

Diesen Beitrag schrieb ich vor 3 Jahren. Behalte das beim Lesen bitte im Hinterkopf.

Ich spiele (wie ihr wisst) momentan viel mit Icinga 2 herum, und [einen Host als client node einzusetzen](#) erschien mir sehr verlockend...



Der Grundgedanke hierbei ist, auf jedem zu überwachenden Host den Icinga 2-Core zu installieren, als `node` zu konfigurieren, als Daemon zu starten und über Port 5665 mit dem Master kommunizieren zu lassen; die Konfiguration wird lokal auf dem Host vorgehalten, und über wenige einfache Anweisungen *zieht* sich der Master diese Konfiguration und integriert sie in die bestehende *Landschaft* — womit ich mir zukünftig das unerfreuliche (und unverschlüsselte!) Gehampel mit [NRPE \(Nagios Remote Plugin Executor\)](#) komplett schenken kann. Anhand meiner privaten Mini-Installation stelle ich dir nun meine Vorgehensweise vor.

## Den Master einrichten

Mein Master ist barbapapa, mein RaspberryPi 2 — er muss als erstes eingerichtet werden.

```
master$ icinga2 node wizard
Welcome to the Icinga 2 Setup Wizard!
```

We'll guide you through all required configuration details.

```
Please specify if this is a satellite setup ('n' installs a master setup)
[Y/n]: n
```

An dieser Stelle ist es sehr wichtig, mit `n` zu reagieren, damit ein `master setup` durchgeführt wird — alle Clients werden dann als `satellite setup` geführt. Es wird ein Zertifikat erstellt, das Feature API wird aktiviert (sofern es das nicht schon ist) — die wenigen Rückfragen beantwortete ich, indem ich auf Enter gehauen habe. Abschließend muss der Dienst durchgestartet werden; der Master ist nun einsatzbereit.

```
Now restart your Icinga 2 daemon to finish the installation!
$ service icinga2 restart
```

## Den Client einrichten

Grundsätzlich ist die Vorgehensweise auf dem Client gleich — mit dem Unterschied, dass hier die erste Frage mit `Y` beantwortet werden muss. Client wird im Beispiel mein Node `spiller.me`:

```
client$ icinga2 node wizard
Welcome to the Icinga 2 Setup Wizard!
```

We'll guide you through all required configuration details.

```
Please specify if this is a satellite setup ('n' installs a master setup)
[Y/n]: Y
Starting the Node setup routine...
```

```
Please specify the common name (CN) [spiller.me]:
Please specify the master endpoint(s) this node should connect to:
Master Common Name (CN from your master setup): barbapapa.bafi.lan
Do you want to establish a connection to the master from this node? [Y/n]:
Please fill out the master connection information:
Master endpoint host (Your master's IP address or FQDN): barbapapa.bafi.lan
Master endpoint port [5665]:
Add more master endpoints? [y/N]: N
Please specify the master connection for CSR auto-signing (defaults to master
endpoint host):
Host [barbapapa.bafi.lan]:
Port [5665]:
information/base: Writing private key to '/etc/icinga2/pki/spiller.me.key'.
information/base: Writing X509 certificate to
'/etc/icinga2/pki/spiller.me.crt'.
information/cli: Fetching public certificate from master (barbapapa.bafi.lan,
5665):
```

Certificate information:

```
Subject:      CN = barbapapa.bafi.lan
Issuer:       CN = Icinga CA
Valid From:   Mar  5 11:52:13 2016 GMT
Valid Until:  Mar  2 11:52:13 2031 GMT
Fingerprint: 1B 69 EF BC 0A 7C 01 32 FB 31 98 76 AD 0F 77 D3 12 B5 FA 2F
```

```
Is this information correct? [y/N]: y
information/cli: Received trusted master certificate.
```

Please specify the request ticket generated on your Icinga 2 master.

Derkennst es: der Client verbindet sich über Port 5665 zum Master und greift sich dort das trusted master certificate ab. Nun muss auf dem Master noch ein request ticket für den Client erstellt werden...

```
master$ cd /etc/icinga2
$ grep TicketSalt constants.conf
const TicketSalt = "ziemlichWirreZeichenkette"
```

... und zwar mit diesem Aufruf:

```
master$ icinga2 pki ticket --cn 'spiller.me' --salt
'ziemlichWirreZeichenkette'
andereWirreZeichenkette
```

Zurück in die Konsole des Client: gestoppt hatten wir an der Stelle, an der wir zur Eingabe des request ticket aufgefordert worden waren — genau hier machen wir nun mit dem eben auf dem Master erstellten salt weiter:

```
Please specify the request ticket generated on your Icinga 2 master.
(Hint: # icinga2 pki ticket --cn 'spiller.me'): andereWirreZeichenkette
information/cli: Requesting certificate with ticket
```

'andereWirreZeichenkette'.

```
information/cli: Writing signed certificate to file
'/etc/icinga2/pki/spiller.me.crt'.
information/cli: Writing CA certificate to file '/etc/icinga2/pki/ca.crt'.
Please specify the API bind host/port (optional):
Bind Host []:
Bind Port []:
Accept config from master? [y/N]: y
Accept commands from master? [y/N]: y
information/cli: Disabling the Notification feature.
Disabling feature notification. Make sure to restart Icinga 2 for these
changes to take effect.
information/cli: Enabling the Apilistener feature.
Enabling feature api. Make sure to restart Icinga 2 for these changes to take
effect.
information/cli: Created backup file '/etc/icinga2/features-
available/api.conf.orig'.
information/cli: Generating local zones.conf.
information/cli: Dumping config items to file '/etc/icinga2/zones.conf'.
information/cli: Created backup file '/etc/icinga2/zones.conf.orig'.
information/cli: Updating constants.conf.
information/cli: Created backup file '/etc/icinga2/constants.conf.orig'.
information/cli: Updating constants file '/etc/icinga2/constants.conf'.
information/cli: Updating constants file '/etc/icinga2/constants.conf'.
Done.
```

Now restart your Icinga 2 daemon to finish the installation!

```
$ service icinga2 restart
```

Beachten musst du, dass der Restart des Dienstes nicht vom Wizard übernommen wird — den musst du selbst auslösen. Der Client baut nun eine Verbindung zum Master auf:

```
...
[2016-03-11 09:23:50 +0100] information/ApiListener: Adding new listener on
port '5665'
[2016-03-11 09:23:50 +0100] information/JsonRpcConnection: Reconnecting to
API endpoint 'barbapapa.bafi.lan' via host 'barbapapa.bafi.lan' and port
'5665'
[2016-03-11 09:23:50 +0100] information/ConfigItem: Activated all objects.
...done.
```

## **update-config**

```
master$ icinga2 node list
Node 'spiller.me' (last seen: Sun Mar 6 12:10:20 2016)
  * Host 'spiller.me'
    * Service 'apt'
    * Service 'disk'
    * Service 'disk /'
    * Service 'http'
```

```
* Service 'icinga'  
* Service 'load'  
* Service 'ping4'  
* Service 'ping6'  
* Service 'procs'  
* Service 'ssh'  
* Service 'swap'  
* Service 'users'
```

Der Master kann ihn *sehen* — den Client und alle Services, die (lokal auf dem Client!) definiert sind. Allerdings erfolgen noch keine Checks: der Master muss die Konfiguration des Clients erst noch in seine eigene integrieren, er wird sie mit dem folgenden Kommando nach `/etc/icinga2/repository.d` schieben:

```
$ icinga2 node update-config  
information/cli: Adding host 'spiller.me' to the repository.  
...  
$ service icinga2 reload
```



Nach wenigen Sekunden erscheint der Client nun in der Übersicht und die ersten Checks laufen an. Und änderst du die Konfiguration auf dem Client, so musst du sie auf dem Master wiederum per `icinga2 node update-config` auf den neuesten Stand bringen. (Im Screenshot siehst du auch, wie sich ein Client darstellt, der abgeschaltet wurde — Zone `videorekorder` ist schon seit über zwölf Stunden nicht mehr erreichbar.)

## Fazit

Betrieb und Einrichtung gestalten sich unkompliziert, und nicht zuletzt aufgrund der verschlüsselten Kommunikation gefällt mir die Sache schon sehr gut. NRPE ist buchstäblich aus dem letzten Jahrhundert, es ist umständlich und unsicher — dummerweise ist es aber auch das, was man kennt, in das man sich nicht neu eindenken muss und das schon auch irgendwie *funktioniert*.

Einen üblen Knoten habe ich nun im Gehirn, was das Zusammenspiel von Master und Clients, dem Icinga Director und/ oder Puppet angeht — *best practice* für größere bestehende Strukturen? Uff.