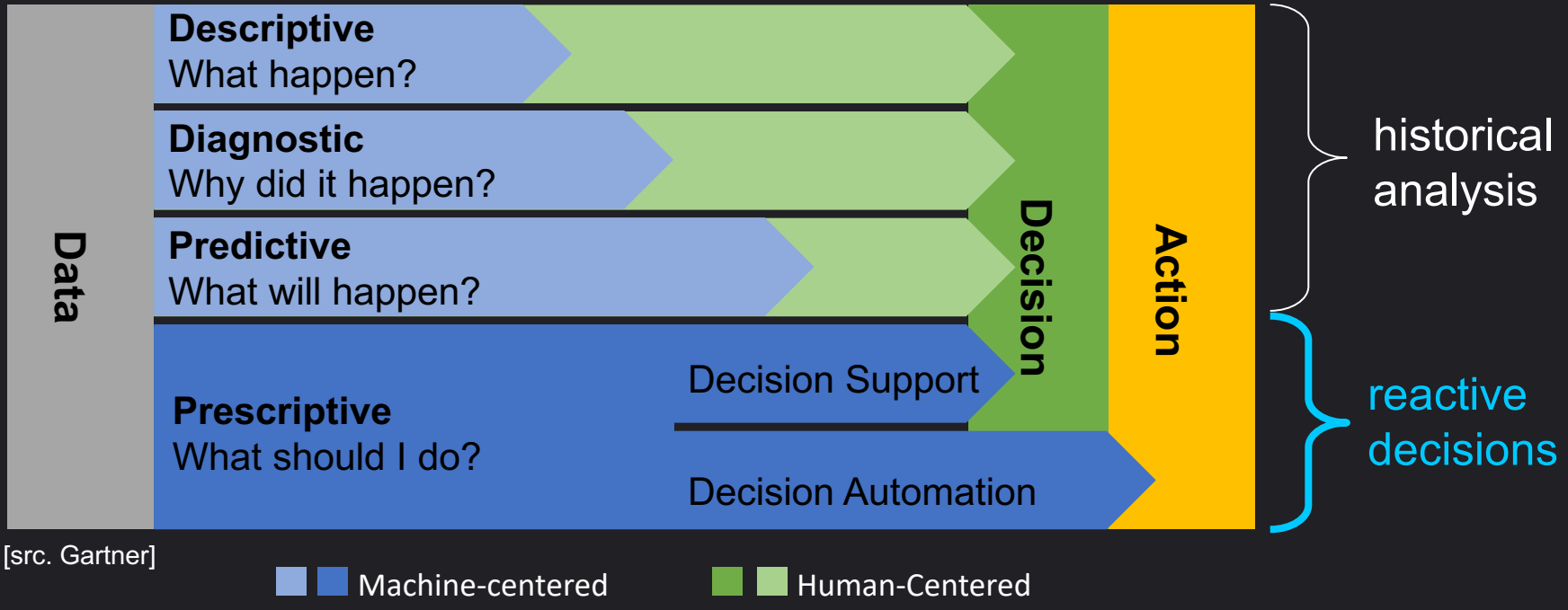
Alerts and Tasks

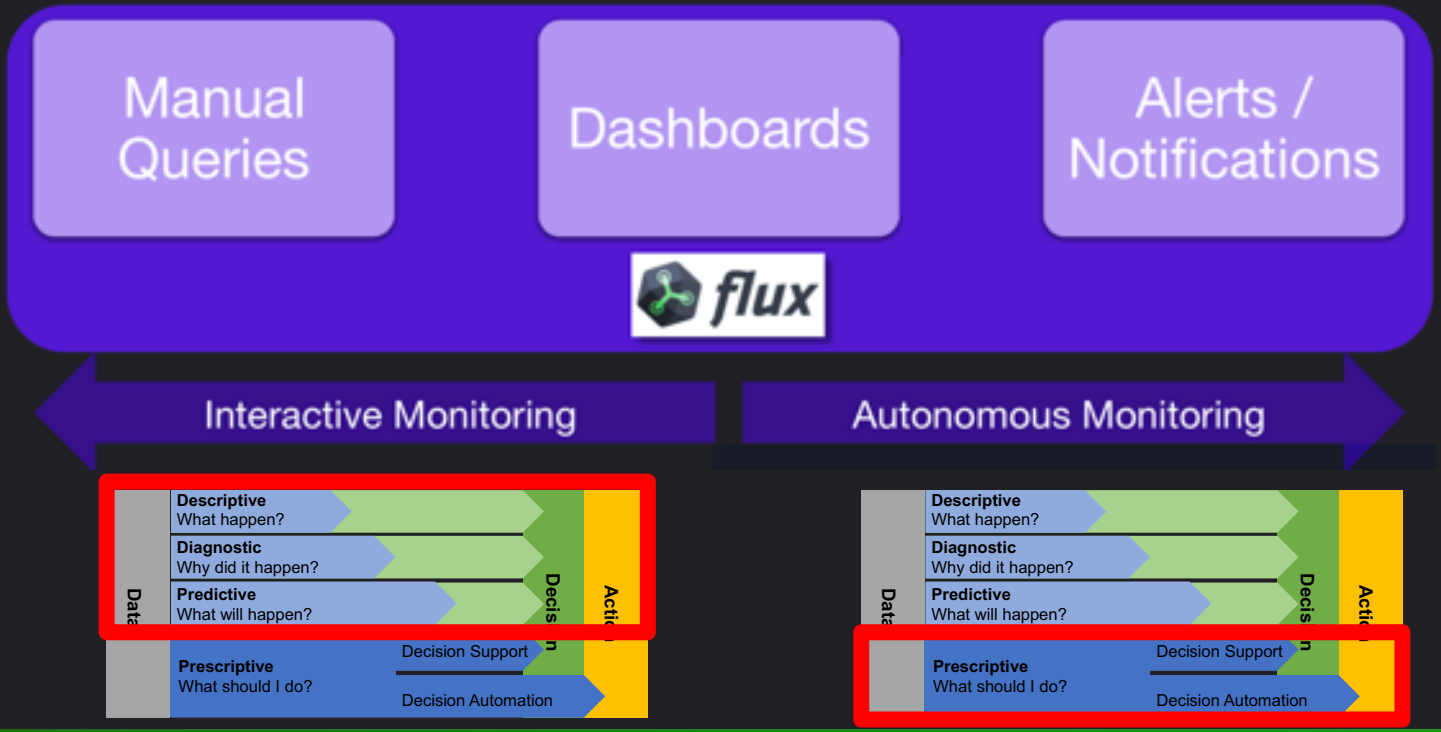
Emanuele Della Valle
Prof. @ Politecnico di Milano
Founder & Partner @ Quantia Consulting
Marco Balduini
Founder & CEO @ Quantia Consulting
Senior Researcher @ Politecnico di Milano

Alerts

Data-driven decision



Interactive vs. Autonomous Monitoring



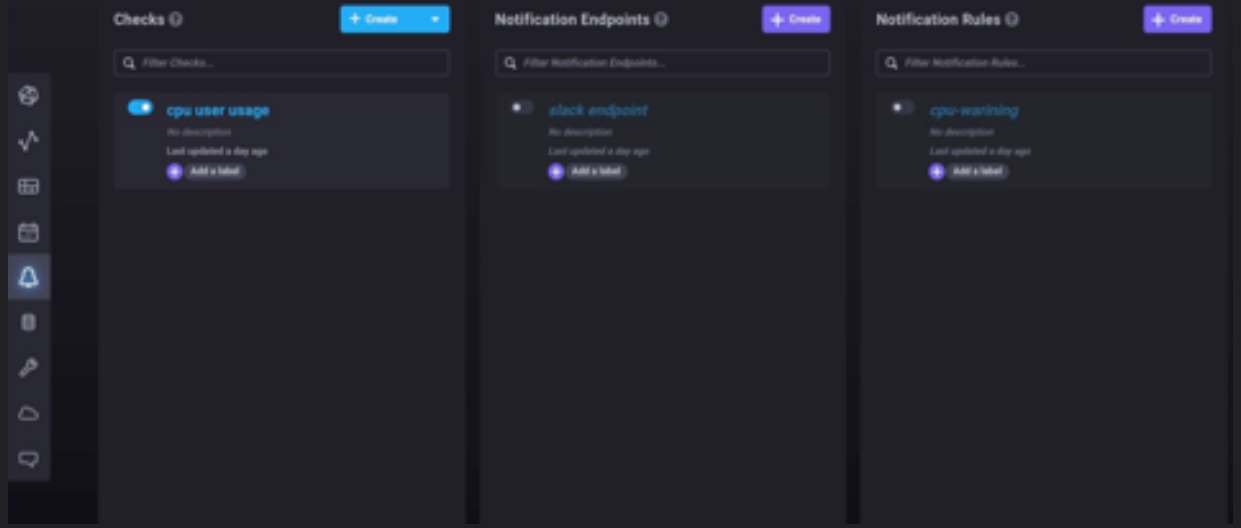
Alerts

Types of Alerts

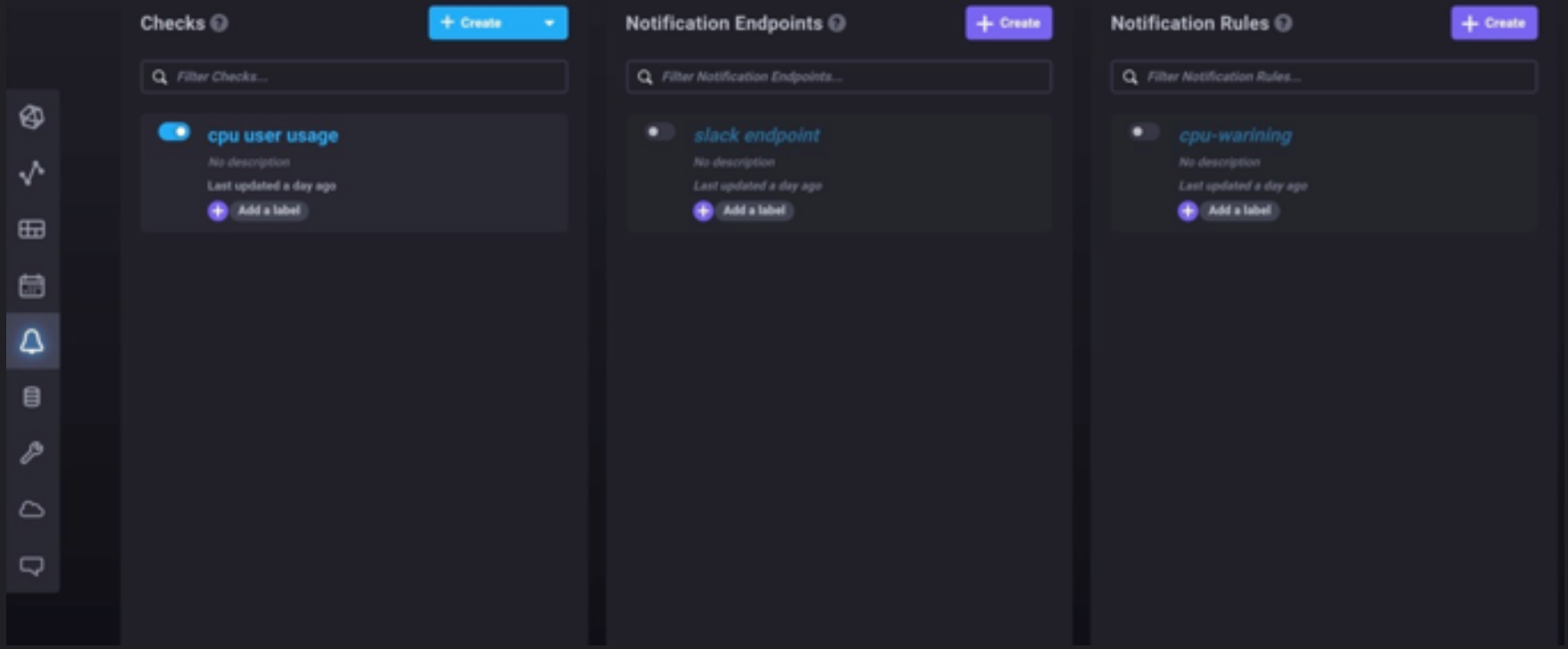
- Threshold
- Deadman

Alerts are defined as

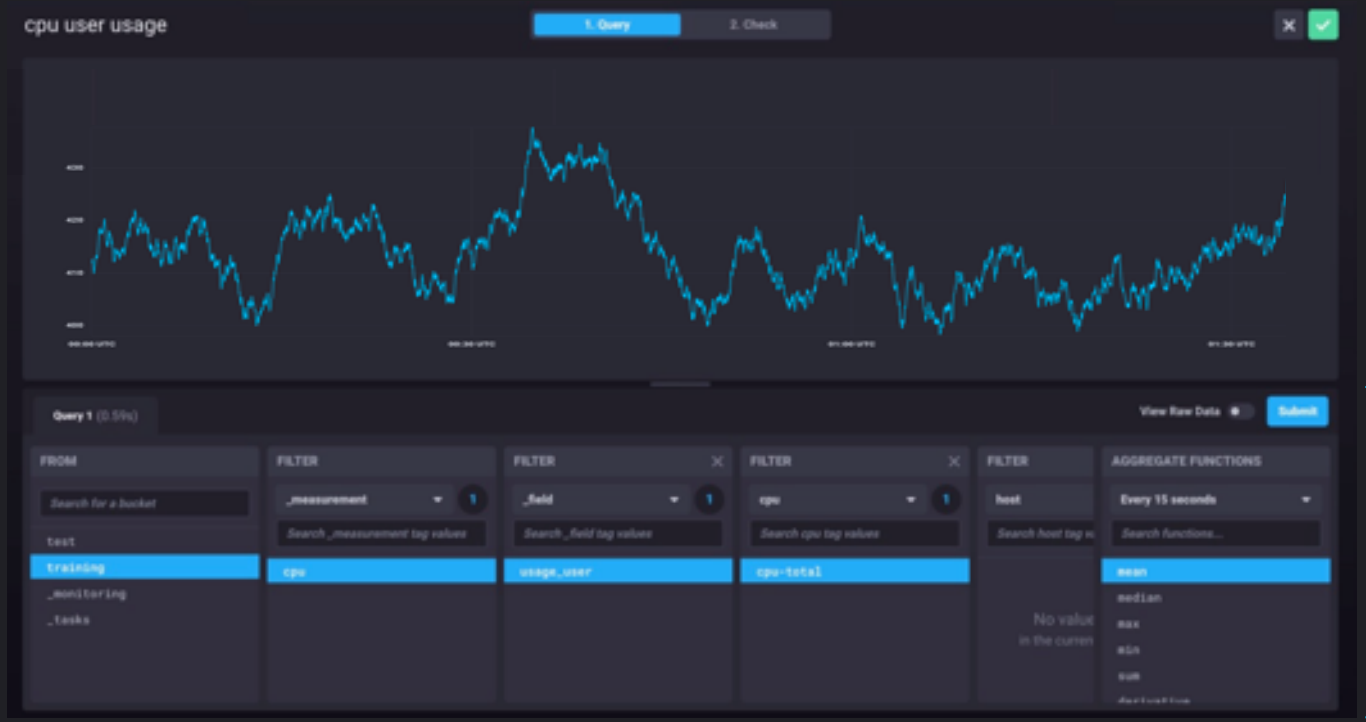
- a base query
- threshold criteria
- condition mapping
- notifications



Alerts UI

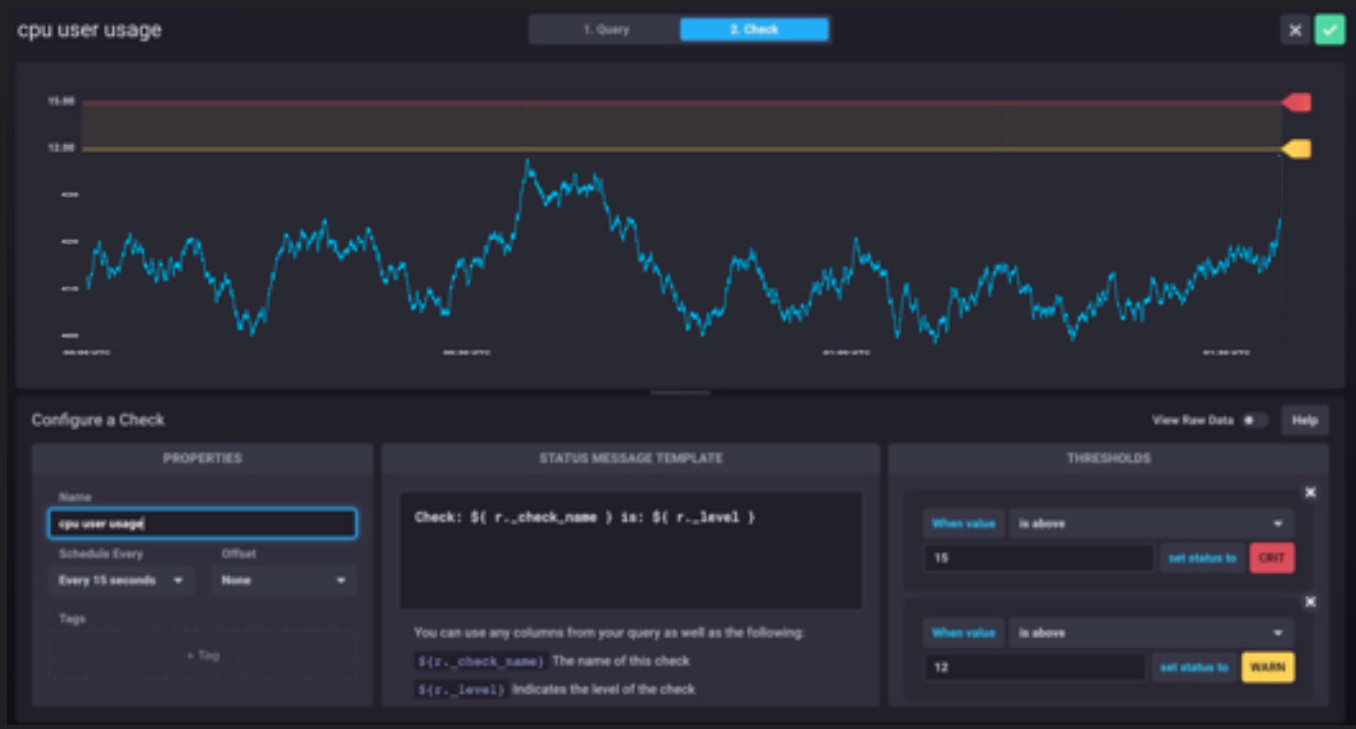

Checks
+ Create
Filter Checks...
cpu user usage
No description
Last updated a day ago
Add a label
Notification Endpoints
+ Create
Filter Notification Endpoints...
slack endpoint
No description
Last updated a day ago
Add a label
Notification Rules
+ Create
Filter Notification Rules...
cpu-warining
No description
Last updated a day ago
Add a label

Alerts UI – check view – query pane



base query

Alerts UI – check view – check pane



critic condition threshold

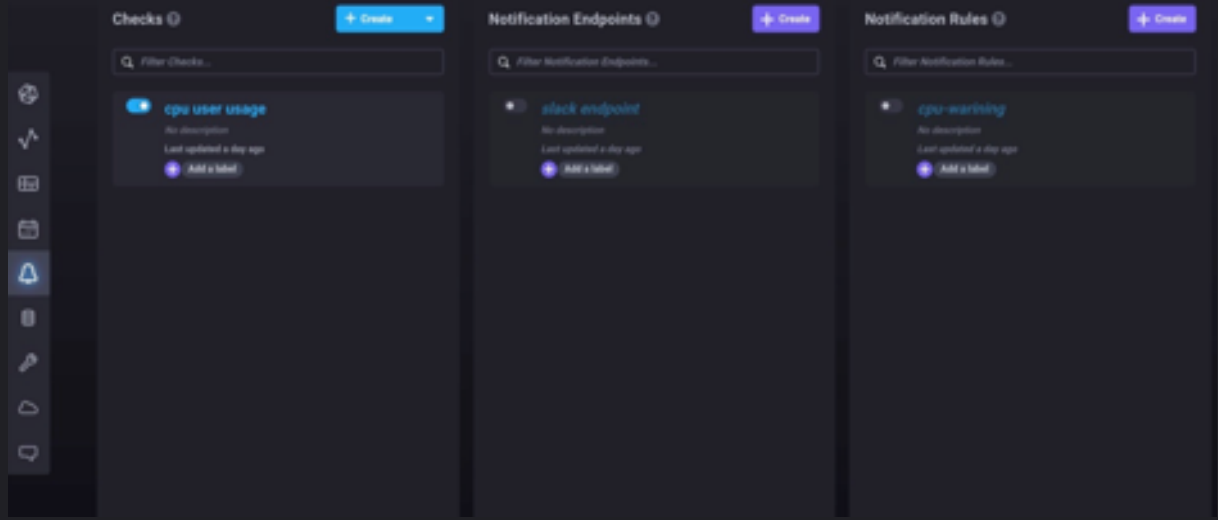
warn condition threshold

threshold criteria condition mapping

Notifications

Possibility to define

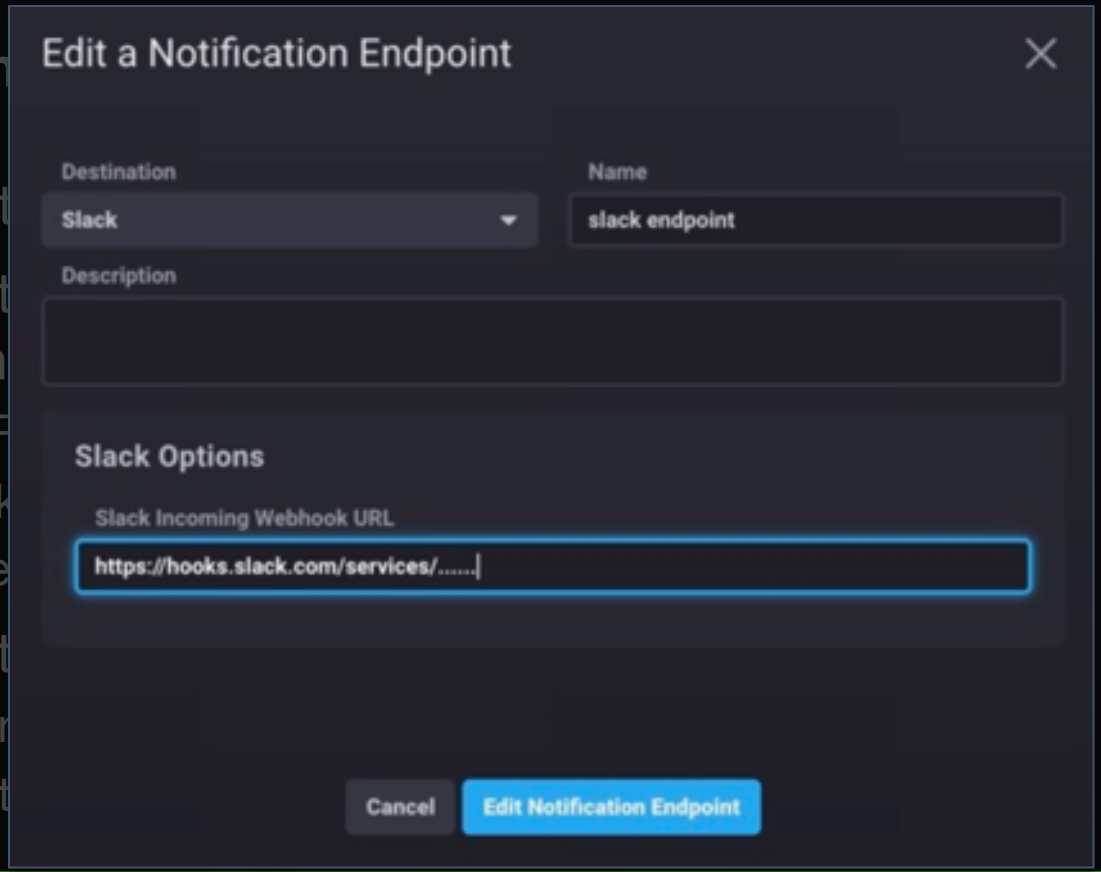
- Notification endpoints on
  - HTTP
  - Slack
  - PagerDuty
- Notification rules
  - When
  - What



Notification

Possibility t

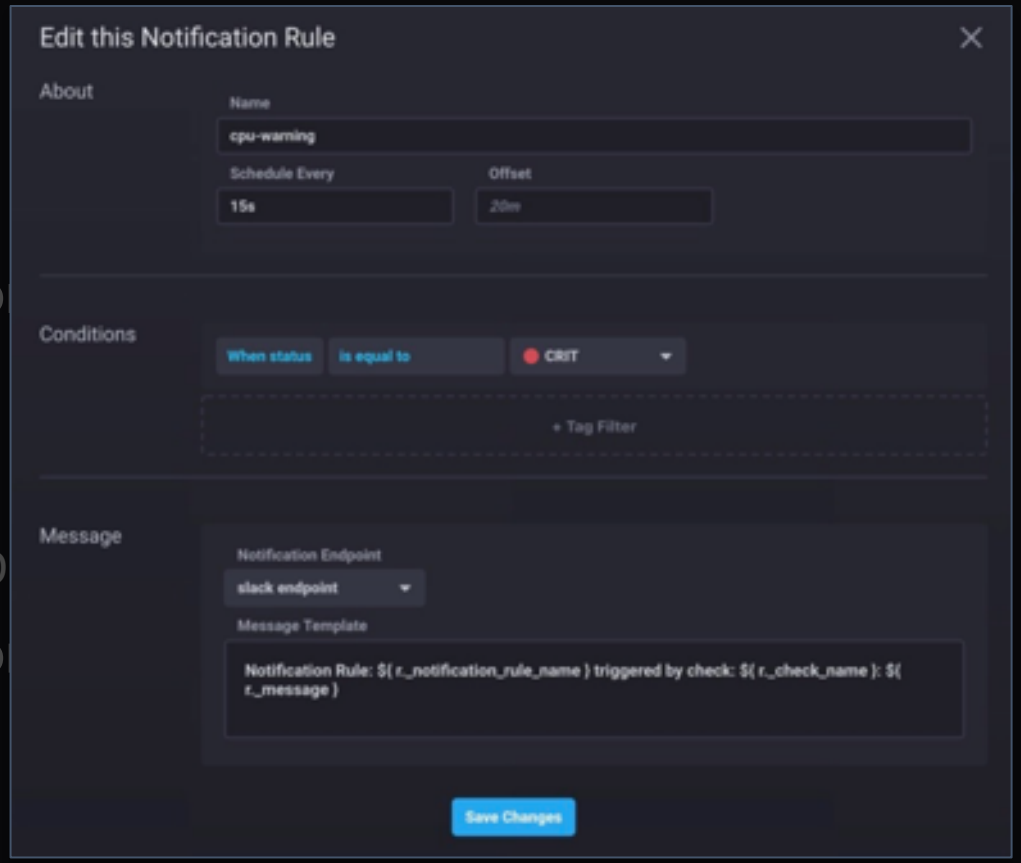
- Notificat endpoin
  - HTTP
  - Slack
  - Page
- Notificat
  - Wher
  - What

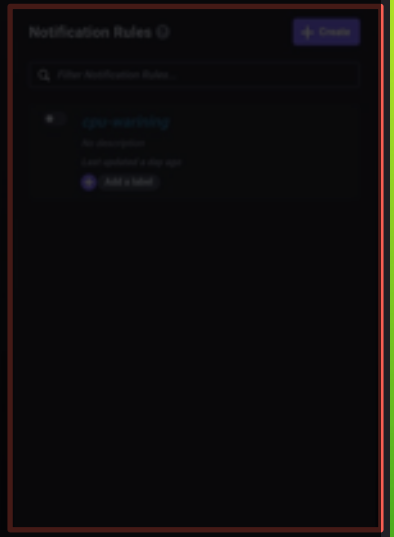


Notifications

Possibility to

- Notificatio
  endpoints
  - HTTP
  - Slack
  - PagerD
- Notificatio
  - When
  - What

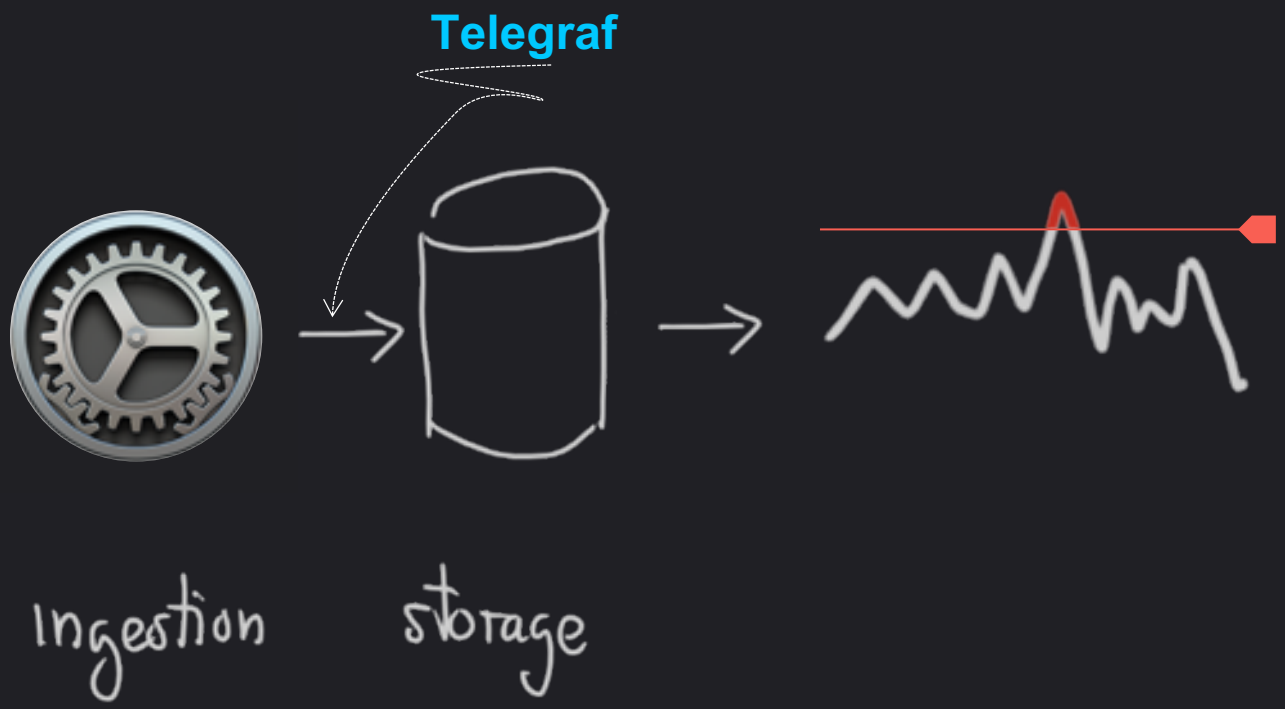




Let’s get dirty!

11

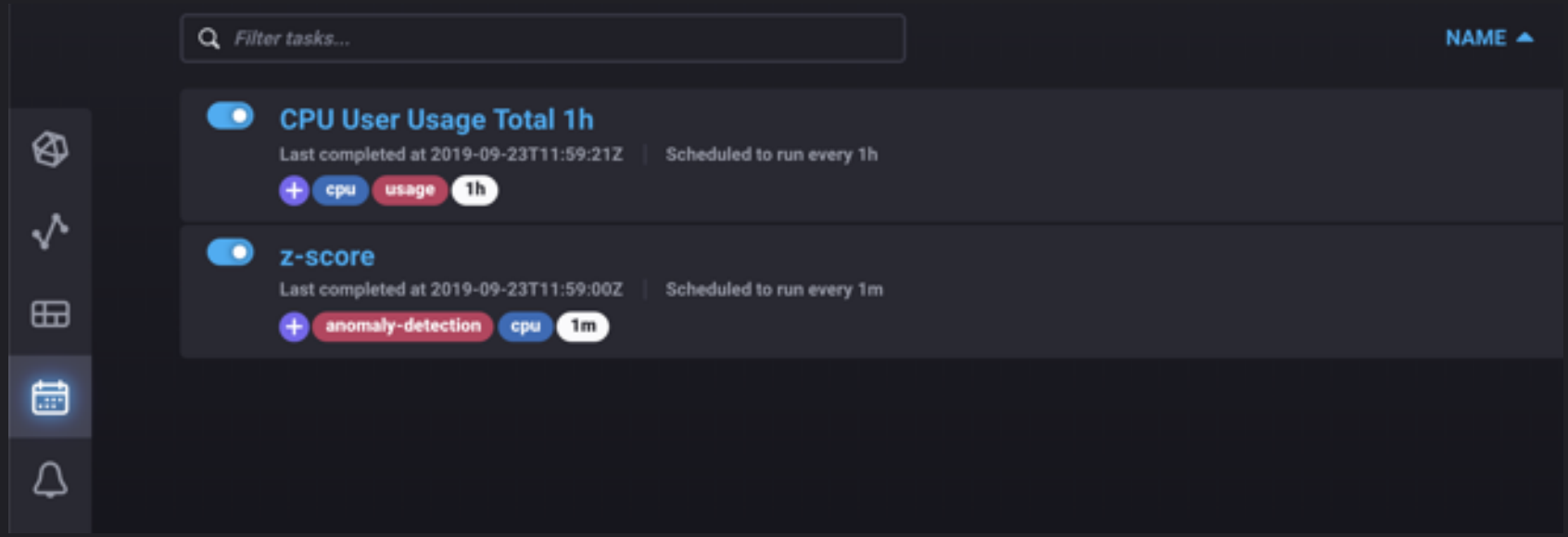
Task - Create an alert to check if the average CPU user usage



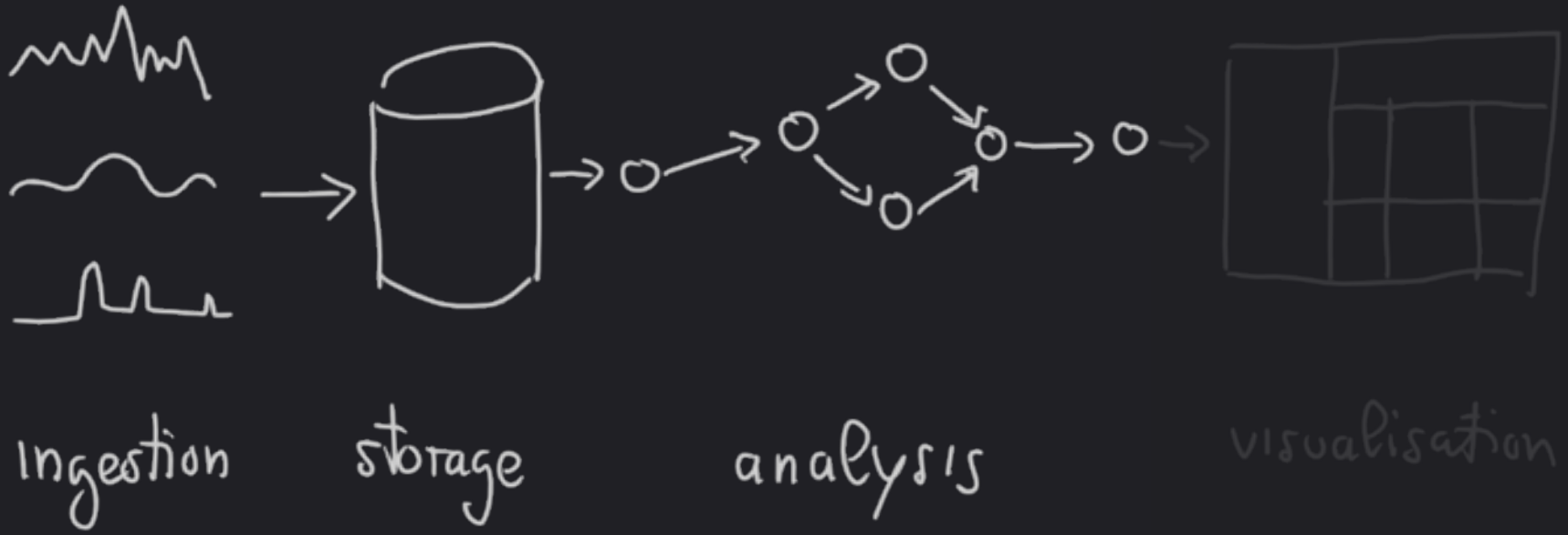
Tasks

Tasks

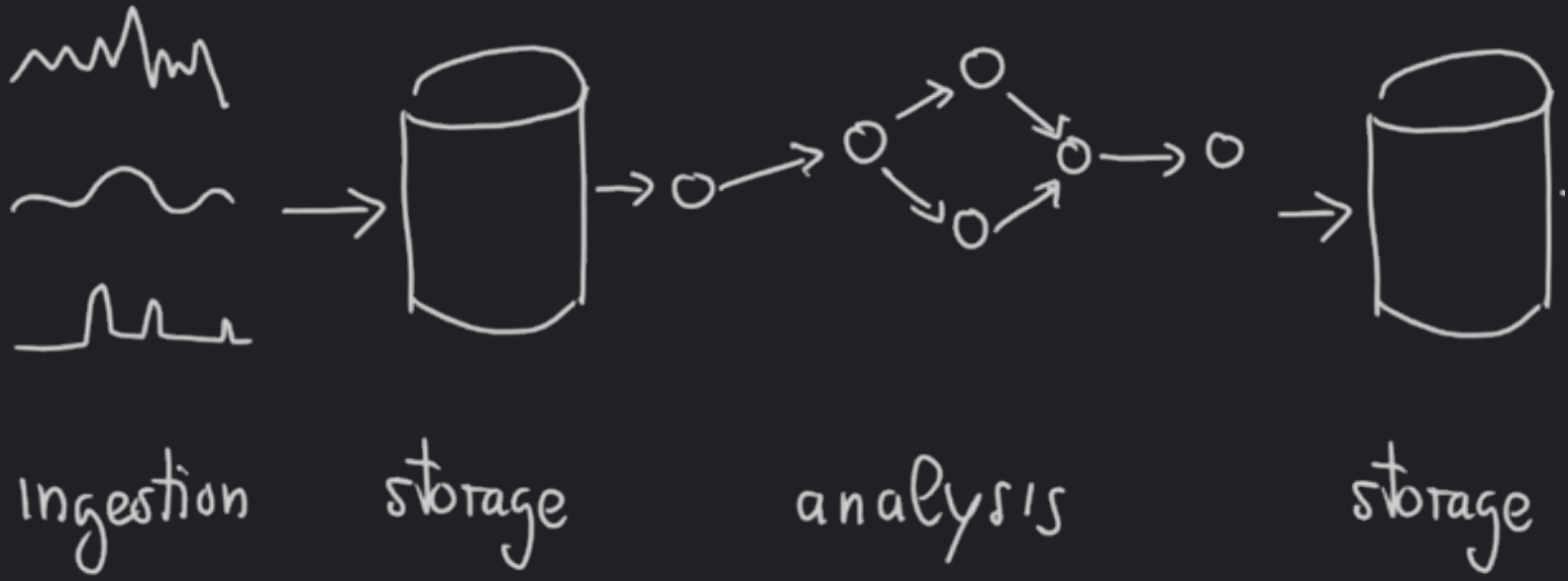
- The power of Flux for continuous processing and analysis



Recall the Data Lifecycle of this morning



Tasks continuously write back to the storage the results of the analysis



Writing into a bucket using to()

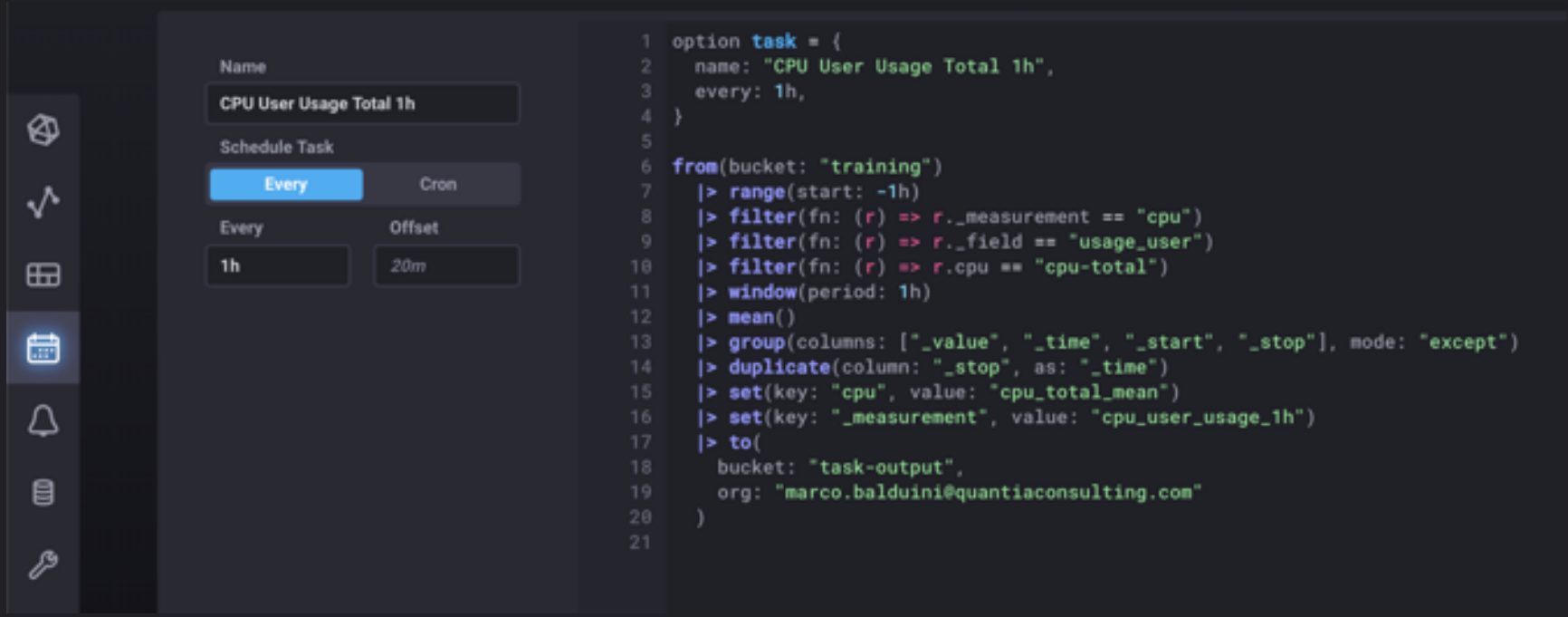
- Syntax

  to(bucket: <<name>>, org:<<organization>>)

- Example

```
from(bucket:"foo")
  |> ...
  |> to(bucket: "bar", org:"some@quantiaconsulting.com")
```

Practical example of continuously downsampling data




```
option task = {
  name: "CPU User Usage Total 1h",
  every: 1h,
}

from(bucket: "training")
  |> range(start: -1h)
  |> filter(fn: (r) => r._measurement == "cpu")
  |> filter(fn: (r) => r._field == "usage_user")
  |> filter(fn: (r) => r.cpu == "cpu-total")
  |> window(period: 1h)
  |> mean()
  |> group(columns: ["_value", "_time", "_start", "_stop"], mode: "except")
  |> duplicate(column: "_stop", as: "_time")
  |> set(key: "cpu", value: "cpu_total_mean")
  |> set(key: "_measurement", value: "cpu_user_usage_1h")
  |> to(
    bucket: "task-output",
    org: "marco.balduini@quantiaconsulting.com"
  )
```

Tasks are Written in the Flux Language

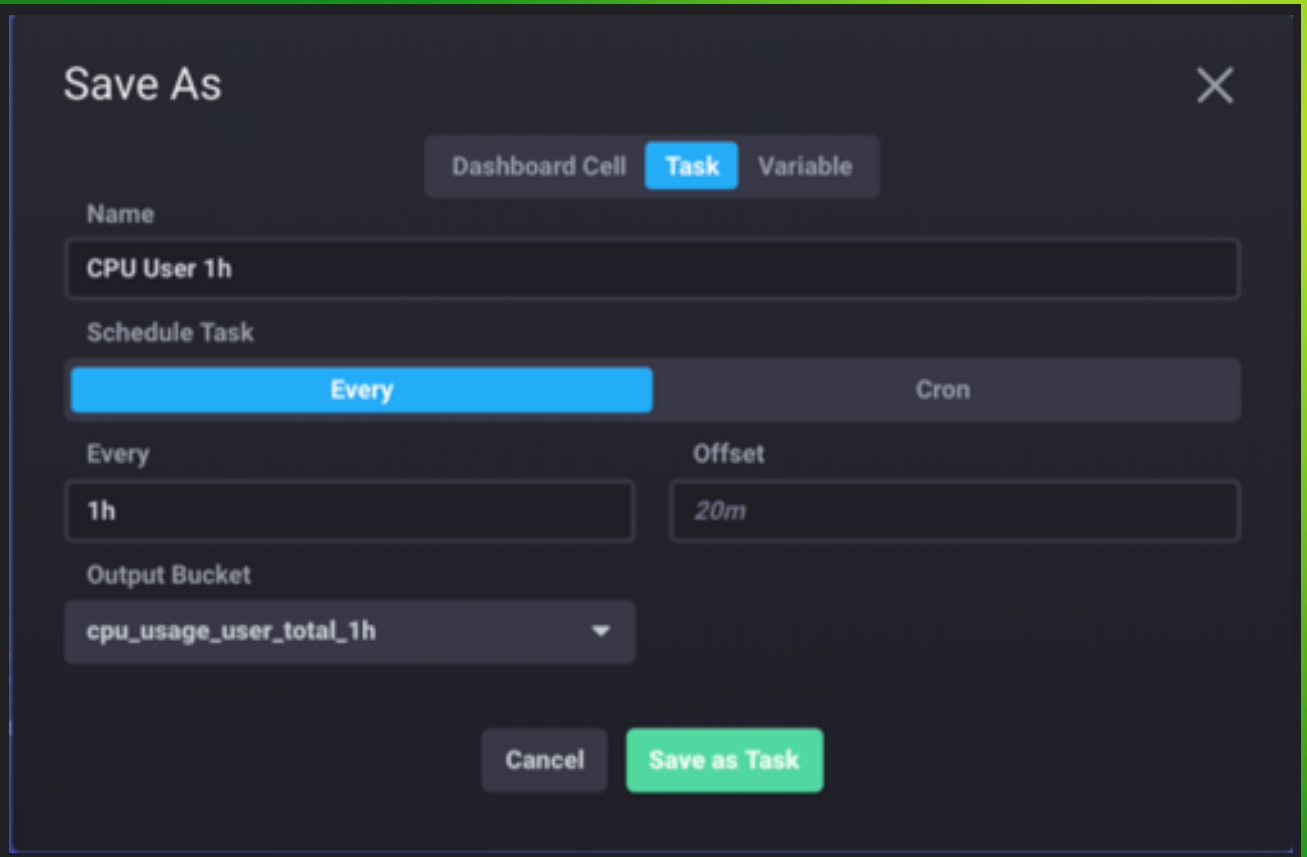
- No need to learn a separate language to write Tasks
- No need to use separate tools for debugging Tasks
- Tasks define all options within the script for portability
- Options include
  - Name
  - Run Interval
    - every
    - CRON syntax

```
option task = {
  name: "CPU User Usage Total 1h",
  every: 1h,
}
```

- Tasks are self contained and can be imported / exported

Best practice

- Prepare the query in the Data Explorer
- Save it as a Task



Let’s get dirty!

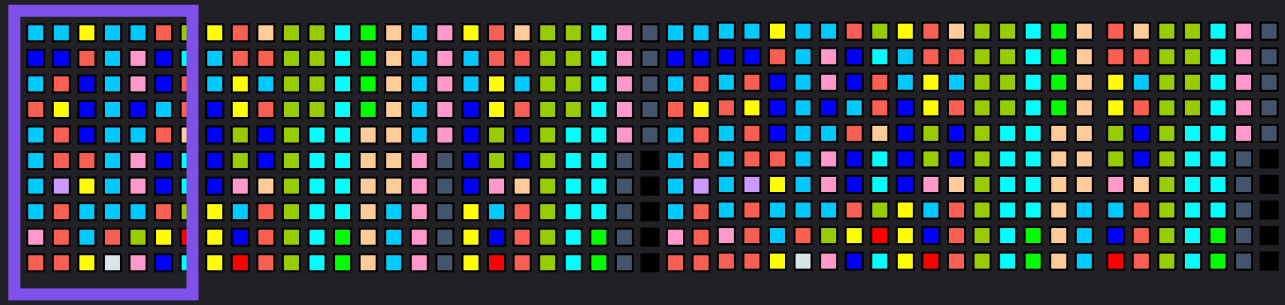
Task

- Go to the data explorer pane
- Create a task that
  - computes a time series with the mean cpu user usage in the last 10 minutes and
  - saves the result to a new bucket
- Schedule it every minute

Take home message

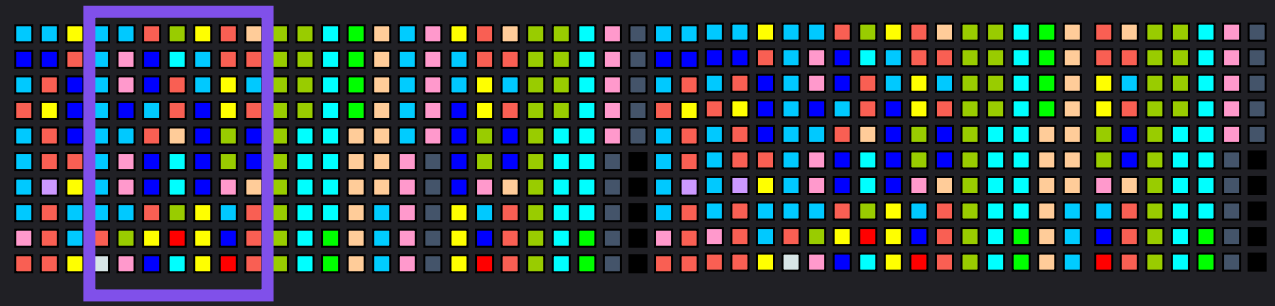
- Flux and tasks allow implementing **sliding windows** (the most common operator in stream processing)

Sliding windows



[1]

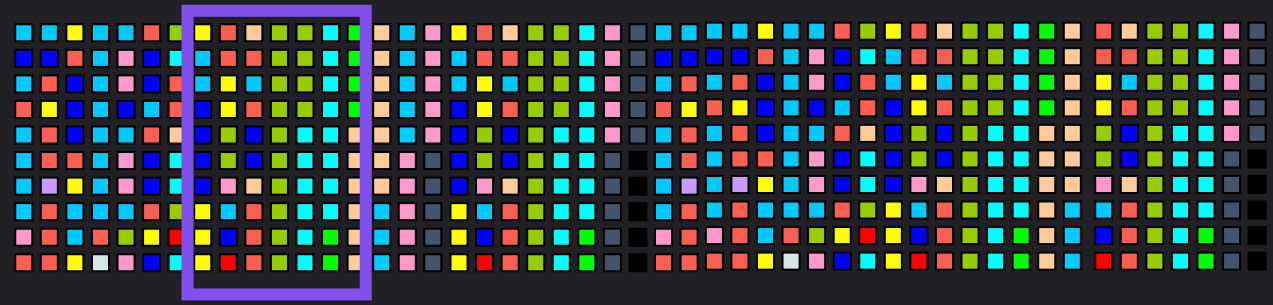
Sliding windows

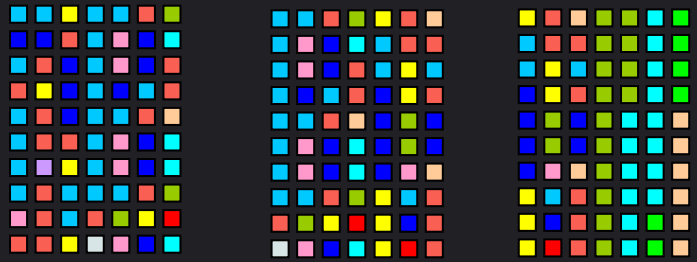


[1]

[2]

Sliding windows





[1] [2] **[3]**

Common Task Use Cases

- Downsampling data (a.k.a. Data Rollups)
- Cleaning data
- Enriching data

Alerts and Tasks

Emanuele Della Valle Prof. @ Politecnico di Milano & Partner @ Quantia Consulting
Marco Balduini Founder & CEO @ Quantia Consulting & Senior Researcher @ Politecnico di Milano