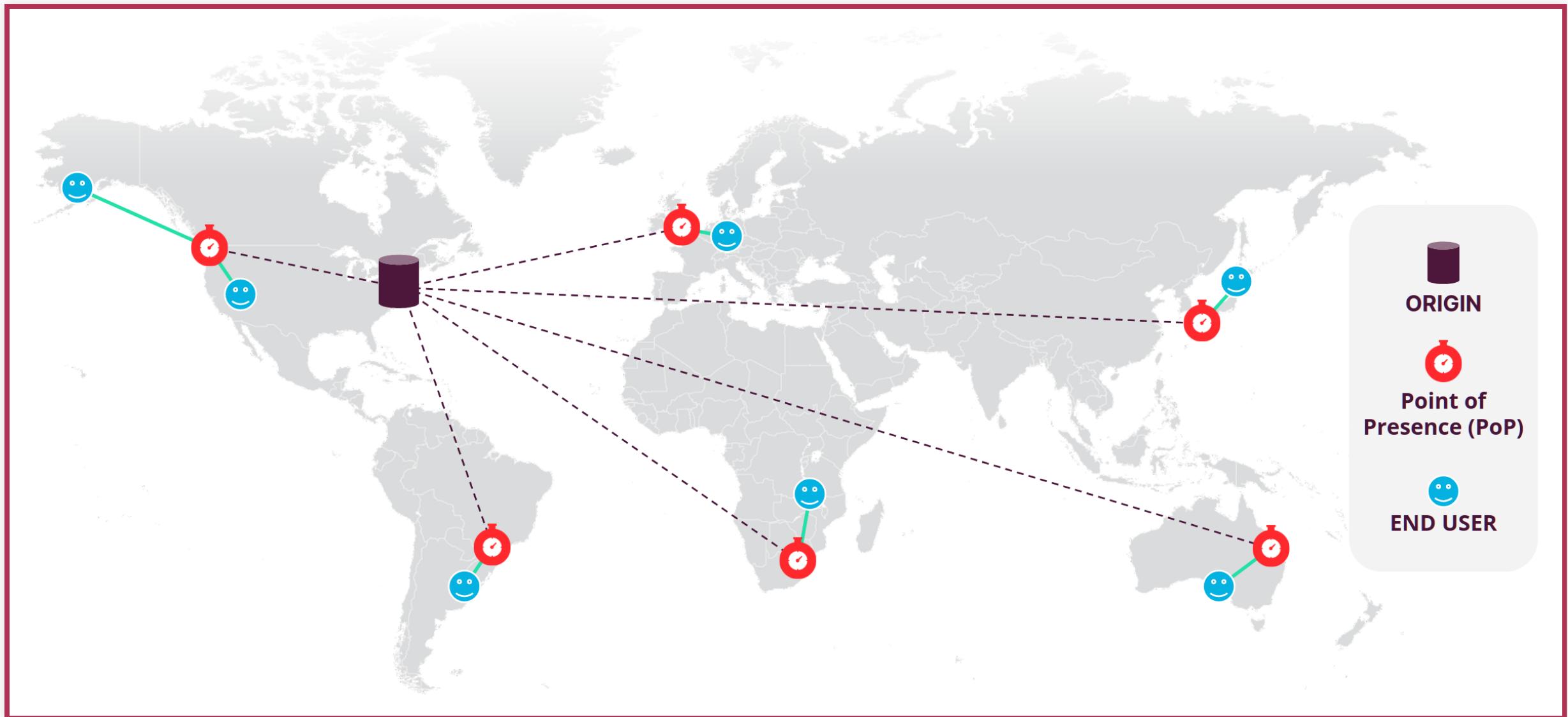
WHAT IS THIS TALK ABOUT?

How K8s and the Strimzi Kafka Operator helped implementing a scalable
telemetry pipeline, core for traffic management automation

The Fastly logo is displayed in a large, bold, red sans-serif font. The letter 'a' is stylized with a small clock face on its left side, and a registered trademark symbol (®) is positioned at the top right of the 'y'.

fastly®

OUR BUSINESS



WHAT MAKES US UNIQUE

Software Defined Network

Programmable Edge

Customer Empowerment Philosophy

Edge Advantages

- Better performance
- Instant scale
- Better UX
- Greater control
- More agility
- Heightened security
- CI/CD integration
- Real-time visibility
- Reduce TCO
- Lower demand on origin
- Reduce time to deploy

OUR CUSTOMERS ACROSS INDUSTRIES



Blackboard®



BuzzFeed

The New York Times



Etsy



CONDÉ NAST

Pinterest



Bankrate®

e EUROSTAR™

PAYCHEX
HR | Payroll | Benefits | Insurance



YOTTA

FASTLY AT A GLANCE



800B+

Requests served daily¹



192 Tbps

Edge capacity²



95%+

Customer satisfaction score³



2,800+

Customers⁴



\$354.3M

2021 Revenue



99%

Annual Revenue Retention⁵

1. As of December 30, 2020

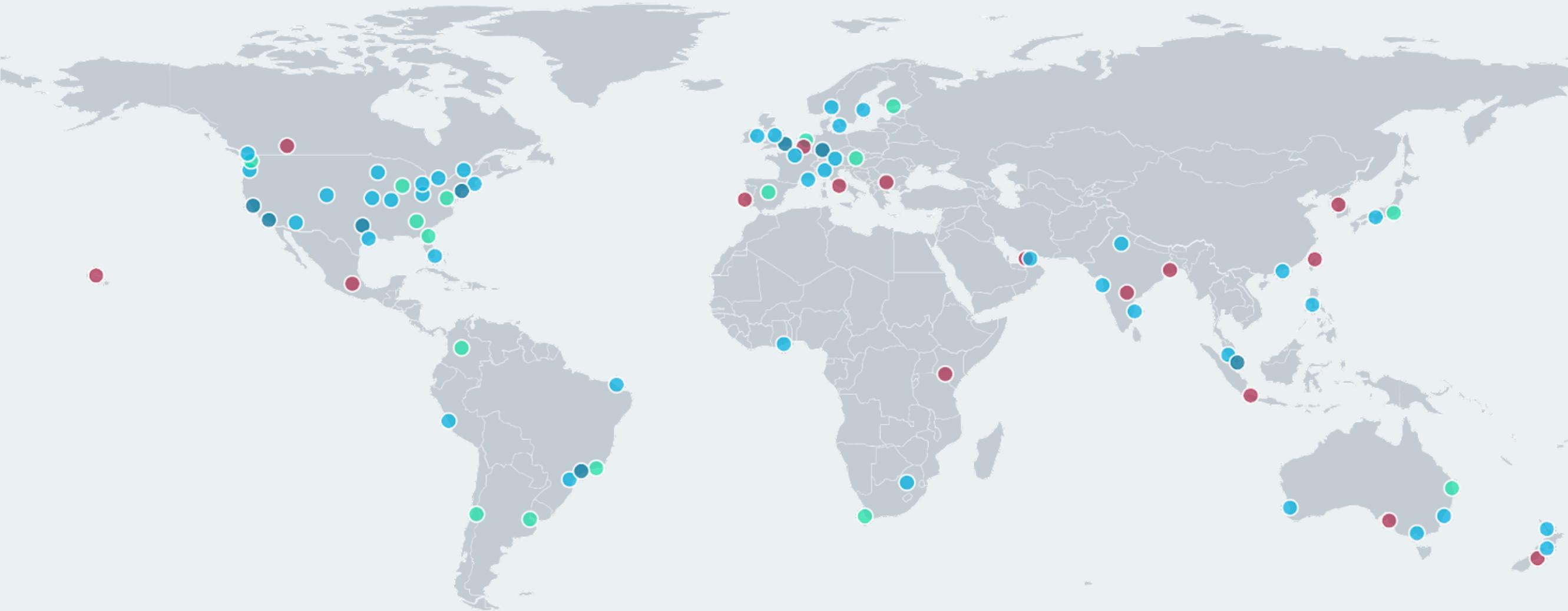
2. As of January, 2022

3. As of February 1, 2022

4. As of February 16, 2022

5. As of February 16, 2022

Fastly Points-of-Presence (POPs)



BUT THERE IS MORE

We have a control plane

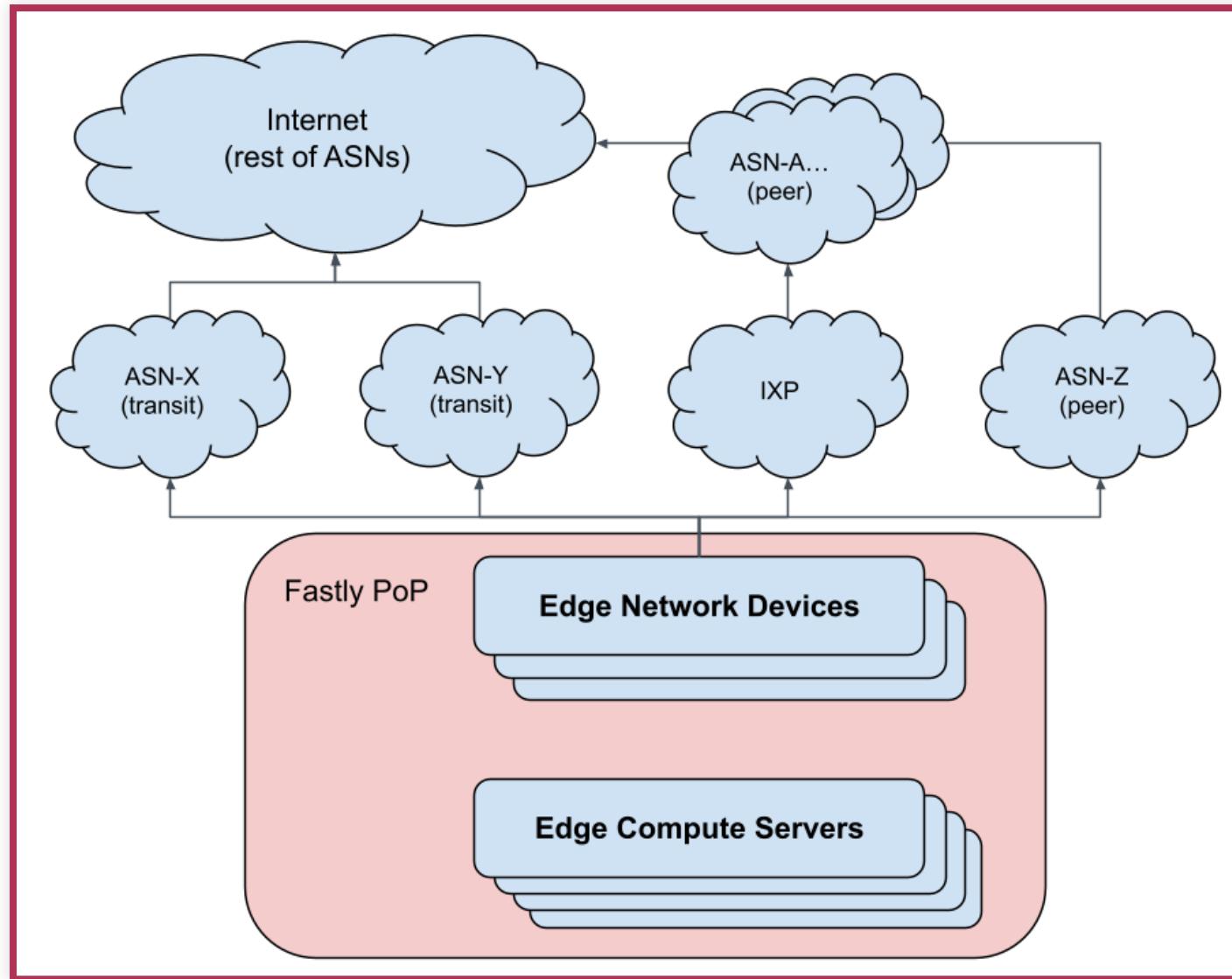


PROJECT AND PLATFORM OVERVIEW



THE PROJECT: AUTOPILOT AND ITS TELEMETRY PIPELINE

CONTEXT: FASTLY AND EGRESS TRAFFIC ENGINEERING



Automation in this problem domain is not a new concept...

How network automation helps Fastly support the world's biggest live-streaming moments



Ryan Landry

Vice President, Technical Operations

NETWORK

PERFORMANCE

REAL-TIME INSIGHTS

STREAMING

Published March 10, 2020

One of the keys to a clean, clear live-streaming experience is properly managing network congestion — something our platform performs mostly automatically, around the clock, and with reduced human intervention. As part of Fox Sports' live event streaming team for [Super Bowl LIV](#), we were able to watch this automation

Want to continue the conversation?

[Schedule time with an expert](#)

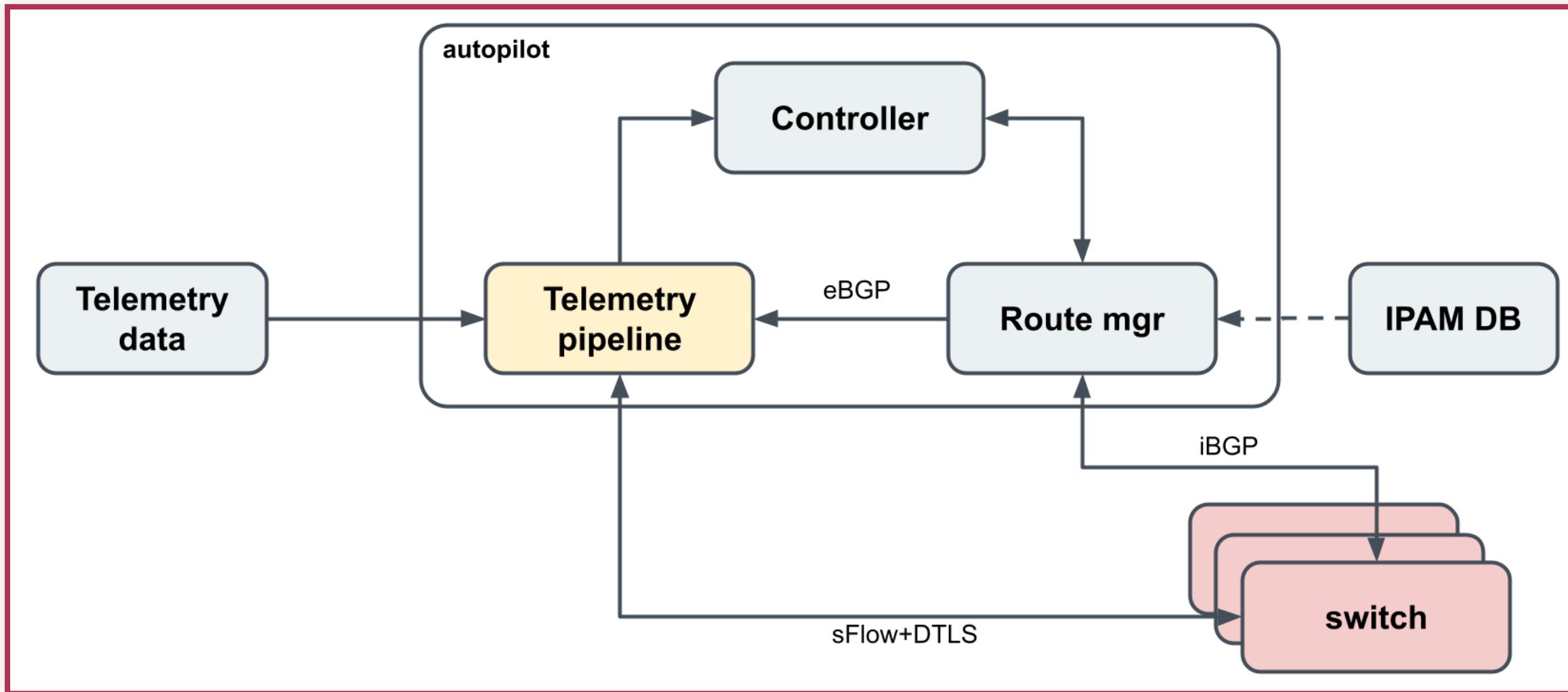
<https://www.fastly.com/blog/network-automation-helps-support-worlds-biggest-live-streaming-moments>

AUTOPilot AS THE LAST ITERATION

- Developed and maintained by the *Network Control and Optimization (NCO)* team
- Initially around 6 people...
 - across 4 different countries
 - and different profiles (Software Engineers, Data Engineers, Network Engineers, SRE)

Note: Tech decisions need to factor team skills and size

AUTOPILOT AT HIGH LEVEL



THE TELEMETRY PIPELINE

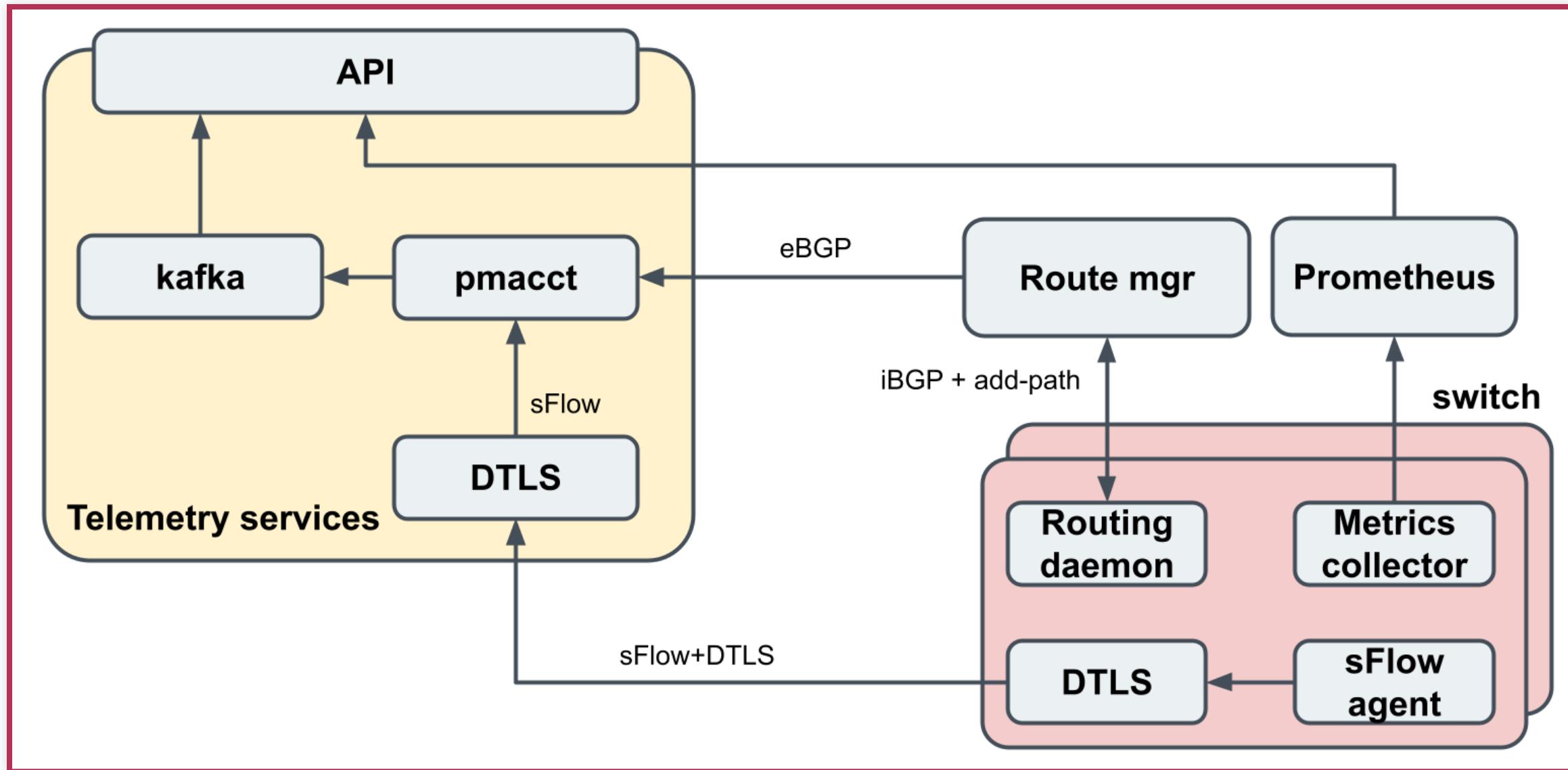
We need to process (lots of) telemetry data...

- Systems metrics
- Flow data (sampling packet headers from our fleet)
- Performance data
- Synthetic traffic / connectivity checks results

... and Autopilot is not the only use case:

- Our NetOps team also gets PoPs traffic insights
- Capacity planning
- New DDoS solutions
- Blog posts :)
- Others we may not even anticipate yet

PIPELINE DESIGN OVERVIEW



IMPLEMENTATION REQUIREMENTS

IMPLEMENTATION REQUIREMENTS

To process telemetry and flow data at **high scale**

IMPLEMENTATION REQUIREMENTS

To process telemetry and flow data at **high scale**

Cloud-agnostic / multi-cloud / multi-region

IMPLEMENTATION REQUIREMENTS

To process telemetry and flow data at **high scale**

Cloud-agnostic / multi-cloud / multi-region

Runtime that let's us deploy/manage apps outside our fleet



We did not have a proper standard

- Very well known patterns to deploy components to our fleet / data-plane...
- ... the control plane was a different story
- NCO had dedicated GKE clusters for specific workloads
- And Kafka@Fastly was just in experimentation phase
 - There was actually opposition to introduce Kafka in the project scope

K8S@FASTLY CIRCA 2020

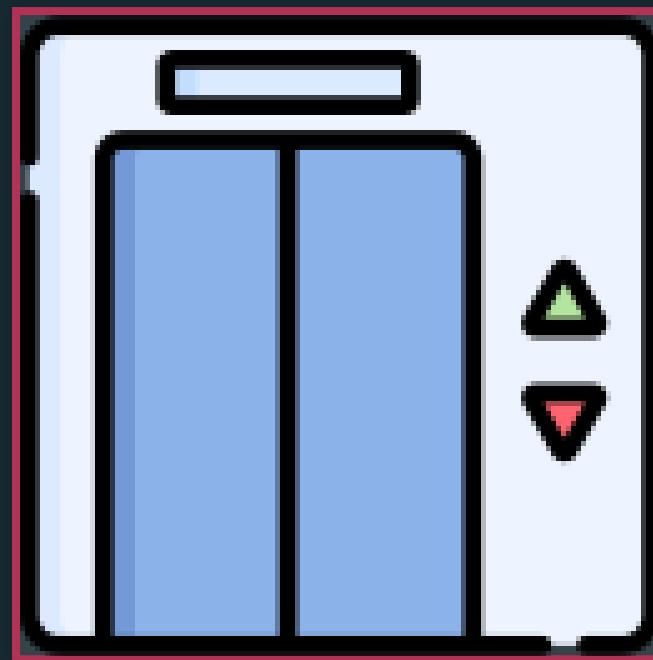


- Very fragmented infrastructure
- Mostly individual initiatives
- No standard way of creating k8s clusters or to deploy applications
- Engineers not necessarily versed on cluster administration
- Heavily used for CI

K8S TEAM

- Initially: New Zealand (2), Canada (1), Spain (1)
- Goals:
 - Build a k8s shared platform
 - Standardize infrastructure and deployments
 - Scale cross-regions, cross-clouds
 - **Elevate** developers capabilities to deploy and scale faster
- The latter brought the project name: **Elevation**

ELEVATION

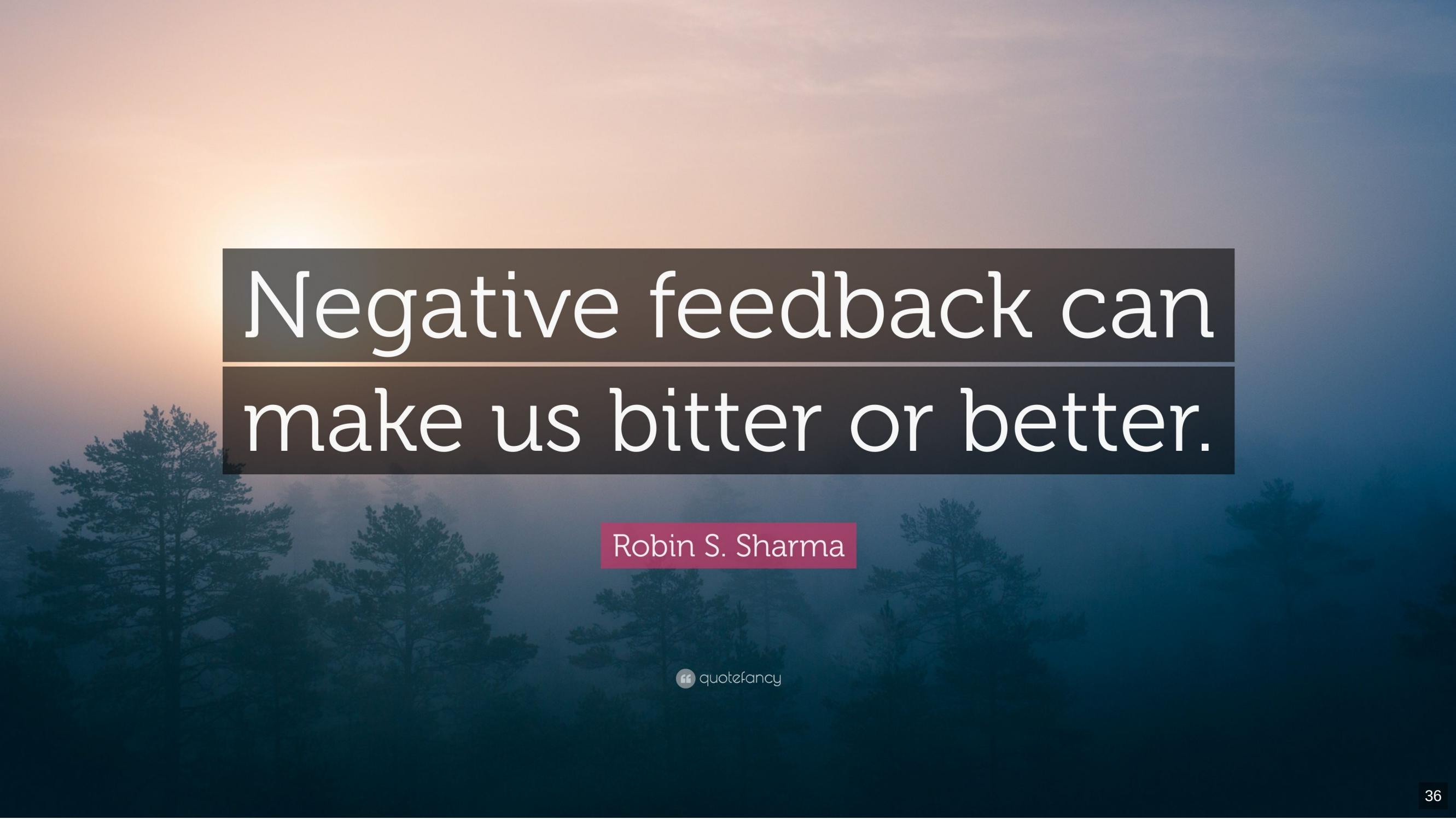


EL ELEVATION 1.0

- 3 Clusters: dev, staging, prod
- 1 region, but designed to span across regions
- 1 cloud, stack designed to be cloud agnostic:
 - Auth with our IdP, not tied to the Cloud IAM
 - Same deployment mechanism: Helm Charts via FluxCD (GitOps)
 - Harbor as Container/Helm Charts registry
 - Vault for secrets management
 - Nginx Ingress
 - Cert-Manager to issue certs from Lets Encrypt
 - Observability: Prometheus/Grafana, FluentD/Splunk
 - Service Mesh with Linkerd

ELEVATION 1.0 FEEDBACK

- Multi-Region, Multi-Cloud
- Reduce the onboarding overhead
- In-Cluster Kafka support
- Steep learning curve
- Seamless integration with our IdP for applications
- Improve service mesh observability



Negative feedback can
make us bitter or better.

Robin S. Sharma



quotefancy

ELLEVATION 2.0 - K8S @FASTLY TODAY

2 new team members: US, UK

More Cloud Providers and Regions

▼ Elevation Clusters - Locations, Providers

Elevation Clusters around the world (Only GOOGLE and AWS)

Leaflet | © OpenStreetMap © CartoDB

Elevation Clusters General Info				
Cluster	Provider	Country	Kubernetes Version	Container Runtime ↑
dev-usc1	GOOGLE	United States		
plat-usc1	GOOGLE	United States		
prd-euw2	GOOGLE	United Kingdom		
prd-usc1	GOOGLE	United States		
prd-asse1	GOOGLE	Singapore		
sbx-usc1	GOOGLE	United States		
stg-usc1	GOOGLE	United States		
prd-sl13	SOFTLAYER	United States		
prd-sl90	SOFTLAYER	United States		
sbx-sl13	SOFTLAYER	United States		

Clusters by Country

United States	13
Singapore	2
United Kingdom	1
France	1

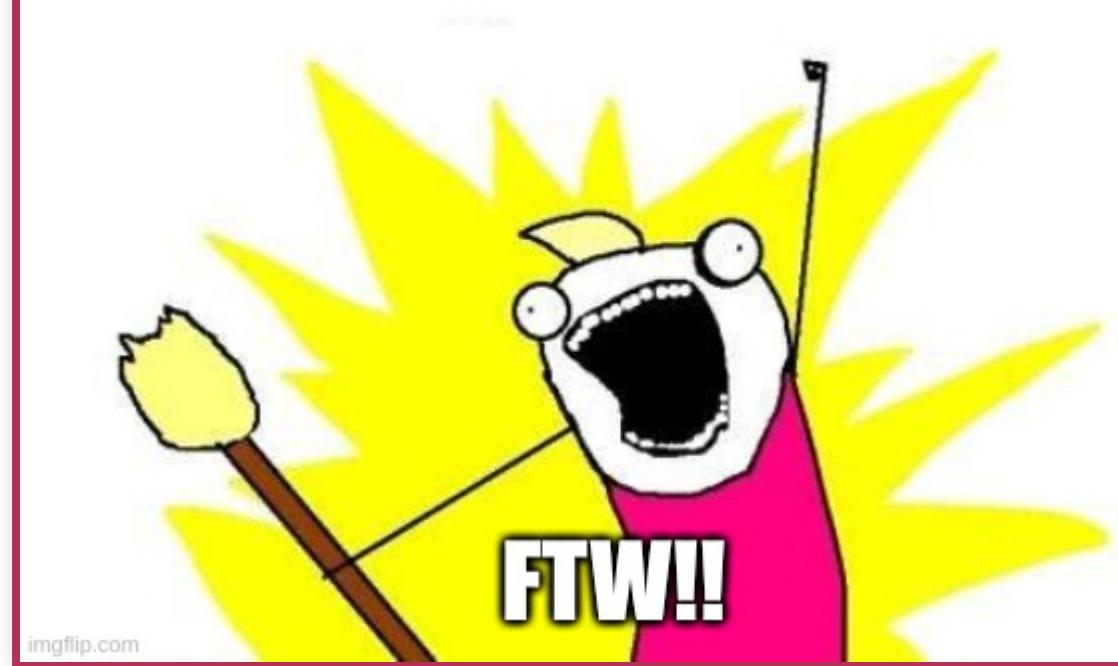
Clusters by Country/State

United States/Iowa	5
United States/Ohio	4
Singapore/	2
United States/Nevada	2
United Kingdom/England	1
United States/California	1
United States/	1
France/Ile-de-France	1

Clusters by Provider

GOOGLE	7
AWS	6
SOFTLAYER	4

K8S OPERATORS



imgflip.com

Onboarding Improvements

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
[...]
rules:
- generate:
    apiVersion: vault.fastly.com/v1beta1
    data:
        spec:
            mountInput:
                options:
                    version: "2"
                    type: kv
                    path: '{{request.object.metadata.name}}'
            kind: SecretEngine
            name: '{{request.object.metadata.name}}'
            namespace: vault
    match:
        any:
        - resources:
            kinds:
            - v1/Namespace
            selector:
                matchLabels:
                    gen-vault-kv-store: enabled
    name: vault-kv-store
```

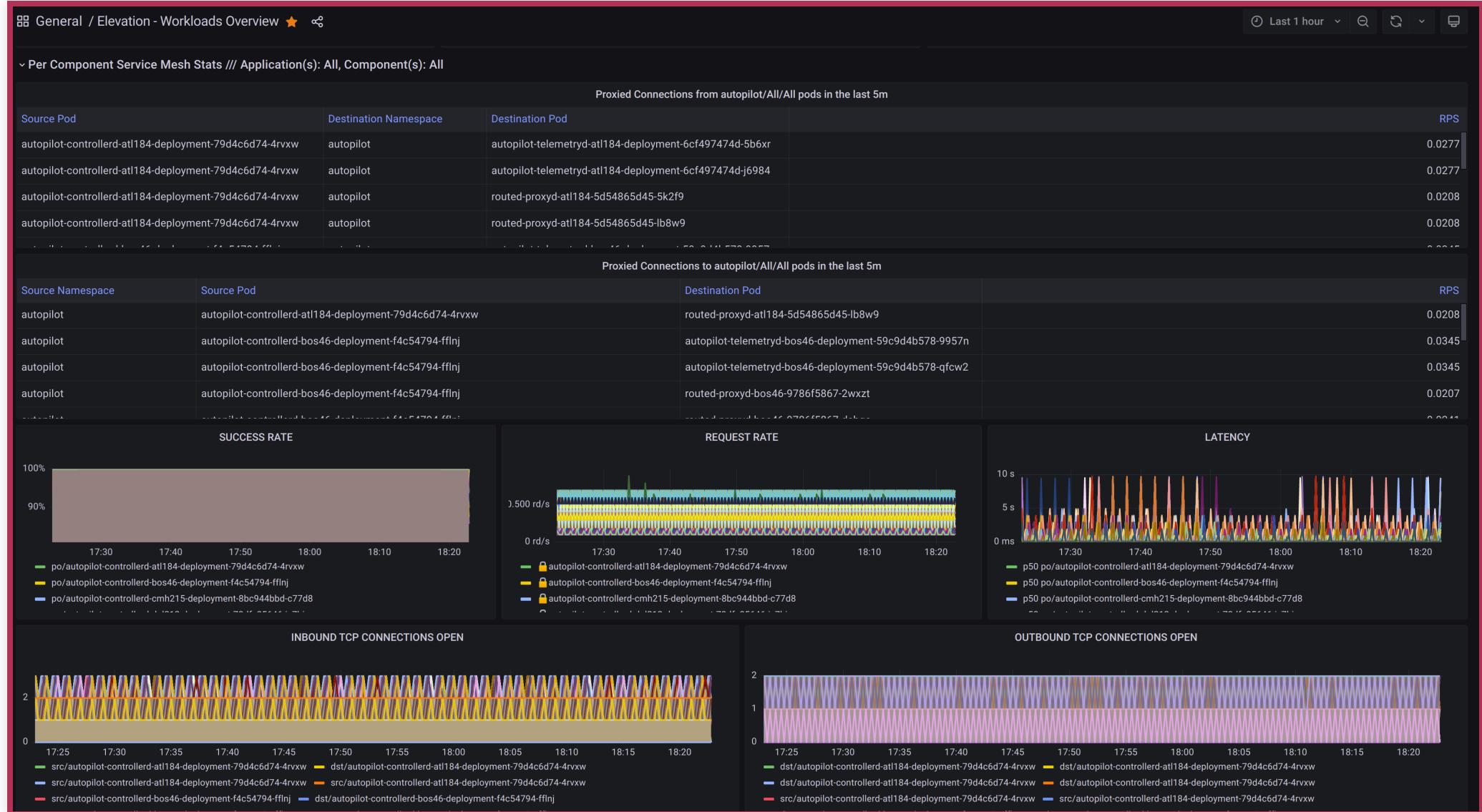
Idp auth as a Service

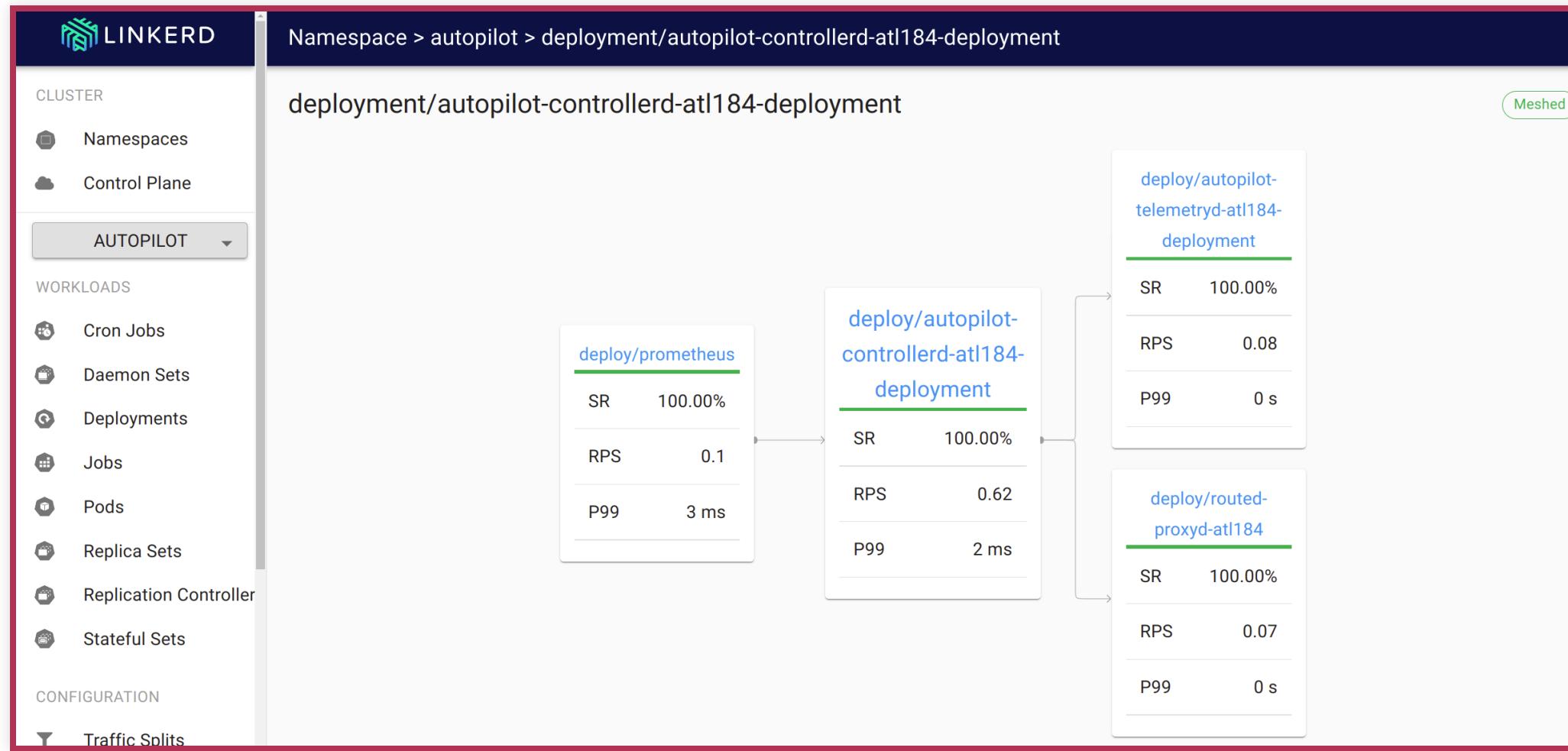
```
nginx.ingress.kubernetes.io/auth-signin: https://vouch.int.usc1.dev.k8s.example.com/login?url=$scheme://$http_host$request_uri&vouch-failcou
nginx.ingress.kubernetes.io/auth-url: https://vouch.int.usc1.dev.k8s.example.com/validate
nginx.ingress.kubernetes.io/auth-response-headers: X-Vouch-User
nginx.ingress.kubernetes.io/auth-snippet: |
    # these return values are used by the @error401 call
    auth_request_set $auth_resp_jwt $upstream_http_x_vouch_jwt;
    auth_request_set $auth_resp_err $upstream_http_x_vouch_err;
    auth_request_set $auth_resp_failcount $upstream_http_x_vouch_failcount;
```

Abstractions

```
app:  
  name: foo  
  components:  
    - name: example  
      containers:  
        - name: web  
          image:  
            repo: fastly/foo-example  
            tag: v1.21.0  
        ports:  
          - number: 8080  
            protocol: HTTP  
            expose: true  
        ingress:  
          http:  
            routes:  
              - host: foo.example.com  
                prefix: /
```

OBSERVABILITY IMPROVEMENTS





▼ Deployments status

Helm Charts - Deployment Status				Flux - Reconciliation Status		
chart	release	version	status	name	ready?	suspended?
routed-proxyd	routed-proxyd	0.0.9	Deployed	controllerd-prd-usc1	Yes	No
flow-proxy	flow-proxy	0.0.18	Deployed	routed-proxy-prd-usc1	Yes	No
telemetry-api	telemetry-api	0.2.11	Deployed	telemetry-api-prd-usc1	Yes	No
controllerd	controllerd	0.2.8	Deployed	telemetry-flow-proxy-prd-usc1	Yes	No
telemetry-sfac...	telemetry-sfacctd	0.0.20	Deployed	telemetry-kafka-prd-usc1	Yes	No
telemetry-kafka	telemetry-kafka	1.0.0	Deployed	telemetry-sfacctd-prd-usc1	Yes	No

In-Cluster Kafka support

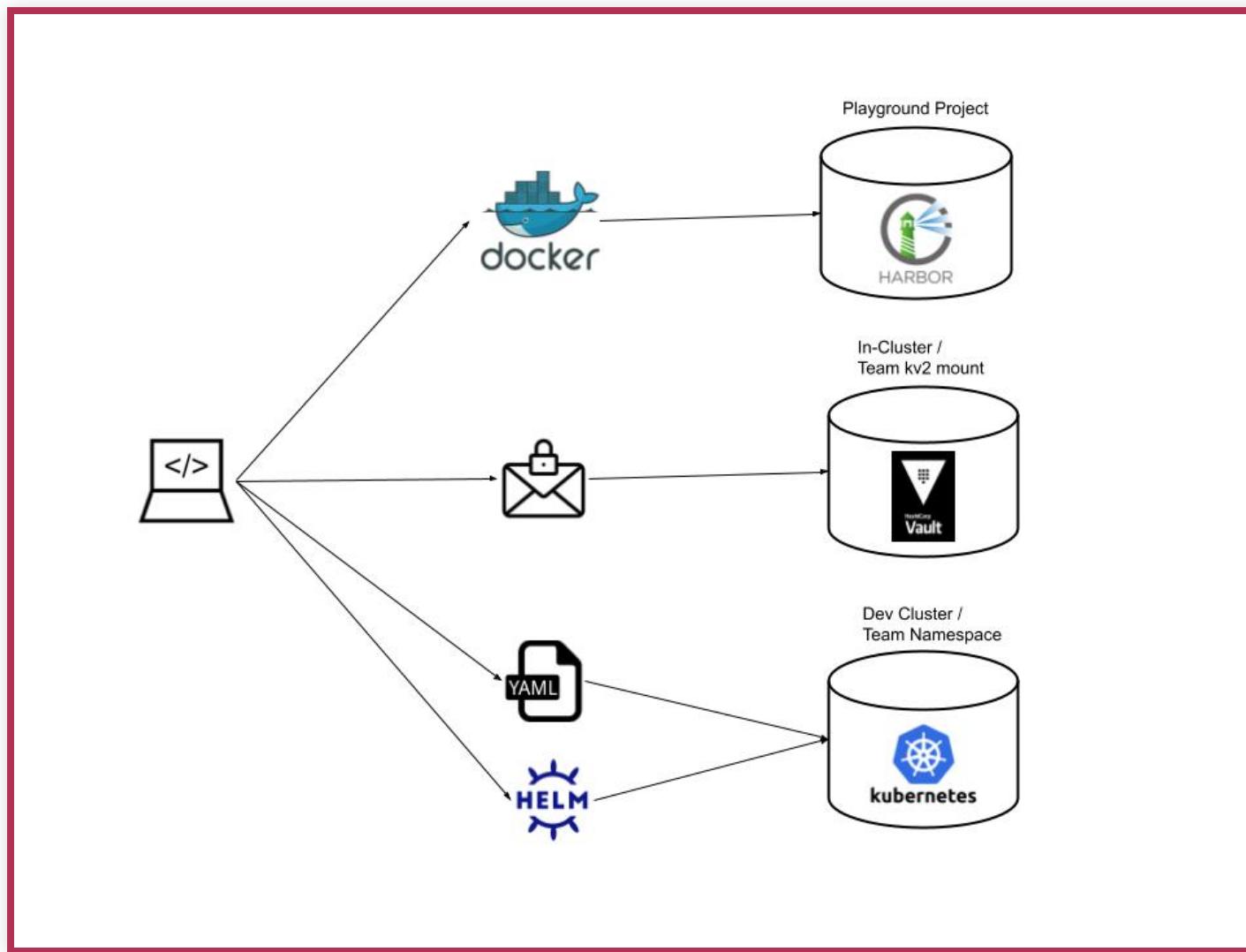


DEPLOYING ON ELEVATION: FROM DEV TO PROD

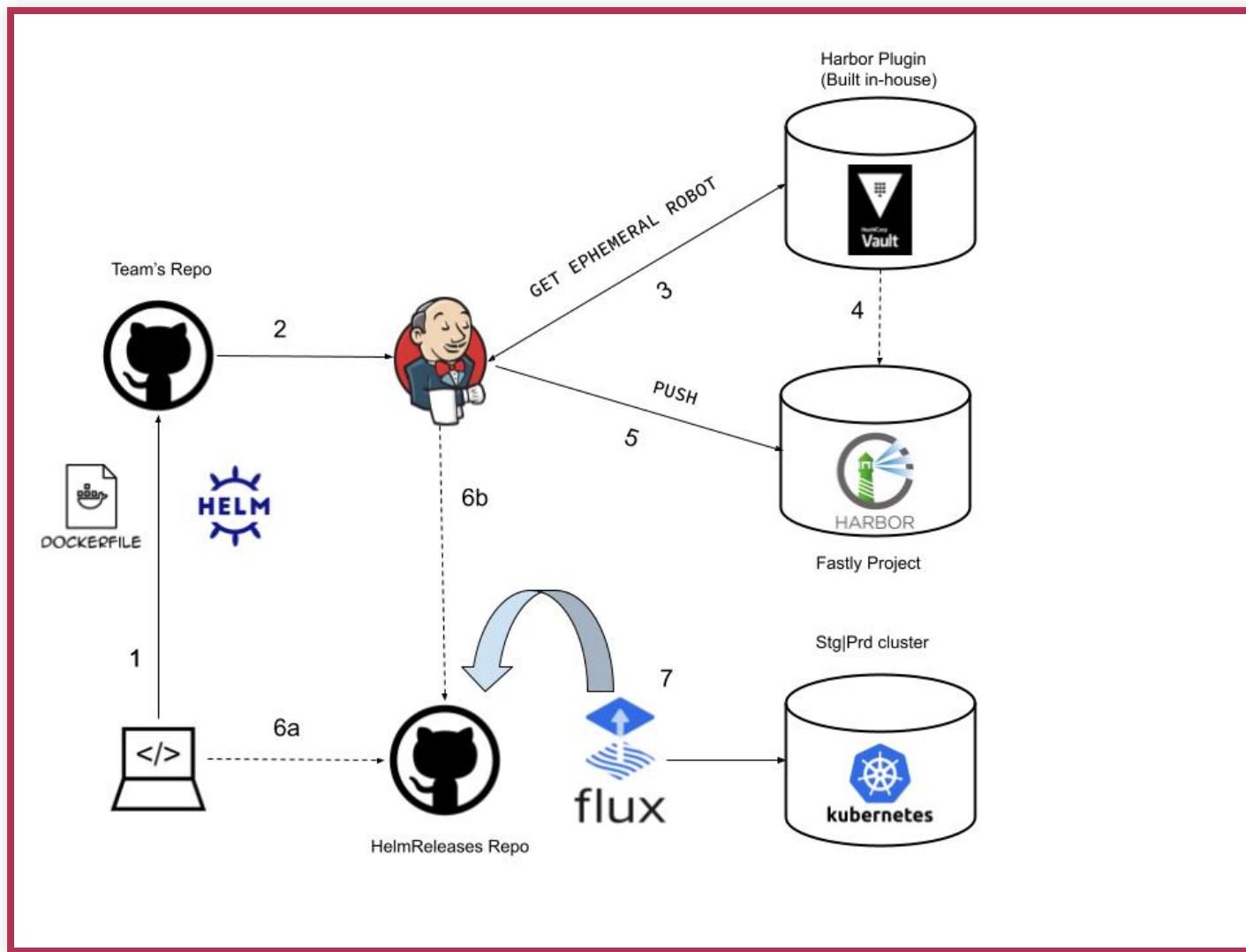


Explore, Build, Deploy

EXPLORE



BUILD AND DEPLOY



THE GOOD
BAD
UGLY

AND THE



THE GOOD



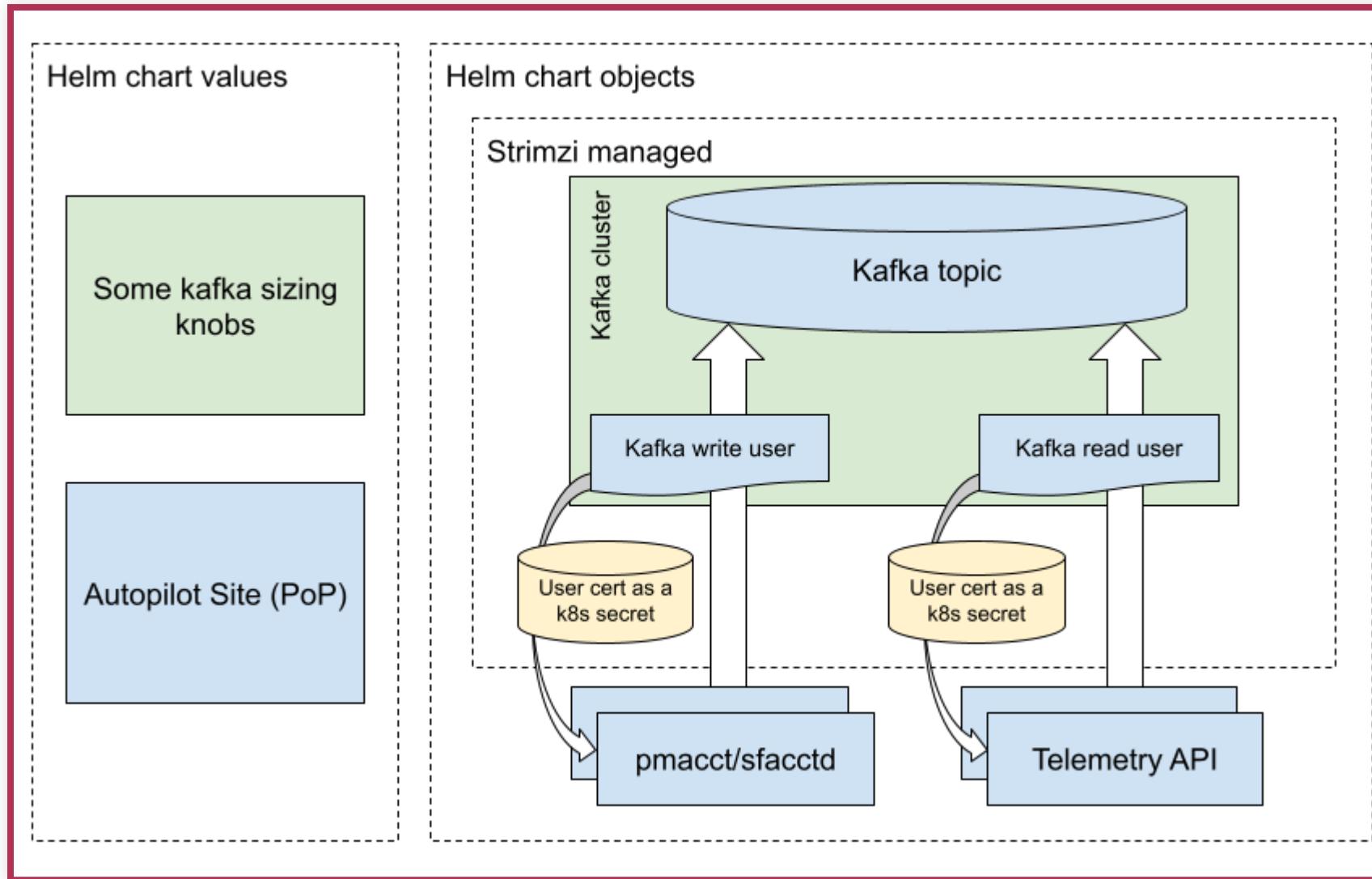


```
apiVersion: kafka.strimzi.io/v1beta2
kind: Kafka
metadata:
  name: my-cluster
spec:
  kafka:
    replicas: 3
    listeners:
      - name: tls
        port: 9093
        type: internal
        tls: true
        authentication:
          type: tls
    storage:
      type: jbod
    volumes:
      - id: 0
        type: persistent-claim
        size: 100Gi
  config:
    offsets.topic.replication.factor: 1
    transaction.state.log.replication.factor: 1
    transaction.state.log.min.isr: 1
    default.replication.factor: 1
    min.insync.replicas: 1
  zookeeper:
    replicas: 3
    storage:
      type: persistent-claim
      size: 100Gi
  entityOperator:
    topicOperator: {}
    userOperator: {}
```

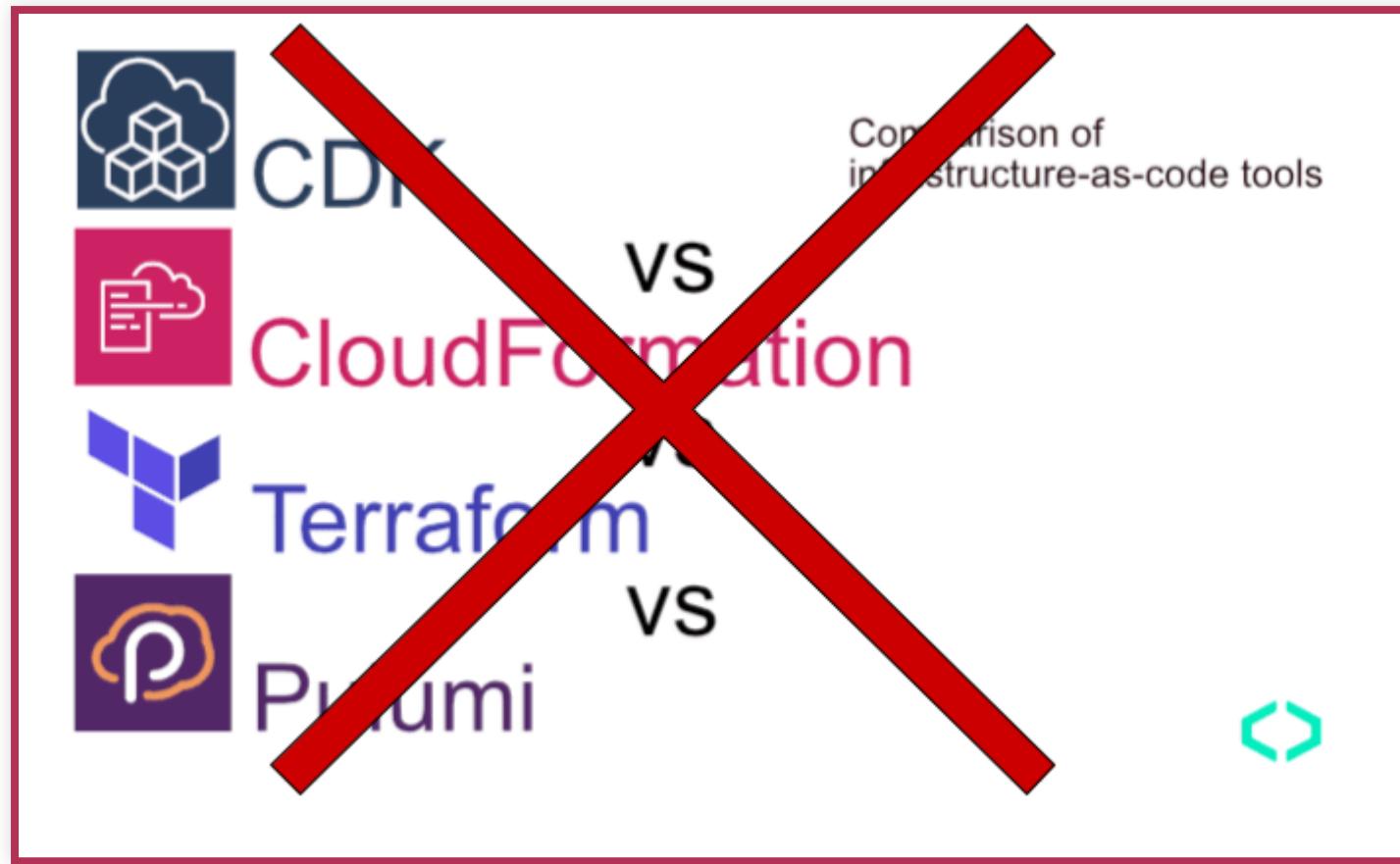
OUR ABSTRACTION

```
spec:  
  releaseName: telemetry-kafka  
  interval: 5m  
  chart:  
    spec:  
      chart: telemetry-kafka  
      version: 1.0.0  
      sourceRef:  
        kind: HelmRepository  
        name: fastly-charts  
        namespace: platform-flux  
values:  
  kafka:  
    replicas: 5  
    volumesSize: 50Gi  
    resources:  
      requests:  
        memory: 10Gi  
        cpu: 500m  
      limits:  
        memory: 16Gi  
        cpu: '4'  
  zookeeper:  
    replicas: 5  
    volumesSize: 20Gi  
  autopilotSites:  
  - PoP-ID1  
  - PoP-ID2  
  - PoP-ID3  
  - PoP-ID4
```

Kafka cluster deployment and configuration is integrated

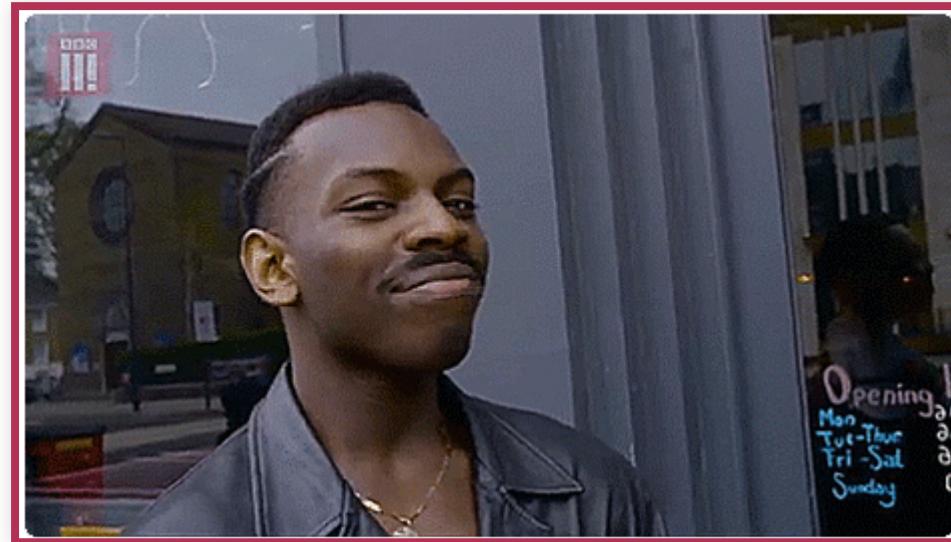


So no other IaC solutions or tooling are required



Not having a problem is better than the best of the solutions

We could even infer Kafka cluster sizing from the # of PoPs we are enabling



PORTABLE

```
$ cp -a prd-usc1/autopilot prd-awsuse1/
$ ls prd-awsuse1/autopilot/
controllerd.yaml  telemetry-api.yaml  telemetry-sfacctd.yaml
flow-proxy.yaml   telemetry-kafka.yaml
$ git commit -am "deploy all the things!" && git push
```

Moving services across clusters with a relatively low effort

```
$ ls -d */autopilot | cat
dev-awsuse2/autopilot
prd-assem1/autopilot
prd-awsapsem1/autopilot
prd-awseuw3/autopilot
prd-awsuse2/autopilot
prd-euw2/autopilot
prd-usc1/autopilot
stg-awsuse2/autopilot
```

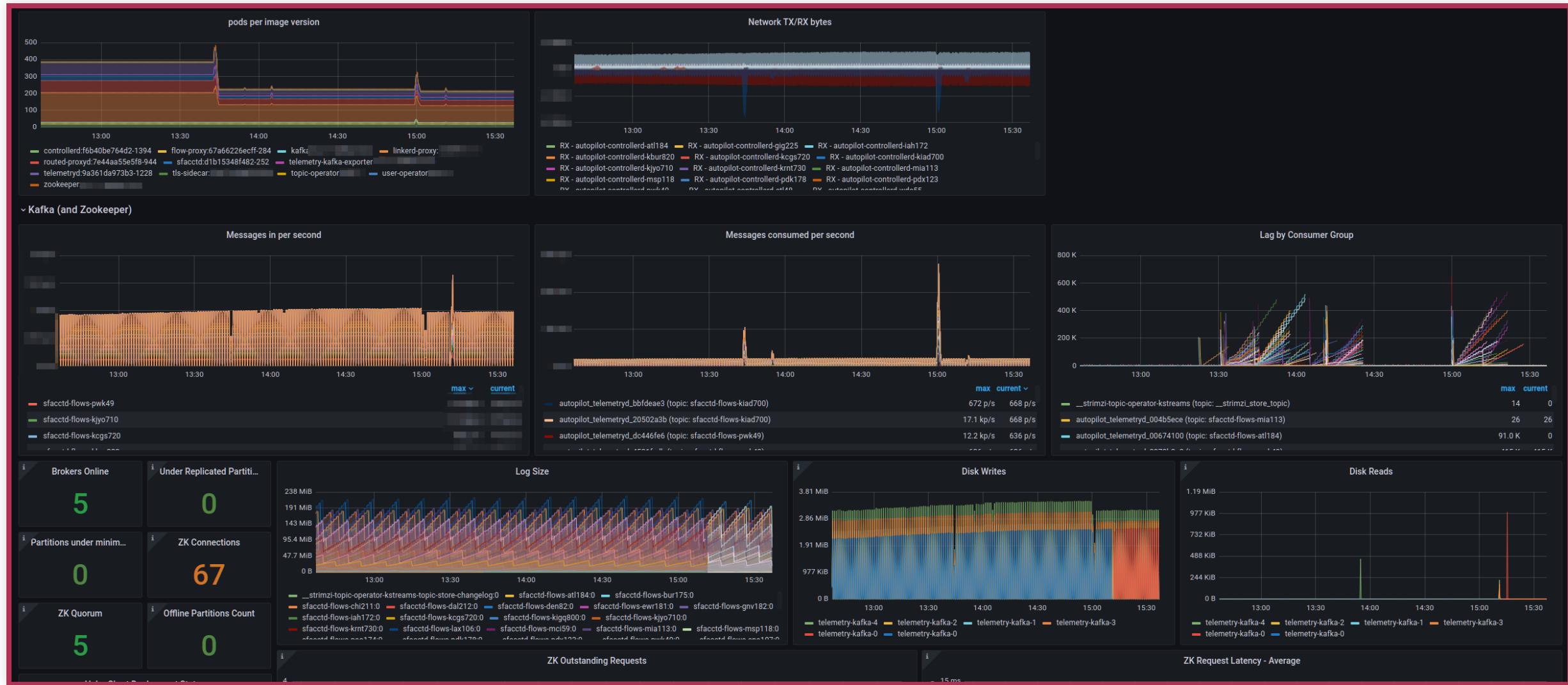
PLUGABLE

SAME TOOLS FOR DAY 2 OPERATIONS

```
$ kubectl get kafkatopics
NAME          CLUSTER    PARTITIONS  REPLICATION FACTOR  READY
consumer-offsets---84e7a678d08f4bd226872e5cd  telemetry   50          3              True
sfacctd-flows-PoP-ID1                         telemetry   1           3              True
sfacctd-flows-PoP-ID2                         telemetry   1           3              True
sfacctd-flows-PoP-ID3                         telemetry   1           3              True
sfacctd-flows-PoP-ID4                         telemetry   1           3              True
```

```
# Other Kafka operations also mapped to K8s primitives!
$ kubectl annotate statefulset cluster-name-kafka strimzi.io/manual-rolling-update=true
$ kubectl annotate statefulset cluster-name-zookeeper strimzi.io/manual-rolling-update=true
```

CONSOLIDATED DASHBOARDS



EASY EVENT CORRELATION

splunk>enterprise Apps ▾

dcaballero@fastly.com ▾ Messages ▾ Settings ▾ Activity ▾

Search Analytics Datasets Reports Alerts Dashboards

New Search

index="elevation_dev_usc1" ("kubernetes.namespace_name"=platform-operators "kubernetes.labels.app.kubernetes.io/instance"="kafka-operator") OR ("kubernetes.labels.app.kubernetes.io/name"="kafka") OR ("kubernetes.labels.app.kubernetes.io/component"="autopilot-telemetry")

✓ 3,503 events (5/6/22 2:51:00.000 PM to 5/6/22 3:51:41.000 PM) No Event Sampling ▾

Events (3,503) Patterns Statistics Visualization

Format Timeline ▾ — Zoom Out + Zoom to Selection × Deselect

List ▾ Format 50 Per Page ▾

< Prev 1 2 3 4 5 6

Time	Event
5/6/22 3:51:38.245 PM	{ [-] cluster: elevation-dev-usc1

kubernetes.container_name

4 Values, 100% of events

Selected Yes No

Reports

Top values Top values by time Rare values

Events with this field

Values	Count	%
telemetryd	3,206	91.522%
linkerd-proxy	143	4.082%
strimzi-cluster-operator	90	2.569%
kafka	64	1.827%

TTT call with code 200 OK" http.host=... http.proto_major=1 http.request.length_bytes=0 http.request.method=GET http.re...
ytes=2 http.response.status=200 http.time_ms=0.008 http.url.path=/health peer.address=... peer.port=32938 span.kind=server sys...
ernetes.var.log.containers.autopilot-telemetryd-rhv1000-deployment-... sourcetype = httpevent

◀ Hide Fields : All Fields

SELECTED FIELDS
a host 3
a source 6
a sourcetype 1

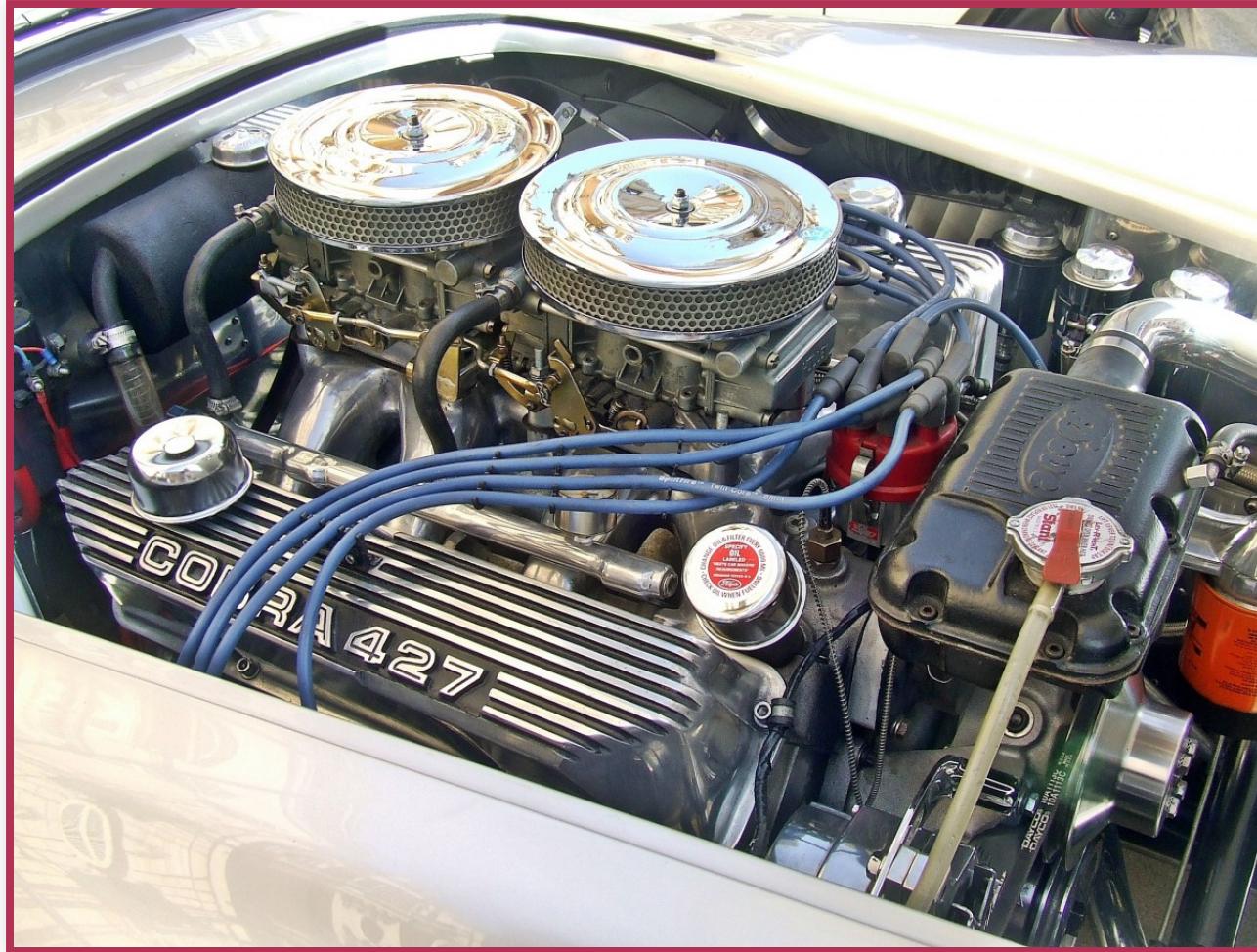
INTERESTING FIELDS
a cluster 1
a environment 1
a index 1
a kubernetes.container_name 4
a kubernetes.host 3
a kubernetes.labels.app.kubernetes.io/component 2
a kubernetes.labels.app.kubernetes.io/name 3
a kubernetes.labels.linkerd.io/control-plane-ns 1
a kubernetes.labels.linkerd.io/proxy-deployment 1

So Kafka, and its management, is not special

THE BAD

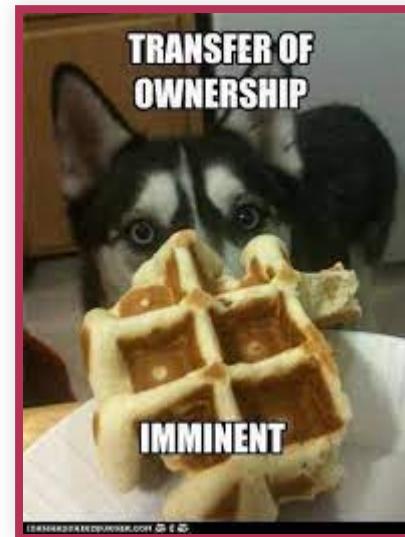


Keeping the operator up-to-date is an ongoing maintenance cost



KAFKA OPERATOR VS KAFKA CLUSTER OWNERSHIP

- The current ownership model requires strong collaboration between the team owning the operator and the team owning the Kafka cluster
- This might not scale well with many teams adopting this pattern



THE UGLY



Multi-cloud, service mesh, UDP, BGP...

Multi-cloud, service mesh, UDP, BGP...



UDP FLOW: STALE CONNTRACK ENTRY

- Autopilot Telemetry API receives a constant UDP flow from Fastly's switches
- We noticed that pods restarted and scheduled on the same node, will provoke UDP packets to be blackholed
- My teammate Danny Kulchinsky discovered a **bug** in Kube-Proxy:
 - Not flushing the relevant conntrack entry belonging to the LB ingress IP, when 0 -> 1+ endpoints
 - Flow will hit conntrack before IPTABLES nat gets applied
 - Conntrack entry will have the LB external IP not the new pod IP

AWS LB SG RULES HARD LIMIT

- AWS LB controller, creates excessive amount of NLB SG rules
 - 1 inbound rule to node's SG for client traffic per allowed client source IP
 - 1 rule for each LB subnet in the VPC for healthchecks
- ALBs, ELBs (deprecated) not an option: TCP/UDP, preserve src IP

AWS LB SG RULES HARD LIMIT (SOLUTION)

- Disable SG rules creation
- Block/Allow traffic based on Calico GNPs, dynamically generated with Kyverno
- Cleanup GNPs with custom Service Controller

IMPACT



Elevation and the Strimzi Operator had a very positive impact in
Autopilot's Telemetry Pipeline development

- Even with a small team with other responsibilities...
- First version of the pipeline was developed and deployed in 4 weeks

 **Add initial gopherup scaffolding ✓**

#1 by lorenzosaino was merged on Apr 25, 2021 • Approved

 **NCO-3473 - telemetryd deployment to elevation dev ✓**

#30 by floatingstatic was merged on May 26, 2021 • Review required

No significant changes were required to scale from 1 -> 10 -> ~ 100 sites

The screenshot shows a GitHub search results page with a red border. The search query in the header is "is:pr Automated Elevation Data Update: kafka". Below the header, there is a button to "Clear current search query, filters, and sorts". The search results list eight pull requests:

- Automated Elevation Data Update: prd-usc1-telemetry-kafka-1.0.1 ✓**
#6132 by fastly-elevation was merged 18 hours ago • Approved
- Automated Elevation Data Update: prd-usc1-telemetry-kafka-0.1.10 ✗**
#4979 opened on Mar 9 by fastly-elevation • Review required
- Automated Elevation Data Update: prd-usc1-telemetry-kafka-0.1.9 ✗**
#4975 by fastly-elevation was closed on Mar 9 • Review required
- Automated Elevation Data Update: prd-usc1-telemetry-kafka-0.1.8 ✓**
#4010 by fastly-elevation was merged on Jan 4 • Approved
- Automated Elevation Data Update: prd-usc1-telemetry-kafka-0.1.7 ✓**
#3805 by fastly-elevation was merged on Dec 14, 2021 • Review required
- Automated Elevation Data Update: prd-usc1-telemetry-sfacctd-0.0.14-5ecc59427cd3-111 ✓**
#3535 by fastly-elevation was merged on Nov 30, 2021 • Approved
- Automated Elevation Data Update: prd-usc1-telemetry-sfacctd-0.0.12-3dd68b2e08b9-93 ✓**
#3478 by fastly-elevation was merged on Nov 23, 2021 • Approved
- Automated Elevation Data Update: prd-usc1-telemetry-kafka-0.1.6 ✓**
#2792 by fastly-elevation was merged on Oct 13, 2021 • Review required

At the top right of the search results, there are buttons for "Author", "Label", and "Projects".

Everyone manages kafka!

master ▾ [elevation-data](#) / [workloads](#) / [prd-usc1](#) / [autopilot](#) / [telemetry-kafka.yaml](#)

 **floatingstatic** NCO-4384 - Decommission bur175, bwi50, chi211, jax209 autopilot compo... [...](#) 

 10 contributors 

 69 lines (69 sloc) | 1.09 KB

```
1 apiVersion: helm.toolkit.fluxcd.io/v2beta1
2 kind: HelmRelease
3 metadata:
4   name: telemetry-kafka-prd-usc1
```



And do you remember this?

... and Autopilot is not the only use case:

- Our NetOps team also gets PoPs traffic insights
- Capacity planning
- New DDoS solutions
- Blog posts :)
- Others we may not even anticipate yet

It is becoming real

CLOSURE



SUMMARY

SUMMARY

Fastly needs an scalable and multi-* platform to run network automation

SUMMARY

Fastly needs an scalable and multi-* platform to run network automation

K8s helps Fastly to run some control plane services

SUMMARY

Fastly needs an scalable and multi-* platform to run network automation

K8s helps Fastly to run some control plane services

K8s + (kafka-)operators + elevation setup was a success for the autopilot-telemetry use case

QUESTIONS?



Thank You!

fastly®

The logo for Fastly, featuring the word "fastly" in a bold, red, sans-serif font. The letter "a" has a small, white clock icon integrated into its center. A registered trademark symbol (®) is positioned at the top right of the letter "y".