

Icinga – Teil 1

Linux-Kurs der Unix-AG

Andreas Teuchert

11. Juli 2014



TU Kaiserslautern



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>

Features

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

Installation

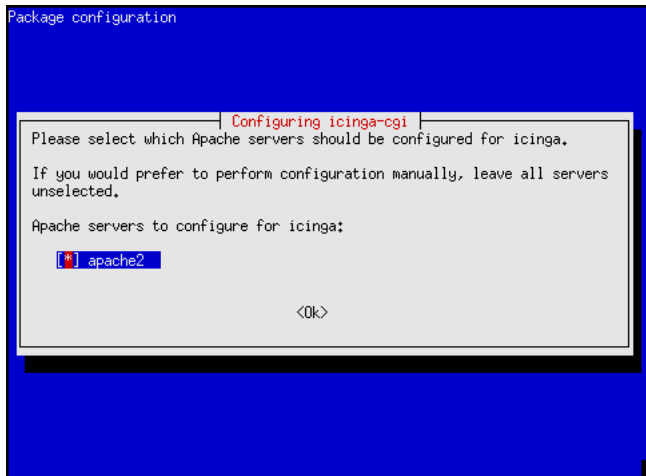
- ▶ unter Debian: Paket `icinga` installieren
- ▶ während der Installation können bereits Einstellungen getätigt werden

Installation: Externe Befehle



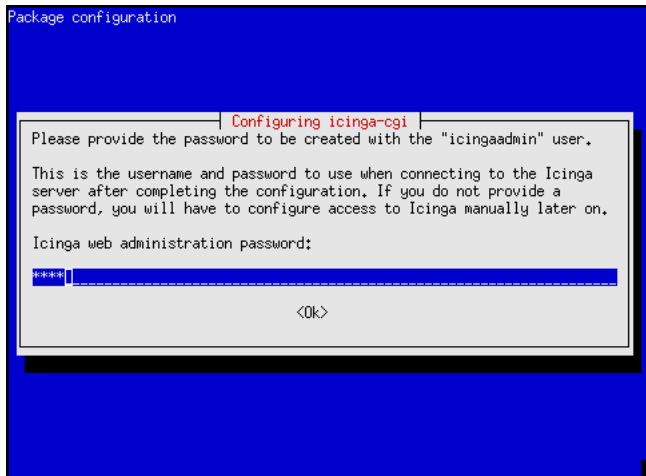
- bei aktivierten externen Befehlen kann Icinga über das Webinterface gesteuert werden

Installation: Apache konfigurieren



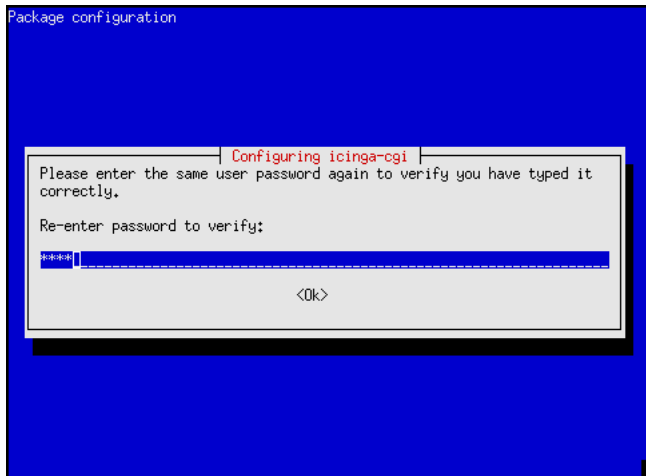
- ▶ ein installierter Apache-Server wird so konfiguriert, dass Icinga über `http://hostname/icinga` erreichbar ist

Installation: Passwort



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

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/`

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

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

```
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

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

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

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.

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    www1
4     contact_groups    web-admins
5     service_description HTTP
6     check_command    check_http
7 }
```

Lab: Weitere Checks einrichten

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