Network Observability
## with IPFIX, Prometheus and Elastic Stack

Rastislav Szabó

rastislav.szabo@pantheon.tech

About the Speaker – Rastislav Szabó

- Staff Engineer at PANTHEON.tech
- Architecting software solutions for networking industry

- Previously focused on network manageablity using NETCONF + YANG
- Currently working on cloud-native networking infrastructure projects

- Open-source contributions: Sysrepo, FD.io, Ligato, Contiv-VPP

PANTHEON.tech

Motivation for Network Observability in K8s

- Network failure identification & alerting
  - unexpected congestion, packet drop, ...
  - between pods on the same node & in the underlying network

- Identification of the bottlenecks
  - equal traffic distribution in the cluster
  - limits of large scale deployments

- Malicious activity detection & investigation

- CNFs (Cloud-Native Network Functions) deployments
  - all of the above becomes even more important
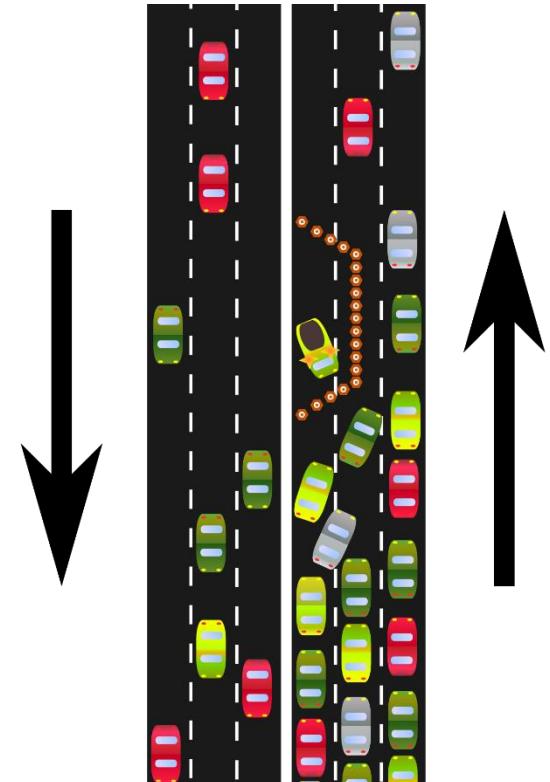
https://en.wikipedia.org/wiki/Traffic_bottleneck

PANTHEON.tech

Options for Network Observability in K8s

- Metrics served by CNI plugins
  - many CNIs export metrics in Prometheus format
  - only some of them actually export helpful data

- Service mesh metrics
  - Istio can collect TCP telemetry data and export them via Prometheus

- DIY / 3rd party tool metrics
  - e.g. monitoring network interfaces within each pod's network namespace

- Metrics are not enough
 - not enough for deeper analysis, e.g. in case of security incidents

Prometheus + Grafana

PANTHEON
.tech
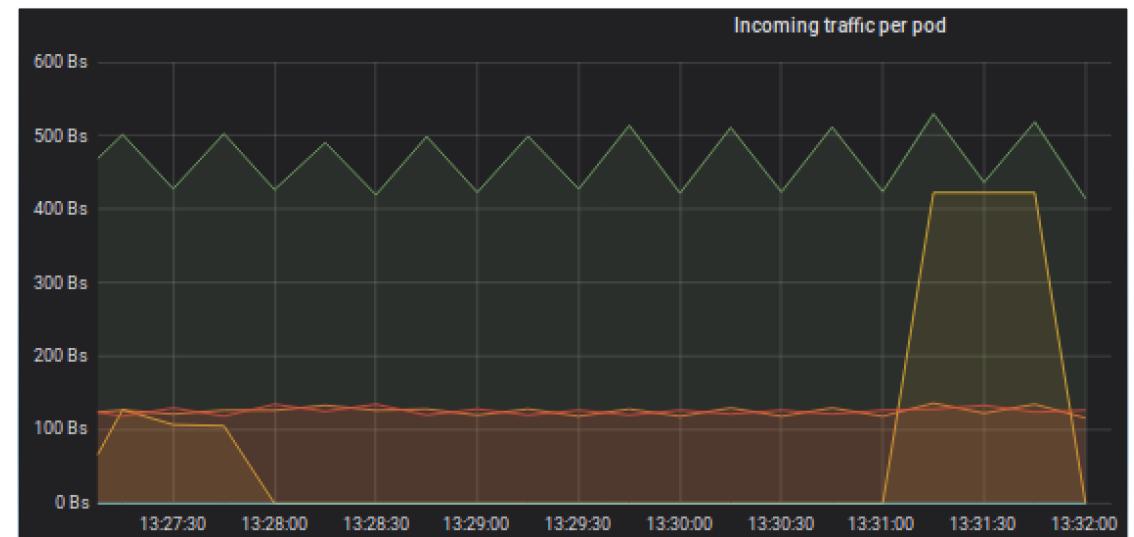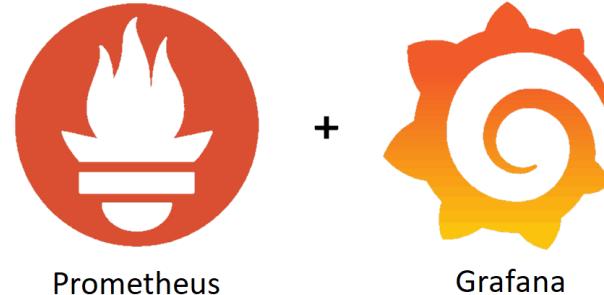
# Per-Pod Interface Metrics by Contiv-VPP CNI

- Good for generating alerts, spotting issues, etc.

- Cannot go back in the history and look e.g. at the details of the traffic that caused a spike on the graph

Network Observability in Traditional Networks

NetFlow / IPFIX
(IP Flow Information Export)

- Protocols for exporting information about each network conversation (flow)

- Flow: n-tuple: src/dst IP+port, IP protocol, ToS, interface, packet + data counts, timestamps, ...

- Flow exporters: routers, switches, probes, other network devices

- Flow collector: reception, storage and pre-processing of flow data

- Analysis tool: analyzes received flow data

https://en.wikipedia.org/wiki/NetFlow

PANTHEON
.tech
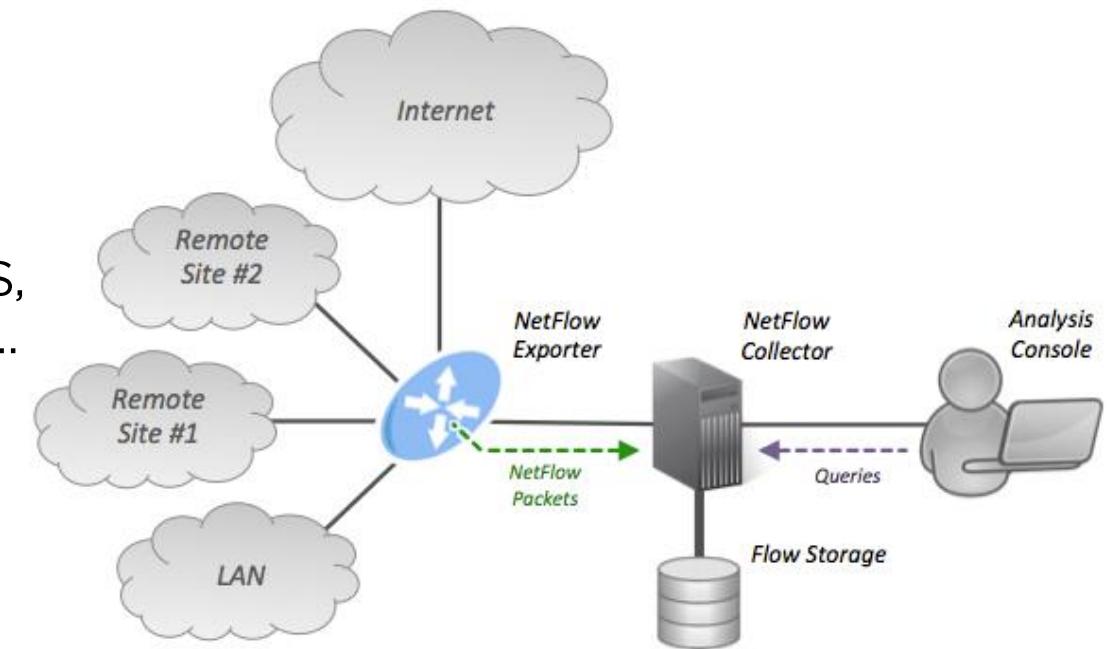
IPFIX in Kubernetes

- Flow exporter: CNI plugin
  - CNI plugin acts as a router/switch between the pods
  - each CNI does the networking differently (e.g. multi-interface pods)
  - traffic is often encapsulated on the way between the nodes

- Cloud-native collector & analyzer can be built using the ELK stack:
  - flow collector: Logstash
  - flow storage: Elasticsearch
  - analysis tool: Kibana

Kubernetes node

pod   pod   pod

CNI   kubernetes

Kubernetes node

pod   pod   pod

CNI   kubernetes

flow packets

Logstash = flow collector

Elasticsearch = flow storage

Operator

Kibana = analysis tool

PANTHEON
.tech
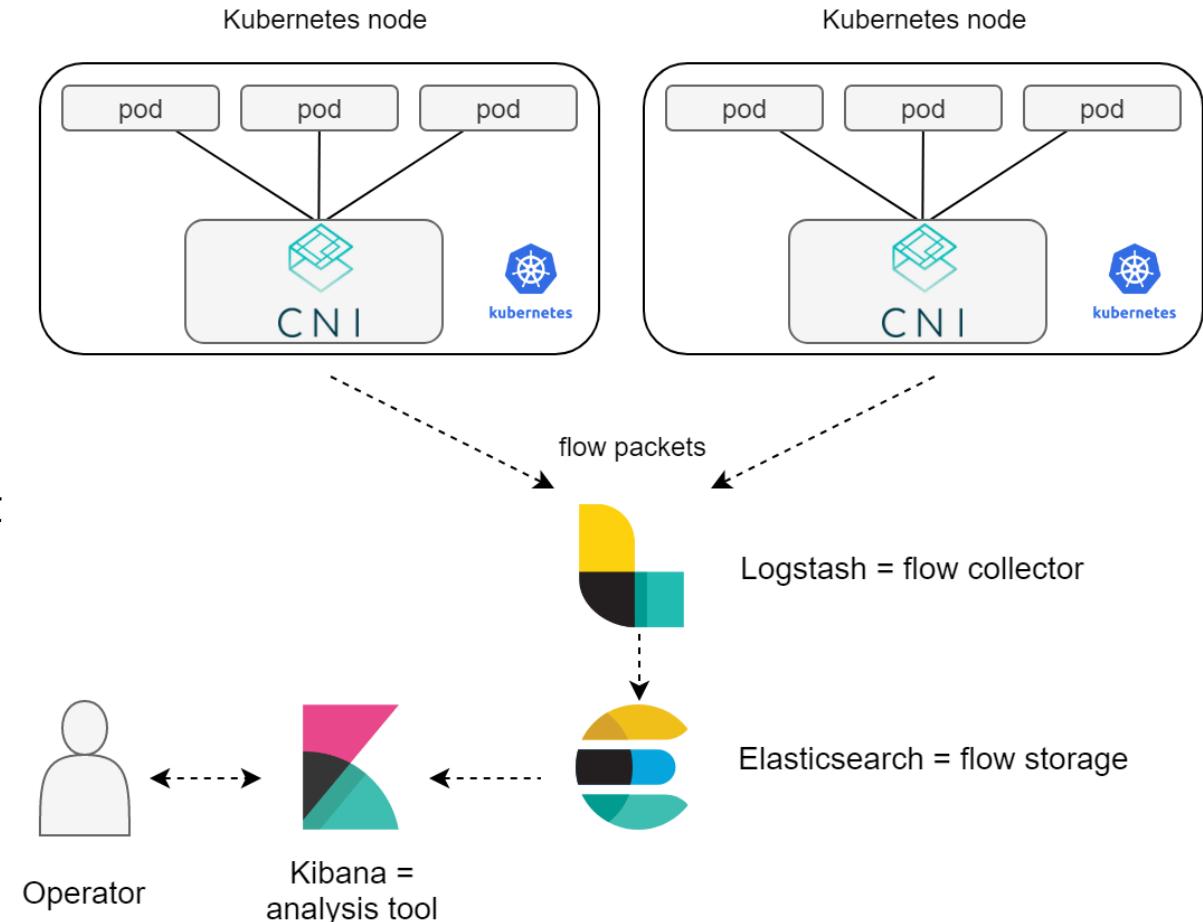
# Enabling IPFIX Export in Contiv-VPP CNI

Contiv-VPP (contivpp.io)

- CNI plugin based on FD.io VPP vSwitch (dataplane) running as a userspace process

- Focused on speed:
  - Vector Packet Processing
  - kube-proxy functionality in the userspace
  - memif interfaces

- Provides features aimed for CNFs (Cloud-Native Network Functions) deployments:
  - multiple pod interfaces
  - service function chaining between the pods

- VPP supports IPFIX - it just needs to be enabled
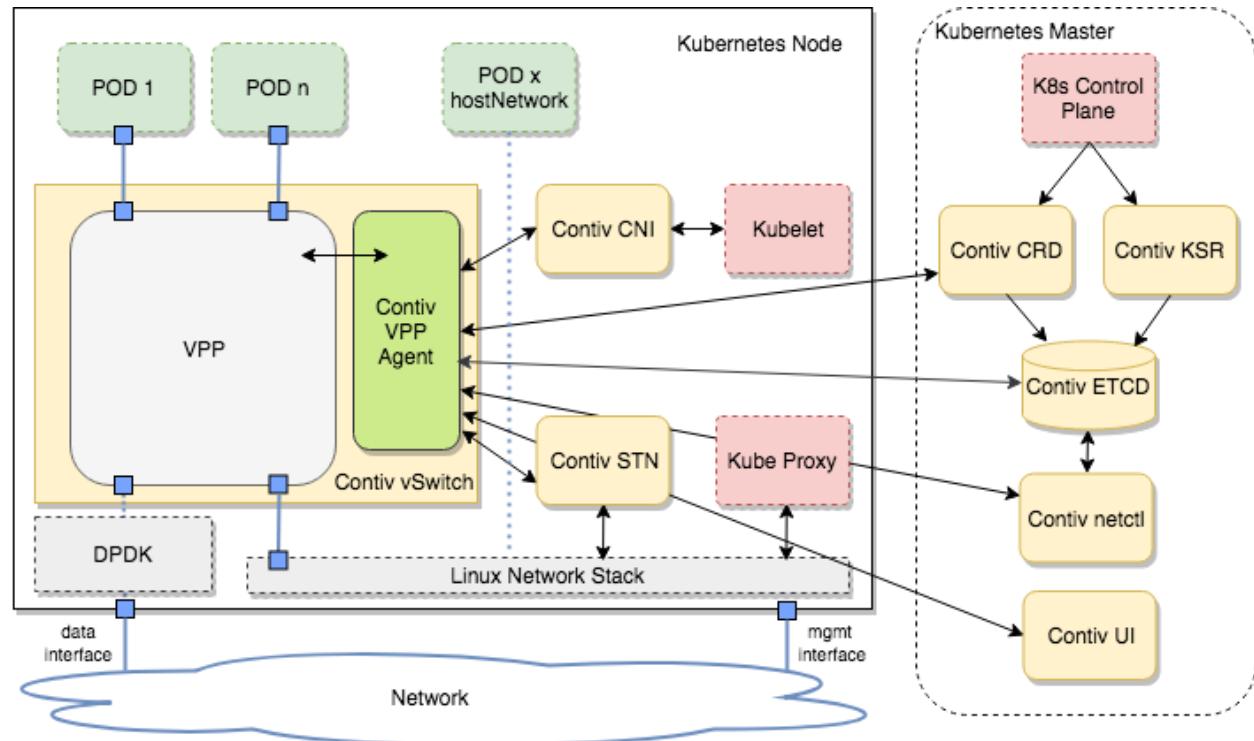
Enabling IPFIX Export in Contiv-VPP CNI

The Contiv-VPP CNI is modular and easily extendable. IPFIX support can be added by writing two tiny plugins:

- Contiv-VPP IPFIX plugin:
  - to enable IPFIX on each vSwitch (for each pod interface)
  - calls ligato.io API

- Ligato.io VPP Agent IPFIX plugin:
  - to enable IPFIX on VPP
  - calls VPP binary API via GoVPP

- FD.io VPP (data plane)
  - already contains IPFIX support
  - but if it was needed, it is extendable via plugins as well

Contiv

Ligato API

VPP API

PANTHEON
.tech
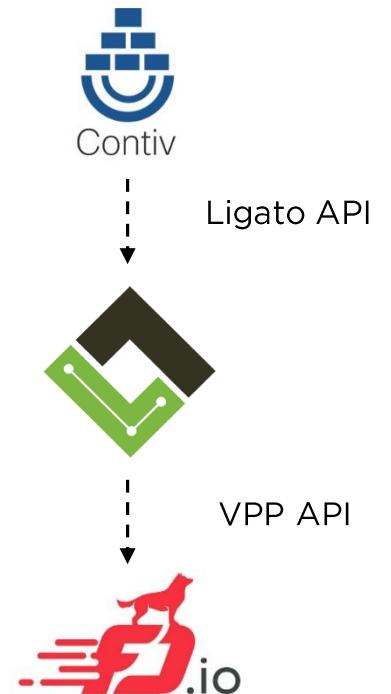
IPFIX Flow Collector & Analyzer based on ELK

github.com/robcowart/elastiflow:

- provides ready-to use ELK-based IPFIX collector & analyzer solution
- Logstash IPFIX/NetFlow/sFlow codec config & filters feeding Elasticsearch
- Kibana dashboards definitions

- Packaged into Docker containers and deployed in the K8s cluster

- Contiv-VPP CNI was configured to send the flow records into the Logstash pod
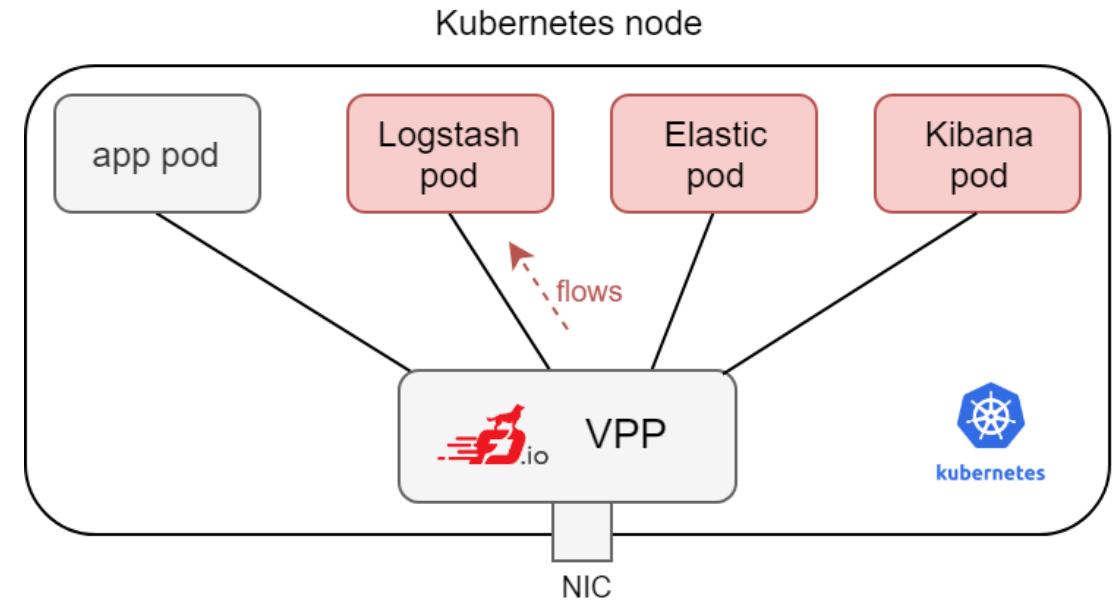

```
$ kubectl get pods
NAME                     READY    STATUS     RESTARTS    AGE
elasticsearch            1/1      Running    0           6d
elastiflow-logstash      1/1      Running    0           6d
kibana                   1/1      Running    0           6d
```

PANTHEON.tech

Kibana/ View on Traffic Flows Between Pods
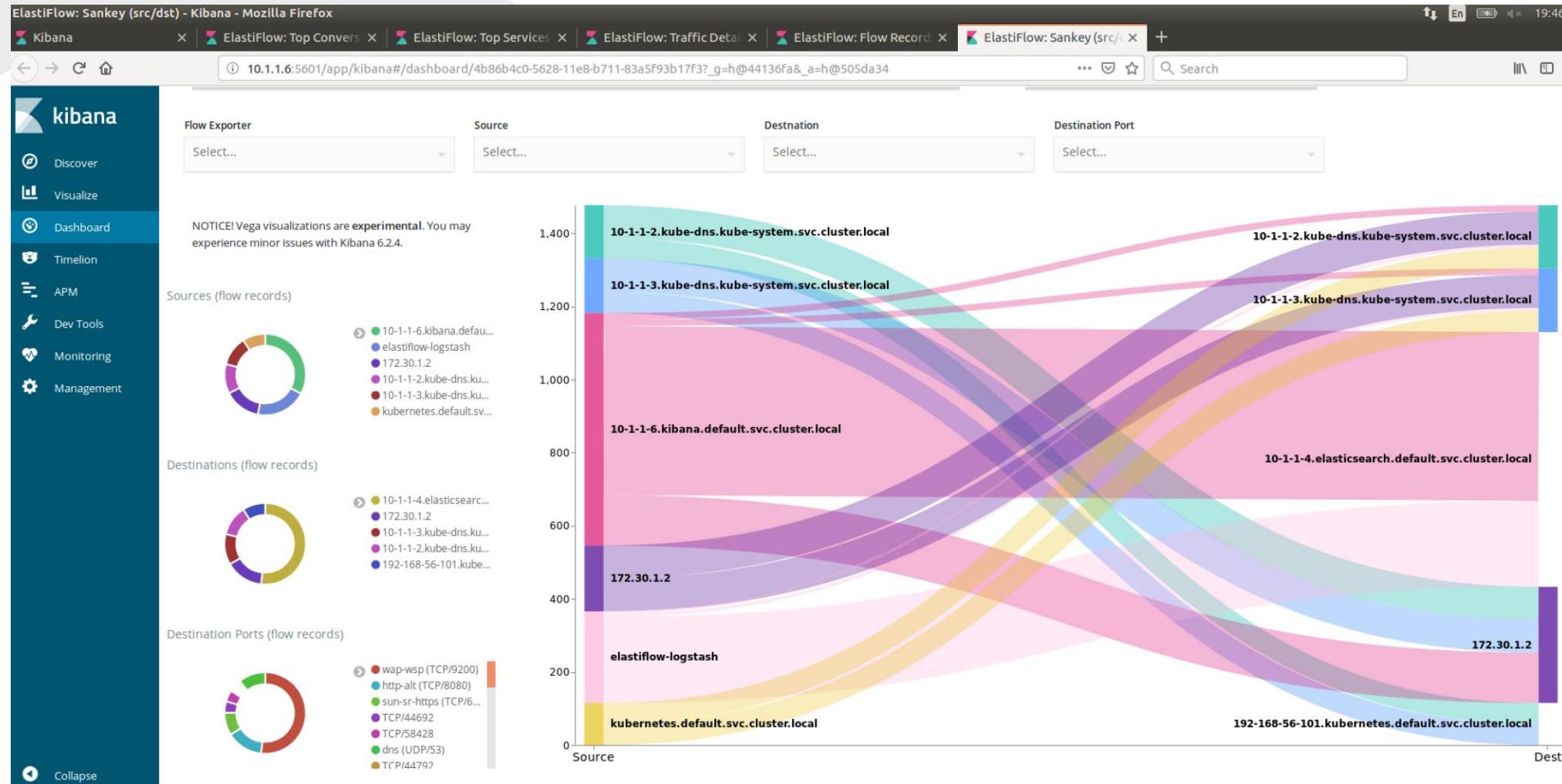
Kibana/ Traffic Details
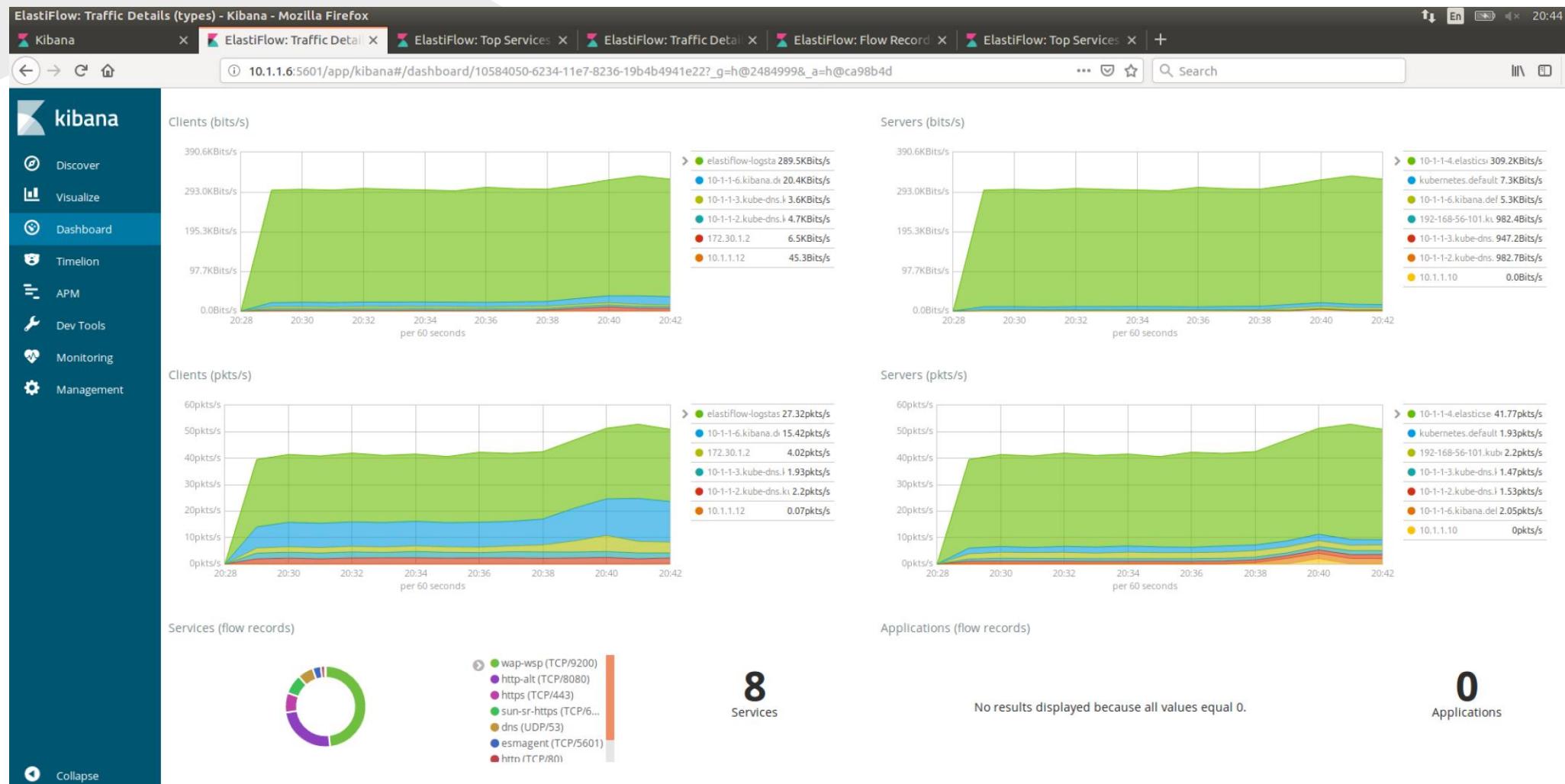
Kibana/ Detailed Flow View

PANTHE⊙N
.tech

Possible Enhancements/ **Scaling**

- One Logstash pod on each node to keep the VPP-to-Logstash flow traffic within the same node

- Clustered Elastisearch deployment (covered by k8s service), ideally keep Logstash-to-Elastic traffic node-local as well

- One Kibana pod is enough (only a user interface)

PANTHEON.tech
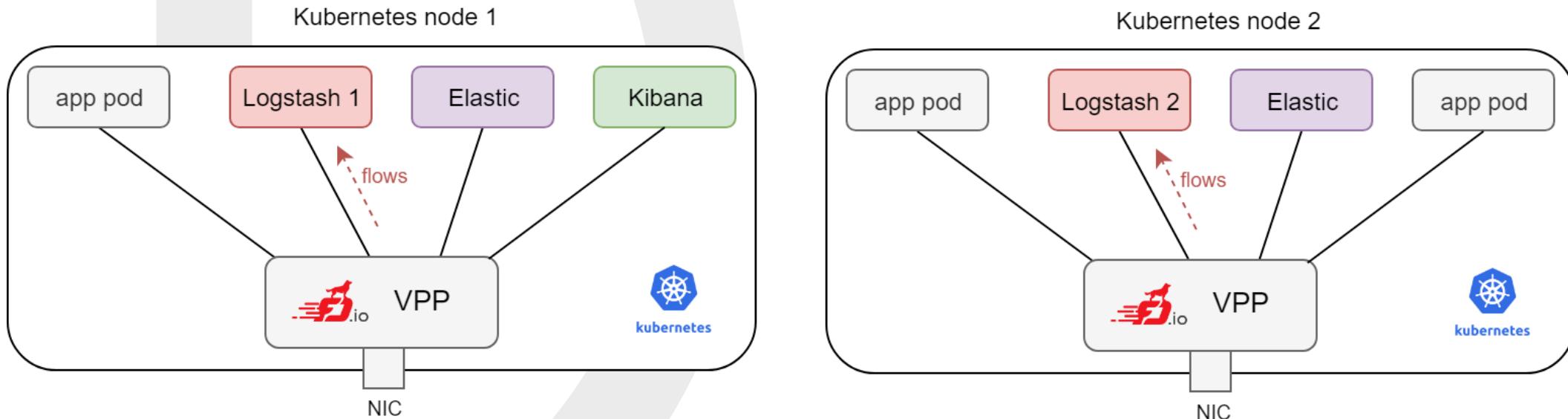
Possible Enhancements/ **More Optimizations**

- Use memif between VPP and flow collector pod

- Use more lightweight flow collector (e.g. github.com/cloudflare/goflow), integrate with memif

- Add Elasticsearch source into Prometheus to provide more metrics
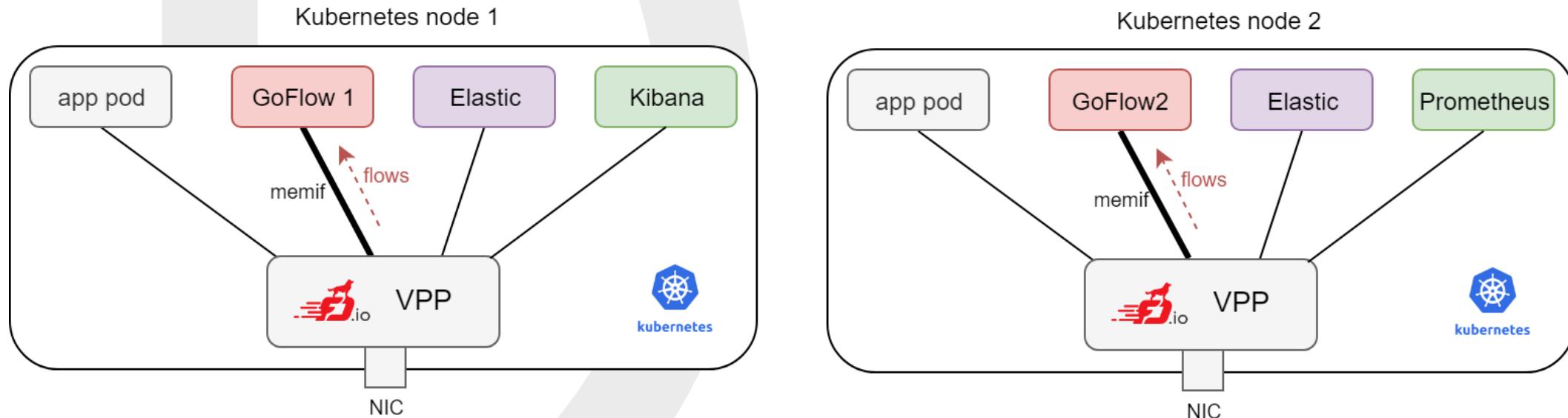
Thank You

Rastislav Szabo

rastislav.szabo@pantheon.tech

https://pantheon.tech