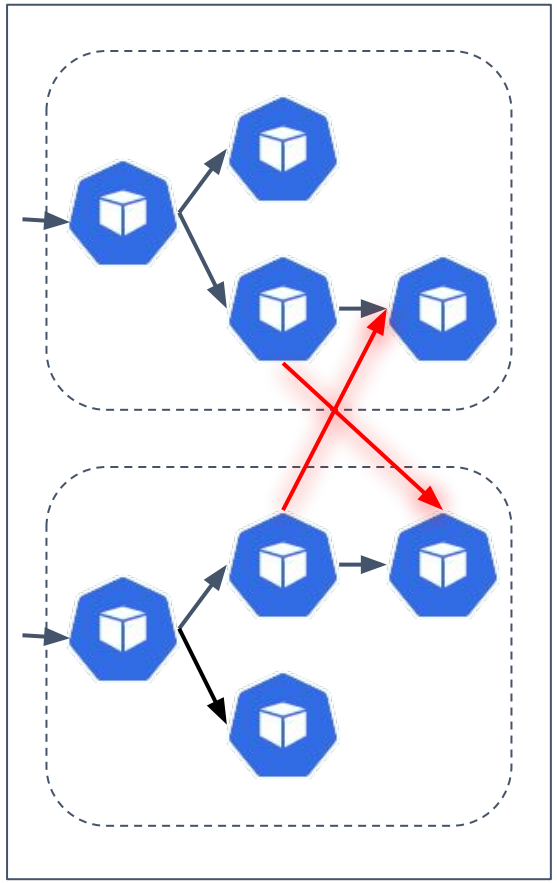
Monitoring Kubernetes with eBPF and Prometheus

KubeCon North America 2018

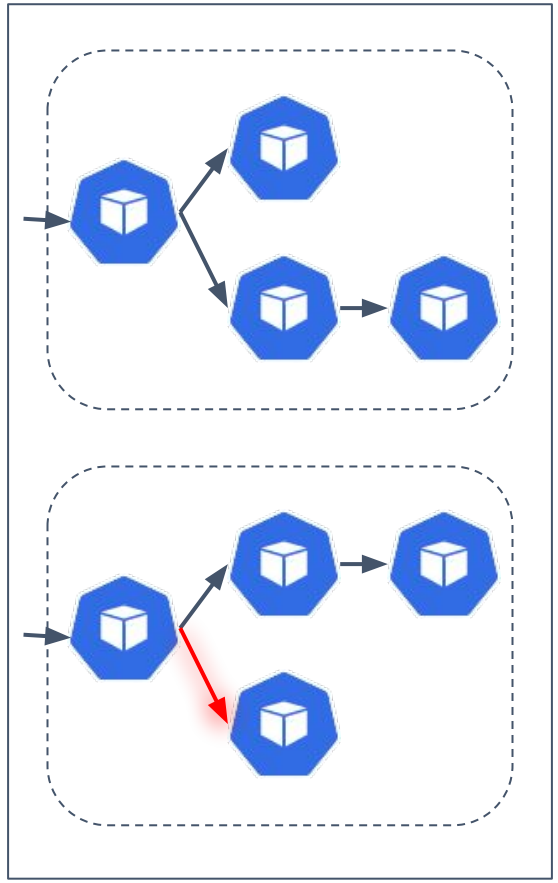
Agenda

- Flow monitoring: benefits
- Getting flow data
- Technology: eBPF
- Tour of our staging cluster
- Productizing: Challenges

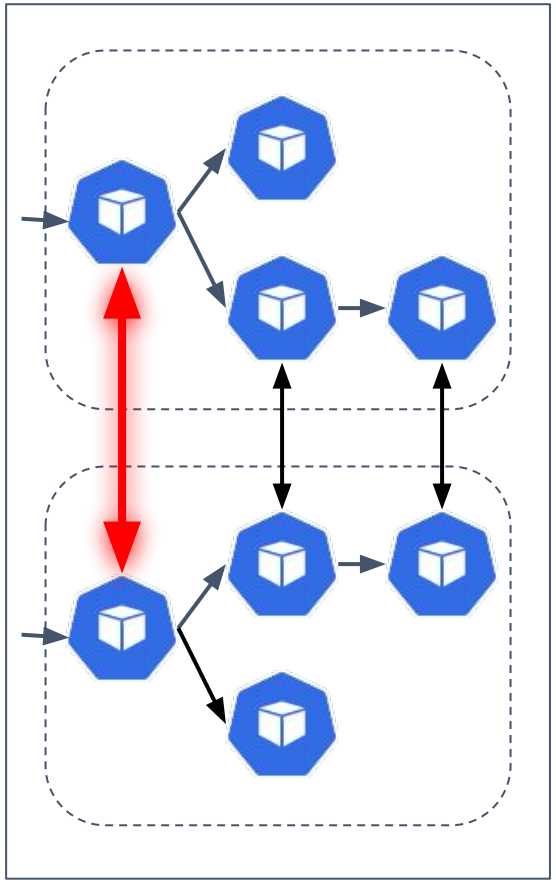
Flow Monitoring: benefits



Architecture, HA,
Env isolation

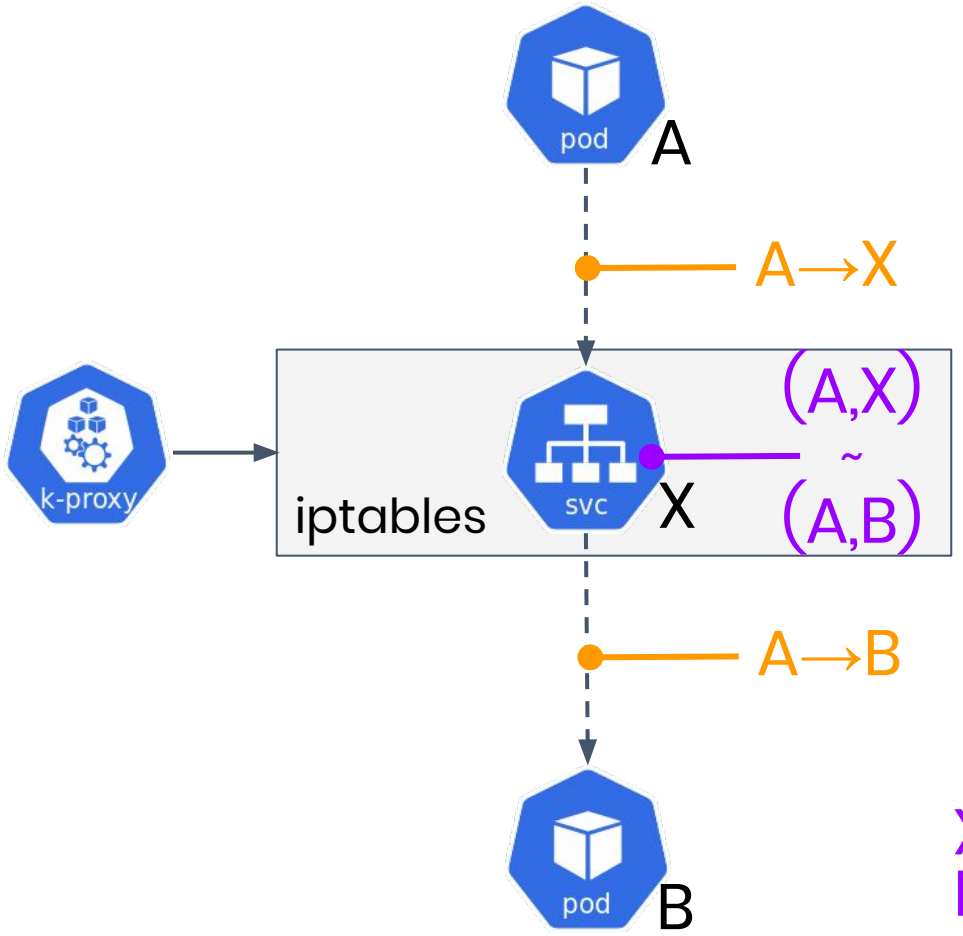


Health



Cost

Getting Flow Data




```
$ kubectl describe pod $POD
Name:           A
Namespace:      staging
...
Status:         Running
IP:             100.101.198.137
Controlled By:  ReplicaSet/A
```

```
# PID=`docker inspect -f '{{.State.Pid}}' $CONTAINER` \
   nsenter -t $PID -n ss -ti

ESTAB  0   0    100.101.198.137:34940   100.65.61.118:8000
     cubic wscale:9,9 rto:204 rtt:0.003/0 mss:1448 cwnd:19
ssthresh:19 bytes_acked:2525112 segs_out:15664
segs_in:15578 data_segs_out:15662 send 73365.3Mbps
lastsnd:384 lastrcv:10265960 lastack:384 rcv_space:29200
minrtt:0.002
```

A X

```
# conntrack -L

tcp      6 86399 ESTABLISHED src=100.101.198.137 A
dst=100.65.61.118 sport=34940 dport=8000
src=100.101.198.147 dst=100.101.198.137 sport=8000
dport=34940 [ASSURED] mark=0 use=1
```

X B A

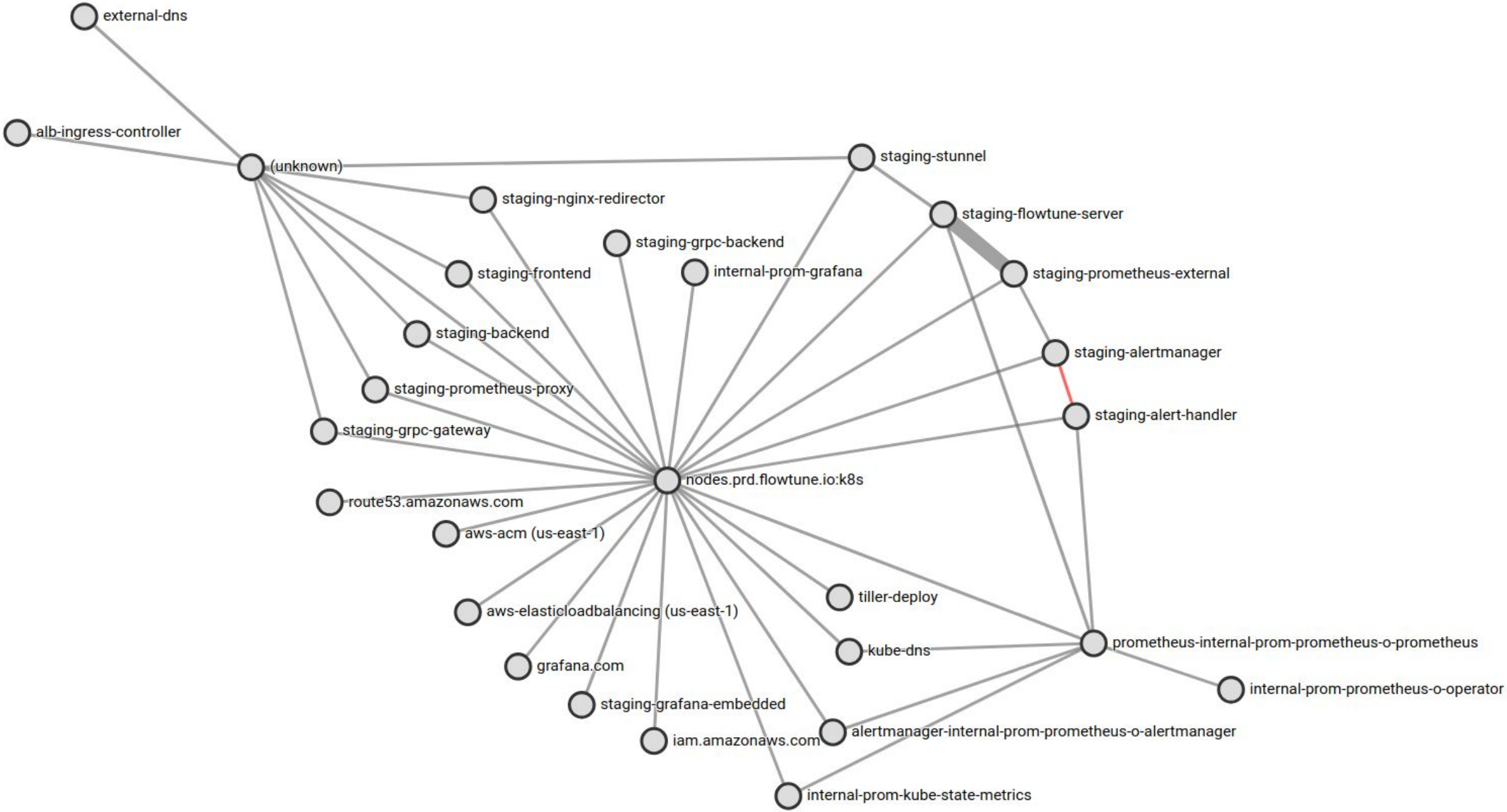
Technology: eBPF

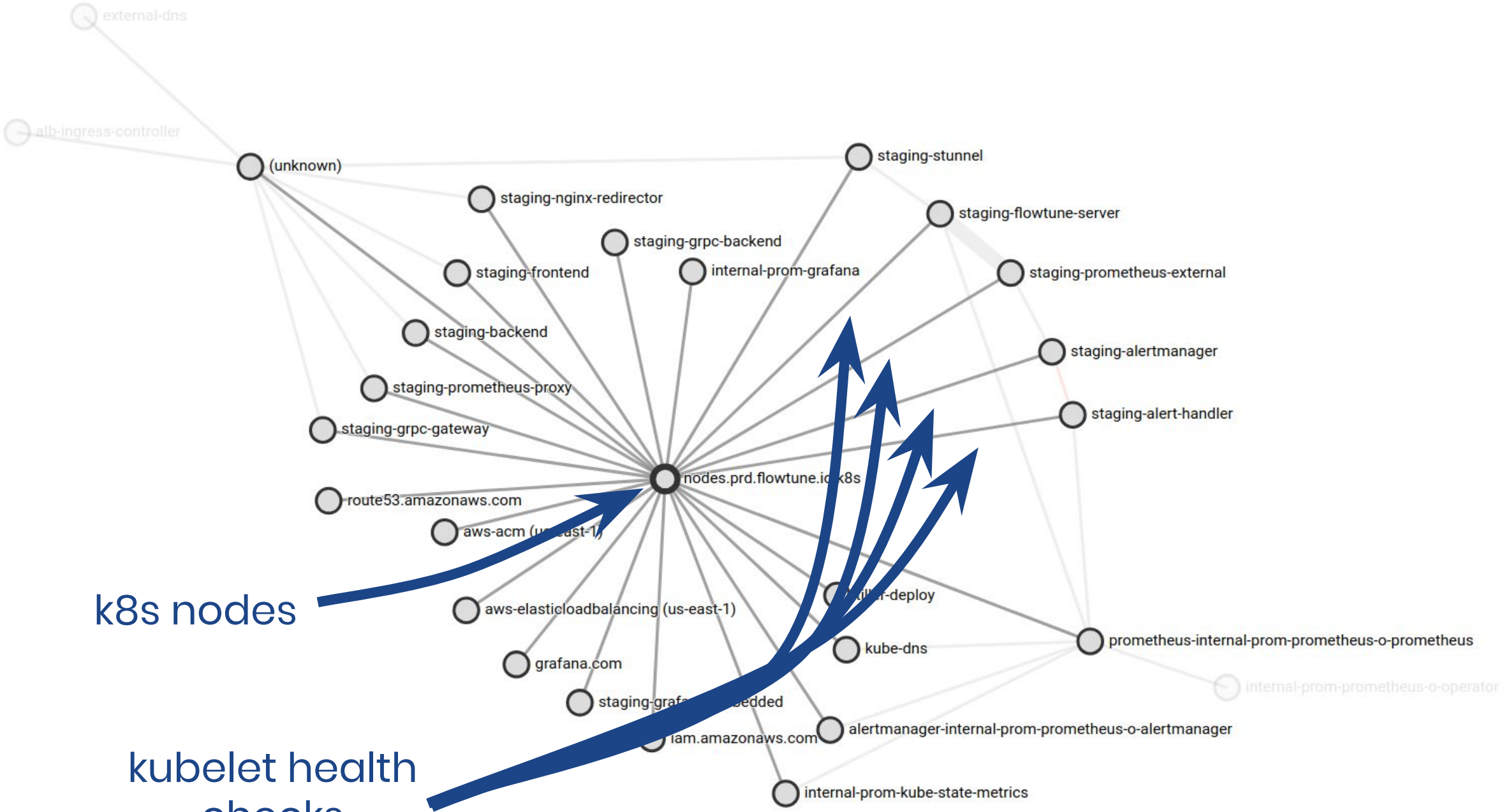
- Linux bpf() system call since 3.18
- Run code on kernel events
- Only changes, more data

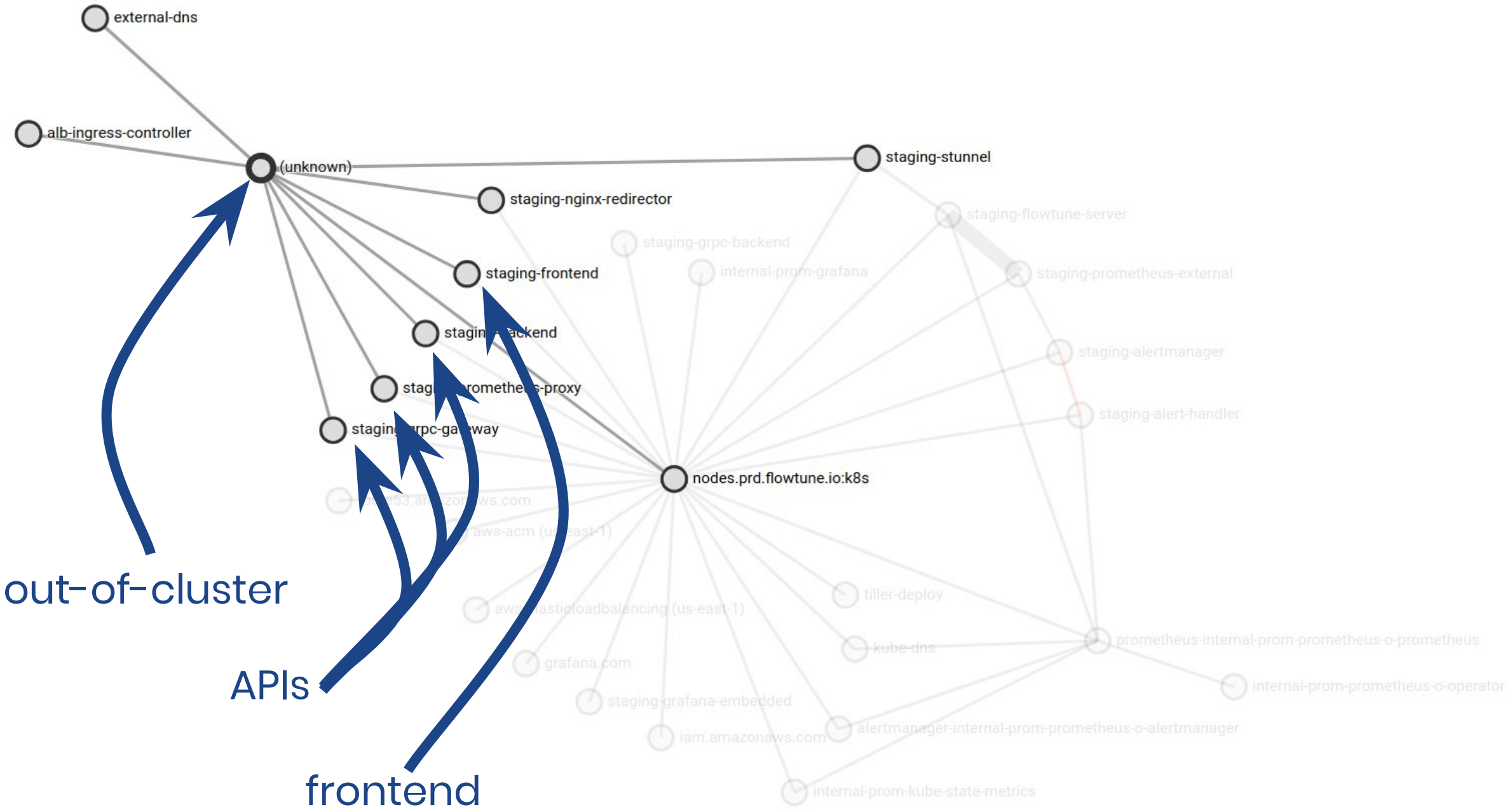
- Safe: In-kernel verifier, read-only
- Fast: JIT-compiled

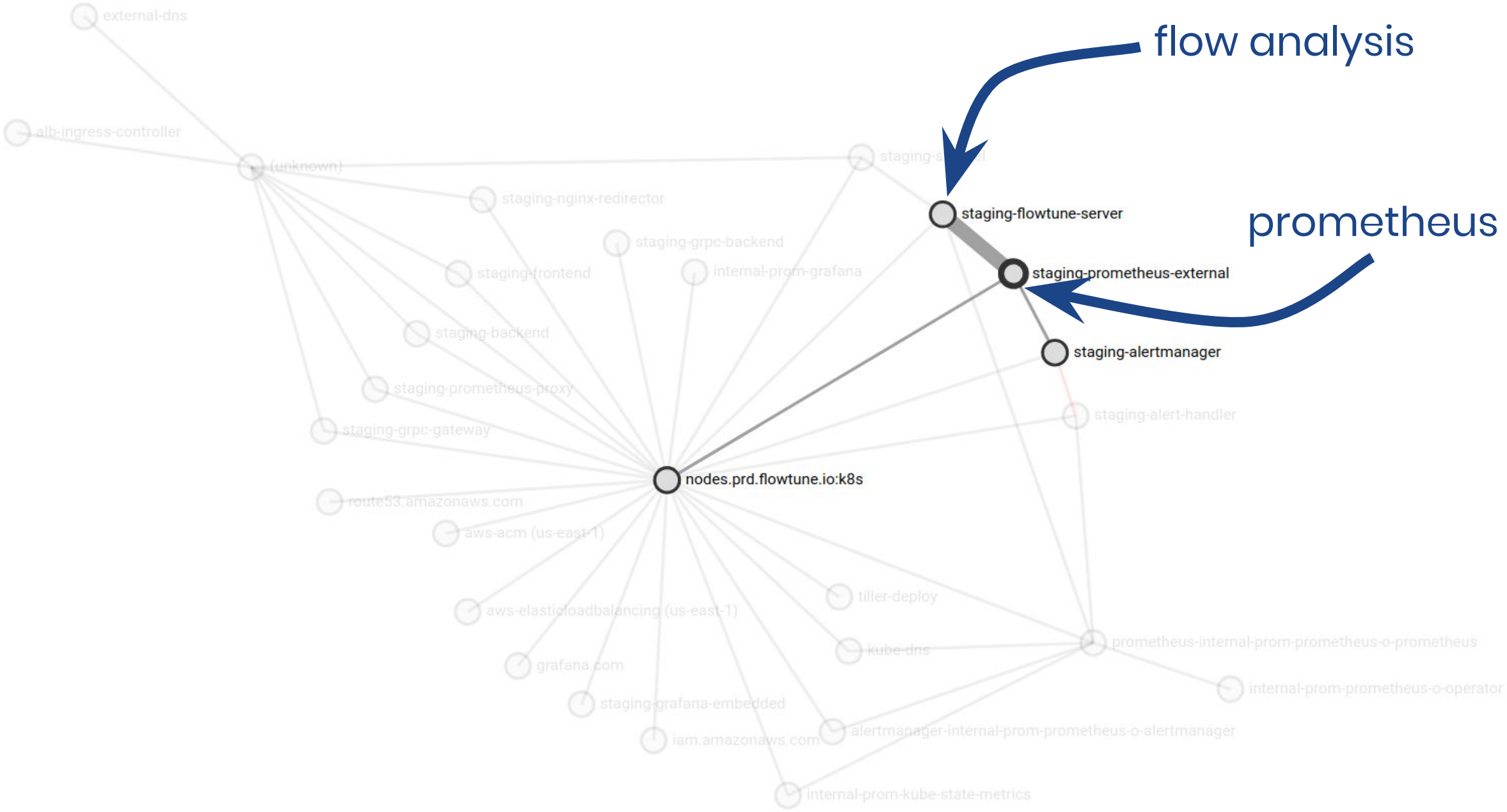
Unofficial BPF mascot by Deirdré Straughan

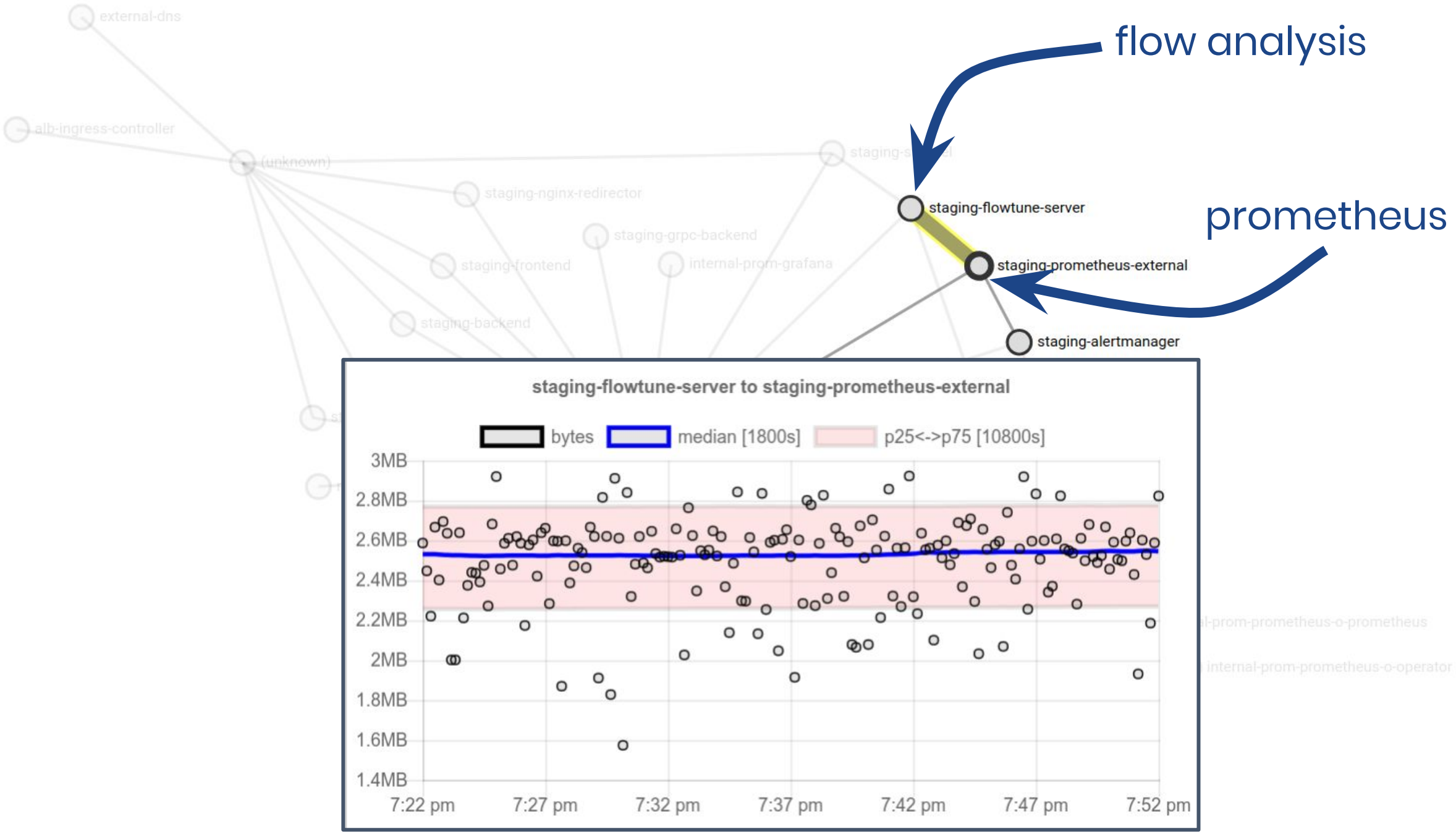
→ **100% coverage + no app changes + low overhead ftw!**


external-dns
alb-ingress-controller
(unknown)
staging-stunnel
staging-nginx-redirector
staging-flowtune-server
staging-grpc-backend
staging-frontend
internal-prom-grafana
staging-prometheus-external
staging-backend
staging-alertmanager
staging-prometheus-proxy
staging-alert-handler
staging-grpc-gateway
nodes.prd.flowtune.io:k8s
route53.amazonaws.com
aws-acm (us-east-1)
tiller-deploy
aws-elasticloadbalancing (us-east-1)
kube-dns
prometheus-internal-prom-prometheus-o-prometheus
grafana.com
internal-prom-prometheus-o-operator
staging-grafana-embedded
alertmanager-internal-prom-prometheus-o-alertmanager
iam.amazonaws.com
internal-prom-kube-state-metrics


external-dns
alb-ingress-controller
(unknown)
staging-stunnel
staging-nginx-redirector
staging-flowtune-server
staging-grpc-backend
staging-frontend
internal-prom-grafana
staging-prometheus-external
staging-backend
staging-alertmanager
staging-prometheus-proxy
staging-alert-handler
staging-grpc-gateway
nodes.prd.flowtune.io k8s
route53.amazonaws.com
aws-acm (us-east-1)
k8s nodes
aws-elasticloadbalancing (us-east-1)
-deploy
prometheus-internal-prom-prometheus-o-prometheus
kube-dns
grafana.com
internal-prom-prometheus-o-operator
staging-graf bedded
alertmanager-internal-prom-prometheus-o-alertmanager
iam.amazonaws.com
kubelet health checks
internal-prom-kube-state-metrics


external-dns
alb-ingress-controller
(unknown)
staging-stunnel
staging-nginx-redirector
staging-frontend
nodes.prd.flowtune.io:k8s
out-of-cluster
APIs
frontend


flow analysis
prometheus
external-dns
alb-ingress-controller
(unknown)
staging-nginx-redirector
staging-grpc-backend
staging-frontend
internal-prom-grafana
staging-backend
staging-prometheus-proxy
staging-grpc-gateway
staging-flowtune-server
staging-prometheus-external
staging-alertmanager
staging-alert-handler
nodes.prd.flowtune.io:k8s
route53.amazonaws.com
aws-acm (us-east-1)
aws-elasticloadbalancing (us-east-1)
tiller-deploy
kube-dns
prometheus-internal-prom-prometheus-o-prometheus
grafana.com
internal-prom-prometheus-o-operator
staging-grafana-embedded
iam.amazonaws.com
alertmanager-internal-prom-prometheus-o-alertmanager
internal-prom-kube-state-metrics


flow analysis
prometheus
staging-flowtune-server
staging-prometheus-external
staging-alertmanager
staging-flowtune-server to staging-prometheus-external
bytes
median [1800s]
p25<->p75 [10800s]
3MB
2.8MB
2.6MB
2.4MB
2.2MB
2MB
1.8MB
1.6MB
1.4MB
7:22 pm
7:27 pm
7:32 pm
7:37 pm
7:42 pm
7:47 pm
7:52 pm

Productizing: Challenges

- **CPU overhead**: profile + iterate → 0.1% CPU
- **Network overhead**: encode efficiently and compress
- **Security**: TLS, OAuth everywhere
- **Real-time:** stream, don't batch → 2 second latency
- Pre-aggregate to manage **cardinality**
- Workload **baselining** for **automatic alerting**

Jonathan Perry <jperry@flowmill.com>
www.flowmill.com

FLOWMILL