

Verteiltes Monitoring am GridKa

Axel Jäger
Steinbuch Centre for Computing



Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft



Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825



- **Karlsruhe Institute of Technology (KIT)**
 - Fusion von Universität Karlsruhe und Forschungszentrum Karlsruhe
 - Zur Zeit aus rechtlichen Gründen noch eine „Kooperation“
 - Endgültige Fusion zur Forschungsuniversität ist in 2009 geplant

- **Steinbuch Centre for Computing**
 - Das Rechenzentrum des KIT als Ergebnis der Fusion der ehemaligen Rechenzentren von Universität KA und Forschungszentrum KA

- **GridKa**
 - Das deutsche Tier-1 Zentrum für die Hochenergiephysik betrieben durch das SCC

- **Axel Jäger**
 - Am GridKa mitverantwortlich für Installation, Betrieb und Überwachung

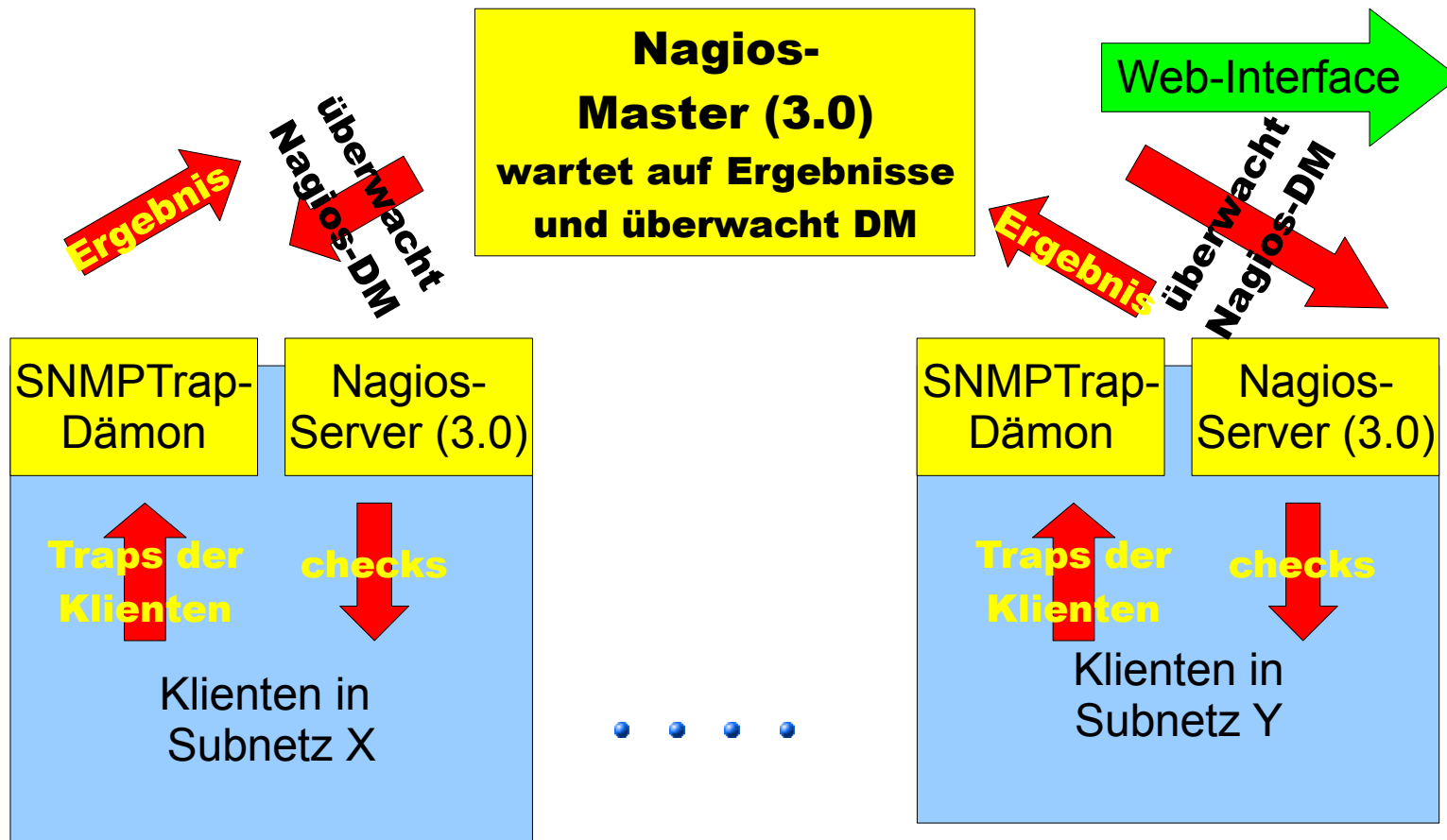
Was wird überwacht ?

- **Das GridKa besteht zur Zeit aus**
 - ca. 6700 CPU-Cores
 - ca. 6 PB = 6000 TB Online-Speicher (netto)

- **Stand heute : ~1900 Hosts mit ~18000 Services**
 - 66 unterschiedliche Service-Definitionen
 - Organisiert in 85 Hostgruppen
 - 53 DM, 1 Master und 1 MySQL-Server für ndo-Datenbank

- **Softwarestand**
 - Nagios 3.0 (13.3.2008)
 - nsca 2.7.2
 - Ndoutils 1.4 b7 (MySQL-Server 4.1)
 - PNP 0.4.3
 - Business Process View 0.9.1

Konzept des verteilten Monitorings



■ Vorteile

- **Höhere Performance der Checks**
- **Checks können öfter ausgeführt werden**
- **Sehr gute Skalierung auch bei großen Installationen**

■ Nachteile

- **Konfiguration wird unübersichtlicher**
 - **Jeder Host/Service muss am Master und am DM unterschiedlich definiert werden (aktiv/passiv)**
 - **Jeder Host/Service muss am Master bekannt sein**
 - **Die Definitionen müssen aufeinander abgestimmt sein**
- **Überwachung muss überwacht werden**

Distributed Monitor am GridKa

- Führt aktive Checks in seinem Subnetz aus und sendet die Ergebnisse mit `send_nasca` an den Master
- Kein selbständiger Nagios-Web-Server
- Weitere Aufgaben dieses Rechners
 - Installations-Server für Subnetz (Rocks-Toolkit)
 - DHCP- und Time-Server sowie Log-Host
 - SNMP-Traphost (SNMPTRAPD)
 - Traphandler schickt über `send_nasca` an Nagios-Master
 - Datensammler für Ganglia
 - etc.

Bisher (Nagios 2.x)

- Eingabedatei für jeden Distributed Monitor (DM) in selbstdefinierten Format
- Skript erzeugt aus diesen Dateien die Nagios-Konfiguration für Nagios 1.x
- Vorteile
 - Schnelle Erweiterung der Gesamtkonfiguration
 - Zusammenhänge zwischen DM und Master im Skript verdrahtet
- Nachteile
 - Unflexibel
 - Neuere Konfigurationsmöglichkeiten konnten nicht genutzt werden
 - Was Skript nicht kennt – kann nicht eingesetzt werden

Ziel

- **Flexible Konfiguration**
- **Weiterhin schnelle Erweiterung der Gesamtkonfiguration**
- **Möglichst keine doppelten Definitionen für Master und DM**
- **Nutzung auch zukünftiger Konfigurationserweiterungen ermöglichen**

Realisierung

- **Zentrale Verwaltung aller Dateien am Master**
- **Eigenes Verzeichnis für jeden DM**
- **Verzeichnis „COMMON“ für alle**
- **Verzeichnis „MASTER“ nur für den Master**
- **Definitionen für DM werden auch am Master eingebunden**
- **Nutzung der „originalen“ Nagios-Konfigurationsdateien**
 - **Keine selbstdefinierten Formate in den Konfigurationsdateien**
- **Nutzung von Templates**
 - **Unterschiede zwischen DM und Master nur in den Templates**
- **Zuordnung der Services zu den Hosts über die Hostgruppen**

Starten des Nagios am DM

■ Problem

- Konfigurationsdateien werden zentral verwaltet
- Vor Reload/Restart müssen die aktuellen Konfigurationsdateien dem DM zur Verfügung stehen
- Bei Konfigurationsänderung nur die betroffenen DM durchstarten

■ Lösung

- Anpassung des Nagios-Startskriptes für DM (neuer Befehl : copy)
- Vor Konfigurationscheck werden alle notwendigen Dateien vom Master auf den DM kopiert (mit scp)
- Dabei wird auch das Verzeichnis „libexec“ kopiert
- Nach erfolgter Kopie wird Datei „copied“ am Master aktualisiert
- Nach erfolgreichem Restart/Reload wird Datei „restarted“ am Master aktualisiert
- Dadurch kann der Master entscheiden ob ein DM auf aktuellem Stand ist – oder nicht

Konfigurationsdateien (1)

■ Dateien im DM-spezifischen Verzeichnis

■ Dateien die auch am Master eingebunden werden müssen

■ myhosts.cfg

- Diese Datei enthält alle Host-Definitionen für die dieser DM zuständig sein soll
- Inklusive Parentdefinition und Hostgruppenzugehörigkeit

■ myservices.cfg

- Diese Datei enthält alle Service-Definitionen, die nicht über Hostgruppen zugeordnet werden.
- Enthält bei mir üblicherweise keine Definitionen

Konfigurationsdateien (2)

■ Dateien im DM-spezifischen Verzeichnis

■ Dateien die nur am DM eingebunden werden

■ localhost.cfg

- Enthält alle Host-Definitionen, die nur am DM bekannt sein dürfen.
- Insbesondere den Parent des DM um eine gültige Parentbeziehung für den DM in der Datei myhosts.cfg definieren zu können
- Definition erfolgt so, dass keinerlei Überwachung für diese Hosts von diesem DM ausgeführt wird.

■ localservices.cfg

- Enthält Service-Definitionen, die nur am DM bekannt sein dürfen
- Bis jetzt keine Notwendigkeit für diese Datei gefunden

■ nagios.cfg

- Enthält alle notwendigen Includes für Hostgruppen und Templates

Konfigurationsdateien (3)

■ Dateien im DM-spezifischen Verzeichnis

■ Dateien für Restart/Reload des DM

■ Copied

- Enthält den Zeitpunkt des letzten Kopierens der Konfiguration

■ Restarted

- Enthält den Zeitpunkt des letzten erfolgreichen Reloads oder Restarts des Nagios-Prozesses am DM

■ Servername

- Enthält den Namen des Servers der diese Konfigurationsdateien benutzen darf

- Eigentlich keine Nagios-Dateien – sind auch zum Funktionieren der Überwachung nicht notwendig - „nur“ zum komfortablen Reload des Nagios-Systems

Konfigurationsdateien (4)

- Dateien im Verzeichnis „COMMON“
 - Dateien für DM und Master
 - **templates.cfg**
 - Enthält die Standard-Templates
 - **timeperiods.cfg**
 - Enthält die Timeperiod-Definitionen
 - **commands.cfg**
 - Enthält die Definitionen für die Kommandos
 - **<hostgruppen_definitionen>.cfg**
 - Enthält die Hostgruppendefinition und deren zugeordneten Service
 - **u.ä.**

Konfigurationsdateien (5)

- Dateien im Verzeichnis „COMMON“
 - Dateien die nur von den DM eingebunden werden
 - `distrib_templates.cfg`
 - Enthält alle DM-spezifischen Templates
 - Aktive Host- und Service-Templates
 - Bezieht sich unter anderem auf die Datei `templates.cfg`

Konfigurationsdateien (6)

■ Dateien im Verzeichnis Master

■ Dateien werden nur vom Master eingebunden

■ master_templates.cfg

- Enthält alle Masterspezifischen Template-Definitionen
- Passive Host- und Service-Templates
- Bezieht sich unter anderem auf die Datei templates.cfg

■ master_services.cfg

- Services, die der Master ausführen muss (z.B. Überwachung der DM)

■ <hostgruppen_definition>.cfg

- Enthält die Zuordnung von Service zu Hostgruppen für Services die nur am Master bekannt sein dürfen (Traps)

■ u.ä.

Was bedeutet das alles ?

- Jede Definition ist nur einmal vorhanden
 - Ausnahme : Parent des DM
- Durch Verwendung der Konfigurationsdateien sowohl für den DM als auch für den Master, sind die Definitionen überall bekannt.
- Einziger Unterschied zwischen DM und Master :
 - `distrib_templates.cfg`
 - `master_templates.cfg`
 - Bei Definitionen in diesen Dateien muss natürlich auf Kompatibilität geachtet werden.
- Durch Zuordnung der Services zu den Hostgruppen verringert sich die Schreiarbeit
- Viele Templates bzw. die Templatehierarchie kann ebenfalls unübersichtlich werden (mehr später)

Beispiel 1

myhosts.cfg (Verzeichnis C01-001)

```
define host{
    use                linux-server
    host_name          c01-001-137
    alias              Compute Node 37 C01-001
    address            10.1.1.137
    hostgroups         Compute-Node,Compute-Node-BMC,Rack_C01-001
    parents            mc01-001-200
}
```

compute-node.cfg (Verzeichnis COMMON – eingebunden in DM und Master)

```
define hostgroup{
    hostgroup_name    Compute-Node
    alias             Compute-Node-Clients
}

define service{
    use                service_ping
    hostgroup_name    Compute-Node
    service_description PING
}
```

Beispiel 2

myhosts.cfg (Verzeichnis A01-004)

```
define host{
    use                linux-server
    host_name          a01-004-169
    address            10.97.4.169
    hostgroups         Admin-Node,Rack_A01-004,Backup-Server
    contact_groups     mwadmin,linux-admins,lfc-admins
    parents            ma01-004-200
}
```

backup_server.cfg (Verzeichnis COMMON – eingebunden in DM und Master)

```
define hostgroup{
    hostgroup_name     Backup-Server ; The name of the hostgroup
    alias              Server with TSM-Backup ; Long name of the group
}
```

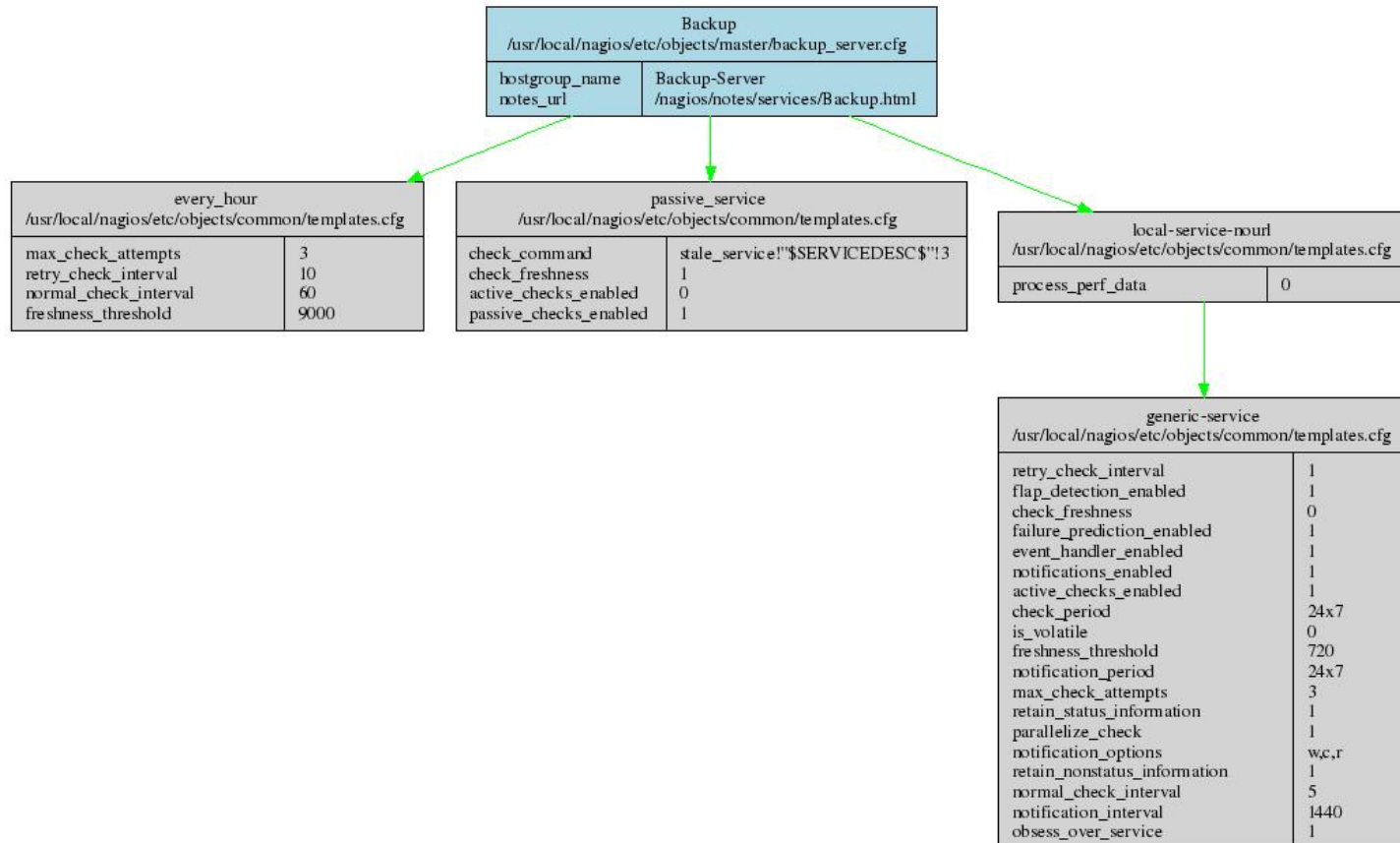
backup_server.cfg (Verzeichnis MASTER – eingebunden nur am Master)

```
define service{
    use                every_hour,passive_service,local-service-nourl
    notes_url          /nagios/notes/services/Backup.html
    hostgroup_name     Backup-Server
    service_description Backup
}
```

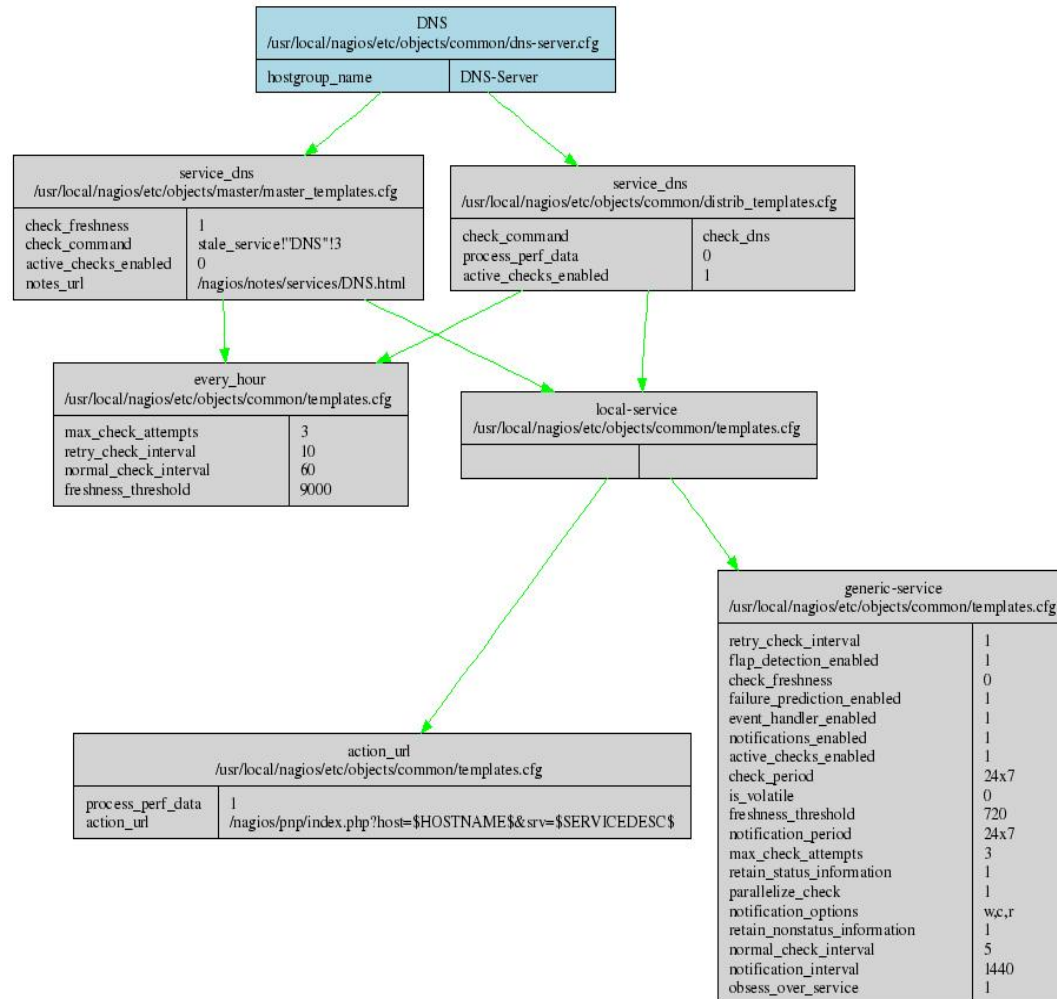
Template-Hierarchie

- Ziel : grafische Darstellung der Abhängigkeiten einzelner Services oder Hosts von benutzten Templates
- Skript durchsucht ab `nagios.cfg` für Master
- Zusätzlich noch die `distrib_templates.cfg` aus Verzeichnis `COMMON`
- Erkennt und durchsucht weitere eingebundene Konfigurationsdateien (`cfg_file=...`)
- Erstellt Grafik für Host- und Service-Definitionen mit den benutzten Templates unter Verwendung von `dot` (`graphviz`)

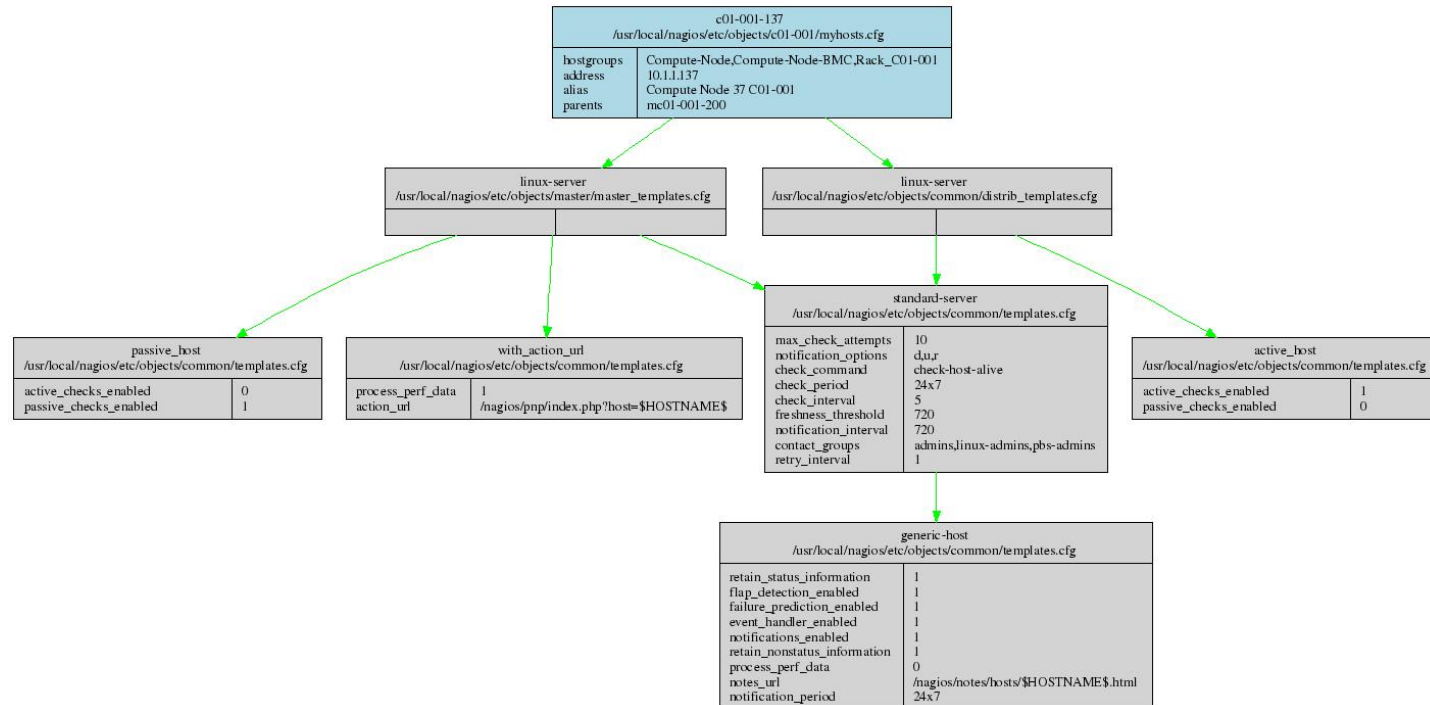
Template-Hierarchie



Template-Hierarchie



Template-Hierarchie (Host)



- **Fragen ?**
- **Anregungen ?**
- **Live-Demo ?**