
Contao in a box

— Entwicklungsumgebung für —
Contao mit Vagrant

von Claudio De Facci
[exploreimpact.de](https://www.exploreimpact.de)

Was ist Vagrant?



Vagrant ist:

- eine Software zur einheitlichen **Erstellung von virtuellen Maschinen**
- Plattformunabhängig (verfügbar für **OSx, Windows und Linux**)
- Wrapper zwischen Provider (hier **Virtual Box**) und Software-Configuration-Management-Anwendungen (hier **Puppet**)
- Besonders geeignet für 'dev'- und 'testing'-Umgebungen, nicht unbedingt für 'prod'

Vorteile

- Schnell ein (Linux-)OS als VM aufsetzen
- Nur initial leicht erhöhter Setup-Aufwand
- Schnell zurücksetzbar auf Default-Werte
- Möglichkeit unterschiedliche Entwicklungsumgebungen parallel aufzusetzen
- Sandboxing ohne Risiko das System ernsthaft zu beschädigen
- Möglichkeit eine einheitliche Entwicklungsumgebung im Team zu verteilen (Base Box)

- Gute Möglichkeit Linux und die Arbeit mit CLI-Tools kennenzulernen

Setup in Windows 10

- Prüfen, ob **VT-X** im BIOS/UEFI aktiv ist
- **Virtual Box** installieren <https://www.virtualbox.org/>
- **Vagrant** installieren <https://www.vagrantup.com/>
- **git for Windows** <https://git-scm.com/download/win>
- **PuTTY** <http://www.putty.org/>
- ggf. **MySQL Workbench** <https://www.mysql.de/products/workbench/>

Eine Box zusammenklicken mit PuPHPet.com

PuPHPet.com = **Puppet** mit UI und PHP-basierter Config-Erstellung

Vorteile:

- **Color-Bash** mit git-Support
- Wichtige **System-Packages** sind **vorinstalliert** und vorkonfiguriert
 - sudo
 - nano
 - git
 - ...
- Konfigurationen sind auf **Development eingestellt**
(z.B. Fehlerausgabe auf 100%)

Eine Konfiguration zusammenklicken

Meine Anpassungen für eine Standard-Config für PHP-Projekte
(Beispielwerte):

- Internal Identifier: contao-project
- IP: 10.10.4.15
- Folder Source: ../http
- PHP + gd
- Hostname: contao-project.local
- RAM: 2048 MB
- Apache + Hosts + mod headers
- MySQL + User & Database

Los geht's...

<https://puphpet.com/>

Ordnerstruktur

Boxen sinnvoll ablegen:

Zum Beispiele unter **C:\Users\\Documents\VMs**

- <Projektname>/
 - box/ Vagrant-Dateien auf ZIP von PuPHPet
 - http/ Shared Folder

Konfiguration prüfen

- box/puphpet/vagrant/**Vagrantfile-local**

- fmode & dmode auf 0777 setzen

- Symlink-Ability

```
data['vm']['synced_folder'].each do |i, folder|
  virtualbox.customize ['setextradata',
                        :id, "VBoxInternal2/SharedFoldersEnableSymlinksCreate//#{i}", '1'
  ]
end
```

- box/puphpet/vagrant/**config.yaml**

- file_mode & dir_mode auf 0777 setzen

Ausführrechte prüfen

Executables mit Admin-Rechten starten lassen:

- Virtual Box
 - C:\Program Files\Oracle\VirtualBox**VBoxHeadless.exe**
- git Bash
 - C:\Program Files\Git**git-bash.exe**

Box starten

In box/-Verzeichnis:

```
$ vagrant up
```

Host-Konfiguration im Host-OS anpassen

- Editor mit Adminrechten starten
- Hosts anpassen/ergänzen:
 - C:\Windows\System32\drivers\etc\hosts

10.10.4.15 project.dev

Wichtig:

Microsoft Edge ignoriert die hosts-Datei unter Win10. Lokale Entwicklung sollte also mit Chrome und Firefox stattfinden.

Box initial einrichten

- Passwort für 'vagrant'-Nutzer ändern
 - **vagrant ssh**
 - **passwd vagrant**
 - Passwort auf '**vagrant**' setzen
- Per PuTTY einloggen
- Public(!)-Key bei BitBucket/GitHub hinterlegen
 - **cat ~/.ssh/id_rsa.pub**

Box testen

- Default-Host testen
 - <http://10.10.4.15> im Browser öffnen
 - Eine Default-Seite sollte zu sehen sein.
- Shared folder testen
 - Dateien in Shared Folder legen über Windows
 - In VM gucken ob alle Dateien da sind:
 - **cd /var/www**
 - **ls -la**

Contao (3.5.x) in der Box

- Contao herunterladen
<https://contao.org/de/download.html>
- Files in Shared Folder der VM packen: http/project.dev
- Installation starten
<http://project.dev/contao/install.php>
- Von hieraus alles wie gehabt...
(Datenbank wie bei PuPHPet angegeben verwenden, oder externe DB.)

Box stoppen

In box/-Verzeichnis:

\$ vagrant halt

Häufig verwendete Befehle in der Übersicht

- `vagrant ssh` Einfaches einloggen auf VM via SSH
- `vagrant up` VM starten/booten
- `vagrant halt` VM anhalten/herunterfahren
- `vagrant reload` VM neu starten (mit Änderungen an Config)
- `vagrant provision` VM resetten auf Config-Werte
- `vagrant destroy` VM zerstören

Base-Box erstellen

Vorbereitung in der VM

- Apt-Cache leeren
sudo apt-get clean
- zero out the hard drive (on Ubuntu)
sudo dd if=/dev/zero of=/EMPTY bs=1M
sudo rm -f /EMPTY
- Clear bash history
cat /dev/null > ~/.bash_history && history -c && exit

Base-Box erstellen

Erstellen der Box im Host-System

- Box packen

vagrant package --output mynew.box

Die gepackte Box-Datei kann nun ans Team versendet werden.
Idealerweise wird diese jedoch noch vorher lokal getestet.

Base-Box erstellen

Nutzen der Box im Host-System

- Add box to your box list

```
vagrant box add my-box /path/to/the/mynew.box
```

- Initialize a new VM with your new image

```
vagrant init my-box
```

- Start the new box

```
vagrant up
```

Ende