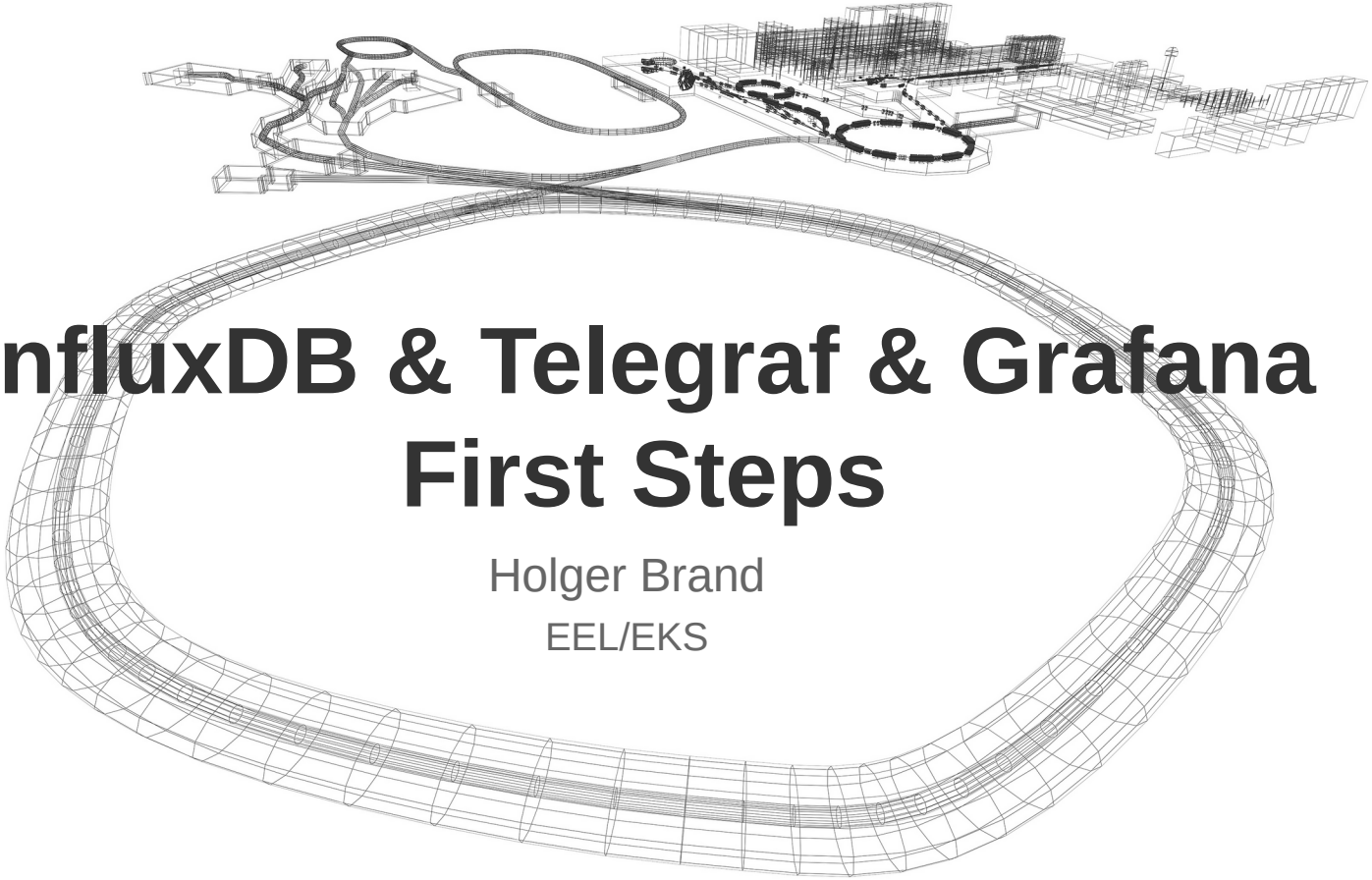


InfluxDB & Telegraf & Grafana
First Steps

Holger Brand
EEL/EKS

CC BY NC SA

Agenda

- 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

InfluxDB

- *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*

Permissions	Limitations	Conditions
✓ Commercial use	✕ Liability	ⓘ License and copyright notice
✓ Modification	✕ Warranty	
✓ Distribution		
✓ Private use		

- *Web Interface*
- *Programming API's.*
  - *Arduino, C#, Dart, Go, Java, JavaScript for browsers*
  - *Kotlin, Node.js, PHP, Python, R, Ruby, Scala, Swift*

InfluxDB – Start in Docker

- 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:?}
```

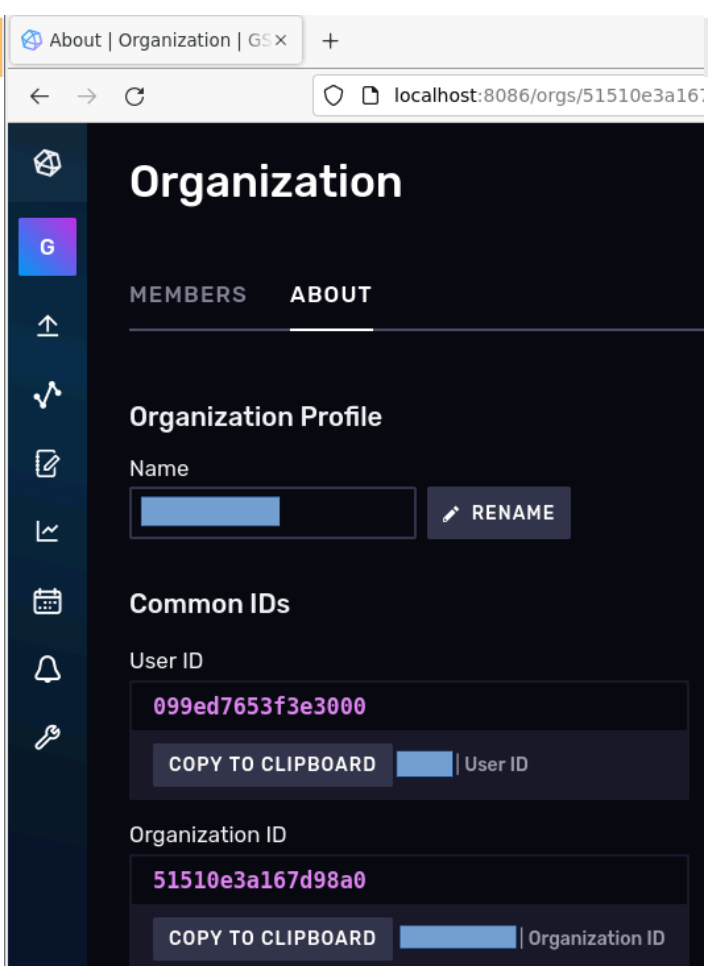
- Preparation

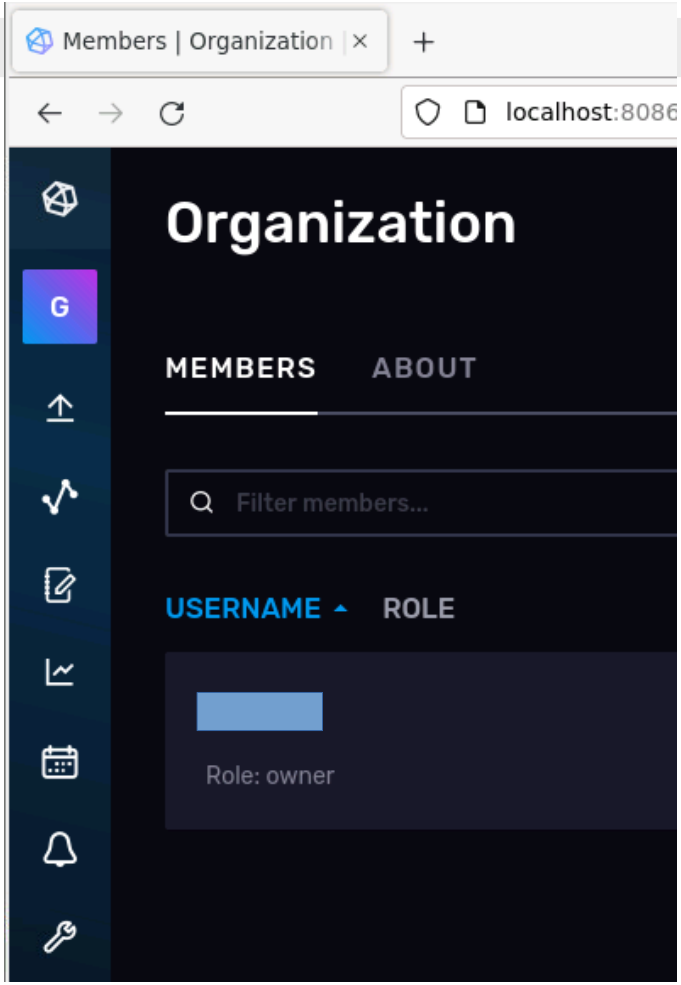
```
mkdir -p ${influxdb_dataDir:?} ${influxdb_configDir:?}
# Initial setup
sudo docker run -d --rm \
  --name "influxdb${influxdb_version:?}_${influxdb_suffix:?}" \
  -p ${influxdb_port:?}:8086 \
  -v ${influxdb_dataDir:?}:/var/lib/influxdb2 \
  -v ${influxdb_configDir:?}:/etc/influxdb2  \
  -e DOCKER_INFLUXDB_INIT_MODE=setup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

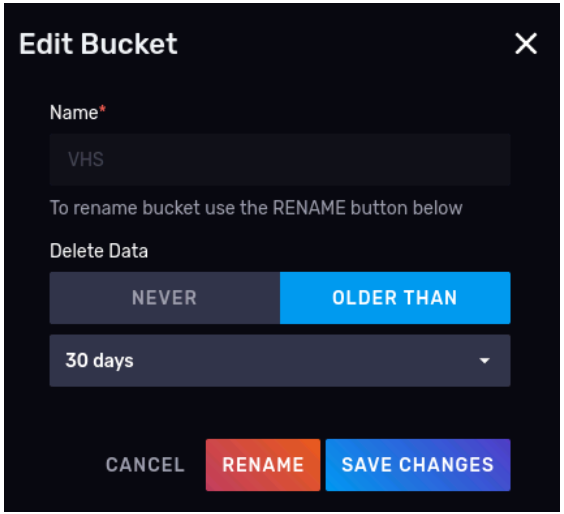
Web-Interface: Organization & Member

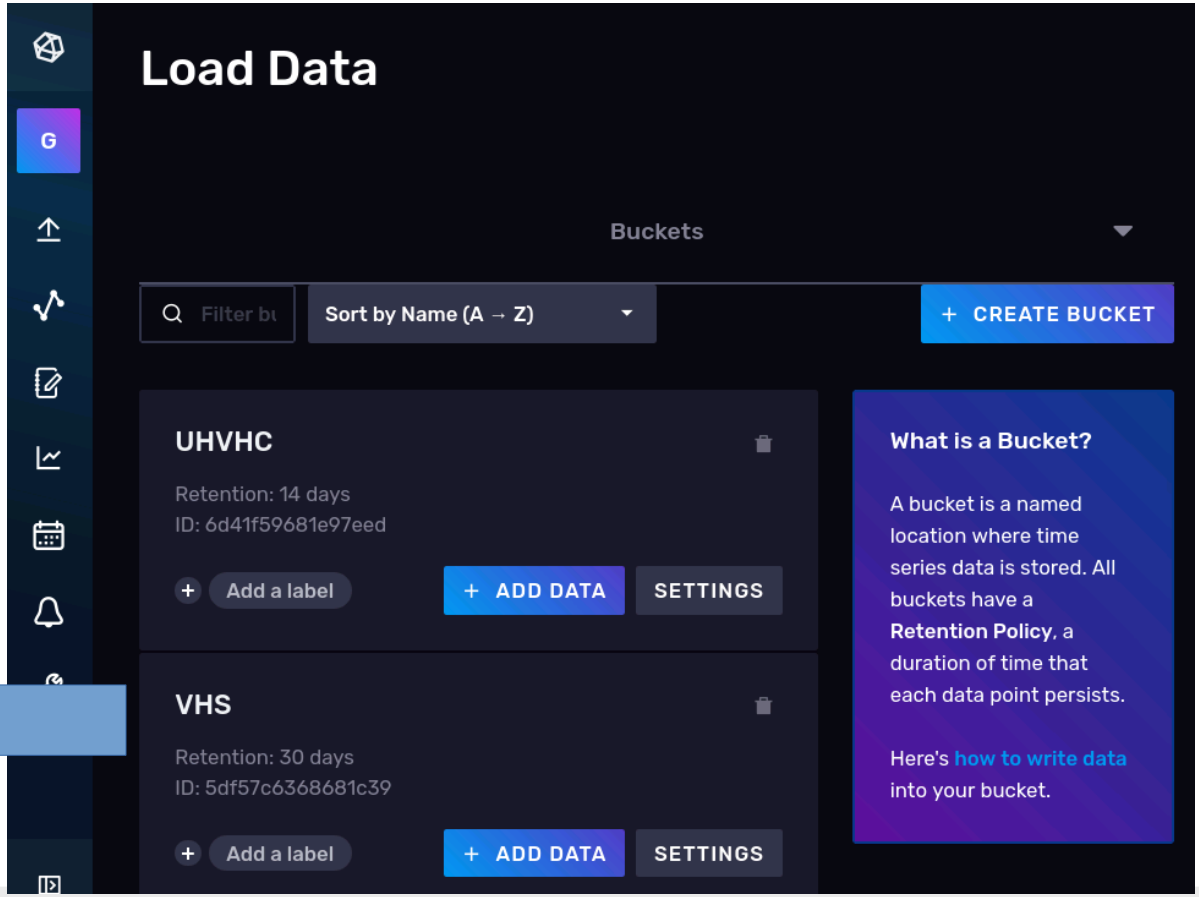




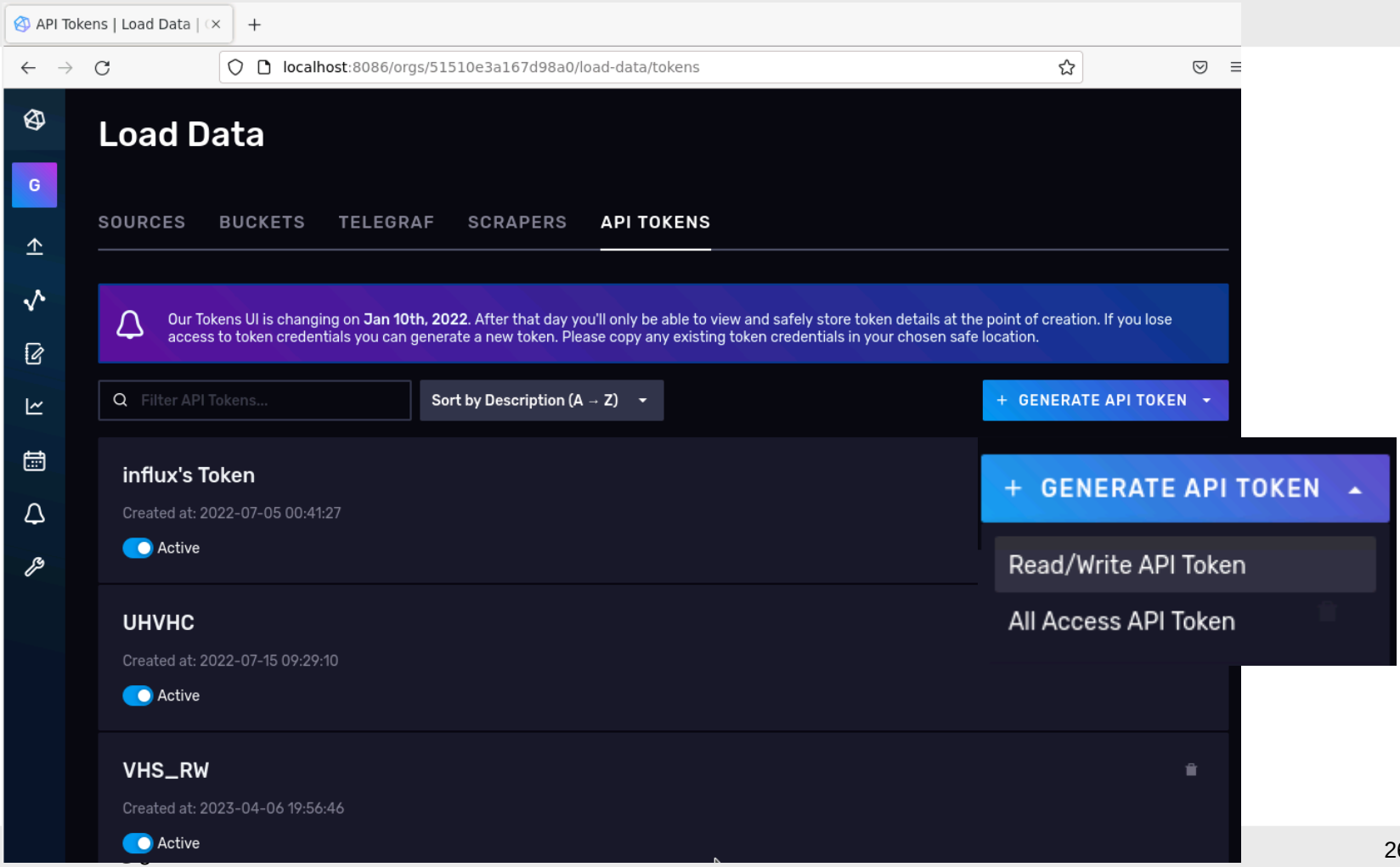
Bucket

- 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.

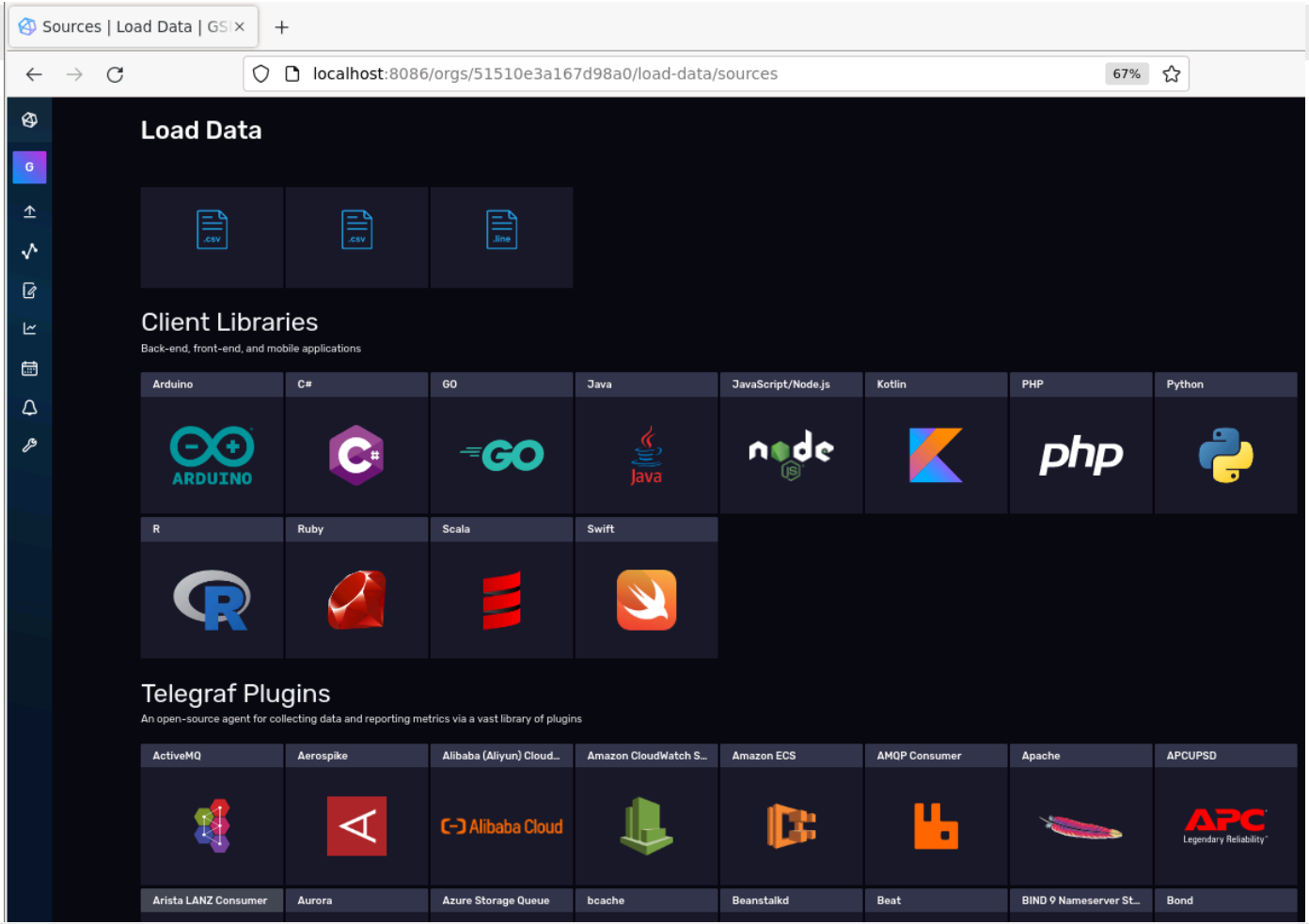




API Access Tokens



Loading Data



Data Explorer

Data Explorer Query

```
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

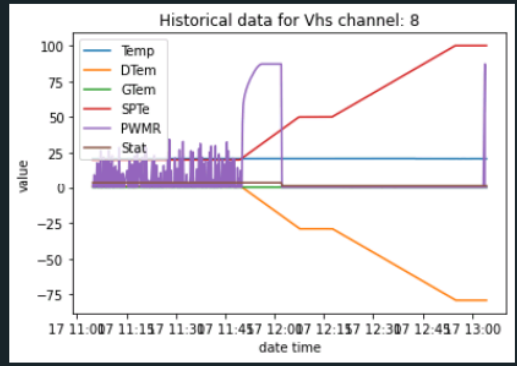
/home/brand/Python/vhs_query_influxdb.py

vhs_query_influxdb.py*

```
from influxdb_client import InfluxDBClient
import matplotlib.pyplot as plt
import time
url =
token
org =
bucket = 'VHS'
t0 = time.time()
time_span = '-2h'
aggregate_window = None
# aggregate_window = '30s'
actor = 'Vhs'
channel = 8
attributes = ('Temperatures', 'DTemperatures', 'GTemperatures', 'SPTemperatures', 'PWMRatios'
queries = []
for attribute in attributes:
    query = 'from(bucket: "VHS") |> range(start: ' + time_span + ') |> filter(fn:(r) => r["Ac
    if aggregate_window:
        query += ' |> aggregateWindow(every: ' + aggregate_window + ', fn: last, createEmpty:
    queries.append(query)
if True:
    with InfluxDBClient(url=url, token=token, org=org) as client:
        plt.title('Historical data for ' + actor + ' channel: ' + str(channel))
        plt.xlabel('date time')
        plt.ylabel('value')
        query_api = client.query_api()
        for ii, query in enumerate(queries):
            # print('Query:', query)
            result = query_api.query(org=org, query=query)
            results = []
            values = []
            times = []
            for table in result:
                for record in table.records:
                    try:
                        # print('record:\n', record)
                        # print('type(record["_time"])', type(record["_time"]))
                        results.append((str(record["_time"]), record.get_value()))
                        times.append(record["_time"])
                        values.append(record.get_value())
                    except Exception:
                        pass
            # print(results)
            plt.plot(times, values, label=attributes[ii][:4])
        plt.legend(loc='best')
        plt.show()
```

vhs_query_influx

106 %

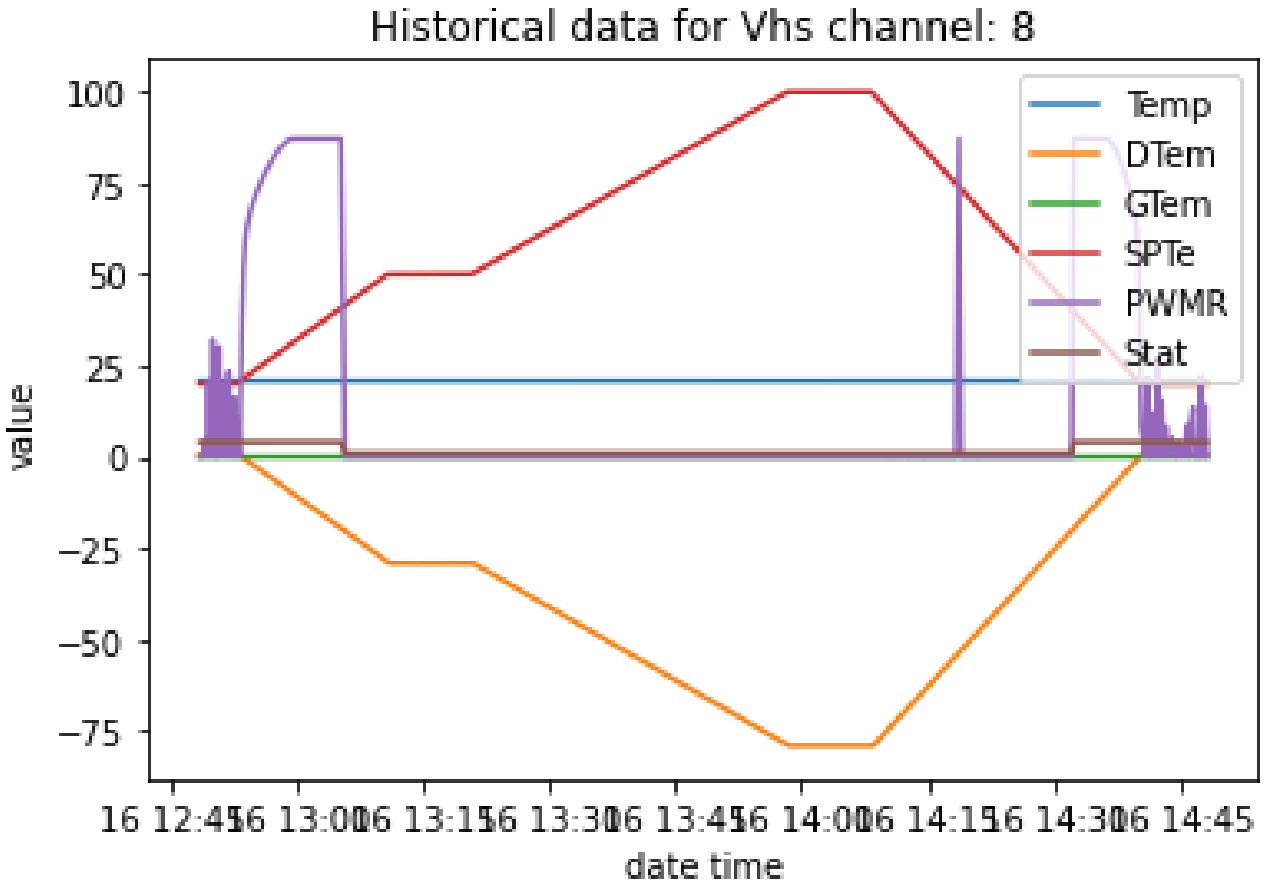


Help Variable Explorer Plots Files

Console 1/A

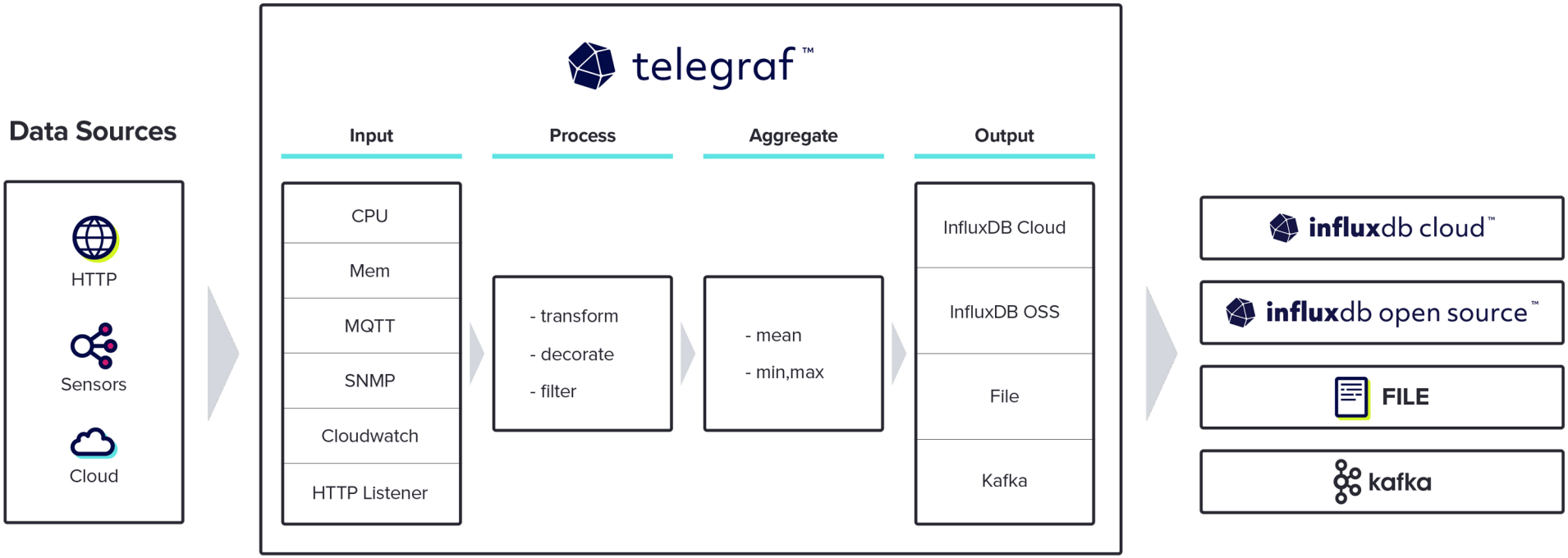
```
Type "copyright", "credits" or "license" for more information.

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/



Telegraf – Plugins
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.

Telegraf – Configuration
Example: VHS

- 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 = "_/_/_"

Telegraf – Start in Docker

- Thanks to P. Zumbruch for help.
- Set Export environment variables

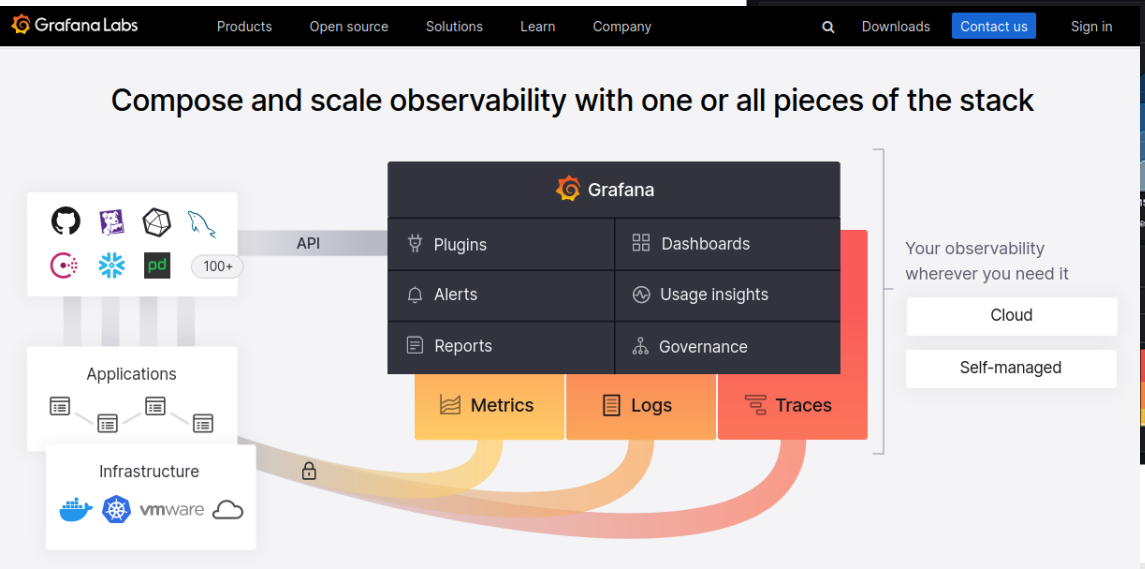
```
export telegraf_suffix=01
export telegraf_configDir=/home/$USER/Container/Telegraf/config/${telegraf_suffix:?}
export telegraf_dataDir=/home/$USER/Container/Telegraf/data/${telegraf_suffix:?}
```

- 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
```

Grafana

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



Dashboards

- 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

Grafana – Start in Docker

- Thanks for help
  - Christian Jacobsen and Jörn Plewka, HZ Hereon.
  - P. Zumbruch
- 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

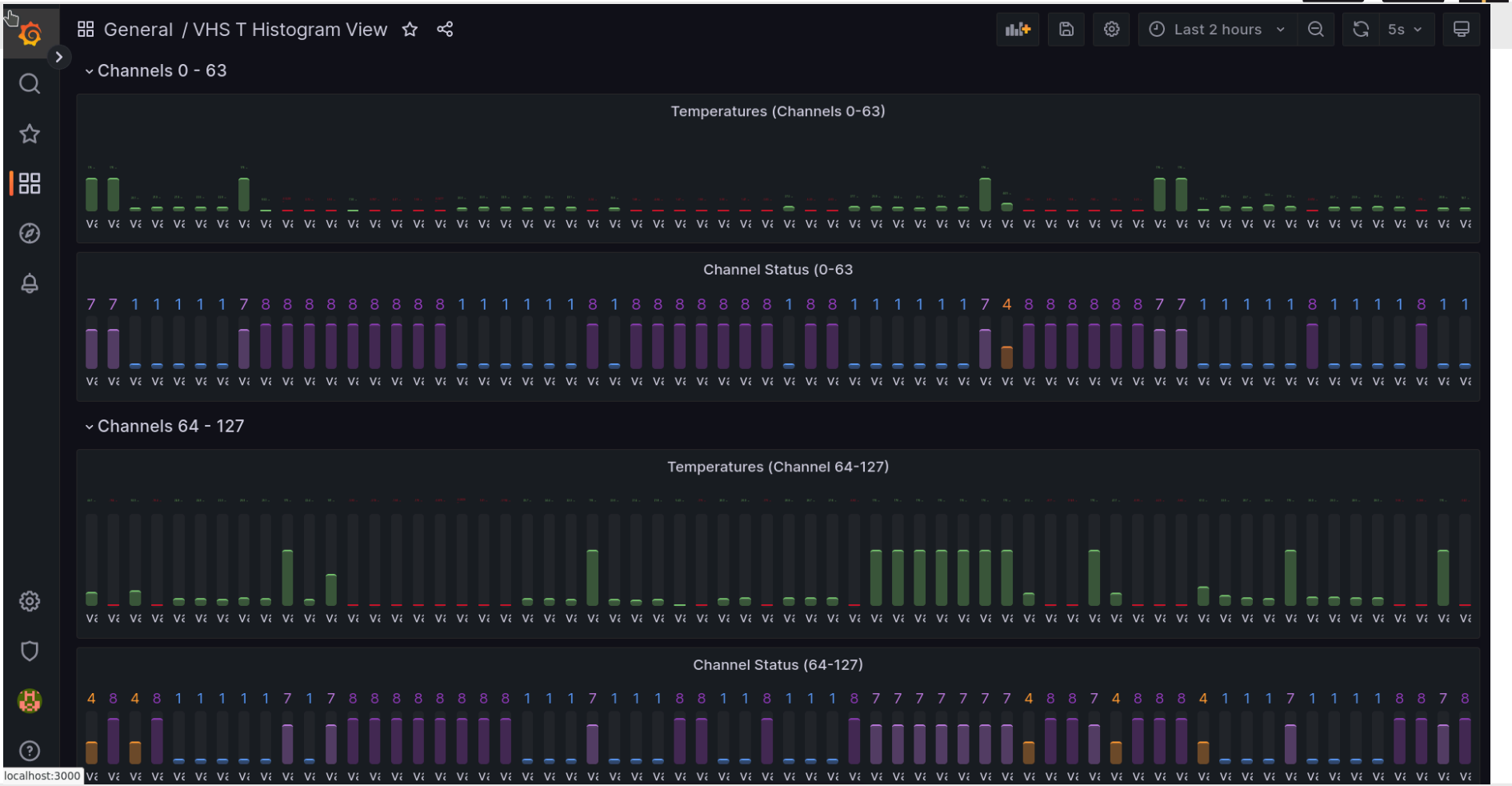
```
source /home/$USER/Container/Grafana/GrafanaEnv.sh
sudo docker or better podman run -d --rm \
  --name "grafana_${grafana_suffix}" \
  --volume ${grafana_configDir:?}:/var/lib/grafana \
  -p 3000:3000 \
  --user 0 \
  grafana/grafana:latest
```

First Trial: VHS Dashboards

First Trial: VHS Dashboards

First Trial: VHS Dashboards

First Trial: VHS Dashboards



First Trial: VHS Dashboards