

# Puppet 4

Was ist neu, was wird anders, was  
geht nicht mehr?

## FFG 2016

Martin Alfke

[ma@example42.com](mailto:ma@example42.com)



# Martin Alfke

CEO example42 GmbH  
Freelance CfgMgmt Expert  
Network

Puppet seit 2007

Puppet Trainer, Consultant

Co-Autor von “Puppet 4  
Essentials”

GitHub ‘voxpupuli’



# Puppet 4

- Agenda
  - Puppet Server <-> Passenger
  - puppet-agent Paket <-> puppet Paket
  - Environments



# Puppet 4

- Agenda II
  - Neues in Puppet 4
    - Daten Typen
    - Iterationen
    - EPP Templates
    - HEREDOC
    - Daten in Environments und Modulen (hiera v4)



# Puppet 4

- Agenda III
  - alte Best Practices und Puppet 4
    - Node Vererbung
    - Vergleiche und Variablen Namen
    - Syntax von Resource Referenzen
    - Bindestrich <-> Minuszeichen
    - Relative Klassennamen und import Funktion



# Puppet 4

- Agenda IV
  - Migration auf Puppet 4
    - Katalog Diff
    - Environment mit Future Parser
    - zweiter Puppet Server



# Puppet Server <-> Passenger

- neuer Puppet Server
  - JVM + Clojure + JRuby
  - min 2 GB RAM
  - puppetserver Paket



# Puppet Server <-> Passenger

- Trapperkeeper
  - Isolation von Diensten
    - CA
    - Catalog Compile
    - File Serving





# Puppet Agent <-> Puppet

- puppet 3.x
  - benötigt System Ruby
  - Puppet 3 braucht Ruby 1.9.3
  - braucht GEM Erweiterungen
  - Konfiguration in /etc/puppet



# Puppet Agent <-> Puppet

- puppet-agent (Puppet 4)
  - braucht Ruby 2.1.7
  - Bundle aller notwendiger Bestandteile (PC 1)
  - Installiert nach /opt/puppetlabs
  - Konfiguration in /etc/puppetlabs/puppet
  - Enterprise und FOSS Agent Paket



# Environments

- statische Environment
- dynamische Environments
- Environment Path



# Environments

- statische Environment
- Sektion in puppet.conf
- Restart Puppet Master nach Änderung notwendig

```
[development]
manifests = /etc/puppet/development/manifests
modules = /etc/puppet/development/modules
```

```
[testing]
manifests = /etc/puppet/testing/manifests
modules = /etc/puppet/testing/modules
```



# Environments

- dynamische Environments
- master Sektion in puppet.conf
- Verwendung von `{environment}`

```
[master] # oder [main]  
manifests = /etc/puppet/{environment}/manifests  
modules = /etc/puppet/{environment}/modules
```



# Environments

- Environment Path
  - master oder main Sektion in puppet.conf
  - neue Environments ohne Neustart sofort verfügbar
  - Puppet 4: /etc/puppetlabs/code/environments/

```
[main]
```

```
environmentpath = /etc/puppetlabs/code/environments
```



# Neues in Puppet 4

- Daten Typen
- Iteration in DSL
- EPP Templates (Embedded Puppet)
- HEREDOC in DSL
- Daten in Environments und Modulen (hiera v4)



# Daten Typen

```
class example42_ssh (  
    $server      = true,  
    $client      = true,  
    $allow_root  = true,  
    $x11_forward = true,  
) {  
    File {  
        owner => 'root',  
        group => 'root',  
        mode  => '0440',  
    }  
    if $server {  
        include ssh::server
```

```
class { 'example42_ssh':  
    server => 'false',  
}
```





# Daten Typen

```
class example42_ssh (
    $server      = true,
    $client      = true,
    $allow_root  = true,
    $x11_forward = true,
) {
    validate_bool($server, $client, $allow_root, $x11_forward)
    File {
        owner => 'root',
        group => 'root',
        mode  => '0440',
    }
    if $server {
```



# Daten Typen

```
class example42_ssh (
  Boolean $server      = true,
  Boolean $client      = true,
  Boolean $allow_root  = true,
  Boolean $x11_forward = true,
) {
  File {
    owner => 'root',
    group => 'root',
    mode  => '0440',
  }
  if $server {
```



# Daten Typen

- Core Daten Typen
  - String
  - Integer, Float, Numeric
  - Boolean
  - Array, Hash
  - Regexp
  - Undef, Default



# Daten Typen

- Abstrakte Daten Typen
  - Scalar
  - Collection, Variant, Enum
  - Data, Pattern
  - Tuple, Struct
  - Optional
  - Catalogentry, Type
  - Any, Callable



# Daten Typen

- Der Inhalt von Daten Typen kann weiter spezifiziert werden:
  - `String[1, 10]` -> 1-10 Buchstaben
  - `Array[String]` -> ein Array aus Strings
  - `Hash[Integer, Hash]` -> ein Hash mit einem Integer als Key und Subhashes



# Daten Typen

- Besonderheiten:
- Bei Gleitkommazahlen zwischen -1 und 1 die führende 0 angeben:

```
class data_test (  
    Integer $int,  
    Real    $real  
)  
{  
    notify { "Int: ${int}, Real: ${real}": }  
}
```

```
class { 'data_test':  
    int  => -1,  
    real => 0.4,  
}
```



# Daten Typen

- Besonderheiten:
- Verwendung des Default Daten Typen:

```
$real_server = $server ? {  
  Boolean => $server,  
  String  => str2bool($server),  
  Default => true,  
}
```



# Iterationen

- In Puppet 3: Array als Title

```
$packages = ['libxml2', 'libjson', 'less']
```

```
package { $packages:  
  ensure => present,  
}
```

```
file { ['/opt/app', '/opt/app/config', '/opt/app/config/conf.d']:  
  ensure => directory,  
}
```





# Iterationen

```
# each
$symlinks = ['puppet', 'hiera', 'facter', 'r10k']

$symlinks.each |String $symlink| {
  file { "/usr/local/bin/${symlink}":
    ensure => link,
    target => "/opt/puppetlabs/puppet/bin/${file}",
  }
}
```



# Iterationen

```
# filter
$pkg_array = [ 'libjson', 'libjson-devel', 'libfoo', 'libfoo-devel' ]
$packages = $pkg_array.filter |$element| {
    $element =~ /devel/
}
# Notice: Packages: [libjson-devel, libfoo-devel]
```

```
# slice
$array = [ '1', '2', '3', '4' ]
$array.slice(2) |$slice| {
    notify { "Slice: ${slice}": }
}
# Notice: Slice: [1, 2]
# Notice: Slice: [3, 4]
```



# EPP Templates

- Puppet Syntax anstelle von Ruby
- epp() Funktion anstelle von template() Funktion
- EPP Templates kennen Parameter (optional)



# EPP Templates

```
class example42_ssh (
  Boolean      $allow_root = true,
  Array[String] $allow_group = ['admins'],
){
  file { '/tmp/sshtest':
    ensure => file,
    content => epp('example42_ssh/sshtest.epp'),
  }
}
```



# EPP Templates

```
# Demo EPP
```

```
#
```

```
<% if $example42_ssh::allow_root { -%>
```

```
Root darf
```

```
<% } else { -%>
```

```
Root darf nicht
```

```
<% } -%>
```

```
Wer darf:
```

```
<% $example42_ssh::allow_group.each |$group| { -%>
```

```
<%= $group %>
```

```
<% } -%>
```



# EPP Templates

- Wichtig:
  - absolute Variablen Scopes verwenden
  - oder
  - Parameter verwenden



# EPP Templates

```
class example42_ssh (  
  Boolean      $allow_root = true,  
  Array[String] $allow_group = ['admins'],  
) {  
  file { '/tmp/sshtest':  
    ensure => file,  
    content => epp('example42_ssh/sshtest.epp',  
      {  
        allow_root => $allow_root,  
        allow_group => $allow_group,  
      }  
    ),  
  }  
}
```



# EPP Templates

```
<% |
  Boolean $allow_root,
  Array[String] $allow_group
| -%>
# Demo EPP
#
<% if $allow_root { -%>
Root darf
<% } else { -%>
Root darf nicht
<% } -%>

Wer darf:
<% $allow_group.each |$group| { -%>
<%= $group %>
<% } -%>
```





# EPP Templates

- Achtung bei Defines und EPP Templates
  - Ein EPP template kann nicht auf den lokalen Scope zugreifen
  - Alle Variablen, die im Template verwendet werden, **müssen** als Parameter übergeben werden



# HEREDOC

- Kleine Config Fragmente im Puppet DSL Code sahen immer schlimm aus:

```
class example42_motd (
){
    $content = 'Welcome to <%= @fqdn %>
This system is managed by Puppet.
Local changes will be overwritten.'
    file { '/etc/motd':
        ensure => file,
        content => inline_template($content),
    }
}
```



# HEREDOC

- HEREDOC unterstützt Fix Code Indentation

```
class example42_motd (
){
  $content = @('EOF')
  Welcome to <%= @fqdn %>
  This system is managed by Puppet.
  Local changes will be overwritten.
  | 'EOF'
  file { '/etc/motd':
    ensure => file,
    content => inline_template($content),
  }
}
```



# HEREDOC

```
class example42_motd (
){
  $content = @('EOF')
  Welcome to <%= $fqdn %>
  This system is managed by Puppet.
  Local changes will be overwritten.
  | 'EOF'
  file { '/etc/motd':
    ensure => file,
    content => inline_epp($content),
  }
}
```



# HEREDOC

- HEREDOC kann direkt Variablen ersetzen:

```
class example42_motd (
){
  $content = @("EOF")
  Welcome to $::fqdn
  This system is managed by Puppet.
  Local changes will be overwritten.
  | EOF
  file { '/etc/motd':
    ensure => file,
    content => $content,
  }
}
```



# Daten in Environments und Modulen (hiera v4)

```
class example42_ssh (  
  $pkgname = $example42_ssh::params::pkgname,  
) inherits example42_ssh::params {  
  # ...  
}
```

```
include example42_ssh
```



# Daten in Environments und Modulen (hiera v4)

- Klassen und automatic data binding (hiera)

```
# os/RedHat.yaml
example42_ssh::pkgname:
  - 'openssh-server'
  - 'openssh-client'
```

```
# os/Debian.yaml
example42_ssh::pkgname:
  - 'openssh-server'
```

```
# os/BSD.yaml
example42_ssh::pkgname:
  - 'ssh'
```



# Daten in Environments und Modulen (hiera v4)

- Params Klasse mit 'fail safe'

```
class example42_ssh::params {
  case $::fact['os']['family'] {
    'Debian': {
      $pkgname = ['openssh']
    }
    'RedHat': {
      $pkgname = ['openssh-server', 'openssh-client']
    }
  }
  default: {
    fail('OS wird nicht unterstützt.')
  }
}
```





# Daten in Environments und Modulen (hiera v4)

- Daten in Modulen
  - metadata.json
  - data\_provider
    - hiera
    - function



# Daten in Environments und Modulen (hiera v4)

- Hiera in Modulen (hiera.yaml)

```
---  
version: 4  
datadir: data  
hierarchy:  
  - name: 'OS family'  
    backend: yaml  
    path: "os/{facts.os.family}"  
  - name: 'Location'  
    backend: yaml  
    path: "location/{facts.location}"  
  - name: 'common'  
    backend: yaml
```



# Daten in Environments und Modulen (hiera v4)

```
# example42_ssh/functions/example42_ssh/data.pp
function example42_ssh::data() {
    $params = {
        'example42_ssh::cfgfile' => '/etc/ssh/sshd_config',
    }
    $os_params = case $facts['os']['family'] {
        'Debian': {
            { 'example42_ssh::pkgname' => ['openssh-server'], }
        }
        default: {
            {}
        }
    }
    $params + $os_params
}
```



# Daten in Environments und Modulen (hiera v4)

- Daten in Environments
  - environment.conf
    - environment\_data\_provider
  - lookup Methode
    - hiera.yaml (v4)
    - function/<env name>/data.pp
    - lib/puppet/functions/<env name>/data.rb



# Daten in Environments und Modulen (hiera v4)

- Migration:
  - lookup() Funktion anstelle von hiera, hiera\_array, hiera\_hash
    - merge Verhalten wird als Parameter angegeben
  - Entfernen der Vererbung in init.pp
  - Entfernen der params Klasse



# alte Best Practices und Puppet 4

- Node Vererbung
- Vergleiche von Daten Typen
- Strikte Variablen Namen
- Bindestrich <-> Minuszeichen
- Ruby DSL
- Relative Klassennamen
- import Funktion



# Node Vererbung

```
node default {  
  contain example42_base  
  contain example42_security  
}
```

```
node 'realnode.example42.com' inherits default {  
  contain example42_application  
}
```



# Node Vererbung

```
class profile::example42_base {  
    # ...  
}  
class profile::example42_security {  
    # ...  
}  
  
class role::example42_application {  
    contain profile::example42_base  
    contain profile::example42_security  
    contain profile::example42_application  
}  
  
node 'realnode.example42.com' {  
    contain role::example42_application  
}
```





# Vergleiche von Daten Typen

```
case $::operatingsystemmajrelease {  
  '8': {  
    notify { "We are ${::operatingsystem} version 8": }  
  }  
}  
  
if $::operatingsystemmajrelease + 0 > 7 {  
  notify { "OS version is greater than 7": }  
}
```



# Vergleiche von Daten Typen

```
$bool = true
$string = 'true'
$array = ['true', true]
$hash = {
    'string' => 'true',
    'array' => ['true'],
    'bool' => true,
}
```

```
if $bool == $string {
    # error
}

if $bool == $array[0] {
    # error
}

if $bool == $hash['bool'] {
    # no error
}
```



# Vergleiche von Daten Typen

```
$int = 123  
$float = 0.22  
$oct = 071  
$hex = 0xaa  
$string = '8'
```

```
$new = $int + $float  
notify { "Sum int and float: ${new}": }
```

```
$new2 = $string + $int  
notify { "Sum string and int: ${new2}": }
```

```
notify { "oct: ${oct}": }  
notify { "hex: ${hex}": }
```

```
$new3 = $hex + $int  
notify { "Sum hex and int: ${new3}": }
```

```
$new4 = $oct + $float  
notify { "Sum oct and float: ${new4}": }
```



# Vergleiche von Daten Typen

```
Warning: Scope(Class[main]): int + float: 'Float[123.22, 123.22]'  
Warning: Scope(Class[main]): string + int 'Integer[131, 131]'  
Warning: Scope(Class[main]): hex + int 'Integer[293, 293]'  
Warning: Scope(Class[main]): oct + float 'Float[57.22, 57.22]'
```



# Variablen Namen

```
$var = 'valid'
```

```
$var_name = 'valid'
```

```
$VAR2 = 'invalid'
```

```
$0var = 'invalid'
```

```
$var-name = 'invalid'
```



# Syntax von Resource Referenzen

```
# valid  
require => File['title']  
require => Package['apache', 'ssl']  
require => [ Package['apache'], Package['ssl'] ]  
require => Service[$service]
```

```
# invalid  
require => File ['title']  
require => Package[Apache]
```



# Bindestrich <-> Minuszeichen

```
# invalid

class my-ssh {
  ...
}

define app::conf-data {
  ...
}

$my-content = epp('my-ssh/my-template.epp')
```



# Relative Klassennamen

```
class profile::mysql {  
    # breaks in Puppet 4, works in Puppet4  
    include mysql  
}
```

```
class profile::mysql {  
    # Works in Puppet 3 and 4  
    include ::mysql  
}
```





# import Funktion

```
# The import function raises an error when called to inform the user that import is no longer supported.  
#  
Puppet::Functions.create_function(:import) do  
  def import(*args)  
    raise Puppet::Pops::SemanticError.new(Puppet::Pops::Issues::DISCONTINUED_IMPORT)  
  end  
end
```



# Migration auf Puppet 4

- Katalog Diff
- Environment mit Future Parser
- zweiter Puppet Server



# Puppet 4

Was ist neu, was wird anders, was  
geht nicht mehr?

## FFG 2016

Martin Alfke

[ma@example42.com](mailto:ma@example42.com)

