kentik®

# Open Edge Network Telemetry

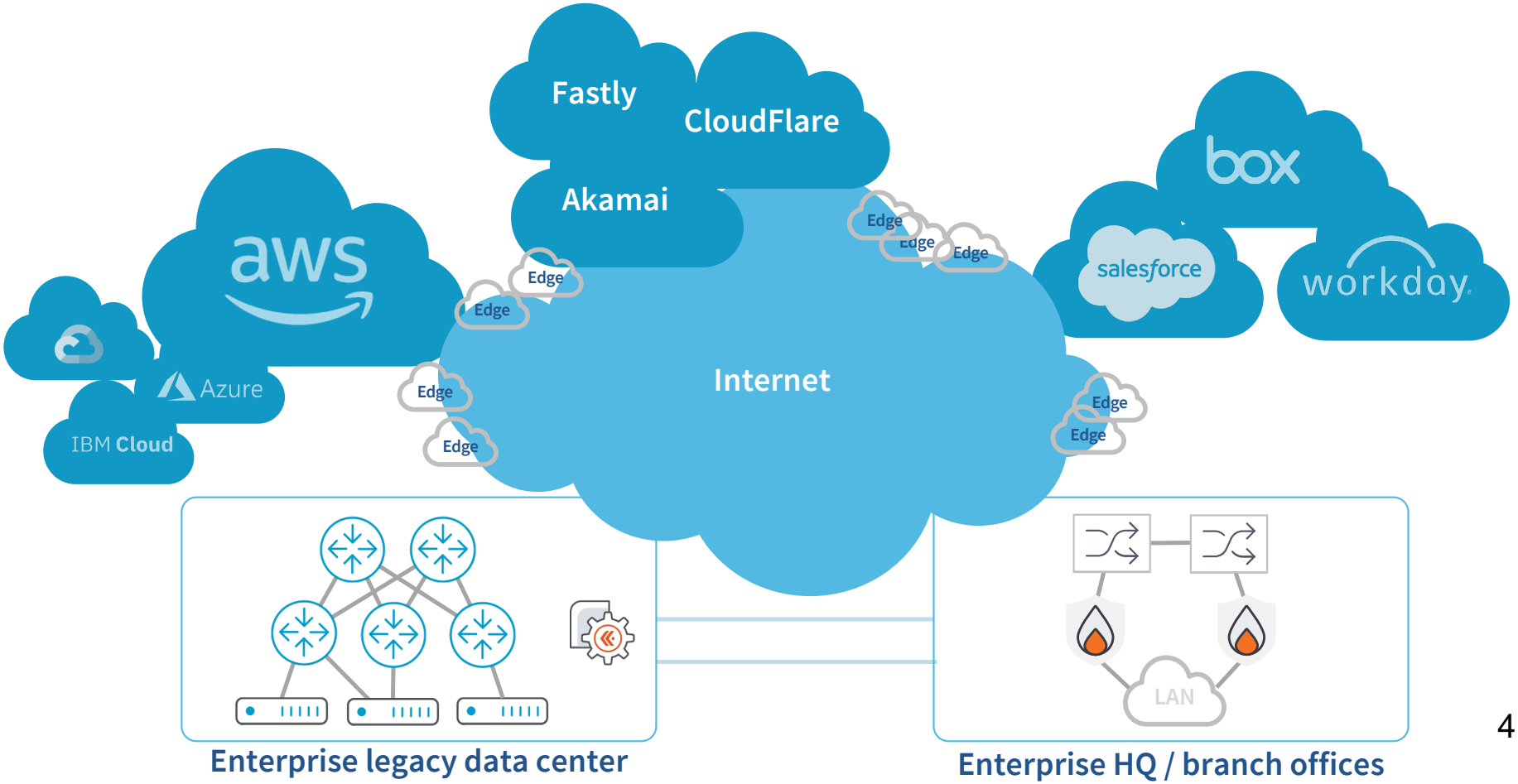
Avi Freedman

# Agenda

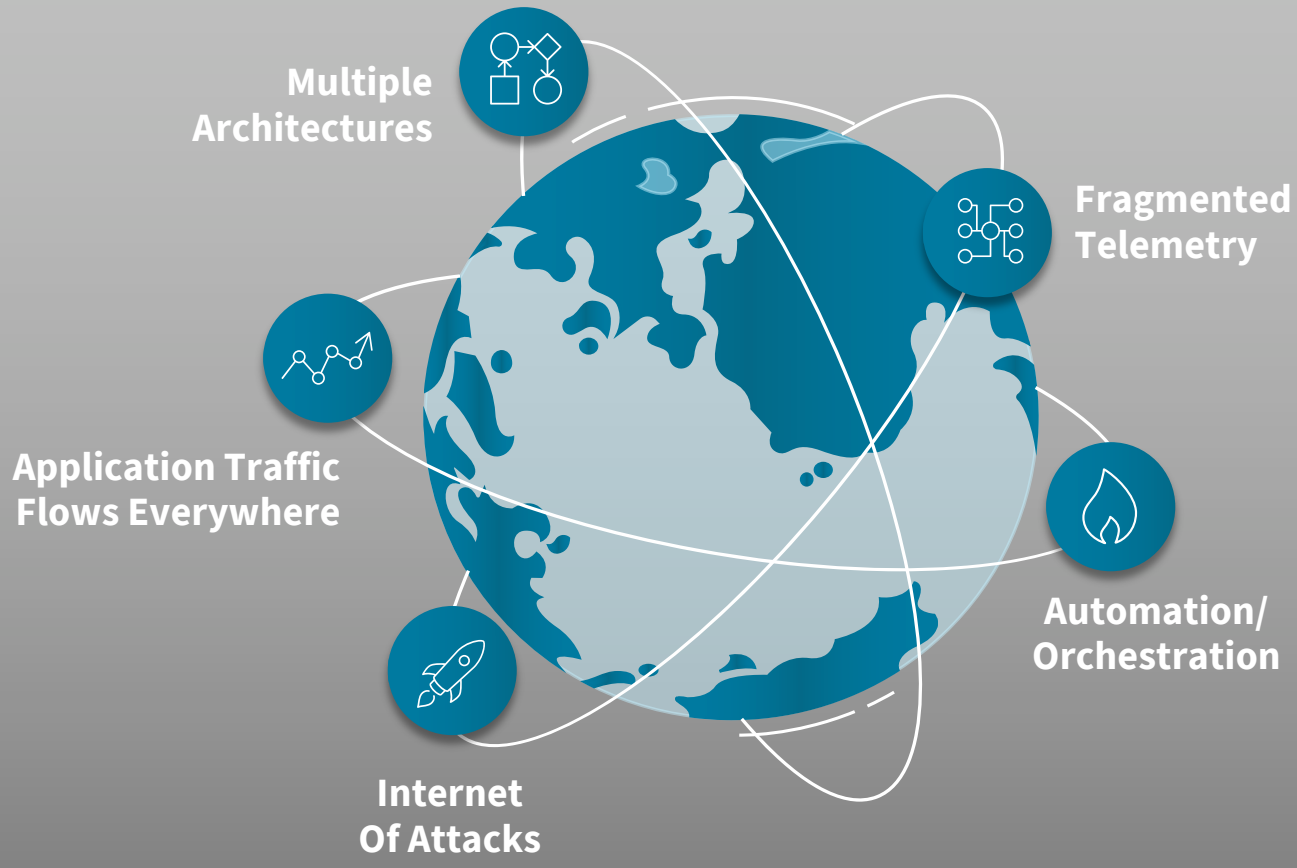
- The Problem
- Open Edge Network Telemetry
- Kentik Labs
- Projects: Agents
- Problem: Telemetry Pipelines
- Projects: ktranslate
- Roadmap
- Questions

# The Problem

# It’s a crazy hybrid world

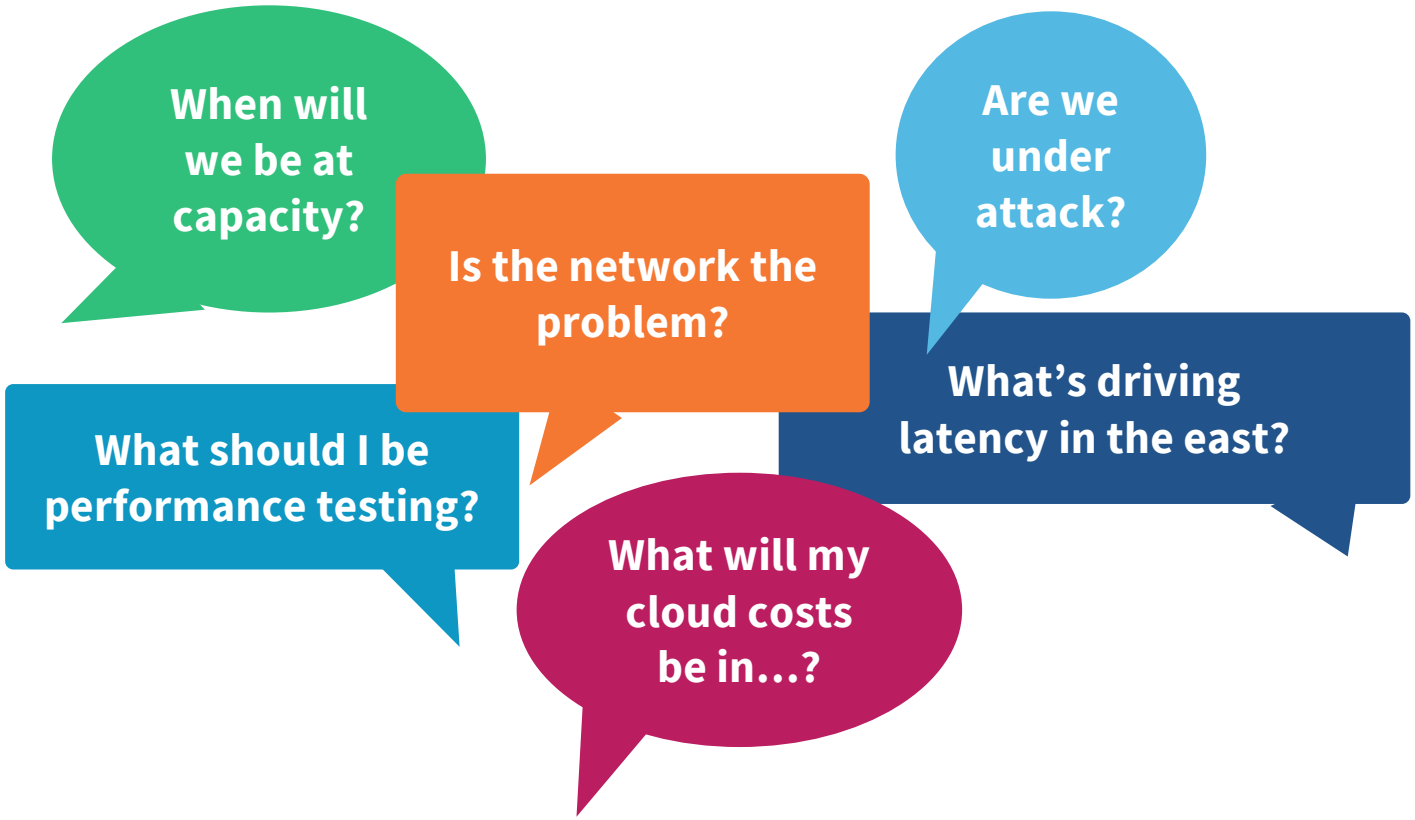


# Networking in 2022



# NETWORK **OBSERVABILITY**

The ability to answer *any* question about *all* your networks



# Getting to Complete Network Observability

- Getting to complete observability requires visibility across all networks
- Which adds huge challenges:
  - Getting visibility into other people's infra (cloud infra/lambda)
  - Deploying agents (eBPF, PCAP, XDP) on compute
  - Telemetry from cloud-native CNI layers
- That require different approaches -not just flow/SNMP any more!

# Kentik Labs Background

# Kentik Labs

- Kentik is a 7-year old network observability company
- Kentik Labs focuses on open source projects for the wider community
- With integrations to data platforms, and orchestration and observability systems

- With first projects:
    - Telemetry bus and transformation (**ktranslate**)
    - Open agents **kappa, convis,** and **ktranslate** (eBPF, PCAP, flow/SNMP)
    - And also (not covering in this presentation):
        - K8s performance (**Odyssey**)
        - Cloud Metadata (**Cloud Meta**)
        - Flow visibility in Prometheus and Grafana (**Kentik Lite**)
        - Scalable low-level ICMP, UDP, and TCP diagnostics (**NetDiag**)



- And release other fun open projects and integrations

# The Vision for ktranslate and agents

- An open edge network telemetry suite
- That can work with any network type and architecture
- And any data platform, both open and commercial
- And gives benefits to DevOps, Security, and other groups

Agents: kprobe, convis, and kappa

# kprobe (PCAP)

- What
  - PCAP-focused traffic monitoring
  - Samples via kernel primitives to do 10G+ monitoring with low CPU
  - Performance instrumentation (at lower speeds) via TCP seq tracking
  - Decodes for DNS, HTTP, and TLS (transport info, not decodes)
  - Written in golang
- When
  - Already launched, same code base in use for years commercially
- How
  - Binary / daemon
  - docker run (with prometheus and grafana for standalone functionality)

# convis and kappa (eBPF)

- convis (Rust)
  - convis is an eBPF agent that runs on servers/VMs/containers
  - It tracks connections+traffic
  - And performance (via tcp_info)
  - And correlates with pid, process name, namespace, k8s pod
- kappa (Rust)
  - Hybrid-mode agent that watches traffic via PCAP and enriches via k8s
  - Clusters in a pod so every flow has src and dst pid/container info
  - Will also monitor VMs, ip_forwarding, and other things not on-kernel
- When
  - Convis already launched
  - kappa in use commercially, OSSing it or next gen in 2022
- How
  - Binary / daemon
  - docker run (with prometheus and grafana for standalone functionality)

# Telemetry Bus: ktranslate

# Telemetry Bus: Motivation

- Early(ish) on, people used tools like samplicator for UDP-based protocols
- And/or still had platforms all polling or consuming separately from network elements
- More recently, people want to standardize on kafka but this doesn't alone solve:
    - Input and output adaptors
    - Transformation and filtering

# Telemetry Bus: Motivation

- Ourr goal is to build towards a distributed platform for taking all types of network telemetry and:
    - Replicating
    - Enriching
    - Filtering
    - Transforming (semantic and syntax)
    - In a scalable, operable way

# Telemetry Bus: ktranslate

- ktranslate (in golang) started taking network telemetry from Kentik with flexible output
- With active customer and partner-focused development in 2021, it can now:
- Take input in:
    - sFlow/NetFlow
    - SNMP
    - VPC Flow
    - Kentik's Cap'n Proto serialization format (kflow, and via that eBPF)
- Filter, re-sample, slice, dice, and roll up, doing:
    - Row (record) filtering
    - Column (tag/attribute) filtering
    - Rollups (to lower cardinality/TSDB)
    - Replication
- Output in/to:
  Prometheus, Influx, New Relic, Kentik, Splunk, Elastic,
  S3, Kafka, Dynatrace, and more

# ktranslate Usage Examples

docker pull kentik/ktranslate:v2

docker run -p 8082:8082 kentik/ktranslate:v2

######################

Output to Prometheus: docker run -p 8082:8082 kentik/ktranslate:v2 -format prometheus -sinks prometheus -prom_listen=:8084 -listen=0.0.0.0:8082

Output to InfluxDB: docker run -p 8082:8082 kentik/ktranslate:v2 -format influx -sinks http -http_url= http://localhost:8086/write?db=kentik -listen=0.0.0.0:8082

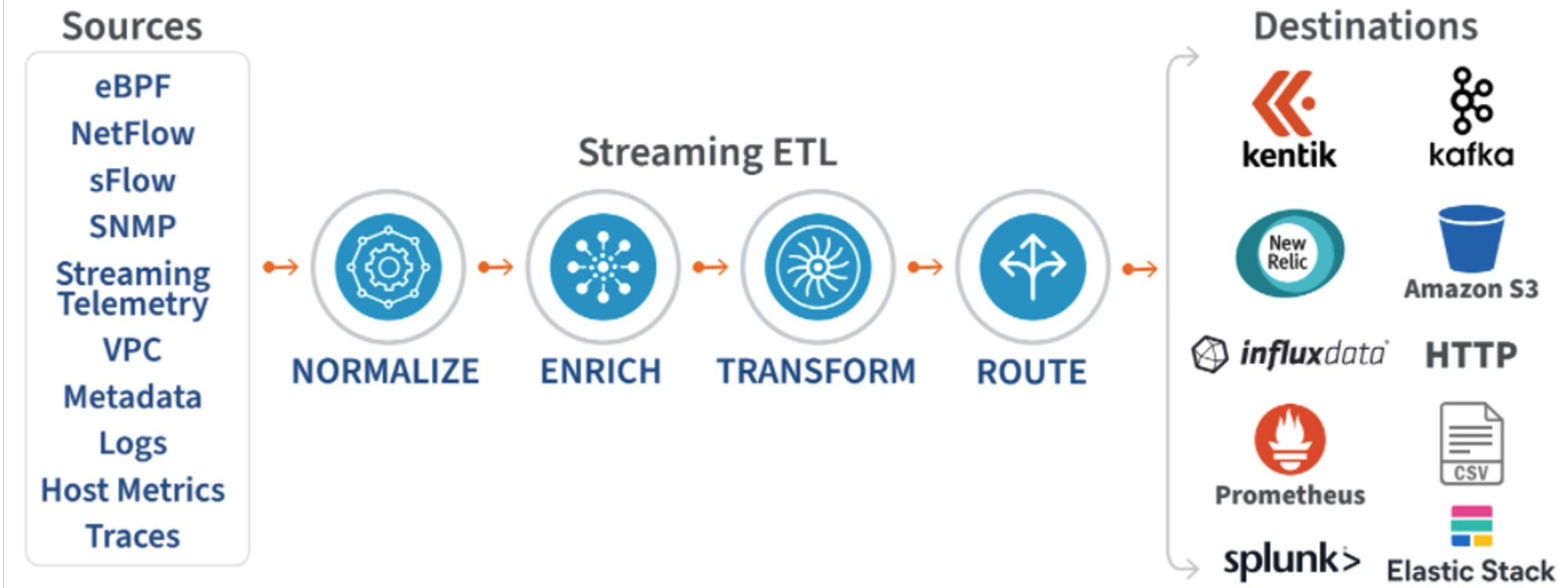
Output as NetFlow: docker run -p 8082:8082 kentik/ktranslate:v2 -format netflow -sinks net -net_server 127.0.0.1:9913 -max_flows_per_message 10 -listen=0.0.0.0:8082

######################

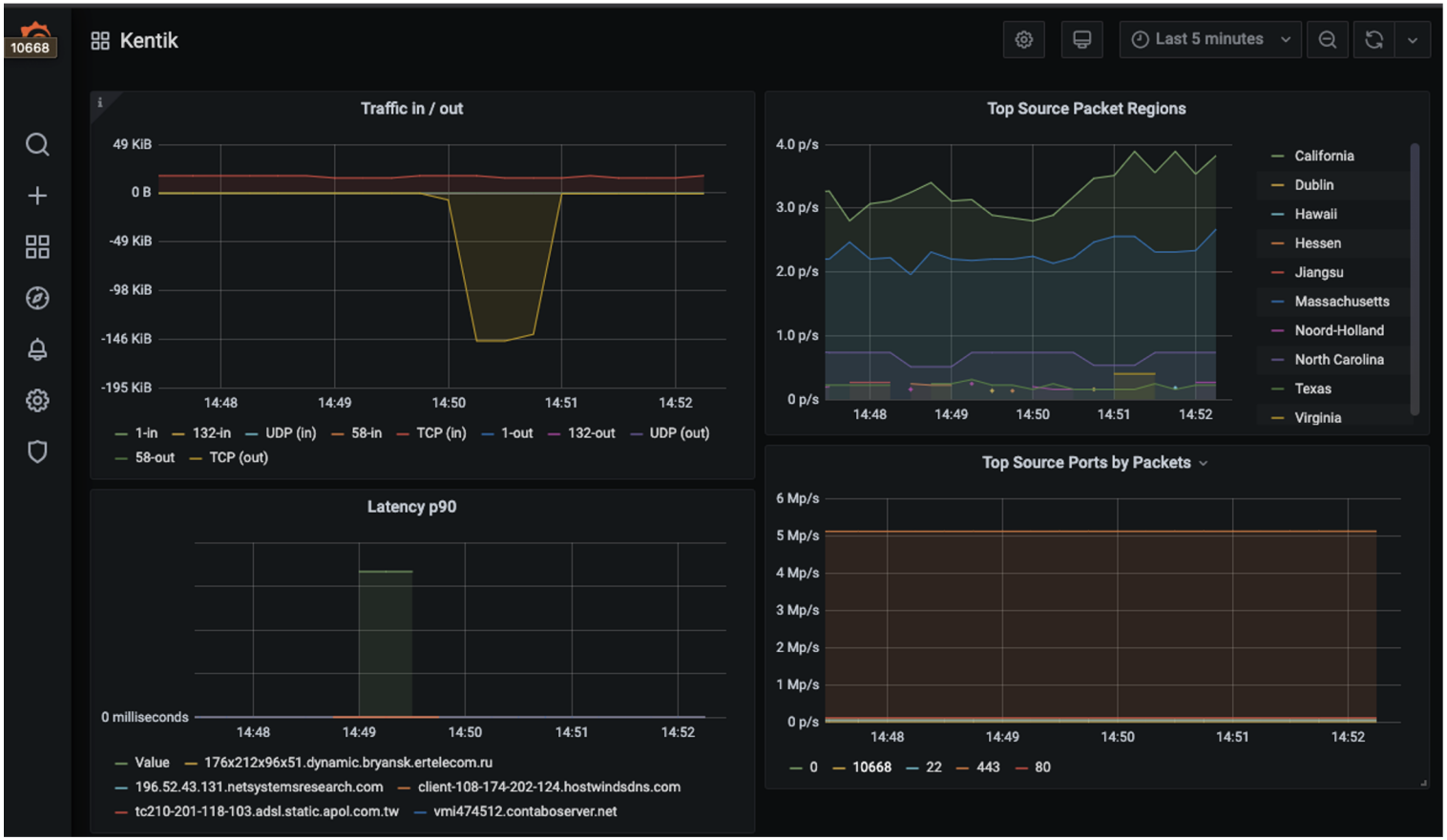
Filtered Output: docker run -p 8082:8082 kentik/ktranslate:v2 -format json -sinks stdout -filters int,l4_src_port,==,80 -listen=0.0.0.0:8082

Rolled-up Output: docker run -p 8082:8082 kentik/ktranslate:v2 -format json -sinks stdout -rollups unique,top_src_addr_by_count_dst_addr,dst_addr,src_addr -rollup_top_k 10 -rollup_interval 60 -rollups sum,in_bytes+out_bytes,l4_src_port -rollup_and_alpha -listen=0.0.0.0:8082

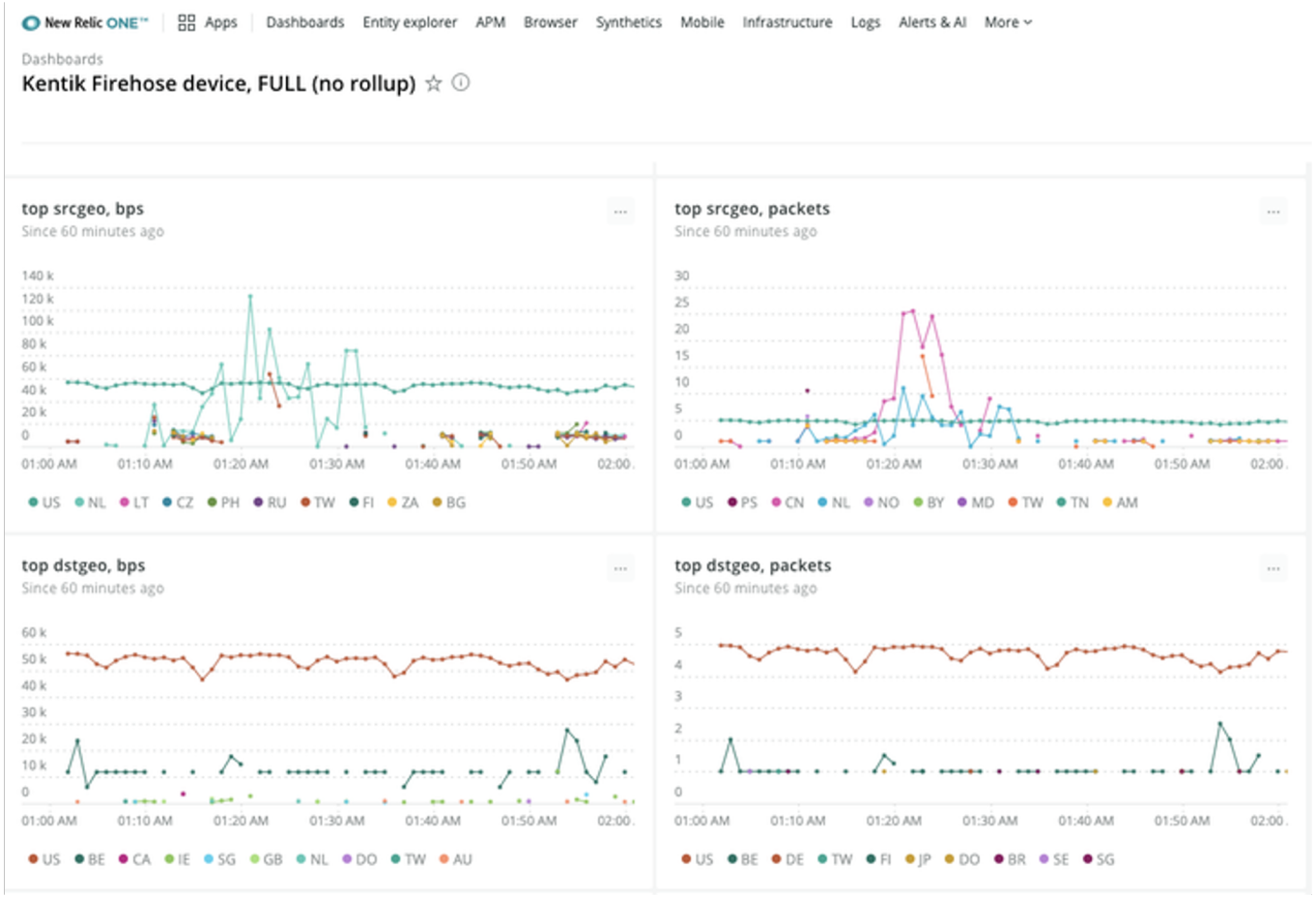
# ktranslate + Agents = Open Source (Network) Telemetry Pipeline



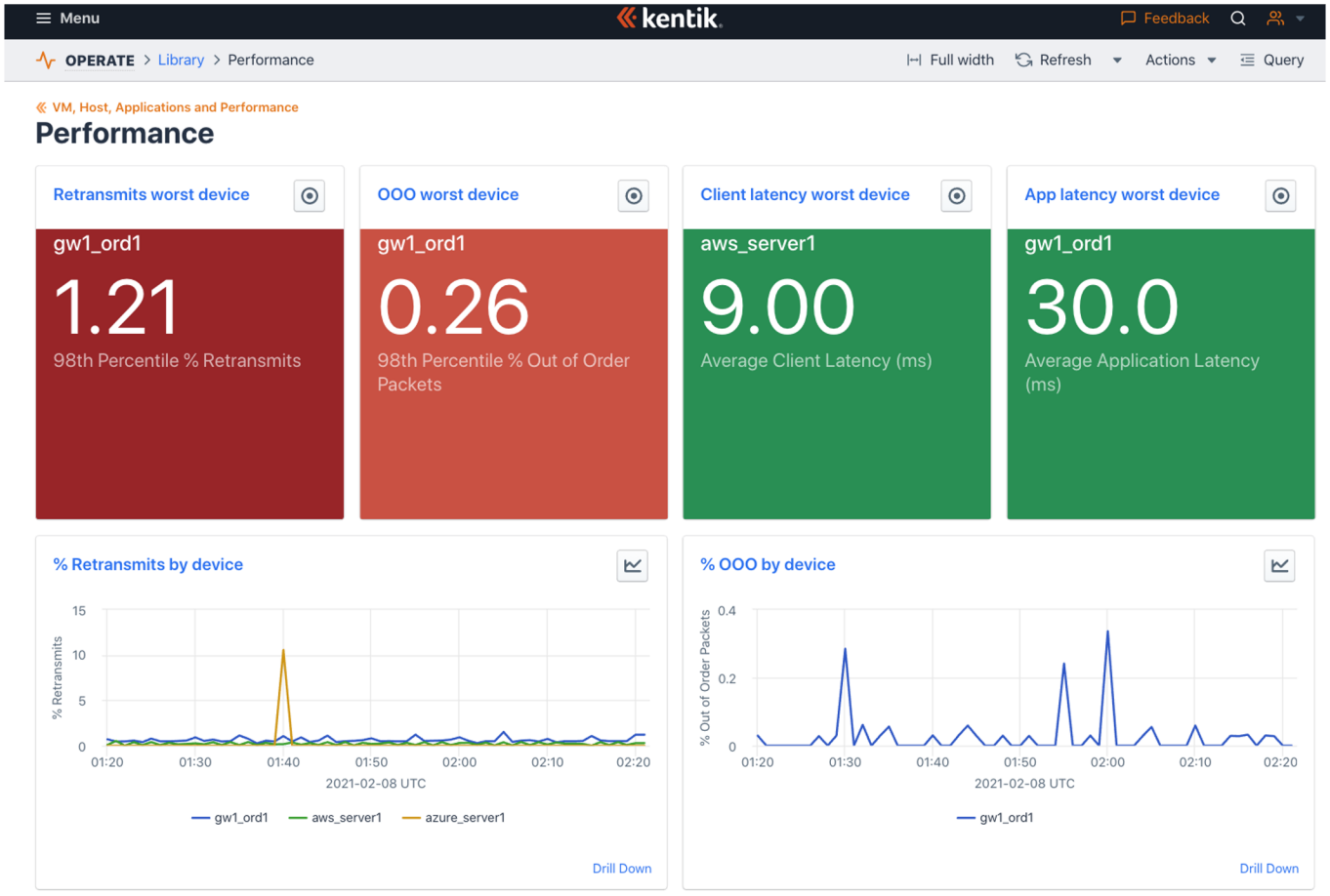
# Same Data Source, Multiple Destinations: Grafana



# Same Data Source, Multiple Destinations: New Relic



# Same Data Source, Multiple Destinations: Kentik



# How to participate

- Download: https://github.com/kentik/ktranslate
- Knowledge Base: https://kb.kentik.com/v0/Fc19.htm
- Community-based support on github and via Discord: https://discord.gg/kentik

# ktranslate Roadmap

# Overall Roadmap

- Our short-term road map is to extend ktranslate into a general observbaility tool, going deeper into network types but also broader in observability

- Current ktranslate focuses:
    - **Sources**:  Streaming Telemetry, Router API, Router CLI, Logs, non-network Metrics
    - **Destinations**: ST, SNMP API
    - **Enrichments**: Generic lookup/streaming join enrichment
    - **Formats**: Adding network-specific output formats
    - **Transformation**: Mapping ST and SNMP semantically
    - **Operability**
        - Telemetry for data observability of streams through ktranslate
        - k8s-based clustering for scale and core/edge deployment

# Use Case Focus: Universal Network Metrics Translation

- The problem
  - Device metrics is a bit of a mess
  - Most networks have devices that don’t support Streaming Telemetry
  - Not trivial to make sense of SNMP + Streaming Telemetry
  - In theory there’s one source of truth and different semantics
  - In practice, not always (esp. cross-vendor)
  - Still requires customer or vendor work to translate

# Use Case Focus: Universal Network Metrics Translation

- Our focus:
  - Use ktranslate to collect, translate, replicate, transform
  - From any of ST, SNMP, API, CLI
  - Sending to all systems that need it
  - Please ping us if interested in working with us! (in the open)

- Other projects:
  - Verizon's panoptes (SNMP-focused)
  - Netflix's gnmi-gateway (very ST-focused)
  - MLB's netpaca-optics (more decision logic)
  - Cisco's (Cisco-focused) Pipeline/bigmuddy: https://github.com/CiscoDevNet/bigmuddy-network-telemetry-pipeline

kentik®

Questions?

Avi Freedman

avi@kentik.com

@avifreedman