

# **VAGRANT**

**Development environments  
made easy.**

# Was ist Vagrant?

- Ein Tool, um Entwicklungsumgebungen auf Basis von virtuellen Maschinen zu erstellen
- "Wrapper" zwischen Virtualisierungssoftware und Systemkonfigurationswerkzeugen
- In Ruby geschrieben :)
- Ein Kommandozeilen-Tool ♥

# Vagrant ist kein(e)

- Klickibunti (~~GUI~~)..zumindest standardmäßig
- eigene Virtualisierungssoftware

# Vagrant ermöglicht

- lokale Entwicklungsumgebungen
- ..die möglichst identisch zur produktiven Umgebungen sind
- schnelles erstellen/löschen von Entwicklungsumgebungen
- Provisionierung von virtuellen Maschinen (Puppet, Chef, SaltStack etc.)
- Nutzung vorkonfigurierter VMs ([VagrantBox.es](http://VagrantBox.es))

# SET UP / INSTALLATION

- VirtualBox installieren
- Vagrant installieren ([Download](#))

```
$ vagrant init hashicorp/precise64      // neue VM initialisieren
```

```
$ vagrant up      // VM starten
```

```
$ vagrant ssh      // SSH in VM
```

## War das alles?

**..grob gesehen ja, aber: Es gibt noch viel mehr**

# VAGRANTFILES - BEISPIEL

```
1. # Example 3
2. # Single box with VirtualBox provider
3. # and Puppet provisioning.
4. # NOTE: Make sure you have the precise32
5. # base box installed...
6. # vagrant box add precise32
http://files.vagrantup.com/precise32.box
7.
8. Vagrant.configure("2") do |config|
9.   config.vm.box = "precise32"
```



puppet

# Hää? Puppet?!?

Jein.. Puppet ist ein sog. Provisioning- bzw. Orchestration-Tool

Mithilfe von Puppet lässt sich der gewünschte Zustand eines Servers durch Code beschreiben. Puppet führt dann (im Optimalfall) alle Änderungen aus, welche zum Erreichen des Zustandes von Nöten sind.

# PUPPET

```
.  
├── manifests  
│   └── site.pp  
└── modules  
    ├── baseconfig  
    │   ├── files  
    │   │   └── bashrc  
    │   └── manifests  
    │       └── init.pp
```

Puppet Manifests enthält site.pp

-> Bestimmt, welche Module angewendet werden.

Puppet Module enthält mehrere Manifeste.

-> Legen fest, was geändert wird.



# PUPPET MANIFEST - BEISPIEL

```
1. # == Class: baseconfig
2. #
3. # Performs initial configuration tasks for all
Vagrant boxes.
4. #
5. class baseconfig {
6.     exec { 'apt-get update':
7.         command => '/usr/bin/apt-get update';
8.     }
9.
```

# Wozu Vagrant?

Ich habe doch <VMProvider>..

Ja, aber:

- Vagrantfiles => persistente und verteilbare Konfigurationen (Entwicklerteams, etc.)
- Vorkonfigurierte VMs/Boxes (kürzere Installationszeit, etc.)
- Erstellung von mehreren VMs auf einmal (man kann in Vagrantfiles mehrere VMs definieren)
- Automatisierte Provisionierung der VMs

**DEMO**

**FRAGEN?**

