Lennart Betz · Thomas Widhalm

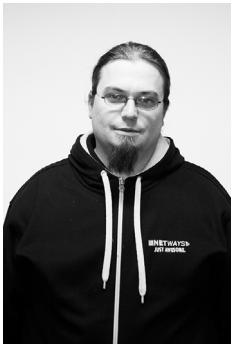
Icinga

Monitoring – Grundlagen und Praxis

iX EDITION

dpunkt.verlag

Lennart Betz arbeitet als Consultant und Trainer bei der Nürnberger NETWAYS GmbH. Seine Hauptarbeitsgebiete sind Planung, Aufbau und Betreuung von Monitoringlösungen, Konfigurationsmanagement und weitere Automatisierungsthemen. Schon früh während seines Mathematikstudiums beschäftigte er sich mit Freier Software und verfolgt dies auch seit dem Abschluss in seiner beruflichen Tätigkeit konsequent weiter.



Thomas Widhalm hilft als Lead Support Engineer Kunden der Netways GmbH beim Beheben von Problemen mit Icinga-Installation. Außerdem unterstützt er als Consultant Kunden bei der Planung, Umsetzung und weiteren Betreuung von Projekten im Bereich Monitoring und Logmanagement. Als Trainer zeichnet er sich für die Schulungen im Bereich Logmanagement verantwortlich. Im Icinga-Team arbeitet er an der Online-Dokumentation mit. Er ist überzeugt, dass Freie Software proprietärer Software überlegen ist und ihre Konzepte auch außerhalb der IT mehr Anwendung finden sollten.

Lennart Betz · Thomas Widhalm

Icinga

Monitoring – Grundlagen und Praxis

dpunkt.verlag

iX EDITION

Lennart Betz, Thomas Widhalm
feedback@icinga-book.net

Lektorat: Dr. Michael Barabas
Projektkoordinierung: Anja Weimer
Copy-Editing: Ursula Zimpfer, Herrenberg
Satz: Lennart Betz
Herstellung: Stefanie Weidner, Frank Heidt
Umschlaggestaltung: Helmut Kraus, *www.exclam.de*
Druck: mediaprint solutions GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über *http://dnb.d-nb.de* abrufbar.

ISBN:
Print 978-3-86490-879-8
PDF 978-3-96910-622-8
ePub 978-3-96910-623-5
mobi 978-3-96910-624-2

1. Auflage 2022

Wieblinger Weg 17
69123 Heidelberg

Hinweis:
Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.

PEFC zertifiziert
Das Papier für dieses Buch stammt aus nachhaltig bewirtschafteten Wäldern und kontrollierten Quellen.
PEFC
PEFC/04-31-0810
www.pefc.de

Schreiben Sie uns:
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es unswissen: *hallo@dpunkt.de.*

5 4 3 2 1 0

Vorwort

Wie mache ich es richtig? Ich vermute durchaus, dass dies eine Frage ist, die Sie zum Kauf dieses Buchs bewogen hat. In den vielen Jahren, die wir als Icinga und die Autoren das Produkt begleiten, ist uns diese Frage ebenfalls immer wieder begegnet. Allerdings gibt es für viele Fragestellungen und Szenarien im Bereich der Infrastrukturüberwachung oft nicht die eine richtige Antwort, um ans Ziel zu kommen. Umgebungen sind unterschiedlich und somit verändert sich auch der passende Lösungsansatz.

Dieses aktualisierte Icinga-Buch geht nochmals detaillierter auf die unterschiedlichen Herausforderungen moderner IT und den Einsatz von Icinga ein. Dabei grenzt sich das Werk vor allem durch die geradlinige Führung des Lesers von anderen Werken und zugegebenermaßen auch der Dokumentation ab. Die Autoren nehmen Sie an die Hand und helfen Ihnen beim richtigen Konfigurieren und Betrieb von Icinga. Dabei ist trotz der Balance zwischen der Betrachtung unterschiedlicher Bedürfnisse stets ein roter Faden erkennbar, der dieses Buch besonders macht. So wird es Sie über den Einstieg und die Konfiguration hinaus im täglichen Betrieb begleiten und Ihnen die Überwachung unterschiedlicher Systeme und Umgebungen nahebringen. Außerdem geht das Buch auf Fragestellungen ein, die sich dem erfahrenen Anwender und Administrator von Icinga stellen: Virtualisierte Umgebungen, APIs, Visualisierung zeitlicher Verläufe und das Thema Hochverfügbarkeit sind detailliert und gleichzeitig leicht zugänglich beschrieben.

Dass die Autoren darüber hinaus über jahrzehntelange praktische Erfahrung mit Icinga verfügen, macht das Buch zu einem Standardwerk für den richtigen Einsatz von Icinga. Der Leser profitiert von dieser Erfahrung, die dem gesamten Buch immer wieder zu entnehmen ist. Daher kann ich Sie zum Kauf des Buchs mit bestem Wissen und Gewissen nur beglückwünschen.

Egal ob Einsteiger oder Fortgeschrittener, wünsche ich Ihnen bei der Einrichtung und dem Betrieb von Icinga mindestens genauso viel Freude wie beim Studieren dieses Buchs. Alles, was Sie dafür brauchen, halten Sie in Ihren »Händen«.

Bernd Erk
CEO Icinga

Einige Zeilen zum Buch selbst

Das vorliegende Buch richtet sich an Administratoren, die Icinga einsetzen, mit dem Gedanken spielen, dies zu tun, oder in absehbarer Zeit zu Icinga migrieren möchten. Die zentralen Komponenten einer Überwachung mit Icinga sind Linux-basiert. Daher ist ein Grundwissen von Linux zwingend für das Verständnis der Materie erforderlich. Das Buch richtet sich aber gleichermaßen auch an erfahrene Linux-Administratoren, die Icinga bereits einsetzen. Es behandelt auch Schnittstellen, die es ermöglichen, Icinga mit anderer Software zu verknüpfen, wie die Integration von Messdaten und deren Darstellung (Graphing) über einer Zeitachse oder das Einbinden von Erkenntnissen aus einem Logmanagement. Nebenbei sind viele Vorschläge enthalten, wie die Überwachung selbst erweitert werden kann.

Struktur dieses Buchs

Dieses Buch besteht aus fünf Teilen, die sich wiederum in einzelne Kapitel gliedern. Der erste Teil gibt eine kleine Einführung in das Thema Monitoring und stellt die einzelnen Softwarekomponenten von Icinga vor.

Der zweite Teil stellt den Icinga-Agenten vor und beschreibt, wie mit dem Agenten Linux- und Windows-Systeme zu überwachen sind.

Teil drei ist der praktischen Überwachung mit unterschiedlichen Plugins vorbehalten. Es werden einige Plugins vorgestellt und beschrieben, wie diese zu installieren und in Icinga zu integrieren sind.

In den letzen beiden Teilen wird darauf eingegangen, wie Icinga erweitert wird und sich in andere Systeme integrieren lässt. Aber auch die Realisierung von verteilter Überwachung mit Icinga und die Erhöhung der Ausfallsicherheit des eigenen Monitorings werden thematisiert.

Der Anhang ist nicht nur ein Nachschlagewerk, sondern vertieft und erweitert Ihr Wissen um die Themenkomplexe Tuning, Security, Updates und Datensicherung.

In diesem Buch behandelte Versionen

Dieses Buch behandelt Icinga 2 in der Version 2.13 und Icinga Web 2 Version 2.8 und Version 2.9. Alle Beispiele zu Installationen beziehen sich, wenn nicht anders angegeben, auf die Distributionen RHEL/CentOS 8 bzw. Debian Bullseye. Dabei wird RHEL wie auch CentOS mit RedHat synonym verwendet.

Das in mehreren Kapiteln eingesetzte und in seiner Benutzung beschriebene MySQL wird ebenfalls als Synonym für MariaDB benutzt.

Typografische Konventionen

In diesem Buch werden folgende typografischen Konventionen verwendet:

- `Nichtproportionalschrift`
  Wird benutzt für Namen von Programmen, Befehlen sowie für Codebeispiele.
- *Kursivschrift*
  Kommt zum Einsatz bei spezifischen Wörtern in Konfigurationen wie Schlüsselwörtern, Objektnamen, deren Attributen oder Custom Variables.
- *`Nichtproportionalschrift kursiv`*
  Wird bei Datei- und Verzeichnisnamen verwendet.

> Kommandos oder Codezeilen, die dem Format dieses Buchs geschuldet einem unnötigen oder falschen Zeilenumbruch unterworfen sind, haben am Ende der Zeilen einen Backslash »\«.

Materialien zum Buch

Die Codebeispiele in diesem Buch, die sich auf die Konfiguration von Icinga beziehen, befinden sich auch online zugänglich in einem Repository[1] auf GitHub. Kommandos auf der Kommandozeile müssen wohl oder übel abgetippt werden.

```
$ git clone https://github.com/lbetz/icinga-book
```

Die Beispiele befinden sich unterhalb des Verzeichnisses */code*. Aufgetretene und entdeckte Fehler an den Beispielen werden wir ebenfalls dort korrigieren sowie generelle Korrekturen am Text unterhalb */errata* hochladen.

Die Beispiele sind nach Kapiteln in Unterverzeichnisse aufgeteilt und dort als Textdateien abgelegt, deren Namen der jeweiligen Codebeispielnummer entsprechen.

[1] https://github.com/lbetz/icinga-book

Danksagung

Lennart Betz

Danke an all die lieben und engagierten Menschen, die sich mit Icinga beschäftigen und durch ihre direkte Arbeit, Zuarbeit durch Issues oder in der Community dieses Projekt ermöglichen. Vielen Dank an Bernd als treibende Kraft, der mich auch moralisch immer unterstützte. Mein besonderer Dank geht auch an Thomas, der mir nicht die Freundschaft kündigte, obwohl ich dieses Buch nahezu ohne ihn geschrieben habe.

Servus
Lennart

Thomas Widhalm

Ich möchte mich bei all den Leuten bedanken, die mich zu dem gemacht haben, der ich heute bin. Vor allem bei meiner Frau, auch weil sie die vielen »Ich muss heute unbedingt an diesem Buch arbeiten«-Tage ertragen hat. Ganz besonders bedanken möchte ich mich bei Lennart, weil er mir die Möglichkeit gegeben hat, genug zu dieser Auflage beizutragen, um meinen Namen am Umschlag zu rechtfertigen. Und das, obwohl ich mich aus persönlichen Gründen weitgehend aus dem Projekt zurückziehen musste und er fast alles alleine gestemmt hat.

Vielen Dank für alles,
Thomas Widhlam

Feedback

Die Qualität jedes Fachbuchs misst sich am Nutzen, den es für seine Leser hat. Wir würden uns freuen, davon zu hören, was Sie als Leser nützlich im Buch fanden und wo wir uns noch verbessern können. So wie sich die beschriebene Software weiterentwickelt, soll es auch unser Buch tun und hiermit geben wir Ihnen die Chance, die Richtung zu beeinflussen, in die es geht.

Für Rückmeldungen zum Buch freuen wir uns über E-Mails an *feedback@icinga-book.net*.

Inhaltsverzeichnis

Einführung

1 Einleitung

Drei Freunde aus den USA, die als Juniordetektive tätig sind, baten uns um Unterstützung bei der Überwachung der IT-Systeme eines großen Secondhandshops. Das »Gebrauchtwarencenter Jonas«, auf dessen Gelände unsere Freunde ihre Zentrale eingerichtet hatten, war ein voller Erfolg und so wurde auch die IT mit der Zeit immer umfangreicher.

Der Erfolg führte zu neuen Services und Applikationen, die natürlich ebenfalls überwacht werden mussten, um rund um die Uhr den reibungslosen Verkauf zu gewährleisten. Dieses Buch beschreibt, wie Icinga im Hintergrund half, diese Ziele zu erreichen, und welchen Weg ich dabei gegangen bin.

Der erste Teil des Buchs definiert, was Monitoring mit Icinga genau bedeutet. Er beschreibt die Architektur von Icinga und führt alle wichtigen Begriffe ein. Nach der Installation werfen wir einen ersten Blick auf die Benutzeroberfläche und gehen anschließend noch auf die Konfiguration der Kernkomponenten ein. Am Ende des ersten Teils wird die Umgebung des Gebrauchtwarencenters vorgestellt und sogleich damit begonnen diese zu überwachen.

Im zweiten Teil steht der Icinga-Agent im Mittelpunkt. Dieser wird zur Überwachung von Linux- und Windows-Systemen verwendet. Mit seiner Hilfe kann beispielsweise die Auslastung der CPU, des Hauptspeichers und der Dateisysteme ermittelt und an Icinga gemeldet werden. Als Alternative zum Agenten, inbesondere für ältere Linux- oder andere Unix-Systeme, wird die Überwachung unter Zuhilfnahme der Secure Shell (SSH) vorgestellt.

Der dritte Teil widmet sich bereits fortgeschritteneren Themen. So startet dieser mit Erläuterungen, wie sich die Benutzeroberfläche Icinga Web 2 weiter ausreizen lässt. Das Formulieren von Filtern und das Anlegen eigener Dashboards wird behandelt, aber auch, wie Icinga Web 2 durch zusätzliche Module erweitert werden oder zur Authentifizierung an ein Microsoft Active Directory (AD) angebunden werden kann. Die Konzepte von Benachrichtigungen und der verteilten Überwachung befinden sich enbenfalls in diesem Teil, wie auch die Einführung in den Icinga Director, der eine grafisch unterstützte Konfiguration ermöglicht.

Ausschließlich um weitere Plugins geht es in Teil vier. Es werden weitere Plugins vorgestellt sowie deren Installation und Anwendung erläutert. Es wird gezeigt, wie Netzwerkdienste der eigenen Infrastruktur überwacht werden, aber auch die zugrunde liegende Hardware sowie Datenbanken und Application Server.

Um Integration geht es im letzten Teil. Hier werden zunächst Businessprozesse eingeführt und gezeigt, wie solche in Icinga definiert und überwacht werden. Die Integration anderer Systeme ist ebenfalls Thema dieses Teils. Kapitel 21 ab Seite 447 beschäftigt sich sehr ausführlich damit, wie von Icinga erfasste Daten in Zeitreihen gespeichert werden können und in die Benutzerobfläche einzubinden sind. Die Überwachung auf bestimmte Logeinträge zeigt Kapitel 23. Zuvor wird jedoch in Kapitel 22 die hiefür erforderliche Schnittstelle von Icinga vorgestellt.

Den Abschluss bildet ein nicht ganz triviales Thema: Hochverfügbarkeit. Es wird erörtert, was wann zu tun ist, um die Fehlertoleranz und damit die Verfügbarkeit einzelner Komponenten oder des Gesamtsystems zu erhöhen. Da das Thema sehr komplex ist und mitunter zusätzliche Software erfordert, nähert sich dieses Kapitel der Problematik auf theoretischer Ebene.

Der Anhang ist nicht nur ein Nachschlagewerk, sondern vermittelt in Abschnitt A viel Wissen über Updates, Datensicherung, Tuning, Troubleshooting und die Absicherung der eigenen Icinga-Umgebung. Die Installation einer Icinga-Umgebung per Hand und aus den Paketquellen beschreibt Anhang D. Der Anhang E gibt hingegen einen kurzen Ausblick auf die IcingaDB, die mittelfristig den IDO als Backend ersetzen soll, wodurch dessen Performanceprobleme der Vergangenheit angehören sollen.

Wer Interesse an der Geschichte rund um Icinga hat, sei hier auf Anhang B verwiesen, der diese kurz zusammengefasst.

Wenn nicht anders beschrieben, sind alle Beispiele, was Installation und Betrieb betrifft, auf RedHat Enterprise Linux (RHEL) in der Version 8 und Debian Bullseye ausgelegt. Mit ein wenig Erfahrung lassen sich die Anleitungen und Beispiele aber auch auf andere Distributionen übertragen.

»Es sind 106 Meilen bis Chicago, wir haben einen vollen Tank, ein halbes Päckchen Zigaretten, es ist dunkel und wir tragen Sonnenbrillen.«

Elwood Blues, 1980

So viel zum Überblick, welche Themen behandelt werden und wo sie zu finden sind. Nun genug der Worte und viel Spaß mit Icinga!

1.1 Monitoring

Der Begriff »Monitoring«[1] bezeichnet die Überwachung von Vorgängen. Es ist ein Überbegriff für alle Arten von systematischen Erfassungen, Messungen oder Beobachtungen eines Vorgangs oder Prozesses mittels technischer Hilfsmittel oder anderer Beobachtungssysteme.

Eine wichtige Funktion des Monitorings besteht darin, bei einem beobachteten Ablauf oder Prozess festzustellen, ob dieser den gewünschten Verlauf nimmt und bestimmte Schwellenwerte eingehalten werden, um andernfalls steuernd eingreifen zu können. Monitoring ist deshalb ein Sondertyp des Protokollierens.

Übertragen auf die IT definiert Wikipedia den »Service-Monitor«[2]. Ein Service-Monitor ist ein Programm, das auf einem Server läuft und Systemressourcen von Servern und deren Rechnernetze überwacht. Des Weiteren gibt es Dienste, die verschiedene Funktionen des Servers von extern in kontinuierlichen Zeitabschnitten überprüfen und die Ergebnisse entsprechend aufbereiten. Bei Fehlfunktionen wird der Betreiber des Servers oder der Services benachrichtigt.

Ohne Monitoring können die Betreiber bzw. die verantwortlichen Administratoren von IT-Systemen nur dann reagieren, wenn ein Anwender eine Störung meldet, sie diese Störung selbst zufällig feststellen oder ein abhängiges System einen Fehler meldet. **Bei dem heute üblichen Verhältnis zwischen der Anzahl von Systemen zu deren Betreuern ist eine kontinuierlich stattfindende, manuelle Kontrolle nicht mehr zu realisieren**. Meldet hingegen ein Anwender einen Ausfall, ist dies nicht nur für genau diesen ärgerlich, sondern auch für alle anderen, die zeitgleich auf das gestörte System zugreifen. Durch die massive Parallelisierung sind dies z. B. bei der Webseite eines Onlineshops viele Tausend potenzielle Kunden, die nun möglicherweise bei einem Marktbegleiter kaufen. Genau solche Ausfälle sollen durch ein Monitoring-System vermieden werden, in dem Probleme frühzeitig erkannt werden und rechtzeitig Gegenmaßnahmen ergriffen werden können.

Gegenmaßnahmen lassen sich nur einleiten, wenn Probleme den Verantwortlichen auch bekannt gemacht werden. Ein Monitoring-System muss über Mechanismen verfügen, Personen oder Gruppen von Personen aktiv zu benachrichtigen.

[1] https://de.wikipedia.org/wiki/Monitoring

[2] https://de.wikipedia.org/wiki/Service-Monitor

1.2 Das Universum um Icinga

Eine Installation zum Monitoring mit Icinga besteht grundlegend aus vier Komponenten und wird ausschließlich auf Linux unterstützt. Die Überwachung weiterer Plattformen wie Windows ist aber natürlich gegeben. Die Kernkomponente, die alles steuert, ist Icinga 2.

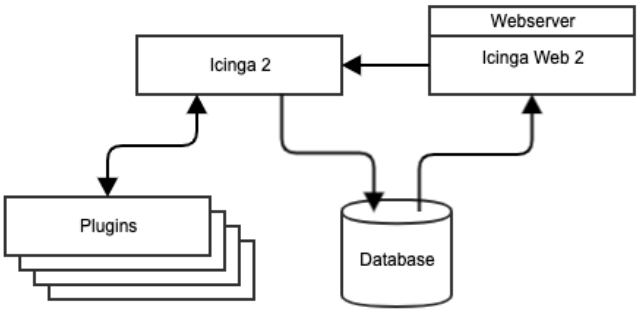


Abbildung 1-1: Die einzelnen Icinga-Komponenten

Die Anzeige des aktuellen Status übernimmt Icinga Web 2, eine Datenbank wird zum Datenaustausch zwischen beiden Komponenten benötigt.

Aus der Benutzeroberfläche Icinga Web 2 heraus steuert der Anwender Icinga 2 interaktiv. Einzelne spezialisierte Plugins übernehmen die Überwachung und Benachrichtigung.

1.2.1 Plugins

Der Name Plugin ist irreführend, da es sich jeweils um ein externes Programm oder Skript handelt. Sie nehmen dabei gesonderte Aufgaben wahr und unterteilen sich danach in Check-Plugins, die den Zustand einer zu überwachenden Komponente ermitteln, und Notification-Plugins, die sich um die Benachrichtigung bei erkannten Problemen kümmern.

Check-Plugins

Bei Check-Plugins gibt der Rückgabewert, auch Exit Code oder Return Code (RC) genannt, den Status der Überwachung an. Beendet sich ein Plugin mit »0«, entspricht das einem OK und bedeutet, der überwachte Dienst funktioniert im Rahmen normaler Parameter.

Die Auswahl des Plugins entscheidet darüber, was überwacht wird. Durch Setzen von Optionen beim Plugin-Aufruf wird angegeben, wo der Dienst läuft, wie er zu überwachen ist und welche Schwellen gelten sollen, um den Zustand OK von WARNING mit dem Exit Code »1« und CRITICAL mit »2« abzugrenzen.

```
$ ./check_http -H www.google.com -w 0.5 -c 1
HTTP OK: HTTP/1.1 200 OK - 13430 bytes in 0.099s response time
$ echo $?
0
```

Worauf sich Schwellenwerte beziehen, richtet sich nach dem gewählten Plugin. Das Plugin `check_http` baut eine Verbindung via Hypertext Transfer Protocol (HTTP) zum Zielsystem auf und ermittelt neben der Antwortzeit auch die Größe der angeforderten Webseite, die Schwellenwerte für WARNING von »0.5« und CRITICAL von »1« beziehen sich aber ausschließlich auf die Antwortzeit.

Das HTTP-Protokoll ist weit komplexer und wird entsprechend von `check_http` unterstützt. Neben der Dokumentation des Plugins sind ebenfalls Kenntnisse des Protokolls von Nutzen. Um z. B. eine durch Transport Layer Security (TLS) abgesicherte Seite auf einem per IP-Adresse spezifizierten namensbasierten Webserver abzufragen, kommen weitere Optionen zum Einsatz:

```
$ ./check_http -H www.google.com.net -I 172.217.22.228 --ssl
HTTP OK: HTTP/1.1 200 OK - 13430 bytes in 0.101s response time
```

Eine aussagekräftige Hilfe, die zumeist mit `--help` auf der Kommandozeile aufgerufen wird, kennzeichnet gute Plugins.

Neben dem Exit Code gibt ein Check-Plugin über seinen **Plugin-Output** in für Menschen verständlicher Form Auskunft über den aktuellen Zustand. Darüber hinaus können zusätzlich vom Plugin ermittelte Werte angefügt sein. Diese Metriken unterscheiden sich je nach Plugin. Bei `check_http` sind es die Größe sowie die benötigte Zeit des Lesens der Webseite.

Eine weitere Eigenschaft von Check-Plugins, die bisher unterschlagen wurde, ist nicht nur das Ermitteln von Metriken, sondern auch deren Aufbereitung zu einer möglichen maschinellen Weiterverarbeitung. So gibt `check_http` zwar Metriken in seinem Output aus, in der erweiterten Ausgabe nach einem »|« werden diese Metriken aber z. B. mit den Schwellenwerten kombiniert. Die Gesamtheit dieser Metriken wird allgemein als **Performance-Daten** bezeichnet.

Jeder einzelne Satz von Performance-Daten identifiziert ein Label, hier *time* und *size*, gefolgt jeweils von einem Gleichheitszeichen. Danach stehen die vom Plugin gemessenen Werte, mittels Semikolon abgetrennt kommen an zweiter und dritter Position die Schwellenwerte. Abschließend können optional noch Minimal- und Maximalwert ausgegeben werden.

```
$ ./check_http -H www.google.com -w 0.5 -c 1
HTTP OK: HTTP/1.1 200 OK - 13527 bytes in 0.074s response time |\
    time=0.074161s;0.500000;1.000000;0.000000 \
    size=13527B;;;0
```

Neben **Remote-Check-Plugins**, die sich mithilfe eines Netzwerkprotokolls zu einem entfernten Rechner verbinden, existieren auch solche, die Ergebnisse durch einen lokalen Aufruf ermitteln. Erforderlich sind diese **Local-Check-Plugins**, um Werte zu ermitteln, die nur lokal zur Verfügung stehen, wie z. B.:

- CPU-Auslastung
- Hauptspeicherbenutzung
- Platzverbrauch auf Dateisystemen
- Laufende Prozesse
- ...

Solche Plugins müssen binärkompatibel zur Plattform sein, auf der sie aufgerufen werden, und unterscheiden sich je nach Betriebssystem auch in der Parametrisierung, aber mitunter auch in den zurückgelieferten Werten. Auf Unix-Systemen wird die CPU-Auslastung traditionell mit Load-Werten[3] angegeben, die in der Aussagekraft über einen auf Windows üblichen Prozentwert hinausgehen.

```
$ ./check_load -w 4,6,8 -c 6,8,10
WARNING - load average: 4.46, 5.02, 3.76
```

Im Gegensatz zu `check_http` werden hier für die drei Werte unterschiedliche Schwellenwerte berücksichtigt.

> Da Check-Plugins die gebräuchlichsten Plugins sind, wird von ihnen meistens in verkürzter Form einfach von Plugins gesprochen. Dabei wird sprachlich selten zwischen Remote und Local unterschieden.

[3] https://en.wikipedia.org/wiki/Load_(computing)

Notification-Plugins

Das Versenden von Benachrichtigungen deligiert Icinga ebenfalls an externe Programme, meistens Skripte. In der Regel kümmert sich jedes Notification-Plugin um einen Benachrichtigungsweg, üblich sind Mail- und SMS-Versand, aber es erfreuen sich inzwischen auch Anbindungen an Messaging-Systeme großer Beliebtheit.

Mit Optionen wird den Plugins übergeben, wer zu benachrichtigen ist und welche Informationen mitgeteilt werden sollen. Das Plugin bereitet die Informationen in der Regel entsprechend zum Übertragungsweg vor dem Versand noch optisch auf.

Damit ist Icinga, wie auch mit den Check-Plugins, einfach und schnell erweiterbar. Notification-Plugins sind im Allgemeinen noch viel leichter selbst zu schreiben als Check-Plugins, da erstere z. B. nur den erfolgten Versand mit einem RC von »0« zu bestätigen haben und nicht zwischen Schwellenwerten unterscheiden müssen. Die Programmlogik ist damit einfacher.

1.2.2 Icinga 2

Beim Projekt[4] Icinga 2 handelt es sich um die unverzichtbare Kernkomponente eines Icinga-Monitoring-Systems. Auf Basis der in den Konfigurationen definierten Überwachung steuert Icinga 2, wann etwas wie zu prüfen ist, wertet den Rückgabewert der entsprechenden Check-Plugins aus und entscheidet, wer im Fehlerfall wie zu benachrichtigen ist. Vereinfacht betrachtet handelt es sich bei Icinga 2 somit um einen »großen« Scheduler[5] zur Transaktionsverarbeitung.

Zusammen mit den Plugins erfüllt Icinga 2 die in Abschnitt 1.1 festgelegten Anforderungen an einen »Service-Monitor«.

Host, Service und andere Objekte

Die zu überwachenden Server und die auf ihnen laufenden Dienste werden Icinga 2 in Konfigurationsdateien bekannt gemacht. Neben diesen Hosts und Services werden auch Benachrichtigungen als Objekte definiert, im Icinga-Umfeld Notification genannt. Zu beachten ist, dass für Icinga auch alles, was mit Local-Check-Plugins überwacht werden soll, ein Service ist. Der Host selbst wird »nur« geprüft, ob er erreichbar ist. Standardmäßig wird so ein Hostalive-Check mittels Ping durchgeführt.

Objekte besitzen je nach Typ unterschiedliche Attribute, die ihre Eigenschaften definieren. Die eben angesprochene Referenzierung zu einem

[4] https://github.com/icinga/icinga2

[5] https://de.wikipedia.org/wiki/Historie_(Transaktionsverarbeitung)

Check Command und damit indirekt zu einem Plugin wird z. B. an einem Service über das Attribut *check_command* hergestellt. Darüber hinaus sind einige Objekttypen mittels sogenannter Custom Variables erweiterbar. Sie können frei gewählt werden und können so zusätzliche Informationen enthalten, wie den Hersteller eines überwachten Geräts oder zur Durchführung der Überwachung benötigte Passwörter.

All diese Objekte stehen in Beziehung zueinander, so gehört ein Service immer zu genau einem Host, eine Notification zu einem Service oder Host. Für jedes Plugin gibt es genau ein Check-Command-Objekt, das beschreibt, mit welchen Optionen das Plugin ausgeführt werden kann. Im Host und Service wird demnach ein Check Command referenziert, das die Überwachung steuert. Bei Benachrichtigungen übernimmt dies zwischen Notification und Plugin das Notification Command mit gleichen Eigenschaften.

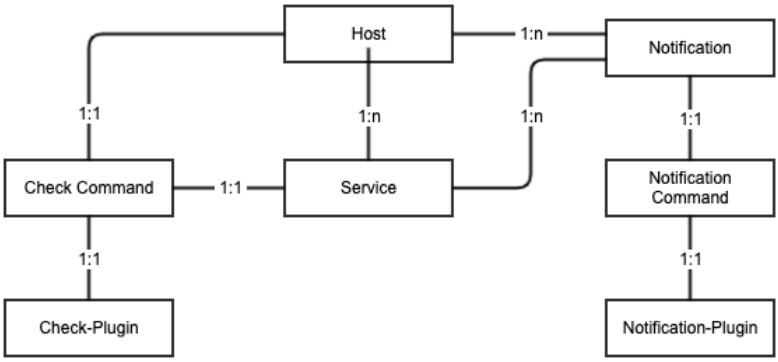


Abbildung 1-2: Zusammenhang zwischen Konfigurationsobjekten

Zusammen mit weiteren Sprachelementen, wie »if-else«-Conditions zur Ablaufsteuerung, Variablen und Funktionen, ist die Konfiguration eine eigene Sprache, die die Spezifikationen einer Domain Specific Language (DSL) erfüllt.

Features

Icinga 2 ist modular aufgebaut und kann durch Features im Funktionsumfang erweitert werden. So lassen sich Informationen von Icinga 2 zu anderen Systemen übertragen. Um beispielsweise einen Graphen für die zeitliche Entwicklung von Festplattenauslastungen zu visualisieren, liefert Icinga 2 Features zur Übertragung von Metriken an diverse nicht zum Icinga-Projekt gehörende Systeme. Die beliebtesten sind InfluxDB und Graphite, PNP4Nagios und openTSDB werden jedoch ebenfalls unterstützt.

Aber selbst die grundlegenden Eigenschaften sind in Features ausgelagert und können damit deaktiviert werden. Sollen keine Benachrichtigungen bearbeitet werden, wird das Feature *notification* abgeschaltet. Auch der Scheduler selbst ist im Feature *checker* implementiert.

Aktive und passive Checks

Werden Hosts und Services in regelmäßigen Abständen mit einem Check-Plugin geprüft, wird von einem aktiven Check gesprochen.

Icinga kennt jedoch auch passive Checks. Hierbei tut Icinga 2 nichts und wartet, dass ihm der Status oder Check-Result von außen mitgeteilt wird. Damit muss Icinga 2 eine Schnittstelle zur Einlieferung solcher Ergebnisse bereitstellen. Genau diese Aufgabe übernimmt das Feature *api*. Es stellt ein REST[6]-Application Programming Interface (API) bereit, das eine verschlüsselte Kommunikation von anderen Servern übers Netzwerk erlaubt.

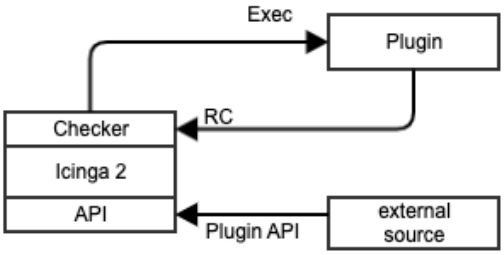


Abbildung 1-3: Icinga 2 – aktive und passive Checks

Bei passiven Checks ist Folgendes zu bedenken: Erfolgt keine regelmäßige Lieferung von Ergebnissen, sondern z. B. nur im Fehlerfall, kann nicht gewährleistet werden, dass der Kommunikationsweg noch funktioniert oder das sendende System noch läuft.

Host- und Servicestatus

Die Überwachung teilt sich wie oben beschrieben in Host und Service auf. Beide werden mit Plugins geprüft, zumindest bei aktiven Checks. Dabei kennt jedes Plugin vier Zustände, OK, WARNING, CRITICAL und UNKNOWN, die es an Icinga 2 übermittelt. Für passive Checks gilt die Beschränkung auf diese vier Zustände verständlicherweise ebenfalls, da Icinga 2 bei der weiteren Verarbeitung keinen Unterschied mehr zwischen aktiv und passiv macht.

[6] https://de.wikipedia.org/wiki/Representational_State_Transfer

Der vierte und letzte Status, den ein Plugin zurückliefern darf, ist UNKNOWN. Er besagt, dass das Plugin nicht in der Lage war, den abzufragenden Zustand zu ermitteln. Das passiert, wenn das Plugin in einen Timeout läuft, weil es den Zustand in der gesetzten Zeit nicht abfragen konnte. Ein weiterer Grund könnte sein, dass Icinga 2 das Plugin beim Aufruf nicht ausführen darf, da der Icinga-Benutzer nicht die Berechtigungen besitzt.

Bei einem Service erschließt sich die Einteilung in WARNING als Vorwarnstufe und CRITICAL für höchste Gefährdung der Produktion, aber was soll dies bei einem Host bedeuten?

Nun, Icinga unterscheidet bei einem Host auch nur zwischen UP und DOWN. Der Grund ist die Überlegung, dass ein Service zu einem Host, der ausgeschaltet bzw. nicht mehr erreichbar ist, mit an absoluter Sicherheit grenzender Wahrscheinlichkeit ebenfalls nicht mehr erreichbar ist. Deshalb muss der Hostalive-Check auch entsprechend hoch angesetzte Schwellenwerte haben, um sicherzustellen, dass der Host wirklich DOWN ist. Standardmäßig liefert ein Ping dann CRITICAL, wenn er Antwortzeiten von jenseits von 5 Sekunden misst oder eine Verlustrate von 100 Prozent bei fünf gesendeten Pings aufweist. Ein Schwellenwert für WARNING ist bei Hosts unerheblich, da Icinga dies als OK auffasst und den Host für UP erklärt.

Return Code	Servicestatus	Hoststatus
0	OK	UP
1	WARNING	UP
2	CRITICAL	DOWN
3	UNKNOWN	DOWN oder UNREACHABLE wird aus Abhängigkeiten berechnet

Tabelle 1-1: Plugin-Return-Code, Service- und Hoststatus

Die Tabelle fasst den Zusammenhang zwischen dem RC des Plugins zum Host- und Servicesatus nochmals zusammen. Dort findet sich auch, dass ein UNKNOWN an einem Host einem DOWN entspricht, UNREACHABLE ist hingegen ein Status, den Icinga berechnet. Ein Host ist nicht erreichbar, wenn Icinga ein Problem auf dem Weg vom Icinga-Server zum Zielhost bekannt ist, z. B. der Ausfall eines Routers. Icinga weiß damit, dass der Host nicht erreicht werden kann, und kann deshalb auch keinen Status ermitteln und »kennzeichnet« ihn als UNREACHABLE.