Checkmk Conference #10

What’s next for Checkmk

Lars Michelsen
Chief Technology Officer
Checkmk GmbH

Martin Hirschvogel
Chief Product Officer
Checkmk GmbH

Building Checkmk together

Checkmk
Conference

Implemented a lot of conference feedback …

Implemented in Checkmk 2.3

Conference #8 votings

- Synthetic monitoring
- REST-API: Dynamic host config endpoints
- Google Authenticator support
- Dev API for GUI extensions
- Ansible collection
- Better HTTP check
- Cisco Meraki monitoring
- … and more coming in Checkmk 2.4

Conference #9 votings

- REST-API: Endpoint for global settings
- REST-API: Improve editing rules
- Improve UI Loading times
- Extreme Networks monitoring plug-ins
- Pure Storage monitoring plug-ins
- K3s support
- … and much more coming in Checkmk 2.4

Future focus input → long-term product strategy (e.g. application monitoring)

… and from the ideas portal

Implemented in Checkmk 2.3

Monitor anything (9)

- Support for NetApp REST-API
- Plugin to monitor MSSQL database server running on Linux
- Pure Storage Integration
- K3s and RKE2 monitoring
- OAuth for mail loop check
- Certificate check: query & check issuing ca
- Enhance monitoring for Kyocera
- Add redfish support in appliance
- Implement check_curl for http2 support

Powerful configuration (3)

- Support AND, OR and NOT for labels in rules
- Automatic label depending for OS
- Periodic service discovery - automatically update host labels

Alerts & analytics (2)

- Predictive Monitoring in Distributed
- Top 10 hosts dashlet

Extensible interfaces (2)

- Dynamic host config endpoint
- Edit Rule Endpoint

Great user experience (6)

- Setup folders: sort case-insensitive
- Show user activating changes
- Filter on folders should work
- Extend the text field length
- Change ‘traffic light’ symbol for matching rules that are deactivated
- Fix "open/close this element" layout

Monitoring platform (3)

- TOTP codes for 2FA
- SNMP AES-256 Privacy protocol
- Checkmk Managed Services Edition with Cloud Edition features

67 ideas implemented since May 2022

Our guiding principles

We do not build any feature without user involvement.

We consider user priorities in our high-level strategy.

We are transparent about our roadmap and progress.

We react quickly to plug-in development requests.

Note: To enable this, we are reworking a lot of fundamental processes internally

Checkmk 2.4 and beyond

Planned for Checkmk 2.4

Likely coming in 2.4. Changes to scope possible.

Accepted for future release

Candidate for a future release, without commitment on timeline. Some aspects could make it to 2.4 as well.

Disclaimer: All is subject to change. Unplanned things can always happen.

Over the last 5 years worked to fulfill this mission...

Cloud Data Center IoT

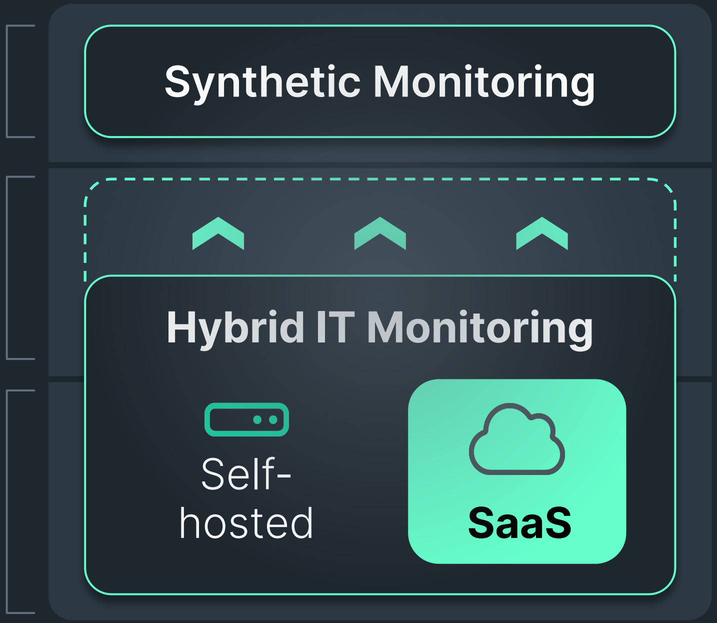
Great user experience
for beginners and experts

Modern monitoring
for the hybrid world

Extensible interfaces
for automation & development

One integrated monitoring
Secure & scalable

… and have already started our next journey

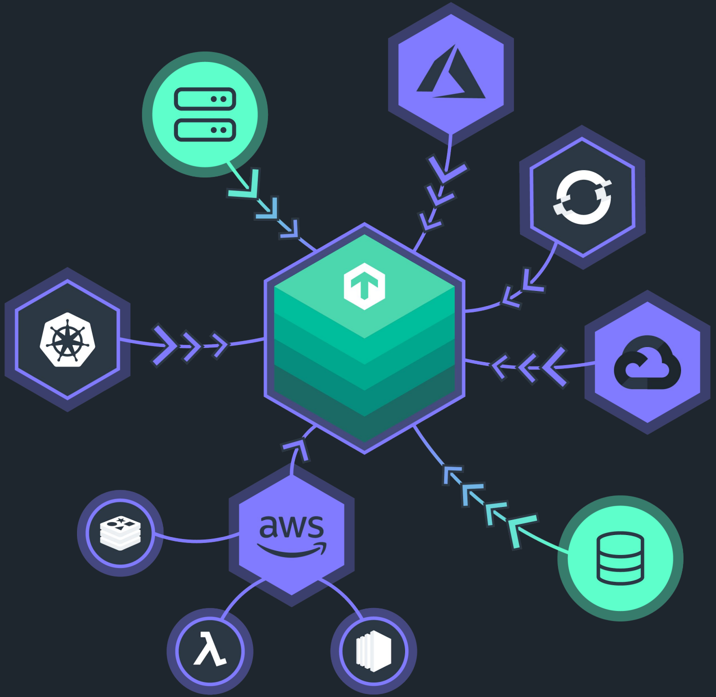


Checkmk
Conference

What’s next for Checkmk Cloud

Checkmk
Conference

Monitor anything with Checkmk Cloud ...



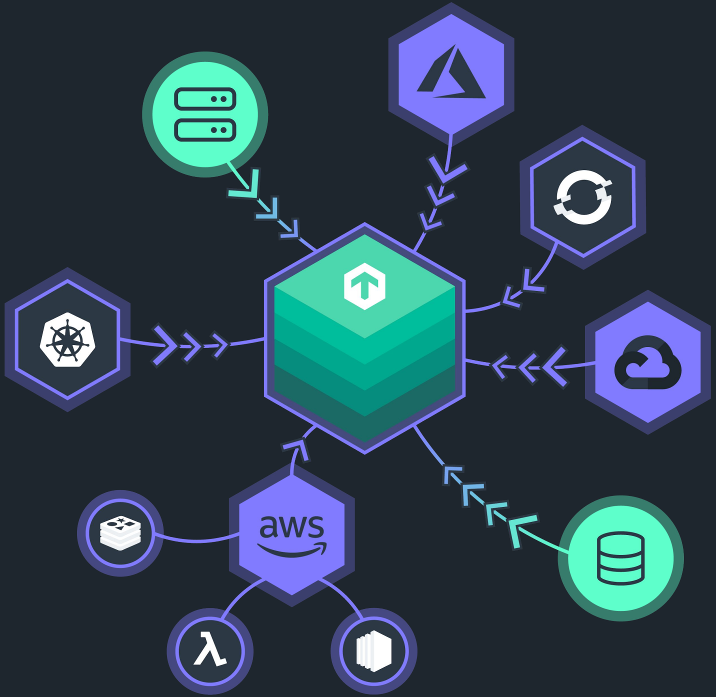
Full hybrid infrastructure coverage

- ‘Automated remote probe’ for
  - Network device monitoring via SNMP
  - Internal services monitoring (e.g. vSphere)
- Simplified push-based Kubernetes monitoring
- Log analytics via Event Console

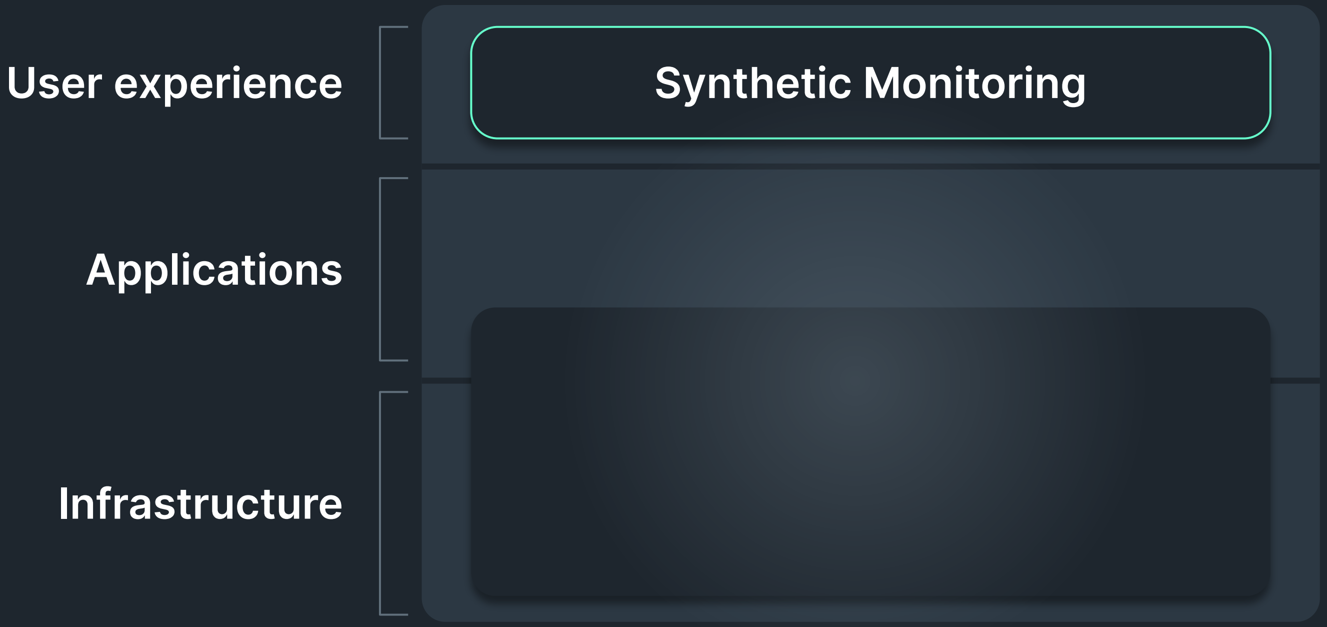
… on a powerful platform

Extended platform

- Single-sign on
- Support for monitoring large infrastructures
- Certified extension store



Application monitoring with Checkmk



User experience

Applications

Infrastructure

Synthetic Monitoring

Faster, simpler test authoring

Accepted for future release

Develop the test → **Run** the test → **Monitor** the test

Create & edit tests directly in Checkmk

For simple and quick tests

Seamless Visual Studio Code integration

Checkmk Synthetic Monitoring extension simplifying the development of tests leveraging Checkmk features

Completely automated workflow

Planned for Checkmk 2.4

Develop the test → **Run** the test → **Monitor** the test

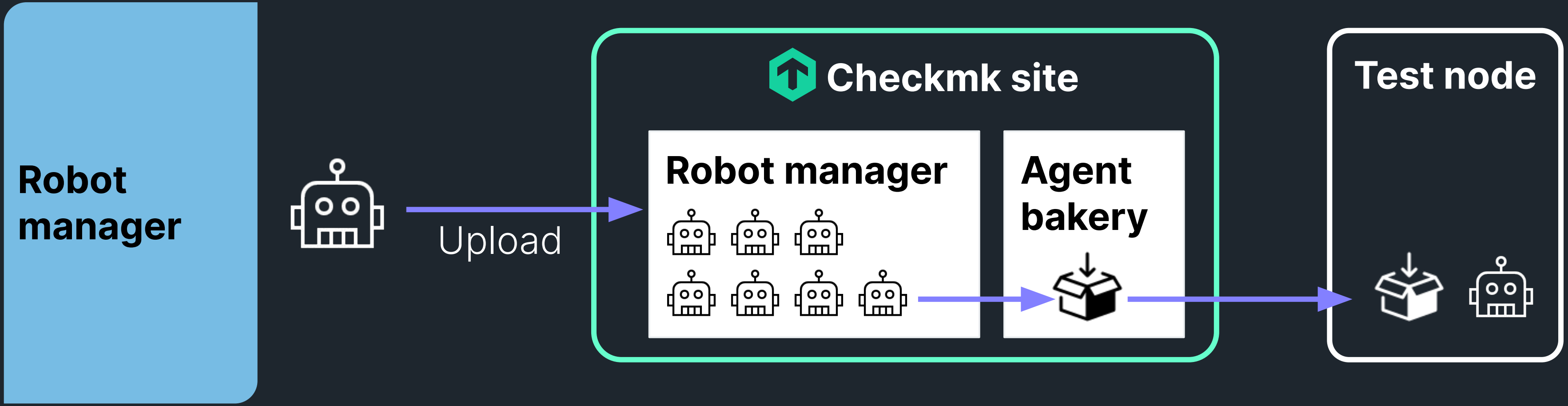
Windows headless desktop tests and Linux support.

Completely automated workflow

Planned for Checkmk 2.4

Develop the test | **Run** the test | **Monitor** the test

Windows headless desktop tests and Linux support.

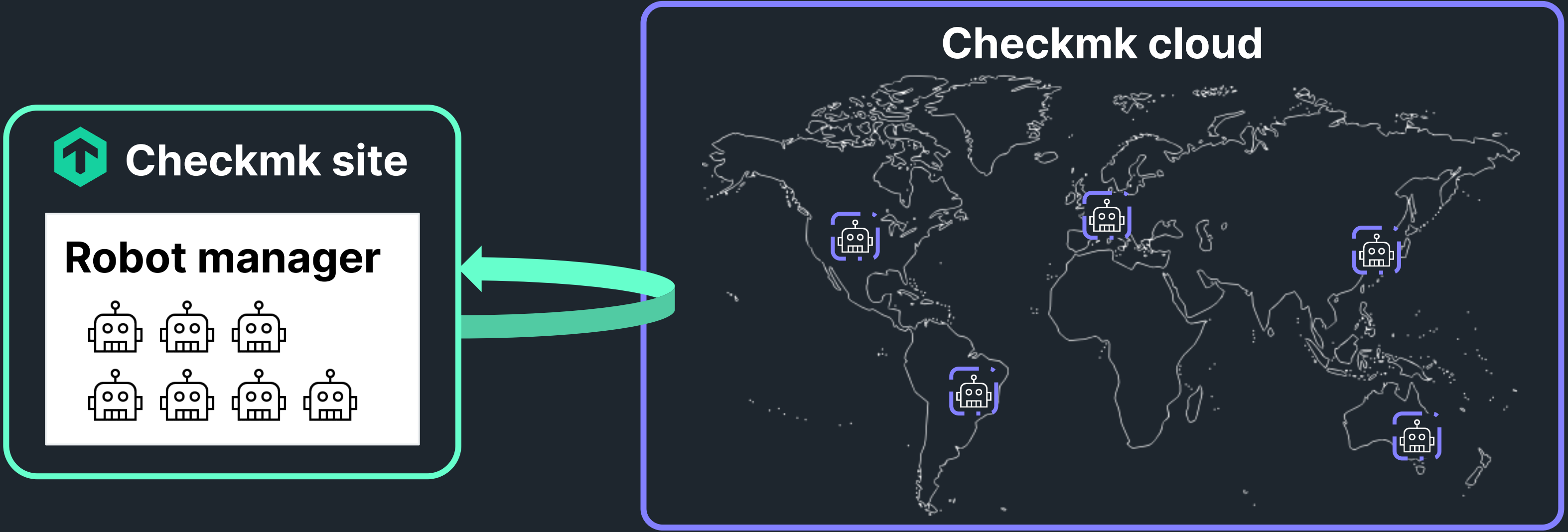


Checkmk
Conference

Run your robots in the Checkmk cloud

Accepted for future release

Develop the test → **Run** the test → **Monitor** the test

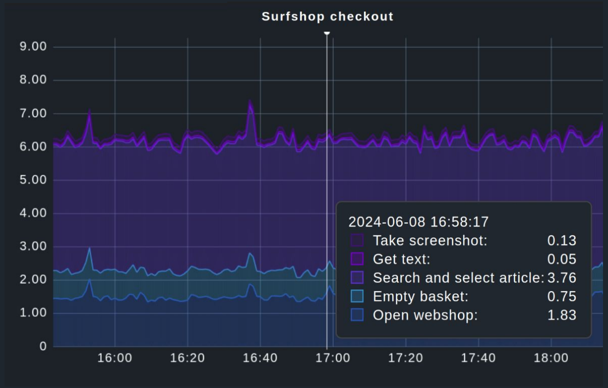


Test from the outside without need for own test infrastructure

More insights and in-depth monitoring

Accepted for future release

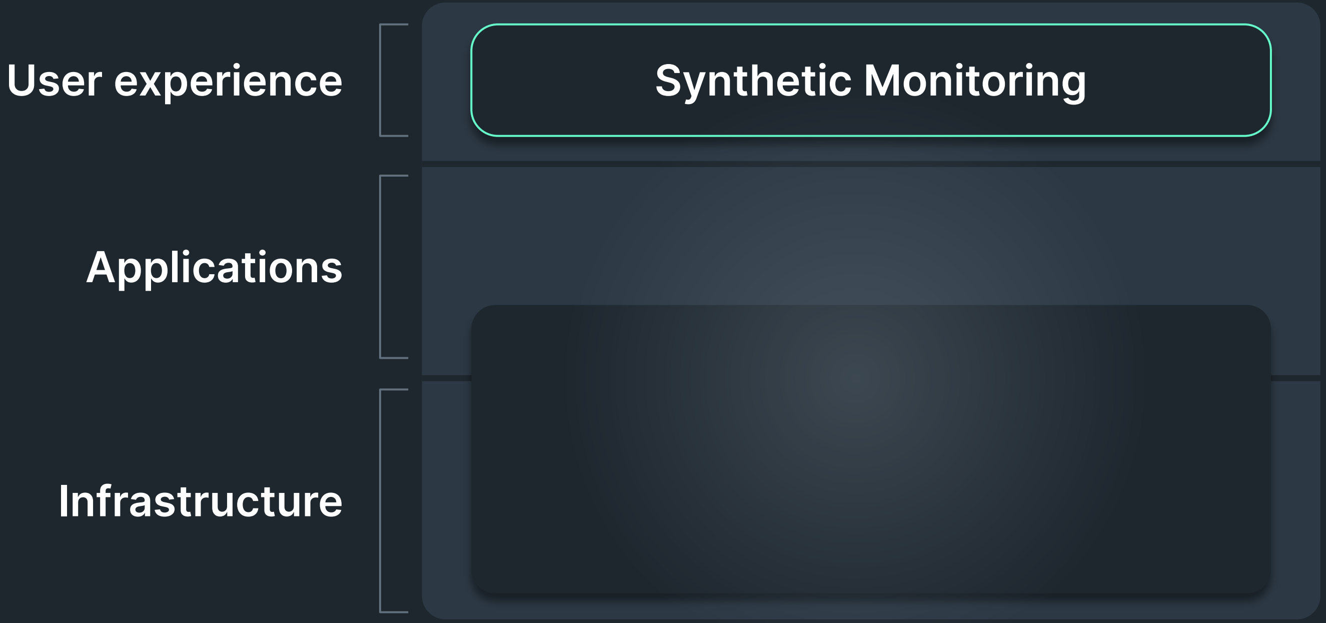
Develop the test → **Run** the test → **Monitor** the test



```
2024-06-08 16:58:17
Take screenshot:              0.13
Get text:                     0.05
Search and select article:    3.76
Empty basket:                 0.75
Open webshop:                 1.83
```

Surfshop: Check-out

19:20	🟩	🟩	🟩	🟥	⬜	⬜
19:10	🟩	🟩	🟩	🟩	🟩	🟩
19:00	🟩	🟩	🟩	🟩	🟩	🟩

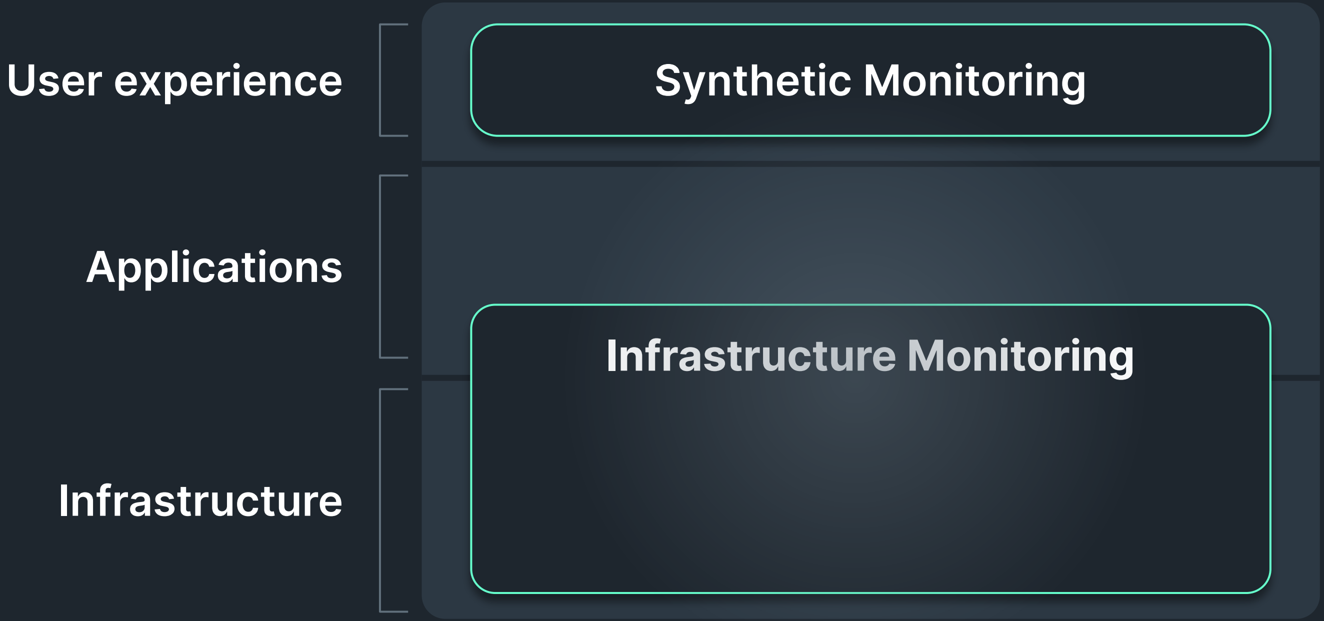


User experience

Synthetic Monitoring

Applications

Infrastructure



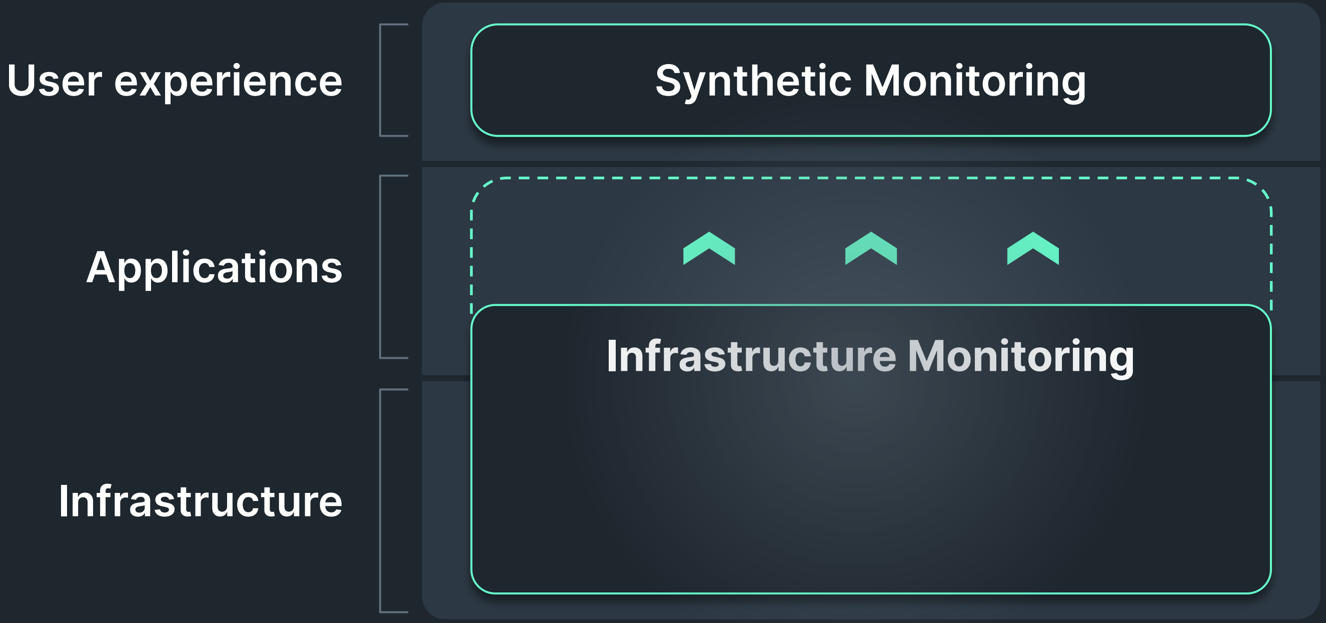
User experience

Applications

Infrastructure

Synthetic Monitoring

Infrastructure Monitoring



User experience

Applications

Infrastructure

Synthetic Monitoring

Infrastructure Monitoring

The two sides of application monitoring

Supported already by Checkmk

E

SAP

Microsoft Active Directory

HAPROXY

splunk>

APACHE ACTIVEMQ

Couchbase

elasticsearch

Jenkins

graylog

RabbitMQ

many more ...

+ active checks

?

The two sides of application monitoring

Supported already by Checkmk

Business applications

Not ‘your’ software

?

The two sides of application monitoring

Supported already by Checkmk

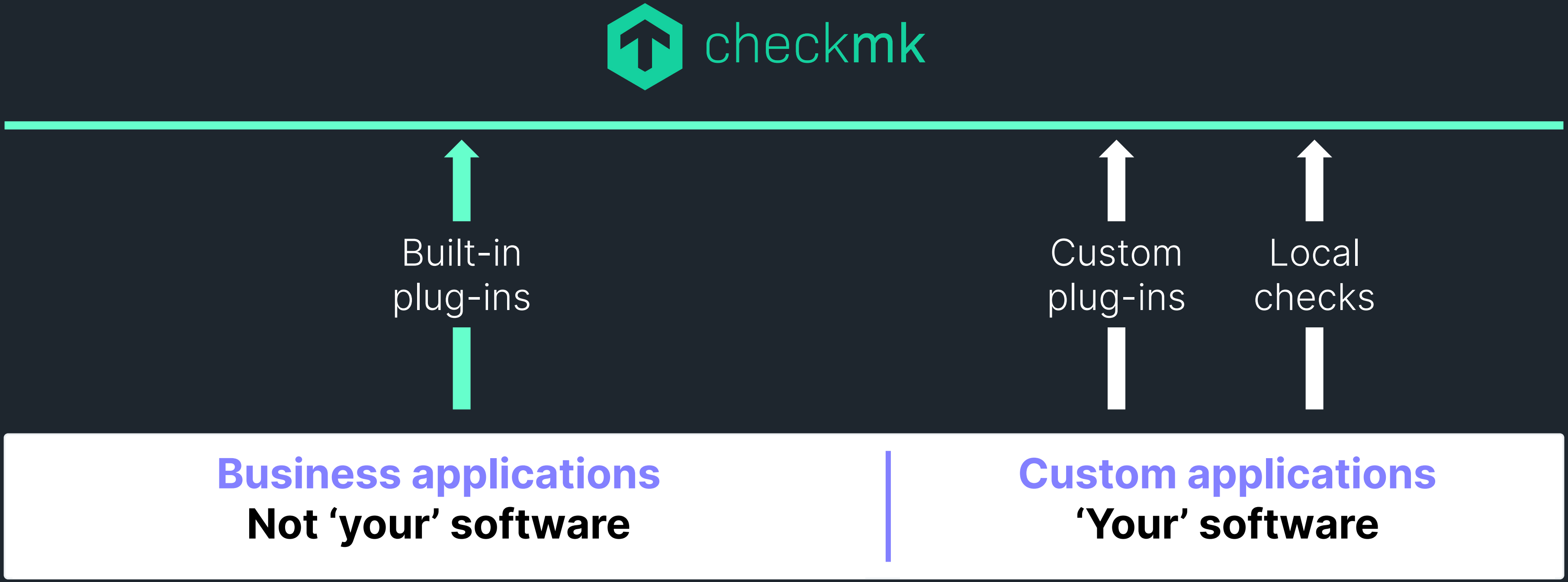
Business applications

Not ‘your’ software

Custom applications

‘Your’ software

Monitoring applications with Checkmk



Standardization being massively pushed

Prometheus

- Large number of exporters generating Prometheus metrics.
- Many applications already expose Prometheus metrics natively.

```
http_server_request_duration_seconds_bucket{method=”get”,path=”/”} 0.0
```

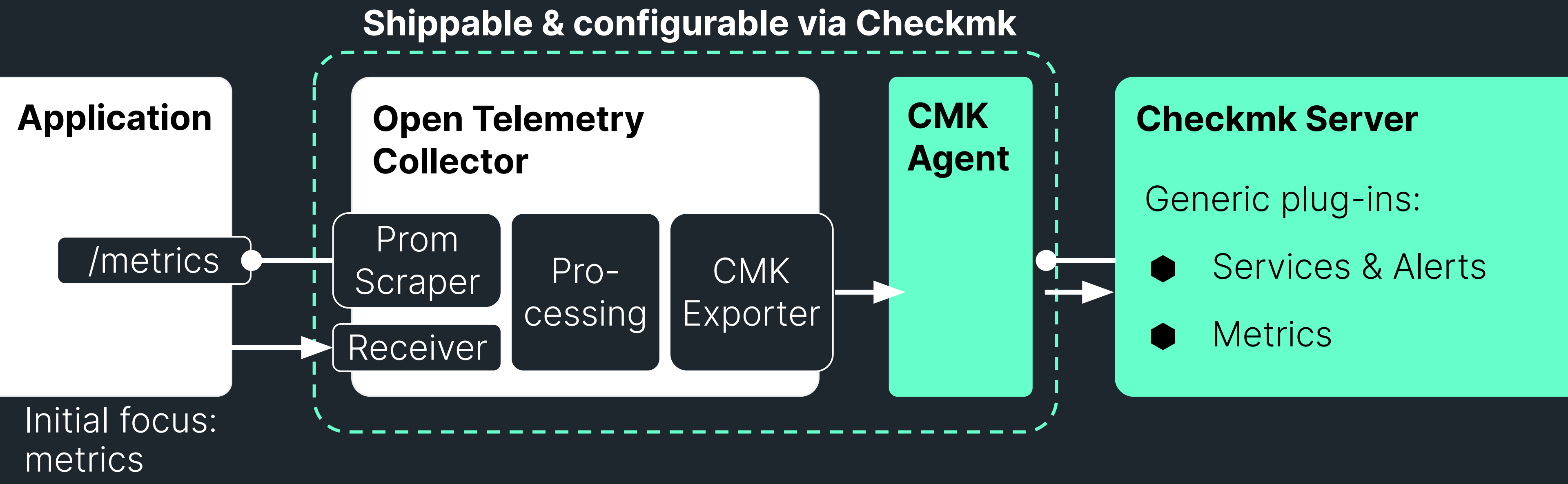
OpenTelemetry

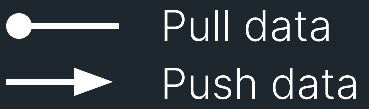
- The (future) standard for observability: metrics, logs and traces.
- For instrumenting, generating, collecting, & exporting telemetry

Makes application monitoring much simpler.

Checkmk OpenTelemetry architecture

Planned for Checkmk 2.4





Checkmk Conference

In-depth application monitoring coming to CMK

Accepted for future release

Monitoring integrations

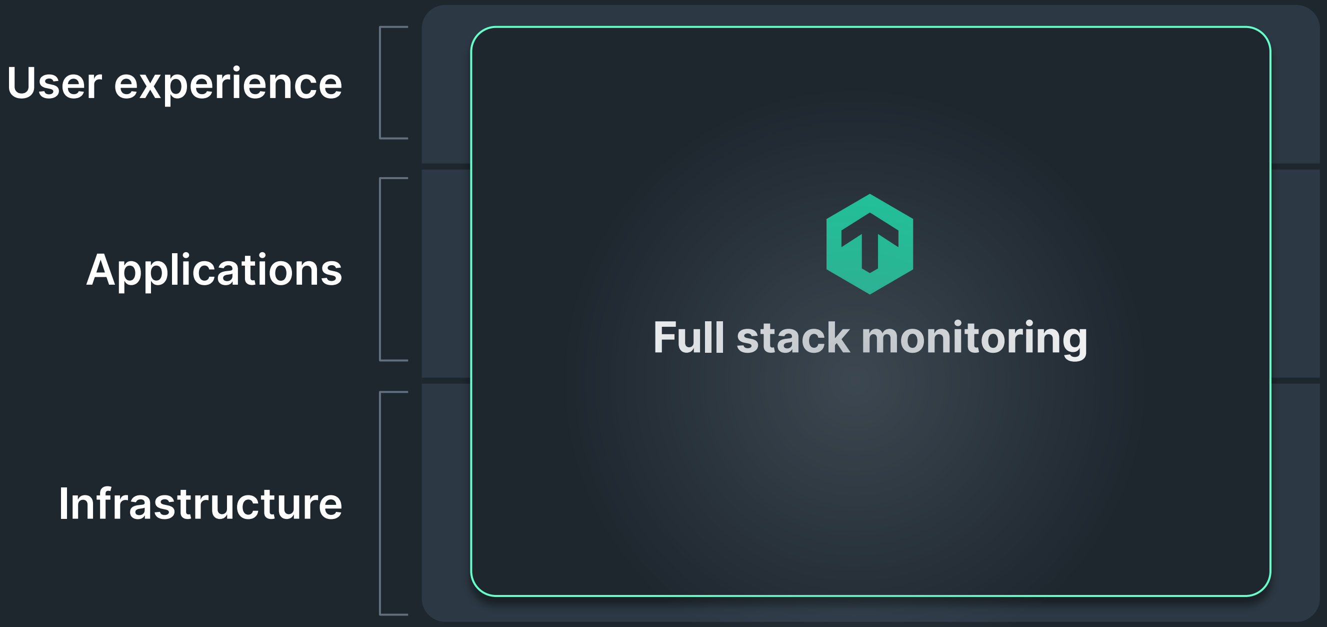
Powerful plug-ins for creating services requiring more complex computations

Specialized plug-ins for typical use cases (e.g. HTTP errors)

Analysis and visualization

Storing of metrics independently of services with new time-series backend

UI for working in a flexible way with those metrics



User experience

Applications

Infrastructure

Full stack monitoring

Our focus for Checkmk 2.4

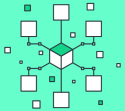
Cloud

Data Center

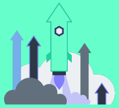
IoT

Great user experience
for beginners and experts

Monitor anything



Powerful configuration

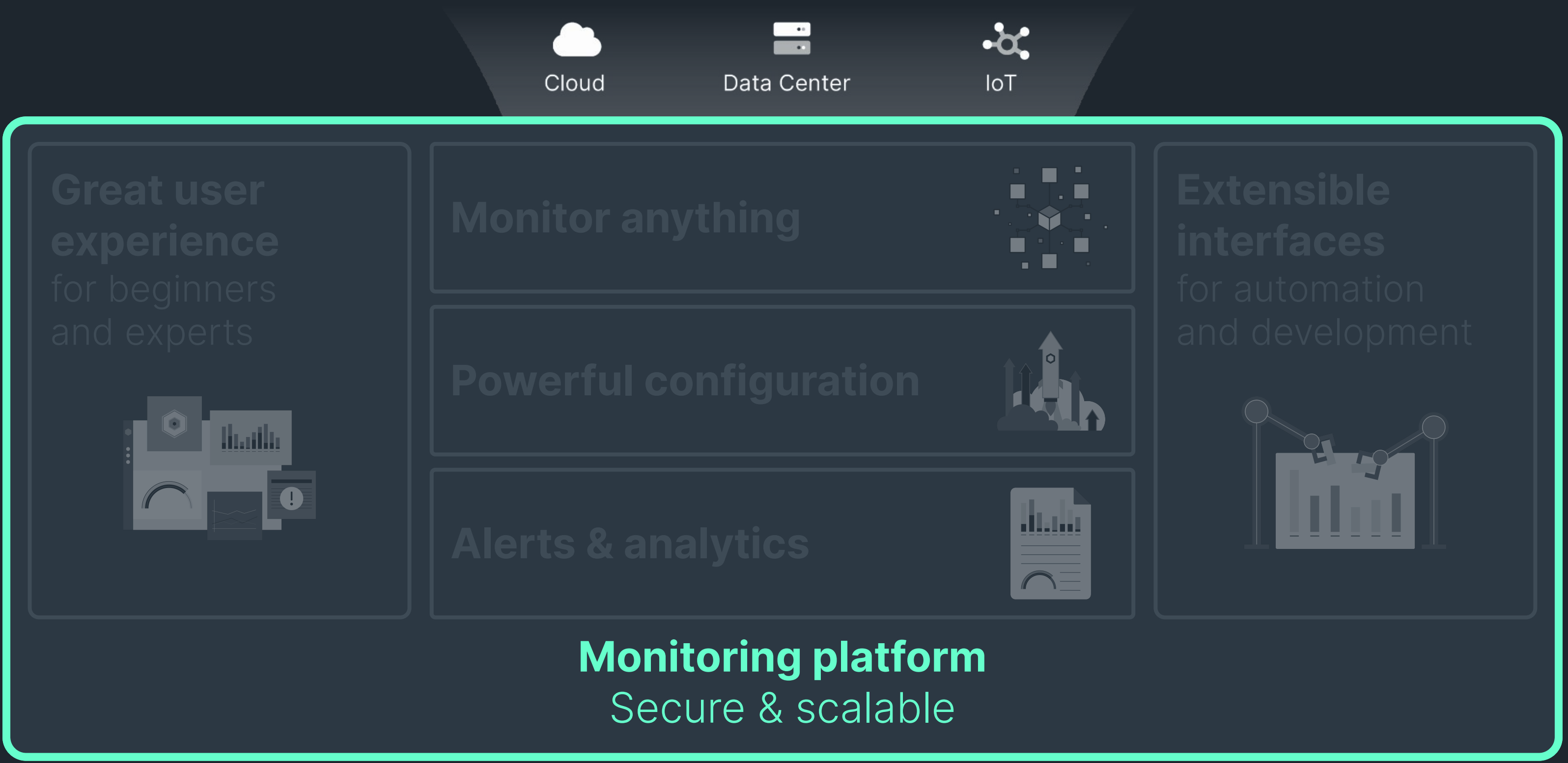


Alerts & analytics

Extensible interfaces
for automation & development

Monitoring platform
Secure & scalable

Our current focus for Checkmk



Cloud

Data Center

IoT

Great user experience
for beginners and experts

Monitor anything

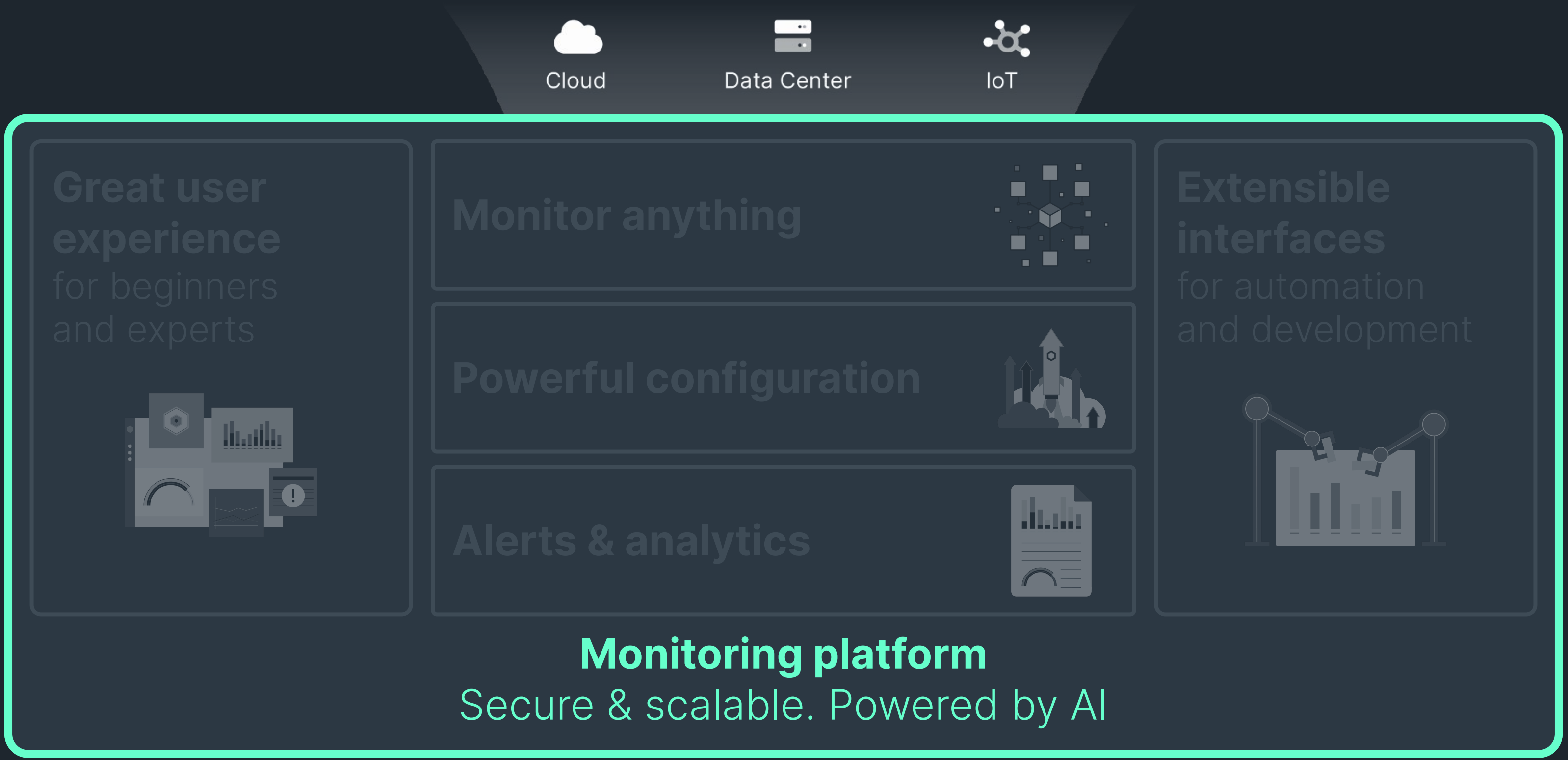
Powerful configuration

Alerts & analytics

Extensible interfaces
for automation and development

Monitoring platform
Secure & scalable

Our current focus for Checkmk



Cloud

Data Center

IoT

Great user experience
for beginners and experts

Monitor anything

Powerful configuration

Alerts & analytics

Extensible interfaces
for automation and development

Monitoring platform
Secure & scalable. Powered by AI

AI powered Checkmk

Predictive monitoring

Natural language interaction

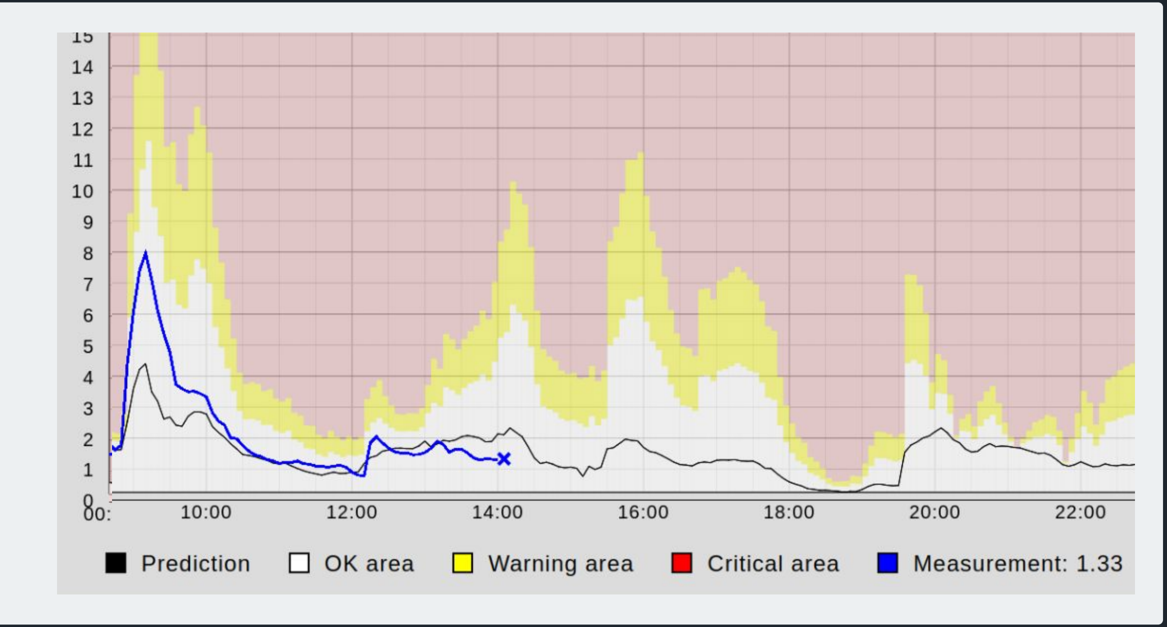
Assisted development

Anomaly detection & correlation

Checkmk
Conference

Predicting individual metrics for intelligent alerts

Accepted for future release

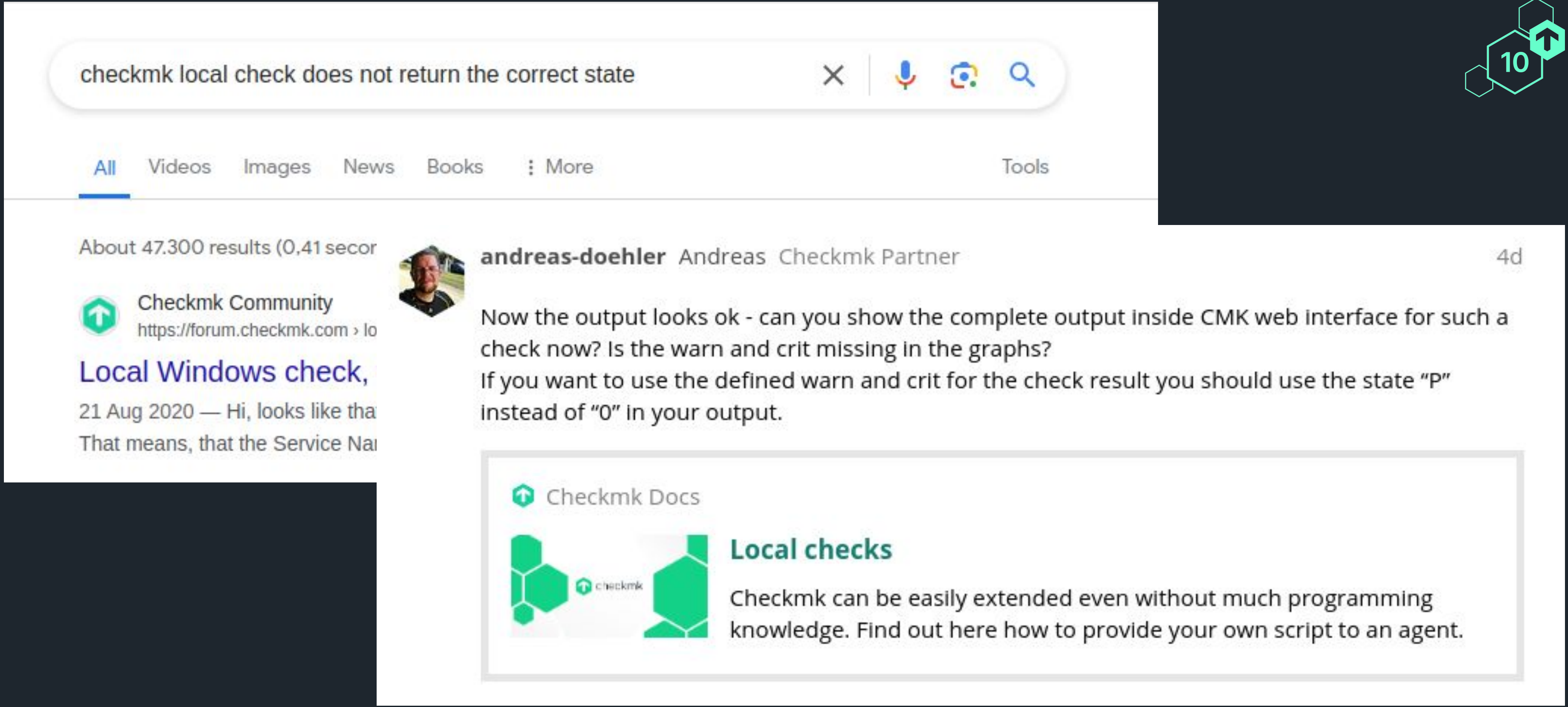


Extend existing capabilities

- Calculates prediction based on historical data learns continuously
- Ideal for detecting spikes, ...

- + **Extend metric coverage**
- + **Improve usability and visualization**

Checkmk Conference



checkmk local check does not return the correct state

About 47.300 results (0,41 secor

Checkmk Community
https://forum.checkmk.com › lo

Local Windows check,

21 Aug 2020 — Hi, looks like tha
That means, that the Service Na

andreas-doehler Andreas Checkmk Partner 4d

Now the output looks ok - can you show the complete output inside CMK web interface for such a check now? Is the warn and crit missing in the graphs?
If you want to use the defined warn and crit for the check result you should use the state “P” instead of “0” in your output.

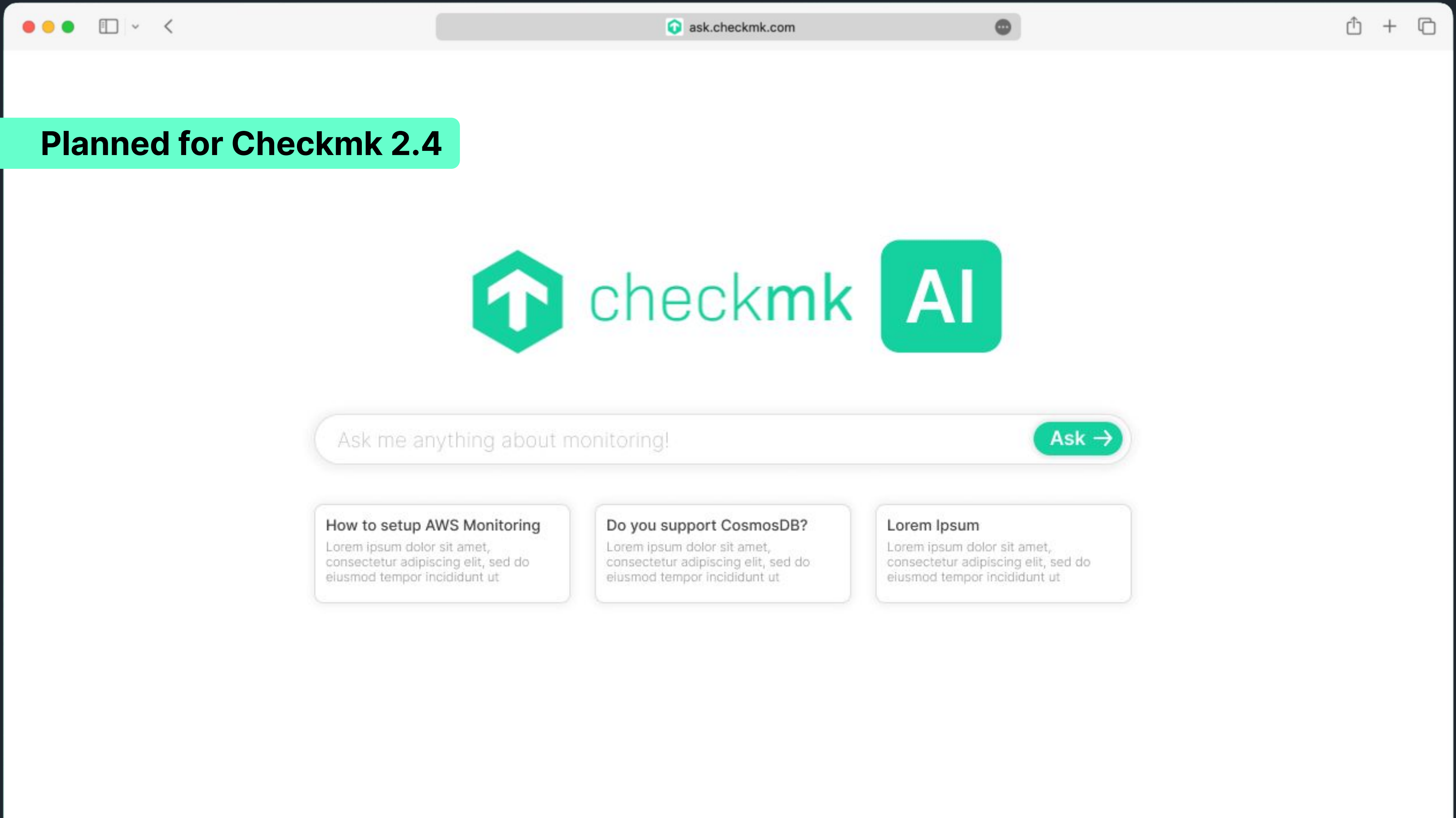
Checkmk Docs

Local checks

Checkmk can be easily extended even without much programming knowledge. Find out here how to provide your own script to an agent.

Imagine your personal ‘Andreas’ …

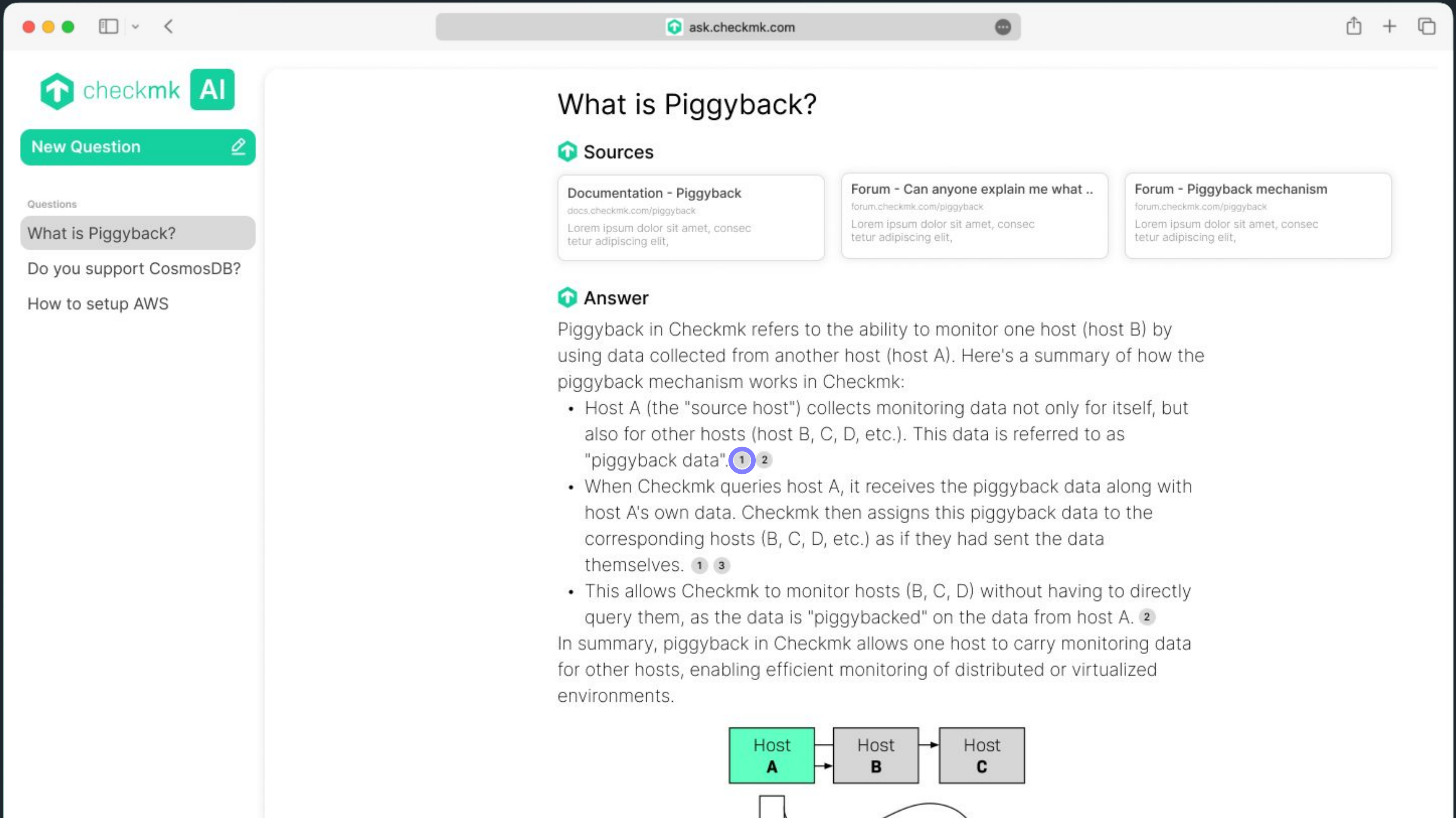
Planned for Checkmk 2.4



How to setup AWS Monitoring
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut

Do you support CosmosDB?
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut

Lorem Ipsum
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut



What is Piggyback?

Sources

Documentation - Piggyback
docs.checkmk.com/piggyback
Lorem ipsum dolor sit amet, consec
tetur adipiscing elit,

Forum - Can anyone explain me what ..
forum.checkmk.com/piggyback
Lorem ipsum dolor sit amet, consec
tetur adipiscing elit,

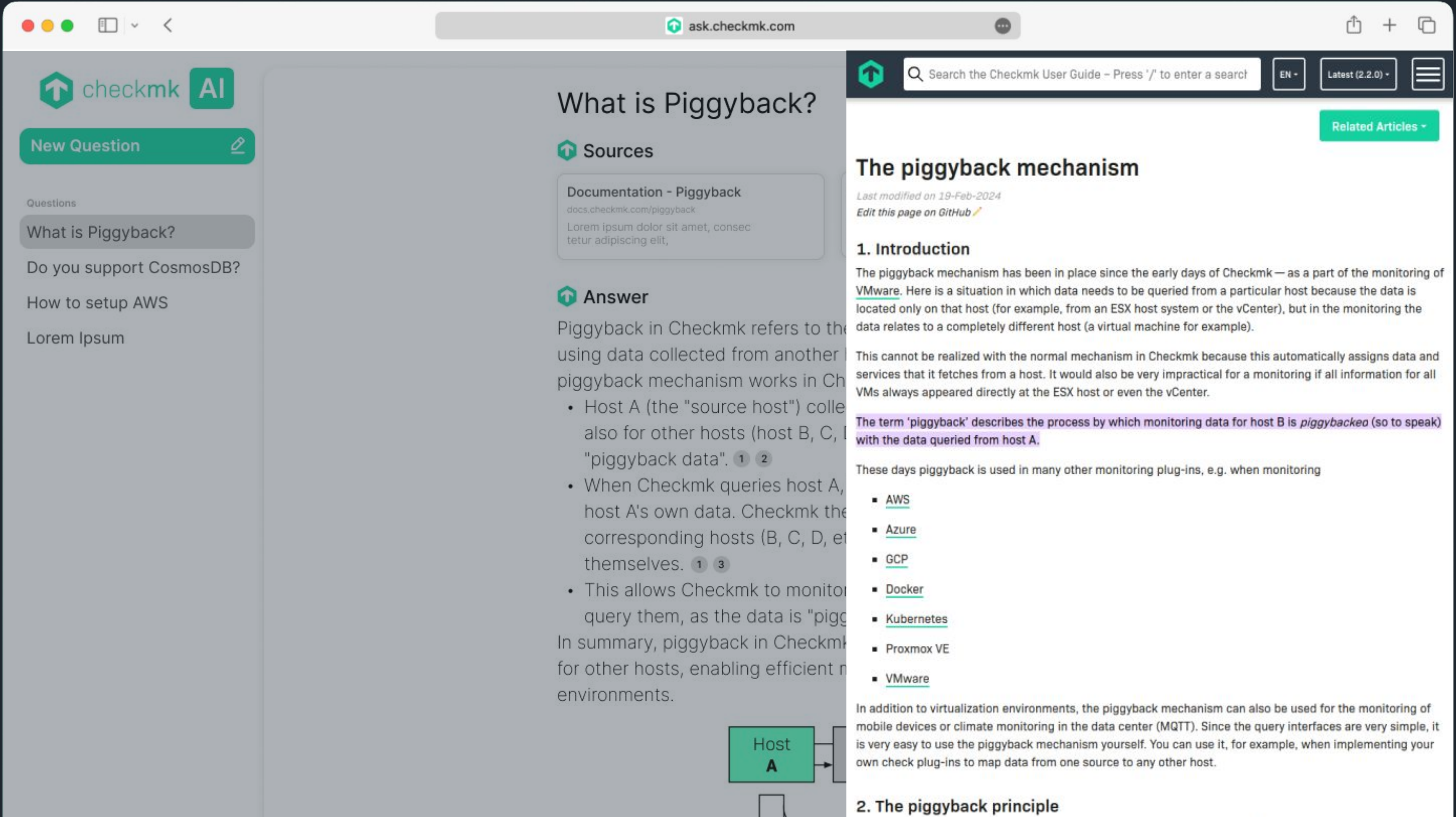
Forum - Piggyback mechanism
forum.checkmk.com/piggyback
Lorem ipsum dolor sit amet, consec
tetur adipiscing elit,

Answer

Piggyback in Checkmk refers to the ability to monitor one host (host B) by using data collected from another host (host A). Here's a summary of how the piggyback mechanism works in Checkmk:

- Host A (the "source host") collects monitoring data not only for itself, but also for other hosts (host B, C, D, etc.). This data is referred to as "piggyback data".[1] [2]
- When Checkmk queries host A, it receives the piggyback data along with host A's own data. Checkmk then assigns this piggyback data to the corresponding hosts (B, C, D, etc.) as if they had sent the data themselves. [1] [3]
- This allows Checkmk to monitor hosts (B, C, D) without having to directly query them, as the data is "piggybacked" on the data from host A. [2]

In summary, piggyback in Checkmk allows one host to carry monitoring data for other hosts, enabling efficient monitoring of distributed or virtualized environments.



What is Piggyback?

Do you support CosmosDB?

How to setup AWS

Lorem Ipsum

What is Piggyback?

Sources

Documentation - Piggyback
docs.checkmk.com/piggyback
Lorem ipsum dolor sit amet, consec tetur adipiscing elit,

Answer

Piggyback in Checkmk refers to the
using data collected from another
piggyback mechanism works in Ch

- Host A (the "source host") colle also for other hosts (host B, C, D "piggyback data". 1 2
- When Checkmk queries host A, host A's own data. Checkmk the corresponding hosts (B, C, D, et themselves. 1 3
- This allows Checkmk to monitor query them, as the data is "pigg

In summary, piggyback in Checkmk for other hosts, enabling efficient m environments.

Host A

The piggyback mechanism

Last modified on 19-Feb-2024
Edit this page on GitHub

1. Introduction

The piggyback mechanism has been in place since the early days of Checkmk — as a part of the monitoring of VMware. Here is a situation in which data needs to be queried from a particular host because the data is located only on that host (for example, from an ESX host system or the vCenter), but in the monitoring the data relates to a completely different host (a virtual machine for example).

This cannot be realized with the normal mechanism in Checkmk because this automatically assigns data and services that it fetches from a host. It would also be very impractical for a monitoring if all information for all VMs always appeared directly at the ESX host or even the vCenter.

The term 'piggyback' describes the process by which monitoring data for host B is *piggybacked* (so to speak) with the data queried from host A.

These days piggyback is used in many other monitoring plug-ins, e.g. when monitoring

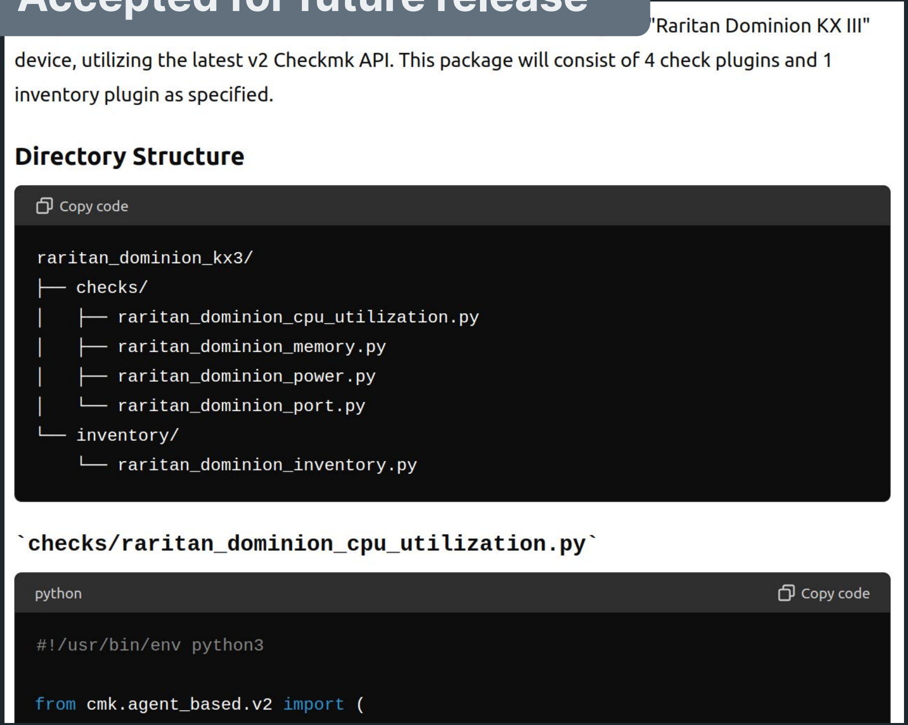
- AWS
- Azure
- GCP
- Docker
- Kubernetes
- Proxmox VE
- VMware

In addition to virtualization environments, the piggyback mechanism can also be used for the monitoring of mobile devices or climate monitoring in the data center (MQTT). Since the query interfaces are very simple, it is very easy to use the piggyback mechanism yourself. You can use it, for example, when implementing your own check plug-ins to map data from one source to any other host.

2. The piggyback principle

Assisted extension development

Accepted for future release



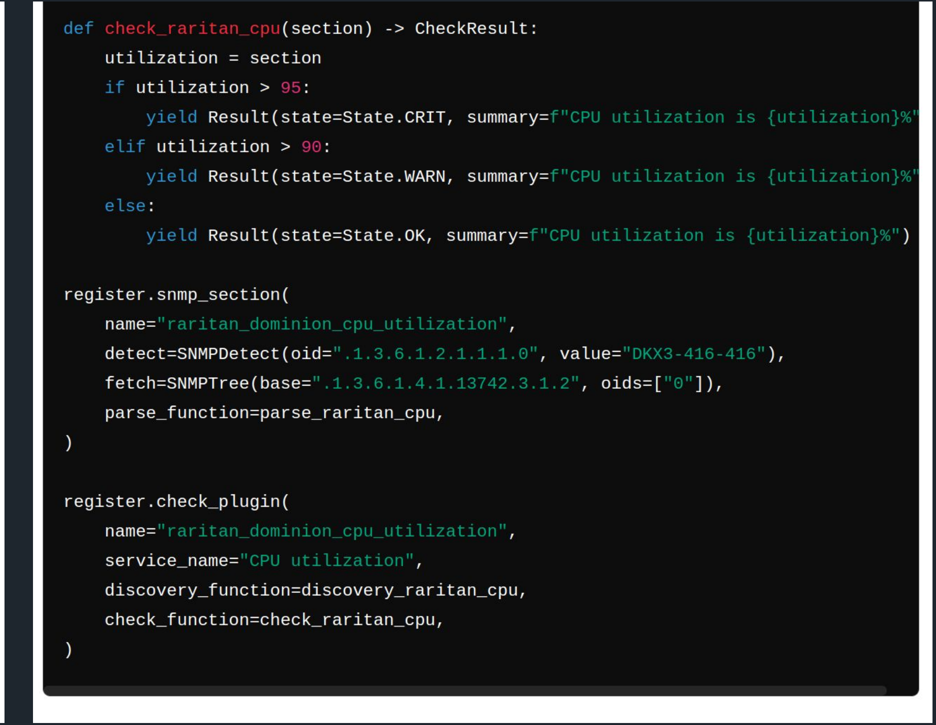
"Raritan Dominion KX III" device, utilizing the latest v2 Checkmk API. This package will consist of 4 check plugins and 1 inventory plugin as specified.

Directory Structure

```
raritan_dominion_kx3/
├── checks/
│   ├── raritan_dominion_cpu_utilization.py
│   ├── raritan_dominion_memory.py
│   ├── raritan_dominion_power.py
│   └── raritan_dominion_port.py
└── inventory/
    └── raritan_dominion_inventory.py
```

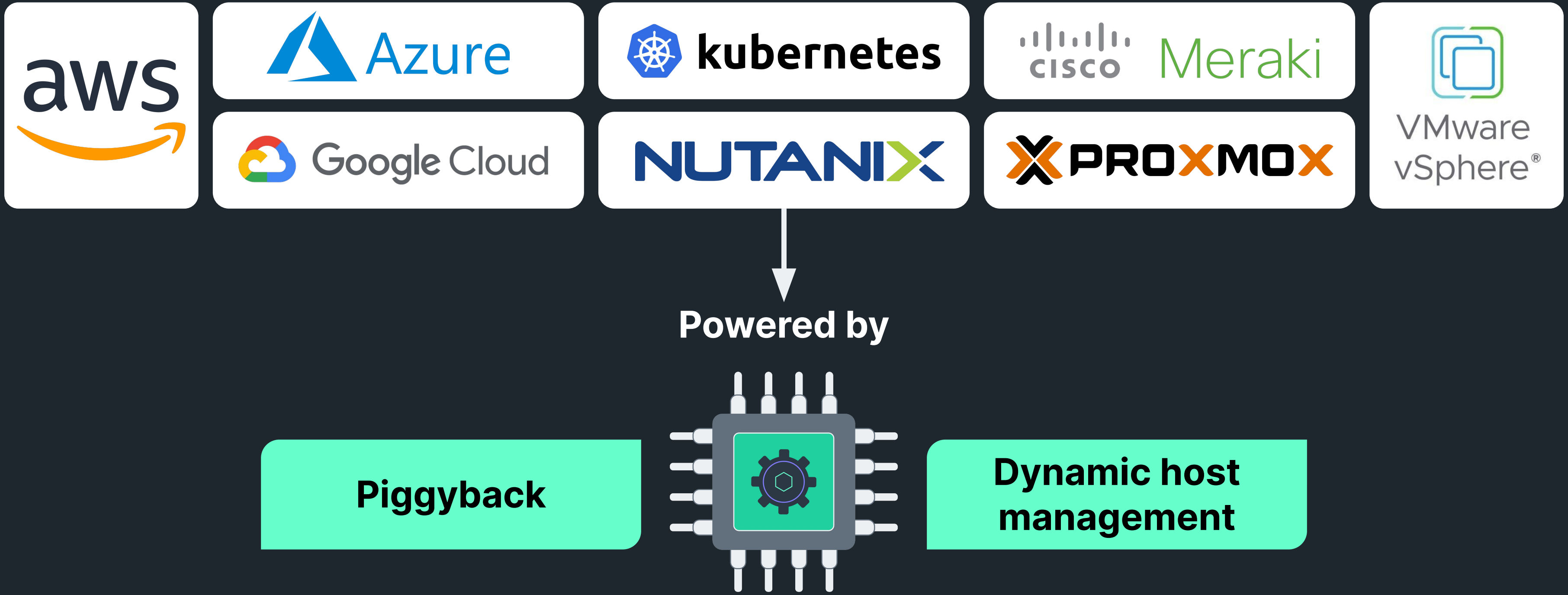
`checks/raritan_dominion_cpu_utilization.py`

```python
#!/usr/bin/env python3

from cmk.agent_based.v2 import (
```



```python
def check_raritan_cpu(section) -> CheckResult:
    utilization = section
    if utilization > 95:
        yield Result(state=State.CRIT, summary=f"CPU utilization is {utilization}%"
    elif utilization > 90:
        yield Result(state=State.WARN, summary=f"CPU utilization is {utilization}%"
    else:
        yield Result(state=State.OK, summary=f"CPU utilization is {utilization}%")

register.snmp_section(
    name="raritan_dominion_cpu_utilization",
    detect=SNMPDetect(oid=".1.3.6.1.2.1.1.1.0", value="DKX3-416-416"),
    fetch=SNMPTree(base=".1.3.6.1.4.1.13742.3.1.2", oids=["0"]),
    parse_function=parse_raritan_cpu,
)

register.check_plugin(
    name="raritan_dominion_cpu_utilization",
    service_name="CPU utilization",
    discovery_function=discovery_raritan_cpu,
    check_function=check_raritan_cpu,
)
```

Monitoring dynamic systems at scale



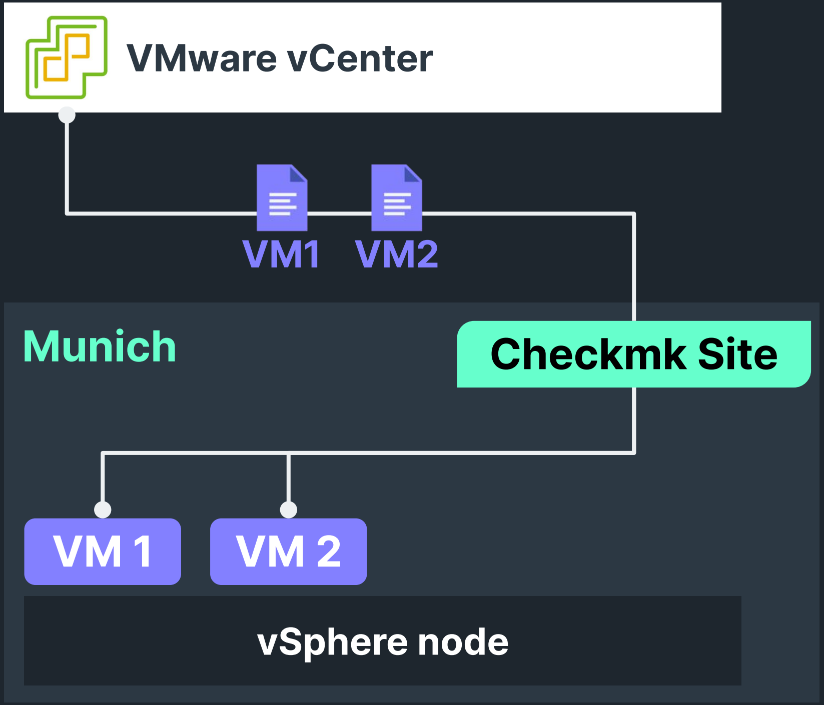
Powered by

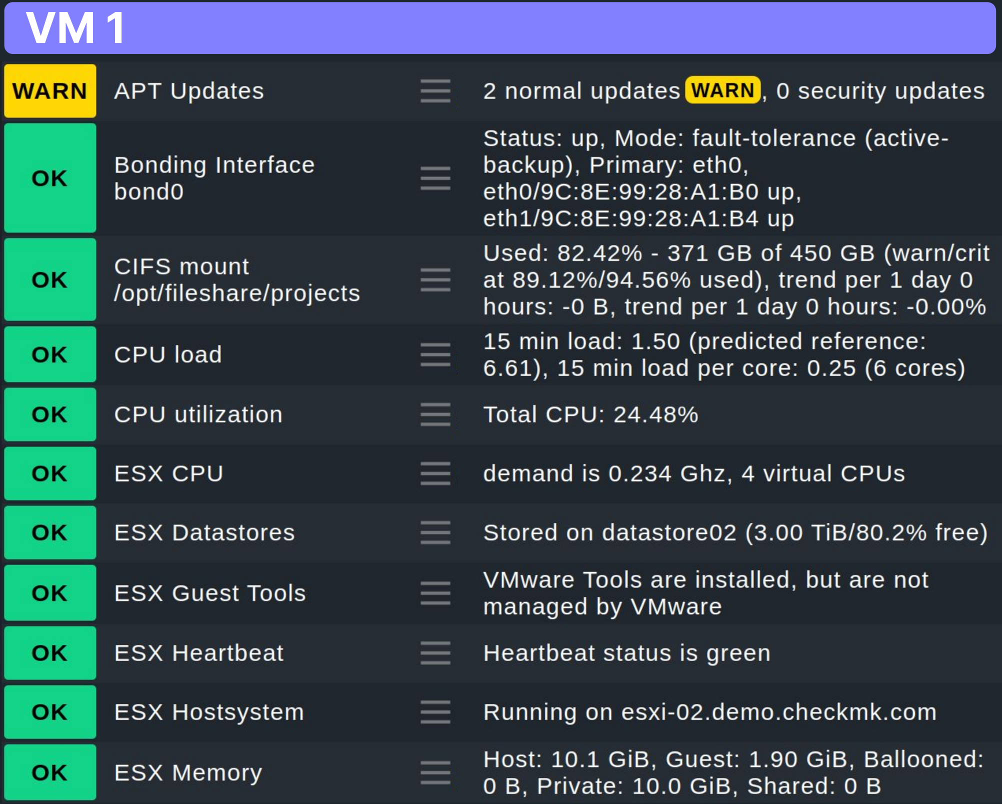
Piggyback

Dynamic host management

Piggyback

Multiple data sources joined together

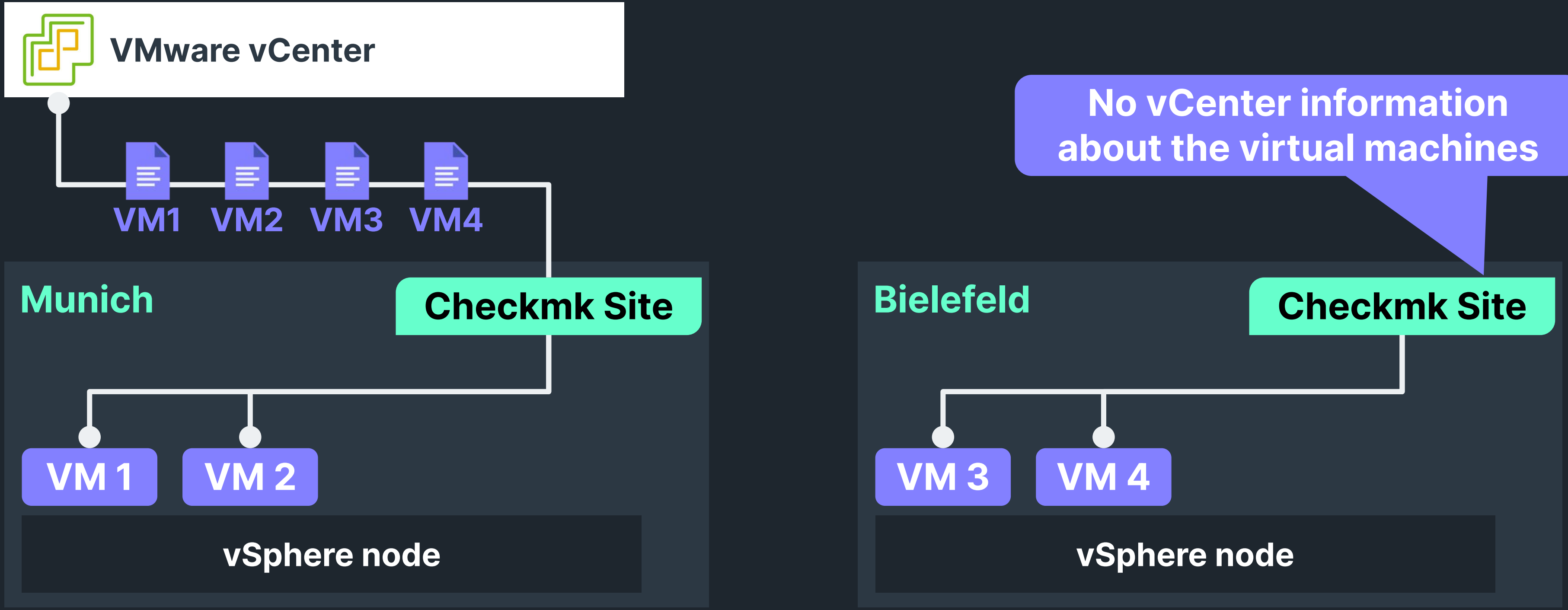




VM 1

WARN	APT Updates	≡	2 normal updates WARN, 0 security updates
OK	Bonding Interface bond0	≡	Status: up, Mode: fault-tolerance (active-backup), Primary: eth0, eth0/9C:8E:99:28:A1:B0 up, eth1/9C:8E:99:28:A1:B4 up
OK	CIFS mount /opt/fileshare/projects	≡	Used: 82.42% - 371 GB of 450 GB (warn/crit at 89.12%/94.56% used), trend per 1 day 0 hours: -0 B, trend per 1 day 0 hours: -0.00%
OK	CPU load	≡	15 min load: 1.50 (predicted reference: 6.61), 15 min load per core: 0.25 (6 cores)
OK	CPU utilization	≡	Total CPU: 24.48%
OK	ESX CPU	≡	demand is 0.234 Ghz, 4 virtual CPUs
OK	ESX Datastores	≡	Stored on datastore02 (3.00 TiB/80.2% free)
OK	ESX Guest Tools	≡	VMware Tools are installed, but are not managed by VMware
OK	ESX Heartbeat	≡	Heartbeat status is green
OK	ESX Hostsystem	≡	Running on esxi-02.demo.checkmk.com
OK	ESX Memory	≡	Host: 10.1 GiB, Guest: 1.90 GiB, Ballooned: 0 B, Private: 10.0 GiB, Shared: 0 B

Piggyback in distributed not shared between sites

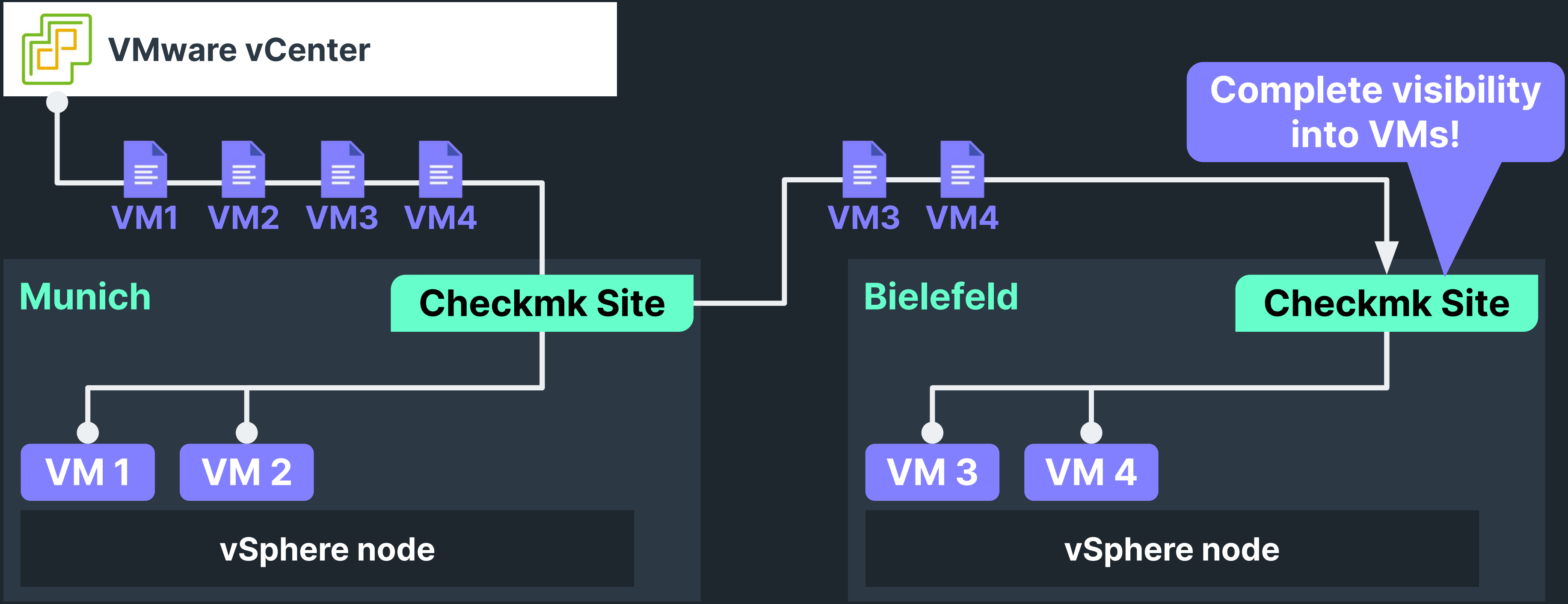


No vCenter information about the virtual machines

Piggyback

Adding distributed piggyback support

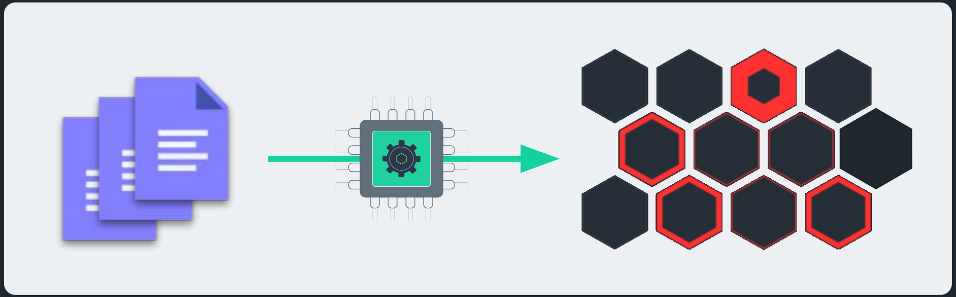
Planned for Checkmk 2.4



Checkmk Conference

Dynamic config

Automatic configuration for dynamic data sources



- Hosts are automatically created with ‘dynamic host management’
- Typical use case: piggyback data

Dynamic config

Improve multi-connection dynamic config handling

Planned for Checkmk 2.4

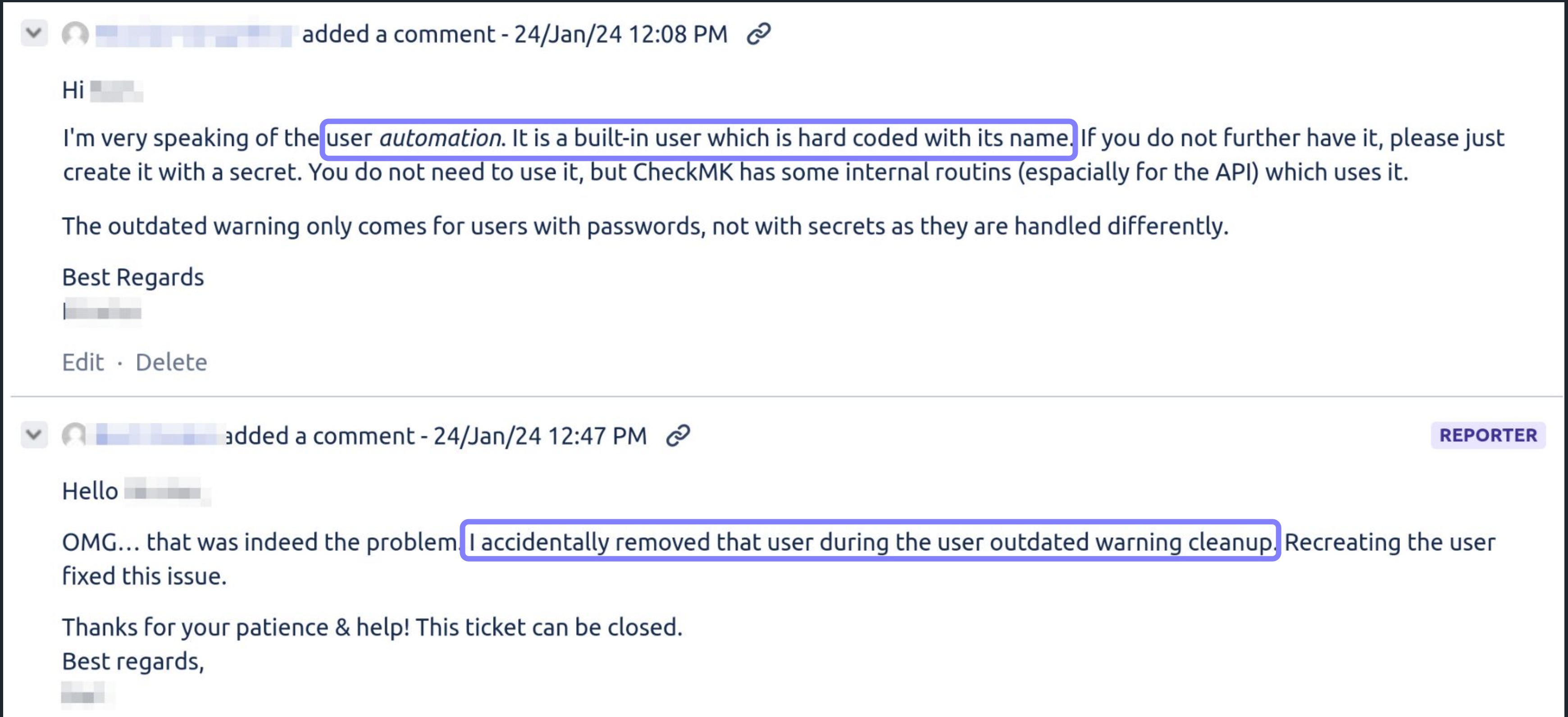
Status quo

- Dynamic configuration daemon does all actions in one execution (create/delete hosts, discover services, activate changes)
- Multiple connections run simultaneously
  - All connections start simultaneously at site start

Planned changes

- Parallel data collection
- Host updates handled in dedicated queues for
  - Create/delete hosts
  - Discover services
  - Activate changes
- Sequential processing within these queues

The (in)famous automation user



[redacted] added a comment - 24/Jan/24 12:08 PM

Hi [redacted]

I'm very speaking of the user *automation*. It is a built-in user which is hard coded with its name. If you do not further have it, please just create it with a secret. You do not need to use it, but CheckMK has some internal routins (espacially for the API) which uses it.

The outdated warning only comes for users with passwords, not with secrets as they are handled differently.

Best Regards
[redacted]

[redacted] added a comment - 24/Jan/24 12:47 PM

Hello [redacted]

OMG… that was indeed the problem. I accidentally removed that user during the user outdated warning cleanup. Recreating the user fixed this issue.

Thanks for your patience & help! This ticket can be closed.
Best regards,
[redacted]

No more dependency to automation user

Planned for Checkmk 2.4

Origin of the automation user

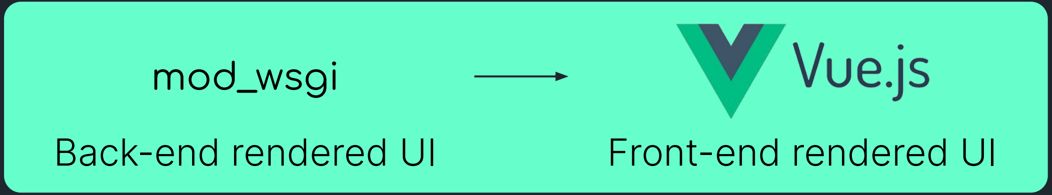
- Enable local automations configuration-less
  - Internally for Checkmk
  - Also for use by users
- Managed by user, but required internally by Checkmk: not great ...
- Used in Dynamic configuration, agent registration, ...

Planned changes

- Replace the mechanism and currently required internal authentication
  - No internal requirement for an automation user anymore
- automation.secret file will no longer be necessary

New UI technology powering a better Checkmk

Planned for Checkmk 2.4



mod_wsgi → Vue.js

Back-end rendered UI

Front-end rendered UI

Improved usability & performance

- We can build UI features faster
- We can build more powerful features (example: multi-step workflows)
- We can build a faster UI

First impact on users in 2.4

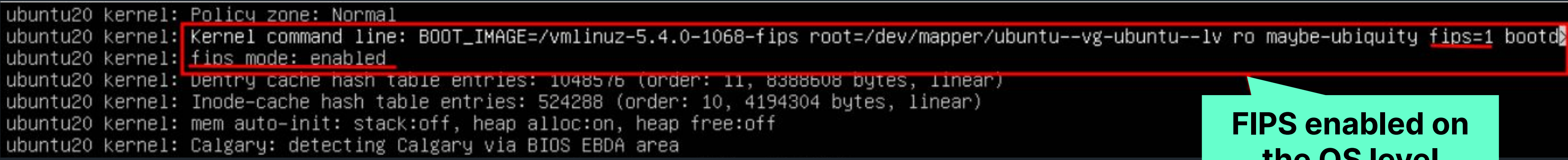
- The graph editor can be operated much more smoothly
- Tables can be searched and sorted more quickly
- Forms give feedback more quickly

FIPS compliance. What is FIPS?

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

Federal Information Processing Standards (FIPS)

- Security standards for federal and defense cybersecurity compliance, specifically focusing on data encryption.
- Mandatory for US federal agencies, institutions that receive federal funding, and many more



```
ubuntu20 kernel: Policy zone: Normal
ubuntu20 kernel: Kernel command line: BOOT_IMAGE=/vmlinuz-5.4.0-1068-fips root=/dev/mapper/ubuntu--vg-ubuntu--lv ro maybe-ubiquity fips=1 bootd
ubuntu20 kernel: fips mode: enabled
ubuntu20 kernel: Dentry cache hash table entries: 1048576 (order: 11, 8388608 bytes, linear)
ubuntu20 kernel: Inode-cache hash table entries: 524288 (order: 10, 4194304 bytes, linear)
ubuntu20 kernel: mem auto-init: stack:off, heap alloc:on, heap free:off
ubuntu20 kernel: Calgary: detecting Calgary via BIOS EBDA area
```

FIPS enabled on the OS level

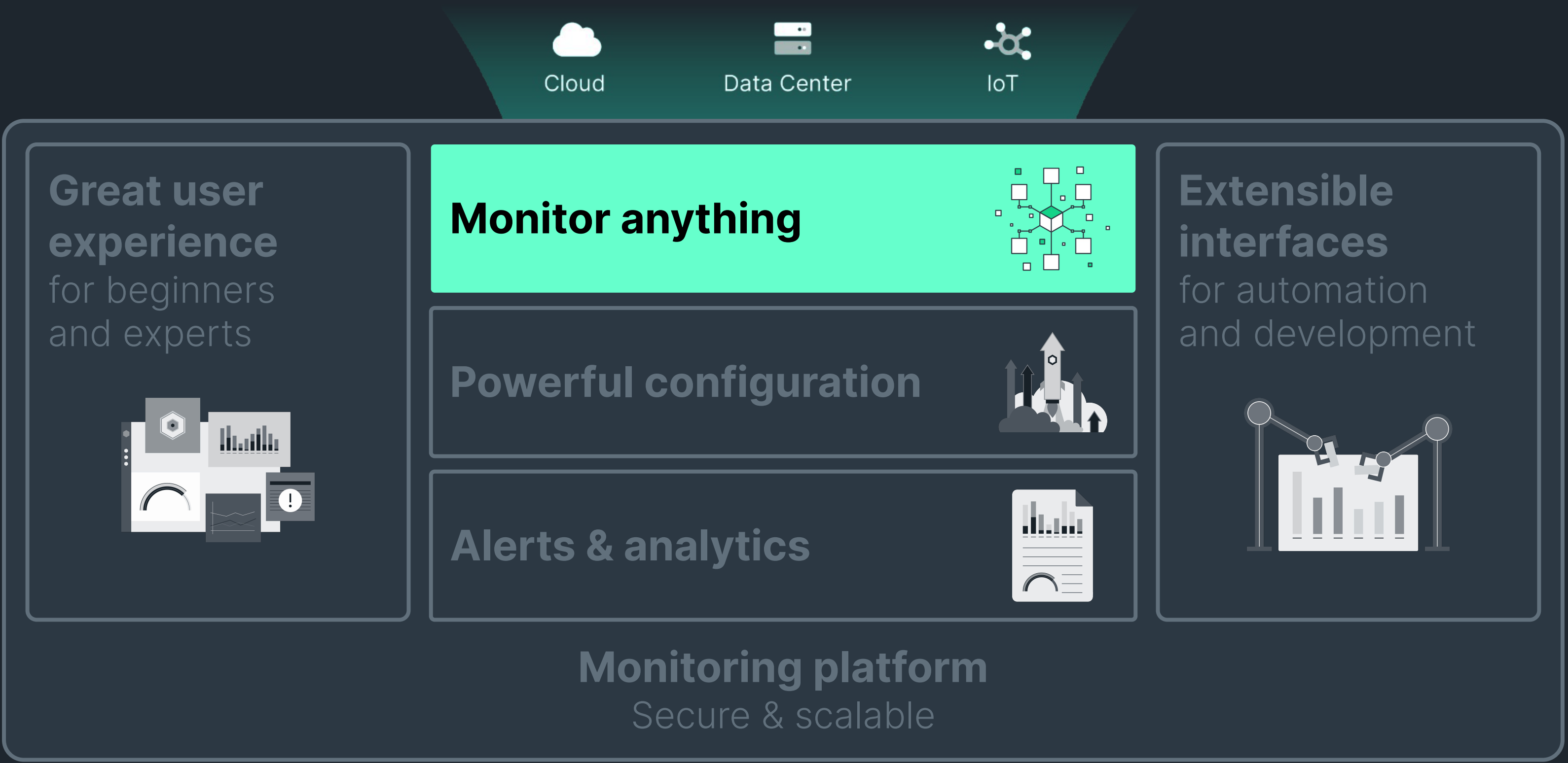
Checkmk should run on FIPS-enabled OS

Planned for Checkmk 2.4

Replace non-supported algorithms like MD5

Done	Agent encryption, livestatus, notification spooler, password hashes, cookie signing, ...
Open	For monitoring server: signature of PDF reports (3rd party library) For agents: signature of baked agents (workaround possible) And anything else preventing Checkmk agents and server to run in a FIPS-enabled OS.

Our current focus for Checkmk



Cloud

Data Center

IoT

Monitor anything

Powerful configuration

Alerts & analytics

Great user experience for beginners and experts

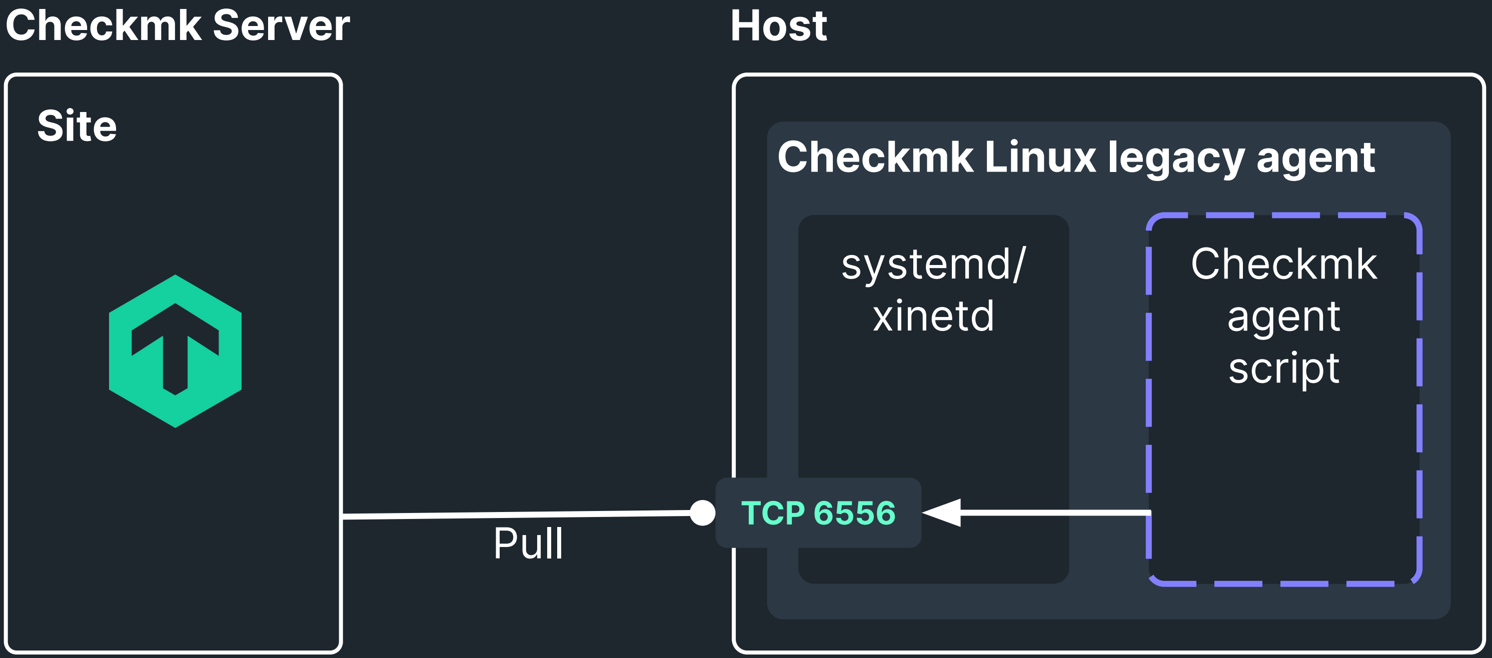
Extensible interfaces for automation and development

Monitoring platform

Secure & scalable

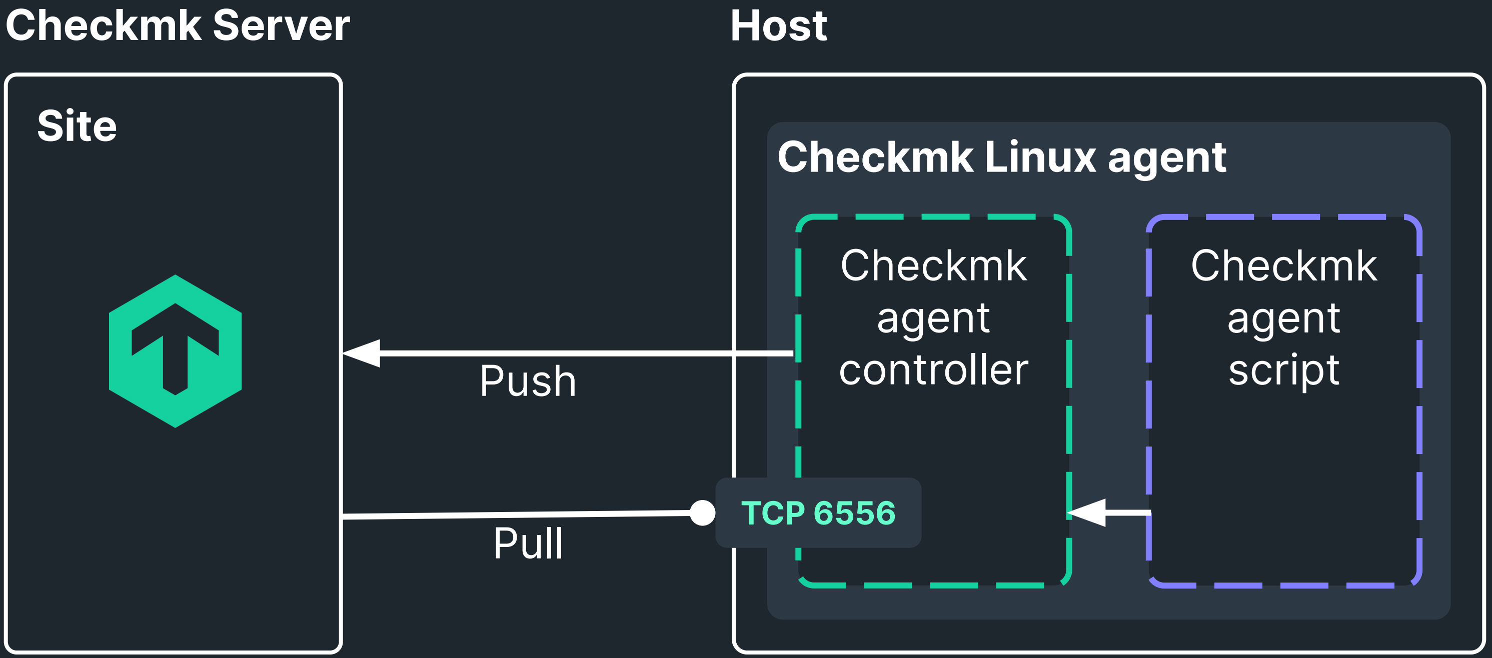
History of the Checkmk agent

Secure as not receiving any incoming data from network



History of the Checkmk agent

Secure as not receiving any incoming data from network



Checkmk Conference

[ ] Executed as root

[ ] Executed as cmk-agent (unprivileged)

New mode: ‘non-root agent’

Agent can be run as unprivileged user

Agent can be updated as unprivileged user

Our path to 'non-root' agent

Planned for Checkmk 2.4

Non-root capable agent script and agent plug-ins

Run commands via sudo

No dependency to package manager

Agent updates using tarballs with less comfort features

Simple controls

Common configurable top level directory for all files

Better integration between server and agents

Current mode (Checkmk 2.3)

Agent controller

```
cmk-agent-ctl register
```

Agent controller registration

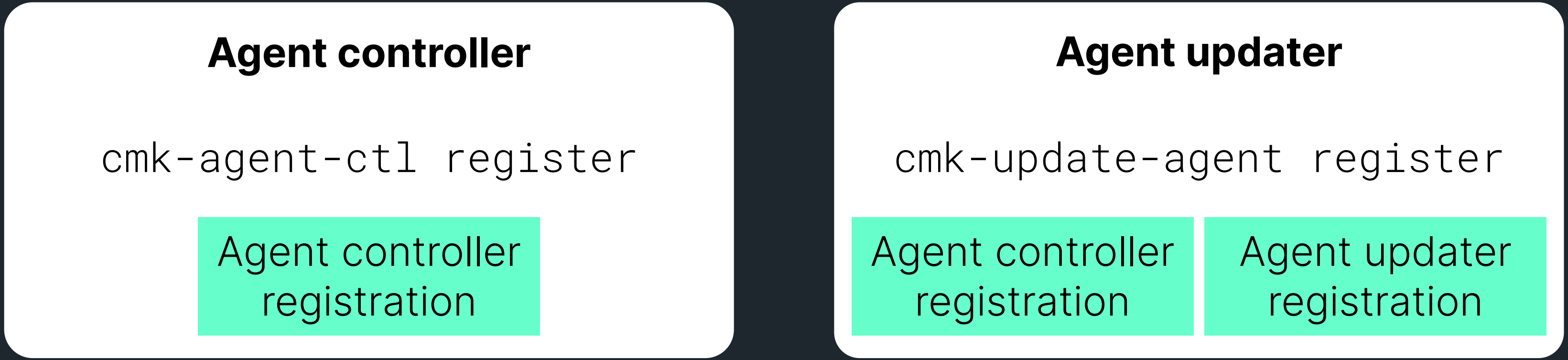
Agent updater

```
cmk-update-agent register
```

Agent updater registration

Better integration between server and agents

Planned for Checkmk 2.4



Agent controller

```
cmk-agent-ctl register
```

Agent controller registration

Agent updater

```
cmk-update-agent register
```

Agent controller registration

Agent updater registration

Agent updater can use both registrations

Better integration between server and agents

Accepted for future release

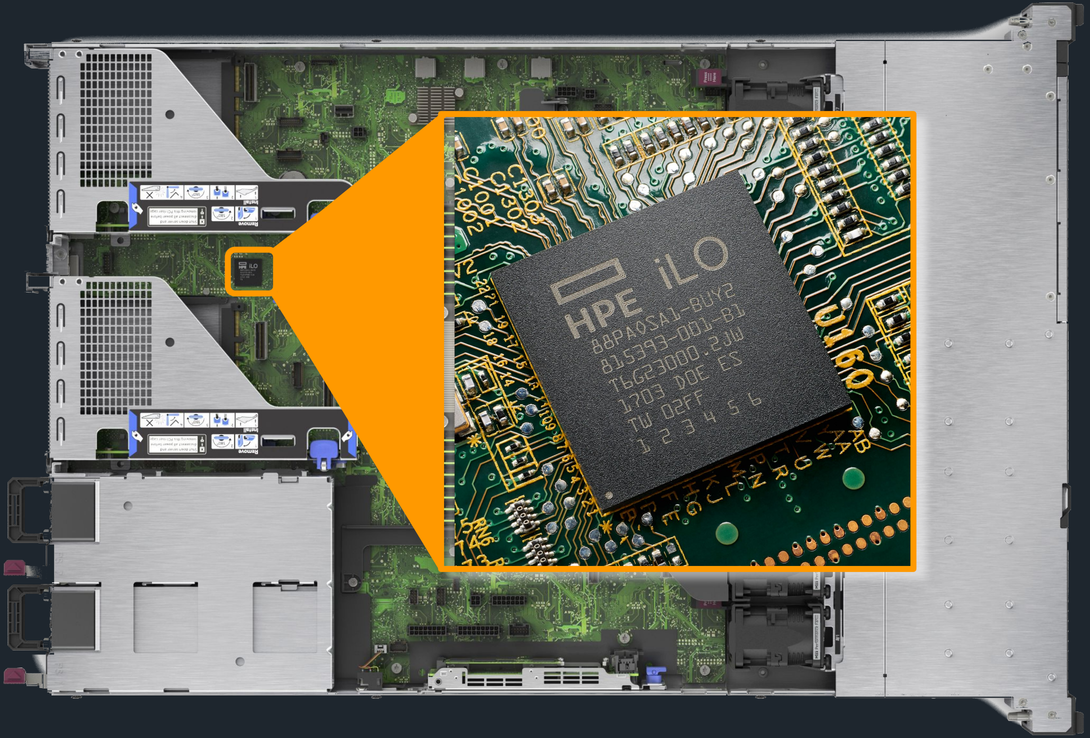
Agent controller

All features of current controller and updater

- Runs on all systems (incl. AIX and Solaris)
- Considers use cases updater currently handles via xinetd/ssh

Steps towards the ‘one’ agent

Out-of-band management and monitoring

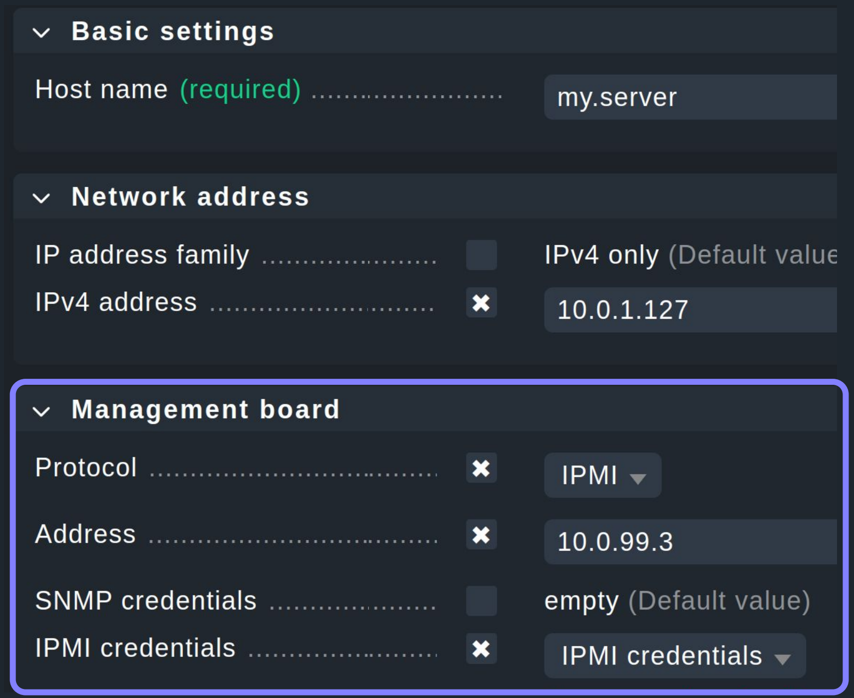


Baseboard management controller

- For remote management, e.g. if server is turned off / crashed
- Often embedded on server, but **independent component** with dedicated network connection
- Provides hardware sensor information (fan speed, power voltages, chassis intrusion)

Monitoring baseboard management controllers

Integrated monitoring via host properties



Basic settings

Host name (required) .... my.server

Network address

IP address family .... □ IPv4 only (Default value

IPv4 address .... ☒ 10.0.1.127

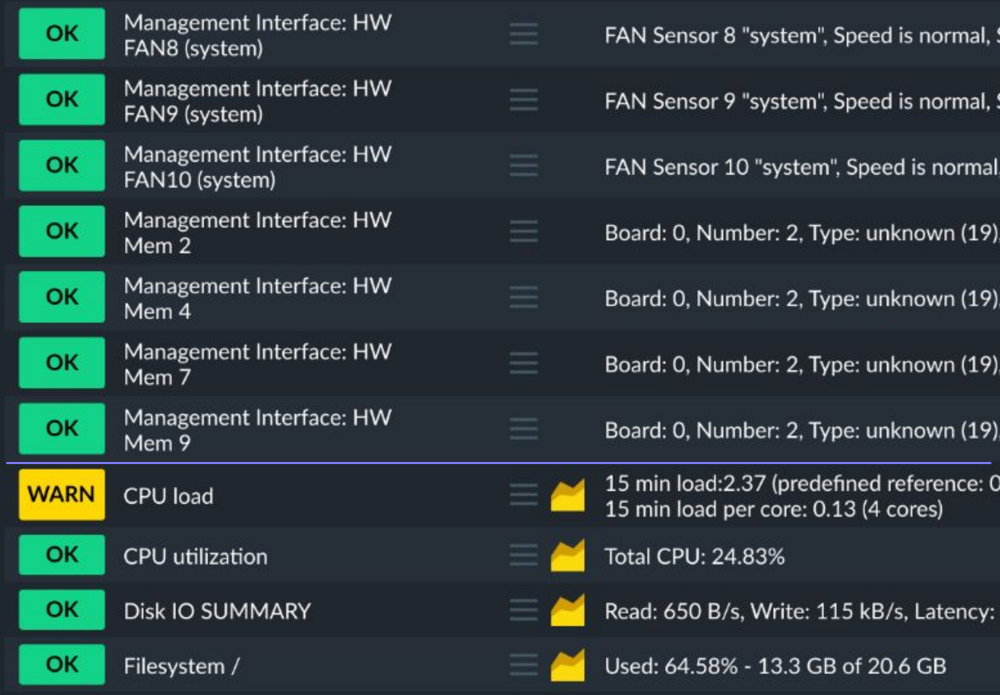
Management board

Protocol .... ☒ IPMI

Address .... ☒ 10.0.99.3

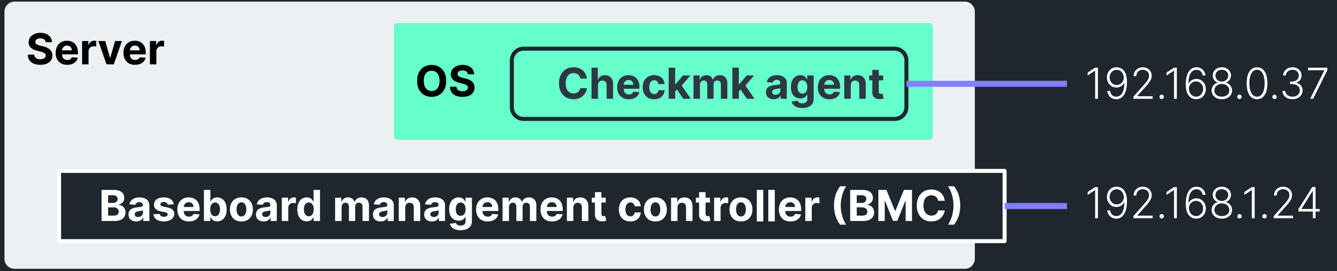
SNMP credentials .... □ empty (Default value)

IPMI credentials .... ☒ IPMI credentials



OK	Management Interface: HW FAN8 (system)	FAN Sensor 8 "system", Speed is normal, S
OK	Management Interface: HW FAN9 (system)	FAN Sensor 9 "system", Speed is normal, S
OK	Management Interface: HW FAN10 (system)	FAN Sensor 10 "system", Speed is normal
OK	Management Interface: HW Mem 2	Board: 0, Number: 2, Type: unknown (19)
OK	Management Interface: HW Mem 4	Board: 0, Number: 2, Type: unknown (19)
OK	Management Interface: HW Mem 7	Board: 0, Number: 2, Type: unknown (19)
OK	Management Interface: HW Mem 9	Board: 0, Number: 2, Type: unknown (19)
WARN	CPU load	15 min load:2.37 (predefined reference: 0 15 min load per core: 0.13 (4 cores)
OK	CPU utilization	Total CPU: 24.83%
OK	Disk IO SUMMARY	Read: 650 B/s, Write: 115 kB/s, Latency:
OK	Filesystem /	Used: 64.58% - 13.3 GB of 20.6 GB

Two different objects shouldn’t be one host



UP	pve-nyc-001.unified.com

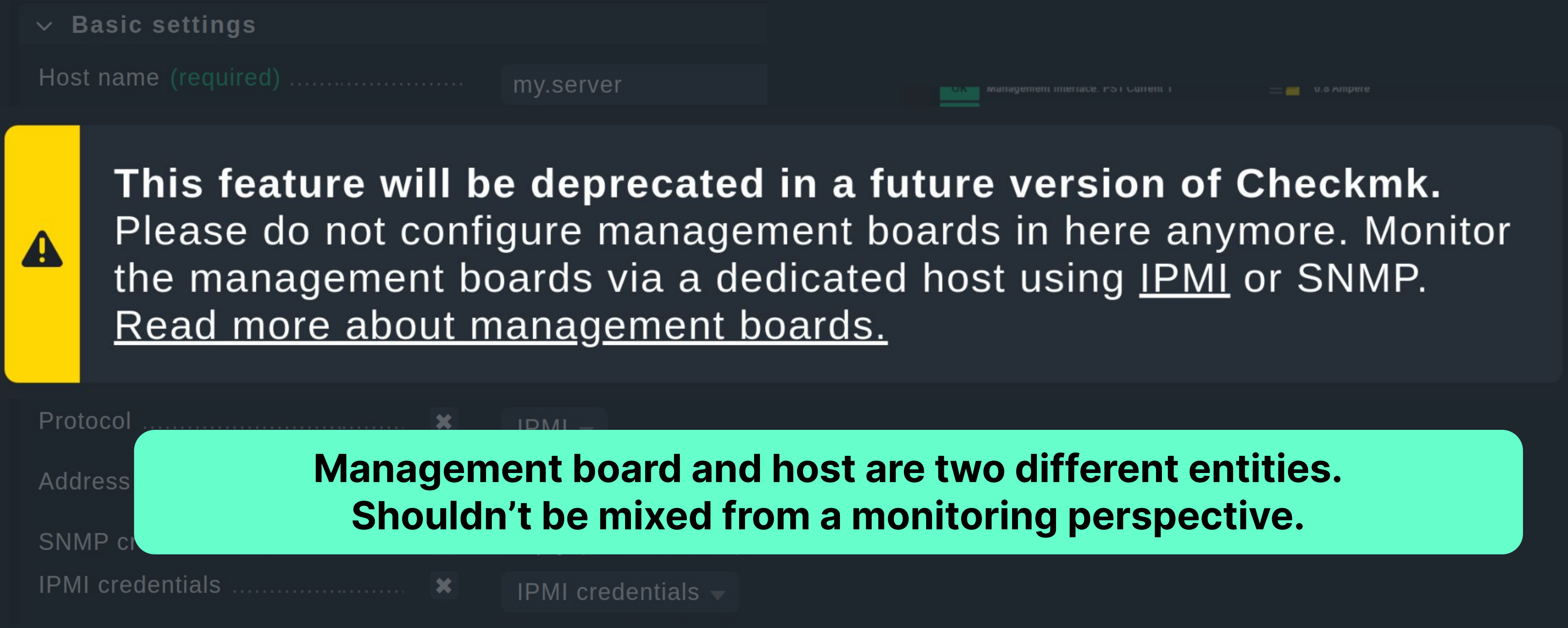
- Unclear monitoring state: what about unreachable OS vs. BMC?
- Dealing with multiple OS complex: duplicate services

UP	pve-nyc-001-ipmi.split.com
DOWN	pve-nyc-001.split.com

- One can be DOWN, while the other is UP

Monitoring baseboard management controllers

Integrated monitoring via host properties



Basic settings

Host name (required) .... my.server

OK Management interface: PSU Current 1 0.8 Ampere

This feature will be deprecated in a future version of Checkmk. Please do not configure management boards in here anymore. Monitor the management boards via a dedicated host using IPMI or SNMP. Read more about management boards.

Protocol .... ✖ IPMI

Address

SNMP cr

IPMI credentials .... ✖ IPMI credentials

Management board and host are two different entities. Shouldn't be mixed from a monitoring perspective.

Checkmk Conference

Recommended path going forward

Monitor as dedicated host

- Dedicated host via SNMP
- Dedicated host with special agent (IPMI Sensors via Freeipmi or IPMItool)
- Dedicated host with new built-in Redfish MKP
  - Generic special agent for all modern out-of-band management hardware
  - Mainlined for Checkmk 2.4

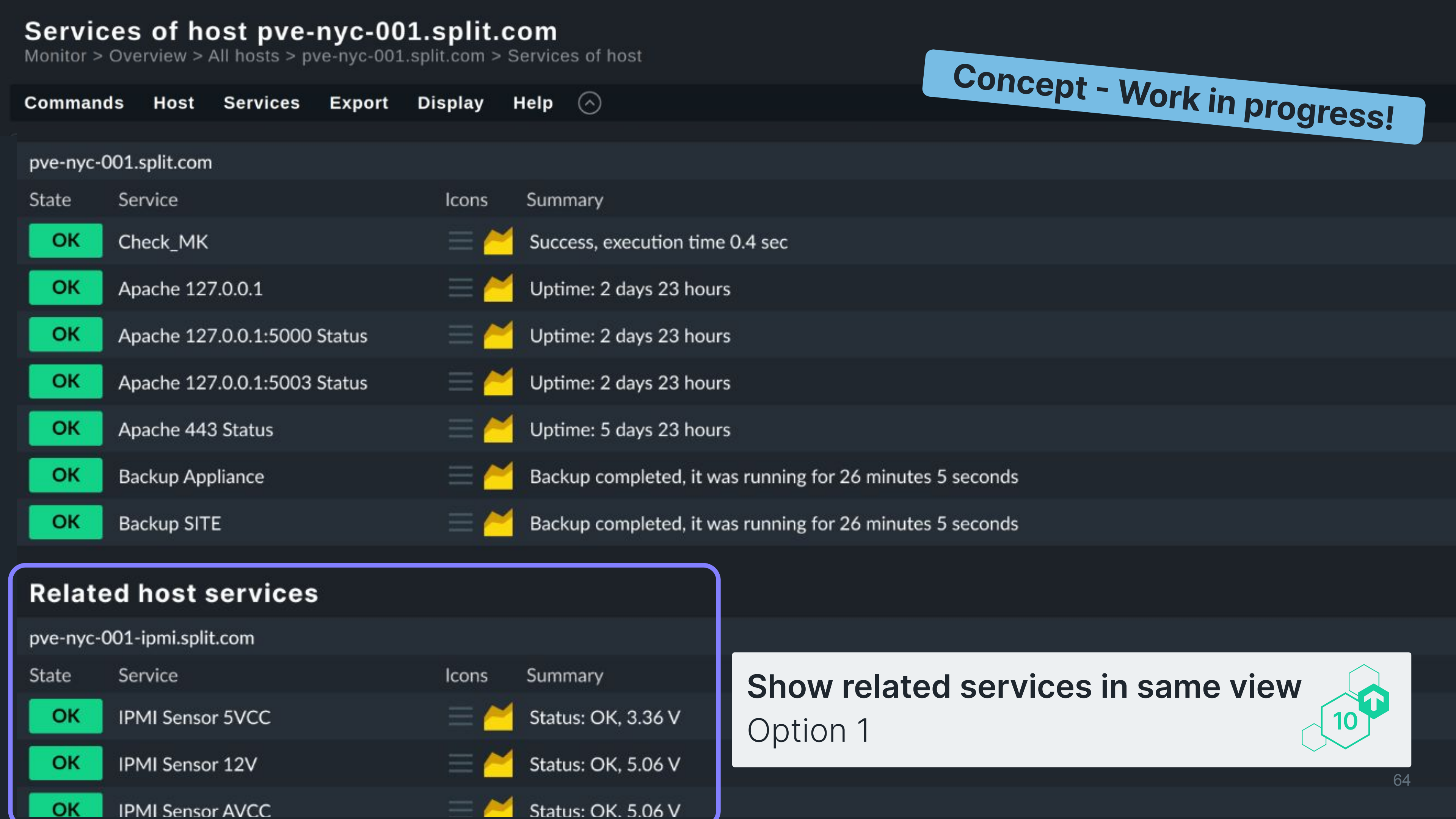
Drawback

- Server (hardware) and OS will appear as two different hosts.

→ **Need to solve underlying need: Still have information in one place**

Planned for Checkmk 2.4

Concept - Work in progress!



Services of host pve-nyc-001.split.com

Monitor > Overview > All hosts > pve-nyc-001.split.com > Services of host

pve-nyc-001.split.com

State	Service	Icons	Summary
OK	Check_MK	≡	Success, execution time 0.4 sec
OK	Apache 127.0.0.1	≡	Uptime: 2 days 23 hours
OK	Apache 127.0.0.1:5000 Status	≡	Uptime: 2 days 23 hours
OK	Apache 127.0.0.1:5003 Status	≡	Uptime: 2 days 23 hours
OK	Apache 443 Status	≡	Uptime: 5 days 23 hours
OK	Backup Appliance	≡	Backup completed, it was running for 26 minutes 5 seconds
OK	Backup SITE	≡	Backup completed, it was running for 26 minutes 5 seconds

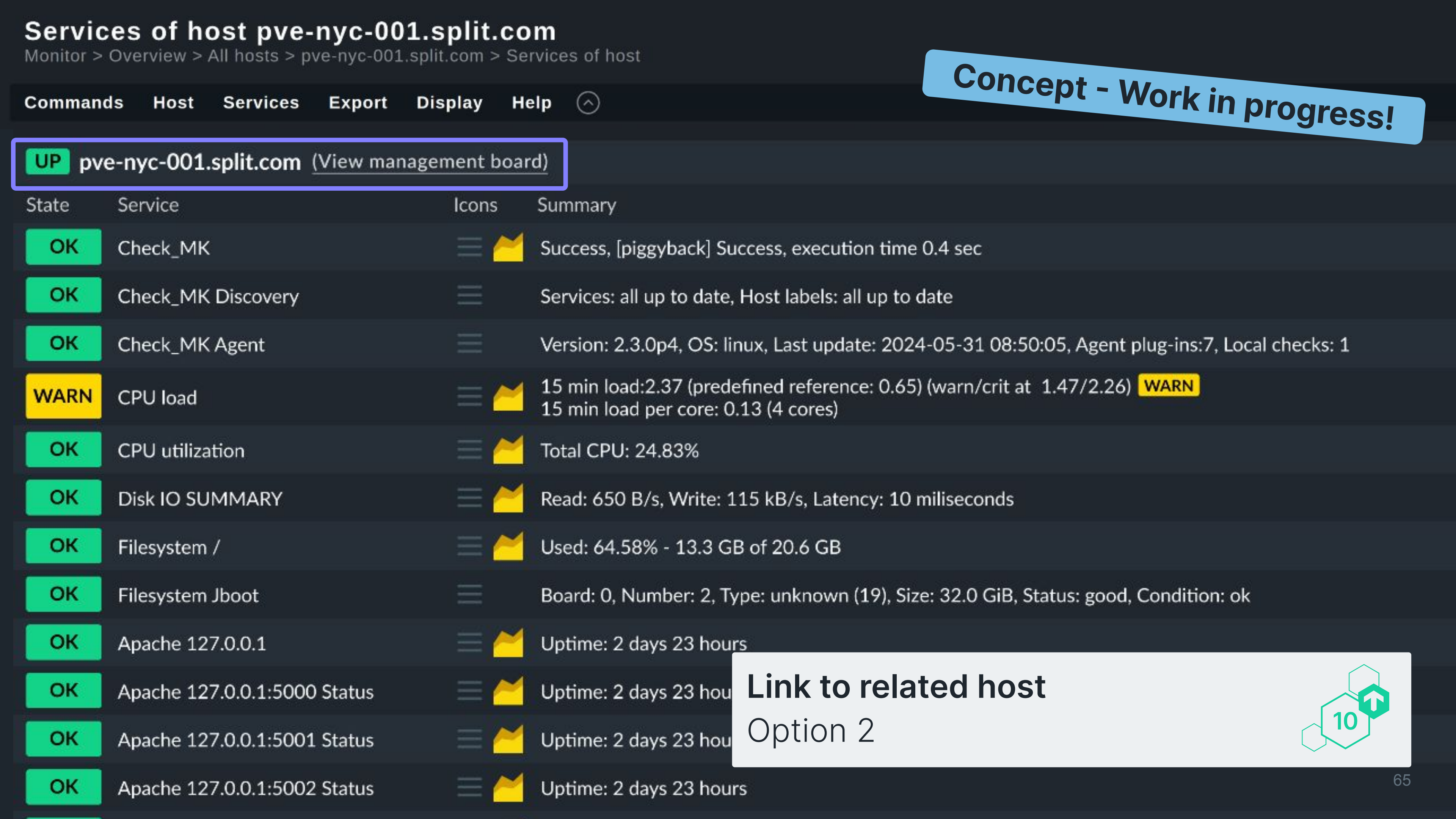
Related host services

pve-nyc-001-ipmi.split.com

State	Service	Icons	Summary
OK	IPMI Sensor 5VCC	≡	Status: OK, 3.36 V
OK	IPMI Sensor 12V	≡	Status: OK, 5.06 V
OK	IPMI Sensor AVCC	≡	Status: OK, 5.06 V

Show related services in same view

Option 1



Services of host pve-nyc-001.split.com

Monitor > Overview > All hosts > pve-nyc-001.split.com > Services of host

Concept - Work in progress!

UP pve-nyc-001.split.com (View management board)

State	Service	Icons	Summary
OK	Check_MK	☰ 📈	Success, [piggyback] Success, execution time 0.4 sec
OK	Check_MK Discovery	☰	Services: all up to date, Host labels: all up to date
OK	Check_MK Agent	☰	Version: 2.3.0p4, OS: linux, Last update: 2024-05-31 08:50:05, Agent plug-ins:7, Local checks: 1
WARN	CPU load	☰ 📈	15 min load:2.37 (predefined reference: 0.65) (warn/crit at 1.47/2.26) WARN 15 min load per core: 0.13 (4 cores)
OK	CPU utilization	☰ 📈	Total CPU: 24.83%
OK	Disk IO SUMMARY	☰ 📈	Read: 650 B/s, Write: 115 kB/s, Latency: 10 miliseconds
OK	Filesystem /	☰ 📈	Used: 64.58% - 13.3 GB of 20.6 GB
OK	Filesystem Jboot	☰	Board: 0, Number: 2, Type: unknown (19), Size: 32.0 GiB, Status: good, Condition: ok
OK	Apache 127.0.0.1	☰ 📈	Uptime: 2 days 23 hours
OK	Apache 127.0.0.1:5000 Status	☰ 📈	Uptime: 2 days 23 hou
OK	Apache 127.0.0.1:5001 Status	☰ 📈	Uptime: 2 days 23 hou
OK	Apache 127.0.0.1:5002 Status	☰ 📈	Uptime: 2 days 23 hours

Link to related host

Option 2

	aws		
Compute / VM	EC2	Compute Engine	Virtual Machine
Storage & Backup	Elastic Block Storage S3 and S3 Glacier	Cloud Storage Filestore	Blob Storage Storage Accounts Recovery Service Vaults
Database	RDS DynamoDB	Cloud SQL	DB for PostgreSQL DB for MySQL
Load Balancer	Application ELB Network ELB	Load Balancer	Load Balancer
Management	Cost and Usage	Cost	Usage Details (cost)
Containers & Kubernetes	Container Service (ECS) Kubernetes Service (EKS)	Cloud Run Kubernetes Engine	Kubernetes Service (AKS)
Functions	Lambda	Cloud Functions	
Networking	Route53 CloudFront WAF		VPN Gateway Traffic Manager
Cache	ElastiCache for Redis	Memorystore	
Other	SNS		AD Connect App Gateway Web Apps

Legend
Existing services

Planned for Checkmk 2.4

	aws		
Compute / VM	EC2	Compute Engine	Virtual Machine
Storage & Backup	Elastic Block Storage S3 and S3 Glacier	Cloud Storage Filestore	Blob Storage Storage Accounts Recovery Service Vaults
Database	RDS DynamoDB **Redshift**	Cloud SQL	DB for PostgreSQL* DB for MySQL* **Cosmos DB**
Load Balancer	Application ELB Network ELB	Load Balancer	Load Balancer
Management	Cost and Usage	Cost	Usage Details (cost)
Containers & Kubernetes	Container Service (ECS) Kubernetes Service (EKS)	Cloud Run Kubernetes Engine	Kubernetes Service (AKS)
Functions	Lambda	Cloud Functions	**Azure Functions**
Networking	Route53 CloudFront WAF		VPN Gateway Traffic Manager
Cache	ElastiCache for Redis	Memorystore	**Redis Cache**
Other	SNS		AD Connect App Gateway Web Apps

Legend

Existing services

Newly planned services

* Adding flexible server support

Plans for extending monitoring coverage

Update on current status

Commissioned projects

Solves immediate need of a customer.

Queue to open again.
Some requests can be implemented for 2.4

Code contributions

Via pull requests

Currently 57 open.
To be tackled in next months.

Ideas portal

Free developments based on community demand

Several accepted.
Only if capacity available

Monitoring plug-in maintenance

Accepted for future release

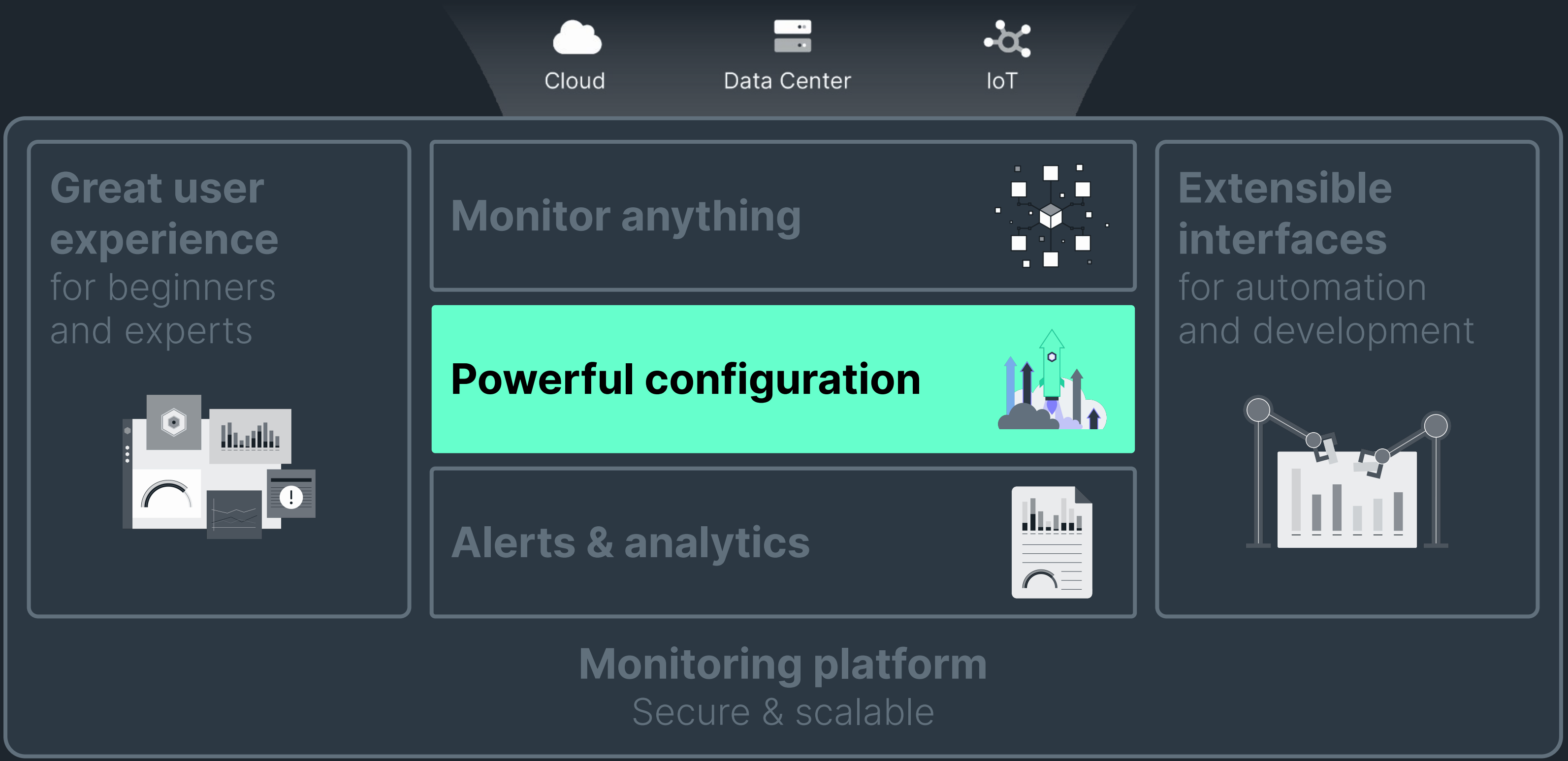
Database monitoring

New MS SQL agent plug-in: test bed for future database monitoring architecture. Depending on feedback during 2.3: Adopt for other SQL DBs

No more VBS

Replace remaining VBS based agent plug-ins (7)

Our current focus for Checkmk



Cloud

Data Center

IoT

Great user experience
for beginners and experts

Monitor anything

Powerful configuration

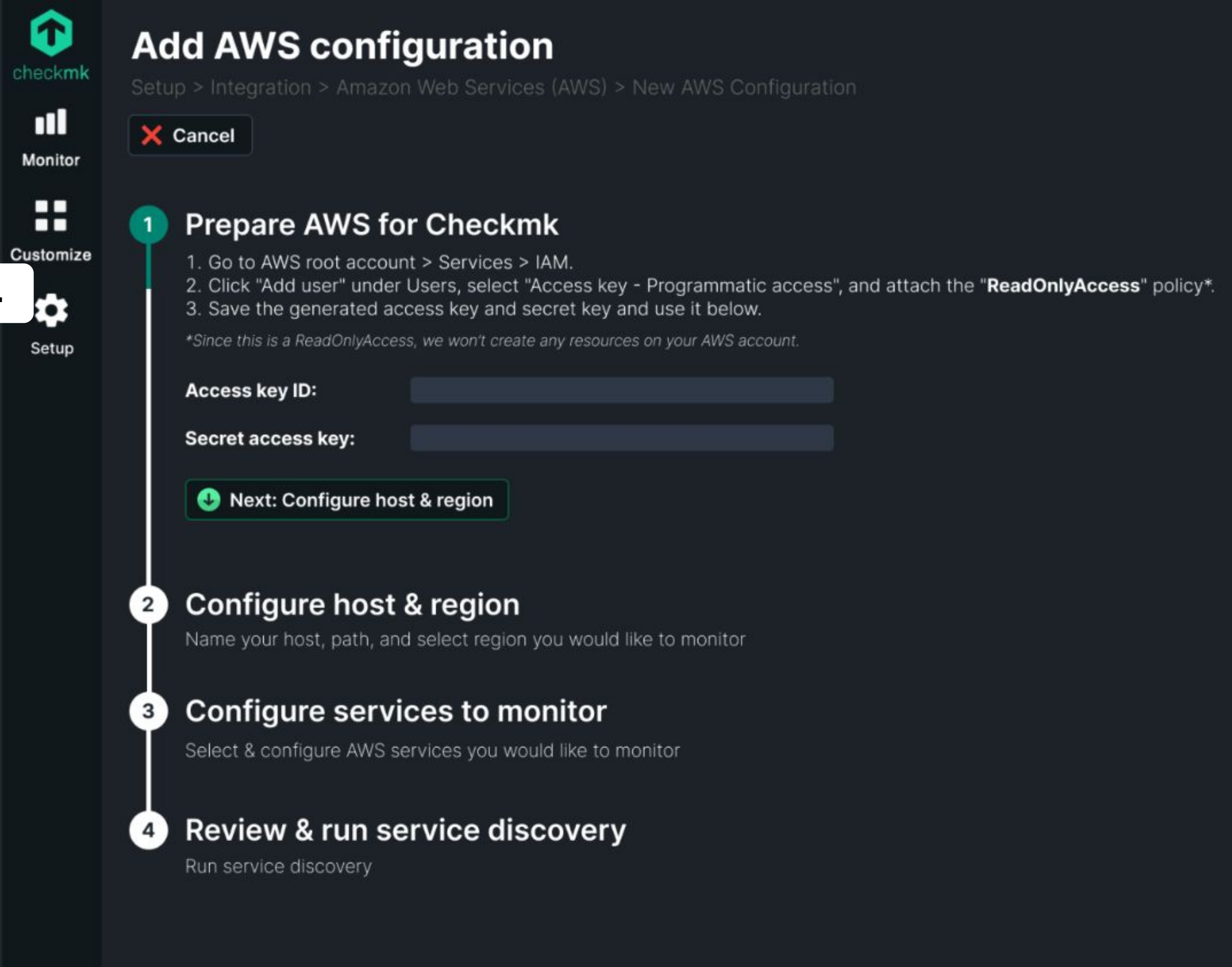
Alerts & analytics

Extensible interfaces
for automation and development

Monitoring platform
Secure & scalable

Making complex things simple.

Planned for Checkmk 2.4



Add AWS configuration

Setup > Integration > Amazon Web Services (AWS) > New AWS Configuration

1 **Prepare AWS for Checkmk**

1. Go to AWS root account > Services > IAM.
2. Click "Add user" under Users, select "Access key - Programmatic access", and attach the "**ReadOnlyAccess**" policy*.
3. Save the generated access key and secret key and use it below.

**Since this is a ReadOnlyAccess, we won't create any resources on your AWS account.*

Access key ID:

Secret access key:

2 **Configure host & region**

Name your host, path, and select region you would like to monitor

3 **Configure services to monitor**

Select & configure AWS services you would like to monitor

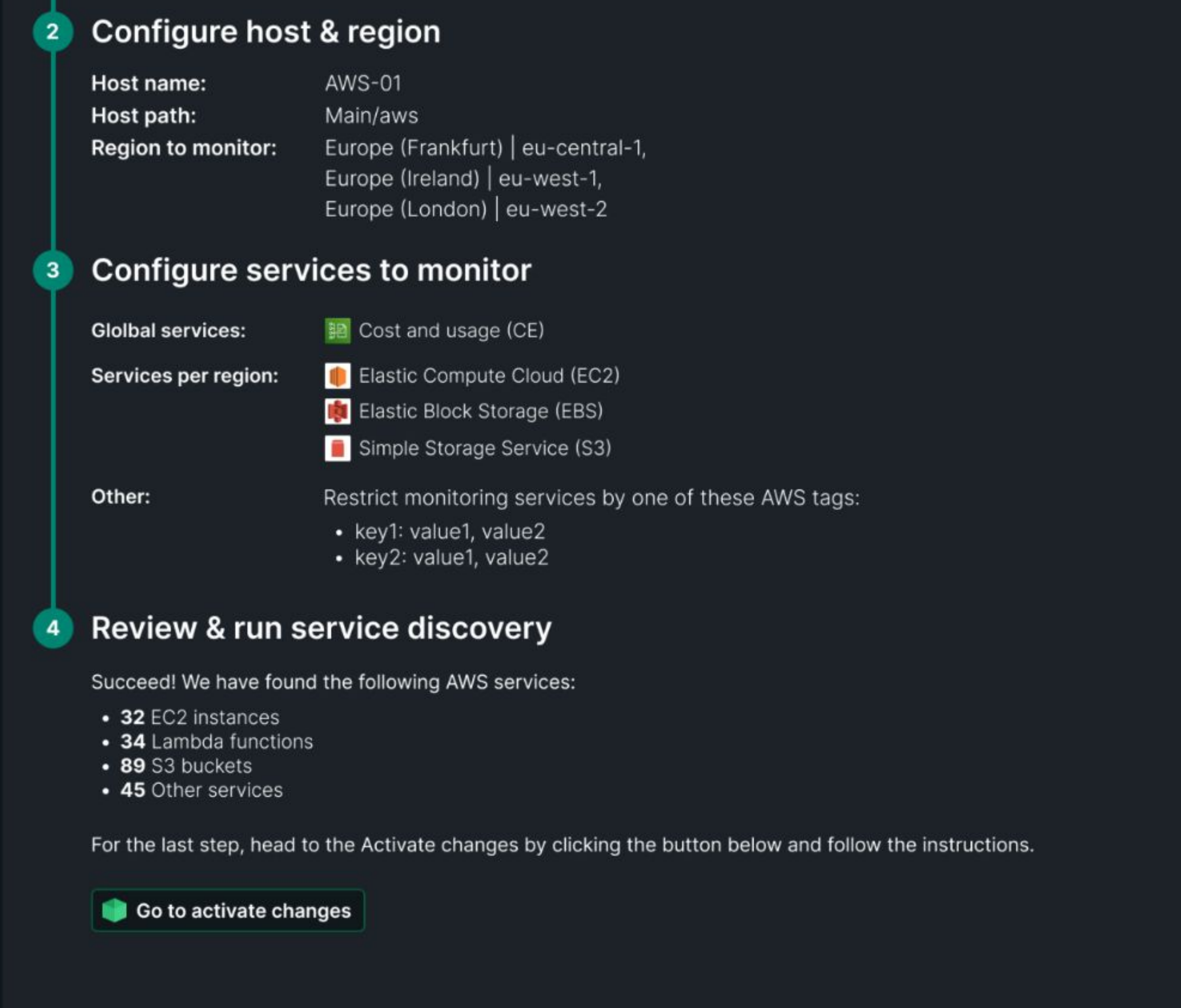
4 **Review & run service discovery**

Run service discovery

Checkmk
Conference

Making complex things simple.

Planned for Checkmk 2.4



2 Configure host & region

Host name: AWS-01
Host path: Main/aws
Region to monitor: Europe (Frankfurt) | eu-central-1,
Europe (Ireland) | eu-west-1,
Europe (London) | eu-west-2

3 Configure services to monitor

Glolbal services: Cost and usage (CE)

Services per region:
Elastic Compute Cloud (EC2)
Elastic Block Storage (EBS)
Simple Storage Service (S3)

Other: Restrict monitoring services by one of these AWS tags:

- key1: value1, value2
- key2: value1, value2

4 Review & run service discovery

Succeed! We have found the following AWS services:

- **32** EC2 instances
- **34** Lambda functions
- **89** S3 buckets
- **45** Other services

For the last step, head to the Activate changes by clicking the button below and follow the instructions.

Cloud setup wizards are just the beginning

Improving multi-step workflows:

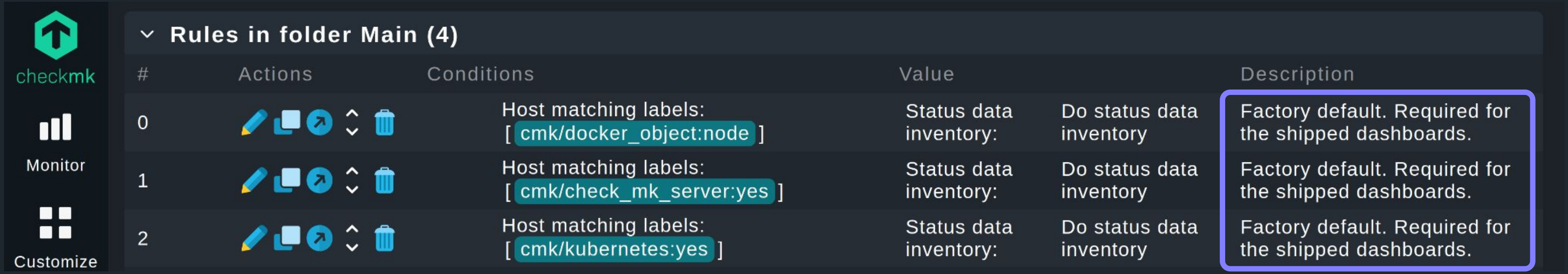
- Get things done quicker without the need for documentation
- Preserve the concept of rules, but remove the pain to find the right rules
- Reduce errors in configuration (e.g. didn’t configure dynamic configuration)

Roll out eventually to further workflows

- Anything creating piggyback data: Kubernetes, VMware vSphere, Proxmox
- Agent bakery

‘Protect’ important rules

Accepted for future release



Rules in folder Main (4)

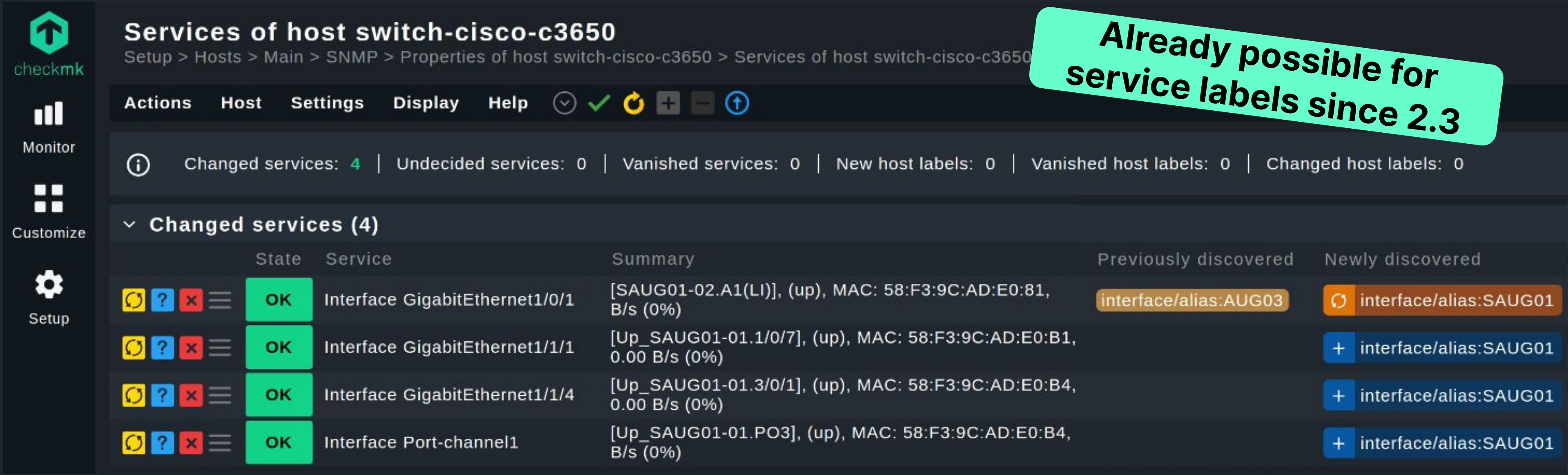
#	Actions	Conditions	Value		Description
0		Host matching labels: [cmk/docker_object:node]	Status data inventory:	Do status data inventory	Factory default. Required for the shipped dashboards.
1		Host matching labels: [cmk/check_mk_server:yes]	Status data inventory:	Do status data inventory	Factory default. Required for the shipped dashboards.
2		Host matching labels: [cmk/kubernetes:yes]	Status data inventory:	Do status data inventory	Factory default. Required for the shipped dashboards.

Important (built-in) rules:

- A few rules are built-in and essential for proper functionality of Checkmk
- Checkmk & partners often create best-practice rules during consulting
- Can be just deleted without any warning

Enable protection via permissions and/or warning dialogs

Refreshing discovered parameters on-the-fly



Services of host switch-cisco-c3650

Setup > Hosts > Main > SNMP > Properties of host switch-cisco-c3650 > Services of host switch-cisco-c3650

Changed services: 4 | Undecided services: 0 | Vanished services: 0 | New host labels: 0 | Vanished host labels: 0 | Changed host labels: 0

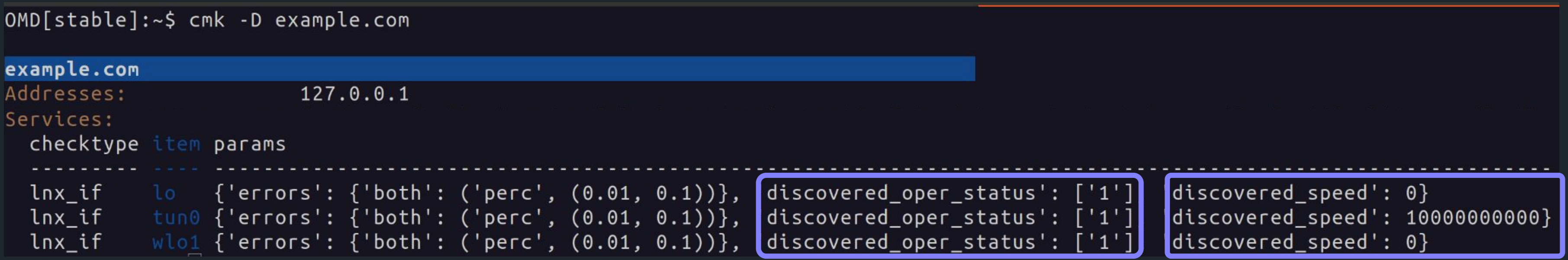
Changed services (4)

	State	Service	Summary	Previously discovered	Newly discovered
⟳ ? ✕ ☰	OK	Interface GigabitEthernet1/0/1	[SAUG01-02.A1(LI)], (up), MAC: 58:F3:9C:AD:E0:81, B/s (0%)	interface/alias:AUG03	⟳ interface/alias:SAUG01
⟳ ? ✕ ☰	OK	Interface GigabitEthernet1/1/1	[Up_SAUG01-01.1/0/7], (up), MAC: 58:F3:9C:AD:E0:B1, 0.00 B/s (0%)		+ interface/alias:SAUG01
⟳ ? ✕ ☰	OK	Interface GigabitEthernet1/1/4	[Up_SAUG01-01.3/0/1], (up), MAC: 58:F3:9C:AD:E0:B4, 0.00 B/s (0%)		+ interface/alias:SAUG01
⟳ ? ✕ ☰	OK	Interface Port-channel1	[Up_SAUG01-01.PO3], (up), MAC: 58:F3:9C:AD:E0:B4, B/s (0%)		+ interface/alias:SAUG01

Already possible for service labels since 2.3

Refreshing discovered parameters on-the-fly

Planned for Checkmk 2.4



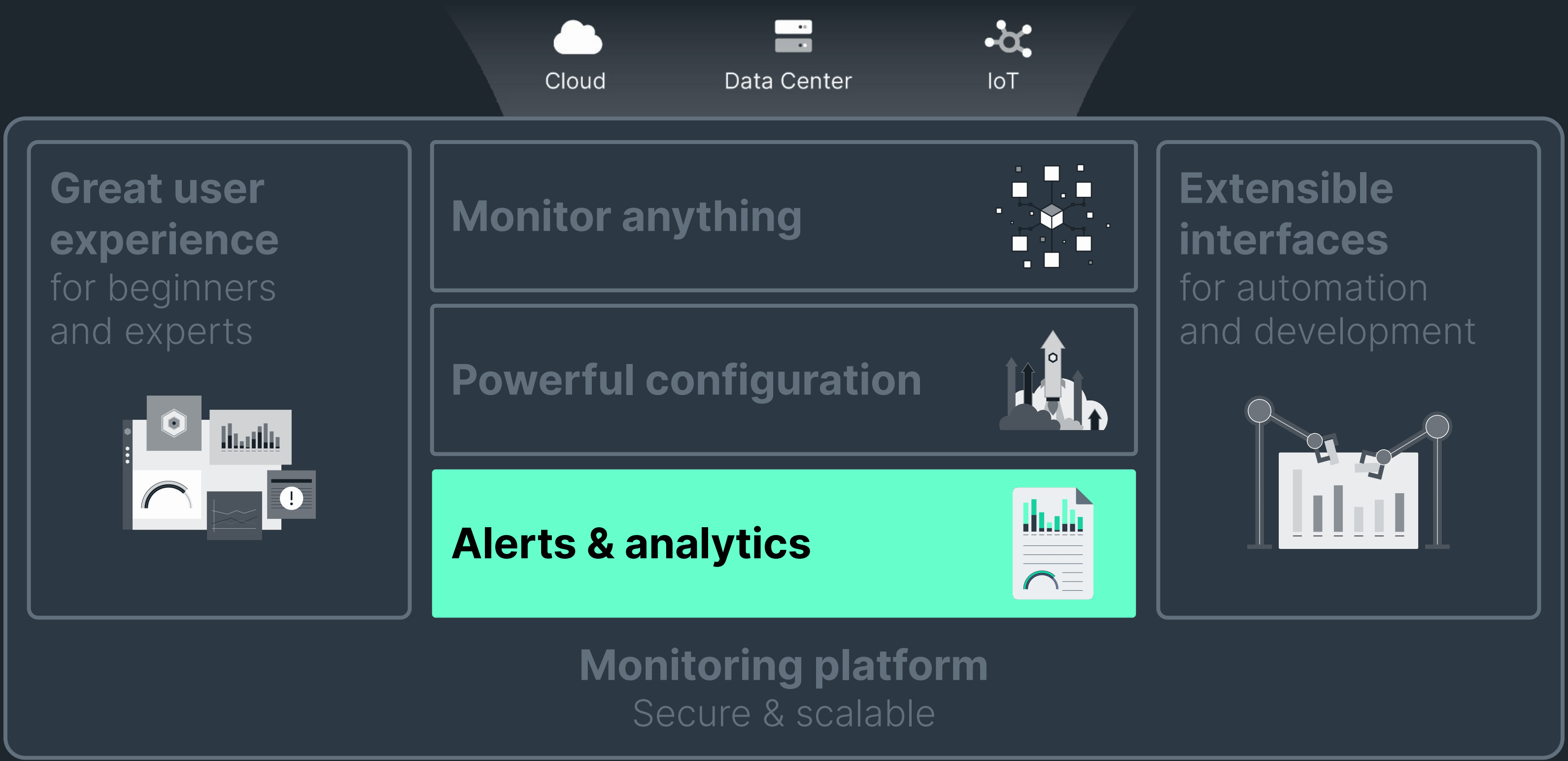
```
OMD[stable]:~$ cmk -D example.com

example.com
Addresses:                  127.0.0.1
Services:
  checktype item params
  --------- ---- ------------------------------------------------------------------------------------------------
  lnx_if    lo   {'errors': {'both': ('perc', (0.01, 0.1))}, discovered_oper_status': ['1']  discovered_speed': 0}
  lnx_if    tun0 {'errors': {'both': ('perc', (0.01, 0.1))}, discovered_oper_status': ['1']  discovered_speed': 10000000000}
  lnx_if    wlo1 {'errors': {'both': ('perc', (0.01, 0.1))}, discovered_oper_status': ['1']  discovered_speed': 0}
```

Discovered parameters:

- Used for defining a desired state, e.g. interface speed & status
- Defining a new desired state requires a ‘rediscovery’: currently only via removing service and re-adding it (tabula rasa) possible

Allow refreshing on the fly similar to service labels

Our current focus for Checkmk



Cloud

Data Center

IoT

Great user experience
for beginners
and experts

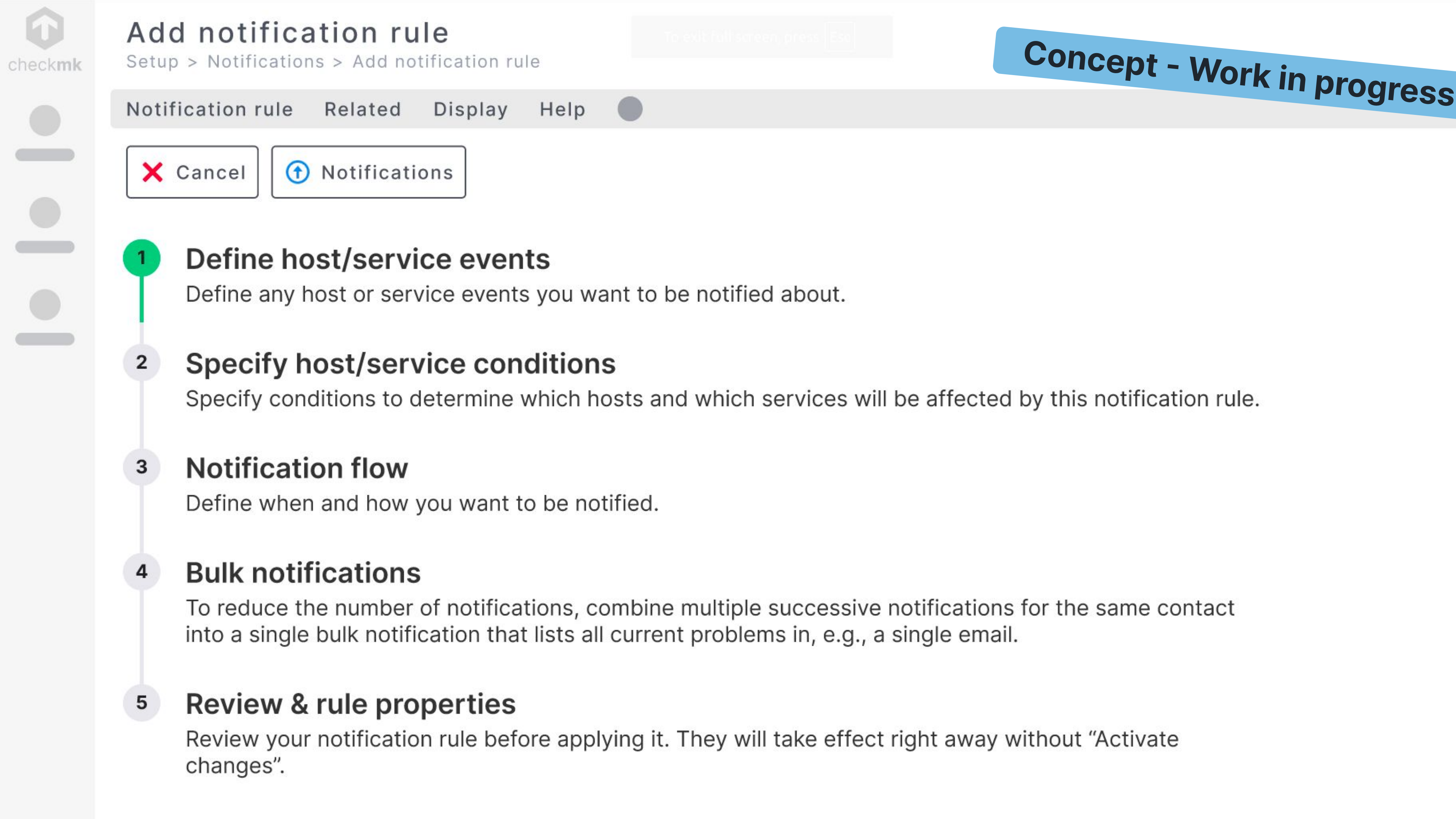
Monitor anything

Powerful configuration

Alerts & analytics

Extensible interfaces
for automation
and development

Monitoring platform
Secure & scalable



Concept - Work in progress

Add notification rule

Setup > Notifications > Add notification rule

To exit full screen, press Esc

1 **Define host/service events**

Define any host or service events you want to be notified about.

2 **Specify host/service conditions**

Specify conditions to determine which hosts and which services will be affected by this notification rule.

3 **Notification flow**

Define when and how you want to be notified.

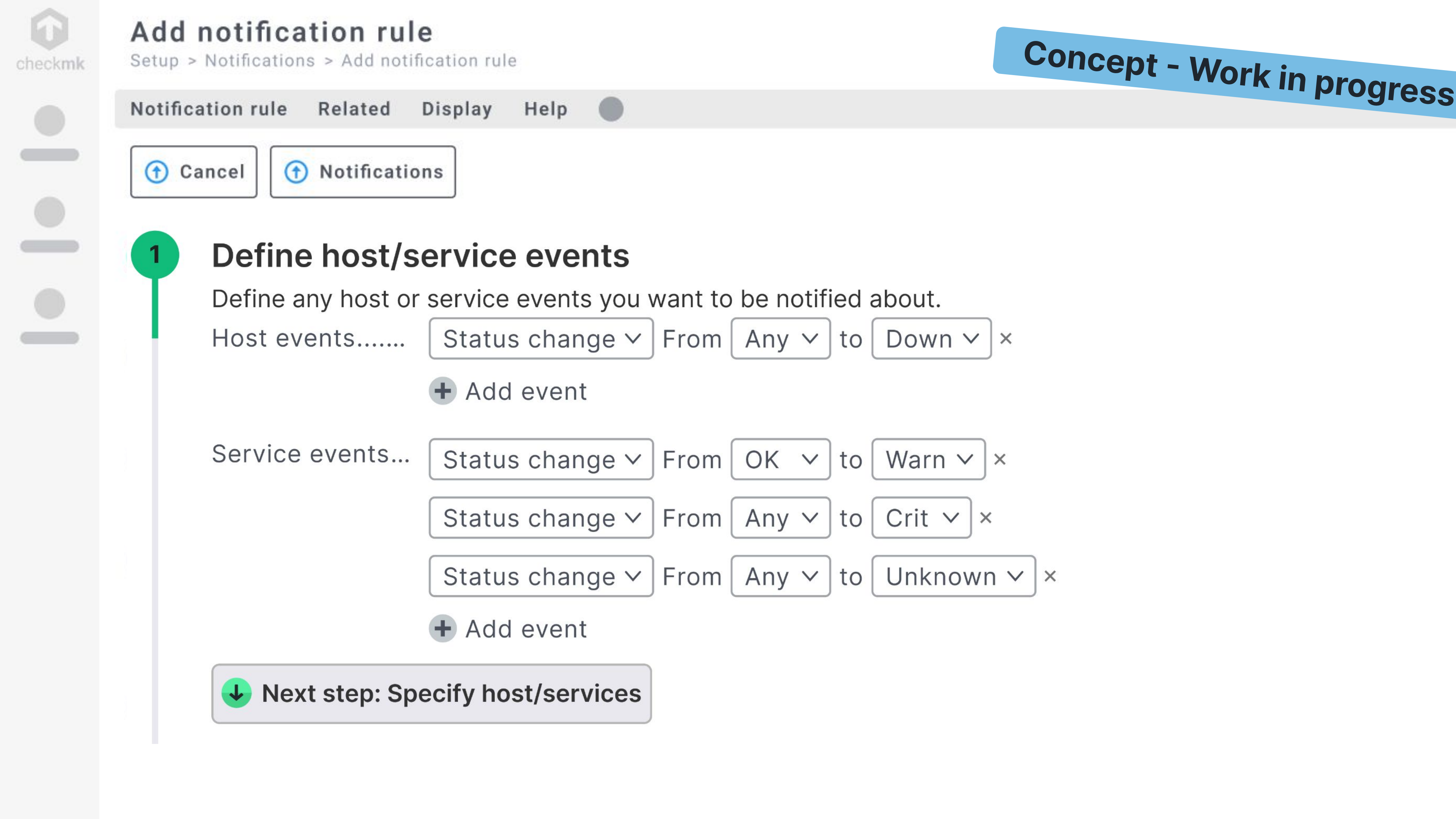
4 **Bulk notifications**

To reduce the number of notifications, combine multiple successive notifications for the same contact into a single bulk notification that lists all current problems in, e.g., a single email.

5 **Review & rule properties**

Review your notification rule before applying it. They will take effect right away without “Activate changes”.

Concept - Work in progress



Add notification rule

Setup > Notifications > Add notification rule

1 Define host/service events

Define any host or service events you want to be notified about.

Host events....... Status change From Any to Down ×

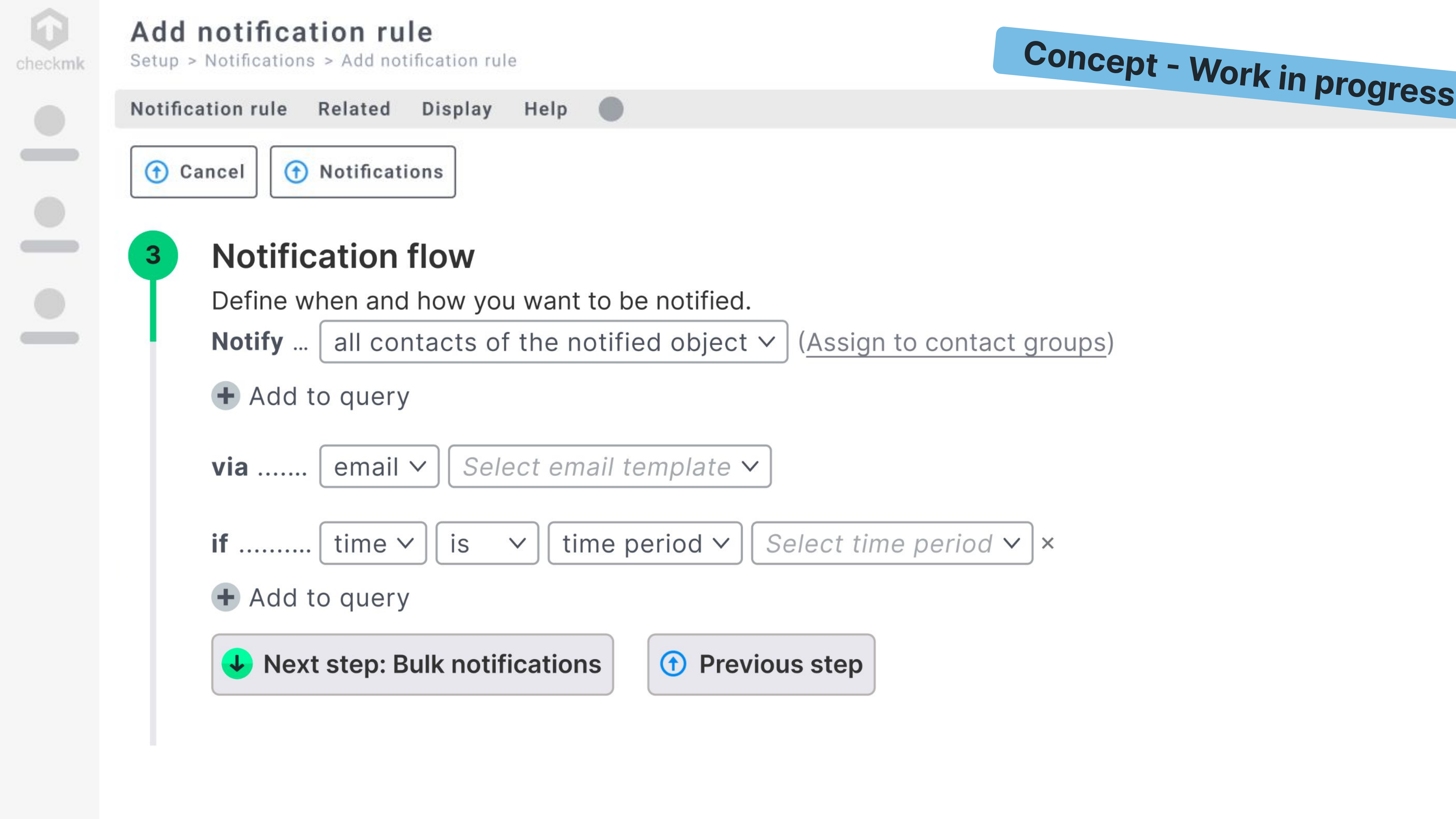
Service events... Status change From OK to Warn ×

Status change From Any to Crit ×

Status change From Any to Unknown ×

Next step: Specify host/services

Concept - Work in progress



Add notification rule

Setup > Notifications > Add notification rule

3 Notification flow

Define when and how you want to be notified.

Notify ... all contacts of the notified object (Assign to contact groups)

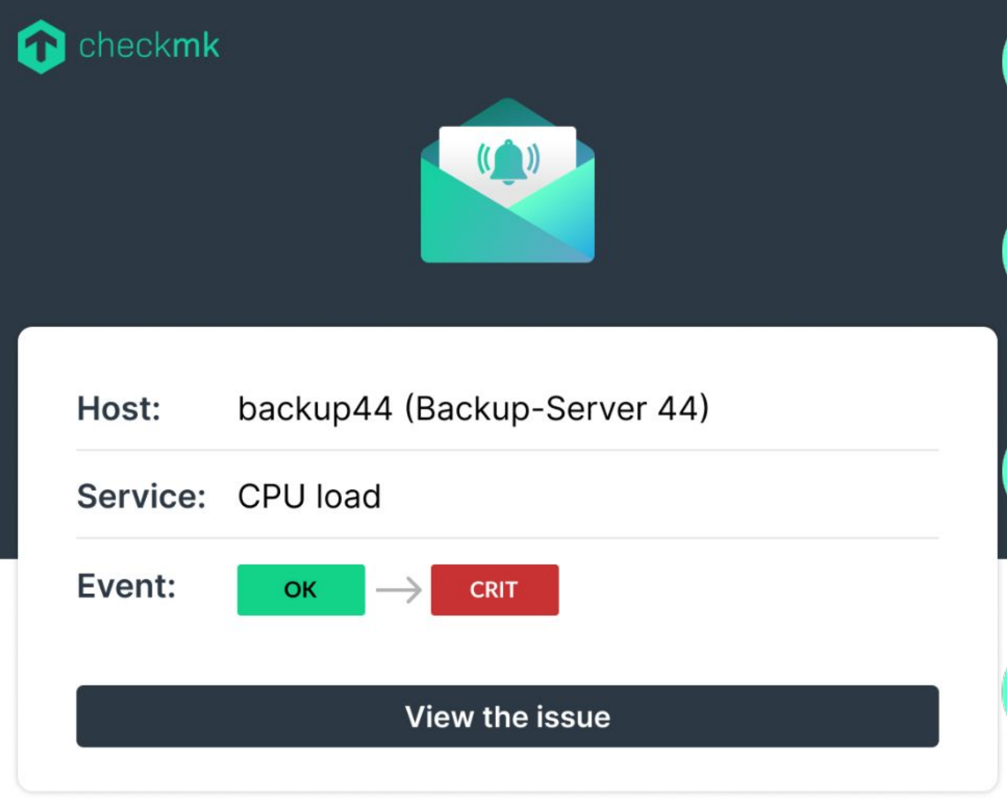
+ Add to query

via .... email *Select email template*

if .... time is time period *Select time period* ×

+ Add to query

Redesigning email notifications



Host: backup44 (Backup-Server 44)

Service: CPU load

Event: OK → CRIT

- ✓ **Focused:** Most important info on top
- ✓ **Modern look and feel:** “New” graphs
- ✓ **Traceability:** Rule that triggered the notification
- ✓ **Informational:** Service labels + Contact groups

Complete rework of notifications

Planned for Checkmk 2.4

A new way to configure notifications

Everything in one place. Smooth. Clever. With good workflows.

Reusable templates and connections

Built-in best practice templates.

Create connections only once and then re-use

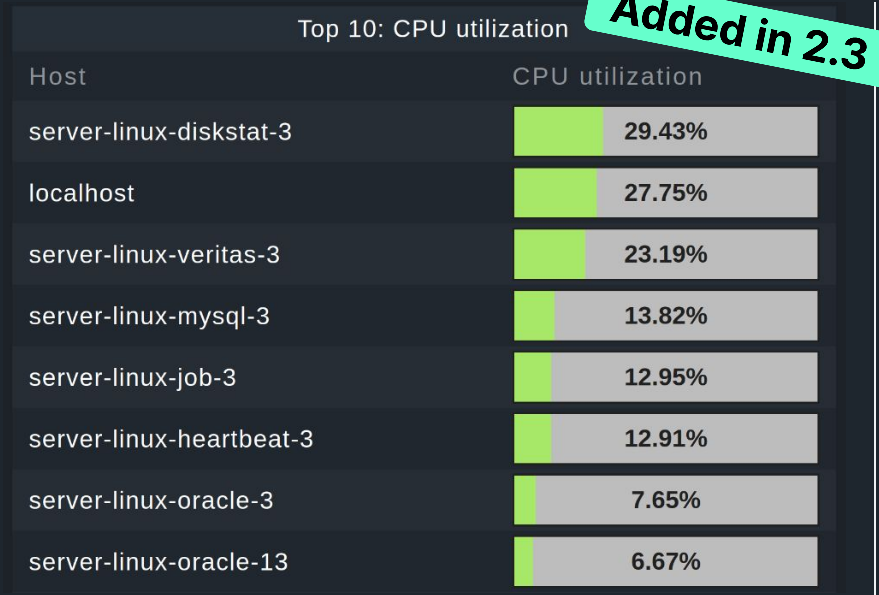
Redesigned notifications

Receive helpful notifications in an easy-to-read modern look

Top lists for graphs

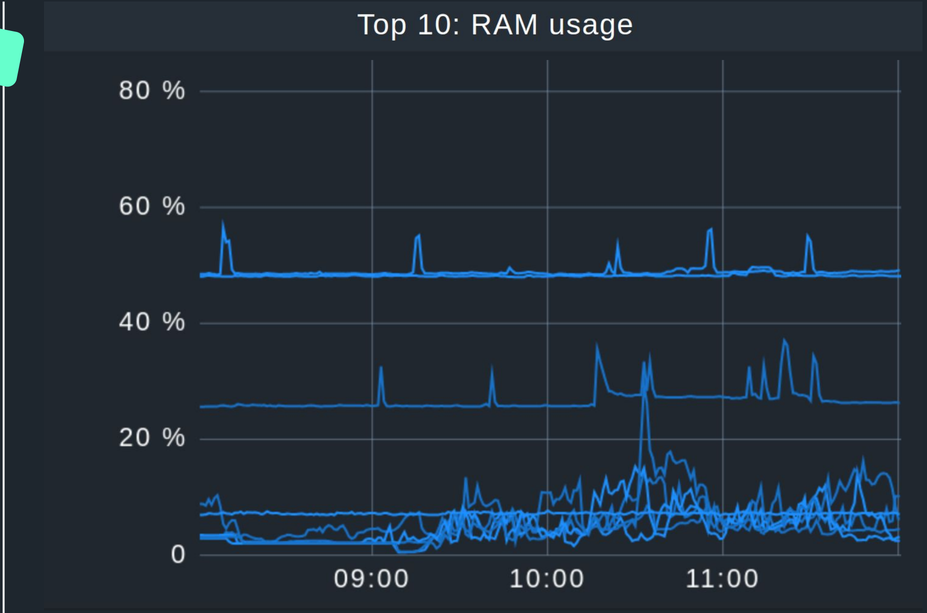
Accepted for future release

Added in 2.3



Top 10: CPU utilization

Host	CPU utilization
server-linux-diskstat-3	29.43%
localhost	27.75%
server-linux-veritas-3	23.19%
server-linux-mysql-3	13.82%
server-linux-job-3	12.95%
server-linux-heartbeat-3	12.91%
server-linux-oracle-3	7.65%
server-linux-oracle-13	6.67%

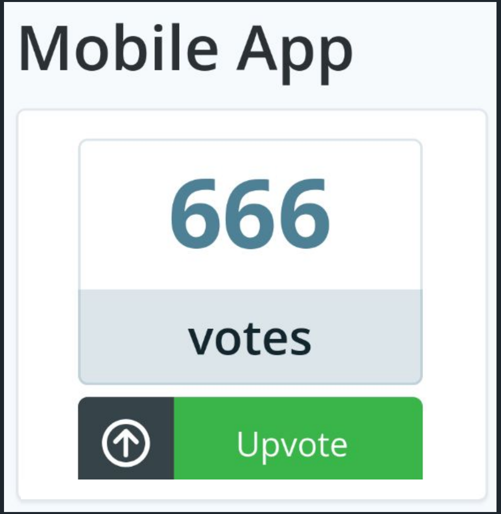


Top 10: RAM usage

Limit graph dashlets to top entries

A mobile app for Checkmk

Accepted for future release



Mobile App

666

votes

Checkmk 2.5

Views: Hosts & services

Actions:
Acknowledge problems, schedule downtimes

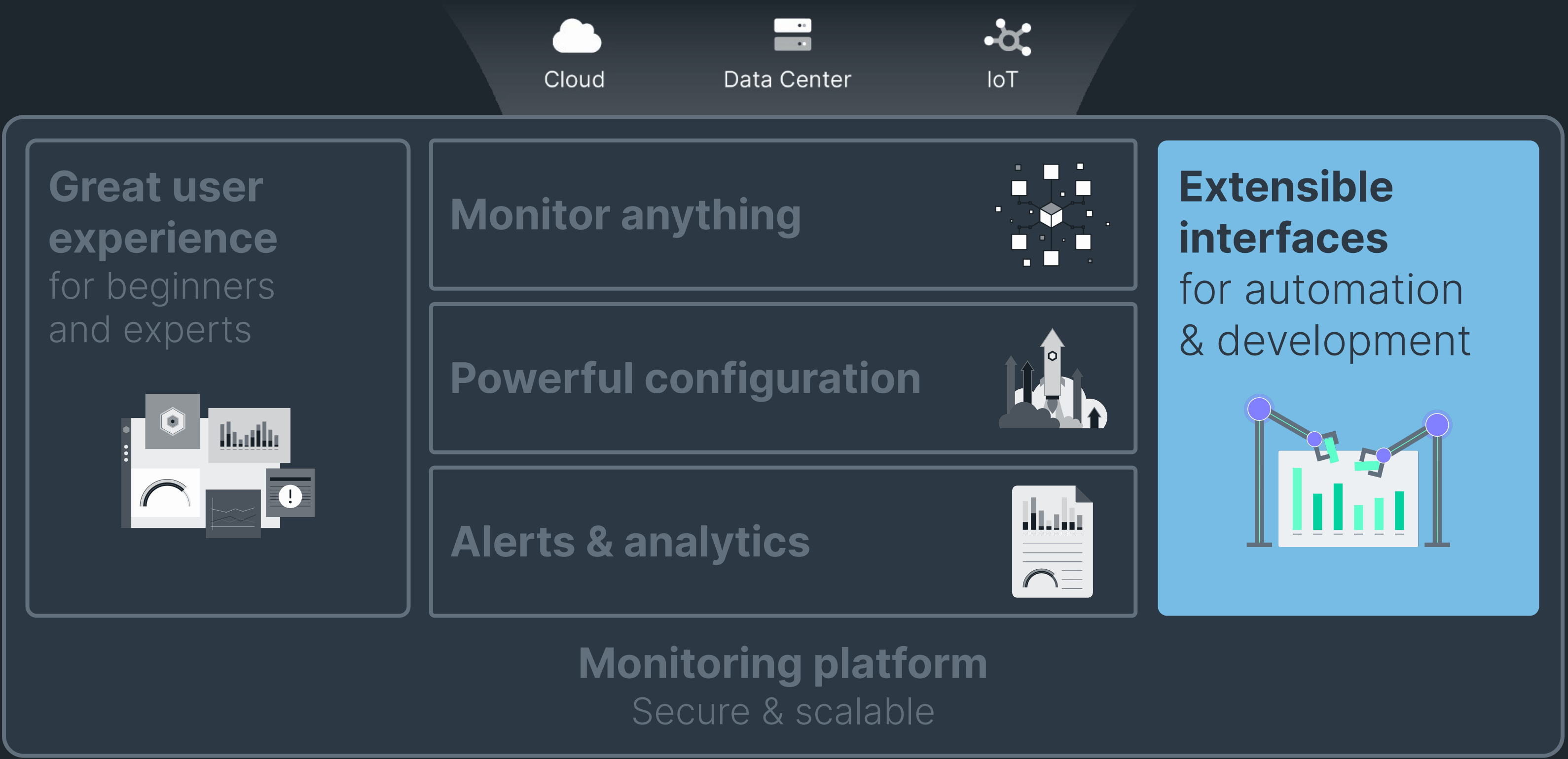
Later

Notifications

Graphs

Note: Initially likely only available for public Checkmk instances. Will replace the mobile view. Out of scope: configuration

Our current focus for Checkmk



Cloud

Data Center

IoT

Great user experience
for beginners
and experts

Monitor anything

Powerful configuration

Alerts & analytics

Extensible interfaces
for automation
& development

Monitoring platform
Secure & scalable

REST-API extended and improved a lot already

Monitoring

Acknowledgements	2.0
Downtimes	2.0
Host status	2.0
Service status	2.0
Event Console	2.2
Metrics	2.2
Comments	2.2
SLA	2.2

Setup

Activate changes	2.0
Service groups	2.0
Service discovery	2.0
Hosts	2.0
Host groups	2.0
Host tag groups	2.0
Contact groups	2.0
Business Intel.	2.0
Folders	2.0
Passwords	2.0
Time periods	2.0

Setup

Agents	2.0
Users	2.0
Rules	2.1
Rulesets	2.1
Auxiliary tags	2.2
Site Mgmt	2.2
User roles	2.2
Audit logs	2.3
Dynamic config	2.3
Notification rules	2.3
LDAP connection	2.4

Checkmk
Conference

Introduce API versioning

Planned for Checkmk 2.4

Time for a versioned API

- For users: Increased reliability and transparency
- For Checkmk: Freedom to implement substantial changes
- Currently discussing:
  - Versioning of entire API
  - Versioning of individual endpoints

Improve REST API performance

Planned for Checkmk 2.4

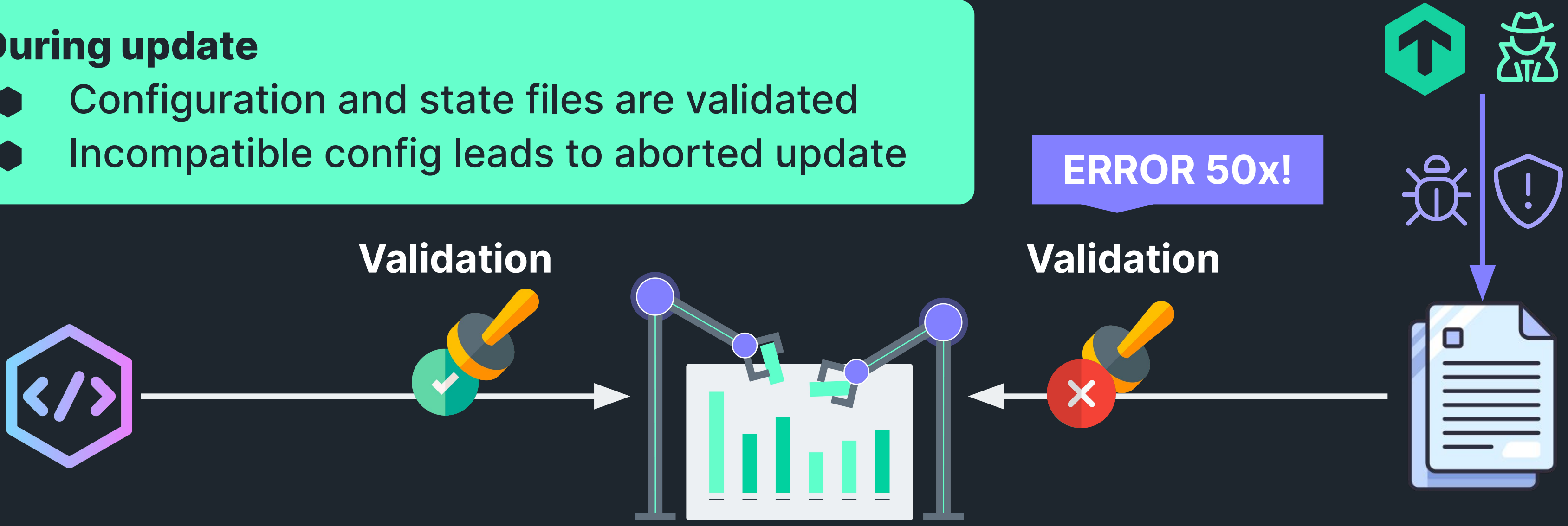
- Many customers configure Checkmk entirely via REST API
- Many singular optimizations done over last years
- Analyzing potential of more substantial changes, e.g. in data serialization / validation
- Focus of improvements are very large environments

Improving the runtime stability of Checkmk

Planned for Checkmk 2.4

During update

- Configuration and state files are validated
- Incompatible config leads to aborted update



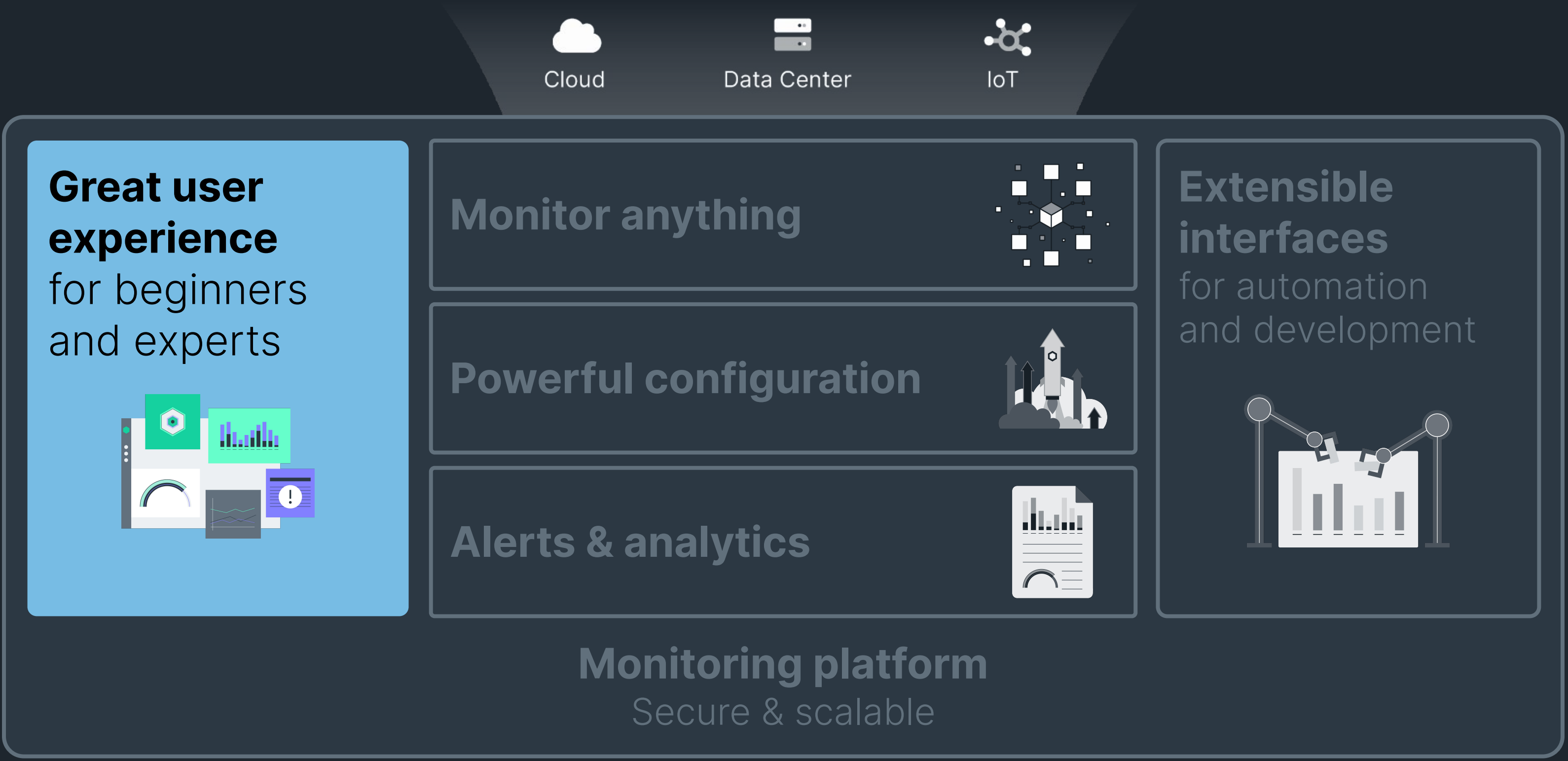
Automation scripts

REST-API

Checkmk data

Checkmk
Conference

Our current focus for Checkmk



Cloud

Data Center

IoT

Great user experience for beginners and experts

Monitor anything

Powerful configuration

Alerts & analytics

Extensible interfaces for automation and development

Monitoring platform

Secure & scalable

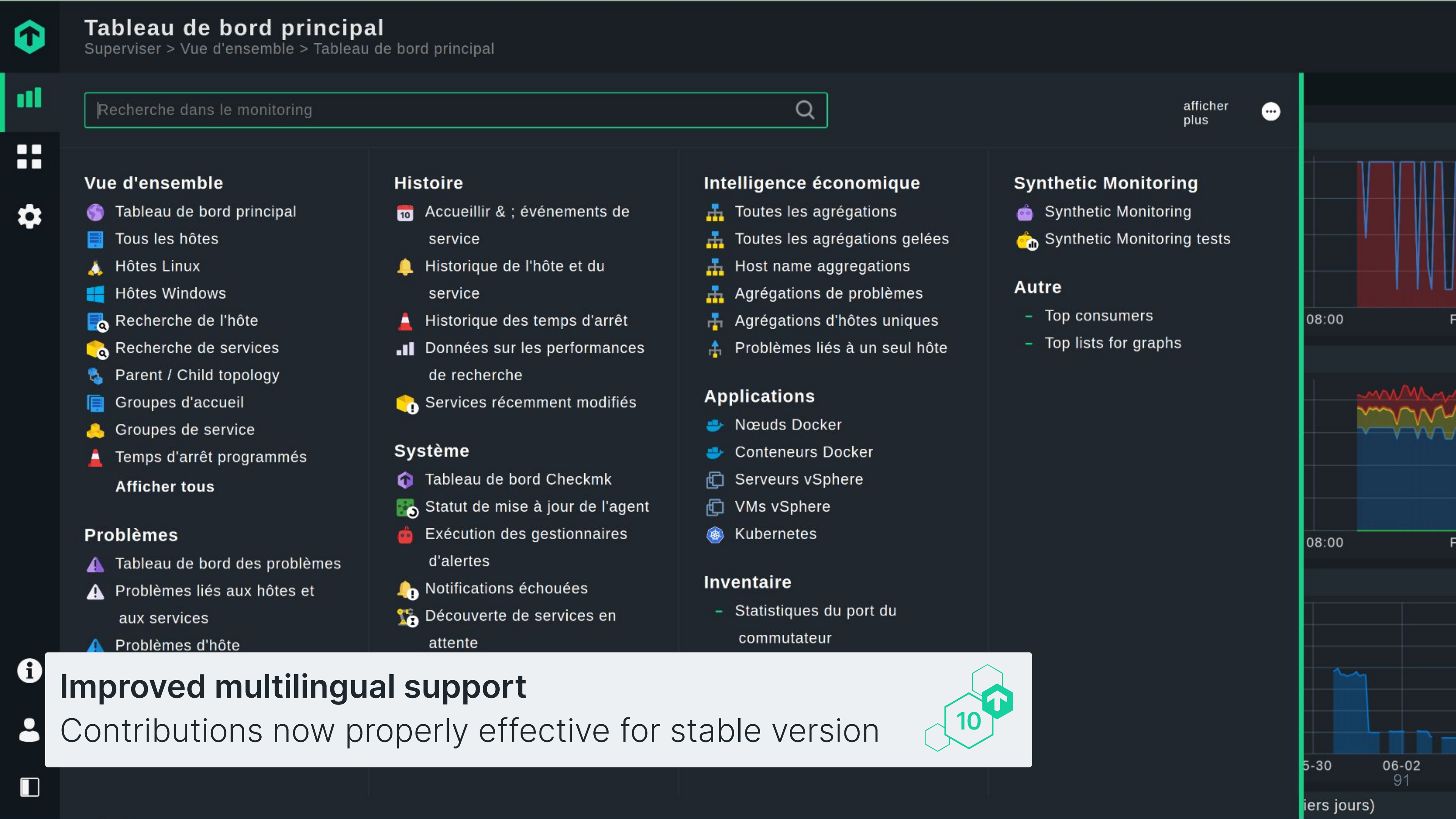


Tableau de bord principal

Superviser > Vue d'ensemble > Tableau de bord principal

Recherche dans le monitoring

afficher plus

Vue d'ensemble
- Tableau de bord principal
- Tous les hôtes
- Hôtes Linux
- Hôtes Windows
- Recherche de l'hôte
- Recherche de services
- Parent / Child topology
- Groupes d'accueil
- Groupes de service
- Temps d'arrêt programmés
- **Afficher tous**

Problèmes
- Tableau de bord des problèmes
- Problèmes liés aux hôtes et aux services
- Problèmes d'hôte

Histoire
- Accueillir & ; événements de service
- Historique de l'hôte et du service
- Historique des temps d'arrêt
- Données sur les performances de recherche
- Services récemment modifiés

Système
- Tableau de bord Checkmk
- Statut de mise à jour de l'agent
- Exécution des gestionnaires d'alertes
- Notifications échouées
- Découverte de services en attente

Intelligence économique
- Toutes les agrégations
- Toutes les agrégations gelées
- Host name aggregations
- Agrégations de problèmes
- Agrégations d'hôtes uniques
- Problèmes liés à un seul hôte

Applications
- Nœuds Docker
- Conteneurs Docker
- Serveurs vSphere
- VMs vSphere
- Kubernetes

Inventaire
- Statistiques du port du commutateur

Synthetic Monitoring
- Synthetic Monitoring
- Synthetic Monitoring tests

Autre
- Top consumers
- Top lists for graphs

Improved multilingual support

Contributions now properly effective for stable version

AI-assisted high-quality translations

Current focus

Product

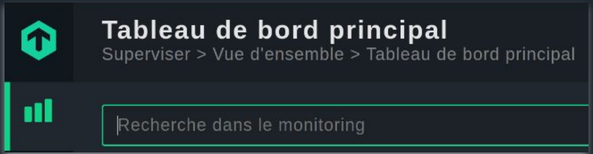
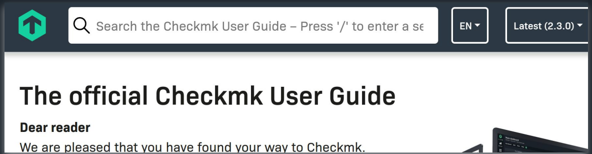


Tableau de bord principal

Superviser > Vue d'ensemble > Tableau de bord principal

Recherche dans le monitoring

User guide

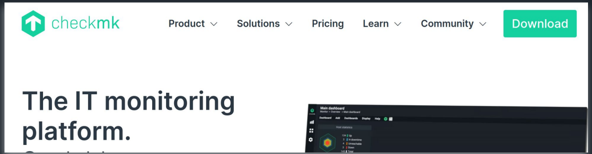


The official Checkmk User Guide

Dear reader

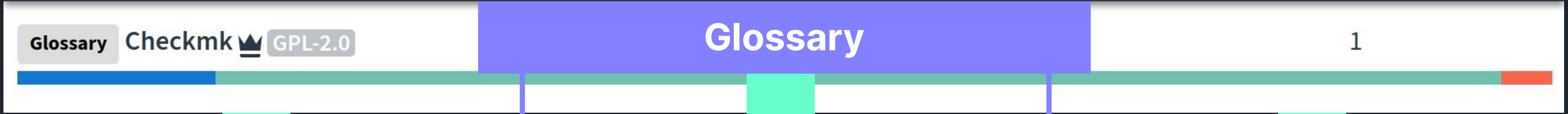
We are pleased that you have found your way to Checkmk.

Website



The IT monitoring platform.

AI-assisted translation

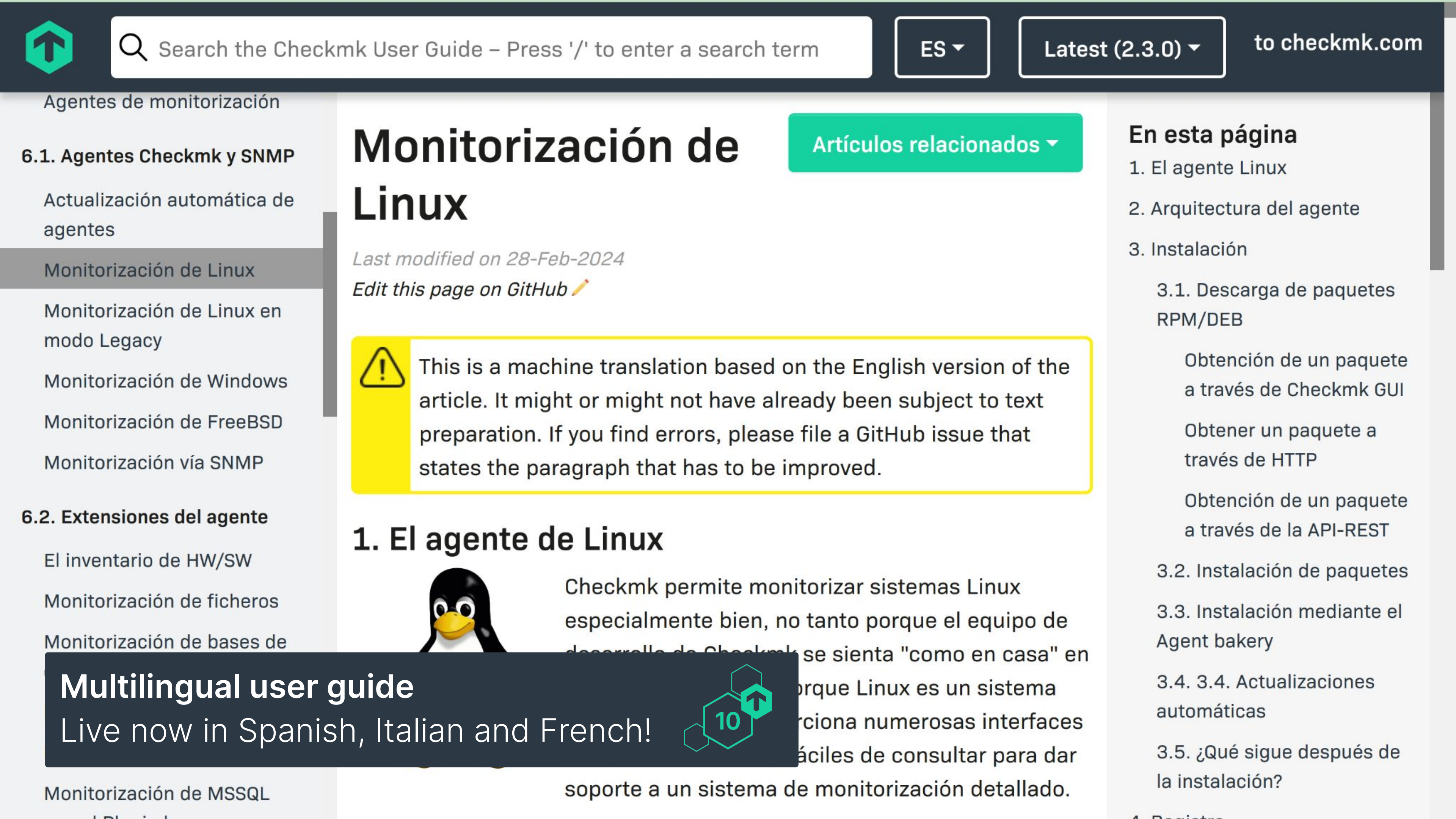


Glossary

Product

User guide

Website



Search the Checkmk User Guide – Press '/' to enter a search term

ES

Latest (2.3.0)

to checkmk.com

Agentes de monitorización

6.1. Agentes Checkmk y SNMP

Actualización automática de agentes

Monitorización de Linux

Monitorización de Linux en modo Legacy

Monitorización de Windows

Monitorización de FreeBSD

Monitorización vía SNMP

6.2. Extensiones del agente

El inventario de HW/SW

Monitorización de ficheros

Monitorización de bases de

Monitorización de MSSQL

Monitorización de Linux

Artículos relacionados

Last modified on 28-Feb-2024

Edit this page on GitHub

This is a machine translation based on the English version of the article. It might or might not have already been subject to text preparation. If you find errors, please file a GitHub issue that states the paragraph that has to be improved.

1. El agente de Linux

Checkmk permite monitorizar sistemas Linux especialmente bien, no tanto porque el equipo de desarrollo de Checkmk se sienta "como en casa" en

rque Linux es un sistema

rciona numerosas interfaces

áciles de consultar para dar soporte a un sistema de monitorización detallado.

En esta página

1. El agente Linux

2. Arquitectura del agente

3. Instalación

3.1. Descarga de paquetes RPM/DEB

Obtención de un paquete a través de Checkmk GUI

Obtener un paquete a través de HTTP

Obtención de un paquete a través de la API-REST

3.2. Instalación de paquetes

3.3. Instalación mediante el Agent bakery

3.4. 3.4. Actualizaciones automáticas

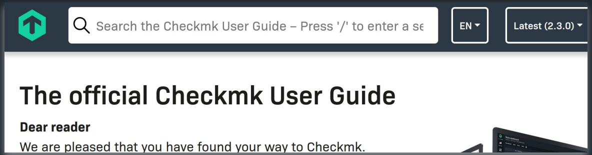
3.5. ¿Qué sigue después de la instalación?

Multilingual user guide

Live now in Spanish, Italian and French!

Help us by fixing translations!

User guide

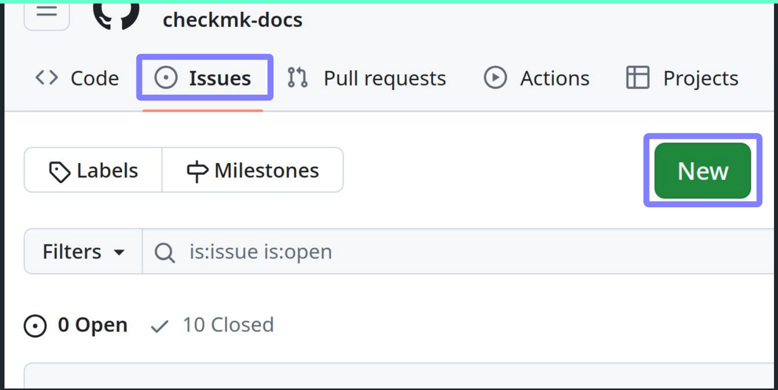


The official Checkmk User Guide

Dear reader

We are pleased that you have found your way to Checkmk.

github.com/checkmk/checkmk-docs



is:issue is:open

0 Open ✓ 10 Closed

Product

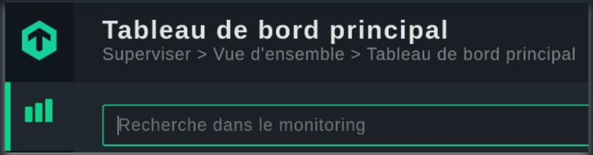
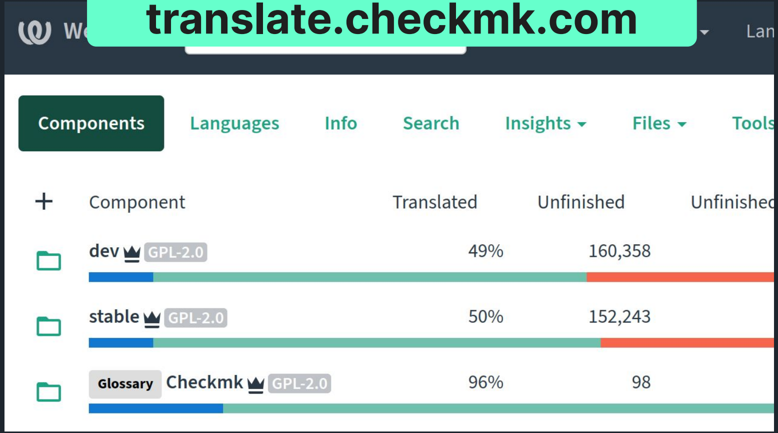


Tableau de bord principal

Superviser > Vue d'ensemble > Tableau de bord principal

Recherche dans le monitoring

translate.checkmk.com



Component	Translated	Unfinished	Unfinished
dev GPL-2.0	49%	160,358	
stable GPL-2.0	50%	152,243	
Glossary Checkmk GPL-2.0	96%	98	

The Checkmk UX strategy

Consistency

Efficiency

Error prevention & recovery

Accessibility

Many smaller projects …

+

… larger projects

Notifications

Setup made easy

No more ‘Show more’

The next big topic: Navigation 2.5

Accepted for future release

Checkmk 2.0
...
Checkmk 2.5

Navigation 2.0

- Monitor
- Customize
- Setup

A lot of positive feedback

+ Search bar for setup
+ Clear split between Monitor & Setup
+ No more scrolling

Improvement areas

- No. of Setup / Monitor entries and structure
- ‘Show more’

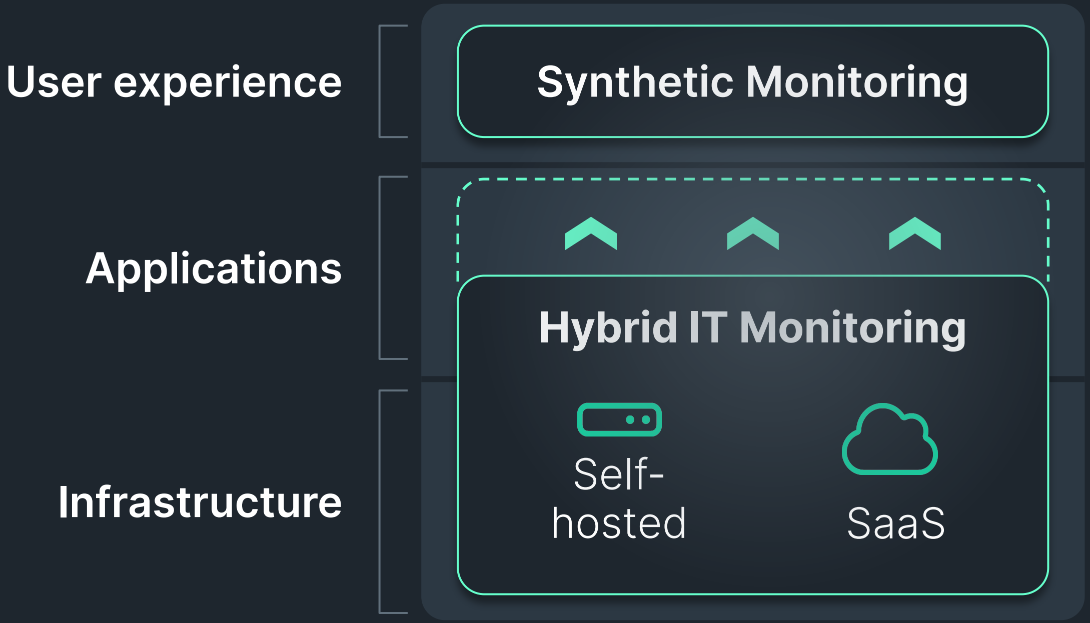
Navigation 2.5

Goal: Balance between rate of change & improvements

Paradigm

- Intuitive
- As few clicks as possible
- Easy for beginners, powerful for Experts

Improving the core and going beyond!



10