Video Meeting vom 5.5.2021
Makerspace Reinach

System Setup

Hardware:

1) Raspberry PI 3B+ (https://www.raspberrypi.org)

Software Komponenten:

2) Node-Red (https://nodered.org)

3) InfluxDB (https://www.influxdata.com)

4) Grafana (https://grafana.com)

Beispiel Szenarien

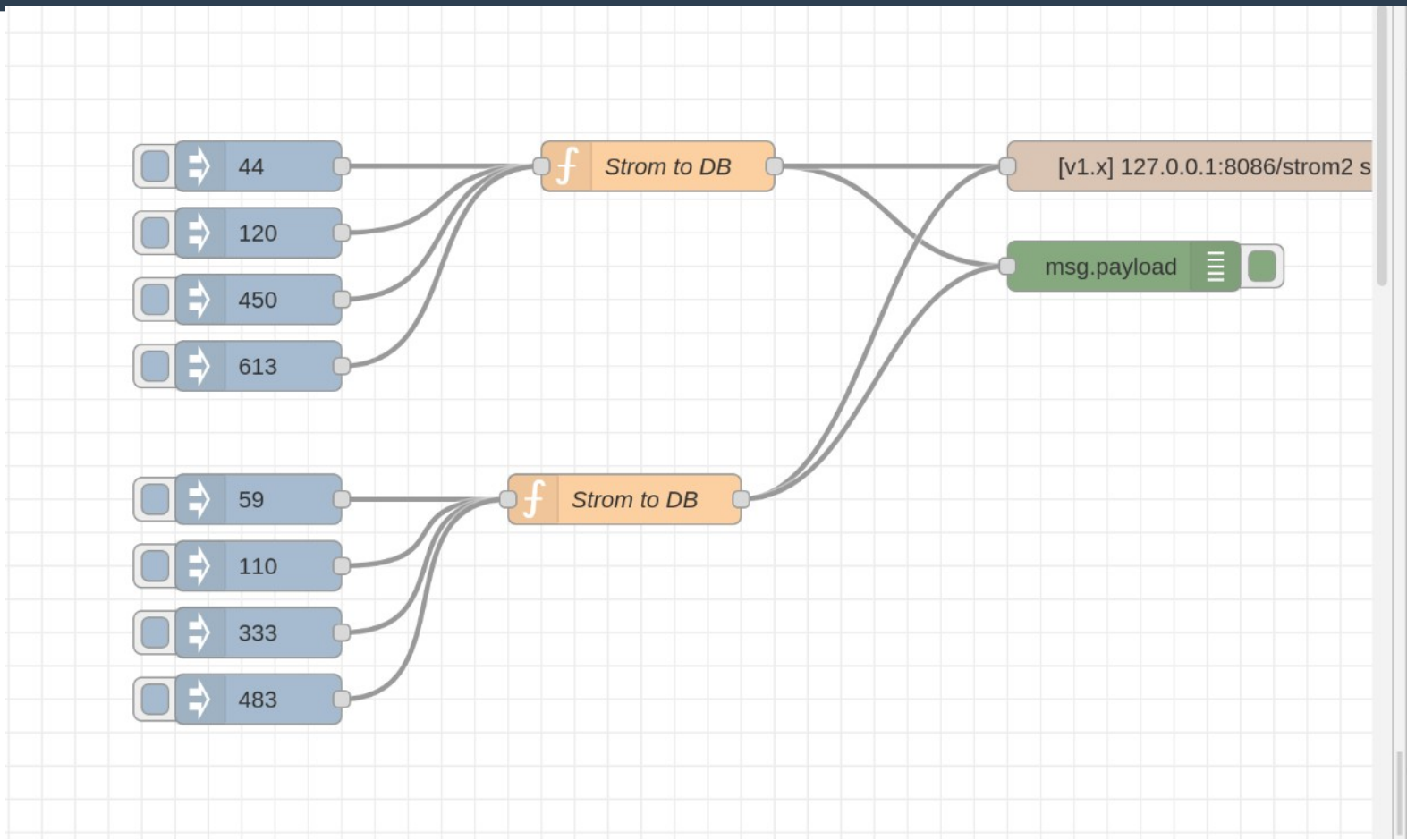
1) Neue Datenbank “strom3” mit Node-Red
2) Aufbewahrungs Zeit einrichten Beispiel: Stromproduktion und Verbrauch “strom2”
3) Daten verdichten Beispiel: RPI CPU
4) Dashboard und Alerts in Grafana

1) Neue Datenbank “strom3” und Node-Red / InfluxBD anlegen

```
pi@pi3w:~ $ influx
Connected to http://localhost:8086 version 1.6.4
InfluxDB shell version: 1.6.4
> create database strom3
```

```
> use strom3
Using database strom3
>
```

1) Neue Datenbank “strom3” und Node-Red / mit Node-Red Test Daten schreiben



1) “strom3” / mit Node-Red Live Daten in die DB schreiben

```
1  var mysolarprod = msg.payload;
2  if ( mysolarprod === undefined){
3      mysolarprod = 0;
4  }
5  msg.payload = [{"PROD": mysolarprod, "USE": 0}];
6  return [msg];
```

Server [v1.x] 127.0.0.1:8086/strom3

Measurement solar

4/25/2021, 9:43:38 PM node: 2853a862.5634d

msg.payload : array[1]

- array[1]
  - 0: object
    - PROD: 44
    - USE: 0

Tabelle

```
> show measurements
name: measurements
name
----
solar
```

```
> select * from solar
name: solar
time                PROD SolarPVA
----                ---- --------
1619379817972484530 44
1619380547298094116 120
1619380549341254812 613
1619381083103792335 613
1619381089710085337 44
```

Datum lesbar

```
> precision rfc3339
> select * from solar
name: solar
time                          PROD SolarPVA
----                          ---- --------
2021-04-25T19:43:37.97248453Z 44
2021-04-25T19:55:47.298094116Z 120
2021-04-25T19:55:49.341254812Z 613
2021-04-25T20:04:43.103792335Z 613
```

2) Analyse Historie Daten “strom(2)” / DB Grösse und Einträge

```
pi@pi3w:~ $ sudo du -sh /var/lib/influxdb/data/*
124K    /var/lib/influxdb/data/auto
88K     /var/lib/influxdb/data/gas
88K     /var/lib/influxdb/data/gps
47M     /var/lib/influxdb/data/_internal
3.1M    /var/lib/influxdb/data/strom
96K     /var/lib/influxdb/data/strom2
84K     /var/lib/influxdb/data/strom3
5.0M    /var/lib/influxdb/data/temp
```

Solar Produktion und Stromverbrauch seit April 2020
=> 3.1MB

```
> select * from solar limit 3
name: solar
time                           SolarPVA
----                           --------
2020-04-01T13:54:56.432181346Z 3537
2020-04-01T13:56:05.422217742Z 3537
```

Solar Produktion und Stromverbrauch seit April 2020
127´819 Einträge

```
> select count(*) from solar
name: solar
time                   count_SolarPVA count_SolarUSE count_StromVERBR
----                   -------------- -------------- ----------------
1970-01-01T00:00:00Z 70507          70683          127819
```

2) Analyse Historie Daten “strom2” / Hilfsmittel

```
> select * into solar_test from solar
name: result
time                  written
----                  -------
1970-01-01T00:00:00Z 198506
```

Measurement kopieren

```
> drop measurement solar_test
> select count(*) from solar_test
```

Measurement löschen

2) Analyse Historie Daten “strom2” / Retention Policy

CREATE RETENTION POLICY "solar_test_oneyear" ON "strom2" DURATION 52w REPLICATION 1

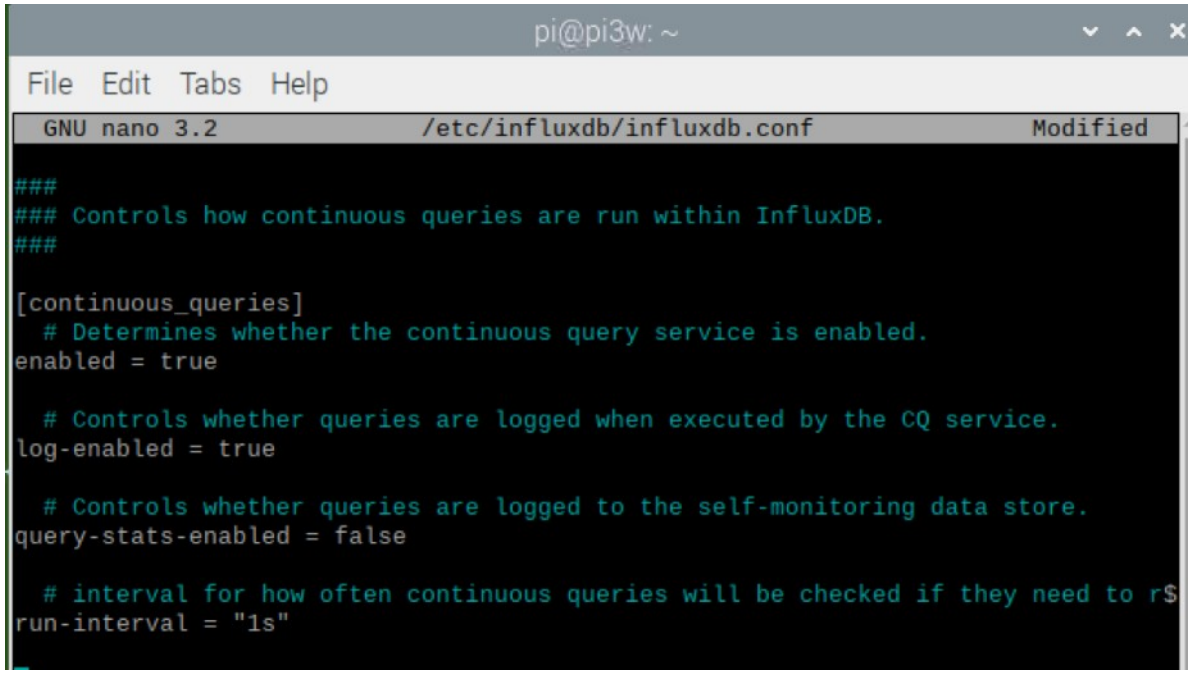
```
> show retention policies
name                duration  shardGroupDuration replicaN default
----                --------  ------------------ -------- -------
autogen             0s        168h0m0s           1        true
solar_test_oneyear 8736h0m0s 168h0m0s           1        false
```

```
> select count(*) from solar_test_oneyear
name: solar_test_oneyear
time                 count_PROD count_SolarPVA count_SolarUSE count_StromVERBR count_TESTTAG count_USE
----                 ---------- -------------- -------------- ---------------- ------------ ---------
1970-01-01T00:00:00Z 16         375            375            757              7            16
>
```

Um Daten mit dieser Policy zu speichern entweder die Messwerte bei INTO angeben oder die Regel mit dem Namen “DEFAULT” als Standardregel festlegen.

3) Daten verdichten Beispiel: RPI CPU / Continuous Query Config

sudo nano /etc/influxdb/influxdb.conf



$ sudo systemctl restart influxdb

3) Daten verdichten Beispiel: RPI CPU / Check Field Types

```
> show field keys
name: rpi3w
fieldKey fieldType
-------- ---------
CPU      string
```

!

```
> show field keys
name: rpi3w
fieldKey fieldType
-------- ---------
CPU      float
```

OK

3) Daten verdichten Beispiel: RPI CPU / Continuous Query + Retention Policy

```
CREATE CONTINUOUS QUERY "rpi3w_test_60s" ON
"rpi" BEGIN SELECT mean("CPU") AS "mean_CPU",
max("CPU") AS "max_CPU", min("CPU") AS "min_CPU"
INTO rpi3w_test_onemin FROM "rpi3w" GROUP BY
time(60s) END
```

```
> show continuous queries
```

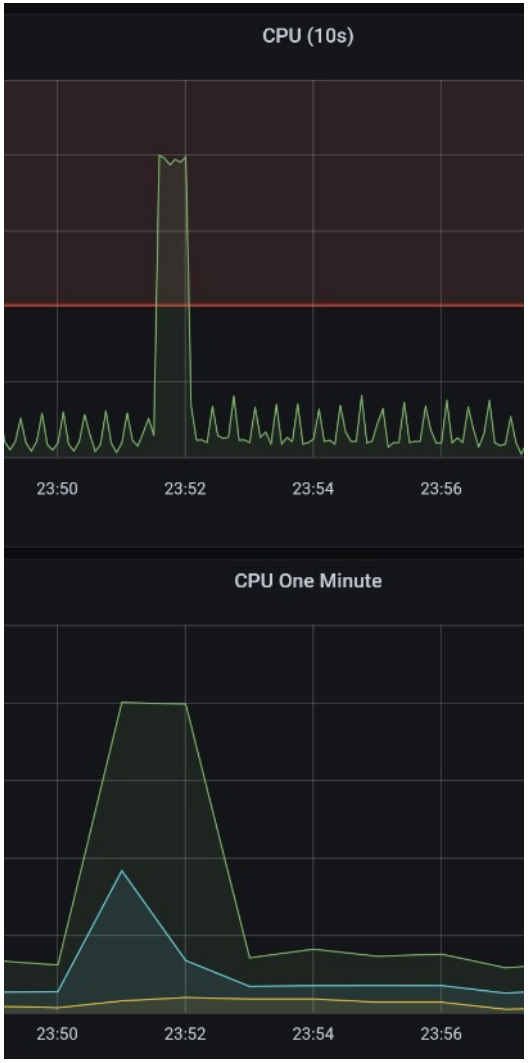
```
name: rpi
name           query
----           -----
rpi3w_test_60s CREATE CONTINUOUS QUERY rpi3w_test_60s ON rpi BEGIN SELECT
n(CPU) AS min_CPU INTO rpi.autogen.rpi3w_test_onemin FROM rpi.autogen.rpi3
```

```
DROP CONTINUOUS QUERY rpi3w_test ON rpi
```

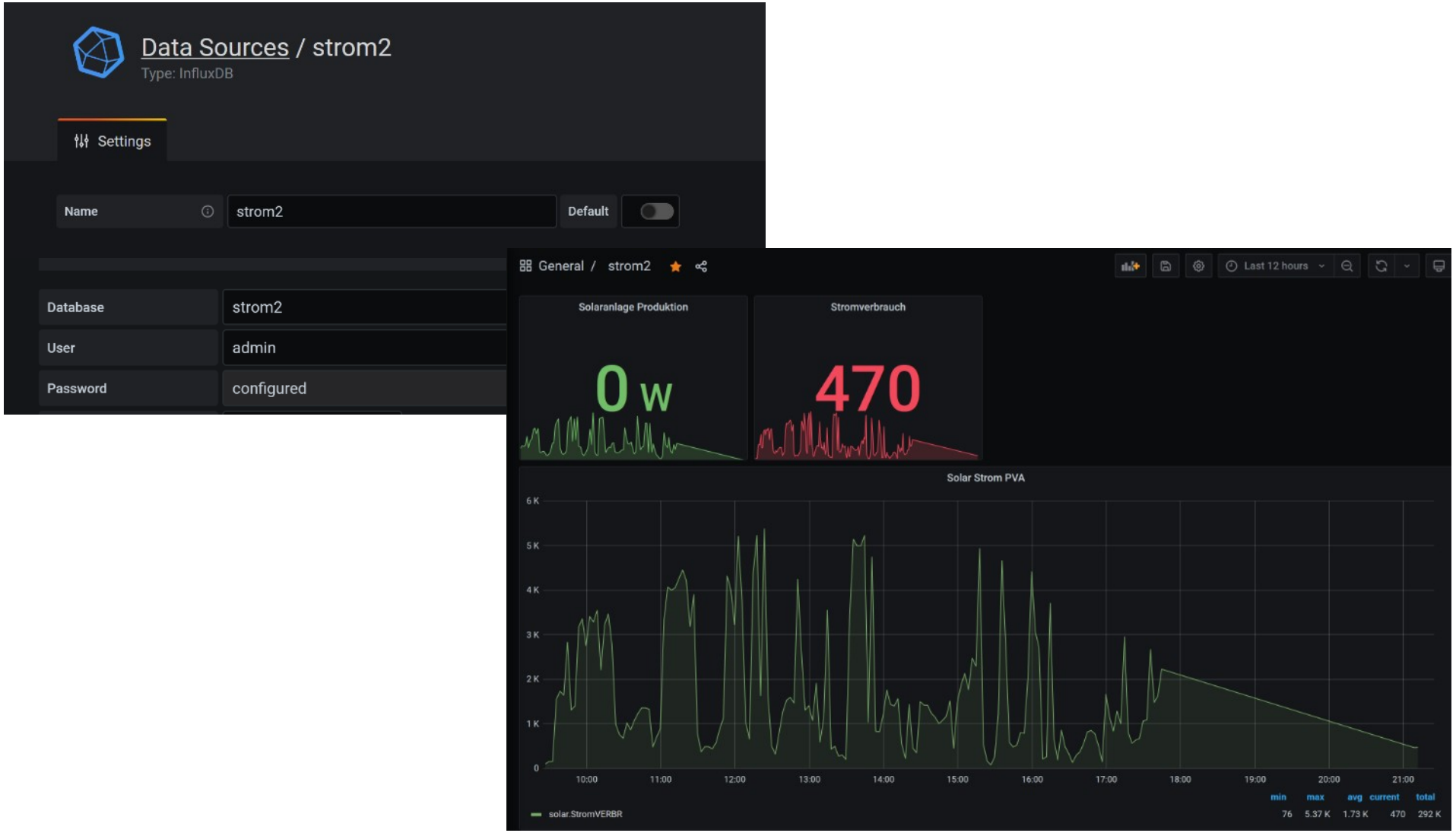
3) Daten verdichten Beispiel: RPI CPU / Daten vergleichen

361 vs. 31

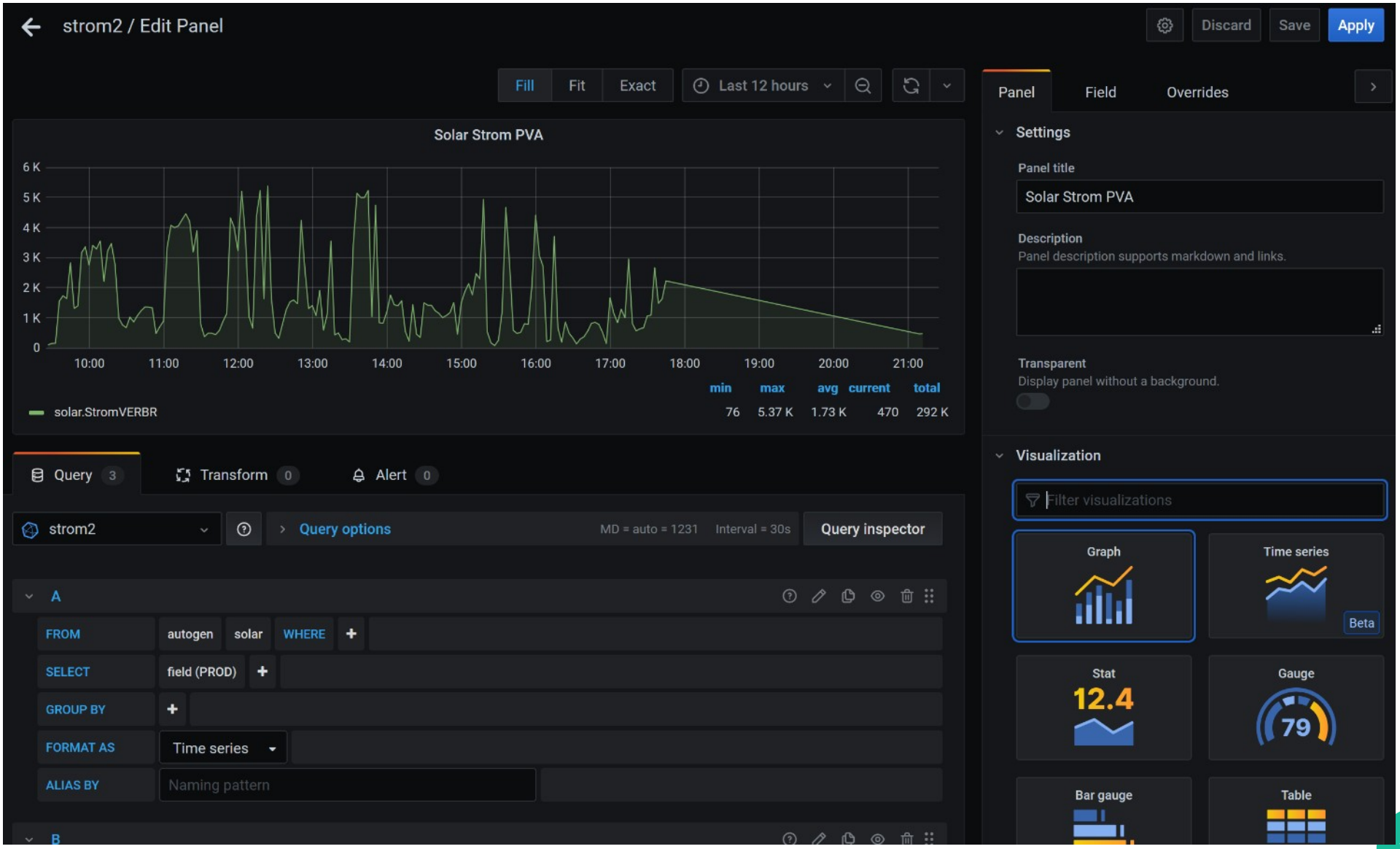
```
> select count(*) from rpi3w
name: rpi3w
time count_CPU
---- ---------
0    361
> select count(*) from rpi3w_test_onemin
name: rpi3w_test_onemin
time count_max_CPU count_mean_CPU count_min_CPU
---- ------------ ------------- -------------
0    31           31             31
```



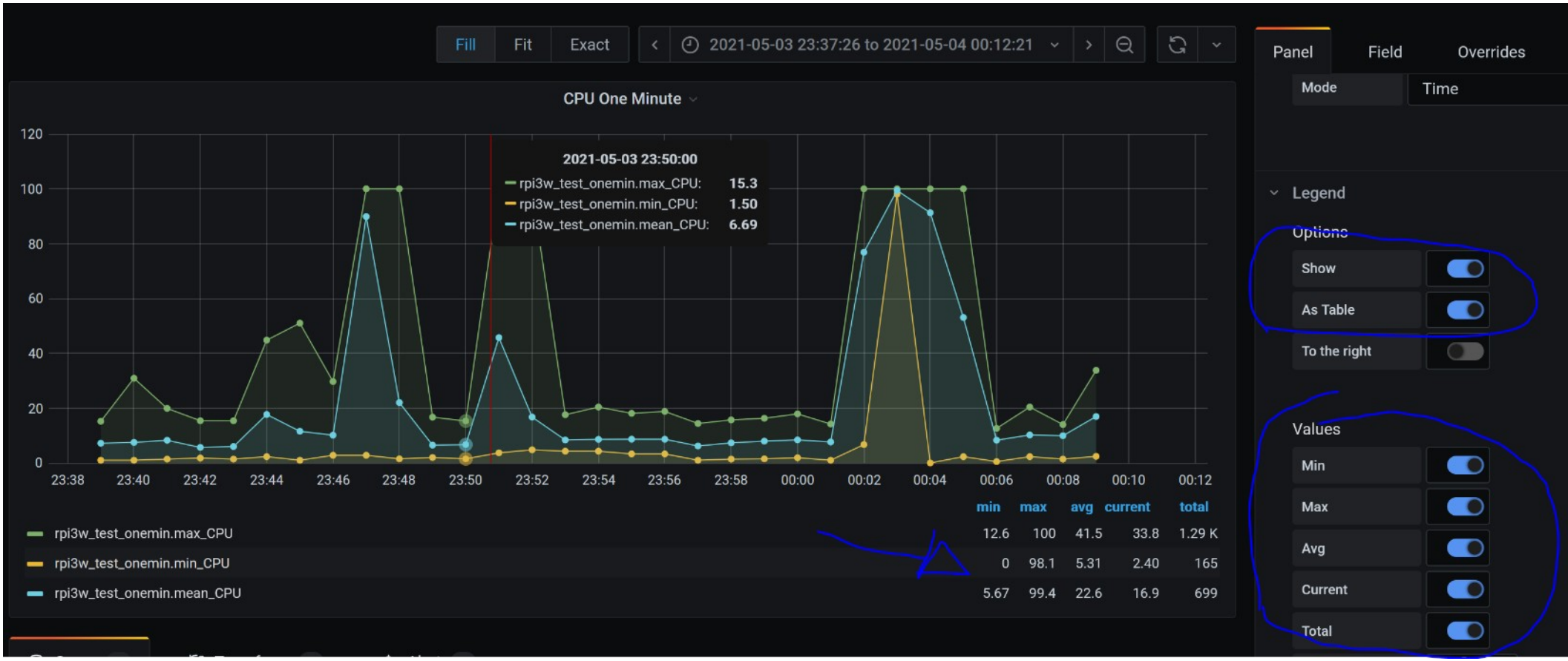
4) Dashboard und Alerts in Grafana
DB Setup & Dashboard



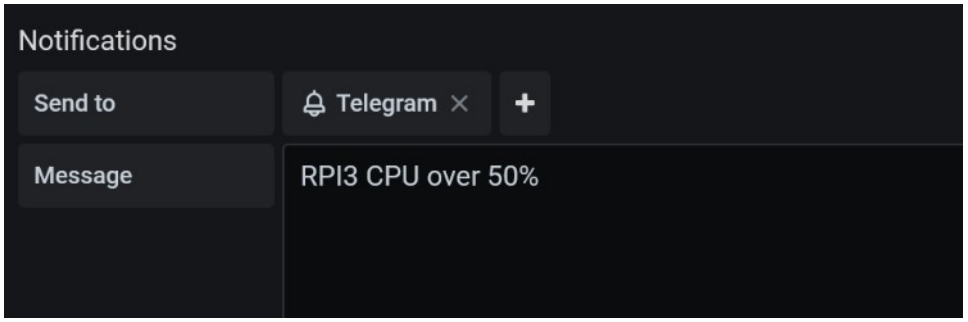
Dashboard und Alerts in Grafana
Dashboard Design

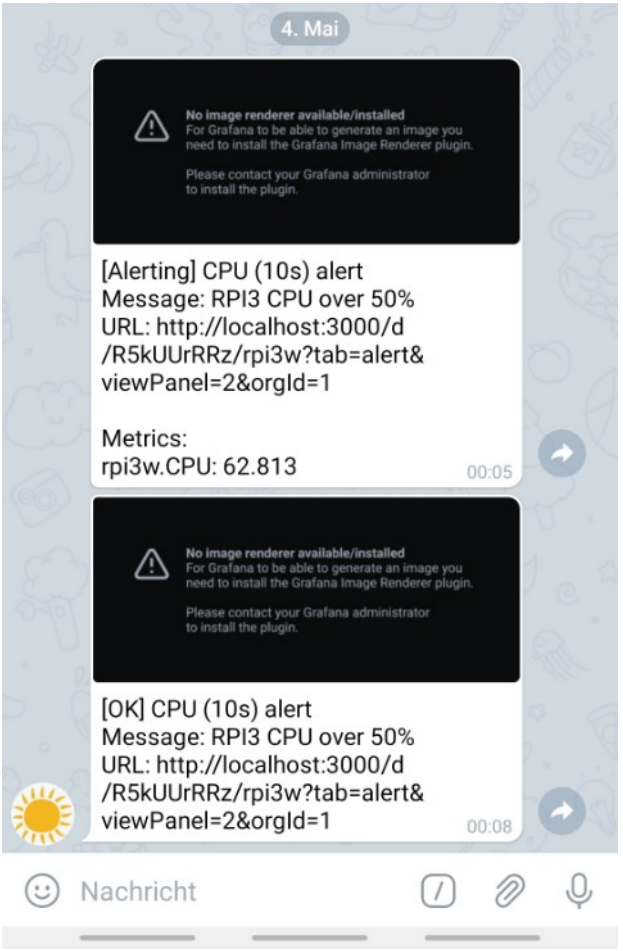


Dashboard und Alerts in Grafana Werte Tabellen

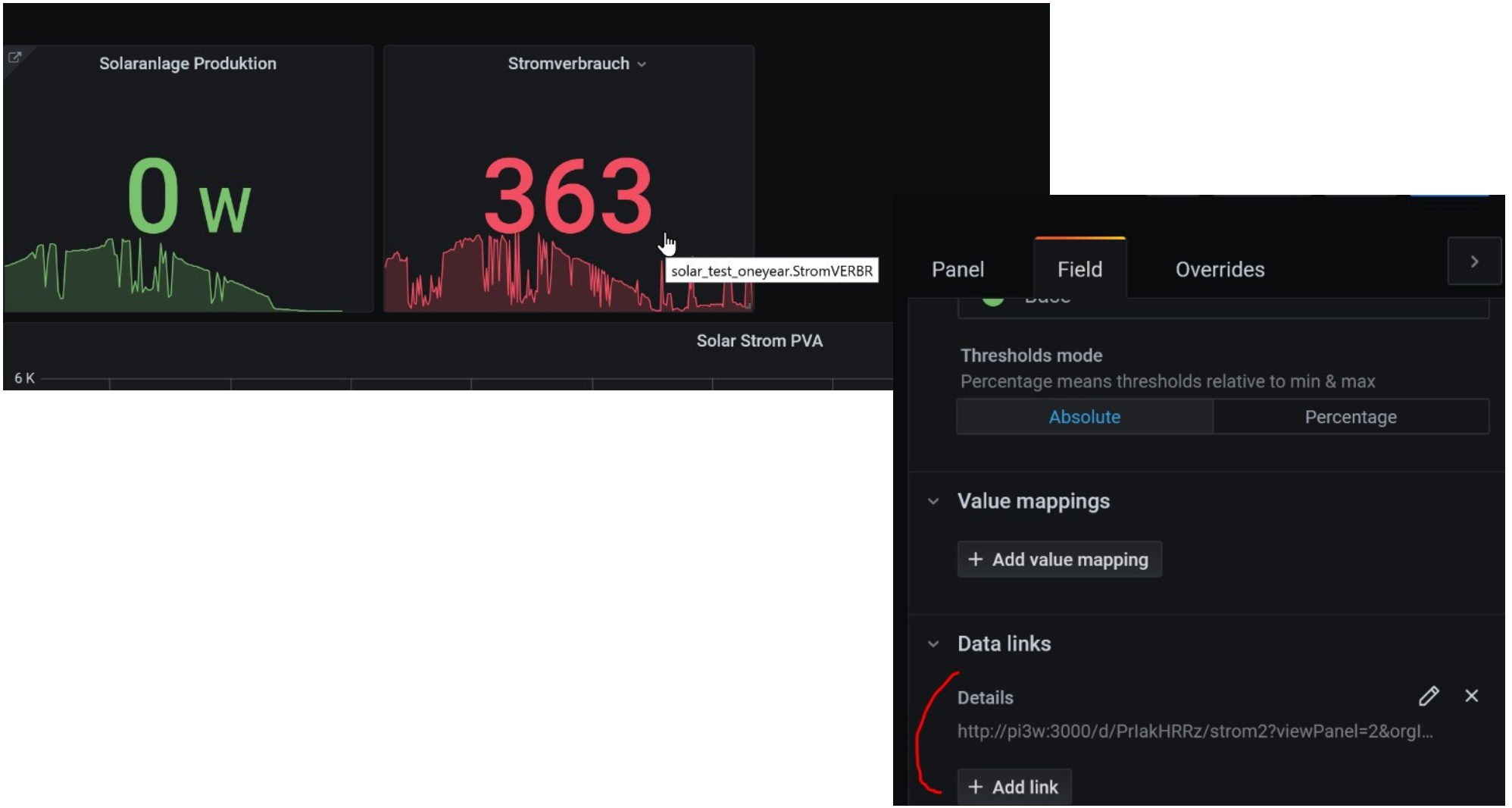


Dashboard und Alerts in Grafana / Alerts





Data Links in Grafana



Fragen?