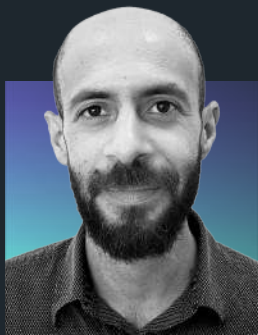


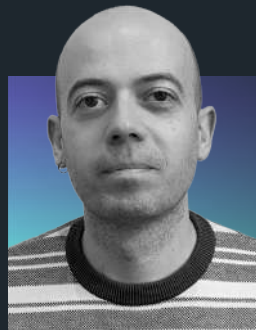


Monitoring at Scale

Azure, Containers, and Virtualization in Checkmk 2.5



Mohamed Saleh
Product Manager
Checkmk



David Gervasoni
Team Lead
Checkmk

Full-stack observability

Infrastructure

Application

Synthetic



Best in class user experience



Faster troubleshooting with embedded AI



Advanced data analytics & visualization



A faster, more scalable platform

Full-stack observability

Infrastructure

Application

Synthetic



Best in class user experience



Faster troubleshooting with embedded AI



Advanced data analytics & visualization



A faster, more scalable platform

Infrastructure monitoring in Checkmk 2.5



Virtualization & Containers

Excellent monitoring out of the box

- Proxmox
- Hyper-V
- Podman

Infrastructure monitoring in Checkmk 2.5



Virtualization & Containers

Excellent monitoring out of the box

- Proxmox
- Hyper-V
- Podman



Cloud

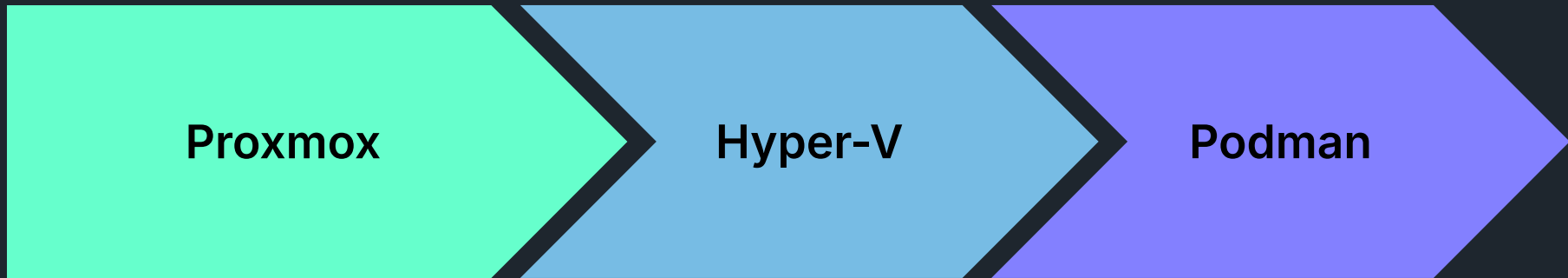
Blueprint for future-proof scale

- Azure



Excellent out of the box virtualization & container monitoring

With...



What excellent monitoring looks like



Simple to set up, built to scale

What excellent monitoring looks like



Simple to set up, built to scale



Coverage at every layer

What excellent monitoring looks like



Simple to set up, built to scale



Coverage at every layer



Inventory and labels that connect the dots

What excellent monitoring looks like



Simple to set up, built to scale



Coverage at every layer



Inventory and labels that connect the dots



Out of the box alerts

Virtualization & container monitoring



Proxmox

Hyper-V

Podman

Proxmox: Nodes

New checks for cluster grade visibility



Capacity awareness



Cluster resilience

Proxmox: Nodes

New checks for cluster grade visibility



Capacity awareness

- CPU allocation
- Memory allocation
- Storage

Proxmox: Nodes

New checks for cluster grade visibility



Capacity awareness

- CPU allocation
- Memory allocation
- Storage



Cluster resilience

- Replication
- Corosync
- HA Manager
- Processes

Proxmox: VM and LXC

New essential workload-level metrics out of the box



Performance

- CPU utilization
- Disk throughput
- Network throughput























Snapshots

- Last snapshot age



State

- Config lock

OK	Proxmox VE CPU Utilization	  	Total CPU (1 min average): 6.49%, CPU cores assigned: 2, Total CPU Core usage: 0.02
OK	Proxmox VE Disk Throughput	  	Disk read: 0.22 B/s, Disk write: 52.0 kB/s
OK	Proxmox VE Memory Usage	  	Usage: 83.65% - 3.35 GiB of 4.00 GiB
OK	Proxmox VE Network Throughput	  	Inbound: 569 B/s, Outbound: 2.78 kB/s
OK	Proxmox VE VM Backup Status	  	Age: 23 hours 17 minutes, Server local start time: 2026-06-12 21:18:08+02:00, Duration: 9 seconds, Bandwidth: 281 MB/s
OK	Proxmox VE VM Info	  	VM ID: 123, Status: running, Type: qemu, Host: pve, Up since 2026-05-07 08:16:56, Uptime: days 12 hours
OK	Proxmox VE VM Snapshot age	 	No snapshot found

All available before deploying agent!

Proxmox: Labels

Navigate your clusters at scale

By cluster

Cluster: prod-zurich

By type

Type: node

Type: lxc

Type: vm

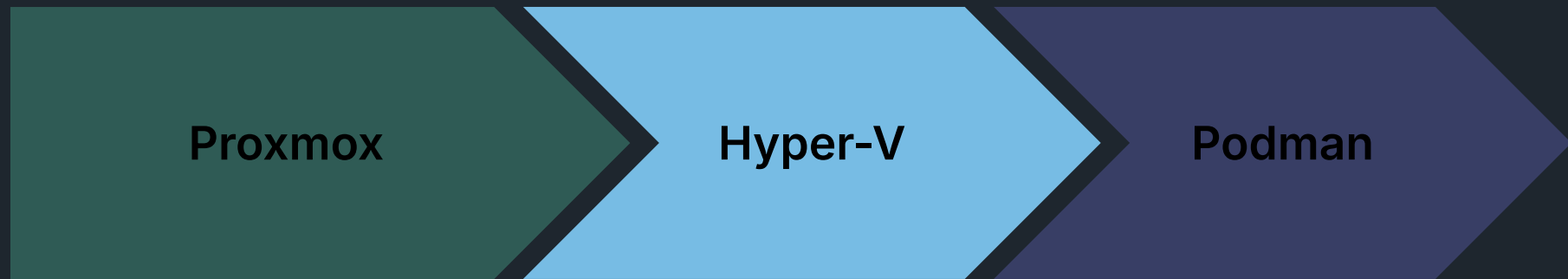


Proxmox

Thank you [@fdriessler](#)
for your great contribution!

inett

Virtualization & container monitoring



Microsoft Hyper-V



Is the VM up?

That's not the
only question

Microsoft Hyper-V



Is the VM up?

That's not the *only* question

Is the VM *configured* right?

- Right Hyper-V generation?
- Integration services healthy?
- On the approved virtual switch?
- Any stale checkpoints?

Microsoft Hyper-V



VM Summary

**Integration
services**

NIC

RAM

Disks

Checkpoints

Microsoft Hyper-V



Contributions

Thank you [@Yogibaer75](#)
for your great contribution!



[Browse](#) [Upload package](#) [How to contribute](#)



Hyper-V Cluster Checks

[Description](#) [All versions](#)

Description

Hyper-V Cluster Checks

Changelog:

- 3.2.7 - small modifications for CMK 2.3
- 3.2.8 - hyperv-vm-nic fix if no data
- 3.2.9 - cluster-roles angepasst
- 3.2.10 - parsing hyperv_io angepasst
- 3.2.11 - more fixes
- 3.2.12 - next fix host io
- 3.2.13 - next fix
- 3.3.0 - migrated to CMK 2.3 API
- 3.3.1 - fixed params handling
- 3.3.2 - typo in check function
- 3.3.3 - some spelling corrected
- 3.3.5 - naming error in bakery rule

Virtualization & container monitoring



Proxmox

Hyper-V

Podman

Podman



247

votes



Upvote

Voting options ▾

Redhat is trying hard to establish Podman as a replacement for Docker, by pushing it as default container engine for Redhat Enterprise products (RHEL, Openshift...).

Sadly, as of today, there is no support for Podman Checks in CMK.

Although there is some compatibility to Docker (docker commands can be aliased to podman), the Docker python module, which agent plugin uses, does not work with Podman.

So a new agent plugin would be needed for Podman support.

📌 Pin this

Released

#Checks&Agents

#Version 2.5

Suggested by: [Mario Sommer](#) mario.sommer@e-steiermark.com (02 Jun, '22) • Upvoted: 16 Mar • [Comments: 22](#)

Edit

Merge

Split

Change status ▾

Tags ▾

Add private note

Ideas Portal: #9 top voted!

Podman: Host



- Container count
- Disk usage
- Pod count
- Summary service

OK	Podman containers	☰ 📁	Total: 5, Running: 5, Created: 0, Paused: 0, Stopped: 0, Exited: 0
OK	Podman disk usage: containers	☰ 📁	Size: 87.8 MiB, Reclaimable: 149 KiB
OK	Podman disk usage: images	☰ 📁	Size: 134 MiB
OK	Podman disk usage: volumes	☰ 📁	Size: 0 B, Reclaimable: 0 B
OK	Podman pods	☰ 📁	Total: 3, Running: 0, Created: 2, Stopped: 1, Dead: 0, Exited: 0, Paused: 0, Degraded: 0

Podman: Containers



- 📊 Status
- 🏥 Health check
- 📊 CPU
- 📊 Memory
- 📊 Container IO
- 📊 Uptime
- 📊 Restarts

OK	Container IO SUMMARY	☰ 📊	Everything looks OK - 2 details available
OK	CPU utilization	☰ 📊	Total CPU: <0.01%
OK	Health	☰	No health check configured or no health status available
OK	Memory	☰ 📊	RAM: <0.01% - 48.0 KiB of 15.4 GiB
OK	Restarts	☰ 📊	Total: 0, In last hour: 0
OK	Status	☰	Running
OK	Uptime	☰ 📊	Up since 2025-12-03 20:07:02, Uptime: 173 days 23 hours

Podman: Containers



- 📍 Status
- 📍 Health check
- 📍 CPU
- 📍 Memory
- 📍 Container IO
- 📍 Uptime
- 📍 Restarts

OK	Container IO SUMMARY	☰ 📄	Everything looks OK - 2 details available
OK	CPU utilization	☰ 📄	Total CPU: <0.01%
OK	Health	☰	No health check configured or no health status available
OK	Memory	☰ 📄	RAM: <0.01% - 48.0 KiB of 15.4 GiB
OK	Restarts	☰ 📄	Total: 0, In last hour: 0
OK	Status	☰	Running
OK	Uptime	☰ 📄	Up since 2025-12-03 20:07:02, Uptime: 173 days 23 hours

Labels: node, pod, user, image, version

Podman: Setup



Auto detect mode

Finds the root socket and all user sockets automatically.

Podman: Setup



Auto detect mode

Finds the root socket and all user sockets automatically.



Socket-less mode

Pull metrics through the Podman CLI.

Virtualization & container monitoring



Proxmox

All the essentials
without the agent

Hyper-V

VM config
fully described

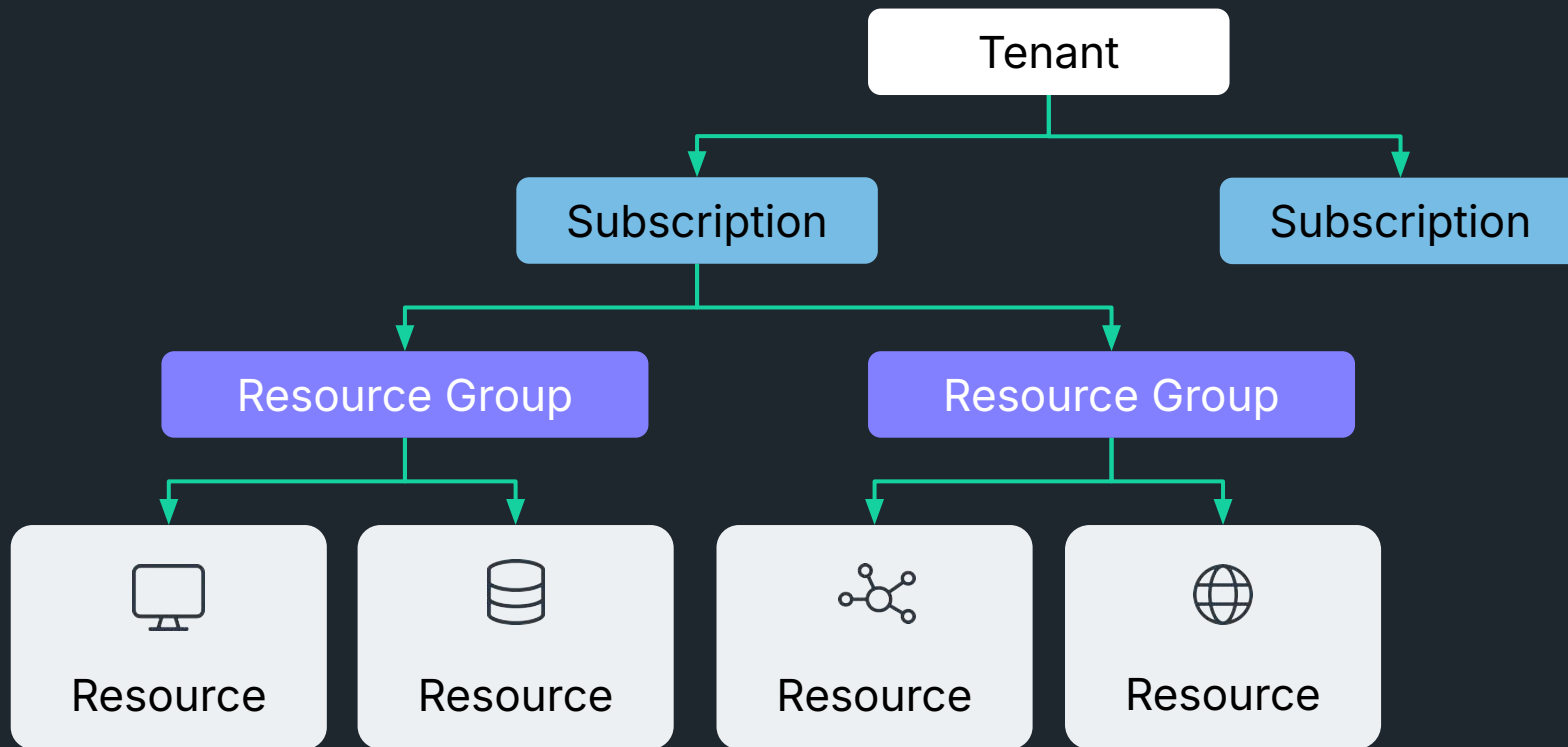
Podman

Rootless and
socket-less mode



Blueprint for future proof cloud monitoring at scale

Azure hierarchy





Where we started

Azure monitoring




Easy to configure



Reliable check plug-ins for core resource types



It worked well for small and medium environments



**Azure environments don't stay
small...**

they scale.

Azure at scale



50+

Subscriptions

1000

s

Resources

5+

Regions

∞

Everything
created
continuously

Azure monitoring at enterprise scale



Setting things up



Performance



Monitoring structure

Setting things up

Checkmk 2.4: Big time commitment right at the start

1

Prepare Azure for Checkmk

1. Register an app in Entra ID. Note down the Directory (tenant) ID, Application (c
2. Assign the built-in role "Reader" to the app in the Access control (IAM). Addition include "Users in Entra ID", "Entra Connect Sync", "App Registrations".
3. Create a client secret for the app and note it down.
4. Return to Checkmk, define a unique configuration name, and use the Subscript

Configuration name
(required)

✘ Subscription ID

Tenant ID / Directory ID
(required)

Client ID / Application ID
(required)

Client Secret

Explicit ▾

Authority

Global ▾

Setting things up

Checkmk 2.5: Getting started in a fraction of the time

One configuration.

Covers the whole org.

- Quick setup
- Filter with Azure tags
- New subscriptions picked up

Configuration name (required)	azure_config_2
Subscriptions to monitor	Monitor all subscriptions ▲ Do not monitor subscriptions Explicit list of subscription IDs Monitor all subscriptions Tag matching subscriptions
Tenant ID / Directory ID (required)	
Client ID / Application ID (required)	
Client Secret	Explicit ▼ *****
Authority	Global ▼
HTTP proxy	<input type="checkbox"/> HTTP proxy

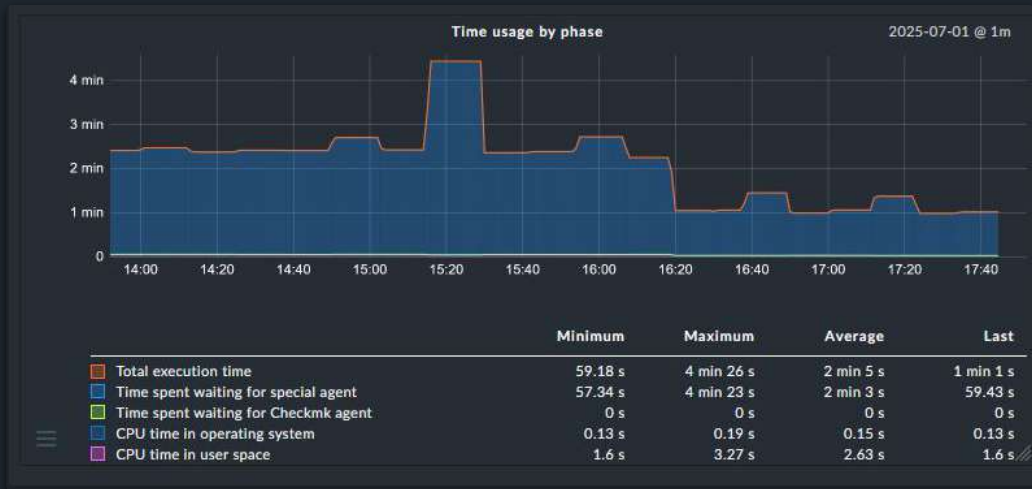
Performance

Checkmk 2.5: reworked special agent for very large setups

Concurrent execution.

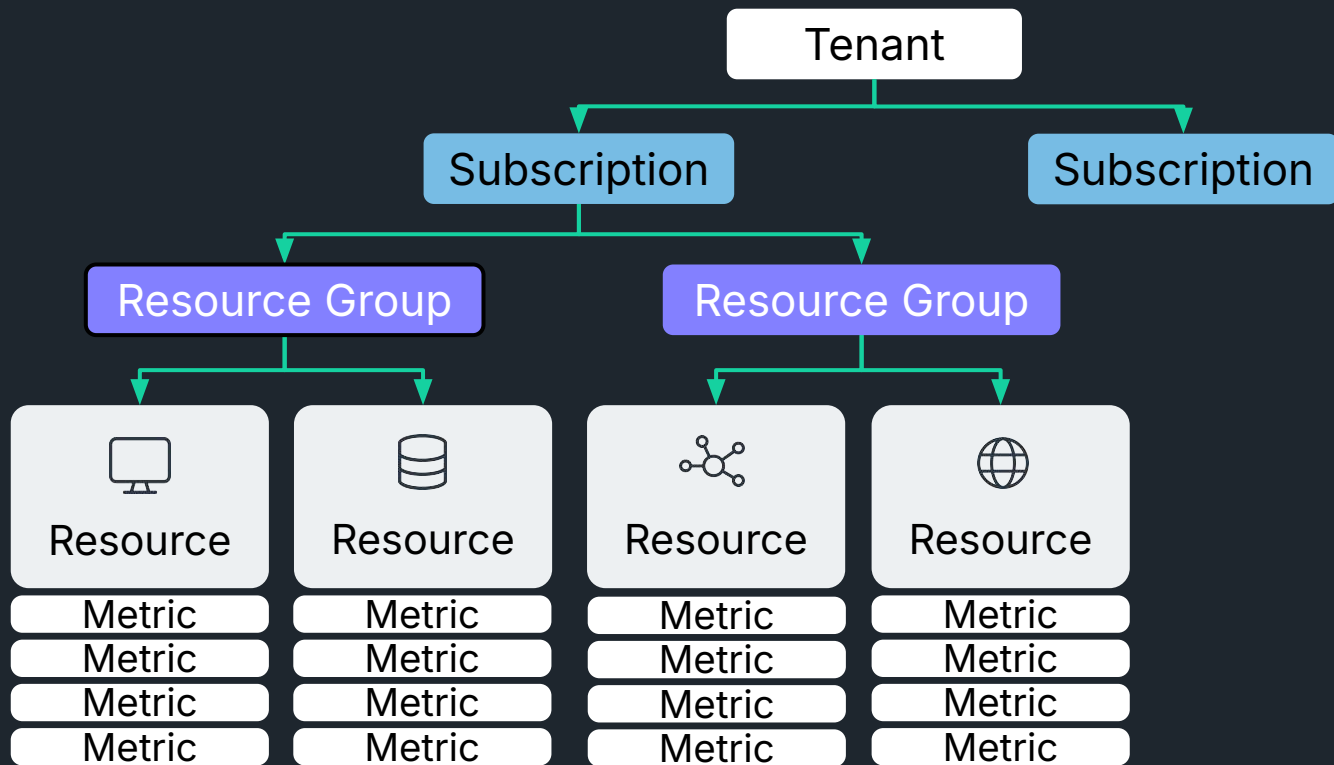
Efficient. Resilient.

- 🛡️ Optimized special agent
- 🛡️ Improved performance
- 🛡️ Less resource usage



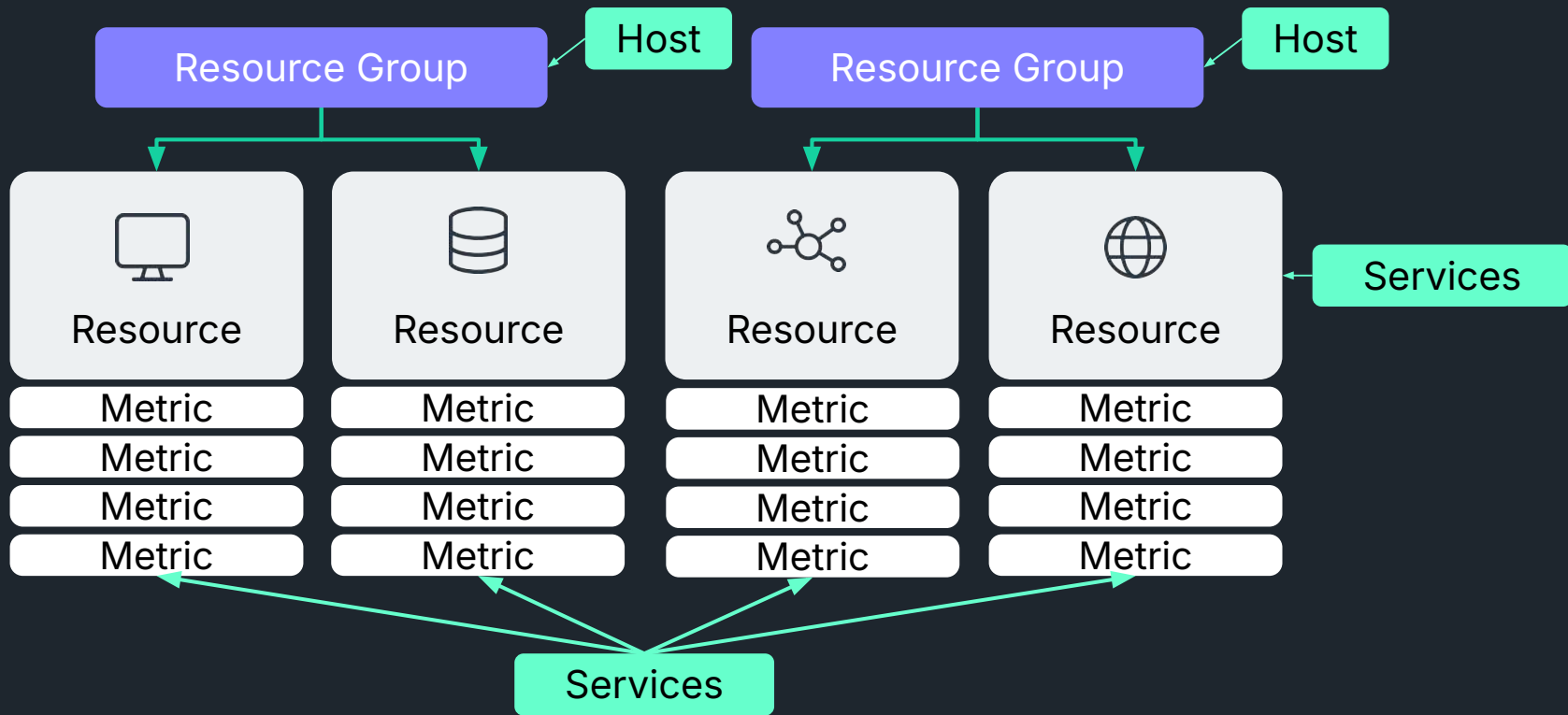
Monitoring structure

Checkmk 2.4: Understanding your data took a while too



Monitoring structure

Checkmk 2.4: Understanding your data took a while too



Monitoring structure

Checkmk 2.5: Native cloud to monitoring structure

The monitoring structure aligns with the Azure hierarchy.

- One host per resource
- Azure metadata inside Checkmk
- Easy data exploration

Monitor > Overview > [All hosts](#)

All hosts

Commands Hosts Export Display Help

Local site stable

State	Host	Icons
UP	marentz-db-server-master_8ee5a95d	
UP	networkwatcherrg_7e5c59ab	
UP	networkwatcherrg_cbd6cc93	
UP	now_host	
UP	relirod-rg_cbd6cc93	
UP	relirod-tast-gw_c38160c7	
UP	rene-cache-redis_74349a40	
UP	rene-firewall_bt29cc3d	
UP	rene_nat_gateway_775fff20	
UP	rg-consulting-weu_7e5c59ab	
UP	rg-dev-weu_7e5c59ab	
UP	rg-dev-weu_cbd6cc93	
UP	rg-hub-network-weu_7e5c59ab	
UP	rg-packer-dev-weu_7e5c59ab	
UP	rg-pm-weu_7e5c59ab	
UP	rg-spokes-network-weu_7e5c59ab	
UP	rgpmweu0f2_85824be2	

Monitor > Overview > Main_dashboard

Dashboard Main_dashboard Filter

Share Settings Clone

Host statistics

- 50 Up
- 0 In downtime
- 0 Unreachable
- 0 Down
- 50 Total

Service statistics

- 332 OK
- 0 In downtime
- 0 On down host
- 58 Warning
- 4 Unknown
- 40 Critical
- 434 Total

Problem notifications

Total host problems

Total service problems

Host overview

Top alerters (last 7 days)

Service
Azure/Cosmos DB Storage
Azure agent info
Temperature Zone 9
Temperature Zone 10
Interface 5
Interface 3
Check_MK
DMD dashboard1 performance
DMD dashboard Notification Spooler
Temperature Zone 0
DMD dashboard Event Console
DMD dashboard performance
Site dashboard statistics
Memory
Check_MK
Check_MK
Check_MK
Check_MK
Check_MK
Check_MK

Link to video: [here!](#)

- 1 dashboard1
- 1 dashboard1
- 1 dashboard1
- 1 azr_cosmosdb_david-cosmos-account_SampleDB_3f40ba0d
- 1 dashboard1
- 1 dashboard1
- 1 dashboard1
- 21 dashboard1
- 3 azr_atw_rene_firewall_bf29cc3d
- 3 azr_ng_reno_nat_gateway_775fff20
- 3 azr_redis_rone-cache-redis_74349a40
- 3 azr_sqldb_marentz-db-server_marentz_db_0ee5a95d
- 3 azure_v2_1

Monitor

Search

Monitor

Customize

Setup

Changes

Help

User

Sidebar

Expanded coverage

Working with the data in Checkmk 2.5: new checks



Redis Cache

Cosmos DB

NAT Gateway

Azure Firewall

Azure monitoring in Checkmk 2.5



Setting things up

- Quick setup
- All subscriptions with 1 rule
- Filter with Azure tags

Performance

- Optimized special agent
- Improved performance
- Less resource usage

Monitoring structure

- Intuitive dashboards
- Easy data exploration
- Azure metadata inside Checkmk

Result

Checkmk **2.5** makes Azure monitoring **easier** to set up, **faster** to run, and **simpler** to operate at **enterprise scale**.

Putting it all together



Virtualization

Proxmox: agentless
node and VM/LCX
visibility

Hyper-V: VM config



Container

Podman: rootless and
socket-less mode



Cloud

Azure: all subscriptions,
ready in minutes



Whatever your infrastructure looks like tomorrow, you'll be able to **monitor** it with Checkmk, **at scale!**

Checkmk #12

Conference