NETDATA

Practical AI with Machine Learning for Observability in Netdata

Costa Tsaousis

About Netdata

- Born out of a need

  While migrating a large infra from on-prem to cloud, we were facing unexplainable issues - existing monitoring solutions (open-source and commercial) failed to diagnose, or even surface.

- Born out of curiosity

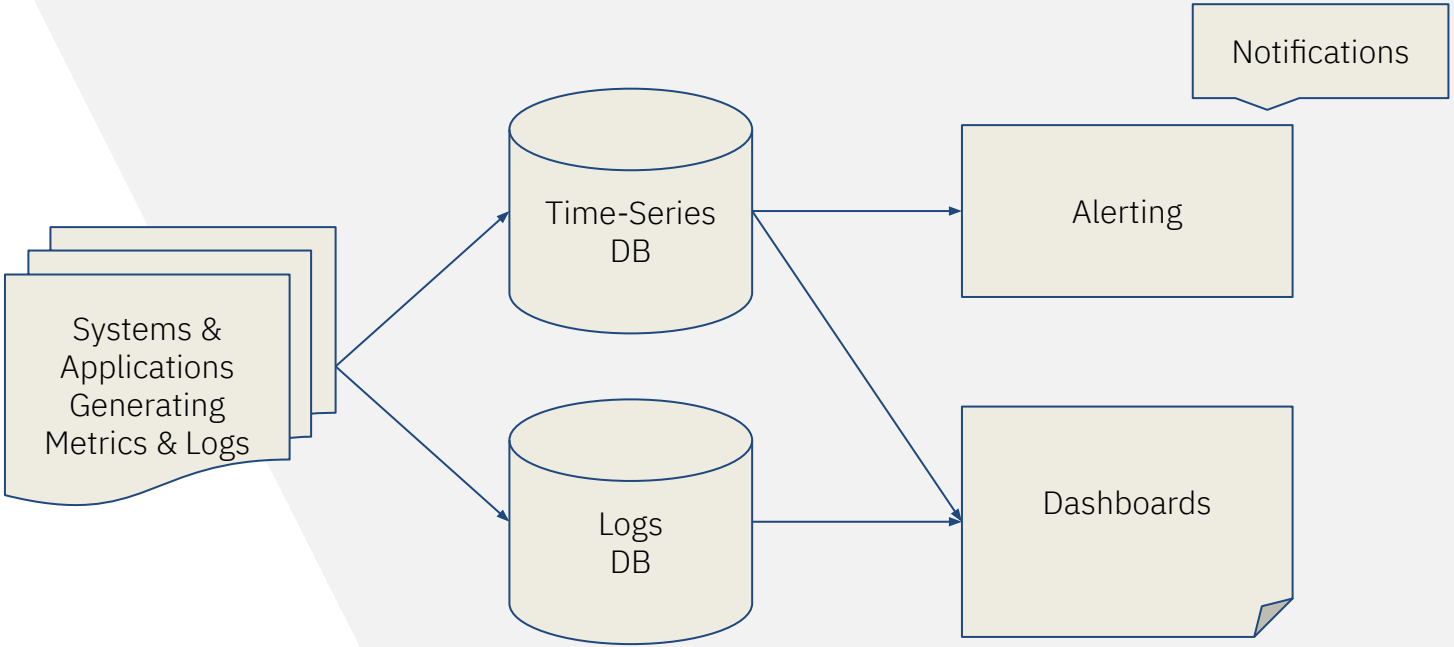
  Experimented to understand if and how a monitoring solution could be real-time, high-resolution, easier, simpler and out of the box.

- Born on GitHub, open-source from the very beginning

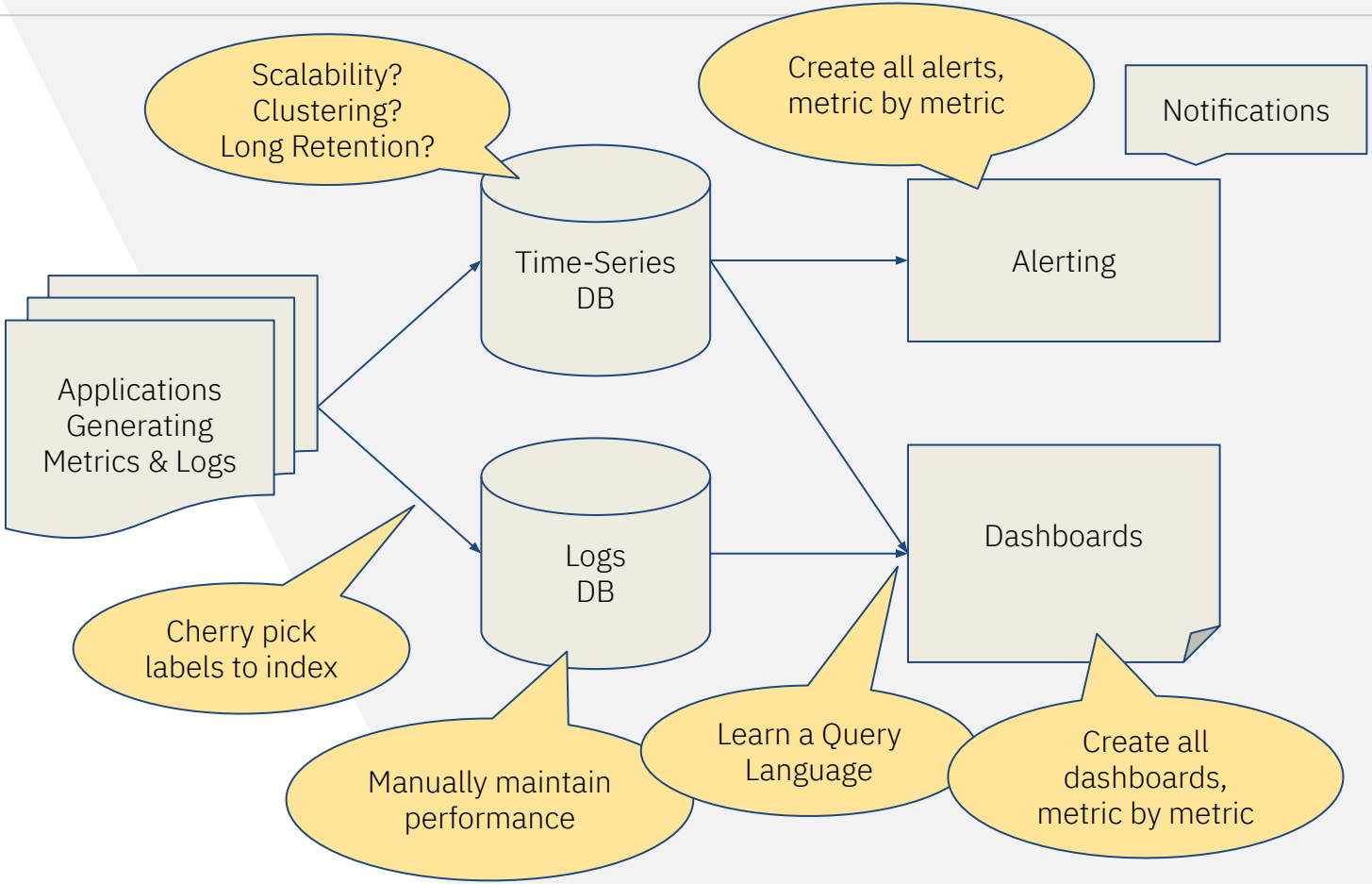
  The love and adoption by the community gave substance and a future to Netdata.

GitHub URL:

How it is different - the traditional way



The traditional monitoring pipeline



The result of the traditional pipeline

Observability
is never good,
is never
comprehensive,
is never ending!

- Requires skills.
  Deep understanding of all metrics.
  Query Languages.
  Good understanding of all technologies involved.
- A lot of moving parts.
  Database servers, application servers, integrations, protocols, statistical skills.
- Usually reflects the skills and the discipline of the engineers who set it up.
- Follows a development process, making troubleshooting slow.

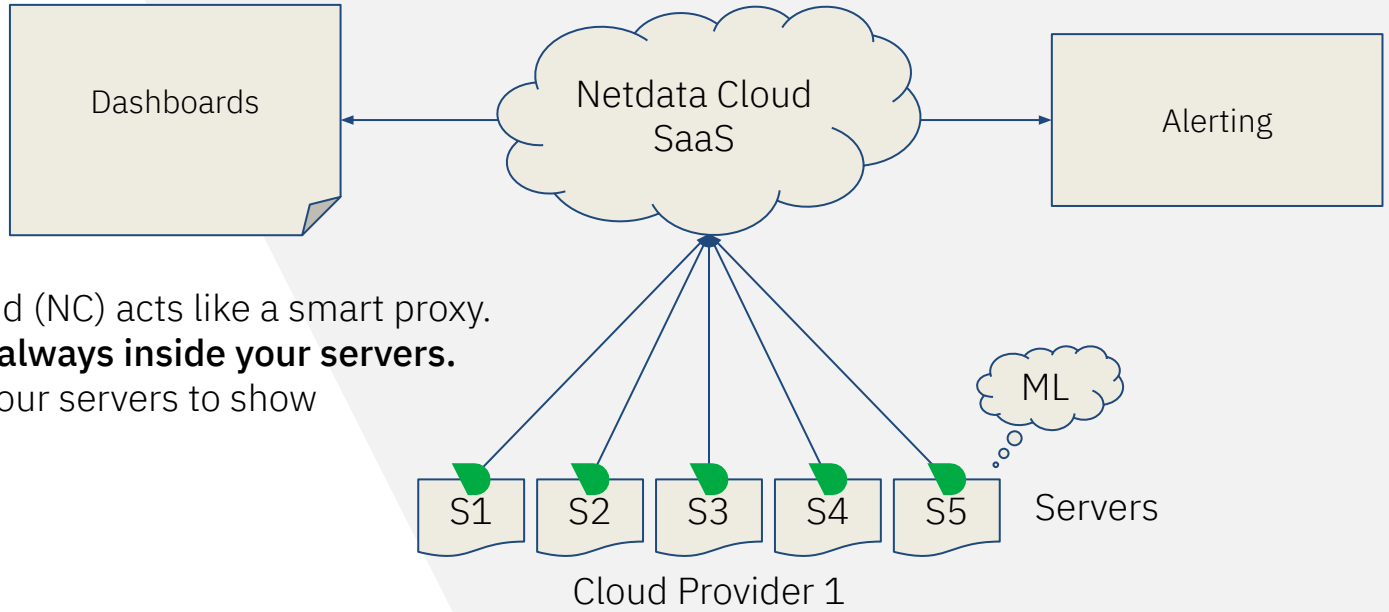
The Netdata way

The Open-Source Netdata Agent is a monitoring in a box

- AUTO-DISCOVER & COLLECT
  800+ integrations supported
- STORE
  Embedded high performance time-series db
- CHECK
  Predefined, custom alerts and notifications
- LEARN
  Machine Learning on every metric individually
- QUERY & SCORE (API)
- VISUALIZE
- STREAM & REPLICATE

The Netdata way - distributed monitoring

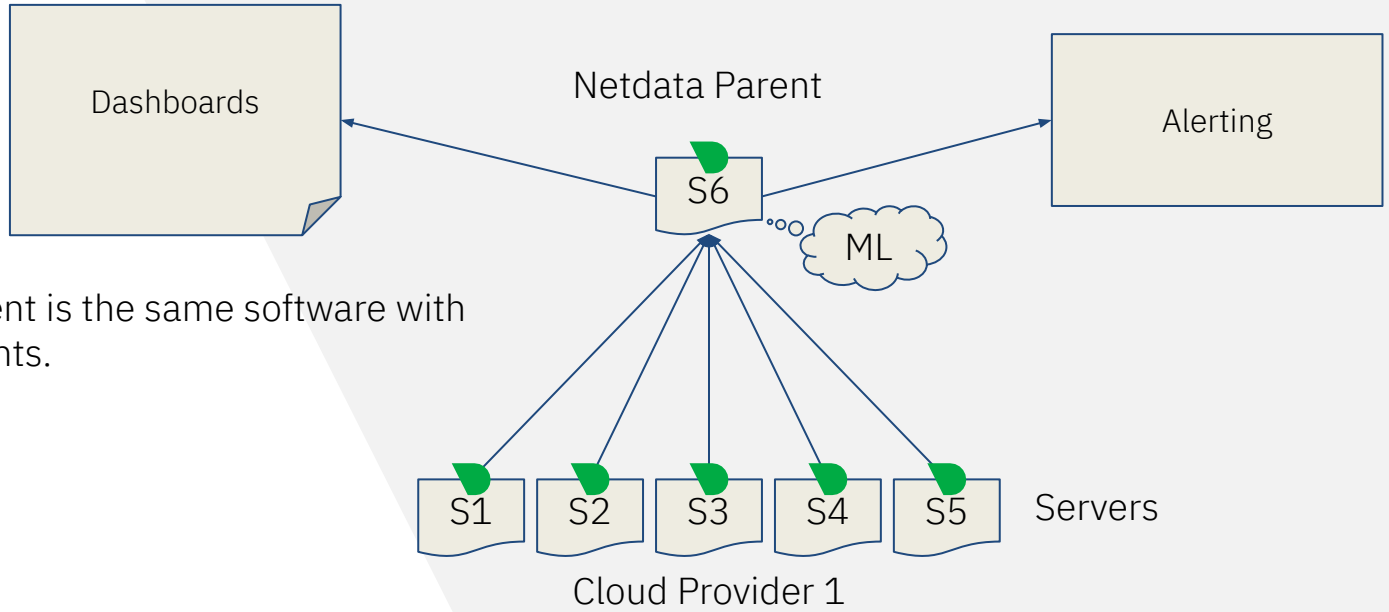
Install the agent on all your systems, Use Netdata Cloud to access your infra.



Netdata Cloud (NC) acts like a smart proxy.
Your data is always inside your servers.
NC queries your servers to show dashboards.

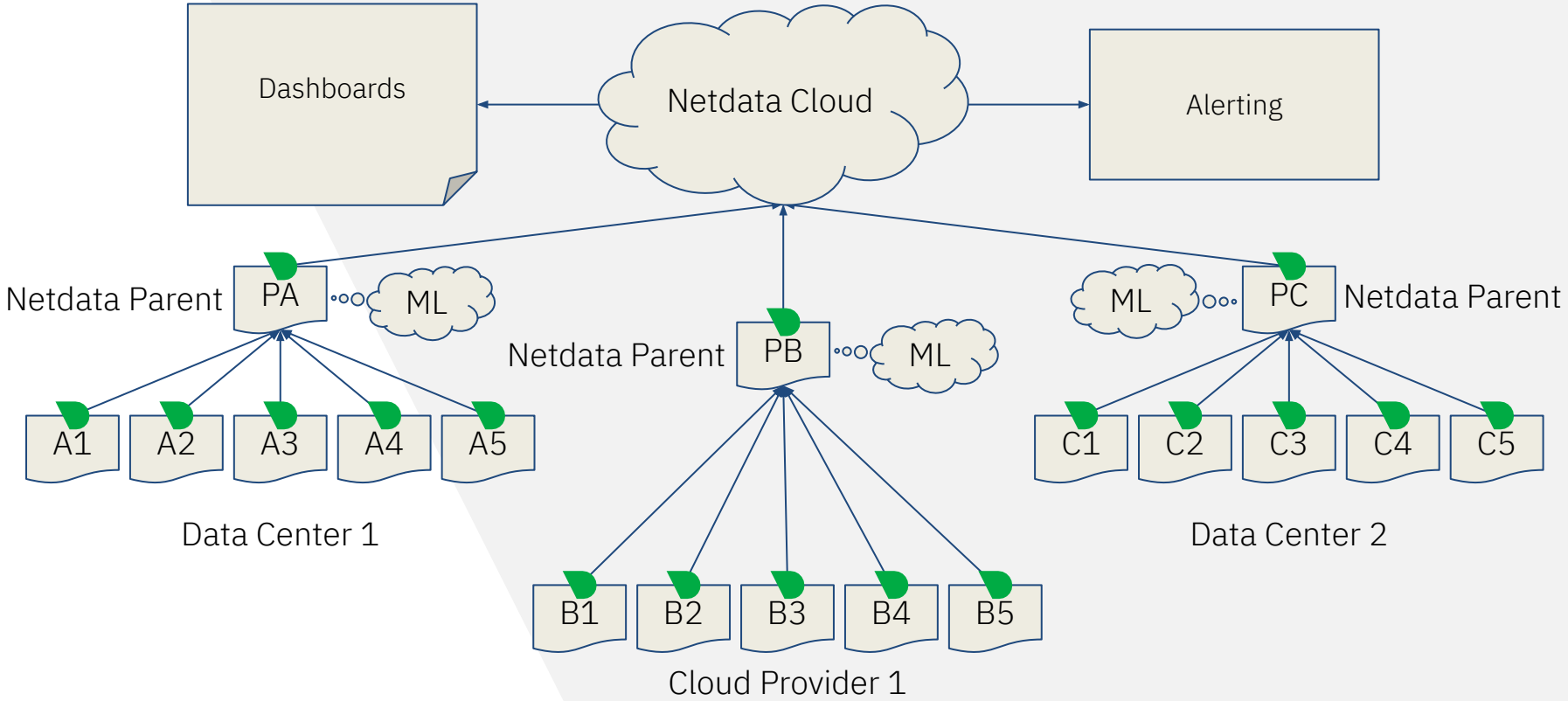
The Netdata way - distributed fully on-prem

Install the agent on all your systems, Use a Netdata Parent to access your infra.

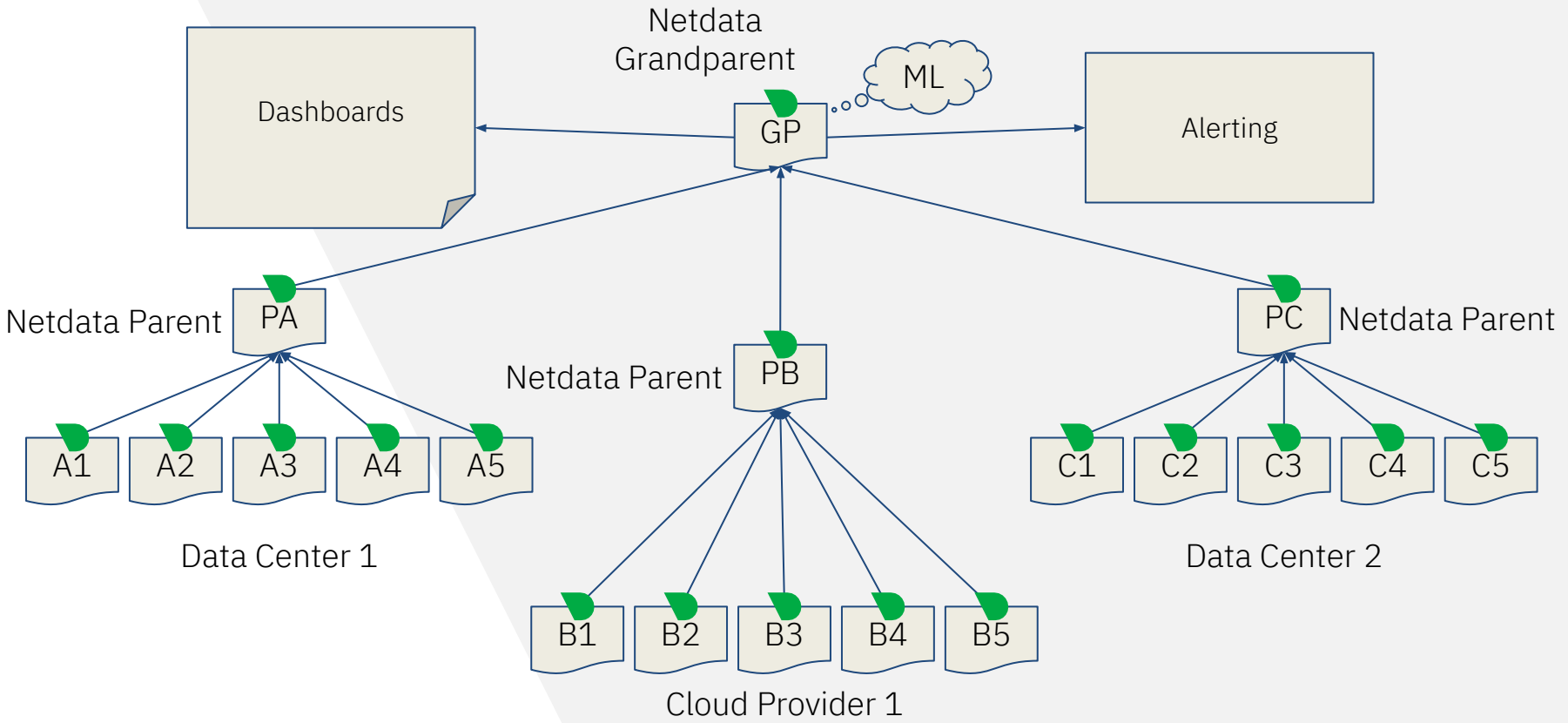


Netdata Parent is the same software with Netdata Agents.

The Netdata way - distributed mixed



The Netdata way - distributed fully on-prem



The Netdata way - the benefits

From Zero
To Hero
Today!

- **High fidelity monitoring**
  All metrics are collected per second.
  Thousands versus hundreds of metrics.
  Hundreds versus dozens of alerts.
- **All metrics are visualized**
  Fully automated infrastructure level dashboards, visualizing all metrics collected.
- **Powerful visualization**
  Slice and dice any dataset, using controls available on all charts. No need to learn a query language.
- **Out of the box alerts**
  Predefined alerts use rolling windows and statistical functions to detect common issues, without fixed thresholds.

Netdata vs Prometheus

Stress tested Netdata Parent and Prometheus with 500 servers, 40k containers, at 2.7 million metrics/s

:full comparison URL

- **-35% CPU Utilization**

  Netdata: 1.8 CPU cores per million of metrics/s
  Prometheus: 2.9 CPU cores per million of metrics/s

- **-49% Peak Memory Consumption**

  Netdata: 49 GIB
  Prometheus: 89 GiB

- **-12% Bandwidth**

  Netdata: 227 Mbps
  Prometheus: 257 Mbps

- **-98% Disk I/O**

  Netdata: 3 MiB/s (no reads, 3 MiB/s writes)
  Prometheus: 129 MiB/s (73.6 MiB/s reads, 55.2 MiB/s writes)

- **-75% Storage Footprint**

  Netdata: 10 days per-sec, 43 days per-min, 467 days per-hour
  Prometheus: 7 days per-sec

AI for Observability

AI for observability is tricky

Google:

All of Our ML Ideas Are Bad

(and We Should Feel Bad)

:URL

Wednesday, 2 October, 2019

Todd Underwood, Google

The vast majority of proposed production engineering uses of Machine Learning (ML) **will never work**. They are structurally unsuited to their intended purposes. There are many key problem domains where SREs want to apply ML but most of them do not have the right characteristics to be feasible in the way that we hope. After addressing the most common proposed uses of ML for production engineering and explaining why they won't work, several options will be considered, including approaches to evaluating proposed applications of ML for feasibility. **ML cannot solve most of the problems most people want it to**, but it can solve some problems. Probably.

How ML Works

Machine Learning is continuous learning.

1. You train a model with sample data and outcomes.
2. You give to it new data and you expect a correct answer.
3. If the answer is wrong, you provide feedback, so that it learns repeatedly.
4. Once the model is good, you apply the model to more use cases.

Sharing of ML models

Is it possible to train a model so good, that it can be applied to different servers or infra?

- Assume there are 2 servers: A, B
- Both run on identical hardware
- Both run the same operating system
- Both run the same application (e.g. a database server)
- Both have the same data

Can we train a ML model on A and use it on B?

Most likely not! *The trained model took into account the workload that was offered to A. If B has even slightly different workload, anomaly detection will be giving a lot of false positives.*

So what it can do?

Using ML we can have a simple and effective way to learn the behavior of our servers

ML is probably the simplest way to model the behavior of individual metrics.

So, given enough past values of a metric, ML can tell us if the value we just collected is an outlier or not.

We call this **Anomaly Detection**.

It is just a bit. True or False. And we store it together with the collected data for every metric.

Over a period of time, we calculate the **Anomaly Rate**, ie the % of samples that found to be anomalous.

Is Anomaly Detection accurate?

ML is noisy.

It has (a low rate) of random false positives.

To reduce the noise, we train **multiple** ML models with data from various time-frames of each metric.

We conclude that we have an outlier only when **all ML models** of a metric agree that the collected value is an outlier.

But there is still some noise.

Since ML is noisy, it cannot be the only trigger for us to wake up at 3AM...

Then, how can we trust it?

The alignment of anomalies across metrics makes ML useful.

Each metric has a somewhat noisy and inaccurate anomaly rate.

But when these noisy and inaccurate anomaly rates are **triggering in a short period of time across multiple metrics**, we can have a strong indication that something unseen in the recent past is happening and is widespread.

How it can help us?

Machine Learning and Anomaly Detection can help us answer these questions

- The dashboard has hundreds of charts. What is important now?
- I face a problem, but I am not sure where to look. Help me find the needle in the haystack!
- Is there a way to find the components, that although unrelated, they may somehow correlate to the problem I face?

ML In Netdata

What does Netdata do with ML?

Default ML Configuration In Netdata

- Trains a ML model per metric, **every 3 hours**, using the last 6 hours of data of each metric.

3 cores per million of metrics, <5% of a core for single nodes.

- Maintains **18 ML models** per metric, covering the last 57 hours (2days, 9 hours).
- Detects anomalies in real-time, while data are being collected.
- All available ML models for a metric need to agree that a collected sample is an outlier, for Netdata to consider it an **Anomaly**.
- Stores Anomaly Rate together with collected data.
- Calculates **Host-level Anomaly Score**.

Anomaly rate on every chart

Netdata's query engine calculates anomaly rates, in one-go, while querying metric data,

for all charts!

- Anomaly rate is always visible.
- Anomaly rate is calculated for all facets of a query (nodes, instances, dimensions, labels).
- Anomaly rate is calculated for all points of each chart.

A Netdata Chart

:Netdata Parent URL

Netdata Cloud Live Demo URL:

A Netdata Chart - controls

Aggregation across metrics

NIDL Controls - review data sources and slice/filter them (NIDL = Nodes, Instances, Dimensions, Labels)

Aggregation across time

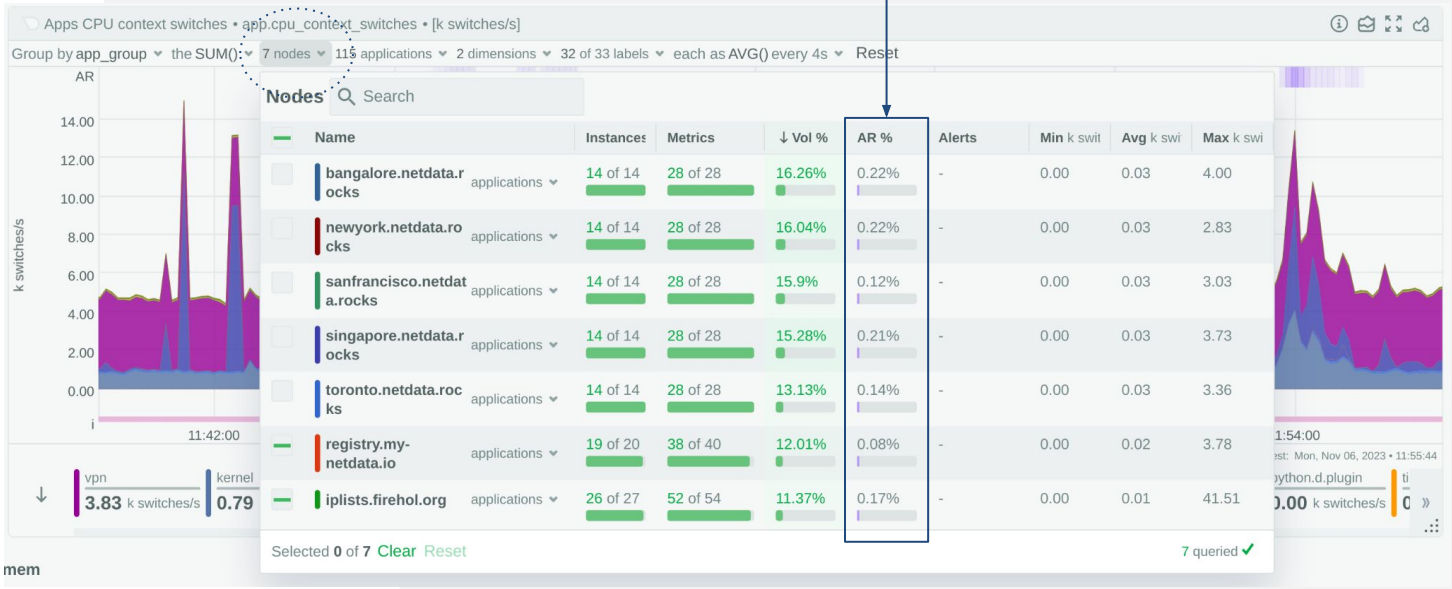
Dice the data

Anomaly rate ribbon

Info ribbon

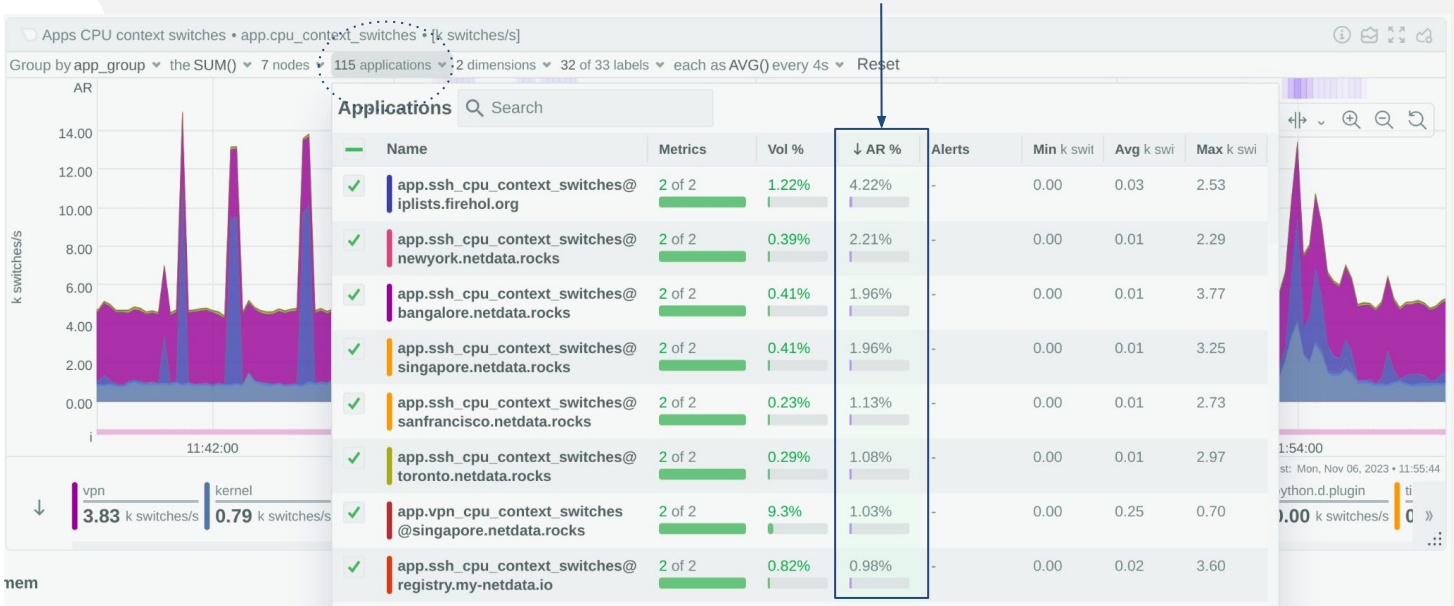
A Netdata Chart - anomaly rate per node

Anomaly
rate per node



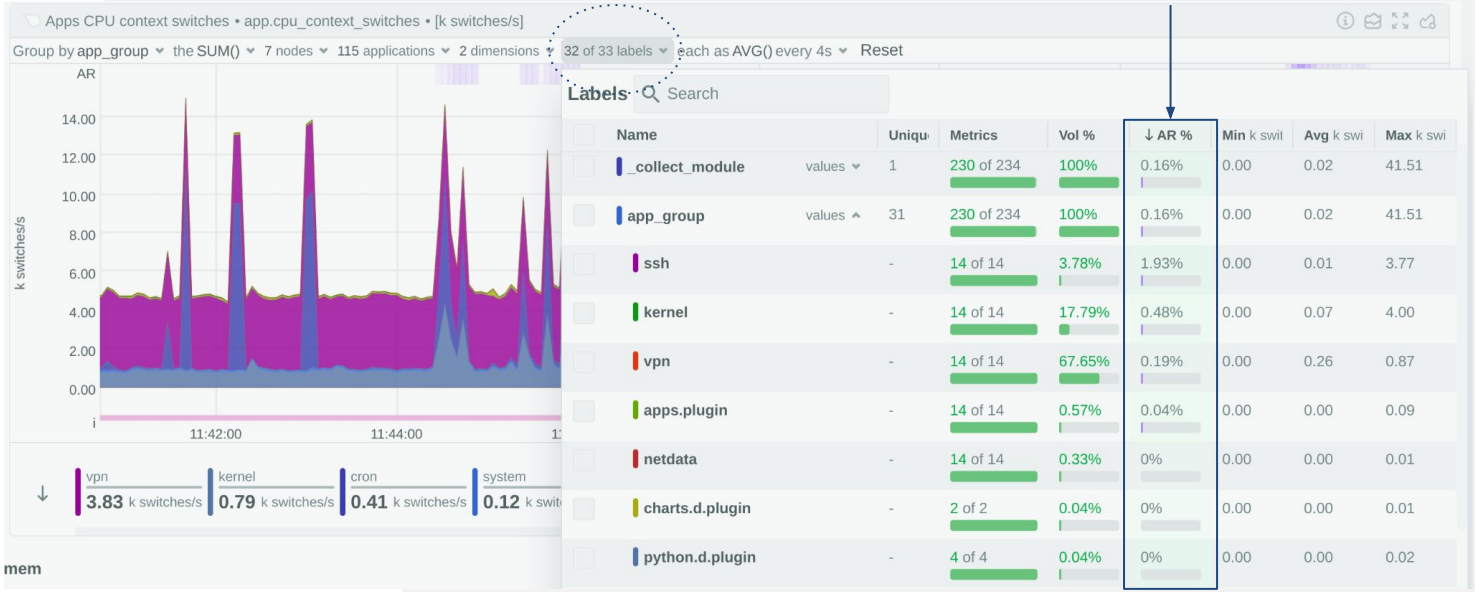
A Netdata Chart - anomaly rate per instance

Anomaly
rate per application instance



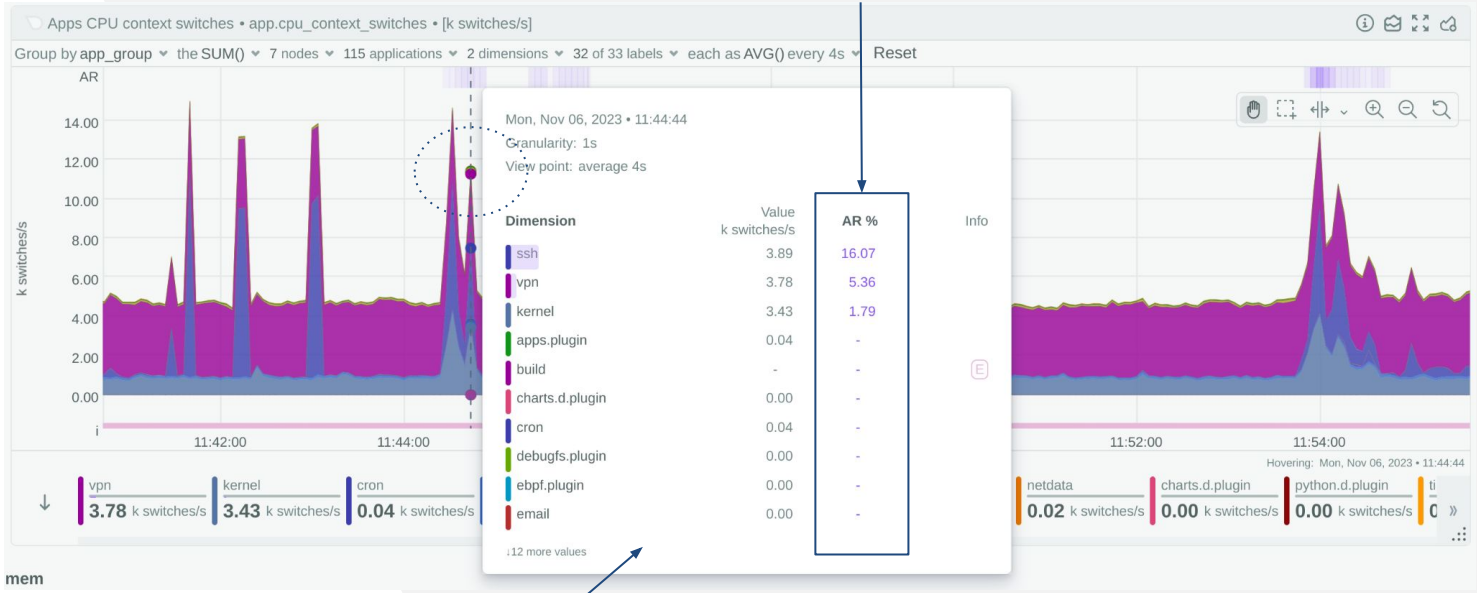
A Netdata Chart - anomaly rate per label

Anomaly
rate per label



A Netdata Chart - anomaly rate per point

Anomaly rate per point on the chart



Popover per point while hovering the chart

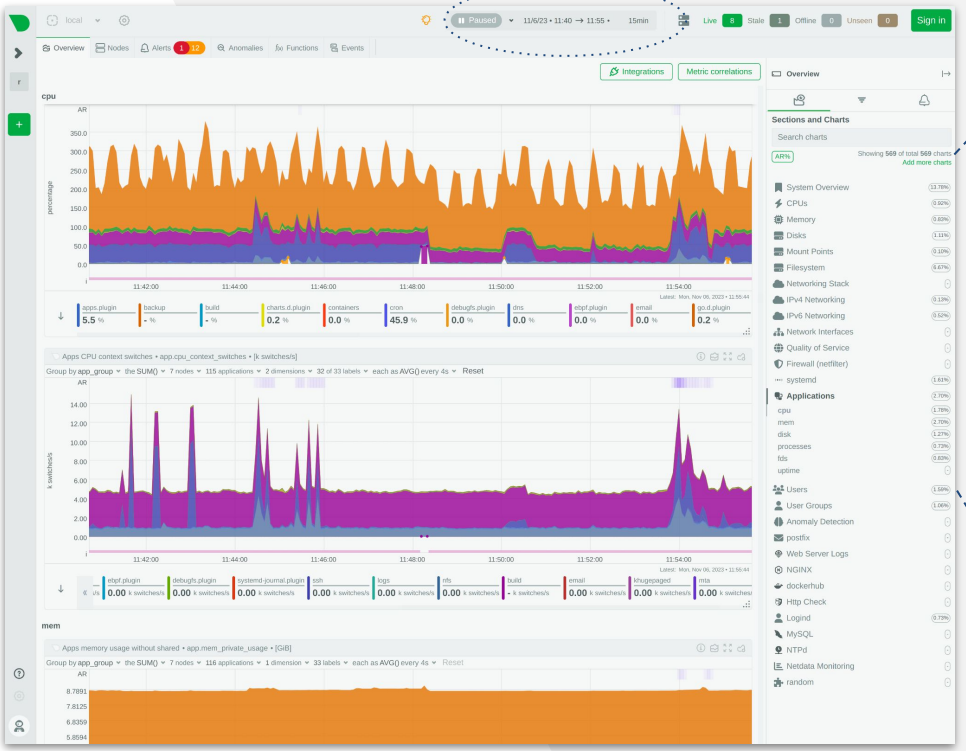
Netdata's scoring engine

Netdata can score all metrics based on their anomaly rate for any given time-frame!

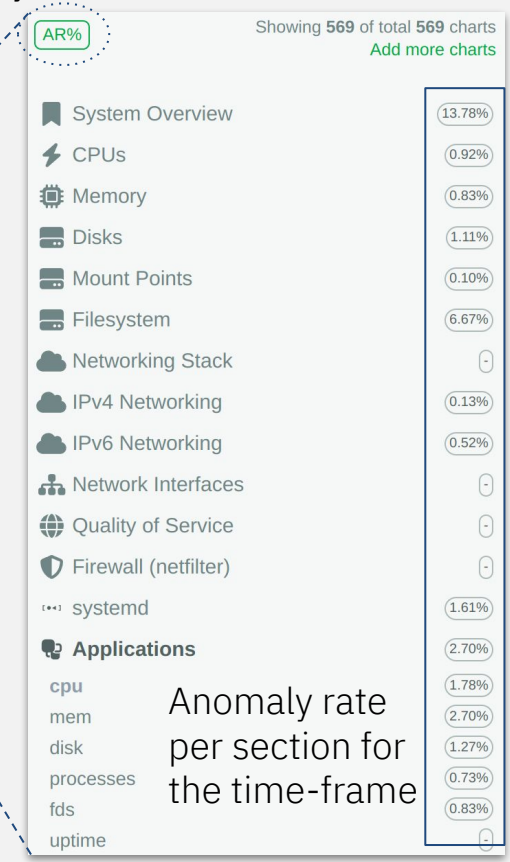
- A **scoring engine**, a unique feature across all monitoring systems.
- All metrics, independently of their context, can be scored across time, based on various parameters, including their anomaly rate.
- **Metrics correlations** is a subset of the scoring engine, that can score metrics based on their rate of change (data), anomaly rate of change (anomaly rate), but also based on volume, similarity, etc.

A Netdata Dashboard - what is anomalous?

Time-frame picker



Anomaly Rate button



Anomaly Advisor

Anomaly advisor assists in finding the needle in the haystack.

- Uses **Host Anomaly Rate** to identify durations of interest.
- **Host Anomaly Rate** is the percentage of the metrics of a host, that were found to be anomalous concurrently.
- So, 10% host anomaly rate, means that 10% of all the metrics the host exposes, were anomalous at the same time, depicting the spread of an anomaly.

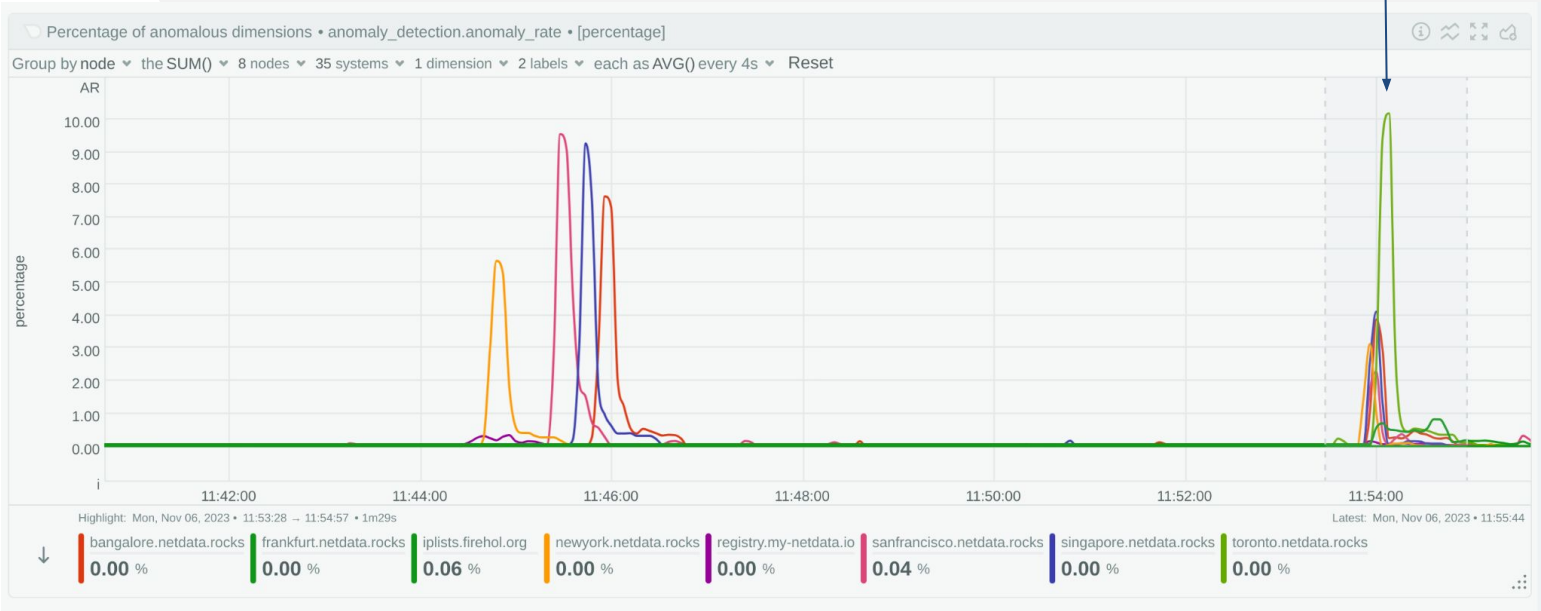
Anomaly Advisor - starting point

Percentage of Host Anomaly Rate

Number of metrics concurrently anomalous

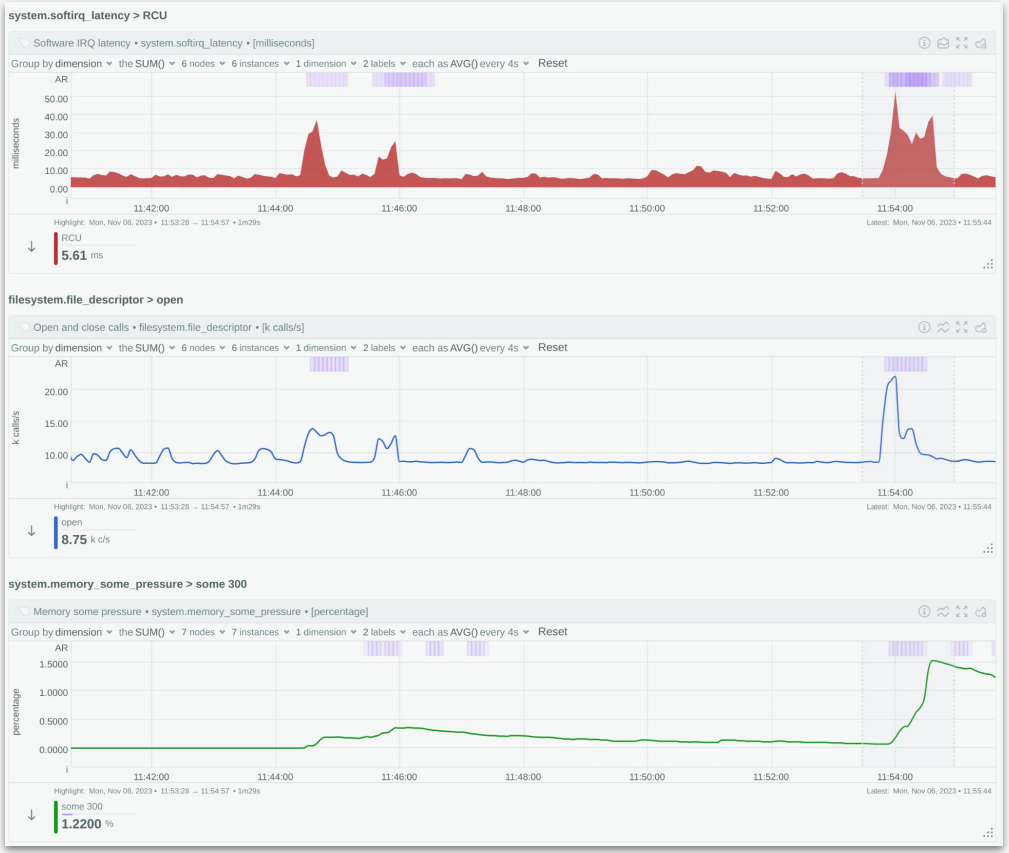
Anomaly Advisor - triggering the analysis

Highlighting an area on the chart, triggers the analysis



Anomaly Advisor - the analysis

Anomaly advisor presents a **sorted list of all metrics**, ordered by their **anomaly rate**, during the highlighted time-frame.



Highlights

Highlights of ML in Netdata

ML is an advisor.

It is everywhere, to help you understand the data better, and identify metrics of interest.

- **Unsupervised** anomaly detection
  (you don't need to configure it, or do anything about it - just use it).
- **For all metrics**, individually
  (learning their behavior, from their past data, including the workload).
- **Available everywhere**
  (on every dashboard, on every chart, on every metric, to help you understand the data better).
- **Scoring engine**
  (to help you find what is important among the thousands of metrics available).
- **Host-level Anomaly Score**
  (to quickly spot widespread anomalies).

What is next for ML in Netdata?

The future of ML in Netdata

Anomaly Rate can be configured to be more or less sensitive.

Different times within each day, and different days within a week or a month, may have different anomaly profiles.

- **ML Profiles**
  - Sensitivity Setting
    (more sensitive = more false positives)
    (less sensitive = less detections)
  - Intended Purpose
    (Security, Troubleshooting, etc)
- **Segment the Time**
  - Detect anomalies based what is usually happening at specific times, each day.

Questions?

:GitHub URL, https://github.com/netdata/netdata

Costa Tsaousis

Monitor your servers, containers, and applications, in high-resolution and in real-time!

netdata.cloud