

iX Edition

## Icinga 2

Ein praktischer Einstieg ins Monitoring

Bearbeitet von  
Lennart Betz, Thomas Widhalm

1. Auflage 2016. Taschenbuch. XVI, 334 S. Paperback  
ISBN 978 3 86490 333 5  
Format (B x L): 16,5 x 24 cm

[Weitere Fachgebiete > EDV, Informatik > Computerkommunikation,  
Computervernetzung > Management von Netzwerken](#)

Zu [Inhaltsverzeichnis](#)

schnell und portofrei erhältlich bei

The logo for beck-shop.de features the text 'beck-shop.de' in a bold, red, sans-serif font. Above the 'i' in 'shop' are three red dots of varying sizes, arranged in a slight arc. Below the main text, the words 'DIE FACHBUCHHANDLUNG' are written in a smaller, red, all-caps, sans-serif font.

**beck-shop.de**  
DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung [beck-shop.de](http://beck-shop.de) ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

Das Berechtigungsmodell ist sehr einfach gehalten. Der angemeldete Benutzer sieht standardmäßig lediglich solche Hosts und Services, für die er bei einem Fehlerfall als zu benachrichtigender Kontakt eingetragen ist. Sonst ist es nur noch möglich, die Einstellungen in der Datei `/etc/icinga/cgi.cfg` so zu konfigurieren, dass er alle Objekte einsehen darf (»`authorized_for_all_hosts`« bzw. »`authorized_for_all_services`«).

In dieser Datei können noch weitere Berechtigungseinschränkungen bzw. Erweiterungen vorgenommen werden, aber auch solche Einstellungen zur optischen Anpassung. Ausführliche Informationen finden sich in der Onlinedokumentation<sup>4</sup> des Icinga-Projektes.

## 2.4 Grundkonfiguration von Icinga 2

Eine Übersicht der einzelnen Dateien befindet sich in Anhang A.1.1 ab Seite 285.

Die zentrale Konfigurationsdatei ist `icinga2.conf`. Alle anderen verwendeten Dateien werden direkt oder indirekt von dort im Stil von C/C++ aus eingebunden.

```
include "constants.conf"
```

In diesem Auszug wird `constants.conf` eingebunden. Benutzt man Double Quotes und keinen absoluten Pfad, wird im selben Verzeichnis als Ausgangspunkt gesucht, hier `/etc/icinga2`.

```
include <itl>
include <plugins>
```

```
include "features-enabled/*.conf"
```

Hingegen wird bei der Verwendung von `<...>` im vorgegebenen Pfad `/usr/share/icinga2/include` nach der entsprechenden Datei gesucht. Dieser Pfad kann sich je nach Installationspaket bzw. Plattform unterscheiden.

```
include_recursive "conf.d"
```

Eine weitere Direktive ist »`include_recursive`«, bei der keine Datei, sondern ein Verzeichnis angegeben wird. Hierbei werden alle Dateien in diesem Verzeichnis und rekursiv absteigend auch alle Dateien in Unterverzeichnissen mit eingelesen.

---

<sup>4</sup><http://docs.icinga.org/latest/de/configcgi.html>

### 2.4.1 Konstanten

Wie in `icinga2.conf` zu sehen, ist `constants.conf` nach `init.conf` die erste der Konfigurationsdateien, die gelesen wird. In ihr werden Konstanten gesetzt, die somit in den nachfolgenden Konfigurationsdateien zur weiteren Benutzung zur Verfügung stehen.

```
const PluginDir = "/usr/lib64/nagios/plugins"  
//const NodeName = "localhost"
```

**Codebeispiel 2.1:** `/etc/icinga2/constants.conf`

Standardmäßig ist »PluginDir« gesetzt und enthält als Wert den Pfad zu den Plugins. Diese Konstante wird dann z. B. in der Icinga Template Library (ITL) innerhalb der Definitionen von Check Commands verwendet. Wird »NodeName« nicht angegeben, ist automatisch der Fully Qualified Domain Name (FQDN) für diese Konstante gesetzt.

### 2.4.2 Icinga Template Library

Bei der ITL handelt es sich um eine Sammlung von Templates und Check Commands. Check Commands regeln in Icinga 2 den Aufruf der Plugins. Zur Erinnerung: Ein Plugin ist ein Programm, das durch eigene Logik den Status eines Service oder Hosts ermittelt. Das heißt, in einer Servicedefinition wird das anzuwendende Check Command angegeben und damit wird bestimmt, welches Plugin für einen Check dieses Service ausgeführt wird.

Die ITL befindet sich in `/usr/share/icinga2/include` und besteht aus mehreren unterschiedlichen Dateien, die sich ggf. gegenseitig einbinden. So zieht »include <plugins>« unter anderem das Laden von `command-plugins.conf` nach sich. In dieser Datei befinden sich Check-Command-Definitionen zu den Plugins aus dem Monitoring-Plugins-Projekt<sup>5</sup>.

```
include "command-plugins.conf"
```

**Codebeispiel 2.2:** `/usr/share/icinga2/include/plugins`

Die Datei `itl` selbst lädt Templates und Methoden nach, wie z. B. »plugin-check-command«, das den Aufruf von Plugins steuert.

Auf diese Weise werden Check-Command-Definitionen für häufig verwendete Plugins mitgeliefert. Im Gegensatz zu anderen Teilen der Konfiguration gibt es nicht viel Wahlmöglichkeit in diesen Definitionen, außerdem sind sie oft lang und eintönig. Durch die fertig ausformulierten Check-Command-Definitionen in der ITL nimmt das Icinga-Team den Anwendern viel Arbeit im Alltag ab. Es lohnt sich, immer wieder einen Blick in die ITL

---

<sup>5</sup><https://www.monitoring-plugins.org>

zu werfen, da sie mit neuen Icinga-2-Versionen regelmäßig erweitert wird. Eigene Check Commands sollte man übrigens keinesfalls in der `itl` hinzufügen, da diese von den Paketen nicht als Konfigurationsdatei angesehen und bei Updates einfach überschrieben wird. Besser aufgehoben sind sie stattdessen in einer eigenen Datei im gleichen Verzeichnis wie die Service-Definitionen, egal ob sie in `/etc/icinga2/conf.d` oder in einer globalen Zone, wie in Abschnitt 6.3 ab Seite 104 beschrieben, liegt.

### 2.4.3 Features

Viel der Funktionalität von Icinga 2 ist in sogenannte Features ausgelagert, die bei Bedarf hinzugeladen werden. Alle zur Auswahl stehenden Features können über einen CLI-Aufruf angezeigt werden. Die Ausgabe zeigt zusätzlich an, ob diese zurzeit aktiviert oder deaktiviert sind.

```
# icinga2 feature list
Disabled features: api debuglog icingastatus combatlog ...
Enabled features: checker command mainlog notification
```

In der Datei `icinga2.conf` werden mittels »include« alle Dateien mit der Endung `.conf` im Verzeichnis `/etc/icinga2/features-enabled` eingebunden. Die dort liegenden Dateien enthalten die Konfiguration des jeweiligen Features mit demselben Namen.

```
# ls -l /etc/icinga2/features-enabled/
insgesamt 0
lrwx... checker.conf -> ../features-available/checker.conf
lrwx... command.conf -> ../features-available/command.conf
lrwx... mainlog.conf -> ../features-available/mainlog.conf
lrwx... notification.conf -> ../features-a.../notification.conf
```

Ist ein Feature aktiviert, zeigt ein Link aus diesem Verzeichnis auf die entsprechende Datei in `/etc/icinga2/features-available`.

```
# ls /etc/icinga2/features-available/
api.conf          command.conf      gelf.conf
ido-mysql.conf    mainlog.conf      statusdata.conf
compatlog.conf    perfddata.conf   graphite.conf
ido-pgsql.conf    notification.conf syslog.conf
checker.conf      debuglog.conf    icingastatus.conf
livestatus.conf
```

Aktiviert man beispielsweise das Feature »api« mit `icinga2 feature enable api` oder deaktiviert es entsprechend mit `icinga2 feature disable api`, muss Icinga 2 danach jeweils neu gestartet werden.

Hier folgt eine Auflistung und eine Kurzbeschreibung der Features und ihrer Funktion. Die Features sind jeweils eigene Objekte und werden entsprechend über Attribute konfiguriert. Standardmäßig sollte die Konfigu-

ration über oben stehende Dateien erfolgen. Ausführlich werden sie in den entsprechenden Kapiteln, in denen sie Anwendung finden, erläutert.

## ApiListener

Mit »api« lässt sich der API Listener steuern. Dieser stellt den TCP-Socket zur Kommunikation mit anderen Icinga-2-Instanzen bereit. Andere Instanzen können dabei Satelliten-Systeme oder auch Icinga-2-Agenten sein.

## CheckerComponent

Das Feature »checker« regelt das Ausführen der aktiven Checks, also den Aufruf der Plugins und das Entgegennehmen der Ergebnisse, d. h. den Exit Code und Stdout mit Output und Performance-Daten.

## CheckResultReader

Bildet zur Kompatibilität das »checkresults«-Verzeichnis von Icinga 1 nach. Dies erleichtert die Migration von verteilten Monitorumgebungen.

## CompatLogger

Das »compatlog« kümmert sich um ein zu Icinga 1 kompatibles Logfile `icinga.log` und dessen Rotation. Der Ablageort für das aktuelle Logfile ist `/var/log/icinga2/compat` und für die archivierten Dateien unterhalb von `archives`. Das Feature wird z. B. für die Icinga Classic UI benötigt, um historische Daten anzuzeigen oder auszuwerten.

## ExternalCommandListener

Mit dem Aktivieren des Features »command« wird die Commandpipe `/var/run/icinga2/cmd/icinga2.cmd` zur Verfügung gestellt. Dabei handelt es sich um einen Unix-Socket, in den Anweisungen in einer von Icinga 1 übernommenen Syntax direkt an Icinga 2 übergeben werden können. Über diese kann Icinga 2 weitreichend gesteuert werden. So können Checkergebnisse eingereicht, Downtimes geplant, Probleme bestätigt und einiges mehr über diesen Weg in den Icinga-2-Kern gebracht werden. Allerdings bleiben diese Informationen nur bis zum Neustart von Icinga 2 gültig, da ab dann wieder die Information aus der statischen Konfiguration zählt.

Für grafische Frontends, mit denen Kommandos gesendet werden dürfen, ist dieses Feature erforderlich. Auch zum Erhalt passiver Checkergebnisse ist eine Commandpipe Voraussetzung.

## FileLogger

Mit »mainlog« wird das Logfile `/var/log/icinga2/icinga2.log` geschrieben. Das Logging lässt sich in den Abstufungen »error«, »warning«, »notice«, »information« und »debug« konfigurieren.

Über »debuglog« kann das Logfile `/var/log/icinga2/debug.log` für Debug-Meldungen angeschaltet werden.

Die Logfiles werden nicht automatisch durch Icinga 2 rotiert, hier muss auf einen externen Mechanismus zurückgegriffen werden. Bei der Installation aus Paketen ist »logrotate« hierfür vorkonfiguriert.

## GelfWriter

Bietet mit »gelf« eine Schnittstelle (Graylog Extended Log Format) zu Graylog<sup>6</sup>.

## GraphiteWriter

Die Schnittstelle »graphite« stellt die Verbindung zu Graphite<sup>7</sup> zur Visualisierung und Analyse der Performance-Daten her.

## IcingaStatusWriter

Schreibt periodisch einen Dump des aktuellen Status und der Performance-Daten von Icinga 2 und allen aktivierten Features in eine JSON-Datei. Dieses Feature ist seit Icinga 2 2.4 abgekündigt und wird mit 2.5 entfallen.

## IdoMysqlConnection

Dieses Feature ist aus dem Paket »icinga2-ido-mysql« gesondert zu installieren. Mit »ido-mysql« werden die anfallenden Daten, wie aktueller Status, Statuswechsel und vieles mehr, in eine MySQL-Datenbank geschrieben. Von dort können sie dann von Add-Ons wie Icinga Web 2 oder NagVis<sup>8</sup> ausgelesen werden.

## IdoPgsqlConnection

Feature zum Ablegen der selben Daten wie bei IdoMysqlConnection in eine Postresql-Datenbank. Für dieses Feature ist das Paket »icinga2-ido-pgsql« zusätzlich zu installieren.

---

<sup>6</sup><http://www.graylog.org>

<sup>7</sup><http://graphite.wikidot.com>

<sup>8</sup><http://www.nagvis.org>

## LivestatusWriter

Aktiviert mit »livestatus« die Schnittstelle selbigen Namens. Verfügbar als TCP- oder UNIX-Socket. Um historische Daten über Livestatus abfragen zu können, wird außerdem der »CompatLogger« benötigt.

## NotificationComponent

Aktiviert bzw. deaktiviert das Benachrichtungssystem von Icinga 2.

## PerfdataWriter

Schreibt Performance-Daten nach `/var/spool/icinga2/perfdata` für die Weiterverarbeitung mit PNP4Nagios<sup>9</sup>.

## StatusDataWriter

Feature »status« zur Kompatibilität schreibt die Datei `status.dat` in das Verzeichnis `/var/cache/icinga2`, die erforderlich ist, um aktuelle Daten in der Classic UI anzuzeigen.

## SyslogLogger

Leitet mit der Aktivierung von »syslog« die Logeinträge in ein syslog-Target um. Die Verwendung von »syslog« anstelle von »mainlog« reduziert den Plattenzugriff. Syslog schreibt nicht unmittelbar auf die Platte, sondern puffert Einträge erst, um sie dann gesammelt zu schreiben, und kann auch genutzt werden, um die Logs nicht (nur) am Icinga-2-Host abzulegen, sondern an einen zentralen Loghost zu senden.

## 2.5 Erläuterung der Konfiguration

Die vom Icinga-Projekt mitgelieferte Beispielf Konfiguration befindet sich im Verzeichnis `/etc/icinga2/conf.d` und definiert eine Grundüberwachung von Icinga und dem Host auf dem es läuft. Darüber hinaus bieten die diversen Beispiele einen guten ersten Einblick in die Möglichkeiten der neuen Icinga-2-Konfigurationsprache. Ist in den folgenden Abschnitten für Dateien kein absoluter Pfad angegeben, bezieht dieser sich immer relativ zu `/etc/icinga2`.

---

<sup>9</sup><http://www.pnp4nagios.org>