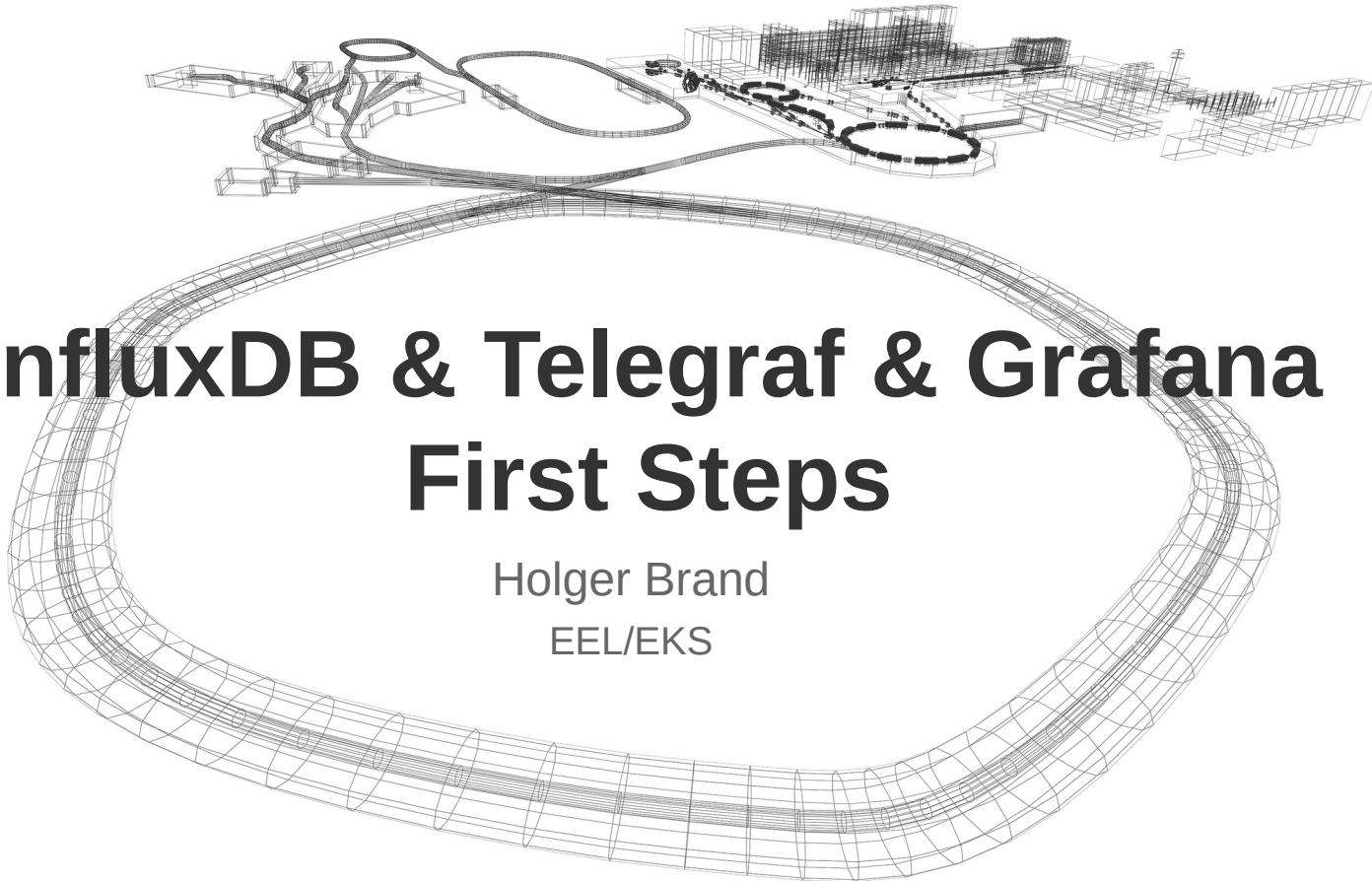


A complex wireframe illustration of a roller coaster track, featuring a large loop and various curves, rendered in a technical, skeletal style.

InfluxDB & Telegraf & Grafana First Steps

Holger Brand
EEL/EKS

■ InfluxDB

- Web-Interface
 - Bucket
 - Access Tokens
 - Data Explorer
- Python Client: Query Example
- Start InfluxDB using Docker

■ Telegraf

- Configuration with MQTT as data source
- Start Telegraf using Docker

■ Grafana

- Configuration with InfluxDB as data source
- Start Grafana using Docker

- *It's About Time. Build on InfluxDB.*
 - *InfluxDB is the time series platform.*
 - *The Time Series Data Platform where developers build IoT, analytics, and cloud applications.*
- *Open Source: <https://github.com/influxdata/influxdb>*
 - *influxdata/influxdb is licensed under the MIT License*
- *Web Interface*
- *Programming API's.*
 - *Arduino, C#, Dart, Go, Java, JavaScript for browsers*
 - *Kotlin, Node.js, PHP, Python, R, Ruby, Scala, Swift*

Permissions	Limitations	Conditions
✓ Commercial use	✗ Liability	④ License and copyright notice
✓ Modification	✗ Warranty	
✓ Distribution		
✓ Private use		

- Thanks for help.

- P. Zumbruch(EEL), D. Bruins (IT)

- Set Export environment variables

```
export influxdb_port=8086
export influxdb_version=2.4
export influxdb_suffix=01
export influxdb_dataDir=/home/$USER/Container/InfluxDB2/data/${influxdb_suffix:?}
export influxdb_configDir=/home/$USER/Container/InfluxDB2/config/${influxdb_suffix:?}
```

- Preparation

```
mkdir -p ${influxdb_dataDir:?} ${influxdb_configDir:?}
# Initial setup
sudo docker run -d --rm \
  --name "influxdb${influxdb_version:?}_${influxdb_suffix:?}" \
  -p ${influxdb_port:?}:8086 \
  -v ${influxdb_dataDir:?}:/var/lib/influxdb2 \
  -v ${influxdb_configDir:?}:/etc/influxdb2 \
  -e DOCKER_INFLUXDB_INIT_MODE=setup \
  -e DOCKER_INFLUXDB_INIT_USERNAME=username \
  -e DOCKER_INFLUXDB_INIT_PASSWORD=password \
  -e DOCKER_INFLUXDB_INIT_ORG=organization \
  -e DOCKER_INFLUXDB_INIT_BUCKET=Test \
  -e DOCKER_INFLUXDB_INIT_RETENTION=4w \
  influxdb:${influxdb_version} && \
  sleep 10 && \
  firefox --new-instance localhost:${influxdb_port}
```

- Run InfluxDB

```
sudo docker or better podman run -d --rm \
  --name "telegraf_${telegraf_suffix}" \
  --volume ${telegraf_configDir:?}/${1:-telegraf.conf}:/etc/telegraf/telegraf.conf:ro \
  telegraf
```

Web-Interface: Organization & Member



Browser tabs: About | Organization | GSI x

Address bar: localhost:8086/orgs/51510e3a167d98a0

Organization

MEMBERS ABOUT

Organization Profile

Name

RENAME

099ed7653f3e3000

COPY TO CLIPBOARD | User ID

51510e3a167d98a0

COPY TO CLIPBOARD | Organization ID

Browser tabs: Members | Organization | x

Address bar: localhost:8086

Organization

MEMBERS ABOUT

USERNAME	ROLE
	Role: owner

Bucket

- A bucket is a named location where time series data is stored.
- All buckets have a Retention Policy, a duration of time that each data point persists.

Edit Bucket

Name*

VHS

To rename bucket use the RENAME button below

Delete Data

NEVER

OLDER THAN

30 days

CANCEL

RENAME

SAVE CHANGES



Load Data

G

↑

↗

📄

📈

📅

🔔

⚙️

🖼️

Buckets

▼

🔍 Filter by

Sort by Name (A → Z) ▼

+ CREATE BUCKET

UHVHC

Retention: 14 days
ID: 6d41f59681e97eed

+ Add a label

+ ADD DATA

SETTINGS

VHS

Retention: 30 days
ID: 5df57c6368681c39

+ Add a label

+ ADD DATA

SETTINGS

What is a Bucket?

A bucket is a named location where time series data is stored. All buckets have a **Retention Policy**, a duration of time that each data point persists.

Here's [how to write data](#) into your bucket.

API Access Tokens



API Tokens | Load Data | +

localhost:8086/orgs/51510e3a167d98a0/load-data/tokens

Load Data

SOURCES BUCKETS TELEGRAF SCRAPERS **API TOKENS**

Our Tokens UI is changing on **Jan 10th, 2022**. After that day you'll only be able to view and safely store token details at the point of creation. If you lose access to token credentials you can generate a new token. Please copy any existing token credentials in your chosen safe location.

Filter API Tokens... Sort by Description (A → Z) + GENERATE API TOKEN

influx's Token Created at: 2022-07-05 00:41:27 ☒ Active	+ GENERATE API TOKEN Read/Write API Token All Access API Token
UHVHC Created at: 2022-07-15 09:29:10 ☒ Active	
VHS_RW Created at: 2023-04-06 19:56:46 ☒ Active	

Loading Data



Sources | Load Data | GSI x +

localhost:8086/orgs/51510e3a167d98a0/load-data/sources 67% ☆

Load Data

Client Libraries

Back-end, front-end, and mobile applications

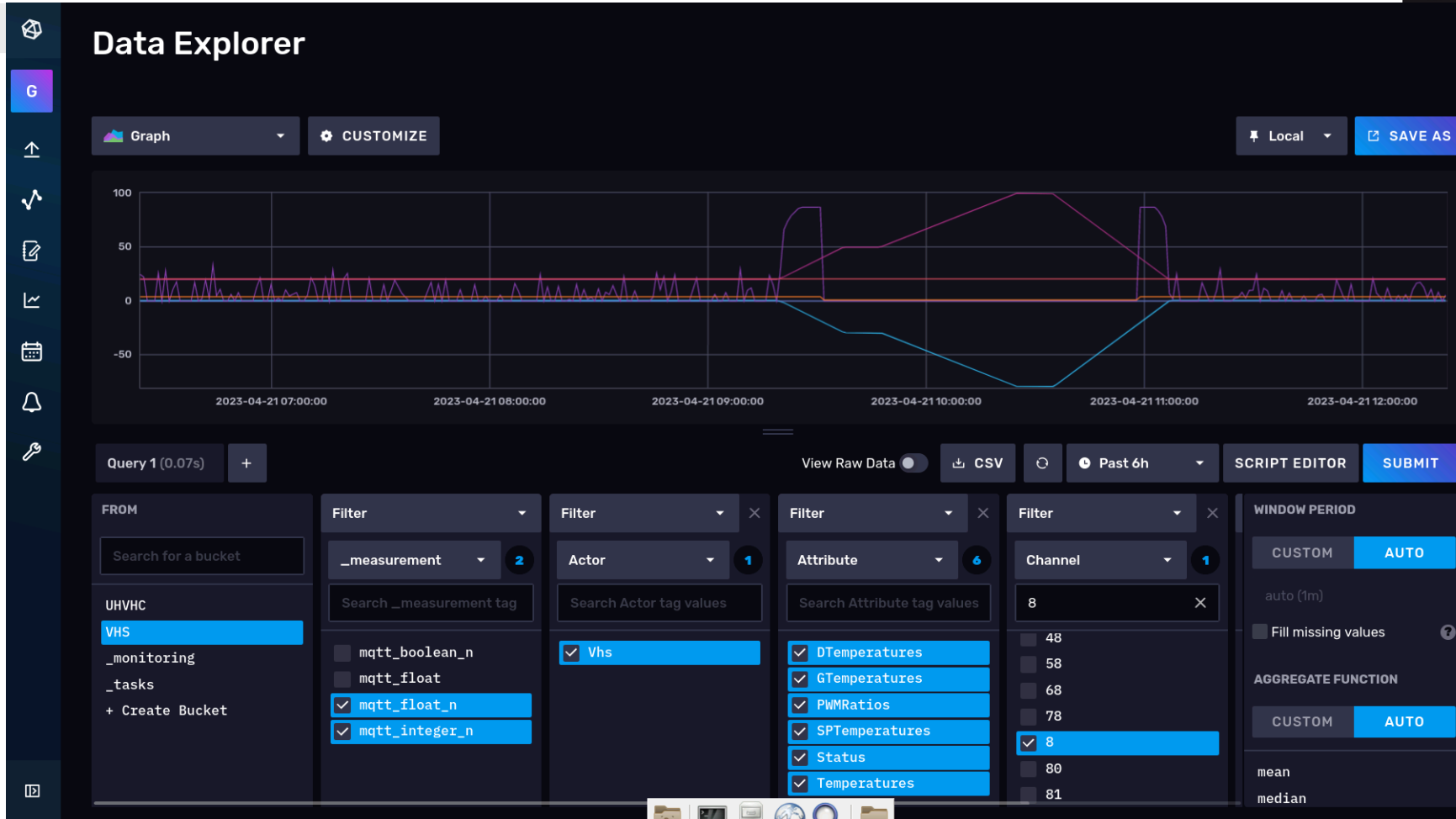
Arduino	C#	GO	Java	JavaScript/Node.js	Kotlin	PHP	Python
R	Ruby	Scala	Swift				

Telegraf Plugins

An open-source agent for collecting data and reporting metrics via a vast library of plugins

ActiveMQ	Aerospike	Alibaba (Aliyun) Cloud...	Amazon CloudWatch S...	Amazon ECS	AMQP Consumer	Apache	APCUPSD
Arista LANZ Consumer	Aurora	Azure Storage Queue	bcache	Beanstalkd	Beat	BIND 9 Nameserver St...	Bond

Data Explorer



```
from(bucket: "VHS")  
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)  
  |> filter(fn: (r) => r["_measurement"] == "mqtt_float_n" or r["_measurement"] ==  
"mqtt_integer_n")  
  |> filter(fn: (r) => r["Actor"] == "Vhs")  
  |> filter(fn: (r) => r["Attribute"] == "DTemperatures" or r["Attribute"] ==  
"GTemperatures" or r["Attribute"] == "PWMRatios" or r["Attribute"] ==  
"SPTemperatures" or r["Attribute"] == "Status" or r["Attribute"] == "Temperatures")  
  |> filter(fn: (r) => r["Channel"] == "8")  
  |> aggregateWindow(every: v.windowPeriod, fn: last, createEmpty: false)  
  |> yield(name: "last")
```

Python Query using influx-client package

Spyder (Python 3.10)

File Edit Search Source Run Debug Consoles Projects Tools View Help

/home/brand/Python/vhs_query_influxdb.py

```
1 from influxdb_client import InfluxDBClient
2 import matplotlib.pyplot as plt
3 import time
4
5 url =
6 token =
7 org =
8 bucket = 'VHS'
9 t0 = time.time()
10 time_span = '-2h'
11 aggregate_window = None
12 # aggregate_window = '30s'
13 actor = 'Vhs'
14 channel = 8
15 attributes = ('Temperatures', 'DTemperatures', 'GTemperatures', 'SPTemperatures', 'PWMRatios')
16 queries = []
17 for attribute in attributes:
18     query = 'from(bucket: "VHS") |> range(start: ' + time_span + ') |> filter(fn:(r) => r["Ac
19     if aggregate_window:
20         query += ' |> aggregateWindow(every: ' + aggregate_window + ', fn: last, createEmpty:
21     queries.append(query)
22 if True:
23     with InfluxDBClient(url=url, token=token, org=org) as client:
24         plt.title('Historical data for ' + actor + ' channel: ' + str(channel))
25         plt.xlabel('date time')
26         plt.ylabel('value')
27         query_api = client.query_api()
28         for ii, query in enumerate(queries):
29             # print('Query:', query)
30             result = query_api.query(org=org, query=query)
31             results = []
32             values = []
33             times = []
34             for table in result:
35                 for record in table.records:
36                     try:
37                         # print('record:\n', record)
38                         # print('type(record["time"])', type(record["time"]))
39                         results.append((str(record["time"]), record.get_value()))
40                         times.append(record["time"])
41                         values.append(record.get_value())
42                     except Exception:
43                         pass
44             # print(results)
45             plt.plot(times, values, label=attributes[ii][:4])
46             plt.legend(loc='best')
47             plt.show()
```

Historical data for Vhs channel: 8

Help Variable Explorer Plots Files

Console 1/A X

type "copyright", "credits" or "license" for more information.

IPython 7.33.0 -- An enhanced Interactive Python.

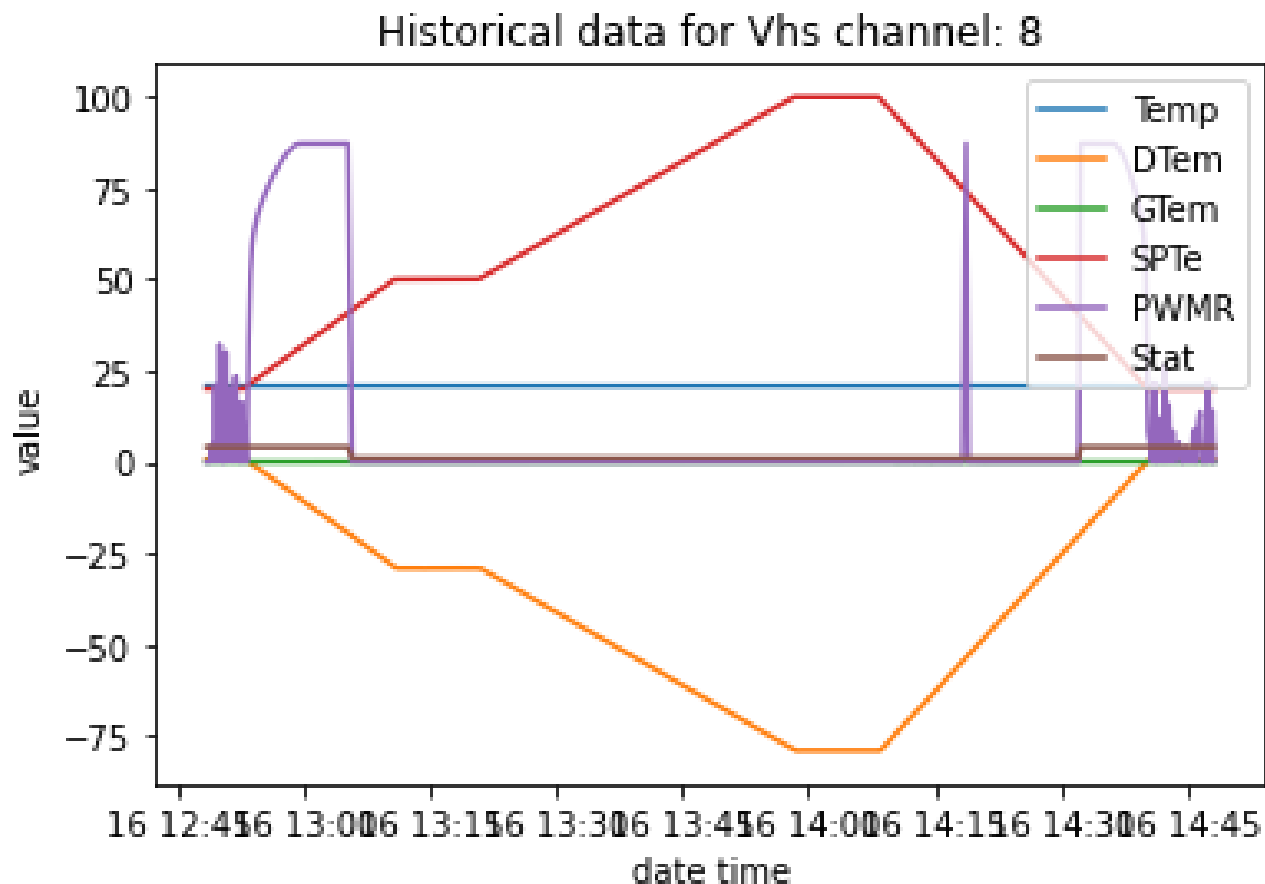
Restarting kernel...

```
In [1]: runfile('/home/brand/Python/vhs_query_influxdb.py', wdir='/home/brand/Python')
Iteration time: 2.1764369010925293

In [2]: runfile('/home/brand/Python/vhs_query_influxdb.py', wdir='/home/brand/Python')
Iteration time: 1.8130607604980469

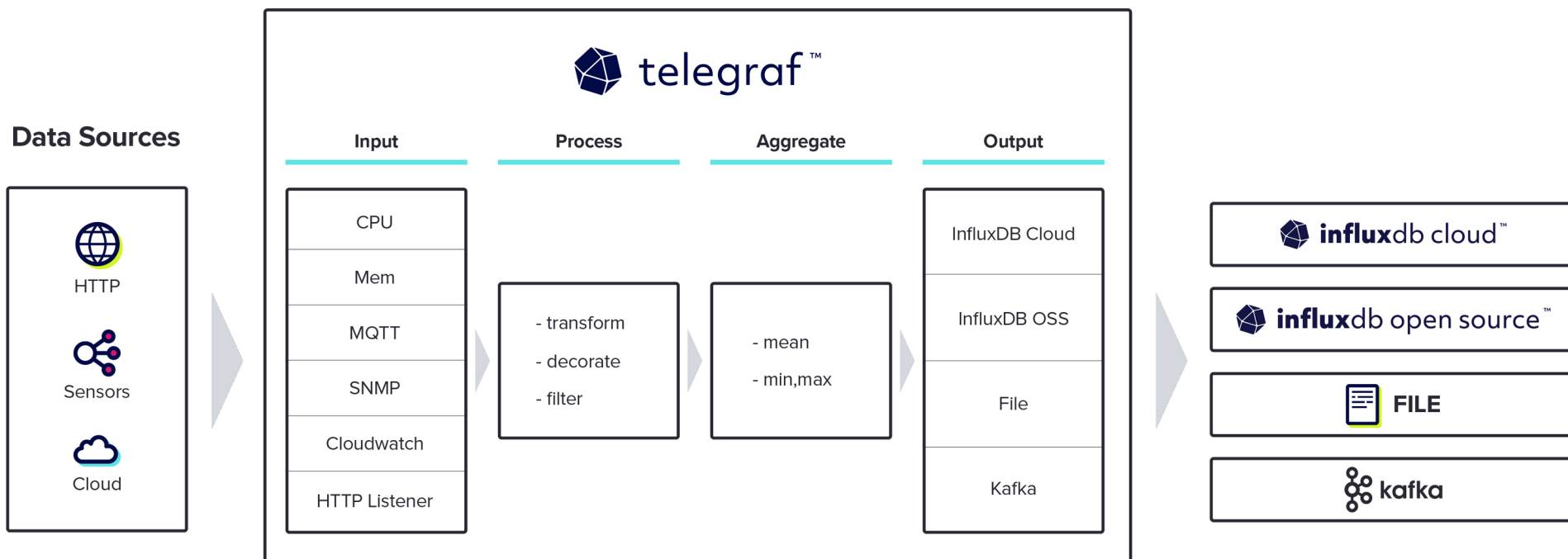
In [3]:
```

Python Query resulting plot



Telegraf – Overview

<https://www.influxdata.com/time-series-platform/telegraf/>



■ Input

- Telegraf Input Plugins collect metrics from the system, services, and 3rd party APIs.
- MQTT: <https://www.influxdata.com/integration/mqtt-telegraf-consumer/>
- OPC-UA: <https://www.influxdata.com/integration/opcua/>

■ Output

- Processor Plugins transform, decorate, and filter metrics before they are sent allowing your data to be sanitized as it arrives.
- MQTT: <https://www.influxdata.com/integration/mqtt-telegraf-producer/>

■ Processor

- Aggregator Plugins create aggregate metrics, for example the average mean, minimum, and maximum from the metrics you have collected, and processed.

■ Aggregator

- Output Plugins write to a variety of datastores, services, & message queues, like InfluxDB, Graphite, OpenTSDB, Datadog, Kafka, MQTT, NSQ, and others.

■ InfluxDB Output-Plugin

- `[[outputs.influxdb_v2]]`
 - `urls = ["http://host:8086"]`
 - `token = "token_string"`
 - `organization = "organisation"`
 - `bucket = "VHS"`

■ MQTT Input-Plugin

- `[[inputs.mqtt_consumer]]`
 - `name_override = "mqtt_float_n"`
 - `servers = ["tcp://broker:1883"]`
 - `topics = [`
 - `"+/Vhs/Temperatures/+",`
 - `"+/Vhs/DTemperatures/+",`
 - `"+/Vhs/GTemperatures/+",`
 - `"+/Vhs/SPTemperatures/+",`
 - `"+/Vhs/PWMRatios/+",` - `data_format = "value"`
 - `data_type = "float"`
 - `[[inputs.mqtt_consumer.topic_parsing]]`
 - `topic = "+/+ / +/+"`
 - `measurement = "_/_/_/measurement"`
 - `tags = "Process/Actor/Attribute/Channel"`
 - `#fields = "_/_/_"`

- Thanks to P. Zumbruch for help.

- Set Export environment variables

```
export telegraf_suffix=01
```

```
export telegraf_configDir=/home/$USER/Container/Telegraf/config/${telegraf_suffix:?}
```

```
export telegraf_dataDir=/home/$USER/Container/Telegraf/data/${telegraf_suffix:?}
```

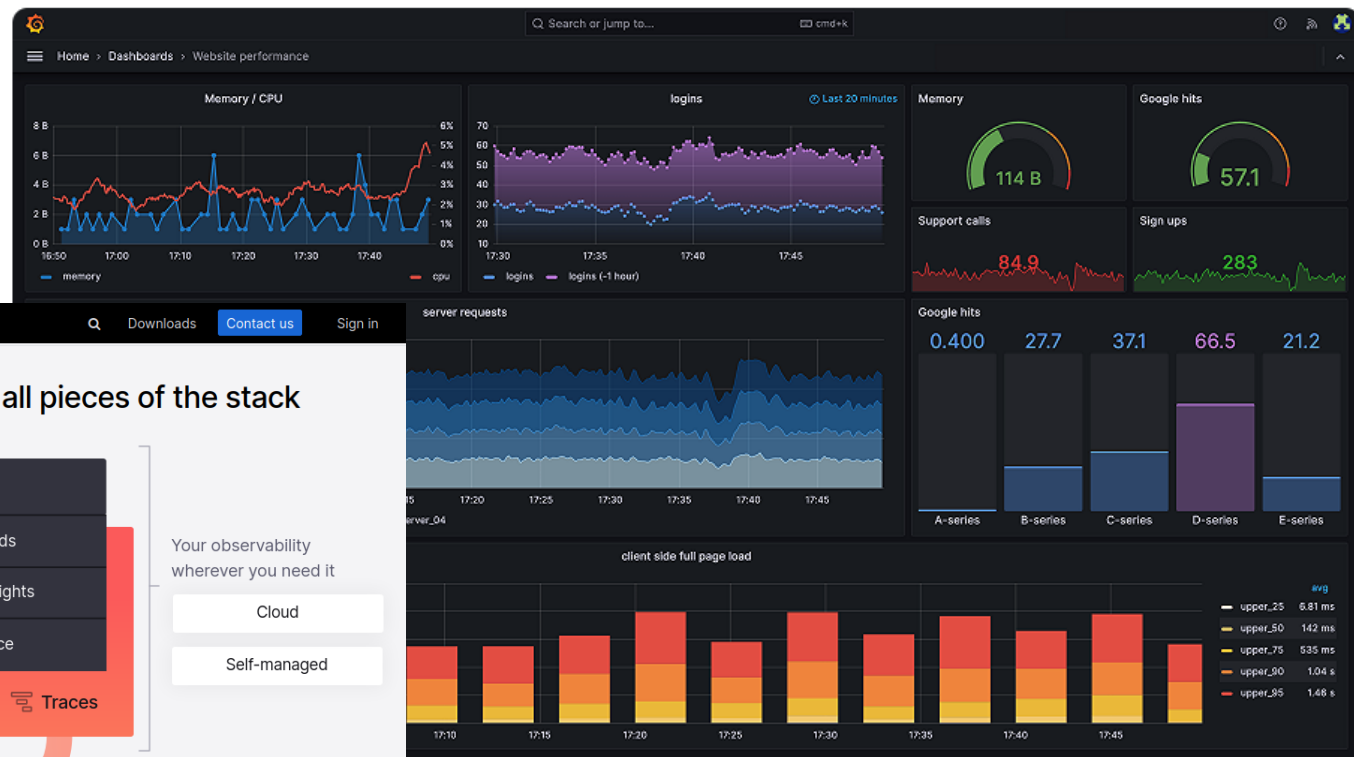
- Run telegraf

```
sudo docker or better podman run -d --rm \
```

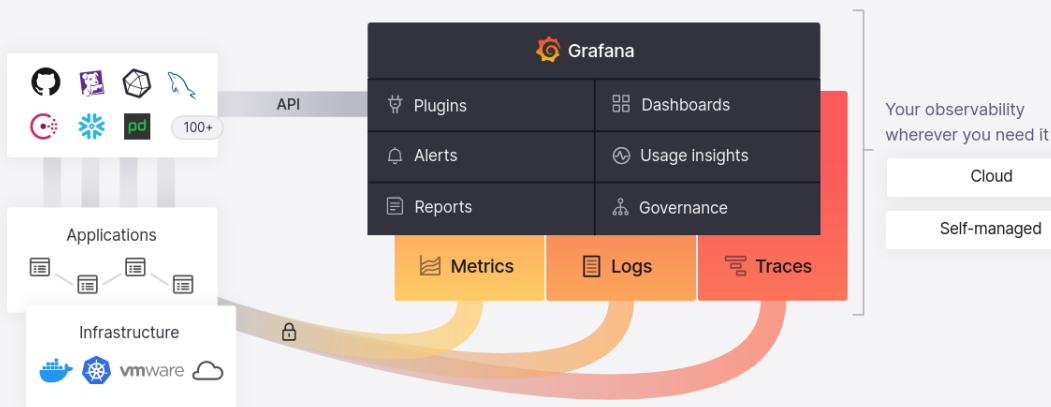
```
--name "telegraf_${telegraf_suffix}" \
```

```
--volume ${telegraf_configDir:?}/${1:-telegraf.conf}:/etc/telegraf/telegraf.conf:ro \ telegraf
```

- Web based Visualization
- Homepage
 - <https://grafana.com>
- Open Source
 - <https://github.com/grafana>



Compose and scale observability with one or all pieces of the stack



- Dashboard anything. Observe everything.
 - Query, visualize, alert on, and understand your data no matter where it's stored.
 - With Grafana you can create, explore, and share all of your data through beautiful, flexible dashboards.
- Panels
 - From heatmaps to histograms. Graphs to geomaps. Grafana has fast and flexible visualizations that allows you to visualize your data, any way you want.
- Plugins
 - Connect your tools and your teams with Grafana plugins. Data source plugins hook into existing data sources via APIs and render the data in real time without requiring you to migrate or ingest your data.
 - e.g. InfluxDB, PostgreSQL, MySQL, SQLite, MS SQL Server, CSV, MQTT, OPC UA, Oracle Database, WebSocket API

- Thanks for help

- Christian Jacobsen and Jörn Plewka, HZ Hereon.
- P. Zumbbruch

- Set Export environment variables

```
export grafana_suffix=01
export grafana_configDir=/home/$USER/Container/Grafana/config/${grafana_suffix:?}
export grafana_dataDir=/home/$USER/Container/Grafana/data/${grafana_suffix:?}
```

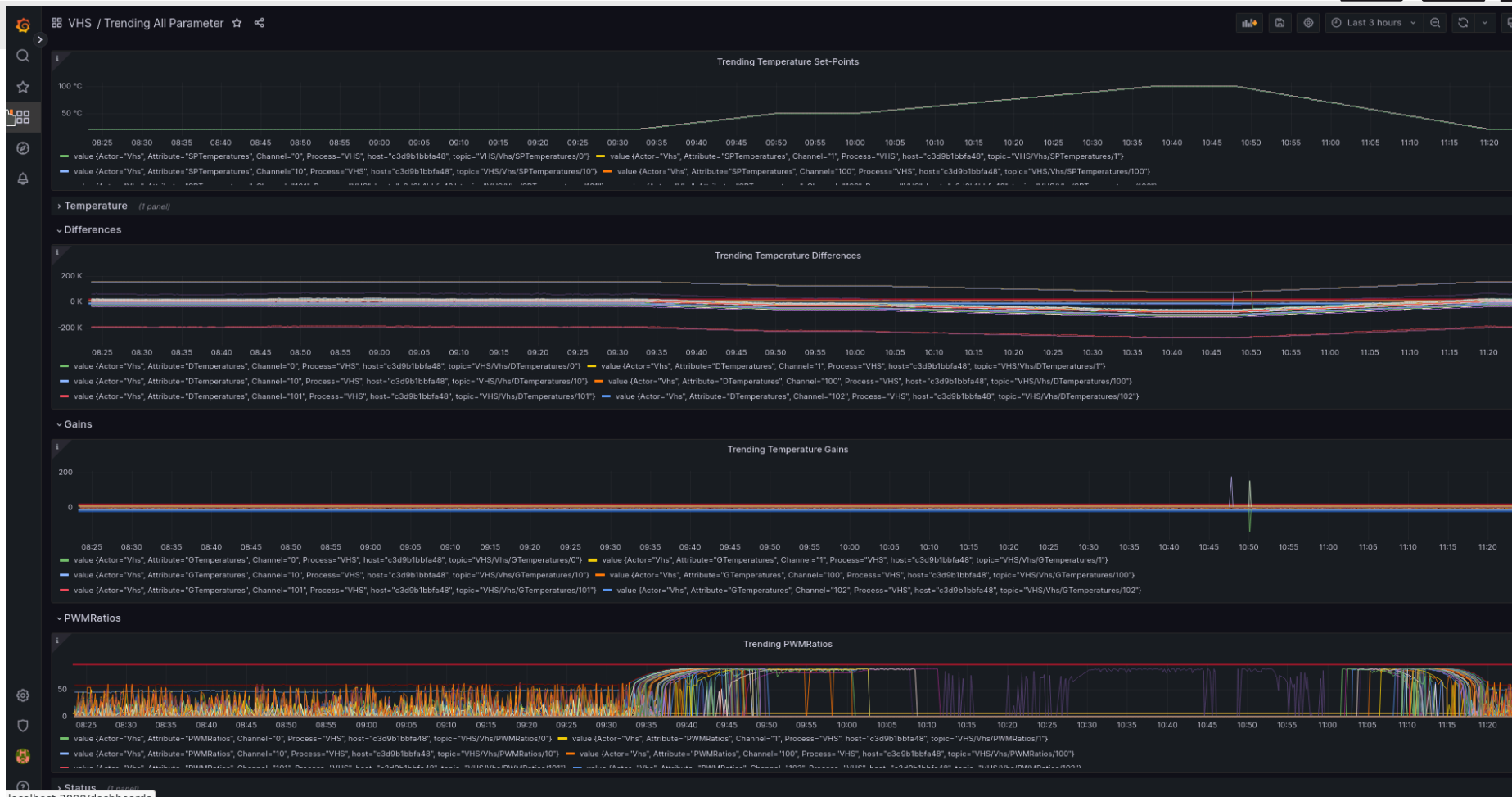
- Preparation

```
source /home/$USER/Container/Grafana/GrafanaEnv.sh
mkdir -p ${grafana_dataDir:?} ${grafana_configDir:?}
```

- Run Grafana

```
source /home/$USER/Container/Grafana/GrafanaEnv.sh
sudo docker or better podman run -d --rm \
--name "grafana_${grafana_suffix}" \
--volume ${grafana_configDir:?}:/var/lib/grafana \
-p 3000:3000 \
--user 0 \
grafana/grafana:latest
```

First Trial: VHS Dashboards



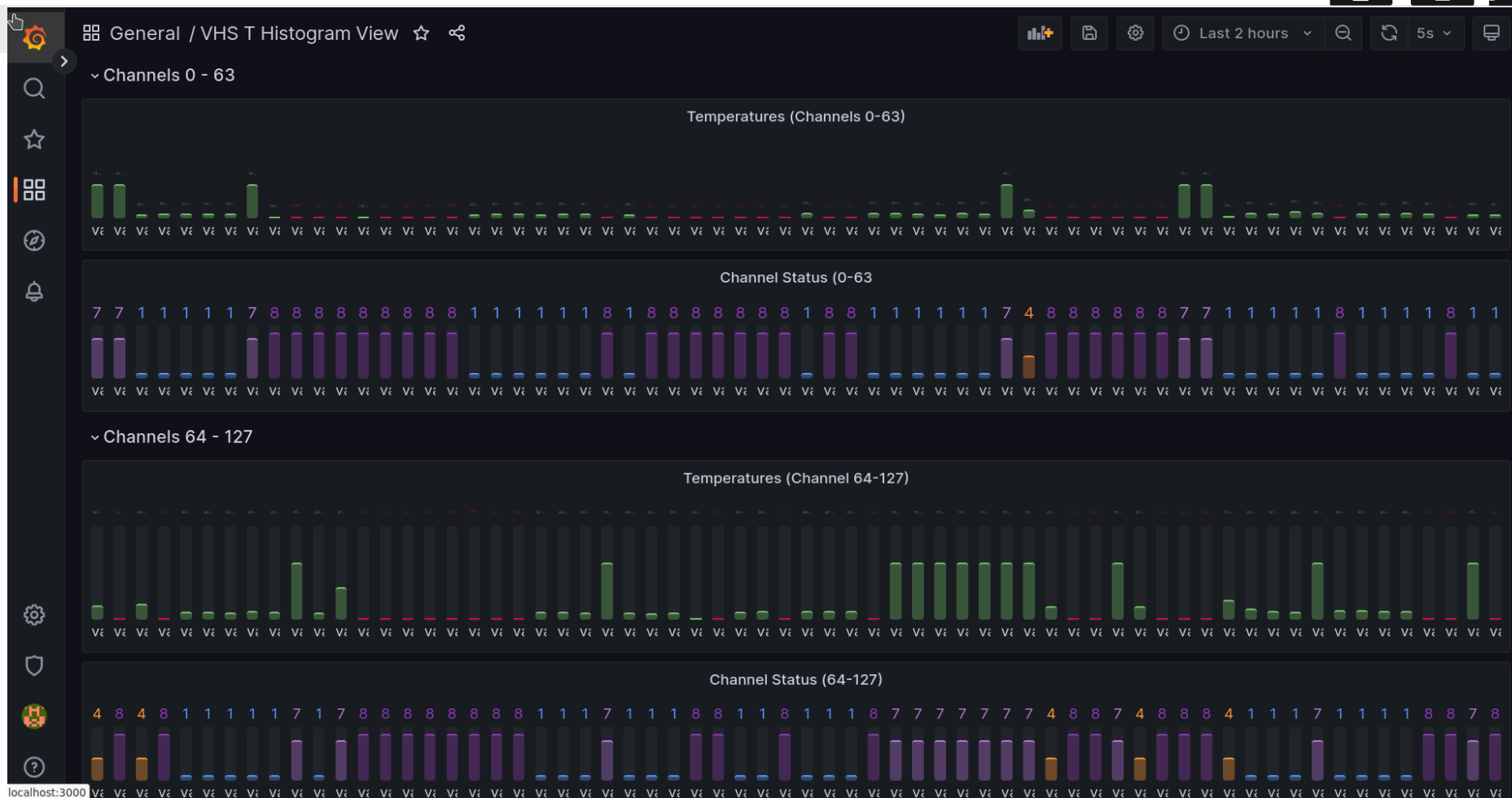
First Trial: VHS Dashboards



First Trial: VHS Dashboards



First Trial: VHS Dashboards



First Trial: VHS Dashboards



VHS-Alerts / Alert Summary

VHS-Alerts

Over-Heated
Firing for 4d 8m 38s
11 instances, 130 hidden by filters

State	Labels	Created
Pending	Actor=Vhs Attribute=DTemperatures Channel=0 Process=VHS alertname=Over-Heated grafana_folder=VHS-Alerts host=c3d9b1bbfa48 topic=VHS/Vhs/DTemperatures/0	2023-04-25 11:15:00
Alerting	Actor=Vhs Attribute=DTemperatures Channel=1 Process=VHS alertname=Over-Heated grafana_folder=VHS-Alerts host=c3d9b1bbfa48 topic=VHS/Vhs/DTemperatures/1	2023-04-21 11:10:00
Pending	Actor=Vhs Attribute=DTemperatures Channel=102 Process=VHS alertname=Over-Heated grafana_folder=VHS-Alerts host=c3d9b1bbfa48 topic=VHS/Vhs/DTemperatures/102	2023-04-25 11:16:00
Alerting	Actor=Vhs Attribute=DTemperatures Channel=109 Process=VHS alertname=Over-Heated grafana_folder=VHS-Alerts host=c3d9b1bbfa48 topic=VHS/Vhs/DTemperatures/109	2023-04-21 11:10:00
Alerting	Actor=Vhs Attribute=DTemperatures Channel=111 Process=VHS alertname=Over-Heated grafana_folder=VHS-Alerts host=c3d9b1bbfa48 topic=VHS/Vhs/DTemperatures/111	2023-04-25 11:06:00
Alerting	Actor=Vhs Attribute=DTemperatures Channel=123 Process=VHS alertname=Over-Heated grafana_folder=VHS-Alerts host=c3d9b1bbfa48 topic=VHS/Vhs/DTemperatures/123	2023-04-21 11:10:00
Alerting	Actor=Vhs Attribute=DTemperatures Channel=15 Process=VHS alertname=Over-Heated grafana_folder=VHS-Alerts host=c3d9b1bbfa48 topic=VHS/Vhs/DTemperatures/15	2023-04-21 11:10:00
Alerting	Actor=Vhs Attribute=DTemperatures Channel=17 Process=VHS alertname=Over-Heated grafana_folder=VHS-Alerts host=c3d9b1bbfa48 topic=VHS/Vhs/DTemperatures/17	2023-04-25 10:53:00
Alerting	Actor=Vhs Attribute=DTemperatures Channel=46 Process=VHS alertname=Over-Heated grafana_folder=VHS-Alerts host=c3d9b1bbfa48 topic=VHS/Vhs/DTemperatures/46	2023-04-21 11:10:00
Pending	Actor=Vhs Attribute=DTemperatures Channel=47 Process=VHS alertname=Over-Heated grafana_folder=VHS-Alerts host=c3d9b1bbfa48 topic=VHS/Vhs/DTemperatures/47	2023-04-25 11:15:00
Alerting	Actor=Vhs Attribute=DTemperatures Channel=8 Process=VHS alertname=Over-Heated grafana_folder=VHS-Alerts host=c3d9b1bbfa48 topic=VHS/Vhs/DTemperatures/8	2023-04-25 10:56:00