Puppet Konfigurationsmanagement

Ulrich Habel <rhaen@pkgbox.de>

Hallo!

Ulrich Habel

Open Source, Perl Evangelist

 @rabenfeder

rhaen@pkgbox.de

Schnellstart

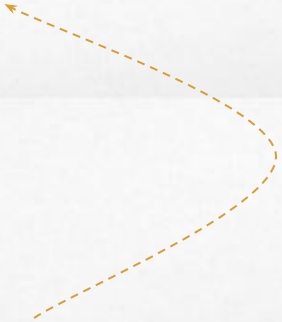
```
package {'memcached':
  ensure => present,
}
```

Schnellstart

```
$ sudo puppet apply -v <datei>
```

Schnellstart

```
package {'memcached':
   ensure => present,
}
```



Prüft ob "memcache" installiert ist,
!= installiert
== nichts wird ausgeführt

Running ssh in a for loop is not a solution!
(Luke Kanies)

Puppet - Fakten

Puppet ist Open Source

Hervorgegangen aus den Erlebnissen rund um cfengine.

Stichpunkte

- Entwickelt in 2004/2005
- Abstraktionlayer für Betriebssysteme und Hardware (facter)
- Apache 2.0 Lizenz
- Äußerst aktives Open Source Projekt

Etabliert!

unterstützte Systeme und Einsatz

Betriebsysteme

- RedHat Linux, CentOS, Scientific Linux, Oracle Linux, Ascendos
- Debian, Ubuntu
- Fedora
- SuSE Enterprise Server
- Gentoo
- Mandriva
- ArchLinux
- FreeBSD > 4.7
- OpenBSD > 4.1
- MacOS X
- Solaris > 10
- Windows*

Puppet ist in Ruby implementiert

Eine Vielzahl von Systemen, die über einen Ruby Interpreter < 1.8.7 verfügen, können Puppet verwenden. Es ist sinnvoll die vorgefertigten Pakete der Distribution zu verwenden, ggf. auf 3rd Party Channels ausweichen (EPEL)

Besonderheiten

- **DSL - Language**
  Puppet stellt damit quasi eine eigene Sprache zur Verfügung, die für die Domäne System Konfiguration entwickelt wurde
- **Standalone oder Client/Server**
  Puppet kann alleine verwandt oder in einer Client Server Architektur angewandt werden
- Klein anfangen -> groß skalieren!
- Prüft auf Compliance und führt gegebenfalls die erforderlichen Schritte durch

* *Windows wird derzeit nur von Puppet Enterprise (non-free) vollständig unterstützt.*

1.

Konfigurationsmanagement

Definitionsversuch (ohne Buzzwords)

Festlegen einer Konfiguration, die jederzeit überprüft und wiederhergestellt werden kann. Abweichungen können angezeigt werden.

Festlegen einer Konfiguration, die jederzeit überprüft und wiederhergestellt werden kann. Abweichungen können angezeigt werden.

Dabei sollte es egal sein ob es sich um 1, 10 oder 1000 Server handelt.

1.

Erweiterung der Config

memcached - der Service

Schnellstart

```
package {'memcached':
  ensure => present,
}

service {'memcached':
  ensure  => running,
  enable  => true,
}
```

Puppet - Interna der DSL

Ressourcen*

- package
- service
- user
- cron
- tidy
- exec
- file
- mount
- notify
- augeas
- sshkey
- mailalias
- [...]

Die Puppet DSL

Die DSL von Puppet ist leicht zu erlernen und erlaubt das Zusammensetzen von kleinen Bausteinen, damit größere Elemente entstehen.

Besonderheiten

- **Ressourcen**
  Puppet denkt in Ressourcen - alles was konfiguriert werden kann ist eine Ressource
- **Reihenfolge der Ausführung**
  Puppet arbeitet die Anweisungen in beliebiger Reihenfolge ab
- **Puppet - als Compliance Tool**
  Puppet ist **kein** Remote Execution Tool - es soll lediglich einen Zustand beschreiben und sicherstellen, dass dieser aktuell ist

* *https://docs.puppetlabs.com/references/latest/type.html*

Puppet - Interna der DSL

Der Puppet Katalog

Puppet bildet aus dem Code (entweder aus einem einzelnen Manifest oder aus einer Node Defintion einen Katalog der Ressourcen.

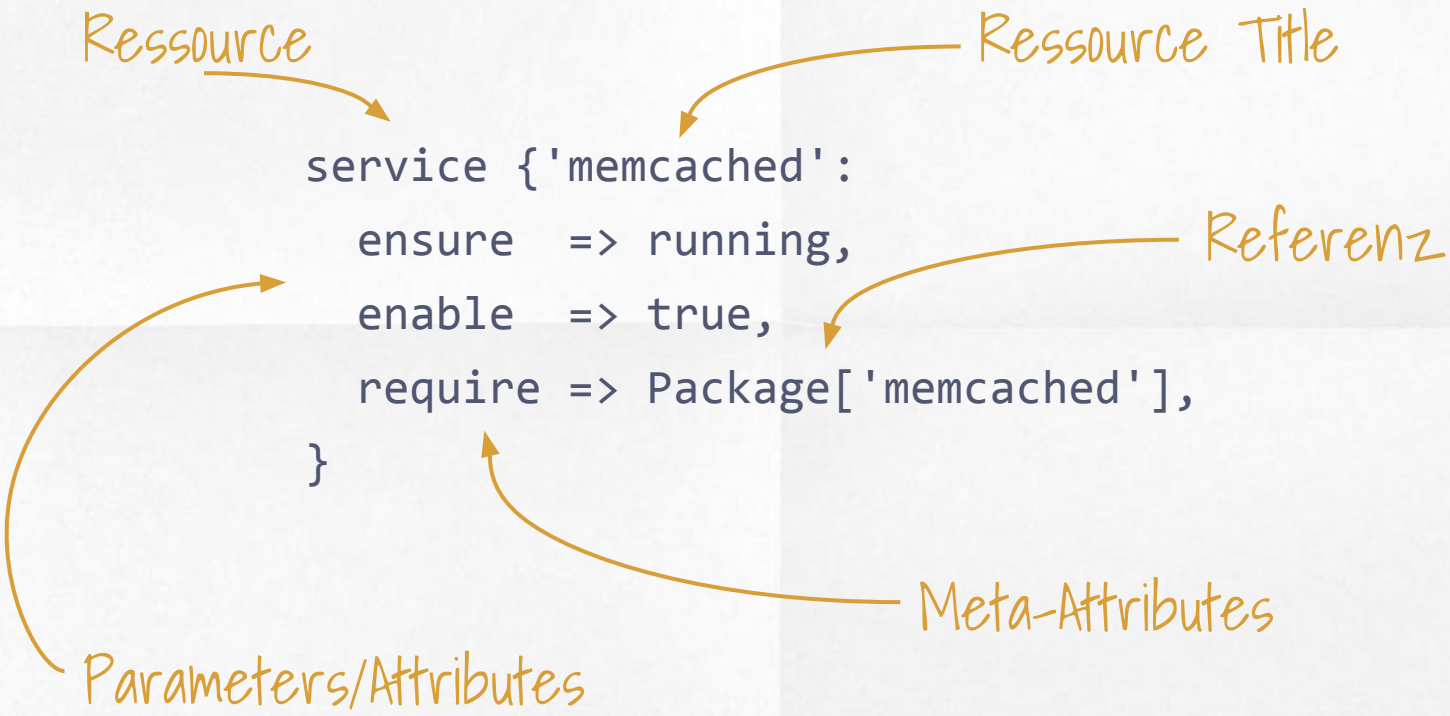
Der Katalog enthält:

- **Ressourcen**
- **Reihenfolge der Ausführung**

  (mit Hilfe von Verweisen)
- **Zustandsdefintion von Ressourcen**

  Was ist konfiguiert?
- **Für wen!**

Elemente einer Ressource



Weitere Sprachregeln

Die Puppet DSL - Teil 2

Ressourcen

- **Ressourcen können nur einmal im Katalog vorkommen**
- **Ressourcen können von anderen Ressourcen abhängen**
- **Abhängigkeiten können statisch - oder dynamisch sein**

  Es ist möglich Variablen zu verwenden, die während der Laufzeit gesetzt werden, daher können sich zur Laufzeit neue Abhängigkeiten ergeben

Variablen, Ressourcen und Referenzen

Die Puppet DSL - Teil 3

- $my_variable = ‘string’, [‘element 1’, ‘element 2’], {hash => value}
- Ressourcen (kennen wir schon!) werden immer klein geschrieben

Referenzen

Referenzen verweisen auf Ressourcen - der erste Buchstabe wird groß geschrieben

```
service {'memcached':
    [...],
    require => Package['memcached'],
    [...],
}
```

Ressourcen können mit Klassen zusammengefasst werden, diese können parametrisiert werden!

Puppet - Klassen

Definition einer Klasse

```
class memcached ($version = 'present') {
    package{'memcached':
      ensure => $version,
    }
}
```

Puppet - Klassen

Definition einer Klasse

```
class memcached ($version = 'present') {
    package{'memcached':
      ensure => $version,
    }
}
```

Aufruf einer Klasse

```
class {'memcached': }
```

Puppet - Klassen (parametrisiert)

Aufruf einer Klasse

```
class {'memcached':
  version => 'latest',
}
```

Modifikation der Klasse - "von aussen"

Klassen, Module - und Fehlendes

Die Puppet DSL - Teil 4

- Klassen können in Module verpackt werden
- Module können Aufgaben zugeordnet werden - “memcached, httpd, …”
- mehrere Module können in weiteren Module verwandt werden
- Mit dieser Art von Kapsulierung können Rollen abgebildet werden

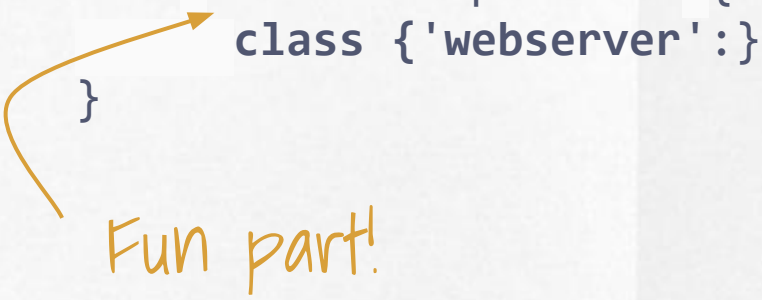
Node Definitionen

Node definition:

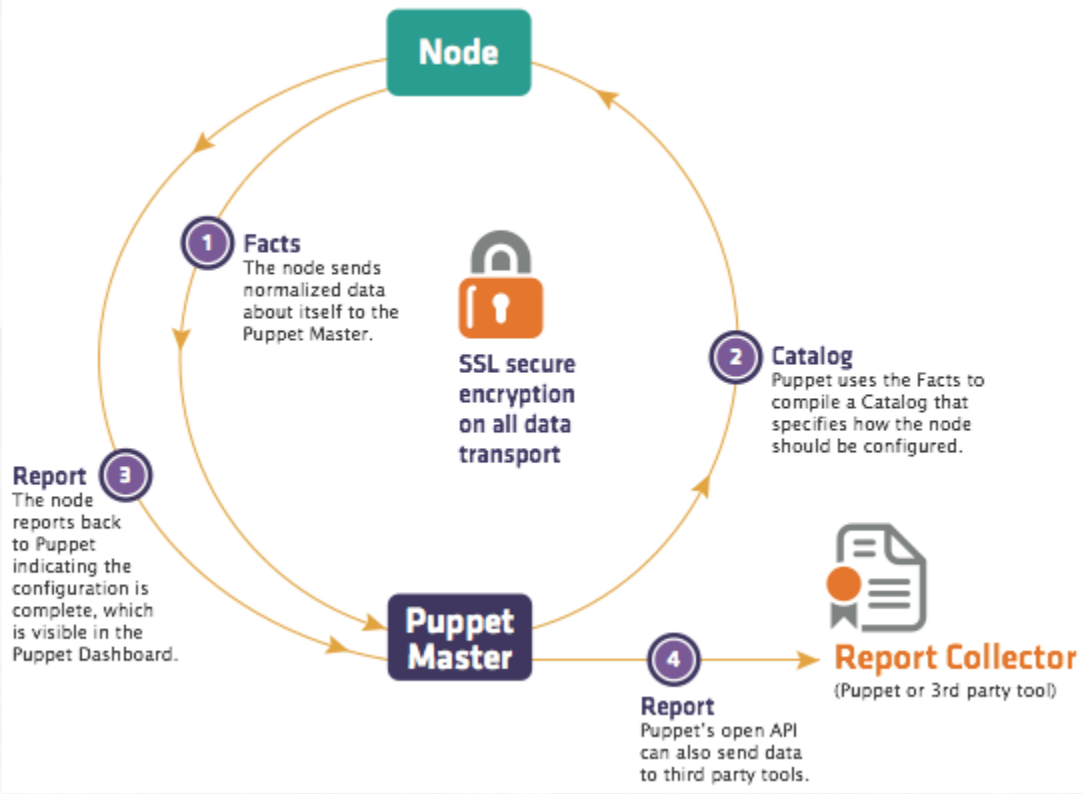
- Eine Node Defintion kann für einen oder mehrere Server gelten

```
node 'www1.example.com' {
      class {'webserver':}
}

node 'www*.example.com' {
      class {'webserver':}
}
```



Dataflow


Node
1
Facts
The node sends normalized data about itself to the Puppet Master.
SSL secure encryption on all data transport
2
Catalog
Puppet uses the Facts to compile a Catalog that specifies how the node should be configured.
Report
3
The node reports back to Puppet indicating the configuration is complete, which is visible in the Puppet Dashboard.
Puppet Master
4
Report
Puppet's open API can also send data to third party tools.
Report Collector
(Puppet or 3rd party tool)

Referenzen

Links:

- Puppet Labs (https://puppetlabs.com/)
- Resource Types (https://docs.puppetlabs.com/references/latest/type.html)
- Language Reference (https://docs.puppetlabs.com/puppet/3.7/reference/)
- PuppetForge (https://forge.puppetlabs.com/)
- Open Source Book - Puppet (http://www.aosabook.org/en/puppet.html)

Thanks!

Fragen?

Kontakt:
@rabenfeder
rhaen@pkgbox.de

Credits

Special thanks to all the people who made and released these awesome resources for free:

- Busy Icons by Olly Holovchenko
- Presentation template by SlidesCarnival
- Photographs by Unsplash
- Backgrounds by Pixeden

Presentation design

This presentations uses the following typographies and colors:

- Titles: **Shadows into light**
- Body copy: **Varela round**

You can download the fonts on this page:
https://www.google.com/fonts#UsePlace:use/Collection:Shadows+Into+Light|Varela+Round

Click on the “arrow button” that appears on the top right 

- Dark grey **#505670**
- Light grey **#979cb8**
- Blue **#01abcf**
- Yellow **#f9ac08**
- Magenta **#ea3a68**
- Green **#aacf20**

You don’t need to keep this slide in your presentation. It’s only here to serve you as a design guide if you need to create new slides or download the fonts to edit the presentation in PowerPoint®