



influx/days

Alerts and Tasks

Emanuele Della Valle

Prof. @ Politecnico di Milano

Founder & Partner @ Quantia Consulting

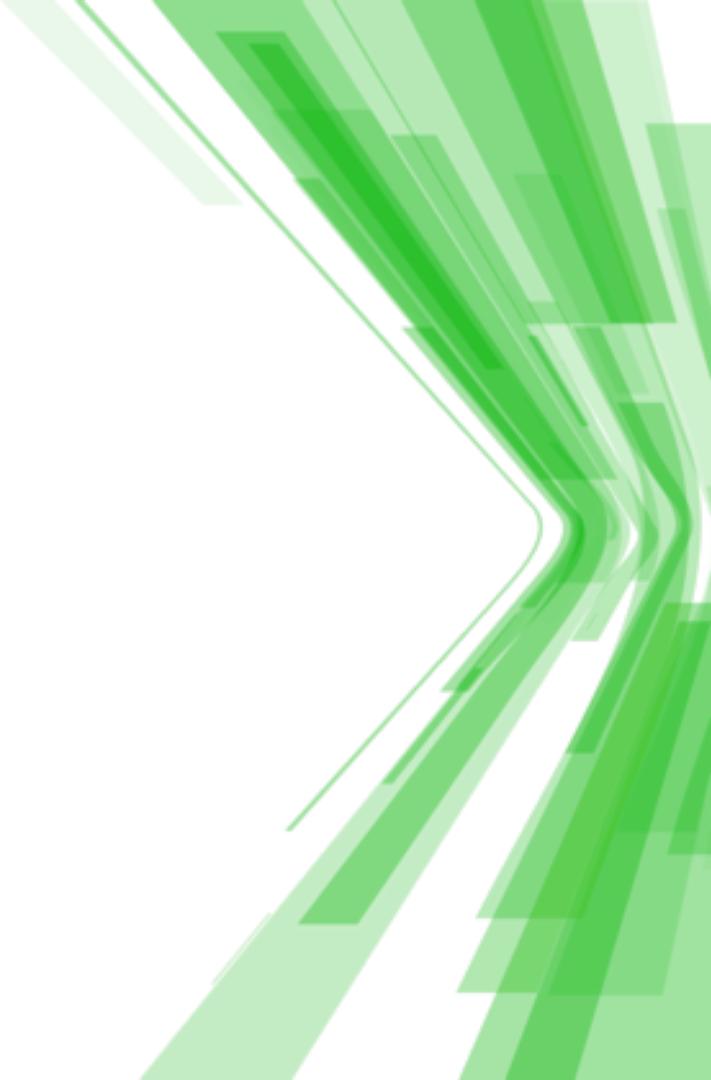
Marco Balduini

Founder & CEO @ Quantia Consulting

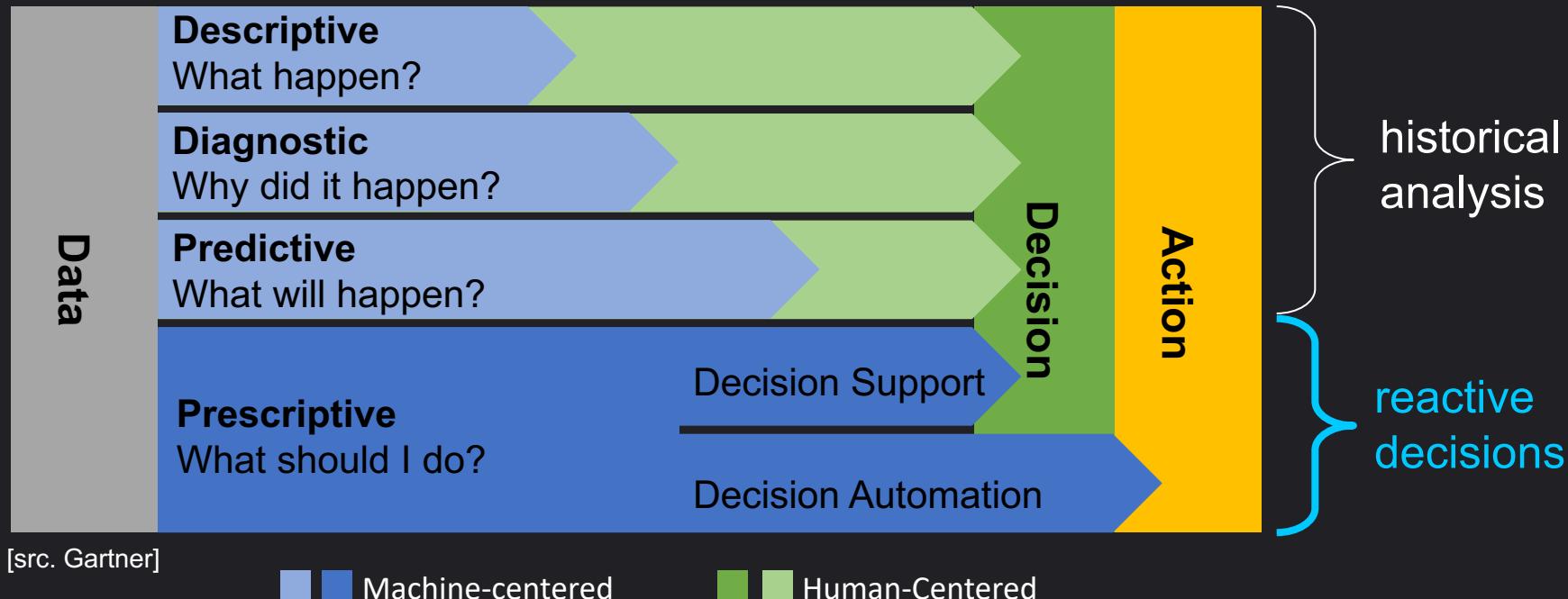
Senior Researcher @ Politecnico di Milano



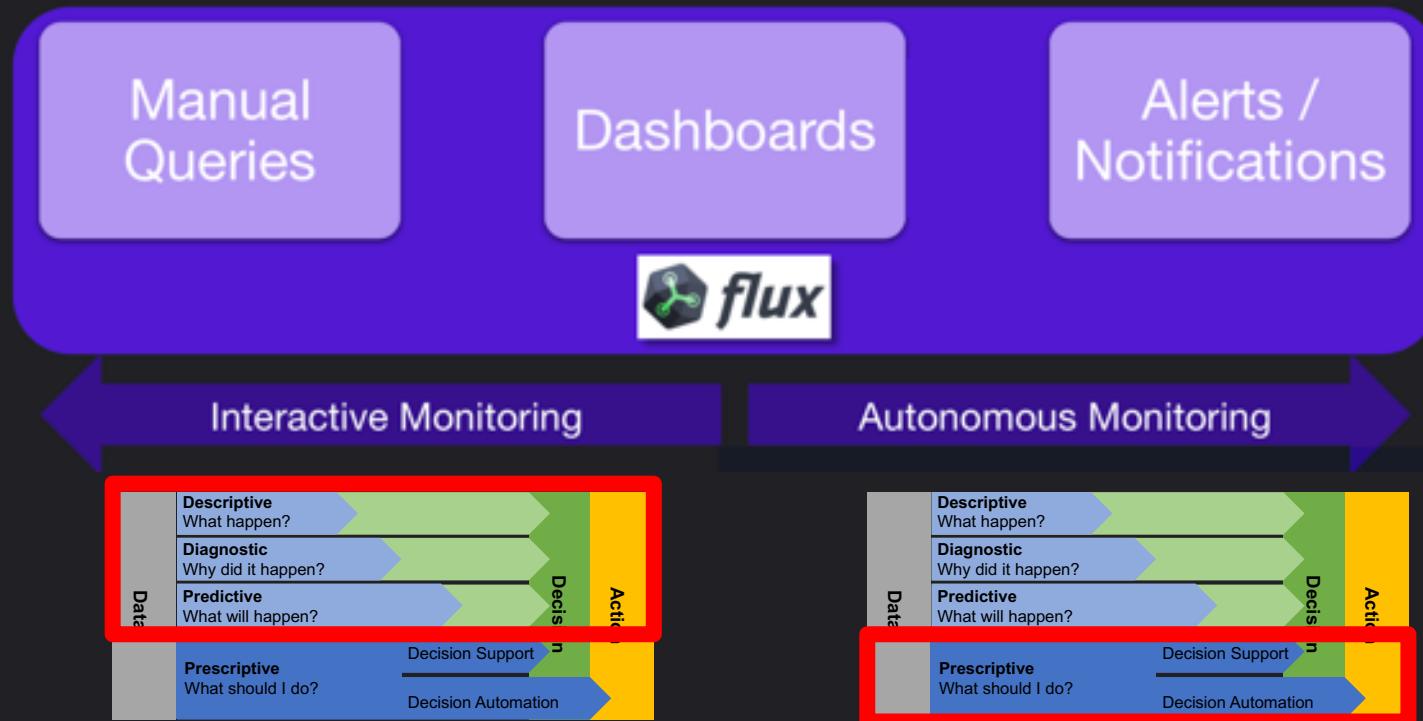
Alerts



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

The screenshot shows the Grafana interface with three main panels:

- Checks**: A list of monitoring checks. One entry is visible: "cpu user usage" (No description, Last updated a day ago). There is a "Create" button at the top.
- Notification Endpoints**: A list of notification endpoints. One entry is visible: "slack endpoint" (No description, Last updated a day ago). There is a "Create" button at the top.
- Notification Rules**: A list of notification rules. One entry is visible: "cpu-warning" (No description, Last updated a day ago). There is a "Create" button at the top.

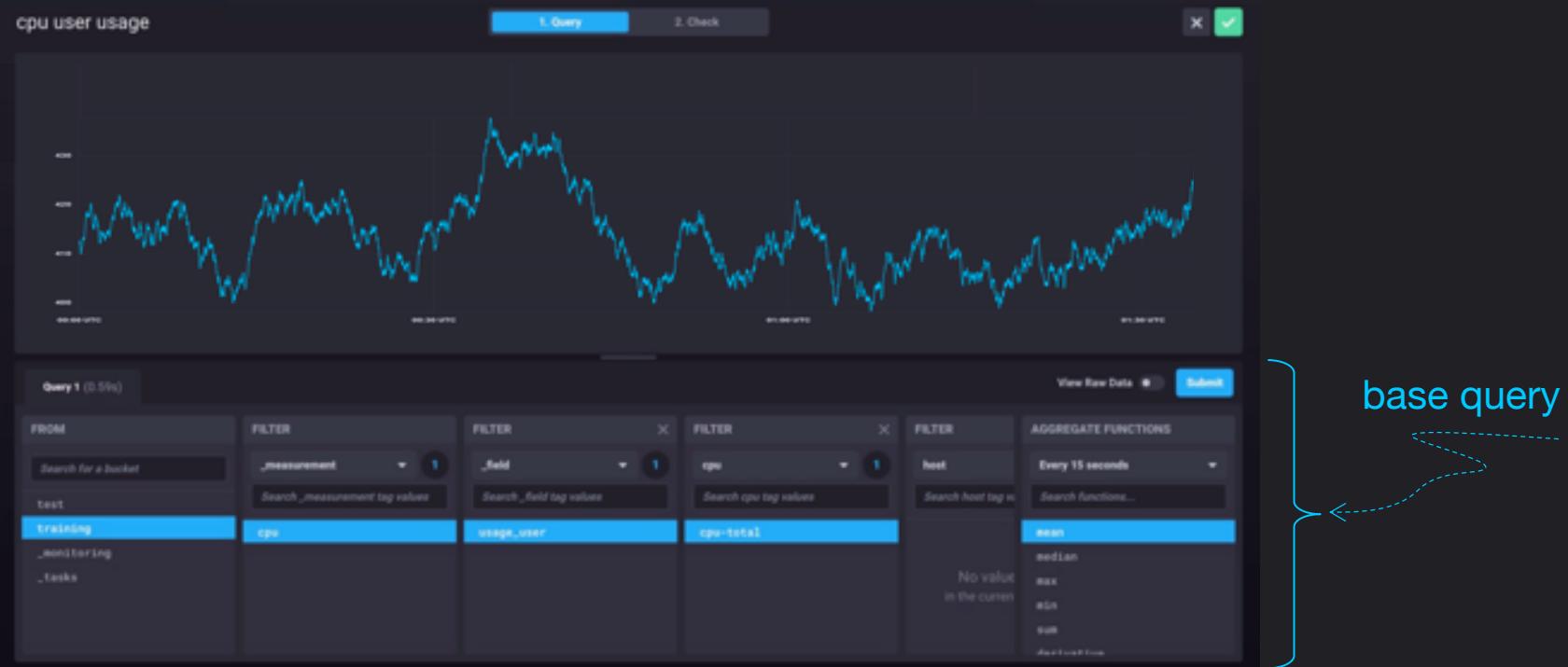
A vertical sidebar on the left contains icons for navigation and configuration, including a gear, a checkmark, a folder, a file, a bell, a wrench, a cloud, and a speech bubble.

Alerts UI

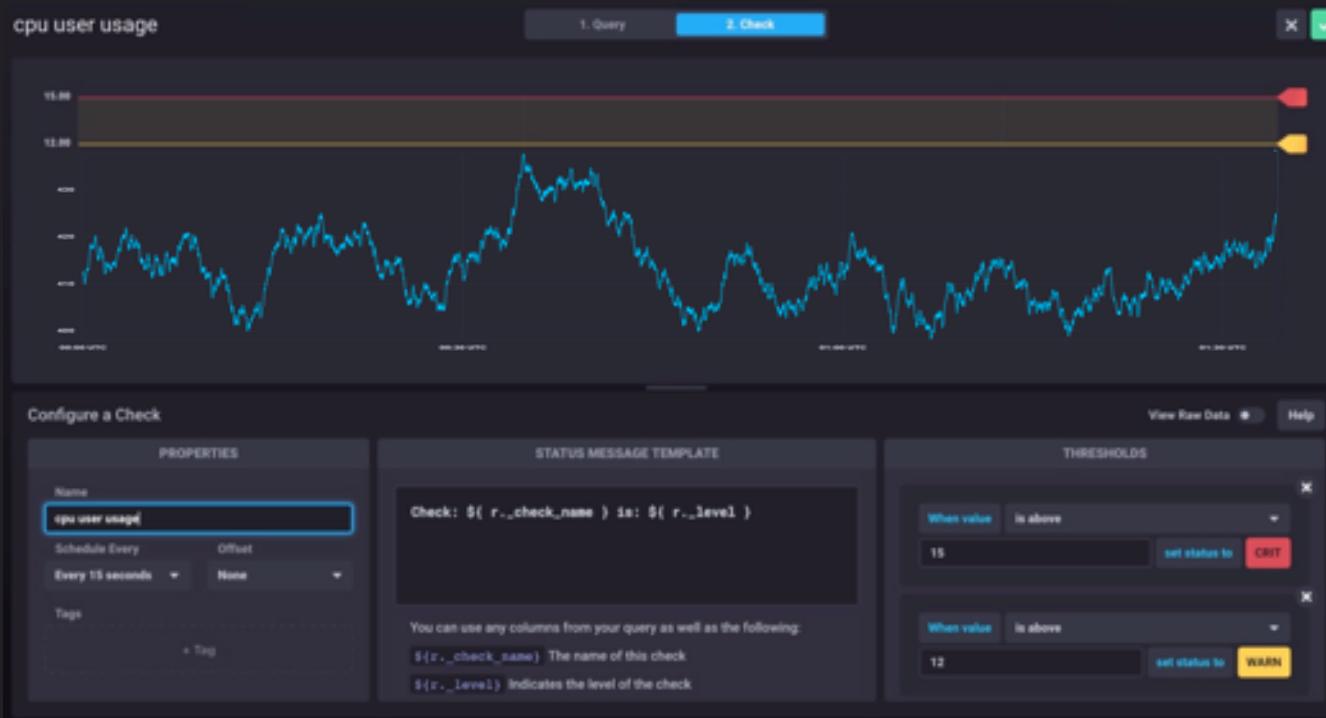
The screenshot shows the Grafana Alerts UI interface. On the left, there is a vertical sidebar with icons for Checks, Notifications, Labels, and other system settings. The main area is divided into three sections: Checks, Notification Endpoints, and Notification Rules.

- Checks:** Shows one entry: "cpu user usage" (status: green, last updated a day ago). It includes a "Filter Checks..." search bar and a "+ Create" button.
- Notification Endpoints:** Shows one entry: "slack endpoint" (status: green, last updated a day ago). It includes a "Filter Notification Endpoints..." search bar and a "+ Create" button.
- Notification Rules:** Shows one entry: "cpu-warning" (status: green, last updated a day ago). It includes a "Filter Notification Rules..." search bar and a "+ Create" button.

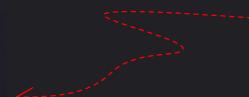
Alerts UI – check view – query pane



Alerts UI – check view – check pane



critic condition threshold



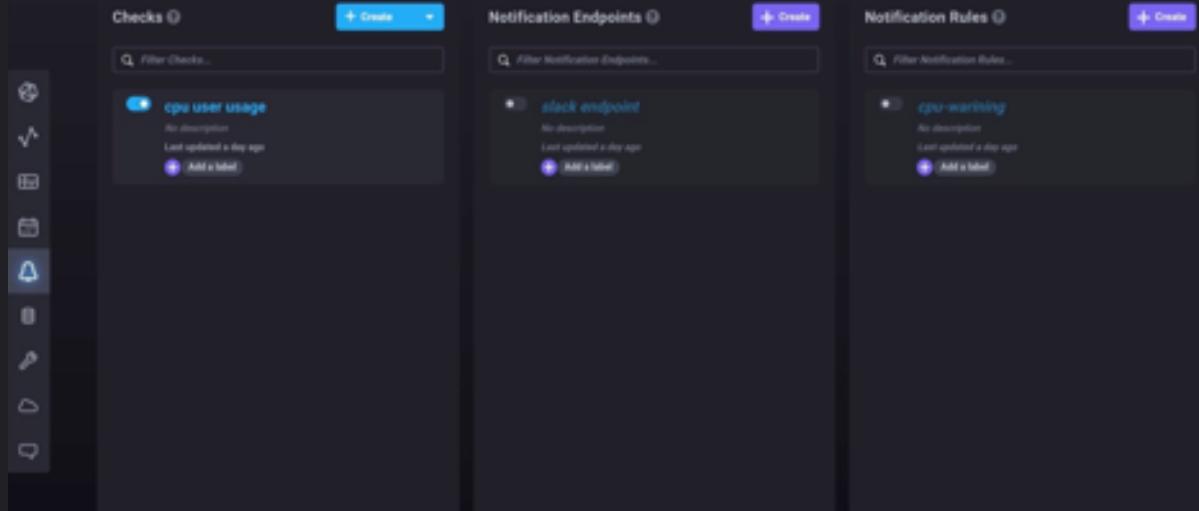
warn condition threshold

threshold criteria condition mapping

Notifications

Possibility to define

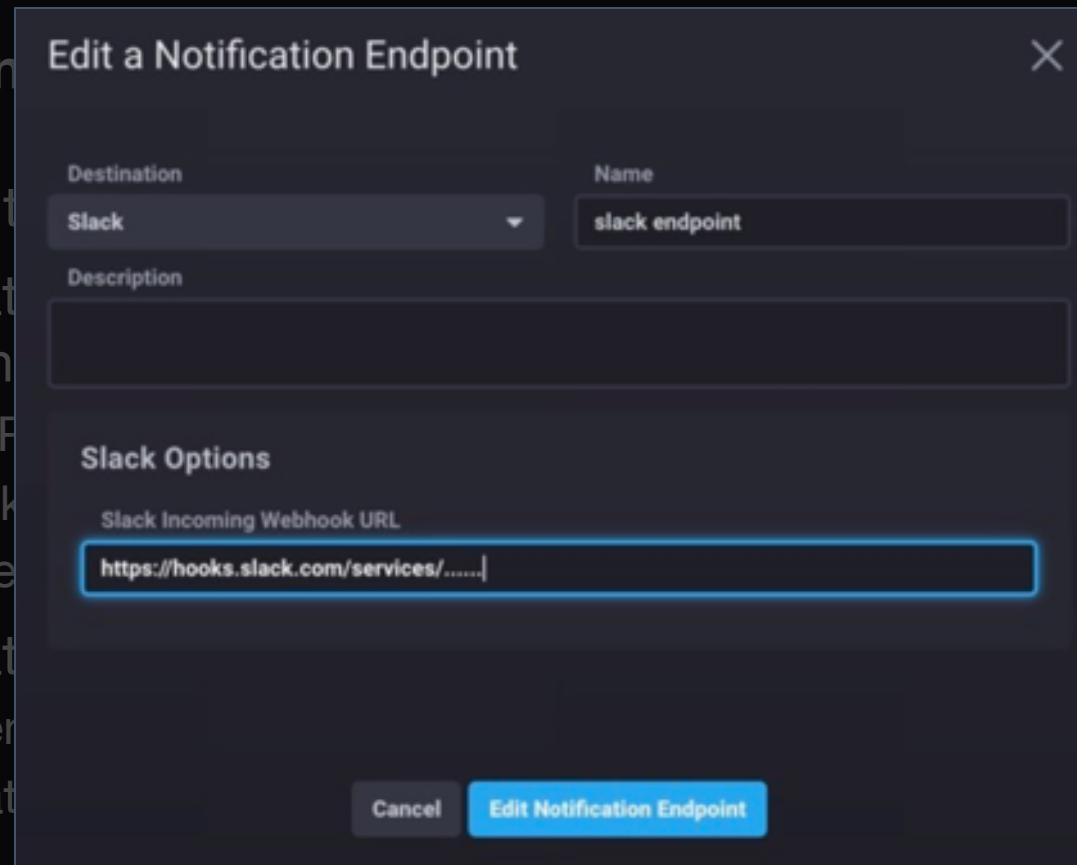
- Notification endpoints on
 - HTTP
 - Slack
 - PagerDuty
- Notification rules
 - When
 - What



Notification

Possibility to

- Notifications endpoint
 - HTTP
 - Slack
 - Page
- Notifications
 - When
 - What



Notification Rules (1) + Create

Filter Notification Rules

* gpu-warning
No description
Last updated a day ago

Add a label

Notifications

Possibility to

- **Notification endpoints**
 - HTTP
 - Slack
 - PagerD
- **Notification**
 - When
 - What

Edit this Notification Rule

About

Name: cpu-warning

Schedule Every: 15s

Offset: 20m

Conditions

When status is equal to CRIT

+ Tag Filter

Message

Notification Endpoint: slack endpoint

Message Template:

```
Notification Rule: ${r_notification_rule_name} triggered by check: ${r_check_name}: ${r_message}
```

Save Changes

Notification Rules

cpu-warning

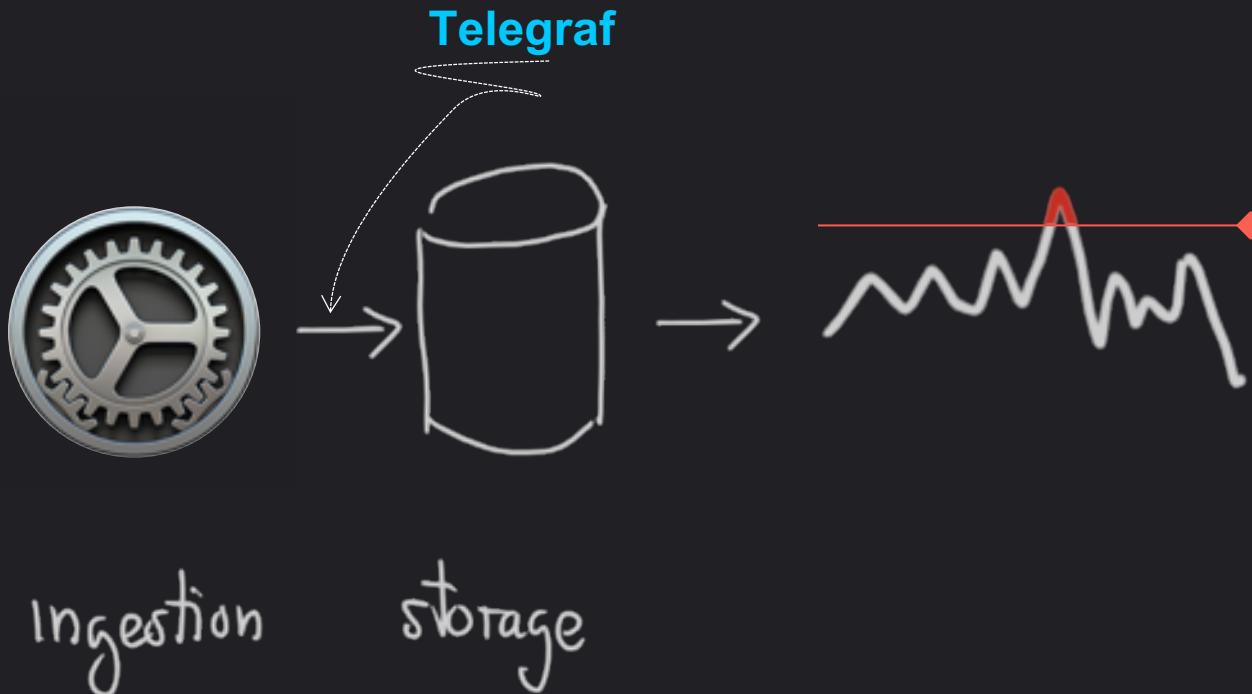
Add a label

Let's get dirty!

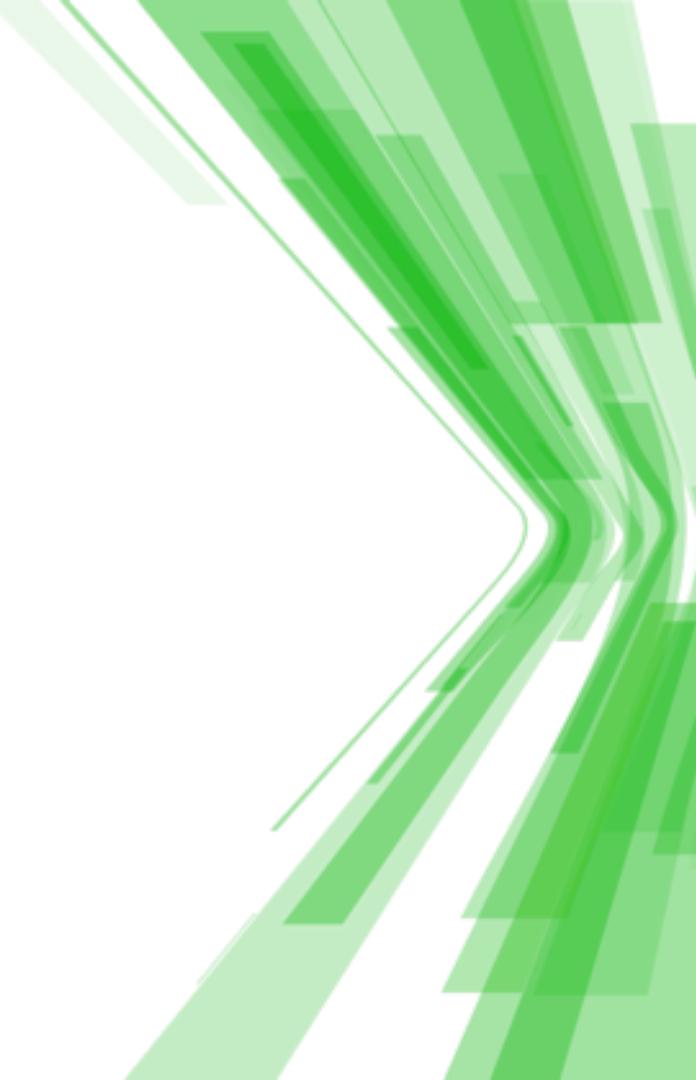


11

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



Tasks



Tasks

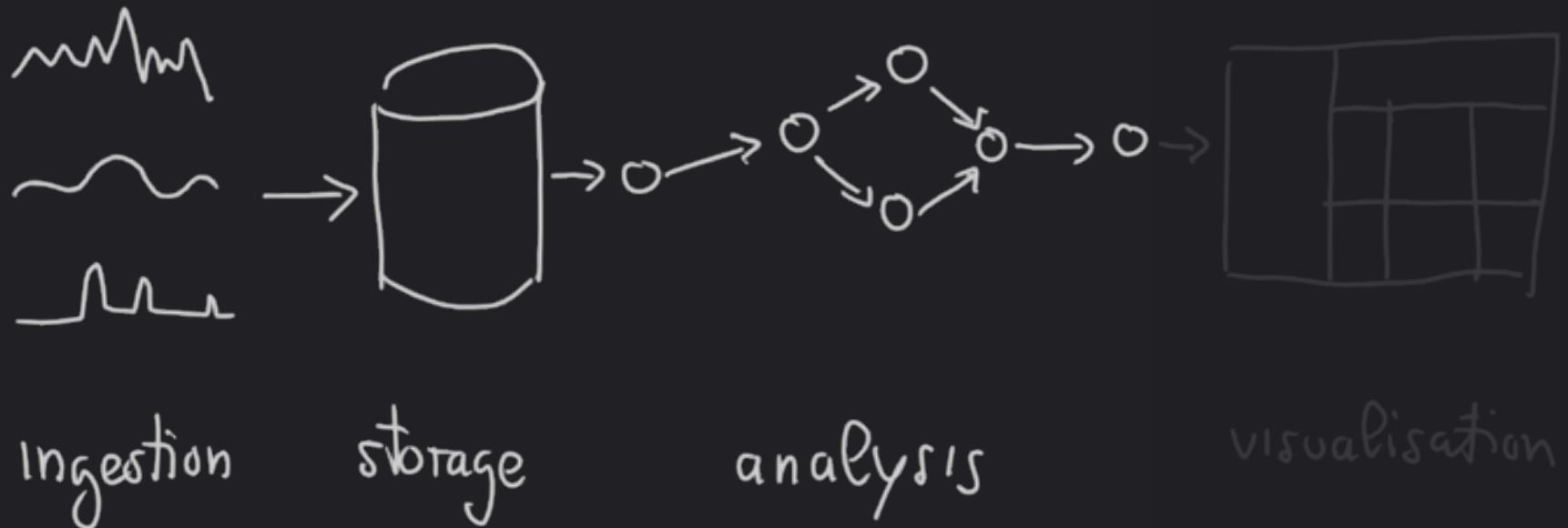
- The power of Flux for continuous processing and analysis

The screenshot shows the Grafana interface with a dark theme. On the left, there is a sidebar with five icons: a cube (Home), a line graph (Explore), a bar chart (Metrics), a calendar (Calendar), and a bell (Notifications). The main area displays two tasks in a list:

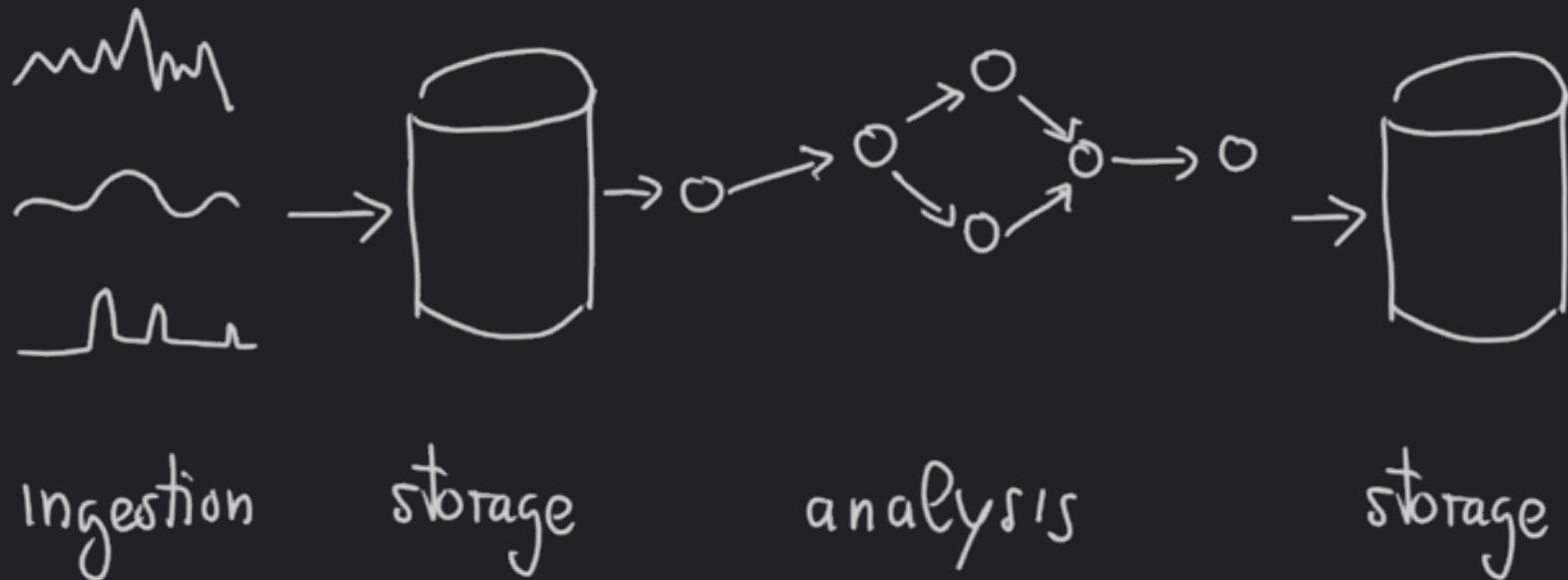
- CPU User Usage Total 1h**
Last completed at 2019-09-23T11:59:21Z | Scheduled to run every 1h
+ cpu usage 1h
- z-score**
Last completed at 2019-09-23T11:59:00Z | Scheduled to run every 1m
+ anomaly-detection cpu 1m

A search bar at the top says "Filter tasks..." and a "NAME ▲" button is on the right.

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

The screenshot shows the Grafana interface with a task configuration panel. On the left, there is a sidebar with several icons: a gear, a checkmark, a calendar, a bell, a document, and a key. The main panel has the following fields:

- Name:** CPU User Usage Total 1h
- Schedule Task:** Every (selected), Cron
- Every:** 1h
- Offset:** 20m

```
1 option task = {
2   name: "CPU User Usage Total 1h",
3   every: 1h,
4 }
5
6 from(bucket: "training")
7 |> range(start: -1h)
8 |> filter(fn: (r) => r._measurement == "cpu")
9 |> filter(fn: (r) => r._field == "usage_user")
10 |> filter(fn: (r) => r.cpu == "cpu-total")
11 |> window(period: 1h)
12 |> mean()
13 |> group(columns: ["_value", "_time", "_start", "_stop"], mode: "except")
14 |> duplicate(column: "_stop", as: "_time")
15 |> set(key: "cpu", value: "cpu_total_mean")
16 |> set(key: "_measurement", value: "cpu_user_usage_1h")
17 |> to(
18   bucket: "task-output",
19   org: "marco.balduini@quantiaconsulting.com"
20 )
21
```

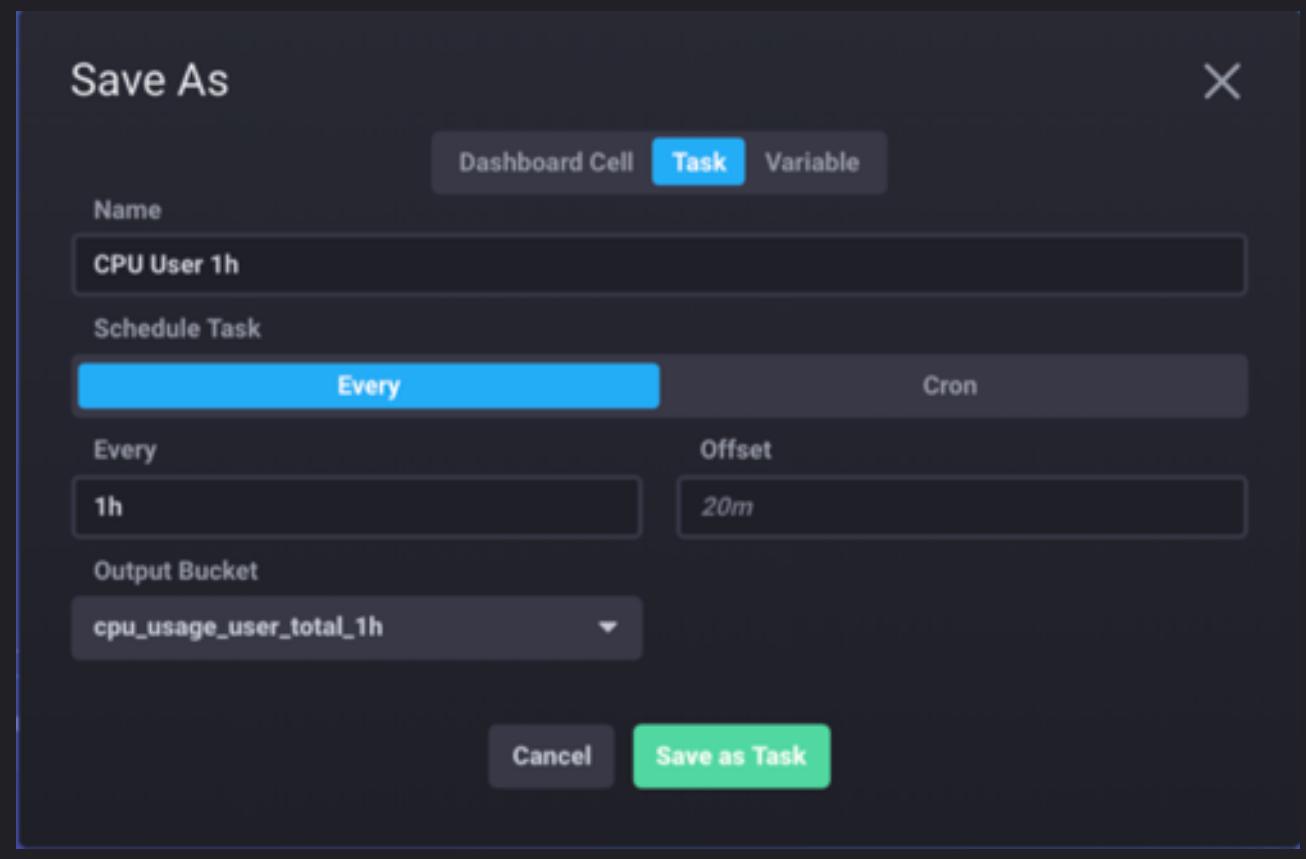
Tasks are Written in the Flux Language

- 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
- Tasks are self contained and can be imported / exported

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

Best practice

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



Let's get dirty!



12

Task

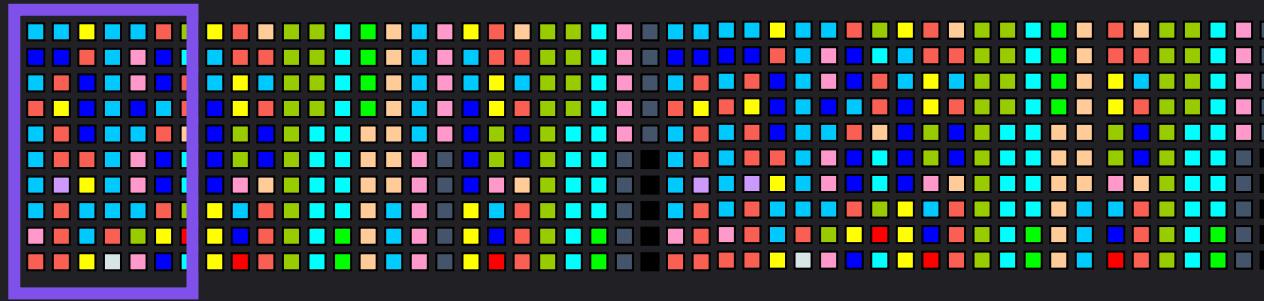
- Go to the data explorer pane
- Create a task that
 - computes a time series with the mean cpu user usage in the last 10 minutes and
 - saves the result to a new bucket
- Schedule it every minute

Take home message

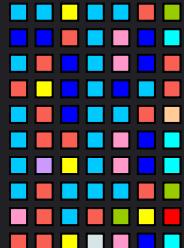
- Flux and tasks allow implementing **sliding windows** (the most common operator in stream processing)



Sliding windows

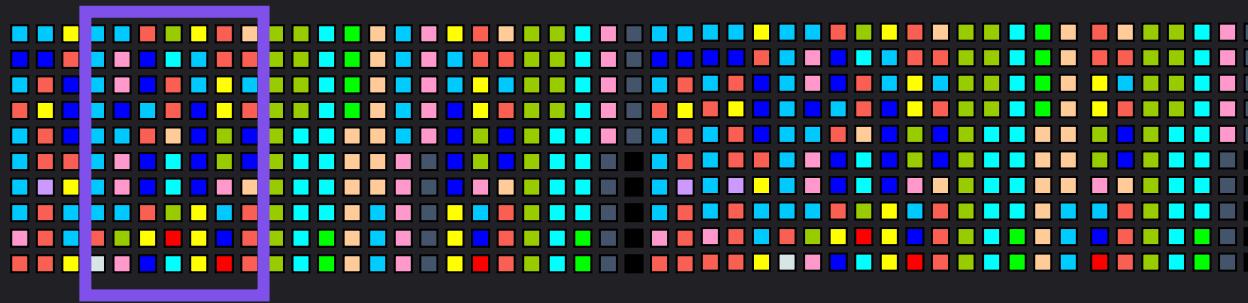


E. Della Valle



[1]

Sliding windows

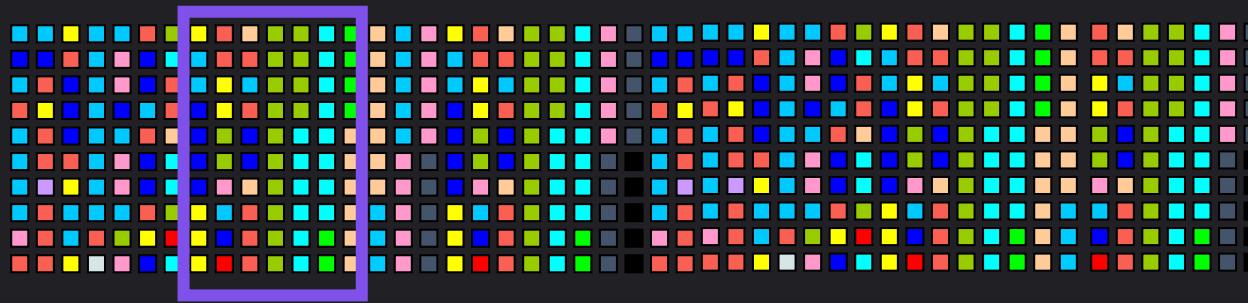


E. Della Valle

[1]

[2]

Sliding windows



E. Della Valle



[1]

[2]

[3]

Common Task Use Cases

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



influx/days



Alerts and Tasks

Emanuele Della Valle Prof. @ Politecnico di Milano & Partner @ Quantia Consulting

Marco Balduini Founder & CEO @ Quantia Consulting & Senior Researcher @ Politecnico di Milano