

www.immobilienscout24.de



Systemmanagement mit RPM und YADT

Eine Lösung für Rechenzentren

Chemnitz | 2012-03-18 | Ralph Angenendt
Application Manager



License: <http://creativecommons.org/licenses/by-nc-nd/3.0/>

Worum geht es?

➔ RPM

- ➔ Bekanntes Paketformat
- ➔ Einfache Nutzung (und Paketierung)
- ➔ Eingebaute Datenverifikation
- ➔ Es gibt ein komplettes Werkzeugset (yum, rpm, etc.)

Worum geht es?

➔ YADT

- ➔ Ein „Augmented Deployment Tool“
- ➔ Zentrales Management von Abhängigkeiten zwischen
 - ➔ Services
 - ➔ Systemen
 - ➔ Softwarepaketen

RPM, huh?

- Sicherlich. Alles andere ist auch schon RPM
 - ➔ Unser eingesetztes OS (RHEL – 100% RPM)
 - ➔ Software die „von außen“ kommt (z.B. EPEL)
 - ➔ Unsere Applikationen
 - ➔ <Nelson>Ha Ha</Nelson>
 - ➔ Aber wir sind auf dem besten Weg dahin

Aber für Konfigurationsmanagement?

- Konfiguration sind auch nur Dateien
- RPM kann gut mit Dateien umgehen
- Es gibt schon Tools, um RPMs auf Rechnern zu installieren
- RPM kann den Inhalt von Paketen verifizieren
- Updates sind einfach

Ihr baut also RPMs für alle eure Rechner?

Um. Nein.

Ihr baut also RPMs für alle eure Rechner?

**Okay,
doch.
Irgendwie.**

Ihr baut also RPMS für alle eure Rechner?

**Wir lassen
bauen.
Automatisch.**

„Config Subversion“

- Die komplette Konfiguration befindet sich in einem Subversion-Repository
 - ➔ Hierarchisch
 - ➔ Unterstützt ein „Data Center“-Layout
 - ➔ Ist leicht verständlich
 - ➔ In einem typischen „Unix“-Layout

„Config Subversion“

- ➔ Von generisch nach spezifisch
- ➔ On-Commit
 - ➔ RPM building
 - ➔ YUM repository generation
- ➔ Also works with dpkg and apt
 - ➔ If you write the code to support it

So sieht's aus

Überschreibt



```
all/  
loc/  
type/  
loctype/  
host/
```

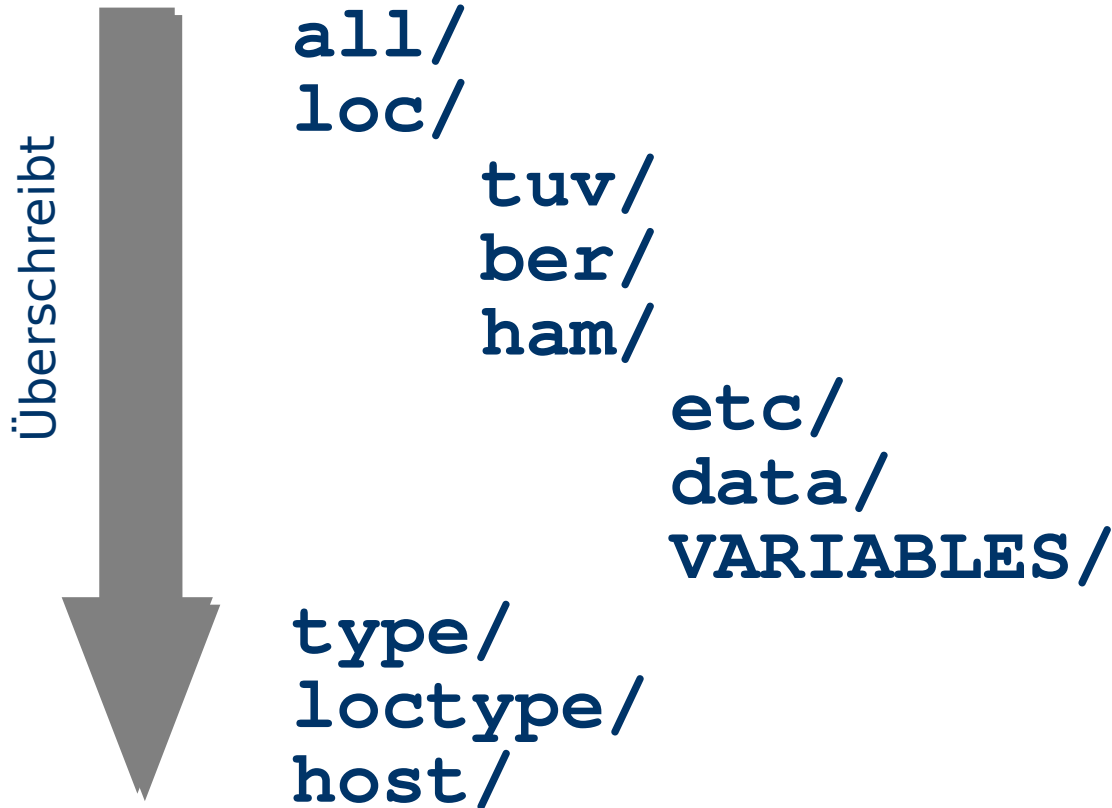
So sieht's aus

Überschreibt

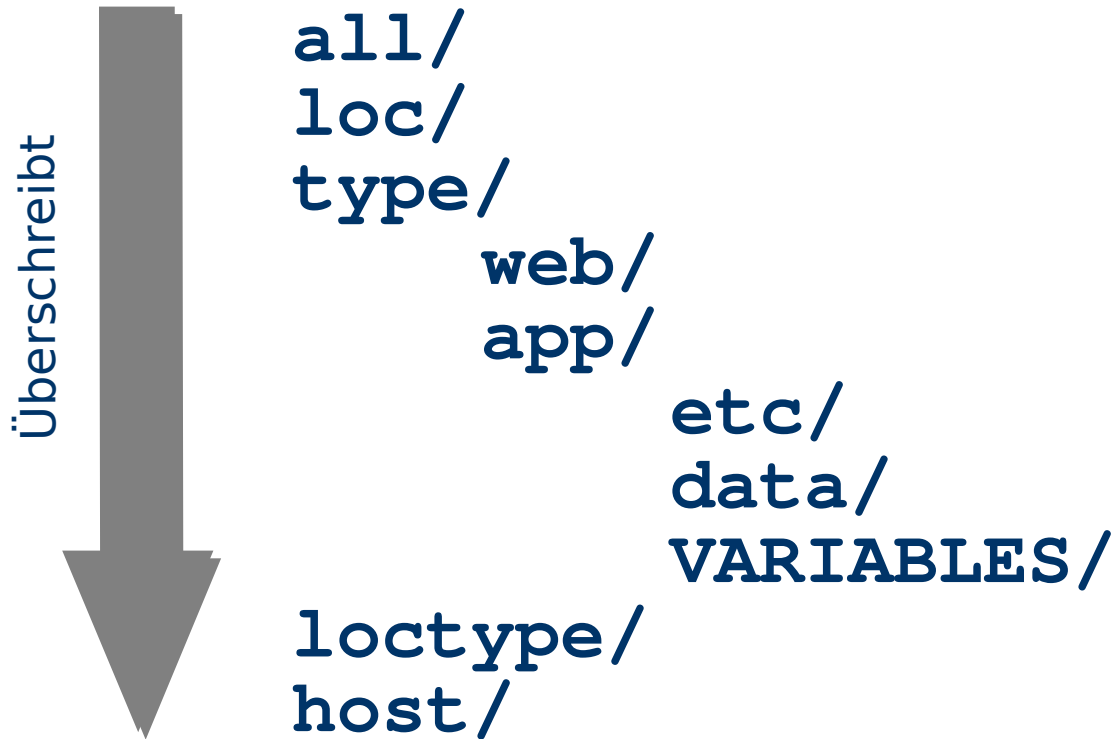


```
all/  
    etc/  
    data/  
    VARIABLES/  
loc/  
type/  
loctype/  
host/
```

So sieht's aus



So sieht's aus



So sieht's aus



So sieht's aus

Überschreibt



```
all/  
loc/  
type/  
loctype/  
host/  
    berweb01/  
    berweb02/  
        etc/  
        data/  
        VARIABLES/
```


VARIABLES?

- VARIABLES/ beinhaltet – ach - Variablen
 - ➔ Viele Hosts haben eine ähnliche Konfiguration
 - ➔ Den Host am besten generell konfigurieren
 - ➔ Alle Hosts benutzen einen Proxy
 - ➔ Proxies in tuv, ber und in ham sind verschieden

Variablen

```
all/etc/proxy.conf:  
  [...]
  proxy_port=3128
  proxy_host=@@@PROXY_HOST@@@
```

```
loc/tuv/VARIABLES/PROXY_HOST:  
  tuvprx.example.com
```

```
loc/ber/VARIABLES/PROXY_HOST:  
  berprx.example.com
```

```
loc/ham/VARIABLE/PROXY_HOST:  
  hamprx.example.com
```

Mehr Spezialitäten

- ➔ Es gibt zwei „Spezialvariablen“
 - ➔ `RPM_PROVIDES`
 - ➔ `config-hostname` (e.g. `config-berweb01`)
 - ➔ `RPM_REQUIRES`
 - ➔ `tomcat, httpd, java-application`
- ➔ `RPM_PROVIDES` wird von Kickstart „required“
- ➔ Inhalt von `RPM_REQUIRES` zieht alle anderen benötigten Pakete für den Host nach

All zusammen jetzt

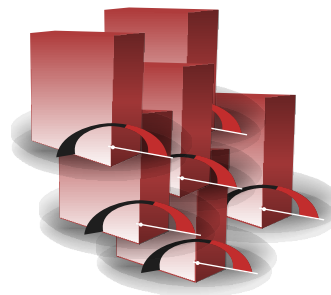
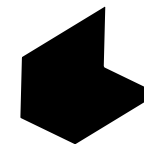
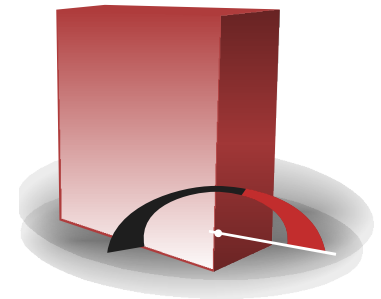


config-rpm-maker

ersetzt
VARIABLES

baut
RPMs

erstellt
YUM-Repository



All zusammen jetzt

➔ yadt-config-rpm-maker

- ➔ Wird als post-commit-hook in SVN eingebunden
- ➔ In python geschrieben
- ➔ Kann mehrere Pakete parallel erstellen
- ➔ Erkennt, welche Pakete nach einer Änderung neu gebaut werden müssen
- ➔ Baut nur das minimal benötigte Set
- ➔ Ist Open Source (GPL)
- ➔ Erhältlich hier: <https://code.google.com/p/yadt/>

Probleme

- ➔ RPM mag ein paar Dinge nicht
 - ➔ z.B. 2 Pakete denen die gleiche Datei gehört
 - ➔ config.d/ wird auch nicht überall benutzt
 - ➔ Generische Konfiguration ist meistens genau das
 - ➔ Das kann eine Installation verhindern

Probleme

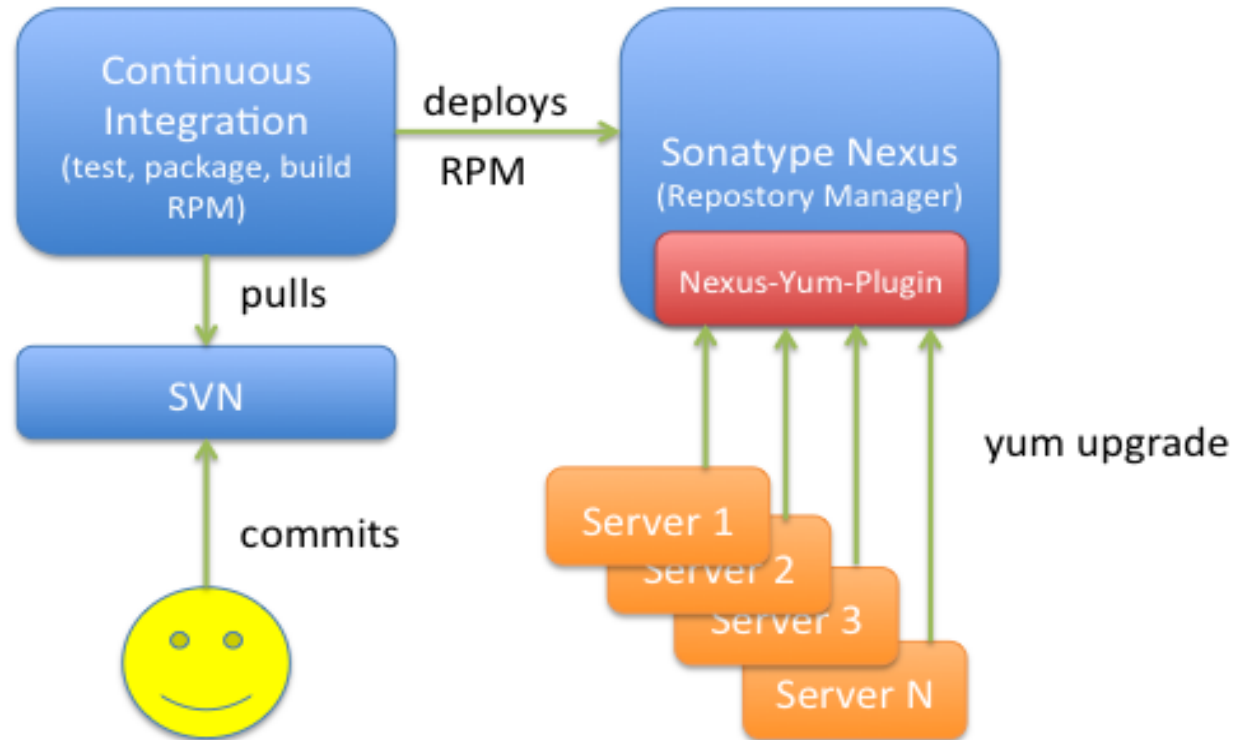
➔ Lösung

- ➔ „Wrapper-Pakete“
- ➔ Diese überschreiben Konfiguration via `%post`
- ➔ Selbstgeschriebene `config.d/`-Struktur
- ➔ Original-Konfig durch „useful use of cat“ von Dateien in eigenem `config.d/` zusammensetzen

Zusammenfassung

- ➔ Die komplette Konfiguration eines Hosts ist in **einem** Paket
- ➔ Die Konfiguration zieht alle benötigte Software mit
- ➔ Einfache Verifikation der installierten Konfiguration
- ➔ Gleiche Toolchain wie für Softwareinstallation
- ➔ Paketformat ist
 - ➔ Wohlbekannt
 - ➔ Relativ einfach (Für mich als „RPM-Person“)
- ➔ Konfiguration wird zentral vorberechnet

Werbung



Nexus Yum Plugin erhältlich unter
<https://code.google.com/p/nexus-yum-plugin/>

YADT

- ➔ Kennt dein Rechenzentrum
 - ➔ Das RZ kann einfach modelliert werden
 - ➔ YAML-basierte Beschreibung von
 - ➔ Services
 - ➔ Applikationen
 - ➔ Hosts
- ➔ Kennt Abhängigkeiten zwischen
 - ➔ Paketen
 - ➔ Services
 - ➔ Systemen

Konfiguration

➔ Definition in einer Datei „target“:

```
name: probau
log-dir: logs

hosts:
- hambau*.example.com
- berbau*.exampe.com
```

Konfiguration

➔ Service-Definition in „yadt.services“:

```
- service1:  
  needs_services: [service2]  
  
- service2:  
  needs_services: [service3]  
  
- service3:
```

Konfiguration

👉 Notation:

→ `service://hostname/servicename`

→ `host://hostname/`

→ `artefact://hostname/package/version`

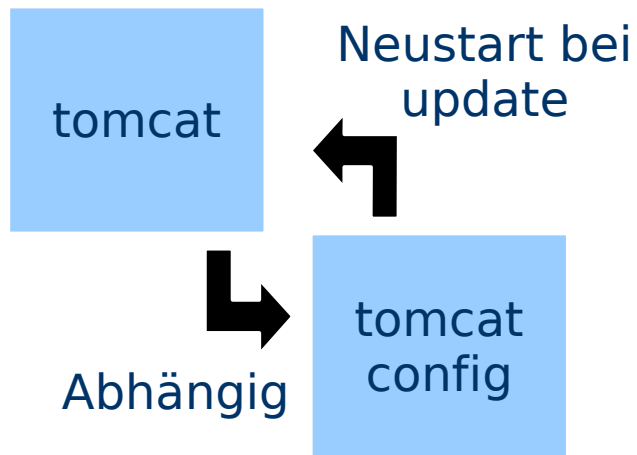
→ `yadt status service://hostname1/httpd`

→ `yadt ignore host://{host2|host33}`

→ `yadt lock -m „REASON“ host://hostname3`

→ `yadt updateartefact artefact://
[host1..host15]/yadt-client`

YADT - Die kleinste Einheit



```
yadt.services:
```

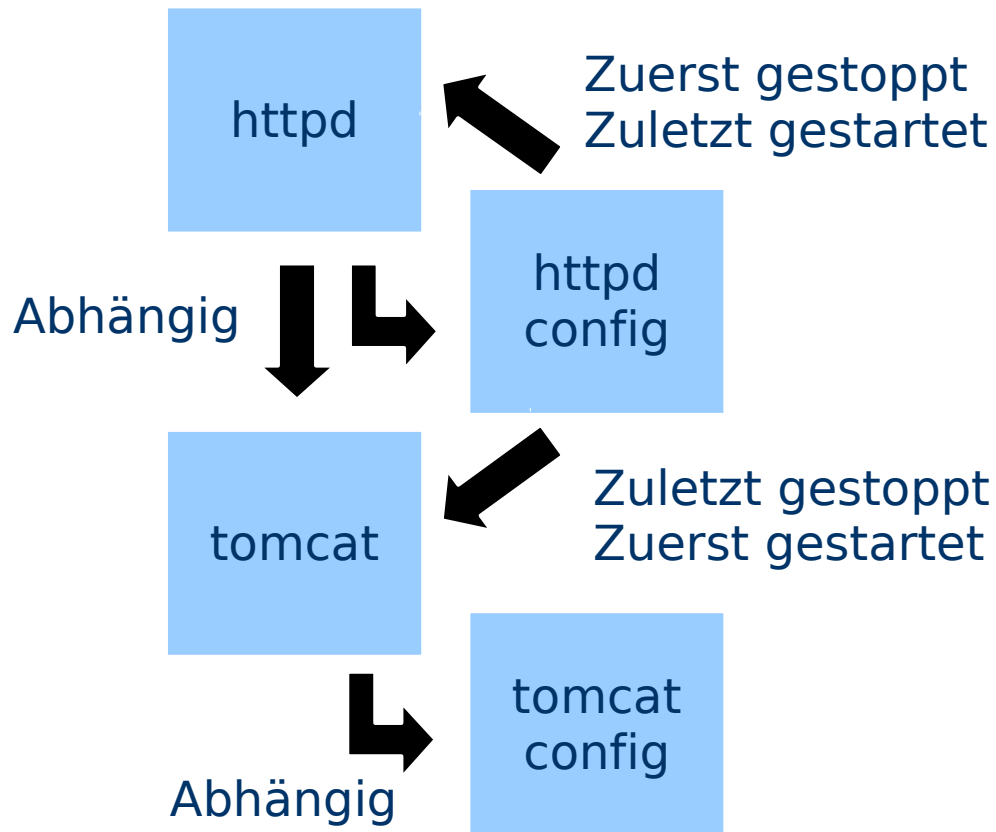
```
- tomcat:
```

```
Target:
```

```
hosts:
```

```
- foo.example.com
```

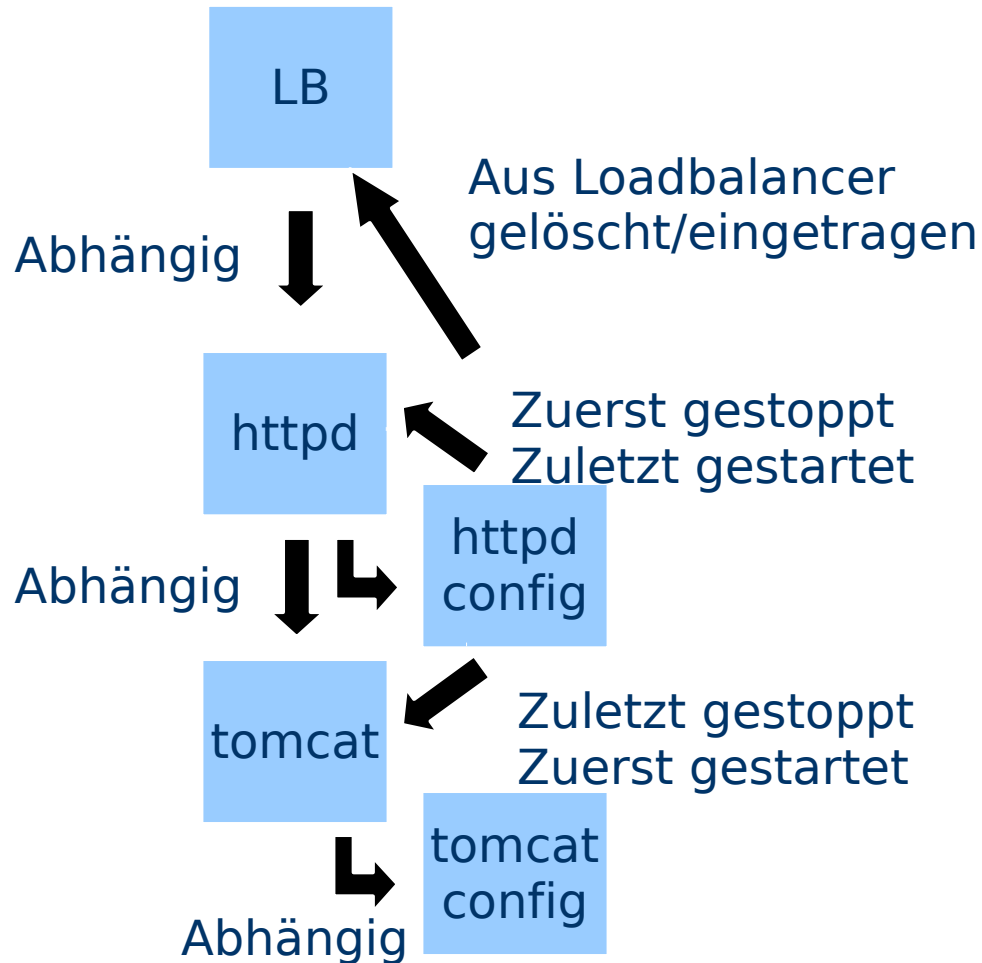
YADT - Einfache Abhängigkeiten



yadt.services:

- httpd:
needs_services: [tomcat]
- tomcat:

YADT - Externe Services



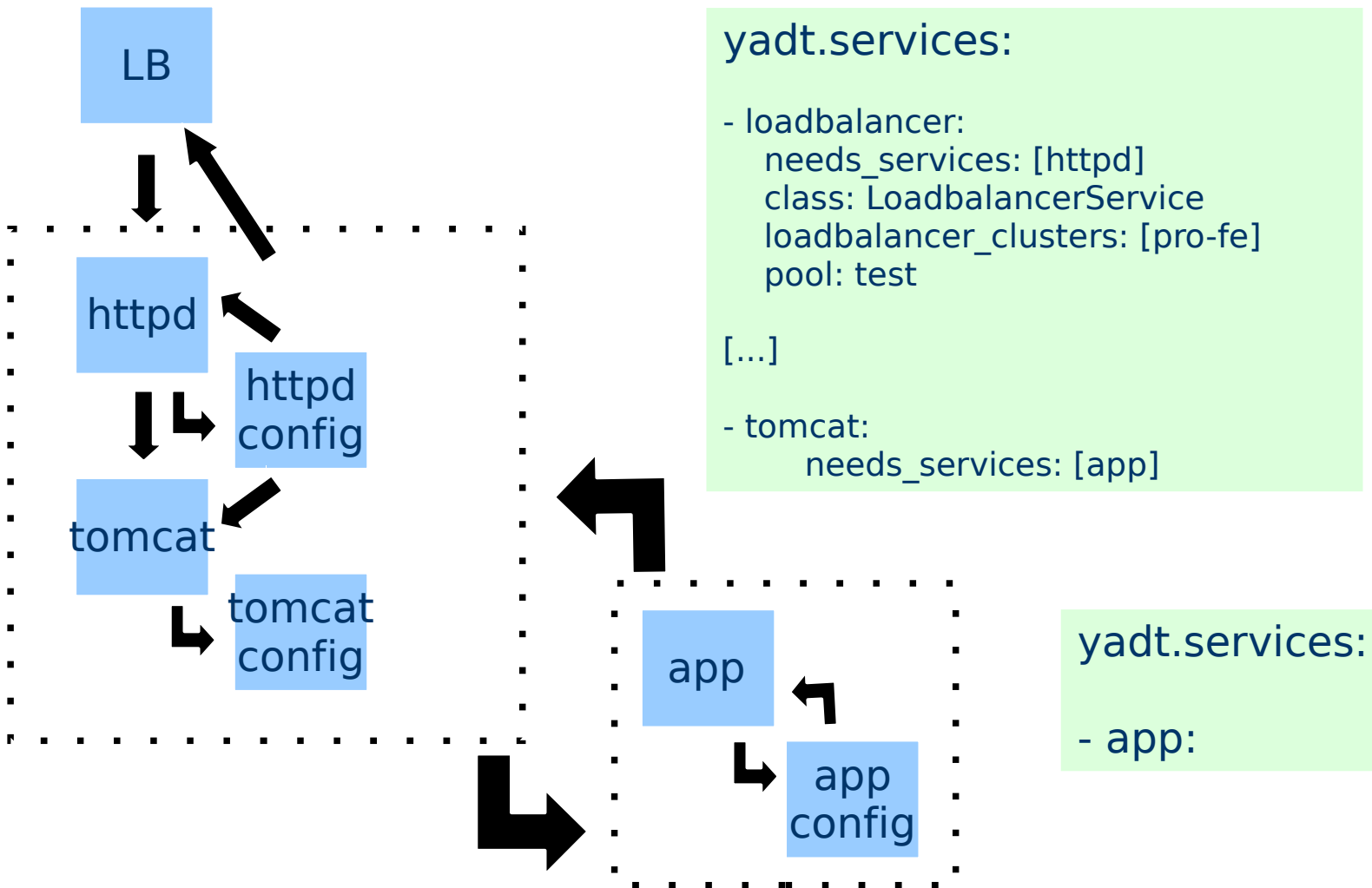
yadt.services:

- loadbalancer:
 - needs_services: [httpd]
 - class: LoadbalancerService
 - loadbalancer_clusters: [pro-fe]
 - pool: test
 - port: 80
 - status_max_tries: 2
- httpd:
 - needs_services: [tomcat]
- tomcat:

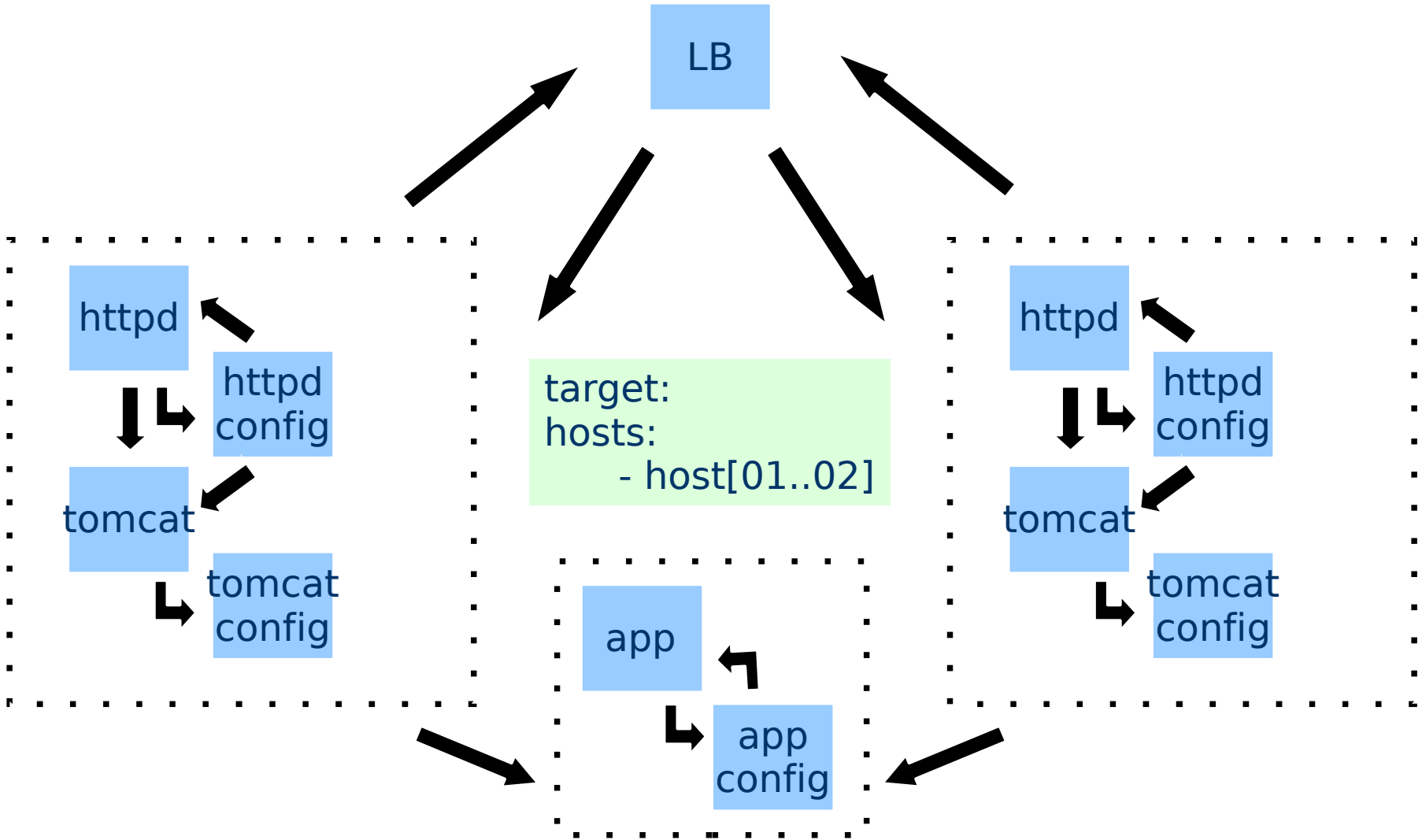
Externe Services

- ➔ YADT hat ein „Service Layer“
 - ➔ Python-Modul
 - ➔ Kann auch Skripte ausführen
 - ➔ Loadbalancer:
 - ➔ Benutzt die F5 Big IP Python-API
 - ➔ Hosts „disablen/enablen“
 - ➔ Wir nutzen das auch, um Nagios ruhig zu stellen
 - ➔ Noch kein Open Source
 - ➔ Generalisierung fehlt

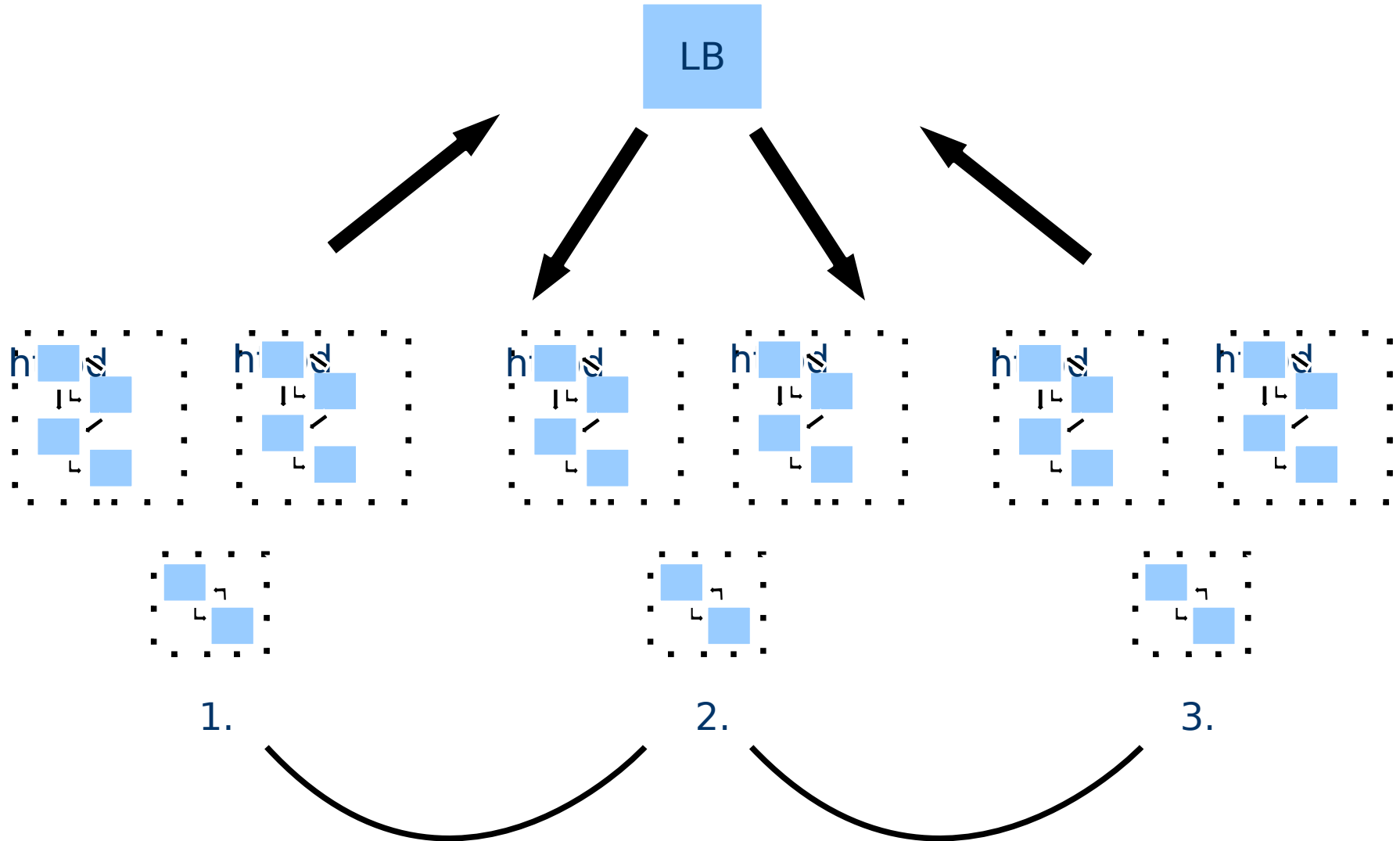
YADT - Services auf anderen Systemen



YADT - Mehr Komplexität!



YADT - Chunks und Wellendeployment



Screenshot!

```
ydtshr001 [Running] - Oracle VM VirtualBox
Machine View Devices Help

ydtclt001
  i frontend(frontservice)
  i middleservice
  i backservice
  i host lock
  u updates

ydtclt002
  i frontend(frontservice)
  i middleservice
  i backservice
  i host lock
  u updates

legend services: i0? up/down/unknown io ignored up/down
legend hosts: iL locked by me/other u update pending

queried 4 seconds ago

status: 100% 0% i 2/2 frontservices up, 0/2 hosts uptodate

[yadt@ydtshr001 ydtclt1]$ _
```

Fazit

- ➊ Konfiguration via RPM funktioniert erstaunlich gut
 - ➔ Man muss um ein paar Probleme herumrouten
 - ➔ Einfache Pflege (für jeden, „just change config“)
 - ➔ Distribution via yum
 - ➔ Ein RPM zieht die ganze Maschine hinterher
 - ➔ Maschine „neu“ aufsetzen?
 - ➔ yum remove config-rpm
 - ➔ yum install config-rpm

Fazit

- YADT ist „work in progress“ – aber zuverlässig
 - ➔ Bei vielen Maschinen im Target manchmal langsam
 - ➔ Vor allem wenn Loadbalancer/Nagios involviert sind
 - ➔ Service-Layer noch nicht Open Source
 - ➔ Einfache Konfiguration
 - ➔ Benötigt ein paketbasiertes System
 - ➔ Bei uns im täglichen Einsatz

Blick in die Zukunft

- ➔ Neue yadt-shell (das Interface) ist in interner „Beta“
- ➔ Yadt bekommt Fähigkeit zu parallelisieren
 - ➔ Um Yadt schneller zu machen
 - ➔ Implementierung des „Server, Rack, Datacenter“-Szenarios
 - ➔ Variable „fault tolerance“
- ➔ Funktionierendes Demosystem wäre nett =:)

Der Schluss (endlich!)

YADT

<https://code.google.com/p/yadt/>

Yadt-rpm-config-maker

<https://code.google.com/p/yadt/>

Nexus YUM plugin

<https://code.google.com/p/nexus-yum-plugin/>

Vielen Dank. Bei Fragen: Einfach kontaktieren!



Kontakt:

Immobilien Scout GmbH
Andreasstraße 10
10243 Berlin

Fon: +49 30 243 01-1036
Email: ralph.angenendt@immobilienscout24.de
URL: www.immobilienscout24.de