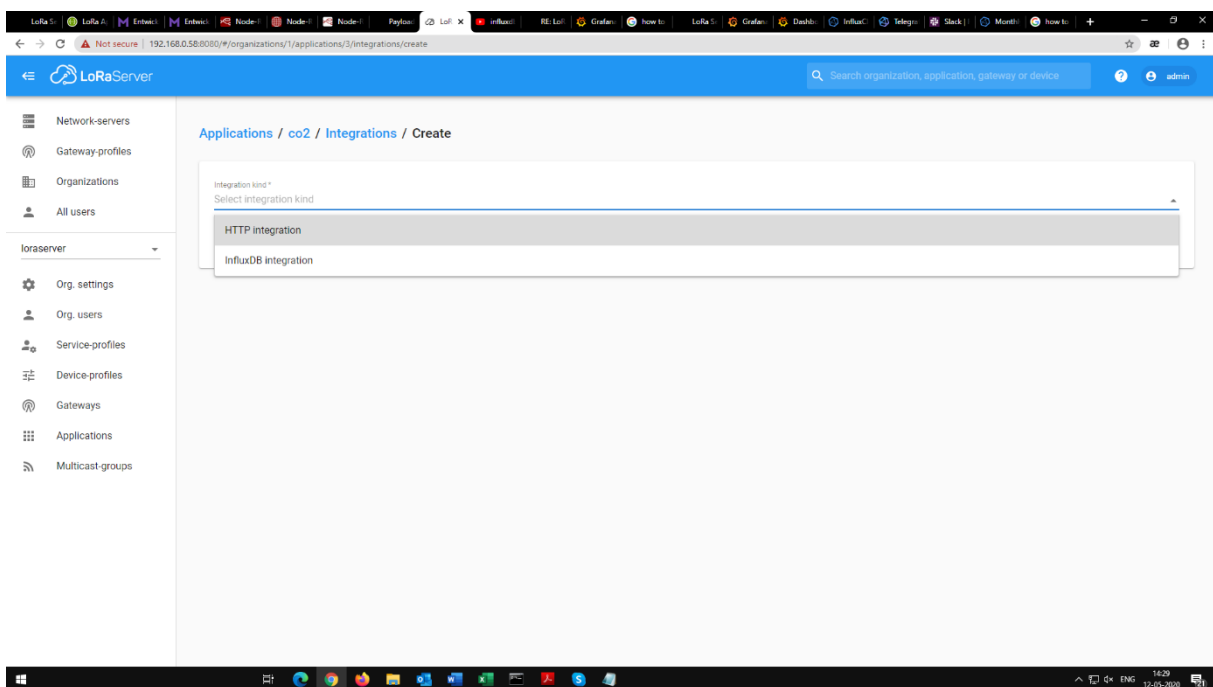


INTEGRATION von INFLUXDB:



Zunächst werden die Daten von the Loraserver direkt an den Influxdb localhost geliefert. Von hier aus muss Influxdb den Zeitstempel sammeln und quantifizieren. Jetzt müssen wir die Messungen aus verschiedenen Datenbanken in Influxdbausführen undüberprüfen.

Es gibt verschiedene Möglichkeiten, influxdb zu hosten, einschließlich der



Hier können wir entweder die INFLUXDB-Integration oder die HTTPS-Integration verwenden, um die Daten in Influxdbzuschreiben. Später verwenden wir verschiedene Methoden, um die Daten zu zeichnen und als Dienst zu laufen, um weitere Probleme mit dem Neustart des Geräts zu vermeiden.

1) HTTPS-Integration:-

Hier ist es vergleichsweise einfach, da wir keine physische Installation von Influxdb auf unseren Geräten benötigen. Wir können einfach die laufende Influxdb Cloud verwenden und unsere Daten dort mit der bereits mitgelieferten Node-Red-Funktion schreiben. Hier möchten wir die Daten direkt auf eine Seite schreiben. Die Daten können direkt von dort aus gelesen werden. Dies geschieht mit LoRaserver. In Integrationen sagen wir HTTP-Integration und geben die spezifische Adresse der Seite an, auf der wir unsere Daten schreiben möchten. Es kann von jedem Server in die Cloud zum Speichern und Reflektieren der Daten sein.

2) InfluxDB-Integration:-

Hier müssen wir Influxdb auf dem System ausführen und die Daten werden an Port 8086 geschrieben, der aktiv auf die eingehenden Daten abhört.

3) Node-Red:

Dies ist eine der einfachsten und einfachsten Methoden zum Integrieren und Übertragen des Datums zum Erstellen von Dashboards. Hier müssen wir die laufende Node-Red-Anwendung verwenden, um die Daten weiterzuleiten. Diese Methode ist für Windows-Benutzer vergleichsweise einfach und kann wie folgt konfiguriert werden:

Node-RED -> Telegraf -> InfluxDBserver

Node-RED:- Node-RED ist eine einfache programmierbare Sprache für Softwareentwicklung und Hausautomation mit leicht verwendbaren Knoten. Wir benötigen einen HTTP-Postknoten mit der Adresse des Ports/der Seite, an dem wir die Daten veröffentlichen möchten.

Telegraf:- Telegraf ist eine kostenlose Windows-Software von InfluxDB. Es ist für Windows-Verwendung als Datenverwaltung als Service ist einfach für Windows im Vergleich zu InfluxDBbestimmt.

Wenn Sie Port 8086 haben, der Influxdb anhört, kann Port 8186 im Allgemeinen verwendet werden, um dies zu hören.

Influxserver:- Influxserver ist ein Online-Server von InfluxDB. Es ist kostenlos für den begrenzten Gebrauch und kann leicht von jedem System aus eingeloggt werden. Hier sollen die Daten von Telegraf mit Token und Buckets gesammelt werden. Die Tokendaten müssen in die Funktion in Node-RED eingegeben werden.



```
1 msg.headers = {
2   'Authorization': 'awWOHzVperz2wG030jFWeUShcsK6c19IVhYe20c0RnmXL4ygQ4ttNEYlwq_J4PBhqYL_W6Y02ih4Y4n-I1KIpw=='
3 }
4 var co2 = msg.payload.data[11]+msg.payload.data[12];
5 var vbat = msg.payload.data[13]*256+msg.payload.data[14];
6 var temp = msg.payload.data[1]*10;
7 var rh = msg.payload.data[3]+ msg.payload.data[4];
8 msg.payload = "sensorData,sensorName=" +msg.payload.deviceName+ " co2=" + co2 + ",vbat=" + vbat + ",temp=" + temp + ",rh=" + rh;
9 return msg;
```

Abb:Node-red function

▼ node properties

☰ Method

🌐 URL

Enable secure (SSL/TLS) connection

Use basic authentication

← Return

📌 Name

Abb:http response node

Telegraf:-

```
[global_tags][agent]
interval = "10s"
round_interval = true metric_batch_size = 1000
metric_buffer_limit = 10000 collection_jitter = "3s" flush_interval = "10s"
flush_jitter = "5s" precision = "" debug = false
quiet = false
logfile = "/var/log/telegraf/telegraf.log"
logfile_rotation_interval = "0d"
logfile_rotation_max_size = "1MB"
logfile_rotation_max_archives = 5 hostname = ""
omit_hostname = false# Cloud 2 instance
[[outputs.influxdb_v2]]
urls = ["https://us-central1-1.gcp.cloud2.influxdata.com"]
token =
"xxXOXzVprz2WG03OjFWeUSHcsK6c19IVhYe20c0RxxXL4ygQ4ttNEYlwq_J4PBhqYL_W6YO2ih
4Y4n-lKlpw=="
organization = "XXXXX"
bucket = "XXXXX"
flush_interval = "30s"# Influx HTTP write listener
[[inputs.influxdb_listener]]
service_address = ":8086" read_timeout = "10s"
write_timeout = "10s" max_body_size = "500MiB" max_line_size = "64KiB" database_tag =
"bucket_name"
```

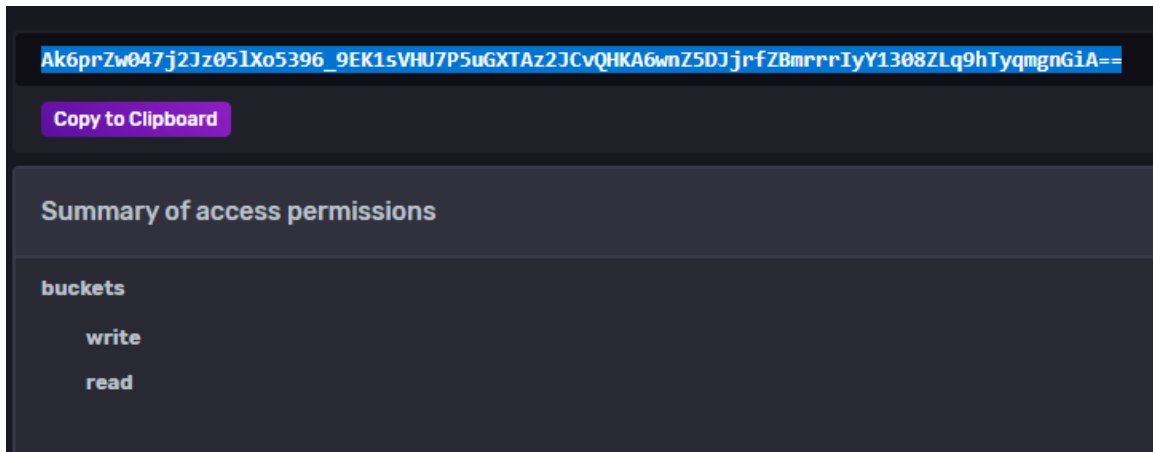


Abb:Data builder InfluxDB

Hier wird aus diesem Token ein ccess zum Lesen/Schreiben von Daten in den Bucket gegeben.

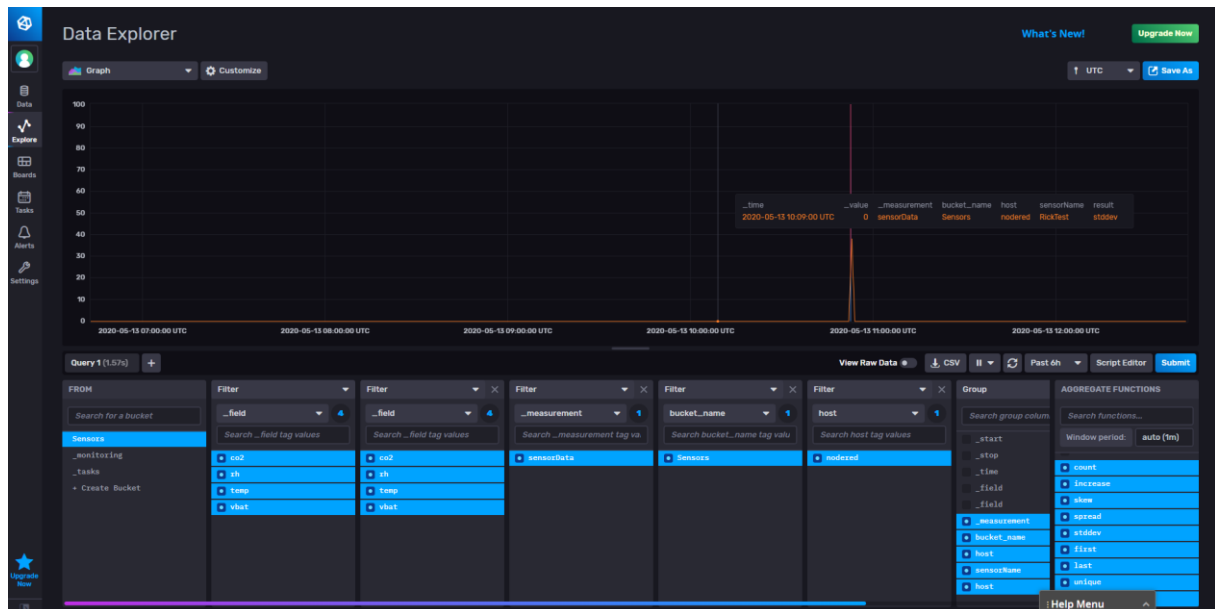


Abb:Data builder InfluxDB

Von dort aus können wir die Daten visualisieren, wie in den Bildern obengezeigt. Hier können wir die Dashboard-Diagramme, Messgeräte, Heatmap und andere Formate erhalten. Wir können auch Schwellenwerte für die einzelnen Metriken festlegen und Benachrichtigungen an die vorgeschriebene Webseite senden oder auch Mails nach unseren Anforderungen senden.

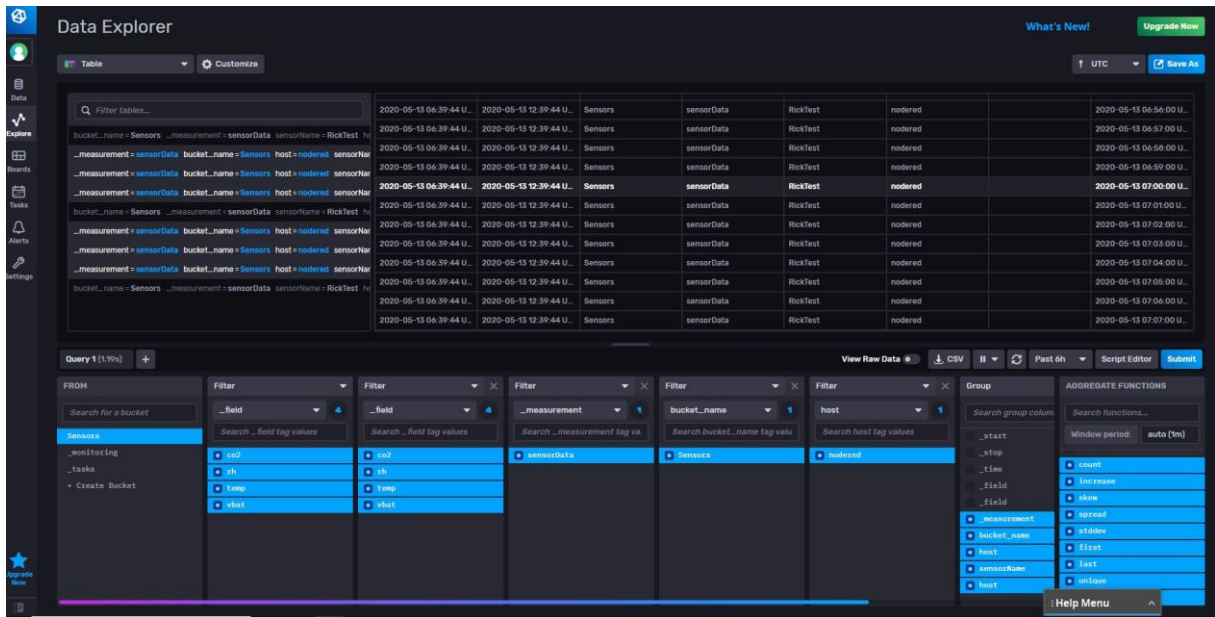


Abb:Data explorer InfluxDB

Vergleich ähnlicher Daten in verschiedenen Visualisierungstools.



Abb:Temperatur Guage InfluxDB

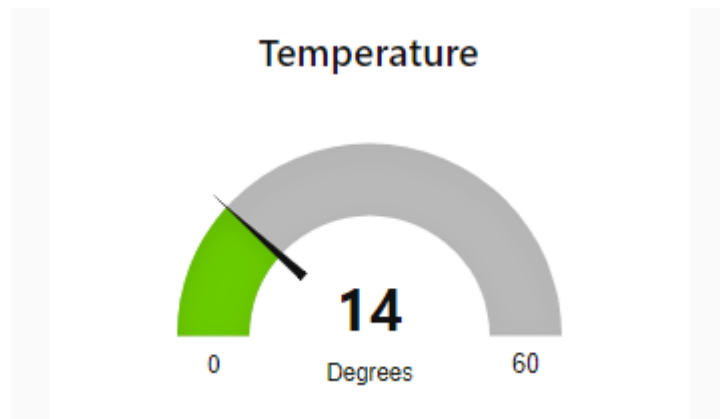


Abb:Temperatur Guage Node-Red

The screenshot shows the Node-RED interface. At the top, a gauge widget displays 40%. Below it, the query editor shows a query with several filters: `_measurement` (sensorData), `_field` (rh), `bucket_name` (Sensors), `host` (node-red), and `sensorName` (EP6-1467). The right sidebar shows aggregate functions like mean, median, max, min, sum, derivative, nonnegative derivative, and distinct.

Relative Humidity

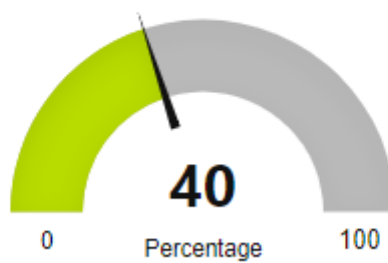


Abb:Temperatur Feuchte-Guage