

# Icinga 2 – Einführung und Übersicht

Jens Link

September 2015

# Einführung

# Inhalte

- ▶ Nagios, Icinga, Icinga2 - Eine Übersicht
- ▶ Monitoring allgemein
- ▶ Konfiguration Icinga2
- ▶ Migration Nagios / Icinga -> Icinga2
- ▶ Icinga2 und Puppet

# Über mich

- ▶ Freelancer
- ▶ Linux seit Lehmanns das auf 35 Disketten verkauft hat
- ▶ IPv6 seit > 2400 Tagen auf dem eigenen Server
- ▶ derzeit drei Icinga Installationen für Kunden
- ▶ ansonsten: Netzwerk (Cisco + Juniper) und DNS und DNSSEC
- ▶ eMail/Jabber: [jenslink@quux.de](mailto:jenslink@quux.de)

## Aktuelles Projekt (eines davon)

- ▶ Verteiltes Monitoring für Hotelnetze
- ▶ Ansatz: Zentraler Server + Probes in den einzelnen Hotels
- ▶ Konfigmanagement mit Puppet
- ▶ Web-GUI für den Kunden (Useradministration, erstellen neuer Sites)
- ▶ Überwiegend ping + SNMP

# Nagios, Icinga, Icinga2

- ▶ 1999 - Ethan Galstad entwickelt NetSaint
- ▶ 2002 - Umbenennung zu Nagios wegen Trademark Problemen
- ▶ 2009 - Icinga als Fork von Nagios
- ▶ 2014 - Icinga2 komplette Neuentwicklung

# Warum der Fork

- ▶ Die Entwicklung von Nagios schien zugunsten einer kommerziellen Variante eingeschlafen zu sein
- ▶ Patches und Erweiterungen wurden nur selten integriert

# Erweiterungen

- ▶ pnp4nagios
- ▶ graphite
- ▶ lconf
- ▶ Icinga Reporting
- ▶ ...

# Icinga Web Frontends

Icinga1 / Icinga2 sind nur der Monitoring Core ohne GUI

Als GUI gibt es:

- ▶ Icinga Classic UI - Sehr ähnlich der Nagios GUI
- ▶ (Icinga web)
- ▶ Icingaweb2
- ▶ Thruk

# Monitoring allgemein (IMHO)

- ▶ Lieber zu viel als zu wenig, man kann später immer noch tunen
- ▶ Einen Service den man nicht überwacht braucht man auch nicht
- ▶ Auch Infrastruktur (Netzwerk, Hardware, Virtualisierung, ...) muss überwacht werden
- ▶ Abhängigkeiten beachten
- ▶ Automatisierung macht das Leben leichter
- ▶ Aus Fehlern lernen
- ▶ Monitoring überwachen

# Icinga2

- ▶ C++ und Boost
- ▶ \*NIX und Windows
- ▶ anderes Format für Configs
- ▶ Modular
- ▶ Clusterfunktionen eingebaut
- ▶ MySQL, PostgreSQL und Livestatus sind eingebaut
- ▶ bessere Performance
- ▶ CLI für Configänderungen
- ▶ Weiterhin: Monitoring über Plugins

# Icinga2 Komponenten

- ▶ Checker
- ▶ Cluster
- ▶ API
- ▶ Perfdata
- ▶ Livestatus
- ▶ Graphite
- ▶ Notification
- ▶ ido\_(mysql|postgres)

# Icinga2 - wesentliche Änderungen

- ▶ Syntax
- ▶ Globale Macros
- ▶ frei definierbare Makros für Kommandos

# Icinga2 Konfiguration

# Installation Icinga 2

- ▶ Komplette anders als Nagios / Icinga
- ▶ Fertige Pakete (für Debian: [debmon.org](http://debmon.org))
- ▶ oder aus den Sourcen

# Plugins

- ▶ Überwachung wie bei Nagios/Icinga auch über Plugins
- ▶ für viele Aufgaben lassen welche im Internet finden
- ▶ selber schreiben muss nicht schwer sein

# Host-Konfiguration

```
object Host "www.example.com" {  
  
    /* Import the default host template defined in `templates.conf`. */  
    import "generic-host"  
  
    /* Specify the address attributes for checks e.g. `ssh` or `http`. */  
    address = 192.0.2.1  
    address6 = 2001:db8:aaaa::1  
  
    vars.os = "Linux"  
    vars.notification["mail"] = {  
  
        /* The UserGroup `icingaadmins` is defined in `users.conf`. */  
        groups = [ "icingaadmins" ]  
    }  
}
```

# Service-Konfiguration (I)

```
apply Service "ping4" {  
    import "generic-service"  
    check_command = "ping4"  
    assign where host.address  
}
```

## Service-Konfiguration (II)

```
apply Service "ssh" {
  import "generic-service"
  check_command = "ssh"
  assign where (host.address || host.address6)
    && host.vars.os == "Linux"
  ignore where host.name == "localhost"
    /* for upgrade safety */
}
```

# Check-Commands (I)

```
template CheckCommand "ping-common" {
    import "plugin-check-command"
    command = [ PluginDir + "/check_ping" ]
    arguments = {
        "-H" = "$ping_address$"
        "-w" = "$ping_wrta$, $ping_wpl%"
        "-c" = "$ping_crta$, $ping_cpl%"
        "-p" = "$ping_packets$"
        "-t" = "$ping_timeout$"
    }

    vars.ping_wrta = 100
    vars.ping_wpl = 5
    vars.ping_crta = 200
    vars.ping_cpl = 15
}
```

# Agent based checks

- ▶ Icinga 2 Client
- ▶ SSH
- ▶ SNMP
- ▶ NRPE
- ▶ NSClient++ (Windows)

# Icinga2 als Client

- ▶ Icinga2 master / client setup über die CLI
- ▶ eigentlich ein normales Icinga, geht auch unter Windows
- ▶ Eigene PKI und passende “Wizards”, die die Arbeit damit erleichtern
- ▶ Konfiguration kann vom Master aus erfolgen
- ▶ Ein Client kann auch Clients haben
- ▶ Auf dem Master unter `/etc/icinga2/repository.d/`

# Icinga2 Multiple Sites

- ▶ icinga2 node wizard auf Master und Client -> Fragen beantworten
- ▶ icinga2 node update-config -> zieht Infos vom Server

# Migration

- ▶ Per Script <https://github.com/Icinga/icinga2-migration>, aber das Ergebnis ist nicht wirklich “schön”
- ▶ Von Hand - Viel Arbeit, aber man kann auch gleichzeitig aufräumen ;-)

## Icinga2 und Puppet

- ▶ Puppet Modul direkt von den Entwicklern:  
<https://github.com/Icinga/puppet-icinga2>
- ▶ Ansatz: Wenn ihr etwas installiert, konfiguriert gleich den passenden Icinga2 Check mit
- ▶ Stichwort: Exported Resources

```
@icinga2::object::host { $::fqdn:
    display_name => $fqdn,
    ipv4_address => $::ipaddress_eth0,
    vars => { os    => 'linux'
    },

    target_dir      => '/etc/icinga2/objects/hosts',
    target_file_name => "${fqdn}.conf",
    target_file_ensure => present,
    target_file_owner  => 'root',
    target_file_group  => 'root',
    target_file_mode   => '0644'
}
```

# Ende

Demo und Fragen

# Idee

Monitoring Platform für Projekte.

Interesse?

<http://lists.quux.de/listinfo/monitoring>