

# DevOps

## Vagrant

Prof. Dr.-Ing. Andreas Heil

 Licensed under a Creative Commons Attribution 4.0 International license. Icons by The Noun Project.

v1.0.2

「\\_(ツ)\_/」

**IT WORKS**  
on my machine

# Was ist Vagrant?

- Tool zum automatisierten Aufsetzen von virtuellen Entwicklungsumgebungen
- Die Idee: Entwicklungsumgebung soll dem produktiven System so ähnliche wie möglich sein
- Gedankenspiel: Wie groß wäre der Unterschied zwischen Ihrer Entwicklungsumgebung und dem Zielsystem?
- Vagrant wurde in Ruby von Mitchel Hashimoto entwickelt

# Anwendungsgebiete

- Testen von Software in unterschiedlichen Umgebungen und Betriebssystemen
- Workflows mit verschiedenen Configuration Management Tools testen (Chef, Puppet, Ansible)
- Identische Umgebungen für unterschiedliche Entwickler / Team-Mitglieder aufsetzen
- Testen von Multi-Server Use Cases

# Vorteile

- Es werden sog. »Basis Images« genutzt
  - Betriebssystem muss nicht installiert werden
- Einstellungen können konfiguriert werden
- Ausführung von Automatisierungs-Tools wie Chef, Puppet oder Ansible

# Nochmal zur Erinnerung

- Wir möchten vermeidbare manuelle Tätigkeiten so gut wie möglich automatisieren
- Wozu?
  - Zeitgewinn
  - Vermeidung von Fehlern
  - Reproduzierbarkeit

# Begriffe

- Provider
  - Virtualisierungstechnologien: VirtualBox, Hyper-V, VMWare, Docker, AWS...)
- Provisionier:
  - Configuration Management Tool (Chef, Puppet, Ansible)
- Box / Vagrantbox
  - Vorlage zur Erstellung einer virtuellen Maschine (VM)
- Vagrantfile:
  - Konfigurationsfile für Vagrant

# Kommandos

- `box {add|remove|prune..}` Fügt eine Box hinzu, entfernt diese etc.
- `global-status` Status aller Vagrant-Umgebungen
- `init` Initialisiert das aktuelle Verzeichnis für eine Box
- `halt` Schaltet die virtuelle Maschinen(n) ab
- `provision` Führt den Provisioner gegen die VMs aus
- `reload` Neustart nach Änderungen
- `ssh` Shell Zugriff auf die Maschine
- `up` Erstellt und Startet die VM gemäß dem Vagrantfile



# Standard Boxes

- <https://app.vagrantup.com/boxes/search>
- Öffentliche Boxes sind kostenlos
- Persönlicher bzw. Unternehmens-Account ermöglicht das Speichern von privaten Boxes

# Multi-Server Setup

Mehrere identische Maschinen möglich:

```
Vagrant.configure("2") do |config|  
  
  # test server  
  config.vm.define "test" do |test|  
    db.vm.box = "ubuntu/xenial64"  
  end  
  
  # performance test server  
  config.vm.define "perf" do |perf|  
    web.vm.box = "ubuntu/xenial64"  
  end  
  
end
```

# Port Forwarding

Gleiches Verhalten auf allen Maschinen

- Standard-Ports auf den virtuellen Maschinen nutzbar
- Ports vom Host-System werden weitergeleitet

```
Vagrant.configure("2") do |config|  
  ...  
  config.vm.network :forwarded_port, guest: 80, host: 8080  
  ...  
end
```

# IaC / Anpassbar

Konfiguration<sup>1</sup> am Beispiel der VMBox Einstellungen

```
Vagrant.configure("2") do |config|  
  ...  
  config.vm.provider "virtualbox" do |v|  
    v.memory = 2048  
    v.cpus = 2  
  end  
  ...  
end
```

# Vagrant und Ansible

Ansible als Provisioner<sup>2</sup>:

- Vorteil Playbook kann automatisch gegen Box ausgeführt werden

```
Vagrant.require_version ">= 1.8.0"

Vagrant.configure(2) do |config|

  config.vm.box = "ubuntu/bionic64"

  config.vm.provision "ansible" do |ansible|
    ansible.verbose = "v"
    ansible.playbook = "playbook.yml"
  end
end
```

# Provisionierung

- `bootstrap.sh` im Verzeichnis des Vagrantfile

```
#!/usr/bin/env bash

apt-get update
apt-get install -y apache2
if ! [ -L /var/www ]; then
    rm -rf /var/www
    ln -fs /vagrant /var/www
fi
```

# Base Box erstellen

Am Beispiel VirtualBox<sup>3</sup>

- Jede in VirtualBox erstellte VM kann verwendet werden
- Harte Anforderungen Seitens Vagrant
  - Der erste Netzwerkadapter muss zwingend ein NAT-Adapter sein
  - MAC Adresse der VM muss bekannt sein, wird später im Vagrantfile eingetragen ( `config.vm.base_mac` )

# Vagrant vs. Docker

- Docker: Bauen, ausliefern und betreiben von verteilten Anwendungen
  - Container laufen im bzw. nutzen das Host-Betriebssystem
- Vagrant: Leichtgewichtiges Tool um virtuelle Umgebungen zu konfigurieren und zu erstellen
  - Virtuelle Maschinen bringen Ihr Betriebssystem mit
  - Docker kann als Provider für Vagrant genutzt werden



# Voraussetzungen

- Provider
  - Z.B. VirtualBox
  - Vagrant installieren
  - Vagrant Box download

# Demo

# Referenzen