Puppenspielereien

Sebastian Hempel

Sebastian Hempel

IT-Consulting Hempel

🔗 https://it-hempel.de

https://github.com/ithempel

@ithempel

Puppet-Master seit 2010

Build Manager

Developer At Work (CC BY) | Robert Lowe

Continuous Integration

One pimping continuous integration light (CC BY) | Tim Lucas

Test-Umgebung

Test (CC BY) | David Bleasdale

Deployment

Dead Admin (CC BY-SA) | Arthur Caranta

Was ist Puppet?

Luke Kanies of Puppet Labs (CC BY-NC 2.0) | Igal Koshevoy

Wie arbeitet Puppet?

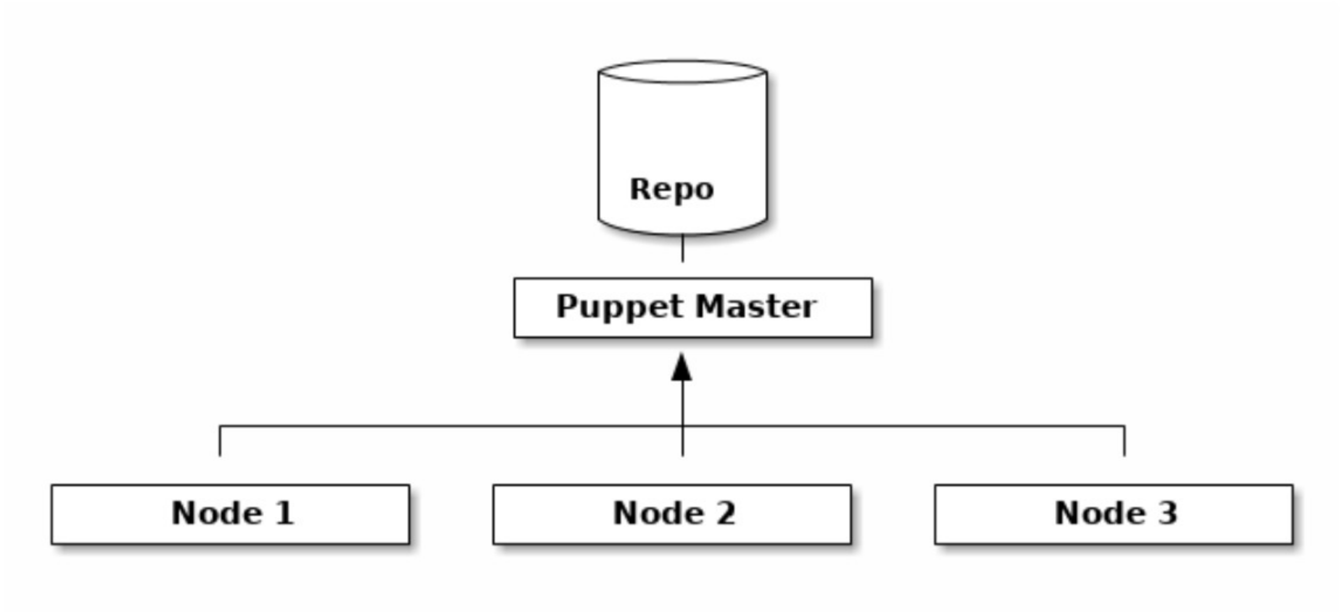
deklarativ

Wie soll es aussehen?

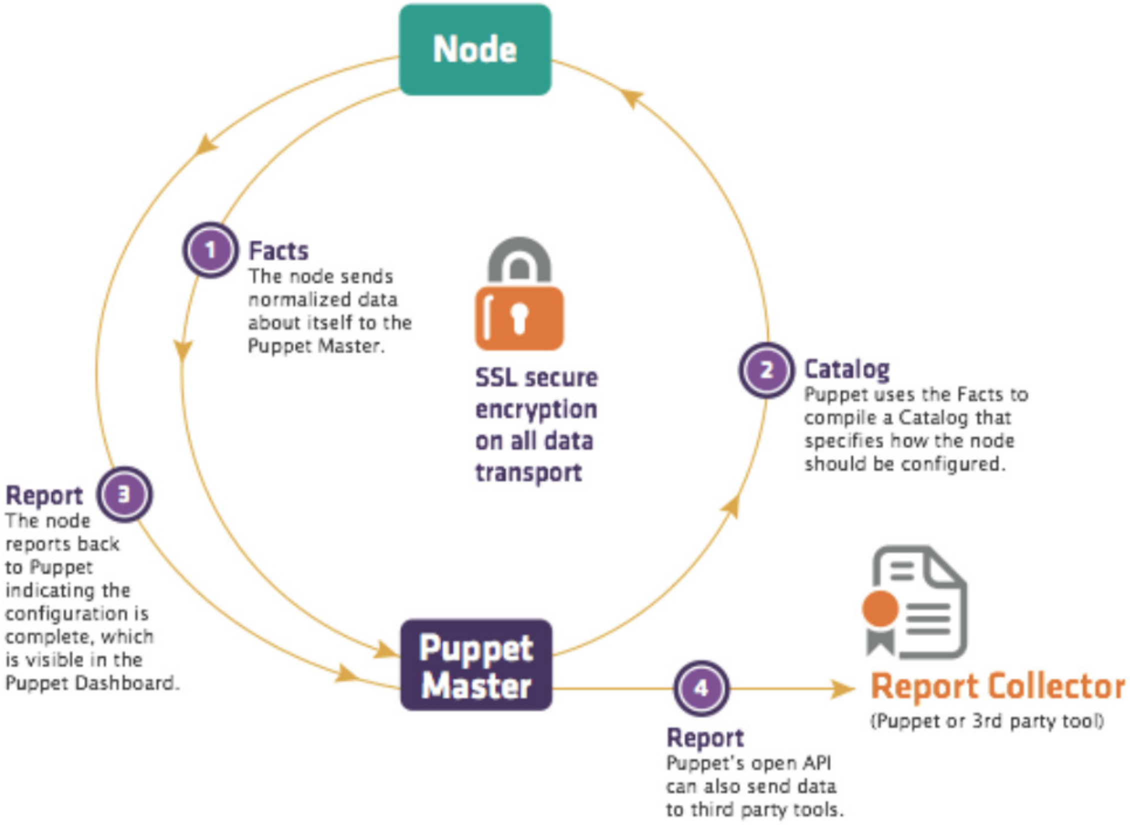
idempotent

mehrfach anwendbar

Architektur von Puppet



Ein Puppet-Lauf



1 Facts
The node sends normalized data about itself to the Puppet Master.

2 Catalog
Puppet uses the Facts to compile a Catalog that specifies how the node should be configured.

3 Report
The node reports back to Puppet indicating the configuration is complete, which is visible in the Puppet Dashboard.

4 Report
Puppet's open API can also send data to third party tools.

SSL secure encryption on all data transport

Puppet Language

Resource

```
file { '/etc/default/jenkins':
  ensure => present,
  source => 'puppet:///modules/jenkins/etc/default/jenkins',
  owner  => 'root',
  group  => 'root',
  mode   => '0644',
}
```

Puppet Language

`class` - einmal verwendbare Konfiguration

```
class apache {
  package { 'httpd':
    ensure => installed,
  }
  file { '/etc/httpd/httpd.conf':
    ensure => file,
    ...
  }
  service { 'httpd':
    ensure  => running,
    enabled => true,
  }
}
```

Puppet Language

`define` - user defined ressource

```
define apache::vhost($servername) {
  file { "/etc/httpd/conf.d/${servername}.conf":
    ensure => file,
    ...
  }
}
```

Puppet Language

Node Definition

```
node 'builderserver.example.com' {
  include 'some_class'
}
```

Puppet Language

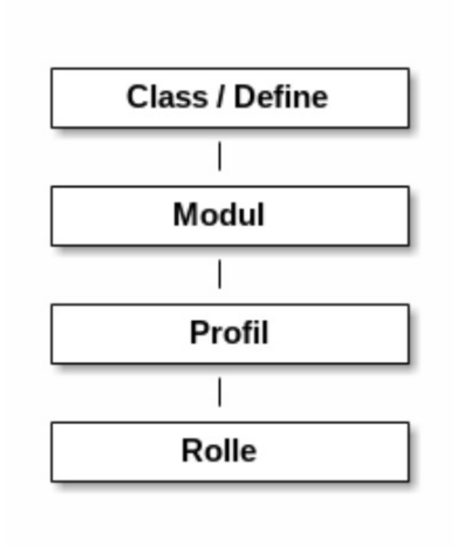
Conditional

```
if $::os['name'] == 'Debian' {
  ...
}

case $::operatingsystem {
  'Solaris':           { include role::solaris }
  'RedHat', 'CentOS':  { include role::redhat  }
  /^(Debian|Ubuntu)$/: { include role::debian  }
  default:             { include role::generic }
}
```

Puppet Language

Strukturierung



Class / Define

Modul

Profil

Rolle

Trennung von Code und Daten

Code

Aus welchen Elementen besteht die Konfiguration?

Daten

Konfigurationswerte / Attribute der Elemente

Trennung von Code und Daten

```
class applicaton_server($port = 80, $listen_ip = $::ipaddress) {
  ...
}
```

myserver.yaml

```
---
application_server::listen_ip: '127.0.0.1'
```

Reuse - Puppet Forge

https://forge.puppetlabs.com/

über 3000 Module

- Puppet Supported
- Puppet Approved

Build-Systeme

Grace Murray Hopper at the UNIVAC keyboard, c. 1960 (CC BY) | Carl Malamud

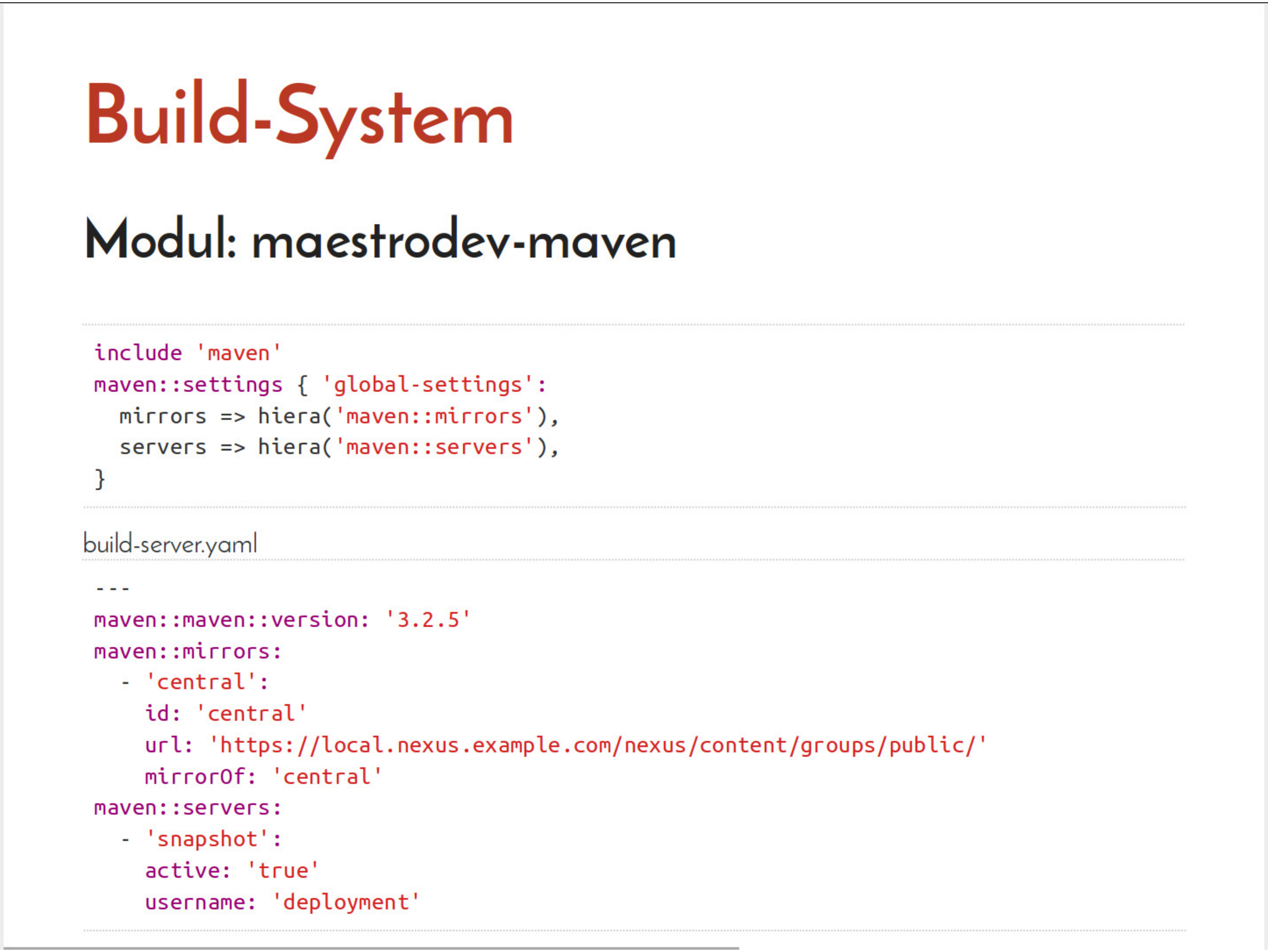
Java

Modul: puppetlabs-java

```
include 'java'
```

build-server.yaml

```
---
java::distribution: 'jdk'
```



Build-System

Modul: maestrodev-maven

```
include 'maven'
maven::settings { 'global-settings':
  mirrors => hiera('maven::mirrors'),
  servers => hiera('maven::servers'),
}
```

build-server.yaml

```
---
maven::maven::version: '3.2.5'
maven::mirrors:
  - 'central':
    id: 'central'
    url: 'https://local.nexus.example.com/nexus/content/groups/public/'
    mirrorOf: 'central'
maven::servers:
  - 'snapshot':
    active: 'true'
    username: 'deployment'
```

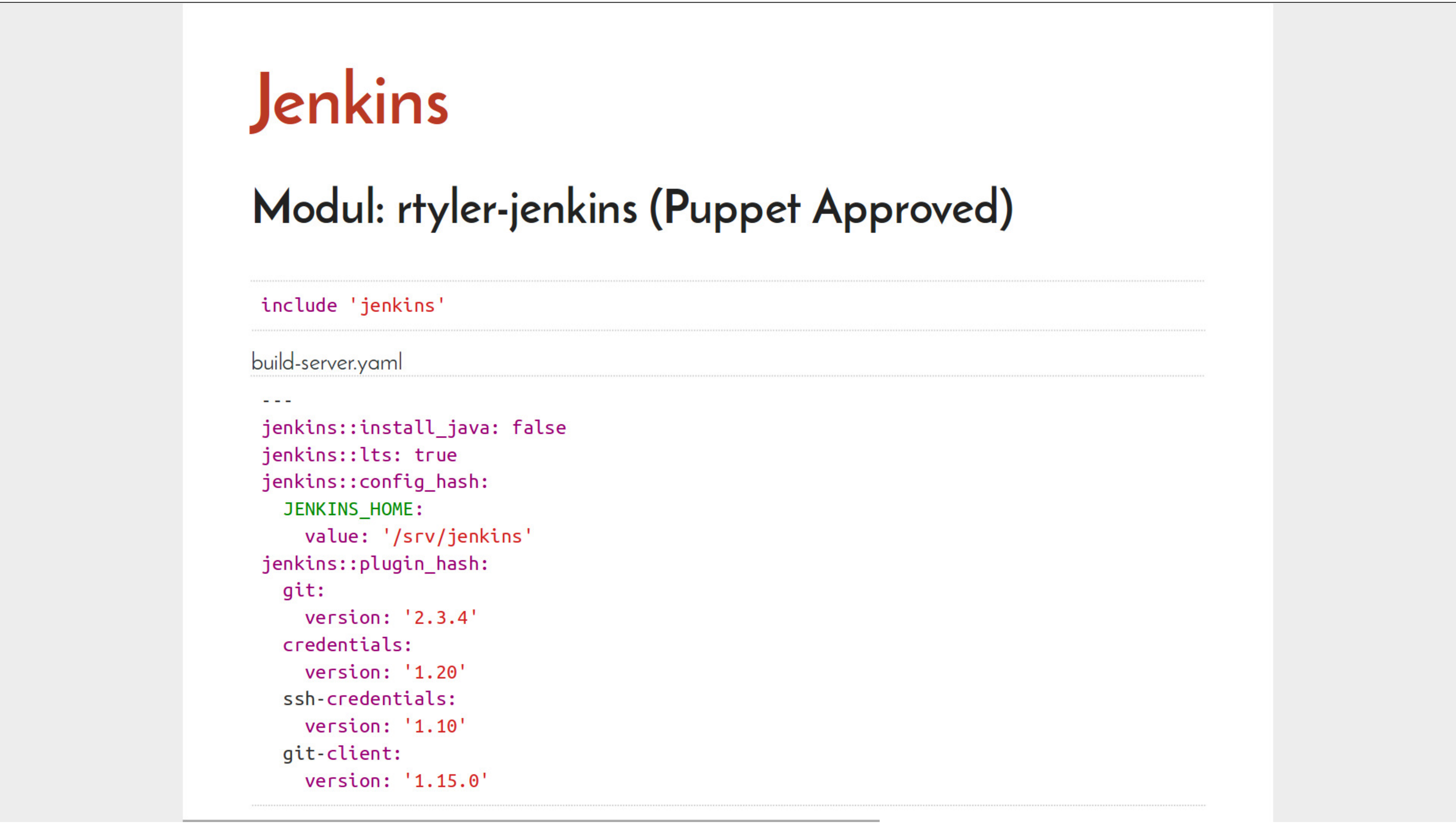
Git

Konfiguration über Template

```
package { 'git':
  ensure => installed,
}
file { '/var/lib/jenkins/.gitconfig':
  ensure  => file,
  content => template(git/gitconfig.erb),
  owner   => 'jenkins',
  group   => 'jenkins',
  mode    => '0644',
}
```

gitconfig.erb

```
[user]
        name = <%= @username %>
        email = <%= @usermail %>
[push]
        default = current
```



Jenkins

Modul: rtyler-jenkins (Puppet Approved)

```
include 'jenkins'
```

build-server.yaml

```
---
jenkins::install_java: false
jenkins::lts: true
jenkins::config_hash:
  JENKINS_HOME:
    value: '/srv/jenkins'
jenkins::plugin_hash:
  git:
    version: '2.3.4'
  credentials:
    version: '1.20'
  ssh-credentials:
    version: '1.10'
  git-client:
    version: '1.15.0'
```

Test- und Abnahme-Systeme

Sarawak, Borneo (CC BY) | One Laptop per Child

Application-Server

Konfiguration von

- DataSource
- JNDI Parameter
- Subsystem (z.B. JMS)
- User Accounts

Konfigurationsdateien

```
file { '/opt/wildfly/standalone/configuration/standalone.xml':
  ensure  => file,
  content => template('wildfly/standalone.xml.erb'),
  owner   => 'wildfly',
  group   => 'wildfly',
  mode    => '0640',
}
```

standalone.xml.erb

```
<datasources>
    <datasource jndi-name="java:jboss/datasources/<%= @database_name %>"
                pool-name="Applicaton"
                enabled="true" use-java-context="true">
        <connection-url>
            jdbc:mysql://localhost/<%= @database_name %>
        </connection-url>
        <driver>mysql</driver>
```

Command Line Tools

Modul: cpitman/puppet-jboss_admin (GitHub)

```
jboss_admin::server {'main':
  base_path => '/opt/jboss'
}

jboss_admin::resource::datasource{"/subsystem=datasources/data-source=${database_name}":
  ensure         => present,
  connection_url => "jdbc:mysql://localhost/${database_name}",
  driver_name    => mysql,
  jndi_name      => "java:jboss/datasources/${database_name}",
  jta            => true,
  user_name      => sa,
  password       => sa,
  server         => main
}
```

weitere Server (z.B. Datenbank)

Modul: puppetlabs-mysql

```
include 'mysql::server'
mysql::db { 'application_database':
  user     => 'db_user',
  password => 'db_password',
  host     => 'localhost',
  grant    => [ INSERT', 'UPDATE', 'SELECT', 'DELETE', ],
}
```

build-server.yaml

```
mysql::server::override_options:
  mysqld:
    bind-address: '127.0.0.1'
    datadir: '/srv/mysql'
```

Development → Operating

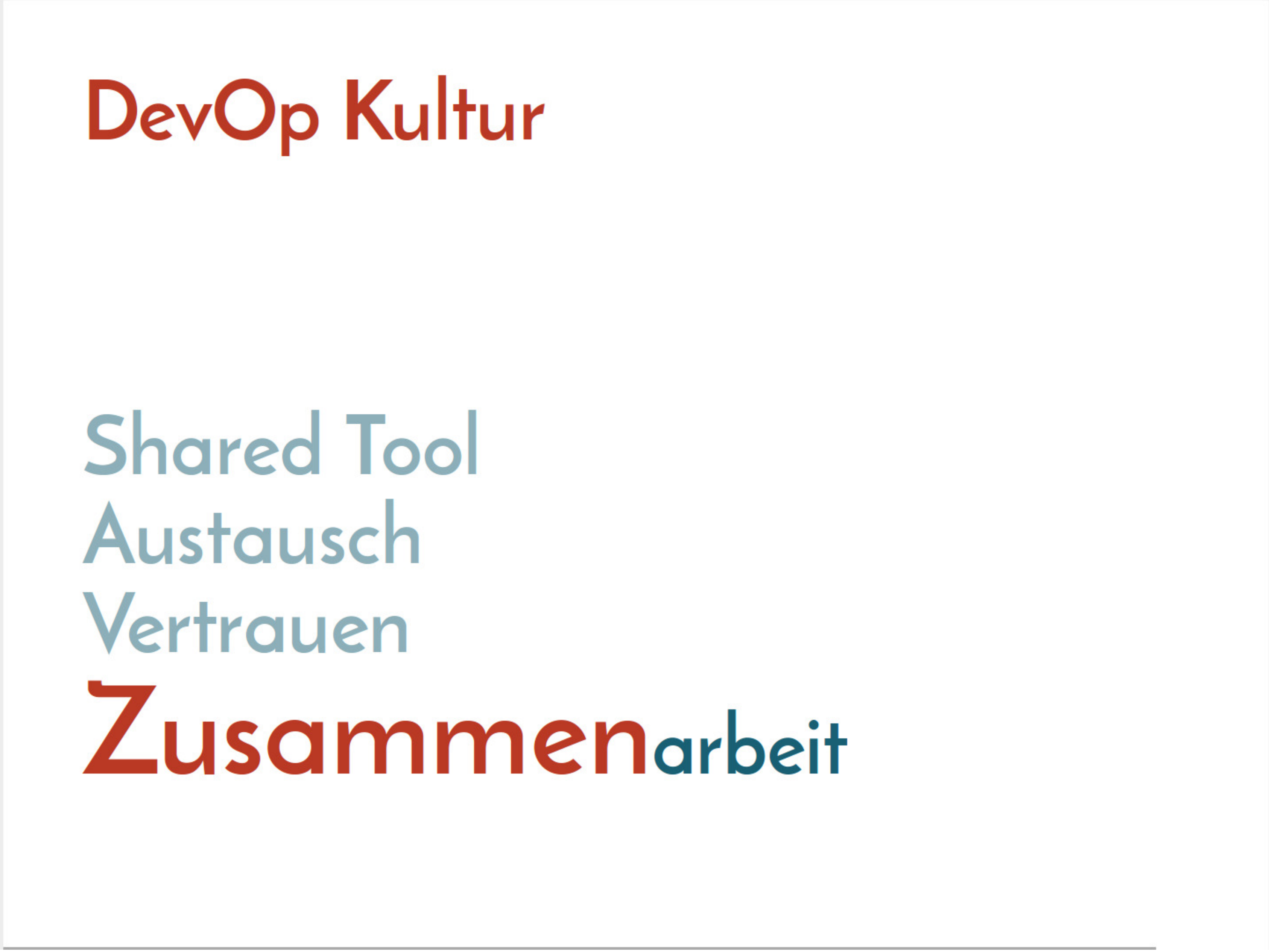
Gene Kim and Patrick Debois (CC BY-NC-SA) | Matt Moor

Development

Entwicklung = Produktion
Test der Konfiguration
Feedback vom Operating

Operating

Konfiguration ist getestet
direkte Übernahme
kein Spezialist notwendig
dokumentiert



DevOp Kultur

Shared Tool
Austausch
Vertrauen
Zusammenarbeit

Konfiguration ist Code

Github (CC BY-NC-SA) | Fumi Yamazaki

Versionsverwaltung

> *Code gehört unter Versionsverwaltung!*

Qualitätssicherung

(Unit-)Tests
Codeanalyse
Refactoring

Fragen?