

Einrichtung Community-Plugins

Level „easy“



Was sind Community-Plugins?

- Nagios-Plugins
- PHP, Perl, Bash ...
- *`/var/lib/nagios/libexec[-customer]/`*

ITSM-SUMMIT 2019

```
10:30:47[root@snvm-esxpfolkers] libexec $> ls
check_apt          check_ifstatus    check_nt           check_scheduler_enqueued.py  check_wave
check_breeze       check_imap        check_ntp         check_sensors             check_wmi.pl
check_by_ssh       check_ircd        check_ntp_peer   check_simap                negate
check_clamd        check_jabber      check_ntp_time   check_smtp                 noCheckCommand.sh
check_cluster      check_load        check_rwstat     check_snmp                 process_perfddata.pl
check_dhcp         check_localhealth.pl  check_oracle     check_spop                 rrd_convert.pl
check_dig          check_log         check_overcr     check_ssh                  rrd_modify.pl
check_disk         check_log_ci.sh  check_ping       check_ssmtp                sv_check_dyn.php
check_disk_smb     check_log.sh     check_pnp_rnds.pl  check_sv_bpmn.pl          tmp
check_dns          check_mailq      check_pop        check_sv_hostgroup.pl     utilize
check_dns.centos  check_mem.pl     check_procs.centos  check_sv_jobs.pl         utils.pm
check_dummy        check_nrtg       check_procs      check_svlocal.php         utils.sh
check_file_age     check_nrtgtraf  check_random      check_sv_servicegroup.pl  vault_wrapper
check_flexlm       check_mysql      check_real       check_swap                 vault_wrapperctl
check_ftp          check_nagios     check_resque_running.py  check_tcp                 verify_pnp_config.pl
check_hpjd         check_nagios.centos  check_resque_running.pyc  check_time                wmic
check_http         check_nginx.sh   check_resque_running.py  check_udp                 check_ups
check_icmp         check_nntp       check_rpc         check_scheduler_enqueued.py  check_uptime
check_ide_smart    check_nntps     check_scheduler_enqueued.py  check_users
check_ifoperstatus  check_nrpe      check_scheduler_enqueued.pyc
10:30:59[root@snvm-esxpfolkers] libexec $>
```

Wie teste ich die Plugins?

- Bash als Nagios-User
- Plugins ausführen
- Abhängigkeiten und Funktion prüfen

```
10:37:43[root@snm-esxfolkers] libexec $> su nagios
10:37:45[nagios@snm-esxfolkers] libexec $> ./check_disk
check_disk: Could not parse arguments
Usage:
  check_disk -w limit -c limit [-W limit] [-K limit] {-p path | -x device}
[-C] [-E] [-e] [-f] [-g group ] [-k] [-l] [-M] [-m] [-R path ] [-r path ]
[-t timeout] [-u unit] [-v] [-X type] [-N type] [-n]
10:37:49[nagios@snm-esxfolkers] libexec $> ./check_disk -w 20% -c 10%
DISK OK - free space: / 46620 MB (91,10% inode=100%); /dev 1883 MB (100,00% inode=100%); /dev/shm 1894 MB (99,99% inode=100%); /run 1709 MB (90,23% inode=100%); /sys/fs/cgroup 1894 MB (100,00% inode=100%); /boot 826 MB (81,46% inode=100%); /home 25675 MB (99,7% inode=100%); /run/user/993 378 MB (100,00% inode=100%); /run/user/2 378 MB (100,00% inode=100%); /run/user/0 378 MB (100,00% inode=100%); | /=4554MB;40940;46057;0;51175 /dev=0MB;1506;1694;0;1883 /dev/shm=0MB;1515;1704;0;1894 /run=184MB;1515;1704;0;1894 /sys/fs/cgroup=0MB;1515;1704;0;1894 /boot=187MB;811;912;0;1014 /home=32MB;20565;23136;0;25707 /run/user/993=0MB;302;340;0;378 /run/user/2=0MB;302;340;0;378 /run/user/0=0MB;302;340;0;378
10:37:58[nagios@snm-esxfolkers] libexec $> |
```

Wie nutze ich das in SNAG-View?

- Neues Checkcommand (Objektadmin)
- Pfad zum Plugin („\$USER1\$“ / „\$USER2\$“)
- Parameter definieren
- Abspeichern
- Service erstellen

Servicebrowser x Objektverwaltung x

Objektauswahl

Checkcommand ▼

Suchen + ?

Id	Name
000Y	ALIVE
000z	check_localhealth
000D	check_nginx
000B	check_resque_running
000C	check_scheduler_enqueued
0014	check_sv_bpmon
000A	check_sv_jobs
000Z	DUMMY
0005	dyn_wrapper_repoll
0017	LX_NRPE_CPU
0018	LX_NRPE_DISK
0019	LX_NRPE_SWAP
001a	LX_NRPE_USERS
0010	SSH
0011	SWAP
0012	TELNET
001A	WIN_NRPE_CLIENTVERSION
001w	WIN_NRPE_CPU
001x	WIN_NRPE_DISK
001V	WIN_NRPE_EVENTLOG
001z	WIN_NRPE_MEM
001y	WIN_NRPE_SRVC
001X	WIN_NRPE_UPTIME

Checkcommand erstellen

Allgemein Sonstiges

Bezeichnung Pflichtfeld ⚙️

Befehlszeile Pflichtfeld ⚙️ ✂️ +

Passwort Tresor nutzen Nein ⚙️

Parameter + ⚙️

Parameter
Warning Percent
Critical Percent

Kategorie ⚙️ ✂️ +

Perfomancedaten sammeln Ja ⚙️



ITSM-SUMMIT 2019

Service erstellen

Allgemein | Monitoring | Erweiterte Einstellungen | Sonstiges

Bezeichnung
Pflichtfeld ⓘ

Templates
 + ⓘ

Checkcommand
 ⓘ | ✂

Parameter
Warning Percent: ⓘ | ✂
Critical Percent: ⓘ



ITSM-SUMMIT 2019

Servicebrowser | Objektverwaltung | Serviceinstanz: SNAG-View - SP_Festplatten-Status

12 | [Icons]

SNAG-View - SP_Festplatten-Status

DISK OK - free space: / 46620 MB (91,09% inode=100%); /dev 1883 MB (100,00% inode=10...

Informationen | Downtimes | Kommentare | Business Process | Verfügbarkeit | Wiki | Performance Daten | Benachrichtigungen | Dynamic Map

SVID: 000E0021
Host: SNAG-View (000E)
Mitglied in:
Serviceprofil: [Serviceprofil](#)
Checkversuche: 1/3 (HARD state)
Check Typ: ACTIVE
Check Latenz / Dauer: 0 / 0 Sekunden
Letzter Check: 05.08.2019 10:49:57
Nächster Check: 05.08.2019 10:54:57
Letzte State Änderung: 05.08.2019 10:49:57 (4s)
Letzte HARD State Änderung: 05.08.2019 10:49:57
Flapping: Nein
Downtime: Nein
Plugin Output: DISK OK - free space: / 46620 MB (91,09% inode=100%); /dev 1883 MB (100,00% inode=100%); /dev/shm 1894 MB (99,99% inode=100%); /run 1709 MB (90,23% inode=100%); /sys/fs/cgroup 1894 MB (100,00% inode=100%); /boot 826 MB (81,46% inode=100%); /home 25675 MB (99,87% inode=100%); /run/user/993 378 MB (100,00% inode=100%); /run/user/2 378 MB (100,00% inode=100%); /run/user/0 378 MB (100,00% inode=100%);
Perfomancedaten: /=4554MB;40940;46057;0;51175 /dev=0MB;1506;1694;0;1883 /dev/shm=0MB;1515;1704;0;1894 /run=184MB;1515;1704;0;1894 /sys/fs/cgroup=0MB;1515;1704;0;1894 /boot=187MB;811;912;0;1014 /home=32MB;20565;23136;0;25707 /run/user/993=0MB;302;340;0;378 /run/user/2=0MB;302;340;0;378 /run/user/0=0MB;302;340;0;378
Long Plugin Output:



Live-Demo

