Visualization

In the following we present our visualizations associated to the following ES indices:

- Count: 48_111_1253
- Restart: 48_111_1252

The values have been sent by an ESP32.
To avoid confusion in the similarity of terms we make the following distinction:

- count: Describes the number of events, i.e. MQTT messages, sent to the IoT platform
- room-count: Describes the number of people in the room. In the _source fields of ES it is indicated by *value*

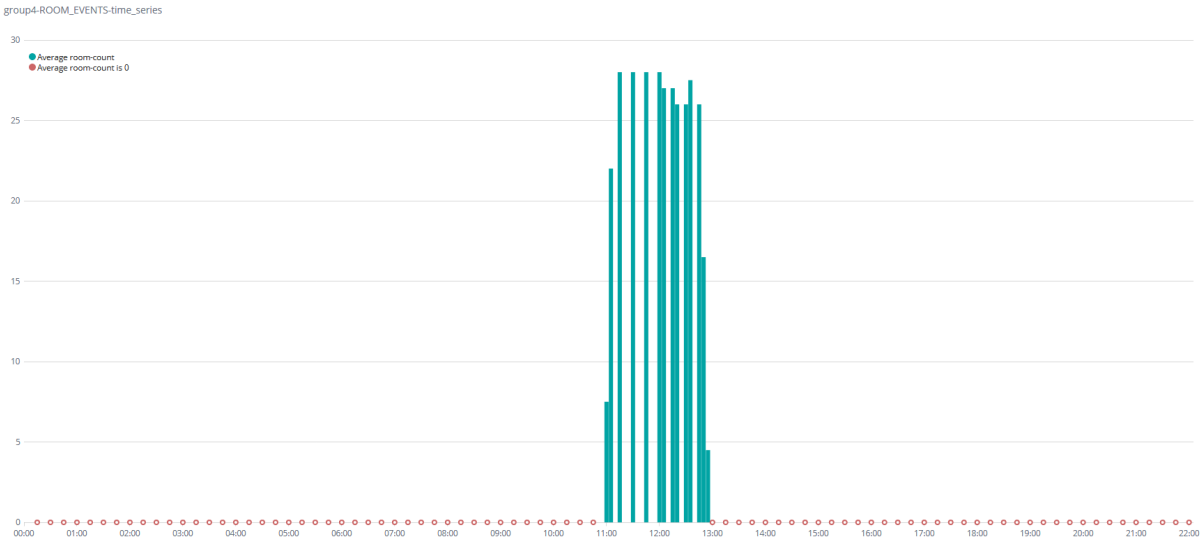
We are using in total 9 visualizations

- Average room-count over time using *Timelion*
- Restart count of device (here an ESP) over time using *Timelion*
- Average filling ratio of the room over time using *Goal*
- Room-count per time slot using *Heatmap*
- Average room-count over time using *Line*
- Remembering latest room-count using *Metric*
- Statistics about room-count using *Data Table*
- Emphasizing latest room-count using *Gauge*
- Counting events per possible room-count using *Vertical Bar*

Average room-count over time using *Timelion*

Settings:

- timefield: timestamp
- metric: avg:*value*
- bucket interval: auto (gets set to 1 minute)
- two expressions
    - green bars if average is greater than 0
    - red dots if average is 0



Interpretation:

- The diagram shows the average room-count on 2021-05-21
- The auto mechanism has set the buckets to 1 minute
- We clearly see that around 11 am people entered the room and left finally all some minutes before 1 pm
- The gaps especially between the bars do not indicate a 0 room-count but empty buckets

Why chosen:

- We get easily a nice overview over the room fill over time and we see when the device has sent messages to the IoT platform
- Second expression (red dots) helps to differentiate between empty buckets and 0 room-count messages

Restart count of ESP over time using *Timelion*

This diagram is similar to the one plotting the room-counter. In the diagram one can see when the restarts happen. Restarts happen...

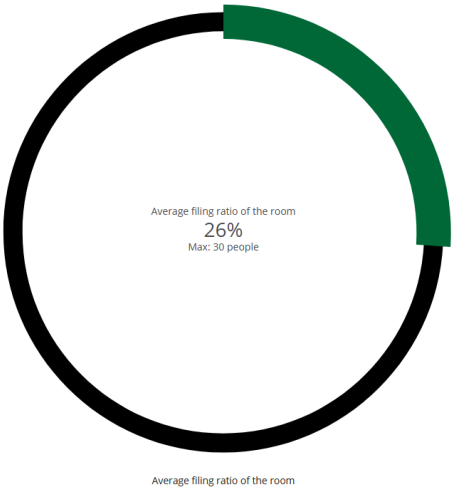
- in the evening so far because of developing the ESP
- at midnight at which time the ESP makes a software restart
- after reconnection when the peer (the IoT platform) had refused the connection for a certain time

Average filling ratio of the room over time using *Goal*

Settings:

- Aggregation: Average
- Field: *value*
- Range from 0 to 30

group4-ROOM_EVENTS-goal



Interpretation:

- The diagram shows the average room-count on 2021-05-21 from 8 am until 8 pm
- With 26% there are on average 7.8 persons in the room.

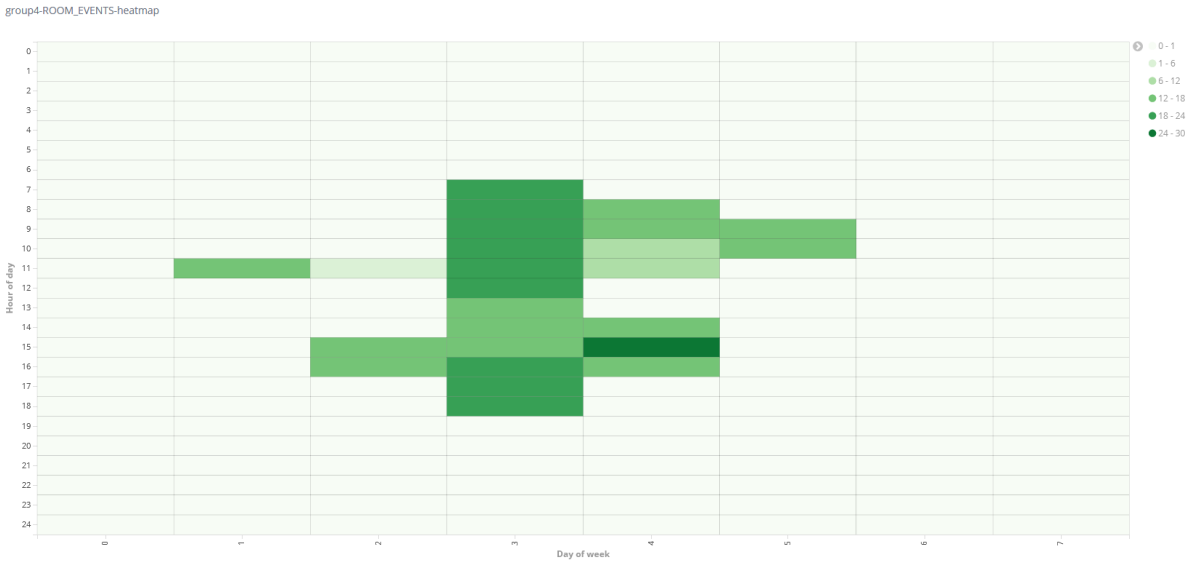
Why chosen:

- Recommended by the staff
- Interesting value to analyze the room attending frequency
- However: 0 room-counts from the evening and night might be included if we set for instance the last seven days as a time interval. This makes the room seem less frequented than it actually is.

Room-count per time slot using *Heatmap*

Settings

- Aggregation: Average
- Field: *value*
- X-Axis: buckets by day of week
- Y-Axis: buckets by hour of day



Interpretation:

- The heatmap uses two kinds of buckets: For the X- and Y-Axis we divide the data into time slots to get a timetable. Then we use buckets for the *value*, i.e. the room-count, itself. With custom ranges over the numbers which the *value* can have, each slot is filled with the range color from the range in which the *value* lies.

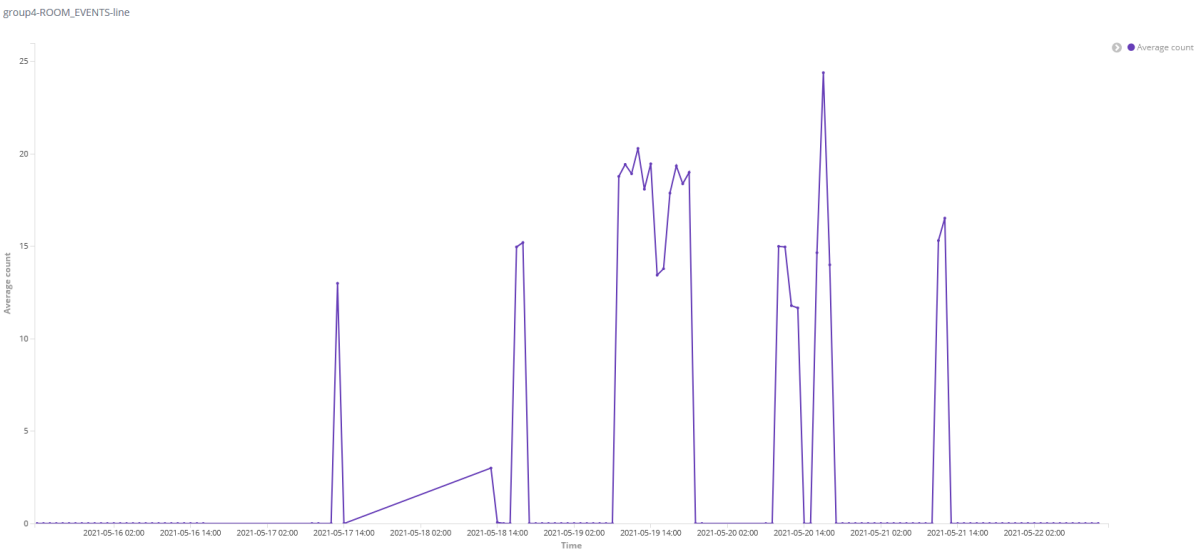
Why chosen:

- We get a timetable that indicates a certain schedule over the week that will help us later in confirming our made predictions.

<u>Average room-count over time using *Line*</u>

Settings:

- Aggregation: Average
- Field: *value*
- X-Axis: buckets by timestamp per 10 minutes
- Y-Axis: average room-count by *value* field



Interpretation:

- The timeline shows the ESP running from noon 2021-05-15 until noon of 2021-05-22
- Reliable room-count values begin with 2021-05-18 in the afternoon after fixing WiFi problems of the ESP because of WiFi channel switches.
- The strange slope from 2021-05-17 noon until 2021-05-18 is due to the ESP being offline for development

Why chosen:

- Similar reasons to the Timelion graph that shows average room-count over time. However, here with the interpolation, the room-count changes are better to understand in a presentation since there are no 0 gaps (caused by the bucket interval) between two successive room-count events.
- Customizable bucket size

Remembering latest room-count using *Metric*

Settings:

- Aggregation: Top Hit
- Field: *value*
- Sort On: timestamp
- Order: Descending

Interpretation (general):

- With the descending order on the timestamp this shows always related to the timestamp the latest received room-count message

Why chosen:

- Easily tracking the latest room-count on the dashboard

Statistics about room-count using *Data Table*

Settings:

- Metrics for *value*: Max, Min, Average, Median
- Buckets by hour for aggregating

Interpretation (general):

- Table with statistics gives statistical information for the room-count

Why chosen:

- More concrete values after observing schedule in the heat map
- Detection of outliers or uncommon numbers, e.g. there is a Max *value* of 31 on 2021-05-19

Emphasizing latest room-count using *Gauge*

Settings:

- Aggregation: Top Hit
- Field: *value*
- Sort On: timestamp
- Order: Descending
- Color Ranges: [0 - 10), [10, 20), [20, 30), [30, 31]

Interpretation (general):

- Shows the latest count similar to the Metric one but the ranges with the color express if the room is (almost) empty, moderately attended, almost full or full

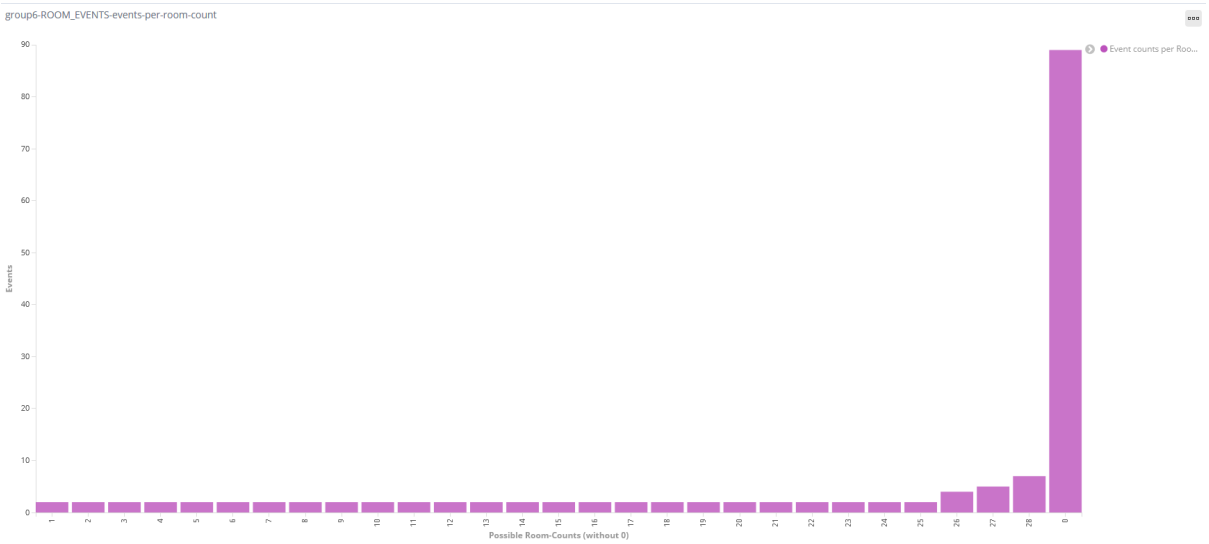
Why chosen:

- The intuitive design with the four ranges in different colors gives the user (even a user who has no idea about the max. room size) feedback how full the room is.

Counting events per possible room-count using *Vertical Bar*

Settings

- Y-Axis basic counting
- X-Axis:
    - Aggregation on field *value* using Terms
    - order ascending the counting result
    - allow 32 bars (including 0 and 31)



Interpretation

- On 2021-05-22 there was one “lecture in the room” and we see a stabilization between 26, 27 and 28 for this

Why chosen:

- It is interesting to see which possible room-count is sent the most often during a certain time period. With this value one might see some stabilization.
- Problem: 0 should be filtered but so far we have not found a possibility to do so (we could eliminate the 0 using the number of bars but for different time periods the set of appearing room-counts differs as well)