

Puppet

Seminar: Betriebssystemadministration

Jan Oberst - 25. Juni 2008

1. setup

- anschließen
- OS installieren
- konfigurieren
- booten

2. auto-deployment

- Updates
- Paketmanagement
- Aufgabenverteilung

Über Puppet

3

- Nennt sich “Resource Abstraction Layer”
 - ähnlich: Cfengine
- Open Source
 - Luke Kanies
 - reductivelabs.com: Commercial Support
- In Ruby geschrieben
 - Konfiguration über Ruby Klassen

Master/Client

4

Puppet Master (Server)

- Speichert alle Konfigurationsdateien
- Kennt Konfiguration für IP/Hostname

Client

- Puppetd übernimmt Konfiguration
- Fragt Server nach einer Konfiguration für sich
- Konfiguration nur einmal anwenden: Idempotent!

Neuer Server?

5

1. Hardware Setup

2. Linux/Unix Installation & Konfiguration

- IP & DNS
- Puppet Client installieren

3. Deployment via Puppet

- Software installieren (Abhängigkeiten)
- Serversettings zuweisen
- Daten laden (Dateien / database dump)

Puppet Config

Packages

7

```
$ssh = $operatingsystem ? {  
    solaris => SMCossh,  
    default => openssh  
}  
  
package {  
    $ssh:  
        ensure => installed,  
        alias => openssh,  
        require => Package[openssl]  
}
```

```
file {  
  "/etc/passwd":  
    ensure => present;  
  "/etc/group":  
    owner  => "root",  
    group  => "root";  
}
```

```
file {  
  "/etc/apt/sources.list":  
    owner => root, group => root, mode => 644,  
    source => "puppet://master/files/apt/sources.list".  
}
```

Schedule

9

```
schedule {  
    maintenance:  
        range => "2 - 4",  
        period => daily,  
        repeat => 1  
}  
exec {  
    "/usr/bin/apt-get update":  
        schedule => daily  
}
```

Cronjobs

10

```
cron {  
  logrotate:  
    command => "/usr/sbin/logrotate",  
    user => root,  
    hour => 2,  
    minute => 0  
}
```

Service

11

```
service {  
  mysql:  
    ensure => running,  
    enable => true,  
    require => Package["mysql-server"],  
    subscribe => File["/etc/my.cnf"],  
}
```

cron, **exec**, **file**, filebucket, group, host, interface, mailalias,
maillist, mount, notify, **package**, schedule, **service**, sshkey,
tidy, user, yumrepo, zones

Klassen

13

Servertypen sind Klassen

- `debian`
- `webserver inherits debian`

Maschinen installieren Klassen

- `node client1 { include debian, webserver, fileserver }`

Beispielklasse: **apt-get**

```

class aptget {
  file {
    "/etc/apt/sources.list":
      owner => root, group => root, mode => 644,
      content => template("apt/sources.list.erb");
    "/etc/apt/preferences":
      owner => root, group => root, mode => 644,
      source => "puppet://srv/files/apt/preferences.$version";
  }
  exec { subscribe-echo:
    command      => "/usr/bin/apt-get -q -q update",
    logoutput     => false,
    refreshonly  => true,
    subscribe    => file["/etc/apt/sources.list"]
  }
}

```

templates/apt/sources.list.erb:

```
deb http://ftp.de.debian.org/debian <%= version %> main
```

Beispielklasse: **MySQL Server**

```
class mysqlserver {
  package { "mysql-server-5.0": ensure => installed }
  package { "mysql-client-5.0": ensure => installed }
  package { "mysql-common": ensure => installed }

  exec {
    "Set MySQL server root password":
    subscribe => [
      Package["mysql-server-5.0"],
      Package["mysql-client-5.0"],
      Package["mysql-common"]
    ],
    refreshonly => true,
    unless => "mysqladmin -uroot -p$mysqlrootpassword status",
    path => "/bin:/usr/bin",
    command => "mysqladmin -uroot password $mysqlrootpassword",
  }
}
```

```
import "functions.pp"
import "classes/*"

node debian {
    include debian
    include sudo
}

node client1 inherits debian {
    $version = 'sid'
    include aptget

    $mysqlrootpassword = "root"
    include mysqlserver
}
```

Live Demo

19

Konfiguration

- Debian Sid
- Puppet installiert
- mit statischer IP&DNS

Installation

- Client 1: Webserver Lighttpd + Template
- Client 2: Webserver Lighttpd + Template und MySQL Server
- Komplette Konfiguration in site.pp

Factor: Erweiterbarkeit

20

```
Factor.add("timezone") do
  confine :operatingsystem => :debian
  setcode do
    File.readlines("/etc/timezone").to_a.last
  end
end
```

Pulling Strings with Puppet: Configuration Management Made Easy

James Turball, Apress 2008

http://howtoforge.com/installing_puppet_on_ubuntu

<http://people.redhat.com/dlutter/puppet-app.html>

<http://highscalability.com/product-puppet-automated-administration-system>

<http://reductivelabs.com/trac/puppet/wiki/PuppetBestPractice>