

1 Intranet Monitor

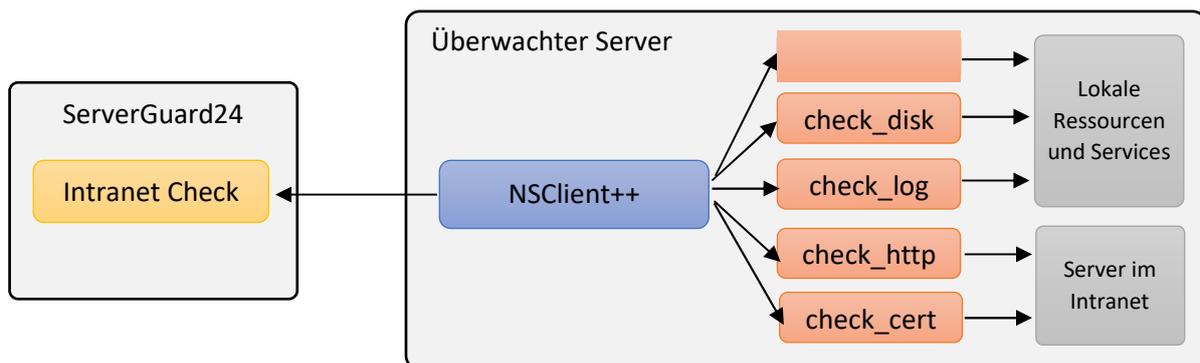
Die Überwachung von Servern im Intranet durch den Intranet Check erfordert die Installation eines Monitoring Agents auf dem überwachten Server.

ServerGuard24 unterstützt den NSClient++. Dieser Agent hat zahlreiche Vorteile

- läuft auf tausenden Systemen und hat sich millionenfach bewährt
- sehr gut dokumentiert
- hat eine aktive Community
- verfügt über eine Vielzahl an Funktionen

Der Agent wird im passiven Modus konfiguriert. Das bedeutet, dass er die Checkergebnisse selbständig über HTTPS an ServerGuard24 sendet. Da kein Zugriff von außen erfolgt, ist **keine** Firewall Freischaltung notwendig.

In der Agent Config werden die Monitoring Commands mit ihren Parametern definiert und mit einem Alias versehen. In ServerGuard24 wird dieser Alias in der Check Konfiguration hinterlegt.



2 NSClient++ Installation

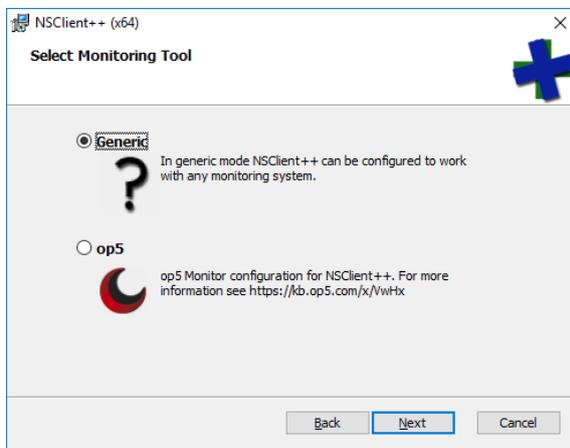
2.1 Einführung

NSClient++ ist für Windows sowie für Linux/Unix verfügbar. Er beinhaltet vordefinierte Check Plugins und bietet die Möglichkeit, eigene Skripts in die Überwachung einzubinden.

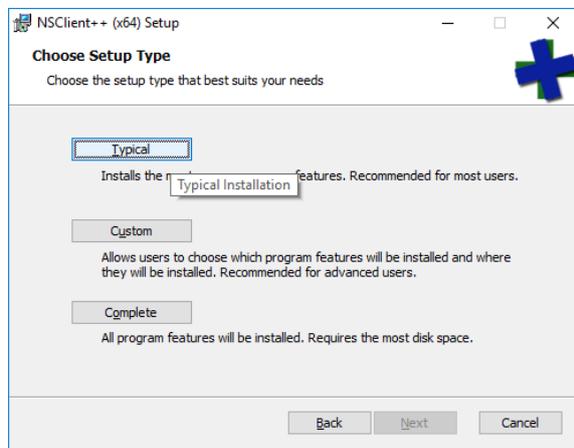
Weiterhin ermöglicht der NSClient++ eine automatische Fehlerbehebung. Dazu führt der Agent definierten Aktionen beim Feststellen bestimmter Systemzustände aus.

Eine ausführliche Dokumentation des NSClients++ erhalten Sie unter <https://www.nscient.org>. Die aktuellste Version kann für Windows und Linux unter <https://www.nscient.org/download> geladen werden.

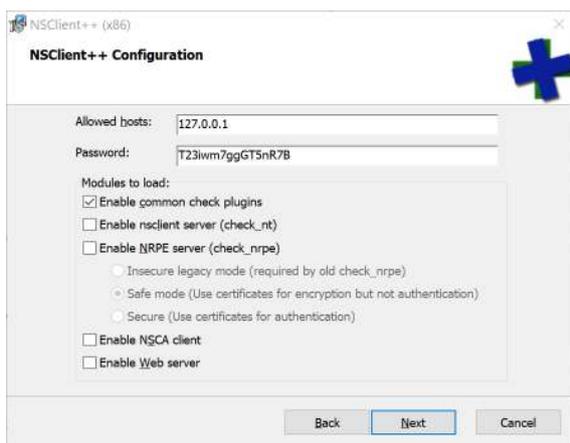
2.2 Windows Installation



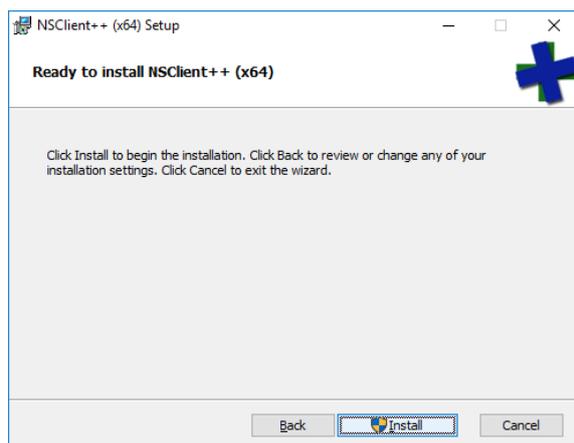
Generische Einstellung für ServerGuard24



Typische Installation installiert alle notwendigen Komponenten



Für Allowed hosts und Password können die Voreinstellungen übernommen werden.



Die Installation kann nun starten.

2.3 Linux Debian Installation

```
apt-get install git build-essential cmake python
apt-get install libssl-dev libboost-all-dev libboost-python-dev
apt-get install libprotobuf-dev protobuf-compiler
apt-get install libcrypto++-dev libcrypto++8
apt-get install libzmq5 libzmq3-dev
apt-get install libtinyxml2-dev libtinyxml2.6.2v5

git clone --recursive https://github.com/mickem/nscp.git
mkdir build
cd build

cmake ../nscp
make
make test
```

In der Version 0.5.3 bricht cmake mit dem Fehler ab, dass die Datei tinyxml2.cpp nicht gefunden werden kann. Hier hilft es, im Verzeichnis /usr/include mit `ln -s tinyxml2.h tinyxml2.cpp` einen Symlink anzulegen.

3 Konfiguration

Die Konfiguration erfolgt in der Datei nsclient.ini. Dort wurden die wichtigsten Einstellungen bereits durch die Angaben während der Installation vorgenommen.

Module

Zur Ausführung des NSClient++ im passiven Modus, muss der NRDPClient und der Scheduler aktiviert werden. Die Aktivierung der weiteren Module hängt davon ab, was Sie auf dem System überwachen möchten.

```
[/modules]

NRDPClient = enabled
Scheduler = enabled
CheckExternalScripts = enabled
CheckHelpers = enabled
CheckDisk = enabled
CheckSystem = enabled
CheckEventLog = enabled
CheckNSCP = disabled
NSCAClient = disabled
```

NRDP Config

Das NRDP Modul meldet die Check Ergebnisse an ServerGuard24 und kann mit Voreinstellungen versehen werden.

```
[/settings/NRDP/client/targets]
[/settings/NRDP/client/targets/default]
token = 12345@jd3udadkao3f
timeout = 30
retries = 3
address = https://psvX.serverguard24.de
```

Den Token legen Sie in den Einstellungen des Intranet Checks fest. Die address entnehmen Sie bitte ebenfalls den Check Eigenschaften. Timeout und Retries bestimmen Dauer sowie die Anzahl der Versuche, die der NSClient unternimmt, die Ergebnisse an ServerGuard24 zu melden.

| Einstellungen | |
|---|-------------------------------|
| Name | CPU Load |
| Adresse | www.sample.de |
| Geografische Mess-Region | Europa |
| Befehl | my_drive |
| Token | 12345@jd3udadkao3f |
| NRDP Adresse | https://psv2.serverguard24.de |
| <input type="button" value="Bearbeiten"/> | |

Einstellungen des Intranet Checks

Check Commands

Check Befehle auf Basis der integrierten Check Module werden in der alias Section definiert.

```
[/settings/external scripts/alias]
my_cpu = check_cpu "warn=load > 80" "crit=load > 90" time=5m
my_memory = check_memory "warn=free < 20%" "crit=free < 10G"
my_drive = check_drivesize "warn=free < 10%" "crit=free < 5%" drive=C:
my_uptime = check_uptime "crit=uptime < 12h"
```

Externe Skripte werden in der scripts Section definiert.

```
[/settings/external scripts/scripts]
my_test_php = php scripts/check_test.php "test"
my_test_vbs = cscript.exe //T:90 //NoLogo scripts/check_test.vbs "test"
my_test_bat = scripts\check_test.bat "test"
```

Check einrichten

Externe Skripte werden in der scripts Section definiert. Der Name der auf `/settings/scheduler/schedules/` folgt, muss in den Check Eigenschaften des Intranet Checks als „Befehl“ eingetragen werden.

```
[/settings/scheduler]
threads=5

[/settings/scheduler/schedules]
[/settings/scheduler/schedules/default]
channel = NRDP

[/settings/scheduler/schedules/my_cpu]
command = my_cpu
interval = 10s

[/settings/scheduler/schedules/my_memory]
command = my_memory
interval = 5m

[/settings/scheduler/schedules/my_drive]
command = my_drive
interval = 1m
```

Die Bezeichnung des Schedulers (z.B. `my_drive`) muss in den Checkeigenschaften im Feld **Befehl** eingetragen werden.

| Einstellungen | |
|---|-------------------------------|
| Name | CPU Load |
| Adresse | www.sample.de |
| Geografische Mess-Region | Europa |
| Befehl | my_drive |
| Token | 12345@jd3udadkao3f |
| NRDP Adresse | https://psv2.serverguard24.de |
| <input type="button" value="Bearbeiten"/> | |

Einstellungen des Intranet Checks

Eine ausführliche Dokumentation aller verfügbaren Check Commands sowie deren Parameter finden Sie unter

<https://docs.nsclient.org/reference/index.html>

Nach erfolgter Konfiguration muss der NSClient Dienst über die Systemsteuerung / Dienste oder die Konsole neu gestartet werden.

```
net stop nscp  
net start nscp
```

4 NSClient++ testen

Der NSClient verfügt über einen Testmodus. Je nach Berechtigung auf die Datei nsclient.log kann es erforderlich sein, nscp mit Adminrechten zu starten.

```
nscp test --log trace
```

Danach können die konfigurierten Checks getestet und Nachrichten abgesendet werden.

```
check_cpu  
check_drivesize  
submit_nrdp --command check_cpu --result OK --message "OK: CPU load is ok.|"total  
5m'=0%;80;90"
```

Die Ergebnisse können auch in der Datei nsclient.log nachgesehen werden.