

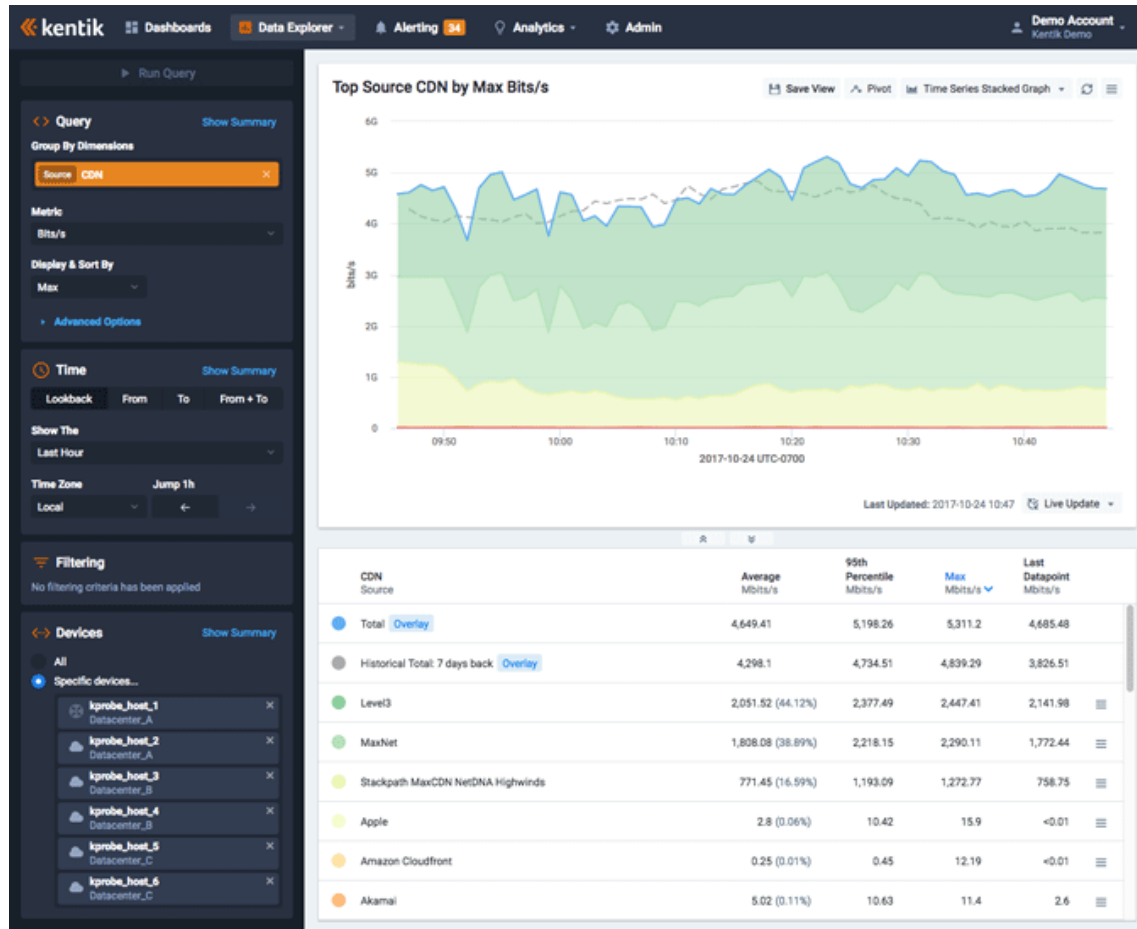
# Akvorado: a Flow Collector and Visualizer

Vincent Bernat

2022-09-16 — FRnOG 36

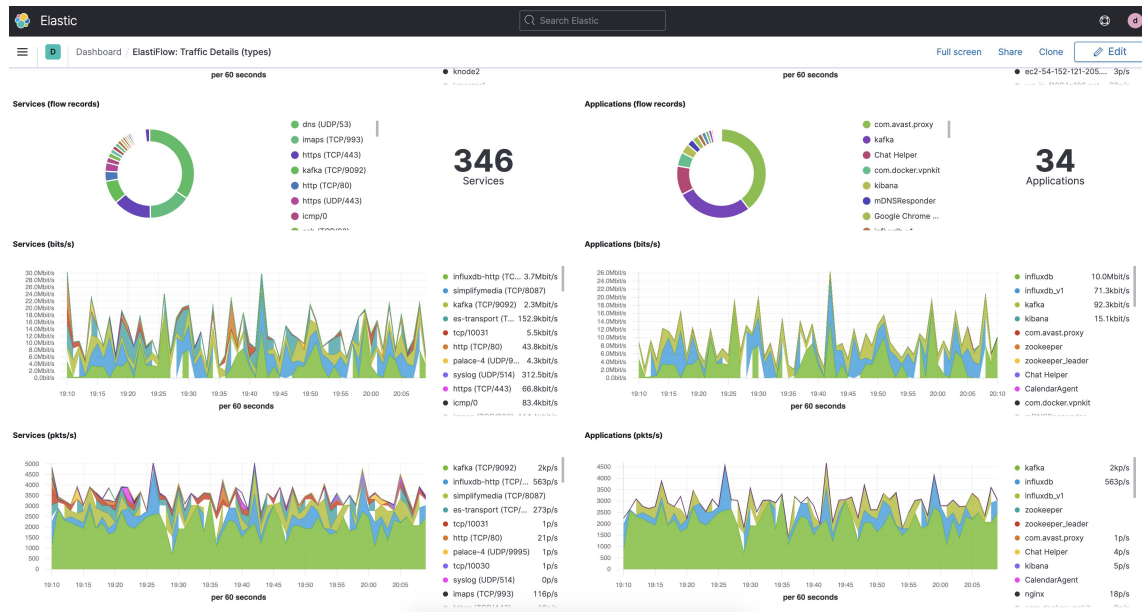
# Existing solutions

# Commercial solution: Kentik



- Easy to setup
- Featureful
- Hosted in the cloud
- Expensive
- Not open source

# Commercial solution: Elastiflow



- Community edition available for free (as in beer) for low volume
- Self-hosted
- Expensive
- Not open source

## DIY solution with open source

- pmacct + RabbitMQ + Elasticsearch + Kibana
- GoFlow2 + Kafka + ClickHouse + Grafana
- vflow + Kafka + Apache Pinot + Apache Superset
- Free (as in beer, as in speech)
- Flexible
- Self-hosted
- Some assembly required

# New player: Akvorado

- NetFlow/IPFIX/sFlow collector
- Enrich data (GeoIP, interface names, classification)
- Serialize with Protobuf and send to **Kafka**
- Store data in **ClickHouse**
- Web UI to query data

## Selling points

- Free (as in beer, as in speech)
- Performant
- No assembly required
- Web UI included
- Self-hosted

# Free

- Published under the AGPLv3 license
- Developed by AS 12322
- No plan to sell anything



# Performant

- Can be deployed on a single VM
- 1 TB, 64 GB, 24 vCPU
- 100k flows/s
- 5 year worth of data
- Older data is trimmed:
  - 1-minute, 5-minutes, 1-hour resolution
  - IP addresses and ports removed


## No assembly required

- Getting started with docker - compose up
- No knowledge of Kafka/ClickHouse required
- Special attention to simplify the operations (logs, metrics, backward compatibility of configuration files, automatic migrations)
- No magic: you still need to read the documentation



## Shipped with a web frontend

- See your data immediately
- Test it on [demo.akvorado.net](https://demo.akvorado.net)

# Homepage

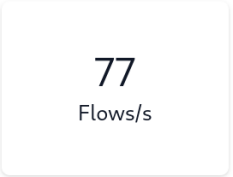
 **Akvorado**  
v1.5.7

[Home](#) [Visualize](#) [Documentation](#)

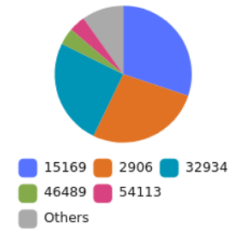
 



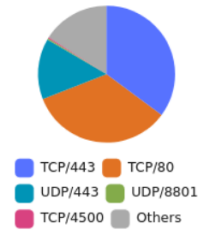
Akvorado is a flow collector, hydrater and exporter. It receives flows, adds some data like interface names and countries, and exports them to Kafka.



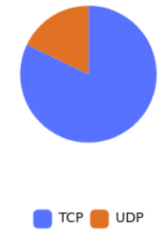
Top source AS



Top source ports



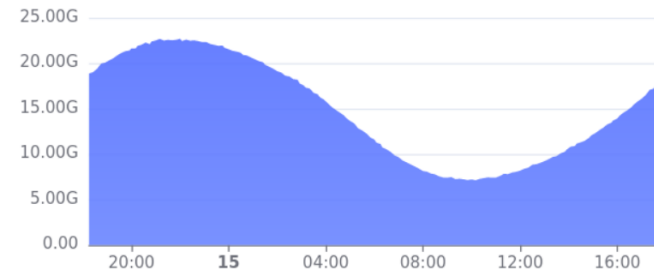
Top protocols



Top source countries



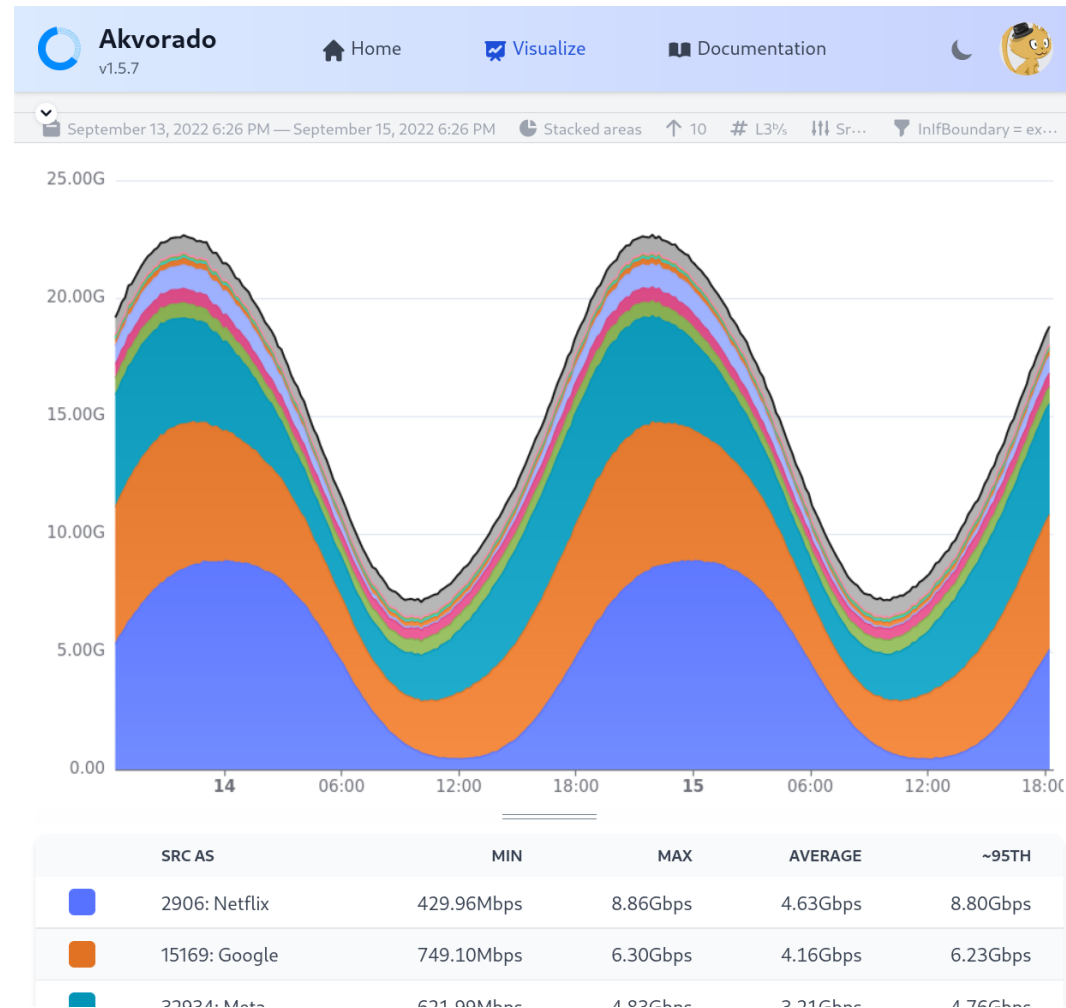
IPv4/IPv6



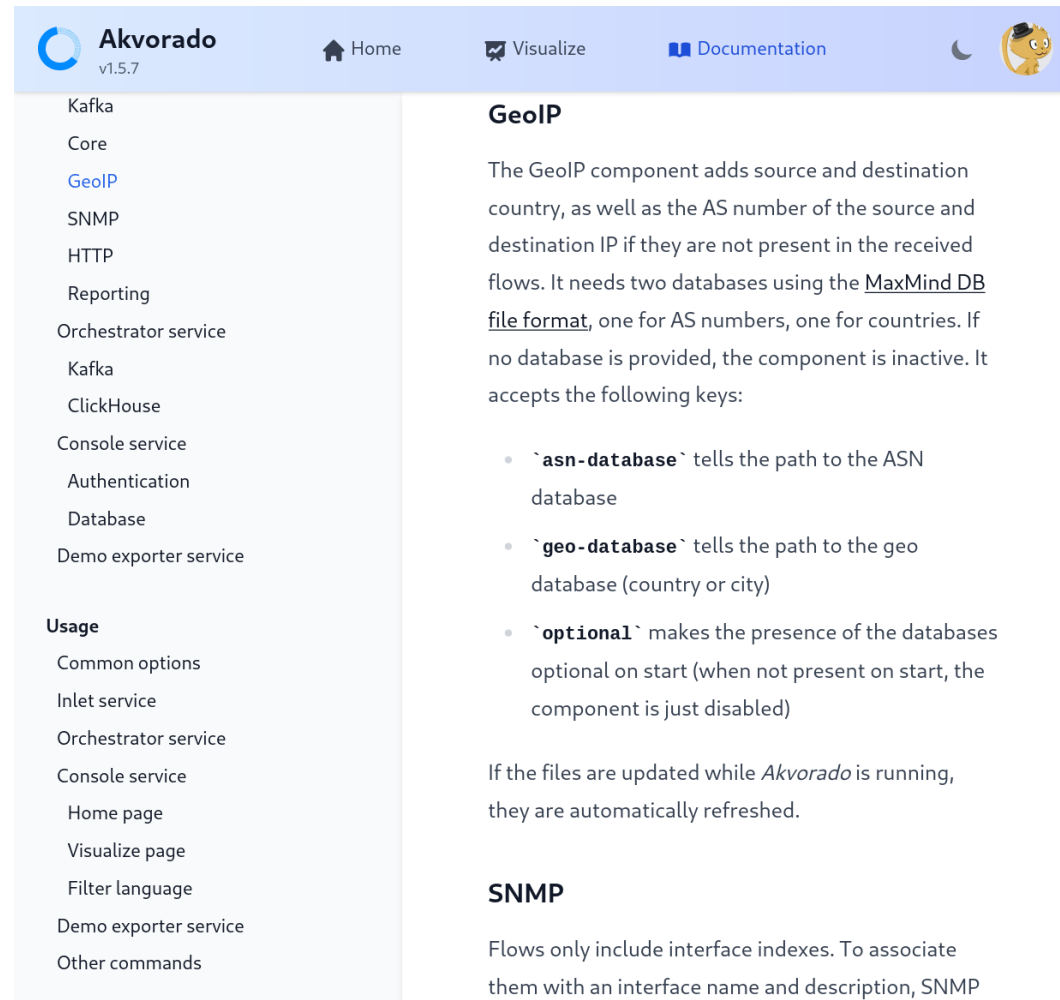
# Dark mode



# Visualize



# Documentation



The screenshot shows the Akvorado v1.5.7 documentation page. The top navigation bar includes links for Home, Visualize, and Documentation. The left sidebar lists various components and services, with GeolP highlighted. The main content area is titled "GeolP" and describes its function in adding source and destination country and AS number information to flows. It lists configuration keys: `asn-database`, `geo-database`, and `optional`. A note mentions automatic refresh of files during runtime. The "Usage" section is partially visible at the bottom.

**Akvorado**  
v1.5.7

Home Visualize Documentation

Kafka  
Core  
**GeolP**  
SNMP  
HTTP  
Reporting  
Orchestrator service  
Kafka  
ClickHouse  
Console service  
Authentication  
Database  
Demo exporter service

**Usage**  
Common options  
Inlet service  
Orchestrator service  
Console service  
Home page  
Visualize page  
Filter language  
Demo exporter service  
Other commands

## GeolP

The GeolP component adds source and destination country, as well as the AS number of the source and destination IP if they are not present in the received flows. It needs two databases using the [MaxMind DB file format](#), one for AS numbers, one for countries. If no database is provided, the component is inactive. It accepts the following keys:

- `asn-database`` tells the path to the ASN database
- `geo-database`` tells the path to the geo database (country or city)
- `optional`` makes the presence of the databases optional on start (when not present on start, the component is just disabled)

If the files are updated while *Akvorado* is running, they are automatically refreshed.

## SNMP

Flows only include interface indexes. To associate them with an interface name and description, SNMP

# Getting started



## Five first minutes

1. Grab the tarball from [GitHub](#)
2. Unpack and rename the directory to akvorado
3. `docker-compose up`
4. Point your browser to port 8081

## Next hour

1. Grab the tarball, unpack and rename the directory.
2. Remove the demo-exporter section from `akvorado.yaml`
3. Remove `akvorado-exporter*` from `docker-compose.yaml`
4. Configure everything else!

# Flows

- No configuration needed on Akvorado side.
- Send IPFIX/Netflow to port 2055.
- Send sFlow to port 6343.

# SNMP

- Interface names and descriptions with SNMP polling
- The source IP used for flows is used to query SNMP
- This is a mandatory step
- Configure the SNMP community in `akvorado.yaml`
- Check the documentation for details

# Router classification

- A router can be attached to a group, role, site, region, and tenant
- Classification is done through rules
- This is not a mandatory step
- Brush up on your regex skills!

## exporter-classifiers:

- `ClassifySiteRegex(Exporter.Name, "^(^[-]+)-", "$1")`
- `Exporter.Name endsWith ".it" && ClassifyRegion("italy")`
- `Exporter.Name matches "^(washington|newyork).*" && ClassifyRegion("usa")`
- `Exporter.Name endsWith ".fr" && ClassifyRegion("france")`

# Interface classification

- Boundary: external, internal, or unclassified.
- Connectivity type: transit, PNI, PPNI, IX, ...
- Provider: Cogent, Telia, Scaleway, ...
- This step is not mandatory, but highly recommended.

## interface-classifiers:

- |  
ClassifyConnectivityRegex(Interface.Description, "(?i)(transit|pni|ppni|ix):? ", "\$1")  
ClassifyProviderRegex(Interface.Description, "[^ ]+ ([^ ]+)", "\$1") &&  
ClassifyExternal()  
- ClassifyInternal()

## Remaining steps

- Check the web interface
- Check the [configuration section](#) in the documentation
- Check the [troubleshooting section](#) too

# Web UI

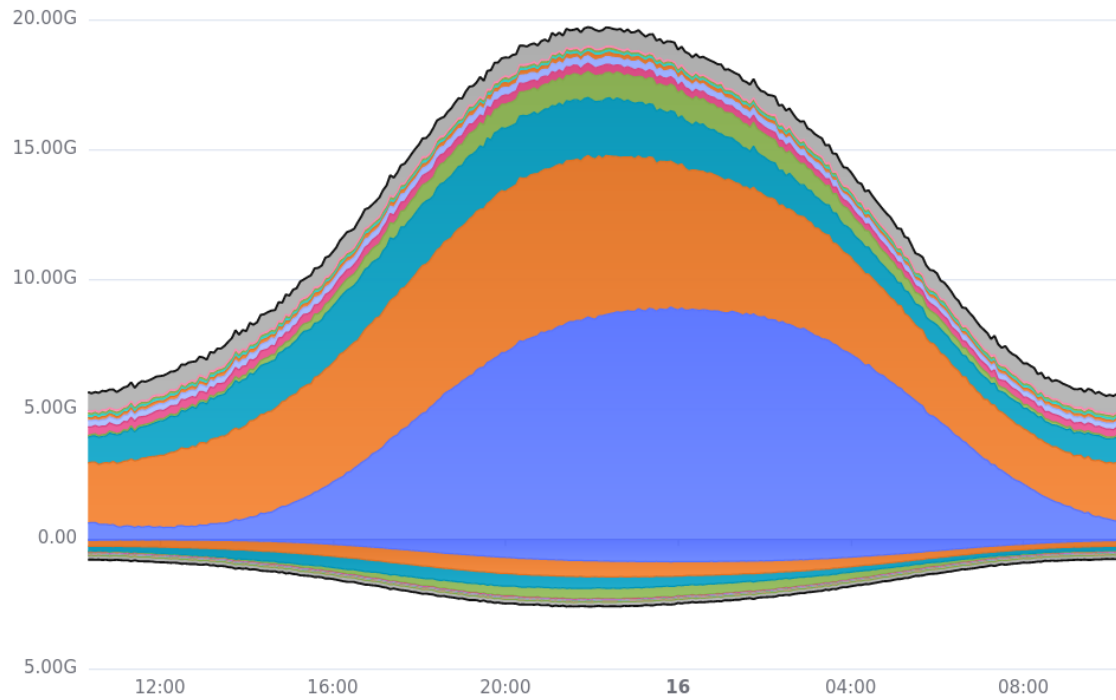
*free*

The image shows a configuration panel for the Akvorado web UI. It includes a 'Unit' selector with options 'L3<sup>b/s</sup>', 'L2<sup>b/s</sup>', and 'P/s', and a blue 'Refresh' button. Below is a 'Graph type' dropdown menu set to 'Stacked areas'. There are two checkboxes: 'Bidirectional' and 'Previous period'. The 'Time range' section has a 'Presets' dropdown menu with 'Last 6 hours' selected and a 'Search...' field. Below this are 'Start' (6 hours ago) and 'End' (now) fields. The 'Dimensions' section has a 'Dimensions' dropdown menu with 'SrcAS X' selected and a 'Search...' field. Below this is a 'Limit' field set to '10'. The 'Filter' section has a 'Filter' label, a 'Ctrl-Space for completions' hint, and a text input field containing 'InIfBoundary = external'. At the bottom is a 'Saved filters' dropdown menu with a 'Search...' field.

- Graph type: stacked areas, lines, grid, or sankey
- Time range: preset or manually specified
- Dimensions: group data by AS, countries, providers, ...
- Filters: using an SQL-like language

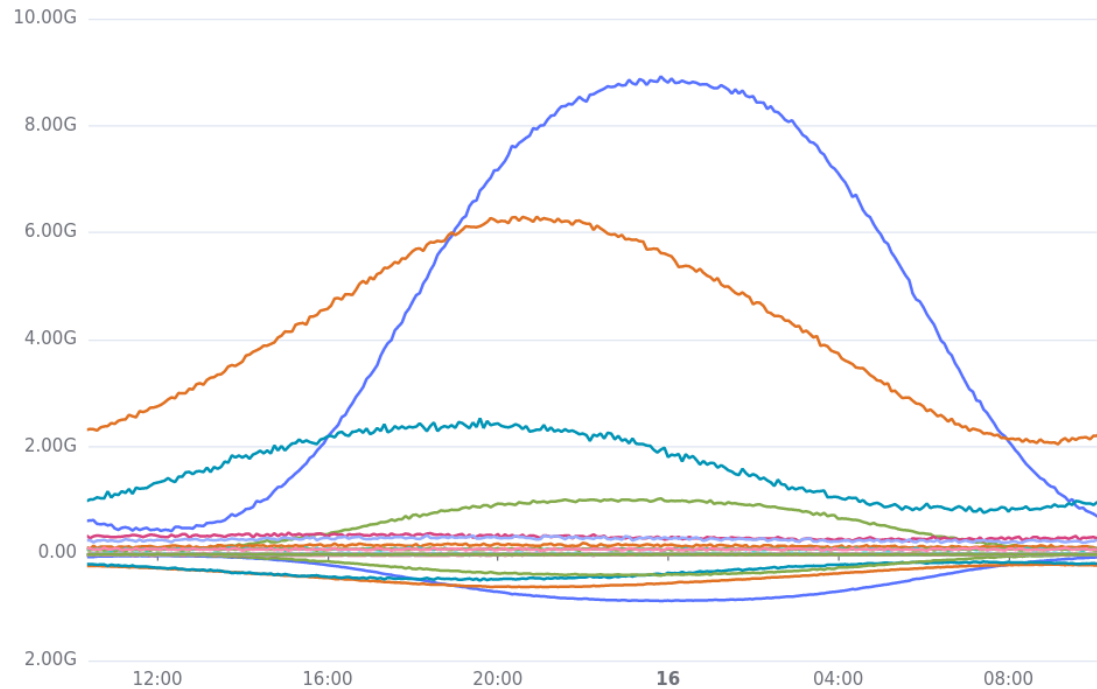


# Examples



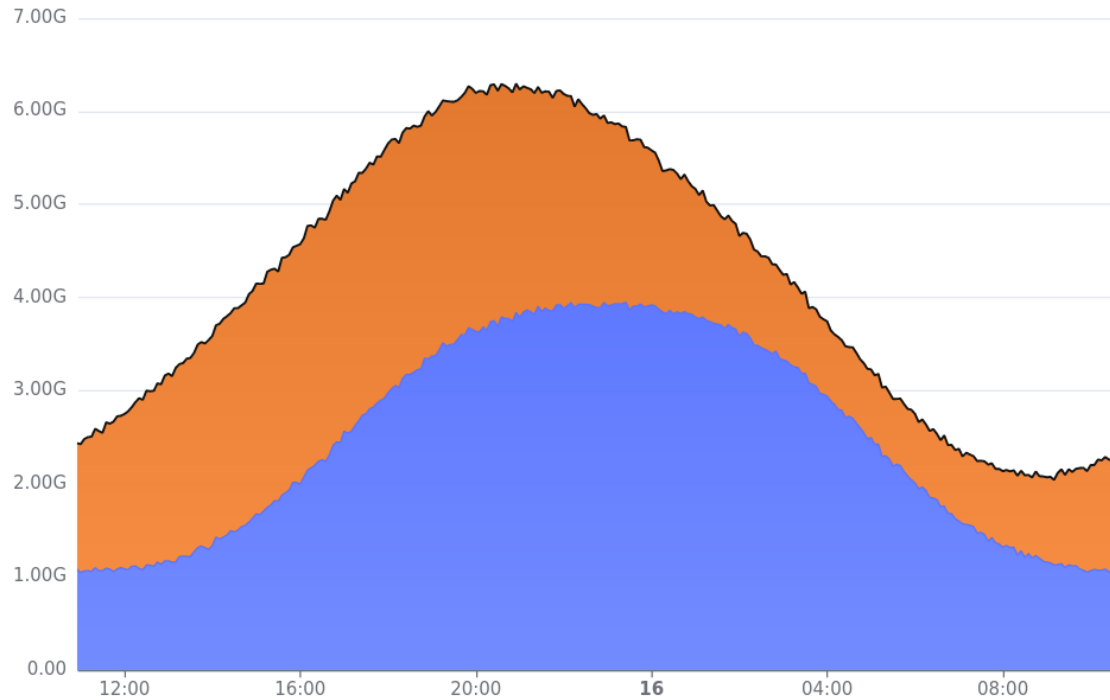
- Stacked areas
- Dimensions: SrcAS
- InIfBoundary = external  
AND InIfProvider =  
"cogent"

# Examples



- Lines
- Dimensions: SrcAS
- InIfBoundary = external  
AND InIfProvider =  
"cogent"

# Examples



	ETYPE	MIN	MAX	AVERAGE	~95TH
■	IPv6	1.04Gbps	3.95Gbps	2.51Gbps	3.92Gbps
■	IPv4	702.26Mbps	2.68Gbps	1.69Gbps	2.63Gbps

- Stacked areas
- Dimensions: EType
- InIfBoundary = external  
AND SrcAS IN (AS15169,  
AS36040)

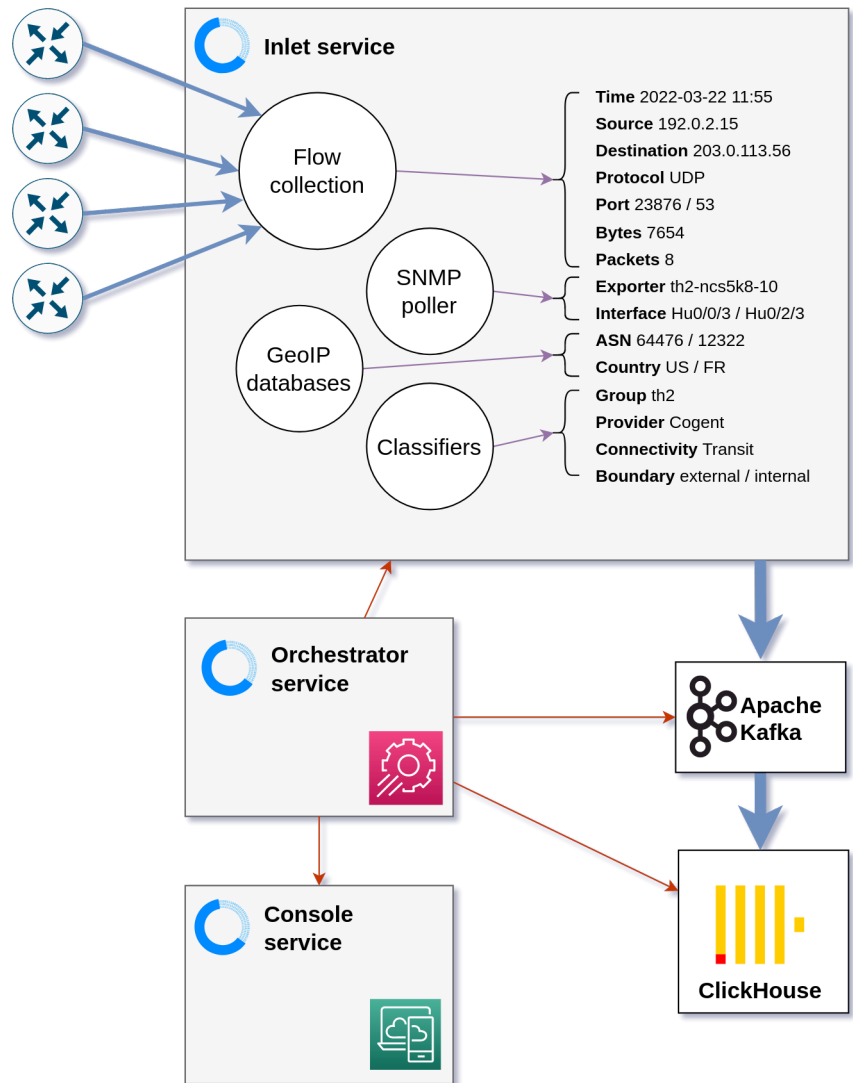


## About AS names

- PeeringDB: most likely correct but incomplete
- RIR: complete but funny names (2906 is AS-SSI)
- Mix them, apply some cleaning rules, export as CSV, build weekly: <https://vincentbernat.github.io/asn2org/>
- Also use **DB-IP** as ARIN refuses to make the data available

# Under the hood

*free*



- Written in Go
- **Inlet**: receive flows, add data, serialize to protobuf, send to Kafka
- **Console**: Web UI (including an API)
- **Orchestrator**: configure Kafka, ClickHouse, inlet, and console

## Open source components (backend)

- [GoFlow2](#) for collecting flows
- [Shopify's sarama](#) to talk with Kafka
- [Prometheus Go client](#) for metrics
- [zeroLog](#) for logging
- [Cobra](#) for command-line handling
- Anton Medvedev's [expr](#) language
- [Gin Web Framework](#)
- [mapstructure](#) + [validator](#) for configuration files

# Open source components (infrastructure)

- Apache Kafka as a bus message
  - Producers and consumers can be distributed
  - Well-known and low overhead
- ClickHouse as a database
  - Columnar database
  - SQL-like language
  - Super performant and low effort
  - Many functions



# Open source components (frontend)

- [Vue.js](#) as the reactive framework
- [Vite](#) as the builder
- [Tailwind](#) as the CSS framework
- [Headless UI](#) and [Flowbite](#) for components
- [CodeMirror](#) as the text editor for filters
- [Apache ECharts](#) for graphs

# Anti-DDOS

- Query ClickHouse every 10 seconds
- Build a FlowSpec rule to mitigate the attack

# ClickHouse query

```
SELECT TimeReceived, DstAddr, protocol, SrcPort, Gbps, flows, sources, countries
FROM (SELECT
  toStartOfMinute(TimeReceived) AS TimeReceived,
  dictGetOrDefault('protocols', 'name', Proto, '???') AS protocol,
  DstAddr, SrcPort,
  SUM(Bytes*SamplingRate*8/1000/1000/1000)/300 AS Gbps,
  quantile(0.1)(Bytes/Packets) AS size, COUNT(*) AS flows,
  uniq(SrcAddr) AS sources, uniq(SrcCountry) AS countries
FROM flows
WHERE TimeReceived > date_sub(minute, 5, now())
GROUP BY TimeReceived, DstAddr, protocol, SrcPort
)
WHERE (sources > 50 OR countries > 10)
AND Gbps > 0.2 AND protocol = 'UDP'
ORDER BY TimeReceived DESC, Gbps DESC
```

# FlowSpec rule

```
# Time: 2022-09-15 23:55:00 - 2022-09-15 23:55:00
# Source: 198.0.2.111, protocol: UDP, port: 123
# Gbps/Mpps: 1.61/0.432, packet size: 468<=X<=468
# Flows: 544, sources: 413, countries: 57

route flow4 {
  dst 78.193.168.168/32;
  sport = 123;
  length = 468;
  proto = 17;
}{
  bgp_ext_community.add((generic, 0x80060000, 0x00000000));
  bgp_community.add((65535, 65282));
};
```

# Future plans

- BGP support for AS paths, communities, origin AS
- Forecasting and anomaly detection
- VRF support
- Dashboards

# Questions?

- GitHub: <https://github.com/akvorado/akvorado>
- Demo: <https://demo.akvorado.net>



