Infrastrukturtests mit der Jenkinsfile

Christopher J. Ruwe <cjr@cruwe.de>

23. März 2017

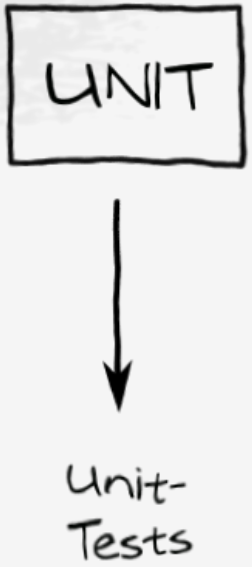
whoami

- Christopher J. Ruwe
- freiberuflicher IT-Infrastruktur Consultant (Unix, Netzwerke)
- DevOps Engineer and Consultant
- Linux, FreeBSD, Solaris

Christopher J. Ruwe
<cjr@cruwe.de>

Test-Systematik

Test-Systematik



Christopher J.
Ruwe
<cjr@cruwe.de>

Test-Systematik

Unit-Tests

isolierte Units:

1) puppet: module
2) ansible: role

Test-Systematik

Unit-Tests

isolierte Units:

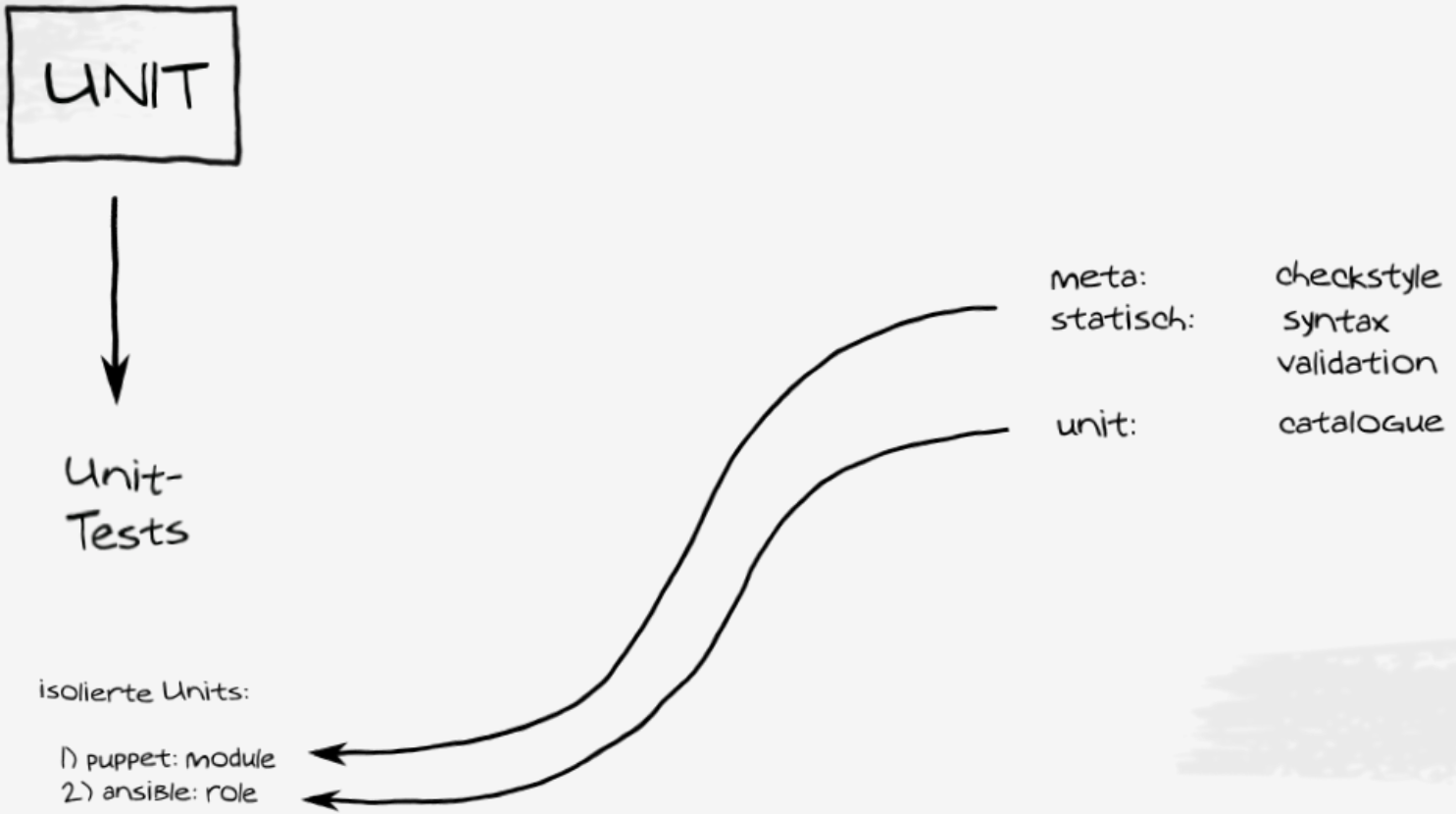
1) puppet: module
2) ansible: role

meta:
statisch:

checkstyle
syntax
validation

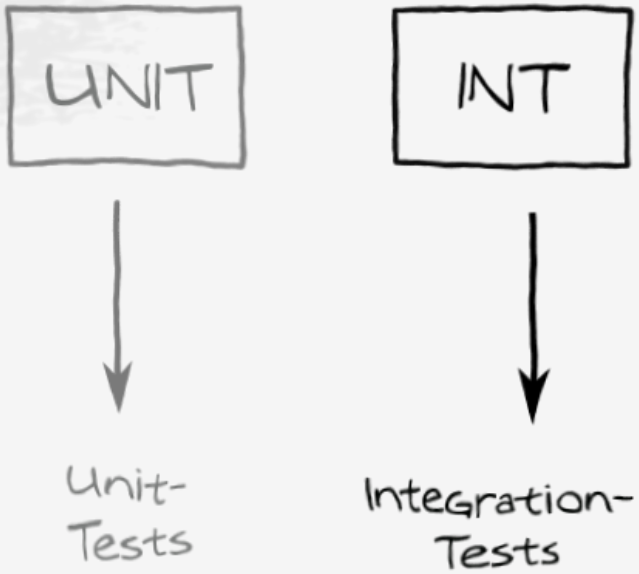
Christopher J. Ruwe
<cjr@cruwe.de>

Test-Systematik



Christopher J. Ruwe
<cjr@cruwe.de>

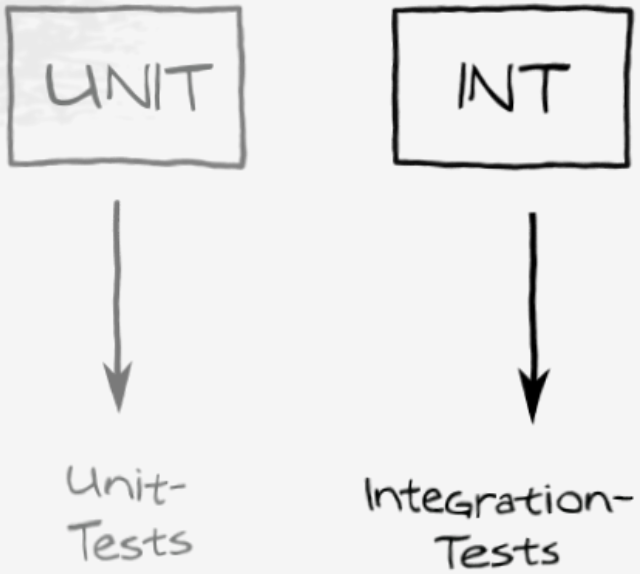
Test-Systematik



isolierte Units:

1) puppet: module
2) ansible: role

Test-Systematik



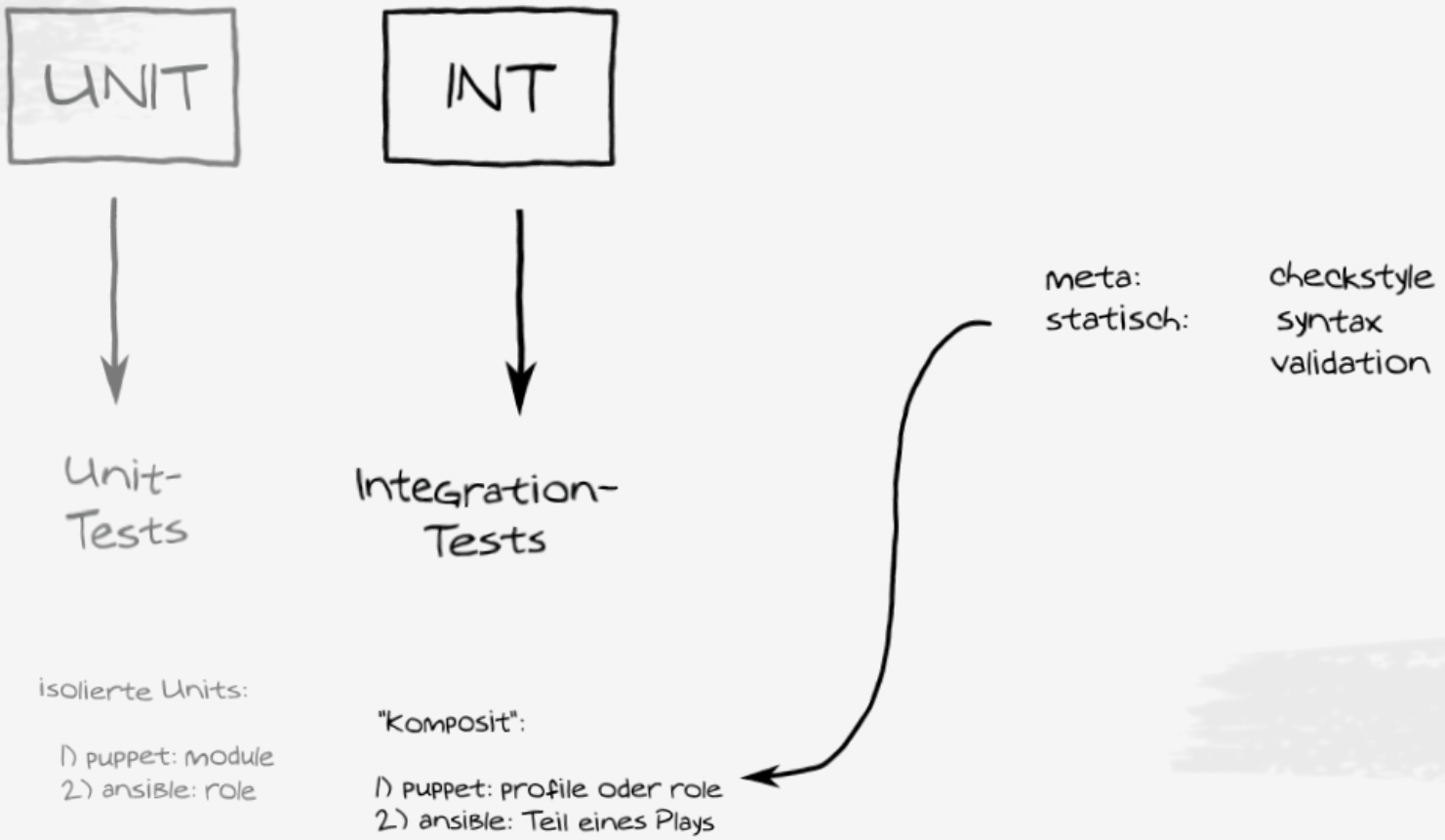
isolierte Units:

1) puppet: module
2) ansible: role

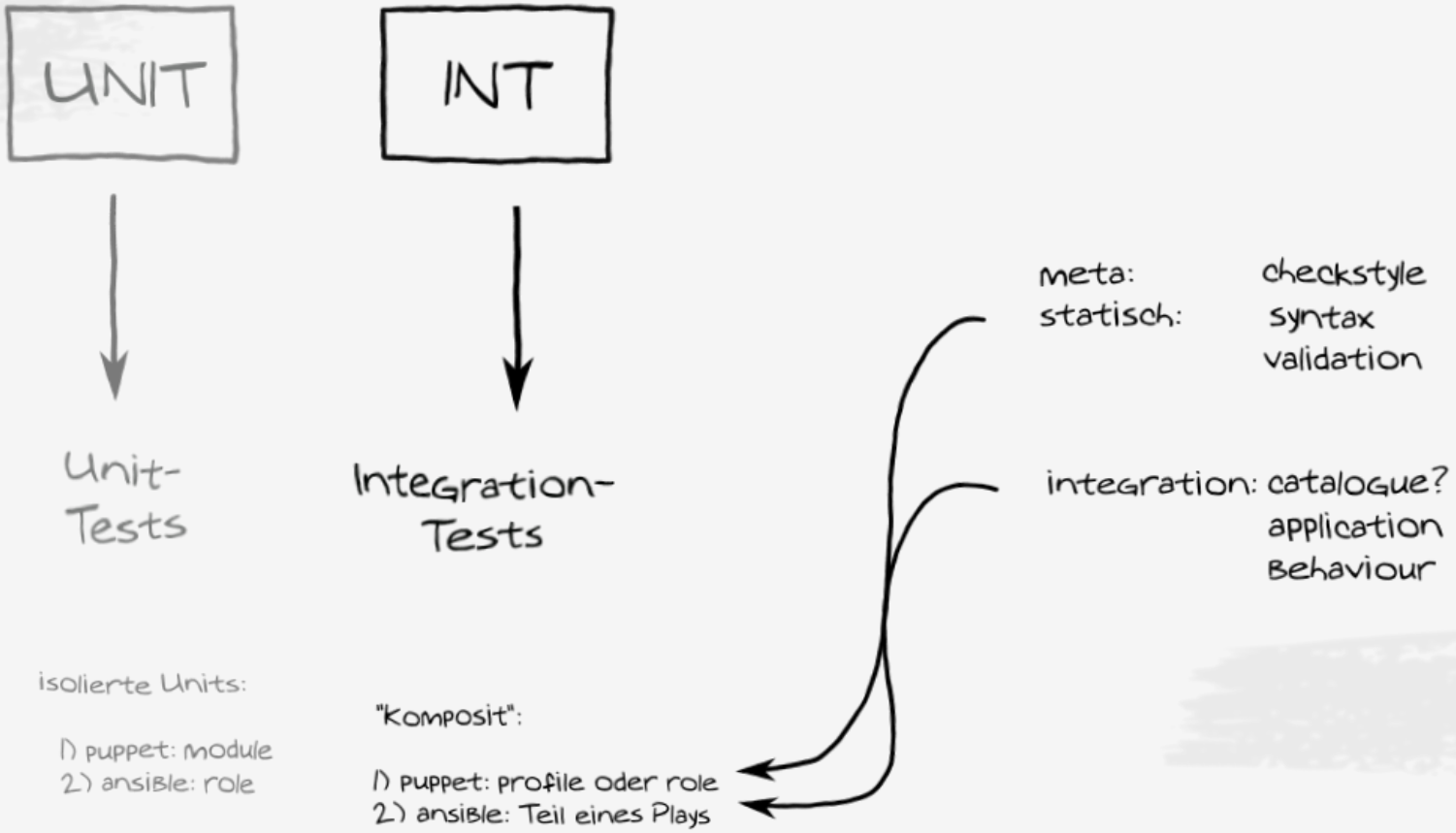
"Komposit":

1) puppet: profile oder role
2) ansible: Teil eines Plays

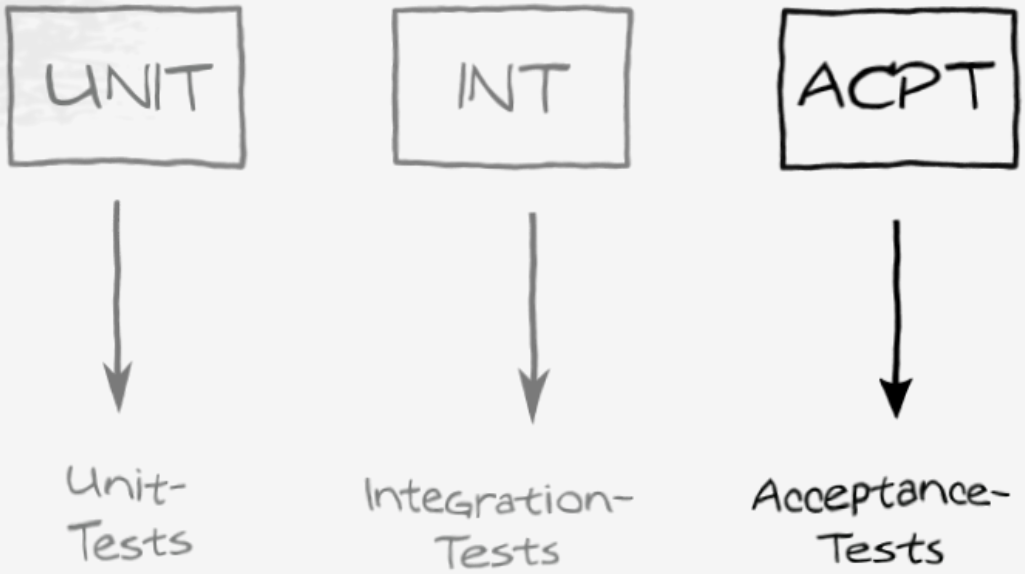
Test-Systematik



Test-Systematik



Test-Systematik



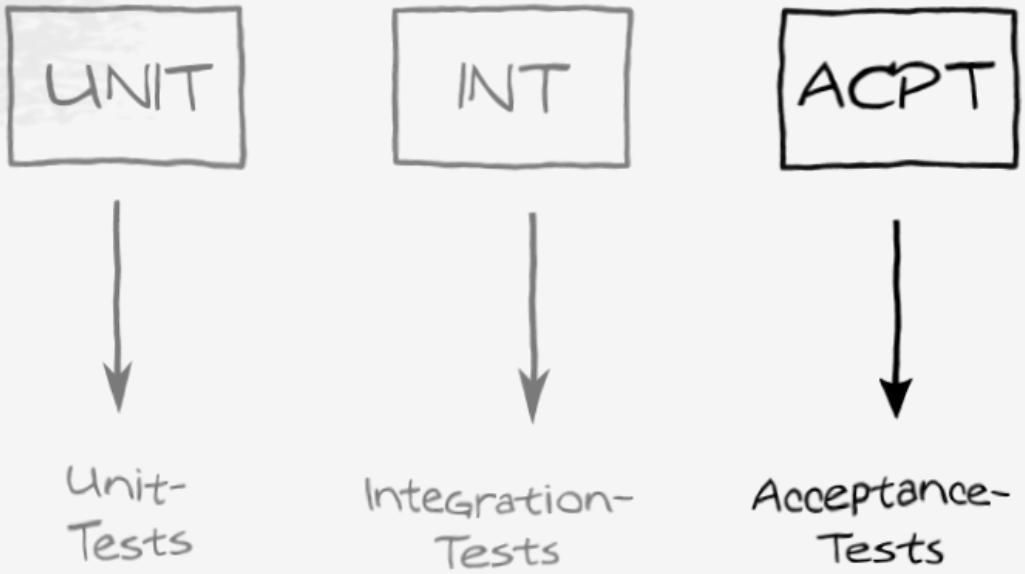
isolierte Units:

1) puppet: module
2) ansible: role

"Komposit":

1) puppet: profile oder role
2) ansible: Teil eines Plays

Test-Systematik



isolierte Units:

1) puppet: module
2) ansible: role

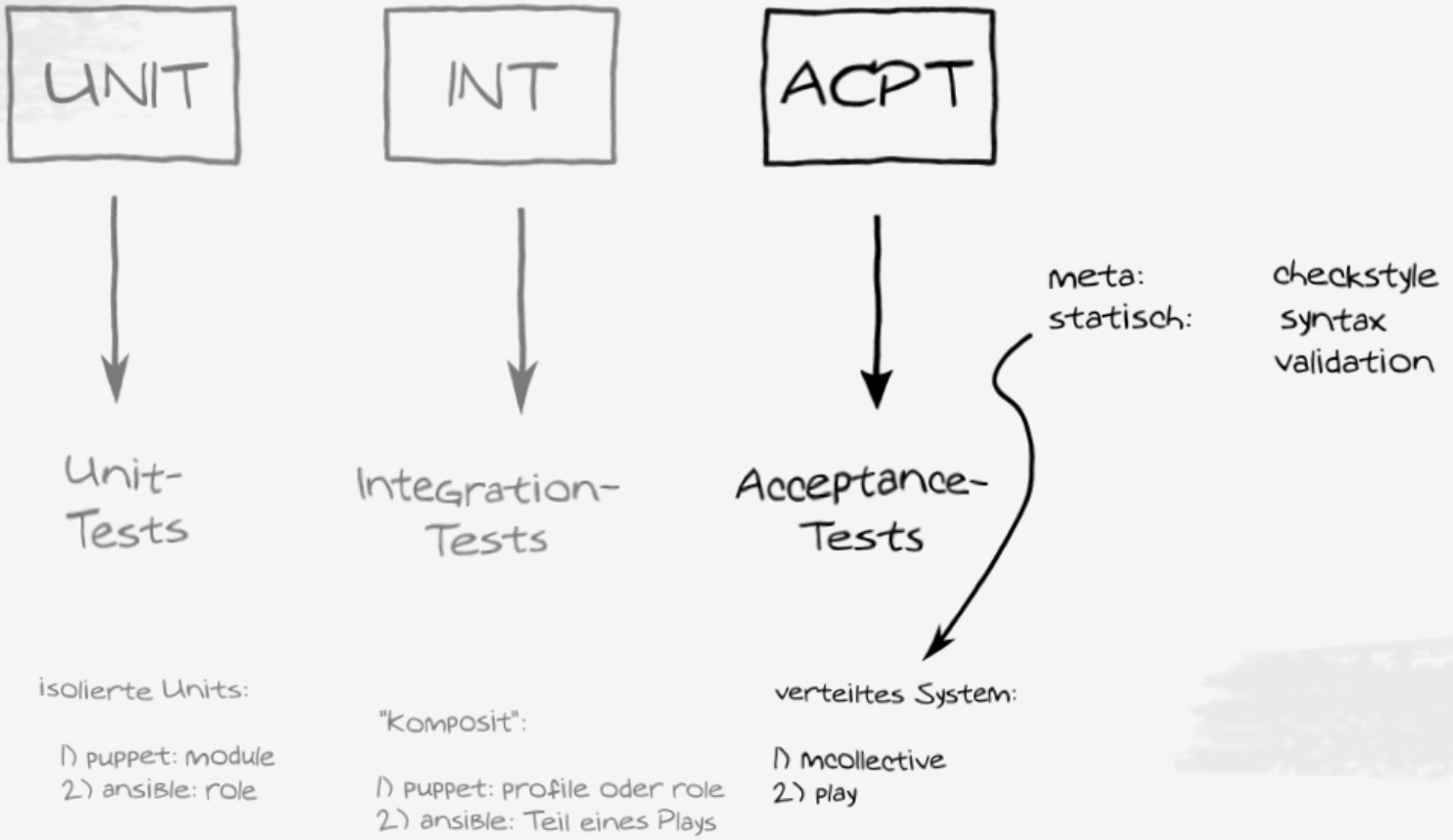
"Komposit":

1) puppet: profile oder role
2) ansible: Teil eines Plays

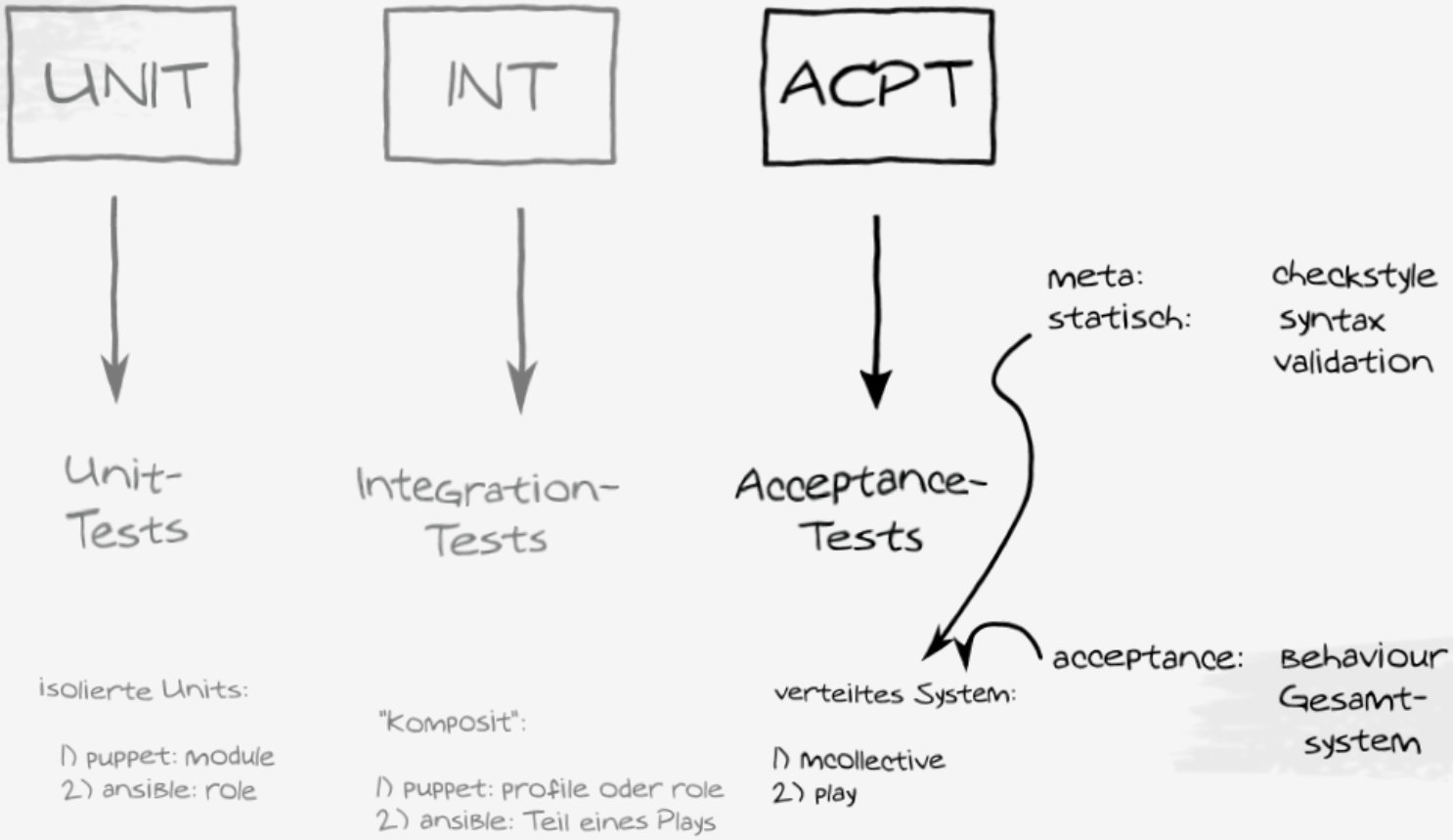
verteiltes System:

1) mcollective
2) play

Test-Systematik



Test-Systematik



Christopher J.
Ruwe
<cjr@cruwe.de>

Test-Systematik

Unit-
Tests

Integration-
Tests

Acceptance-
Tests

meta: checkstyle
statisch: syntax
validation

unit: catalogue

integration: catalogue?
application
Behaviour

acceptance: Behaviour
Gesamt-
system

isolierte Units:

1) puppet: module
2) ansible: role

"Komposit":

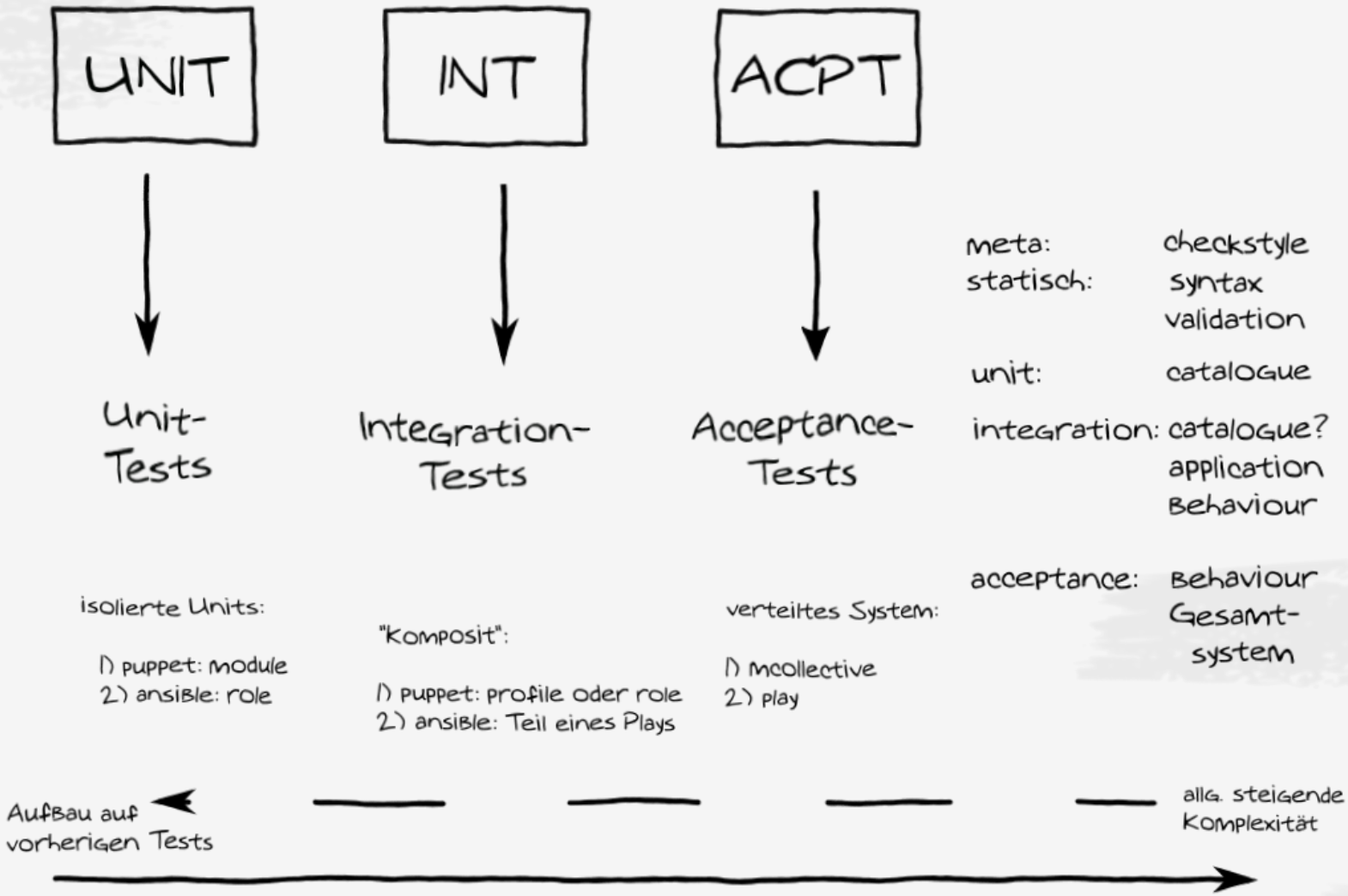
1) puppet: profile oder role
2) ansible: Teil eines Plays

verteiltes System:

1) mcollective
2) play

allg. steigende
Komplexität

Test-Systematik



Test-Systematik

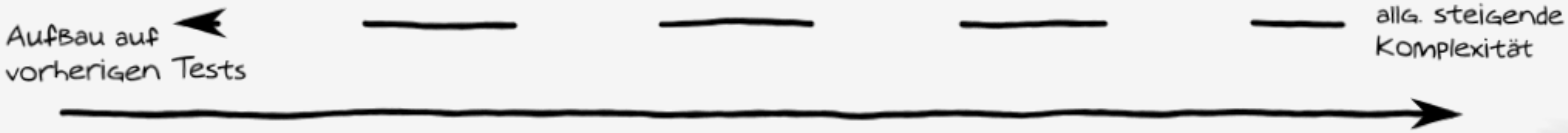
Christopher J. Ruwe
<cjr@cruwe.de>

Unit-Tests

isolierte Units:

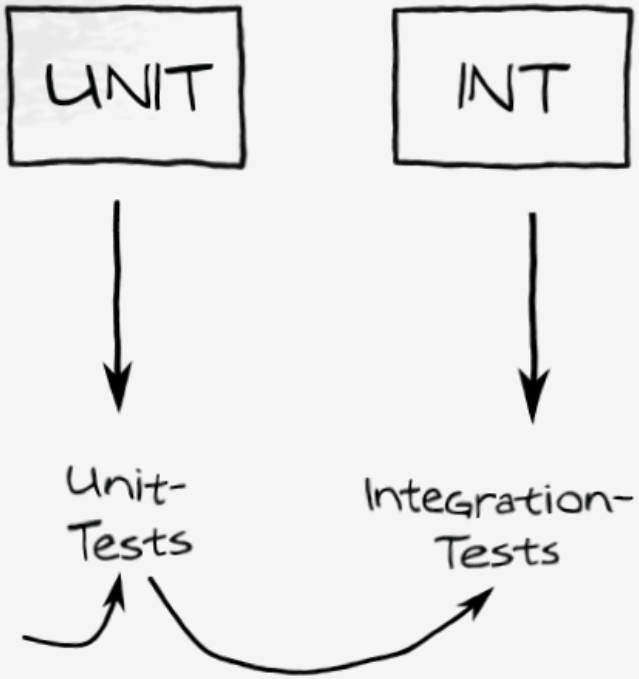
1) puppet: module
2) ansible: role

meta:	checkstyle
statisch:	syntax validation
unit:	catalogue



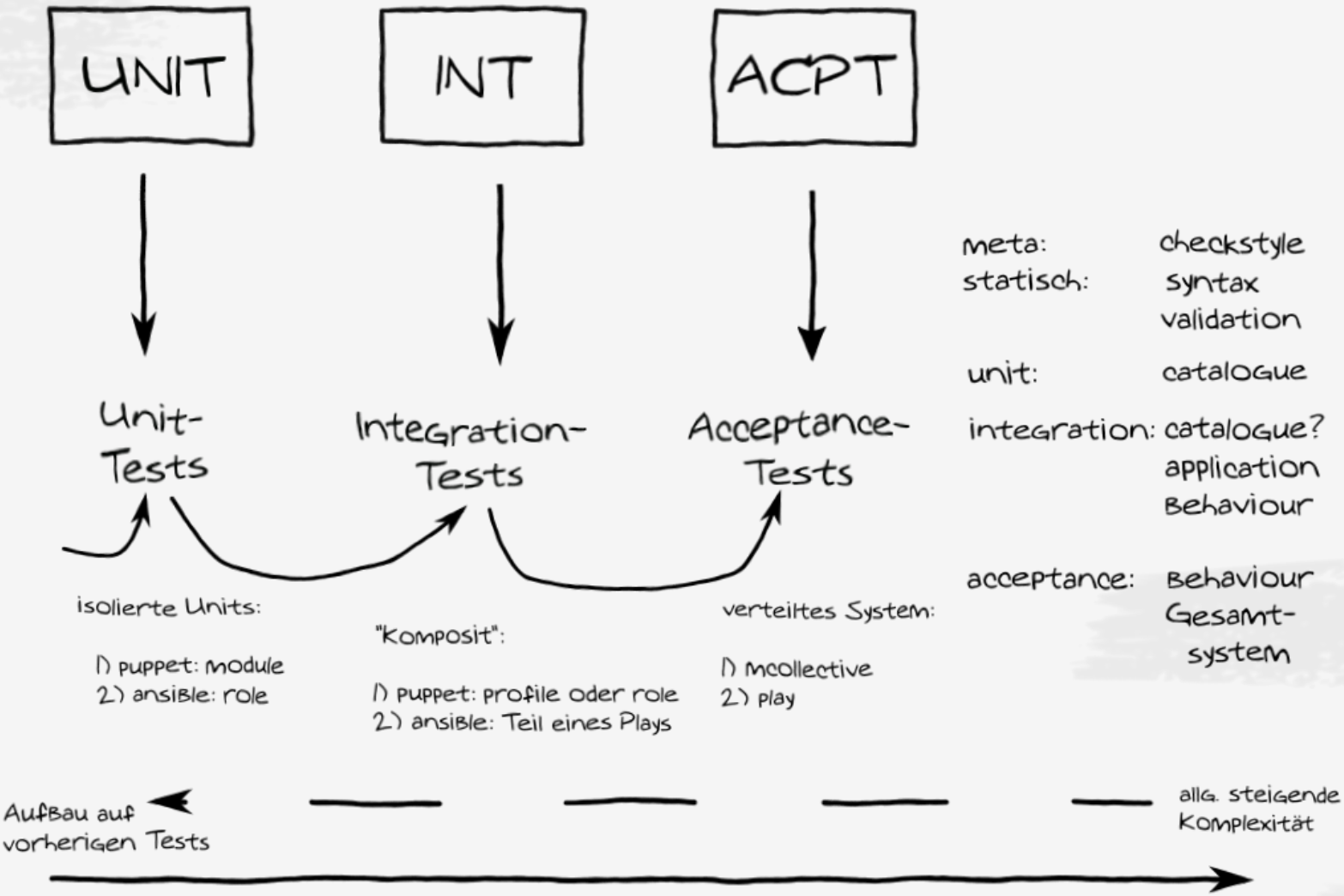
Christopher J.
Ruwe
<cjr@cruwe.de>

Test-Systematik



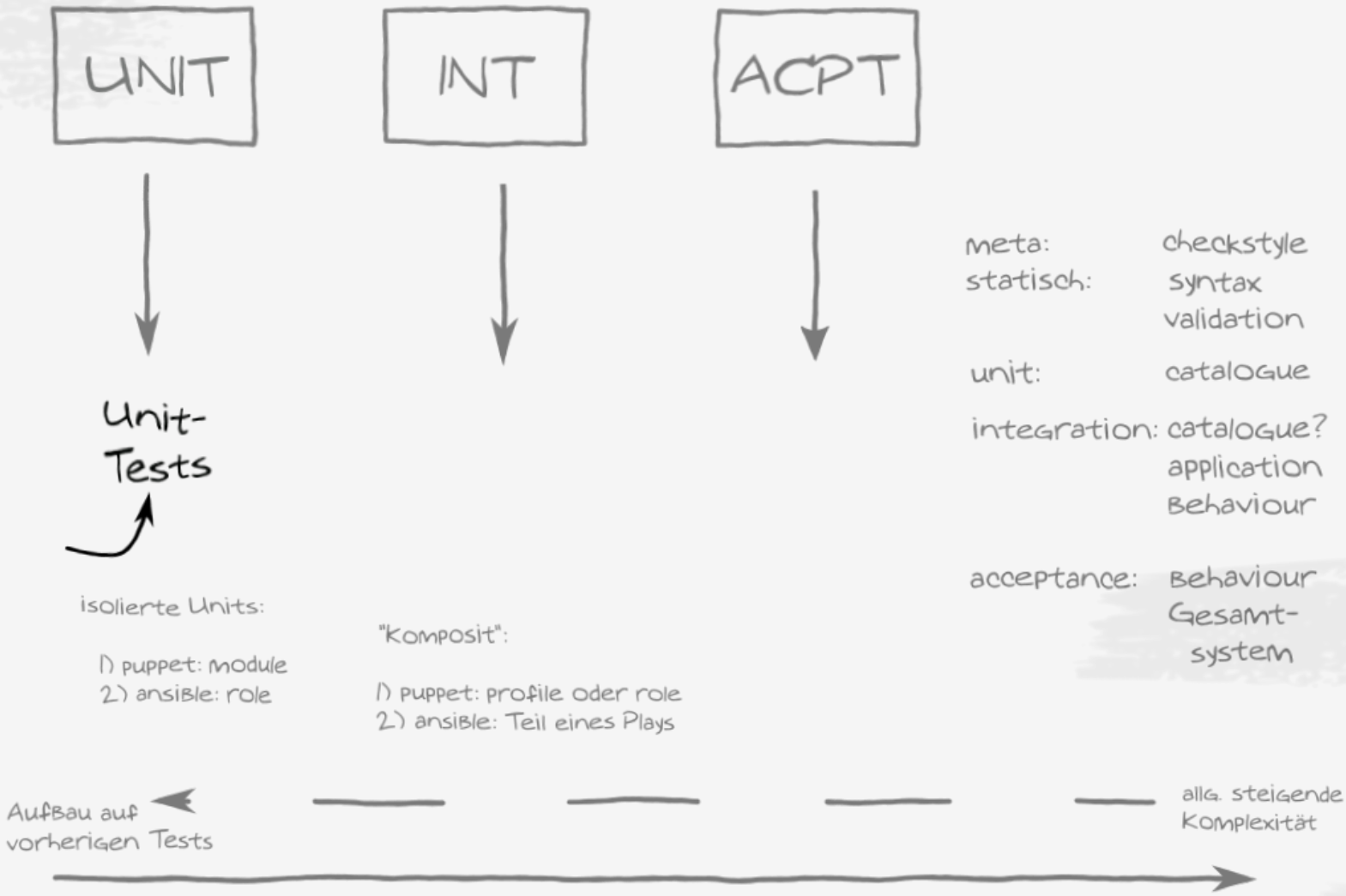
meta:	checkstyle
statisch:	syntax
	validation
unit:	catalogue
integration:	catalogue?
	application
	Behaviour

Test-Systematik

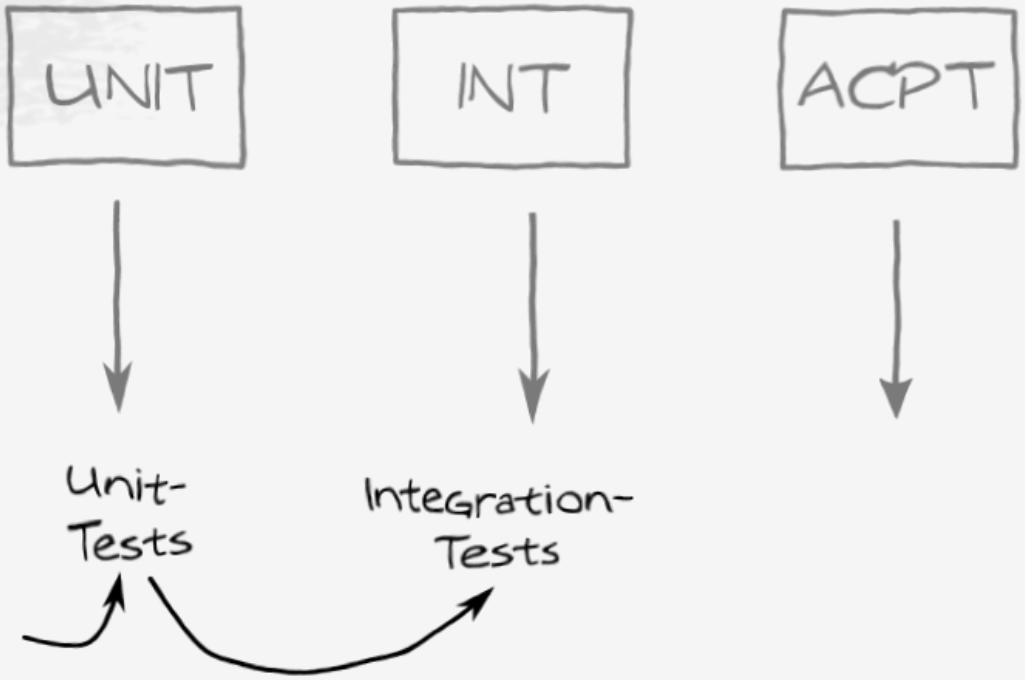


Christopher J.
Ruwe
<cjr@cruwe.de>

Test-Systematik



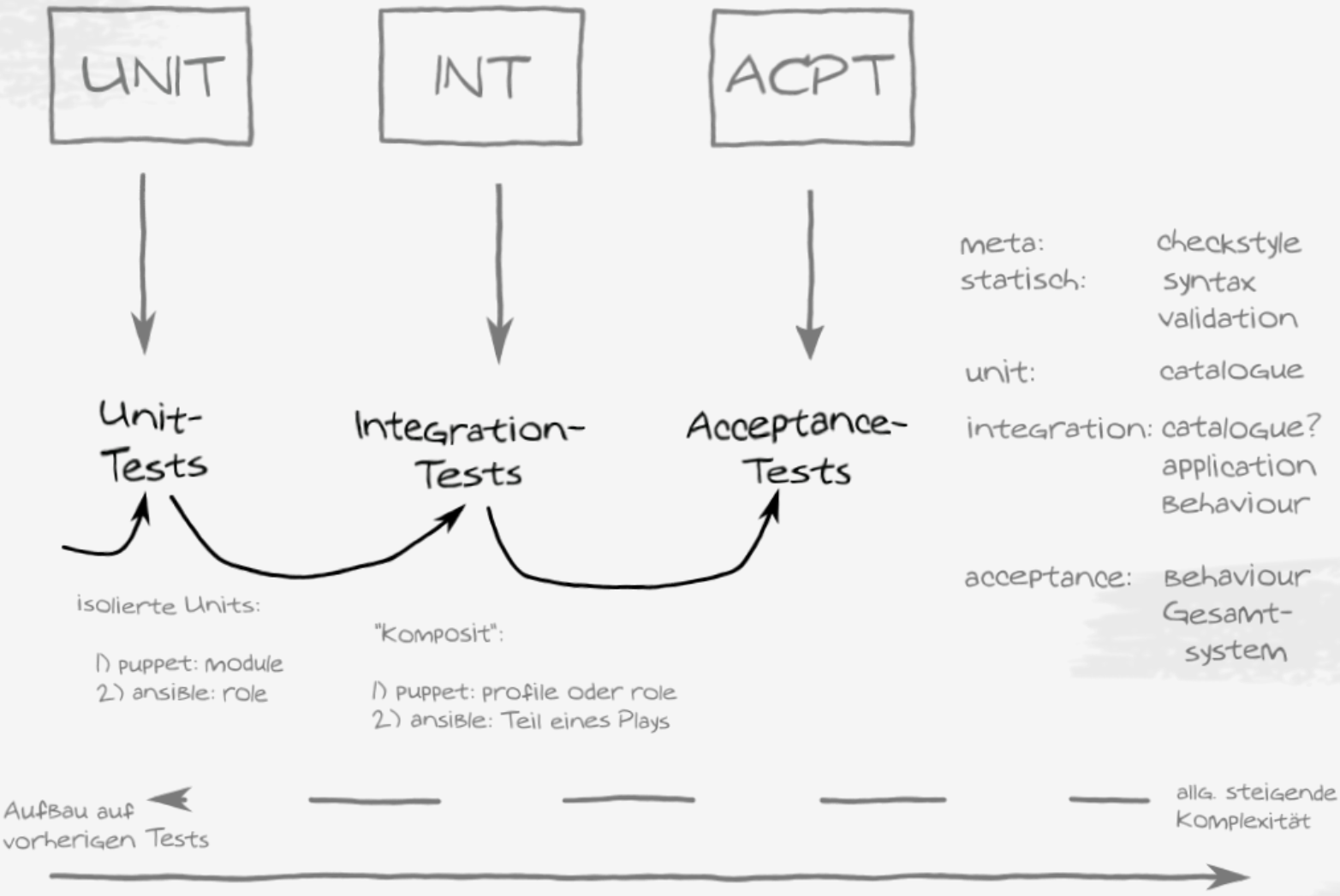
Test-Systematik



meta:	checkstyle
statisch:	syntax validation
unit:	catalogue
integration:	catalogue? application Behaviour
acceptance:	Behaviour Gesamt-system

Christopher J.
Ruwe
<cjr@cruwe.de>

Test-Systematik



Syntaktische Tests

Christopher J. Ruwe <cjr@cruwe.de>


```
find manifests -name "*.pp"  -print0 |
      xargs -0 -n 1 puppet parser validate --noop
```

```
find manifests -name "*.yaml" -print0 |
      xargs -0 -n 1 yamllint
```

Syntaktische Tests

Christopher J. Ruwe
<cjr@cruwe.de>


```
find manifests -name "*.pp"  -print0 |
      xargs -0 -n 1 puppet parser validate --noop
```

```
find manifests -name "*.yaml" -print0 |
      xargs -0 -n 1 yamllint
```

Christopher J. Ruwe
<cjr@cruwe.de>

Checkstyle

```
RuboCop::RakeTask.new(:rubocop) do |t|
  t.options =
    [
      '--display-cop-names',
      '--extra-details',
      '--fail-level=W',
      '--require=rubocop/formatter/checkstyle_formatter',
      '--format=RuboCop::Formatter::CheckstyleFormatter',
      '--out=reports/xml/rubocop-checkstyle.xml',
      '--format=json',
      '--out=reports/json/rubocop.json',
      '--format=html',
      '--out=reports/html/rubocop.html',
      '--format=progress',
    ]
end
```

Christopher J. Ruwe
<cjr@cruwe.de>

Puppet Catalogue Test

http://rspec-puppet.com/tutorial/


```
describe 'logrotate::rule' do
  let(:title) { 'nginx' }

  it { is_expected.to contain_class('logrotate::rule') }

  it do
    is_expected.to contain_file('/etc/logrotate.d/nginx').with({
      'ensure' => 'present',
      'owner'  => 'root',
      <...>}
  end

RSpec::Core::RakeTask.new(:spec) do |t|
  t.pattern = 'spec/*/*_spec.rb'
end
```

Puppet Catalogue Test

Christopher J. Ruwe <cjr@cruwe.de>

http://rspec-puppet.com/tutorial/

```
describe 'logrotate::rule' do
  let(:title) { 'nginx' }

  it { is_expected.to contain_class('logrotate::rule') }

  it do
    is_expected.to contain_file('/etc/logrotate.d/nginx').with({
      'ensure' => 'present',
      'owner'  => 'root',
       <...>}
   end

RSpec::Core::RakeTask.new(:spec) do |t|
  t.pattern = 'spec/*/*_spec.rb'
end
```

Christopher J. Ruwe <cjr@cruwe.de>

Puppet Catalogue Test

http://rspec-puppet.com/tutorial/


```
describe 'logrotate::rule' do
  let(:title) { 'nginx' }

  it { is_expected.to contain_class('logrotate::rule') }

  it do
    is_expected.to contain_file('/etc/logrotate.d/nginx').with({
      'ensure' => 'present',
      'owner'  => 'root',
       <...>}
   end

RSpec::Core::RakeTask.new(:spec) do |t|
  t.pattern = 'spec/*/*_spec.rb'
end
```

Puppet Catalogue Test

```
describe 'role::webserver::devstage' do


it { is_expected.to contain_class('nginx') }
it { is_expected.to contain_nginx__vhost('<thevhost>') }
it { is_expected.to contain_service('nginx')
                    .with_ensure('running') }
```

Applikation eines Catalogue / Play

http://kitchen.ci:

Wir erzeugen aus einem Datensatz (deklarativ!) eine virtuelle Maschine

```
bundle exec kitchen create www-debian-stretch-docker
```

Applizieren den Code (`puppet apply`, `ansible-play`)

```
bundle exec kitchen converge www-debian-stretch-docker
```

und verfizieren das Ergebnis.

```
bundle exec kitchen converge www-debian-stretch-docker
```

Applikation eines Catalogue / Play

http://kitchen.ci:

Wir erzeugen aus einem Datensatz (deklarativ!) eine virtuelle Maschine

```
bundle exec kitchen create www-debian-stretch-docker
```

Applizieren den Code (`puppet apply`, `ansible-play`)

```
bundle exec kitchen converge www-debian-stretch-docker
```

und verfizieren das Ergebnis.

```
bundle exec kitchen converge www-debian-stretch-docker
```

Applikation eines Catalogue / Play

http://kitchen.ci:

Wir erzeugen aus einem Datensatz (deklarativ!) eine virtuelle Maschine

```
bundle exec kitchen create www-debian-stretch-docker
```

Applizieren den Code (`puppet apply`, `ansible-play`)

```
bundle exec kitchen converge www-debian-stretch-docker
```

und verfizieren das Ergebnis.

```
bundle exec kitchen converge www-debian-stretch-docker
```

Applikation eines Catalogue / Play

http://kitchen.ci:

Wir erzeugen aus einem Datensatz (deklarativ!) eine virtuelle Maschine

```
bundle exec kitchen create www-debian-stretch-docker
```

Applizieren den Code (puppet apply, ansible-play)

```
bundle exec kitchen converge www-debian-stretch-docker
```

und verfizieren das Ergebnis.

```
bundle exec kitchen converge www-debian-stretch-docker
```

Serverspec - :ssh remotely

```
describe 'Check the webserver on distant machine.' do
  describe service('nginx') do
    it { should be_enabled }
    it { should be_running }
  end

  describe port(80) do
    it { should be_listening.with('tcp') }
    it { should be_listening.with('tcp6') }
  end

  describe process('nginx') do
    its(:user) { should eq 'www-data' }
    its(:count) { should eq 4 }
  end
end
```

Serverspec - :exec locally

Christopher J. Ruwe <cjr@cruwe.de>


```
describe 'Check connections from localhost.' do
  describe command("curl -I http://www.cruwe.de") do
    its(:stdout) { should match /HTTP\/1.1 301/ }
    its(:stdout) { should match /Location: https:\/\/<...>/ }
    its(:stdout) { should match /Server: nginx/ }
    its(:stdout) { should match /Content-Type: text\/html/ }
    its(:stderr) { should match /X-Clacks-Overhead:
                                 GNU Terry Pratchett/ }
  end
end
```

Serverspec - :ssh on various remotes

```
describe 'Check www from localhost.' do
  <...>
end
```


Christopher J. Ruwe
<cjr@cruwe.de>

Serverspec - :ssh on various remotes

```
describe 'Check www from localhost.' do
  <...>
end

describe 'Check app-server from frontend.' do
  <...>
end
```


Christopher J. Ruwe
<cjr@cruwe.de>

Serverspec - :ssh on various remotes

```
describe 'Check www from localhost.' do
  <...>
end

describe 'Check app-server from frontend.' do
  <...>
end

describe 'Check service bus from app-server.' do
  <...>
end
```


Christopher J. Ruwe <cjr@cruwe.de>

Serverspec - :ssh on various remotes

```
describe 'Check www from localhost.' do
  <...>
end

describe 'Check app-server from frontend.' do
  <...>
end

describe 'Check service bus from app-server.' do
  <...>
end

describe 'Check DB from app-server.' do
  <...>
end
```


Christopher J. Ruwe
<cjr@cruwe.de>

Automatisierung mit dressiertem Affen

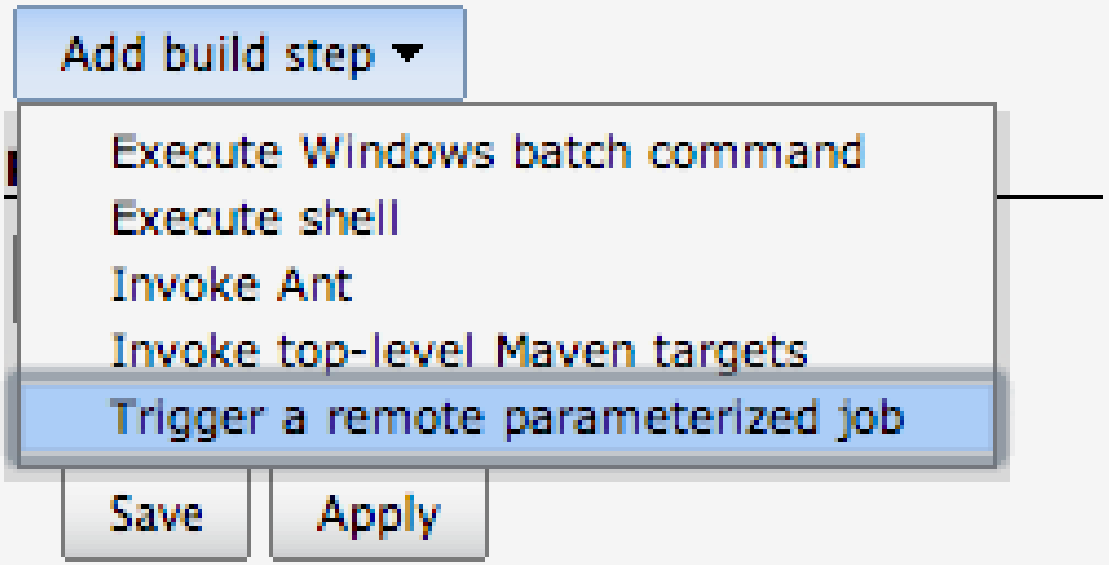

```
[cjr@lamport:/home/cjr/media/src/puppet/hb22-site]
$ find manifests -name <...>          (<feature>↓29↑33|+1)
```

Automatisierung mit dressiertem Affen


```
[cjr@lamport:/home/cjr/media/src/puppet/hb22-site]
$ find manifests -name <...>          (<feature>↓29↑33|+1)
```

```
[cjr@lamport:/home/cjr/media/src/puppet/hb22-site]
$ rspec -c -f documentation  <..>    (<feature>↓29↑33|+1)
```

dressierter Affe „Jenkins"

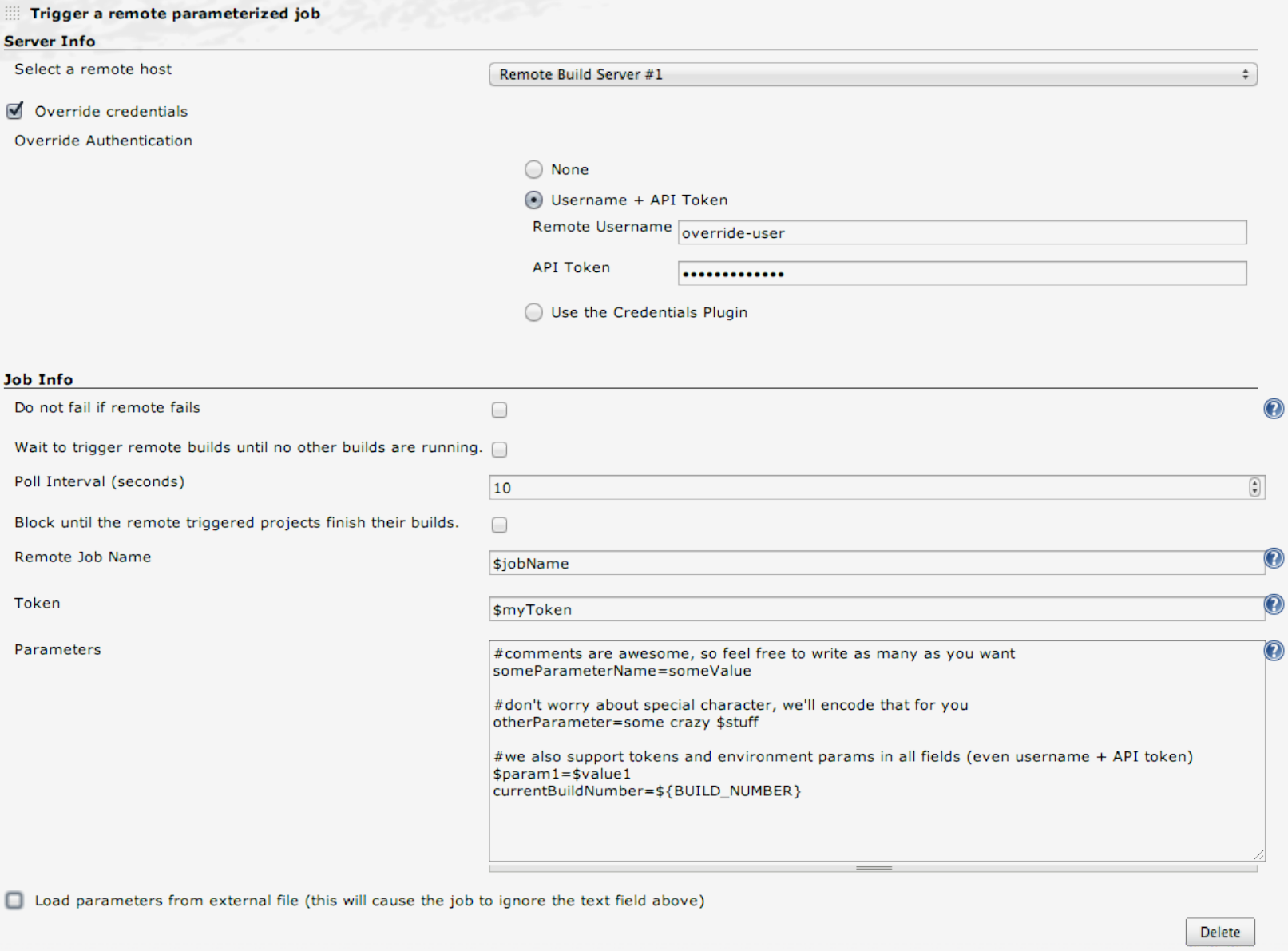


Christopher J. Ruwe
<cjr@cruwe.de>

dressierter Affe „Jenkins"

Infrastrukturtests mit der Jenkinsfile

Christopher J. Ruwe <cjr@cruwe.de>



dressierter Affe „Jenkins"

Infrastrukturtests mit der Jenkinsfile

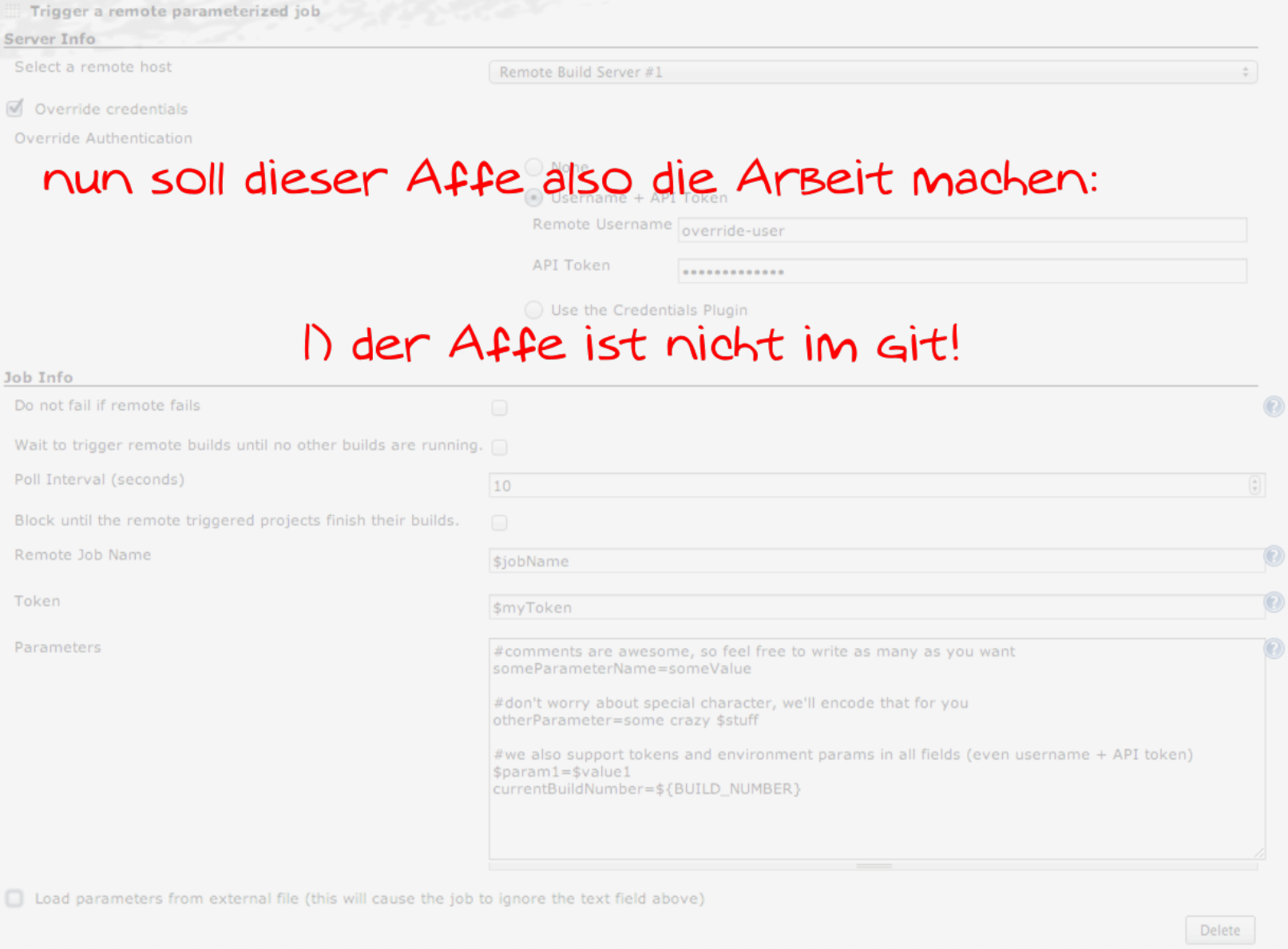
Christopher J. Ruwe <cjr@cruwe.de>

nun soll dieser Affe also die Arbeit machen:

dressierter Affe „Jenkins"

Infrastrukturtests mit der Jenkinsfile

Christopher J. Ruwe <cjr@cruwe.de>

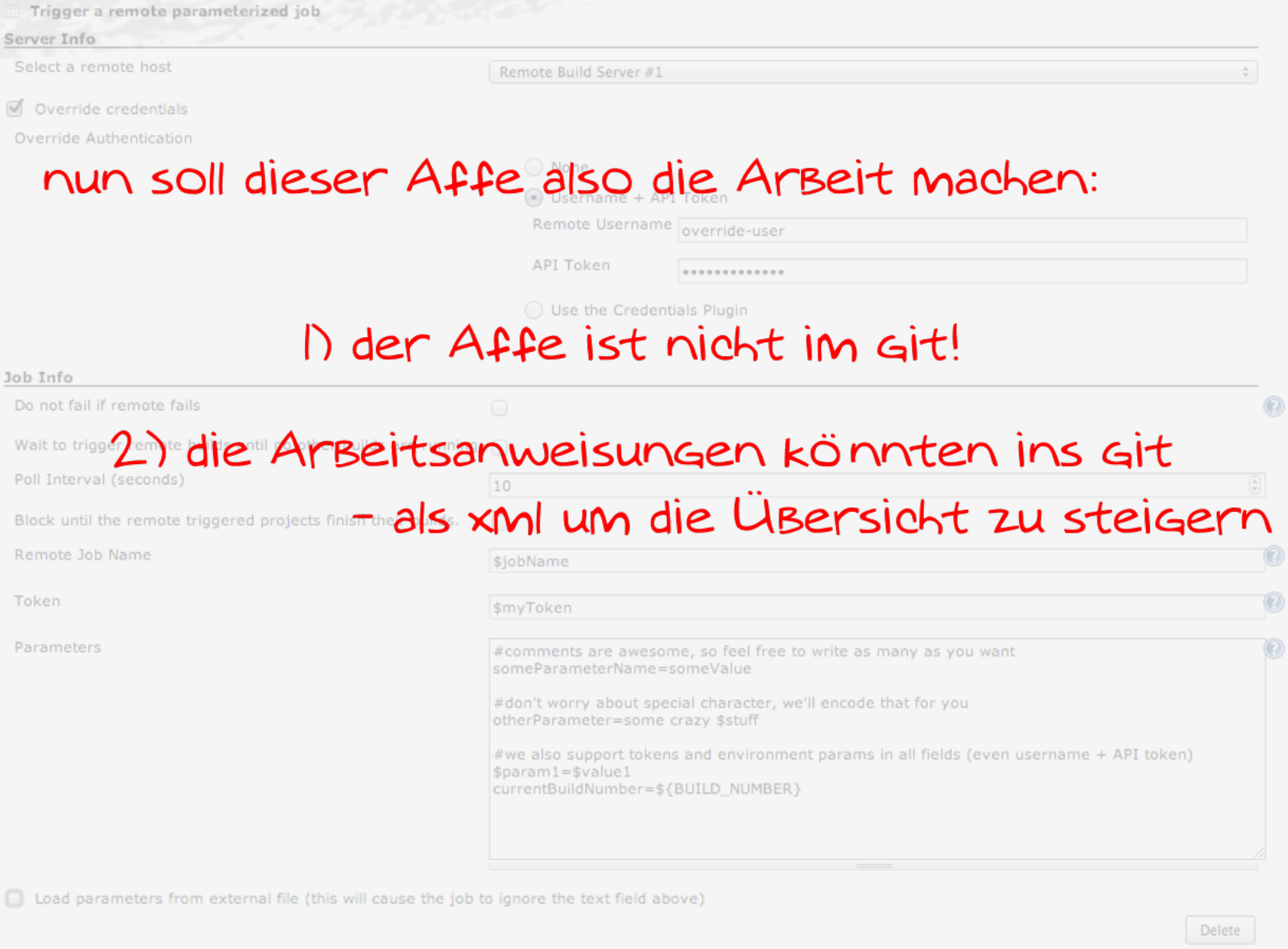


nun soll dieser Affe also die Arbeit machen:

1) der Affe ist nicht im Git!

dressierter Affe „Jenkins"

Christopher J. Ruwe
<cjr@cruwe.de>



nun soll dieser Affe also die Arbeit machen:

1) der Affe ist nicht im Git!

2) die Arbeitsanweisungen könnten ins Git
- als xml um die Übersicht zu steigern

dressierter Affe „Jenkins"

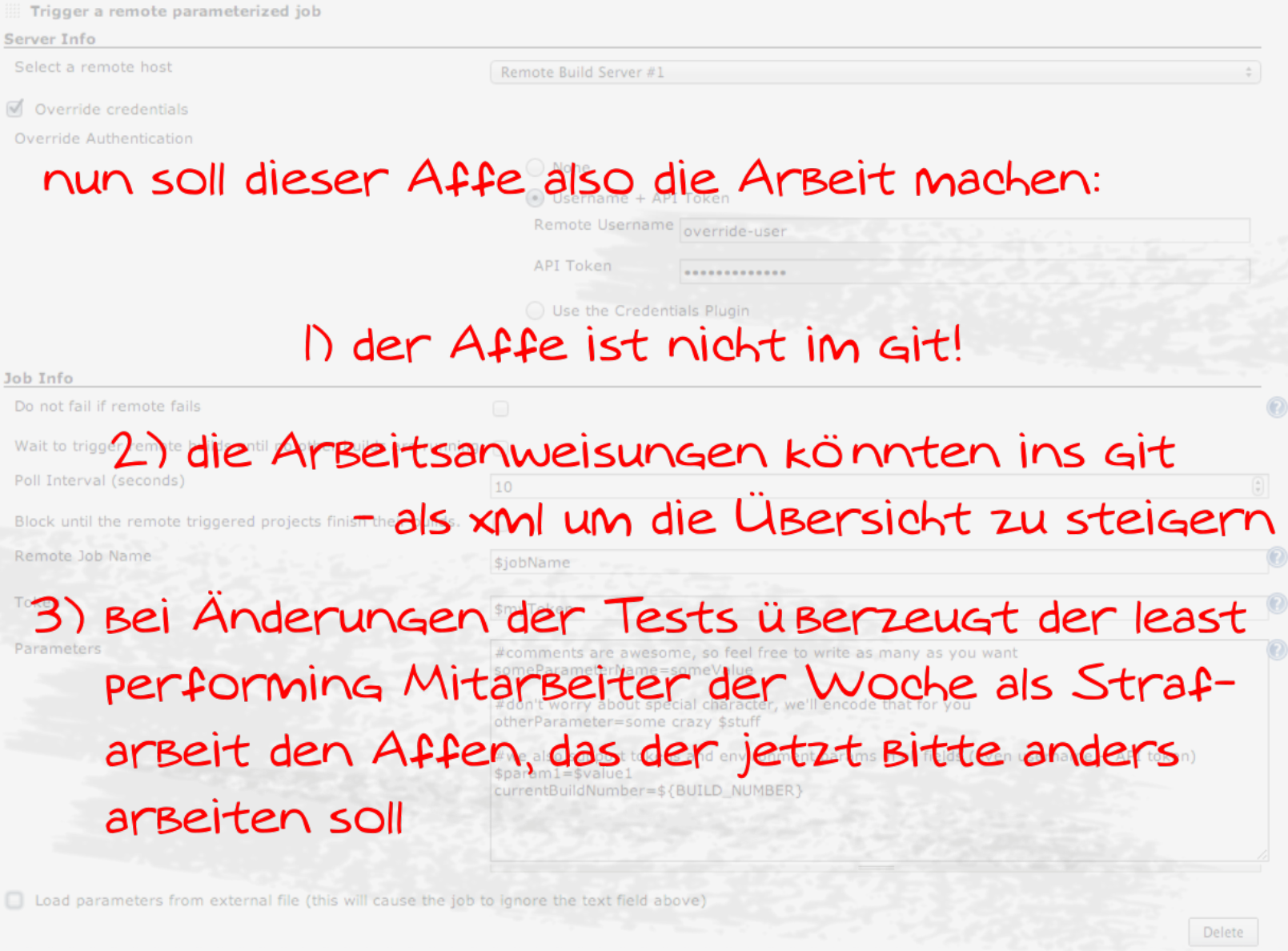
Infrastrukturtests mit der Jenkinsfile

Christopher J. Ruwe <cjr@cruwe.de>


```
<?xml version='1.0' encoding='UTF-8'?>
<org.jenkinsci.plugins.workflow.multibranch.WorkflowMultiBranchProject plugin="workflow-multibranch@2.9.2">
  <actions/>
  <description></description>
  <properties/>
  <folderViews class="com.cloudbees.hudson.plugins.folder.views.DefaultFolderViewHolder" plugin="cloudbees-folder@5.14">
    <views>
      <hudson.model.AllView>
        <owner class="org.jenkinsci.plugins.workflow.multibranch.WorkflowMultiBranchProject" reference="../../.."/>
        <name>All</name>
        <filterExecutors>false</filterExecutors>
        <filterQueue>false</filterQueue>
        <properties class="hudson.model.View$PropertyList"/>
      </hudson.model.AllView>
    </views>
    <tabBar class="hudson.views.DefaultViewsTabBar"/>
  </folderViews>
  <healthMetrics>
    <com.cloudbees.hudson.plugins.folder.health.WorstChildHealthMetric plugin="cloudbees-folder@5.14">
      <nonRecursive>false</nonRecursive>
    </com.cloudbees.hudson.plugins.folder.health.WorstChildHealthMetric>
  </healthMetrics>
  <icon class="com.cloudbees.hudson.plugins.folder.icons.StockFolderIcon" plugin="cloudbees-folder@5.14"/>
  <orphanedItemStrategy class="com.cloudbees.hudson.plugins.folder.computed.DefaultOrphanedItemStrategy" plugin="cloudbees-folder@5.
    <pruneDeadBranches>true</pruneDeadBranches>
    <daysToKeep>0</daysToKeep>
    <numToKeep>0</numToKeep>
  </orphanedItemStrategy>
  <triggers/>
  <sources class="jenkins.branch.MultiBranchProject$BranchSourceList" plugin="branch-api@1.11.1">
    <data>
      <jenkins.branch.BranchSource>
        <source class="jenkins.plugins.git.GitSCMSource" plugin="git@3.0.1">
          <id>22767748-49aa-470b-8593-695f6d9386c0</id>
          <remote>https://gogs.hb22.cruwe.de:4430/hb22/hb22-site.git</remote>
          <credentialsId></credentialsId>
          <includes>*</includes>
          <excludes></excludes>
          <ignoreOnPushNotifications>false</ignoreOnPushNotifications>
        </source>
```

Christopher J. Ruwe
<cjr@cruwe.de>

dressierter Affe „Jenkins"



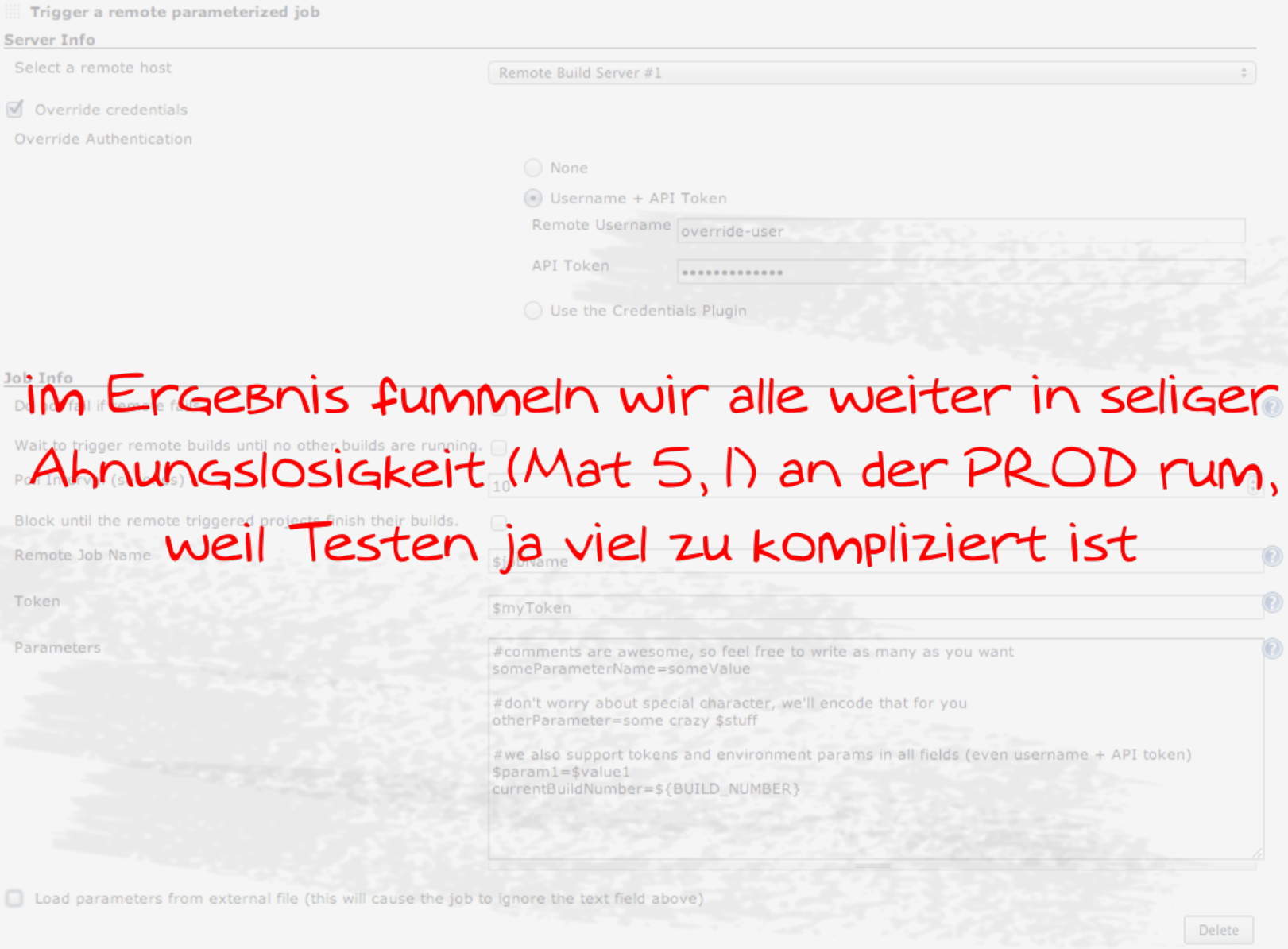
nun soll dieser Affe also die Arbeit machen:

1) der Affe ist nicht im Git!

2) die Arbeitsanweisungen könnten ins Git
- als xml um die Übersicht zu steigern

3) Bei Änderungen der Tests überzeugt der least performing Mitarbeiter der Woche als Straf-arbeit den Affen, das der jetzt bitte anders arbeiten soll

dressierter Affe „Jenkins"



im Ergebnis fummeln wir alle weiter in seliger Ahnungslosigkeit (Mat 5, 1) an der PROD rum, weil Testen ja viel zu kompliziert ist

Infrastrukturtests mit der Jenkinsfile

Christopher J. Ruwe <cjr@cruwe.de>

Jenkinsfile

Christopher J. Ruwe <cjr@cruwe.de>

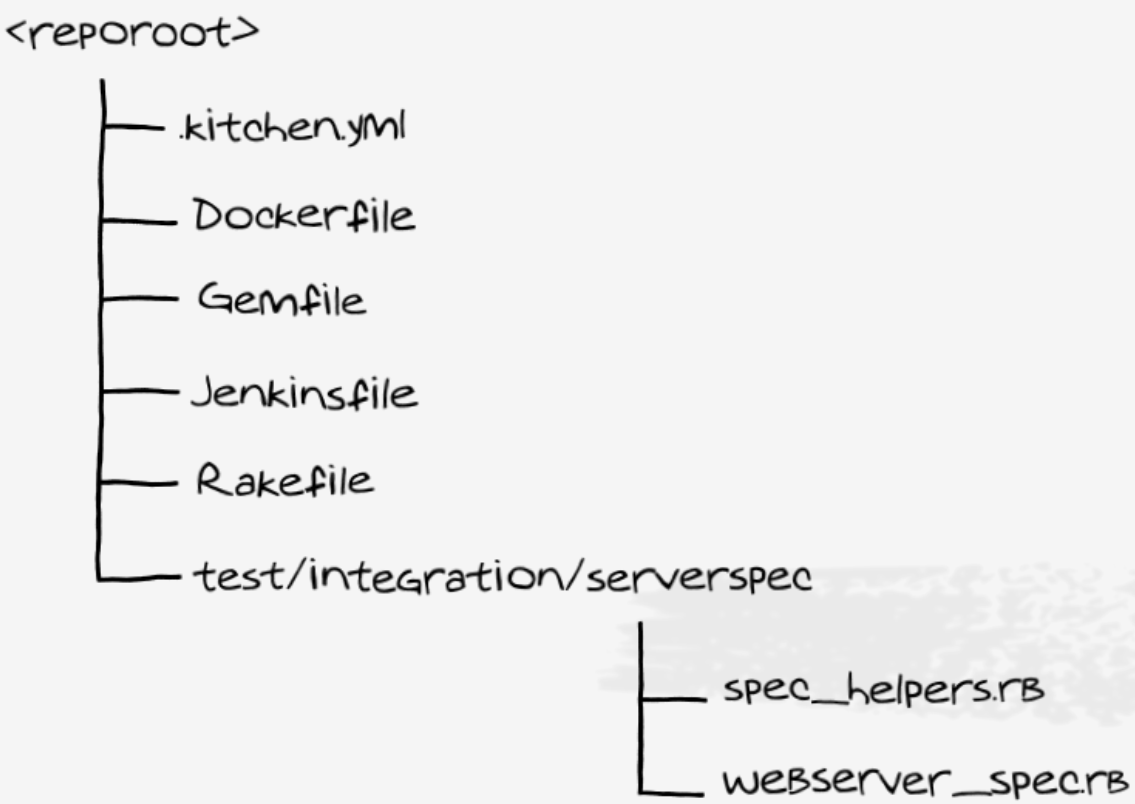

```
stage('Validate style') {
  parallel(
    'rubocop': {
      try {
        sh '''#!/bin/bash -l
              rvm use 2.2
              bundle exec rake rubocop
           '''
      } catch (err) { } finally {
        step(
         [ $class          : "hudson.plugins.checkstyle.CheckStylePublisher",
           failedTotalAll : '5',
           healthy         : "95",
           pattern         : "reports/xml/rubocop-checkstyle.xml", ]
        )
      }
    },
    'puppet_lint': {<..>} catch (err) {}
       finally {
         step(
           [ $class        : 'WarningsPublisher',
             <...> ]
         )
       }
    },
    'yaml_lint': { <...>
```

Jenkinsfile

```
stage('Integrate') {
  try{
    wrap([$class: 'AnsiColorBuildWrapper', 'colorMapName': 'XTerm']) {
      sh '''#!/bin/bash -l
          set -e
          set -o pipefail

          rvm use 2.2

          eval "$(ssh-agent)"

          bundle exec kitchen create docker || true
          ssh-add ./.kitchen/docker_id_rsa
          bundle exec kitchen create docker
          bundle exec kitchen converge docker
          bundle exec kitchen verify docker
        '''
    }
  } catch (err){ } finally {
    sh '''#!/bin/bash -l
          rvm use 2.2

          bundle exec kitchen destroy docker ; rm -Rf .kitchen
          pkill ssh-agent

        '''
    junit '**/reports/xml/serverspec-result.xml'
  }
}
```

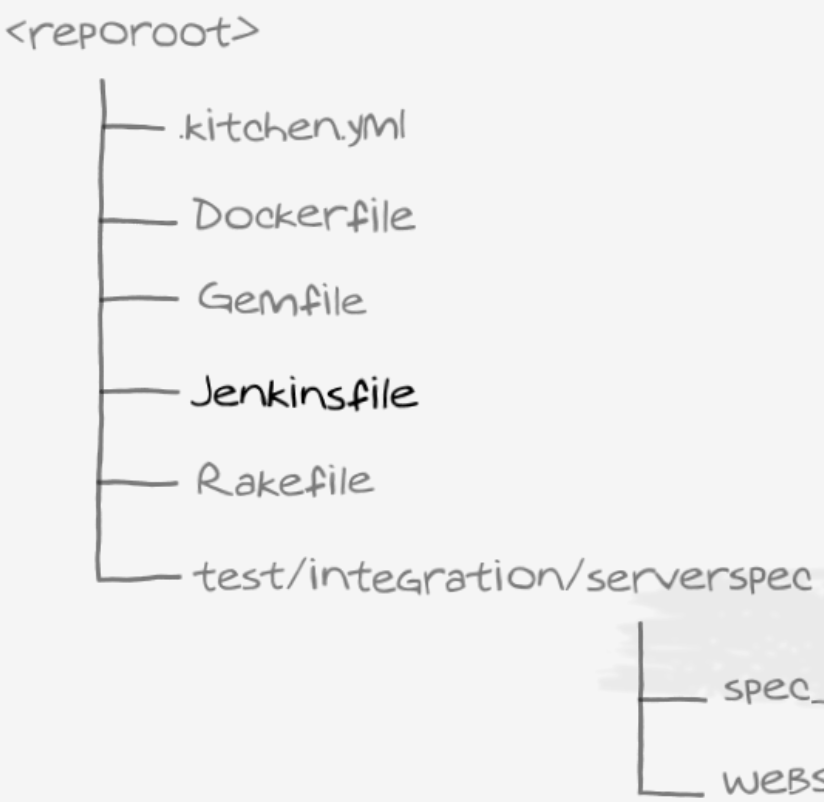

Christopher J. Ruwe
<cjr@cruwe.de>

Jenkinsfile - Repo



Jenkinsfile - Repo

Christopher J. Ruwe
<cjr@cruwe.de>



definiert die Pipeline

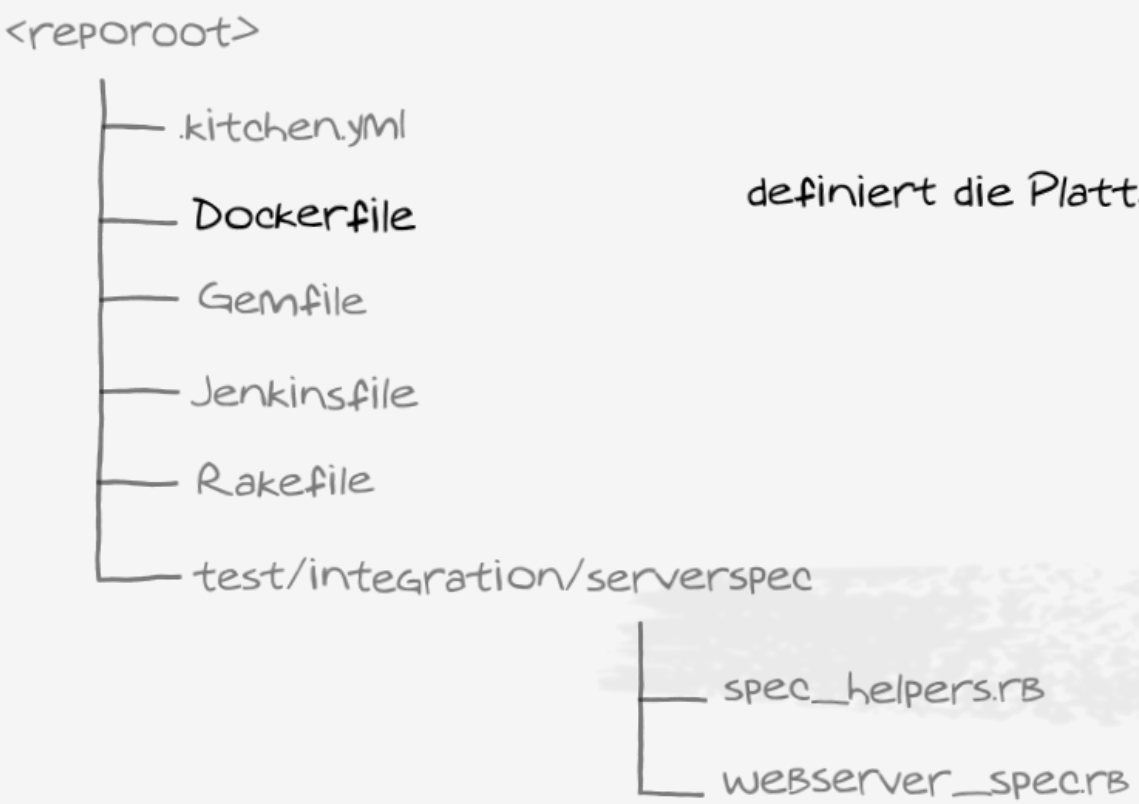
Jenkinsfile - Repo

Christopher J. Ruwe
<cjr@cruwe.de>

<reporoot>
- .kitchen.yml
- Dockerfile
- Gemfile
- Jenkinsfile
- Rakefile
- test/integration/serverspec
  - spec_helpers.rb
  - webserver_spec.rb

definiert Softeware-Abhängigkeiten

Jenkinsfile - Repo



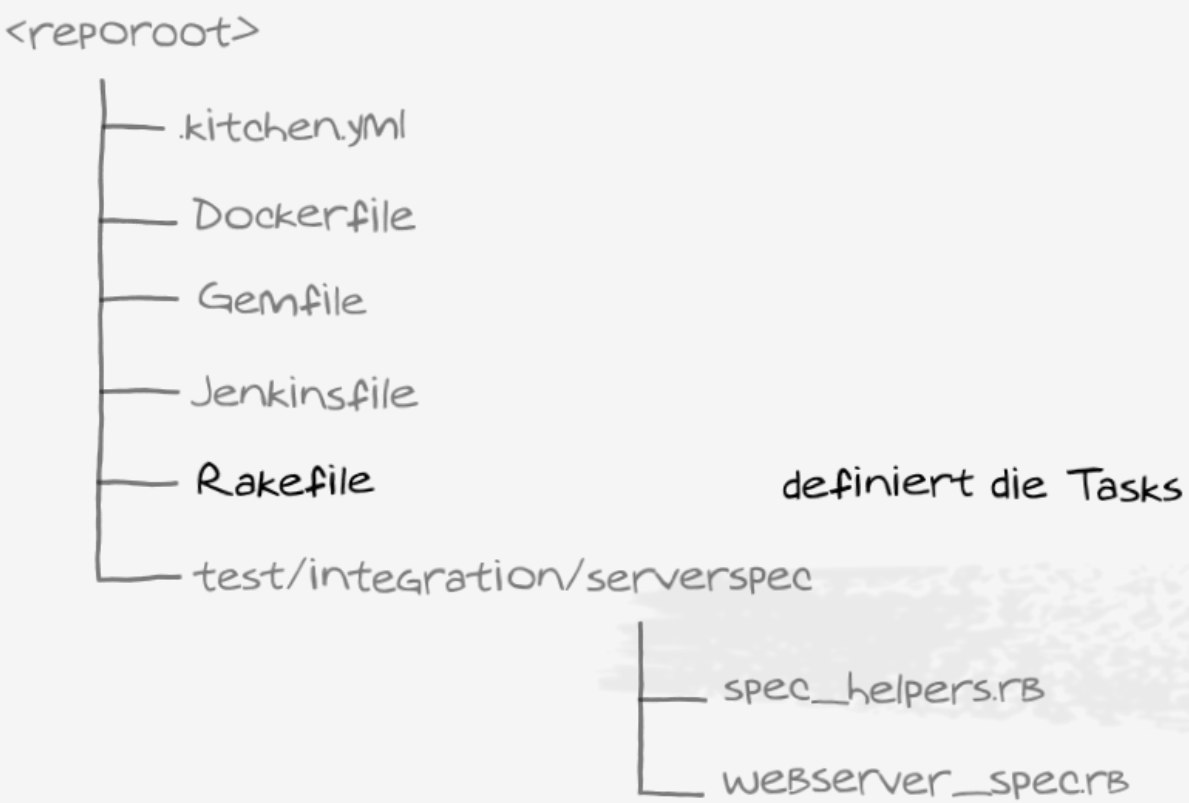
definiert die Plattform

Jenkinsfile - Repo

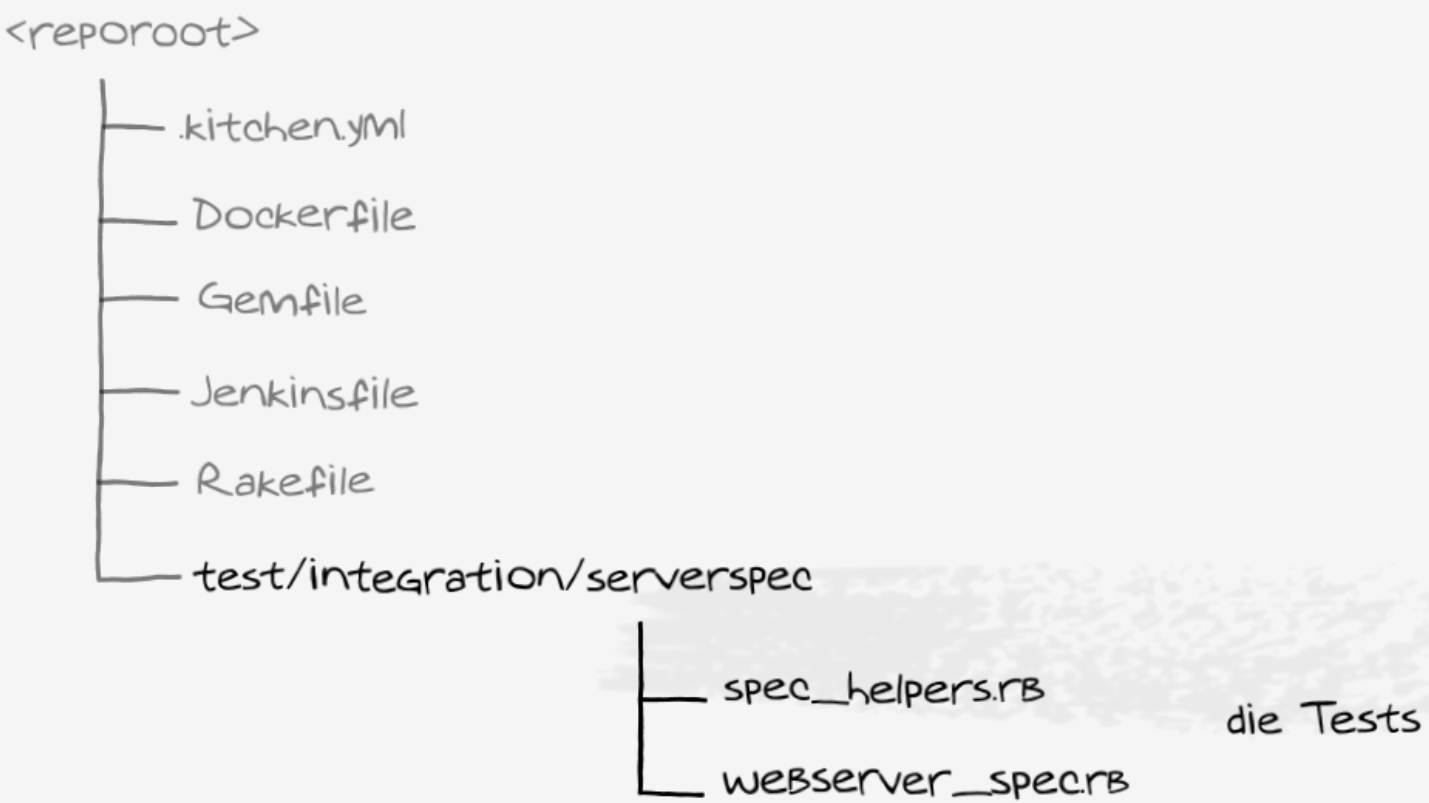
verbindet Plattform und Provisionierer

Jenkinsfile - Repo

Christopher J. Ruwe
<cjr@cruwe.de>



Jenkinsfile - Repo



Jenkinsfile - Outputs

Christopher J. Ruwe
<cjr@cruwe.de>

Pipeline refine_jenkinsfile_testing

Full project name: multi-pipeline/refine_jenkinsfile_testing

Recent Changes

Stage View

	Checkout	Prepare environment
Average stage times:	887ms	16s
#2 Mar 11 14:51 No Changes	420ms	7s failed
#1 Mar 11 14:49 No Changes	1s	24s failed

Jenkinsfile - Outputs

Pipeline refine_jenkinsfile_testing

Full project name: multi-pipeline/refine_jenkinsfile_testing

Recent Changes

Stage View

Jenkinsfile - Outputs

Pipeline refine_jenkinsfile_testing

Full project name: multi-pipeline/refine_jenkinsfile_testing

Recent Changes

Stage View

	Checkout	Prepare environment	Validate syntax	Validate style	Integrate
Average stage times:	715ms	13s	22s	13s	5min 12s
#3 Mar 11 14:57 No Changes	372ms	7s	22s	13s	5min 12s
#2 Mar 11 14:51 No Changes	420ms	7s failed			
#1 Mar 11 14:49 No Changes	1s	24s failed			

Infrastrukturtests mit der Jenkinsfile

Christopher J. Ruwe <cjr@cruwe.de>

Jenkinsfile - Outputs

Christopher J. Ruwe
<cjr@cruwe.de>

Build #3 (Mar 11, 2017 2:57:35 PM)

Started 6 min 50 sec ago
Took 5 min 57 sec

add description

Started by anonymous user

This run spent:

- 11 ms waiting in the queue;
- 5 min 57 sec building on an executor;
- 5 min 57 sec total from scheduled to completion.

Revision: afb2852408e1173d5f886607ac992533567520c3

- refine_jenkinsfile_testing

yaml-lint - Warnings: 6 warnings from one analysis.

- 6 new warnings

Puppet-Lint Warnings: 0 warnings.

- No warnings since build 3.
- New zero warnings highscore: no warnings since yesterday!
- During parsing an error has been reported.

Checkstyle: 29 warnings from one analysis.

- 29 new warnings
- Plug-in Result: - 29 warnings exceed the threshold of 5 by 24

Test Result (1 failure)
Service "sshd".Check the sshd. Service "sshd" should be running

Jenkinsfile - Outputs

Christopher J. Ruwe <cjr@cruwe.de>

CheckStyle Result

Warnings Trend

All Warnings	New Warnings	Fixed Warnings
29	29	0

Summary

Total	High Priority	Normal Priority	Low Priority
29	0	0	29

Details

Folders | Files | Types | Warnings | Details | New

Source Folder	Total	Distribution
-	5	
scripts	1	
test/integration/serverspec	23	
Total	29	

Jenkinsfile - Outputs

Christopher J. Ruwe
<cjr@cruwe.de>

Test Result

1 failures

14 tests
Took 0.6 sec.
add description

All Failed Tests

Test Name	Duration	Age
Service "sshd".Check the sshd. Service "sshd" should be running	39 ms	1

All Tests

Package	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
(root)	0.6 sec	1	+1	0		13	+13	14	+14

Jenkinsfile - Outputs

Christopher J. Ruwe <cjr@cruwe.de>

Failed

Service "sshd".Check the sshd. Service "sshd" should be running

Failing for the past 1 build (Since #3)
Took 39 ms.
add description

Error Message

```
failed Check the sshd. Service "sshd" should be running
```

Stacktrace

```
        expected Service "sshd" to be running
["./test/integration/serverspec/base_spec.rb:19:in `block (3 levels) in <top (required)>'"]
```

Ausblick

an dieser Stelle ergeben sich weitere Möglichkeiten:

Ausblick

an dieser Stelle ergeben sich weitere Möglichkeiten:

- in der build-stage docker zu ersetzen und über eine VM- oder Cloud-API oder Tools wie Terraform et al. Maschinen zu provisionieren ist eine Fingerübung

Ausblick

an dieser Stelle ergeben sich weitere Möglichkeiten:

- in der build-stage docker zu ersetzen und über eine VM- oder Cloud-API oder Tools wie Terraform et al. Maschinen zu provisionieren ist eine Fingerübung

- damit sind on-demand Testumgebungen und eng am Code entwickelte Continuous Integration Pipelines möglich

Christopher J. Ruwe
<cjr@cruwe.de>

Ausblick

an dieser Stelle ergeben sich weitere Möglichkeiten:

- in der Build-stage docker zu ersetzen und über eine VM- oder Cloud-API oder Tools wie Terraform et al. Maschinen zu provisionieren ist eine Fingerübung
- damit sind on-demand Testumgebungen und eng am Code entwickelte Continuous Integration Pipelines möglich
- andere Ebene: serverspec Tests können fürs Verfügbarkeits-Monitoring verwendet werden - es gibt Plugins für Nagios und Sensu