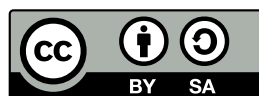


Icinga – Teil 1

Andreas Teuchert

11. Juli 2014



Icinga

- ▶ 2009 als Fork des Nagios-Cores entstanden
- ▶ Nagios-Hauptentwickler wollte Patches/Weiterentwicklungen nicht aufnehmen
- ▶ Nagios/Icinga sind der Industriestandard für Monitoring
- ▶ <http://www.nagios.com/users>

Icinga entstand 2009 als „Fork“ des Nagios-Core. Im Gegensatz zu der Entwicklung der Nagios-Plugins, an der eine Vielzahl an Entwicklern beteiligt ist, wird der Nagios-Core nur von einem Entwickler betreut. Die dadurch entstandenen langsamen Entwicklungsprozesse führt zur Unzufriedenheit einiger Entwickler.

Icinga hat das Ziel, ein Ersatz für den Nagios-Core zu sein, aber im Gegensatz zu Nagios für Beiträge Dritter offen zu sein.

Version 1 von Icinga ist mit Nagios kompatibel, sodass vorhandene Konfigurationen und Plugins weiterverwendet werden können. Version 2 wurde vollständig neu geschrieben und hat dadurch größere Unterschiede zu Icinga 1 bzw. Nagios. Zur Zeit ist Icinga 2 aber nur wenig verbreitet. Icinga 1 bzw. Nagios hingegen werden in nahezu jedem größeren Netzwerk eingesetzt.

Features

- ▶ Überwachung von Servern und darauf laufenden Diensten
- ▶ Performanzüberwachung
- ▶ Plugins
- ▶ Nagios Remote Plugin Executor (NRPE)
- ▶ Webinterface
- ▶ Benachrichtigung per E-Mail, SMS, etc.

Die Hauptaufgabe von Icinga ist die Überwachung von Servern und darauf laufenden Diensten. Weiterhin kann Icinga auch Statistiken über die Performanz der Dienste führen.

Über Plugins kann Icinga nahezu alles, von der Erreichbarkeit über die Systemlast bis zur Temperatur, überwachen.

Manche Eigenschaften eines Systems (z. B. Auslastung und Festplattenbelegung) können nicht direkt über das Netzwerk abgefragt werden. Für diese Fälle kann auf dem zu überwachenden System der Nagios Remote Plugin Executor (NRPE) installiert werden. Dieser Dienst erlaubt dem Icinga-Server das Ausführen von Checks auf entfernten Systemen.

Den aktuellen Status der überwachten Systeme und Dienste lässt sich über das Icinga-Webinterface abfragen. Über das Webinterface können auch diverse Einstellungen, wie das Abschalten von Benachrichtigungen für einzelne Dienste oder das Vor-Ankündigen von Wartungsarbeiten getätigt werden.

Bei Problemen kann Icinga vorher festgelegte Kontaktpersonen u. a. per E-Mail und SMS benachrichtigen.

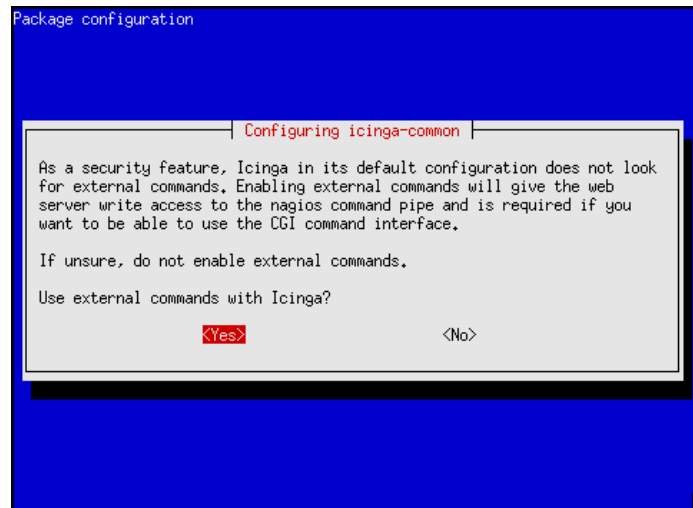
Installation

- ▶ unter Debian: Paket `icinga` installieren
- ▶ während der Installation können bereits Einstellungen getätigt werden

Unter Debian kann Icinga über das Paketsystem (Paket `icinga`) installiert werden. Während der Installation können bereits einige Einstellungen getätigt werden.

Möglicherweise ebenfalls interessant ist das Paket `icinga-doc`. Dieses enthält die Dokumentation zu Icinga, die nach der Installation über das Webinterface aufgerufen werden kann.

Installation: Externe Befehle



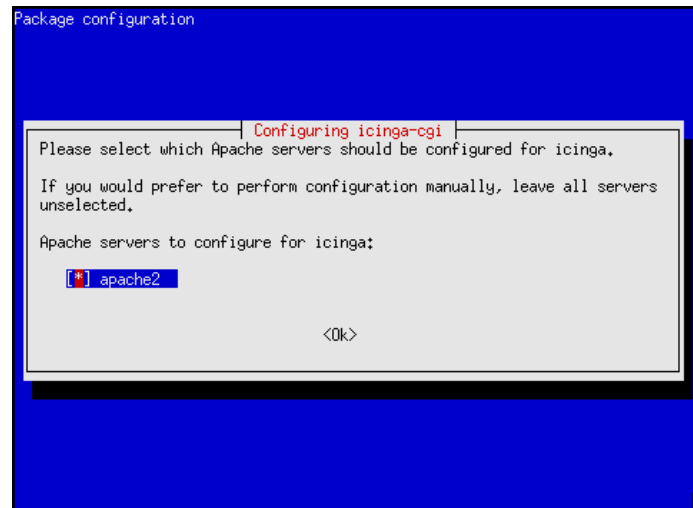
- ▶ bei aktivierten externen Befehlen kann Icinga über das Webinterface gesteuert werden

Externe Befehle erlauben es, Icinga über das Webinterface zu steuern und einige Einstellungen zu ändern. Dadurch können z. B. Benachrichtigungen deaktiviert oder Wartungsarbeiten eingetragen werden.

Da externe Befehle bei falscher Anwendung ein Sicherheitsrisiko darstellen können, sind sie standardmäßig nicht aktiv.

Nachträglich können externe Befehle mit Hilfe der Anweisungen in `/usr/share/doc/icinga/README.Debian` aktiviert werden.

Installation: Apache konfigurieren

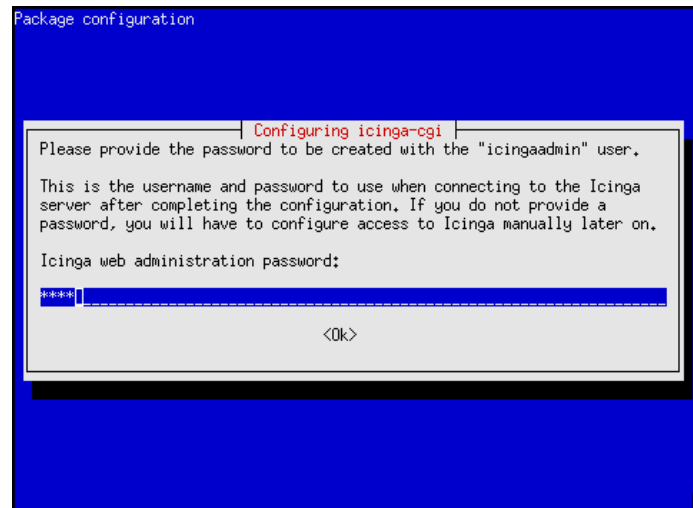


- ▶ ein installierter Apache-Server wird so konfiguriert, dass Icinga über `http://hostname/icinga` erreichbar ist

Ist auf dem System bereits ein Apache HTTP-Server installiert, kann er während der Icinga-Installation schon direkt so konfiguriert werden, dass Icinga über das Webinterface erreichbar ist.

Dies wird realisiert, indem ein Symlink von `/etc/apache2/conf.d/icinga.conf` nach `/etc/icinga/apache2.conf` angelegt wird. Soll die Konfiguration nach der Installation angepasst werden, kann die Datei `/etc/apache2/conf.d/icinga.conf` gelöscht oder editiert werden.

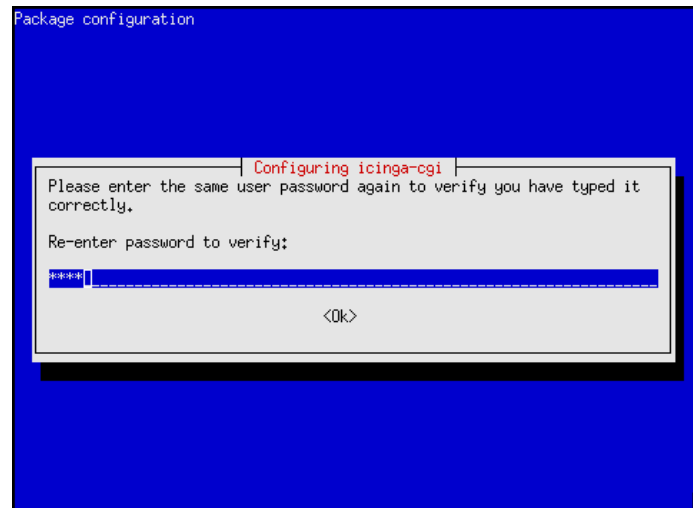
Installation: Passwort



- ▶ es kann ein Passwort für den `icingaadmin`-Benutzer des Webinterfaces festgelegt werden

Das Icinga-Webinterface ist passwortgeschützt. Bei der Installation kann ein Passwort für den Benutzer `icingaadmin` vergeben werden. Dieses kann nachträglich geändert werden, indem die Datei `/etc/icinga/htpasswd.users` mit `htpasswd` editiert wird.

Installation: Passwort bestätigen



- ▶ das Passwort nochmals bestätigen

Konfiguration

- ▶ nach der Installation: Webinterface unter `http://hostname/icinga`
- ▶ Benutzer: `icingaadmin`, während der Installation festgelegtes Passwort
- ▶ einige Dienste auf `localhost` werden überwacht
- ▶ Icinga-Konfiguration: `/etc/icinga/`

Nach der Installation ist das Icinga-Webinterface, sofern die Konfiguration eines Apache-Servers bei der Installation aktiviert wurde, unter `http://hostname/icinga` erreichbar. Zur Anmeldung wird der Benutzername `icingaadmin` und das während der Installation festgelegte Passwort benötigt.

Es werden auch bereits einige Dienste auf dem lokalen System überwacht.

Konfigurationsänderungen können durch Editieren der Dateien unter `/etc/icinga/` vorgenommen werden. Dort können auch weitere zu überwachende Rechner und Dienste eingetragen werden. Nach Konfigurationsänderungen oder dem Eintragen neuer Dienste muss der Icinga-Dienst neugestartet werden.

Lab: Icinga installieren

- ▶ Icinga über das Paket-System installieren
- ▶ im Web-Interface die vordefinierten Hosts und Services betrachten

Icinga-Konfiguration in `/etc/icinga/`

- ▶ `apache2.conf`: Konfiguration des Webservers
- ▶ `cgi.conf`: Konfiguration des Webinterfaces
- ▶ `htpasswd.users`: Benutzer und Passwörter für das Webinterface
- ▶ `icinga.conf`: Hauptkonfiguration
- ▶ `objects/`: Objektdefinitionen

In `/etc/icinga/` befinden sich einige Konfigurationsdateien.

In `apache2.conf` kann die Konfiguration des Apache-Servers bezüglich Icinga verändert werden. Dies ist nur möglich, wenn der Symlink in `/etc/apache2/conf.d/` bei der Installation angelegt und seitdem nicht gelöscht wurde.

In `cgi.conf` kann das Webinterface konfiguriert werden. Es kann z. B. ein anderer Benutzer als `icingaadmin` als Administrator festgelegt werden.

In `htpasswd.users` können Benutzer und ihre Passwörter eingetragen werden. Diese Benutzer können sich dann im Webinterface anmelden. Sie können allerdings nur Hosts und Services anzeigen, für die sie als Contacts eingetragen sind. Einzig die in `cgi.conf` unter `authorized_for_all_services` eingetragenen Benutzer dürfen den Status aller Hosts anzeigen. Die Datei `htpasswd.users` kann mit `htpasswd` bearbeitet werden.

In `icinga.conf` können Einstellungen für den Icinga-Core festgelegt werden. Für die meisten Installationen sind die Standard-Einstellungen ausreichend.

Im Verzeichnis `objects/` befinden sich Objekt-Konfigurationen. Dazu zählen z. B. Hosts und Services. Objekte werden im folgenden besprochen.

Objekte

- ▶ Hosts: zu überwachende Server
- ▶ Hostgroups: Gruppen von Servern
- ▶ Services: Dienste der Server(-Gruppen)
- ▶ Contacts: Kontaktdaten für Benachrichtigungen
- ▶ Contactgroups: Gruppen von Kontakten
- ▶ Beziehungen zwischen Objekten

Hosts, Hostgroups, Services, Contacts, Contactgroups und einige weitere Entitäten werden in der Icinga-Terminologie als Objekte bezeichnet.

Zwischen Objekten können Beziehungen bestehen, so können Hosts in Hostgroups gegliedert werden und einzelnen Hosts oder ganzen Hostgroups Services zugewiesen werden. Zur Festlegung von Benachrichtigungen können Hosts, Hostgruppen oder Services Contacts oder Contactgroups zugewiesen werden.

Die einzelnen Objekttypen werden im folgenden besprochen.

Hosts

```
1 define host {
2     use          generic-host
3     host_name    localhost
4     alias        Mein Rechner
5     address      127.0.0.1
6 }
```

- ▶ Host-Vorlage `generic-host` wird verwendet
- ▶ Name (z. B. für das Webinterface): `localhost`
- ▶ Alias wird zusätzlich zum Namen im Webinterface angezeigt
- ▶ Adresse: Adresse unter der der Rechner zu erreichen ist (IPv4, IPv6 oder DNS-Name)
- ▶ Erreichbarkeit wird per Ping überwacht

Hosts stellen zu überwachende Server dar. Einige Angaben sind zur Definition eines Hosts notwendig. Dazu gehört der `host_name`, der verwendet wird, um auf den Host zu verweisen (z. B. um ihm Services zuzuweisen) und auch im Webinterface angezeigt wird. Zusätzlich kann dem Host über `alias` eine zusätzliche Beschreibung zugewiesen werden. Diese wird ebenfalls im Webinterface angezeigt. Die `address` wird verwendet um den Host zu erreichen. Daher muss es sich um einen auflösbaren DNS-Namen oder um eine IPv4- oder IPv6-Adresse handeln. Es können noch eine Vielzahl an weiteren Einstellungen gesetzt werden. Durch die Zeile `use generic-host` werden diese auf sinnvolle Standardwerte gesetzt.

So definierte Hosts werden regelmäßig per ICMP-Ping überwacht.

Hostgroups

```
1 define hostgroup {
2   hostgroup_name  mail-server-fra
3   alias           Mail-Server Standort Frankfurt
4   members         mail-fra1, mail-fra2, mail-fra3
5 }
```

- ▶ Gruppe von Hosts (mit gemeinsamen Eigenschaften)
- ▶ Übersicht im Webinterface

Hostgroups stellen eine Gruppe von Hosts mit gemeinsamen Eigenschaften dar. Im Webinterface können diese dann übersichtlich dargestellt werden. Einer Hostgroup können über das Attribut `members` Hosts hinzugefügt werden. Durch `members *` können alle definierten Hosts dieser Hostgroup hinzugefügt werden.

Es ist möglich, Hostgroups Services zuzuweisen. Diese werden dann auf allen Hosts der Hostgruppe überwacht. Dadurch kann die Konfiguration sehr vereinfacht werden.

Services

```
1 define service {
2   hostgroup_name      http-server
3   # oder (nicht hostgroup_name und host_name):
4   host_name           www1
5   service_description HTTP
6   check_command       check_http
7   use                 generic-service
8 }
```

- ▶ zu überwachende Dienste auf einem Host oder einer Hostgroup
- ▶ `service_description` wird im Webinterface angezeigt
- ▶ `check_command` legt den Befehl fest mit dem der Dienst überprüft wird

Services stellen zu überwachende Dienste auf einem Host oder einer Hostgruppe dar. Dazu muss eine `hostgroup_name`- oder `host_name`-Zeile angegeben werden. Es darf aber nur eine der beiden Zeilen angegeben werden. Es können dann jeweils ein oder mehrere Hosts (bzw. Hostgroups) durch Komma getrennt angegeben werden.

Durch `service_description` wird eine Beschreibung dieses Dienstes angegeben. Diese wird im Webinterface angezeigt. Es muss auch ein `check_command` angegeben werden, welches festlegt, mit welchem Plugin der Dienst überwacht wird. Die installierten Plugins befinden sich üblicherweise unter `/usr/lib/nagios/plugins/`. Beispiele sind `check_http` und `check_ssh`.

Durch `use generic-service` werden auch hier die meisten Einstellungen auf sinnvolle Standardwerte gesetzt.

Contacts

```
1 define contact {
2   contact_name mustermann
3   alias          Max Mustermann
4   email          mustermann@example.com
5 }
```

- ▶ Kontaktdaten zur Benachrichtigung bei Problemen
- ▶ auch als Benutzername für das Webinterface wenn ein Passwort in `htpasswd.users` hinterlegt wurde.

Contacts repräsentieren Personen die bei Problemen benachrichtigt werden. Dazu sollte eine E-Mail-Adresse angegeben werden.

Außerdem dient der `contact_name` als Benutzername im Webinterface, falls ein Passwort in `htpasswd.users` hinterlegt wurde.

Contactgroups

```
1 define contactgroup {
2   contactgroup_name admin
3   alias             Administratoren
4   members           mustermann, musterfrau, support
5 }
```

- ▶ Gruppe von Contacts
- ▶ vereinfacht das Festlegen von Benachrichtigungen

Contacts – Beispiel

```
1 define host {
2   use          generic-host
3   host_name    mail-srv1
4   alias       Mail-Server 1
5   address     2001:db8:ce67:e48a::1
6   contacts    mustermann
7 }
```

```
1 define service {
2   use          generic-service
3   host_name    ww1
4   contact_groups    web-admins
5   service_description    HTTP
6   check_command    check_http
7 }
```

In diesem Beispiel wurde für den Host `mail-srv1` der Contact `mustermann` für Benachrichtigungen eingetragen.

Bei Problemen mit dem HTTP-Server auf `ww1` wird die Contactgroup `web-admins` benachrichtigt.

Lab: Weitere Checks einrichten

- ▶ Webserver und SSH-Server auf eigenem Rechner überwachen
- ▶ Partner-Rechner als Host eintragen
- ▶ SSH-Dienst auf Partner-Rechner überwachen