

Nagios auf Raspi installieren

Installation auf Raspi OS Twister funktioniert doch.

Wahrscheinlich auf allen Raspi OS ist ja alles Linux.

Befehle von folgender Seite übernommen.

Link: <https://pimylifeup.com/raspberry-pi-nagios/>

Nagios ist eine beliebte Open-Source-Software zur Überwachung von Systemen, Netzwerken und Infrastruktur.

Die Befehle zur Installation, falls die Seite vom Web verschwindet:

```
sudo apt update
```

```
sudo apt full-upgrade
```

Pakete installieren:

Dieser Befehl installiert mehrere Pakete, die wir benötigen. Diese Pakete enthalten den Compiler, den wir zum Kompilieren der Nagios-Software benötigen. Außerdem installieren wir den [Apache-Webserver](#), der für die Weboberfläche von Nagios verwendet wird.

```
sudo apt install -y autoconf build-essential wget unzip apache2 apache2-utils php libgd-dev snmp libnet-snmp-perl gettext libssl-dev wget bc gawk dc libmcrypt-dev
```

Herunterladen und Kompilieren von Nagios

```
cd /tmp
```

Nagios-Quellcode auf unseren Raspberry Pi herunterladen.

```
wget -O nagios.tar.gz https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.tar.gz
```

Archiv extrahieren

```
tar xzf nagios.tar.gz
```

Wechseln Sie nun in das Nagios-Verzeichnis und konfigurieren Sie die Software zum Kompilieren.

```
cd /tmp/nagioscore-nagios-4.4.6/
```

```
./configure --with-httpd-conf=/etc/apache2/sites-enabled
```

Wir übergeben den Pfad zu dem Ort, an dem unsere Apache2-Konfiguration gespeichert werden soll. Dieser Pfad wird später im Tutorial verwendet.

Lassen Sie uns nun Nagios kompilieren, indem Sie den folgenden Befehl ausführen.

```
make all
```

Nagios auf dem Raspberry Pi einrichten

Wir nutzen den `make` Befehl zum Erstellen des Benutzers und der Gruppe, die Nagios ausführen muss.

```
sudo make install-groups-users
```

```
sudo usermod -a -G nagios www-data
```

Installieren Sie als Nächstes die kompilierten Binärdateien in unserem Betriebssystem, indem Sie den folgenden Befehl verwenden.

```
sudo make install
```

Führen Sie den folgenden Befehl aus, um den Nagios-Core-Daemon zu installieren.

```
sudo make install-daemoninit
```

Externes Befehlsverzeichnis einrichten.

```
sudo make install-commandmode
```

Unser nächster Schritt besteht darin, die Beispielkonfigurationsdatei erneut zu kopieren

```
sudo make install-config
```

Diese Konfigurationsdateien werden für den Betrieb von Nagios benötigt. Ohne die Konfigurationsdateien wird die Software nicht geladen.

Dieser Befehl installiert die erforderlichen Konfigurationsdateien in dem Verzeichnis, das wir bei der Konfiguration des Makefiles angegeben haben.

Wir werden auch zwei `a2enmod` Befehle verwenden, um sicherzustellen, dass die erforderlichen Apache-Module aktiviert sind.

```
sudo make install-webconf
```

```
sudo a2enmod rewrite
```

```
sudo a2enmod cgi
```

In unserem letzten Schritt erstellen wir einen Apache-Benutzer, mit dem Sie auf die Nagios-Oberfläche auf Ihrem Raspberry Pi zugreifen.

Der folgende Befehl erstellt einen Benutzer namens *nagiosadmin*. Sie werden aufgefordert, ein Passwort für diesen Benutzer anzugeben.

```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

Nagios auf dem Raspberry Pi starten

Unser erster Schritt besteht darin, den Apache-Webserver neu zu starten, indem Sie den folgenden Befehl ausführen.

```
sudo systemctl restart apache2
```

Aktivieren Sie als Nächstes den Nagios-Dienst und starten Sie ihn, indem Sie die folgenden beiden Befehle ausführen.

```
sudo systemctl enable nagios
```

```
sudo systemctl start nagios
```

Durch die Aktivierung des Dienstes erlauben wir Nagios, beim Booten auf Ihrem Raspberry Pi zu starten

Sie können überprüfen, ob Nagios auf Ihrem Raspberry Pi gestartet wurde, indem Sie den folgenden Befehl ausführen.

```
sudo systemctl status nagios
```

Wenn alles wie vorgesehen funktioniert, sollte der folgende Text an das Terminal ausgegeben werden.

```
Active: active (running)
```

Installation der Nagios-Plugins

Damit Nagios richtig funktioniert, müssen wir seine Plugins installieren. Ohne sie wird die Nagios-Software sehr wenig Funktionalität haben.

Wechseln Sie zuerst in unsere `/tmp` Verzeichnis.

```
cd /tmp
```

Nagios-Plugins herunterladen

```
wget -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/releases/download/release-2.3.3/nagios-plugins-2.3.3.tar.gz
```

Extrahieren Sie nun den Plugin-Quellcode in unser aktuelles Verzeichnis, indem Sie den folgenden Befehl verwenden.

```
tar xzf nagios-plugins.tar.gz
```

Als nächstes wechseln wir in unser neu erstelltes Verzeichnis und konfigurieren die Plugins für die Kompilierung.

```
cd /tmp/nagios-plugins-2.3.3
```

```
./configure
```

Sobald der Konfigurationsprozess abgeschlossen ist, können wir die Nagios-Plugins kompilieren, indem wir den folgenden Befehl ausführen.

```
Make
```

Beenden Sie diesen Vorgang, indem Sie die Nagios-Plugins installieren

```
sudo make install
```

Um sicherzustellen, dass Nagios die neuen Plugins lädt, starten Sie die Software neu, indem Sie den folgenden Befehl ausführen.

```
sudo systemctl restart nagios
```

Verbinden mit dem Nagios-Webinterface

Um auf die Nagios-Weboberfläche zugreifen zu können, müssen Sie die IP-Adresse des Raspberry Pi kennen.

Sie können die lokale IP-Adresse Ihres Raspberry Pi abrufen, indem Sie den folgenden Befehl ausführen.

```
hostname -I
```

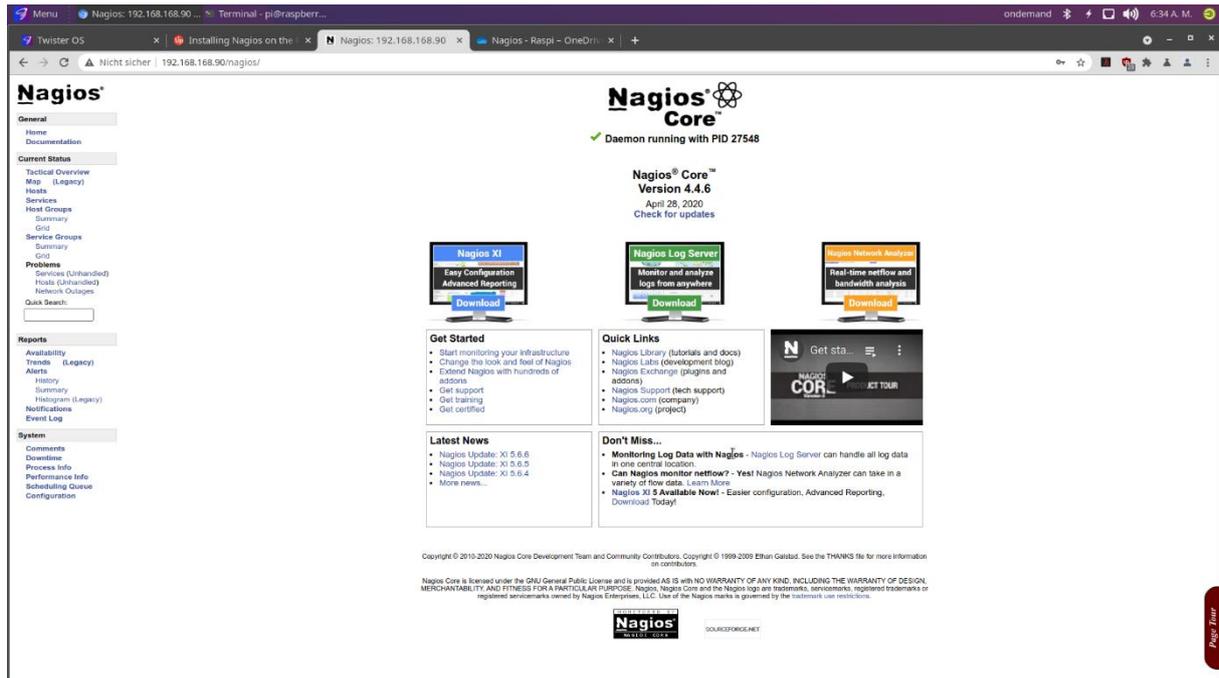
```
ifconfig
```

Um sich mit der Nagios-Weboberfläche zu verbinden, müssen Sie zur IP-Adresse Ihres Pi gehen, gefolgt von `/nagios`.

```
http://127.0.0.1/nagios
```

```
username: nagiosadmin
```

```
Password: xxxx
```



Hinzufügen eines Geräts zu Nagios

In den nächsten beiden Abschnitten zeigen wir Ihnen, wie Sie ein Gerät zur Überprüfung durch Nagios hinzufügen können. Dieser Vorgang ist relativ einfach, erfordert jedoch die Änderung einiger Konfigurationsdateien.

Hinzufügen Ihres Hosts zu Nagios

In diesem Abschnitt zeigen wir Ihnen, wie Sie Ihren Host zu einer Konfigurationsdatei hinzufügen, die Nagios lesen wird.

Dazu müssen wir eine Konfigurationsdatei erstellen, in der wir die verschiedenen Details des Geräts definieren, das wir überwachen möchten.

Wechseln wir zunächst in den Konfigurationsordner „**objects**“ für Nagios.

Dieser Ordner enthält standardmäßig einige Beispielformatdateien. Diese Dateien können nützlich sein, um zu sehen, wie man Nagios konfiguriert.

```
cd /usr/local/nagios/etc/objects
```

Lassen Sie uns nun eine Konfigurationsdatei erstellen, in der wir die Konfiguration unseres Hosts angeben.

Für dieses Beispiel nennen wir diese Datei `pimylifeuphost.cfg`, verwenden Sie jedoch am besten einen Namen, der Ihr Gerät am besten beschreibt.

```
sudo nano laptopithost.cfg
```

Innerhalb dieser Datei müssen wir die folgenden Zeilen eingeben.

Sie müssen sicherstellen, dass Sie die IP-Adresse des Geräts angeben, das Sie überprüfen möchten.

```
define host {  
  
    use          linux-server      ; Host group to use  
  
    host_name    IT-KURT host      ; Name of this host  
  
    alias        laptopithost      ; Alias  
  
    address      192.168.178.49    ; IP Address  
  
}
```

Mit dieser Grundkonfiguration können wir zumindest überwachen, ob das Gerät online ist, indem wir eine Ping-Anfrage an es senden.

Wenn Sie fertig sind, speichern Sie die Datei, indem Sie drücken `CTRL + X`, gefolgt von `Y`, dann `ENTER`

Als nächstes müssen wir unsere Nagios-Konfiguration ändern, damit sie unsere neue Konfigurationsdatei lesen kann.

```
sudo nano /usr/local/nagios/etc/nagios.cfg
```

Suchen Sie in dieser Datei die folgende Zeile und fügen Sie unsere neue Konfigurationszeile darunter hinzu.

Finden

```
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
```

Unten hinzufügen

```
cfg_file=/usr/local/nagios/etc/objects/laptopithost.cfg
```

Wenn wir Nagios jetzt neu starten, wird es wissen, dass es diese Datei lesen und unsere darin definierten Einstellungen laden soll.

Speichern Sie die Konfigurationsdatei, indem Sie `CTRL + X`, gefolgt von `Y` drücken, dann `ENTER`

Wir können jetzt den Nagios-Dienst auf unserem Raspberry Pi neu starten, indem wir den folgenden Befehl verwenden.

sudo systemctl restart nagios

Sie sollten jetzt Ihren neuen Host sehen können, indem Sie zur Seite „**Hosts**“ im Nagios-Dashboard gehen.

The screenshot shows the Nagios web interface. The main content area displays the following information:

- Current Network Status:** Last Updated: Fri Feb 25 10:38:27 C-ST 2022. Updated every 90 seconds. Nagios® Core™ 4.4.4 - www.nagios.org. Logged in as nagiosadmin.
- Host Status Totals:**

Up	Down	Unreachable	Pending
2	0	0	0
- Service Status Totals:**

Ok	Warning	Unknown	Critical	Pending
8	0	0	0	0
- Host Status Details For All Host Groups:**

Host	Status	Last Check	Duration	Status Information
IT-KURT Host	UP	02-25-2022 10:37:50	0d 0h 20m 37s	PING OK - Packet loss = 0%, RTA = 2.19 ms
localhost	UP	02-25-2022 10:33:29	0d 0h 42m 54s	PING OK - Packet loss = 0%, RTA = 0.17 ms

Derzeit führt Nagios nur einen einfachen Ping-Check durch, um zu sehen, ob die Hosts online sind.

In unserem nächsten Abschnitt zeigen wir Ihnen, wie Sie dem Host Dienste hinzufügen können, um Dinge wie den Status von SSH oder seiner Webschnittstelle zu überprüfen.

Hinzufügen eines Dienstes zu Ihrem Host

In diesem Abschnitt zeigen wir Ihnen, wie Sie Ihrem Host ein paar zusätzliche Prüfungen hinzufügen.

Für diese Anleitung richten wir zwei Dienste ein, die die [check_http](#) und [check_ssh](#) auf unserem Host verwenden.

Beide Befehle können verwendet werden, um eine ganze Menge zu tun, aber wir werden ihre grundlegendsten Verwendungen anwenden.

Wir können mit der Änderung dieser Datei beginnen, indem wir den folgenden Befehl auf unserem Raspberry Pi ausführen

sudo nano /usr/local/nagios/etc/objects/laptopithost.cfg

. Innerhalb dieser Datei müssen wir unten einen neuen Abschnitt hinzufügen, der unsere Dienste definiert.

Jeder Dienst muss in einem separaten Block definiert werden, daher erstellen wir für dieses Tutorial zwei dieser Blöcke.

```
define service {  
  
    use                local-service  
  
    host_name          IT-KURT Host  
  
    service_description SSH  
  
    check_command      check_ssh  
  
}
```

```
define service {  
  
    use                local-service  
  
    host_name          IT-KURTHost  
  
    service_description HTTP  
  
    check_command      check_http  
  
}
```

Mit diesen beiden Blöcken überprüft Nagios automatisch den Status der SSH- und HTTP-Ports, die auf unserem angegebenen Host ausgeführt werden.

Wir sollten jetzt überprüfen, ob unsere Konfigurationsänderungen gültig sind. Wir können das tun, indem wir den folgenden Befehl ausführen.

Dieser Befehl weist die Nagios-Software an, den Inhalt der Konfigurationsdateien zu überprüfen.

```
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Wir können Nagios jetzt neu starten, damit unsere Änderungen auf unseren Raspberry Pi geladen werden

```
sudo systemctl restart nagios
```

. Wenn Sie nun die Nagios-Weboberfläche öffnen, gehen Sie zum **Dienste** Menü

Auf dieser Seite können Sie den Status Ihrer SSH- und HTTP-Dienste Ihres neuen Hosts sehen.

The screenshot shows the Nagios web interface. The main content area displays 'Service Status Details For All Hosts' with a table listing services for 'IT-KURKT Host' and 'localhost'. The table has columns for Host, Service, Status, Last Check, Duration, Attempt, and Status Information.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
IT-KURKT Host	HTTP	CRITICAL	02-25-2022 11:03:07	0d 0h 15m 7s	4/4	CRITICAL - Socket timeout
IT-KURKT Host	SSH	CRITICAL	02-25-2022 11:04:07	0d 0h 14m 7s	4/4	CRITICAL - Socket timeout
localhost	Current Load	OK	02-25-2022 11:04:44	0d 0h 13m 21s	1/4	OK - load average: 0.22, 0.57, 0.52
localhost	Current Users	OK	02-25-2022 11:07:14	0d 0h 12m 43s	1/4	USERS OK - 2 users currently logged in
localhost	HTTP	OK	02-25-2022 11:07:14	0d 0h 12m 6s	1/4	HTTP OK: HTTP/1.1 200 OK - 10977 bytes in 0,001 second response time
localhost	PING	OK	02-25-2022 11:08:29	0d 0h 11m 28s	1/4	PING OK - Packet loss = 0%, RTA = 0.13 ms
localhost	Root Partition	OK	02-25-2022 11:05:21	0d 0h 10m 51s	1/4	DISK OK - free space: 49742 MB (86,00% inode=91%)
localhost	SSH	OK	02-25-2022 11:08:36	0d 0h 35m 13s	1/4	SSH OK - OpenSSH_7.9p1 Raspbian-10rdeb10u2rpt1 (protocol 2.0)
localhost	Swap Usage	OK	02-25-2022 11:07:51	0d 0h 9m 36s	1/4	SWAP OK - 100% free (99 MB out of 99 MB)
localhost	Total Processes	OK	02-25-2022 11:08:30	0d 0h 8m 58s	1/4	PROCS OK: 70 processes with STATE = RSZDT

Sie können mehr über die Verwendung von Nagios erfahren, indem Sie sich die [offizielle Nagios Core-Dokumentation ansehen](#).