

CHECK_MK BEGINNER GUIDE

Author: Marco Reale

Version: 1.0 – October 2016

Disclaimer:

Please consider this guide just as a bunch of notes and NOT as a professionally written document. My intention is to give something back to the community and I haven't any relation with the company behind Check_MK.

I assume no responsibility for the accuracy, completeness, or usefulness of any information or for damages resulting from the procedures provided. Furthermore, this documentation is supplied "as is" without guarantee or warranty, expressed or implied, including without limitation, any warranty of fitness for a specific purpose.

I sincerely thank the Check_MK mailing list users because without their help I would not have been able to write this guide.

Special thanks to:

Jolyon Brown: Help in translation

Mike Hulsman: Solution for Monitoring Microsoft Terminal Services

Brian Binder: Solution for Monitoring Microsoft Windows Event Log

Andreas Döhler: Explanation about Folders,Tags & Hostgroup

Evy Bongers: Explanation about Folders,Tags & Hostgroup

Apologies to anyone I've missed.

Summary

INTRODUCTION	4
Check_MK Setup.....	7
WATO – The Graphical User Interface.....	9
Views - Pane	10
Configuration – Pane	12
Users	13
Apply Changes	15
Managing agents	16
Agent Installation on Linux	16
Agent Installation on Windows	17
Devices Management.....	18
Folders	18
Tags.....	21
Hostgroup.....	24
Linux Devices	25
File System Monitoring.....	30
Linux Process Monitoring	34
Log Files	37
Windows Devices.....	40
Windows Event Viewer.....	40
Windows Services.....	44
Microsoft SQL Server	49
Microsoft Terminal Services	51
Network Devices.....	53
Managing Thresholds	58
Hardware & Software Inventory	62
Using custom plugins.....	69
Local Checks.....	70
MRPE – Nagios Plugins	71
MKP plugins	71
Monitor Apache Webserver	75
Monitor Mysql Server.....	77
Monitor Physical Hardware	81
Monitor Vmware	84

Add vSphere Virtual Center	84
Add ESXi host managed by Vcenter.....	88
Add standalone ESXi hosts	90
Virtual Machines additional checks.....	92
Managing SNMP Traps	94
Managing Notifications	102
Contact group	104
Analysis.....	106
Check_MK Update.....	107
Package installation.....	107
Switching to the new version	108
Conclusion	110

INTRODUCTION

Every system administrator should know the current state of infrastructure they are responsible for. There is nothing worse than realising much too late that a service is down or, even worse, to have users notify you of problems you hadn't yet noticed.

A good monitoring solution provides automated reporting of errors and malfunctions allowing immediate intervention. In addition, this automation frees IT personnel from having to keep constant watch over all infrastructure - servers, desktop computers, applications, traffic, etc. so they can use their time for other tasks.

Unfortunately, not all companies understand the importance of such monitoring until there are serious problems that affect their business. In my career I've had situations where managers asked me for the reasons behind serious and continuous performance problems - and why we were not able to quickly identify them. I've always replied that without a good monitoring solution, we were blind. I don't want to claim that monitoring itself prevents any kind of outage or can ensure 100% uptime, because clearly there are other important factors to consider (even organizational aspects). But believe me, it is extremely important and helps prevent many potential outages.

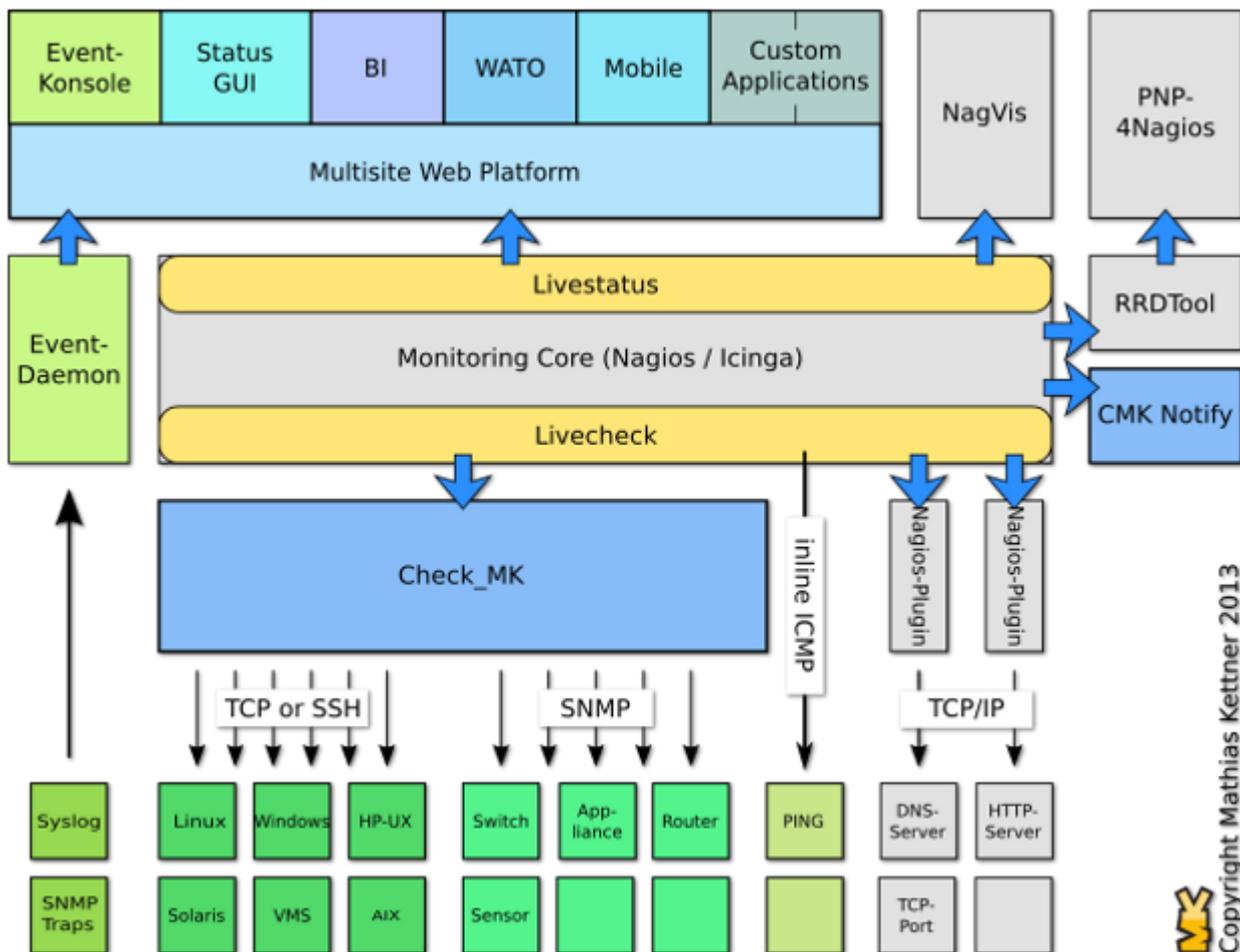
In my opinion, an Enterprise monitoring solution should provide the following features out of the box: scalability, multi-tenancy, granular access to hosts and services, customizable dashboards, notifications, good performance graphs, automatic inventory of services being monitored, certified plugins for all enterprise class hardware/software, understanding of parent & child relationships, flexibility in creating custom checks and, last but not least, should be easy to install, use and maintain.

The market has plenty of solutions (both free and commercial), but most of them are difficult and time-consuming with a steep learning curve and/or quite expensive. Over the last few years I've worked with many products and, even though I must admit that I had some nice results, I never really found something that completely satisfied me. I always find them lacking something or other.

One day though I came across Check_MK. A solution that, in a nutshell, claimed to make Nagios much easier and more powerful to use.

As the official site states (http://mathias-kettner.com/check_mk.html), Check_MK is a comprehensive IT monitoring solution in the tradition of Nagios. The main developer for the product is Mathias Kettner and the company he has formed around it is located in Munich, Germany.

The following diagram (taken from the official website) shows how with the help of Check_MK and Nagios, a complete monitoring solution can be assembled. The coloured boxes represent the components of the Check_MK-Project.



Check_MK is available as a 100% open source package (known as the “Raw Edition (CRE)”) and as a professionally supported “Enterprise Edition (CEE)” that comes with a lot of additional features such as:

- Agent bakery (packaging of individual monitoring agents)
- High performance and low latency via Check_MK Micro Core
- Reporting: Creation of individualized reports in PDF format
- Improved performance on distributed setup
- Auto expiration of acknowledged alarms
- Recurring planned-downtimes

You also can deploy the CEE by using or a hardware or software appliance.

For a complete comparison, please have a look at http://mathias-kettner.com/check_mk_introduction.html

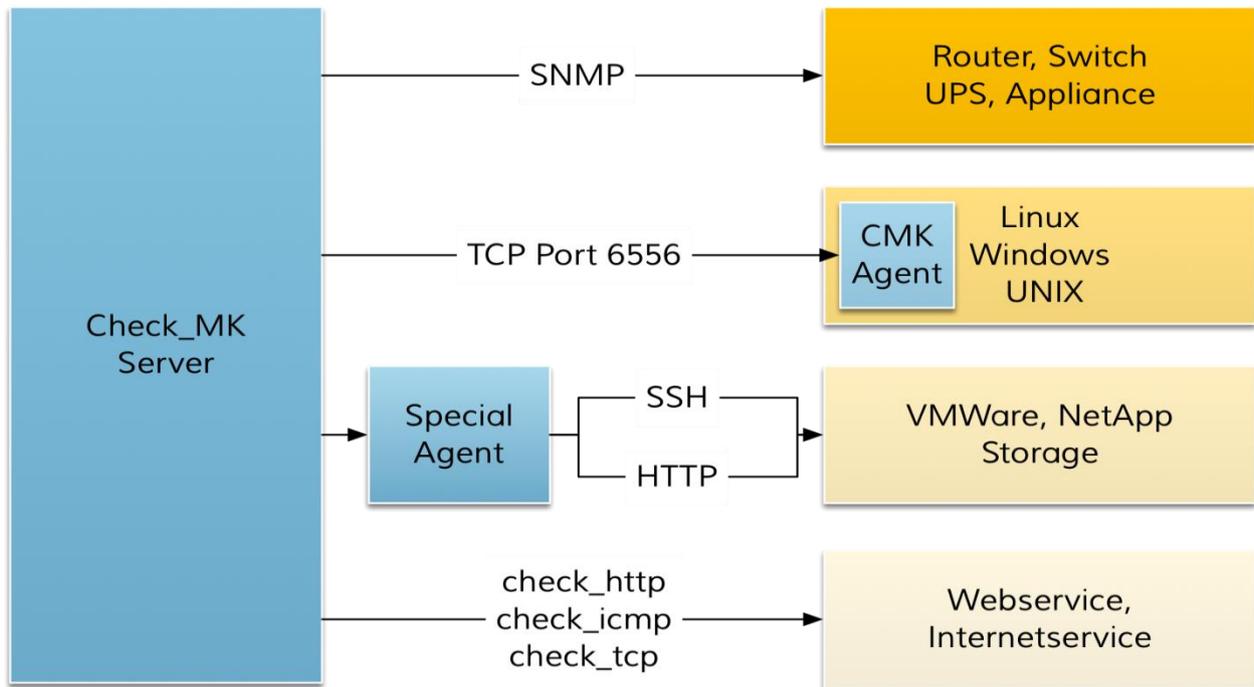
In my opinion, these are the main Check_MK key features:

- Fully compatible with Nagios
- Excellent performance even on large environments
- Scale-out/Distributed monitoring with centralized management
- Graphical User Interface (WATO)
- Shorter learning curve compared to other Nagios solutions

- Hundreds of certified plugins and supported devices
- Agent based monitoring for OSs and SNMP for network devices

The following table shows the four different ways that Check_MK can access services to be monitored:

(ref: https://mathias-kettner.com/cms_wato_monitoringagents.html)



Check_MK Setup

In this guide I'm going to show how to set up and get started with Check_MK Raw Edition on a Centos 7.2 virtual machine. I always suggest installing the latest stable version that, at the time of writing (October 2016), is 1.2.8p13.

Download Check_MK

On 2015-05-07 we have changed the way Check_MK is packaged and made available to you for download. Details can be found [here](#). Please also note our [article about the version numbers of Check_MK](#).

Check_MK Raw Edition (CRE)

The *Check_MK Raw Edition* is a full-blown IT monitoring solution - available under the terms of the GNU GPL version 2 and a couple of other open source licenses. You can use, modify and pass along the CRE **for free** as long as you comply with these licenses.

Branch	Newest Release	Change log
1.2.6 (old-stable)	1.2.6p16	Changes
1.2.8 (stable)	1.2.8p13	Changes

Check_MK Enterprise Edition (CEE)

The *Check_MK Enterprise Edition* is based on the Raw Edition but contains many additional enterprise-class features and also entitles you to get professional manufacturer support. In order to download it you need a valid [subscription](#).

You also can deploy the CEE by using or [hardware or software appliance](#).

Branch	Newest Release	Daily Build	Change log
1.2.6 (old-stable)	1.2.6p16	not available	Changes
1.4.0i1 (innovation)	1.4.0i1	not available	Changes

Requirements for TEST environment:

- Centos 7 64bit with 2vCPU, 4GB RAM, 30GB HD
- Working internet connection
- EPEL repository enabled
- SELinux disabled or properly configured

Please note that the above list is just for a TEST environment; to properly size a production server, there are many variables to consider such as the number of monitored services and the hardware you are going to place Check_MK on (carefully distinguishing between physical and virtual hardware).

There are some handy notes about sizing considerations at the following link: https://mathias-kettner.de/checkmk_checkmk_benchmarks.html

Step by step setup:

- 1) Install Centos 7.2 64 bit
- 2) Check internet connection and enable EPEL repository
- 3) Download the last version of Check_MK and place it in `/tmp/setup_checkmk/`
(Please note that in this guide I started with 1.2.8p11 - just because later I'll show how to update to 1.2.8p13. If this is the first time you are going to setup Check_mk, download the latest version!)
- 4) `cd /tmp/setup_checkmk/`
- 5) `yum localinstall -ivh check-mk-raw-1.2.8p11-el7-36.x86_64.rpm`
- 6) Create your first OMD site. You just have to choose a site name, like prod or test or whatever you like (in this example I have chosen "mysite"). Then, as root user, you simply type:

```
omd create mysite
```

- 7) Using a browser, point to <http://ip/mysite>
- 8) Login to using default credentials:

```
username: omdadmin  
password: omd
```

That's all! As you can see the setup is really easy. I'd even describe it as being "windows like" - but without the need to reboot 😊

WATO – The Graphical User Interface

WATO is a nice and powerful GUI through which it is possible to manage hosts and services being monitored with Check_MK. Just please note that by using WATO, you can avoid the use of the command line for many tasks but not all of them. Why? The best explanation is provided by Mathias on his website from which I took the following screenshot:

(ref: https://mathias-kettner.de/checkmk_wato.html)

WATO - Check_MK's Web Administration Tool

Dieser Artikel wird nicht mehr gepflegt und ist unter Umständen nicht mehr gültig!

1. Introduction

WATO is Check_MK's new graphical administration tool. It is a web based user interface for managing hosts and services to be monitored with Check_MK. However, WATO is no tool for configuring all aspects of Check_MK and Nagios. Why?

The basic idea behind WATO is that - when it comes to monitoring - usually one or few persons are responsible for setting up and maintaining the actual monitoring server. They spend a lot of time with the system and its internals and usually have no difficulties with editing text based configuration files - or even prefer them over a GUI.

The "users" of the monitoring, however, just need their systems to be monitored but do not have the time to learn how to write valid configuration files. Nor is it their job. So whenever they need any modification in the monitoring - for example if a new server has been set up, a switch configuration has changed or a database instance has been removed - they ask the monitoring team to adapt their configuration accordingly. Those changes make up a substantial part of the daily workload of the monitoring team.

WATO allows you to move these daily tasks to the users by providing them a GUI for managing their hosts and services themselves. The monitoring team can spend their time with their actual work - tuning the system, implementing new checks, configuring general rules, and so on.

1.1. Why not using NagiosQL, LConf, NConf, Centreon or other GUIs?

The first simple reason is: They do not support Check_MK and probably never will. Another reason is: WATO directly supports Check_MK's inventory mechanism and thus an auto-detection of services. And last but not least, WATO takes into account that the different needs of monitorings admins and users should be reflected in the GUI.

That said, I found that I could do most tasks using just the GUI. Moreover every new version seems to add some WATO module so which brings into the GUI some tasks which previously had to be performed manually.

This is the main WATO window that provides a global overview of Host and Services statistics as well as a list of recent events.

Host Statistics

Up	11
Down	0
Unreachable	0
In Downtime	0
Total	11

Service Statistics

OK	644
In Downtime	0
On Down host	0
Warning	2
Unknown	0
Critical	4
Total	650

Service Problems (unhandled)

State	Host	Service	Icons	Status detail
CRIT	centos7tst1	Interface 4		CRIT - [eno50336512] (down) CRIT MAC: 00:0c:29:dc:9a:dc, assuming 10.00 Gbit/s
CRIT	localhost	Postfix Queue		CRIT - deferred queue length is 120 (Levels at 10/20) CRIT , active queue

On the left side there are two main sections: *Views* and *Configuration*

Views - Pane

In this section there are many views of different components like these:

Service problems

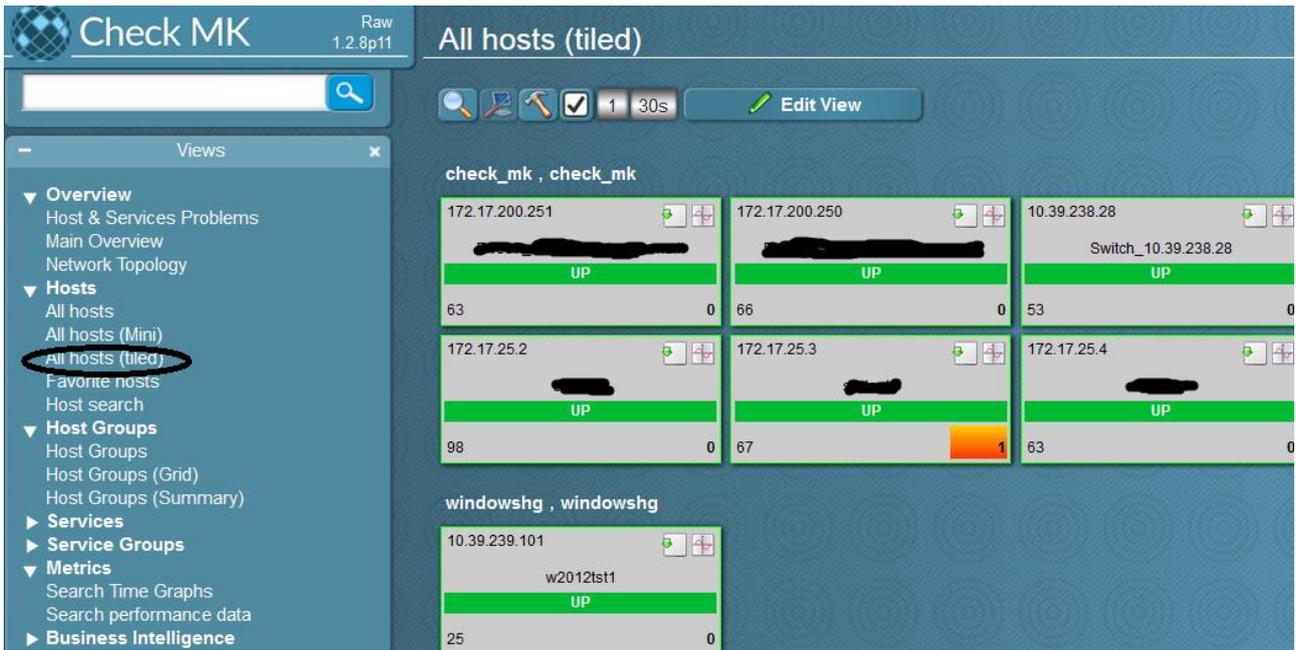
1 30s Edit View Availability

CRIT

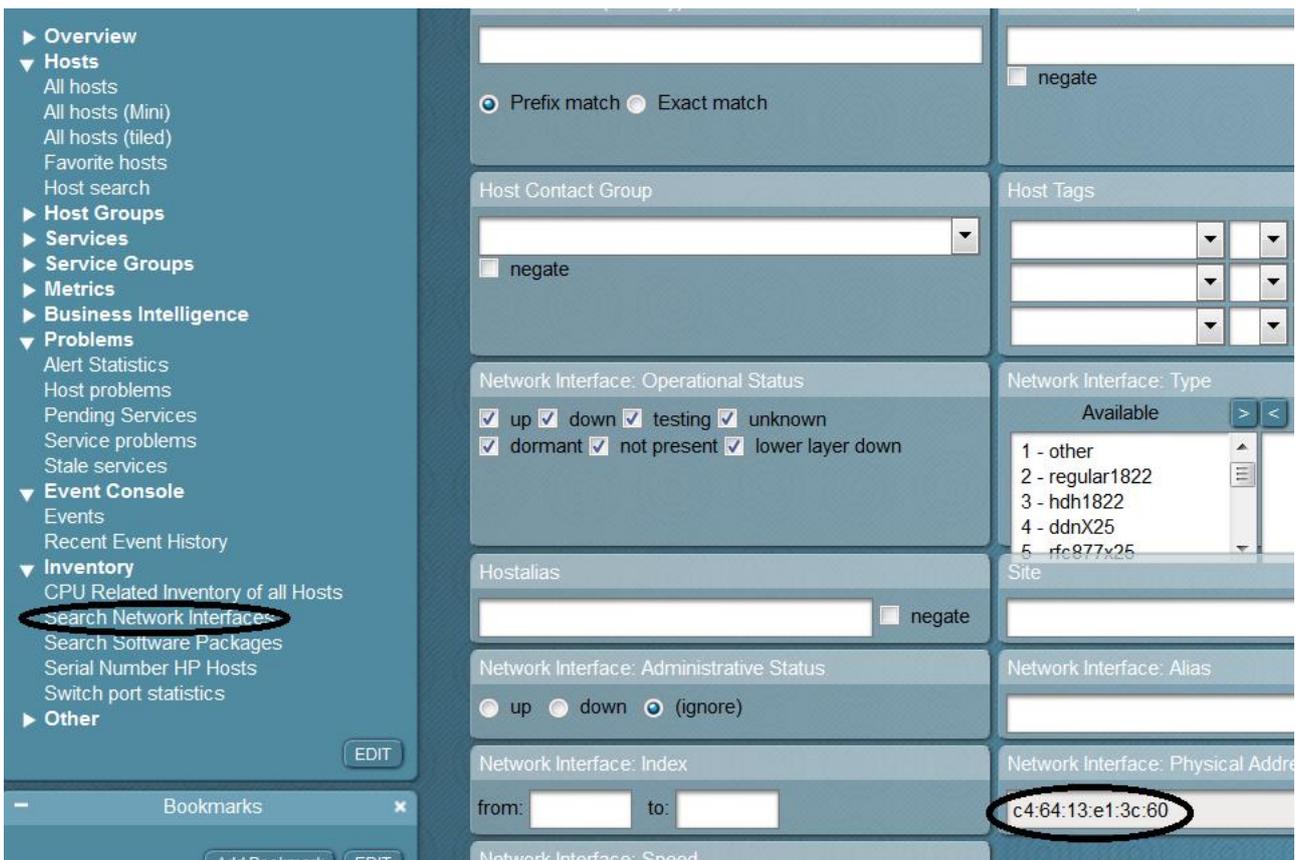
State	Host	Service	Icons	Status detail
CRIT	centos7tst1	Interface 4		CRIT - [enc
CRIT	localhost	Postfix Queue		CRIT - defe
CRIT	localhost	[REDACTED]_TEST_Filecount_/var/log		CRIT - CRIT
CRIT	centos7tst1	[REDACTED]_TEST_Filecount_/var/log		CRIT - CRIT

WARN

State	Host	Service	Icons	Status detail
WARN	localhost	Filesystem /opt		WARN - 80 MB / 24 hours
WARN	localhost	Filesystem /home		WARN - 80 / 24 hours



It's also possible to do some useful searches. For example - did you ever try to find the switch port of a specific MAC or IP address? With WATO, this can be done with just a couple of clicks.



Host	Index	Description	Alias	Status	Admin	Used	Speed	Last Change	Physical Address (MAC)
[REDACTED]	13	GigabitEthernet1/13	CrData2_iLo (172.17.4.4)	up	up	used	100 Mbit/s	-147 days ago	C4:64:13:E1:3C:60

Configuration - Pane

This menu is divided into many sections but by clicking on *Main Menu* you can access all of them from a single point

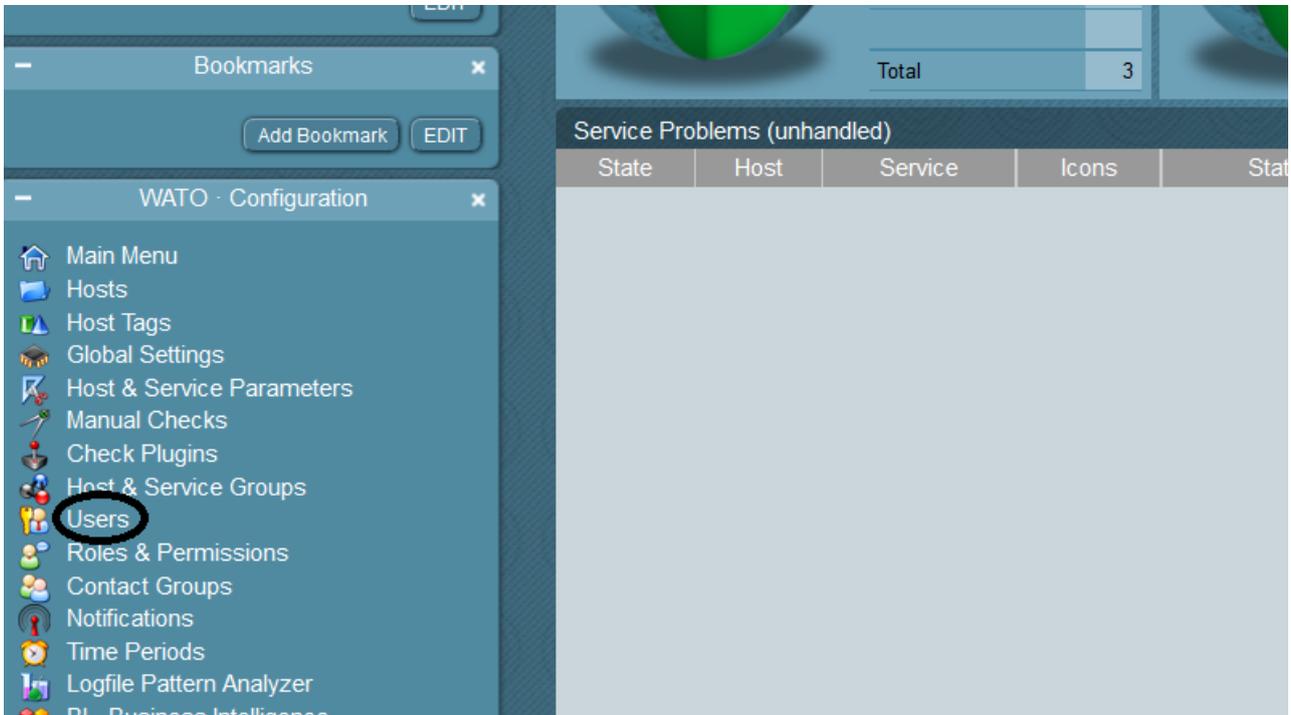
The screenshot shows the WATO - Check_MK's Web Administration Tool interface. On the left, a navigation pane lists various sections: Hosts, Host Groups, Services, Service Groups, Metrics, Business Intelligence, Problems, Event Console, Inventory, and Other. Below this is a Bookmarks section and a WATO Configuration section where 'Main Menu' is highlighted with a red circle. The main content area features a 'No Changes' notification and a grid of configuration modules, each with an icon and a brief description:

- Hosts**: Manage monitored hosts and services and the hosts' folder structure.
- Host Tags**: Tags classify hosts and are the fundament of configuration of hosts and services.
- Host & Service Parameters**: Check parameters and other configuration variables on hosts and services.
- Manual Checks**: Configure fixed checks without using service discovery.
- Host & Service Groups**: Organize your hosts and services in groups independent of the tree structure.
- Users**: Manage users of the monitoring system.
- Contact Groups**: Contact groups are used to assign persons to hosts and services.
- Notifications**: Rules for the notification of contacts about host and service problems.
- Logfile Pattern Analyzer**: Analyze logfile pattern rules and validate logfile patterns against custom text.
- BI - Business Intelligence**: Configuration of Check_MK's Business Intelligence component.
- Backup & Restore**: [Icon]
- Custom Icons**: [Icon]

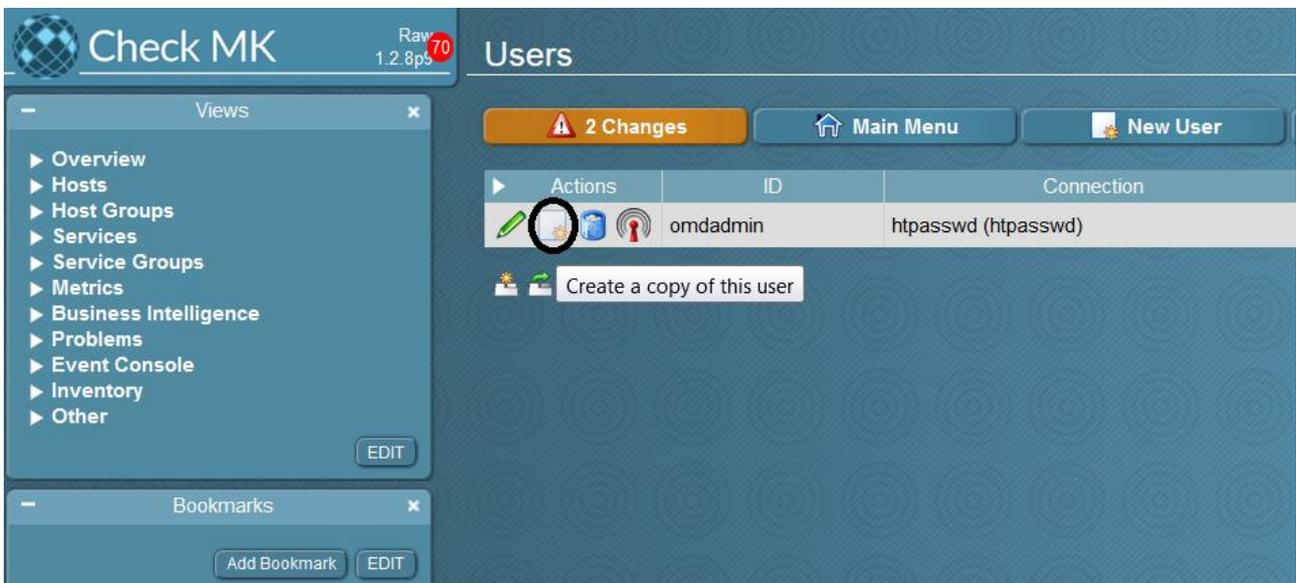
I'm not going to describe each sub-menu but will cover a few of them in the following section.

Users

One of the first tasks that should be performed after the setup is the creation of users. Everyone who is going to use check_MK should have their own custom credentials. This is done using WATO: *Users*



It's possible to create a new user by cloning an existing one:



Check MK Raw 1.2.8pt 70

← All Users

Views

- Overview
- Hosts
- Host Groups
- Services
- Service Groups
- Metrics
- Business Intelligence
- Problems
- Event Console
- Inventory
- Other

Bookmarks

WATO Configuration

- Main Menu
- Hosts
- Host Tags
- Global Settings
- Host & Service Parameters
- Manual Checks
- Check Plugins
- Host & Service Groups
- Users
- Roles & Permissions
- Contact Groups
- Notifications
- Time Periods
- Logfile Pattern Analyzer
- BI - Business Intelligence
- Distributed Monitoring
- Backup & Restore
- Custom Icons
- Monitoring Agents
- Event Console

2 changes

Master Control

© Mathias Kettner

Save

Identity

Username: [redacted]
Full name: [redacted]
Email address: [redacted]
Pager address: [redacted]

Security

Authentication

- Normal user login with password
- password: [redacted]
- repeat: [redacted] (optional)
- Enforce change: Change password at next login or access
- Automation secret for machine accounts
- [redacted]

Disable password: disable the login to this account

Roles

- Administrator
- Guest user
- Normal monitoring user

Contact Groups

- Everything

Personal Settings

Language: Default: English

Visibility of Hosts/Services (Webservice): Export only hosts and services the user is a contact for

Visibility of Hosts/Services: Only show hosts and services the user is a contact for

Disable Notifications: Temporarily disable all notifications!

Start-URL to display in main frame: dashboard.py

Apply Changes

Whenever changes are made in the configuration, we need to restart check_mk by clicking on the *Changes* button followed by *Activate Changes*



The screenshot shows the Check MK interface with the 'Users' page selected. A yellow notification bar at the top indicates '3 Changes'. The table below lists two users: 'realem' and 'omdadmin', both using 'htpasswd (htpasswd)' for authentication.

Actions	ID	Connection	Auth
  	realem	htpasswd (htpasswd)	Password
  	omdadmin	htpasswd (htpasswd)	Password



The screenshot shows the 'Pending changes to activate' page. A yellow notification bar at the top indicates 'Activate Changes!'. The table below lists one pending change: 'skytest' with status 'online'.

Actions	ID	Alias	Status
	skytest	Local site skytest	online

Changes that are not yet activated

Host	Time	User	Description
localhost	2016-08-19 14:42:25	omdadmin	Created new host localhost.

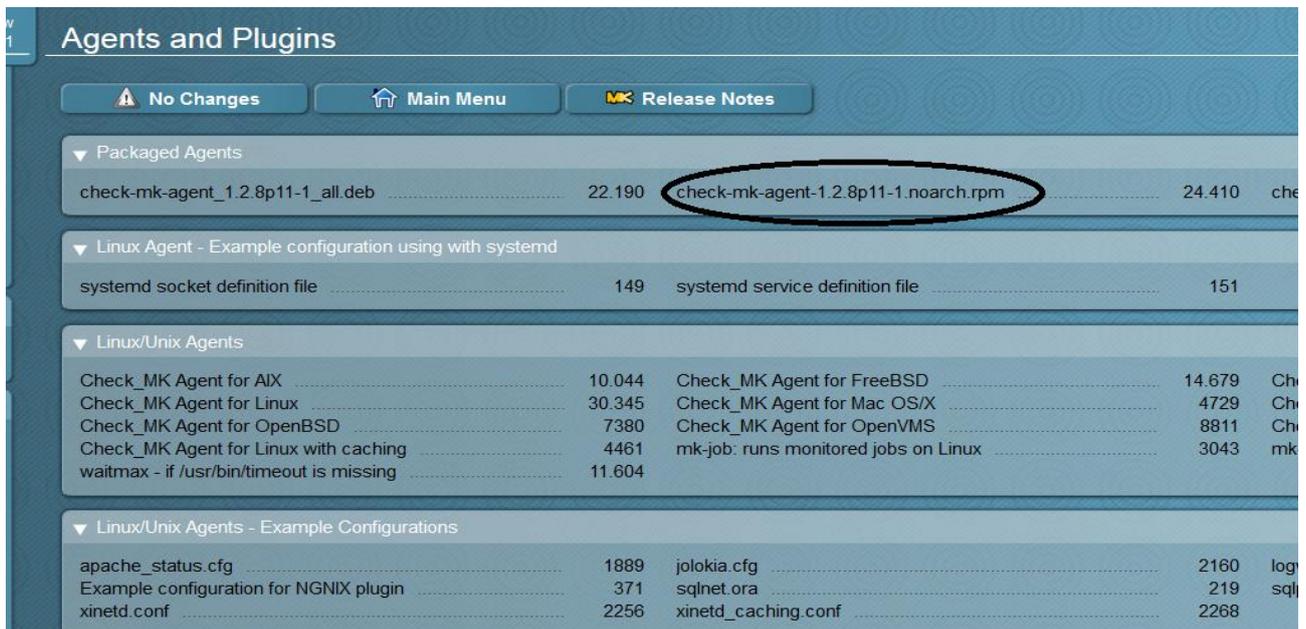
Managing agents

Agents for many operating systems are available in *WATO, Monitoring Agents*. There are *rpm* and *deb* packages but a manual installation is possible too. The Enterprise version provides a feature called *agent bakery* that allows the creation of custom packages; combined with the *Automatic Agent Update feature* available since version 1.2.8, the effort needed to update agents is extremely reduced, especially in large environments. Running agents will listen on port TCP 6556.

Agent Installation on Linux

We are going to install the agent on localhost (where *check_mk* is running) using *rpm*. Installing the Agent via RPM or DEB is very easy. All you have to do is to make sure *xinetd* is installed first and then install the package.

Click on *WATO, Monitoring Agents* and select *check_mk-agent rpm*



You can download or copy it manually:

```
[root@checkmktst1 linux]# pwd
/tmp/setup_checkmk/agents/linux
[root@checkmktst1 linux]# wget http://localhost/mysite/check_mk/agents/check-mk-agent-1.2.8p11-1.noarch.rpm
--2016-08-19 14:37:13-- http://localhost/mysite/check_mk/agents/check-mk-agent-1.2.8p11-1.noarch.rpm
Resolving localhost (localhost)... ::1, 127.0.0.1
Connecting to localhost (localhost)|::1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24407 (24K) [application/x-rpm]
Saving to: 'check-mk-agent-1.2.8p11-1.noarch.rpm.1'

100%[=====] 24,407  --.-K/s  in 0s

2016-08-19 14:37:13 (473 MB/s) - 'check-mk-agent-1.2.8p11-1.noarch.rpm.1' saved [24407/24407]
```

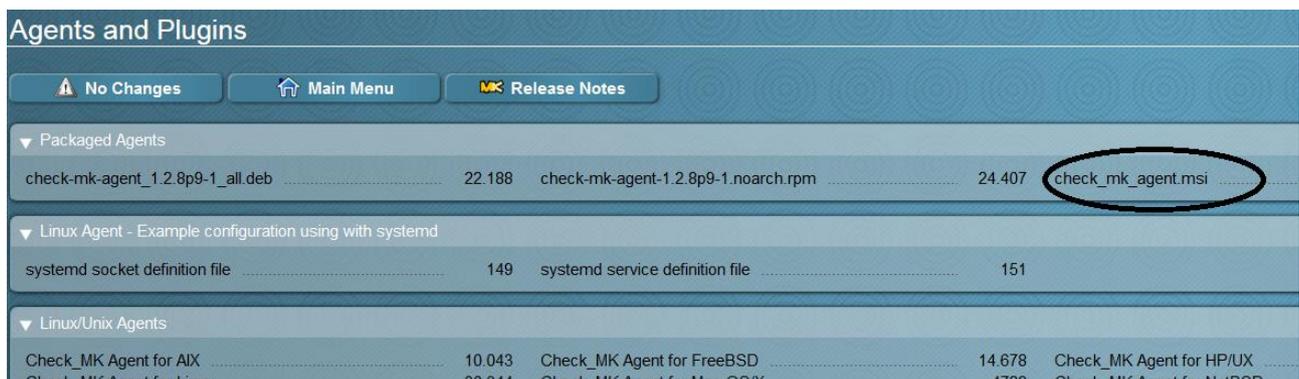
```
[root@checkmktst1 linux]# rpm -ivh check-mk-agent-1.2.8p11-1.noarch.rpm
Preparing...      ##### [100%]
Updating / installing...
 1:check-mk-agent-1.2.8p11-1  ##### [100%]
Reloading xinetd...
Redirecting to /bin/systemctl reload xinetd.service
```

The xinetd file should look like this:

```
/etc/xinetd.d/check_mk
service check_mk
{
    type                = UNLISTED
    port                = 6556
    socket_type         = stream
    protocol            = tcp
    wait                = no
    user                = root
    server              = /usr/bin/check_mk_agent
# configure the IP address(es) of your Nagios server here:
#    only_from          = 127.0.0.1 10.0.20.1 10.0.20.2
    disable             = no
}
```

Agent Installation on Windows

Download “check_mk_agent.msi” and install it on all servers that you need to monitor.



Restart the agent using:

```
net stop check_mk_agent && net start check_mk_agent
```

Devices Management

Managing devices doesn't just mean adding or removing devices but also applying checks, creating rules, thresholds and (last but not least) organizing them. In check_MK, this is achieved using *Folders, Tags and Hostgroup*.

Basically, these are just different ways to achieve a common purpose: organizing devices so that configuring them is easy even with a large number of hosts.

Managing hundreds or even thousands of devices could be very difficult without a proper classification that allows rules to be applied to groups of objects instead of single entities.

The best analogy that comes to my mind is Microsoft Active Directory that allows policies to be created for the entire domain (the root), sites or even just organizational units.

The question is, which one of them should be used?

Well, I opened a thread about this topic on the Check_MK English mailing list and I received some good advice from expert users which, most of the time, suggest using both of them.

Let's say you have 1000 devices in one site; in this case you can create folders for each category such as Windows servers, Linux, UPS, Storage etc.

If you have 20 sites and 500 hosts you may want to create one folder for each site e.g. London, New York, Paris and so on and then create subfolders for Linux, Windows, UPS etc.

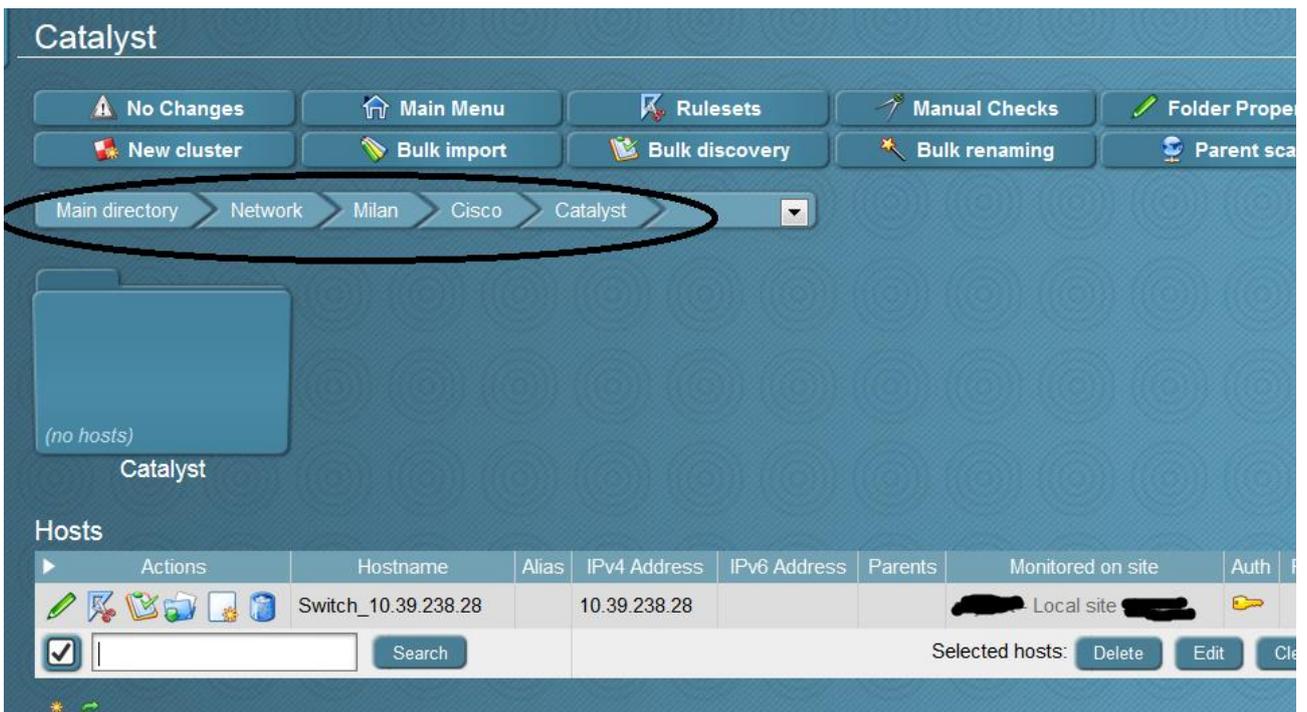
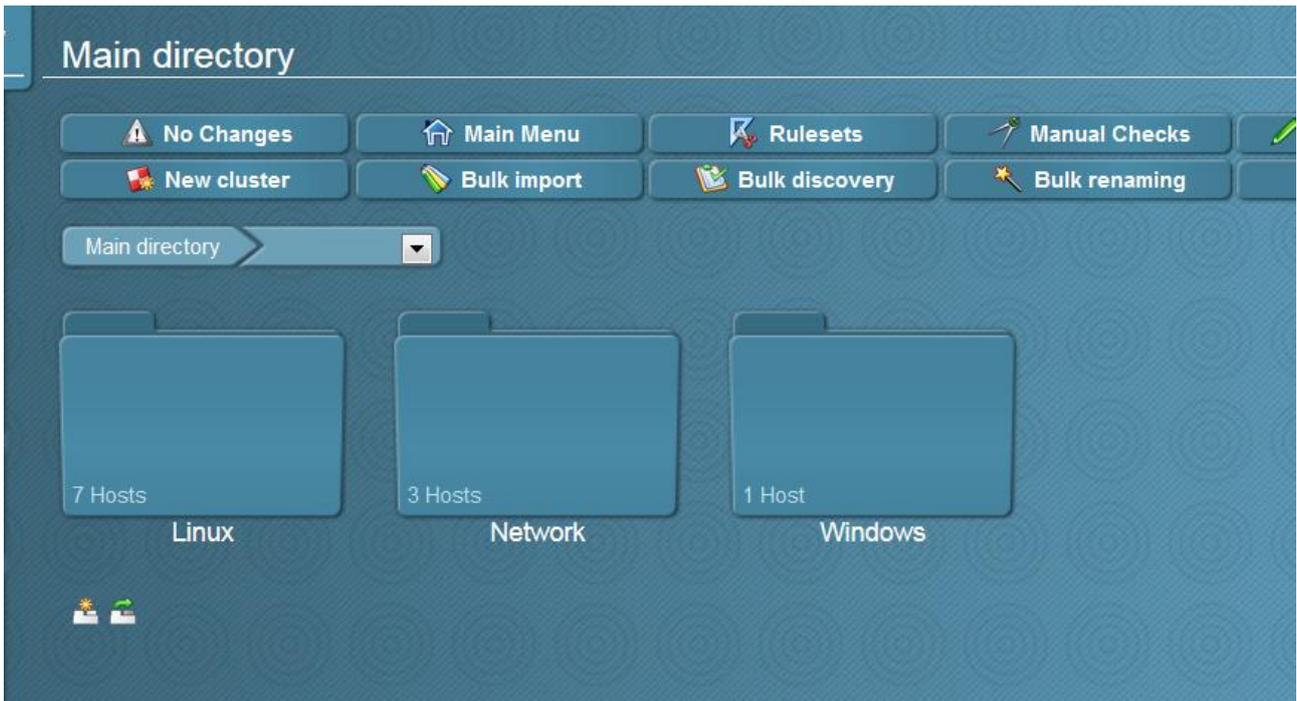
But another option would be to create host tags.

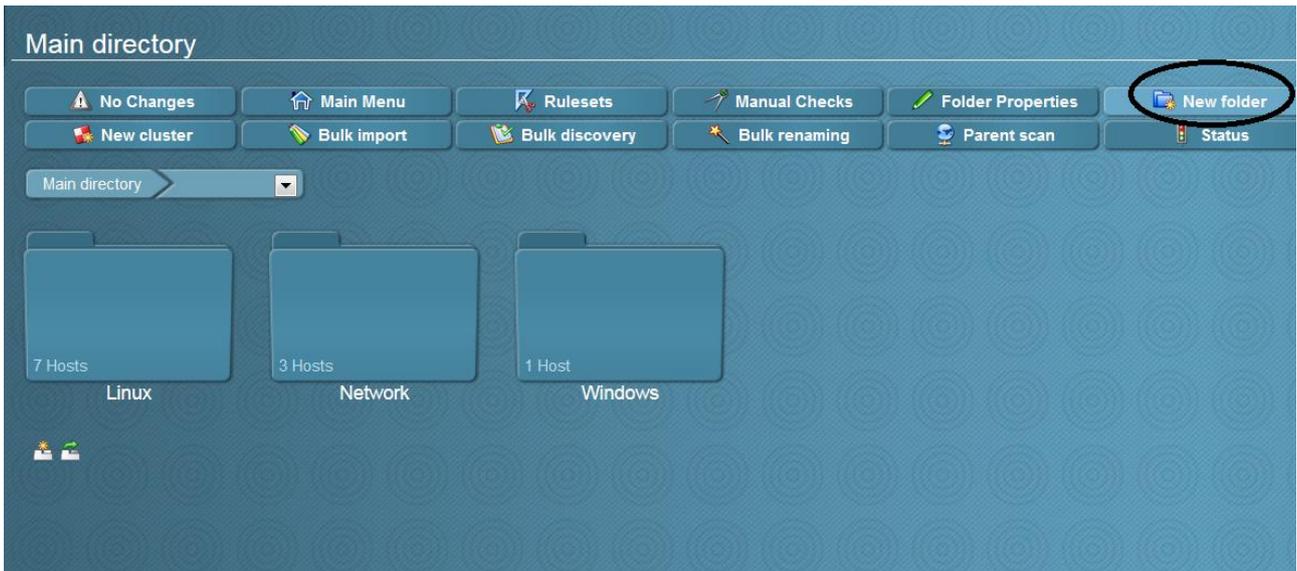
Also hostgroups can make searching for hosts easier. You can create directories per project and assign host tags accordingly. Using these host tags, you can assign hosts to project specific hostgroups, which makes it possible to search for all hosts in a specific project. Also, you can allow customers or users within your company access to their specific projects by making them contacts for their project specific host groups.

Folders

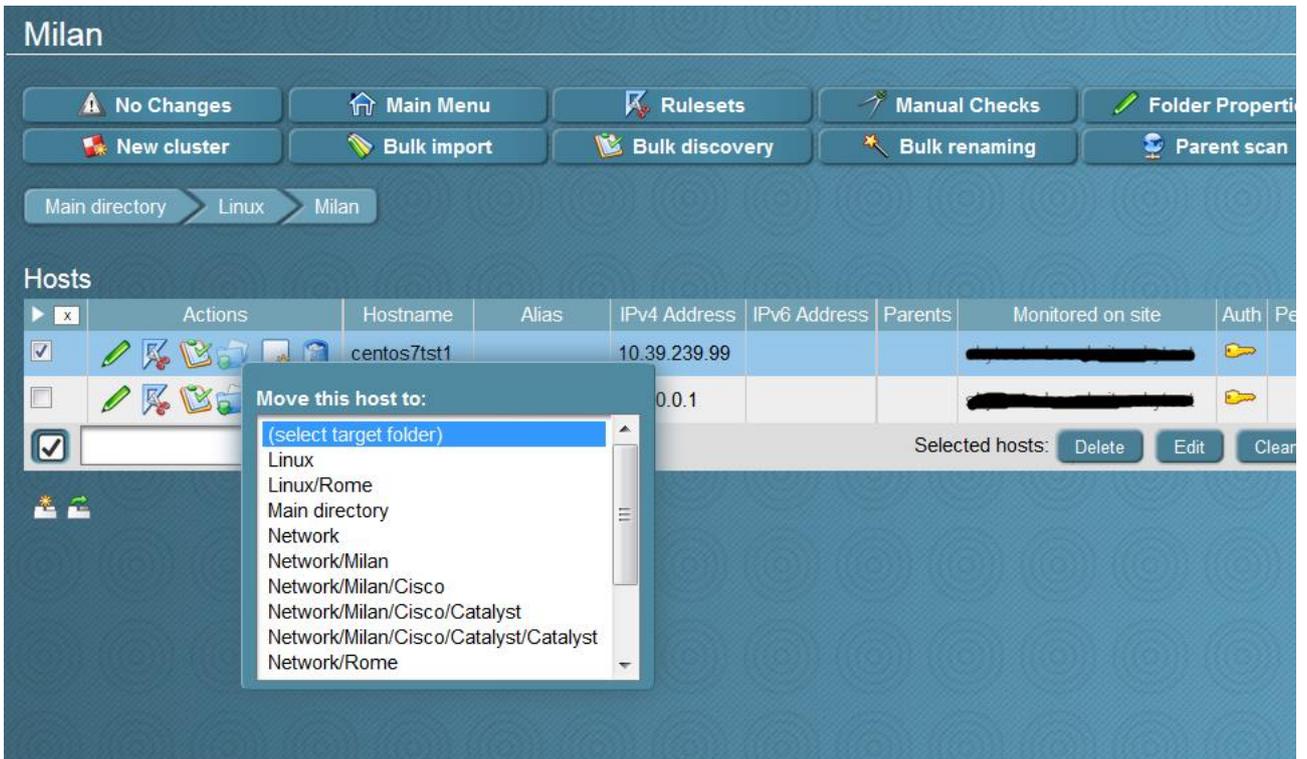
By default there is just the *Main directory* where devices are placed if no specific folder is chosen for them. Click on *WATO, Hosts, New folder* to add more folders.

In this picture there are some folders within the Main directory and each one of them contains other sub folders and devices accordingly to their topology



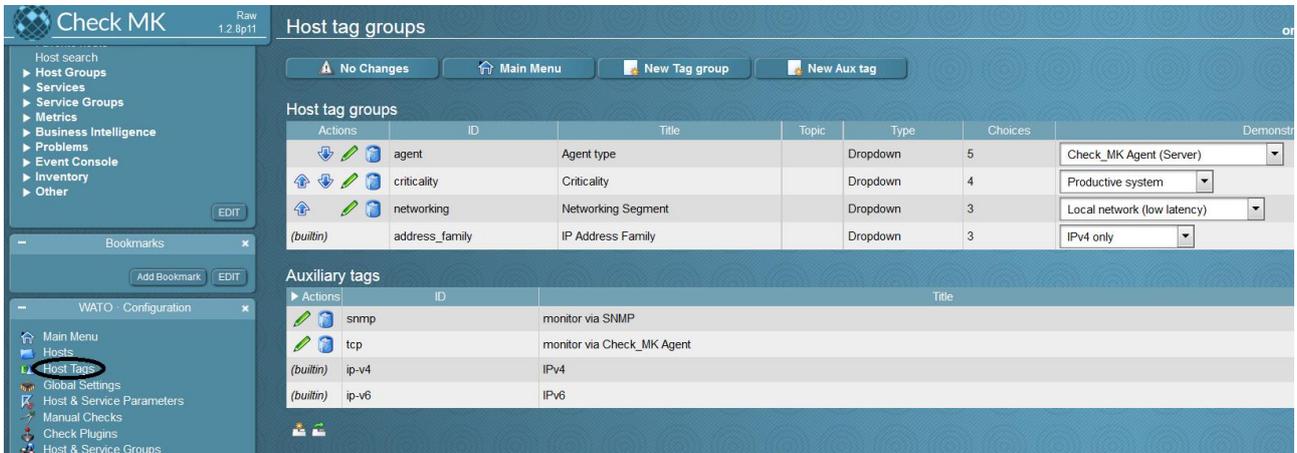


To place a device in folders, select the device and click on the *folder* icon

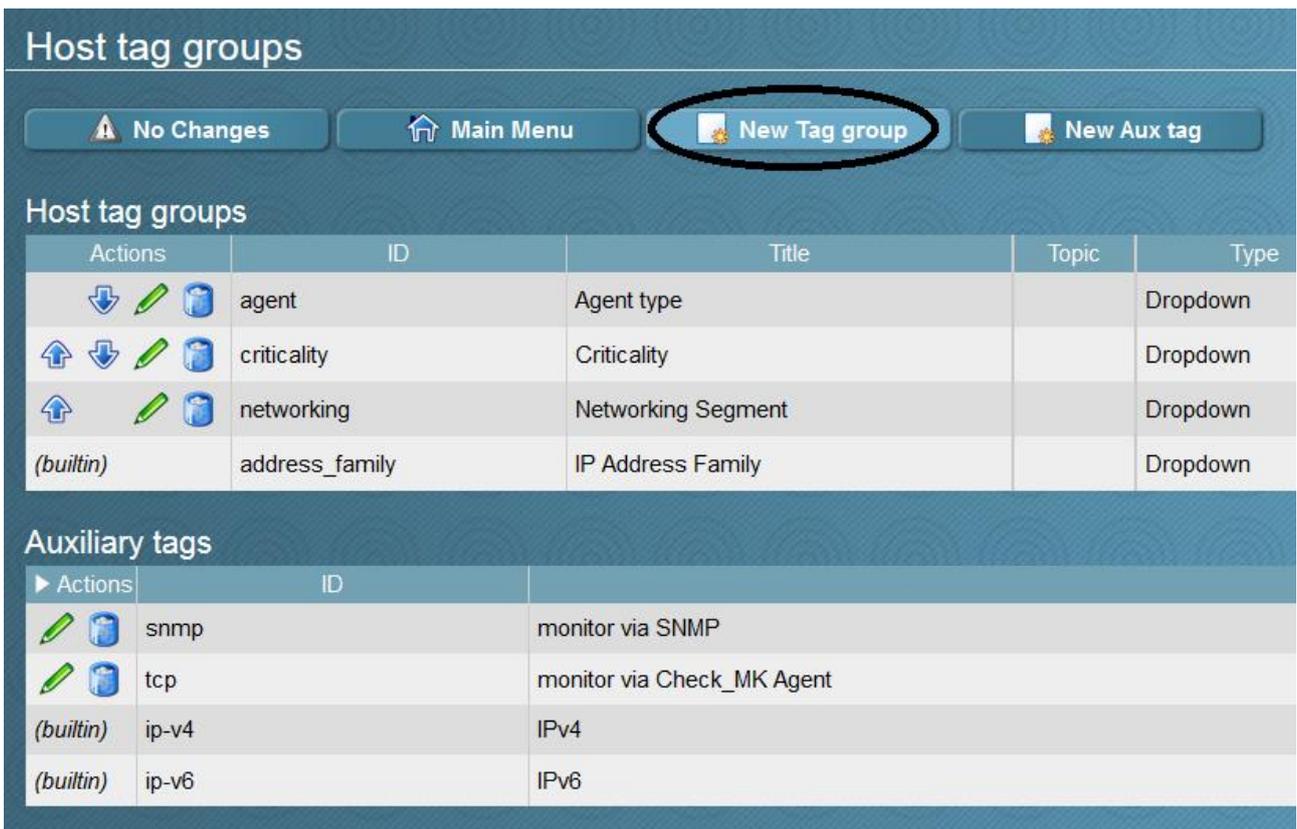


Tags

Clicking on *WATO*, *Host Tags* shows that there are some *Tags* already in place



To create a new Tag:



Create new tag group

[← All Hosttags](#)

▼ Edit group

Internal ID Linux

Title* Linux

Topic* Create New Topic ▾

Choices [Add tag choice](#)

*These texts may be localized depending on the users' language. You can configure the localizations [in the global settings](#).

Create new tag group

[← All Hosttags](#)

▼ Edit group

Internal ID Linux

Title* Linux

Topic* Create New Topic ▾

Choices

Tag ID	Description*	
Intranet_Production	All Linux servers for our Intranet Production	► Auxiliary tags
tranet_Development	All Linux servers for our Intranet Development/Te	► Auxiliary tags

[Add tag choice](#)

*These texts may be localized depending on the users' language. You can configure the localizations [in the global settings](#).

[Save](#)

Tags can be applied during the *New host* wizard process - or after by editing the properties of the device.

Create new host

Folder

Main directory

General Properties

Hostname Intranet_test1

Basic settings

Permissions empty (Default value)

Alias empty (Default value)

IPv4 Address empty (Default value)

Parents empty (Default value)

Monitored on site skytest - Local site skytest (Default value)

Host tags

Agent type Check_MK Agent (Server)

Criticality Productive system (Default value)

Networking Segment Local network (low latency) (Default value)

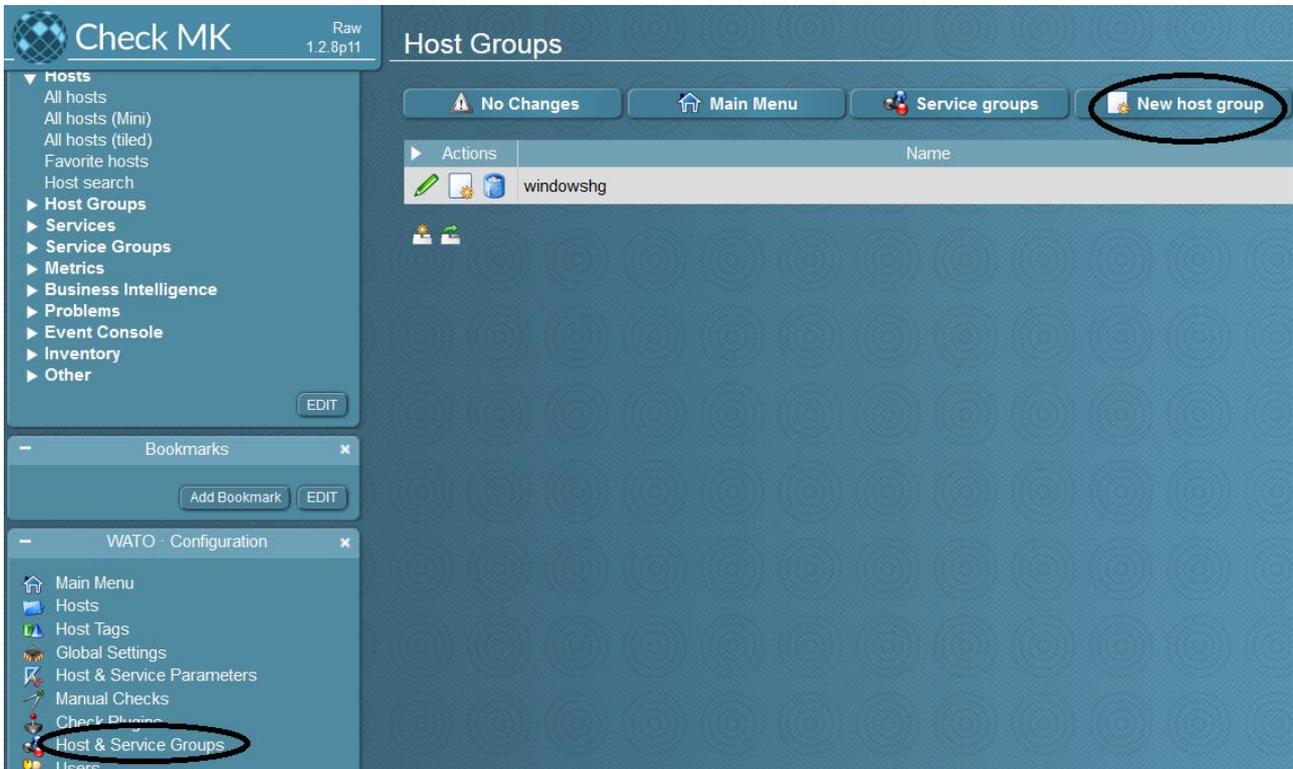
Linux **All Linux servers for our Intranet Production**

IP Address Family All Linux servers for our Intranet Development/Test

Save & go to Services Save & Finish Save & Test

Hostgroup

To create a new hostgroup click on *WATO, New host group*

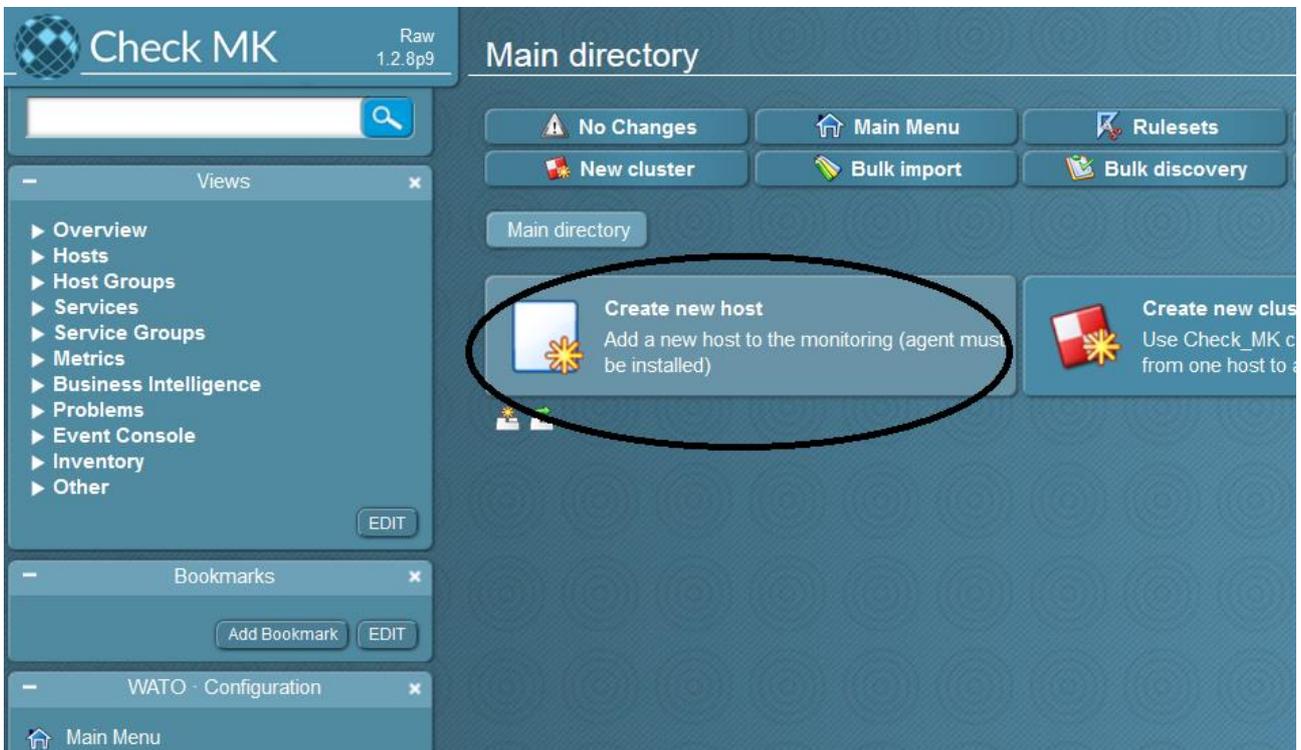


To do Host group assignment: *WATO, Host & Service Parameters, Grouping*

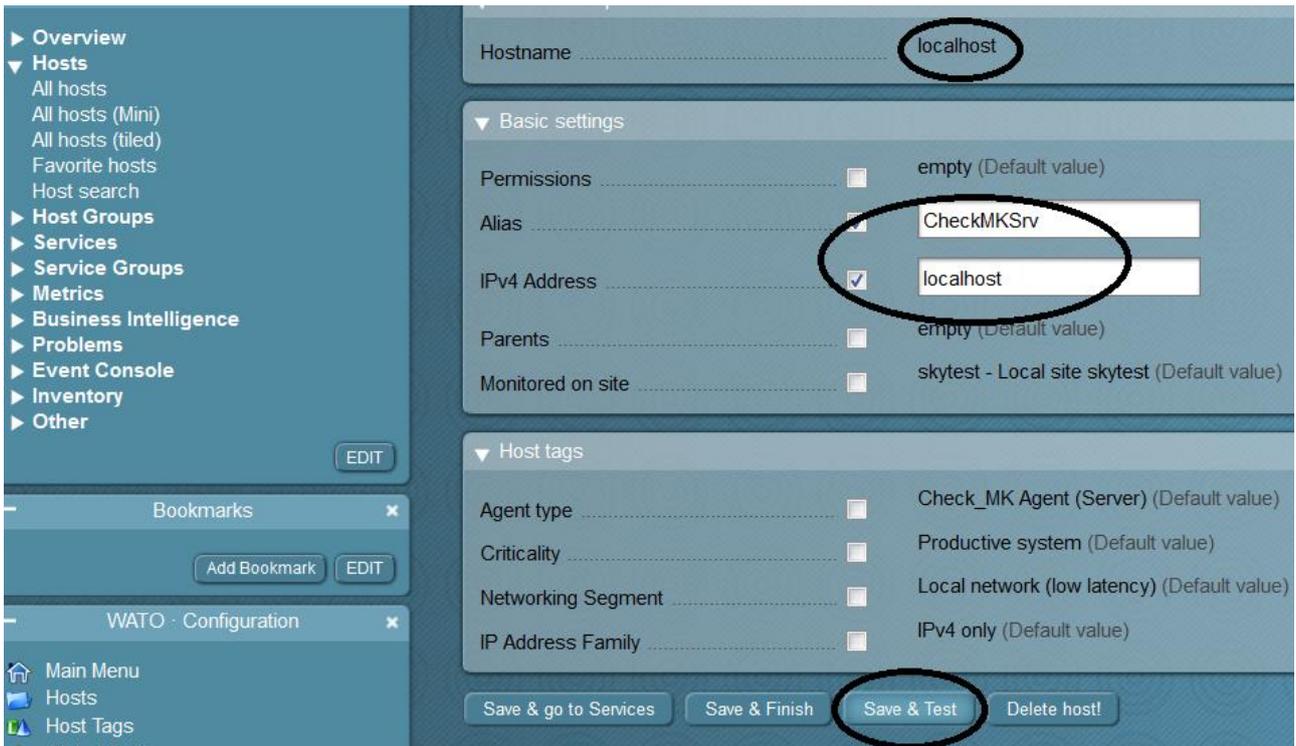


Linux Devices

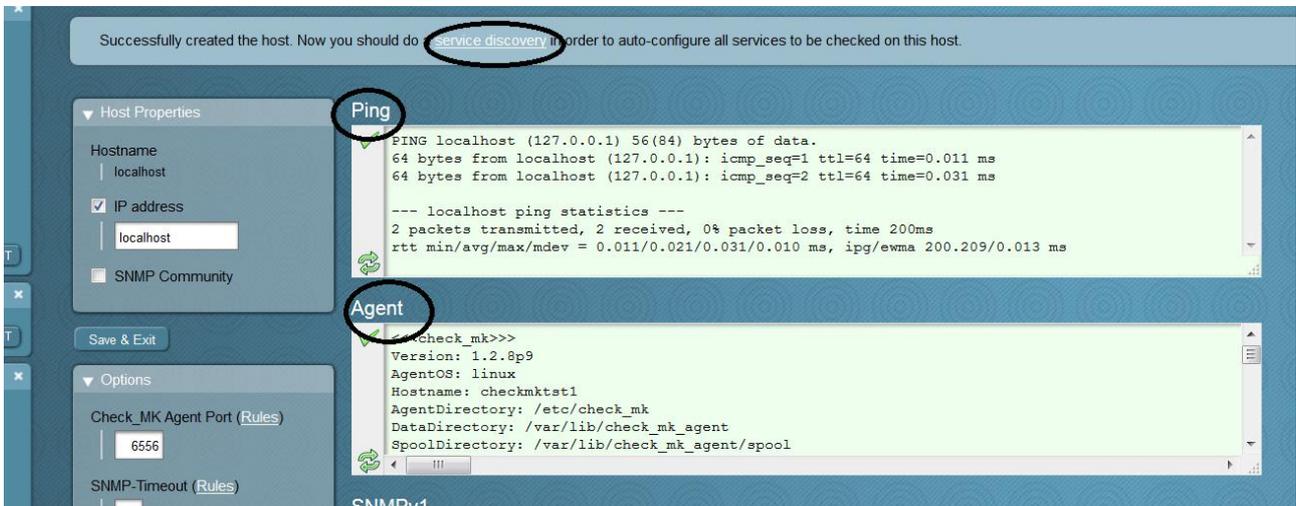
To add, remove a device, use *WATO: Hosts*



As Agent Type, leave the default *Check_MK Agent (Server)*



This is the output of an agent that is replying correctly



Click on *Service Discovery* and *Save manual check configuration*

Status	Checkplugin	Item	Service Description	Plugin out
OK	cpu.loads	None	CPU load	15 min load 0.12
OK	cpu.threads	None	Number of threads	181 threads
OK	df	/	Filesystem /	28.1% used (4.91 of 17.46 GB), trend: 0.00 B / 24 hours
OK	df	/boot	Filesystem /boot	42.4% used (210.52 of 496.67 MB), trend: 0.00 B / 24 hours
OK	diskstat	SUMMARY	Disk IO SUMMARY	Utilization: 0.0%, Read: 0.00 B/s, Write: 0.00 B/s, Average Wait: 0.00 ms Latency: 0.00 ms
PEND	kernel	Context Switches	Kernel Context Switches	WAITING - Counter based check, cannot be done offline
PEND	kernel	Major Page Faults	Kernel Major Page Faults	WAITING - Counter based check, cannot be done offline
PEND	kernel	Process Creations	Kernel Process Creations	WAITING - Counter based check, cannot be done offline
OK	kernel.util	None	CPU utilization	user: 1.9% system: 0.6% wait: 0.2% steal: 0.0% quest: 0.0% total: 2.8%

Apply changes

check MK Raw 1.2.8p9

2 Changes Main Menu Rulesets Manual Checks

New cluster Bulk import Bulk discovery Bulk renaming

Saved check configuration of host [localhost] with 20 services

Main directory

Hosts

Actions	Hostname	Alias	IPv4 Address	IPv6 Address	Parents	Monitored on
	localhost	CheckMKSrv	localhost			skytest - Local site s

Search

Raw 1.2.8p9

Pending changes to activate

Main Menu **Activate Changes!** Discard Changes! Site Configuration Aut

Actions	ID	Alias	Status	Version
	skytest	Local site skytest	online	1.2.8p9

Changes that are not yet activated

	2016-08-19 15:20:49	omdadmin	Saved check configuration of host [localhost] with 20 services
localhost	2016-08-19 15:20:16	omdadmin	Created new host localhost.

After a couple of minutes, we'll be able to see the list of all services the agent is monitoring on the host, along with their full status and their 'Perf-O-Meters' that show performance metrics where applicable.

Check MK Raw 1.2.8p9 All hosts

Views

- Overview
- Hosts
 - All hosts
 - All hosts (Mini)
 - All hosts (tiled)
 - Favorite hosts
 - Host search
- Host Groups
- Services
- Service Groups
- Metrics
- Business Intelligence
- Problems
- Event Console

Local site skytest

state	Host	Icons	OK	Wa	Un	Cr	Pd
UP	localhost		20	0	0	0	2

Services of Host localhost 34 rows omdadmin (admin) 14:29

WATO Host/Svc notific. Inventory Network Interfaces Edit View Availability

State	Service	Icons	Status detail	Age	Checked	Perf-O-Meter
OK	Check_MK		OK - Agent version 1.2.8p9, execution time 0.8 sec	2016-10-07 16:04:24	36 sec	757 ms
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found	2016-10-04 17:05:04	82 min	
OK	Check_MK HW/SW Inventory		OK - found 17146 entries	2016-09-30 12:33:05	114 min	
OK	Apache 127.0.0.1:5000 Status		OK - Uptime: 14 days, IdleWorkers: 9, BusyWorkers: 1, OpenSlots: 246, TotalSlots: 256, CPUload: 0.01, ReqPerSec: 0.10, BytesPerReq: 5369.85, BytesPerSec: 1774.93, States: (Waiting: 9, SendingReply: 1)	2016-09-30 12:34:33	35 sec	14 d
OK	CPU load		OK - 15 min load 0.07	2016-09-30 12:34:33	35 sec	0.0400
OK	CPU utilization		OK - user: 7.0%, system: 2.4%, wait: 0.0%, steal: 0.0%, guest: 0.0%, total: 9.4%	2016-09-30 14:53:33	35 sec	9.4%
OK	Disk IO LVM centos-root		OK - Utilization: 0.3%, Read: 1.93 kB/s, Write: 165.39 kB/s, Average Wait: 0.43 ms, Average Read Wait: 7.76 ms, Average Write Wait: 0.28 ms, Latency: 0.14 ms	2016-09-30 12:34:33	35 sec	2 kB/s / 165 kB/s
OK	Disk IO LVM centos-swap		OK - Utilization: 0.0%, Read: 0.00 B/s, Write: 0.00 B/s, Average Wait: 0.00 ms, Average Read Wait: 0.00 ms, Average Write Wait: 0.00 ms, Latency: 0.00 ms	2016-09-30 12:34:33	35 sec	0.00 B/s / 0.00 B/s
OK	Disk IO sda		OK - Utilization: 0.3%, Read: 1.93 kB/s, Write: 165.39 kB/s, Average Wait: 0.41 ms, Average Read Wait: 7.76 ms, Average Write Wait: 0.25 ms, Latency: 0.14 ms	2016-09-30 12:34:33	35 sec	2 kB/s / 165 kB/s
OK	Disk IO SUMMARY		OK - Utilization: 0.3%, Read: 1.93 kB/s, Write: 165.39 kB/s, Average Wait: 0.41 ms, Average Read Wait: 7.76 ms, Average Write Wait: 0.25 ms, Latency: 0.14 ms	2016-09-30 12:34:33	35 sec	2 kB/s / 165 kB/s
OK	Events		OK - no events for localhost/127.0.0.1	2016-10-10 17:11:52	15 sec	
OK	Filesystem /		OK - 40.2% used (7.02 of 17.46 GB), trend: +11.28 MB / 24 hours	2016-10-05 17:38:10	35 sec	40.2%
OK	Filesystem /boot		OK - 42.4% used (210.52 of 496.67 MB), trend: 0.00 B / 24 hours	2016-10-05 17:38:10	35 sec	42.4%
OK	Interface 2		OK - [eno16777984] (up) MAC: 00:0c:29:10:91:d2, 10.00 Gbit/s, in: 8.10 kB/s(0.0%), out: 3.32 kB/s(0.0%)	2016-09-30 12:34:33	35 sec	0.0% / 0.0%

A preview of detailed Performance Graphs (rrd) are accessible hovering the mouse over the graph icon.

State	Service	Icons	Status detail
OK	Check_MK		OK - Agent version 1.2.8p9, execution time 0.5 sec
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found
OK	Check_MK HW/SW Inventory		OK - found 17146 entries
OK	Apache 127.0.0.1:5000 Status		OK - Uptime: 14 days, IdleWorkers: 7, BusyWorkers: 1, OpenSockets: 10, CPUload: 0.01, ReqPerSec: 0.20, BytesPerReq: 5362.94, BytesPerSec: 1072.58
OK	CPU load		
OK	CPU utilization		
OK	Disk IO LVM centos-root		
OK	Disk IO LVM centos-swap		
OK	Disk IO sda		
OK	Disk IO SUMMARY		
OK	Events		
OK	Filesystem /		
OK	Filesystem /boot		
OK	Interface eth0		

Clicking on the icon causes the graphs to be displayed in a new window

Service details localhost -> Apache 127.0.0.1:5000 Status

Host: localhost Service: Apache 127.0.0.1:5000 Status

4 Hours 20.10.16 8:31 - 20.10.16 12:31

Datasource: Apache Status

localhost: Apache_127.0.0.1_5000_Status

Connections

Total Slots: 256

Component	Last	Max	Average
StartingUp	0.0	0.0	0.0
Waiting	9.0	9.0	6.4
Logging	0.0	0.0	0.0
DNS	0.0	0.0	0.0
SendingReply	1.0	2.2	1.0
ReadingRequest	0.0	0.0	0.0
Closing	0.0	0.0	0.0
IdleCleanup	0.0	0.0	0.0
Finishing	0.0	0.0	0.0
Keepalive	0.0	0.0	0.0
UsedSlots	10.0	10.0	7.4

Datasource: Requests/sec

localhost: Apache_127.0.0.1_5000_Status Requests/sec

File System Monitoring

By default, Check_MK creates a service for every filesystem and a specific service called *Disk IO Summary* that measures the throughput of block devices (disks) on Linux hosts. You can either have a single check for every single disk or a summary check (which is the default) summing up the throughput of all disks together.

OK	CPU load	OK - 15 min load 0.05
OK	CPU utilization	OK - user: 2.9%, system: 0.9%, wait: 0.0%, steal: 0.0%, guest: 0.0%, total: 3.8%
OK	Disk IO SUMMARY	OK - Utilization: 0.0%, Read: 0.00 B/s, Write: 35.44 kB/s, Average Wait: 0.29 ms, Average Re Wait: 0.29 ms, Latency: 0.02 ms
OK	Filesystem /	OK - 29.5% used (5.15 of 17.46 GB), trend: +11.94 MB / 24 hours
OK	Filesystem /boot	OK - 42.4% used (210.52 of 496.67 MB), trend: 0.00 B / 24 hours
OK	Interface 2	OK - [eno16777984] (up) MAC: 00:0c:29:10:91:d2, 10.00 Gbit/s, in: 1.10 kB/s(0.0%), out: 1.68
OK	Interface 3	OK - [eno335572481] (up) MAC: 00:0c:29:10:91:dc, 10.00 Gbit/s, in: 201.47 B/s(0.0%), out: 0.0

It's easy to change the default behavior as follows. Using *WATO: Host & Service Parameters, Parameters for discovered services, Storage, Filesystems and Files*

The screenshot shows the 'Rule-Based Configuration of Host & Service Parameters' interface. At the top, there are navigation buttons: 'No Changes', 'Main Menu', 'Used Rulesets', and 'Ineffective rules'. Below these is a 'Main directory' dropdown menu and a search bar for rule sets. The main content area contains several configuration options, each with a scissors icon and a description:

- Active checks (HTTP, TCP, etc.)**: Configure active networking checks like HTTP and TCP
- Access to Agents**: Settings concerning the connection to the Check_MK and SNMP agents
- Hardware/Software-Inventory**: Configuration of the Check_MK Hardware and Software Inventory System
- Grouping**: Assignment of host & services to host, service and contacts groups.
- Parameters for discovered services**: Levels and other parameters for checks found by the Check_MK service discovery. (This option is circled in black in the image.)
- Event Console**: Settings and Checks dealing with the Check_MK Event Console

Temperature, Humidity, Electrical Parameters, etc.			
Storage, Filesystems and Files			
Brocade FibreChannel ports	0	DR:BD roles and diskstates	0
ESX Datastores (used space and growth)	0	ESX Hostsystem Maintenance Mode	0
FibreChannel Ports (FCMGMT MIB)	0	File Grouping Patterns	0
Filesystem grouping patterns	0	Filesystem mount options (Linux/UNIX)	0
HP-UX Multipath Count	0	Heartbeat CRM general status	0
IBM SVC Pool Capacity	0	IBM SVC: Levels for total disk latency	0
IBM SVC: Options for SVC Hosts Check	0	Levels for disk IO	0
Linux Multipath Inventory	0	Linux and Solaris Multipath Count	0
MongoDB Collection Size	0	MongoDB Flushes	0
MongoDB Memory	0	NetApp Snapshot Reserve	0
NetApp Volumes	0	Netapp FC Port throughput	0
OpenHardwareMonitor S.M.A.R.T.	0	RAID: state of a single disk	0
Size and age of single files	0	Size, age and count of file groups	0
		Discovery mode for Disk IO check	0
		ESX Multipath Count	0
		Filer Disk Levels (NetApp)	0
		Filesystems (used space)	0
		Heartbeat CRM resource	0
		IBM SVC: Options for S	0
		Levels on disk IO (old style checks)	0
		MongoDB Assert Rates	0
		MongoDB Locks	0
		NetApp Snapvaults / Snapmirror Lag Time	0
		Number of Running Bossock Fibers	0
		Remaining blank tapes in DIVA CSM Devices	0
		Volume Groups (LVM)	0

Create a new rule

Discovery mode for Disk IO check

No Changes | **Main Menu** | **Parameters for disco...**

Main directory: ▼

Matching: The first matching rule defines the parameter.
There are no rules defined in this set.

Create rule in folder: Linux ▼

New rule Discovery mode for Disk IO check

Abort

This rule controls which and how many checks will be created for monitoring individual physical and logical disks. Note: the option *Create a summary for all read*, settings, but it will be removed there soon.

▼ Rule Options

Description: Default mode for Disk IO check

Comment: Change default behavior to have IO for each device instead summary

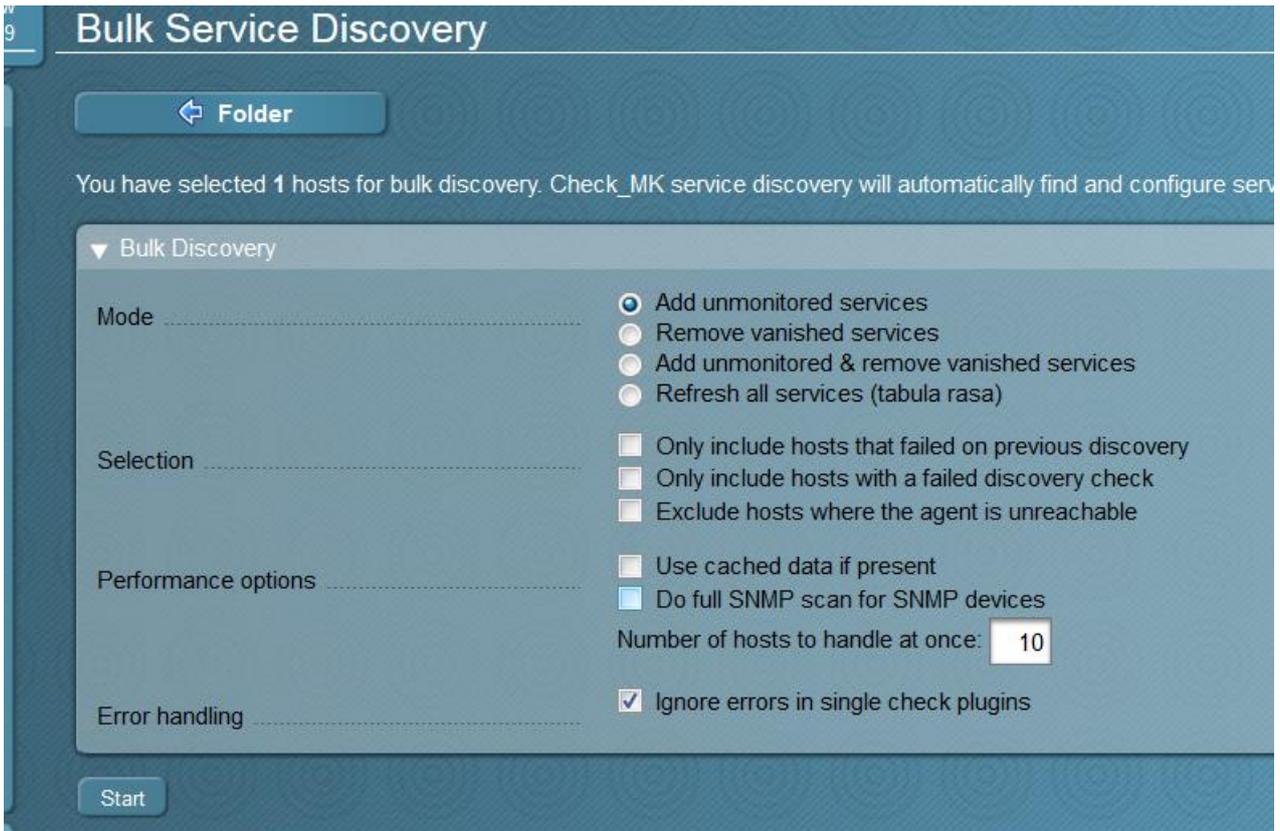
Documentation-URL:

Rule activation: do not apply this rule

▼ Discovery mode for Disk IO check

- Create a summary over all physical disks
- Create a separate check for each physical disk
- Create a separate check for each LVM volume (Linux)
- Create a separate check for each VxVM volume (Linux)

Do a Service discovery to add new services



Bulk Service Discovery

```
localhost: discovery successful
```

FINISHED.

Total hosts	1
Failed hosts	0
Skipped hosts	0
Services added	3
Services removed	0
Services kept	22
Total services	25

Click Finish and apply changes. The filesystem output should change to something like this:

localhost			
State	Service	Icons	Status detail
OK	Check_MK	 	OK - Agent version 1.2.8p9, execution time 0.4 sec
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found
OK	Check_MK HW/SW Inventory		OK - found 16992 entries
OK	Apache 127.0.0.1:5000 Status	 	OK - Uptime: 7 days, IdleWorkers: 8, BusyWorkers: 2, OpenSlots: 246, TotalS BytesPerReq: 3764.05, BytesPerSec: 20098.98, States: (Waiting: 8, SendingF
OK	CPU load	 	OK - 15 min load 0.08
OK	CPU utilization	 	OK - user: 17.7%, system: 3.3%, wait: 0.7%, steal: 0.0%, guest: 0.0%, total: 21
OK	Disk IO LVM centos-root	 	OK - Utilization: 1.0%, Read: 50.70 kB/s, Write: 22.00 kB/s, Average Wait: 2.56 Write Wait: 0.11 ms, Latency: 1.88 ms
OK	Disk IO LVM centos-swap	 	OK - Utilization: 0.0%, Read: 0.00 B/s, Write: 0.00 B/s, Average Wait: 0.00 ms, Wait: 0.00 ms, Latency: 0.00 ms
OK	Disk IO sda	 	OK - Utilization: 1.0%, Read: 50.70 kB/s, Write: 22.00 kB/s, Average Wait: 2.70 Write Wait: 0.11 ms, Latency: 1.99 ms
OK	Disk IO SUMMARY	 	OK - Utilization: 1.0%, Read: 50.70 kB/s, Write: 22.00 kB/s, Average Wait: 2.70 Write Wait: 0.11 ms, Latency: 1.99 ms
OK	Filesystem /	 	OK - 29.5% used (5.15 of 17.46 GB), trend: +11.93 MB / 24 hours

Linux Process Monitoring

Monitoring of Linux processes is achieved using the **ps** plugin. This looks through the list of current running processes for those matching a certain name or regular expression (and optionally for those owned by a certain user). It's also possible to define thresholds for the number of running processes as well for cpu or memory usage etc.

If you also need performance data, the **ps.perf** plugin does exactly the same as **ps** but, as might be expected, outputs performance data.

Let's monitor the **httpd** process:

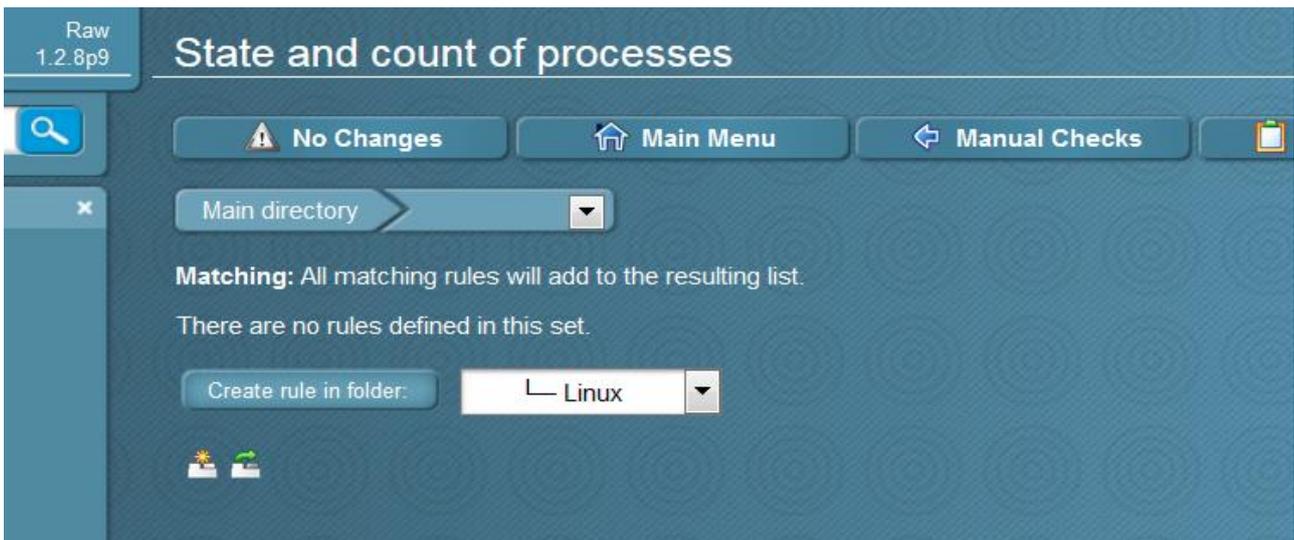
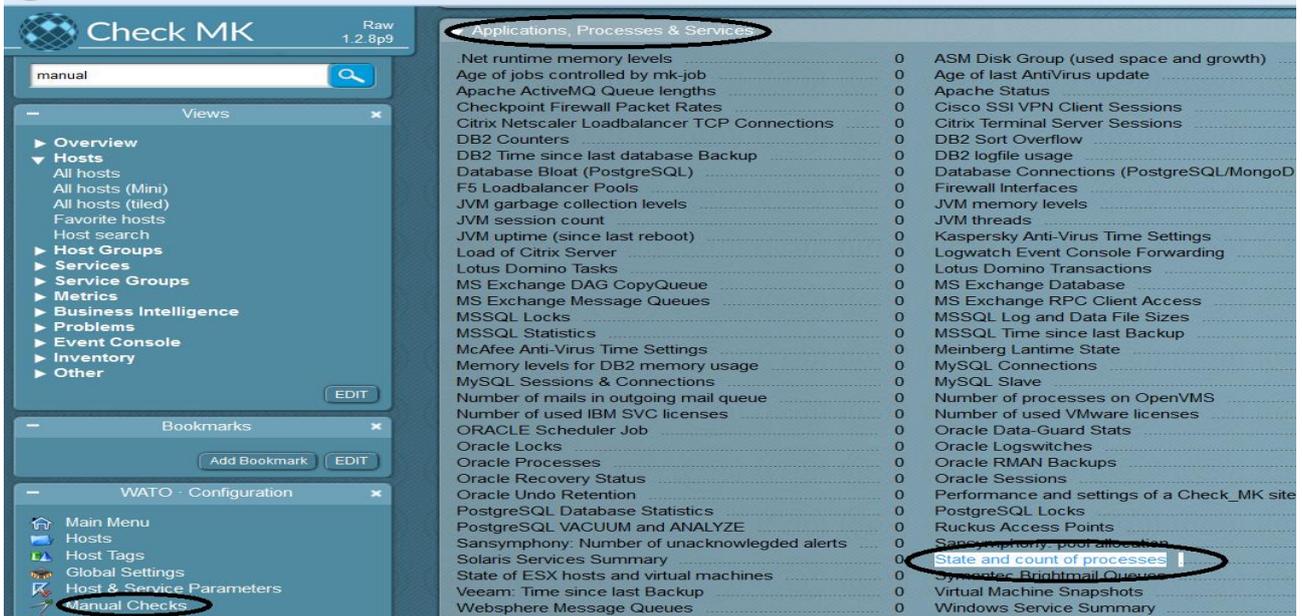
As a first step I suggest checking the specific command line arguments of the process from the shell:

```
[root@checkmktst1 ~]# ps -ef | grep httpd
mysite 395 12169 0 11:10 ?    00:00:01 /usr/sbin/httpd -f /omd/sites/mysite/etc/apache/apache.conf
apache 396 1029 0 11:10 ?    00:00:00 /usr/sbin/httpd -DFOREGROUND
root 1029 1 0 Aug26 ?    00:00:13 /usr/sbin/httpd -DFOREGROUND
apache 1928 1029 0 Aug29 ?    00:00:02 /usr/sbin/httpd -DFOREGROUND
apache 3606 1029 0 11:23 ?    00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 4427 1029 0 11:25 ?    00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 7587 1029 0 11:34 ?    00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 8519 1029 0 11:36 ?    00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 8944 1029 0 11:37 ?    00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 9086 1029 0 11:37 ?    00:00:00 /usr/sbin/httpd -DFOREGROUND
mysite 12169 1 0 Aug26 ?    00:00:10 /usr/sbin/httpd -f /omd/sites/mysite/etc/apache/apache.conf
root 14266 3560 0 11:55 pts/0 00:00:00 grep --color=auto httpd
apache 19129 1029 0 Aug31 ?    00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 19486 1029 0 10:24 ?    00:00:00 /usr/sbin/httpd -DFOREGROUND
mysite 19513 12169 0 10:24 ?    00:00:03 /usr/sbin/httpd -f /omd/sites/mysite/etc/apache/apache.conf
apache 19514 1029 0 10:24 ?    00:00:00 /usr/sbin/httpd -DFOREGROUND
mysite 19545 12169 0 10:24 ?    00:00:02 /usr/sbin/httpd -f /omd/sites/mysite/etc/apache/apache.conf
mysite 19579 12169 0 10:24 ?    00:00:02 /usr/sbin/httpd -f /omd/sites/mysite/etc/apache/apache.conf
mysite 19686 12169 0 00:00 ?    00:00:00 /usr/sbin/httpd -f /omd/sites/mysite/etc/apache/apache.conf
mysite 19688 12169 0 00:00 ?    00:00:04 /usr/sbin/httpd -f /omd/sites/mysite/etc/apache/apache.conf
mysite 19689 12169 0 00:00 ?    00:00:02 /usr/sbin/httpd -f /omd/sites/mysite/etc/apache/apache.conf
mysite 19690 12169 0 00:00 ?    00:00:05 /usr/sbin/httpd -f /omd/sites/mysite/etc/apache/apache.conf
mysite 19978 12169 0 00:01 ?    00:00:04 /usr/sbin/httpd -f /omd/sites/mysite/etc/apache/apache.conf
mysite 27189 12169 0 10:51 ?    00:00:02 /usr/sbin/httpd -f /omd/sites/mysite/etc/apache/apache.conf
mysite 31447 12169 0 11:04 ?    00:00:02 /usr/sbin/httpd -f /omd/sites/mysite/etc/apache/apache.conf
```

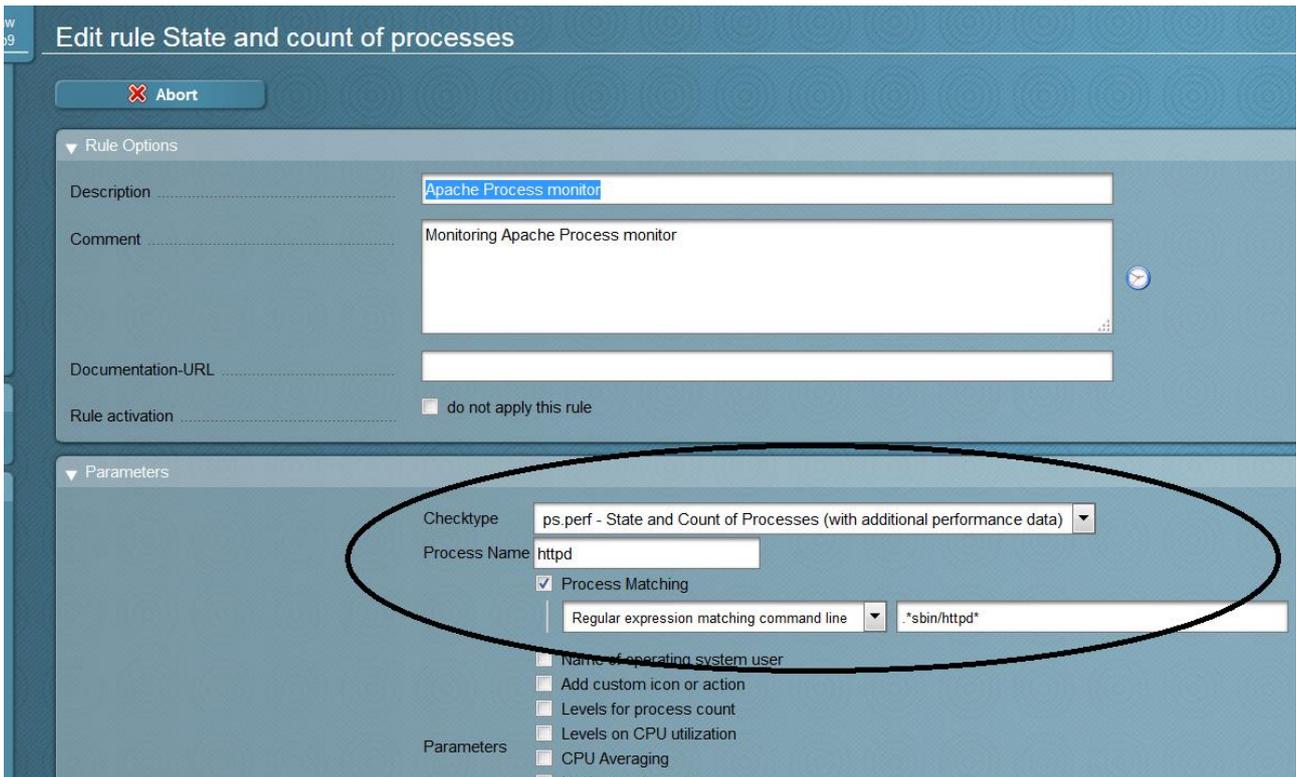
Count how many **httpd** processes are running:

```
[root@checkmktst1 ~]# pidof httpd | wc -w
24
```

Now we can define the service check for the Apache process using the GUI: WATO: *Manual Checks, Application Processes & Services, State and count processes*



Because we need to get all processes that have the string `"/sbin/httpd *`, we need to use a simple regular expression. The plugin homepage guides us through these steps.



Don't forget to save and apply changes!

Following this we should see that the new service check has been applied and useful performance graphs are being generated.

OK	Number of threads	 	OK - 184 threads
OK	OMD [redacted] apache	 	OK - 0.07 Requests/s, 0.00 Seconds serving/s, 143.46 B Sent/s
OK	OMD [redacted] performance	 	OK - Host Checks: 0.2/s, Service Checks: 11.9/s, Process Creations: 0.5/s, Livestatus Connects: 0.1/s, Livestatus Requests: 0.2/s, Log Messages: 0.0/s, 11 Hosts, 703 Services, Core version: 3.5.0, Livestatus version: 1.2.8p13
OK	OMD [redacted] status		OK - running
WARN	Postfix Queue	 	WARN - deferred queue length is 17 (Levels at 10/20) WARN , active queue length is 0
OK	Process httpd	 	OK - 24 processes 6766.6 MB virtual, 631.2 MB resident, 0.0% CPU, youngest running for 87 min oldest running for 18 hours
OK	[redacted]CUSTOMSCRIPT_TEST_Filecount_/tmp	 	OK - 8 files in /tmp
CRIT	[redacted]CUSTOMSCRIPT_TEST_Filecount_/var/log	 	CRIT - CRITICAL - 60 files in /var/log
OK	[redacted]_TST_TCPCHECK_SMTTP	 	TCP OK - 0.001 second response time on 127.0.0.1 port 25
OK	TCP Connections	 	OK - ESTABLISHED: 2, TIME_WAIT: 5, LISTEN: 10
OK	Uptime	 	OK - up since Wed Oct 26 16:38:30 2016 (0d 18:40:01)



Log Files

Logfiles on Linux are monitored using the *logwatch* extension for the *check_mk_agent*.

-Copy *mk_logwatch* in the plugin directory.

```
cp /opt/omd/versions/1.2.8p9.cre/share/check_mk/agents/plugins/mk_logwatch /usr/lib/check_mk_agent/plugins
```

Create the file */etc/check_mk/logwatch.cfg* with the following text:

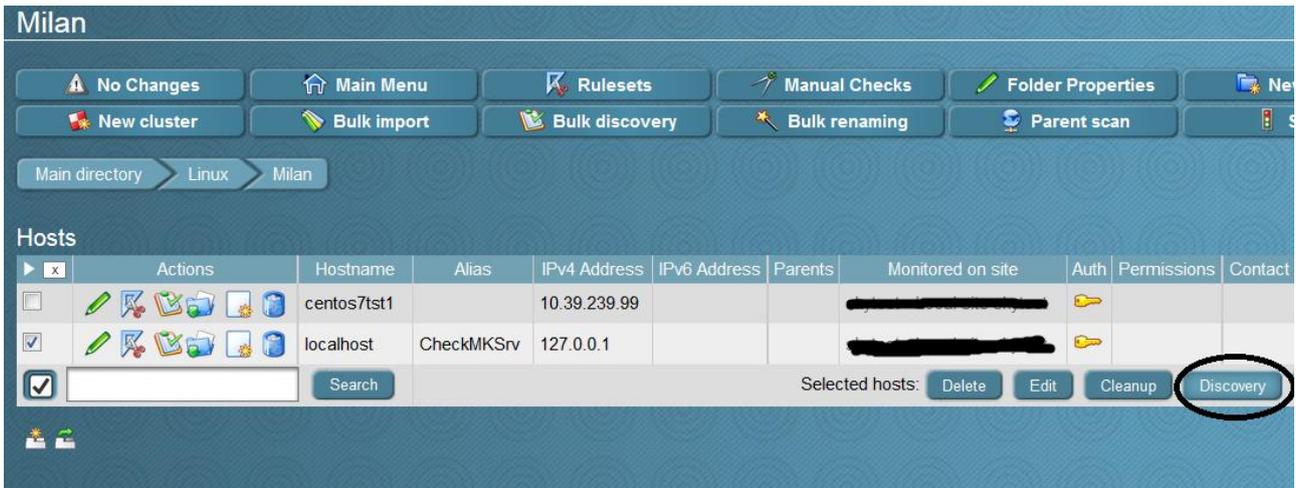
```
/var/log/messages
C Error*
R TEST: This is a fake error, monitoring a logfile just as test \1
```

The first line specified the text file we want to monitor; the second means that if the agent finds the expression "Error" (followed by any words) a critical error will be created. The last creates a rewrite rule, customizing the message that will be displayed within the GUI

-Restart the agent

service xinetd restart

-Do a discovery on localhost so that the new check will be automatically added



-Activate changes

-Do a test

echo "Error" >> /var/log/messages

-Test the agent from command line

su - mysite

<<<job>>>

<<<local>>>

<<<logwatch>>>

[[[/var/log/messages]]

Aug 26 15:35:01 checkmktst1 systemd: Created slice user-986.slice.

Aug 26 15:35:01 checkmktst1 systemd: Starting user-986.slice.

Aug 26 15:35:01 checkmktst1 systemd: Started Session 409 of user mysite.

Aug 26 15:35:01 checkmktst1 systemd: Starting Session 409 of user mysite.

Aug 26 15:35:01 checkmktst1 systemd: Started Session 410 of user mysite.

Aug 26 15:35:01 checkmktst1 systemd: Starting Session 410 of user mysite.

Aug 26 15:35:01 checkmktst1 systemd: Started Session 411 of user mysite.

Aug 26 15:35:01 checkmktst1 systemd: Starting Session 411 of user mysite.

Aug 26 15:35:02 checkmktst1 systemd: Removed slice user-986.slice.

Aug 26 15:35:02 checkmktst1 systemd: Stopping user-986.slice.

C TEST: This is a fake error, monitoring a logfile just as test \1

Aug 26 15:35:09 checkmktst1 su: (to mysite) root on pts/0

-Look at WATO to check if the CRITICAL has been generated

All hosts

3 30s Edit View Availability

Local site skytest

state	Host	Icons	OK	Wa	Un	Cr	Pd	state	Host	Icons	OK
UP	localhost	 	23	0	0	1	0	UP	Switch_10.39.238.28	 	53

CRIT Services of host localhost

2 30s WATO Services Inventory Inventory History

localhost

State	Service	Icons	Status detail	Age	Checked
CRIT	Log /var/log/messages		CRIT - 1 CRIT messages (Last worst: "TEST: This is a fake error, monitoring a logfile just as test \1")	6 sec	6 sec

Windows Devices

Download and Install the *check_mk_agent.msi* on the Windows server. The same steps that we carried out for Linux also apply to Windows devices.



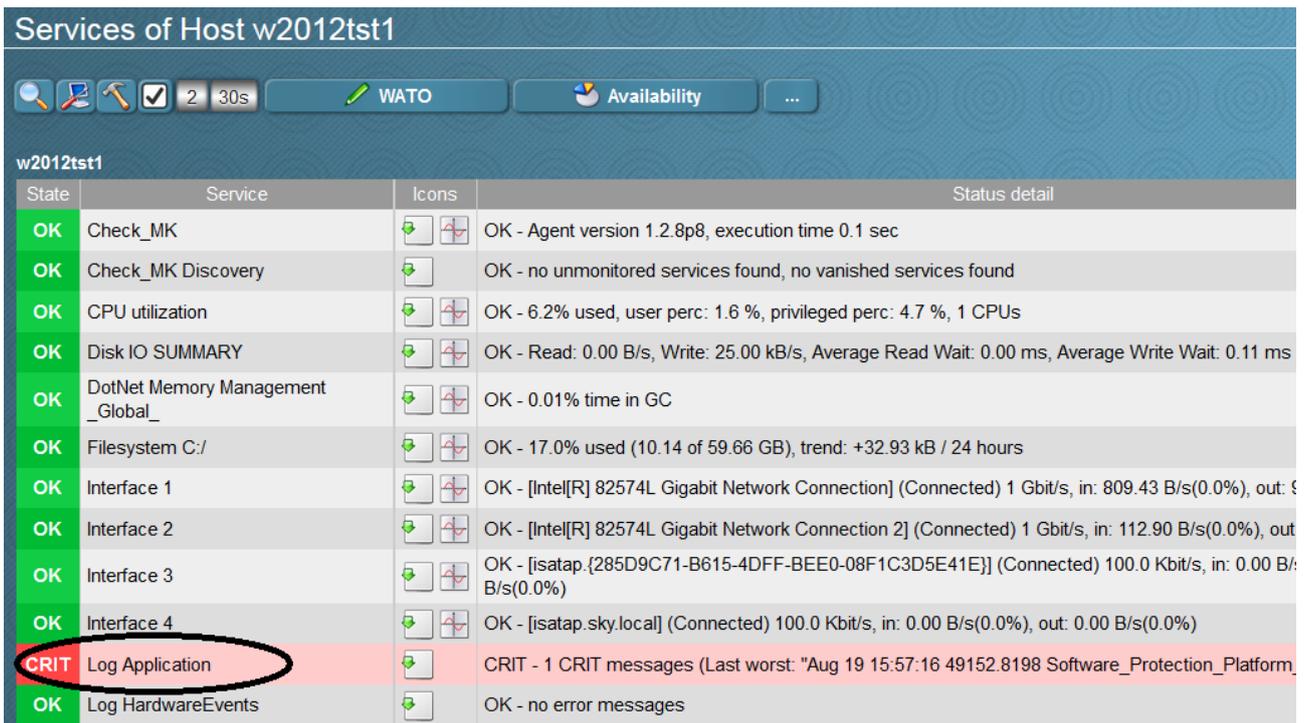
All hosts 3 rows omdadmin (admin) 15:59

Local site skytest

state	Host	Icons	OK	Wa	Un	Cr	Pd	state	Host	Icons	OK	Wa	Un	Cr	Pd	state	Host	Icons	OK	Wa	Un	Cr	Pd
UP	localhost		22	0	0	0	0	UP	Switch_10.39.238.28		52	0	0	0	0	UP	w2012tst1		20	0	0	0	0

Windows Event Viewer

By default the Windows agent sends all non-informational messages to the Check_MK server. We can see here that Check_MK automatically detected an error in the Windows Event Log.



Services of Host w2012tst1

w2012tst1

State	Service	Icons	Status detail
OK	Check_MK		OK - Agent version 1.2.8p8, execution time 0.1 sec
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found
OK	CPU utilization		OK - 6.2% used, user perc: 1.6 %, privileged perc: 4.7 %, 1 CPUs
OK	Disk IO SUMMARY		OK - Read: 0.00 B/s, Write: 25.00 kB/s, Average Read Wait: 0.00 ms, Average Write Wait: 0.11 ms
OK	DotNet Memory Management _Global_		OK - 0.01% time in GC
OK	Filesystem C:/		OK - 17.0% used (10.14 of 59.66 GB), trend: +32.93 kB / 24 hours
OK	Interface 1		OK - [Intel[R] 82574L Gigabit Network Connection] (Connected) 1 Gbit/s, in: 809.43 B/s(0.0%), out: 9
OK	Interface 2		OK - [Intel[R] 82574L Gigabit Network Connection 2] (Connected) 1 Gbit/s, in: 112.90 B/s(0.0%), out
OK	Interface 3		OK - [isatap.{285D9C71-B615-4DFF-BEE0-08F1C3D5E41E}] (Connected) 100.0 Kbit/s, in: 0.00 B/s(0.0%)
OK	Interface 4		OK - [isatap.sky.local] (Connected) 100.0 Kbit/s, in: 0.00 B/s(0.0%), out: 0.00 B/s(0.0%)
CRIT	Log Application		CRIT - 1 CRIT messages (Last worst: "Aug 19 15:57:16 49152.8198 Software_Protection_Platform
OK	Log HardwareEvents		OK - no error messages

Since the agent is completely configuration-less, it doesn't do specific filtering of events. It simply looks for messages of type **Warning** or **Error**. This behavior can be changed by creating a file called *check_mk.ini* in the agent directory but, in my opinion, this isn't the best way - if you have hundreds of servers, re-deploying the configuration file and restarting all agents can be a pain. A better approach is to create "centralized" rules which specify a list of "windows event id" or strings for each "Windows event log" that you consider critical. I know that this solution requires some time to optimize, but with a bit of experience

(and Google searching!), it can have excellent results. For example, in my environment I added some rules relating to Oracle (e.g. "ORA-RAC"), MSSQL (e.g. cluster failed) etc.

Click on *Logfile Pattern Analyzer*, *Edit Logfile Rules*

The screenshot shows the Check MK interface for the Logfile Pattern Analyzer. The left sidebar contains a navigation menu with 'Logfile Pattern Analyzer' circled. The main area features a 'Main Menu' button and a circled 'Edit Logfile Rules' button. Below these is a 'Try Pattern Match' section with input fields for Hostname, Logfile, and Text to match. A 'Try out' button is also present. The 'Logfile Patterns' section shows a table with one rule highlighted in green:

Match	State	Pattern
	CRIT	testeventviewer*
	OK	

In this picture you can see the rule for *System event log*. **Please pay attention to the order of the rules!** See that ignore is on the bottom, and then I'm adding values on the top as they fire from the top down. Note the *WARNING* or *CRITICAL* entries I'm making for the specific entries I've added.

Abort

You can define one or several patterns (regular expressions) in each logfile pattern rule. These patterns are applied to the selected logfiles to reclassify the matching log messages. The first pattern which matches a line will be used for reclassifying a message. You can use the [Logfile Pattern Analyzer](#) to test the rules you defined here.
Select "Ignore" as state to get the matching logs deleted. Other states will keep the log entries but reclassify the state of them.

▼ Conditions

Folder Main directory ▾

Host tags Agent type: ignore ▾
Criticality: ignore ▾
Networking Segment: ignore ▾
Site Location: ignore ▾
monitor via SNMP: ignore ▾
monitor via Check_MK Agent: ignore ▾

Explicit hosts Specify explicit host names

Logfile Specify explicit values
System\$

▼ Logfile pattern rules

State	Pattern (Regex)	Comment
IGNORE ▾	Ntfs	
CRITICAL ▾	[Ff]ailure	
CRITICAL ▾	[Pp]hysical [Dd]isk	
WARNING ▾	[Pp]redictive	
CRITICAL ▾	2048	
CRITICAL ▾	2050	
CRITICAL ▾	2052	
CRITICAL ▾	2057	
CRITICAL ▾	2065	
CRITICAL ▾	2092	
CRITICAL ▾	2094	
CRITICAL ▾	2121	
CRITICAL ▾	last unexpected shutdown	
CRITICAL ▾	kernel power manager	
CRITICAL ▾	operating system is shutting down	
WARNING ▾	operating system started	
IGNORE ▾		

▼ Additional options

Comment

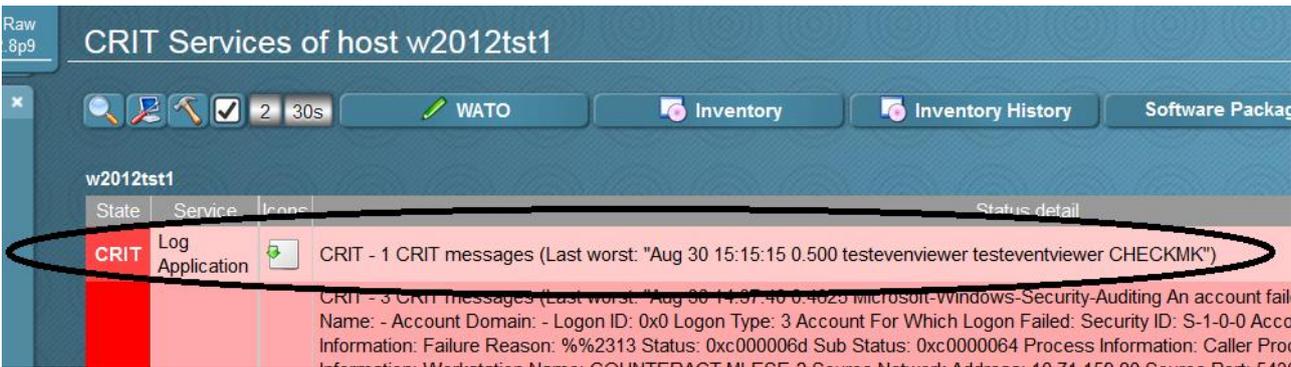
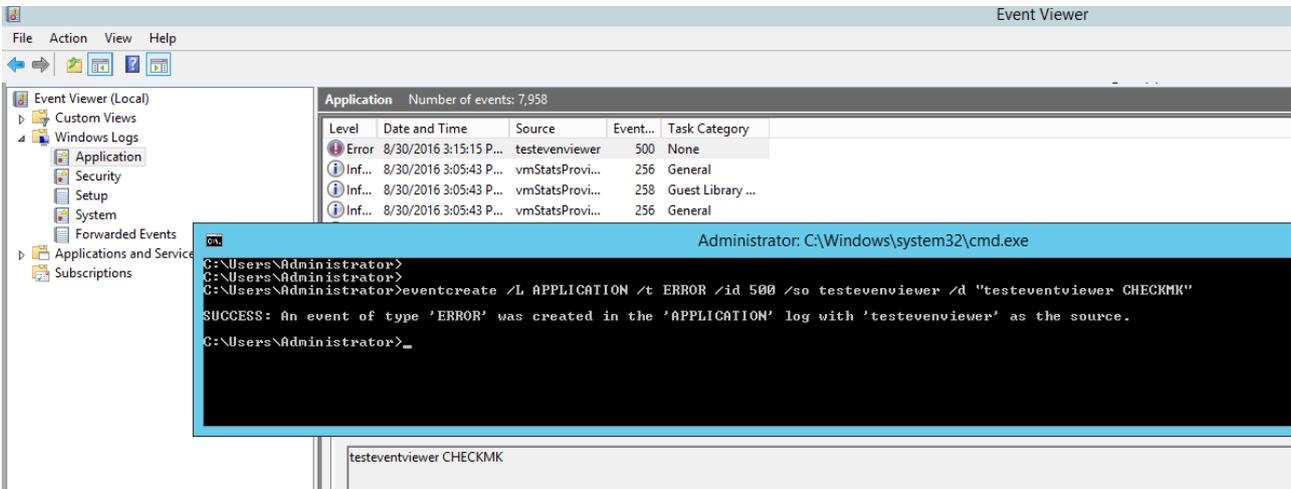
Documentation-URL

Rule activation do not apply this rule



I test these on a Windows server using *eventcreate* (in this example I'm using the string "testeventviewer CHECKMK" that isn't present in the previous screenshot but for which I've added a rule in my configuration)

```
C:\Users\Administrator>eventcreate /L APPLICATION /t ERROR /id 500 /so testeviewer /d "testeventviewer CHECKMK"
```



Windows Services

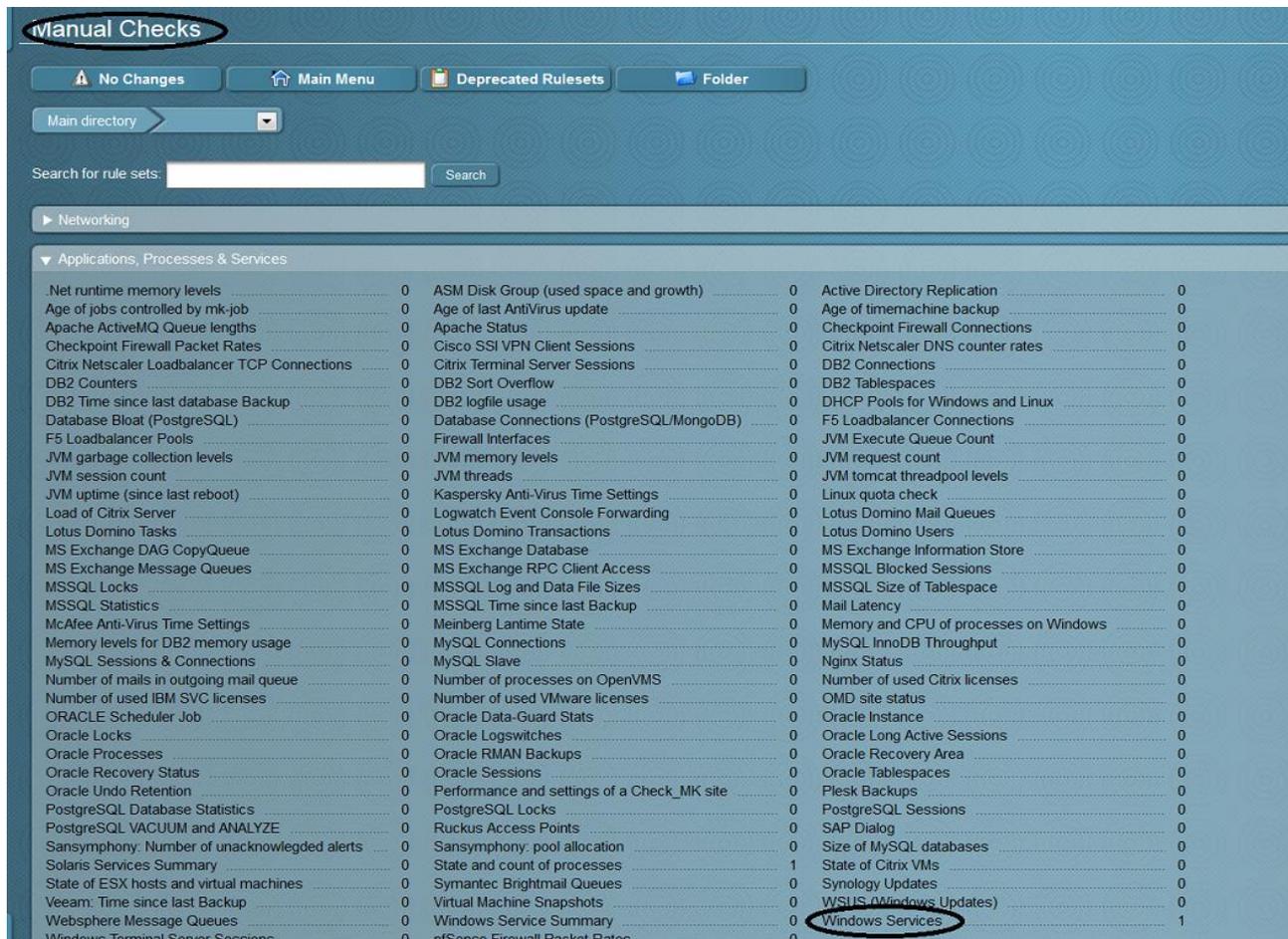
It's also possible to monitor Windows Services but, in this case, we need to specify the name of the services that we would like to monitor. We can specify a list of services that should be monitored on all hosts or just on some of them. In this example I'll show how to monitor "Terminal Server Service" on host `w2012tst1`.

In order to monitor services you first need to determine which services are of interest to you. The easiest way is to look at the raw output of the agent and look for the section `<<<services>>>`. You can use `cmk -d` for this:

```
OMD[mysite]:~$ cmk -d w2012tst1 | fgrep -A 100 '<<<services>>>' | grep -i running
BFE running/auto Base Filtering Engine
BrokerInfrastructure running/auto Background Tasks Infrastructure Service
CertPropSvc running/demand Certificate Propagation
Check_MK_Agent running/auto Check_MK_Agent
COMSysApp running/demand COM+ System Application
CryptSvc running/auto Cryptographic Services
...
...
...
TermService running/demand Remote Desktop Services
```

The first column of the output is the exact internal name of the service. Let's say you want to check if TermService (Windows Terminal Server) is running on host `w2012tst1`.

WATO, Manual Checks



The screenshot shows the WATO Manual Checks interface. At the top, there is a "Manual Checks" header. Below it, there are several buttons: "No Changes", "Main Menu", "Deprecated Rulesets", and "Folder". A "Main directory" dropdown menu is set to "Networking". A search bar is present with the text "Search for rule sets:" and a "Search" button. The main content area is titled "Applications, Processes & Services" and contains a long list of services to monitor. The list is organized into two columns. The first column lists various services, and the second column lists their corresponding internal names. The service "Windows Terminal Server Sessions" is highlighted in blue, and its internal name "TermService" is circled in red. Other services listed include .Net runtime memory levels, Apache ActiveMQ Queue lengths, Checkpoint Firewall Packet Rates, Citrix Netscaler Loadbalancer TCP Connections, DB2 Counters, DB2 Time since last database Backup, Database Bloat (PostgreSQL), F5 Loadbalancer Pools, JVM garbage collection levels, JVM session count, JVM uptime (since last reboot), Load of Citrix Server, Lotus Domino Tasks, MS Exchange DAG CopyQueue, MS Exchange Message Queues, MSSQL Locks, MSSQL Statistics, McAfee Anti-Virus Time Settings, Memory levels for DB2 memory usage, MySQL Sessions & Connections, Number of mails in outgoing mail queue, Number of used IBM SVC licenses, ORACLE Scheduler Job, Oracle Locks, Oracle Processes, Oracle Recovery Status, Oracle Undo Retention, PostgreSQL Database Statistics, PostgreSQL VACUUM and ANALYZE, Sansymphony: Number of unacknowledged alerts, Solaris Services Summary, State of ESX hosts and virtual machines, Veeam: Time since last Backup, Websphere Message Queues, and Windows Firewall Packet Rates.

Create a rule like this one:

Edit rule Windows Services omdadmin (admin) 15:50

Rule Options

Description: Terminal Server Service monitored on host w2012tst1

Comment: [Empty text area]

Documentation-URL: [Empty text area]

Rule activation: do not apply this rule

Parameters

Checktype: services - Windows Services

Name of the service: TermService

Parameters: Alternative names for the service
 Services states
 State if no entry matches
 Add custom icon or action

Conditions

Folder: L - Windows

Host tags: Agent type: ignore, Criticality: ignore, Networking Segment: ignore, IP Address Family: ignore, monitor via SNMP: ignore, monitor via Check_MK Agent: ignore, IPv4: ignore, IPv6: ignore

Explicit hosts: Specify explicit host names
w2012tst1

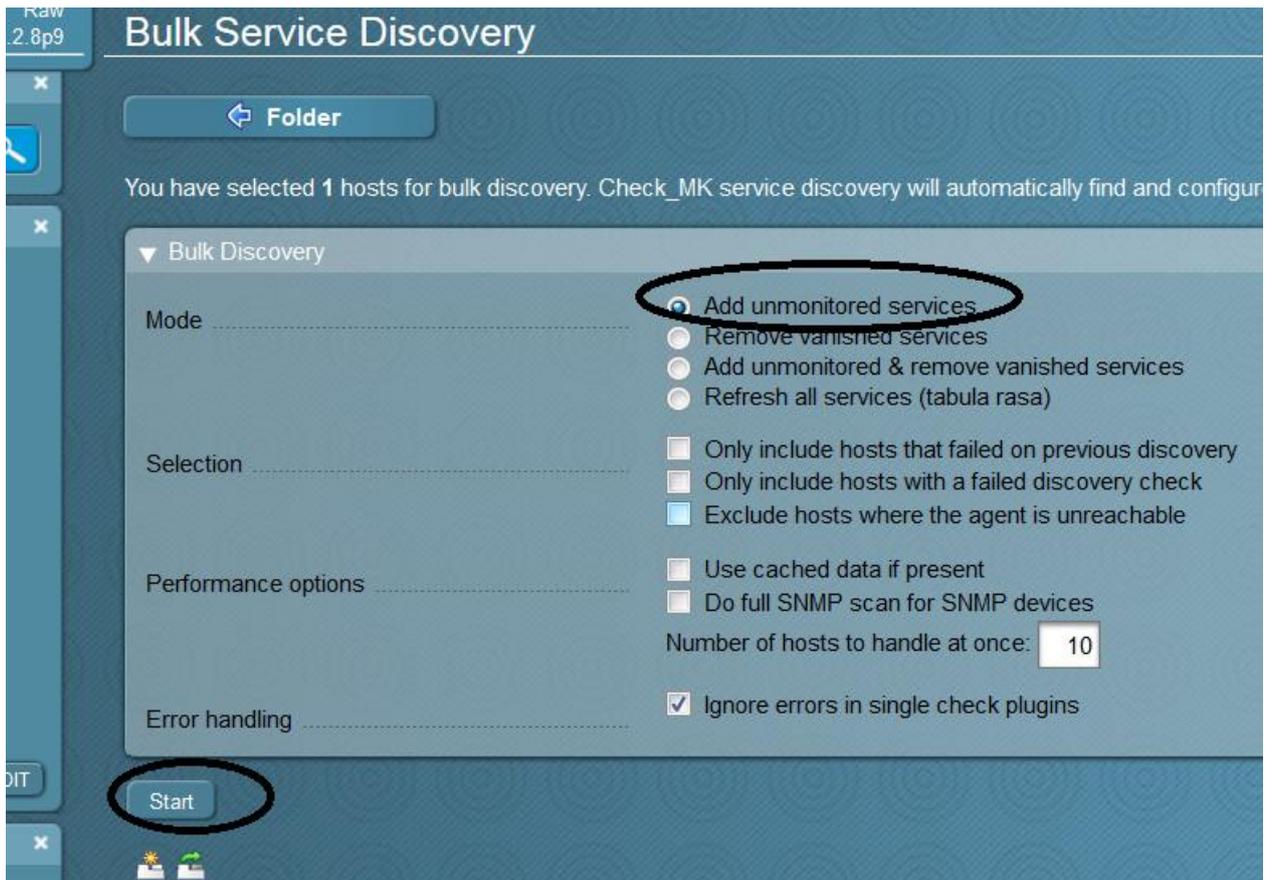
Force a host discovery using the command line or the GUI:

Main directory > Search results for folder Main directory

Hosts

Actions	Hostname	Alias	IPv4 Address	IPv6 Address	Parents	Monitored on site	Auth	Permissions	Contact Groups
<input checked="" type="checkbox"/>	w2012tst1		10.39.239.101			Local site			lan ip-v4 site:cl

Search Selected hosts:



OK	Log Windows PowerShell		OK - no error messages
OK	Memory and pagefile		OK - Memory usage: 37.2% (0.7/2.0 GB), Commit Charge: 32.6% (0.8/2.4 GB)
OK	Processor Queue		OK - 15 min load 5.06
OK	Service TermService		OK - Remote Desktop Services: running (start type is demand)
OK	Services Summary		OK - 139 services, 43 services in autostart - of which 2 services are stopped (RemoteRegistry, ignored)
OK	System Time		OK - Offset is 1 sec (warn/crit at 30/60 sec)
OK	Uptime		OK - up since Fri Aug 19 17:13:32 2016 (3d 00:17:06)

What WATO has done is write the following configuration file:

```
OMD[mysite]:/opt/omd/sites/mysite$ cat
/opt/omd/sites/mysite/etc/check_mk/conf.d/wato/windows/rules.mk
```

```
# Created by WATO
# encoding: utf-8
```

```
logwatch_rules = [
    ( ('C', u'testeventviewer*', u''), ('I', u'', u''), ['/' + FOLDER_PATH + '/+'],
    ALL_HOSTS, [u'Application$'], {'comment': u'This filter decides which events to take from
    the "Application" Windows Event Log\n', 'description': u'Filter Application Windows Event
    Log'} ),
] + logwatch_rules
```

```

static_checks.setdefault('services', [])

static_checks['services'] = [
    ( ('services', 'TermService', {}), ['/' + FOLDER_PATH + '/+'], ['w2012tst1'],
    {'description': u'Terminal Server Service monitored on host w2012tst1'} ),
] + static_checks['services']

host_groups = [
    ( 'windowshg', ['/' + FOLDER_PATH + '/+'], ALL_HOSTS, {'comment': u'All hosts in
Windows folder are automatically placed in the windowshg hostgroupu\n', 'description':
u'Windows hostgroup assignement'} ),
] + host_groups

```

If you have some services that should always be running on ALL windows hosts, the best way is create the rule that applies to ALL Windows Hosts; to do that, don't fill the *Explicit hosts* option.

The screenshot shows the 'Edit rule Windows Services' interface in Nagios WATO. The 'Rule Options' section contains a description field with the text 'Terminal Server Service monitored on ALL Windows Hosts' and a comment field with the text 'On ALL Windows Hosts, Terminal Services should always be up and running'. The 'Parameters' section shows a dropdown for 'Checktype' set to 'services - Windows Services' and a text field for 'Name of the service' containing 'TermService'. Below this are checkboxes for 'Alternative names for the service', 'Services states', 'State if no entry matches', and 'Add custom icon or action'. The 'Conditions' section shows a dropdown for 'Folder' set to 'L - Windows'. Below this are several dropdown menus for 'Agent type', 'Criticality', 'Networking Segment', 'IP Address Family', 'monitor via SNMP', 'monitor via Check_MK Agent', 'IPv4', and 'IPv6', all set to 'ignore'. At the bottom, there is an unchecked checkbox for 'Specify explicit host names' and a 'Save' button.

This time, WATO changed the following configuration file:

```

OMD[mysite]:/opt/omd/sites/mysite$ tail -5
/opt/omd/sites/mysite/etc/check_mk/conf.d/wato/rules.mk

```

```

static_checks['services'] = [

```

```
( ('services', 'TermService', {}), [], ALL_HOSTS, {'comment': u'On ALL Windows Hosts,
Terminal Services should always be up and running\n', 'description': u'Terminal Server
Service monitored on ALL Windows Hosts'} ),
] + static_checks['services']
```

Let's add DnsCache (windows DNS client) to the monitored services, and do a test by stopping the DNSClient.

To add the new service, create another rule using WATO or manually change the configuration file and reload the configuration:

```
OMD[mysite]:/opt/omd/sites/mysite$ cat
/opt/omd/sites/mysite/etc/check_mk/conf.d/wato/windows/rules.mk

static_checks['services'] = [
    ( ('services', 'TermService', {}), ['/' + FOLDER_PATH + '/+'], ALL_HOSTS, {'comment':
u'On ALL Windows Hosts, Terminal Services should always be up and running\n',
'description': u'Terminal Server Service monitored on ALL Windows Hosts'} ),
    ( ('services', 'Dnscache', {}), ['/' + FOLDER_PATH + '/+'], ALL_HOSTS, {'comment':
u'The windows service that manage dnsclient should be always up and running\n',
'description': u'DnsCache monitored on ALL Windows Hosts'} ),
] + static_checks['services']
```

Now stop the service on the windows host and wait a minute. A *CRITICAL* service should be displayed!

Icons	OK	Wa	Un	Cr	Pd	state	Host	Icons	OK	Wa	Un	Cr
	53	0	0	0	0	UP	w2012tst1		25	0	0	1

refresh: 30 s



Microsoft SQL Server

This is accomplished using the plugin “mssql.vbs” the documentation for which says the following:

The current implementation of the check uses the "trusted authentication" where no user/password needs to be created in the MSSQL server instance by default. It is only needed to grant the user as which the Check_MK windows agent service is running access to the MSSQL database.

Another option is to create a mssql.ini file in MK_CONFDIR and write the credentials of a database user to it which shall be used for monitoring:

```
[auth]
type = db
username = monitoring
password = secret-pw
```

I tested against Microsoft SQL Server 2014 64bit on Windows 2012 R2 using the default “trusted authentication”. This didn’t require any steps either on the SQL side or in check_mk.ini

Steps:

- Copy mssql.vbs from check_mk host to the agent plugin folder, in my case: *C:\Program Files (x86)\check_mk\plugins*
- Restart the agent
- Do a service discovery adding unmonitored services
- Activate Changes

46 new services were added to my w2012tst1 host:

OK	SQLServer _Total File Sizes	 	OK - Data Files: 68.31 MB, Log files Used: 2.26 MB, Log Files total: 4.96 MB
OK	SQLServer _Total Transactions		OK - Transactions: 0.0/s, Tracked Transactions: 0.0/s, Write Transactions: 0.0/s
OK	SQLServer master File Sizes	 	OK - Data Files: 4.00 MB, Log files Used: 609.00 kB, Log Files total: 2.24 MB
OK	SQLServer master Transactions		OK - Transactions: 0.0/s, Tracked Transactions: 0.0/s, Write Transactions: 0.0/s
OK	SQLServer model File Sizes	 	OK - Data Files: 2.19 MB, Log files Used: 405.00 kB, Log Files total: 504.00 kB
OK	SQLServer model Transactions		OK - Transactions: 0.0/s, Tracked Transactions: 0.0/s, Write Transactions: 0.0/s
OK	SQLServer msdb File Sizes	 	OK - Data Files: 14.12 MB, Log files Used: 504.00 kB, Log Files total: 504.00 kB
OK	SQLServer msdb Transactions		OK - Transactions: 0.0/s, Tracked Transactions: 0.0/s, Write Transactions: 0.0/s
OK	SQLServer mssqlsystemresource File Sizes	 	OK - Data Files: 40.00 MB, Log files Used: 438.00 kB, Log Files total: 1.24 MB
OK	SQLServer mssqlsystemresource Transactions		OK - Transactions: 0.0/s, Tracked Transactions: 0.0/s, Write Transactions: 0.0/s
OK	SQLServer tempdb File Sizes	 	OK - Data Files: 8.00 MB, Log files Used: 355.00 kB, Log Files total: 504.00 kB
OK	SQLServer tempdb Transactions		OK - Transactions: 0.0/s, Tracked Transactions: 0.0/s, Write Transactions: 0.0/s
OK	SQLServer:Buffer_Manager None buffer_cache_hit_ratio	 	OK - 100%
OK	SQLServer:Catalog_Metadata _Total cache_hit_ratio	 	OK - 66%
OK	SQLServer:Catalog_Metadata master cache_hit_ratio	 	OK - 72%
OK	SQLServer:Catalog_Metadata model cache_hit_ratio	 	OK - 0%
OK	SQLServer:Catalog_Metadata msdb cache_hit_ratio	 	OK - 28%
OK	SQLServer:Catalog_Metadata mssqlsystemresource cache_hit_ratio	 	OK - 74%
OK	SQLServer:Catalog_Metadata tempdb cache_hit_ratio	 	OK - 60%
OK	SQLServer:Locks _Total Locks		OK - Requests: 5.9/s, Timeouts: 0.0/s, Deadlocks: 0.0/s, Waits: 0.0/s

Note: Since version 1.2.8p13, Microsoft Sql Server 2016 is also supported

Check_MK - Werks

The software development of Check_MK is organized in so called *Werks*. A Werk is any change or bug fix that has influence on the user's experience. Each Werk has a unique ID, one of the levels *Trivial Change*, *Prominent Change* or *Major Feature* and one of the classes *Bug Fix*, *Feature* or *Security Fix*.

Whenever you make an update to a new Check_MK version please make sure that you have understood all incompatible changes. You might have to adapt your configuration.

If you like to get informed about new werks, you can subscribe to [various mailinglists](#) which inform you about werks of specific levels.

Edition: Branch: Till:

Show only incompatible werks

Version 1.2.8p13

#3967 Checks & Agents: [rاران emx](#): Fixed broken check (wrong include temperature.includes)

Trivial Change

Bug Fix

#3937 Checks & Agents: [check_mk_agent.aix](#): fixed handling of mailq command

Trivial Change

Bug Fix

#3960 Checks & Agents: [if.include](#): fixed wrong order if interface groups are configured

Trivial Change

Bug Fix

#3903 Checks & Agents: [mssql.vbs](#): Fixed support for MSSQL server 2016

Trivial Change

Bug Fix

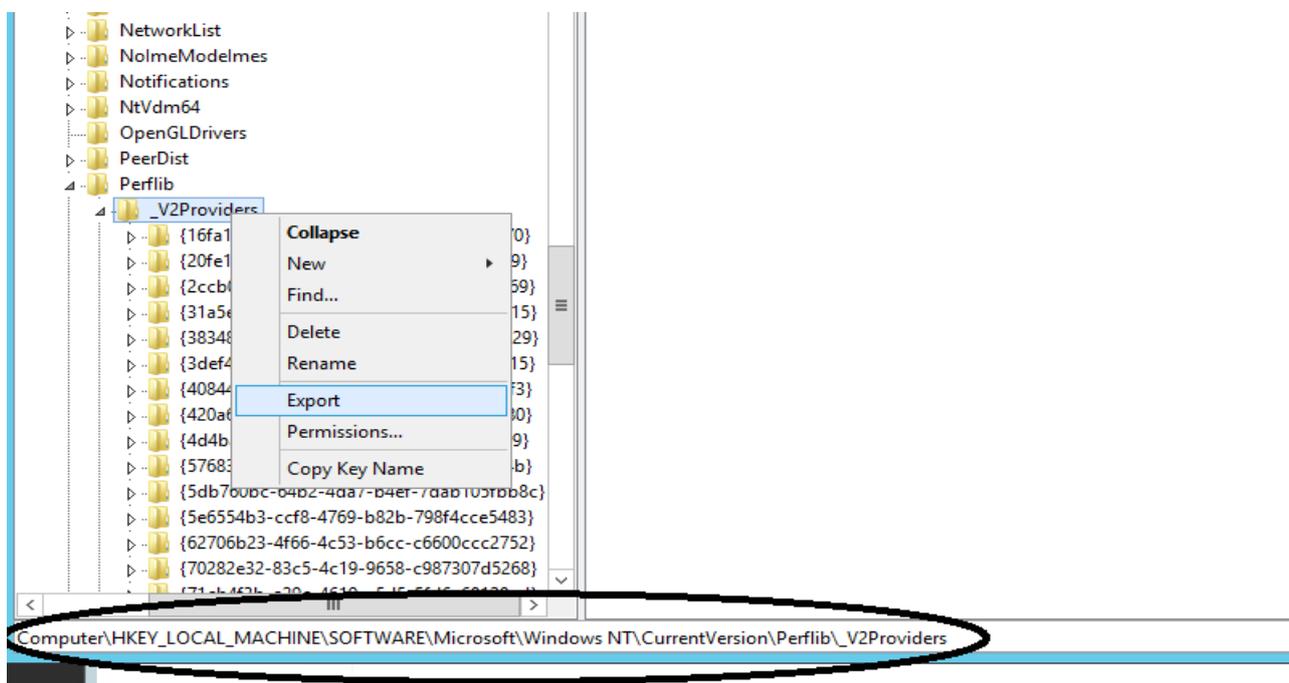
Microsoft Terminal Services

Several Windows checks are based on *Performance Counters*. These are special objects provided by the Windows operating system that contain information about throughput, queue lengths, latencies and other numbers of the system and applications like MS Exchange, MSSQL, IIS etc

Because there is no native support for Terminal Services, we need to take advantage of Performance Counters. I fought a little bit with this task but, thanks again to the mailing list, I was able to do it in this way:

- On the windows host run *regedit* and export the following key:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\_V2Providers]
```



- Open the file and search the string *Terminal Services*

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows  
NT\CurrentVersion\Perflib\_V2Providers\{f3b975e7-e068-4f66-81ef-b23e0a0e64c9}\{fc9e399c-  
c70a-4458-8430-ca249c371eb3}]  
"NameResource"=dword:00000001  
"ExplainResource"=dword:00000003  
"NeutralName"="Terminal Services"  
"InstanceType"=dword:00000000  
"First Counter"=dword:00000780  
"Last Counter"=dword:00000786  
"CounterBlock"=hex:01,00,00,00,00,00,01,00,01,00,00,00,00,00,00,00,64,00,00,00,\  
00,00,00,00,05,00,00,00,07,00,00,00,00,00,00,00,ff,ff,ff,ff,ff,ff,ff,ff,\  
.....
```

- Take the hexadecimal value of "First Counter" and convert in decimal. In this case:

00000780 = 1920

- Edit *check_mk.ini* and add the following string in the *winperf* section

```
[winperf]
# Select counters to extract. The following counters
# are needed by checks shipped with check_mk.
# counters = 10332:msx_queues
# counters = 638:tcp_conn
counters = 1920:ts_sessions
```

- Restart the agent
- Do a service discovery adding unmonitored services

OK	Processor Queue			OK - 15 min load 5.54
OK	Service Dnscache			OK - DNS Client: running (start type is auto)
OK	Service TermService			OK - Remote Desktop Services: running (start type is
OK	Services Summary			OK - 156 services, 53 services in autostart - of which : spnsvc, TrustedInstaller). 0 services stopped but ignor
OK	Sessions			OK - 1 Active, 2 Inactive
OK	SQLServer _Total File Sizes			OK - Data Files: 68.31 MB, Log files Used: 2.26 MB, L
OK	SQLServer _Total Transactions			OK - Transactions: 0.0/s, Tracked Transactions: 0.0/s,
OK	SQLServer master File Sizes			OK - Data Files: 4.00 MB, Log files Used: 609.00 kB, L
OK	SQL Server master Transactions			OK - Transactions: 0.0/s, Tracked Transactions: 0.0/s



Network Devices

Network devices (switches, router, firewall, balancer etc.) are monitored using the SNMP protocol. SNMP uses UDP as its transport protocol. If management traffic needs to traverse firewalls, make sure that the following default ports are open:

- UDP 161: Used when management stations communicate with agents, e.g. Polling
- UDP 162: Used when agents send unsolicited Traps to the management station

During the wizard, please be sure to select *SNMP (Networking device, Appliance)* in the Agent type combo box.

The screenshot shows a configuration wizard interface with the following details:

- Folder:** Main directory
- General Properties:** Hostname: Switch_10.39.238.28
- Basic settings:**
 - Permissions: empty (Default value)
 - Alias: empty (Default value)
 - IPv4 Address: 10.39.238.28
 - SNMP Community: [Masked]
 - Parents: empty (Default value)
 - Monitored on site: skytest - Local site skytest (Default value)
- Host tags:**
 - Agent type: SNMP (Networking device, Appliance) [Selected]
 - Criticality: Productive system (Default value)
 - Networking Segment: Local network (low latency) (Default value)
 - IP Address Family: IPv4 only (Default value)
- Buttons:** Save & go to Services, Save & Finish, Save & Test [Highlighted]

Successfully created the host. Now you should do a **service discovery** in order to auto-configure all services to be checked on this host.

Host Properties

Hostname
Switch_10.39.238.28

IP address
10.39.238.28

SNMP Community

Save & Exit

Options

Check_MK Agent Port (Rules)
6556

SNMP-Timeout (Rules)
1 sec

SNMP-Retries (Rules)
5

Datasource Program (Rules)

Ping

```

✓ PING 10.39.238.28 (10.39.238.28) 56(84) bytes of data.
64 bytes from 10.39.238.28: icmp_seq=1 ttl=255 time=1.27 ms
64 bytes from 10.39.238.28: icmp_seq=2 ttl=255 time=0.689 ms

--- 10.39.238.28 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 200ms
rtt min/avg/max/mdev = 0.689/0.983/1.277/0.294 ms, ipg/ewma 200.526/1.203 ms

```

Agent

```

✗ Cannot get data from TCP port 10.39.238.28:6556: [Errno 111] Connection refused

```

SNMPv1

```

✓ sysDescr: Cisco IOS Software, C3560 Software (C3560-IPSERVICESK9-M), Version RELEASE SOFTWARE (fc3) Technical Support: http://www.cisco.com/techsupport C1986-2011 by Cisco Systems, Inc. Compiled Thu 22-Dec-11 00:24 by prod_rel_te
sysContact:
sysName: msgpdl_sw1.technology.milano.it
sysLocation:

```

SNMPv2c

```

✓ sysDescr: Cisco IOS Software, C3560 Software (C3560-IPSERVICESK9-M), Version RELEASE SOFTWARE (fc3) Technical Support: http://www.cisco.com/techsupport C1986-2011 by Cisco Systems, Inc. Compiled Thu 22-Dec-11 00:24 by prod_rel_te

```

Click on *Service Discover*, *Save manual check configuration*

Raw 2.8p9

Services of host Switch_10.39.238.28 (might be cached data)

Folder Status Properties Parameters Diagnostic Full

Activate missing **Save manual check configuration** Automatic Refresh (Tabula Rasa) Show Check Parameters

Available (missing) services				
Status	Checkplugin	Item	Service Description	Plugin output
OK	cisco_cpu	None	CPU utilization	8.0% utilization in the last 5 minutes
OK	cisco_fan	Switch 1 Fan 1	FAN Switch 1 Fan 1	State is: normal (1)
OK	cisco_mem	I/O	Mem used I/O	43.3% (3.46 MB) of 8.00 MB used
OK	cisco_mem	Processor	Mem used Processor	42.6% (26.20 MB) of 61.57 MB used
OK	cisco_power	Sw1	Power Sw1	state: normal, source: AC
OK	cisco_temperature	SW 1 Sensor 1	Temperature SW 1 Sensor 1	45 °C
OK	if64	10101	Interface 10101	[GigabitEthernet0/1] (up) MAC: 00:1c:b1:33:30:81, 1 Gbit/s
OK	if64	10102	Interface 10102	[GigabitEthernet0/2] (up) MAC: 00:1c:b1:33:30:82, 1 Gbit/s
OK	if64	10103	Interface 10103	[GigabitEthernet0/3] (up) MAC: 00:1c:b1:33:30:83, 100 Mbit/s
OK	if64	10104	Interface 10104	[GigabitEthernet0/4] (up) MAC: 00:1c:b1:33:30:84, 1 Gbit/s
OK	if64	10105	Interface 10105	[GigabitEthernet0/5] (up) MAC: 00:1c:b1:33:30:85, 1 Gbit/s

As usual apply the changes and wait a while to have the new device appear.

The screenshot shows the Nagios 'All hosts' page. At the top, there are navigation icons and a search bar. Below that, the 'Local site skytest' section displays a table of hosts. The table has columns for 'state', 'Host', 'Icons', 'OK', 'Wa', 'Un', 'Cr', and 'Pd'. The first row shows 'localhost' with a state of 'UP' and OK count of 22. The second row shows 'Switch_10.39.238.28' with a state of 'UP' and OK count of 52. The number 52 is circled in black. Below the table, there are icons for host status and a tooltip showing the IP address '10.39.238.28'.

The screenshot shows the Nagios 'Services of Host Switch_10.39.238.28' page. At the top, there are navigation icons and a search bar. Below that, the 'Switch_10.39.238.28' section displays a table of services. The table has columns for 'State', 'Service', 'Icons', 'Status detail', 'Age', 'Checked', and 'Perf.-O-Meter'. The services listed are:

State	Service	Icons	Status detail	Age	Checked	Perf.-O-Meter
OK	Check_MK		OK - execution time 1.0 sec	50 sec	50 sec	1.03 s
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found	3 min	3 min	
OK	CPU utilization		OK - 8.0% utilization in the last 5 minutes	50 sec	50 sec	8%
OK	FAN Switch 1 Fan 1		OK - State is: normal (1)	50 sec	50 sec	
OK	Interface 10101		OK - [GigabitEthernet0/1] (up) MAC: 00:1c:b1:33:30:81, 1 Gbit/s	50 sec	50 sec	
OK	Interface 10102		OK - [GigabitEthernet0/2] (up) MAC: 00:1c:b1:33:30:82, 1 Gbit/s	50 sec	50 sec	
OK	Interface 10103		OK - [GigabitEthernet0/3] (up) MAC: 00:1c:b1:33:30:83, 100 Mbit/s	50 sec	50 sec	
OK	Interface 10104		OK - [GigabitEthernet0/4] (up) MAC: 00:1c:b1:33:30:84, 1 Gbit/s	50 sec	50 sec	
OK	Interface 10105		OK - [GigabitEthernet0/5] (up) MAC: 00:1c:b1:33:30:85, 1 Gbit/s	50 sec	50 sec	

In this case a *CRITICAL* service will fire up in case of hardware failure and, depending on the check parameters, *WARN* or *CRIT* when the port status changes (i.e. is down), when the link speed changes (e.g. a port expected to be set to 1Gbit/s operates only at 100Mbit/s), when the absolute or percentage traffic of a port exceeds certain levels or if the rate of errors or discards exceeds configurable limits.

By default, Check_MK doesn't inventory Port-Channels. Port-Channels are aggregated physical interfaces which are usually used for inter-switch connectivity. After a Google search, I found a post explaining how to fix that: https://sitweak.wordpress.com/2012/08/16/monitoring-port-channel-on-cisco-switchesrouters-with-check_mk/

I don't understand the reason behind that choice - in my opinion the default should be to always monitor.

Rule-Based Configuration of Host & Service Parameters

No Changes Main Menu Used Rulesets Ineffective rules Dep

Main directory

Search for rule sets:

- Active checks (HTTP, TCP, etc.)**
Configure active networking checks like HTTP and TCP
- Grouping**
Assignment of host & services to host, service and contacts groups.
- Access to Agents**
Settings concerning the connection to the Check_MK and SNMP agents
- Parameters for discovered services**
Levels and other parameters for checks found by the Check_MK service discovery.
- Hardware/Software-Inventory**
Configuration of the Check_MK Hardware and Software Inventory System
- Event Console**
Settings and Checks dealing with the Check_MK Event Console

Hardware/Software-Inventory

No Changes Main Menu All Rulesets Folder

Main directory

▼ Hardware/Software-Inventory
Do hardware/software Inventory 1 Export List of Software packages as CSV file 0 **Parameters for switch port inventory 0**

Raw
2.8p9

Parameters for switch port inventory

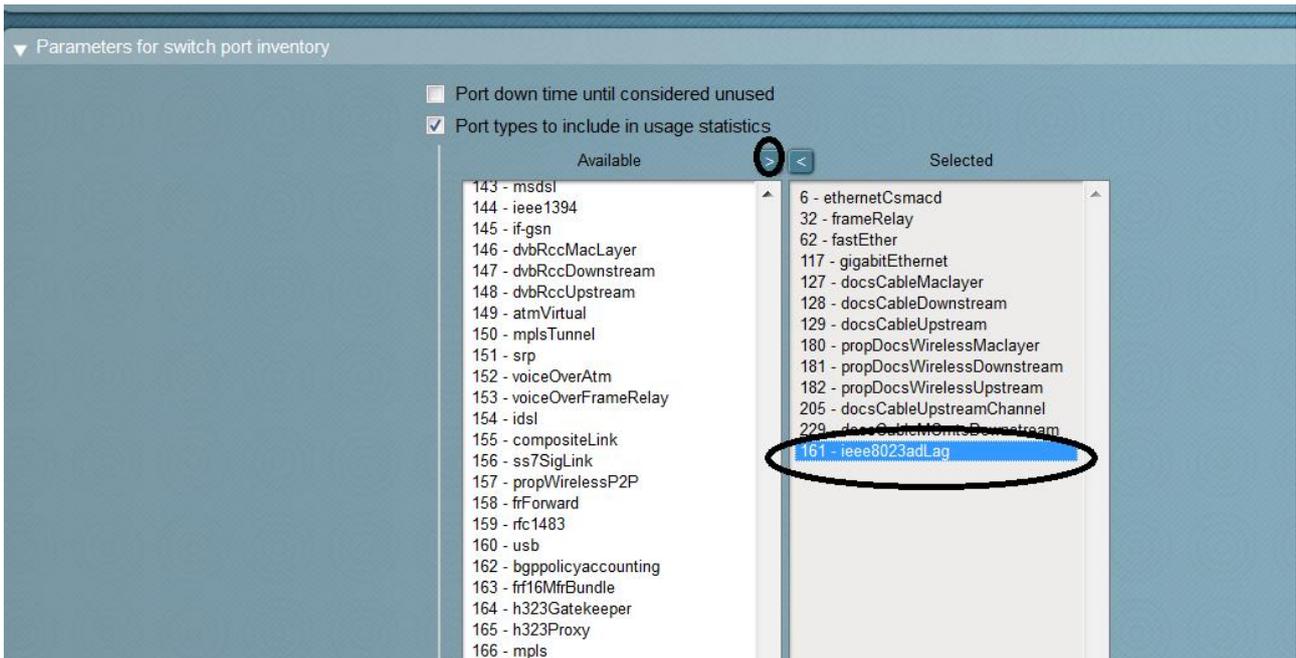
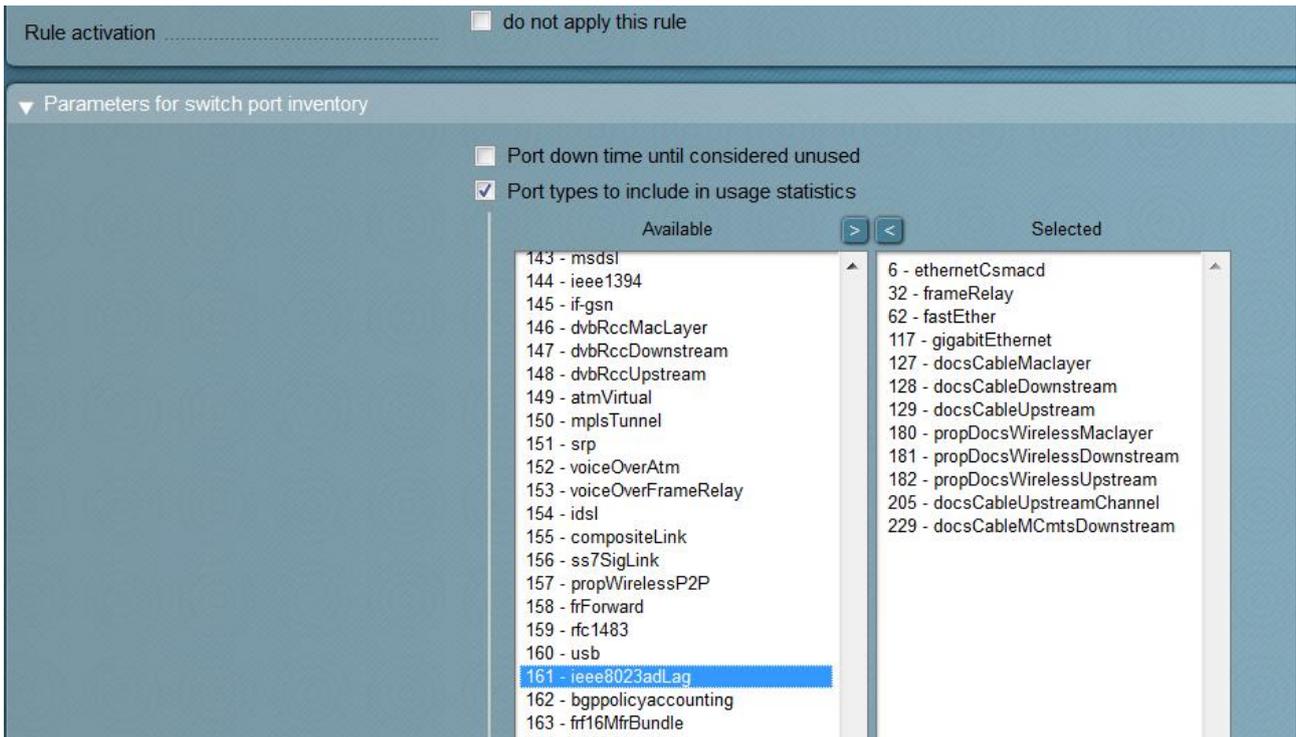
No Changes Main Menu Hardware/Software-In... Used Rulesets

Main directory

Matching: Each parameter is defined by the first matching rule where that parameter is set (checked).

There are no rules defined in this set.

Create rule in folder: ▼



Click Save and Activate changes

Managing Thresholds

A threshold is a range with an alert level, either warning or critical. The theory is that the plugin will do some sort of check which returns back a numerical value, or metric, which is then compared to the warning and critical thresholds. To avoid useless alerts, I suggest to define a certain number of *check attempts* before to send out alarms and notifications. For example: CPU spikes are quite usual and normal thus it would be useful to be notified only when its consumption is too much high for more than a specified time period.

In this example, a CPU threshold is setup so that a *CRITICAL* service will be created only if the percentage of CPU utilization is above 90 % for more than 5 minutes.

WATO, Host & Service Parameters, Parameters for discovered services, CPU utilization on Linux/UNIX

Parameters for discovered services

Main directory: [v]

- ▶ Networking
- ▶ Discovery - automatic service detection
- ▶ Applications, Processes & Services
- ▶ Temperature, Humidity, Electrical Parameters, etc.
- ▶ Storage, Filesystems and Files
- ▼ Operating System Resources

APT Updates	0	CPU load (not utilization!)	0	CPU utilization for Appliances	0
CPU utilization for simple devices	0	CPU utilization of Devices with Modules	0	CPU utilization on Linux/UNIX	1
Cisco Memory Usage	0	Cisco Nexus Supervisor Memory Usage	0	FPGA utilization	0
Flash Space Usage	0	Innovaphone Memory Usage	0	Juniper Memory Usage	0
Juniper Modules Memory Usage	0	Main memory usage (UNIX / Other Devices)	0	Main memory usage of ESX host system	0
Main memory usage of devices with modules	0	Main memory usage of simple devices	0	Memory Pages Statistics	0
Memory and Swap usage on Arbor devices	0	Memory and Swap usage on Linux	0	Memory and pagefile levels for Windows	0
Netscaler Memory Usage	0	Number of Logins on System	0	Number of kernel events per second	0
Number of threads	0	Number of used states of OpenBSD PF engine	0	State of NTP peer	0
State of NTP time synchronisation	0	Statgrab Memory Usage	0	Storage Processor Utilization	0
Uptime since last reboot	0	Virtual machine (for example ESX) guest tools status	0	Virtual machine (for example ESX) heartbeat status	0
Windows system time offset	0				

Raw 1.2.8p11

CPU utilization on Linux/UNIX

Main directory: [v]

Matching: Each parameter is defined by the first matching rule where that parameter is set (checked).

There are no rules defined in this set.

Monitoring Configuration

1 Changes Main Menu All RuleSets Folder

Main directory

Service Checks

Check period for active services	0	Check period for passive Check_MK services	0	Enable/disable active checks for services	0
Enable/disable passive checks for services	1	Enable/disable processing of perdata for services	0	Maximum number of check attempts for service	0
Normal check interval for service checks	1	Retry check interval for service checks	0		

Host Checks

Check period for hosts	0	Host Check Command	0	Maximum number of check attempts for host	0
Normal check interval for host checks	0	Retry check interval for host checks	0		

Notifications

Inventory and Check_MK settings

Clustered services	0	Disabled checks	0	Disabled services	0
Hosts to be monitored	0	Periodic service discovery	0		

Various

Clustered services for overlapping clusters	0	Custom icons or actions for hosts in status GUI	0	Custom icons or actions for services in status GUI	0
Icon image for hosts in status GUI	0	Icon image for services in status GUI	0	Service period for hosts	0
Service period for services	0				

The maximum number of failed checks until a service is hard. Only hard state trigger notifications.

Maximum number of check attempts for service

1 Changes Main Menu Monitoring Configura... Used RuleSets

Main directory

Matching: The first matching rule defines the parameter.
There are no rules defined in this set.

Create rule in folder: Linux

Icons: [Sun] [Green Arrow]

New rule Maximum number of check attempts for service

Abort

The maximum number of failed checks until a service problem state will be considered as hard. Only hard state trigger notifications.

Rule Options

Description: Max Check Attempt for Linux Cpu Utilization

Comment: [Text Area]

Documentation-URL: [Text Field]

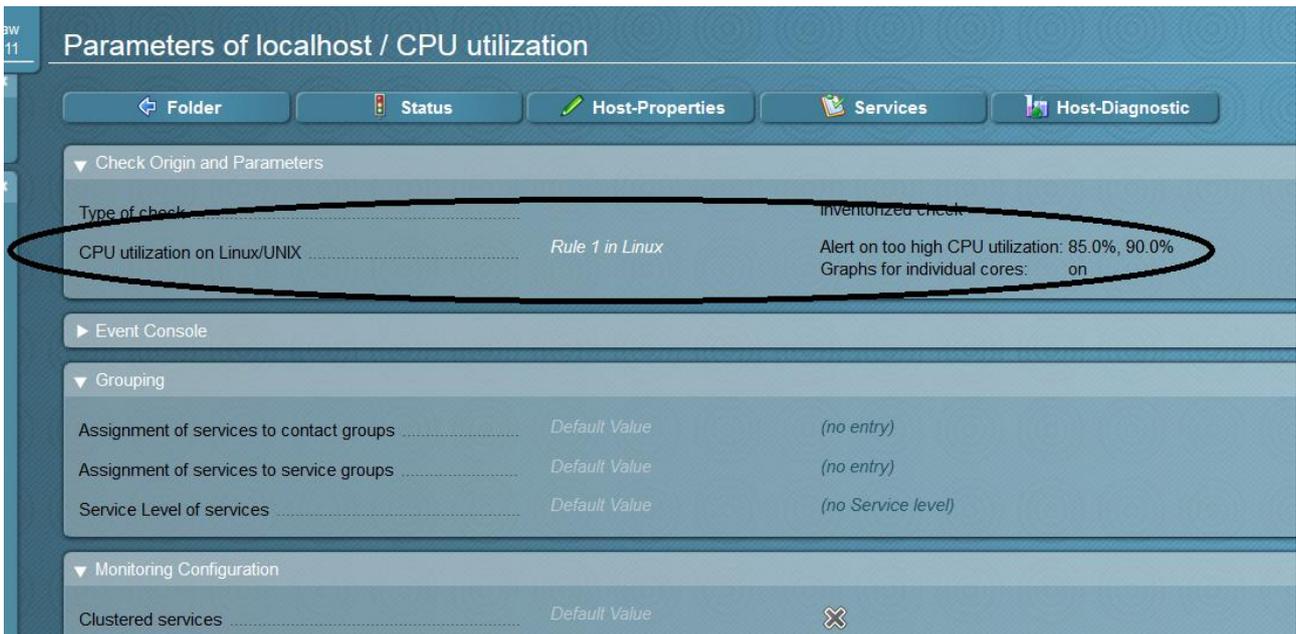
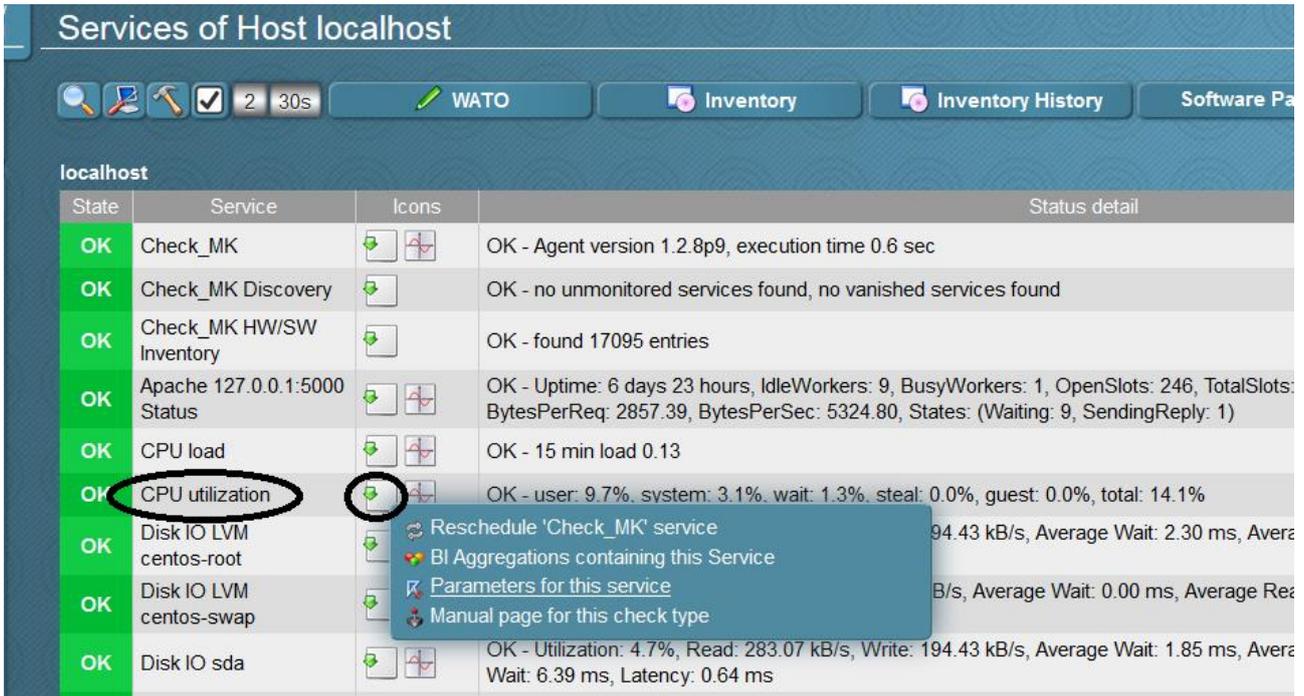
Rule activation: do not apply this rule

Maximum number of check attempts for service: 5

Conditions

Folder: Linux

To check if the rule has been applied, an easy way is to choose a server and look for the "CPU Utilization" service parameters



Let's do some testing using the *stress* utility

```
[root@checkmktst1 ~]# stress --cpu 8 --timeout 600
stress: info: [12082] dispatching hogs: 8 cpu, 0 io, 0 vm, 0 hdd
```

After 5 minutes, the service should be in CRITICAL state

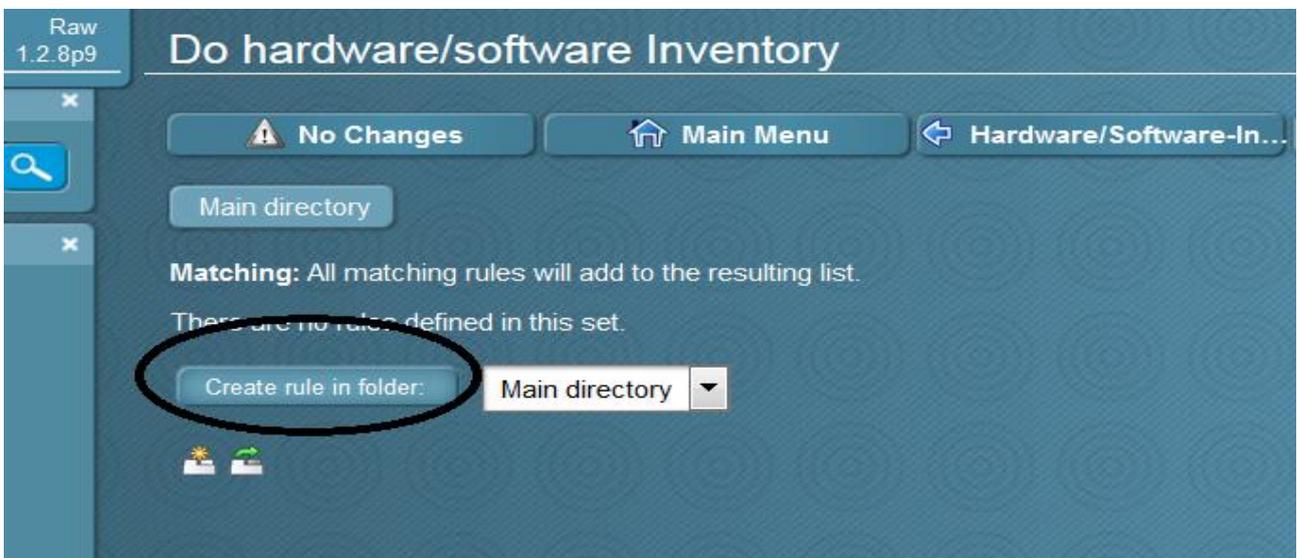
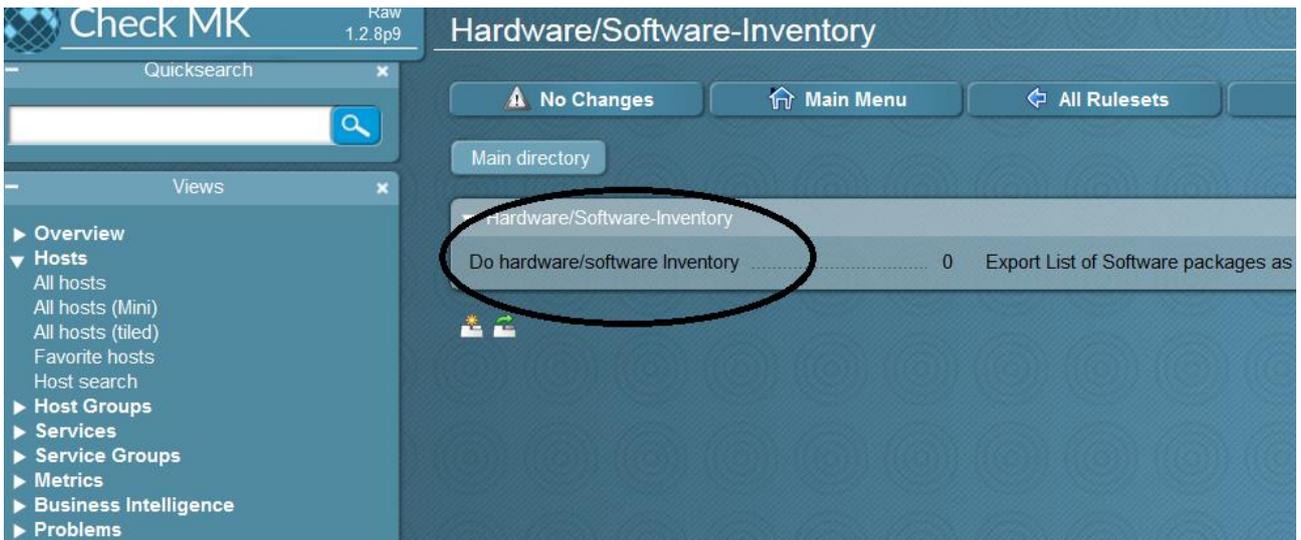
Services of Host localhost							30 rows	omdadmin (admin)	14:54
State	Service	Icons	Status detail	Age	Checked	Perf-O-Meter			
OK	Check_MK		OK - Agent version 1.2.8p9, execution time 3.0 sec	2 hrs	60 sec	3.03 s			
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found	2 hrs	17 min				
OK	Check_MK HW/SW Inventory		OK - found 17095 entries	2 hrs	2 hrs				
OK	Apache 127.0.0.1:5000 Status		OK - Uptime: 6 days 23 hours, IdleWorkers: 9, BusyWorkers: 1, OpenSlots: 246, TotalSlots: 256, CPUload: 0.01, ReqPerSec: 0.10, BytesPerReq: 2868.65, BytesPerSec: 699.73, States: (Waiting: 9, SendingReply: 1)	2 hrs	57 sec	7 d			
OK	CPU load		OK - 15 min load 3.40	2 hrs	57 sec	8.71			
CRIT	CPU utilization		CRIT - user: 98.4%, system: 1.6%, wait: 0.0%, steal: 0.0%, guest: 0.0%, total: 100.0% (warn/crit at 85.0%/90.0%) CRIT	6 min	57 sec	100%			
OK	Disk IO LVM centos-root		OK - Utilization: 3.5%, Read: 12.66 kB/s, Write: 134.26 kB/s, Average Wait: 7.34 ms, Average Read Wait: 5.00 ms, Average Write Wait: 7.72 ms, Latency: 1.56 ms	2 hrs	57 sec	13 kB/s / 134 kB/s			
OK	Disk IO LVM centos-swap		OK - Utilization: 0.0%, Read: 0.00 B/s, Write: 0.00 B/s, Average Wait: 0.00 ms, Average Read Wait: 0.00 ms, Average Write Wait: 0.00 ms, Latency: 0.00 ms	2 hrs	57 sec	0.00 B/s / 0.00 B/s			
OK	Disk IO sda		OK - Utilization: 3.5%, Read: 12.66 kB/s, Write: 134.26 kB/s, Average Wait: 6.18 ms, Average Read Wait: 5.00 ms, Average Write Wait: 6.39 ms, Latency: 1.64 ms	2 hrs	57 sec	13 kB/s / 134 kB/s			

Hardware & Software Inventory

Check_MK supports hardware & software inventories. While SNMP devices don't require any additional components, for Windows & Linux devices we need a plugin.

The first step is to enable *Hardware/Software-Inventory* by creating a rule:

The screenshot displays the Check_MK configuration interface. On the left, a navigation menu lists various categories: All Hosts (mini), All hosts (tiled), Favorite hosts, Host search, Host Groups, Services, Service Groups, Metrics, Business Intelligence, Problems, Event Console, Inventory, and Other. The 'Inventory' category is selected. Below the menu, there are sections for 'Bookmarks' and 'WATO · Configuration'. The 'WATO · Configuration' section includes links for Main Menu, Hosts, Host Tags, Global Settings, Host & Service Parameters (circled in red), and Manual Checks. The main area shows several configuration cards: 'Active checks (HTTP, TCP, etc.)', 'Access to Agents', 'Hardware/Software-Inventory' (circled in black), 'Grouping', 'Parameters for...', and 'Event Console'. The 'Hardware/Software-Inventory' card is highlighted, indicating it is the current configuration point.



✖ Abort

All hosts configured via this ruleset will do a hardware and software inventory. For each configured host a new active check will be created. You should also create a rule between a couple of hours and one day. **Note:** in order to get any useful result for agent based hosts make sure that you have installed the agent plugin `mk_inventory`.

▼ Rule Options

Description Hardware & Software Inventory

Comment Hardware & Software Inventory

Documentation-URL

Rule activation do not apply this rule

▼ Do hardware/software Inventory

State when software changes are detected

State when hardware changes are detected

State when inventory fails

▼ Conditions

Folder Main directory ▼

Host tags Agent type: ignore ▼

Click *Save* and remember to apply changes.

Now it's time to install the plugin for both Linux and Windows server:

Linux:

-Copy the "mk_inventory" plugin in the "local" folder of the linux agent. In my case the path is:

```
/usr/lib/check_mk_agent/local/mk_inventory
```

Make sure it is executable

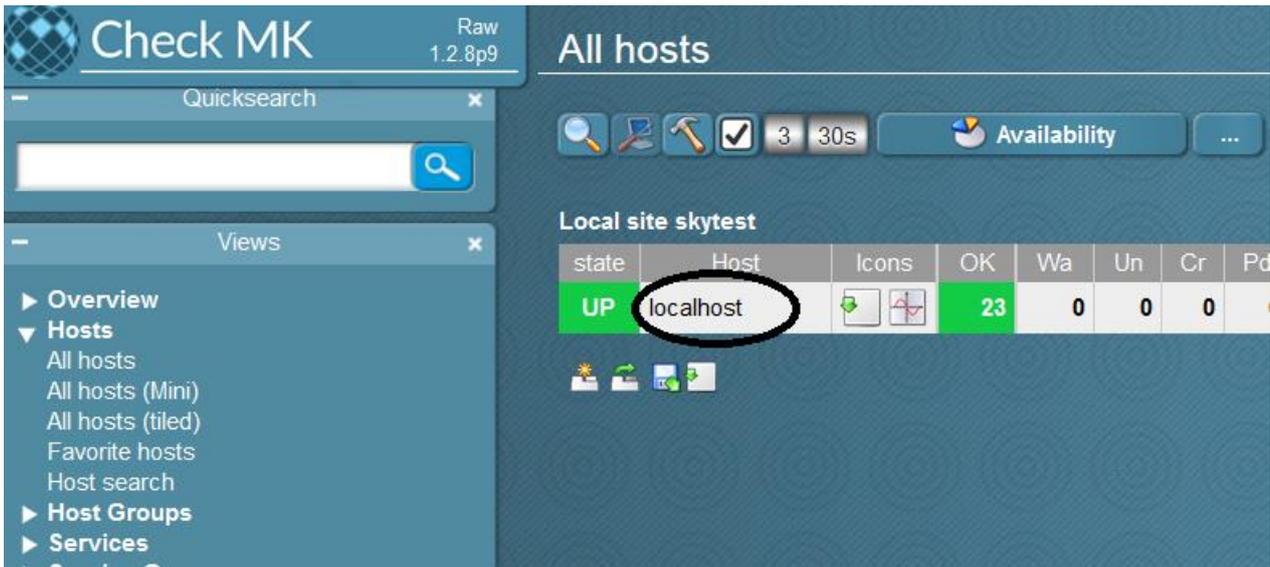
```
chmod +x /usr/lib/check_mk_agent/local/mk_inventory
```

If you are not sure about it, you can check it by simply running the agent from the command line and checking the output which should show the current configuration:

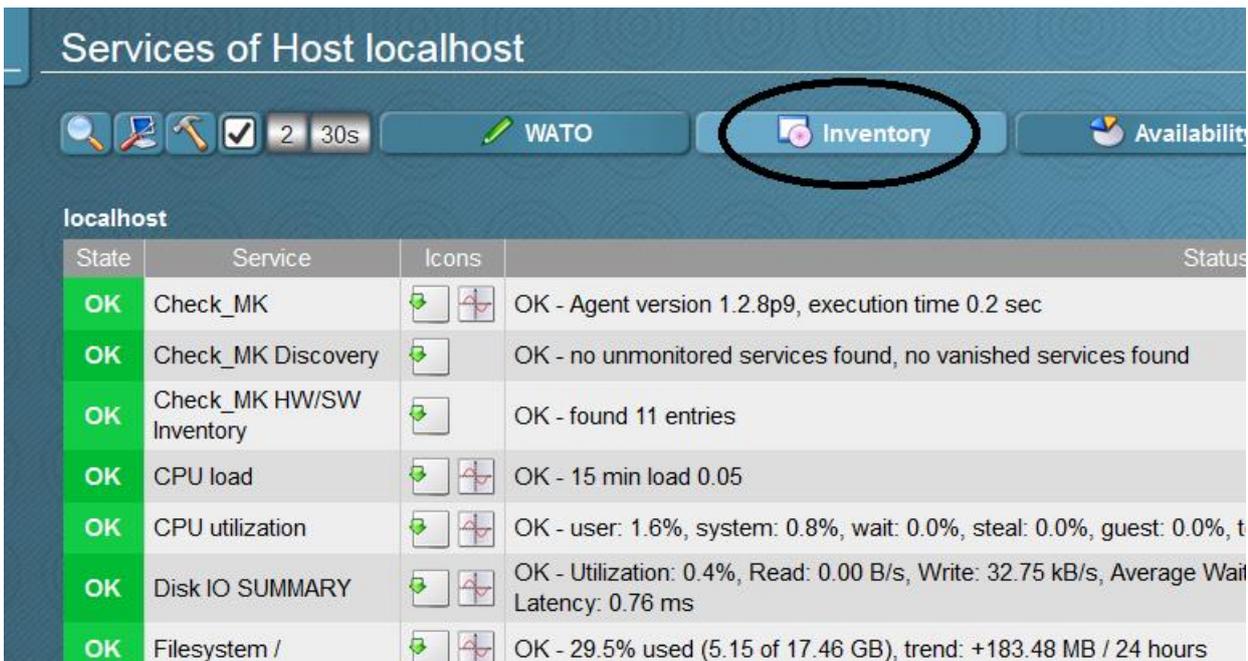
```
OMD[mysite]:~$ /usr/bin/check_mk_agent
<<<check_mk>>>
Version: 1.2.8p9
AgentOS: linux
Hostname: checkmktst1
AgentDirectory: /etc/check_mk
DataDirectory: /var/lib/check_mk_agent
SpoolDirectory: /var/lib/check_mk_agent/spool
PluginsDirectory: /usr/lib/check_mk_agent/plugins
LocalDirectory: /usr/lib/check_mk_agent/local
.....
```

-Force inventory on check_mk server:

OMD[mysite]:~\$ cmk -i



The screenshot shows the Check MK 'All hosts' overview page. The top left features the 'Check MK' logo and version 'Raw 1.2.8p9'. Below it is a 'Quicksearch' bar. A sidebar on the left lists navigation options: Overview, Hosts (expanded), All hosts, All hosts (Mini), All hosts (tiled), Favorite hosts, Host search, Host Groups, and Services. The main content area is titled 'All hosts' and includes a toolbar with icons for search, edit, hammer, checkmark, and a '3 30s' timer. An 'Availability' button is also present. Below the toolbar is a table titled 'Local site skytest' with columns: state, Host, Icons, OK, Wa, Un, Cr, and Pd. The 'localhost' entry is circled in black, showing a state of 'UP' and '23' OK services. Below the table are several small status icons.



The screenshot shows the 'Services of Host localhost' page. The title is 'Services of Host localhost'. The toolbar includes search, edit, hammer, checkmark, a '2 30s' timer, a 'WATO' button, an 'Inventory' button (circled in black), and an 'Availability' button. Below the toolbar is a table with columns: State, Service, Icons, and Status. The table lists several services, all with an 'OK' state. The 'Inventory' service is highlighted in green.

State	Service	Icons	Status
OK	Check_MK	 	OK - Agent version 1.2.8p9, execution time 0.2 sec
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found
OK	Check_MK HW/SW Inventory		OK - found 11 entries
OK	CPU load	 	OK - 15 min load 0.05
OK	CPU utilization	 	OK - user: 1.6%, system: 0.8%, wait: 0.0%, steal: 0.0%, guest: 0.0%, t
OK	Disk IO SUMMARY	 	OK - Utilization: 0.4%, Read: 0.00 B/s, Write: 32.75 kB/s, Average Wait Latency: 0.76 ms
OK	Filesystem /	 	OK - 29.5% used (5.15 of 17.46 GB), trend: +183.48 MB / 24 hours

Hostname: localhost

Inventory Tree

- Hardware
 - BIOS
 - Chassis
 - Processor
 - Memory (RAM)
 - System

Manufacturer	VMware, Inc.
Product	VMware Virtual Platform
Serial Number	VMware-56 4d fe 15 f2 57 70 68-90 2a 67 09 86 10 91 d2
Uuid	564DFE15-F257-7068-902A-6709861091D2
Version	None
- Networking
 - Hostname: checkmktst1
 - IP Addresses
 - Routes
- Software
 - Operating System

Code Name	Core
Kernel Architecture	x86_64
Kernel Version	3.10.0-327.28.2.el7.x86_64
Name	CentOS Linux release 7.2.1511
Type	linux
Version	7.2.1511
 - Packages

Name	Version	Architecture	Type	Description
ebtables	2.0.10	x86_64	rpm	Ethernet Bridge frame table administration tool
openssh	6.6.1p1	x86_64	rpm	An open source implementation of SSH protocol versions 1 and 2
hunspell-en-GB	0.20121024	noarch	rpm	UK English hunspell dictionaries
alsa-plugins-pulseaudio	1.0.27	x86_64	rpm	Alsa to PulseAudio backend
xorg-x11-server-Xorg	1.17.2	x86_64	rpm	Xorg X server
emacs-filesystem	24.3	noarch	rpm	Emacs filesystem layout
openssh-clients	6.6.1p1	x86_64	rpm	An open source SSH client applications
libfreehand	0.1.1	x86_64	rpm	A library for import of Macromedia/Adobe FreeHand documents
hwloc	1.7	x86_64	rpm	Portable Hardware Locality - portable abstraction of hierarchical architecture
python-six	1.9.0	noarch	rpm	Python 2 and 3 compatibility utilities
libspectre	0.2.7	x86_64	rpm	A library for rendering PostScript(TM) documents

Windows:

-Copy the script "mk_inventory.vbs" in the "local" directory of Check_MK agent. In my case it was C:\Program Files (x86)\check_mk\local\mk_inventory.vbs.

-Restart the windows service

net stop check_mk_agent && net start check_mk_agent

Force inventory on the server side:

cmk -i

Click on the windows host to check what has been discovered:

state	Host	Icons	OK	Wa	Un	Cr	Pd	state	Host	Icons	OK	Wa
UP	Switch_10.39.238.28		53	0	0	0	0	UP	w2012tst1		21	0

10.39.239.101

Services of Host w2012tst1

WATO **Inventory** Availability

State	Service	Icons	Status detail
OK	Check_MK		OK - Agent version 1.2.8p8, execution time 0.1 sec
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found
OK	Check_MK HW/SW Inventory		OK - found 9 entries
OK	CPU utilization		OK - 0.2% used, user perc: 0.1 %, privileged perc: 0.1 %, 1 CPUs
OK	Disk IO SUMMARY		OK - Read: 0.00 B/s, Write: 4.31 kB/s, Average Read Wait: 0.00 ms, Average Write Wait: 0.11 ms
OK	DotNet Memory Management _Global_		OK - 0.00% time in GC
OK	Filesystem C:/		OK - 17.0% used (10.14 of 59.66 GB), trend: +737.85 kB / 24 hours
OK	Interface 1		OK - [Intel(R) 82574L Gigabit Network Connection] (Connected) 1 Gbit/s, in: 112.41 B/s(0.0%), out: 483.26 B/s(0.0%)
OK	Interface 2		OK - [Intel(R) 82574L Gigabit Network Connection 2] (Connected) 1 Gbit/s, in: 333.21 B/s(0.0%), out: 10.16 B/s(0.0%)

Inventory Tree

- Hardware
 - BIOS
 - Processor
 - Memory (RAM)
 - Total swap space: 384.00 MB
 - Total usable RAM: 2.00 GB
 - Storage
 - System
 - Graphic Cards
- Networking
 - Hostname: w2012tst1
- Software
 - Operating System
 - Packages

Open this table for filtering

Name	Version	Architecture	Type	Description	Install Date	Language	Path	Publisher
Microsoft Visual C++ 2008 Redistributable -x86 9.0.30729.6161	9.0.30729.6161		registry	Microsoft Visual C++ 2008 Redistributable -x86 9.0.30729.6161	2016-08-08	1033		Microsoft Corporation
Microsoft Silverlight	5.1.50428.0		registry	Microsoft Silverlight	2016-08-09	1033	c:\Program Files\Microsoft Silverlight\	Microsoft Corporation
VMware Tools	10.0.6.3560309		registry	VMware Tools	2016-08-08	1040	C:\Program Files\VMware\VMware Tools\	VMware, Inc.
check_mk_agent			registry	Check_MK Agent 1.2.8p8				
Mozilla Firefox 48.0 (x86 it)	48.0		registry	Mozilla Firefox 48.0 (x86 it)			C:\Program Files (x86)\Mozilla Firefox	Mozilla
MozillaMaintenanceService	48.0		registry	Mozilla Maintenance Service				Mozilla
Microsoft Visual C++ 2008 Redistributable -x86 9.0.30729.4148	9.0.30729.4148		registry	Microsoft Visual C++ 2008 Redistributable -x86 9.0.30729.4148	2016-08-08	1033		Microsoft Corporation
Microsoft Visual C++ 2008 Redistributable -x86 9.0.30729.6161	9.0.30729.6161		registry	Microsoft Visual C++ 2008 Redistributable -x86 9.0.30729.6161	2016-08-09	1033		Microsoft Corporation
VMware Tools	10.0.6.3560309		wmi		2016-08-08	1040		VMware, Inc.

Cisco:

Views

- Overview
- Hosts
 - All hosts**
 - All hosts (Mini)
 - All hosts (tiled)
 - Favorite hosts
 - Host search
- Host Groups
- Services
 - Service Groups
 - Metrics
 - Business Intelligence
 - Problems

Local site skytest

state	Host	Icons	OK	Wa	Un	Cr	Pd	state	Host	Icons
UP	localhost		22	0	0	0	1	UP	Switch 10.39.238.28	

Click on *Inventory* button

Raw 2.8p9

Services of Host Switch_10.39.238.28

WATO Host status Service graphs
 Event History of Host Events of Monitored ... **Inventory** Inventory History
 Host downtimes Host comments Host Aggregations Network Interfaces

Switch_10.39.238.28

State	Service	Icons	Status detail
OK	Check_MK		OK - execution time 1.0 sec
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found
OK	Check_MK HW/SW Inventory		OK - found 1193 entries
OK	CPU utilization		OK - 9.0% utilization in the last 5 minutes
OK	FAN Switch 1 Fan 1		OK - State is: normal (1)

Raw 2.8p9

Inventory of host Switch_10.39.238.28

WATO Availability ...

Hostname Switch_10.39.238.28

Inventory Tree

- Hardware
- Networking
- Software

Inventory of host Switch_10.39.238.28 1 row omdadmin (adm)

WATO Availability ...

Hostname Switch_10.39.238.28

Inventory Tree

- Hardware
 - System

Model Name	WS-C3560G-48TS-S
Product	Cisco IOS Software, C3560 Software (C3560-IPSERVICESK9-M), Version 15.0(1)SE2, RELEASE SOFTWARE (fc3) Technical Support: http://www.cisco.com/tech/1986-2011 by Cisco Systems, Inc. Compiled Thu 22-Dec-11 00:24 by prod_rel_team
Serial Number	FOC1126Y21N
- Networking

Interfaces	2
Ports	52
Ports available	9

Open this t

Index	Description	Alias	Status	Admin	Used	Speed	Last Change	Physical Address (MAC)
1	Vlan1		down	down		1 Gbit/s	140 days ago	00:1C:B1:33:30:C0
381	Vlan381		up	up		1 Gbit/s	140 days ago	00:1C:B1:33:30:C1
391	Vlan391		up	up		1 Gbit/s	119 days ago	00:1C:B1:33:30:C2
10101	GigabitEthernet0/1		up	up	used	1 Gbit/s	29 days ago	00:1C:B1:33:30:81
10102	GigabitEthernet0/2		up	up	used	1 Gbit/s	29 days ago	00:1C:B1:33:30:82
10103	GigabitEthernet0/3		up	up	used	100 Mbit/s	140 days ago	00:1C:B1:33:30:83
10104	GigabitEthernet0/4		up	up	used	1 Gbit/s	29 days ago	00:1C:B1:33:30:84
10105	GigabitEthernet0/5		up	up	used	1 Gbit/s	29 days ago	00:1C:B1:33:30:85
10106	GigabitEthernet0/6		up	up	used	100 Mbit/s	140 days ago	00:1C:B1:33:30:86
10107	GigabitEthernet0/7		up	up	used	1 Gbit/s	-41 days ago	00:1C:B1:33:30:87
10108	GigabitEthernet0/8		up	up	used	1 Gbit/s	-41 days ago	00:1C:B1:33:30:88
10109	GigabitEthernet0/9		up	up	used	100 Mbit/s	140 days ago	00:1C:B1:33:30:89
10110	GigabitEthernet0/10		up	up	used	1 Gbit/s	41 days ago	00:1C:B1:33:30:8A

Using custom plugins

Sometimes it's necessary to create custom checks and Check_MK makes this possible using *Local Checks*, *MRPE* or *MKP*.

As with *folders*, *Tags* and *Hostgroups* they are three different ways of doing the same thing and each one of them has pros and cons.

This is a summary:

Local Checks are used whenever you want something really quick and simple. Just create a script with your preferred language and place it on the monitored machine.

Pros:

- easy and asynchronous

Cons:

- no central management using WATO, all parameters will be managed inside the script.

MRPE is useful if you want a soft migration from NRPE to Check_MK.

Pros:

- supports any kind of Nagios plugin.

Cons:

- all plugins on localhost are called at the same time, once per cycle; there is no way to call some more often than others.
- The plugins are called in direct sequence - one after another. No parallelization takes place.

MKP is the native plugin format and is definitely the best/preferred method. The new packaging mechanism of Check_MK supports you in distributing your extensions and using extensions from other people by allowing you to easily create, install, update and remove packages of extensions, which are portable between all installations of Check_MK - regardless of the installations paths chosen at setup.

Pros:

- Native format, Portability, WATO support, overall efficiency

Cons:

- Requires python knowledge

Local Checks

Check_MK also has the concept of “*local checks*” that are very easy and straightforward to use and give the ability to run any kind of script or program on an agent.

Example:

a) Create a script like this and place it in the *local* directory of the Check_MK agent

```
#!/bin/bash
DIRS="/var/log /tmp"

for dir in $DIRS
do
    count=$(ls $dir | wc --lines)
    if [ $count -lt 50 ] ; then
        status=0
        statustxt=OK
    elif [ $count -lt 100 ] ; then
        status=1
        statustxt=WARNING
    else
        status=2
        statustxt=CRITICAL
    fi
    echo "$status Filecount_$dir count=$count;50;100;0; $statustxt - $count files in $dir"
done
```

If you don't know the path to the local directory just do the following:

```
[root@centos7tst1 ~]# /usr/bin/check_mk_agent | grep -i local
Hostname: centos7tst1
LocalDirectory: /usr/lib/check_mk_agent/local
.....
```

b) Do an inventory of the host running

```
cmk -I centos7tst1
```

c) The new service should show up

OK	Nginx 127.0.0.1:80 Status			Accepted/Handled: 0.03/s
OK	Number of threads			OK - 121 threads
OK	Postfix Queue			OK - deferred queue length is 0, active queue length is 0
OK	CUSTOMSCRIPT_TEST_Filecount_/tmp			OK - 4 files in /tmp
CRIT	CUSTOMSCRIPT_TEST_Filecount_/var/log			CRIT - CRITICAL - 57 files in /var/log
OK	TCP Connections			OK - ESTABLISHED: 2, TIME_WAIT: 1, LISTEN: 9
OK	Uptime			OK - up since Fri Oct 14 11:58:07 2016 (0d 02:17:07)

MRPE – Nagios Plugins

These require just a couple of steps:

- a) Copy the plugin into the agent *plugin* directory.
- b) Create a configuration file *mrpe.cfg* and place it in the agent's configuration directory; if you did not change that at setup, the complete path is */etc/check_mk/mrpe.cfg*.

```
/etc/check_mk/mrpe.cfg
LOAD      /usr/lib/nagios/plugins/check_load -w 2 -c 5
FS_var    /usr/lib/nagios/plugins/check_disk /var
FS_hirn   /usr/lib/nagios/plugins/check_disk /hirn
Aptitude  /usr/lib/nagios/plugins/check_apt
Smart_sda /usr/lib/nagios/plugins/check_ide_smart -d /dev/sda -n
```

- c) Inventory the host

```
cmk -I --checks=mrpe somehost123
```

MKP plugins

Instead of using *Local checks* or *MRPE*, there are lot of external plugins available in the native Check_MK format (mkp). There is a catalog on https://mathias-kettner.de/checkmk_check_catalogue.html but it's also possible to create your own using python.

To show the installation, I chose *MTR*, a nice plugin which is very useful to use when troubleshooting network problems. It was created by BenV and you can download it from his website: <https://notes.benv.junerules.com/mtr/>

The reason why I think this plugin is really great is that it uses *MTR*, a tool that combines the functionality of the 'traceroute' and 'ping' programs in a single network diagnostic tool.

As the documentation for *mtr* states, it investigates the network connection between the host *mtr* runs on and a user-specified destination host. After it determines the address of each network hop between the machines, it sends a sequence ICMP ECHO requests to each one to determine the quality of the link to each machine. As it does this, it prints running statistics about each machine. For more information please visit its website <https://www.bitwizard.nl/mtr/>

On the Check_MK host:

- Download the latest version from the website and place in */tmp*
- Install using *mkp*

```
OMD[mysite]:~$ mkp install /tmp/mtr-0.5.2.mkp
```

- Copy the plugin and the configuration file onto the machine where you want to run the pings from. Please note that you need to place the plugin in the agent's *plugins* folder and the associated *cfg* file in the agent's configuration folder

```
[root@checkmktst1 tmp]# scp /opt/omd/sites/mysite/local/share/check_mk/agents/mtr
root@10.39.239.99:/usr/lib/check_mk_agent/plugins/
[root@checkmktst1 tmp]# scp
/opt/omd/sites/mysite/local/share/check_mk/agents/cfg_examples/mtr.cfg
root@10.39.239.99:/etc/check_mk/
```

On the client machine:

- Amend the configuration file, adding hosts that you need to monitor:

```
[root@centos7tst1 tmp]# cat /etc/check_mk/mtr.cfg
# Mtr Check_MK configuration

# NOTE: your MTR report shouldn't take longer than 15 minutes

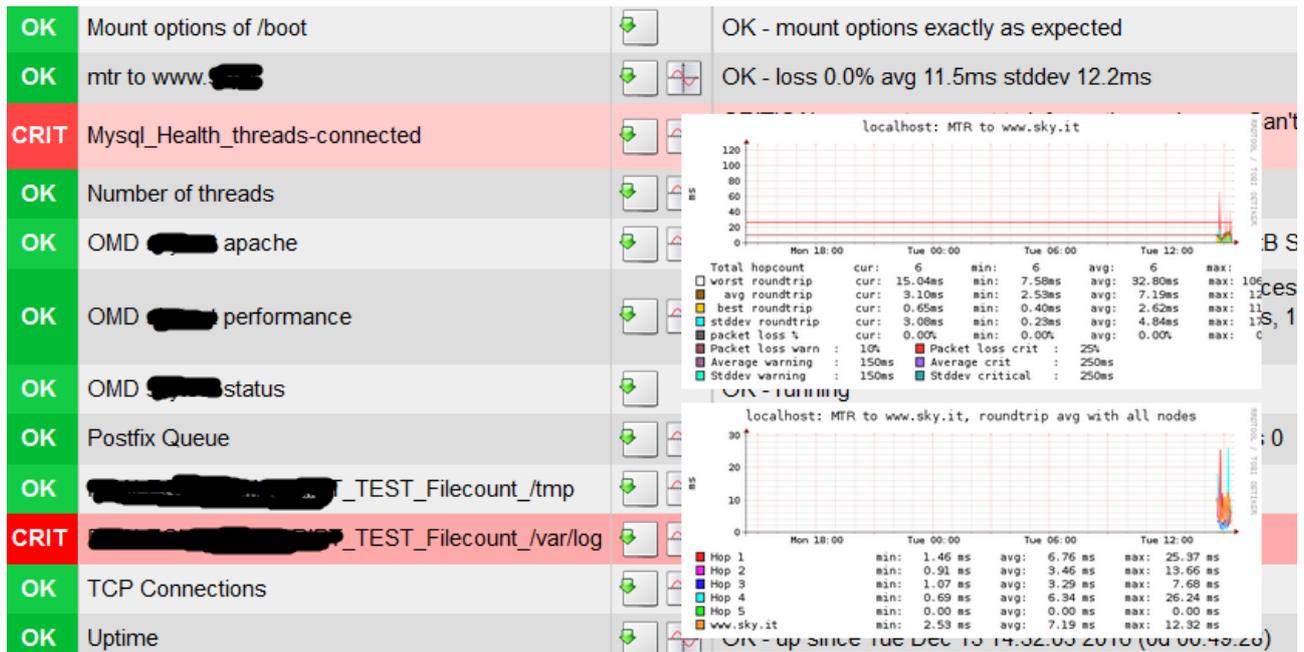
# [DEFAULTS]
# type=icmp      # icmp, tcp or udp
# count=10       # number of pings per mtr report
# force_ipv4=0   # force ipv4, exclusive with force_ipv6
# force_ipv6=0   # force ipv6, exclusive with force_ipv4
# size=64        # packet size
# time=0         # minimum time between runs, 0 / default means run if mtr doesn't run
anymore
# port=80        # UDP/TCP port to connect to
# dns=0          # Use DNS resolution to lookup addresses
# address=       # Bind to source address
# interval=     # time MTR waits between sending pings
# timeout=      # ping Timeout, see mtr man page

[www.google.com]
type = icmp
force_ipv4 = true

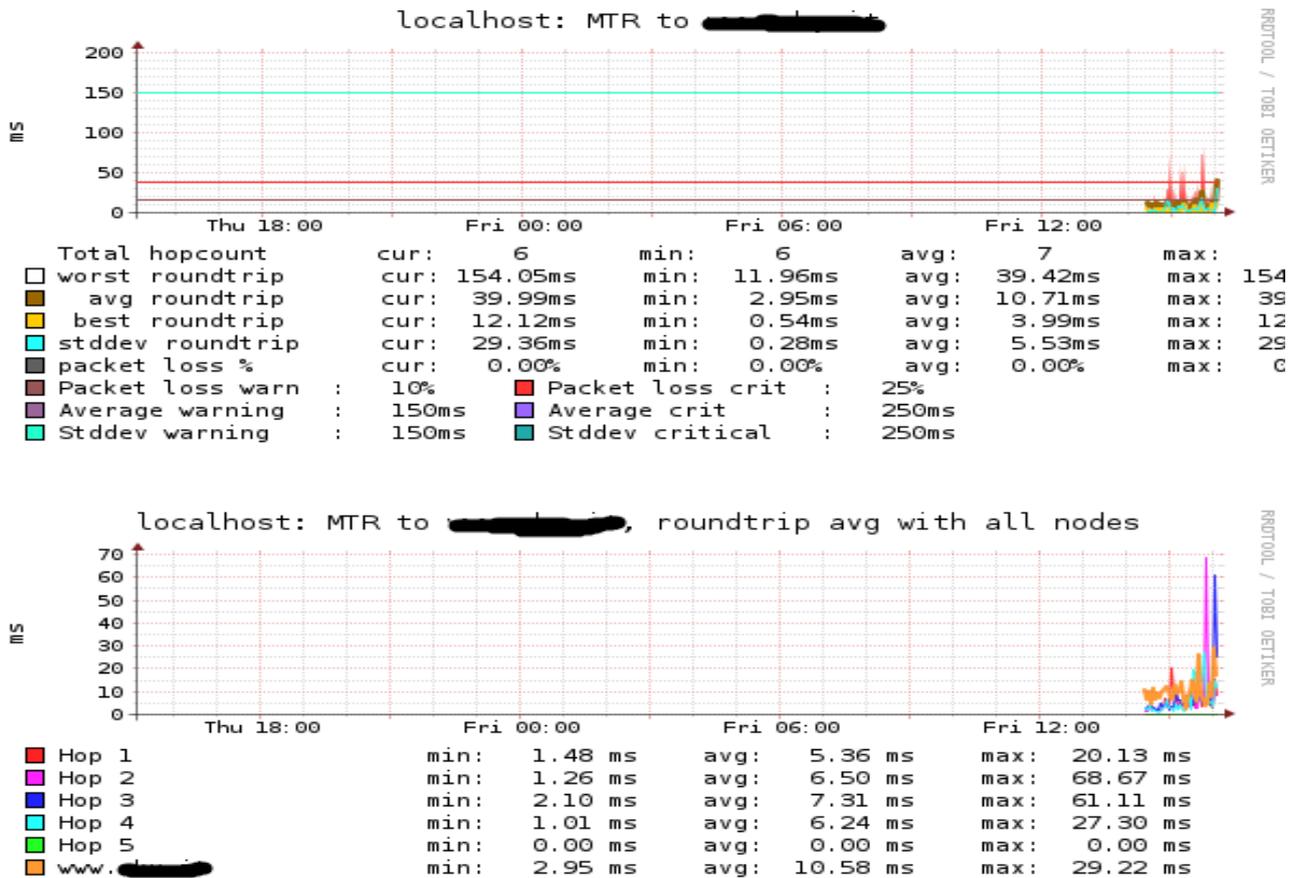
[ipv6.google.com]
type = icmp
force_ipv6 = true
```

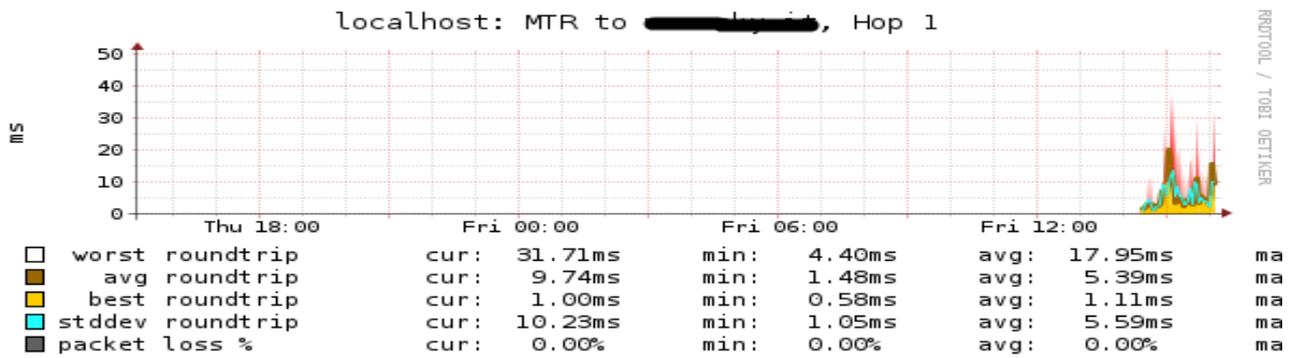
- Restart the agent
- Do a service discovery adding unmonitored services

This is the result:

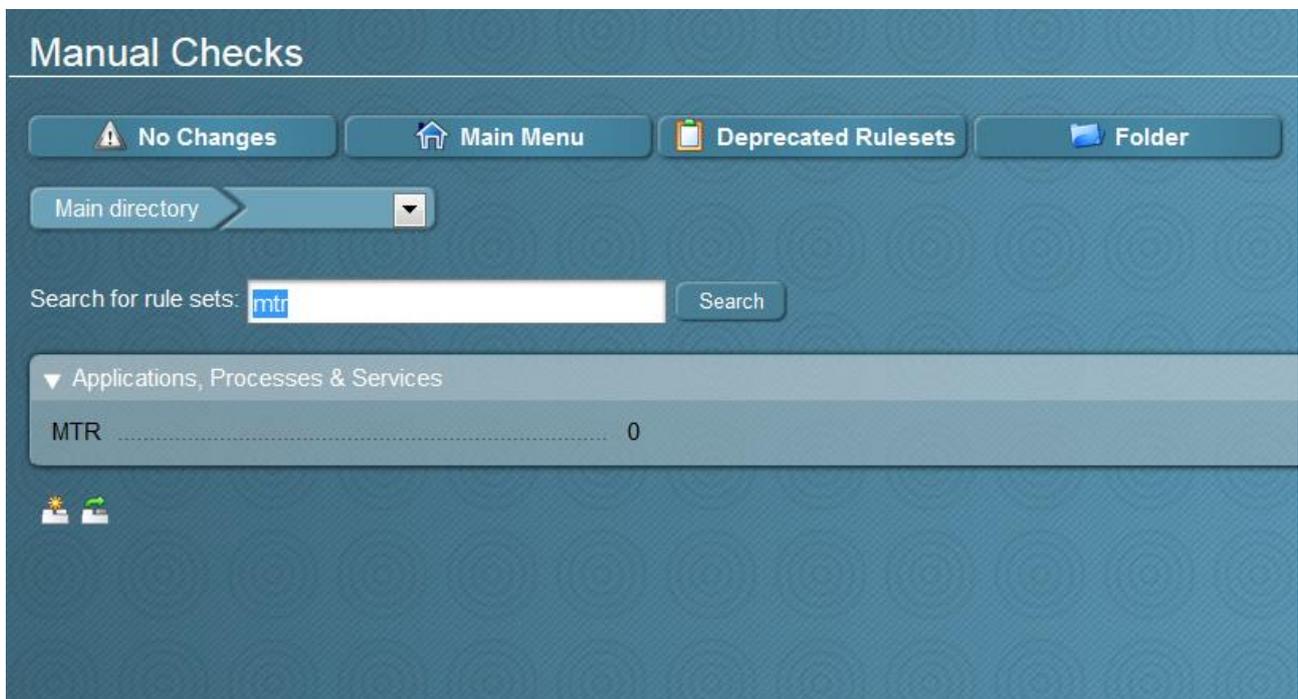


I don't think any comment is necessary here, this is really amazing!





Because this is a native plugin, it's possible to manage parameters using *WATO, Manual Checks*



New rule MTR

▼ Rule Options

Description

Comment

Documentation-URL

Rule activation do not apply this rule

▼ Parameters

Checktype

MTR destination

Average roundtrip time in ms

Warning at

Critical at

Standard deviation of roundtrip times in ms

Warning at

Critical at

Parameters

Packet loss in percentage

Warning at

Critical at

Monitor Apache Webserver

In this example, I'm going to monitor Apache using its *server-status* module that must be manually enabled in the Apache configuration file.

```
<IfModule mod_status.c>
  <Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1 ::1
  </Location>

  # Keep track of extended status information for each request
  ExtendedStatus On
</IfModule>
```

- Copy the apache plugin in the agent folder

```
cp -pi /opt/omd/versions/1.2.8p9.cre/share/check_mk/agents/plugins/apache_status
/usr/lib/check_mk_agent/plugins/
```

- Do a service discovery and apply changes

Services of Host localhost

State	Service	Icons	Status detail
OK	Check_MK		OK - Agent version 1.2.8p9, execution time 0.3 sec
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found
OK	Check_MK HW/SW Inventory		OK - found 16992 entries
OK	Apache 127.0.0.1:5000 Status		OK - Uptime: 37 min, IdleWorkers: 6, BusyWorkers: 2, OpenSlots: 248, TotalSlots: 256, CPUload: 1.02, ReqPerSec: 0.33, BytesPerReq: 6500.65, BytesPerSec: 341.33, States: (Waiting: 6, SendingReply: 2)
OK	CPU load		OK - 15 min load 0.06
OK	CPU utilization		OK - user: 6.9%, system: 4.0%, wait: 0.0%, steal: 0.0%, guest: 0.0%, total: 10.9%
OK	Disk IO SUMMARY		OK - Utilization: 0.0%, Read: 0.00 B/s, Write: 1.00 kB/s, Average Wait: 0.00 ms, Average Read Wait: 0.00 ms, Average Write Wait: 0.00 ms, Latency: 0.00 ms
OK	Filesystem /		OK - 20.6% used (5.17 of 17.16 GB), total: 18.06 MB / 24 hours

Host: localhost Service: Apache 127.0.0.1:5000 Status

4 Hours 26.08.16 9:56 - 26.08.16 13:56

Datasource: Apache Status

localhost: Apache_127.0.0.1_5000_Status

Connections

8.0
6.0
4.0
2.0
0.0

12:00 12:20 12:40 13:00 13:20 13:40 14:00 14:20 14:40 15:00 15:20 15:40

■ Total Slots: 256

StartingUp	Last	0.0	Max	0.0	Average	0.0
Waiting	Last	7.0	Max	7.0	Average	7.0
Logging	Last	0.0	Max	0.0	Average	0.0
DNS	Last	0.0	Max	0.0	Average	0.0
SendingReply	Last	1.0	Max	1.0	Average	1.0
ReadingRequest	Last	0.0	Max	0.0	Average	0.0
Closing	Last	0.0	Max	0.0	Average	0.0
IdleCleanup	Last	0.0	Max	0.0	Average	0.0
Finishing	Last	0.0	Max	0.0	Average	0.0
Keepalive	Last	0.0	Max	0.0	Average	0.0
UsedSlots	Last	8.0	Max	8.0	Average	8.0

Datasource: Requests/sec

localhost: Apache_127.0.0.1_5000_Status Requests/sec

800 m
600 m
400 m
200 m
0

12:00 12:20 12:40 13:00 13:20 13:40 14:00 14:20 14:40 15:00 15:20 15:40

■ ReqPerSec 0.1/s Last 0.8/s Max 0.2/s Average

Datasource: Bytes/sec

localhost: Apache_127.0.0.1_5000_Status Bytes/sec

Search

Actions

My basket

Basket is empty

Multisite links

Host: localhost
Service: Apache 127.0.0.1:5000

Time ranges

- Overview
- 4 Hours
- 25 Hours
- One Week
- One Month
- One Year

Services

- Host Perfdata
- Apache 127.0.0.1:5000
- Check_MK
- CPU load
- CPU utilization
- Disk IO SUMMARY
- Filesystem /
- Filesystem /boot
- Interface 2
- Interface 3
- Kernel Context Sw
- Kernel Major Page
- Kernel Process Cre
- Memory
- Number of threads
- OMD skytest apach
- OMD skytest perfor
- Postfix Queue

Monitor Mysql Server

The base agent doesn't include native support but check_mk created *mk_mysql* official plugin. I did a test on mariadb 5.5 on centos 7.2 64 bit but the same applies to other mysql versions even when running on Windows

- On mysql server, create a user only for monitoring, giving to it the rights with following SQL statement

```
MariaDB [(none)]> GRANT SELECT, SHOW DATABASES ON *.* TO 'mysqlmonitor'@'localhost' IDENTIFIED BY 'mysqlmonitor';
```

- Copy the plugin from check_mk to the mysql host

```
[root@checkmktst1]# scp /opt/omd/versions/1.2.8p13.cre/share/check_mk/agents/plugins/mk_mysql root@10.39.239.99:/usr/lib/check_mk_agent/plugins
```

- Create the file *mysql.cfg* in the agent configuration folder.

```
[root@centos7tst1 ~]# cat /etc/check_mk/mysql.cfg

[client]
user=mysqlmonitor
password=mysqlmonitor
```

- Change *mysql.cfg* permissions. Setting mode 400 ensures it will not be readable for non-root users:

```
chmod 400 /etc/check_mk/mysql.cfg
```

- Restart the agent on the client machine
- Do a service discovery adding unmonitored services

OK	Memory		OK - RAM used: 230.75 MB of 1.80 GB, Swap used: 0.00 B of 2.00 GB, Total virtual 230.75 MB of 3.80 GB (5.9%),
OK	Mount options of /		OK - mount options exactly as expected
OK	Mount options of /boot		OK - mount options exactly as expected
OK	██████████		OK - loss 0.0% avg 12.3ms stddev 0.7ms
OK	MySQL Connections mysql		OK - Max. parallel Connections: 1 (Max.: 151): 0.66%
OK	MySQL InnoDB IO mysql		OK - 0.00 B/sec read, 0.00 B/sec write
OK	MySQL Instance mysql		OK - MySQL Deamon is alive
OK	MySQL Sessions mysql		OK - 1 total, 1 running, 0.44 connections/s
OK	MySQL Version mysql		OK - Version: 5.5.50-MariaDB
OK	Nginx 127.0.0.1:80 Status		OK - Active: 1 (0 reading, 1 writing, 0 waiting), Requests: 0.11/s (1.00/Connection), Accepted/Handled: 0.11/s

Whenever a Mysql fail should occur, you will be warned

UNKN	MySQL Connections mysql		UNKNOWN - Connection information are missing
UNKN	MySQL InnoDB IO mysql	  	UNKNOWN - check failed - please submit a crash report!
CRIT	MySQL Instance mysql		CRIT - mysqladmin: connect to server at 'localhost' failed
UNKN	MySQL Sessions mysql	  	UNKNOWN - check failed - please submit a crash report!
UNKN	MySQL Version mysql	 	UNKNOWN - check failed - please submit a crash report!

As suggested in the documentation, you should also monitor the mysql process, error log, innodb redo log etc.

It's also possible to monitoring any parameter you ever want, using the community plugin available at <http://exchange.check-mk.org/>

To be honest, I expected something more by this plugin because Mysql is a really widespread product and there are lot of metrics that should be monitored out of the box. I'm sure that it will be improved over time but, in the meanwhile, I decided to use the excellent *check_mysql_health* from [Console Labs](#).

There are a couple of possible paths:

- Install *check_mysql_health* on the *check_mk* host and create active checks for each parameter
- Install *check_mysql_health* directly on the mysql server and configure *MPRE*

I tested both of them but I'm going to show you only the second one because it is my preferred method In this scenario the Mysql server's hostname is centos7tst1 with ip address 10.39.239.99

Steps:

- On the Mysql server download and Install the plugin

```
[root@centos7tst1 tmp]# wget
https://labs.consol.de/assets/downloads/nagios/check_mysql_health-2.2.2.tar.gz
[root@centos7tst1 tmp]# tar xzvf check_mysql_health-2.2.2.tar.gz
[root@centos7tst1 tmp]# cd check_mysql_health-2.2.2/
[root@centos7tst1 check_mysql_health-2.2.2]# ./configure --
prefix=/usr/lib/check_mk_agent/plugins --with-nagios-user=root --with-nagios-group=root
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
.....
.....

[root@centos7tst1 check_mysql_health-2.2.2]# make && make install
```

- Create the *MRPE* configuration file in */etc/check_mk/mrpe.cfg*

```
mysqlhealth_connection-time /usr/lib/check_mk_agent/plugins/libexec/check_mysql_health --
hostname 10.39.239.99 --username mysqlmonitor --password mysqlmonitor --mode connection-
time
mysqlhealth_uptime /usr/lib/check_mk_agent/plugins/libexec/check_mysql_health --hostname
10.39.239.99 --username mysqlmonitor --password mysqlmonitor --mode uptime
mysqlhealth_threads-connected /usr/lib/check_mk_agent/plugins/libexec/check_mysql_health
--hostname 10.39.239.99 --username mysqlmonitor --password mysqlmonitor --mode threads-
connected
mysqlhealth_threadcache-hitrate
/usr/lib/check_mk_agent/plugins/libexec/check_mysql_health --hostname 10.39.239.99 --
username mysqlmonitor --password mysqlmonitor --mode threadcache-hitrate
```

```

mysqlhealth_qcache-hitrate /usr/lib/check_mk_agent/plugins/libexec/check_mysql_health --
hostname 10.39.239.99 --username mysqlmonitor --password mysqlmonitor --mode qcache-
hitrate
mysqlhealth_qcache-lowmem-prunes
/usr/lib/check_mk_agent/plugins/libexec/check_mysql_health --hostname 10.39.239.99 --
username mysqlmonitor --password mysqlmonitor --mode qcache-lowmem-prunes
mysqlhealth_keycache-hitrate /usr/lib/check_mk_agent/plugins/libexec/check_mysql_health -
-hostname 10.39.239.99 --username mysqlmonitor --password mysqlmonitor --mode keycache-
hitrate
mysqlhealth_bufferpool-hitrate /usr/lib/check_mk_agent/plugins/libexec/check_mysql_health
--hostname 10.39.239.99 --username mysqlmonitor --password mysqlmonitor --mode
bufferpool-hitrate
mysqlhealth_bufferpool-wait-free
/usr/lib/check_mk_agent/plugins/libexec/check_mysql_health --hostname 10.39.239.99 --
username mysqlmonitor --password mysqlmonitor --mode bufferpool-wait-free
mysqlhealth_log-waits /usr/lib/check_mk_agent/plugins/libexec/check_mysql_health --
hostname 10.39.239.99 --username mysqlmonitor --password mysqlmonitor --mode log-waits
mysqlhealth_tablecache-hitrate /usr/lib/check_mk_agent/plugins/libexec/check_mysql_health
--hostname 10.39.239.99 --username mysqlmonitor --password mysqlmonitor --mode
tablecache-hitrate
mysqlhealth_table-lock-contention
/usr/lib/check_mk_agent/plugins/libexec/check_mysql_health --hostname 10.39.239.99 --
username mysqlmonitor --password mysqlmonitor --mode table-lock-contention
mysqlhealth_index-usage /usr/lib/check_mk_agent/plugins/libexec/check_mysql_health --
hostname 10.39.239.99 --username mysqlmonitor --password mysqlmonitor --mode index-usage
mysqlhealth_slow-queries /usr/lib/check_mk_agent/plugins/libexec/check_mysql_health --
hostname 10.39.239.99 --username mysqlmonitor --password mysqlmonitor --mode slow-queries
mysqlhealth_long-running-procs /usr/lib/check_mk_agent/plugins/libexec/check_mysql_health
--hostname 10.39.239.99 --username mysqlmonitor --password mysqlmonitor --mode long-
running-procs
mysqlhealth_open-files /usr/lib/check_mk_agent/plugins/libexec/check_mysql_health --
hostname 10.39.239.99 --username mysqlmonitor --password mysqlmonitor --mode open-files

```

- Change *Mysql* permission to allow connections coming from 10.39.239.99 that is the primary ip address of the machine.

```

MariaDB [(none)]> GRANT SELECT, SHOW DATABASES ON *.* TO 'mysqlmonitor'@'10.39.239.99'
IDENTIFIED BY 'mysqlmonitor';

```

- On the *check_mk* host run a new host inventory

```

cmk -II centos7tst1
cmk -R

```

CRIT	mysqlhealth_bufferpool-hitrate	 	CRIT - CRITICAL - innodb buffer pool hitrate at 68.63%
OK	mysqlhealth_bufferpool-wait-free	 	OK - 0 innodb buffer pool waits in 60 seconds (0.0000/sec)
OK	mysqlhealth_connection-time	 	OK - 0.03 seconds to connect as mysqlmonitor
CRIT	mysqlhealth_index-usage	 	CRIT - CRITICAL - index usage 5.16%
CRIT	mysqlhealth_keycache-hitrate	 	CRIT - CRITICAL - myisam keycache hitrate at 50.00%
OK	mysqlhealth_log-waits	 	OK - 0 innodb log waits in 60 seconds (0.0000/sec)
OK	mysqlhealth_long-running-procs	 	OK - 0 long running processes
OK	mysqlhealth_open-files	 	OK - 4.98% of the open files limit reached (51 of max. 1024)
OK	mysqlhealth_qcache-hitrate	 	OK - query cache hitrate 0.00% (because it's turned off)
OK	mysqlhealth_qcache-lowmem-prunes	 	OK - 0 query cache lowmem prunes in 60 seconds (0.00/sec)
OK	mysqlhealth_slow-queries	 	OK - 0 slow queries in 60 seconds (0.00/sec)
OK	mysqlhealth_table-lock-contention	 	OK - table lock contention 0.00% (uptime < 10800)
OK	mysqlhealth_tablecache-hitrate	 	OK - table cache hitrate 273.33%, 10.25% filled
CRIT	mysqlhealth_threadcache-hitrate	 	CRIT - CRITICAL - thread cache hitrate 0.08%
OK	mysqlhealth_threads-connected	 	OK - 2 client connection threads
OK	mysqlhealth_uptime	 	OK - database is up since 65 minutes

There are lot of parameters that can be monitored and it's even possible run sql statemens using `-mode sql` defining also thresholds. Please refer to the official documentation to get more more informations.

Ps: Reading werks, I noticed that `check_mysql_health` should be already included in the upcoming version 1.4.

#7570 packages: Fixed potential deadlock when talking to rrdcached	Trivial Change Bug Fix
#7557 packages: Updated NSCA to be compatible with clients of newer distros	inco... Trivial Change Bug Fix
#7503 packages: Base URL redirects preserve https protocols now	Trivial Change Bug Fix
#7507 packages: check_icmp : Fixed not using configured ping levels since IPv6 implementation	Trivial Change Bug Fix
#7506 packages: Shipping <code>check_mysql_health</code> , <code>check_oracle_health</code> , <code>check_nrpe</code> and <code>check_multi</code> again	Trivial Change New Feature
#7554 packages: Fixed possible broken Check_MK web cron job when HTTPS is configured	Trivial Change Bug Fix

Anyway, you still need to manually install the plugin on the monitored server if you are going to use the second path

Monitor Physical Hardware

To properly monitor hardware (FAN, CPU, MEMORY, DISKS etc.) from the likes of HP or Dell, the first step is to install and configure the agents on the running OS. Because the procedure is very simple and there are many guides that show how to achieve exactly that, I'll just show the "nagios" part for an HP ProLiant running Redhat 5.x

- Change *SNMPD* configuration

Because the default *snmpd* configuration doesn't expose all OIDs, we need to change the configuration by adding or changing the following entries:

```
vi /etc/snmp/snmpd.conf
```

```
-----snmp.conf-----  
  
# sec.name source community  
com2sec notConfigUser default public  
  
# groupName securityModel securityName  
group notConfigGroup v1 notConfigUser  
group notConfigGroup v2c notConfigUser  
  
# Make at least snmpwalk -v 1 localhost -c public system fast again.  
# name incl/excl subtree mask(optional)  
view all included .1  
view systemview included .1.3.6.1.2.1.1  
view systemview included .1.3.6.1.2.1.25.1.1  
  
# group context sec.model sec.level prefix read write notif  
access notConfigGroup "" any noauth exact all none none  
  
-----
```

- restart the *snmpd* service

```
service snmpd restart
```

- Test the new configuration using *snmpwalk*

From *check_MK*, check if we can get the model using *snmpwalk*

```
[root@checkmktst1 ~]# snmpwalk -v2c -c public 172.17.25.1 .1.3.6.1.4.1.232.2.2.4.2.0  
SNMPv2-SMI::enterprises.232.2.2.4.2.0 = STRING: "ProLiant BL460c G7"
```

- Add the device changing the Agent type to: *Dual: Check_MK Agent + SNMP* and do a Service discovery

▼ General Properties

Hostname ██████████

▼ Basic settings

Permissions empty (Default value)

Alias empty (Default value)

IPv4 Address 172.17.25.2

Parents empty (Default value)

Monitored on site skytest - Local site skytest (Default value)

▼ Host tags

Agent type Check_MK Agent (Server)

Criticality

Networking Segment

IP Address Family

Check_MK Agent (Server)
 Check_MK Agent (Server)
 SNMP (Networking device, Appliance)
 Legacy SNMP device (using V1)
Dual: Check_MK Agent + SNMP
 No Agent

Bulk Service Discovery

← Folder

You have selected 1 hosts for bulk discovery. Check_MK service discovery will automatically find and configure services to be checked on

▼ Bulk Discovery

Mode Add unmonitored services

Remove vanished services

Add unmonitored & remove vanished services

Refresh all services (tabula rasa)

Selection Only include hosts that failed on previous discovery

Only include hosts with a failed discovery check

Exclude hosts where the agent is unreachable

Performance options Use cached data if present

Do full SNMP scan for SNMP devices

Number of hosts to handle at once:

Error handling Ignore errors in single check plugins

Start

Bulk Service Discovery

← Folder

Bulk Service Discovery

```
skytest2: discovery successful
```

FINISHED.

Total hosts	1
Failed hosts	0
Skipped hosts	0
Services added	49
Services removed	0
Services kept	66
Total services	115

Finish Restart

OK	TCP Connections		OK - ESTABLISHED: 99, TIME_WAIT: 16, LISTEN: 41	2016-09-08 14:16:29	57 sec	
OK	Temperature 1 ambient		OK - 18.0 °C	2 hrs	56 sec	18 °C
OK	Temperature 2 cpu		OK - 40.0 °C	2 hrs	56 sec	40 °C
OK	Temperature 3 cpu		OK - 40.0 °C	2 hrs	56 sec	40 °C
OK	Temperature 4 memory		OK - 30.0 °C	2 hrs	56 sec	30 °C
OK	Temperature 5 memory		OK - 32.0 °C	2 hrs	56 sec	32 °C
OK	Temperature 6 storage		Open the action menu °C	2 hrs	56 sec	35 °C
OK	Temperature 7 memory		OK - 31.0 °C	2 hrs	56 sec	31 °C
OK	Temperature 8 memory		OK - 35.0 °C	2 hrs	56 sec	35 °C
OK	Temperature 9 ioBoard		OK - 55.0 °C	2 hrs	56 sec	55 °C
OK	Temperature 10 ioBoard		OK - 42.0 °C	2 hrs	56 sec	42 °C
OK	Temperature 11 ioBoard		OK - 34.0 °C	2 hrs	56 sec	34 °C
OK	Temperature 12 system		OK - 29.0 °C	2 hrs	56 sec	29 °C
OK	Temperature 13 system		OK - 21.0 °C	2 hrs	56 sec	21 °C
OK	Uptime		OK - up since Mon Jul 11 11:02:45 2016 (84d 06:56:44)	2016-09-08 14:16:29	56 sec	84 d

Monitor Vmware

In order to monitor VMware ESXi and vCenter Server, Check_MK has implemented a plugin that uses the vSphere API that is much more efficient than other free plugins like `check_esx3.pl` or `check_vmware_api.pl`. In the VMware World, basically there are 2 kinds of environments:

- ESXi free – Should be used just for test or lab, no support, no vcenter, no backup using external tools (apis locked out)
- vSphere that comes with different licensing options – It does include vCenter + a certain number of ESXi hosts depending on the licence

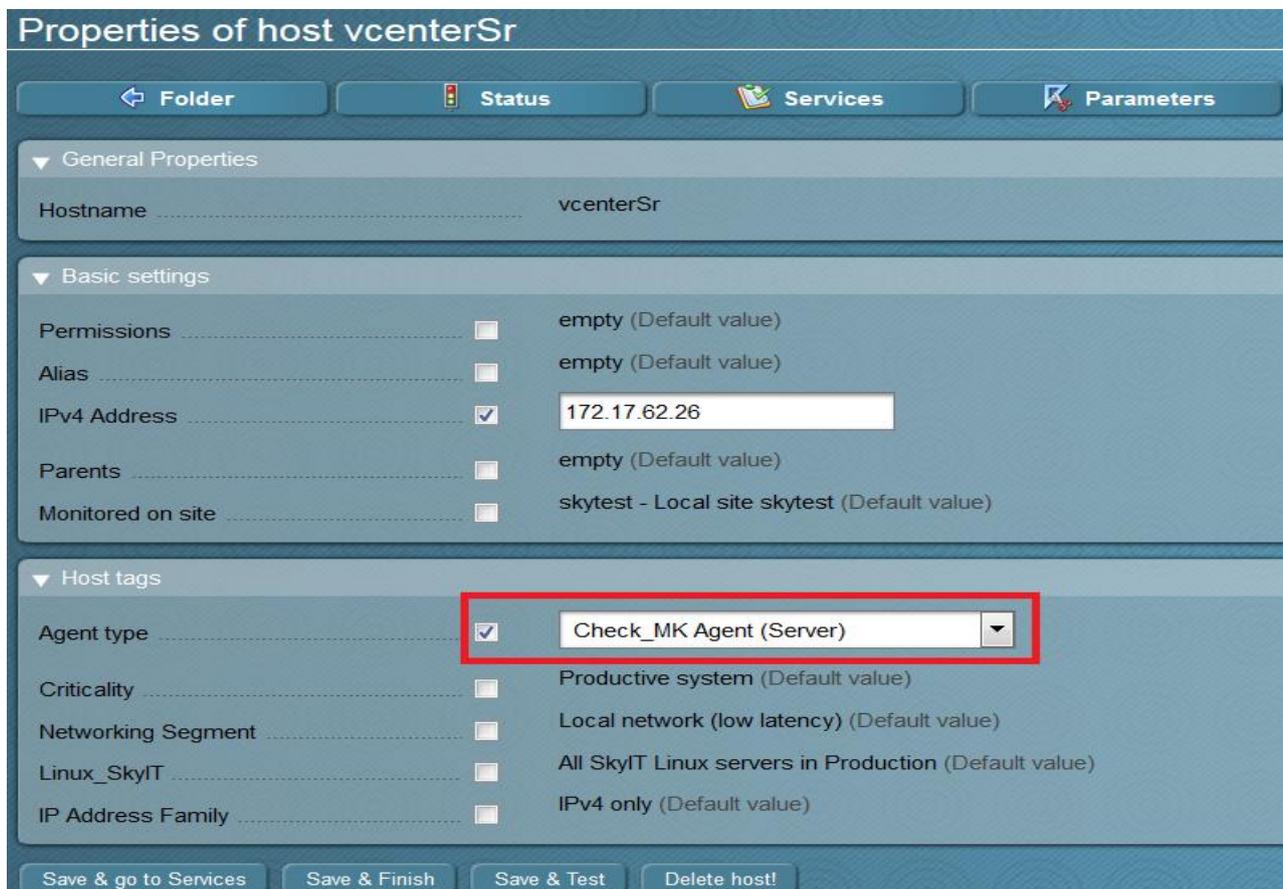
In both cases, monitoring has the following requirements:

Read-only user on vsphere side

Tcp port 443 (check_mk towards vsphere)

Add vSphere Virtual Center

Add the vcenter host entering the *Hostname*, *IPv4 Address* and as Agent select Check_MK Agent even though it isn't really installed. Click on *Save & Finish*



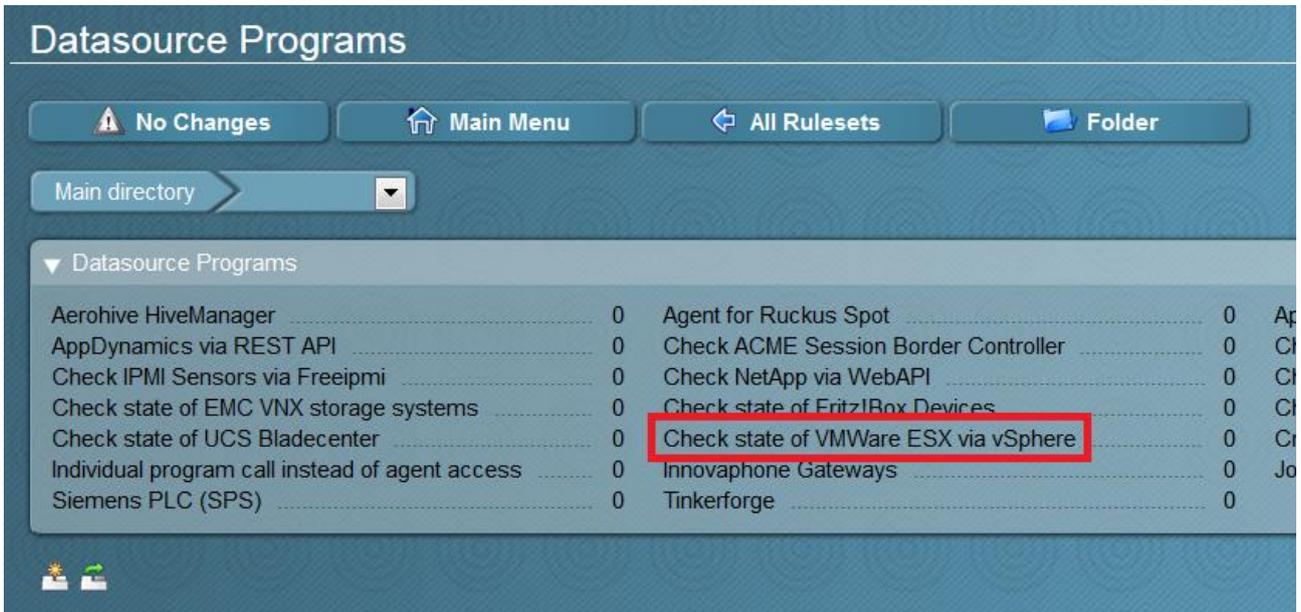
The screenshot shows the 'Properties of host vcenterSr' configuration page. The page is divided into several sections:

- General Properties:** Hostname is set to 'vcenterSr'.
- Basic settings:** Includes fields for Permissions, Alias, IPv4 Address (172.17.62.26), Parents, and Monitored on site (skytest - Local site skytest).
- Host tags:** Includes fields for Agent type (Check_MK Agent (Server)), Criticality, Networking Segment, Linux_SkyIT, and IP Address Family.

The 'Agent type' dropdown menu is highlighted with a red box, indicating that 'Check_MK Agent (Server)' is selected. At the bottom of the page, there are buttons for 'Save & go to Services', 'Save & Finish', 'Save & Test', and 'Delete host!'.

Click *Save & Finish*

To enable the advanced monitoring, in WATO configuration go to **Host & Service Parameters, Datasource Programs** and select **Check state of VMware ESX via vSphere**.



Create a new rule by clicking the button **Create rule in folder** and fill others fields as shown in the picture below. Just please note that:

- as vSphere User Name, I created an ad-hoc user that has just read-only permission:



- Is possible to define which kind of informations to retrieve: *Host Systems, Virtual Machines, Datastores, Performance counters, License*.

You can select all of them at the price of a longer check execution time

Abort

This rule selects the vSphere agent instead of the normal Check_MK Agent and allows monitoring of VMWare ESX via the vSphere API. You can configure your connection settings here.

Rule Options

Description: vcenterSr
Comment: Virtual Center Rome
Documentation-URL:
Rule activation: do not apply this rule

Check state of VMWare ESX via vSphere

vSphere User name: usermonitoring@vsphere.local
vSphere secret: [redacted]
 TCP Port number
SSL certificate checking:
 Deactivated
 Use hostname
 Use other hostname: [redacted]
 Connect Timeout: 60 seconds
Retrieve information about...
 Host Systems
 Virtual Machines
 Datastores
 Performance Counters
 License Usage
 Display ESX Host power state on: The queried ESX system (vCenter / Host)
 Display VM power state on: The queried ESX system (vCenter / Host)
Spaces in hostnames: Replace with underscores
Type of query: Queried host is the vCenter
Placeholder VMs:
 Do not monitor placeholder VMs
Compatibility mode:
 Support ESX 4.1 (using slower PySphere implementation)

Conditions

Folder: Main directory
Host tags:
Agent type: ignore
Criticality: ignore
Networking Segment: ignore
Linux: ignore
IP Address Family: ignore
monitor via SNMP: ignore
monitor via Check_MK Agent: ignore
IPv4: ignore
IPv6: ignore
Explicit hosts:
 Specify explicit host names
vcenterSr
 Negate: make rule apply for all but the above hosts

Save

Click Save and do a new Bulk Service Discovery to add unmonitored services

Bulk Service Discovery

Folder

You have selected 1 hosts for bulk discovery. Check_MK service discovery will automatically find and configure services to be checked.

Bulk Discovery

- Mode
- Add unmonitored services
 - Remove vanished services
 - Add unmonitored & remove vanished services
 - Refresh all services (tabula rasa)
- Selection
- Only include hosts that failed on previous discovery
 - Only include hosts with a failed discovery check
 - Exclude hosts where the agent is unreachable
- Performance options
- Use cached data if present
 - Do full SNMP scan for SNMP devices
- Number of hosts to handle at once:
- Error handling
- Ignore errors in single check plugins

Start



Activate changes and look the discovered services

Services of Host vcenterSr 24 rows omdadmin (admin) 18:04

WATO Host/Svc notific. Host history Host/Svc history Edit View Availability

State	Service	Icons	Status detail	Age	Checked	Perf-O-Meter
OK	Check_MK		OK - Agent version 5.5, execution time 3.7 sec	5 min	50 sec	3.73 s
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found	8 min	6 min	
OK	Check_MK HW/SW Inventory		OK - found 51 entries	6 min	6 min	
OK	Filesystem datastore1 1		OK - 0.73% used (972.00 MB of 129.25 GB), trend: 0.00 B / 24 hours	5 min	47 sec	0.73%
OK	Filesystem datastore1 2		OK - 0.35% used (973.00 MB of 271.75 GB), trend: 0.00 B / 24 hours	5 min	47 sec	0.35%
OK	Filesystem datastore1		OK - 0.73% used (972.00 MB of 129.25 GB), trend: 0.00 B / 24 hours	5 min	47 sec	0.73%
OK	Filesystem DATASTORE_ESXIRMSR1		OK - 31.2% used (478.95 GB of 1.50 TB), trend: 0.00 B / 24 hours, uncommitted: 424.60 GB, provisioning: 58.8%	5 min	47 sec	31.2%
OK	Filesystem DATASTORE_ESXIRMSR2		OK - 14.3% used (220.32 GB of 1.50 TB), trend: 0.00 B / 24 hours, uncommitted: 5.26 GB, provisioning: 14.7%	5 min	47 sec	14.3%
OK	Filesystem DATASTORE_ESXIRMSR3		OK - 0.062% used (981.00 MB of 1.50 TB), trend: 0.00 B / 24 hours	5 min	47 sec	0.062%
OK	HostSystem esxirmsr1		OK - power state: poweredOn	5 min	47 sec	
OK	HostSystem esxirmsr2		OK - power state: poweredOn	5 min	47 sec	
OK	HostSystem esxirmsr3		OK - power state: poweredOn	5 min	47 sec	
OK	Object count		OK - Virtualmachines: 11, Hostsystems: 3	5 min	47 sec	
WARN	VM New_Virtual_Machine		WARN - power state: poweredOff, defined on [esxirmsr2]	5 min	47 sec	
WARN	VM RHEL65_ENV		WARN - power state: poweredOff, defined on [esxirmsr1]	5 min	47 sec	
OK	VM srcms1		OK - power state: poweredOn, running on [esxirmsr1]	5 min	47 sec	
OK	VM srdm1		OK - power state: poweredOn, running on [esxirmsr1]	5 min	47 sec	
OK	VM srdm1872test		OK - power state: poweredOn, running on [esxirmsr1]	5 min	47 sec	
OK	VM srpsgw1		OK - power state: poweredOn, running on [esxirmsr2]	5 min	47 sec	
OK	VM srpxtve1		OK - power state: poweredOn, running on [esxirmsr1]	5 min	47 sec	
OK	VM srtwc1		OK - power state: poweredOn, running on [esxirmsr1]	5 min	47 sec	
OK	VM srtwc1873test		OK - power state: poweredOn, running on [esxirmsr1]	5 min	47 sec	
OK	VM srxtvs1		OK - power state: poweredOn, running on [esxirmsr2]	5 min	47 sec	
OK	VM VMware_vCenter_Server_Appliance_Sc		OK - power state: poweredOn, running on [esxirmsr1]	5 min	47 sec	

refresh: 30 secs

Lot of nice informations are retrieved from vCenter such as:

- `esx_vsphere_datastores`, shows all datastores (shared and local!) connected to ESXi hosts managed by the vCenter Server.
- `esx_vsphere_licenses`, shows all VMware licenses stored on the vCenter Server (in fact the License Manager on the Platform Services Controller)
- `esx_vsphere_objects`, shows connected ESXi hosts and VMs running on these hosts.

This is a basic monitoring and you could even stop here but there are a lot of precious informations missing such as interfaces usage on every single hosts, HBA status, datastore read/write/latency etc.

A good VMware administrator should know the vital importance of these metrics, in particular the latency on datastores that caused me some headaches in the past. So let's go on adding ESXi hosts.

Note: To monitor the Vcenter host itself (like any other standard server) it is enough to install `check_MK` agent. Just please note that, in case of VCSA (linux virtual center appliance), we must allow incoming traffic on port 6556.

There is a step by step guide on this blog: https://paulgrevink.wordpress.com/2016/08/22/check_mk-and-vsphere-vcenter-server/

Add ESXi host managed by Vcenter

Under *WATO*, choose, *Hosts* and *New Host* enter the Hostname, IP and under **Agent Type** place a tick and select *Check_MK Agent*. Just please note even I'm using root, a read-only user is recommended.

The screenshot shows the 'Create new host' configuration page in WATO. The page is divided into several sections:

- General Properties:** Hostname is set to 'esximi1'.
- Basic settings:** Permissions, Alias, Parents, and Monitored on site are unchecked. IPv4 Address is checked and set to '172.17.62.21'. Other fields are set to their default values.
- Host tags:** Agent type is checked and set to 'Check_MK Agent (Server)'. Other tags like Criticality, Networking Segment, Linux_SkyIT, and IP Address Family are unchecked.

The 'Agent type' field is highlighted with a red box. At the bottom of the page, there are three buttons: 'Save & go to Services', 'Save & Finish', and 'Save & Test'.

Click *Save & Finish*.

To avoid duplicated alarms, for each ESXi host managed by a vCenter Server we must create a new the rule configuring items in this way:

- *Host Systems*, Select, will show detailed status of the ESXi host.
- *Virtual Machines*, do not Select, already set on the vCenter Server.
- *Datastores*, do not Select, already set on the vCenter Server.
- *Performance Counters*, Select, will show performance counters of the ESXi hosts.
- *License Usage*, do not Select.

The screenshot shows the Nagios configuration interface for editing a rule. The rule is titled "Check state of VMWare ESX via vSphere". The interface is divided into several sections:

- Rule Options:** Includes fields for Description (esxim1), Comment, and Documentation-URL. The "Rule activation" checkbox is unchecked, with the text "do not apply this rule".
- Check state of VMWare ESX via vSphere:** This section contains various configuration options:
 - vSphere User name: root
 - vSphere secret: masked with dots
 - TCP Port number: unchecked
 - SSL certificate checking: Deactivated (selected), Use hostname, Use other hostname
 - Connect Timeout: 60 seconds
 - Retrieve information about...: Host Systems (checked), Virtual Machines (unchecked), Datastores (unchecked), Performance Counters (checked), License Usage (unchecked)
 - Display ESX Host power state on: checked, The ESX Host (selected)
 - Display VM power state on: unchecked
 - Spaces in hostnames: Replace with underscores
 - Type of query: Queried host is a host system
 - Placeholder VMs: Do no monitor placeholder VMs (checked)
 - Compatibility mode: Support ESX 4.1 (using slower PySphere implementation) (unchecked)
- Conditions:** Includes a Folder dropdown (Main directory) and a list of host tags (Agent type, Criticality, Networking Segment, Linux_SkyIT, IP Address Family, monitor via SNMP, monitor via Check_MK Agent, IPv4, IPv6) all set to "ignore". The "Explicit hosts" section is checked, with "esxim1" entered in the field. A "Negate" checkbox is unchecked.

At the bottom left, there is a "Save" button.

Do a service discovery adding unmonitored services and activate changes. Host's specific informations such as Cpu/Memory, Datastore read/write/latency and network interfaces and HBA status will be displayed.

State	Service	Icons	Status detail
OK	Check_MK		OK - Agent version 5.5, execution time 1.4 sec
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found
OK	Check_MK HW/SW Inventory		OK - found 44 entries
OK	CPU utilization		OK - 24.3% used, 9.41GHz/38.65GHz, 2 sockets, 8 cores/socket, 32 threads
OK	Datastore IO SUMMARY		OK - Read: 1.00 kB/s, Write: 299.00 kB/s, Latency: 0.00 ms
OK	Disk IO SUMMARY		OK - Read: 34.67 kB/s, Write: 337.00 kB/s, Latency: 0.00 ms
OK	Hardware Sensors		OK - All sensors are in normal state
OK	HostSystem esximi1		OK - power state: poweredOn
OK	Interface 1		OK - [vmnic0] (up) MAC: d8:9d:67:19:22:94, 1 Gbit/s, in: 21.00 kB/s(0.0%), out: 35.00 kB/s(0.0%)
OK	Interface 2		OK - [vmnic1] (up) MAC: d8:9d:67:19:22:95, 1 Gbit/s, in: 0.00 B/s(0.0%), out: 0.00 B/s(0.0%)
OK	Interface 3		OK - [vmnic2] (up) MAC: d8:9d:67:19:22:96, 1 Gbit/s, in: 0.00 B/s(0.0%), out: 0.00 B/s(0.0%)
OK	Interface 4		OK - [vmnic3] (up) MAC: d8:9d:67:19:22:97, 1 Gbit/s, in: 0.00 B/s(0.0%), out: 0.00 B/s(0.0%)
OK	Interface 5		OK - [vmnic4] (up) MAC: ac:16:2d:a1:5f:6c, 1 Gbit/s, in: 0.00 B/s(0.0%), out: 0.00 B/s(0.0%)
OK	Interface 7		OK - [vmnic6] (up) MAC: ac:16:2d:a1:5f:6e, 1 Gbit/s, in: 0.00 B/s(0.0%), out: 0.00 B/s(0.0%)
OK	Interface 8		OK - [vmnic7] (up) MAC: ac:16:2d:a1:5f:6f, 1 Gbit/s, in: 0.00 B/s(0.0%), out: 0.00 B/s(0.0%)
OK	Maintenance Mode		OK - System not in Maintenance mode
OK	Memory used		OK - 43% used - 41.50 GB/95.97 GB
OK	Multipath 60002ac00000000000000002000154ca		OK - Type naa/fc, 2 active, 0 dead, 0 disabled, 0 standby, 0 unknown
OK	Multipath 60002ac00000000000000003000154ca		OK - Type naa/fc, 2 active, 0 dead, 0 disabled, 0 standby, 0 unknown
OK	Multipath 60002ac00000000000000004000154ca		OK - Type naa/fc, 2 active, 0 dead, 0 disabled, 0 standby, 0 unknown
OK	Multipath 600508b1001c701a21d8812fdab171c4		OK - Type naa/sas, 1 active, 0 dead, 0 disabled, 0 standby, 0 unknown
OK	Object count		OK - Virtualmachines: 0
OK	Overall state		OK - Entity state: green, Power state: poweredOn

Add standalone ESXi hosts

The procedure is pretty much the same as that used to add hosts managed by the vCenter apart that all options have to be selected during the ruleset creation .

Create new host

Folder: Main directory > SkyIT > Milan

General Properties

Hostname: vmwaretst1

Basic settings

Permissions: empty (Default value)

Alias: empty (Default value)

IPv4 Address: 10.39.239.97

Parents: empty (Default value)

Monitored on site: skytest - Local site skytest (Default value)

Host tags

Agent type: Check_MK Agent (Server)

Criticality: Productive system (Default value)

Networking Segment: Local network (low latency) (Default value)

Linux_SkyIT: All SkyIT Linux servers in Production (Default value)

IP Address Family: IPv4 only (Default value)

Buttons: Save & go to Services, Save & Finish, Save & Test

Abort

This rule selects the vSphere agent instead of the normal Check_MK Agent and allows monitoring of VMWare ESX via the vSphere API. You can configure your connection settings here.

Rule Options

Description: vmwarest1
Comment:
Documentation-URL:
Rule activation: do not apply this rule

Check state of VMWare ESX via vSphere

vSphere User name: root
vSphere secret:
TCP Port number:
SSL certificate checking:
 Deactivated
 Use hostname
 Use other hostname:
Connect Timeout: 60 seconds
Retrieve information about...
 Host Systems
 Virtual Machines
 Datastores
 Performance Counters
 License Usage
Display ESX Host power state on: The ESX Host
Display VM power state on: The queried ESX system (vCenter / Host)
Spaces in hostnames: Replace with underscores
Type of query: Queried host is a host system
Placeholder VMs: Do no monitor placeholder VMs
Compatibility mode: Support ESX 4.1 (using slower PySphere implementation)

Conditions

Folder: Main directory
Host tags: Agent type: ignore, Criticality: ignore, Networking Segment: ignore, Linux_SkyIT: ignore, IP Address Family: ignore, monitor via SNMP: ignore, monitor via Check_MK Agent: ignore, IPv4: ignore, IPv6: ignore
Explicit hosts: Specify explicit host names: vmwarest1
 Negate: make rule apply for all but the above hosts

Save

This is a standalone hp dl 360g7 running ESXi free. The critical service is related to a power supply in failed state.

Services of Host vmwaretst1

WATO Host status Host/Svc notific. Host/Svc history

State	Service	Icons	Status detail
OK	Check_MK		OK - Agent version 6.0, execution time 0.9 sec
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found
OK	Check_MK HW/SW Inventory		OK - found 44 entries
OK	CPU utilization		OK - 3.1% used, 0.82GHz/26.82GHz, 2 sockets, 6 cores/socket, 24 threads
OK	Datastore IO SUMMARY		OK - Read: 53.00 kB/s, Write: 271.00 kB/s, Latency: 0.00 ms
OK	Disk IO SUMMARY		OK - Read: 53.00 kB/s, Write: 271.33 kB/s, Latency: 0.00 ms
OK	Filesystem LocalStorage		OK - 48.1% used (144.92 of 301.00 GB), trend: 0.00 B / 24 hours, uncommitted: 93.32 GB, provisioning: 79.1%
CRIT	Hardware Sensors		CRIT - Power Supply 2 Power Supply 2: Failure status --- Assert: Red (Sensor is operating under critical conditions) CRIT Off Line-Disabled: Red (Sensor is operating under critical conditions) CRIT VMware Rollup Health State: Red (Sensor is operating under critical conditions) CRIT
OK	HostSystem vmwaretst1		OK - power state: poweredOn
OK	Interface 1		OK - [vmnic0] (up) MAC: e4:11:5b:ea:a3:ec, 1 Gbit/s, in: 0.00 B/s(0.0%), out: 9.00 kB/s(0.0%)
OK	Interface 2		OK - [vmnic1] (up) MAC: e4:11:5b:ea:a3:ee, 1 Gbit/s, in: 68.00 kB/s(0.1%), out: 4.00 kB/s(0.0%)
OK	Interface 3		OK - [vmnic2] (up) MAC: e4:11:5b:ea:a3:e8, 1 Gbit/s, in: 0.00 B/s(0.0%), out: 0.00 B/s(0.0%)
OK	License Evaluation Mode		OK - 1 Key(s), used 1 out of 1 licenses
OK	Maintenance Mode		OK - System not in Maintenance mode

Virtual Machines additional checks

As soon as you will install check_mk agent on virtual machines, additional checks we'll added and, a great thing about that, is that performance metrics (cpu/ram) will be retrieved directly from vcenter or ESXi host and not from the OS. This is very important because in a VMware environment, whenever you look at performance, what it really important is to know the real resources assigned by the host and not those that OS believe to have. A good example is cpu ready where the guest report high cpu usage but in reality isn't having the right resources because there is competition on the host side. I won't go through the details because this is out of topic; if you want more informations about that, please have a look at the following link: <http://www.logicmonitor.com/blog/2013/02/25/a-tale-of-two-metrics-windows-cpu-or-vcenter-vm-cpu/>

After the agent installation on the guest, check_MK warned me about some missing services

w2012tst1

State	Service	Icons	Status detail
WARN	Check_MK Discovery		WARN - 8 unmonitored services (esx_vsphere_vm.snapshots:1, esx_vsphere_vm.cpu:1, esx_vsphere_vm.heartbeat:1, esx_vsphere_vm.guest_tools:1, esx_vsphere_vm.running_on:1, esx_vsphere_vm.services found)

Doing a new service discovery, they immediately appeared

w2012tst1			
State	Service	Icons	Status detail
OK	Check_MK		OK - Agent version 1.2.8p8, execution time 1.0 sec
OK	Check_MK Discovery		OK - no unmonitored services found, no vanished services found
OK	Check_MK HW/SW Inventory		OK - found 2091 entries, hardware changes
OK	CPU utilization		OK - 1.2% used, user perc: 0.6 %, privileged perc: 0.4 %, 1 CPUs
OK	Disk IO SUMMARY		OK - Read: 0.00 B/s, Write: 0.00 B/s, Average Read Wait: 0.00 ms, Average Write Wait: 0.00 ms
OK	DotNet Memory Management_Global_		OK - 0.00% time in GC
OK	ESX CPU		OK - demand is 0.016 Ghz, 1 virtual CPUs
OK	ESX Datastores		OK - Stored on LocalStorage (301.00 GB/51.9% free)
WARN	ESX Guest Tools		WARN - VMware Tools are installed, but the version is not current
OK	ESX Heartbeat		OK - Heartbeat status is green
OK	ESX Hostsystem		OK - Running on esxmitst1
OK	ESX Memory		OK - Host: 1.14 GB, Guest: 266.00 MB, Ballooned: 0.00 B, Private: 1.11 GB, Shared: 7.00 MB
OK	ESX Name		OK - w2012tst1
OK	ESX Snapshots		OK - No snapshots found
OK	Filesystem C:/		OK - 31.6% used (18.85 of 59.66 GB), trend: +82.97 MB / 24 hours
OK	Interface 1		OK - [Intel[R] 82574L Gigabit Network Connection] (Connected) 1 Gbit/s, in: 38.58 B/s(0.0%), out:

Managing SNMP Traps

Nowadays, every good Enterprise monitoring solution has the ability to manage incoming SNMP Traps but some do it better than others. I had a frustrating experience with some tools but Check_MK, as usual, does it really well and in a clear and simple way.

Our goal is:

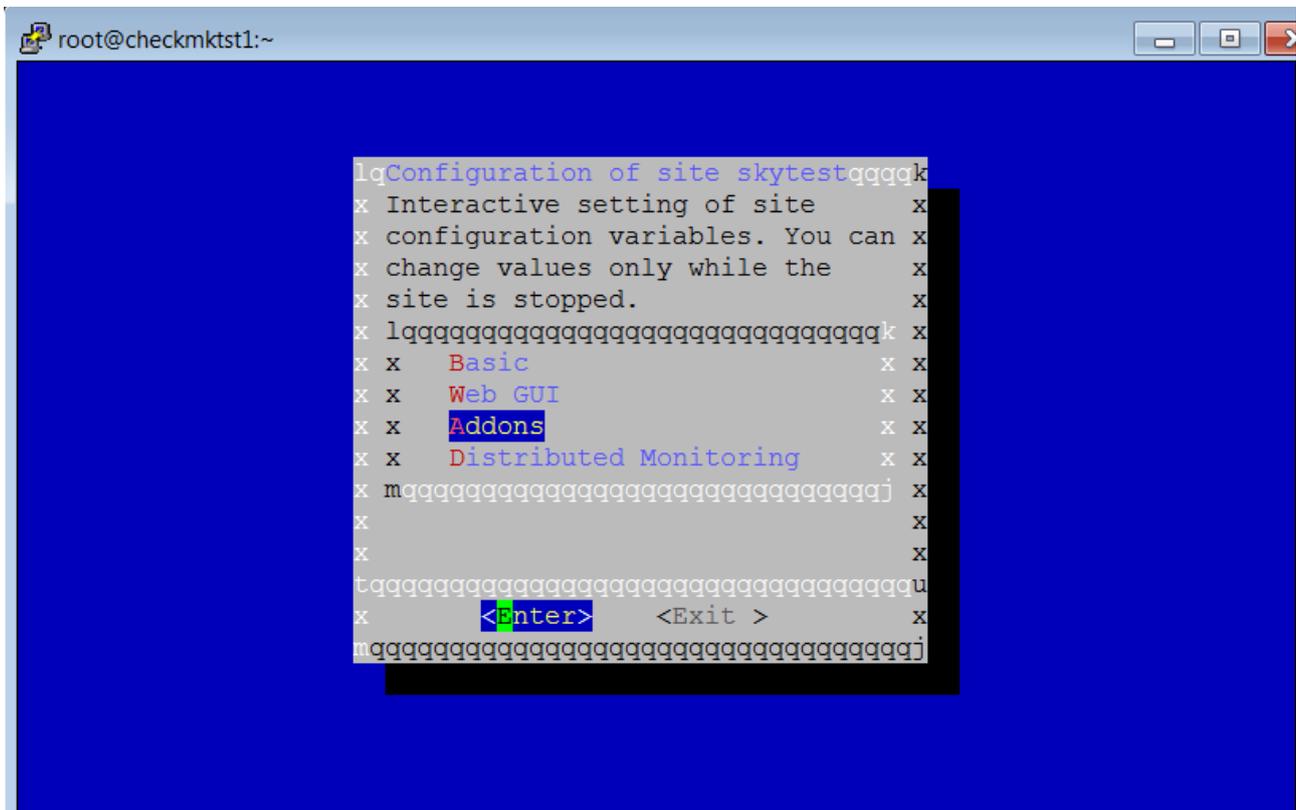
- receive incoming traps
- do a regex or filtering if necessary
- decide the level of criticality
- generate a service AUTOMATICALLY assigned to the monitored device
- AUTOCLEAR function meaning that if we receive an "OK" trap, the service should change from red (critical) to green (OK)

I'm going to list all the required steps but please note that I found the official documentation a little bit outdated and, depending on your environment (distribution as well Check_MK version and installation), some additional steps could be required.

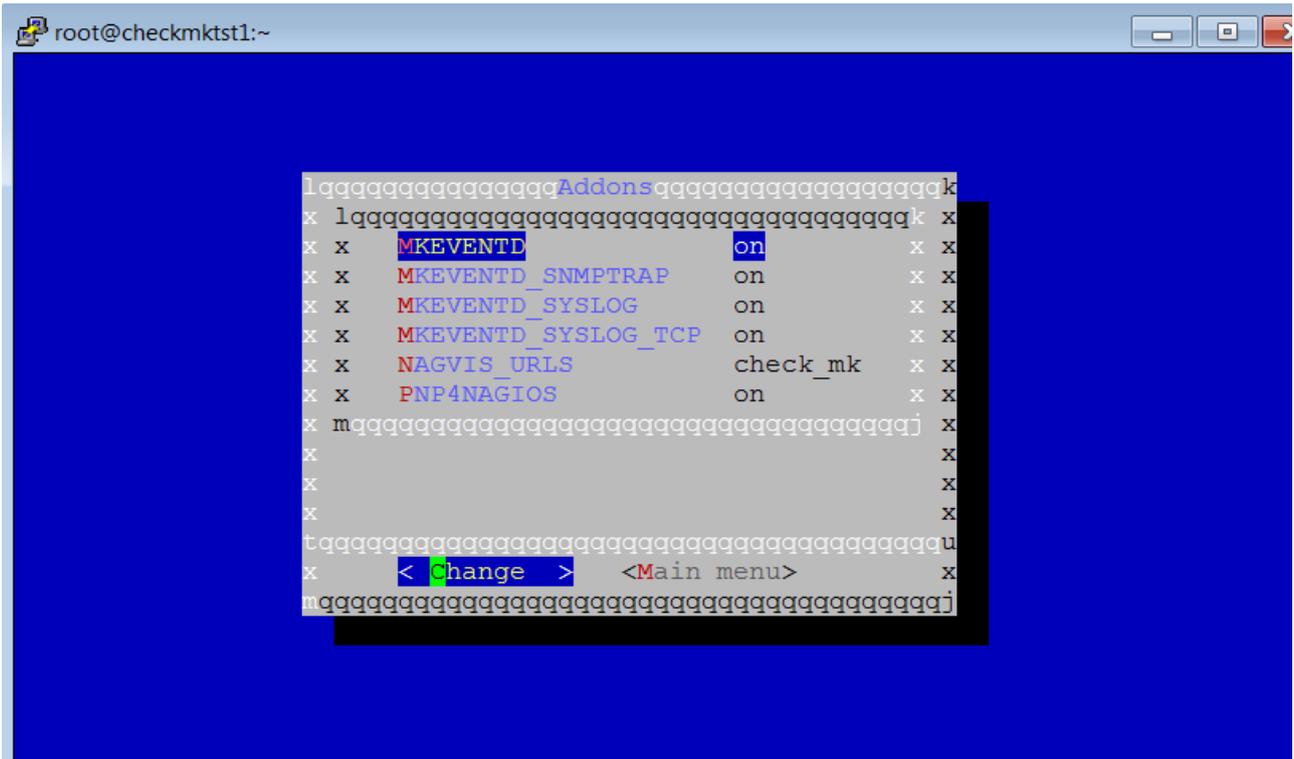
- Connect to Check_MK host and, from the command line, run:

```
[root@checkmktst1 ~]# su - mysite
OMD[mysite]:~$ omd config
```

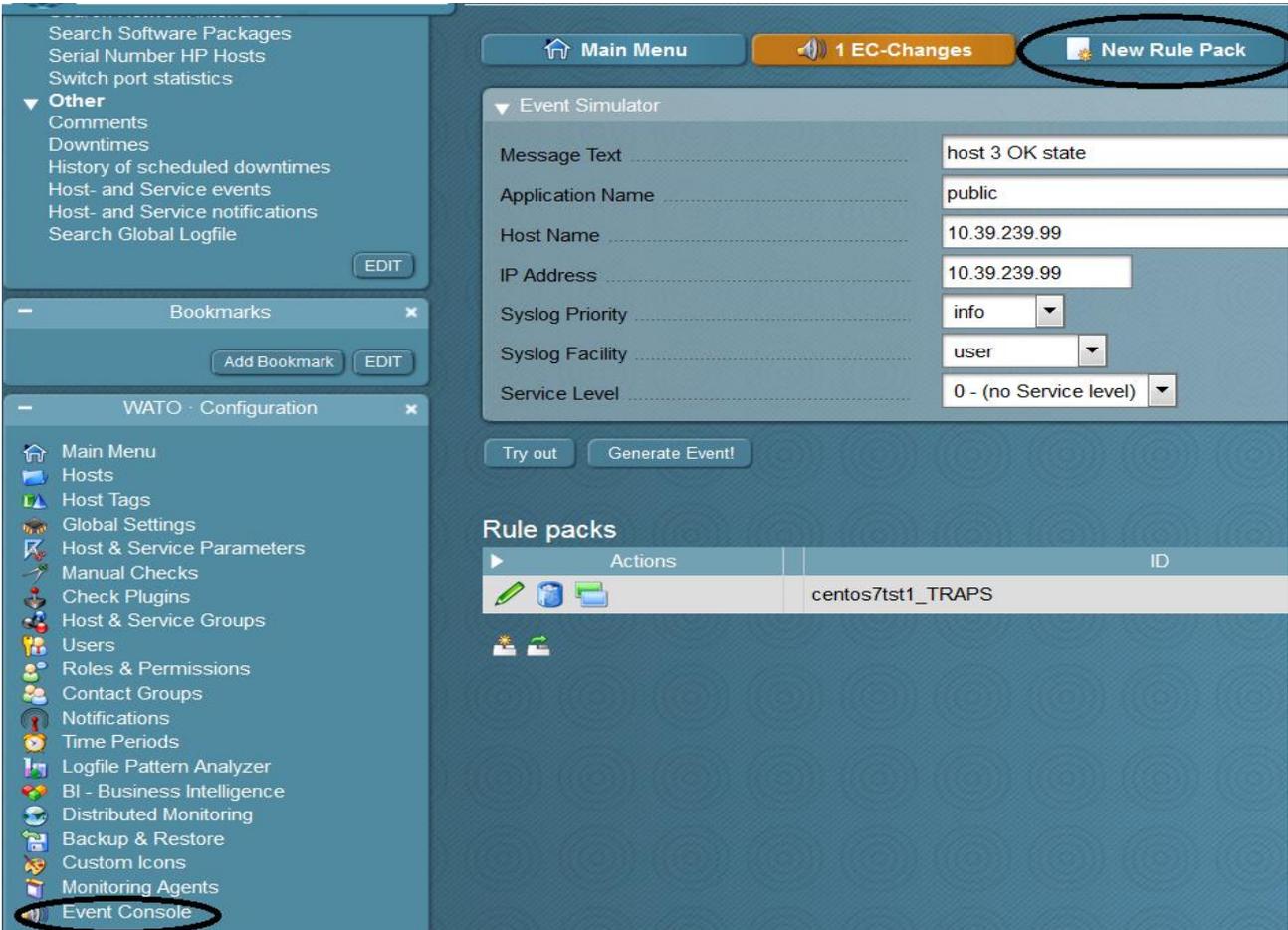
- Enable embedded *MKEVENTD_SNMPTRAP* and *MKEVENTD_SYSLOG*



```
root@checkmktst1:~
Configuration of site skytest
x Interactive setting of site x
x configuration variables. You can x
x change values only while the x
x site is stopped. x
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x Basic x x
x x Web GUI x x
x x Addons x x
x x Distributed Monitoring x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
x x
x tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqU
x <Enter> <Exit > x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```



- WATO-Configuration, Event Console, New Rule Pack



- Click the button *Edit the rules in this pack*

Event Console Rule Packages

Main Menu | 1 EC-Changes | New Rule Pack | Reset

Event Simulator

Message Text: host 3 OK state

Application Name: public

Host Name: 10.39.239.99

IP Address: 10.39.239.99

Syslog Priority: info

Syslog Facility: user

Service Level: 0 - (no Service level)

Try out | Generate Event!

Rule packs

Actions	ID
  	centos7tst1_TRAPS

Edit the rules in this pack

- Create a new rule like this

Rule Package centos7tst1_TRAPS

Rule Packs | No EC-Changes | **New Rule** | Properties

Event Simulator

Message Text: host 3 OK state

Application Name: public

Host Name: 10.39.239.99

IP Address: 10.39.239.99

Syslog Priority: info

Syslog Facility: user

Service Level: 0 - (no Service level)

Try out | Generate Event!

Actions	ID	State	Priority	Facility	Service Level
  	centos7tst1_HOST_CRITICAL	CRIT			(no Service level)

Main Menu

Rule Packs

1 EC-Changes

Clear Rule

Rule Properties

Rule ID centos7tst1_HK
Description Critical if host alarm
Comment
Documentation-URL
Rule activation do not apply this rule

Matching Criteria

Text to match host (*) critical state
Match host
Match original source IP address
Match syslog application (tag)
Match syslog priority
Match syslog facility
Match service level
Match only during timeperiod
Text to cancel event(s) host (*) OK state
Syslog priority to cancel event
Invert matching Negate match. Execute this rule if the upper conditions are **not** fulfilled.

Outcome & Action

Rule type Normal operation - process message according to action settings
State CRIT
Service Level 0 - (no Service level)
Contact Groups
Actions Send monitoring notification
Actions when cancelling Send monitoring notification
Do Cancelling-Actions when Always when an event is being cancelled
Automatic Deletion Delete event immediately after the actions

Counting & Timing

Count messages in defined interval
Expect regular messages
Delay event creation
Limit event lifetime

Rewriting

Rewrite message text
Rewrite hostname
Rewrite application
Add comment
Add contact information

Save

- Reload the configuration

The screenshot shows the 'Event Console Rule Packages' interface. The top navigation bar contains several buttons: 'Main Menu', '3 EC-Changes' (highlighted with a red circle), 'New Rule Pack', 'Reset Counters', and 'Server Status'. Below this is the 'Event Simulator' section, which includes a form with the following fields and values:

- Message Text: host 3 OK state
- Application Name: public
- Host Name: 10.39.239.99
- IP Address: 10.39.239.99
- Syslog Priority: info
- Syslog Facility: user
- Service Level: 0 - (no Service level)

Below the form are two buttons: 'Try out' and 'Generate Event!'. At the bottom of the interface is a 'Rule packs' table:

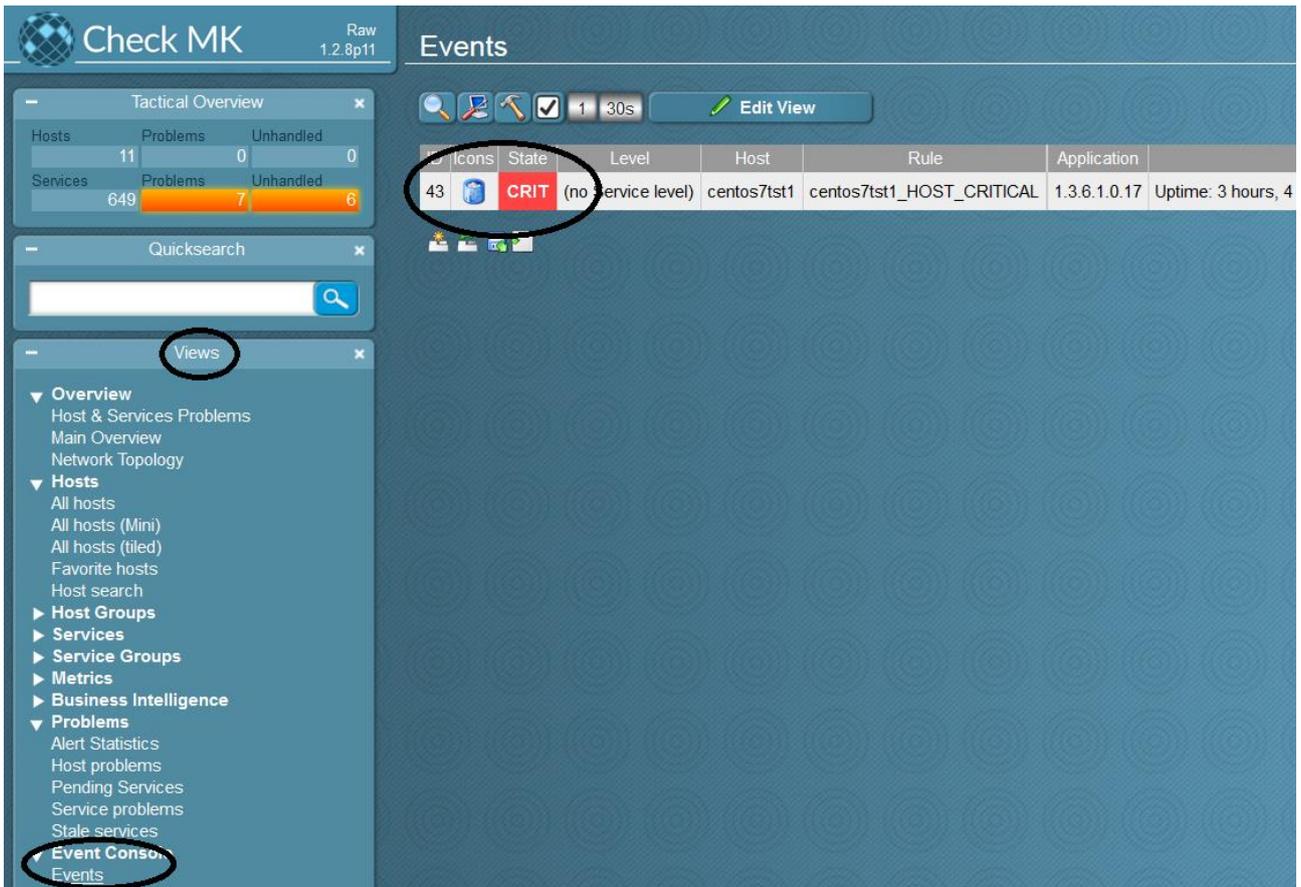
Actions	ID	
  	centos7tst1_TRAPS	centos7tst1_TRAPS

- Test the configuration

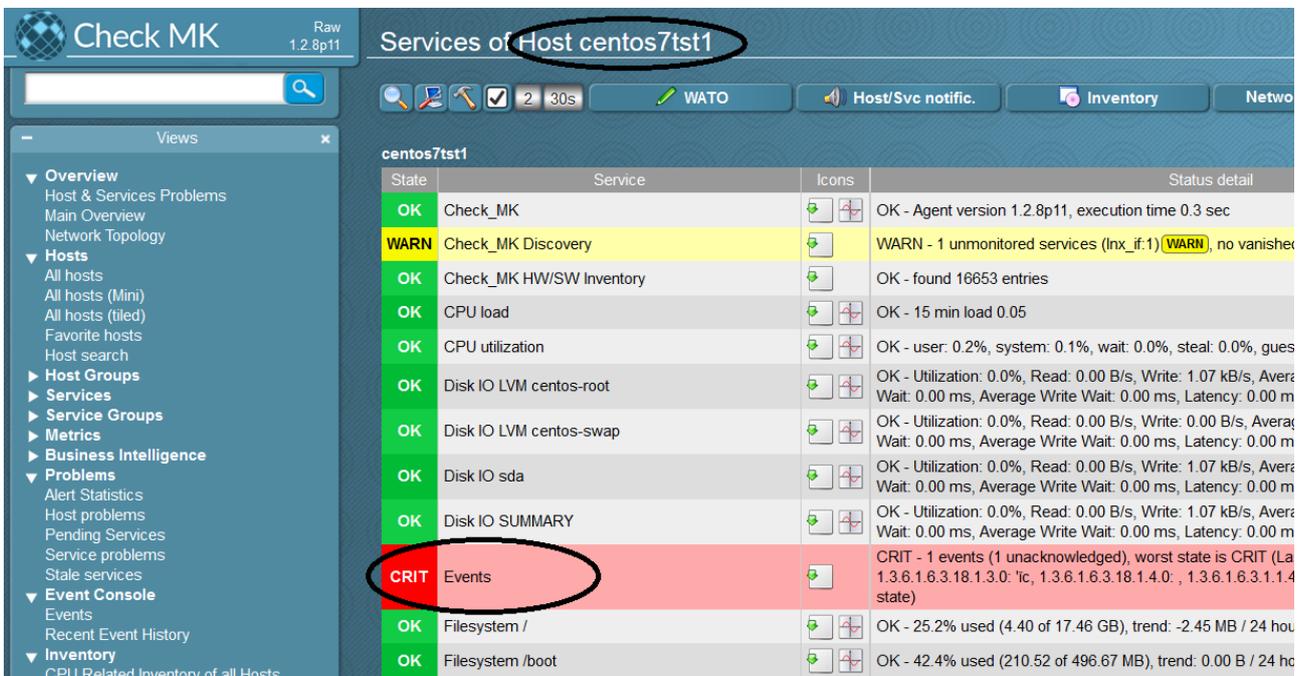
From the "centos7tst1" host run:

```
[root@centos7tst1 ~]# snmptrap -v 1 -c public 10.39.239.100 .1.3.6.1 10.39.239.99 6 17 ''
.1.3.6.1 s "host 3 critical state"
```

- Check if event has been created *WATO-View, Events*



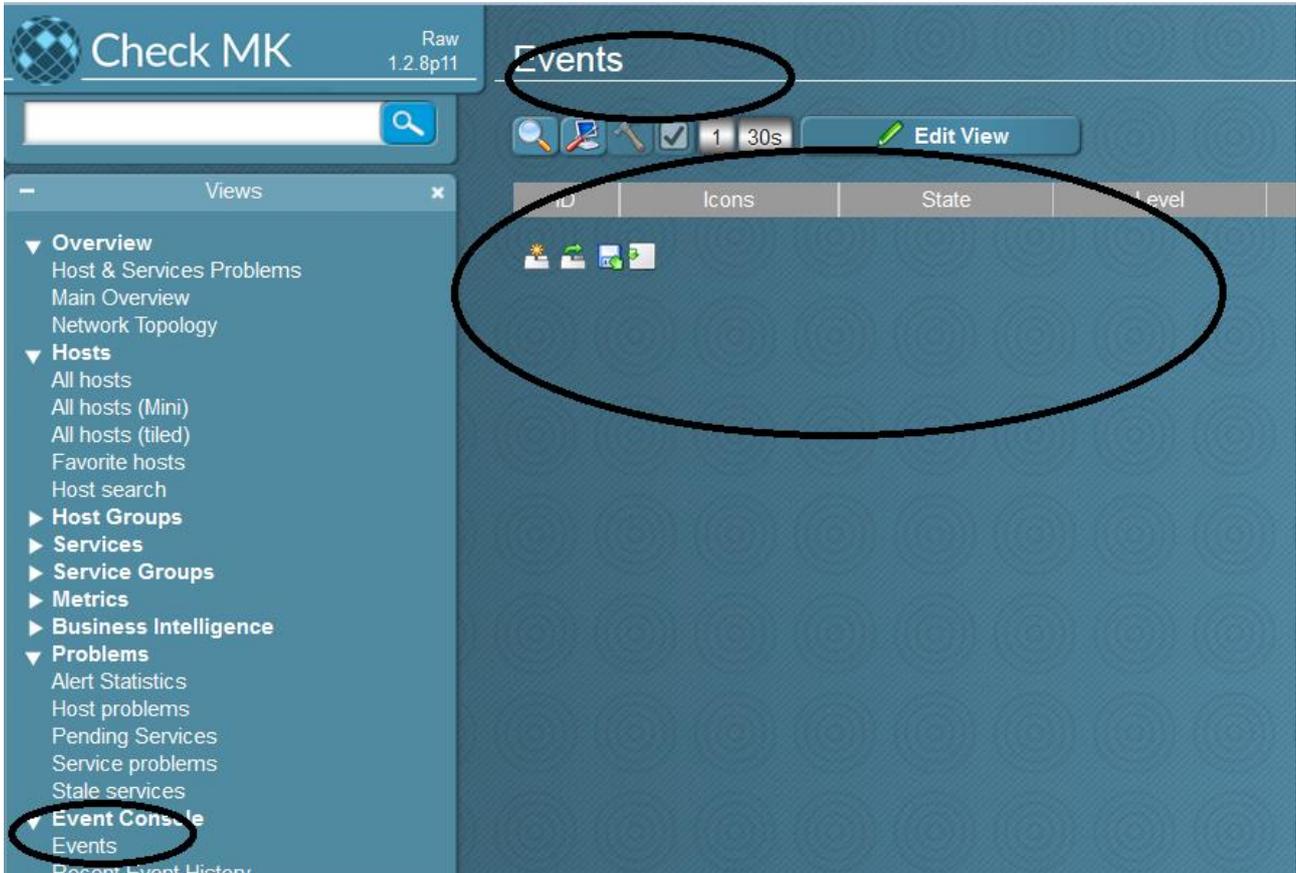
- Check that event has been AUTOMATICALLY associated to the correct host “centos7tst1”



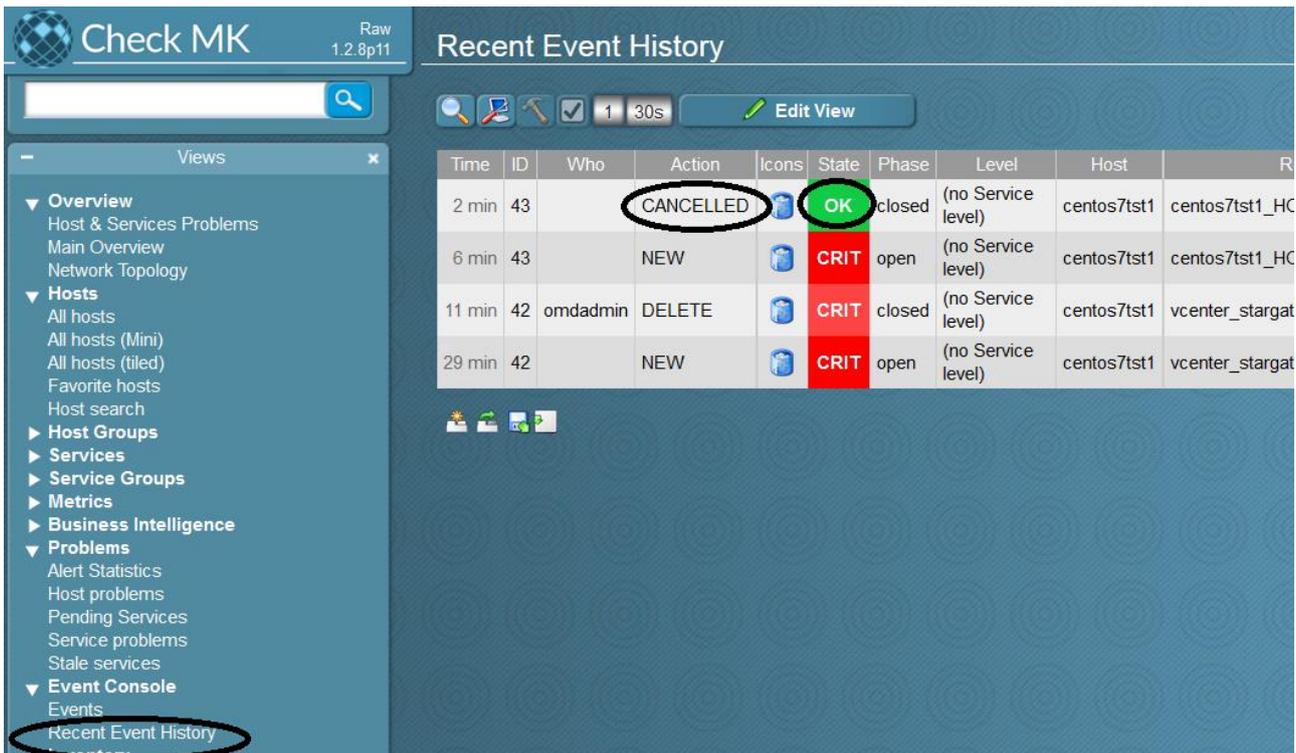
- Test the AUTOCLEAR mechanism is working correctly sending the following trap from the remote host

```
[root@centos7tst1 ~]# snmptrap -v 1 -c public 10.39.239.100 .1.3.6.1 10.39.239.99 6 17 '' .1.3.6.1 s "host 3 OK state"
```

No open events should be displayed in *WATO-Views, Events*



But in *WATO-Views, Recent Event History* we can see that even the OK message has been received



The *Event* service should be now green (OK)

The screenshot shows the Check_MK interface for the host 'centos7tst1'. The left sidebar contains a navigation menu with categories like Overview, Hosts, Host Groups, Services, Metrics, Business Intelligence, and Problems. The 'Hosts' section is expanded, and 'all hosts' is selected. The main area displays a table of services for the host. The 'Events' service is circled in green and has a green status indicator. The 'Check_MK Discovery' service is highlighted in yellow and has a yellow status indicator. The 'Check_MK' service is circled in black and has a green status indicator.

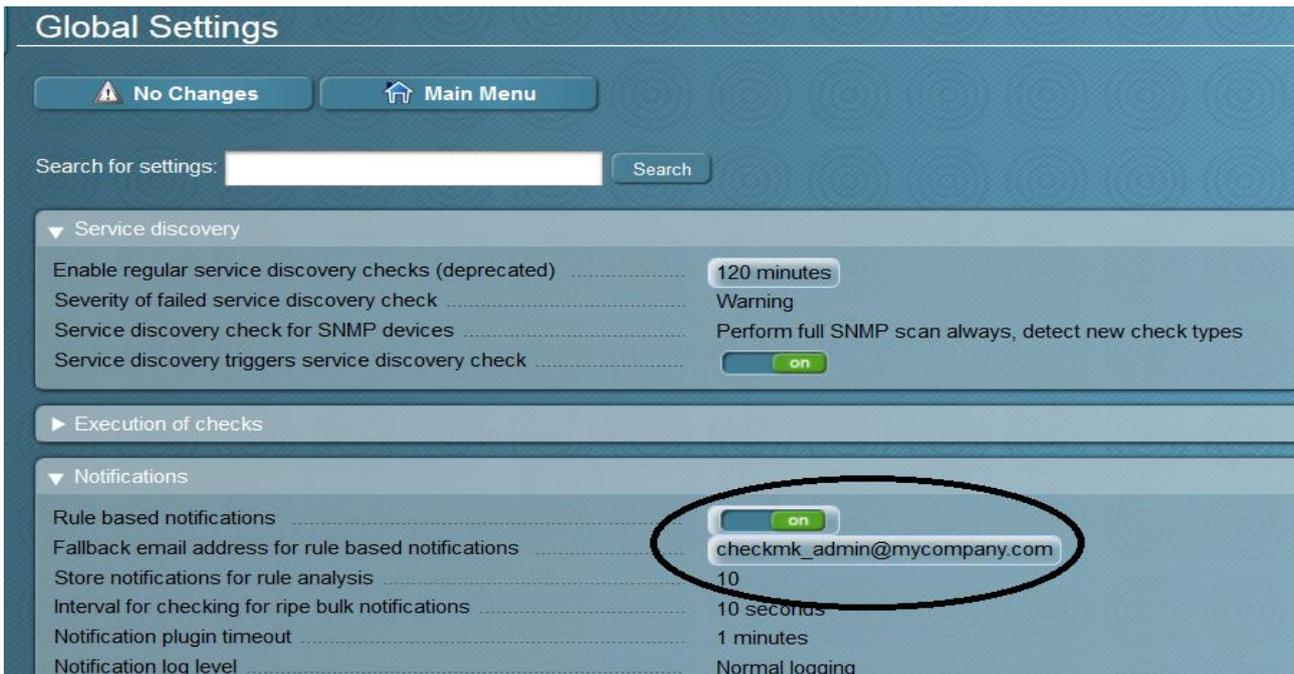
State	Service	Icons	Status details
OK	Check_MK		OK - Agent version 1.2.8p11, execution time 0.3 sec
WARN	Check_MK Discovery		WARN - 1 unmonitored services (lnx_if:1) (WARN), no v
OK	Check_MK HW/SW Inventory		OK - found 16653 entries
OK	CPU load		OK - 15 min load 0.05
OK	CPU utilization		OK - user: 0.2%, system: 0.2%, wait: 0.0%, steal: 0.0%
OK	Disk IO LVM centos-root		OK - Utilization: 0.0%, Read: 0.00 B/s, Write: 1.46 kB/s, Wait: 0.00 ms, Average Write Wait: 0.05 ms, Latency:
OK	Disk IO LVM centos-swap		OK - Utilization: 0.0%, Read: 0.00 B/s, Write: 0.00 B/s, Wait: 0.00 ms, Average Write Wait: 0.00 ms, Latency:
OK	Disk IO sda		OK - Utilization: 0.0%, Read: 0.00 B/s, Write: 1.46 kB/s, Wait: 0.00 ms, Average Write Wait: 0.06 ms, Latency:
OK	Disk IO SUMMARY		OK - Utilization: 0.0%, Read: 0.00 B/s, Write: 1.46 kB/s, Wait: 0.00 ms, Average Write Wait: 0.06 ms, Latency:
OK	Events		OK - no events for centos7tst1/10.39.239.99

Managing Notifications

Notifications are quite a complex topic and Check_MK works very hard to make them as flexible as possible. Once again the best explanation of the thinking behind this comes from the Check_MK documentation: https://mathias-kettner.de/checkmk_rbn.html

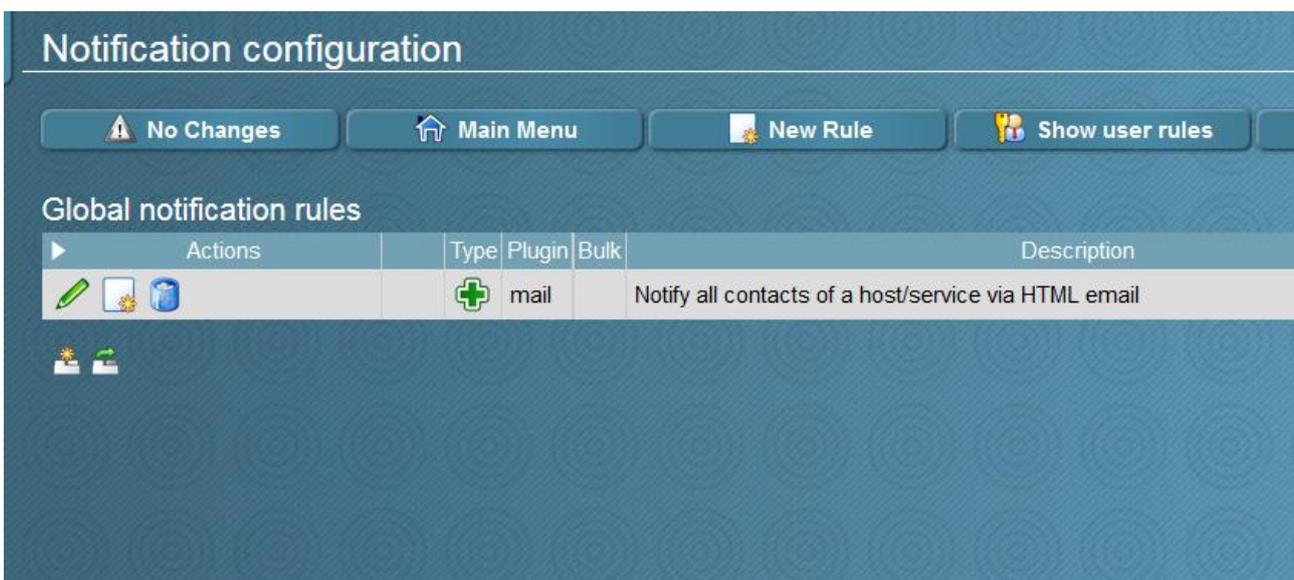
Basically, notifications are managed using the new *RBN* (Rule Based Notifications) that add extra flexibility to the previous mechanism called *Flexible Notifications* by providing the separation of contact-assignment and notification.

The first step is to enable RBN and a fallback address



The screenshot shows the 'Global Settings' interface. Under the 'Notifications' section, the 'Rule based notifications' toggle is set to 'on' and is circled in black. The 'Fallback email address for rule based notifications' field is also circled and contains the email address 'checkmk_admin@mycompany.com'. Other settings include 'Enable regular service discovery checks (deprecated)' set to '120 minutes', 'Severity of failed service discovery check' set to 'Warning', 'Service discovery check for SNMP devices' set to 'Perform full SNMP scan always, detect new check types', and 'Service discovery triggers service discovery check' set to 'on'.

Now create a *Notification Rule* or change the existing one: *WATO, Notifications*



The screenshot shows the 'Notification configuration' page. At the top, there are buttons for 'No Changes', 'Main Menu', 'New Rule', and 'Show user rules'. Below this is a table titled 'Global notification rules'.

Actions	Type	Plugin	Bulk	Description
  		mail		Notify all contacts of a host/service via HTML email

At the bottom left, there are icons for a sun and a mail envelope.

There are plenty of parameters that should satisfy all needs

Edit notification rule 0 omdadmin (admin) 16:53

[← All Rules](#)

General Properties

Description: Notify all contacts of a host/service via HTML email

Comment:

Documentation-URL:

Rule activation: do not apply this rule

Overriding by users: allow users to deactivate this notification

Notification Method

Notification Method: HTML Email

Call with the following parameters:

- From: Address
- Reply-To: Address
- Subject for host notifications
- Subject for service notifications
- Information to be displayed in the email body
- URL prefix for links to Check_MK
- Display graphs among each other
- Notification sort order for bulk notifications

Notification Bulking:

Contact Selection

All contacts of the notified object: Notify all contacts of the notified host or service.

All users: Notify all users

All users with an email address: Notify all users that have configured an email address in their profile

The following users:

The members of certain contact groups:

The following explicit email addresses: externalusr@externalcompany.com

Restrict by custom macros:

Restrict by contact groups:

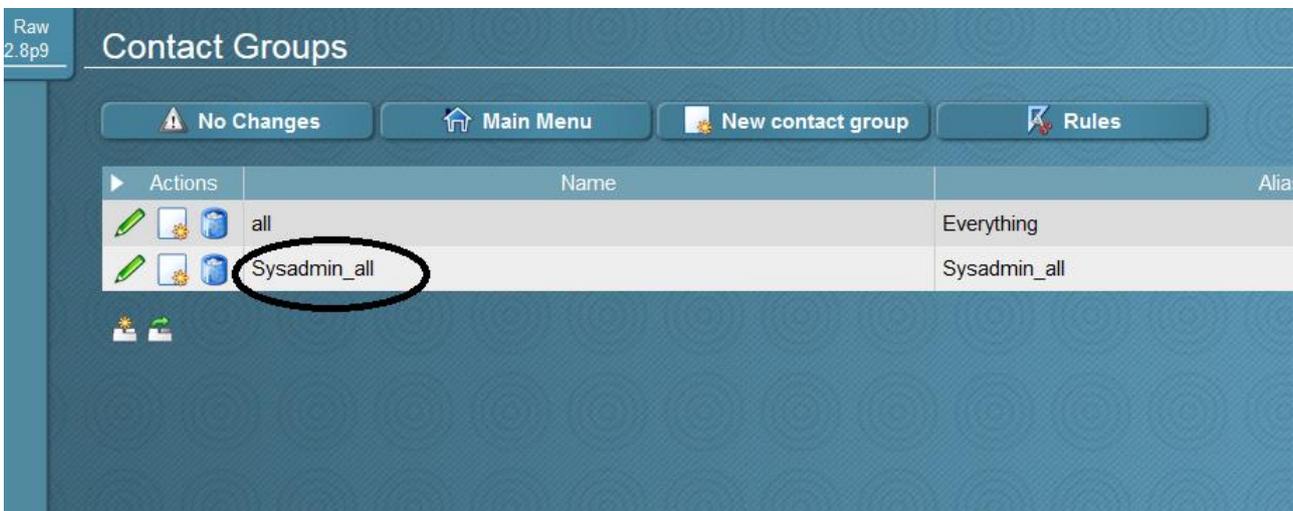
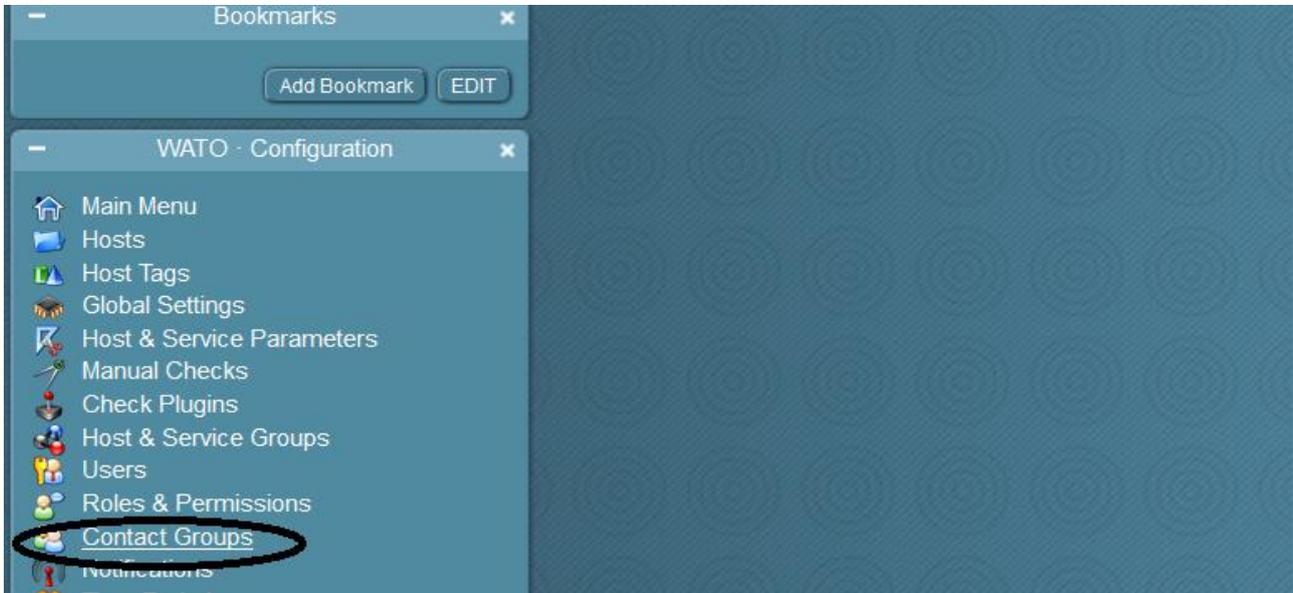
Conditions

- Folder:
- Match Host Tags:
- Match Host Groups:
- Match only the following hosts:
- Exclude the following hosts:
- Match Service Groups:
- Match only the following services:
- Exclude the following services:
- Match the following check types:
- Match the output of the check plugin:
- Match Contacts:
- Match Contact Groups:
- Match service level:
- Match only during timeperiod:
- Match host event type:
- Match service event type:
- Restrict to nth to mth notification:
- Throttle periodic notifications:
- Match notification comment:
- Event Console alerts:

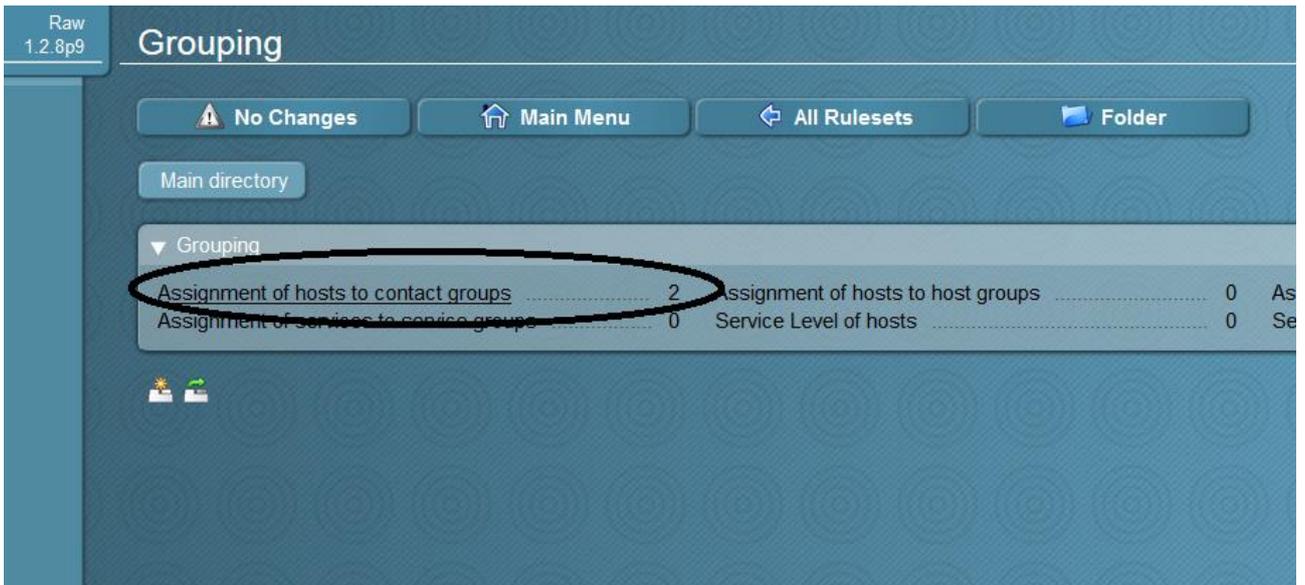
[Save](#)

Contact group

Sometimes it is necessary to notify all people who are members of a specified contact group. This is done with the module *Contact Groups*. In this example, I created the *Sysadmin_all* contact group cloning the existing one called *all*



Important: put some hosts/services into that contact group. WATO: *Host & Service Parameters / Grouping / Assignment of hosts/services to contact groups*.



Create a user, enter an email address and put him into that contact group: WATO: *Users & Contacts*

▼ Identity

Username realem

Full name test

Email address test@mycompany.com

Pager address

▼ Security

Authentication Normal user login with password

password:

repeat: (optional)

Enforce change: Change password at next login or access

Automation secret for machine accounts



Disable password disable the login to this account

Roles Administrator

Guest user

Normal monitoring user

▼ Contact Groups

Everything

Sysadmin_all

Activate Changes in WATO

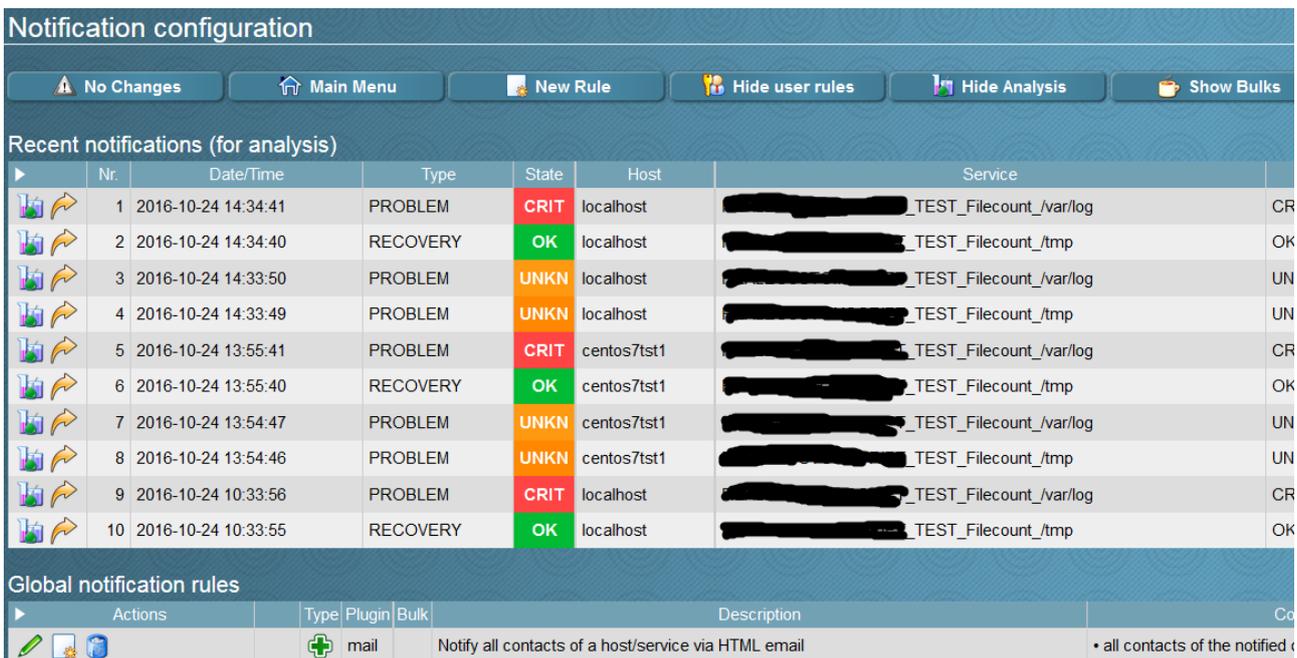
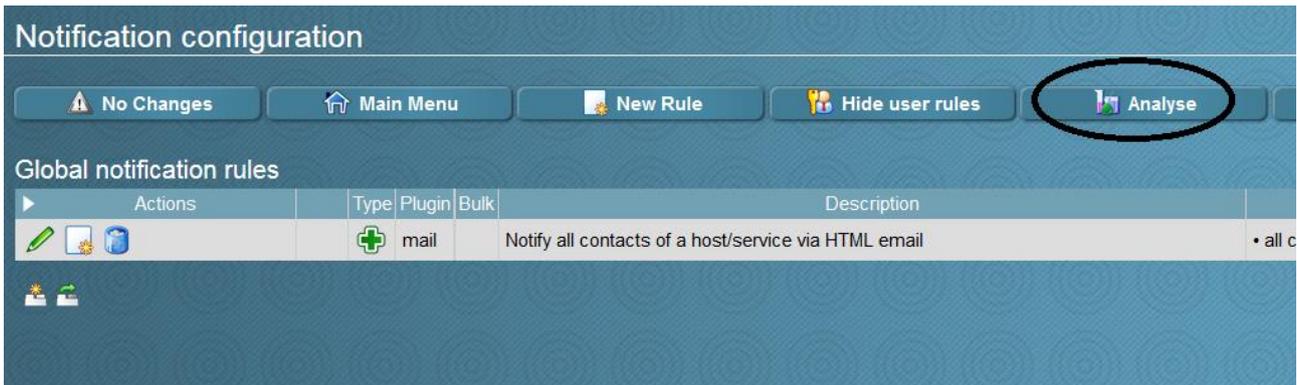
Analysis

To have alert notifications sent via email, make sure that your monitoring server is correctly setup so that it can send them. Test this with

```
echo "Mailbody" | mail -s "Testsubject" test@mycompany.com
```

If everything is setup properly, you should receive emails as soon as a *CRITICAL* service is detected. I also suggest to check the email log file, in my case */var/log/maillog* when troubleshooting this.

An *Analysis* tool is also available in the *Notifications Configuration* menu



Check_MK Update

The update process is generally very simple but, before proceeding, don't forget to take a backup and read the release notes very carefully. Problems could arise (especially with major upgrades) and it's good to have a backout process just in case.

Package installation

Download the latest package for your distribution and install it as shown:

```
[root@checkmktst1 ~]# cd /tmp/
[root@checkmktst1 tmp]# wget https://mathias-kettner.de/support/1.2.8p13/check-mk-raw-1.2.8p13-e17-36.x86_64.rpm
--2016-10-21 11:33:06-- https://mathias-kettner.de/support/1.2.8p13/check-mk-raw-1.2.8p13-e17-36.x86_64.rpm
Resolving mathias-kettner.de (mathias-kettner.de)... 178.248.246.154
Connecting to mathias-kettner.de (mathias-kettner.de)|178.248.246.154|:443... connected.
HTTP request sent, awaiting response... 200 OK
```

```

Length: 60640908 (58M) [application/x-redhat-package-manager]
Saving to: 'check-mk-raw-1.2.8p13-el7-36.x86_64.rpm'

91% [=====> ] 55,312,384 924KB/s in 47s

2016-10-21 11:33:53 (1.13 MB/s) - Connection closed at byte 55312384. Retrying.

--2016-10-21 11:33:54-- (try: 2) https://mathias-kettner.de/support/1.2.8p13/c
heck-mk-raw-1.2.8p13-el7-36.x86_64.rpm
Connecting to mathias-kettner.de (mathias-kettner.de)|178.248.246.154|:443... co
nected.
HTTP request sent, awaiting response... 206 Partial Content
Length: 60640908 (58M), 5328524 (5.1M) remaining [application/x-redhat-package-m
anager]
Saving to: 'check-mk-raw-1.2.8p13-el7-36.x86_64.rpm'

100%[++++++>] 60,640,908 1.76MB/s in 2.9s

2016-10-21 11:33:57 (1.76 MB/s) - 'check-mk-raw-1.2.8p13-el7-36.x86_64.rpm' save
d [60640908/60640908]

[root@checkmktst1 tmp]# rpm -Uvh check-mk-raw-1.2.8p13-el7-36.x86_64.rpm
Preparing... ##### [100%]
Updating / installing...
 1:check-mk-raw-1.2.8p13-el7-36 ##### [100%]
New default version is 1.2.8p13.cre.

```

Switching to the new version

Switch to the new version using the OMD command:

```

[root@checkmktst1 tmp]# su - mysite
Last login: Thu Oct 20 16:59:39 CEST 2016 on pts/0
OMD[mysite]:~$ omd stop
Removing Crontab...OK
Stopping dedicated Apache for site mysite.....OK
Stopping nagios.....OK
Stopping npcd...OK
Stopping rrdcached...waiting for termination...OK
Stopping mkeventd...killing 15658.....OK
OMD[mysite]:~$ omd update

```

```
lqqqqqqqqqChoose target versionqqqqqqqqqqkk
x Please choose the version this site should x
x be updated to x
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x 1.2.8p9.cre Version 1.2.8p9.cre x x
x x 1.2.8p13.cre Version 1.2.8p13.cre x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
x x
x x
tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
x <Update now> < Cancel > x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

```
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x You are going to update the site skytest from x
x version 1.2.8p11.cre to version 1.2.8p13.cre. x
x This will include updating all of you x
x configuration files and merging changes in the x
x default files with changes made by you. In case x
x of conflicts your help will be needed. x
tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
x <Update!> < Abort > x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

OMD[mysite]:~\$ omd start

Check MK Raw 1.2.8p13 All hosts

Services 650 Problems 6 Unhandled 5

Quicksearch

Views

- Overview
 - Host & Services Problems
 - Main Overview
 - Network Topology
- Hosts
 - All hosts
 - All hosts (Mini)
 - All hosts (tiled)
 - Favorite hosts

Local site skytest

state	Host	Icons	OK	Wa	Un	Cr	Po
UP	centos7tst1		27	0	0	2	0
UP	[REDACTED]		66	0	0	0	0
UP	[REDACTED]		66	1	0	0	0
UP	sysnet		48	0	0	0	0

Conclusion

I don't claim that check_MK is the best existing monitoring tool simply because I didn't try all existing products but I can safely say that it is the best I have ever used. I have been also impressed by their clearness about the product's price: no complicated licensing model that force you to contact some sales manager (but they are willing to help you and know very well the product). In my opinion the price of the Enterprise Edition is ridiculous compared to other products and provides some nice additional features (apart the support) that it's worth a try.

The documentation is also very good even there are room for improvements and I have been able to monitor lot of enterprise class devices in few days without any headache thank also to the check_MK mailing list that is very active.