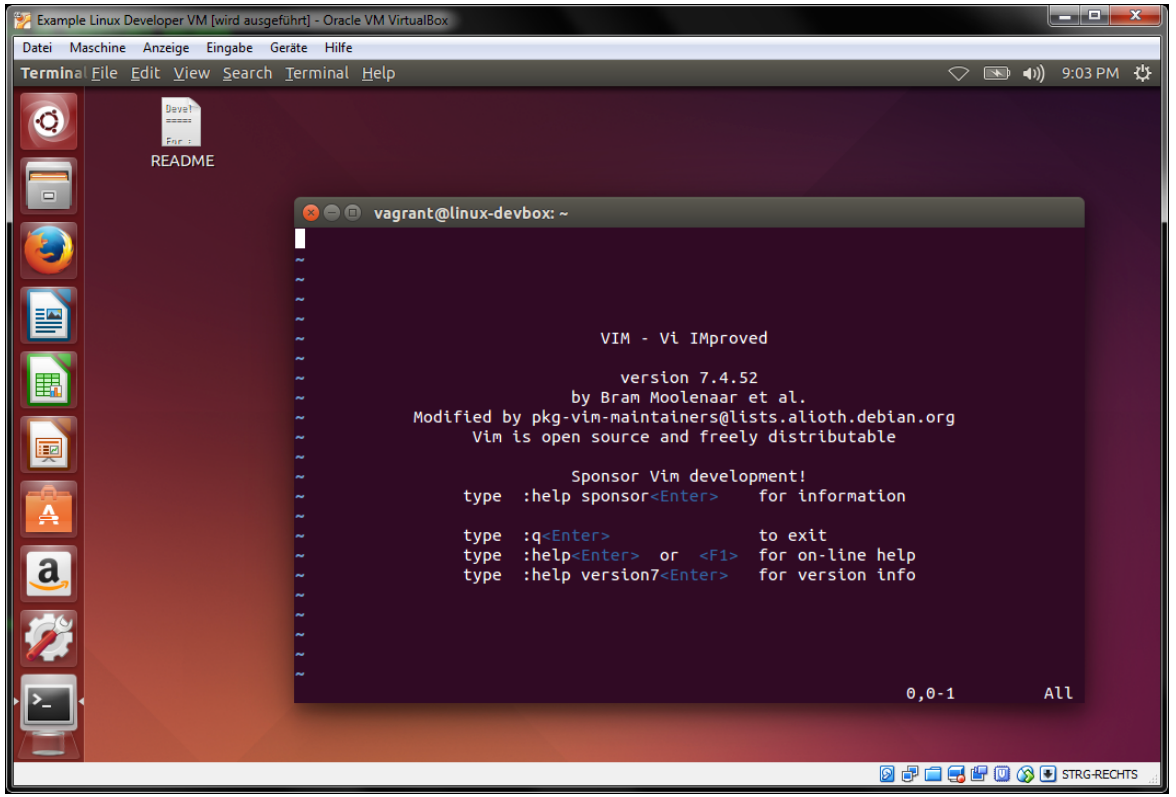


zühlke
empowering ideas

Automatisierte Entwickler VMs

„works on my machine“ zählt nicht mehr ;-)

About

Torben Knerr

github/tknerr

@tknerr_de

- Seit 10 Jahren bei Zühlke
- Software Architekt und Infrastructure-as-Code Enthusiast
- In verschiedensten Projekten unterwegs...
- ...und immer wieder:

“works on my machine“

- Let‘s fix this! ☺

Das Problem mit *“works on my machine“*...

- „Das nervt. Seit zwei Wochen auf dem Projekt und noch nichts kompiliert bei mir“
- „Warum nutzt eigentlich jeder hier eine andere Java / Maven / Eclipse Version?“
- „Aha, mit der Eclipse Installation von Peter, dem Plugins Verzeichnis von Manuela, und den Einstellungen von Kurt funktioniert es also. Und so machen es alle?!?“
- „Also Kurt hat bei sich noch ein paar Umgebungsvariablen angepasst. Den Trick mit den Registry Einstellungen kennst du auch, oder?“
- „Euer Buildserver läuft wirklich auf nem ganz anderen Betriebssystem?!?“
- „Warum ist die Doku zum Aufsetzen der Entwicklungsumgebung eigentlich IMMER veraltet?“

Eine mögliche Lösung: **Automatisierte Entwickler VMs**

VMs bieten eine konsistente, isolierte und reproduzierbare Umgebung

Warum Automatisieren?

- Schlanker Source Code gegenüber generiert fetter VM-Image
- Bessere Reproduzierbarkeit, dadurch geringere Fehleranfälligkeit
- Alles unter Versionskontrolle: Historie, Diffs, Pull Requests, usw...
- Mit Configuration Management Tools (Chef, Ansible, etc): Deklarative Spezifikation des Zielzustands statt imperativer Skripte

Infrastructure-As-Code

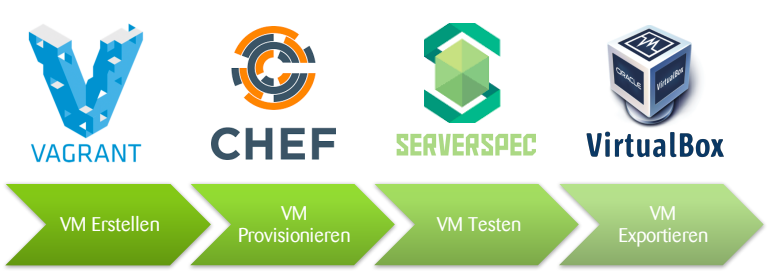
Mein Ziel für heute

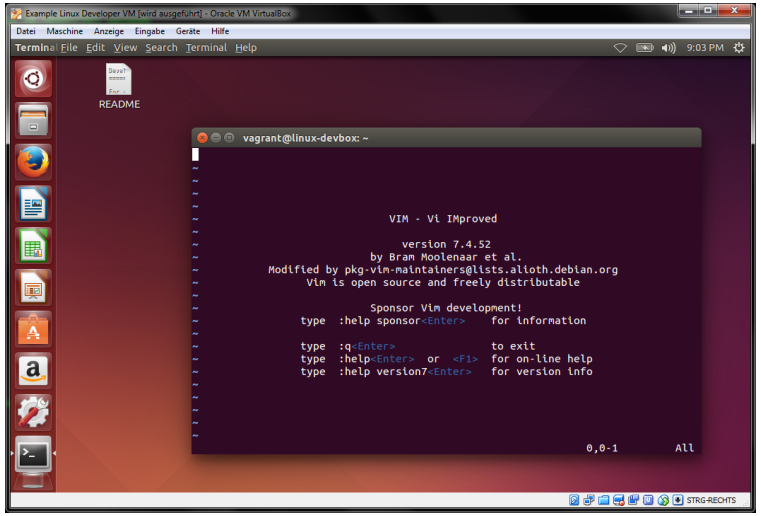
- Grundlegendes Verständnis für die Automatisierung von Entwickler VMs schaffen
- Gerade genug Chef + Vagrant Einführung, um die Zusammenhänge zu verstehen
- Direkt nutzbare Vorlagen für einen „Kickstart“ mitgeben

Automatisierte Entwickler VMs

zühlke
empowering ideas

Anforderungen und kurze Vorstellung der Toolchain





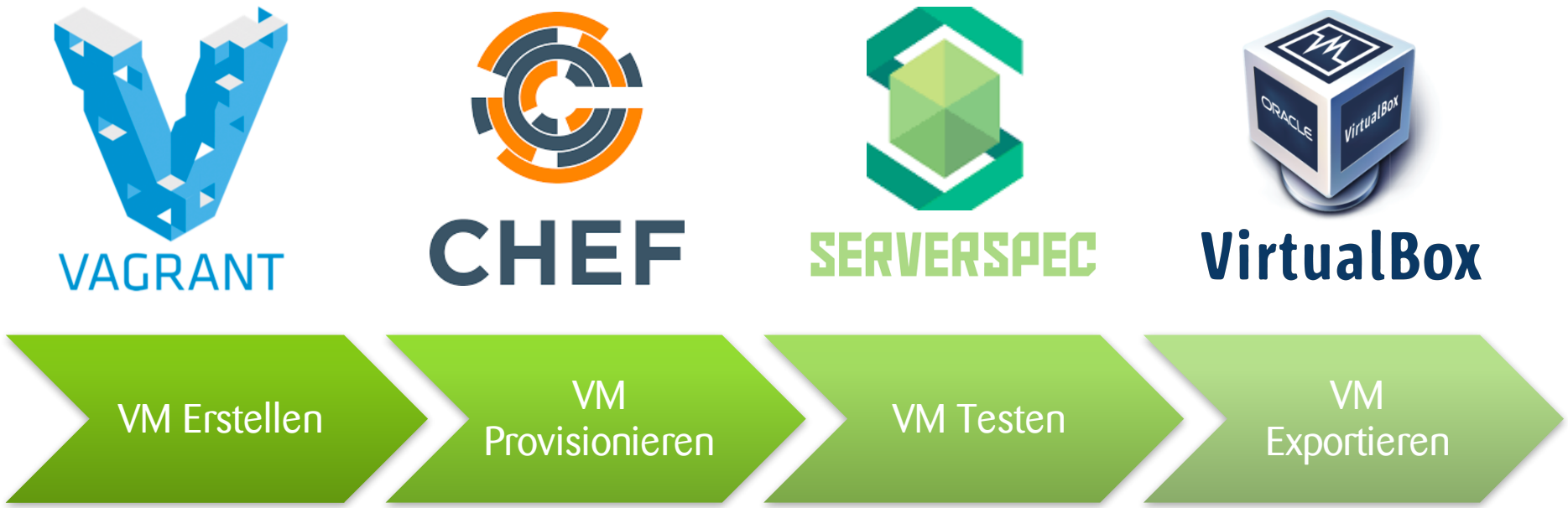
Anforderungen an eine Automatisierte Entwickler VM

Diese Eigenschaften sollte eine Entwickler VM erfüllen:

- **Aktualisierbarkeit** – die VM soll sich selbst aktualisieren können
- **Testbarkeit** – die Funktionstüchtigkeit der VM soll stets abgesichert sein
- **Installierbarkeit** – die VM soll als Standard .ova Image verteilt werden
- **Anpassbarkeit** – die VM soll von den Entwicklern einfach anpassbar sein
- **Nachvollziehbarkeit** – Änderungen an der VM sollen transparent sein

Eine Toolchain zum Automatisieren von Entwickler VMs

(die Tools werden anschließend in einer Demo kurz vorgestellt)



Was ist Vagrant?

…ein Tool um VMs automatisiert zu erstellen und zu provisionieren

- Ein **Vagrantfile** beschreibt die VMs
- Einfache **Befehle** um mit den VMs zu interagieren:
  - vagrant up
  - vagrant provision
  - vagrant ssh
  - vagrant destroy
  - …
- Mehrere **Provider**: VirtualBox, VMWare, Docker, AWS, …
- Dazu verschiedene **Provisioner**: Shell, Chef, Ansible, us…

Demo Time!

```
Vagrant.configure("2") do |config|
  config.vm.box = "boxcutter/ubuntu-16.04"

  # web server VM
  config.vm.define "web" do |web|
    web.vm.provision "shell", inline: "apt-get install apache2 -y"
    web.vm.network "private_network", ip: "192.168.40.80"
  end

  # database VM
  config.vm.define "db" do |db|
    db.vm.provision "shell", inline: "apt-get install postgresql -y"
    db.vm.network "private_network", ip: "192.168.40.81"
  end
end
```

https://www.vagrantup.com/docs

Was ist Chef?

…ein Configuration Management Tool um die VMs zu konfigurieren

- Ein **Cookbook** ist eine wiederverwendbare Einheit
- Darin enthalten sind **Recipes**, Templates, Libraries, etc..
- In einem **Recipe** werden **Resourcen** beschrieben:
  - file
  - service
  - package
  - user
  - …
- Der **chef-client** „konvergiert“ das System in den spezifizierten Zustand

Demo Time!

```
package 'apache2' do
  action :install
end

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

file '/var/www/html/index.html' do
  content '<html>Etka rocks!</html>'
  action :create
  notifies :restart, "service[apache2]"
end
```

https://learn.chef.io/tutorials/

Was ist Serverspec?

…ein Framework um die VMs automatisiert zu testen

- Basierend auf **RSpec** (Rubyisten kennen das ☺)
- Stellt **Matcher** für typische System Ressourcen bereit:
  - file
  - package
  - service
  - user
  - …
- Unterstützt verschiedene **Backends** (z.B. local, SSH, WinRM).
- ..und **Betriebssysteme** (diverse Linux Distributionen, Windows)

Demo Time!

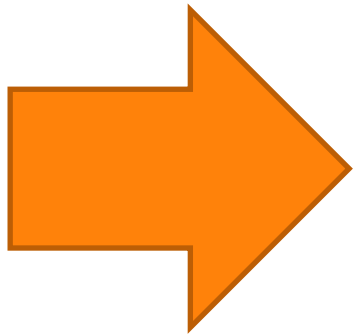
```
require 'spec_helper'

describe package('apache2') do
  it { should be_installed }
end

describe service('apache2') do
  it { should be_enabled   }
  it { should be_running   }
end

describe port(80) do
  it { should be_listening }
end

describe file('/var/www/sample.html') do
  it { should be_file }
  it { should contain 'test all the things!' }
end
```

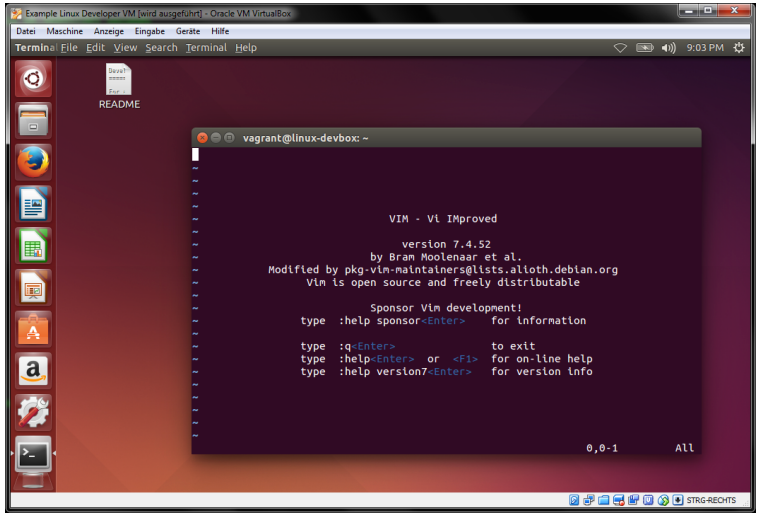
http://serverspec.org/

Beispiel: Java Entwickler VM

zühlke
empowering ideas

Eine minimale Entwickler VM mit Java, Maven und Eclipse





Template Projekt für Entwickler VMs (mit dieser Toolchain)

https://github.com/Zuehlke/linux-developer-vm

Zuehlke / **linux-developer-vm**

Unwatch 67 | Star 2 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse

Template project for a minimal Chef-managed Linux developer VM

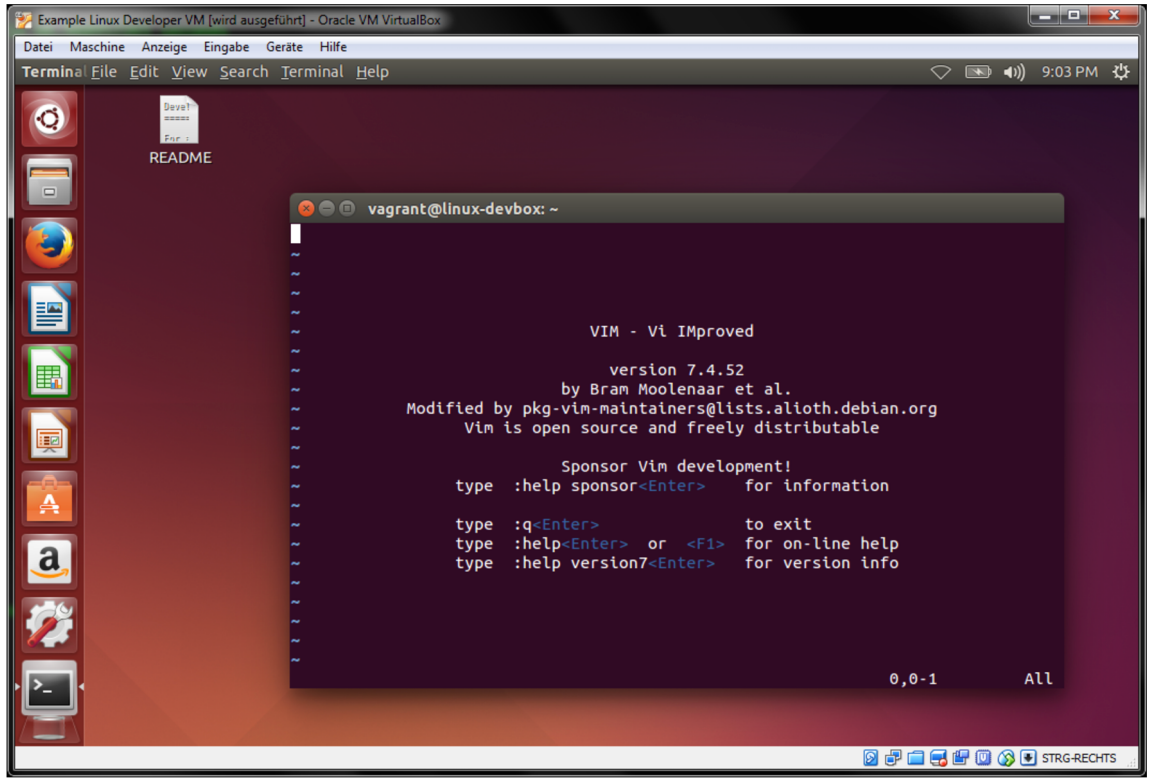
linux | development-vm | chef | vagrant | Manage topics

139 commits | 3 branches | 0 releases

Branch: master | New pull request | Create new f

tknerr committed on **GitHub** Merge pull request #20 from Zuehlke/feature/fix-cmdline-flags

cookbooks/vm	extract vim into a separate recipe / spec test instead
scripts	fix handling of commandline flags
.gitignore	more small changes taken from the linus-kitchen example
CHANGELOG.md	remove Git from this developer VM to keep it a minimum example
LICENSE	Create LICENSE
README.md	improve wording in the README
Vagrantfile	we need to cache the whole ~/.berkshelf directory, otherwise we
circle.yml	adapt circle.yml to pass the --provision-only / --verify-only flags
linux_devbox.png	add screenshot of developer VM, update vagrant output



It's meant to be copy/pasted and filled with life. The `cookbooks/vm` directory contains the recipes for setting up the VM and the tests that come along with it. All your specific customizations go in there!

Szenario: Java Entwickler VM

Wie erstellen wir eine Java Entwickler VM?

Minimal sollten diese Tools enthalten sein:

- Java
- Maven
- Eclipse

Schritte:

1. Template Projekt klonen und VM starten
2. Chef Rezepte für die Installation / Konfiguration o.g. Tools erstellen
3. Mit Serverspec Tests absichern

Szenario: Java Entwickler VM

https://github.com/tknerr/etka2017-developer-vm/pull/1

Step 1: Customize Template for "Etka Developer VM" #1

Merged **tknerr** merged 4 commits into `master` from `feature/1-customize-template-project` 12 hours ago

Conversation 0 | Commits 4 | Files changed 4

tknerr commented 12 hours ago • edited — Owner

After forking the Zuehlke/linux-developer-vm template project, you should make the following adjustments to make it an "Etka Developer VM":

1. search / replace "Linux Developer VM" with "Etka Developer VM" in `README.md` and `Vagrantfile`
2. adjust hostname in `Vagrantfile` and package name in `README.md`
3. adjust URLs tp point to tknerr/etka2017-developer-vm instead of Zuehlke/linux-developer-vm in `README.md`
4. adjust information in the vm cookbook's `metadata.rb` and `README.md`

Szenario: Java Entwickler VM

https://github.com/tknerr/etka2017-developer-vm/pull/2

Step 2: Install Oracle JDK #2

Merged tknerr merged 3 commits into `master` from `feature/2-install-ora`

Conversation 0 · Commits 3 · Files changed 5

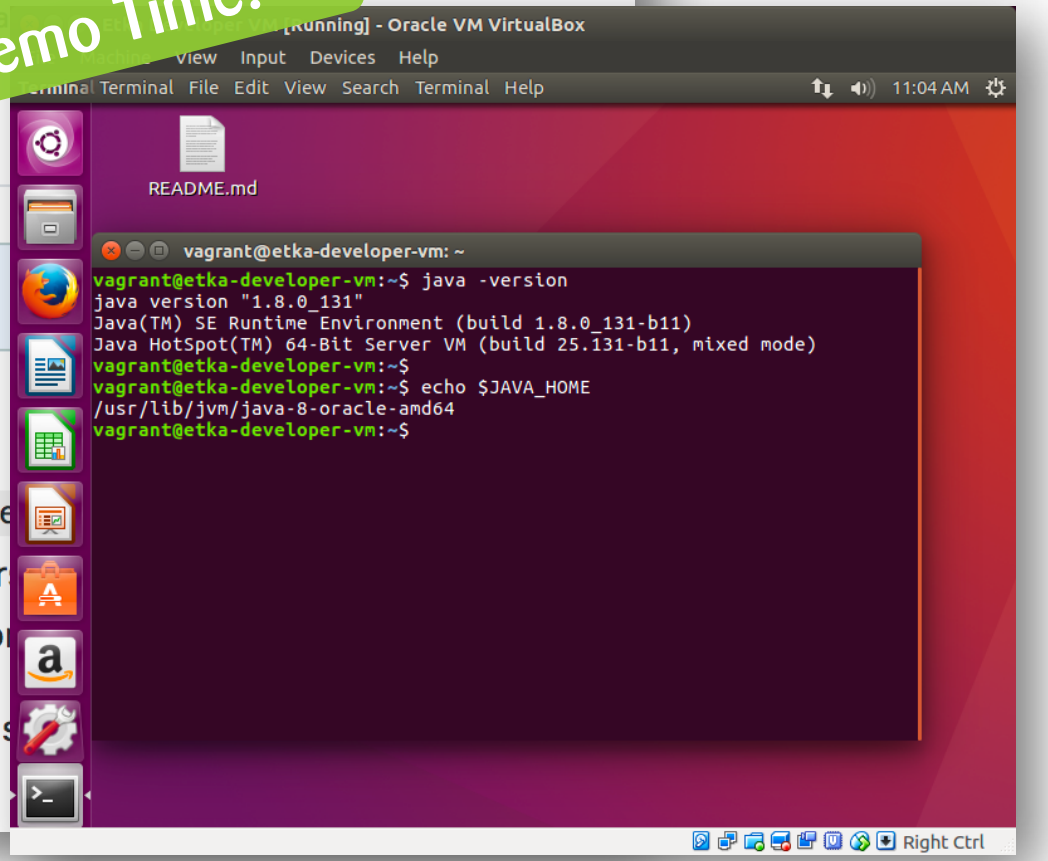
tknerr commented 11 hours ago • edited

Install Oracle JDK version 8u131 and add tests:

- add java cookbook dependency in `metadata.rb` and update `Be`
- add `cookbooks/vm/recipes/java.rb` configuring the parameter
- add `cookbooks/vm/spec/integration/recipes/java_spec.rb` fo

When running `vagrant up` / `vagrant provision` again you should s
tests passing):

Demo Time!



Szenario: Java Entwickler VM

https://github.com/tknerr/etka2017-developer-vm/pull/3

Step 3: Install Maven #3

Merged tknerr merged 5 commits into `master` from `feature/3-install-maven`

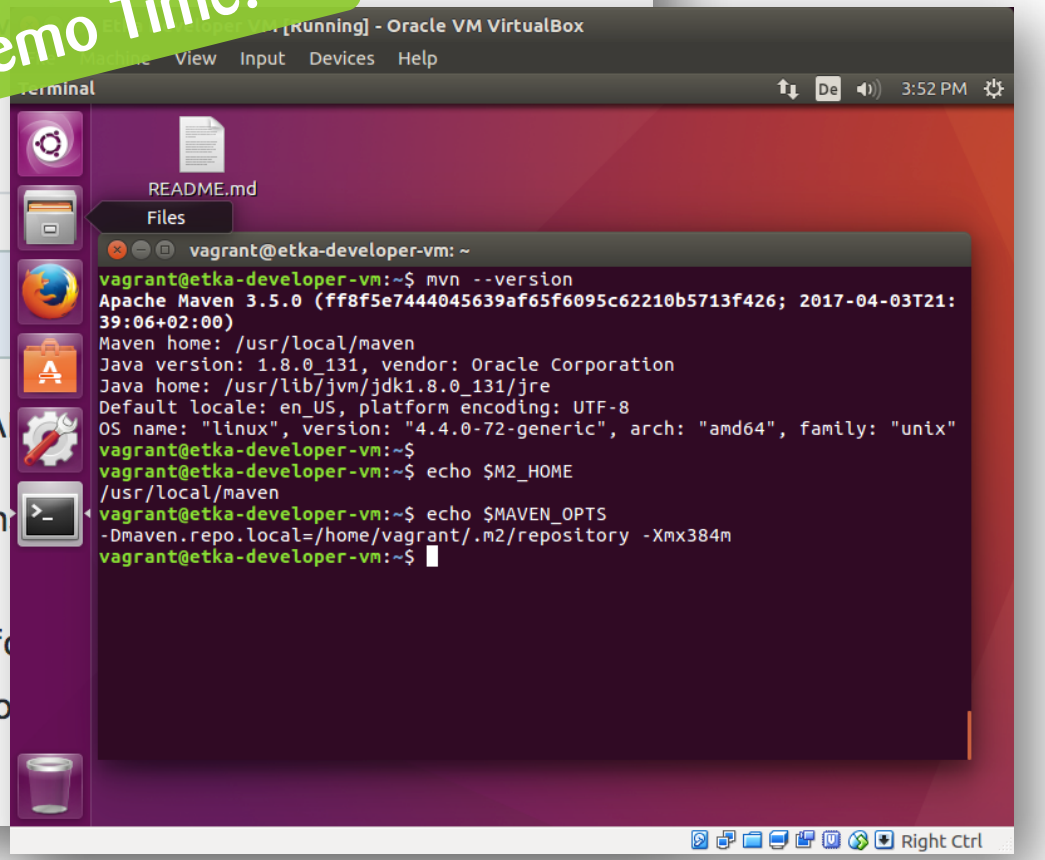
Conversation 0 | Commits 5 | Files changed 5

tknerr commented 11 hours ago • edited

Install Apache Maven 3.5.0, add tests and update the Desktop REA

- add `cookbooks/vm/recipes/maven.rb` to download / extract / in
- also set the `M2_HOME` and `MAVEN_OPTS` environment variables
- add `cookbooks/vm/spec/integration/recipes/maven_spec.rb` f
- adds instructions to the VM Desktop `README.md` with instructio deploy a sample RESTful web application using maven

Demo Time!



Szenario: Java Entwickler VM

https://github.com/tknerr/etka2017-developer-vm/pull/5

Step 4: Install Eclipse #5

Merged tknerr merged 6 commits into `master` from `feature/4-install-ec`

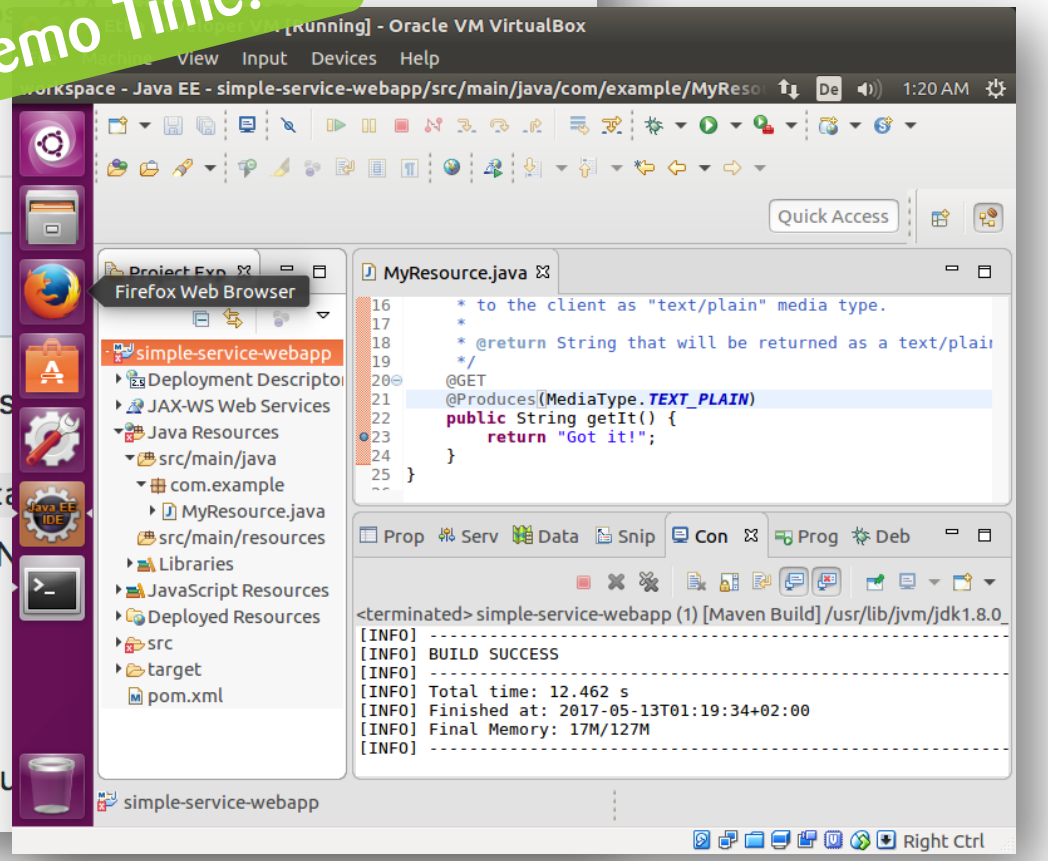
Conversation 0 | Commits 6 | Files changed 6

tknerr commented 3 hours ago • edited

Install Eclipse JEE Neon R3 (4.6.3), add tests, and update instructions

- add the ark cookbook as a dependency to `cookbooks/vm/metadata`
- use the `ark` resource to download / extract / install Eclipse JEE N `cookbooks/vm/recipes/eclipse.rb`
- add tests for the Eclipse installation in `cookbooks/vm/spec/integration/recipes/eclipse_spec.rb`
- update the desktop `README.md` with instructions how to debug ou

Demo Time!



Zusammengefasst…

- Keine Angst vor der Toolchain
- Infrastructure-as-Code ist kodifiziertes Wissen
- Deklarative Spezifikation des Zielzustands vs imperativer Skripte
- Automatisiertes Testen von Infrastruktur ist gar nicht schwer
- Vorlagen als “Kickstart” nutzen und anpassen!

Entwickler VMs funktionieren super…

- Bei großen sowie kleinen Teams
- Bei häufigem Wechsel zwischen Projekten / Toolchains
- Beim Ramp-up neuer Team Mitglieder
- Im regulierten Umfeld (Validierung)
- Auch als Windows VMs
- In Kombination mit Automatisierter CI Infrastruktur

…und helfen bei ***“works on my machine”***

Links und Ressourcen

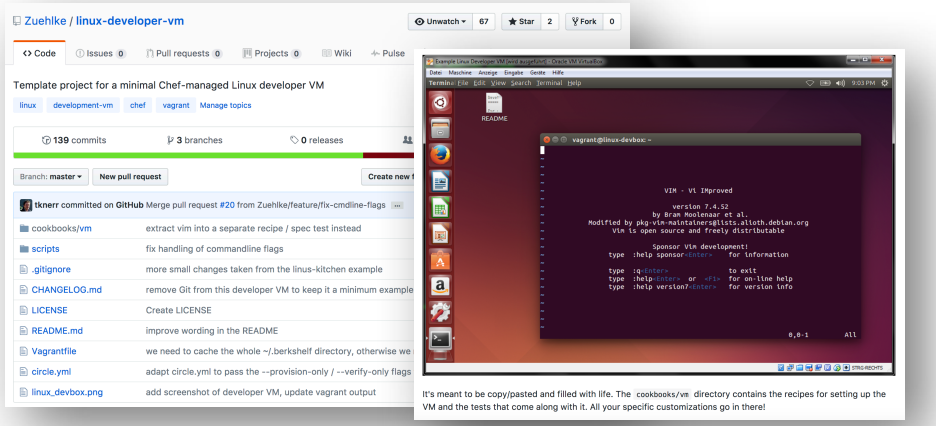
Zum Wiederverwenden und Anpassen…

Linux **Developer VM Template**:
https://github.com/Zuehlke/linux-developer-vm

Ein paar **Beispiele** für darauf basierende Entwickler VMs:
https://github.com/Zuehlke/java-developer-vm
https://github.com/tknerr/linus-kitchen

Demo Repository zu diesem Vortrag:
https://github.com/tknerr/etka2017-demo-repo

More **Open Source** @Zuehlke:
https://zuehlke.github.io



Vielen Dank! ☺

Und lassen Sie uns gerne hier weiter diskutieren: