

Installation Agent für andere Betriebssysteme ohne systemd / xinetd (Anbindung per ssh) tar.gz

Beschreibung

Es gibt Systeme auf denen gibt es kein Systemd und xinetd.

Der Checkmk Service lauscht normalerweise auf einen TCP Port.

Da es aber keine Services ohne xinetd und systemd gibt, gibt es eine weitere Methode. und zwar das abrufen des Agents über ssh.

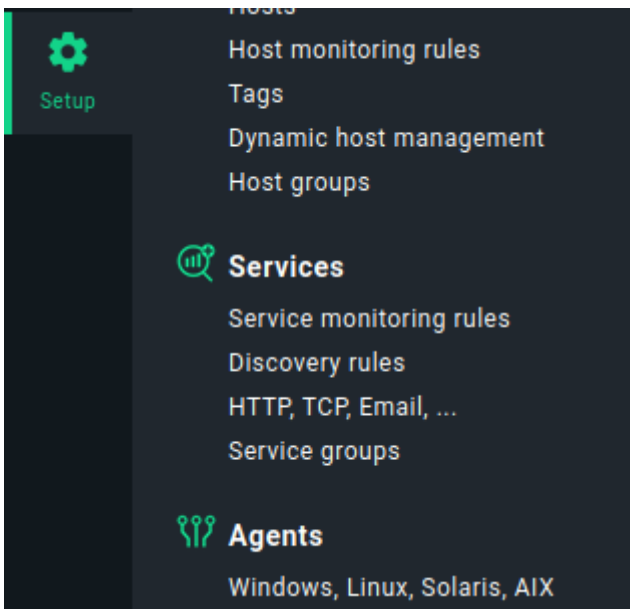
Allerdings funktioniert für diese Methode die Agent Bakery natürlich nicht.

Agents für die Hosts installieren / entpacken

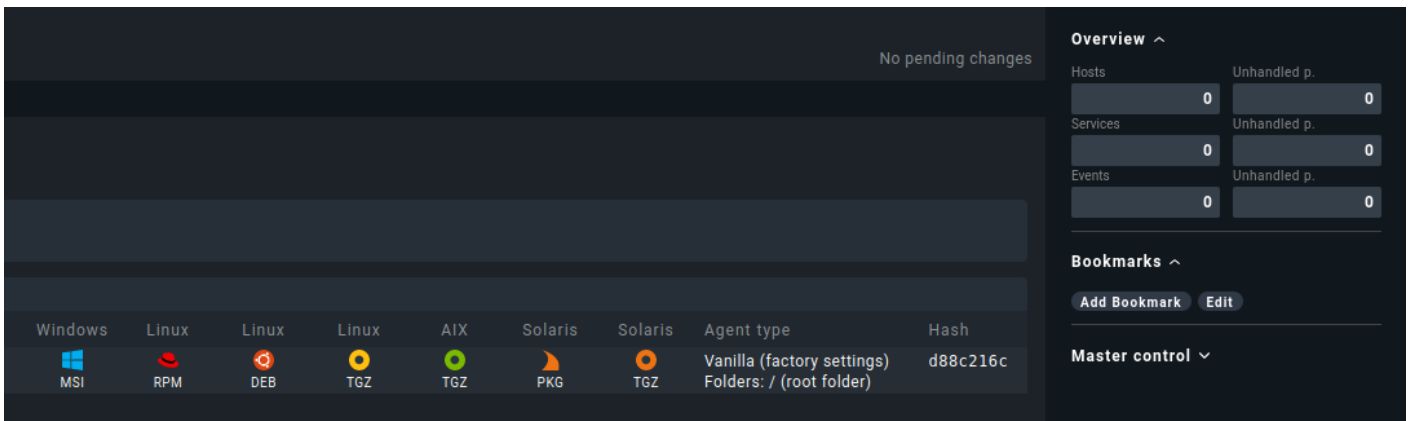
Auf dem zu überwachenden Host wo der Agent installiert werden soll, ein Verzeichnis checkmk_setup im root verzeichnis erstellen

```
mkdir -p /root/checkmk_setup
```

Nun mit dem Webbrowser auf dem Checkmk Server einloggen und dann Dazu auf Setup -> Agents -> Windows, Linux, Solaris , AIX anklicken



Nun auf tar.gz bei Linux mit dem gelben Symbol anklicken und downloaden



Nun per scp die Datei auf den zu überwachenden Server ins Home Verzeichnis von root/checkmk_stup übertragen check_mk übertragen

```
scp Downloads/check-mk-agent_2.1.0p16-d88c216c6ad53a29.tar.gz root@192.168.0.30:/root/checkmk_stup
```

Nun die check-mk-agent_2.1.0p16-d88c216c6ad53a29.tar.gz tar auf dem zu überwachendem Host entpacken

```
tar -xf check-mk-agent_2.1.0p16-d88c216c6ad53a29.tar.gz -C /
```

SSH Schlüssel falls nicht schon vorhanden auf der Checkmkinstanz erstellen

Auf dem Checkmk server inloggen.
Dann in die instanz einloggen

```
omd su <instanzname>
```

Beispiel

```
omd su monitoring
```

Nun den Schlüssel erstellen ohne passphrase

```
ssh-keygen -t ed25519
```

Ausgabe

```
OMD[monitoring]:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/omd/sites/monitoring/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /omd/sites/monitoring/.ssh/id_ed25519.
Your public key has been saved in /omd/sites/monitoring/.ssh/id_ed25519.pub.
The key fingerprint is:
cc:87:34:d2:ed:87:ed:f7:1b:ec:58:1f:7c:23:00:e2 monitoring@mycmkserver
The key's randomart image is:
+--[ED25519 256]--+
|          |
|   ..    |
|  ..+..   |
|   =.+..o |
|    ES+.o |
|   .o.o   |
|     ...B.|
|     .+.*|
|     .o+  |
+-----+
```

Die Schlüssel liegen dann im `.ssh` Verzeichnis des Monitoring Verzeichnisses.
Dateien anzeigen

```
ll .ssh
```

Ausgabe

```
OMD[monitoring]:~$ ll .ssh
total 8
-rw----- 1 monitoring monitoring 1679 Feb 20 14:18 id_ed25519
```

```
-rw-r--r-- 1 monitoring monitoring 398 Feb 20 14:18 id_ed25519.pub
```

Der private Schlüssel heißt `id_ed25519` und ist nur für den Instanzbenutzer lesbar (`-rw-----`) — und das ist auch gut so! Der öffentliche Schlüssel `id_ed25519.pub` sieht etwa so aus:

Ausgabe:

```
OMD[monitoring]:~$ cat .ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGb6AaqRPIbEmDnBkeIW3Q6Emb5lr2QEbWEQLmA5pb48
monitoring@mycmkserver
```

Nun auf dem zu überwachendem Host in einer zweiten SSH-Sitzung einloggen.
und die `authorized_keys` Datei bearbeiten

```
nano /root/.ssh/authorized_keys
```

Dort am ende den öffentlichen Schlüssel eintragen.

Davor aber noch ein command, der dient dazu, das nur der Agent aufgerufen werden kann und auch gleichzeitig beim einloggen aufgerufen wird.

Die ganze zeile in der `authorized_keys` Datei sähe dann so aus

```
command="/usr/bin/check_mk_agent" ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIGb6AaqRPIbEmDnBkeIW3Q6Emb5lr2QEbWEQLmA5pb48
monitoring@mycmkserver
```

Nun ausloggen.

Testen der Verbindung

Nun auf dem Checkmkserver wieder in die instanz einloggen, wenn nicht noch von der vorherigen Sitzung eingeloggt

```
omd su <instanzname>
```

Beispiel

```
omd su monitoring
```

Nun einfach `ssh root@<iphostname>` und den Fingerprint des Schlüssels mit der Eingabe von `yes` bestätigen.

Der Fingerprint wird nur beim ersten mal abgefragt.

```
ssh root@<ipfromtargetserver>
```

Ausgabe

```
OMD[monitoring]:~$ ssh root@myhost23
The authenticity of host 'myhost23 (10.11.12.13)' can't be established.
ECDSA key fingerprint is SHA256:IWgVK+LtsMgjHUbdsA1FK12PdmVQGqaEY4TE8TEps3w.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
<<<check_mk>>>
Version: 2.1.0b6
AgentOS: linux
Hostname: myhost123
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
<<<df>>>
```

fertig, Verbindung steht.

Einbinden des Hosts in Checkmk

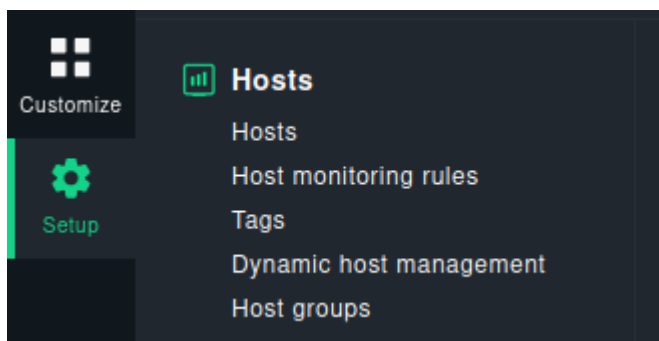
Nun da die SSH Verbindung hergestellt ist, muss jetzt die SSH verbindung im Host in Checkmk eingetragen werden.

Dazu in der Weboberfläche von check mk gehen, den Host anlegen wenn noch nicht geschehen. Sollte der Host schon bestehen. Kann das anlegen natürlich übersprungen werden.

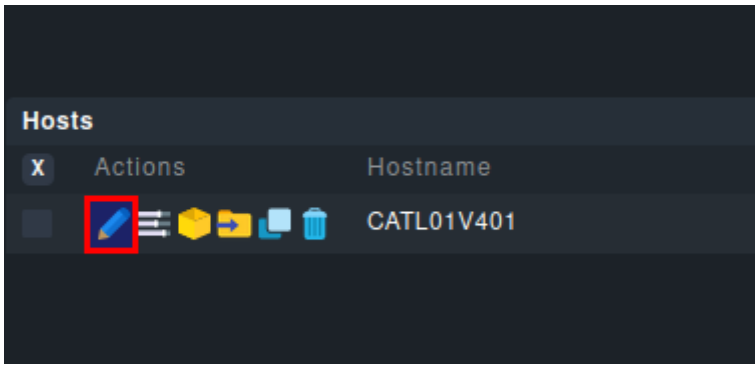
Wichtig ist nur das in den Host Einstellungen folgendes gemacht ist

Hostanpassen

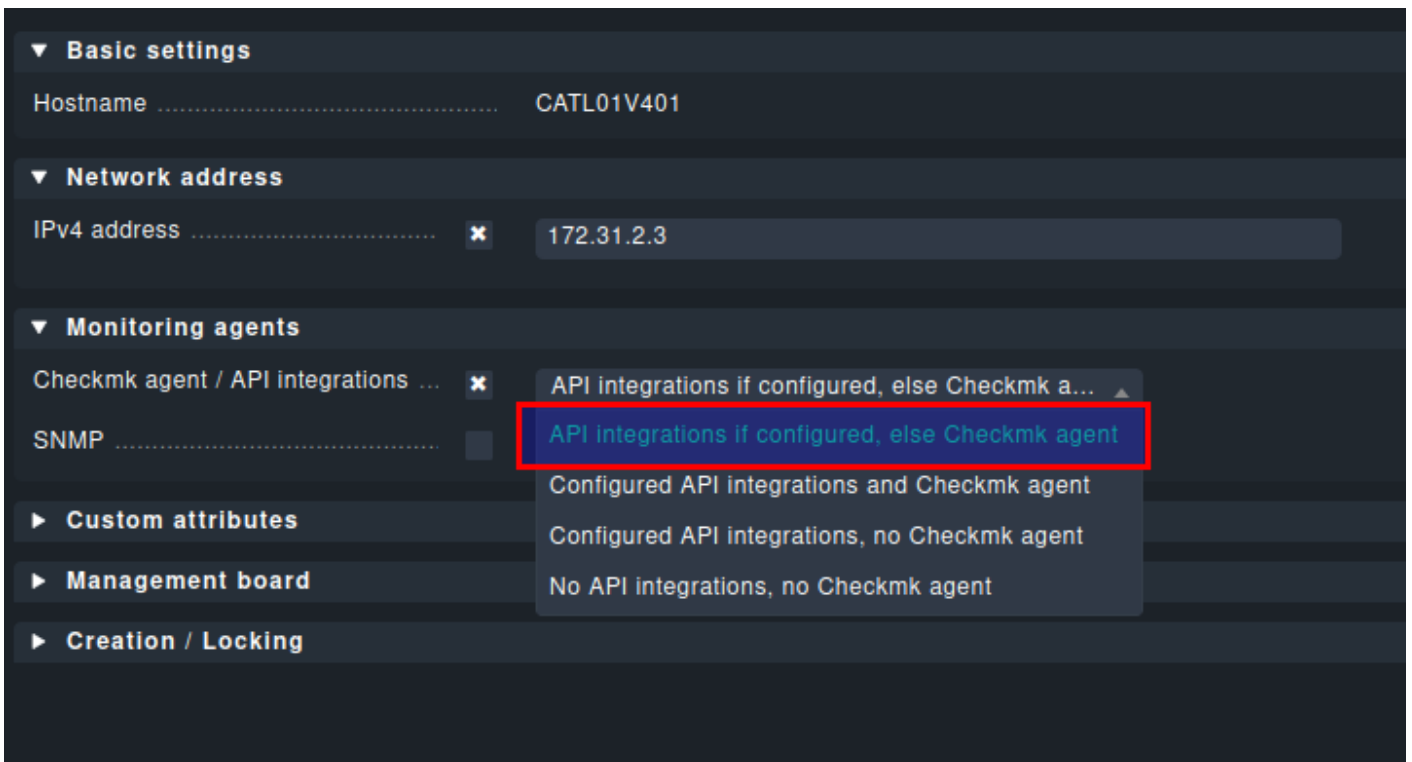
Dazu wieder auf Setup -> Hosts



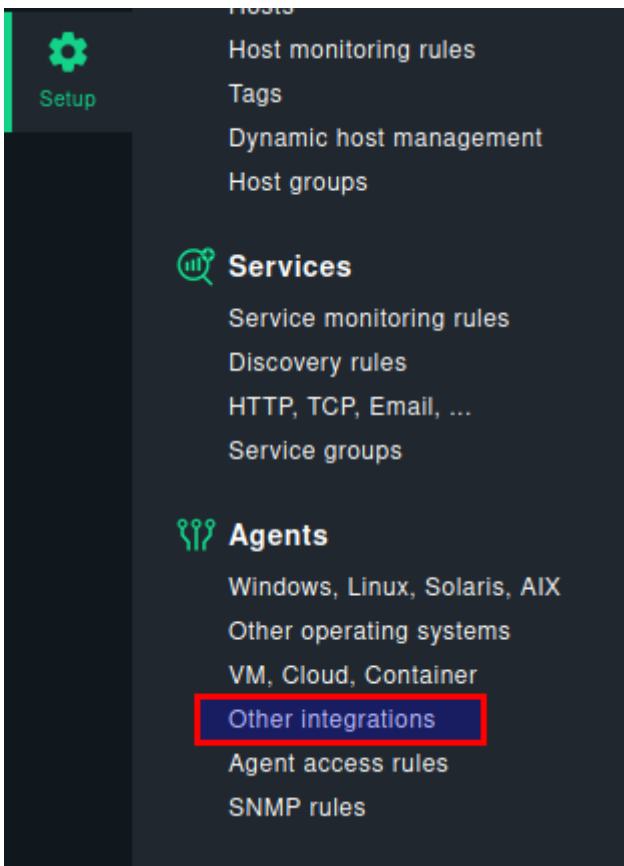
Nun auf edit Host für den Host den man editieren will gehen (Stift)



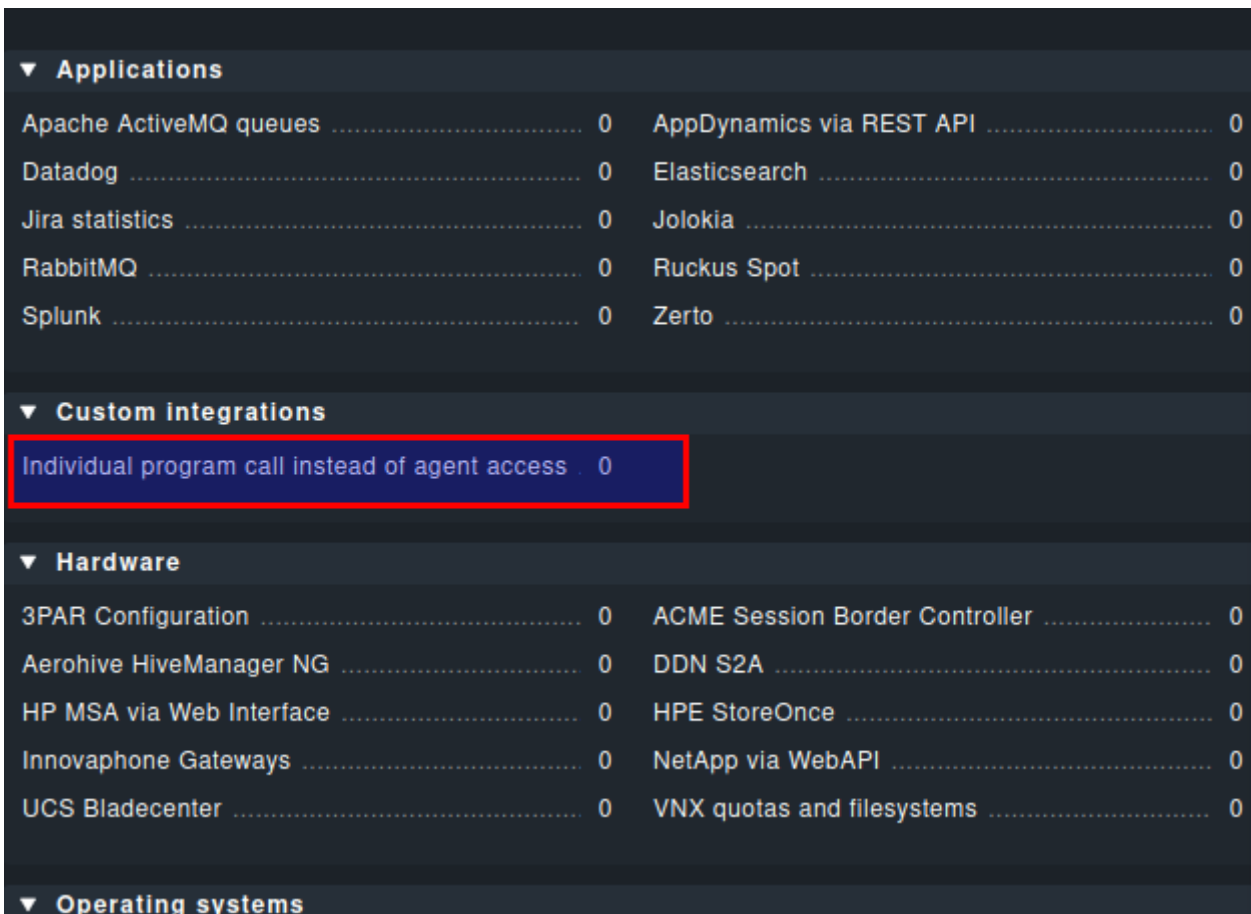
Nun unter Monitoring Agents / API intergration: API integrations if configured, else Checkmk agent auswählen



Wenn die Einstellungen beim Host getätigt sind
Geht es weiter auf : Setup -> Agents -> Other integrations




-> Custom integrations -> Individual program call instead of agent access klicken





Nun auf Add rule klicken

Individual program call instead of agent access

Setup > Agents > Other integrations > Individual program call instead of agent access

Related Rules Display Help 

 Add rule

 Other integrations

Matching: The first matching rule defines the parameter.

There are no rules defined in this set.

Hier nun folgende Sachen ausfüllen:

Description : Agent connection through SSH

Command to Execute : `ssh -T root@$HOSTADDRESS$`

Explicit Hosts: unse(r)e Server die, die Verbindung über SSH aufbauen sollen.

Hinweis: Wir nutzen hier als Variable die `$HOSTADDRESS$` dies bedeutet, dazu zum verbinden die IP vom Host genommen wird.

nun auf save klicken.

Rule Related Display Help

Save Abort Individual program call instead of agent access


For agent based checks Check_MK allows you to specify an alternative program that should be called by Check_MK instead of connecting the agent via TCP. That program is for example useful for monitoring via SSH. You can use monitoring macros here. The most important are:

- \$HOSTADDRESS\$: The IP address of the host
- \$HOSTNAME\$: The name of the host
- \$_HOSTTAGS\$: List of host tags
- \$_HOSTADDRESS_4\$: The IPv4 address of the host
- \$_HOSTADDRESS_6\$: The IPv6 address of the host
- \$_HOSTADDRESS_FAMILY\$: The primary address family of the host

All custom attributes defined for the host are available as \$_HOST[VARIABLE]\$. Replace [VARIABLE] with the *upper case* name of your variable. For example, a host attribute can only be used with the permission "Can add or modify executables".
 HINT: The individual program is called from the current working directory. You should therefore specify absolute path names in scripts (by using environment variables like \$PWD in the console).

Rule Properties

Description Agent connection through SSH

Comment 

Documentation URL

Rule activation do not apply this rule

Individual program call instead of agent access

Command line to execute ssh -T root@\$HOSTADDRESS\$

Conditions

Condition type Explicit conditions

Folder Main

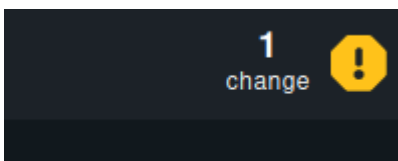
Host tags Add tag condition

Host labels Add label condition

Explicit hosts (Select hostname)

Negate: make rule apply for all but the above hosts

Nun wieder oben rechts auf change klicken





Nun auf activate on selected sites

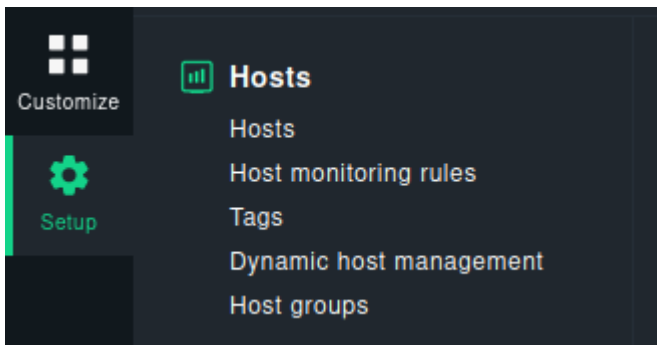
Changes Related Display Help

Activate on selected sites

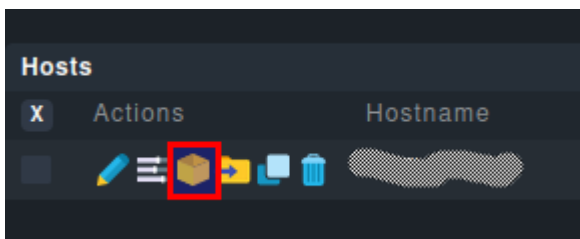
Activation status

Actions	Site	Status
 	Local site catmonitoring	online

Nun wieder auf Setup -> Hosts



Nun auf edit Services für den Host gehen (gelber Würfel)



Nun wenn gewünscht über das X Services deaktivieren, die NICHT überwacht werden sollen. Soll alles übernommen werden einfach auf Accept all

Accept all Rescan Monitor undecided services Remove vanished services Properties of host CATL01V402

Undecided services: 26 | Vanished services: 0 | New host labels: 2 | Vanished host labels: 0 | Changed host labels: 0

No fresh discovery information available. Using latest cached information. Please perform a rescan in case you want to discover the current state.

▼ Discovered host labels (2)

Status	Host labels
New	cmk/device_type:vm cmk/os_family:linux

▼ Undecided services - currently not monitored (26)

State	Service	Status detail
+ x OK	Check_MK Agent	Version: 2.1.0p20, OS: linux, Transport via SSH, Agent plugins: 0, Local checks:
+ x OK	CPU load	15 min load: 0.00, 15 min load per core: 0.00 (2 cores)
+ x OK	CPU utilization	Total CPU: 2.09%
+ x OK	Disk IO SUMMARY	Initializing counters
+ x WARN	Filesystem /	89.16% used (3.10 of 3.48 GB, warn/crit at 80.00%/90.00%)
+ x OK	Filesystem /var	43.32% used (17.37 of 40.09 GB)
+ x OK	Filesystem /var/efw	19.72% used (23.66 of 119.96 MB)
+ x OK	Filesystem /var/log	29.41% used (7.85 of 26.69 GB)
+ x OK	Interface 02	[br0], (up), MAC: 00:50:56:9E:09:57, Speed: unknown
+ x OK	Interface 05	[eth0], (up), MAC: 00:50:56:9E:54:66, Speed: 10 GBit/s
+ x OK	Interface 06	[eth1], (up), MAC: 00:50:56:9E:09:57, Speed: 10 GBit/s
+ x OK	Interface 07	[tap0], (up), MAC: BE:CA:FA:A1:8C:54, Speed: 10 MBit/s

Fertig

Version #11

Erstellt: 20 November 2022 13:08:39 von Admin

Zuletzt aktualisiert: 22 Februar 2023 10:29:07 von Admin