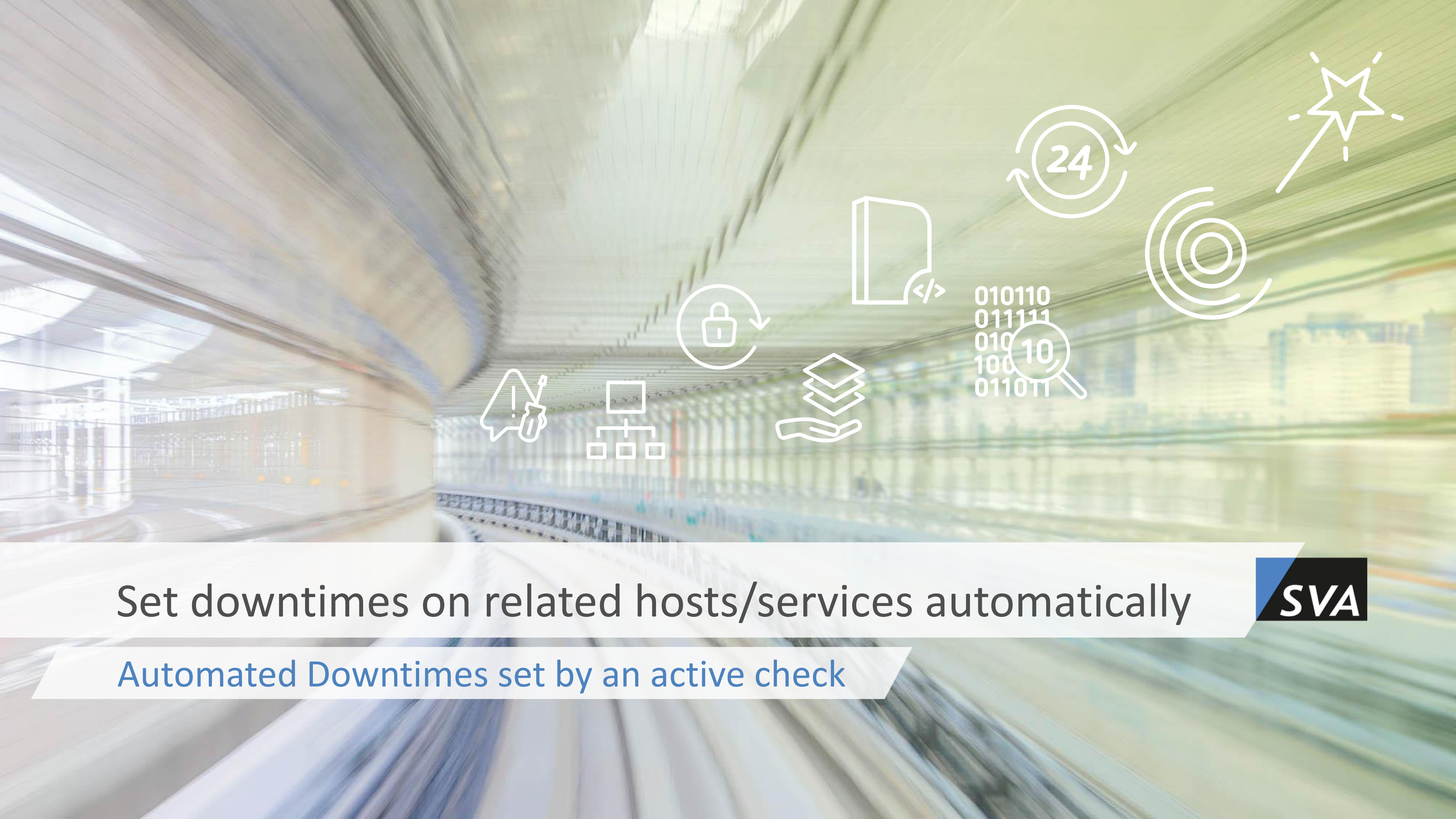


Set downtimes on related hosts/services automatically

Automated Downtimes set by an active check

/ Automated Downtimes

1 / Origin story: Detect maintenance from ESXi

2 / Use cases & Configuration

3 / Dependency detection

4 / Thorny technical challenges

5 / How to use & contribute

/ Speaker

Michael Höß

- Monitoring at SVA
- Playing with tech when not working with tech

Gerd Stolz

- Monitoring at SVA
- Checkmk user since 1.2.4
- Dad & hobby Ninja athlete

/ SVA

Biggest owner-operated system integrator in Germany

Steady growth with more than 3.200 employees in Germany

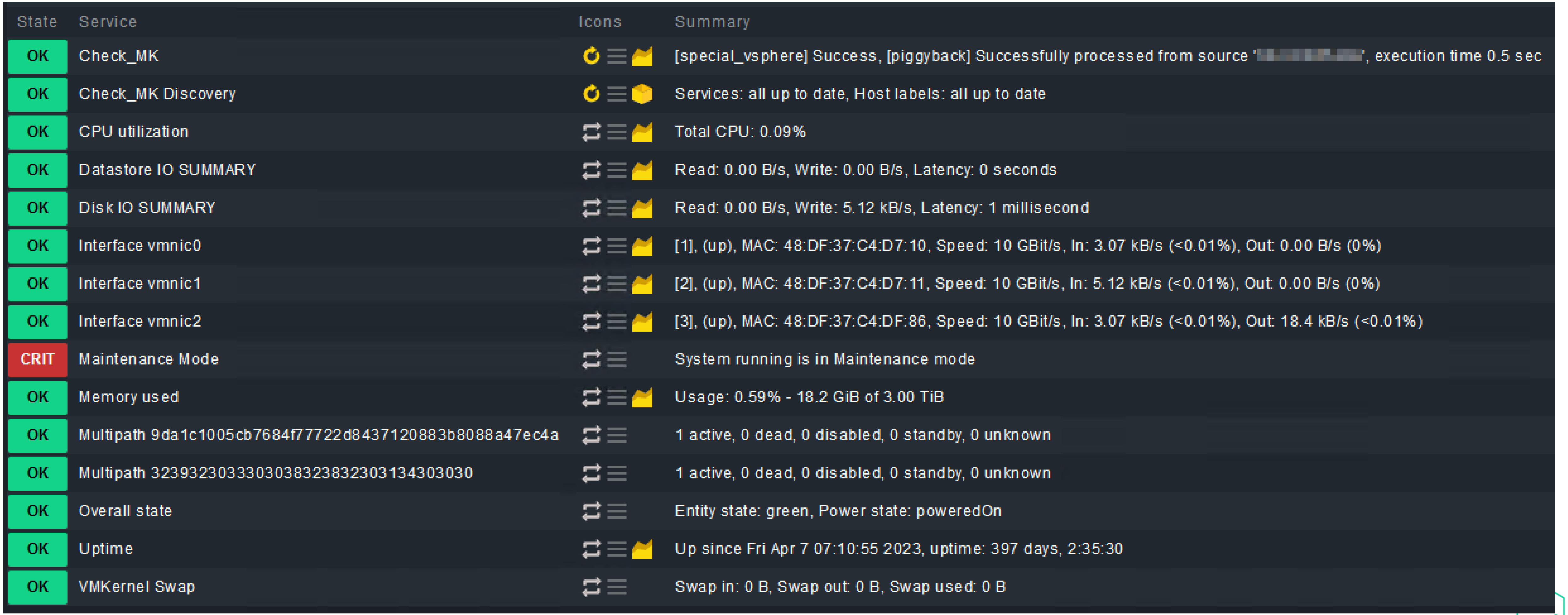
27 locations in Germany

More than 100 vendor partnerships

Extensive Demo Lab with hardware + software solutions

24x7 Operations Center based in Wiesbaden

/ Detect maintenance ESXi

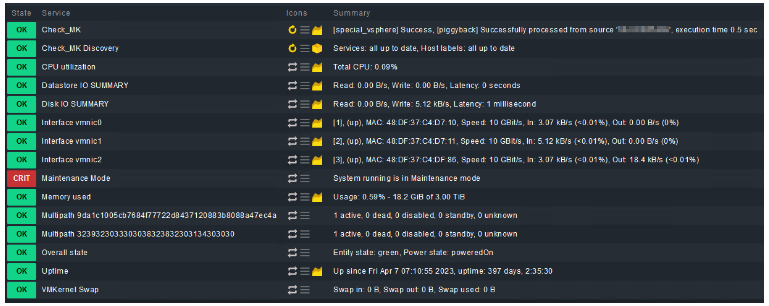


State	Service	Icons	Summary
OK	Check_MK	⟳ ≡ ◢	[special_vsphere] Success, [piggyback] Successfully processed from source '[redacted]', execution time 0.5 sec
OK	Check_MK Discovery	⟳ ≡ ⬢	Services: all up to date, Host labels: all up to date
OK	CPU utilization	⇄ ≡ ◢	Total CPU: 0.09%
OK	Datastore IO SUMMARY	⇄ ≡ ◢	Read: 0.00 B/s, Write: 0.00 B/s, Latency: 0 seconds
OK	Disk IO SUMMARY	⇄ ≡ ◢	Read: 0.00 B/s, Write: 5.12 kB/s, Latency: 1 millisecond
OK	Interface vmnic0	⇄ ≡ ◢	[1], (up), MAC: 48:DF:37:C4:D7:10, Speed: 10 GBit/s, In: 3.07 kB/s (<0.01%), Out: 0.00 B/s (0%)
OK	Interface vmnic1	⇄ ≡ ◢	[2], (up), MAC: 48:DF:37:C4:D7:11, Speed: 10 GBit/s, In: 5.12 kB/s (<0.01%), Out: 0.00 B/s (0%)
OK	Interface vmnic2	⇄ ≡ ◢	[3], (up), MAC: 48:DF:37:C4:DF:86, Speed: 10 GBit/s, In: 3.07 kB/s (<0.01%), Out: 18.4 kB/s (<0.01%)
CRIT	Maintenance Mode	⇄ ≡	System running is in Maintenance mode
OK	Memory used	⇄ ≡ ◢	Usage: 0.59% - 18.2 GiB of 3.00 TiB
OK	Multipath 9da1c1005cb7684f77722d8437120883b8088a47ec4a	⇄ ≡	1 active, 0 dead, 0 disabled, 0 standby, 0 unknown
OK	Multipath 3239323033330303832383230313430 3030	⇄ ≡	1 active, 0 dead, 0 disabled, 0 standby, 0 unknown
OK	Overall state	⇄ ≡	Entity state: green, Power state: poweredOn
OK	Uptime	⇄ ≡ ◢	Up since Fri Apr 7 07:10:55 2023, uptime: 397 days, 2:35:30
OK	VMKernel Swap	⇄ ≡	Swap in: 0 B, Swap out: 0 B, Swap used: 0 B

/ Detect maintenance ESXi

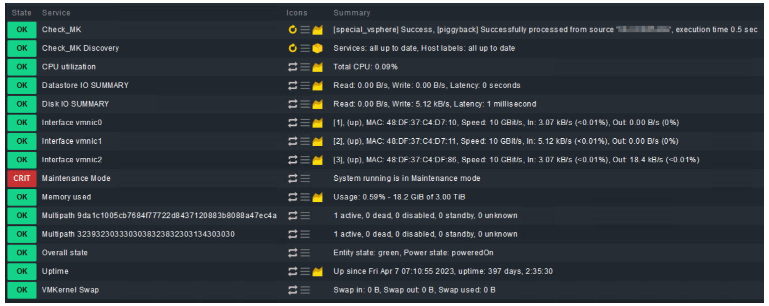
VMware ESXi hosts are usually set to maintenance by hand

- Setting the downtime in Checkmk is an additional manual task



State	Service	Icons	Summary
OK	Check_MK	↻ ☰	[special_vsphere] Success, [piggyback] Successfully processed from source '[redacted]', execution time 0.5 sec
OK	Check_MK Discovery	↻ ☰	Services: all up to date, Host labels: all up to date
OK	CPU utilization	⇄ ☰	Total CPU: 0.09%
OK	Datastore IO SUMMARY	⇄ ☰	Read: 0.00 B/s, Write: 0.00 B/s, Latency: 0 seconds
OK	Disk IO SUMMARY	⇄ ☰	Read: 0.00 B/s, Write: 5.12 kB/s, Latency: 1 millisecond
OK	Interface vmnic0	⇄ ☰	[1], (up), MAC: 48:DF:37:C4:D7:10, Speed: 10 GBit/s, In: 3.07 kB/s (<0.01%), Out: 0.00 B/s (0%)
OK	Interface vmnic1	⇄ ☰	[2], (up), MAC: 48:DF:37:C4:D7:11, Speed: 10 GBit/s, In: 5.12 kB/s (<0.01%), Out: 0.00 B/s (0%)
OK	Interface vmnic2	⇄ ☰	[3], (up), MAC: 48:DF:37:C4:DF:86, Speed: 10 GBit/s, In: 3.07 kB/s (<0.01%), Out: 18.4 kB/s (<0.01%)
CRIT	Maintenance Mode	⇄ ☰	System running is in Maintenance mode
OK	Memory used	⇄ ☰	Usage: 0.59% - 18.2 GiB of 3.00 TiB
OK	Multipath 9da1c1005cb7684f77722d8437120883b8088a47ec4a	⇄ ☰	1 active, 0 dead, 0 disabled, 0 standby, 0 unknown
OK	Multipath 3239323033303038323832303134303030	⇄ ☰	1 active, 0 dead, 0 disabled, 0 standby, 0 unknown
OK	Overall state	⇄ ☰	Entity state: green, Power state: poweredOn
OK	Uptime	⇄ ☰	Up since Fri Apr 7 07:10:55 2023, uptime: 397 days, 2:35:30
OK	VMKernel Swap	⇄ ☰	Swap in: 0 B, Swap out: 0 B, Swap used: 0 B

/ Detect maintenance ESXi



State	Service	Icons	Summary
OK	Check_MK	↻ ≡ 📈	[special_vsphere] Success, [piggyback] Successfully processed from source '[redacted]', execution time 0.5 sec
OK	Check_MK Discovery	↻ ≡ 📦	Services: all up to date, Host labels: all up to date
OK	CPU utilization	⇄ ≡ 📈	Total CPU: 0.09%
OK	Datastore IO SUMMARY	⇄ ≡ 📈	Read: 0.00 B/s, Write: 0.00 B/s, Latency: 0 seconds
OK	Disk IO SUMMARY	⇄ ≡ 📈	Read: 0.00 B/s, Write: 5.12 kB/s, Latency: 1 millisecond
OK	Interface vmnic0	⇄ ≡ 📈	[1], (up), MAC: 48:DF:37:C4:D7:10, Speed: 10 GBit/s, In: 3.07 kB/s (<0.01%), Out: 0.00 B/s (0%)
OK	Interface vmnic1	⇄ ≡ 📈	[2], (up), MAC: 48:DF:37:C4:D7:11, Speed: 10 GBit/s, In: 5.12 kB/s (<0.01%), Out: 0.00 B/s (0%)
OK	Interface vmnic2	⇄ ≡ 📈	[3], (up), MAC: 48:DF:37:C4:DF:86, Speed: 10 GBit/s, In: 3.07 kB/s (<0.01%), Out: 18.4 kB/s (<0.01%)
CRIT	Maintenance Mode	⇄ ≡	System running is in Maintenance mode
OK	Memory used	⇄ ≡ 📈	Usage: 0.59% - 18.2 GiB of 3.00 TiB
OK	Multipath 9da1c1005cb7684f77722d8437120883b8088a47ec4a	⇄ ≡	1 active, 0 dead, 0 disabled, 0 standby, 0 unknown
OK	Multipath 3239323033303038323832303134303030	⇄ ≡	1 active, 0 dead, 0 disabled, 0 standby, 0 unknown
OK	Overall state	⇄ ≡	Entity state: green, Power state: poweredOn
OK	Uptime	⇄ ≡ 📈	Up since Fri Apr 7 07:10:55 2023, uptime: 397 days, 2:35:30
OK	VMKernel Swap	⇄ ≡	Swap in: 0 B, Swap out: 0 B, Swap used: 0 B

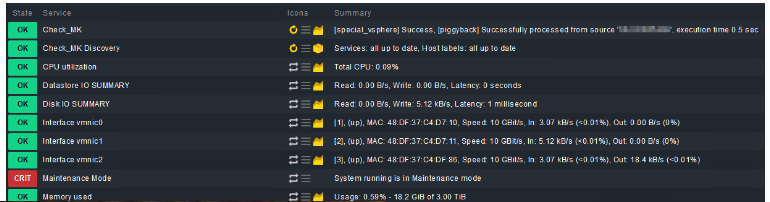
VMware ESXi hosts are usually set to maintenance by hand

- Setting the downtime in Checkmk is an additional manual task

- Solution
  - check that monitors if the service „Maintenance Mode“ has „is in maintenance mode“ in the plugin output
    - If so => set host into downtime
    - If not => remove downtime

  - Simple installation via MKP and configuration of active check in the GUI

/ Detect maintenance ESXi



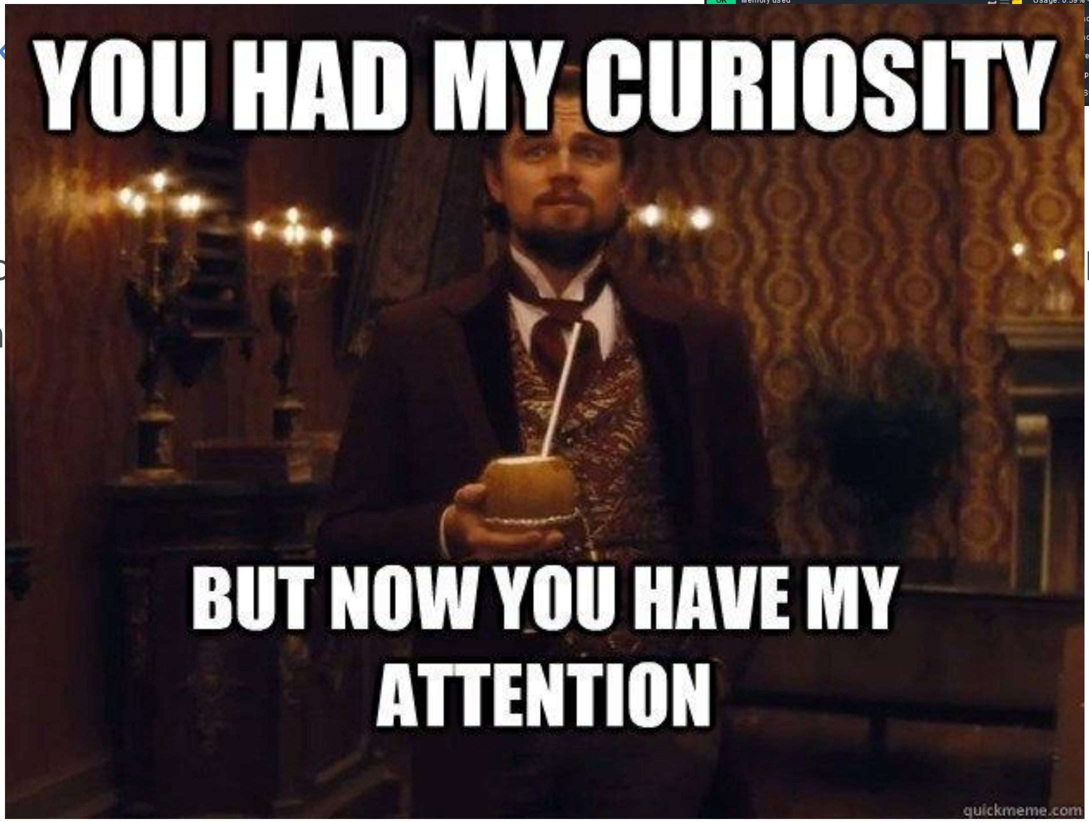
State	Service	Icons	Summary
OK	Check_MK		[special_vsphere] Success, [piggyback] Successfully processed from source '[redacted]', execution time 0.5 sec
OK	Check_MK Discovery		Services: all up to date, Host labels: all up to date
OK	CPU utilization		Total CPU: 0.09%
OK	Datastore IO SUMMARY		Read: 0.00 B/s, Write: 0.00 B/s, Latency: 0 seconds
OK	Disk IO SUMMARY		Read: 0.00 B/s, Write: 5.12 kB/s, Latency: 1 millisecond
OK	Interface vmnic0		[1], (up), MAC: 48:DF:37:C4:D7:10, Speed: 10 GBit/s, In: 3.07 kB/s (<0.01%), Out: 0.00 B/s (0%)
OK	Interface vmnic1		[2], (up), MAC: 48:DF:37:C4:D7:11, Speed: 10 GBit/s, In: 5.12 kB/s (<0.01%), Out: 0.00 B/s (0%)
OK	Interface vmnic2		[3], (up), MAC: 48:DF:37:C4:DF:86, Speed: 10 GBit/s, In: 3.07 kB/s (<0.01%), Out: 18.4 kB/s (<0.01%)
CRIT	Maintenance Mode		System running is in Maintenance mode
OK	Memory used		Usage: 0.59% - 18.2 GiB of 3.00 TiB
			d, 0 disabled, 0 standby, 0 unknown
			d, 0 disabled, 0 standby, 0 unknown
			een, Power state: poweredOn
			pr 7 07:10:55 2023, uptime: 397 days, 2:35:30
			Swap out: 0 B, Swap used: 0 B

VMware ESXi hosts are usually set to maintenance by hand

- Setting the downtime in Check

- Solution
  - check that monitors if the servic lugin output
    - If so => set host into downtim
    - If not => remove downtime

  - Simple installation via MKP and



/ Supported use cases

Further setups customers currently use:

Criteria: Host downtime Target: auto dependencies

- e.g. preventing dependend hosts & services (mainly switch ports) from raising alerts when a host is worked on

- In CMK manually set a downtime on a host
- automatically set downtimes on hosts + services containing the hosts name and also
  child-hosts defined via parent child relations

/ Supported use cases

Further setups customers currently use:

Criteria: Host downtime Target: auto dependencies

- e.g. preventing dependend hosts & services (mainly switch ports) from raising alerts when a host is worked on

- In CMK manually set a downtime on a host
- automatically set downtimes on hosts + services containing the hosts name and also child-hosts defined via parent child relations

Host downtime + setting service-downtimes

- e.g. preventing HTTPS-Checks (on the loadbalancer) to raise alarms when the web server is updated
- In CMK manually set a downtime on a host
- Automatically sets a list of specified services into downtime

Service sets other service into downtime

- E.g. database is in downtime, web application will not respond
- Reacts on plugin-output of a service

/ Future use cases

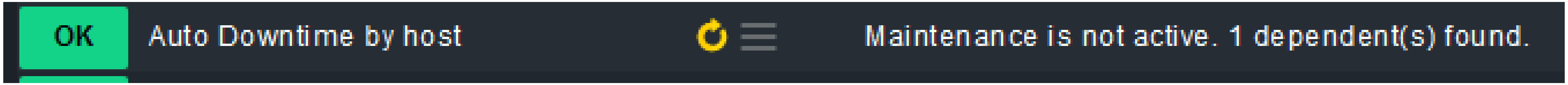
In upcoming release we will also support those use cases

Downtimes based on host and service states

- Host state
- Service state
- Complex state via BI aggregation services,
  e.g. set all NTP-checks in downtime,
  when the BI aggregation indicates that all NTP-servers are down

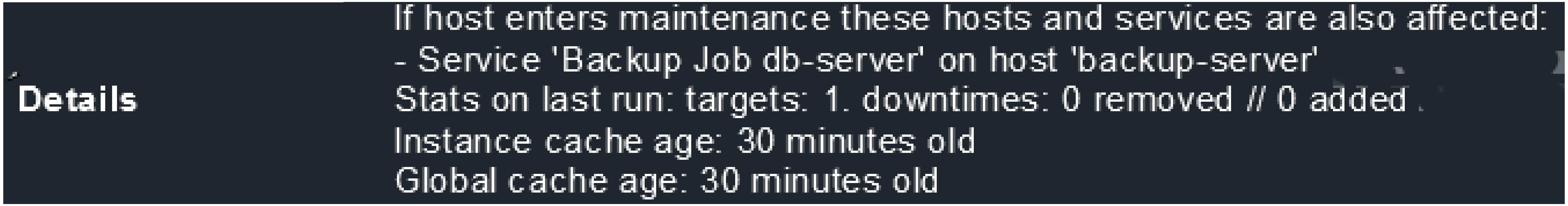
/ Configuration

When the plugin is running it looks somewhat like this:



OK Auto Downtime by host Maintenance is not active. 1 dependent(s) found.

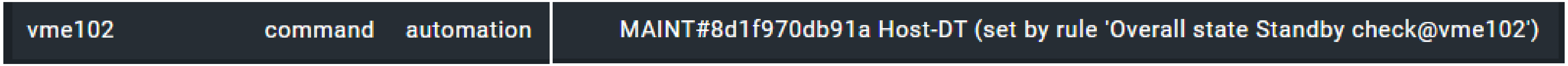
In the details some extra infos can be seen:



Details

```
If host enters maintenance these hosts and services are also affected:
- Service 'Backup Job db-server' on host 'backup-server'
Stats on last run: targets: 1. downtimes: 0 removed // 0 added .
Instance cache age: 30 minutes old
Global cache age: 30 minutes old
```

and created downtimes look like this:



vme102	command	automation	MAINT#8d1f970db91a Host-DT (set by rule 'Overall state Standby check@vme102')

But first we need some configuration:

/ Configuration

The configuration consists of 5 parts

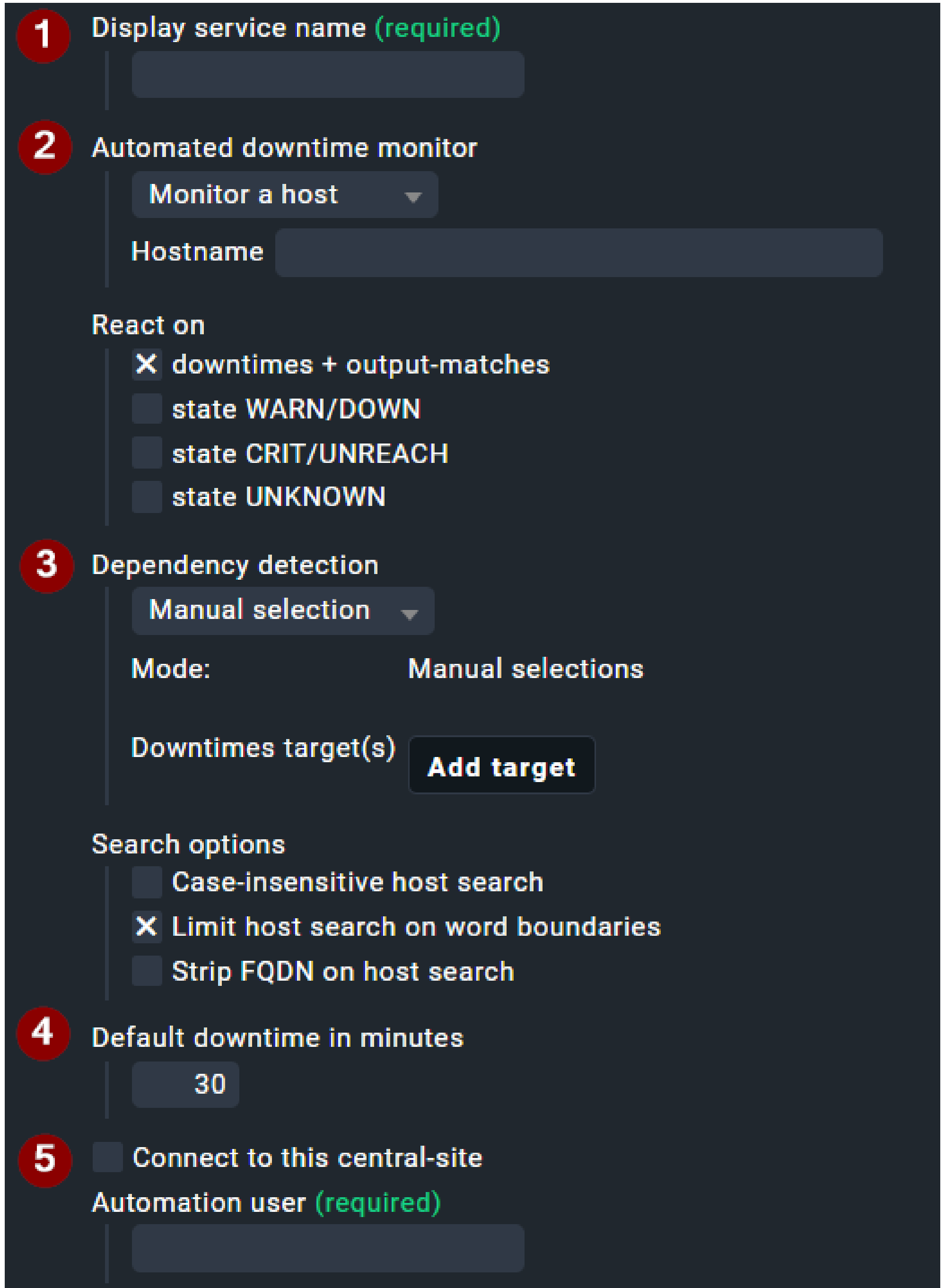
1) Name of the service (choose as you like)

2) Defining criteria, i.e. which host/services to observe

3) Define hosts/services that should be set to downtime

4) How long to set the downtime (how often do we have to renew)

5) Options for connecting to the API and working in distributed setups



1 Display service name (required)

2 Automated downtime monitor

Monitor a host

Hostname

React on

☒ downtimes + output-matches

□ state WARN/DOWN

□ state CRIT/UNREACH

□ state UNKNOWN

3 Dependency detection

Manual selection

Mode: Manual selections

Downtimes target(s)

Search options

□ Case-insensitive host search

☒ Limit host search on word boundaries

□ Strip FQDN on host search

4 Default downtime in minutes

30

5 □ Connect to this central-site

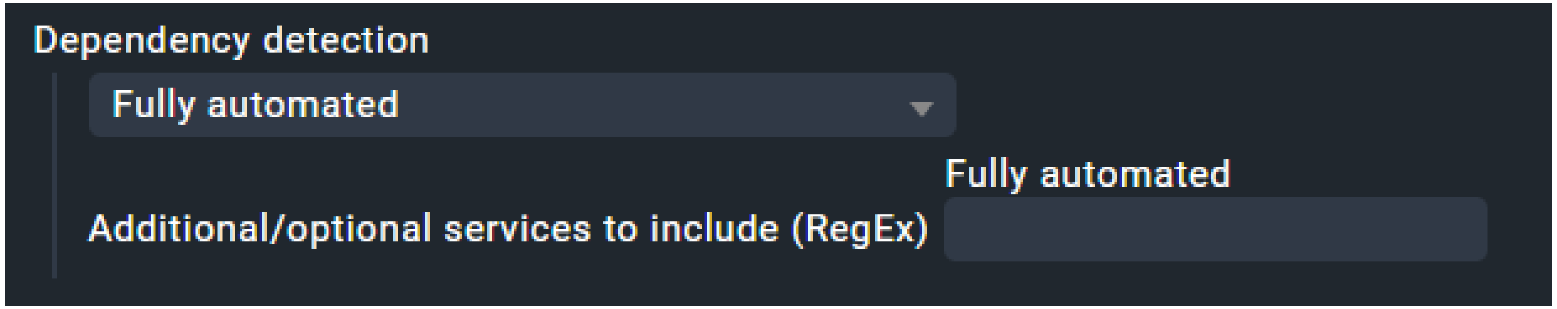
Automation user (required)

/ Dependency detection

Multiple modes for detecting dependencies are available:

Fully automated

- This option searches for hosts and services which contain the name of the host the plugin is bound to
- Also includes children defined via parent-child-relationships
- Optionally allows to include services defined by regex
- Downtimes are applied recursively for dependencies of dependencies
  - i.e. Downtime on SRV-ESX-01 => Downtime on SRV-ESX-01-ILO => Downtime on „Interface SRV-ESX-01-ILO“ on network switches



Dependency detection

Fully automated

Fully automated

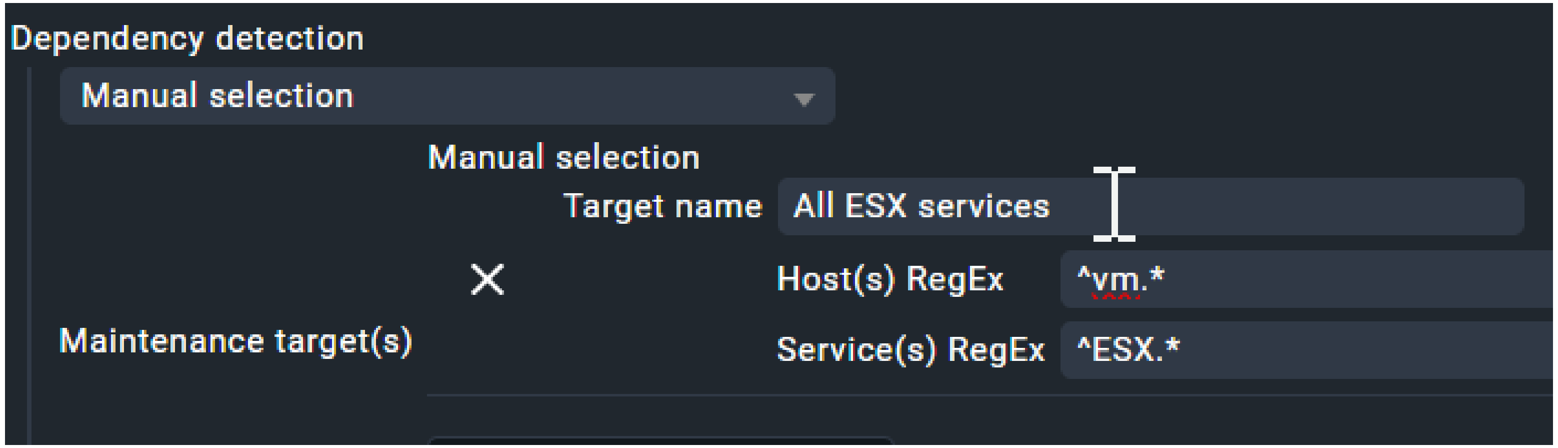
Additional/optional services to include (RegEx)

/ Dependency detection

Multiple modes for detecting dependencies are available:

Manual selections

- Define dependencies for hosts and/or services via regex
- Also useful for dependencies between databases and applications



Dependency detection

Manual selection

Manual selection

Target name: All ESX services

Maintenance target(s)

Host(s) RegEx: ^vm.*

Service(s) RegEx: ^ESX.*

/ Thorny technical challenges

- Central and remote
  - Livestatus doesn‘t cut it, lets migrate to the Rest-API
- Uh, oh, Performance went down
  - Rest-API is much more powerful, but also much slower than Livestatus
  - Now we have mix of
    - Livestatus (where sufficient)
    - Rest-API (do as much as possible in one call)
    - *Caching*

“There are only two hard things in Computer Science: cache invalidation and naming things.”

– unknown number of smart people

/ How to use / contribute

- „Automated Downtimes“ on exchange.checkmk.com

- https://github.com/svalabs/check_mk_automated_downtimes

=> Code contributions welcome <=

- FAQ:

1. Is it free? - Of course
2. Does it work on CRE? – It sure does.
3. Does it work with distributed sites? – Yes ☺.