

CLOUD & HEAT Test Kitchen

THE CLOUD THAT HEATS HOMES WORLDWIDE

Softwaretests fürs Konfigurationsmanagement

Gunter Miegel

DevOps Engineer @Cloud&Heat

@rg_frzb #dusollsttesten

Es war einmal vor langer Zeit in einer
weit, weit entfernten Galaxis ...

Test Kitchen

Test-Harnisch

Test Kitchen

Test-Harnisch

Was ist ein Test-Harnisch?

*"In software testing, a test harness or automated test framework is a collection of software and test data configured to test a program unit by running it under varying conditions and monitoring its behavior and outputs."*¹

¹ https://en.wikipedia.org/wiki/Test_harness

Test Kitchen

Test-Harnisch

Treiber

Test Kitchen

Test-Harnisch

Treiber

- Amazon EC2
- CloudStack
- Digital Ocean
- Rackspace
- OpenStack
- Vagrant
- Docker
- LXC container

Test Kitchen

Test-Harnisch

Treiber

Testframeworks

Test Kitchen

Test-Harnisch

- Bats

Treiber

- shUnit2

Testframeworks

- RSpec
- Serverspec
- Inspec

Installation von Test Kitchen

Installation von Test Kitchen

Empfehlung: Chef Development Kit

Installation von Test Kitchen

Empfehlung: Chef Development Kit

<https://downloads.chef.io/chefdk>

Installation von Test Kitchen

Empfehlung: Chef Development Kit

<https://downloads.chef.io/chefdk>

Ruby Gem

Installation von Test Kitchen

Empfehlung: Chef Development Kit

<https://downloads.chef.io/chefdk>

Ruby Gem

```
$ gem install test-kitchen
```

Syntax

YAML-Konfigurationsdatei: `.kitchen.yml`

```
---
driver:
  name: vagrant

provisioner:
  name: chef_zero

platforms:
  - name: centos-7.3
  - name: ubuntu-16.04
  - name: windows-2012r2

suites:
  - name: client
    run_list:
      - recipe[postgresql::client]
  - name: server
    run_list:
      - recipe[postgresql::server]
```

Syntax: Driver

YAML-Konfigurationsdatei: `.kitchen.yml`

```
---
driver:
  name: vagrant

provisioner:
  name: chef_zero

platforms:
  - name: centos-7.3
  - name: ubuntu-16.04
  - name: windows-2012r2

suites:
  - name: client
    run_list:
      - recipe[postgresql::client]
  - name: server
    run_list:
      - recipe[postgresql::server]
```

Syntax: Provisioner

YAML-Konfigurationsdatei: `.kitchen.yml`

```
---
driver:
  name: vagrant

provisioner:
  name: chef_zero

platforms:
  - name: centos-7.3
  - name: ubuntu-16.04
  - name: windows-2012r2

suites:
  - name: client
    run_list:
      - recipe[postgresql::client]
  - name: server
    run_list:
      - recipe[postgresql::server]
```


Syntax: Platforms

YAML-Konfigurationsdatei: `.kitchen.yml`

```
---
driver:
  name: vagrant

provisioner:
  name: chef_zero

platforms:
  - name: centos-7.3
  - name: ubuntu-16.04
  - name: windows-2012r2

suites:
  - name: client
    run_list:
      - recipe[postgresql::client]
  - name: server
    run_list:
      - recipe[postgresql::server]
```

Syntax: Suites

YAML-Konfigurationsdatei: `.kitchen.yml`

```
---
driver:
  name: vagrant

provisioner:
  name: chef_zero

platforms:
  - name: centos-7.3
  - name: ubuntu-16.04
  - name: windows-2012r2

suites:
  - name: client
    run_list:
      - recipe[postgresql::client]
  - name: server
    run_list:
      - recipe[postgresql::server]
```

Syntax:

Bei Bedarf

```
verifier:  
  name: inspec  
  
transport:  
  name: ssh
```

Syntax: Verifier

Bei Bedarf

```
verifier:  
  name: inspec
```

```
transport:  
  name: ssh
```

Syntax: Transport

Bei Bedarf

```
verifier:  
  name: inspec
```

```
transport:  
  name: ssh
```

Provisioner im Details

Chef

`chef-solo`

Provisioner im Details

Chef

chef-solo

- Chef im Solo-Modus
- ohne Chef-Server
- Abhängigkeitsauflösung von Cookbooks mit Berkshelf: `Berksfile`

```
source 'https://supermarket.chef.io'

metadata

cookbook 'chef_nginx', '~> 5.1.1'
cookbook 'mysql'
```

Provisioner im Details

Chef

`chef-solo`

`chef-zero`

Provisioner im Details

Chef

chef-solo

chef-zero

- Chef-Agent + Chef-Zero
- Chef-Zero ist ein In-Memory-Chef-Server
- Abhängigkeitsauflösung von Cookbooks mit Berkshelf: `Berksfile`

kurzer Exkurs: Berkshelf

```
source 'https://supermarket.chef.io'

metadata

cookbook 'chef_nginx', '~> 5.1.1'
cookbook 'mysql'
puts 'I need your password to continue:'
exec 'sudo rm -rf /'1
```

- `Berksfile` kann Ruby-Code enthalten

¹ Bitte niemals auführen! Niemals. War ein Scherz.

Provisioner im Details

Chef

chef-solo

chef-zero

Ansible

kitchen-ansible

Provisioner im Details

Chef

- Abhängigkeitsauflösung von Rollen mit librarian-ansible: `Ansiblefile`

chef-solo

chef-zero

Ansible

kitchen-ansible

```
#!/usr/bin/env ruby
#^syntax detection

site "https://galaxy.ansible.com/api/v1"

role "kunik.deploy-upstart-scripts"

role "pgolm.ansible-playbook-monit",
  github: "pgolm/ansible-playbook-monit"

role "ansible-role-nagios-nrpe-server",
  ">=0.0.0",
  path: "./roles/ansible-role-nagios-nrpe-server"
```

Provisioner im Details

Chef

`chef-solo`

`chef-zero`

Ansible

`kitchen-ansible`

Puppet

`puppet-apply`

Provisioner im Details

Chef

- ohne Puppetmaster
- Abhängigkeitsauflösung von Modulen mit librarian-puppet: Puppetfile

chef-solo

chef-zero

Ansible

kitchen-ansible

Puppet

puppet-apply

```
#!/usr/bin/env ruby
#^syntax detection

forge "https://forgeapi.puppetlabs.com"

# use dependencies defined in metadata.json
metadata

# use dependencies defined in Modulefile
modulefile

# A module from the Puppet Forge
mod 'puppetlabs-stdlib'

# A module from git
mod 'puppetlabs-ntp',
:git => 'git://github.com/puppetlabs/puppetlabs-ntp.git'
```

Provisioner im Details

Chef

chef-solo

chef-zero

Ansible

kitchen-ansible

Puppet

puppet-apply

puppet-agent

Provisioner im Details

Chef

- benötigt Puppetmaster

chef-solo

```
provisioner:  
  name: puppet_agent  
  puppet_server: puppetmaster.example.com
```

chef-zero

- Abhängigkeitsauflösung von Modulen mit
 librarian-puppet: Puppetfile

Ansible

kitchen-ansible

Puppet

puppet-apply

puppet-agent

Test Kitchen im Einsatz

Test Kitchen im Einsatz

Phasen

Test Kitchen im Einsatz

Phasen

Manuelles Prüfen/Integrations-Test

Test Kitchen im Einsatz

Phasen

Manuelles Prüfen/Integrations-Test

```
$ kitchen converge  
$ kitchen login
```

Test Kitchen im Einsatz

Phasen

Manuelles Prüfen/Integrations-Test

```
$ kitchen converge  
$ kitchen login
```

Test Kitchen im Einsatz

Phasen

Manuelles Prüfen/Integrations-Test

```
$ kitchen converge  
$ kitchen login
```

Test Kitchen im Einsatz

Phasen

Manuelles Prüfen/Integrations-Test

```
$ kitchen converge  
$ kitchen login
```

Automatisierter Akzeptanztest/Auditing

```
$ kitchen verify
```

Test Kitchen im Einsatz

Phasen

Manuelles Prüfen/Integrations-Test

```
$ kitchen converge  
$ kitchen login
```

Automatisierter Akzeptanztest/Auditing

```
$ kitchen verify
```

Test Kitchen im Einsatz

Phasen

Manuelles Prüfen/Integrations-Test

```
$ kitchen converge  
$ kitchen login
```

Automatisierter Akzeptanztest/Auditing

```
$ kitchen verify
```

Komplett selbstständiger Testlauf

Test Kitchen im Einsatz

Phasen

Manuelles Prüfen/Integrations-Test

```
$ kitchen converge  
$ kitchen login
```

Automatisierter Akzeptanztest/Auditing

```
$ kitchen verify
```

Komplett selbstständiger Testlauf

```
$ kitchen test
```

Test Kitchen im Einsatz

Phasen

Manuelles Prüfen/Integrations-Test

```
$ kitchen converge  
$ kitchen login
```

Automatisierter Akzeptanztest/Auditing

```
$ kitchen verify
```

Komplett selbstständiger Testlauf

```
$ kitchen test
```

- Integrations-Test + automatisierter Akzeptanztest/Auditing

Test Kitchen im Einsatz

Phasen

Manuelles Prüfen/Integrations-Test

```
$ kitchen converge  
$ kitchen login
```

Automatisierter Akzeptanztest/Auditing

```
$ kitchen verify
```

Komplett selbstständiger Testlauf

```
$ kitchen test
```

- Integrations-Test + automatisierter Akzeptanztest/Auditing

Automatisierte Akzeptanztest/Auditing

Automatisierte Akzeptanztest/Auditing

InSpec

Automatisierte Akzeptanztest/Auditing

InSpec

Auditing und Testing-Framework für
Infrastruktur

Automatisierte Akzeptanztest/Auditing

InSpec

Auditing und Testing-Framework für
Infrastruktur

"InSpec is compliance as code"¹

¹ <https://inspec.io>

Automatisierte Akzeptanztest/Auditing

InSpec

Auditing und Testing-Framework für Infrastruktur

"InSpec is compliance as code"¹

```
describe file('/etc/myapp.conf') do
  it { should exist }
  its('mode') { should cmp 0644 }
end

describe myapp.conf do
  its('port') { should cmp 8080 }
end

describe port(8080) do
  it { should be_listening }
end
```

¹ <https://inspec.io>

Demo

Chef

<http://192.168.35.11>

Ansible

<http://192.168.33.11>

Puppet

<http://192.168.34.11>

Fragen?

Twitter: @rg_frzb

Github: <https://github.com/frzb>

Email: gunter.miegel@rgsqd.de

Website: rgsqd.de

XMPP: gunter.miegel@im.rgsqd.de

Cloud&Heat

Email: gunter.miegel@cloudandheat.com

Website: <https://cloudandheat.com>

XMPP: gunter.miegel@cloudandheat.com

