

Streaming API Guide

Published
2021-07-19

RELEASE

Table of Contents

[Introduction](#)

[Configuration](#)

[Using the Streaming API](#)

Introduction

This guide describes how to extract data from Paragon Active Assurance via the product's streaming API.

The API as well as the streaming client are included in the Paragon Active Assurance installation. However, a bit of configuration is needed before you can use the API. This is covered in the ["Configuration" on page 1](#) chapter.

Configuration

IN THIS SECTION

- [Basic Configuration | 1](#)
- [Security Recommendations: Restricting Access to the Kafka Instance | 2](#)

Basic Configuration

The streaming API uses Kafka to transfer metrics from Paragon Active Assurance. You therefore need to do the following to get the streaming API up and running:

- Add the following to `/etc/netrounds/netrounds.conf`:

```
KAFKA_METRICS_ENABLED = True
```

- In `/etc/netrounds/metrics.yaml` set

```
streaming-api: true
```

- Run

```
sudo systemctl stop kafka
```

- In `/etc/kafka/server.properties` replace `localhost:9092` with `0.0.0.0:9092` for the listeners and add

```
advertised.listeners=PLAINTEXT://ip_or_url:9092
```

Here, `ip_or_url` should be set to the address that the clients will use. It is important that `ip_or_url` is routable; setting `0.0.0.0` as for the listeners results in an error.

NOTE:

[Apache Kafka documentation](#)

Then run

```
sudo systemctl start kafka
```

- Enable the metrics and TimescaleDB services (skip the latter if you are not using TimescaleDB):

```
sudo systemctl enable netrounds-timescaledb
sudo systemctl enable netrounds-metrics
sudo systemctl start netrounds-timescaledb netrounds-metrics
```

- Finally, restart the services:

```
sudo systemctl restart netrounds* apache2
```

Security Recommendations: Restricting Access to the Kafka Instance

At present, there is no authorization built into the streaming API. However, you can restrict access to the API by configuring your firewall. Here is how to set up persistent iptables rules for allowing and disallowing access to Kafka:

Install the iptables-persistent package:

```
sudo apt-get install iptables-persistent
```

In order to permanently add rules for IPv4, open the file `/etc/iptables/rules.v4` and insert rules similar to the following:

```
*nat
:PREROUTING ACCEPT [1:52]
:INPUT ACCEPT [1:52]
-A INPUT -p tcp --dport 9092 -s IP_ADDRESS -j ACCEPT
-A INPUT -p tcp --dport 9092 -j DROP
:OUTPUT ACCEPT [1:60]
:POSTROUTING ACCEPT [1:60]
COMMIT
```

Here, we allow access to the Kafka port (9092) for a specific IP address, while blocking all other connections to that port.

Similarly, in order to add IPv6 rules, edit the file `/etc/iptables/rules.v6` according to the code below:

```
*filter
:INPUT ACCEPT [5856:1295124]
-A INPUT -p tcp --dport 9092 -s IPv6_ADDRESS -j ACCEPT
-A INPUT -p tcp --dport 9092 -j DROP
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [74127:5871992]
-A INPUT -p tcp -m tcp --dport 42595 -j ACCEPT
COMMIT
```

NOTE: Important: *Reboot* the machine.

Next, check if the instructions are applied by running the command

```
sudo iptables -L -v -n
```

Using the Streaming API

IN THIS SECTION

- General | 4
- Protobuf Schemas | 4
- Kafka Topic Names | 5
- Examples of Using the Streaming API | 5

General

The streaming API fetches both test and monitor data. It is not possible to single out one of these categories.

The streaming API does not fetch data from script-based tests (those represented by a rectangle instead of a jigsaw piece in the Control Center GUI).

Protobuf Schemas

Protocol buffers, or protobufs for short, are Google's language-neutral, platform-neutral, and extensible mechanism for serializing structured data. Code and documentation are found [here](#).

The streaming API uses protobufs ...

TODO include protobuf schema in docs; need to explain schema structure

Kafka Topic Names

The Kafka topic names for the streaming API are as follows, where %s is the short name of the Control Center account (indicated when creating the account):

```
const (  
    exporterName      = "kafka"  
    metadataTopicTpl = "paa.public.accounts.%s.metadata"  
    metricsTopicTpl  = "paa.public.accounts.%s.metrics"  
)
```

Examples of Using the Streaming API

The examples that follow are found in the directory `TODO` .

Setup

Build the Docker image from the directory `client-examples`:

```
docker build -t netrounds/client-examples .
```

Client Usage

To read all messages from the *metrics* topic using the Docker image just built, run the following:

```
docker run --rm -it -v $(pwd):/shared netrounds/client-examples python3 client.py --kafkaURLs  
localhost:9092 --topic 'netrounds.public.accounts.ACCOUNT_NAME.metrics'
```

To read all messages from the *metadata* topic using the Docker image, run:

```
docker run --rm -it -v $(pwd):/shared netrounds/client-examples python3 client.py --kafkaURLs  
localhost:9092 --topic 'netrounds.public.accounts.ACCOUNT_NAME.metadata'
```

Running Against Localhost

If Kafka runs on the local server, and you would like to fetch data from it, you need to use the `--network host` switch in the docker command:

```
docker run --rm -it -v $(pwd):/shared netrounds/client-examples --network host python3  
client.py --kafkaURLs localhost:9092 --topic 'netrounds.public.accounts.ACCOUNT_NAME.metadata'
```