

# Monitoring einrichten

Das Monitoring selbst wird in Monit durch die [Mainkonfigurationsdatei](#) `/etc/monit/monitrc/` (Bereich Monitoring) eingerichtet.

Im Ersten teil werde ich Ihnen einen übersiecht der Möglichkeiten geben. Im zweiten teil gebe ich Ihnen ein paar konkrete Beispiele.

## Nach Änderungen der Konfigurationsdatei muss folgender befehl ausgeführt werden

```
monit -t
```

## Übersucht

### Process

```
CHECK PROCESS <unique name> <PIDFILE <path> | MATCHING <regex>>
```

<path> is the absolute path to the program's pid-file. A pid-file is a file, containing a Process's unique ID. If the pid-file does not exist or does not contain the PID number of a running process, Monit will call the entry's start method if defined.

### File

```
CHECK FILE <unique name> PATH <path>
```

<path> is the absolute path to the file. If the file does not exist, Monit will call the entry's start method if defined, if <path> does not point to a regular file type (for instance a directory), Monit will disable monitoring of this entry. If Monit runs in passive mode or the start method is not defined, Monit will just send an alert on error.

### Fifo

```
CHECK FIFO <unique name> PATH <path>
```

<path> is the absolute path to the fifo. If the fifo does not exist, Monit will call the entry's start method if defined, if <path> does not point to a fifo type (for instance a directory), Monit will disable

monitoring of this entry. If Monit runs in passive mode or the start method is not defined, Monit will just send an alert on error.

## Filesystem

```
CHECK FILESYSTEM <unique name> PATH <string>
```

<path> is the path to the device/disk, mount point or NFS/CIFS/FUSE connection string. If the filesystem becomes unavailable, Monit will call the service's start method if defined. If Monit runs in passive mode or the start method is not defined, Monit will just send an alert on error.

## Directory

```
CHECK DIRECTORY <unique name> PATH <path>
```

<path> is the absolute path to the directory. If the directory does not exist, Monit will call the entry's start method if defined. If <path> does not point to a directory, monit will disable monitoring of this entry. If Monit runs in passive mode or the start methods is not defined, Monit will just send an alert on error.

## Remote host

```
CHECK HOST <unique name> ADDRESS <host>
```

The host address can be specified as a hostname string or as an IP-address string on a dotted decimal format. Such as, „tildeslash.com“ or „64.87.72.95“.

## System

```
CHECK SYSTEM <unique name>
```

The unique name is usually the local host name, but any descriptive name can be used. If you use the variable \$HOST as the name, it will expand to the hostname. This check allows one to monitor general system resources such as CPU usage, total memory usage or load average. The unique name is used as the system hostname in mail alerts and as the initial name of the host entry in M/Monit.

## Program

```
CHECK PROGRAM <unique name> PATH <executable file> [TIMEOUT <number>
```

## SECONDS ]

<path> is the absolute path to the executable program or script. The status test allows one to check the program's exit status. If the program does not finish executing within <number> seconds, Monit will terminate it. The default program timeout is 300 seconds (5 minutes). The output of the program is recorded and made available in the User Interface and in alerts, by default up to 512 bytes. You can change the output limit using the set limits statement).

## Network

```
CHECK NETWORK <unique name> <ADDRESS <ipaddress> | INTERFACE <name>>
```

<ipaddress> is the IPv4 or IPv6 address of the monitored network interface. It is also possible to use interface name, such as „eth0“ on Linux.

## Beispiele

### Host Überwachung

```
check system $HOST
  if loadavg (1min) > 4 then alert
  if loadavg (5min) > 2 then alert
  if memory usage > 75% then alert
  if swap usage > 25% then alert
  if cpu usage (user) > 70% then alert
  if cpu usage (system) > 30% then alert
  if cpu usage (wait) > 20% then alert
```

### SSH Überwachung

```
check process SSH with pidfile /var/run/sshd.pid
  start program "/bin/bash -c '/etc/init.d/ssh start'"
  stop program "/bin/bash -c '/etc/init.d/ssh stop'"
  if failed port 22 protocol ssh then restart
  if 5 restarts within 5 cycles then timeout
```

## Apache2 Überwachung

```
check process Apache2 with pidfile /var/run/apache2/apache2.pid
group www
start program = "/bin/bash -c '/etc/init.d/apache2 start'"
stop program = "/bin/bash -c '/etc/init.d/apache2 stop'"
if failed host www.netgraphtech.de port 80 protocol http then restart
if 5 restarts within 5 cycles then timeout
```

## MySQL Überwachung

```
check process MySQL with pidfile /var/run/mysqld/mysqld.pid
group database
start program = "/bin/bash -c '/etc/init.d/mysql start'"
stop program = "/bin/bash -c '/etc/init.d/mysql stop'"
if failed host 127.0.0.1 port 3306 then restart
if 5 restarts within 5 cycles then timeout
```

## Fail2ban Überwachung

```
check process Fail2ban with pidfile /var/run/fail2ban/fail2ban.pid
start program = "/bin/bash -c '/etc/init.d/fail2ban start'"
stop program = "/bin/bash -c '/etc/init.d/fail2ban stop'"
```

## Teamspeak3 Überwachung

```
check process Teamspeak3 with pidfile /usr/local/teamspeak3/teamspeak3-
server_linux_amd64/ts3server.pid
start program "/bin/bash -c '/etc/init.d/teamspeak3 start'"
stop program "/bin/bash -c '/etc/init.d/teamspeak3 stop'"
if failed host netgraphtech.de port 10011 then restart
if 5 restarts within 5 cycles then timeout
```

## Festplatten Überwachung

```
check device root with path /dev/sda3
```

From:

<https://www.wiki.netgraphtech.de/> - **NetGraphTech**

Permanent link:

[https://www.wiki.netgraphtech.de/doku.php?id=wiki:monit\\_config](https://www.wiki.netgraphtech.de/doku.php?id=wiki:monit_config)

Last update: **2024/05/28 23:28**

