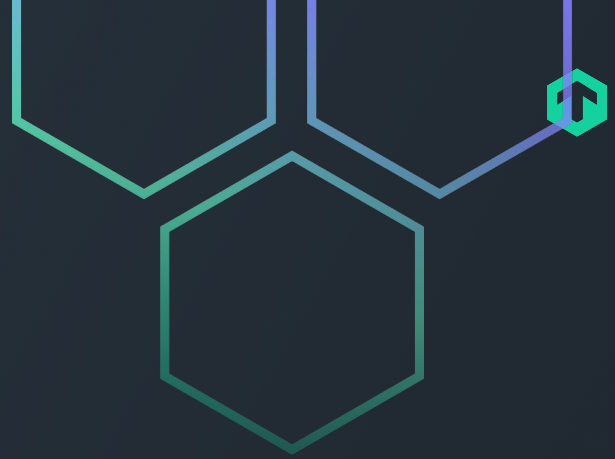
checkmk

Application monitoring in Checkmk What's new?

Community call, Oct 21st 2025

Niklas Pulina
Product Manager
Checkmk GmbH
checkmk.com

Agenda



01 Recap

02 Demo time: latest improvements

03 What’s next?

04 Special guest

05 Q&A

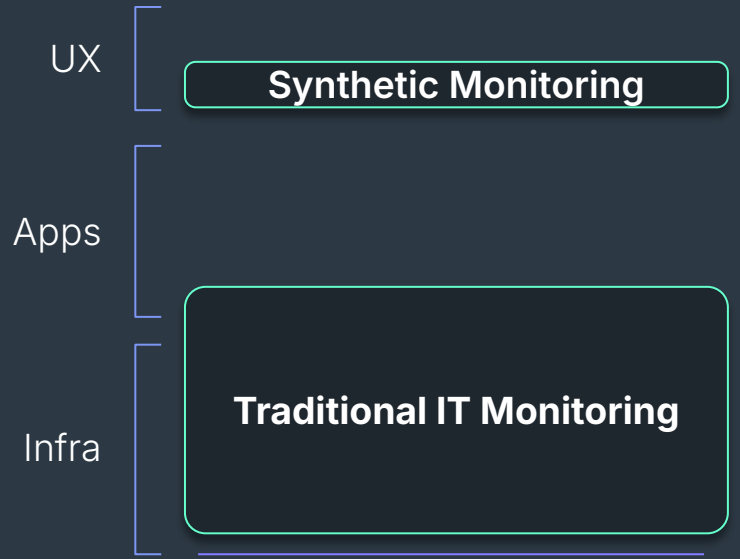
Recap

The challenge

What problem do we want to solve?

The IT monitoring market is evolving due to the shift towards cloud-native technologies and DevOps operating models. There's a growing move towards hybrid environments that combine traditional and cloud-native infrastructure.

Checkmk has historically excelled in monitoring traditional infrastructure. However, modern cloud-native applications are more dynamic and complex, requiring different monitoring approaches, often referred to as "observability."

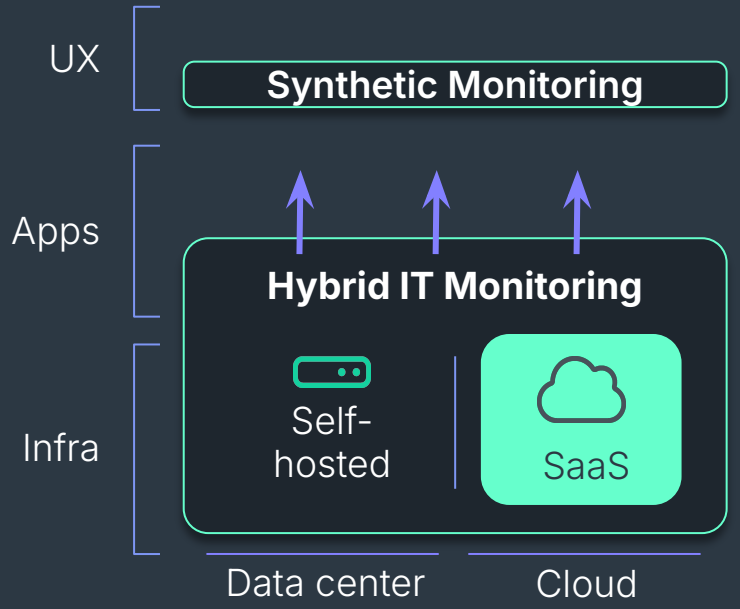


The solution

Application monitoring in Checkmk

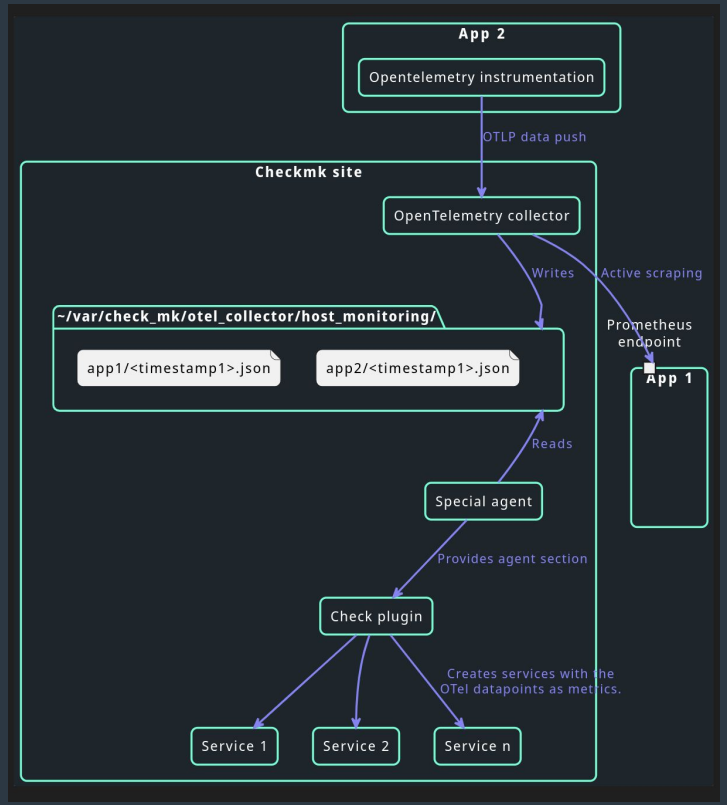
To provide a more comprehensive platform that covers both traditional and cloud-native environments, Checkmk expanded its focus on application monitoring in 2.4.

This strategic shift allows Checkmk to support customers throughout their transformation journeys and cater to the changing requirements of IT monitoring in a hybrid world.



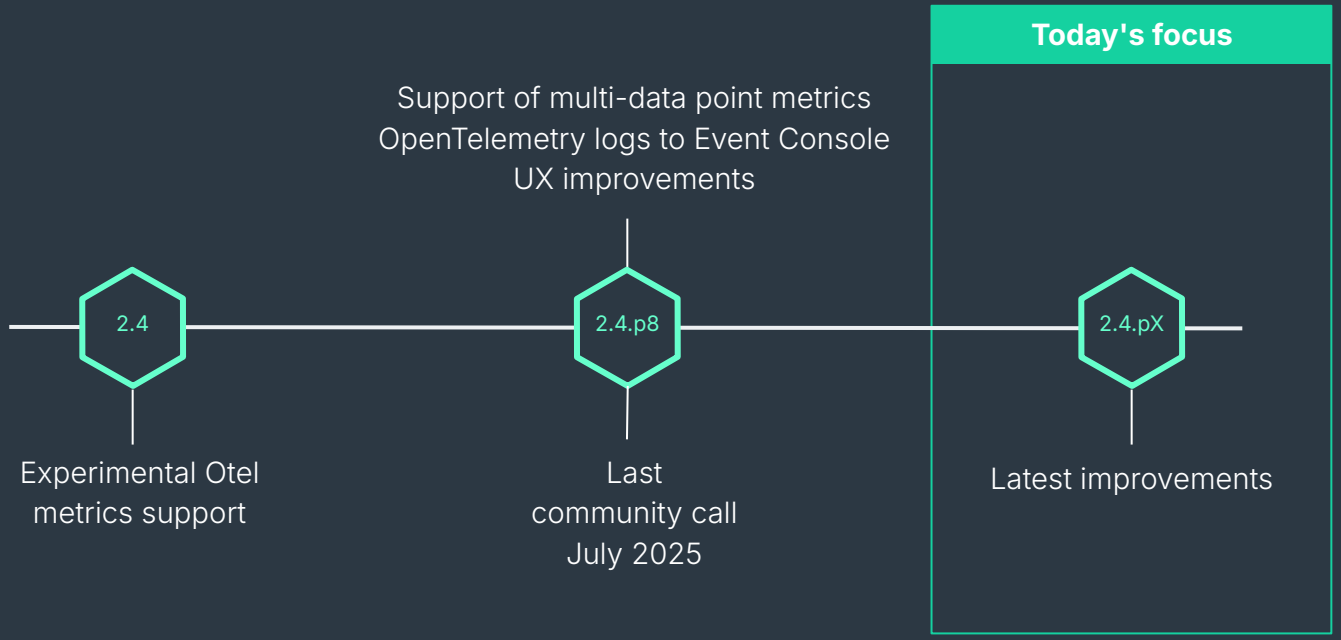
The architecture in Checkmk 2.4

(A simplified version of it)



The delivery

What has happened since the release of Checkmk 2.4.0?

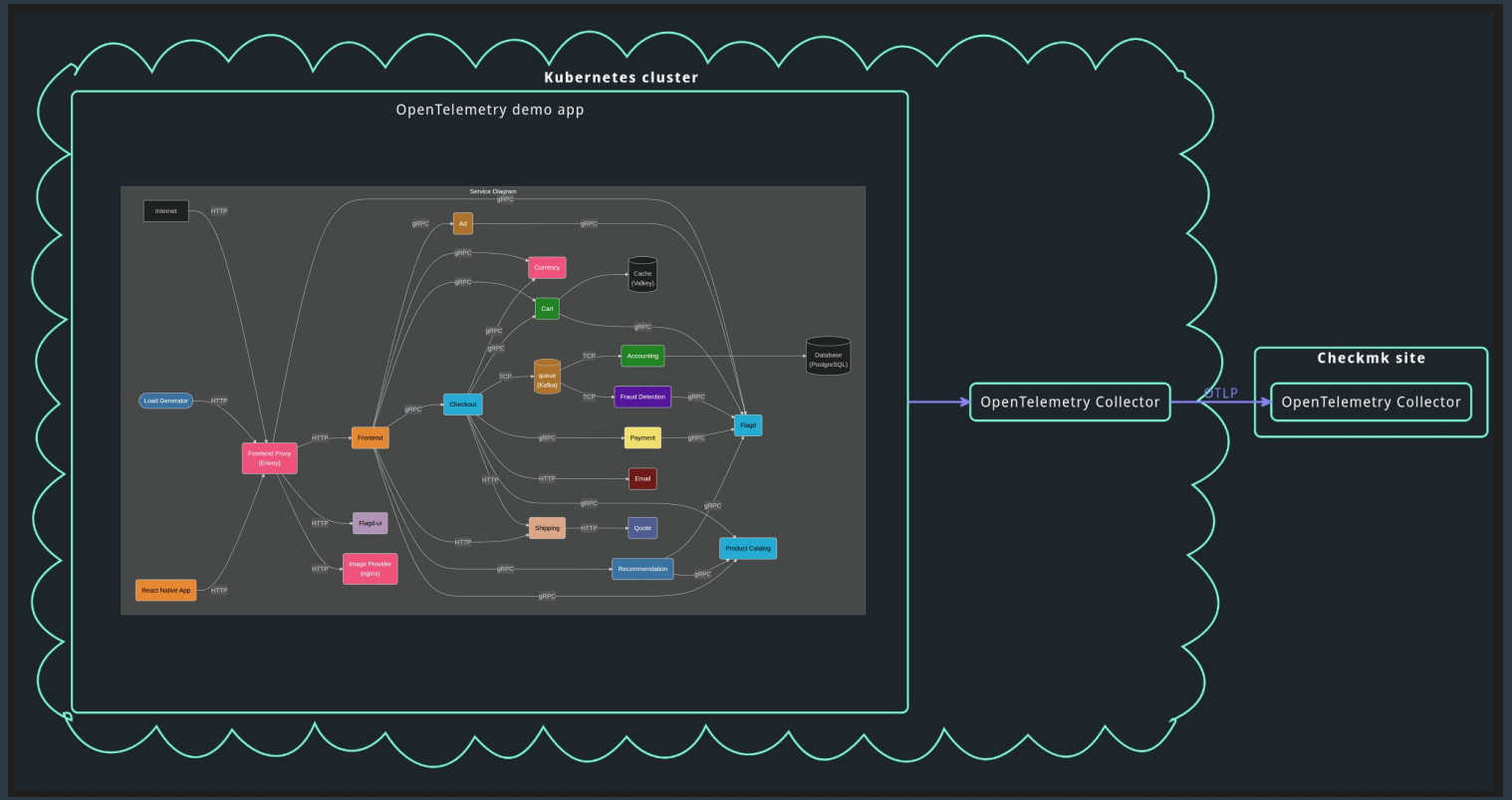


Demo time: latest improvements

Interrupt and ask whenever anything is unclear!

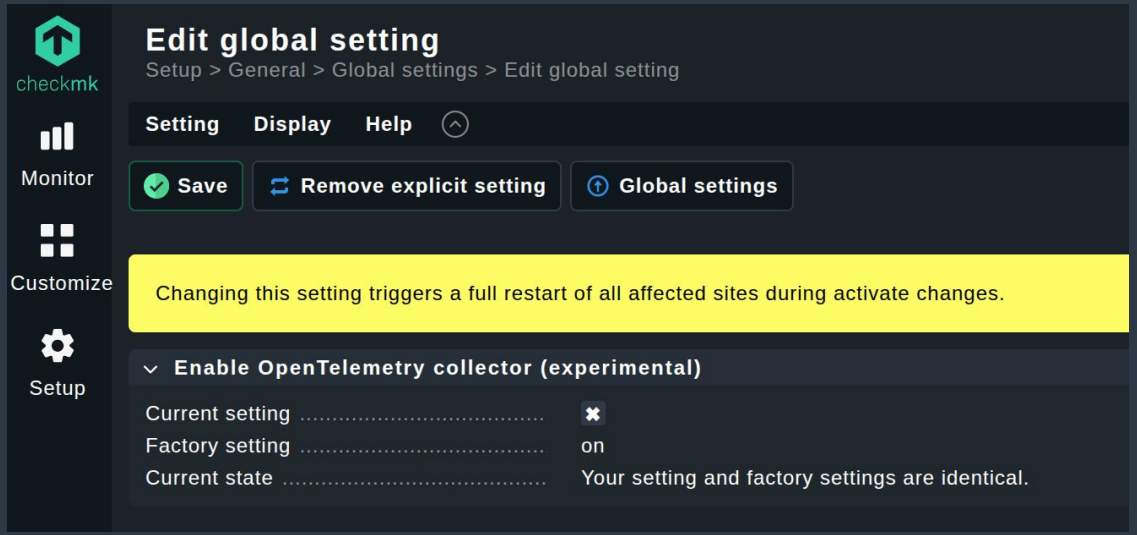
Today’s demo setup

The OpenTelemetry demo app, deployed in a Kubernetes cluster



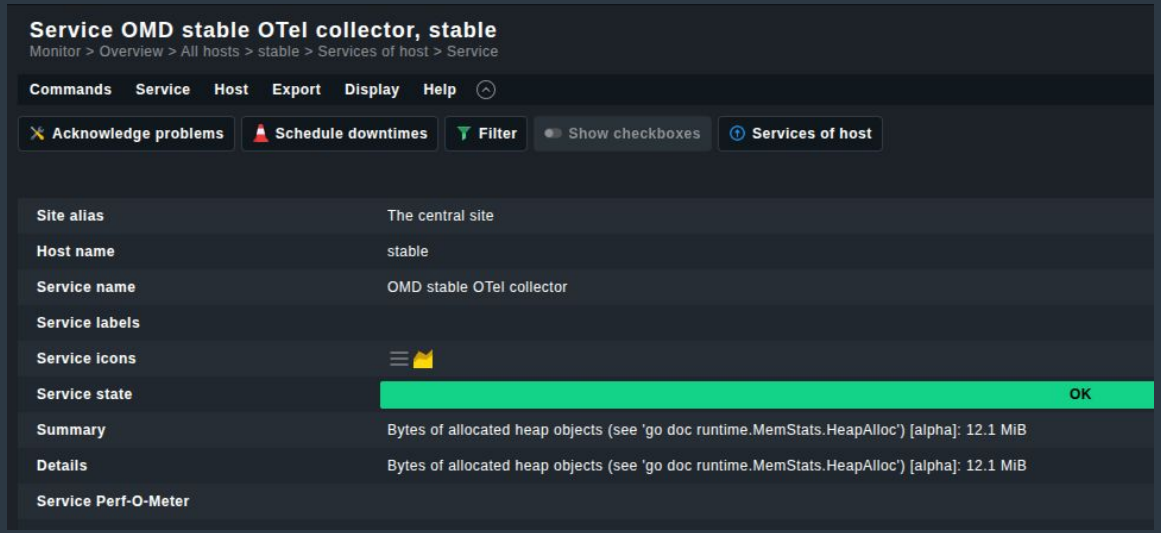
Start the OpenTelemetry Collector from the UI

You can now start the OpenTelemetry Collector from Checkmk's UI, and do no longer need to use OMD.



Monitor the Collector’s memory consumption

There is a new service named **OMD <site> OTel collector** that monitors the memory usage of the collector using the self-monitoring data.

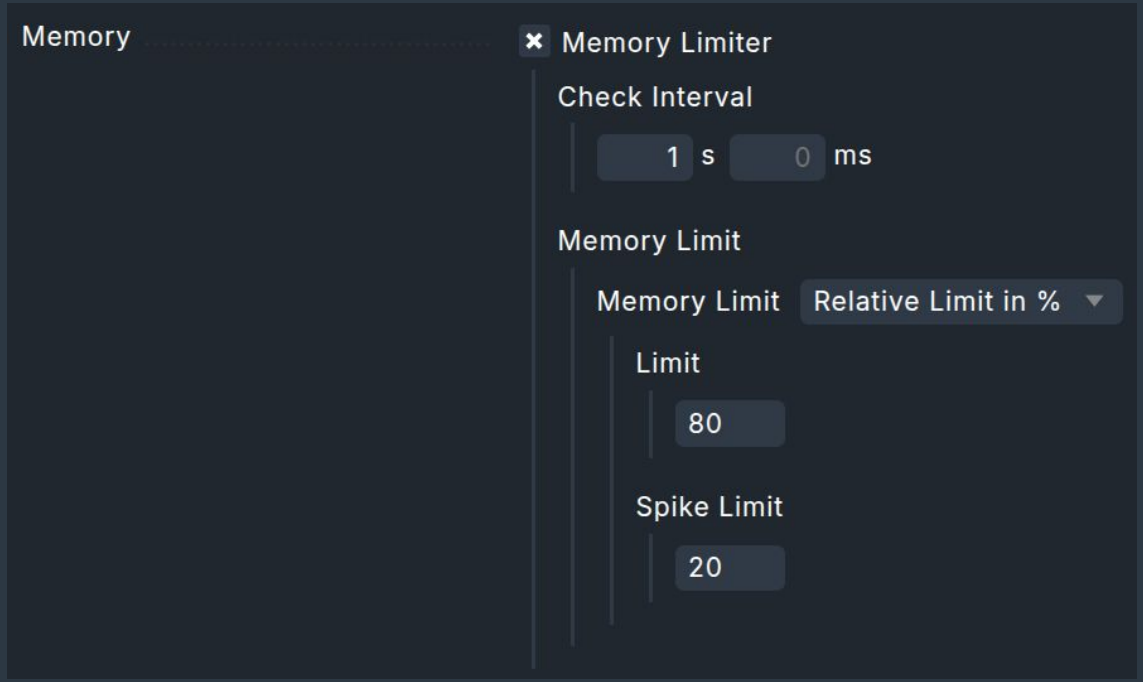


Limit collector memory usage

Checkmk 2.5

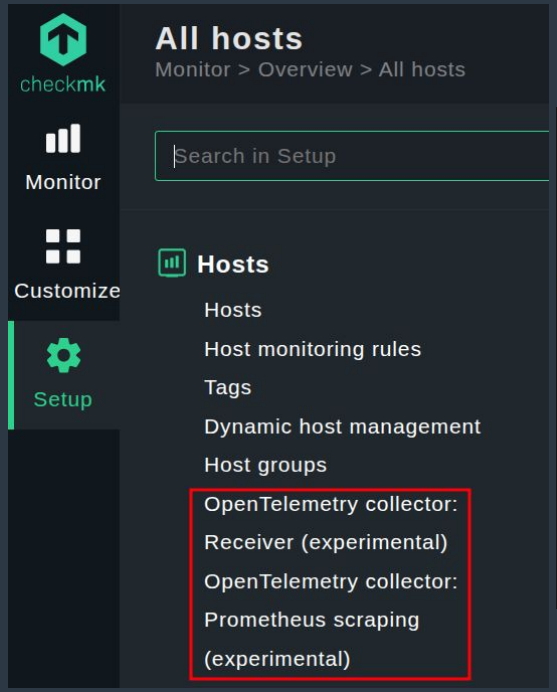
To ensure the OTel collector does not consume too much memory, you can now set custom memory limits.

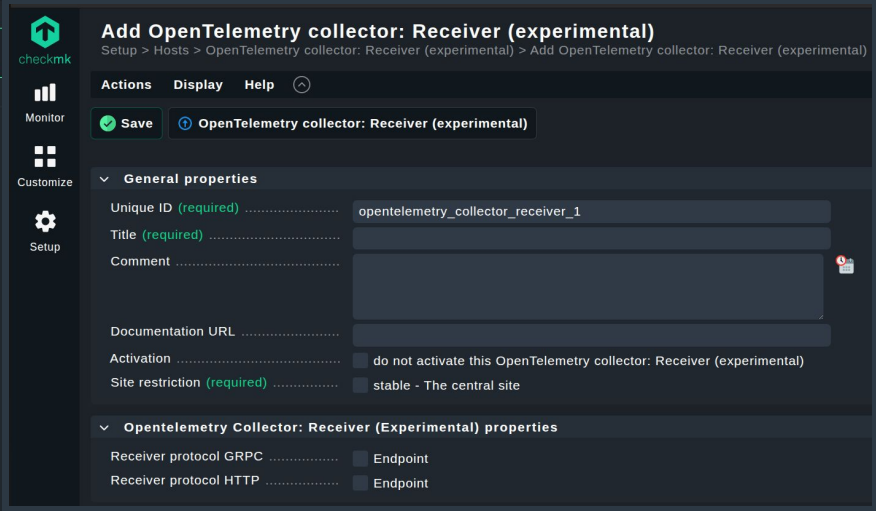
By default, the “hard” memory limit is equal to 80% of the available memory on startup, the “soft” memory limit is set to 60%.



Split collector configuration

You now configure the OTel receivers and the prometheus scraping using individual rule sets.





Better handling of histograms

OpenTelemetry histograms consist of three main elements:

- **Count**: The total number of measurements recorded
- **Sum**: The sum of all recorded values
- **Buckets** (optional): Predefined ranges that track how many values fall within each range

Previously, only the count value of histograms was emitted as metric.

Now, histograms get normalized to their delta between two check cycles.
Eventually, the metric is calculated as follows ...

Δ sum / Δ count

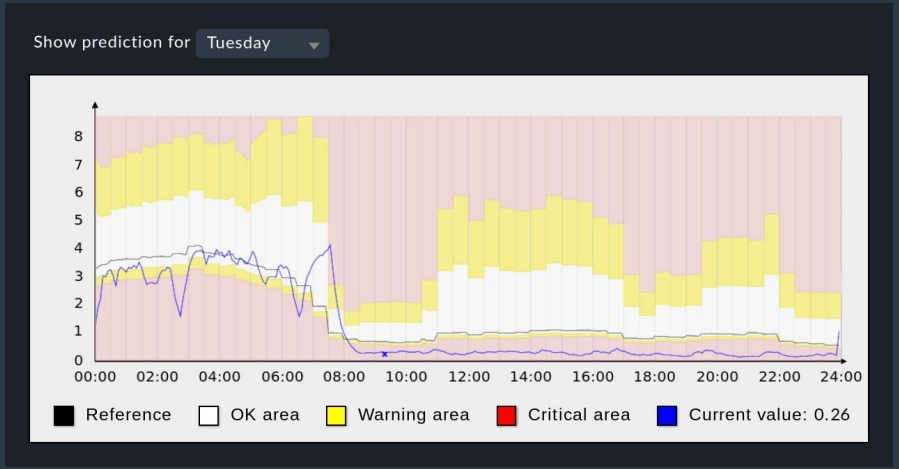
... to give a more meaningful representation of the values behind.

Support for predictive levels

Checkmk 2.5

You can now configure predictive levels when configuring specific OTel metrics and use it for alerting.

Previously, predictive levels were not available for OpenTelemetry metrics.

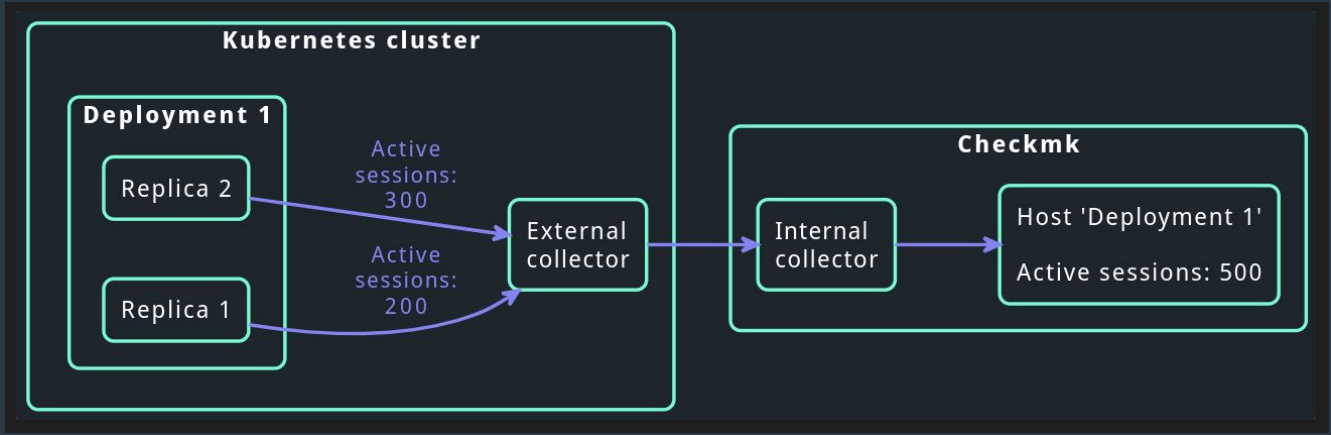


Multi-emitter metrics aggregation

You can configure how the OTel metrics emitted by multiple entities get aggregated when mapped to a single host in Checkmk.

Selectable methods are:

- Latest
- Average
- Max
- Min
- Sum

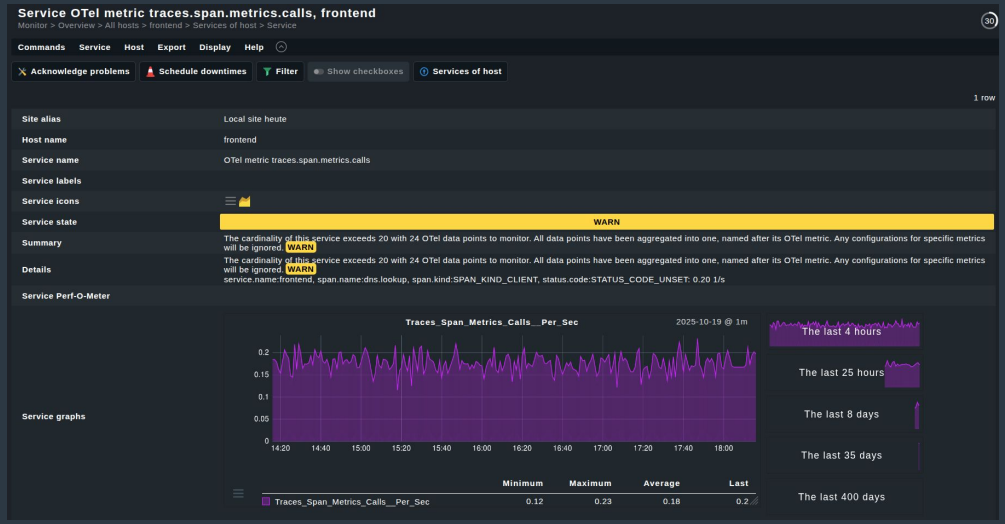


High-cardinality metrics aggregation

If an OpenTelemetry metric which is mapped to one Checkmk service has more than 20 unique combinations of datapoint attributes, the resulting time series get aggregated into one Checkmk metric.

Selectable aggregation methods are:

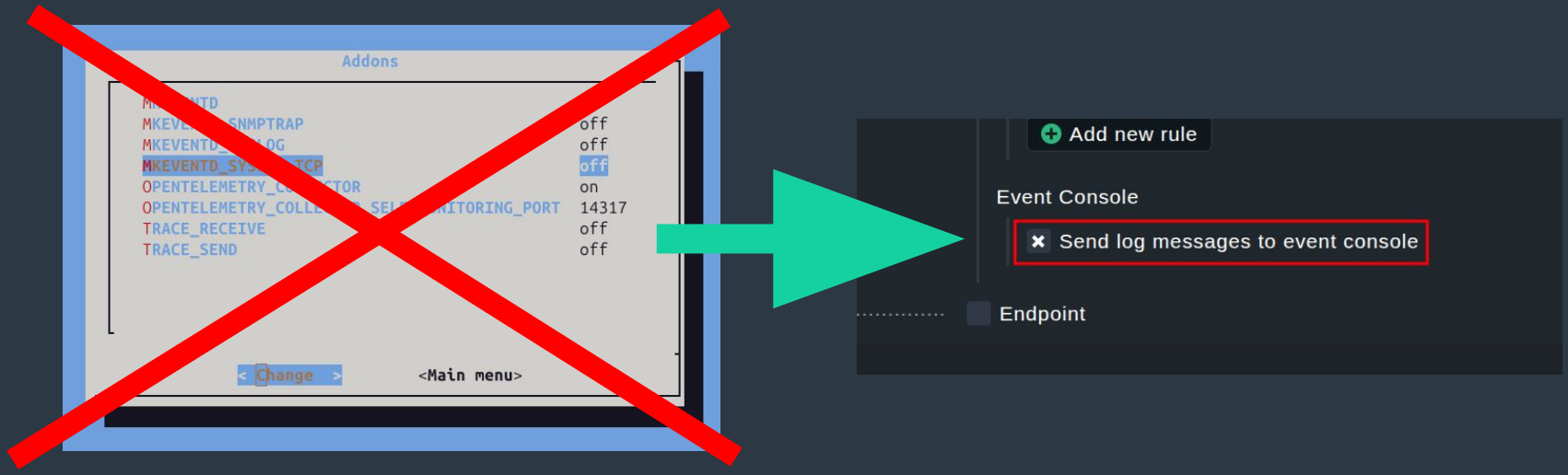
- Latest
- Average
- Max
- Min
- Sum



Syslog logs via Unix socket

Previously, you had to make the Event Console listen on port 514 to make it receive syslog logs converted from OpenTelemetry logs.

Now you can conveniently pass them to the Event Console via Unix socket.

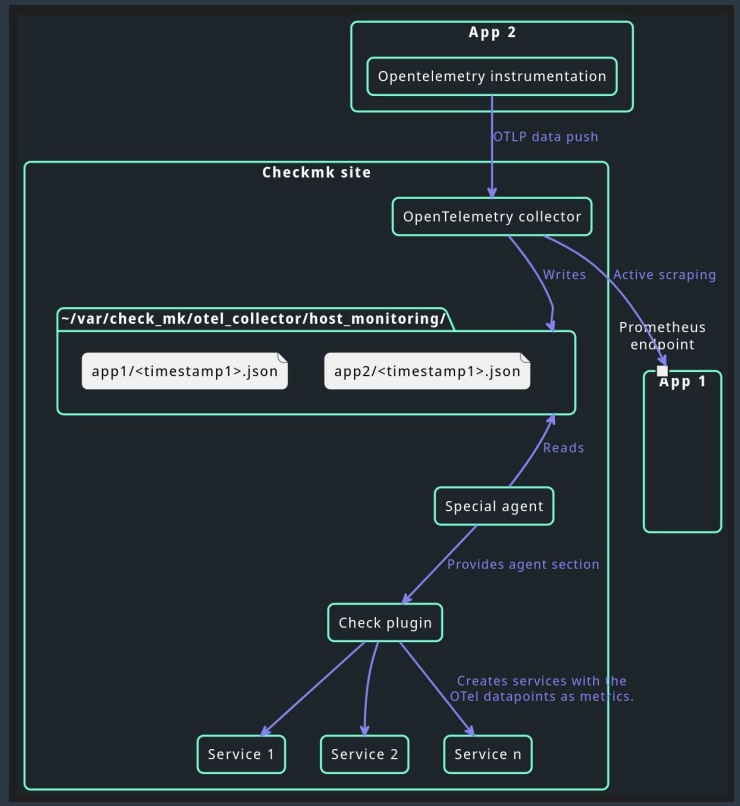


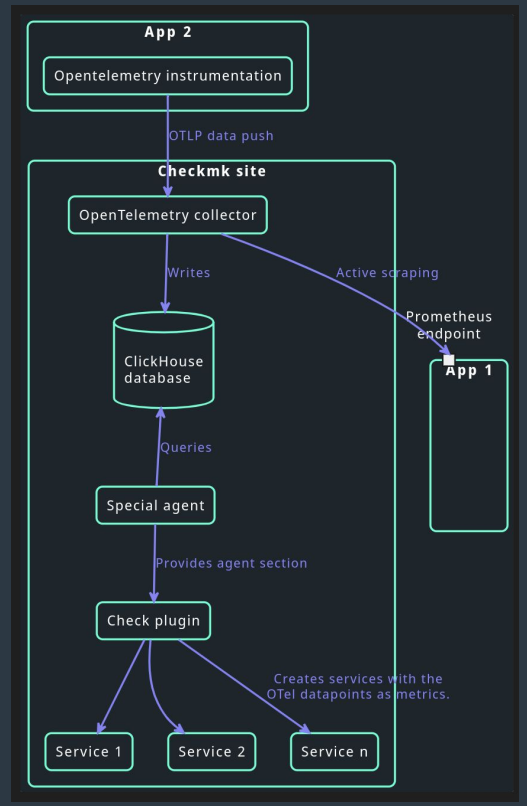
What’s next?

A new metric backend

Enabling a wide range of new use cases

Checkmk 2.5





Next steps

Let's think about Checkmk 2.6