NetApp-Plugins für Nagios

Ingo Lantschner

2009-10-19

Die Administrator-freundlichen Plugins zur Überwachung von NetApp

1 Alleinstellungsmerkmale

- Strukturierte, einheitliche Datenabfrage und Authentifizierung (HTTP/XML)
- Einheitliches Interface und Ausgabe
- Implementierungs- und wartungsfreundlich ⇒ stabile Konfiguration im Nagios
- Umfangreiches, laufend erweitertes Angebot an überwachten Werten (Auslastung, Lag-Time, SnapVaults, SnapMirrors, Cluster, Hardware, ...)
- Rasche, unkomplizierte Reaktion auf Kundenwünsche

2 Hintergrund

Auf Grund mehrfacher Anfragen nach Plugins zur Überwachung von NetApp-Filern, wurde von mir eine komplette Suite entwickelt. In Zusammenarbeit mit meinen Kunden entwickle ich Plugins, die sich im praktischen Einsatz rasch unkompliziert und stabil implementieren lassen.

- Alle Plugins authentifizieren sich mittels Benutzername und Passwort; entweder mit dem Schalter `-u <username\%pass>` oder einer Textdatei, die mittel `-f` spezifiziert wird.
- Alle Plugins fragen ihre Daten via HTTP in XML-Form von der NetApp-API ab, Freischaltungen an der Firewall für SNMP, SSH oder andere Protokolle sind nicht mehr nötig. Auf Wunsch kann auch mit SSL verschlüsselt werden.
- Mehrfache Instanzen (Aggregate, SnapMirrors, ...) werden vom Plugin automatisch erkannt und überwacht, ohne dass deren Namen ausdrücklich konfiguriert werden müssen. Das Auflisten, Nachtragen oder Löschen von z.B. Aggregaten für die Auslastungsüberwachung ist nicht mehr nötig. Das reduziert den Aufwand bei der Implementierung und erhöht die Stabilität des Monitorings auch langfristig.

3 Verzeichnis aller angebotenen Plugins

Das folgende Verzeichnis wird direkt aus dem Quellcode der Plugins erzeugt. Die Texte sind daher in Englisch. Für jedes Plugin wird eine kurze Zusammenfassung der wichtigsten Parameter angegeben (`'Usage'`), dann ein Auszug aus der Hilfe und Beispiele. Was hier angezeigt wird, sehen Sie dann auch bei den Plugins bei Angabe des Schalters `--help`[1]. (Selbstverständlich ist die Hilfe des jeweiligen Plugins noch weit ausführlicher, da jeder Schalter nochmals einzeln erklärt wird.)

3.1 check_netapp_ops

Checks operations/second and transferrate.

Usage

```
check_netapp_ops -H <hostname|IP> -u <username%pass> -z
http_ops|cifs_ops|net_data_sent|... [-w <n> -c <n> ] [-help] [...]",
```

Help

This plugin checks various performance counters (Mostly operations/second and transfer-rates) of the NetApp-system. See –counter|-z for possible counters.

Not all counters are supported on every system. The plugin will tell you, if you have chosen a non-existing counter and exit with UNKNOWN.

WARNING: Some of the counters have privilege-level DIAG. These are counters used for diagnosis purposes. Some of these counters can report incorrect data too!

HINT: print_netapp_sysinfo and check_netapp_status list all available counters together with its description and privilege-level of a given system.

Performance-data will be returned for every counter.

Examples

$ check_netapp_ops -H netapp01 -u nagios%pass -z http_ops
→ Check the http-operations per second on netapp01 using the default-thresholds.

$ check_netapp_ops -H netapp01 -u nagios%pass -z nfs_ops
→ Check the nfs-operations per second on netapp01 using the default-thresholds.

$ check_netapp_ops -H netapp01 -u nagios%pass -z net_data_sent -w 10000 -c 50000
→ Check the transfer-rate (kbits/second send to the network) of netapp01. Warn if 10 Mbit/s are exceeded ...

[1] Im Sinne übersichtlichen und effizient wartebaren Programmcodes, dehne wir so das DRY-Prinzip (Don't Repeat Yourself) auch auf die Dokumentation und Vertriebsunterlagen aus. Ich denke es kommt letztendlich auch der von mir geschriebenen Software zu gute, wenn die Hilfetexte in einer Form verfasst werden, die auch für den Vertrieb tauglich ist. Jedenfalls bemühe ich mich seit dem um so mehr, diese leicht verständlich und auf den Punkt gebracht zu formulieren.

$ check_netapp_ops -H netapp01 -u nagios%pass -z streaming_pkts
→ Check the number of streamed packets on netapp01 using the default-thresholds.

3.2 check_netapp_snapreserve

Checks the snapreserve on all or one specified volume(s) of a NetApp filers.

Usage

```
check_netapp_snapreserve -H <hostname|IP> -u <username%password> [-n
<volume name>] [ [-w|c x:y%] | [-w|c x:yGB] ] [-help] [...]",
```

Help

This plugin checks for the snapreserves size on all volumes (on exactly one volume if –volume|-n is set).

Thresholds can be set in GB or % - the unit must be appended and the same for both thresholds.

Performance-data is printed in % and GB.

Examples

$ check_netapp_snapreserve -H netapp01 -u nagios%pass -n vo1
→ Checks vol1 with default-values.

$ check_netapp_snapreserve -H netapp01 -u nagios%pass -n vol1 -w 25:% -c 15:%
→ Checks vol1; warns if snapreserve is smaller than 25% and critical if smaller than 15% of the volume-size.

$ check_netapp_snapreserve -H netapp01 -u nagios%pass -w 10:20% -c :40
→ Checks all volumes on netapp01; warns if at least one volumes snapreserve is either smaller than 10% or larger than 20%; critical if larger than 40%.

3.3 check_netapp_snapshots

Checks the space occupied by snapshops on volumes.

Usage

```
check_netapp_snapshots -H <filer> -f auth [ -n <volname> -b vol|reserve
-w <n>% -c <n>% ... ] ",
```

Help

This plugin checks the size used by snapshots. Depending on the –base the total cumulated size of all snapshots is either compared to the size of the snap-reserve or the size of the volume.

Alerts if the snapshots occupy more than a certain part of the snap-reserve or total size of the volume. The thresholds are in %.

Skipped Volumes: If the option –instance|n is missing, all volumes are checked.

Volumes will be skipped, if at least one of the following is true:

- The size is smaller or equal zero (volume restricted or offline)

- 'snapshot-blocks-reserved' is not defined

- Volume is excluded by means of –exclude|X

Use -v to see, which volumes are skipped.

Performance-data is printed in % and blocks (1k) - depending on the various –perf_... switches.

Timeout defaults to 30. Due to the complexity of this plugin, a lot of requests are sent to the filer (depending on the number of snapshots and volumes). Therefore it may be advisable to have a timeout even › 30 seconds.

Examples

$ check_netapp_snapshots -H filer -f auth
→ Check all volumes and warn if the defaults for warn and crit are exceeded.

$ check_netapp_snapshots -H filer -f auth –base=vol
→ Same as above (–base defaults to vol), snapsize is calculated relative to the volumesize

$ check_netapp_snapshots -H filer -f auth –base=relative
→ Same as above, but the snapsize is calculated relative to the snapreserve (and not the volumesize) Results always in a critical exit, if at least one volume has no snapreserve. (see examples below and –check_only for a solution)

$ check_netapp_snapshots -H filer -f auth -n vol0 -w 65% -c 80%
→ Check vol0 and warn if more than 65% of the volumes size is used for snapshots, critical if more than 80%

$ check_netapp_snapshots -H filer -f auth –base=reserve –check_only=with_reserve
→ Check all volumes, which do have a snap-reserve. Calculate the relative usage by snapshots based on the size of the snap-reserve.

$ check_netapp_snapshots -H filer -f auth –base=vol –check_only=no_reserve
→ Check all volumes which do not have a snap-reserve. Calculate the relative usage by snapshots based on the size of the volume.

3.4 check_netapp_usage

Checks the space-usage of aggregates or volumes.

Usage

```
check_netapp_usage -H <hostname|IP> -u <username%password> -o
<vol|aggr> [-n <instance name> | -X <instance name>] [-w %|GB] [-c
%|GB] [-perf_uom=GB|%] [-help] [...]",
```

Help

This plugin checks the used space on either aggregates or volumes (depending on –object|-o aggr|vol).

Specific volumes or aggregates can be excluded by using –exclude|-X. With the optional –instance|-n only the named volume or aggregate is checked.

Thresholds can be set in % or GB. The type of Perfdata (relative or absolute) can be defined with the optional switch –perf_uom=%|GB

Performance-data will be printed for all checked volumes/aggregates.

Examples

$ check_netapp_usage -H netapp01 -u nagios%pass -o vol
→ Checks all volumes on netapp01 using the default thresholds.

$ check_netapp_usage -H netapp01 -u nagios%pass -o aggr
→ Checks all aggregates on netapp01 using the default thresholds.

$ check_netapp_usage -H netapp01 -u nagios%pass -o aggr -w 50% -c 98%
→ Checks all aggregates on netapp01. Returns a warning, if at least one volume is more than 50% used and critical-alarm if more that 98% used.

$ check_netapp_usage -H netapp01 -u nagios%pass -o vol -X vol4
→ Checks all volumes on netapp01, except vol4. Uses the default thresholds.

$ check_netapp_usage -H netapp01 -u nagios%pass -o vol -X vol3 -X vol4
→ Checks all volumes on netapp01, except vol3 and vol4. Uses the default thresholds.

$ check_netapp_usage -H netapp01 -u nagios%pass -o vol -n vol2 -w 50% -c 80%
→ Returns a warning, if vol2 is more than 50% used.

$ check_netapp_usage -H netapp01 -u nagios%pass -o vol -n vol2 -w 700GB -c 850GB
→ Returns a warning, if more than 700GB used on vol2. Perfdata is still in %!

$ check_netapp_usage -H netapp01 -u nagios%pass -o vol -n vol2 -w 700GB -c 850GB –perf_uom=GB
→ Same as above, but now perfdata is in GB.

3.5 check_netapp_status

Checks the global Status of NetApp filers.

Usage

```
check_netapp_status -H <hostname|IP> -u <username%pass> | -f <authfile>
 [-help] [...]",
```

Help

This plugin checks the global Status (A string describing the global status, including a description of the condition (if any) that caused the status to be anything other than ok).

Examples

$ check_netapp_status -H netapp01 -u nagios%pass
→ Check the global Status of netapp01.

3.6 check_netapp_snapmirror

Checks the state, lag-time and transfer-duration of SnapMirros.

Usage

```
check_netapp_snapmirror -H <hostname|IP> -u <username%password> [-e
lag_time|last_transfer_duration|... ] [-l <location>] [-help] [...]",
```

Help

This plugin checks for many parameters of SnapMirrors.

* pair-state, see -v for defaults.

* pair-transfer-status, see -v for defaults.

* lag-time

* last-transfer-duration

* Transfer errors result in a warning-exit.

Thresholds are set for lag-time and last-transfer-time in seconds.

Performance-data is printed for lag-time and transfer-time.

You can use this plugin for either overall-checks (all SnapMirrors on the filer) or define a specific volume or qtree. If used for overall-checking specific SnapMirrors can be excluded on the command-line. See the examples for details.

Examples

$ check_netapp_snapmirror -H netapp01 -u nagios%pass
→ Checks the lag-time of all snapmirror-sources on netapp01 with the default-thresholds.

$ check_netapp_snapmirror -H netapp01 -u nagios%pass -X netapp01:vol4
→ Checks the lag-time of all snapmirror-sources on netapp01 with the exception of vol4 using the default-thresholds.

$ check_netapp_snapmirror -H netapp01 -u nagios%pass -X netapp01:vol2 -X netapp01:vol4 -w 120 -c 360
→ Checks the lag-time of all snapmirror-sources on netapp01 with the exception of vol2 and vol4. Thresholds are set to 120 and 360 seconds.

$ check_netapp_snapmirror -H netapp01 -u nagios%pass -l vo1 -e pair_state –ok_pairstate=snapmirrored –ok_pairstate=uninitialized –w_pairstate=source
→ Checks vol1, ok if state is either snapmirrored or uninitialized, warning if source.

$ check_netapp_snapmirror -H netapp01 -u nagios%pass -e last_transfer_duration -w 360 -c 1800
→ Checks the last-transfer-duration of all snapmirror-sources on netapp01. Warning if more than 5 minutes (360s) and critical if more than one hour (1800 s).

3.7 check_netapp_quotas

This plugin checks the usage of quotas.

Usage

```
check_netapp_quotas -H <hostname|IP> -u <username%pass> [-path=<path>
| -volume=<name>] [-help] [...]",
```

Help

This plugin monitors the usage of quotas on a NetApp-Filer. All quotas of volumes an qtrees will be monitored based on thresholds of the internal NetApp-quota-system. Performancedate will be returned in any case - even if not thresholds are set.

The monitored objects can be limited by using the optional switches –path and –volume.

Performance data: All Disk-values in KByte - the thresholds are taken directly from the NetAp-Quota-System. Soft-Limit=Warning / Limit=Critical",

Examples

$ check_netapp_quotas -H netapp01 -u nagios%pass
→ Check all qotas on netapp01.

$ check_netapp_quotas -H netapp01 -u nagios%pass –path=/vol/vol0/users
→ Check quotas in /vol/vol0/users.

3.8 check_netapp_iSCSI-adapter

Checks the state (online, offline, ...) of iSCSI-adapters.

Usage

```
check_netapp_iSCSI-adapter -H <hostname|IP> -u <username%password> [-n
<iSCSI-Adapter name>] [-help] [...]",
```

Help

This plugin checks the state of one or all iSCSI-Adapters on a filer.

Reasonable defaults are set, for which state is considered OK, warning or critical.

These defaults can be changed using command-line parameters.

Verbose mode is helpful, to check which adapter-state corresponds to which nagios-exit.

Examples

$ check_netapp_iSCSI-adapter -H netapp01 -u nagios%pass
→ Checks all iSCSI-adapters on netapp01 using default-values.

$ check_netapp_iSCSI-adapter -H netapp01 -u nagios%pass –ok_state=offline –ok_state=online –empty_c_state_default
→ Checks all iSCSI-adapters on netapp01, return OK is adapter is either online or offline.

$ check_netapp_iSCSI-adapter -H netapp01 -u nagios%pass -v
→ Have a look, which adapter-states considered OK, ... by default.

3.9 check_netapp_cluster

Checks the cluster service.

Usage

```
check_netapp_cluster -H <hostname|IP> -u <username%pass> [-e
state|time_master] [-help] [...]",
```

Help

This plugin checks for various parameters of clustered NetApp-filers.

* state, see -v for defaults.

* time-master-status

Which state or status are considered OK, warning or critical is defined by defaults, which can be changed by command-line parameters.

Examples

$ check_netapp_cluster -H node1 -u nagios%pass
→ Checks the cluster-state on node1 using the defaults.

$ check_netapp_cluster -H node1 -u nagios%pass -e time_master
→ Checks the time-master-status on node1 using the defaults.

$ check_netapp_cluster -H node1 -u nagios%pass -e time_master –ok_time_master_status=slave –w_time_master_status=master
→ Checks the time-master-status on node1. Exit OK if it is slave and warn if status is master.

3.10 check_netapp_raidstatus

Checks the raid-status of volumes or aggregates.

Usage

```
check_netapp_raidstatus -H <hostname|IP> -u <username%pass> -o vol|aggr
[-n <name>] [-help] [...]",
```

Help

This plugin checks the raid-status of volumes or aggregates on NetApp filers. Reasonable defaults are set by the script (degraded -› CRITICAL, ...).

To change the defaults, you can edit the corresponding arrays in the script.

Examples

$ check_netapp_raidstatus -H netapp01 -u nagios%pass -o aggr
→ Check the raidstatus of all aggregates on netapp01

$ check_netapp_raidstatus -H netapp01 -u nagios%pass -o vol -n vol0
→ Check the raidstatus of the volume "vol0ön netapp01

$ check_netapp_raidstatus -H netapp01 -u nagios%pass -o aggr -n aggr1
→ Check the raidstatus of the aggregate äggr1ön netapp01

3.11 check_netapp_snapvault

Checks the SnapVaults-state, duration of last transfer and lag-time on NetApp secondary filers.

Usage

```
check_netapp_snapvault -H <hostname|IP> -u <username%password> -p
<path> [-w_lag=<n> -c_lag=<n> -w_dur=<n> ... ] [-help] [...]",
```

Help

This plugin checks many parameters of a SnapVault relationship by contacting the secondary system.

* State, see -v for defaults.

* lag-time

* last-transfer-duration

* Transfer errors result at least in a warning-exit (even if you consider the state OK by command-line arguments).

You can change this and other default-behavior by several command-line-arguments like –ignore_transfer_err.

See above.

Thresholds are set for lag-time and last-transfer-time in seconds.

Performance-data is printed for lag-time and transfer-time.

Examples

$ check_netapp_snapvault -H netapp01 -u nagios%pass -p /vol/vol0/qtree1
→ Checks qtree1 with default-values. netapp01 must be the secondary filer.

$ check_netapp_snapvault -H netapp01 -u nagios%pass -p /vol/vol0/qtree1 –w_lag=500000 –c_lag=900000
→ Same as above, but with defined values for the lag-time. (Warning if the lagtime exceeds 500000 seconds, ...)

3.12 check_netapp_snapsizeavailable

Checks for the size available for taking a snapshot.

Usage

```
check_netapp_snapsizeavailable -H <hostname|IP> -u <username%password>
-n <volume name> -w <n>:GB -c <n>:GB ",
```

Help

This plugin checks for the size available for taking a snapshot.

Examples

./check_netapp_availablesnapsize.pl –alpha -f auth -H simbig -n vol0 -w 5:GB -c 1:GB
→ NETAPP_AVAILABLESNAPSIZE WARNING - Size available for taking snaphots: 1.115 GByte | size-available=1.115GB;5:;1:

$ check_netapp_snapsizeavailable -H netapp01 -u nagios%pass -n vo1
→ Checks vol1 ...

3.13 check_netapp_disk

Checks for broken disks.

Usage

```
check_netapp_disk -H <hostname|IP> -u <username%pass> [-help] [...]",
```

Help

This plugin checks for offline, pre-failed or replacing disks.

Examples

$ check_netapp_disk -H netapp01 -u nagios%pass
→ Check for offline/prefailed/replacing disks on netapp01

3.14 check_netapp_utilization

Checks utilization of CPU and disks.

Usage

```
check_netapp_utilization -H <hostname|IP> -u <username%pass> -o
disk|processor [-w <n> -c <n> ] [-help] [...]",
```

Help

This plugin checks the utilization in % of disks and processor.

All instances of this object are enumerated and the corresponding counters are retrieved two times with a delta (defined with –delta|-d) in between. Based on these values the plugin computes the utilization in percent.

Performance-data will be returned for each instance (f.e. for each disk, or CPU0, CPU1, ...).

Examples

$ check_netapp_utilization -H netapp01 -u nagios%pass -o disk
→ Check the utilization of all disks on netapp01 using the default-thresholds.

$ check_netapp_utilization -H netapp01 -u nagios%pass -o processor -w 80 -c 90
→ Check the utilization of all processors on netapp01 and warn if one of them is over 80%.

3.15 check_netapp_hardware

Checks fans, power-supplies, nvram and temperature-sensors.

Usage

```
check_netapp_hardware -H <hostname|IP> -u <username%pass> -o
<temp|cool|power|nvram> [-help] [...]",
```

Help

This plugin checks environmental information and hardware like cooling-devices, power-supplies or nvram for filer and its shelfs.

No thresholds are set and used since the internal logic of NetApp already defines low and high critical values for temperature or fans.

In case of a non-OK-device or sensor found, at least the shelf-number is returned to Nagios, so that further diagnostics can be accomplished straight forward.

F.e. "NETAPP_HARDWARE CRITICAL - Temperature-sensor(s) out of recommended range, failed or missing!

Shelf 1 temperature sensor 1 (over_temperature_failure)"

The output uses the multiline-capability of Nagios 3.

Performance data is returned for the temperature sensors in degree Celsius.

F.e. "NETAPP_HARDWARE OK - The temperature of all sensors is within the recommended ranges.

| s1e1=24C;; s1e2=24C;; s1e3=24C;; s2e1=24C;; s2e2=24C;; s2e3=24C;;"

The short-label for the perfdata has the format s‹shelfNo›e‹elementNo›

Examples

$ check_netapp_hardware -H netapp01 -u nagios%pass -o nvram
→ Check the nvram-status on netapp01

$ check_netapp_hardware -H netapp01 -u nagios%pass -o power
→ Check all power-supplies on the filer netapp01 and its shelves.

$ check_netapp_hardware -H netapp01 -u nagios%pass -o temp
→ Check the temperature-sensors on the filer netapp01 and all its shelves. Returns the current temperature of each shelf as performance-data.