HOCHSCHULE DER MEDIEN

Vortrag

Infrastructure as Code

System Engineering and Management

Studiengang Computer Science and Media

Korbinian Kram kk114@hdm-stuttgart.de
Simon Lipke sl110@hdm-stuttgart.de
Patrik Jakober pj010@hdm-stuttgart.de
Sebastian Löw sl106@hdm-stuttgart.de
Steffen Hinderer sh219@hdm-stuttgart.de

Agenda

- Vagrant
- Chef
- Puppet
- Demo der Projekte

Was wollen wir erreichen

- Einfaches Konfigurieren von Servern
  - Schnelles Bereitstellen
  - Wenig Kenntnisse von Systemadministration nötig
  - Unterstützung von Debian basierten Systemen
  - VirtualBox als Virtualisierungsumgebung

Was ist entstanden

- Vagrant Files
- Chef Recipes
- Puppet Manifests
- README als Anleitung

Was haben wir gemacht (1)

- Out of the Box Systeme mit:
  - LAMP/LEMP
    - HTTP-Server, MySQL-Server, PHP Integration
  - Java Application Server
    - JBoss
    - Tomcat
  - Ruby on Rails

Was haben wir gemacht (2)

- Absichern der Systeme für produktiven Einsatz (1)
  - SSH hardening
    - No Root login
    - Port
    - Password Authentication
    - User Management
      - User / System-User
      - Sudo
      - Passwort
      - Private keys hinterlegen

Was haben wir gemacht (3)

- Absichern der Systeme für produktiven Einsatz (2)
  - MySQL hardening
    - Root Password setzten/ändern
    - Disallow remote root login
    - Test Database löschen
    - Anonyme User löschen

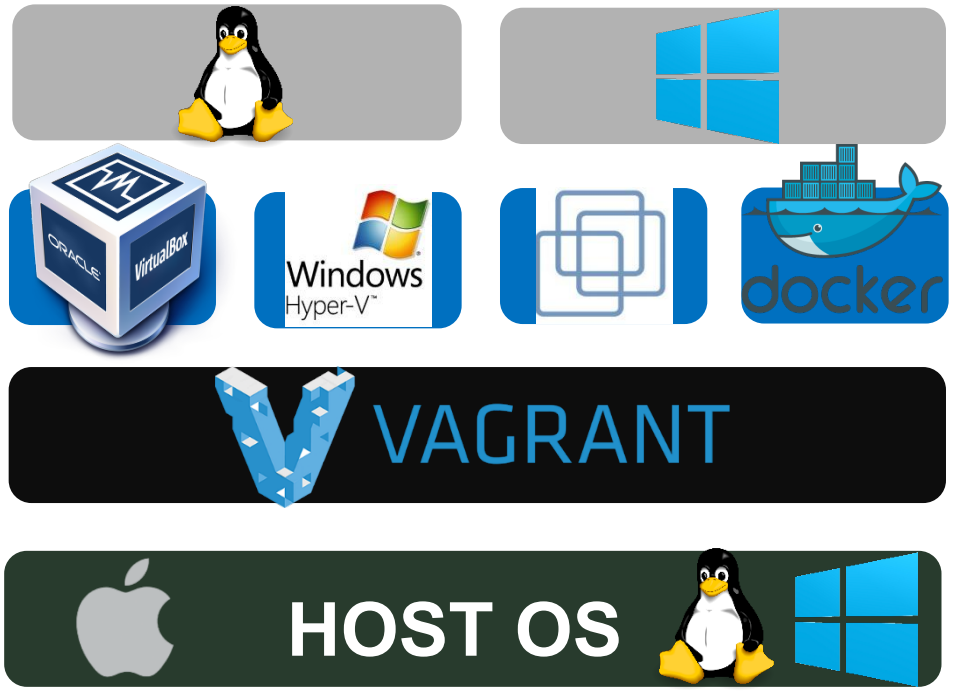
Was haben wir gemacht (4)

- Absichern der Systeme für produktiven Einsatz (3)
  - NGINX hardening
    - Performance-Optimierung
    - Servertokens ausschalten
    - User anpassen
    - HTTP Header anpassen
    - Default site überschreiben/deaktivieren

Vagrant

Über Vagrant

- Initiiert von Mitchel Hashimoto (2010)
- Open Source

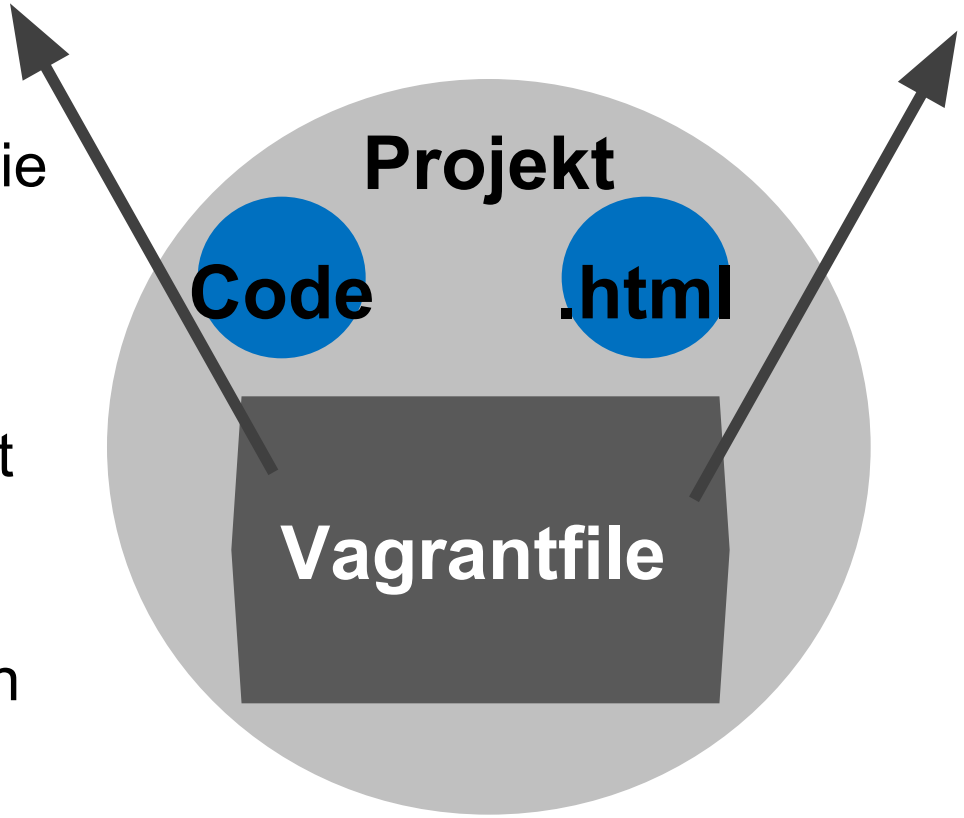


Vorteile von Vagrant

- Schnelle Entwicklungszyklen
- Gleiche Testumgebung
- Gleiche Abhängigkeiten in den Umgebungen
- Gleiche Versionen von Abhängigkeiten
- Keine Ausreden „auf meinem System hat es funktioniert“
- Konsistenter Workflow zum Testen und Entwickeln
- Lokales Testen bevor man in die Cloud geht
- Entlastung der Testabteilung/Admins

Arbeiten mit Vagrant

- Ruby
- Definiert die Art der Maschine
- Versioniert
- Sucht im kompletten Pfad



- Definiert Konfiguration der Maschine
- Definiert Konfiguration des Containers

- /home/mitchellh/projects/foo/Vagrantfile

Basis Konfiguration des Containers / Maschine

ubuntu/trusty64
chef/centos-6.5
Windows 7

```
Vagrant.configure(2) do |config|
    config.vm.box = "hashicorp/precise32“
    config.vm.network "private_network", ip:
  "192.168.33.46"
   config.vm.hostname = "puppet-jboss“
end
```

Konfiguration der Guest Maschine

- File
  - config.vm.provision "file", source: "~/.gitconfig", destination: ".gitconfig"
- Shell
  - config.vm.provision "shell", path: "script.sh“
- Provision – Software
  - Ansible, CFEngine, **Chef**, **Puppet**, Docker, Salt
- Synced Folder
  - config.vm.synced_folder "src/", "/srv/website"

Konfiguration der Guest Maschine

- Chef

```
config.vm.provision "chef_solo" do |chef|
    chef.roles_path = "roles"
    chef.add_role("jboss")
 end
```

```
|-- vagrantfile
|-- attributes
|-- roles
|-- |-- jboss.rb
|-- recipes
|-- templates
```

- Puppet

```
config.vm.provision :puppet do |puppet|
    puppet.module_path = ["modules","puppet/modules" ]
    puppet.options = ['--verbose']
    puppet.manifests_path = "puppet/manifests"
    puppet.manifest_file = "site.pp„
end
```

```
|-- vagrantfile
|-- puppet
|-- |-- manifests
|-- |-- modules
|-- |-- |-- jboss-config
```

Wichtige Commands

- vagrant init
  - Anlegen eines Standard Vagrant Files
- vagrant up
  - Starten der Maschine und durchlaufen des Provisioning
- vagrant provison
  - Auf laufender Maschine die Provisioning Dateien neu einlesen und ausführen
- vagrant provision --provision-with shell
  - Nur das Provisioning für die Shell erneut ausführen
- vagrant ssh
  - Login über ssh auf der Guest Maschine

Wichtige Commands

- vagrant rdp
  - Windows Pendant zu ssh
- vagrant suspend
  - Standby der Maschine
- vagrant halt
  - Herunterfahren der Maschine
- vagrant destroy
  - Herunterfahren und Löschen aller Dateien, die während des Startens erzeugt wurden

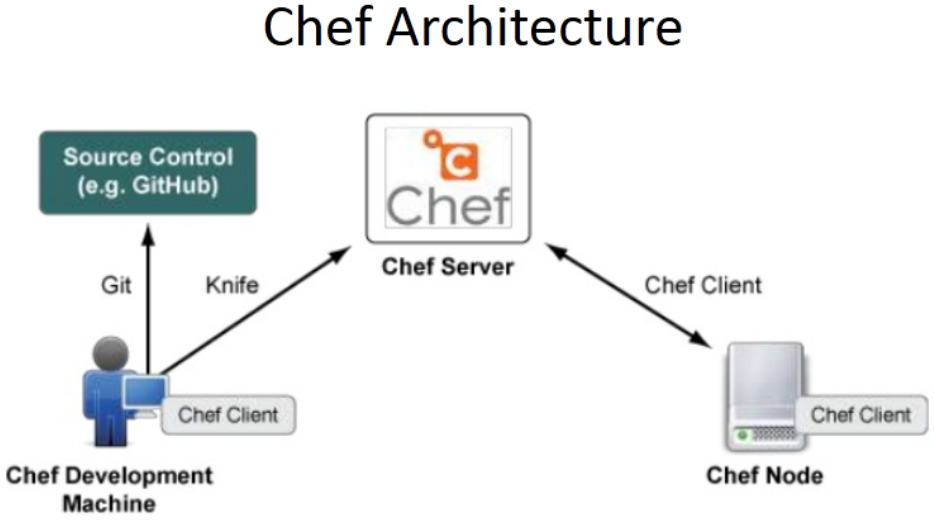
Chef

Was ist Chef?

- *„Chef is a powerful automation platform that transforms complex infrastructure into code”* **(http://docs.chef.io/chef_overview.html)**
- **Open Source Projekt**
- **Erster Commit März 2008**
- **Geschrieben in Ruby und Erlang**

Architektur

- **Resource - Recipe - Cookbook - Role**
- **Templates / Databags**
- **Chef Server / Chef Node**
- **Chef-solo**




```
package 'apache2'

service 'apache2' do
   action [:enable, :start]
end

file '/var/www/html/index.html' do
  content '<html>
  <body>
    <h1>hello world</h1>
  </body>
</html>'
end
```

Tools

- ChefDK
  - Knife
  - Kitchen
  - Berkshelf / librarian-chef

https://supermarket.chef.io/cookbooks?order=most_downloaded

2291 Cookbooks RSS

Sort by **Most Downloaded** | Most Followed | Recently Updated

mysql 6.0.23 Updated June 21, 2015

chef

Provides mysql_service, mysql_config, and mysql_client resources

```
cookbook 'mysql', '~> 6.0.23'
```

SUPPORTED PLATFORMS

106,905,803 TOTAL DOWNLOADS 557 FOLLOWERS Follow

java 1.31.0 Updated June 2, 2015

agileorbit

Installs Java runtime.

```
cookbook 'java', '~> 1.31.0'
```

SUPPORTED PLATFORMS

64,161,856 TOTAL DOWNLOADS 297 FOLLOWERS Follow

apache2 3.1.0 Updated May 25, 2015

svanzoest

Installs and configures all aspects of apache2 using Debian style symlinks with helper definitions

```
cookbook 'apache2', '~> 3.1.0'
```

Puppet

Studiengang: Computer Science and Media, Autor: Simon Lipke, Vorlesung:System Engineering and Management, E-Mail: sl110@hdm-stuttgart.de

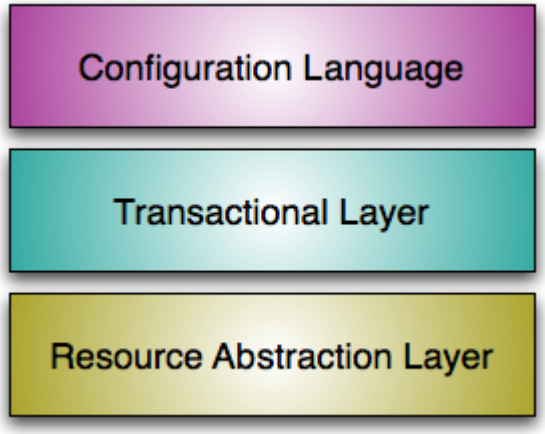
Was ist Puppet?

Automatisierung und Verwaltung von IT-Infrastruktur

	Open Source Puppet	Puppet Enterprise
Puppet Language (DSL)	✔	✔
PuppetDB	✔	✔
Puppet Server	✔	✔
Puppet Apps (PNM, PCM)	-	✔
Commercial only Features (Reporting, RBAC)	-	✔
Modules & Integration	-	✔
Enterprise Support	-	✔

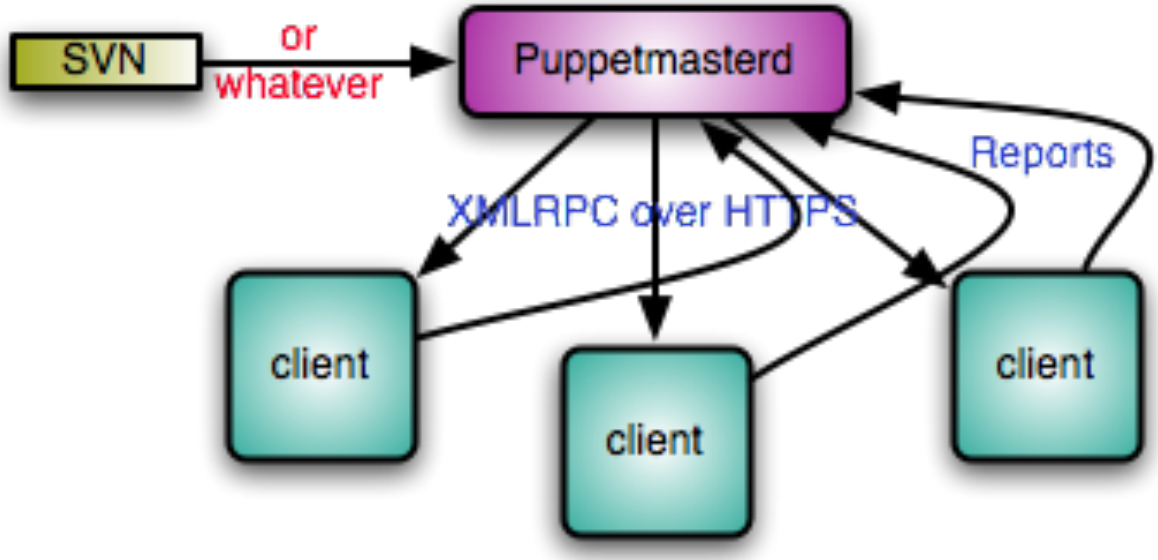
Vorteile von Puppet

- Idempotent
- Multi-Plattform
  - Linux (RHEL, Ubuntu, …)
  - Windows
  - Mac
- Einfaches Teilen von Puppet-Code
- Deklarativ



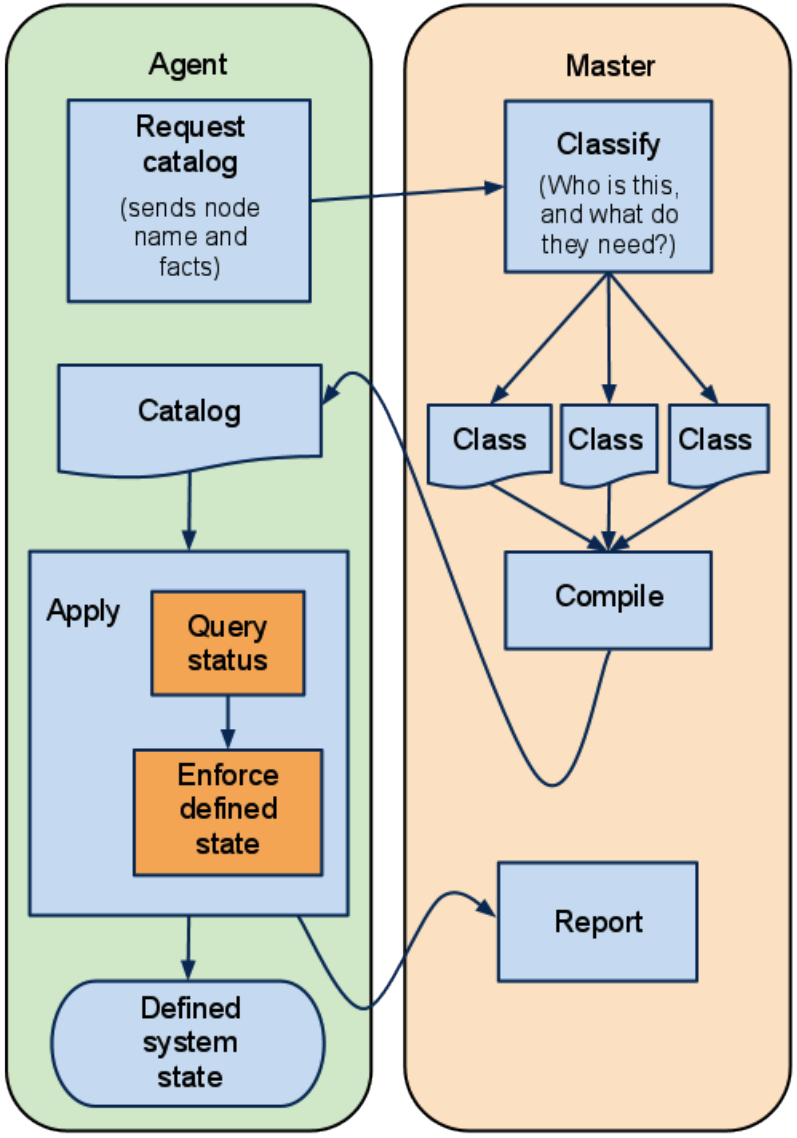
Architektur

- Agent/Master (Client/Server) Architektur



- Puppet Apply

Funktionsweise



Komponenten

- Ressourcen

```
# A resource declaration:
file { '/etc/passwd':
  ensure => file,
  owner  => 'root',
  group  => 'root',
  mode   => '0600',
}
```

Komponenten

- Ressourcen

```
package { 'openssh-server':
    ensure => installed,
}

file { '/etc/ssh/sshd_config':
    source  => 'puppet:///modules/sshd/
sshd_config',
    owner   => 'root',
    group   => 'root',
    mode    => '640',
    notify  => Service['sshd'], # sshd
               will restart whenever you
               edit this file.
    require => Package['openssh-server'],
}

service { 'sshd':
    ensure => running,
    enable => true,
    hasstatus => true,
    hasrestart => true,
}
```

Komponenten

- System-Typen
  - **package**
  - **service**
  - **file**
  - cron
  - user
  - group
  - ...

Komponenten

- Klassen

```
# A class with no parameters
class base::linux {
  file { '/etc/passwd':
    owner => 'root',
    group => 'root',
    mode  => '0644',
  }
  file { '/etc/shadow':
    owner => 'root',
    group => 'root',
    mode  => '0440',
  }
}
```

Komponenten

- Klassen mit Parametern

```
# A class with parameters
class apache (String $version = 'latest') {
  package {'httpd':
    ensure => $version, # Using the class parameter from above
    before => File['/etc/httpd.conf'],
  }
  file {'/etc/httpd.conf':
    ensure  => file,
    owner   => 'httpd',
    content => template('apache/httpd.conf.erb'), # Template from a module
  }
  service {'httpd':
    ensure    => running,
    enable    => true,
    subscribe => File['/etc/httpd.conf'],
  }
}
```

Komponenten

- Nodes (Optional)

```
# /etc/puppetlabs/puppet/manifests/site.pp
node 'www1.example.com' {
  include common
  include apache
  include squid
}
node 'db1.example.com' {
  include common
  include mysql
}
```

Komponenten

- Facter

```
MacBook-Pro:puppet simon$ facter
architecture => x86_64
facterversion => 2.4.3
fqdn => MacBook-Pro
gid => staff
hardwareisa => i386
hardwaremodel => x86_64
hostname => MacBook-Pro
id => simon
interfaces => lo0,gif0,stf0,en0,en1,fw0,en2,p2p0,bridge0,vboxnet0
ipaddress => 192.168.1.224
ipaddress_en1 => 192.168.1.224
ipaddress_lo0 => 127.0.0.1
ipaddress_vboxnet0 => 192.168.33.1
is_virtual => false
kernel => Darwin
kernelmajversion => 14.3
kernelrelease => 14.3.0
kernelversion => 14.3.0
macaddress => 60:c5:47:8d:00:c0
macaddress_bridge0 => 3e:07:54:02:34:00
macaddress_en0 => 3c:07:54:20:ee:da
macaddress_en1 => 60:c5:47:8d:00:c0
macaddress_en2 => d2:00:14:e3:bf:c0
macaddress_fw0 => a4:b1:97:ff:fe:4e
```

Module (http://docs.puppetlabs.com/module_cheat_sheet.pdf)

Beispiel: <modules>/apache:

```
|-- apache
        |-- manifests
                |-- init.pp (enthält Klasse apache)
                |-- vhost.pp (enthält Klasse apache::vhost)
        |-- files
                |-- httpd.conf
        |-- lib
        |-- templates
                |-- vhost.erb (Ruby ERB Templates)
```

Module & Erweiterungen

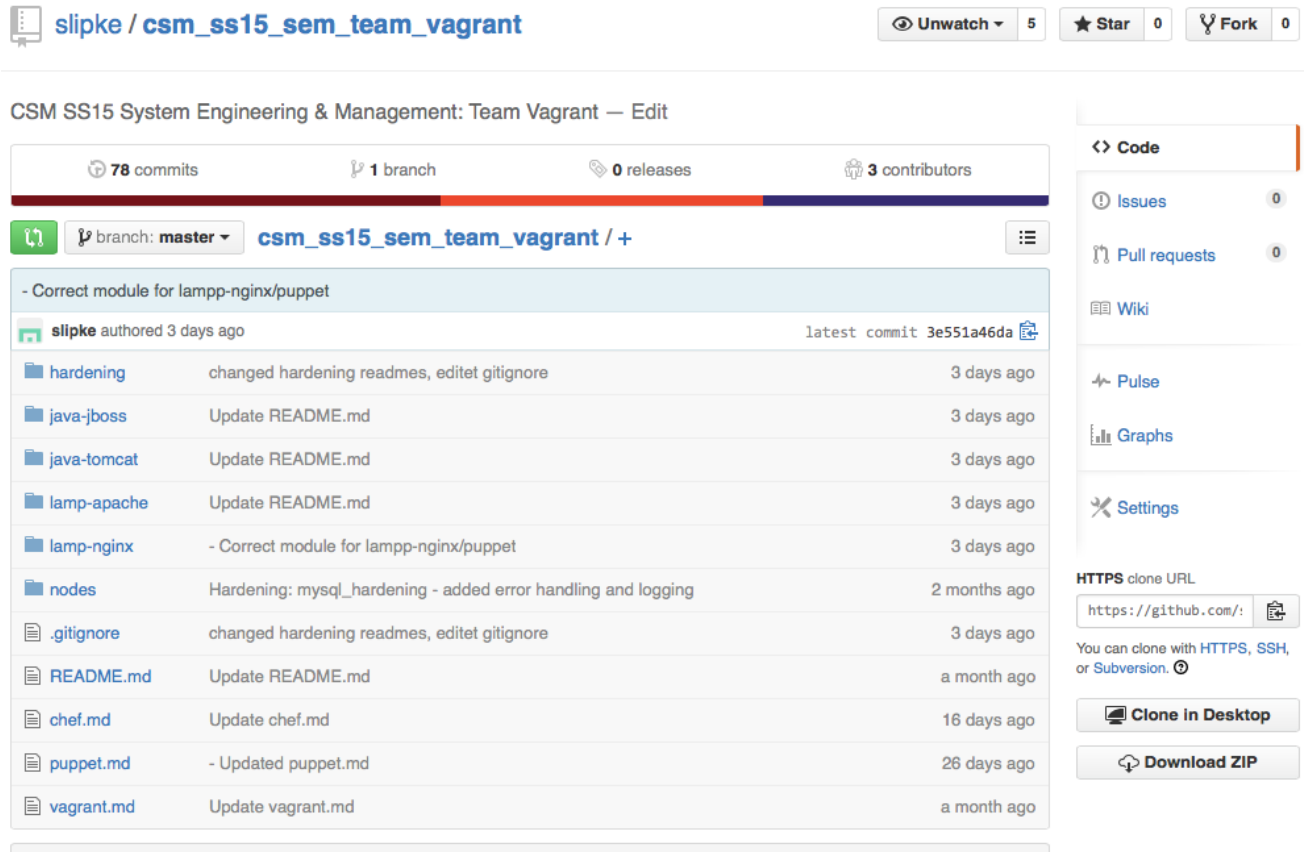
- Puppet forge (https://forge.puppetlabs.com)

- GitHub (https://github.com)

Demo der Projekte

Unser Projekt auf GitHub

https://github.com/slipke/csm_ss15_sem_team_vagrant



Quellen

- www.puppetlabs.com
- http://docs.chef.io/chef_overview.html
- https://github.com/chef/chef
- https://docs.vagrantup.com/v2/