Monitoring Microsoft Azure – a practitioner's example

CHECKMK CONFERENCE #6 – MUNICH, APRIL 29, 2020

tribe29

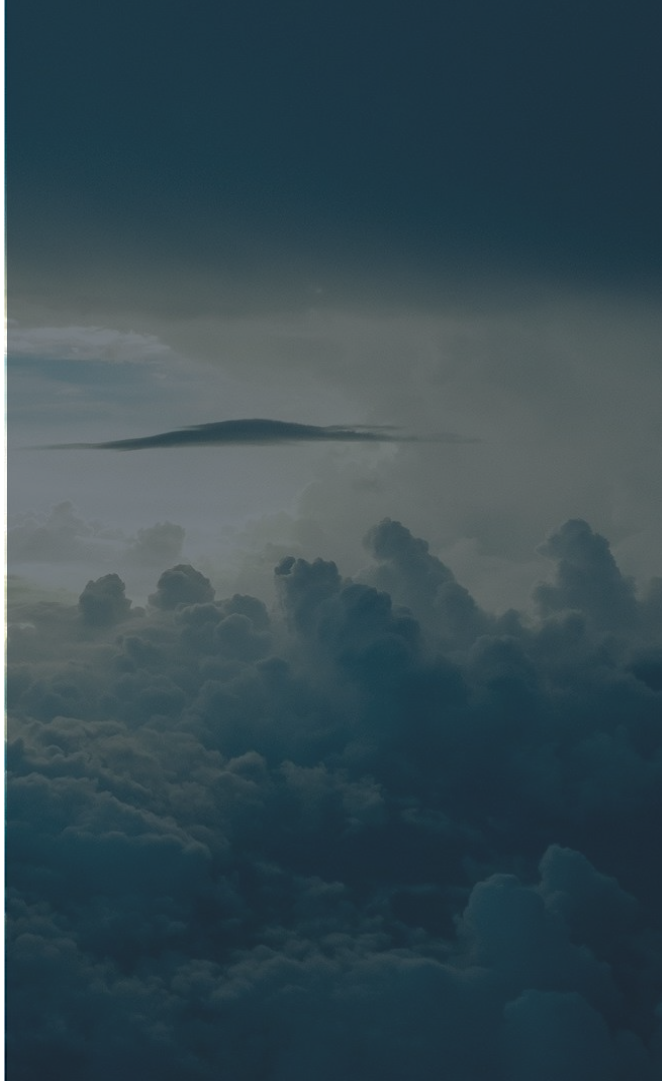
About me

- Andreas Döhler
  - Linux since 1998 (from time to time)
  - Using Checkmk since 2009
  - Spare time mostly sport & electronics & forum answer guy :)
- Bechtle
  - 75 locations in D-A-CH region
  - IT-Systemhaus & IT-E-Commerce
  - 11.500 Employees
  - Home Location Chemnitz-Dresden-Cottbus

Monitoring the cloud…

- Situation: Customer decided to migrate entire on-premises datacenter to Microsoft Azure
- Customer is a medium-sized company with 60-70 VMs
- Needs to be able monitor his IT in the cloud as well
- No cloud expert myself, but had to do it
- Share my lessons learned today

If you are an IT service provider,
you might face the same challenge next.

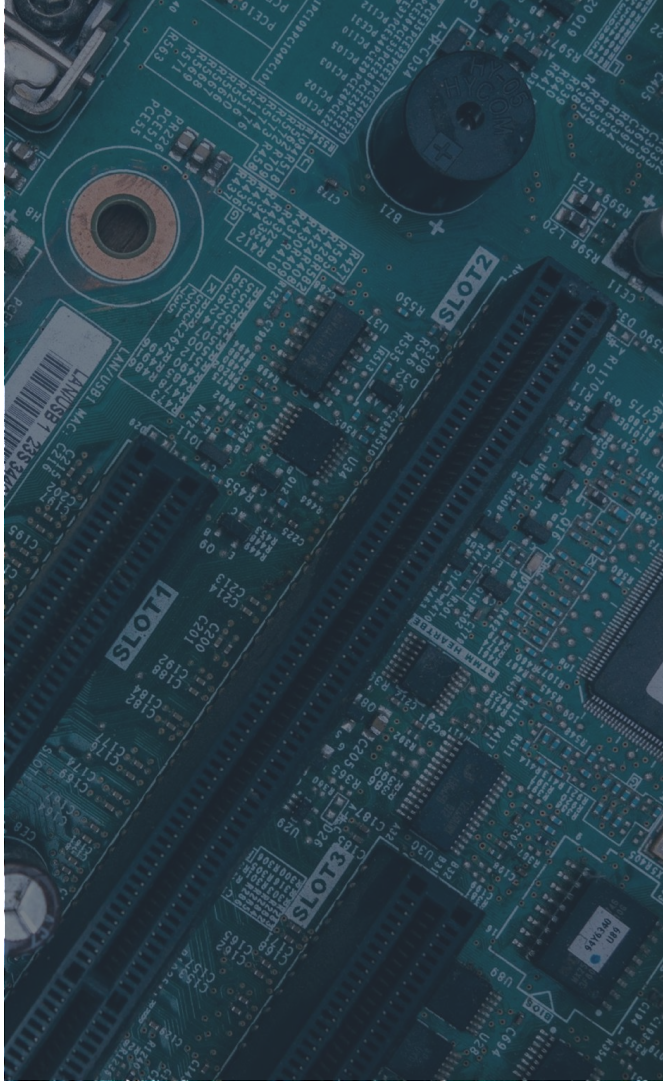


Agenda

1.Checkmk server deployment in the cloud

2.Monitoring cloud services

3.What’s next: End-to-End monitoring



Deploying Checkmk the easy or challenging way

Option 1

Use the virtual appliance

Option 2

Install standard package on a VM

The virtual appliance comes with many benefits

- Built-in HA possible
- No need for managing the operating system
- Easy deployment

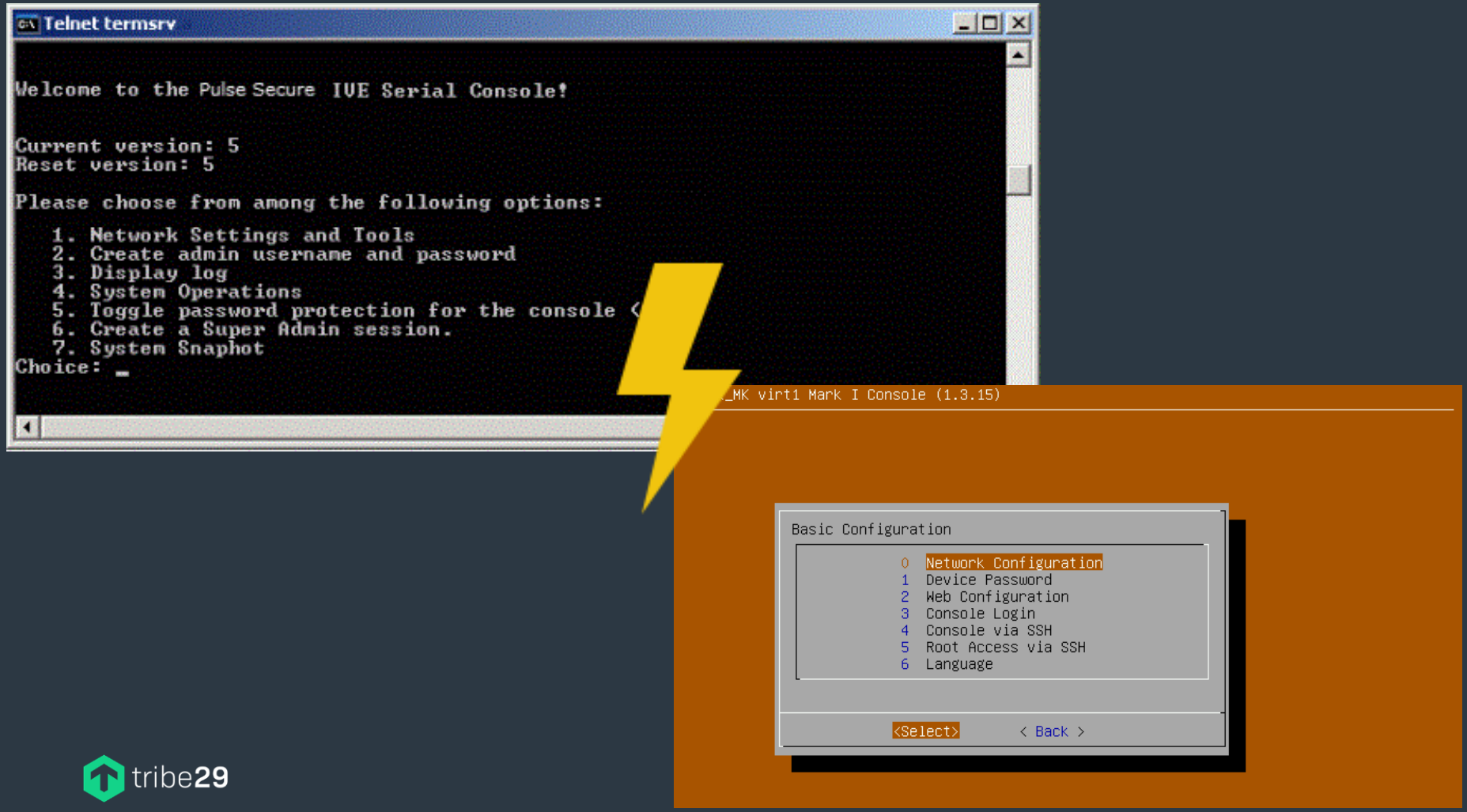
Ideal for deployment of Checkmk in virtual environments … like a cloud

But I ran into problems...

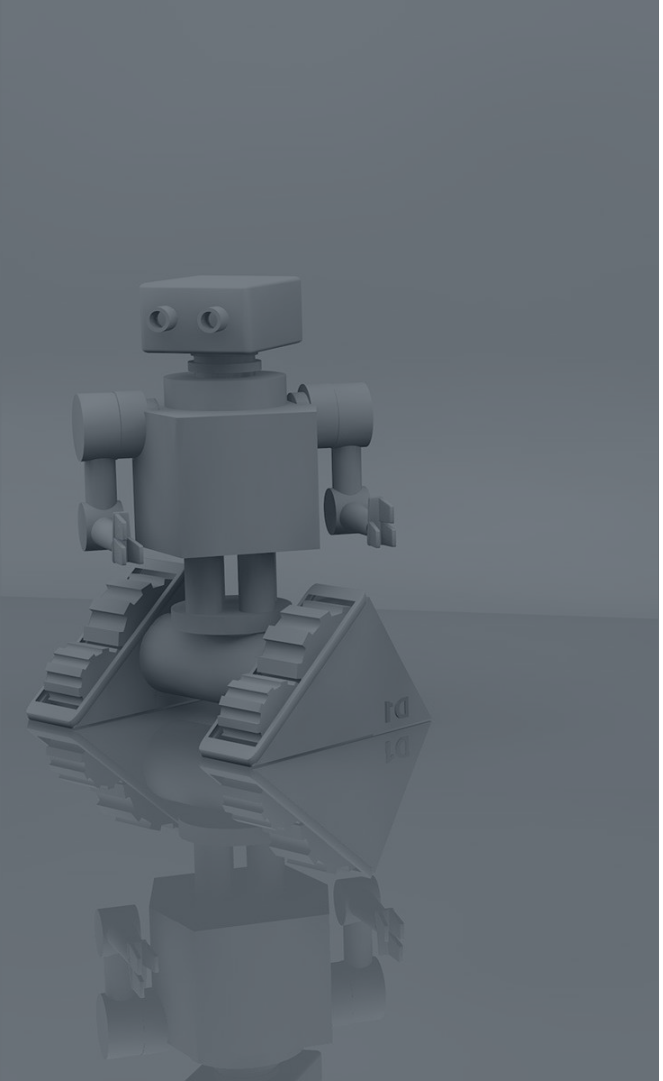
- Azure VM without any IP configuration only provide **serial console access**
- This is a “real” serial console
- Test with preconfigured appliance image from local Hyper-V - disk init step could be solved with this

Had to decide as every upload of a preconfigured VM takes time.

This was the challenging option


Telnet termsrv
Welcome to the Pulse Secure IVE Serial Console!
Current version: 5
Reset version: 5
Please choose from among the following options:
1. Network Settings and Tools
2. Create admin username and password
3. Display log
4. System Operations
5. Toggle password protection for the console (
6. Create a Super Admin session.
7. System Snaphot
Choice:
_MK virt1 Mark I Console (1.3.15)
Basic Configuration
0 Network Configuration
1 Device Password
2 Web Configuration
3 Console Login
4 Console via SSH
5 Root Access via SSH
6 Language
<Select>
< Back >

tribe29

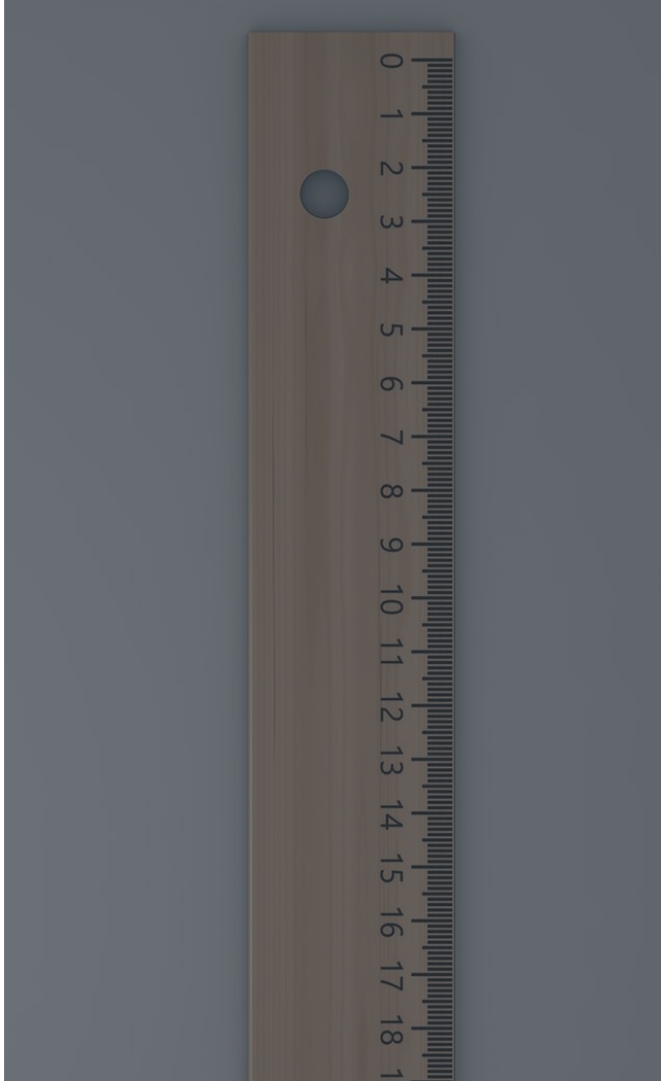


The solution: Just deploy Checkmk on a VM

- Creating a VM is easy
- Fully automated deployment of Linux inside Azure
- Installing Checkmk on VM is very easy
- Decision needed how big the VM should be → monthly cost

Sizing the VM

- In the cloud, rather ~20 services per host instead of 30-40 services
- How many cores are needed?
  - 4 cores - 2 would be enough if no graphic environment is present for some web tests
- How much memory?
  - 8 GB would be enough but you need to go with the templates from Azure, in our case the 4 core machine had 32 GB
- How much HDD?
  - A small 32 GB is here enough if you don’t plan to monitor many hosts (no network devices in our setup)

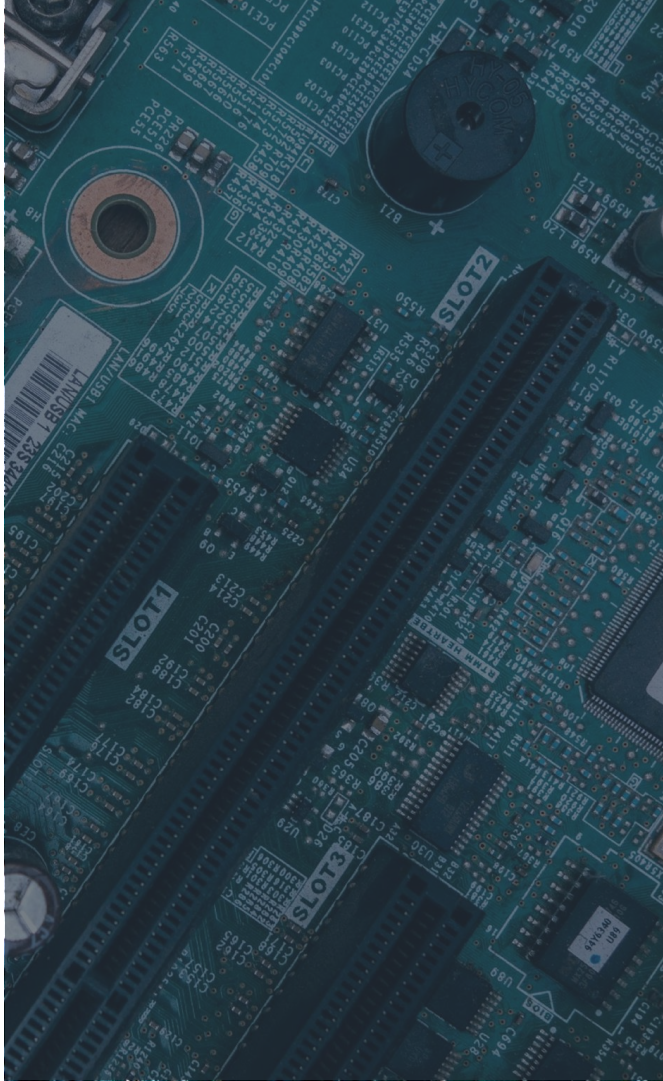


What about my network infrastructure?

- You might be able to migrate your servers into the cloud
- Your switches and office network do not migrate to the cloud though
- Instead of monitoring them from the cloud, create an extra Checkmk site locally

Agenda

1. Checkmk server deployment in the cloud
2. **Monitoring cloud services**
3. What’s next: End-to-End monitoring



What happened to all my servers?

Database Servers

Database Services

Load Balancers

Load Balancing Service

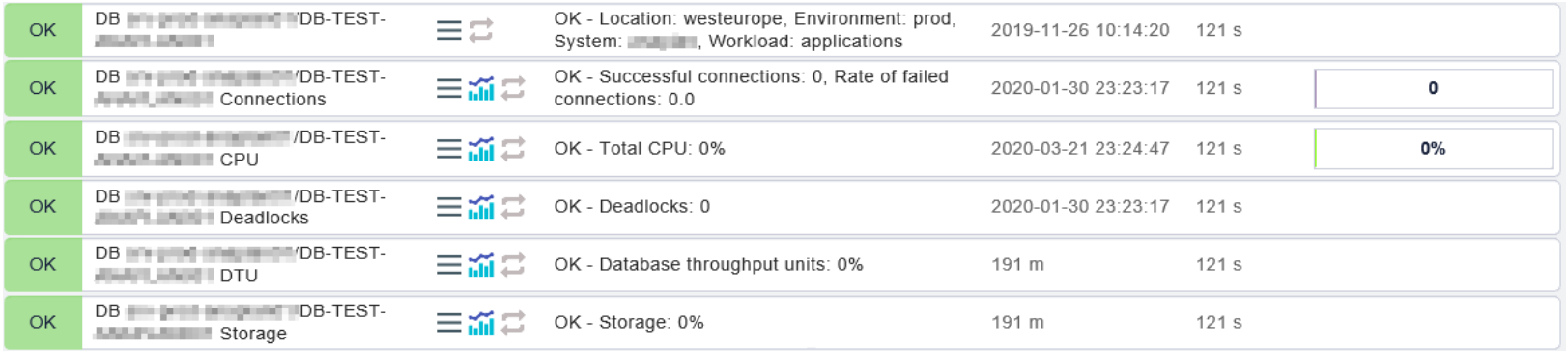
Application Servers

VM with Application

What does that mean for my monitoring?

- Monitor with Azure Special Agent
  - Define the resource groups you want to see. Start with one or two for testing.
  - Compare the results from the Special Agent with your Azure dashboard
  - Not all resource types existing in Azure are supported by the Special Agent at the moment
- You already get most of the data you need
- Complement with Linux or Windows agent for virtual machines in the cloud

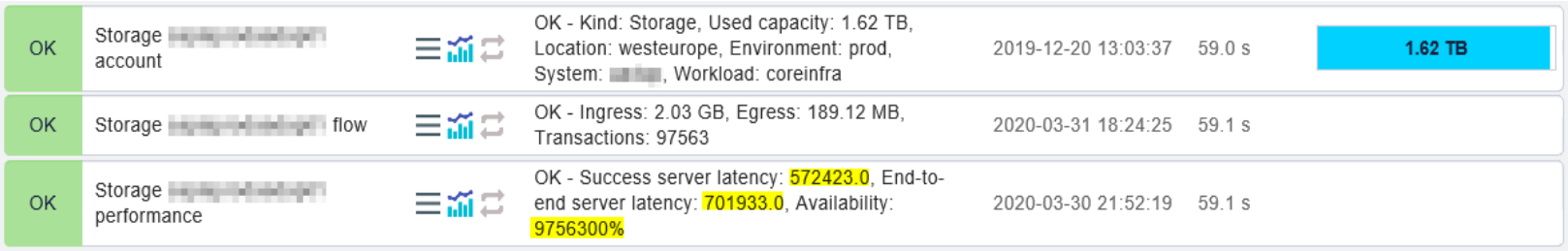
Monitoring databases in Azure



OK	DB /DB-TEST-	OK - Location: westeurope, Environment: prod, System: , Workload: applications	2019-11-26 10:14:20	121 s	
OK	DB /DB-TEST- Connections	OK - Successful connections: 0, Rate of failed connections: 0.0	2020-01-30 23:23:17	121 s	0
OK	DB /DB-TEST- CPU	OK - Total CPU: 0%	2020-03-21 23:24:47	121 s	0%
OK	DB /DB-TEST- Deadlocks	OK - Deadlocks: 0	2020-01-30 23:23:17	121 s	
OK	DB /DB-TEST- DTU	OK - Database throughput units: 0%	191 m	121 s	
OK	DB /DB-TEST- Storage	OK - Storage: 0%	191 m	121 s	

- Use Azure Special Agent + Azure SQL Databases Checks
- Problems
  - Not all data sent from API, from time to time

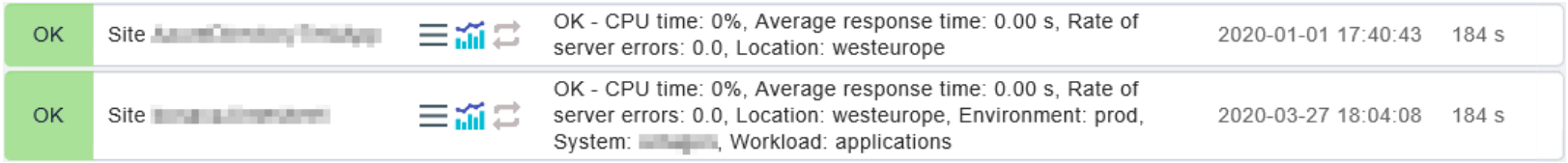
Monitoring storage accounts in Azure



OK	Storage [redacted] account	OK - Kind: Storage, Used capacity: 1.62 TB, Location: westeurope, Environment: prod, System: [redacted], Workload: coreinfra	2019-12-20 13:03:37	59.0 s	1.62 TB
OK	Storage [redacted] flow	OK - Ingress: 2.03 GB, Egress: 189.12 MB, Transactions: 97563	2020-03-31 18:24:25	59.1 s	
OK	Storage [redacted] performance	OK - Success server latency: 572423.0, End-to-end server latency: 701933.0, Availability: 9756300%	2020-03-30 21:52:19	59.1 s	

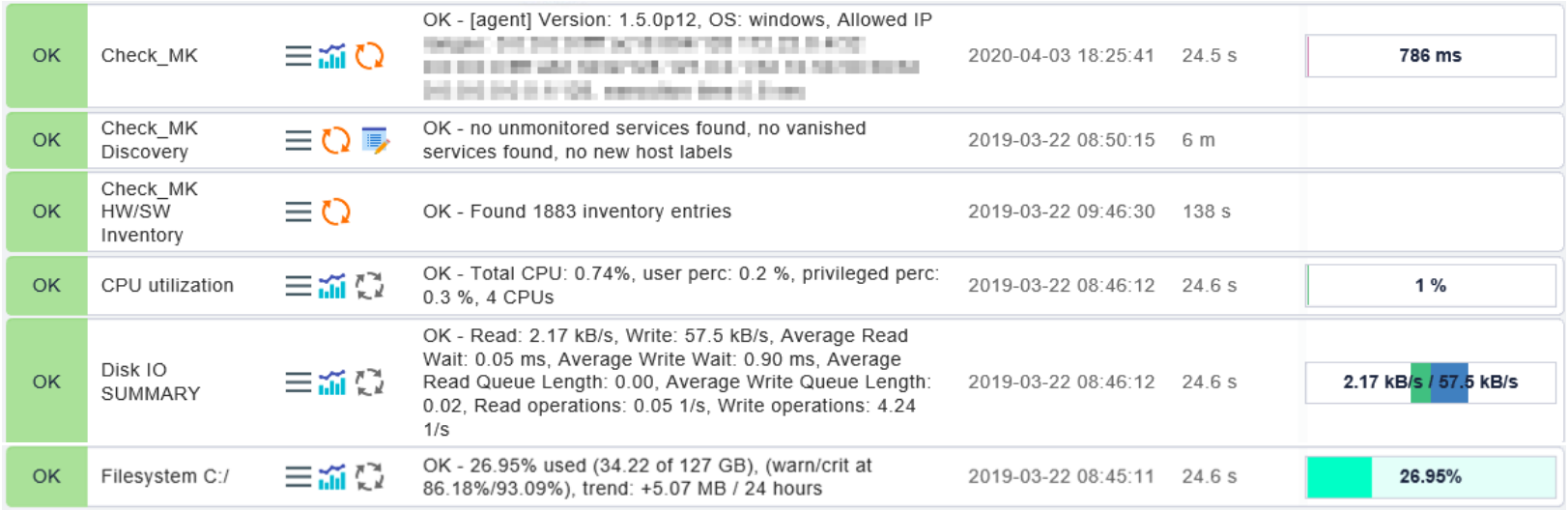
- Use Azure Special Agent + Azure Storage Accounts Check
- Strange values from latency
  - As long as both values are in the same range it is ok
  - Example are 5700ms and 7019ms

Monitoring networking & web services in Azure



- Use Azure Special Agent + Azure Virtual Network Gateway Check
- In my environment only Sites are monitored and used
- Agent can also give info on:
  - Point-to-site connections
  - Point-to-site bandwidth
  - Site-to-site bandwidth

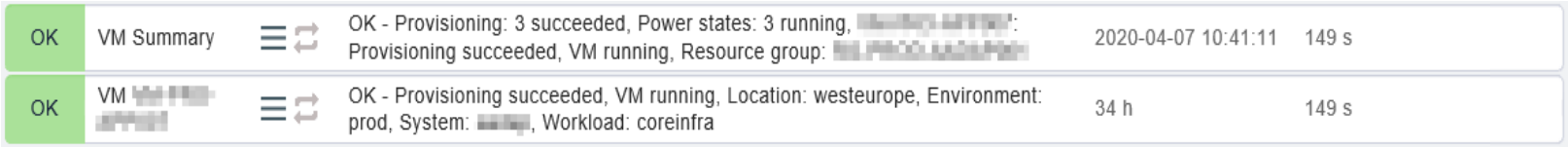
Monitoring application servers in Azure



OK	Check_MK	OK - [agent] Version: 1.5.0p12, OS: windows, Allowed IP [redacted]	2020-04-03 18:25:41	24.5 s	786 ms
OK	Check_MK Discovery	OK - no unmonitored services found, no vanished services found, no new host labels	2019-03-22 08:50:15	6 m	
OK	Check_MK HW/SW Inventory	OK - Found 1883 inventory entries	2019-03-22 09:46:30	138 s	
OK	CPU utilization	OK - Total CPU: 0.74%, user perc: 0.2 %, privileged perc: 0.3 %, 4 CPUs	2019-03-22 08:46:12	24.6 s	1 %
OK	Disk IO SUMMARY	OK - Read: 2.17 kB/s, Write: 57.5 kB/s, Average Read Wait: 0.05 ms, Average Write Wait: 0.90 ms, Average Read Queue Length: 0.00, Average Write Queue Length: 0.02, Read operations: 0.05 1/s, Write operations: 4.24 1/s	2019-03-22 08:46:12	24.6 s	2.17 kB/s / 57.5 kB/s
OK	Filesystem C:/	OK - 26.95% used (34.22 of 127 GB), (warn/crit at 86.18%/93.09%), trend: +5.07 MB / 24 hours	2019-03-22 08:45:11	24.6 s	26.95%

- Nothing has changed - just use the Checkmk Agent on the VM + configure the firewall
- In my environment mostly Windows VMs

Monitoring virtual machines



OK	VM Summary		OK - Provisioning: 3 succeeded, Power states: 3 running, [redacted]: Provisioning succeeded, VM running, Resource group: [redacted]	2020-04-07 10:41:11	149 s
OK	VM [redacted]		OK - Provisioning succeeded, VM running, Location: westeurope, Environment: prod, System: [redacted], Workload: coreinfra	34 h	149 s

- There is also a check for an overview of VMs
- Basically like the VMware vSphere check
- A little bit more basic compared to VMware
- Data is per resource group

Other challenges when monitoring in the cloud

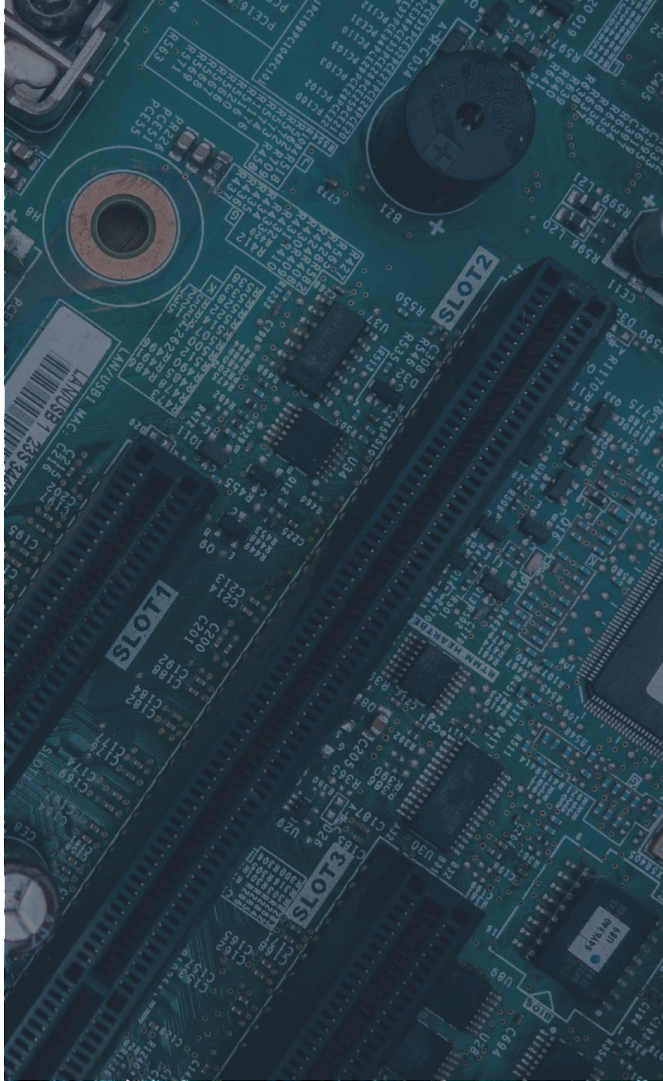
- Behavior of your queried API
  - Sometimes you get only partial data but also no error message
  - Staleness is very helpful for such situations if you don’t want many messages “Item not found in agent output”
  - With Mortiz Kiemer from t29: Build a solution that checks go stale if data is missing
- Very complex infrastructure if you don’t know Azure from the ground up
- Services not covered at the moment
  - Azure Vaults, Action Books, Automation Accounts and Connections as example
- API Limits are no problem with 40 resource groups at the moment

No check plug-in for my cloud service available...

- What about writing it yourself?
  - The Azure API is not “simple”
  - You can always ask tribe29, I heard they have new developers
  - The special agent gives you the found resources also if there is no check available
- Be creative
  - HTTP checks for load balancers
  - Own checks for BizTalk queues and their status

Agenda

1. Checkmk server deployment in the cloud
2. Monitoring cloud services
3. **What’s next: End-to-End monitoring**



The black box problem

- General problem in the cloud: a lot of abstraction
- You don’t see the inner working of a “service”, but get a “black box”
- Can be solved with end-to-end testing
- Fortunately, most applications in the cloud are web applications!

Next steps: Implement E2E monitoring

- There are many tools, e.g. Sakuli - however ...
- Current plan:
  - Headless container with Chrome
  - Control via Python API
  - Results as Checkmk plug-in or local check and integrate into my monitoring
- Anyone with experiences or tips? Reach out to me

Thank you

tribe29 GmbH
Kellerstraße 29
81667 München
Deutschland

Web — tribe29.com
E-Mail — mail@tribe29.com