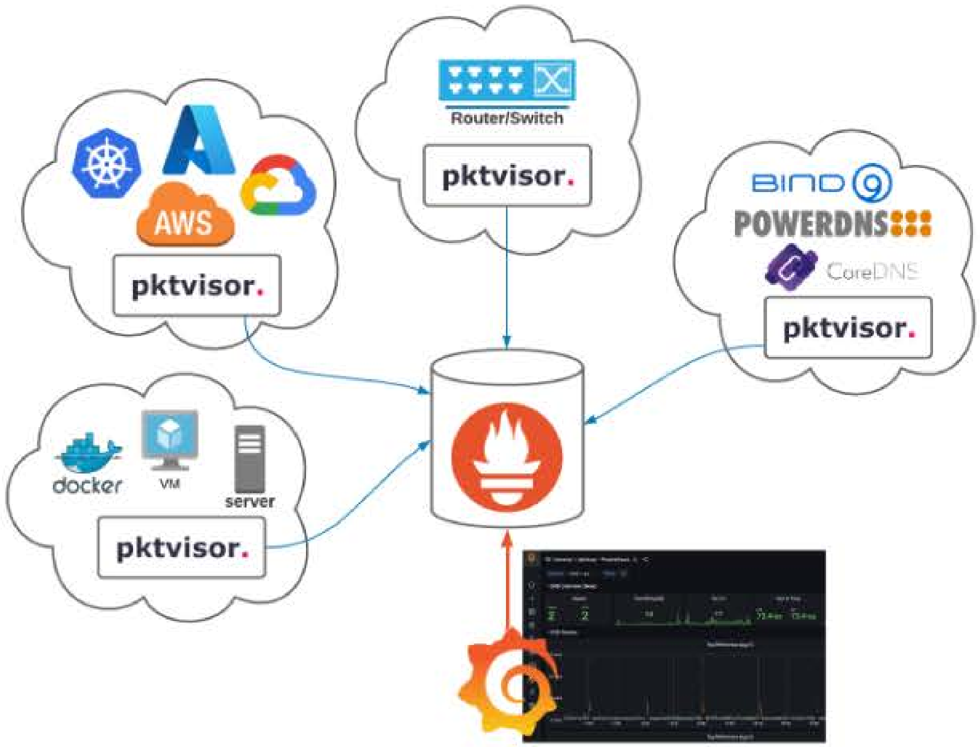
pktvisor.

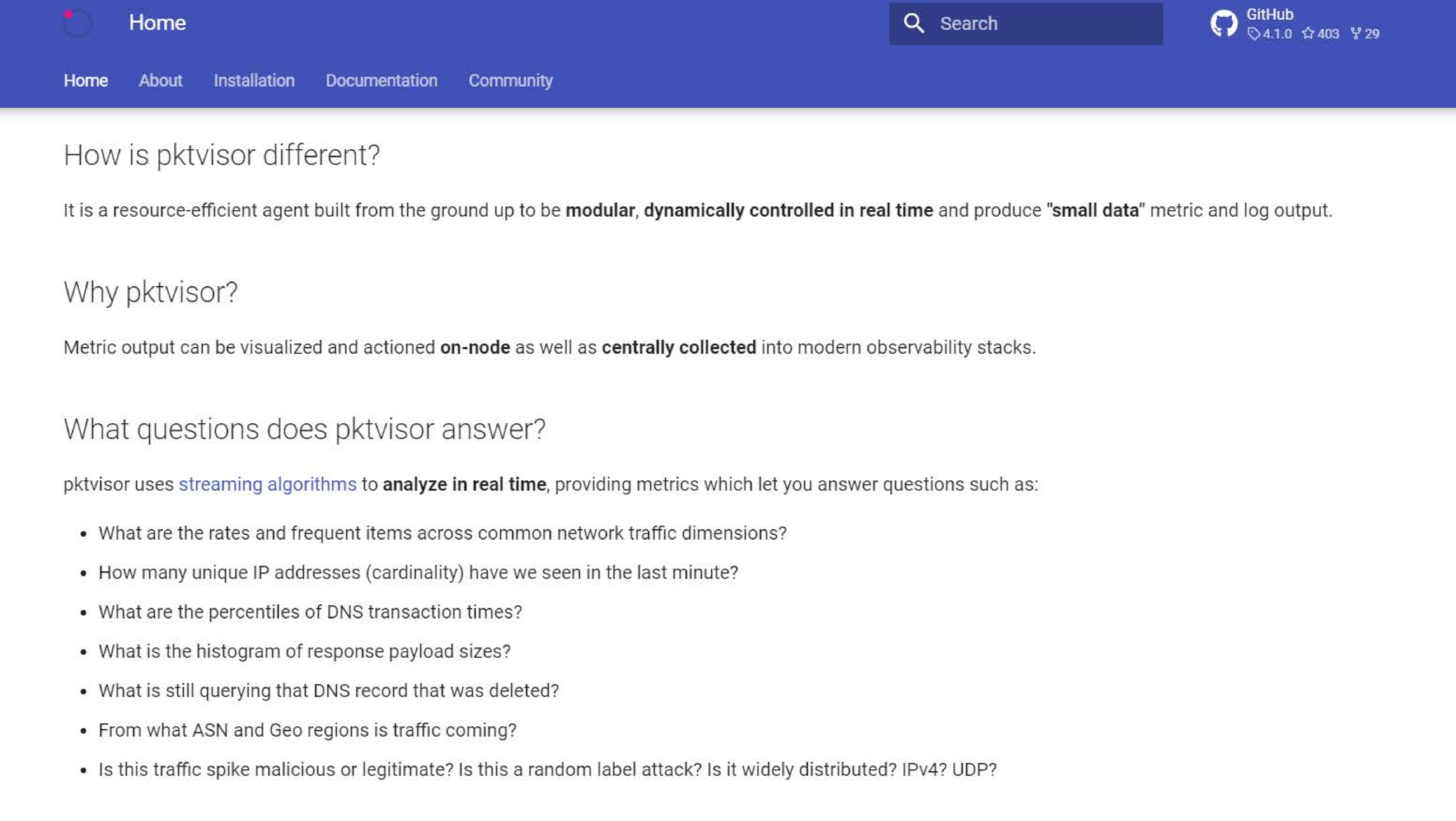
Open source, network observability agent for analysis at the edge

Get Started with pktvisor



What is pktvisor?

pktvisor (pronounced "packet visor") is an **observability agent** for analyzing high volume, information dense network data streams and extracting actionable insights **directly from the edge**.



How is pktvisor different?

It is a resource-efficient agent built from the ground up to be **modular**, **dynamically controlled in real time** and produce "**small data**" metric and log output.

Why pktvisor?

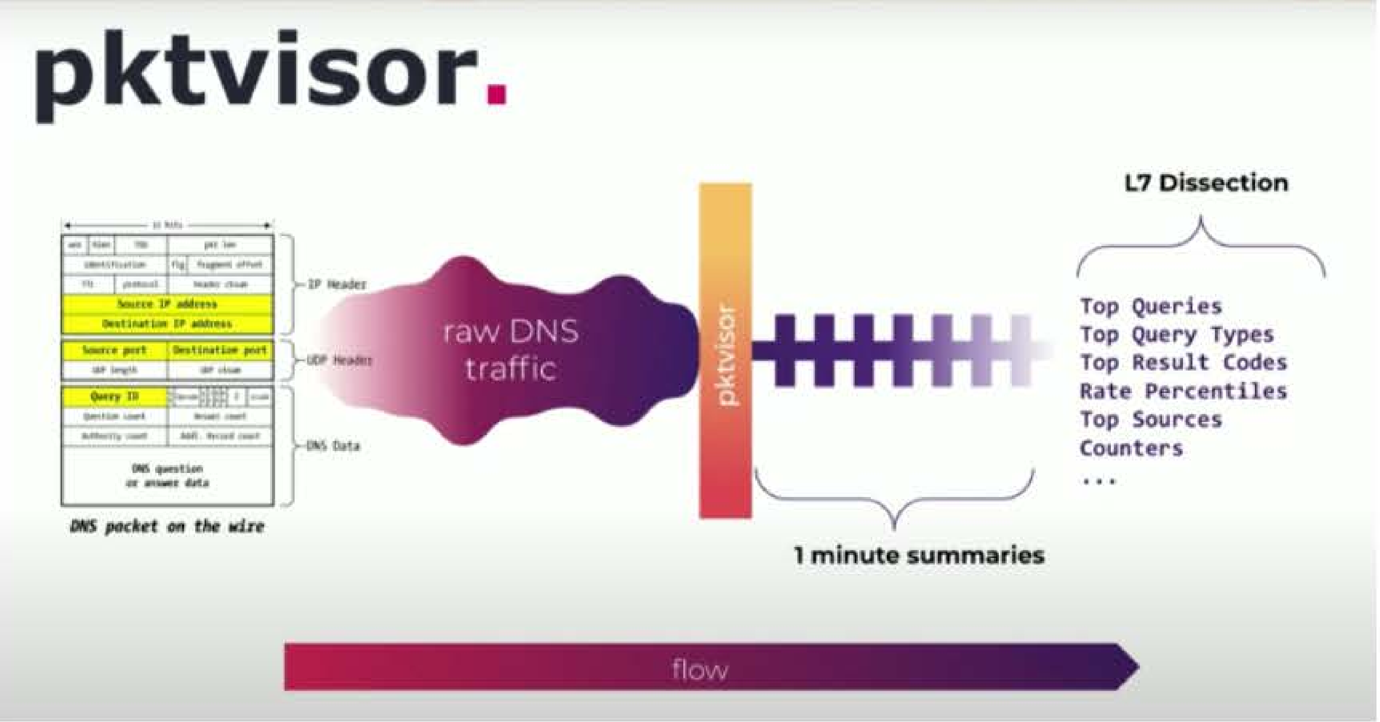
Metric output can be visualized and actioned **on-node** as well as **centrally collected** into modern observability stacks.

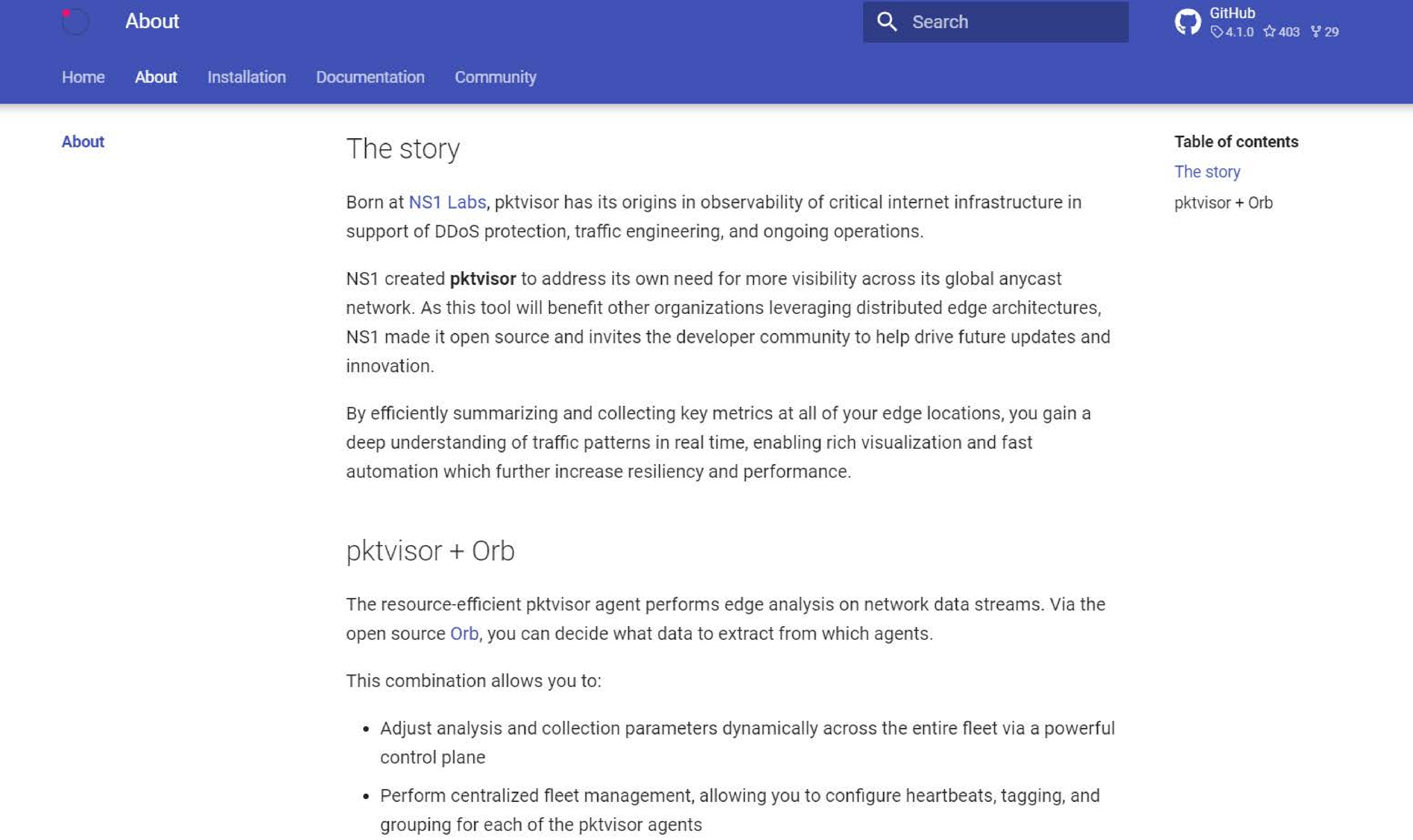
What questions does pktvisor answer?

pktvisor uses streaming algorithms to **analyze in real time**, providing metrics which let you answer questions such as:

- What are the rates and frequent items across common network traffic dimensions?
- How many unique IP addresses (cardinality) have we seen in the last minute?
- What are the percentiles of DNS transaction times?
- What is the histogram of response payload sizes?
- What is still querying that DNS record that was deleted?
- From what ASN and Geo regions is traffic coming?
- Is this traffic spike malicious or legitimate? Is this a random label attack? Is it widely distributed? IPv4? UDP?

About


pktvisor.
32 bits
ver
hlen
TOS
pkt len
identification
flg
fragment offset
TTL
protocol
header cksum
Source IP address
Destination IP address
IP Header
Source port
Destination port
UDP length
UDP cksum
UDP Header
Query ID
Question count
Answer count
Authority count
Addl. Record count
DNS question
or answer data
DNS Data
DNS packet on the wire
raw DNS
traffic
pktvisor
L7 Dissection
Top Queries
Top Query Types
Top Result Codes
Rate Percentiles
Top Sources
Counters
...
1 minute summaries
flow



The story

Born at NS1 Labs, pktvisor has its origins in observability of critical internet infrastructure in support of DDoS protection, traffic engineering, and ongoing operations.

NS1 created **pktvisor** to address its own need for more visibility across its global anycast network. As this tool will benefit other organizations leveraging distributed edge architectures, NS1 made it open source and invites the developer community to help drive future updates and innovation.

By efficiently summarizing and collecting key metrics at all of your edge locations, you gain a deep understanding of traffic patterns in real time, enabling rich visualization and fast automation which further increase resiliency and performance.

pktvisor + Orb

The resource-efficient pktvisor agent performs edge analysis on network data streams. Via the open source Orb, you can decide what data to extract from which agents.

This combination allows you to:

- Adjust analysis and collection parameters dynamically across the entire fleet via a powerful control plane
- Perform centralized fleet management, allowing you to configure heartbeats, tagging, and grouping for each of the pktvisor agents

Installation

Docker

Get started quickly with pktvisor via the public Docker image. The image contains the collector agent (`pktvisord`), the command-line UI (`pktvisor-cli`), and the pcap and dnstap file analyzer (`pktvisor-reader`). You will specify which tool to operate when running the container.

- *Pull the container*

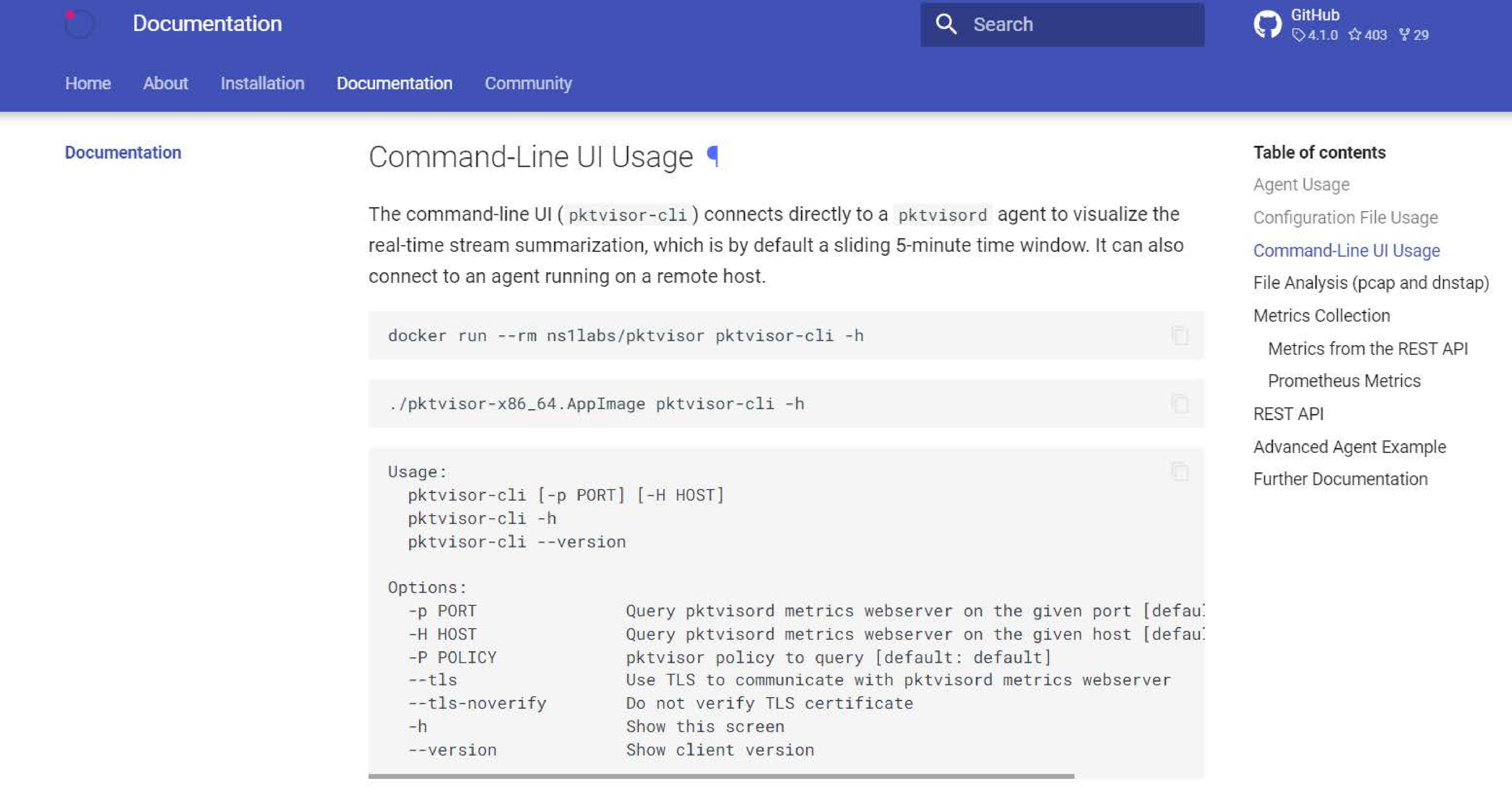
  ```
  docker pull ns1labs/pktvisor
  ```

Or use `ns1labs/pktvisor:latest-develop` to get the latest development version.

- *Start the collector agent*

  ```
  docker run --net=host -d ns1labs/pktvisor pktvisord eth0
  ```

This will start in the background and stay running. Note that the final two arguments select `pktvisord` agent and the `eth0` ethernet interface for packet capture. You may substitute `eth0` for any known interface on your device. *Note that this step requires Docker host networking* to observe traffic outside the container, and that currently only Linux supports host networking.



Command-Line UI Usage

The command-line UI (`pktvisor-cli`) connects directly to a `pktvisord` agent to visualize the real-time stream summarization, which is by default a sliding 5-minute time window. It can also connect to an agent running on a remote host.

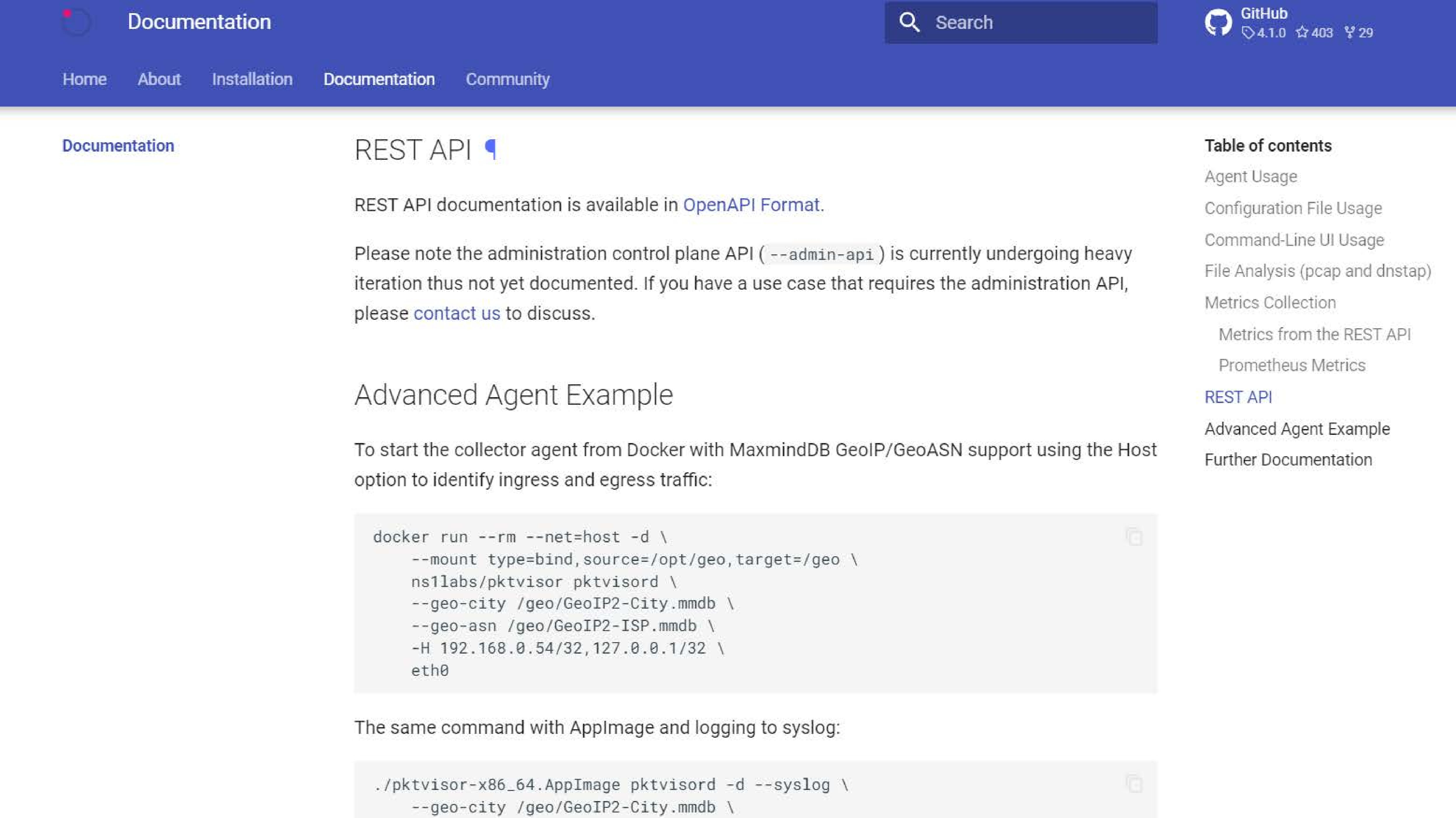
```
docker run --rm ns1labs/pktvisor pktvisor-cli -h
```

```
./pktvisor-x86_64.AppImage pktvisor-cli -h
```

```
Usage:
  pktvisor-cli [-p PORT] [-H HOST]
  pktvisor-cli -h
  pktvisor-cli --version

Options:
  -p PORT           Query pktvisord metrics webserver on the given port [defau
  -H HOST           Query pktvisord metrics webserver on the given host [defau
  -P POLICY         pktvisor policy to query [default: default]
  --tls             Use TLS to communicate with pktvisord metrics webserver
  --tls-noverify    Do not verify TLS certificate
  -h                Show this screen
  --version         Show client version
```

Table of contents



REST API ¶

REST API documentation is available in OpenAPI Format.

Please note the administration control plane API (`--admin-api`) is currently undergoing heavy iteration thus not yet documented. If you have a use case that requires the administration API, please contact us to discuss.

Advanced Agent Example

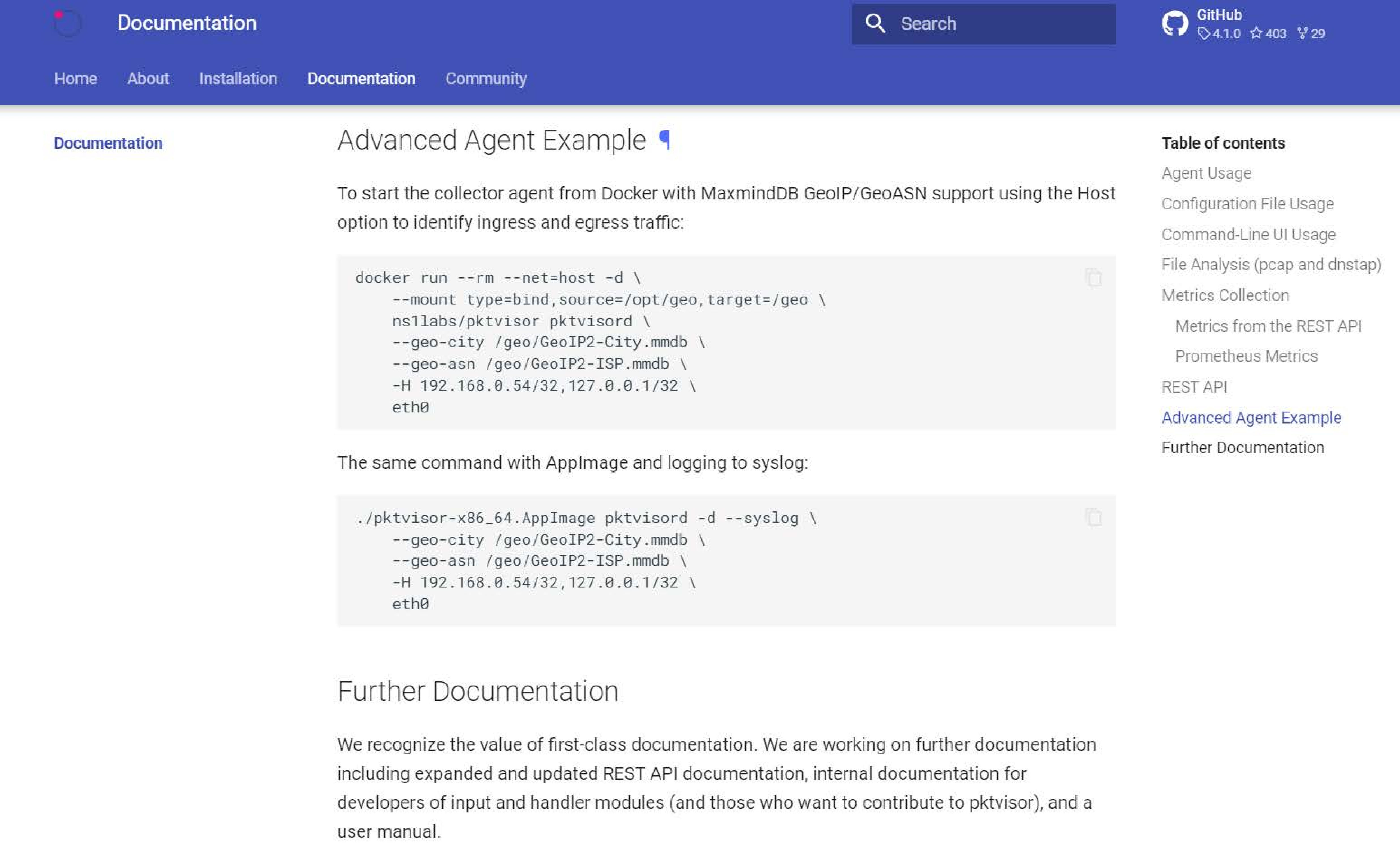
To start the collector agent from Docker with MaxmindDB GeoIP/GeoASN support using the Host option to identify ingress and egress traffic:

```
docker run --rm --net=host -d \
    --mount type=bind,source=/opt/geo,target=/geo \
    ns1labs/pktvisor pktvisord \
    --geo-city /geo/GeoIP2-City.mmdb \
    --geo-asn /geo/GeoIP2-ISP.mmdb \
    -H 192.168.0.54/32,127.0.0.1/32 \
    eth0
```

The same command with AppImage and logging to syslog:

```
./pktvisor-x86_64.AppImage pktvisord -d --syslog \
    --geo-city /geo/GeoIP2-City.mmdb \
```

Table of contents



Advanced Agent Example ¶

To start the collector agent from Docker with MaxmindDB GeoIP/GeoASN support using the Host option to identify ingress and egress traffic:

```
docker run --rm --net=host -d \
    --mount type=bind,source=/opt/geo,target=/geo \
    ns1labs/pktvisor pktvisord \
    --geo-city /geo/GeoIP2-City.mmdb \
    --geo-asn /geo/GeoIP2-ISP.mmdb \
    -H 192.168.0.54/32,127.0.0.1/32 \
    eth0
```

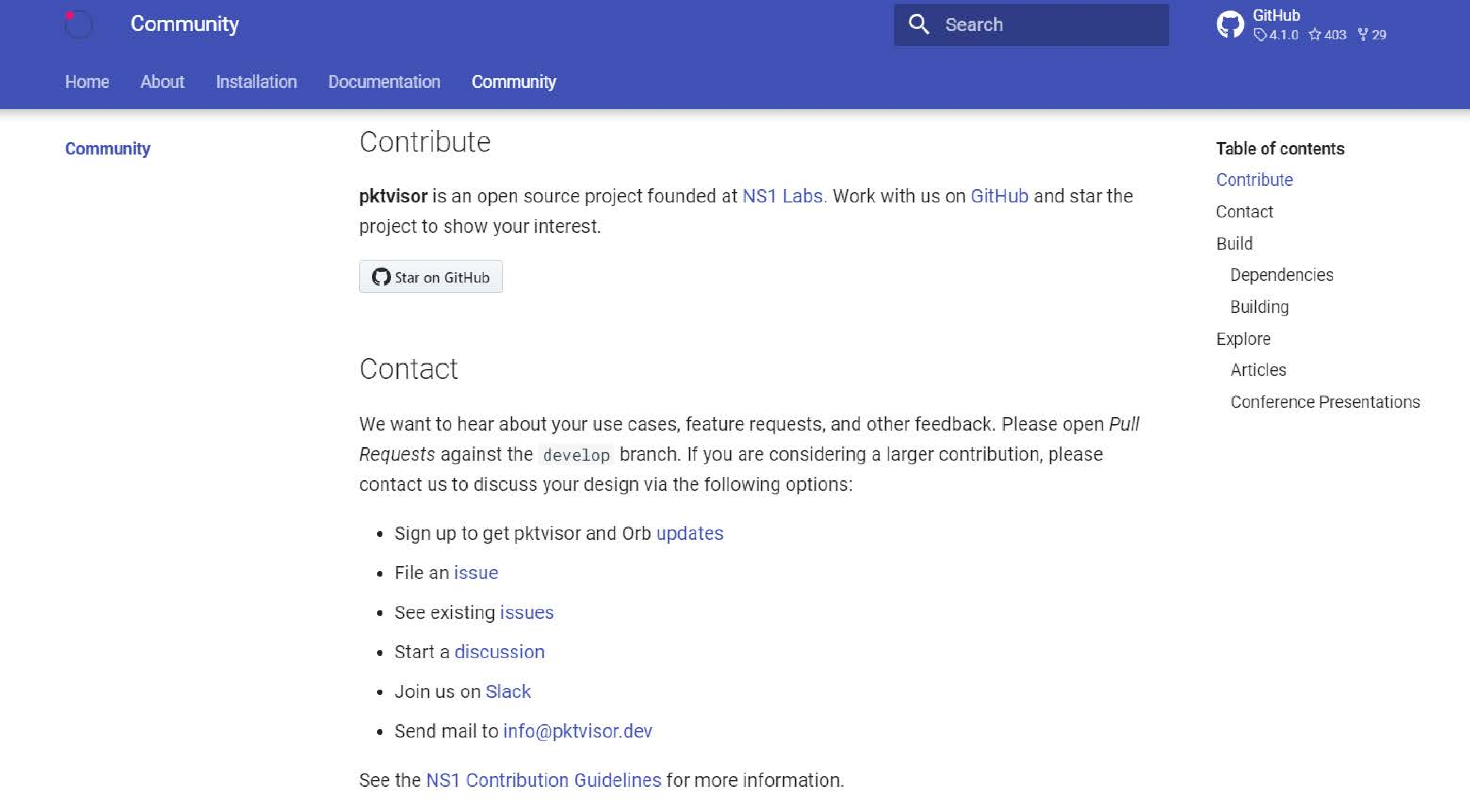
The same command with AppImage and logging to syslog:

```
./pktvisor-x86_64.AppImage pktvisord -d --syslog \
    --geo-city /geo/GeoIP2-City.mmdb \
    --geo-asn /geo/GeoIP2-ISP.mmdb \
    -H 192.168.0.54/32,127.0.0.1/32 \
    eth0
```

Further Documentation

We recognize the value of first-class documentation. We are working on further documentation including expanded and updated REST API documentation, internal documentation for developers of input and handler modules (and those who want to contribute to pktvisor), and a user manual.

Table of contents



Contribute

pktvisor is an open source project founded at NS1 Labs. Work with us on GitHub and star the project to show your interest.

Contact

We want to hear about your use cases, feature requests, and other feedback. Please open *Pull Requests* against the `develop` branch. If you are considering a larger contribution, please contact us to discuss your design via the following options:

- Sign up to get pktvisor and Orb updates
- File an issue
- See existing issues
- Start a discussion
- Join us on Slack
- Send mail to info@pktvisor.dev

See the NS1 Contribution Guidelines for more information.

Table of contents